**Amazon Recommender System**

**Team : ML\_DATA**

**Dhruv Panchal(001055844)**

**Nikhil Joshi(001029458)**

**Sourav Sharma(001374865)**

**Vinod Kumar Mullangi(001057553)**

**Problem Statement** :-

The world of retail is changing rapidly. Many brick and mortar locations are closing and being replaced by online stores, direct-to-consumer brands and subscription/membership services. However, while the breadth of assortment is something that drives customers to a website, a lot of eCommerce platforms fail to sell through a high percentage of their merchandise. This is often due to poor user browsing experience. Customers can spend hours scrolling through hundreds, sometimes thousands of items of merchandise never finding an item they like. Shoppers need to be provided suggestions based on their likes and needs in order to create a better shopping environment that boosts sales and increases the time spent on a website.

This series will demonstrate a popular solution to this problem, recommender engines.

**Importance** :-

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet. [Information retrieval systems](https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems), such as Google, DevilFinder and Altavista have partially solved this problem but prioritization and personalization (where a system maps available content to user’s interests and preferences) of information were absent. This has increased the demand for [recommender systems](https://www.sciencedirect.com/topics/computer-science/recommender-systems) more than ever before. Recommender systems are beneficial to both service providers and users. They reduce transaction costs of finding and selecting items in an online shopping environment. Recommendation systems have also proved to improve decision making process and quality. In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products. In scientific libraries, recommender systems support users by allowing them to move beyond catalog searches. Therefore, the need to use efficient and accurate recommendation techniques within a system that will provide relevant and dependable recommendations for users cannot be over-emphasized.

**Work** :-

We proposed below following algorithms to build a robust recommender system for Amazon Electronic Products.

1. Popularity based

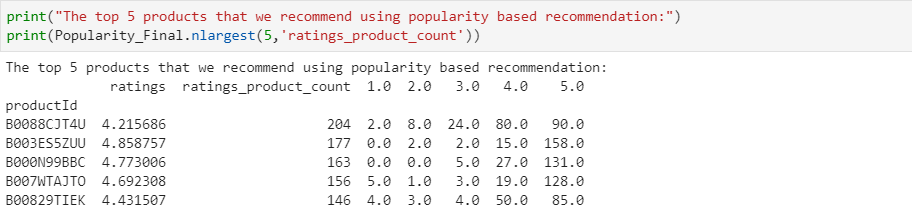
2. Collaborative filtering using Singular Value Decomposition

3. Collaborative filtering using KNNWithmeans (user-user similarity)

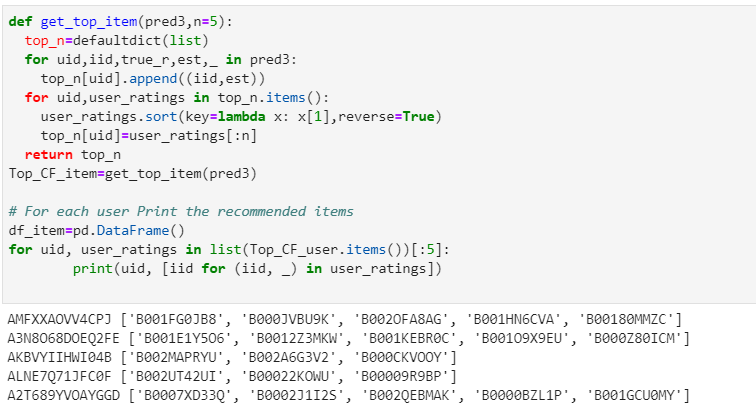
4. Collaborative filtering using KNNWithmeans (item-item similarity)

5. Hybrid Recommendation System

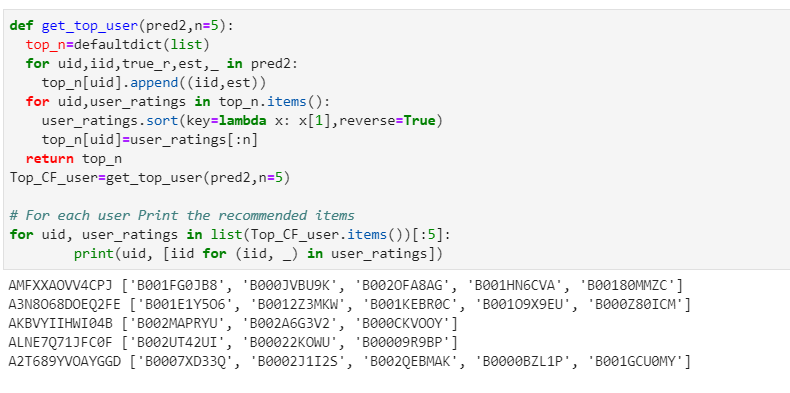
The system using Popularity based recommendation model that works on principle of popularity that identifies the products that are popular among users. This will give the users recommendation of products that are in trend, high in demand and are bought by users.



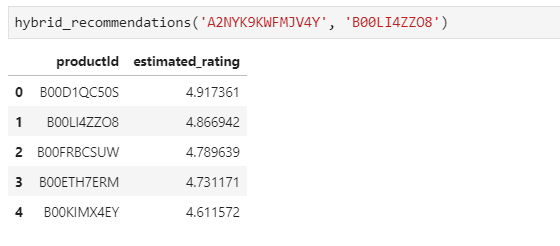
Collaborative filtering method to overcome scalability issue by generating a table of similar items offline through the use of item-to-item matrix. The system then recommends other products which are similar online according to the users purchase history.



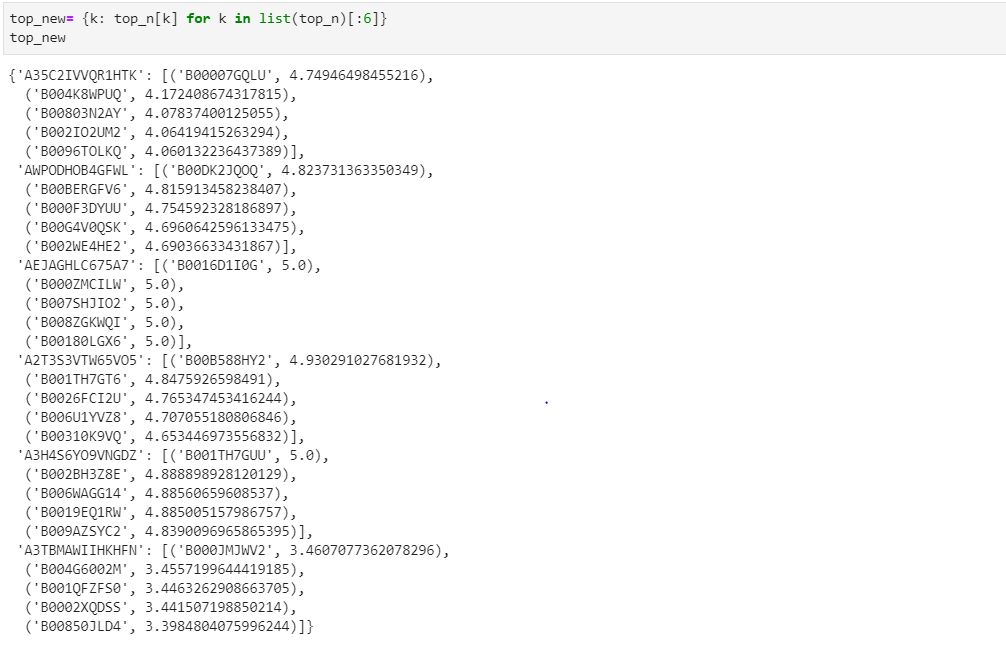
On the other hand, user-user collaborative filtering is an algorithmic framework where the neighboring users are identified based on the similarity with the active user, and then scoring of the items is done based on neighbor’s ratings followed by a recommendation of an item based item’s scores by the recommendation system.



Whereas, hybrid filtering focuses on combining benefits of both Content-Based and Collaborative filtering algorithms, For example: CF provide recommendations using rating matrix now what happens when there is no rating given by a user (new user) then in such case the contents of user-item (CB filtering) can be used for recommendations.



Singular value decomposition (SVD) is a collaborative filtering method. The aim for the code implementation is to provide users with Electronic Product recommendation from the latent features of item-user matrices.



**Conclusion** :-

We achieved to propound top 5 suggestions to users using all the above-mentioned algorithms and in the event reaching to a goal of reducing the painstaking task for our customers to remember what they like and which products they prefer.

Here, [Recommender systems](https://www.sciencedirect.com/topics/computer-science/recommender-systems) open new opportunities of retrieving personalized information on the Internet. It also helps to alleviate the problem of information overload which is a very common phenomenon with [information retrieval systems](https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems) and enables users to have access to products and services which are not readily available to users on the system. Various [learning algorithms](https://www.sciencedirect.com/topics/computer-science/learning-algorithm) used in generating recommendation models and [evaluation metrics](https://www.sciencedirect.com/topics/computer-science/evaluation-metric) used in measuring the quality and performance of [recommendation algorithms](https://www.sciencedirect.com/topics/computer-science/recommendation-algorithm) were discussed.

Popularity based system was useful in providing the users with products those have received good ratings and were on high demand. However, there was no variety and personalization to the user.

Whereas, collaborative filtering with KNNwithmeans was good in providing the recommendations with personalization by selecting products those were purchased by similar users or products that were similar to other products the user has bought/liked.

With SVD we were able to predict what rating a user will provide if a product is recommended to a user depending on other ratings he had provided, And recommendation can be done by setting a threshold value and recommend products that have high rating.

On comparing all the recommendations, we have built, each has its own advantages and disadvantages.

**References** :-

1. <https://towardsdatascience.com/web-scraping-recommender-systems-project-1d360fa678e4>
2. <https://towardsdatascience.com/how-to-build-a-model-based-recommendation-system-using-python-surprise-2df3b77ab3e5>
3. <https://towardsdatascience.com/deep-learning-based-recommender-systems-3d120201db7e>
4. https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf
5. https://medium.com/@madasamy/introduction-to-recommendation-systems-and-how-to-design-recommendation-system-that-resembling-the-9ac167e30e95
6. https://towardsdatascience.com/product-recommender-using-amazon-review-dataset-e69d479d81dd