IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Maxwell Kilman

October 15, 2024

Outline

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

Executive Summary

Methodology:

- SpaceX REST API: **collecting** raw data
- EDA: **wrangling**
- **Visualizing** using bar charts, scatter plots, line charts, graphs
- **Analyzing** with SQL to obtain relevant metrics
- **Mapping** sites of launches with Folium
- Creating a **Dashboard** using Plotly
- **Predictive Analysis** (classification models)

Results:

EDA: Positive correlation between year and launch success %

Screenshots: Interactive maps/dashboard

Predictive Analysis: All models successful

Introduction

- Project background and context
- Problems you want to find answers

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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

Place your flowchart of SpaceX API calls here

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

Place your flowchart of web scraping here

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

EDA with Data Visualization

Scatter Plot: Flight Number vs. Launch Site:

- Visualizes the distribution of launches across different sites over time.

Scatter Plot: Payload vs. Launch Site:

- Shows the relationship between payload mass and launch sites.

Bar Chart: Success Rate by Orbit Type:

- Displays the success rate of missions based on orbit type.

Scatter Plot: Flight Number vs. Orbit Type:

- Tracks the types of orbits targeted over the progression of flights.

Scatter Plot: Payload vs. Orbit Type:

- Visualizes payload mass distribution across different orbit types.

Line Chart: Yearly Average Success Rate:

- Tracks changes in the yearly average success rate of launches.

EDA with SQL

- Drop Table
- Create New Table
- List Launch Sites
- Filter Sites by Prefix
- Total Payload for NASA
- Average Payload for Booster F9 v1.1
- Boosters with Successful Drone Ship Landings and Payload Between 4000-6000kg
- Count of Successful and Failed Missions
- Maximum Payload of Booster
- Failed 2015 Landings
- Rank Landing Outcomes by Frequency
- https://github.com/MaxHK24/IBM_Capstone/blob/main/SQL_Notebook_for_Peer_Assignment.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

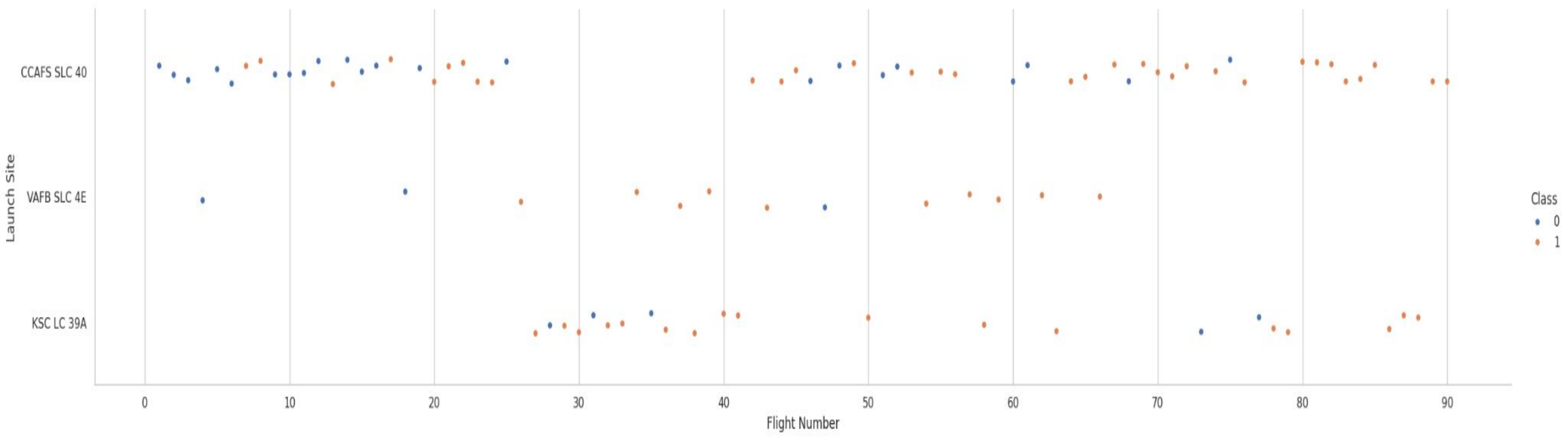
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

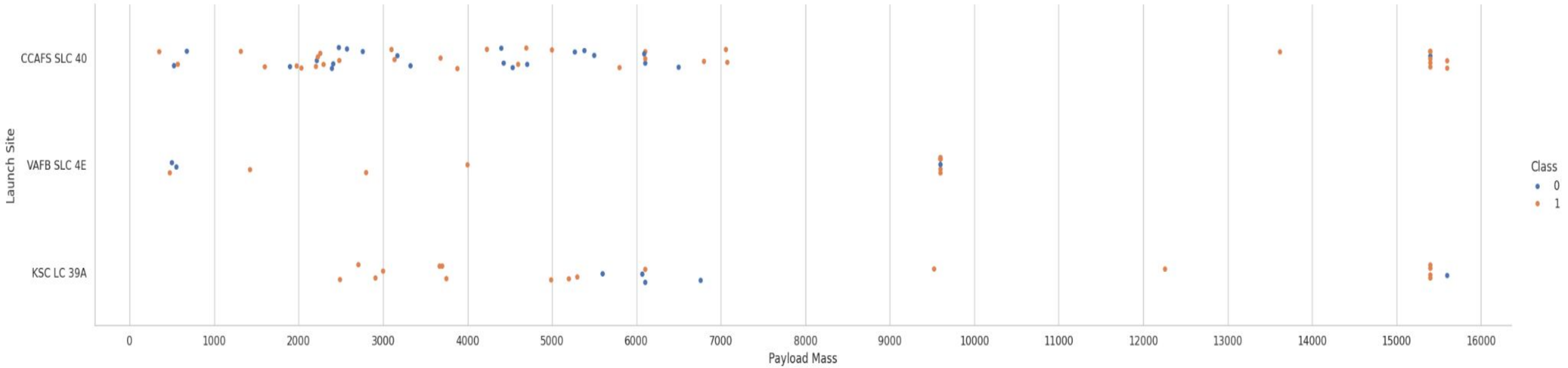
Flight Number vs. Launch Site

Blue dots indicate failed launches. Orange dots indicate successful launches. Overall, CCAFS SLC 40 has about half of all launches, although the other two launch sites have higher success rates. At all three launch sites there is a positive correlation between flight number and success rate.



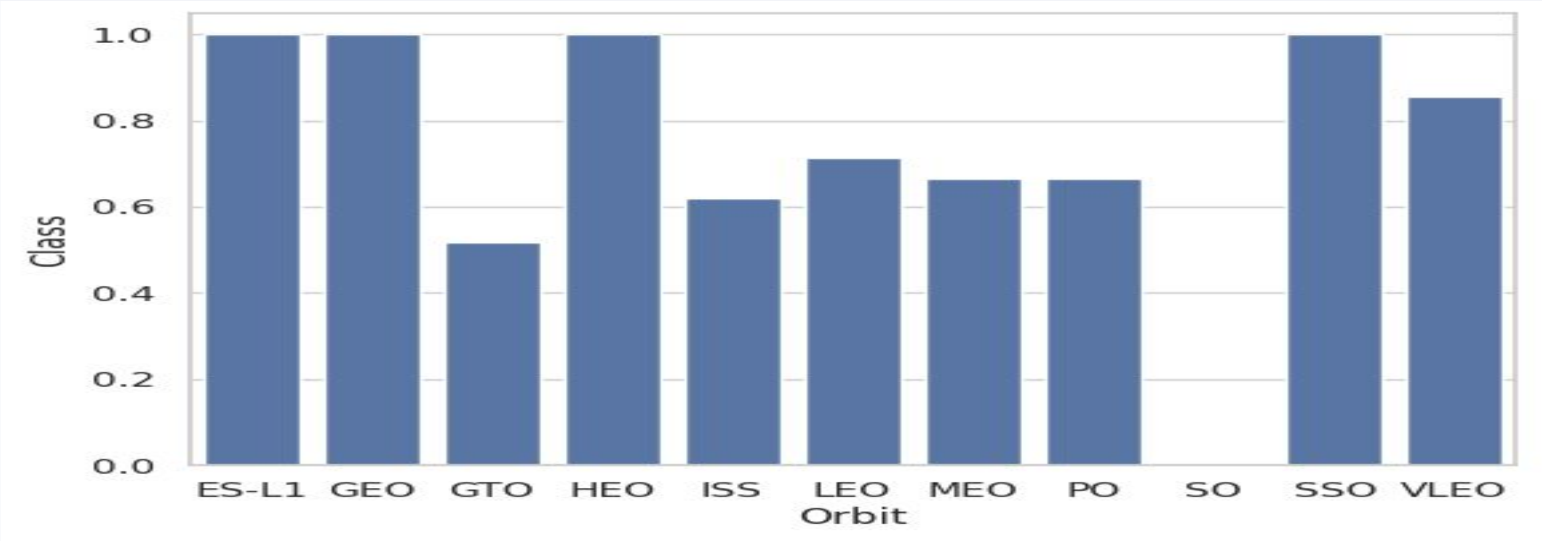
Payload vs. Launch Site

Blue = failed launch. Orange = successful launch. There is a correlation between higher payload mass and launch success; this persists across all launch sites. Launches with payload mass > 7000 kg were overwhelmingly successful. Notably, however, KSC LC 39A succeeded in all launches of payloads < 5500 kg.



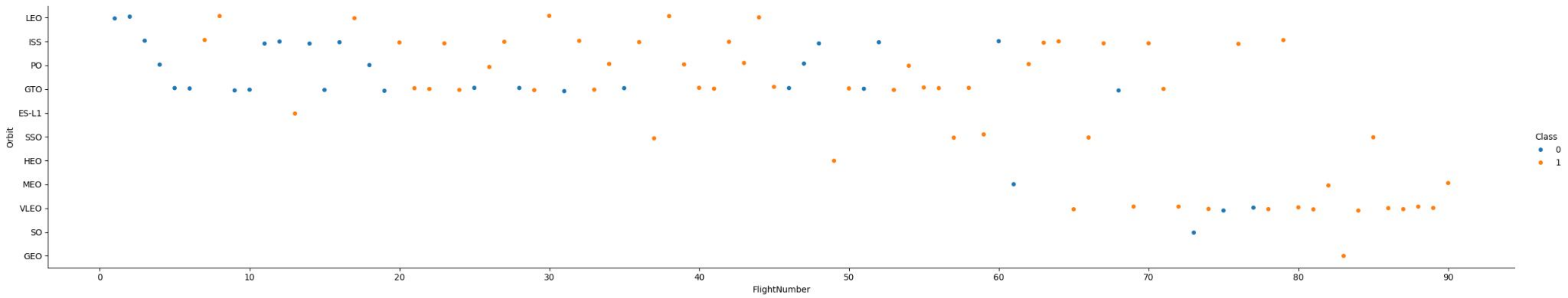
Success Rate vs. Orbit Type

The y-axis represents percentage success rate divided by 100, i.e. 1.0 represents a 100% success rate. The ES-L1, GEO, HEO, and SSO orbits all had a 100% success rate. Conversely, SO had a 0% success rate. All other orbits' success rates ranged between 50% and 85%.



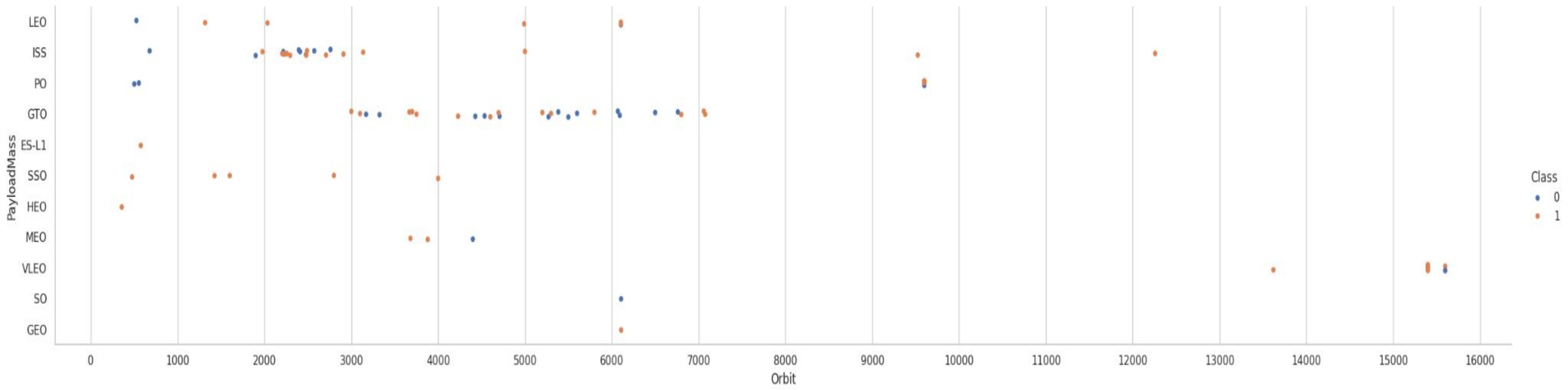
Flight Number vs. Orbit Type

Blue = failed launch. Orange = successful launch. The positive correlation between flight number and success rate persists across all orbit classes, though GTO is a notable outlier in how weak the correlation is. The correlation is strongest for LEO.



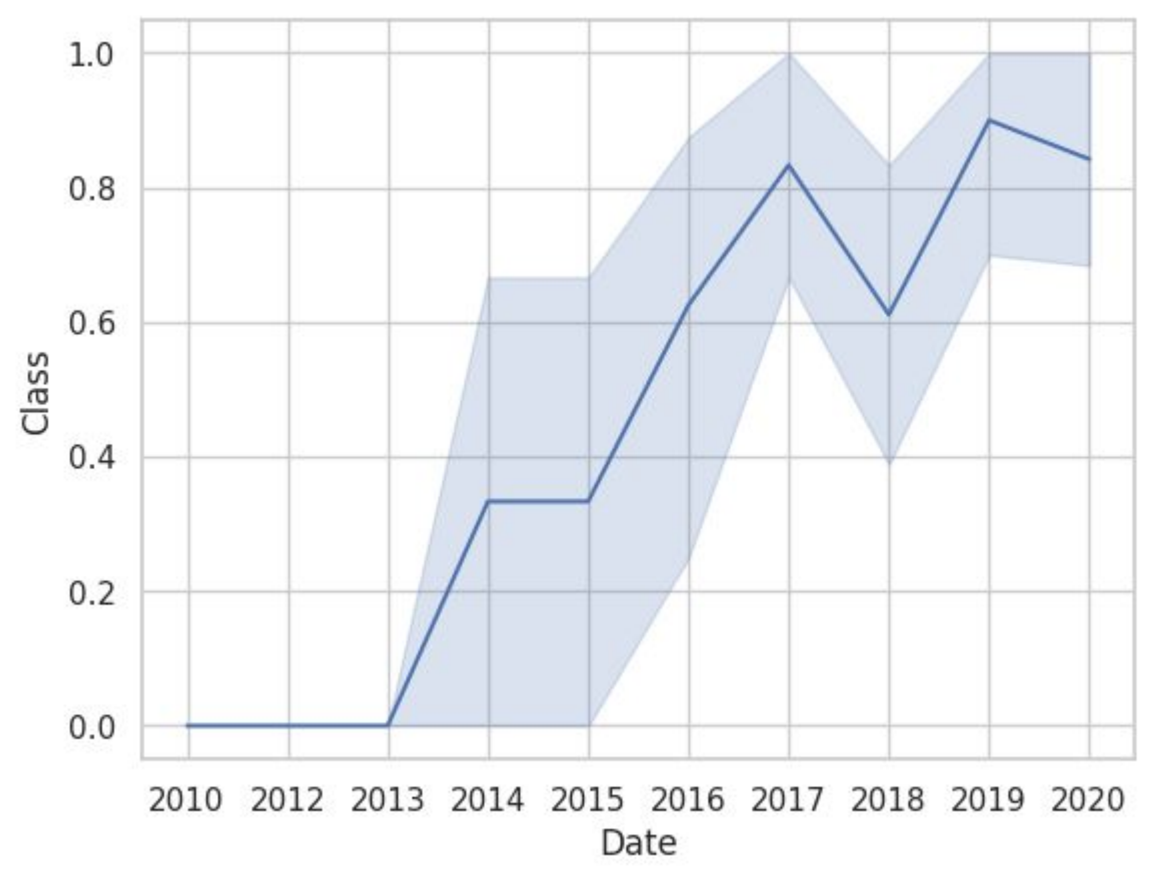
Payload vs. Orbit Type

Most orbits (particularly LEO, ISS, and PO) exhibit a positive correlation between payload mass and success rate. However, this trend is reversed for GTO.



Launch Success Yearly Trend

From 2010, the success rate continually either increased or plateaued on an annual basis until 2018; it has fluctuated considerably since then.



All Launch Site Names

- This query counts the number of launches for each unique Launch_Site in the SPACEXTABLE and then from the subquery only selects the Launch_Site columns

```
%sql SELECT Launch_Site FROM (SELECT Launch_Site, COUNT(*) FROM SPACEXTABLE GROUP BY Launch_Site) AS grouped_launch_sites;
```

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

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

Launch Site Names Begin with 'CCA'

- I told the query to iterate the rows of SPACEXTBL and select each successive row for which the Launch_Site column entry started with ‘CCA’ until it had selected a maximum of five rows.

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%'LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- I use a subquery to first select the payload masses for those launches whose customer is “NASA (CRS)” and then summate them in the outer query

Task 3

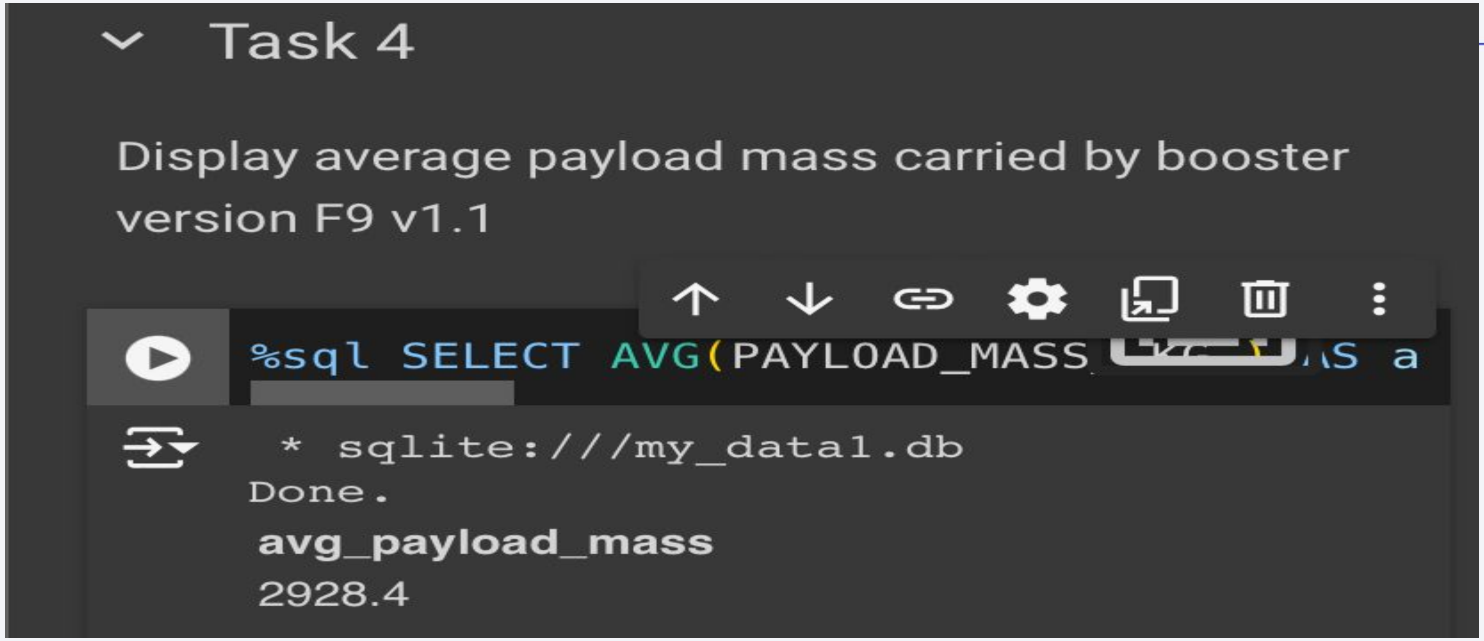
Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM (SELECT PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE Customer = '
```

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

total_payload_mass
45596

Average Payload Mass by F9 v1.1



Query: SELECT AVG(PAYLOAD_MASS__KG_) AS avg_payload_mass FROM (SELECT PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1');

explanation: inner query selects payload mass from all records where the booster version is Falcon 9 v1.1. Outer query then calculates the average of these payload masses.

First Successful Ground Landing Date

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

Hint:Use min function

```
SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground
```

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

min(DATE)
2015-12-22

Query: SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';

The query concatenates the entries in the spacextbl where the outcome value is listed as a successful ground landing, and then chooses that which has the “minimum” (smallest, e.g. earliest) date value

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM ( SELECT Booster_Version, PAYLOAD_MASS__KG_,
Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' )
AS filtered_boosters WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

- This query first filters records from the SPACEXTABLE where the landing outcome was a successful drone ship landing (Landing_Outcome = 'Success (drone ship)'). From this it further selects boosters with a payload mass between 4000 and 6000 kg. The final output lists the booster versions that met both conditions, showing successful drone ship landings within the specified payload range.

Total Number of Successful and Failure Mission Outcomes

- This query first filters the SPACEXTABLE to select only records where the mission outcome was either 'Success' or 'Failure (in flight)'. The outer query then counts the total number of these filtered records, providing the total count of missions that either succeeded or failed in flight.

List the total number of successful and failure mission outcomes

```
[ ] %sql SELECT COUNT(*) FROM ( SELECT Mission_Outcome FROM SPACEXTABLE WHERE Mission_Outcome = 'Success' OR Mission_Outcome = 'Failure (in flight)' ) AS filtered_missions;
```

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

COUNT(*)
99

Boosters Carried Maximum Payload

The inner query selects the highest value of PAYLOAD_MASS__KG_. The outer query selects from SPACEXTABLE the specific Booster_Version for which the PAYLOAD_MASS__KG is equal to that selected by the inner query.

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

```
[ ] %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = ( SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE );
```

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

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

The query retrieves records of failed drone ship landings from the year 2015 and then converts the month portion of the date into the corresponding month name (e.g., "01" to "January"). It then selects the month, booster version, launch site, and landing outcome for each failed landing.

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
[ ] %sql SELECT CASE WHEN substr(Date, 6, 2) = '01' THEN 'January' WHEN substr(Date, 6, 2) = '02' THEN 'February' WHEN substr(Date, 6, 2) = '03' THEN 'March' WHEN substr(Date,
```

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

Month_Name	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

I filter records with WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' to exclusively pull records that fall within that date range. GROUP BY Landing_Outcome groups these records by each unique landing outcome, and COUNT(*) AS outcome_count enumerates occurrences of each. ORDER BY outcome_count DESC sorts the results in descending order.

```
[ ] %sql SELECT Landing_Outcome, COUNT(*) AS outcome_count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY outcome_count DESC;

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

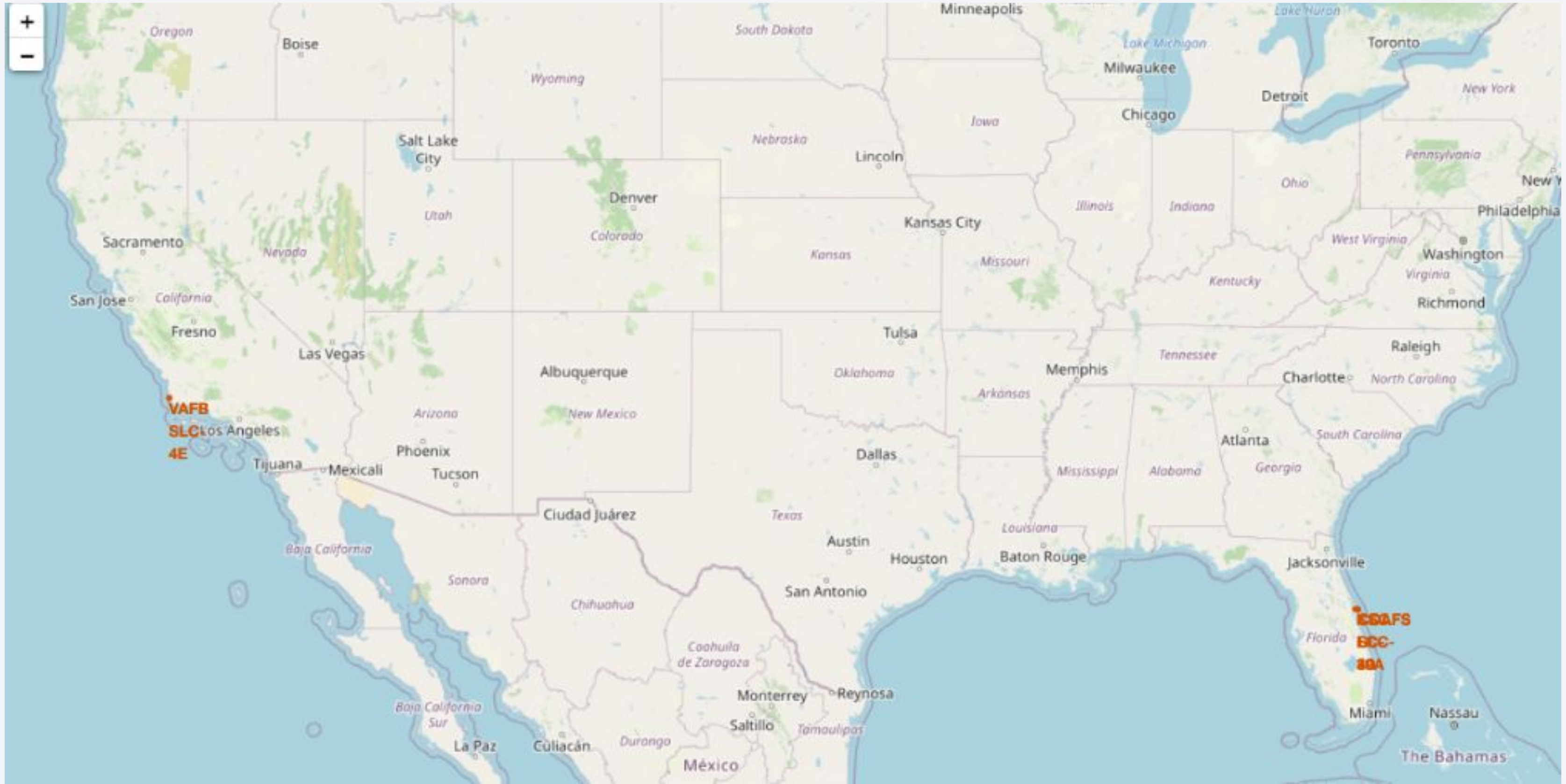
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

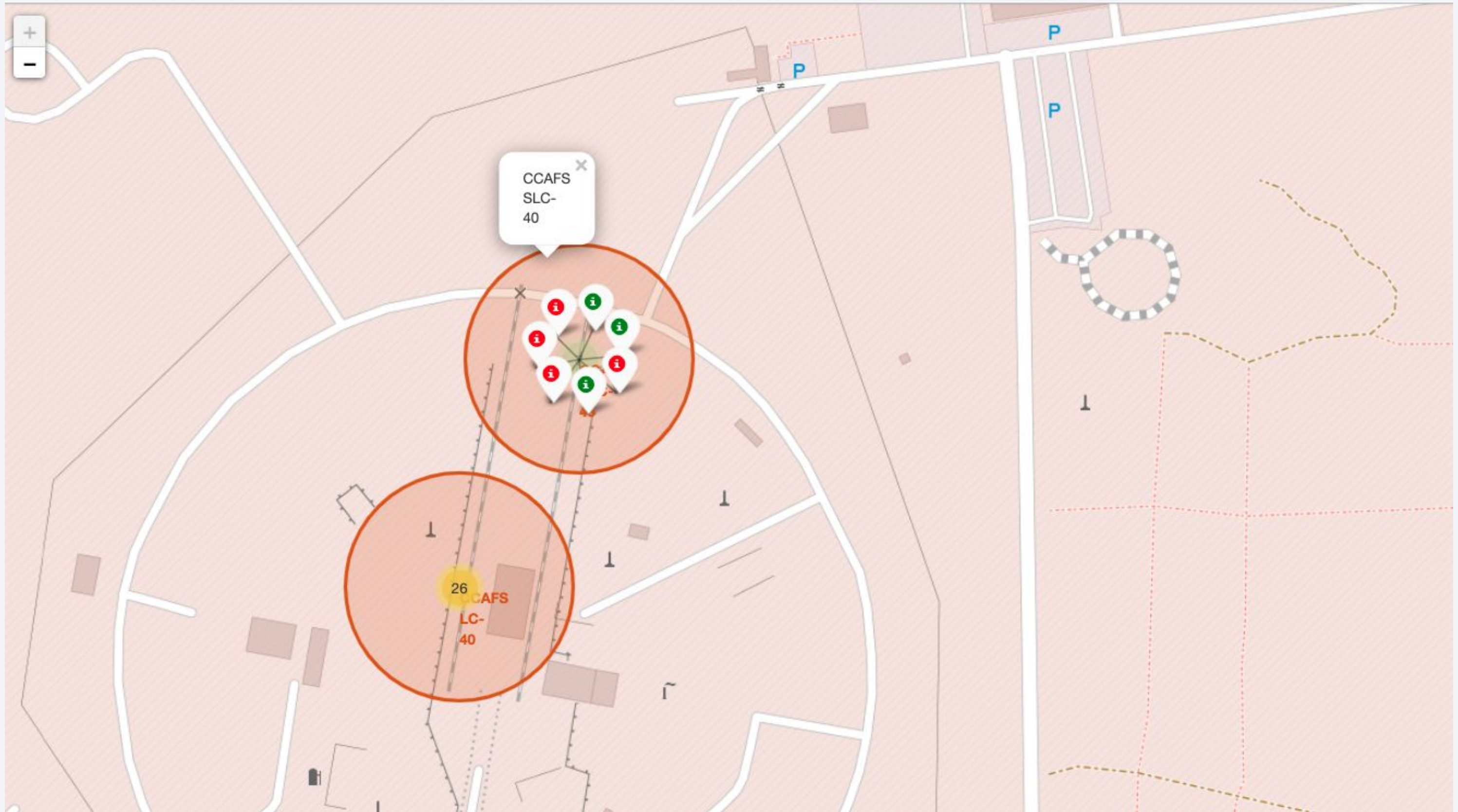
Launch Site locations

The launch sites are in Brevard County, FL and Santa Barbara County, CA – both year-round warm weather locations on the coast with climatic conditions that are consistently suitable for launches

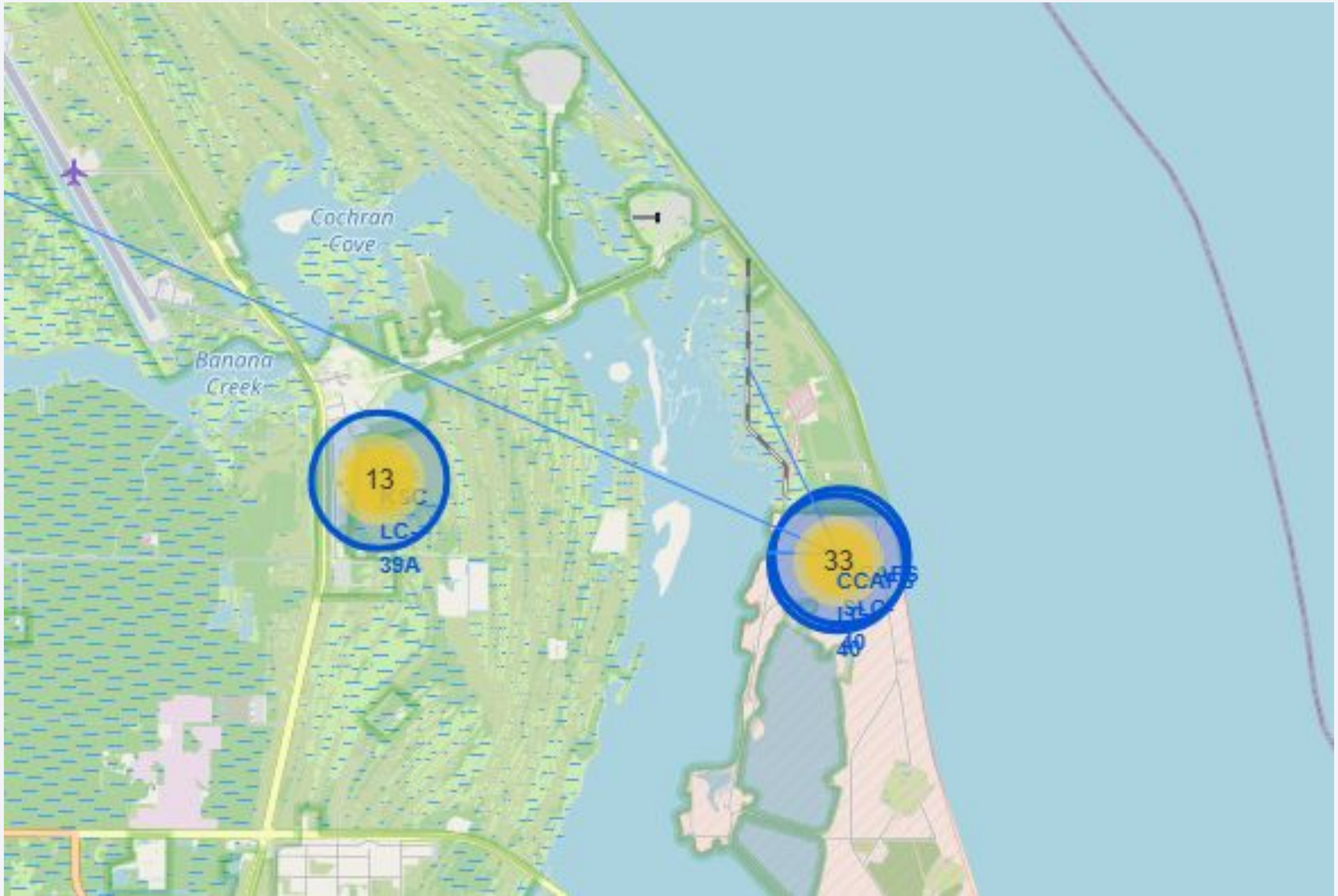


Launch Outcomes

Red = failed launch. Green = successful launch.



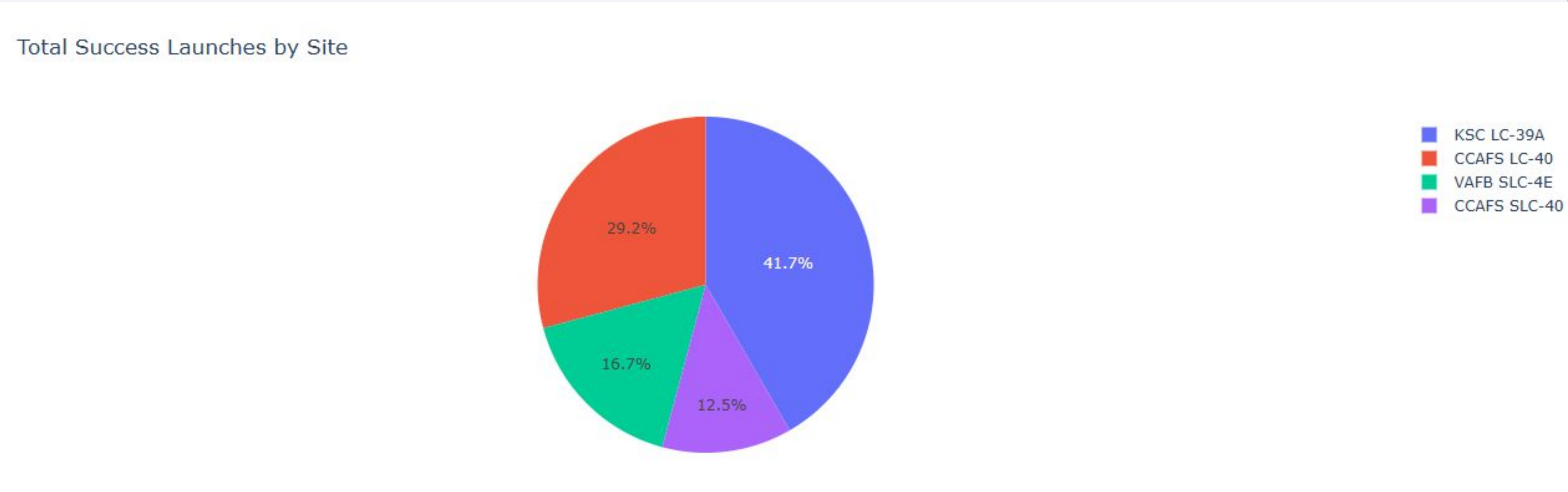
Launch Site Proximities



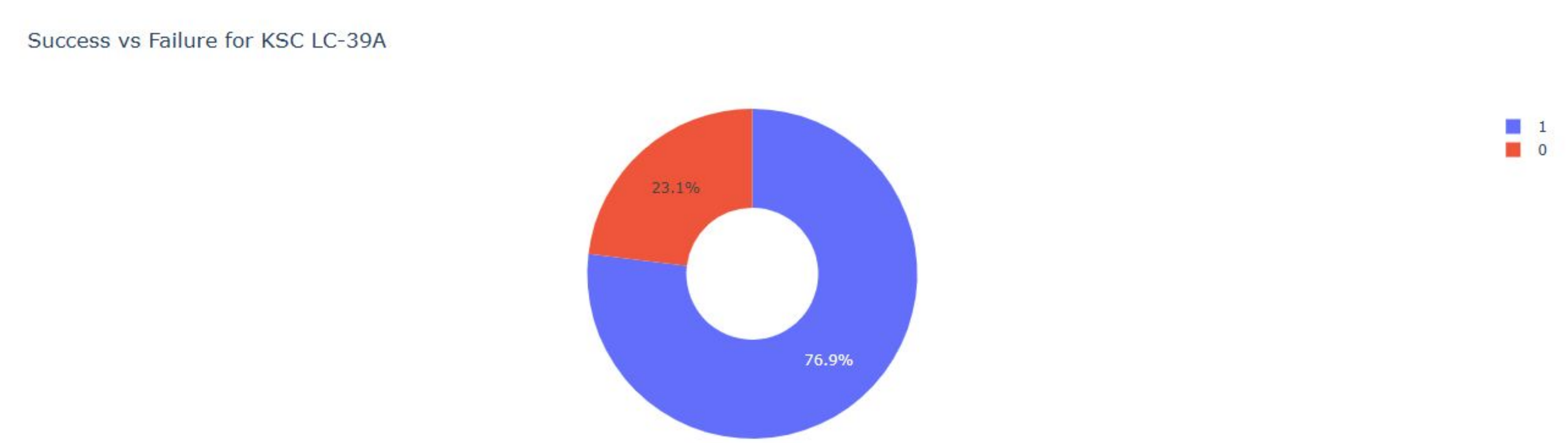
Section 4

Build a Dashboard with Plotly Dash

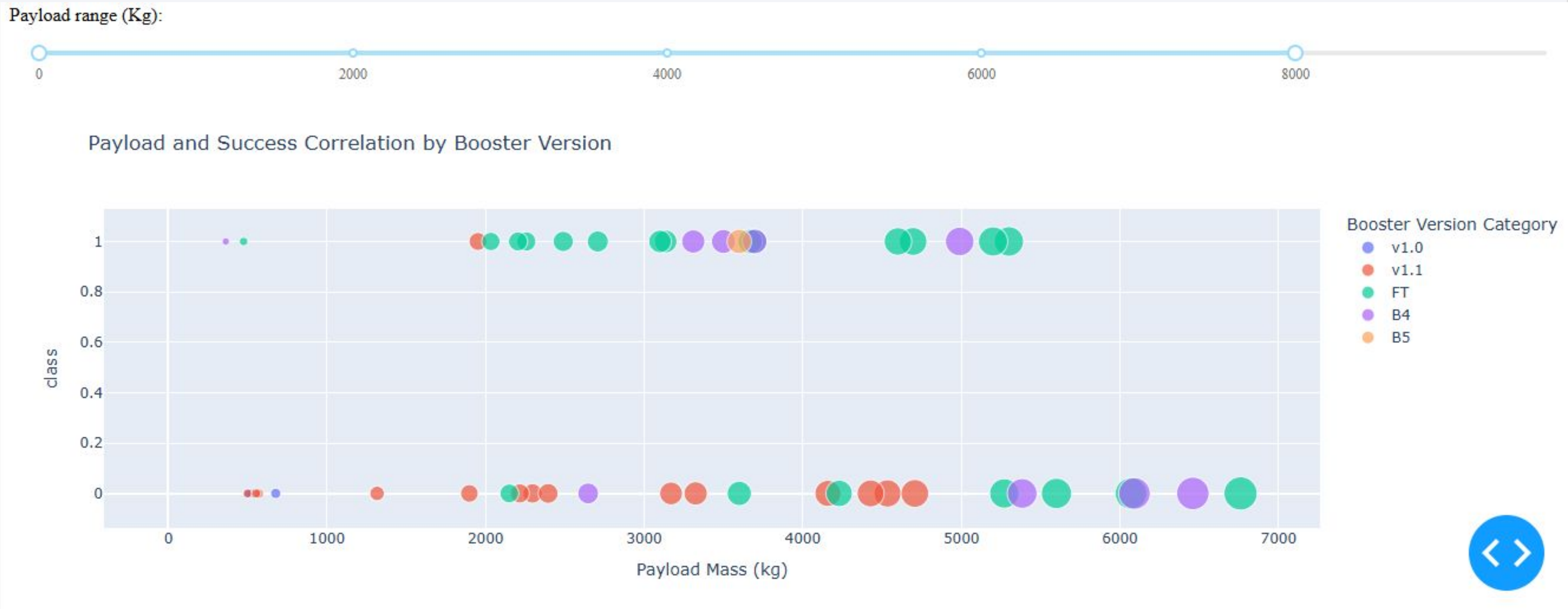
Launch Success by Site



KSC success/failure



Payload and Launch Outcome



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!