NAME: Suraj Bhardwaj

NJIT UCID: srb34

Email Address: [srb34@njit.edu](mailto:srb34@njit.edu)

Github Link: <https://github.com/xxsurajbxx/CS634MidtermProject>

Github Username: xxsurajbxx

10/7/2024

Professor: Yasser Abduallah

CS 634 101 Data Mining

**Midterm Project Report**

*Implementation and Code Usage*

**Apriori Algorithm Implementation In Retail Data Mining**

**Abstract:**

In this research project, I use the Apriori, FP Growth, and brute force algorithms, basic data mining techniques, to investigate correlations in retail transactions. I evaluate the algorithm's efficacy and efficiency by putting it into practice and using a variety of data mining methods, methodologies, and principles. In addition, I create and design unique tools to construct a model that draws insightful conclusions from the transaction data.

**Introduction:**

A potent technique for locating hidden correlations and patterns in big datasets is data mining. The Algorithms studied in this research, well-known association rule mining methods, are the focus of this research; specifically how they can be applied in a retail setting. We examine the fundamental ideas and principles of data mining that underpin our methodology, such as the creation and presentation of association rules.

In this implementation, I created association rules and identified frequently occurring itemsets using a bespoke retail dataset and various association rule mining methods. The following were the primary steps in this process:

* Fabricating data which will soon be mined for association rules and saving the information in CSV files.
* The datasets are loaded from CSV files.
* Preserving item uniqueness and order by preprocessing the collection.
* Obtaining user feedback to determine confidence and minimal support thresholds.
* Candidate item sets are iteratively generated and often updated using the brute force, Apriori, and FP Growth algorithms.

**Core Concepts and Principles:**

**Frequent Itemset Discovery:**

The brute force algorithm, Apriori, and FP Growth algorithms are algorithms to find frequent itemsets from a transactional dataset. These itemsets represent items that are frequently purchased and have an association with each other. They help to recognize patterns in consumer preferences and behaviors.

**Support and Confidence:**

Two key metrics in data mining are support and confidence. Support measures how frequently an item or itemset occurs, while confidence assesses the likelihood of items being purchased together. These metrics guide our analysis.

**Association Rules:**

By determining strong association rules, I identify which items are commonly purchased together. These rules are instrumental for optimizing sales strategies, such as recommendations.

**Project Workflow:**

This project follows a structured methodology with several

**Data Loading and Preprocessing:**

Initially the transactional data is loaded in from a user selected retail store dataset. Every transaction consists of the items that one customer has bought in a transaction.

**Determination of Minimum Support and Confidence:**

The user is prompted to give input regarding the minimum support and confidence values to only produce significant patterns.

**Iteration Through all the Possible Itemsets:**

The brute force algorithm entails generating a set of candidate k-itemsets which is the set of all possible k length combinations of items. K iteratively gets larger, starting at 1 until none of the k length itemsets are frequent.

**Support Count Calculation:**

We determine the support of each potential itemset by calculating the number of transactions that contain the itemset. While some itemsets are discarded, those that fulfill the minimum support criterion are kept.

**Confidence Calculation:**

We assess the degree of association between itemsets by calculating the confidence of association rules. Carefully comparing the support values for individual products and itemsets is necessary in this stage.

**Association Rule Generation:**

Association rules that meet the standards for minimal confidence and minimum support are extracted. These guidelines provide insightful information about which products are frequently bought together.

**Results and Evaluation:**

The brute force method is evaluated for its effectiveness and efficiency compared to the Apriori and FP Growth algorithm implemented in the mlxtend python library. The runtime of the brute force algorithm is compared to these two algorithms, measured through a python library called timeit. It should be noted that in timing the algorithms, the Apriori and FP growth algorithms use a function from mlxtend to generate the association rules from the input frequent itemsets while the brute force algorithm uses a different function to do this. Their runtimes are quite similar so it does not affect the results very much however should still be noted.

**Conclusion:**

In summary, our effort shows how data mining ideas, tenets, and techniques can be used. Using retail transaction data, we successfully applied the Apriori Algorithm to derive significant association rules. The customization of algorithms, iterative "brute force" methodology, and strict adherence to user-specified criteria demonstrate the effectiveness of data mining in identifying significant patterns that support retail industry decision-making.

*Screenshots*

The first part of my program generates the csv files.

Here is the code that generates the csv files.

A screenshot of a computer screen

Description automatically generated

Here is what the CSV files look like

A screenshot of a computer

Description automatically generated

In Part 2 I define some helper functions to make the brute force code more readable

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

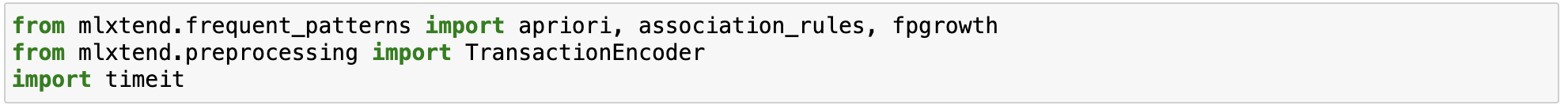
Description automatically generated

I also define a class for the association rules

A screen shot of a computer code

Description automatically generated

In part 3 I import some additional classes



I also gather user input

A screenshot of a computer code

Description automatically generated

The data is then read in



A wrapper function is used to time the functions runtimes

A screenshot of a computer code

Description automatically generated

The brute force algorithm is ran and timed

A screenshot of a computer

Description automatically generated

The data is then reformatted for the Apriori and FP Growth algorithms



The Apriori algorithm is ran and timed, and the output is formatted clearly

A screenshot of a computer

Description automatically generated

The FP growth algorithm is ran and timed, and the output is formatted clearly

A screenshot of a computer program

Description automatically generated

Here is what the program looks like when ran in the terminal.

A black screen with white text

Description automatically generated

Here is the output from the brute force algorithm.

A screen shot of a computer

Description automatically generated

Here is the output from the Apriori algorithm

A screen shot of a computer

Description automatically generated

Here is the output from the FP-Growth algorithm

A computer screen with white text

Description automatically generated

***Other***

The source code (.py file) and data sets (.csv files) will be attached to the zip file. *Link to Git Repository* https://github.com/mahumabid/Apriori\_Algorithm