

Joshua Kobuskie
Professor Yasser Abdullaah
CS-634-101 Data Mining
6 October 2024

Midterm Project Report - Apriori Algorithm

Abstract:

Association rule mining plays an important role in discovering novel, non-trivial, and potentially useful patterns that are utilized by computational systems today, including those responsible for market basket analysis, product colocation, and fraud detection. With significant real-world applications, developing a thorough understanding of association rules, the frequent itemsets used to create them, and frequent itemset generation techniques provides valuable experience in data mining. Throughout this project, three iterations of frequent pattern data mining techniques are explored. First, I develop and investigate a brute force implementation of the Apriori algorithm, examining both its procedure and effectiveness. This fundamental data mining technique is used to determine frequent itemsets, and to inform association rule generation. To confirm the results of my brute force implementation, the mlxtend python library is then called with an optimized Apriori algorithm implementation and an association rule generation function. Finally, the FP-Growth algorithm is utilized to demonstrate how advanced data structures can reduce the execution time of frequent itemset generation to inform association rule creation. The runtime of each of these implementations is compared with sample datasets containing at least 20 transactions. The development and comparison of these different data mining techniques helps to establish a deeper understanding of frequent itemsets and association rules, as well as the efficiency of these data mining techniques at scale.

Introduction:

Data mining is a valuable process in which knowledge and patterns can be extracted from large datasets, revealing non-trivial, implicit, previously unknown, and potentially useful information. Within data mining, many different techniques exist to analyze various types of data. A popular form of data mining for understanding frequent itemsets and generating association rules is the Apriori algorithm. The Apriori algorithm is used to analyze a database of transactions and determine the frequent itemsets based upon a specified support and confidence level. The support level represents how frequently an itemset appears in a database, and the confidence level represents the likelihood of a given association rule generated from frequent itemsets occurring in a transaction. The concepts explored within the Apriori algorithm and association rules are applied in many industries, but are especially prevalent in retail shopping. Association rules can be used to perform market basket analysis and recommend products frequently bought together or co-locate products in a physical store to promote increased purchasing and generate higher profits. Throughout this project, the Apriori algorithm, association rules, and the FP-Growth algorithm will be evaluated to demonstrate their real-world implementation and impacts.

The purpose of the Apriori algorithm is to identify frequent itemsets that can be used to generate association rules. In order to identify frequent itemsets within the data, the database of transactions must be traversed. By traversing through the transactions, it is possible to identify the frequency of a given itemset based on its number of occurrences within the database. With the count of occurrences of an itemset in the database, a support level can be calculated as the total number of occurrences divided by the total number of transactions. An example of support calculation is shown below:

$$\text{Frequency} = \text{Occurrences of item A in transactions}$$

$$\text{Total} = \text{Number of total transactions}$$

$$\text{Support} = \text{Frequency}/\text{Total}$$

If this value is greater than or equal to the support level provided by the user, the itemset is considered to be frequent and is stored for later use. Itemsets are generated as combinations of the original set of unique items, starting with itemsets of size 1 and increasing to itemsets of at most the total number of unique items. The number of items in each itemset can be represented as a K-itemset, where K is the number of items. If no frequent itemsets are discovered for any itemsets of size K, then no supersets of the K-itemset will have any frequent itemsets and the algorithm can terminate its search. This concept is referred to as the Apriori principle. Once all frequent itemsets have been discovered, association rules can be generated from subset combinations within each itemset with 2 or more items, also called the antecedent. The confidence of each rule can be calculated by dividing the support of the itemset by the support of the antecedent. If the confidence is greater than or equal to the confidence level specified by the user, the consequent can be determined, which represents the frequent itemset without the antecedent. Finally, the antecedent, consequent, and confidence for an association rule can be stored and output to the user. An example of a frequent itemset and association rule is shown below:

$$\text{Itemset} = \{\text{antecedent}, \text{consequent}\}$$

$$\text{Confidence} = \text{Support}(\text{antecedent} \& \text{consequent}) / \text{Support}(\text{antecedent})$$

$$\text{Association Rule} = [\text{antecedent}] \rightarrow [\text{consequent}] \text{ Confidence: XX\%}$$

The Apriori algorithm is a classic data mining technique that utilizes a brute force approach to identify frequent itemsets and generate association rules. The FP-Growth algorithm builds on the concepts of the Apriori algorithm and utilizes an advanced tree-like structure to generate frequent itemsets. This approach makes a significant improvement on the overall time complexity of the frequent itemset identification process.

In this project, I have implemented a brute force approach to the Apriori algorithm and generated 7 transactional datasets that can be used to test and compare the output of this algorithm within a

simulated retail environment. This implementation generates both the frequent itemsets discovered, as well as the association rules, based on the user specified dataset, support, and confidence level. Additionally, a comparison of my brute force implementation of the Apriori algorithm, the output of the mlxtend python library Apriori algorithm, and the output of the mlxtend python library FP-Growth algorithm are provided to ensure accuracy and compare the time to execute each of these different implementations.

Key steps in my implementation include:

- Creating CSV files that contain the transactions needed for this project
- Receiving user input for the selection of the dataset, support, and confidence
- Loading and preprocessing the data to extract a set of unique items
- Iteratively generating possible frequent itemsets and checking for their frequency within the transactions using the brute force method and user specified support levels
- Iteratively generating possible association rules and checking for likelihood based on the user specified confidence levels

Core Concepts and Principles:

Having an understanding of the core concepts and principles implemented within this project allows for a deeper understanding of the algorithms investigated.

Frequent Itemset Discovery:

Frequent itemsets are sets of unique items that satisfy the minimum required support. The entirety of the itemset must occur within a transaction to be considered in the count of occurrences to determine support. Itemsets can consist of K elements, where K represents the number of unique items within each itemset. Each K-itemset will have at most C_K^N combinations that form candidate itemsets for a given number of unique items N and a given itemset size K. Frequent itemsets provide insight into customer purchasing habits that can provide business value.

Support and Confidence:

Support and confidence play a crucial role in determining frequent itemsets and association rules, and are user defined values. Support represents the frequency of an itemset in relation to all transactions. Confidence represents the likelihood of items being purchased together.

Apriori Principle:

The Apriori Principle states that any superset of a non-frequent itemset must also be non-frequent, and any subset of a frequent itemset must also be a frequent itemset. This principle is valuable in limiting the candidate set generation of the Apriori algorithm, and

can be used to stop evaluating larger K-itemsets if no frequent itemsets have been discovered for the current size K-itemset.

Association Rules:

Association rules represent a claim that a customer is likely to purchase other specified items given that they are purchasing an item. For example, a person who buys Item A also buys Item B. Association rules help to determine items frequently bought together and can be effectively used to develop product placement or advertising strategies.

Project Workflow

Dependencies:

In order for the program to run successfully, the following libraries must be installed prior to running the program:

- csv
- itertools
- time
- mlxtend
- mlxtend.frequent_patterns
- mlxtend.preprocessing
- pandas

The csv library, itertools library, and time library come standard with most python installations and do not require additional configuration.

To install the mlxtend and pandas libraries, run the following commands from the command line:

```
python3 -m pip install mlxtend  
python3 -m pip install pandas
```

To start the program, run the following command from the command line:

```
python3 kobuskie_joshua_midtermproject.py
```

Data Initialization:

Prior to beginning the program, transactional data was initialized for each store. A dataset was created for each of the retail stores represented and this data is saved into CSV files for future use. The "amazonTransactions.csv", "bestBuyTransactions.csv", "kMartTransactions.csv", "nikeTransactions.csv", "genericTransactions.csv", "customTransactions.csv", and "walmartTransactions.csv" files must be present in the current directory in order for the program to run properly. Datasets 1 through 6 have been

created based on the examples provided. Dataset 7 has been built using GenAI, and contains 1000 transactions with 30 unique items to simulate Walmart transactional data.

Input Selection and Validation:

Upon start, the user is welcomed to the Apriori Algorithm on the command line and prompted to enter the following user specified variables.

Determination of Dataset:

The user is provided with a list of datasets and is prompted to enter an integer value between 1 and 7 on the command line to select the corresponding dataset. If a value outside of this range or a non-integer value is entered, an error is created and the program terminates.

Determination of Support:

The user is prompted to enter an integer value between 1 and 100 on the command line to select the minimum support. If a value outside of this range or a non-integer value is entered, an error is created and the program terminates.

