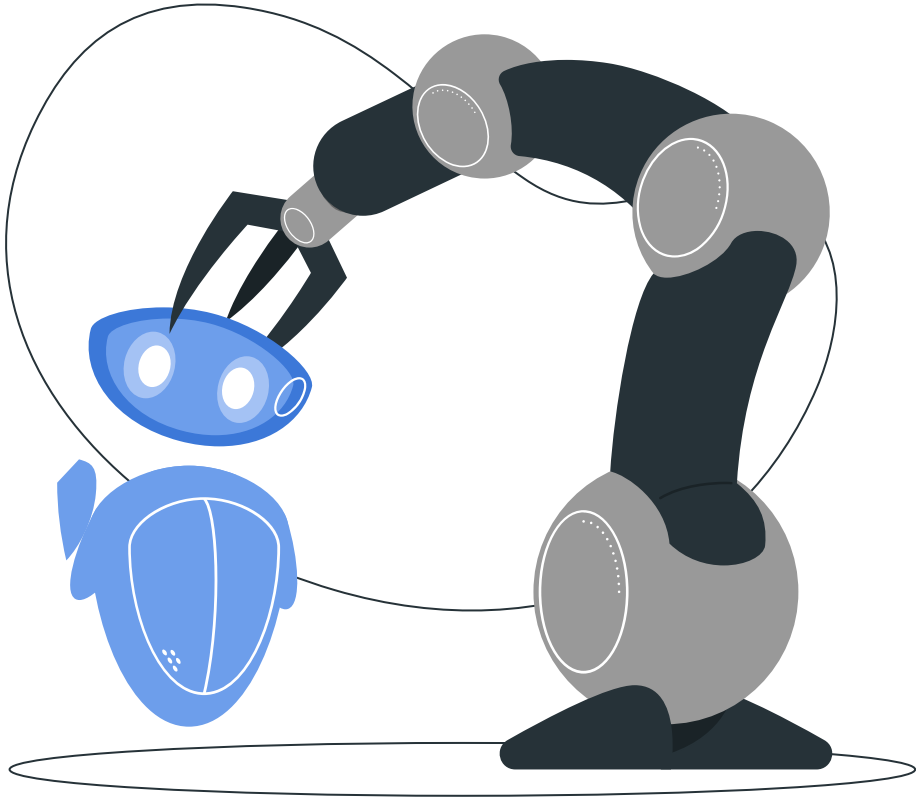
Final Deliverable

Recommendation System Through Association Rule Analysis



Khoa Nguyen, Leah Huynh, Trang Nguyen, Vy Ho, Harry Tan, John Tran

Agenda

1. Introduction
2. Dataset
3. Methods
4. Recommendation System
5. Marketing Recommendations & Insights
6. Conclusion
5. Next step

Team Members

Harry Tan
Data Science
& Economics

Khoa Nguyen
Data Science

John Tran
Data Science
& Economics

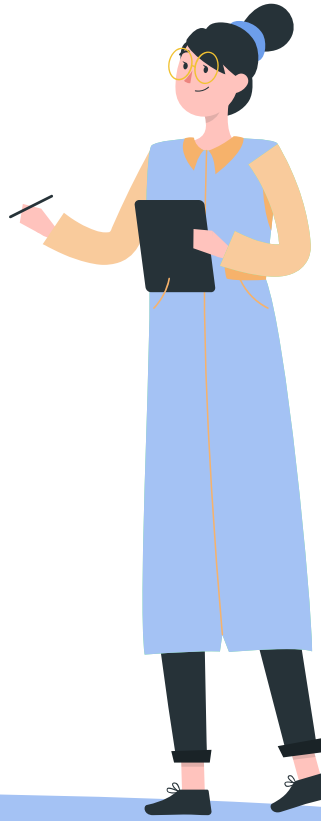
Lea Huynh
Data Science

Vy Ho
Data Science & Statistics

Trang Nguyen
Data Science &
Economics

1 Introduction

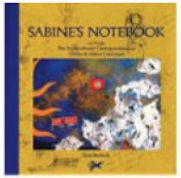
Revolutionize personalized product recommendations through a combination of Clustering and Association Rule Mining techniques.



