|  |
| --- |
| Apriori Algorithm |

Table of Contents

[Introduction 3](#_Toc42812918)

[Description 3](#_Toc42812919)

[Propose 3](#_Toc42812920)

[Assumption 3](#_Toc42812921)

[Requirement 4](#_Toc42812922)

[Steps to Run 4](#_Toc42812923)

[Source Code: 6](#_Toc42812924)

[Output 10](#_Toc42812925)

# Introduction

This project will discuss a data mining algorithm called the Apriori Algorithm. It uses frequent itemsets to generate association rules. We will iteratively find frequent itemsets with 1 to k-combinations. In addition, this algorithm stated that a subset of a frequent itemset must also be a frequent itemset.

# Description

We will implement the Apriori algorithm and analysis the runtime of algorithm against brute force method. We will show and prove that the algorithm is faster than brute force on generating the frequent itemset. Also, the results for both methods should be identical.

Apriori algorithm will find all the frequent of each item. Then, apply the minimum support to find all the frequent sets with k items in a database. This process is called pruning because superset of a non-frequent itemset must be non-frequent. It will continue to generate the next k-itemset until no frequent set is found. It will need to calculate the confidence of each frequent set and apply minimum confidence to generate the association rules.

Support and confidence for itemsets are represented by formulas below. For example, take A and B.

Support (A) = Number of transactions in which A appears

Total number transactions

Confidence (A 🡪 B) = Support (A U B)

Support(A)

# Propose

Brute force will create the combinations of k-itemset and it will use subroutine from Apriori Algorithm to filter out the minimum support. Both methods will then utilize the same module to generate orders with the minimum confidence. It will read each database or files separately and capture the runtime of each method. Also, it will print all association rules with the support and confidence percent.

# Assumption

For our dataset, it will be using 30 items and 20 transactions stored in a file or database. In this project, we will be using 5 CSV files. The filename will be provided in the code. The files containing all 20 transactions should be present in the same directory where the source code reside. Lastly, the support and confident threshold will be provided from the user.

# Requirement

* Jupyter Notebbok
  + Install [Anaconda](https://docs.anaconda.com/anaconda/install/) – This is be used to open Jupyter Notebook. It’s a Python distribution that simplifies package management. It installs with many data science libraries and Jupyter.
  + pip command is an alternative to install Jupyter. First, make sure the pip is in latest version.

|  |
| --- |
| pip3 install --upgrade pip |

Second, install the Jupyter Notebook using the command below.

|  |
| --- |
| pip3 install jupyter |

* 5 CSV files that will be included with this project.
* It will need NumPy and Pandas packages.

# Steps to Run

This project will be using Python 3 and Jupyter Notebook.

1. From the command line/terminal, we can start the notebook by running the command below.

jupyter notebook

1. It will open the tab from a web browser and select the python file. Since it is using Jupyter notebook, the file extension will be *ipynb*.

In my machine, the file is located in */Deskstop/CS\_634/*.

A close up of a beach

Description automatically generated

1. Once the file is open, you can click “Cell” and select “Run All”.

A screenshot of a cell phone

Description automatically generated

1. The last cell will print out the results/output.

A screenshot of a social media post

Description automatically generated

# Source Code:

*#!/usr/bin/env python  
# coding: utf-8  
  
# In[49]:***import** time **as** time  
**import** numpy **as** np  
**import** pandas **as** pd  
  
  
*# In[50]:***def** removeNaN(data):  
 resultList = [] *# return data without NaN* **for** i **in** range(0,len(data)):  
 resultList.append(list(data.iloc[i].dropna()))  
 **return** resultList  
  
  
*# In[51]:***def** allItem(data):  
 resultList=[] *# return list of all possible items* **for** transaction **in** data:  
 **for** item **in** transaction:  
 **if not** [item] **in** resultList:  
 resultList.append([item])  
 resultList=np.sort(resultList)  
 **return** list(map(frozenset, resultList))  
  
  
*# In[52]:***def** frequentItemset(data, itemset):  
 countList = {} *# count each candidate in our dataset* **for** d **in** data:  
 **for** i **in** itemset:  
 **if** i.issubset(d):  
 **if not** i **in** countList:   
 countList[i]=1  
 **else**:   
 countList[i] += 1   
   
 **return** countList  
  
  
*# In[53]:***def** minSupport(frequentItem, numItems, pctSupport = 0.2):  
 resultList=[] *# items that meets the minimum support* supInfo = {} *# dictionary that will contains support%  
   
 # calculate the frequency* **for** i **in** frequentItem:  
 support = round(frequentItem[i]/numItems,2)  
 supInfo[i] = support  
 **if** support >= pctSupport:  
 resultList.insert(0,i)  
 **return** resultList, supInfo  
  
  
*# In[54]:***def** generateItemset(frequentItem, n):   
 resultList = []  
 *#print(frequentItem)* **for** i **in** range(len(frequentItem)):  
 **for** j **in** range(i+1, len(frequentItem)):   
 item1 = list(frequentItem[i])[:n-2]  
 item1.sort()  
 item2 = list(frequentItem[j])[:n-2]  
 item2.sort()  
 **if** item1==item2: *#if first k-2 elements are equal  
 #print(frequentItem[i] | frequentItem[j])* resultList.append(frequentItem[i] | frequentItem[j])   
   
 **return** resultList  
  
  
*# In[55]:***def** apriori(dataSet, pctSupport = 0.5): *# minimum support is defaulted to 50%  
 # give us all items in our data* itemSet = allItem(dataSet)  
 item = list(map(set, dataSet)) *# map our dataset into list* numItems = len(item)  
   
 frequentItem = frequentItemset(item,itemSet)  
 frequentItem,supInfo = minSupport(frequentItem,numItems,pctSupport)  
 frequentItem=[frequentItem]; n = 2  
  
 *# it will generate the i-itemset, calculate the frequenty and use to identify minimum support* **while** (len(frequentItem[n-2]) > 0):  
 nItemSet = generateItemset(frequentItem[n-2], n) *#creates i-itemset that met our minimum support* nFrequentItem = frequentItemset(item, nItemSet)  
 nFrequentItem, nSupInfo=minSupport(nFrequentItem, numItems, pctSupport)  
   
 supInfo.update(nSupInfo)  
 frequentItem.append(nFrequentItem)  
 n += 1  
   
 **return** frequentItem, supInfo  
  
  
*# In[56]:***def** generateRules(frequentSet, suppData, pctConf=0.7): *#supportData is a dict coming from scanD* resultList = []  
 **for** i **in** range(0,len(frequentSet)):  
 **for** freqSet **in** frequentSet[i]:  
 itemList = []  
 **for** item **in** freqSet:  
 itemList.append(frozenset([item]))  
   
 *#print(itemList)* **if** (i > 1):  
 rulePerm(freqSet, itemList, suppData, resultList, pctConf)  
 **else**:  
 minConf(freqSet, itemList, suppData, resultList, pctConf)  
 **return** resultList   
  
  
*# In[57]:***def** minConf(X, item, suppData, returnList, pctConf=0.7):  
 prunedItem = []   
 **for** Y **in** item:  
   
 *# the confidence is calculated as Support(X and Y)/Support(X)* pct = suppData[X]/suppData[X-Y]   
   
 **if** pct >= pctConf:   
 pct = round(pct,2) *# round to two decimal places* returnList.append((X-Y, Y, pct)) *# X -> Y; confidence* prunedItem.append(Y)  
   
 **return** prunedItem  
  
  
*# In[58]:***def** rulePerm(X, itemList, suppData, resultList, pctConf=0.7): *# identify permutation for all the assoication rule* num = len(itemList[0])  
   
 *# capture the confidence on a data set {B C D} where B C -> D and B D -> C* **if** (len(X) > (num + 1)):   
 X\_num= generateItemset(itemList, num+1)  
   
 *# calculate the confidence and retrieve ones that meets our threshold* X\_num = minConf(X, X\_num, suppData, resultList, pctConf)  
 **if** (len(X\_num) > 1):  
 rulePerm(X, X\_num, suppData, resultList, pctConf)  
  
  
*# In[59]:***def** printRule(fileName, rules, suppData):  
 result = **''** print(**'Association Rules (%s rules):'** % len(rules))  
 **for** rule **in** rules:  
 *# burgers --> ground beef [20.0%, 80.0%]* assocRule = **', '**.join(rule[0]) + **' --> '** + **', '**.join(rule[1])  
 suppPct = str(100\*suppData[frozenset.union(rule[0],rule[1])])  
 confPct = str(round(100\*rule[2],2))  
 result = **'\t'** + assocRule + **' ['** + suppPct + **'%, '** + confPct + **'%]'** print(result)  
  
  
*# In[60]:***def** aprioriAlgorithm(fileName, pctMinSup, pctMinConf):  
 data=pd.read\_csv(fileName)  
 items=removeNaN(data)  
 *# run apriori algorithm* frequentSet, suppData = apriori(items,pctMinSup)  
 frequentSet = frequentSet[1:] *# 1-itemsets becuase no rule is generated  
   
 # generate association rule* rules= generateRules(frequentSet,suppData, pctMinConf)  
 printRule(fileName, rules, suppData) *# print all rules  
  
  
# In[61]:***def** combinationItem(items, num):  
 items = list(items)  
 numItems = len(items)  
 **if** num > numItems:  
 **return** index = np.arange(num)  
 *# return our first combination* **yield** list(items[i] **for** i **in** index)  
  
 **while True**: *# loop till no combination is available* **for** i **in** reversed(range(num)):  
 **if** index[i] != i + numItems - num:  
 **break  
 else**:  
 **return** index[i] += 1  
  
 **for** j **in** range(i + 1, num):  
 index[j] = index[j - 1] + 1  
  
 **yield** list(items[i] **for** i **in** index)  
  
 *# In[62]:  
  
  
# remove nested list and combine into one list***def** removeNestedList(thisList,result = []):   
 **for** l **in** thisList:   
 **if** type(l) == list:   
 removeNestedList(l, result)   
 **else**:   
 result.append(l)   
 **return** result  
  
  
*# In[63]:***def** bruteForce(fileName, pctMinSup, pctMinConf):  
 data = pd.read\_csv(fileName)   
 items=removeNaN(data)  
 itemSet = allItem(items)  
 itemList = [list(item) **for** item **in** itemSet]  
 output = removeNestedList(itemList,[])  
   
 bruteForceItem = []  
 bruteForceSup = {}  
   
 k = 1 *# number of itemset combination* numItems = len(items)  
 **while True**:  
 *# k combination* combined = [frozenset(x) **for** x **in** combinationItem(output, k)]  
   
 *# get frequent on each combination* frequentItem = frequentItemset(items,combined)  
   
 *# filter out combination with our minimum support* frequentItem,supInfo = minSupport(frequentItem,numItems,pctMinSup)  
   
 **if** frequentItem == []: **break** *# get out once k-itemset did not meet minimum support* k = k+1 *# increment k-itemset* bruteForceSup.update(supInfo)  
 bruteForceItem.append(frequentItem)  
  
 bruteForceItem = bruteForceItem[1:]  
 *# run generate association rule* rules= generateRules(bruteForceItem,bruteForceSup, pctMinConf)  
 printRule(fileName, rules, bruteForceSup) *# print out rules  
  
  
# In[64]:*minimum\_support = .2  
minimum\_confidence = .7  
files = [**'./store\_a\_data.csv'**,**'./store\_b\_data.csv'**,  
 **'./store\_c\_data.csv'**,**'./store\_d\_data.csv'**,  
 **'./store\_e\_data.csv'**]  
  
*# we will loop per file and run Apiori Algorithm and Brute Force  
# and capture the time and print out the association rules..*i = 1  
**for** fileName **in** files:  
 print(**'Database #%d - Filename: %s \n'** % (i,fileName))  
 print(**'Apriori Algorithm:'**)  
 print(**'------------------'**)  
 start\_time = time.time() *# store* aprioriAlgorithm(fileName, minimum\_support, minimum\_confidence)  
 print(**'Total time running: %s seconds'** % str(time.time()-start\_time))  
 print()  
 print(**'Brute Force:'**)  
 print(**'------------------'**)  
 start\_time = time.time()  
 bruteForce(fileName, minimum\_support, minimum\_confidence)  
 print(**'Total time running: %s seconds'** % str(time.time()-start\_time))  
 print()  
 i=i+1

# Output

The minimum support is set to 20 % and minimum confidence is set to 70%.

