Market Basket Analysis of Drug Prescription Data

Tejaswini Shekar

Table of of Contents:

A. Research Question

1. Proposal of Question
2. Goal of Analysis

B. Market Basket Justification

1. Explanation of Market Basket
2. Market Basket Assumption

C. Data Preparation and Analysis

1. Transforming the Data Set
2. Generation of Association Rules
3. Association Rules Table
4. Top Three Rules

D. Data Summary and Implications

1. Analysis Results - Support, Lift, and Confidence
2. Practical Significance of Findings
3. Course of Action

E. Panopto

F. Code Sources

G. Other Sources

A1. Research Question

My research question is, "Can I use market basket analysis to find relationships/associations between drugs prescribed in the medical data set?"

A2. Goal of Analysis

The goal of the analysis is to identify relationships between the prescribed drugs and utilise this information to drive strategic decision-making in the hospital. Knowing which drug combinations are prescribed together most frequently can help ensure that the hospital pharmacy always has adequate inventory. Also, the relationships between the prescribed drugs can provide insight into any underlying trends in patient illnesses and potential areas that healthcare professionals can further investigate.

B1. Explanation of Market Basket Analysis

Market basket analysis is a data mining technique that finds the strength of product relationships by looking at previous transactions and identifying trends/patterns. It is based on association rules and is used for product placement in stores, product bundling, and customer retention. The Apriori algorithm is one association rule mining method that works by pruning.

The Apriori algorithm uses the Apriori principle. Its parameters are support, lift and confidence. The algorithm generates frequent itemsets from the data based on whether they have a support value greater than the minimum threshold value provided, called the "minsup". Then, it generates association rules from these itemsets. The Apriori principle states that all subsets of frequent itemsets must also be frequent. It is a result of the anti-monotone property of support, which means that if we drop an item from a set, the support value of the new itemset generated will either be the same or greater. For example, if the set {Drug_1, Drug_2} has a support value less than the minsup, any set (e.g. {Drug_1, Drug_2, Drug_3}) containing that subset will also have a support value below the minimum threshold. This principle makes searching for association rules in the data set more efficient.

The steps of analysis using the Apriori algorithm include itemset generation followed by rule generation.

1. Itemset Generation: First, all the frequent itemsets with with only one drug and support ≥ minsup are generated. Next, itemsets of length two are generated for all possible combinations of the first itemsets. The itemsets with support< minsup are pruned. Then, itemsets containing three drugs are generated from the 2-drug sets and pruned again. This process is repeated with increasing set lengths to obtain the most frequent items with support values greater than the threshold.

2. Rule Generation: After the frequent itemsets are generated, the possible association rules are generated by the binary partition of each itemset. The generated association rules are then pruned based on the threshold metric provided. The metric used in this analysis is lift, with a minimum threshold value of 1. The final association rules table generated can be sorted by various metrics to find the top rules. (Garg, 2018)

The expected outcome is to obtain a table of association rules for the prescribed drugs and rank these rules by lift to find the top three. The top rules will show which drugs have the strongest associations/relationships and can be used to drive decision-making in the hospital.

B2. Market Basket Assumption

Market basket analysis is based on the assumption that the presence of two or more items in a basket implies that these products complement each other and therefore, the purchase of one item leads to the purchase of other items. In this analysis, the assumption would be that if a prescription contains two or more drugs, it is because these drugs are related to each other in some way. The prescribing of one drug may in turn lead to the prescription of another (Hua's Analysis, n.d.).

C1. Transforming the Data Set

The data set will be transformed in the following steps:

1. Rows of null/nan values will be dropped.
2. The dataframe will be transformed into a list of lists containing the prescriptions.
3. Using TransactionEncoder, this list will be encoded and an array will be returned.

4. The resulting array will be converted into a dataframe with column headings representing the prescribed drugs and rows containing True/False values for each drug. True means that the drug was prescribed for the patient and False indicates that the drug was not prescribed.
5. Any columns of null values will be dropped from the dataframe.
6. The final cleaned and transformed dataframe will be exported as a csv file and uploaded along with the PA.

```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules

import warnings
warnings.filterwarnings("ignore")

# load csv file into dataframe
df = pd.read_csv("./medical_market_basket.csv")

# overview of data
df.head()
```

```
      Presc01            Presc02                    Presc03
Presc04  \
0         NaN                NaN                        NaN
NaN
1  amlodipine  albuterol aerosol                allopurinol
pantoprazole
2         NaN                NaN                        NaN
NaN
3  citalopram            benicar  amphetamine salt combo xr
NaN
4         NaN                NaN                        NaN
NaN

     Presc05     Presc06     Presc07      Presc08     Presc09
Presc10  \
0        NaN         NaN         NaN          NaN         NaN
NaN
```

```
1  lorazepam  omeprazole  mometasone  fluconozole  gabapentin
pravastatin
2        NaN         NaN         NaN          NaN         NaN
NaN
3        NaN         NaN         NaN          NaN         NaN
NaN
4        NaN         NaN         NaN          NaN         NaN
NaN

  Presc11   Presc12                  Presc13           Presc14
Presc15  \
0     NaN       NaN                      NaN               NaN
NaN
1  cialis  losartan  metoprolol succinate XL  sulfamethoxazole
abilify
2     NaN       NaN                      NaN               NaN
NaN
3     NaN       NaN                      NaN               NaN
NaN
4     NaN       NaN                      NaN               NaN
NaN

          Presc16        Presc17       Presc18       Presc19
Presc20
0             NaN            NaN           NaN           NaN
NaN
1  spironolactone  albuterol HFA  levofloxacin  promethazine
glipizide
2             NaN            NaN           NaN           NaN
NaN
3             NaN            NaN           NaN           NaN
NaN
4             NaN            NaN           NaN           NaN
NaN
```

```
# check shape of dataframe - 15,002 rows and 20 columns
df.shape
```

```
(15002, 20)
```

```
# example of transaction/prescription from the data set
df.iloc[1]
```

```
Presc01                 amlodipine
Presc02         albuterol aerosol
Presc03                allopurinol
Presc04               pantoprazole
Presc05                  lorazepam
Presc06                 omeprazole
Presc07                 mometasone
```

```
Presc08                fluconozole
Presc09                 gabapentin
Presc10                pravastatin
Presc11                     cialis
Presc12                   losartan
Presc13    metoprolol succinate XL
Presc14          sulfamethoxazole
Presc15                    abilify
Presc16            spironolactone
Presc17              albuterol HFA
Presc18                levofloxacin
Presc19                promethazine
Presc20                  glipizide
Name: 1, dtype: object
```

```python
# check for null values, data types of columns
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
```

```python
# remove null values
df = df[df["Presc01"].notna()]
```

```python
# ensure null values have been dropped
df.head(10)
```

```
                  Presc01              Presc02                     Presc03  \
1              amlodipine    albuterol aerosol                 allopurinol   
3              citalopram              benicar   amphetamine salt combo xr   
5               enalapril                  NaN                         NaN   
7              paroxetine          allopurinol                         NaN   
9                 abilify         atorvastatin                  folic acid   
11                 cialis                  NaN                         NaN   
13  hydrochlorothiazide            glyburide                         NaN   
15              metformin   salmeterol inhaler              sertraline HCI   
17             metoprolol           carvedilol                    losartan   
19              glyburide                  NaN                         NaN   

         Presc04    Presc05     Presc06     Presc07      Presc08    Presc09  \
1   pantoprazole  lorazepam  omeprazole  mometasone  fluconozole  gabapentin   
3            NaN        NaN         NaN         NaN          NaN        NaN   
5            NaN        NaN         NaN         NaN          NaN        NaN   
7            NaN        NaN         NaN         NaN          NaN        NaN   
9       naproxen   losartan         NaN         NaN          NaN        NaN   
11           NaN        NaN         NaN         NaN          NaN        NaN   
13           NaN        NaN         NaN         NaN          NaN        NaN   
15           NaN        NaN         NaN         NaN          NaN        NaN   
17           NaN        NaN         NaN         NaN          NaN        NaN   
19           NaN        NaN         NaN         NaN          NaN        NaN   

        Presc10 Presc11   Presc12                  Presc13    Presc14  \
```

```
1   pravastatin  cialis  losartan  metoprolol succinate XL
sulfamethoxazole
3           NaN     NaN       NaN                       NaN
NaN
5           NaN     NaN       NaN                       NaN
NaN
7           NaN     NaN       NaN                       NaN
NaN
9           NaN     NaN       NaN                       NaN
NaN
11          NaN     NaN       NaN                       NaN
NaN
13          NaN     NaN       NaN                       NaN
NaN
15          NaN     NaN       NaN                       NaN
NaN
17          NaN     NaN       NaN                       NaN
NaN
19          NaN     NaN       NaN                       NaN
NaN

    Presc15          Presc16        Presc17       Presc18       Presc19
\
1   abilify  spironolactone  albuterol HFA  levofloxacin  promethazine

3       NaN             NaN            NaN           NaN           NaN

5       NaN             NaN            NaN           NaN           NaN

7       NaN             NaN            NaN           NaN           NaN

9       NaN             NaN            NaN           NaN           NaN

11      NaN             NaN            NaN           NaN           NaN

13      NaN             NaN            NaN           NaN           NaN

15      NaN             NaN            NaN           NaN           NaN

17      NaN             NaN            NaN           NaN           NaN

19      NaN             NaN            NaN           NaN           NaN


      Presc20
1   glipizide
3         NaN
5         NaN
7         NaN
9         NaN
11        NaN
```

```
13        NaN
15        NaN
17        NaN
19        NaN
```

```python
# check shape of dataframe
df.shape
```

```
(7501, 20)
```

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

# use transaction encoder to encode the list
encoder = TransactionEncoder()
array = encoder.fit(rows).transform(rows)

# convert array to dataframe
df_encoded = pd.DataFrame(array, columns = encoder.columns_)

#check column names of dataframe
for col in df_encoded.columns:
    print(col)
```

```
Duloxetine
Premarin
Yaz
abilify
acetaminophen
actonel
albuterol HFA
albuterol aerosol
alendronate
allopurinol
alprazolam
amitriptyline
amlodipine
amoxicillin
amphetamine
amphetamine salt combo
amphetamine salt combo xr
atenolol
atorvastatin
azithromycin
benazepril
benicar
boniva
bupropion sr
```

```
carisoprodol
carvedilol
cefdinir
celebrex
celecoxib
cephalexin
cialis
ciprofloxacin
citalopram
clavulanate K+
clonazepam
clonidine HCI
clopidogrel
clotrimazole
codeine
crestor
cyclobenzaprine
cymbalta
dextroamphetamine XR
diazepam
diclofenac sodium
doxycycline hyclate
enalapril
escitalopram
esomeprazole
ezetimibe
fenofibrate
fexofenadine
finasteride
flovent hfa 110mcg inhaler
fluconozole
fluoxetine HCI
fluticasone
fluticasone nasal spray
folic acid
furosemide
gabapentin
glimepiride
glipizide
glyburide
hydrochlorothiazide
hydrocodone
hydrocortisone 2.5% cream
ibuprophen
isosorbide mononitrate
lansoprazole
lantus
levofloxacin
levothyroxine sodium
```

```
lisinopril
lorazepam
losartan
lovastatin
meloxicam
metformin
metformin HCI
methylprednisone
metoprolol
metoprolol succinate XL
metoprolol tartrate
mometasone
nan
naproxen
omeprazole
oxycodone
pantoprazole
paroxetine
pioglitazone
potassium Chloride
pravastatin
prednisone
pregabalin
promethazine
quetiapine
ranitidine
rosuvastatin
salmeterol inhaler
sertraline HCI
simvastatin
spironolactone
sulfamethoxazole
synthroid
tamsulosin
temezepam
topiramate
tramadol
trazodone HCI
triamcinolone Ace topical
triamterene
trimethoprim DS
valaciclovir
valsartan
venlafaxine XR
verapamil SR
viagra
zolpidem

# a nan column is present
# drop nan column
```

```
df_final = df_encoded.drop(['nan'], axis = 1)
df_final
```

```
      Duloxetine  Premarin    Yaz  abilify  acetaminophen  actonel  \
0          False     False  False     True          False    False
1          False     False  False    False          False    False
2          False     False  False    False          False    False
3          False     False  False    False          False    False
4          False     False  False     True          False    False
...          ...       ...    ...      ...            ...      ...
7496       False     False  False    False          False    False
7497       False     False  False    False          False    False
7498       False     False  False    False          False    False
7499       False     False  False    False          False    False
7500       False     False  False    False          False    False

      albuterol HFA  albuterol aerosol  alendronate  allopurinol  ...
\
0              True               True        False         True  ...

1             False              False        False        False  ...

2             False              False        False        False  ...

3             False              False        False         True  ...

4             False              False        False        False  ...

...             ...                ...          ...          ...  ...

7496          False              False        False        False  ...

7497          False              False        False        False  ...

7498          False              False        False        False  ...

7499          False              False        False        False  ...

7500          False              False        False        False  ...


      trazodone HCI  triamcinolone Ace topical  triamterene
trimethoprim DS  \
0             False                      False        False
False
1             False                      False        False
False
2             False                      False        False
False
3             False                      False        False
False
```

```
4               False                            False        False
False
...               ...                              ...          ...
...
7496            False                            False        False
False
7497            False                            False        False
False
7498            False                            False        False
False
7499            False                            False        False
False
7500            False                            False        False
False

      valaciclovir  valsartan  venlafaxine XR  verapamil SR  viagra
zolpidem
0            False      False           False         False   False
False
1            False      False           False         False   False
False
2            False      False           False         False   False
False
3            False      False           False         False   False
False
4            False      False           False         False   False
False
...            ...        ...             ...           ...     ...
...
7496         False      False           False         False   False
False
7497         False      False           False         False   False
False
7498         False      False           False         False   False
False
7499         False      False           False         False   False
False
7500         False      False           False         False   False
False

[7501 rows x 119 columns]
```

```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7501 entries, 0 to 7500
Columns: 119 entries, Duloxetine to zolpidem
dtypes: bool(119)
memory usage: 871.8 KB
```

```python
# export final dataframe to csv file
df_final.to_csv("./transformed_medical.csv")
```

C2. Generation of Association Rules

The cleaned and transformed data set will be mined using the Apriori algorithm to generate association rules. The minimum threshold value of support for generating frequent itemsets is 0.02. The minimum threshold of lift for generating association rules is 1.

```python
# generate frequent itemsets using apriori algorithm
frequent_sets = apriori(df_final, min_support = 0.02,
use_colnames=True)
frequent_sets
```

```
      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              (diazepam, losartan)
100  0.022930            (metoprolol, diazepam)
101  0.020131  (doxycycline hyclate, glyburide)
102  0.028530             (glyburide, losartan)

[103 rows x 2 columns]
```

```python
# generate association rules with lift > 1
rules = association_rules(frequent_sets, metric = 'lift',
min_threshold = 1, num_itemsets=len(df_final))
rules
```

```
                  antecedents                  consequents  antecedent
support  \
0                (amlodipine)                    (abilify)
0.071457
1                   (abilify)                 (amlodipine)
0.238368
2    (amphetamine salt combo)                    (abilify)
0.068391
3                   (abilify)     (amphetamine salt combo)
0.238368
4                   (abilify)  (amphetamine salt combo xr)
0.238368
..                        ...                          ...
...
89                 (diazepam)                 (metoprolol)
0.163845
```

```
90     (doxycycline hyclate)                  (glyburide)
0.095054
91              (glyburide)        (doxycycline hyclate)
0.170911
92              (glyburide)                   (losartan)
0.170911
93               (losartan)                  (glyburide)
0.132116

    consequent support   support  confidence      lift
representativity  \
0             0.238368  0.023597    0.330224  1.385352
1.0
1             0.071457  0.023597    0.098993  1.385352
1.0
2             0.238368  0.024397    0.356725  1.496530
1.0
3             0.068391  0.024397    0.102349  1.496530
1.0
4             0.179709  0.050927    0.213647  1.188845
1.0
..                 ...       ...         ...       ...               .
..
89            0.095321  0.022930    0.139951  1.468215
1.0
90            0.170911  0.020131    0.211781  1.239135
1.0
91            0.095054  0.020131    0.117785  1.239135
1.0
92            0.132116  0.028530    0.166927  1.263488
1.0
93            0.170911  0.028530    0.215943  1.263488
1.0

    leverage  conviction  zhangs_metric   jaccard  certainty
kulczynski
0   0.006564    1.137144       0.299568  0.082441   0.120604
0.214609
1   0.006564    1.030562       0.365218  0.082441   0.029655
0.214609
2   0.008095    1.183991       0.356144  0.086402   0.155399
0.229537
3   0.008095    1.037830       0.435627  0.086402   0.036451
0.229537
4   0.008090    1.043158       0.208562  0.138707   0.041372
0.248515
..       ...         ...            ...       ...        ...          .
..
89  0.007312    1.051893       0.381390  0.097065   0.049333
```

```
0.190255
90  0.003885    1.051852        0.213256  0.081887   0.049296
0.164783
91  0.003885    1.025766        0.232768  0.081887   0.025118
0.164783
92  0.005950    1.041786        0.251529  0.103934   0.040110
0.191435
93  0.005950    1.057436        0.240286  0.103934   0.054316
0.191435

[94 rows x 14 columns]
```

C3. Association Rules Table

The association rules table is provided in this section again to comply with the rubric. The metric used to generate the association rules was lift, with a minimum threshold value of 1. A lift of atleast 1 indicates that the presence of the antecedent *increases* the liklihood of occurrence of the consequent. A total of 95 association rules were generated.

NOTE: I have explicitly mentioned the value for the parameter "num_itemsets" since the latest version of mlxtend throws an error without it. The value of num_itemsets should be the number of transactions in the input data, (Stack Overflow, 2024).

```
rules = association_rules(frequent_sets, metric = 'lift',
min_threshold = 1, num_itemsets=len(df_final))
rules

                 antecedents                    consequents  antecedent
support  \
0               (amlodipine)                      (abilify)
0.071457
1                  (abilify)                   (amlodipine)
0.238368
2    (amphetamine salt combo)                     (abilify)
0.068391
3                  (abilify)       (amphetamine salt combo)
0.238368
4                  (abilify)    (amphetamine salt combo xr)
0.238368
..                       ...                            ...
...
89                (diazepam)                   (metoprolol)
0.163845
90     (doxycycline hyclate)                    (glyburide)
0.095054
91               (glyburide)          (doxycycline hyclate)
0.170911
92               (glyburide)                    (losartan)
0.170911
```

```
93                (losartan)                  (glyburide)
0.132116

    consequent support   support  confidence      lift
representativity  \
0              0.238368  0.023597    0.330224  1.385352
1.0
1              0.071457  0.023597    0.098993  1.385352
1.0
2              0.238368  0.024397    0.356725  1.496530
1.0
3              0.068391  0.024397    0.102349  1.496530
1.0
4              0.179709  0.050927    0.213647  1.188845
1.0
..                  ...       ...         ...       ...               .
..
89             0.095321  0.022930    0.139951  1.468215
1.0
90             0.170911  0.020131    0.211781  1.239135
1.0
91             0.095054  0.020131    0.117785  1.239135
1.0
92             0.132116  0.028530    0.166927  1.263488
1.0
93             0.170911  0.028530    0.215943  1.263488
1.0

    leverage  conviction  zhangs_metric   jaccard  certainty
kulczynski
0   0.006564    1.137144       0.299568  0.082441   0.120604
0.214609
1   0.006564    1.030562       0.365218  0.082441   0.029655
0.214609
2   0.008095    1.183991       0.356144  0.086402   0.155399
0.229537
3   0.008095    1.037830       0.435627  0.086402   0.036451
0.229537
4   0.008090    1.043158       0.208562  0.138707   0.041372
0.248515
..       ...         ...            ...       ...        ...          .
..
89  0.007312    1.051893       0.381390  0.097065   0.049333
0.190255
90  0.003885    1.051852       0.213256  0.081887   0.049296
0.164783
91  0.003885    1.025766       0.232768  0.081887   0.025118
0.164783
92  0.005950    1.041786       0.251529  0.103934   0.040110
```

```
0.191435
93  0.005950    1.057436        0.240286  0.103934   0.054316
0.191435

[94 rows x 14 columns]
```

C4. Top Three Rules

The top three association rules were found by sorting the rules by their lift in descending order:

1. If carvedilol, then lisinopril (lift: 2.291162)
2. If lisinopril, then carvedilol (lift: 2.291162)
3. If glipizide, then carvedilol (lift: 1.999758)

```
# find top 3 rules
top_3_rules = rules.sort_values("lift", ascending=False).head(3)
top_3_rules

     antecedents   consequents  antecedent support  consequent support
\
75  (carvedilol)  (lisinopril)            0.174110            0.098254

74  (lisinopril)  (carvedilol)            0.098254            0.174110

72   (glipizide)  (carvedilol)            0.065858            0.174110


     support  confidence      lift  representativity  leverage
conviction  \
75  0.039195    0.225115  2.291162               1.0  0.022088
1.163716
74  0.039195    0.398915  2.291162               1.0  0.022088
1.373997
72  0.022930    0.348178  1.999758               1.0  0.011464
1.267048

    zhangs_metric   jaccard  certainty  kulczynski
75       0.682343  0.168096   0.140684    0.312015
74       0.624943  0.168096   0.272197    0.312015
72       0.535186  0.105651   0.210764    0.239939
```

D1. Analysis Results - Support, Confidence, and Lift

The results of my analysis show that the top three rules, ranked by lift, are

1. If carvedilol, then lisinopril (support: 0.039195, confidence: 0.225115, lift: 2.291162)
2. If lisinopril, then carvedilol (support: 0.039195, confidence: 0.398915, lift: 2.291162)
3. If glipizide, then carvedilol (support: 0.022930, confidence: 0.348178, lift: 1.999758)

Support: Support is the frequency of occurrence of an itemset. It is the fraction of transactions in which the itemset is seen. The support value helps in identifying rules worth considering for further analysis. If a rule has a very low support, there isn't enough information on the relationship between the items and therefore no conclusions can be drawn from it. I used a minimum support threshold value of 0.02 in the Apriori algorithm to obtain the most frequent itemsets from the data.

Support(X->Y) = (Transactions Containing both X and Y)/(Total Number of Transactions)

The support (frequency of occurrence) of the top 3 rules in the data set are 0.039195 for the first two rules and 0.022930 for the third rule. The support for the first two rules is the same since they have the same itemset. The formula for support does not change based on which item is the antecedent/consequent.

Confidence: Confidence is the liklihood of occurrence of the consequent Y, given that the antecedent X is already present. It is the conditional probability that drug Y will also be prescribed given that drug X is already prescribed, or, the fraction of X-containing prescriptions that also have drug Y. However, the confidence of any rule with a frequently occurring consequent will be high, regardless of what the antecedent is. This drawback is overcome by using lift. (Garg, 2018) Confidence can be calculated mathematically by:

Confidence(X->Y) = (Transactions Containing both X and Y)/ (Transactions Containing X)

Confidence(X->Y) = Support(X->Y)/ Support(X)

The results of the analysis show that the confidence of the top three rules are 0.225115, 0.398915 and 0.348178 respectively. For the first rule, this means that 0.225 (or 22.5%) of all prescriptions containing carvedilol also have lisinopil. Similarly, 0.398 of all lisinopril prescriptions (second rule) and 0.348 of all glipizide prescriptions (third rule) also have carvedilol.

Lift: Lift is the conditional probability of a consequent Y given antecedent X, divided by the frequency of occurrence of Y. It can also be expressed as the Confidence of {X}->{Y} over Support of Y.

Lift(X->Y) = ((Transactions Containing both X and Y)/(Transactions Containing X)) / (Fraction of Transactions Containing Y)

Lift(X->Y) = Confidence(X->Y)/ Support(Y) = Support(X->Y)/ (Support(X))(Support(Y))

If the rule "If X -> Then Y" is true, then the value of lift will be greater than 1. (Garg, 2018) A minimum lift threshold value of 1 was set when generating the association rules.

The first 2 of the top 3 rules have the same lift (2.291162) because they have the same itemsets (antecedent and consequent are interchanged and this does not affect calculation of lift). The third top rule has a lift of 1.999758.

The first rule can be interpreted as *linsinopril being 2.29 times more likely to be prescribed when the patient has been prescribed carvedilol*, and vice versa where *carvedilol is 2.29 times more likely to be prescribed when the patient has been prescribed linsinopril.* The third rule indicates that *carvedilol is 1.9997 times more likely to be prescribed when glipizide has been prescribed.*

D2. Practical Significance of Findings

Based on the results of the analysis, it is clear that carvedilol and lisinopril are often prescribed together. This makes sense logically since they are both medications used in the treatment of cardiovascular diseases. A patient with heart disease or any cardiovascular illness may need a combination of medications to treat their condition, and this is reflected in the analysis results. The third rules shows that glipizide, an anti-diabetic medication, is prescribed with carvedilol often as well. This is also no surprise since diabetes increases the risk for cardiovascular illnesses, and patients with one condition may also have the other (American Heart Association, n.d.)

Practically, these results indicate that the hospital drug dispensary should keep carvedilol and lisinopril in inventory together, as one is usally prescribed along with the other and a low stock of carvedilol probably means a low stock of lisinopril as well. This will ensure that patients get all their medications efficiently and in a timely manner.

Another point of practical significance is that based on the pattern of drug prescription, doctors can conclude that many patients suffering from cardiovascular illnesses may also need to be routinely tested for diabetes.

D3. Course of Action

The next course of action would be to delve deeper into the top association rules and uncover more patterns in drug prescriptions. These insights can aid in efficient management of hospital drug inventory and provide additional insight into the trends of illnesses among patients. I would also recommend that the hospital implement a policy of routine testing of blood sugar levels for all patients with chronic cardiovascular disease for early detection of diabetes.

E. Panopto

Link to Panopto recording: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=19a76548-c979-4f4e-9d0d-b23f00cafc91

F. Code Sources

1. Stack Overflow. (November 2024.) *association_rules() Positional argument problem.* https://stackoverflow.com/questions/79195649/association-rules-positional-argument-problem.

G. Other Sources

1. Garg, A. (September 4, 2018). *Complete guide to Association Rules (1/2).* Medium. https://towardsdatascience.com/association-rules-2-aa9a77241654.

2. Hua's Analysis. (n.d.) *Market Basket Analysis.* https://sarahtianhua.wordpress.com/portfolio/market-basket-analysis/#:~:text=The%20underlying%20assumption%20in%20market,lead%20to%20purchase%20of%20others.

3. Garg, A. (September 17, 2018). *Complete guide to Association Rules (2/2)*. Medium. https://towardsdatascience.com/complete-guide-to-association-rules-2-2-c92072b56c84.

4. American Heart Association. (n.d.) *Cardiovascular Disease and Diabetes.* https://www.heart.org/en/health-topics/diabetes/diabetes-complications-and-risks/cardiovascular-disease--diabetes.