Identify Luxury Hotel Groupings with Bayesian Clustering

Katherine Botz, Juan Luis Gómez Chanclón

Introduction

This project aims to analyze hotel reviews pulled from the website Booking.com, in order to identify any natural clustering using Bayesian methods. The dataset contains 515,000 customer reviews and rating of 1493 luxury hotels across Europe.

In the original dataset there is a wealth of information, including location of the hotel (Address, Longitude and Latitude), the overall score of the hotel, the text and classification (Positive or Negative) of each review itself as well as specific information about each reviewer's stay at the hotel, and reviewer's country of origin.

When visiting websites like Booking.com to read reviews and select a hotel, a customer must consider a number of options. We are interested in the characteristics of the hotels listed on such websites to see if natural groupings of hotels emerge. We have limited the dataset to the following variables for analysis:

- Hotel
- Location: City, Country (Latitude, Longitude)
- Average Score
- Total Number of Reviews
- Total Negative Word Counts
- Total Positive Word Counts
- Total Number of Reviews the Reviewer Has Given
- The Score that the Reviewer Gave

It should be mentioned that the cities included in this dataset are:

- Amsterdam, Netherlands
- Barcelona, Spain
- London, United Kingdom
- Milan, Italy
- Paris, France
- Vienna, Austria

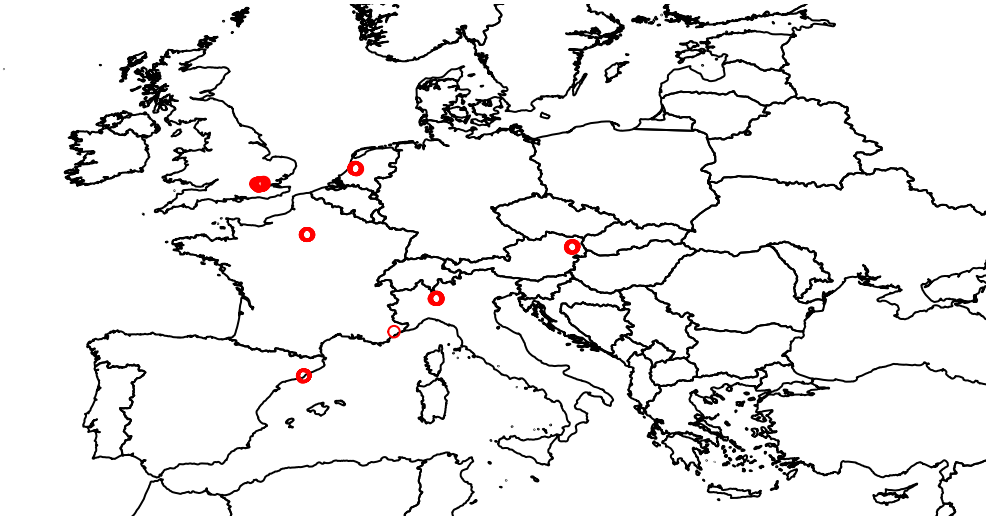
Since this dataset is quite large, for ease of computation we are going to summarize and average the individual review metrics for each hotel. In total there are 1475 hotels in the dataset across these six cities.

Analysis

First let's take a look at the hotels on a map, to better visualize the representation of each city in the dataset.

```
newmap <- getMap(resolution = "high")
plot(newmap, xlim = c(-12, 35), ylim = c(35, 60), asp = 1,
     main = "Locations of the Studied Hotels")
points(hotels$lng, hotels$lat, col = "red", cex = .8)
```

Locations of the Studied Hotels



We will run an MCMC algorithm to obtain a sample from the posterior, using a fixed number of mixture components, K, for the Prior. Now, we're not sure what is the natural clustering or grouping of hotels in our dataset, nor how many clusters is best to represent the data. Therefore we will try a range of $K = 2, \ldots, 9$ for the number of clusters and calculate the Deviance Information Criterion (DIC) for each fit. The best model will have the lowest DIC.

