THROUGH ASSOCIATION RULE MINING

MARKET BASKET ANALYSIS

BY: SHRADDHA SAHA

CONTENTS

OBJECTIVE

- A dataset from an E-commerce startup is here to analyze and obtain an action plan to increase their sales.

DATA COLLECTION

- An MS Excel file named "OnlineClothing" with two sheets is provided
- "Data" for order history and

OrderID	CustomerID	PurchaseDate (yy-mm-dd)	ProductID	Product	Quantity	UnitPrice
1001	101	6/27/2023	P01	Summer Cap	2	100
1001	101	6/27/2023	P02	Sunglasses	1	50
1002	102	7/28/2023	P07	Kurta	1	150
1003	103	7/29/2023	P03	Half Sleeve T-shirt	1	200
1003	103	7/29/2023	P04	Capri	2	350
1004	104	8/31/2023	P05	Saree	1	400
1004	104	8/31/2023	P06	Earrings	1	30

- "ProductLookup" for Product description.

Product	ProductID	Season	Price
Summer Cap	P01	Summer	100
Sunglasses	P02	Summer	50
Half Sleeve T-shirt	P03	Summer	200
Capri	P04	Summer	350

KEY CONCEPTS

- **Association Rule Mining** is a powerful data mining technique used to uncover hidden relationships between items within large datasets.
- It is particularly valuable in analyzing customer behavior, such as determining which products are frequently purchased together in a grocery store. Beyond retail, it finds applications in various fields, including healthcare (e.g., finding correlations between symptoms and diagnoses) and marketing (e.g., identifying cross-selling opportunities).
- Also known as Market Basket Analysis or Affinity Analysis, this method helps businesses make data-driven decisions by revealing patterns that may not be immediately apparent.

- **Itemset:** A collection of items. For example, {bread, milk, butter} is an itemset.
- **Transaction:** A record of items purchased together. For instance, a customer buying bread, milk, and eggs would be a transaction.
- **Association Rule:** It is a statement that describes the relationship between items within a dataset.
- It typically follows the form:
  If {antecedent items} **then** {consequent items}
- A rule like {bread, milk} -> {butter} suggests that customers who buy bread and milk are likely to also buy butter.

KEY CONCEPTS (CONT.)

- **Antecedents**: This is the antecedent item sets of the association rule. In association rule mining, an antecedent is the item or items that appear on the left-hand side of the rule.
- **Consequents**: This is the consequent item sets of the association rule. The consequent is the item or items that appear on the right-hand side of the rule.
- **Support:** It measures the proportion of transactions that contain a particular itemset (combination of items). It helps identify how frequently an itemset occurs in the dataset.

Support(Antecedent, Consequent) = P(Antecedent ∩Consequent)

- **Confidence**: It measures the likelihood that a customer who buys the antecedent items will also buy the consequent items. High confidence indicates a strong association between the antecedent and consequent.

Confidence(Antecedent ->Consequent) = P(Consequent | Antecedent)

- **Lift**: It measures how much more likely the consequent items are to be bought when the antecedent items are purchased, compared to the likelihood of buying the consequent items without considering the antecedents. A lift value greater than 1 indicates a positive association.

Lift(Antecedent -> Consequent) = P(Consequent | Antecedent)/ P(Consequent)

TOOLS USED

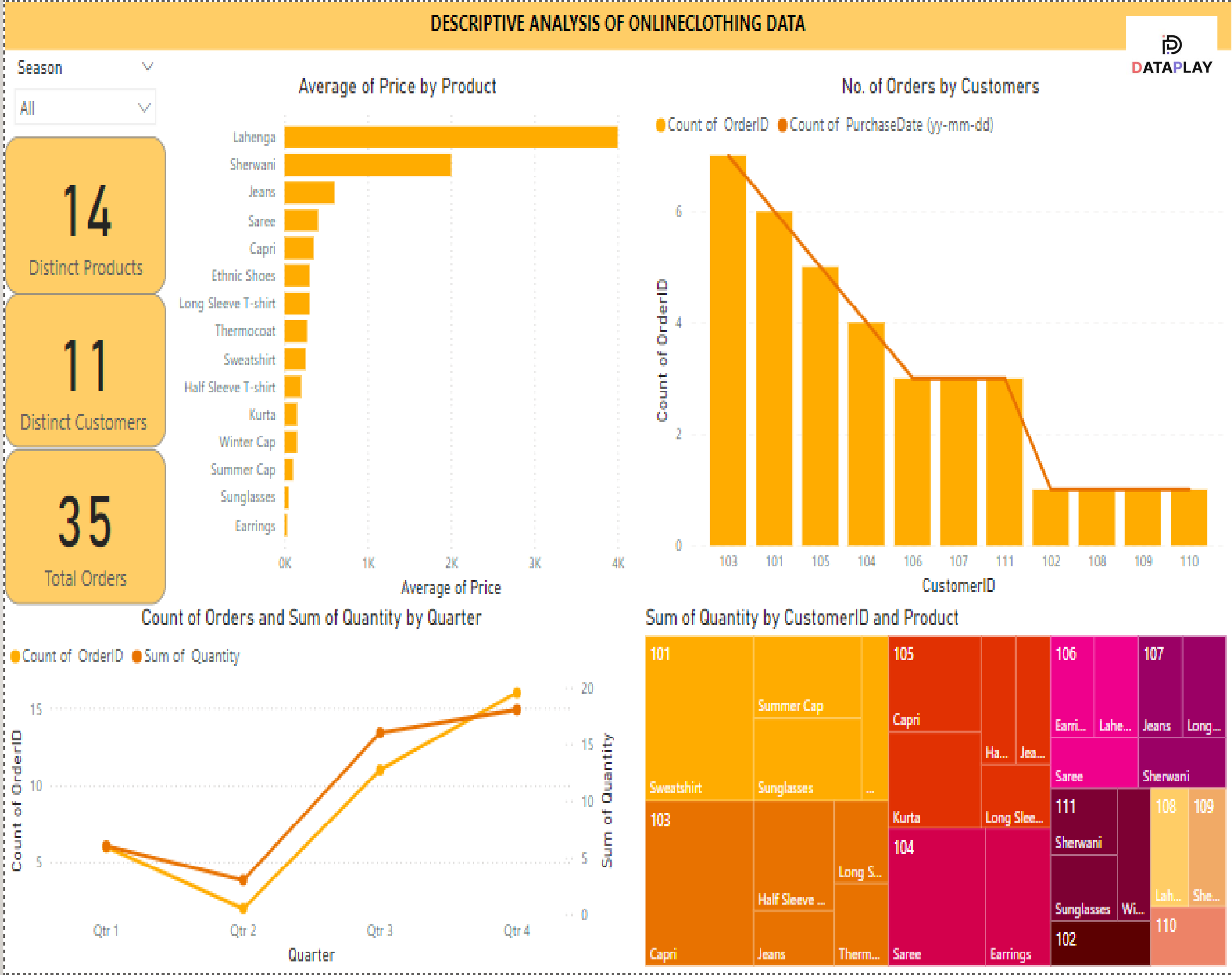
- For basic visualizations, we used **Microsoft Power BI**; used *DAX* for newly calculated columns, and then created *Dashboards*.
- For Association Rule Mining, we worked on Jupiter Notebook using **Python** Programming Language.

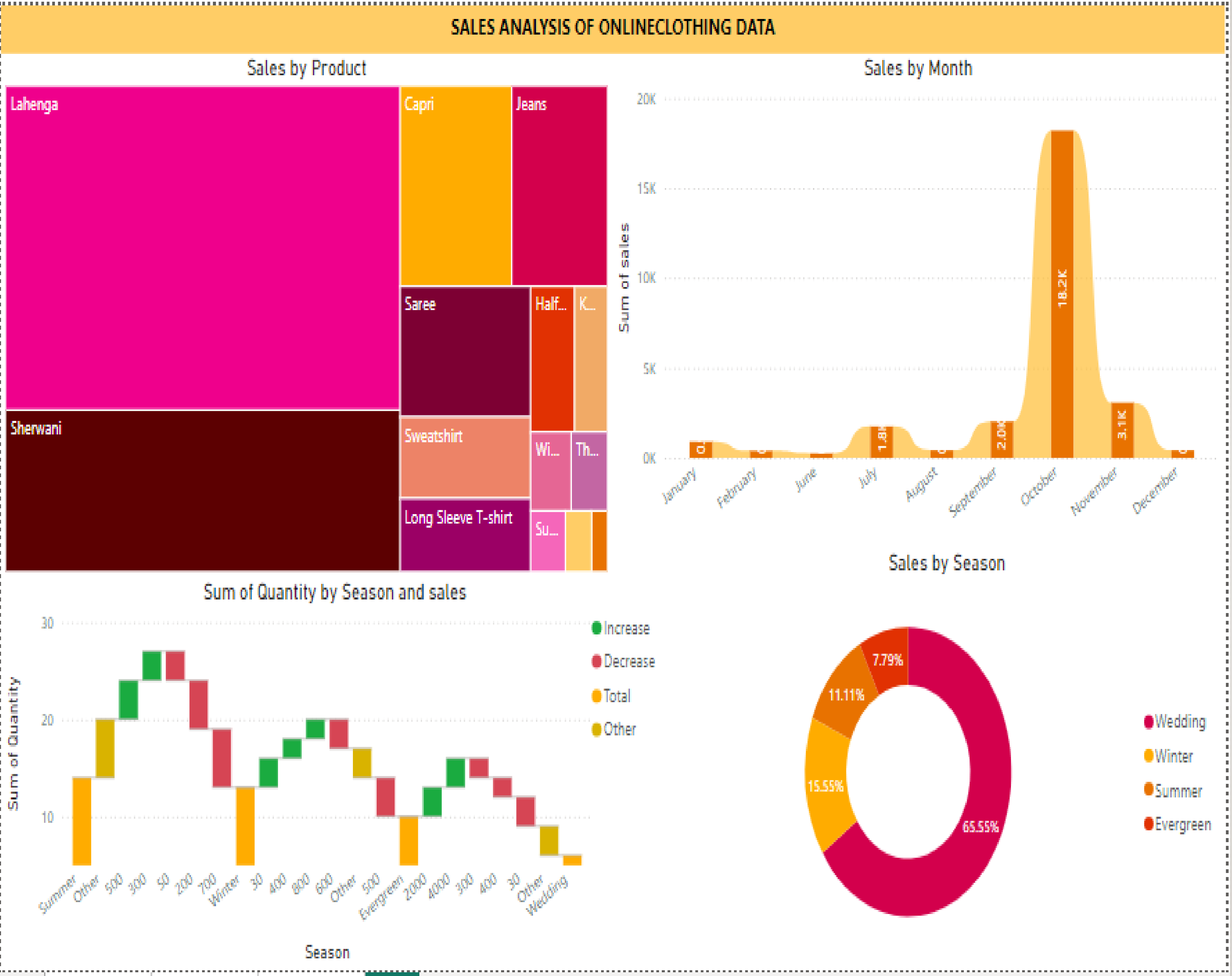
WORKFLOW

- Let us see the next 2 slides for understanding basic descriptive analysis in **Power BI** dashboards.

DATAPLAY

- Average price of Lehenga is the highest followed by that of Sherwani.
- The comparison between no. of orders and quantity of products ordered is shown through a line chart.
- The customer with ID 103 stands the highest in terms of maximum no. of Orders placed.
- But the Tree map shows that customer with ID 101 takes the same highest place with 103 for ordering maximum quantity of products.





- Now, we have calculated "Sales" using DAX from the "Price" and "Quantity" column.
- Now, we see that maximum sales was from Lehenga
- The Donut chart depicts the percentage share for Wedding season to be the highest with 65.55% of total sales.
- October becomes the first ranker in terms of total sales seen in Ribbon chart.
- In the waterfall chart, the increase and decrease of sales with no. of quantity and different seasons is shown.

WORKFLOW (CONT.)

- Next, concentrate on the **Python** programming for **Association Rule Mining**.

- Imported the ***mlxtend*** library here for working with association_rules and apriori algorithm.
- '**Apriori**' acknowledges the prior knowledge of frequent itemsets that the algorithm uses in computation.
- Read the data using pandas.
- Before starting the association rule mining, it is necessary to check if there is any missing value and take relevant steps.
- Also, search for duplicate rows, here we have no duplicate row.

In [1]:
```
#Import necessary libraries:

import numpy as np
import pandas as pd
import mlxtend
```

In [2]:
```
from mlxtend.frequent_patterns import association_rules, apriori
```

In [3]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [4]:
```
# Read the data:
market=pd.read_excel("OnlineClothing.xlsx",sheet_name="Data")
```

In [6]:
```
#checking for missing values:

market.isnull().sum()
```

In [7]:
```
#checking for duplicate rows:
market[market.duplicated()]
```

Out[7]:

OrderID	CustomerID	PurchaseDate (yy-mm-dd)	ProductID	Product	Quantity	UnitPrice

- We extracted two relevant columns only i.e, "OrderID" and "Product".
- Encoding is done using *.size().unstack()* to create a matrix with each product as a column and transactions as rows filling non-purchases with zero.
- Next, the encoded data is converted to binary data by *lambda* function.
- Here all values greater than zero are converted to 1 (indicating a purchase) and kept zero otherwise.

In [9]:
```
# transforming the data for applying ariori alogorithm:

transaction_data = market[[" OrderID ","Product"]].copy()
```

In [12]:
```
# One-hot encoding the transaction data
encoded_transaction_data = transaction_data.groupby([' OrderID ', 'Product']).size().unstack(fill_value=0)
encoded_transaction_data = encoded_transaction_data.applymap(lambda x: 1 if x > 0 else 0)
```

In [13]:
```
encoded_transaction_data
```

Out[13]:

Product	Capri	Earrings	Half Sleeve T-shirt	Jeans	Kurta	Lahenga	Long Sleeve T-shirt	Saree	Sherwani	Summer Cap	Sunglasses	Sweatshirt	Thermocoat	Winter Cap
OrderID														
1001	0	0	0	0	0	0	0	0	0	1	1	0	0	0
1002	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1003	1	0	1	0	0	0	0	0	0	0	0	0	0	0
1004	0	1	0	0	0	0	0	1	0	0	0	0	0	0
1005	1	0	1	0	0	0	0	0	0	0	0	0	0	0
1006	0	0	0	0	1	0	0	0	0	0	0	0	0	0

DATAPLAY

- Now, our data is ready for applying the Apriori algorithm.
- To check the frequencies of all products, we took *support = 0.01* at first.
- Finally, we have chosen *min_support=0.1* as increasing it restricts results to more common item sets, while decreasing it includes rarer item sets.
- As *confidence* can be misleading if the consequent item is very popular, so chosen *lift*.
- Now *min_threshold* for *lift* is taken as a lift value greater than 1 indicates that the antecedent and consequent are positively correlated.

In [14]:
```
#checking frequency of all products:
support = 0.01
frequent_items = apriori(encoded_transaction_data,min_support=support, use_colnames=True)
frequent_items.sort_values('support')
```

Out[14]:

	support	itemsets
9	0.043478	(Summer Cap)
12	0.043478	(Thermocoat)
17	0.043478	(Sunglasses, Summer Cap)
13	0.086957	(Winter Cap)
11	0.086957	(Sweatshirt)
18	0.086957	(Sunglasses, Winter Cap)

In [20]:
```
# Applying the Apriori algorithm
frequent_itemsets = apriori(encoded_transaction_data, min_support=0.1, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']]
```

Out[20]:

	antecedents	consequents	support	confidence	lift
0	(Capri)	(Half Sleeve T-shirt)	0.130435	1.0	7.666667
1	(Half Sleeve T-shirt)	(Capri)	0.130435	1.0	7.666667
2	(Earrings)	(Saree)	0.130435	1.0	7.666667
3	(Saree)	(Earrings)	0.130435	1.0	7.666667
4	(Long Sleeve T-shirt)	(Jeans)	0.130435	1.0	7.666667
5	(Jeans)	(Long Sleeve T-shirt)	0.130435	1.0	7.666667

** Here, one thing to notice that the pairs are coming twice by swapping the (antecedents, consequents) places only, support, confidence and lift values remaining same; But this metrics' values would differ if the no. of orders would not be same for swapped pairs .

INSIGHTS

- Lift is a measure of how much more likely the antecedent and consequent items are to appear together compared to what would be expected if they were independent of each other.
- As we have filtered out only the product sets with lift value >1 , all the obtained product sets are significant.
- Therefore, **(Capri, Half Sleeve T-shirt), (Earrings, Saree), (Long Sleeve T-shirt, Jeans)** these 3 pairs should be strategically placed, encouraging customers to buy those to increase sales and for customer satisfaction.

* But above all, the dataset should be larger for better analysis.

Check out this link below for the full assignment work :

https://github.com/ShraddhaSaha/Market-Basket-Analysis

Thank You!

shraddhaasaha@gmail.com