Determination of Confidence:

The user is prompted to enter an integer value between 1 and 100 on the command line to select the minimum confidence. If a value outside of this range or a non-integer value is entered, an error is created and the program terminates.

Data Loading and Preprocessing:

The support and confidence levels entered by the user are converted into a float representing the percentage of support and confidence required. The selected dataset is loaded from the corresponding CSV file, and each transaction is read into an array for future traversal. The set of unique items is generated from the array of transactions by traversing all transactions and adding each unique item to the set. A frequent itemsets dictionary is also initialized to store the frequent itemsets and the support for each itemset.

Iterate over Candidate Itemsets of Size K:

Using a brute force method, candidate itemsets of size K will be created, starting with K=1 and increasing by 1 during each iteration. Candidate itemset generation is repeated only if at least 1 frequent itemset was found for the previous value of K and the candidate itemset size does not exceed the number of unique items.

Computer Combinations:

Using the itertools library, the candidate set of K-itemsets is created by calculating the combinations of the unique itemset of the specified size K. Each combination is then searched for within the transactional array, and its frequency is calculated.

Check Support and Store Frequency:

If the frequency divided by the total number of transactions for a given candidate itemset is greater than or equal to the support value specified by the user, the itemset is considered to be frequent and the itemset and support level are stored in the frequent itemsets dictionary as a key-value pair. If a frequent itemset is found, the looping parameter will also be updated to ensure exploration of the next set of candidate itemsets of size K+1.

Update Candidate Set Size to K+1:

After checking all candidate itemsets of size K, the candidate set size will be incremented by 1 from K to K+1. The program begins the iteration process again, performing the same evaluation to determine if any frequent itemsets were found during the prior iteration. If no frequent itemsets were found, the program will terminate and not explore any larger K-itemsets based on the Apriori Principle.

Print Frequent Itemsets:

For each itemset identified as frequent and stored in the frequent itemsets dictionary, the frequent itemset and support value for that itemset are printed to the user on the command line.

Iterate over Frequent Itemsets to Find Association Rules:

An association rules array is initialized to store association rules as discovered. The stored frequent itemsets previously discovered are now iterated over for all itemsets with at least 2 unique items.

Generate Increasing Large Combinations as an Antecedent for each Itemset:

Each frequent itemset is composed of unique items. All subsets of the current frequent itemset being evaluated will be generated, starting with subsets of size 1 and increasing until one less than the size of the frequent itemset. Each subset generated will represent the antecedent in the association rules to be tested. For each antecedent, which must be a frequent itemset previously explored due to the Apriori Principle, the support of the itemset can be divided by the support of the antecedent by finding their associated values in the frequent itemsets dictionary. The resulting value is the confidence of the association rule.

Check Confidence and Store Association Rules:

If the confidence of the association rule is greater than or equal to the user specified confidence level, the consequent can be calculated by removing the antecedent from the itemset, and the association rule is stored as the antecedent, consequent, confidence, and support in a tuple.

Print Association Rules:

For each association rule identified and stored in the association rules array, the antecedent, consequent, confidence, and support is printed to the user on the command line.

Results and Evaluation:

The accuracy of my brute force implementation is verified by comparing the results of two library-based implementations of the Apriori and FP-growth algorithm to determine frequent itemsets and association rules. The time taken to execute each of these methods is then compared to illustrate the differences in execution time for each implementation.

Existing Libraries used to Validate:

The mlxtend library was used to validate the results of my brute force implementation. The same dataset, support, and confidence as previously specified by the user are used to inform the Apriori and FP-Growth algorithms and recalculate the frequent itemsets. This information is then passed to the mlxtend association rule function to determine the association rules for both the Apriori and FP-Growth algorithms. The same print output is generated by iterating through the output of each algorithm.

Difference in Data Representation:

A notable difference in the implementation of my brute force algorithm and the mlxtend implementations is the use of a pandas dataframe. My implementation of the brute force algorithm relies on a 2D array to represent the transactions, where the mlxtend algorithms take in a pandas dataframe. To create this dataframe, the transactions were first encoded and transformed to ensure equal dimensions in each transaction in the dataframe. This difference does not impact the data itself, but does change how it is represented and stored. The use of a pandas dataframe may have an influence on the timing of these algorithms as compared to the use of a 2D array.

Timing:

To time the execution of each algorithm, the time library was used. The start time is recorded when preprocessing has been completed and the algorithm begins identifying frequent itemsets and association rules. The stop time is then determined for both the frequent itemset identification and the association rule identification at the end of printing

the frequent itemsets and at the end of printing the association rules, respectively. The time to determine frequent itemsets and the time to determine association rules is then calculated as the difference between the stop and start times, and this is calculated for each of the three algorithms and printed to the user on the command line.

Conclusion:

The resulting frequent itemsets and association rules generated by my brute force algorithm, the Apriori algorithm, and the FP-Growth algorithm were identical. This proves the accuracy of my implementation and its ability to effectively determine both frequent patterns and association rules. All three algorithms produced the same results for both the frequent itemsets and association rules generated. The timing for each of the algorithms varies based on the support, confidence, and size of the dataset selected. To limit variability and maximize differences in execution time during testing, each of the algorithms and datasets were evaluated with a 1% support and 1% confidence level. To further highlight the difference in execution time for each implementation, a large dataset of simulated Walmart data was created containing 1000 transactions and 30 unique items. This dataset is large enough to identify notable differences in the execution time of each algorithm. The execution times for each algorithm and dataset are captured in the table below:

	Brute Force		Apriori		FP-Growth	
	Frequent Itemsets	Association Rules	Frequent Itemsets	Association Rules	Frequent Itemsets	Association Rules
1. Amazon	0.0078539848330	0.0084412097930	0.0136640071900	0.0439291000400	0.0094308853150	0.0478100776700
2. Best Buy	0.0308790206900	1.0211620330000	0.1142792702000	5.8131058220000	0.1127932072000	5.8106958870000
3. K-Mart	0.0102679729500	0.0546538829800	0.0289101600600	0.3200948238000	0.0235400199900	0.3176569939000
4. Nike	0.0309779644000	1.0232639310000	0.1172528267000	5.8782582280000	0.1094920635000	5.8669419290000
5. Generic	0.0006921291351	0.0019209384920	0.0098090171810	0.0212318897200	0.0051259994510	0.0214879512800
6. Custom	0.0007150173187	0.0011184215550	0.0085160732270	0.0125629901900	0.0039908885960	0.0120627880100
7. Walmart	383.95836400000	0.0247290134400	0.1625268459000	0.0350232124300	0.1095581055000	0.0721299648300

This data may represent some unforeseen bias, as the state of the computer could impact performance, especially for small execution times of under a hundredth of a second. Based on the runtimes of each algorithm with each dataset and consistent support and confidence levels, we can observe trends in the execution time of each algorithm. With smaller datasets, my brute force implementation tends to perform better at identifying the frequent itemsets than both the Apriori and FP-Growth algorithms. This can be seen in every dataset with 20 or less transactions,

including datasets 1 through 6. This is likely caused by the overhead of creating more complex data structures that reduce the overall time complexity, but cause a performance limitation with small datasets. Once the frequent itemsets had been identified, the association rules were able to be generated in roughly the same time for both the Apriori and FP-Growth algorithms, but these times were always slightly slower than that of my brute force implementation. This could be due to the way that the library being used calculates the association rules, or from the data structure that the frequent items are returned in.

When testing with the large dataset, the difference in efficiency of each algorithm becomes much more clear. Dataset 7, the Walmart dataset with 1000 transactions and 30 unique items, had a runtime of over 6:22 minutes to generate the frequent itemsets with my brute force implementation. This can be compared to the 0.1625 seconds required by the Apriori algorithm, and the 0.1096 seconds required by the FP-Growth algorithm. This drastic difference in execution time is caused by the brute force algorithm's inability to limit the candidate itemset space and traversing the entire dataset for each candidate itemset. By limiting the candidate itemset space through optimization, the Apriori algorithm can significantly reduce the number of itemsets to be explored and therefore reduce the number of traversals through the dataset. The FP-Growth algorithm is able to further limit its traversal of the dataset by using a tree-like structure, providing further enhanced performance and reducing the runtime of frequent itemset generation by another 32.5% in this transactional dataset. This difference in performance will continue to grow as the number of transactions grows from 1000 to millions or billions of transactions. Therefore, the FP-Growth algorithm can be seen to be the fastest of the three algorithms as the size of the dataset increases.