/Users/jonathanvidal/Desktop/Summer2020/venv/bin/python "/Applications/PyCharm CE.app/Contents/helpers/pydev/pydevd.py" --multiproc --qt-support=auto --client 127.0.0.1 --port 53017 --file /Users/jonathanvidal/Desktop/Summer2020/venv/lib/vidal\_jonathan\_midtermproj.py

pydev debugger: process 9827 is connecting

Connected to pydev debugger (build 192.7142.56)

Database #1 - Filename: ./store\_a\_data.csv

Apriori Algorithm:

------------------

Association Rules (1 rules):

mineral water --> eggs [20.0%, 80.0%]

Total time running: 0.057038068771362305 seconds

Brute Force:

------------------

Association Rules (1 rules):

mineral water --> eggs [20.0%, 80.0%]

Total time running: 0.11385488510131836 seconds

Database #2 - Filename: ./store\_b\_data.csv

Apriori Algorithm:

------------------

Association Rules (2 rules):

spaghetti --> tomatoes [20.0%, 80.0%]

avocado --> chicken [20.0%, 100.0%]

Total time running: 0.028553009033203125 seconds

Brute Force:

------------------

Association Rules (2 rules):

spaghetti --> tomatoes [20.0%, 80.0%]

avocado --> chicken [20.0%, 100.0%]

Total time running: 0.12259507179260254 seconds

Database #3 - Filename: ./store\_c\_data.csv

Apriori Algorithm:

------------------

Association Rules (2 rules):

burgers --> ground beef [20.0%, 80.0%]

spaghetti --> tomatoes [20.0%, 100.0%]

Total time running: 0.029591798782348633 seconds

Brute Force:

------------------

Association Rules (2 rules):

burgers --> ground beef [20.0%, 80.0%]

spaghetti --> tomatoes [20.0%, 100.0%]

Total time running: 0.10416293144226074 seconds

Database #4 - Filename: ./store\_d\_data.csv

Apriori Algorithm:

------------------

Association Rules (124 rules):

soup --> magazines [20.0%, 80.0%]

fresh bread --> ground beef [20.0%, 80.0%]

chocolate --> cookies [30.0%, 75.0%]

milk --> chocolate [30.0%, 100.0%]

chocolate --> milk [30.0%, 75.0%]

milk --> cookies [30.0%, 100.0%]

chocolate --> pancakes [30.0%, 75.0%]

pancakes --> chocolate [30.0%, 100.0%]

pancakes --> cookies [30.0%, 100.0%]

milk --> pancakes [30.0%, 100.0%]

pancakes --> milk [30.0%, 100.0%]

chocolate --> brownies [30.0%, 75.0%]

brownies --> chocolate [30.0%, 100.0%]

brownies --> cookies [30.0%, 100.0%]

milk --> brownies [30.0%, 100.0%]

brownies --> milk [30.0%, 100.0%]

pancakes --> brownies [30.0%, 100.0%]

brownies --> pancakes [30.0%, 100.0%]

pasta --> tomatoes [20.0%, 100.0%]

pasta --> spaghetti [20.0%, 100.0%]

spaghetti --> tomatoes [30.0%, 100.0%]

tomatoes --> spaghetti [30.0%, 86.0%]

pasta, spaghetti --> tomatoes [20.0%, 100.0%]

tomatoes, pasta --> spaghetti [20.0%, 100.0%]

pancakes, milk --> brownies [30.0%, 100.0%]

brownies, milk --> pancakes [30.0%, 100.0%]

brownies, pancakes --> milk [30.0%, 100.0%]

pancakes, cookies --> brownies [30.0%, 100.0%]

brownies, pancakes --> cookies [30.0%, 100.0%]

brownies, cookies --> pancakes [30.0%, 100.0%]

cookies, milk --> brownies [30.0%, 100.0%]

brownies, milk --> cookies [30.0%, 100.0%]

brownies, cookies --> milk [30.0%, 100.0%]

brownies, chocolate --> pancakes [30.0%, 100.0%]

pancakes, chocolate --> brownies [30.0%, 100.0%]

pancakes, brownies --> chocolate [30.0%, 100.0%]

chocolate, milk --> brownies [30.0%, 100.0%]

brownies, milk --> chocolate [30.0%, 100.0%]

brownies, chocolate --> milk [30.0%, 100.0%]

cookies, chocolate --> brownies [30.0%, 100.0%]

brownies, chocolate --> cookies [30.0%, 100.0%]

brownies, cookies --> chocolate [30.0%, 100.0%]

cookies, milk --> pancakes [30.0%, 100.0%]

