Taste of Chicago:
A Research into The Taste of Different Neighborhoods

1. Introduction

1.1 Background

The City of Chicago is the 3rd largest city in the United States, it is home to about 2.7 million people. I lived there for several years in two neighborhoods – from Hyde Park in the southern suburb to West Loop in downtown. I found the types of restaurants so different – in Hyde Park there were mostly fast food restaurants while West Loop might be the most condensed neighborhood of fine dining restaurants. Therefore, I would like to understand the different taste between neighborhoods using data science techniques.

1.2 Business Problem

The problem I'm trying to analyze is: if someone is looking to open a restaurant in Chicago, where would I recommend that they open it?

The idea is to categorically segment the neighborhoods of Chicago into major clusters based on their food taste. I would then compare groups of neighborhoods by demographic statistics such as population density, per capita income and so on. From these two angles, I will get a better understanding of the taste and potential consumers' profile. After that, I can give specific recommendations to potential stakeholder based on the type and pricings of his restaurant.

How would we define an area's food taste? For this problem, we would utilize FourSquare API to find top 100 restaurants within each one neighborhood. We would group them by food types and aggregate weights of numbers of each food type.

1.3 Stakeholders

The result of this analysis can be utilized by a potential food vendor hoping to open a new restaurant. Also, it can be used to understand the distribution of different cultures and cuisines over Chicago.

2 Data

The following sources are used:

2.1 Stanford Digital Repository

https://purl.stanford.edu/xq082nw3443

The geographic and demographic data source of Chicago is downloaded from 'Stanford Digital Repository'. In the file 'hoods3155lite.dbf', it contains major US cities neighborhoods, latitudes, longitudes, average and medium household income and so on. For the purpose of this project, we will focus on the city of Chicago. We will keep and rename relevant features and discard the others. The features to keep are:

- 'Neighborhood': name of neighborhood

- 'Longitude': longitude
- 'Latitude': latitude
- 'POPDENSITY': population density
- 'DIVERSITY': diversity index
- 'PC_INCOME': per capita income
- 'MEDAGE_CY': median age
- 'UNEMPRT_CY': unemployment pct 2010
- 'MEDVAL_CY': median home value

	Neighborhood	Longitude	Latitude	POPDENSITY	DIVERSITY	PC_INCOME	MEDAGE_CY	UNEMPRT_CY	MEDVAL_CY
0	Chatham	-87.616624	41.7385	12681.364151	7.839623	23599	40.605660	19.645283	126001
1	North Center	-87.684523	41.9473	17814.894231	65.521154	39612	35.211538	10.348077	405797
2	O'hare	-87.847436	41.9633	6591.975000	34.500000	28952	44.060000	9.175000	247836
3	Washington Park	-87.617580	41.7916	13106.814286	11.428571	14888	29.214286	35.517857	194914
4	Garfield Ridge	-87.766976	41.7997	9271.435000	51.655000	22925	40.620000	12.911667	168838

2.2 Foursquare API

https://developer.foursquare.com/docs

Foursquare API, a location data provider, will be used to make API calls to retrieve data about venues in different neighborhoods. Venues retrieved from all the neighborhoods are categorized broadly into ‘Arts & Entertainment’, ‘College & University’, ‘Event’, ‘Food’, ‘Nightlife Spot’, ‘Outdoors & Recreation’, etc. Under each category, there are detailed subcategories. For example, ‘Food’ category contains ‘Fast Food’, ‘American Restaurant’, ‘Deli/Bodega’, ‘Pizza Place’ and so on, there are more than 200 food subcategories.

	name	categories	lat	lng
0	Dunkin'	Donut Shop	41.736741	-87.612562
1	Garrett Popcorn Shops	Snack Place	41.736535	-87.605829
2	Mather's More than a Café	Café	41.743548	-87.623089
3	Kam's Chop Suey	Chinese Restaurant	41.743330	-87.623933
4	Chipotle Mexican Grill	Fast Food Restaurant	41.735792	-87.625955

3 Methodology

- section which represents the main component of the report where you discuss and describe any
- exploratory data analysis that you did,
- any inferential statistical testing that you performed, if any, and
- what machine learnings were used and why.

3.1 Load Chicago Geographic and Demographic Data

I load the desired data from Stanford Digital Repository, the dataset's name is "U.S. Neighborhoods greenness measures and social variables". It contains variables related to (A) parks, open space, greenness, and "pavedness" (impervious surface) together with (B) a number of demographic variables from the 2010 U.S. census. For the purpose of my project, I only keep relevant geographic and demographic data and discard the others.
I downloaded the .rar package and find the major data file named 'hoods3155lite.dbf'. In order to load .dbf and convert them into DataFrame, I imported a method 'Dbf5' from python library 'simpledbf', which allows simple transformation from dbf to DataFrame.

```
dbf = Dbf5('hoods3155lite.dbf')
df = dbf.to_dataframe()
df.head()
```

	STATE	CITY	NAME	REGIONID	SHAPE_LENG	SHAPE_AREA	X	Y	REGION_ID	LA_CITY	...	MEDHINC_CY	MEDFINC_CY	AVGFINC_C
0	CA	Long Beach	Airport Area	272732.0	17308.184793	8.359173e+06	-118.154496	33.8167	272732	0	...	73346	72176	8505
1	CA	Long Beach	Alamitos Heights	272737.0	4385.328031	8.623082e+05	-118.125871	33.7738	272737	0	...	83046	98472	12889
2	CA	Long Beach	Belmont Heights	272933.0	7952.049738	2.498741e+06	-118.151191	33.7639	272933	0	...	66667	87687	10395
3	CA	Long Beach	Belmont Shore	113713.0	5727.509526	1.621976e+06	-118.137396	33.7589	113713	0	...	83425	99794	12360
4	CA	Long Beach	Bixby Area	272968.0	11518.915038	4.495448e+06	-118.176421	33.8405	272968	0	...	62332	67868	8071

5 rows × 44 columns

```
df.columns
```

```
Index(['STATE', 'CITY', 'NAME', 'REGIONID', 'SHAPE_LENG', 'SHAPE_AREA', 'X',
       'Y', 'REGION_ID', 'LA_CITY', 'REGIONID_1', 'DG_N', 'DG_NINV', 'DP_N',
       'DP_NINV', 'PCTPARK_N', 'MEANEQ_N', 'DG_MEAN', 'DP_MEAN', 'PCT_PARK',
       'MEAN_EQ', 'YOUNGFOLKS', 'POPDENSITY', 'DIVERSITY', 'PC_INCOME',
       'AVG_HINC', 'AVG_HVAL', 'PCT_OWN', 'PCT_RENT', 'PCT_WHITE',
       'PCT_HISPAN', 'PCT_BLACK', 'MEDAGE_CY', 'UNEMPRT_CY', 'MEDHINC_CY',
       'MEDFINC_CY', 'AVGFINC_CY', 'MEDVAL_CY', 'DG_STD', 'DP_STD', 'CENTROID',
       'NEW_GEOM', 'PAVED', 'CITY_PARKS'],
      dtype='object')
```

3.2 Data cleaning

For the purpose of this project, we will focus on the city of Chicago. We will keep and rename relevant features and discard the others. Now we obtain a DataFrame of Chicago with its 77 neighborhoods, geographic information and several important demographic info came from 2010 census.

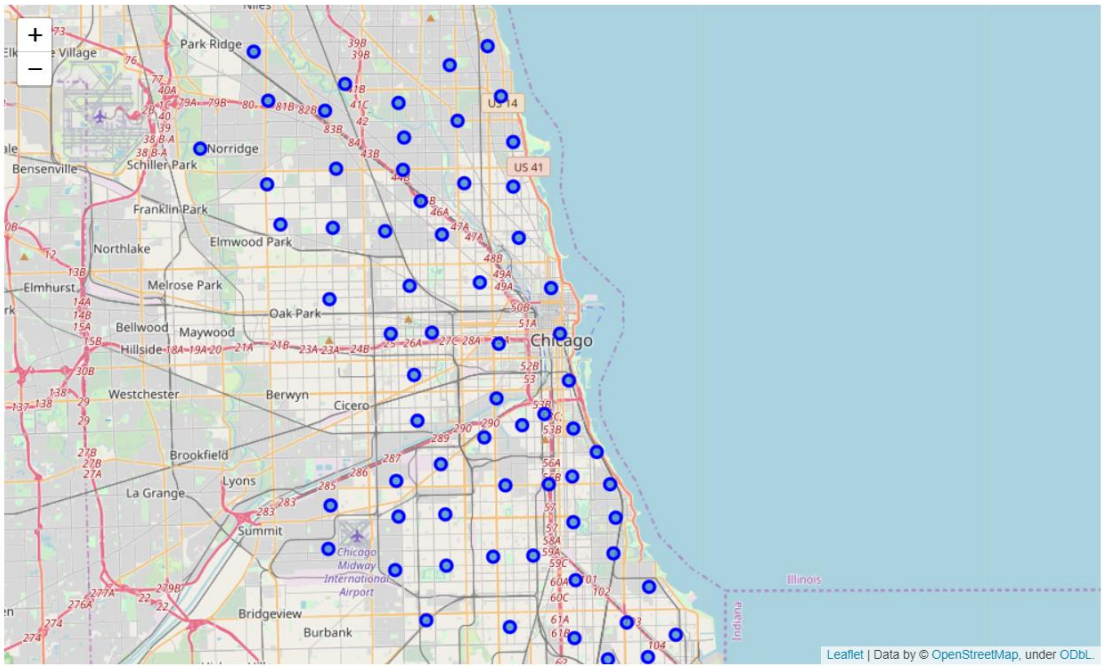
```
# keep chicago and desired features
df_chicago = df[df['CITY']=='Chicago']
df_chicago = df_chicago[['NAME', 'X', 'Y', 'POPDENSITY', 'DIVERSITY', 'PC_INCOME','MEDAGE_CY', 'UNEMPRT_CY', 'MEDVAL_CY']]

# change column names to more familiar ones
df_chicago.columns = ['Neighborhood','Longitude','Latitude','POPDENSITY', 'DIVERSITY', 'PC_INCOME','MEDAGE_CY', 'UNEMPRT_CY', 'MEDVAL_CY']
df_chicago.reset_index(drop=True, inplace=True)
print(df_chicago.shape)
df_chicago.head()
```

```
(77, 9)
```

	Neighborhood	Longitude	Latitude	POPDENSITY	DIVERSITY	PC_INCOME	MEDAGE_CY	UNEMPRT_CY	MEDVAL_CY
0	Chatham	-87.616624	41.7385	12681.364151	7.839623	23599	40.605660	19.645283	126001
1	North Center	-87.684523	41.9473	17814.894231	65.521154	39612	35.211538	10.348077	405797
2	O'hare	-87.847436	41.9633	6591.975000	34.500000	28952	44.060000	9.175000	247836
3	Washington Park	-87.617580	41.7916	13106.814286	11.428571	14888	29.214286	35.517857	194914
4	Garfield Ridge	-87.766976	41.7997	9271.435000	51.655000	22925	40.620000	12.911667	168838

The result of neighborhood and lat-long is visualized with python library folium.



3.3 Explore neighborhoods' restaurants with FourSquare API

We can fetch certain category of venues by calling FourSquare with a URL request specifying center geolocation, radius, number limit on response, and most importantly, the category id. For our purpose to explore taste, I find the unique FourSquare id for food, and put it in a variable: FOODID = '4d4b7105d754a06374d81259'. Of course, the client id and client secret are also required. The formulated url request is shown below:

```
# Now, let's get the top 100 food places that are in Chatham within a radius of 1000 meters.
# First, let's create the GET request URL.
radius = 1000
LIMIT = 100
# the id for food category, this was found in FourSquare website, when a request contains the category, it only fetches venues within
# this specific category
FOODID = '4d4b7105d754a06374d81259'
url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&categoryId={}&limit={}'.format(CL
# send the get request
results = requests.get(url).json()
results
```

A screenshot of part of the result is shown below. We can see that for each food venue, FourSquare assigns it to a subcategory. For the first case, food place named 'Dunkin' is divided into 'Donut Shop' subcategory.

```
'venue': {'id': '4bfd96d74cf820a16eb0ecf4',
 'name': "Dunkin'",
 'location': {'address': '448 E 87th St',
  'crossStreet': 'at Cottage Grove',
  'lat': 41.73674056104464,
  'lng': -87.61256230679408,
  'labeledLatLngs': [{'label': 'display',
    'lat': 41.73674056104464,
    'lng': -87.61256230679408}],
  'distance': 390,
  'postalCode': '60619',
  'cc': 'US',
  'city': 'Chicago',
  'state': 'IL',
  'country': 'United States',
  'formattedAddress': ['448 E 87th St (at Cottage Grove)',
   'Chicago, IL 60619',
   'United States']},
 'categories': [{'id': '4bf58dd8d48988d148941735',
   'name': 'Donut Shop',
   'pluralName': 'Donut Shops',
   'shortName': 'Donuts',
   'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/donuts_',
    'suffix': '.png'},
   'primary': True}],
 'photos': {'count': 0, 'groups': []}},
'referralId': 'e-0-4bfd96d74cf820a16eb0ecf4-0'},
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
```


I write a function to fetch the categories information only and put it into a new DataFrame named 'chicago_venues'. I start off by test it on the first neighborhood, when it works, I write a loop to repeat the process to all neighborhoods. The resulted DataFrame is a large one with 2870 rows and 7 columns.

```
(2870, 7)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Chatham	41.7385	-87.616624	Dunkin'	41.736741	-87.612562	Donut Shop
1	Chatham	41.7385	-87.616624	Garrett Popcorn Shops	41.736535	-87.605829	Snack Place
2	Chatham	41.7385	-87.616624	Mather's More than a Café	41.743548	-87.623089	Café
3	Chatham	41.7385	-87.616624	Kam's Chop Suey	41.743330	-87.623933	Chinese Restaurant
4	Chatham	41.7385	-87.616624	Chipotle Mexican Grill	41.735792	-87.625955	Fast Food Restaurant

Since the project took me several days to finish, it would be annoying to repeat the FourSquare API calls every time I continue my analysis. So I dump the Dataframe into a .pkl file. This way when I come back to the project, I only need to read the .pkl file, which is within one second, instead of waiting for minutes to repeat the API calls.

```
# save data to chicago_venues.pkl, when we continue to conduct analysis, we don't need to repeat calls to FourSquare API again
chicago_venues.to_pickle("chicago_venues.pkl")
```

```
chicago_venues = pd.read_pickle("chicago_venues.pkl")
chicago_venues.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Chatham	41.7385	-87.616624	Dunkin'	41.736741	-87.612562	Donut Shop
1	Chatham	41.7385	-87.616624	Garrett Popcorn Shops	41.736535	-87.605829	Snack Place
2	Chatham	41.7385	-87.616624	Mather's More than a Café	41.743548	-87.623089	Café
3	Chatham	41.7385	-87.616624	Kam's Chop Suey	41.743330	-87.623933	Chinese Restaurant
4	Chatham	41.7385	-87.616624	Chipotle Mexican Grill	41.735792	-87.625955	Fast Food Restaurant

3.4 Feature Engineering

I order to apply machine learning techniques, we have to do some feature engineering process.

Firstly, I convert the food subcategories into dummy variables.

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

# add neighborhood column back to dataframe
chicago_onehot['Neighborhood'] = chicago_venues['Neighborhood']

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

print(chicago_onehot.shape)
chicago_onehot.head()
```

```
(2870, 97)
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	BBQ Joint	Bagel Shop	Bakery	...	Taco Place	Taiwanese Restaurant	Tapas Restaurant	Tex-Mex Restaurant
0	Chatham	0	0	0	0	0	0	0	0	0	...	0	0	0	0
1	Chatham	0	0	0	0	0	0	0	0	0	...	0	0	0	0
2	Chatham	0	0	0	0	0	0	0	0	0	...	0	0	0	0
3	Chatham	0	0	0	0	0	0	0	0	0	...	0	0	0	0
4	Chatham	0	0	0	0	0	0	0	0	0	...	0	0	0	0

5 rows × 97 columns

Secondly, I group venues by neighborhood, and take the mean count of each category in each neighborhood. The resulting DataFrame is stored in ‘chicago_grouped’.

```
# group venues by neighborhood, take the mean of each category
chicago_grouped = chicago_onehot.groupby(["Neighborhood"]).mean().reset_index()
print(chicago_grouped.shape)
chicago_grouped.head()
```

```
(76, 97)
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	BBQ Joint	Bagel Shop	Bakery	...	Taco Place	Taiwanese Restaurant	Tapas Restaurant
0	Albany Park	0.0	0.0	0.031250	0.0	0.0	0.031250	0.031250	0.000000	0.031250	...	0.046875	0.0	0.0
1	Archer Heights	0.0	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.083333	...	0.041667	0.0	0.0
2	Armour Square	0.0	0.0	0.064516	0.0	0.0	0.048387	0.000000	0.016129	0.048387	...	0.000000	0.0	0.0
3	Ashburn	0.0	0.0	0.142857	0.0	0.0	0.000000	0.071429	0.000000	0.000000	...	0.000000	0.0	0.0
4	Auburn Gresham	0.0	0.0	0.095238	0.0	0.0	0.000000	0.047619	0.000000	0.047619	...	0.000000	0.0	0.0

5 rows × 97 columns

3.5 KMeans Clustering to group neighborhoods with food types

I set the number of clusters to be 5. Then import KMeans from sklearn.clusters, use it to fit Chicago_grouped_clustering. Then print out the resulting labels.

```
# set number of clusters
kclusters = 5

chicago_grouped_clustering = chicago_grouped.drop('Neighborhood', 1)

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

# check cluster labels generated
print(kmeans.labels_)
```

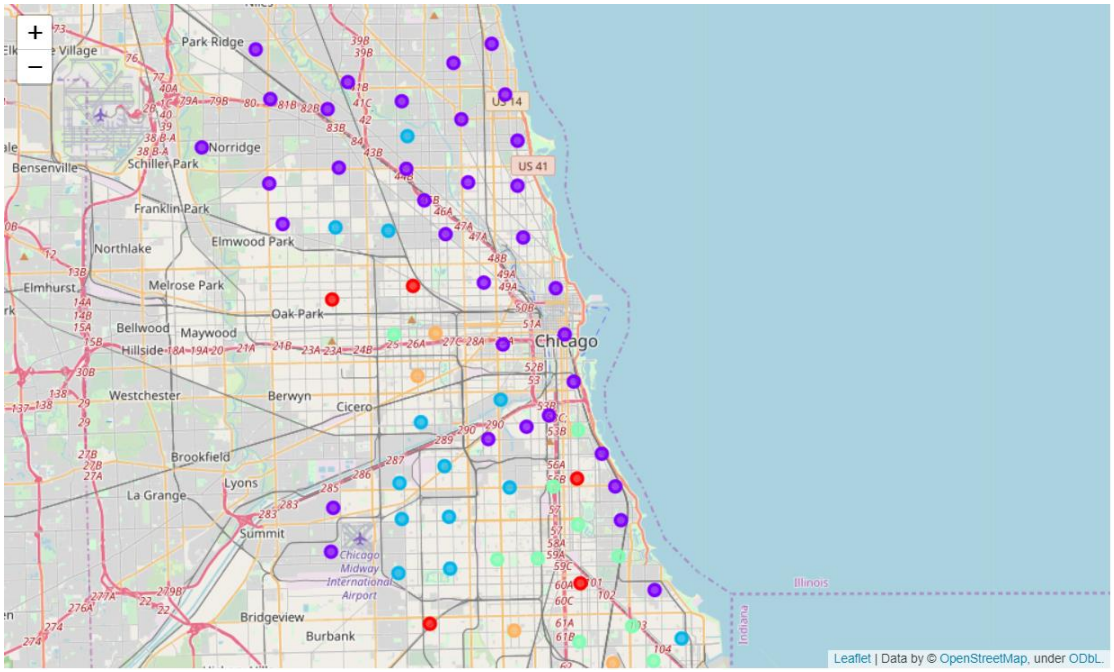
```
[2 2 1 0 4 0 3 1 2 1 1 2 4 3 3 2 1 3 1 4 2 1 1 3 1 3 2 1 0 0 2 2 0 1 1 1 1
 1 1 1 1 1 2 1 1 0 1 1 1 1 2 1 4 1 1 1 1 1 4 4 1 3 2 2 1 1 0 3 2 3 3 2 1 1
 1 3]
```

4 Results & Discussion

4.1 Clusters in geo and demographics

The resulting clusters are shown in the map below. Each cluster is assigned a unique color.

- Cluster 0: red
- Cluster 1: purple
- Cluster 2: blue
- Cluster 3: green
- Cluster 4: orange



I also calculate the average population density, diversity, per capita income, medium age, unemployment rate and medium home value.

```
chicago_merged.groupby('Cluster Labels 1').mean().drop(['Longitude','Latitude'],axis=1)
```

Cluster Labels 1	POPDENSITY	DIVERSITY	PC_INCOME	MEDAGE_CY	UNEMPRT_CY	MEDVAL_CY
0	13635.509258	27.751672	19533.857143	34.905602	23.570329	151878.857143
1	17327.022482	53.640663	30281.918919	36.606176	13.159220	256659.810811
2	17298.096562	73.360604	17349.400000	30.851330	18.338592	154517.000000
3	12611.215708	14.798024	17675.363636	34.328857	27.660768	135440.000000
4	10574.164968	20.430978	16521.000000	32.565829	27.068369	119362.666667

4.2 Clusters with 10 most common food types

Next displays the 10 hottest restaurant types for each neighborhood.

Cluster 0

```
chicago_merged.loc[chicago_merged['Cluster Labels 1'] == 0, chicago_merged.columns[displayCols]]
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
6	Ashburn	American Restaurant	Fast Food Restaurant	Fried Chicken Joint	Wings Joint	Seafood Restaurant	BBQ Joint	Chinese Restaurant	Italian Restaurant	Mexican Restaurant	Pizza Place
16	Morgan Park	BBQ Joint	Fast Food Restaurant	Burger Joint	Pizza Place	American Restaurant	Mexican Restaurant	Diner	Donut Shop	Dumpling Restaurant	Eastern European Restaurant
45	Washington Heights	Sandwich Place	Fried Chicken Joint	American Restaurant	BBQ Joint	Donut Shop	Food	Fast Food Restaurant	Chinese Restaurant	Ethiopian Restaurant	Dim Sum Restaurant
52	Austin	Food	Donut Shop	Sandwich Place	Seafood Restaurant	American Restaurant	BBQ Joint	Breakfast Spot	Chinese Restaurant	Fast Food Restaurant	Fried Chicken Joint
66	Grand Boulevard	Fast Food Restaurant	BBQ Joint	American Restaurant	Fried Chicken Joint	Deli / Bodega	Seafood Restaurant	Wings Joint	Burger Joint	Food	Mexican Restaurant
68	Humboldt Park	Fast Food Restaurant	Food	Wings Joint	Donut Shop	Latin American Restaurant	Seafood Restaurant	American Restaurant	BBQ Joint	Chinese Restaurant	Fried Chicken Joint
76	Grand Crossing	American Restaurant	Fast Food Restaurant	Seafood Restaurant	Fried Chicken Joint	BBQ Joint	Pizza Place	Vegetarian / Vegan Restaurant	Chinese Restaurant	Bakery	Food

Cluster 1 (not all is shown below)

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
1	North Center	Pizza Place	Mexican Restaurant	Sandwich Place	American Restaurant	Fast Food Restaurant	Donut Shop	Sushi Restaurant	Bakery	Mediterranean Restaurant	Latin American Restaurant
2	O'hare	Pizza Place	Chinese Restaurant	Café	Mediterranean Restaurant	Bakery	Dumpling Restaurant	Seafood Restaurant	Italian Restaurant	Sushi Restaurant	Fast Food Restaurant
4	Garfield Ridge	Pizza Place	Hot Dog Joint	American Restaurant	Café	Fast Food Restaurant	Sandwich Place	Breakfast Spot	Chinese Restaurant	Donut Shop	Eastern European Restaurant
5	Beverly	Pizza Place	Sandwich Place	Bakery	Burger Joint	Fried Chicken Joint	Chinese Restaurant	Italian Restaurant	Hot Dog Joint	Donut Shop	Caribbean Restaurant
7	Forest Glen	Fast Food Restaurant	Restaurant	Sandwich Place	Indian Restaurant	Diner	Café	Italian Restaurant	Pizza Place	Asian Restaurant	Thai Restaurant
8	Edison Park	Italian Restaurant	Pizza Place	Mexican Restaurant	Chinese Restaurant	American Restaurant	Bakery	Soup Place	Greek Restaurant	French Restaurant	Food
11	Lincoln Park	American Restaurant	Mexican Restaurant	Pizza Place	Sushi Restaurant	Italian Restaurant	Sandwich Place	Café	Donut Shop	Fried Chicken Joint	Hot Dog Joint
13	Jefferson Park	Pizza Place	Chinese Restaurant	Mexican Restaurant	Greek Restaurant	American Restaurant	Deli / Bodega	Fast Food Restaurant	Seafood Restaurant	Restaurant	Burger Joint
15	Loop	Sandwich Place	Italian Restaurant	Pizza Place	Donut Shop	American Restaurant	Café	Mediterranean Restaurant	Snack Place	Poke Place	Food Truck
17	Near South Side	American Restaurant	Food Court	Fast Food Restaurant	Italian Restaurant	Pizza Place	Café	Deli / Bodega	Sandwich Place	Burger Joint	Restaurant
20	Mount Greenwood	Pizza Place	BBQ Joint	Deli / Bodega	Restaurant	Mexican Restaurant	Italian Restaurant	Taco Place	Breakfast Spot	Food Truck	Empanada Restaurant
22	Uptown	Vietnamese Restaurant	Chinese Restaurant	Pizza Place	Mexican Restaurant	Thai Restaurant	Diner	Asian Restaurant	Sushi Restaurant	Sandwich Place	American Restaurant

Cluster 2

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
14	Hermosa	Mexican Restaurant	Fast Food Restaurant	Pizza Place	Taco Place	Food	Diner	Cuban Restaurant	Chinese Restaurant	Burrito Place	Greek Restaurant
18	South Chicago	Mexican Restaurant	Bakery	Pizza Place	American Restaurant	Food	Italian Restaurant	Southern / Soul Food Restaurant	Burger Joint	Ethiopian Restaurant	Dim Sum Restaurant
25	East Side	Mexican Restaurant	Pizza Place	Chinese Restaurant	Sandwich Place	Fast Food Restaurant	Italian Restaurant	Bakery	Taco Place	BBQ Joint	Food
33	Chicago Lawn	Fast Food Restaurant	Mexican Restaurant	Pizza Place	American Restaurant	Sandwich Place	Fish & Chips Shop	Donut Shop	Cafeteria	Breakfast Spot	Latin American Restaurant
36	Hegewisch	Mexican Restaurant	Food Court	Snack Place	Wings Joint	Deli / Bodega	Dim Sum Restaurant	Diner	Donut Shop	Dumpling Restaurant	Eastern European Restaurant
37	Lower West Side	Mexican Restaurant	Food	Pizza Place	Food Truck	Bakery	Sandwich Place	Gastropub	Breakfast Spot	Donut Shop	Chinese Restaurant
38	Brighton Park	Mexican Restaurant	Seafood Restaurant	Pizza Place	Donut Shop	Sandwich Place	Taco Place	Fast Food Restaurant	Café	Burger Joint	Breakfast Spot
42	Belmont Cragin	Mexican Restaurant	Donut Shop	Sandwich Place	Chinese Restaurant	Fast Food Restaurant	Burger Joint	Food	Diner	Cuban Restaurant	Restaurant
43	New City	Mexican Restaurant	Pizza Place	Chinese Restaurant	Fast Food Restaurant	Food	Sandwich Place	Bakery	Food Truck	Fried Chicken Joint	American Restaurant
46	Albany Park	Mexican Restaurant	Pizza Place	Chinese Restaurant	Korean Restaurant	Sandwich Place	Donut Shop	Fast Food Restaurant	Taco Place	Fried Chicken Joint	Wings Joint
54	Archer Heights	Mexican Restaurant	Pizza Place	Fast Food Restaurant	Food	Seafood Restaurant	Bakery	Chinese Restaurant	Restaurant	Sandwich Place	Donut Shop
57	West Lawn	Mexican Restaurant	Chinese Restaurant	Seafood Restaurant	Fast Food Restaurant	Pizza Place	Hot Dog Joint	Caribbean Restaurant	Diner	Sandwich Place	Donut Shop
58	West Elsdon	Mexican Restaurant	Pizza Place	Fast Food Restaurant	Seafood Restaurant	Sandwich Place	Chinese Restaurant	American Restaurant	Restaurant	Food	Eastern European Restaurant
73	South Lawndale	Mexican Restaurant	Pizza Place	Bakery	Fast Food Restaurant	Restaurant	Food	Chinese Restaurant	Sandwich Place	Donut Shop	Seafood Restaurant
74	Gage Park	Mexican Restaurant	Bakery	Fast Food Restaurant	Taco Place	Sandwich Place	Pizza Place	Asian Restaurant	Food	Fried Chicken Joint	Deli / Bodega

Cluster 3

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Chatham	Fast Food Restaurant	Chinese Restaurant	Sandwich Place	Fried Chicken Joint	Donut Shop	Wings Joint	Breakfast Spot	Food	Pizza Place	Café
3	Washington Park	Fast Food Restaurant	Pizza Place	Fried Chicken Joint	Breakfast Spot	Food Truck	Donut Shop	Deli / Bodega	Food Court	Empanada Restaurant	Dim Sum Restaurant
10	Woodlawn	Fast Food Restaurant	Pizza Place	Chinese Restaurant	Sandwich Place	BBQ Joint	Food	Café	English Restaurant	Dim Sum Restaurant	Diner
12	Englewood	Fast Food Restaurant	Café	Wings Joint	Food	Mexican Restaurant	Donut Shop	Restaurant	Sandwich Place	Seafood Restaurant	Chinese Restaurant
21	Fuller Park	Fast Food Restaurant	Food	Restaurant	Sandwich Place	Pizza Place	Steakhouse	Bakery	Chinese Restaurant	Fried Chicken Joint	American Restaurant
28	West Garfield Park	Fast Food Restaurant	Food	Fried Chicken Joint	Sandwich Place	Taco Place	Café	Middle Eastern Restaurant	Caribbean Restaurant	Pizza Place	Cafeteria
29	Roseland	Fast Food Restaurant	Fried Chicken Joint	Food	Donut Shop	Chinese Restaurant	Sandwich Place	Fish & Chips Shop	Wings Joint	Empanada Restaurant	Deli / Bodega
31	West Englewood	Fast Food Restaurant	American Restaurant	Sandwich Place	Fried Chicken Joint	Food	Pizza Place	Wings Joint	Food Truck	Food Court	German Restaurant
39	Avalon Park	Fast Food Restaurant	Food	Chinese Restaurant	Burger Joint	Fried Chicken Joint	Diner	Fish & Chips Shop	Pizza Place	Restaurant	Caribbean Restaurant
48	Douglas	Fast Food Restaurant	Sandwich Place	Pizza Place	Wings Joint	Snack Place	Fried Chicken Joint	Café	Restaurant	Donut Shop	Southern / Soul Food Restaurant
65	Calumet Heights	Fast Food Restaurant	Fried Chicken Joint	Sandwich Place	Food	Chinese Restaurant	Pizza Place	Mexican Restaurant	Wings Joint	Empanada Restaurant	Dim Sum Restaurant

Cluster 4

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
9	Riverdale	Food	Fast Food Restaurant	Falafel Restaurant	Deli / Bodega	Dim Sum Restaurant	Diner	Donut Shop	Dumpling Restaurant	Eastern European Restaurant	Empanada Restaurant
19	North Lawndale	Food	Fast Food Restaurant	Fried Chicken Joint	Hot Dog Joint	Café	Food Truck	Pizza Place	Seafood Restaurant	Restaurant	Bakery
32	Auburn Gresham	Fast Food Restaurant	Food	American Restaurant	Bakery	Greek Restaurant	Seafood Restaurant	Mexican Restaurant	Southern / Soul Food Restaurant	Chinese Restaurant	Dim Sum Restaurant
40	Burnside	Food	Fast Food Restaurant	Seafood Restaurant	Wings Joint	Southern / Soul Food Restaurant	Deli / Bodega	Caribbean Restaurant	Dim Sum Restaurant	Diner	Donut Shop
51	Pullman	Food	American Restaurant	Fried Chicken Joint	Food Court	Wings Joint	Falafel Restaurant	Dim Sum Restaurant	Diner	Donut Shop	Dumpling Restaurant
71	East Garfield Park	Food	American Restaurant	Diner	Hot Dog Joint	Bakery	Café	Seafood Restaurant	Pizza Place	Southern / Soul Food Restaurant	Burger Joint

4.3 Discussion

From the map we can see that purple dots (Cluster 1) are spread out along the coast line and major highway in the north side. The demographic statistics shows that they are the richest. These neighborhoods have the highest population density, highest per capita income, highest medium age, lowest unemployment rate and highest medium home value. As for their taste, they are very diversified with pizza places, Chinese, American and Japanese cuisines among the most popular. Fast food seems not in their consideration at all.

The second most common dots are blue (Cluster 2). They mainly lie in southwest and northwest suburbs. Census shows that these neighborhoods have the highest population diversity, youngest residents, second to highest per capita income, second to lowest unemployment rate and second to highest medium home value. As for their taste, Mexican food is undoubtedly the most popular (ranked 1st for all but one). Besides, pizza, Chinese

and taco places are among the most popular. Their passion to fast food seems moderate (ranked about 5th place). I would guess these are neighborhoods where Mexican community mainly resides.

Green dots (Cluster 3) mainly lie in the southern part of Chicago suburb. These neighborhoods have the lowest population diversity, and highest unemployment rate. As for their taste, fast food is definitely the No.1 choice. Besides, cafe, sandwich, donut and fried chicken (these are relatively cheaper among all type of food) all seem very popular.

The other two colors appear less common in the map.

Red dots (Cluster 0) are sparsely spread out in southern and western suburbs. These neighborhoods have medium demographic statistic numbers. As for their taste, their favorite is fast food such as fried chicken, burger and donut.

Orange dots (Cluster 4) are in the southern most areas and west suburbs. These neighborhoods have the lowest population density, lowest per capita income, and lowest home value. As for their taste, they don't seem to spare the time differentiate between different types of food, so "food" and fast food are among the most popular. Other fast and cheap options are also under their considerations like hot dog, deli, donut and cafe.

4.4 Recommendations

Based on the results, my recommendations for potential food vendors really depend on the type of food and pricings they are interested in.

If he or she is to open a pricy fine dining restaurants, no matter which cuisine it is, I definitely recommend to pick a location along the coast line of northern highway (Cluster 1). These are the places where people are willing to and capable of paying a good price for food.

If the restaurant is Mexican, he should find a place in one of Cluster 2. Since community there have a real passion for Mexican food. Fast food with medium to high price may also be considered as they have the 2nd highest per capita income.

If he or she just wants to open a cheap fast food place, neighborhoods in southern and western suburbs (Cluster 0 and Cluster 3) should be taken into consideration. These two clusters have the lowest population diversity amongst all. Fast food to consider should be like cafe, sandwich, donut and fried chicken.

Neighborhoods in Cluster 4 may not be suitable for opening new restaurants since they have the lowest population density.

5 Conclusion

In this project I try to answer the question: where should a potential food vendor pick his business location and why? I apply k-Means clustering algorithm to a multi-dimensional dataset with food types aggregated based on neighborhoods, calculate some demographic statistics and visualize the clustering result. The neighborhoods of Chicago are segmented into 5 clusters. Based on analysis, I give corresponding recommendations for high-end fine dining, Mexican restaurants and cheap fast food respectively, some other neighborhoods should be avoided to start a new restaurant. Besides, I find that people's preference for food is highly correlated with their economic status.

Data

The results of this project can be improved by fetching the price information for each food venue in FourSquare API, which would require a premium account to do so. This way we can better divide the subcategory of food, as well as assess the correlation between social economic status and pricings/tastes of food.

I also applied a simplified approach in fetching food venues. That is, I simply picked the centroid of each neighborhood, and fetch top 100 venues within 1 km of the centroid. This is not very precise as it may miss some popular venues out of the 1 km radius. It can be improved with a precise boundary data of Chicago neighborhood.

Algorithm

The decision of number of clusters is kind of arbitrary. If with more time, I would try to apply elbow method to run KMeans algorithm on different choices of cluster numbers, and then pick the most suitable one.