Group Tito	
Lucas McKenzie	251201874
Oluwatitomi Adeleke	251206286
Eldar Zulic	251230378
Khalid Zabalawi	251247169

Project Overview

Project name

AI-Based Product Recommendation System

Project Report: Collaborative Filtering Recommendation System

1. Introduction

The implemented artificial intelligence (AI) in the product recommendation system leverages collaborative filtering, a popular technique in recommender systems. Collaborative filtering relies on the idea that users who have exhibited similar behaviors or preferences in the past are likely to have similar preferences in the future. In this context, the system analyzes the historical purchase data of users, identifying patterns and similarities among their product choices. Through the use of cosine similarity, the system measures the similarity between users based on their purchase history, creating a user-product matrix. This matrix serves as the foundation for generating personalized recommendations. The AI-driven system then identifies users with similar preferences and suggests products that have been positively received by these analogous users. The collaborative filtering approach enables the system to make insightful and context-aware recommendations, enhancing the overall user experience.

The collaborative filtering approach is implemented through a series of steps, beginning with the creation of a mapping DataFrame to correlate product categories and subcategories between different datasets. The system then merges the relevant datasets, encoding categorical columns using a label encoder. The resulting user-product matrix captures the interactions between users and products. To enhance the accuracy of recommendations, cosine similarity is calculated between users, enabling the system to identify those with similar preferences. When a user requests recommendations, the system identifies analogous users, analyzes their unrated products, and suggests items that align with the user's preferences. The utilization of collaborative filtering, along with AI techniques, ensures a dynamic and personalized recommendation system that adapts to users' evolving preferences based on historical interactions, creating a seamless and user-centric shopping experience

2. System Design

2.1 Data Loading and Preprocessing

The project commenced with the loading of two datasets, namely grocery_sells.csv and Online_Retail_Categorized.csv. A judiciously chosen subset of the data was utilized for testing purposes. A critical element in this phase was the creation of a mapping DataFrame (customer_mapping) that establishes connections between customers and specific product categories and subcategories.

2.2 Data Merging and Encoding

The amalgamation of grocery sells data with the customer mapping DataFrame yielded merged_data. This amalgamation extended further, entwining with online retail data to produce online_retail_with_names. Employing the scikit-learn LabelEncoder played a central role in transforming categorical columns ('Customer Name' and 'Description') into numerical entities.

2.3 User-Product Matrix and Similarity Calculation

The construction of the user-product matrix (user_product_matrix) using pandas' pivot_table function marked a pivotal step. This matrix, representing customer product quantities, served as the basis for computing cosine similarity between users—a metric indicating collaborative resonance.

2.4 Recommendation Logic

Central to the system's architecture is the get_recommendations function. This function orchestrates the generation of product suggestions by identifying similar users, exploring uncharted products, and presenting a personalized list of recommendations based on user preferences.

3. Challenges Faced

The project encountered several challenges:

3.1 Data Integration

Integrating grocery sells data with online retail necessitated meticulous mapping and merging due to disparities in column names and structures.

3.2 Limited Rows and Readability

The sea of data, vast and overwhelming, posed a challenge to readability. To navigate this challenge, we introduced a constraint on the number of rows considered:

```
# Set the maximum number of rows to be considered
head_size = 2000
grocery_sells = grocery_sells.head(head_size)
online_retail = online_retail.head(head_size)
```

3.3 User Identification

Identifying users and ensuring their presence in both datasets presented a challenge, which the system adeptly manages.

4. Solutions Implemented

4.1 Data Cleaning and Standardization

To overcome integration challenges, data cleaning techniques were employed to standardize column names and structures, facilitating a seamless merge.

4.2 Robust User Identification

In response to elusive users, the system implements robust user validation, prompting users to provide a valid customer name for a resilient experience.

5. Actual Usage

The system facilitates an interactive experience by prompting users for a customer name, leading to personalized recommendations based on collaborative filtering.

6. Reference Outputs

The system outputs a curated list of recommended products for the specified user—a manifestation of collaborative wisdom that attests to the system's understanding of shared preferences.

7. Conclusion

Collaborative filtering, a potent technique in recommendation systems, found expression in our Python implementation. The system seamlessly integrated disparate datasets, navigating challenges with finesse. Users are guided toward accurate and personalized recommendations.

8. Future Enhancements

Future iterations may involve delving into advanced collaborative filtering techniques, refining the handling of sparse datasets, and incorporating user feedback for continuous improvement.

9. Output Demonstration

As our exploration progresses, we now provide a substantive demonstration through an augmented code excerpt, affording insight into the advancing functionalities of our collaborative filtering recommendation system.

```
-------------------------------------------------------------------------------------

Welcome to the collaborative filtering AI-based product recommendation system!

Here is the possible unique customerIDs (or choices!):
[19, 3, 4, 37, 0, 12, 32, 44, 26, 11, 8, 14, 29, 34, 47, 41, 10, 16, 5, 18, 23, 43, 30, 49, 13, 45, 42,
 20, 27, 22, 46, 15, 7, 25, 21, 48, 40, 38, 1, 36, 39, 28, 33, 31, 24, 2, 9, 35, 17, 6]

Please enter the customerID you want recommendations for from the list above (e.g: 19):
```

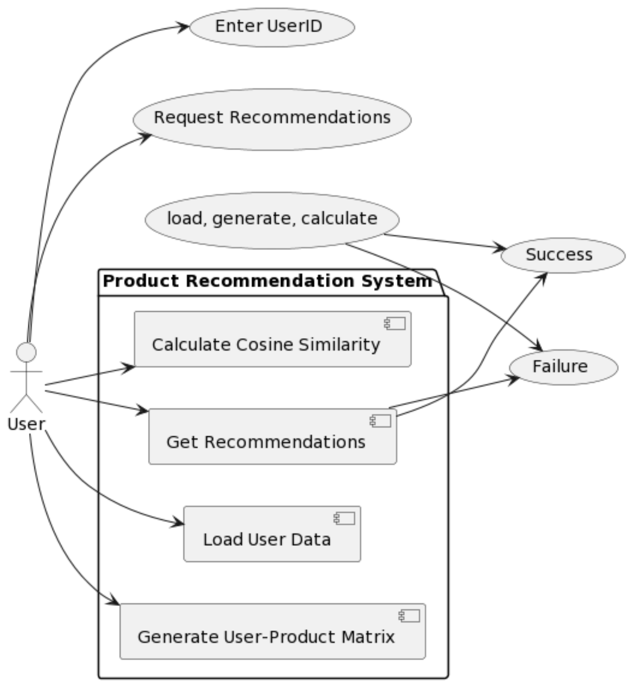
Outputs

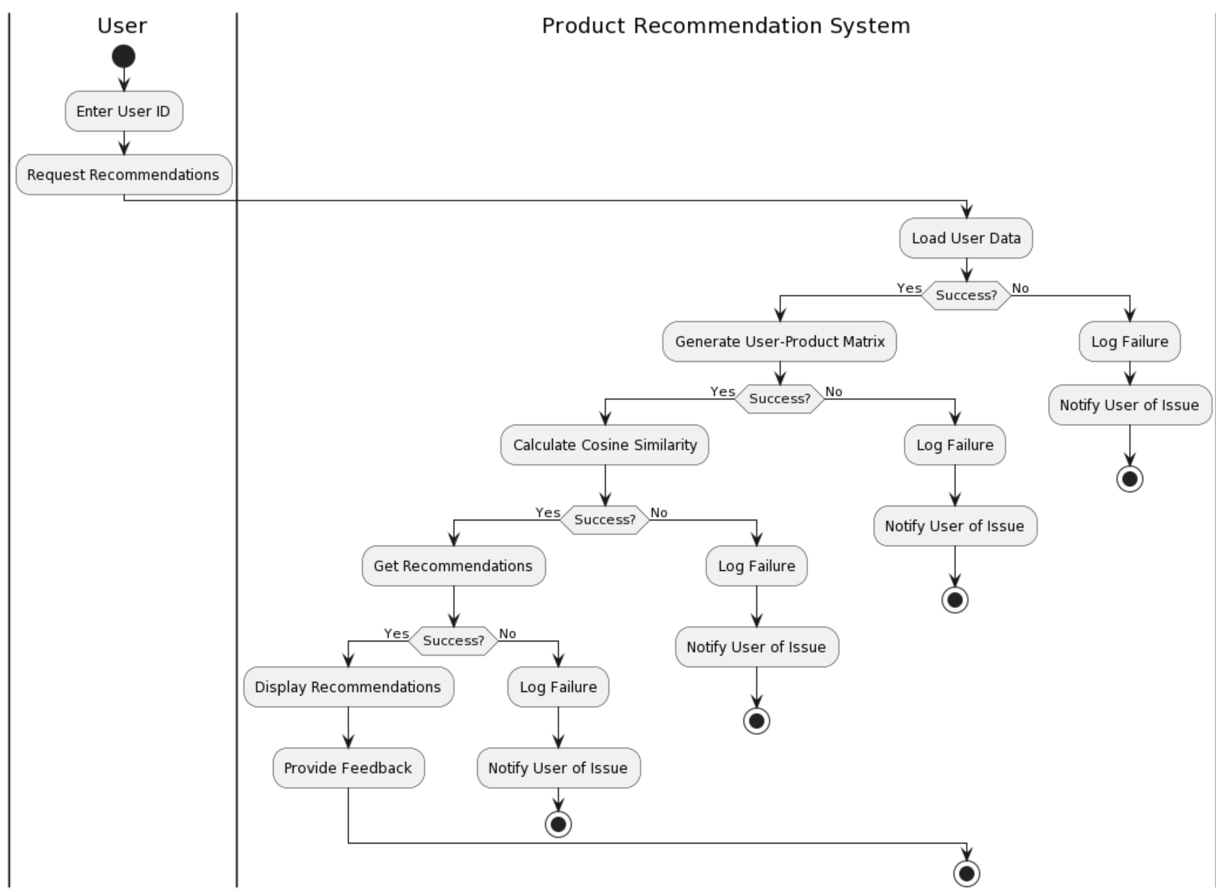
```
-------------------------------------------------------------------------------------

Welcome to the collaborative filtering AI-based product recommendation system!

Here is the possible unique customerIDs (or choices!):
[19, 3, 4, 37, 0, 12, 32, 44, 26, 11, 8, 14, 29, 34, 47, 41, 10, 16, 5, 18, 23, 43, 30, 49, 13, 45, 42,
 20, 27, 22, 46, 15, 7, 25, 21, 48, 40, 38, 1, 36, 39, 28, 33, 31, 24, 2, 9, 35, 17, 6]

Please enter the customerID you want recommendations for from the list above (e.g: 19): 19
Please enter the number of product recommendations you want (e.g: 3): 3

Recommended products for user 19:
SILVER FISHING GNOME
WHITE AND BLUE CERAMIC OIL BURNER
CAKESTAND, 3 TIER, LOVEHEART
```

```
-------------------------------------------------------------------------------------
Here is the possible unique customerIDs (or choices!):
[19, 3, 4, 37, 0, 12, 32, 44, 26, 11, 8, 14, 29, 34, 47, 41, 10, 16, 5, 18, 23, 43, 30, 49, 13, 45, 42,
 20, 27, 22, 46, 15, 7, 25, 21, 48, 40, 38, 1, 36, 39, 28, 33, 31, 24, 2, 9, 35, 17, 6]

Please enter the customerID you want recommendations for from the list above (e.g: 19): 44
Please enter the number of product recommendations you want (e.g: 3): 5

Recommended products for user 44:
CAKE STAND LOVEBIRD 2 TIER WHITE
WHITE AND BLUE CERAMIC OIL BURNER
SILVER FISHING GNOME
FUSCHIA VOILE POINTY SHOE DEC
CAKESTAND, 3 TIER, LOVEHEART
```

This enhancement unfolds a dynamic user interaction. Users now have the freedom to choose their customerID and specify the number of product recommendations, fostering a more engaging and user-friendly experience.

Diagrams


Enter UserID
Request Recommendations
load, generate, calculate
Success
Failure
Product Recommendation System
Calculate Cosine Similarity
Get Recommendations
Load User Data
Generate User-Product Matrix
User


User
Product Recommendation System
Enter User ID
Request Recommendations
Load User Data
Success?
Yes
No
Generate User-Product Matrix
Log Failure
Notify User of Issue
Calculate Cosine Similarity
Get Recommendations
Display Recommendations
Provide Feedback