Github Project:

The code created for this project can be accessed using the following link:

<https://github.com/joshuakobuskie/FA24-CS634-101>

This repository belongs to my GitHub account joshuakobuskie, which is associated with both my NJIT and personal emails. My emails are jsk47@njit.edu and joshkobuskie@gmail.com respectively.

Tutorial:

Open the command line and navigate to the directory in which the file is saved. In this example, the file is named kobuskie_joshua_midtermproject.py.

You should see something similar to the following:

```
[joshuakobuskie@JSK64 ~ % cd Desktop  
[joshuakobuskie@JSK64 Desktop % cd CS634  
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject  
joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % █
```

Run the following command:

```
python3 kobuskie_joshua_midtermproject.py
```

```
[joshuakobuskie@JSK64 ~ % cd Desktop
[joshuakobuskie@JSK64 Desktop % cd CS634
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py]
```

You should see the following output:

```
[joshuakobuskie@JSK64 ~ % cd Desktop
[joshuakobuskie@JSK64 Desktop % cd CS634
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
```

Enter an integer number between 1 and 7 to select the dataset for use:

```
[joshuakobuskie@JSK64 ~ % cd Desktop
[joshuakobuskie@JSK64 Desktop % cd CS634
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
1
```

Enter an integer number between 1 and 100 to select the support level:

```
[joshuakobuskie@JSK64 ~ % cd Desktop
[joshuakobuskie@JSK64 Desktop % cd CS634
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
1
You have selected dataset 1: Amazon
Please select the minimum support level in % (value 1 to 100):
50
```

Enter an integer number between 1 and 100 to select the confidence level:

```
[joshuakobuskie@JSK64 ~ % cd Desktop
[joshuakobuskie@JSK64 Desktop % cd CS634
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
1
You have selected dataset 1: Amazon
Please select the minimum support level in % (value 1 to 100):
50
You have selected a minimum support level of 50%
Please select the minimum confidence level in % (value 1 to 100):
70
```

Observe the output generated by the program and compare the frequent itemsets, association rules, and execution time for each of the three algorithms. The itemsets and association rules generated with each algorithm should be identical for a given input:

```
[joshuakobuskie@JSK64 ~ % cd Desktop
[joshuakobuskie@JSK64 Desktop % cd CS634
[joshuakobuskie@JSK64 CS634 % cd kobuskie_joshua_midtermproject
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
1
You have selected dataset 1: Amazon
Please select the minimum support level in % (value 1 to 100):
50
You have selected a minimum support level of 50%
Please select the minimum confidence level in % (value 1 to 100):
70
You have selected a minimum confidence level of 70%

Brute Force Frequent Itemsets

Frequent Itemset 1: [Android Programming: The Big Nerd Ranch]
Support: 65.00%

Frequent Itemset 2: [A Beginner's Guide]
Support: 55.00%

Frequent Itemset 3: [Java For Dummies]
Support: 65.00%

Frequent Itemset 4: [Java: The Complete Reference]
Support: 50.00%

Frequent Itemset 5: [Java For Dummies, Java: The Complete Reference]
Support: 50.00%

Generated Brute Force Frequent Itemsets in 0.0017919540405273438 seconds

Brute Force Association Rules

Association Rule 1: [Java For Dummies] -> [Java: The Complete Reference]
Confidence: 76.92%
Support: 50.00%

Association Rule 2: [Java: The Complete Reference] -> [Java For Dummies]
Confidence: 100.00%
Support: 50.00%

Generated Brute Force Association Rules in 5.125999450683594e-05 seconds
#####
```

Apriori Frequent Itemsets

```
Frequent Itemset 1: [A Beginner's Guide]
Support: 55.00%

Frequent Itemset 2: [Android Programming: The Big Nerd Ranch]
Support: 65.00%

Frequent Itemset 3: [Java For Dummies]
Support: 65.00%

Frequent Itemset 4: [Java: The Complete Reference]
Support: 50.00%

Frequent Itemset 5: [Java For Dummies, Java: The Complete Reference]
Support: 50.00%
```

Generated Apriori Frequent Itemsets in 0.0051000118255615234 seconds

Apriori Association Rules

```
Association Rule 1: [Java For Dummies] -> [Java: The Complete Reference]
Confidence: 76.92%
Support: 50.00%

Association Rule 2: [Java: The Complete Reference] -> [Java For Dummies]
Confidence: 100.00%
Support: 50.00%
```

Generated Apriori Association Rules in 0.0033600330352783203 seconds
#####

FP-Growth Frequent Itemsets

```
Frequent Itemset 1: [Java For Dummies]
Support: 65.00%

Frequent Itemset 2: [Android Programming: The Big Nerd Ranch]
Support: 65.00%

Frequent Itemset 3: [A Beginner's Guide]
Support: 55.00%

Frequent Itemset 4: [Java: The Complete Reference]
Support: 50.00%

Frequent Itemset 5: [Java For Dummies, Java: The Complete Reference]
Support: 50.00%
```

Generated FP-Growth Frequent Itemsets in 0.0020563602447509766 seconds

FP-Growth Association Rules

```
Association Rule 1: [Java For Dummies] -> [Java: The Complete Reference]
Confidence: 76.92%
Support: 50.00%

Association Rule 2: [Java: The Complete Reference] -> [Java For Dummies]
Confidence: 100.00%
Support: 50.00%
```

Generated FP-Growth Association Rules in 0.0027379989624023438 seconds
#####

Screenshots

An example execution has been captured for each of the 7 datasets with unique support and confidence levels specified by the user.

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
1
You have selected dataset 1: Amazon
Please select the minimum support level in % (value 1 to 100):
50
You have selected a minimum support level of 50%
Please select the minimum confidence level in % (value 1 to 100):
70
You have selected a minimum confidence level of 70%

Brute Force Frequent Itemsets

Frequent Itemset 1: [Java For Dummies]
Support: 65.00%

Frequent Itemset 2: [A Beginner's Guide]
Support: 55.00%

Frequent Itemset 3: [Java: The Complete Reference]
Support: 50.00%

Frequent Itemset 4: [Android Programming: The Big Nerd Ranch]
Support: 65.00%

Frequent Itemset 5: [Java For Dummies, Java: The Complete Reference]
Support: 50.00%

Generated Brute Force Frequent Itemsets in 0.001978158950805664 seconds

Brute Force Association Rules

Association Rule 1: [Java For Dummies] -> [Java: The Complete Reference]
Confidence: 76.92%
Support: 50.00%

Association Rule 2: [Java: The Complete Reference] -> [Java For Dummies]
Confidence: 100.00%
Support: 50.00%

Generated Brute Force Association Rules in 5.030632019042969e-05 seconds
#####
```

Apriori Frequent Itemsets

```
Frequent Itemset 1: [A Beginner's Guide]
Support: 55.00%

Frequent Itemset 2: [Android Programming: The Big Nerd Ranch]
Support: 65.00%

Frequent Itemset 3: [Java For Dummies]
Support: 65.00%

Frequent Itemset 4: [Java: The Complete Reference]
Support: 50.00%

Frequent Itemset 5: [Java For Dummies, Java: The Complete Reference]
Support: 50.00%
```

Generated Apriori Frequent Itemsets in 0.005342960357666016 seconds

Apriori Association Rules

```
Association Rule 1: [Java For Dummies] -> [Java: The Complete Reference]
Confidence: 76.92%
Support: 50.00%

Association Rule 2: [Java: The Complete Reference] -> [Java For Dummies]
Confidence: 100.00%
Support: 50.00%
```

Generated Apriori Association Rules in 0.0033888816833496094 seconds

#####

FP-Growth Frequent Itemsets

```
Frequent Itemset 1: [Java For Dummies]
Support: 65.00%

Frequent Itemset 2: [Android Programming: The Big Nerd Ranch]
Support: 65.00%

Frequent Itemset 3: [A Beginner's Guide]
Support: 55.00%

Frequent Itemset 4: [Java: The Complete Reference]
Support: 50.00%

Frequent Itemset 5: [Java For Dummies, Java: The Complete Reference]
Support: 50.00%
```

Generated FP-Growth Frequent Itemsets in 0.00203704833984375 seconds

FP-Growth Association Rules

