

**Mid-Term Project Report**

**Apriori and FP Growth Algorithms Implementation Using Python**

**NAME:** Yashwanth Reddy Boddireddy

**NJIT UCID:** yb244  
**Email Address:** yb244@njit.edu  
**Professor:** Yasser Abduallah  
CS 634 - 104 Data Mining

Contents

[Abstract: 4](#_Toc160993043)

[1. Introduction 4](#_Toc160993044)

[2. Foundational Concepts 4](#_Toc160993045)

[Frequent Itemsets: 4](#_Toc160993046)

[Association Rules: 4](#_Toc160993047)

[Support: 4](#_Toc160993048)

[Confidence: 4](#_Toc160993049)

[Lift: 4](#_Toc160993050)

[3. The Apriori Algorithm 4](#_Toc160993051)

[Implementation: 4](#_Toc160993052)

[Advantages: 4](#_Toc160993053)

[Limitations: 5](#_Toc160993054)

[Real-World Applications: 5](#_Toc160993055)

[4. The FP-Growth Algorithm 5](#_Toc160993056)

[Implementation: 5](#_Toc160993057)

[Advantages: 5](#_Toc160993058)

[Limitations: 5](#_Toc160993059)

[Real-World Applications: 5](#_Toc160993060)

[5. Comparison and Considerations 5](#_Toc160993061)

[6. Detailed Steps of Implementation for Apriori 5](#_Toc160993062)

[7.Detailed Steps of Implementation for FP Growth 6](#_Toc160993063)

[Project Workflow: 7](#_Toc160993064)

[1. Introduction 7](#_Toc160993065)

[2. Dataset Selection 7](#_Toc160993066)

[1) Amazon 7](#_Toc160993067)

[2) Best Buy 8](#_Toc160993068)

[3) K-mart 9](#_Toc160993069)

[4) Nike 10](#_Toc160993070)

[5) Generic 10](#_Toc160993071)

[3. User Inputs for Minimum Support and Confidence 12](#_Toc160993072)

[4. Brute-Forced Apriori Implementation 12](#_Toc160993073)

[5. Validation with Python Package of Apriori 15](#_Toc160993074)

[6. Validation with FP-Growth Algorithm 17](#_Toc160993075)

[7. Comparing time taken by each algorithm 18](#_Toc160993076)

[8. Results based on different minimum support and confidence on each data set. 19](#_Toc160993077)

[For Amazon Data set: 19](#_Toc160993078)

[For BestBuy Dataset: 22](#_Toc160993079)

[For Kmart Dataset: 24](#_Toc160993080)

[For Nike Dataset: 26](#_Toc160993081)

[For Generic Dataset: 28](#_Toc160993082)

[9. Performance Analysis 29](#_Toc160993083)

[Results: 29](#_Toc160993084)

[Conclusion: 29](#_Toc160993085)

[Recommendations: 29](#_Toc160993086)

[Steps to run the Program: 30](#_Toc160993087)

[STEP 1: Clone the repository. 30](#_Toc160993088)

[STEP 2: Create Virtual Environment. 30](#_Toc160993089)

[STEP 3: Install required libraries. 30](#_Toc160993090)

[STEP 4: Run the python file. 30](#_Toc160993091)

[Conclusion 31](#_Toc160993092)

[References and Links: 31](#_Toc160993093)

# Abstract:

This project undertakes a comprehensive analysis of transaction data from an online retail dataset to uncover underlying patterns and associations between customer purchases using the Apriori and FP-Growth algorithms. By applying these data mining techniques, the study aims to identify frequent item sets and generate association rules that reveal customer buying behaviors, particularly in the realm of programming and web development books. The project explores the theoretical underpinnings of association rule learning, detailing the implementation of both brute-forced and optimized algorithmic approaches to efficiently process the dataset. Through this analytical endeavor, the project demonstrates the practical applications of data mining in enhancing retail strategy, from inventory management to personalized marketing. The insights derived from this analysis are intended to inform decision-making processes, enabling retailers to align their offerings more closely with customer interests and trends.

# 1. Introduction

Data mining encompasses techniques and methodologies for uncovering patterns, associations, and correlations within large volumes of data. Among these, the Apriori and FP-Growth algorithms stand out for their efficiency in frequent itemset mining and association rule learning, pivotal for understanding consumer behaviour, enhancing product recommendations, and optimizing inventory management.

