Documentation for Ecommerce recommender System

1. Business Understanding

Overview

In today's digital age, many activities are conducted online with the click of a button, whether it's shopping, communicating, playing games, or working. Online shopping platforms like Amazon, Jumia, Kilimall, and Jiji have revolutionized the way consumers purchase goods. They offer a vast array of products—from groceries and electronics to kitchenware and home appliances—delivered directly to consumers' doorsteps. However, the convenience of online shopping comes with its challenges. Consumers are often faced with an overwhelming number of choices, with numerous products from different manufacturers and sellers, each offering various functionalities and ratings. The abundance of options can lead to decision fatigue, where consumers spend a significant amount of time browsing before selecting a product that meets their needs. To enhance the shopping experience, increase customer satisfaction, and boost sales, there is a growing need for intelligent systems that can provide quick and personalized products and which use sophisticated algorithms to suggest products tailored to individual user profiles.

Traditional recommendation systems, such as collaborative filtering and content-based methods, each have their strengths and limitations. To overcome these limitations, hybrid recommender systems have emerged as a powerful solution. By combining multiple recommendation techniques, hybrid systems aim to deliver more accurate, relevant, and personalized recommendations. These systems leverage the strengths of each method to provide a comprehensive and nuanced understanding of user preferences. For example, a hybrid system might integrate collaborative filtering with content-based filtering, combining insights from user interactions with detailed product attributes to refine recommendations. The proposed hybrid recommender system aims to address the challenges of online shopping by providing users with quick and personalized product suggestions. By employing a combination of recommendation techniques, the system will enhance the shopping experience, reduce decision fatigue, and increase overall customer satisfaction. This approach not only benefits users by simplifying their decision-making process but also supports businesses in driving sales and improving customer engagement.

Challenges

1. Volume of Choices:

The primary challenge in online shopping is the vast number of available products. Consumers are often overwhelmed by the plethora of options, leading to extended browsing times and frustration. This challenge is exacerbated by decision fatigue, where the abundance of alternatives makes it difficult for users to promptly select the best option.

2. Complexity of Recommendation Algorithms:

Creating a system that accurately reflects individual user preferences requires sophisticated algorithms capable of understanding and predicting user needs. The diversity of products and the variability in user preferences make this task even more complex.

3. User Experience:

Ensuring a seamless and user-friendly interaction with the recommender system and chatbot is crucial. Users expect a smooth, intuitive interface when engaging with the system. The current interface restricts the users to the input that can be fed into the system leaving minimal space for users to engage flexibly with the system. Any friction in this interaction can lead to dissatisfaction and reduced effectiveness of the tool.

Proposed Solution

The goal is to develop a recommendation system that interacts with users, understands their product requirements, and suggests suitable products based on given specifications. The recommender system will use machine learning algorithms to provide accurate and personalized recommendations in response to user queries. We also plan to incorporate a chatbot into the system in order to provide an easier interaction interface to the user. The chatbot system will enable dynamic user input to be fed into the recommender system based on the user's requirements.

Here's how the proposed solution will work:

1. **User Interaction**:

The recommender system and the chatbot will engage users through a conversational interface, allowing them to specify their product requirements and preferences. By understanding user input in natural language, the chatbot and the recommender system can interpret needs more accurately, translating these into actionable queries to retrieve relevant product recommendations.

2. **Recommendation Engine**:

The core for the chatbot and the recommender system will be its recommendation engine, powered by machine learning algorithms. These algorithms will analyze user inputs, historical data, and product features to generate tailored recommendations. The engine will consider factors such as user preferences, past interactions, and current trends to deliver suggestions that align closely with the user's specific needs.

3. **Continuous Learning**:

The recommender system and chatbot will be designed to learn from user interactions over time, improving its recommendations as it gathers more data about user preferences and behaviors. This iterative learning process will help the chatbot provide increasingly accurate and relevant recommendations, making it more effective and valuable to users.

2. Metrics of Success

I. Recommendation Accuracy:

Percentage of correct recommendations based on user feedback. This will be a key indicator of effectiveness of the chatbot and the recommender system in understanding and meeting user needs. Root Mean Squared Error of close to 0 to evaluate model efficiency.

II. Response Time:

Average time taken by the chatbot and recommender system to respond to user queries. Quick response times are crucial for maintaining user engagement and satisfaction.

Future Metrics of success

a) User Satisfaction:

Measured through user feedback, satisfaction surveys, and ratings. High user satisfaction will reflect the success of the chatbot and recommender system in enhancing the shopping experience.

b) Engagement Rate:

Number of interactions per user and frequency of use. A high engagement rate indicates that users find value in the chatbot and are likely to return.

c) Conversion Rate:

Percentage of users who make a purchase based on the recommendation from chatbot and recommender system. This metric will directly measure their impact on sales.

3. Conclusion

Creating a recommender system and a chatbot that provides product recommendations based on user specifications addresses the need for quick and personalized shopping assistance. By leveraging NLP and machine learning, the chatbot can enhance the user experience, drive customer satisfaction, and increase sales. Continuous learning and adaptation to market

dynamics will ensure that the chatbot remains effective and relevant in a rapidly changing online marketplace.

4. Problem Statement

Consumers often struggle to find the right product that meets their specific needs due to the vast array of options available. This can lead to decision fatigue and dissatisfaction with the online shopping experience. There is a need for a system that can assist users by providing personalized product suggestions based on their unique specifications. Decision fatigue can result in a negative shopping experience, characterized by prolonged search times, increased dissatisfaction, and potentially missed opportunities for making purchases. In addition, the complexity of product options makes it difficult for users to make well-informed decisions without assistance. They need a system that can quickly sift through the myriad of choices and present them with options that align closely with their preferences and requirements.

To address these issues, there is a pressing need for a more sophisticated recommendation system that can integrate multiple techniques to provide accurate and personalized product suggestions. Such a system would not only improve the efficiency of the shopping process but also enhance user satisfaction by delivering recommendations that are tailored to individual preferences and needs. The problem, therefore, is to develop a hybrid recommender system that combines various recommendation techniques to offer users a seamless and personalized shopping experience. This system must effectively handle the complexity of product options and user preferences, reduce decision fatigue, and ultimately lead to increased customer satisfaction and sales for businesses. By addressing these challenges, the hybrid recommender system will enhance the overall quality of online shopping and support both consumers and retailers in navigating the digital marketplace more effectively.

5. Objectives

1. Conduct Exploratory Data Analysis (EDA) to understand user behavior and product attributes.

2. Create an algorithm that matches user specifications with suitable products.

3. Develop a recommendation system that provides accurate and relevant product recommendations.

4. Develop a chatbot that is user-friendly and intuitive to enhance user experience.

5. Iterate on the system based on continuous learning and data-driven insights.

DATA

We sourced for data by web scraping using scrapy through Flipkart "https://www.flipkart.com/" to details about the products they do sell. Scrapy was used to do the scrapping. The data we scrapped contained about 12041 rows and 12 columns. We then looked for an additional dataset of mobile phones from the same website and it contained 3114 rows and 8 columns. We then combined these two datasets and used them in our project.

DATA UNDERSTANDING

Our combined new dataset contained the following columns

category_1 - Contained a general category of the products eg. 'Sports, Books and more'

category_2 - Contained a more specific category of category_1 eg. 'Exercise Fitness'

category_3 - contained a specified category of category_2 eg. Cardio Equipment

title - Holds the name and description of the product

product_rating - the rating of the product

selling_price

mrp - marked retail price

seller_name

seller_rating

Brand - Phone brand

Model - Model of the phone

Color

Memory - RAM size

Storage

Original Price

DATA CLEANING

Data Cleaning process

Handling Missing Values:

Title Column: The title column had a few missing values. We deleted those rows that had no titles from our dataset. This was done to maintain data integrity as well as avoiding any ambiguity in identifying products sold in Flipkart's platform.

Product Rating Column: We had some missing entries in the product_rating column. From the identification made on these missing values, we substituted them with placeholders as we planned on handling them in the subsequent analysis phase.

Combining Datasets:

After scraping and obtaining the two datasets, we combined them into a single dataset to allow for a more detailed analysis. Merging required careful matching of the columns as well as ensuring that no data was lost or misaligned during this process.

Data Type Consistency:

We checked numerical columns such as selling_price, mrp and Original Price to make sure that they are in correct float or integer formats. Moreover, we transformed categorical columns such as category_1, category_2, category_3, and Brand into the appropriate string format.

Data Duplication:

We went ahead and eliminated duplicate entries from our dataset so that biases do not influence any analysis. This will help us to retain uniqueness and accuracy of our data by removing any duplicate records found.

Standardizing Data Entries:

We conducted standardization to ensure consistency in data entries especially for categorical variables like Brand, Model and Color. For instance brand spelling variations due to spelling mistakes or case sensitivity were fixed.

Conclusion:

The next analysis was strongly based on the data cleaning process. Our clean and reliable dataset was prepared by eliminating duplications, consistent data types, and incomplete data. The quality of records was improved through this careful approach to data preparation resulting in more accurate conclusions derived from analysis. Therefore with this cleaned dataset we can now move forward in exploring other aspects that will require finding out the trends, patterns and key insights from product data gathered from Flipkart.

EXPLORATORY DATA ANALYSIS

Analysis was done on the already cleaned dataset in order to gain insights on the relationships and distribution of various variables in the dataset.

We plotted the distribution of the column ‘category_2’ since it's a more specific column of the products and we derived that Mobiles had the highest representation in our dataset, followed by babycare, accessories and ethnic wear respectively.

We further plotted the top 5 category_3 with respect to the top 10 category 2. This plot enabled us to know the top distributions of products per category. We also plotted the top 5 priced category_3 per category_2.

We wanted to know if there is a correlation between the product price and the product rating. The calculation of the correlation coefficient of product price and product rating gave us a '-0.03208'. This means that there is no relationship between the two variables.

We also did some analysis on the 'title' column using NLP techniques since it would be the model to understand the queries searched by the user and provide relevant output based on the query.

MODELING

In this section, we built classification models using

1. SVDpp(Singular Value Decomposition Plus Plus)
2. SVD(Singular Value Decomposition)
3. KNNWithMean
4. Baseline Model / KNNBasic
5. Hybrid

The best performing ones were tuned and ensembled to produce one model however this did not exhibit better performance. Further scaling of the dataset was done and cross validation included to improve the accuracy score

SET UP

This is the process to follow when one wants to check on this work on their local machines or even contribute to this project.

All these should be **done in the terminal.**

1. Clone the Repository:

```
git clone https://github.com/yourusername/your-repo-name.git
```

```
cd your-repo-name
```

2. Create and Activate a Virtual Environment:

```
python3 -m venv env
source env/bin/activate
# On Windows, use `env\Scripts\activate`
```

3. Create your branch and move into your branch

```
git checkout -b branch-name
```

4. Install Dependencies:

Generate a requirements.txt file using this command

```
pip freeze > requirements.txt
```

Install the required packages:

```
pip install -r requirements.txt
```

Backend Setup

1. Set Up the Database:

Run the following commands to create and apply database migrations:

```
python manage.py makemigrations
python manage.py migrate
```

2. Create a Superuser:

Create a superuser to access the Django admin panel:

```
python manage.py createsuperuser
```

3. Run the Development Server:

```
python manage.py runserver
```

Frontend Setup

If using plain HTML/CSS/JavaScript, ensure your static files are correctly placed in the `static` directory of your Django project.

Running the Application

1. Start the Backend Server:

```
python manage.py runserver
```

2. Access the Application:

   Copy this url and paste it on the web browser `http://127.0.0.1:8000`

Usage

- **Admin Panel**: Access the Django admin panel at `http://127.0.0.1:8000/admin/` using the superuser credentials.

- **Task Management**: Use the web interface to search and get recommendations for products.