IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamed Hafedh Ibrahim
31/08/2022

Outline

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

Executive Summary

- Four classification algorithms will be tested on SpaceX data in order to predict the success of rockets landing.
- The data will be collected and cleaned. Different chart will be displayed to understand the data then the best classifier will be chosen to predict the success of rockets landing.

Introduction

- SpaceX 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- This project aims to predict if the Falcon 9 first stage will land successfully depending on different parameters such as launch site, payload mass, etc.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection Methodology

- Request and parse the SpaceX launch data using the GET request
- Filter the dataframe to only include Falcon 9 launches
- Clean the data

Data Collection – SpaceX API

- Request and parse the SpaceX launch data using the GET request
- Convert the json result into a pandas dataframe

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdc
```

We should see that the request was successfull with the 200 status response code

```
In [10]: response.status_code
Out[10]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas

```
In [30]: # Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/01-Spacex-data-collection-api.ipynb

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, "html.parser")
```

Print the page title to verify if the `BeautifulSoup` object was created properly

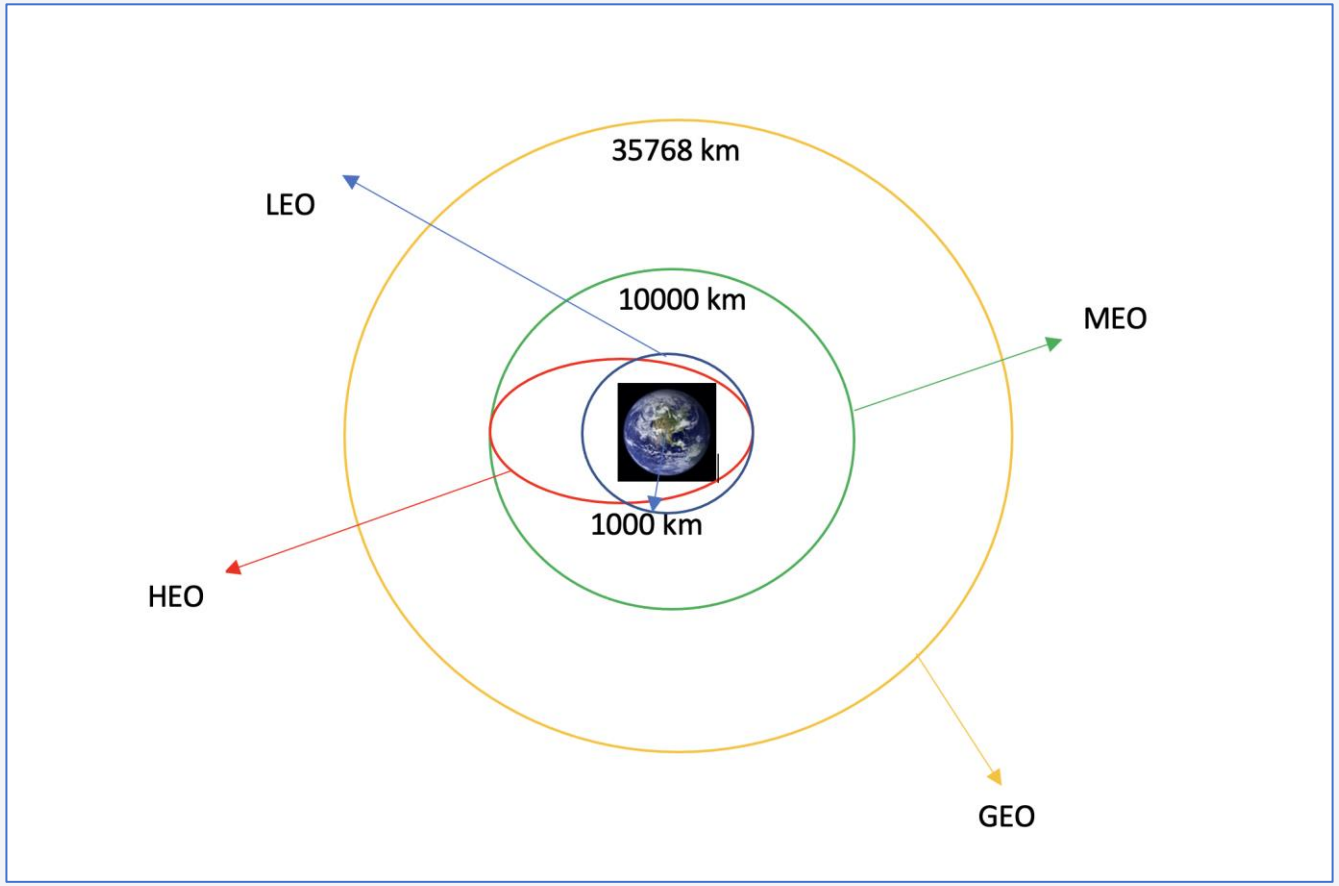
```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/02-WebScraping.ipynb

Data Wrangling

- 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
- Create a landing outcome label from Outcome column



For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/03-Spacex-Data-wrangling.ipynb

EDA with Data Visualization

- Visualize the relationship between Flight Number and Launch Site
- Visualize the relationship between Payload and Launch Site
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between FlightNumber and Orbit type¶
- Visualize the relationship between Payload and Orbit type
- Visualize the launch success yearly trend

For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/05-Spacex-DataVisualization.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass.
- List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/04-Spacex-sql-queries.ipynb

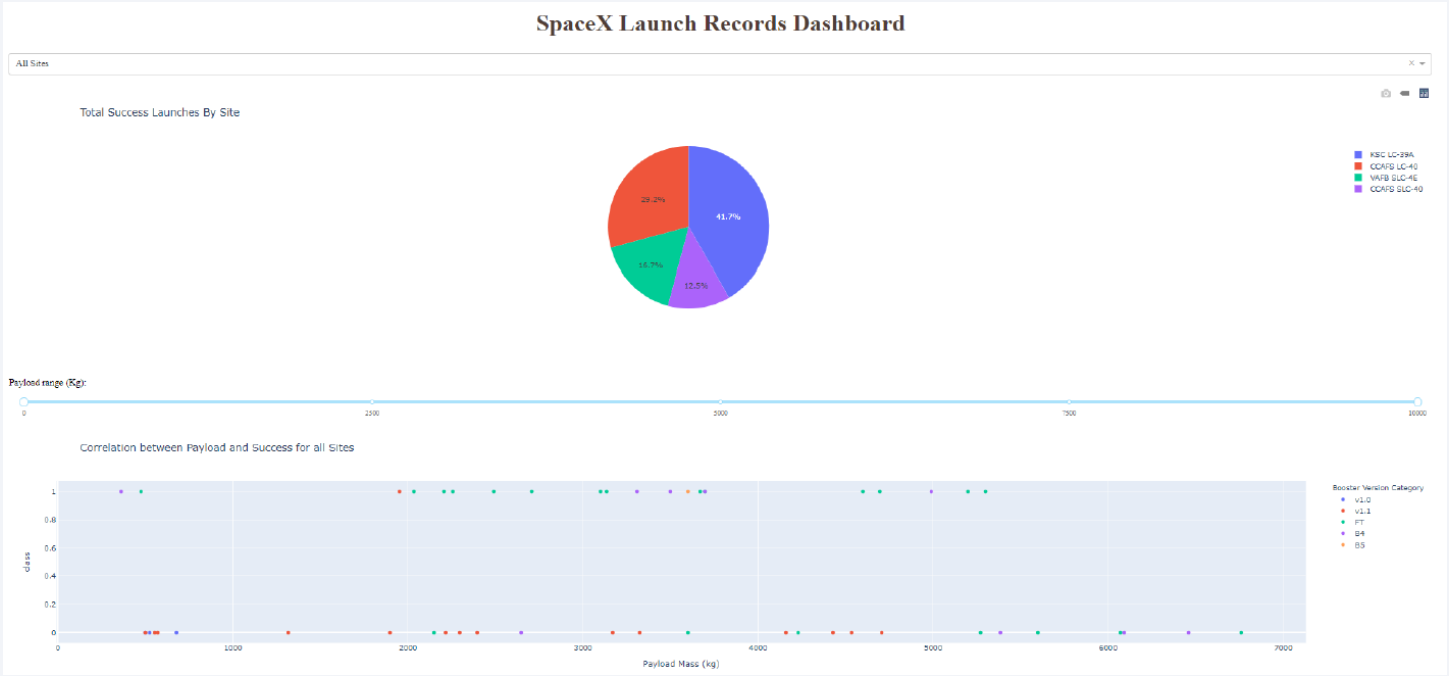
Build an Interactive Map with Folium

- Mark 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

For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/06-Spacex-launch-site-location.ipynb

Build a Dashboard with Plotly Dash

- The objective is to perform an interactive visual analytics on SpaceX launch data in real-time.
- The Plotly Dash application is composed of two main divisions:
  - ✓ A dynamic pie chart of total success launches by site,
  - ✓ A dynamic scatter plot of correlation between Payload and success for all sites



For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/spacex_dash_app.py

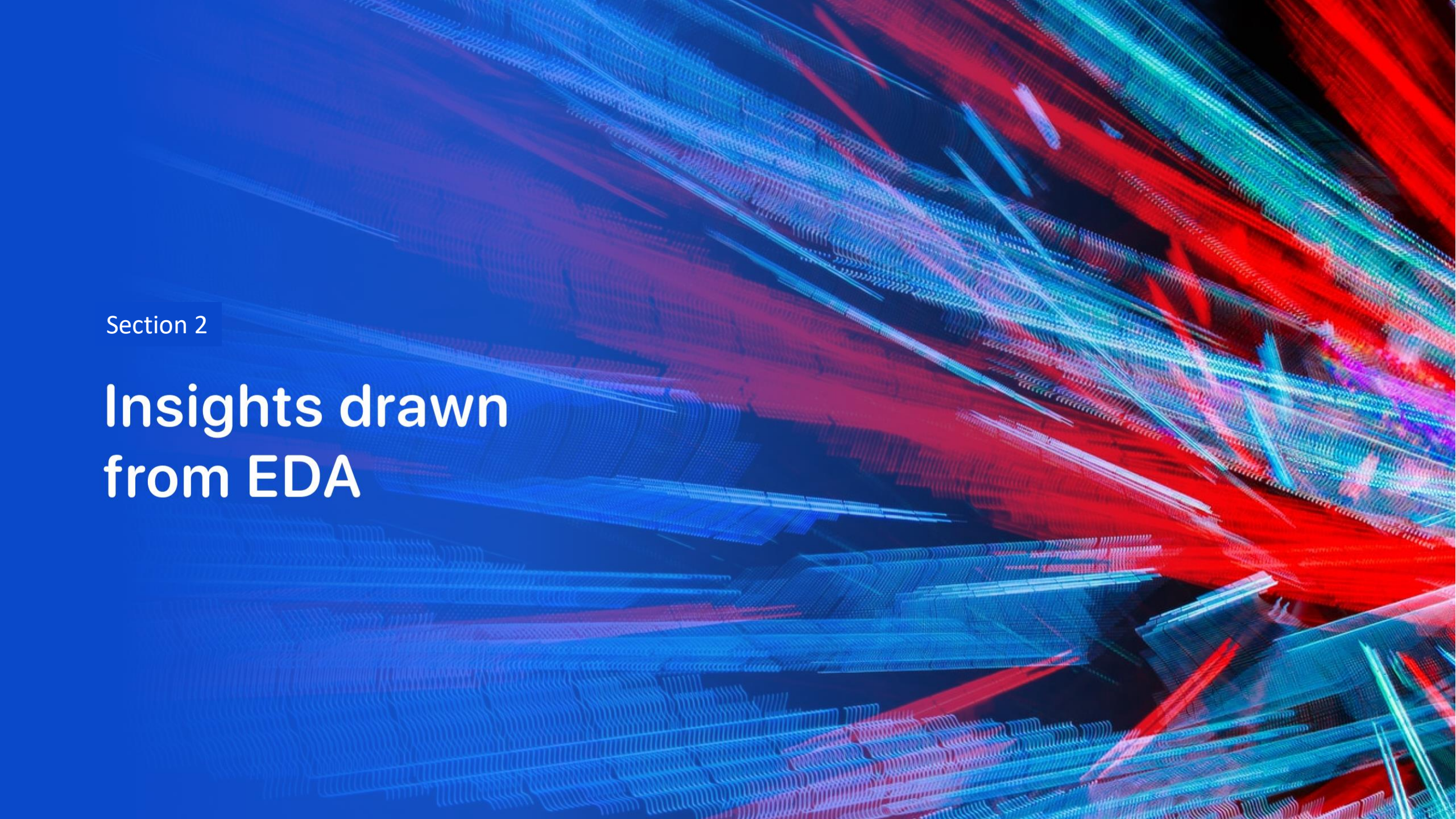
Predictive Analysis (Classification)

- Four classification algorithms were used to find the best classifier of landing success
- K-Nearest Neighbors, Decision Tree, SVM and Logistic regression models were created, trained and evaluated.
- Accuracy scores were compared in order to find the best classifier

For more details: https://github.com/medhibrahim/IBMCapstoneProject/blob/main/07-SpaceX-Machine-Learning-Prediction.ipynb

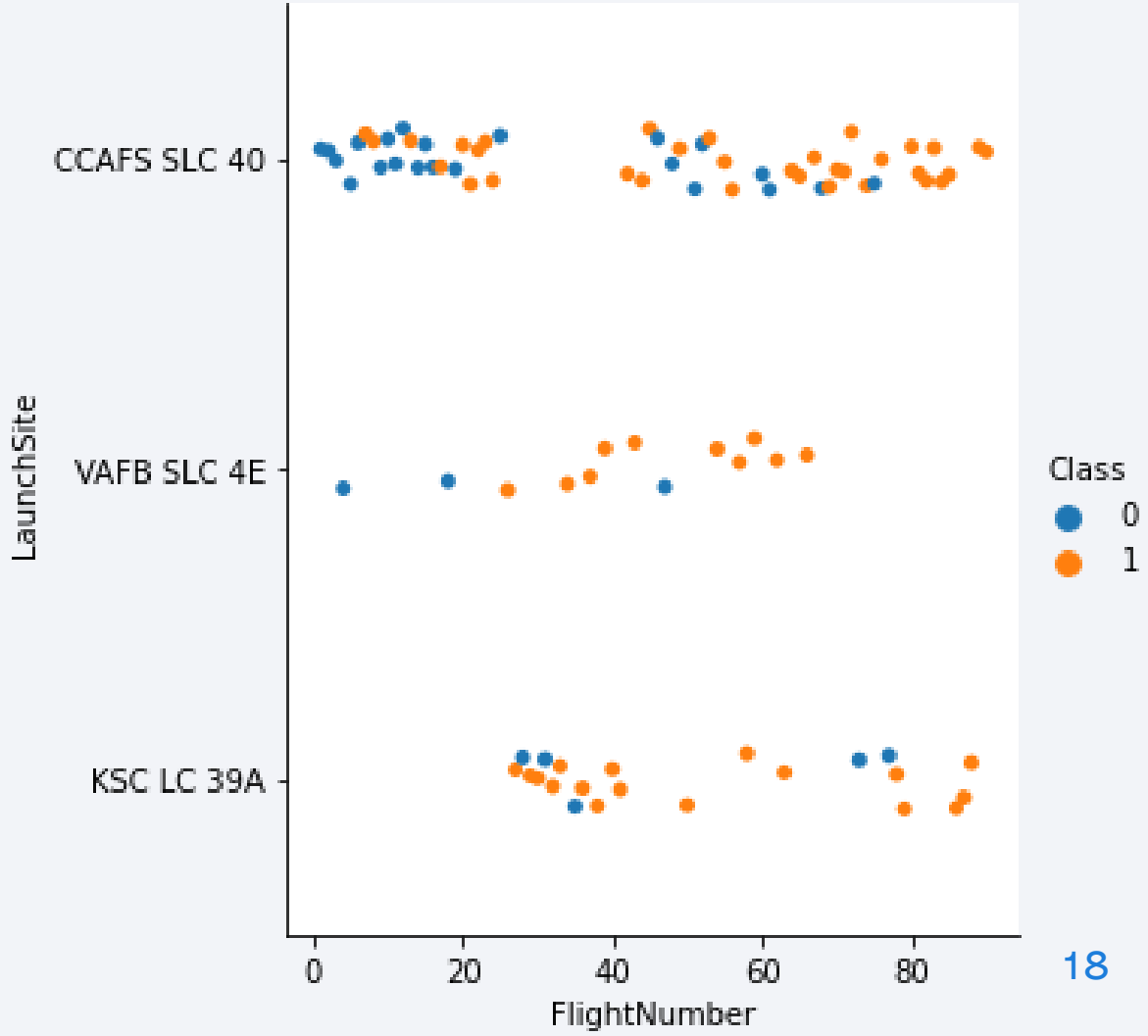
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

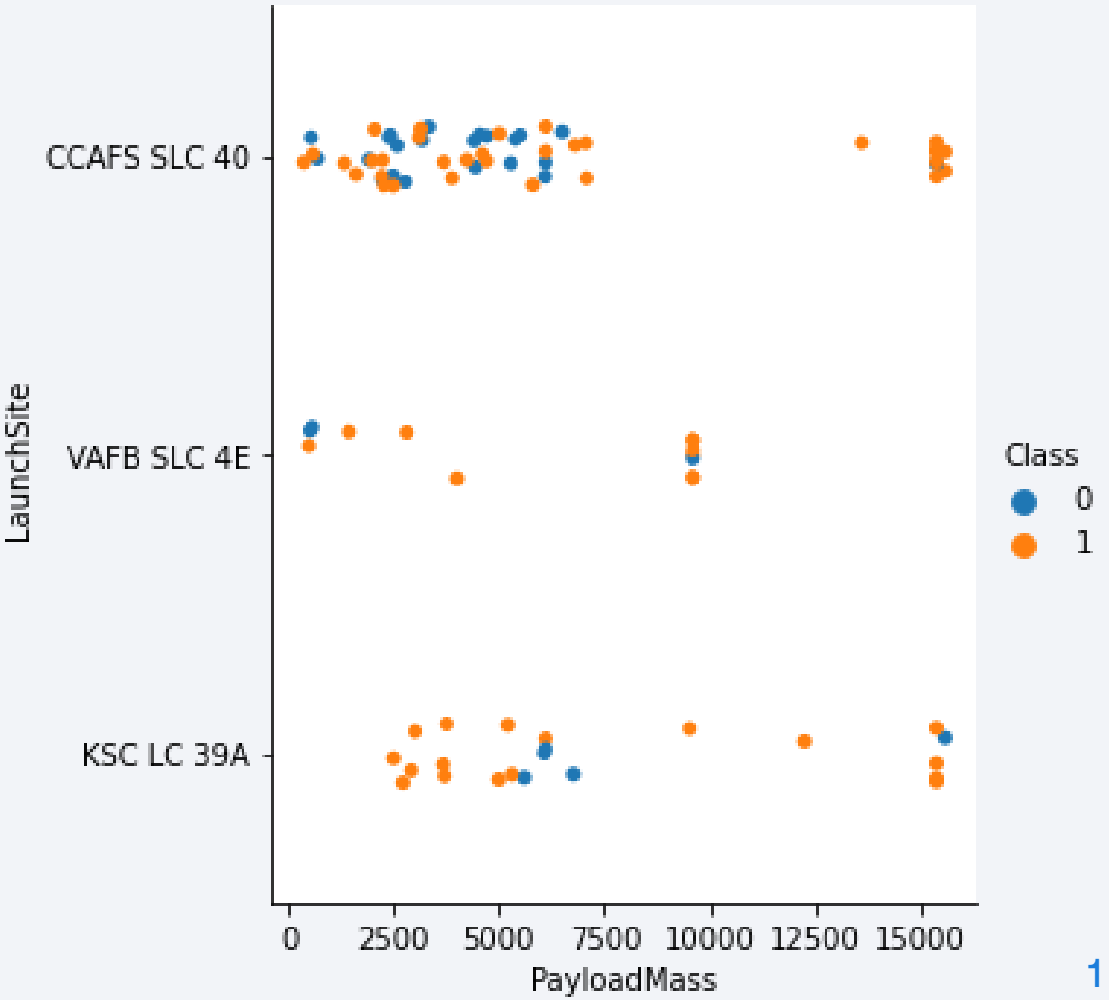
Flight Number vs. Launch Site

- The first thing we can notice is that CCAFS SLC 40 is the launch site of most flights.
- All the flights with flight number >= 80 are a success.
- Most flights that are launched from VAFB SLC 4E are a success.



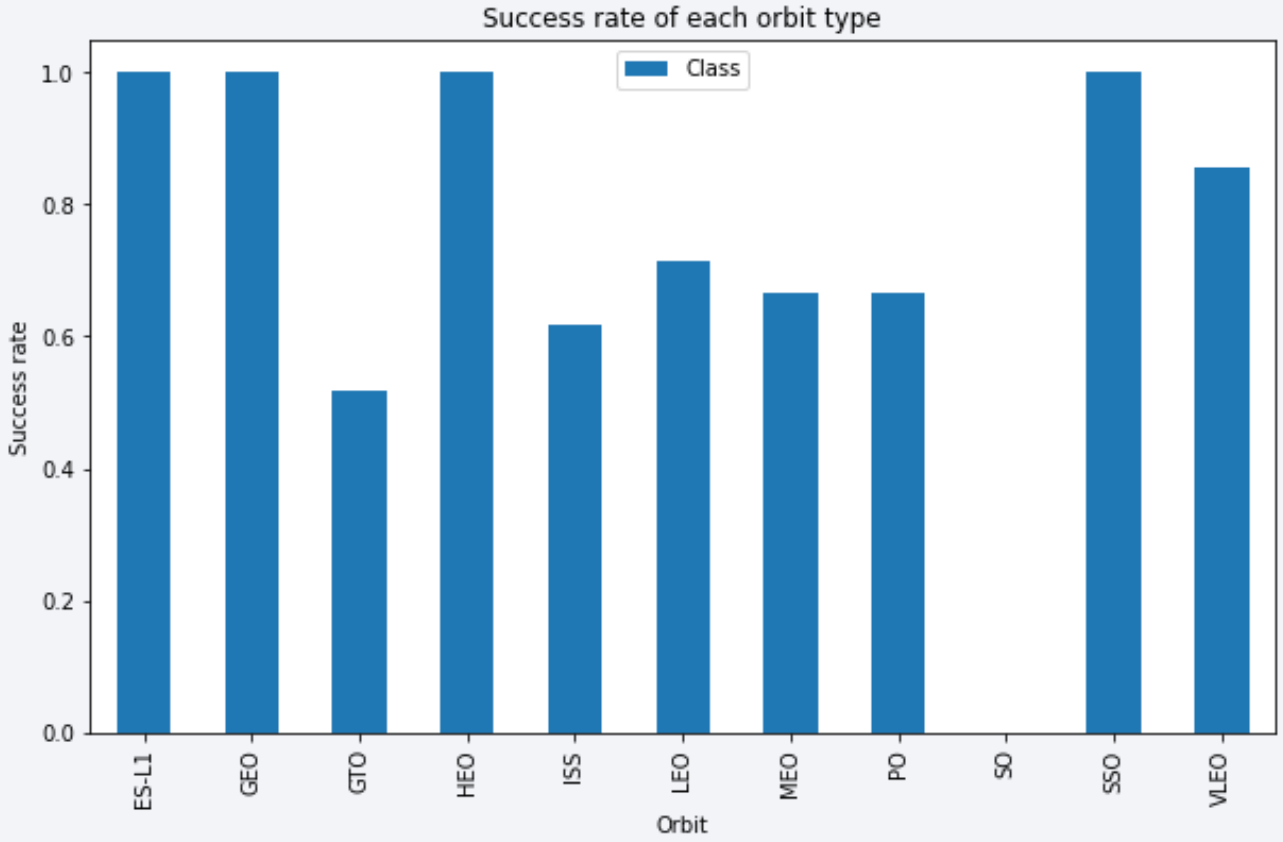
Payload vs. Launch Site

- Most flights have a payload mass <8000 kg.
- For VAFN SLC 4E launch site, there are not rockets launched for heavy payload (greater than 10000 kg).
- Most flights with heavy payload mass (greater than 8000 kg) are a success.



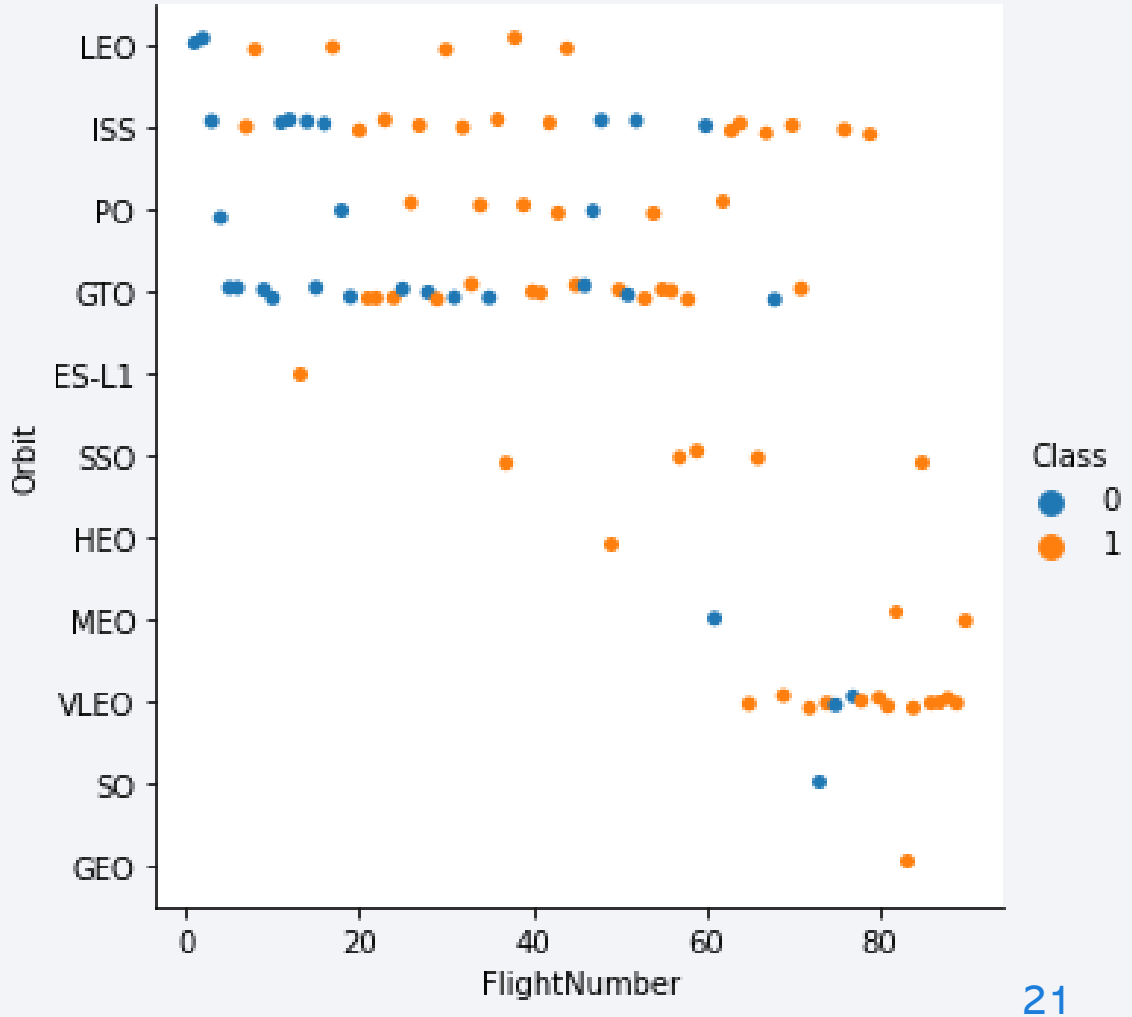
Success Rate vs. Orbit Type

- The orbits ES-L1, GEO, HEO and SSO have a success rate of 100% while the flights targeting SO orbit never succeeded.



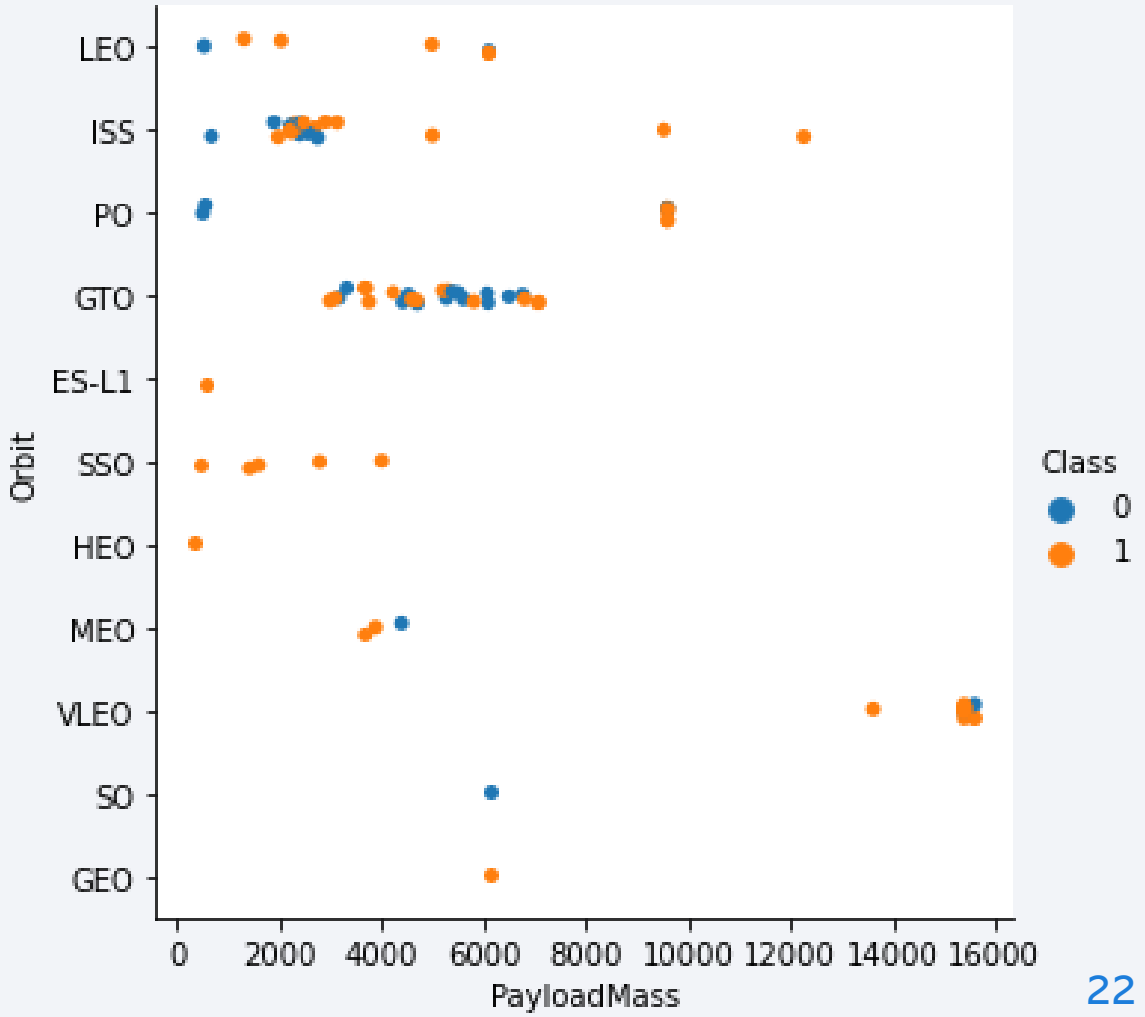
Flight Number vs. Orbit Type

- The recent flights are essentially targeting the orbit VLEO.
- The flights with number > 80 are a success.
- Most of flights are targeting the orbits LEO, ISS, PO, GTO and VLEO.



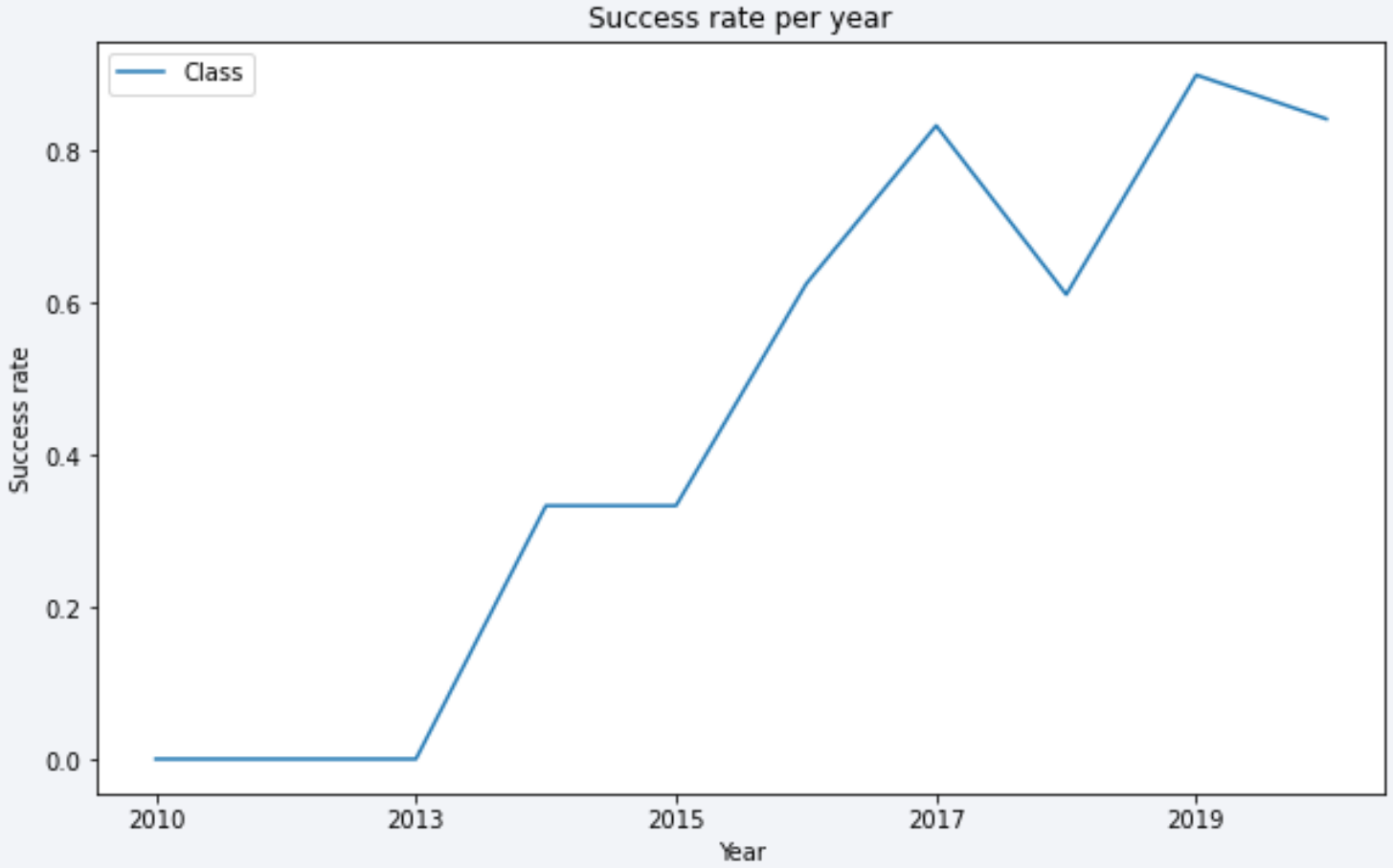
Payload vs. Orbit Type

- The recent flights are essentially targeting the orbit VLEO.
- The flights with number > 80 are a success.
- Most of flights are targeting the orbits LEO, ISS, PO, GTO and VLEO.



Launch Success Yearly Trend

Sucess rate since 2013 kept increasing till 2020



All Launch Site Names

```
In [17]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
```

Out[17]:

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

- We should add DISTINCT in order to have unique names of the different launch sites.

Launch Site Names Begin with 'CCA'

- Adding LIKE “CCA%” enables to have launch site that begin with ‘CCA’
- LIMIT 5 : limits the selection to 5 sites

In [22]:
```
%%sql
SELECT * FROM SPACEXTBL
WHERE Launch_Site LIKE "CCA%" LIMIT 5;
```

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

Out[22]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We select the sum of payload mass with the condition that the customer is “NASA (CRS)”.
- The total payload carried by boosters from NASA (CRS) is 45596 kg.

```
In [23]: %%sql
         SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL
         WHERE Customer = "NASA (CRS)";
          * sqlite:///my_data1.db
         Done.
Out[23]: SUM(PAYLOAD_MASS__KG_)
                            45596
```

Average Payload Mass by F9 v1.1

```
In [37]: %%sql
         SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
         WHERE Booster_Version = "F9 v1.1";

          * sqlite:///my_data1.db
         Done.

Out[37]: AVG(PAYLOAD_MASS__KG_)
                        2928.4
```

- We select the average of payload mass with the condition that Booster version is “F9 v1.1”.
- The average payload carried by booster version “F9 v1.1” is 2928.4 kg.

First Successful Ground Landing Date

In [46]:
```
%%sql
SELECT MIN(Date) FROM SPACEXTBL
WHERE "Landing _Outcome" == "Success (ground pad)";
```

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

Out[46]:

MIN(Date)
01-05-2017

- We select the first date by selecting the minimum of dates.
- As “Landing _Ouctome” is a complex name, we should put it between “”.

Successful Drone Ship Landing with Payload between 4000 and 6000

- We select the booster version with the required conditions. We can use BETWEEN for the lower and upper limits of the payload_mass.

In [50]:
```
%%sql
SELECT Booster_Version FROM SPACEXTBL
WHERE "Landing _Outcome" == "Success (drone ship)"
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Out[50]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We use GROUP BY in order to group the count of results by the mission_outcome.
- We notice that Success is displayed twice as apparently there is a space added to "Success" for one row in the database. That's why it was not grouped with the others.

In [84]:
```
%%sql
SELECT Mission_Outcome, COUNT(*) AS "Number of missions" FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

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

Out[84]:

Mission_Outcome	Number of missions
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We need a nested query in order to have the names of booster which have carried the maximum payload mass.

```
In [56]: %%sql
SELECT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
 * sqlite:///my_data1.db
Done.
```

Out[56]:

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

2015 Launch Records

In [59]:
```
%%sql
SELECT substr(Date, 4, 2) as month, "Landing _Outcome", Booster_Version, Launch_Site
FROM SPACEXTBL
WHERE "Landing _Outcome" == "Failure (drone ship)"
AND substr(Date,7,4)='2015'
```

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

Out[59]:

month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Substr enables to extract the month and/or the year from the date which enables to have the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

```
In [82]: %%sql
SELECT "Landing _Outcome", COUNT(*)
FROM SPACEXTBL
WHERE Date BETWEEN "04-06-2010" AND "20-03-2017"
AND "Landing _Outcome" LIKE "%Success%"
GROUP BY "Landing _Outcome"
ORDER BY COUNT(*) DESC

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

Out[82]:

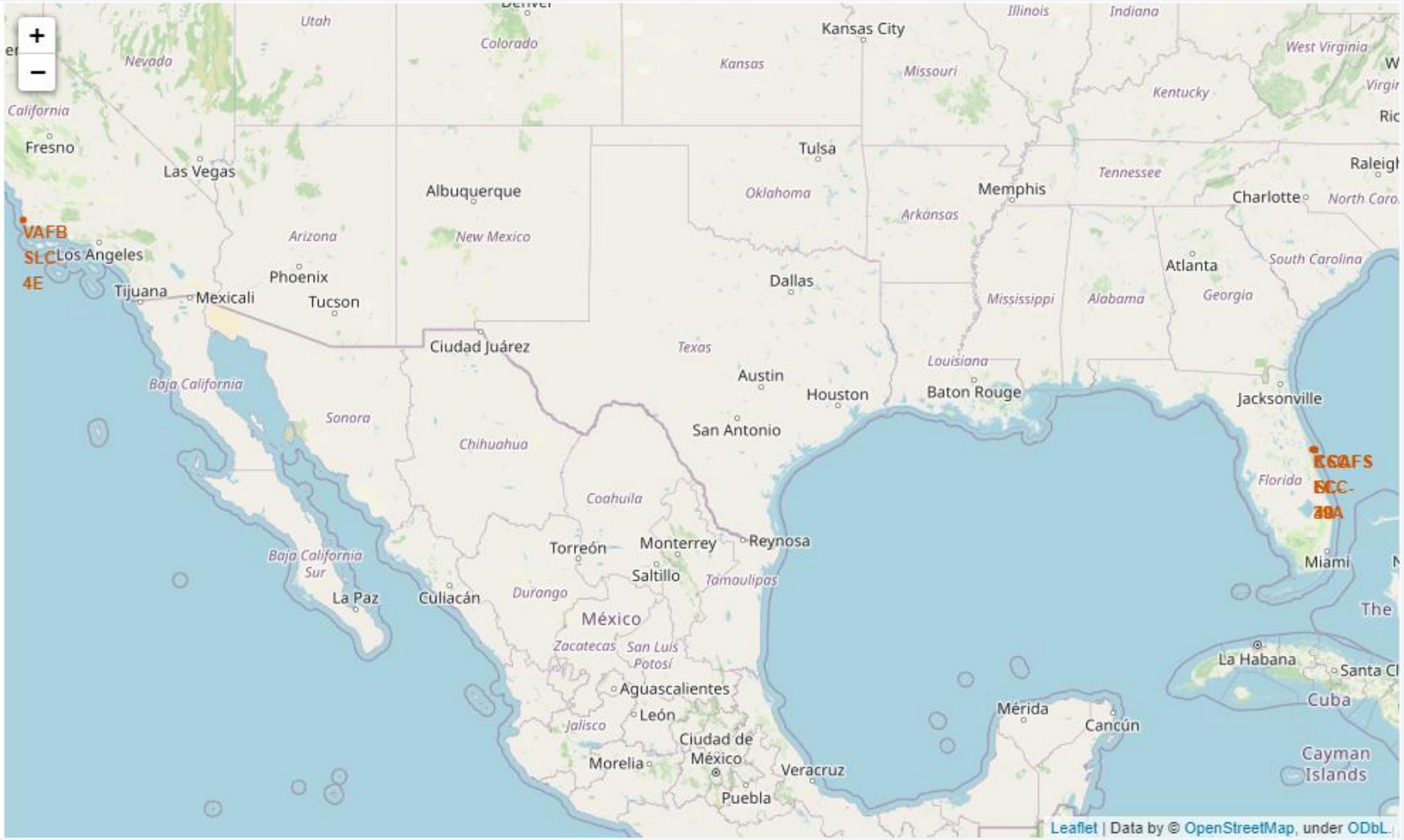
Landing _Outcome	COUNT(*)
Success	20
Success (drone ship)	8
Success (ground pad)	6

- In this query we should use 2 conditions, a GROUP BY to group by “Landing _Outcome” and an “ORDER BY” to order the result by descending order.


Section 3
Launch Sites
Proximities Analysis

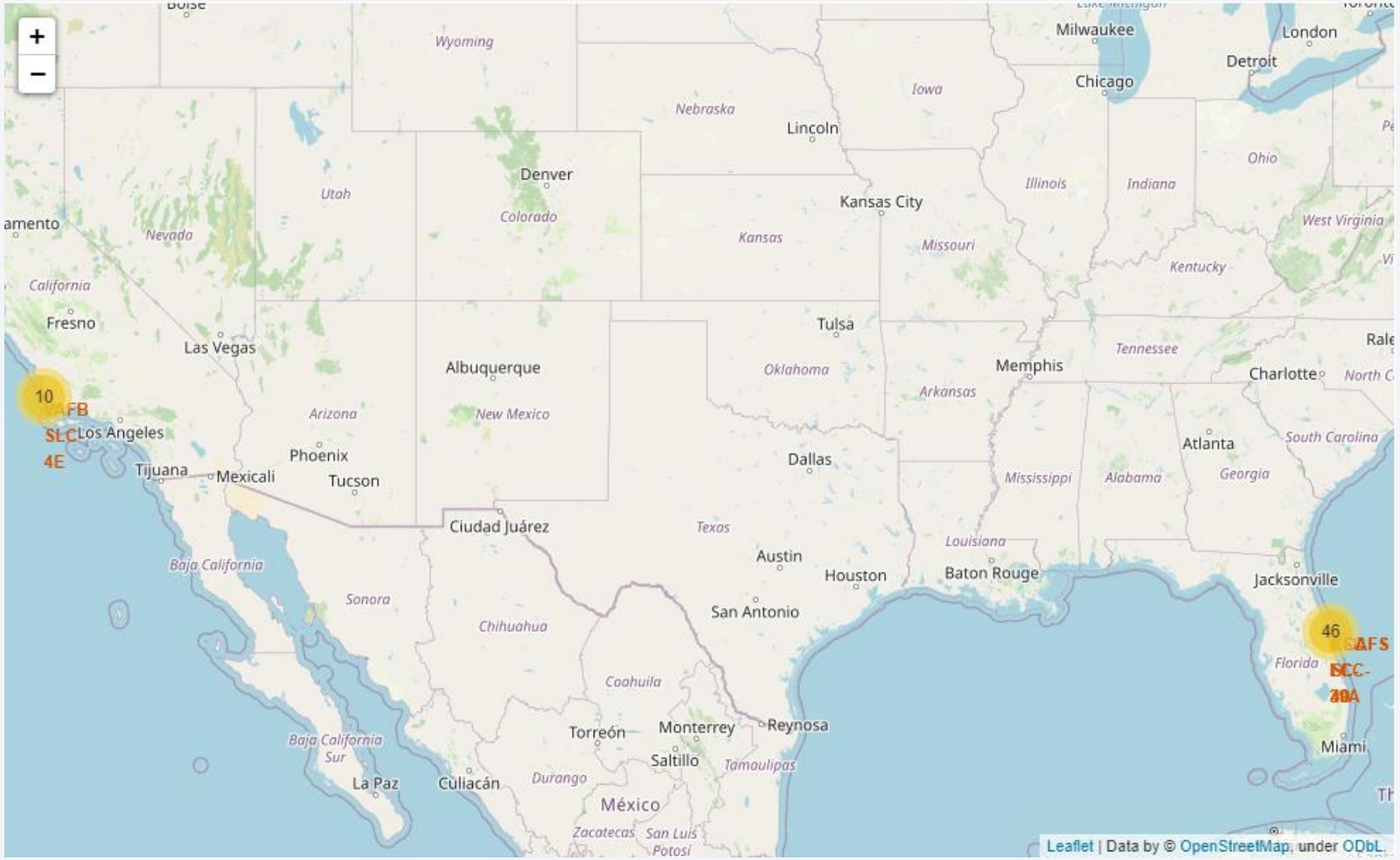
Launch sites' location

The launch sites are located in coastal areas either at the east or the west of the US.

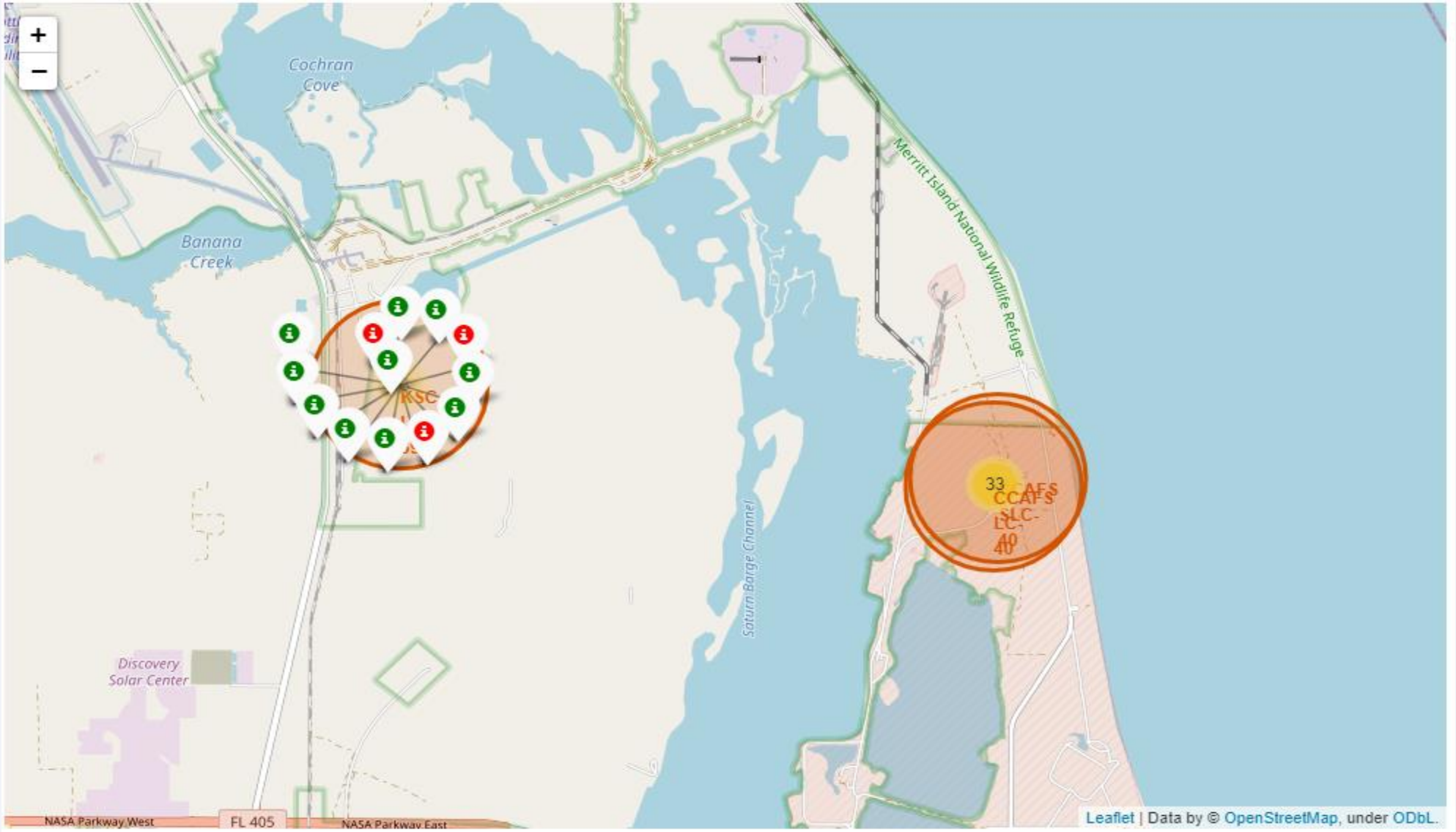


Success/failed launches for each site

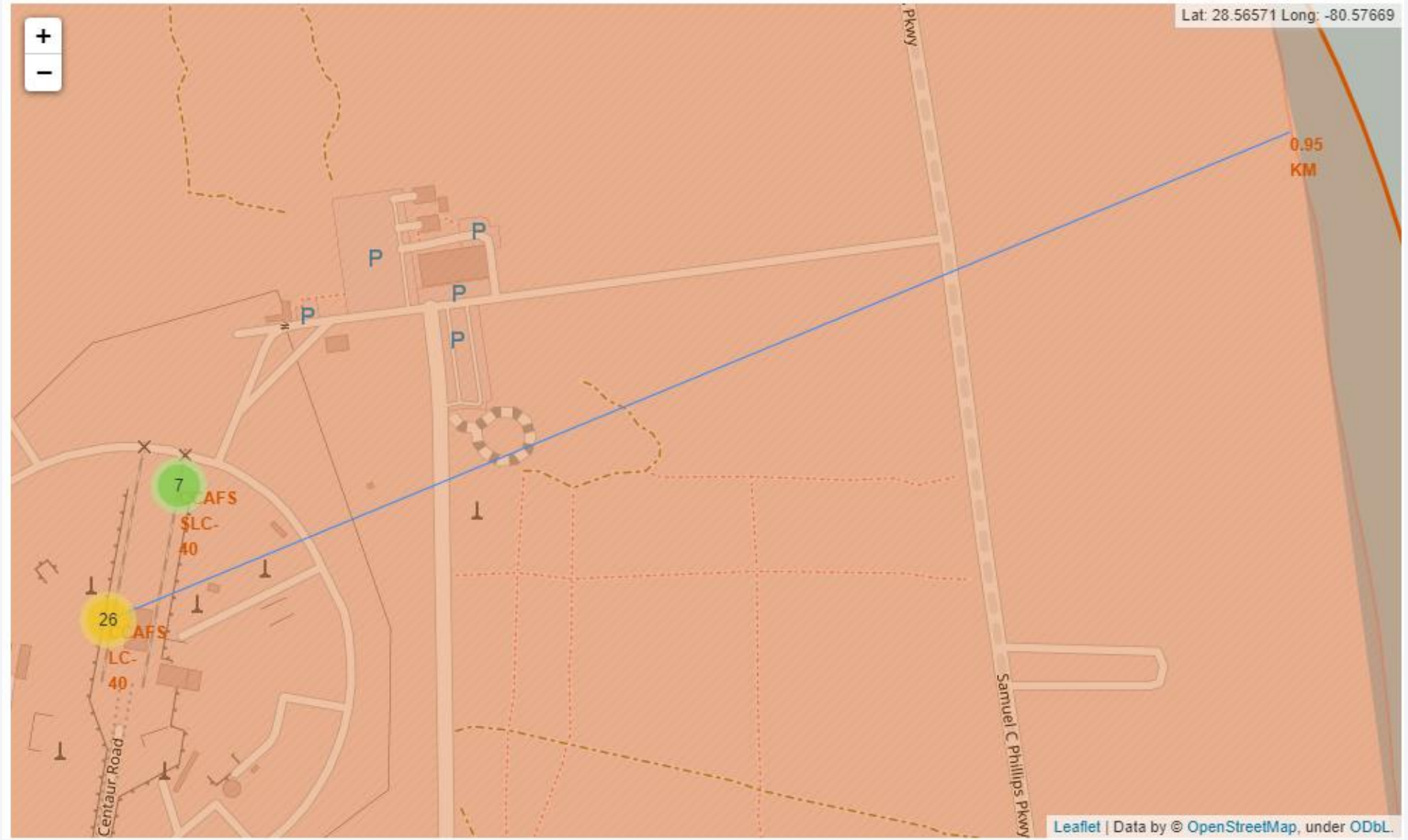
The number of launches per launch site is displayed in this map. A zoom will be displayed in the next map where we can see the success (in green)/ failed (in red) launches for each site



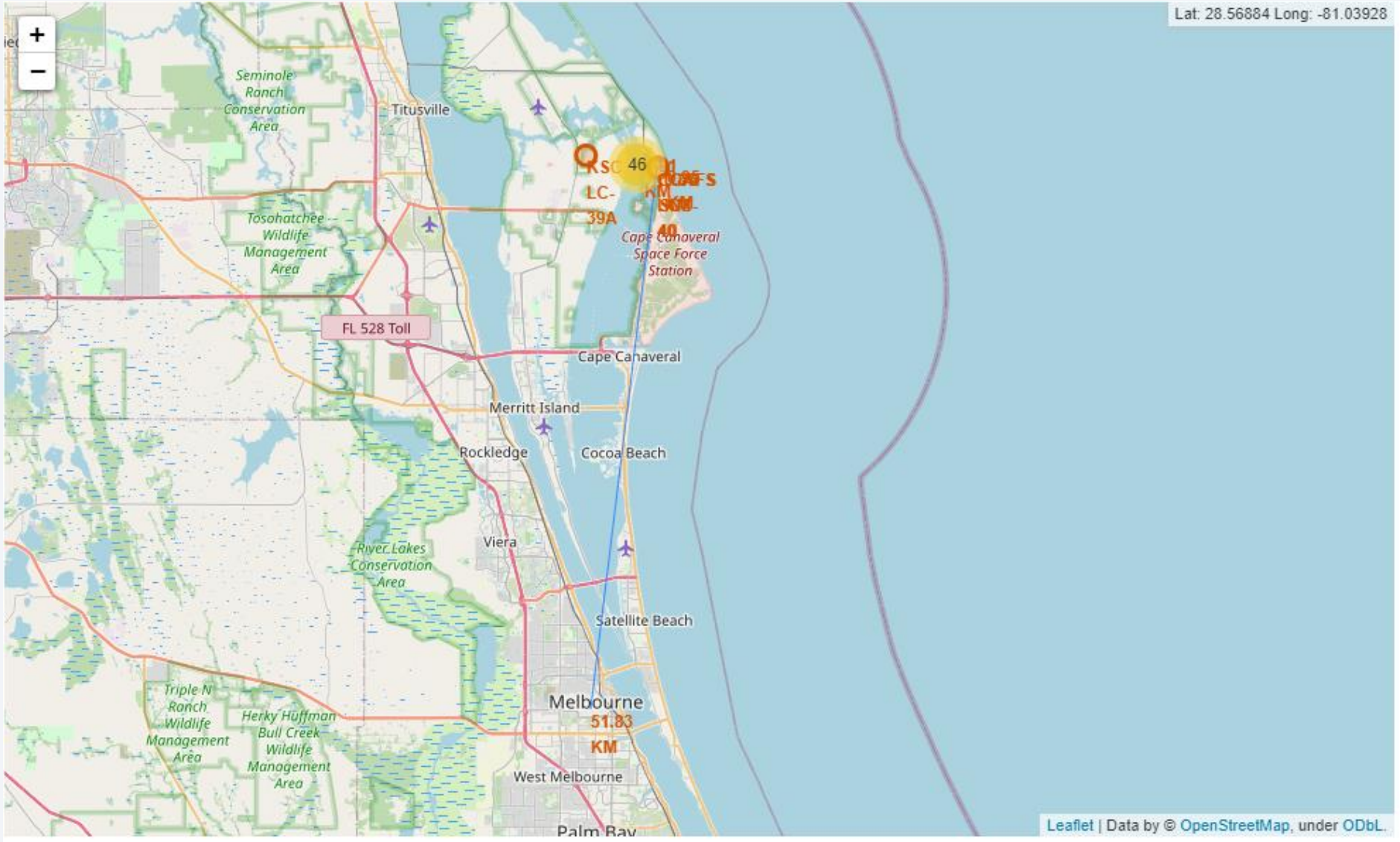
Success/failed launches for each site



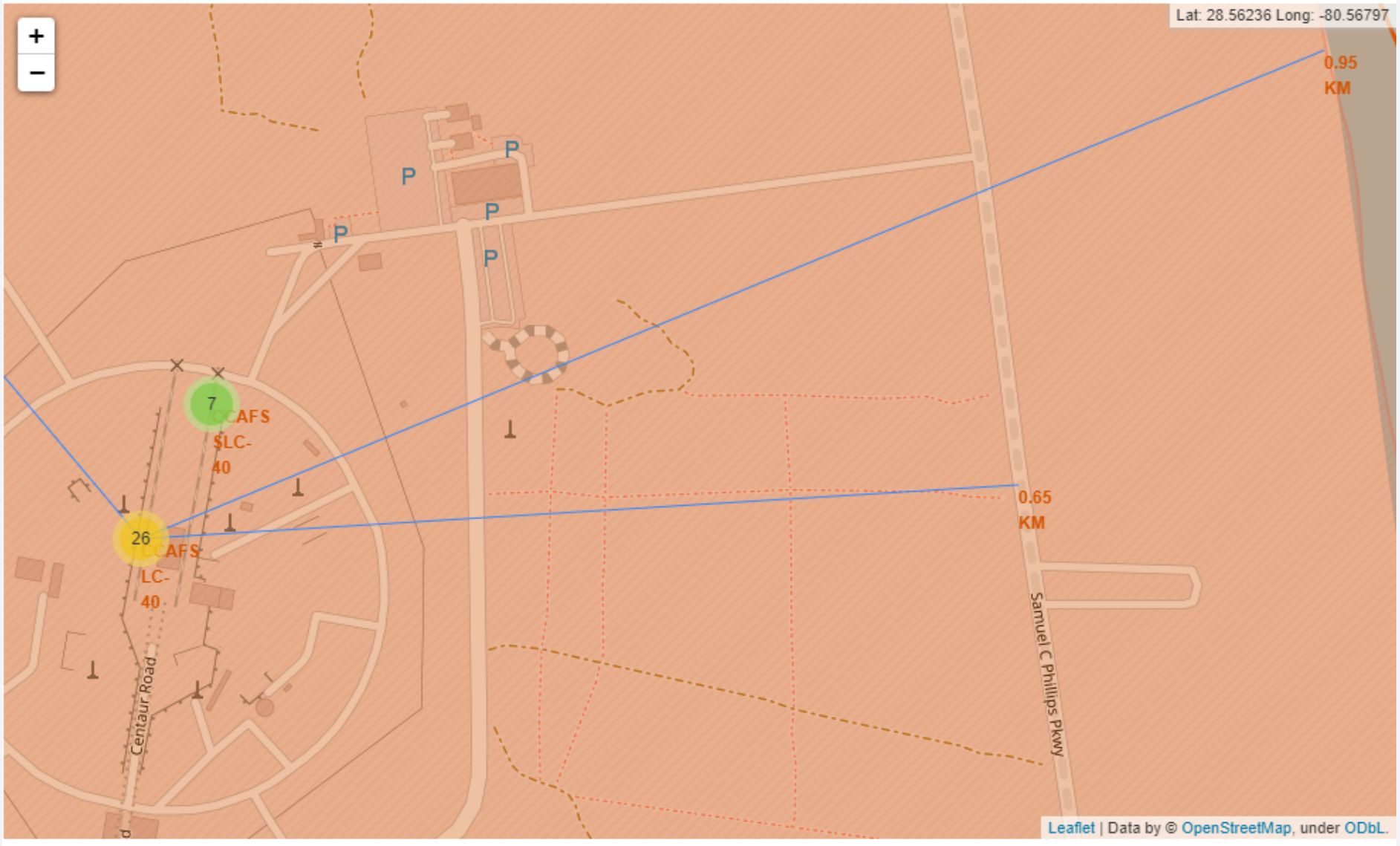
Distances CCAFS LC-40 and nearest coastline



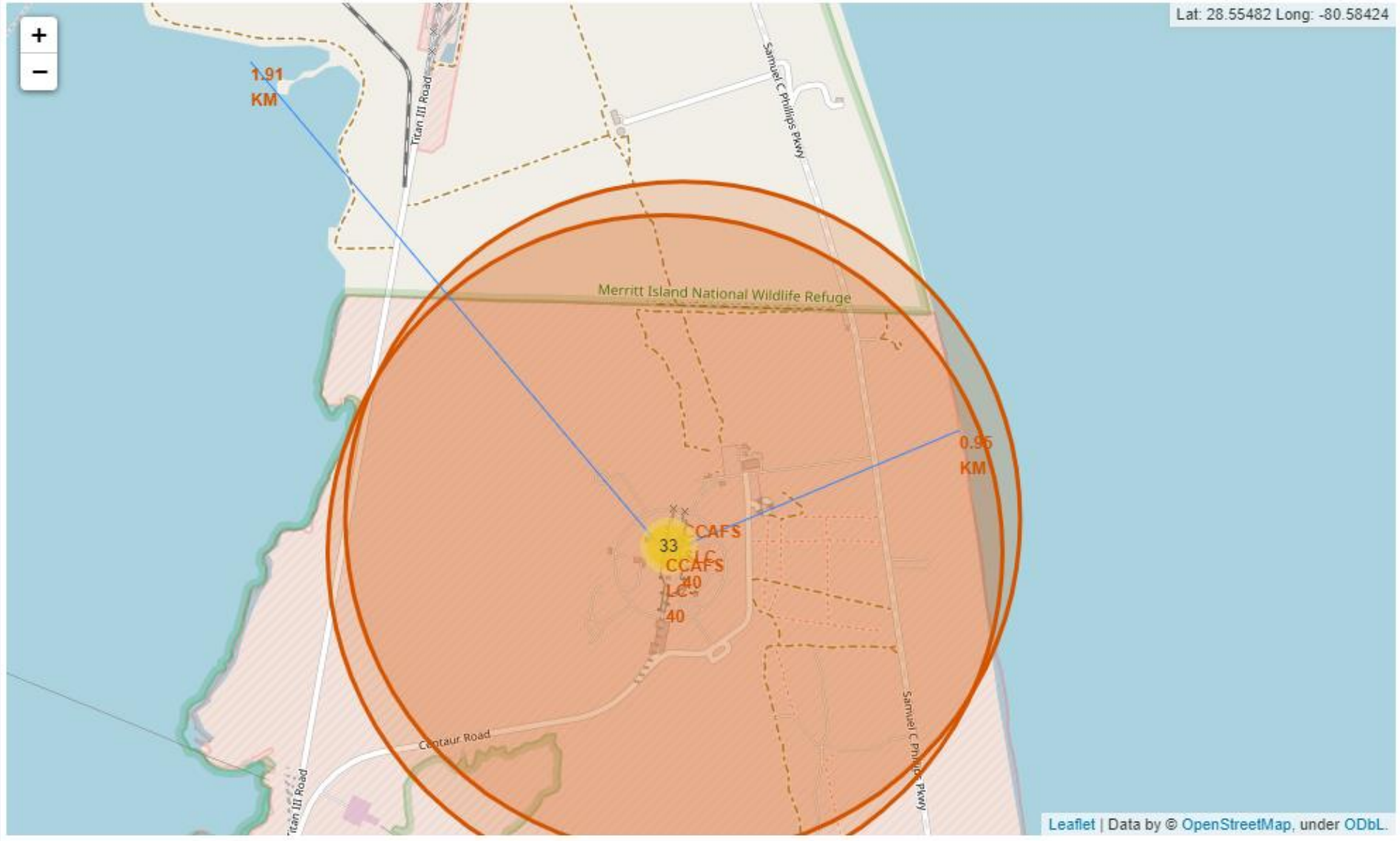
Distances CCAFS LC-40 and nearest city

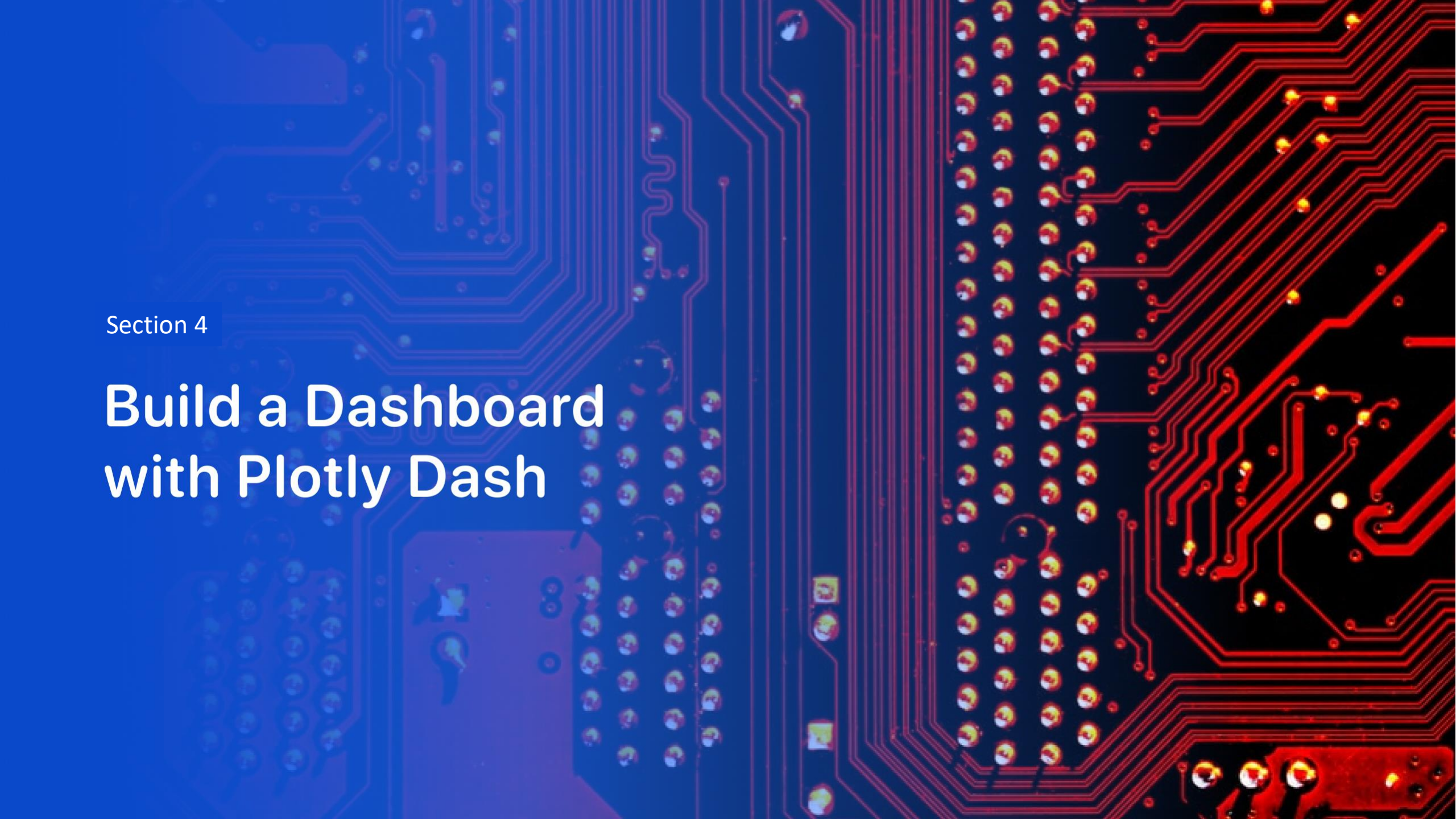


Distances CCAFS LC-40 and nearest highway

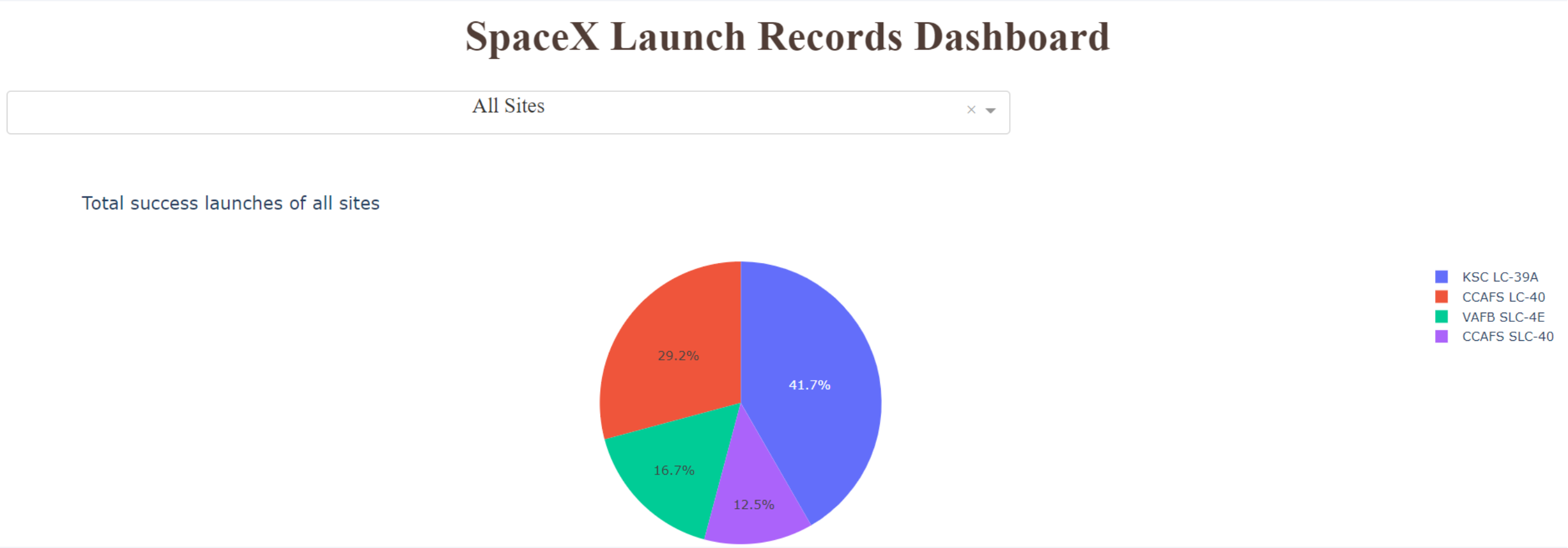


Distances CCAFS LC-40 and nearest railway

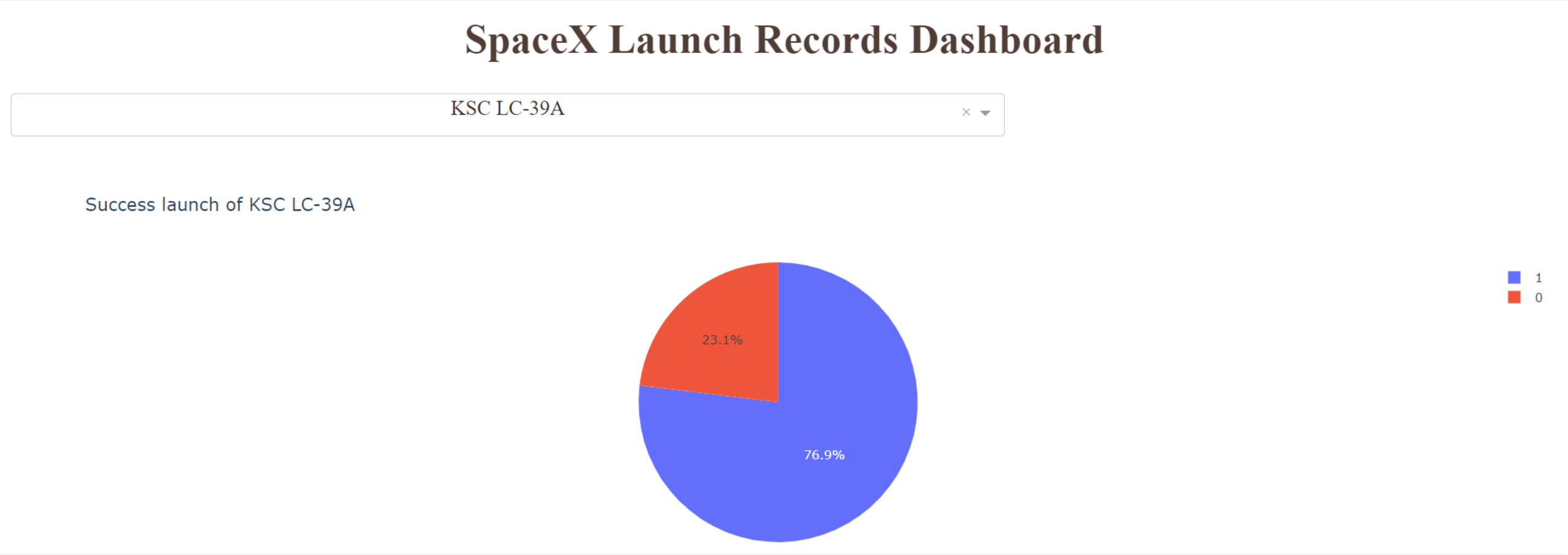



Section 4
Build a Dashboard with Plotly Dash

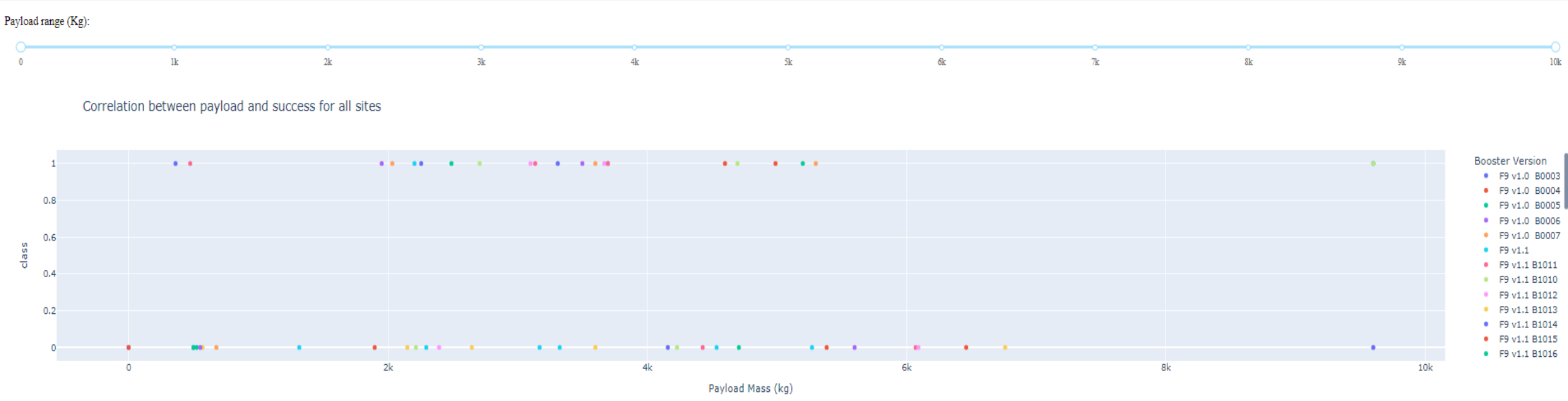
Total success launches of all sites



Launch site with highest success rate



Correlation between payload and success for all sites



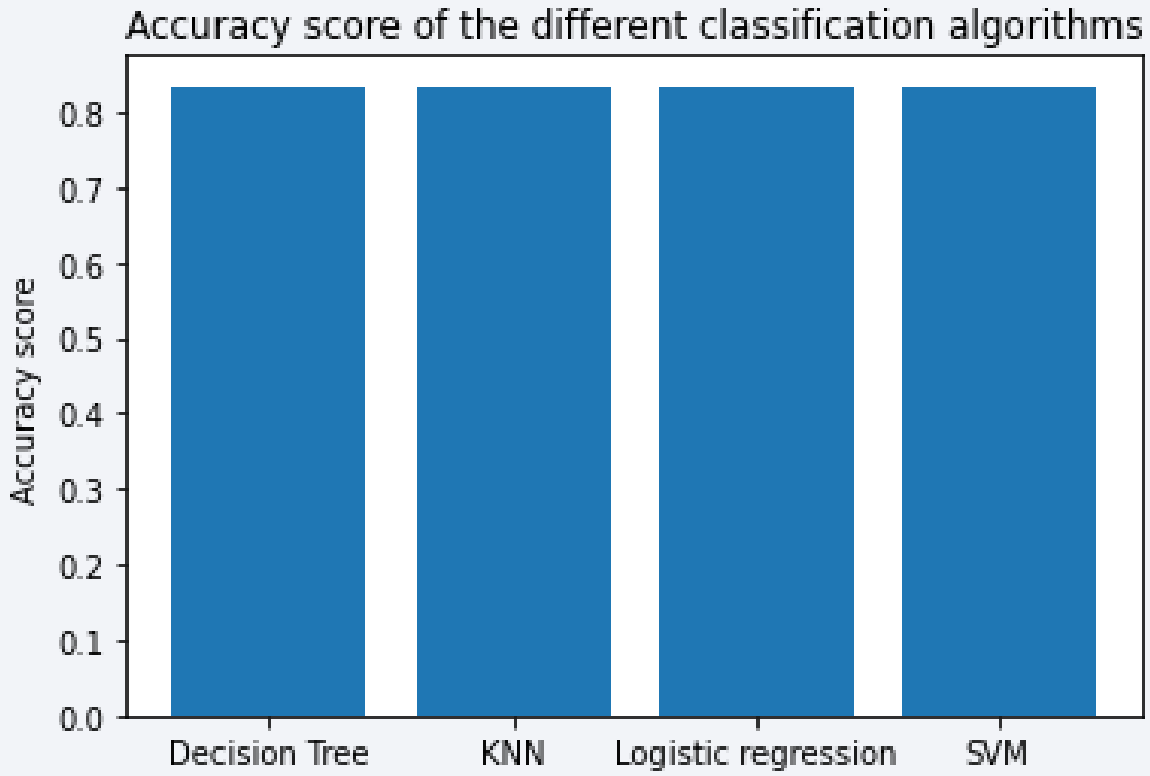
Different payload ranges and launch record dashboards are detailed in the Appendix.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All the models give the same accuracy result of 83.33%.
- We can use the Decision Tree or any other algorithms as all the algorithms are giving the same accuracy.

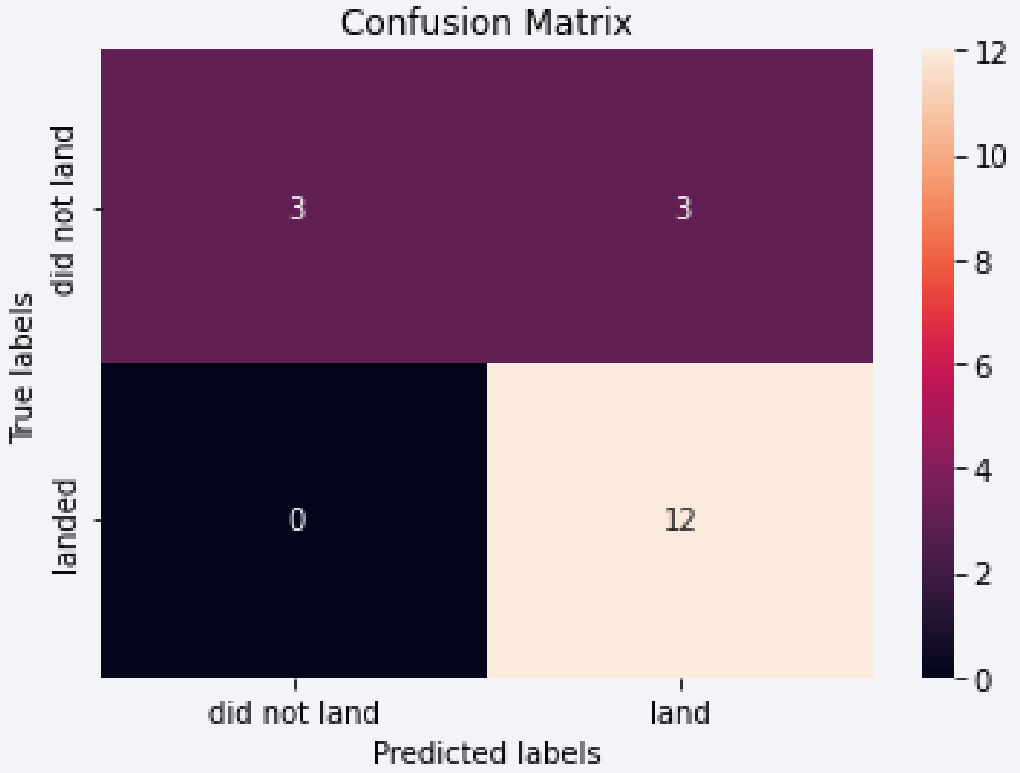



```
In [37]: scores = {'Logistic regression': lr_score, 'SVM': svm_score, 'Decision Tree': tree_score, 'KNN': knn_score}
         print("The method that performs best is", max(scores, key=scores.get))

         The method that performs best is Logistic regression
```

Confusion Matrix

- The model correctly predicted all the landed rockets. It still has some erroneous predictions of the rockets that did not land with giving 3 false positive results.



Conclusions

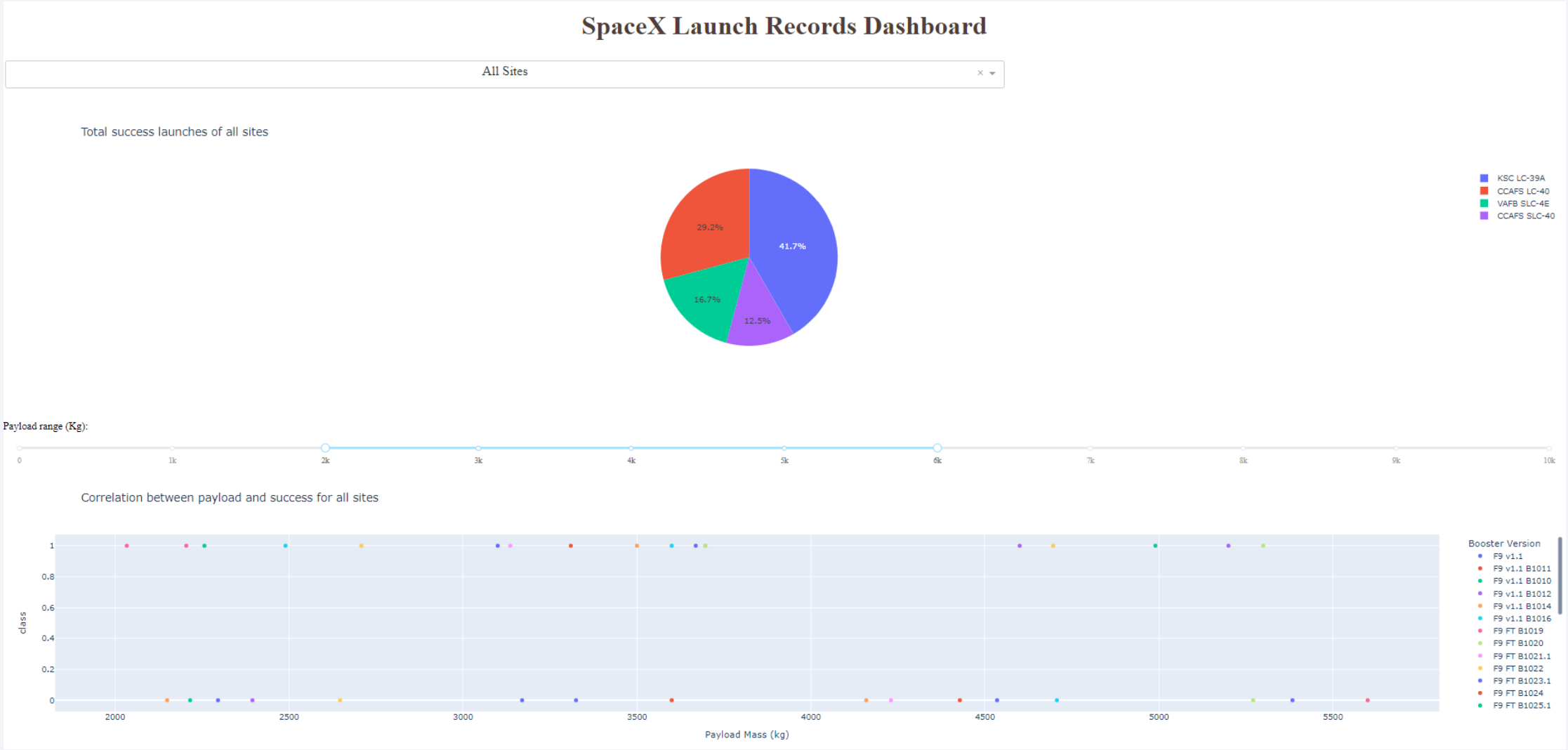
- This project aims to predict if the Falcon 9 first stage will land successfully depending on different parameters such as launch site, payload mass, etc.
- Since 2013, Success rate kept increasing which means that Spacex landing technology is becoming more reliable through time.
- Heavy payloads (greater than 10000 kg) have more change to succeed. As VAB SLC-4E never launched a rocket with a payload greater than 10000 kg, it is better to focus on the other launch sites especially KSC LC39-A which has the highest success rate.
- Four classification models were used to predict the landing success depending on different parameters. All of them have an acceptable accuracy so they can be reliable to predict the success of a landing.

Appendix

- Success rate of the 4 launch sites

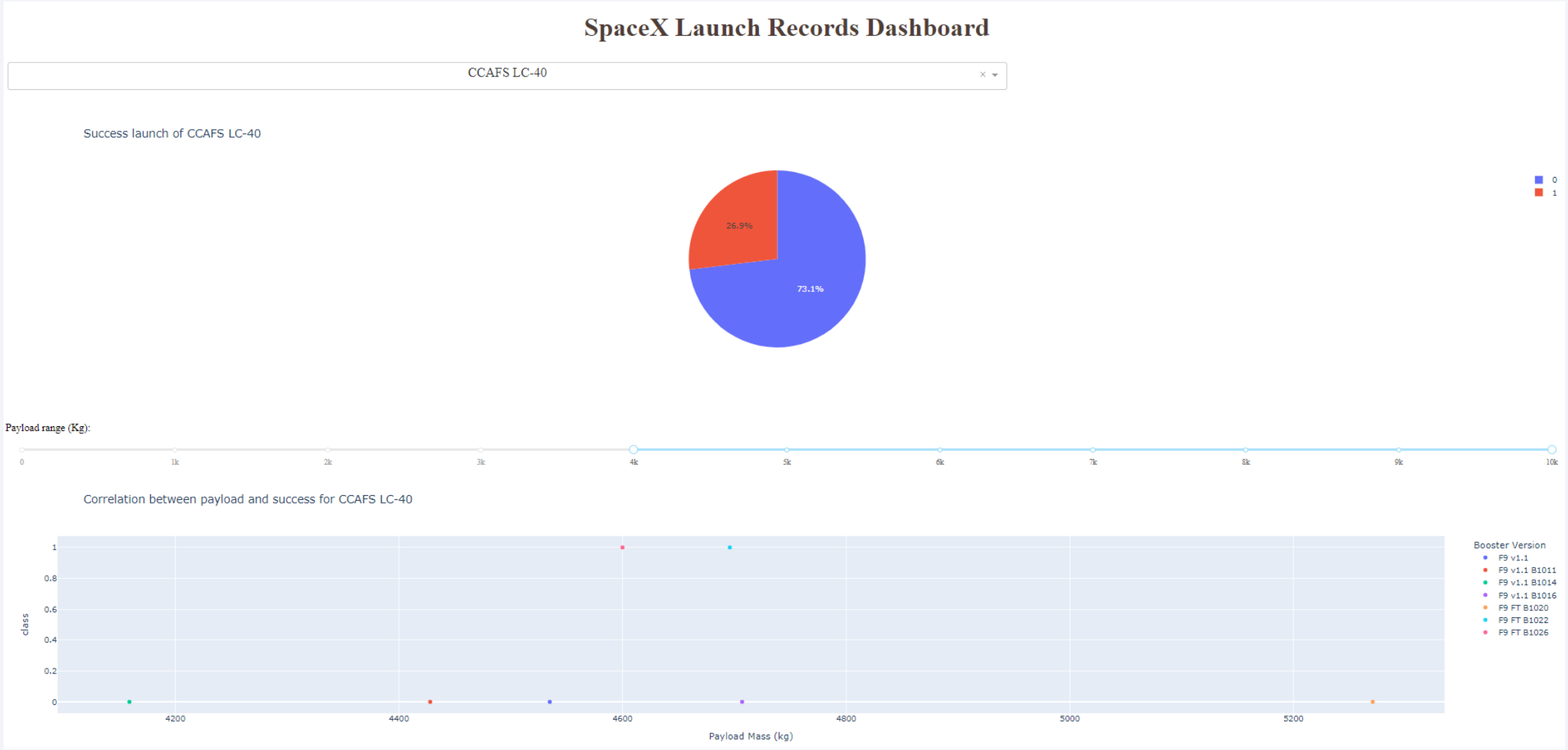
Appendix

- Total success and correlation for payload between 2000 and 6000 kg



Appendix

- Success and correlation for payload greated than 4000 kg for CCAFS LC40



Thank you!