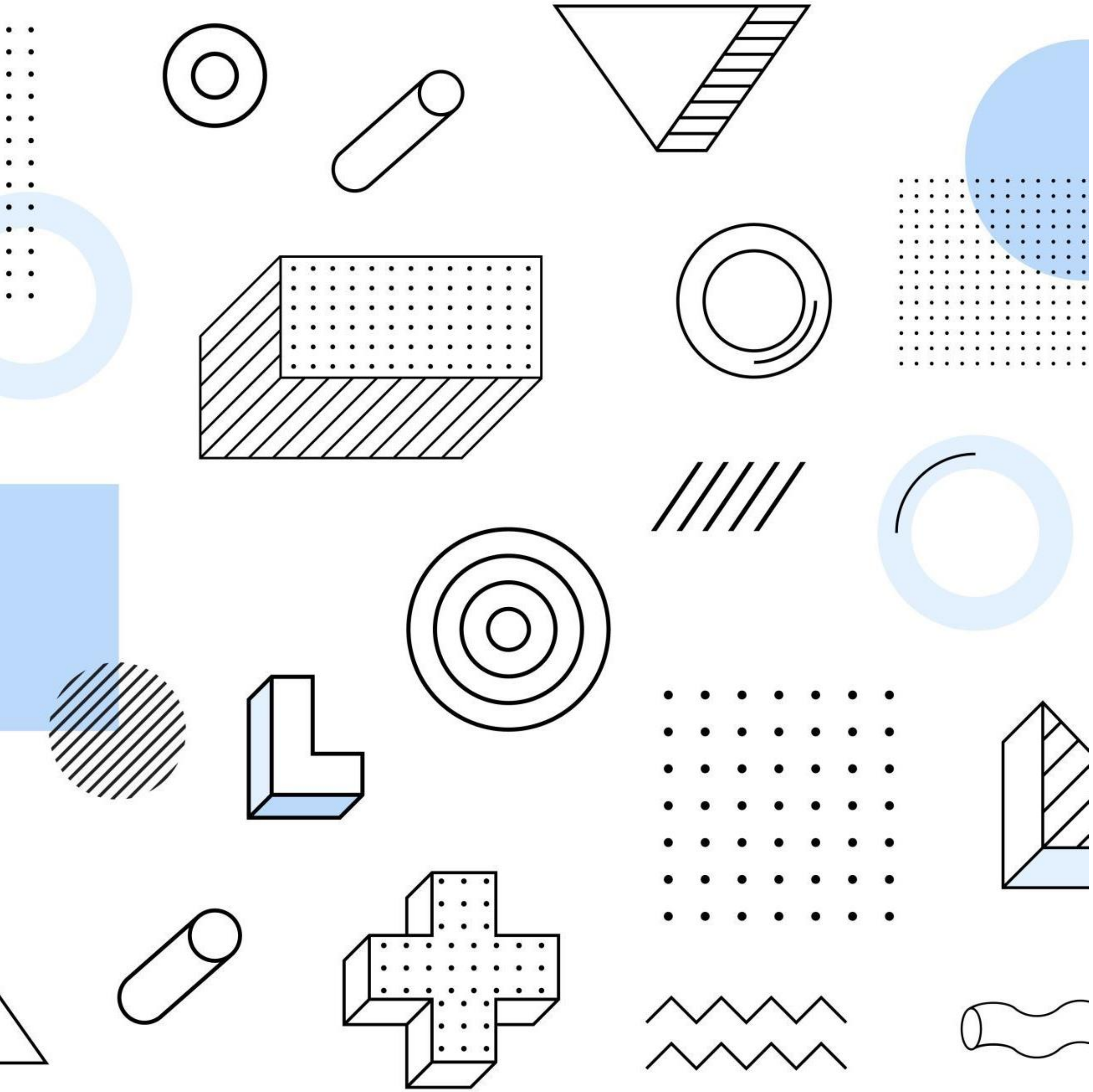


IBM Applied Data Science Capstone Project

Presented By:

Soumesh khuntia

10th Nov, 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

Summary of methodologies

- Data Collection
- Data Wrangling
- Exploratory Data Analysis with Data Visualization
- Exploratory Data Analysis with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly & Dash
- Predictive Analysis (Classification)

Summary of all results

- Exploratory Data Analysis
- Interactive Analytics Demo in screenshots
- Predictive Analysis results

Introduction

SpaceX is the most successful company of the commercial space age, making space travel affordable. The company advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Based on public information and machine learning models, we are going to predict if SpaceX will reuse the first stage.

Methodology

Data Collection Methodology

- Using SpaceX Rest API
- Using Web Scraping from Wikipedia

Performed data wrangling

- Filtering the data
- Dealing with missing values
- Using One Hot Encoding to prepare the data to a binary classification

Performed exploratory data analysis (EDA) using visualization and SQL Performed interactive visual analytics using Folium and Plotly Dash Performed predictive analysis using classification models - Building, tuning and evaluation of classification models to ensure the best results

Data Collection

This process involved a combination of API requests from SpaceX **REST API** and **Web Scrapping** data from a table in SpaceX's Wikipedia entry.

We had to use both data collection methods to get complete information about the launches for a more detailed analysis.

Data Columns obtained from **SpaceX REST API**:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Data Columns obtained by using **Wikipedia Web Scrapping**:

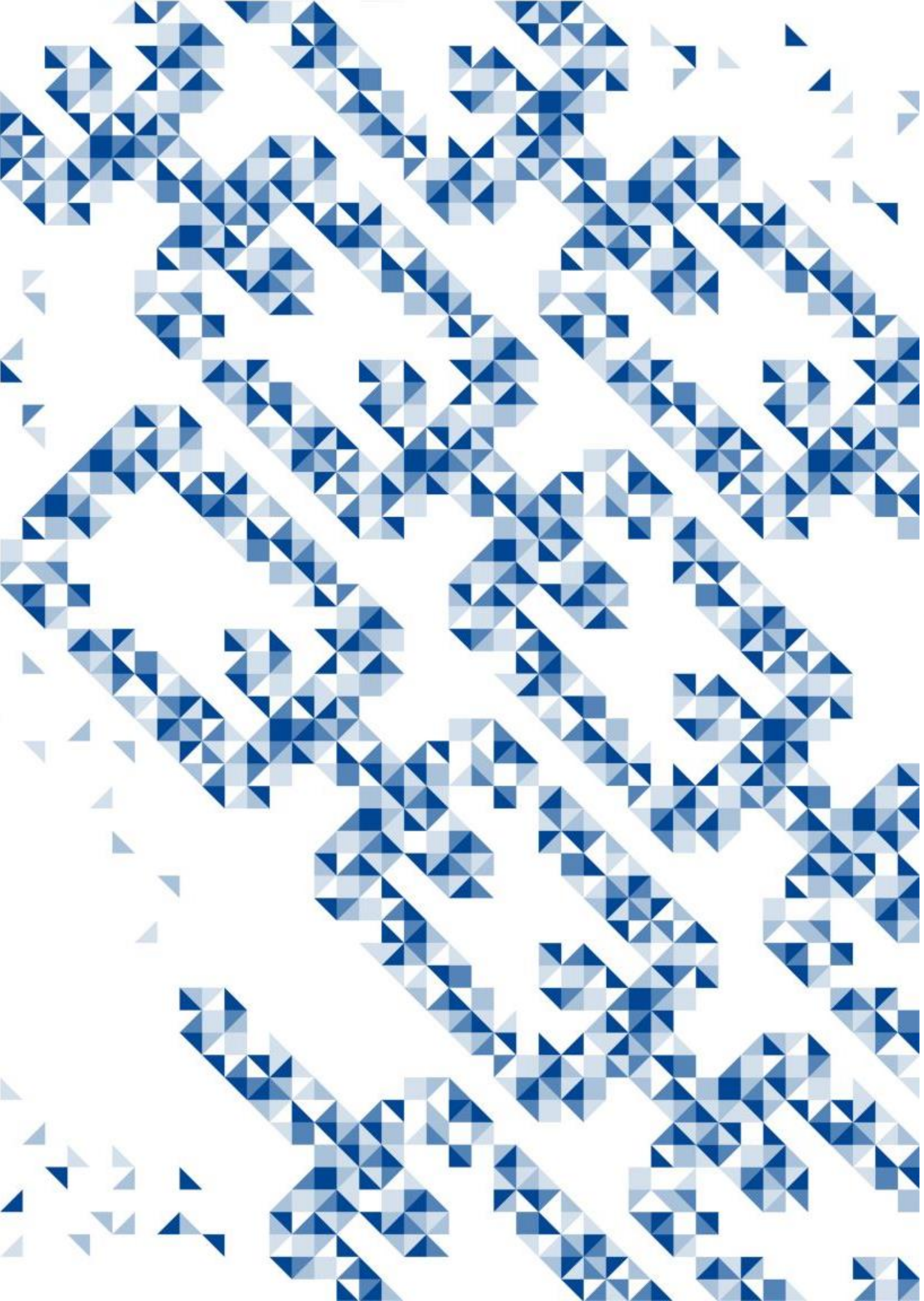
Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection - SpaceX API

- Requesting rocket launch date from SpaceX API
- Decoding the response content using .json() and turning it into a dataframe using .json_normalize()
- Requesting needed information about the launches from SpaceX API by applying custom functions
- Constructing data we have obtained into a dictionary
- Creating a dataframe from the dictionary
- Filtering the dataframe to only include Falcon-9 launches
- Replacing missing values of payload mass columns with calculated .mean()
- Exporting the data to CSV

Data Collection – Web Scrapping

1. Requesting Falcon-9 launch from Wikipedia
2. Creating a **BeautifulSoup** object from the HTML response
3. Extracting all column names from the HTML table header
4. Collecting the data by parsing HTML tables
5. Constructing data we've obtained into a dictionary
6. Creating a dataframe from the dictionary
7. Exporting the data into CSV



Data Wrangling

- Perform **Exploratory Data Analysis** and determine training labels
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type
- Creating a landing outcome label from **Outcome column**
- Exporting the data to CSV

EDA with Data Visualization

Charts were plotted: **Flight Number vs. Payload Mass**, **Flight Number vs. Launch Site**, **Payload Mass vs. Launch Site**, **Orbit Type vs. Success Rate**, **Flight Number vs. Orbit Type**, **Payload Mass vs Orbit Type** and Success Rate Yearly Trend Scatter plots show the relationship between variables.

If a relationship exists, they could be used in machine learning model. Bar charts show comparisons among discrete categories. The goal is to show the relationship between the specific categories being compared and a measured value. Line charts show trends in data over time (time series)

EDA with SQL

Performed SQL queries:

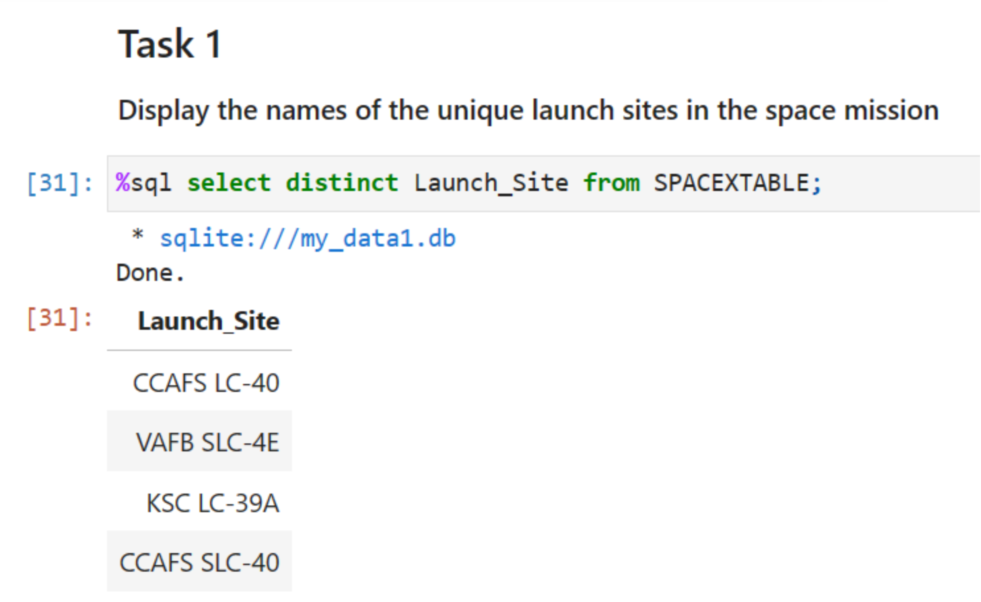
- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string ‘CCA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
- Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20 in descending order

Interactive Visual Analytics using Folium, Plotly and Dash

Markers of all Launch Sites: - Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location. - Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts. Coloured Markers of the launch outcomes for each Launch Site: - Added coloured Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates. Distances between a Launch Site to its proximities: - Added coloured Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City

Results

EDA with SQL



Task 1

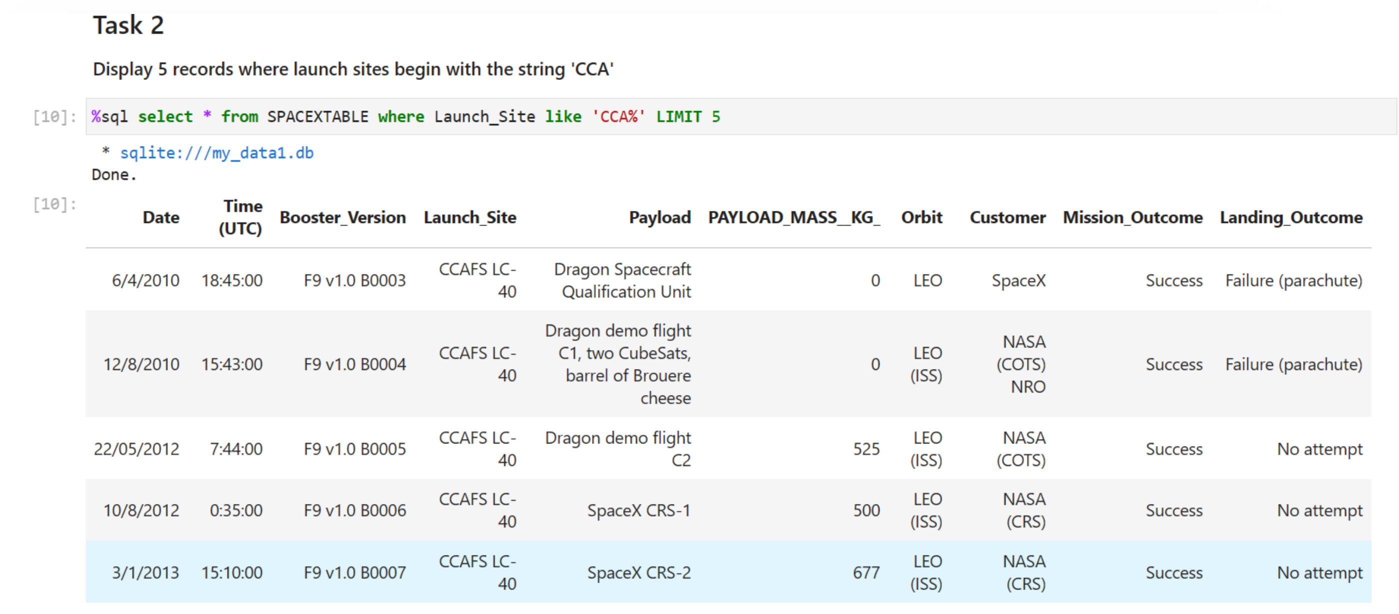
Display the names of the unique launch sites in the space mission

```
[31]: %sql select distinct Launch_Site from SPACEXTABLE;
```

```
 * sqlite:///my_data1.db
Done.
```

[31]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40



Task 2

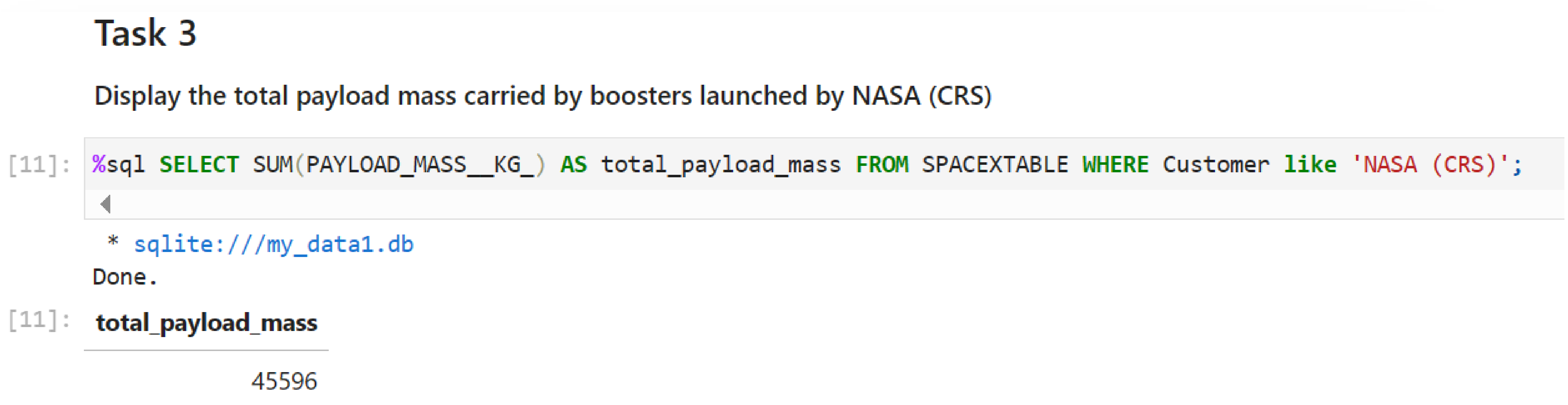
Display 5 records where launch sites begin with the string 'CCA'

```
[10]: %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
12/8/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt



Task 3

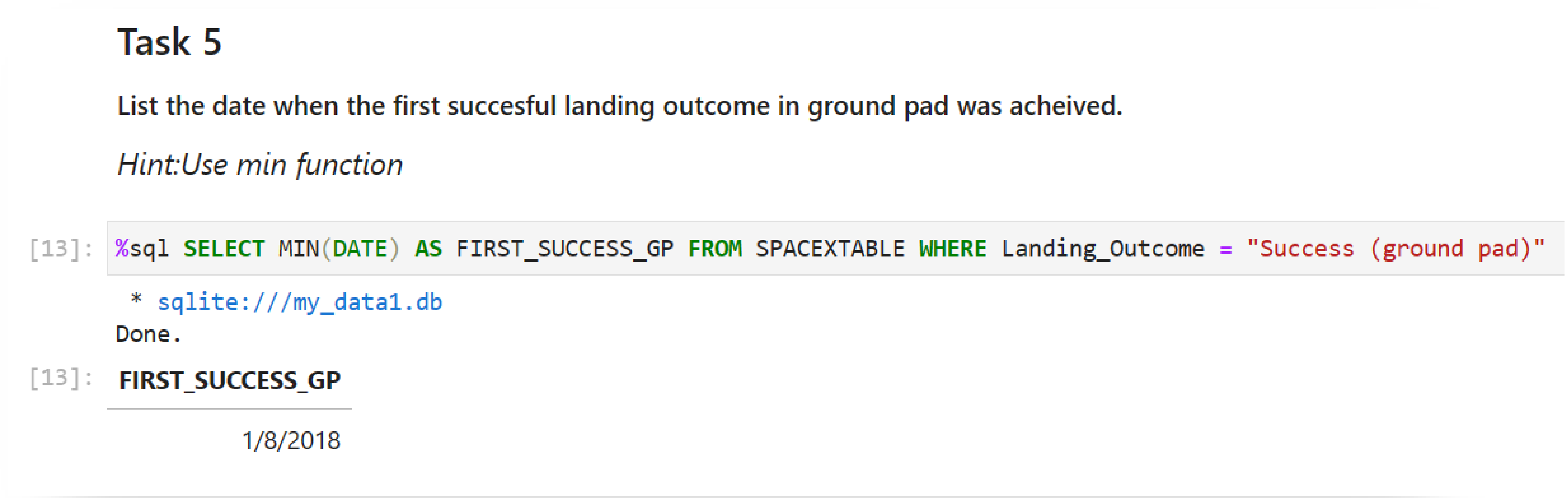
Display the total payload mass carried by boosters launched by NASA (CRS)

```
[11]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTABLE WHERE Customer like 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

[11]:

total_payload_mass
45596



Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

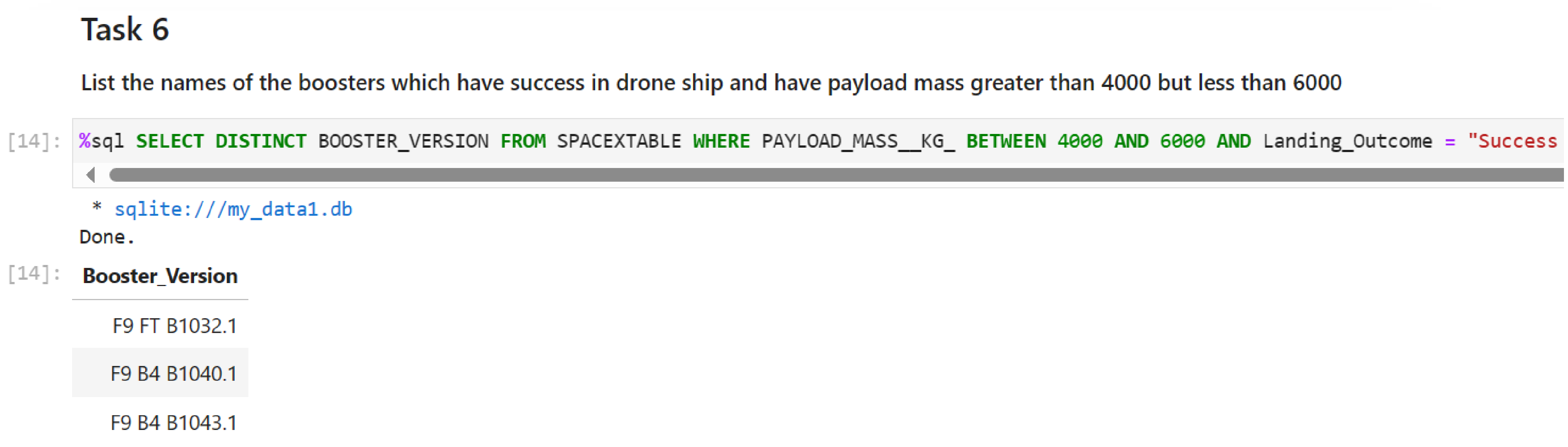
Hint:Use min function

```
[13]: %sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)"
```

```
 * sqlite:///my_data1.db
Done.
```

[13]:

FIRST_SUCCESS_GP
1/8/2018



Task 6

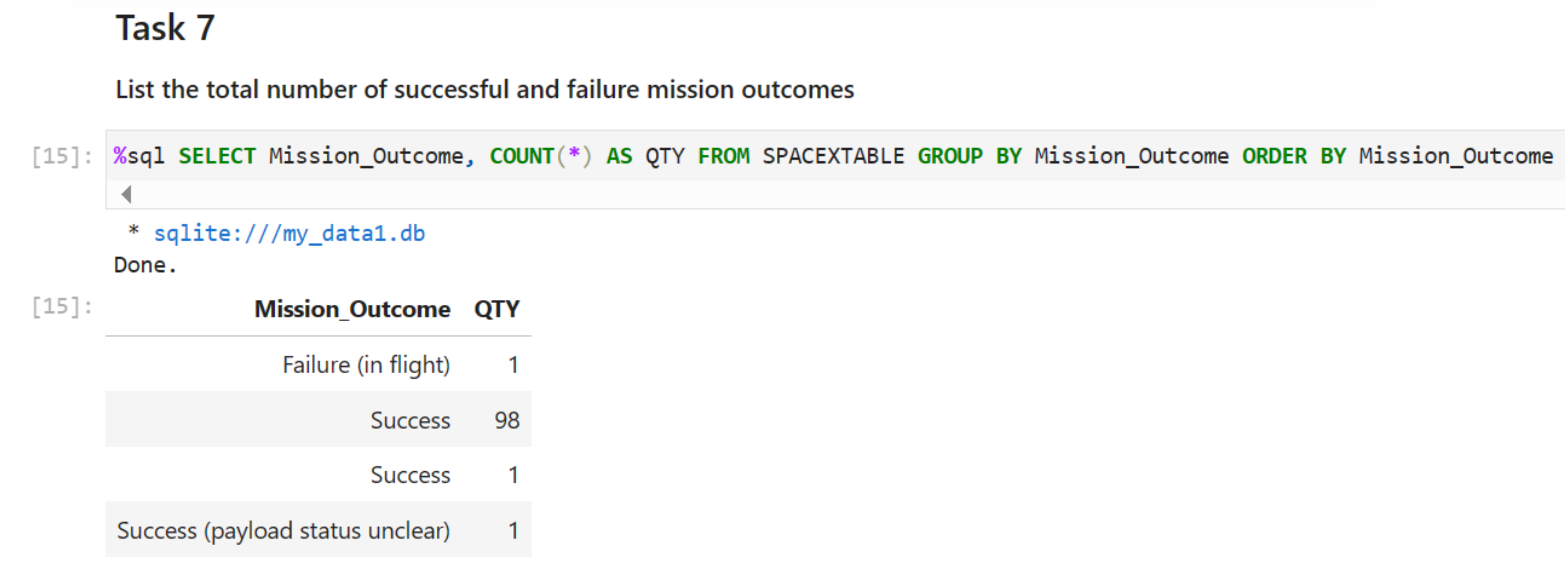
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[14]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome = "Success
```

```
 * sqlite:///my_data1.db
Done.
```

[14]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1



Task 7

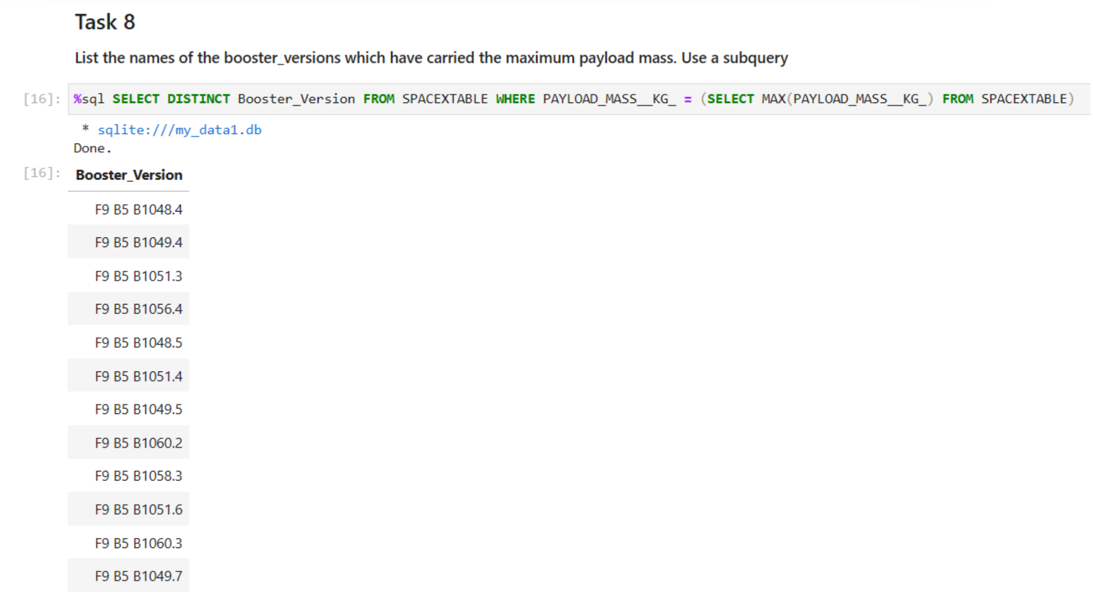
List the total number of successful and failure mission outcomes

```
[15]: %sql SELECT Mission_Outcome, COUNT(*) AS QTY FROM SPACEXTABLE GROUP BY Mission_Outcome ORDER BY Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

[15]:

Mission_Outcome	QTY
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1



Task 8

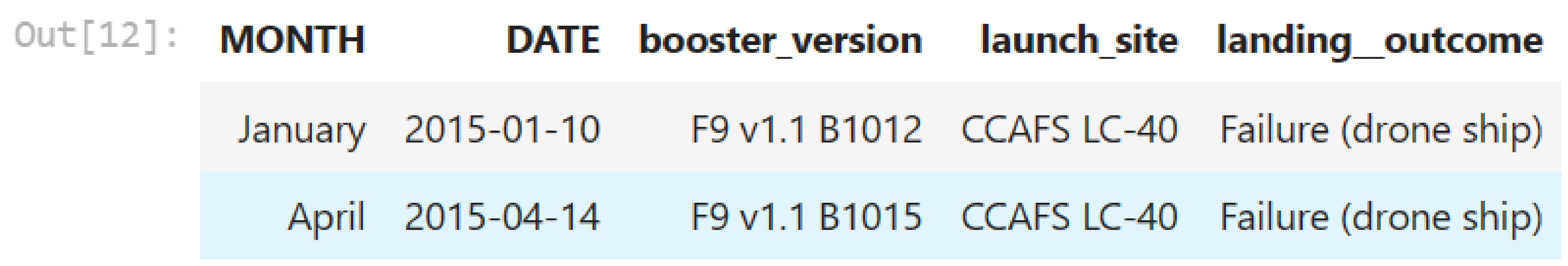
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[16]: %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

```
 * sqlite:///my_data1.db
Done.
```

```
[16]: Booster_Version
      F9 B5 B1048.4
      F9 B5 B1049.4
      F9 B5 B1051.3
      F9 B5 B1056.4
      F9 B5 B1048.5
      F9 B5 B1051.4
      F9 B5 B1049.5
      F9 B5 B1060.2
      F9 B5 B1058.3
      F9 B5 B1051.6
      F9 B5 B1060.3
      F9 B5 B1049.7
```

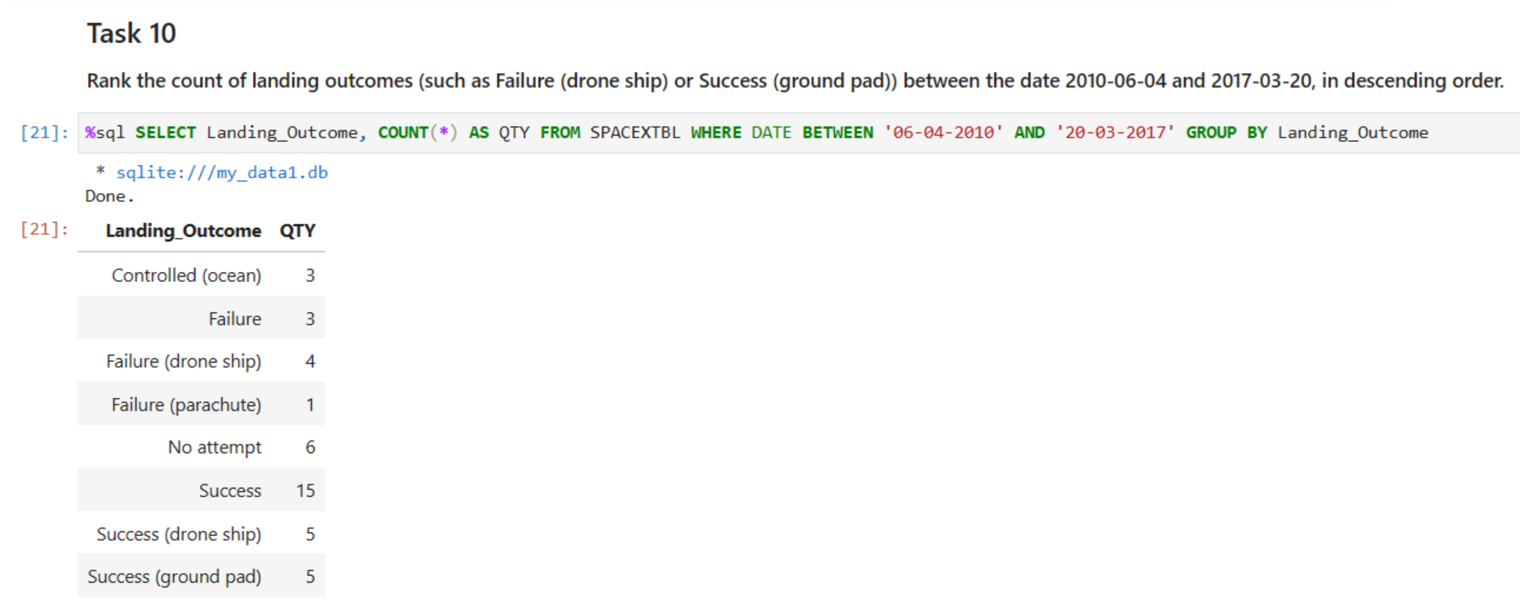
Task 9



Out[12]:

MONTH	DATE	booster_version	launch_site	landing__outcome
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Task 10



Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

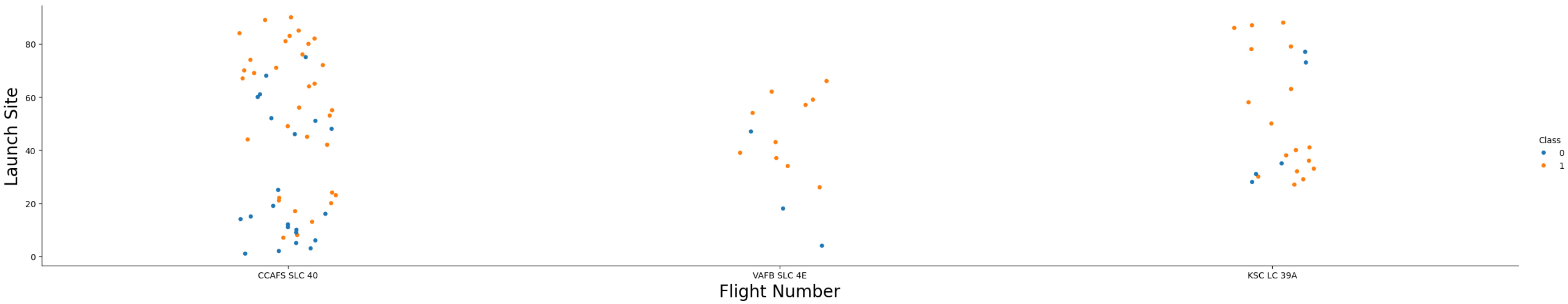
```
[21]: %sql SELECT Landing_Outcome, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '06-04-2010' AND '20-03-2017' GROUP BY Landing_Outcome
```

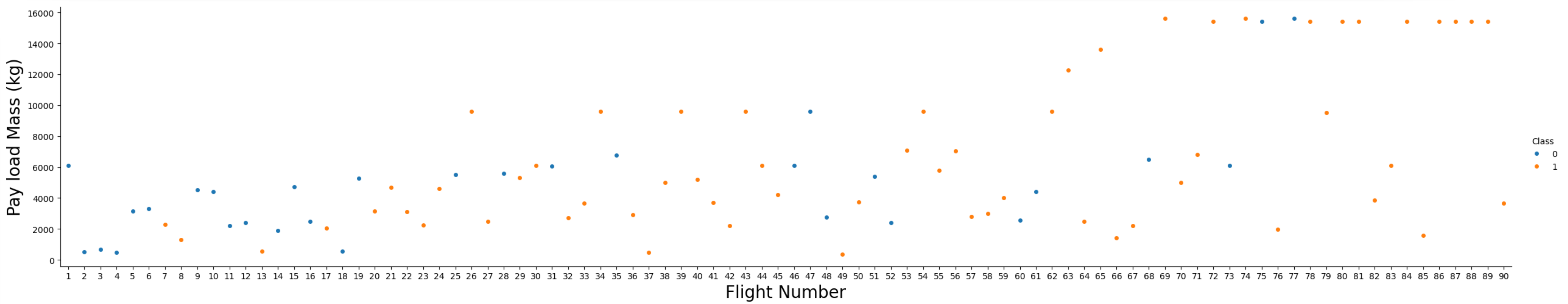
```
 * sqlite:///my_data1.db
Done.
```

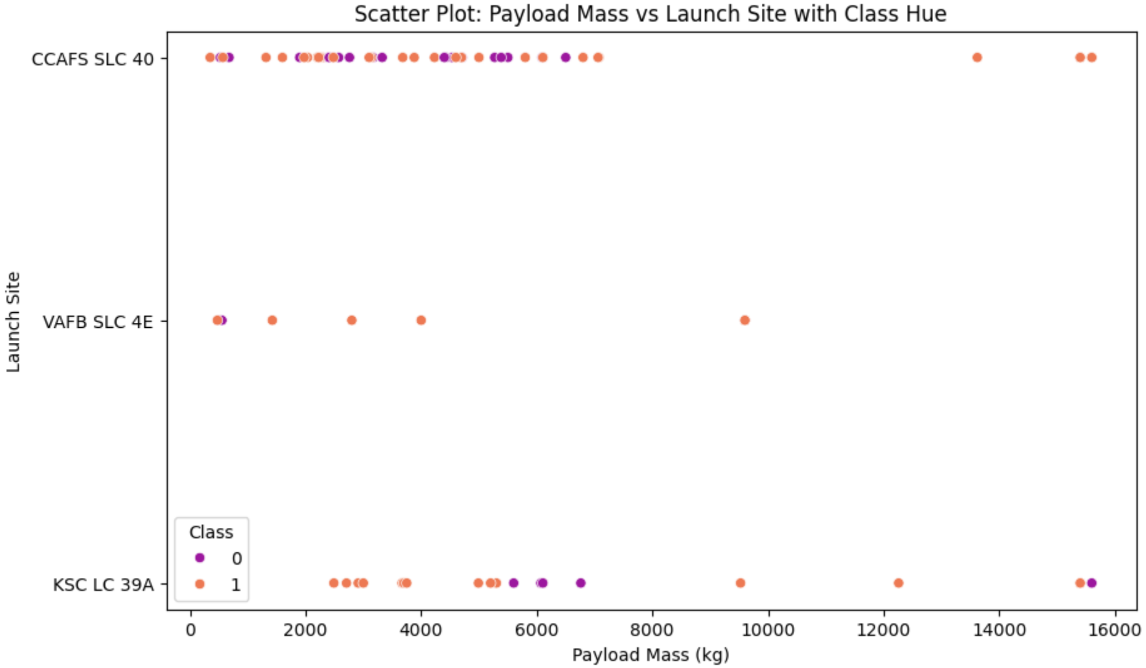
[21]:

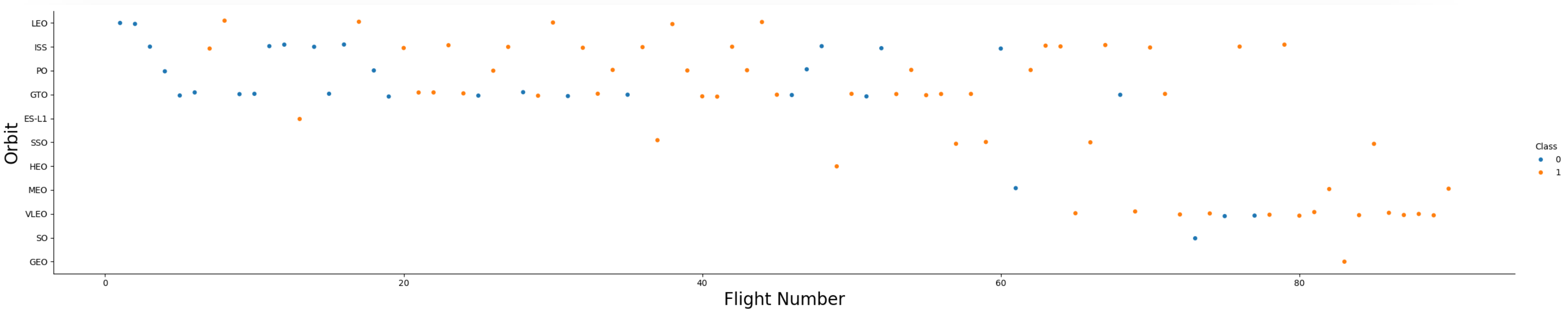
Landing_Outcome	QTY
Controlled (ocean)	3
Failure	3
Failure (drone ship)	4
Failure (parachute)	1
No attempt	6
Success	15
Success (drone ship)	5
Success (ground pad)	5

EDA using Pandas, Seaborn, Matplotlib Results

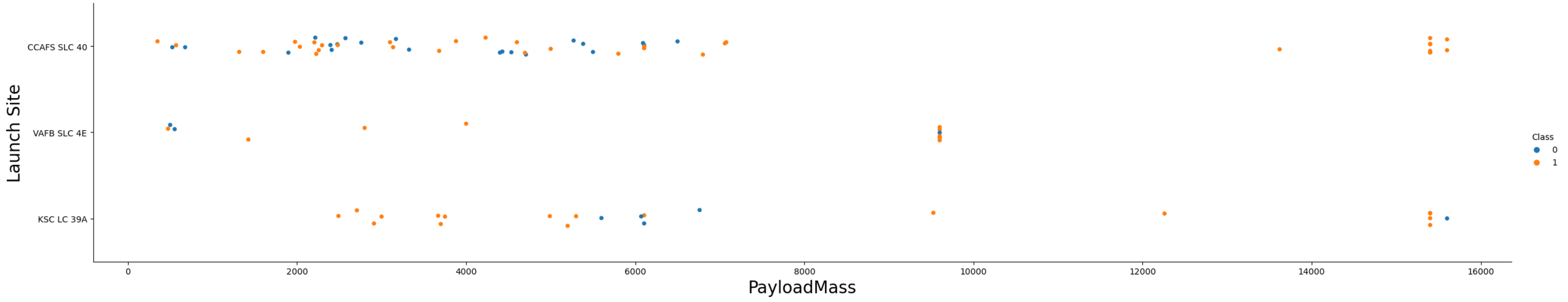

Launch Site
80
60
40
20
0
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Flight Number
Class
0
1

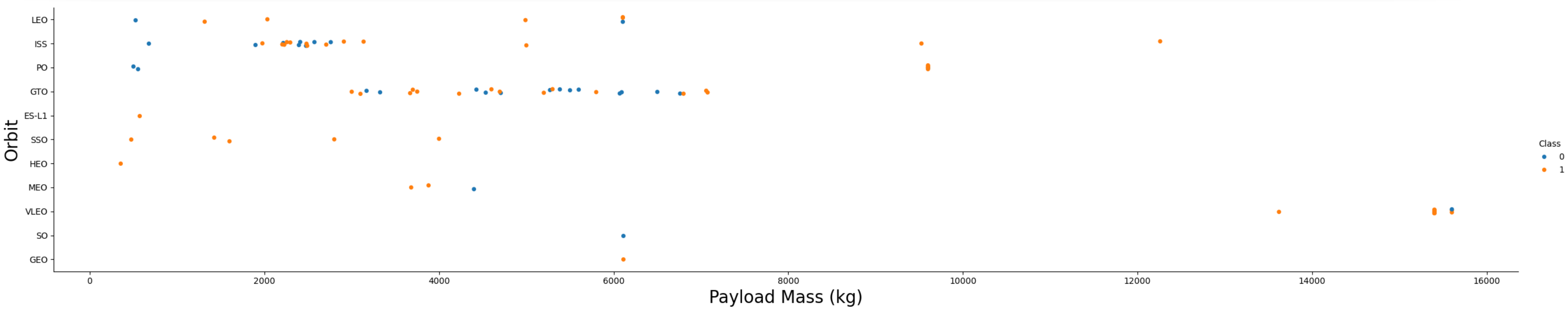

Pay load Mass (kg)
16000
14000
12000
10000
8000
6000
4000
2000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
Flight Number
Class
0
1


Scatter Plot: Payload Mass vs Launch Site with Class Hue
Launch Site
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Class
● 0
● 1
0
2000
4000
6000
8000
10000
12000
14000
16000
Payload Mass (kg)

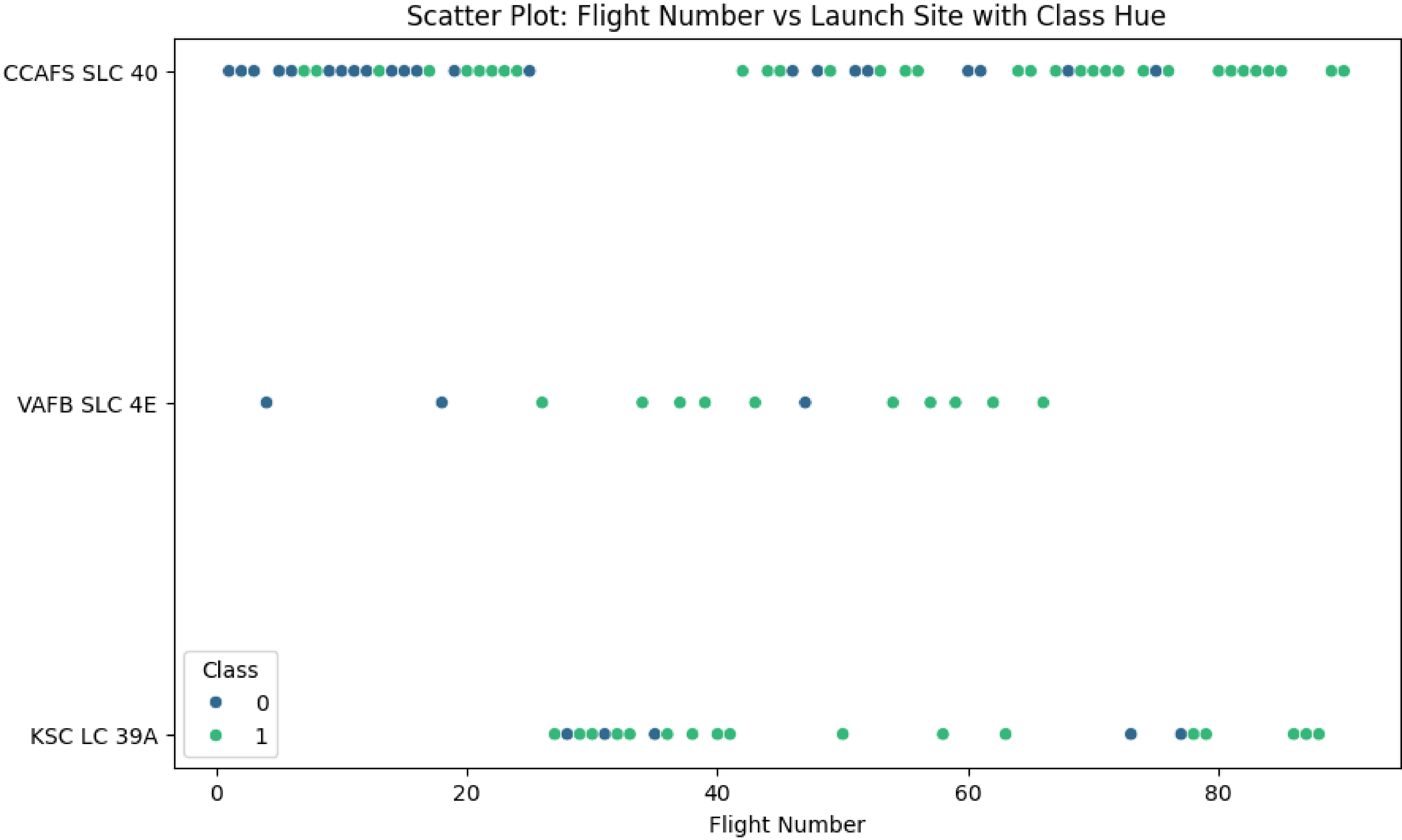


Orbit
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
0
20
40
60
80
Flight Number
Class
● 0
● 1

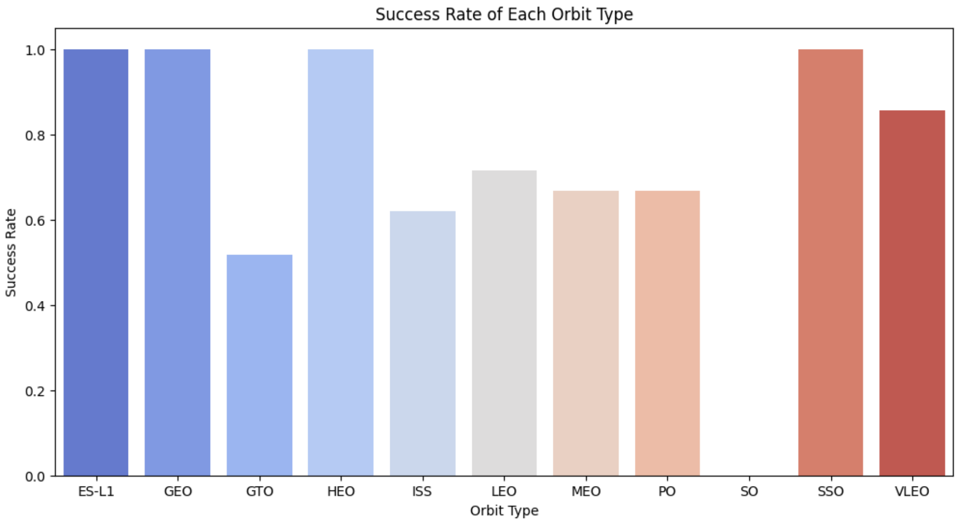

CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Launch Site
0
2000
4000
6000
8000
10000
12000
14000
16000
PayloadMass
Class
0
1



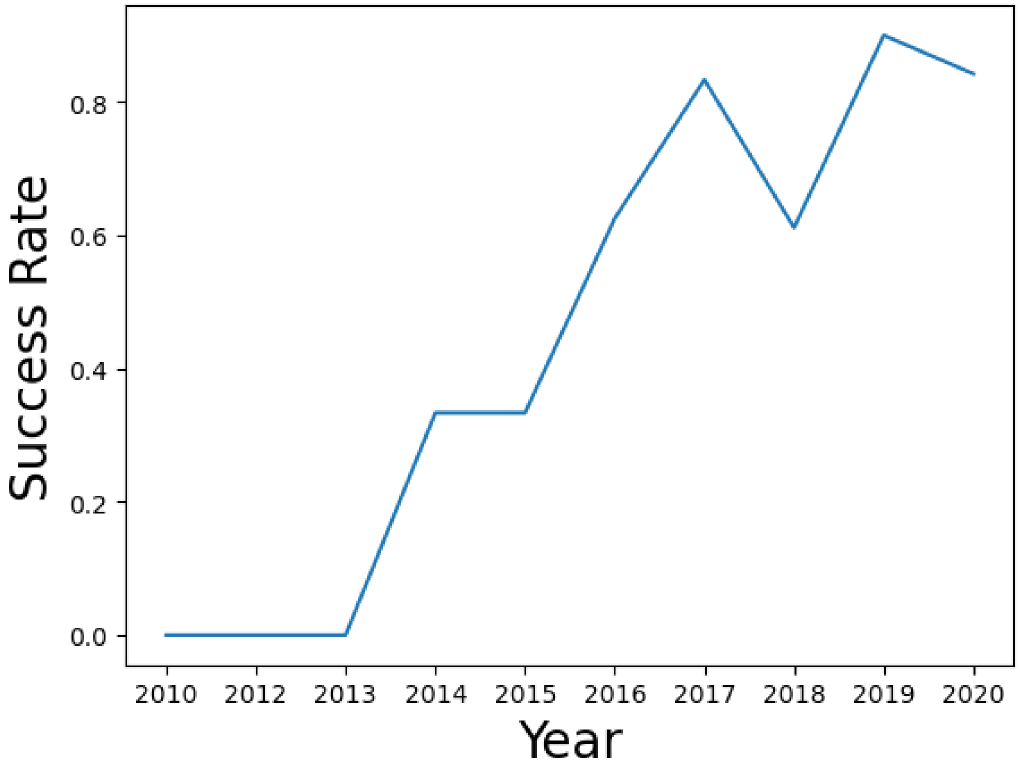
Orbit
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
0
2000
4000
6000
8000
10000
12000
14000
16000
Payload Mass (kg)
Class
0
1


Scatter Plot: Flight Number vs Launch Site with Class Hue
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Class
0
1
0
20
40
60
80
Flight Number

Success Rate

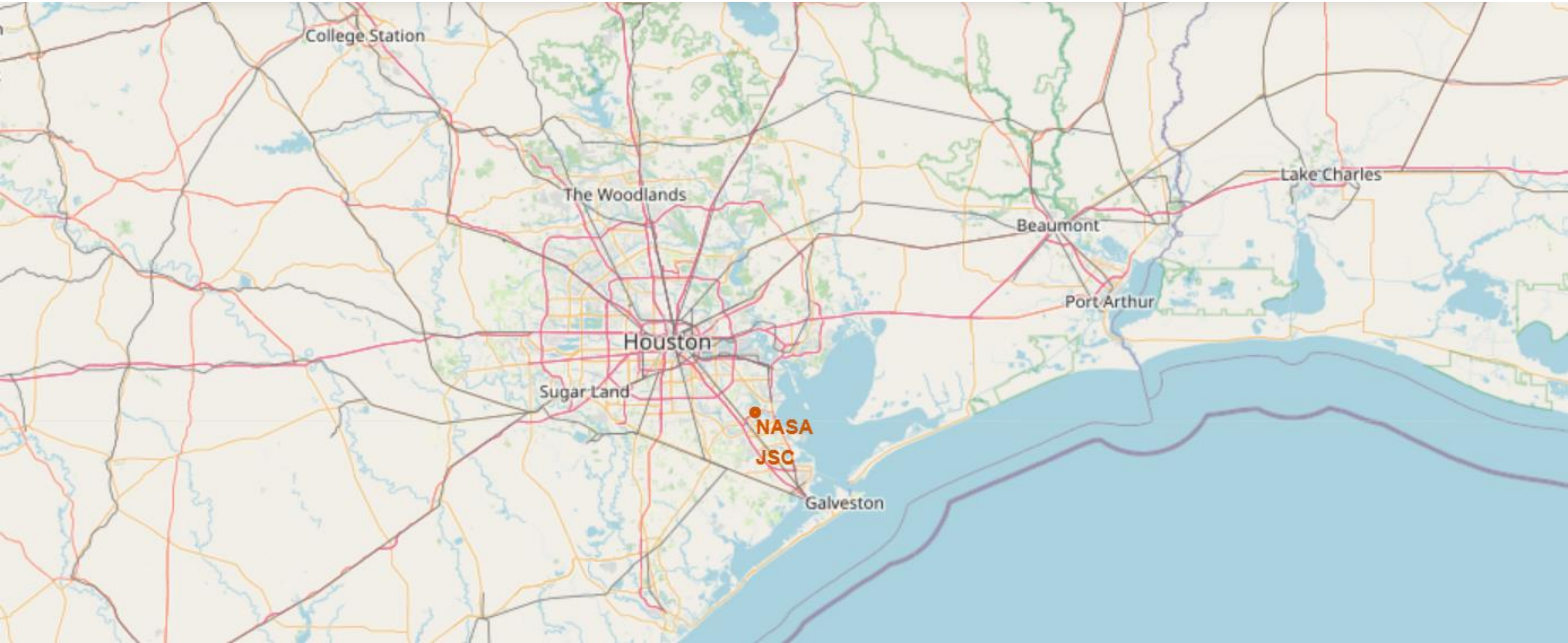


Success Rate in Years

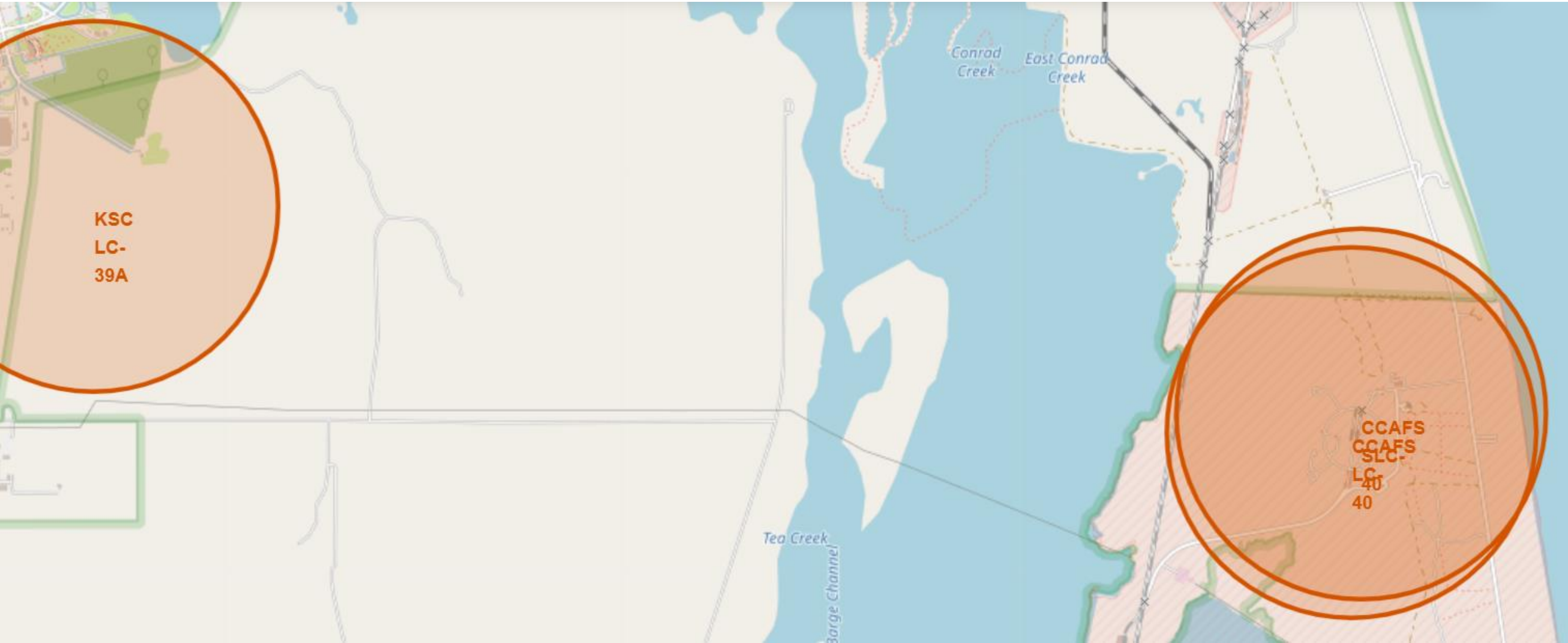


Visualizing Geospatial Data using Folium

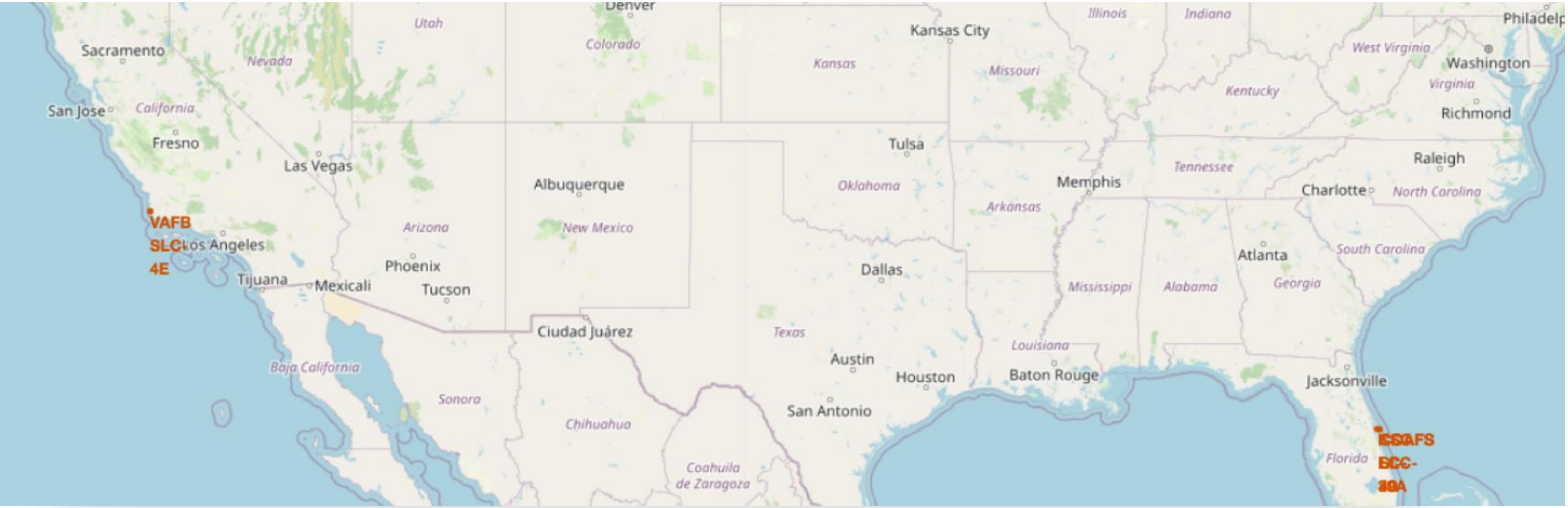
Co-ordinates and Markers

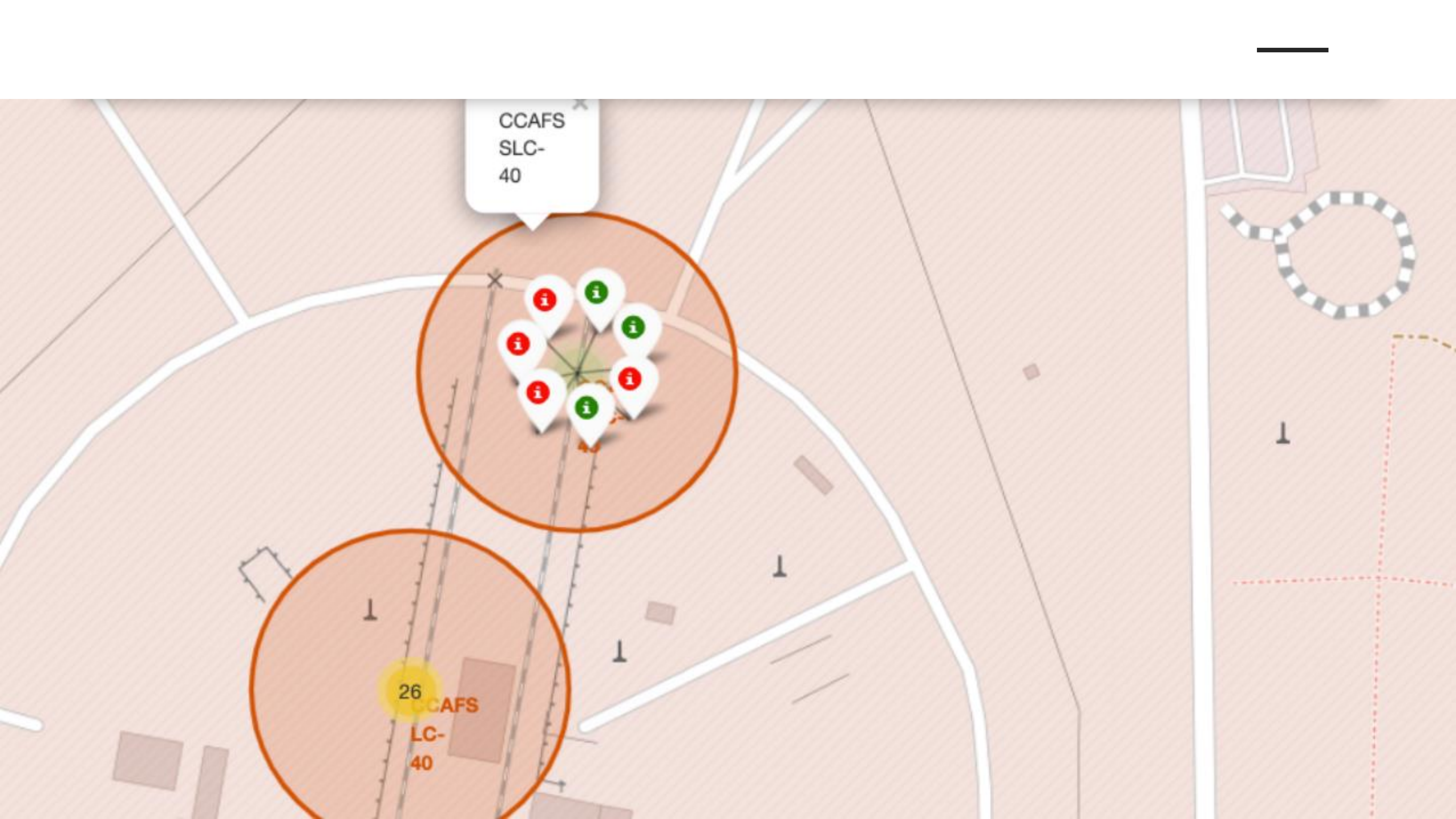


Geospatial Markers

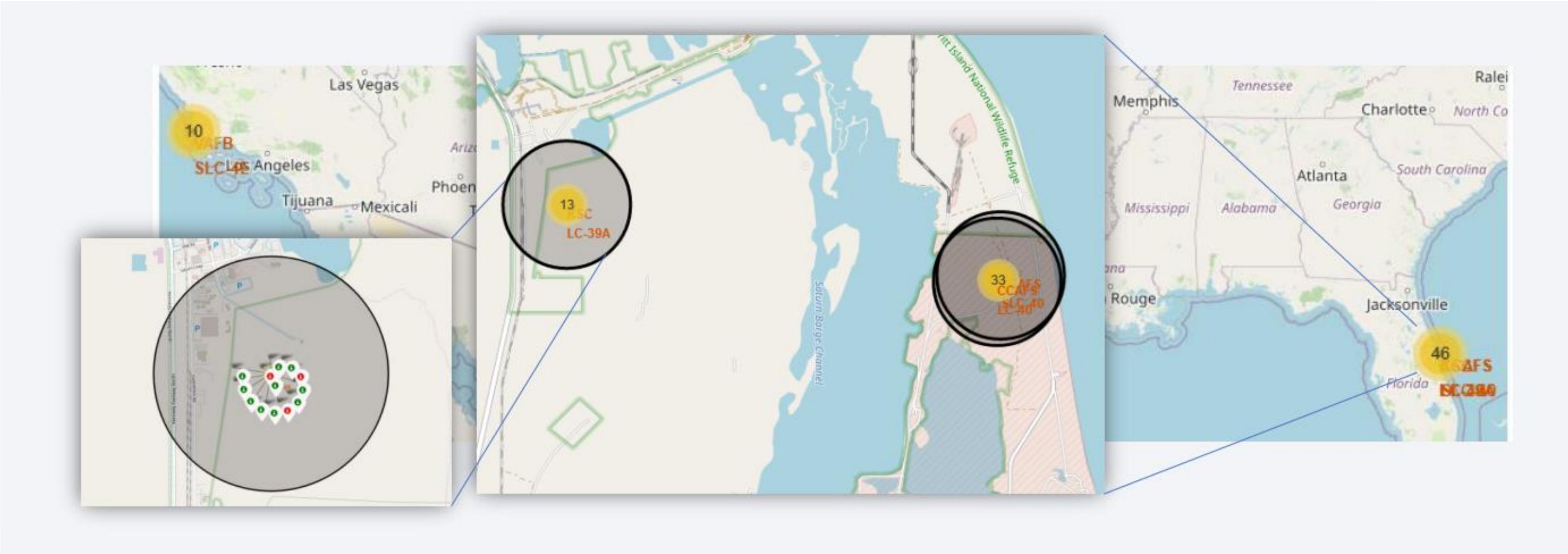


Similar Location Markers

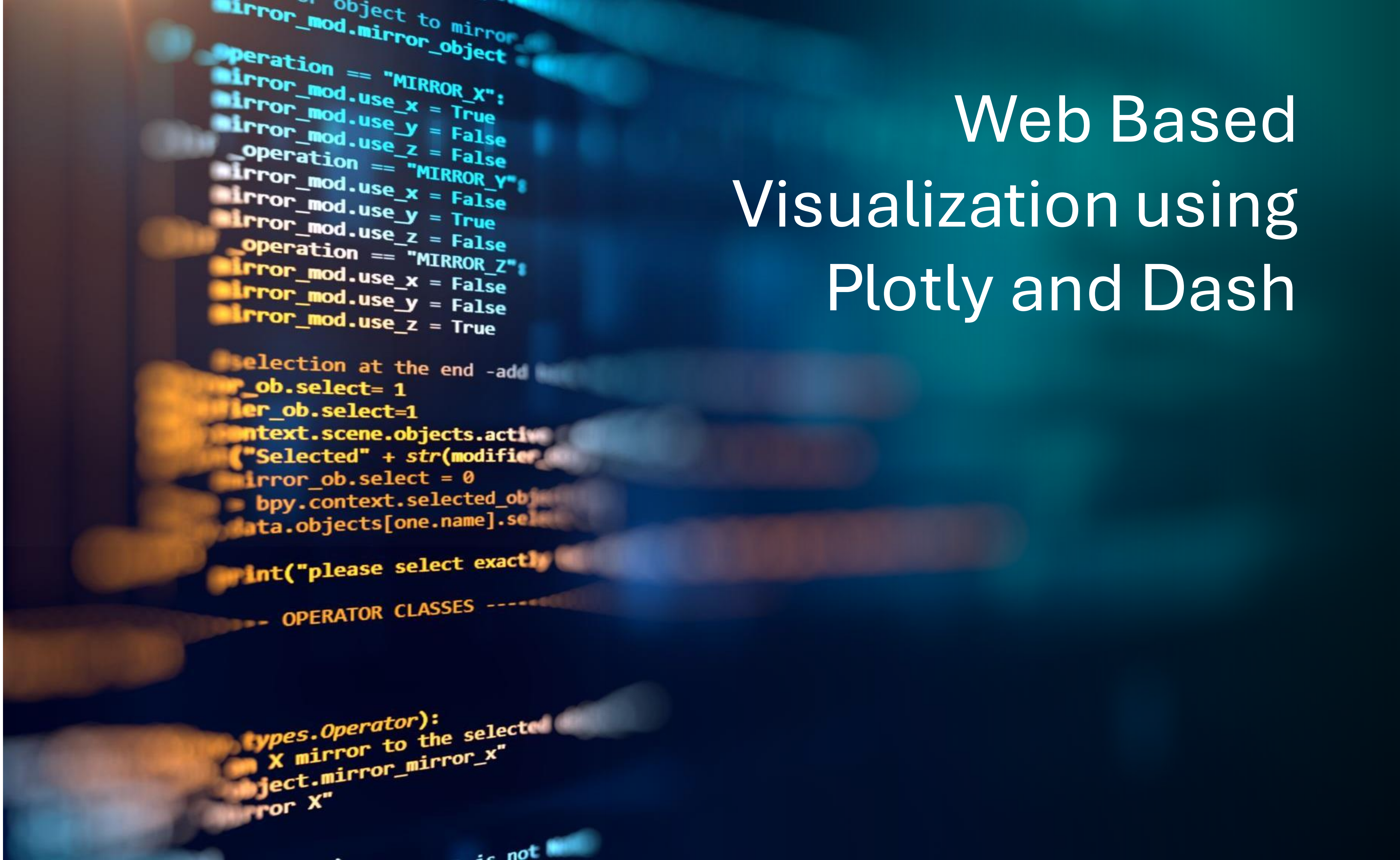



CCAFS
SLC-
40
26
CCAFS
LC-
40

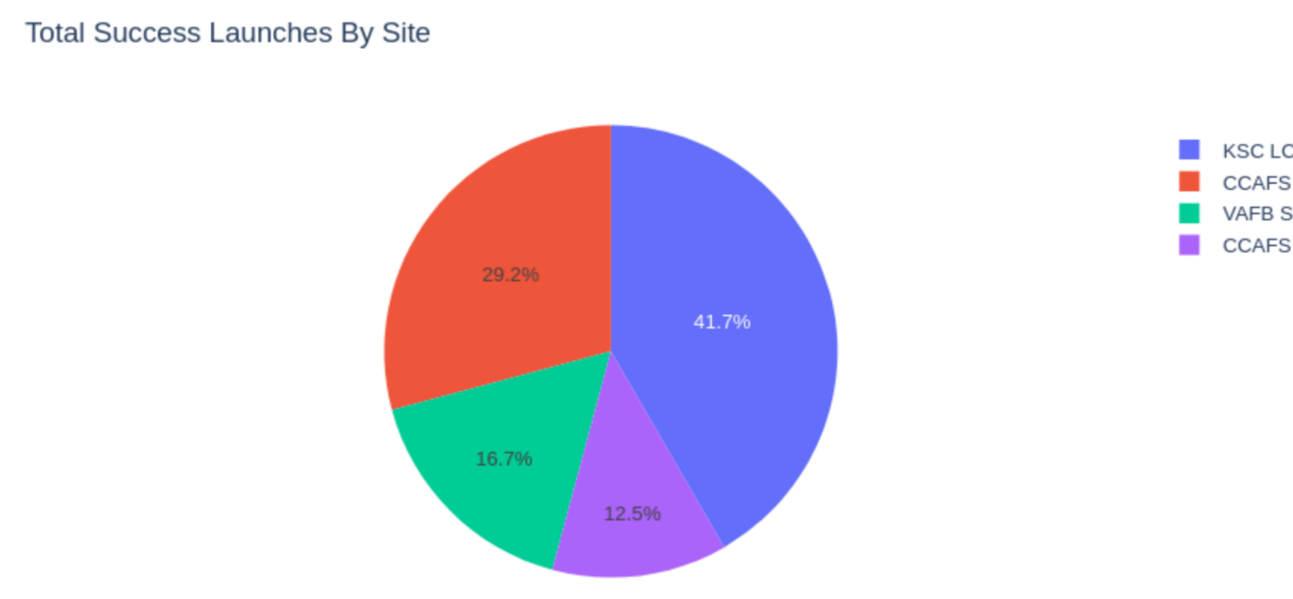
KSC LC-39A Launch Site

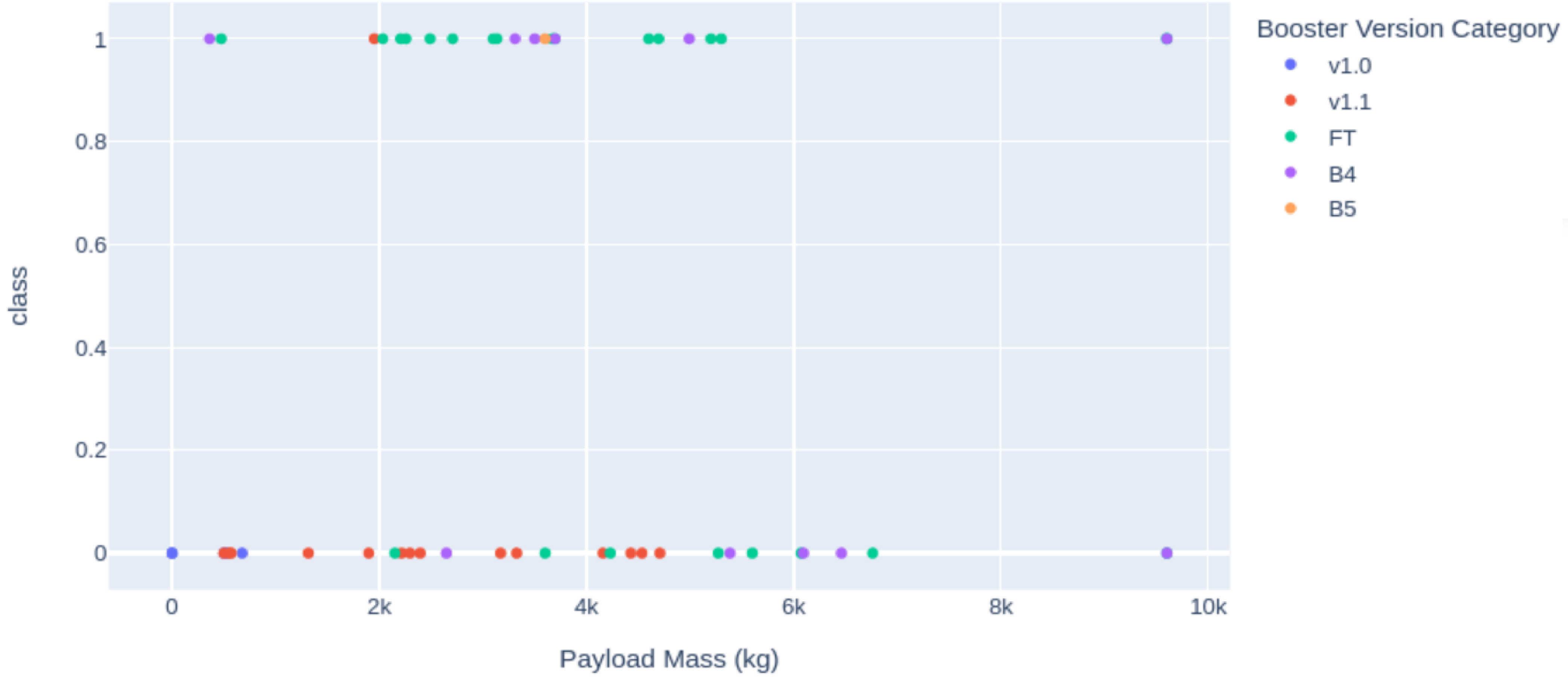


Web Based Visualization using Plotly and Dash

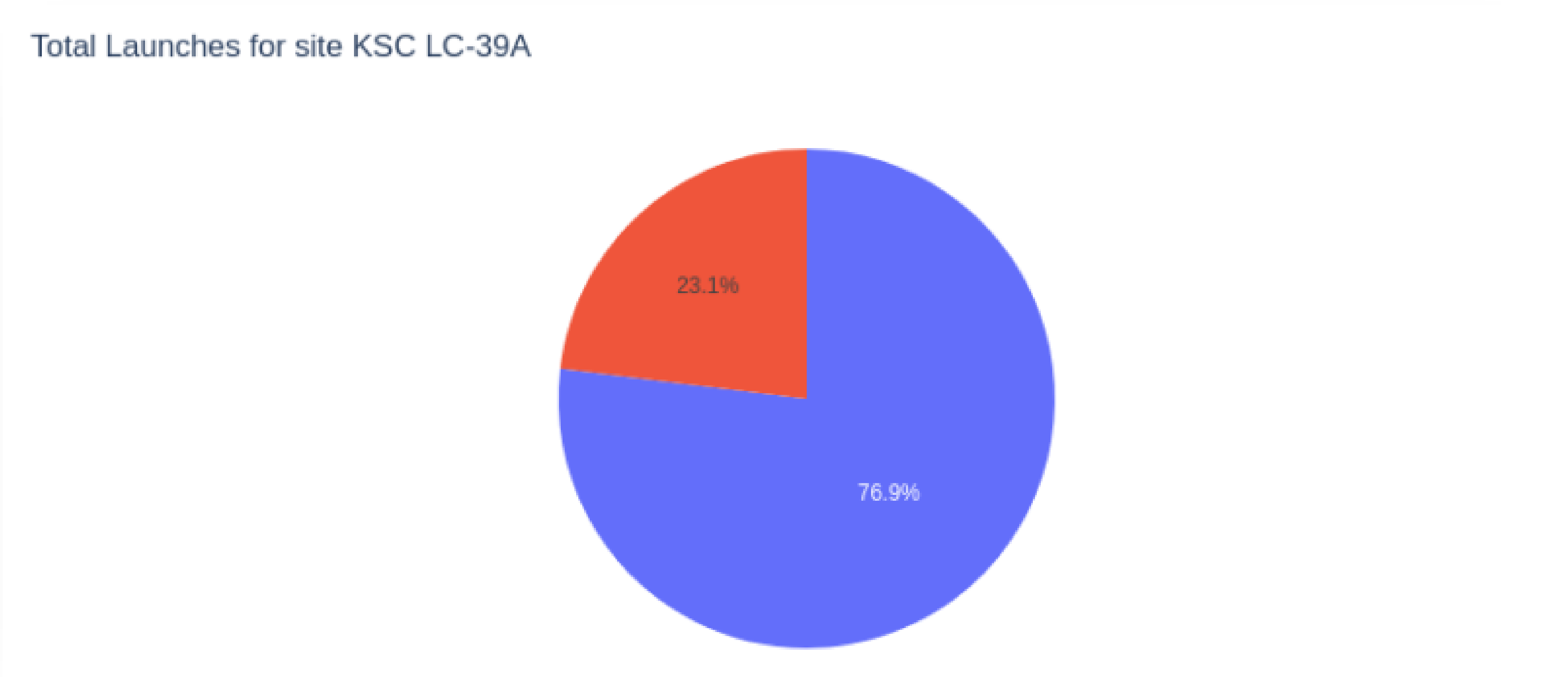


```
    object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
 operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
 _ob.select= 1
 er_ob.select=1
 ntext.scene.objects.active
 ("Selected" + str(modifier
 irror_ob.select = 0
 bpy.context.selected_ob
 ata.objects[one.name].se

 int("please select exactly

 - OPERATOR CLASSES ----

 types.Operator):
  X mirror to the selected
 ject.mirror_mirror_x"
 rror X"
```

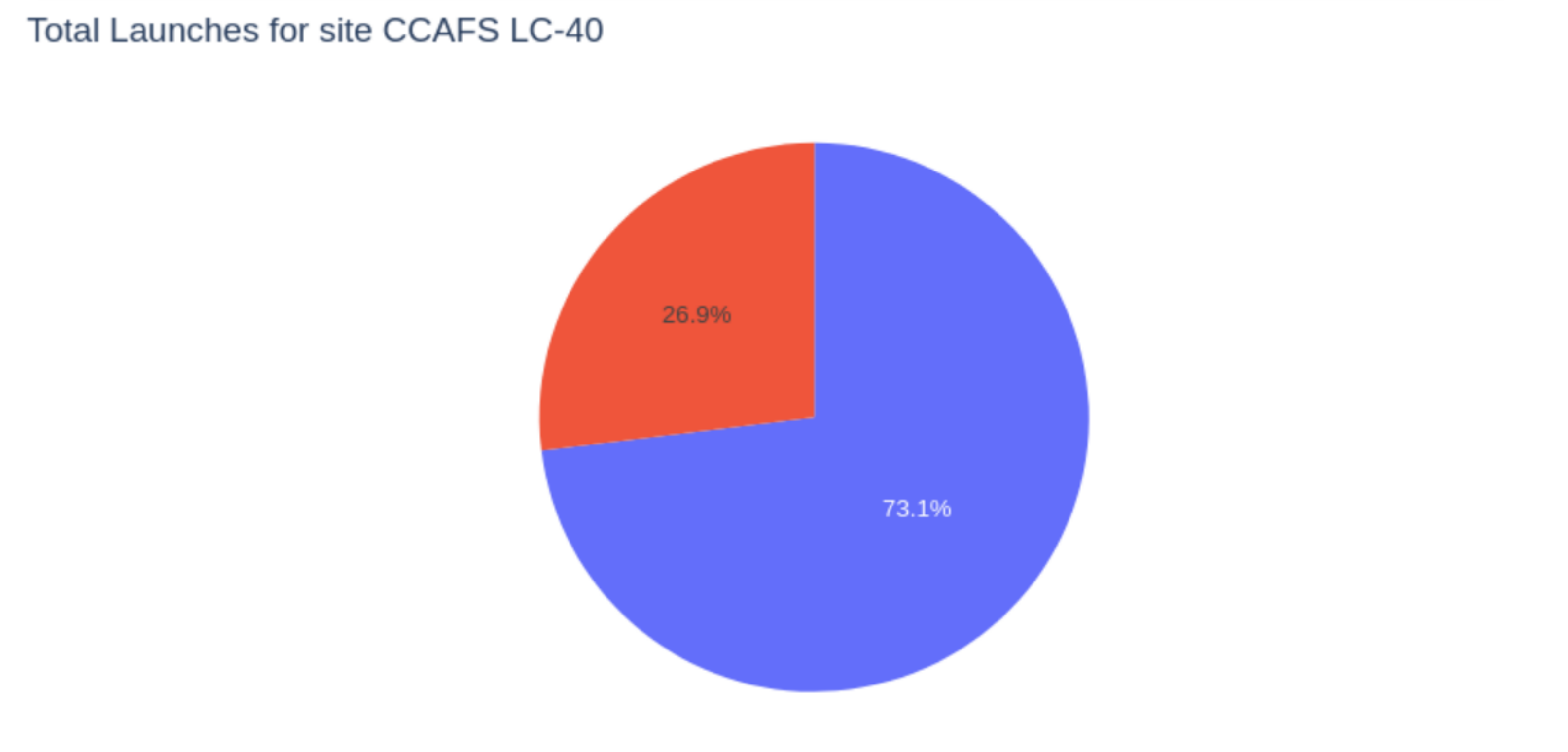

Total Success Launches By Site
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40


Booster Version Category
v1.0
v1.1
FT
B4
B5
class
1
0.8
0.6
0.4
0.2
0
0
2k
4k
6k
8k
10k
Payload Mass (kg)

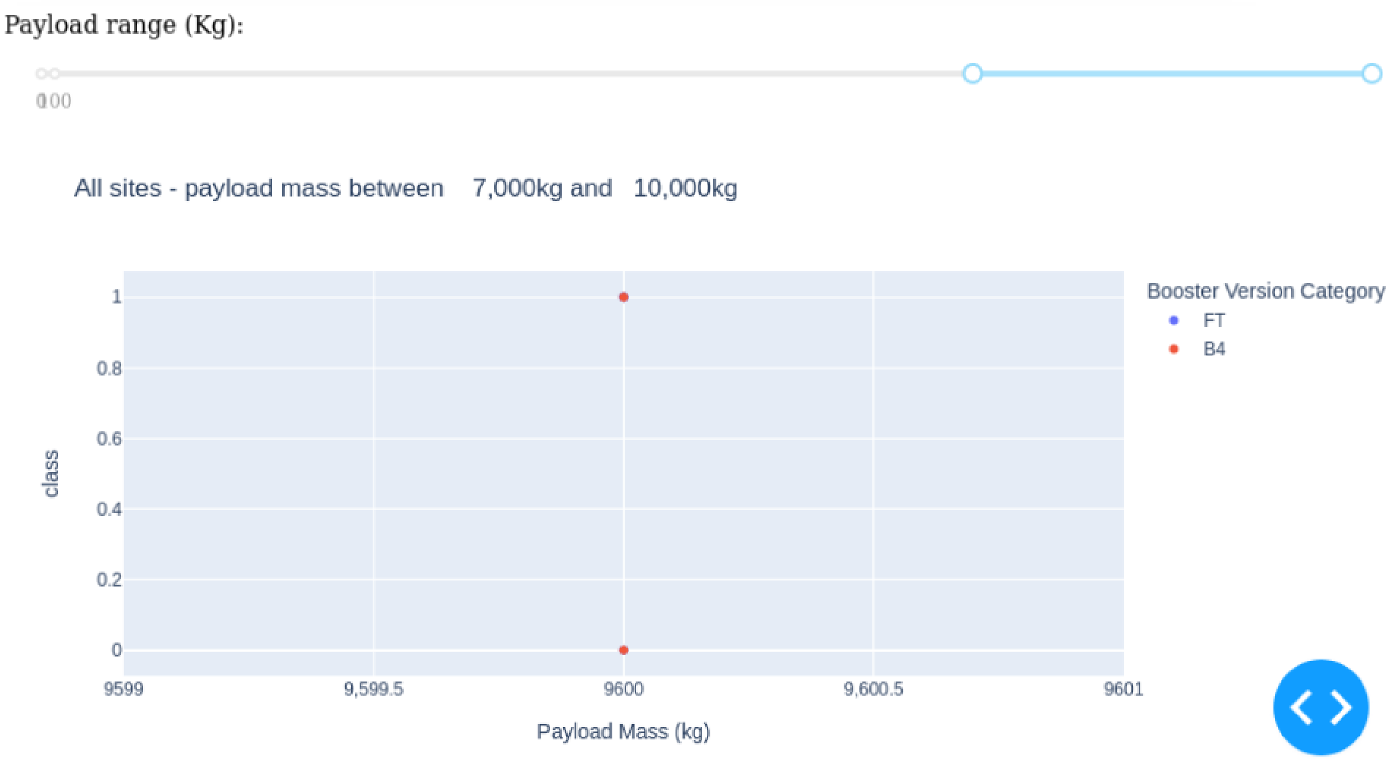
Total Launches By KSC LC-39A



Total Launches By CCAFS LC-40



Payload vs Launch Outcome

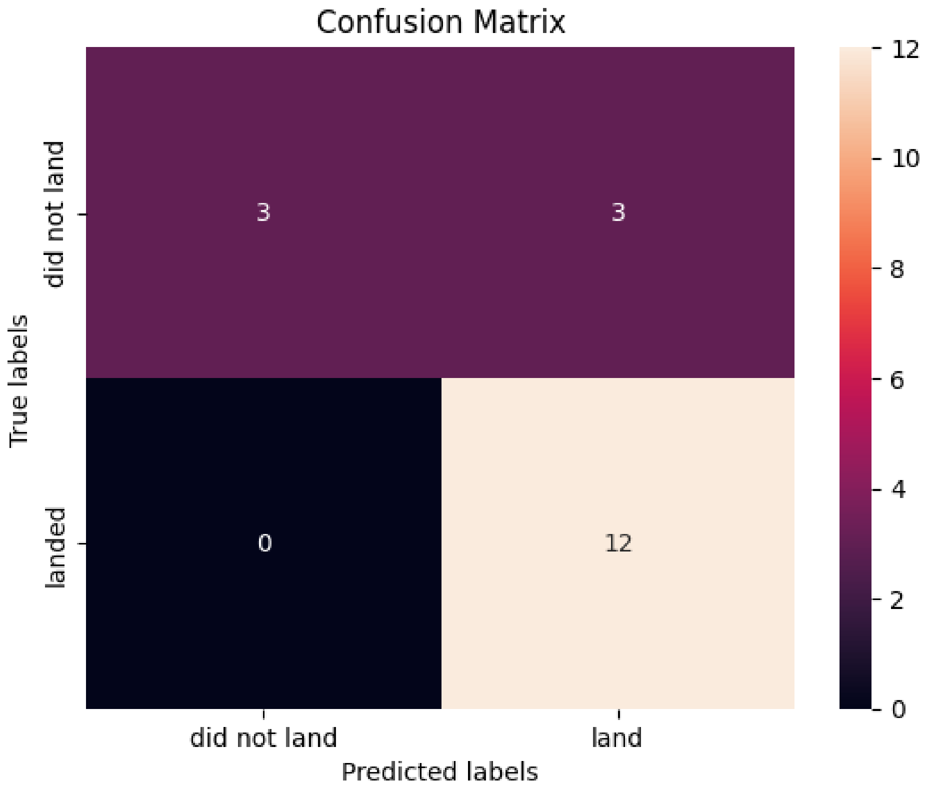


Payload range (Kg):

Results of Predictive Analysis (Classification)

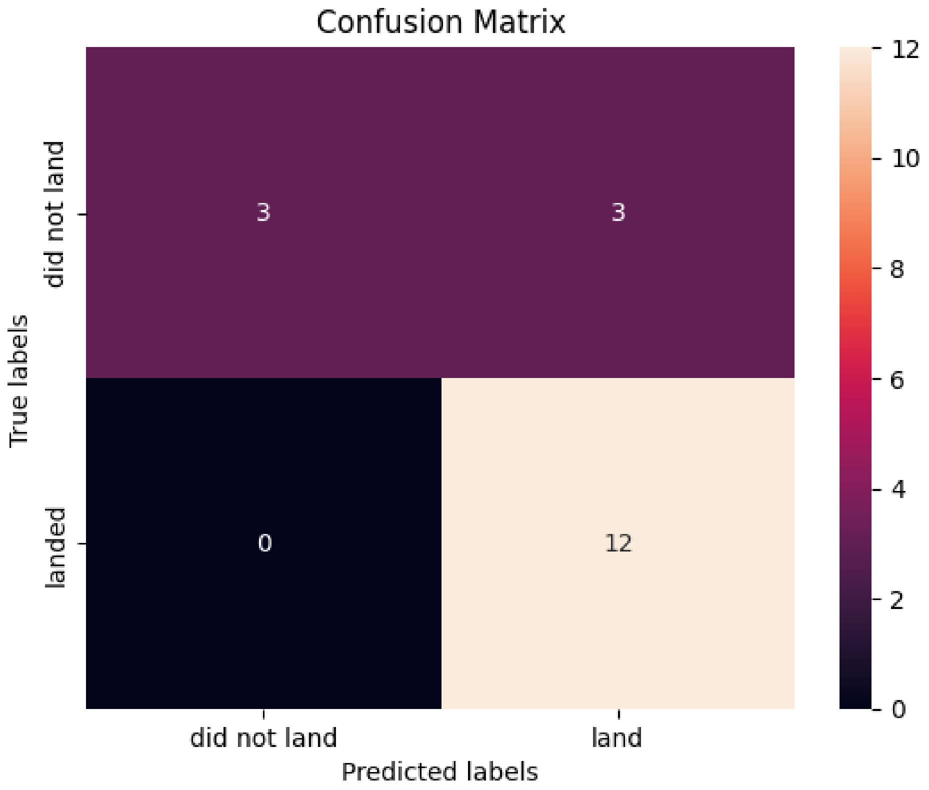
Confusion Matrix

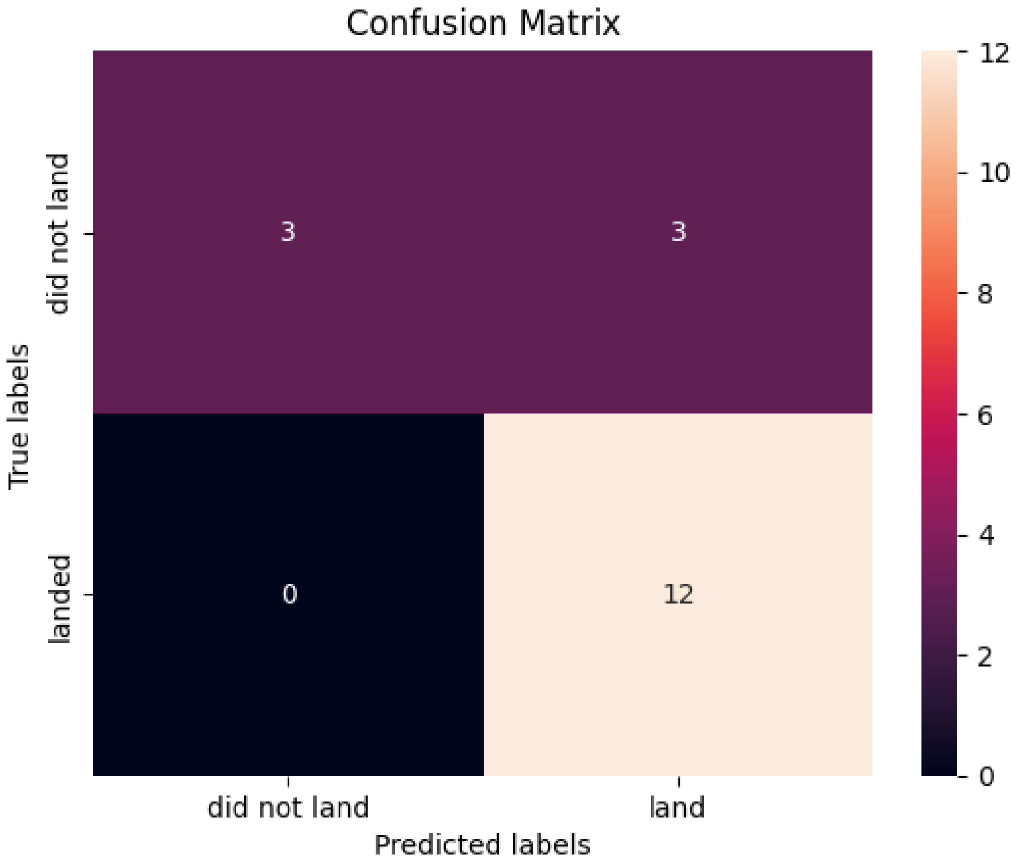
SVM



Confusion Matrix

Decision Tree



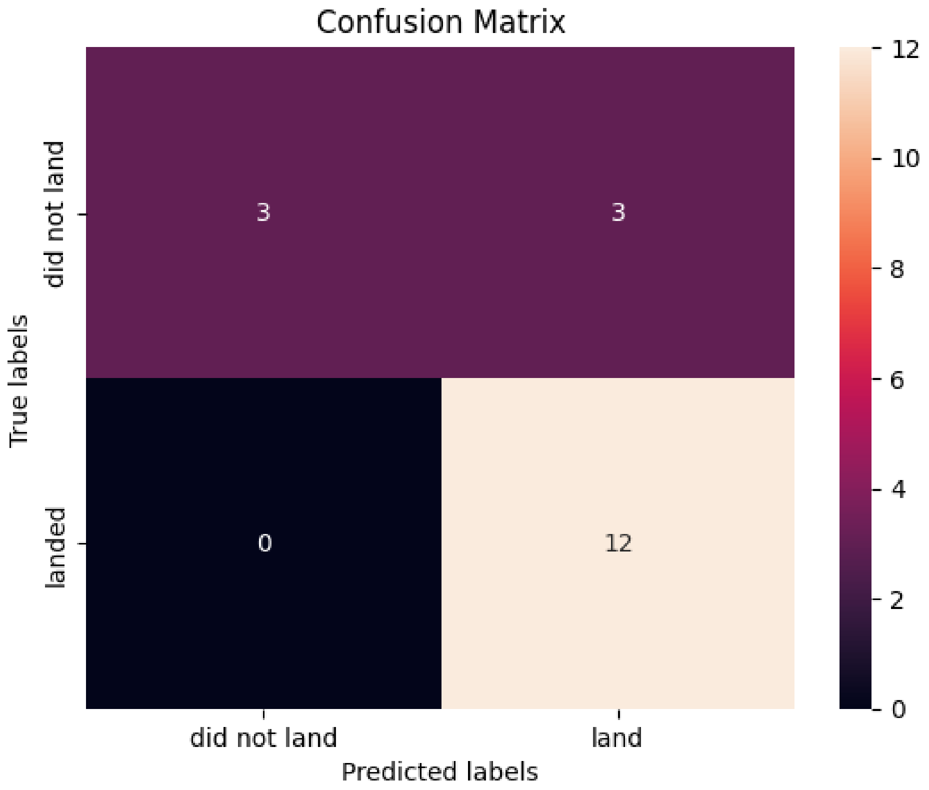


Confusion Matrix

KNN

Confusion Matrix

Logistic Regression



Predictive Analysis Results

THE BEST LAUNCH IS KSC LC-39A

WITH EVOLUTION OF PROCESS AND ROCKETS, SUCCESSFUL LANDING OUTCOMES SEEM TO IMPROVE OVER TIME.

DECISION TREE CLASSIFIER HAS THE MOST ACCURATE RESULTS AROUND 87.7% WHICH CAN BE USED TO PREDICT SUCCESSFUL LANDINGS AND INCREASE PROFITS.

Conclusion

- Decision Tree Model is the best algorithm for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate