Western Governors University

D212 – Data Mining II – Task 3

By Krista Moik

Table of Contents

Part I: Research Question

A1. Proposal of Question

Using the provided medical_market_basket CSV, my research question is: **Which prescriptions are most commonly prescribed together?**

A2. Defined Goal

My goal is to use market basket analysis to find trends in the provided prescription data to see if certain prescriptions are more likely to be prescribed with other prescriptions.

Part II: Market Basket Justification

B1. Explanation of Market Basket

Market Basket Analysis is a data mining technique used to understand purchasing patterns by locating product groupings and identifying items that are purchased together. This analysis looks at historical transactional data to create rules showing the relationships between items that are often purchased together. The rules indicate the likelihood of an item being purchased if another item is already being purchased. The items are described as antecedent (the IF part of the rule) and as consequent (the THEN part of the rule). These insights are then used by businesses for insights involving such decisions like promotions and product placement (Kadlaskar, 2024). Market Basket Analysis, for purposes of this data set, will analyze the WGU-provided CSV of prescription records to find trends in them, such as which prescriptions are most often prescribed with each other. I expect an outcome of rules showing which prescriptions are most often prescribed with each other.

B2. Transaction Example

Below is one transaction example from the data set:

```
#example transaction
df.iloc[5]
```

```
Presc01    enalapril
Presc02          NaN
Presc03          NaN
Presc04          NaN
Presc05          NaN
Presc06          NaN
Presc07          NaN
Presc08          NaN
Presc09          NaN
Presc10          NaN
Presc11          NaN
Presc12          NaN
Presc13          NaN
Presc14          NaN
Presc15          NaN
Presc16          NaN
Presc17          NaN
Presc18          NaN
Presc19          NaN
Presc20          NaN
Name: 5, dtype: object
```

This transaction shows 1 prescription (out of 20 total possible), which was enalapril.

B3. Market Basket Assumption

One assumption of Market Basket Analysis is that co-occurrence of 2 or more items implies that the purchase of one will lead to the purchase of others (Kadlaskar, 2024).

Part III: Data Preparation and Analysis

C1. Transforming the Data Set

After loading the data set, the data was viewed to obtain information on the data types and shape:

```
df.head(5)
```

	Presc01	Presc02	Presc03	Presc04	Presc05	Presc06	Presc07	Presc08	Presc09	Presc10	Presc11	Presc12	Presc13	
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	amlodipine	albuterol aerosol	allopurinol	pantoprazole	lorazepam	omeprazole	mometasone	fluconozole	gabapentin	pravastatin	cialis	losartan	metoprolol succinate XL	sul
2	NaN	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	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

```
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
 14  Presc15  25 non-null     object
 15  Presc16  8 non-null      object
 16  Presc17  4 non-null      object
 17  Presc18  4 non-null      object
 18  Presc19  3 non-null      object
 19  Presc20  1 non-null      object
dtypes: object(20)
memory usage: 2.3+ MB
```

```
df.shape
```

```
(15002, 20)
```

Nulls were dropped:

```
#drop nulls
df=df[df['Presc01'].notna()]
```

```
df.shape
```

```
(7501, 20)
```

The data was transformed into a list of lists in preparation for TransactionEncoder (Western Governors University, n.d.):

```
#dataframe converted to list of lists
rows=[]
for i in range (0,7501):
    rows.append([str(df.values[i,j])
for j in range (0,20)])
```

The list was then fed into TransactionEncoder (Western Governors University, n.d.):

```
#list fed to TransactionEncoder
DE=TransactionEncoder()
array=DE.fit(rows).transform(rows)
```

```
#return array to DataFrame
transaction=pd.DataFrame(array, columns=DE.columns_)
```

```
#display transaction
transaction
```

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendronate	allopurinol	...	trazodone HCl	triamcinolone Ace topical	triamterene	trimethoprim DS	valaciclovir	valsartan	venlafaxine XR	verapamil SR	viagra	zolpidem
0	False	False	False	True	False	False	True	True	False	True	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	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	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	True	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

7496	False	False	False	False	False	False	False	False	False	False	...	False
	False	False	False	False	False	False	False	False	False			
7497	False	False	False	False	False	False	False	False	False	False	...	False
	False	False	False	False	False	False	False	False	False			
7498	False	False	False	False	False	False	False	False	False	False	...	False
	False	False	False	False	False	False	False	False	False			
7499	False	False	False	False	False	False	False	False	False	False	...	False
	False	False	False	False	False	False	False	False	False			
7500	False	False	False	False	False	False	False	False	False	False	...	False
	False	False	False	False	False	False	False	False	False			

7501 rows × 120 columns

Additional nulls were dropped:

```
#remove NAN column from transformed dataset
clean_df=transaction.drop(['nan'], axis=1)
clean_df.head(7501)
```

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendronate	allopurinol	...	trazodone HCI	triamcinolone Ace topical	triamterene
0	False	False	False	True	False	False	True	True	False	True	...	False	False	False
1	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
3	False	False	False	False	False	False	False	False	False	True	...	False	False	False
4	False	False	False	True	False	False	False	False	False	False	...	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7496	False	False	False	False	False	False	False	False	False	False	...	False	False	False
7497	False	False	False	False	False	False	False	False	False	False	...	False	False	False
7498	False	False	False	False	False	False	False	False	False	False	...	False	False	False
7499	False	False	False	False	False	False	False	False	False	False	...	False	False	False
7500	False	False	False	False	False	False	False	False	False	False	...	False	False	False

7501 rows × 119 columns

Final cleaned data set was saved as KMoikD212_transf.csv (which is attached):

```
#save to csv
clean_df.to_csv('C:/Users/Kmoik WGU/Desktop/KMoikD212_transf.csv', index=False)
clean_df.columns
```

```
Index(['Duloxetine', 'Premarin', 'Yaz', 'abilify', 'acetaminophen', 'actonel',
       'albuterol HFA', 'albuterol aerosol', 'alendronate', 'allopurinol',
       ...
       'trazodone HCI', 'triamcinolone Ace topical', 'triamterene',
       'trimethoprim DS', 'valaciclovir', 'valsartan', 'venlafaxine XR',
       'verapamil SR', 'viagra', 'zolpidem'],
      dtype='object', length=119)
```

A copy of the full code used in this task is attached as ipynb and pdf files titled KMoikD212Code3.

C2. Code Execution

To generate association rules with the Apriori algorithm, I first created the Apriori rules (Western Governors University, n.d.):

```
#creating apriori rules
rules=apriori(df, min_support=0.02, use_colnames=True)
rules.head(5)
```

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

A copy of the full code used in this task is attached as ipynb and pdf files titled KMoikD212Code3.

C3. Association Rules Table

The rules table was created with the values for support, lift, and confidence for all 20 prescriptions in the data set (Western Governors University, n.d.):

```
#creating rules table
rul_table=association_rules(rules, metric='lift', min_threshold=1)
rul_table.head(20)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	0.006564	1.137144	0.299568
1	(abilify)	(amlodipine)	0.238368	0.071457	0.023597	0.098993	1.385352	0.006564	1.030562	0.365218
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	0.008095	1.183991	0.356144
3	(abilify)	(amphetamine salt combo)	0.238368	0.068391	0.024397	0.102349	1.496530	0.008095	1.037830	0.435627
4	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815	0.193648
5	(abilify)	(amphetamine salt combo xr)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158	0.208562
6	(atorvastatin)	(abilify)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650	0.409465
7	(abilify)	(atorvastatin)	0.238368	0.129583	0.047994	0.201342	1.553774	0.017105	1.089850	0.467950
8	(carvedilol)	(abilify)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314	0.369437
9	(abilify)	(carvedilol)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008	0.400606
10	(abilify)	(cialis)	0.238368	0.076523	0.023997	0.100671	1.315565	0.005756	1.026851	0.314943
11	(cialis)	(abilify)	0.076523	0.238368	0.023997	0.313589	1.315565	0.005756	1.109585	0.259747
12	(citalopram)	(abilify)	0.087188	0.238368	0.024397	0.279817	1.173883	0.003614	1.057552	0.162275
13	(abilify)	(citalopram)	0.238368	0.087188	0.024397	0.102349	1.173883	0.003614	1.016889	0.194486
14	(clopidogrel)	(abilify)	0.059992	0.238368	0.022797	0.380000	1.594172	0.008497	1.228438	0.396502
15	(abilify)	(clopidogrel)	0.238368	0.059992	0.022797	0.095638	1.594172	0.008497	1.039415	0.489364
16	(dextroamphetamine XR)	(abilify)	0.081056	0.238368	0.027463	0.338816	1.421397	0.008142	1.151921	0.322617
17	(abilify)	(dextroamphetamine XR)	0.238368	0.081056	0.027463	0.115213	1.421397	0.008142	1.038604	0.389252
18	(diazepam)	(abilify)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357	0.308965
19	(abilify)	(diazepam)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256	0.339197

A copy of the full code used in this task is attached as ipynb and pdf files titled KMoikD212Code3.

C4. Top Three Rules

Using the Apriori algorithm, I generated the below top three rules (Western Governors University, n.d.):

```
#sorting rules by metric - confidence
top_three_rules=rul_table.sort_values('confidence', ascending=False).head(3)
top_three_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
30	(metformin)	(abilify)	0.050527	0.238368	0.023064	0.456464	1.914955	0.011020	1.401255	0.503221
24	(glipizide)	(abilify)	0.065858	0.238368	0.027596	0.419028	1.757904	0.011898	1.310962	0.461536
28	(lisinopril)	(abilify)	0.098254	0.238368	0.040928	0.416554	1.747522	0.017507	1.305401	0.474369

```
#sorting rules by metric - lift
top_three_rules=rul_table.sort_values('lift', ascending=False).head(3)
top_three_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
75	(carvedilol)	(lisinopril)	0.174110	0.098254	0.039195	0.225115	2.291162	0.022088	1.163716	0.682343
74	(lisinopril)	(carvedilol)	0.098254	0.174110	0.039195	0.398915	2.291162	0.022088	1.373997	0.624943
72	(glipizide)	(carvedilol)	0.065858	0.174110	0.022930	0.348178	1.999758	0.011464	1.267048	0.535186

```
#sorting rules by metric - support
top_three_rules=rul_table.sort_values('support', ascending=False).head(3)
top_three_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
8	(carvedilol)	(abilify)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314	0.369437
9	(abilify)	(carvedilol)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008	0.400606
19	(abilify)	(diazepam)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256	0.339197

The first rule, confidence, is the proportion of all transactions that involve all the items in the data set divided by the proportion of all transactions involving just one of the items, which provides the probability that someone will purchase the consequent. The second rule, lift, is the likelihood that when the antecedent is purchased the consequent is also purchased. The third rule, support, is a measure of the frequency a particular item appears in the data set (Sivek, 2020).

Part IV: Data Summary and Implications

D1. Significance of Support, Lift, and Confidence Summary

The top confidence values were between 0.41 and 0.45, indicating a 41-45% chance that the consequent occurred when the antecedent was present.

The top lift values were between 1.99 to 2.29, indicating a strong co-occurrence between the antecedents and consequents and they are more likely to be purchased/prescribed together.

The top support values were approximately 0.05, indicating the prescriptions occurred in approximately 5% of all transactions. In my analysis, the prescriptions that appear most frequently in the data set are carvedilol and abilify.

D2. Practical Significance of Findings

Using the top three rules table (Western Governors University, n.d.):

```
#sorted rules
sorted_rules=rul_table[(rul_table['lift']>0.08)]
sorted_rules.head(3)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	0.006564	1.137144	0.299568
1	(abilify)	(amlodipine)	0.238368	0.071457	0.023597	0.098993	1.385352	0.006564	1.030562	0.365218
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	0.008095	1.183991	0.356144

When Amlodipine, a medication used as a calcium channel blocker to lower blood pressure, is prescribed, there is a 33% chance that Abilify, an antidepressant medication, is also prescribed. This co-occurrence has a lift of 1.38 even though Amlodipine only accounted for 2% of all prescriptions.

When Abilify is prescribed, there is a 9% chance that Amlodipine is also prescribed. This co-occurrence has a lift of 1.38 even though Abilify only accounted for 7% of all prescriptions.

When Amphetamine Salt Combo, a medication for ADHD, is prescribed, there is a 35% chance that Abilify is also prescribed. This co-occurrence has a lift of 1.49 even though Amphetamine Salt Combo only accounted for 2% of all prescriptions.

Knowing and understanding co-occurrent relationships between different medications can provide insight on the patients as well as indicate a need for additional research.

D3. Course of Action

As predicted, my analysis did provide information and trends regarding prescription co-occurrence. It is interesting that Amlodipine and Abilify are often antecedents and consequents for each other. This could show that they are often prescribed together as one may help with side effects from the other. However, it could also mean additional research might be needed to determine if one of these medications is in fact the cause of the side effects requiring the need for the consequent prescription. I would recommend obtaining additional information on the patients prescribed these medications to gain more understanding of their co-occurrence.

Part V: Attachments

E. Panopto Video of Code and Programs

The link to my Panopto video is:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=448491df-aebb-427a-bc42-b1330162ba1c

F. Sources of Third-Party Code

Western Governors University. (n.d.). *Data Mining II - D212 Task 3*. WGU. [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=db85c4f1-0da5-4bde-a1a4-b07c0019d46d

G. Sources

Kadlaskar, Amruta. (2024). *Market Basket Analysis: A Comprehensive Guide for Businesses*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-market-basket-analysis/

Sivek, Susan Currie. (2020). *Market Basket Analysis 101: Key Concepts*. Towards Data Science. https://towardsdatascience.com/market-basket-analysis-101-key-concepts-1ddc6876cd00

H. Professional Communication