ASSOCIATION RULE

Project Report

ASSOCIATION RULE:

Introduction:

Association Rule is a rule-based machine learning technique used to find patterns (relationships, structures) in the data. In this report, we present the findings of an association rule analysis conducted on a **retail dataset**. The goal of the analysis was to identify frequently associated items and extract meaningful insights that can be used to optimize business strategies and improve customer experience.

STEPS:

Step 1: Import Libraries:

We imported the necessary libraries including numpy, pandas, and mlxtend. These libraries provide the required functions and tools for data manipulation, analysis, and association rule mining.

```
#import Libraries
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

Step 2: Data Understanding:

Loaded the dataset using pandas and displayed the first and last few rows to understand the data structure.

Printing First 5 rows :

	0	1	2	3	4	5	6
0	Bread	Wine	Eggs	Meat	Cheese	Pencil	Diaper
1	Bread	Cheese	Meat	Diaper	Wine	Milk	Pencil
2	Cheese	Meat	Eggs	Milk	Wine	NaN	NaN
3	Cheese	Meat	Eggs	Milk	Wine	NaN	NaN
4	Meat	Pencil	Wine	NaN	NaN	NaN	NaN

Printing Last 5 rows:

	0	1	2	3	4	5	6
310	Bread	Eggs	Cheese	NaN	NaN	NaN	NaN
311	Meat	Milk	Pencil	NaN	NaN	NaN	NaN
312	Bread	Cheese	Eggs	Meat	Pencil	Diaper	Wine
313	Meat	Cheese	NaN	NaN	NaN	NaN	NaN
314	Eggs	Wine	Bagel	Bread	Meat	NaN	NaN

To understand the shape of the dataset, We used the shape attribute of the DataFrame, which provides the number of rows and columns in the dataset.

Checking the shape of Dataset

```
(315, 7)
```

There are 315 Rows and 7 Columns in Dataset.

Using Describe Function :

	count	unique	top	freq
0	315	9	Bread	74
1	285	9	Meat	47
2	245	9	Eggs	52
3	187	9	Milk	45
4	133	9	Wine	36
5	71	9	Pencil	13
6	41	9	Bread	11

Step 3: Data Preprocessing:

Firstly, We extracted the unique items from the dataset using the unique() function. Then we Stored the unique items in the **items** variable.

Next, We performed one-hot encoding on the dataset to convert it into a suitable format for association rule analysis. For each transaction in the dataset, Then We created a dictionary where the keys were the unique items, We set the values to 1 if the item was present in the transaction and 0 if the item was absent.

Finally. we stored the encoded values in a new DataFrame named ohe_df, where each column represents an item and each row represents a transaction.

	Milk	Bagel	Meat	Bread	Cheese	Wine	Diaper	Pencil	Eggs
0	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	0
2	1	0	1	0	1	1	0	0	1
3	1	0	1	0	1	1	0	0	1
4	0	0	1	0	0	1	0	1	0
...	...	...	...	...	...	...	...	...	...
310	0	0	0	1	1	0	0	0	1
311	1	0	1	0	0	0	0	1	0
312	0	0	1	1	1	1	1	1	1
313	0	0	1	0	1	0	0	0	0
314	0	1	1	1	0	1	0	0	1

315 rows × 9 columns

Step 4: Association Rules:

We applied the Apriori algorithm to find frequent itemsets in the encoded DataFrame, ohe_df. This was done using the apriori() function from the mlxtend.frequent_patterns module. We set the minimum support threshold to 0.2, indicating that an itemset should appear in at least 20% of the transactions to be considered frequent.

The frequent itemsets are stored in the freq_items DataFrame, which contains the itemsets and their corresponding support values.

	support	itemsets
0	0.501587	(Milk)
1	0.425397	(Bagel)
2	0.476190	(Meat)
3	0.504762	(Bread)
4	0.501587	(Cheese)

We generated association rules from the frequent itemsets, We used the association_rules() function from the mlxtend.frequent_patterns module. We set the metric parameter to "confidence" and the minimum threshold to 0.6, indicating that we are interested in rules with a confidence of at least 60%.

- The resulting association rules were stored in the rules DataFrame.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(Milk)	(Cheese)	0.501587	0.501587	0.304762	0.607595	1.211344	0.053172	1.270148	0.350053
1	(Cheese)	(Milk)	0.501587	0.501587	0.304762	0.607595	1.211344	0.053172	1.270148	0.350053
2	(Bagel)	(Bread)	0.425397	0.504762	0.279365	0.656716	1.301042	0.064641	1.442650	0.402687
3	(Meat)	(Cheese)	0.476190	0.501587	0.323810	0.680000	1.355696	0.084958	1.557540	0.500891
4	(Cheese)	(Meat)	0.501587	0.476190	0.323810	0.645570	1.355696	0.084958	1.477891	0.526414
5	(Eggs)	(Meat)	0.438095	0.476190	0.266667	0.608696	1.278261	0.058050	1.338624	0.387409
6	(Wine)	(Cheese)	0.438095	0.501587	0.269841	0.615942	1.227986	0.050098	1.297754	0.330409
7	(Eggs)	(Cheese)	0.438095	0.501587	0.298413	0.681159	1.358008	0.078670	1.563203	0.469167
8	(Milk, Meat)	(Cheese)	0.244444	0.501587	0.203175	0.831169	1.657077	0.080564	2.952137	0.524816
9	(Milk, Cheese)	(Meat)	0.304762	0.476190	0.203175	0.666667	1.400000	0.058050	1.571429	0.410959
10	(Meat, Cheese)	(Milk)	0.323810	0.501587	0.203175	0.627451	1.250931	0.040756	1.337845	0.296655
11	(Meat, Cheese)	(Eggs)	0.323810	0.438095	0.215873	0.666667	1.521739	0.074014	1.685714	0.507042
12	(Eggs, Cheese)	(Meat)	0.298413	0.476190	0.215873	0.723404	1.519149	0.073772	1.893773	0.487091
13	(Meat, Eggs)	(Cheese)	0.266667	0.501587	0.215873	0.809524	1.613924	0.082116	2.616667	0.518717

Step 5: Analysis and Observations:

The percentage of rules with a confidence greater than or equal to 0.6 was calculated.

```
Percentage of rules with confidence >= 0.6: 7512.12%
```

Additionally, we showcased the first few rows of frequent itemsets and association rules using the head() function.

```
associated_items.head()
```

```
/usr/local/lib/python3.10/dist-packages/i
  and should_run_async(code)
```

	Item1	Item2	Support	Confidence
0	Milk	Cheese	0.304762	0.607595
1	Cheese	Milk	0.304762	0.607595
2	Bagel	Bread	0.279365	0.656716
3	Meat	Cheese	0.323810	0.680000
4	Cheese	Meat	0.323810	0.645570

Step 6: Conclusion:

- We Noticed interesting connections, like "Milk" and "Cheese" often being bought together with a confidence of 60.76%. This means if someone buys "Milk," there's a good chance they'll also buy "Cheese."

- We also found a strong link between "Meat" and "Cheese" with a confidence of 68%, suggesting a likely joint purchase.

- Almost 28% of the time, if someone grabs Bagels, they're also taking Bread. around 65.67%.