MARKET BASKET INSIGHT

PHASE 4: DEVELOPMENT PART 2

INTRODUCTION:

Market basket insight, often referred to as "Market Basket Analysis" or "Affinity Analysis," is a data-driven technique used by businesses, particularly in the retail and e-commerce sectors, to gain a deeper understanding of customer purchasing behavior and identify patterns in the products or items that are frequently bought together. This analysis is instrumental in making data-driven decisions for optimizing various aspects of a business, such as inventory management, product recommendations, marketing strategies, and store layout design.

Market basket insight is achieved through various techniques, including association rule mining, collaborative filtering, and machine learning models. These techniques uncover patterns such as item co-occurrences, purchase sequences, and customer preferences. By applying these insights, businesses can improve customer satisfaction, increase revenue, and make more informed decisions regarding their product offerings and marketing strategies.

In an era where data plays a crucial role in business success, market basket insight is a valuable tool that empowers organizations to better understand their customers and adapt to their evolving needs and preferences. It has become an integral part of data-driven decision-making in the retail and e-commerce industries, ultimately leading to more efficient and customer-focused operations.

FEATURE ENGINEERING:

critical step in the process of generating market basket insights from transaction data. Market basket analysis, often associated with retail or e-commerce, involves understanding the relationships between products or items that customers purchase together. The goal is to identify patterns and associations to inform marketing, inventory management, and pricing strategies. Feature engineering can help extract valuable information from your data. Here are some feature engineering techniques and ideas for market basket insight:

1. **Transaction-Level Features**:

   - **Transaction Count**: Count the number of transactions made by each customer. This could indicate their shopping frequency.

   - **Transaction Value**: Calculate the total value of each transaction. This can help identify high-value customers.

2. **Item-Level Features**:

   - **Item Frequency**: Count how often each item is purchased. This can help you identify popular items.

   - **Item Association**: Calculate item co-occurrence or lift, which indicates how often items are bought together. High lift values suggest strong associations.

   - **Item Category**: Group items into categories or departments. This can help you understand which types of products are frequently purchased together.

3. **Customer-Level Features**:

   - **Customer Segmentation**: Use clustering techniques to group customers with similar buying behavior. This can help in tailoring marketing strategies.

   - **Average Transaction Value**: Calculate the average value of a customer's transactions. This can help identify high-value customers.

   - **Recency, Frequency, Monetary (RFM)**: Create RFM features to segment customers based on recency of their last purchase, frequency of purchases, and monetary value.

4. **Time-Based Features**:

   - **Time of Day/Week/Month**: Analyze when customers tend to shop. You may find that certain items are more popular during specific times.

   - **Seasonality**: Explore seasonal patterns to optimize inventory and promotions.

5. **Sequential Patterns**:

   - **Sequential Item Lists**: Create sequences of items purchased in each transaction. Analyze sequential patterns, such as "first A, then B."

   - **Markov Models**: Build Markov models to understand the probability of transitioning from one item to another in a sequence.

6. **Market Basket Metrics**:

   - **Support**: Calculate the support of an itemset, indicating the percentage of transactions that contain the itemset.

   - **Confidence**: Measure how often item B is purchased when item A is purchased. High confidence indicates a strong association.

- **Lift**: Compute lift to understand how much more likely item B is purchased when item A is in the basket. High lift values are indicative of strong associations.

7. **Text Analysis**:

- If you have item descriptions or reviews, perform text analysis to extract keywords or sentiments associated with products. This can provide additional insights into customer preferences.

8. **Demographic Data**:

- If available, incorporate customer demographics such as age, gender, location, and income to understand how different customer segments behave.

9. **External Data**:

- Utilize external data sources like weather, holidays, or economic indicators to identify factors that influence shopping behavior.

10. **Dimensionality Reduction**:

- Use techniques like PCA or t-SNE to reduce the dimensionality of the data when dealing with a large number of items or features.

MODEL TRAINING:

Training a machine learning model using market basket insights involves applying algorithms to transaction data to make predictions, recommendations, or gain further insights. Here are some common approaches for model training in market basket analysis:

1. **Association Rules**:

   - **Apriori Algorithm**: Apriori is a classic algorithm for finding frequent itemsets and generating association rules. These rules can be used to make product recommendations or optimize in-store product placements.

   - **FP-Growth**: Another frequent itemset mining algorithm that can efficiently discover patterns in transaction data.

   Example: If a customer buys milk and bread, they are likely to purchase butter.

2. **Collaborative Filtering**:

   - Collaborative filtering methods use the purchase history of multiple users to make recommendations. These methods can be used to recommend products to customers based on what similar customers have bought.

   Example: If Customer A and Customer B have similar purchase histories and Customer A buys a product, it is recommended to Customer B.

3. **Matrix Factorization**:

   - Matrix factorization techniques such as Singular Value Decomposition (SVD) or Alternating Least Squares (ALS) can be used to factorize the purchase matrix to make recommendations.

Example: Decomposing the purchase history matrix can help predict missing values, i.e., what products a customer might purchase.

4. **Market Basket Analysis Algorithms**:

- Specific market basket analysis algorithms like Eclat, SPADE, or CARMA can be used to identify sequential patterns in the transaction data, which can be helpful for product placement or recommendation.

Example: If customers often buy chips before buying beer, these items can be placed near each other in the store.

5. **Machine Learning Models**:

- You can use machine learning models, such as decision trees, random forests, or neural networks, to predict customer behavior or recommend products based on customer profiles and transaction history.

Example: Building a model that takes into account customer demographics, past purchase history, and external data to predict which products a customer is likely to buy.

6. **Deep Learning Models**:

- Deep learning models, such as recurrent neural networks (RNNs) or sequence-to-sequence models, can capture sequential patterns in customer purchases.

Example: Using an RNN to predict the next product in a customer's shopping sequence.

7. **Hybrid Models**:

- Combining various techniques like collaborative filtering, content-based filtering, and matrix factorization can often result in more accurate and personalized recommendations.

8. **Evaluation Metrics**:

- Regardless of the model you choose, you'll need to establish evaluation metrics to measure its performance, such as accuracy, precision, recall, F1-score, or mean average precision (MAP).

9. **Deployment**:

- Once you've trained your model, you can deploy it in your retail or e-commerce system to provide real-time recommendations or insights.

10. **A/B Testing**:

- Implement A/B testing to assess the effectiveness of your recommendations and continuously optimize your model.

EVALUATION:

1. Importing Necessary Dependencies

```
import numpy as npimport pandas as pd

import plotly.express as pximport plotly.graph_objects as goimport plotly.figure_factory as fffrom plotly.offline import download_plotlyjs, init_notebook_mode, iplot

from mlxtend.preprocessing import TransactionEncoderfrom mlxtend.frequent_patterns import apriori, association_rulesimport networkx as nx
```

2. Loading and Reading Dataset

```
bakeryDF=pd.read_csv("../input/bakery/Bakery.csv")bakeryDF.head()
```

	TransactionNo	Items	DateTime	Daypart	DayType
0	1	Bread	2016-10-30 09:58:11	Morning	Weekend
1	2	Scandinavian	2016-10-30 10:05:34	Morning	Weekend
2	2	Scandinavian	2016-10-30 10:05:34	Morning	Weekend
3	3	Hot chocolate	2016-10-30 10:07:57	Morning	Weekend
4	3	Jam	2016-10-30 10:07:57	Morning	Weekend

```
print("Database dimension :", bakeryDF.shape)print("Database size      :", bakeryDF.size)
```

Database dimension : (20507, 5)

Database size   : 102535

```
bakeryDF.info()
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 20507 entries, 0 to 20506

Data columns (total 5 columns):

 #   Column   Non-Null Count Dtype

--- ------   -------------- -----

 0   TransactionNo 20507 non-null int64

 1   Items   20507 non-null object

 2   DateTime   20507 non-null object

 3   Daypart   20507 non-null object

 4   DayType   20507 non-null object

dtypes: int64(1), object(4)

memory usage: 801.2+ KB

```
bakeryDF['TransactionNo'].nunique()
```

```
bakeryDF.describe(include=object)
```

	Items	DateTime	Daypart	DayType

	Items	DateTime	Daypart	DayType
count	20507	20507	20507	20507
unique	94	9465	4	2
top	Coffee	2017-11-02 14:08:27	Afternoon	Weekday
freq	5471	11	11569	12807

Data Summary:

Overview

The dataset provides transaction details of all items purchased between 2016 and 2017 from the bakery online. The dataset has 20507 entries over 9000 transactions, and 4 columns.

- Number of variables: 1
- Numeric variables: 1
- Categorical variables: 4
- Number of observations: 20507
- Total number of transactions: 9465
- Missing cells : 0

Variables

- TransactionNo : 9465 distinct values
- Items has a high cardinality: 94 distinct values
- DateTime has a high cardinality: 9182 distinct values
- Daypart has 4 distinct values
- DayType has 2 distinct values

3. Data Exploration and Visualization

3.1 Let's look into the frequent items and the best sellers

```
itemFrequency = bakeryDF['Items'].value_counts().sort_values(ascending=False)itemFrequency.head(10)
```

Coffee 5471

Bread 3325

Tea 1435

Cake 1025

Pastry 856

Sandwich 771

Medialuna 616

Hot chocolate 590

Cookies 540

```
fig = px.bar(itemFrequency.head(20), title='20 Most Frequent Items', color=itemFrequency.head(20), color_continuous_scale=px.colors.sequential.Mint)fig.update_layout(margin=dict(t=50, b=0, l=0, r=0), titlefont=dict(size=20), xaxis_tickangle=-45, plot_bgcolor='white', coloraxis_showscale=False)fig.update_yaxes(showticklabels=False, title=' ')fig.update_xaxes(title=' ')fig.update_traces(texttemplate='%{y}', textposition='outside', hovertemplate = '<b>%{x}</b><br>No. of Transactions: %{y}')fig.show()
```

54713325143510258567716165905403793743703693693423273182771931 85CoffeeBreadTeaCakePastrySandwichMedialunaHot chocolateCookiesBrownieFarm HouseMuffinAlfajoresJuiceSoupSconeToastScandinavianTrufflesCoke20 Most Frequent Items

Coffee is the best-selling product by far, followed by bread and tea.

3.2 Let's look into the peak hours of sales

In [9]:

```
peakHours = bakeryDF.groupby('Daypart')['Items'].count().sort_values(ascending=False)peakHours
```

Out[9]:

Daypart

Afternoon 11569

Morning 8404

Evening 520

Night 14

Name: Items, dtype: int64

In [10]:

```
fig = go.Figure(data=[go.Pie(labels=['Afternoon','Morning','Evening','Night'],
```

Market basket insight, often referred to as "Market Basket Analysis" or "Affinity Analysis," is a data-driven technique used by businesses, particularly in the retail and e-commerce sectors, to gain a deeper understanding of customer purchasing behavior and identify patterns in the products or items that are frequently bought together. This analysis is instrumental in making data-driven decisions for optimizing various aspects of a business, such as inventory management, product recommendations, marketing strategies, and store layout design.

Market basket insight is rooted in the observation that customers tend to purchase certain products in combination, and these buying patterns can reveal valuable insights that businesses can leverage to enhance their operations. The primary goal of market basket analysis is to uncover associations, correlations, and trends within transaction data to answer questions like:

1. **What items are commonly purchased together?** This helps retailers understand complementary products that can be bundled or co-located in stores.

2. **How can we improve product recommendations?** By identifying purchasing patterns, businesses can recommend additional products to customers based on their current selections.

3. **What is the impact of product placements on sales?** Analyzing which products are often bought together can inform store layout and shelf arrangement decisions.

4. **Which customers exhibit similar buying behaviors?** Segmenting customers based on their purchase history can lead to more targeted marketing and personalized offers.

Market basket insight is achieved through various techniques, including association rule mining, collaborative filtering, and machine learning models. These techniques uncover patterns such as item co-occurrences, purchase sequences, and customer preferences. By applying these insights, businesses c an improve customer satisfaction, increase revenue, and make more informed decisions regarding their product offerings and marketing strategies.

In an era where data plays a crucial role in business success, market basket insight is a valuable tool that empowers organizations to better understand their customers and adapt to their evolving needs and preferences. It has become an integral part of data-driven decision-making in the retail and e-commerce industries, ultimately leading to more efficient and customer-focused operations.

Afternoon56.4%Morning41%Evening2.54%Night0.0683%Peak Selling Hours

The bakery seems to be making most of its sales in the afternoon everyday with over 56% of the sales. Sales fall sharply after that. However the bakery makes a decent amount of sales in the morning as well.

3.3 Further let's look into the monthly and weekly sales

Need to extract months and days from the dataset for further analysis.

In [11]:

```
dateTime=pd.to_datetime(bakeryDF['DateTime'])bakeryDF['Day']=dateTime.dt.day_name()bakeryDF['Month']=dateTime.dt.month_name()bakeryDF['Year']=dateTime.dt.yearbakeryDF.head(5)
```

Out[11]:

	TransactionNo	Items	DateTime	Daypart	DayType	Day	Month	Year
0	1	Bread	2016-10-30 09:58:11	Morning	Weekend	Sunday	October	2016
1	2	Scandinavian	2016-10-30 10:05:34	Morning	Weekend	Sunday	October	2016
2	2	Scandinavian	2016-10-30 10:05:34	Morning	Weekend	Sunday	October	2016
3	3	Hot chocolate	2016-10-30 10:07:57	Morning	Weekend	Sunday	October	2016
4	3	Jam	2016-10-30 10:07:57	Morning	Weekend	Sunday	October	2016

In [12]:

