D212_PA3

April 4, 2024

1 D212 PA 3 code - Doug Haunsperger

1.1 Data Preparation

1.1.1 Do initial package import and data read

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Install package if needed for environment
#import sys
#!{sys.executable} -m pip install mlxtend

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

df = pd.read_csv('medical_market_basket_nonblank.csv')
#view first 5 rows
df.head(5)
```

[1]:
```
   Presc01                    Presc02      Presc03                    Presc04  \
0  abilify               atorvastatin   folic acid                   naproxen
1  abilify             spironolactone          NaN                        NaN
2  abilify                        NaN          NaN                        NaN
3  abilify  amphetamine salt combo xr  clopidogrel                   diazepam
4  abilify                   diazepam  allopurinol  amphetamine salt combo xr

     Presc05 Presc06 Presc07 Presc08 Presc09 Presc10 Presc11 Presc12 Presc13  \
0   losartan     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
1        NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
2        NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
3  glyburide     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
4        NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN

  Presc14 Presc15 Presc16 Presc17 Presc18 Presc19 Presc20
0     NaN     NaN     NaN     NaN     NaN     NaN     NaN
```

```
1     NaN     NaN     NaN     NaN     NaN     NaN     NaN
2     NaN     NaN     NaN     NaN     NaN     NaN     NaN
3     NaN     NaN     NaN     NaN     NaN     NaN     NaN
4     NaN     NaN     NaN     NaN     NaN     NaN     NaN
```

[2]:
```
# Convert DF to list of lists for transactionEncoder, dropping NaN values
# Code ref: https://stackoverflow.com/a/58436257 (Boston, 2019).

df_out = df.apply(lambda x: list(x.dropna().values), axis=1).tolist()

te = TransactionEncoder()
te_ary = te.fit(df_out).transform(df_out)
enc_df = pd.DataFrame(te_ary, columns=te.columns_)

enc_df.head()
```

[2]:
```
   Duloxetine  Premarin    Yaz  abilify  acetaminophen  actonel  \
0       False     False  False     True          False    False
1       False     False  False     True          False    False
2       False     False  False     True          False    False
3       False     False  False     True          False    False
4       False     False  False     True          False    False

   albuterol HFA  albuterol aerosol  alendronate  allopurinol  …  \
0          False              False        False        False  …
1          False              False        False        False  …
2          False              False        False        False  …
3          False              False        False        False  …
4          False              False        False         True  …

   trazodone HCl  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

   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

[5 rows x 119 columns]
```

1.1.2 Output cleaned and encoded data set

[3]:
```
enc_df.to_csv('enc_medical_markbask.csv', index=False)
```

1.2 Data Analysis

1.2.1 Perform market basket analysis

[4]:
```
# Define minimum number of patients using a drug or drug combo to consider

min_patients = 5
tot_patients = enc_df.shape[1]


frequent_itemsets = apriori(enc_df, min_support= min_patients/tot_patients,␣
 ↪use_colnames=True, max_len = 3)
rules = association_rules(frequent_itemsets, metric="lift",  min_threshold = 1)

display(frequent_itemsets.sort_values(by='support', ascending=False))

print("Rules identified: ", len(rules))
display(rules.sort_values(by='zhangs_metric', ascending=False))
```

	support	itemsets
1	0.238368	(abilify)
5	0.179709	(amphetamine salt combo xr)
7	0.174110	(carvedilol)
18	0.170911	(glyburide)
12	0.163845	(diazepam)
22	0.132116	(losartan)
6	0.129583	(atorvastatin)
21	0.098254	(lisinopril)
25	0.095321	(metoprolol)
13	0.095054	(doxycycline hyclate)
9	0.087188	(citalopram)
11	0.081056	(dextroamphetamine XR)
14	0.080389	(ezetimibe)
2	0.079323	(alprazolam)
8	0.076523	(cialis)
3	0.071457	(amlodipine)
4	0.068391	(amphetamine salt combo)
17	0.065858	(glipizide)
20	0.063325	(levofloxacin)
28	0.062525	(paroxetine)
10	0.059992	(clopidogrel)
32	0.059725	(abilify, carvedilol)
27	0.058526	(naproxen)
33	0.052660	(abilify, diazepam)

```
16  0.052393                          (furosemide)
15  0.051060                         (fenofibrate)
30  0.050927  (amphetamine salt combo xr, abilify)
23  0.050527                           (metformin)
24  0.049460                     (methylprednisone)
31  0.047994                (abilify, atorvastatin)
26  0.047460             (metoprolol succinate XL)
0   0.046794                               (Premarin)
19  0.043061                                 (lantus)
29  0.042528                       (spironolactone)
```

```
Rules identified:  8
```

```
                   antecedents                 consequents  \
2                    (abilify)               (atorvastatin)
3                (atorvastatin)                    (abilify)
4                    (abilify)                  (carvedilol)
5                  (carvedilol)                    (abilify)
6                    (abilify)                   (diazepam)
7                   (diazepam)                    (abilify)
1                    (abilify)  (amphetamine salt combo xr)
0  (amphetamine salt combo xr)                    (abilify)

   antecedent support  consequent support   support  confidence      lift  \
2            0.238368            0.129583  0.047994    0.201342  1.553774
3            0.129583            0.238368  0.047994    0.370370  1.553774
4            0.238368            0.174110  0.059725    0.250559  1.439085
5            0.174110            0.238368  0.059725    0.343032  1.439085
6            0.238368            0.163845  0.052660    0.220917  1.348332
7            0.163845            0.238368  0.052660    0.321400  1.348332
1            0.238368            0.179709  0.050927    0.213647  1.188845
0            0.179709            0.238368  0.050927    0.283383  1.188845

   leverage  conviction  zhangs_metric
2  0.017105    1.089850       0.467950
3  0.017105    1.209650       0.409465
4  0.018223    1.102008       0.400606
5  0.018223    1.159314       0.369437
6  0.013604    1.073256       0.339197
7  0.013604    1.122357       0.308965
1  0.008090    1.043158       0.208562
0  0.008090    1.062815       0.193648
```

[ ]: