Understanding User Ratings

Contents

Introduction

In this problem, we will use a dataset comprised of google reviews on attractions from 23 categories. Google user ratings range from 1 to 5 and average user ratings per category is pre-calculated. The data set is populated by capturing user ratings from Google reviews. Reviews on attractions from 23 categories across Europe are considered. Each observation represents a user.

Dataset: ratings.csv

Our dataset has the following columns:

- ***userId***: a unique integer identifying a user
- ***churches***, ***resorts***, ***beaches***,.., ***monuments__***, ***gardens***: the average rating that this user has rated any attraction corresponding to these categories. For example, the user with userID = User 1 has parks = 3.65, which means that the average rating of all the parks this user rated is 3.65. It can be assumed that if an average rating is 0, then that is the average rating. It is not the case that the user has not rated that category.

In this problem, we aim to cluster users by their average rating per category. Hence, users in the same cluster tend to enjoy or dislike the same categories.

Exercices

Problem 1 : Exploratory Data Analysis

Read the dataset ratings.csv into a dataframe called ratings.

```
## 'data.frame':    5456 obs. of  24 variables:
##  $ userid       : chr  "User 1" "User 2" "User 3" "User 4" ...
##  $ churches     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ resorts      : num  0 0 0 0.5 0 0 5 5 5 5 ...
##  $ beaches      : num  3.63 3.63 3.63 3.63 3.63 3.63 3.63 3.63 3.64 3.64 ...
##  $ parks        : num  3.65 3.65 3.63 3.63 3.63 3.63 3.63 3.63 3.64 3.64 ...
##  $ theatres     : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ museums      : num  2.92 2.92 2.92 2.92 2.92 2.92 2.92 2.92 2.92 2.92 ...
##  $ malls        : num  5 5 5 5 5 5 3.03 5 3.03 5 ...
##  $ zoo          : num  2.35 2.64 2.64 2.35 2.64 2.63 2.35 2.63 2.62 2.35 ...
##  $ restaurants  : num  2.33 2.33 2.33 2.33 2.33 2.33 2.33 2.33 2.32 2.32 ...
##  $ pubs         : num  2.64 2.65 2.64 2.64 2.64 2.65 2.64 2.64 2.63 2.63 ...
##  $ burger_shops : num  1.69 1.69 1.69 1.69 1.69 1.69 1.68 1.68 1.67 1.67 ...
##  $ hotels       : num  1.7 1.7 1.7 1.7 1.7 1.69 1.69 1.69 1.68 1.67 ...
##  $ juice_bars   : num  1.72 1.72 1.72 1.72 1.72 1.72 1.71 1.71 1.7 1.7 ...
##  $ art_galleries: num  1.74 1.74 1.74 1.74 1.74 1.74 1.75 1.74 0.75 0.74 ...
##  $ dance_clubs  : num  0.59 0.59 0.59 0.59 0.59 0.59 0.59 0.6 0.6 0.59 ...
##  $ pools        : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0 0 ...
##  $ gyms         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bakeries     : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0 ...
##  $ spas         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ cafes        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ view_points  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ monuments    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ gardens      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##     userid             churches        resorts         beaches
##  Length:5456        Min.   :0.000   Min.   :0.000   Min.   :0.000
##  Class :character   1st Qu.:0.920   1st Qu.:1.360   1st Qu.:1.540
##  Mode  :character   Median :1.340   Median :1.905   Median :2.060
##                     Mean   :1.456   Mean   :2.320   Mean   :2.489
##                     3rd Qu.:1.810   3rd Qu.:2.683   3rd Qu.:2.740
##                     Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##      parks           theatres        museums          malls
##  Min.   :0.830   Min.   :1.120   Min.   :1.110   Min.   :1.120
##  1st Qu.:1.730   1st Qu.:1.770   1st Qu.:1.790   1st Qu.:1.930
##  Median :2.460   Median :2.670   Median :2.680   Median :3.230
##  Mean   :2.797   Mean   :2.959   Mean   :2.893   Mean   :3.351
##  3rd Qu.:4.093   3rd Qu.:4.312   3rd Qu.:3.840   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##       zoo         restaurants         pubs        burger_shops
##  Min.   :0.860   Min.   :0.840   Min.   :0.810   Min.   :0.780
##  1st Qu.:1.620   1st Qu.:1.800   1st Qu.:1.640   1st Qu.:1.290
##  Median :2.170   Median :2.800   Median :2.680   Median :1.690
##  Mean   :2.541   Mean   :3.126   Mean   :2.833   Mean   :2.078
##  3rd Qu.:3.190   3rd Qu.:5.000   3rd Qu.:3.530   3rd Qu.:2.285
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##                                                  NA's   :1
##      hotels        juice_bars    art_galleries    dance_clubs
##  Min.   :0.770   Min.   :0.760   Min.   :0.000   Min.   :0.000
##  1st Qu.:1.190   1st Qu.:1.030   1st Qu.:0.860   1st Qu.:0.690
##  Median :1.610   Median :1.490   Median :1.330   Median :0.800
##  Mean   :2.126   Mean   :2.191   Mean   :2.207   Mean   :1.193
##  3rd Qu.:2.360   3rd Qu.:2.740   3rd Qu.:4.440   3rd Qu.:1.160
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##      pools             gyms           bakeries           spas
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00
##  1st Qu.:0.5800   1st Qu.:0.5300   1st Qu.:0.5200   1st Qu.:0.54
##  Median :0.7400   Median :0.6900   Median :0.6900   Median :0.69
##  Mean   :0.9492   Mean   :0.8224   Mean   :0.9698   Mean   :1.00
##  3rd Qu.:0.9100   3rd Qu.:0.8400   3rd Qu.:0.8600   3rd Qu.:0.86
##  Max.   :5.0000   Max.   :5.0000   Max.   :5.0000   Max.   :5.00
##
##      cafes          view_points      monuments        gardens
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.5700   1st Qu.:0.740   1st Qu.:0.790   1st Qu.:0.880
##  Median :0.7600   Median :1.030   Median :1.070   Median :1.290
##  Mean   :0.9658   Mean   :1.751   Mean   :1.531   Mean   :1.561
##  3rd Qu.:1.0000   3rd Qu.:2.070   3rd Qu.:1.560   3rd Qu.:1.660
##  Max.   :5.0000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##                                                   NA's   :1
```

1.1

How many users are in the dataset?

```
## [1] 5456
```

Answer : 5456

1.2

How many categories are rated in the dataset?

```
## [1] 23
```

Answer : 23

1.3

Note that there are some NA's in the data. **Which columns have missing data?**

```
##     userid             churches        resorts         beaches
##  Length:5456        Min.   :0.000   Min.   :0.000   Min.   :0.000
##  Class :character   1st Qu.:0.920   1st Qu.:1.360   1st Qu.:1.540
##  Mode  :character   Median :1.340   Median :1.905   Median :2.060
##                     Mean   :1.456   Mean   :2.320   Mean   :2.489
##                     3rd Qu.:1.810   3rd Qu.:2.683   3rd Qu.:2.740
##                     Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##      parks          theatres        museums          malls
##  Min.   :0.830   Min.   :1.120   Min.   :1.110   Min.   :1.120
##  1st Qu.:1.730   1st Qu.:1.770   1st Qu.:1.790   1st Qu.:1.930
##  Median :2.460   Median :2.670   Median :2.680   Median :3.230
##  Mean   :2.797   Mean   :2.959   Mean   :2.893   Mean   :3.351
##  3rd Qu.:4.093   3rd Qu.:4.312   3rd Qu.:3.840   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##       zoo         restaurants         pubs        burger_shops
##  Min.   :0.860   Min.   :0.840   Min.   :0.810   Min.   :0.780
##  1st Qu.:1.620   1st Qu.:1.800   1st Qu.:1.640   1st Qu.:1.290
##  Median :2.170   Median :2.800   Median :2.680   Median :1.690
##  Mean   :2.541   Mean   :3.126   Mean   :2.833   Mean   :2.078
##  3rd Qu.:3.190   3rd Qu.:5.000   3rd Qu.:3.530   3rd Qu.:2.285
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##                                                  NA's   :1
##      hotels        juice_bars    art_galleries    dance_clubs
##  Min.   :0.770   Min.   :0.760   Min.   :0.000   Min.   :0.000
##  1st Qu.:1.190   1st Qu.:1.030   1st Qu.:0.860   1st Qu.:0.690
##  Median :1.610   Median :1.490   Median :1.330   Median :0.800
##  Mean   :2.126   Mean   :2.191   Mean   :2.207   Mean   :1.193
##  3rd Qu.:2.360   3rd Qu.:2.740   3rd Qu.:4.440   3rd Qu.:1.160
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##      pools             gyms           bakeries           spas
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00
##  1st Qu.:0.5800   1st Qu.:0.5300   1st Qu.:0.5200   1st Qu.:0.54
##  Median :0.7400   Median :0.6900   Median :0.6900   Median :0.69
##  Mean   :0.9492   Mean   :0.8224   Mean   :0.9698   Mean   :1.00
##  3rd Qu.:0.9100   3rd Qu.:0.8400   3rd Qu.:0.8600   3rd Qu.:0.86
```

```
##  Max.   :5.0000   Max.   :5.0000   Max.   :5.0000   Max.   :5.00
##
##      cafes         view_points      monuments        gardens
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.5700   1st Qu.:0.740   1st Qu.:0.790   1st Qu.:0.880
##  Median :0.7600   Median :1.030   Median :1.070   Median :1.290
##  Mean   :0.9658   Mean   :1.751   Mean   :1.531   Mean   :1.561
##  3rd Qu.:1.0000   3rd Qu.:2.070   3rd Qu.:1.560   3rd Qu.:1.660
##  Max.   :5.0000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##                                                   NA's   :1
```

Answer :

1. resorts
2. parks
3. museums
4. malls
5. restaurants
6. **burger_shops**
7. juice_bars
8. dance_clubs
9. bakeries
10. cafes
11. **gardens**

1.4

What will happen if NA values are replaced with the value 0?

1. **Categories with missing values will be penalized.**
2. Categories with missing values will be rewarded.
3. The dataset and task will not be affected. This is the most fair way to handle the missing values.

1.5

To deal with the missing values, we will simply remove the observations with the missing values first (there are more sophisticated ways to work with missing values, but for this purpose removing the observations is fine since we do not lose a significant amount of observations). Run the following code:

ratings = ratings[rowSums(is.na(ratings)) == 0,]

```
## [1] 5454
```

How many users are there now?

Answer : 5454

1.6

Which category has the highest mean score?

```
## malls
##     8
```

Answer :

1. resorts
2. beaches
3. theatres
4. **malls**
5. juice_bars
6. drama
7. hotels
8. gyms

_Problem 2 : Preparing the Data

2.1

Before performing clustering on the dataset, **which variable(s) should be removed?**

1. gyms
2. **userid**
3. burger_shops and gardens
4. Not enough information

2.2

Remove the necessary column from the dataset and rename the new data frame points.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.810   1.640   2.680   2.833   3.527   5.000
```

Now, we will normalize the data.

What will the maximum value of pubs be after applying mean-var normalization? Answer without actually normalizing the data.

1. 5
2. 1
3. **Not enough information**

2.3

Normalize the data using the following code:

library(caret)

preproc = preProcess(points)

pointsnorm = predict(preproc, points)

What is the maximum value of juice_bars after the normalization?

```
## [1] 1.782152
```

Answer : 1.782152

Problem 3 : Clustering

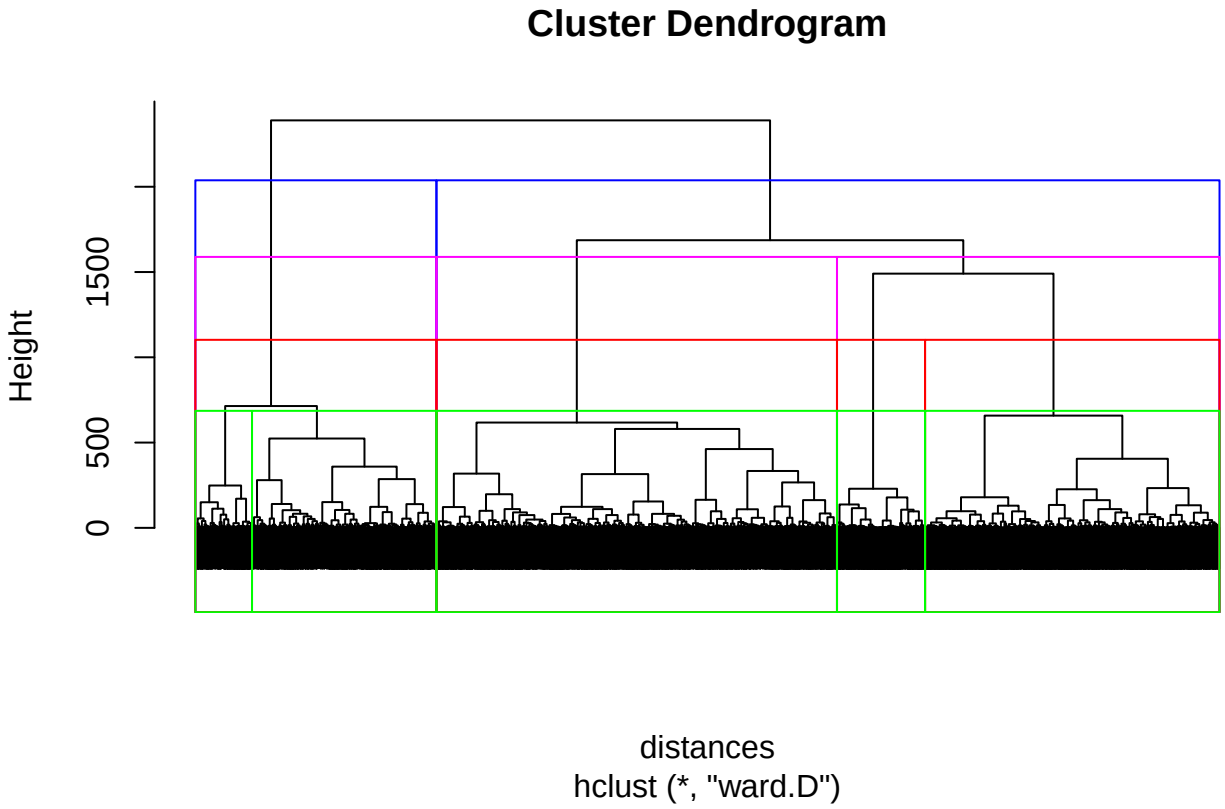
3.1

Create a dendogram using the following code:

distances = dist(pointsnorm, method = “euclidean”)

dend = hclust(distances, method = “ward.D”)

plot(dend, labels = FALSE)



3.1.1 Based on the dendrogram, **how many clusters do you think would NOT be appropriate for this problem?**

Answer :

1. 2
2. 3
3. 4
4. **5**

3.1.2 Based on this dendogram, in choosing the number of clusters, **what is the best option?**

Answer : 4

3.2

Set the random seed to 100, and run the k-means clustering algorithm on your normalized dataset, *setting the number of clusters to 4*.

How many observations are in the largest cluster?

```
## [1] 2424 1942  286  802
```

Answer : 1996

Problem 4 : Conceptual Questions

4.1

True or False: **If we ran k-means clustering a second time without making any additional calls to set.seed, we would expect every observation to be in the same cluster as it is now.**

Answer : FALSE

4.2

True or False: **K-means clustering is sensative to outliers.**

Answer : TRUE

4.3

Why do we typically use cluster centroids to describe the clusters?

1. The cluster centroid gives the values of every single observation in the cluster, and therefore exactly describes the cluster.
2. **The cluster centroid captures the average behavior in the cluster, and can be used to summarize the general pattern in the cluster.**
3. The cluster centroid captures the average behavior in the cluster, relative to the other clusters. So by just computing a single cluster centroid, we can understand how the cluster differs from the other clusters.

4.4

Is "overfitting" a problem in clustering?

1. No, we don't have test data, so it is impossible to evaluate k-means out-of-sample
2. **Yes, at the extreme every data point can be assigned to its own cluster.**
3. It depends on the application.

4.5

Is "multicollinearity" a problem in clustering?

1. No, because we aren't trying to find coefficients in our model.
2. **Yes, multicollinearity could cause certain features to be overweighted in the distances calculations.**
3. It depends on the application.

Problem 5 : Understanding the Clusters

5.1

Which cluster has the user with the lowest average rating in restaurants?

```
## [1] -1.641056 -1.677908 -1.685278 -1.611574
```

1. Cluster 1
2. Cluster 2
3. Cluster 3
4. **Cluster 4**

5.2

Which of the clusters is best described as "users who have mostly enjoyed churches, pools, gyms, bakeries, and cafes"?

```
## [[1]]
##   churches      pools       gyms   bakeries      cafes
## -0.4868612 -0.2536922 -0.2221465 -0.2095564 -0.3022523
##
## [[2]]
##    churches       pools        gyms    bakeries       cafes
##  0.07599018 -0.21912910 -0.28662692 -0.28127739 -0.15966984
##
## [[3]]
##    churches       pools        gyms    bakeries       cafes
##  0.57016791 -0.05316375  0.09146060  0.67502295  0.77209223
##
## [[4]]
## churches    pools     gyms bakeries    cafes
## 1.084178 1.316339 1.332861 1.073752 1.024838
```

1. **Cluster 1**
2. Cluster 2
3. Cluster 3
4. Cluster 4

5.3

Which cluster seems to enjoy being outside, but does not enjoy as much going to the zoo or pool?

```
## [[1]]
##        zoo      pools
##  0.5199747 -0.2536922
##
## [[2]]
##        zoo      pools
## -0.1868644 -0.2191291
```

```
##
## [[3]]
##         zoo       pools
## -0.80480971 -0.05316375
##
## [[4]]
##        zoo      pools
## -0.8321102  1.3163385
```

1. Cluster 1
2. Cluster 2
3. Cluster 3
4. **Cluster 4**