



## Chapter 8 - exercise 2

### Map: Mexico

1. Tạo biểu đồ có center là Mexico (location= [23.6345, -102.5528]) với zoom level là 4.
2. Tạo Mapbox Bright Map với center là Mexico with zoom level 6.3.

### Map với Marker

**San Francisco Police Department Incidents năm 2016 - được cung cấp từ cộng đồng tin dữ liệu công cộng San Francisco. Các sự cố bắt nguồn từ hệ thống báo cáo sự cố tội phạm của Sở cảnh sát San Francisco (SFPD). Được cập nhật hàng ngày, hiển thị dữ liệu cho cả năm 2016. Địa chỉ và vị trí đã được ẩn danh bằng cách di chuyển đến giữa khối (mid-block) hoặc đến một giao lộ (intersection)**

1. Đọc dữ liệu Police\_Department\_Incidents\_-PreviousYear\_\_2016\_.csv và gán vào df\_incidents, tìm hiểu về dữ liệu với: info, head, shape
2. Làm sạch dữ liệu: có đến 150.500 tội phạm, diễn ra vào năm 2016. Tạo bộ dữ liệu mới chỉ lấy 100 tội phạm đầu tiên trong bộ dữ liệu này. => in shape
3. Tạo biểu đồ có center là San Francisco (location= [37.77, -122.42]) với zoom level là 12.
4. Đưa 100 điểm tội phạm lên bản đồ (với marker)
5. Thêm pop-up text sẽ được hiển thị thông tin Category khi người dùng di chuyển chuột qua chuột qua marker.
6. Để cho bản đồ khỏi rối, bỏ đi các location marker và chỉ thêm text vào từng circle marker
7. Nhóm các markers vào các cluster. Mỗi cluster sẽ hiển thị số lượng các tội phạm trong mỗi neighborhood. Những cụm này có thể được coi là từng nhóm của San Francisco để phân tích riêng sau này. (Hãy sử dụng MarkerCluster object và thêm tất cả các data point trong dataframe vào object này.

```
In [7]: import folium
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

### Map

```
In [8]: ### type your answer here
mexico_latitude = 23.6345
mexico_longitude = -102.5528

mexico_map = folium.Map(location=[mexico_latitude, mexico_longitude], zoom_start=4
mexico_map
```

Out[8]:



```
In [9]: ### Create a map of Mexico to visualize its hill shading and natural vegetation. U
mexico_map = folium.Map(location=[mexico_latitude, mexico_longitude], zoom_start=4
mexico_map
```

Out[9]:



# Map với Marker

**San Francisco Police Department Incidents năm 2016 - được cung cấp từ cổng thông tin dữ liệu công cộng San Francisco. Các sự cố bắt nguồn từ hệ thống báo cáo sự cố tội phạm của Sở cảnh sát San Francisco (SFPD). Được cập nhật hàng ngày, hiển thị dữ liệu cho cả năm 2016. Địa chỉ và vị trí đã được ẩn danh bằng cách di chuyển đến giữa khối (mid-block) hoặc đến một giao lộ (intersection)**

1. Đọc dữ liệu `Police_Department_Incidents_-_PreviousYear__2016_.csv` và gán vào `df_incidents`, tìm hiểu về dữ liệu với: `info`, `head`, `shape`
2. Làm sạch dữ liệu: có đến 150.500 tội phạm, diễn ra vào năm 2016. Tạo bộ dữ liệu mới chỉ lấy 100 tội phạm đầu tiên trong bộ dữ liệu này. => `in shape`
3. Tạo biểu đồ có center là San Francisco (`location= [37.77, -122.42]`) với zoom level là 12.
4. Đưa 100 điểm tội phạm lên bản đồ (với marker)
5. Thêm pop-up text sẽ được hiển thị thông tin Category khi người dùng di chuyển chuột qua chuột qua marker.
6. Để cho bản đồ khỏi rối, bỏ đi các location marker và chỉ thêm text vào từng circle marker
7. Nhóm các markers vào các cluster. Mỗi cluster sẽ hiển thị số lượng các tội phạm trong mỗi neighborhood. Những cụm này có thể được coi là từng nhóm của San Francisco để phân tích riêng sau này. (Hãy sử dụng MarkerCluster object và thêm tất cả các data point trong dataframe vào object này.

```
In [10]: # read dataset
df_incidents = pd.read_csv('data/Police_Department_Incidents_-_Previous_Year__2016
df_incidents.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150500 entries, 0 to 150499
Data columns (total 13 columns):
IncidntNum    150500 non-null int64
Category      150500 non-null object
Descript      150500 non-null object
DayOfWeek     150500 non-null object
Date          150500 non-null object
Time          150500 non-null object
PdDistrict    150499 non-null object
Resolution    150500 non-null object
Address       150500 non-null object
X             150500 non-null float64
Y             150500 non-null float64
Location      150500 non-null object
PdId          150500 non-null int64
dtypes: float64(2), int64(2), object(9)
memory usage: 14.9+ MB
```

So each row consists of 13 features:



1. **IncidentNum**: Incident Number
2. **Category**: Category of crime or incident
3. **Descript**: Description of the crime or incident
4. **DayOfWeek**: The day of week on which the incident occurred
5. **Date**: The Date on which the incident occurred
6. **Time**: The time of day on which the incident occurred
7. **PdDistrict**: The police department district
8. **Resolution**: The resolution of the crime in terms whether the perpetrator was arrested or not
9. **Address**: The closest address to where the incident took place
10. **X**: The longitude value of the crime location
11. **Y**: The latitude value of the crime location
12. **Location**: A tuple of the latitude and the longitude values
13. **PdId**: The police department ID

In [11]: `df_incidents.head()`

Out[11]:

	IncidentNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolution
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	ARREST, BOOKED
1	120058272	WEAPON LAWS	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	ARREST, BOOKED
2	141059263	WARRANTS	WARRANT ARREST	Monday	04/25/2016 12:00:00 AM	14:59	BAYVIEW	ARREST, BOOKED
3	160013662	NON-CRIMINAL	LOST PROPERTY	Tuesday	01/05/2016 12:00:00 AM	23:50	TENDERLOIN	NONE
4	160002740	NON-CRIMINAL	LOST PROPERTY	Friday	01/01/2016 12:00:00 AM	00:30	MISSION	NONE

In [12]: `df_incidents.shape`

Out[12]: (150500, 13)

In [13]: `# get the first 100 crimes in the df_incidents dataframe  
limit = 100  
df_incidents = df_incidents.iloc[0:limit, :]`

In [14]: `df_incidents.shape`

Out[14]: (100, 13)

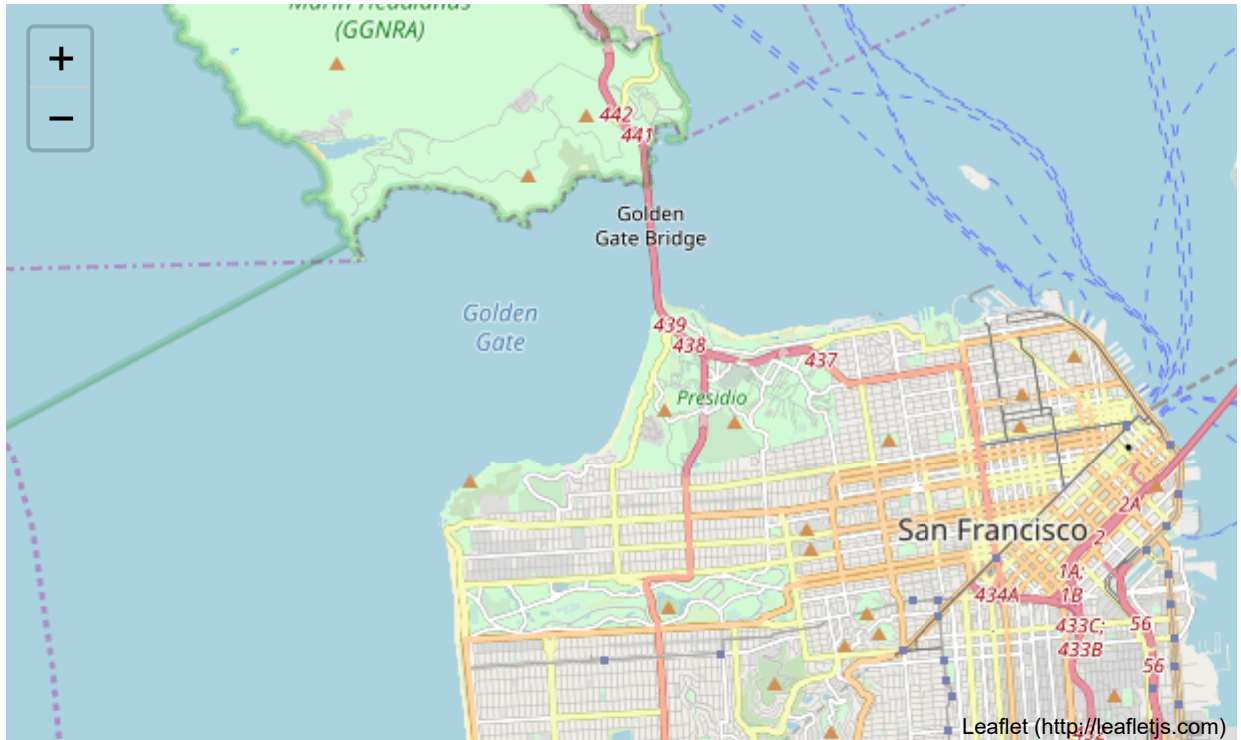
Now that we reduced the data a little bit, let's visualize where these crimes took place in the city of San Francisco. We will use the default style and we will initialize the zoom level to 12.

```
In [16]: # San Francisco Latitude and Longitude values
latitude = 37.77
longitude = -122.42
```

```
In [17]: # create map and display it
sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)

# display the map of San Francisco
sanfran_map
```

Out[17]:



```

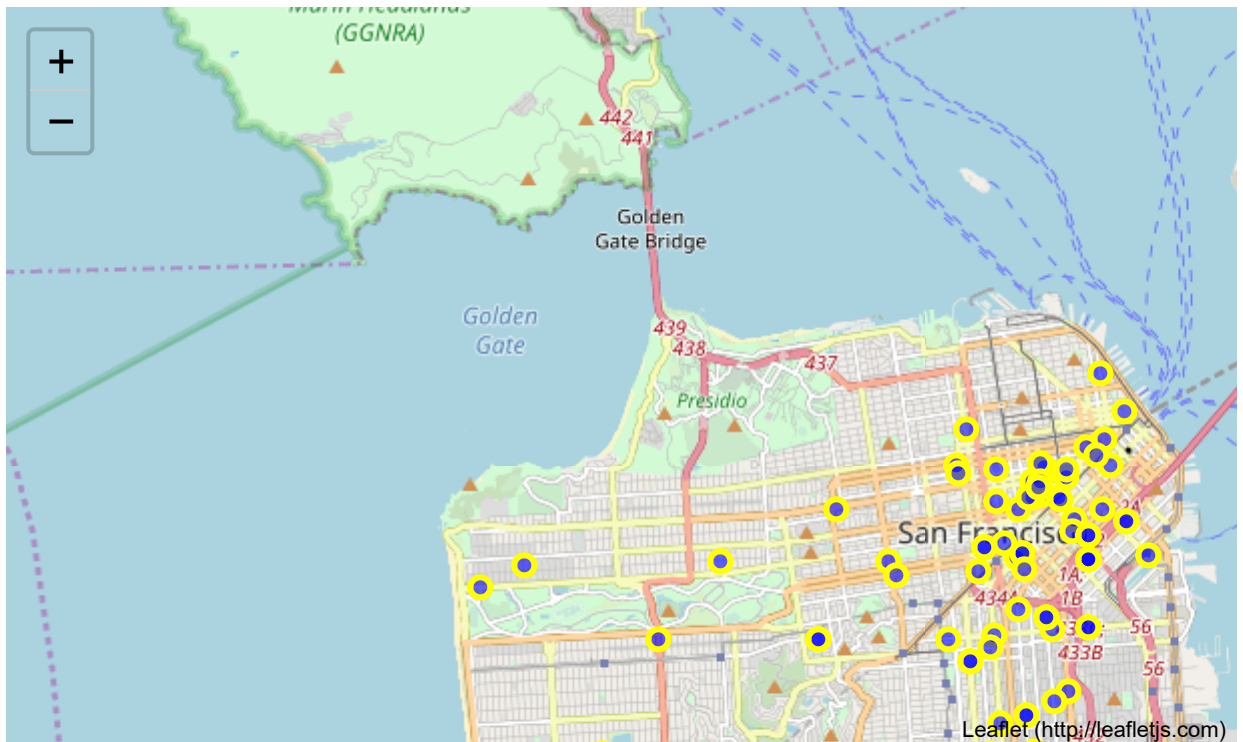
In [18]: # instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# Loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.features.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

# add incidents to map
sanfran_map.add_child(incidents)

```

Out[18]:



```

In [19]: # instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# Loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.features.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

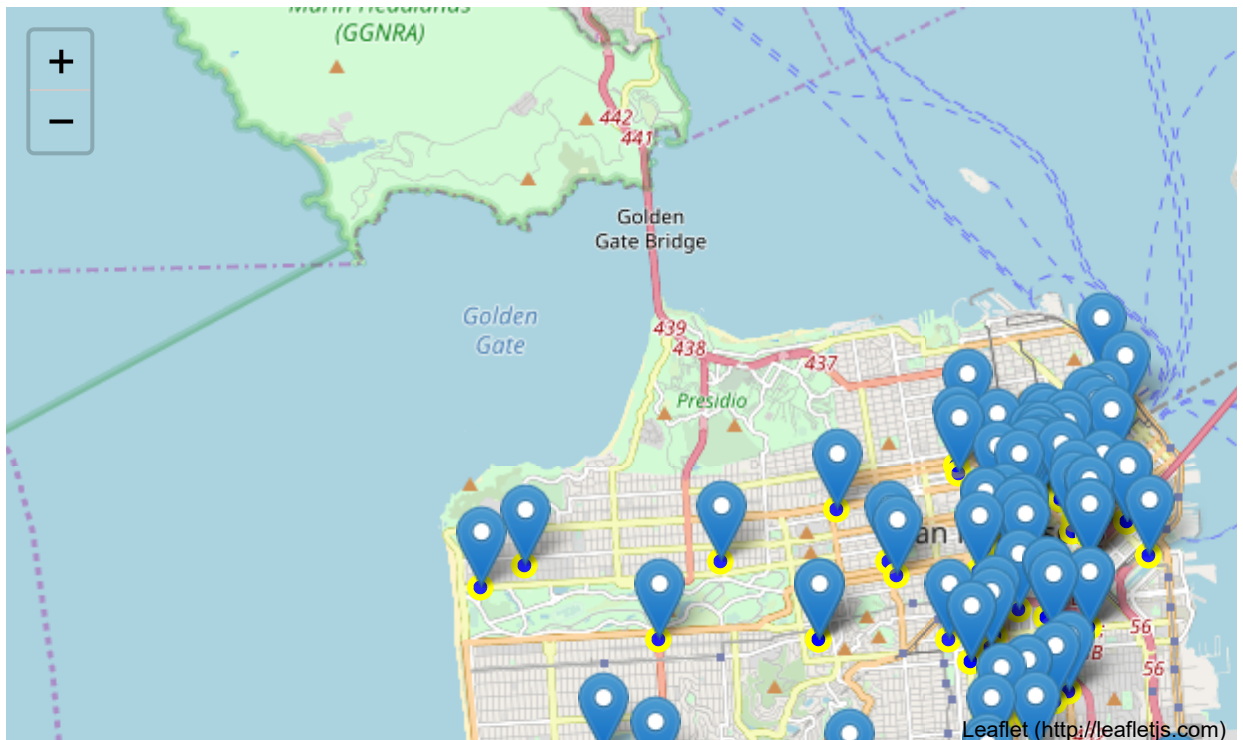
# add pop-up text to each marker on the map
latitudes = list(df_incidents.Y)
longitudes = list(df_incidents.X)
labels = list(df_incidents.Category)

for lat, lng, label in zip(latitudes, longitudes, labels):
    folium.Marker([lat, lng], popup=label).add_to(sanfran_map)

# add incidents to map
sanfran_map.add_child(incidents)

```

Out[19]:



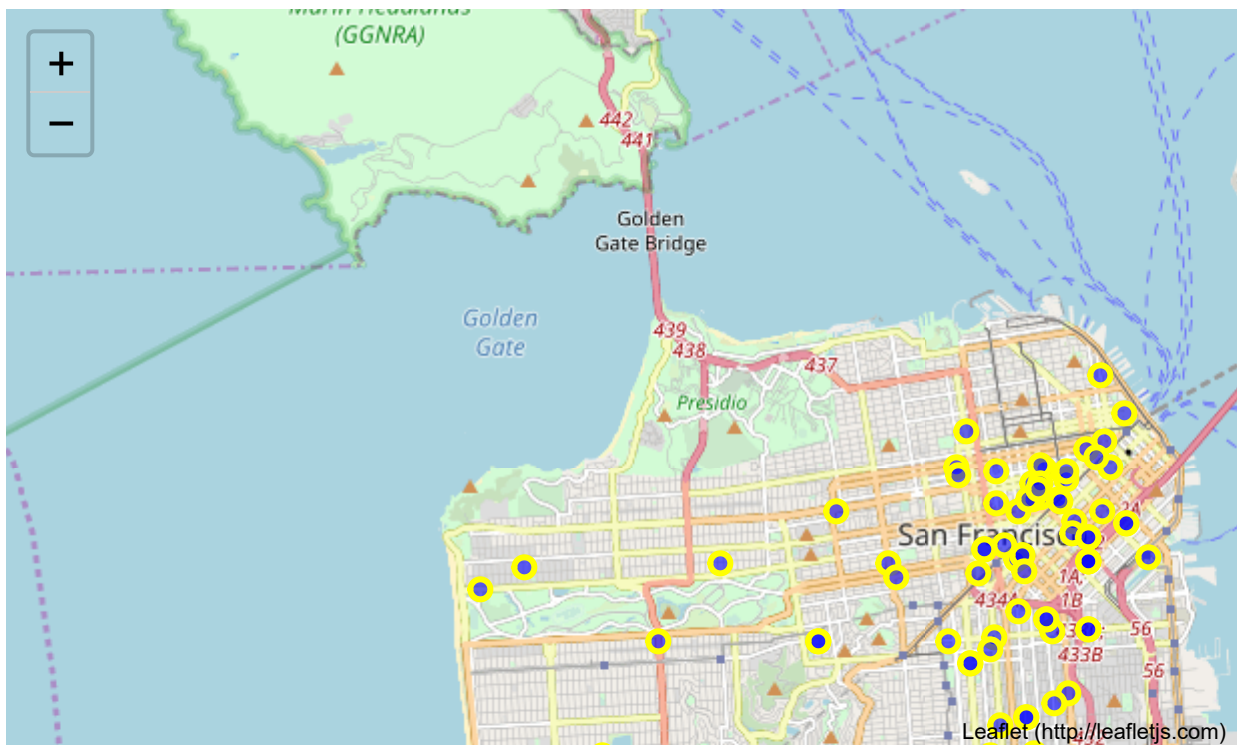


```
In [20]: # create map and display it
sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)

# Loop through the 100 crimes and add each to the map
for lat, lng, label in zip(df_incidents.Y, df_incidents.X, df_incidents.Category):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5, # define how big you want the circle markers to be
        color='yellow',
        fill=True,
        popup=label,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(sanfran_map)

# show map
sanfran_map
```

Out[20]:





```
In [21]: from folium import plugins

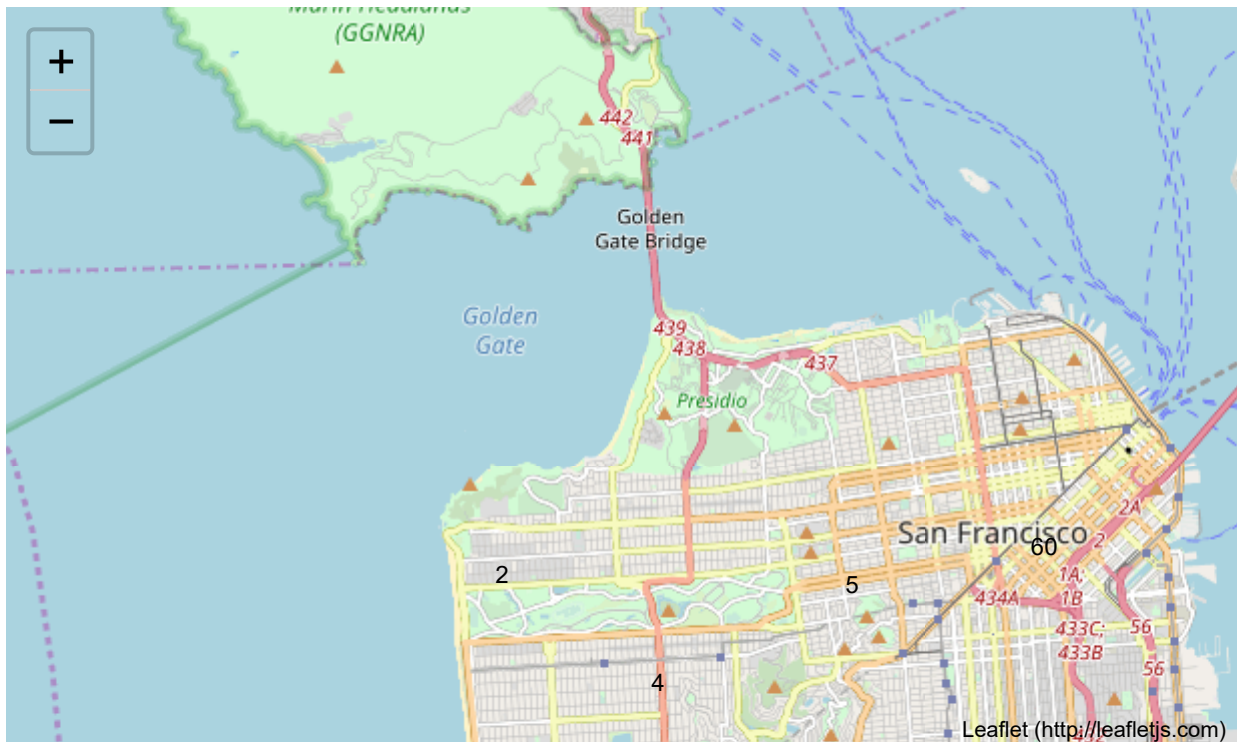
# Let's start again with a clean copy of the map of San Francisco
sanfran_map = folium.Map(location = [latitude, longitude], zoom_start = 12)

# instantiate a mark cluster object for the incidents in the dataframe
incidents = plugins.MarkerCluster().add_to(sanfran_map)

# Loop through the dataframe and add each data point to the mark cluster
for lat, lng, label, in zip(df_incidents.Y, df_incidents.X, df_incidents.Category)
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(incidents)

# display map
sanfran_map
```

Out[21]:



Copyright © 2018 [Cognitive Class \(https://cognitiveclass.ai/?utm\\_source=bducopyrightlink&utm\\_medium=dswb&utm\\_campaign=bdu\)](https://cognitiveclass.ai/?utm_source=bducopyrightlink&utm_medium=dswb&utm_campaign=bdu). This notebook and its source code are released under the terms of the [MIT License \(https://bigdatauniversity.com/mit-license/\)](https://bigdatauniversity.com/mit-license/).

In [ ]:



