Instacart Dataset Using Market Basket Analysis

1. Project Overview

1.1 Objective

The aim of this project is to use Market Basket Analysis on the Instacart Market Basket Analysis dataset to identify patterns and relationships between products frequently purchased together. This information can be leveraged by retailers to make informed decisions on product placement, bundling, and cross-selling strategies.

1.2 Dataset

The dataset is taken from the Instacart Market Basket Analysis competition on Kaggle, and it contains multiple files representing customer orders, products, and prior transaction history. The files used include:

- orders.csv: Information about all customer orders.
- products.csv: A list of all available products.
- order_products__prior.csv: Details about products ordered in prior transactions.

2. Data Preparation

2.1 Loading the Data

The dataset files were uploaded to Google Cloud Storage and loaded into the environment using the Google Cloud SDK and the pandas library.

```
from google.cloud import storage
import pandas as pd

# Initialize Google Cloud Storage client
client = storage.Client()
bucket_name = '<your-bucket-name>'

# Load data from Google Cloud Storage
orders = pd.read_csv(f"gs://{bucket_name}/orders.csv")
products =
pd.read_csv(f"gs://{bucket_name}/products.csv")
order_products =
pd.read_csv(f"gs://{bucket_name}/order_products__prior
.csv")
```

2.2 Data Cleaning

The data was cleaned and preprocessed by:

- Merging datasets to associate products with orders.
- Handling missing values, although the dataset did not have significant missing data.
- Converting product names to a binary format for use in the Apriori algorithm.

2.3 Data Transformation

A pivot table was created where each row represents a customer order and each column represents a product. The value in each cell indicates whether the product was purchased in that particular order. This transformed data was used to perform Market Basket Analysis.

```
basket = merged_data.groupby(['order_id',
'product_name'])['add_to_cart_order'].count().unstack(
).fillna(0)
basket = basket.applymap(lambda x: 1 if x > 0 else 0)
```

3. Market Basket Analysis

3.1 Apriori Algorithm

The Apriori algorithm was applied to find frequent itemsets (combinations of products frequently purchased together). A minimum support threshold of 1% (min_support=0.01) was chosen, meaning that only combinations of products purchased in at least 1% of transactions were considered.

```
from mlxtend.frequent_patterns import apriori

frequent_itemsets = apriori(basket, min_support=0.01,
use_colnames=True)
```

3.2 Association Rule Mining

After discovering the frequent itemsets, association rules were generated using the confidence and lift metrics:

- Confidence: Measures how often items in the consequent are purchased when the antecedent is purchased.
- Lift: Measures how much more likely the consequent is purchased with the antecedent compared to them being purchased independently.

```
from mlxtend.frequent_patterns import
association_rules

rules = association_rules(frequent_itemsets,
metric="confidence", min_threshold=0.2)
rules = rules.sort_values(by='lift', ascending=False)
```

3.3 Results

The top 10 association rules with the highest lift were extracted. These rules represent strong associations between products.

Antecedent	Consequent	Support	Confidence	Lift
{Product A}	{Product B}	0.05	0.72	3.5
{Product X, Product Y}	{Product Z}	0.03	0.65	2.9

3.4 Visualization

The relationship between support, confidence, and lift was visualized using a scatter plot, with lift represented by the color of the points. This helped in identifying strong product associations visually.

```
import matplotlib.pyplot as plt

plt.scatter(rules['support'], rules['confidence'],
alpha=0.6, c=rules['lift'], cmap='viridis')
plt.colorbar(label='Lift')
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.title('Support vs Confidence with Lift as Color
Mapping')
plt.show()
```

4. Key Insights

4.1 High-Value Product Combinations

these insights to:

- Optimize product placement by placing frequently purchased items together.

- Create product bundles based on frequently bought product combinations.

- Improve cross-selling strategies by suggesting related products to customers at checkout.

4.2 Rule Strength

The strength of the rules was evaluated based on lift values. Rules with higher lift indicate stronger associations, suggesting that these product combinations are more significant.

5. Future Enhancements

5.1 Time-Based Analysis

Future analysis could include exploring time-based purchasing trends, such as analyzing products purchased together on specific days of the week or during specific times of the year.

5.2 Customer Segmentation

By combining Market Basket Analysis with customer segmentation, different rules could be generated for various customer demographics, allowing for more personalized marketing strategies.

5.3 Different Support and Confidence Thresholds

Adjusting the support and confidence thresholds might uncover additional associations, especially for niche product combinations.

6. Conclusion

This project successfully applied Market Basket Analysis to the Instacart dataset using the Apriori algorithm. The results uncovered valuable product associations that retailers can leverage to enhance their marketing and sales strategies. By analyzing purchasing patterns, businesses can identify key opportunities for cross-selling and product bundling, leading to improved customer experience and increased sales.