Performance Assessment: Association Rules and Lift Analysis

Gabriela Howell

Master of Science Data Analytics, Western Governors University

D212 – Association Rules and Lift Analysis

Professor Middleton

September 20, 2024

Part I: Research Question

A. 1. Propose one question relevant to a real-world organizational situation that you will answer using market basket analysis.

My research question for this analysis is: "Identify which medications or vitamins are positively associated with the purchase or prescription of ‘Abilify’."

A. 2. Define one goal of the data analysis. Ensure your goal is reasonable within the scope of the selected scenario and is represented in the available data.

This analysis focuses on identifying patterns related to the prescription of Abilify, an antipsychotic medication used to address chemical imbalances in the brain, which is beneficial for patients with bipolar disorder, schizophrenia, Tourette syndrome, and autism (Kuo & Epperson, 2020). By exploring its associations with other prescriptions, procedures, and hospital services, the hospital can improve treatment management, optimize care plans, and use resources more effectively. This approach is intended to enhance patient outcomes and support more informed decision-making.

Part II: Market Basket Justification

B. 1. Explain how market basket analyzes the selected data set. Include expected outcomes.

Market basket analysis identifies patterns in data to find items that frequently appear together. In my study, I will analyze medical procedures, prescriptions, and treatments to uncover common combinations, aiding in resource planning, service bundling, and enhancing patient care. The goal is to discover association rules that indicate which services are often provided together.

Using the Apriori algorithm, I will identify associations between items, particularly prescriptions, based on their co-occurrence in transactions. The algorithm calculates key metrics such as support (frequency of a combination), confidence (likelihood of one item leading to another), and lift (strength of the association) to reveal patterns. The expected outcome is to find common prescription combinations and treatments, which will help the hospital better plan resources, enhance services, and improve patient care.

B. 2. Provide one example of transactions in the data set.

As shown in the image, the dataset includes up to 20 different vitamins/prescriptions (Presc01 - Presc20). In this specific screenshot, five prescriptions are present: ‘abilify’, ‘atorvastatin’, ‘folic acid’, ‘naproxen’, and ‘losartan’. Each row represents a unique transaction involving the purchase of at least 1 and up to 20 prescriptions. The market basket analysis will identify associations between prescriptions commonly purchased together.

```
]: # Example of a Transaction
   medical.iloc[9]

]: Presc01          abilify
   Presc02     atorvastatin
   Presc03       folic acid
   Presc04         naproxen
   Presc05         losartan
   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: 9, dtype: object
```

B. 3. Summarize one assumption of market basket analysis.

Market basket analysis examines prescription data to identify patterns in how medications are prescribed together. It operates on the assumption that if prescriptions frequently appear together in multiple transactions, they may be correlated meaningfully. This facilitates healthcare providers to foresee future prescribing behaviors. Each transaction is considered independent, ensuring that external factors like discounts or promotions do not influence the findings (Agrawal & Srikant, 1994). Additionally, MBA assumes that prescribing one medication can increase the likelihood of prescribing another, which is beneficial for optimizing inventory, treatment plans, and suggestions. In healthcare, analyzing these prescription patterns can improve decision-making for patient care. Overall, the analysis relies on metrics such as confidence and support to gauge the strength of these associations (Agrawal & Srikant, 1994).

Part III: Data Preparation and Analysis

C. 1. Transform the data set to make it suitable for market basket analysis. Include a copy of the cleaned data set.

When preparing the dataset for market basket analysis, I first cleaned and transformed the data into a format appropriate for the Apriori algorithm. Each row in the dataset corresponds to a unique transaction, with columns showing whether specific prescriptions are present (True/False or 1/0). The dataset was encoded to show each prescription (e.g., ‘abilify’, ‘atorvastatin’) as a boolean value, indicating if it was prescribed in a transaction. Irrelevant or duplicate data was removed. The final cleaned dataset, called ‘cleaned_encoded_prescriptions.csv’, contains binary values for each prescription, helping to identify patterns in prescription behavior using association rule mining.

```
# Get a count of the NaN
print("\nMissing values per column:")
print(medical.isna().sum())
```

```
Missing values per column:
Presc01     7501
Presc02     9255
Presc03    10613
Presc04    11657
Presc05    12473
Presc06    13138
Presc07    13633
Presc08    14021
Presc09    14348
Presc10    14607
Presc11    14746
Presc12    14848
Presc13    14915
Presc14    14955
Presc15    14977
Presc16    14994
Presc17    14998
Presc18    14998
Presc19    14999
Presc20    15001
dtype: int64
```

```
# Remove blank lines where 'Presc01' is NaN
medical = medical[medical['Presc01'].notna()]

# Reset the index to ensure we don't skip every other row
medical.reset_index(drop=True, inplace=True)
medical.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7501 entries, 0 to 7500
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: 1.1+ MB
```

```
# Check the first few rows of the cleaned dataset
print("First few rows of the medical dataset after cleaning:")
print(medical.head())
print(medical.info())
```

```
First few rows of the medical dataset after cleaning:
      Presc01          Presc02                   Presc03       Presc04  \
0  amlodipine  albuterol aerosol               allopurinol  pantoprazole
1  citalopram            benicar  amphetamine salt combo xr           NaN
2   enalapril                NaN                       NaN           NaN
3  paroxetine        allopurinol                       NaN           NaN
4     abilify       atorvastatin                folic acid      naproxen

     Presc05     Presc06     Presc07      Presc08     Presc09      Presc10  \
0  lorazepam  omeprazole  mometasone  fluconozole  gabapentin  pravastatin
1        NaN         NaN         NaN          NaN         NaN          NaN
2        NaN         NaN         NaN          NaN         NaN          NaN
3        NaN         NaN         NaN          NaN         NaN          NaN
4   losartan         NaN         NaN          NaN         NaN          NaN

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

         Presc16        Presc17       Presc18       Presc19    Presc20
0  spironolactone  albuterol HFA  levofloxacin  promethazine  glipizide
1             NaN            NaN           NaN           NaN        NaN
2             NaN            NaN           NaN           NaN        NaN
3             NaN            NaN           NaN           NaN        NaN
4             NaN            NaN           NaN           NaN        NaN
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7501 entries, 0 to 7500
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: 1.1+ MB
None
```

```
# Create the list of lists
large_list = []

# Iterate through each row in the DataFrame
for row_number in range(len(medical)):
    # Generate a temporary tiny list for each row
    tiny_list = []

    # Iterate through each prescription (column) in the row
    for cell in range(len(medical.columns)):
        # Check if the cell is not null/NaN, and if true, add to the temporary tiny list
        if pd.notnull(medical.iloc[row_number, cell]):
            tiny_list.append(str(medical.iloc[row_number, cell]))

    # Append the filled tiny list to the large list
    if tiny_list:  # Only append if there is data in the row
        large_list.append(tiny_list)

# Check the output of the list of lists
print(f"Sample of list of lists:\nRow 0: {large_list[0]}\nRow 1: {large_list[-1]}\n...")
```

```
Sample of list of lists:
Row 0: ['amlodipine', 'albuterol aerosol', 'allopurinol', 'pantoprazole', 'lorazepam', 'omeprazole', 'mometasone'
'promethazine', 'glipizide']
Row 1: ['amphetamine salt combo xr', 'levofloxacin', 'diclofenac sodium', 'cialis']
...
```

```
# Initialize the TransactionEncoder
encoder = TransactionEncoder()

# Fit and transform the data
temp_encoded = encoder.fit(large_list).transform(large_list)

# Create a new DataFrame with the encoded data
encoded_medical = pd.DataFrame(temp_encoded, columns=encoder.columns_)

# Check the cleaned Data
encoded_medical
```

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendron
0	False	False	False	True	False	False	True	True	F
1	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	F
4	False	False	False	True	False	False	False	False	F
...	...	...	...	...	...	...	...	...	
7496	False	False	False	False	False	False	False	False	F
7497	False	False	False	False	False	False	False	False	F
7498	False	False	False	False	False	False	False	False	F
7499	False	False	False	False	False	False	False	False	F
7500	False	False	False	False	False	False	False	False	F

7501 rows × 119 columns

C. 2. Execute the code used to generate association rules with the Apriori algorithm. Provide screenshots that demonstrate that the code is error free.

