

CS634 – Data Mining

Midterm Project Report

Name: Nandni Patel

Email: np888@njit.edu

Course: CS634 – Data Mining

Instructor: Dr. Yasser Abdallah

1. Introduction

This report documents the process of implementing frequent itemset mining and association rule learning. This report documents the process of implementing frequent itemset mining and association rule learning using three different approaches. The objective is to explore and compare how each method identifies patterns and relationships within transactional data. The methods implemented are:

- Brute Force Method – implemented from scratch to understand the fundamental logic behind frequent itemset generation and rule formation.
- Apriori Algorithm – implemented using a Python library to demonstrate an efficient, candidate-generation-based approach.
- FP-Growth Algorithm – implemented using a Python library to explore a more scalable and memory-efficient alternative to Apriori.

2. Dataset Creation

2.1 Items:

I created 5 different datasets, each representing transactions from a specific store: Amazon, BestBuy, Kmart, Nike, and Generic. Each dataset contains a unique list of items corresponding to the type of products sold at that store.

Below is the item summary for each:

1. Amazon:
Programming and web development books such as A Beginner's Guide, Java: The Complete Reference, Java For Dummies, Head First Java 2nd Edition, and HTML and CSS: Design and Build Websites.
2. BestBuy:
Electronics and accessories like Digital Camera, Laptop, Printer, Flash Drive, Microsoft Office, Speakers, and External Hard Drive.
3. Kmart:
Home and bedding products such as Quilts, Bedspreads, Decorative Pillows, Sheets, and Kids Bedding.
4. Nike:
Sportswear and athletic items including Running Shoe, Soccer Shoe, Sweatshirts, Rash Guard, Tech Pants, and Hoodies.

5. Generic:

A simple labeled dataset with items A–F, used to illustrate smaller-scale, controlled transactions.

2.2 Transactions:

For each store, I created at least 20 deterministic transactions. Each transaction represents a basket of items that a customer might buy together. The transactions were designed manually to reflect logical purchasing patterns (for example, “Laptop” often appears with “Laptop Case” and “Microsoft Office”).

Below is a short example snippet from the Amazon dataset:

TID	Transactions
1	A Beginner’s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch
2	A Beginner’s Guide, Java: The Complete Reference, Java For Dummies
3	A Beginner’s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
4	Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition, Beginning Programming with Java

Each CSV file contains similar transaction data for its respective store:

- Amazon.csv
- BestBuy.csv
- Kmart.csv
- Nike.csv
- Generic.csv

All files follow the same consistent format:

TID, Transactions

1, Item1, Item2, Item3

2, Item2, Item4, Item5

2.3 Dataset Notes:

- All datasets were created manually in Excel and then exported as CSV files.
- Each dataset is deterministic, transactions were not randomly generated. Logical associations were maintained (e.g., “Printer” often pairs with “Flash Drive” and “Microsoft Office”).
- The datasets are intentionally kept lightweight (a few KB each) for ease of sharing and use in association rule mining or market basket analysis exercises.
- Each CSV file can be easily imported into Python (using pandas) or data mining tools like Orange, Weka, or RapidMiner for analysis.

3. Brute Force Algorithm

The Brute Force Algorithm is a straightforward method that checks all possible combinations of items to find those meeting the minimum support and confidence thresholds. It evaluates every itemset and rule exhaustively to identify strong

associations. Although simple and easy to implement, it becomes very slow for large datasets due to the exponential number of combinations.

3.1 Method:

- Generate all possible 1-itemsets and calculate their support by checking how many transactions contain each item.
- Identify itemsets that satisfy the minimum support threshold (user-defined).
- Combine frequent k-itemsets to generate new candidate (k+1)-itemsets.
- Repeat the process until no more frequent itemsets can be generated.
- Using the frequent itemsets, generate association rules that satisfy the minimum confidence threshold provided by the user.
- Display all frequent itemsets and their support, followed by association rules and their confidence.

3.2 Example Run:

a) Dataset: Amazon.csv

Parameters: Support = 0.3, Confidence = 0.6

```
Running Brute Force Algorithm
```

```
♦ Checking 1-itemsets ...
```

```
{ "A Beginner's Guide" } (support=0.55)  
{ 'Android Programming: The Big Nerd Ranch' } (support=0.65)  
{ 'Beginning Programming with Java' } (support=0.30)  
{ 'Head First Java 2nd Edition' } (support=0.40)  
{ 'Java For Dummies' } (support=0.65)  
{ 'Java: The Complete Reference' } (support=0.50)
```

```
♦ Checking 2-itemsets ...
```

```
{ 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition' } (support=0.30)  
{ "A Beginner's Guide", 'Java: The Complete Reference' } (support=0.45)  
{ "A Beginner's Guide", 'Java For Dummies' } (support=0.45)  
{ 'Java For Dummies', 'Java: The Complete Reference' } (support=0.50)  
{ 'Android Programming: The Big Nerd Ranch', 'Java: The Complete Reference' } (support=0.30)  
{ 'Android Programming: The Big Nerd Ranch', "A Beginner's Guide" } (support=0.30)  
{ 'Android Programming: The Big Nerd Ranch', 'Java For Dummies' } (support=0.45)
```

```
♦ Checking 3-itemsets ...
```

```
{ 'Android Programming: The Big Nerd Ranch', 'Java For Dummies', 'Java: The Complete Reference' } (support=0.30)  
{ "A Beginner's Guide", 'Java For Dummies', 'Java: The Complete Reference' } (support=0.45)
```

```
♦ Checking 4-itemsets ...
```

```

=== Association Rules (Brute Force) ===
{'Head First Java 2nd Edition'} → {'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.75)
{'A Beginner's Guide'} → {'Java: The Complete Reference'} (support=0.45, confidence=0.82)
{'Java: The Complete Reference'} → {'A Beginner's Guide'} (support=0.45, confidence=0.90)
{'A Beginner's Guide'} → {'Java For Dummies'} (support=0.45, confidence=0.82)
{'Java For Dummies'} → {'A Beginner's Guide'} (support=0.45, confidence=0.69)
{'Java For Dummies'} → {'Java: The Complete Reference'} (support=0.50, confidence=0.77)
{'Java: The Complete Reference'} → {'Java For Dummies'} (support=0.50, confidence=1.00)
{'Java: The Complete Reference'} → {'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.60)
{'Android Programming: The Big Nerd Ranch'} → {'Java For Dummies'} (support=0.45, confidence=0.69)
{'Java For Dummies'} → {'Android Programming: The Big Nerd Ranch'} (support=0.45, confidence=0.69)
{'Java: The Complete Reference'} → {'Android Programming: The Big Nerd Ranch', 'Java For Dummies'} (support=0.30, confidence=0.60)
{'Android Programming: The Big Nerd Ranch', 'Java For Dummies'} → {'Java: The Complete Reference'} (support=0.30, confidence=0.67)
{'Android Programming: The Big Nerd Ranch', 'Java: The Complete Reference'} → {'Java For Dummies'} (support=0.30, confidence=1.00)
{'Java For Dummies', 'Java: The Complete Reference'} → {'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.60)
{'A Beginner's Guide'} → {'Java For Dummies', 'Java: The Complete Reference'} (support=0.45, confidence=0.82)
{'Java For Dummies'} → {'A Beginner's Guide', 'Java: The Complete Reference'} (support=0.45, confidence=0.69)
{'Java: The Complete Reference'} → {'A Beginner's Guide', 'Java For Dummies'} (support=0.45, confidence=0.90)
{'A Beginner's Guide', 'Java For Dummies'} → {'Java: The Complete Reference'} (support=0.45, confidence=1.00)
{'A Beginner's Guide', 'Java: The Complete Reference'} → {'Java For Dummies'} (support=0.45, confidence=1.00)
{'Java For Dummies', 'Java: The Complete Reference'} → {'A Beginner's Guide'} (support=0.45, confidence=0.90)

=== Brute Force Summary ===
Dataset used: Amazon.csv
Total transactions: 20
Frequent itemsets found: 15
Association rules found: 20
Execution time: 0.505897 s