Introduction

Top picks for you

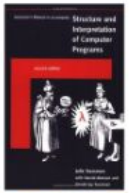
Aviation Metal & Alloys Pure Titanium Wire 0.50mm x 5M For Medical Uses or High Strength...
4.0 ★ 13
₹701.00



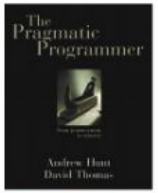
Sabine's Notebook: In Which the Extraordinary Correspondence of Griffin & Sabine Continues (Griffin and Sabine)
4.5 ★ 167

Customers Who Bought This Item Also Bought

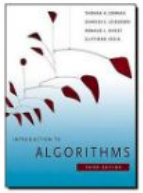
The Little Schemer - 4th Edition
› Daniel P. Friedman
4.0 ★ 64
Paperback
$36.00 Prime



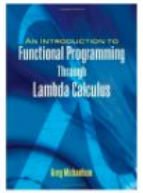
Instructor's Manual t/a Structure and Interpretation of Computer Programs...
› Gerald Jay Sussman
3.5 ★ 5
Paperback
$28.70 Prime



The Pragmatic Programmer: From Journeyman to Master
› Andrew Hunt
4.5 ★ 328
Paperback
$32.59 Prime



Introduction to Algorithms, 3rd Edition (MIT Press)
› Thomas H. Cormen
4.5 ★ 313
#1 Best Seller in Computer Algorithms
Hardcover
$66.32 Prime



An Introduction to Functional Programming Through Lambda...
› Greg Michaelson
4.5 ★ 23
Paperback
$20.70 Prime

Frequently Bought Together

+

Total price: $83.09
Add both to Cart
Add both to List

☑ **This item:** Structure and Interpretation of Computer Programs - 2nd Edition (MIT Electrical Engineering and... by Harold Abelson Paperback $50.50
☑ The Pragmatic Programmer: From Journeyman to Master by Andrew Hunt Paperback $32.59

Objectives

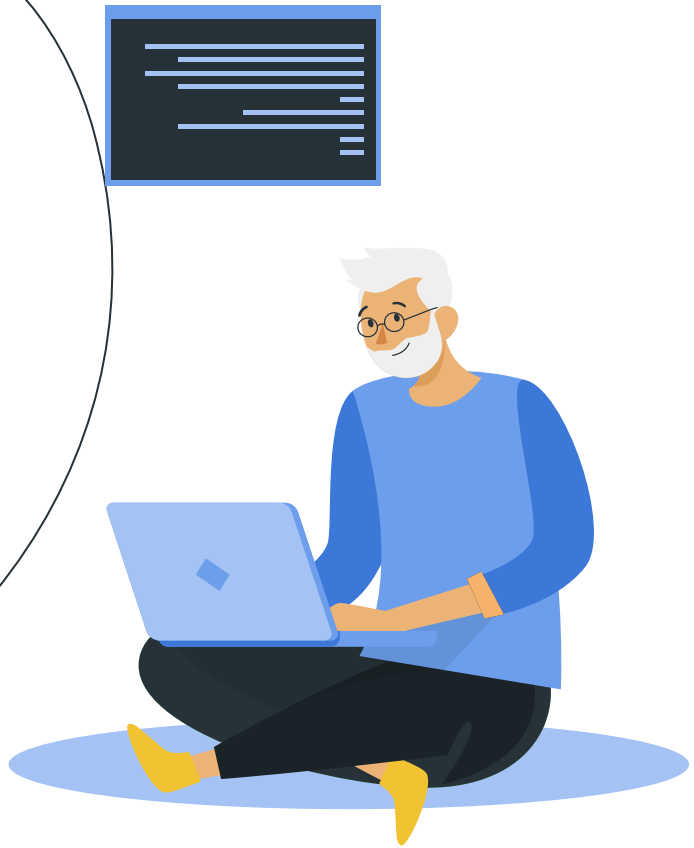
- Using ML algorithms and Data Mining, we identify customer segments to refine product recommendations, elevating the overall shopping experience

- Generated insights benefit Sales Analysts, Marketing Leads, Product Managers, and Business Owners, aiding in achieving goals like sales revenue growth and retention rate improvement

- ML/DMA enables the understanding of customer behavior via unsupervised learning (e.g., K-means clustering, RFM, and MBA) and empowers handling extensive data volumes, crucial for comprehensive data mining techniques in our project

2 Dataset

Purchase Data from various Retailers Across Multiple Countries



Dataset Overview

Online Retail Transaction

- **Number of row:** 522064

- **Number of attribute: 7**
  1. **BillNo:** 6-digit number, transaction identifier
  2. **Itemname:** Product name
  3. **Quantity:** Units per transaction
  4. **Date:** Transaction timestamp
  5. **Price:** Product price
  6. **CustomerID**: 5-digit number, customer identifier
  7. **Country:** Customer's residence

1	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
2	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850	United Kingdom
3	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850	United Kingdom
4	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850	United Kingdom
5	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850	United Kingdom
6	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850	United Kingdom

Source: Kaggle

Data Cleaning

- Filter out countries (ie. UK) with too many data points to prevent kernel from crashing
- Handle missing values by dropping rows labeled ‘NaN’ or null
- Remove purchases that were not successful, or having ‘Invoice_No’ labeled ‘C’ for ‘Cancellation’
- Correct date/time columns to appropriate date/time type and price columns to float
- Perform string manipulation to fix misspellings and extract Item names

United Kingdom	487622
Germany	9042
France	8408
Spain	2485
Netherlands	2363
Belgium	2031
Switzerland	1967
Portugal	1501
Australia	1185
Norway	1072
Italy	758
Sweden	451
Unspecified	446
Austria	398
Poland	330
Japan	321

Kernel crashing from 450,000+ rows

```
1  df.isnull().sum()
```

BillNo	0
Item	1455
Quantity	0
Date	0
Price	0
CustomerID	133490
Country	0

NaNs to be dropped

Customer Activity Dashboard

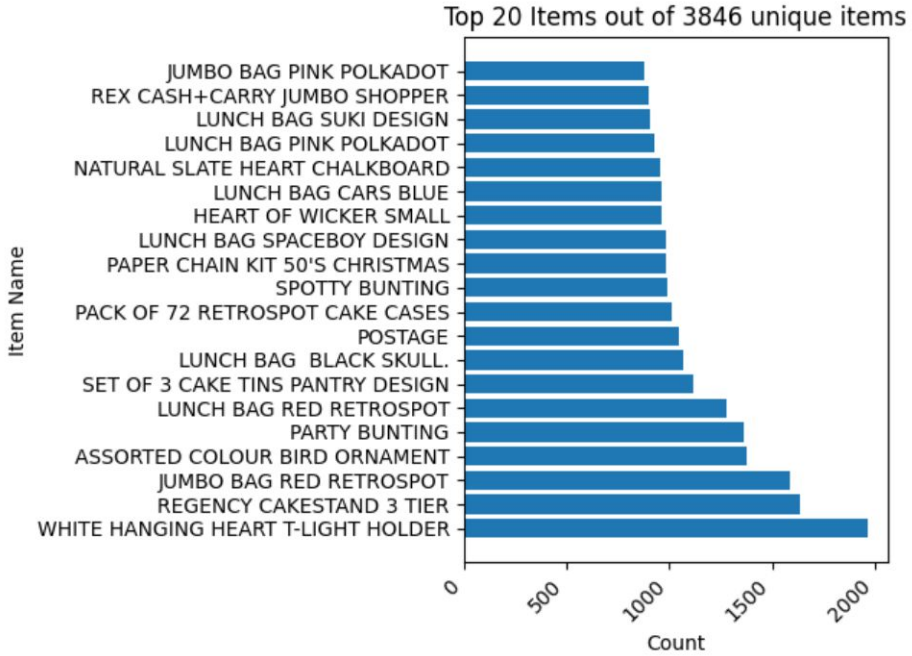
Created an interactive dashboard that summarizes statistics for different users

