CS 634

Apriori Algorithm Implementation

Midterm project implementation 1

Himani Patel

Hpp5@njit.edu

Contents

Introduction……1

Apriori Algorithm Overview……..1

Requirement……..1

Software……..1

Hardware……...1

Name of Source Code Files……...1

List of Data set Files……....1

Implementation Overview……2

Implementation of algorithm in Python……..2

Testing Implementation and Datasets……4

Console Mode……4

Amazon……4

BestBuy……5

KMart……5

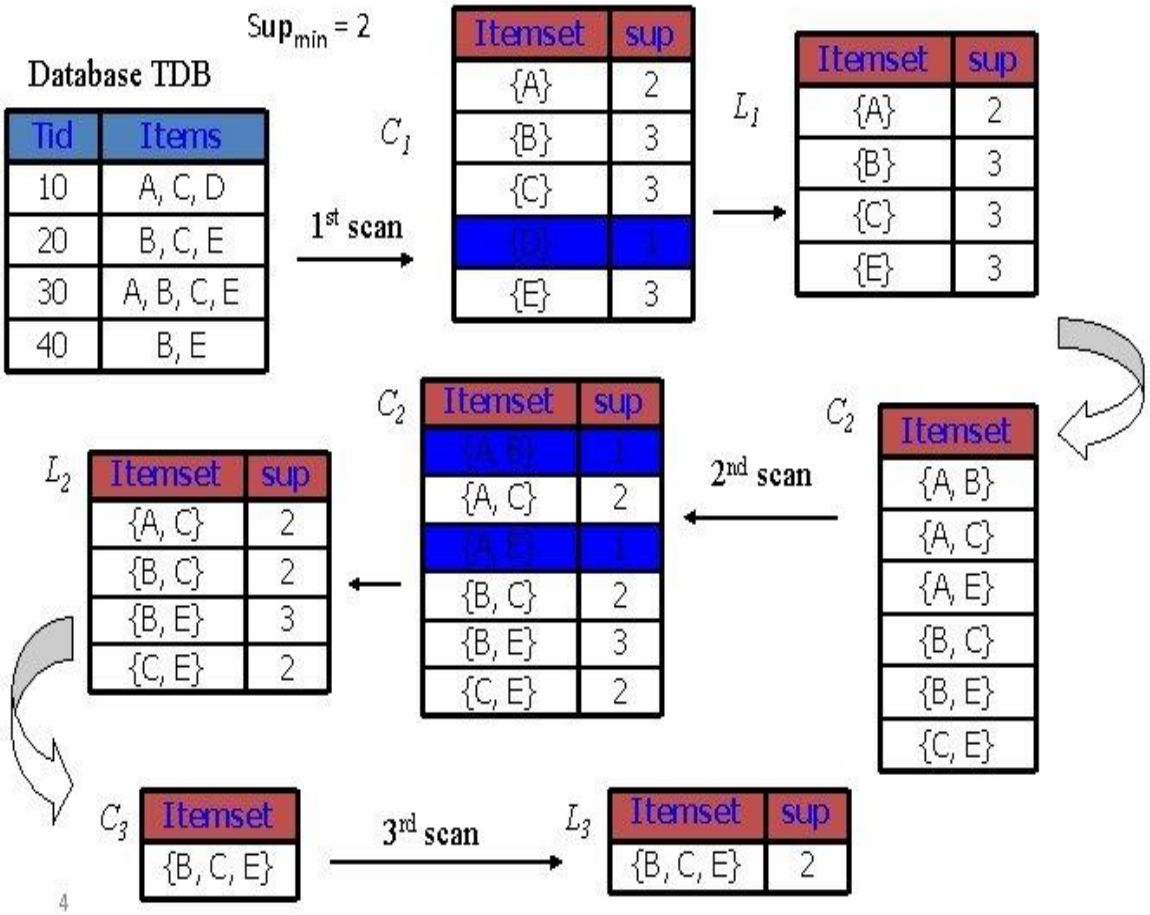
Nike……6

Custom Dataset……7

Implementation Logs and Execution Output……7

GitHub Link……14

The Apriori Algorithm—An Example



Introduction

Apriori Algorithm Overview:

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.

Requirements:

Software Version:
Python 3.8.8
Visual Studio Code

Hardware:
Mac Os Version 13.0.1

Name of source code files:
Algo.py

List of dataset Files:
1) Amazon
2) BestBuy
3) K-mart
4) Nike
5) Custom Dataset

Implementation Overview

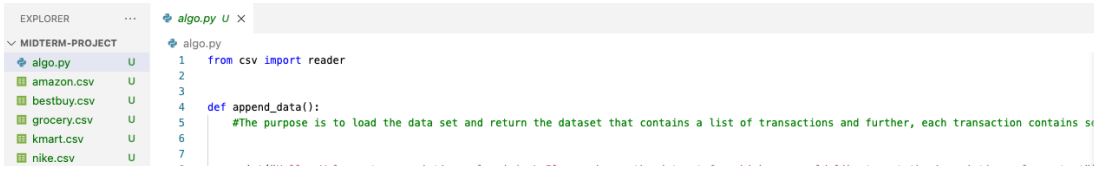
The implementation is built from scratch. It uses python as the main programming language. The implementation uses Database to store the items sets and transactions.

When you run the program, it will prompt you with a number of choices to choose from based on the dataset type, support and confidence.

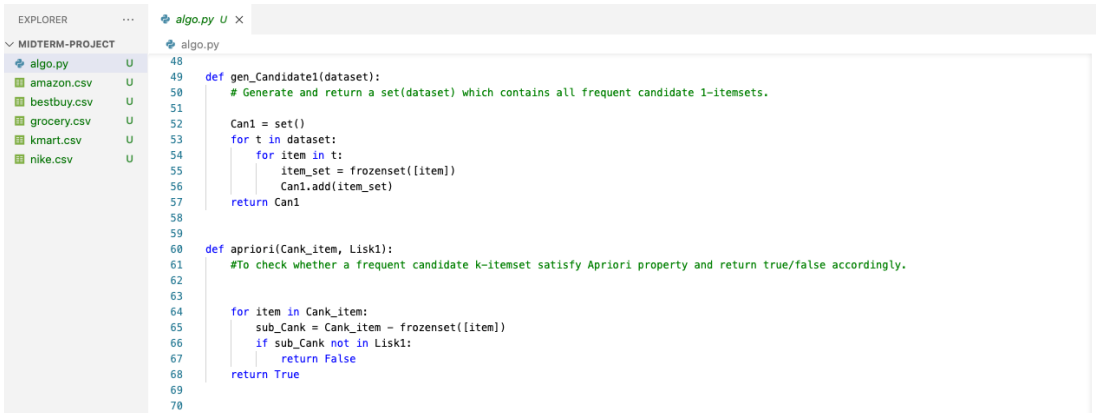
```
(base) himanis-air:midterm-project himanipatel$ python algo.py
Hello, Welcome to association rule mining
Please choose the dataset for which you would like to get the Association rules output
Please type
1 for Amazon
2 for BestBuy
3 for KMART
4 for Nike
5 for Custom
3
You selected (3)KMART Dataset
Enter the minimum support: 0.4
Enter the minimum Confidence: 0.2
```

Implementation of algorithm in Python:

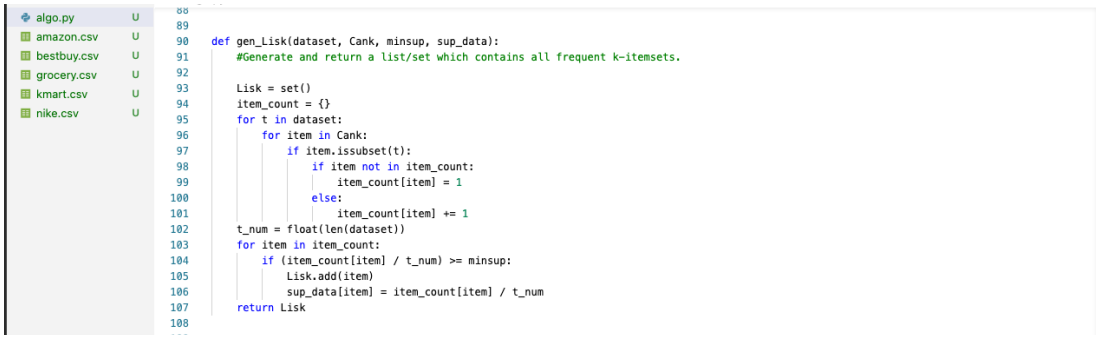
Step 1: Importing the required libraries



