SpaceY Project

Gabriel Silva de Melo
27/09/2024

Outline

Executive Summary

Summary of Methodologies

- Data collection with SpaceX API and web scraping.
- Data wrangling to transform the data and make it better for analysis.
- Turning data into a dataframe for exploration.
- Data analysis for finding useful information.
- Data visualization with plots and charts.
- Predictive analysis using machine learning models.

Summary of results

- Launching tends to improve over time.
- All launch sites are near the cost.
- The booster F9 v1.1 has an average payload of 2534.66
- Most ML models tested had the same accuracy, only Decision Trees performed slightly better.

Introduction

SpaceX can make rocket launching cheaper than other companies since they can reuse the first stage. The goal of this project is to predict when the landing of the first stage will be successful.

Explore

- Discover how variables like payload, orbit, launch site and number of flights affects the rate of success of first stage recovery.
- Find the best model to predict the outcome of the mission.

Section 1

Methodology

Methodology

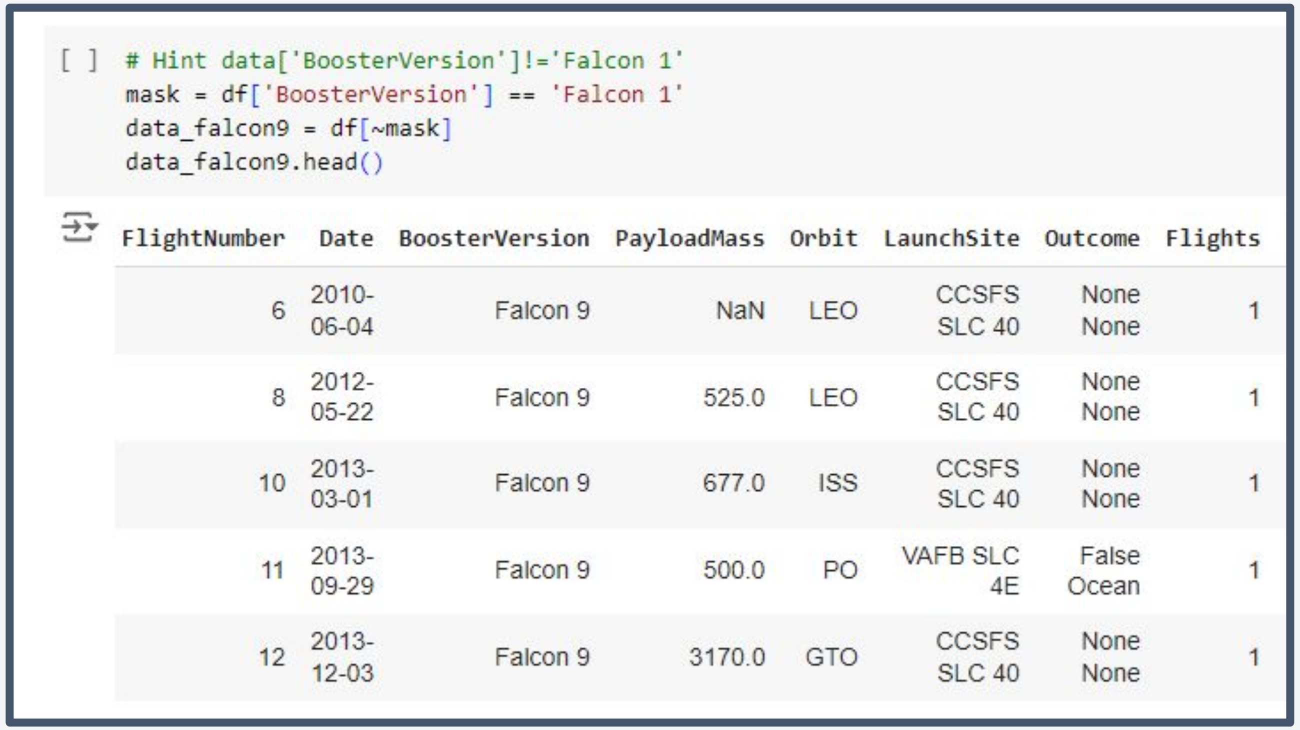
- **Data collection:** SpaceX API and web scraping on wikipedia.
- **Data wrangling:** Pandas and Numpy were used to clean the data and to create a new ‘Class’ column to make it possible to use it as a dependent variable for the ML models.
- **Exploratory data analysis (EDA):** SQL was used to find key information on the data, plots were also used to find correlation on the data.
- **Interactive visual analytics**: Folium was used to build interactive maps and pinpoint launch sites on the map, also, Dash was used to display charts in a interactive way.
- **Predictive analysis**: Scikit Learn was used to build and train ML models using the available data, looking to predict the ‘Class’.

Data Collection

- The Data collection process involved using the SpaceX API and also some web scraping on the wikipedia page using requests and Beautifulsoup.

Data Collection – SpaceX API

- Collecting data from the SpaceX API involved making a request to the API, which returned a json file that was opened with pandas to make possible retrieving the data.
- This data was added to several lists that later became the Data Frame used for the project.
- Data was filtered so that the dataframe only contained Falcon 9 data.



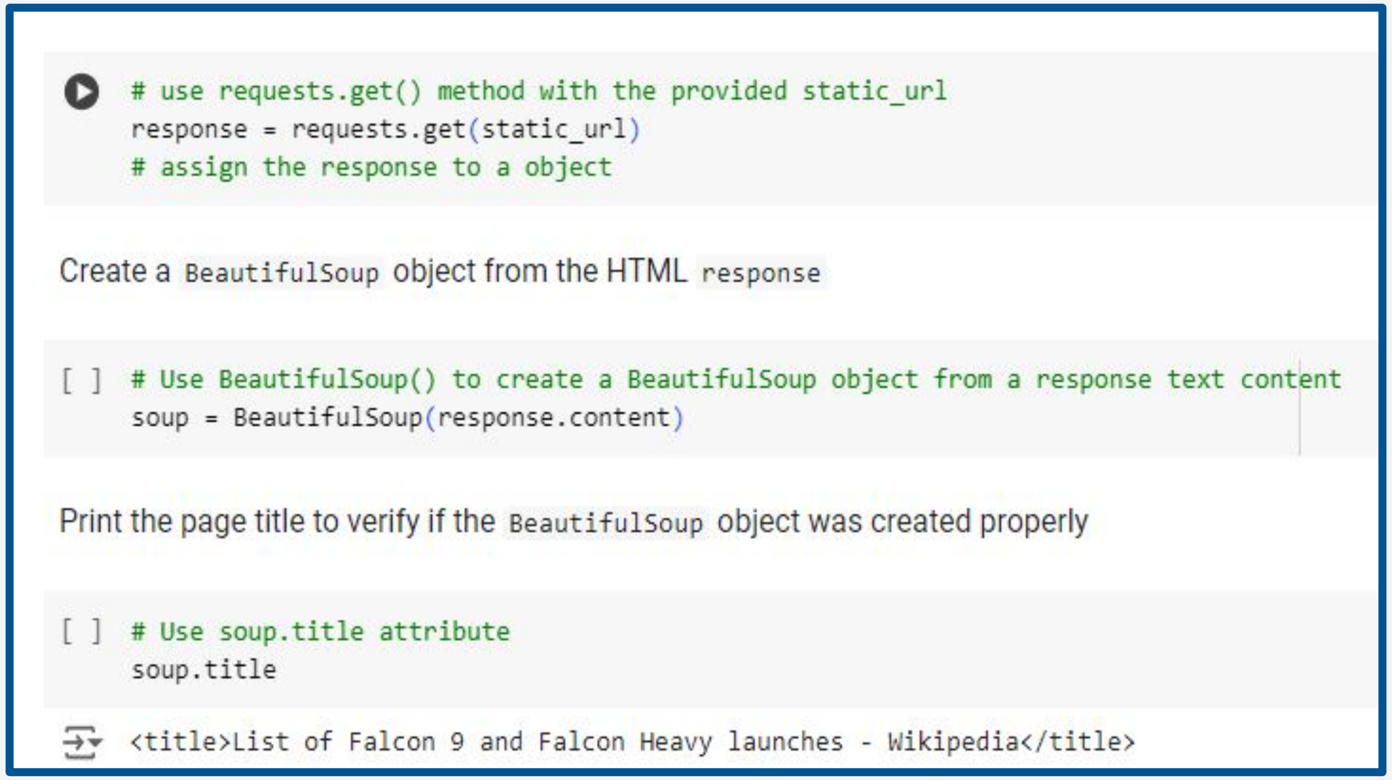
```
[ ] # Hint data['BoosterVersion']!='Falcon 1'
    mask = df['BoosterVersion'] == 'Falcon 1'
    data_falcon9 = df[~mask]
    data_falcon9.head()
```

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1
8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1
10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1
11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1
12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/01%20Data%20Collection%20API%20notebook.ipynb

Data Collection - Scraping

- Request and Beautifulsoup were used to web scrape a table about SpaceX from a Wikipedia page.
- After scraping data was turned into a pandas dataframe.



