NAME: Kartikey Patel
NJIT UCID: kp888
Email Address: kp888@njit.edu
13/10/2024
Professor: Yasser Abduallah
CS 634 101 Data Mining

Midterm Project Report

Implementation and Code Usage

Abstract:

Data mining offers a powerful way to uncover hidden patterns and associations within large datasets. This project focuses on comparing three key methods in association rule mining: Brute Force, Apriori, and FP-Tree. The process begins by identifying frequent items within transaction lists and calculating their support based on user-defined parameters, removing items that don't meet the minimum threshold.

The Brute Force method checks all possible item combinations to find frequent itemsets, but can be time-consuming for large datasets. In contrast, the Apriori Algorithm iteratively builds larger itemsets, filtering out those that don't meet the support threshold, offering a more balanced approach. The FP-Tree method uses a tree structure to efficiently encode the dataset, reducing the need for multiple scans and making it highly effective for large datasets.

Introduction:

In this project, I utilized all three methods on a custom dataset related to a retail store to compare their effectiveness in identifying frequent itemsets and association rules. Key steps involved:

- Initializing dictionaries for tracking candidate and frequent itemsets,
- Loading data and itemsets from CSV files,
- Preprocessing the data for item order and uniqueness, and
- Gathering user input for minimum support and confidence thresholds.
- Each method was then implemented to generate frequent itemsets and association rules. Through this comparative analysis, we gained valuable insights into the advantages and limitations of each method, enhancing our understanding of how they can be applied to uncover meaningful associations in retail transactions.

Core Concepts and Principles:

Discovery of Frequent Itemsets:

The central focus of my project revolves around the implementation and analysis of three primary techniques in association rule mining: Brute Force, Apriori, and FP-Tree. These methods are designed to identify frequent itemsets, representing groups of items that frequently appear together in transactional data. Uncovering these itemsets is instrumental in understanding consumer purchasing habits and preferences, ultimately aiding in enhancing sales strategies. The project explores how each technique contributes to the overall discovery of significant item associations.

Understanding Support and Confidence:

Support and confidence are two essential metrics in association rule mining that direct the analytical process and inform decision-making. Support quantifies the frequency with which a particular item or itemset appears in the dataset, while confidence evaluates the probability that items will be purchased together. These metrics are critical for determining the strength and relevance of identified patterns, providing a foundation for evaluating the usefulness of the mined association rules.

Unveiling Association Rules:

Detecting robust association rules is crucial for interpreting the relationships between items and for optimizing business sales strategies. These rules emphasize the combinations of items that are often bought together, making it possible to implement targeted marketing campaigns and offer personalized product recommendations to customers. Such insights not only improve the customer experience but also enhance retail efficiency and profitability by aligning with consumer behavior patterns.

Project Workflow:

My project follows a structured workflow, encompassing the implementation of Brute Force, Apriori, and FP-Tree methods:

Data Loading and Preprocessing:

The process starts by loading transactional data sourced from retail store datasets, where each entry contains a set of items purchased by individual customers. Preprocessing the data is a critical step, as it involves filtering for unique items and organizing them in a predetermined sequence to maintain data consistency. This initial phase ensures that the datasets are prepared for accurate analysis and prevents redundant or incorrect data from skewing the results.

Setting Minimum Support and Confidence Levels:

User-defined parameters are an integral component of this data mining project. Specifically, users provide input regarding the minimum thresholds for support and confidence, which serve as essential criteria for filtering out less relevant patterns. By adjusting these levels, users can focus the analysis on the most meaningful associations, enabling the discovery of itemsets that hold significant importance in practical applications.

Iterative Generation of Candidate Itemsets:

The project employs an iterative approach using the Brute Force, Apriori, and FP-Tree methods, focusing on generating candidate itemsets of increasing sizes. The process begins with individual items (itemsets of size K = 1) and continues incrementally to itemsets of size K = 2, K = 3, and beyond. The 'brute force' technique is applied here to exhaustively create all potential combinations of itemsets, ensuring a comprehensive analysis of possible associations. This exhaustive method ensures that no potential frequent itemset is overlooked.