```
Association Rule 1: [Java For Dummies] -> [Java: The Complete Reference]
Confidence: 76.92%
Support: 50.00%

Association Rule 2: [Java: The Complete Reference] -> [Java For Dummies]
Confidence: 100.00%
Support: 50.00%
```

Generated FP-Growth Association Rules in 0.002702951431274414 seconds

#####

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
2
You have selected dataset 2: Best Buy
Please select the minimum support level in % (value 1 to 100):
60
You have selected a minimum support level of 60%
Please select the minimum confidence level in % (value 1 to 100):
80
You have selected a minimum confidence level of 80%

Brute Force Frequent Itemsets

Frequent Itemset 1: [Anti-Virus]
Support: 70.00%

Frequent Itemset 2: [Lab Top]
Support: 60.00%

Frequent Itemset 3: [Flash Drive]
Support: 65.00%

Frequent Itemset 4: [Lab Top Case]
Support: 70.00%

Frequent Itemset 5: [Anti-Virus, Lab Top Case]
Support: 60.00%

Generated Brute Force Frequent Itemsets in 0.003381013870239258 seconds

Brute Force Association Rules

Association Rule 1: [Anti-Virus] -> [Lab Top Case]
Confidence: 85.71%
Support: 60.00%

Association Rule 2: [Lab Top Case] -> [Anti-Virus]
Confidence: 85.71%
Support: 60.00%

Generated Brute Force Association Rules in 5.316734313964844e-05 seconds
#####
```

Apriori Frequent Itemsets

Frequent Itemset 1: [Anti-Virus]
Support: 70.00%

Frequent Itemset 2: [Flash Drive]
Support: 65.00%

Frequent Itemset 3: [Lab Top]
Support: 60.00%

Frequent Itemset 4: [Lab Top Case]
Support: 70.00%

Frequent Itemset 5: [Anti-Virus, Lab Top Case]
Support: 60.00%

Generated Apriori Frequent Itemsets in 0.005012035369873047 seconds

Apriori Association Rules

Association Rule 1: [Anti-Virus] → [Lab Top Case]
Confidence: 85.71%
Support: 60.00%

Association Rule 2: [Lab Top Case] → [Anti-Virus]
Confidence: 85.71%
Support: 60.00%

Generated Apriori Association Rules in 0.0033490657806396484 seconds
#####

FP-Growth Frequent Itemsets

Frequent Itemset 1: [Anti-Virus]
Support: 70.00%

Frequent Itemset 2: [Flash Drive]
Support: 65.00%

Frequent Itemset 3: [Lab Top Case]
Support: 70.00%

Frequent Itemset 4: [Lab Top]
Support: 60.00%

Frequent Itemset 5: [Anti-Virus, Lab Top Case]
Support: 60.00%

Generated FP-Growth Frequent Itemsets in 0.0020627975463867188 seconds

FP-Growth Association Rules

Association Rule 1: [Anti-Virus] → [Lab Top Case]
Confidence: 85.71%
Support: 60.00%

Association Rule 2: [Lab Top Case] → [Anti-Virus]
Confidence: 85.71%
Support: 60.00%

Generated FP-Growth Association Rules in 0.0026950836181640625 seconds
#####

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
3
You have selected dataset 3: K-Mart
Please select the minimum support level in % (value 1 to 100):
45
You have selected a minimum support level of 45%
Please select the minimum confidence level in % (value 1 to 100):
65
You have selected a minimum confidence level of 65%

Brute Force Frequent Itemsets

Frequent Itemset 1: [Kids Bedding]
Support: 60.00%

Frequent Itemset 2: [Sheets]
Support: 50.00%

Frequent Itemset 3: [Decorative Pillows]
Support: 50.00%

Frequent Itemset 4: [Bed Skirts]
Support: 55.00%

Frequent Itemset 5: [Shams]
Support: 55.00%

Frequent Itemset 6: [Kids Bedding, Sheets]
Support: 50.00%

Frequent Itemset 7: [Kids Bedding, Bed Skirts]
Support: 50.00%

Frequent Itemset 8: [Kids Bedding, Shams]
Support: 45.00%

Frequent Itemset 9: [Sheets, Bed Skirts]
Support: 45.00%

Frequent Itemset 10: [Bed Skirts, Shams]
Support: 45.00%

Frequent Itemset 11: [Kids Bedding, Sheets, Bed Skirts]
Support: 45.00%

Generated Brute Force Frequent Itemsets in 0.00400996208190918 seconds
```

Brute Force Association Rules

Association Rule 1: [Kids Bedding] -> [Sheets]
Confidence: 83.33%
Support: 50.00%

Association Rule 2: [Sheets] -> [Kids Bedding]
Confidence: 100.00%
Support: 50.00%

Association Rule 3: [Kids Bedding] -> [Bed Skirts]
Confidence: 83.33%
Support: 50.00%

Association Rule 4: [Bed Skirts] -> [Kids Bedding]
Confidence: 90.91%
Support: 50.00%

Association Rule 5: [Kids Bedding] -> [Shams]
Confidence: 75.00%
Support: 45.00%

Association Rule 6: [Shams] -> [Kids Bedding]
Confidence: 81.82%
Support: 45.00%

Association Rule 7: [Sheets] -> [Bed Skirts]
Confidence: 90.00%
Support: 45.00%

Association Rule 8: [Bed Skirts] -> [Sheets]
Confidence: 81.82%
Support: 45.00%

Association Rule 9: [Bed Skirts] -> [Shams]
Confidence: 81.82%
Support: 45.00%

Association Rule 10: [Shams] -> [Bed Skirts]
Confidence: 81.82%
Support: 45.00%

Association Rule 11: [Kids Bedding] -> [Sheets, Bed Skirts]
Confidence: 75.00%
Support: 45.00%

Association Rule 12: [Sheets] -> [Bed Skirts, Kids Bedding]
Confidence: 90.00%
Support: 45.00%

Association Rule 13: [Bed Skirts] -> [Sheets, Kids Bedding]
Confidence: 81.82%
Support: 45.00%

Association Rule 14: [Kids Bedding, Sheets] -> [Bed Skirts]
Confidence: 90.00%
Support: 45.00%

Association Rule 15: [Kids Bedding, Bed Skirts] -> [Sheets]
Confidence: 90.00%
Support: 45.00%

Association Rule 16: [Sheets, Bed Skirts] -> [Kids Bedding]
Confidence: 100.00%
Support: 45.00%

Generated Brute Force Association Rules in 0.0002002716064453125 seconds
#####

Apriori Frequent Itemsets

Frequent Itemset 1: [Bed Skirts]
Support: 55.00%

Frequent Itemset 2: [Decorative Pillows]
Support: 50.00%

Frequent Itemset 3: [Kids Bedding]
Support: 60.00%

Frequent Itemset 4: [Shams]
Support: 55.00%

Frequent Itemset 5: [Sheets]
Support: 50.00%

Frequent Itemset 6: [Bed Skirts, Kids Bedding]
Support: 50.00%

Frequent Itemset 7: [Bed Skirts, Shams]
Support: 45.00%

Frequent Itemset 8: [Bed Skirts, Sheets]
Support: 45.00%

Frequent Itemset 9: [Kids Bedding, Shams]
Support: 45.00%

Frequent Itemset 10: [Sheets, Kids Bedding]
Support: 50.00%

Frequent Itemset 11: [Bed Skirts, Kids Bedding, Sheets]
Support: 45.00%

Generated Apriori Frequent Itemsets in 0.006353855133056641 seconds

Apriori Association Rules

```
Association Rule 1: [Bed Skirts] -> [Kids Bedding]
Confidence: 90.91%
Support: 50.00%

Association Rule 2: [Kids Bedding] -> [Bed Skirts]
Confidence: 83.33%
Support: 50.00%

Association Rule 3: [Bed Skirts] -> [Shams]
Confidence: 81.82%
Support: 45.00%

Association Rule 4: [Shams] -> [Bed Skirts]
Confidence: 81.82%
Support: 45.00%

Association Rule 5: [Bed Skirts] -> [Sheets]
Confidence: 81.82%
Support: 45.00%

Association Rule 6: [Sheets] -> [Bed Skirts]
Confidence: 90.00%
Support: 45.00%

Association Rule 7: [Kids Bedding] -> [Shams]
Confidence: 75.00%
Support: 45.00%

Association Rule 8: [Shams] -> [Kids Bedding]
Confidence: 81.82%
Support: 45.00%

Association Rule 9: [Sheets] -> [Kids Bedding]
Confidence: 100.00%
Support: 50.00%

