QM-7063 Data Mining
Professor: Dr. Abdulrashid
Learning Practice 3 – Noah L. Schrick

```
# Learning Practice 3 for the University of Tulsa's QM-7063 Data Mining Course
#  Recommender Systems
# Professor: Dr. Abdulrashid, Spring 2023
# Noah L. Schrick - 1492657

import heapq
from collections import defaultdict

import pandas as pd
import matplotlib.pylab as plt
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

from surprise import Dataset, Reader, KNNBasic
from surprise.model_selection import train_test_split
```

 # Problem 14.1

An analyst at a subscription-based satellite radio company has been given a sample of data from their customer database, with the goal of finding groups of customers who are associated with one another. The data consist of company data, together with purchased demographic data that are mapped to the company data (see Table 14.13). The analyst decides to apply association rules to learn more about the associations between customers. Comment on this approach.

This is a good approach for exploring associative relationships between customers. Since there is company data mixed with demographic data, the association rules can yield better results and demonstrate better associations since purchases can be examined with respect to age, location, number of dependents, and any other demographic data available.

 # Problem 14.3

We again consider the data in CourseTopics.csv describing course purchases at Statistics.com (see Problem 14.2 and data sample in Table 14.14). We want to provide a course recommendation to a student who purchased the Regression and Forecast courses. Apply user-based collaborative filtering to the data. You will get a Null matrix. Explain why this happens.

```
## Read in Course Topics data
courses_df = pd.read_csv('Coursetopics.csv')

# Convert to format usable for surprise similarities
courses_df['Index'] = range(1, len(courses_df) + 1)
course_melt = courses_df.melt(id_vars =['Index'], value_vars =['Intro', 'DataMining', 'Survey', 'Cat Data',
'Regression', 'Forecast', 'DOE', 'SW'],
        var_name ='Course', value_name ='Taken')

reader = Reader(rating_scale=(0, 1))
data = Dataset.load_from_df(course_melt[['Index', 'Course', 'Taken']], reader)
trainset = data.build_full_trainset()
```

```
# NOTE: The following will error. This is expected and part of the question. Explanation in the corresponding answer.
#sim_options = {'name': 'cosine', 'user_based': True}  # compute cosine similarities between users
#algo = KNNBasic(sim_options=sim_options)
#algo.fit(trainset)
```

The provided dataset is composed of binary values for "have taken" or "have not taken" various courses. The dataset represents "have not taken" with a 0, and "taken" with a 1. The dataset is considered a sparse matrix, since each user has only taken a few of the listed courses. Due to the sparsity, when computing the cosine between users, many computations involve comparing a user's "not taken" course to another user's "not taken" course. This leads to difficulties with the cosine computation since the denominator will be zero, causing a float division error. This can be remedied by using "NULL" values, which are supported in the surprise package, by using a sparse matrix cosine similarity function, or error-checking for zeroes during the cosine similarity computation.

 # Problem 14.4

The data shown in Table 14.15 and the output in Table 14.16 are based on a subset of a dataset on cosmetic purchases (Cosmetics.csv) at a large chain drugstore. The store wants to analyze associations among purchases of these items for purposes of point-of-sale display, guidance to sales personnel in promoting cross-sales, and guidance for piloting an eventual time-of-purchase electronic recommender system to boost cross-sales. Consider first only the data shown in Table 14.15, given in binary matrix form.

a. Select several values in the matrix and explain their meaning.

b. Consider the results of the association rules analysis shown in Table 14.16.

i. For the first row, explain the “confidence” output and how it is calculated.

ii. For the first row, explain the “support” output and how it is calculated.

iii. For the first row, explain the “lift” and how it is calculated.

iv. For the first row, explain the rule that is represented there in words.

c. Now, use the complete dataset on the cosmetics purchases (in the file Cosmetics.csv). Using Python, apply association rules to these data (for apriori use min_support=0.1 and use_colnames=True, for association_rules use default parameters).

i. Interpret the first three rules in the output in words.

ii. Reviewing the first couple of dozen rules, comment on their redundancy and how you would assess their utility.

a)

Transaction 1: Customer purchased Blush, Nail Polish, Brushes, Concealer, and Bronzer.
Transaction 6: Customer purchased Concealer.
Transaction 11: Customer purchased Nail Polish and Bronzer.

b)

i) Confidence is the measure of uncertainty regarding the association rule. The confidence is calculated using the number of transactions that contain both antecedents and consequents, divided by the number of transactions with the antecedent. For the first row, a 30.23% confidence is obtained. The confidence is rather low due to the size of the dataset. Only 12 transactions occurred, and of these 12, only 1 transaction purchased eyebrow pencils.

ii) Support is the degree in which the data supports the validity of an association rule. It is the number of antecedent and consequent itemsets. Support may also be represented in percentage form, which is the number of antecedent and consequent itemset occurrences divided by the total number of entries. The first row has a support value of 0.013, which corresponds to the one entry for eyebrow pencils, divided by the 77 total entries.

iii) Lift is the strength of an association rule assuming the antecedents and consequents are independent. The support is obtained similar to the calculation for confidence, but is now based on independence. Lift is equal to the confidence divided by the benchmark confidence, where benchmark confidence is obtained by the number of transactions with the consequent, divided by the total number of transactions. The first row has a lift of 7.198... which corresponds to the confidence (0.3023) divided by (1/24).

iv) If Blush, Concealer, Mascara, Eye Shadow, and Lipstick is purchased, then Eyebrow Pencils are also purchased.

```
## c
# Read in Cosmetics data
cosmetics_df = pd.read_csv('Cosmetics.csv')
cosmetics_df = cosmetics_df.drop('Trans. ', axis=1)
# create frequent itemsets
itemsets = apriori(cosmetics_df, min_support=0.1, use_colnames=True)

# and convert into rules
rules = association_rules(itemsets)
#rules.sort_values(by=['lift'], ascending=False).head(6)

print(rules.head(12))
#print(rules.sort_values(by=['lift'], ascending=False)
#      .drop(columns=['antecedent support', 'consequent support', 'conviction'])
#      .head(6))
```

```
              antecedents    consequents  antecedent support  \
0                (Brushes)  (Nail Polish)               0.149
1                (Mascara)   (Eye shadow)               0.357
2              (Eye shadow)      (Mascara)               0.381
3         (Blush, Lip liner)    (Concealer)               0.124
4           (Blush, Mascara)   (Eye shadow)               0.184
5         (Blush, Eye shadow)      (Mascara)               0.182
6      (Nail Polish, Mascara)   (Eye shadow)               0.134
7   (Nail Polish, Eye shadow)      (Mascara)               0.131
8        (Lip liner, Bronzer)    (Concealer)               0.128
9         (Eyeliner, Bronzer)    (Concealer)               0.146
10      (Lip liner, Eyeliner)    (Concealer)               0.130
11       (Mascara, Concealer)   (Eye shadow)               0.204

    consequent support  support  confidence      lift  leverage  conviction
0                0.280    0.149    1.000000  3.571429  0.107280         inf
1                0.381    0.321    0.899160  2.359999  0.184983    6.138417
2                0.357    0.321    0.842520  2.359999  0.184983    4.083050
3                0.442    0.108    0.870968  1.970515  0.053192    4.324500
4                0.381    0.169    0.918478  2.410704  0.098896    7.593067
5                0.357    0.169    0.928571  2.601040  0.104026    9.002000
6                0.381    0.119    0.888060  2.330865  0.067946    5.529733
```

7	0.357	0.119	0.908397	2.544529	0.072233	7.019417
8	0.442	0.103	0.804687	1.820560	0.046424	2.856960
9	0.442	0.119	0.815068	1.844046	0.054468	3.017333
10	0.442	0.120	0.923077	2.088409	0.062540	7.254000
11	0.381	0.179	0.877451	2.303021	0.101276	5.051040

c)

i)

If Brushes are purchased, then Nail Polish is purchased.
If Mascara is purchased, then Eye Shadow is purchased.
If Eye Shadow is purchaed, then Mascara is purchased.

ii)

The first couple dozen of rules are permutations of themselves, and have a degree of redundancy due to these permutations. Sorting the list of rules by lift would allow for an easier assessment of utility since rules with greater lift would be grouped together at the top of the list.