Step 2: Loading and exploring the data



Step 3: Generating itemset k-1




```
def gen_Lisk(dataset, Cank, minsup, sup_data):
    #Generate and return a list/set which contains all frequent k-itemsets.

    Lisk = set()
    item_count = {}
    for t in dataset:
        for item in Cank:
            if item.issubset(t):
                if item not in item_count:
                    item_count[item] = 1
                else:
                    item_count[item] += 1
    t_num = float(len(dataset))
    for item in item_count:
        if (item_count[item] / t_num) >= minsup:
            Lisk.add(item)
            sup_data[item] = item_count[item] / t_num
    return Lisk
```

```
def gen_L(dataset, k, minsup):
    #Generate all frequent itemsets by the value of minimum support's input.

    sup_data = {}
    Can1 = gen_Candidate1(dataset)
    List1 = gen_Lisk(dataset, Can1, minsup, sup_data)
    Lisksub1 = List1.copy()
    List = []
    List.append(Lisksub1)
    for i in range(2, k+1):
        Ci = gen_Cank(Lisksub1, i)
        Li = gen_Lisk(dataset, Ci, minsup, sup_data)
        Lisksub1 = Li.copy()
        List.append(Lisksub1)
    return List, sup_data
```

Step 5: Generating Association rules

```
def association_rules(List, sup_data, minconf):
    #Generate and return association rules.

    asso_rules = []
    sublist = []
    for i in range(0, len(List)):
        for freq_set in List[i]:
            for sub_set in sublist:
                if sub_set.issubset(freq_set):
                    conf = sup_data[freq_set] / sup_data[freq_set - sub_set]
                    rule = (freq_set - sub_set, sub_set, conf)
                    if conf >= minconf and rule not in asso_rules:
                        # print freq_set-sub_set, " => ", sub_set, "conf: ", conf
                        asso_rules.append(rule)
            sublist.append(freq_set)
    return asso_rules
```

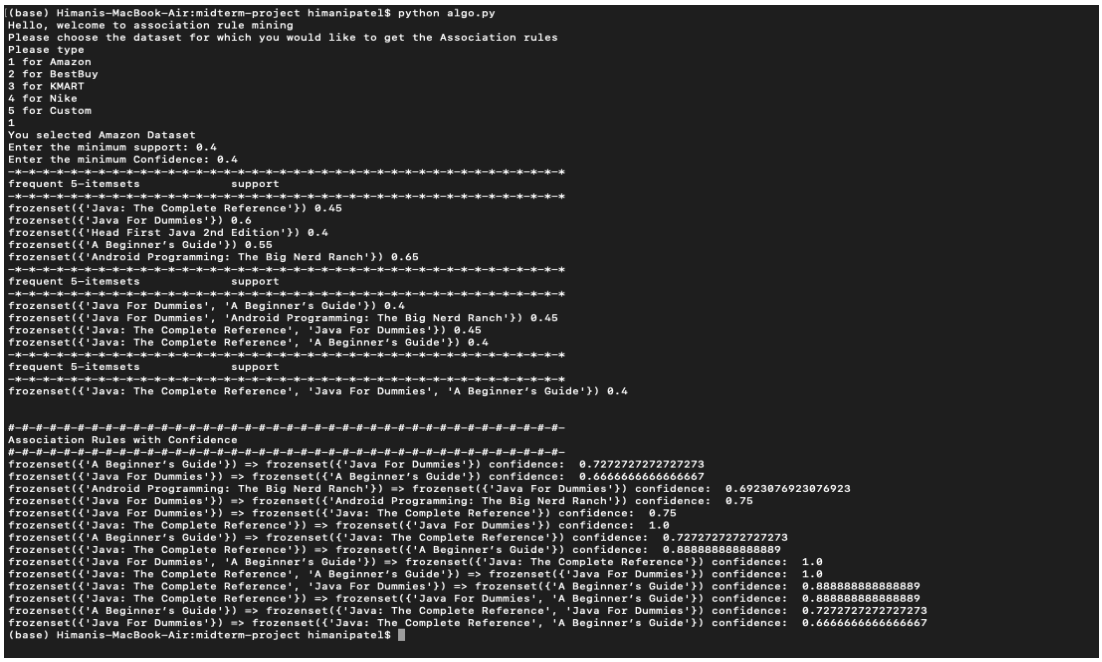
Testing Implementation and Datasets:

Console Mode

This section describes several tests for each data set type with different support and confidence values. It tests each data set respectively.

1)Amazon

Running amazon data set with minimum support 0.4 and minimum confidence 0.4



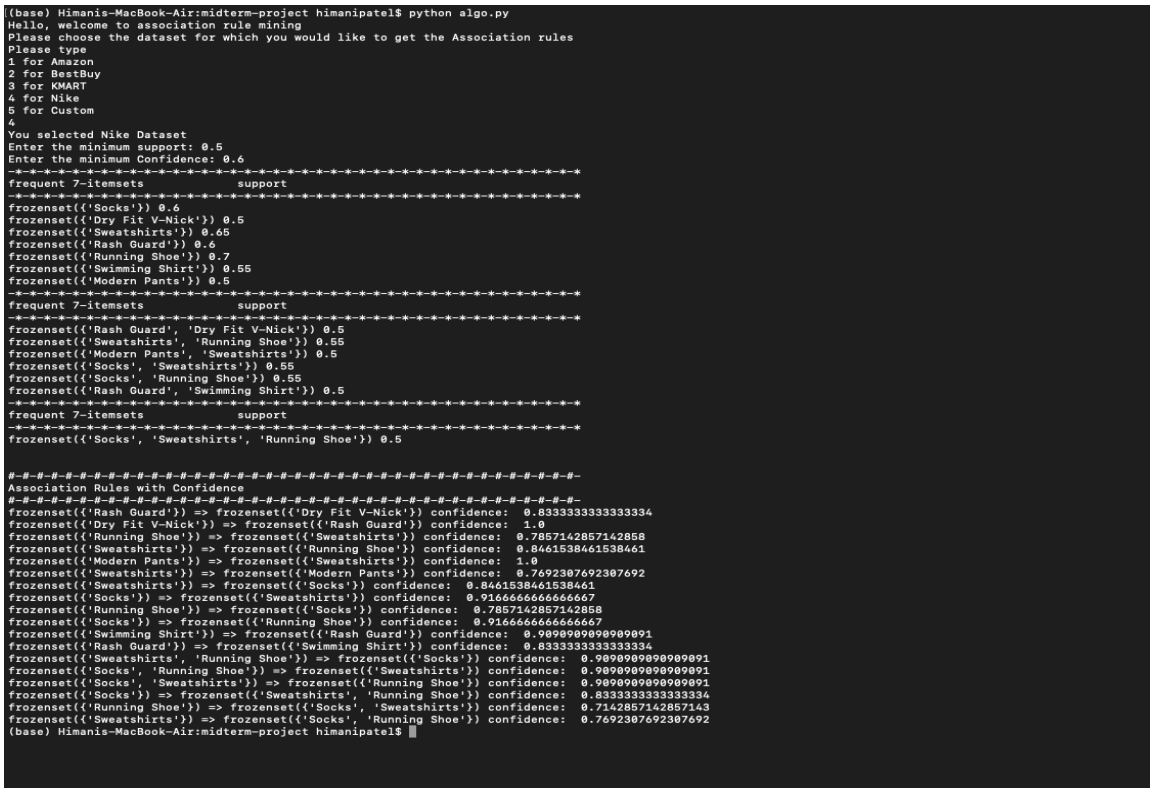
2) BestBuy

Running BestBuy data set with minimum support 0.5 and minimum confidence 0.5

```
(base) Himanis-MacBook-Air:midterm-project himanipatel$ python algo.py
Hello, welcome to association rule mining
Please choose the dataset for which you would like to get the Association rules
Please type
1 for Amazon
2 for BestBuy
3 for KMART
4 for Nike
5 for Custom
2
You selected BestBuy Dataset
Enter the minimum support: 0.5
Enter the minimum Confidence: 0.5
-+-+-+
```

4) Nike

Running Nike data set with minimum support 0.5 and minimum confidence 0.6



```
(base) Himanis-MacBook-Air:midterm-project himanipatel$ python algo.py
Hello, welcome to association rule mining
Please choose the dataset for which you would like to get the Association rules
Please type
1 for Amazon
2 for BestBuy
3 for KMART
4 for Nike
5 for Custom
4
You selected Nike Dataset
Enter the minimum support: 0.5
Enter the minimum Confidence: 0.6
-*-*-*
```

5) Custom Dataset

Running Nike data set with minimum support 0.5 and minimum confidence 0.6

```
(base) Himanis-MacBook-Air:midterm-project himanipatel$ python algo.py
Hello, welcome to association rule mining
Please choose the dataset for which you would like to get the Association rules
Please type
1 for Amazon
2 for BestBuy
3 for KMART
4 for Nike
5 for Custom
5
You selected Custom Dataset
Enter the minimum support: 0.2
Enter the minimum Confidence: 0.1
-*-*-*
```

frozenset({'Head First Java 2nd Edition'}) 0.4
frozenset({'Java For Dummies'}) 0.6
frozenset({'A Beginner’s Guide'}) 0.55
frozenset({'Android Programming: The Big Nerd Ranch'}) 0.65
-*-*-*

For BestBuy:

```
(base) Himanis-MacBook-Air:midterm-project himanipatel$ python algo.py
Hello, Welcome to association rule mining
Please choose the dataset for which you would like to get the Association rules output
Please type
1 for Amazon
2 for BestBuy
3 for KMART
4 for Nike
5 for Custom
2
You selected (2)BestBuy Dataset
Enter the minimum support: 0.5
Enter the minimum Confidence: 0.5
-*-*-*
```

frozenset({'Flash Drive'}) => frozenset({'Anti-Virus'}) confidence: 0.7692307692307692
frozenset({'Anti-Virus'}) => frozenset({'Flash Drive'}) confidence: 0.7142857142857143
frozenset({'Lab Top Case'}) => frozenset({'Lab Top'}) confidence: 0.7142857142857143
frozenset({'Lab Top'}) => frozenset({'Lab Top Case'}) confidence: 0.8333333333333334
frozenset({'Microsoft Office'}) => frozenset({'Flash Drive'}) confidence: 1.0
frozenset({'Flash Drive'}) => frozenset({'Microsoft Office'}) confidence: 0.8461538461538461
frozenset({'Anti-Virus'}) => frozenset({'Lab Top'}) confidence: 0.7142857142857143
frozenset({'Lab Top'}) => frozenset({'Anti-Virus'}) confidence: 0.8333333333333334

For KMart:

(base) Himanis-MacBook-Air:midterm-project himanipatel$ python algo.py

Hello, Welcome to association rule mining

Please choose the dataset for which you would like to get the Association rules output

Please type
1 for Amazon
2 for BestBuy
3 for KMART
4 for Nike
5 for Custom
3
You selected (3)KMART Dataset
Enter the minimum support: 0.4
Enter the minimum Confidence: 0.3
-*-*-*

#-#-#-

-*-*-*

5 for Custom
5
You selected (5)Custom Dataset
Enter the minimum support: 0.2
Enter the minimum Confidence: 0.1
-*-*-*

frozenset({'SUGER'}) => frozenset({'BREAD'}) confidence: 0.6666666666666667
frozenset({'CORNFLAKES'}) => frozenset({'COFFEE'}) confidence: 0.6666666666666667
frozenset({'COFFEE'}) => frozenset({'CORNFLAKES'}) confidence: 0.5
frozenset({'MAGGI'}) => frozenset({'TEA'}) confidence: 0.8
frozenset({'TEA'}) => frozenset({'MAGGI'}) confidence: 0.5714285714285715

Github Link: https://github.com/Hpp5/CS-634-Midterm-project