Calculation of Support Counts:

Each candidate itemset undergoes a support calculation, which involves counting the number of transactions that include the itemset. Only those itemsets that meet or exceed the predefined minimum support threshold are kept for further analysis, while those that fall short are excluded. This process ensures that the retained itemsets are statistically significant and relevant for generating association rules, enhancing the overall quality of the findings.

Evaluation of Confidence:

The next phase involves evaluating the confidence of the generated association rules, which reflects the strength of relationships between items in the dataset. This assessment entails a detailed comparison of support values for both individual items and larger itemsets, ensuring that the derived rules represent reliable and meaningful patterns. Confidence analysis is key to identifying which rules hold the most practical value for predicting customer behavior.

Extraction of Association Rules:

Association rules that satisfy both the minimum support and confidence thresholds are identified and extracted. These rules offer significant insights into items that are frequently purchased together, facilitating informed business decisions. The extraction of such rules forms the basis for understanding recurring patterns in the dataset, providing a foundation for strategic marketing and inventory management.

Results and Evaluation:

The success and efficiency of the project are assessed through performance metrics such as support, confidence, and the quality of the resulting association rules. Additionally, a comparative analysis is performed between the custom implementations of Brute Force, Apriori, and FP-Tree methods and their corresponding library versions, evaluating their reliability and computational efficiency. This comparison offers valuable insights into the trade-offs between manual implementation and pre-built algorithms.

Conclusion:

In summary, this project demonstrates the practical use of core data mining methodologies, including the Brute Force, Apriori, and FP-Tree algorithms, to derive meaningful association rules from retail transaction datasets. Through an iterative approach and the integration of custom algorithms, the project highlights the effectiveness of data mining in revealing valuable patterns, which can drive informed decision-making in retail environments. This work underscores the value of association rule mining for both academic research and real-world applications in business strategy development.

Screenshots

Here are what the csv files (This program takes in 5 separate csv files: Item Names & Transactions).

Figure 1 : Amazon CSV file.

amazon

Transaction ID	Books
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

Figure 2 : BestBuy CSV file.

BestBuy

Transaction ID	Items
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

Figure 3 : kmart CSV file.

Kmart

Transaction ID	Items
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

Figure 4 : Nike CSV file.

nike

Transaction ID	Items
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

Figure 5 : Generic CSV file.

Generic

Transaction ID	Items
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

Below are screenshots of the code from python file:

Importing Necessary libraries

```
import csv
import itertools
import time
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules, fpgrowth
import pandas as pd
from brute_force_module import brute_force, print_frequent_itemsets, print_rules # Using the brute_force_module code to run the brute force method code
```

Brute Force Method:

```
# Brute Force
def brute_force(transactions, min_support, min_confidence):
    items = set(item for transaction in transactions for item in transaction)
    itemsets = []
    for i in range(1, len(items) + 1):
        itemsets.extend(itertools.combinations(items, i))
    frequent_itemsets = {}
    for itemset in itemsets:
        frequency = sum(1 for transaction in transactions if set(itemset).issubset(transaction))
        support = frequency / len(transactions)
        if support >= min_support:
            frequent_itemsets[itemset] = support
    rules = generate_association_rules(frequent_itemsets, transactions, min_confidence)

    return frequent_itemsets, rules
```

Defining Association rule generation function:

```
# Function to generate association rules
def generate_association_rules(frequent_itemsets, transactions, min_confidence):
    rules = []
    for itemset, support in frequent_itemsets.items():
        if len(itemset) > 1:
            for i in range(1, len(itemset)):
                for antecedent in itertools.combinations(itemset, i):
                    consequent = set(itemset) - set(antecedent)
                    antecedent_transactions = sum(1 for transaction in transactions if set(antecedent).issubset(transaction))
                    if antecedent_transactions > 0:
                        confidence = support / (antecedent_transactions / len(transactions))
                        if confidence >= min_confidence:
                            rules.append((antecedent, consequent, support, confidence))
    return rules
```

