# Market basket Insights using python

Team members

1.Maneesh.C

2.Diwahar.S

3.Rajamani.N

4.Muthukumar.S

5.Bavankumar.V

**Phase 2 Submission Document**

**Project: Market Basket Insights**



**Introduction:**

**Market basket insights refer to the valuable information and patterns that can be extracted from the analysis of consumer shopping baskets in retail environments. It involves the examination of the products and items that customers purchase together in a single shopping transaction. By studying market baskets, businesses can gain a deeper understanding of consumer preferences, behavior, and trends, which is crucial for making informed decisions, optimizing inventory management, and enhancing the overall shopping experience.**

**Analyzing market baskets helps businesses identify cross-selling opportunities, create targeted marketing campaigns, optimize store layouts, and improve product recommendations. This insight can also assist in inventory management by ensuring that products are stocked efficiently to meet customer demand. In essence, market basket insights empower businesses to make data-driven decisions that not only boost revenue and customer satisfaction but also enhance the overall efficiency and effectiveness of their operations.**

**Content for project phase 2:**

**Understand Customer Behavior: Analyze transaction data to gain insights into consumer behavior and purchasing patterns.**

**Optimize Inventory: Identify products that are frequently bought together to improve inventory management and stocking strategies.**

**Enhance Marketing: Use insights to create targeted marketing campaigns and improve product recommendations.**

**Improve Store Layout: Utilize data to optimize store layouts and product placements for increased sales.**

**Data source:**

**A good Data source for Market basket insights should be accurate , complete covering the geographic area of interest, accessible.**

**Data set link:** (([**https://www.kaggle.com/datasets/farjanakabirsamanta/analytics-case-studyecommerce**](https://www.kaggle.com/datasets/farjanakabirsamanta/analytics-case-studyecommerce)**)**

**Transaction ID:**

**This column contains a unique identifier for each transaction. It is typically a numeric or alphanumeric code assigned to each purchase or order. The Transaction ID helps differentiate and track individual purchases.**

**Purchased Items:**

**This column lists the items that were bought in each transaction. The "Purchased Items" column may contain a comma-separated list of product names or item codes. It represents the products or items that a customer added to their shopping cart or basket during the transaction.**

**In a real-world dataset, you might encounter additional columns that provide more context and information, such as:**

**Customer ID:**

**A unique identifier for each customer. It helps associate transactions with specific customers and enables customer-level analysis.**

**Transaction Date/Time:**

**The date and time when the transaction occurred. This information can be essential for analyzing purchase patterns over time.**

**Store Location:**

**If there are multiple store locations, this column can specify where the transaction took place. It helps assess store-specific trends and performance.**

**Product Description:**

**A detailed description of each product or item, including attributes like brand, category, price, and more. This column is crucial for understanding the characteristics of the purchased items.**

**Transaction Total:**

**The total cost of the transaction, including the sum of all purchased items. It is essential for financial analysis and assessing the value of each transaction.**

**Payment Method:**

**The payment method used for the transaction, such as cash, credit card, or mobile payment. This information is useful for understanding customer payment preferences.**

**Discounts or Coupons:**

**Any discounts, promotions, or coupons applied to the transaction, along with their details. This column helps analyze the impact of discounts on purchasing behavior.**

**Return or Refund Status:**

**Indicates whether the transaction included any returns or refunds. This information is essential for tracking customer satisfaction and managing product returns.**

**Pandas:**

**Pandas is a powerful data manipulation library that allows you to load, clean, and manipulate your dataset efficiently. You can install Pandas using pip:**

**Copy code**

**pip install pandas**

**2. mlxtend:**

**The mlxtend library provides a comprehensive set of tools for market basket analysis, including the Apriori algorithm. You can install it using pip:**

**Copy code**

**pip install mlxtend**

**To use Apriori from mlxtend, you can follow these steps:**

**python**

**Copy code**

**from mlxtend.frequent\_patterns import apriori**

**from mlxtend.frequent\_patterns import association\_rules**

**# Load your dataset into a Pandas DataFrame**

**df = pd.read\_csv("your\_dataset.csv")**

**# Perform market basket analysis using Apriori**

**frequent\_itemsets = apriori(df, min\_support=0.1, use\_colnames=True)**

**# Generate association rules**

**rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.0)**

**# Display the resulting association rules**

**print(rules)**

**3. apyori (Alternative to mlxtend):**

**If you prefer a simpler library, you can use apyori, which is lightweight and straightforward for Apriori-based market basket analysis. Install it using pip:**

**Copy code**

**pip install apyori**

**To use apyori, you can do the following:**

**python**

**Copy code**

**from apyori import apriori**

**# Load your dataset into a list of transactions**

**transactions = [**

**["item1", "item2", "item3"],**

**["item2", "item4"],**

**# Add more transactions**

**]**

**# Perform market basket analysis using Apriori**

**results = list(apriori(transactions, min\_support=0.1, min\_confidence=0.5))**

**# Display the results**

**for result in results:**

**print(result)**

**How to train and test:**

**Training the Model: In the context of market basket analysis, training involves finding frequent itemsets and generating association rules from historical transaction data. The frequent itemsets represent combinations of items that occur frequently in transactions, while the association rules indicate relationships between items. Typically, you determine the support and confidence thresholds for frequent itemset mining during the training phase.**

**Testing or Applying the Model: Once the model is trained, you can use it to make predictions or generate insights on new or future transactions. This phase involves applying the learned rules and patterns to understand customer behavior, optimize inventory, create marketing strategies, or improve store layouts.**

**Here's a step-by-step guide to train and test a market basket analysis model using Python:**

**1. Training the Model:**

**a. Data Preprocessing: Load your historical transaction data into a Pandas DataFrame or format it as a list of transactions, depending on the library you choose (e.g., mlxtend or apyori). Clean and preprocess the data as needed.**

**b. Frequent Itemset Mining: Use the Apriori algorithm (from mlxtend or apyori) to find frequent itemsets in the training data. You set the min\_support parameter to control the minimum support threshold, which determines which itemsets are considered frequent. For example:**

**python**

**Copy code**

**from mlxtend.frequent\_patterns import apriori**

**frequent\_itemsets = apriori(df, min\_support=0.1, use\_colnames=True)**

**c. Association Rule Generation: Generate association rules from the frequent itemsets using a library like mlxtend:**

**python**

**Copy code**

**from mlxtend.frequent\_patterns import association\_rules**

**rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.0)**

**2. Testing or Applying the Model:**

**a. New Transaction Data: To test the model, you need new transaction data. This data should represent recent or upcoming transactions that you want to analyze for market basket insights.**

**b. Data Preprocessing: Preprocess the new transaction data in the same way as the training data, ensuring it is in a format that can be used with your chosen library (Pandas DataFrame or list of transactions).**

**c. Apply the Trained Rules: Use the rules generated during training to analyze the new transaction data. Check if the new transactions exhibit any item associations or patterns that were discovered during training.**

**d. Interpret Results: Examine the insights or recommendations derived from the model's application to the new data. For example, identify which items are frequently purchased together, assess the impact of marketing strategies, or make inventory management decisions based on the analysis.**

**The Metrices Used for accuracy test:**

**Support: Support measures the frequency with which a rule occurs in the dataset. It is the proportion of transactions that contain both the antecedent and consequent items. Higher support values indicate that the rule is more common. However, very high support might lead to trivial rules, while very low support may make rules less useful.**

**Confidence: Confidence measures how often the rule is true. It's the conditional probability of the consequent given the antecedent. Higher confidence indicates that the rule is more likely to be true when the antecedent is present. A high-confidence rule suggests a strong association between the items.**

**Lift: Lift is a measure of how much more likely the consequent is to be true compared to random chance, given that the antecedent is true. A lift value greater than 1 indicates that the rule has a positive impact on the likelihood of the consequent occurring, meaning it's a valuable rule. A lift value of 1 suggests no association, and a value less than 1 suggests a negative impact.**

**Leverage: Leverage quantifies the difference between the observed frequency of the rule and what would be expected if the items were statistically independent. It's a measure of the additional occurrence of the rule compared to random chance. A positive leverage indicates that the rule is non-randomly occurring.**

**Conviction: Conviction measures the degree to which the consequent item is dependent on the antecedent. It's calculated as (1 - confidence of not having the consequent) divided by (1 - confidence of having the consequent). High conviction values suggest strong association between the antecedent and consequent.**

**Conclusion and further work:(Phase 2)**

**Project conclusion:**

* **In the phase 2 conclusion,we will summarise the key findings and insights from the advanced technologies.We will reiterate the impact of these technologies on improving the accuracy and robustness of Market Basket Insights.**
* **Future work:We will discuss potential avenues for future work,such a incooperating additional data source exploring deep learning models for market Insights.**