pancakes, milk --> cookies [30.0%, 100.0%]

pancakes, cookies --> milk [30.0%, 100.0%]

chocolate, milk --> pancakes [30.0%, 100.0%]

pancakes, milk --> chocolate [30.0%, 100.0%]

pancakes, chocolate --> milk [30.0%, 100.0%]

cookies, chocolate --> pancakes [30.0%, 100.0%]

pancakes, chocolate --> cookies [30.0%, 100.0%]

pancakes, cookies --> chocolate [30.0%, 100.0%]

chocolate, milk --> cookies [30.0%, 100.0%]

cookies, milk --> chocolate [30.0%, 100.0%]

cookies, chocolate --> milk [30.0%, 100.0%]

chocolate, milk --> pancakes, cookies [30.0%, 100.0%]

cookies, milk --> pancakes, chocolate [30.0%, 100.0%]

cookies, chocolate --> pancakes, milk [30.0%, 100.0%]

pancakes, milk --> cookies, chocolate [30.0%, 100.0%]

pancakes, chocolate --> cookies, milk [30.0%, 100.0%]

pancakes, cookies --> chocolate, milk [30.0%, 100.0%]

milk --> pancakes, cookies, chocolate [30.0%, 100.0%]

chocolate --> pancakes, cookies, milk [30.0%, 75.0%]

pancakes --> cookies, chocolate, milk [30.0%, 100.0%]

chocolate, milk --> brownies, cookies [30.0%, 100.0%]

cookies, milk --> brownies, chocolate [30.0%, 100.0%]

cookies, chocolate --> brownies, milk [30.0%, 100.0%]

brownies, milk --> cookies, chocolate [30.0%, 100.0%]

brownies, chocolate --> cookies, milk [30.0%, 100.0%]

brownies, cookies --> chocolate, milk [30.0%, 100.0%]

milk --> brownies, cookies, chocolate [30.0%, 100.0%]

chocolate --> brownies, cookies, milk [30.0%, 75.0%]

brownies --> cookies, chocolate, milk [30.0%, 100.0%]

pancakes, chocolate --> brownies, cookies [30.0%, 100.0%]

cookies, chocolate --> brownies, pancakes [30.0%, 100.0%]

pancakes, cookies --> brownies, chocolate [30.0%, 100.0%]

brownies, chocolate --> pancakes, cookies [30.0%, 100.0%]

brownies, pancakes --> cookies, chocolate [30.0%, 100.0%]

brownies, cookies --> pancakes, chocolate [30.0%, 100.0%]

chocolate --> brownies, cookies, pancakes [30.0%, 75.0%]

pancakes --> brownies, cookies, chocolate [30.0%, 100.0%]

brownies --> pancakes, cookies, chocolate [30.0%, 100.0%]

pancakes, milk --> brownies, cookies [30.0%, 100.0%]

cookies, milk --> brownies, pancakes [30.0%, 100.0%]

pancakes, cookies --> brownies, milk [30.0%, 100.0%]

brownies, milk --> pancakes, cookies [30.0%, 100.0%]

brownies, pancakes --> cookies, milk [30.0%, 100.0%]

brownies, cookies --> pancakes, milk [30.0%, 100.0%]

milk --> brownies, cookies, pancakes [30.0%, 100.0%]

pancakes --> brownies, cookies, milk [30.0%, 100.0%]

brownies --> pancakes, cookies, milk [30.0%, 100.0%]

chocolate, milk --> brownies, pancakes [30.0%, 100.0%]

pancakes, milk --> brownies, chocolate [30.0%, 100.0%]

pancakes, chocolate --> brownies, milk [30.0%, 100.0%]

brownies, milk --> pancakes, chocolate [30.0%, 100.0%]

brownies, chocolate --> pancakes, milk [30.0%, 100.0%]

brownies, pancakes --> chocolate, milk [30.0%, 100.0%]

milk --> chocolate, brownies, pancakes [30.0%, 100.0%]

chocolate --> brownies, pancakes, milk [30.0%, 75.0%]

pancakes --> brownies, chocolate, milk [30.0%, 100.0%]

brownies --> pancakes, chocolate, milk [30.0%, 100.0%]

pancakes, chocolate, milk --> brownies, cookies [30.0%, 100.0%]

pancakes, cookies, milk --> brownies, chocolate [30.0%, 100.0%]

