Market Basket Analysis with Python Apriori Algorithm

By: Annet Nasimiyu Chebukati

Table of Contents

Market Basket Analysis

- **Market Basket Analysis (MBA)** is a data mining technique used to uncover associations between items by analyzing customer purchasing behavior. It identifies patterns of items often purchased together. These patterns are valuable for various business strategies like cross-selling, up-selling, and promotional offers.
- In this project, I will apply MBA to a dataset from a grocery e-commerce site. The goal is to understand customer purchasing behavior and provide insights to improve sales and customer satisfaction.

Data Preparation

Dataset

The dataset I'll be using is Groceries_dataset.csv, which contains the following columns:

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

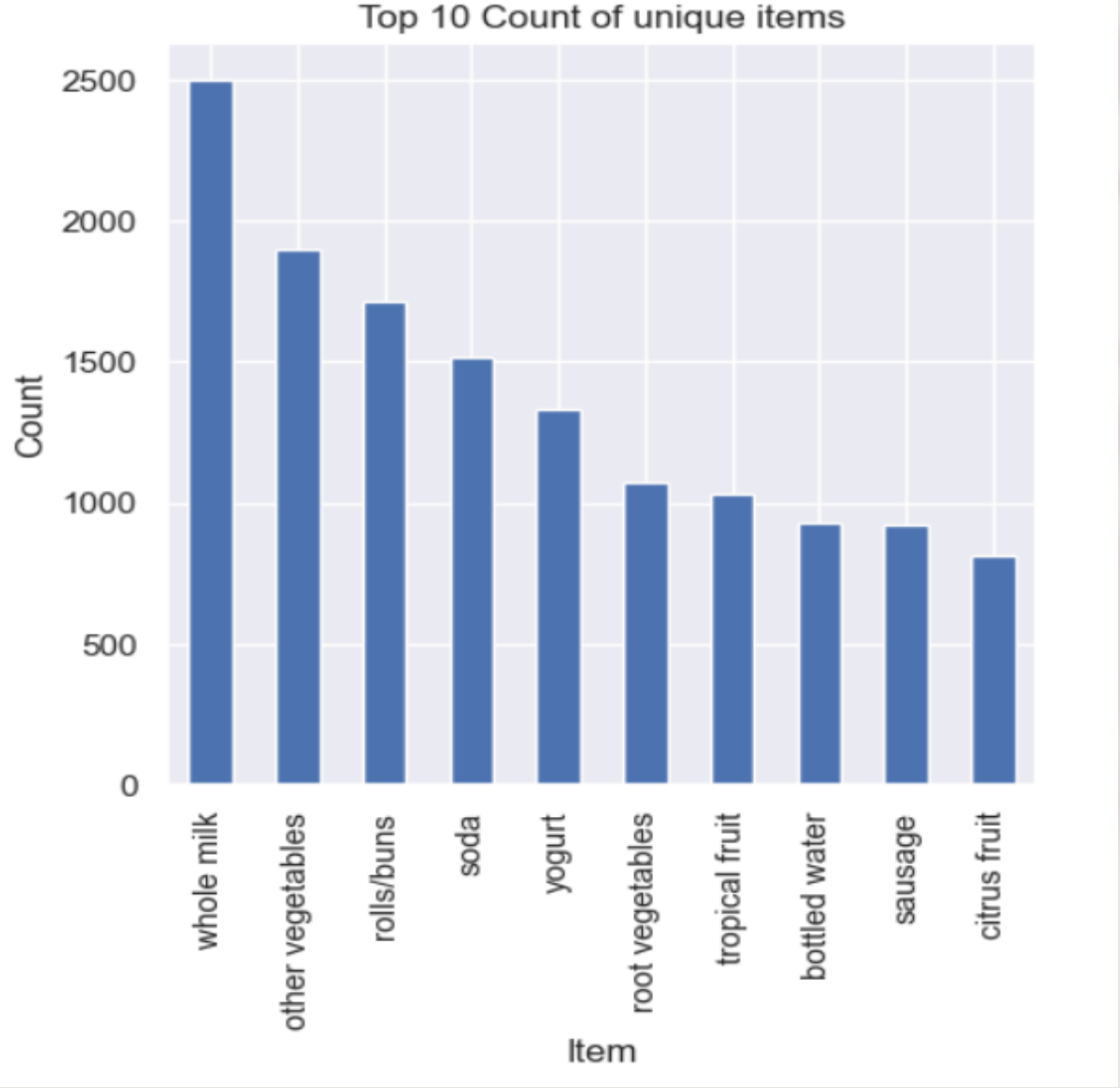
I imported the required python packages and loaded the dataset and prepared it for analysis. Involved cleaning the data, handling missing values, and transforming the data into a suitable format for MBA.

	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

1. Understanding the data by summarizing its main characteristics.



Exploratory Data Analysis

2. Creating Transactions

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

Itemset: An itemset is 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: A transaction refers to an itemset that corresponds to a customer's order. It represents a single shopping event in which a customer purchases one or more items together. In the analysis, each row in the transactions data frame represents a unique transaction, which is created by grouping the groceries_df data frame 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**: This measures how frequently a group of items occur together as a percentage of your store's transactions.
- **Confidence**: This is the ratio between transactions that include the combination of items versus transactions that only contain a single item from the set.
- **Lift**: This 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**: This 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**: The Zhang metric is an extension of the Lift metric and is mainly used to measure the disassociation between items.

Apriori Algorithm

One-hot Encoding the Transaction Data

One-hot encoding is a process of converting categorical data into a format that could be provided to ML algorithms to improve predictions. With one-hot, I 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

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

Pruning is the 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

- Used **apriori()** to identify frequent itemsets. min_support set the item frequency threshold used for pruning.
- Generating **association rules** from the frequent itemsets that were generated using the Apriori algorithm. It uses the "support" metric to evaluate the significance of the rules and only returns rules that have a minimum threshold of support equal to 0.
- Next, I used **Zhang's metric** to evaluate the strength of association rules. Ranges from -1 to 1.I'm interested in rules that indicate a strong positive relationship between items. I filtered for rules with a positive Zhang's metric.(rules = rules[rules['zhangs_metric'] > 0])
- Filtered the rules further with using **confidence metric**. (rules = rules[(rules['confidence'] > 0.13)])

Visualization (Heatmap)

Interpretation &Insight

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. The 'support' value measures how frequently the antecedent and consequent appear together in the dataset.
- By examining these 'support' values, you can easily identify which combinations of items have high support (i.e., are frequently purchased together).

Insight

- The items '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. This is indicated by the high support values for these rules.
- The confidence values for these rules are also relatively high, meaning that when the antecedent item is purchased, the consequent item is also likely to be purchased.

Recommendations

Based on these insights, a possible strategy could be to place these items near each other in the store to encourage customers to purchase them together.

Additionally, promotional strategies could be developed around these popular items to increase sales.

Thank you

annetnasimiyuchebukati@gmail.com

Linkedin

Github

Portfolio

Phone: +254 719406701