The Battle of the Neighborhoods

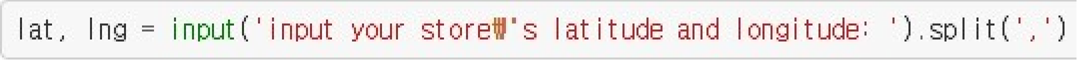
1. Introduction

Taking a walk, we may notice that the shops in certain places change frequently. In this case, we can see that stores of the same type are already thriving. In order to prevent the catastrophe of missing some stores of same category, the theme of this project was selected. I am going to make a recommendation system for person who wants to be a shopkeeper. This machine learning model will help people to know what kind of category they should avoid.

2. Data

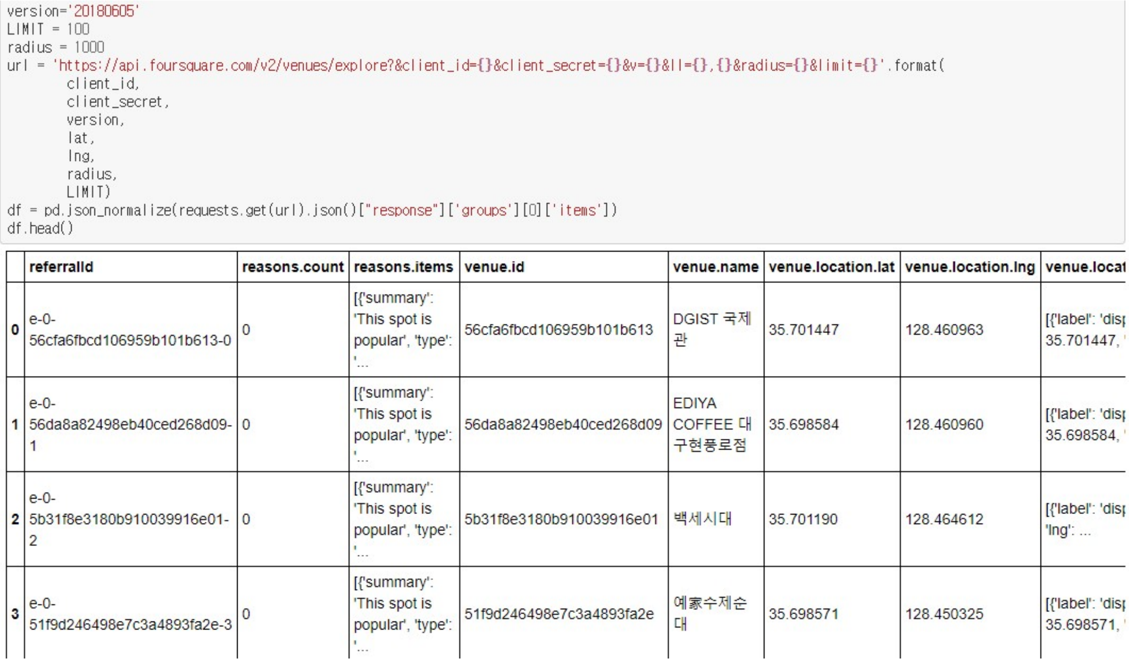
We will use two kinds of data. First, the user's store's latitude and longitude are entered. Second, get store information near your store's location via foursquare api. I am going to set radius as 1000 meter, limit results to 100.

2.1. First, get latitude and longitude of the store.



```
lat, lng = input('input your store₩'s latitude and longitude: ').split(',')
```

2.2. Second, using foursquare api, get a dataframe of stores nearby



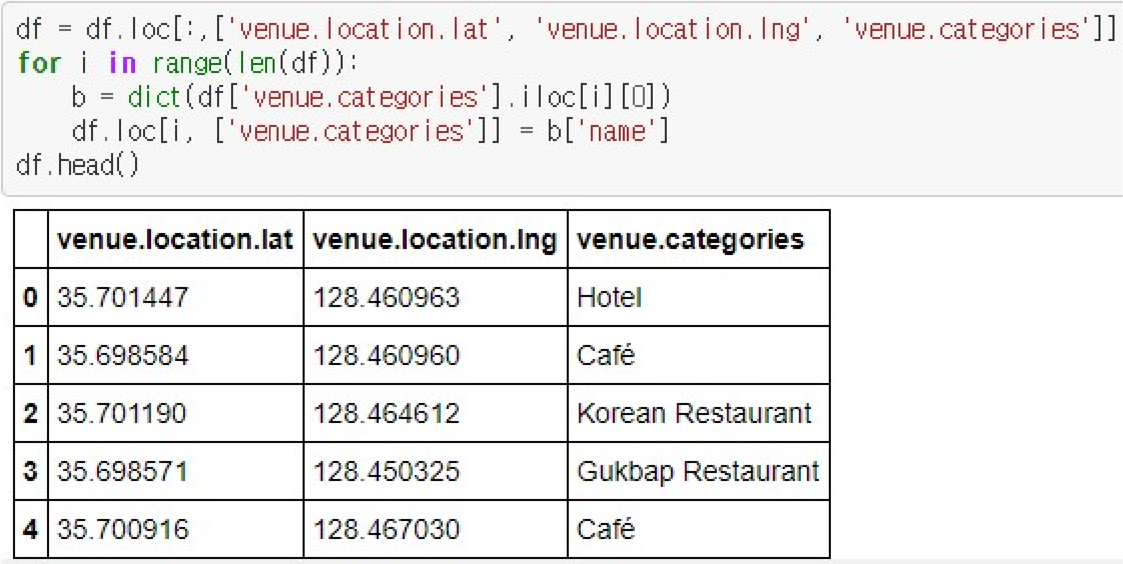
```
version='20180605'
LIMIT = 100
radius = 1000
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
        client_id,
        client_secret,
        version,
        lat,
        lng,
        radius,
        LIMIT)
df = pd.json_normalize(requests.get(url).json()["response"]['groups'][0]['items'])
df.head()
```

	referralId	reasons.count	reasons.items	venue.id	venue.name	venue.location.lat	venue.location.lng	venue.locat
0	e-0-56cfa6fbcd106959b101b613-0	0	[{'summary': 'This spot is popular', 'type': '...	56cfa6fbcd106959b101b613	DGIST 국제관	35.701447	128.460963	[{'label': 'disp 35.701447, '
1	e-0-56da8a82498eb40ced268d09-1	0	[{'summary': 'This spot is popular', 'type': '...	56da8a82498eb40ced268d09	EDIYA COFFEE 대구현풍로점	35.698584	128.460960	[{'label': 'disp 35.698584, '
2	e-0-5b31f8e3180b910039916e01-2	0	[{'summary': 'This spot is popular', 'type': '...	5b31f8e3180b910039916e01	백세시대	35.701190	128.464612	[{'label': 'disp 'lng': ...
3	e-0-51f9d246498e7c3a4893fa2e-3	0	[{'summary': 'This spot is popular', 'type': '...	51f9d246498e7c3a4893fa2e	예家수제순대	35.698571	128.450325	[{'label': 'disp 35.698571, '

3. Methodology

3.1. Preprocess

Sice algorithm only needs location and category, I left latitude, longitude and category data and eraised rest of dataframe.

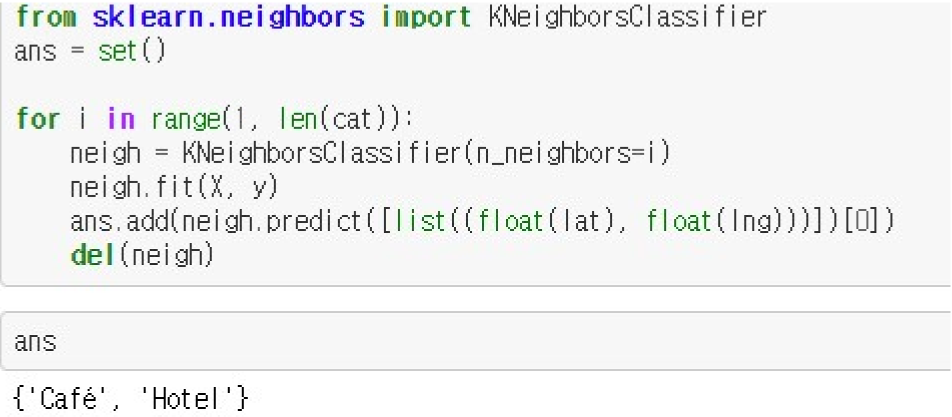


```
df = df.loc[:,['venue.location.lat', 'venue.location.lng', 'venue.categories']]
for i in range(len(df)):
    b = dict(df['venue.categories'].iloc[i][0])
    df.loc[i, ['venue.categories']] = b['name']
df.head()
```

	venue.location.lat	venue.location.lng	venue.categories
0	35.701447	128.460963	Hotel
1	35.698584	128.460960	Café
2	35.701190	128.464612	Korean Restaurant
3	35.698571	128.450325	Gukbap Restaurant
4	35.700916	128.467030	Café

3.2. Machine Learning

My goal was to give a list about coategories of store a person should avoid. So i used a k-nearest neighbors algorithm. First, I made a set of categories so I can get max number of K. K starting from 1 until length of set made right before, algorithm will determine what category will that place will fit in. Being classified in a specific category means that even if there is a store in that category, it does not have any advantage.



```
from sklearn.neighbors import KNeighborsClassifier
ans = set()

for i in range(1, len(cat)):
    neigh = KNeighborsClassifier(n_neighbors=i)
    neigh.fit(X, y)
    ans.add(neigh.predict([list((float(lat), float(lng)))])[0])
    del(neigh)
```

```
ans
```

```
{'Café', 'Hotel'}
```

4. Results

As cell right above, by machine learning user could get list of categories that he/she should avoid.

5. Discussion

So user who was planning to open the shop at (35.704713616784005, 128.45710541327045), should avoid Café and Hotel.

6. Conclusion

I only used KNN and didn't changed any hyperparameter except number of neighbors. If there were datasets that shows such as category of the shop, term of the shop survived, I think it would be able to make a model to predict the income of shop at specific place and shop of specific category.