Optimizing Product Recommendations Using a Hybrid Filtering System for Amazon Electronics Dataset

Prince Osei, Annapia Borraccino, Ravi Tripathi

Problem Statement

- **Source**: Amazon Electronics products dataset
- **Link** :https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset/data?select=All+Electronics.csv
- **Goal**: Build a scalable hybrid recommendation system for e-commerce using the Electronics section of the Amazon 2023 dataset.
- **Approach:** Combine content-based and collaborative filtering techniques.
- **Objective:** Address cold-start issues and enhance recommendation diversity.

Dataset Overview

Dataset Description:

- **Rows**: 9,600
- **Columns**: 9
  - name: Product name (8,800 unique values)
  - main_category: Main product category (1 unique value)
  - sub_category: Subcategory of the product (1 unique value)
  - ratings: Product ratings (39 unique values, some values like 'Get' treated as 0.0)
  - no_of_ratings: Number of ratings (3,455 unique values)
  - discount_price: Discounted price (1,608 unique values)
  - actual_price: Actual price (1,068 unique values)

Missing Values:

- ratings: 95 missing values
- no_of_ratings: 95 missing values
- discount_price: 484 missing values
- actual_price: 70 missing values

Exploratory Data Analysis

Data Cleaning Steps:

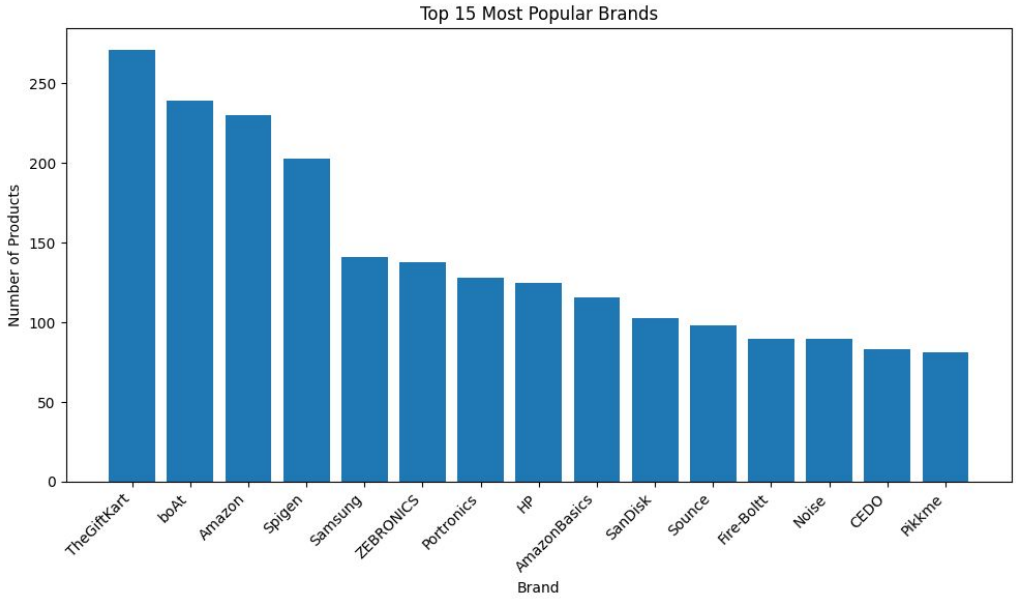
- **Price Columns**: Removed '₹' and commas from discount_price and actual_price and converted them to float.
- **Ratings Column**: Removed the word "Get" from ratings, converting it to float (with 'Get' treated as 0.0).
- **Remove** stopwords and **lowercasing**.

New Columns Created:

- **Manufacturer**: Extracted the manufacturer name from the first word of each entry in the name column and added it as a new column.
- **Discount Price**: Calculated the difference between actual_price and discounted_price.
- **Discount Percentage**: Created a new column representing the discount percentage

Visualizations and results

manufacturer	ratings
ESR	4.422034
Logitech	4.371930
ProElite	4.338095
SanDisk	4.337864
TheGiftKart	4.323985
Spigen	4.309852
TP-Link	4.283607
Robustrion	4.248052
Fire-Boltt	4.166667
Samsung	4.164539
AmazonBasics	4.156034
Amozo	4.153704
OpenTech®	4.132787
Lenovo	4.124590
Pikkme	4.101235



Text Embeddings using TF-IDF

TF-IDF is used to extract features from unstructured text data.

Term Frequency (TF): Measures how frequently a word appears in a document.

Inverse Document Frequency (IDF): Measures how important a word is by considering how common or rare it is across all documents.

TF-IDF Score: The product of TF and IDF values, giving more weight to unique and informative words.

Text Embeddings using BERT

Tokenization: Splitting texts into chunks

Positional Embeddings: Adds positional encoding to the tokenized texts

Attention Mechanism: Uses the self-attention layer to generate a contextualized embedding of the tokens (input text)

Content-Based Filtering

Feature Extraction: Features from the item description, metadata, or other content are extracted using techniques like TF-IDF, word embeddings (like BERT).

Resultant Vector/Matrix: In the case of BERT, we get a matrix that has the contextualized embeddings.

Similarity Calculation: Compute similarity scores between the user's profile (or an item the user has interacted with) and other items. Common similarity measures include cosine similarity, Euclidean distance, or Pearson correlation.

Generate Recommendation: Rank the items based on their similarity scores and recommend those with the highest scores.

Collaborative Filtering

- identifies patterns in user-item interactions to recommend items
- **challenge:** the dataset lacks explicit user data, so the manufacturers are used as proxies for user preferences

MAIN STEPS:

1. **Item-Item Matrix:** a pivot table maps products to manufacturers with ratings as values
2. **Cosine similarity:** measure product relationships, creating a collaborative similarity matrix

Hybrid System

- combination of content-based and collaborative filtering to generate **diverse product suggestions**
- calculates **similarity scores** from both approaches — content-based using item description, and collaborative filtering using interaction patterns such as ratings

Evaluation

The system ranks and returns the **top 5 suggestions** for a selected product while evaluating their average similarity and diversity to ensure meaningful results.

Evaluating the hybrid system using TF-IDF

```
Hybrid Recommendations for 'Apple 20W USB-C Power Adapter (for iPhone, iPad & AirPods)':
1. Apple 30W USB-C Power Adapter
2. Apple 5W USB Power Adapter (for iPhone)
3. Apple MagSafe Charger (for iPhone, AirPods Pro, AirPods with Wireless Charging Case)
4. Apple AirPods (3rd Generation) with Lightning Charging Case
5. Apple AirPods Pro (2nd Generation)

Average Similarity of Recommendations: 0.8707
Diversity of Recommendations: 0.2382
```

Evaluating the hybrid system using BERT

```
Hybrid Recommendations for 'Apple 20W USB-C Power Adapter (for iPhone, iPad & AirPods)':
1. Apple 5W USB Power Adapter (for iPhone)
2. Apple MagSafe Charger (for iPhone, AirPods Pro, AirPods with Wireless Charging Case)
3. Apple AirPods (3rd Generation) with Lightning Charging Case
4. Apple 30W USB-C Power Adapter
5. Apple iPhone 13 (128GB) - Starlight

Average Similarity of Recommendations: 0.9798
Diversity of Recommendations: 0.0288
```

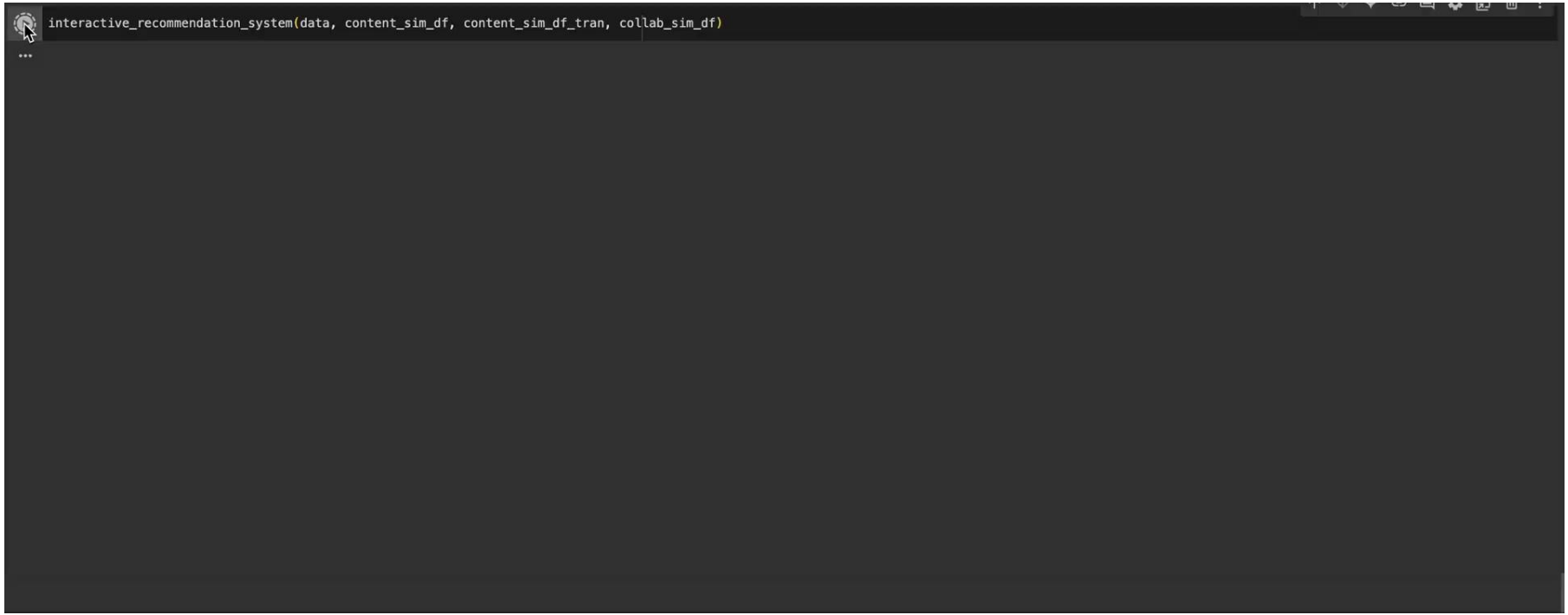
Introducing diversification in combination with TF-IDF results

- adding diversification to recommendations ensuring that suggested products are not only relevant but also different
- using TF-IDF based content similarity: it balances **average similarity**, which measures alignment with the selected product, and **diversity,** which ensures distinct suggestions

```
Diverse Recommendations for 'Apple Lightning to USB Cable (2m):
1. Apple 30W USB-C Power Adapter
2. Apple Airpods (3rd Generation) with Lightning Charging Case
3. Apple Lightning to USB Camera Adapter, USB 3.0 OTG Cable for iPhone/iPad to Connect...
4. Apple iPhone 13 (256GB) - Midnight
5. Apple EarPods with 3.5mm Headphone Plug

Average Similarity of Recommendations: 0.7092
Diversity of Recommendations: 0.3064
```

Simple Demo


interactive_recommendation_system(data, content_sim_df, content_sim_df_tran, collab_sim_df)

Future Work

- improve the system to get higher diversity score
    - variety of products
- get diverse manufacturers to show up
- refine features and pre-processing methods
- try alternative embedding models like **RoBERTa**, or domain-specific transformers for more nuanced content-based similarity