```
import numpy as np
import sklearn.cluster
```

```
data = np.arange(0,100)
```

```
data = list(zip(data, data))
```

```
from sklearn.cluster import KMeans
```

```
model = KMeans(n_clusters=3, init='random', max_iter=50)
```

```
model.fit(data)
```

```
     KMeans(algorithm='auto', copy_x=True, init='random', max_iter=50, n_clusters=3,
            n_init=10, n_jobs=None, precompute_distances='auto', random_state=None,
            tol=0.0001, verbose=0)
```

```
model.cluster_centers_
```

```
     array([[50. , 50. ],
            [83. , 83. ],
            [16.5, 16.5]])
```

Importing required libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

Creating dataframes

```
DATA_FOLDER = '/content/drive/MyDrive/data/uber_rideshare/'
```

```
apr14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-apr14.csv')
may14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-may14.csv')
```

```
may14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-may14.csv')
jun14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-jun14.csv')
jul14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-jul14.csv')
aug14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-aug14.csv')
sep14 = pd.read_csv(DATA_FOLDER+'uber-raw-data-sep14.csv')
```

```
merged_df = pd.concat([apr14, may14, jun14, jul14, aug14, sep14])
merged_df
```

```
                  Date/Time      Lat       Lon     Base
      0       4/1/2014 0:11:00  40.7690  -73.9549  B02512
      1       4/1/2014 0:17:00  40.7267  -74.0345  B02512
      2       4/1/2014 0:21:00  40.7316  -73.9873  B02512
      3       4/1/2014 0:28:00  40.7588  -73.9776  B02512
      4       4/1/2014 0:33:00  40.7594  -73.9722  B02512
     ...                   ...      ...       ...     ...
1028131  9/30/2014 22:57:00  40.7668  -73.9845  B02764
1028132  9/30/2014 22:57:00  40.6911  -74.1773  B02764
1028133  9/30/2014 22:58:00  40.8519  -73.9319  B02764
1028134  9/30/2014 22:58:00  40.7081  -74.0066  B02764
1028135  9/30/2014 22:58:00  40.7140  -73.9496  B02764

4534327 rows × 4 columns
```

String to datetime conversion

```
apr14['Date/Time'] = pd.to_datetime(apr14['Date/Time'], format='%m/%d/%Y %H:%M:%S')
may14['Date/Time'] = pd.to_datetime(may14['Date/Time'], format='%m/%d/%Y %H:%M:%S')
jun14['Date/Time'] = pd.to_datetime(jun14['Date/Time'], format='%m/%d/%Y %H:%M:%S')
jul14['Date/Time'] = pd.to_datetime(jul14['Date/Time'], format='%m/%d/%Y %H:%M:%S')
aug14['Date/Time'] = pd.to_datetime(aug14['Date/Time'], format='%m/%d/%Y %H:%M:%S')
sep14['Date/Time'] = pd.to_datetime(sep14['Date/Time'], format='%m/%d/%Y %H:%M:%S')
merged_df['Date/Time'] = pd.to_datetime(merged_df['Date/Time'], format='%m/%d/%Y %H:%M:%S')
```

```
dfs = [apr14, may14, jun14, jul14, aug14, sep14, merged_df]
current_df = dfs[0]
```

Rideshare histogram

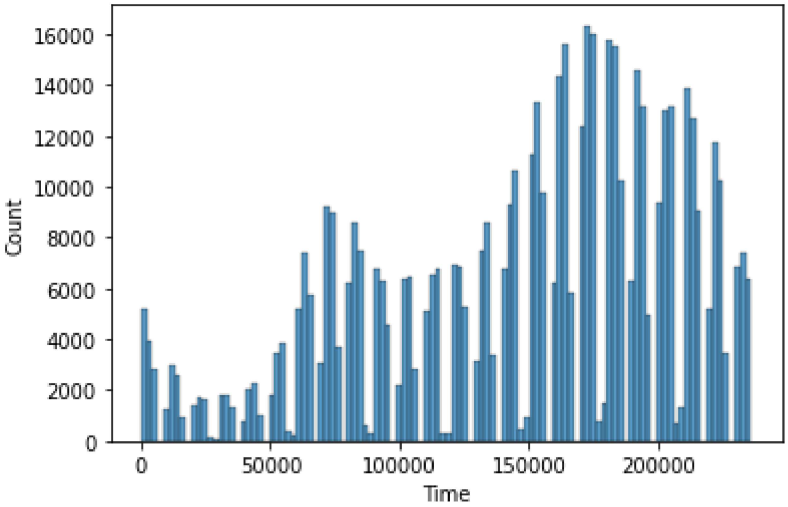
```
current_df['Time'] = current_df['Date/Time'].dt.time.apply(lambda x: int(x.strftime('%H%M%S')
current_df
```

	Date/Time	Lat	Lon	Base	Time
0	2014-04-01 00:11:00	40.7690	-73.9549	B02512	1100
1	2014-04-01 00:17:00	40.7267	-74.0345	B02512	1700
2	2014-04-01 00:21:00	40.7316	-73.9873	B02512	2100
3	2014-04-01 00:28:00	40.7588	-73.9776	B02512	2800
4	2014-04-01 00:33:00	40.7594	-73.9722	B02512	3300
...	...	...	...	...	...
564511	2014-04-30 23:22:00	40.7640	-73.9744	B02764	232200
564512	2014-04-30 23:26:00	40.7629	-73.9672	B02764	232600
564513	2014-04-30 23:31:00	40.7443	-73.9889	B02764	233100
564514	2014-04-30 23:32:00	40.6756	-73.9405	B02764	233200
564515	2014-04-30 23:48:00	40.6880	-73.9608	B02764	234800

564516 rows × 5 columns

```
sns.histplot(current_df['Time'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f64cdc938d0>
```



Filtering morning and evening rides

```
morning_df_idx = (current_df['Time'] > 50000) & (current_df['Time'] < 110000)
morning_df = current_df[morning_df_idx]
```

```
evening_df_idx = (current_df['Time'] > 150000) & (current_df['Time'] < 220000)
evening_df = current_df[evening_df_idx]
```

```
morning_df
```

```
                   Date/Time      Lat       Lon    Base    Time
30       2014-04-01 05:08:00  40.7141  -74.0094  B02512   50800
31       2014-04-01 05:12:00  40.7893  -73.9709  B02512   51200
32       2014-04-01 05:18:00  40.7747  -73.9910  B02512   51800
33       2014-04-01 05:19:00  40.7689  -73.9876  B02512   51900
34       2014-04-01 05:23:00  40.7744  -74.0149  B02512   52300
...                      ...      ...       ...     ...     ...
564028   2014-04-30 10:55:00  40.7665  -73.9514  B02764  105500
564029   2014-04-30 10:55:00  40.7266  -73.9076  B02764  105500
564030   2014-04-30 10:56:00  40.7365  -73.9816  B02764  105600
564031   2014-04-30 10:57:00  40.7710  -73.8659  B02764  105700
564032   2014-04-30 10:58:00  40.7300  -73.9867  B02764  105800

111422 rows × 5 columns
```

```
evening_df
```

```
                        Date/Time      Lat       Lon     Base    Time
```

```
morning_coordinates = morning_df[['Lat','Lon']].sample(10000,random_state = 10).values
evening_coordinates = evening_df[['Lat','Lon']].sample(10000,random_state = 10).values
```

```
   499   2014-04-01 15:03:00   40.7079   -74.0093   B02512   150300
```

Installing folium

for plotting coordinates

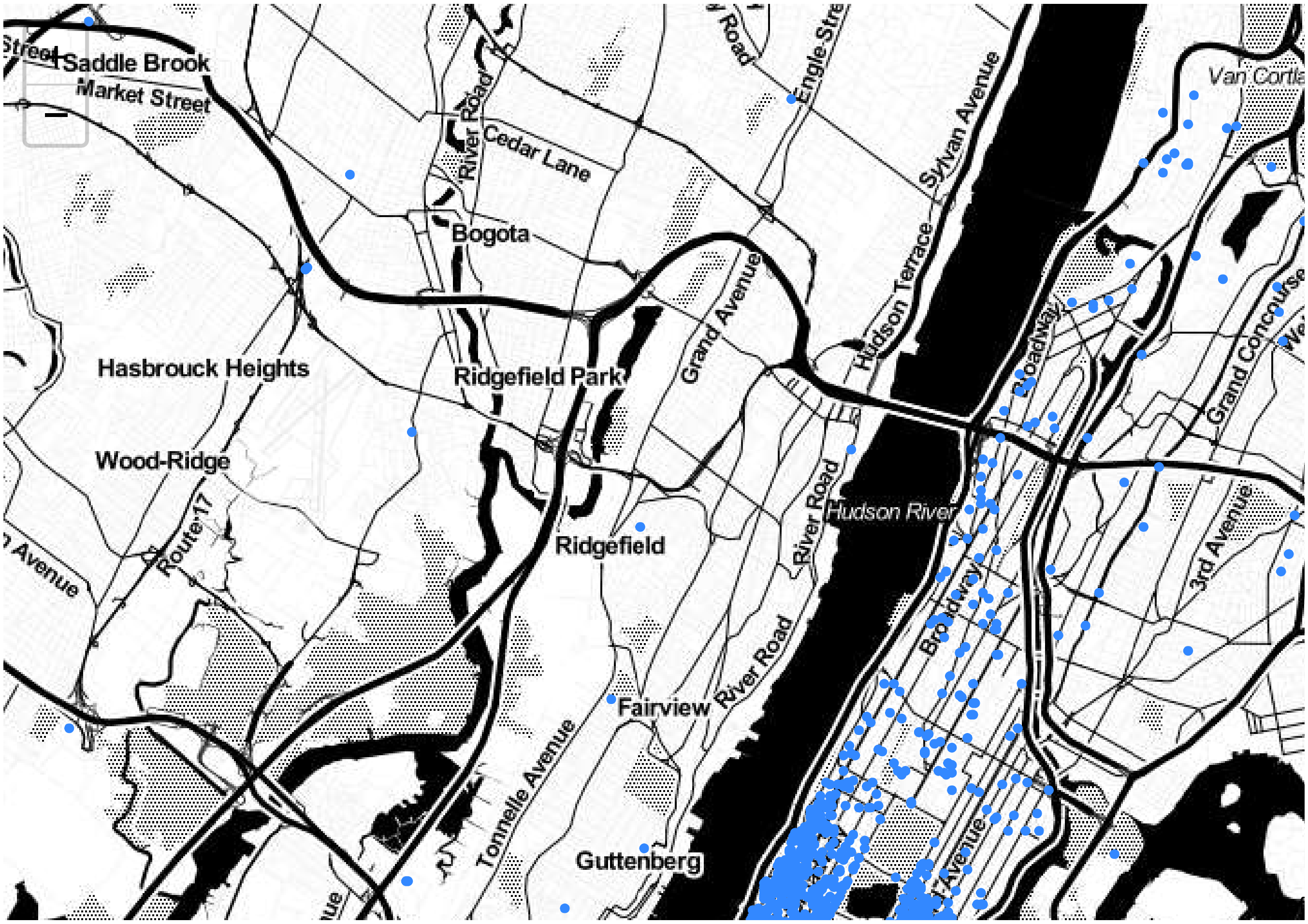
```
!pip install folium
```

```
    Requirement already satisfied: folium in /usr/local/lib/python3.7/dist-packages (0.8.3)
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from fol
    Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from foliu
    Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from fo
    Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from 
    Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-package
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
    Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
```

```
import folium
```

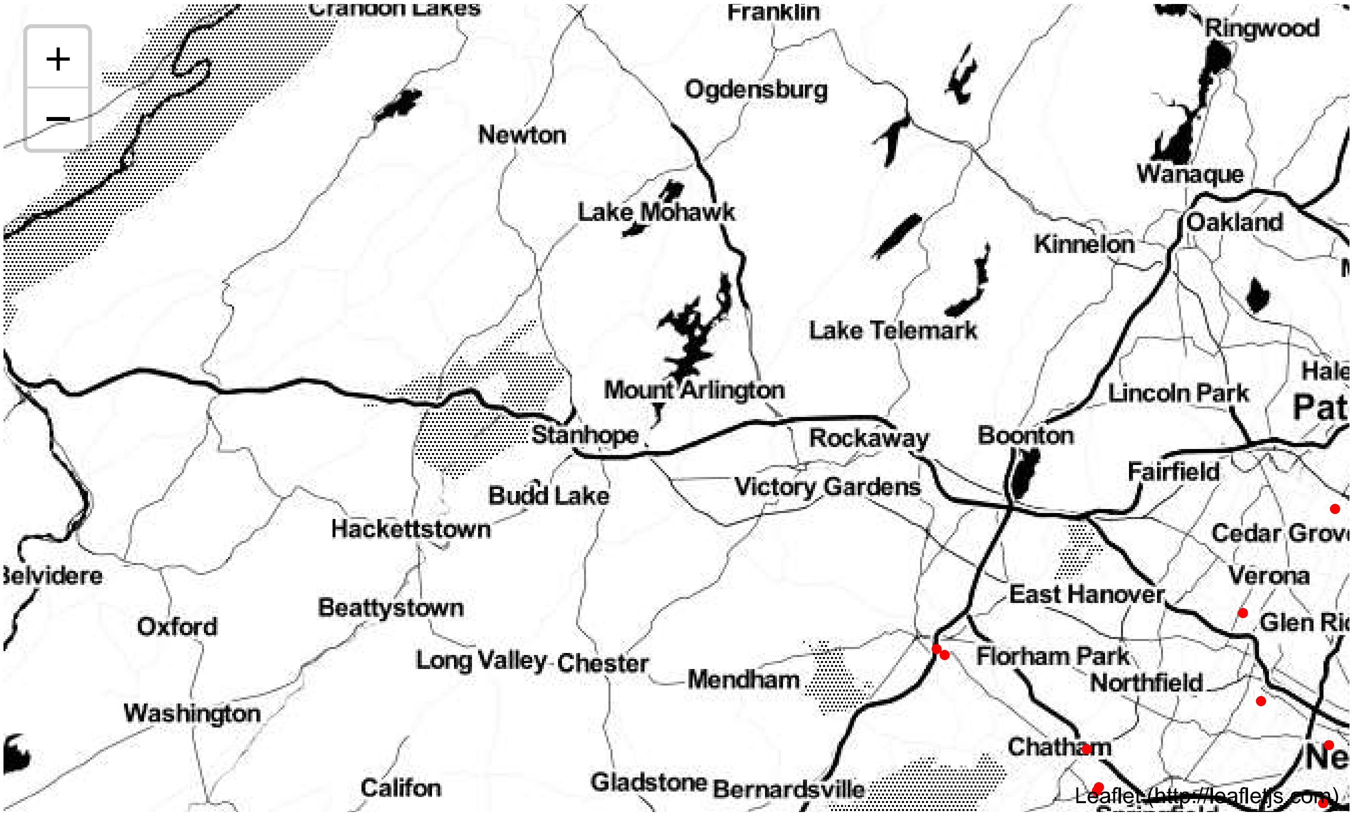
Plotting morning rides on map

```
morning_map = folium.Map(location=[40.79658011772687, -73.87341741832425], zoom_start = 12, t
for coordinate in morning_coordinates:
  folium.CircleMarker(radius=1,location=coordinate,fill=True).add_to(morning_map)
morning_map
```



Plotting evening rides on map

```
evening_map = folium.Map(location=[40.79658011772687, -73.87341741832425], zoom_start = 12, t
for coordinate in evening_coordinates:
  folium.CircleMarker(radius=1,location=coordinate,color="#FF0000",fill=True).add_to(evening_
evening_map
```



Importing KMeans

```
from sklearn.cluster import KMeans
import numpy as np
```

Finding clusters

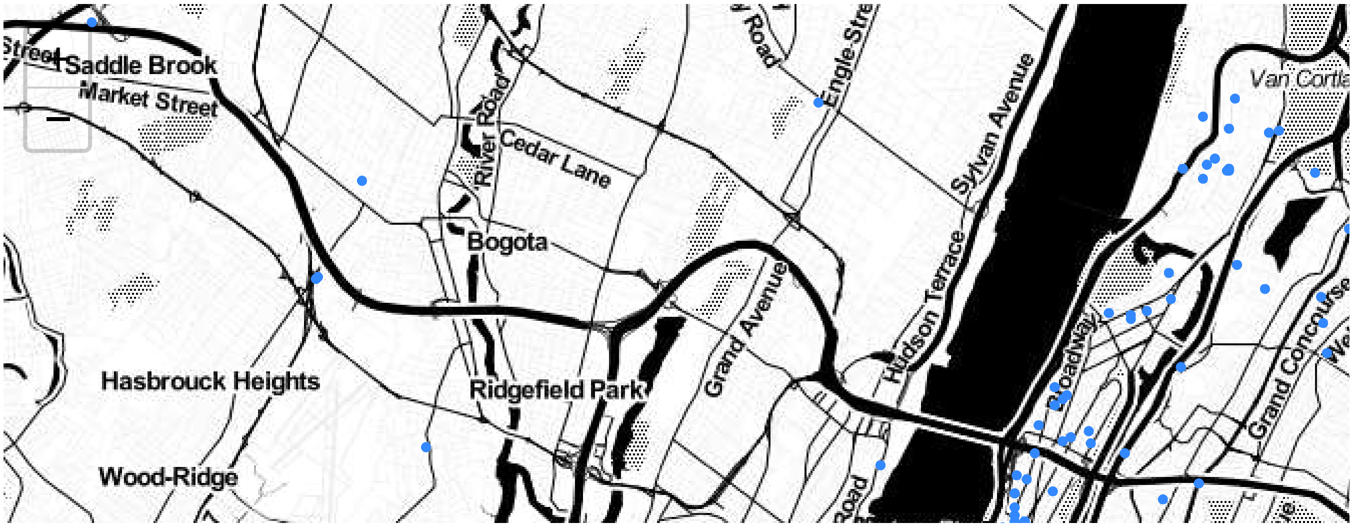
```
n_clusters = 6
model = KMeans(n_clusters=n_clusters, init='random', max_iter=300)
model.fit(morning_df[['Lat', 'Lon']])
model.fit(morning_df[['Lat', 'Lon']])
```

```
KMeans(algorithm='auto', copy_x=True, init='random', max_iter=300, n_clusters=6,
       n_init=10, n_jobs=None, precompute_distances='auto', random_state=None,
       tol=0.0001, verbose=0)
```

```
morning_centroids = model.cluster_centers_
morning_centroids
```

```
array([[ 40.69828782, -74.20574989],
       [ 40.68791073, -73.96539961],
       [ 40.7734237 , -73.96738427],
       [ 40.79314572, -73.86491743],
       [ 40.73297674, -73.99562057],
       [ 40.66314606, -73.77362212]])
```

```
for i, coordinate in enumerate(morning_centroids):
    folium.Marker(coordinate, popup='Centroid {}'.format(i+1), icon=folium.Icon(color='red'))
morning_map
```



for evening

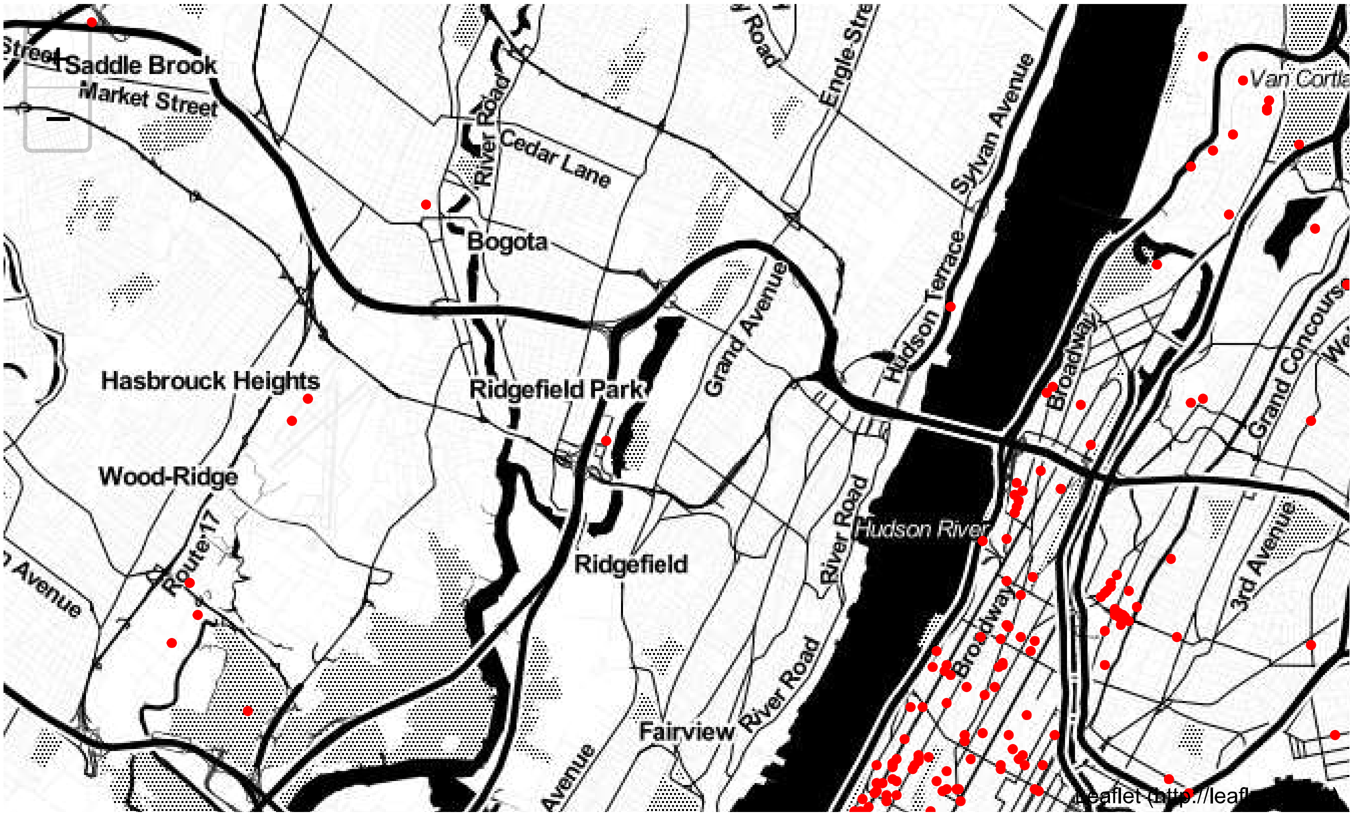
```
n_clusters = 6
model = KMeans(n_clusters=n_clusters, init='random', max_iter=300)
model.fit(evening_df[['Lat','Lon']])
```

```
KMeans(algorithm='auto', copy_x=True, init='random', max_iter=300, n_clusters=6,
       n_init=10, n_jobs=None, precompute_distances='auto', random_state=None,
       tol=0.0001, verbose=0)
```

```
evening_centroids = model.cluster_centers_
evening_centroids
```

```
array([[ 40.79451631, -73.86947377],
       [ 40.73072445, -73.99941624],
       [ 40.69961741, -74.20066416],
       [ 40.68838102, -73.9681331 ],
       [ 40.65714118, -73.7743413 ],
       [ 40.76291046, -73.97455432]])
```

```
for i, coordinate in enumerate(evening_centroids):
    folium.Marker(coordinate, popup='Centroid {}'.format(i+1), icon=folium.Icon(color='blue')
evening_map
```



Finding clusters in whole selected dataframe

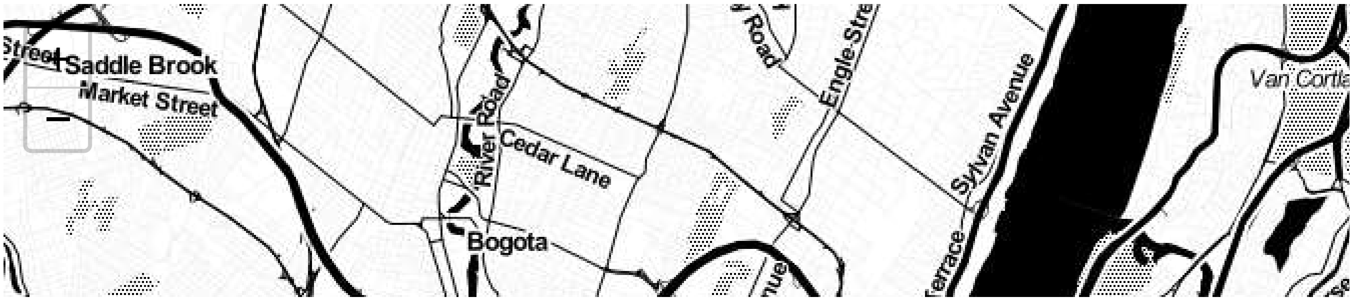
```
n_clusters = 8
model = KMeans(n_clusters=n_clusters, init='random', max_iter=300)
model.fit(current_df[['Lat','Lon']])
```

```
    KMeans(algorithm='auto', copy_x=True, init='random', max_iter=300, n_clusters=8,
           n_init=10, n_jobs=None, precompute_distances='auto', random_state=None,
           tol=0.0001, verbose=0)
```

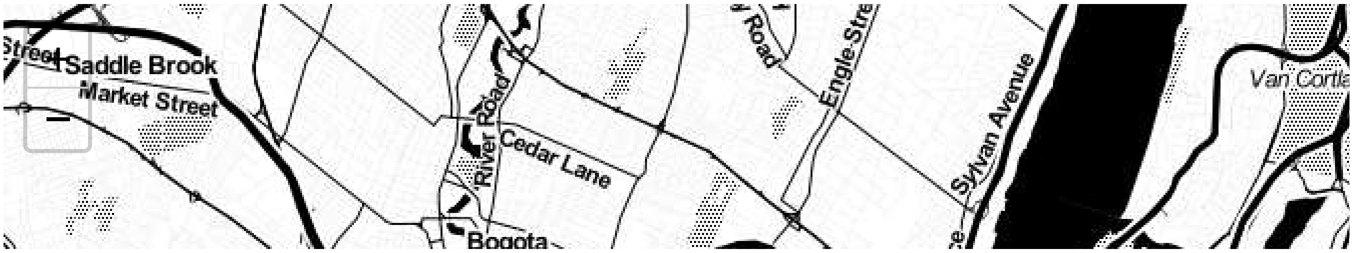
```
centroids = model.cluster_centers_
centroids
```

```
     array([[ 40.68792472, -73.96466082],
            [ 40.78151003, -73.87035803],
            [ 40.78190196, -73.95900665],
            [ 40.70056661, -74.20165533],
            [ 40.72758487, -74.00039024],
            [ 40.65588366, -73.77949539],
            [ 40.97239631, -73.61628852],
            [ 40.75570984, -73.98143572]])
```

```
map = folium.Map(location=[40.79658011772687, -73.87341741832425], zoom_start = 12, tiles='St
for i, coordinate in enumerate(centroids):
    folium.Marker(coordinate, popup='Centroid {}'.format(i+1), icon=folium.Icon(color='blue')
map
```



```
new_ride = (40.70647056912189, -73.91116590442799)
folium.Marker(new_ride, popup='New Rider', icon=folium.Icon(color='green')).add_to(map)
map
```

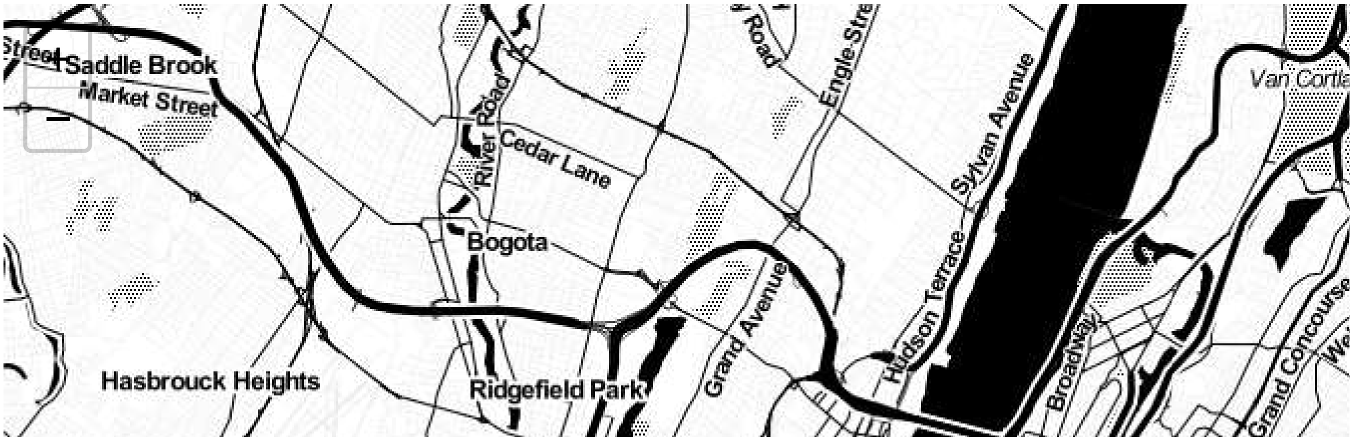


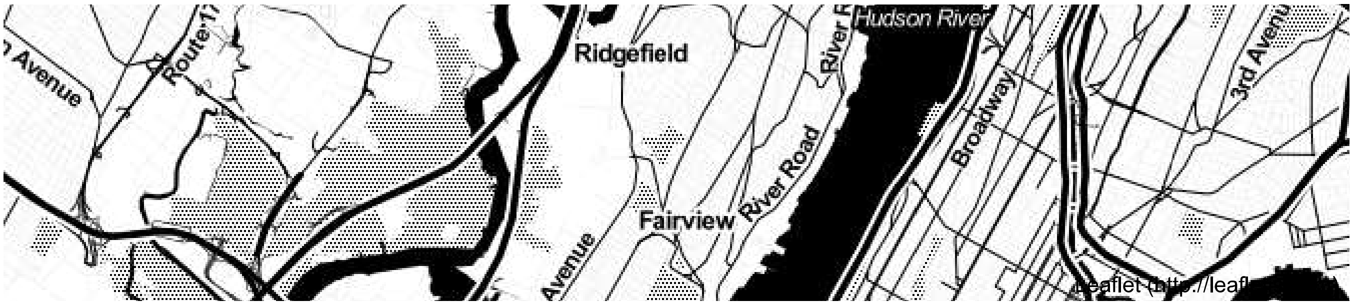
```
centroid_idx = model.predict([new_ride])
```

```
centroids[centroid_idx]
```

```
array([[ 40.68792472, -73.96466082]])
```

```
folium.Marker(centroids[centroid_idx][0], icon=folium.Icon(color='yellow')).add_to(map)
map
```


Street
Saddle Brook
Market Street
River Road
Cedar Lane
Bogota
Hasbrouck Heights
Ridgefield Park
Grand Avenue
Engle Stre
Hudson Terrace
Sylvan Avenue
Broadway
Van Cortla
Grand Concourse
We


Avenue
Route 17
Ridgefield
Fairview
River Road
River R
Hudson River
Broadway
3rd Avenu
Avenue
Leaflet (http://leafl