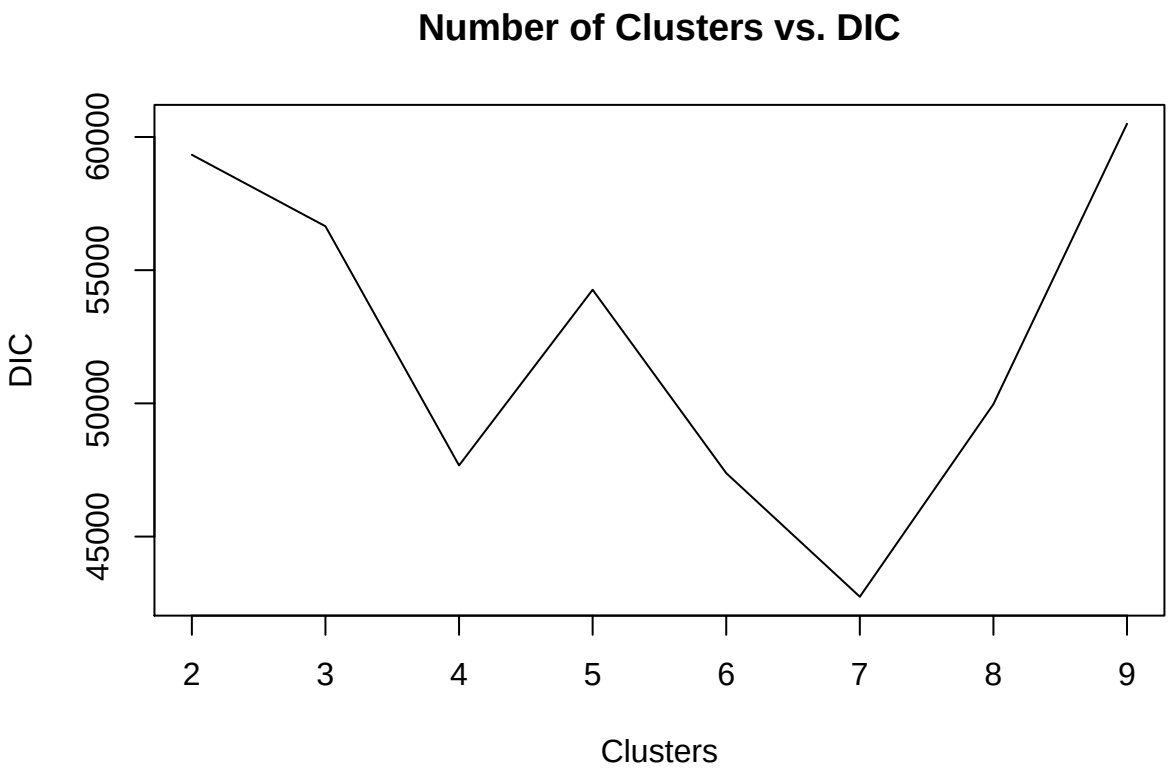
```
set.seed(123)

VARS <- names(hotels)[1:7]
nMCMC <- c(burn=5000, keep=10000, thin=5, info=1000)

#A finite, known constant equal to k:
DIC_fixed <- c()
for (k in 2 : 9){
  Prior <- list(priorK = "fixed", Kmax = k)
  fit <- NMixMCMC(y0 = hotels[, VARS], prior = Prior, nMCMC = nMCMC,
                  scale = list(shift=0, scale=1), PED = F)
  DIC_fixed <- c(DIC_fixed, fit$DIC$DIC)
}
```

Now we plot the DIC compared to the number of clusters, to determine the optimal number of clusters for this dataset.

```
plot(2:9, DIC_fixed, type='l', main="Number of Clusters vs. DIC",
     xlab="Clusters", ylab="DIC")
```

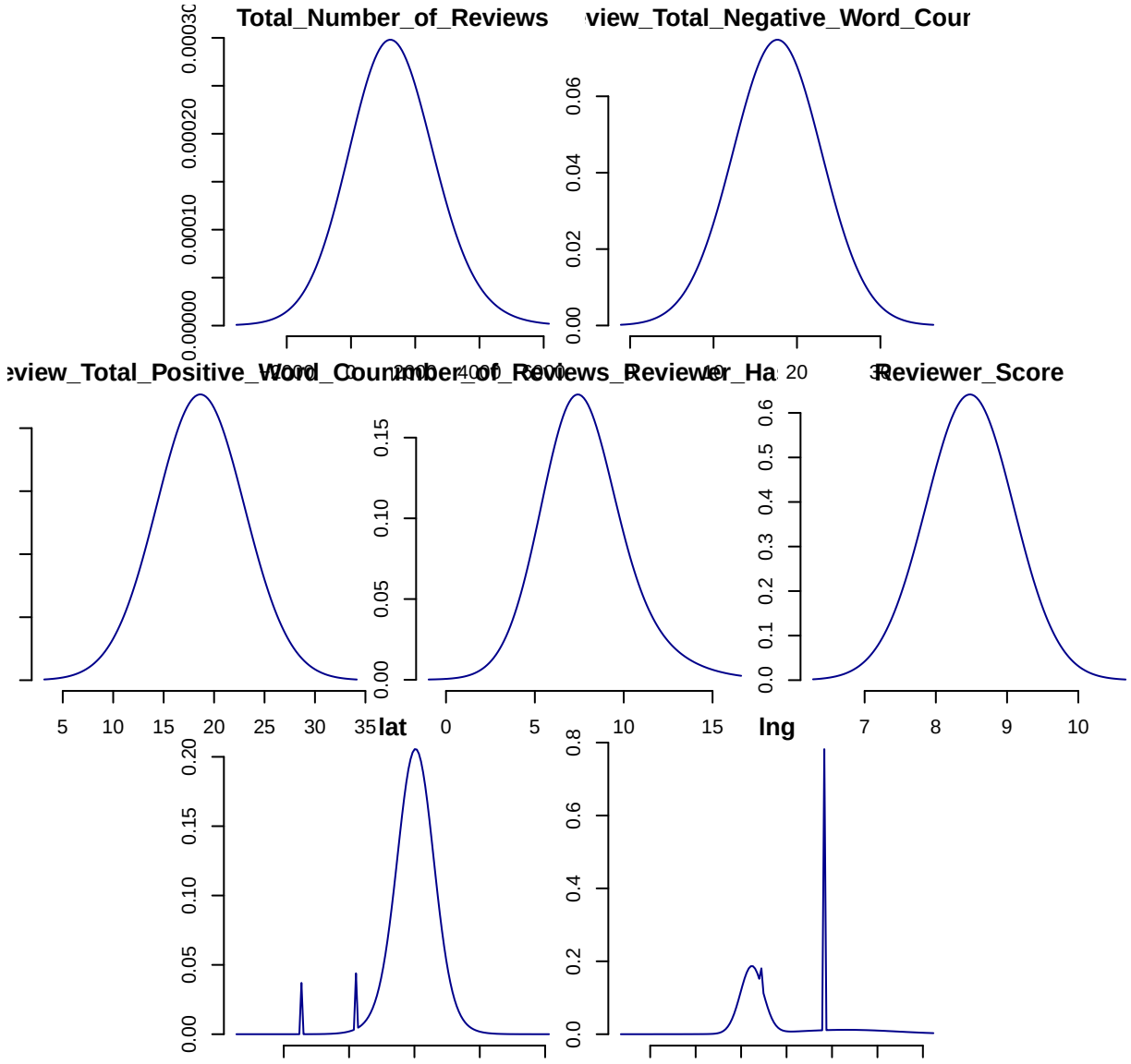


Even though the model with 7 clusters has the best DIC, we find that the optimal number of clusters is 4. This is to account for any bias in the DIC due to a greater number of clusters.

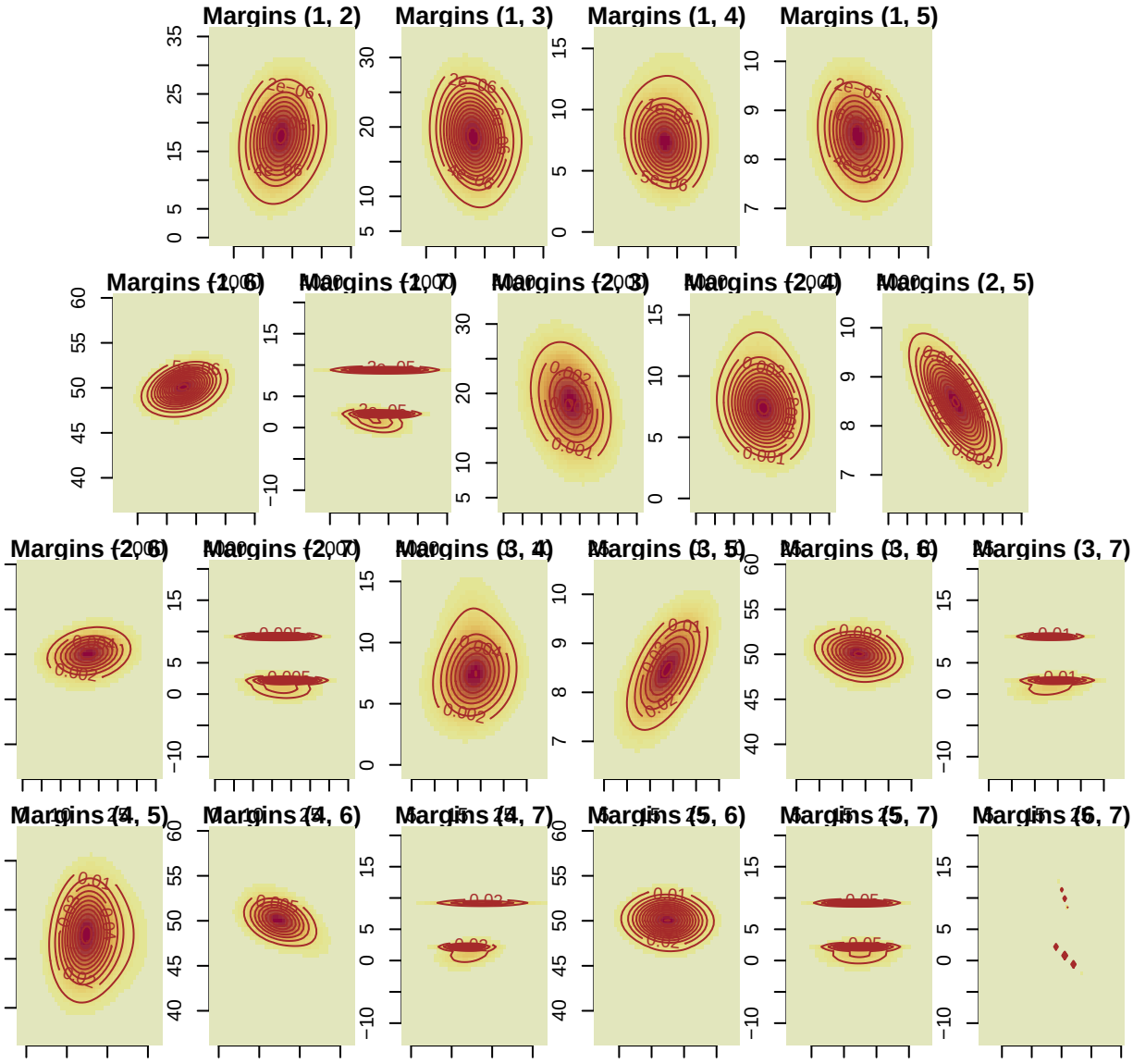
```
Prior <- list(priorK = "fixed", Kmax = 4)
fit <- NMixMCMC(y0 = hotels[, VARS], prior = Prior, nMCMC = nMCMC,
                scale = list(shift=0, scale=1), PED = F)
```

For the optimal 4 clusters, we plot the estimations of the univariate and pairwise bivariate marginal posterior predictive density for each marginal variable.

```
par(mar=c(1,1,1,1))
pdens1 <- NMixPredDensMarg(fit, lgrid=150)
plot(pdens1, main=VARS, xlab=VARS)
```



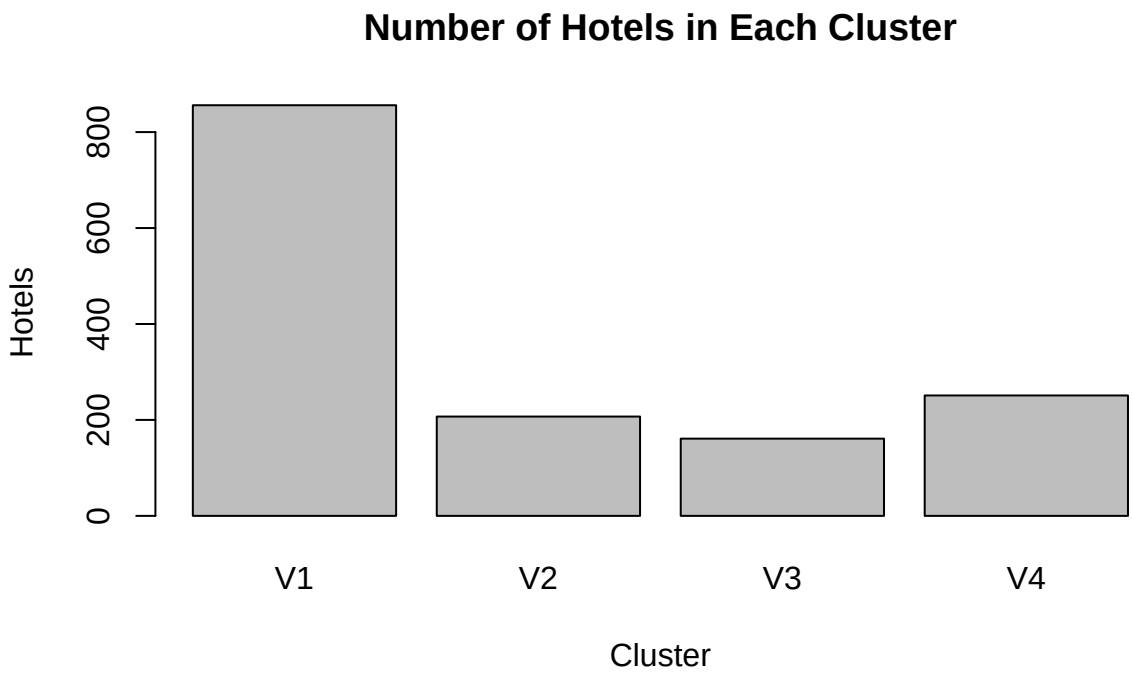

```
pdens2 <- NMixPredDensJoint2(fit)
plot(pdens2, xylab=VARS)
```



To get a better idea of the hotel assignment to each cluster, we analyze the metrics of the clusters themselves. Below you can see the number of hotels in each cluster, and the distribution of hotels across clusters.

```
##
##  V1  V2  V3  V4
## 856 207 161 251
```

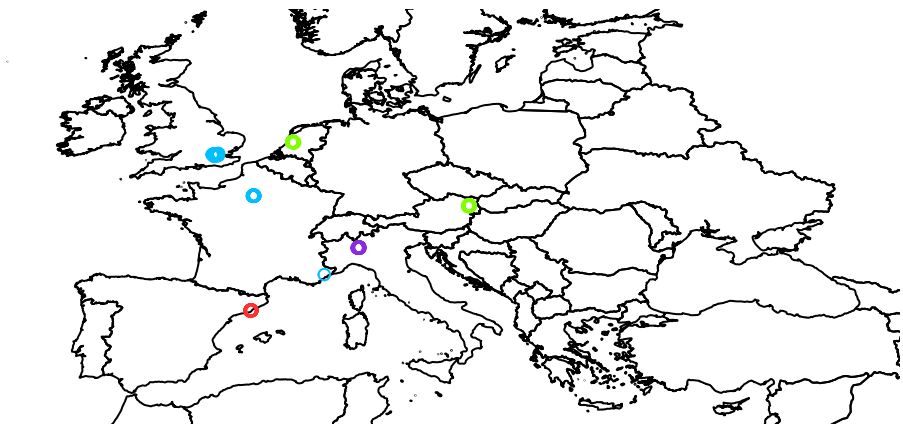
```
##
##        V1        V2        V3        V4
## 0.5803390 0.1403390 0.1091525 0.1701695
```



Now we analyze the characteristics of the hotels in each cluster to have a better idea of what each clustering represents. Let's plot the map of Europe to visualize the distribution of clusters geographically.

```
colors.cluster <- c("deepskyblue1","firebrick1",
                    "blueviolet", "chartreuse")[clusters$V2]

newmap <- getMap(resolution = "high")
plot(newmap, xlim = c(-12, 35), ylim = c(35, 60), asp = 1,
     main = "Locations of the Studied Clusters")
points(hotels$lng, hotels$lat, col = colors.cluster, cex = .8)
```



From this view it appears that the clusters have been assigned based on location. To be sure, let's look at the hotels in London alone.

```
plot(hotels[hotels$lat > 51 & hotels$lng < 0, ]$lng,
     hotels[hotels$lat > 51 & hotels$lng < 0, ]$lat,
     col = colors.cluster, main = "London Hotels Clustering Distribution",
     xlab="Longitude", ylab="Latitude")
```

Now we can see that hotels from the same city actually belong to different clusters. This indicates that the `lat` and `lon` attributes are not the main clustering attributes. To get an idea of the different characteristics of each cluster, we look at the mean of each attribute in each cluster.

```
fit$poster.mean.mu
```

```
##           m1       m2       m3       m4       m5       m6        m7
## j1 1057.643 17.25936 18.38788 7.209149 8.487369 50.09249  1.180045
## j2 1418.502 18.21623 19.23296 7.738887 8.476075 42.06388  3.313190
## j3 1604.805 17.81087 18.27252 9.434025 8.318966 45.31919  8.325339
## j4 1823.131 18.18967 19.19810 9.031916 8.482982 49.42770 11.274080
```

Now we can characterize each of the clusters as follows:

- **Cluster #1 (Blue)**: Here we have hotels with the lowest total number of reviews. This can mean two things, either these hotels are not very visited, or they are new hotels, so not many reviews are stored. The positive reviews usually have more words than the negative ones (as happens in all the other clusters). Also, we observe that usually, the reviewers of these hotels are people that don't have a lot of reviews on the webpage. Finally, even though the location isn't the most indicative clustering feature, we can see that these hotels are usually in London. This last affirmation can be easily proved taking a look at the `London Hotels clustering Distribution` plotted before, as it can be seen how most of the hotels are blue.
- **Cluster #2 (Red)**: Hotels with, on average, 50% more reviews compared to the hotels in Cluster #1. Also, these hotels are the ones in which the reviewers have written longer reviews, but as happened before, we can observe that those reviewers are usually inexperienced people. The average hotel in this cluster is placed in the Barcelona area.
- **Cluster #3 (Violet)**: This is a very similar cluster to #2. The main differences are that these hotels have more reviews, on average, and less words in their reviews. Other than that, the reviewers of these hotels are the most experienced ones, as they have, on average, 9.5 written reviews. The principal geographical location is Milan.
- **Cluster #4 (Green)**: The most reviewed hotels are placed in this community. This could be an

indicator that this cluster gathers the hotels that are older or bigger, and therefore have the largest amount of commentaries. Reviewers that have written these reviews are more experienced people, and the word count for positive and negative reviews is, more or less, the same as in the other clusters. If we needed to place this community in a map, they would be mostly placed around Vienna, in Austria.

Conclusions

Using a dataset of luxury European hotel reviews from Booking.com, we have done a clustering analysis using Bayesian methods. For this dataset we have found that the optimal number of clusters is 4, and analyzed the characteristics that define each grouping. We have found groupings based on higher overall score, number of reviews for each hotel, geographical location, and number of words (postive or negative) in each review. Practically, each group of hotels represents a type of hotel that the customer might prefer when searching for a hotel on the website.

References

https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe