Battle of the Neighborhoods

Using Data Science to find the best neighbourhood to open a Chinese Restaurant in San Jose, CA

Bo Dong

This project aims to utilize all Data Science Concepts learned in the IBM Data Science Professional Course. We define a Business Problem, the data that will be utilized and using that data, we are able to analyze it using Machine Learning tools. In this project, we will go through all the processes in a step by step manner from problem designing, data preparation to final analysis and finally will provide a conclusion that can be leveraged by the business stakeholders to make their decisions.

Table of Contents

1. Introduction:

San Jose is a large city surrounded by rolling hills in Silicon Valley, a major technology hub in California's Bay Area. Architectural landmarks, from the 1883 Italianate-style Oddfellows building to Spanish Colonial Revival structures, make up the downtown historic district.

San José is the cultural, financial, and political center of Silicon Valley and the largest city in Northern California, by both population and area. San Jose is the county seat of Santa Clara County, the most affluent county in California and one of the most affluent counties in the United States. San Jose is notable as a center of innovation, for its affluence, Mediterranean climate, and extremely high cost of living. Its location within the booming high tech industry as a cultural, political, and economic center has earned the city the nickname "Capital of Silicon Valley".

San Jose is one of the wealthiest major cities in the United States and the world, and has the third-highest GDP per capita in the world (after Zürich, Switzerland and Oslo, Norway). The San Jose Metropolitan Area has the most millionaires and the most billionaires in the United States per capita. With a median home price of $1,085,000, San Jose has the most expensive housing market in the country and the fifth most expensive housing market in the world.

2. Target Audience

This project is aimed towards Entrepreneurs or Business owners who want to open a new Chinese Restaurant or grow their current business. The analysis will provide vital information that can be used by the target audience.

3. Data Overview

The data that will be required will be a combination of CSV files that have been prepared for the purposes of the analysis from multiple sources which will provide the list of San Jose's zip codes and corresponding latitude and longitude, and Venue data via Foursquare. The Venue data will help find which zip code is best suitable to open a Chinese restaurant.

First we will import all Python libraries.

```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!pip install geopy # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

!pip install folium # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
print('Libraries imported.')
```

Then we will download a csv file with all zip codes in San Jose. We need to clean up this data by removing zip codes from other places and removing zip codes with the same geo-point (latitude and longitude). The result is a list of 32 zip codes in San Jose.

```
df = pd.read_csv('https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-lo
df.drop(df[df['State'] != 'CA'].index, inplace = True)
df.drop_duplicates(subset=['geopoint'], inplace=True)
df.head()
```

Now we are ready to download the list of Chinese restaurants in San Jose using the zip code list from above. We will use API calls to FourSquare and pass in the CategoryID of Chinese Restaurant.

We will put the result in a dataframe for further processing.

```
CLIENT_ID = 'S3DCBMSDPSFP4NKZQ5AATH0B3ARVGWWTK1EIUPIGVER1EMEE' # your Foursquare ID
CLIENT_SECRET = 'GYOJDQOCXILZDUAOJLLVNAVKURJDFJ3VW5OAGWLIXUBY51X3' # your Foursquare Secret
version = '20201103' # Foursquare required field
limit = 100 # max response from Foursquare API
categoryId = '4bf58dd8d48988d145941735'  # Chinese Restaurant

responses=[]
for zip in df['Zip']:
    # create the API request URL
    url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&l
        CLIENT_ID,
        CLIENT_SECRET,
        version,
        limit,
        categoryId,
        zip
        )
    print(url)
    # make the GET request
    responses.append(requests.get(url).json()["response"]['groups'][0]['items'])

print("Foursquare requests are done.")
```

Now we have downloaded the list of Chinese Restaurants in San Jose, we need to clean up the list by removing duplicates and items belonging to other cities or categories.

```
venues = pd.DataFrame(columns=['name', 'lat', 'lng', 'city', 'state', 'zipcode', 'category'])

for i, response in enumerate(responses):
  for j, item in enumerate(response):
    v = item['venue']
    loc = v['location']
    if loc['city'] =='San Jose' and 'postalCode' in loc and len(v['categories']) > 0:
      venues.loc[len(venues)] = [v['name'], loc['lat'], loc['lng'], loc['city'], loc['state']

venues.drop_duplicates(subset=['name', 'lat', 'lng', 'city', 'zipcode', 'category'], keep='la
venues.drop(venues[~venues['category'].str.contains("Restaurant")].index, inplace=True)
venues.drop(venues[venues['category'] == "American Restaurant"].index, inplace=True)
venues.loc[venues['zipcode'] == "95122-1414", 'zipcode'] = '95122'

venues.reset_index(drop=True, inplace=True)
venues.head()
```

4. Methodology

Here we can use the K-Means Clustering machine learning method to group by zip code by the number of Chinese Restaurants in the zip code.

```
zipcode = venues.groupby('zipcode').count().sort_values(by='name', ascending=False)
scaler = StandardScaler()
scaled_features = scaler.fit_transform(zipcode)

kclusters = 3
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(scaled_features)
zipcode.insert(0, 'Cluster', kmeans.labels_)
zipcode.head()
```

Now we have the clustered the zip code by the number of Chinese restaurants in them, we can color code each restaurant on a map:

blue = high density (good zip code)

green = mid density (OK zip code)

red = low density (bad zip code)

```
sj_city = 'San Jose, CA, USA'
geolocator = Nominatim(user_agent="explorer")
sj_location = geolocator.geocode(sj_city)
sj_latitude = sj_location.latitude
sj_longitude = sj_location.longitude
print (sj_latitude,sj_longitude )

# set color scheme for the clusters blue - high density, green - mid density, red - low densi
rainbow = ['blue', 'green', 'red']

# create map of San Jose using latitude and longitude values
map = folium.Map(location=[sj_latitude, sj_longitude], zoom_start=10)
map = folium.Map(location=[sj_latitude, sj_longitude], zoom_start=10)
# add markers to map
for index, row in venues.iterrows():
    #print(index)
    cluster = zipcode.loc[row['zipcode']]['Cluster']
    folium.CircleMarker(
        [row['lat'], row['lng']],
        radius=5,
        popup=folium.Popup(row['name'], parse_html=True),
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7,
        parse_html=False
    ).add_to(map)
```

Now let's see the map in action.

```
map
```

5. Discussion

From the map above, we can see clearly that there are two neighbors with high concentration of Chinese Restaurants:

1. The first area is San Jose / Cupertino. This is the headquarter of Apple Inc, the most valuable company in the world. There are many Chinese engineers living in Cupertino, so there are many affluent families. The Chinese restaurants in this area are usually upscale and have highly educated clients.

2. The second area is along Murphy Ave in San Jose. This area is mostly residential with a few large shopping centers. The population is this is less affluent and the housing prices are much lower than Cupertino area. But the San Jose city has spent a great deal of money to revitalize this neighborhood. So the area is considered upcoming with great potential.

6. Conclusion

In a fast-moving world, there are many real-life problems or scenarios where data can be used to find solutions to those problems. Like seen in the example above, data was used to cluster neighborhoods in San Jose based on the number of Chinese Restaurants in 32 zip codes. The results can help a business to decide where to open a new Chinese Restaurant in San Jose.

I have made use of some frequently used python libraries to scrap web-data, use Foursquare API to get the listing of Chinese Restaurants in San Jose and put them on the Folium map.

Similarly, data can also be used to solve other problems, which most businesses face often. Potential for this kind of analysis in a real-life problem is discussed in great detail. Also, some of the drawbacks and chance for improvements to represent even more realistic pictures are mentioned.