GREAT LAKES
INSTITUTE OF MANAGEMENT, CHENNAI

Recommendation Engine for Camera Rental Services (Klachak)

Submitted towards partial fulfillment of the criteria for award of PGPBABI

Submitted By

Jayaprakash Nallathambi	BACJAN17022
Abhishek Ramachandran	BACJAN17002
Harish Ganesan	BACJAN17020
Santhosh Murali	BACJAN17062

Under the Guidance of

Dr**. Monica Mittal**

East Coast Road, Manamai Village, Thirukazhukundram Taluk,
Kancheepuram District, Manamai,
Tamil Nadu 603102
Phone: 044 3080 9000

Acknowledgements

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to Dr Monica Mittal for her guidance and constant supervision as well as for providing necessary information regarding the project & also for the support in completing the project. Her experience and support guided us to make the project development process look simple. She provided valuable inputs at every step of the project.

We would also like to express our gratitude towards Mr. Ganesh Siddhamalli, Co-Founder of Klachaks, Velachery, Chennai, for his offer and co-operation which helped us in completion of this project.

Last but not the least we wish to thank Dr. Prof. P. K. Viswanathan , our course Director, for constant supervision, guidance and for being a source of inspiration in helping us to work on this project.

Date : Nov 15, 2017

Place : Chennai

Jayaprakash Nallathambi

Abhishek Ramachandran

Harish Ganesan

Santhosh Murali

Certificate of Completion

I hereby certify that the project titled “Recommendation Engine for Camera Rental Services (Klachak)” was undertaken and completed under my supervision by Jayaprakash Nallathambi, Abhishek Ramachandran, Harish Ganesan & Santhosh Murali all the four students of Postgraduate Program in Business Analytics and Business Intelligence (PGPBABIJAN2017).

Date : Nov 15, 2017 (Dr. Monica Mittal)

Place : Chennai

Mentor

Table of Contents

Acknowledgements 2
Certificate of Completion 4
Table of Contents 6
List of Figures 8
List of Tables 8
Executive Summary 9
Introduction 10
Camera Rental Services Industry 10
Klachak.com 12
Equipment Rentals 12
Photography Services 13
Problem/Opportunity Statement 13
Analysis Methodology 14
Methodology 14
Data Source 15
Primary Data Source 15
Secondary Data Sources 15
Tools & Techniques 15
Limitations 16
Literature Review 17
Recommender Systems 17
Content Based Filtering: 17
Collaborative Filtering Algorithms: 17
Hybrid Recommendation System: 18
Exploratory Data Analysis 18
Sales Yearly Trend. 18
Seasonal Trend 19
Percentage of Revenue 20
Cohort Analysis 22
Model Development 24

Data Extraction and Data Preparation. 24
How RFM value is segmented. 24
Mapping Cheat Sheet got Rating Derivation 25
Model Building and Evaluation 25
Collaborative Filtering [CF] 25
Content Based Filtering 31
Why Hybrid Approach? 36
Data Partitioning to test model fiteness 38
Sample Output 38
Results for Klachaks Recommender System 39
Sentiment Analysis 41
Data Cleansing Steps 41
Web Scrapping – Using Scrapy 42
Scrapy Data Flow 42
Implementation 45
Overall Architecture 45
Post Implementation 46
Recommendation Systems in Action 46
Conclusion 47
Future Scope 47
References 48

List of Figures

Figure 1 11
Figure 2 14
Figure 3 18
Figure 4 19
Figure 5 20
Figure 6 21
Figure 7 22
Figure 8 23
Figure 9 28
Figure 10 29
Figure 11 29
Figure 12 30
Figure 13 31
Figure 14 33
Figure 15 34
Figure 16 34
Figure 17 35
Figure 18 35
Figure 20 42
Figure 21 43
Figure 22 45
Figure 23 46
Figure 24 46

List of Tables

Table 1 10
Table 2 15
Table 3 24
Table 4 25
Table 5 25
Table 6 27
Table 7 32
Table 8 39
Table 9 39
Table 10 40
Table 11 40
Table 12 40
Table 13 41
Table 14 44

Executive Summary

Klachak.com is an online camera and accessories rental company, enabling photographers to rent cameras, accessories or photography services online. The company is positioned itself as a marketplace for amateur photographers to professionals across Tamil Nadu. Klachak is the first e-commerce website in Tamil Nadu operated in Hybrid model of Managed Market place and Inventory model for Camera Rentals. Initially company wanted to find out its state of market position and its some performance measures through exploratory data analysis, which later expanded its scope to build a Recommendation System for its website. This will help customer to analyze various other options before renting the camera, as well as for the company to up-rent and cross-rent of products.

The project takes on following sequence to complete, starting from understanding the business through several meetings with Klachak, then deep diving into transaction data of klachak. Through several exploratory analysis and visualization understanding the data well. Exploiting and preparing the data for Developing Recommendation Engine through Content and Collaborative Filtering based hybrid model. Evaluate and fine tune the model. Finally deploy the Program in Klachak website.

The duration of Data Extracted was from Feb-2014 to Oct-2017 for performing Exploratory data analysis. This gave enough space for understanding customer behavior and inventory performance. For building Recommendation engine same duration of data is used but with limited fields. The completed model is evaluated based on Root Mean Square Error between Actual and Predicted. Post implementation, data will be streaming near Realtime and model will become functional.

Once it is deployed in Production, it should improve Customer Satisfaction in addition to Customer Experience. As well as, Improved inventory performance as a auxiliary results.

Introduction

Camera Rental Services Industry

This is a budding industry with only very few active players. So, what is the business? Let’s assume that Someone called “John doe” is a excellent Photographer and would not mind crossing any limits to get best click. He wanders around the globe to capture the perfroct moments. Every photographer has their own gears which they used carry all around with them ever. {gear: Camera Body, Lenses, Filters, Tripod, Lights etc). John doe also has his own set of gears, he has Canon 5D Mark II Body, Canon 70-200 IS2 and Canon 16-35 L lens, Tripod and Travel bag. His total gear value is provided below.

Canon 5D M2	**₹ 70,000.00**
Canon 70-200 IS2	₹ 120,000.00
Canon 16-35 L	₹ 110,000.00
Tripod	₹ 27,000.00
Bag	₹ 3,000.00
Total	**₹ 330,000.00**

Table 1

With his limited existing set of gears, he limits his creativity. With Canon 70-200 IS2, he could capture long range objects but still not a very extensive telescopic range and f-stops are not wide enough. On the other hand, he has 16-35mm wide angle lenses which will cover larger area and minimal depth. So, whatever photograph he takes it will either long distance and not too long with moderate depth information or a wide angle. Suppose if he wants to take a portrait his gear pack will not help him. He has to have either “Canon EF 35mm f/1.4L II USM” or “Canon EF 50mm F/1.2L USM”. It will cost him additional ₹ 200,000 or ₹ 100,000 respectively. If he wants to go for macro photography, he has to invest ~₹ 100,000 additionally. John Doe can afford to buy these new lenses. But, there are Hobbyists, cost sensitive photography professionals, who might not invest such a large amount. Also, not all lenses are used frequently and portable. Application of Camera and lenses are very dependent on time and situation(event). Not everyone can afford to buy all kinds of cameras and lenses. This a costly profession.

That’s where the rental services industry identified the opportunity. These companies will have an inventory of all kinds of cameras, lenses and its accessories from almost all brands, (it can be be also based on Managed market model). They rent these gears to customers and the rent will be charged based on daily basis and also rent is dependent on the cost of the gear. This way it becomes win-win deal for both the company and Customers. All their inventory has now started minting money while Customers(Photographers) need not invest whooping money for purchasing lenses, instead spend very little fraction of the cost of the gear as rent and also gain access to wide range of options.

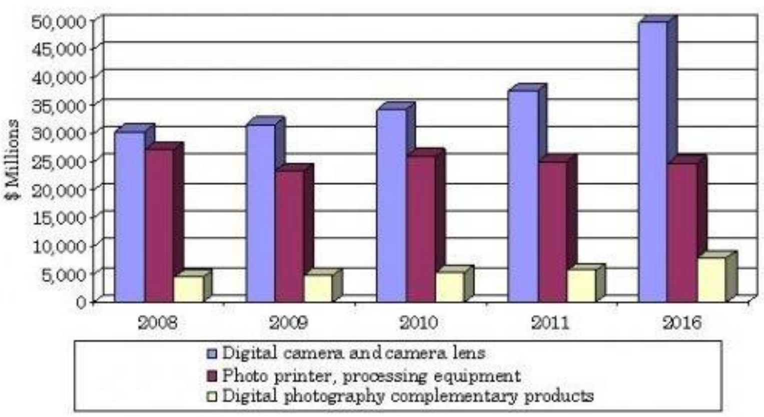


Figure 1

BBC Research has released a new report stating that the digital photography industry has an annual growth rate of 3.8%. Valued at $68.4 billion last year, the global market will reach an estimated value of $82.5 billion by 2016. The study defined the market as a combination of camera equipment, printing equipment, and complementary products. While the photo printing industry is predicted to struggle and lose $300 million between now and 2016, digital cameras and lenses will reportedly do just fine: they have a healthy annual growth rate of 5.8%.

Though it seems to be very attractive, this industry is not an exception from challenges from market and competition.

Top 5 challenges are listed below.

- Inventory cost
  - Average lens cost comes out to be ₹135000.
  - If the Rental company wants to keep all the lenses available in market, it will go beyond $100 M.
- Narrow customer segment

- Unlike other rentals products like Car, garments etc, where anyone can be a prospective customer. In Camera rental services, it is mostly the Photography aspirants and professionals. Their population is negligible.

- Services and handling
  - Services and handling charges for these gears are very costly.
- Securing the devices
  - These are very costly devices . It requires at most care and make sure the gears that are rented out are safe. Customer's background verification test is also important to avoid gear loses.
- Pricing
  - Not all the camera gears go out for rent in same frequency and not everything is of same price. Some gears are less expensive with aggressive movement, some are very expensive and little movement. Rental pricing should be customized at the gear level to meet early break even.

Though we have such challenges, the industry is growing as more and more people are interested in Photography. Not only that, Social media and media services over the internet are complementing the camera industry for is growth.

Klachak.com

Klachak.com is a company based out of Chennai, founded by a team of entrepreneurs, who are professionals in their own fields of expertise, yet have a great passion and skill in the art of photography. The company was formed with a vision to make photography accessible to all sincere enthusiasts who would like to create art with light.

It facilitates artists of every level to overcome their barriers to engage with photography, and help inspire creativity and ingenuity in this form of art. It's aim is to provide our discerning clientele services of every kind that is related to the field of photography including education, lens rental, and experiences on field both within and outside the country.

Equipment Rentals

Klachak provides a top of the line service in equipment rentals to assist professionals and amateurs to hire equipment that is usually out of reach financially. Be it lenses, cameras or any other photography related product, Klachak aims to make it accessible at a fair price, while still maintaining the best quality of service. Whether you have a wedding to shoot, a friend's birthday party to capture, or bring home memories of wild animals in the forest.

Photography Services

Through both in-house photographers, as well as through selected professionals in our network, Klachak provides a range of photography services such as educational workshops, commercial shoots, portfolio development, for both individuals, corporates or advertising agencies. We also hold a high-quality stock photography collection, that can be licensed for a wide range of needs.

Problem/Opportunity Statement

"*Primary Objective is to help users in finding the cameras, lenses or any other camera related gadget they would like to rent by predicting similarity score or a list of top 5 or 10 recommended items for the given users based on the transaction history, by using Recommender System Algorithms".* To Reach that Goal, we have to perform very extensive Exploratory Data Analysis, which will produce Key Reports and Dashboards. This also gives opportunity to Perform Cohort Analysis to capture loyal customer and RFM Analysis as well for customer segmentation and base the recommendation on its weightage , It can recommend the Items to users through offline channels also.

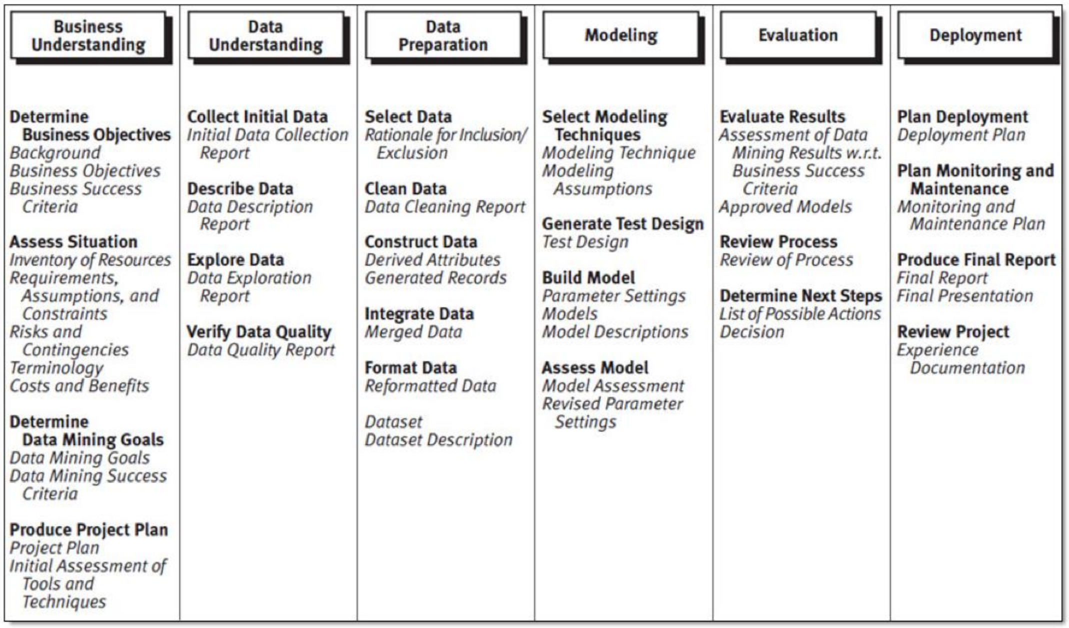
Exploratory reports will further help Klachaks in,

1. Identifying all the factors contributing for more sales (in our case rentals) and revisit the Business Strategy accordingly.
2. How can Klachak take advantage of seasonality in rental activities.

Analysis Methodology

Methodology

This project will follow CRISP-DM model as a framework and methodology.



Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives *Background* *Business Objectives* *Business Success Criteria* **Assess Situation** *Inventory of Resources* *Requirements, Assumptions, and Constraints* *Risks and Contingencies* *Terminology* *Costs and Benefits* **Determine Data Mining Goals** *Data Mining Goals* *Data Mining Success Criteria* **Produce Project Plan** *Project Plan* *Initial Assessment of Tools and Techniques*	**Collect Initial Data** *Initial Data Collection Report* **Describe Data** *Data Description Report* **Explore Data** *Data Exploration Report* **Verify Data Quality** *Data Quality Report*	**Select Data** *Rationale for Inclusion/ Exclusion* **Clean Data** *Data Cleaning Report* **Construct Data** *Derived Attributes* *Generated Records* **Integrate Data** *Merged Data* **Format Data** *Reformatted Data* *Dataset* *Dataset Description*	**Select Modeling Techniques** *Modeling Technique* *Modeling Assumptions* **Generate Test Design** *Test Design* **Build Model** *Parameter Settings* *Models* *Model Descriptions* **Assess Model** *Model Assessment* *Revised Parameter Settings*	**Evaluate Results** *Assessment of Data Mining Results w.r.t. Business Success Criteria* *Approved Models* **Review Process** *Review of Process* **Determine Next Steps** *List of Possible Actions* *Decision*	**Plan Deployment** *Deployment Plan* **Plan Monitoring and Maintenance** *Monitoring and Maintenance Plan* **Produce Final Report** *Final Report* *Final Presentation* **Review Project** *Experience Documentation*

Figure 2

Understanding the business and data are paramount for this project. Once we have data, We will proceed with preparing the data for further Statistical and descriptive analysis. This data preparation will include data cleaning, masking, missing value treatment, removing bias by applying various sampling techniques.,etc., Before even proceeding with the project, initial study of data has been conducted as proof of concept and measure the feasibility.

Data preparation will be very critical. Wherever necessary we will apply transformation logics, like converting categorical variable to set of dummy variables, clubbing to variables to form a new variable, segmenting continuous variable to convert them to categorical variable,., etc., As we have data from 3 data sources, it is important to establish a relationship between them through some key variables.

Once the data set is prepared, Descriptive Statistics will be applied again on the prepared data to understand the data and ensure the data is normal. Descriptive Stats includes studies like, central tendency, frequency distribution, variance study etc., Once the required data is prepared, formatted, it will be subjected to model building based on Recommendation Systems Algorithm. In Our case, we are using a hybrid approach of using content based and Collaborative Filtering based algorithms, Specifically Memory based.

Once the Model is Built, it is evaluated based on Root Mean Square Error Value for its prediction accuracy, Later it will be deployed in Klachak production system.

Data Source

Primary Data Source

All the transaction details and customer information are pulled from Klachak's Primary Database which is in MySQL. The data will be pulled using MySQL Connectors and exported to Excel which will be used for Exploratory Data Analysis, Building RFM, Building Recommendation Systems and Evaluation.

Secondary Data Sources

We will access https://www.dpreview.com website to scrap, customers review and expert review of all lenses and cameras. Which will be used for sentimental analysis, that will later fed to Model for deriving User-Item Ratings.

Data is also pulled from "flickr" database, for capturing equipment usage ranking across globe that will be published to user post recommendation..

Primary Data Source	Secondary Data Source
Sales/Transaction Data from Klachaks	Detailed Product Information from https://www.dpreview.com Detailed Application of Product is obtained from https://www.flickr.com/

Table 2

This data will help us in improving the accuracy of recommendation based on Situation.

Tools & Techniques

We have conducted our analysis using Python 3.6.* in Anaconda Distribution. Data visualization are done using Tableau 13 and few of them in Excel. Transaction information is stitched along with the Klachak's content management system tables and data bases. It was extremely difficult to identify the tables and key columns. Later We have to perform multiple joints and extract meaningful and correct data. On the other hand, dpreview.com does not expose any API's, so we had to perform Webscarping using scarpy in python. Data from flickr is obtained through API call.

Recommendation System Algorithms are memory and process intensive. Especially for predicting and scoring the rating for n items will require N X N Matrix and each cell will have to go through Cosine Distance calculation of 2 vectors. Number items will keep growing along the time, so we need a data base which is scalable. We opted for NoSQL database, MongoDB.

Just not limited to the above tools, we have used following libraries and techniques.

1. Data Integration using Python
2. Data Profiling using Python and Tableau
3. Data Treatment using Python libraries using Scikit learn, numpy and pandas
4. Visualization in python using matplotlib
5. Collaborative Filtering using Pandas and numpy

Limitations

Building a recommendation engine for Klachaks has following limitations.

1. Data Sparsity :

   The reported matrix of user-item ratings is usually very sparse (up to 99%) due to users' lack of knowledge or incentives to rate items. In addition, for the new users or new items, in general, they report or receive only a few or no ratings. Both issues will prevent the CF from providing effective recommendations, because users' preference is hard to extract. Here it is ~97%
2. As the underlying data is transaction based data, probability of a product which is in the inventory for long duration being rented out will be more than the one which is new.
3. Probability of any new product added to the system being picked for recommendation might be low. Which we have tried to address using cold start mechanism. Using Hybrid approach.
4. User Rating for the products are not enabled or effectible used in Klachak's website. whereas Content bases filtering or collaborative filtering uses only user rating for recommendation. As workaround, we have derived the user-item rating using various factors, implicitly. Which is explained in model building section.

5. Model Evaluation: there are very limited techniques available for evaluating the accuracy and performance for Recommender Systems Algorithm.

Literature Review

Recommender Systems

This is how "Joonseok Lee" puts it, Any software system which is actively suggesting an item or a group of items to buy, subscribe, invest or to rent can be referred as a Recommender System. Any campaign or advertisement can also be considered as a recommendation. But, we mainly consider, however, a narrower definition of "personalized" recommendation system that base recommendation on user and item specific information, on Realtime streaming information.

Recommender System algorithm can be broadly can be classified as 2 categories

- Content Based Filtering
- Collaborative Filtering

Collaborative filtering can further be classified as "Memory based" and "Model based" each has its pros and cons and also highly dependent on the type of underlying data, number of users, number of items, how sparse the data is, etc.,

Content Based Filtering:

This is explicitly based on domain knowledge concerning the users and item. Both User profile and item profile plays a key role in this recommendation system. Further, domain knowledge may correspond to user information such as Age, Gender, Occupation, Geographical location etc. In case of Item, it may be like, product features, like, lens type, max ISO, fstops, Shutter Speed, Brand, Price Etc.,

Collaborative Filtering Algorithms:

Collaborative filtering does not use user or item information with the exception of a partially observed rating matrix. The rating matrix holds ratings of items (columns) by users (rows) and is typically binary, for example like vs. do not like, or ordinal, for example, one to five stars recommendation. The rating matrix may also be gathered implicitly based on user activity. Similar to our case.

Hybrid Recommendation System:

Both the Content Based Filtering and Collaborative Filtering Algorithms are combined to form Hybrid Recommendation Systems. Mostly or Generally, we go for Hybrid system if the domain for which the recommender system built is having “Cold Start” problem. We will look into what is a cold start problem in detail in our “Model Development” Section. There are Other Domain Specific reason why people choose Hybrid Recommendation System.

Exploratory Data Analysis

Before dwelling into preparing for development of model, we have to understand the data well. As Klachak does not have any explicit rating mechanism, an alternative method is used to derive best possible rating based on the observations. In order to identify Key fields that can be used for deriving the rating the following study is conducted. This study will also help in understanding the Klachaks business position and its performance.

Sales Yearly Trend.

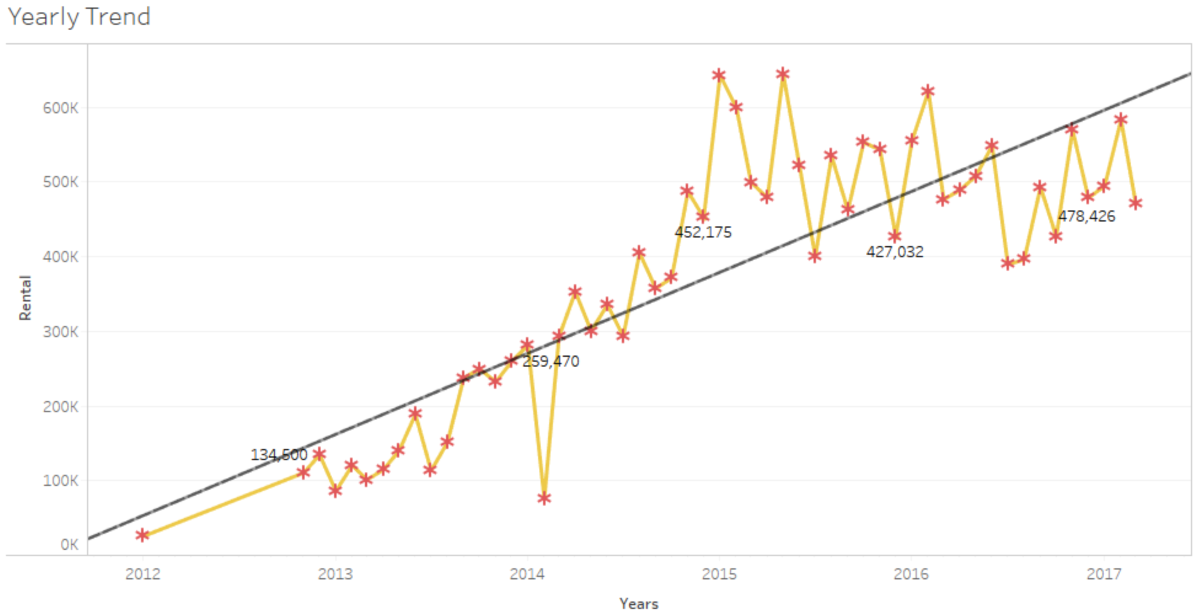


Figure 3

Figure 3, shows the Sales (Revenue from Rentals) over the period of 5 years. We can clearly see that overall sales have increased significantly. But, if we can just look at the past 3 years we can see that the trend is almost horizontal. Considering if this trend continues, this will have an impact on profit and it becomes very important for the company to find new customers, hold on to existing loyal customer and also where ever possible, perform up-sell or cross-rent the product.

Seasonal Trend

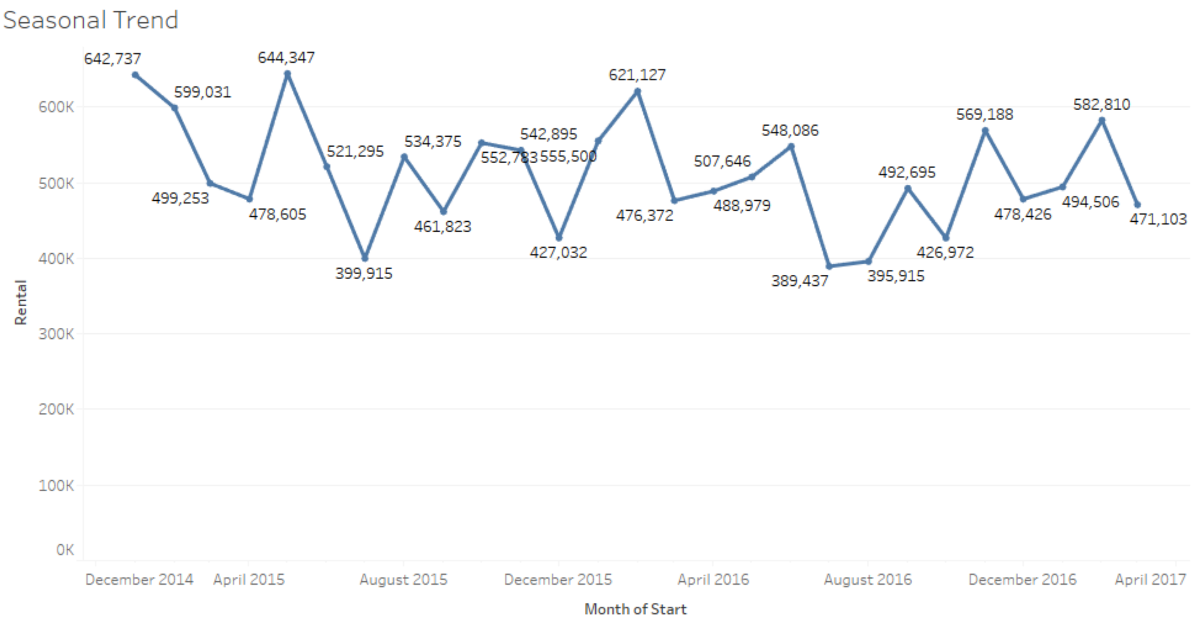


Figure 4

Inference: Three seasonal peaks per year in the months as January/February, July/August and October/November.

Percentage of Revenue

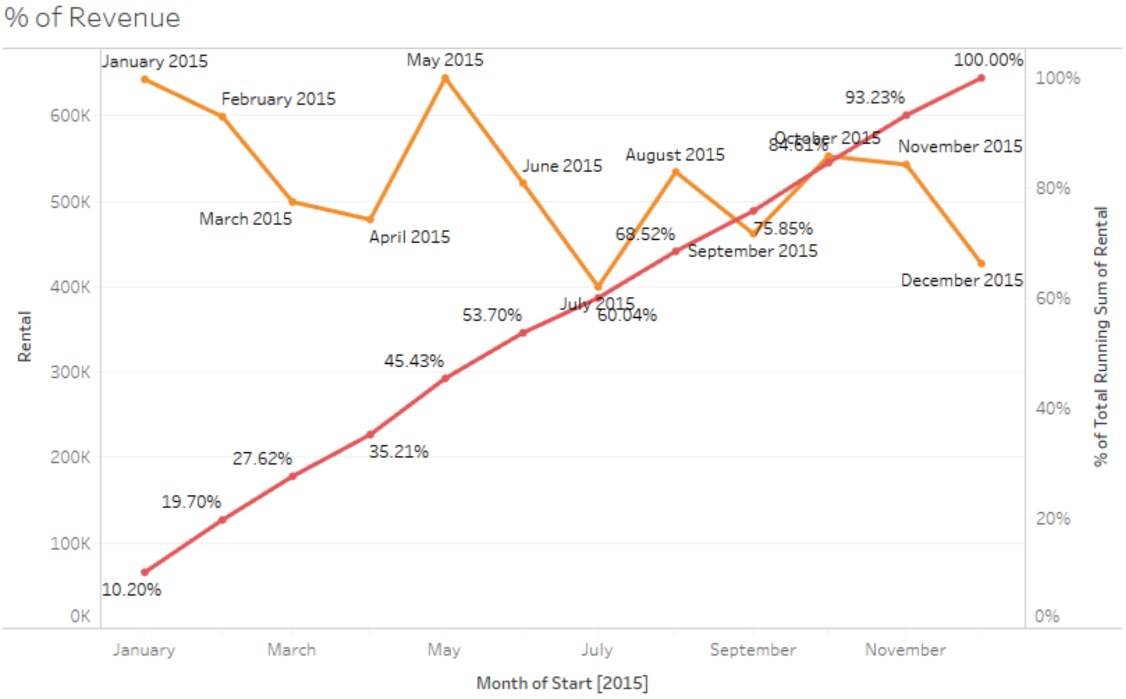


Figure 5

Inference: For the Year 2015, 60% of revenue achieved in first six months. This is a pattern that can be observed year over year. This correlates very well with previous observation on seasonality. There are 2 seasons before third Quarter of every year, these contributes heavily on sales.

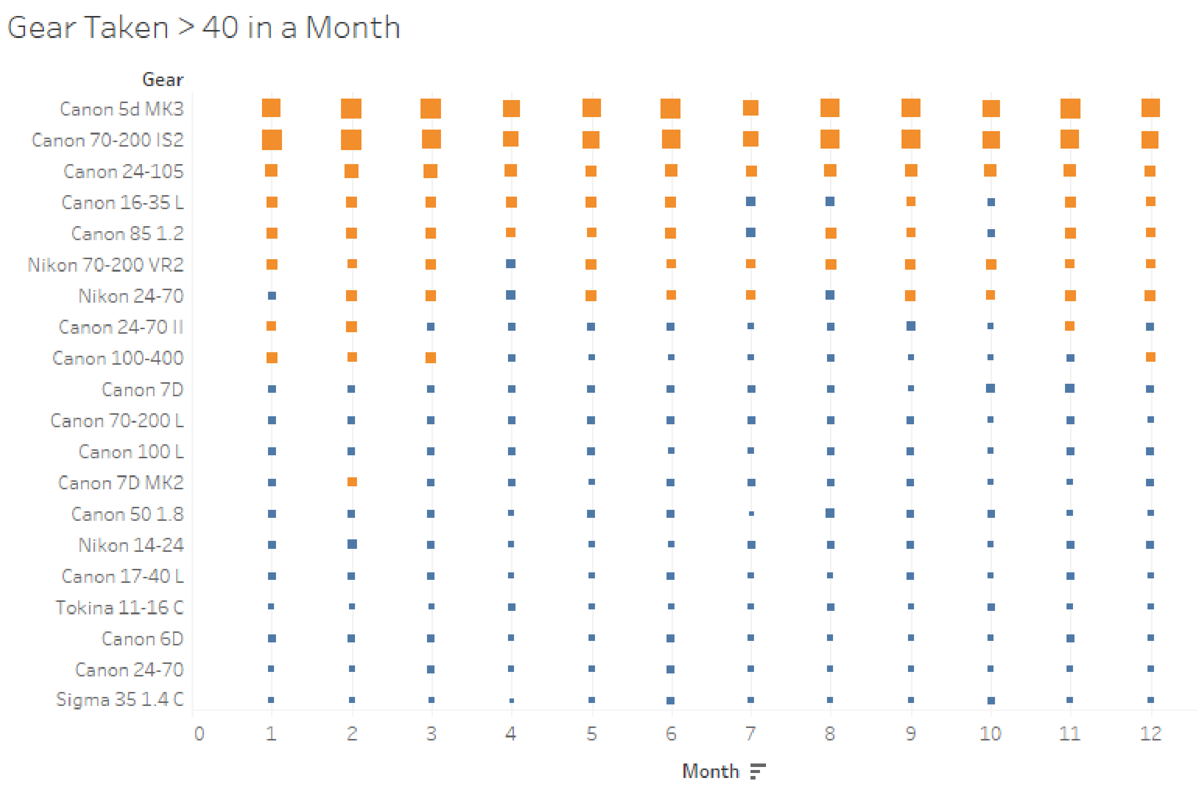


Figure 6

Seasonality and type of cameras rented confirms that most of the transactions were related to marriage events or other festivals. Also, Canon is contributing more for sales. Top5 are Canon Brand.

<u>Recommendation</u>:

1. Replace cameras rented very few times like Nikon 200-400 VR2, Nikon 60 F4, Canon 1D MK4, etc. (extensive list provided separately) with frequently rented cameras like Cannon 5d MK3, Canon 70-200 IS2, Canon 24-105 (preferred mostly for marriage events) to increase revenue. This purely based on the Recommender that is to be built which will also consider the purpose of the rent.
2. Promotional exercise in the later part of the year between August to December will increase revenue during off-season

Cohort Analysis

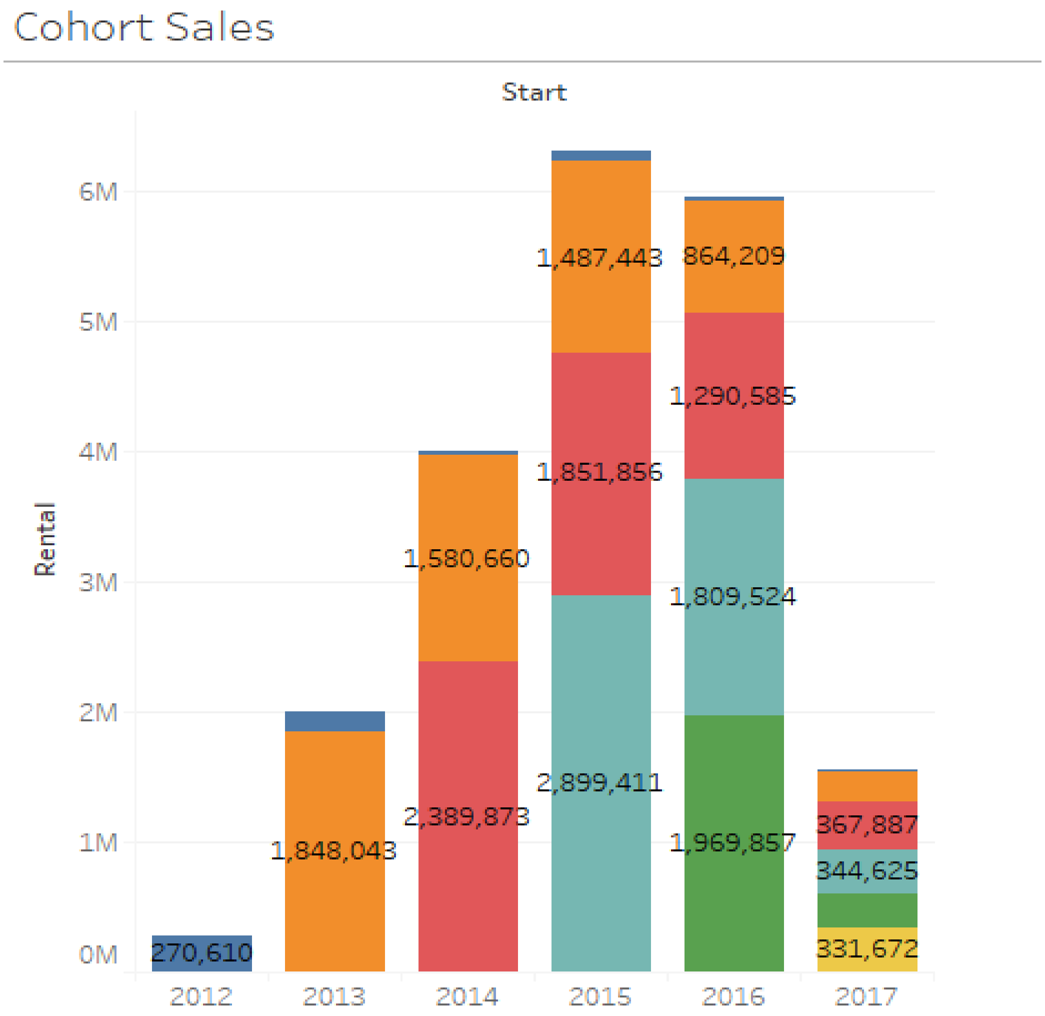


Figure 7

Inference: Customer who rented in 2013 or 2014 rented frequently and they are most loyal ones, cohort on loyalty will explain more.

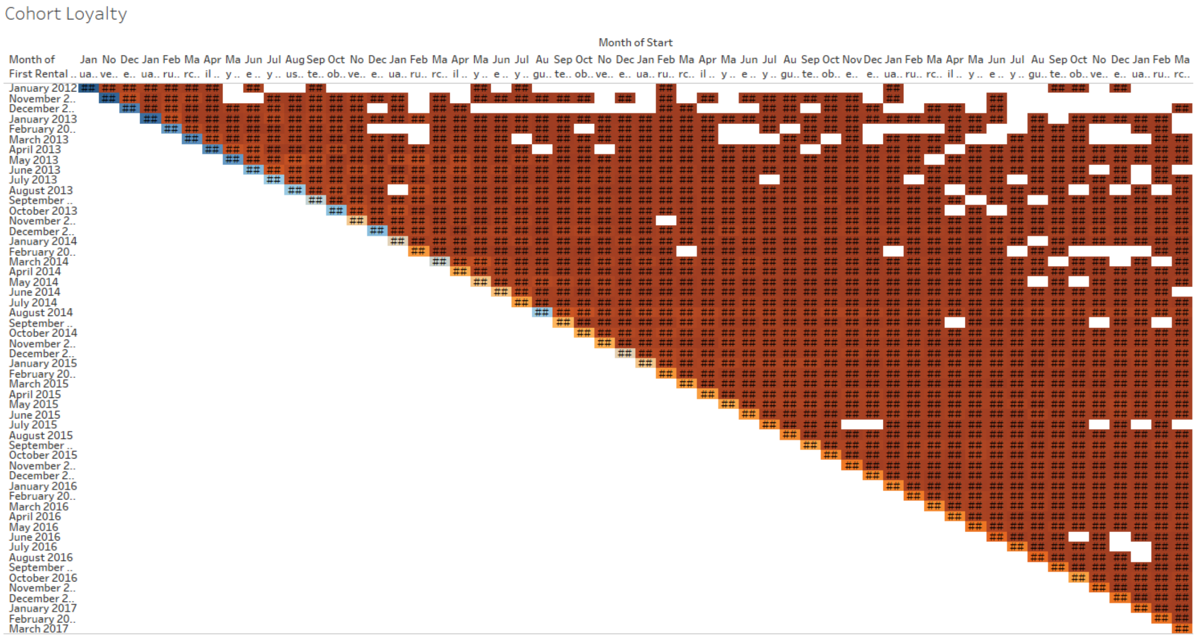


Figure 8

Inference: Very few new customers are renting in the last two years and high share of revenue generated by customers from initial years. Does that mean, people who become customers of Klachaks always stays as customer because of the value they see in it? It can be true, what can also be true is, Kalachaks is a near to monopoly in Chennai when it comes to Camera Rentals.

Recommendation

Recommending any advertisement/promotional exercises to bring new customers which is not done till now by Klachak. Also, Klachacks needs to be proactive and focus on retaining its customer base and also aggressively look for adding new customers to it Loyalty list.

Cohort and Inventory flow visualization helps us to understand the strength of our customer base as well as fast running product. We will basically use customer loyalty score as well as product inventory performance for deriving the initial rating of user-item.

This can be further enhanced by performing RFM analysis, Segment the Users and assign weightage for deriving the rating based on segmentation. Before getting into RFM, we will prepare the data, understand the model and understand how RFM contributes for Collaborative filtering model.

Model Development

Developing the model for collaborative filtering involves 3 Stage process as listed below.

1. Data Extraction and Data Preparation
2. Model Building and Evaluation
3. Deploying in Live system.

Data Extraction and Data Preparation.

Input for this model comes from 2 different data sources. Primarily one from Klachaks Native Data base and other from a website dpreview. For building content based filtering we need features of the product, which will be obtained from dpreiew . But, there are limitations with this approach, because Camera Body has a separate set of features versus lenses. For, Content based filtering all the attribute type should be same across all the products. We have added the solution for this in our future scope, bit for now, we will proceed with Product Name similarity for Content based filtering.

As it was mentioned earlier that klachak.com does not capture the user rating for each item, we have to derive those as Collaborative filtering is designed based on user rating. Apart from user rating, collaborative filtering algorithm just needs only the user id/name and product id/name, which is directly pulled form Klachak's database.

Based on the RFM Score and number of time a particular product is rented by the same customer, probability weightage of rating scale will be applied to randomly derive the rating. Rating will be of scale 1 to 5, 1 being low an d5 being high,

How RFM value is segmented.

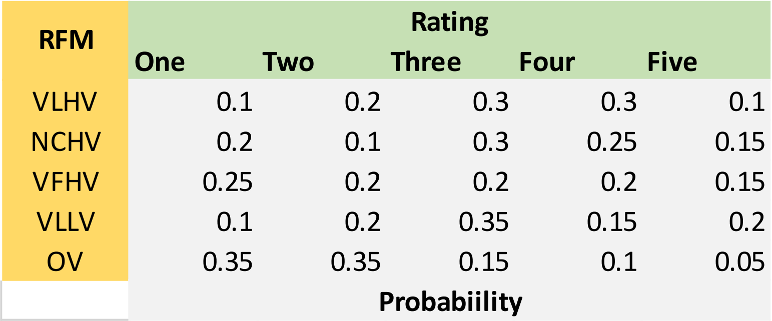
RFM Score	Segmentation Type
444,445,455,555,554,544	Very Loyal and High Value Customers (VLHV)
4*5,5*4	New Customers and High Value (NCHV)
*45,*54	Very Frequent and High Value Customers (VFHV)
441,442,443,451,452,453,551,552,553,541,542,543	Very Loyal and Low Value Customers (VLLV)
Rest	Onetime Visitors.

Table 3

Based on the RFM Segmentation Type and Number of times a user has rented a particular product we will assign probability and the rating randomly

Mapping Cheat Sheet got Rating Derivation

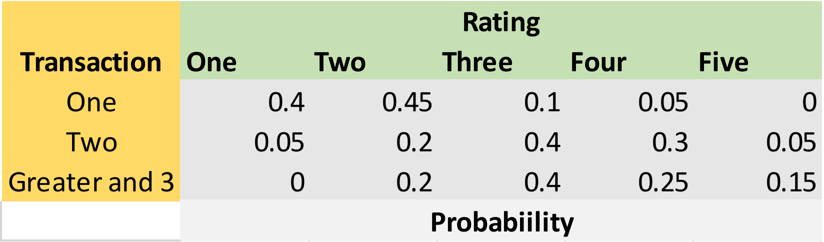
RFM Based Rating Weighted Probability



RFM	Rating				
	One	Two	Three	Four	Five
VLHV	0.1	0.2	0.3	0.3	0.1
NCHV	0.2	0.1	0.3	0.25	0.15
VFHV	0.25	0.2	0.2	0.2	0.15
VLLV	0.1	0.2	0.35	0.15	0.2
OV	0.35	0.35	0.15	0.1	0.05
	Probabiility				

Table 4

Transaction Count Based Weighted Probability



	Rating				
Transaction	One	Two	Three	Four	Five
One	0.4	0.45	0.1	0.05	0
Two	0.05	0.2	0.4	0.3	0.05
Greater and 3	0	0.2	0.4	0.25	0.15
	Probabiility				

Table 5

These probability weights are provided by Kolchak after discussion. We have taken the average for each product and customer.

Model Building and Evaluation

Why are we putting hybrid approach of recommendation system instead of specifically content based or collaborative filter based? We will come to that shortly after explaining the how model works.

Collaborative Filtering [CF]

In Badrul Sarwar's words "The goal of a collaborative filtering algorithm is to suggest new items or to predict the utility of a certain item for a particular user based on the user's previous likings and the opinions of other like-minded users. In a typical CF scenario, there is a list of m users U = {u1, u2, . . . , um} and a list of n items I = {i1, i2, . . . , in}. Each user U_i has a list of items I_{ui} , which the user has expressed his/her opinions about. Opinions can be explicitly given by the user as a rating score, generally within a certain numerical scale, or can be implicitly derived from purchase records"

There exists a distinguished user $u_a \in U$ called the active user for whom the task of a collaborative filtering algorithm is to find an item likeliness that can be of two forms, Prediction and Recommendation.

Prediction is a numerical value, Pa, j , expressing the predicted likeliness of item $i_j \notin I_{ua}$ for the active user U_a. This predicted value is within the same scale (e.g., from 1 to 5) as the opinion values provided by U_a.

Recommendation is a list of N items, Ir ⊂ I, that the active user will like the most. Note that the recommended list must be on items not already purchased by the active user, i.e $I_r \cap I_{u_a} = \emptyset$ This interface of CF algorithms is also known as Top-N recommendation.

Though there are 2 types of algorithm for CF, one is user based and the other is item based, given the former one has limitations like scalability and sparsity issues, we are using item based Collaborative Filtering.

Item Based Collaborative Filtering

Item-based approach looks into the set of items the target user has rated and computes how similar they are to the target item i and then selects k most similar items {i1, i2, . . . , ik }. At the same time their corresponding similarities {si1, si2, . . . , sik } are also computed. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. We describe these two aspects namely,

the similarity computation

and, the prediction generation in details here.

Before getting to the next step, lets understand our data, We do not have User Opinion or rating for each item, we only have following information, with that we will implicitly derive the Rating for each product each user.

Here we use the sales transaction data obtained from Klachaks DB.

Variable	Type	Description
ProductID	Numeric	Unique identifier of the product
ProductName	String	Name of the Product

Cust_Id	Numeric	Unique Identifier of the Customer
Customer Name	String	Name of the customer

Table 6

First Input for Deriving the Rating – Sales Data

Aggregate the Transaction Count bases on Customer and Product. This will result in number of times a customer rented the given product. Based on the count we can randomly assign the rating in way that, if the count is more give more weightage for Rating Score 4 and 5, if the count is one or two, flip the weightage for rating 1 and 2, any number between the maximum and minimum count adjust the Probability weight of random number relatively.

Second Input for Deriving the Rating – RFM Segments

To make the Rating Derivation more accurate, We will use the RFM Segment information also clubbed with the Sales data in assigning the probability weightage for Rating Score. Please refer figure 10 to understand more.

Third Input for Deriving the Rating – Sentiment Score

To make it more precise, This system will perform web scraping of dpreview and obtain the Sentimental score. Based on the Sentiment Score this rating will be again adjusted. Then we arrive at the final rating value from Scale 1 to 5 for each item that user has rented.

Sparsity Test

There are 3 Things that will affect the Prediction and Recommendation accuracy of CF algorithm.

1. Number of Customers
2. Number of Products
3. Rating Sparsity.

We have around 2118 Unique Customers, with 176 unique products to choose from. In other words we have 372, 768 combination of possible user item ratings, of which we have 12982 combinations rated. Which means, out of all possible ways of capturing rating by every user for every product, we have only 3.49% rated or ratings derived. We still have to Predict the Rating for remaining 96.51% of combinations or in other words, 359,786 ratings needs be predicted.

Generally 98% Sparsity is considered sufficient for Building Recommendation System, we have 96.51%, which holds good for us to proceed.

When looking at Rating sparsity of each product, which means, 2118 Unique customers, how many of them rated for each product, we get the following graphs.

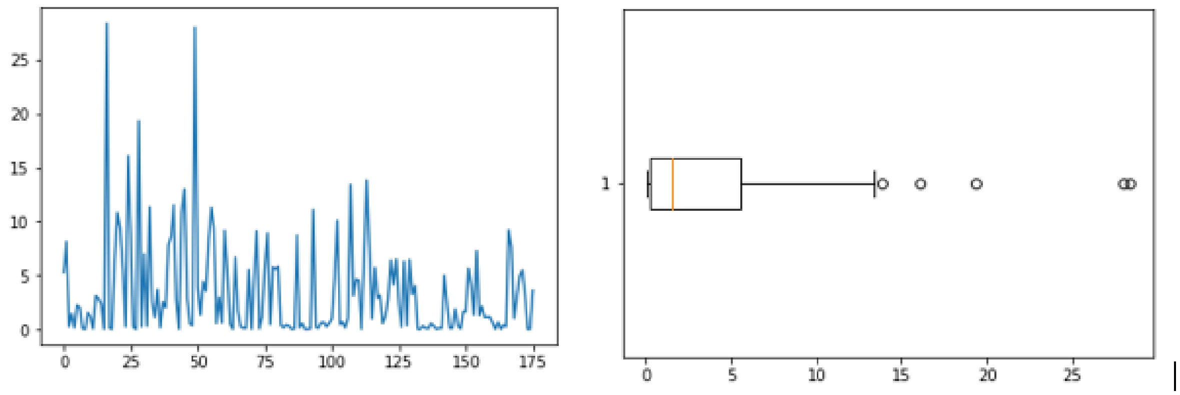
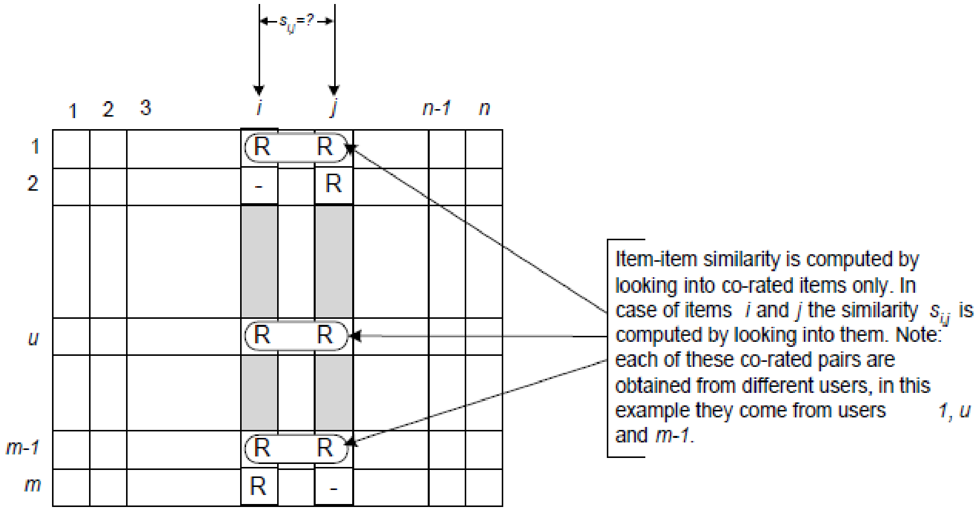


Figure 9

Nearly 28% of Products have dense Ratings, while remaining 72 percentage very weak density with respect to rating. This might impact the similarity score calculation, to overcome it, we are centering the data based on Item rating average.

Item Similarity Computation

One critical step in the item-based collaborative filtering algorithm is to compute the similarity between items and then to select the most similar items. The basic idea in similarity computation between two items i and j is to first isolate the users who have rated both of these items and then to apply a similarity computation technique to determine the similarity si, j . Figure 2 illustrates this process, here the matrix rows represent users and the columns represent items. There are a number of different ways to compute the similarity between items. Here we present three such methods. These are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity.



Item-item similarity is computed by looking into co-rated items only. In case of items *i* and *j* the similarity $s_{i,j}$ is computed by looking into them. Note: each of these co-rated pairs are obtained from different users, in this example they come from users *1*, *u* and *m-1*.

Figure 10

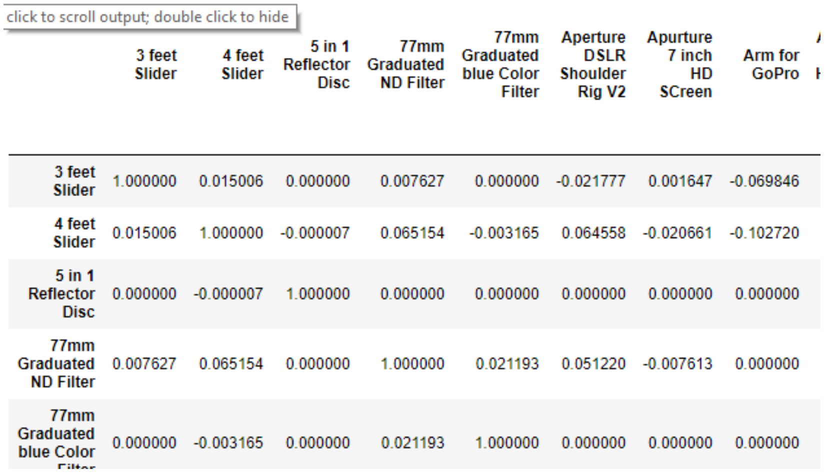
In our case we have taken "adjusted-cosine similarity" for calculating the Similarity Score,

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_u)^2}}.$$

Here $\bar{R}_u$ is the average of the u-th user's ratings.

Equation 1

Sample Output from Item-Item Similarity Matrix



click to scroll output; double click to hide

	3 feet Slider	4 feet Slider	5 in 1 Reflector Disc	77mm Graduated ND Filter	77mm Graduated blue Color Filter	Aperture DSLR Shoulder Rig V2	Apurture 7 inch HD SCreen	Arm for GoPro
3 feet Slider	1.000000	0.015006	0.000000	0.007627	0.000000	-0.021777	0.001647	-0.069846
4 feet Slider	0.015006	1.000000	-0.000007	0.065154	-0.003165	0.064558	-0.020661	-0.102720
5 in 1 Reflector Disc	0.000000	-0.000007	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
77mm Graduated ND Filter	0.007627	0.065154	0.000000	1.000000	0.021193	0.051220	-0.007613	0.000000
77mm Graduated blue Color Filter	0.000000	-0.003165	0.000000	0.021193	1.000000	0.000000	0.000000	0.000000

Figure 11

Prediction Generation

There are couple of ways we can generate the Rating prediction for those remaining 359,786 possible combinations,

1. Using Weighted Sum
2. Using Regression.

This project used Weighted Sum to Predict the Ratings.

This method computes the prediction on an item i for a user u by computing the sum of the ratings given by the user on the items similar to i . Each ratings is weighted by the corresponding similarity s i, j between items i and j

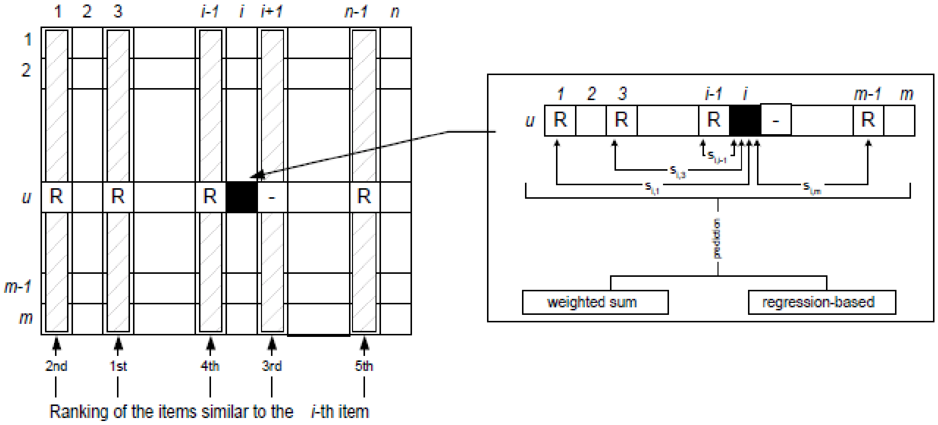


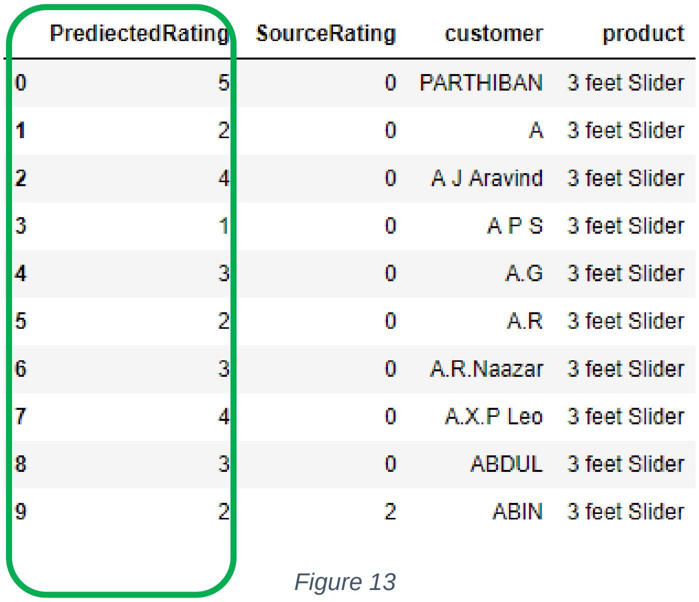
Figure 12

Formula is,

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}} (s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}} (|s_{i,N}|)}$$

Equation 2

Predicted Output (Sample)



	PrediectedRating	SourceRating	customer	product
0	5	0	PARTHIBAN	3 feet Slider
1	2	0	A	3 feet Slider
2	4	0	A J Aravind	3 feet Slider
3	1	0	A P S	3 feet Slider
4	3	0	A.G	3 feet Slider
5	2	0	A.R	3 feet Slider
6	3	0	A.R.Naazar	3 feet Slider
7	4	0	A.X.P Leo	3 feet Slider
8	3	0	ABDUL	3 feet Slider
9	2	2	ABIN	3 feet Slider

Figure 13

Content Based Filtering

When Content-Based Approach Make Sense?

But sometimes CF isn't a viable option; let's say we want to make recommendations to a customer viewing a product details page, who just got there from a Google SERP link. We don't know anything about this customer, so we can't build a matrix of purchases. But we can use a content-based system to recommend similar products. In that sense, content-based recommenders can solve the "**cold-start**" problem that CF systems have. To put it in simple terms, Collaborative Filtering is not helping much in case of New Products or New Users. Because both will not have history.

They can also provide a measure of automated curation when you have a strong indication of buying intent for a particular product (like when a lead comes from a Google search for a term related to that product). If you're interested in the Canon EF 70-200mm f/2.8 L USM, you might also love the Nikon 70-200mm f/2.8G AF-S IF-ED VR. A content-based engine performs better at picking up related products like this, without a bunch of manual curation (those products don't appear together in Google search results).

As part of data preparation, the abbreviated Product Name from company is matched with dpreview database and Standard name was extracted, which then used for further processing. Original Product Name in company's db will be like 'Canon 70-200 IS2' and it's equivalent Standard product name is 'Canon EF 70-200mm f/2.8L IS USM'

There is something which needs to be noted here, Generally, in any content based approach, the model looks for feature/attribute similarity to match the closeness of the product. For instance, if we take the garments, it has many different classification and each will have its own attribute, like, Kids Clothing, Women, Men and each of the above will have attributes like, brand, color, size, age, type, fabric etc., based on attributes similarity we find the close match of the product.

Our original dataset from Company does not have Product and Attributes separately, instead it has only the Product Name formatted in its own standard. After converting to the Standard name, we use "Only Product Name" for finding similarly, instead of attributes as mentioned in garments example. That is because, our Product Name itself has all the attribute values embed in it. Canon EF 70-200mm f/2.8L IS USM,

Canon EF 70-200mm f/2.8L IS USM					
Brand	***Type***	***Focal Length***	***f-stop***	***Image Stabilization [IS]***	***UltraSonic Motors (USM)***
Canon	EF	70-200mm	2.8L	Yes	Yes

Table 7

Building Content Based Approach

We're going to use a simple Natural Language Processing technique called TF-IDF (Term Frequency - Inverse Document Frequency) to parse through the descriptions, identify distinct phrases in each item's description, and then find 'similar' products based on those phrases.

Concept of TF-IDF

TF-IDF stands for Term Frequency times Inverse document frequency.TF stands for Term Frequency. It tells us about how often does the term you are talking about appear in the document? How relevant is it to the document? For e.g. how many times the keyword "tripod" appeared in the documents OR how many times a tag is applied for a product name.

IDF stands for Inverse Document Frequency. It tells us how rare it is for a document to have this term or for a tag to be applied to the product name. We calculate it by taking the inverse of how many documents have this tag divided by total number of documents. What it will do is that if a term appears in a large number of documents, it will provide us with a low IDF value.

TF-IDF works by looking at all one, two, and three-word phrases (uni-, bi-, and tri-grams to NLP folks) that appear multiple times in a product name(the "term frequency") and divides them by the number of times those same phrases appear in all product names. So terms that are 'more distinct' to a particular product ("16-35mm" in item) get a higher score, and terms that appear often, but also appear often in other products ("canon" /"Nikon") get a lower score.

Once we have the TF-IDF terms and scores for each product, we'll use a measurement called cosine similarity to identify which products are 'closest' to each other.

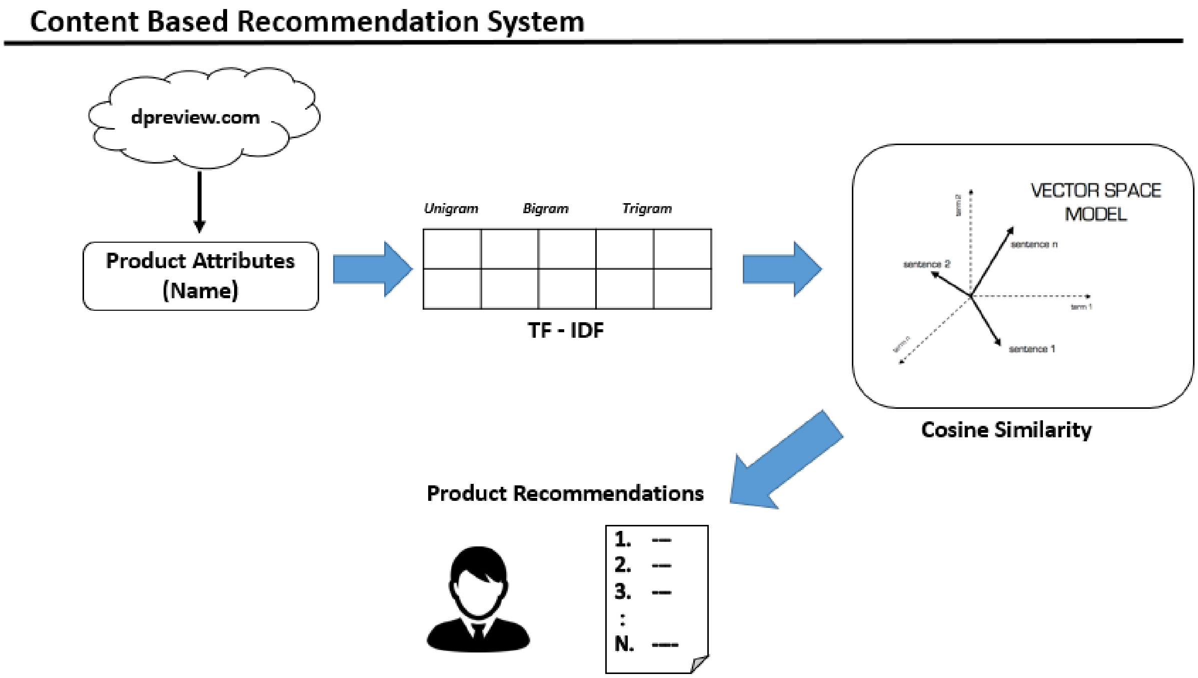


Figure 14

The Cosine Similarity

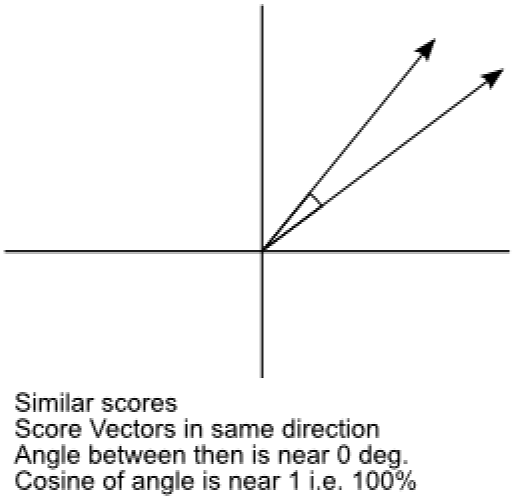
The cosine similarity between two vectors (or two documents on the Vector Space) is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude, it can be seen as a comparison between documents on a normalized space because we’re not taking into the consideration only the magnitude of each word count (tf-idf) of each document, but the angle between the documents. What we have to do to build the cosine similarity equation is to solve the equation of the dot product for the $\cos\theta$:

$$\vec{a} \cdot \vec{b} = \|\vec{a}\|\|\vec{b}\| \cos\theta$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

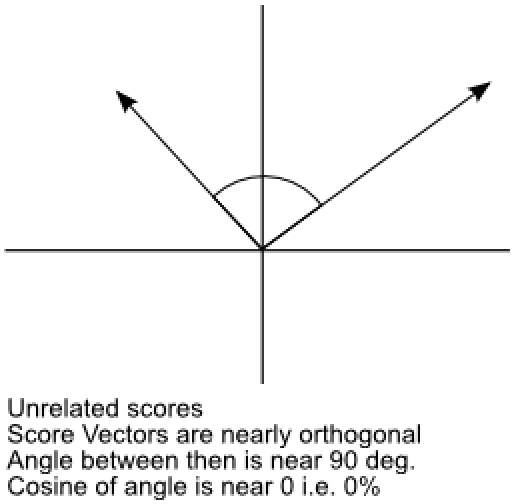
Equation 3

And that is it, this is the cosine similarity formula. Cosine Similarity will generate a metric that says how related are two documents by looking at the angle instead of magnitude, like in the examples below:



Similar scores
Score Vectors in same direction
Angle between then is near 0 deg.
Cosine of angle is near 1 i.e. 100%

Figure 15



Unrelated scores
Score Vectors are nearly orthogonal
Angle between then is near 90 deg.
Cosine of angle is near 0 i.e. 0%

Figure 16

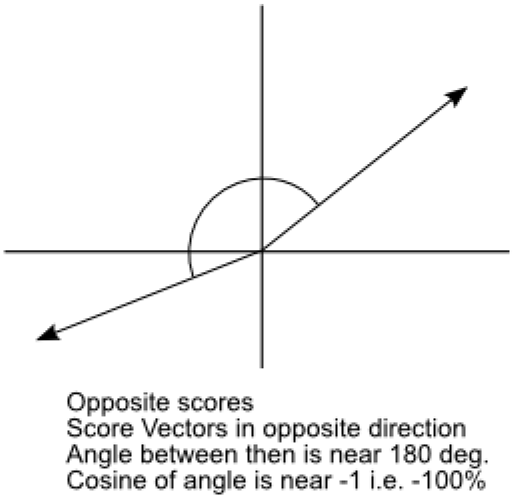

Opposite scores
Score Vectors in opposite direction
Angle between then is near 180 deg.
Cosine of angle is near -1 i.e. -100%

Figure 17

The Cosine Similarity values for different documents, 1 (same direction), 0 (90 deg.), -1 (opposite directions).

Note that even if we had a vector pointing to a point far from another vector, they still could have an small angle and that is the central point on the use of Cosine Similarity, the measurement tends to ignore the higher term count on documents.

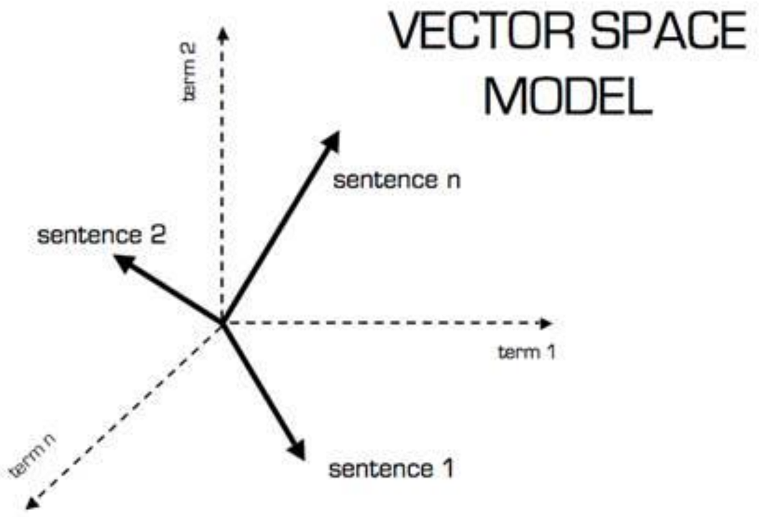


Figure 18

In this case, Python's SciKit Learn has both a TF-IDF and cosine similarity implementation.

From the list of products identified as "closest" K number of products are recommended.

Why Hybrid Approach?

Going back to the Question of Why Hybrid approach and why not either of them, is because, of the following g reason.

As not every product is having equal number of transaction, we might not be able to recommend similar product for those products as there won't be any intersection of ratings, also, any new product will not be entering into the systems as it will not have any transactions or ratings, basically a "Cold Start" Problem.

Hence for such Products that have no transaction or only one transaction we are using content based filtering as it will match based on feature similarity and not rating similarity. But this will have a redundancy effect. The content-based recommendation engine will only recommend products related to predefined categories and may never recommend anything in other categories as the user never purchased those products before. There will be no surprise factor.

Hybrid System of recommendation engine is built by combining various recommender systems to build a more robust system. By combining various recommender systems, we can replace the disadvantages of one system with the advantages of another system and thus build a more robust system. For example, by combining collaborative filtering methods, where the model fails when new items don't have ratings, with content-based systems, where feature information about the items is available, new items can be recommended more accurately and efficiently. Hence, we address the 'Cold Start' problem as well as "No Surprise" problem.

There is also advantage of hybrid recommendation engines, this approach will increase the efficiency of recommendations compared to the individual recommendation techniques. This approach also suggests a good mix of recommendations to the users, both at the personalized level and at the neighborhood level

Weighted method

In this method the final recommendations would be the combination, mostly linear, of recommendation results of all the available recommendation engines. At the beginning of the deployment of this weighted hybrid recommendation engine, equal weights will be given to each of the results from available recommendation engines, and gradually the weights will be adjusted by evaluating the responses from the users to recommendations. In our case, we have Content based and Collaborative filtering. The weight for Content based is 0.7 and collaborative is 0.3. The reason for this is, if we can go back to Cohort analysis, we see that majority of revenue is coming from repeating customers or loyal customers. Those customers have tendency to rent same product during every visit, for them the recommendation engine will offer products similar to one they usually rent, this achieved using Content Based Filtering and hence it's wright is more. The System will also offer them some products that are liked by others and still can be of customers interest, which CF takes care of.

Performance measurement

Precision and recall are used to measure the performance.

In recommender systems, for the final the user the most important result is to receive an ordered list of recommendations, from best to worst. In fact, in some cases the user doesn't care much about the exact ordering of the list - a set of few good recommendations is fine. Taking this fact into evaluation of our recommender systems, we could apply classic information retrieval metrics to evaluate: Precision and Recall. These metrics are widely used on information retrieving scenario and applied to domains such as search engines, which return some set of best results for a query out of many possible results.

We could say that the 'Precision' is the proportion of top recommendations that are relevant, considering some definition of relevant for your problem domain, in our case similarity and user likeliness. So if we say 'Precision at 10' would be this proportion judged from the top 10 results. The 'Recall' would measure the proportion of all relevant results included in the top results.

The precision is the proportion of recommendations that are good recommendations, and recall is the proportion of good recommendations that appear in top recommendations.

$$\text{Precision} = \frac{|originaly\ rated\ products| \cap |to\ recomend\ products|}{|to\ recomend\ products|}$$

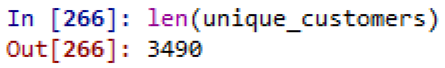
Equation 4

$$\text{Recall} \quad = \frac{|originaly\ rated\ products| \cap |to\ recomend\ products|}{|originally\ rated\ products|}$$

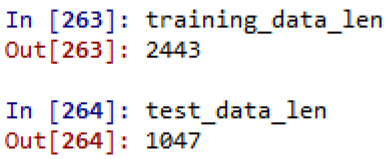
Equation 5

Data Partitioning to test model fiteness

As stated above evaluating the Recommendation system is very complex and we do not have one standard method. We have approached in a classic way of dividing the data as training and test data and validate the model fitness. We have divided the sales transaction data with respect to user in the ratio of 70:30. 70% as Training and 30% Test.



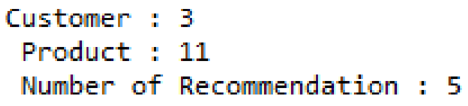
```
In [266]: len(unique_customers)
Out[266]: 3490
```



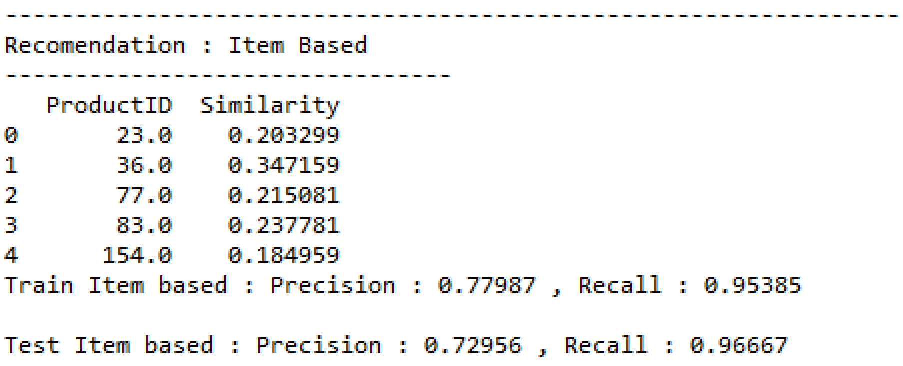
```
In [263]: training_data_len
Out[263]: 2443

In [264]: test_data_len
Out[264]: 1047
```

Sample Output

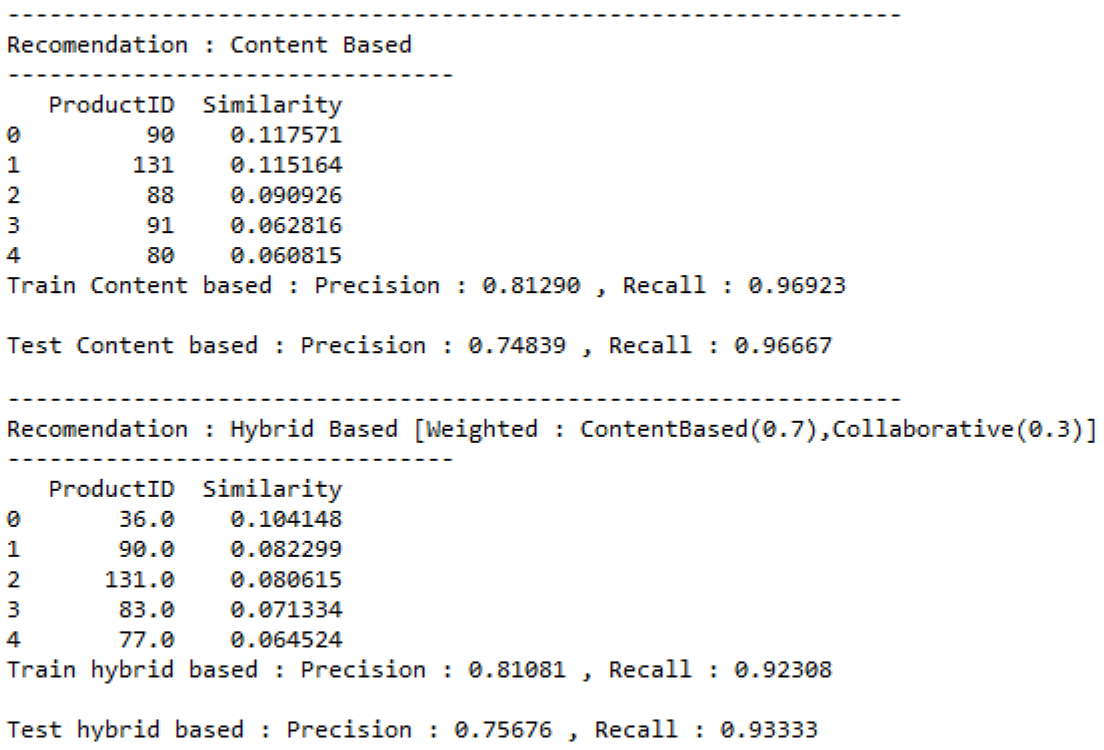


```
Customer : 3
 Product : 11
 Number of Recommendation : 5
```



```
-------------------------------------------------------------
Recomendation : Item Based
-------------------------------
   ProductID  Similarity
0       23.0    0.203299
1       36.0    0.347159
2       77.0    0.215081
3       83.0    0.237781
4      154.0    0.184959
Train Item based : Precision : 0.77987 , Recall : 0.95385

Test Item based : Precision : 0.72956 , Recall : 0.96667
```



```
-----------------------------------------------------------------
Recomendation : Content Based
--------------------------------
   ProductID  Similarity
0         90    0.117571
1        131    0.115164
2         88    0.090926
3         91    0.062816
4         80    0.060815
Train Content based : Precision : 0.81290 , Recall : 0.96923

Test Content based : Precision : 0.74839 , Recall : 0.96667

-----------------------------------------------------------------
Recomendation : Hybrid Based [Weighted : ContentBased(0.7),Collaborative(0.3)]
--------------------------------
   ProductID  Similarity
0       36.0    0.104148
1       90.0    0.082299
2      131.0    0.080615
3       83.0    0.071334
4       77.0    0.064524
Train hybrid based : Precision : 0.81081 , Recall : 0.92308

Test hybrid based : Precision : 0.75676 , Recall : 0.93333
```

Results for Klachaks Recommender System

Collaborative Filtering	***Precision***	***Recall***
Train	78%	95%
Test	73%	96%

Table 8

We can see that Recall is 95% nut precision is only 78% in training, which tells that, though we have a high proportion of good recommendation appear at the top of the recondition, the precision is only 78% which pulls down the overall recommendation performance. When we are looking at the model fitness, Results between training and test are close. And It is generally accepted to be a good recommendation if precision is more than 70%.

Content Filtering	***Precision***	***Recall***
Train	81%	96%
Test	74%	96%

Table 9

Though the Precision level has increased but it is still not significant. If we can have Precision and Recall more or less equal, we can say our model is a good fit. Precision of both training and test data are more than 70%, which holds good.

Hybrid Filtering	***Precision***	***Recall***
Train	81%	92%
Test	77%	93%

Table 10

This is good. We have Precision of 81% and recall of 92%, overall recommendation. Even model fits well between Train and Test, with close Precision value and more than 70%. Even when we see among all three, Hybrid Stands out better.

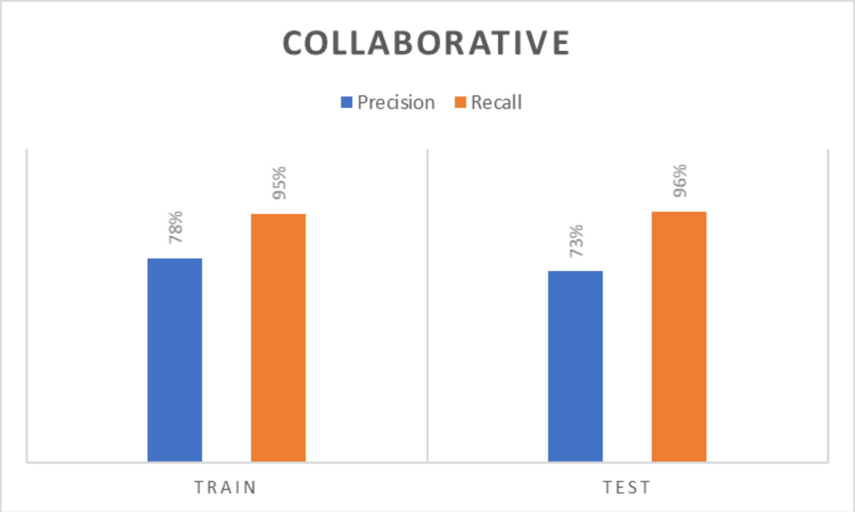


Table 11

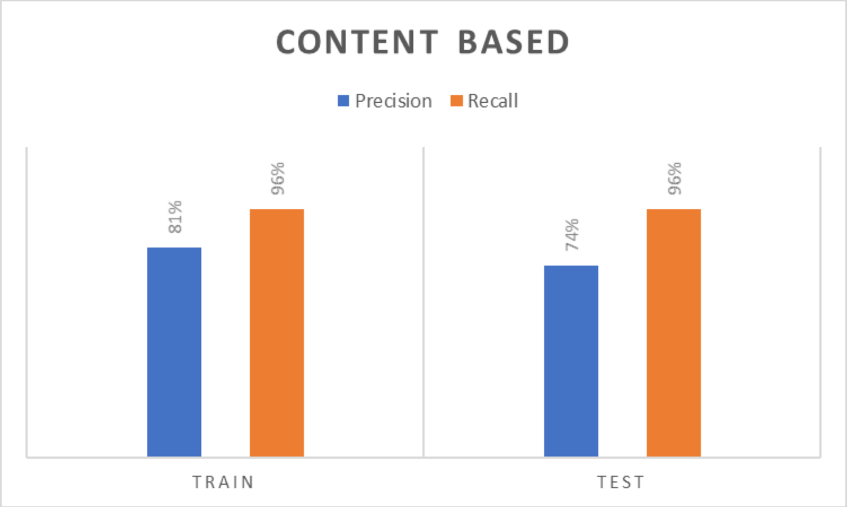


Table 12

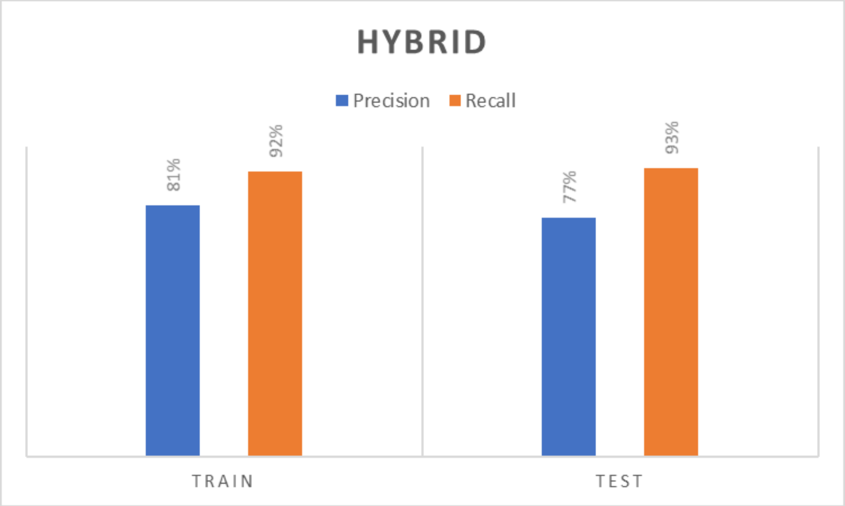


Table 13

Sentiment Analysis

User reviews posts for each product are web scrapped from dpreview.com. Collected the review text for each product then they are processed as individual documents.

Data Cleansing Steps

- Stripping of HTML tags
- Removal of special characters

After data cleansing, sentiment scores are calculated for each document using NLTK library's VADER lexicon. VADER stands for **V**alence **A**ware **D**ictionary and s**E**ntiment **R**easoner. It is a lexicon with a rule-based sentiment analysis framework that is specially built for analyzing sentiment from social media resources. It contains all the necessary sentiment scores associated with various terms, including words, emoticons, and even slang language based tokens. Each feature was rated on a scale from "[-4] Extremely Negative" to "[4] Extremely Positive" , with allowance for "[0] Neutral".

Our interest is to show how much negative feedback the product has received and hence after calculating sentiment scores for each document, the average negative % is taken for all and displayed while being recommended.

One constraint here is that all products in Klachak's are not having reviews in dpreview.com. As a future scope, we need to consider other sites for reviews.

Web Scrapping – Using Scrapy

Klachak inventory sources very limited products metadata information. Hence, we scrapped dprewview.com for full product specifications, attribute scorings, products reviews by dpreview.com and users. Scrapped Product specifications and attribute scorings were used for building recommendation model and reviews are converted into scored using sentiment analysis.

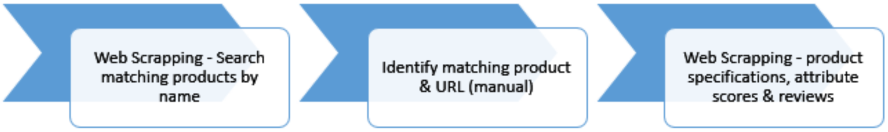

Web Scrapping - Search matching products by name
Identify matching product & URL (manual)
Web Scrapping - product specifications, attribute scores & reviews

Figure 19

Scrapy Data Flow

Data flow in Scrapy is controlled by the execution engine, and the steps are

1. The Engine gets the initial Requests to crawl from the Spider.
2. The Engine schedules the Requests in the Scheduler and asks for the next Requests to crawl.
3. The Scheduler returns the next Requests to the Engine.
4. The Engine sends the Requests to the Downloader, passing through the Downloader Middlewares.
5. Once the page finishes downloading, the Downloader generates a Response (with that page) and sends it to the Engine, passing through the Downloader Middlewares.
6. The Engine receives the Response from the Downloader and sends it to the Spider for processing, passing through the Spider Middleware.
7. The Spider processes the Response and returns scraped items and new Requests (to follow) to the Engine, passing through the Spider Middleware.
8. The Engine sends processed items to Item Pipelines, then send processed Requests to the Scheduler and asks for possible next Requests to crawl.
9. The process repeats (from step 1) until there are no more requests from the Scheduler.

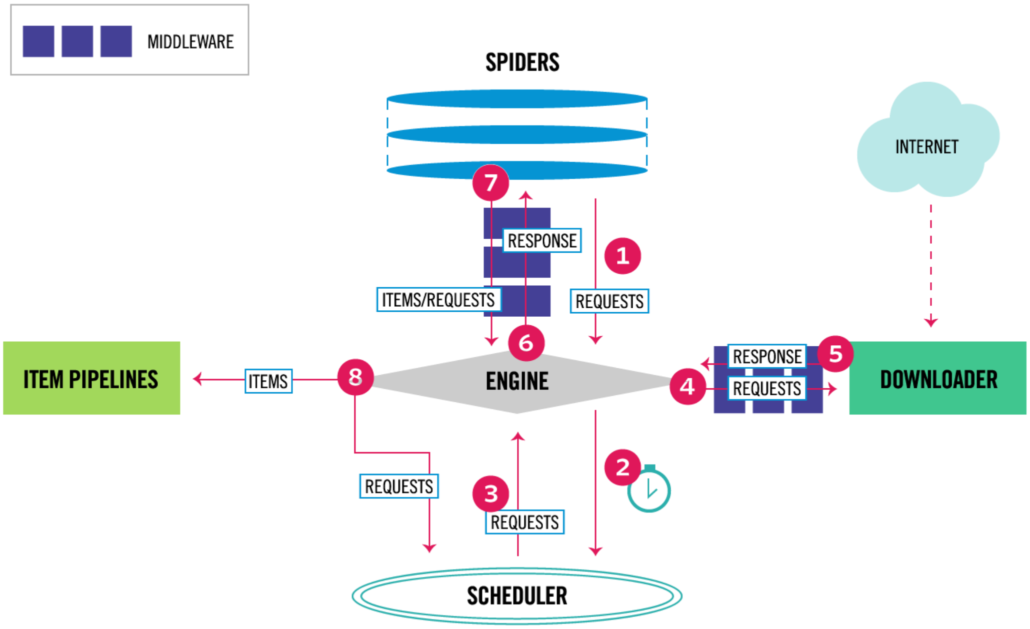


Figure 20

Products	neg
Canon 1.4x EF Extender III	0.03
Canon 55-250mm IS EF-S II	0.056
Canon EF 100-400mm f/4.5-5.6L IS USM	0.057
Canon EF 100mm f/2.8L IS Macro USM	0.038
Canon EF 135mm f/2L USM	0.043
Canon EF 16-35mm f/2.8L II USM	0.043
Canon EF 180mm f/3.5L Macro USM	0.101
Canon EF 24-105mm f/4L IS USM	0.072
Canon EF 24-70mm f/2.8 L USM	0.084
Canon EF 24-70mm f/2.8L USM II	0.045
Canon EF 24mm f/1.4L II USM	0.023
Canon EF 300mm f/2.8L IS II USM	0.055
Canon EF 35mm f/1.4L USM	0.063
Canon EF 400mm f/4 DO IS USM	0.08
Canon EF 50mm f/1.2L USM	0.072
Canon EF 50mm f/1.4 USM	0.065
Canon EF 50mm f/1.8 II	0.074
Canon EF 70-200mm f/2.8 L USM	0.097
Canon EF 85mm f/1.2L II USM	0.046
Canon EOS 1D Mark IV	0.068
Canon EOS 5D MK III	0.06
Canon EOS 5D MK IV	0.076

Canon EOS 5D Mark II	0.097
Canon EOS 5DS DSLR Camera	0.038
Canon EOS 5DSR DSLR Camera	0.058
Canon EOS 6D	0.062
Canon EOS 7D	0.075
Canon EOS 7D Mark II	0.056
Canon Extender EF 2x III	0.022
Nikon 105mm f/2.8G AF-S VR IF-ED Micro	0.069
Nikon 14-24mm f/2.8G AF-S ED	0.063
Nikon 16-35mm f/4G ED AF-S VR	0.096
Nikon 24-70mm f/2.8G AF-S ED	0.074
Nikon 28-300mm VR	0.074
Nikon 35mm f/1.8G AF-S DX	0.083
Nikon 500mm f/4G AF-S ED VR	0.059
Nikon 50mm f/1.8D AF	0.064
Nikon 70-200mm f/2.8G AF-S ED VR II	0.034
Nikon 85mm f/1.4G AF-S	0.048
Nikon D4	0.071
Nikon D500	0.068
Nikon D700	0.062
Nikon D7000	0.081
Nikon D7100	0.062
Nikon D750	0.049
Nikon D800	0.057
Nikon D800E	0.061
Nikon D810	0.055
Nikon D850 DSLR Camera	0.053
Tokina 11-16mm f/2.8 DX for Nikon	0.122

Table 14

Implementation

Overall Architecture

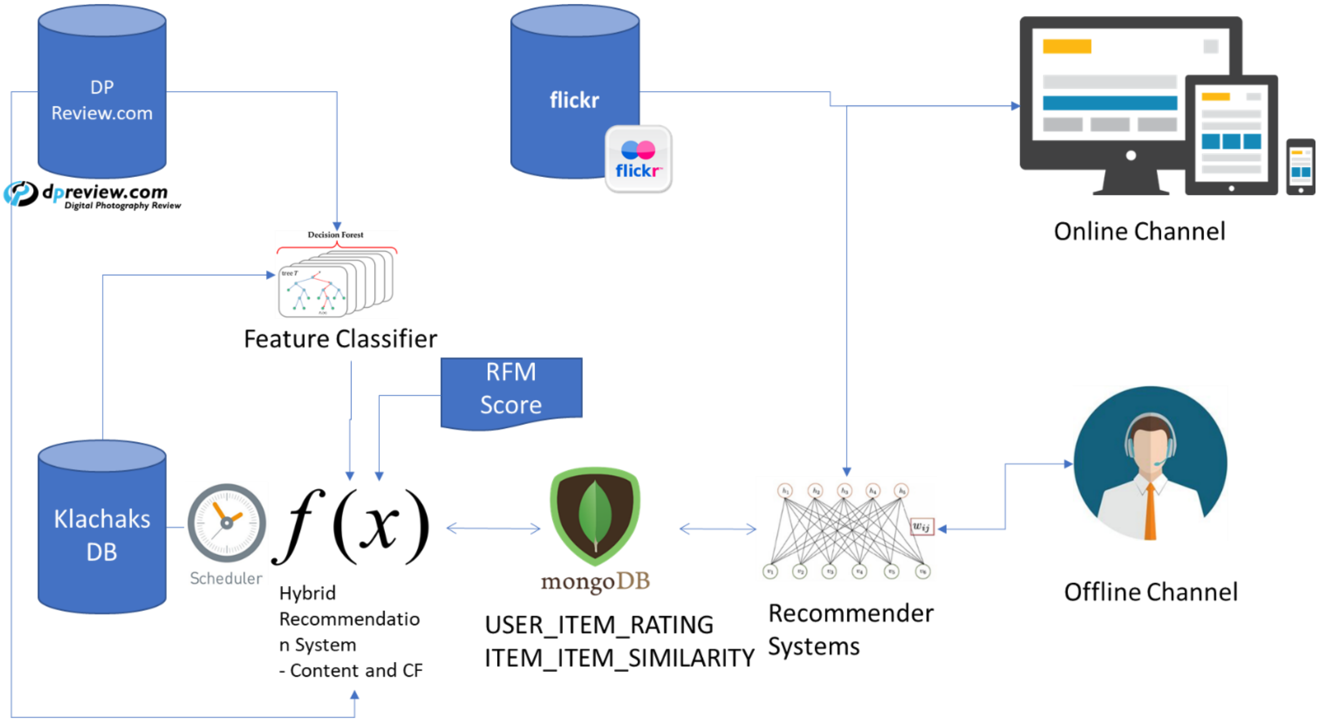


Figure 21

Post Implementation

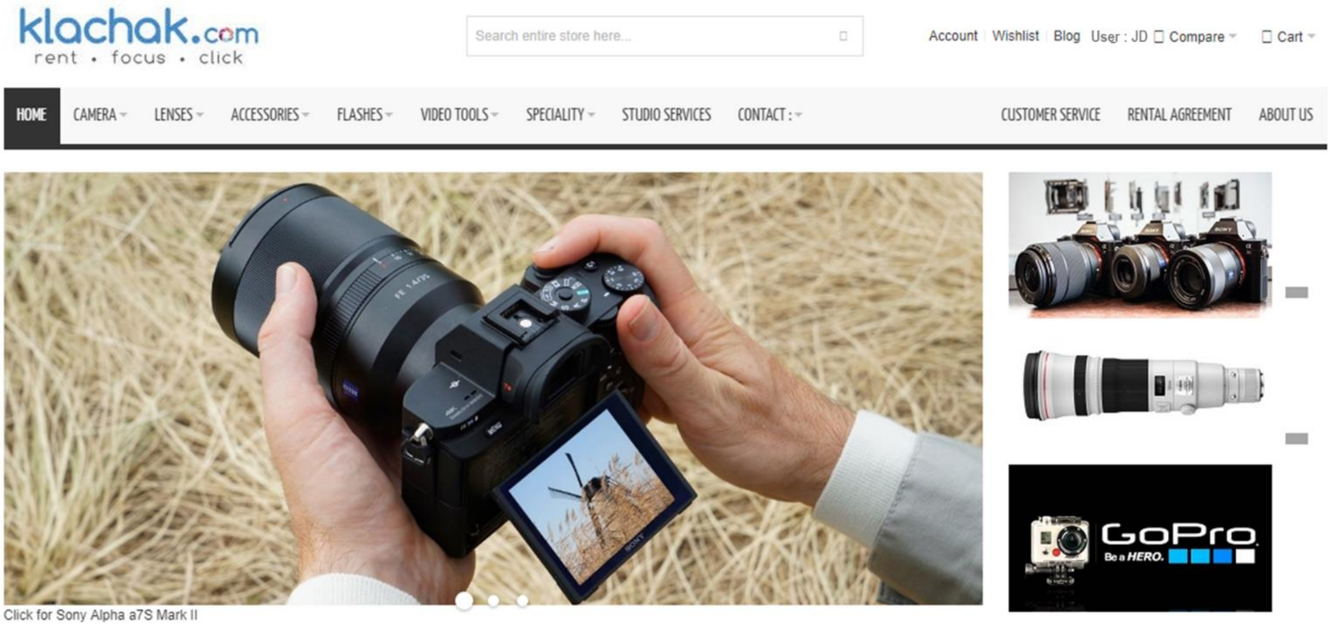
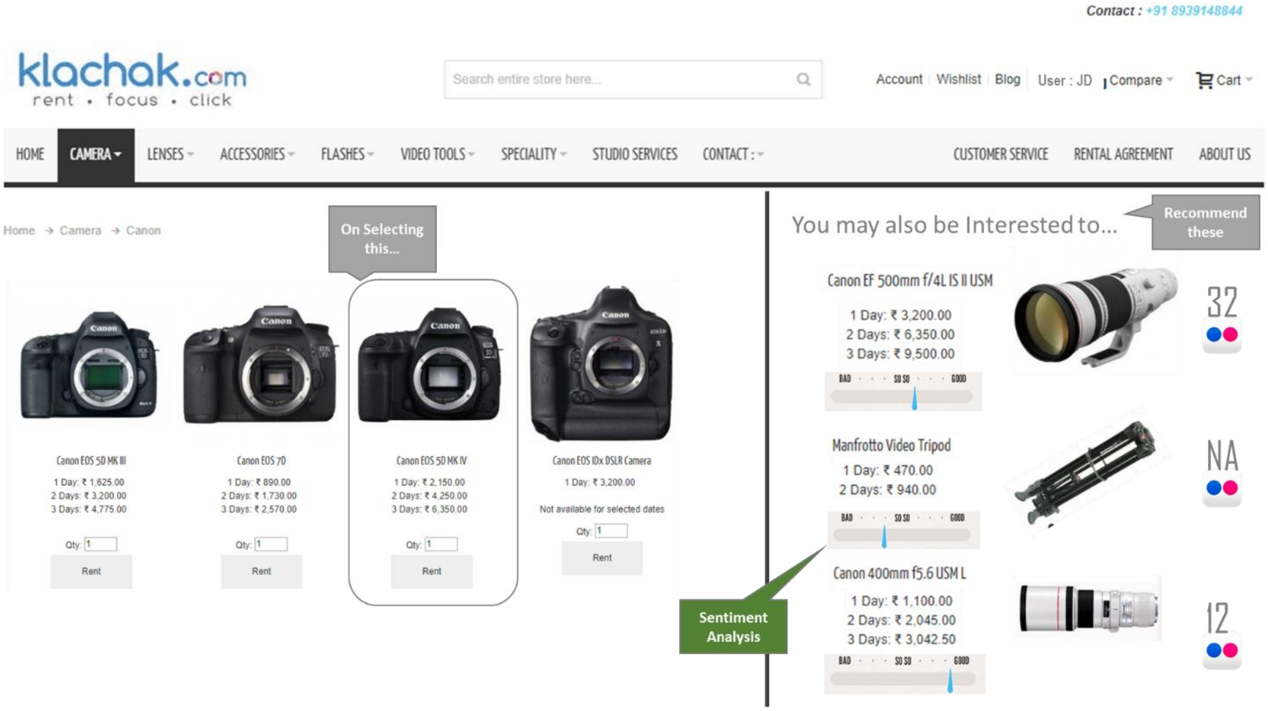


Figure 22

Recommendation Systems in Action



Canon EOS 5D MK III
1 Day: ₹ 1,625.00
2 Days: ₹ 3,200.00
3 Days: ₹ 4,775.00
Qty: 1
Rent

Canon EOS 7D
1 Day: ₹ 890.00
2 Days: ₹ 1,730.00
3 Days: ₹ 2,570.00
Qty: 1
Rent

Canon EOS 5D MK IV
1 Day: ₹ 2,150.00
2 Days: ₹ 4,250.00
3 Days: ₹ 6,350.00
Qty: 1
Rent

Canon EOS IDx DSLR Camera
1 Day: ₹ 3,200.00
Not available for selected dates
Qty: 1
Rent

You may also be Interested to...

Canon EF 500mm f/4L IS II USM
1 Day: ₹ 3,200.00
2 Days: ₹ 6,350.00
3 Days: ₹ 9,500.00

Manfrotto Video Tripod
1 Day: ₹ 470.00
2 Days: ₹ 940.00

Canon 400mm f5.6 USM L
1 Day: ₹ 1,100.00
2 Days: ₹ 2,045.00
3 Days: ₹ 3,042.50

Figure 23

Conclusion

The Hybrid Model, based on Content based and Collaborative filtering model has shown effective results in recommending the products for the sued based on feature similarity as well as item similarity. This will also help Klachaks to upsell and cross sell the items effectively. As RFM being one of the byproduct of this project, it will help Klachaks in perform targeted campaigns as well.

Future Scope

1. For Content Based Filtering, We need to Extract the features from multiple sources, like dpreview, flickr & Instagram
2. Use Analytical Hierarchy Processing for Enhancing the User Decision making process post recommendation.
3. Use Conjoint Analysis for deriving Part Worth Utility, intron Use it as input for Deriving user-item rating in more precise way.
4. Real time in-memory computing using spark so that it will be scalable going forward as and when the size of the item matrix and user matrix grows.

References

[1] Item-based Collaborative Filtering Recommendation Algorithms.

- Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl

[2] A Comparative Study of Collaborative Filtering Algorithms.

-Joonseok Lee, Mingxuan Sun, Guy Lebanon

[3] Mining Massive Datasets – Recommender Systems

- Leskovec, Rajaraman and Ullman

[4] A Hybrid Film Recommender System Based on Random Forest

- ANDREAS BROMMUND & DAVID SKEPPSTEDT