pancakes, cookies, chocolate --> brownies, milk [30.0%, 100.0%]

cookies, chocolate, milk --> brownies, pancakes [30.0%, 100.0%]

brownies, pancakes, milk --> cookies, chocolate [30.0%, 100.0%]

pancakes, brownies, chocolate --> cookies, milk [30.0%, 100.0%]

brownies, chocolate, milk --> pancakes, cookies [30.0%, 100.0%]

brownies, cookies, pancakes --> chocolate, milk [30.0%, 100.0%]

brownies, cookies, milk --> pancakes, chocolate [30.0%, 100.0%]

brownies, cookies, chocolate --> pancakes, milk [30.0%, 100.0%]

pancakes, milk --> brownies, cookies, chocolate [30.0%, 100.0%]

pancakes, chocolate --> brownies, cookies, milk [30.0%, 100.0%]

chocolate, milk --> brownies, cookies, pancakes [30.0%, 100.0%]

pancakes, cookies --> brownies, chocolate, milk [30.0%, 100.0%]

cookies, milk --> pancakes, brownies, chocolate [30.0%, 100.0%]

cookies, chocolate --> brownies, pancakes, milk [30.0%, 100.0%]

brownies, pancakes --> cookies, chocolate, milk [30.0%, 100.0%]

brownies, milk --> pancakes, cookies, chocolate [30.0%, 100.0%]

brownies, chocolate --> pancakes, cookies, milk [30.0%, 100.0%]

brownies, cookies --> pancakes, chocolate, milk [30.0%, 100.0%]

pancakes --> brownies, cookies, chocolate, milk [30.0%, 100.0%]

milk --> brownies, cookies, pancakes, chocolate [30.0%, 100.0%]

chocolate --> brownies, cookies, pancakes, milk [30.0%, 75.0%]

brownies --> pancakes, cookies, chocolate, milk [30.0%, 100.0%]

Total time running: 0.030452966690063477 seconds

Brute Force:

------------------

Association Rules (124 rules):

soup --> magazines [20.0%, 80.0%]

fresh bread --> ground beef [20.0%, 80.0%]

spaghetti --> tomatoes [30.0%, 100.0%]

tomatoes --> spaghetti [30.0%, 86.0%]

pasta --> spaghetti [20.0%, 100.0%]

pasta --> tomatoes [20.0%, 100.0%]

brownies --> pancakes [30.0%, 100.0%]

pancakes --> brownies [30.0%, 100.0%]

milk --> brownies [30.0%, 100.0%]

brownies --> milk [30.0%, 100.0%]

milk --> pancakes [30.0%, 100.0%]

pancakes --> milk [30.0%, 100.0%]

brownies --> cookies [30.0%, 100.0%]

pancakes --> cookies [30.0%, 100.0%]

milk --> cookies [30.0%, 100.0%]

chocolate --> brownies [30.0%, 75.0%]

brownies --> chocolate [30.0%, 100.0%]

chocolate --> pancakes [30.0%, 75.0%]

pancakes --> chocolate [30.0%, 100.0%]

milk --> chocolate [30.0%, 100.0%]

chocolate --> milk [30.0%, 75.0%]

chocolate --> cookies [30.0%, 75.0%]

pasta, spaghetti --> tomatoes [20.0%, 100.0%]

tomatoes, pasta --> spaghetti [20.0%, 100.0%]

brownies, milk --> pancakes [30.0%, 100.0%]

pancakes, milk --> brownies [30.0%, 100.0%]

pancakes, brownies --> milk [30.0%, 100.0%]

brownies, cookies --> pancakes [30.0%, 100.0%]

pancakes, brownies --> cookies [30.0%, 100.0%]

pancakes, cookies --> brownies [30.0%, 100.0%]

cookies, milk --> brownies [30.0%, 100.0%]

brownies, milk --> cookies [30.0%, 100.0%]

brownies, cookies --> milk [30.0%, 100.0%]

cookies, milk --> pancakes [30.0%, 100.0%]

pancakes, milk --> cookies [30.0%, 100.0%]

pancakes, cookies --> milk [30.0%, 100.0%]

brownies, chocolate --> pancakes [30.0%, 100.0%]

pancakes, brownies --> chocolate [30.0%, 100.0%]

pancakes, chocolate --> brownies [30.0%, 100.0%]

