Neighborhoods of London and Paris

IBM Capstone Project

1 Introduction

London and Paris are two of the most popular touristic cities in Europe. For the first time in 10 years, London has been beaten by Paris in the battle for visitors. The French capital received the highest number of tourists of any European city in 2019, attracting 19.1 million foreign visitors. Despite Paris's tourism growth, visitor spending was still higher in London: overnight international visitor spend for the latter was 16.5 billion dollars last year, compared to 14 billion dollars in the former. The objective of this project is to group the attractions of London and Paris respectively and draw insights to explain the growth in tourism in Paris and why Paris does not have a higher tourism revenue than London.

2 Data

The data for this project has been retrieved and processed through multiple sources.

2.1 Neighborhoods

The data of the neighborhoods in Paris is extracted from a JSON file found at French government website. For London, data is scraped from a Wikipedia webpage.

Code

```
# For Paris
paris_raw=pd.read_json("correspondances-code-insee-code-postal.json")
paris_raw.head()


# For London
url_london = "https://en.wikipedia.org/wiki/List_of_areas_of_London"
wiki_london_url = requests.get(url_london)
wiki_london_url

wiki_london_data = pd.read_html(wiki_london_url.text)
wiki_london_data
wiki_london_data = wiki_london_data[1]
wiki_london_data
```

2.2 Geocoding

ArcGIS Online enables you to connect people, locations, and data using interactive maps. Work with smart, data-driven styles and intuitive analysis tools that deliver location intelligence. Share your insights with the world or specific groups.

More specifically, we use ArcGIS to get the geo locations of the neighborhoods of London. The following columns are added to our initial dataset which prepares our data.

Code

```
from arcgis.geocoding import geocode
from arcgis.gis    import GIS
gis = GIS()

def get_x_y_uk(address1):
 lat_coords = 0
 lng_coords = 0
 g = geocode(address='{}, London, England,
adsGBR'.format(address1))[0]

 lng_coords = g['location']['x']
 lat_coords = g['location']['y']
 return str(lat_coords) +","+ str(lng_coords)

 # Checking sample data
 c = get_x_y_uk('SE2')
 c
```

2.3 Venue Data for Paris and London

From the location data obtained after Web Scraping and JSON file, the venue data is found out by passing in the required parameters to the FourSquare API. This and creates data frame to contain all the venue details along with the respective neighborhoods.

Code

```
LIMIT=100
def getNearbyVenues(names, latitudes,
longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes,
longitudes):
        print(name)

        # create the API request URL
        url =
'https://api.foursquare.com/v2/venues/explor
e?&client_id={}&client_secret={}&v={}&ll={},{}
&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)
        # make the GET request
        results = 
requests.get(url).json()["response"]['groups'][0
]['items']

        # return only relevant information for 
each nearby venue
        venues_list.append([(
            name, 
            lat, 
            lng, 
            v['venue']['name'], 
            v['venue']['location']['lat'], 
            v['venue']['location']['lng'],  
            v['venue']['categories'][0]['name']) for v 
in results])

    nearby_venues = pd.DataFrame([item for 
venue_list in venues_list for item in 
venue_list])
    nearby_venues.columns = ['Neighbourhood', 
                  'Neighbourhood Latitude', 
                  'Neighbourhood Longitude', 
                  'Venue', 
                  'Venue Latitude', 
                  'Venue Longitude', 
                  'Venue Category']
    
    return(nearby_venues)
```

3 Methodology

A thorough analysis of the principles of methods, rules, and postulates employed have been made in order to ensure the inferences to be made are as accurate as possible.

3.1 Folium

Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library. All cluster visualization is done with help of Folium which in turn generates a Leaflet map made using OpenStreetMap technology.

Code

```
# Creating the map of London
map_clusters_london = folium.Map(location=[london_lat_coords, london_lng_coords], zoom_start=12)

# set color scheme for the clusters
x = np.arange(k_num_clusters)
ys = [i + x + (i*x)**2 for i in range(k_num_clusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(london_data_nonan['latitude'], london_data_nonan['longitude'], london_data_nonan['borough'], london_data_nonan['Cluster Labels']):
    label = folium.Popup('Cluster ' + str(int(cluster) +1) + '\n' + str(poi) , parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[int(cluster-1)],
        fill=True,
        fill_color=rainbow[int(cluster-1)]
        ).add_to(map_clusters_london)

map_clusters_london
```

Code

```
# Creating the map of Paris
map_Paris= folium.Map(location=[paris_lat_coords, paris_lng_coords], zoom_start=12)
map_Paris


# adding markers to map
for latitude, longitude, borough, town in zip(paris_combined_data['Latitude'], paris_combined_data['Longitude'], paris_combined_data['nom_comm'], paris_combined_data['nom_dept']):
    label = '{}, {}'.format(town, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [latitude, longitude],
        radius=5,
        popup=label,
        color='Blue',
        fill=True,
        fill_opacity=0.8
        ).add_to(map_Paris)

map_Paris
```

3.2 One hot encoding

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. For the K-means Clustering Algorithm, all unique items under Venue Category are one-hot encoded.

Code

```
Paris_venue_cat = pd.get_dummies(venues_in_Paris[['Venue Category']], prefix="", prefix_sep="")
Paris_venue_cat

London_venue_cat = pd.get_dummies(venues_in_London[['Venue Category']], prefix="", prefix_sep="")
London_venue_cat
```

3.3 Top 10 most common venues

Due to high variety in the venues, only the top 10 common venues are selected and a new DataFrame is made, which is used to train the K-means Clustering Algorithm.

Code

```
# A function to get the top 10 most common venue
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Code

```
# create a new dataframe for Paris
neighborhoods_venues_sorted_paris = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted_paris['Neighbourhood'] = Paris_grouped['Neighbourhood']

for ind in np.arange(Paris_grouped.shape[0]):
    neighborhoods_venues_sorted_paris.iloc[ind, 1:] = return_most_common_venues(Paris_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted_paris.head()
```

Code

```
# create a new dataframe for London
neighborhoods_venues_sorted_london = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted_london['Neighbourhood'] = London_grouped['Neighbourhood']

for ind in np.arange(London_grouped.shape[0]):
   neighborhoods_venues_sorted_london.iloc[ind, 1:] = return_most_common_venues(London_grouped.
iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted_london.head()
```

3.4 K-means clustering

The venue data is then trained using K-means Clustering Algorithm to get the desired clusters to base the analysis on. K-means was chosen as the variables (Venue Categories) are huge, and in such situations K-means will be computationally faster than other clustering algorithms.

Code

```
paris_data_nonan.loc[paris_data_nonan['Cluster Labels'] == 1,
paris_data_nonan.columns[[1] + list(range(5, paris_data_nonan.shape[1]))]]

paris_data_nonan.loc[paris_data_nonan['Cluster Labels'] == 2, paris_data_nonan.columns[[1] +
list(range(5, paris_data_nonan.shape[1]))]]

…

paris_data_nonan.loc[paris_data_nonan['Cluster Labels'] == 5, paris_data_nonan.columns[[1] +
list(range(5, paris_data_nonan.shape[1]))]]


london_data_nonan.loc[london_data_nonan['Cluster Labels'] == 1,
london_data_nonan.columns[[1] + list(range(5, london_data_nonan.shape[1]))]]

london_data_nonan.loc[london_data_nonan['Cluster Labels'] == 2,
london_data_nonan.columns[[1] + list(range(5, london_data_nonan.shape[1]))]]

…

london_data_nonan.loc[london_data_nonan['Cluster Labels'] == 5,
london_data_nonan.columns[[1] + list(range(5, london_data_nonan.shape[1]))]]
```

4 Results and Discussion

Geographically, Paris is relatively small in size compared to London. The city is divided into 20 boroughs or “arrondissement” as they call it. Three of the five clusters show that the most common venue is a type of restaurant. Cluster 2 (7, 8, 14 and 17 arrondissement) is a French restaurant. . Cluster 3 (1-6, 9-11, 15, 18 and 19 arrondissement) is a French/Italian restaurant. . Cluster 5 (13 arrondissement) is a French restaurant. Furthermore, hotels, pubs and coffee shops are often the 2nd to 4th most common venue on these clusters. Museums, galleries and historical site are present throughout the different clusters.

London covers an area of 600 square miles, while Paris is squeezed into 40 square miles. The capital has 12 more boroughs than its French counterpart. The most common venue across all the clusters are coffee shops, hotel and pub. Also, cluster 3 and 5 shows that historic sites are accompanied with a lot bus stations. Although some restaurants are present in the clusters, there is a lack of English cuisines. However, the capital of England has an abundant of supermarket, grocery or convenient store.

As you can see from the clusters, the city of Paris offers a wide variety of cuisines and eateries. On top of that the city has a vast choice of cultural attractions like historical sites and museums. According to study made by Universitat Pompeu Fabra (Barcelona), cultural attractions and gastronomy are the critical elements that attract foreign visitors to a new country. London does have the cultural attractions like Paris but comes up short on the local cuisine. This explains the growth in tourism in Paris compared to London.

A pound is worth less than a euro and Paris attracted more visitors than London. Logically, Paris should have a higher tourism revenue. However, that is not the case. London generated 2.5 billion dollars more. You can see from the clusters that metro, bus or train station are often the top three most common venues. London has an extensive public transport which makes traveling easier. The city also has an abundant of restaurants are stores. According to study made by University of Surrey, consumer’s behavior in a foreign country is heavily influenced by the cost of accessibility. London offers a cheap way to get around the city and offers a variety of shops too. This explains why London has a higher revenue from tourists.

5 Conclusion

The purpose of this project was to explain the growth in tourism in Paris and why Paris does not have a higher tourism revenue than London. We answered both questions by clustering different neighborhoods through their similarities of venues. Cultural attractions and gastronomy are important to attract tourists and if you want to increase the potential revenue from each tourist, ease of transportation is a critical factor.