Lora Milam
Western Governors University
D212 Data Mining II
4 April 2023

D212 Performance Assessment Task 3

1 Introduction

Lora Milam Masters Data Analytics (2/19/2023) Program Mentor:d212@wgu.edu

1.1 Research Question

This analysis will investigate whether there are potential relationships between purchased prescriptions.

1.2 Research Goal

The goal of this analysis is to identify purchasing tendencies of patients and determine which medications they are likely to buy based on their purchase history.

2 Technique Justification

2.1 Explanation of Market Basket

In summary, Market Basket Analysis utilizes association rules to predict the likelihood of products being purchased together. These association rules count the frequency of items that occur together and look for pairings that occur more frequently than expected (TechTarget) .

The expected outcome of the apriori algorithm is that it will identify purchasing relationships.

2.2 Transaction Example

A transaction example of the dataset would be the purchasing relationship between “amlodipine” and “abilify”.

2.3 Market Basket Assumption

One assumption of the Market Basket Analysis is,“that customers who purchase a specific item are more likely to purchase another specific item or group of items”(Indeed).

3 Data Preparation and Analysis

In [1]:
```
pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\mel\anaconda3\lib\site-packages (0.21.0)
Requirement already satisfied: setuptools in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (63.4.1)
Requirement already satisfied: joblib>=0.13.2 in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (1.1.0)
Requirement already satisfied: pandas>=0.24.2 in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (1.4.4)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (3.5.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (1.21.5)
Requirement already satisfied: scipy>=1.2.1 in c:\users\mel\anaconda3\lib\site-packages (from mlxtend) (1.9.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(1.4.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxten
d) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2
1.3)
Requirement already satisfied: cycler>=0.10 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.1
1.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (9.
2.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\mel\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(3.0.9)
Requirement already satisfied: pytz>=2020.1 in c:\users\mel\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2022.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\mel\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxt
end) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\mel\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.
0.0->mlxtend) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
# Libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

In [3]:
```
# Display Settings
pd.set_option('display.max_columns', None)
```

In [4]:
```
# Import dataset into Pandas dataframe
df = pd.read_csv('medical_market_basket.csv')
df
```

2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	citalopram	benicar	amphetamine salt combo xr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
14997	clopidogrel	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14998	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14999	alprazolam	losartan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15001	amphetamine salt combo xr	levofloxacin	diclofenac sodium	cialis	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

15002 rows × 20 columns

3.1 Transforming the Dataset

In [5]:
```
# Review dataset
# Variables within dataset
df.columns
```

Out[5]:
```
Index(['Presc01', 'Presc02', 'Presc03', 'Presc04', 'Presc05', 'Presc06',
       'Presc07', 'Presc08', 'Presc09', 'Presc10', 'Presc11', 'Presc12',
       'Presc13', 'Presc14', 'Presc15', 'Presc16', 'Presc17', 'Presc18',
       'Presc19', 'Presc20'],
      dtype='object')
```

In [6]:
```
# Dataset dimensions
df.shape
```

Out[6]:
```
(15002, 20)
```

In [7]:
```
# Summary stats of variables
df.describe()
```

Out[7]:

	Presc01	Presc02	Presc03	Presc04	Presc05	Presc06	Presc07	Presc08	Presc09	Presc10	Presc11	Presc12	Presc13	Presc14	Presc15	Pres
count	7501	5747	4389	3345	2529	1864	1369	981	654	395	256	154	87	47	25	
unique	115	117	115	114	110	106	102	97	88	80	66	50	43	28	19	
top	abilify	abilify	abilify	abilify	losartan	glyburide	losartan	losartan	losartan	losartan	cialis	losartan	losartan	losartan	celebrex	spironolact
freq	577	484	375	201	153	107	96	67	57	31	22	15	8	4	3	

In [8]:
```
# Review datatype of variables
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15002 entries, 0 to 15001
Data columns (total 20 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Presc01  7501 non-null   object
 1   Presc02  5747 non-null   object
 2   Presc03  4389 non-null   object
 3   Presc04  3345 non-null   object
 4   Presc05  2529 non-null   object
 5   Presc06  1864 non-null   object
 6   Presc07  1369 non-null   object
 7   Presc08  981 non-null    object
 8   Presc09  654 non-null    object
 9   Presc10  395 non-null    object
 10  Presc11  256 non-null    object
 11  Presc12  154 non-null    object
 12  Presc13  87 non-null     object
 13  Presc14  47 non-null     object
```

```
In [9]: # Determine unique prescriptions
        print(df.nunique())

Presc01    115
Presc02    117
Presc03    115
Presc04    114
Presc05    110
Presc06    106
Presc07    102
Presc08     97
Presc09     88
Presc10     80
Presc11     66
Presc12     50
Presc13     43
Presc14     28
Presc15     19
Presc16      8
Presc17      3
Presc18      3
Presc19      3
Presc20      1
dtype: int64
```

3.2 Code Execution

```
In [10]: # Determine if there are any Null values
         df.isna().any()

Out[10]: Presc01    True
         Presc02    True
         Presc03    True
         Presc04    True
         Presc05    True
         Presc06    True
         Presc07    True
         Presc08    True
         Presc09    True
         Presc10    True
         Presc11    True
         Presc12    True
         Presc13    True
         Presc14    True
         Presc15    True
         Presc16    True
         Presc17    True
         Presc18    True
         Presc19    True
         Presc20    True
         dtype: bool
```

In [11]:
```
# Drop rows that are entirely Null
df = df.dropna(how = 'all')
df.shape
```
Out[11]:
```
(7501, 20)
```
In [12]:
```
# Create a list of lists from Dataframe
trans_list = df.stack().groupby(level = 0).apply(list).tolist()
trans_list
```
Out[12]:
```
[['amlodipine',
  'albuterol aerosol',
  'allopurinol',
  'pantoprazole',
  'lorazepam',
  'omeprazole',
  'mometasone',
  'fluconozole',
  'gabapentin',
  'pravastatin',
  'cialis',
  'losartan',
  'metoprolol succinate XL',
  'sulfamethoxazole',
  'abilify',
  'spironolactone',
  'albuterol HFA',
  'levofloxacin',
  'promethazine',
  'glipizide'],
```
In [13]:
```
# Transform list of lists into array with TransactionEncoder
trans_enc = TransactionEncoder()
array = trans_enc.fit(trans_list).transform(trans_list)
```
In [14]:
```
# Create new dataframe
cleandf = pd.DataFrame(array, columns = trans_enc.columns_)
cleandf.head()
```
Out[14]:

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendronate	allopurinol	alprazolam	amitriptyline	amlodipine	amoxicillin
0	False	False	False	True	False	False	True	True	False	True	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	True	False	False	False	False
4	False	False	False	True	False	False	False	False	False	False	False	False	False	False

In [15]:
```
# Transaction Example
#cleandf[cleandf.columns[cleandf.iloc[0] == True ]]
```
In [16]:
```
# Save clean dataframe
cleandf.to_csv('D212_Part3_Clean_Data.csv', index = False)
```

3.3 Association Rules Table

In [17]:
```
# Apriori algorithm
frequent_itemsets = apriori(cleandf, min_support = .02, use_colnames = True)
frequent_itemsets.head()
```

Out[17]:

	support	itemsets
0	0.046794	(Premarin)
1	0.238368	(abilify)
2	0.020397	(albuterol aerosol)
3	0.033329	(allopurinol)
4	0.079323	(alprazolam)

In [18]:
```
# Association rules
rules = association_rules(frequent_itemsets,
                          metric = "lift",
                          min_threshold = 1.0)
rules.head()
```

Out[18]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	0.006564	1.137144
1	(abilify)	(amlodipine)	0.238368	0.071457	0.023597	0.098993	1.385352	0.006564	1.030562
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	0.008095	1.183991
3	(abilify)	(amphetamine salt combo)	0.238368	0.068391	0.024397	0.102349	1.496530	0.008095	1.037830
4	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815

In [19]:
```
# Pruning by confidence
pruned_rules = rules[rules['confidence']>.2]
```

3.4 Top Three Rules

In [20]:
```
pruned_rules.head()
```

Out[20]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	0.006564	1.137144
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	0.008095	1.183991
4	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815
5	(abilify)	(amphetamine salt combo xr)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158
6	(atorvastatin)	(abilify)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650

4 Data Summary and Implications

4.1 Significance of Support, Lift, and Confidence Summary

The lift metric measures the tendency two medications are sold together. To be considered significant, values must be higher than one.

In [21]:
```python
# Sorting rules by lift
sorted_rules = rules.sort_values('lift', ascending = False).head(3)
sorted_rules
```

Out[21]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
75	(carvedilol)	(lisinopril)	0.174110	0.098254	0.039195	0.225115	2.291162	0.022088	1.163716
74	(lisinopril)	(carvedilol)	0.098254	0.174110	0.039195	0.398915	2.291162	0.022088	1.373997
72	(glipizide)	(carvedilol)	0.065858	0.174110	0.022930	0.348178	1.999758	0.011464	1.267048

The confidence metric measures how often the antecedent is purchased with the consequent. Confidence determines the probability the consequent will be purchased if the antecedent has been purchased.

In [22]:
```python
# Sorting rules by confidence
sorted_rules = rules.sort_values('confidence', ascending = False).head(3)
sorted_rules
```

Out[22]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
31	(metformin)	(abilify)	0.050527	0.238368	0.023064	0.456464	1.914955	0.011020	1.401255
25	(glipizide)	(abilify)	0.065858	0.238368	0.027596	0.419028	1.757904	0.011898	1.310962
29	(lisinopril)	(abilify)	0.098254	0.238368	0.040928	0.416554	1.747522	0.017507	1.305401

The support metric indicates the relative concentration of a medication in the dataset. To be considered significant, value must be higher than zero.

In [25]:
```python
# Sorting rules by support
sorted_rules = rules.sort_values('support', ascending = False).head(3)
sorted_rules
```

Out[25]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
8	(abilify)	(carvedilol)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008
9	(carvedilol)	(abilify)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
19	(abilify)	(diazepam)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256

4.2 Practical Significance of Findings

The practical significance of this analysis is that there is now a numeric metric associated with lift, confidence, and support. For online purchases, these metrics can aid in suggesting additional medications for a patient based on what is currently in their cart.

4.3 Course of Action

A course of action that could be taken would be to find the most likely purchasing relationships and then use this information to increase sales. For example, bundling medications that have a high purchasing relationship with discounts would more incentivise customers to purchase them.

5 Supporting Documentation

5.1 Video

This can be found within the attached file ‘Panopto Recording’.

5.2 Sources

Indeed. (2022, October 12). FAQ: What is market basket analysis? (types plus examples). Indeed. Retrieved April 7, 2023, from https://sg.indeed.com/career-advice/career-development/market-basket-analysis

TechTarget. “What Is Market Basket Analysis?” Customer Experience, TechTarget, 31 Jan. 2023, https://www.techtarget.com/searchcustomerexperience/definition/market-basket-analysis.

Western Governors University. (n.d.). D212 Data Mining II. Salt Lake City.