**# =========================**

**# Association Analysis**

**# =========================**

# Association analysis (commonly known as 'Market Basket Analysis') is used

# to find the frequent patterns from huge amounts of data and produce

# association rules from those frequent patterns.

# We will use the apriori algorithm to find the frequent itemsets. From

# each frequent itemset, we will produce the association rules using the

# confidence and lift values.

# To understand this tutorial, you need to be familiar with the pandas

# dataframe.

import pandas as pd

from apyori import apriori

# if you have not installed 'apyori' yet, visit the following site for

# installation: https://pypi.org/project/apyori/

# ===========================================

# Read the grocery transaction data.

# ===========================================

f = open('groceries.txt','r')

groceries = []

# Convert the text data to the list of lists.

for line in f:

groceries.append(line.strip().split(","))

f.close()

len(groceries)

groceries[0:10]

# Another option could be as follows:

#

# f = open('groceries.txt','r')

# groceries = []

# while True:

# line = f.readline()

# if line =="":

# break

# groceries.append(line.strip().split(","))

# arules = list(apriori(groceries))

# ====================================================

# Find frequent itemsets using the apriori algorithm.

# ====================================================

arules = list(\

apriori(groceries, min\_support=0.02, min\_confidence=.1, min\_lift=1))

# 'arules' contains a list of 'RelationRecord's. Each 'RelationRecord'

# contains a frequent itemset and possible rules from the itemset.

# See below for an example for arules[65] at the bottom this tutorial.

# =================================

# Find the association rules.

# =================================

# Convert the lists to a dataframe

cols\_to\_display = ['FreqItemSet','RelSup','LHS','RHS','Confidence','Lift']

association\_rules = pd.DataFrame(columns=cols\_to\_display)

# Initialize the association\_rules

for item in arules:

r ={}

f\_itemset = item[0] # frequent itemset

if len(f\_itemset) == 1: # Do not use the rules with 1-frequent itemset.

continue

for rule in item[2]:

r['FreqItemSet'] = f\_itemset

rel\_sup = item[1] # relative support

r['RelSup'] = rel\_sup

LHS = rule[0]

r['LHS']= LHS

RHS = rule[1]

r['RHS']= RHS

Conf = rule[2]

r['Confidence']= Conf

Lift = rule[3]

r['Lift'] = Lift

association\_rules = association\_rules.append(r, ignore\_index=True)

# Show some association rules

association\_rules

association\_rules.head()

association\_rules.head()[['LHS','RHS','Confidence']]

print('Number of association rules: ', len(association\_rules))

# Show the top ten rules sorted by some attributes

# ==================

# # Queries

# ==================

# sorting\_criteria = ['RelSup','Lift','Confidence']

sorted\_rules = association\_rules.sort\_values(by=['Lift'], ascending=False)

sorted\_rules = association\_rules.sort\_values(by=['Confidence'], ascending=False)

print(sorted\_rules.iloc[0:10, 1:7])

# Find the rules containing certain items from the if-part (LHS)

s = {'whole milk'}

# s = {'yogurt','whole milk'}

df = pd.DataFrame(columns=cols\_to\_display)

for \_, r in association\_rules.iterrows():

if r['LHS'].issuperset(s):

df = df.append(r[cols\_to\_display], ignore\_index=True)

ans = df.sort\_values(by=['Lift'], ascending=False).iloc[0:10,1:7]

# Find the rules containing certain items from the then-part (RHS)

s = {'whole milk'}

# s = {'yogurt','whole milk'}

df = pd.DataFrame(columns=cols\_to\_display)

for \_, r in association\_rules.iterrows():

if r['RHS'].issuperset(s):

df = df.append(r[cols\_to\_display], ignore\_index=True)

df.sort\_values(by=['Lift'], ascending=False).iloc[0:10,1:7]

## End of tutorial ############################