**ASSIGNMENT XI**

**Project Title:**

Product Recommendation System using Machine Learning

**Team Leader:**

- Ejumalla Saikiran

**Team members:**

1. J. Sinduja

2. A. Bhavya Sri

3. S. Lakshanya

4. Naveen Rampa

**Submission Date:**

**07/04/2024**

**Methodology**

1. **DataSource**

The dataset is acquired from LMS platform.

Each row of data represents a transaction for a particular item and the attributes correspond to the following:

InvoiceNo : Unique identifier for transaction

StockCode : Unique identifier for the stock item being purchased

Description : Description of item

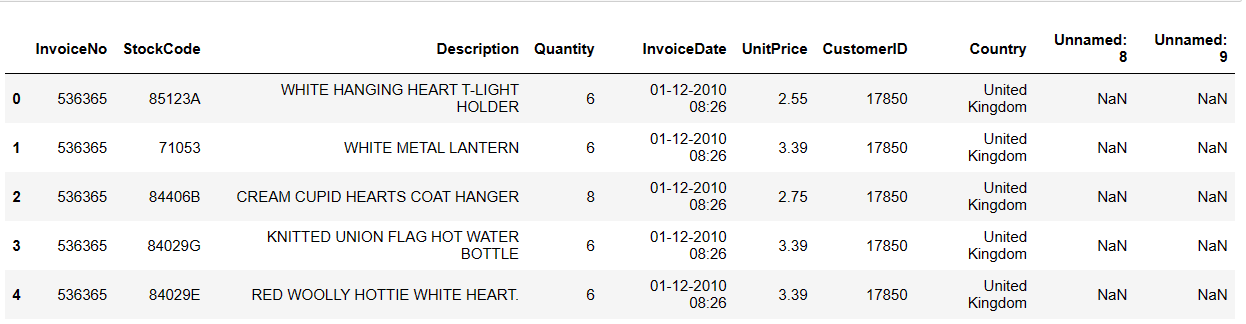
Quantity : Number of units purchased

InvoiceDate : Date of purchase

UnitPrice : Cost of one unit of the item

CustomerID : Unique Identifier for customer

Country : Country of transaction



The occurrence of "Unnamed: 8" and "Unnamed: 9" columns resulted from errors in the data. These issues will be addressed during the exploratory data analysis (EDA) and preprocessing stages.

**Shape of the data**

****

The dataset comprises 541,909 rows and 10 columns in total.

1. **Data Pre-processing**

**2.1 Examining and Removal of Extra Columns:**

The column named ‘Unnamed: 8’ and ‘Unnamed: 9’ are not supposed to be present in the dataset. These additional columns seem to have arisen from instances where the value of one column has spilled over into adjacent columns.



We have observed a pattern that which ever rows were identified as “spilled” have numeric values in Country column. Before handling those rows, we opted to drop both the columns ‘Unnamed: 8’ and ‘Unnamed: 9’ which resulted in the following dataset :



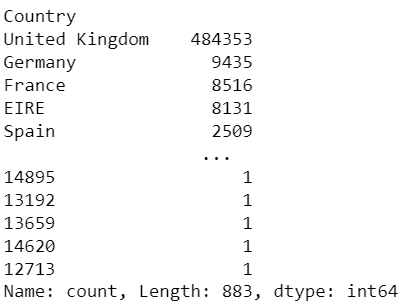
**2.2 Handling Duplicates:**

Duplicates were identified within the dataset and subsequently removed to ensure data integrity and accuracy in subsequent analyses. By eliminating duplicates, the dataset was streamlined and optimized for various data exploration and modeling tasks.

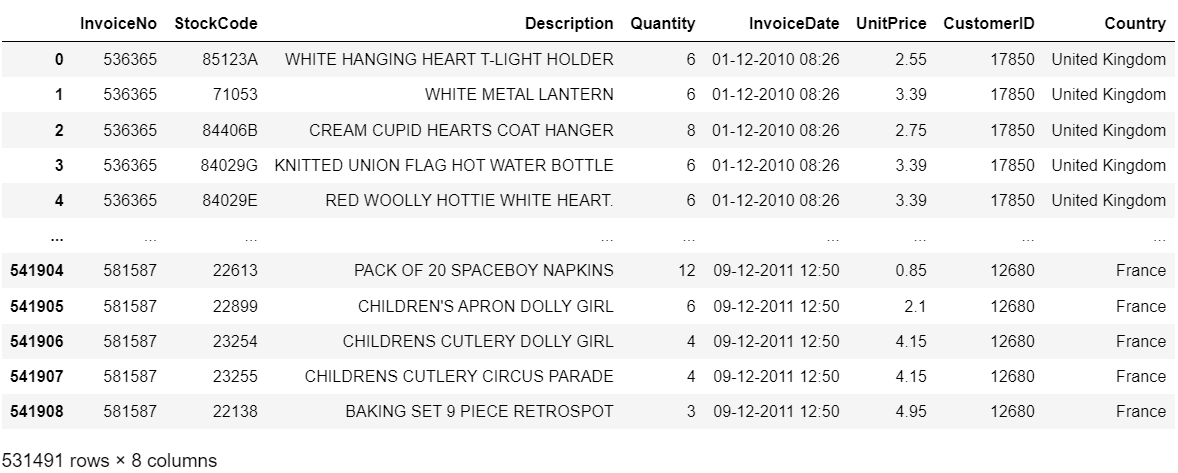


**2.3 Data Cleaning using 'Country' Column:**

As stated before, it is apparent that the 'Country' column contained values other than country names, including numeric values.

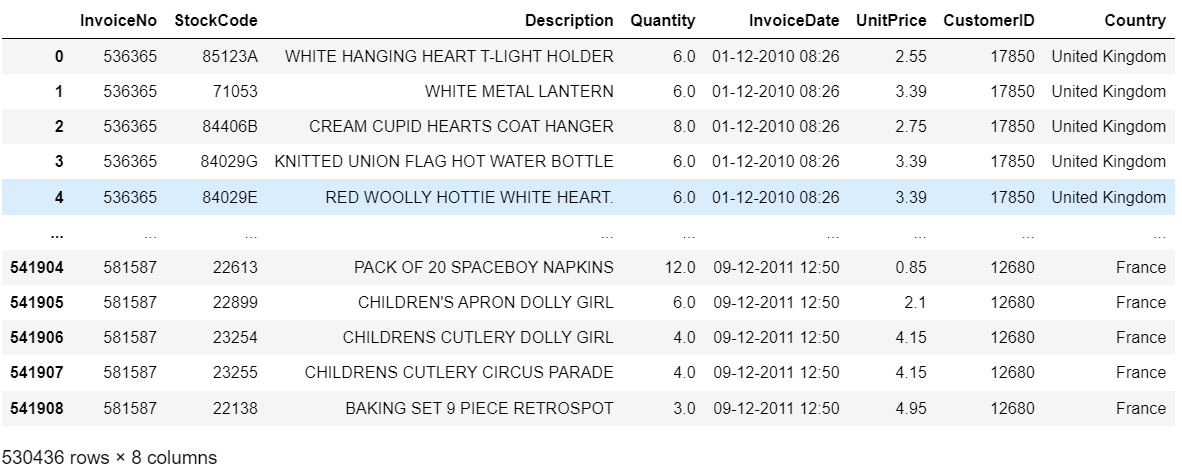


This indicates that rows may have been shifted or misaligned. Recognizing this pattern, we leveraged the presence of numeric values in the 'Country' column as a heuristic to identify and remove error rows from the dataset. By filtering out rows containing numeric values in the 'Country' column, we effectively eliminated the shifted rows and improved the overall quality and integrity of the dataset.



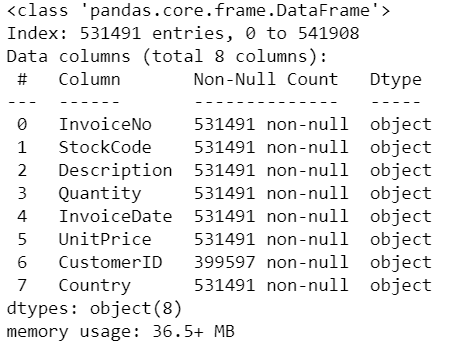
**2.4 Handling Missing Values:**

Upon further inspection, it was discovered that the dataset contained missing values in three columns: 'Description', 'Country', and 'Quantity'.By eliminating rows with missing values in critical columns such as 'Description' and 'Country', data integrity and consistency were maintained.

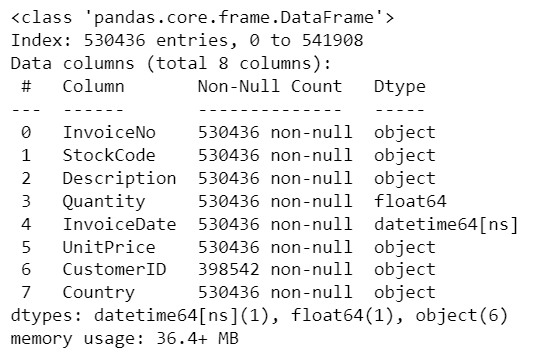


**2.5 Correcting Data Types:**

Upon examination of the dataset, it was observed that the 'InvoiceDate' and 'Quantity' columns were incorrectly assigned the data type 'object'.



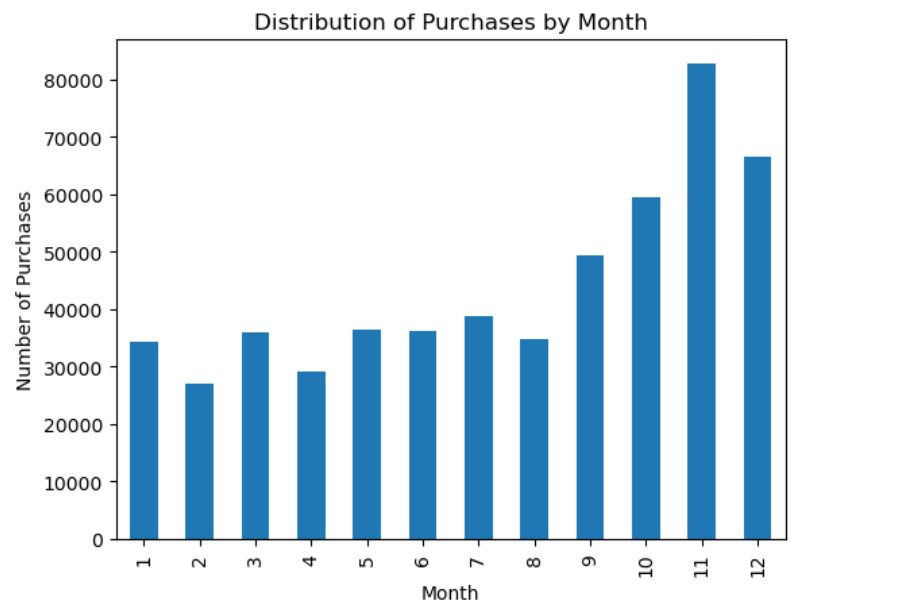
This data type is not suitable for numerical or datetime values, which could lead to errors in subsequent analyses or modeling tasks. To rectify this issue, the data types of these columns were converted to appropriate types.



1. **Exploratory Data Analysis**

**3.1 Distribution of Purchases by Month:**

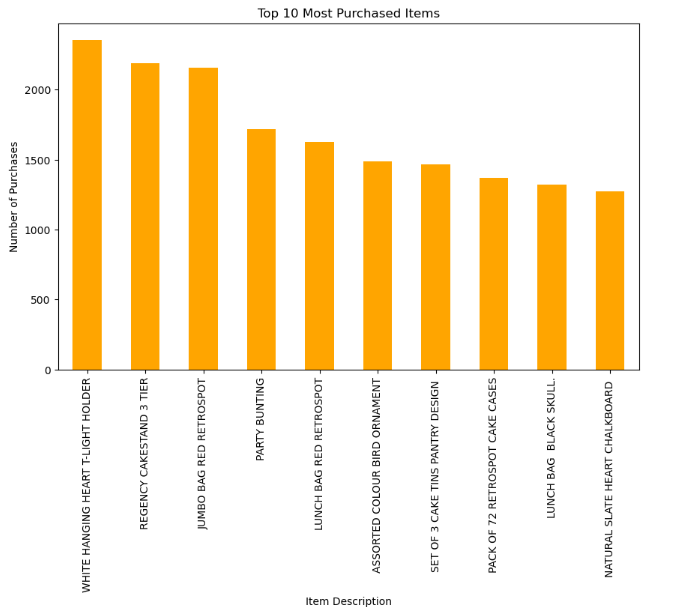
The bar plot illustrates the frequency of purchases made across different months, providing a visual representation of the variation in sales volume over time.



Observation : It was observed that the highest number of purchases occurred during November, followed closely by December. This observation suggests a potential surge in sales activity during the holiday season.

**3.2 Top 10 Most Purchased Items:**

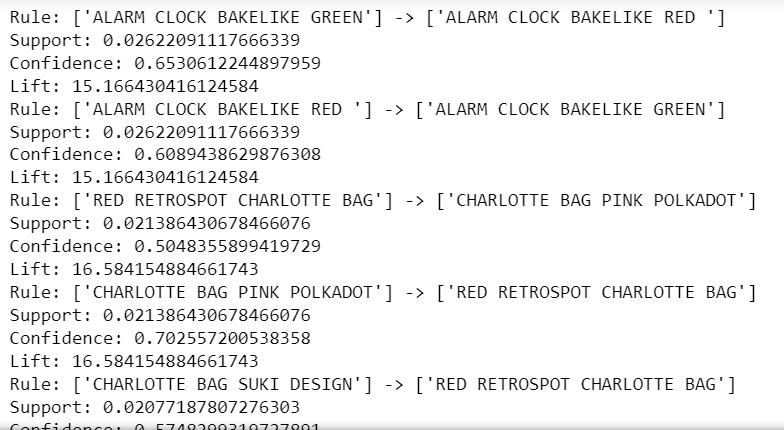
The bar plot below illustrates the top 10 most purchased items within the dataset, providing valuable insights into consumer preferences and product popularity.



1. **Evaluation of Recommendation Algorithms**

**4.1 Association Rule Mining:**

Association rule mining is a technique used to discover interesting relationships or patterns hidden in large datasets. In the context of recommendation systems, it involves finding associations between items based on their co-occurrence in transactions. For example these were some of the rules generated :



**Reasons for Usage:** Association rule mining was initially considered due to its simplicity and interpretability. It can uncover implicit relationships between items and generate recommendations based on these associations.

**Challenges and Decision to Drop:** However, due to the large size of the dataset, setting appropriate thresholds for minimum support and minimum confidence resulted in a limited number of rules. Even with considerable rules, the generated recommendations lacked relevance and were often impractical. Additionally, association rule mining struggled to provide recommendations for certain products, making it less suitable for our purpose.

**Test Case :** GREEN REGENCY TEACUP AND SAUCER



Observation : It can be seen that for this test case exactly 3 products were recommended and their relevance is also quite good.

1. **Test Case :** ALARM CLOCK BAKELIKE GREEN



Observation : In this test case the model was only able to predict one recommended product which is insufficient.

**4.2 : k-Nearest Neighbors (kNN) Model :**

The kNN algorithm is a simple and intuitive method used for both classification and recommendation tasks. In the context of recommendation systems, it identifies similar users or items based on past interactions and makes predictions or recommendations based on the interactions of nearest neighbors.

**Reasons for Usage:** kNN was chosen for its simplicity and flexibility. It can capture complex relationships between users and items and is relatively easy to implement.

**Challenges and Decision to Drop:** One of the main challenges with the kNN model was the need to create a trainset, which involved selecting a subset of the dataset for training purposes. This led to issues when attempting to make recommendations for products outside the trainset. Furthermore, the model's predictions were inconsistent, often providing random recommendations for certain products, which raised concerns about its accuracy and reliability.

**1.Test Case :** WHITE METAL LANTERN



Observation : For the above test case we got 3 predictions and they were quite relevant too.

1. **Test Case :** BLACK TEA,COFFEE, SUGAR JARS

IMG_256

Observation : The above test case’s product is not in the train set due to which the model could not predict a product which is a drawback.

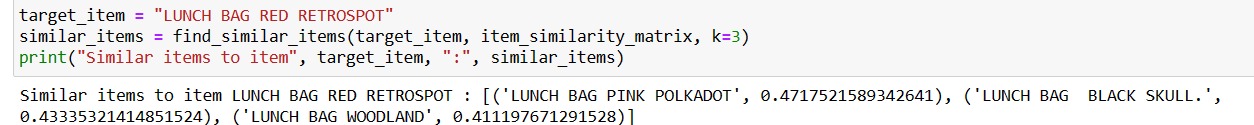
**4.3 Item-Item Collaborative Filtering :**

In item-based collaborative filtering, the algorithm operates on an item-item similarity matrix. Rows and columns represent items, and cells contain similarity scores between pairs of items. This matrix is derived from user-item interaction data but is focused on item-item relationships rather than user-item interactions.

**Reasons for Usage**: Item-item collaborative filtering emerged as the most promising approach due to its ability to provide accurate and relevant recommendations for every product in the dataset. It offered consistent performance and required less computational resources compared to association rule mining and kNN.

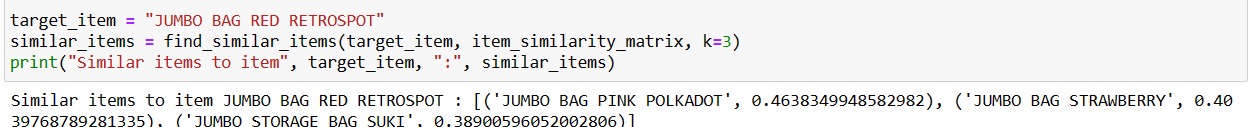
**Decision to Adopt:** Based on its effectiveness and efficiency, item-item collaborative filtering was selected as the preferred recommendation algorithm for our system. It provided valuable insights and reliable recommendations, making it well-suited for our recommendation task.

1. Test Case : LUNCH BAG RED RETROSPOT



Observations : The recommendations for the above product are very accurate, which shows us that this model is far better than other models.

1. Test Case : JUMBO BAG RED RETROSPOT



Observations : The recommendations for the above product is also very accurate and close to reality.

1. **Final Implementation**

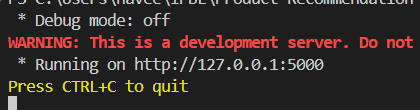
For the final implementation, a web application is developed using Flask showcasing the implementation of a product recommendation system. The recommendation system is based on Item-Item collaborative filtering.

The implementation involves generating two key matrices: the item-user matrix and the item similarity matrix. These matrices are computed within a Jupyter notebook environment. Once computed, the results are saved in a file.

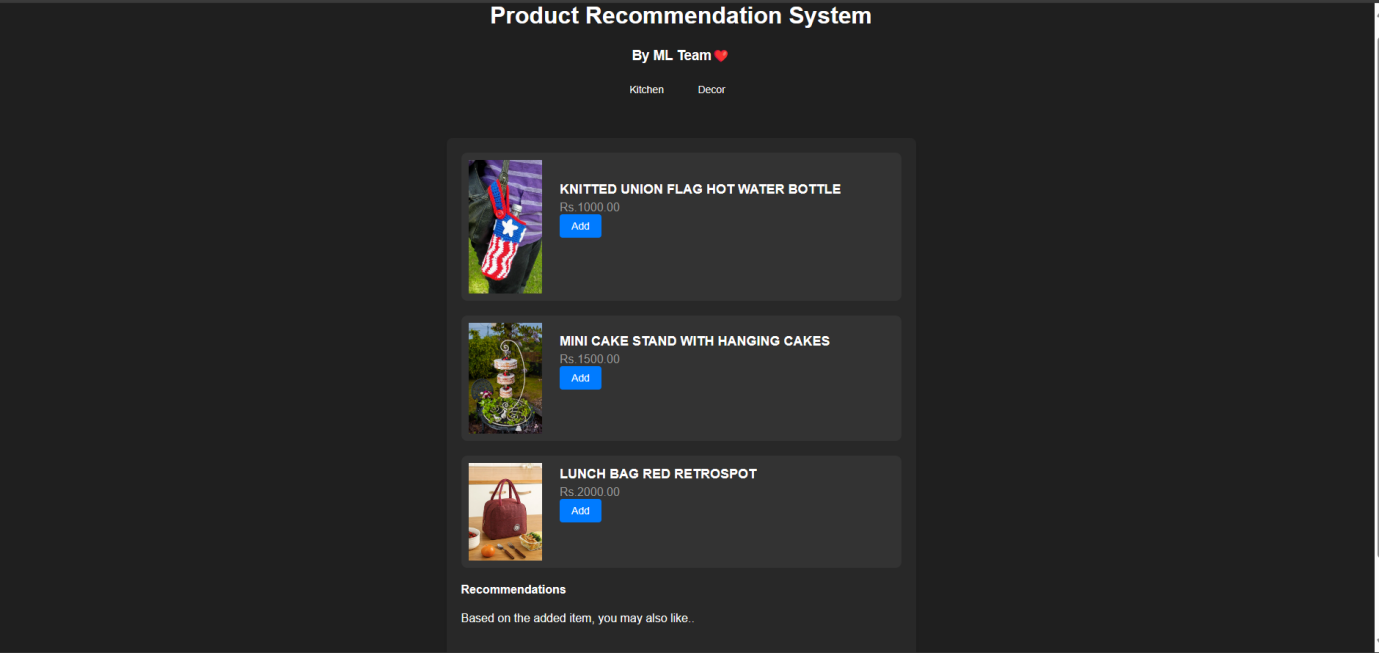
In the Flask application, these precomputed matrices are utilized for providing recommendations to users based on their interactions with items. The Flask app serves as an interface for users to interact with the recommendation system, where they can receive personalized product recommendations based on their past behavior or preferences.

The demonstration of the website implementation is as follows:

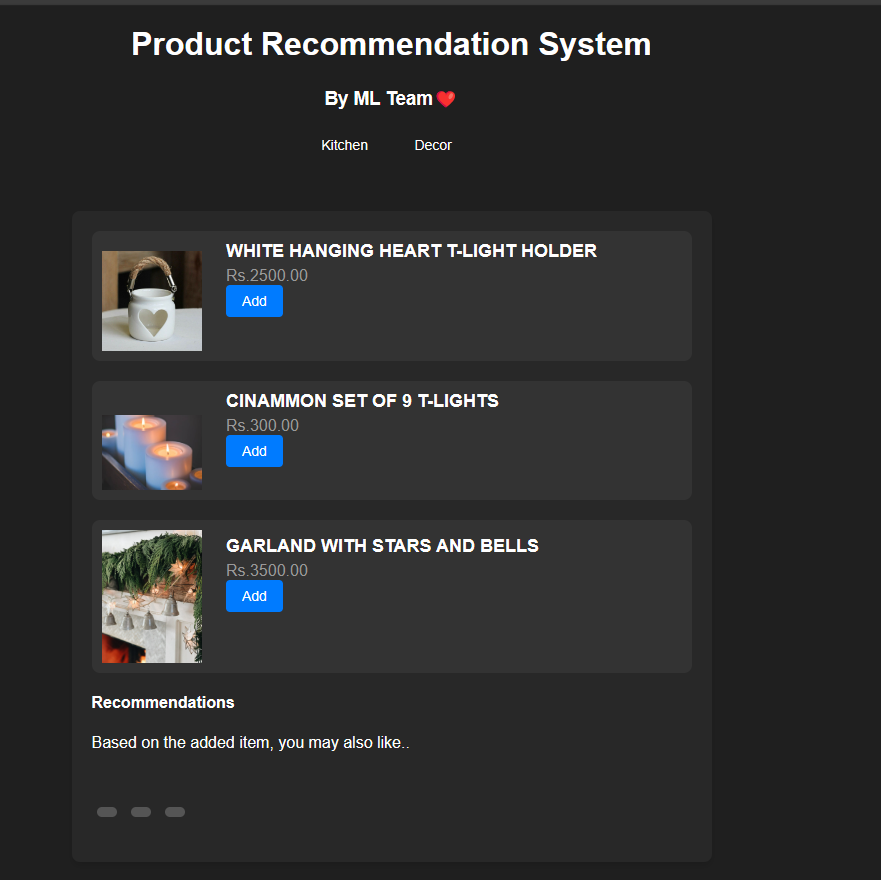
1. Launch the web app by executing the Python code, which will deploy the application on localhost.



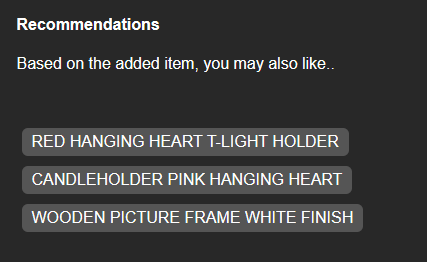
1. Open the deployed web app in a browser. If the deployment is successful the following webpage is rendered.



1. Users can select either the Kitchen or Decor category to view available products. By default, the Kitchen category is selected. The Decor view is displayed below.



1. When the "Add" button is clicked, recommendations based on the added item will be generated and displayed below.



Above are the results obtained when clicked on add button of WHITE HANGING HEART T-LIGHT HOLDER.

**Conclusion**:

Through the exploration of various recommendation algorithms, including association rule mining, k-nearest neighbors (kNN), and item-item collaborative filtering, we gained valuable insights into their strengths, weaknesses, and applicability to our dataset. Despite the initial appeal of association rule mining for its simplicity and interpretability, it proved to be impractical due to the large dataset size and challenges in generating relevant recommendations.

Similarly, while kNN offered flexibility and the ability to capture complex relationships between users and items, its inconsistent predictions and reliance on a predefined trainset limited its effectiveness for our recommendation task.

In contrast, item-item collaborative filtering emerged as the most promising approach, providing accurate and relevant recommendations for every product in the dataset. Its ability to leverage similarities between items and offer consistent performance made it well-suited for our recommendation system.