CS634-Data Mining

Midterm Project Report

Student Name: Rajeev Kumar Alahari

Email: ra833@njit.edu

Course: CS634-Data Mining

Instructor: Dr. Yasser Abduallah

1 Introduction

This report focuses on implementing and comparing three key methods for frequent itemset mining and association rule learning, a core concept in data mining. These methods aim to uncover hidden relationships between items in transaction datasets to support decision-making and recommendation systems.

Brute Force:

The brute-force approach was implemented from scratch to demonstrate the basic working principle of frequent itemset mining. It systematically generates all possible item combinations and checks their frequency in the dataset. Though simple and educational, it is computationally expensive and inefficient for large datasets.

Apriori (mlxtend):

The Apriori algorithm was implemented using the `mlxtend` Python library, which efficiently prunes unpromising itemsets based on the Apriori property. It uses a level-wise search, reducing the number of candidate sets and improving performance compared to brute force. However, it can still be slow with very dense or large datasets.

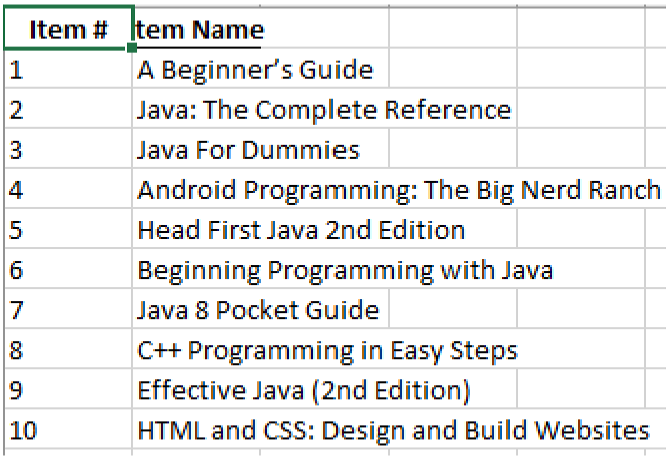
FP-Growth (mlxtend):

The FP-Growth algorithm, also implemented using `mlxtend`, overcomes Apriori’s candidate generation limitation by compressing transactions into a compact FP-Tree structure. It recursively extracts frequent itemsets from the tree, making it significantly faster and more scalable. This method is ideal for large-scale data mining applications.

2 Dataset Creation

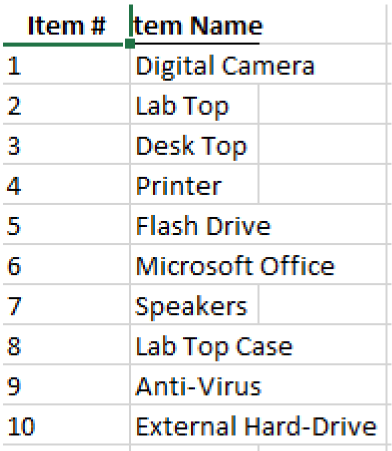
2.1 Items:

Amazon Items:



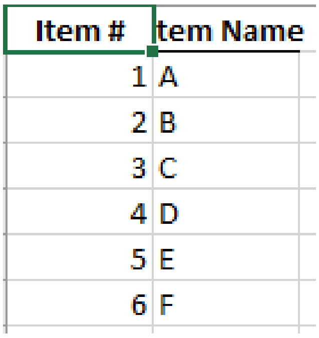
Item #	Item Name
1	A Beginner's Guide
2	Java: The Complete Reference
3	Java For Dummies
4	Android Programming: The Big Nerd Ranch
5	Head First Java 2nd Edition
6	Beginning Programming with Java
7	Java 8 Pocket Guide
8	C++ Programming in Easy Steps
9	Effective Java (2nd Edition)
10	HTML and CSS: Design and Build Websites

BestBuy items:



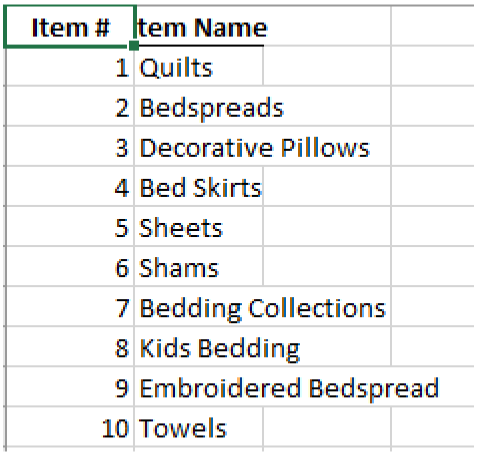
Item #	Item Name
1	Digital Camera
2	Lab Top
3	Desk Top
4	Printer
5	Flash Drive
6	Microsoft Office
7	Speakers
8	Lab Top Case
9	Anti-Virus
10	External Hard-Drive

Generic Items:



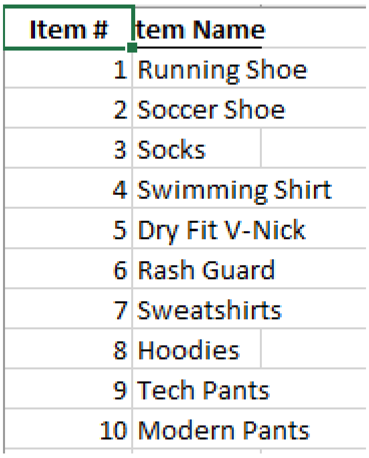
Item #	Item Name
1	A
2	B
3	C
4	D
5	E
6	F

Kmart Items:



Item #	Item Name
1	Quilts
2	Bedspreads
3	Decorative Pillows
4	Bed Skirts
5	Sheets
6	Shams
7	Bedding Collections
8	Kids Bedding
9	Embroidered Bedspread
10	Towels

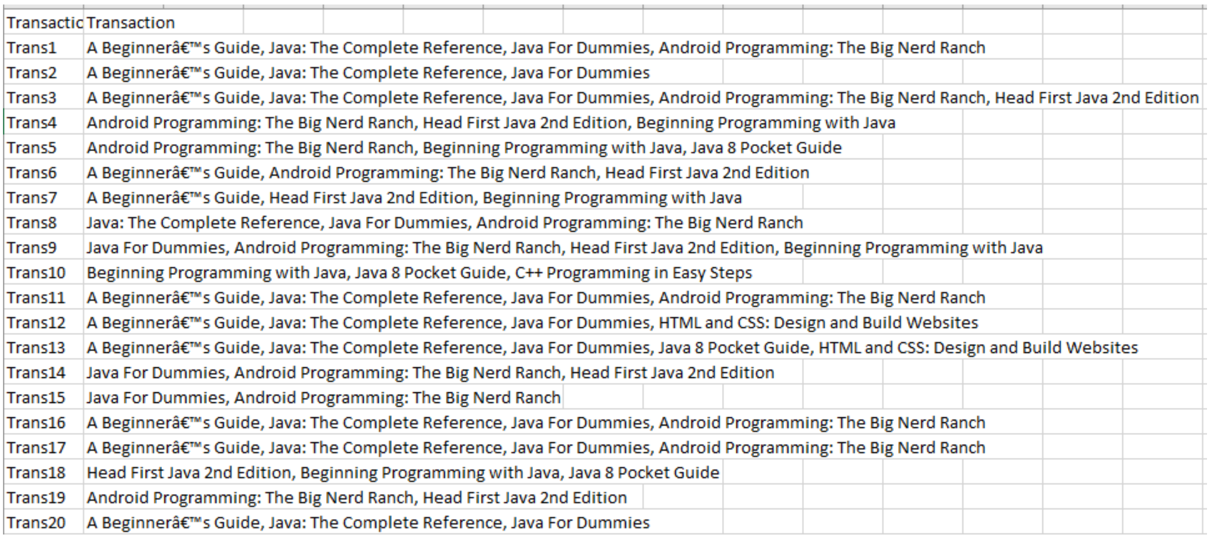
Nike Items:



Item #	Item Name
1	Running Shoe
2	Soccer Shoe
3	Socks
4	Swimming Shirt
5	Dry Fit V-Nick
6	Rash Guard
7	Sweatshirts
8	Hoodies
9	Tech Pants
10	Modern Pants

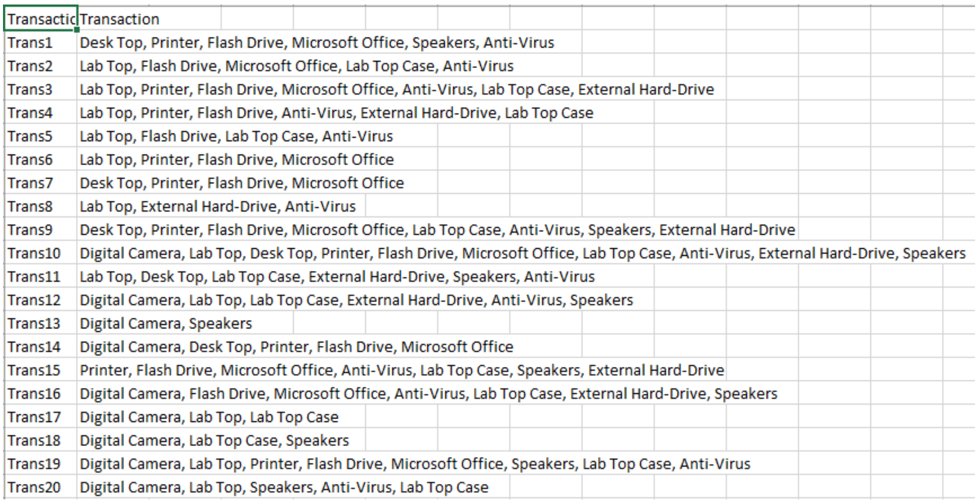
2.2 Transactions:

Amazon Transactions:



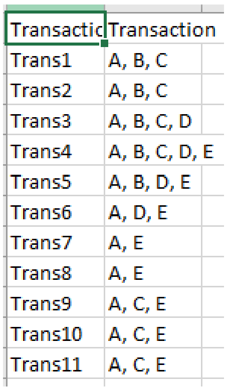
Transactic	Transaction
Trans1	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch
Trans2	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies
Trans3	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans4	Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition, Beginning Programming with Java
Trans5	Android Programming: The Big Nerd Ranch, Beginning Programming with Java, Java 8 Pocket Guide
Trans6	A Beginnerâ€™s Guide, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans7	A Beginnerâ€™s Guide, Head First Java 2nd Edition, Beginning Programming with Java
Trans8	Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch
Trans9	Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition, Beginning Programming with Java
Trans10	Beginning Programming with Java, Java 8 Pocket Guide, C++ Programming in Easy Steps
Trans11	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch
Trans12	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, HTML and CSS: Design and Build Websites
Trans13	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, Java 8 Pocket Guide, HTML and CSS: Design and Build Websites
Trans14	Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans15	Java For Dummies, Android Programming: The Big Nerd Ranch
Trans16	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch
Trans17	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch
Trans18	Head First Java 2nd Edition, Beginning Programming with Java, Java 8 Pocket Guide
Trans19	Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans20	A Beginnerâ€™s Guide, Java: The Complete Reference, Java For Dummies

BestBuy Transactions



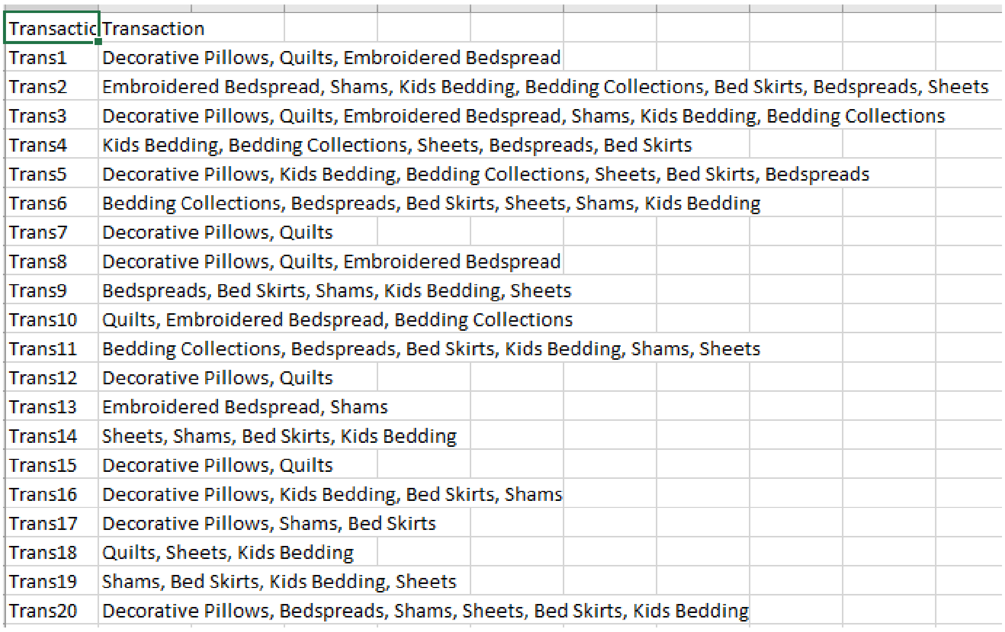
Transactic	Transaction
Trans1	Desk Top, Printer, Flash Drive, Microsoft Office, Speakers, Anti-Virus
Trans2	Lab Top, Flash Drive, Microsoft Office, Lab Top Case, Anti-Virus
Trans3	Lab Top, Printer, Flash Drive, Microsoft Office, Anti-Virus, Lab Top Case, External Hard-Drive
Trans4	Lab Top, Printer, Flash Drive, Anti-Virus, External Hard-Drive, Lab Top Case
Trans5	Lab Top, Flash Drive, Lab Top Case, Anti-Virus
Trans6	Lab Top, Printer, Flash Drive, Microsoft Office
Trans7	Desk Top, Printer, Flash Drive, Microsoft Office
Trans8	Lab Top, External Hard-Drive, Anti-Virus
Trans9	Desk Top, Printer, Flash Drive, Microsoft Office, Lab Top Case, Anti-Virus, Speakers, External Hard-Drive
Trans10	Digital Camera, Lab Top, Desk Top, Printer, Flash Drive, Microsoft Office, Lab Top Case, Anti-Virus, External Hard-Drive, Speakers
Trans11	Lab Top, Desk Top, Lab Top Case, External Hard-Drive, Speakers, Anti-Virus
Trans12	Digital Camera, Lab Top, Lab Top Case, External Hard-Drive, Anti-Virus, Speakers
Trans13	Digital Camera, Speakers
Trans14	Digital Camera, Desk Top, Printer, Flash Drive, Microsoft Office
Trans15	Printer, Flash Drive, Microsoft Office, Anti-Virus, Lab Top Case, Speakers, External Hard-Drive
Trans16	Digital Camera, Flash Drive, Microsoft Office, Anti-Virus, Lab Top Case, External Hard-Drive, Speakers
Trans17	Digital Camera, Lab Top, Lab Top Case
Trans18	Digital Camera, Lab Top Case, Speakers
Trans19	Digital Camera, Lab Top, Printer, Flash Drive, Microsoft Office, Speakers, Lab Top Case, Anti-Virus
Trans20	Digital Camera, Lab Top, Speakers, Anti-Virus, Lab Top Case

Generic Transactions



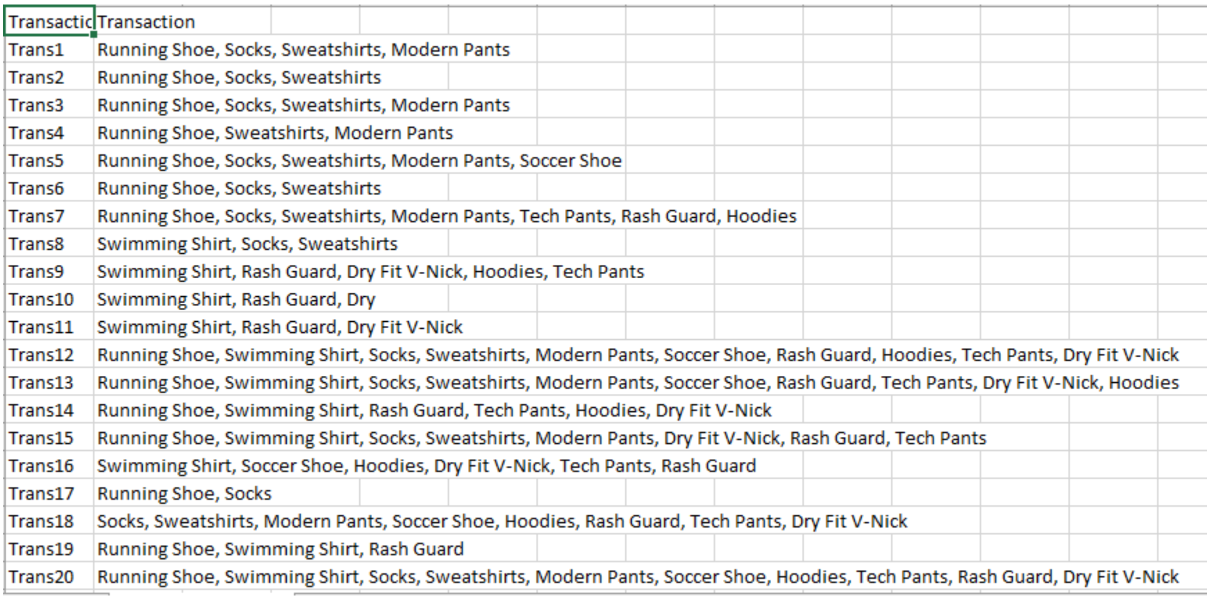
Transactic	Transaction
Trans1	A, B, C
Trans2	A, B, C
Trans3	A, B, C, D
Trans4	A, B, C, D, E
Trans5	A, B, D, E
Trans6	A, D, E
Trans7	A, E
Trans8	A, E
Trans9	A, C, E
Trans10	A, C, E
Trans11	A, C, E

Kmart Transactions



Transactic	Transaction
Trans1	Decorative Pillows, Quilts, Embroidered Bedspread
Trans2	Embroidered Bedspread, Shams, Kids Bedding, Bedding Collections, Bed Skirts, Bedspreads, Sheets
Trans3	Decorative Pillows, Quilts, Embroidered Bedspread, Shams, Kids Bedding, Bedding Collections
Trans4	Kids Bedding, Bedding Collections, Sheets, Bedspreads, Bed Skirts
Trans5	Decorative Pillows, Kids Bedding, Bedding Collections, Sheets, Bed Skirts, Bedspreads
Trans6	Bedding Collections, Bedspreads, Bed Skirts, Sheets, Shams, Kids Bedding
Trans7	Decorative Pillows, Quilts
Trans8	Decorative Pillows, Quilts, Embroidered Bedspread
Trans9	Bedspreads, Bed Skirts, Shams, Kids Bedding, Sheets
Trans10	Quilts, Embroidered Bedspread, Bedding Collections
Trans11	Bedding Collections, Bedspreads, Bed Skirts, Kids Bedding, Shams, Sheets
Trans12	Decorative Pillows, Quilts
Trans13	Embroidered Bedspread, Shams
Trans14	Sheets, Shams, Bed Skirts, Kids Bedding
Trans15	Decorative Pillows, Quilts
Trans16	Decorative Pillows, Kids Bedding, Bed Skirts, Shams
Trans17	Decorative Pillows, Shams, Bed Skirts
Trans18	Quilts, Sheets, Kids Bedding
Trans19	Shams, Bed Skirts, Kids Bedding, Sheets
Trans20	Decorative Pillows, Bedspreads, Shams, Sheets, Bed Skirts, Kids Bedding

Nike Transactions



Transactic	Transaction
Trans1	Running Shoe, Socks, Sweatshirts, Modern Pants
Trans2	Running Shoe, Socks, Sweatshirts
Trans3	Running Shoe, Socks, Sweatshirts, Modern Pants
Trans4	Running Shoe, Sweatshirts, Modern Pants
Trans5	Running Shoe, Socks, Sweatshirts, Modern Pants, Soccer Shoe
Trans6	Running Shoe, Socks, Sweatshirts
Trans7	Running Shoe, Socks, Sweatshirts, Modern Pants, Tech Pants, Rash Guard, Hoodies
Trans8	Swimming Shirt, Socks, Sweatshirts
Trans9	Swimming Shirt, Rash Guard, Dry Fit V-Nick, Hoodies, Tech Pants
Trans10	Swimming Shirt, Rash Guard, Dry
Trans11	Swimming Shirt, Rash Guard, Dry Fit V-Nick
Trans12	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Soccer Shoe, Rash Guard, Hoodies, Tech Pants, Dry Fit V-Nick
Trans13	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Soccer Shoe, Rash Guard, Tech Pants, Dry Fit V-Nick, Hoodies
Trans14	Running Shoe, Swimming Shirt, Rash Guard, Tech Pants, Hoodies, Dry Fit V-Nick
Trans15	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Dry Fit V-Nick, Rash Guard, Tech Pants
Trans16	Swimming Shirt, Soccer Shoe, Hoodies, Dry Fit V-Nick, Tech Pants, Rash Guard
Trans17	Running Shoe, Socks
Trans18	Socks, Sweatshirts, Modern Pants, Soccer Shoe, Hoodies, Rash Guard, Tech Pants, Dry Fit V-Nick
Trans19	Running Shoe, Swimming Shirt, Rash Guard
Trans20	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Soccer Shoe, Hoodies, Tech Pants, Rash Guard, Dry Fit V-Nick

2.3 Dataset notes:

I have extracted the professor's dataset and converted to csv format and saved those files.

3 Brute Force Algorithm

3.1 Method:

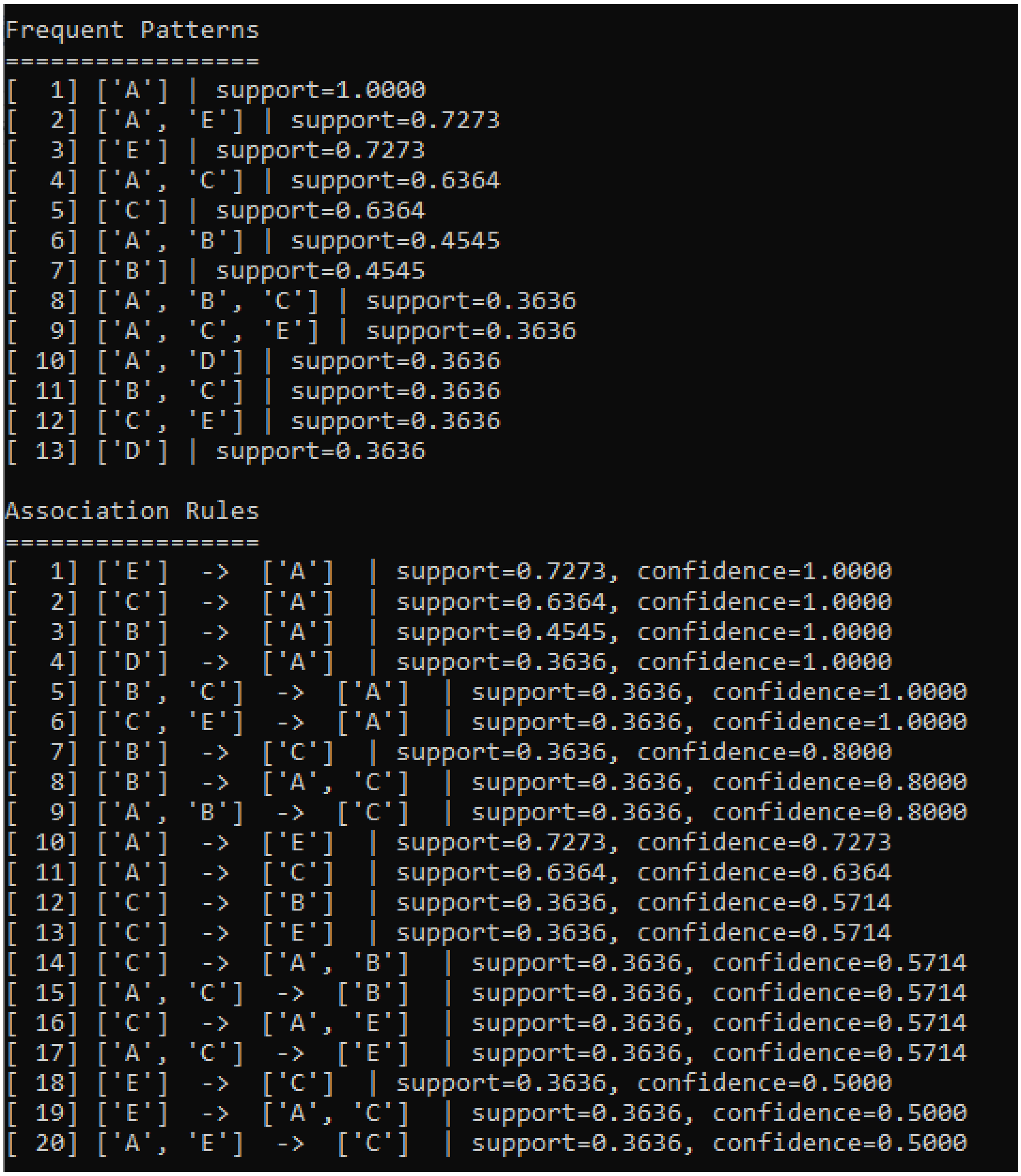
1. The brute-force method systematically generates all possible combinations of items from the dataset.
2. It then checks each combination's occurrence across all transactions to calculate its support value.
3. Only itemsets meeting the minimum support threshold are considered frequent.
4. After finding frequent itemsets, association rules are generated by computing confidence for all possible splits.
5. Though conceptually simple, this approach becomes computationally infeasible for large datasets due to exponential growth of combinations.

3.2 Run:

Dataset: Generic_Transactions.csv

Parameters: Support=0.3, Confidence=0.5

Output:



```
Frequent Patterns
=================
[  1] ['A'] | support=1.0000
[  2] ['A', 'E'] | support=0.7273
[  3] ['E'] | support=0.7273
[  4] ['A', 'C'] | support=0.6364
[  5] ['C'] | support=0.6364
[  6] ['A', 'B'] | support=0.4545
[  7] ['B'] | support=0.4545
[  8] ['A', 'B', 'C'] | support=0.3636
[  9] ['A', 'C', 'E'] | support=0.3636
[ 10] ['A', 'D'] | support=0.3636
[ 11] ['B', 'C'] | support=0.3636
[ 12] ['C', 'E'] | support=0.3636
[ 13] ['D'] | support=0.3636

Association Rules
=================
[  1] ['E']  ->  ['A']  | support=0.7273, confidence=1.0000
[  2] ['C']  ->  ['A']  | support=0.6364, confidence=1.0000
[  3] ['B']  ->  ['A']  | support=0.4545, confidence=1.0000
[  4] ['D']  ->  ['A']  | support=0.3636, confidence=1.0000
[  5] ['B', 'C']  ->  ['A']  | support=0.3636, confidence=1.0000
[  6] ['C', 'E']  ->  ['A']  | support=0.3636, confidence=1.0000
[  7] ['B']  ->  ['C']  | support=0.3636, confidence=0.8000
[  8] ['B']  ->  ['A', 'C']  | support=0.3636, confidence=0.8000
[  9] ['A', 'B']  ->  ['C']  | support=0.3636, confidence=0.8000
[ 10] ['A']  ->  ['E']  | support=0.7273, confidence=0.7273
[ 11] ['A']  ->  ['C']  | support=0.6364, confidence=0.6364
[ 12] ['C']  ->  ['B']  | support=0.3636, confidence=0.5714
[ 13] ['C']  ->  ['E']  | support=0.3636, confidence=0.5714
[ 14] ['C']  ->  ['A', 'B']  | support=0.3636, confidence=0.5714
[ 15] ['A', 'C']  ->  ['B']  | support=0.3636, confidence=0.5714
[ 16] ['C']  ->  ['A', 'E']  | support=0.3636, confidence=0.5714
[ 17] ['A', 'C']  ->  ['E']  | support=0.3636, confidence=0.5714
[ 18] ['E']  ->  ['C']  | support=0.3636, confidence=0.5000
[ 19] ['E']  ->  ['A', 'C']  | support=0.3636, confidence=0.5000
[ 20] ['A', 'E']  ->  ['C']  | support=0.3636, confidence=0.5000
```

4 Apriori and FP-Growth

4.1 Apriori Algorithm

1. Apriori improves efficiency by applying the Apriori principle — “if an itemset is frequent, all its subsets must be frequent.”
2. It begins by finding frequent 1-itemsets, then iteratively extends them to k-itemsets using only previously frequent ones.
3. At each iteration, infrequent itemsets are pruned early, significantly reducing the search space.

4. The algorithm scans the database multiple times to calculate supports and identify qualifying itemsets.
5. Finally, association rules are derived from the frequent itemsets that meet both support and confidence thresholds.

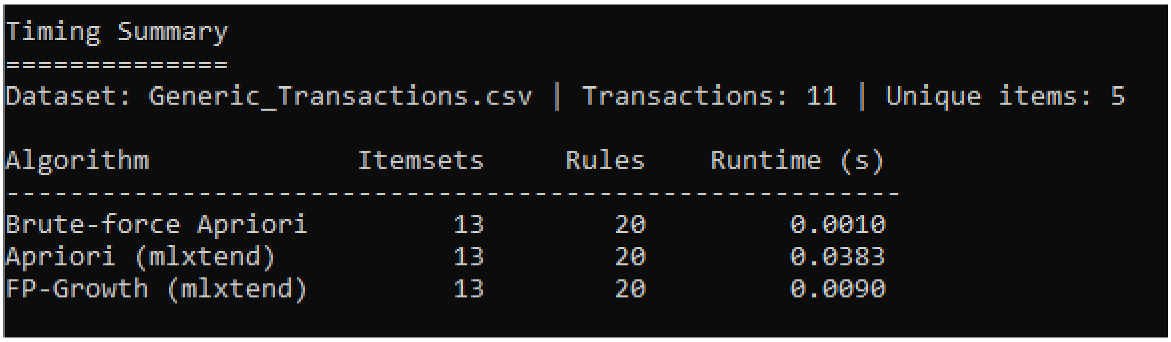
Results: Same as Brute Force.

4.2 FP-Growth Algorithm

1. FP-Growth eliminates candidate generation by using an efficient tree-based structure (FP-Tree) to store transactions.
2. It first scans the database to identify frequent items and orders them by descending frequency.
3. Each transaction is then inserted into the FP-Tree, sharing common prefixes to compress the dataset.
4. The algorithm recursively mines conditional FP-Trees to discover frequent patterns directly.
5. As it avoids multiple database scans, FP-Growth achieves much faster performance on large and dense datasets.

Results: Same as Brute Force.

4.3 Timing Comparison



```
Timing Summary
==============
Dataset: Generic_Transactions.csv | Transactions: 11 | Unique items: 5

Algorithm             Itemsets     Rules    Runtime (s)
-------------------------------------------------------
Brute-force Apriori         13        20         0.0010
Apriori (mlxtend)           13        20         0.0383
FP-Growth (mlxtend)         13        20         0.0090
```

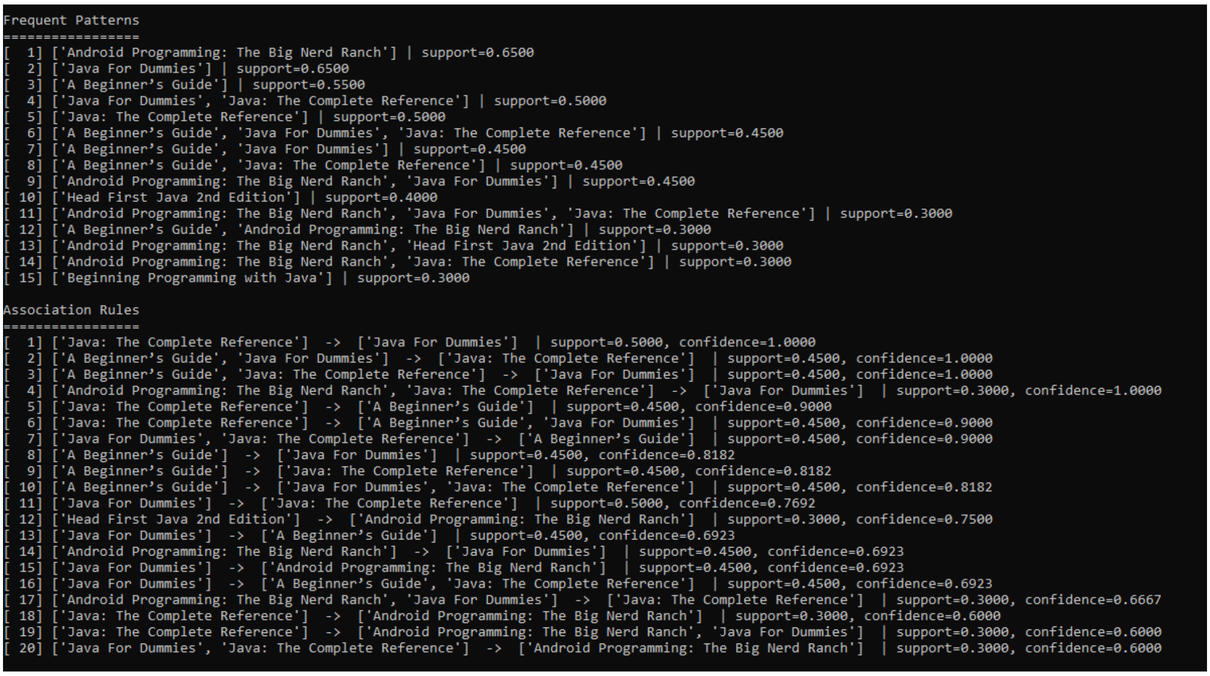
Based on the results we can say that brute force here runs quicker than the apriori and

Fp-growth because here the dataset we are dealing with is very small in size.

For large datasets the mlxtend versions of algorithms tends to perform better with good scalability.

5 Multiple Parameters

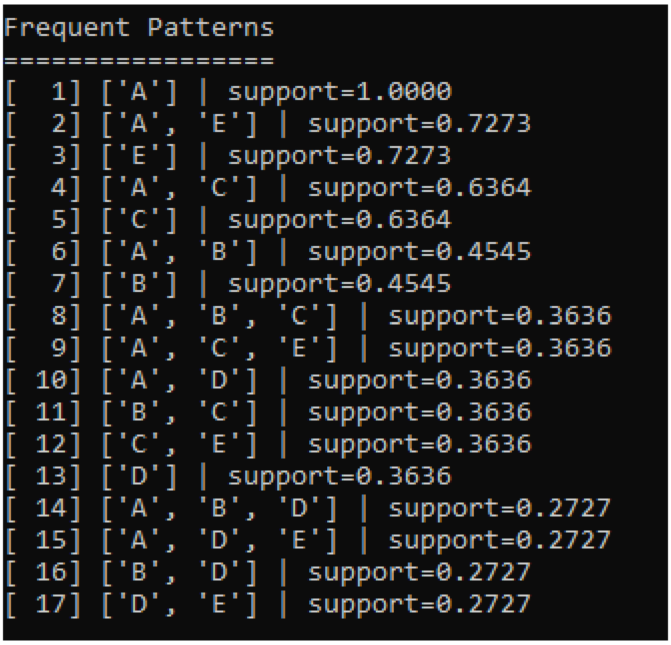
- Support = 0.3, Confidence = 0.6 for Amazon Dataset



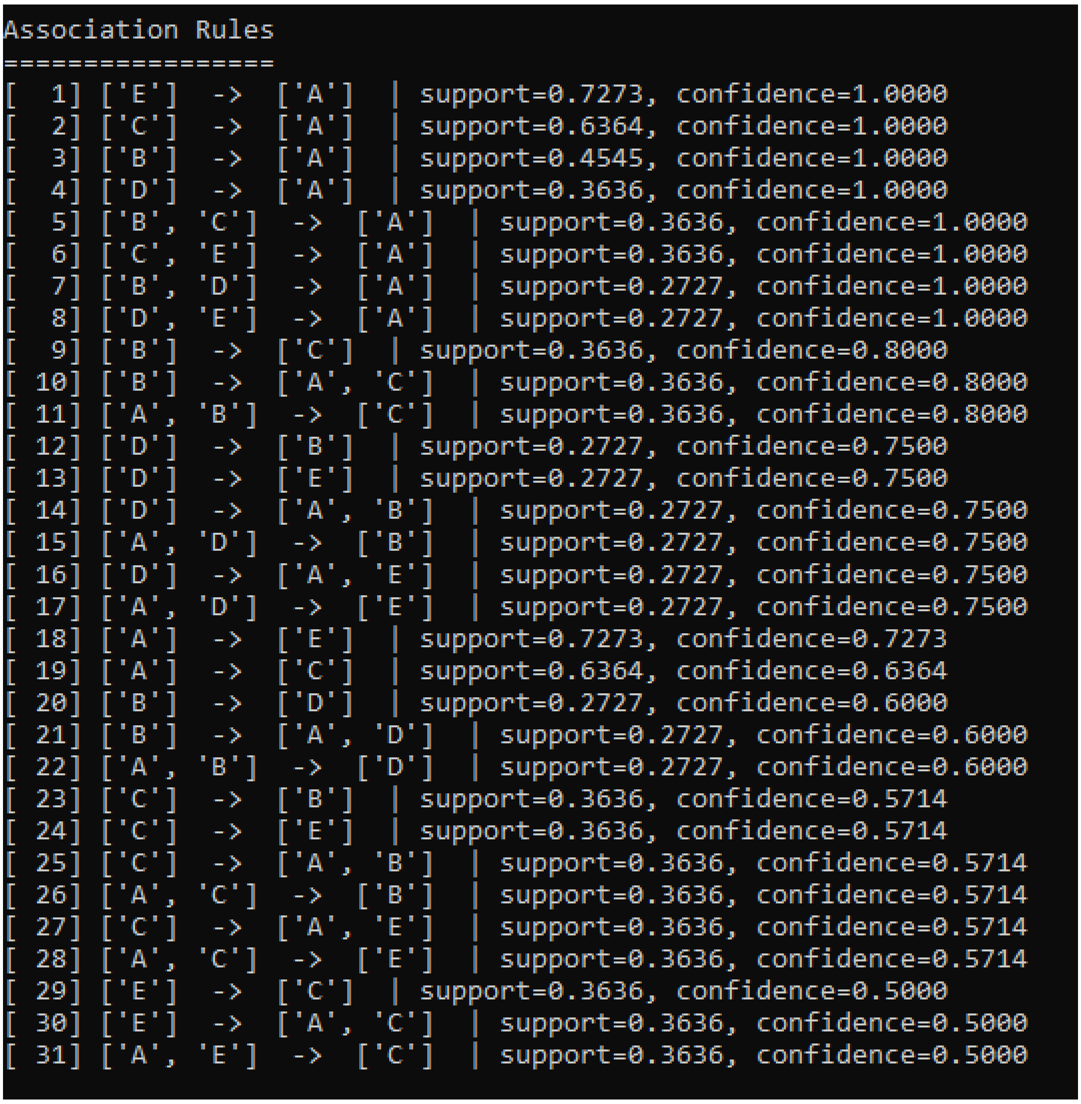
```
Frequent Patterns
=================
[  1] ['Android Programming: The Big Nerd Ranch'] | support=0.6500
[  2] ['Java For Dummies'] | support=0.6500
[  3] ['A Beginner’s Guide'] | support=0.5500
[  4] ['Java For Dummies', 'Java: The Complete Reference'] | support=0.5000
[  5] ['Java: The Complete Reference'] | support=0.5000
[  6] ['A Beginner’s Guide', 'Java For Dummies', 'Java: The Complete Reference'] | support=0.4500
[  7] ['A Beginner’s Guide', 'Java For Dummies'] | support=0.4500
[  8] ['A Beginner’s Guide', 'Java: The Complete Reference'] | support=0.4500
[  9] ['Android Programming: The Big Nerd Ranch', 'Java For Dummies'] | support=0.4500
[ 10] ['Head First Java 2nd Edition'] | support=0.4000
[ 11] ['Android Programming: The Big Nerd Ranch', 'Java For Dummies', 'Java: The Complete Reference'] | support=0.3000
[ 12] ['A Beginner’s Guide', 'Android Programming: The Big Nerd Ranch'] | support=0.3000
[ 13] ['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'] | support=0.3000
[ 14] ['Android Programming: The Big Nerd Ranch', 'Java: The Complete Reference'] | support=0.3000
[ 15] ['Beginning Programming with Java'] | support=0.3000

Association Rules
=================
[  1] ['Java: The Complete Reference']  ->  ['Java For Dummies']  | support=0.5000, confidence=1.0000
[  2] ['A Beginner’s Guide', 'Java For Dummies']  ->  ['Java: The Complete Reference']  | support=0.4500, confidence=1.0000
[  3] ['A Beginner’s Guide', 'Java: The Complete Reference']  ->  ['Java For Dummies']  | support=0.4500, confidence=1.0000
[  4] ['Android Programming: The Big Nerd Ranch', 'Java: The Complete Reference']  ->  ['Java For Dummies']  | support=0.3000, confidence=1.0000
[  5] ['Java: The Complete Reference']  ->  ['A Beginner’s Guide']  | support=0.4500, confidence=0.9000
[  6] ['Java: The Complete Reference']  ->  ['A Beginner’s Guide', 'Java For Dummies']  | support=0.4500, confidence=0.9000
[  7] ['Java For Dummies', 'Java: The Complete Reference']  ->  ['A Beginner’s Guide']  | support=0.4500, confidence=0.9000
[  8] ['A Beginner’s Guide']  ->  ['Java For Dummies']  | support=0.4500, confidence=0.8182
[  9] ['A Beginner’s Guide']  ->  ['Java: The Complete Reference']  | support=0.4500, confidence=0.8182
[ 10] ['A Beginner’s Guide']  ->  ['Java For Dummies', 'Java: The Complete Reference']  | support=0.4500, confidence=0.8182
[ 11] ['Java For Dummies']  ->  ['Java: The Complete Reference']  | support=0.5000, confidence=0.7692
[ 12] ['Head First Java 2nd Edition']  ->  ['Android Programming: The Big Nerd Ranch']  | support=0.3000, confidence=0.7500
[ 13] ['Java For Dummies']  ->  ['A Beginner’s Guide']  | support=0.4500, confidence=0.6923
[ 14] ['Android Programming: The Big Nerd Ranch']  ->  ['Java For Dummies']  | support=0.4500, confidence=0.6923
[ 15] ['Java For Dummies']  ->  ['Android Programming: The Big Nerd Ranch']  | support=0.4500, confidence=0.6923
[ 16] ['Java For Dummies']  ->  ['A Beginner’s Guide', 'Java: The Complete Reference']  | support=0.4500, confidence=0.6923
[ 17] ['Android Programming: The Big Nerd Ranch', 'Java For Dummies']  ->  ['Java: The Complete Reference']  | support=0.3000, confidence=0.6667
[ 18] ['Java: The Complete Reference']  ->  ['Android Programming: The Big Nerd Ranch']  | support=0.3000, confidence=0.6000
[ 19] ['Java: The Complete Reference']  ->  ['Android Programming: The Big Nerd Ranch', 'Java For Dummies']  | support=0.3000, confidence=0.6000
[ 20] ['Java For Dummies', 'Java: The Complete Reference']  ->  ['Android Programming: The Big Nerd Ranch']  | support=0.3000, confidence=0.6000
```

- Support = 0.2, Confidence = 0.5 for Generic Dataset

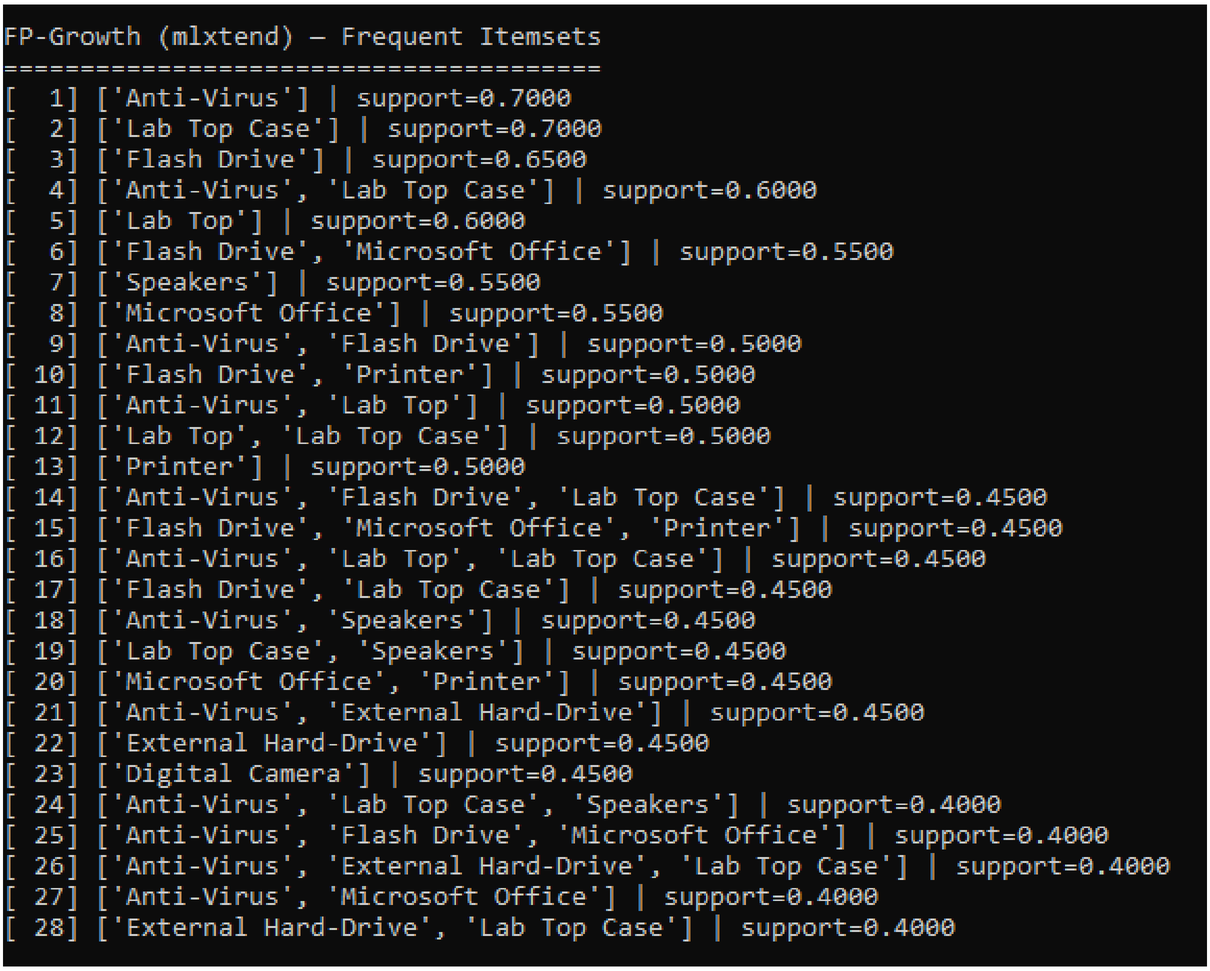


```
Frequent Patterns
=================
[  1] ['A'] | support=1.0000
[  2] ['A', 'E'] | support=0.7273
[  3] ['E'] | support=0.7273
[  4] ['A', 'C'] | support=0.6364
[  5] ['C'] | support=0.6364
[  6] ['A', 'B'] | support=0.4545
[  7] ['B'] | support=0.4545
[  8] ['A', 'B', 'C'] | support=0.3636
[  9] ['A', 'C', 'E'] | support=0.3636
[ 10] ['A', 'D'] | support=0.3636
[ 11] ['B', 'C'] | support=0.3636
[ 12] ['C', 'E'] | support=0.3636
[ 13] ['D'] | support=0.3636
[ 14] ['A', 'B', 'D'] | support=0.2727
[ 15] ['A', 'D', 'E'] | support=0.2727
[ 16] ['B', 'D'] | support=0.2727
[ 17] ['D', 'E'] | support=0.2727
```

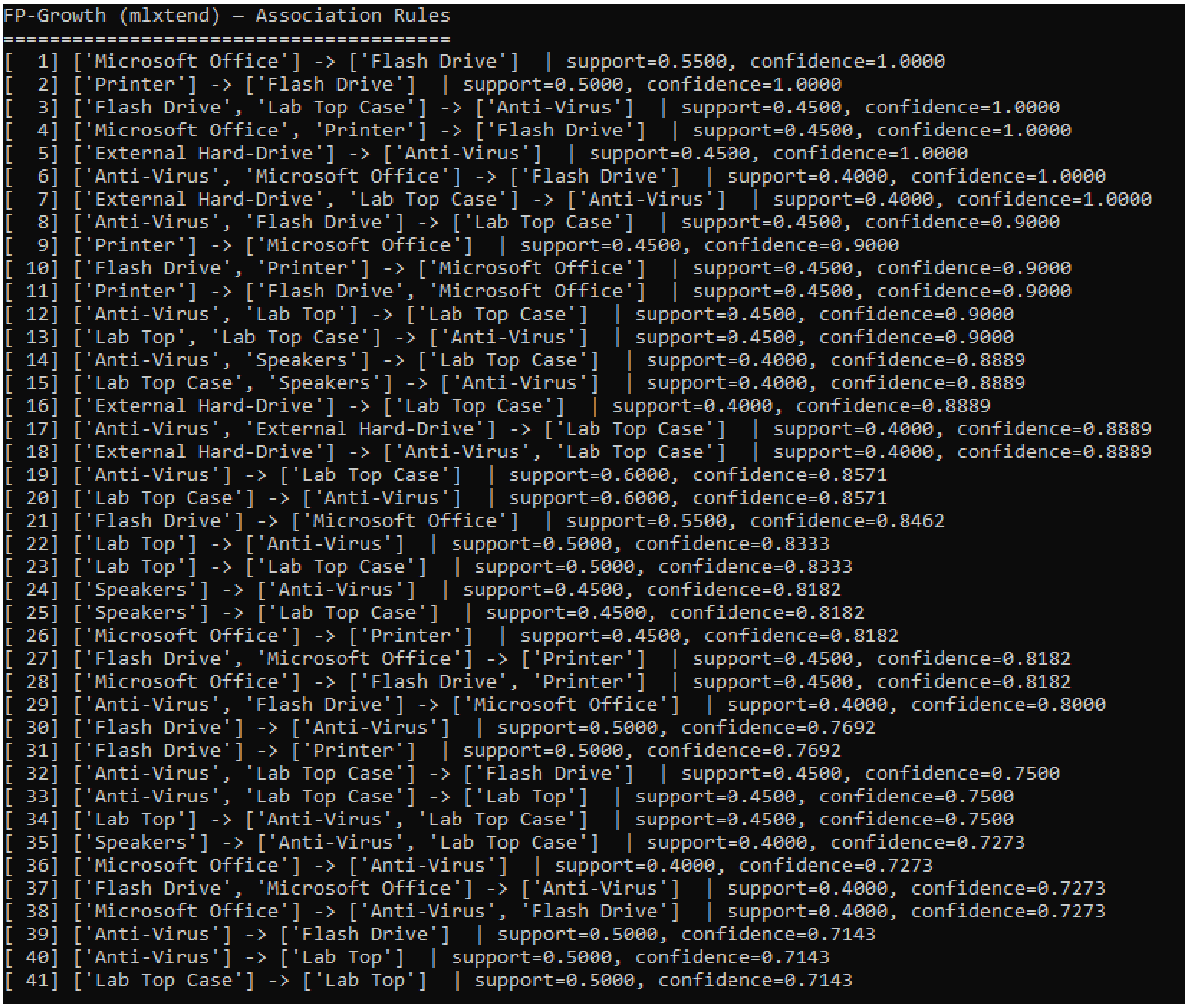


```
Association Rules
=================
[  1] ['E']  ->  ['A']  | support=0.7273, confidence=1.0000
[  2] ['C']  ->  ['A']  | support=0.6364, confidence=1.0000
[  3] ['B']  ->  ['A']  | support=0.4545, confidence=1.0000
[  4] ['D']  ->  ['A']  | support=0.3636, confidence=1.0000
[  5] ['B', 'C']  ->  ['A']  | support=0.3636, confidence=1.0000
[  6] ['C', 'E']  ->  ['A']  | support=0.3636, confidence=1.0000
[  7] ['B', 'D']  ->  ['A']  | support=0.2727, confidence=1.0000
[  8] ['D', 'E']  ->  ['A']  | support=0.2727, confidence=1.0000
[  9] ['B']  ->  ['C']  | support=0.3636, confidence=0.8000
[ 10] ['B']  ->  ['A', 'C']  | support=0.3636, confidence=0.8000
[ 11] ['A', 'B']  ->  ['C']  | support=0.3636, confidence=0.8000
[ 12] ['D']  ->  ['B']  | support=0.2727, confidence=0.7500
[ 13] ['D']  ->  ['E']  | support=0.2727, confidence=0.7500
[ 14] ['D']  ->  ['A', 'B']  | support=0.2727, confidence=0.7500
[ 15] ['A', 'D']  ->  ['B']  | support=0.2727, confidence=0.7500
[ 16] ['D']  ->  ['A', 'E']  | support=0.2727, confidence=0.7500
[ 17] ['A', 'D']  ->  ['E']  | support=0.2727, confidence=0.7500
[ 18] ['A']  ->  ['E']  | support=0.7273, confidence=0.7273
[ 19] ['A']  ->  ['C']  | support=0.6364, confidence=0.6364
[ 20] ['B']  ->  ['D']  | support=0.2727, confidence=0.6000
[ 21] ['B']  ->  ['A', 'D']  | support=0.2727, confidence=0.6000
[ 22] ['A', 'B']  ->  ['D']  | support=0.2727, confidence=0.6000
[ 23] ['C']  ->  ['B']  | support=0.3636, confidence=0.5714
[ 24] ['C']  ->  ['E']  | support=0.3636, confidence=0.5714
[ 25] ['C']  ->  ['A', 'B']  | support=0.3636, confidence=0.5714
[ 26] ['A', 'C']  ->  ['B']  | support=0.3636, confidence=0.5714
[ 27] ['C']  ->  ['A', 'E']  | support=0.3636, confidence=0.5714
[ 28] ['A', 'C']  ->  ['E']  | support=0.3636, confidence=0.5714
[ 29] ['E']  ->  ['C']  | support=0.3636, confidence=0.5000
[ 30] ['E']  ->  ['A', 'C']  | support=0.3636, confidence=0.5000
[ 31] ['A', 'E']  ->  ['C']  | support=0.3636, confidence=0.5000
```

- Support = 0.4, Confidence = 0.7 for BestBuy Dataset



```
FP-Growth (mlxtend) — Frequent Itemsets
=======================================
[  1] ['Anti-Virus'] | support=0.7000
[  2] ['Lab Top Case'] | support=0.7000
[  3] ['Flash Drive'] | support=0.6500
[  4] ['Anti-Virus', 'Lab Top Case'] | support=0.6000
[  5] ['Lab Top'] | support=0.6000
[  6] ['Flash Drive', 'Microsoft Office'] | support=0.5500
[  7] ['Speakers'] | support=0.5500
[  8] ['Microsoft Office'] | support=0.5500
[  9] ['Anti-Virus', 'Flash Drive'] | support=0.5000
[ 10] ['Flash Drive', 'Printer'] | support=0.5000
[ 11] ['Anti-Virus', 'Lab Top'] | support=0.5000
[ 12] ['Lab Top', 'Lab Top Case'] | support=0.5000
[ 13] ['Printer'] | support=0.5000
[ 14] ['Anti-Virus', 'Flash Drive', 'Lab Top Case'] | support=0.4500
[ 15] ['Flash Drive', 'Microsoft Office', 'Printer'] | support=0.4500
[ 16] ['Anti-Virus', 'Lab Top', 'Lab Top Case'] | support=0.4500
[ 17] ['Flash Drive', 'Lab Top Case'] | support=0.4500
[ 18] ['Anti-Virus', 'Speakers'] | support=0.4500
[ 19] ['Lab Top Case', 'Speakers'] | support=0.4500
[ 20] ['Microsoft Office', 'Printer'] | support=0.4500
[ 21] ['Anti-Virus', 'External Hard-Drive'] | support=0.4500
[ 22] ['External Hard-Drive'] | support=0.4500
[ 23] ['Digital Camera'] | support=0.4500
[ 24] ['Anti-Virus', 'Lab Top Case', 'Speakers'] | support=0.4000
[ 25] ['Anti-Virus', 'Flash Drive', 'Microsoft Office'] | support=0.4000
[ 26] ['Anti-Virus', 'External Hard-Drive', 'Lab Top Case'] | support=0.4000
[ 27] ['Anti-Virus', 'Microsoft Office'] | support=0.4000
[ 28] ['External Hard-Drive', 'Lab Top Case'] | support=0.4000
```



```
FP-Growth (mlxtend) — Association Rules
========================================
[  1] ['Microsoft Office'] -> ['Flash Drive']  | support=0.5500, confidence=1.0000
[  2] ['Printer'] -> ['Flash Drive']  | support=0.5000, confidence=1.0000
[  3] ['Flash Drive', 'Lab Top Case'] -> ['Anti-Virus']  | support=0.4500, confidence=1.0000
[  4] ['Microsoft Office', 'Printer'] -> ['Flash Drive']  | support=0.4500, confidence=1.0000
[  5] ['External Hard-Drive'] -> ['Anti-Virus']  | support=0.4500, confidence=1.0000
[  6] ['Anti-Virus', 'Microsoft Office'] -> ['Flash Drive']  | support=0.4000, confidence=1.0000
[  7] ['External Hard-Drive', 'Lab Top Case'] -> ['Anti-Virus']  | support=0.4000, confidence=1.0000
[  8] ['Anti-Virus', 'Flash Drive'] -> ['Lab Top Case']  | support=0.4500, confidence=0.9000
[  9] ['Printer'] -> ['Microsoft Office']  | support=0.4500, confidence=0.9000
[ 10] ['Flash Drive', 'Printer'] -> ['Microsoft Office']  | support=0.4500, confidence=0.9000
[ 11] ['Printer'] -> ['Flash Drive', 'Microsoft Office']  | support=0.4500, confidence=0.9000
[ 12] ['Anti-Virus', 'Lab Top'] -> ['Lab Top Case']  | support=0.4500, confidence=0.9000
[ 13] ['Lab Top', 'Lab Top Case'] -> ['Anti-Virus']  | support=0.4500, confidence=0.9000
[ 14] ['Anti-Virus', 'Speakers'] -> ['Lab Top Case']  | support=0.4000, confidence=0.8889
[ 15] ['Lab Top Case', 'Speakers'] -> ['Anti-Virus']  | support=0.4000, confidence=0.8889
[ 16] ['External Hard-Drive'] -> ['Lab Top Case']  | support=0.4000, confidence=0.8889
[ 17] ['Anti-Virus', 'External Hard-Drive'] -> ['Lab Top Case']  | support=0.4000, confidence=0.8889
[ 18] ['External Hard-Drive'] -> ['Anti-Virus', 'Lab Top Case']  | support=0.4000, confidence=0.8889
[ 19] ['Anti-Virus'] -> ['Lab Top Case']  | support=0.6000, confidence=0.8571
[ 20] ['Lab Top Case'] -> ['Anti-Virus']  | support=0.6000, confidence=0.8571
[ 21] ['Flash Drive'] -> ['Microsoft Office']  | support=0.5500, confidence=0.8462
[ 22] ['Lab Top'] -> ['Anti-Virus']  | support=0.5000, confidence=0.8333
[ 23] ['Lab Top'] -> ['Lab Top Case']  | support=0.5000, confidence=0.8333
[ 24] ['Speakers'] -> ['Anti-Virus']  | support=0.4500, confidence=0.8182
[ 25] ['Speakers'] -> ['Lab Top Case']  | support=0.4500, confidence=0.8182
[ 26] ['Microsoft Office'] -> ['Printer']  | support=0.4500, confidence=0.8182
[ 27] ['Flash Drive', 'Microsoft Office'] -> ['Printer']  | support=0.4500, confidence=0.8182
[ 28] ['Microsoft Office'] -> ['Flash Drive', 'Printer']  | support=0.4500, confidence=0.8182
[ 29] ['Anti-Virus', 'Flash Drive'] -> ['Microsoft Office']  | support=0.4000, confidence=0.8000
[ 30] ['Flash Drive'] -> ['Anti-Virus']  | support=0.5000, confidence=0.7692
[ 31] ['Flash Drive'] -> ['Printer']  | support=0.5000, confidence=0.7692
[ 32] ['Anti-Virus', 'Lab Top Case'] -> ['Flash Drive']  | support=0.4500, confidence=0.7500
[ 33] ['Anti-Virus', 'Lab Top Case'] -> ['Lab Top']  | support=0.4500, confidence=0.7500
[ 34] ['Lab Top'] -> ['Anti-Virus', 'Lab Top Case']  | support=0.4500, confidence=0.7500
[ 35] ['Speakers'] -> ['Anti-Virus', 'Lab Top Case']  | support=0.4000, confidence=0.7273
[ 36] ['Microsoft Office'] -> ['Anti-Virus']  | support=0.4000, confidence=0.7273
[ 37] ['Flash Drive', 'Microsoft Office'] -> ['Anti-Virus']  | support=0.4000, confidence=0.7273
[ 38] ['Microsoft Office'] -> ['Anti-Virus', 'Flash Drive']  | support=0.4000, confidence=0.7273
[ 39] ['Anti-Virus'] -> ['Flash Drive']  | support=0.5000, confidence=0.7143
[ 40] ['Anti-Virus'] -> ['Lab Top']  | support=0.5000, confidence=0.7143
[ 41] ['Lab Top Case'] -> ['Lab Top']  | support=0.5000, confidence=0.7143
```

6 GitHub Repository

Link: https://github.com/rajeevalahari/alahari_rajeevkumar_midtermproject

7 How to run the code

7.1 Install Requirements:

1. The basic requirements needed are pandas and mlxtend.
   pip install pandas
   pip install mlxtend

(OPTIONAL)

1. If you face any issues with requirements I have frozen the requirements in my machine
2. You can install them by changing directory to alahari_rajeevkumar_midtermproject folder and run
   pip install -r requirements.txt

3. If you don’t want to install requirements in our entire system and want to create a virtual environment and install requirements in it and use it for the program execution.

   ```
   python -m venv venv
   venv\Scripts\activate
   ```

   this creates virtual environment and activates it in windows.

7.2 Run Options

PYTHON FILE

1. Extract the file:
   Unzip “alahari_rajeevkumar_midtermproject.zip” to a folder on your computer.
2. Open Command Prompt (Windows) or Terminal (macOS/Linux).
3. Change into the src folder
4. Run the program:
   Windows: python code.py
   macOS/Linux: python3 code.py
5. Main menu options:
   Enter 1–5 to select a dataset (each number is a different dataset).
   Enter 0 to Exit.
6. After selecting a dataset:
   You will be asked for support (e.g., 0.30) and confidence (e.g., 0.40).
   The program generates frequent itemsets and association rules using:
   Brute Force
   Apriori (mlxtend)
   FP-Growth (mlxtend)
   It also prints timing comparisons across all three approaches.
7. Run again or exit:
   When prompted, type:
   y → returns to the home page (run again)
   n → exits the program

IPYNB FILE

1. Open your notebook tool (Jupyter notebook).
2. Change into the notebook folder or open the notebook from there:
3. Run the notebook cell by cell from top to bottom.
4. Behavior is the same as the Python CLI:
   You’ll get prompts for support and confidence.
   Frequent itemsets, association rules, and timing comparisons are displayed.

RUN FROM GITHUB

1. Clone the repository:
   git clone https://github.com/rajeevalahari/alahari_rajeevkumar_midtermproject
2. Enter the project folder:
3. Follow the same steps as above:
   For python file: cd into src and run python code.py (or python3 on macOS/Linux).

For Notebook: cd into notebook and run the notebook cell by cell.

8 Conclusion

This project demonstrated and compared three core approaches to frequent itemset mining and association rule learning Brute Force (from scratch), Apriori, and FP-Growth (via `mlxtend`)—across multiple retail-style datasets. Brute Force served as a transparent baseline, showing the mechanics of exhaustive candidate generation and why the search space grows exponentially. Apriori improved practicality by pruning with the downward-closure property, while FP-Growth avoided candidate generation entirely through an FP-Tree, yielding the most scalable path for larger or denser datasets. On our relatively small datasets, all three approaches produced consistent frequent itemsets and rules, with timing differences modest; however, the algorithms' theoretical strengths indicate that Apriori and especially FP-Growth are the preferred choices for real-world, large-scale mining.

From an engineering standpoint, care was taken to make the command-line workflow reliable and user-friendly. Defensive programming techniques include strict input validation for the dataset menu (accepting only 0–5, with clear re-prompts), numeric checks for support and confidence (enforcing valid floats in the [0, 1] range), and loop-based prompts that gracefully handle mistakes without crashing. Additional safeguards—such as try/except blocks, file/path existence checks, and clean exit options—ensure that the program fails safely, guides the user to correct inputs, and supports repeated runs from a single session. Together, these design choices make the implementation both pedagogically clear and robust enough for exploratory analysis.