A Data Mining Approach to Identify Associations Between Job Titles and Skills in Job Vacancies

Sadeepa Kuruppu
Division of Information Technology
Institute of Technology
University of Moratuwa
Diyagama, Homagama, Sri Lanka
sandeepak@itum.mrt.ac.lk

Kalpana Galappaththi
Division of Information Technology
Institute of Technology
University of Moratuwa
Diyagama, Homagama, Sri Lanka
kgalappaththi@itum.mrt.ac.lk

Abstract— **At present, job vacancies appear in online job vacancy repositories. They are available in the form of images or in the form of Portable Digital Format Documents. Text information is embedded in them. By mining that text information, current dynamics of the job market can be identified. As the Information Technology industry is dynamic, it is worth understanding what associations exist between job titles and technologies that frequently appear together, when applying for a job. For this purpose two algorithms can be used. They are Apriori algorithm and Frequent Pattern Growth algorithm. Among these two algorithms, this study emphasizes the importance of using Frequent Pattern Growth algorithm because it has eliminated the issue of performing so many scans in the database, which lead the Apriori algorithm less efficient. Frequent Pattern Growth algorithm used to mine association rules which exist between job titles and technologies required for them. The aim of the study is to mine how technologies appear associated with each other in job vacancies. Job seekers can be aware of what technologies have association trends in the job market and that would be helpful to reduce the gap that exists between skills of job seekers and industry demands.**

Keywords—Data mining, Association Rule Mining, Apriori, FP Growth

I. INTRODUCTION

Identifying the associations with job titles and technologies required for different jobs will be helpful for job seekers to match their qualifications, technical skills, and knowledge with job opportunities. Then job seekers will be able to upgrade their technical qualifications for jobs. Educational authorities can update their curriculum in accordance with the current trends in the job market. Identifying and following the associations between the job titles and technologies take special importance for the Information Technology (IT) profession, as new technologies are emerging day by day. Therefore, it is important if there is a way those job seekers can identify what technologies are appearing in association in the IT industry job market. Job holders, job seekers, and youth will be able to adjust their technical skills in order to adhere to the current dynamics in the IT industry [1-3].

In order to mine hidden patterns, various algorithms should be applied to the data source. This research has focused on one of the descriptive data mining techniques, which is known as the Association Rule Mining (ARM). Association Rule Mining is a data mining technique which is used to find frequent patterns, correlations, or associations among the sets of the items in a database and also it is used to discover relationships among a large set of variables in a data set [4]. In this study, ARM was used to derive the associations with job titles and technologies related to a particular job and to mine how technologies are associated with each other.

II. BACKGROUND

When mining the association rule, two major algorithms are taken into consideration. They are “Apriori algorithm” and “Frequent Pattern” (FP Growth) algorithm.

The “Apriori” algorithm was introduced by “Agrawal” and “Srikant” [5][38]. The name “Apriori” has been given to this algorithm because it uses a-prior knowledge of frequent itemsets properties [6-10]. Fig. 1, shows the process that happens in “Apriori” algorithm.

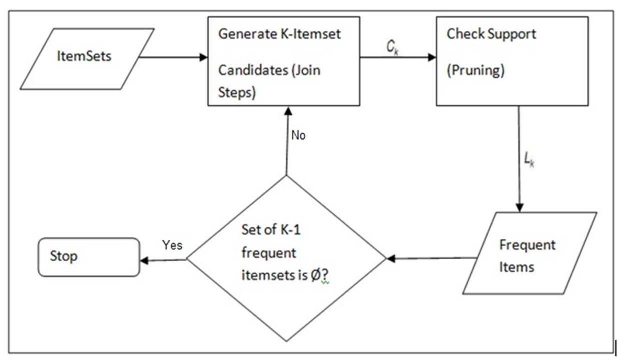


Fig. 1. Flowchart of “Apriori” algorithm

The drawback of the “Apriori” algorithm is that it needs to scan the whole database many times; also it becomes slow when the database becomes large [7] [12]. To perform many scans in the database needs much computing power, in that context “Apriori” is less efficient when the database be-comes larger.

The algorithm used in this study was the “Frequent Pattern-Growth (FP- Growth)” algorithm because it has eliminated the above mentioned weaknesses which exist in the “Apriori” algorithm. When comparing the functionalities of two algorithms; “Apriori” and FP Growth, the FP growth algorithm was much effective than the “Apriori” algorithm because, the FP- Growth algorithm makes fewer scans of the database, making it practically usable for large databases like text [13]. FP-Growth algorithm was introduced by Han et al in 2000 [14]. The FP-Growth algorithm can be introduced as an improved version of the “Apriori” algorithm. The FP Growth algorithm avoids the candidates generating process and passes fewer scans over the database [15].

FP Growth algorithm uses an extended pre-fix tree structure, called the FP tree [16]. The purpose of the FP tree is to mine the most frequent pattern [17]. FP-Tree is a compressed representation of the original database, and also only frequent items are used to construct the FP- tree, and the other irrelevant information is pruned. FP Tree is made up of initial itemsets of the database. Each node of the FP tree represents an item of item set. In the FP tree root node represents null while lower nodes represent itemsets [17]. FP-Tree is much smaller than the original database and reduces the required number of database scans to two [18].

During the process of FP growth algorithm at first database is scanned, to find frequent 1-itemset. Here, single items in itemset are captured. Based on the minimum threshold support value, all the items that are less than the minimum threshold support value are removed when finding frequent items [17]. Then the selected items are stored according to their descending values, that list is known as F-List. After that FP Tree is created based on F-List. As the last step, the conditional FP tree is created in the sequence of reverse order of F-List [19-27].

```
Procedure FP-Growth(Tree, α)
// α is an itemset in transactional database
// β is an itemset in α's conditional pattern-base
{
if Tree contains a single prefix path                    // Mining single prefix-path FP-tree
   then {
         let P be the single prefix-path part of Tree;
         let Q be the multipath part with the top branching node replaced by a null root;
         for each combination (denoted as β) of the nodes in the path P do
            generate pattern β ∪ α with support = minimum support of nodes in β;
         let freq_pattern_set(P) be the set of patterns so generated; }
   else let Q be Tree;
for each item a_i in Q do {                               // Mining multipath FP-tree
    generate pattern β = a_i ∪ α with support = a_i .support;
    construct β's conditional pattern-base and then β's conditional FP-tree Treeβ ;
    if Treeβ = ∅
      then call FP-Growth(Treeβ, β);
     let freq pattern set(Q) be the set of patterns so generated; }
return(freq_pattern_set(P) ∪ freq_pattern_set(Q) ∪ (freq_pattern_set(P) × freq_pattern_set(Q)))
```

Fig. 2. FP-Growth algorithm pseudocode

Fig. 2 shows the pseudocode of the FP Growth algorithm. The FP algorithm contains two major parameters. They are Tree = FP-tree, and α = null. When FP-tree has only one path P, for each combination of β vertex path P is created by using a set of β ∪ α. For that support equal to the minimum support of items which are belong to the setβ are used. When FP-tree is having multiple paths, In that case for each element in the array, αi Tree header is created with a set of β = αi ∪ α supporting corresponding elements αi. Next, the conditional pattern base of β and conditional FP-tree pattern of β are generated. That is denoted by Treeβ. Then, it is verified whether Treeβ is empty or not. If it is empty, the algorithm ends, If it is not empty, the procedure FP-Growth is restarted with parameters of Tree = Treeβ, and α = β. The last statement of the procedure returns the three sets of the generated frequent patterns from P, Q, and P×Q [7],[16-23].

III. METHODOLOGY

To identify the associations between job titles and technologies in the IT industry, this study has used the Frequent Pattern (FP-Growth) algorithm as the Association Rule Mining algorithm. In the study, mining associations were done to identify what technologies that are associated with job titles, and how technologies are associated with each other in the IT industry.

An association rule consists of two components. Antecedent (if) is the first component. Consequent (then) is the second component. An antecedent is an item found within the data. Consequent is the item found in combination with the antecedent [24-28].

The association rule consists of two steps. The first step is used to find all frequent itemsets. The second step is used to generate strong associations from frequent itemsets [13].

An association rule implies in the form of X⇒Y, where X, Y⊂ I are sets of items called itemsets and X ∩ Y = ø. Here, X is known as the antecedent, and Y is known as the consequent [31 -32]. The Association Rule Mining was first introduced by Agrawal in 1993 [5]. According to Agrawal, Let I=I1, I2, · · · , Im be a set of m distinct attributes, T be the transaction that contains a set of items such that T ⊆ I, D be a database with different transaction records Ts [13].

The two important basic measures in association rules are support and confidence [20-21]. Support of an association rule is defined as the percentage of records that contain X ∪ Y to the total number of records in the database D [31-33]. Support value is calculated as given in "Equation 1".

In an association rule, confidence is defined as the percentage of the number of transactions that contain X∪Y to the total number of records that contain X [29]. Confidence shows the items that appear one after the other. Also, confidence can be defined as an indication between two items that how often the mined rules are found to be true. Rules with support value greater than user-defined minimum support value and confidence value greater than user-defined minimum confidence value are called valid association rules [34]. The confidence value is calculated as given in "Equation 2".

The lift value represents whether the association is positive or negative also shows the strength of the relationship which exists between X and Y. If the lift value is greater than one, it indicates a positive relationship between the itemsets. If the lift value is less than one, it indicates a negative relationship, and if the lift value equals one, the itemsets are independent and there exists no relationship between them [34]. Lift value is calculated as given in "Equation 3".

$$\text{Support (XY)} = \frac{\text{Support count of XY}}{\text{Total number of transactions in D}} \tag{1}$$

$$\text{Confidence}(X|Y) = \frac{\text{Support (XY)}}{\text{Support (X)}} \tag{2}$$

$$\text{Lift (XY)} = \frac{\text{P (XY)}}{\text{P(X)P(Y)}} \tag{3}$$

The trained dataset consists of five hundred job vacancies. The data set was collected using an online site that advertises IT industry job vacancies in Sri Lanka. The texts embedded in those images were extracted by using Optical Character Recognition Algorithm (OCR). The extracted data from job vacancy images needed to preprocess as they do not conform to a uniform structure. So, the texts were subjected to a pre-processing process to remove unwanted noise from text data and filter the necessary text data. During the text pre-processing stage first, the texts were subjected to the tokenization process. In the process of tokenization, sentences are separated into words [35]. After that stop words are being removed from the text. Stop words can be introduced as determiners, prepositions, and conjunctions. They do not contribute to the content or context of the textual document. As the last stage of pre-processing, stemming is done. During the process of stemming, prefixes and suffixes are removed. The purpose of stemming is to find the root or the base of a word. The cleansed data can be used in the data mining process.

“RapidMiner”, which is a data science and machine learning platform was used as an integrated environment in the data mining process [36]. Fig. 3 shows the processes used in “RapidMiner” to mine the association rules. Here four processes were used. They are “Retrieve data set, Numerical to binomial, FP-Growth, and Create association rules”. Using the “Retrieve Data Set” operator data set was loaded to the “RapidMiner”. “Numerical to Binomial” operator was used to change numerical attributes that appear in the dataset to binomial type. “FP-Growth” operator was used to calculate all frequent itemsets using FP-tree. “Create Association Rules” operator was used to generate the association rules from the given dataset [36].

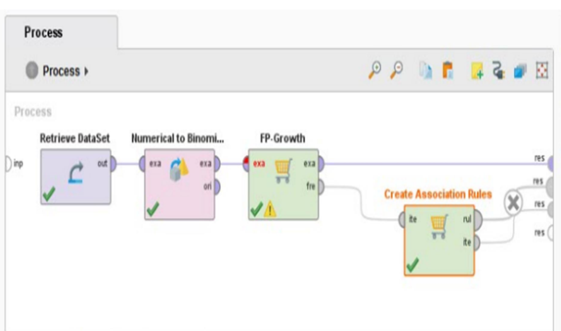


Fig. 3. Processes in “RapidMiner”

IV. RESULTS

Fig. 4-12 shows the results obtained after training the dataset with “RapidMiner”. Fig. 4 shows some of the frequent itemsets which were mined with their support value. Item 1 is the antecedent value while item 2, item 3, and item 4 are known as the consequent values. According to the last record in Fig. 4, the items HTML, JavaScript, CSS, and Web developer are associated with each other with the support value of 0.213, that means out of the total records given in the data set a ratio of 0.213 contains the items HTML, JavaScript, CSS, and Web developer together. Also, when the item “HTML” appears in the dataset, the items “JavaScript, CSS, and Web developer” followed with it or they can be defined as the consequents of the item “HTML”.

FrequentItemSets (FP-Growth)

Size	Support	Item 1	Item 2	Item 3	Item 4
2	0.235	CSS	WebDeveloper		
2	0.213	PHP	WebDeveloper		
2	0.228	Android	MobileppDeveloper		
2	0.228	Android	ios		
2	0.213	C	C#		
2	0.228	MobileppDeveloper	ios		
3	0.232	HTML	javascript	CSS	
3	0.199	HTML	javascript	PHP	
3	0.217	HTML	javascript	WebDeveloper	
3	0.202	HTML	CSS	PHP	
3	0.232	HTML	CSS	WebDeveloper	
3	0.202	HTML	PHP	WebDeveloper	
3	0.217	javascript	CSS	WebDeveloper	
3	0.199	javascript	PHP	WebDeveloper	
3	0.228	Android	MobileppDeveloper	ios	
4	0.213	HTML	javascript	CSS	WebDeveloper

Fig. 4. Frequent itemsets with their support values

Fig. 5 and Fig. 6 show all the association rules that were mined from the dataset. Each rule given here has indicated its relevant confidence value. If the confidence value is higher, then that the validity of that rule also increases. The first rule indicated in Fig. 5 shows that the confidence value between the items HTML, CSS, and JavaScript, web developer is 0.806. In this association rule, HTML and CSS are the antecedents whereas JavaScript and web developer are the consequents.

AssociationRules

```
Association Rules
[HTML, CSS] --> [javascript, WebDeveloper] (confidence: 0.806)
[CSS] --> [javascript, WebDeveloper] (confidence: 0.808)
[HTML] --> [javascript, CSS] (confidence: 0.829)
[HTML] --> [CSS, WebDeveloper] (confidence: 0.829)
[javascript] --> [WebDeveloper] (confidence: 0.838)
[WebDeveloper] --> [PHP] (confidence: 0.841)
[WebDeveloper] --> [HTML, javascript, CSS] (confidence: 0.841)
[HTML] --> [javascript] (confidence: 0.842)
[HTML, javascript] --> [PHP] (confidence: 0.844)
[HTML, WebDeveloper] --> [PHP] (confidence: 0.846)
[javascript] --> [HTML, CSS] (confidence: 0.851)
[WebDeveloper] --> [HTML, javascript] (confidence: 0.855)
[WebDeveloper] --> [javascript, CSS] (confidence: 0.855)
[HTML] --> [WebDeveloper] (confidence: 0.855)
[CSS] --> [HTML, javascript] (confidence: 0.863)
[CSS] --> [HTML, WebDeveloper] (confidence: 0.863)
[javascript] --> [HTML] (confidence: 0.865)
[javascript] --> [CSS] (confidence: 0.865)
[javascript, WebDeveloper] --> [PHP] (confidence: 0.871)
[HTML, CSS] --> [javascript] (confidence: 0.875)
[HTML, CSS] --> [WebDeveloper] (confidence: 0.875)
[CSS] --> [javascript] (confidence: 0.877)
[CSS] --> [WebDeveloper] (confidence: 0.877)
[HTML, WebDeveloper] --> [javascript, CSS] (confidence: 0.892)
[WebDeveloper] --> [javascript] (confidence: 0.899)
[HTML, javascript] --> [CSS, WebDeveloper] (confidence: 0.906)
[javascript, CSS] --> [HTML, WebDeveloper] (confidence: 0.906)
```

Fig. 5. Support value between items

The confidence value indicates the reliability of mined rules. The last rules given in Fig.6, indicates as follows;

```
[MobileAppDeveloper, ios] --> [Android]
(confidence: 1.000)
```

That rule represents 100% of the times where “Mobile app developer”, and “IOS” items appear, with them the item Android also appear together.

```
[javascript, WebDeveloper] --> [HTML, CSS] (confidence: 0.935)
[WebDeveloper] --> [HTML] (confidence: 0.942)
[HTML] --> [CSS] (confidence: 0.947)
[HTML, PHP] --> [javascript] (confidence: 0.947)
[javascript, PHP] --> [HTML] (confidence: 0.947)
[javascript, PHP] --> [WebDeveloper] (confidence: 0.947)
[PHP, WebDeveloper] --> [HTML] (confidence: 0.948)
[javascript, WebDeveloper] --> [HTML] (confidence: 0.952)
[javascript, WebDeveloper] --> [CSS] (confidence: 0.952)
[HTML, PHP] --> [CSS] (confidence: 0.965)
[HTML, PHP] --> [WebDeveloper] (confidence: 0.965)
[Android] --> [MobileppDeveloper] (confidence: 0.969)
[Android] --> [ios] (confidence: 0.969)
[Android] --> [MobileppDeveloper, ios] (confidence: 0.969)
[HTML, WebDeveloper] --> [CSS] (confidence: 0.969)
[HTML, javascript, WebDeveloper] --> [CSS] (confidence: 0.983)
[javascript, CSS, WebDeveloper] --> [HTML] (confidence: 0.983)
[HTML, javascript] --> [CSS] (confidence: 0.984)
[javascript, CSS] --> [HTML] (confidence: 0.984)
[CSS, WebDeveloper] --> [HTML] (confidence: 0.984)
[CSS] --> [HTML] (confidence: 0.986)
[MobileppDeveloper] --> [Android] (confidence: 1.000)
[ios] --> [Android] (confidence: 1.000)
[MobileppDeveloper] --> [ios] (confidence: 1.000)
[ios] --> [MobileppDeveloper] (confidence: 1.000)
[CSS, PHP] --> [HTML] (confidence: 1.000)
[MobileppDeveloper] --> [Android, ios] (confidence: 1.000)
[Android, MobileppDeveloper] --> [ios] (confidence: 1.000)
[ios] --> [Android, MobileppDeveloper] (confidence: 1.000)
[Android, ios] --> [MobileppDeveloper] (confidence: 1.000)
[MobileppDeveloper, ios] --> [Android] (confidence: 1.000)
```

Fig. 6. Support value between items

Fig. 7 shows the FP-Growth tree structure. This is the graphical representation of the entire database in a compressed form. Between the associated items, there is the relevant association rule number indicated.

Fig. 7. FP-Growth tree

Fig. 8 shows a segment of the FP-Growth tree structure. It represents the association rule that exists between the two items C and C#. When the rule is being selected the support, confidence, lift, and other related values are indicated. As the lift value given here is 3.97 it denotes a positive relationship between the two items and also the confidence value is very high which is 0.92, this can be introduced as a strong association rule.

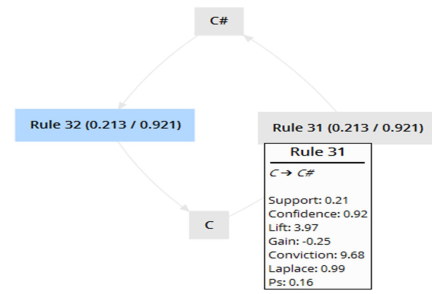
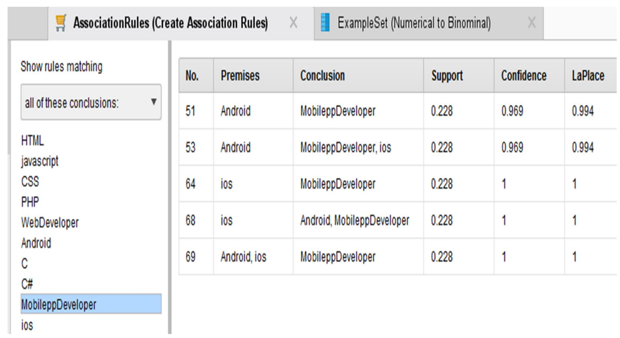


Fig. 8. Support value and Confidence value exist between rules

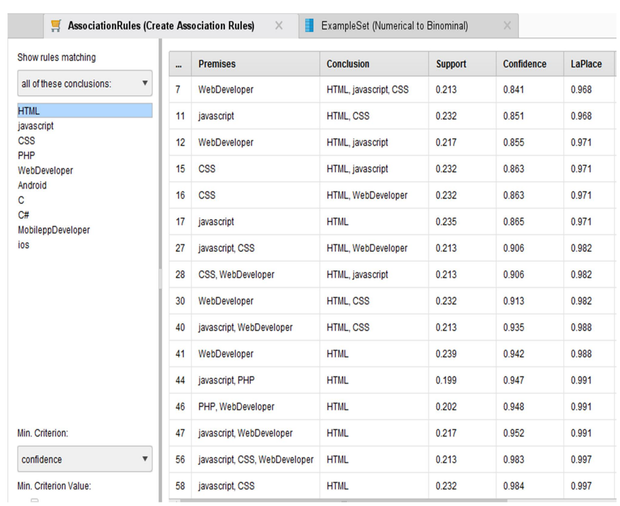
Fig. 9 shows the associated items with the item "Mobile app developer". The support value and confidence value indicated between the items are given in front of each record [37].



No.	Premises	Conclusion	Support	Confidence	LaPlace
51	Android	MobileppDeveloper	0.228	0.969	0.994
53	Android	MobileppDeveloper, ios	0.228	0.969	0.994
64	ios	MobileppDeveloper	0.228	1	1
68	ios	Android, MobileppDeveloper	0.228	1	1
69	Android, ios	MobileppDeveloper	0.228	1	1

Fig. 9. Support value and Confidence value exist between items: "Android", "mobile app developer", and "IOS"

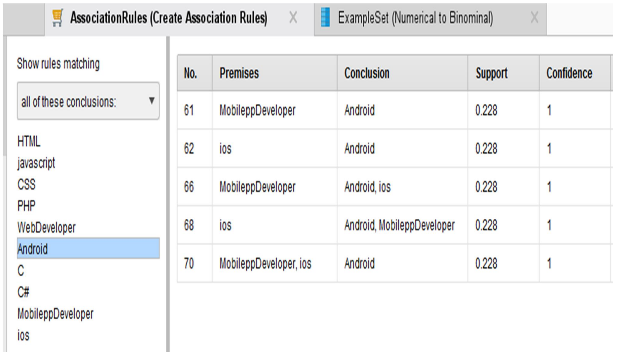
Fig. 10 shows the associated items with the item "HTML". The items associated with "HTML" are: "CSS, Web developer, and JavaScript". Fig. 9 shows different combinations of the item about how they have been appearing with their support value and confidence value.



...	Premises	Conclusion	Support	Confidence	LaPlace
7	WebDeveloper	HTML, javascript, CSS	0.213	0.841	0.968
11	javascript	HTML, CSS	0.232	0.851	0.968
12	WebDeveloper	HTML, javascript	0.217	0.855	0.971
15	CSS	HTML, javascript	0.232	0.863	0.971
16	CSS	HTML, WebDeveloper	0.232	0.863	0.971
17	javascript	HTML	0.235	0.865	0.971
27	javascript, CSS	HTML, WebDeveloper	0.213	0.906	0.982
28	CSS, WebDeveloper	HTML, javascript	0.213	0.906	0.982
30	WebDeveloper	HTML, CSS	0.232	0.913	0.982
40	javascript, WebDeveloper	HTML, CSS	0.213	0.935	0.988
41	WebDeveloper	HTML	0.239	0.942	0.988
44	javascript, PHP	HTML	0.199	0.947	0.991
46	PHP, WebDeveloper	HTML	0.202	0.948	0.991
47	javascript, WebDeveloper	HTML	0.217	0.952	0.991
56	javascript, CSS, WebDeveloper	HTML	0.213	0.983	0.997
58	javascript, CSS	HTML	0.232	0.984	0.997

Fig. 10. Support value and Confidence value exist between items: "HTML", "JavaScript", "CSS", and "Web developer"

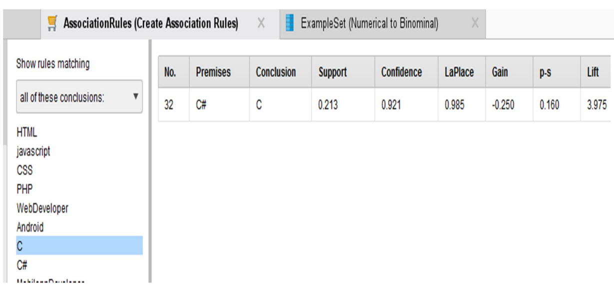
Fig. 11 shows the associated items with the item "Android". The related items with "Android" are "Mobile app developer" and "IOS'. The support value and the confidence value that exist between the items are given in front of each record.



No.	Premises	Conclusion	Support	Confidence
61	MobileppDeveloper	Android	0.228	1
62	ios	Android	0.228	1
66	MobileppDeveloper	Android, ios	0.228	1
68	ios	Android, MobileppDeveloper	0.228	1
70	MobileppDeveloper, ios	Android	0.228	1

Fig. 11. Support value and Confidence value exist between items: “Android”, “mobile app developer”, and “IOS”

Fig. 12 shows the associated items with the item “C”. The related item with “C” is only “C#”. This rule can be defined as a very strong rule because its confidence value is very high which is 0.921, as well as, this rule is having a positive lift value. The support value, confidence value, LaPlace value, Gain value, p-s value, and Lift value that exists between the items are given in front of each record.



No.	Premises	Conclusion	Support	Confidence	LaPlace	Gain	p-s	Lift
32	C#	C	0.213	0.921	0.985	-0.250	0.160	3.975

Fig. 12. Support value, Confidence value, LaPlace value, Gain value, p-s value, and Lift value exist between items: “C#”, and “C”

V. CONCLUSION

This study aims to find the trending association rules that exist in the IT industry job market. For that, the text embedded on job vacancies was extracted and they were subjected to a data mining process by using the FP-Growth algorithm. In this study, “RapidMiner” was used as a data mining platform. The results obtained were discussed in the “Results” section. In this study, most trending job titles and technologies that frequently appeared with them were mined. Mining the trending job titles and trending technologies will be helpful for job seekers to get a better understanding of the industry dynamics. This is very specific to the IT industry as it is having varieties of emerging technologies day by day.

REFERENCES

1. Wowczko, I. (2015). Skills and Vacancy Analysis with Data Mining Techniques.*Informatics*,*2* (4),31–49.
2. T Smith, D., & Ali, A. (2014). Analyzing computer programming job-trend using web data mining. *Issues in Informing Science and Information Technology*, *11*, 203-214.
3. Sibarani, E. M., Scerri, S., Morales, C., Auer, S., & Collarana, D. (2017). Ontology-guided Job Market Demand Analysis. *Proceedings of the 13th International Conference on Semantic Systems - Semantics2017*.
4. Thilagavathi, K., & Shanmugapriya, V. (2014). A survey on text mining techniques. *International Journal of Study in Computer Applications and Robotics,* 41-49.
5. Srikant, R., & Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *ACM SIGMOD Record, 25*(2), 1-12
6. Natei, K. N., Viradiya, J., & Sasikumar, S. (2018). Extraction text from image document and displaying its related information. *K.N.Natei Jouranl of Engineering Study and Publications,* 27-33.
7. Zhao, Q., & Bhowmick, S. S. (2003). Association rule mining. *Technical Report, CAIS, Nanyang Technological University, Singapore*.
8. Association Rule Mining I. (n.d.). *Principles of Data Mining,* 187-201.
9. Malik, U. (n.d.). Association Rule Mining via Apriori Algorithm in Python. Retrieved October 10, 2020, from https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/
10. D, K. (2018, March 15). Mining Frequent itemsets - Apriori Algorithm. Retrieved November 09, 2020, from https://dwgeek.com/mining-frequent-itemsets-apriori-algorithm.html/
11. Khare, P., & Gupta, H. (2013). Result evaluation of transaction and occurrences based on density and zonal minimum support. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*.
12. Guo, Y., & Wang, Z. (2010). A vertical format algorithm for mining frequent item sets. *2010 2nd International Conference on Advanced Computer Control.*
13. Mahmood, S., Shahbaz, M., & Guergachi, A. (2014). Negative and Positive Association Rules Mining from Text Using Frequent and Infrequent Itemsets. *The Scientific World Journal, 2014*, 1-11.
14. Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record, 29*(2), 1-12.
15. Solanki, S. K., & Patel, J. T. (2015). A Survey on Association Rule Mining. *2015 Fifth International Conference on Advanced Computing & Communication Technologies*.
16. Korczak, J., & Skrzypczak, P. (2012). FP-Growth in Discovery of Customer Patterns. *Lecture Notes in Business Information Processing Data-Driven Process Discovery and Analysis,* 120-133
17. Best Data Science & Machine Learning Platform. (2020, November 09). Retrieved November 14, 2020, from https://rapidminer.com/
18. Korczak, J., & Skrzypczak, P. (2012). FP-Growth in discovery of customer patterns. *Lecture Notes in Business Information Processing,*
19. Ceravolo, P., Guetl, C., & Rinderle-Ma, S. (2018). *Data-Driven Process Discovery and Analysis 6th IFIP WG 2.6 International Symposium, SIMPDA 2016, Graz, Austria, December 15-16, 2016, Revised Selected Papers*. Cham: Springer International Publishing.
20. Korczak J., Skrzypczak P. (2012) FP-Growth in Discovery of Customer Patterns. In: Aberer K., Damiani E., Dillon T. (eds) Data-Driven Process Discovery and Analysis. *SIMPDA 2011. Lecture Notes in Business Information Processing, vol 116. Springer, Berlin, Heidelberg*
21. Rahman A., Ezeife C.I., Aggarwal A.K. (2010) WiFi Miner: An Online Apriori-Infrequent Based Wireless Intrusion System. In: Gaber M.M., Vatsavai R.R., Omitaomu O.A., Gama J., Chawla N.V., Ganguly A.R. (eds) Knowledge Discovery from Sensor Data. Sensor-KDD 2008. *Lecture Notes in Computer Science, vol 5840. Springer, Berlin, Heidelberg.*
22. Touya, K., & Fakir, M. (2014). Mining Students' Learning Behavior in Moodle System. *Journal of Information Technology Research (JITR), 7*(4), 12-26.
23. Apriori Algorithm in Data Mining: Implementation With Examples. (2020, September 13). Retrieved November 14, 2020, from https://www.softwaretestinghelp.com/apriori-algorithm/
24. Manimaran, J., & Velmurugan, T. (2013). A survey of association rule mining in text applications. *2013 IEEE International Conference on Computational Intelligence and Computing Research*.
25. Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining — a general survey and comparison. *ACM SIGKDD Explorations Newsletter, 2*(1), 58-64.
26. Mishra, M., & K., S. (2017). Text Classification based on Association Rule Mining Technique. *International Journal of Computer Applications,*
27. Zhou, L., & Yau, S. (2010). Association Rule and Quantitative Association Rule Mining among Infrequent Items. *Rare Association Rule Mining and Knowledge Discovery,*
28. Kosters, W. A., Pijls, W., & Popova, V. (n.d.). Complexity Analysis of Depth First and FP-Growth Implementations of APRIORI. *Machine Learning and Data Mining in Pattern Recognition Lecture Notes in Computer Science,* 284-292.
29. Kotsiantis, S., & Kanellopolous, D. (2006). Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering,* 71-82.

30. Indira, K., & Kanmani, S. (2014). Association rule mining through adaptive parameter control in particle swarm optimization. *Computational Statistics, 30*(1), 251-277.
31. Duwairi, R., & Ammari, H. (2016). An enhanced CBAR algorithm for improving recommendation systems accuracy. *Simulation Modelling Practice and Theory, 60*, 54-68. doi:10.1016/j.simpat.2015.10.001
32. Goyal, N., & Jain, S. K. (2016). A comparative study of different frequent pattern mining algorithm for uncertain data: A survey. *2016 International Conference on Computing, Communication and Automation (ICCCA)*.
33. Sridevi, R., & Ramaraj, E. (2014). Multidimensional Quantitative Rule Generation Algorithm for Transactional Database. *International Journal of Computer Applications, 99*(2), 40-44.
34. Kaur, J., & Madan, N. (2015). Association Rule Mining: A Survey. *International Journal of Hybrid Information Technology, 8*(7), 239-242.
35. Second international conference on computer networks and communication technologies. (2020).
36. Best Data Science & Machine Learning Platform. (2020, November 09). Retrieved November 14, 2020, from https://rapidminer.com/
37. Neysiani, B. S., Soltani, N., Mofidi, R., & Nadimi-Shahraki, M. H. (2019). Improve Performance of Association Rule-Based Collaborative Filtering Recommendation Systems using Genetic Algorithm. *International Journal of Information Technology and Computer Science, 11*(2), 48-55.
38. Shahbazi, N., Soltani, R., Gryz, J., & An, A. (2016). Building FP-Tree on the Fly: Single-Pass Frequent Itemset Mining. *Machine Learning and Data Mining in Pattern Recognition Lecture Notes in Computer Science,* 387-400.
39. Goyal, N., & Jain, S. K. (2016). A comparative study of different frequent pattern mining algorithm for uncertain data: A survey. *2016 International Conference on Computing, Communication and Automation (ICCCA)*.
40. Zeng, Y., Yin, S., Liu, J., & Zhang, M. (2015). Research of Improved FP-Growth Algorithm in Association Rules Mining. *Scientific Programming, 2015*
41. Zeng, H., & Wang, B. (2010). An improved algorithm of FP - tree growth based on mapping. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*.
42. (n.d.). Retrieved November 01, 2020, from http://athena.ecs.csus.edu/~associationcw/FpGrowth.html
43. Guo-rong, L., & Xi-zheng, Z. (2006). Collaborative filtering based recommendation system for product bundling. *2006 International Conference on Management Science and Engineering*.
44. Mandrai, P., & Barskar, R. (2013). A novel algorithm for optimization of association rule WITH karnagh map and genetic algorithm. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*.