The Battle of Neighborhoods (Week 2)

Table of Contents

1:- Introduction : Business Problem

Problem Statement: Finding a good Neighborhood for opening an Indian Restaurant in New Delhi, IN

This study aims at helping people planning on opening an Indian Restaurant in the New Delhi,IN area by choosing the right location by providing data about Neighborhoods and nearby competitors. New Delhi is an urban district located in the city of Delhi which serves as the capital of India. New Delhi is the largest commercial city in northern India.

Target Audience:

1- Working From Home Population ordering meal-ready meals looking for a nearby meal-prepping restaurant.

2- Freelancers looking into opening a restaurant. This will provide an analysis whether the venture is feasible or not.

Importing Libraries

In [1]:
```
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

#!conda install -c conda-forge geopy --yes # 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.pyplot as plt # plotting library
%matplotlib inline
import matplotlib.cm as cm
import matplotlib.colors as colors

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

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): - Collecting package metadata (current_repodata.json)\ done
Solving environment: done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json)/ done
Solving environment: done
Solving environment| done
done

## Package Plan ##

  environment location: /home/spark/shared/conda/envs/python

  added / updated specs:
    - folium=0.5.0

The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    altair-4.2.0               |     pyhd8ed1ab_1         711 KB  conda-forge
    attrs-21.4.0               |     pyhd8ed1ab_0          49 KB  conda-forge
    branca-0.5.0               |     pyhd8ed1ab_0          26 KB  conda-forge
    brotlipy-0.7.0             |py39hb9d737c_1004         342 KB  conda-forge
    ca-certificates-2022.6.15  |       ha878542_0         149 KB  conda-forge
    certifi-2022.6.15          |   py39hf3d152e_0         155 KB  conda-forge
    cffi-1.14.6                |   py39he32792d_0         227 KB  conda-forge
    charset-normalizer-2.0.12  |     pyhd8ed1ab_0          35 KB  conda-forge
    cryptography-37.0.2        |   py39hd97740a_0         1.5 MB  conda-forge
    entrypoints-0.4            |     pyhd8ed1ab_0           9 KB  conda-forge
    folium-0.5.0               |            py_0          45 KB  conda-forge
    idna-3.3                   |     pyhd8ed1ab_0          55 KB  conda-forge
    importlib-metadata-4.11.4  |   py39hf3d152e_0          33 KB  conda-forge
    importlib_resources-5.8.0  |     pyhd8ed1ab_0          22 KB  conda-forge
    jinja2-3.1.2               |     pyhd8ed1ab_1          99 KB  conda-forge
    jsonschema-4.6.0           |     pyhd8ed1ab_0          62 KB  conda-forge
    libblas-3.9.0              |15_linux64_openblas          12 KB  conda-forge
    libcblas-3.9.0             |15_linux64_openblas          12 KB  conda-forge
    libgfortran-ng-12.1.0      |      h69a702a_16          23 KB  conda-forge
    libgfortran5-12.1.0        |      hdcd56e2_16         1.8 MB  conda-forge
    liblapack-3.9.0            |15_linux64_openblas          12 KB  conda-forge
    libopenblas-0.3.20         |pthreads_h78a6416_0        10.1 MB  conda-forge
    markupsafe-2.1.1           |   py39hb9d737c_1          22 KB  conda-forge
    numpy-1.22.3               |   py39hc58783e_2         6.8 MB  conda-forge
```

```
  openssl-1.1.1o             |       h166bdaf_0         2.1 MB  conda-forge
  pandas-1.4.2               |   py39h1832856_1        12.5 MB  conda-forge
  pycparser-2.21             |     pyhd8ed1ab_0         100 KB  conda-forge
  pyopenssl-22.0.0           |     pyhd8ed1ab_0          49 KB  conda-forge
  pyrsistent-0.18.1          |   py39hb9d737c_1          92 KB  conda-forge
  pysocks-1.7.1              |   py39hf3d152e_5          28 KB  conda-forge
  python-dateutil-2.8.2      |     pyhd8ed1ab_0         240 KB  conda-forge
  python_abi-3.9             |           2_cp39           4 KB  conda-forge
  pytz-2022.1                |     pyhd8ed1ab_0         242 KB  conda-forge
  requests-2.28.0            |     pyhd8ed1ab_0          52 KB  conda-forge
  six-1.16.0                 |     pyh6c4a22f_0          14 KB  conda-forge
  toolz-0.11.2               |     pyhd8ed1ab_0          48 KB  conda-forge
  urllib3-1.26.9             |     pyhd8ed1ab_0         100 KB  conda-forge
  vincent-0.4.4              |             py_1          28 KB  conda-forge
  zipp-3.8.0                 |     pyhd8ed1ab_0          12 KB  conda-forge
  ------------------------------------------------------------
                                       Total:        37.8 MB

The following NEW packages will be INSTALLED:

  altair             conda-forge/noarch::altair-4.2.0-pyhd8ed1ab_1
  attrs              conda-forge/noarch::attrs-21.4.0-pyhd8ed1ab_0
  branca             conda-forge/noarch::branca-0.5.0-pyhd8ed1ab_0
  brotlipy           conda-forge/linux-64::brotlipy-0.7.0-py39hb9d737c_1004
  cffi               conda-forge/linux-64::cffi-1.14.6-py39he32792d_0
  charset-normalizer conda-forge/noarch::charset-normalizer-2.0.12-pyhd8ed1ab_0
  cryptography       conda-forge/linux-64::cryptography-37.0.2-py39hd97740a_0
  entrypoints        conda-forge/noarch::entrypoints-0.4-pyhd8ed1ab_0
  folium             conda-forge/noarch::folium-0.5.0-py_0
  idna               conda-forge/noarch::idna-3.3-pyhd8ed1ab_0
  importlib-metadata conda-forge/linux-64::importlib-metadata-4.11.4-py39hf3d152e_0
  importlib_resourc~ conda-forge/noarch::importlib_resources-5.8.0-pyhd8ed1ab_0
  jinja2             conda-forge/noarch::jinja2-3.1.2-pyhd8ed1ab_1
  jsonschema         conda-forge/noarch::jsonschema-4.6.0-pyhd8ed1ab_0
  libblas            conda-forge/linux-64::libblas-3.9.0-15_linux64_openblas
  libcblas           conda-forge/linux-64::libcblas-3.9.0-15_linux64_openblas
  libgfortran-ng     conda-forge/linux-64::libgfortran-ng-12.1.0-h69a702a_16
  libgfortran5       conda-forge/linux-64::libgfortran5-12.1.0-hdcd56e2_16
  liblapack          conda-forge/linux-64::liblapack-3.9.0-15_linux64_openblas
  libopenblas        conda-forge/linux-64::libopenblas-0.3.20-pthreads_h78a6416_0
  markupsafe         conda-forge/linux-64::markupsafe-2.1.1-py39hb9d737c_1
  numpy              conda-forge/linux-64::numpy-1.22.3-py39hc58783e_2
  pandas             conda-forge/linux-64::pandas-1.4.2-py39h1832856_1
  pycparser          conda-forge/noarch::pycparser-2.21-pyhd8ed1ab_0
  pyopenssl          conda-forge/noarch::pyopenssl-22.0.0-pyhd8ed1ab_0
  pyrsistent         conda-forge/linux-64::pyrsistent-0.18.1-py39hb9d737c_1
  pysocks            conda-forge/linux-64::pysocks-1.7.1-py39hf3d152e_5
  python-dateutil    conda-forge/noarch::python-dateutil-2.8.2-pyhd8ed1ab_0
  python_abi         conda-forge/linux-64::python_abi-3.9-2_cp39
  pytz               conda-forge/noarch::pytz-2022.1-pyhd8ed1ab_0
  requests           conda-forge/noarch::requests-2.28.0-pyhd8ed1ab_0
  six                conda-forge/noarch::six-1.16.0-pyh6c4a22f_0
  toolz              conda-forge/noarch::toolz-0.11.2-pyhd8ed1ab_0
  urllib3            conda-forge/noarch::urllib3-1.26.9-pyhd8ed1ab_0
  vincent            conda-forge/noarch::vincent-0.4.4-py_1
  zipp               conda-forge/noarch::zipp-3.8.0-pyhd8ed1ab_0

The following packages will be UPDATED:
```

```
  ca-certificates    pkgs/main::ca-certificates-2022.4.26-~ --> conda-forge::ca-certificates-2022.6.15-ha878542_0
  certifi            pkgs/main::certifi-2022.5.18.1-py39h0~ --> conda-forge::certifi-2022.6.15-py39hf3d152e_0

The following packages will be SUPERSEDED by a higher-priority channel:

  openssl              pkgs/main::openssl-1.1.1o-h7f8727e_0 --> conda-forge::openssl-1.1.1o-h166bdaf_0



Downloading and Extracting Packages
cryptography-37.0.2  | 1.5 MB    |                                       |   0%
## Package Plan ##

  environment location: /home/spark/shared/conda/envs/python

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    altair-4.2.0               |     pyhd8ed1ab_1         711 KB  conda-forge
    attrs-21.4.0               |     pyhd8ed1ab_0          49 KB  conda-forge
    branca-0.5.0               |     pyhd8ed1ab_0          26 KB  conda-forge
    brotlipy-0.7.0             |py39hb9d737c_1004         342 KB  conda-forge
    ca-certificates-2022.6.15  |       ha878542_0         149 KB  conda-forge
    certifi-2022.6.15          |   py39hf3d152e_0         155 KB  conda-forge
    cffi-1.14.6                |   py39he32792d_0         227 KB  conda-forge
    charset-normalizer-2.0.12  |     pyhd8ed1ab_0          35 KB  conda-forge
    cryptography-37.0.2        |   py39hd97740a_0         1.5 MB  conda-forge
    entrypoints-0.4            |     pyhd8ed1ab_0           9 KB  conda-forge
    folium-0.5.0               |            py_0          45 KB  conda-forge
    idna-3.3                   |     pyhd8ed1ab_0          55 KB  conda-forge
    importlib-metadata-4.11.4  |   py39hf3d152e_0          33 KB  conda-forge
    importlib_resources-5.8.0  |     pyhd8ed1ab_0          22 KB  conda-forge
    jinja2-3.1.2               |     pyhd8ed1ab_1          99 KB  conda-forge
    jsonschema-4.6.0           |     pyhd8ed1ab_0          62 KB  conda-forge
    libblas-3.9.0              |15_linux64_openblas          12 KB  conda-forge
    libcblas-3.9.0             |15_linux64_openblas          12 KB  conda-forge
    libgfortran-ng-12.1.0      |      h69a702a_16          23 KB  conda-forge
    libgfortran5-12.1.0        |      hdcd56e2_16         1.8 MB  conda-forge
    liblapack-3.9.0            |15_linux64_openblas          12 KB  conda-forge
    libopenblas-0.3.20         |pthreads_h78a6416_0        10.1 MB  conda-forge
    markupsafe-2.1.1           |   py39hb9d737c_1          22 KB  conda-forge
    numpy-1.22.3               |   py39hc58783e_2         6.8 MB  conda-forge
    openssl-1.1.1o             |       h166bdaf_0         2.1 MB  conda-forge
    pandas-1.4.2               |   py39h1832856_1        12.5 MB  conda-forge
    pycparser-2.21             |     pyhd8ed1ab_0         100 KB  conda-forge
    pyopenssl-22.0.0           |     pyhd8ed1ab_0          49 KB  conda-forge
    pyrsistent-0.18.1          |   py39hb9d737c_1          92 KB  conda-forge
    pysocks-1.7.1              |   py39hf3d152e_5          28 KB  conda-forge
    python-dateutil-2.8.2      |     pyhd8ed1ab_0         240 KB  conda-forge
    python_abi-3.9             |           2_cp39           4 KB  conda-forge
    pytz-2022.1                |     pyhd8ed1ab_0         242 KB  conda-forge
    requests-2.28.0            |     pyhd8ed1ab_0          52 KB  conda-forge
    six-1.16.0                 |     pyh6c4a22f_0          14 KB  conda-forge
    toolz-0.11.2               |     pyhd8ed1ab_0          48 KB  conda-forge
```

```
  urllib3-1.26.9             |     pyhd8ed1ab_0         100 KB  conda-forge
  vincent-0.4.4              |             py_1          28 KB  conda-forge
  zipp-3.8.0                 |     pyhd8ed1ab_0          12 KB  conda-forge
  ------------------------------------------------------------
                                       Total:        37.8 MB

The following NEW packages will be INSTALLED:

  altair             conda-forge/noarch::altair-4.2.0-pyhd8ed1ab_1
  attrs              conda-forge/noarch::attrs-21.4.0-pyhd8ed1ab_0
  branca             conda-forge/noarch::branca-0.5.0-pyhd8ed1ab_0
  brotlipy           conda-forge/linux-64::brotlipy-0.7.0-py39hb9d737c_1004
  cffi               conda-forge/linux-64::cffi-1.14.6-py39he32792d_0
  charset-normalizer conda-forge/noarch::charset-normalizer-2.0.12-pyhd8ed1ab_0
  cryptography       conda-forge/linux-64::cryptography-37.0.2-py39hd97740a_0
  entrypoints        conda-forge/noarch::entrypoints-0.4-pyhd8ed1ab_0
  folium             conda-forge/noarch::folium-0.5.0-py_0
  idna               conda-forge/noarch::idna-3.3-pyhd8ed1ab_0
  importlib-metadata conda-forge/linux-64::importlib-metadata-4.11.4-py39hf3d152e_0
  importlib_resourc~ conda-forge/noarch::importlib_resources-5.8.0-pyhd8ed1ab_0
  jinja2             conda-forge/noarch::jinja2-3.1.2-pyhd8ed1ab_1
  jsonschema         conda-forge/noarch::jsonschema-4.6.0-pyhd8ed1ab_0
  libblas            conda-forge/linux-64::libblas-3.9.0-15_linux64_openblas
  libcblas           conda-forge/linux-64::libcblas-3.9.0-15_linux64_openblas
  libgfortran-ng     conda-forge/linux-64::libgfortran-ng-12.1.0-h69a702a_16
  libgfortran5       conda-forge/linux-64::libgfortran5-12.1.0-hdcd56e2_16
  liblapack          conda-forge/linux-64::liblapack-3.9.0-15_linux64_openblas
  libopenblas        conda-forge/linux-64::libopenblas-0.3.20-pthreads_h78a6416_0
  markupsafe         conda-forge/linux-64::markupsafe-2.1.1-py39hb9d737c_1
  numpy              conda-forge/linux-64::numpy-1.22.3-py39hc58783e_2
  pandas             conda-forge/linux-64::pandas-1.4.2-py39h1832856_1
  pycparser          conda-forge/noarch::pycparser-2.21-pyhd8ed1ab_0
  pyopenssl          conda-forge/noarch::pyopenssl-22.0.0-pyhd8ed1ab_0
  pyrsistent         conda-forge/linux-64::pyrsistent-0.18.1-py39hb9d737c_1
  pysocks            conda-forge/linux-64::pysocks-1.7.1-py39hf3d152e_5
  python-dateutil    conda-forge/noarch::python-dateutil-2.8.2-pyhd8ed1ab_0
  python_abi         conda-forge/linux-64::python_abi-3.9-2_cp39
  pytz               conda-forge/noarch::pytz-2022.1-pyhd8ed1ab_0
  requests           conda-forge/noarch::requests-2.28.0-pyhd8ed1ab_0
  six                conda-forge/noarch::six-1.16.0-pyh6c4a22f_0
  toolz              conda-forge/noarch::toolz-0.11.2-pyhd8ed1ab_0
  urllib3            conda-forge/noarch::urllib3-1.26.9-pyhd8ed1ab_0
  vincent            conda-forge/noarch::vincent-0.4.4-py_1
  zipp               conda-forge/noarch::zipp-3.8.0-pyhd8ed1ab_0

The following packages will be UPDATED:

  ca-certificates    pkgs/main::ca-certificates-2022.4.26-~ --> conda-forge::ca-certificates-2022.6.15-ha878542_0
  certifi            pkgs/main::certifi-2022.5.18.1-py39h0~ --> conda-forge::certifi-2022.6.15-py39hf3d152e_0

The following packages will be SUPERSEDED by a higher-priority channel:

  openssl              pkgs/main::openssl-1.1.1o-h7f8727e_0 --> conda-forge::openssl-1.1.1o-h166bdaf_0



Downloading and Extracting Packages
cryptography-37.0.2  | 1.5 MB    | ##################################### | 100%
cryptography-37.0.2  | 1.5 MB    | ##################################### | 100%
```

```
pyrsistent-0.18.1    | 92 KB     | ##################################### | 100%
pyrsistent-0.18.1    | 92 KB     | ##################################### | 100%
libgfortran-ng-12.1. | 23 KB     | ##################################### | 100%
libgfortran-ng-12.1. | 23 KB     | ##################################### | 100%
markupsafe-2.1.1     | 22 KB     | ##################################### | 100%
markupsafe-2.1.1     | 22 KB     | ##################################### | 100%
liblapack-3.9.0      | 12 KB     | ##################################### | 100%
liblapack-3.9.0      | 12 KB     | ##################################### | 100%
certifi-2022.6.15    | 155 KB    | ##################################### | 100%
certifi-2022.6.15    | 155 KB    | ##################################### | 100%
attrs-21.4.0         | 49 KB     | ##################################### | 100%
attrs-21.4.0         | 49 KB     | ##################################### | 100%
jsonschema-4.6.0     | 62 KB     | ##################################### | 100%
jsonschema-4.6.0     | 62 KB     | ##################################### | 100%
idna-3.3             | 55 KB     | ##################################### | 100%
idna-3.3             | 55 KB     | ##################################### | 100%
brotlipy-0.7.0       | 342 KB    | ##################################### | 100%
brotlipy-0.7.0       | 342 KB    | ##################################### | 100%
pysocks-1.7.1        | 28 KB     | ##################################### | 100%
pysocks-1.7.1        | 28 KB     | ##################################### | 100%
ca-certificates-2022 | 149 KB    | ##################################### | 100%
ca-certificates-2022 | 149 KB    | ##################################### | 100%
pytz-2022.1          | 242 KB    | ##################################### | 100%
pytz-2022.1          | 242 KB    | ##################################### | 100%
libblas-3.9.0        | 12 KB     | ##################################### | 100%
libblas-3.9.0        | 12 KB     | ##################################### | 100%
cffi-1.14.6          | 227 KB    | ##################################### | 100%
cffi-1.14.6          | 227 KB    | ##################################### | 100%
entrypoints-0.4      | 9 KB      | ##################################### | 100%
entrypoints-0.4      | 9 KB      | ##################################### | 100%
numpy-1.22.3         | 6.8 MB    | ##################################### | 100%
numpy-1.22.3         | 6.8 MB    | ##################################### | 100%
pyopenssl-22.0.0     | 49 KB     | ##################################### | 100%
pyopenssl-22.0.0     | 49 KB     | ##################################### | 100%
altair-4.2.0         | 711 KB    | ##################################### | 100%
altair-4.2.0         | 711 KB    | ##################################### | 100%
openssl-1.1.1o       | 2.1 MB    | ##################################### | 100%
openssl-1.1.1o       | 2.1 MB    | ##################################### | 100%
importlib_resources- | 22 KB     | ##################################### | 100%
importlib_resources- | 22 KB     | ##################################### | 100%
python-dateutil-2.8. | 240 KB    | ##################################### | 100%
python-dateutil-2.8. | 240 KB    | ##################################### | 100%
requests-2.28.0      | 52 KB     | ##################################### | 100%
requests-2.28.0      | 52 KB     | ##################################### | 100%
folium-0.5.0         | 45 KB     | ##################################### | 100%
folium-0.5.0         | 45 KB     | ##################################### | 100%
toolz-0.11.2         | 48 KB     | ##################################### | 100%
toolz-0.11.2         | 48 KB     | ##################################### | 100%
urllib3-1.26.9       | 100 KB    | ##################################### | 100%
urllib3-1.26.9       | 100 KB    | ##################################### | 100%
pycparser-2.21       | 100 KB    | ##################################### | 100%
pycparser-2.21       | 100 KB    | ##################################### | 100%
pandas-1.4.2         | 12.5 MB   | ##################################### | 100%
pandas-1.4.2         | 12.5 MB   | ##################################### | 100%
branca-0.5.0         | 26 KB     | ##################################### | 100%
branca-0.5.0         | 26 KB     | ##################################### | 100%
jinja2-3.1.2         | 99 KB     | ##################################### | 100%
jinja2-3.1.2         | 99 KB     | ##################################### | 100%
zipp-3.8.0           | 12 KB     | ##################################### | 100%
```

```
zipp-3.8.0            | 12 KB     | ##################################### | 100%
libgfortran5-12.1.0   | 1.8 MB    | ##################################### | 100%
libgfortran5-12.1.0   | 1.8 MB    | ##################################### | 100%
importlib-metadata-4  | 33 KB     | ##################################### | 100%
importlib-metadata-4  | 33 KB     | ##################################### | 100%
libopenblas-0.3.20    | 10.1 MB   | ##################################### | 100%
libopenblas-0.3.20    | 10.1 MB   | ##################################### | 100%
charset-normalizer-2  | 35 KB     | ##################################### | 100%
charset-normalizer-2  | 35 KB     | ##################################### | 100%
six-1.16.0            | 14 KB     | ##################################### | 100%
six-1.16.0            | 14 KB     | ##################################### | 100%
vincent-0.4.4         | 28 KB     | ##################################### | 100%
vincent-0.4.4         | 28 KB     | ##################################### | 100%
python_abi-3.9        | 4 KB      | ##################################### | 100%
python_abi-3.9        | 4 KB      | ##################################### | 100%
libcblas-3.9.0        | 12 KB     | ##################################### | 100%
libcblas-3.9.0        | 12 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Verifying transaction: done
Executing transaction: done
Executing transaction: done
done
Libraries imported.
Libraries imported.
```

2:- Explore Dataset

Delhi has a total of 9 boroughs and 163 neighborhoods. In order to analyse the neighborhoods and explore them, we will essentially need a dataset that contains the 9 boroughs and the neighborhoods that exist in each borough as well as the the latitude and logitude coordinates of each neighborhood.

Luckily, this dataset exists for free on the Kaggle's website, here is the link to the dataset: https://www.kaggle.com/shaswatd673/delhi-neighborhood-data#delhi_dataSet.csv

In [2]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.

if os.environ.get('RUNTIME_ENV_LOCATION_TYPE') == 'external':
    endpoint_eca9f9996044428689017d0845ce0971 = 'https://s3.eu.cloud-object-storage.appdomain.cloud'
else:
    endpoint_eca9f9996044428689017d0845ce0971 = 'https://s3.private.eu.cloud-object-storage.appdomain.cloud'

client_eca9f9996044428689017d0845ce0971 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='RIBm4b8rRmBXPfi299BHd-cr842HgPGgs72FcuR-qrkv',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url=endpoint_eca9f9996044428689017d0845ce0971)
```

```
body = client_eca9f9996044428689017d0845ce0971.get_object(Bucket='analysinghousepriceinbengaluru-donotdelete-pr-kyhnquzbpe6qfb',Key='delhi_dataSet.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

# If you are reading an Excel file into a pandas DataFrame, replace `read_csv` by `read_excel` in the next statement.
df_delhi = pd.read_csv(body)
df_delhi.head()
```

Out[2]:

	Unnamed: 0	Borough	Neighborhood	latitude	longitude
0	0	North West Delhi	Adarsh Nagar	28.614193	77.071541
1	1	North West Delhi	Ashok Vihar	28.699453	77.184826
2	2	North West Delhi	Azadpur	28.707657	77.175547
3	3	North West Delhi	Bawana	28.799660	77.032885
4	4	North West Delhi	Begum Pur	NaN	NaN

Out[2]:

	Unnamed: 0	Borough	Neighborhood	latitude	longitude
0	0	North West Delhi	Adarsh Nagar	28.614193	77.071541
1	1	North West Delhi	Ashok Vihar	28.699453	77.184826
2	2	North West Delhi	Azadpur	28.707657	77.175547
3	3	North West Delhi	Bawana	28.799660	77.032885
4	4	North West Delhi	Begum Pur	NaN	NaN

In [3]:
```
df_delhi.drop('Unnamed: 0', axis = 1, inplace= True)
```

In [4]:
```
df_delhi.head()
```

Out[4]:

	Borough	Neighborhood	latitude	longitude
0	North West Delhi	Adarsh Nagar	28.614193	77.071541
1	North West Delhi	Ashok Vihar	28.699453	77.184826
2	North West Delhi	Azadpur	28.707657	77.175547
3	North West Delhi	Bawana	28.799660	77.032885
4	North West Delhi	Begum Pur	NaN	NaN

Out[4]:

	Borough	Neighborhood	latitude	longitude
0	North West Delhi	Adarsh Nagar	28.614193	77.071541
1	North West Delhi	Ashok Vihar	28.699453	77.184826

	Borough	Neighborhood	latitude	longitude
2	North West Delhi	Azadpur	28.707657	77.175547
3	North West Delhi	Bawana	28.799660	77.032885
4	North West Delhi	Begum Pur	NaN	NaN

In [5]:
```
df_delhi=df_delhi.dropna()
df_delhi.head(6)
```

Out[5]:

	Borough	Neighborhood	latitude	longitude
0	North West Delhi	Adarsh Nagar	28.614193	77.071541
1	North West Delhi	Ashok Vihar	28.699453	77.184826
2	North West Delhi	Azadpur	28.707657	77.175547
3	North West Delhi	Bawana	28.799660	77.032885
5	North West Delhi	Dhaka	39.031714	-90.261223
6	North West Delhi	Jahangirpuri	28.725972	77.162658

Out[5]:

	Borough	Neighborhood	latitude	longitude
0	North West Delhi	Adarsh Nagar	28.614193	77.071541
1	North West Delhi	Ashok Vihar	28.699453	77.184826
2	North West Delhi	Azadpur	28.707657	77.175547
3	North West Delhi	Bawana	28.799660	77.032885
5	North West Delhi	Dhaka	39.031714	-90.261223
6	North West Delhi	Jahangirpuri	28.725972	77.162658

Print the boroughs and neighborhoods for dataframe

In [6]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(df_delhi['Borough'].unique()),
        df_delhi.shape[0]
    )
)
```

```
The dataframe has 9 boroughs and 163 neighborhoods.
```
The dataframe has 9 boroughs and 163 neighborhoods.

Use geopy library to get the latitude and longitude values of Delhi

In [ ]:
```
address = 'Delhi, IN'
```

```
geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Delhi are {}, {}.'.format(latitude, longitude))
```

Create map of Delhi using latitude and longitude values

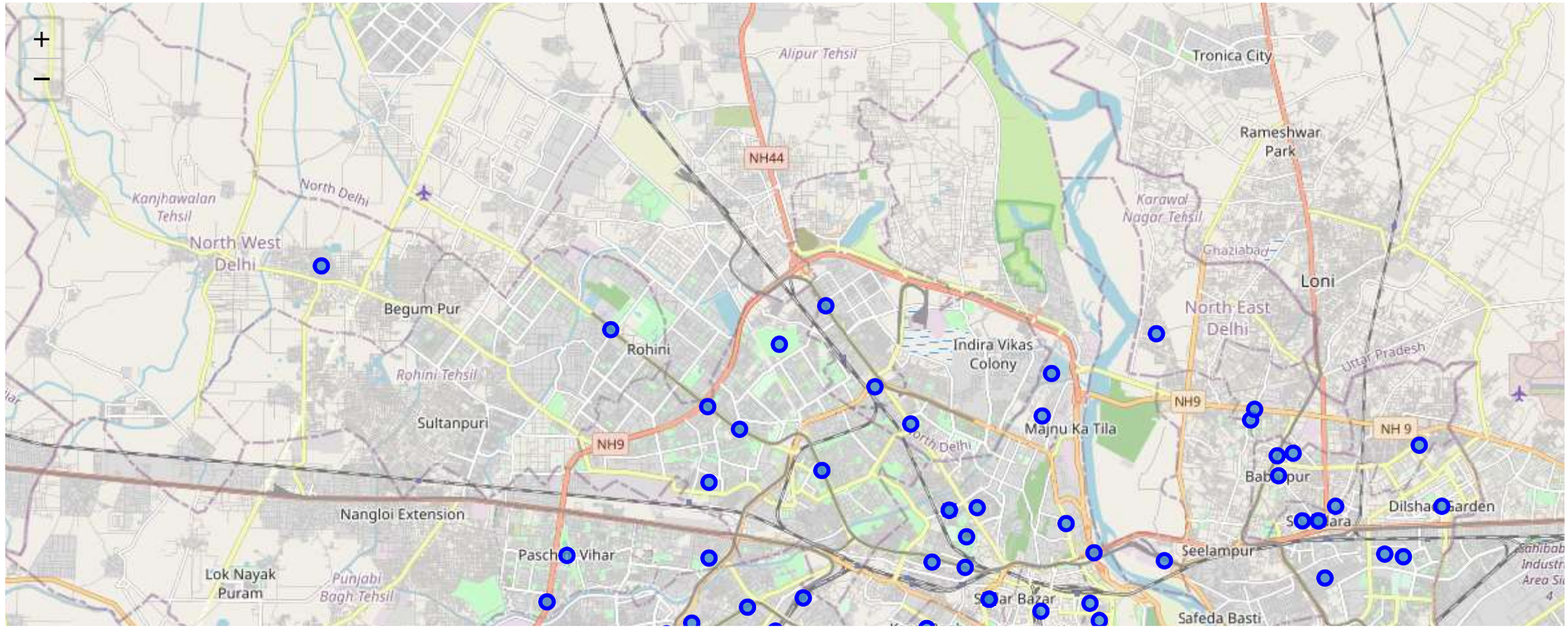
In [8]:

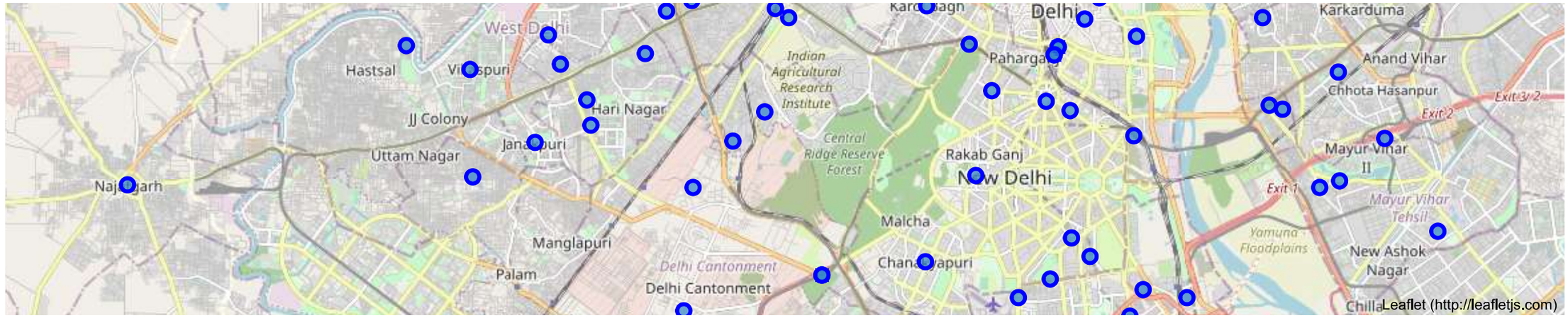
```
map_delhi= folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(df_delhi['latitude'], df_delhi['longitude'], df_delhi['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_delhi)

map_delhi
```

Out[8]:





3:-Methodology

In this section, we will conduct exploratory data analysis. We have New Delhi area to explore neighborhood and check the competition for Indian Restaurant. We will get information on common venes in New Delhi with the help of Foursquare API. We will also be using K-means clustering.

let's simplify the above map and segment and cluster only the neighborhoods in New Delhi. So let's slice the original dataframe and create a new dataframe of the New Delhi data.

In [9]:
```
ND_data =df_delhi[df_delhi['Borough'] == 'New Delhi'].reset_index(drop=True)
ND_data.head()
```

Out[9]:

	Borough	Neighborhood	latitude	longitude
0	New Delhi	Barakhamba Road	28.629142	77.226149
1	New Delhi	Chanakyapuri	28.594677	77.188521
2	New Delhi	Connaught Place	28.631383	77.219792
3	New Delhi	Gautampuri	28.511570	77.302623
4	New Delhi	Gole Market	28.633719	77.205627

Let's get the geographical coordinates of New Delhi.

In [10]:
```
address = 'New Delhi, IN'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of North East Delhi are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of North East Delhi are 28.6138954, 77.2090057.
```

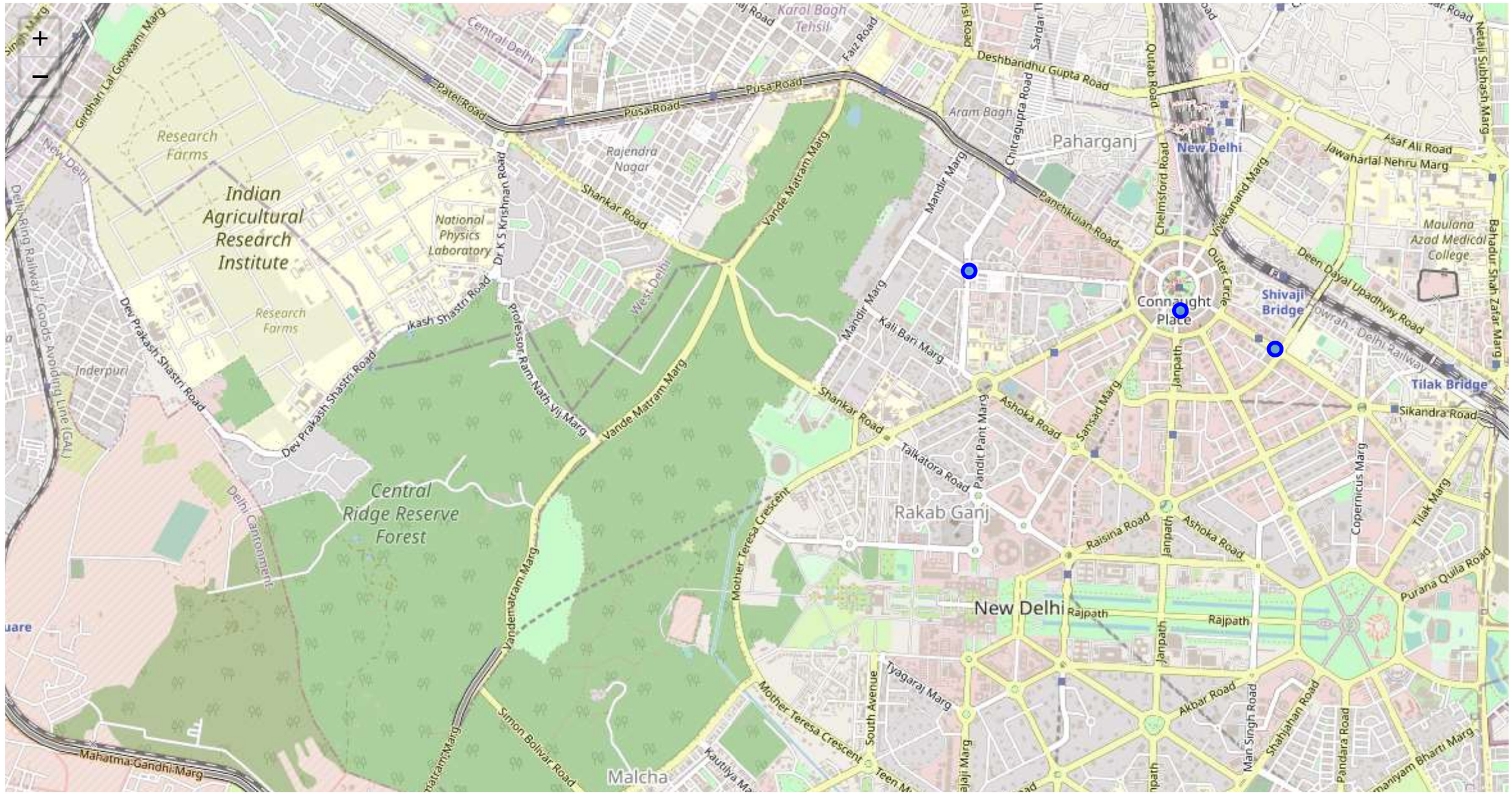
Create map of New Delhi using latitude and longitude values

In [11]:
```
map_ND= folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(ND_data['latitude'], ND_data['longitude'], ND_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_ND)

map_ND
```

Out[11]:



Define Foursquare Credentials and Version

In [12]:
```
#utilizing the Foursquare API to explore the neighborhoods


CLIENT_ID = 'QJWL3DNWSTUVOO1BHEQB4HAIQ3BGUEW3NPQGKYZBDD320Y21' # your Foursquare ID
CLIENT_SECRET = 'SAK1UAJFLQHYNREZ0K1VEJWYFP0LK3TKNTP22YNYWDPRUGN5' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: QJWL3DNWSTUVOO1BHEQB4HAIQ3BGUEW3NPQGKYZBDD320Y21
CLIENT_SECRET:SAK1UAJFLQHYNREZ0K1VEJWYFP0LK3TKNTP22YNYWDPRUGN5
```

Explore neighborhood in New Delhi

In [13]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=1000, LIMIT = 30):

    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/explore?&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 = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [14]:

```
# Now run the above function on each neighborhood and create a new dataframe called ND_venues.

ND_venues = getNearbyVenues(names=ND_data['Neighborhood'],
                                latitudes=ND_data['latitude'],
                                longitudes=ND_data['longitude']
                            )
```

```
Barakhamba Road
Chanakyapuri
Connaught Place
Gautampuri
Gole Market
Golf Links
INA Colony
Khan Market
Pragati Maidan
```

In [15]:

```
print(ND_venues.shape)
ND_venues
```

```
(186, 7)
```

Out[15]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Barakhamba Road	28.629142	77.226149	Tamasha	28.629663	77.221835	Gastropub
1	Barakhamba Road	28.629142	77.226149	Naturals Ice Cream	28.634455	77.222139	Ice Cream Shop
2	Barakhamba Road	28.629142	77.226149	The Lalit Hotel	28.631110	77.227450	Hotel
3	Barakhamba Road	28.629142	77.226149	Connaught Place | कनॉट प्लेस (Connaught Place)	28.632731	77.220018	Plaza
4	Barakhamba Road	28.629142	77.226149	Rajdhani Thali	28.629999	77.220401	Indian Restaurant
5	Barakhamba Road	28.629142	77.226149	Triveni Terrace Cafe	28.627295	77.232677	Café
6	Barakhamba Road	28.629142	77.226149	The Imperial	28.625548	77.218664	Hotel
7	Barakhamba Road	28.629142	77.226149	Cha Bar | चा बार	28.630920	77.222194	Tea Room
8	Barakhamba Road	28.629142	77.226149	Farzi Cafe	28.632581	77.221125	Molecular Gastronomy Restaurant
9	Barakhamba Road	28.629142	77.226149	Johnny Rockets	28.630457	77.219594	Bistro

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
10	Barakhamba Road	28.629142	77.226149	United Coffee House	28.632477	77.221120	Indian Restaurant
11	Barakhamba Road	28.629142	77.226149	Triveni Kala Sangam | त्रिवेणी कला संगम	28.627278	77.232137	Arcade
12	Barakhamba Road	28.629142	77.226149	Bengali Market | बंगाली मार्केट | বাংলা বাজার	28.629498	77.232020	Indian Restaurant
13	Barakhamba Road	28.629142	77.226149	Wenger's	28.633412	77.218292	Bakery
14	Barakhamba Road	28.629142	77.226149	Chaayos	28.631629	77.220364	Café
15	Barakhamba Road	28.629142	77.226149	Fabindia	28.632012	77.217729	Clothing Store
16	Barakhamba Road	28.629142	77.226149	Starbucks	28.632011	77.217731	Coffee Shop
17	Barakhamba Road	28.629142	77.226149	Panchayat Paan Parlour	28.634073	77.222645	Indian Restaurant
18	Barakhamba Road	28.629142	77.226149	Nathu's Pastry Shop	28.629422	77.232174	Bakery
19	Barakhamba Road	28.629142	77.226149	Unplugged	28.634345	77.221399	Lounge
20	Barakhamba Road	28.629142	77.226149	Ugrasen ki Baoli (Agrasen ki Baoli)	28.625820	77.224955	Historic Site
21	Barakhamba Road	28.629142	77.226149	The Spice Route	28.625577	77.218065	Asian Restaurant
22	Barakhamba Road	28.629142	77.226149	Khan Chacha | खान चाचा | خان چاچا	28.634202	77.220780	Indian Restaurant
23	Barakhamba Road	28.629142	77.226149	Pind Balluchi	28.630318	77.217600	North Indian Restaurant
24	Barakhamba Road	28.629142	77.226149	Chew	28.632002	77.222706	Asian Restaurant
25	Barakhamba Road	28.629142	77.226149	National School Of Drama | नेशनल स्कूल ऑफ ड्रामा	28.624658	77.233771	Theater
26	Barakhamba Road	28.629142	77.226149	Sagar Ratna	28.635487	77.220650	Indian Restaurant
27	Barakhamba Road	28.629142	77.226149	Starbucks	28.630106	77.220540	Coffee Shop
28	Barakhamba Road	28.629142	77.226149	Odeon Social	28.634414	77.220936	Café
29	Barakhamba Road	28.629142	77.226149	Barbeque Nation	28.630253	77.220985	BBQ Joint
30	Chanakyapuri	28.594677	77.188521	Nehru Park | नेहरू पार्क (Nehru Park)	28.591798	77.192860	Park
31	Chanakyapuri	28.594677	77.188521	Amour Bistro	28.601569	77.185923	Café
32	Chanakyapuri	28.594677	77.188521	Sanadige	28.601969	77.187020	Karnataka Restaurant
33	Chanakyapuri	28.594677	77.188521	Moti Mahal Delux	28.601677	77.187106	Indian Restaurant
34	Chanakyapuri	28.594677	77.188521	Lázeez Affaire	28.602237	77.186044	Indian Restaurant
35	Chanakyapuri	28.594677	77.188521	Kastro's House of Cigars	28.595765	77.198408	Smoke Shop
36	Chanakyapuri	28.594677	77.188521	Sidewok	28.602500	77.186422	Asian Restaurant
37	Chanakyapuri	28.594677	77.188521	Santushti Shopping Centre	28.595054	77.198204	Shopping Mall
38	Chanakyapuri	28.594677	77.188521	FEZ Restaurant	28.602451	77.186323	Moroccan Restaurant
39	Chanakyapuri	28.594677	77.188521	Skooter	28.596717	77.197053	Nightclub

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
40	Chanakyapuri	28.594677	77.188521	Sagar Ratna	28.595551	77.195764	Indian Restaurant
41	Chanakyapuri	28.594677	77.188521	The Oudh	28.596453	77.195404	Indian Restaurant
42	Chanakyapuri	28.594677	77.188521	nU.Delhi QBA	28.601498	77.186332	Pub
43	Chanakyapuri	28.594677	77.188521	The Ashok hotel	28.596844	77.195885	Hotel
44	Chanakyapuri	28.594677	77.188521	Lap The Club	28.596684	77.197869	Nightclub
45	Chanakyapuri	28.594677	77.188521	Fujiya	28.601526	77.186129	Chinese Restaurant
46	Chanakyapuri	28.594677	77.188521	Pangaea Lounge Delhi	28.596917	77.195493	Nightclub
47	Chanakyapuri	28.594677	77.188521	The Qube	28.599253	77.188648	Restaurant
48	Chanakyapuri	28.594677	77.188521	ACSA	28.598794	77.185086	General Entertainment
49	Chanakyapuri	28.594677	77.188521	The Pavilion	28.599917	77.183819	Coffee Shop
50	Chanakyapuri	28.594677	77.188521	Shraman	28.597384	77.195735	Vegetarian / Vegan Restaurant
51	Chanakyapuri	28.594677	77.188521	S Nom Nom	28.597093	77.196414	Chinese Restaurant
52	Chanakyapuri	28.594677	77.188521	Zerruco	28.597979	77.196227	Italian Restaurant
53	Chanakyapuri	28.594677	77.188521	Diggin Cafe	28.595135	77.197830	Café
54	Chanakyapuri	28.594677	77.188521	Basil & Thyme @ Santushti	28.595104	77.197970	Mediterranean Restaurant
55	Chanakyapuri	28.594677	77.188521	The Treat	28.592801	77.178855	Food Truck
56	Connaught Place	28.631383	77.219792	Connaught Place | कनॉट प्लेस (Connaught Place)	28.632731	77.220018	Plaza
57	Connaught Place	28.631383	77.219792	Fabindia	28.632012	77.217729	Clothing Store
58	Connaught Place	28.631383	77.219792	Wenger's	28.633412	77.218292	Bakery
59	Connaught Place	28.631383	77.219792	Starbucks	28.632011	77.217731	Coffee Shop
60	Connaught Place	28.631383	77.219792	Rajdhani Thali	28.629999	77.220401	Indian Restaurant
61	Connaught Place	28.631383	77.219792	HOTEL SARAVANA BHAVAN	28.632319	77.216445	South Indian Restaurant
62	Connaught Place	28.631383	77.219792	Naturals Ice Cream	28.634455	77.222139	Ice Cream Shop
63	Connaught Place	28.631383	77.219792	Farzi Cafe	28.632581	77.221125	Molecular Gastronomy Restaurant
64	Connaught Place	28.631383	77.219792	Johnny Rockets	28.630457	77.219594	Bistro
65	Connaught Place	28.631383	77.219792	Jain Chawal Wale	28.631763	77.215919	Food Truck
66	Connaught Place	28.631383	77.219792	Pind Balluchi	28.630318	77.217600	North Indian Restaurant
67	Connaught Place	28.631383	77.219792	United Coffee House	28.632477	77.221120	Indian Restaurant
68	Connaught Place	28.631383	77.219792	Chaayos	28.631629	77.220364	Café
69	Connaught Place	28.631383	77.219792	Tamasha	28.629663	77.221835	Gastropub

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
70	Connaught Place	28.631383	77.219792	Khan Chacha | खान चाचा | خان چاچا	28.634202	77.220780	Indian Restaurant
71	Connaught Place	28.631383	77.219792	The Imperial	28.625548	77.218664	Hotel
72	Connaught Place	28.631383	77.219792	Unplugged	28.634345	77.221399	Lounge
73	Connaught Place	28.631383	77.219792	Cha Bar | चा बार	28.630920	77.222194	Tea Room
74	Connaught Place	28.631383	77.219792	Sagar Ratna	28.635487	77.220650	Indian Restaurant
75	Connaught Place	28.631383	77.219792	Odeon Social	28.634414	77.220936	Café
76	Connaught Place	28.631383	77.219792	Lord of the drinks	28.631748	77.216896	Bar
77	Connaught Place	28.631383	77.219792	Nizam's Kathi Kabab | निज़ाम काठी कबाब	28.634858	77.219462	Indian Restaurant
78	Connaught Place	28.631383	77.219792	Berco's	28.632407	77.217323	Chinese Restaurant
79	Connaught Place	28.631383	77.219792	Starbucks	28.630106	77.220540	Coffee Shop
80	Connaught Place	28.631383	77.219792	Nando's	28.630947	77.219721	Portuguese Restaurant
81	Connaught Place	28.631383	77.219792	Panchayat Paan Parlour	28.634073	77.222645	Indian Restaurant
82	Connaught Place	28.631383	77.219792	Odeon Pan Shop	28.634234	77.220798	Food & Drink Shop
83	Connaught Place	28.631383	77.219792	Kwality Restaurant	28.629744	77.217431	Indian Restaurant
84	Connaught Place	28.631383	77.219792	Route 04	28.634890	77.220225	Bar
85	Connaught Place	28.631383	77.219792	The Beer Cafe	28.629139	77.219603	Beer Garden
86	Gautampuri	28.511570	77.302623	Local tea shop	28.513199	77.296747	Snack Place
87	Gautampuri	28.511570	77.302623	Sagar Ratna	28.507452	77.297201	Indian Restaurant
88	Gautampuri	28.511570	77.302623	Pind Balluchi	28.507440	77.297209	Indian Restaurant
89	Gautampuri	28.511570	77.302623	Faridabad Railway Station	28.504600	77.299299	Train Station
90	Gautampuri	28.511570	77.302623	Tughlakabad Railway Station	28.504789	77.296243	Train Station
91	Gole Market	28.633719	77.205627	My Bar	28.640716	77.210683	Bar
92	Gole Market	28.633719	77.205627	Kaleva	28.633299	77.208488	Snack Place
93	Gole Market	28.633719	77.205627	Malahotra Restaurant	28.641052	77.210333	Indian Restaurant
94	Gole Market	28.633719	77.205627	Sakura	28.631310	77.208104	Japanese Restaurant
95	Gole Market	28.633719	77.205627	Ramakrishna Ashram Marg Metro Station	28.639131	77.208567	Light Rail Station
96	Gole Market	28.633719	77.205627	Metropolis Hotel / Tourist Home	28.640703	77.210654	Breakfast Spot
97	Gole Market	28.633719	77.205627	Appetite German bakery	28.640857	77.211759	Snack Place
98	Gole Market	28.633719	77.205627	natraj yes please	28.640664	77.209441	Hotel
99	Gole Market	28.633719	77.205627	Café Festa	28.641017	77.210302	Indian Restaurant

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
100	Golf Links	28.595970	77.231163	Delhi Golf Club	28.599968	77.233288	Golf Course
101	Golf Links	28.595970	77.231163	The Big Chill Cafe	28.600686	77.227636	Italian Restaurant
102	Golf Links	28.595970	77.231163	Perch	28.599987	77.226350	Café
103	Golf Links	28.595970	77.231163	SODABOTTLEOPENERWALA	28.600141	77.226273	Irani Cafe
104	Golf Links	28.595970	77.231163	Khan Market | खान मार्केट (Khan Market)	28.600342	77.226923	Market
105	Golf Links	28.595970	77.231163	Indian Accent	28.592026	77.238006	Indian Restaurant
106	Golf Links	28.595970	77.231163	Bahri Sons Booksellers	28.599977	77.227134	Bookstore
107	Golf Links	28.595970	77.231163	The Lodhi Hotel	28.591669	77.238131	Hotel
108	Golf Links	28.595970	77.231163	Smoke House Deli	28.599850	77.226900	French Restaurant
109	Golf Links	28.595970	77.231163	Yellow Brick Road	28.601884	77.229298	Restaurant
110	Golf Links	28.595970	77.231163	L'Opera	28.599784	77.226144	Café
111	Golf Links	28.595970	77.231163	Public Affair	28.600030	77.226478	Lounge
112	Golf Links	28.595970	77.231163	Full Circle Bookstore	28.600200	77.227198	Bookstore
113	Golf Links	28.595970	77.231163	The Oberoi	28.596564	77.239634	Hotel
114	Golf Links	28.595970	77.231163	La Vie	28.600449	77.226952	Mediterranean Restaurant
115	Golf Links	28.595970	77.231163	Élan @ The Lodhi	28.591599	77.238207	Restaurant
116	Golf Links	28.595970	77.231163	Anokhi	28.600381	77.227859	Boutique
117	Golf Links	28.595970	77.231163	Khan Chacha	28.600618	77.227237	Indian Restaurant
118	Golf Links	28.595970	77.231163	On The Waterfront	28.592061	77.237951	Hotel Bar
119	Golf Links	28.595970	77.231163	Good Earth	28.599638	77.226448	Furniture / Home Store
120	Golf Links	28.595970	77.231163	La Bodega	28.600276	77.227653	Mexican Restaurant
121	Golf Links	28.595970	77.231163	Mamagoto	28.600469	77.227272	Chinese Restaurant
122	Golf Links	28.595970	77.231163	Wok In The Clouds	28.600502	77.227230	Indian Restaurant
123	Golf Links	28.595970	77.231163	Vivanta by Taj - Ambassador	28.601690	77.229008	Hotel
124	Golf Links	28.595970	77.231163	Harry's - Singapore's Favorite Bar	28.600158	77.226878	Bar
125	Golf Links	28.595970	77.231163	The Oberoi Patisserie and Delicatessen	28.596443	77.240084	Bakery
126	Golf Links	28.595970	77.231163	360	28.596600	77.239609	Restaurant
127	Golf Links	28.595970	77.231163	Starbucks	28.600893	77.227091	Coffee Shop
128	Golf Links	28.595970	77.231163	The Blue Door Cafe	28.600068	77.226582	Café
129	Golf Links	28.595970	77.231163	Smokeys BBQ & Grill	28.600465	77.227136	BBQ Joint

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
130	INA Colony	28.579472	77.212451	Dilli Haat | दिल्ली हाट (Dilli Haat)	28.573132	77.208810	Market
131	INA Colony	28.579472	77.212451	Nagaland Stall	28.573070	77.206521	Restaurant
132	INA Colony	28.579472	77.212451	Thyagraj Sports Complex	28.577528	77.218159	Athletics & Sports
133	INA Colony	28.579472	77.212451	INA Metro Station	28.574083	77.209296	Train Station
134	INA Colony	28.579472	77.212451	Chidambaram's New Madras Hotel	28.580657	77.220943	Indian Restaurant
135	INA Colony	28.579472	77.212451	Hot Chimney	28.583702	77.220574	Indian Restaurant
136	INA Colony	28.579472	77.212451	Safdarjung Airport	28.582953	77.211254	Airport
137	INA Colony	28.579472	77.212451	Kerala Hotel	28.574073	77.209929	Indian Restaurant
138	INA Colony	28.579472	77.212451	East Kidwai Nagar Market	28.573780	77.216217	Dim Sum Restaurant
139	INA Colony	28.579472	77.212451	Golden Bakery	28.584691	77.219703	Bakery
140	INA Colony	28.579472	77.212451	Dewan's Tea And Coffee	28.580040	77.221904	Coffee Shop
141	Khan Market	28.600135	77.226491	The Big Chill Cafe	28.600686	77.227636	Italian Restaurant
142	Khan Market	28.600135	77.226491	SODABOTTLEOPENERWALA	28.600141	77.226273	Irani Cafe
143	Khan Market	28.600135	77.226491	Perch	28.599987	77.226350	Café
144	Khan Market	28.600135	77.226491	Khan Market | खान मार्केट (Khan Market)	28.600342	77.226923	Market
145	Khan Market	28.600135	77.226491	Bahri Sons Booksellers	28.599977	77.227134	Bookstore
146	Khan Market	28.600135	77.226491	L'Opera	28.599784	77.226144	Café
147	Khan Market	28.600135	77.226491	Public Affair	28.600030	77.226478	Lounge
148	Khan Market	28.600135	77.226491	Smoke House Deli	28.599850	77.226900	French Restaurant
149	Khan Market	28.600135	77.226491	Full Circle Bookstore	28.600200	77.227198	Bookstore
150	Khan Market	28.600135	77.226491	Yellow Brick Road	28.601884	77.229298	Restaurant
151	Khan Market	28.600135	77.226491	La Vie	28.600449	77.226952	Mediterranean Restaurant
152	Khan Market	28.600135	77.226491	Khan Chacha	28.600618	77.227237	Indian Restaurant
153	Khan Market	28.600135	77.226491	Varq | वर्क	28.604547	77.223781	Indian Restaurant
154	Khan Market	28.600135	77.226491	Anokhi	28.600381	77.227859	Boutique
155	Khan Market	28.600135	77.226491	Good Earth	28.599638	77.226448	Furniture / Home Store
156	Khan Market	28.600135	77.226491	Mamagoto	28.600469	77.227272	Chinese Restaurant
157	Khan Market	28.600135	77.226491	La Bodega	28.600276	77.227653	Mexican Restaurant
158	Khan Market	28.600135	77.226491	Wok In The Clouds	28.600502	77.227230	Indian Restaurant
159	Khan Market	28.600135	77.226491	Harry's - Singapore's Favorite Bar	28.600158	77.226878	Bar

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
160	Khan Market	28.600135	77.226491	Delhi Golf Club	28.599968	77.233288	Golf Course
161	Khan Market	28.600135	77.226491	Starbucks	28.600893	77.227091	Coffee Shop
162	Khan Market	28.600135	77.226491	The House of Ming	28.604504	77.223592	Chinese Restaurant
163	Khan Market	28.600135	77.226491	Vivanta by Taj - Ambassador	28.601690	77.229008	Hotel
164	Khan Market	28.600135	77.226491	The Blue Door Cafe	28.600068	77.226582	Café
165	Khan Market	28.600135	77.226491	Wasabi by Morimoto	28.604580	77.223672	Japanese Restaurant
166	Khan Market	28.600135	77.226491	Smokeys BBQ & Grill	28.600465	77.227136	BBQ Joint
167	Khan Market	28.600135	77.226491	Gulati Restaurant	28.608010	77.229989	Indian Restaurant
168	Khan Market	28.600135	77.226491	Machan	28.604657	77.223706	Restaurant
169	Khan Market	28.600135	77.226491	The Chatter House	28.600298	77.227061	Café
170	Khan Market	28.600135	77.226491	Ricks	28.604600	77.223714	Hotel Bar
171	Pragati Maidan	28.623459	77.242512	National School Of Drama | नेशनल स्कूल ऑफ ड्रामा	28.624658	77.233771	Theater
172	Pragati Maidan	28.623459	77.242512	Shri Ram Centre for Performing Arts	28.627068	77.233291	Theater
173	Pragati Maidan	28.623459	77.242512	Costa Coffee	28.617321	77.245805	Coffee Shop
174	Pragati Maidan	28.623459	77.242512	Pragati maidan hall -18	28.621372	77.241556	Art Gallery
175	Pragati Maidan	28.623459	77.242512	Le Meridien Hotel	28.624484	77.239979	Hotel
176	Pragati Maidan	28.623459	77.242512	Killer ITO Xing	28.627666	77.241253	Plaza
177	Pragati Maidan	28.623459	77.242512	32nd India International Trade Fair, Pragati M...	28.619173	77.244183	Flea Market
178	Pragati Maidan	28.623459	77.242512	Pearey Lal Bhawan	28.628233	77.241488	Performing Arts Venue
179	Pragati Maidan	28.623459	77.242512	Turant	28.620974	77.237221	Chinese Restaurant
180	Pragati Maidan	28.623459	77.242512	Shankar's International Doll Museum	28.629240	77.241405	Art Museum
181	Pragati Maidan	28.623459	77.242512	Tilak Bridge Railway Station	28.627775	77.237828	Train Station
182	Pragati Maidan	28.623459	77.242512	fabindia	28.629612	77.244705	Clothing Store
183	Pragati Maidan	28.623459	77.242512	Abix's Crib	28.620314	77.235682	Furniture / Home Store
184	Pragati Maidan	28.623459	77.242512	Udupi Cafe	28.630797	77.241259	Udupi Restaurant
185	Pragati Maidan	28.623459	77.242512	Mandi House Metro Station	28.625733	77.233681	Light Rail Station

In [16]:
```
# to check how many venues were returned for each neighborhood
ND_venues.groupby('Neighborhood').count()
```

Out[16]:

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Barakhamba Road	30	30	30	30	30	30
Chanakyapuri	26	26	26	26	26	26
Connaught Place	30	30	30	30	30	30
Gautampuri	5	5	5	5	5	5
Gole Market	9	9	9	9	9	9
Golf Links	30	30	30	30	30	30
INA Colony	11	11	11	11	11	11
Khan Market	30	30	30	30	30	30
Pragati Maidan	15	15	15	15	15	15

In [17]:
```
print('There are {} uniques categories.'.format(len(ND_venues['Venue Category'].unique())))
```

```
There are 60 uniques categories.
```

Analyze Each Neighborhood

In [18]:
```
# one hot encoding
ND_onehot = pd.get_dummies(ND_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
ND_onehot['Neighborhood'] = ND_venues['Neighborhood']

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

ND_onehot.head()
```

Out[18]:

	Neighborhood	Airport	Arcade	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	BBQ Joint	Bakery	Bar	Beer Garden	Bistro	Bookstore	Boutique	Breakfast Spot	Café	Chinese Restaurant	Clothing Store	Coffee Shop	Dim Sum Restaurant	I
0	Barakhamba Road	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	Barakhamba Road	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	Barakhamba Road	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

	Neighborhood	Airport	Arcade	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	BBQ Joint	Bakery	Bar	Beer Garden	Bistro	Bookstore	Boutique	Breakfast Spot	Café	Chinese Restaurant	Clothing Store	Coffee Shop	Dim Sum Restaurant	I
3	Barakhamba Road	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	Barakhamba Road	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

In [19]:
```
ND_onehot.shape
```

Out[19]: (186, 61)

let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

In [20]:
```
ND_grouped = ND_onehot.groupby('Neighborhood').mean().reset_index()
ND_grouped
```

Out[20]:

	Neighborhood	Airport	Arcade	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	BBQ Joint	Bakery	Bar	Beer Garden	Bistro	Bookstore	Boutique	Breakfast Spot	Café	Chinese Restaurant	Clothing Store
0	Barakhamba Road	0.000000	0.033333	0.000000	0.000000	0.066667	0.000000	0.033333	0.066667	0.000000	0.000000	0.033333	0.000000	0.000000	0.000000	0.100000	0.000000	0.033333
1	Chanakyapuri	0.000000	0.000000	0.000000	0.000000	0.038462	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.076923	0.076923	0.000000
2	Connaught Place	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0.066667	0.033333	0.033333	0.000000	0.000000	0.000000	0.066667	0.033333	0.033333
3	Gautampuri	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	Gole Market	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.111111	0.000000	0.000000	0.000000	0.000000	0.111111	0.000000	0.000000	0.000000
5	Golf Links	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0.033333	0.033333	0.000000	0.000000	0.066667	0.033333	0.000000	0.100000	0.033333	0.000000
6	INA Colony	0.090909	0.000000	0.000000	0.000000	0.000000	0.090909	0.000000	0.090909	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	Khan Market	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.033333	0.000000	0.033333	0.000000	0.000000	0.066667	0.033333	0.000000	0.133333	0.066667	0.000000
8	Pragati Maidan	0.000000	0.000000	0.066667	0.066667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.066667	0.066667

In [21]:
```
ND_grouped.shape
```

Out[21]: (9, 61)

Function to sort the venues in descending order

In [22]:
```
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]
```

Let's create the new dataframe and display the top 9 venues for each neighborhood.

In [27]:
```
num_top_venues = 9

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = ND_grouped['Neighborhood']

for ind in np.arange(ND_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(ND_grouped.iloc[ind, :], num_top_venues)
```

```
neighborhoods_venues_sorted.head()
```

Out[27]:

	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
0	Barakhamba Road	Indian Restaurant	Café	Asian Restaurant	Hotel	Bakery	Coffee Shop	Ice Cream Shop	Arcade	Molecular Gastronomy Restaurant
1	Chanakyapuri	Indian Restaurant	Nightclub	Café	Chinese Restaurant	Vegetarian / Vegan Restaurant	Hotel	Mediterranean Restaurant	General Entertainment	Moroccan Restaurant
2	Connaught Place	Indian Restaurant	Café	Bar	Coffee Shop	Ice Cream Shop	Gastropub	Plaza	Portuguese Restaurant	Food Truck
3	Gautampuri	Indian Restaurant	Train Station	Snack Place	North Indian Restaurant	Irani Cafe	Italian Restaurant	Japanese Restaurant	Karnataka Restaurant	Light Rail Station
4	Gole Market	Indian Restaurant	Snack Place	Light Rail Station	Breakfast Spot	Bar	Hotel	Japanese Restaurant	Karnataka Restaurant	North Indian Restaurant

Transform dataframe for clustering

In [28]:
```
ND_grouped_clustering =ND_grouped.drop('Neighborhood', axis=1)
```

Find optimum k value for k-means clustering

In [29]:
```
SSE = []
for cluster in range(1,9):
    kmeans = KMeans(n_jobs = -1, n_clusters = cluster, init='k-means++')
    kmeans.fit(ND_grouped_clustering)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
frame = pd.DataFrame({'Cluster':range(1,9), 'SSE':SSE})
plt.figure(figsize=(12,6))
plt.plot(frame['Cluster'], frame['SSE'], marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/usr/local/share/jupyter/kernels/python39/scripts/launch_ipykernel.py in <module>
      1 SSE = []
      2 for cluster in range(1,9):
----> 3     kmeans = KMeans(n_jobs = -1, n_clusters = cluster, init='k-means++')
      4     kmeans.fit(ND_grouped_clustering)
      5     SSE.append(kmeans.inertia_)

TypeError: __init__() got an unexpected keyword argument 'n_jobs'
```

When we changed the cluster value from 1 to 3, the inertia value reduced very sharply. This decrease in the inertia value reduces and eventually becomes constant as we increase the number of clusters further. Here, we can choose any number of clusters between 4 and 8. We can have 5, 6, or even 8 clusters. We must also look at the computation cost while deciding the number of clusters. If we increase the number of clusters, the computation cost will also increase. So, We will now run K-means clustering for K=4.

Cluster the neighborhood into 4 clusters

In [26]:
```
kclusters = 4

ND_grouped_clustering = ND_grouped.drop('Neighborhood', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[26]:
```
array([0, 0, 0, 2, 1, 0, 0, 0, 3], dtype=int32)
```

Let's create a new dataframe that includes the cluster as well as the top 9 venues for each neighborhood.

In [27]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_ )

ND_merged = ND_data

# merge ND_grouped with toronto_data to add latitude/longitude for each neighborhood
ND_merged = ND_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')
```

```
ND_merged.head() # check the last columns!
```

Out[27]:

	Borough	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	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue
0	New Delhi	Barakhamba Road	28.629142	77.226149	0	Indian Restaurant	Hotel	Lounge	Café	Bakery	Plaza	Gastropub	Ice Cream Shop	Coffee Shop
1	New Delhi	Chanakyapuri	28.594678	77.188521	0	Indian Restaurant	Nightclub	Café	Chinese Restaurant	Park	Asian Restaurant	Coffee Shop	Hotel	Italian Restaurant
2	New Delhi	Connaught Place	28.631383	77.219792	0	Indian Restaurant	Café	South Indian Restaurant	Lounge	Bistro	Gastropub	Ice Cream Shop	Food Truck	Flea Market
3	New Delhi	Gautampuri	28.511570	77.302623	2	Indian Restaurant	Train Station	Snack Place	Café	Hotel	Historic Site	Golf Course	Gastropub	Furniture / Home Store
4	New Delhi	Gole Market	28.633719	77.205627	1	Snack Place	Indian Restaurant	Japanese Restaurant	Light Rail Station	Bar	Breakfast Spot	Hotel	Coffee Shop	Historic Site

In [28]:
```
ND_merged= ND_merged.dropna()
```

In [29]:
```
ND_merged['Cluster Labels'].round(0)
```

Out[29]:
```
0    0
1    0
2    0
3    2
4    1
5    0
6    0
7    0
8    3
Name: Cluster Labels, dtype: int32
```

In [30]:
```
ND_merged['Cluster Labels'] = ND_merged['Cluster Labels'].astype(int)
```

In [31]:
```
ND_merged.head(10)
```

Out[31]:

	Borough	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	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue
0	New Delhi	Barakhamba Road	28.629142	77.226149	0	Indian Restaurant	Hotel	Lounge	Café	Bakery	Plaza	Gastropub	Ice Cream Shop	Coffee Shop
1	New Delhi	Chanakyapuri	28.594678	77.188521	0	Indian Restaurant	Nightclub	Café	Chinese Restaurant	Park	Asian Restaurant	Coffee Shop	Hotel	Italian Restaurant

	Borough	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	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue
2	New Delhi	Connaught Place	28.631383	77.219792	0	Indian Restaurant	Café	South Indian Restaurant	Lounge	Bistro	Gastropub	Ice Cream Shop	Food Truck	Flea Market
3	New Delhi	Gautampuri	28.511570	77.302623	2	Indian Restaurant	Train Station	Snack Place	Café	Hotel	Historic Site	Golf Course	Gastropub	Furniture / Home Store
4	New Delhi	Gole Market	28.633719	77.205627	1	Snack Place	Indian Restaurant	Japanese Restaurant	Light Rail Station	Bar	Breakfast Spot	Hotel	Coffee Shop	Historic Site
5	New Delhi	Golf Links	28.595970	77.231163	0	Indian Restaurant	Restaurant	Café	Hotel	Coffee Shop	Bookstore	Dessert Shop	Hotel Bar	Golf Course
6	New Delhi	INA Colony	28.579472	77.212451	0	Indian Restaurant	Restaurant	Antique Shop	Athletics & Sports	Bakery	Coffee Shop	Market	Metro Station	Airport
7	New Delhi	Khan Market	28.600135	77.226491	0	Indian Restaurant	Café	Restaurant	Coffee Shop	Bookstore	Chinese Restaurant	Mexican Restaurant	Boutique	Mediterranean Restaurant
8	New Delhi	Pragati Maidan	28.623459	77.242512	3	Theater	Indian Restaurant	Train Station	Clothing Store	Udupi Restaurant	Furniture / Home Store	Art Museum	Coffee Shop	Café

4:- Results

Finally, let's visualize the resulting clusters

In [32]:

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
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(ND_merged['latitude'], ND_merged['longitude'], ND_merged['Neighborhood'], ND_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[32]: Make this Notebook Trusted to load map: File -> Trust Notebook

Examine Clusters

Cluster 1

In [33]:
```
ND_merged.loc[ND_merged['Cluster Labels'] == 0, ND_merged.columns[[1] + list(range(5, ND_merged.shape[1]))]]
```

Out[33]:

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

	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
0	Barakhamba Road	Indian Restaurant	Hotel	Lounge	Café	Bakery	Plaza	Gastropub	Ice Cream Shop	Coffee Shop
1	Chanakyapuri	Indian Restaurant	Nightclub	Café	Chinese Restaurant	Park	Asian Restaurant	Coffee Shop	Hotel	Italian Restaurant
2	Connaught Place	Indian Restaurant	Café	South Indian Restaurant	Lounge	Bistro	Gastropub	Ice Cream Shop	Food Truck	Flea Market
5	Golf Links	Indian Restaurant	Restaurant	Café	Hotel	Coffee Shop	Bookstore	Dessert Shop	Hotel Bar	Golf Course
6	INA Colony	Indian Restaurant	Restaurant	Antique Shop	Athletics & Sports	Bakery	Coffee Shop	Market	Metro Station	Airport
7	Khan Market	Indian Restaurant	Café	Restaurant	Coffee Shop	Bookstore	Chinese Restaurant	Mexican Restaurant	Boutique	Mediterranean Restaurant

Cluster 2

In [34]:
```
ND_merged.loc[ND_merged['Cluster Labels'] == 1, ND_merged.columns[[1] + list(range(5, ND_merged.shape[1]))]]
```

Out[34]:

	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
4	Gole Market	Snack Place	Indian Restaurant	Japanese Restaurant	Light Rail Station	Bar	Breakfast Spot	Hotel	Coffee Shop	Historic Site

Cluster 3

In [35]:
```
ND_merged.loc[ND_merged['Cluster Labels'] == 2, ND_merged.columns[[1] + list(range(5, ND_merged.shape[1]))]]
```

Out[35]:

	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
3	Gautampuri	Indian Restaurant	Train Station	Snack Place	Café	Hotel	Historic Site	Golf Course	Gastropub	Furniture / Home Store

Cluster 4

In [36]:
```
ND_merged.loc[ND_merged['Cluster Labels'] == 3, ND_merged.columns[[1] + list(range(5, ND_merged.shape[1]))]]
```

Out[36]:

	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
8	Pragati Maidan	Theater	Indian Restaurant	Train Station	Clothing Store	Udupi Restaurant	Furniture / Home Store	Art Museum	Coffee Shop	Café

Here, We can clearly see that Cluster 1 is the cluster with the restaurants as the most common venue.

We will plot the mean value of Indian restaurants for each selected neighborhood in the cluster to study the presence of competition in each neighborhood.

In [37]:
```
ND_grouped_select = pd.DataFrame()
ND_grouped_select['Neighborhood'] = ND_merged['Neighborhood']

ND_grouped_select = ND_grouped_select.join(ND_grouped.set_index('Neighborhood'), on='Neighborhood')
ND_grouped_select.dropna(inplace=True)
ND_grouped_select.sort_values('Indian Restaurant',ascending=False,inplace=True)

print(ND_grouped_select.shape)
ND_grouped_select.head()
```

```
(9, 61)
```

Out[37]:

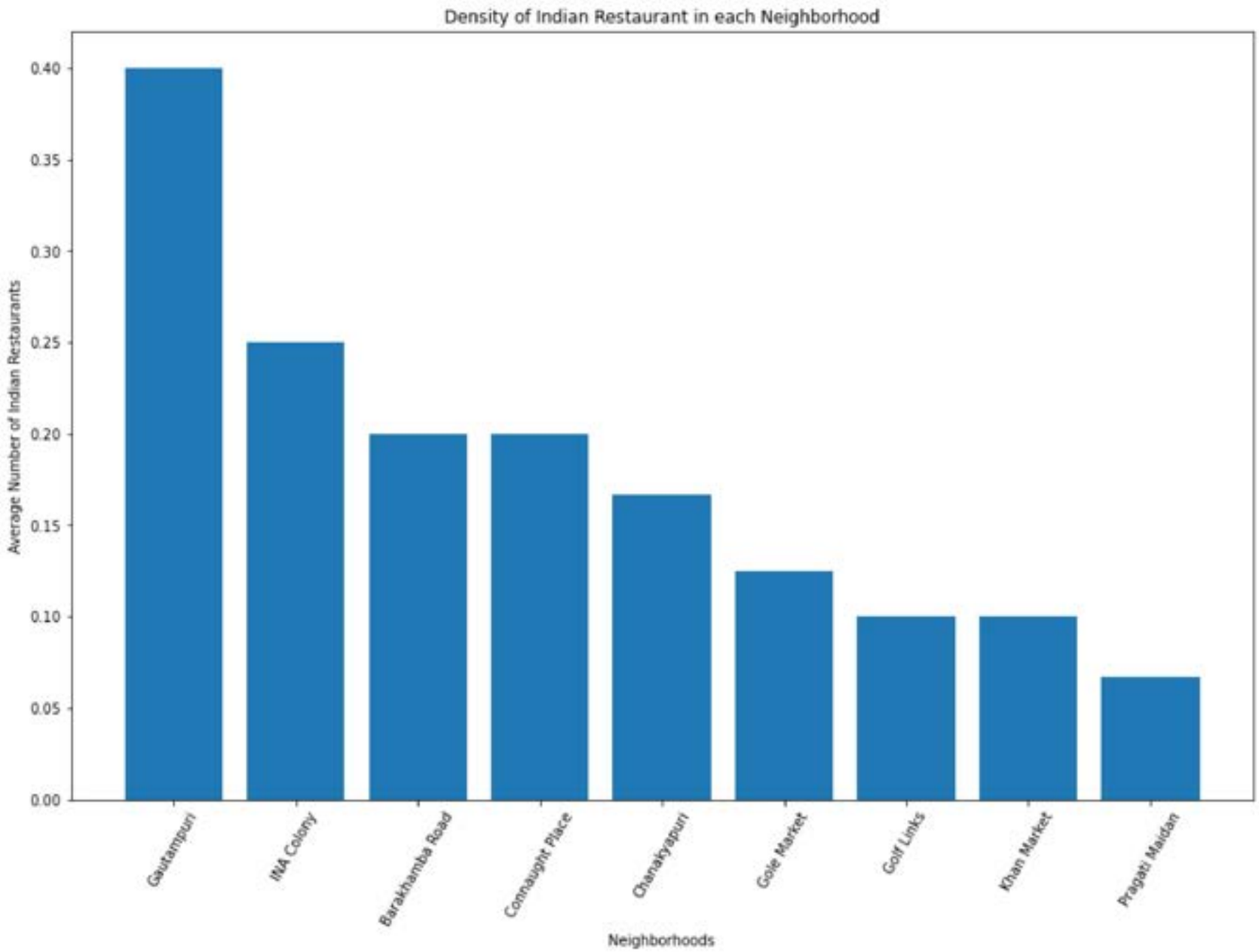
	Neighborhood	Airport	Antique Shop	Arcade	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	BBQ Joint	Bakery	Bar	Bistro	Bookstore	Boutique	Breakfast Spot	Café	Chinese Restaurant	Clothing Store	
3	Gautampuri	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.000000	0.000000	0.000000	[
6	INA Colony	0.083333	0.083333	0.000000	0.000000	0.0	0.000000	0.083333	0.000000	0.083333	0.000000	0.000000	0.0	0.0	0.0	0.000000	0.000000	0.000000	[
0	Barakhamba Road	0.000000	0.000000	0.033333	0.033333	0.0	0.033333	0.000000	0.033333	0.066667	0.033333	0.033333	0.0	0.0	0.0	0.066667	0.000000	0.000000	[
2	Connaught Place	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.033333	0.033333	0.033333	0.033333	0.0	0.0	0.0	0.100000	0.033333	0.033333	[
1	Chanakyapuri	0.000000	0.000000	0.000000	0.000000	0.0	0.041667	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.083333	0.083333	0.000000	[

There are 9 neighborhoods in New Delhi, we will use to get the average venues data

In [38]:
```
label = list(ND_grouped_select['Neighborhood'])
height = list(ND_grouped_select['Indian Restaurant'])

index = np.arange(len(label))

plt.figure(figsize=(15,10))
plt.bar(index,height)
plt.xlabel('Neighborhoods')
plt.ylabel('Average Number of Indian Restaurants')
plt.xticks(rotation=60)
plt.xticks(index,label)
plt.title('Density of Indian Restaurant in each Neighborhood')
plt.show()
```



5:- Discussion

From the graph, we can see that the first neighborhood, Gautampuri, has the most competition for Indian Restaurants in New Delhi. This indicating that it has the greatest obstacles in opening a new restaurant. Gautampuri has almost double the competition than any other neighborhoods. Connaught Place is the second neighborhood with the most Indian restaurants, followed by INA Colony. Barakhamba Road and Chanakyapuri has almost same but moderate competition. However, the following neighborhoods have moderate competition, which will enable a new business to establish easily:

Gole Market

Golf Links

Khan Market

Pragati Maidan

Note that these recommendations are based on some assumptions of the analysis,like:

1- radius of the oppourtunity of each neighborhood was considered as 1000 meters from the location,

2- recommendation opportunities are based on absence of a restaurant which is likely to be appreciated in the top 9 venues

6:- Conclusion

This project recommends some of the ideal places to open Indian restaurant in New Delhi, India.The analysis shows there are better chances for opening restaurant in Gole Market,Golf Links,Khan Market, and Pragati Maidan.INA Colony, Barakhamba Road and Chanakyapuri are also good location. This analysis can be helpful for the individuals looking for opening a restaurant or expanding business. This analysis shows the feasible venture and competion landscape of the area.

Final decission on optimal restaurant location will be made by stakeholders based on specific characteristics of neighborhoods and locations in every recommended zone, taking into consideration additional factors like attractiveness of each location , levels of noise / proximity to major roads, real estate availability, prices, social and economic dynamics of every neighborhood etc.

In [ ]: