Launch Sites Locations Analysis with Folium

Nick Belgau

The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.

In this notebook, we will be performing more interactive visual analytics using `Folium` to find geographical patterns about launch sites.

Objectives

The following is accomplished in this notebook:

- Marking all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities

Import Libraries

In [1]:
```
!pip3 install folium
!pip3 install wget
```

```
/opt/conda/envs/Python-3.7-OpenCE/lib/python3.7/site-packages/secretstorage/d
hcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated,
use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-OpenCE/lib/python3.7/site-packages/secretstorage/u
til.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use
int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting folium
  Downloading folium-0.12.1-py2.py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 6.8 MB/s  eta 0:00:01
Requirement already satisfied: numpy in /opt/conda/envs/Python-3.7-OpenCE/li
b/python3.7/site-packages (from folium) (1.19.2)
Collecting branca>=0.3.0
  Downloading branca-0.4.2-py3-none-any.whl (24 kB)
Requirement already satisfied: jinja2>=2.9 in /opt/conda/envs/Python-3.7-Open
CE/lib/python3.7/site-packages (from folium) (2.11.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.7-OpenCE/
lib/python3.7/site-packages (from folium) (2.22.0)
Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/envs/Python-3.7
-OpenCE/lib/python3.7/site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-
3.7-OpenCE/lib/python3.7/site-packages (from requests->folium) (2021.5.30)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /opt/conda/envs/Pytho
n-3.7-OpenCE/lib/python3.7/site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in /opt/conda/envs/Python-3.7-O
penCE/lib/python3.7/site-packages (from requests->folium) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /op
t/conda/envs/Python-3.7-OpenCE/lib/python3.7/site-packages (from requests->fo
lium) (1.25.9)
Installing collected packages: branca, folium
Successfully installed branca-0.4.2 folium-0.12.1
/opt/conda/envs/Python-3.7-OpenCE/lib/python3.7/site-packages/secretstorage/d
hcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated,
use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-OpenCE/lib/python3.7/site-packages/secretstorage/u
til.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use
int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting wget
  Downloading wget-3.2.zip (10 kB)
Building wheels for collected packages: wget
  Building wheel for wget (setup.py) ... done
  Created wheel for wget: filename=wget-3.2-py3-none-any.whl size=9681 sha256
=5ac64bcd3551b1cf6d4ea70a8ba5af3972344b8dfb48cd5386c842944c32467f
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/a1/b6/7c/0e63e34eb066341
81c63adacca38b79ff8f35c37e3c13e3c02
Successfully built wget
Installing collected packages: wget
Successfully installed wget-3.2
```

In [5]:
```
import folium
import wget
import pandas as pd
```

In [6]:
```
from folium.plugins import MarkerCluster
from folium.plugins import MousePosition
from folium.features import DivIcon
```

Mark all launch sites on a map

Add each site's location on a map using site's latitude and longitude coordinates using augmented dataset: `spacex_launch_geo.csv'.

In [7]:
```
# Download and read the `spacex_launch_geo.csv`
spacex_csv_file = wget.download('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv')
spacex_df=pd.read_csv(spacex_csv_file)
```

Define coordinates for each site

In [8]:
```
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

Out[8]:

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

Pin coordinates on the map.

Create a folium `Map` object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.

In [9]:
```
# Start location is NASA Johnson Space Center
nasa_coordinate = [29.559684888503615, -95.0830971930759]
site_map = folium.Map(location=nasa_coordinate, zoom_start=10)
```

Use `folium.Circle` to add a highlighted circle area with a text label on a specific coordinate.

In [10]:
```
# Create a blue circle at NASA Johnson Space Center's coordinate with a popup
 label showing its name
circle = folium.Circle(nasa_coordinate, radius=1000, color='#d35400', fill=Tru
e).add_child(folium.Popup('NASA Johnson Space Center'))

# Create a blue circle at NASA Johnson Space Center's coordinate with a icon s
howing its name
marker = folium.map.Marker(
    nasa_coordinate,
    # Create an icon as a text label
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'N
ASA JSC',
        )
    )
site_map.add_child(circle)
site_map.add_child(marker)
```

Out[10]: Make this Notebook Trusted to load map: File -> Trust Notebook

Add a circle for each launch site in data frame `launch_sites`

In [11]:
```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)


