

E-Commerce Recommendation System

Team Members :

Vishnu Prasanth Reddy

Aravind Kumar Sankara Narayanan

Saloni Katiyar

Table of Contents

Abstract :	2
Introduction :	2
Problem Statement :	3
Proposed System :	4
Project Pipeline :	7
Data Preprocessing :	7
Data Cleaning :	8
Feature Engineering Techniques :	8
Feature Selection :	8
EXPLORATORY DATA ANALYSIS :	9
Univariate Analysis :	10
Graphical Univariate Analysis :	10
Bivariate Analysis :	12
Graphical Bivariate Analysis :	12
Multivariate Analysis :	13
Graphical Multivariate Analysis:	13
MODELING:	14
CONCLUSION:	17
REFERENCES:	17

Abstract :

As the world is becoming more digital, we are already getting used to a lot of personalized experience and the algorithm to build recommender systems. Almost every web-based platform is using some recommender system to provide customized content. While most recommender systems still follow the usage of user- and item-based collaborative filtering processes, they have begun to also take into consideration the other social aspects of people's lives. While this method can be more efficient and effective in relaying recommendations to users, based on their preferences. In this project we will cover the topics that relates both item based collaborative filtering and user-based collaborative filtering and develop a machine learning model that incorporates these techniques using real time data.

Introduction :

Recommendations are a great way to monetize user behaviour data that businesses capture. Recommendation algorithms identifies hidden behaviours amongst users and products using raw users rating data

People are always looking or products and products come in variety of forms. The product might be of different types such as videos, books or groceries and within these categories there will be large variety of products to choose from. Services like amazon have millions of books and services like Netflix may have thousands of movies. People who are looking for this products also come in variety of forms. Every person is unique, hence, different people have different preferences and requirements in the products they are looking for. For an instance, on an e-commerce website some users may be looking for books and movies, others might be looking for stationary.

If you look for users and products there are clearly some sort of connections between them. Imagine if you are someone who is observing how users are behaving at a store. Then you could see that some people have an affinity or preference towards specific type of products that they need. This affinity between user and product is measured by **user-product relationship**.

If you look within products then you can see that some products would be similar to others, for example, books of same genre or dishes from the same cuisine. The extent of similarity between any pair of products can be measured using **product-product relationship**.

Finally, if you look within the group of users then you will notice that some people are similar in nature. Similarity between two people could be because they have common interests, for instance, they like to read the same book or they have a common friend. Such similarity between users can be measured using **user-user relationship**.

These relationships can provide tremendous insight. Let's say, you are a book store owner and you know which person has affinity for which kind of books then it becomes easy for you to recommend the right book to them. If you have a social networking account then you can use the knowledge of user to user relationship to recommend users to each other. Any relationship that we find among users and product can provide you with tremendous insight and that insight can be directly monetized in the form of recommendation system. Knowing the right product to recommend to the user can help us do things like:

- Create personalised promotional emails
- Personalised homepage
- Personalised notifications

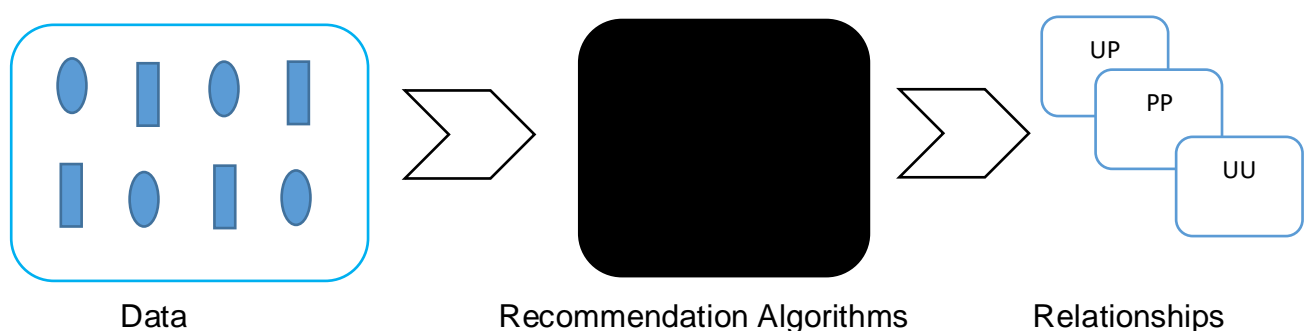


Fig 1.1 Workflow on establishing relationships using Recsys Algorithm

Problem Statement:

Our problem statement deals with providing a personalized recommendation system for ecommerce websites by tailoring the contents based on needs of each user. The following personalization techniques are incorporated to solve both customer and business requirements:

- Recall customers history, primarily how they act as expressed by what they viewed and purchased, identifying the customers frequently bought together items, the frequency of items purchased or viewed.

- Deliver the right promotion, content, recommendations for a customer based on actions, preferences and interests.

Proposed System :

Data can be in the form of the following:

- User behaviour data : Ratings, clicks, purchases
- User demographic data : Age , location, education, income
- Product attribute data : Genre, Cast, Cuisine

This data is fed to the recommendation algorithm which finally gives us the desired recommendation.

Recommendation algorithm can be broadly classified into three different categories. The purpose of the recommendation algorithm is to extract information from data about what kinds of relationship exists between user and product.

One of the most common uses of this data is to extract knowledge about what products a user already liked, viewed or purchased and see what other products we should recommend to the user. We can do this by three ways:

- Content Based Filtering
- Collaborative Filtering
- Association Rules Learning

Content Based Filtering :

To find products with similar attributes. This is known as **content based filtering**, where we are finding products based on similarity of attributes.

Collaborative Filtering :

To find products liked by similar attributes. This is known as **collaborative filtering**, where we are finding products with the help of other user or with the help of collaboration of other user.

Association Rules Learning :

To find complementary products. This is known as **association rules learning**, where we are finding a rule to associate with another.

In this project we are using **collaborative filtering**. So, we will dive deeper into it.

Collaborative filtering is one of the most used approach and it gives better result than content based recommendation system. YouTube and Netflix uses this type of recommendation system.

In short, it is based on the assumption that if user A watches movies like avenger and star wars and the user B shares the same interest i.e. he also watches the same movies, then the chances of movies watched by user B is most likely to be recommended to user A.

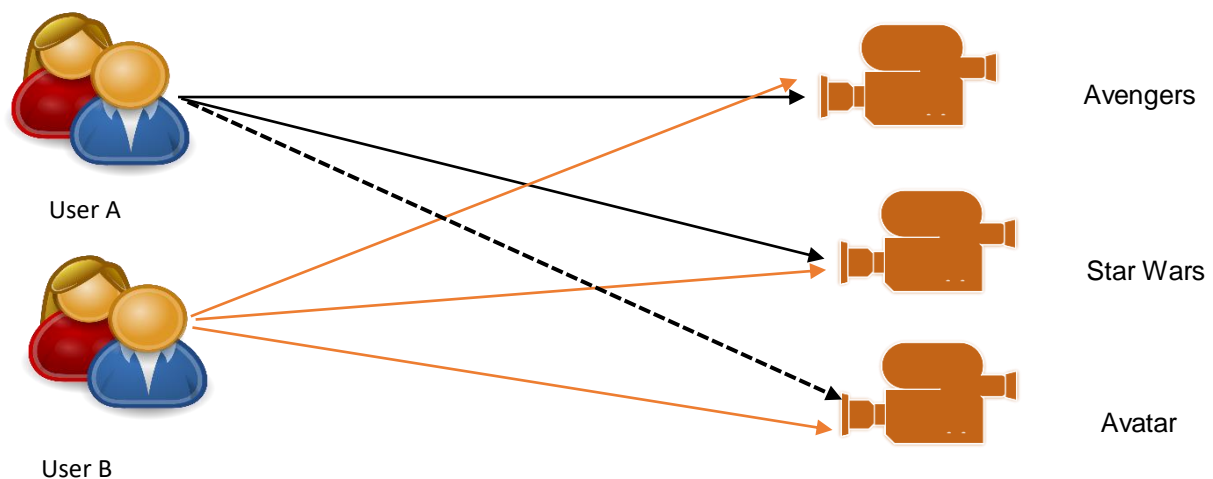


Fig 1.2 Collaborative Filtering Method

Content based algorithm requires a product attribute database, whereas, in case of collaborative filtering it requires only easily captures user behaviour data and no other data is required.

For every user and product combination all we need is to get a matrix like purchases, page views, clicks and ratings, these are the attributes which measures users affinity towards the product.



Fig 1.3 Workflow on Collaborative Filtering

Collaborative filtering requires single measure of affinity for the product by the user this is known as ratings. It can be one of these affinity matrix or combination of them.

There are two types of rating:

- Explicit rating: User directly gives rating either in a survey or on the website.
- Implicit rating: If it is an indirect measure of affinity such as through clicks or page views then it is called an implicit rating.

The rating data is represented in the form of matrix. User are represented along the rows and products are represented along the columns. Each cell represents a single rating whereas blank cell represents the rating for unseen products. Cells which are filled are used to predict values of the blank cells. Different techniques can be used to fill the blanks such as nearest neighbour model and latent factor analysis.

	P1	P2	P3	P4	P5
U1	1		2	2	
U2		2		5	
U3	3		4		1
U4	1	2		3	
U5			4		0

Fig 1.4 User- Item Matrix Utilization

Project Pipeline :

The below architectural diagram shows the pipeline of each phases followed throughout the course of this project.

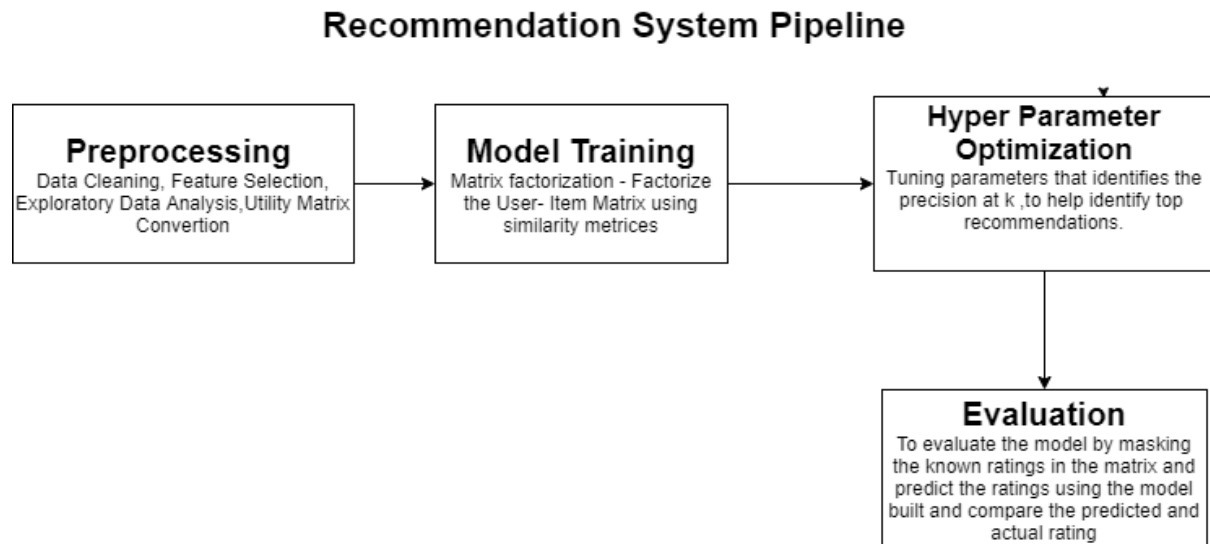


Fig 1.5 Project Pipeline indicating the tasks involved in each phases of the project

Data Preprocessing :

The Data preprocessing stage mainly focused on cleaning the data the following covers the detailed process and techniques which is involved in extracting, cleaning and preparing data in more structured way. We tend to analyze the data with feature engineering techniques and utilize them with various highly efficient methods and algorithms in order to obtain the best quality data that can be used to fit in the model.

The following are the techniques involved in the Data Preprocessing stage,

- Data Cleaning
- Feature Engineering techniques
- Exploratory Data Analysis
- Utility Matrix Convertio

Data Cleaning :

The dataset used includes missing values and irrelevant records that does not serve any purpose. Since the primary objective of this recommendation system is to recommend the customers with items based on their previous purchases or items belonging to same categories, it is important to select items that are restricted to atleast one category. Hence sampling of records is done based on category ID to fetch only records that belong to a particular category.

Feature Engineering Techniques :

The project includes the following feature engineering techniques for dimensionality reduction,

- Feature Selection
- Feature Creation

Feature Selection :

Having irrelevant features in your data reduce the accuracy of your model, hence it is important to select only the important features that fits the model well. The dataset where the recommended system is build upon is primarily based on products that are belonging to categories. Hence the item ID, category values and date of purchase are the attributes chosen under feature selection.

Feature Creation :

To build a utility matrix based on user-item relationship it is important to differentiate whether customer purchased the item or not. Since the data does not contain any user ratings. It is difficult to come up with the user-item matrix. Hence we explicitly rated by creating a feature where Transaction made by customer on a particular item is marked as 1 and add to cart done by customer on a item is marked as 0. With this attribute the utility matrix is formed which is explained in the modeling phase

EXPLORATORY DATA ANALYSIS :

Since we are dealing E-commerce website related products, the datasets provides information about the products and the customer details. Two Datasets are used in this project. One is the item properties dataset with 20 million instances and 4 attributes consisting of all the properties related to the products available such as item ID ,category ID the particular item belongs to, second is the Events dataset with 2 million instances and 5 attributes consisting of the item ID, Event made by the customer and the customer ID.

Below are is the architectural diagram describing the relationship between the datasets,

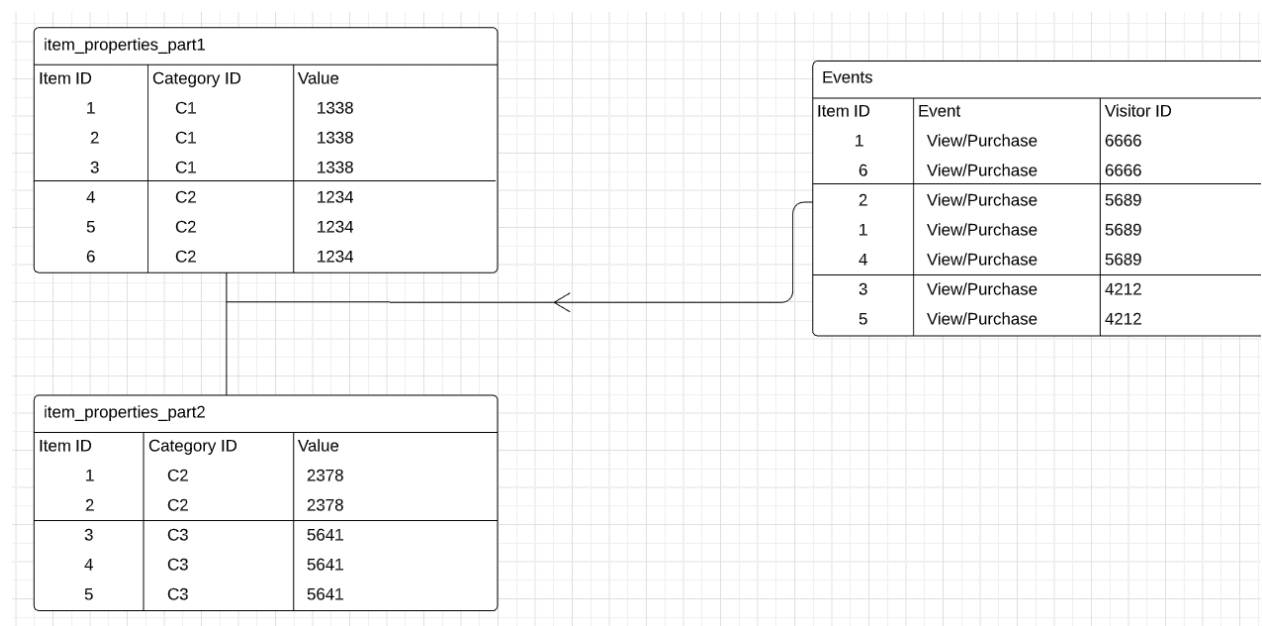


Fig 1.6 Architecture diagram representing the entity relationship between different datas

The above diagram establishes the entity relationship between the datasets Item_properties and Events. Item ID is the common attribute between both the datasets.

Further we will classify our Exploratory data analysis into the following phases as below,

- Univariate Analysis
- Graphical Univariate Analysis
- Bivariate Analysis
- Graphical Bivariate Analysis
- MultiVariate Analysis
- Graphical Multi Variate Analysis

Univariate Analysis :

In the Univariate Analysis phase , we distinguish the unique items available in the market. Out of 788214 items purchased in the particular duration, 417503 items are unique items. Relatively Out of 2756101 total visitors , there are 1407580 unique visitors.

The number of unique visitors is further decomposed into different groups segregating the visitors who bought atleast one item and visitors who did not buy anything. Through data exploration techniques it is observed that out of 1407580 unique visitors there are 11719 visitors who actually bought atleast one item. With this result it is obvious that the remaining 1395861 visitors did not buy anything but just viewed the items.

With respect to the event dataset, it is important to know the number of items which occurred the most , this helps our model to recommend items based on events. Using univariate analysis, the number of events has been identified based on event types as shown below,

Events	Count
View	2664312
AddToCart	69332
Transaction	22457

Fig 2.2 Number of events occurred in each Event Type

Graphical Univariate Analysis :

The Univariate Analysis done in the previous step has been visualized to identify the patterns and trend of customer events in the E-commerce website. Below line chart visualizes the trend of events occurred in the E-commerce website by the customers.

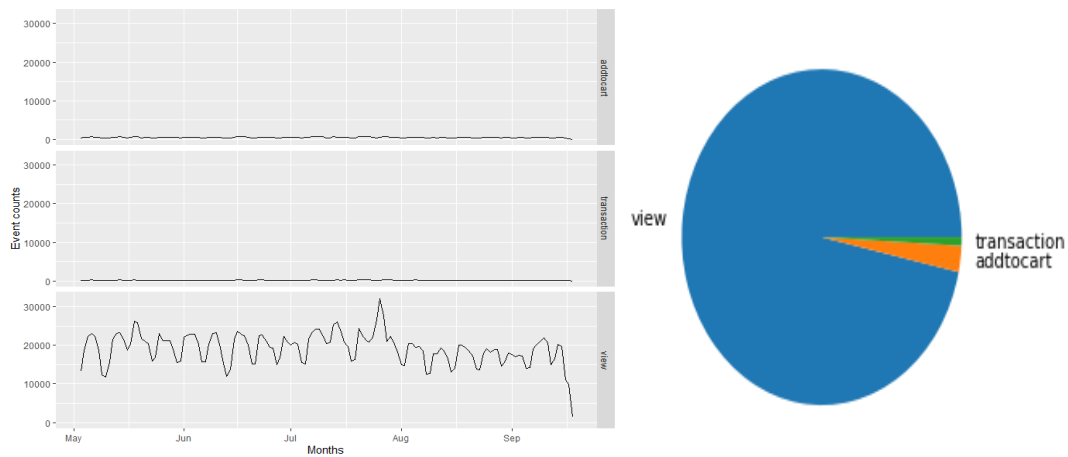


Fig 1.7 Trends and Pattern on Type of Events occurred over time

The analysis infers that the view events occurs at larger amount whereas addtocart and transaction events occur minimal, indicating that the customers views number of items of their choice to come up with the best product of their own preference.

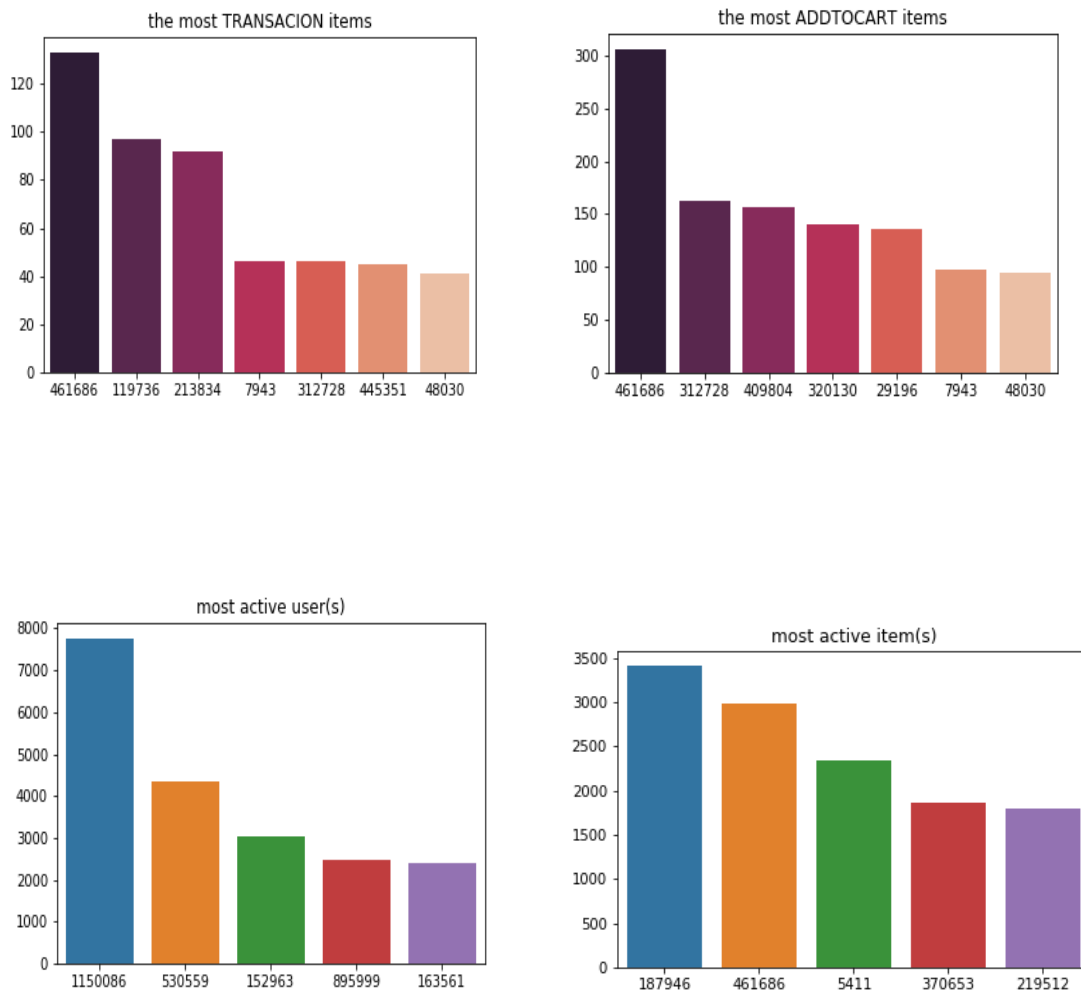


Fig 1.8 Identifying the most favorite items and most active users

The above visualization clearly specifies each product that are high on demand and the customers who are considered as frequent visitors based on the number of times they visit the website.

Bivariate Analysis :

In bivariate Analysis stage, the entity relationship between two attributes has been established. Considering each items belonging to a particular category, it is important to identify those items that belong to a particular category . Hence the number of items belonging to each category and the list of items that belong to the category has been summarized using multivariate analysis techniques.

	itemid	categoryid	items
0	[1957, 207072, 347416, 458345, 167416, 123751,...	0	135
1	[390448, 465859, 211677, 304558, 31165, 25820,...	1	867
2	[245380, 449019, 26377, 96493]	10	4
3	[134781, 135974, 291251, 425166, 352564, 38543...	100	10
4	[269839, 100740, 449301]	1000	3

Fig 1.9 Bivariate Analysis between attributes category, itemID

Graphical Bivariate Analysis :

Based on the above analysed data , category wise each items has been appeared for the customers by means of view, addtocart or transaction has been visualized. This further visualizes which items in a particular category gains more attraction among customer and based on that it can be recommended to other customers who are viewing the items in same category.

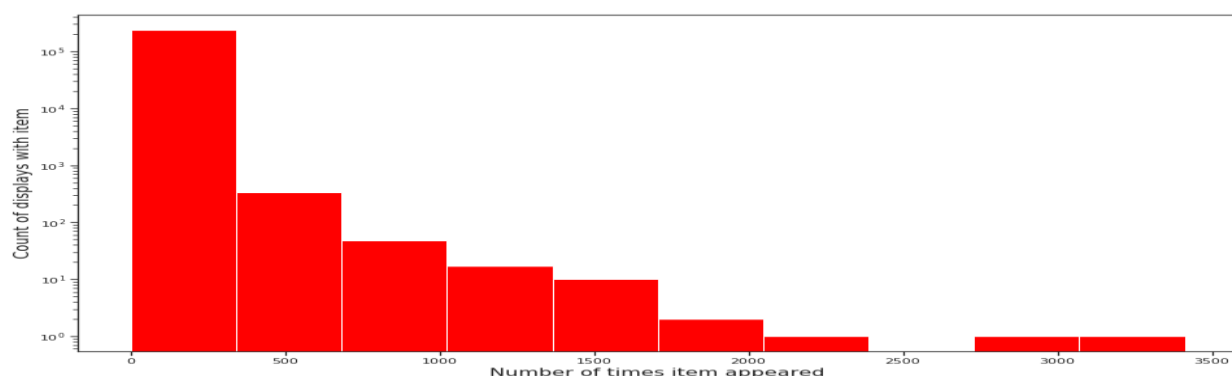


Fig 2.0 Frequently purchased items in each category

Multivariate Analysis :

The Multivariate Analysis establishes the correlation between more than two attributes. In our data, from the previous analysis we segregated the number of event categories like the number of view count, add to cart, and transaction count. Further grouping the attributes with respect to the date of visit, the number of events done on each day has been explored.

Index	timestamp	event	Eventcountspersday
0	2015-05-03	addtocart	296
1	2015-05-03	transaction	83
2	2015-05-03	view	13304
3	2015-05-04	addtocart	579
4	2015-05-04	transaction	154
5	2015-05-04	view	18681

Fig 2.1 Multivariate Analysis on Number of event types and event counts datewise.

The above figure clearly helps us understand the days where there is huge amount of customers checked in the website . These data will be widely used from the business point of view to identify the number of visitors visiting during weekdays or weekends and based on that pattern , the business can take decisions on availing offers for customers and thereby increasing the revenue.

Graphical Multivariate Analysis:

To visualize the overall relationship between attributes different combinations of entity relations has been established and visually plotted in the form of graphs to help this data driven recommendation system to provide the correct recommendations for the customers.

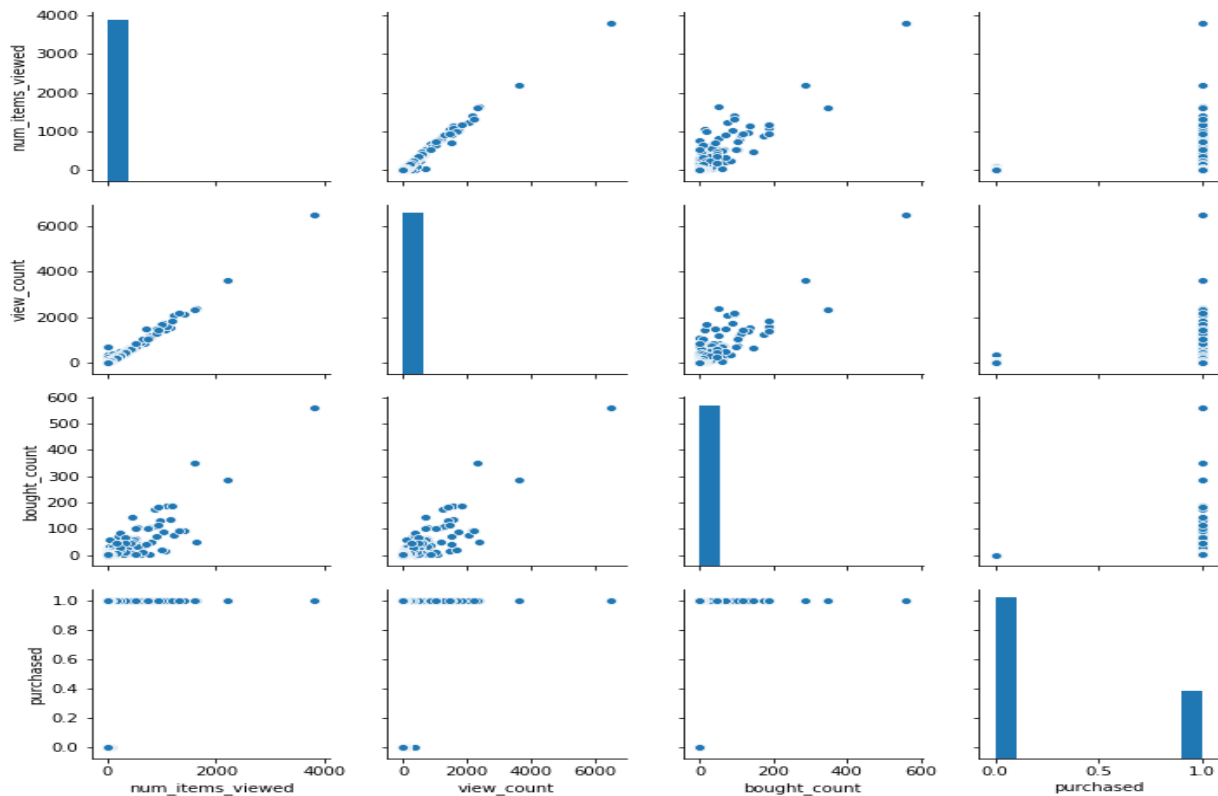


Fig 2.2 Multi variate analysis on event types

The plot above clearly indicates that the higher the view count, the higher the chances of that visitor buying something.

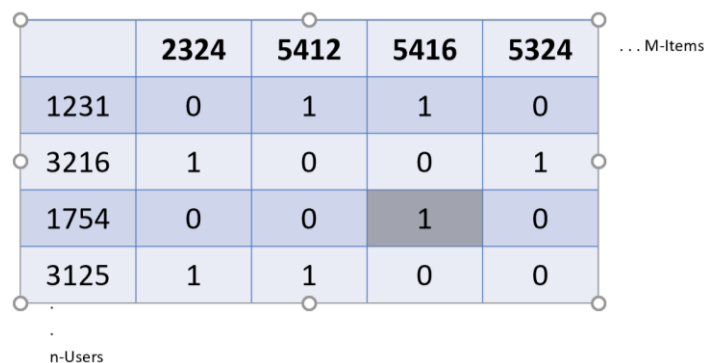
MODELING:

Since the dataset didn't have ratings. We have created Rating variable by implicitly rating the items based on user activity. We have filtered out viewed instances and incorporated only 'AddtoCart' and 'Transaction' only. Rating is equal to '1' if the user has purchased any item and '0' if the user has added the item to the cart.

Visitorid	ItemID	event	Rating
231212	62131	AddtoCart	0
113251	15246	Transaction	1
231451	5321	Transaction	1

Fig 2.3 Feature Creation on Explicit Ratings based on events

Using the ‘visitorid’, ‘ItemID’ and ‘Rating’. We can create User-Item binary matrix which is a traditional collaborative filtering matrix factorization method. Pure collaborative filtering approach cannot tackle the cold-start problem. Therefore, the train and test split must be done so instances of user or books in the testing set must have remaining instances in the training set. The training set matrix will have 31032 rows i.e users and 21076 columns i.e items.



	2324	5412	5416	5324	... M-Items
1231	0	1	1	0	
3216	1	0	0	1	
1754	0	0	1	0	
3125	1	1	0	0	
... n-Users					

Fig 2.4 User-Item Utility Matrix Transformation

We have used LightFM to compute matrix factorization and generalize recommendations. LightFM enables the integration of both item and user data into the traditional algorithm for matrix factorization. It defines each user and items as sum of the latent representations of their features such that recommendations will generalize new user to new items.

we have built a model using default parameters i.e.

- Number of components = 2
- Loss = 'bpr'
- Training AUC_Score : 0.5622
- Testing AUC_Score : 0.616
- Precision@k = 0.0026

The AUC score of LightFM on training set is just 6% more than any random model which is not satisfactory. Surprisingly, the LightFM model worked better on Testing set than training set. In addition, precision @k is very less.

Precision@K measures the proportion of positive items among the K highest-ranked items.

As such, it's very focused on the ranking quality at the top of the list: it doesn't matter how good or bad the rest of your ranking is as long as the first K items are mostly positive. This would be an appropriate metric to be show users the items that ate at the very top of the list.

To optimize the cost function, I've trained the algorithm with different combinations of parameters leading to hyper parameter tuning and captured the best estimator that performed better using AUC as evaluation metric.

Best Estimator Parameters:

- n_components : 110
- loss_function : warp-kos
- Learning rate : 0.1

AUC_SCORE:

- Training set: 0.9984
- Testing set: 0.9003

Precision @k=3:

- Training set:0.4428
- Testing set: 0.2918

There is a significant increase in AUC Score after hyper tuning. From the precision scores we can infer that top 3 recommended items are 44.28% relevant to user.

CONCLUSION:

Since we have used implicit binary ratings, traditional binary matrix factorization method is used for building the recommender system. In the future, all the user activities including clicks and viewed can be used and implicitly rate the item on a scale on 1-3. With the obtained matrix, we can implement a user-user similarity metrics using similarity metrics such as cosine similarity, jacard similarity and pearson similarity. This methodology helps in scoring the rating for items that were not rated by the users based on similar users. After scoring the ratings, we can recommend the items according to the ranking order. The biggest advantage of collaborative filtering is feature selection is not needed. The features required are unique users, items and their corresponding ratings.

REFERENCES:

1. Recommendation Systems medium: <https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a>
2. LightFM : <https://github.com/lyst/lightfm> ,
<https://making.lyst.com/lightfm/docs/home.html>