**Recommendation System**

**Overview**

The main goal of this project is to create a product recommendation system. This is based on the user choices, behaviour and ratings from an ecommerce platform.

**Introduction**

A recommendation system is a software tool or algorithm designed to suggest items or content to users based on various factors, such as past behavior, preferences, or item characteristics. These systems aim to personalize user experiences by predicting what users might find interesting or useful.

For this project I have considered an e-commerce store data named 'olist'. I have performed preprocessing and applied algorithms such as collaborative filtering, matrix factorization, content-based filtering and KMeans to derive insights from the data and to get the required recommendations for the user data.

The goal was to find the use cases from the data and make suggestions to the customers based on their shopping behaviour and relevant data. We are leveraging the data of customers to increase our customer retention rate, revenue and also to increase the number of loyal long-term customers. By following these approaches, we can satisfy the customers with our data by providing the relevant services and also, we can focus on building the revenue.

**Data Overview**

The Olist dataset comprises several files, each containing different information:

Dataset link: <https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=olist_customers_dataset.csv>

olist\_orders\_dataset.csv - Contains information about orders, including order IDs, customer IDs, order status, purchase timestamps, and more.

olist\_order\_items\_dataset.csv - Contains details about items within each order, including product IDs, order IDs, and quantities.

olist\_products\_dataset.csv - Contains product information such as product IDs, product category names, and product descriptions.

olist\_customers\_dataset.csv - Contains customer information, including customer IDs and location details.

olist\_reviews\_dataset.csv - Contains customer reviews, including review scores, comments, and timestamps.

**Data Preprocessing**

This step involves cleaning the collected data and preparing the data to give it to the recommendation algorithms. Below are some important steps:

* Data Cleaning: Remove duplicates, handle missing values, and ensure consistency across datasets.
* Merging Datasets: Combine relevant datasets to create a unified view of orders, products, and customers.
* Feature Engineering: Create new features that can help improve the recommendation system, such as average product rating, total purchases per customer, etc.
* Normalization: Normalize numerical features to ensure they contribute equally to the distance calculations.

In my dataset which I have chosen, there were many null values in the orders dataset. So, to remove those I have considered only the values which had a status of ‘delivered’ for that order. Because the status of remaining where unknown and they do no good while refining our data, and further these values just hinder our process.

For the other columns, I have replaced the null values with 0, and few of them where deleted. In most of the cases I have taken columns only relevant to our analysis. As I mentioned, this olist dataset is relatively big and it has over 1 lakh rows in it. So to process this large amount of data and to derive insights, only particular features where considered like customer\_id, product\_id, review\_score, review\_id, price, purchase\_date etc. These are the main one’s which I took.

**Model Building**

This is the main implementation for a recommendation system. Here we are going to use different approaches to come to a conclusion for our recommendation.

**Part 1**

For one type of implementation, I have used the RFM framework to build our customer segmentation model. RFM stands for Recency, Frequency, Monetary Value. This would allow us to create the following segments:

* Low Value: Customers who are less active than others, not very frequent buyers and generate very low revenue.
* Mid Value: In the middle of everything. Uses the platform fairly frequently and generates moderate revenue.
* High Value: The group we don’t want to lose. High Revenue, Frequency and Low Inactivity.

Now, I have used K-means clustering to assign customers to a particular sub-category based on categories such as Recency, Frequency and Monetary. In the end, I have created a matrix which shows the relationship between these three categories.

For recency, the inactive days were considered by calculating the last customer purchase to the latest date present in the dataset. Later, the distribution of the data based on these attributes is shown. These values are further divided into clusters for the proper visualization of the data.

Then later, the frequency is calculated based on the number of orders for that particular customer, the distribution of the data is shown using bar plot representation. Here again, the clusters of the data are studied.

The labels are renamed for the clarity of the representation.

For the Monetary part, the amount paid by each customer is considered. The data is divided into clusters based on a number. By this we can easily differentiate the high value customers who are spending more money buying the products of the olist.

In the end a final matrix is constructed based on the customer data to see the insights between all these attributes and how they are effecting the sales of the olist and their ecommece platform.

More detailed information is available in my github python files: <https://github.com/JyothirmaiV13/Quest-labs-Ai/blob/main/FinalDoc.ipynb>

**Part 2**

This approach is more general compared to the previous one. Here I have used different algorithms collaboratively to achieve the results.

Firstly, I have imported the relevant datasets, and merged the required data into a single DataFrame to make the operations easy.

After calculating the Average order value for each user and the frequently bought together items, average ratings for the products I have created a matrix using Matrix Factorization using all these values to observe the relationship between these values. Then generated the final ratings matrix. The next step is I have created the product recommendation using this matrix and implemented KNN algorithm for this.

Later on, using this matrix, I have further designed the SVD algorithm which gives the top scores of the products which are similar and generates the Top Recommended Products as an output for the given customer ID.

The future work and the detailed explanation of the code is present in my github. <https://github.com/JyothirmaiV13/Quest-labs-Ai/tree/main>