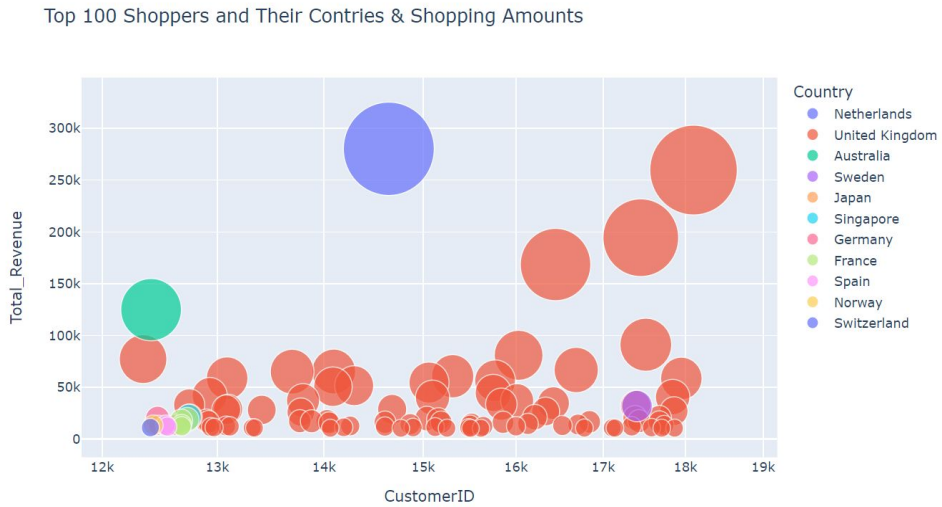
1.) Enter UserID
2.) Created 3 views
   1. Most purchased items
   2. Total amount spent
   3. Most recent transaction

```
Would you like to look at anything else?: Yes

Select which categories to view:
 1. Most purchased items
 2. Total amount spent
 3. Most recent transaction

2
You have spent a total of: $5391.21
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Depre
  and should_run_async(code)
Enter the your UserID: 17850

Select which categories to view:
 1. Most purchased items
 2. Total amount spent
 3. Most recent transaction

1
            Top 5 most purchased items  Quantity
0                 WHITE METAL LANTERN       122
1   WHITE HANGING HEART T-LIGHT HOLDER       122
2  KNITTED UNION FLAG HOT WATER BOTTLE       110
3       CREAM CUPID HEARTS COAT HANGER       108
4            HAND WARMER RED POLKA DOT       108

Would you like to look at anything else?: Yes

Select which categories to view:
 1. Most purchased items
 2. Total amount spent
 3. Most recent transaction

3
Your most recent transaction was 02.12.2010 15:27.
You bought the following items:
                                  Price
4485  HAND WARMER RED POLKA DOT   1.85
4486     HAND WARMER UNION JACK   1.85

Would you like to look at anything else?: No
```

Exploratory Data Analysis





In conclusion, given the wide range of transactions and diverse product categories observed in the dataset, implementing customer segmentation is a strategic move

3 Methods

RFM - Customer Segmentation and
K-Mean Clustering Results

Recency Frequency Monetary Theory (RFM)

Recency: Measures how recently a customer made a purchase or interacted with the business.

Frequency: Analyzes how often a customer engages or makes purchases within a specified timeframe

Monetary: Evaluates the total value or monetary contribution of a customer's purchases over a period

Application:

- Use the three factors above to segment customers into tiers or groups depending on their purchase behavior
- This helps businesses improve their marketing strategies, which in turn increases sales revenue and customer loyalty

Recency Frequency Monetary Implementation (RFM)

Visualize

	Recency int64	Frequency int64	MonetaryValue flo...	R category	F category	M category	RFM_Segment obj...	RFM_Score int64
123..	326	1	77183.6	1	1	4	1.01.04.0	6
123..	3	182	4310	4	4	4	4.04.04.0	12
123..	19	73	1757.55	3	3	4	3.03.04.0	10
123..	311	17	334.4	1	1	2	1.01.02.0	4
123..	37	85	2506.04	3	3	4	3.03.04.0	10

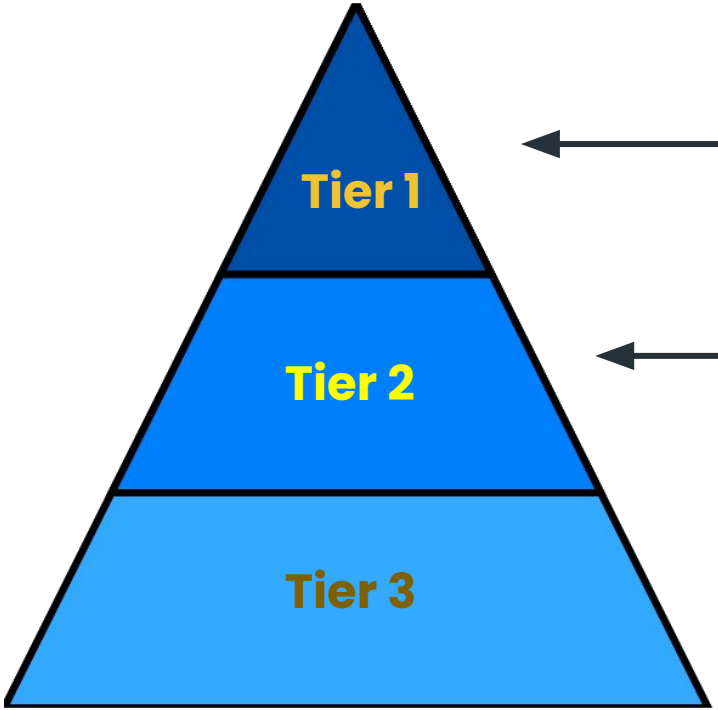
Recency: use order_date to count the number of days between hypothetical today and the last transaction

Frequency: count the total BillNo

Monetary: sum of totalPrice

RFM_score: sum of Recency, Frequency, and Monetary

Customer Segmentation



Most active customers or those with RFM scores above 9

Moderately active customers or those with RFM scores between 5 and 9

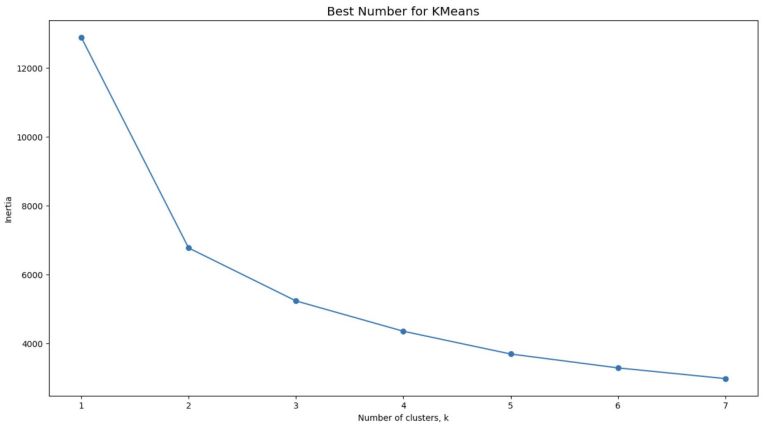
Least active customers or those with RFM scores below 5

K-Mean Clustering Implementation

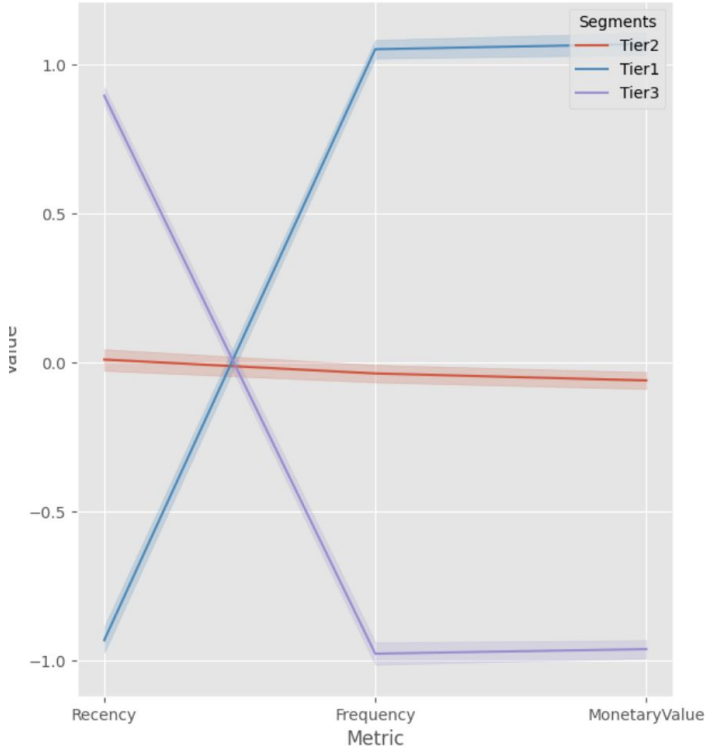
Perform K-mean clustering over 'Recency','Frequency','Monetary Value'

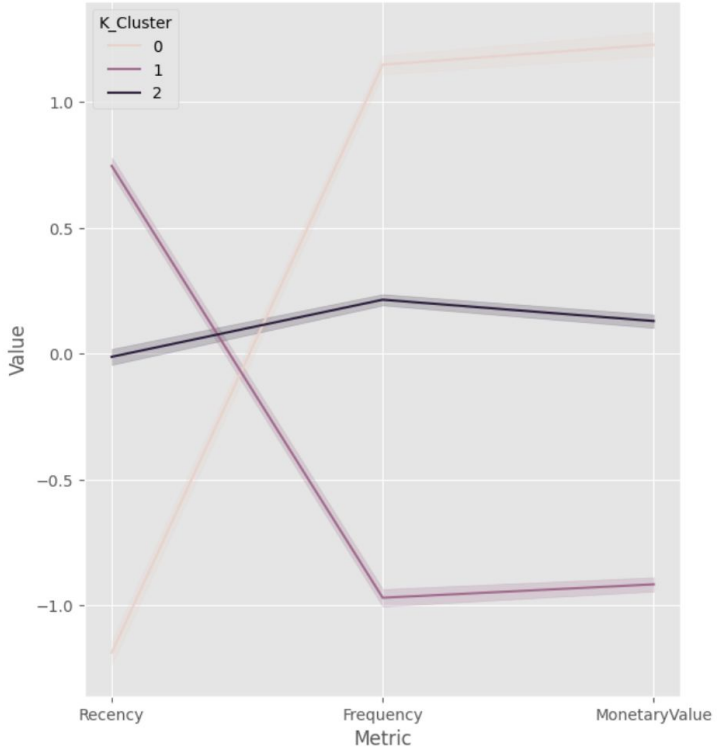
1 `rfm_kmean_k3`

	Recency int64 1 - 374	Frequency int64 1 - 7673	MonetaryValue flo... 3.75 - 280206.019...	K_Cluster int32 0 - 2
123..	326	1	77183.6	2
123..	3	182	4310	0
123..	19	73	1757.55	2
123..	311	17	334.4	1
123..	37	85	2506.04	2
123..	205	4	89	1
123..	233	58	1079.4	2
123..	215	13	459.4	1
123..	23	59	2811.43	2
123..	34	131	6207.67	0



RFM vs K-Mean Clustering



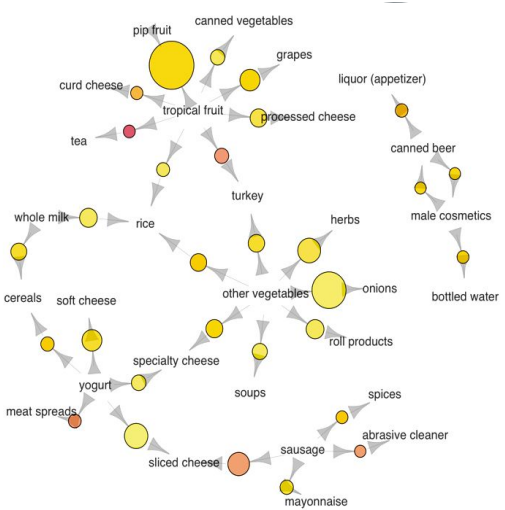


4 Item-Based Recommendation System

Promoting Sales with an Item-Based Recommendation System

Apriori Algorithm Theory

- ML algorithm that widely used method for association rule mining and mining frequent itemsets in transactional data.
  - Uses an iterative approach to find frequent itemsets based on the "**Apriori property**."
  - If {Milk, Bread} is a frequent itemset (appearing in at least X transactions), then all its subsets ({Milk} and {Bread}) must also be frequent
- <u>Benefits:</u>
  - Simplicity and ease of implementation.
    - Efficient pruning technique for reducing computational complexity.
    - Scalability to handle large datasets.
  - Widely used in industry & studied in academia.



Recommendation System Implementation

1.) Association Rules Filtering:

- Filters rules based on input items to identify relevant associations. (Lift > 1, Min_support > 0.1)

2.) Lift-Based Sorting:

- Sorts filtered rules by lift in descending order for stronger associations.

3.) Top Recommendations Extraction:

- Extracts consequents from rules, representing recommended items.

4.) Diverse Recommendations:

- Ensures a minimum number of recommendations by supplementing with top-ranked items, providing diverse and unique suggestions.

```python
# Function to generate association rules from a basket of items
def generate_association_rules(basket, min_support=0.1, metric="lift", min_threshold= 1):
    # Use Apriori algorithm to find frequent itemsets
    frequent_itemsets = apriori(basket, min_support=min_support, use_colnames=True)

    # Generate association rules using frequent itemsets
    rules = association_rules(frequent_itemsets, metric=metric, min_threshold=min_threshold)

    return rules

# Function to get recommendations based on input items and association rules
def get_recommendations(rules, input_items, num_recommendations=5):
    # Filter association rules based on input items
    filtered_rules = rules[rules['antecedents'].apply(lambda x: set(input_items).issubset(set(x)))]

    # Sort rules by lift in descending order and extract consequents
    recommended_items = filtered_rules.sort_values(by=['lift'], ascending=False)['consequents'].values

    # Ensure at least 'num_recommendations' recommendations
    if len(recommended_items) < num_recommendations:
        # Get additional items from the top-ranked items (excluding duplicates)
        additional_items = rules['consequents'].head(num_recommendations - len(recommended_items)).values
        recommended_items = np.concatenate([recommended_items, additional_items])

    # Flatten the frozensets and remove duplicates using a list
    flat_items = [item for sublist in recommended_items for item in sublist]
    unique_items = list(set(flat_items))

    # Return the top 'num_recommendations' recommended items
    return unique_items[:num_recommendations]
```

Recommendation System

- User Interaction Process:
  - **Input Tier Preference:**
    - Prompt the user to specify their preferred tier (e.g., Tier 1, Tier 2, Tier 3).
    - Example: "Ask user to input the tier (Example 1)."
  - **Item Selection by Shopper:**
    - Allow the shopper to input a specific item of interest.
    - Example: "Ask shopper to input the item. (Example: Alarm Clock Bakelike Green)
- **Recommendation Output:**
  - Utilize the Apriori algorithm to recommend the top 5 items based on the specified tier and selected item.
  - Provide personalized recommendations tailored through associations and co-occurrences among items purchased together in transactions.

Enter the tier number (1, 2, or 3):

1

Preprocessing....

Enter items (comma-separated):

ALARM CLOCK BAKELIKE GREEN

Top 5 Recommendations for Tier 1 Customers:
['SET OF 3 REGENCY CAKE TINS', 'LUNCH BAG APPLE DESIGN', 'WOODLAND CHARLOTTE BAG', 'PLASTERS IN TIN SPACEBOY', 'RED RETROSPOT MINI CASES']

5 Marketing Recommendations and Insights

Unveiling Strategic Marketing Recommendations and Customer Insights

Promotional Strategies For Retail Store

- Crafting promotional strategies for each RFM segment
- Implementing cross-selling tactics based on market basket insights
- Integrate Dynamic In-App Recommendations for User based on item purchasing

Business Strategies For Retail Store

- Use Summary Dashboard and Customer Segmentation to derive more insights
- Collaboration with online banking/e-wallet companies for discount and promotion

Potential Ethical Problems

1. Unauthorized Use of Personal Information

2. Invasive Marketing Strategy

3. Data Breaches

4. Unintended Discrimination

Potential Limitations

1. Limited demographic data for context analysis.
2. Static recommendations, not dynamically adapting to customer interests.
3. Difficulty in recommending new or niche products.
4. Limited historical data may favor established products.

→ **Acquire and integrate additional demographic data for richer context.**

→ **Collect user feedback to refine recommendations and address biases.**

→ **Develop algorithms that adapt to changing customer interests over time.**

Conclusions 6

Project overview:

- Conducted RFM Analysis for customer insights.
- Developed personalized product recommendat
- Segmented customers into 'Tier 1', 'Tier 2', 'Tier 3'.
- Used RFM quantiles and K-Means clustering.
- Implemented Apriori Algorithm for association r
- Recommends top 5 products per segment.
- Utilizes 'lift' metric for optimization.
- Enhances sales and customer satisfaction.
- Powerful tools for targeted marketing.

Business aspect: The recommendation system can enhance the precision of product recommendations and draw insights from relevant stakeholders like Sales Analysts, Customers, and Business Owners.

Next Steps

Performance Metrics

- **A/B Testing**
  - Real users selected at random see the new model, and their behavior is compared to users who saw the old model
- **Average Order Value (AOV)**
  - AOC should increase if the recommender system provides relevant product
- **Click-through rate (CTR)**
  - Higher CTR mean that customers find the recommendations relevant and interesting

q & a

Appendix

Source

1. https://www.analyticsvidhya.com/blog/2022/05/market-basket-analysis-based-on-rfm-analysis/
2. https://medium.com/nerd-for-tech/market-basket-analysis-1c38613fdd6b
3. https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/

Thank You For Listening!