Recommender System on Contractor Business

Richer Ge

October 21, 2020

1. Introduction

1.1 Background

In a city of your choice, if a contractor is trying to start their own business in Jinan, where would you recommend that they setup their office? This is a typical clustering problem with dataset labeled.

1.2 Business and Data Understanding

Obviously, it should be close to provider and Customer markets. And data needed should be geological location information about specific borough and the neighborhoods in that borough.

We assume it is in Jinan, China. And in China borough and the neighborhoods are not quiet distracted.

2. Data acquisition and cleaning

2.1 Data Sources

We will need data about different venues in different neighborhoods of that specific borough. In order to gain that information we will use "Foursquare" locational information. A typical request from Foursquare will be in this format:

1. CLIENT INFORMATION

2. GEO PARAMETERS

```
In [143]: LIMIT = 100 # limit of number of venues returned by Foursquare API
          radius = 3000 # define radius
          # create URL
          url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={
          },{}&radius={}&limit={}'.format(
              CLIENT_ID,
              CLIENT_SECRET,
              VERSION,
              neighborhood_latitude,
              neighborhood_longitude,
              radius,
              LIMIT)
          # display URL
          # url

In [144]: results = requests.get(url).json()
          #results
```

What's more, by requesting from the website about all post codes in China, I draw the paticular table on 'http://tools.2345.com/yb.htm'.

2.2 Data cleaning

Here is the data cleaning part in beautifulsoup to clean the nonetype and make everty elements unique.

```
# learn about copy and deepcopy
# pre: remove none type
none_type1='\xa0'
none_type2='邮政编码'
none_type3='市、县、区名'
while none_type1 in Borough:
    Borough.remove(none_type1)
while none_type1 in Postcode:
    Postcode.remove(none_type1)
while none_type2 in Postcode:
    Postcode.remove(none_type2)
while none_type3 in Borough:
    Borough.remove(none_type3)


# then integrate with common values
unique_p = set(Postcode)
print('num of unique Postal codes:', len(unique_p))
Postcode_u      = []
Borough_u       = []
for postcode_unique_element in unique_p:
    p_var = ''; b_var = '';
    for postcode_idx, postcode_element in enumerate(Postcode):
        if postcode_unique_element == postcode_element:
            p_var = postcode_element;
            b_var = Borough[postcode_idx]
            Postcode_u.append(p_var)
            Borough_u.append(b_var)
#Postcode_u
#Borough_u
#table_sd
```

```
num of unique Postal codes: 123
```

	Postcode	Borough	Latitude	Longitude
1	251600	商河县	37.32	117.15
20	250300	长清区	36.55	116.73
36	250200	章丘市	36.72	117.53
62	251400	济阳县	36.98	117.22
95	250100	历城区	36.68	117.07
115	250000	历下区	36.67	117.08
116	250000	市中区	35.40	116.58
117	250000	槐荫区	36.65	116.93
118	250000	天桥区	36.68	116.98
149	250400	平阴县	36.28	116.45

3. Exploratory Data Analysis

3.1 Find top 100 in Lixia

Identifying Postal Codes (and then Neighborhoods) in Jinan.

Connecting to Foursquare and Retrieving Locational Data for Each Venue in Every Neighborhood.

After finding the list of neighborhoods, we then connect to the Foursquare API to gather information about venues inside each and every neighborhood. For each neighborhood, we have chosen the radius to be 3000 meter. The two pictures below distributively

showing the clusters from the standpoint of geo location and common-sharing.

Out[183]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
历下区	3	3	3	3	3	3
历城区	2	2	2	2	2	2
天桥区	4	4	4	4	4	4
市中区	1	1	1	1	1	1
平阴县	1	1	1	1	1	1
章丘市	1	1	1	1	1	1

	Postcode	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
2	250200	章丘市	36.72	117.53	4	Park	Ice Cream Shop	Hotel	Hostel	Fast Food Restaurant
4	250100	历城区	36.68	117.07	1	Hostel	Fast Food Restaurant	Park	Ice Cream Shop	Hotel
5	250000	历下区	36.67	117.08	2	Chinese Restaurant	Convenience Store	Park	Ice Cream Shop	Hotel
6	250000	市中区	35.40	116.58	0	Ice Cream Shop	Park	Hotel	Hostel	Fast Food Restaurant
8	250000	天桥区	36.68	116.98	2	Hotel	Fast Food Restaurant	Arts & Crafts Store	Park	Ice Cream Shop
9	250400	平阴县	36.28	116.45	3	Hotel	Park	Ice Cream Shop	Hostel	Fast Food Restaurant

4. Predictive Modeling

Processing the Retrieved Data and Creating a DataFrome for All the Venues inside Jinan.

Then One-hot Encoding !

Applying one of Machine Learning Techniques (K-Means Clustering)

Now, we focus on the centers of clusters and compare them for their “Total Types”. The group which its center has the highest "Total Sum" will be our best recommendation to the contractor. {Note: Total Sum = Sum of Total Types}. This algorithm although is pretty straightforward yet is strongly powerful. We can see the below picture.

Out[215]:

	Arts & Crafts Store	Chinese Restaurant	Convenience Store	Fast Food Restaurant	Hostel	Hotel	Ice Cream Shop	Park	Total Sum
G1	0.000	0.000000	0.000000	0.000	0.0	0.00	1.0	0.0	1.0
G2	0.000	0.000000	0.000000	0.500	0.5	0.00	0.0	0.0	1.0
G3	0.125	0.000000	0.000000	0.125	0.0	0.75	0.0	0.0	1.0
G4	0.000	0.000000	0.000000	0.000	0.0	0.00	0.0	1.0	1.0
G5	0.000	0.666667	0.333333	0.000	0.0	0.00	0.0	0.0	1.0

Here we can easily see G1 has the greastest feature of glocery business.

5. Conclusions

Although some techniques are quite straightforward but powerful. And Although this is quite simple, further more I will use more structural datasets to evaluate NBA games to improve my data analysis ability.

Finally, I have one word for everybody sharing the course: Hard work pays off.