# 2. Foundational Concepts

Before delving into the algorithms, it is crucial to understand the foundational concepts underlying data mining efforts:

Frequent Itemsets: Sets of items that appear together in a dataset with frequency above a user-defined threshold.

Association Rules: Implications that identify the likelihood of itemsets occurring together, formatted as X⇒Y, where X and Y are distinct itemsets.

Support: The proportion of transactions in the dataset that include a specific itemset, indicating its frequency.

Confidence: A measure of how often items in Y appear in transactions that contain X, reflecting the reliability of the association.

Lift: Indicates the strength of a rule over the random co-occurrence of X and Y, with values greater than 1 suggesting a positive association.

# 3. The Apriori Algorithm

The Apriori algorithm identifies frequent itemsets and association rules through an iterative process, leveraging the downward closure property to reduce the search space efficiently.

Implementation: Iteratively generates candidate itemsets and eliminates those with support below the threshold.

Advantages: Simplicity and ease of implementation.

Limitations: Performance issues with large datasets due to multiple database scans and high computational overhead.

Real-World Applications: Market basket analysis, cross-marketing strategies, and catalog design.

# 4. The FP-Growth Algorithm

FP-Growth streamlines frequent itemset discovery without candidate generation, using a compact FP-tree structure to encapsulate data relationships.

Implementation: Constructs an FP-tree from the dataset, then mines frequent itemsets directly from the tree.

Advantages: Significantly faster than Apriori, especially on large datasets, and requires only two passes over the dataset.

Limitations: Complexity in understanding and implementing the FP-tree structure and potential memory constraints.

Real-World Applications: E-commerce recommendations, fraud detection in banking, and patient data analysis in healthcare.

# 5. Comparison and Considerations

While Apriori offers a foundational approach to understanding item associations, its scalability issues make FP-Growth preferable for handling voluminous and complex datasets. The choice between algorithms depends on the specific requirements of the analysis, including dataset size, itemset complexity, and computational resources.

# 6. Detailed Steps of Implementation for Apriori

1. **Collect Data**: Assemble a dataset of transactions where each transaction is a set of items purchased together.
2. **Set Minimum Support and Confidence**:
   * **Support** is the percentage of transactions that include a particular itemset.
   * **Confidence** is a measure of the likelihood that an itemset is purchased when another itemset is purchased.
3. **Generate Candidate Itemsets**:
   * Start with candidate itemsets of size 1 (C1), which are just the individual items.
4. **Determine Frequent Itemsets**:
   * Count the frequency of each candidate itemset in the dataset.
   * Compare it with the minimum support threshold.
   * Retain those that meet or exceed the threshold. These are your frequent itemsets (F1).
5. **Create Larger Itemsets**:
   * Use the frequent itemsets found in the previous step to generate new candidate itemsets of larger sizes (C2, C3, etc.).
   * This process involves joining itemsets with themselves and pruning subsets that are not frequent.
6. **Repeat Determination of Frequent Itemsets**:
   * Repeat steps 4 and 5 for larger and larger itemsets until no new frequent itemsets are found.
7. **Generate Association Rules**:
   * For every frequent itemset, generate all possible rules.
   * Calculate the confidence of each rule.
   * Keep only those rules that meet the minimum confidence threshold.

# 7.Detailed Steps of Implementation for FP Growth

1. **Collect Data**: Begin with a dataset of transactions, where each transaction is a list of items purchased together.
2. **Create the FP-Tree**:
   * **Step 1: Count item frequencies** and order items in individual transactions by descending frequency.
   * **Step 2: Build the tree**. Starting with a null root, add paths for transactions. Share nodes, when possible, which means that if two transactions have a common prefix, they share the initial portion of their paths.
3. **Mine Frequent Itemsets**:
   * For each item, starting from the least frequent, create a conditional base (a sub-dataset of transactions containing that item).
   * Build a conditional FP-Tree for this base.
   * Recursively mine this tree, appending the item to the generated suffixes, to find all frequent itemsets involving this item.
4. **Generate Association Rules**:
   * From the frequent itemsets discovered, generate rules, applying a minimum confidence threshold to filter these rules.