chocolate, milk --> brownies [30.0%, 100.0%]

brownies, milk --> chocolate [30.0%, 100.0%]

brownies, chocolate --> milk [30.0%, 100.0%]

chocolate, milk --> pancakes [30.0%, 100.0%]

pancakes, milk --> chocolate [30.0%, 100.0%]

pancakes, chocolate --> milk [30.0%, 100.0%]

cookies, chocolate --> brownies [30.0%, 100.0%]

brownies, chocolate --> cookies [30.0%, 100.0%]

brownies, cookies --> chocolate [30.0%, 100.0%]

cookies, chocolate --> pancakes [30.0%, 100.0%]

pancakes, chocolate --> cookies [30.0%, 100.0%]

pancakes, cookies --> chocolate [30.0%, 100.0%]

chocolate, milk --> cookies [30.0%, 100.0%]

cookies, milk --> chocolate [30.0%, 100.0%]

cookies, chocolate --> milk [30.0%, 100.0%]

brownies, milk --> pancakes, cookies [30.0%, 100.0%]

cookies, milk --> pancakes, brownies [30.0%, 100.0%]

brownies, cookies --> pancakes, milk [30.0%, 100.0%]

pancakes, milk --> brownies, cookies [30.0%, 100.0%]

pancakes, brownies --> cookies, milk [30.0%, 100.0%]

pancakes, cookies --> brownies, milk [30.0%, 100.0%]

milk --> pancakes, cookies, brownies [30.0%, 100.0%]

brownies --> pancakes, cookies, milk [30.0%, 100.0%]

pancakes --> brownies, cookies, milk [30.0%, 100.0%]

brownies, milk --> pancakes, chocolate [30.0%, 100.0%]

chocolate, milk --> pancakes, brownies [30.0%, 100.0%]

brownies, chocolate --> pancakes, milk [30.0%, 100.0%]

pancakes, milk --> brownies, chocolate [30.0%, 100.0%]

pancakes, brownies --> chocolate, milk [30.0%, 100.0%]

pancakes, chocolate --> brownies, milk [30.0%, 100.0%]

milk --> pancakes, chocolate, brownies [30.0%, 100.0%]

brownies --> pancakes, chocolate, milk [30.0%, 100.0%]

chocolate --> pancakes, brownies, milk [30.0%, 75.0%]

pancakes --> brownies, chocolate, milk [30.0%, 100.0%]

brownies, chocolate --> pancakes, cookies [30.0%, 100.0%]

brownies, cookies --> pancakes, chocolate [30.0%, 100.0%]

cookies, chocolate --> pancakes, brownies [30.0%, 100.0%]

pancakes, brownies --> cookies, chocolate [30.0%, 100.0%]

pancakes, chocolate --> brownies, cookies [30.0%, 100.0%]

pancakes, cookies --> brownies, chocolate [30.0%, 100.0%]

brownies --> pancakes, cookies, chocolate [30.0%, 100.0%]

chocolate --> pancakes, cookies, brownies [30.0%, 75.0%]

pancakes --> brownies, cookies, chocolate [30.0%, 100.0%]

chocolate, milk --> brownies, cookies [30.0%, 100.0%]

cookies, milk --> brownies, chocolate [30.0%, 100.0%]

cookies, chocolate --> brownies, milk [30.0%, 100.0%]

brownies, milk --> cookies, chocolate [30.0%, 100.0%]

brownies, chocolate --> cookies, milk [30.0%, 100.0%]

brownies, cookies --> chocolate, milk [30.0%, 100.0%]

milk --> brownies, cookies, chocolate [30.0%, 100.0%]

chocolate --> brownies, cookies, milk [30.0%, 75.0%]

brownies --> cookies, chocolate, milk [30.0%, 100.0%]

chocolate, milk --> pancakes, cookies [30.0%, 100.0%]

cookies, milk --> pancakes, chocolate [30.0%, 100.0%]

cookies, chocolate --> pancakes, milk [30.0%, 100.0%]

pancakes, milk --> cookies, chocolate [30.0%, 100.0%]

pancakes, chocolate --> cookies, milk [30.0%, 100.0%]

pancakes, cookies --> chocolate, milk [30.0%, 100.0%]

milk --> pancakes, cookies, chocolate [30.0%, 100.0%]

chocolate --> pancakes, cookies, milk [30.0%, 75.0%]