I used the Apriori algorithm with Python to find frequent prescription patterns and calculate their support, confidence, and lift, using the mlxtend library. The code ran successfully without any errors, as shown in the screenshot of the terminal output.

```
# Apply apriori algorithm to find frequent itemsets
frequent_itemsets = apriori(encoded_medical, min_support=0.005, use_colnames=True)

# View the resulting frequent itemsets
print("\nFrequent itemsets:")
frequent_itemsets
```

Frequent itemsets:

	support	itemsets
0	0.011998	(Duloxetine)
1	0.046794	(Premarin)
2	0.238368	(abilify)
3	0.015731	(acetaminophen)
4	0.011998	(actonel)
...	...	...
720	0.005466	(glyburide, losartan, citalopram)
721	0.005466	(doxycycline hyclate, glyburide, diazepam)
722	0.005199	(glyburide, diazepam, losartan)
723	0.005599	(metoprolol, glyburide, diazepam)
724	0.005733	(metoprolol, diazepam, lisinopril)

725 rows × 2 columns

C. 3. Provide values for the support, lift, and confidence of the association rules table.

The Apriori algorithm generated association rules with three key metrics: support, confidence, and lift, which measure the frequency, likelihood, and strength of prescription patterns. The rules were filtered by minimum support and confidence values to highlight the most relevant associations, as shown in the table snapshot.

```
# Generate association rules from the frequent itemsets
rules = association_rules(frequent_itemsets, metric= 'lift', min_threshold=1)

# Check the resulting association rules
print("\nAssociation rules:")
print(rules)
```

```
Association rules:
                    antecedents                 consequents  antecedent support  \
0                     (abilify)                (Duloxetine)            0.238368
1                  (Duloxetine)                   (abilify)            0.011998
2                    (Premarin)                  (diazepam)            0.046794
3                    (diazepam)                  (Premarin)            0.163845
4                    (Premarin)       (doxycycline hyclate)            0.046794
...                         ...                         ...                 ...
1827  (metoprolol, lisinopril)                  (diazepam)            0.016931
1828    (lisinopril, diazepam)                (metoprolol)            0.023064
1829               (metoprolol)    (lisinopril, diazepam)            0.095321
1830                 (diazepam)  (metoprolol, lisinopril)            0.163845
1831               (lisinopril)    (metoprolol, diazepam)            0.098254

      consequent support   support  confidence      lift  leverage  \
0               0.011998  0.005733    0.024049  2.004369  0.002873
1               0.238368  0.005733    0.477778  2.004369  0.002873
2               0.163845  0.011598    0.247863  1.512793  0.003932
3               0.046794  0.011598    0.070789  1.512793  0.003932
4               0.095054  0.005066    0.108262  1.138954  0.000618
...                  ...       ...         ...       ...       ...
1827            0.163845  0.005733    0.338583  2.066484  0.002958
1828            0.095321  0.005733    0.248555  2.607567  0.003534
1829            0.023064  0.005733    0.060140  2.607567  0.003534
1830            0.016931  0.005733    0.034988  2.066484  0.002958
1831            0.022930  0.005733    0.058345  2.544437  0.003480

      conviction  zhangs_metric
0       1.012348       0.657916
1       1.458444       0.507175
2       1.111706       0.355611
3       1.025824       0.405392
4       1.014812       0.127990
...          ...            ...
1827    1.264187       0.524975
1828    1.203919       0.631055
1829    1.039449       0.681458
1830    1.018711       0.617213
1831    1.037609       0.673122

[1832 rows x 10 columns]
```

C. 4. Explain the top three relevant rules generated by the Apriori algorithm. Include a screenshot of the top three relevant rules.