# Project Workflow:

# 1. Introduction

This report provides a comprehensive overview of the process and findings from the analysis of transaction data using association rule learning algorithms. The primary objective is to utilize Apriori and FP-Growth algorithms to uncover meaningful patterns and associations among the transactions within various datasets including Amazon, BestBuy, K-Mart, Nike, and Generic.

# 2. Dataset Selection

The analysis begins with the selection of a dataset by the user. The available options are 1) Amazon, 2) BestBuy, 3) K-Mart, 4) Nike, and 5) Generic. Based on the user's choice, the corresponding dataset is loaded for analysis. This allows for flexibility in the analysis and the opportunity to explore different transaction patterns across various retail contexts.

Below are the datasets used in this project. You can give any of the dataset to the program unless the format is same as below.

## 1) Amazon

|  |  |
| --- | --- |
| **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 |

Table 1 Amazon Item Names

|  |  |
| --- | --- |
| Transaction ID | Transaction |
| Trans1 | A Beginner’s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch |
| Trans2 | A Beginner’s Guide, Java: The Complete Reference, Java For Dummies |
| Trans3 | A Beginner’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 Beginner’s Guide, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition |
| Trans7 | A Beginner’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 Beginner’s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch |
| Trans12 | A Beginner’s Guide, Java: The Complete Reference, Java For Dummies, HTML and CSS: Design and Build Websites |
| Trans13 | A Beginner’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 Beginner’s Guide, Java: The Complete Reference, Java For Dummies, Android Programming: The Big Nerd Ranch |
| Trans17 | A Beginner’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 Beginner’s Guide, Java: The Complete Reference, Java For Dummies |

Table 2 Amazon Data Sets and Transactions

## 2) Best Buy

|  |  |
| --- | --- |
| **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 |

Table 3 Best Buy Items Names

|  |  |
| --- | --- |
| Transaction  ID | 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 |

Table 4 Best Buy Data Sets Transactions

## 3) K-mart

|  |  |
| --- | --- |
| **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 |

Table 5 K-Mart data items

|  |  |
| --- | --- |
| Transaction ID | 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 |

Table 6 K-Mart Data Sets Transactions

## 4) Nike

|  |  |
| --- | --- |
| **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 |

Table 7 Nike Data Items

|  |  |
| --- | --- |
| Transaction ID | 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 |

Table 8 Nike Data Sets Transactions

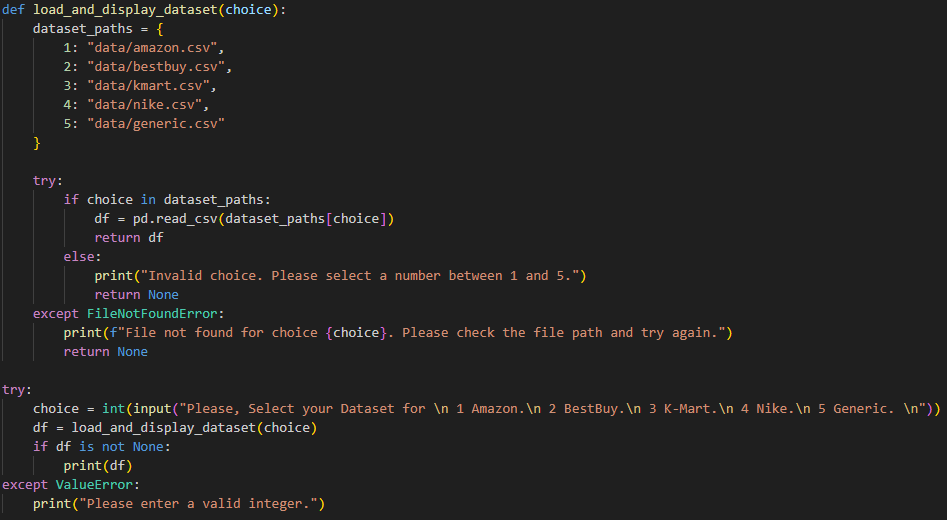
## 5) Generic

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

Table 9 Generic Items Names

|  |  |
| --- | --- |
| Transaction  ID | 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 |

Table 10 Generic Data Sets and Transactions



If we select 1 as our data, then the result will be.A screenshot of a computer program

Description automatically generated

# 3. User Inputs for Minimum Support and Confidence

The user is prompted to enter minimum support and confidence levels for the analysis. These thresholds are critical in determining the frequency and strength of the association rules derived from the data. Minimum support defines the lowest level of frequency an itemset must have to be considered relevant, while minimum confidence indicates the reliability of the inferred rules.

A black background with white text

Description automatically generated

A black background with white text

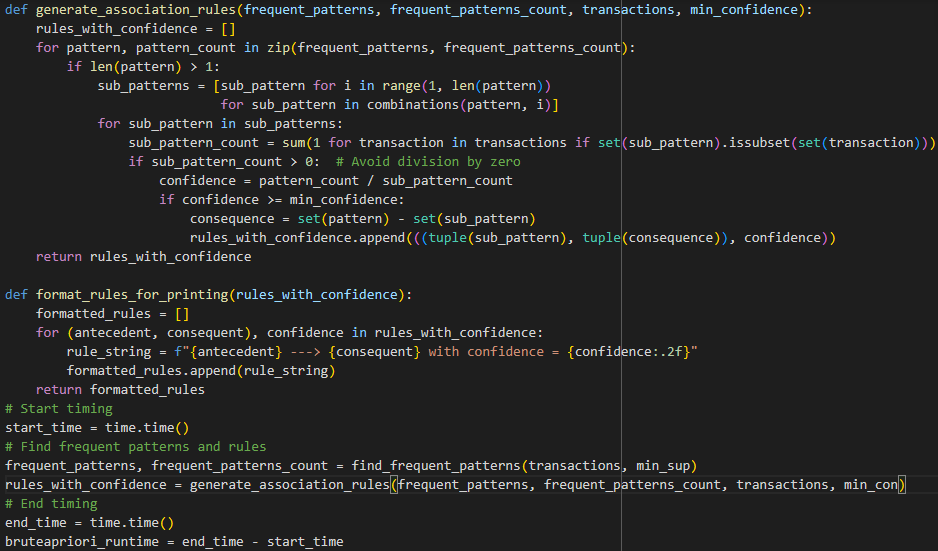
Description automatically generated

# 4. Brute-Forced Apriori Implementation

The brute-forced Apriori implementation involves manually processing the transaction data to identify frequent itemsets and generate association rules based on the user-defined minimum support and confidence. This section outlines the step-by-step process of the algorithm, including the preprocessing of transactions, computation of frequent itemsets, and rule generation.

A computer screen shot of a program code

Description automatically generated



A computer screen shot of a program

Description automatically generated

**Results:**

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

# 5. Validation with Python Package of Apriori

To validate the results obtained from the brute-forced approach, the 'mlxtend' library's Apriori function is used. This section compares the performance and output of the manual implementation with the package-based approach, highlighting the efficiency and accuracy of using a well-optimized library for association rule learning.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

# 6. Validation with FP-Growth Algorithm

The FP-Growth algorithm offers an efficient alternative to Apriori for finding frequent itemsets without candidate generation. This section describes the application of FP-Growth, using the 'mlxtend' library, to the transaction data and compares its performance and findings with the Apriori implementations.

A screen shot of a computer program

Description automatically generated

A computer screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

# 7. Comparing time taken by each algorithm

A computer screen shot of white text

Description automatically generated

A black background with white text

Description automatically generated

# 8. Results based on different minimum support and confidence on each data set.

## For Amazon Data set:

1. Support: 0.01 (1%), Confidence: 0.7 (70%)

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated



1. Support: 0.2 (20%), Confidence: 0.7 (70%)

A computer screen shot of a black screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. Support: 0.5 (50%), Confidence: 0.7 (70%)

A screenshot of a computer program

Description automatically generated

1. Support: 0.3 (30%), Confidence: 0.5 (50%)

A screenshot of a computer program

Description automatically generated

## For BestBuy Dataset:

1. Support: 0.5 (50%), Confidence: 0.7 (70%)

A computer screen shot of a computer program

Description automatically generated

A black background with white text

Description automatically generated

1. Support: 0.4 (40%), Confidence: 0.7 (70%)

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. Support: 0.01 (1%), Confidence: 0.7 (70%)

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

## For Kmart Dataset:

1. Support: 0.2 (20%), Confidence: 0.5 (50%)

A computer screen shot of a black screen

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Support: 0.1 (10%), Confidence: 0.5 (50%)

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A black background with white text

Description automatically generated

## For Nike Dataset:

1. Support: 0.5 (50%), Confidence: 0.5 (50%)

A screenshot of a computer program

Description automatically generated

1. Support: 0.4 (40%), Confidence: 0.5 (50%)

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screenshot of a computer code

Description automatically generated

## For Generic Dataset:

1. Support: 0.1 (10%), Confidence: 0.1 (10%)

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A black background with white text

Description automatically generated

1. Support: 0.3 (30%), Confidence: 0.5 (50%)

A screenshot of a computer program

Description automatically generated

# 9. Performance Analysis

The program implements a brute-force method alongside the **mlxtend** package's Apriori and FP-Growth algorithms to validate the results and compare performance times:

* **Brute-Forced Apriori**: Custom implementation to understand the underlying process of the Apriori algorithm.
* **Apriori (mlxtend)**: Utilizes the **mlxtend** implementation for efficient computation of frequent itemsets and association rules.
* **FP-Growth (mlxtend)**: Employs the FP-Growth algorithm for potentially faster performance in finding frequent itemsets, especially beneficial for large datasets.

## Results:

The output includes the frequent itemsets and the derived association rules that meet the specified minimum support and confidence levels. Each rule is accompanied by its confidence level, providing insight into the strength of the association.

## Conclusion:

The program effectively demonstrates the application of association rule mining in analyzing transaction data to uncover patterns and relationships between items. The use of both Apriori and FP-Growth algorithms showcases the versatility in approach depending on the dataset size and complexity. Performance comparison highlights the efficiency of the **mlxtend** implementations, making them suitable for practical applications in data mining projects.

## Recommendations:

For optimal performance and results, users are advised to adjust the minimum support and confidence thresholds based on the dataset characteristics and the specific goals of the analysis. Larger datasets may benefit from the FP-Growth algorithm due to its efficiency in handling dense data structures.

# Steps to run the Program:

Minimum requirements to run the program:

* **CPU**: Intel Core i3 or equivalent AMD
* **RAM**: 4GB (8GB recommended for smoother performance)
* **Storage**: 256GB SSD (for faster read/write speeds)
* **OS**: Windows 10, macOS, or a modern Linux distribution
* **Software**: Latest version of Python, Jupyter Notebook or Visual Studio Code

## STEP 1: Clone the repository.

Clone the repository from the repository to your local machine using Git. You can do this by running the following command in your terminal or Instead of cloning repository, one can download the zip file located in the repository.

git clone https://github.com/yashwanthreddy7178/Boddireddy\_Yashwanth\_Reddy\_midtermproj.git

## STEP 2: Create Virtual Environment.

Create a conda environment after opening the repository. It is recommended to do this project in a virtual environment to avoid conflicts with other Python packages you may have installed.

conda **create** -n myenv python -y  
conda **activate** myenv

or

**python** -m venv **env**  
**source** **env**/bin/activate

Note: On Windows, use `env\Scripts\activate`

## STEP 3: Install required libraries.

install the requirements!

pip **install** -r requirements.txt

## STEP 4: Run the python file.

You can use either python apriori.py

or

run it using jupyter notebook by opening the `Apriori.ipynb` file.

Note: Please make sure you have installed the required packages before running the code. If any package is missing, please install it from the requirements.txt file.

python apriori.py

Follow the prompts: The script will prompt you to enter a number corresponding to the dataset you want to load. Enter a number between 1 and 5.

Enter the minimum confidence: The script will then prompt you to enter the minimum confidence for the association rules. Enter a valid floating-point number (e.g., 0.1).

# Conclusion

This report summarizes the application and comparison of Brute forced Apriori, Apriori and FP-Growth algorithms for transaction data analysis. FP-Growth's efficiency, especially in terms of runtime, demonstrates its suitability for large-scale data analysis. The insights gained from this analysis have significant implications for understanding consumer behavior and optimizing retail strategies.

# References and Links:

**Note to the Grader:** I have created a repository using my personal account. Because my college email is linked with it. If I remove that mail id, I will lose all the benefits of GitHub pro and student benefits. So, please consider this. Thank You.

**GitHub Repository link:** <https://github.com/yashwanthreddy7178/Boddireddy_Yashwanth_Reddy_midtermproj>

**References:**

1. Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules.
2. Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation.
3. https://www.learnpython.org/
4. https://www.w3schools.com/python/
5. https://stackoverflow.com/questions/
6. Lectures Notes