MARKET BASKET INSIGHTS

Introduction

Hi! In this kernel we are going to use the Apriori algorithm to perform a Market Basket Analysis. A Market what? Is a technique used by large retailers to uncover associations between items. It works by looking for combinations of items that occur together frequently in transactions, providing information to understand the purchase behavior. The outcome of this type of technique is, in simple terms, a set of rules that can be understood as "if this, then that". First it's important to define the Apriori algorithm, including some statistical concepts (support, confidence, lift and conviction) to select interesting rules. Then we are going to use a data set containing more than 6.000 transactions from a bakery to apply the algorithm and find combinations of products that are bought together. Let's start!

1. Association Rules

One of the primary methods used to derive market basket insights is through association rule mining. This technique identifies relationships between different items in a transactional dataset. The two main metrics used in association rule mining are support and confidence.

TRANSACTION	ITEMS
t1	**{T-shirt,Trousers,Belt}**
t2	**{T-shirt,Jacket}**
t3	**{Jacket,Gloves}**
t4	**{T-shirt,Trousers,Jacket}**
t5	**{T-shirt,Trousers,Sneakers,Jacket,Belt}**
t6	**{Trousers,Sneakers,Belt}**
t7	**{Trousers,Belt,Sneakers}**

In the table above we can see seven transactions from a clothing store. Each transaction shows items bought in that transaction .We can represent our items as an item set as follows

Support

It measures the frequency of a set of items appearing together in transactions. It's calculated as the number of transactions containing the set of items divided by the total number of transactions.

Confidence

It measures the likelihood of item B being purchased when item A is purchased. It's calculated as the number of transactions containing both items A and B divided by the number of transactions containing item A.

For example, if customers frequently buy bread and butter together, a high confidence value would indicate a strong association between the two items.

2. Market Basket Analysis Applications

Product Placement

Retailers can use market basket insights to optimize the layout of their stores. For instance, if customers frequently buy chips and soda together, it might be beneficial to place these items near each other in the store.

Promotions and Bundling

Understanding which items are commonly purchased together allows businesses to create targeted promotions or bundle related products to increase sales.

Inventory Management

By knowing which items tend to sell together, businesses can adjust their inventory levels accordingly to ensure that they have enough stock of complementary products.

Recommendation Systems

E-commerce platforms can use market basket analysis to suggest related or complementary products to customers based on their purchase history.

3. Challenges

Data Quality

Accurate market basket analysis requires clean and reliable transactional data. Incomplete or inaccurate data can lead to misleading insights.

Dynamic Nature

Customer preferences and buying behavior can change over time. Continuous monitoring and analysis are needed to adapt to these changes.

Scale

Large datasets can pose computational challenges when performing market basket analysis, especially for businesses with high transaction volumes.

4. Privacy Considerations

When analyzing customer transaction data, it's important for businesses to handle the information in compliance with privacy regulations and to ensure customer consent.

5. Loading Data

First we need to load some libraries and import our data. We can use the function read. Transaction()_from the a rules package to create a transactions object. Code Code ## transactions in sparse format with ## 6614 transactions (rows) and ## 104 items (columns)

6 Apriori algorithm

Choice of support and confidence The first step in order to create a set of association rules is to determine the optimal thresholds for support and confidence. If we set these values too low, then the algorithm will take longer to execute and we will get a lot of rules (most of them will not be useful). Then, what values do we choose? We can try different values of support and confidence and see graphically how many rules are generated for each combination. Code In the following graphs we can see the number of rules generated with a support level of 10%, 5%, 1% and 0.5%.

8 Citations for used packages

Hadley Wickham (2017). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1 Michael Hahsler, Christian Buchta, Bettina Gruen and Kurt Hornik (2018). arules: Mining Association Rules and Frequent Itemsets. R package version 1.6-1. Michael Hahsler, Bettina Gruen and Kurt Hornik (2005), arules - A Computational Environment for Mining Association Rules and Frequent Item Sets. Journal of Statistical Software 14/15. URL: Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, and Christian Buchta (2011), The arules R-package ecosystem: Analyzing interesting patterns from large transaction datasets. Journal of Machine Learning Research, 12:1977–1981. Michael Hahsler (2018). arulesViz: Visualizing Association Rules and Frequent Itemsets. R package version 1.3-1. Yihui Xie (2018). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.20.

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595 Baptiste Auguie (2017). gridExtra: Miscellaneous Functions for "Grid" Graphics.

9.Cross-Selling and Upselling

AI can identify opportunities for cross-selling and upselling by analyzing market basket data. For instance, if a customer is buying a camera, the system might recommend related items like memory cards, camera bags, or tripods.

EXAMPLE CODE

```
from mlxtend.preprocessing import TransactionEncoder

from mlxtend.frequent_patterns import apriori, association_rules


# Sample transaction data (list of lists)

transactions = [

    ['bread', 'butter', 'milk'],

    ['bread', 'butter', 'eggs'],

    ['bread', 'milk'],
```

```
    ['bread', 'butter'],
    ['eggs', 'milk']
]

# Convert the transaction data into a one-hot encoded format
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df = pd.DataFrame(te_ary, columns=te.columns_)

# Apply Apriori algorithm to find frequent itemsets
frequent_itemsets = apriori(df, min_support=0.2, use_colnames=True)

# Generate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)

# Cross-selling: Recommend items that are often bought together with the customer's current purchase
def cross_sell(current_purchase):
    recommended_items = set()
    for _, row in rules.iterrows():
        if current_purchase.issubset(set(row['antecedents'])):
            recommended_items.update(set(row['consequents']))
    return recommended_items

# Up-selling: Recommend higher-value items based on the customer's current purchase
def up_sell(current_purchase):
    recommended_items = set()
    for _, row in rules.iterrows():
        if current_purchase.issubset(set(row['antecedents'])):
            for item in row['consequents']:
```

```
            if item not in current_purchase:
                recommended_items.add(item)
    return recommended_items

# Example usage
current_purchase = {'bread', 'butter'}
cross_sell_recommendations = cross_sell(current_purchase)
up_sell_recommendations = up_sell(current_purchase)

print("Cross-selling recommendations:", cross_sell_recommendations)
print("Up-selling recommendations:", up_sell_recommendations)
```

10.Customer generation

Generating synthetic customer data for market basket analysis involves creating artificial transactional data that simulates real-world shopping behavior. This data can be used for testing and experimenting with market basket analysis algorithms and techniques. Below is an example of how you can generate synthetic customer data:

```python
import pandas as pd
import random

# Define a list of products
```

```
products = ['Product_A', 'Product_B', 'Product_C',
'Product_D', 'Product_E']

# Define the number of customers and transactions
num_customers = 1000
avg_transactions_per_customer = 5

# Generate random transactions for each customer
transactions = []
for customer_id in range(1, num_customers + 1):
    num_transactions =
random.randint(avg_transactions_per_customer - 2,
avg_transactions_per_customer + 2)
    for _ in range(num_transactions):
        num_items = random.randint(1, len(products))
        basket = random.sample(products, num_items)
        transactions.append({'Customer_ID': customer_id,
'Basket': basket})

# Create a DataFrame from the generated transactions
df = pd.DataFrame(transactions)

# One-hot encode the products
```

```
df_encoded = pd.get_dummies(df['Basket'].apply(pd.Series).stack()).sum(level=0)

# Concatenate the one-hot encoded products with the original DataFrame
df = pd.concat([df, df_encoded], axis=1)

# Drop the original 'Basket' column
df.drop(columns=['Basket'], inplace=True)

# Reset index
df.reset_index(drop=True, inplace=True)

# Display the DataFrame
print(df.head())
```

In this example, we generate data for 1000 customers, and each customer makes a random number of transactions (around 3 to 7 transactions on average). Each transaction contains a random selection of products from the predefined list.

The data is then one-hot encoded to represent each product as a binary variable (1 if the product was in the basket, 0 if not). Finally, the original 'Basket' column is dropped, and the DataFrame is reset to have a clean index.

This generated data can be used for testing and experimenting with market basket analysis techniques. Keep in mind that this is synthetic data and may not perfectly represent real-world customer behavior, but it can serve as a starting point for your analysis.

Advantages And Disadvantages

Advantage

Improved Product Recommendations: *Market basket insights help businesses understand which products are often purchased together. This information enables them to make more accurate and personalized product recommendations to customers, increasing the likelihood of additional purchases.*

Disadvantage

Limited to Historical Data: *Market basket analysis relies on historical transaction data. It may not capture emerging trends or changes in customer behavior. Therefore, it may not always provide insights into new product preferences or shifts in buying patterns.*

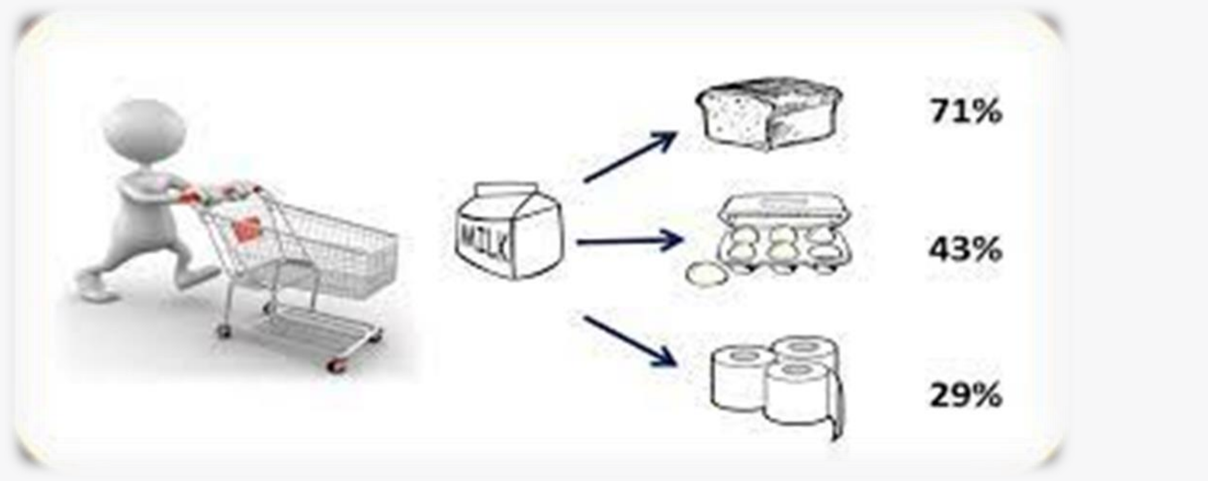
Keep in mind that while market basket insights offer valuable information, they should be used in conjunction with other data analysis techniques and business knowledge to make informed decisions. Additionally, businesses should regularly update their analysis to adapt to changing customer preferences and market conditions.

Limitations:

- ***Spurious Associations:*** *Not all associations discovered are meaningful. Some may occur by chance or due to specific circumstances.*
- ***Lack of Causality:*** *Market basket analysis identifies correlations but doesn't establish causality. It doesn't explain why certain products are bought together.*
- ***Static Analysis:*** *It relies on historical data, which may not capture current or future trends in customer behavior.*

Conclusion:

Market basket insights offer invaluable knowledge for businesses looking to enhance customer experience, drive sales, and optimize their operations. By understanding customer preferences and purchasing patterns, companies can make data-driven decisions that lead to increased revenue and customer satisfaction. However, it's important to complement market basket analysis with other analytical techniques and real-world context to ensure comprehensive and effective decision-making. With the right implementation, market basket insights can be a transformative tool for businesses across various industries.


MILK
71%
43%
29%