



Implementing Apriori algorithm in Python

Last Updated : 11 Jan, 2023

Prerequisites: [Apriori Algorithm](#)

Apriori Algorithm is a Machine Learning algorithm which is used to gain insight into the structured relationships between different items involved. The most prominent practical application of the algorithm is to recommend products based on the products already present in the user's cart. **Walmart** especially has made great use of the algorithm in suggesting products to its users.

Dataset : [Groceries data](#)

Implementation of algorithm in Python:

Step 1: Importing the required libraries

Python3

```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

Step 2: Loading and exploring the data

Create a Perfect Portfolio as a Content ...

Python3

```
# Changing the working location to the location of the file
```

```
cd C:\Users\Dev\Desktop\Kaggle\Apriori Algorithm
```

```
# Loading the Data
```

```
data = pd.read_excel('Online_Retail.xlsx')
```

```
data.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Python3

```
# Exploring the columns of the data
```

```
data.columns
```

```
Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',  
      'UnitPrice', 'CustomerID', 'Country'],  
      dtype='object')
```

Python3

```
# Exploring the different regions of transactions
```

```
data.Country.unique()
```

```
array(['United Kingdom', 'France', 'Australia', 'Netherlands', 'Germany',  
      'Norway', 'EIRE', 'Switzerland', 'Spain', 'Poland', 'Portugal',  
      'Italy', 'Belgium', 'Lithuania', 'Japan', 'Iceland',  
      'Channel Islands', 'Denmark', 'Cyprus', 'Sweden', 'Austria',  
      'Israel', 'Finland', 'Bahrain', 'Greece', 'Hong Kong', 'Singapore',  
      'Lebanon', 'United Arab Emirates', 'Saudi Arabia',  
      'Czech Republic', 'Canada', 'Unspecified', 'Brazil', 'USA',  
      'European Community', 'Malta', 'RSA'], dtype=object)
```

Step 3: Cleaning the Data

Python3

```
# Stripping extra spaces in the description
```

```
data['Description'] = data['Description'].str.strip()
```

```
# Dropping the rows without any invoice number
```

```
data.dropna(axis = 0, subset = ['InvoiceNo'], inplace = True)
```

```
data['InvoiceNo'] = data['InvoiceNo'].astype('str')
```

```
# Dropping all transactions which were done on credit
```

```
data = data[~data['InvoiceNo'].str.contains('C')]
```

Step 4: Splitting the data according to the region of transaction

Python3

```
# Transactions done in France
```

```
basket_France = (data[data['Country'] == "France"]  
                 .groupby(['InvoiceNo', 'Description'])['Quantity']  
                 .sum().unstack().reset_index().fillna(0)  
                 .set_index('InvoiceNo'))
```

```
# Transactions done in the United Kingdom
```

```
basket_UK = (data[data['Country'] == "United Kingdom"]
```

```
.groupby(['InvoiceNo', 'Description'])['Quantity']  
.sum().unstack().reset_index().fillna(0)  
.set_index('InvoiceNo'))  
  
# Transactions done in Portugal  
basket_Por = (data[data['Country'] == "Portugal"]  
              .groupby(['InvoiceNo', 'Description'])['Quantity']  
              .sum().unstack().reset_index().fillna(0)  
              .set_index('InvoiceNo'))  
  
basket_Sweden = (data[data['Country'] == "Sweden"]  
                 .groupby(['InvoiceNo', 'Description'])['Quantity']  
                 .sum().unstack().reset_index().fillna(0)  
                 .set_index('InvoiceNo'))
```

Step 5: Hot encoding the Data

Python3

```
# Defining the hot encoding function to make the data suitable  
# for the concerned libraries  
def hot_encode(x):  
    if(x<= 0):  
        return 0  
    if(x>= 1):  
        return 1  
  
# Encoding the datasets  
basket_encoded = basket_France.applymap(hot_encode)  
basket_France = basket_encoded  
  
basket_encoded = basket_UK.applymap(hot_encode)  
basket_UK = basket_encoded  
  
basket_encoded = basket_Por.applymap(hot_encode)  
basket_Por = basket_encoded  
  
basket_encoded = basket_Sweden.applymap(hot_encode)  
basket_Sweden = basket_encoded
```

Step 6: Building the models and analyzing the results

a) France:

Python3

```
# Building the model
frq_items = apriori(basket_France, min_support = 0.05, use_colnames = True)

# Collecting the inferred rules in a dataframe
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
print(rules.head())
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
44	(JUMBO BAG WOODLAND ANIMALS)	(POSTAGE)	0.076531	0.765306	0.076531	1.000	1.306667	0.017961	inf
258	(PLASTERS IN TIN CIRCUS PARADE, RED TOADSTOOL ...)	(POSTAGE)	0.051020	0.765306	0.051020	1.000	1.306667	0.011974	inf
270	(PLASTERS IN TIN WOODLAND ANIMALS, RED TOADSTO...)	(POSTAGE)	0.053571	0.765306	0.053571	1.000	1.306667	0.012573	inf
301	(SET/6 RED SPOTTY PAPER CUPS, SET/20 RED RETRO...)	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.127551	0.099490	0.975	7.644000	0.086474	34.897959
302	(SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...)	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.137755	0.099490	0.975	7.077778	0.085433	34.489796

From the above output, it can be seen that paper cups and paper and plates are bought together in France. This is because the French have a culture of having a get-together with their friends and family atleast once a week. Also, since the French government has banned the use of plastic in the country, the people have to purchase the paper-based alternatives.

b) United Kingdom:

Python3

```
frq_items = apriori(basket_UK, min_support = 0.01, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
print(rules.head())
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
116	(BEADED CRYSTAL HEART PINK ON STICK)	(DOTCOM POSTAGE)	0.011036	0.037928	0.010768	0.975728	25.725872	0.010349	39.637371
2019	(SUKI SHOULDER BAG, JAM MAKING SET PRINTED)	(DOTCOM POSTAGE)	0.011625	0.037928	0.011196	0.963134	25.393807	0.010755	26.096206
2296	(HERB MARKER THYME, HERB MARKER MINT)	(HERB MARKER ROSEMARY)	0.010714	0.012375	0.010232	0.955000	77.173095	0.010099	21.947227
2302	(HERB MARKER PARSLEY, HERB MARKER ROSEMARY)	(HERB MARKER THYME)	0.011089	0.012321	0.010553	0.951691	77.240055	0.010417	20.444951
2300	(HERB MARKER THYME, HERB MARKER PARSLEY)	(HERB MARKER ROSEMARY)	0.011089	0.012375	0.010553	0.951691	76.905682	0.010416	20.443842

If the rules for British transactions are analyzed a little deeper, it is seen that the British people buy different colored tea-plates together. A reason behind this may be because typically the British enjoy tea very much and often collect different colored tea-plates for different occasions.

c) Portugal:

Python3

```
frq_items = apriori(basket_Por, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
print(rules.head())
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1170	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET 12 COLOUR PENCILS SPACEBOY)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf
1171	(SET 12 COLOUR PENCILS SPACEBOY)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf
1172	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET OF 4 KNICK KNACK TINS LONDON)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf
1173	(SET OF 4 KNICK KNACK TINS LONDON)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf
1174	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET OF 4 KNICK KNACK TINS POPPIES)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf

On analyzing the association rules for Portuguese transactions, it is observed that Tiffin sets (Knick Knack Tins) and color pencils. These two products

typically belong to a primary school going kid. These two products are required by children in school to carry their lunch and for creative work respectively and hence are logically make sense to be paired together.

d) **Sweden:**

Python3

