# Association Rules

**Instructions:**

Please share your answers filled in-line in the word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: Vishvash C Batch ID:** 23012024

**Topic: Association Rules**

# 

**Guidelines:**

**1. An assignment submission is considered complete only when the correct and executable code(s) is submitted along with the documentation explaining the method and results. Failing to submit either of those will be considered an invalid submission and will not be considered a correct submission.**

**2. Ensure that you submit your assignments correctly and in full. Resubmission is not allowed.**

**3. Post the submission you can evaluate your work by referring to the keys provided. (will be available only post the submission).**

Hints:

1. Business Problem
   1. **What is the business objective?**
   2. **Are there any constraints?**
2. Work on each feature of the dataset to create a data dictionary as displayed in the below image**:**



1. Data Pre-processing
   1. Data Cleaning, Feature Engineering, etc.
2. Model Building

4.1 Application of Apriori Algorithm

* 1. Build the most frequent item sets and plot the rules
  2. Work on Codes

5. Deployment

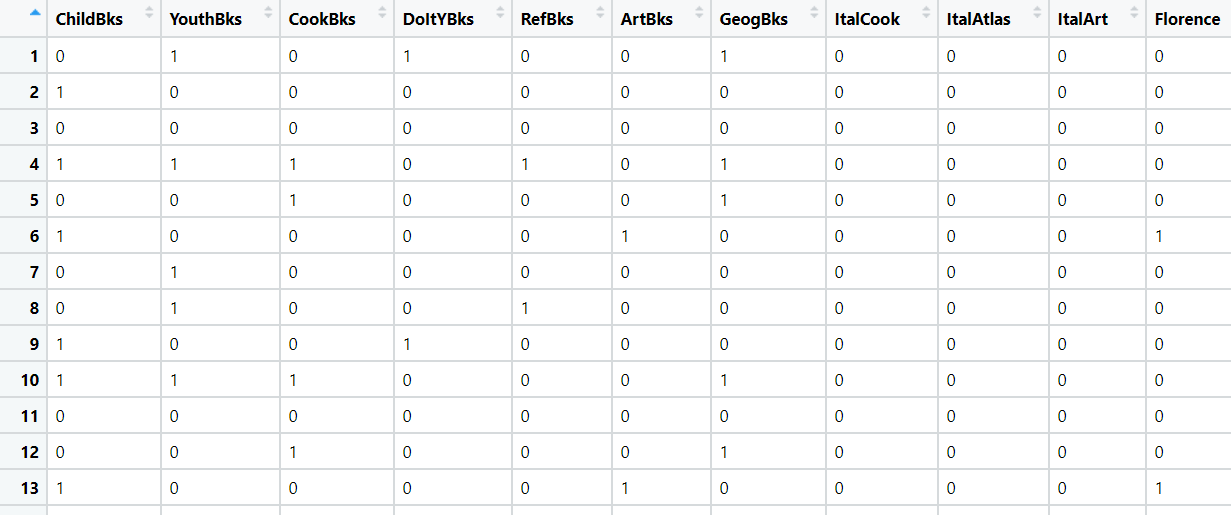
5.1 Deploy solutions

6. Write about the benefits/impact of the solution - in what way does the business (client) benefit from the solution provided?

**Problem Statement: -**

Q1. Kitabi Duniya, a famous bookstore in India, was established before Independence, the growth of the company was incremental year by year, but due to the online selling of books and widespread Internet access, its annual growth started to collapse. As a Data Scientist, you must help this heritage bookstore gain its popularity back and increase the footfall of customers and provide ways to improve the business exponentially to an expected value at a 25% improvement of the current rate. Apply the pattern mining techniques (Association Rules Algorithm) to identify ways to improve sales. Explain the rules (patterns) identified, and visually represent the rules in graphs for a clear understanding of the solution.

**1.) Data: Books.csv**



**Business Objective:**

**Business Constraints:**

**Success Criteria:**

**Business Success Criteria:**

**ML Success Criteria:**

**Economic Success Criteria:**

**Questions to Ignite your Thinking process:**

Q1. Which library/package is used for the Association rules algorithm?

The mlxtend library in Python is commonly used for Association rules algorithms.

Q2. Which functions are used in the Association rules algorithm?

Functions such as apriori and association\_rules are commonly used for generating association rules from transactional datasets.

Q3. What is the keyword used to import any package to the Python session’s memory?

The import keyword is used to import any package to the Python session’s memory.

Q4. What type of data is usually worked in Association rules?

Transactional data or itemsets data is usually worked with in Association rules, where each transaction consists of a set of items.

Q5. Association rules are also named as

Association rules are also named as Market Basket Analysis/ Affinity analysis/ Relationship analysis.

Q6. What is the IF part called in an Association rule?

The IF part in an Association rule is called the antecedent.

Q7. What is the THEN part called in an Association rule?

The THEN part in an Association rule is called the consequent.

Q8. In which sector is the Association rules algorithm mainly used?

The Association rules algorithm is mainly used in retail and e-commerce sectors.

Q9. What do slotting fees mean?

Slotting fees refer to the charges paid by suppliers to retailers for securing a slot or space on store shelves for their products.

Q10. How is Support calculated in the Association rules algorithm?

Support is calculated as the ratio of the number of transactions containing both the antecedent and the consequent to the total number of transactions.

Q11. What is the drawback of Support in Association rules algorithm?

The drawback of Support is that it doesn't consider the frequency of individual items independently, which may lead to inaccurate rule generation.

Q12. To remove the infrequent items from data which algorithm is used?

To remove infrequent items from data, the Apriori algorithm is commonly used in Association rules.

Q13. The drawback of Support is captured by

The drawback of Support is captured by the inability to consider the importance of individual items independently.

Q14. How is confidence calculated in the Association rules algorithm?

Confidence is calculated as the ratio of the number of transactions containing both the antecedent and the consequent to the number of transactions containing the antecedent.

Q15. What is the drawback of Confidence in Association rules algorithm?

Biased confidence as high support may lead to high confidence

Q16. How is the Lift ratio calculated?

The Lift ratio is calculated as the ratio of confidence to the benchmark confidence

(#A&C/#A)/(#C/#T)

Q17. What is the threshold value of the Lift ratio of a rule to declare it as a good Association rule?

Typically, Lift>1 indicates a rule that is useful in finding consequent item sets. However, the specific threshold for declaring a rule as "good" may vary depending on the context and domain knowledge.



**Associate Rule:**

# -\*- coding: utf-8 -\*-

"""

Created on Mon Mar 25 22:00:46 2024

@author: Lenovo

"""

'''

# Data Mining Unsupervised Learning / Descriptive Modeling - Association Rule Mining

# Problem Statement

Kitabi Duniya, a famous bookstore in India, was established before Independence, the growth of the company was incremental year by year, but due to the online selling of books and widespread Internet access, its annual growth started to collapse. As a Data Scientist, you must help this heritage bookstore gain its popularity back and increase the footfall of customers and provide ways to improve the business exponentially to an expected value at a 25% improvement of the current rate. Apply the pattern mining techniques (Association Rules Algorithm) to identify ways to improve sales. Explain the rules (patterns) identified, and visually represent the rules in graphs for a clear understanding of the solution.

# `CRISP-ML(Q)` process model describes six phases:

#

# 1. Business and Data Understanding

# 2. Data Preparation

# 3. Model Building

# 4. Model Evaluation

# 5. Deployment

# 6. Monitoring and Maintenance

# \*\*Objective(s):\*\* Maximize Sales

#

# \*\*Constraints:\*\* Minimize Marketing Cost

# \*\*Success Criteria\*\*

#

# - \*\*Business Success Criteria\*\*: Improve the sales in Retail Store

by 15% - 20%

#

# - \*\*ML Success Criteria\*\*: Accuracy : NA;

Performance : Complete processing within 5 mins on every quarter data

#

# - \*\*Economic Success Criteria\*\*: Increase the Store profits by

atleast 15%

#

# \*\*Proposed Plan:\*\*

# Identify the Association between the books being purchased by the customers

from the store

# ## Data Collection

# Data:

# The daily transactions made by the customers are captured by the store.

#

# Description:

# A total of 2001 transactions data captured for the month.

'''

# Mlxtend (machine learning extensions) is a Python library of useful tools for

# the day-to-day data science tasks.

# pip install mlxtend

# Install the required packages if not available

import pandas as pd

import matplotlib.pyplot as plt

from mlxtend.frequent\_patterns import apriori, association\_rules

from mlxtend.preprocessing import TransactionEncoder

from sqlalchemy import create\_engine, text

import pickle

engine = create\_engine("mysql+pymysql://{user}:{pw}@localhost/{db}".format(user = "root", pw = "1234", db = "retail")) # database

# Read csv file

data = pd.read\_csv(r"C:/Users/Lenovo/Downloads/Study material/Data Science/Association Rules/Assignments/Data Set/book.csv" )

data.head()

# Load the data into MySQL DB

data.to\_sql('bookshop', con = engine, if\_exists = 'replace', chunksize = 1000, index = False)

##############

# Read data from database

sql = text('select \* from bookshop;')

transf\_df = pd.read\_sql\_query(sql, con = engine.connect())

transf\_df.head()

# Suppress the Warnings

import warnings

warnings.filterwarnings("ignore")

### Elementary Analysis ###

# Most popular items

count = transf\_df.loc[:, :].sum()

# Generates a series

pop\_item = count.sort\_values(axis = 0, ascending = False).head(10)

# Convert the series into a dataframe

pop\_item = pop\_item.to\_frame() # type casting

# Reset Index

pop\_item = pop\_item.reset\_index()

pop\_item

pop\_item = pop\_item.rename(columns = {"index": "items", 0: "count"})

pop\_item

# Data Visualization

# get\_ipython().run\_line\_magic('matplotlib', 'inline')

plt.rcParams['figure.figsize'] = (10, 6) # rc stands for runtime configuration

plt.style.use('dark\_background')

pop\_item.plot.barh()

plt.title('Most popular items')

plt.gca().invert\_yaxis() # gca means "get current axes"

help(apriori)

# Itemsets

frequent\_itemsets = apriori(transf\_df, min\_support = 0.035, max\_len = 3, use\_colnames = True)

frequent\_itemsets

# Most frequent itemsets based on support

frequent\_itemsets.sort\_values('support', ascending = False, inplace = True)

frequent\_itemsets

# Association Rules

rules = association\_rules(frequent\_itemsets, metric = "lift", min\_threshold = 1)

rules.head(10)

rules.sort\_values('lift', ascending = False).head(10)

# Handling Profusion of Rules (Duplication elimination)

def to\_list(i):

return (sorted(list(i)))

# Sort the items in Antecedents and Consequents based on Alphabetical order

ma\_X = rules.antecedents.apply(to\_list) + rules.consequents.apply(to\_list)

# Sort the merged list of items - transactions

ma\_X = ma\_X.apply(sorted)

rules\_sets = list(ma\_X)

# No duplication of transactions

unique\_rules\_sets = [list(m) for m in set(tuple(i) for i in rules\_sets)]

# Capture the index of unique item sets

index\_rules = []

for i in unique\_rules\_sets:

index\_rules.append(rules\_sets.index(i))

index\_rules

# Rules without any redudancy

rules\_no\_redundancy = rules.iloc[index\_rules, :]

rules\_no\_redundancy

# Sorted list and top 10 rules

rules10 = rules\_no\_redundancy.sort\_values('lift', ascending = False).head(10)

rules10

rules10.plot(x = "support", y = "confidence", c = rules10.lift,

kind = "scatter", s = 12, cmap = plt.cm.coolwarm)

rules10.info()

# Store the rules on to SQL database

# Database do not accepting frozensets

# Removing frozenset from dataframe

rules10['antecedents'] = rules10['antecedents'].astype('string')

rules10['consequents'] = rules10['consequents'].astype('string')

rules10['antecedents'] = rules10['antecedents'].str.removeprefix("frozenset({")

rules10['antecedents'] = rules10['antecedents'].str.removesuffix("})")

rules10['consequents'] = rules10['consequents'].str.removeprefix("frozenset({")

rules10['consequents'] = rules10['consequents'].str.removesuffix("})")

rules10.to\_sql('bookshop\_ar', con = engine, if\_exists = 'replace', chunksize = 1000, index = False)

**Output:**

count = transf\_df.loc[:, :].sum()

count

Out[290]:

ChildBks 846

YouthBks 495

CookBks 862

DoItYBks 564

RefBks 429

ArtBks 482

GeogBks 552

ItalCook 227

ItalAtlas 74

ItalArt 97

Florence 217

dtype: int64

pop\_item

Out[300]:

items count

0 CookBks 862

1 ChildBks 846

2 DoItYBks 564

3 GeogBks 552

4 YouthBks 495

5 ArtBks 482

6 RefBks 429

7 ItalCook 227

8 Florence 217

9 ItalArt 97

frequent\_itemsets

Out[306]:

support itemsets

2 0.4310 (CookBks)

0 0.4230 (ChildBks)

3 0.2820 (DoItYBks)

6 0.2760 (GeogBks)

12 0.2560 (ChildBks, CookBks)

.. ... ...

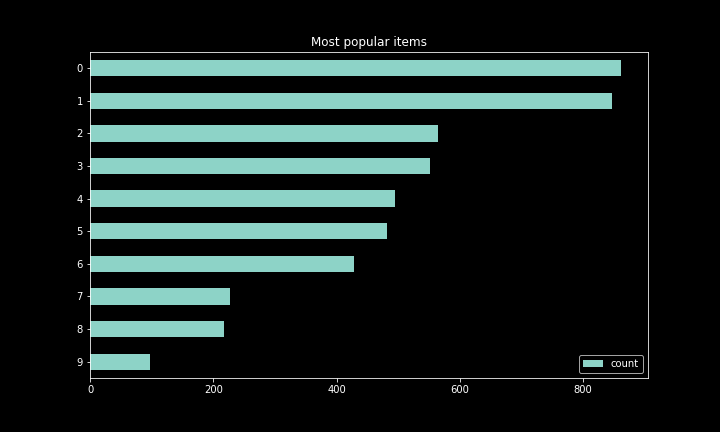
40 0.0370 (ItalAtlas, RefBks)

96 0.0365 (ArtBks, ItalCook, DoItYBks)

18 0.0360 (ItalArt, ChildBks)

68 0.0360 (ItalArt, ChildBks, ArtBks)

98 0.0360 (ArtBks, GeogBks, ItalCook)



rules10 = rules\_no\_redundancy.sort\_values('lift', ascending = False).head(10)

rules10

Out[325]:

antecedents consequents ... conviction zhangs\_metric

350 (ItalArt, CookBks) (ItalCook) ... 10.384714 0.913354

356 (ItalArt) (ItalCook) ... 3.908659 0.896696

364 (ItalAtlas) (RefBks) ... inf 0.815680

374 (ItalArt, ChildBks) (ArtBks) ... inf 0.787344

302 (ItalArt) (ArtBks) ... inf 0.797688

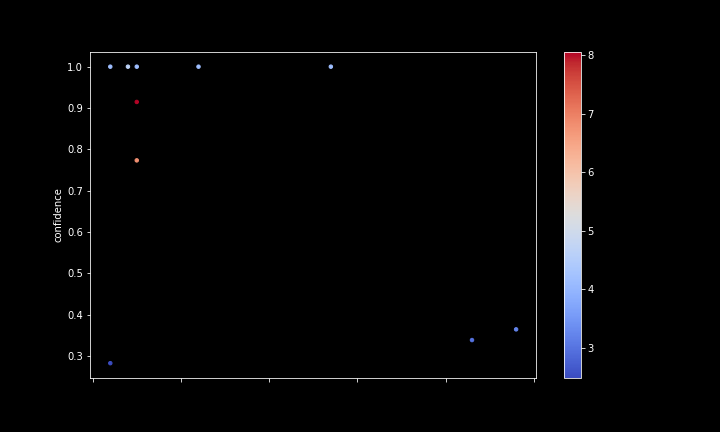
344 (ItalArt, CookBks) (ArtBks) ... inf 0.791449

358 (ItalArt, ItalCook) (ArtBks) ... inf 0.788571

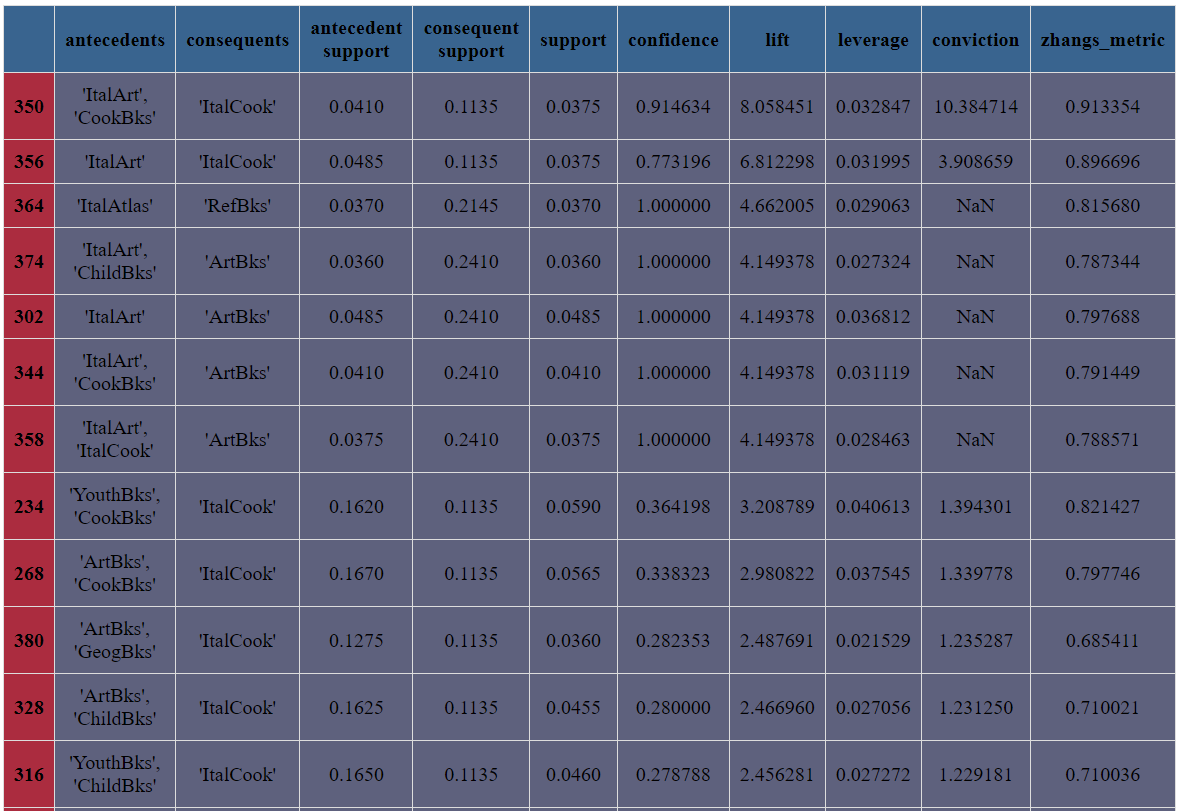
234 (YouthBks, CookBks) (ItalCook) ... 1.394301 0.821427

268 (ArtBks, CookBks) (ItalCook) ... 1.339778 0.797746

380 (ArtBks, GeogBks) (ItalCook) ... 1.235287 0.685411



**Deployment of Association Rule using Flask**

****