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

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

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

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>
```

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/02%20Data%20Collection%20Web%20Scrapping.ipynb

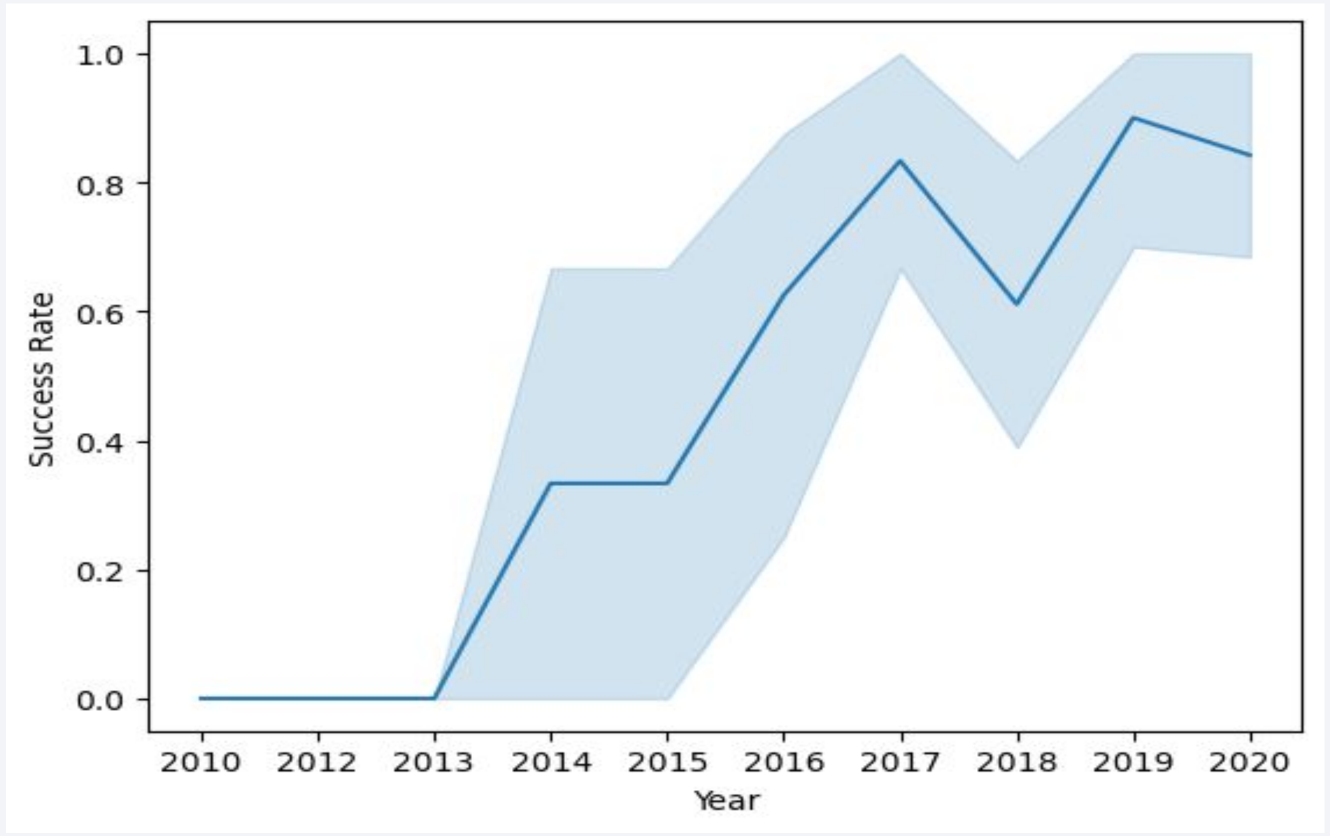
Data Wrangling

- Data needed to be processed because there were NaN values, these were replaced by the average of the columns they were in.
- A new column was created to better fit the model used to predict the outcome of the launch.
- At this part we were able to see how many null values there were and also see the number of launches per launch site.

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/03%20Data%20wrangling.ipynb

EDA with Data Visualization

- Visualization was used to see the correlation of variables and also to find out how the success rate of rocket launches is increasing.



https://github.com/awesomegab/IBM-DS-Capstone/blob/main/04%20Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- SQL was used to find information such as:
    - Unique launch sites.
    - Records for launch site ‘CCA’.
    - Total payload carried by boosters launched by NASA
    - Average payload carried by booster version F9 v1.1
    - and more.

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/05%20SQL%20EDA%20notebook.ipynb

Interactive Map with Folium

- Folium was used to create interactive maps and mark the locations of the launch sites, making it possible to see that launch sites are always located near the coast and that most launch sites are near the Equator line.
- Also distance from the nearest city, railway and roadway were measured using the Folium map.