Association Rule 10: [Kids Bedding] -> [Sheets]
Confidence: 83.33%
Support: 50.00%

Association Rule 11: [Bed Skirts, Kids Bedding] -> [Sheets]
Confidence: 90.00%
Support: 45.00%

Association Rule 12: [Bed Skirts, Sheets] -> [Kids Bedding]
Confidence: 100.00%
Support: 45.00%

Association Rule 13: [Sheets, Kids Bedding] -> [Bed Skirts]
Confidence: 90.00%
Support: 45.00%

Association Rule 14: [Bed Skirts] -> [Sheets, Kids Bedding]
Confidence: 81.82%
Support: 45.00%

Association Rule 15: [Kids Bedding] -> [Bed Skirts, Sheets]
Confidence: 75.00%
Support: 45.00%

Association Rule 16: [Sheets] -> [Bed Skirts, Kids Bedding]
Confidence: 90.00%
Support: 45.00%
```

Generated Apriori Association Rules in 0.004534006118774414 seconds
#####

FP-Growth Frequent Itemsets

Frequent Itemset 1: [Decorative Pillows]

Support: 50.00%

Frequent Itemset 2: [Kids Bedding]

Support: 60.00%

Frequent Itemset 3: [Shams]

Support: 55.00%

Frequent Itemset 4: [Bed Skirts]

Support: 55.00%

Frequent Itemset 5: [Sheets]

Support: 50.00%

Frequent Itemset 6: [Kids Bedding, Shams]

Support: 45.00%

Frequent Itemset 7: [Bed Skirts, Kids Bedding]

Support: 50.00%

Frequent Itemset 8: [Bed Skirts, Shams]

Support: 45.00%

Frequent Itemset 9: [Sheets, Kids Bedding]

Support: 50.00%

Frequent Itemset 10: [Sheets, Bed Skirts]

Support: 45.00%

Frequent Itemset 11: [Sheets, Kids Bedding, Bed Skirts]

Support: 45.00%

Generated FP-Growth Frequent Itemsets in 0.0026149749755859375 seconds

FP-Growth Association Rules

Association Rule 1: [Kids Bedding] -> [Shams]
Confidence: 75.00%
Support: 45.00%

Association Rule 2: [Shams] -> [Kids Bedding]
Confidence: 81.82%
Support: 45.00%

Association Rule 3: [Bed Skirts] -> [Kids Bedding]
Confidence: 90.91%
Support: 50.00%

Association Rule 4: [Kids Bedding] -> [Bed Skirts]
Confidence: 83.33%
Support: 50.00%

Association Rule 5: [Bed Skirts] -> [Shams]
Confidence: 81.82%
Support: 45.00%

Association Rule 6: [Shams] -> [Bed Skirts]
Confidence: 81.82%
Support: 45.00%

Association Rule 7: [Sheets] -> [Kids Bedding]
Confidence: 100.00%
Support: 50.00%

Association Rule 8: [Kids Bedding] -> [Sheets]
Confidence: 83.33%
Support: 50.00%

Association Rule 9: [Sheets] -> [Bed Skirts]
Confidence: 90.00%
Support: 45.00%

Association Rule 10: [Bed Skirts] -> [Sheets]
Confidence: 81.82%
Support: 45.00%

Association Rule 11: [Sheets, Kids Bedding] -> [Bed Skirts]
Confidence: 90.00%
Support: 45.00%

Association Rule 12: [Sheets, Bed Skirts] -> [Kids Bedding]
Confidence: 100.00%
Support: 45.00%

Association Rule 13: [Bed Skirts, Kids Bedding] -> [Sheets]
Confidence: 90.00%
Support: 45.00%

Association Rule 14: [Sheets] -> [Bed Skirts, Kids Bedding]
Confidence: 90.00%
Support: 45.00%

Association Rule 15: [Kids Bedding] -> [Sheets, Bed Skirts]
Confidence: 75.00%
Support: 45.00%

Association Rule 16: [Bed Skirts] -> [Sheets, Kids Bedding]
Confidence: 81.82%
Support: 45.00%

Generated FP-Growth Association Rules in 0.004056215286254883 seconds
#####

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
4
You have selected dataset 4: Nike
Please select the minimum support level in % (value 1 to 100):
50
You have selected a minimum support level of 50%
Please select the minimum confidence level in % (value 1 to 100):
90
You have selected a minimum confidence level of 90%

Brute Force Frequent Itemsets

Frequent Itemset 1: [Dry Fit V-Neck]
Support: 50.00%

Frequent Itemset 2: [Socks]
Support: 65.00%

Frequent Itemset 3: [Running Shoe]
Support: 70.00%

Frequent Itemset 4: [Swimming Shirt]
Support: 55.00%

Frequent Itemset 5: [Rash Guard]
Support: 60.00%

Frequent Itemset 6: [Sweatshirts]
Support: 65.00%

Frequent Itemset 7: [Modern Pants]
Support: 50.00%

Frequent Itemset 8: [Dry Fit V-Neck, Rash Guard]
Support: 50.00%

Frequent Itemset 9: [Socks, Running Shoe]
Support: 55.00%

Frequent Itemset 10: [Socks, Sweatshirts]
Support: 60.00%

Frequent Itemset 11: [Running Shoe, Sweatshirts]
Support: 55.00%

Frequent Itemset 12: [Swimming Shirt, Rash Guard]
Support: 50.00%

Frequent Itemset 13: [Sweatshirts, Modern Pants]
Support: 50.00%

Frequent Itemset 14: [Socks, Running Shoe, Sweatshirts]
Support: 50.00%

Generated Brute Force Frequent Itemsets in 0.006016969680786133 seconds
```

Brute Force Association Rules

Association Rule 1: [Dry Fit V-Neck] → [Rash Guard]

Confidence: 100.00%

Support: 50.00%

Association Rule 2: [Socks] → [Sweatshirts]

Confidence: 92.31%

Support: 60.00%

Association Rule 3: [Sweatshirts] → [Socks]

Confidence: 92.31%

Support: 60.00%

Association Rule 4: [Swimming Shirt] → [Rash Guard]

Confidence: 90.91%

Support: 50.00%

Association Rule 5: [Modern Pants] → [Sweatshirts]

Confidence: 100.00%

Support: 50.00%

Association Rule 6: [Socks, Running Shoe] → [Sweatshirts]

Confidence: 90.91%

Support: 50.00%

Association Rule 7: [Running Shoe, Sweatshirts] → [Socks]

Confidence: 90.91%

Support: 50.00%

Generated Brute Force Association Rules in 0.00011205673217773438 seconds

#####

Apriori Frequent Itemsets

Frequent Itemset 1: [Dry Fit V-Neck]
Support: 50.00%

Frequent Itemset 2: [Modern Pants]
Support: 50.00%

Frequent Itemset 3: [Rash Guard]
Support: 60.00%

Frequent Itemset 4: [Running Shoe]
Support: 70.00%

Frequent Itemset 5: [Socks]
Support: 65.00%

Frequent Itemset 6: [Sweatshirts]
Support: 65.00%

Frequent Itemset 7: [Swimming Shirt]
Support: 55.00%

Frequent Itemset 8: [Rash Guard, Dry Fit V-Neck]
Support: 50.00%

Frequent Itemset 9: [Sweatshirts, Modern Pants]
Support: 50.00%

Frequent Itemset 10: [Swimming Shirt, Rash Guard]
Support: 50.00%

Frequent Itemset 11: [Socks, Running Shoe]
Support: 55.00%

Frequent Itemset 12: [Running Shoe, Sweatshirts]
Support: 55.00%

Frequent Itemset 13: [Socks, Sweatshirts]
Support: 60.00%

Frequent Itemset 14: [Socks, Running Shoe, Sweatshirts]
Support: 50.00%

Generated Apriori Frequent Itemsets in 0.0069429874420166016 seconds

Apriori Association Rules

Association Rule 1: [Dry Fit V-Neck] -> [Rash Guard]

Confidence: 100.00%

Support: 50.00%

Association Rule 2: [Modern Pants] -> [Sweatshirts]

Confidence: 100.00%

Support: 50.00%

Association Rule 3: [Swimming Shirt] -> [Rash Guard]

Confidence: 90.91%

Support: 50.00%

Association Rule 4: [Socks] -> [Sweatshirts]

Confidence: 92.31%

Support: 60.00%

Association Rule 5: [Sweatshirts] -> [Socks]

Confidence: 92.31%

Support: 60.00%

Association Rule 6: [Socks, Running Shoe] -> [Sweatshirts]

Confidence: 90.91%

Support: 50.00%

Association Rule 7: [Running Shoe, Sweatshirts] -> [Socks]

Confidence: 90.91%

Support: 50.00%

Generated Apriori Association Rules in 0.0037622451782226562 seconds

#####

FP-Growth Frequent Itemsets

Frequent Itemset 1: [Running Shoe]
Support: 70.00%

Frequent Itemset 2: [Sweatshirts]
Support: 65.00%

Frequent Itemset 3: [Socks]
Support: 65.00%

Frequent Itemset 4: [Modern Pants]
Support: 50.00%

Frequent Itemset 5: [Rash Guard]
Support: 60.00%

Frequent Itemset 6: [Swimming Shirt]
Support: 55.00%

Frequent Itemset 7: [Dry Fit V-Neck]
Support: 50.00%

Frequent Itemset 8: [Running Shoe, Sweatshirts]
Support: 55.00%

Frequent Itemset 9: [Socks, Sweatshirts]
Support: 60.00%

Frequent Itemset 10: [Socks, Running Shoe]
Support: 55.00%

Frequent Itemset 11: [Socks, Running Shoe, Sweatshirts]
Support: 50.00%

Frequent Itemset 12: [Sweatshirts, Modern Pants]
Support: 50.00%

Frequent Itemset 13: [Swimming Shirt, Rash Guard]
Support: 50.00%

Frequent Itemset 14: [Rash Guard, Dry Fit V-Neck]
Support: 50.00%

Generated FP-Growth Frequent Itemsets in 0.003030061721801758 seconds

FP-Growth Association Rules

Association Rule 1: [Socks] → [Sweatshirts]

Confidence: 92.31%

Support: 60.00%

Association Rule 2: [Sweatshirts] → [Socks]

Confidence: 92.31%

Support: 60.00%

Association Rule 3: [Socks, Running Shoe] → [Sweatshirts]

Confidence: 90.91%

Support: 50.00%

Association Rule 4: [Running Shoe, Sweatshirts] → [Socks]

Confidence: 90.91%

Support: 50.00%

Association Rule 5: [Modern Pants] → [Sweatshirts]

Confidence: 100.00%

Support: 50.00%

Association Rule 6: [Swimming Shirt] → [Rash Guard]

Confidence: 90.91%

Support: 50.00%

Association Rule 7: [Dry Fit V-Neck] → [Rash Guard]

Confidence: 100.00%

Support: 50.00%

Generated FP-Growth Association Rules in 0.0034780502319335938 seconds

#####

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
5
You have selected dataset 5: Generic
Please select the minimum support level in % (value 1 to 100):
40
You have selected a minimum support level of 40%
Please select the minimum confidence level in % (value 1 to 100):
100
You have selected a minimum confidence level of 100%

Brute Force Frequent Itemsets

Frequent Itemset 1: [C]
Support: 63.64%

Frequent Itemset 2: [A]
Support: 100.00%

Frequent Itemset 3: [E]
Support: 72.73%

Frequent Itemset 4: [B]
Support: 45.45%

Frequent Itemset 5: [C, A]
Support: 63.64%

Frequent Itemset 6: [A, E]
Support: 72.73%

Frequent Itemset 7: [A, B]
Support: 45.45%

Generated Brute Force Frequent Itemsets in 0.0003910064697265625 seconds

Brute Force Association Rules

Association Rule 1: [C] -> [A]
Confidence: 100.00%
Support: 63.64%

Association Rule 2: [E] -> [A]
Confidence: 100.00%
Support: 72.73%

Association Rule 3: [B] -> [A]
Confidence: 100.00%
Support: 45.45%

Generated Brute Force Association Rules in 7.891654968261719e-05 seconds
#####
```

Apriori Frequent Itemsets

Frequent Itemset 1: [A]

Support: 100.00%

Frequent Itemset 2: [B]

Support: 45.45%

Frequent Itemset 3: [C]

Support: 63.64%

Frequent Itemset 4: [E]

Support: 72.73%

Frequent Itemset 5: [A, B]

Support: 45.45%

Frequent Itemset 6: [A, C]

Support: 63.64%

Frequent Itemset 7: [A, E]

Support: 72.73%

Generated Apriori Frequent Itemsets in 0.005223989486694336 seconds

Apriori Association Rules

Association Rule 1: [B] → [A]

Confidence: 100.00%

Support: 45.45%

Association Rule 2: [C] → [A]

Confidence: 100.00%

Support: 63.64%

Association Rule 3: [E] → [A]

Confidence: 100.00%

Support: 72.73%

Generated Apriori Association Rules in 0.0033919811248779297 seconds

#####

FP-Growth Frequent Itemsets

Frequent Itemset 1: [A]

Support: 100.00%

Frequent Itemset 2: [C]

Support: 63.64%

Frequent Itemset 3: [B]

Support: 45.45%

Frequent Itemset 4: [E]

Support: 72.73%

Frequent Itemset 5: [A, C]

Support: 63.64%

Frequent Itemset 6: [A, B]

Support: 45.45%

Frequent Itemset 7: [A, E]

Support: 72.73%

Generated FP-Growth Frequent Itemsets in 0.0021567344665527344 seconds

FP-Growth Association Rules

Association Rule 1: [C] → [A]

Confidence: 100.00%

Support: 63.64%

Association Rule 2: [B] → [A]

Confidence: 100.00%

Support: 45.45%

Association Rule 3: [E] → [A]

Confidence: 100.00%

Support: 72.73%

Generated FP-Growth Association Rules in 0.0027861595153808594 seconds

#####

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
6
You have selected dataset 6: Custom
Please select the minimum support level in % (value 1 to 100):
66
You have selected a minimum support level of 66%
Please select the minimum confidence level in % (value 1 to 100):
66
You have selected a minimum confidence level of 66%

Brute Force Frequent Itemsets

Frequent Itemset 1: [juice]
Support: 66.67%

Frequent Itemset 2: [milk]
Support: 66.67%

Frequent Itemset 3: [pen]
Support: 66.67%

Frequent Itemset 4: [cheese]
Support: 83.33%

Frequent Itemset 5: [juice, milk]
Support: 66.67%

Frequent Itemset 6: [pen, cheese]
Support: 66.67%

Generated Brute Force Frequent Itemsets in 0.0003829002380371094 seconds

Brute Force Association Rules

Association Rule 1: [juice] -> [milk]
Confidence: 100.00%
Support: 66.67%

Association Rule 2: [milk] -> [juice]
Confidence: 100.00%
Support: 66.67%

Association Rule 3: [pen] -> [cheese]
Confidence: 100.00%
Support: 66.67%

Association Rule 4: [cheese] -> [pen]
Confidence: 80.00%
Support: 66.67%

Generated Brute Force Association Rules in 8.7738037109375e-05 seconds
#####
```

