**DATA MINING PROJECT – 1**

SUBMITTED BY : KAUSTUBH GANER

Data Set : [UCI Adult Census Data Set](https://archive.ics.uci.edu/ml/datasets/adult)

Data Preprocessing :

* The data had various columns like age,final weight,education number, etc whose values were numeric, so I removed them as there were chances of same values being present in different columns which might have confused the algorithm and given us some irrelevant outputs. For example 40 can be someone’s age as well as number of hours someone worked in a week.
* So the data I used only includes columns that have categorical values. The columns I included were all categorical like workclass,education,martial-status,etc.

Algorithms used :

1. Apriori (BFS)

2. Apriori with modification(random sampling) (BFS)

3. FP-growth (DFS)

Apriori :

* The Apriori algorithm is based on Breadth First Search approach. It generates frequent patterns by making the itemsets using pairing such as single item itemset, 2 item itemset and so on.
* It scans the database in each of its iteration, so it gets more time consuming as size of dataset increases, hence it is not used when the size of data set is very large.

Apriori Modification :

* The modification I did to Apriori algorithm was that of random sampling.In the modification, using the pandas.sample() function, I randomly selected one-third rows of the dataset, i.e approximately 11000 rows/transactions.And as a result the time for running the algorithm reduced by a third and there was not difference in the frequent itemset generated by the original model and modified model.
* From the modification I realized that after a certain amount of observations, the increase in dataset size or increase in number of samples does not make much sense as the patterns are repetative. I have given the comparison of all the three models with respect to time required to run and number of frequent itemsets given as output below.

FP growth :

* The FP Growth algorithm is based upon Depth First Search Approach. It generates a conditional tree for every item in the data and it only requires only one scan of the database in the beginning and due to this it consumes very less time even for large datasets.

Results:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Apriori – without modification | Apriori – with modification | FP Growth |
| Number of Frequent itemsets detected | 17.92 seconds | 5.95 seconds | 0.25 seconds |
| Time Required for generating frequent itemsets | 141 | 138 - 141 | 141 |

Text

Description automatically generated

Fig – 1 : Time required to run Apriori Algorithms

Text

Description automatically generated

Fig – 2 : Time required to run FP Growth Algorithm

The number of frequent itemsets generated by the Apriori algorithm with modification is varying as we are randomly selecting data, but even though we are selecting 1/3rd data points, hardly 2 – 3 frequent itemsets are getting missed.

Resources :

1. <https://analyticsindiamag.com/apriori-vs-fp-growth-in-market-basket-analysis-a-comparative-guide/>
2. <https://www.youtube.com/watch?v=UP4ezNZfcH0&t=448s>
3. <https://towardsdatascience.com/fp-growth-frequent-pattern-generation-in-data-mining-with-python-implementation-244e561ab1c3>
4. <https://github.com/chonyy/apriori_python>
5. <https://github.com/chonyy/fpgrowth_py>
6. <https://www.youtube.com/watch?v=VB8KWm8MXss>
7. <https://github.com/asaini/Apriori>
8. <https://archive.ics.uci.edu/ml/datasets/adult>
9. <https://towardsdatascience.com/apriori-association-rule-mining-explanation-and-python-implementation-290b42afdfc6>
10. <https://github.com/kefuzhu/University-of-Rochester/blob/master/DSC440/Programming-Project/Apriori_KefuZhu.py>
11. <https://github.com/kefuzhu/University-of-Rochester/blob/master/DSC440/Programming-Project/FP-growth_KefuZhu.py>
12. Data Mining: Concepts and Techniques, 3/E