# Capstone Project - Battle of the Neighborhoods (Week 1)

---

## 1. Introduction and Business Understanding

### 1.1 Problem Description

**The business problem we are working to solve is:** The United States 2021 Presidential Inauguration Day is rapidly approaching and many people may be (irresponsibly) traveling to Washington D.C. to observe. How can we support visitors of various backgrounds and tastes, by categorizing and visualizing the cast number of D.C. neighborhoods by primary restaurants?

### 1.2 Background Discussion

With a population of nearly 690,000 people in a land mass 5.6% the size of Rhode Island, Washington D.C. can expect some of the almost 330 million U.S. citizens, and international visitors, to converge on the capitol city on January 20th. The city is used to this kind of tourism and is well-known for its diverse culinary options; 2,233 to be exact!

A huge time-sink for many travelers is taking the time to find out what kind of food you want to have, where you can have it, and who does it the best for a reasonable price. With all the apps and opinions available to the average person nowadays, it quickly becomes an overwhelming amount of data to process. Plus, nobody wants to look like a tourist!

How can we use machine learning algorithms and the power of Foursquare's location data to make this process of finding a place to eat simpler? In this project, I will be using Foursquare and clustering algorithms to analyze and group the neighborhoods of Washington D.C. by their primary restaurant types in an attempt to answer this problem.

---

## 2. Data Requirements

To pursue this project, we will need the following open source data:

A. Data that contains all the neighborhoods of Washington D.C. and their basic location information.
Source: This will be sourced from Wikipedia: https://commons.wikimedia.org/wiki/Category:Neighborhoods_in_Washington,_D.C.

B. Utilization: I will use BeautifulSoup to scrape the neighborhoods of Washington D.C. from the Wikipedia webpage and obtain their prospective latitudes and longitudes using the geocoder from GeoPy. I will then process the data and toss out any invalid data entries.

C. Data that contains all the restaurants located in each of the ~80 neighborhoods of Washington D.C.
Source: Foursquare API
Utilization: Using Foursquare, we will input the locational coordinates of each neighborhood to obtain all the venues for each area. Then, we will filter out the restaurants for each area and drop the rest. Beyond this will be the performing of the machine learning analysis.

---

# 3. Methodology

## 3.1 Data Preparation

### 3.1.1 Scraping Washington D.C. Neighborhoods List from Wikipedia

I began by opening Wikipedia up to the Neighborhoods in Washington D.C. page and utilizing the BeautifulSoup package to scrape the list into a table to create a data frame. I used Pandas to transform the tabular data into a basic data frame that merely holds the names of the neighborhoods. I also took the time here to drop all the extraneous scraped data such as underscores, parenthesis, the word “Washington D.C.”, etc. Here’s a peek at the first ten neighborhoods we scraped:

| | **neighborhood** |
|---|---|
| **1** | Adams Morgan |
| **2** | Anacostia |
| **3** | Barney Circle |
| **4** | Barry Farm |
| **5** | Benning Ridge |
| **6** | Berkley |
| **7** | Blagden Alley-Naylor Court Historic District |
| **8** | Bloomingdale |
| **9** | Brightwood |
| **10** | Brookland |

### 3.1.2 Obtaining Geographical Information Using GeoPy

The next step for our data frame is to use the geocoder class of GeoPy to obtain the coordinates of all of our neighborhoods. Here, we notice that some neighborhoods did not

return information regardless of the searching method, so we drop those entries from the data set. Our resulting data frame looks as such:

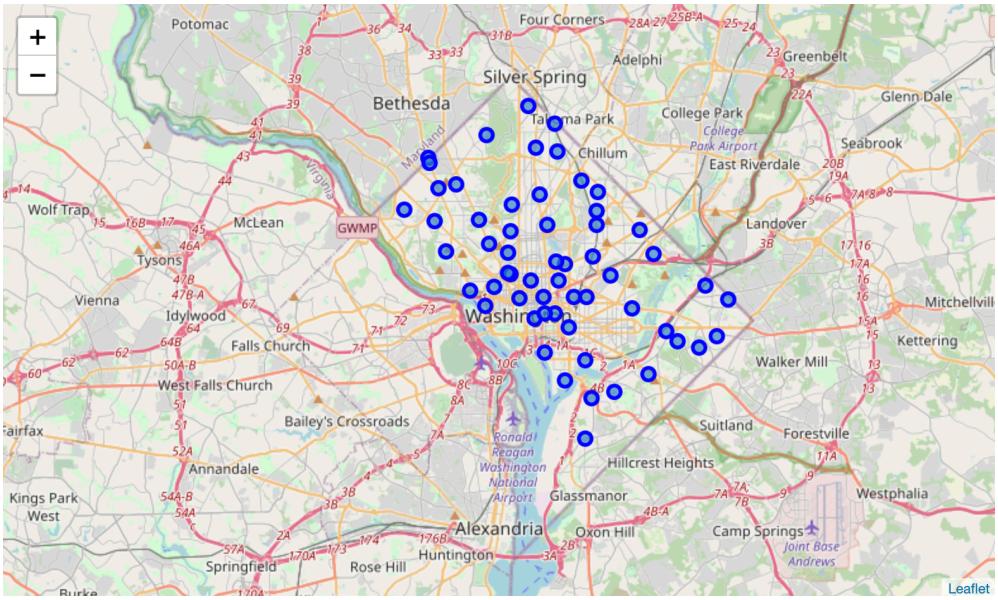
```
#Convert an address into latitude and longitude values
from geopy.geocoders import Nominatim
import warnings
warnings.filterwarnings("ignore", category=np.VisibleDeprecationWarning)
geolocator = Nominatim(user_agent="DC_explorer")

df['search']= (df['neighborhood']+", Washington, D.C.").apply(geolocator.geocode)

#A handful of neighborhoods did not return coordinates so we will drop them for the
he experiment
nan_value = float("NaN")
df.replace("None", nan_value, inplace=True)
df.dropna(subset = ["search"], inplace=True)

df['search'].apply(lambda x: (x.latitude, x.longitude))
df[['point', 'coords']] = df['search'].apply(pd.Series)
df.drop(['search'], axis=1, inplace=True)
df.reset_index(drop=True, inplace=True)
df.head()
```

| | neighborhood | point | coords |
|---|---|---|---|
| **0** | Adams Morgan | Adams Morgan, Washington, District of Columbia... | (38.9215002, -77.0421992) |
| **1** | Anacostia | Anacostia, Washington, District of Columbia, U... | (38.8625806, -76.98444095341674) |
| **2** | Barry Farm | Barry Farm, Washington, District of Columbia, ... | (38.8598039, -76.9969706) |
| **3** | Benning Ridge | Benning Ridge, Washington, District of Columbi... | (38.88135145, -76.93863030975535) |
| **4** | Bloomingdale | Bloomingdale, Washington, District of Columbia... | (38.9167782, -77.0113652) |

I then followed this up by splitting the “coords” column into “latitude” and “longitude” columns and concatenating them back onto the main data frame.

### 3.2 Exploratory Data Analysis

Performing exploratory data analysis at this stage will allow me to discover hidden relationship and properties the data holds, as well as provide helpful insights to readers and users. At this point, I used the Folium library to visualize Washington D.C. and its many neighborhoods on a map using the data extracted thus far:



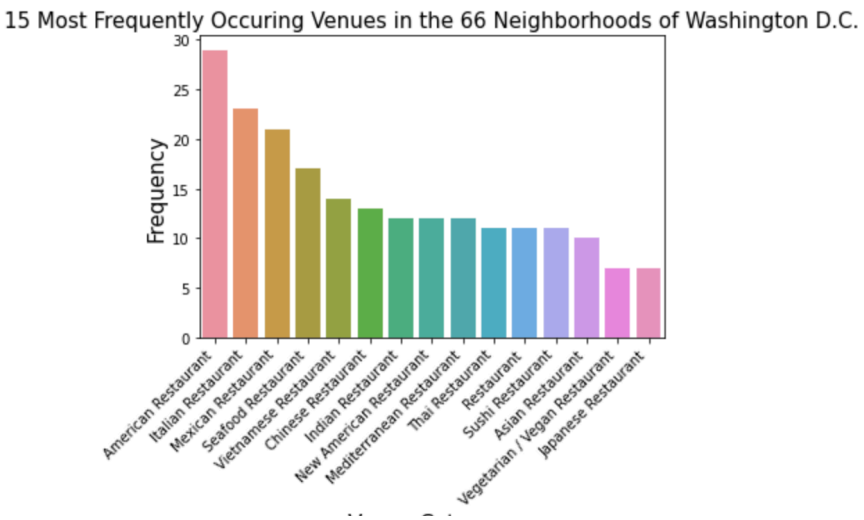
### 3.2.1 Using Foursquare API

The next step brings us to the use of Foursquare API to obtain the venues for each neighborhood in a 500 meter radius. We then create a new data frame that filters for exclusively restaurants and arrange them according to their venue category. This list indicates to us that the most common restaurant type in D.C. is the American Restaurant:

```
DC_restaurants = DC_venues[DC_venues['Venue Category'].str.contains('Restaurant')].reset_index(
DC_restaurants.index = np.arange(1, len(DC_restaurants )+1)
print (DC_restaurants['Venue Category'].value_counts())
```

```
American Restaurant                29
Italian Restaurant                 23
Mexican Restaurant                 21
Seafood Restaurant                 17
Vietnamese Restaurant              14
Chinese Restaurant                 13
Indian Restaurant                  12
New American Restaurant            12
Mediterranean Restaurant           12
Thai Restaurant                    11
Restaurant                         11
Sushi Restaurant                   11
Asian Restaurant                   10
Vegetarian / Vegan Restaurant       7
Japanese Restaurant                 7
Latin American Restaurant           6
Southern / Soul Food Restaurant     6
Fast Food Restaurant                6
Korean Restaurant                   5
Caribbean Restaurant                5
Ethiopian Restaurant                5
French Restaurant                   5
Middle Eastern Restaurant           4
Spanish Restaurant                  4
Tapas Restaurant                    4
Portuguese Restaurant               4
South American Restaurant           3
Greek Restaurant                    3
Ramen Restaurant                    3
Peruvian Restaurant                 3
Brazilian Restaurant                2
Falafel Restaurant                  2
Scandinavian Restaurant             2
Cajun / Creole Restaurant           2
Gluten-free Restaurant              2
Afghan Restaurant                   2
Eastern European Restaurant         2
Dumpling Restaurant                 1
Filipino Restaurant                 1
Israeli Restaurant                  1
Tex-Mex Restaurant                  1
Xinjiang Restaurant                 1
German Restaurant                   1
Name: Venue Category, dtype: int64
```

Next, I used the seaborn and matplotlib libraries to analyze the frequency of the top 15 venue types in the surrounding area.



This bar graph shows us the large diversity of restaurants in D.C., so there's something new and delicious for everyone's tastes!

What I am curious about is whether these restaurant types are widespread throughout D.C. or if they're concentrated in specific neighborhoods. To find out, let's analyze the leading 5 restaurant venues for each neighborhood and ignore the rest.

To do so, let's start by one-hot encoding the data frame based on the venue categories:

```
#Do one hot encoding to help find the frequency of restaurants
DC_onehot = pd.get_dummies(DC_restaurants[['Venue Category']], prefix="", prefix_sep="")

#add neighborhood column back to dataframe
DC_onehot['Neighborhood'] = DC_restaurants['Neighborhood']

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

DC_onehot.head()
```

| | Neighborhood | Afghan Restaurant | American Restaurant | Asian Restaurant | Brazilian Restaurant | Cajun / Creole Restaurant | Caribbean Restaurant | Chinese Restaurant | Dumpling Restaurant | Eastern European Restaurant |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Adams Morgan | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Adams Morgan | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | Adams Morgan | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | Adams Morgan | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | Adams Morgan | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 45 columns

Follow it up by using the groupby method on the neighborhood column and calculating the mean of the frequency of occurrence of each category. This will give us ratios of how common a restaurant is in that location, as seen below:

| | Neighborhood | Afghan Restaurant | American Restaurant | Asian Restaurant | Brazilian Restaurant | Cajun / Creole Restaurant | Caribbean Restaurant | Chinese Restaurant | Dumpling Restaurant | Easter Europea Restaurar |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Adams Morgan | 0.076923 | 0.000000 | 0.153846 | 0.076923 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.07692 |
| 1 | Anacostia | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.00000 |
| 2 | Bloomingdale | 0.000000 | 0.000000 | 0.250000 | 0.000000 | 0.000000 | 0.250000 | 0.000000 | 0.000 | 0.00000 |
| 3 | Brightwood | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.333333 | 0.000 | 0.00000 |
| 4 | Brookland | 0.000000 | 0.333333 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.00000 |
| 5 | Burrville | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.00000 |
| 6 | Cathedral Heights | 0.000000 | 0.500000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.00000 |
| 7 | Chinatown | 0.000000 | 0.000000 | 0.071429 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.00000 |
| 8 | Cleveland Park | 0.000000 | 0.000000 | 0.125000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000 | 0.00000 |

The next step I followed was to arrange and display the 15 most common restaurants in each neighborhood based on the above one-hot data frame. Here's how the head of it came out:

| | 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 | Adams Morgan | Asian Restaurant | Mediterranean Restaurant | Japanese Restaurant | New American Restaurant | Brazilian Restaurant | Eastern European Restaurant | Empanada Restaurant | Falafel Restaurant | Italian Restaurant |
| 1 | Anacostia | American Restaurant | Xinjiang Restaurant | Falafel Restaurant | Israeli Restaurant | Indian Restaurant | Greek Restaurant | Gluten-free Restaurant | German Restaurant | French Restaurant |
| 2 | Bloomingdale | Asian Restaurant | Mexican Restaurant | Caribbean Restaurant | Italian Restaurant | Xinjiang Restaurant | Fast Food Restaurant | Indian Restaurant | Greek Restaurant | Gluten-free Restaurant |
| 3 | Brightwood | Mexican Restaurant | Chinese Restaurant | Southern / Soul Food Restaurant | Xinjiang Restaurant | Falafel Restaurant | Indian Restaurant | Greek Restaurant | Gluten-free Restaurant | German Restaurant |
| 4 | Brookland | American Restaurant | Ethiopian Restaurant | Mexican Restaurant | Indian Restaurant | Vietnamese Restaurant | Cajun / Creole Restaurant | Caribbean Restaurant | Israeli Restaurant | Asian Restaurant |

With this dataframe, I can now use KMeans clustering in an attempt to group the neighborhoods according to similar primary venues. This will allow me to perform prescriptive analysis to help a user identify which neighborhood area is ideal for them to visit based on their food preferences. Included below is my code:

```
kclusters = 6

DC_grouped_clustering = DC_grouped.drop('Neighborhood', 1)
kmeans = KMeans(n_clusters = kclusters, random_state = 0).fit(DC_grouped_clustering)

kmeans.labels_[0:10]
```

```
array([3, 1, 3, 3, 1, 0, 1, 4, 3, 1], dtype=int32)
```

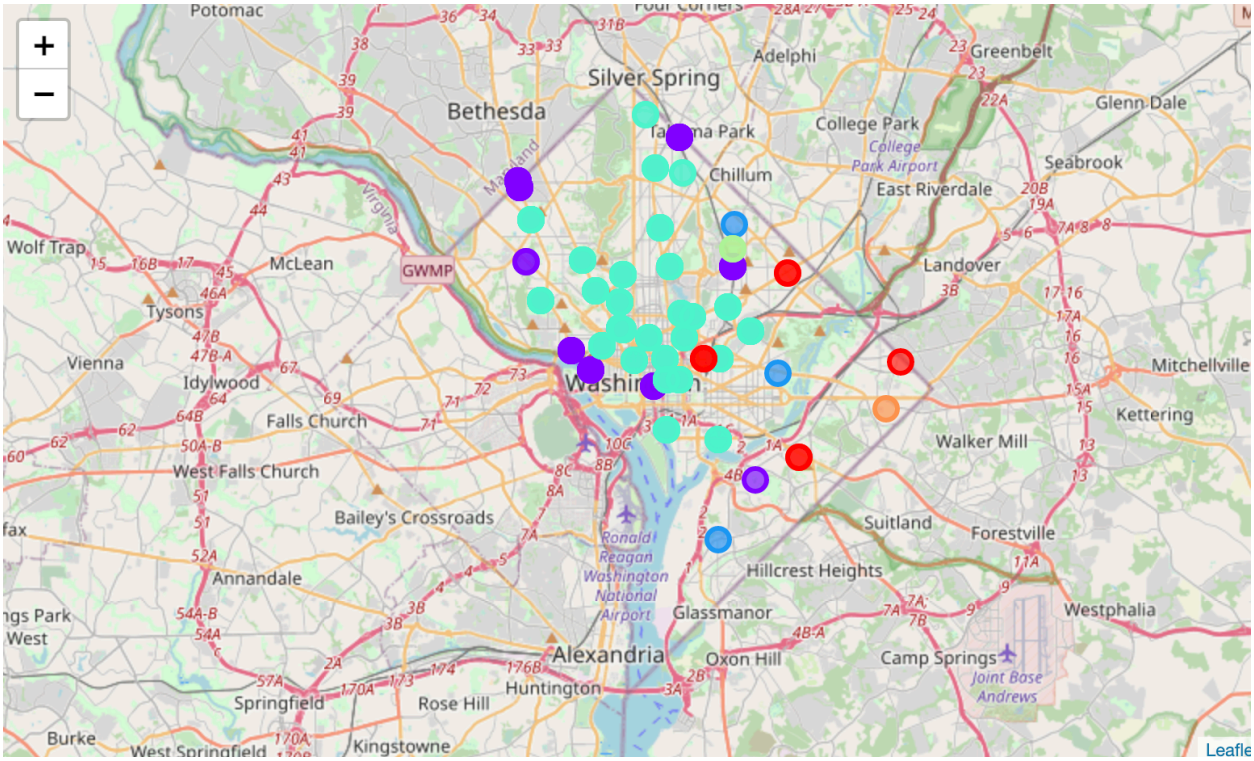
```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

DC_merged = DC_restaurants
DC_merged = DC_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborh
DC_merged.drop('Venue Category',axis = 1, inplace = True)
```

```
DC_merged.head()
```

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Mos Commo Venu |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Adams Morgan | 38.9215 | -77.042199 | Lapis | 38.921302 | -77.043890 | 3 | Asian Restaurant | Mediterranean Restaurant | Japanes Restaurar |
| 2 | Adams Morgan | 38.9215 | -77.042199 | Donburi | 38.921673 | -77.042385 | 3 | Asian Restaurant | Mediterranean Restaurant | Japanes Restaurar |
| 3 | Adams Morgan | 38.9215 | -77.042199 | Tail Up Goat | 38.923522 | -77.043099 | 3 | Asian Restaurant | Mediterranean Restaurant | Japanes Restaurar |
| 4 | Adams Morgan | 38.9215 | -77.042199 | Amsterdam Falafelshop | 38.921162 | -77.041959 | 3 | Asian Restaurant | Mediterranean Restaurant | Japanes Restaurar |
| 5 | Adams Morgan | 38.9215 | -77.042199 | Mintwood Place | 38.922053 | -77.043611 | 3 | Asian Restaurant | Mediterranean Restaurant | Japanes Restaurar |

5 rows × 22 columns

Using the Folium library again, I can overlay my clusters on the map of D.C. to identify which areas share similar primary restaurant categories:




## 4. Results and Discussion

Our analysis has yielded a number of insights into how Washington D.C.'s restaurant venues are distributed by cuisine type. This would prove very helpful for people traveling to D.C. for the inauguration, or even people with business ideas hoping to open a new restaurant in a location where their offered cuisine is less common.

We found that:

- American restaurants are the most common restaurants out of all categories in the neighborhoods, with most other types falling around a lower, yet common frequency.
- Many of the central neighborhoods, where downtown D.C. lies, share the same cluster #3, meaning that they all offer very similar dining options.
- As we move away from downtown D.C., the clusters begin to diverge further, indicating that culinary options expand as you move into the border neighborhoods.
- With cluster #3 encompassing the most neighborhoods, it it worth mentioning that although American restaurants are the most common overall, cluster #3 doesn't even have American restaurants in the top 5 most common venues. The top 5 in cluster #3 are Asian,

Mediterranean, New American, and a wider spread for the other two most common. American restaurants fall into the 8th most common venue for this cluster.

Though this clustering is based entirely on the data that we have obtained from Foursquare, we have to recognize that we have not taken all the factors into consideration. We haven't taken into account the price range of the restaurants, how accessible they are to public transportation, their ratings, or their reviews, for that matter. This being said, our analysis does offer a broad overview into the restaurant locality and how the neighborhoods' restaurants are distributed over the area of D.C.

---

## 5. Conclusion and Closing

I utilized libraries that are fairly common, including the web-scraper BeautifulSoup, the geolocation library GeoPy, the location data provider Foursquare, and the Folium package to visualize the results of our analysis over a real map of Washington D.C. Also included in my analysis were discussions about how else this information could be leveraged for entrepreneurs, the shortcomings of our methodology, and how future studies could be improved.

When there are so many restaurants and so many apps and opinions out on the internet, it quickly becomes overwhelming to do something as simple as finding a place to eat. Luckily, with the power of data, we can find innovative solutions to these problems and tailor the answers to each customer. In this study, we used machine learning to cluster the 66 neighborhoods of Washington D.C. according to their top 5 most common restaurants for the purpose of guiding 2021 inauguration-goers, or really any travelers, to areas where their tastes can be satisfied.