Apriori Frequent Itemsets

Frequent Itemset 1: [cheese]
Support: 83.33%

Frequent Itemset 2: [juice]
Support: 66.67%

Frequent Itemset 3: [milk]
Support: 66.67%

Frequent Itemset 4: [pen]
Support: 66.67%

Frequent Itemset 5: [pen, cheese]
Support: 66.67%

Frequent Itemset 6: [juice, milk]
Support: 66.67%

Generated Apriori Frequent Itemsets in 0.005101203918457031 seconds

Apriori Association Rules

Association Rule 1: [pen] → [cheese]
Confidence: 100.00%
Support: 66.67%

Association Rule 2: [cheese] → [pen]
Confidence: 80.00%
Support: 66.67%

Association Rule 3: [juice] → [milk]
Confidence: 100.00%
Support: 66.67%

Association Rule 4: [milk] → [juice]
Confidence: 100.00%
Support: 66.67%

Generated Apriori Association Rules in 0.003551959991455078 seconds
#####

FP-Growth Frequent Itemsets

Frequent Itemset 1: [cheese]
Support: 83.33%

Frequent Itemset 2: [pen]
Support: 66.67%

Frequent Itemset 3: [milk]
Support: 66.67%

Frequent Itemset 4: [juice]
Support: 66.67%

Frequent Itemset 5: [pen, cheese]
Support: 66.67%

Frequent Itemset 6: [juice, milk]
Support: 66.67%

Generated FP-Growth Frequent Itemsets in 0.002040863037109375 seconds

FP-Growth Association Rules

Association Rule 1: [pen] → [cheese]
Confidence: 100.00%
Support: 66.67%

Association Rule 2: [cheese] → [pen]
Confidence: 80.00%
Support: 66.67%

Association Rule 3: [juice] → [milk]
Confidence: 100.00%
Support: 66.67%

Association Rule 4: [milk] → [juice]
Confidence: 100.00%
Support: 66.67%

Generated FP-Growth Association Rules in 0.0028929710388183594 seconds
#####

```
[joshuakobuskie@JSK64 kobuskie_joshua_midtermproject % python3 kobuskie_joshua_midtermproject.py
CS634-101 Apriori Midterm Project - Joshua Kobuskie
Please select a dataset by number:
1. Amazon
2. Best Buy
3. K-Mart
4. Nike
5. Generic
6. Custom
7. Walmart
7
You have selected dataset 7: Walmart
Please select the minimum support level in % (value 1 to 100):
10
You have selected a minimum support level of 10%
Please select the minimum confidence level in % (value 1 to 100):
40
You have selected a minimum confidence level of 40%

Brute Force Frequent Itemsets

Frequent Itemset 1: [Cereal]
Support: 27.40%

Frequent Itemset 2: [Pet Food]
Support: 17.20%

Frequent Itemset 3: [Nuts]
Support: 26.00%

Frequent Itemset 4: [Frozen Foods]
Support: 19.90%

Frequent Itemset 5: [Tea]
Support: 17.90%

Frequent Itemset 6: [Baking Goods]
Support: 19.00%

Frequent Itemset 7: [Spices]
Support: 17.50%

Frequent Itemset 8: [Ice Cream]
Support: 19.20%

Frequent Itemset 9: [Bacon]
Support: 11.80%

Frequent Itemset 10: [Grains]
Support: 16.80%

Frequent Itemset 11: [Eggs]
Support: 19.30%

Frequent Itemset 12: [Rice]
Support: 26.20%

Frequent Itemset 13: [Juice]
Support: 16.00%

Frequent Itemset 14: [Milk]
Support: 19.10%

Frequent Itemset 15: [Snacks]
Support: 17.60%

Frequent Itemset 16: [Soda]
Support: 18.10%

Frequent Itemset 17: [Granola]
Support: 10.20%

Frequent Itemset 18: [Pasta]
Support: 17.10%
```

```
Frequent Itemset 19: [Yogurt]
Support: 25.30%

Frequent Itemset 20: [Bread]
Support: 18.80%

Frequent Itemset 21: [Chicken]
Support: 16.60%

Frequent Itemset 22: [Condiments]
Support: 16.20%

Frequent Itemset 23: [Baby Products]
Support: 17.30%

Frequent Itemset 24: [Fruits]
Support: 19.40%

Frequent Itemset 25: [Toiletries]
Support: 19.90%

Frequent Itemset 26: [Coffee]
Support: 16.40%

Frequent Itemset 27: [Cleaning Supplies]
Support: 17.30%

Frequent Itemset 28: [Vegetables]
Support: 24.40%

Frequent Itemset 29: [Cheese]
Support: 30.10%

Frequent Itemset 30: [Butter]
Support: 29.20%

Frequent Itemset 31: [Sauces]
Support: 24.70%

Frequent Itemset 32: [Beef]
Support: 16.20%

Frequent Itemset 33: [Cereal, Milk]
Support: 11.60%

Frequent Itemset 34: [Nuts, Ice Cream]
Support: 11.60%

Frequent Itemset 35: [Bacon, Eggs]
Support: 11.80%

Frequent Itemset 36: [Eggs, Cheese]
Support: 11.10%

Frequent Itemset 37: [Rice, Chicken]
Support: 10.40%

Frequent Itemset 38: [Granola, Fruits]
Support: 10.20%

Frequent Itemset 39: [Pasta, Cheese]
Support: 10.40%

Frequent Itemset 40: [Yogurt, Fruits]
Support: 12.50%

Frequent Itemset 41: [Bread, Butter]
Support: 13.00%

Frequent Itemset 42: [Chicken, Vegetables]
Support: 10.10%
```

Generated Brute Force Frequent Itemsets in 5.622416019439697 seconds

Brute Force Association Rules

Association Rule 1: [Cereal] -> [Milk]
Confidence: 42.34%
Support: 11.60%

Association Rule 2: [Milk] -> [Cereal]
Confidence: 60.73%
Support: 11.60%

Association Rule 3: [Nuts] -> [Ice Cream]
Confidence: 44.62%
Support: 11.60%

Association Rule 4: [Ice Cream] -> [Nuts]
Confidence: 60.42%
Support: 11.60%

Association Rule 5: [Bacon] -> [Eggs]
Confidence: 100.00%
Support: 11.80%

Association Rule 6: [Eggs] -> [Bacon]
Confidence: 61.14%
Support: 11.80%

Association Rule 7: [Eggs] -> [Cheese]
Confidence: 57.51%
Support: 11.10%

Association Rule 8: [Chicken] -> [Rice]
Confidence: 62.65%
Support: 10.40%

Association Rule 9: [Granola] -> [Fruits]
Confidence: 100.00%
Support: 10.20%

Association Rule 10: [Fruits] -> [Granola]
Confidence: 52.58%
Support: 10.20%

Association Rule 11: [Pasta] -> [Cheese]
Confidence: 60.82%
Support: 10.40%

Association Rule 12: [Yogurt] -> [Fruits]
Confidence: 49.41%
Support: 12.50%

Association Rule 13: [Fruits] -> [Yogurt]
Confidence: 64.43%
Support: 12.50%

Association Rule 14: [Bread] -> [Butter]
Confidence: 69.15%
Support: 13.00%

Association Rule 15: [Butter] -> [Bread]
Confidence: 44.52%
Support: 13.00%

Association Rule 16: [Chicken] -> [Vegetables]
Confidence: 60.84%
Support: 10.10%

Association Rule 17: [Vegetables] -> [Chicken]
Confidence: 41.39%
Support: 10.10%

Generated Brute Force Association Rules in 0.0004057884216308594 seconds
#####

Apriori Frequent Itemsets

Frequent Itemset 1: [Baby Products]
Support: 17.30%

Frequent Itemset 2: [Bacon]
Support: 11.80%

Frequent Itemset 3: [Baking Goods]
Support: 19.00%

Frequent Itemset 4: [Beef]
Support: 16.20%

Frequent Itemset 5: [Bread]
Support: 18.80%

Frequent Itemset 6: [Butter]
Support: 29.20%

Frequent Itemset 7: [Cereal]
Support: 27.40%

Frequent Itemset 8: [Cheese]
Support: 30.10%

Frequent Itemset 9: [Chicken]
Support: 16.60%

Frequent Itemset 10: [Cleaning Supplies]
Support: 17.30%

Frequent Itemset 11: [Coffee]
Support: 16.40%

Frequent Itemset 12: [Condiments]
Support: 16.20%

Frequent Itemset 13: [Eggs]
Support: 19.30%

Frequent Itemset 14: [Frozen Foods]
Support: 19.90%

Frequent Itemset 15: [Fruits]
Support: 19.40%

Frequent Itemset 16: [Grains]
Support: 16.80%

Frequent Itemset 17: [Granola]
Support: 10.20%

Frequent Itemset 18: [Ice Cream]
Support: 19.20%

Frequent Itemset 19: [Juice]
Support: 16.00%

Frequent Itemset 20: [Milk]
Support: 19.10%

Frequent Itemset 21: [Nuts]
Support: 26.00%

Frequent Itemset 22: [Pasta]
Support: 17.10%

Frequent Itemset 23: [Pet Food]
Support: 17.20%

Frequent Itemset 24: [Rice]
Support: 26.20%

Frequent Itemset 25: [Sauces]
Support: 24.70%

Frequent Itemset 26: [Snacks]
Support: 17.60%

Frequent Itemset 27: [Soda]
Support: 18.10%

Frequent Itemset 28: [Spices]
Support: 17.50%

Frequent Itemset 29: [Tea]
Support: 17.90%

Frequent Itemset 30: [Toiletries]
Support: 19.90%

Frequent Itemset 31: [Vegetables]
Support: 24.40%

Frequent Itemset 32: [Yogurt]
Support: 25.30%

Frequent Itemset 33: [Eggs, Bacon]
Support: 11.80%

Frequent Itemset 34: [Butter, Bread]
Support: 13.00%

Frequent Itemset 35: [Milk, Cereal]
Support: 11.60%

Frequent Itemset 36: [Eggs, Cheese]
Support: 11.10%

Frequent Itemset 37: [Pasta, Cheese]
Support: 10.40%

Frequent Itemset 38: [Chicken, Rice]
Support: 10.40%

Frequent Itemset 39: [Vegetables, Chicken]
Support: 10.10%

Frequent Itemset 40: [Fruits, Granola]
Support: 10.20%

Frequent Itemset 41: [Yogurt, Fruits]
Support: 12.50%

Frequent Itemset 42: [Nuts, Ice Cream]
Support: 11.60%

Generated Apriori Frequent Itemsets in 0.011141061782836914 seconds

Apriori Association Rules

Association Rule 1: [Eggs] -> [Bacon]

Confidence: 61.14%

Support: 11.80%

Association Rule 2: [Bacon] -> [Eggs]

Confidence: 100.00%

Support: 11.80%

Association Rule 3: [Butter] -> [Bread]

Confidence: 44.52%

Support: 13.00%

Association Rule 4: [Bread] -> [Butter]

Confidence: 69.15%

Support: 13.00%

Association Rule 5: [Milk] -> [Cereal]

Confidence: 60.73%

Support: 11.60%

Association Rule 6: [Cereal] -> [Milk]

Confidence: 42.34%

Support: 11.60%

Association Rule 7: [Eggs] -> [Cheese]

Confidence: 57.51%

Support: 11.10%

Association Rule 8: [Pasta] -> [Cheese]

Confidence: 60.82%

Support: 10.40%

Association Rule 9: [Chicken] -> [Rice]

Confidence: 62.65%

Support: 10.40%

Association Rule 10: [Vegetables] -> [Chicken]

Confidence: 41.39%

Support: 10.10%

Association Rule 11: [Chicken] -> [Vegetables]

Confidence: 60.84%

Support: 10.10%

Association Rule 12: [Fruits] -> [Granola]

Confidence: 52.58%

Support: 10.20%

Association Rule 13: [Granola] -> [Fruits]

Confidence: 100.00%

Support: 10.20%

Association Rule 14: [Yogurt] -> [Fruits]

Confidence: 49.41%

Support: 12.50%

Association Rule 15: [Fruits] -> [Yogurt]

Confidence: 64.43%

Support: 12.50%

Association Rule 16: [Nuts] -> [Ice Cream]

Confidence: 44.62%

Support: 11.60%

Association Rule 17: [Ice Cream] -> [Nuts]

Confidence: 60.42%

Support: 11.60%

Generated Apriori Association Rules in 0.004624128341674805 seconds
#####

FP-Growth Frequent Itemsets

Frequent Itemset 1: [Cereal]
Support: 27.40%

Frequent Itemset 2: [Rice]
Support: 26.20%

Frequent Itemset 3: [Sauces]
Support: 24.70%

Frequent Itemset 4: [Toiletries]
Support: 19.90%

Frequent Itemset 5: [Tea]
Support: 17.90%

Frequent Itemset 6: [Snacks]
Support: 17.60%

Frequent Itemset 7: [Cleaning Supplies]
Support: 17.30%

Frequent Itemset 8: [Chicken]
Support: 16.60%

Frequent Itemset 9: [Coffee]
Support: 16.40%

Frequent Itemset 10: [Baby Products]
Support: 17.30%

Frequent Itemset 11: [Nuts]
Support: 26.00%

Frequent Itemset 12: [Ice Cream]
Support: 19.20%

Frequent Itemset 13: [Beef]
Support: 16.20%

Frequent Itemset 14: [Cheese]
Support: 30.10%

Frequent Itemset 15: [Milk]
Support: 19.10%

Frequent Itemset 16: [Soda]
Support: 18.10%

Frequent Itemset 17: [Yogurt]
Support: 25.30%

Frequent Itemset 18: [Frozen Foods]
Support: 19.90%

Frequent Itemset 19: [Fruits]
Support: 19.40%

Frequent Itemset 20: [Pet Food]
Support: 17.20%

Frequent Itemset 21: [Vegetables]
Support: 24.40%

Frequent Itemset 22: [Eggs]
Support: 19.30%

Frequent Itemset 23: [Baking Goods]
Support: 19.00%

Frequent Itemset 24: [Spices]
Support: 17.50%

Frequent Itemset 25: [Bacon]
Support: 11.80%

Frequent Itemset 26: [Granola]
Support: 10.20%

Frequent Itemset 27: [Condiments]
Support: 16.20%

Frequent Itemset 28: [Butter]
Support: 29.20%

Frequent Itemset 29: [Grains]
Support: 16.80%

Frequent Itemset 30: [Pasta]
Support: 17.10%

Frequent Itemset 31: [Juice]
Support: 16.00%

Frequent Itemset 32: [Bread]
Support: 18.80%

Frequent Itemset 33: [Chicken, Rice]
Support: 10.40%

Frequent Itemset 34: [Vegetables, Chicken]
Support: 10.10%

Frequent Itemset 35: [Nuts, Ice Cream]
Support: 11.60%

Frequent Itemset 36: [Milk, Cereal]
Support: 11.60%

Frequent Itemset 37: [Yogurt, Fruits]
Support: 12.50%

Frequent Itemset 38: [Eggs, Cheese]
Support: 11.10%

Frequent Itemset 39: [Eggs, Bacon]
Support: 11.80%

Frequent Itemset 40: [Fruits, Granola]
Support: 10.20%

Frequent Itemset 41: [Pasta, Cheese]
Support: 10.40%

Frequent Itemset 42: [Butter, Bread]
Support: 13.00%

Generated FP-Growth Frequent Itemsets in 0.02637004852294922 seconds

FP-Growth Association Rules

Association Rule 1: [Chicken] → [Rice]
Confidence: 62.65%
Support: 10.40%

Association Rule 2: [Vegetables] → [Chicken]
Confidence: 41.39%
Support: 10.10%

Association Rule 3: [Chicken] → [Vegetables]
Confidence: 60.84%
Support: 10.10%

Association Rule 4: [Nuts] → [Ice Cream]
Confidence: 44.62%
Support: 11.60%

Association Rule 5: [Ice Cream] → [Nuts]
Confidence: 60.42%
Support: 11.60%

Association Rule 6: [Milk] → [Cereal]
Confidence: 60.73%
Support: 11.60%

Association Rule 7: [Cereal] → [Milk]
Confidence: 42.34%
Support: 11.60%

Association Rule 8: [Yogurt] → [Fruits]
Confidence: 49.41%
Support: 12.50%

Association Rule 9: [Fruits] → [Yogurt]
Confidence: 64.43%
Support: 12.50%

Association Rule 10: [Eggs] → [Cheese]
Confidence: 57.51%
Support: 11.10%

Association Rule 11: [Eggs] → [Bacon]
Confidence: 61.14%
Support: 11.80%

Association Rule 12: [Bacon] → [Eggs]
Confidence: 100.00%
Support: 11.80%

Association Rule 13: [Fruits] → [Granola]
Confidence: 52.58%
Support: 10.20%

Association Rule 14: [Granola] → [Fruits]
Confidence: 100.00%
Support: 10.20%

Association Rule 15: [Pasta] → [Cheese]
Confidence: 60.82%
Support: 10.40%

Association Rule 16: [Butter] → [Bread]
Confidence: 44.52%
Support: 13.00%

Association Rule 17: [Bread] → [Butter]
Confidence: 69.15%
Support: 13.00%

Generated FP-Growth Association Rules in 0.004269838333129883 seconds
#####