**Report**

**Battle of Neighborhood**

**-Harika Penjerla**

**Project Requirements**

**Firstly, you will be required to submit the following:**

A description of the problem and a discussion of the background.

A description of the data and how it will be used to solve the problem.

**Secondly, the final deliverables of the project will be:**

A link to your Notebook on your Github repository, showing your code.

A full report consisting of all of the following components:

Introduction where you discuss the business problem and who would be interested in this project.

Data where you describe the data that will be used to solve the problem and the source of the data.

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.

Results section where you discuss the results.

Discussion section where you discuss any observations you noted and any recommendations you can make based on the results.

Conclusion section where you conclude the report.

**Business Problem and Background:**

I’m really enthusiastic to start a business, mostly interested in Restaurants, coffee shop or preschool. For IBM coursera capstone project I want to use my skills to predict the recommended location to open coffee shop in Cleveland, Ohio. Cleveland is a beautiful city in Ohio with nice and kind people. It has decent number of restaurants and coffee shops. In this analysis trying to find location for successful business. Even tough it is stressed out city in US, I believe there is good chances to run a successful coffee shop.

**Coffee is culture**

Going back in time, we’ve seen that coffee is not just a drink. It is actually a lot more. In the past, people gathered in houses to share a cup of coffee and exchange information.

In North America and a lot of western European countries, Starbucks and other coffee chains have dominated the market because they are providing to the customers a complete experience around coffee. Coffee and friends, coffee and work, coffee and snacks and coffee to go! The Irish mix coffee with whiskey and they call it “Irish coffee”. In Italy the espresso and espresso machine were born and made it the favorite whole-day beverage of Italians. In Greece, there is the “kafenio”, an old fashioned coffee shop for old gentlemen, where they drink Greek coffee and exchange political ideas or they play cards and a board game called “tavli”. Then also, in some countries, like Colombia or Brazil, the whole economy is based in coffee.

So this makes us really curious to know more interesting facts about coffee as a commodity itself but also as a beverage.

For more info please visit: <https://www.agiboo.com/16-interesting-facts-about-coffee/>

**Audience**

Business investors who want to start a coffee shop in particular area Cleveland,Ohio.

Novice people who has interest in exploring the data realted insights for starting their own business.

**Data Selection**

The data is collected from foursquare API about coffee shops in Clelveand, Ohio and visulaize the clean dataframe for recommending the ideal location.

**Methodology**

Here steps used in analysis and modeling the project:

1. Importing libraries needed.
2. Setting up credentials for foursquare to fetch the data and all necessary steps to get data.
3. Converting json to dataframe and cleaning the data
4. Visulaizing the cofee shops using folium
5. Apply K-means clustereing.
6. Recommendation/Conclusion based on analysis.

**References**

<https://labs.cognitiveclass.ai/tools/jupyterlab/lab/tree/labs/DP0701EN/DP0701EN-3-3-2-Neighborhoods-New-York-py-v1.0.ipynb> <https://stackoverflow.com/questions/48085110/no-module-named-folium-plugins-python-3-6> <https://medium.com/@aboutiana/a-brief-guide-to-using-foursquare-api-with-a-hands-on-example-on-python-6fc4d5451203> <https://developer.foursquare.com/> <https://towardsdatascience.com/data-101s-spatial-visualizations-and-analysis-in-python-with-folium-39730da2adf> <https://dev.to/chijade/capstone-project-the-battle-of-neighborhoods-4ha3>

In [30]:

*# Libraries needed are imported here*

**import** **sys**

**import** **numpy** **as** **np**

**import** **requests** *# library for handle requests*

**import** **pandas** **as** **pd** *# library for data analsysis*

!pip install folium

**import** **folium** *# library for map rendering*

**import** **json**

**import** **io**

**import** **matplotlib.pyplot** **as** **plt**

**from** **bs4** **import** BeautifulSoup

**from** **sklearn.cluster** **import** KMeans

**from** **geopy.geocoders** **import** Nominatim *#location to latitude and longtitude*

**from** **folium** **import** plugins

print('Libraries Imported')

*# Setting up foursquare credentials to fetch data*

CLIENT\_ID = 'N4GIWTCJ0DPW3ST5P0EEVB25W0JHGW5ZTBJZKDKUBFTJGOK0' *# your Foursquare ID*

CLIENT\_SECRET = 'KRLYG4052BTLWMRWK445JLY10SHUBRLCOBRMAQLO1RRZ1PF0' *# your Foursquare Secret*

VERSION = '20180604'

LIMIT = 30

print('Your credentails:')

print('CLIENT\_ID: ' + CLIENT\_ID)

print('CLIENT\_SECRET:' + CLIENT\_SECRET)

Your credentails:

CLIENT\_ID: N4GIWTCJ0DPW3ST5P0EEVB25W0JHGW5ZTBJZKDKUBFTJGOK0

CLIENT\_SECRET:KRLYG4052BTLWMRWK445JLY10SHUBRLCOBRMAQLO1RRZ1PF0

In [7]:

*# Parameters*

request\_parameters = {

"client\_id": CLIENT\_ID,

"client\_secret": CLIENT\_SECRET,

"v": '20180605',

"section": "coffee",

"near": "Cleveland",

"radius": 1000,

"limit": 25}

data = requests.get("https://api.foursquare.com/v2/venues/explore", params=request\_parameters)

In [12]:

d = data.json()["response"]

d.keys()

Out[12]:

dict\_keys(['suggestedFilters', 'geocode', 'headerLocation', 'headerFullLocation', 'headerLocationGranularity', 'query', 'totalResults', 'suggestedBounds', 'groups'])

In [15]:

*# Checking cleveland city latitude and longitude and also all others values*

d["headerLocationGranularity"], d["headerLocation"], d["headerFullLocation"], d["suggestedBounds"], d["geocode"]

*# In geocode we can examine more details about the location like state, country, latitude, longitude etc.*

Out[15]:

('city',

'Cleveland',

'Cleveland',

{'ne': {'lat': 41.5041843172057, 'lng': -81.68390763895238},

'sw': {'lat': 41.49608830723969, 'lng': -81.69999756762437}},

{'what': '',

'where': 'cleveland',

'center': {'lat': 41.4995, 'lng': -81.69541},

'displayString': 'Cleveland, OH, United States',

'cc': 'US',

'geometry': {'bounds': {'ne': {'lat': 41.604436, 'lng': -81.532744},

'sw': {'lat': 41.390628, 'lng': -81.878984}}},

'slug': 'cleveland-ohio',

'longId': '72057594043078465'})

In [16]:

d["groups"][0].keys()

Out[16]:

dict\_keys(['type', 'name', 'items'])

In [17]:

d["groups"][0]["type"], d["groups"][0]["name"]

Out[17]:

('Recommended Places', 'recommended')

In [18]:

items = d["groups"][0]["items"]

print("number of items: **%i**" % len(items))

items[0]

In [23]:

*#json to dataframe and cleaning the data*

df\_raw = []

**for** item **in** items:

venue = item["venue"]

categories, id, name, location = venue["categories"], venue["id"], venue["name"], venue["location"]

print(location)

**assert** len(categories) == 1

shortname = categories[0]["shortName"]

address = ''

**if** hasattr(location, 'address'):

address = location['address']

**if** **not** "postalCode" **in** location:

**continue**

postalcode = location["postalCode"]

lat = location["lat"]

lng = location["lng"]

datarow = (id, name, shortname, address, postalcode, lat, lng)

df\_raw.append(datarow)

df = pd.DataFrame(df\_raw, columns=["id", "name", "shortname", "address", "postalcode", "lat", "lng"])

print("found **%i** cafes" % len(df))

df.head()

df.tail()

df.dtypes

df.info()

df.shape

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 24 entries, 0 to 23

Data columns (total 7 columns):

id 24 non-null object

name 24 non-null object

shortname 24 non-null object

address 24 non-null object

postalcode 24 non-null object

lat 24 non-null float64

lng 24 non-null float64

dtypes: float64(2), object(5)

memory usage: 1.4+ KB

Out[23]:

(24, 7)

In [27]:

cleveland\_coordinates = d["geocode"]["center"]

cleveland\_coordinates

Out[27]:

{'lat': 41.4995, 'lng': -81.69541}

In [31]:

*#visulaizing the data of coffee shops in cleveland*

cle\_map = folium.Map(location=[41.4995, -81.69541], zoom\_start=14)

**def** add\_markers(df):

**for** (j, row) **in** df.iterrows():

label = folium.Popup(row["name"], parse\_html=**True**)

folium.CircleMarker(

[row["lat"], row["lng"]],

radius=5,

popup=label,

color='green',

fill=**True**,

fill\_color='#3186cc',

fill\_opacity=0.7,

parse\_html=**False**).add\_to(cle\_map)

add\_markers(df)

hm\_data = df[["lat", "lng"]].to\_numpy().tolist()

cle\_map.add\_child(plugins.HeatMap(hm\_data))

cle\_map

Out[31]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [36]:

latitude = 41.4995

longitude= -81.69541

map\_cleveland = folium.Map(location=[latitude, longitude], zoom\_start=17)

add\_markers(df)

folium.CircleMarker(

[lat, lng],

radius=15,

popup="O Day Roi!",

color='purple',

fill=**True**,

fill\_color='#3186cc',

fill\_opacity=0.7,

parse\_html=**False**).add\_to(map\_cleveland)

map\_cleveland

*# We can notice here that the East 6th street near Superior avenue east is the recommended location for coffee shop.*

Out[36]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [39]:

df

Out[39]:

In [41]:

print('There are **{}** uniques categories.'.format(len(df['id'].unique())))

There are 24 uniques categories.

In [42]:

*# one hot encoding*

onehot = pd.get\_dummies(df[['name']], prefix="", prefix\_sep="")

*# add neighborhood column back to dataframe*

onehot['name'] = df['name']

*# move neighborhood column to the first column*

fixed\_columns = [onehot.columns[-1]] + list(onehot.columns[:-1])

onehot = onehot[fixed\_columns]

onehot.head()

Out[42]:

In [43]:

onehot.shape

Out[43]:

(24, 19)

In [44]:

grouped = onehot.groupby('name').mean().reset\_index()

grouped

Out[44]:

In [45]:

grouped.shape

Out[45]:

(18, 19)

In [46]:

*# set number of clusters*

kclusters = 4

clustering = grouped.drop('name', 1)

*# run k-means clustering*

kmeans = KMeans(n\_clusters=kclusters, random\_state=0).fit(clustering)

*# check cluster labels generated for each row in the dataframe*

kmeans.labels\_[0:10]

Out[46]:

array([0, 0, 0, 0, 3, 0, 0, 0, 0, 0], dtype=int32)

In [49]:

*# add clustering labels*

grouped.insert(0, 'Cluster Labels', kmeans.labels\_)

Conclusion:

*# As said before, We can notice here that the East 6th street near Superior avenue east is the recommended location for coffee shop. It is also clearly good suggetion since we can see shopping mall, library and some banks near the location where we can expect more business.*

*#Further more in future I may plan to extend this project using data about cleveland neighborhood and foursquare API data to predict more insights and perform clustering also.*