D212 Data Mining II

OFM4 Association Rules and Lifting Analysis (Task 3)

Performance Assessment

Hillary Osei (Student ID #011039266)

Western Governors University, College of Information Technology

Program Mentor: Dan Estes

August 23, 2025

Table of Contents

Part I: Research Question

Section A1. Proposal of Question

The research question addressed is “What other medications are commonly prescribed or purchased alongside Losartan?”

Section A2. Defined Goal

The goal of this analysis is to identify medication patterns involving Losartan. These insights can support safer prescribing, highlight potential drug interactions, and inform treatment plans for patients with hypertension, heart failure, and other conditions treated with Losartan (NHS, n.d.). In addition, the findings can guide inventory planning and help determine which medications should be prioritized in stock.

Part II: Market Basket Justification

Section B1. Explanation of Market Basket

Market basket is a data mining technique used for identifying patterns in purchasing behavior. It goes through every transaction in a database and calculates how likely one item is to be purchased alongside another item. It answers questions like “If a customer buys X, how likely are they to also buy Y?” (Jain, 2025) For example, in a grocery store, it might reveal that a customer who buys bread is also likely to buy butter. Market basket uses an association rule, which is similar to an IF-THEN clause. The antecedent, or the IF, is the item(s) already selected and the consequent, or the THEN, is the item(s) that is likely to be selected next. In the bread–butter example, bread would be the antecedent because it is the known purchase, and butter would be the consequent because it is the item whose likelihood of purchase is being measured in relation to bread.

Relating back to the research question, Losartan would be the antecedent, the known prescribed or purchased medication. Any other medication that is found to frequently occur with Losartan in the dataset would be the consequent. The analysis is expected to identify a ranked list of medications that most frequently occur with Losartan in patient prescriptions or purchases.

Section B2. Transaction Example

Below is an example of a transaction in the dataset:

```
[4]: # Show example of a transaction
     df.iloc[25]
```

```
[4]: Presc01                  paroxetine
     Presc02                  citalopram
     Presc03                     abilify
     Presc04    amphetamine salt combo xr
     Presc05                 fenofibrate
     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: 25, dtype: object
```

Section B3. Market Basket Assumption

Market basket analysis assumes that the data entries in the dataset are consistent, meaning that items are recorded in a standardized and accurate way across all transactions (Deniran, 2023). For example, the same medication should not appear under multiple spellings or formats (e.g., "Losartan," "losartan potassium," "LOSARTAN 50MG") unless those variations are intentional and meaningful for the analysis. Consistency ensures that the algorithm can correctly group identical items and measure their frequency. If the data is inconsistent due to typos, variations in naming, or missing information, results may be inaccurate, as related items could be treated as being separate.

Part III: Data Preparation and Analysis

Section C1. Transforming the Data Set

These are the following steps for transforming the dataset to make it suitable for market basket analysis:

1. Check for duplicated values

```
[6]: df.duplicated()

[6]: 0        False
     1        False
     2         True
     3        False
     4         True
              ...
     14997     True
     14998     True
     14999     True
     15000     True
     15001    False
     Length: 15002, dtype: bool
```

2. Check for null values

```
[7]: df.isnull().sum()

[7]: 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
```

3. Handle missing values in rows, reset index

```
[8]: # Handle missing values, remove rows w/ not null values
df = df[df['Presc01'].notna()]
# Reset index
df.reset_index(drop=True, inplace=True)
```

```
[9]: df.head()
```

[9]:

	Presc01	Presc02	Presc03	Presc04	Presc05	Presc06	Presc07	Presc08	Presc09	Presc10	Presc11	Presc12	Presc13	
0	amlodipine	albuterol aerosol	allopurinol	pantoprazole	lorazepam	omeprazole	mometasone	fluconozole	gabapentin	pravastatin	cialis	losartan	metoprolol succinate XL	sulfam
1	citalopram	benicar	amphetamine salt combo xr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	enalapril	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	paroxetine	allopurinol	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	abilify	atorvastatin	folic acid	naproxen	losartan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

4. Check shape of dataframe

```
[10]: df.shape
```

```
[10]: (7501, 20)
```

5. Extract 7,501 rows from the first 20 columns, remove any NaN or empty values, and store the remaining cleaned entries for each row as a transaction in a list

```
[11]: from mlxtend.preprocessing import TransactionEncoder
transactions = []
for i in range(min(7501, len(df))):  # up to 7501 rows
    temp_small = []
    for j in range(20):  # first 20 columns
        val = df.values[i, j]  # get the cell
        if pd.notna(val):  # skip NaNs
            s = str(val).strip()
            if s != "":  # skip empty strings
                temp_small.append(s)
    transactions.append(temp_small)
```

6. Convert the list of transactions into a one-hot encoded DataFrame using TransactionEncoder, then printing the clean dataframe

```
[12]: # Instantiate TransactionEncoder
      encoder = TransactionEncoder()
      array = encoder.fit(transactions).transform(transactions)

      clean_df = pd.DataFrame(array, columns = encoder.columns_)
```

```
[13]: clean_df
```

[13]:

	Duloxetine	Premarin	Yaz	abilify	acetaminophen	actonel	albuterol HFA	albuterol aerosol	alendronate	allopurinol	...	trazodone HCl	triamcinolone Ace topical	triamterene	trime
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

7. Get data profile for cleaned dataframe

```
[14]: clean_df.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
```

8. Save cleaned dataframe file (note: copy of dataset is attached titled ‘d212-task3.csv’)

```
[15]: # Save cleaned file
      clean_df.to_csv('d212_task3.csv')
```

Section C2. Code Execution

To understand the relationship between items in the dataset, association rules are generated in two main steps. First, the Apriori algorithm is applied to identify all frequent item sets. Next, the association_rules function is used to get the full association rules from these itemsets, calculating key metrics such as support, confidence, and lift.

In the code block below, the Apriori algorithm identifies frequent itemsets, groups of medications that occur in patient transactions more often. The apriori function is applied to the one-hot encoded clean dataframe “clean_df” with min_support being set to 0.02, meaning that combinations present in at least 2% of the transactions are retained.

```
[16]: # Use the Apriori algorithm to generate frequent itemsets
frequent_itemsets = apriori(clean_df, min_support = 0.02, use_colnames = True)
frequent_itemsets
```

[16]:

	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	(diazepam, lisinopril)
99	0.023464	(losartan, diazepam)
100	0.022930	(diazepam, metoprolol)
101	0.020131	(glyburide, doxycycline hyclate)
102	0.028530	(losartan, glyburide)

103 rows × 2 columns

Next, frequent itemsets are transformed into association rules using the association_rules function. Lift is used as the evaluation metric with a minimum threshold of 1, which means that rules that have a positive association between items are kept. The output lists each rule with its antecedent (the "if" part) and the consequent (the "then" part), along with other metrics such as support and confidence.

```
[17]: # Use association_rules with a lift of greater than 1
rules = association_rules(frequent_itemsets, metric = 'lift', min_threshold = 1.0)
rules
```

[17]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	cert
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	1.0	0.006564	1.137144	0.299568	0.082441	0.12
1	(abilify)	(amlodipine)	0.238368	0.071457	0.023597	0.098993	1.385352	1.0	0.006564	1.030562	0.365218	0.082441	0.02
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	1.0	0.008095	1.183991	0.356144	0.086402	0.15
3	(abilify)	(amphetamine salt combo)	0.238368	0.068391	0.024397	0.102349	1.496530	1.0	0.008095	1.037830	0.435627	0.086402	0.03
4	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	1.0	0.008090	1.062815	0.193648	0.138707	0.05
...	...	...	...	...	...	...	...	...	...	...	...	...	
89	(metoprolol)	(diazepam)	0.095321	0.163845	0.022930	0.240559	1.468215	1.0	0.007312	1.101015	0.352502	0.097065	0.09
90	(glyburide)	(doxycycline hyclate)	0.170911	0.095054	0.020131	0.117785	1.239135	1.0	0.003885	1.025766	0.232768	0.081887	0.02
91	(doxycycline hyclate)	(glyburide)	0.095054	0.170911	0.020131	0.211781	1.239135	1.0	0.003885	1.051852	0.213256	0.081887	0.04
92	(losartan)	(glyburide)	0.132116	0.170911	0.028530	0.215943	1.263488	1.0	0.005950	1.057436	0.240286	0.103934	0.05
93	(glyburide)	(losartan)	0.170911	0.132116	0.028530	0.166927	1.263488	1.0	0.005950	1.041786	0.251529	0.103934	0.04

94 rows × 14 columns

Section C3. Association Rules Table

Below is the association rules tables, with the values for support, lift, and confidence:

```
[17]: # Use association_rules with a lift of greater than 1
rules = association_rules(frequent_itemsets, metric = 'lift', min_threshold = 1.0)
rules
```

[17]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	cert
0	(amlodipine)	(abilify)	0.071457	0.238368	0.023597	0.330224	1.385352	1.0	0.006564	1.137144	0.299568	0.082441	0.12
1	(abilify)	(amlodipine)	0.238368	0.071457	0.023597	0.098993	1.385352	1.0	0.006564	1.030562	0.365218	0.082441	0.02
2	(amphetamine salt combo)	(abilify)	0.068391	0.238368	0.024397	0.356725	1.496530	1.0	0.008095	1.183991	0.356144	0.086402	0.15
3	(abilify)	(amphetamine salt combo)	0.238368	0.068391	0.024397	0.102349	1.496530	1.0	0.008095	1.037830	0.435627	0.086402	0.03
4	(amphetamine salt combo xr)	(abilify)	0.179709	0.238368	0.050927	0.283383	1.188845	1.0	0.008090	1.062815	0.193648	0.138707	0.05
...	...	...	...	...	...	...	...	...	...	...	...	...	
89	(metoprolol)	(diazepam)	0.095321	0.163845	0.022930	0.240559	1.468215	1.0	0.007312	1.101015	0.352502	0.097065	0.0
90	(glyburide)	(doxycycline hyclate)	0.170911	0.095054	0.020131	0.117785	1.239135	1.0	0.003885	1.025766	0.232768	0.081887	0.0
91	(doxycycline hyclate)	(glyburide)	0.095054	0.170911	0.020131	0.211781	1.239135	1.0	0.003885	1.051852	0.213256	0.081887	0.04
92	(losartan)	(glyburide)	0.132116	0.170911	0.028530	0.215943	1.263488	1.0	0.005950	1.057436	0.240286	0.103934	0.05
93	(glyburide)	(losartan)	0.170911	0.132116	0.028530	0.166927	1.263488	1.0	0.005950	1.041786	0.251529	0.103934	0.0

94 rows × 14 columns

Section C4. Top Three Rules

To find the top three rules, the association rules are filtered with both a lift greater than 1.9 and a confidence above 0.3, then sorted by lift in descending order. The results show the strongest associations in the dataset based on these filters. The three rules are patients who receive lisinopril are also likely to receive carvedilol, those who take glipizide also often take carvedilol, and those prescribed metformin also receive abilify.

```
[18]: top_3_rules = rules[(rules['lift'] > 1.9) & (rules['confidence'] > 0.3)].sort_values(by=['lift'], ascending= False)
top_3_rules
```

[18]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certai
75	(lisinopril)	(carvedilol)	0.098254	0.174110	0.039195	0.398915	2.291162	1.0	0.022088	1.373997	0.624943	0.168096	0.272
72	(glipizide)	(carvedilol)	0.065858	0.174110	0.022930	0.348178	1.999758	1.0	0.011464	1.267048	0.535186	0.105651	0.210
30	(metformin)	(abilify)	0.050527	0.238368	0.023064	0.456464	1.914955	1.0	0.011020	1.401255	0.503221	0.086760	0.286

Part IV: Data Summary and Implications

Section D1. Significance of Support, Lift, and Confidence Summary

In market basket analysis, support represents how frequently an item, or in this case Losartan, occurs in a dataset in relation to all transactions. Confidence represents out of the transactions that include Losartan, how many also contain consequent medications. Lift represents the ratio between the confidence of Losartan and the support of the specific consequent medication. (Zhang, n.d.).

The analysis shows that Losartan is most associated with medications such as glyburide, diazepam, carvedilol, and amphetamine salt combo XR. The support values for these combinations range from 2.3% to 2.85%, meaning that they occur in this proportion of all transactions. The confidence values range from ~14% to 21%, meaning that when Losartan is prescribed, these medications are also prescribed in the same percentage of cases. The lift values range from 1.07 to 1.26, meaning that there is a positive association and that the items are occurring together more often enough for it to not be a random chance (Codefinity, n.d.).

```
[21]: antecedent_df = rules[rules['antecedents'] == {'losartan'}]
      consequent_df = rules[rules['consequents'] == {'losartan'}]
      losartan_df = pd.concat([antecedent_df, consequent_df])
      losartan_df
```

[21]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	cert
52	(losartan)	(amphetamine salt combo xr)	0.132116	0.179709	0.025463	0.192735	1.072479	1.0	0.001721	1.016135	0.077869	0.088920	0.01
76	(losartan)	(carvedilol)	0.132116	0.174110	0.026530	0.200807	1.153335	1.0	0.003527	1.033405	0.153188	0.094852	0.03
86	(losartan)	(diazepam)	0.132116	0.163845	0.023464	0.177598	1.083943	1.0	0.001817	1.016724	0.089231	0.086106	0.01
92	(losartan)	(glyburide)	0.132116	0.170911	0.028530	0.215943	1.263488	1.0	0.005950	1.057436	0.240286	0.103934	0.0
53	(amphetamine salt combo xr)	(losartan)	0.179709	0.132116	0.025463	0.141691	1.072479	1.0	0.001721	1.011156	0.082387	0.088920	0.0
77	(carvedilol)	(losartan)	0.174110	0.132116	0.026530	0.152374	1.153335	1.0	0.003527	1.023900	0.160977	0.094852	0.02
87	(diazepam)	(losartan)	0.163845	0.132116	0.023464	0.143206	1.083943	1.0	0.001817	1.012944	0.092617	0.086106	0.01
93	(glyburide)	(losartan)	0.170911	0.132116	0.028530	0.166927	1.263488	1.0	0.005950	1.041786	0.251529	0.103934	0.0

Section D2. Practical Significance of Findings

The data analysis shows clear prescription patterns. The lift values above 1 indicate that prescriptions such as glyburide, carvedilol, diazepam, and amphetamine salt combo X are given with Losartan more often than expected by chance. This suggests a common treatment pattern, such as managing hypertension alongside diabetes, in the case of glyburide (Mayo Clinic, n.d.), or heart failure. Knowing these patterns can assist healthcare providers in keeping an eye on possible drug interactions and make informed decisions in prescribing medications. Additionally, it gives the hospital notice on which medications are likely to be needed together, making it easier to plan purchases, manage stock levels, and avoid shortages.

Section D3. Course of Action

Based on these results, the hospital could begin monitoring common drug combinations with Losartan to make sure they are appropriate and safe, especially for patients with multiple conditions. Systems can flag unusual or high-risk combinations so that it can be reviewed before finalizing orders. On the other hand, making sure that there is enough glyburide, carvedilol, or other consequent medications on stock when Losartan use is high should also be taken into account.

Part V: Attachments

Section E. Panopto Video

Link to Panopto here:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=348f5817-42bc-4f72-8510-b3420188becd

Section F. Sources for Third-Party Code

Selvaraj, N. (2023, April 24). How to Perform Market Basket Analysis in Python. 365 Data Science. https://365datascience.com/tutorials/python-tutorials/market-basket-analysis/

Section G. Sources

Codefinity. (n.d.). *Support, Confidence, and Lift Measures*. Codefinity. https://codefinity.com/courses/v2/a7e17f02-2cc9-4b92-abe0-cc8710d7011e/d2fce24c-4b70-427a-815a-afeb7ae6e604/bf978c9d-0090-4671-a662-a7277eb34e7d

Deniran, O. H. (2023, November 27). *Boosting Sales with Data: The Power of Market Basket Analysis in Retail*. Medium. https://medium.com/@chemistry8526/boosting-sales-with-data-the-power-of-market-basket-analysis-in-retail-c79cc10a14df

Jain, S. (2025, July 23). *Market Basket Analysis in Data Mining*. GeeksforGeeks. Retrieved August 14, 2025, from https://www.geeksforgeeks.org/data-science/market-basket-analysis-in-data-mining/

Mayo Clinic. (n.d.). *Glyburide (oral route) - Side effects & dosage*. Mayo Clinic. Retrieved August 14, 2025, from https://www.mayoclinic.org/drugs-supplements/glyburide-oral-route/description/drg-20072094

NHS. (n.d.). *About losartan*. NHS. Retrieved August 14, 2025, from https://www.nhs.uk/medicines/losartan/about-losartan/

Zhang, L. (n.d.). *Understanding Support, Confidence, Lift for Market Basket (Affinity) Analysis*. The Data School. Retrieved August 14, 2025, from https://www.thedataschool.co.uk/a/liu-zhang/understanding-lift-for-market-basket-analysis/