The Battle of Neighborhoods

Introduction

How excited when the holiday season is coming, and you plan to have a wonderful trip with your family. And then the first problem you need take time to tackle is where to go. You may have many different options, the places you would like to visit, and they all look attractive.

Sometimes it is quite hard to choose a desired one and convince your partner too.

An application that can compare the optional destinations and determine how similar or dissimilar they are would be useful for this case. Practically, the app can group the destinations into different categories, and thus it will help you make a decision based on the grouped result.

E.g. you may have 5 possible destinations to travel: Tenerife, Santorini, Maldives, Phuket, Honolulu. They are all perfect islands for travel during vacation. It would be helpful to find out whether those islands are similar or dissimilar. E.g. If Maldives and Phuket fall into one group, but they are in different groups than Tenerife and Santorini, for the users who want to experience two different kinds of islands, they can choose one of them from Maldives and Phuket and another from Tenerife and Santorini.

Data

For each of places, we can get the neighborhoods, longitude and latitude value of the neighborhoods from https://www.geonames.org/.

E.g.

	Name	Country	Feature class	Latitude	Longitude
1	Emporeío Emborio,Emborion,Emboríon,Emporeio,Emporeion,Emporeío,Emporeíon,Εμπορείο,Εμπορειον	Greece, South Aegean Kykládes > Santorini	populated place population 1,946	N 36° 21' 29''	E 25° 26' 46''

We can read the result table into Pandas dataframe.

The latitude and longitude values are in the format as

N 36° 21' 29'', E 25° 26' 46''

We use the following formula to translate the latitude and longitude value into decimal format

Decimal Degrees = degrees + (minutes/60) + (seconds/3600)

After the transformation, we will get the dataframe like

1	Neighborhood	Country	Feature class	Latitude	Longitude
0	EmporeÃo Emborio	Greece, South AegeanKyklÃ¡des > Santorini	populated placepopulation 1,946	36.358056	25.446111
1	OÃa Apano Meria	Greece, South AegeanKyklÃ¡des > Santorini	populated placepopulation 3,376	36.462500	25.376111

Then for each of places, we can get one neighborhood dataframe and then we merge them into a big one as ‘Neighborhoods’. That is our data we use for analysis.

Methodology

We would like to compare those places and know if they are similar. The basic idea is we run the k-means clusters on the features of the places, if two places are in the same group, we consider them as similar.

We do the following of steps to get the features of places.

Get venues for each neighborhood

For each of neighborhood in the dataframe 'Neighborhoods', we call the foursquare api:

https://api.foursquare.com/v2/venues/search?client_id={{client_id}}&client_secret={{client_secret}}&v={{v}}&ll=&intent=browse&radius=10000&limit=100

The longitude and latitude of neighborhoods are got from the previous step, and the radius is using 5 km. The limit of result is 100.

The result would be dataframe of neighborhoods' venues that that a typical user is likely to checkin to at the provided. The information of a venue we used would be name and category. For example

venue name	venue category
Pico del Teide	Mountain Hut

Thus we get 100 venues for each of neighborhoods and eventually we get a new big dataframe of neighborhood venues

Get top 10 most common venue category for each neighborhood

We then one hot the venue's category for each venues, group them by the neighborhood and calculate the means for each category. For each neighborhood we get the top 10 most common venues by sorting the mean of category in descending order.

Run k-means clusters

Upon the resulted dataframe, we run k means clustering by using clustering number 5, add clustering labels for neighborhoods in the new dataframe.

Then we group again the dataframe by the place name and sum the number of neighborhoods for each of clustering labels.

E.g. For Tenerife, we got 5, 0, 32, 0, 8 for lables 0, 1, 2, 3, 4. It means in Tenerife, there are 5 neighborhoods under cluster 0, 32 under cluster 2 and 8 under cluster 4. 0 under cluster 1 and 3.

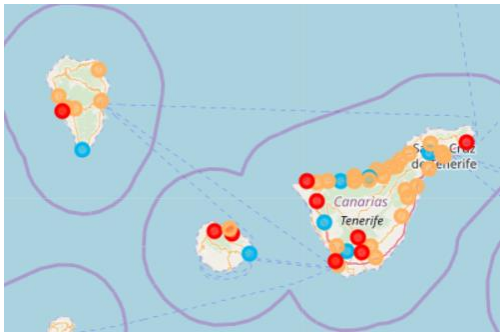
Thus we got a new dataframe: columns are the number of neighborhoods under each cluster and the rows are the places.

We again run the k-means clustering against the new dataframe by using cluster number 4.

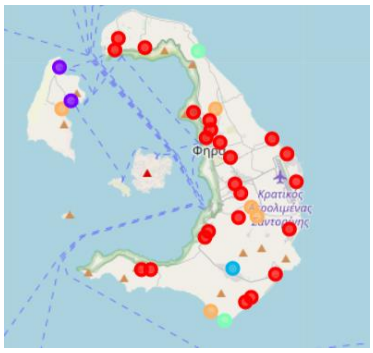
Result and discussion

For each of places' neighborhoods, we labeled them in the map as after running k-means clustering

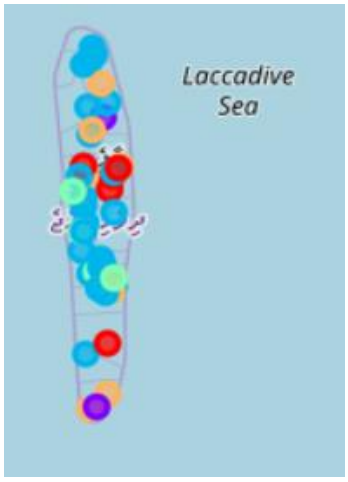
Tenerife



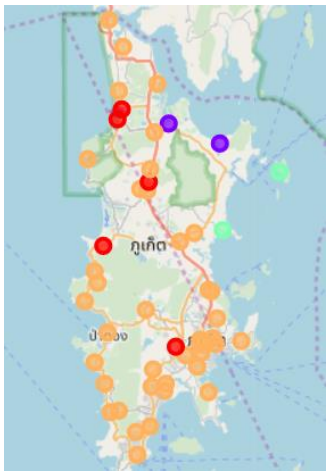
Santorini



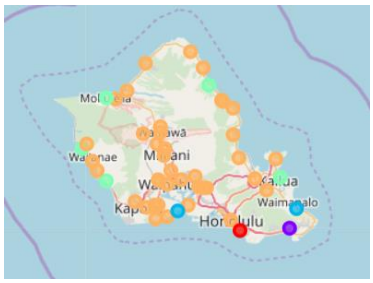
Maldives



Puhket



Honolulu



We got the dataframe of number of neighnohoods for each cluster lable as

Dataframe-1

	neighbohood cluster 0	neighbohood cluster 1	neighbohood cluster 2	neighbohood cluster 3	neighbohood cluster 4
Tenerife	9	0	7	0	29
Santorini	23	2	1	2	5
Maldives	5	2	23	5	9
Phuket	5	2	0	2	39
Honolulu	1	1	2	5	39

After running again the k-means clustering against this dataframe by using cluster number as 4, we got

Dataframe-2

	neighbohood cluster 0	neighbohood cluster 1	neighbohood cluster 2	neighbohood cluster 3	neighbohood cluster 4	label
Tenerife	9	0	7	0	29	3
Santorini	23	2	1	2	5	2
Maldives	5	2	23	5	9	1
Phuket	5	2	0	2	39	0
Honolulu	1	1	2	5	39	0

Phuket and Honolulu have the same label. This looks in inline with the fact that both of these two places have most neighborhoods under cluster 4.

Discussion

There are many of facts that can affect the results. E.g. as we get the neighborhoods for each places from https://www.geonames.org/ by search the feature class as “city, village” and limit the neighborhood as first 50 sorted by population, we might lost some popular venues because the neighborhood is not in the result. And also search the venues for each of neighborhood by specifying the radius as 5 km, some of the actual famous venues, sight, places may not be contained in the result e.g. the mountains that is not within 5km of any neighborhoods. In addition, we rely on the foursquare API to get the features of each place, this can create bias since restaurant, hotels may always be the most common places that the user will visit. Last, we run the k-means clustering against the Dataframe-1, which only contains 5 places as an example. By using the cluster number 4 where the places number is 5. Changing the number of clusters will affect the grouping. E.g. if we use 3 as cluster number

	neighbohood cluster 0	neighbohood cluster 1	neighbohood cluster 2	neighbohood cluster 3	neighbohood cluster 4	label
Tenerife	9	0	7	0	29	0
Santorini	23	2	1	2	5	2
Maldives	5	2	23	5	9	1
Phuket	5	2	0	2	39	0
Honolulu	1	1	2	5	39	0

Phuket and Honolulu will be still in the same label, which is the same as using cluster number 4. Tenerife is under also under label 0, which seems inline with the fact that it has also most neighborhoods under cluster 4.

If using cluster number as 2

	neighbohood cluster 0	neighbohood cluster 1	neighbohood cluster 2	neighbohood cluster 3	neighbohood cluster 4	label
Tenerife	9	0	7	0	29	0
Santorini	23	2	1	2	5	1
Maldives	5	2	23	5	9	1
Phuket	5	2	0	2	39	0
Honolulu	1	1	2	5	39	0

The Santorini and Maldives are in one group while Tenerife, Phuket and Honolulu are in the another.

Conclusion

In this project, we demonstrate how to compare the places by analyzing their neighborhood information. We used FourSquare APIs to get venues information of neighborhoods. K-means clustering is used to clustering the neighborhoods and the places. The results show the grouping the places are inline with the clustering the neighborhoods of each places. In our example, Phuket and Honolulu are similar than rest of three places. Tenerife is similar to Phuket and Honolulu, than Maldives and Santorini.