**Project Requirement:-**

|  |
| --- |
| **Context** |
| The csv file contains 17 fields. The description of each field is as below:  Hotel\_Address: Address of hotel.  Review\_Date: Date when reviewer posted the corresponding review.  Average\_Score: Average Score of the hotel, calculated based on the latest comment in the last year.  Hotel\_Name: Name of Hotel  Reviewer\_Nationality: Nationality of Reviewer  Negative\_Review: Negative Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Negative'  Review\_Total\_Negative\_Word\_Counts: Total number of words in the negative review.  Positive\_Review: Positive Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Positive'  Review\_Total\_Positive\_Word\_Counts: Total number of words in the positive review.  Reviewer\_Score: Score the reviewer has given to the hotel, based on his/her experience  Total\_Number\_of\_Reviews\_Reviewer\_Has\_Given: Number of Reviews the reviewers has given in the past.  Total\_Number\_of\_Reviews: Total number of valid reviews the hotel has.  Tags: Tags reviewer gave the hotel.  days\_since\_review: Duration between the review date and scrape date.  Additional\_Number\_of\_Scoring: There are also some guests who just made a scoring on the service rather than a review. This number indicates how many valid scores without review in there.  lat: Latitude of the hotel  lng: longtitude of the hotel |
|  |
|  |
|  |

**Author:** Arijit Pal (aryanarijit@gmail.com),

Abhishek Biswas ([abby33459@gmail.com](mailto:abby33459@gmail.com))

**Review and comment based recommendation system biased to hotel**

**Introduction :-**

This project is focused on to build two things:-

1. Basic recommendation system to each hotel and also promotional advice.
2. Make cluster of hotels based on rating and user review.

**Benefit:-**

As a hotel manager , if any person follow the process of recommendation of any hotel, she/she can understand the ethical value of the hotel to customer or visitors and based on that he/she can take decision to further improvement or can understand the promotional strategy.

**Data Preparation and Idea:-**

Below is the list of columns that we consider for K means clustering for three groups :-

*highly satisfied*

*satisfied*

*not satisfied*

The columns are :-

nationality, negative words count, positive word count, total no of reviews, review score

Now, we shall consider any one hotel for the analysis, we can continue the same for any hotel.

Our aim is to start an unsupervised algorithm to group the data for each hotel into three categories and add the category in a new column named “Feedback”. And add another column in the data set named “recommend “ and put value 0 for the category “*not satisfied*“ and put vale 1 for the categories *highly satisfied,satisfied.*

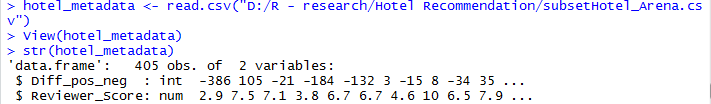
Now if any customer from any country wants to book the site, then the hotel will get recommended if the existing data has more counts of 1 than count of 0s.

And the hotel manager will concentrate more to promote the hotel to that country where the count of 0s are more than 1.

We are started to work with unsupervised algorithm k-means clustering and then add labels of category to use supervised algorithm like Decision Tree and Ensemble methods, KNN, SVM to predict the category further for new set of future data based on new entry in the booking list and reviews.

**K Means:-**

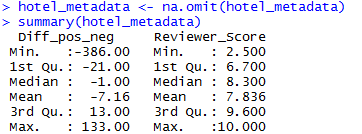
Import the data in R - (K means Algo)



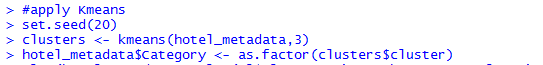
Checking rows:

nrow(hotel\_metadata)

Omitting nulls



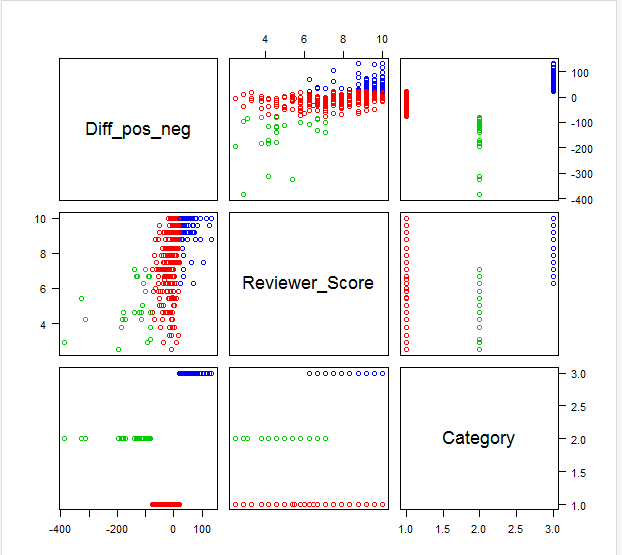
Apply K means



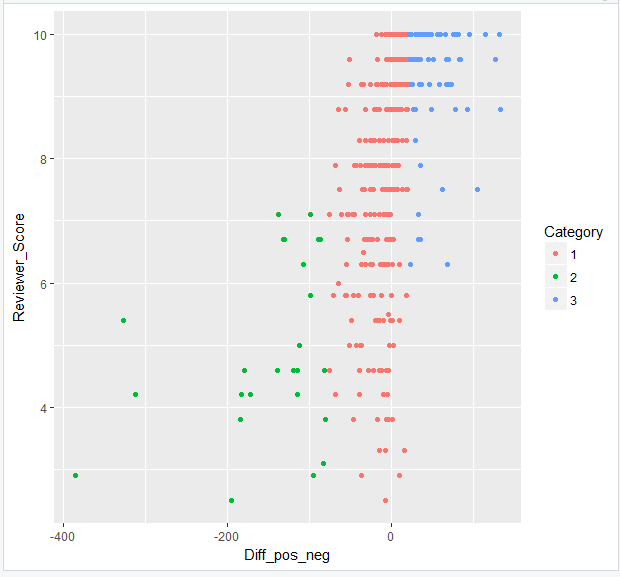
We have considered the value of K =3

Pair plot :







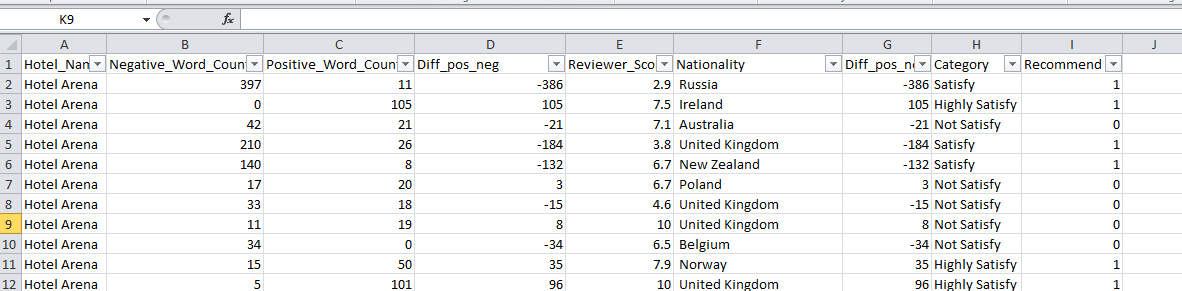


We have clearly understand that high review score values are grouped under category 3 largely which is indicating highly satisfy because positive word count – negative word count value is high means having high positive word count.

Export data :

write.csv(hotel\_metadata,"D:/R - research/Hotel Recommendation/hotel\_arena\_label.csv", quote = FALSE, row.names = TRUE)

Now we need to add the new column in main data set of hotel Arena.( file : Hotel\_Arena.csv)



This data set will be used further to predict the category for new set of bookings, review based on user.

Now, we shall apply supervised clustering algorithms to predict the value of recommend column for new set of entries.

**Review and comment based recommendation system non biased to hotel and region**

(K means Algo - for all hotel)

Now we shall apply the same K means concept for all records irrespective of hotel.

Pair plot after applying K means

Code :

hotel\_metadata <- read.csv("D:/R - research/Hotel Recommendation/data\_prep\_rvw\_all.csv")

View(hotel\_metadata)

str(hotel\_metadata)

nrow(hotel\_metadata)

#preprocess data to avoid null values

hotel\_metadata <- na.omit(hotel\_metadata)

summary(hotel\_metadata)

#apply Kmeans

set.seed(20)

clusters <- kmeans(hotel\_metadata,3)

# Save the cluster number in the dataset as column 'Category'

hotel\_metadata$Category <- as.factor(clusters$cluster)

# Inspect 'clusters'

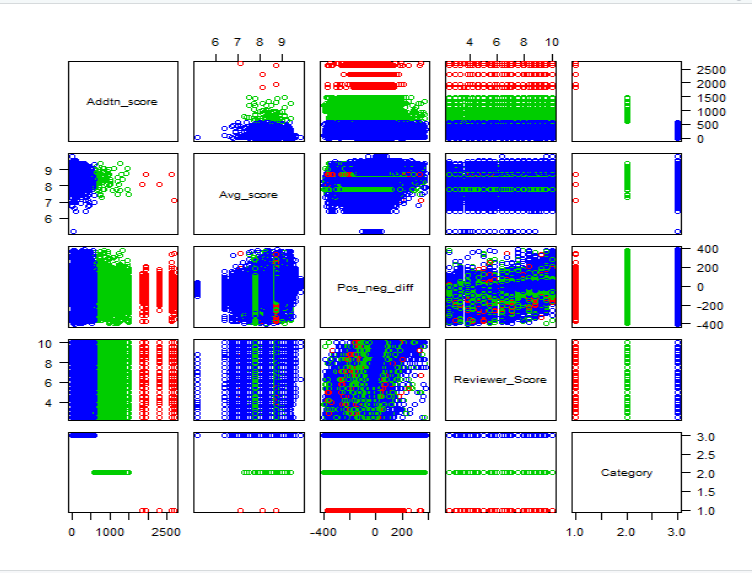
str(clusters)

library(ggplot2)

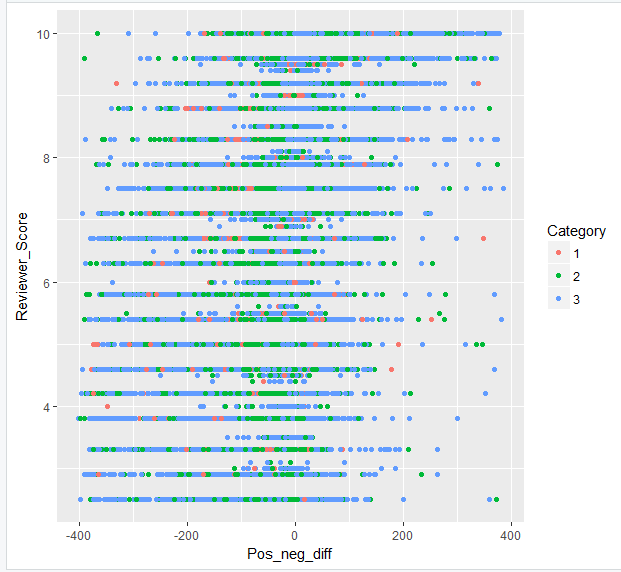
#You can plot the graph and cluster centroid using the following command.

plot(hotel\_metadata, col =(clusters$cluster +1) , pch=1, cex=1, las=1)

ggplot(hotel\_metadata, aes(Pos\_neg\_diff,Reviewer\_Score, color = Category)) + geom\_point()



Plot of group :



We cannot come to any decision about the satisfaction level of user.

Lets run the algorithm again after removing the column additional score and avg score.

# after removing avg score and additional score column

hotel\_metadata <- read.csv("D:/R - research/Hotel Recommendation/data\_prep\_rvw\_all\_v2.csv")

View(hotel\_metadata)

str(hotel\_metadata)

nrow(hotel\_metadata)

#preprocess data to avoid null values

hotel\_metadata <- na.omit(hotel\_metadata)

summary(hotel\_metadata)

#apply Kmeans

set.seed(20)

clusters <- kmeans(hotel\_metadata,3)

# Save the cluster number in the dataset as column 'Category'

hotel\_metadata$Category <- as.factor(clusters$cluster)

# Inspect 'clusters'

str(clusters)

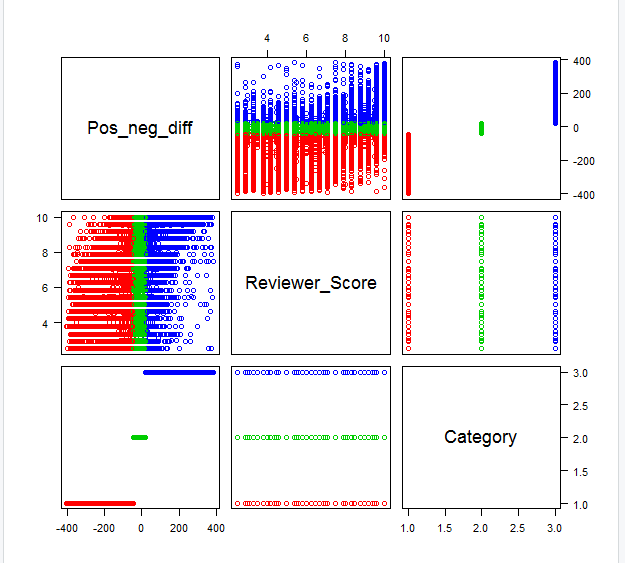
library(ggplot2)

#You can plot the graph and cluster centroid using the following command.

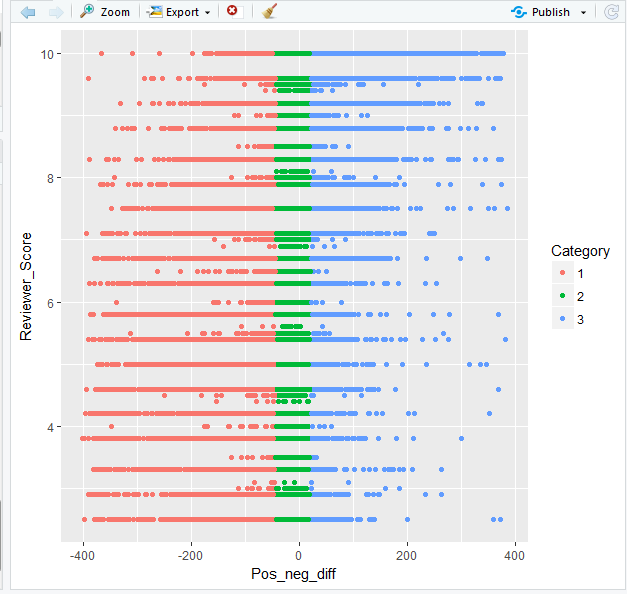
plot(hotel\_metadata, col =(clusters$cluster +1) , pch=1, cex=1, las=1)

ggplot(hotel\_metadata, aes(Pos\_neg\_diff,Reviewer\_Score, color = Category)) + geom\_point()

Pair plot :



Lets check the group/clustering



This group is better to interpret about the satisfaction level of user.

If user gives high or good review then there should be more concentration of positive words and if user gives less review then there should be more concentration of negative words.

And the average feedback comment should be scattered in all review points.

write.csv(hotel\_metadata,"D:/R - research/Hotel Recommendation/allhotel\_all\_label.csv", quote = FALSE, row.names = TRUE)

We have prepared a new excel by appending the data into the main file from the exported file and named the new file as “Supervised\_data\_prep\_all\_hotel.csv”

**Only Review based recommendation system non biased to hotel and region**

(K means Algo - for all hotel - only review)

Now we shall consider the average review and additional score to group the hotels into 3 clusters.

hotel\_metadata <- read.csv("D:/R - research/Hotel Recommendation/review\_hotel.csv")

View(hotel\_metadata)

str(hotel\_metadata)

nrow(hotel\_metadata)

#preprocess data to avoid null values

hotel\_metadata <- na.omit(hotel\_metadata)

summary(hotel\_metadata)

#apply Kmeans

set.seed(20)

clusters <- kmeans(hotel\_metadata,3)

# Save the cluster number in the dataset as column 'Category'

hotel\_metadata$Category <- as.factor(clusters$cluster)

# Inspect 'clusters'

str(clusters)

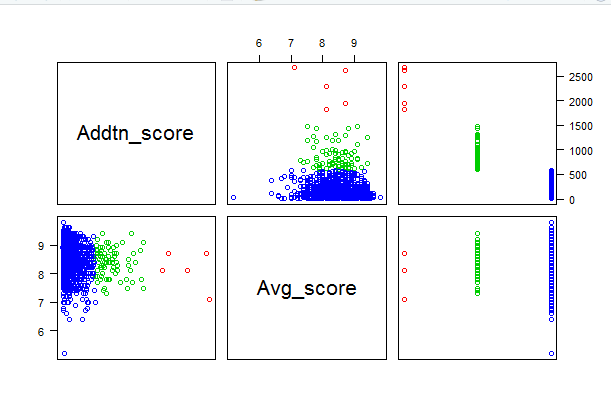
library(ggplot2)

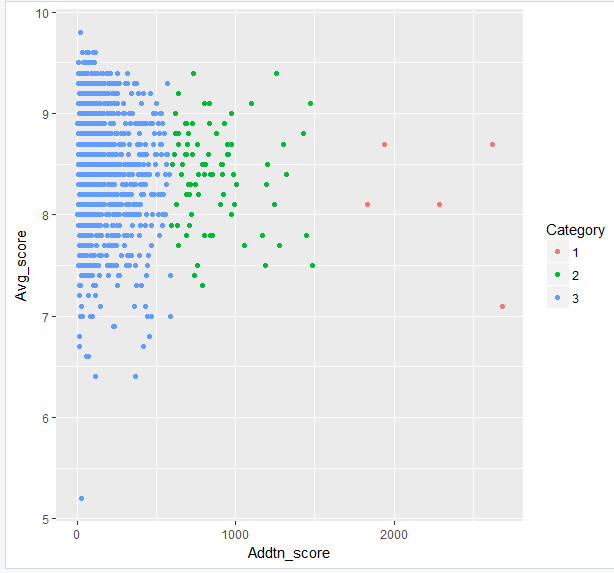
#You can plot the graph and cluster centroid using the following command.

plot(hotel\_metadata, col =(clusters$cluster +1) , pch=1, cex=1, las=1)

ggplot(hotel\_metadata, aes(Addtn\_score,Avg\_score, color = Category)) + geom\_point()

Pair plot





Very less and countable dots(red) are seen in this plot which have average avg\_score but high additional score.

So we can remove them as outlier .

Next, we noticed that most of dots are ploted in a way that inspite of having more additional score, they have moralless same avg score. So we cannot come to any decision about the satisfaction level.

So we should not proceed with this kind of clustering for recommendation.

**Region based recommendation system**

(region)

hotel\_metadata <- read.csv("D:/R - research/Hotel Recommendation/region.csv")

View(hotel\_metadata)

str(hotel\_metadata)

nrow(hotel\_metadata)

#preprocess data to avoid null values

hotel\_metadata <- na.omit(hotel\_metadata)

summary(hotel\_metadata)

#Elbow Method for finding the optimal number of clusters

set.seed(123)

# Compute and plot wss for k = 2 to k = 15.

k.max <- 15

data <- hotel\_metadata

wss <- sapply(1:k.max,

function(k){kmeans(data, k, nstart=50,iter.max = 15 )$tot.withinss})

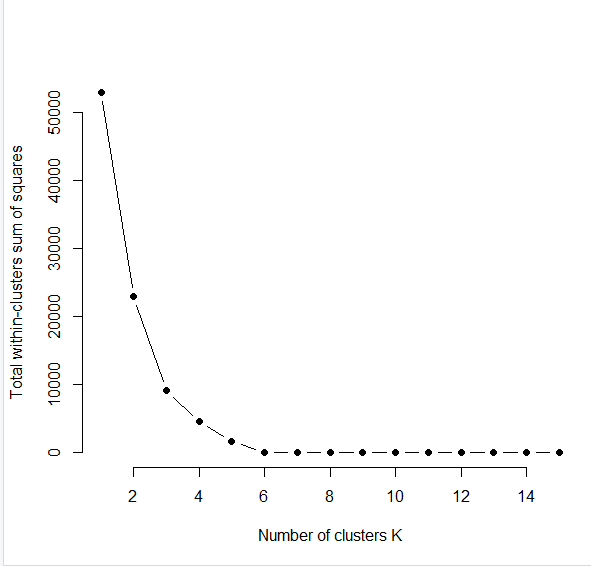
wss

plot(1:k.max, wss,

type="b", pch = 19, frame = FALSE,

xlab="Number of clusters K",

ylab="Total within-clusters sum of squares")



We can consider K=5

#apply Kmeans

set.seed(20)

clusters <- kmeans(hotel\_metadata,5)

# Save the cluster number in the dataset as column 'Category'

hotel\_metadata$Category <- as.factor(clusters$cluster)

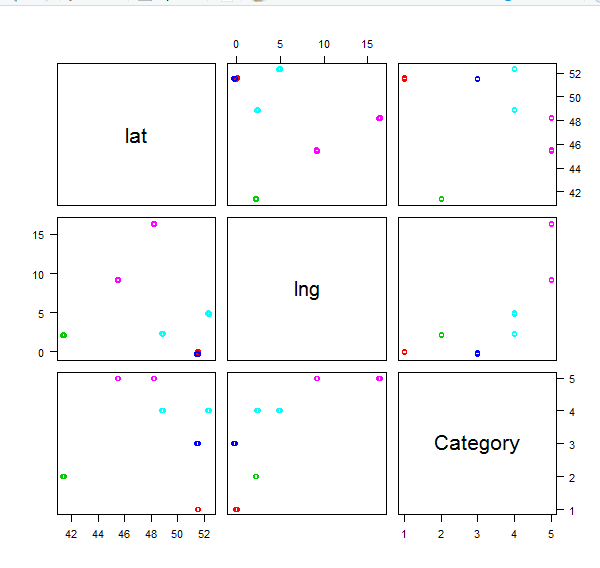
# Inspect 'clusters'

str(clusters)

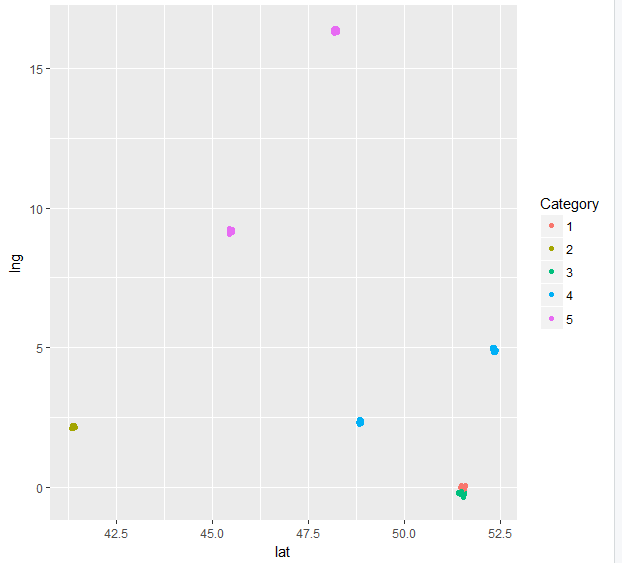
library(ggplot2)

#You can plot the graph and cluster centroid using the following command.

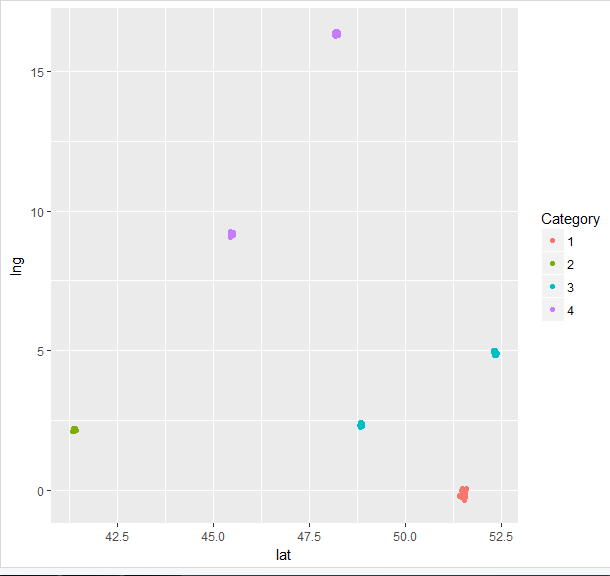
plot(hotel\_metadata, col =(clusters$cluster +1) , pch=1, cex=1, las=1)



ggplot(hotel\_metadata, aes(lat,lng, color = Category)) + geom\_point()



Lets try k=4



It’s looking perfect.

Let’s take export

write.csv(hotel\_metadata,"D:/R - research/Hotel Recommendation/hotel\_region.csv", quote = FALSE, row.names = TRUE)

We shall prepare a excel hotel\_region recommend.csv

If any hotel is booked by user then the other same category hotels will be recommended with the hotel.