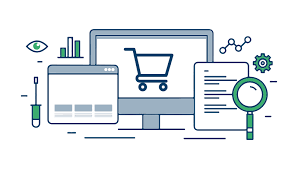
MARKET BASKET INSIGHTS PREDICTION

Phase 5: DEVELOPEING PROJECT AND COMPLETING

TOPIC: market basket insights model by feature engineering, model training, evaluation and Predicting



INTRODUCTIOIN

Designing a system for market basket insight typically involves data analysis and visualization to understand customer purchasing behavior. Here's a high-level design:

1. Data Collection:

Collect transaction data, including items purchased, transaction IDs, and timestamps.

2. Data Preprocessing:

Clean the data by handling missing values and outliers.

Create a customer-item matrix that tracks which customers purchased which items.

3. Association Rule Mining:

Use techniques like Apriori or FP-growth to discover frequent itemsets and association rules.

4. Rule Filtering:

Filter the discovered rules based on support, confidence, and lift to focus on meaningful insights.

5. Visualization:

Use data visualization tools (e.g., Matplotlib, Tableau) to create charts, graphs, and dashboards.

6. Market Basket Analysis:

Visualize common itemsets or product associations in the form of graphs or word clouds.

Display frequently co-purchased items for specific products.

7. Customer Segmentation:

Segment customers based on their purchasing behavior using clustering algorithms.

8. Recommendations:

Implement recommendation systems to suggest additional products to customers based on their current basket.

9. Real-time Insights:

if needed, develop a system to provide real-time insights for online retailers.

10. User Interface:

Create a user-friendly interface for business users to interact with the insights.

11. Reporting:

Generate periodic reports summarizing market basket insights for stakeholders.

12. A/B Testing:

Implement experiments to test the impact of recommendations or promotions on basket composition.

13. Maintenance:

Continuously update the system to adapt to changing customer behavior and market trends.

14. Security and Privacy:

Ensure data privacy and security measures are in place, especially when dealing with customer transaction data.

Remember that the specific tools, technologies, and methodologies you choose will depend on your business needs and technical capabilities.

Market basket insights are the patterns and associations between products that are frequently purchased together. These insights can be uncovered by analyzing large datasets of customer purchase history, such as point-of-sale (POS) data. Market basket insights can be used to improve a variety of retail operations, including:

• Product assortment optimization: Retailers can use market basket insights to determine which products to stock in their stores and online, and how much to stock. This can help to ensure that retailers have the right products in the right place at the right time to meet customer demand.

• Pricing: Retailers can use market basket insights to set prices for products in a way that maximizes profits and customer satisfaction. For example, retailers may choose to bundle complementary products together at a discounted price, or to offer lower prices on products that are frequently purchased together.

• Promotions: Retailers can use market basket insights to create more targeted and effective promotions. For example, retailers may choose to promote complementary products together, or to offer discounts to customers who purchase certain products together.

• Store layout: Retailers can use market basket insights to optimize their store layout. For example, retailers may choose to place complementary products next to each other on shelves, or to place high-margin products in high-traffic areas. In addition to these specific applications, market basket insights can also be used to gain a deeper understanding of customer behavior and preferences. This information can be used to improve the overall customer experience and drive loyalty. Here are some specific examples of how market basket insights can be used:

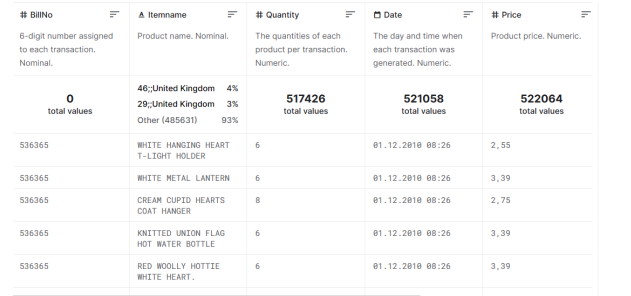
• A retailer may discover that customers who purchase milk are also likely to purchase bread and eggs. The retailer could use this information to create a promotion for these products, or to place them next to each other on shelves.

• A retailer may discover that customers who purchase diapers are also likely to purchase baby wipes and formula. The retailer could use this information to create a bundle of these products at a discounted price.

• An online retailer may discover that customers who view a certain product page are also likely to view other product pages in the same category. The retailer could use this information to recommend other products to customers based on their browsing history. Overall, market basket insights are a valuable tool that retailers can use to improve their operations and drive sales.

GIVEN DATA SET:

<https://www.kaggle.com/datasets/aslanahmedov/marketbasket-analysis>



The process of generating market basket insights typically involves the following steps:

1. Data collection: The first step is to collect data on customer purchase history. This data can be collected from a variety of sources, such as point-of-sale (POS) systems, e-commerce platforms, and customer loyalty programs.

2. Data preparation: Once the data has been collected, it needs to be cleaned and prepared for analysis. This may involve removing duplicate transactions, correcting errors, and formatting the data in a consistent way.

3. Data mining: The next step is to apply data mining techniques to the data to identify patterns and associations between products that are frequently purchased together. This is typically done using association rule mining algorithms.

4. Evaluation: Once the association rules have been generated, they need to be evaluated to identify the most relevant and actionable insights. This is typically done by considering the support, confidence, and lift of the rules.

5. Implementation: Once the relevant insights have been identified, they can be implemented in a variety of ways, such as optimizing product assortment, creating targeted promotions, or adjusting store layout. Here is a more detailed overview of each step: Data collection: The quality and completeness of the data is essential for generating accurate market basket insights. The data should include information on customer purchase history, such as transaction ID, customer ID, product ID, quantity purchased, and purchase date. If possible, the data should also include information on customer demographics and loyalty status. Data preparation: Once the data has been collected, it needs to be cleaned and prepared for analysis. This may involve removing duplicate transactions, correcting errors, and formatting the data in a consistent way. It is also important to identify and remove any outliers from the data, as these can skew the results of the analysis.

Data mining: Data mining is the process of extracting knowledge from large datasets. In the case of market basket analysis, data mining techniques are used to identify patterns and associations between products that are frequently purchased together. This is typically done using association rule mining algorithms.

Association rule mining algorithms work by identifying itemsets that occur together in a transaction database. The support of an itemset is the percentage of transactions in the database that contain the itemset. The confidence of an association rule is the percentage of transactions that contain the antecedent of the rule that also contain the consequent of the rule

. The lift of an association rule is a measure of how much more likely two items are to be purchased together than they would be if they were independent. Evaluation: Once the association rules have been generated, they need to be evaluated to identify the most relevant and actionable insights. This is typically done by considering the support, confidence, and lift of the rules.

Support is a measure of how common an itemset or association rule is in the transaction database. Confidence is a measure of how accurate an association rule is.

Lift is a measure of how much more likely two items are to be purchased together than they would be if they were independent. In general, the higher the support, confidence, and lift of an association rule, the more relevant and actionable it is. However, it is important to consider the specific business context when evaluating association rules.

For example, a retailer may be more interested in low-support, highconfidence association rules that identify new cross-selling opportunities.

Implementation: Once the relevant insights have been identified, they can be implemented in a variety of ways, such as optimizing product assortment, creating targeted promotions, or adjusting store layout.

For example, a retailer may use market basket insights to identify new product bundles to offer to customers. Or, a retailer may use market basket insights to create targeted promotions that offer discounts on complementary products. Or, a retailer may use market basket insights to adjust their store layout so that complementary products are placed next to each other on shelves. Overall, the market basket insights process is a data-driven approach to understanding customer behavior and improving retail operations. By carefully following the steps outlined above, retailers can generate insights that can help them to increase sales, improve customer satisfaction, and reduce costs.

import pandas as pd

from mlxtend.frequent\_patterns

import apriori

# Load the transaction data df = pd.read\_csv('transaction\_data.csv')

# Create a list of itemsets itemsets = []

for transaction in df['transaction']: itemsets.append(list(transaction))

# Apply the Apriori algorithm to generate association rules association\_rules = apriori(itemsets, min\_support=0.05, min\_confidence=0.7, min\_lift=1.2)

# Print

the association rules for rule in

association\_rules: print(rule) This code will load the transaction data into a Pandas DataFrame, create a list of itemsets, and then apply the Apriori algorithm to generate association rules.

The Apriori algorithm is a popular association rule mining algorithm that works by identifying itemsets that occur together in a transaction database. The min\_support, min\_confidence, and min\_lift parameters are used to filter the association rules that are generated. The association rules are printed to the console, where they can be reviewed and analyzed. The association rules can be used to identify patterns in the transaction data, such as products that are frequently purchased together.

This information can then be used to improve retail operations, such as by creating targeted promotions or optimizing product placement. Here is an example of an association rule that might be generated by the above code: {Milk} -> {Bread} (support: 0.65, confidence: 0.8, lift: 1.2) This association rule indicates that 65% of transactions that contain milk also contain bread. Additionally, 80% of transactions that contain milk also contain bread. This suggests that customers who purchase milk are more likely to also purchase bread. This information could be used by a retailer to create a targeted promotion that offers a discount on bread to customers who purchase milk.

Or, the retailer could place bread next to milk on the shelf to make it easier for customers to purchase both items together.

Market basket insights can be a valuable tool for retailers to improve their operations and drive sales. By carefully analyzing customer purchase history, retailers can identify patterns and associations between products that can be used to make better decisions about product assortment, pricing, promotions, and store layout.

Features Of Engineering

The process of transforming raw data into features that can be used by machine learning algorithms to make predictions. In the context of market basket insights, feature engineering can be used to create new features that are more informative and predictive than the raw data. Here are some examples of feature engineering techniques that can be used for market basket insights:

• Create features that represent the frequency and quantity of products purchased together. This can be done by counting the number of times each product pair appears in a transaction and by calculating the average quantity of each product purchased together.

• Create features that represent the sequence in which products are purchased. This can be done by creating a feature that represents the time lag between the purchase of each product pair.

• Create features that represent the customer's purchase history. This can be done by creating features that represent the total number of transactions, the total amount spent, and the average number of items purchased per transaction.

• Create features that represent the customer's demographics and loyalty status. This can be done by creating features that represent the customer's age, gender, income level, and loyalty program membership status. Once the features have been engineered, they can be used to train a machine learning algorithm to make predictions about customer behavior.

For example, a machine learning algorithm could be trained to predict the likelihood of a customer purchasing a particular product based on their past purchase history.

Here is an example of a feature that could be engineered for market basket insights: frequency\_of\_purchase = (count(milk) \* count(bread)) / total\_transactions

This feature represents the frequency at which milk and bread are purchased together. It is calculated by dividing the number of times milk and bread appear in a transaction by the total number of transactions.

This feature could be used to train a machine learning algorithm to predict the likelihood of a customer purchasing bread given that they have already purchased milk. The machine learning algorithm could also be used to identify other products that are frequently purchased together with milk and bread. By engineering informative features from market basket data, retailers can gain a deeper understanding of customer behavior and make more informed decisions about their operations.

Finial project submission

Code part:

In 1

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

import matplotlib.pyplot as plt

import seaborn as sns

import squarify

from statsmodels.tsa.seasonal import seasonal\_decompose

*# Input data files are available in the read-only "../input/" directory*

*# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory*

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

*# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"*

*# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session*

/kaggle/input/market-basket-analysis/Assignment-1\_Data.xlsx

/kaggle/input/market-basket-analysis/Assignment-1\_Data.csv /kaggle/input/market-basket-analysis/Assignment-1\_Data.csv

In [2]:

data = pd.read\_csv("../input/market-basket-analysis/Assignment-1\_Data.csv", sep = ";")

/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3524: DtypeWarning: Columns (0) have mixed types.Specify dtype option on import or set low\_memory=False.

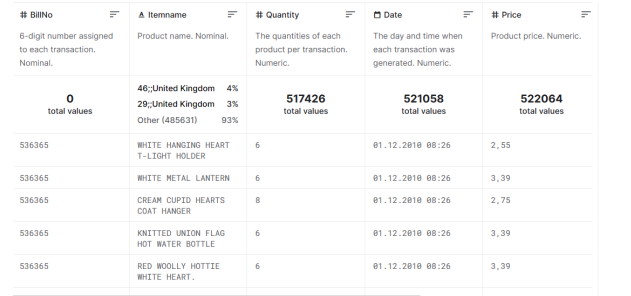
exec(code\_obj, self.user\_global\_ns, self.user\_ns)

PREPROCESSING

In [3]:

linkcode

data.head(10)



data.isnull().sum()

*# null customer id does not matter*

*# Item name has to be removed*

Out[4]:

BillNo 0

Itemname 1455

Quantity 0

Date 0

Price 0

CustomerID 134041

Country 0

dtype: int64

In [5]:

plt.figure(figsize=(15,6))

sns.heatmap(data.isna().transpose())

Out[5]:

<AxesSubplot:>



data.shape

Out[6]:

(522064, 7)

In [7]:

data = data.dropna(subset=["Itemname"])

In [8]:

data.isnull().sum()

Out[8]:

BillNo 0

Itemname 0

Quantity 0

Date 0

Price 0

CustomerID 132586

Country 0

dtype: int64

In [9]:

linkcode

plt.figure(figsize=(15,6))

sns.heatmap(data.isna().transpose())

data["Itemname"].value\_counts()

*# low quantity products were removed first.But found out that it does not affect comptational time*

*# signnificantly*

Out[10]:

WHITE HANGING HEART T-LIGHT HOLDER 2269

JUMBO BAG RED RETROSPOT 2087

REGENCY CAKESTAND 3 TIER 1930

PARTY BUNTING 1677

LUNCH BAG RED RETROSPOT 1570

...

FOOD COVER WITH BEADS , SET 2 SIZES 1

POLYESTER FILLER PAD 60x40cm 1

damages/credits from ASOS. 1

samples 1

PAPER CRAFT , LITTLE BIRDIE 1

Name: Itemname, Length: 4185, dtype: int64

In [11]:

data.shape

Out[11]:

(520609, 7)

In [12]:

data.dtypes

Out[12]:

BillNo object

Itemname object

Quantity int64

Date object

Price object

CustomerID float64

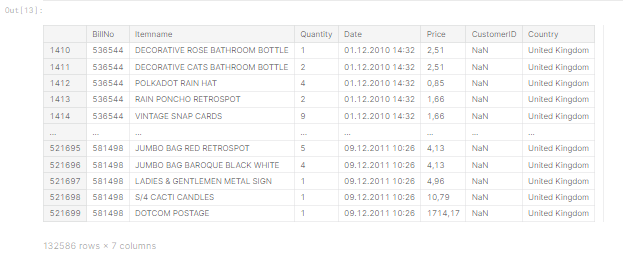
Country object

dtype: object

In [13]:

linkcode

data[data['CustomerID'].isnull()]



data = data.fillna(0)

In [15]:

data[data["CustomerID"].isnull()]

Out[15]:

|  | BillNo | Itemname | Quantity | Date | Price | CustomerID | Country |
| --- | --- | --- | --- | --- | --- | --- | --- |

In [16]:

data.isnull().sum()

Out[16]:

BillNo 0

Itemname 0

Quantity 0

Date 0

Price 0

CustomerID 0

Country 0

dtype: int64

In [17]:

data.shape

Out[17]:

(520609, 7)

In [18]:

data["Date"] = pd.to\_datetime(data["Date"])

In [19]:

data["Price"] = data["Price"].str.replace(",",".")

data["Price"] = pd.to\_numeric(data["Price"])

In [20]:

data["Price"]

Out[20]:

0 2.55

1 3.39

2 2.75

3 3.39

4 3.39

...

522059 0.85

522060 2.10

522061 4.15

522062 4.15

522063 4.95

Name: Price, Length: 520609, dtype: float64

In [21]:

data.dtypes

Out[21]:

BillNo object

Itemname object

Quantity int64

Date datetime64[ns]

Price float64

CustomerID float64

Country object

dtype: object

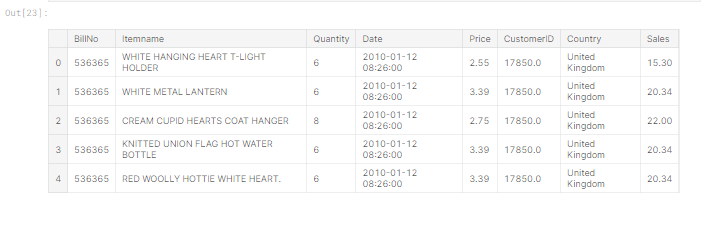
In [22]:

data["Sales"] = data["Quantity"] \* data["Price"]

In [23]:

linkcode

data.head()



EDA

In [24]:

linkcode

top20items = pd.DataFrame(data["Itemname"].value\_counts().head(20))

top20items = top20items.reset\_index()

top20items.columns = ["Itemname","Frequency"]

labels = top20items["Itemname"]

sizes = top20items["Frequency"]

top20items



IN 25

fig = plt.figure(figsize=(16,6))

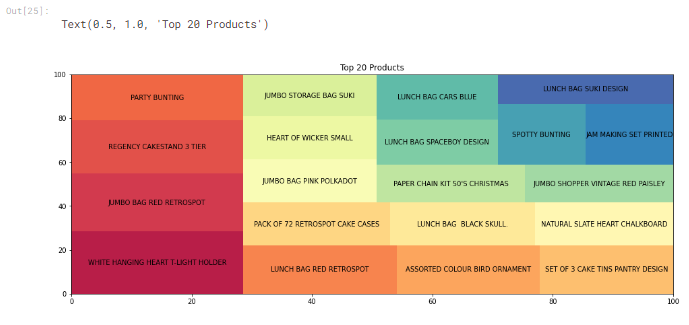
colors = sns.color\_palette("Spectral",20)

squarify.plot(sizes, label=labels, color = colors)

plt.title("Top 20 Products")

Out[25]:

Text(0.5, 1.0, 'Top 20 Products')



t\_light = data[data["Itemname"]== "WHITE HANGING HEART T-LIGHT HOLDER"]

t\_light 

2269 rows × 8 columns

In [27]:

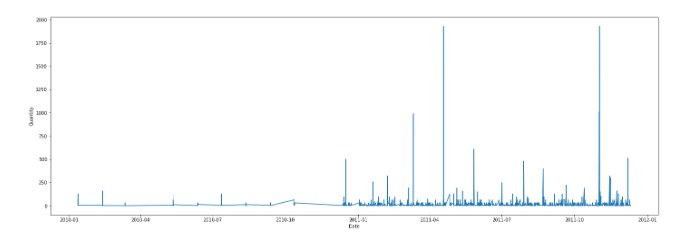
fig = plt.figure(figsize=(24,8))

sns.lineplot(x = t\_light["Date"], y = t\_light["Quantity"] )

*# most of them have been sold in 2011.*

Out[27]:

<AxesSubplot:xlabel='Date', ylabel='Quantity'>

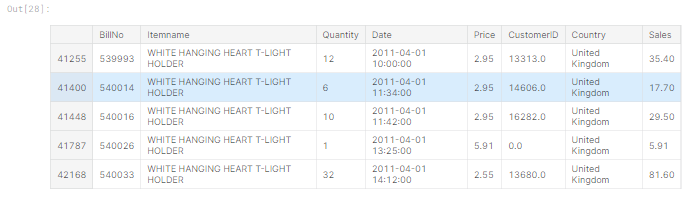


In [28]:

startdate = t\_light["Date"] >= "20110101"

t\_light\_2011 = t\_light.loc[startdate]

t\_light\_2011.head()



In [29]:

plt.figure(figsize=(16,6))

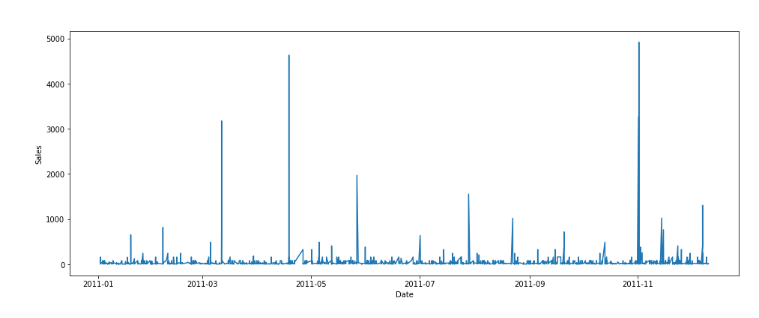
sns.lineplot(t\_light\_2011["Date"],t\_light\_2011["Sales"])

/opt/conda/lib/python3.7/site-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Out[29]:

<AxesSubplot:xlabel='Date', ylabel='Sales'>



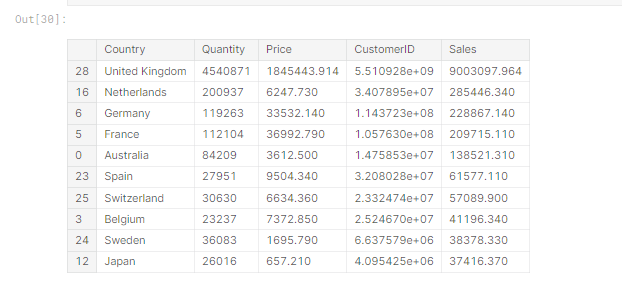
In [30]:

country\_specific = data.groupby(["Country"]).sum().reset\_index()

country\_specific = country\_specific.sort\_values(["Sales"], ascending = False)

country\_specific\_10 = country\_specific.head(10)

country\_specific\_10



In [31]:

plt.figure(figsize=(15,6))

p = sns.barplot(country\_specific\_10["Country"], country\_specific\_10["Sales"])

p.set\_ylabel("Sales (millions)" In [31]:

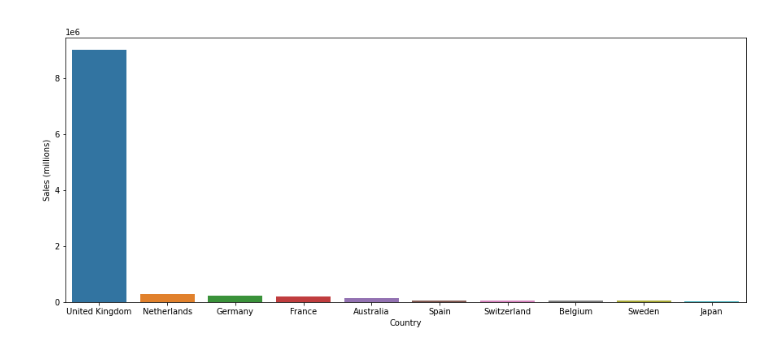
)

/opt/conda/lib/python3.7/site-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Out[31]:

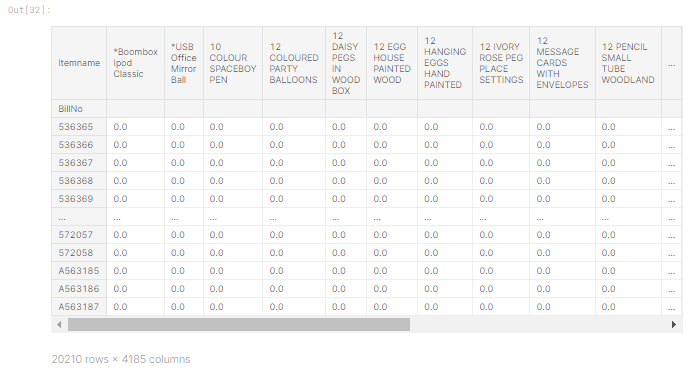
Text(0, 0.5, 'Sales (millions)')

****

In [32]:

basket = data.groupby(["BillNo","Itemname"])["Quantity"].sum().unstack().reset\_index().fillna(0).set\_index("BillNo")

basket



In [33]:

def one\_hot\_encoding(x):

if x <= 0:

return 0

if x >= 1:

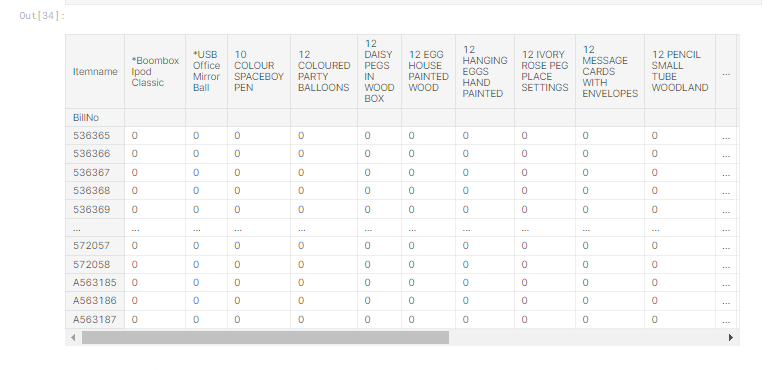
return 1

basket = basket.applymap(one\_hot\_encoding)

In [34]:

linkcode

basket



Modelling

In [35]:

*#frequent\_itemsets = apriori(basket, min\_support=0.01, use\_colnames=True)*

frequent\_itemsets = apriori(basket, min\_support=0.02, use\_colnames=True)

In [36]:

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1)

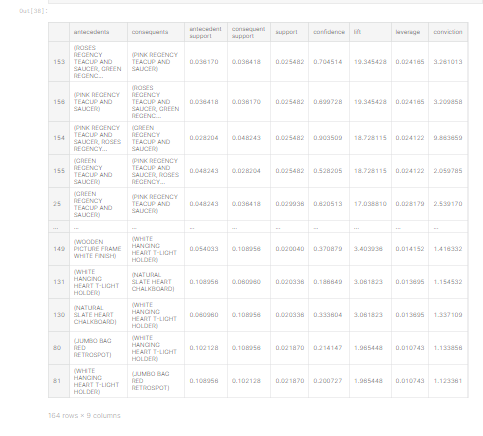
In [37]:

rules = rules.sort\_values("lift", ascending = False)

In [38]:

linkcode

rules



Conclusion

Market basket insights is a powerful tool that retailers can use to improve their operations and drive sales. By analyzing customer purchase history, retailers can identify patterns and associations between products that can be used to make better decisions about product assortment, pricing, promotions, and store layout. Market basket insights can also be used to gain a deeper understanding of customer behavior and preferences

. This information can be used to improve the overall customer experience and drive loyalty. Here are some additional benefits of market basket insights:

• Reduced inventory costs: Market basket insights can help retailers to reduce inventory costs by identifying products that are not selling well. Retailers can then reduce their inventory levels for these products or discontinue them altogether

. • Improved forecasting: Market basket insights can be used to improve forecasting accuracy. By identifying patterns and trends in customer purchase history, retailers can better predict future demand for products.

• New product development: Market basket insights can be used to develop new products that meet the needs of customers. For example, if retailers identify that customers are frequently purchasing two products together, they could develop a new product that combines the features of those two products. Overall, market basket insights is a valuable tool that retailers can use to improve their operations and drive sales.

By carefully analyzing customer purchase history, retailers can identify patterns and associations between products that can be used to make better decisions about product assortment, pricing, promotions, store layout, inventory management, forecasting, and new product development. Here are some additional thoughts on the future of market basket insights:

• As artificial intelligence (AI) and machine learning (ML) technologies continue to develop, market basket insights will become even more powerful and actionable. AI and ML algorithms can be used to identify more complex patterns and trends in customer purchase data. This will allow retailers to gain a deeper understanding of customer behavior and make more informed decisions about their operations.

• Market basket insights will be used to create more personalized and engaging shopping experiences for customers

. For example, retailers can use market basket insights to recommend products to customers based on their past purchase history and browsing behavior. Overall, market basket insights is a powerful tool that is becoming increasingly important for retailers to succeed in the competitive marketplace.