Defining Frequent Itemsets function:

```
# Function to print frequent itemsets
def print_frequent_itemsets(frequent_itemsets, method):
    if frequent_itemsets:
        itemsets_df = pd.DataFrame(
            [(set(itemset), support) for itemset, support in frequent_itemsets.items()],
            columns=['itemsets', 'support']
        )
        itemsets_df['itemsets'] = itemsets_df['itemsets'].apply(lambda x: ', '.join(x))
        print(f"{method} Frequent Itemsets:")
        print(itemsets_df.to_string(index=False))
    else:
        print(f"No frequent itemsets found using {method}.")
    print("\n")
```

Reading csv files and printing association rule function:

```
# Function to print association rules
def print_rules(rules, method):
    if rules:
        rules_df = pd.DataFrame(rules, columns=['antecedents', 'consequents', 'support', 'confidence'])
        rules_df['antecedents'] = rules_df['antecedents'].apply(lambda x: ', '.join(x))
        rules_df['consequents'] = rules_df['consequents'].apply(lambda x: ', '.join(x))
        print(f"{method} Association Rules:")
        print(rules_df[['antecedents', 'consequents', 'support', 'confidence']].to_string(index=False))
    else:
        print(f"No association rules found using {method}.")
    print("\n")
```

```
def read_data(filename):
    transactions = []
    with open(filename, 'r') as file:
        csv_reader = csv.reader(file)
        next(csv_reader)  # Skip header row
        for row in csv_reader:
            transactions.append(row[1].split(", "))
    return transactions
```

Apriori and FP-Tree Method:

```
def run_apriori_fpgrowth(transactions, min_support, min_confidence):
    te = TransactionEncoder()
    te_ary = te.fit(transactions).transform(transactions)
    df = pd.DataFrame(te_ary, columns=te.columns_)

    #Apriori

    start_time = time.time()
    frequent_itemsets_apriori = apriori(df, min_support=min_support, use_colnames=True)
    rules_apriori = association_rules(frequent_itemsets_apriori, metric="confidence", min_threshold=min_confidence)
    end_time = time.time()
    print(f"Apriori Execution Time: {end_time - start_time} seconds \n")
    print("Apriori Frequent Itemsets")
    print(frequent_itemsets_apriori, "\n")
    if not rules_apriori.empty:
        print("Apriori Association Rules : ")
        # Convert antecedents and consequents to strings without square brackets
        rules_apriori['antecedents'] = rules_apriori['antecedents'].apply(lambda x: ', '.join(x))
        rules_apriori['consequents'] = rules_apriori['consequents'].apply(lambda x: ', '.join(x))
        print(rules_apriori[['antecedents', 'consequents', 'support', 'confidence']].head(10).to_string(index=False))
    else:
        print("No association rules found.")
    print("\n")
```

```
    # FP-Growth

    start_time = time.time()
    frequent_itemsets_fp = fpgrowth(df, min_support=min_support, use_colnames=True)
    rules_fp = association_rules(frequent_itemsets_fp, metric="confidence", min_threshold=min_confidence)
    end_time = time.time()
    print(f"FP-Growth Execution Time: {end_time - start_time} seconds \n")
    print("FP-Growth Frequent Itemsets")
    print(frequent_itemsets_fp, "\n")
    if not rules_fp.empty:
        print("FP-Growth Association Rules : ")
        # Convert antecedents and consequents to strings without square brackets
        rules_fp['antecedents'] = rules_fp['antecedents'].apply(lambda x: ', '.join(x))
        rules_fp['consequents'] = rules_fp['consequents'].apply(lambda x: ', '.join(x))
        print(rules_fp[['antecedents', 'consequents', 'support', 'confidence']].head(10).to_string(index=False))
    else:
        print("No association rules found.")
    print("\n")
```

Main:

```python
def main():
    datasets = ["amazon.csv", "BestBuy.csv", "Kmart.csv", "nike.csv", "Generic.csv"]
    dataset_choice = int(input("Choose a dataset (1-5): \n1.Amazon\n2.BestBuy\n3.Kmart\n4.Nike\n5.Generic\n"))
    min_support = float(input("Enter minimum support (as a decimal): "))
    min_confidence = float(input("Enter minimum confidence (as a decimal): "))

    transactions = read_data(datasets[dataset_choice - 1])

    # Brute Force
    start_time = time.time()
    frequent_itemsets, rules = brute_force(transactions, min_support, min_confidence)
    end_time = time.time()
    print(f"\nBrute Force Execution Time: {end_time - start_time} seconds \n")
    print_frequent_itemsets(frequent_itemsets, method="Brute Force")
    print_rules(rules, method="Brute Force")


    # Apriori and FP-Growth
    run_apriori_fpgrowth(transactions, min_support, min_confidence)

if __name__ == "__main__":
    main()
```

Below are screenshots to show the running of the program:

```
C:\Windows\System32\cmd.e

Choose a dataset (1-5):
1.Amazon
2.BestBuy
3.Kmart
4.Nike
5.Generic
1
Enter minimum support (as a decimal): 0.5
Enter minimum confidence (as a decimal): 0.4

Brute Force Execution Time: 0.01558232307434082 seconds

Brute Force Frequent Itemsets:
                                      itemsets  support
       Android Programming: The Big Nerd Ranch     0.65
                  Java: The Complete Reference     0.50
                            A Beginner's Guide     0.55
                              Java For Dummies     0.65
Java For Dummies, Java: The Complete Reference     0.50


Brute Force Association Rules:
                 antecedents                  consequents  support  confidence
Java: The Complete Reference             Java For Dummies      0.5    1.000000
            Java For Dummies Java: The Complete Reference      0.5    0.769231


Apriori Execution Time: 0.01550602912902832 seconds

Apriori Frequent Itemsets
   support                                        itemsets
0     0.55                            (A Beginner's Guide)
1     0.65         (Android Programming: The Big Nerd Ranch)
2     0.65                              (Java For Dummies)
3     0.50                  (Java: The Complete Reference)
4     0.50  (Java For Dummies, Java: The Complete Reference)

Apriori Association Rules :
                 antecedents                  consequents  support  confidence
            Java For Dummies Java: The Complete Reference      0.5    0.769231
Java: The Complete Reference             Java For Dummies      0.5    1.000000


FP-Growth Execution Time: 0.0013859272003173828 seconds
```

For Amazon Transactions:

```
Brute Force Execution Time: 0.004131555557250976 6 seconds

Brute Force Frequent Itemsets:
                                 itemsets  support
       Android Programming: The Big Nerd Ranch 0.650000
                  Java: The Complete Reference 0.500000
                             Java For Dummies 0.650000
                            A Beginner's Guide 0.550000
Java For Dummies, Java: The Complete Reference 0.500000


Brute Force Association Rules:
                  antecedents                  consequents  support  confidence
Java: The Complete Reference              Java For Dummies 0.500000    1.000000
             Java For Dummies Java: The Complete Reference 0.500000    0.769231


Apriori Execution Time: 0.007901668548583984 seconds

Apriori Frequent Itemsets
   support                                        itemsets
0 0.550000                            (A Beginner's Guide)
1 0.650000         (Android Programming: The Big Nerd Ranch)
2 0.650000                              (Java For Dummies)
3 0.500000                  (Java: The Complete Reference)
4 0.500000  (Java For Dummies, Java: The Complete Reference)

Apriori Association Rules :
                  antecedents                  consequents  support  confidence
             Java For Dummies Java: The Complete Reference 0.500000    0.769231
Java: The Complete Reference              Java For Dummies 0.500000    1.000000


FP-Growth Execution Time: 0.0 seconds

FP-Growth Frequent Itemsets
   support                                        itemsets
0 0.650000                              (Java For Dummies)
1 0.650000         (Android Programming: The Big Nerd Ranch)
2 0.550000                            (A Beginner's Guide)
3 0.500000                  (Java: The Complete Reference)
4 0.500000  (Java For Dummies, Java: The Complete Reference)

FP-Growth Association Rules :
                  antecedents                  consequents  support  confidence
             Java For Dummies Java: The Complete Reference 0.500000    0.769231
Java: The Complete Reference              Java For Dummies 0.500000    1.000000
```

For BestBuy transactions:

```
Brute Force Execution Time: 0.008385419845581055 seconds

Brute Force Frequent Itemsets:
                  itemsets  support
              Lab Top Case 0.700000
               Flash Drive 0.650000
                Anti-Virus 0.700000
                   Lab Top 0.600000
Lab Top Case, Anti-Virus 0.600000


Brute Force Association Rules:
 antecedents  consequents  support  confidence
Lab Top Case   Anti-Virus 0.600000    0.857143
  Anti-Virus Lab Top Case 0.600000    0.857143


Apriori Execution Time: 0.00910639762878418 seconds

Apriori Frequent Itemsets
    support                    itemsets
0 0.700000                (Anti-Virus)
1 0.650000               (Flash Drive)
2 0.600000                   (Lab Top)
3 0.700000              (Lab Top Case)
4 0.600000  (Lab Top Case, Anti-Virus)

Apriori Association Rules :
 antecedents  consequents  support  confidence
Lab Top Case   Anti-Virus 0.600000    0.857143
  Anti-Virus Lab Top Case 0.600000    0.857143


FP-Growth Execution Time: 0.0 seconds

FP-Growth Frequent Itemsets
    support                    itemsets
0 0.700000                (Anti-Virus)
1 0.650000               (Flash Drive)
2 0.700000              (Lab Top Case)
3 0.600000                   (Lab Top)
4 0.600000  (Lab Top Case, Anti-Virus)

FP-Growth Association Rules :
 antecedents  consequents  support  confidence
Lab Top Case   Anti-Virus 0.600000    0.857143
  Anti-Virus Lab Top Case 0.600000    0.857143
```

For Kmart Transactions:

```

Brute Force Execution Time: 0.00601959228515625 seconds

Brute Force Frequent Itemsets:
               itemsets  support
      Decorative Pillows 0.500000
                  Sheets 0.500000
                   Shams 0.550000
            Kids Bedding 0.600000
              Bed Skirts 0.550000
    Kids Bedding, Sheets 0.500000
Kids Bedding, Bed Skirts 0.500000


Brute Force Association Rules:
 antecedents  consequents  support  confidence
      Sheets Kids Bedding 0.500000    1.000000
Kids Bedding       Sheets 0.500000    0.833333
Kids Bedding   Bed Skirts 0.500000    0.833333
  Bed Skirts Kids Bedding 0.500000    0.909091


Apriori Execution Time: 0.010286331176757812 seconds

Apriori Frequent Itemsets
   support                   itemsets
0 0.550000               (Bed Skirts)
1 0.500000       (Decorative Pillows)
2 0.600000             (Kids Bedding)
3 0.550000                    (Shams)
4 0.500000                   (Sheets)
5 0.500000 (Kids Bedding, Bed Skirts)
6 0.500000     (Kids Bedding, Sheets)

Apriori Association Rules :
 antecedents  consequents  support  confidence
Kids Bedding   Bed Skirts 0.500000    0.833333
  Bed Skirts Kids Bedding 0.500000    0.909091
Kids Bedding       Sheets 0.500000    0.833333
      Sheets Kids Bedding 0.500000    1.000000


FP-Growth Execution Time: 0.0024073123931884766 seconds

FP-Growth Frequent Itemsets
   support                   itemsets
0 0.500000       (Decorative Pillows)
1 0.600000             (Kids Bedding)
2 0.550000                    (Shams)
3 0.550000               (Bed Skirts)
4 0.500000                   (Sheets)
5 0.500000 (Kids Bedding, Bed Skirts)
6 0.500000     (Kids Bedding, Sheets)

FP-Growth Association Rules :
 antecedents  consequents  support  confidence
Kids Bedding   Bed Skirts 0.500000    0.833333
  Bed Skirts Kids Bedding 0.500000    0.909091
Kids Bedding       Sheets 0.500000    0.833333
      Sheets Kids Bedding 0.500000    1.000000


```

For Nike Transactions:

```

Brute Force Execution Time: 0.02388691902160645 seconds

Brute Force Frequent Itemsets:
           itemsets  support
         Rash Guard 0.600000
       Running Shoe 0.700000
              Socks 0.650000
        Sweatshirts 0.650000
Socks, Sweatshirts 0.600000


Brute Force Association Rules:
antecedents consequents  support  confidence
      Socks Sweatshirts 0.600000    0.923077
Sweatshirts       Socks 0.600000    0.923077


Apriori Execution Time: 0.0028810501098632812 seconds

Apriori Frequent Itemsets
   support              itemsets
0 0.600000          (Rash Guard)
1 0.700000        (Running Shoe)
2 0.650000               (Socks)
3 0.650000         (Sweatshirts)
4 0.600000  (Socks, Sweatshirts)

Apriori Association Rules :
antecedents consequents  support  confidence
      Socks Sweatshirts 0.600000    0.923077
Sweatshirts       Socks 0.600000    0.923077


FP-Growth Execution Time: 0.015687227249145508 seconds

FP-Growth Frequent Itemsets
   support              itemsets
0 0.700000        (Running Shoe)
1 0.650000         (Sweatshirts)
2 0.650000               (Socks)
3 0.600000          (Rash Guard)
4 0.600000  (Socks, Sweatshirts)

FP-Growth Association Rules :
antecedents consequents  support  confidence
      Socks Sweatshirts 0.600000    0.923077
Sweatshirts       Socks 0.600000    0.923077
```

For Generic Transactions:

```
Brute Force Execution Time: 0.0 seconds

Brute Force Frequent Itemsets:
itemsets  support
       C 0.636364
       A 1.000000
       E 0.727273
    C, A 0.636364
    E, A 0.727273


Brute Force Association Rules:
antecedents consequents  support  confidence
          C           A 0.636364    1.000000
          A           C 0.636364    0.636364
          A           E 0.727273    0.727273
          E           A 0.727273    1.000000


Apriori Execution Time: 0.002692699432373047 seconds

Apriori Frequent Itemsets
   support itemsets
0 1.000000      (A)
1 0.636364      (C)
2 0.727273      (E)
3 0.636364   (C, A)
4 0.727273   (E, A)

Apriori Association Rules :
antecedents consequents  support  confidence
          C           A 0.636364    1.000000
          A           C 0.636364    0.636364
          E           A 0.727273    1.000000
          A           E 0.727273    0.727273


FP-Growth Execution Time: 0.0 seconds

FP-Growth Frequent Itemsets
   support itemsets
0 1.000000      (A)
1 0.636364      (C)
2 0.727273      (E)
3 0.636364   (C, A)
4 0.727273   (E, A)

FP-Growth Association Rules :
antecedents consequents  support  confidence
          C           A 0.636364    1.000000
          A           C 0.636364    0.636364
          E           A 0.727273    1.000000
          A           E 0.727273    0.727273
```

Other

The source code (.py file) and data sets (.csv files) will be attached to the zip file.

Link to Git Repository

https://github.com/kartikeyypatel/MidTerm_DataMining_Project