Recommendation System using SVM and K-means Clustering

Shikha Tiwari
Khoury College of Computer Sciences
Northeastern University
Boston, USA
tiwari.shi@northeastern.edu

1. Introduction -

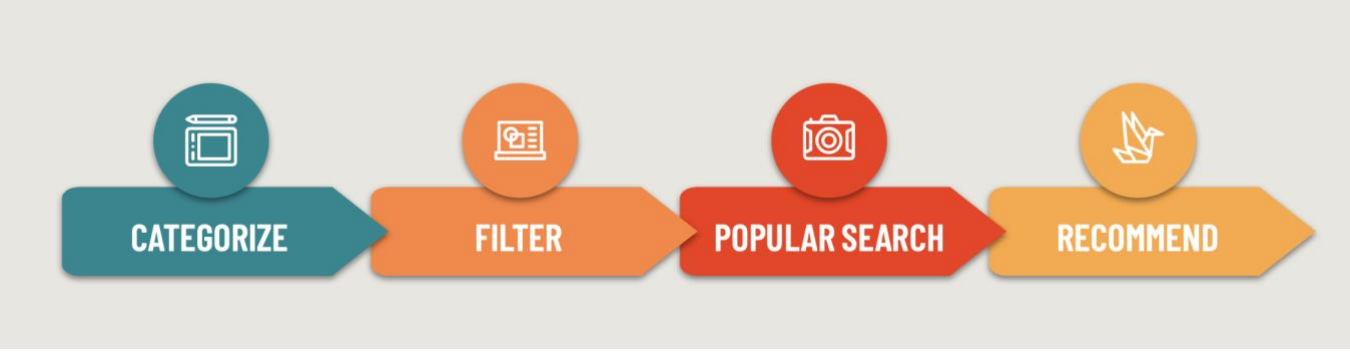
Recommendation systems are crucial in enhancing user experience and engagement across various platforms by suggesting items of interest based on user preferences and behaviour. In this report, we explore two recommendation approaches: matrix factorization with Support Vector Machine (SVM) and cosine similarity, alongside a popularity-based system.

Matrix Factorization with SVM:
Matrix factorization techniques aim to decompose a user-item interaction matrix into latent factors, capturing underlying patterns in user-item interactions. SVM, a supervised learning algorithm, is integrated with matrix factorization to enhance recommendation performance. By leveraging SVM's ability to handle non-linear relationships and complex decision boundaries, it further refines the recommendation process, particularly useful in scenarios with sparse or noisy data.

Cosine Similarity:
Cosine similarity measures the cosine of the angle between two vectors, representing the similarity between items or users in a high-dimensional space. In recommendation systems, cosine similarity computes the similarity between user-item vectors, recommending items similar to those previously liked or interacted with by the user. Its simplicity and effectiveness make it a popular choice, especially in scenarios where computational efficiency is crucial.

Popularity-Based System:
A popularity-based recommendation system recommends items based on their overall popularity or frequency of interaction across users. While simplistic, it serves as a baseline for comparison and can provide valuable insights, particularly for new users or in the absence of personalized data. However, it may suffer from the "popularity bias," recommending only popular items and overlooking niche or less-known options.

2. Popularity Based Recommendation System -



In these system, we categorize the products like in ecommerce dataset we had product categories and for each category we will pick the popular product. In the project we have Event Type which have three events - view, cart and purchase and while performing the EDA I found that most of the interaction of users with product was of type event so recommending product with more count will be bias and we will be recommending only products which have been viewed more so we deduce a weight on basis of event type on the ratio of how events were distributed.