# For each launch site, add a Circle object based on its coordinate (Lat, Lon
g) values.
# In addition, add Launch site name as a popup label

for index, site in launch_sites_df.iterrows():
    circle = folium.Circle([site['Lat'], site['Long']], color='#d35400', radiu
s=50, fill=True).add_child(folium.Popup(site['Launch Site']))
    marker = folium.Marker(
    [site['Lat'], site['Long']],
    # Create an icon as a text label
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % si
te['Launch Site'],
        )
    )
    site_map.add_child(circle)
    site_map.add_child(marker)

site_map
```

Out[11]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [10]:
```
spacex_df.tail(10)
```

Out[10]:

	Launch Site	Lat	Long	class
46	KSC LC-39A	28.573255	-80.646895	1
47	KSC LC-39A	28.573255	-80.646895	1
48	KSC LC-39A	28.573255	-80.646895	1
49	CCAFS SLC-40	28.563197	-80.576820	1
50	CCAFS SLC-40	28.563197	-80.576820	1
51	CCAFS SLC-40	28.563197	-80.576820	0
52	CCAFS SLC-40	28.563197	-80.576820	0
53	CCAFS SLC-40	28.563197	-80.576820	0
54	CCAFS SLC-40	28.563197	-80.576820	1
55	CCAFS SLC-40	28.563197	-80.576820	0

Create markers for all launch records. If a launch was successful `(class=1)`, then we use a green marker and if a launch was failed, we use a red marker `(class=0)`

Note that a launch only happens in one of the four launch sites, which means many launch records will have the exact same coordinate. Marker clusters can be a good way to simplify a map containing many markers having the same coordinate.

In [13]:
```
marker_cluster = MarkerCluster()
```

In [14]:
```
# Function to assign color to launch outcome
def assign_marker_color(launch_outcome):
    if launch_outcome == 1:
        return 'green'
    else:
        return 'red'

spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
spacex_df.tail(10)
```

Out[14]:

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

In [15]:
```
site_map.add_child(marker_cluster)

for index, record in spacex_df.iterrows():
    marker = folium.Marker([record['Lat'], record['Long']],
                   icon=folium.Icon(color='white', icon_color=record['marker_co
lor']))
    marker_cluster.add_child(marker)
site_map
```

Out[15]: Make this Notebook Trusted to load map: File -> Trust Notebook

Calculate the distances between a launch site to its proximities

Explore and analyze the proximities of launch sites.

`MousePosition` gives coordinate for a mouse over a point on the map

In [16]:
```
formatter = "function(num) {return L.Util.formatNum(num, 5);};"
mouse_position = MousePosition(
    position='topright',
    separator=' Long: ',
    empty_string='NaN',
    lng_first=False,
    num_digits=20,
    prefix='Lat:',
    lat_formatter=formatter,
    lng_formatter=formatter,
)

site_map.add_child(mouse_position)
site_map
```

Out[16]: Make this Notebook Trusted to load map: File -> Trust Notebook

Calculate the distance between two points on the map based on their `Lat` and `Long` values using the following method:

In [17]:
```
from math import sin, cos, sqrt, atan2, radians

def calculate_distance(lat1, lon1, lat2, lon2):
    # approximate radius of earth in km
    R = 6373.0

    lat1 = radians(lat1)
    lon1 = radians(lon1)
    lat2 = radians(lat2)
    lon2 = radians(lon2)

    dlon = lon2 - lon1
    dlat = lat2 - lat1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    distance = R * c
    return distance
```

In [18]:
```
#Distance to Highway
coordinates = [
    [28.56342, -80.57674],
    [28.411780, -80.820630]]

lines=folium.PolyLine(locations=coordinates, weight=1)
site_map.add_child(lines)
distance = calculate_distance(coordinates[0][0], coordinates[0][1], coordinate
s[1][0], coordinates[1][1])
distance_circle = folium.Marker(
    [28.411780, -80.820630],
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#252526;"><b>%s</b></div>' % "
{:10.2f} KM".format(distance),
        )
    )
site_map.add_child(distance_circle)
site_map
```

Out[18]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [19]:
```
#Distance to Florida City

coordinates = [
    [28.56342, -80.57674],
    [28.5383, -81.3792]]

lines=folium.PolyLine(locations=coordinates, weight=1)
site_map.add_child(lines)
distance = calculate_distance(coordinates[0][0], coordinates[0][1], coordinate
s[1][0], coordinates[1][1])
distance_circle = folium.Marker(
    [28.5383, -81.3792],
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#252526;"><b>%s</b></div>' % "
{:10.2f} KM".format(distance),
        )
    )
site_map.add_child(distance_circle)
site_map
```

Out[19]: Make this Notebook Trusted to load map: File -> Trust Notebook

Some questions and answers about our map data:

Are launch sites in close proximity to railways? No

Are launch sites in close proximity to highways? No

Are launch sites in close proximity to coastline? Yes

Do launch sites keep certain distance away from cities? Yes