pancakes --> cookies, chocolate, milk [30.0%, 100.0%]

pancakes, chocolate, milk --> brownies, cookies [30.0%, 100.0%]

pancakes, cookies, milk --> brownies, chocolate [30.0%, 100.0%]

pancakes, cookies, chocolate --> brownies, milk [30.0%, 100.0%]

cookies, chocolate, milk --> brownies, pancakes [30.0%, 100.0%]

brownies, pancakes, milk --> cookies, chocolate [30.0%, 100.0%]

pancakes, brownies, chocolate --> cookies, milk [30.0%, 100.0%]

brownies, chocolate, milk --> pancakes, cookies [30.0%, 100.0%]

brownies, cookies, pancakes --> chocolate, milk [30.0%, 100.0%]

brownies, cookies, milk --> pancakes, chocolate [30.0%, 100.0%]

brownies, cookies, chocolate --> pancakes, milk [30.0%, 100.0%]

pancakes, milk --> brownies, cookies, chocolate [30.0%, 100.0%]

pancakes, chocolate --> brownies, cookies, milk [30.0%, 100.0%]

chocolate, milk --> brownies, cookies, pancakes [30.0%, 100.0%]

pancakes, cookies --> brownies, chocolate, milk [30.0%, 100.0%]

cookies, milk --> pancakes, brownies, chocolate [30.0%, 100.0%]

cookies, chocolate --> brownies, pancakes, milk [30.0%, 100.0%]

brownies, pancakes --> cookies, chocolate, milk [30.0%, 100.0%]

brownies, milk --> pancakes, cookies, chocolate [30.0%, 100.0%]

brownies, chocolate --> pancakes, cookies, milk [30.0%, 100.0%]

brownies, cookies --> pancakes, chocolate, milk [30.0%, 100.0%]

pancakes --> brownies, cookies, chocolate, milk [30.0%, 100.0%]

milk --> brownies, cookies, pancakes, chocolate [30.0%, 100.0%]

chocolate --> brownies, cookies, pancakes, milk [30.0%, 75.0%]

brownies --> pancakes, cookies, chocolate, milk [30.0%, 100.0%]

Total time running: 20.886988878250122 seconds

Database #5 - Filename: ./store\_e\_data.csv

Apriori Algorithm:

------------------

Association Rules (16 rules):

cake --> pancakes [20.0%, 80.0%]

spaghetti --> pasta [20.0%, 80.0%]

pasta --> spaghetti [20.0%, 100.0%]

cake --> brownies [25.0%, 100.0%]

milk --> brownies [25.0%, 71.0%]

ground beef --> eggs [20.0%, 100.0%]

pancakes --> brownies [35.0%, 100.0%]

brownies --> pancakes [35.0%, 70.0%]

mineral water --> salmon [30.0%, 86.0%]

salmon --> mineral water [30.0%, 86.0%]

soup --> brownies [30.0%, 75.0%]

brownies, cake --> pancakes [20.0%, 80.0%]

pancakes, cake --> brownies [20.0%, 100.0%]

brownies, milk --> soup [20.0%, 80.0%]

soup, milk --> brownies [20.0%, 100.0%]

soup, pancakes --> brownies [20.0%, 100.0%]

Total time running: 0.03652620315551758 seconds

Brute Force:

------------------

Association Rules (16 rules):

cake --> pancakes [20.0%, 80.0%]

spaghetti --> pasta [20.0%, 80.0%]

pasta --> spaghetti [20.0%, 100.0%]

cake --> brownies [25.0%, 100.0%]

milk --> brownies [25.0%, 71.0%]

ground beef --> eggs [20.0%, 100.0%]

mineral water --> salmon [30.0%, 86.0%]

salmon --> mineral water [30.0%, 86.0%]

soup --> brownies [30.0%, 75.0%]

brownies --> pancakes [35.0%, 70.0%]

pancakes --> brownies [35.0%, 100.0%]

brownies, cake --> pancakes [20.0%, 80.0%]

pancakes, cake --> brownies [20.0%, 100.0%]

brownies, milk --> soup [20.0%, 80.0%]

soup, milk --> brownies [20.0%, 100.0%]

soup, pancakes --> brownies [20.0%, 100.0%]

Total time running: 0.5040731430053711 seconds

Process finished with exit code 0