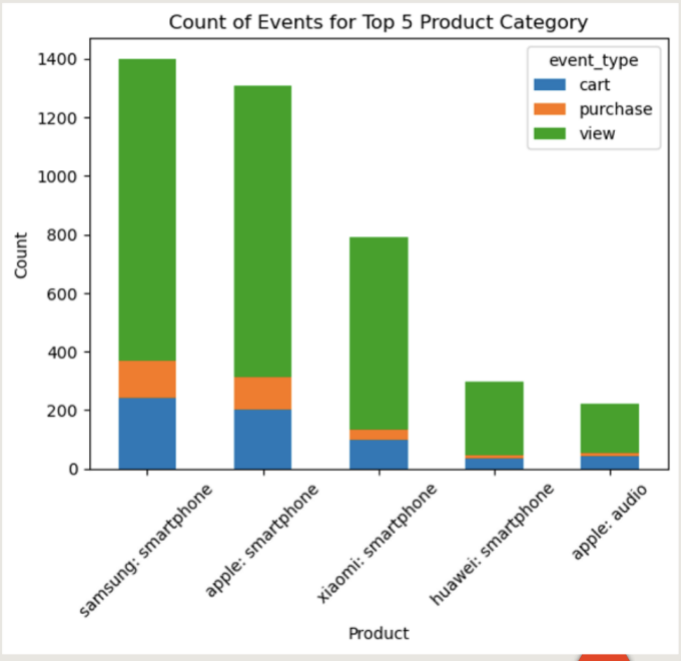
Weight Factor

Weight Formula -

Option 1 - On basis of value count

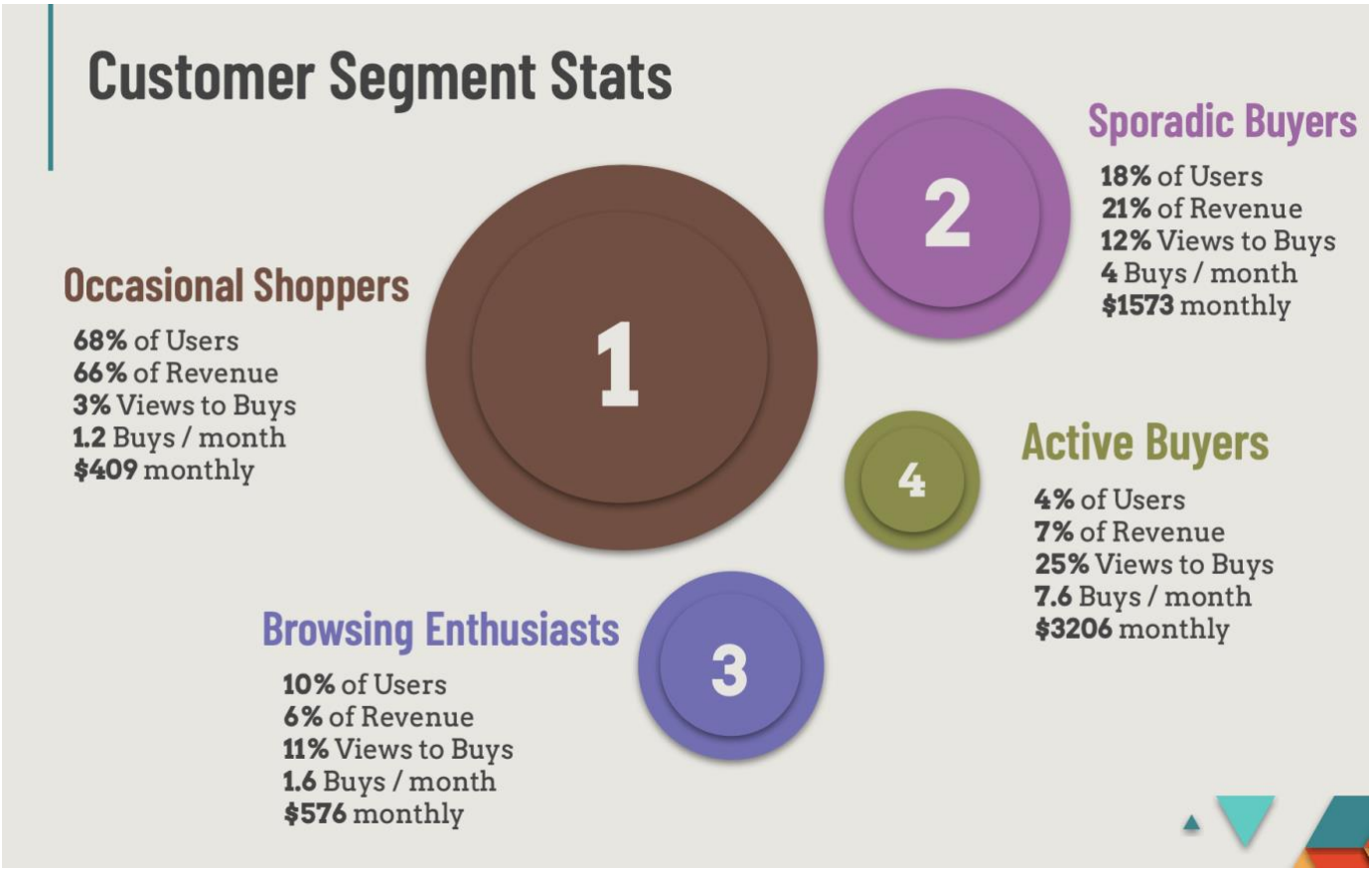
Option 2 - Weighted Formula using Event_type

Event_Type	User_Rating
View	1
Cart	8
Purchase	16



3. Collaborative Filtering based recommendation system-

For this, I started with clustering users on their purchase history using K-means clustering technique and got 4 clusters of users - occasional shoppers, sporadic buyers, browsing enthusiasts and active buyers.



As we can observe in the above 4 clusters of users, each clusters have different set and they interact with the product in a different way and generate different set of revenue so we need to target the users separately hence we developed recommendation system for each set of users targeting them specifically to get the similar products.

Item-Item Collaborative filtering -

So for this we utilized matrix factorization and have item user matrix with productids and user ids and for value we utilized Event_type weight assigned as our ratings which user gave(as this is how they interacted with the product), once we had our matrix I used SVM method to populate the sparse matrix and then further used cosine similarity to get the similar products which can be recommended to the users if they are browsing one product.

4. Product Overview-

This is how the system will work and will give trending (popularity based system) and item based similar products(collaborative recommendation system).