Expedia – Recommendation System

R Nomads

Avanthi Namburi

Jayasree Akula

Meghana Kasula

Seemantini Chincholkar

Seshi Harianathan

Shaloma Ghosh

CONTENTS

Executive Summary 2
Introduction 2
Problem Statement 2
Dataset Specifications 3
Data Pre-processing 3
Data Exploration and Visualization 4
Exploring site_name 4
Hotel Cluster – Mobile – Package Relationship 4
Marketing Channel - investigating the behavior of marketing channel. 6
Conventional Recommender System 6
Network Analysis 8
Item Based Filtering 11
Conclusion – Business Value and Application & Future Scope 11
Appendix: 12
References 12

Executive Summary

In this information age, retrieving information is as simple as breathing. But since every website online has some service or product to offer, it has become increasingly important that the services offered stand out amongst the plethora of competition already existing. The numerous analytical tools in the market today are a medium for not only tapping market trends but also provide an edge to online services. We, through the dataset provided from Expedia will explore concepts like Recommender System, Networks, Graph Mining, Associations rules and produce a model to recommend most relevant hotel clusters to a respective user.

Introduction

Recommendation engines have had an enormous influence on the typical consumer. Today's consumers buy products recommended by Amazon, watch movies recommended by Netflix, and read Facebook posts organized according to its internal recommendation engine, and they visit restaurants recommended by yelp. Even the ads they see are recommended based on their previous online behavior.

If applied well, recommendation engines can boost profit for companies immensely and thus serve as selling point for them.

Recommender systems is defined as a system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting objects in a larger space of possible options.

There are mainly two types of recommendation systems.

- Collaborative Filtering
- Content-Based Filtering

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users.

Content-based filtering methods are based on a description of the item and a profile of the user's preference. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). Various candidate items are compared with items previously rated by the user and the best-matching items are recommended.

Problem Statement

Expedia has provided us with logs of customer behavior. These include what customers searched for, how they interacted with search results (click/book), if the search result was a travel package.

Following are our Objectives:

1. Our objective is to recommend hotel clusters based on user behavior and content using the two types of recommendation engines, collaborative and content based.
2. Understand the Network between various variables like hotels, users ,etc and indicate the top hotels in each node.

Dataset Specifications

The dataset has been taken from the following webise: https://www.kaggle.com/c/expedia-hotel-recommendations/data

Following table explains all the variables:

Variable Name	Data Dictionary	Variable Name	Data Dictionary
date_time	Timestamp	srch_co	Checkout date
site_name	ID of the Expedia point of sale	srch_adults_cnt	The number of adults specified in the hotel room
posa_continent	ID of continent associated with site_name	srch_children_cnt	The number of (extra occupancy) children specified in the hotel room
user_location_country	The ID of the country the customer is located	srch_rm_cnt	The number of hotel rooms specified in the search
user_location_region	The ID of the region the customer is located	srch_destination_id	ID of the destination where the hotel search was performed
user_location_city	The ID of the city the customer is located	srch_destination_type_id	Type of destination
orig_destination_distance	Physical distance between a hotel and a customer at the time of search.	is_booking	1 if a booking, 0 if a click
user_id	ID of user	cnt	Numer of similar events in the context of the same user session
is_mobile	1 when a user connected from a mobile device, 0 otherwise	hotel_continent	Hotel continent
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	hotel_country	Hotel country
channel	ID of a marketing channel	hotel_market	Hotel market
srch_ci	Checkin date	hotel_cluster	ID of a hotel cluster

Data Pre-processing

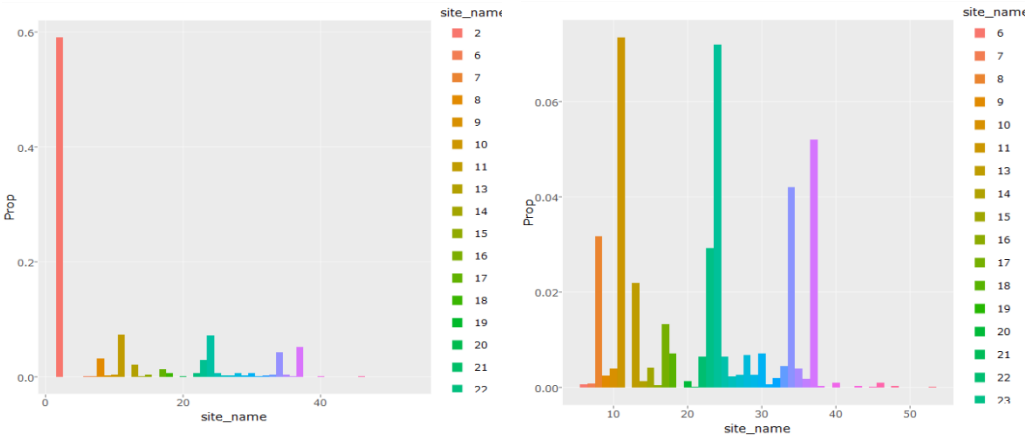
- We have selected random 10000 with booking = 1 rows from train dataset covering unique 9980 users.
- Expedia has a limitation which says the number of Adults or rooms cannot be 0.
- Hence, we imputed the values and replaced 0 with 1 as default.
- There were instances where original distance was NA. We used the highest value to replace it.
- Most of the variables were of type Factor but they were represented numerically.
- Hence we transformed all such variables into factors using as.factor function.

Data Exploration and Visualization

We are given logs of visitors at different Expedia sites and are asked to predict the hotel clusters in the test set. Expedia aims to use customer data to improve their hotel recommendations. We analyzed this dataset and tried to get some insight of it.

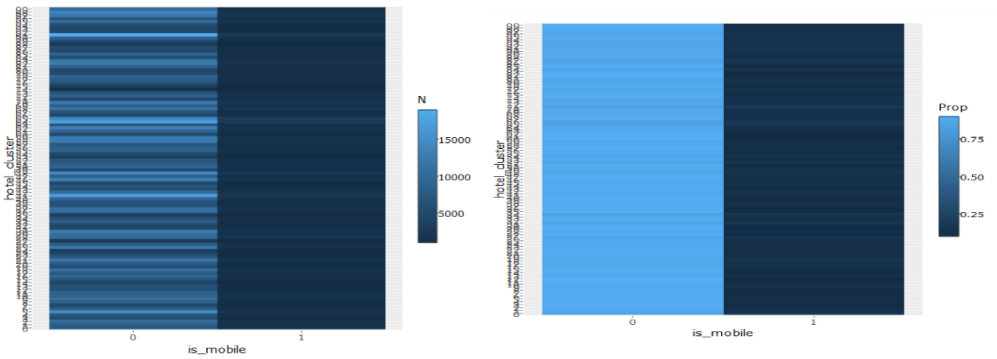
We converted the types of some columns. is_mobile, is_package, channel, posa_continent, hotel_continent as categorical variables by using the *as.factor* function. There are some other categorical varaibles such as hotel_cluster, hotel_country, user_location, etc. but these will be left as integer values, since they have too manyentries, which makes it harder to deal with them when plotting.

Exploring site_name



The Frequency of usage of each site_name shows us that site_name 2 is used most often. We constructed another graph to observe the frequency of other sites to see other proportions better.

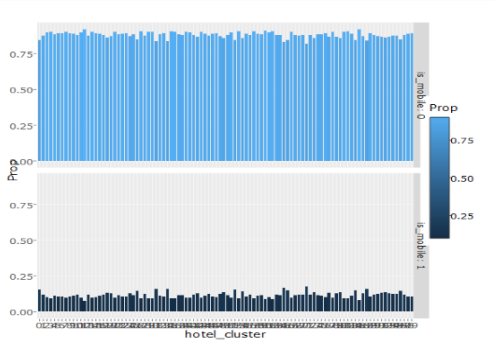
Hotel Cluster – Mobile – Package Relationship



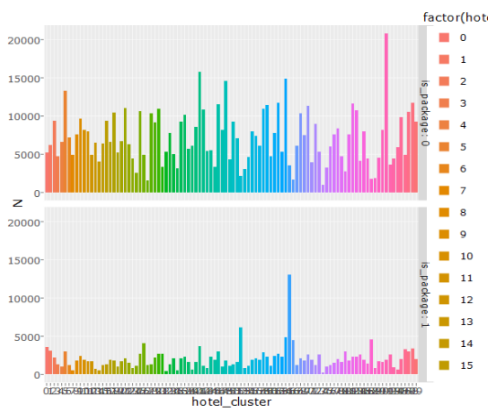
Insight into hotel cluster – mobile relationship.

We observed that most customers are not using mobile. Tile graph is used where each point is a rectangle and colored based on the intensity. We can see that proportion of people no using mobiles are more.

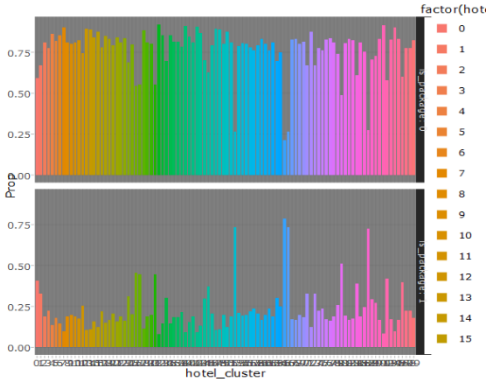
This begs the question, are there any hotels where users used mobile dominantly? So, at each hotel cluster, we decided to check the proportions. The proportions are calculated over each hotel_cluster. Thus, the probabilities of both mobile user and non-user sums up to 1 for each hotel_cluster.



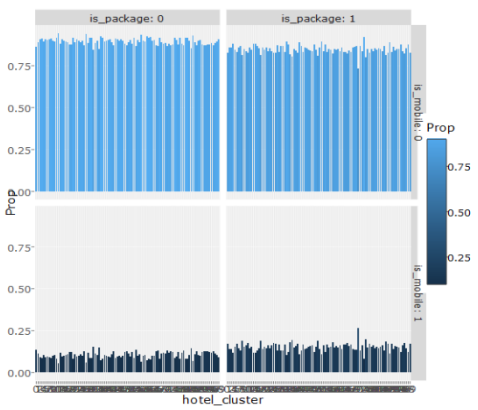
Bar graph could be a better, visualization involve making a comparison.So no differnce between being we can clearly see the desktop users are more for every hotel_clusters.



Let's check the relation between is_package and hotel_cluster. Here we have used bar graph. In general, the hotel clusters or more acted upon in is_package = 0, But just for hotel cluster 65, it is almost equal in both.

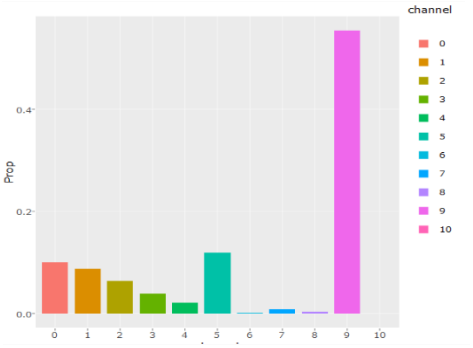


The above graph tells us that some hotel clusters have more visits. But to see if any hotel cluster is more related to being searched within a package, we will need to plot the proportions within each hotel_cluster. Some clusters such as 52, 65, 66, and 87 are more related to being in a

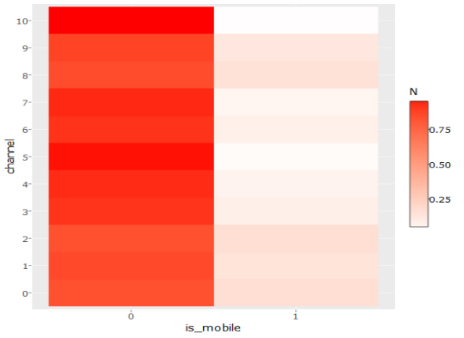


Trying to find a relationship between hotel_cluster, is_mobile, and is_package. First, when is_mobile is 1, does is_package change or stay the same? As, can be seen, it stays the same, slightly higher proportions of is_package 1 and is_mobile1. Second, when is_package has a large value for a hotel_cluster, does it behave similarly in is_mobile? As can be seen from second graph, is_mobile behaves the same in both of its categories, suggesting indifference to is_package.

Marketing Channel - investigating the behavior of marketing channel.



Channel 9 dominates all with a huge difference. Let's see if there is any relationship between channels and is_mobile variables.



Channel 0, 1, 2, 8 and 9 have above average proportion in general with mobile users, suggesting that these marketing channels are optimum for users using mobiles for Expedia.

Conventional Recommender System

Generally, recommendation systems work on user ratings. In case of Expedia data, we only had the user logs or the transaction data. There was no column which had data for rating. To overcome this problem and build a traditional recommendation engine, we decided to create ratings using the concept of **implicit feedback**.

We decided to subset the data based on the srch_destination_id as that is one of the inputs to the Expedia welcome page.

	4	6	7	15
35370	0	0	0	0
155713	0	0	0	0
164696	0	0	2	0
167244	0	0	0	1
212552	0	0	0	0

After getting the data for the mentioned destination, we calculated a user-item matrix taking rows as unique users and column as the unique clusters. The element in this matrix was number of bookings. Here user 164696 made 2 bookings for Cluster type 7.To bring the values to one similar scale, we used the Sigmoid Function

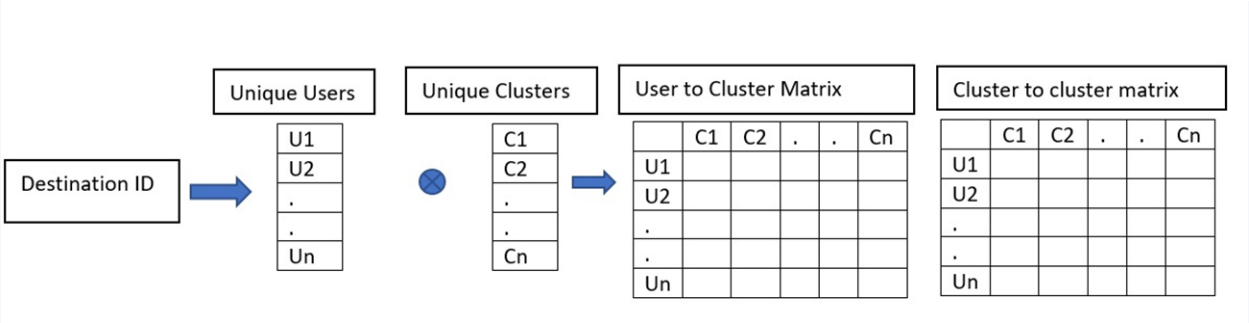
```
final_data = final_data1 / (1 + final_data1)
```

This brings all the matrix element values to a scale of 0 – 1. Now we have our **implicit ratings**.

Content Based Filtering: We perform this by calculating the cluster similarity between all the hotel clusters which fall under the search criteria.

Input: srch_destination_id and user_id

Output: a matrix with top 10 neighbors for each cluster for that destination



For calculating the similarity, we use the cosine similarity function

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

We use this formula to calculate distance between 2 clusters and the sorted them in order on their closeness. The values passed to this function are from the User to cluster matrix shown above

```
getCosine <- function(x,y)
{
  this.cosine <- sum(x*y) / (sqrt(sum(x*x)) * sqrt(sum(y*y)))
  return(this.cosine)
}
```

	4	6	7	15	16
1	1	0	0	0	0.0000000
2	0	1	0	0	0.0000000
3	0	0	1	0	0.0000000
4	0	0	0	1	0.0000000
5	0	0	0	0	1.0000000
6	0	0	0	0	0.0000000
7	0	0	0	0	0.0000000
8	0	0	0	0	0.5773503

	Hotel cluster	1	2	3
1	0	1	2	3
2	1	2	20	18
3	10	3	20	21
4	14	4	1	2
5	18	5	13	19

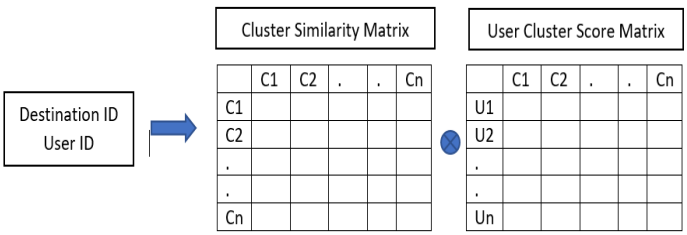
```
expedia_cluster_similarity[i,j] <- getCosine(as.matrix(final_data[i]),as.matrix(final_data[j]))
```

After calculating the similarities, we sort them in decreasing order to get the neighboring clusters for which the specified destination is mentioned. For cluster 10, the neared neighboring clusters are 3,20 & 21.

Collaborative Filtering: Here we calculate user score based on the cluster similarity and user history. With the score we retrieve the top clusters that users similar to the input user_id have selected.

Input: srch_destination_id and user_id

Output: Top 5 clusters for given user id based on the selection of other similar users



We create a function getScore to find out the score between the users

```
getScore <- function(history, similarities)
{
  x <- sum(history*similarities)/sum(similarities)
  x
}
```

We calculate the user score using the getScore formula and then sort the top 10 clusters which our algorithm suggests for that respective user.

```
# We then calculate the score for that product and that user
holder[i,j]<-getScore(similarities=topN.similarities,history=topN.userPurchases)

#View(holder)
#ncol(expedia_scores_holder)

# Lets make our recommendations pretty
expedia_scores_holder <- matrix(NA, nrow=nrow(expedia_user_scores),ncol=ncol(expedia_user_scores),dimnames=list(rownames(expedia_user_scores)))
for(i in 1:nrow(expedia_user_scores))
{
  expedia_scores_holder[i,] <- names(head(n=100,(expedia_user_scores[,order(expedia_user_scores[i,],decreasing=TRUE)])[i,]))
}
```

	user id	1	2	3
1	1880	32	35	34
2	1916	99	79	19
3	2956	35	76	94
4	4397	45	24	88
5	6159	94	99	41
6	9265	10	71	79

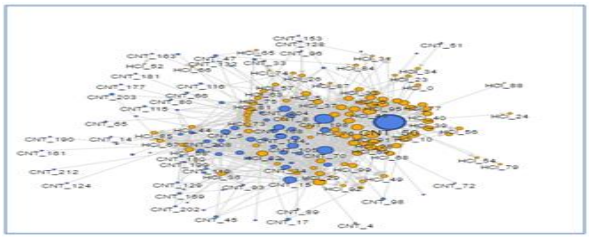
For user 35070, the top clusters recommended based on similar user's purchases are clusters 50,16,42,91,4

Network Analysis

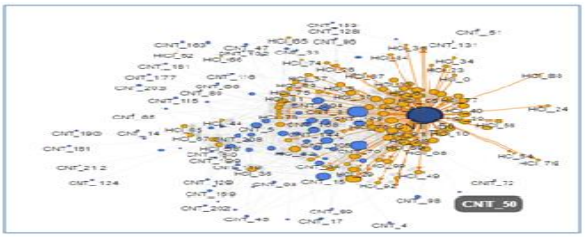
Social network analysis (SNA) is the process of investigating social structures through the use of networks and graph theory. It characterizes networked structures in terms of nodes (individual actors, people, or things within the network) and the ties, edges, or links (relationships or interactions) that connect them. Examples of social structures commonly visualized through social network analysis include social media networks, collaboration graphs, kinship, disease transmission, and sexual relationships. These networks are often visualized through sociograms in which nodes are represented as points and ties are represented as lines.

As team RNoMads, we tried to use the concept of SNA for finding associations between different attributes of the Expedia booking historical data and build Recommendation systems based on that.

Before jumping into the approach, here are the social network diagrams of the columns, hotel_country,hotel_cluster weighted by the user_id column. we can visually identify what are the major nodes.



Picture (A) - Complete network of Country-Hotel Cluster based on user.



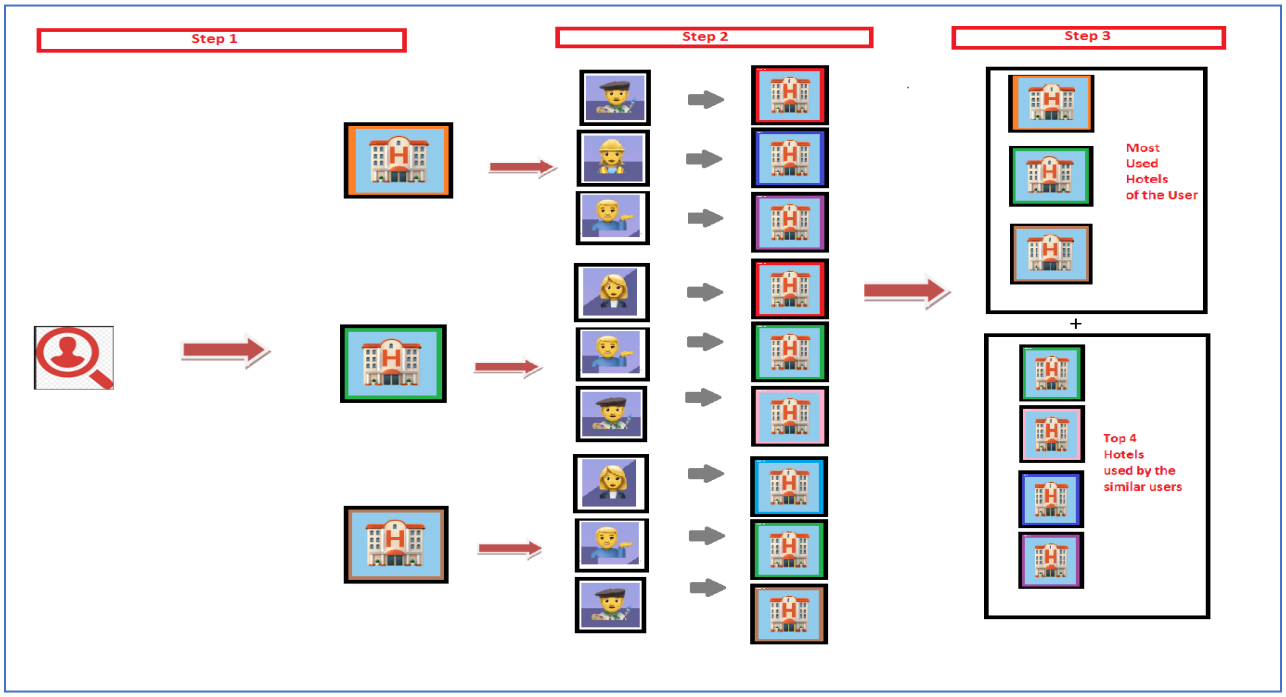
Picture (B) - Connections of CNT-50 - different Hotel Clusters.

Hence, we decided to utilize this feature of SNA for recommendation systems and following are the two proposed recommendation systems:

- Collaborative Filtering based on user history
- Item based Filtering based on popularity

Collaborative Filtering based on user history :

In this approach, which applies to a search of user with historical data, a user performs search based on the desired criteria and clicks on the search button. For demonstration we used the search criteria as "Hotel_Country". In step 1, Top N clusters for Hotel_Country are retrieved using igraph note associations of hotel_country to hotel_cluster.
In step 2, The input will be the result from step 1 (hotel clusters of hotel_country search) and then associations are obtained for users for these hotel_clusters.
In step 3,The user list is obtained from step 2, and then the most popular clusters used by these users is obtained.
Hence the final recommended list will be the clusters from step 1 + step 3.Pictorial representation is as follows.



We used igraph package to create the network associations between countries and clusters and below is the code for that

```
#creating a social network graph between columns user_id_country,hotel_Cluster
# to see the association
Hgraph = graph_from_data_frame(df, directed = TRUE, vertices = NULL)
vgraph = as_data_frame(Hgraph, what="vertices")
egraph = as_data_frame(Hgraph, what="edges")
```

The code of the function that performs the Collaboration filtering using network nodes is as follows:

```
SN_Collaborative_Filtering = function(user,country,n)
{
  #Top N Clusters from the users previous history(step1)
  R_Step1 = User_Top_N_Clusters(user,country,n)
  #converting the results to a vector, this list will be passed to next step.
  User_CLuster_List = R_Step1

  #step 2 Top Users of the clusters obtained from Top Clusters received from step 1
  R_Step2 = Clusters_Versus_Top_Users(User_CLuster_List,25)
  R_Step2_data = as.data.frame(Clusters_Versus_Top_Users(User_CLuster_List,25))

  #obtaining the list fo users from the output.This list will be passed to step 3.
  Top_Cluster_Top_Users = R_Step2_data

  #step 3 Top Clusters of the top users obtained from step 2
  R_Step3 = Top_User_Cluster_User_Cluster(Top_Cluster_Top_Users,length(Top_Cluster_Top_Users))
  Top_User_Cluster_User_Cluster_List = R_Step3

  if(nrow(Top_User_Cluster_User_Cluster_List)==0)
  {
      if(nrow(User_CLuster_List)==0)
      {
        print("No relevent search results")
      }
      else
      {
        head(unique(User_CLuster_List$to),n)
      }

  }
  else
  {
    Top_User_Cluster_User_Cluster_List = rbind(User_CLuster_List,Top_User_Cluster_User_Cluster_List)
    head(unique(Top_User_Cluster_User_Cluster_List$to),n)
  }}
```

Code for Eigen Vector Centrality

```
#To display the Eigen Vectors of the Network(Users grouped through country across clusters)
EV <- evcent(Hgraph)
sort(unlist(EV), decreasing=TRUE)
x=data.frame(head(EV$vector,20))
colnames(x)=c("Eigen Vector Centrality")
z = data.frame( head(rownames(as.data.frame(EV)),20), x$`Eigen Vector Centrality`)
colnames(z) = c("USER_COUNTRY","Eigen Vector Centrality")
head(z,10)
```

Which means the important people from the network are the people with higher Eigen Vector Centrality.

Output from our Expedia Dataset

	USER_COUNTRY	Eigen Vector Centrality
1	U_35370_50	0.0104202409
2	U_35370_151	0.0001363984
3	U_35370_46	0.0052147981
4	U_35370_105	0.0008233725
5	U_35370_162	0.0001135251
6	U_35370_126	0.0011273114
7	U_35370_106	0.0017848985
8	U_39375_204	0.0007983921
9	U_39375_208	0.0001864099
10	U_45567_50	0.0014638352

```
##To display the Top most Clusters based on the Degree (User_Country_Clusters)
Degree <-degree(Hgraph,mode="total")
sort(unlist(Degree), decreasing=TRUE)
aa=head(as.data.frame(Degree),20)
z = data.frame(head(rownames(as.data.frame(aa)),20), aa$Degree)
colnames(z) = c("Popular People","Degree")
head(z,10)
```

Beside are the results for the popularity in terms of the Degree(no of connections)

	Popular People	Degree
1	U_35370_50	4
2	U_35370_151	1
3	U_35370_46	1
4	U_35370_105	1
5	U_35370_162	2
6	U_35370_126	1
7	U_35370_106	3
8	U_39375_204	1
9	U_39375_208	1
10	U_45567_50	2

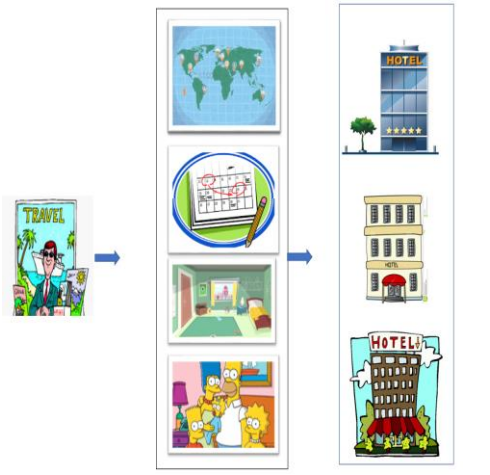
```
> SN_Collaborative_Filtering(user=181651,country=8,n=10)
[1] HCl_29 HCl_36 HCl_61 HCl_82 HCl_99 HCl_46 HCl_4
26 Levels: HCl_29 HCl_36 HCl_61 HCl_82 HCl_99 HCl_11 HCl_2 HCl_21 HCl_28 HCl_30 HCl_4 HCl_41 HCl_42 HCl_43 HCl_46 HCl_48 HCl_5 HCl_50 ... HCl_90
> SN_Collaborative_Filtering(user=513782,country=50,n=10)
 [1] HCl_7  HCl_21 HCl_42 HCl_28 HCl_91 HCl_59 HCl_2  HCl_48 HCl_13 HCl_15
38 Levels: HCl_13 HCl_15 HCl_16 HCl_2 HCl_21 HCl_25 HCl_28 HCl_33 HCl_42 HCl_48 HCl_51 HCl_59 HCl_7 HCl_73 HCl_77 HCl_9 HCl_91 HCl_94 ... HCl_98
> SN_Collaborative_Filtering(user=513782,country=125,n=10)
[1] "No relevent search results"
> SN_Collaborative_Filtering(user=51782,country=50,n=10)
[1] "No relevent search results"
```

Item Based Filtering

The above recommendations are based on User History, this recommendation is based on the Booking history and is not based on any specific user. This recommends the most booked hotel clusters based on different search inputs.

In this process, we considered only the booking history of all the users. Using the package “igraph” we have come up with edges list, which has all the associations between the nodes. Considering nodes to be the input criteria & Hotel clusters. Associations are all the connection between nodes.

For a search criteria, this function searches all the associations pertaining to that search and displays the required number of hotel clusters. If the Hotel clusters are less than the number of required outputs it displays the most booked hotel clusters related to that country along with the search results.




```
> SNA_Item_based_filter_generic(country = 99,checkin = NA,checkout =NA,room =1,adults = 4,children = NA,n=5)
  top_clusters.to
1          HCl_46
2          HCl_82
3          HCl_81
4          HCl_36
5          HCl_67

> SNA_Item_based_filter_generic(country = 50,checkin = "2017-04-22",checkout = "2017-04-27",room =1,adults = 4,children = NA,n=5)
  top_clusters.to
1          HCl_17
2          HCl_26
3          HCl_41
4          HCl_48
5          HCl_94

> SNA_Item_based_filter_generic(country = 121,checkin = "2017-05-23",checkout ="2017-05-27",room =2,adults = 4,children = NA,n=5)
  top_clusters.to
1          HCl_73
2          HCl_92
3          HCl_97
4          HCl_26
5          HCl_22

> SNA_Item_based_filter_generic(country = 50,checkin = NA,checkout =NA,room =NA,adults = NA,children = NA,n=5)
  top_clusters.to
1          HCl_91
2          HCl_48
3          HCl_42
4          HCl_28
5          HCl_50
```

Conclusion – Business Value and Application & Future Scope

Benefits of using Network Analysis for Recommendation systems -

Minimum programming is required, the time required for calculation is very less. Very good for initial insights.

Benefits of using Traditional Recommender systems -

we can use these outputs in support of any machine learning algorithms being built on data.

Conclusion –

when there is no budget or scope for machine learning Recommendations using Network Analysis will be the best.

Applications of Traditional Recommender systems

Not only we can recommend users page, we can make a optimized recommendation systems based on the matrix obtained.

Market Basket Analysis –

Formulated rules, future scope to provide recommendations

Appendix:

Packages Explored/Used:

- data.table
- igraph
- RecommenderLab
- plotly
- arules
- arulesviz
- sqldf
- ggplots
- reshape
- rgdal
- rgraphbiz
- rsqlite
- graphics
- gridmatrix
- stringR

References

http://sna.stanford.edu/lab.php?l=1

http://igraph.org/r/doc/graph_from_data_frame.html

https://www.r-bloggers.com/recommender-systems-101-a-step-by-step-practical-example-in-r/

https://en.wikipedia.org/wiki/Recommender_system

http://www.salemmarafi.com/code/collaborative-filtering-r/

https://www.r-bloggers.com/expedia-data-analysis-part-1/