```
frq_items = apriori(basket_Sweden, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, metric = "lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending = [False, False])
print(rules.head())
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(12 PENCILS SMALL TUBE SKULL)	(PACK OF 72 SKULL CAKE CASES)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	inf
1	(PACK OF 72 SKULL CAKE CASES)	(12 PENCILS SMALL TUBE SKULL)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	inf
4	(36 DOILIES DOLLY GIRL)	(ASSORTED BOTTLE TOP MAGNETS)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	inf
5	(ASSORTED BOTTLE TOP MAGNETS)	(36 DOILIES DOLLY GIRL)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	inf
180	(CHILDRENS CUTLERY DOLLY GIRL)	(CHILDRENS CUTLERY CIRCUS PARADE)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	inf

On analyzing the above rules, it is found that boys' and girls' cutlery are paired together. This makes practical sense because when a parent goes shopping for cutlery for his/her children, he/she would want the product to be a little customized according to the kid's wishes.

"This course is very well structured and easy to learn. Anyone with zero experience of data science, python or ML can learn from this. This course makes things so easy that anybody can learn on their own. It's helping me a lot. Thanks for creating such a great course."- **Ayushi Jain | Placed at Microsoft**

Now's your chance to unlock high-earning job opportunities as a [Data Scientist](#)! Join our [Complete Machine Learning & Data Science Program](#) and get a 360-degree learning experience mentored by industry experts.

Get hands on practice with **40+ Industry Projects**, **regular doubt solving sessions**, and much more. [Register for the Program today!](#)

A Alind...

20

Next Article

Python | Decision tree implementation

Similar Reads

Apriori Algorithm

Prerequisite - Frequent Item set in Data set (Association Rule Mining) Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent...

5 min read

Implementing the AdaBoost Algorithm From Scratch

In this article, we will learn about the AdaBoost classifier and its practical implementation over a dataset. AdaBoost algorithm falls under ensemble...

6 min read

Implementing DBSCAN algorithm using Sklearn

Prerequisites: DBSCAN Algorithm Density Based Spatial Clustering of Applications with Noise(DBSCAN) is a clustering algorithm which was...

3 min read

Implementing PCA in Python with scikit-learn

In this article, we will learn about PCA (Principal Component Analysis) in Python with scikit-learn. Let's start our learning step by step. WHY PCA? Wh...

4 min read

Implementing Rich getting Richer phenomenon using Barabasi Albert...

Prerequisite- Introduction to Social Networks, Barabasi Albert Graph In social networks, there is a phenomenon called Rich getting Richer also known as...

4 min read

[View More Articles](#)

Article Tags :

[python](#)

[Machine Learning](#)

Practice Tags :

[Machine Learning](#)

[python](#)



A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305



Company

[About Us](#)

[Legal](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

Explore

[Hack-A-Thons](#)

[GfG Weekly Contest](#)

[DSA in JAVA/C++](#)

[Master System Design](#)

[Master CP](#)

Advertise with us
GFG Corporate Solution
Placement Training Program

GeeksforGeeks Videos
Geeks Community

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning Tutorial
ML Maths
Data Visualisation Tutorial
Pandas Tutorial
NumPy Tutorial
NLP Tutorial
Deep Learning Tutorial

HTML & CSS

HTML
CSS
Web Templates
CSS Frameworks
Bootstrap
Tailwind CSS
SASS
LESS
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Competitive Programming

Top DS or Algo for CP
Top 50 Tree
Top 50 Graph
Top 50 Array
Top 50 String
Top 50 DP
Top 15 Websites for CP

System Design

High Level Design
Low Level Design
UML Diagrams

JavaScript

JavaScript Examples
TypeScript
ReactJS

Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Preparation Corner

Company-Wise Recruitment Process
Resume Templates
Aptitude Preparation
Puzzles
Company-Wise Preparation

Management & Finance

Management
HR Management
Finance
Organisational Behaviour
Marketing

More Tutorials

Software Development
Software Testing
Product Management
SEO - Search Engine Optimization
Linux
Excel
All Cheatsheets

NextJS
AngularJS
NodeJS
Lodash
Web Browser

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
World GK

Free Online Tools

Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved