Capstone Project – The battle of neighborhoods

Lima foodies guide

Hidemi Kiyan

May, 2019

1. Introduction

1.1. Backgroud

Peru has become a magnet for tourists: according to experts, peruvian cuisine is the ultimate attraction. In the last 10 years, Peru has been recognized as one of the world's best culinary destinations, an attractive option for foodie tourism. The country's gastronomic boom owes a great deal to its biodiversity along with its multicultural heritage.

In the last years, there has been an increase of travelers who go to Lima for discovering the gastronomy of the city. According to data from the Peruvian Government, in recent years the amount of the budget from tourists used for local gastronomy has doubled from 5% to 10%, representing an income of USD 350 million per year.

1.2. Problem

From among many options of food places, it can be complicated to restaurant-goers or travelers making a choice.

1.3. Target audience

- Travelers
- Restaurant-goers

2. Data adquisition and cleaning

2.1. Get data of Peru districts from INEI

- Datasource: http://webinei.inei.gob.pe:8080/sisconcode/proyecto/index.htm?proyectoTitulo=UBIGEO&proyectoId=3
- Description: Get

2.2. Get the coordinates of Peru

- Datasource: Geopy
- Description: Get coordinates of Peru using geocoder class of Geopy client.

2.3. Restaurants in each district of Lima

- Datasource: Forsquare API
- Description: Get all the venues in each district of Lima by using Foursquare API.

3. Methodology

3.1.Data preparation

In order to explore restaurants in each district of Lima, we need a dataset that contains geographic information at district level as well as the the latitude and logitude coordinates of each one. For my convenience, I downloaded the file as .csv and placed it on my github repository: https://raw.githubusercontent.com/Hidemi-km/Capstone_Project/master/geodir-ubigeo-inei.csv

We use pandas library to create the initial dataframe:

```
peru_data=pd.read_csv('https://raw.githubusercontent.com/Hidemi-km/Capstone_Project/master/geodir-ubigeo-inei.csv')
peru_data.head()
```

	Ubigeo	Distrito	Provincia	Departamento	Poblacion	Superficie	Y	X
0	10101	Chachapoyas	Chachapoyas	Amazonas	29171	153.78	-6.2294	-77.8714
1	10102	Asuncion	Chachapoyas	Amazonas	288	25.71	-6.0317	-77.7122
2	10103	Balsas	Chachapoyas	Amazonas	1644	357.09	-6.8375	-78.0214
3	10104	Cheto	Chachapoyas	Amazonas	591	56.97	-6.2558	-77.7003
4	10105	Chiliquin	Chachapoyas	Amazonas	687	143.43	-6.0778	-77.7392

Once tha data is loaded, we procceed to rename the columns so that they make sense. Then, clean the dataset to remove a few unnecessary columns and rows and make sure we only have tha data that corresponds to Lima:

```
peru_data.rename(columns={'Ubigeo':'Código','Y':'Latitud', 'X':'Longitud'}, inplace=True)
peru_data=peru_data.drop(columns=['Poblacion'])
peru_data.columns
```

```
Index(['Código', 'Distrito', 'Provincia', 'Departamento', 'Superficie',
       'Latitud', 'Longitud'],
      dtype='object')
```

```
lima_data=peru_data[peru_data['Provincia']=='Lima']
```

```
lima_data.head()
```

	Código	Distrito	Provincia	Departamento	Superficie	Latitud	Longitud
1280	150101	Lima	Lima	Lima	21.98	-12.0467	-77.0322
1281	150102	Ancon	Lima	Lima	285.45	-11.7764	-77.1703
1282	150103	Ate	Lima	Lima	77.72	-12.0256	-76.9242
1283	150104	Barranco	Lima	Lima	3.33	-12.1494	-77.0247
1284	150105	Breña	Lima	Lima	3.22	-12.0567	-77.0536

Next, we will get the coordinates of Peru using geocoder class from Geopy class as follow:

```
address = 'Lima, PE'

geolocator = Nominatim(user_agent="peru_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Peru are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Peru are -12.0621065, -77.0365256.
```

3.2.Exploratory data analysis

Let's create a map with Folium library to visualize geographic details of Lima districts.

Then we are going to start utilizing the Foursquare API to get the top 100 venues that are in Lima within a radius of 1000 meters. Since we are interested only in restaurants, we will create a dataframe with only venues that have the word "Restaurant" in "Venue Category".

```
restaurants_venues= lima_venues[lima_venues['Venue Category'].str.contains('Restaurant')]
restaurants_venues.reset_index(drop=True)
restaurants_venues.head()
```

	District	District Latitude	District Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
14	Lima	-12.0467	-77.0322	Tanta	-12.045269	-77.031490	Peruvian Restaurant
18	Lima	-12.0467	-77.0322	Olamo Terraza	-12.046480	-77.030799	South American Restaurant
20	Lima	-12.0467	-77.0322	Hanna	-12.048474	-77.033224	Restaurant
22	Lima	-12.0467	-77.0322	Avellaneda' s Restaurant	-12.046282	-77.032920	Restaurant
26	Lima	-12.0467	-77.0322	Al Sazón de Walter	-12.047799	-77.033240	Restaurant

We can notice that 34 venue categories were returned by Foursquare and Lince is the district that concentrates more restaurants.

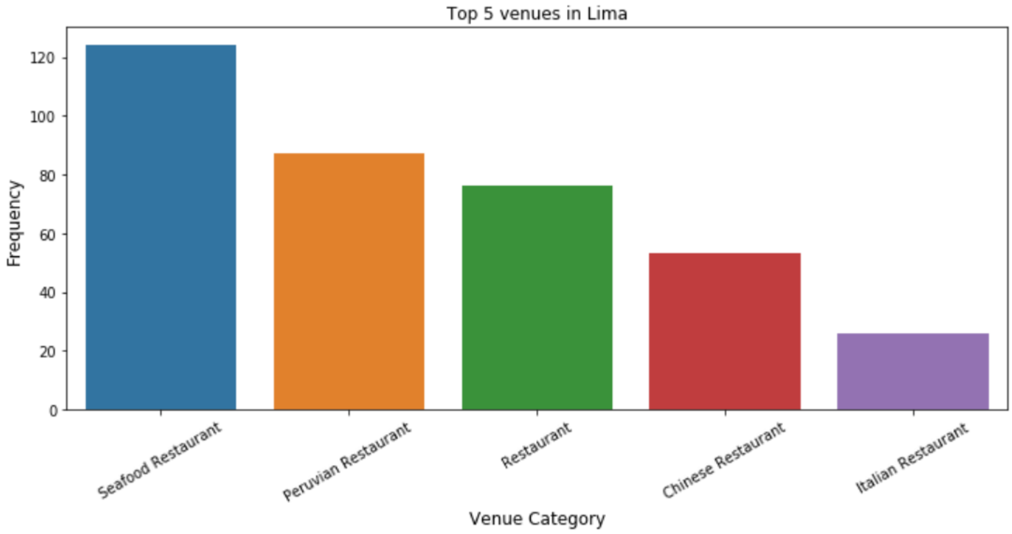
```
restaurants_venues['Venue Category'].value_counts().shape[0]
```

34

```
dist_venues=restaurants_venues['District'].value_counts().to_frame(name='Frequency')
dist_venues=dist_venues.reset_index()
dist_venues.rename(index=str, columns={'index': 'District'}, inplace=True)
dist_venues.head()
```

	District	Frequency
0	Lince	42
1	Jesus Maria	41
2	San Isidro	34
3	San Borja	28
4	Lima	27

Therefore, we can see that Seafood Restaurants top the charts as we can see in the plot below:



Next step is analyzing each district to get information about the top 5 venues of each one. To do that, we Will proceed as follows:

- Create a dataframe with pandas one hot encoding for the venue categories.

```
# one hot encoding
lima_onehot = pd.get_dummies(restaurants_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
lima_onehot['District'] = restaurants_venues['District']

# move neighborhood column to the first column
fixed_columns = [lima_onehot.columns[-1]] + list(lima_onehot.columns[:-1])
lima_onehot = lima_onehot[fixed_columns]

lima_onehot.head()
```

	District	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Belgian Restaurant	Cajun / Creole Restaurant	Cantonese Restaurant	Chinese Restaurant	Comfort Food Restaurant	...	Scandinav Restau
14	Lima	0	0	0	0	0	0	0	0	0	...	
18	Lima	0	0	0	0	0	0	0	0	0	...	
20	Lima	0	0	0	0	0	0	0	0	0	...	
22	Lima	0	0	0	0	0	0	0	0	0	...	
26	Lima	0	0	0	0	0	0	0	0	0	...	

- Use pandas groupby on districts column and calculate the mean of the frequency of ocurrence of each venue category.

```
district_groupby =lima_onehot.groupby('District').mean().reset_index()
district_groupby.head()
```

	District	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Belgian Restaurant	Cajun / Creole Restaurant	Cantonese Restaurant	Chinese Restaurant	Comfort Food Restaurant	...	Scandin: Restai
0	Ancon	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	...	
1	Ate	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	...	
2	Barranco	0.0	0.0	0.0	0.0	0.043478	0.0	0.0	0.043478	0.0	...	
3	Breña	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.055556	0.0	...	
4	Carabayllo	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	...	

- Output each district along with the top 5 most common venues.

```
num_top_venues = 5

for hood in district_groupby['District']:
    print("----"+hood+"----")
    temp = district_groupby[district_groupby['District'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Ancon----
                     venue  freq
0       Seafood Restaurant   1.0
1      American Restaurant   0.0
2       Mexican Restaurant   0.0
3  New American Restaurant   0.0
4      Peruvian Restaurant   0.0
```

3.3.Clustering

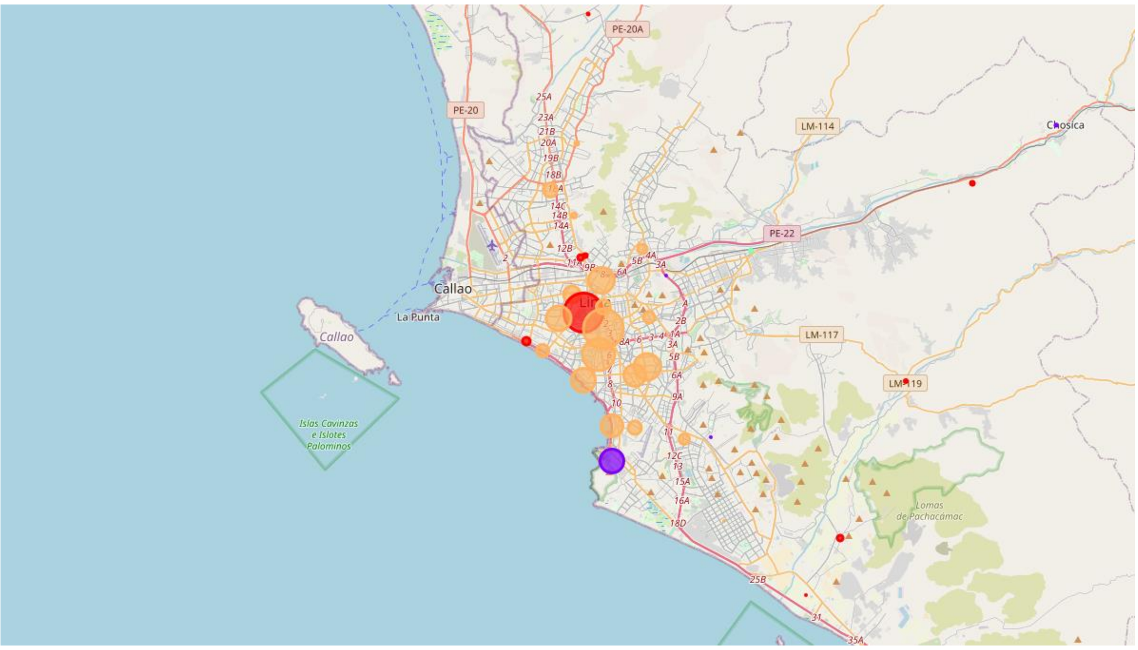
Finally, we will use K-Means to cluster all districts into 5 clusters based on the frequency of venue categories.

```
kclusters = 5

lima_clustering = district_groupby.drop('District', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(lima_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

```
array([1, 3, 4, 4, 0, 0, 1, 0, 4, 4], dtype=int32)
```

```
district_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
lima_merged = lima_data
lima_merged = lima_merged.join(district_venues_sorted.set_index('District'), on='Distrito')
lima_merged = lima_merged.dropna()
lima_merged.reset_index(drop=True)

lima_merged.head()
```

We can represent these 5 clusters in a leaflet map using Folium library, as below:



Where the radius of the circles represent the number of restaurants as most common venue for the corresponding district.

4. Results and discussion
   As result of the exploratory analysis and clustering, we find out some interesting insights that might be useful to travelers and restaurants-goers which are summerize below:
   - Seafood restaurants top the charts of most common venues.
   - Lince, Jesus Maria and San Isidro which are neighboring districts, concentrate the highest number of restaurants in Lima.
   - Miraflores, Barranco, San Isidro, Santiago de Surco and Lima which are the most touristics districts in Lima, fall under the same cluster.
   - The south of Lima is the zone with less number of restaurants.

5. Conclusion
   This Project give as a notion of how we can apply data-science in real scenarios. In these case, we used data to cluter districts in Lima based on venue categories. As result, we got 5 clusters that can help travelers or restaurant-goers to improve their culinary experience.
   We can realize that advancement of technologies is not only revolutionizing aspects of the business, but also those related to daily life.