```
mpd = bakeryDF.groupby('Day')['Items'].count().sort_values(ascending=False)mpd
```

Out[12]:

Day

Saturday 3554

```
Friday       3266
Sunday       3118
Monday       3035
Tuesday      2645
Thursday     2601
Wednesday    2288
Name: Items, dtype: int64
```

Market basket insight, often referred to as "Market Basket Analysis" or "Affinity Analysis," is a data-driven technique used by businesses, particularly in the retail and e-commerce sectors, to gain a deeper understanding of customer purchasing behavior and identify patterns in the products or items that are frequently bought together. This analysis is instrumental in making data-driven decisions for optimizing various aspects of a business, such as inventory management, product recommendations, marketing strategies, and store layout design.

Market basket insight is rooted in the observation that customers tend to purchase certain products in combination, and these buying patterns can reveal valuable insights that businesses can leverage to enhance their operations. The primary goal of market basket analysis is to uncover associations, correlations, and trends within transaction data to answer questions like:

1. **What items are commonly purchased together?** This helps retailers understand complementary products that can be bundled or co-located in stores.

2. **How can we improve product recommendations?** By identifying purchasing patterns, businesses can recommend additional products to customers based on their current selections.

3. **What is the impact of product placements on sales?** Analyzing which products are often bought together can inform store layout and shelf arrangement decisions.

4. **Which customers exhibit similar buying behaviors?** Segmenting customers based on their purchase history can lead to more targeted marketing and personalized offers.

Market basket insight is achieved through various techniques, including association rule mining, collaborative filtering, and machine learning models. These techniques uncover patterns such as item co-occurrences, purchase sequences, and customer preferences. By applying these insights, businesses can improve customer satisfaction, increase revenue, and make more informed decisions regarding their product offerings and marketing strategies.

In an era where data plays a crucial role in business success, market basket insight is a valuable tool that empowers organizations to better understand their customers and adapt to their evolving needs and preferences. It has become an integral part of data-driven decision-making in the retail and e-commerce industries, ultimately leading to more efficient and customer-focused operations.

3554326631183035264526012288SaturdayFridaySundayMondayTuesdayThursdayWednesdayMost Productive Day

For obvious reasons, the sales are high as expected during the weekends. However the sales seem to be quite uniform rest of the days.

```
mpm = bakeryDF.groupby('Month')['Items'].count().sort_values(ascending=False)mpm
```

Month

March 3220

November 3076

January 3027

February 2748

December 2647

April 1048

October 1041

May 924

July   741

June   739

August   700

September   596

Name: Items, dtype: int64

```
fig = px.bar(mpm, title='Most Productive Month', color=mpm, color_continuous_scale=px.colors.sequential.Mint)fig.update_layout(margin=dict(t=50, b=0, l=0, r=0), titlefont=dict(size=20), xaxis_tickangle=0, plot_bgcolor='white', coloraxis_showscale=False)fig.update_yaxes(showticklabels=False, title=' ')fig.update_xaxes(title=' ')fig.update_traces(texttemplate='%{y}', textposition='outside', hovertemplate = '<b>%{x}</b><br>No. of Transactions: %{y}')fig.show()
```

3220307630272748264710481041924741739700596MarchNovemberJanuaryFebruaryDecemberAprilOctoberMayJulyJuneAugustSeptemberMost Productive Month

The bakery seems to be heavily occupied and makes most of its business from November to March.

EDA Summary:

Coffee is the best-selling product by far, followed by bread and tea. The bakery seems to be making most of its sales in the afternoon everyday with over 56% of the sales. Sales fall sharply after that. However the bakery makes a decent amount of sales in the morning as well. For obvious reasons, the sales are high as expected during the weekends. However the sales seem to be quite uniform rest of the days. The bakery seems to be heavily occupied and makes most of its business from November to March.

4. Association Rules Generation

4.1 Data Preparation for Association Rule Mining

Apriori algorithm requires a dataframe with all the transactions one hot encoded for all the items.

list of all the transactions

```
transactions=[]for item in bakeryDF['TransactionNo'].unique():

    lst=list(set(bakeryDF[bakeryDF['TransactionNo']==item]['Items']))
```

```
    transactions.append(lst)
```

```
transactions[0:10]
```

[['Bread'],

ut[16]

['Scandinavian'],

['Hot chocolate', 'Jam', 'Cookies'],

['Muffin'],

['Bread', 'Coffee', 'Pastry'],

['Muffin', 'Pastry', 'Medialuna'],

['Tea', 'Coffee', 'Pastry', 'Medialuna'],

['Bread', 'Pastry'],

['Muffin', 'Bread'],

['Scandinavian', 'Medialuna']]

one hot encoding

```
te = TransactionEncoder()encodedData = te.fit(transactions).transform(transactions)data = pd.DataFrame(encodedData, columns=te.columns_)data.head()
```

	Adjustment	Afternoon with the baker	Alfajores	Argentina Night	Art Tray	Bacon	Baguette	Bakewell	Bare Popcorn	Basket	...	The BART	The Nomad	Tiffin	Toast	Truffles	Tshirt	Valentine's card	Vegan Feast	Vegan mincepie	Victorian Sponge
0	False	False	Fals	False	Fal	Fals	Fals	Fals	Fals	Fals	...	Fal	Fals	Fal	Fal	Fals	Fal	False	Fals	Fals	Fals

	Adjustment	Afternoon with the baker	Alfajores	Argentina Night	Art Tray	Bacon	Baguette	Bakewell	Bare Popcorn	Basket	...	The BART	The Nomad	Tiffin	Toast	Truffles	Tshirt	Valentine's card	Vegan Feast	Vegan mincepie	Victorian Sponge
			e		se	e	e	e	e	e		se	e	se	se	e	se		e	e	e
1	False	False	False	False	False	False	False	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	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

5 rows × 94 columns

4.2 Association Rules Generation

frequent items

:

```
frequentItems= apriori(data, use_colnames=True, min_support=0.02)frequentItems.head()
```

	support	itemsets
0	0.036344	(Alfajores)

	support	itemsets
1	0.327205	(Bread)
2	0.040042	(Brownie)
3	0.103856	(Cake)
4	0.478394	(Coffee)

```
rules = association_rules(frequentItems, metric="lift", min_threshold=1)rules.antecedents = rules.antecedents.apply(lambda x: next(iter(x)))rules.consequents = rules.consequents.apply(lambda x: next(iter(x)))rules.head()
```

Out[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	Bread	Pastry	0.327205	0.086107	0.029160	0.089119	1.034977	0.000985	1.003306
1	Pastry	Bread	0.086107	0.327205	0.029160	0.338650	1.034977	0.000985	1.017305
2	Cake	Coffee	0.103856	0.478394	0.054728	0.526958	1.101515	0.005044	1.102664
3	Coffee	Cake	0.478394	0.103856	0.054728	0.114399	1.101515	0.005044	1.011905
4	Cake	Tea	0.103856	0.142631	0.023772	0.228891	1.604781	0.008959	1.111865

4.3 Rules Visualization

In [20]:

```
network_A = list(rules["antecedents"].unique())network_B = list(rules["consequents"].unique())node_list = list(set(network_A + network_B))G = nx.Graph()
for i in node_list:

    G.add_node(i)for i,j in rules.iterrows():

    G.add_edges_from([(j["antecedents"], j["consequents"])])pos = nx.spring_layout(G, k=0.5, dim=2, iterations=400)for n, p in pos.items():

    G.nodes[n]['pos'] = p

edge_trace = go.Scatter(x=[], y=[], line=dict(width=0.5, color='#888'), hoverinfo='none', mode='lines')

for edge in G.edges():

    x0, y0 = G.nodes[edge[0]]['pos']

    x1, y1 = G.nodes[edge[1]]['pos']
```

```
    edge_trace['x'] += tuple([x0, x1, None])
    edge_trace['y'] += tuple([y0, y1, None])
node_trace = go.Scatter(x=[], y=[], text=[], mode='markers', hoverinfo='text',
    marker=dict(showscale=True, colorscale='Burg', reversescale=True, color=[], size=15,
    colorbar=dict(thickness=10, title='Node Connections', xanchor='left', titleside='right')))
for node in G.nodes():
    x, y = G.nodes[node]['pos']
    node_trace['x'] += tuple([x])
    node_trace['y'] += tuple([y])
for node, adjacencies in enumerate(G.adjacency()):
    node_trace['marker']['color']+=tuple([len(adjacencies[1])])
    node_info = str(adjacencies[0]) +'<br>No of Connections: {}'.format(str(len(adjacencies[1])))
    node_trace['text']+=tuple([node_info])
fig = go.Figure(data=[edge_trace, node_trace],
    layout=go.Layout(title='Item Connections Network', titlefont=dict(size=20),
    plot_bgcolor='white', showlegend=False, margin=dict(b=0,l=0,r=0,t=50),
    xaxis=dict(showgrid=False, zeroline=False, showticklabels=False),
    yaxis=dict(showgrid=False, zeroline=False, showticklabels=False)))
iplot(fig)
```

12345678Node ConnectionsItem Connections Network

5. Refining Rules

The confidence for a very frequent consequent is always high even if there is a very weak association. So this doesn't give a clearer picture. Here, coffee is by far the most frequent item and the best seller. It can therefore be recommended anyway with every other item. So, we can drop the rules recommending coffee to get a clearer picture of the real unknown rules generated from the data.

In [21]:

```
iMarket basket insight, often referred to as "Market Basket Analysis" or "Affin
ity Analysis," is a data-driven technique used by businesses, particularly in t
he retail and e-commerce sectors, to gain a deeper understanding of custom
er purchasing behavior and identify patterns in the products or items that are
 frequently bought together. This analysis is instrumental in making data-driv
en decisions for optimizing various aspects of a business, such as inventory
management, product recommendations, marketing strategies, and store lay
out design.



Market basket insight is rooted in the observation that customers tend to pur
chase certain products in combination, and these buying patterns can reveal
 valuable insights that businesses can leverage to enhance their operations.
The primary goal of market basket analysis is to uncover associations, correl
ations, and trends within transaction data to answer questions like:



1. **What items are commonly purchased together?** This helps retailers un
derstand complementary products that can be bundled or co-located in store
s.



2. **How can we improve product recommendations?** By identifying purcha
sing patterns, businesses can recommend additional products to customers
based on their current selections.



3. **What is the impact of product placements on sales?** Analyzing which p
roducts are often bought together can inform store layout and shelf arrange
ment decisions.



4. **Which customers exhibit similar buying behaviors?** Segmenting custo
mers based on their purchase history can lead to more targeted marketing a
nd personalized offers.
```

Market basket insight is achieved through various techniques, including association rule mining, collaborative filtering, and machine learning models. These techniques uncover patterns such as item co-occurrences, purchase sequences, and customer preferences. By applying these insights, businesses can improve customer satisfaction, increase revenue, and make more informed decisions regarding their product offerings and marketing strategies.

In an era where data plays a crucial role in business success, market basket insight is a valuable tool that empowers organizations to better understand their customers and adapt to their evolving needs and preferences. It has become an integral part of data-driven decision-making in the retail and e-commerce industries, ultimately leading to more efficient and customer-focused operations.

Out[21]:

	index	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	5	Tea	Cake	0.142631	0.103856	0.023772	0.166667	1.604781
1	4	Cake	Tea	0.103856	0.142631	0.023772	0.228891	1.604781
2	19	Coffee	Toast	0.478394	0.033597	0.023666	0.049470	1.472431
3	12	Coffee	Medialuna	0.478394	0.061807	0.035182	0.073542	1.189878
4	14	Coffee	Pastry	0.478394	0.086107	0.047544	0.099382	1.154168
5	10	Coffee	Juice	0.478394	0.038563	0.020602	0.043065	1.116750
6	16	Coffee	Sandwich	0.478394	0.071844	0.038246	0.079947	1.112792
7	3	Coffee	Cake	0.478394	0.103856	0.054728	0.114399	1.101515
8	6	Coffee	Cookies	0.478394	0.054411	0.028209	0.058966	1.083723
9	9	Coffee	Hot chocolate	0.478394	0.058320	0.029583	0.061837	1.060311
10	0	Bread	Pastry	0.327205	0.086107	0.029160	0.089119	1.034977
11	1	Pastry	Bread	0.086107	0.327205	0.029160	0.338650	1.034977

Summary:

Insights

- Coffee is the bestseller of this bakery and it shows association with 8 other items.

- Over 11% coffee lovers also buy cake along with while almost 10% of them buy pastry along with it.
- Over 16% of tea consumers also buy cakes and over 22% cake lovers also buy tea.
- Among the pastry lovers, over 33% of them also buy bread, while nearly 9% of those who buy pastry also buy bread.