```

b) Dataset: Nike.csv

Parameters: Support = 0.5, Confidence = 0.6

Running Brute Force Algorithm

- ◆ Checking 1-itemsets ...
 - { 'Modern Pants' } (support=0.50)
 - { 'Rash Guard' } (support=0.60)
 - { 'Running Shoe' } (support=0.70)
 - { 'Socks' } (support=0.65)
 - { 'Sweatshirts' } (support=0.65)
 - { 'Swimming Shirt' } (support=0.55)
- ◆ Checking 2-itemsets ...
 - { 'Sweatshirts', 'Modern Pants' } (support=0.50)
 - { 'Running Shoe', 'Socks' } (support=0.55)
 - { 'Sweatshirts', 'Running Shoe' } (support=0.55)
 - { 'Swimming Shirt', 'Rash Guard' } (support=0.50)
 - { 'Sweatshirts', 'Socks' } (support=0.60)
- ◆ Checking 3-itemsets ...
 - { 'Sweatshirts', 'Running Shoe', 'Socks' } (support=0.50)
- ◆ Checking 4-itemsets ...

```

=== Association Rules (Brute Force) ===
{'Modern Pants'} → {'Sweatshirts'} (support=0.50, confidence=1.00)
{'Sweatshirts'} → {'Modern Pants'} (support=0.50, confidence=0.77)
{'Running Shoe'} → {'Socks'} (support=0.55, confidence=0.79)
{'Socks'} → {'Running Shoe'} (support=0.55, confidence=0.85)
{'Running Shoe'} → {'Sweatshirts'} (support=0.55, confidence=0.79)
{'Sweatshirts'} → {'Running Shoe'} (support=0.55, confidence=0.85)
{'Rash Guard'} → {'Swimming Shirt'} (support=0.50, confidence=0.83)
{'Swimming Shirt'} → {'Rash Guard'} (support=0.50, confidence=0.91)
{'Socks'} → {'Sweatshirts'} (support=0.60, confidence=0.92)
{'Sweatshirts'} → {'Socks'} (support=0.60, confidence=0.92)
{'Running Shoe'} → {'Sweatshirts', 'Socks'} (support=0.50, confidence=0.71)
{'Socks'} → {'Sweatshirts', 'Running Shoe'} (support=0.50, confidence=0.77)
{'Sweatshirts'} → {'Running Shoe', 'Socks'} (support=0.50, confidence=0.77)
{'Running Shoe', 'Socks'} → {'Sweatshirts'} (support=0.50, confidence=0.91)
{'Sweatshirts', 'Running Shoe'} → {'Socks'} (support=0.50, confidence=0.91)
{'Sweatshirts', 'Socks'} → {'Running Shoe'} (support=0.50, confidence=0.83)

=== Brute Force Summary ===
Dataset used: Nike.csv
Total transactions: 20
Frequent itemsets found: 12
Association rules found: 16
Execution time: 0.504632 s

```

- c) Dataset: Generic.csv
Parameters: Support = 0.3, Confidence = 0.7

Running Brute Force Algorithm

◆ Checking 1-itemsets ...

```

{'A'} (support=1.00)
{'B'} (support=0.50)
{'C'} (support=0.60)
{'D'} (support=0.50)
{'E'} (support=0.70)
{'F'} (support=0.45)

```

◆ Checking 2-itemsets ...

```

{'D', 'A'} (support=0.50)
{'D', 'F'} (support=0.30)
{'E', 'A'} (support=0.70)
{'D', 'B'} (support=0.30)
{'C', 'A'} (support=0.60)
{'C', 'E'} (support=0.30)
{'D', 'E'} (support=0.35)
{'B', 'C'} (support=0.35)
{'F', 'A'} (support=0.45)
{'F', 'E'} (support=0.30)
{'B', 'A'} (support=0.50)

```

◆ Checking 3-itemsets ...

```

{'C', 'E', 'A'} (support=0.30)
{'D', 'E', 'A'} (support=0.35)
{'F', 'E', 'A'} (support=0.30)
{'B', 'C', 'A'} (support=0.35)
{'D', 'F', 'A'} (support=0.30)
{'D', 'B', 'A'} (support=0.30)

```

◆ Checking 4-itemsets ...


```

=== Association Rules (Brute Force) ===
{'D'} → {'A'} (support=0.50, confidence=1.00)
{'A'} → {'E'} (support=0.70, confidence=0.70)
{'E'} → {'A'} (support=0.70, confidence=1.00)
{'C'} → {'A'} (support=0.60, confidence=1.00)
{'D'} → {'E'} (support=0.35, confidence=0.70)
{'B'} → {'C'} (support=0.35, confidence=0.70)
{'F'} → {'A'} (support=0.45, confidence=1.00)
{'B'} → {'A'} (support=0.50, confidence=1.00)
{'C', 'E'} → {'A'} (support=0.30, confidence=1.00)
{'D'} → {'E', 'A'} (support=0.35, confidence=0.70)
{'D', 'A'} → {'E'} (support=0.35, confidence=0.70)
{'D', 'E'} → {'A'} (support=0.35, confidence=1.00)
{'F', 'E'} → {'A'} (support=0.30, confidence=1.00)
{'B'} → {'C', 'A'} (support=0.35, confidence=0.70)
{'B', 'A'} → {'C'} (support=0.35, confidence=0.70)
{'B', 'C'} → {'A'} (support=0.35, confidence=1.00)
{'D', 'F'} → {'A'} (support=0.30, confidence=1.00)
{'D', 'B'} → {'A'} (support=0.30, confidence=1.00)

=== Brute Force Summary ===
Dataset used: Generic.csv
Total transactions: 20
Frequent itemsets found: 23
Association rules found: 18
Execution time: 0.503241 s

```

- d) Dataset: BestBuy.csv
Parameters: Support = 0.4, Confidence = 0.8

```

Running Brute Force Algorithm

♦ Checking 1-itemsets ...
{'Anti-Virus'} (support=0.70)
{'Digital Camera'} (support=0.45)
{'External Hard-Drive'} (support=0.45)
{'Flash Drive'} (support=0.65)
{'Lap Top'} (support=0.60)
{'Lap Top Case'} (support=0.70)
{'Microsoft Office'} (support=0.55)
{'Printer'} (support=0.50)
{'Speakers'} (support=0.55)

♦ Checking 2-itemsets ...
{'Printer', 'Flash Drive'} (support=0.50)
{'Lap Top Case', 'Flash Drive'} (support=0.45)
{'External Hard-Drive', 'Anti-Virus'} (support=0.45)
{'Anti-Virus', 'Microsoft Office'} (support=0.40)
{'External Hard-Drive', 'Lap Top Case'} (support=0.40)
{'Anti-Virus', 'Lap Top Case'} (support=0.60)
{'Anti-Virus', 'Lap Top'} (support=0.50)
{'Speakers', 'Lap Top Case'} (support=0.45)
{'Anti-Virus', 'Speakers'} (support=0.45)
{'Anti-Virus', 'Flash Drive'} (support=0.50)
{'Microsoft Office', 'Printer'} (support=0.45)
{'Lap Top Case', 'Lap Top'} (support=0.50)
{'Microsoft Office', 'Flash Drive'} (support=0.55)

♦ Checking 3-itemsets ...
{'Microsoft Office', 'Printer', 'Flash Drive'} (support=0.45)
{'Lap Top Case', 'Anti-Virus', 'Lap Top'} (support=0.45)
{'Anti-Virus', 'Microsoft Office', 'Flash Drive'} (support=0.40)
{'Anti-Virus', 'Lap Top Case', 'Flash Drive'} (support=0.45)
{'External Hard-Drive', 'Anti-Virus', 'Lap Top Case'} (support=0.40)
{'Anti-Virus', 'Speakers', 'Lap Top Case'} (support=0.40)

♦ Checking 4-itemsets ...

```

```

=== Association Rules (Brute Force) ===
{'Printer'} → {'Flash Drive'} (support=0.50, confidence=1.00)
{'External Hard-Drive'} → {'Anti-Virus'} (support=0.45, confidence=1.00)
{'External Hard-Drive'} → {'Lap Top Case'} (support=0.40, confidence=0.89)
{'Anti-Virus'} → {'Lap Top Case'} (support=0.60, confidence=0.86)
{'Lap Top Case'} → {'Anti-Virus'} (support=0.60, confidence=0.86)
{'Lap Top'} → {'Anti-Virus'} (support=0.50, confidence=0.83)
{'Speakers'} → {'Lap Top Case'} (support=0.45, confidence=0.82)
{'Speakers'} → {'Anti-Virus'} (support=0.45, confidence=0.82)
{'Microsoft Office'} → {'Printer'} (support=0.45, confidence=0.82)
{'Printer'} → {'Microsoft Office'} (support=0.45, confidence=0.90)
{'Lap Top'} → {'Lap Top Case'} (support=0.50, confidence=0.83)
{'Flash Drive'} → {'Microsoft Office'} (support=0.55, confidence=0.85)
{'Microsoft Office'} → {'Flash Drive'} (support=0.55, confidence=1.00)
{'Microsoft Office'} → {'Printer', 'Flash Drive'} (support=0.45, confidence=0.82)
{'Printer'} → {'Microsoft Office', 'Flash Drive'} (support=0.45, confidence=0.90)
{'Microsoft Office', 'Flash Drive'} → {'Printer'} (support=0.45, confidence=0.82)
{'Printer', 'Flash Drive'} → {'Microsoft Office'} (support=0.45, confidence=0.90)
{'Microsoft Office', 'Printer'} → {'Flash Drive'} (support=0.45, confidence=1.00)
{'Anti-Virus', 'Lap Top'} → {'Lap Top Case'} (support=0.45, confidence=0.90)
{'Lap Top Case', 'Lap Top'} → {'Anti-Virus'} (support=0.45, confidence=0.90)
{'Anti-Virus', 'Flash Drive'} → {'Microsoft Office'} (support=0.40, confidence=0.80)
{'Anti-Virus', 'Microsoft Office'} → {'Flash Drive'} (support=0.40, confidence=1.00)
{'Anti-Virus', 'Flash Drive'} → {'Lap Top Case'} (support=0.45, confidence=0.90)
{'Lap Top Case', 'Flash Drive'} → {'Anti-Virus'} (support=0.45, confidence=1.00)
{'External Hard-Drive'} → {'Anti-Virus', 'Lap Top Case'} (support=0.40, confidence=0.89)
{'External Hard-Drive', 'Anti-Virus'} → {'Lap Top Case'} (support=0.40, confidence=0.89)
{'External Hard-Drive', 'Lap Top Case'} → {'Anti-Virus'} (support=0.40, confidence=1.00)
{'Anti-Virus', 'Speakers'} → {'Lap Top Case'} (support=0.40, confidence=0.89)
{'Speakers', 'Lap Top Case'} → {'Anti-Virus'} (support=0.40, confidence=0.89)

=== Brute Force Summary ===
Dataset used: BestBuy.csv
Total transactions: 20
Frequent itemsets found: 28
Association rules found: 29
Execution time: 0.503471 s

