APRIL 15, 2024

PERFORMANCE ASSESSMENT

D212

ROBERT PATTON
WESTERN GOVERNORS' UNIVERSITY
Rpatt33@wgu.edu

Table of Contents

Part I: Research Question

A. Purpose of Data Mining Report

1. Using market basket analysis, can we determine which medications are commonly prescribed with diazepam? Can we improve patient care knowing this information?
2. In determining which medications are commonly prescribed with diazepam, medical providers and hospital executives can attempt to explain if there are underlying medical conditions that require diazepam as a primary medication for patients. With this information, we can determine if it is necessary for patients to maintain a medication regimen or if other factors can treat or improve a medical condition and reduce the number of medications a patient has to take.

Part II: Market Basket Justification

B. Reasons for Market Basket Analysis

1. We use market basket analysis in data mining as an effort to uncover relationships between items purchased together. When using market basket analysis in data science, a set of lists is analyzed for combinations of items frequently purchased together and calculates statistics based on the frequency of that item purchased. For example, if a data analyst was looking at a data set of grocery store purchases and wanted to know how likely a customer is to buy creamer with coffee, market basket analysis would help them understand that relationship and frequency. A rule is created that “if coffee, then creamer” and instantiates that a customer is more likely to purchase creamer when they purchase coffee. For this project, we are expecting to find rules that include diazepam where “if diazepam, then X or if X then diazepam”.
2. Below is an example of a transaction in the data set. For patient number two, shown as 3 in the screenshot, they were prescribed and purchased citalopram, Benicar, and amphetamine salt combo xr.

```
#drop blank rows from dataset
presc_df=presc_df[presc_df['Presc01'].notna()]
presc_df.head(3)
```

	Presc01	Presc02	Presc03	Presc04	Pr
1	amlodipine	albuterol aerosol	allopurinol	pantoprazole	lora
3	citalopram	benicar	amphetamine salt combo xr	NaN	
5	enalapril	NaN	NaN	NaN	

3. In market basket analysis, one assumption is that when two items have frequent occurrences in a transaction, the purchase of one of those items will lead to the purchase of the other. These items are therefore complements of one another (Hua 2015).

Part III: Data Preparation and Analysis

C. Prepare and Perform Market Basket Analysis

1. To prepare for performing a market basket analysis we need to first import our appropriate Python packages, read in our CSV file, and begin examining the data shape, missing values, and duplicate values. The data set we are working with consists of lists of prescription transactions for patients in a hospital. There are empty rows in the data set that need to be dealt with and removed before performing the analysis. We also need to transform the data by creating a list of lists so that true and false values show for each patient's transaction history. Finally, we can export the cleaned and prepared file. See the screenshots below for preparing data.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

```
presc_df=pd.read_csv('/Users/robertpatton/Desktop/Desktop – Robert's MacBook Pro/D212/Task 3/medical_market_basket.csv',)
```

```
#Look at data info
presc_df.head(5)
```

	Presc01	Presc02	Presc03	Presc04	Presc05	Presc06	Presc07	Presc08	Presc09	Presc10	Presc11	Presc12	Presc13	Presc14	Presc15	Presc16	Presc1
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
1	amlodipine	albuterol aerosol	allopurinol	pantoprazole	lorazepam	omeprazole	mometasone	fluconozole	gabapentin	pravastatin	cialis	losartan	metoprolol succinate XL	sulfamethoxazole	abilify	spironolactone	albuter HF
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
3	citalopram	benicar	amphetamine salt combo xr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal

```
#drop blank rows from dataset
presc_df=presc_df[presc_df['Presc01'].notna()]
presc_df
```

	Presc01	Presc02	Presc03	Presc04	Presc05	Presc06	Presc07	Presc08	Presc09	Presc10	Presc11	Presc12	Presc13	Presc14	Presc15	Presc16
1	amlodipine	albuterol aerosol	allopurinol	pantoprazole	lorazepam	omeprazole	mometasone	fluconozole	gabapentin	pravastatin	cialis	losartan	metoprolol succinate XL	sulfamethoxazole	abilify	spironolactone
3	citalopram	benicar	amphetamine salt combo xr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	enalapril	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	paroxetine	allopurinol	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	abilify	atorvastatin	folic acid	naproxen	losartan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14993	amphetamine	clotrimazole	lantus	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14995	citalopram	metoprolol	amphetamine salt combo xr	glyburide	celebrex	losartan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14997	clopidogrel	NaN	NaN	NaN	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	NaN	NaN	NaN	NaN
15001	amphetamine salt combo xr	levofloxacin	diclofenac sodium	cialis	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7501 rows × 20 columns

```python
#Create a list of lists
listoflists=[]
for i in range (0, 7501):
        listoflists.append([str(presc_df.values[i,j])
for j in range (0, 20)])
```

```python
#Feed list of lists into TransactionEncoder
TE=TransactionEncoder()
array=TE.fit(listoflists).transform(listoflists)
#Convert transaction encoder array back to dataframe
transaction=pd.DataFrame(array, columns= TE.columns_)
```

```python
#Look at transaction endoded dataframe
transaction
```

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendronate	allopurinol	...	trazodone HCI	triamcinolone Ace topical	triamterene	trimethoprim DS	valaciclovir	valsartan	venlafaxine XR
0	False	False	False	True	False	False	True	True	False	True	...	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
2	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
4	False	False	False	True	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
7497	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
7499	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

7501 rows × 120 columns

```python
#Drop the nan column in the dataframe
cleaned_presc= transaction.drop(['nan'], axis=1)
cleaned_presc.head(7501)
```

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendronate	allopurinol	...	trazodone HCI	triamcinolone Ace topical	triamterene	trimethoprim DS	valaciclovir	valsartan	venlafaxine XR
0	False	False	False	True	False	False	True	True	False	True	...	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
2	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
4	False	False	False	True	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
7497	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
7499	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

7501 rows × 119 columns

```python
#Export cleaned data to csv file
cleaned_presc.to_csv('/Users/robertpatton/Desktop/cleaned_presc.csv', index=False)
```

2. The next step is to generate association rules utilizing two Python functions. Both functions come from the mlxtend package and include the apriori algorithm and the association_rules function. The apriori algorithm will generate our frequent item sets using specific guideline thresholds we set. The association_rules function will complement the apriori algorithm by further specifying thresholds for finding transaction relationships. Screenshots are provided that show the code executing without error.

```
#Create apriori support rule
rules=apriori(cleaned_df, min_support= 0.02, use_colnames= True)
rules
```

	support	itemsets
0	0.046794	(Premarin)
1	0.238368	(abilify)
2	0.020397	(albuterol aerosol)
3	0.033329	(allopurinol)
4	0.079323	(alprazolam)
...	...	...
98	0.023064	(lisinopril, diazepam)
99	0.023464	(losartan, diazepam)
100	0.022930	(metoprolol, diazepam)
101	0.020131	(glyburide, doxycycline hyclate)
102	0.028530	(losartan, glyburide)

103 rows × 2 columns

```
#Create a rules table for data frame
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	(abilify)	(carvedilol)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008	0.400606
9	(carvedilol)	(abilify)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314	0.369437
10	(cialis)	(abilify)	0.076523	0.238368	0.023997	0.313589	1.315565	0.005756	1.109585	0.259747
11	(abilify)	(cialis)	0.238368	0.076523	0.023997	0.100671	1.315565	0.005756	1.026851	0.314943
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

3. As required in the rubric, I am including the rules table with values for support, lift, and confidence. This was also done in part C2 but will be included here again per the rubric.

```python
#Create a rules table for data frame
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_metri
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	0.006564	1.137144	0.29956
1	(abilify)	(amlodipine)	0.238368	0.071457	0.023597	0.098993	1.385352	0.006564	1.030562	0.36521
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	0.008095	1.183991	0.35614
3	(abilify)	(amphetamine salt combo)	0.238368	0.068391	0.024397	0.102349	1.496530	0.008095	1.037830	0.43562
4	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815	0.19364
5	(abilify)	(amphetamine salt combo xr)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158	0.20856
6	(atorvastatin)	(abilify)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650	0.40946
7	(abilify)	(atorvastatin)	0.238368	0.129583	0.047994	0.201342	1.553774	0.017105	1.089850	0.46795
8	(abilify)	(carvedilol)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008	0.40060
9	(carvedilol)	(abilify)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314	0.36943
10	(cialis)	(abilify)	0.076523	0.238368	0.023997	0.313589	1.315565	0.005756	1.109585	0.25974
11	(abilify)	(cialis)	0.238368	0.076523	0.023997	0.100671	1.315565	0.005756	1.026851	0.31494
12	(citalopram)	(abilify)	0.087188	0.238368	0.024397	0.279817	1.173883	0.003614	1.057552	0.16227
13	(abilify)	(citalopram)	0.238368	0.087188	0.024397	0.102349	1.173883	0.003614	1.016889	0.19448
14	(clopidogrel)	(abilify)	0.059992	0.238368	0.022797	0.380000	1.594172	0.008497	1.228438	0.39650
15	(abilify)	(clopidogrel)	0.238368	0.059992	0.022797	0.095638	1.594172	0.008497	1.039415	0.48936
16	(dextroamphetamine XR)	(abilify)	0.081056	0.238368	0.027463	0.338816	1.421397	0.008142	1.151921	0.32261
17	(abilify)	(dextroamphetamine XR)	0.238368	0.081056	0.027463	0.115213	1.421397	0.008142	1.038604	0.38925
18	(diazepam)	(abilify)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357	0.30896
19	(abilify)	(diazepam)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256	0.33919

4. The top three rules can be seen in the screenshot below. The top three transactions contain a lift threshold of 1.9 and a confidence of 0.3.

```python
#Sort with multi metric rules for top 3
sorted_rules=rul_table[(rul_table['lift']> 1.9) & (rul_table['confidence'] > 0.3)].sort_values(by= ['lift'], ascending=False)
sorted_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
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
30	(metformin)	(abilify)	0.050527	0.238368	0.023064	0.456464	1.914955	0.011020	1.401255	0.503221

Part IV: Data Summary and Implications

D. Summarize Data Analysis

1. The screenshots below provide context to summarize my analysis for this performance assessment. To answer my research question, I first wanted to see how often diazepam showed up based on the created rules table. I checked to see how often diazepam came up as both an antecedent and a consequent.

```
rul_table.antecedents.value_counts()

(abilify)                       18
(carvedilol)                    12
(amphetamine salt combo xr)      9
(diazepam)                       8
(atorvastatin)                   7
(glyburide)                      6
(metoprolol)                     5
(lisinopril)                     4
(doxycycline hyclate)            4
(losartan)                       4
(citalopram)                     4
(glipizide)                      2
(amphetamine salt combo)         2
(amlodipine)                     2
(dextroamphetamine XR)           1
(clopidogrel)                    1
(fenofibrate)                    1
(levofloxacin)                   1
(metformin)                      1
(cialis)                         1
(naproxen)                       1
Name: antecedents, dtype: int64
```

```
#Observe counts for diazepam as consequent in final set rules
rul_table.consequents.value_counts()

(abilify)                       18
(carvedilol)                    12
(amphetamine salt combo xr)      9
(diazepam)                       8
(atorvastatin)                   7
(glyburide)                      6
(metoprolol)                     5
(doxycycline hyclate)            4
(lisinopril)                     4
(losartan)                       4
(citalopram)                     4
(amlodipine)                     2
(amphetamine salt combo)         2
(glipizide)                      2
(dextroamphetamine XR)           1
(levofloxacin)                   1
(clopidogrel)                    1
(metformin)                      1
(naproxen)                       1
(cialis)                         1
(fenofibrate)                    1
Name: consequents, dtype: int64
```

With the screenshots above, diazepam appears 8 times as an antecedent and a consequent. My next step is to now produce a table that will allow me to determine positive medication relationships with diazepam.

```python
#Look at diazepam transactions
ant_df = rul_table[rul_table['antecedents'] == {'diazepam'}]
con_df = rul_table[rul_table['consequents'] == {'diazepam'}]
diazepam_df = pd.concat([ant_df, con_df])
diazepam_df
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
18	(diazepam)	(abilify)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357	0.308965
47	(diazepam)	(amphetamine salt combo xr)	0.163845	0.179709	0.033196	0.202604	1.127397	0.003751	1.028711	0.135143
59	(diazepam)	(atorvastatin)	0.163845	0.129583	0.032129	0.196094	1.513276	0.010898	1.082736	0.405645
68	(diazepam)	(carvedilol)	0.163845	0.174110	0.039195	0.239219	1.373952	0.010668	1.085581	0.325505
83	(diazepam)	(glyburide)	0.163845	0.170911	0.034395	0.209927	1.228284	0.006393	1.049383	0.222275
85	(diazepam)	(lisinopril)	0.163845	0.098254	0.023064	0.140765	1.432669	0.006965	1.049476	0.361180
87	(diazepam)	(losartan)	0.163845	0.132116	0.023464	0.143206	1.083943	0.001817	1.012944	0.092617
89	(diazepam)	(metoprolol)	0.163845	0.095321	0.022930	0.139951	1.468215	0.007312	1.051893	0.381390
19	(abilify)	(diazepam)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256	0.339197
46	(amphetamine salt combo xr)	(diazepam)	0.179709	0.163845	0.033196	0.184718	1.127397	0.003751	1.025603	0.137757
58	(atorvastatin)	(diazepam)	0.129583	0.163845	0.032129	0.247942	1.513276	0.010898	1.111823	0.389677
69	(carvedilol)	(diazepam)	0.174110	0.163845	0.039195	0.225115	1.373952	0.010668	1.079070	0.329550
82	(glyburide)	(diazepam)	0.170911	0.163845	0.034395	0.201248	1.228284	0.006393	1.046827	0.224169
84	(lisinopril)	(diazepam)	0.098254	0.163845	0.023064	0.234735	1.432669	0.006965	1.092635	0.334908
86	(losartan)	(diazepam)	0.132116	0.163845	0.023464	0.177598	1.083943	0.001817	1.016724	0.089231
88	(metoprolol)	(diazepam)	0.095321	0.163845	0.022930	0.240559	1.468215	0.007312	1.101015	0.352502

Looking at all diazepam transactions, based on the thresholds created in our rules table, the medication appears 8 times as an antecedent and 8 times as a consequent. This means that there are 8 transactions where if diazepam is prescribed as an antecedent, there are 8 other medications that are likely to be prescribed with it to a patient. Conversely, if any of those 8 other medications are prescribed to a patient, diazepam is likely to be prescribed as a consequent. The goal here is to determine where the higher positive medication relationship exists so that medical care providers or hospital executives can implement better care practices.

In the table above, certain metrics can help us determine where the most significant medication relationships exist, and the likelihood of those medications being prescribed together or at some point in the patient's treatment. Therefore, I will elaborate on what the support, confidence, and lift metrics mean to this analysis.

Support is a calculation for the proportion of all transactions that contain an association rule (Sivek 2020). For example, like the grocery store example, I provided early "if coffee the creamer". In the table above, we can see that of the 7501 transactions, 5.2% contain diazepam and Abilify, 3.3% contain diazepam and amphetamine salt combo xr, 3.2% contain diazepam and atorvastatin, and so forth.

Confidence is a proportion of transactions for a set association rule divided by the total of a single transaction itself, this is where we attain specificity to the judgment of an association rule (Sivek 2020). For example, in the table above we can see that, if diazepam then Abilify, there is a 32% confidence level or probability that if someone is prescribed diazepam, they will also be prescribed Abilify. Confidence levels closer to one are more significant.

Lift is the calculation where we exceed expectations of diazepam and Abilify, or any of the other 8 medication combos, being prescribed together. Some people may get prescribed diazepam only, some may get prescribed Abilify only, and some will be prescribed both. When our lift metric is above 1, this can tell the analyst that there is a positive relationship between the set association rule and the antecedent does indeed increase the likelihood of the consequent occurring in the transaction (Sivek 2020). For example, in the table above, if diazepam then Abilify has a lift of 1.34, indicating a positive relationship between those two prescriptions. Furthermore, if we look through our list a bit more, we can see that if diazepam then atorvastatin has a lift of 1.51, an even stronger relationship between the two medications being prescribed together.

2. When deciding which medication is the antecedent, looking at the confidence metric would provide an analyst with solidification. An example would be if diazepam then abilify, where the confidence metric is 32% an indication that more often than not diazepam will be prescribed first over abilify. With these results, it would come down to the medical care providers and hospital executives to decide which medical conditions they are most concerned with, and the medications prescribed for treatment to focus on for improving patient quality of care. Diazepam treats mood disorders like anxiety, depression, bipolar disorder, etc. Stakeholders interested in patient care and outcomes would have to determine what it is they want to improve or find out about medication relationships. As an example, what makes a patient need a mood disorder drug and a blood pressure drug? Is there a correlation between those medical conditions and the need for medications to treat them? If we were to take diazepam and then Abilify as our main drug relationship for research, I think the relationship is practically reasonable. Both drugs are used to treat mood disorders and it would make complete sense if a patient was on both medications to treat a severe mood disorder. Over 32% of transactions include diazepam, which is quite shocking, but this could indicate a large part of the population suffers from some sort of mood disorder.

3. If the hospital were solely concerned with the association of diazepam and Abilify, I think the best course of action would be to focus on what has caused mood disorders in some patients and evaluate the options to get patients off those medications at some point. The American Addiction Center states that long-term use of diazepam (Valium) can cause symptoms for which it was prescribed in the first place, such as anxiety and sleeplessness. There are generally alternative methods to treat and deal with conditions like anxiety and panic attacks, such as therapy and meditation, and promoting methods aside from medications could be beneficial to patient health in the long term. Seeing that 32% of patients are prescribed diazepam should alert healthcare providers and hospital executives that maybe the medication is being too commonly prescribed and meeting with doctors about such a metric can help reduce the number of patients being on such a dependent drug.

Part V: Attachments

E. Panopto Video

1. The link is included in the submission.

F. Sources

Hua. (2015, September 1). *Market basket analysis*. HUA'S Analysis. https://sarahtianhua.wordpress.com/portfolio/market-basket-analysis/#:~:text=The%20underlying%20assumption%20in%20market,lead%20to%20purchase%20of%20others.

Susan Currie Sivek, Ph. D. (2020, November 17). *Market basket analysis 101: Key concepts*. Medium. https://towardsdatascience.com/market-basket-analysis-101-key-concepts-1ddc6876cd00

Armstrong, L., Generes, W., Regan, J., Kelley, R., Fuller, K., & Ackermann, K. (2023, July 10). *Side effects of valium: Long-term use & addiction*. American Addiction Centers. https://americanaddictioncenters.org/valium-treatment/long-term