Market Basket Analysis with Python Apriori Algorithm

By: Annet Nasimiyu Chebukati

Table of Contents

Market Basket Analysis

Market Basket Analysis (MBA)

What? A data mining technique that uncovers associations between items by analyzing customer purchasing behavior.

Why? Identifies patterns of items often purchased together, valuable for business strategies like cross-selling, up-selling, and promotional offers.

Project Application: Apply MBA to a dataset from a grocery e-commerce site.

Goal: Understand customer purchasing behavior and provide insights to improve sales and customer satisfaction.

Data Preparation

Dataset: Groceries_dataset.csv

Columns:

- Member_number: ID of the customer who bought an item.
- Date: The date the item was bought.
- itemDescription: Name of the item bought.

Preparation: Imported required Python packages, loaded the dataset, and prepared it for analysis. This involved cleaning the data, handling missing values, and transforming the data into a suitable format for MBA.

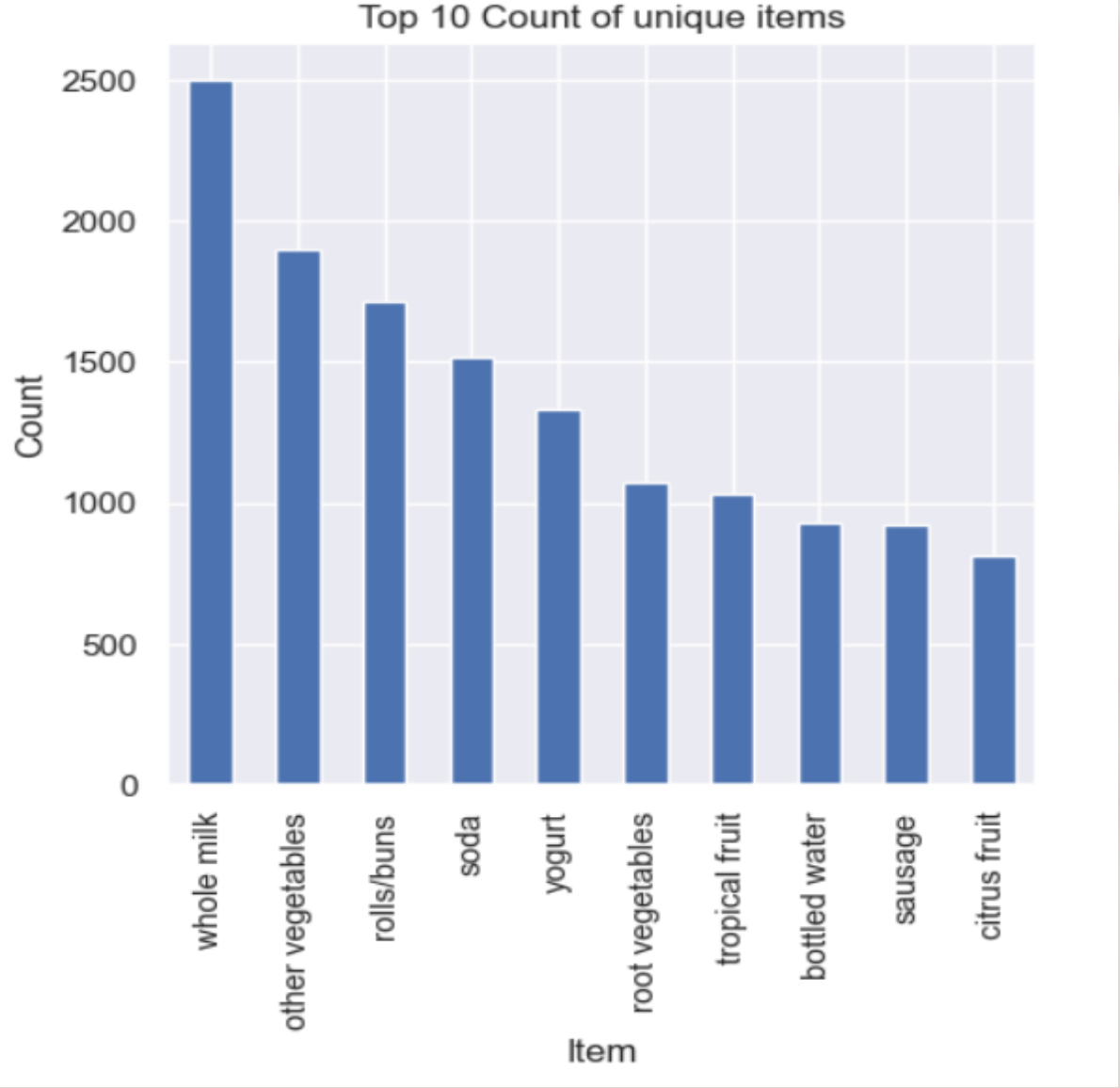
	Column	Count	Null?	Data type
1	Member_number	38765	Non-null	Object
2	Date	38765	Non-null	Datetime64[ns]
3	ItemDescription	38765	Non-null	object

Exploratory Data Analysis

Exploratory Data Analysis

EDA allows us to understand the data by summarizing its main characteristics often with visual methods.

- Understand the patterns and relationships within the data.
- Extract important variables.
- Provide a foundation for further data preparation before modeling.



Exploratory Data Analysis

Creating Transactions

Item: a product purchased from the store. In the groceries_df data frame, an item is represented by the 'itemDescription' column.

Itemset: a collection of one or more items. I aggregated 'itemDescription' into a list for each group of 'Member_number' and 'Date'. This list can be considered as an itemset, representing a collection of items bought in each transaction.

Transaction: an itemset that corresponds to a customer's order. A single shopping event in which a customer purchases one or more items together. Each row in the transactions data frame represents a unique transaction, created by grouping the groceries_df by 'Member_number' and 'Date' and aggregating the 'itemDescription' into a list.

Apriori Algorithm

Apriori Algorithm

Association Rules and Metrics

Association Rule: An association rule is an "if-then" relationship between two itemsets.

Metric: A metric is a measure of the strength of association between two itemsets.

- **Support**: Measures how frequently a group of items occur together as a percentage of your store's transactions.
- **Confidence**: The ratio between transactions that include the combination of items versus transactions that only contain a single item from the set.
- **Lift**: Measures how well your predictions match real-world results. It's the ratio of the observed support to that expected if the antecedent and consequent were independent.
- **Leverage**: Measures the difference between the observed frequency of the antecedent and consequent appearing together and the frequency that would be expected if they were independent.
- **Zhangs metric**: An extension of the Lift metric and is mainly used to measure the disassociation between items.

Apriori Algorithm

One-hot Encoding the Transaction Data

The process of converting categorical data into a format that could be provided to ML algorithms to improve predictions. Converted each categorical value into a new categorical value and assign a binary value of 1 or 0.

- True: Indicates that a particular categorical value is present for a given observation.
- False: Indicates that a particular categorical value is not present for a given observation.

The Apriori Algorithm and Pruning

Apriori algorithm identifies frequent (high support) itemsets using the **Apriori principle**, which states that a set that contains an infrequent item is also infrequent.

Pruning; Process of removing itemsets or association rules, typically based on the application of a metric threshold.

The **mlxtend** module will be used to apply the Apriori algorithm, perform pruning, and compute association rules.

Apriori Algorithm

Applying the Apriori algorithm

- **Step 1: Identify Frequent Itemsets:** Used apriori() to identify frequent itemsets. The min_support parameter was set to define the item frequency threshold for pruning.
- **Step 2: Generate Association Rules:** Generated association rules from the frequent itemsets using the Apriori algorithm. The "support" metric was used to evaluate the significance of the rules, returning only those with a minimum threshold of support equal to 0.
- **Step 3: Evaluate Rule Strength:** Used Zhang's metric to evaluate the strength of association rules. This metric ranges from -1 to 1. Rules indicating a strong positive relationship between items were of interest, so rules with a positive Zhang's metric were filtered (rules = rules[rules['zhangs_metric'] > 0]).
- **Step 4: Filter Rules**: Further filtered the rules using the confidence metric (rules = rules[(rules['confidence'] > 0.13)]).

Visualization (Heatmap)

Interpretation &Insight

Heatmap Interpretation:

Each cell in the heatmap corresponds to a rule, with the antecedent as the column and the consequent as the row.

The value annotated in each cell represents the ‘support’ value of the corresponding rule, measuring how frequently the antecedent and consequent appear together in the dataset.

Insight:

High support values indicate frequently purchased combinations of items. For instance, ‘frankfurter’, ‘ham’, ‘semi-finished bread’, ‘processed cheese’, ‘whole milk, yogurt’, ‘whole milk, sausage’, ‘yogurt, sausage’, ‘detergent’, ‘rolls/buns, yogurt’, ‘packaged fruit/vegetables’, ‘rolls/buns, soda’, ‘rolls/buns, sausage’, ‘soda, sausage’, and ‘seasonal products’ are frequently purchased with their corresponding items.

High confidence values for these rules suggest that when the antecedent item is purchased, the consequent item is also likely to be purchased.

Recommendations

Store Layout: Based on the insights, a possible strategy could be to place frequently purchased items near each other in the store. This could encourage customers to purchase them together, enhancing the shopping experience and potentially increasing sales.

Promotional Strategies: Develop promotional strategies around these popular items. This could include discounts on combined purchases, special offers, or highlighting these items in marketing campaigns. Such strategies could attract more customers and increase sales.

Thank you

annetnasimiyuchebukati@gmail.com

Linkedin

Github

Portfolio

Phone: +254 719406701