```

e) Dataset: Kmart.csv

Parameters: Support = 0.3, Confidence = 0.5

```

♦ Checking 1-itemsets ...
{'Bed Skirts'} (support=0.55)
{'Bedding Collections'} (support=0.35)
{'Bedspreads'} (support=0.35)
{'Decorative Pillows'} (support=0.50)
{'Embroidered Bedspread'} (support=0.30)
{'Kids Bedding'} (support=0.60)
{'Quilts'} (support=0.40)
{'Shams'} (support=0.55)
{'Sheets'} (support=0.50)

♦ Checking 2-itemsets ...
{'Bedspreads', 'Bed Skirts'} (support=0.35)
{'Shams', 'Kids Bedding'} (support=0.45)
{'Bedding Collections', 'Kids Bedding'} (support=0.30)
{'Shams', 'Bed Skirts'} (support=0.45)
{'Sheets', 'Kids Bedding'} (support=0.50)
{'Bedspreads', 'Kids Bedding'} (support=0.35)
{'Shams', 'Sheets'} (support=0.35)
{'Bedspreads', 'Sheets'} (support=0.35)
{'Bed Skirts', 'Kids Bedding'} (support=0.50)
{'Sheets', 'Bed Skirts'} (support=0.45)
{'Quilts', 'Decorative Pillows'} (support=0.30)

♦ Checking 3-itemsets ...
{'Bedspreads', 'Bed Skirts', 'Kids Bedding'} (support=0.35)
{'Shams', 'Bed Skirts', 'Sheets'} (support=0.35)
{'Bedspreads', 'Bed Skirts', 'Sheets'} (support=0.35)
{'Bedspreads', 'Kids Bedding', 'Sheets'} (support=0.35)
{'Shams', 'Bed Skirts', 'Kids Bedding'} (support=0.40)
{'Shams', 'Kids Bedding', 'Sheets'} (support=0.35)
{'Sheets', 'Bed Skirts', 'Kids Bedding'} (support=0.45)

♦ Checking 4-itemsets ...
{'Shams', 'Bed Skirts', 'Kids Bedding', 'Sheets'} (support=0.35)
{'Bedspreads', 'Bed Skirts', 'Kids Bedding', 'Sheets'} (support=0.35)

```

```

=== Association Rules (Brute Force) ===
{'Bed Skirts'} → {'Bedspreads'} (support=0.35, confidence=0.64)
{'Bedspreads'} → {'Bed Skirts'} (support=0.35, confidence=1.00)
{'Kids Bedding'} → {'Shams'} (support=0.45, confidence=0.75)
{'Shams'} → {'Kids Bedding'} (support=0.45, confidence=0.82)
{'Bedding Collections'} → {'Kids Bedding'} (support=0.30, confidence=0.86)
{'Kids Bedding'} → {'Bedding Collections'} (support=0.30, confidence=0.50)
{'Bed Skirts'} → {'Shams'} (support=0.45, confidence=0.82)
{'Shams'} → {'Bed Skirts'} (support=0.45, confidence=0.82)
{'Kids Bedding'} → {'Sheets'} (support=0.50, confidence=0.83)
{'Sheets'} → {'Kids Bedding'} (support=0.50, confidence=1.00)
{'Bedspreads'} → {'Kids Bedding'} (support=0.35, confidence=1.00)
{'Kids Bedding'} → {'Bedspreads'} (support=0.35, confidence=0.58)
{'Shams'} → {'Sheets'} (support=0.35, confidence=0.64)
{'Sheets'} → {'Shams'} (support=0.35, confidence=0.70)
{'Bedspreads'} → {'Sheets'} (support=0.35, confidence=1.00)
{'Sheets'} → {'Bedspreads'} (support=0.35, confidence=0.70)
{'Bed Skirts'} → {'Kids Bedding'} (support=0.50, confidence=0.91)
{'Kids Bedding'} → {'Bed Skirts'} (support=0.50, confidence=0.83)
{'Bed Skirts'} → {'Sheets'} (support=0.45, confidence=0.82)
{'Sheets'} → {'Bed Skirts'} (support=0.45, confidence=0.90)
{'Decorative Pillows'} → {'Quilts'} (support=0.30, confidence=0.60)
{'Quilts'} → {'Decorative Pillows'} (support=0.30, confidence=0.75)
{'Bed Skirts'} → {'Bedspreads', 'Kids Bedding'} (support=0.35, confidence=0.64)
{'Bedspreads'} → {'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=1.00)
{'Kids Bedding'} → {'Bedspreads', 'Bed Skirts'} (support=0.35, confidence=0.58)
{'Bedspreads', 'Bed Skirts'} → {'Kids Bedding'} (support=0.35, confidence=1.00)
{'Bed Skirts', 'Kids Bedding'} → {'Bedspreads'} (support=0.35, confidence=0.70)
{'Bedspreads', 'Kids Bedding'} → {'Bed Skirts'} (support=0.35, confidence=1.00)
{'Bed Skirts'} → {'Shams', 'Sheets'} (support=0.35, confidence=0.64)
{'Shams'} → {'Sheets', 'Bed Skirts'} (support=0.35, confidence=0.64)
{'Sheets'} → {'Shams', 'Bed Skirts'} (support=0.35, confidence=0.70)
{'Shams', 'Bed Skirts'} → {'Sheets'} (support=0.35, confidence=0.78)
{'Sheets', 'Bed Skirts'} → {'Shams'} (support=0.35, confidence=0.78)
{'Shams', 'Sheets'} → {'Bed Skirts'} (support=0.35, confidence=1.00)
{'Bed Skirts'} → {'Bedspreads', 'Sheets'} (support=0.35, confidence=0.64)
{'Bedspreads'} → {'Sheets', 'Bed Skirts'} (support=0.35, confidence=1.00)
{'Bed Skirts', 'Kids Bedding'} → {'Sheets'} (support=0.45, confidence=0.90)
{'Sheets', 'Bed Skirts'} → {'Kids Bedding'} (support=0.45, confidence=1.00)
{'Sheets', 'Kids Bedding'} → {'Bed Skirts'} (support=0.45, confidence=0.90)
{'Bed Skirts'} → {'Shams', 'Kids Bedding', 'Sheets'} (support=0.35, confidence=0.64)
{'Kids Bedding'} → {'Shams', 'Bed Skirts', 'Sheets'} (support=0.35, confidence=0.58)
{'Shams'} → {'Sheets', 'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=0.64)
{'Sheets'} → {'Shams', 'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=0.70)
{'Bed Skirts', 'Kids Bedding'} → {'Shams', 'Sheets'} (support=0.35, confidence=0.70)
{'Shams', 'Bed Skirts'} → {'Sheets', 'Kids Bedding'} (support=0.35, confidence=0.78)
{'Sheets', 'Bed Skirts'} → {'Shams', 'Kids Bedding'} (support=0.35, confidence=0.78)
{'Kids Bedding'} → {'Sheets', 'Bed Skirts'} (support=0.35, confidence=0.78)
{'Sheets', 'Kids Bedding'} → {'Shams', 'Bed Skirts'} (support=0.35, confidence=0.70)
{'Shams', 'Sheets'} → {'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=1.00)
{'Shams', 'Bed Skirts', 'Kids Bedding'} → {'Sheets'} (support=0.35, confidence=0.87)
{'Sheets', 'Bed Skirts', 'Kids Bedding'} → {'Shams'} (support=0.35, confidence=0.78)
{'Shams', 'Bed Skirts', 'Sheets'} → {'Kids Bedding'} (support=0.35, confidence=1.00)
{'Shams', 'Kids Bedding', 'Sheets'} → {'Bed Skirts'} (support=0.35, confidence=1.00)
{'Bed Skirts'} → {'Bedspreads', 'Kids Bedding', 'Sheets'} (support=0.35, confidence=0.64)
{'Bedspreads'} → {'Sheets', 'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=1.00)
{'Kids Bedding'} → {'Bedspreads', 'Bed Skirts', 'Sheets'} (support=0.35, confidence=0.58)
{'Sheets'} → {'Bedspreads', 'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=1.00)
{'Bedspreads', 'Bed Skirts'} → {'Sheets', 'Kids Bedding'} (support=0.35, confidence=0.70)
{'Bed Skirts', 'Kids Bedding'} → {'Bedspreads', 'Sheets'} (support=0.35, confidence=0.70)
{'Sheets', 'Bed Skirts'} → {'Bedspreads', 'Kids Bedding'} (support=0.35, confidence=0.78)
{'Bedspreads', 'Kids Bedding'} → {'Sheets', 'Bed Skirts'} (support=0.35, confidence=1.00)
{'Bedspreads', 'Sheets'} → {'Bed Skirts', 'Kids Bedding'} (support=0.35, confidence=1.00)
{'Sheets', 'Kids Bedding'} → {'Bedspreads', 'Bed Skirts'} (support=0.35, confidence=0.70)
{'Bedspreads', 'Bed Skirts', 'Kids Bedding'} → {'Sheets'} (support=0.35, confidence=1.00)
{'Bedspreads', 'Bed Skirts', 'Sheets'} → {'Kids Bedding'} (support=0.35, confidence=1.00)
{'Sheets', 'Bed Skirts', 'Kids Bedding'} → {'Bedspreads'} (support=0.35, confidence=0.78)
{'Bedspreads', 'Kids Bedding', 'Sheets'} → {'Bed Skirts'} (support=0.35, confidence=1.00)

=== Brute Force Summary ===
Dataset used: Kmart.csv
Total transactions: 20
Frequent itemsets found: 29
Association rules found: 92
Execution time: 0.503597 s

```


4. Apriori

The Apriori algorithm is a classic method for mining frequent itemsets and generating association rules. It works by iteratively expanding frequent itemsets, using the principle that any subset of a frequent itemset must also be frequent.

4.1 Method:

- Generate all 1-itemsets and calculate their support.
- Filter out itemsets that do not meet the minimum support threshold.
- Use the remaining frequent itemsets to generate larger (k-item) candidate itemsets.
- Repeat the process until no more frequent itemsets can be formed.
- From the frequent itemsets, generate association rules that satisfy the minimum confidence threshold.

4.2 Example Run (each dataset):

a) Dataset: Amazon.csv

Parameters: Support = 0.3, Confidence = 0.6

```
Running Apriori Algorithm...
```

```
=== Frequent Itemsets (Apriori) ===
{A Beginner's Guide} (support=0.55)
{Android Programming: The Big Nerd Ranch} (support=0.65)
{Beginning Programming with Java} (support=0.30)
{Head First Java 2nd Edition} (support=0.40)
{Java For Dummies} (support=0.65)
{Java: The Complete Reference} (support=0.50)
{A Beginner's Guide, Android Programming: The Big Nerd Ranch} (support=0.30)
{A Beginner's Guide, Java For Dummies} (support=0.45)
{A Beginner's Guide, Java: The Complete Reference} (support=0.45)
{Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition} (support=0.30)
{Android Programming: The Big Nerd Ranch, Java For Dummies} (support=0.45)
{Android Programming: The Big Nerd Ranch, Java: The Complete Reference} (support=0.30)
{Java For Dummies, Java: The Complete Reference} (support=0.50)
{A Beginner's Guide, Java For Dummies, Java: The Complete Reference} (support=0.45)
{Android Programming: The Big Nerd Ranch, Java For Dummies, Java: The Complete Reference} (support=0.30)
```

```
=== Association Rules (Brute Force) ===
{'Head First Java 2nd Edition'} → {'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.75)
{'Java For Dummies'} → {'Java: The Complete Reference'} (support=0.50, confidence=0.77)
{'Java: The Complete Reference'} → {'Java For Dummies'} (support=0.50, confidence=1.00)
{'Java: The Complete Reference'} → {'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.60)
{'Android Programming: The Big Nerd Ranch'} → {'Java For Dummies'} (support=0.45, confidence=0.69)
{'Java For Dummies'} → {'Android Programming: The Big Nerd Ranch'} (support=0.45, confidence=0.69)
{'A Beginner's Guide'} → {'Java: The Complete Reference'} (support=0.45, confidence=0.82)
{'Java: The Complete Reference'} → {'A Beginner's Guide'} (support=0.45, confidence=0.90)
{'A Beginner's Guide'} → {'Java For Dummies'} (support=0.45, confidence=0.82)
{'Java For Dummies'} → {'A Beginner's Guide'} (support=0.45, confidence=0.69)
{'A Beginner's Guide'} → {'Java For Dummies', 'Java: The Complete Reference'} (support=0.45, confidence=0.82)
{'Java For Dummies'} → {'A Beginner's Guide', 'Java: The Complete Reference'} (support=0.45, confidence=0.69)
{'Java: The Complete Reference'} → {'A Beginner's Guide', 'Java For Dummies'} (support=0.45, confidence=0.90)
{'A Beginner's Guide', 'Java For Dummies'} → {'Java: The Complete Reference'} (support=0.45, confidence=1.00)
{'A Beginner's Guide', 'Java: The Complete Reference'} → {'Java For Dummies'} (support=0.45, confidence=1.00)
{'Java For Dummies', 'Java: The Complete Reference'} → {'A Beginner's Guide'} (support=0.45, confidence=0.90)
{'Java: The Complete Reference'} → {'Java For Dummies', 'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.60)
{'Java For Dummies', 'Android Programming: The Big Nerd Ranch'} → {'Java: The Complete Reference'} (support=0.30, confidence=0.67)
{'Android Programming: The Big Nerd Ranch', 'Java: The Complete Reference'} → {'Java For Dummies'} (support=0.30, confidence=1.00)
{'Java For Dummies', 'Java: The Complete Reference'} → {'Android Programming: The Big Nerd Ranch'} (support=0.30, confidence=0.60)
```

```
=== Brute Force Summary ===
Dataset used: Amazon.csv
Total transactions: 20
Frequent itemsets found: 15
Association rules found: 20
Execution time: 0.001938 s
```

- b) Dataset: Nike.csv
Parameters: Support = 0.5, Confidence = 0.6

```
Running Apriori Algorithm...

=== Frequent Itemsets (Apriori) ===
{Modern Pants} (support=0.50)
{Rash Guard} (support=0.60)
{Running Shoe} (support=0.70)
{Socks} (support=0.65)
{Sweatshirts} (support=0.65)
{Swimming Shirt} (support=0.55)
{Modern Pants, Sweatshirts} (support=0.50)
{Rash Guard, Swimming Shirt} (support=0.50)
{Running Shoe, Socks} (support=0.55)
{Running Shoe, Sweatshirts} (support=0.55)
{Socks, Sweatshirts} (support=0.60)
{Running Shoe, Socks, Sweatshirts} (support=0.50)

=== Association Rules (Apriori) ===
Sweatshirts → Modern Pants (support=0.50, confidence=0.77)
Modern Pants → Sweatshirts (support=0.50, confidence=1.00)
Swimming Shirt → Rash Guard (support=0.50, confidence=0.91)
Rash Guard → Swimming Shirt (support=0.50, confidence=0.83)
Running Shoe → Socks (support=0.55, confidence=0.79)
Socks → Running Shoe (support=0.55, confidence=0.85)
Sweatshirts → Running Shoe (support=0.55, confidence=0.85)
Running Shoe → Sweatshirts (support=0.55, confidence=0.79)
Sweatshirts → Socks (support=0.60, confidence=0.92)
Socks → Sweatshirts (support=0.60, confidence=0.92)
Running Shoe, Sweatshirts → Socks (support=0.50, confidence=0.91)
Socks, Sweatshirts → Running Shoe (support=0.50, confidence=0.83)
Running Shoe, Socks → Sweatshirts (support=0.50, confidence=0.91)
Sweatshirts → Running Shoe, Socks (support=0.50, confidence=0.77)
Running Shoe → Socks, Sweatshirts (support=0.50, confidence=0.71)
Socks → Running Shoe, Sweatshirts (support=0.50, confidence=0.77)

=== Apriori Summary ===
Dataset used: Nike.csv
Total transactions: 20
Frequent itemsets found: 12
Association rules found: 16
Execution time: 0.018947 s
```

- c) Dataset: Generic.csv
Parameters: Support = 0.3, Confidence = 0.7

```
Running Apriori Algorithm...

=== Frequent Itemsets (Apriori) ===
{A} (support=1.00)
{B} (support=0.50)
{C} (support=0.60)
{D} (support=0.50)
{E} (support=0.70)
{F} (support=0.45)
{A, B} (support=0.50)
{A, C} (support=0.60)
{A, D} (support=0.50)
{A, E} (support=0.70)
{A, F} (support=0.45)
{B, C} (support=0.35)
{B, D} (support=0.30)
{C, E} (support=0.30)
{D, E} (support=0.35)
{D, F} (support=0.30)
{E, F} (support=0.30)
{A, B, C} (support=0.35)
{A, B, D} (support=0.30)
{A, C, E} (support=0.30)
{A, D, E} (support=0.35)
{A, D, F} (support=0.30)
{A, E, F} (support=0.30)
```

```

=== Association Rules (Apriori) ===
B → A (support=0.50, confidence=1.00)
C → A (support=0.60, confidence=1.00)
D → A (support=0.50, confidence=1.00)
E → A (support=0.70, confidence=1.00)
A → E (support=0.70, confidence=0.70)
F → A (support=0.45, confidence=1.00)
B → C (support=0.35, confidence=0.70)
D → E (support=0.35, confidence=0.70)
B, C → A (support=0.35, confidence=1.00)
A, B → C (support=0.35, confidence=0.70)
B → A, C (support=0.35, confidence=0.70)
B, D → A (support=0.30, confidence=1.00)
C, E → A (support=0.30, confidence=1.00)
D, E → A (support=0.35, confidence=1.00)
A, D → E (support=0.35, confidence=0.70)
D → A, E (support=0.35, confidence=0.70)
D, F → A (support=0.30, confidence=1.00)
E, F → A (support=0.30, confidence=1.00)

```

```

=== Apriori Summary ===
Dataset used: Generic.csv
Total transactions: 20
Frequent itemsets found: 23
Association rules found: 18
Execution time: 0.005853 s

```

- d) Dataset: BestBuy.csv
Parameters: Support = 0.4, Confidence = 0.8

Running Apriori Algorithm...

```

=== Frequent Itemsets (Apriori) ===
{Anti-Virus} (support=0.70)
{Digital Camera} (support=0.45)
{External Hard-Drive} (support=0.45)
{Flash Drive} (support=0.65)
{Lap Top} (support=0.60)
{Lap Top Case} (support=0.70)
{Microsoft Office} (support=0.55)
{Printer} (support=0.50)
{Speakers} (support=0.55)
{Anti-Virus, External Hard-Drive} (support=0.45)
{Anti-Virus, Flash Drive} (support=0.50)
{Anti-Virus, Lap Top} (support=0.50)
{Anti-Virus, Lap Top Case} (support=0.60)
{Anti-Virus, Microsoft Office} (support=0.40)
{Anti-Virus, Speakers} (support=0.45)
{External Hard-Drive, Lap Top Case} (support=0.40)
{Flash Drive, Lap Top Case} (support=0.45)
{Flash Drive, Microsoft Office} (support=0.55)
{Flash Drive, Printer} (support=0.50)
{Lap Top, Lap Top Case} (support=0.50)
{Lap Top Case, Speakers} (support=0.45)
{Microsoft Office, Printer} (support=0.45)
{Anti-Virus, External Hard-Drive, Lap Top Case} (support=0.40)
{Anti-Virus, Flash Drive, Lap Top Case} (support=0.45)
{Anti-Virus, Flash Drive, Microsoft Office} (support=0.40)
{Anti-Virus, Lap Top, Lap Top Case} (support=0.45)
{Anti-Virus, Lap Top Case, Speakers} (support=0.40)
{Flash Drive, Microsoft Office, Printer} (support=0.45)

```



```

=== Association Rules (Apriori) ===
External Hard-Drive → Anti-Virus (support=0.45, confidence=1.00)
Lap Top → Anti-Virus (support=0.50, confidence=0.83)
Anti-Virus → Lap Top Case (support=0.60, confidence=0.86)
Lap Top Case → Anti-Virus (support=0.60, confidence=0.86)
Speakers → Anti-Virus (support=0.45, confidence=0.82)
External Hard-Drive → Lap Top Case (support=0.40, confidence=0.89)
Microsoft Office → Flash Drive (support=0.55, confidence=1.00)
Flash Drive → Microsoft Office (support=0.55, confidence=0.85)
Printer → Flash Drive (support=0.50, confidence=1.00)
Lap Top → Lap Top Case (support=0.50, confidence=0.83)
Speakers → Lap Top Case (support=0.45, confidence=0.82)
Microsoft Office → Printer (support=0.45, confidence=0.82)
Printer → Microsoft Office (support=0.45, confidence=0.90)
Anti-Virus, External Hard-Drive → Lap Top Case (support=0.40, confidence=0.89)
External Hard-Drive, Lap Top Case → Anti-Virus (support=0.40, confidence=1.00)
External Hard-Drive → Anti-Virus, Lap Top Case (support=0.40, confidence=0.89)
Anti-Virus, Flash Drive → Lap Top Case (support=0.45, confidence=0.90)
Flash Drive, Lap Top Case → Anti-Virus (support=0.45, confidence=1.00)
Anti-Virus, Microsoft Office → Flash Drive (support=0.40, confidence=1.00)
Anti-Virus, Flash Drive → Microsoft Office (support=0.40, confidence=0.80)
Lap Top, Lap Top Case → Anti-Virus (support=0.45, confidence=0.90)
Anti-Virus, Lap Top → Lap Top Case (support=0.45, confidence=0.90)
Anti-Virus, Speakers → Lap Top Case (support=0.40, confidence=0.89)
Lap Top Case, Speakers → Anti-Virus (support=0.40, confidence=0.89)
Microsoft Office, Printer → Flash Drive (support=0.45, confidence=1.00)
Flash Drive, Microsoft Office → Printer (support=0.45, confidence=0.82)
Flash Drive, Printer → Microsoft Office (support=0.45, confidence=0.90)
Microsoft Office → Flash Drive, Printer (support=0.45, confidence=0.82)
Printer → Flash Drive, Microsoft Office (support=0.45, confidence=0.90)

=== Apriori Summary ===
Dataset used: BestBuy.csv
Total transactions: 20
Frequent itemsets found: 28
Association rules found: 29
Execution time: 0.031925 s

```

- e) Dataset: Kmart.csv
Parameters: Support = 0.3, Confidence = 0.5

```

Running Apriori Algorithm...

=== Frequent Itemsets (Apriori) ===
{Bed Skirts} (support=0.55)
{Bedding Collections} (support=0.35)
{Bedspreads} (support=0.35)
{Decorative Pillows} (support=0.50)
{Embroidered Bedspread} (support=0.30)
{Kids Bedding} (support=0.60)
{Quilts} (support=0.40)
{Shams} (support=0.55)
{Sheets} (support=0.50)
{Bed Skirts, Bedspreads} (support=0.35)
{Bed Skirts, Kids Bedding} (support=0.50)
{Bed Skirts, Shams} (support=0.45)
{Bed Skirts, Sheets} (support=0.45)
{Bedding Collections, Kids Bedding} (support=0.30)
{Bedspreads, Kids Bedding} (support=0.35)
{Bedspreads, Sheets} (support=0.35)
{Decorative Pillows, Quilts} (support=0.30)
{Kids Bedding, Shams} (support=0.45)
{Kids Bedding, Sheets} (support=0.50)
{Shams, Sheets} (support=0.35)
{Bed Skirts, Bedspreads, Kids Bedding} (support=0.35)
{Bed Skirts, Bedspreads, Sheets} (support=0.35)
{Bed Skirts, Kids Bedding, Shams} (support=0.40)
{Bed Skirts, Kids Bedding, Sheets} (support=0.45)
{Bed Skirts, Shams, Sheets} (support=0.35)
{Bedspreads, Kids Bedding, Sheets} (support=0.35)
{Kids Bedding, Shams, Sheets} (support=0.35)
{Bed Skirts, Bedspreads, Kids Bedding, Sheets} (support=0.35)
{Bed Skirts, Kids Bedding, Shams, Sheets} (support=0.35)

```


=== Association Rules (Apriori) ===

Bedspreads → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts → Bedspreads (support=0.35, confidence=0.64)
Bed Skirts → Kids Bedding (support=0.50, confidence=0.91)
Kids Bedding → Bed Skirts (support=0.50, confidence=0.83)
Shams → Bed Skirts (support=0.45, confidence=0.82)
Bedspreads → Shams (support=0.45, confidence=0.82)
Sheets → Bed Skirts (support=0.45, confidence=0.90)
Bed Skirts → Sheets (support=0.45, confidence=0.82)
Bedding Collections → Kids Bedding (support=0.30, confidence=0.86)
Kids Bedding → Bedding Collections (support=0.30, confidence=0.50)
Bedspreads → Kids Bedding (support=0.35, confidence=1.00)
Kids Bedding → Bedspreads (support=0.35, confidence=0.58)
Sheets → Bedspreads (support=0.35, confidence=0.70)
Bedspreads → Sheets (support=0.35, confidence=1.00)
Quilts → Decorative Pillows (support=0.30, confidence=0.75)
Decorative Pillows → Quilts (support=0.30, confidence=0.60)
Shams → Kids Bedding (support=0.45, confidence=0.82)
Kids Bedding → Shams (support=0.45, confidence=0.75)
Sheets → Kids Bedding (support=0.50, confidence=1.00)
Kids Bedding → Sheets (support=0.50, confidence=0.83)
Sheets → Shams (support=0.35, confidence=0.70)
Shams → Sheets (support=0.35, confidence=0.64)
Bed Skirts, Bedspreads → Kids Bedding (support=0.35, confidence=1.00)
Bedspreads, Kids Bedding → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding → Bedspreads (support=0.35, confidence=0.70)
Bedspreads → Bed Skirts, Kids Bedding (support=0.35, confidence=1.00)
Bed Skirts → Bedspreads, Kids Bedding (support=0.35, confidence=0.64)
Kids Bedding → Bed Skirts, Bedspreads (support=0.35, confidence=0.58)
Bed Skirts, Sheets → Bedspreads (support=0.35, confidence=0.78)
Bedspreads, Sheets → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Bedspreads → Sheets (support=0.35, confidence=1.00)
Sheets → Bed Skirts, Bedspreads (support=0.35, confidence=0.70)
Bed Skirts → Bedspreads, Sheets (support=0.35, confidence=0.64)
Bedspreads → Bed Skirts, Sheets (support=0.35, confidence=1.00)
Bed Skirts, Shams → Kids Bedding (support=0.40, confidence=0.89)
Kids Bedding, Shams → Bed Skirts (support=0.40, confidence=0.89)

Kids Bedding → Shams, Sheets (support=0.35, confidence=0.58)
Shams → Kids Bedding, Sheets (support=0.35, confidence=0.64)
Bed Skirts, Kids Bedding, Sheets → Bedspreads (support=0.35, confidence=0.78)
Bed Skirts, Bedspreads, Sheets → Kids Bedding (support=0.35, confidence=1.00)
Bedspreads, Kids Bedding, Sheets → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Bedspreads, Kids Bedding → Sheets (support=0.35, confidence=1.00)
Bed Skirts, Sheets → Bedspreads, Kids Bedding (support=0.35, confidence=0.78)
Kids Bedding, Sheets → Bed Skirts, Bedspreads (support=0.35, confidence=0.70)
Bedspreads, Sheets → Bed Skirts, Kids Bedding (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding → Bedspreads, Sheets (support=0.35, confidence=0.70)
Bed Skirts, Bedspreads → Kids Bedding, Sheets (support=0.35, confidence=1.00)
Bedspreads, Kids Bedding → Bed Skirts, Sheets (support=0.35, confidence=1.00)
Sheets → Bed Skirts, Bedspreads, Kids Bedding (support=0.35, confidence=0.70)
Bed Skirts → Bedspreads, Kids Bedding, Sheets (support=0.35, confidence=0.64)
Kids Bedding → Bed Skirts, Bedspreads, Sheets (support=0.35, confidence=0.58)
Bedspreads → Bed Skirts, Kids Bedding, Sheets (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding, Sheets → Shams (support=0.35, confidence=0.78)
Bed Skirts, Shams, Sheets → Kids Bedding (support=0.35, confidence=1.00)
Kids Bedding, Shams, Sheets → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding, Shams → Sheets (support=0.35, confidence=0.87)
Bed Skirts, Sheets → Kids Bedding, Shams (support=0.35, confidence=0.78)
Kids Bedding, Sheets → Bed Skirts, Shams (support=0.35, confidence=0.70)
Shams, Sheets → Bed Skirts, Kids Bedding (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding → Shams, Sheets (support=0.35, confidence=0.70)
Bed Skirts, Shams → Kids Bedding, Sheets (support=0.35, confidence=0.78)
Kids Bedding, Shams → Bed Skirts, Sheets (support=0.35, confidence=0.78)
Sheets → Bed Skirts, Kids Bedding, Shams (support=0.35, confidence=0.70)
Bed Skirts → Kids Bedding, Shams, Sheets (support=0.35, confidence=0.64)
Kids Bedding → Bed Skirts, Shams, Sheets (support=0.35, confidence=0.58)
Shams → Bed Skirts, Kids Bedding, Sheets (support=0.35, confidence=0.64)

=== Apriori Summary ===

Dataset used: Kmart.csv
Total transactions: 20
Frequent itemsets found: 29
Association rules found: 92
Execution time: 0.004182 s

5. FP-Growth

The FP-Growth (Frequent Pattern Growth) algorithm is an efficient method for finding frequent itemsets without generating candidate sets. It uses a compact data structure called an FP-tree to store item frequencies and relationships, reducing the need for repeated database scans.

5.1 Method:

- Scan the dataset to find the frequency of each item and filter out infrequent ones.
- Build an FP-tree where each path represents item combinations in transactions.
- For each item, extract conditional patterns and construct conditional FP-trees.
- Recursively mine these trees to find all frequent itemsets.
- Generate association rules based on the discovered frequent itemsets.

5.2 Example Run (each dataset):

a) Dataset: Amazon.csv

Parameters: Support = 0.3, Confidence = 0.6

```
Running FP-Growth Algorithm...
```

```
=== Frequent Itemsets (FP-Growth) ===
```

```
{Java For Dummies} (support=0.65)
{Android Programming: The Big Nerd Ranch} (support=0.65)
{A Beginner's Guide} (support=0.55)
{Java: The Complete Reference} (support=0.50)
{Head First Java 2nd Edition} (support=0.40)
{Beginning Programming with Java} (support=0.30)
{Android Programming: The Big Nerd Ranch, Java For Dummies} (support=0.45)
{A Beginner's Guide, Java For Dummies} (support=0.45)
{A Beginner's Guide, Android Programming: The Big Nerd Ranch} (support=0.30)
{Java For Dummies, Java: The Complete Reference} (support=0.50)
{A Beginner's Guide, Java: The Complete Reference} (support=0.45)
{Android Programming: The Big Nerd Ranch, Java: The Complete Reference} (support=0.30)
{A Beginner's Guide, Java For Dummies, Java: The Complete Reference} (support=0.45)
{Android Programming: The Big Nerd Ranch, Java For Dummies, Java: The Complete Reference} (support=0.30)
{Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition} (support=0.30)
```

```
=== Association Rules (FP-Growth) ===
```

```
Android Programming: The Big Nerd Ranch → Java For Dummies (support=0.45, confidence=0.69)
Java For Dummies → Android Programming: The Big Nerd Ranch (support=0.45, confidence=0.69)
A Beginner's Guide → Java For Dummies (support=0.45, confidence=0.82)
Java For Dummies → A Beginner's Guide (support=0.45, confidence=0.69)
Java For Dummies → Java: The Complete Reference (support=0.50, confidence=0.77)
Java: The Complete Reference → Java For Dummies (support=0.50, confidence=1.00)
A Beginner's Guide → Java: The Complete Reference (support=0.45, confidence=0.82)
Java: The Complete Reference → A Beginner's Guide (support=0.45, confidence=0.90)
Java: The Complete Reference → Android Programming: The Big Nerd Ranch (support=0.30, confidence=0.60)
A Beginner's Guide, Java For Dummies → Java: The Complete Reference (support=0.45, confidence=1.00)
A Beginner's Guide, Java: The Complete Reference → Java For Dummies (support=0.45, confidence=1.00)
Java For Dummies, Java: The Complete Reference → A Beginner's Guide (support=0.45, confidence=0.90)
A Beginner's Guide → Java For Dummies, Java: The Complete Reference (support=0.45, confidence=0.82)
Java For Dummies → A Beginner's Guide, Java: The Complete Reference (support=0.45, confidence=0.69)
Java: The Complete Reference → A Beginner's Guide, Java For Dummies (support=0.45, confidence=0.90)
Android Programming: The Big Nerd Ranch, Java For Dummies → Java: The Complete Reference (support=0.30, confidence=0.67)
Android Programming: The Big Nerd Ranch, Java: The Complete Reference → Java For Dummies (support=0.30, confidence=1.00)
Java For Dummies, Java: The Complete Reference → Android Programming: The Big Nerd Ranch (support=0.30, confidence=0.60)
Java: The Complete Reference → Android Programming: The Big Nerd Ranch, Java For Dummies (support=0.30, confidence=0.60)
Head First Java 2nd Edition → Android Programming: The Big Nerd Ranch (support=0.30, confidence=0.75)
```

```
=== FP-Growth Summary ===
```

```
Dataset used: Amazon.csv
Total transactions: 20
Frequent itemsets found: 15
Association rules found: 20
Execution time: 0.003572 s
```

- b) Dataset: Nike.csv
Parameters: Support = 0.5, Confidence = 0.6

```
Running FP-Growth Algorithm...
```

```
=== Frequent Itemsets (FP-Growth) ===
```

```
{Running Shoe} (support=0.70)
{Sweatshirts} (support=0.65)
{Socks} (support=0.65)
{Modern Pants} (support=0.50)
{Rash Guard} (support=0.60)
{Swimming Shirt} (support=0.55)
{Running Shoe, Sweatshirts} (support=0.55)
{Socks, Sweatshirts} (support=0.60)
{Running Shoe, Socks} (support=0.55)
{Running Shoe, Socks, Sweatshirts} (support=0.50)
{Modern Pants, Sweatshirts} (support=0.50)
{Rash Guard, Swimming Shirt} (support=0.50)
```

```
=== Association Rules (FP-Growth) ===
```

```
Sweatshirts → Running Shoe (support=0.55, confidence=0.85)
Running Shoe → Sweatshirts (support=0.55, confidence=0.79)
Sweatshirts → Socks (support=0.60, confidence=0.92)
Socks → Sweatshirts (support=0.60, confidence=0.92)
Running Shoe → Socks (support=0.55, confidence=0.79)
Socks → Running Shoe (support=0.55, confidence=0.85)
Running Shoe, Sweatshirts → Socks (support=0.50, confidence=0.91)
Socks, Sweatshirts → Running Shoe (support=0.50, confidence=0.83)
Running Shoe, Socks → Sweatshirts (support=0.50, confidence=0.91)
Sweatshirts → Running Shoe, Socks (support=0.50, confidence=0.77)
Running Shoe → Socks, Sweatshirts (support=0.50, confidence=0.71)
Socks → Running Shoe, Sweatshirts (support=0.50, confidence=0.77)
Sweatshirts → Modern Pants (support=0.50, confidence=0.77)
Modern Pants → Sweatshirts (support=0.50, confidence=1.00)
Swimming Shirt → Rash Guard (support=0.50, confidence=0.91)
Rash Guard → Swimming Shirt (support=0.50, confidence=0.83)
```

```
=== FP-Growth Summary ===
```

```
Dataset used: Nike.csv
Total transactions: 20
Frequent itemsets found: 12
Association rules found: 16
Execution time: 0.003598 s
```

- c) Dataset: Generic.csv
Parameters: Support = 0.3, Confidence = 0.7

```
Running FP-Growth Algorithm...
```

```
=== Frequent Itemsets (FP-Growth) ===
```

```
{A} (support=1.00)
{C} (support=0.60)
{B} (support=0.50)
{D} (support=0.50)
{E} (support=0.70)
{F} (support=0.45)
{A, C} (support=0.60)
{C, E} (support=0.30)
{A, C, E} (support=0.30)
{A, B} (support=0.50)
{B, C} (support=0.35)
{B, D} (support=0.30)
{A, B, C} (support=0.35)
{A, B, D} (support=0.30)
{A, D} (support=0.50)
{D, E} (support=0.35)
{A, D, E} (support=0.35)
{A, E} (support=0.70)
{A, F} (support=0.45)
{D, F} (support=0.30)
{E, F} (support=0.30)
{A, D, F} (support=0.30)
{A, E, F} (support=0.30)
```

```
=== Association Rules (FP-Growth) ===
```

```
C → A (support=0.60, confidence=1.00)
C, E → A (support=0.30, confidence=1.00)
B → A (support=0.50, confidence=1.00)
B → C (support=0.35, confidence=0.70)
B, C → A (support=0.35, confidence=1.00)
A, B → C (support=0.35, confidence=0.70)
B → A, C (support=0.35, confidence=0.70)
B, D → A (support=0.30, confidence=1.00)
D → A (support=0.50, confidence=1.00)
D → E (support=0.35, confidence=0.70)
D, E → A (support=0.35, confidence=1.00)
A, D → E (support=0.35, confidence=0.70)
D → A, E (support=0.35, confidence=0.70)
E → A (support=0.70, confidence=1.00)
A → E (support=0.70, confidence=0.70)
F → A (support=0.45, confidence=1.00)
D, F → A (support=0.30, confidence=1.00)
E, F → A (support=0.30, confidence=1.00)
```

```
=== FP-Growth Summary ===
```

```
Dataset used: Generic.csv
Total transactions: 20
Frequent itemsets found: 23
Association rules found: 18
Execution time: 0.007586 s
```


- d) Dataset: BestBuy.csv
Parameters: Support = 0.4, Confidence = 0.8

Running FP-Growth Algorithm...

```
=== Frequent Itemsets (FP-Growth) ===
{Anti-Virus} (support=0.70)
{Flash Drive} (support=0.65)
{Speakers} (support=0.55)
{Microsoft Office} (support=0.55)
{Printer} (support=0.50)
{Lap Top Case} (support=0.70)
{Lap Top} (support=0.60)
{External Hard-Drive} (support=0.45)
{Digital Camera} (support=0.45)
{Anti-Virus, Lap Top Case} (support=0.60)
{Anti-Virus, Flash Drive} (support=0.50)
{Flash Drive, Lap Top Case} (support=0.45)
{Anti-Virus, Flash Drive, Lap Top Case} (support=0.45)
{Anti-Virus, Speakers} (support=0.45)
{Lap Top Case, Speakers} (support=0.45)
{Anti-Virus, Lap Top Case, Speakers} (support=0.40)
{Flash Drive, Microsoft Office} (support=0.55)
{Anti-Virus, Microsoft Office} (support=0.40)
{Anti-Virus, Flash Drive, Microsoft Office} (support=0.40)
{Flash Drive, Printer} (support=0.50)
{Microsoft Office, Printer} (support=0.45)
{Flash Drive, Microsoft Office, Printer} (support=0.45)
{Anti-Virus, Lap Top} (support=0.50)
{Lap Top, Lap Top Case} (support=0.50)
{Anti-Virus, Lap Top, Lap Top Case} (support=0.45)
{Anti-Virus, External Hard-Drive} (support=0.45)
{External Hard-Drive, Lap Top Case} (support=0.40)
{Anti-Virus, External Hard-Drive, Lap Top Case} (support=0.40)
```

```
=== Association Rules (FP-Growth) ===
Anti-Virus → Lap Top Case (support=0.60, confidence=0.86)
Lap Top Case → Anti-Virus (support=0.60, confidence=0.86)
Anti-Virus, Flash Drive → Lap Top Case (support=0.45, confidence=0.90)
Flash Drive, Lap Top Case → Anti-Virus (support=0.45, confidence=1.00)
Speakers → Anti-Virus (support=0.45, confidence=0.82)
Speakers → Lap Top Case (support=0.45, confidence=0.82)
Anti-Virus, Speakers → Lap Top Case (support=0.40, confidence=0.89)
Lap Top Case, Speakers → Anti-Virus (support=0.40, confidence=0.89)
Microsoft Office → Flash Drive (support=0.55, confidence=1.00)
Flash Drive → Microsoft Office (support=0.55, confidence=0.85)
Anti-Virus, Microsoft Office → Flash Drive (support=0.40, confidence=1.00)
Anti-Virus, Flash Drive → Microsoft Office (support=0.40, confidence=0.80)
Printer → Flash Drive (support=0.50, confidence=1.00)
Microsoft Office → Printer (support=0.45, confidence=0.82)
Printer → Microsoft Office (support=0.45, confidence=0.90)
Microsoft Office, Printer → Flash Drive (support=0.45, confidence=1.00)
Flash Drive, Microsoft Office → Printer (support=0.45, confidence=0.82)
Flash Drive, Printer → Microsoft Office (support=0.45, confidence=0.90)
Microsoft Office → Flash Drive, Printer (support=0.45, confidence=0.82)
Printer → Flash Drive, Microsoft Office (support=0.45, confidence=0.90)
Lap Top → Anti-Virus (support=0.50, confidence=0.83)
Lap Top → Lap Top Case (support=0.50, confidence=0.83)
Lap Top, Lap Top Case → Anti-Virus (support=0.45, confidence=0.90)
Anti-Virus, Lap Top → Lap Top Case (support=0.45, confidence=0.90)
External Hard-Drive → Anti-Virus (support=0.45, confidence=1.00)
External Hard-Drive → Lap Top Case (support=0.40, confidence=0.89)
Anti-Virus, External Hard-Drive → Lap Top Case (support=0.40, confidence=0.89)
External Hard-Drive, Lap Top Case → Anti-Virus (support=0.40, confidence=1.00)
External Hard-Drive → Anti-Virus, Lap Top Case (support=0.40, confidence=0.89)
```

```
=== FP-Growth Summary ===
Dataset used: BestBuy.csv
Total transactions: 20
Frequent itemsets found: 28
Association rules found: 29
Execution time: 0.005280 s
```

e) Dataset: Kmart.csv

Parameters: Support = 0.3, Confidence = 0.5

Running FP-Growth Algorithm...

```
=== Frequent Itemsets (FP-Growth) ===
{Decorative Pillows} (support=0.50)
{Quilts} (support=0.40)
{Embroidered Bedspread} (support=0.30)
{Kids Bedding} (support=0.60)
{Shams} (support=0.55)
{Bed Skirts} (support=0.55)
{Sheets} (support=0.50)
{Bedspreads} (support=0.35)
{Bedding Collections} (support=0.35)
{Decorative Pillows, Quilts} (support=0.30)
{Kids Bedding, Shams} (support=0.45)
{Bed Skirts, Kids Bedding} (support=0.50)
{Bed Skirts, Shams} (support=0.45)
{Bed Skirts, Kids Bedding, Shams} (support=0.40)
{Kids Bedding, Sheets} (support=0.50)
{Bed Skirts, Sheets} (support=0.45)
{Shams, Sheets} (support=0.35)
{Bed Skirts, Kids Bedding, Sheets} (support=0.45)
{Kids Bedding, Shams, Sheets} (support=0.35)
{Bed Skirts, Shams, Sheets} (support=0.35)
{Bed Skirts, Kids Bedding, Shams, Sheets} (support=0.35)
{Bedspreads, Sheets} (support=0.35)
{Bed Skirts, Bedspreads} (support=0.35)
{Bedspreads, Kids Bedding} (support=0.35)
{Bed Skirts, Bedspreads, Sheets} (support=0.35)
{Bedspreads, Kids Bedding, Sheets} (support=0.35)
{Bed Skirts, Bedspreads, Kids Bedding} (support=0.35)
{Bed Skirts, Bedspreads, Kids Bedding, Sheets} (support=0.35)
{Bedding Collections, Kids Bedding} (support=0.30)
```

```
=== Association Rules (FP-Growth) ===
Quilts → Decorative Pillows (support=0.30, confidence=0.75)
Decorative Pillows → Quilts (support=0.30, confidence=0.60)
Shams → Kids Bedding (support=0.45, confidence=0.82)
Kids Bedding → Shams (support=0.45, confidence=0.75)
Bed Skirts → Kids Bedding (support=0.50, confidence=0.91)
Kids Bedding → Bed Skirts (support=0.50, confidence=0.83)
Shams → Bed Skirts (support=0.45, confidence=0.82)
Bed Skirts → Shams (support=0.45, confidence=0.82)
Bed Skirts, Shams → Kids Bedding (support=0.40, confidence=0.89)
Kids Bedding, Shams → Bed Skirts (support=0.40, confidence=0.89)
Bed Skirts, Kids Bedding → Shams (support=0.40, confidence=0.80)
Shams → Bed Skirts, Kids Bedding (support=0.40, confidence=0.73)
Bed Skirts → Kids Bedding, Shams (support=0.40, confidence=0.73)
Kids Bedding → Bed Skirts, Shams (support=0.40, confidence=0.67)
Sheets → Kids Bedding (support=0.50, confidence=1.00)
Kids Bedding → Sheets (support=0.50, confidence=0.83)
Sheets → Bed Skirts (support=0.45, confidence=0.90)
Bed Skirts → Sheets (support=0.45, confidence=0.82)
Sheets → Shams (support=0.35, confidence=0.70)
Shams → Sheets (support=0.35, confidence=0.64)
Bed Skirts, Sheets → Kids Bedding (support=0.45, confidence=1.00)
Kids Bedding, Sheets → Bed Skirts (support=0.45, confidence=0.90)
Bed Skirts, Kids Bedding → Sheets (support=0.45, confidence=0.90)
Sheets → Bed Skirts, Kids Bedding (support=0.45, confidence=0.90)
Bed Skirts → Kids Bedding, Sheets (support=0.45, confidence=0.82)
Kids Bedding → Bed Skirts, Sheets (support=0.45, confidence=0.75)
Kids Bedding, Sheets → Shams (support=0.35, confidence=0.70)
Shams, Sheets → Kids Bedding (support=0.35, confidence=1.00)
Kids Bedding, Shams → Sheets (support=0.35, confidence=0.78)
Sheets → Kids Bedding, Shams (support=0.35, confidence=0.70)
Kids Bedding → Shams, Sheets (support=0.35, confidence=0.58)
Shams → Kids Bedding, Sheets (support=0.35, confidence=0.64)
Bed Skirts, Sheets → Shams (support=0.35, confidence=0.78)
Shams, Sheets → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Shams → Sheets (support=0.35, confidence=0.78)
```

```

Bedspreads → Bed Skirts, Sheets (support=0.35, confidence=1.00)
Bedspreads, Sheets → Kids Bedding (support=0.35, confidence=1.00)
Kids Bedding, Sheets → Bedspreads (support=0.35, confidence=0.70)
Bedspreads, Kids Bedding → Sheets (support=0.35, confidence=1.00)
Sheets → Bedspreads, Kids Bedding (support=0.35, confidence=0.70)
Bedspreads → Kids Bedding, Sheets (support=0.35, confidence=1.00)
Kids Bedding → Bedspreads, Sheets (support=0.35, confidence=0.58)
Bed Skirts, Bedspreads → Kids Bedding (support=0.35, confidence=1.00)
Bedspreads, Kids Bedding → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding → Bedspreads (support=0.35, confidence=0.70)
Bedspreads → Bed Skirts, Kids Bedding (support=0.35, confidence=1.00)
Bed Skirts → Bedspreads, Kids Bedding (support=0.35, confidence=0.64)
Kids Bedding → Bed Skirts, Bedspreads (support=0.35, confidence=0.58)
Bed Skirts, Kids Bedding, Sheets → Bedspreads (support=0.35, confidence=0.78)
Bed Skirts, Bedspreads, Sheets → Kids Bedding (support=0.35, confidence=1.00)
Bedspreads, Kids Bedding, Sheets → Bed Skirts (support=0.35, confidence=1.00)
Bed Skirts, Bedspreads, Kids Bedding → Sheets (support=0.35, confidence=1.00)
Bed Skirts, Sheets → Bedspreads, Kids Bedding (support=0.35, confidence=0.78)
Kids Bedding, Sheets → Bed Skirts, Bedspreads (support=0.35, confidence=0.70)
Bedspreads, Sheets → Bed Skirts, Kids Bedding (support=0.35, confidence=1.00)
Bed Skirts, Kids Bedding → Bedspreads, Sheets (support=0.35, confidence=0.70)
Bed Skirts, Bedspreads → Kids Bedding, Sheets (support=0.35, confidence=1.00)
Bedspreads, Kids Bedding → Bed Skirts, Sheets (support=0.35, confidence=1.00)
Sheets → Bed Skirts, Bedspreads, Kids Bedding (support=0.35, confidence=0.70)
Bed Skirts → Bedspreads, Kids Bedding, Sheets (support=0.35, confidence=0.64)
Kids Bedding → Bed Skirts, Bedspreads, Sheets (support=0.35, confidence=0.58)
Bedspreads → Bed Skirts, Kids Bedding, Sheets (support=0.35, confidence=1.00)
Bedding Collections → Kids Bedding (support=0.30, confidence=0.86)
Kids Bedding → Bedding Collections (support=0.30, confidence=0.50)

```

```

=== FP-Growth Summary ===
Dataset used: Kmart.csv
Total transactions: 20
Frequent itemsets found: 29
Association rules found: 92
Execution time: 0.005019 s

```

6. Time Comparison

Dataset	Min Support	Min Confidence	Algorithm	Execution Time (sec)
Amazon	0.3	0.6	Brute Force	0.502129
			Apriori	0.010001
			FP Growth	0.011027
Nike	0.5	0.6	Brute Force	0.504632
			Apriori	0.018947
			FP Growth	0.003598
Generic	0.3	0.7	Brute Force	0.503241
			Apriori	0.005853
			FP Growth	0.007586
BestBuy	0.4	0.8	Brute Force	0.503471
			Apriori	0.031925
			FP Growth	0.005280
Kmart	0.3	0.5	Brute Force	0.503597
			Apriori	0.004182
			FP Growth	0.005019

7. How to Run the Code

a) Python Environment:

- Use Python 3.10 to 3.12 (recommended).
- Avoid Python 3.13 as some libraries like mlxtend may not be fully compatible yet.
- Install Anaconda or Miniconda (optional but recommended) to easily manage packages and environments.
- Alternatively, you can use a virtual environment (venv) to isolate your project dependencies:
 - `python -m venv env`
 - `source env/bin/activate` (For macOS/Linux)
 - `env\Scripts\activate` (For Windows)

b) Install Required Libraries (bash/powershell/cmd):

- `pip install pandas numpy mlxtend jupyter`

c) Verify installation (python):

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, fpgrowth
print("All libraries installed successfully!")
```

d) Development Environment Options:

1. Jupyter Notebook:

- Launch Jupyter Notebook.
- Run each cell sequentially (Shift + Enter) to execute and visualize results for Brute Force, Apriori, and FP-Growth algorithms.

2. Visual Studio Code (VS Code):

- Open the project folder: `cs634_nandnipatel_midtermproject/`
- Open any script in the `src/` folder:
 - `brute_force.py`
 - `apriori_runner.py`
 - `fpgrowth_runner.py`
- Run the script using Terminal command: `python src/brute_force.py`
- Follow on-screen prompts to:
 - Select dataset (from /data)
 - Enter minimum support and confidence thresholds
 - View output of frequent itemsets, association rules, and execution time.

8. Conclusion

- Brute Force algorithm performs optimally on small data but with growing data size or item variety becomes inefficient.
- Apriori algorithm is also scalable compared to Brute Force but uses additional computation time since it takes multiple scanning of the database.
- FP-Growth is most efficient in that it does not generate candidates, especially for large data or low support values.

- With increasing support and confidence levels, the count of frequent itemsets and association rules decreases in all three algorithms as desired.
- Overall, FP-Growth works best with improved scalability, Apriori provides average performance, and Brute Force is a simple benchmark to understand frequent pattern mining.

9. GitHub Repository Structure ([Link to GitHub](#))

```
patel_nandni_midtermproject/  
|  
├─ data/  
|   ├─ Amazon.csv  
|   ├─ BestBuy.csv  
|   ├─ Generic.csv  
|   ├─ Kmart.csv  
|   └─ Nike.csv  
|  
├─ notebooks/  
|   └─ project_demo.ipynb  
|  
├─ report/  
|   └─ patel_nandni_midtermproject.pdf  
|  
├─ screenshots/  
|   ├─ Output1.png ... Output8.png  
|   └─ Time comparison.png  
|  
├─ src/  
|   ├─ apriori_runner.py  
|   ├─ brute_force.py  
|   └─ fpgrowth_runner.py  
|  
├─ README.md  
└─ requirements.txt
```

Note: GitHub is linked with my personal email (nandnip1402@gmail.com)