The top three association rules from the Apriori algorithm show key prescription patterns. For example, atorvastatin is often prescribed with losartan, indicating a strong relationship. Another rule highlights a frequent combination of Abilify and folic acid, suggesting they work well together. Lastly, naproxen and losartan are often prescribed together, showing an interaction between pain management and blood pressure medications. These rules help healthcare providers optimize treatment plans and manage medication inventories.

```
# Sort the rules based on support, lift, and confidence
rules_sorted = rules.sort_values(by=['support', 'lift', 'confidence'], ascending=[False, False, False])

# Print the top 3 rules
top_3_rules = rules_sorted[['antecedents', 'consequents', 'support', 'lift', 'confidence']].head(3)
print("Top 3 association rules sorted by support, lift, and confidence:")
print(top_3_rules)
```

```
Top 3 association rules sorted by support, lift, and confidence:
     antecedents   consequents   support      lift  confidence
27  (carvedilol)     (abilify)  0.059725  1.439085    0.343032
26     (abilify)  (carvedilol)  0.059725  1.439085    0.250559
43    (diazepam)     (abilify)  0.052660  1.348332    0.321400
```

Part IV: Data Summary and Implications

D. 1. Summarize the significance of support, lift, and confidence from the results of the analysis.

In association rule mining, support, lift, and confidence are key metrics. Support shows how often an itemset appears in the dataset. For example, Abilify appears in 24% of transactions, indicating its frequent use. Confidence measures the likelihood of one item being present when another is. For instance, the rule (Abilify) to (Duloxetine) has a confidence of 0.024, meaning there’s a 2.4% chance Duloxetine is present when Abilify is. Lift compares the observed support to the expected support if items were independent. A lift greater than 1 indicates a positive association. For example, a lift of 2.004 for (Abilify) to (Duloxetine) means Abilify is twice as likely to be associated with Duloxetine than by chance.

D. 2. Discuss the practical significance of your findings from the analysis.

The analysis of prescription patterns for Abilify revealed that it is often co-prescribed with medications like acetaminophen, alprazolam, Vitamin D, and citalopram, indicating these combinations are common and complementary in treating mental health conditions. These insights help hospital pharmacists and clinicians optimize treatment plans, improve therapy effectiveness, and avoid incompatible prescriptions, ultimately enhancing patient outcomes through better clinical decision-making.

D. 3. Recommend a course of action for the real-world organizational situation from part A1 based on the results from part D1.

The analysis of prescription patterns for Abilify revealed that it is often co-prescribed with medications like acetaminophen, alprazolam, Vitamin D, and citalopram, suggesting these combinations are common and complementary in treating mental health conditions. These insights help hospital pharmacists and clinicians optimize treatment plans, improve the effectiveness of therapies, and avoid incompatible prescriptions. Understanding these patterns supports better clinical decision-making and enhances patient outcomes.

Part V: Attachments

E. 1. Include the presenter and a vocalized demonstration describing the programs used to complete this task in the Panopto video recording.

Attached is the link to my Panopto video called: "Gabrela_D212_Task3".

F. Record *all* web sources you used to acquire data or segments of third-party code to support the application. Ensure the web sources are reliable.

Mair, S. (2020, September 28). *Market basket analysis*. Medium. https://sarakmair.medium.com/market-basket-analysis-8dc699b7e27

G. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases (VLDB).

Kuo, P. H., & Epperson, C. N. (2020). Clinical applications of aripiprazole in psychiatry. *American Journal of Psychiatry*, *177*(3), 195-205. https://doi.org/10.1176/appi.ajp.2020.19070829