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/06%20Locations%20Analysis%20with%20Folium.ipynb

Dashboard with Plotly Dash

- Pie chart with successful launches for each launch site.
- Slider of paymass.
- Scatter plot of payload vs success rate.

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/07%20spacex_dash_app.py.txt

Predictive Analysis (Classification)

- Several Scikit learn models were utilized to predict ‘Class’ column, all combined with GridSearchCV.
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K-Nearest Neighbor

https://github.com/awesomegab/IBM-DS-Capstone/blob/main/08%20Machine%20Learning%20Prediction.ipynb

Results

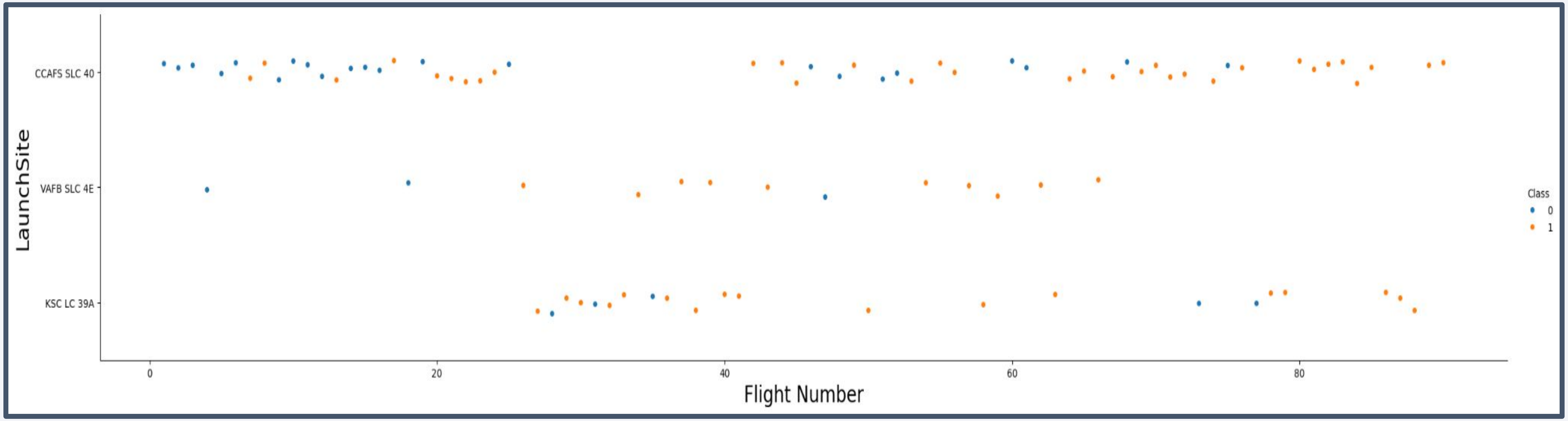
- Launching tends to improve over time.
- All launch sites are near the cost.
- The booster F9 v1.1 has an average payload of 2534.66
- Most ML models tested had the same accuracy, only Decision Trees performed slightly better.

Section 2

Insights drawn from EDA

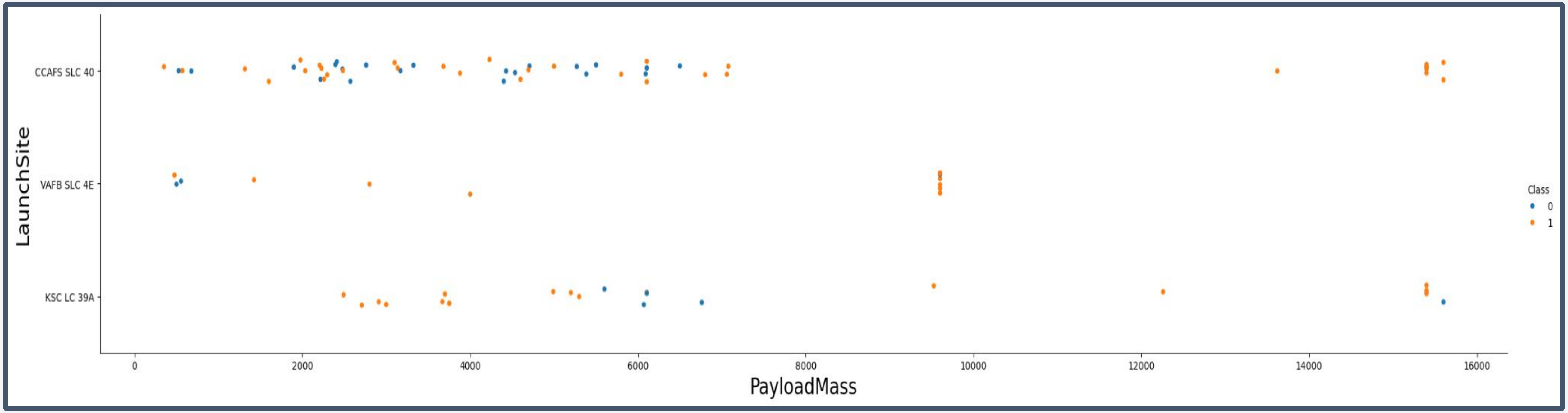
Flight Number vs. Launch Site

Blue points = failure
Orange points = success

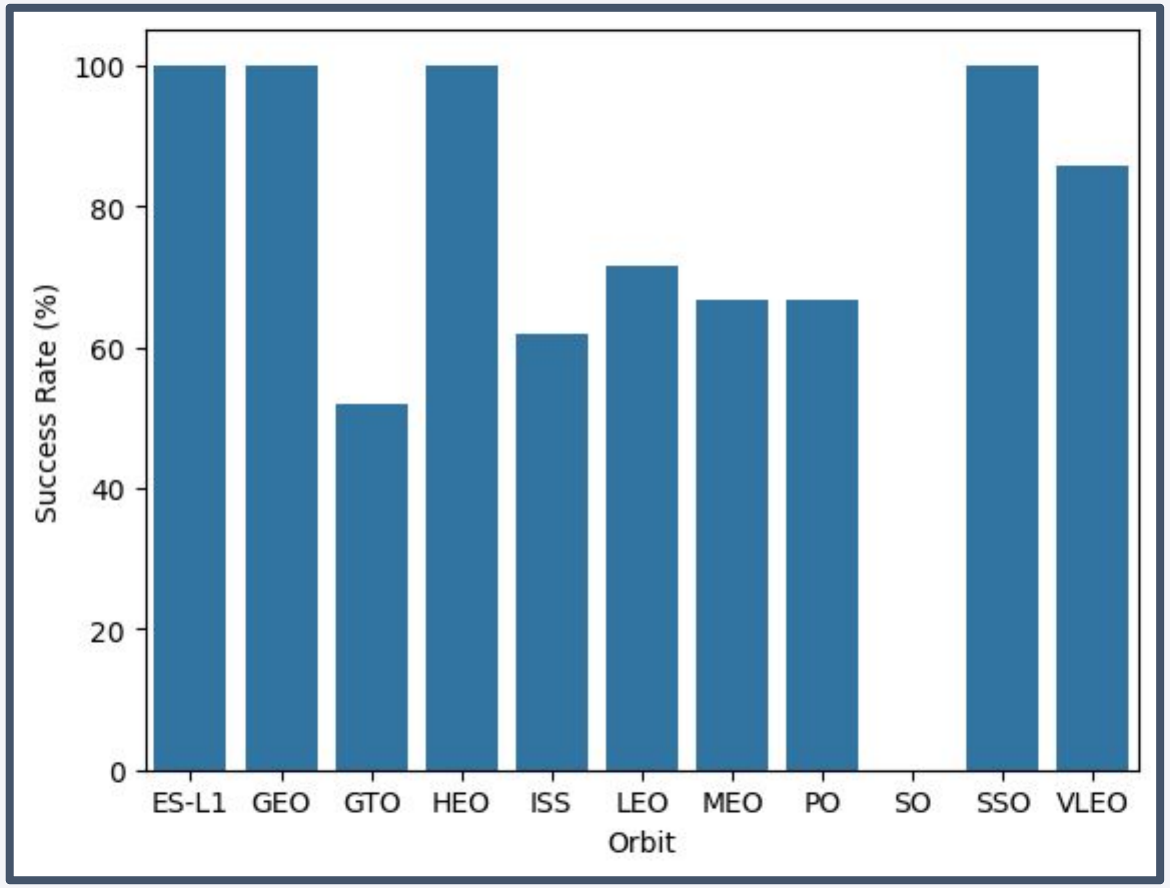


Payload vs. Launch Site

Blue points = failure
Orange points = success



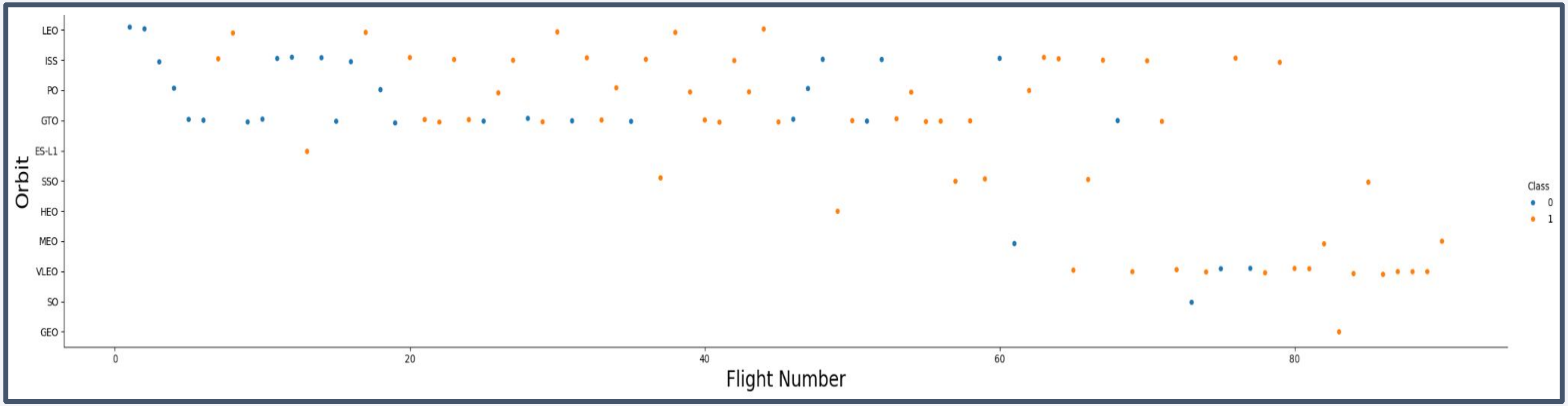
Success Rate vs. Orbit Type



Note that some orbits have 100% of success rate.

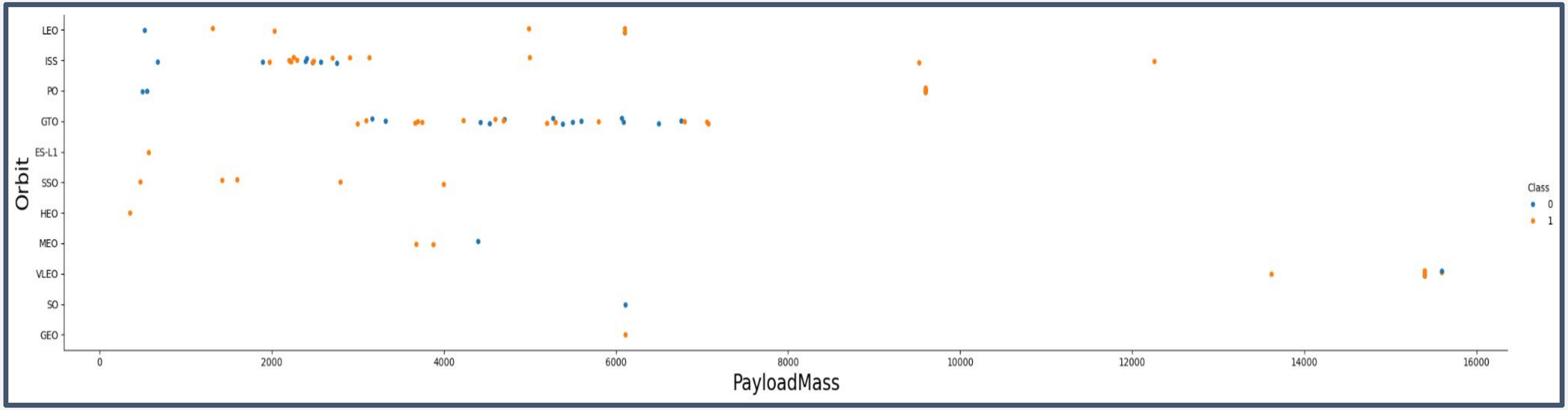
Flight Number vs. Orbit Type

Blue points = failure
Orange points = success

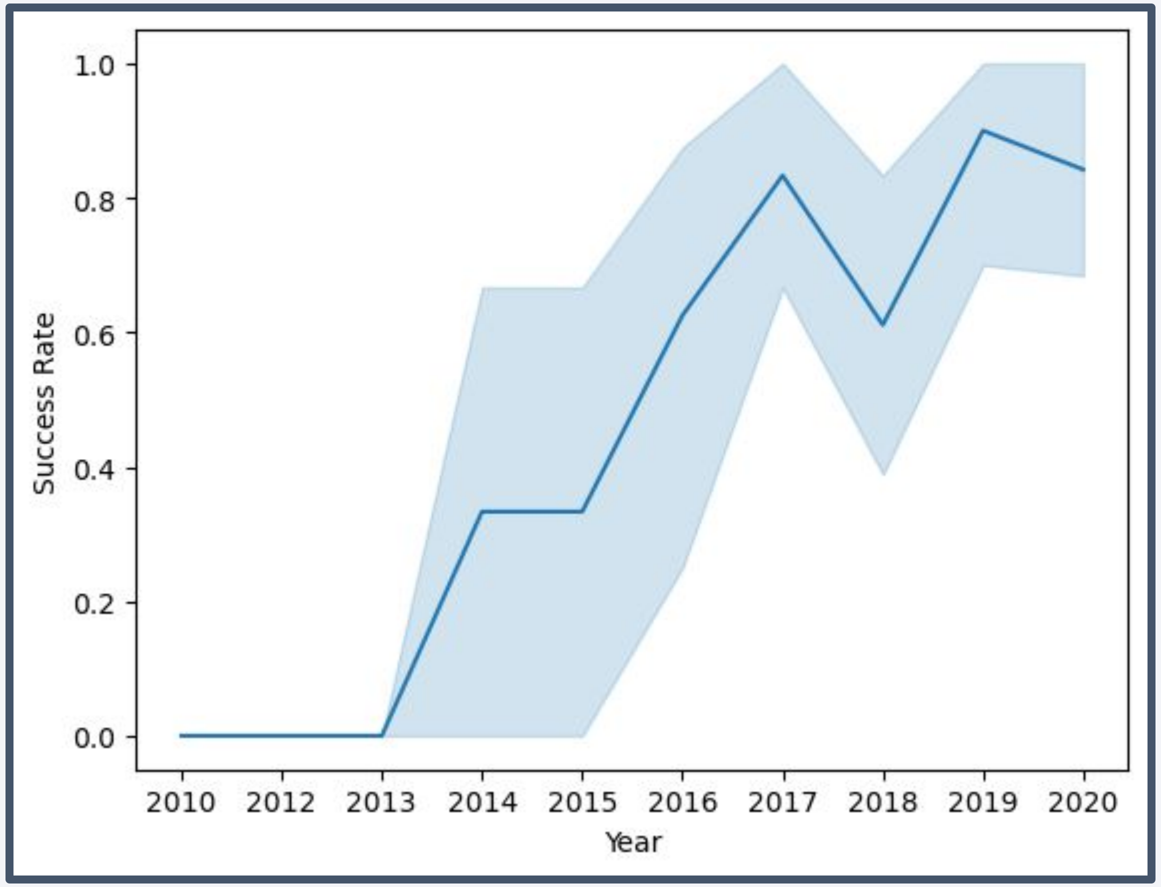


Payload vs. Orbit Type

Blue points = failure
Orange points = success



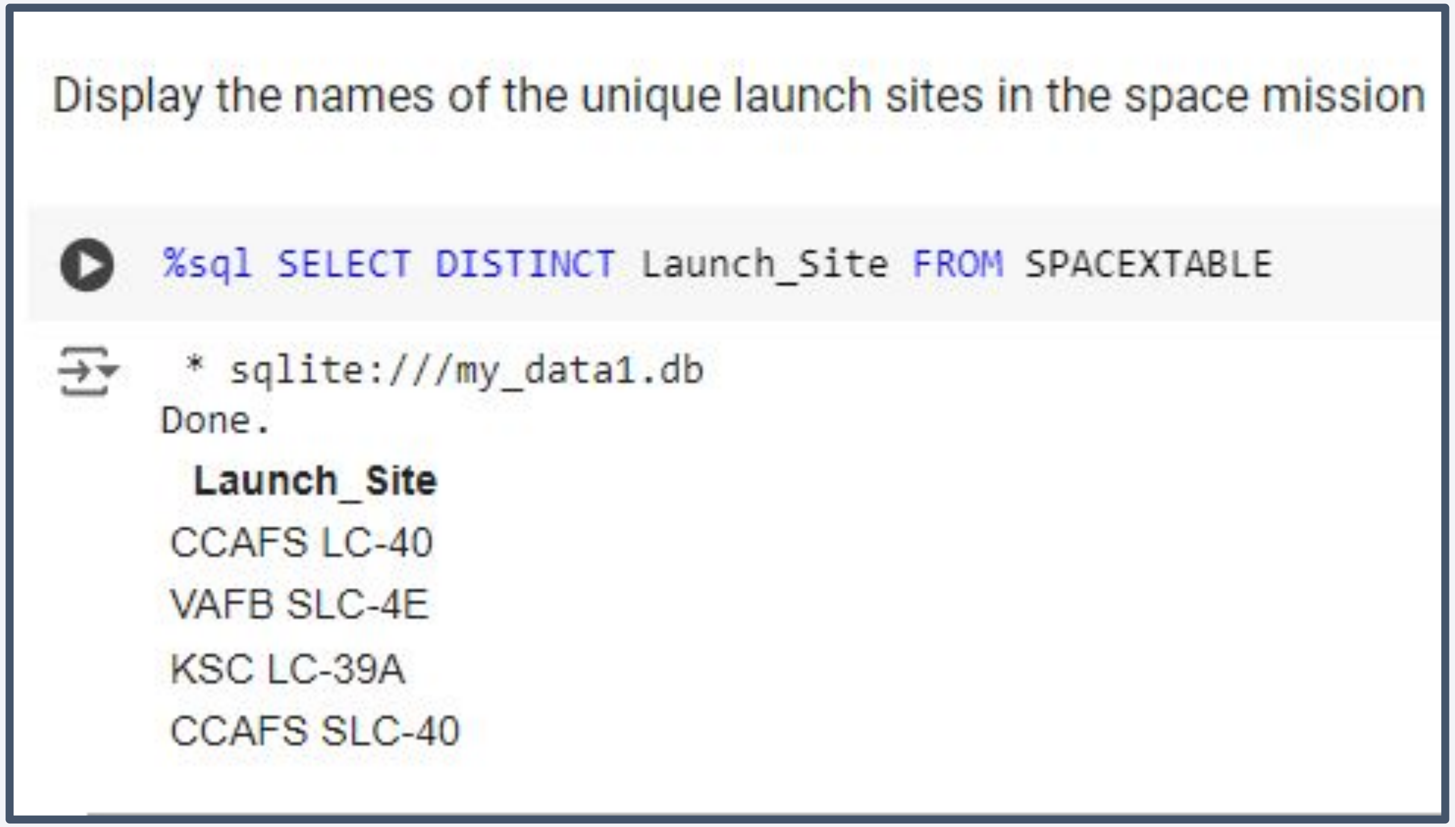
Launch Success Yearly Trend



Note that success rate has increased massively since 2013.

All Launch Site Names

Retrieving the names of the launch sites:



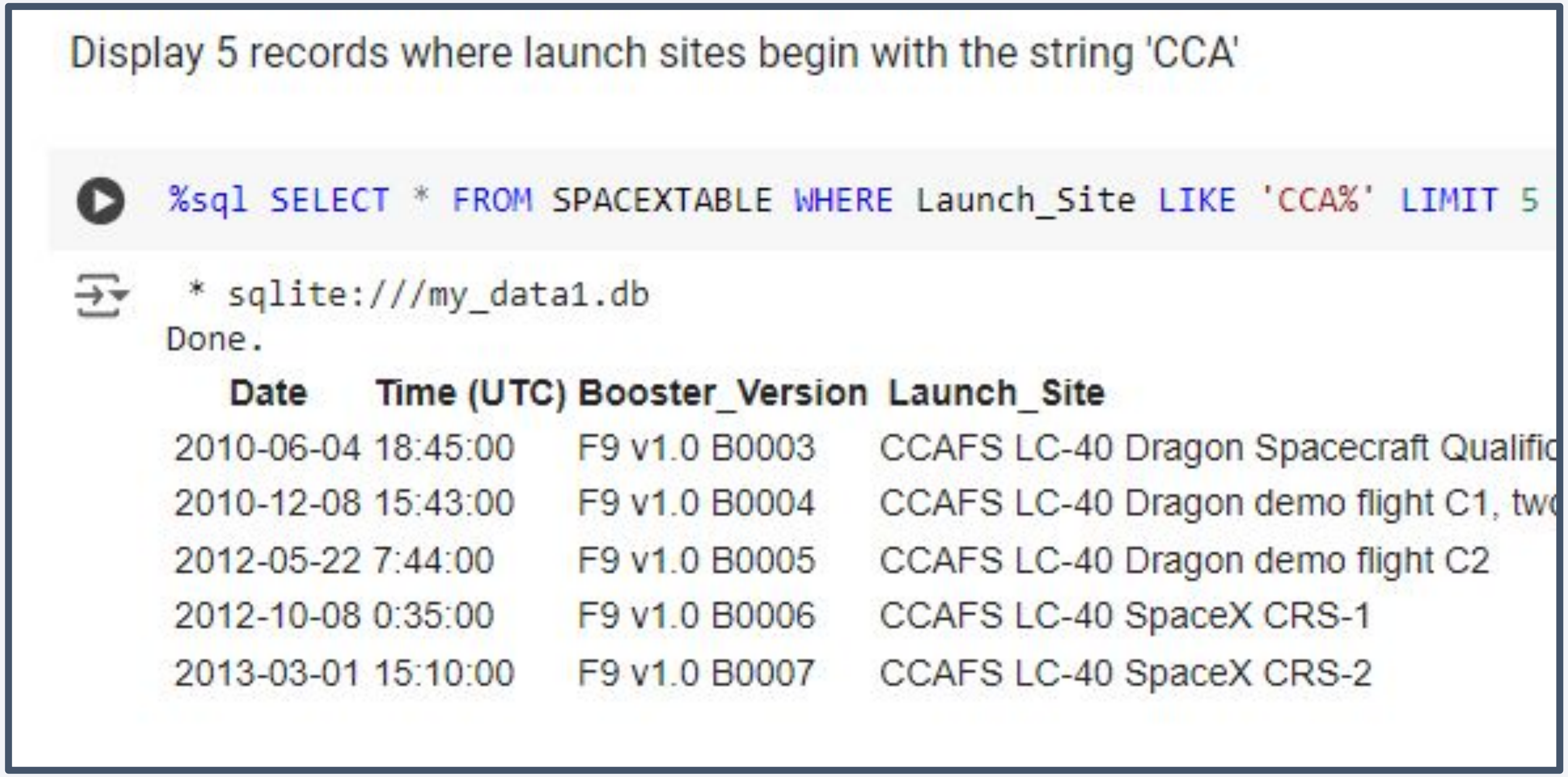
Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

```
 * sqlite:///my_data1.db
Done.
 Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

Retrieving records from a specific launch site:



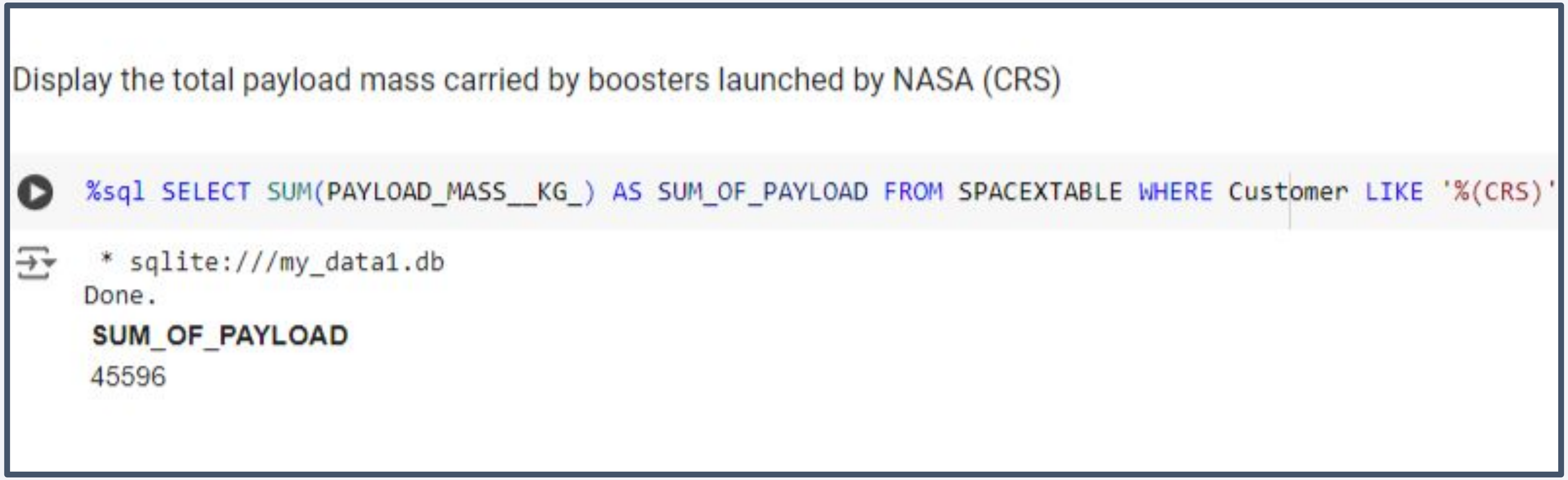
Display 5 records where launch sites begin with the string 'CCA'

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

```
 * sqlite:///my_data1.db
Done.
    Date      Time (UTC) Booster_Version  Launch_Site
2010-06-04 18:45:00     F9 v1.0 B0003    CCAFS LC-40 Dragon Spacecraft Qualific
2010-12-08 15:43:00     F9 v1.0 B0004    CCAFS LC-40 Dragon demo flight C1, two
2012-05-22 7:44:00      F9 v1.0 B0005    CCAFS LC-40 Dragon demo flight C2
2012-10-08 0:35:00      F9 v1.0 B0006    CCAFS LC-40 SpaceX CRS-1
2013-03-01 15:10:00     F9 v1.0 B0007    CCAFS LC-40 SpaceX CRS-2
```

Total Payload Mass

Retrieving the sum of payload carried by boosters launched by NASA:



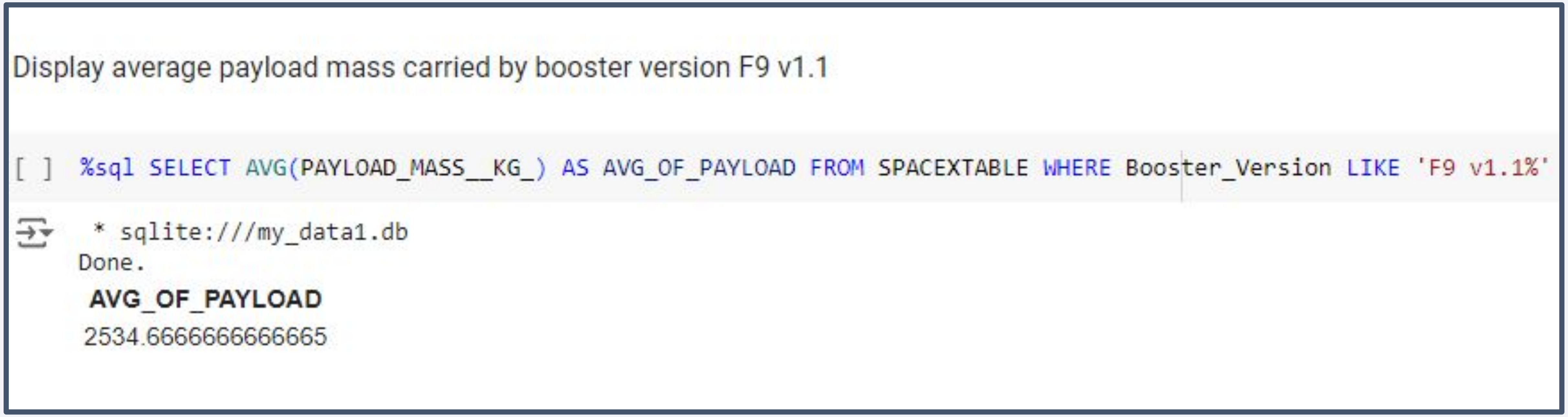
Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_OF_PAYLOAD FROM SPACEXTABLE WHERE Customer LIKE '%(CRS)'
```

```
 * sqlite:///my_data1.db
Done.
SUM_OF_PAYLOAD
45596
```

Average Payload Mass by F9 v1.1

Retrieving the average payload for booster version F9 v1.1:



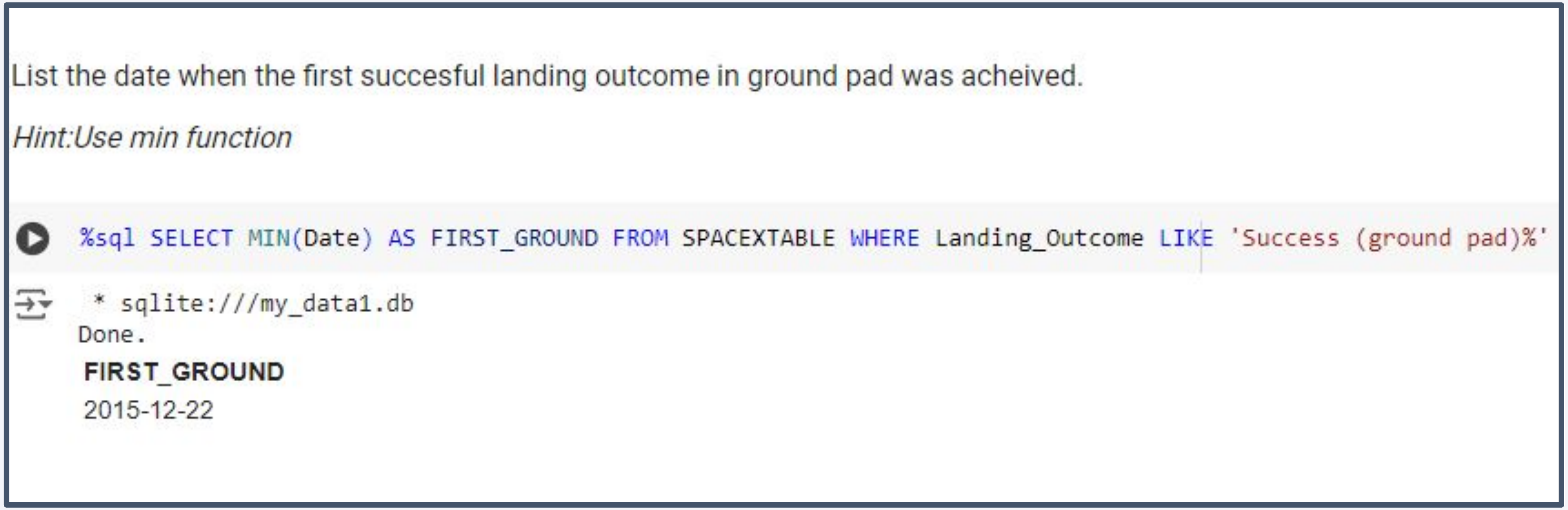
Display average payload mass carried by booster version F9 v1.1

```
[ ]  %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_OF_PAYLOAD FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

```
 * sqlite:///my_data1.db
Done.
AVG_OF_PAYLOAD
2534.6666666666665
```

First Successful Ground Landing Date

Retrieving the date of the first successful landing:



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

Hint:Use min function

```
%sql SELECT MIN(Date) AS FIRST_GROUND FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)%'
```

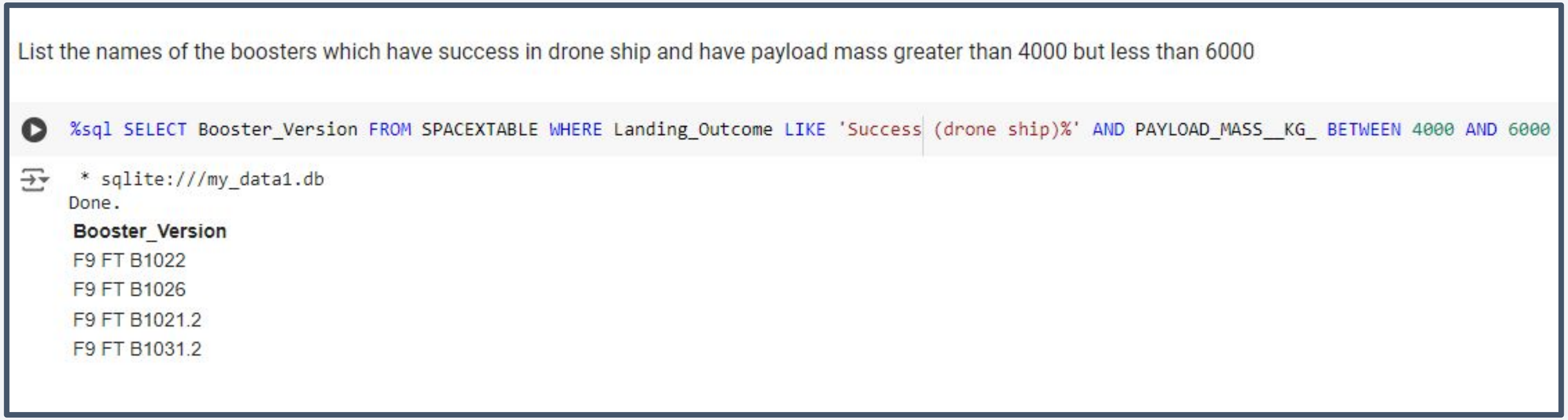
```
 * sqlite:///my_data1.db
Done.
```

FIRST_GROUND

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Retrieving the list of booster that had success with greater payloads:



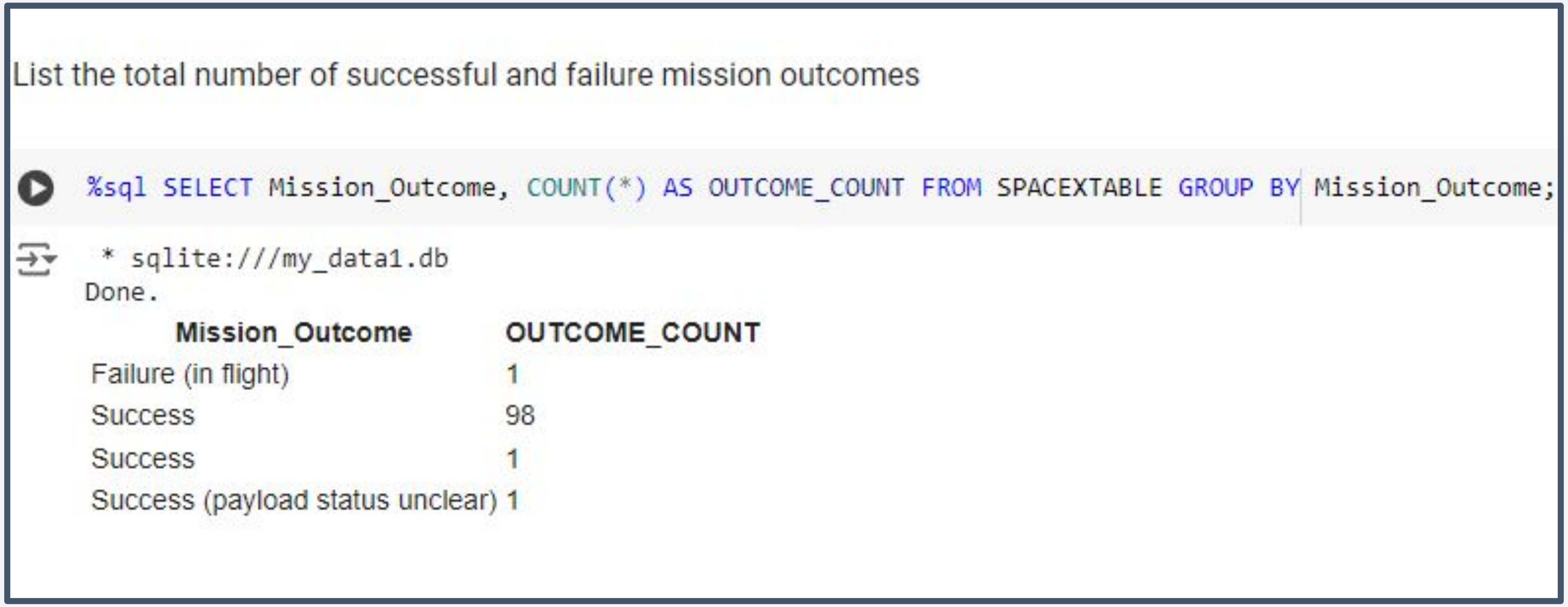
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (drone ship)%' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

```
 * sqlite:///my_data1.db
Done.
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2
```

Total Number of Successful and Failure Mission Outcomes

Total outcome:


List the total number of successful and failure mission outcomes

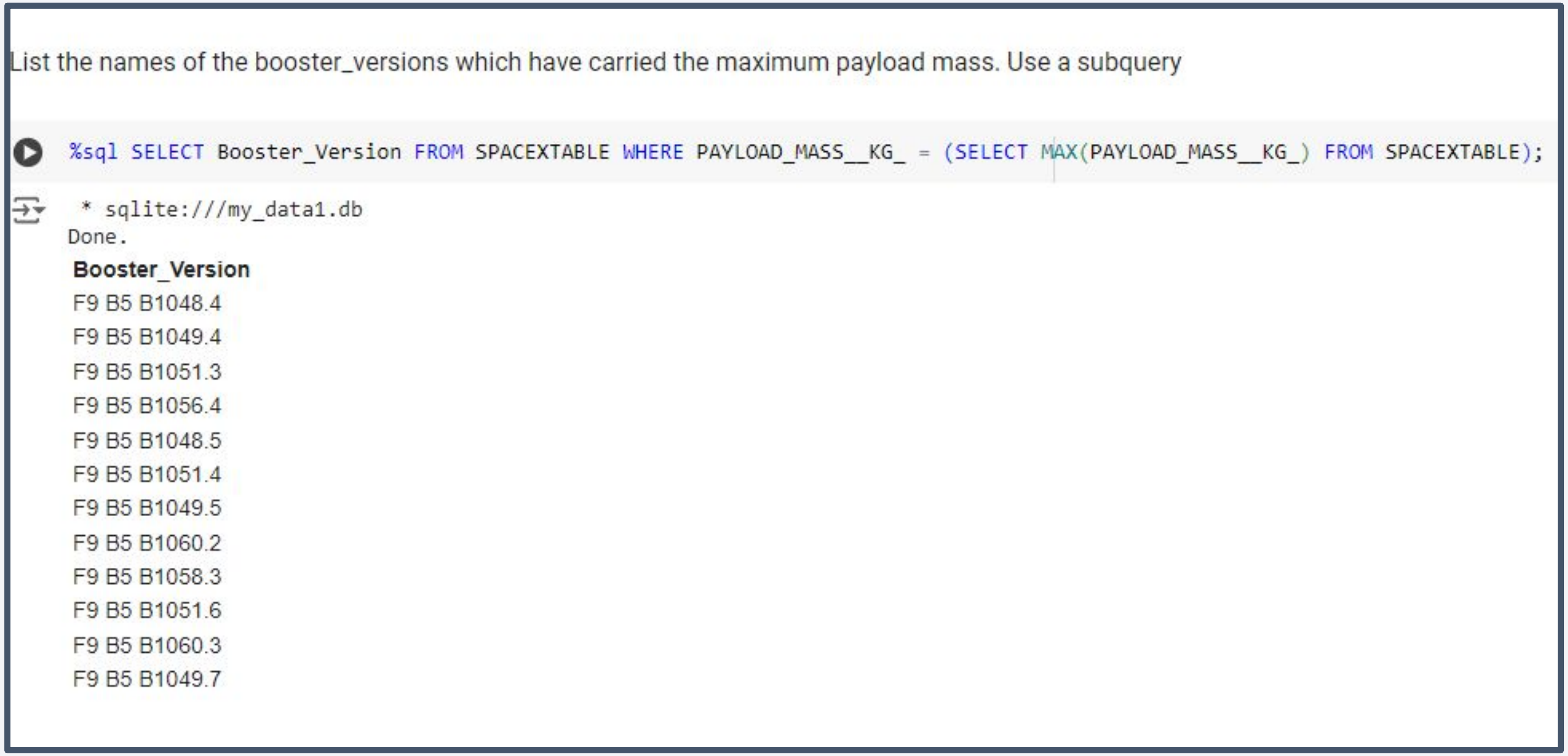
```
%sql SELECT Mission_Outcome, COUNT(*) AS OUTCOME_COUNT FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

Boosters which carried the max payload:



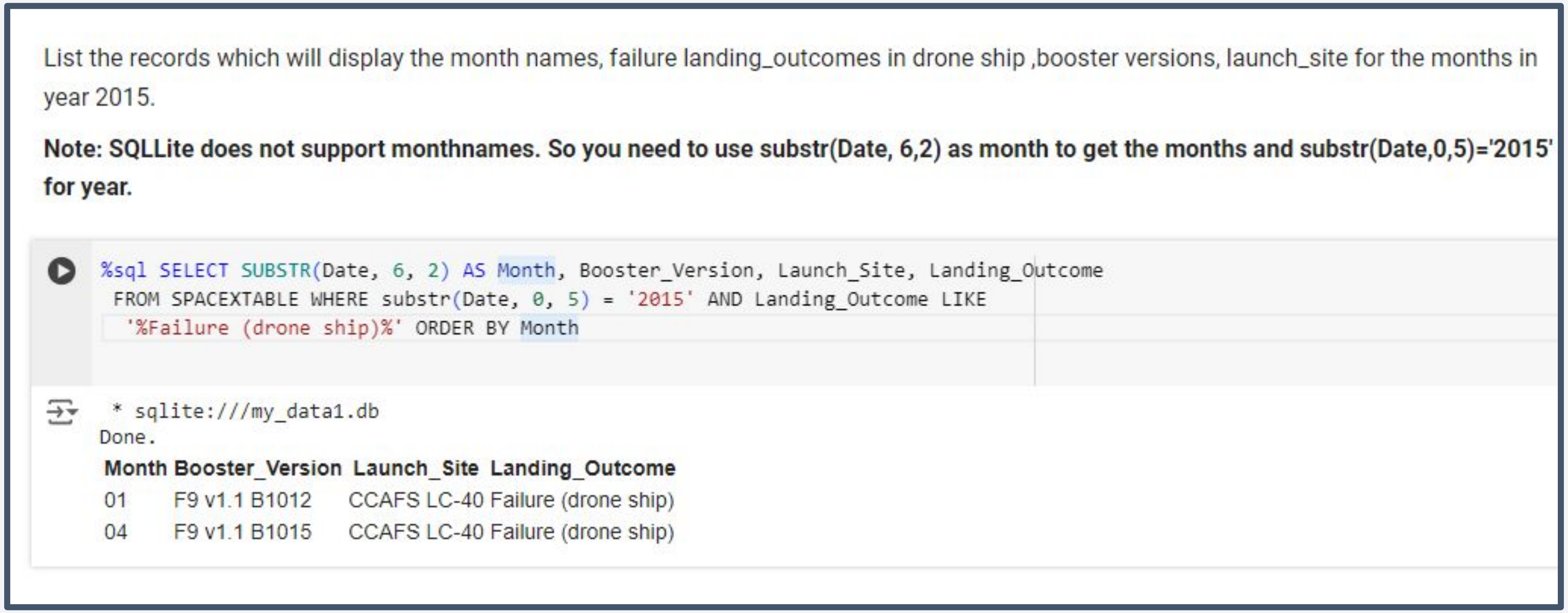
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%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

2015 launches:



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.

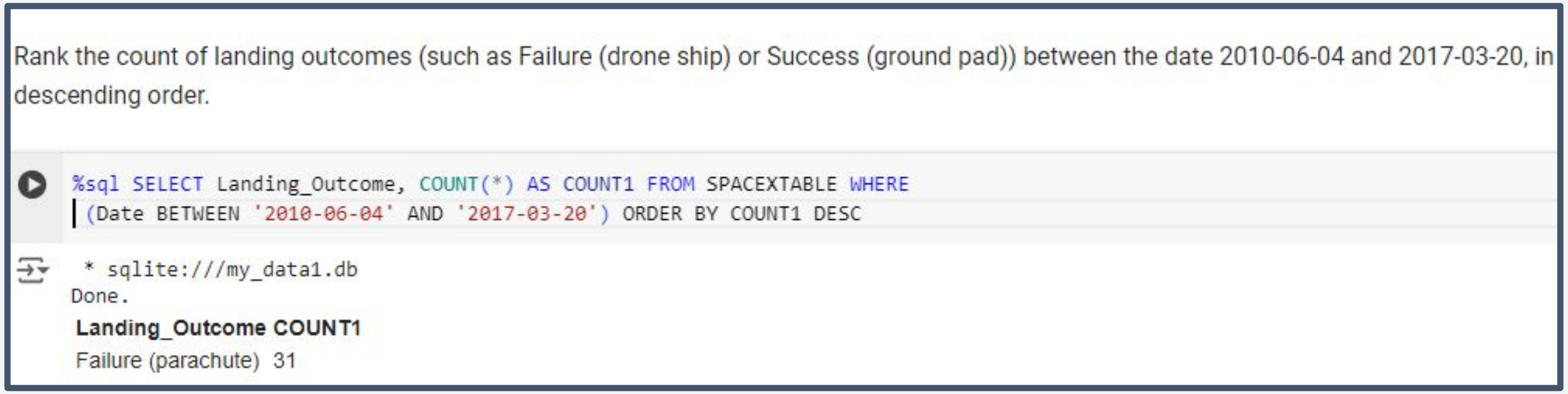
Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT SUBSTR(Date, 6, 2) AS Month, Booster_Version, Launch_Site, Landing_Outcome
 FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome LIKE
  '%Failure (drone ship)%' ORDER BY Month
```

```
 * sqlite:///my_data1.db
Done.
Month Booster_Version  Launch_Site  Landing_Outcome
01     F9 v1.1 B1012    CCAFS LC-40 Failure (drone ship)
04     F9 v1.1 B1015    CCAFS LC-40 Failure (drone ship)
```

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

Landing outcomes:



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.

```
%sql SELECT Landing_Outcome, COUNT(*) AS COUNT1 FROM SPACEXTABLE WHERE
(Date BETWEEN '2010-06-04' AND '2017-03-20') ORDER BY COUNT1 DESC
```

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

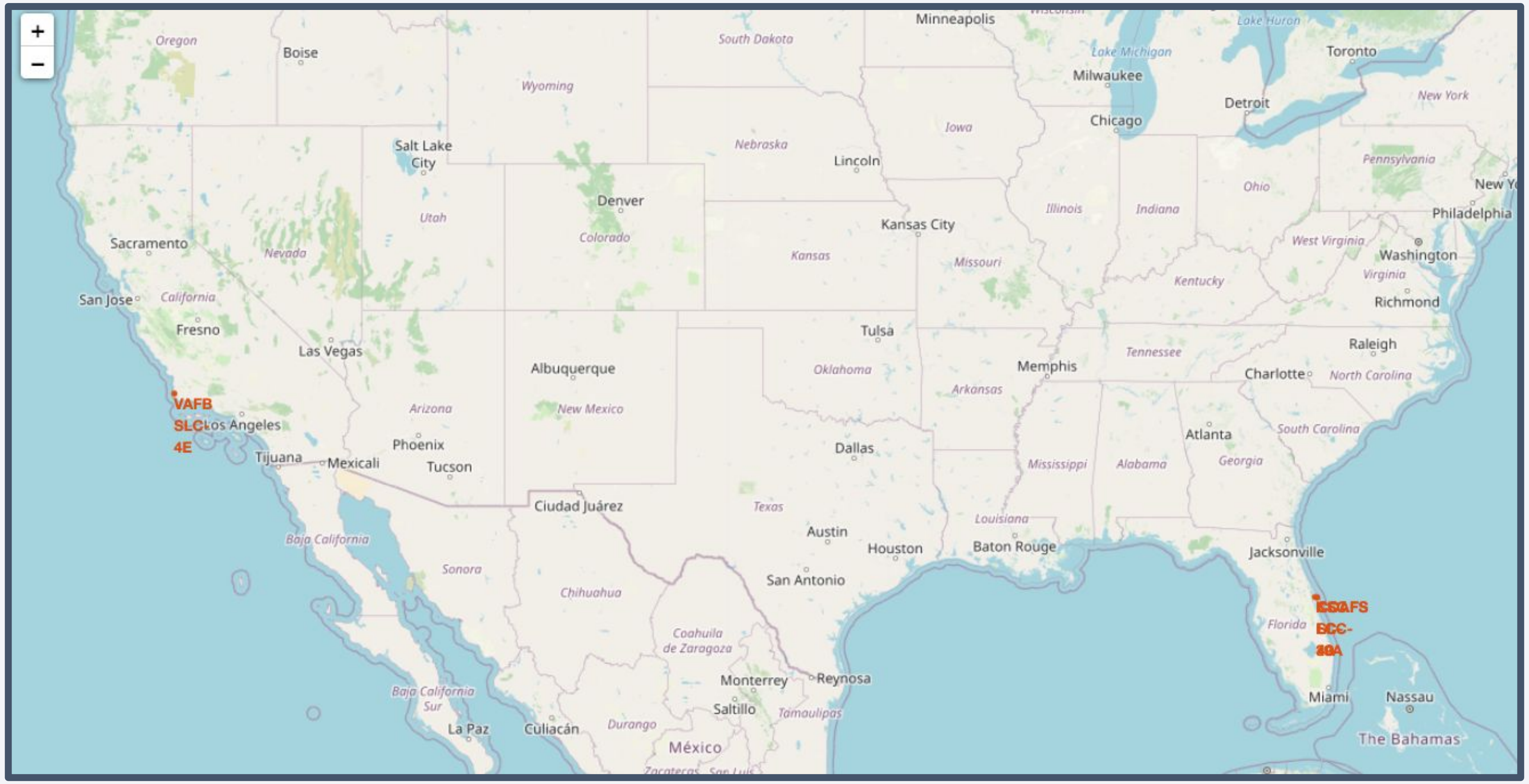
Landing_Outcome	COUNT1
Failure (parachute)	31

Section 3

Launch Sites Proximities Analysis

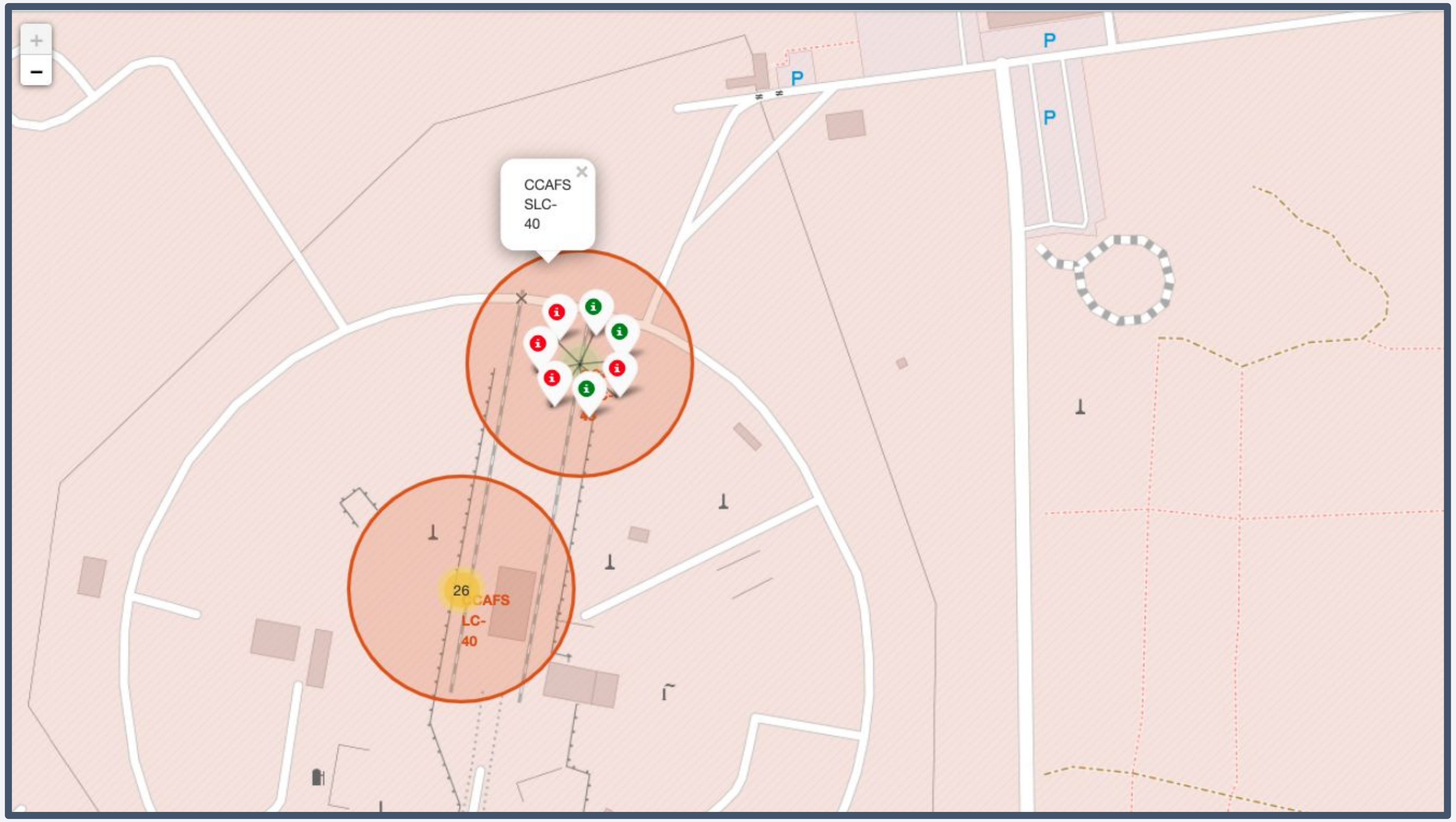
Folium Map Screenshot

All launch sites marked on the map:



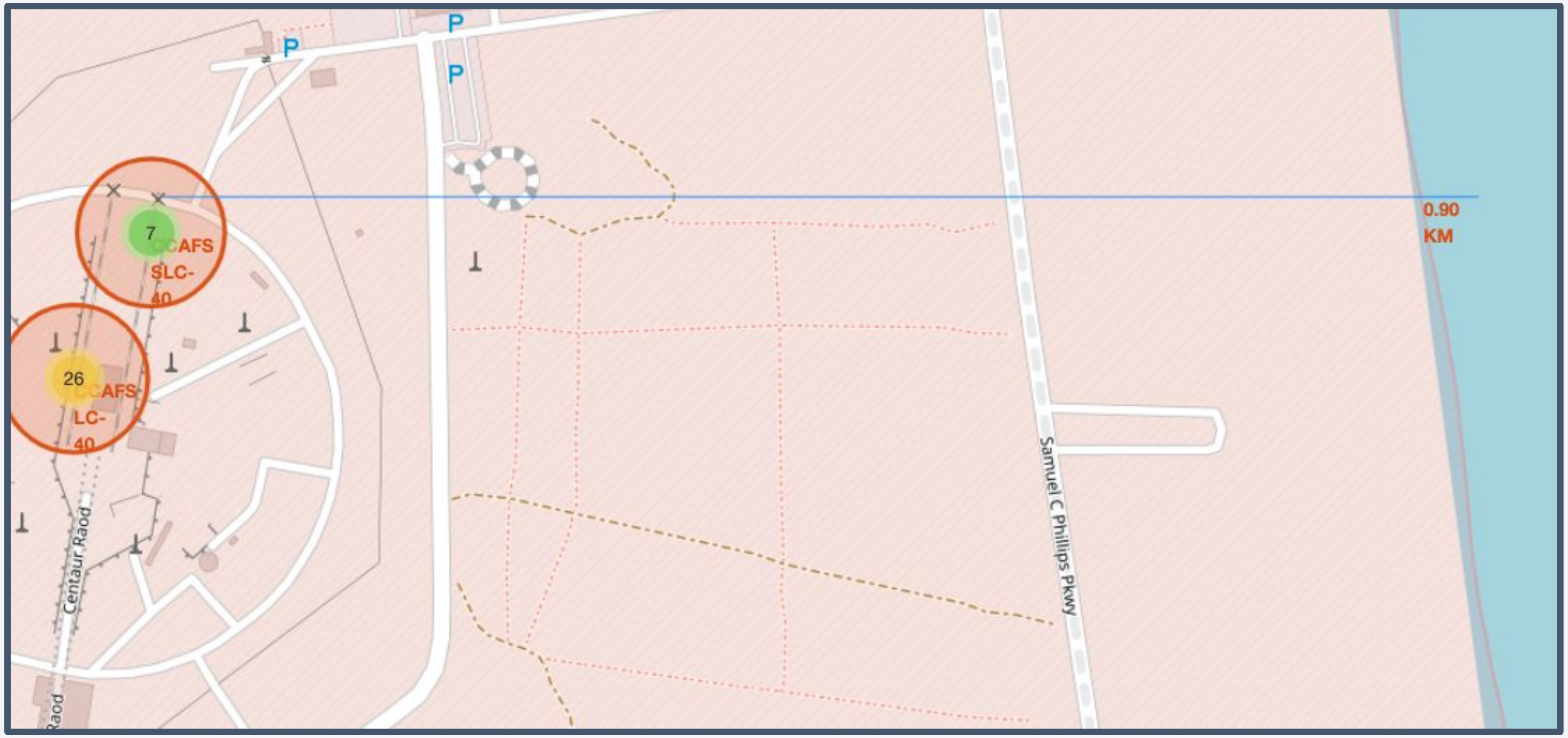
Folium Map outcomes

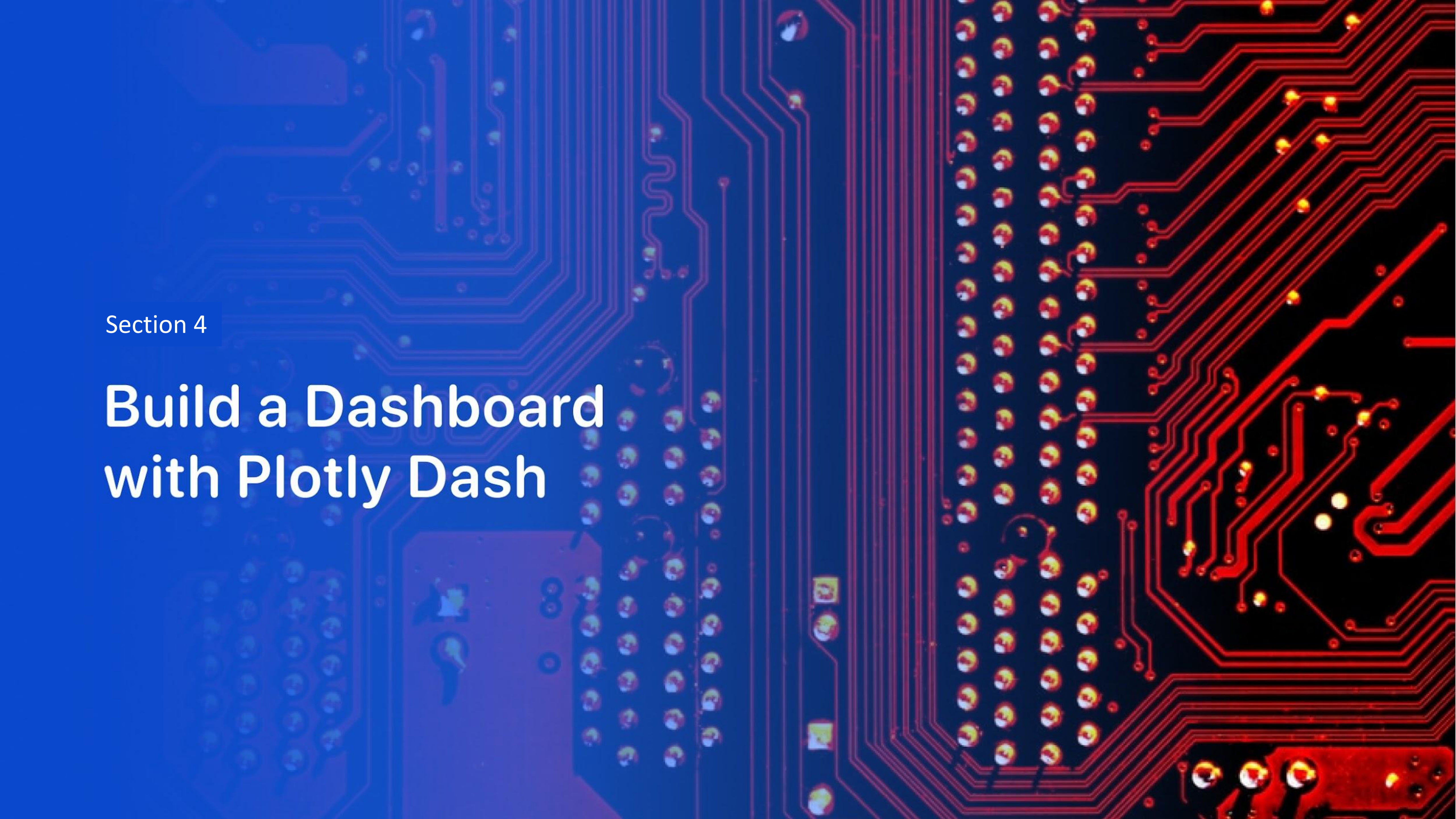
Successful outcomes marked as green and failed as red:



Folium Map measuring distance

Distance calculated from launch site to coast:



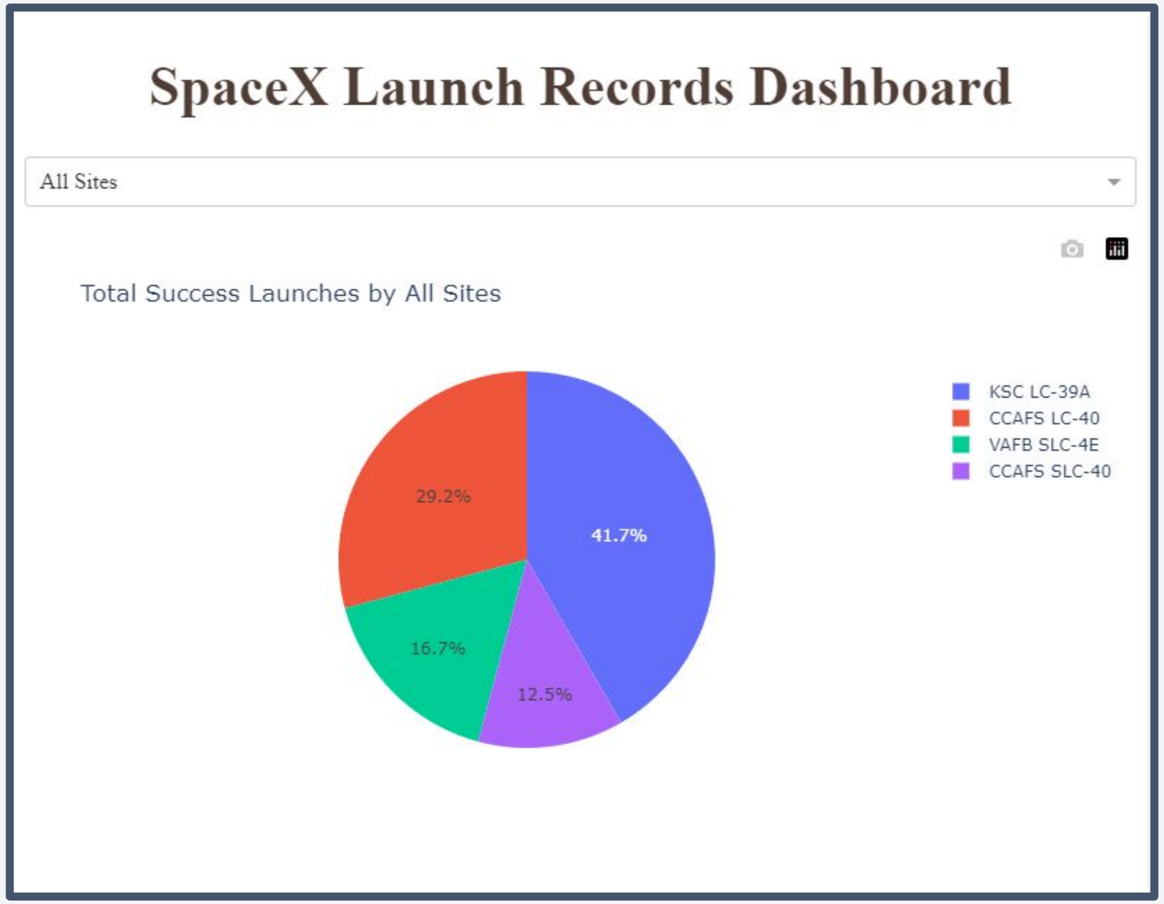


Section 4

Build a Dashboard with Plotly Dash

Dash dashboard

Successful launches:

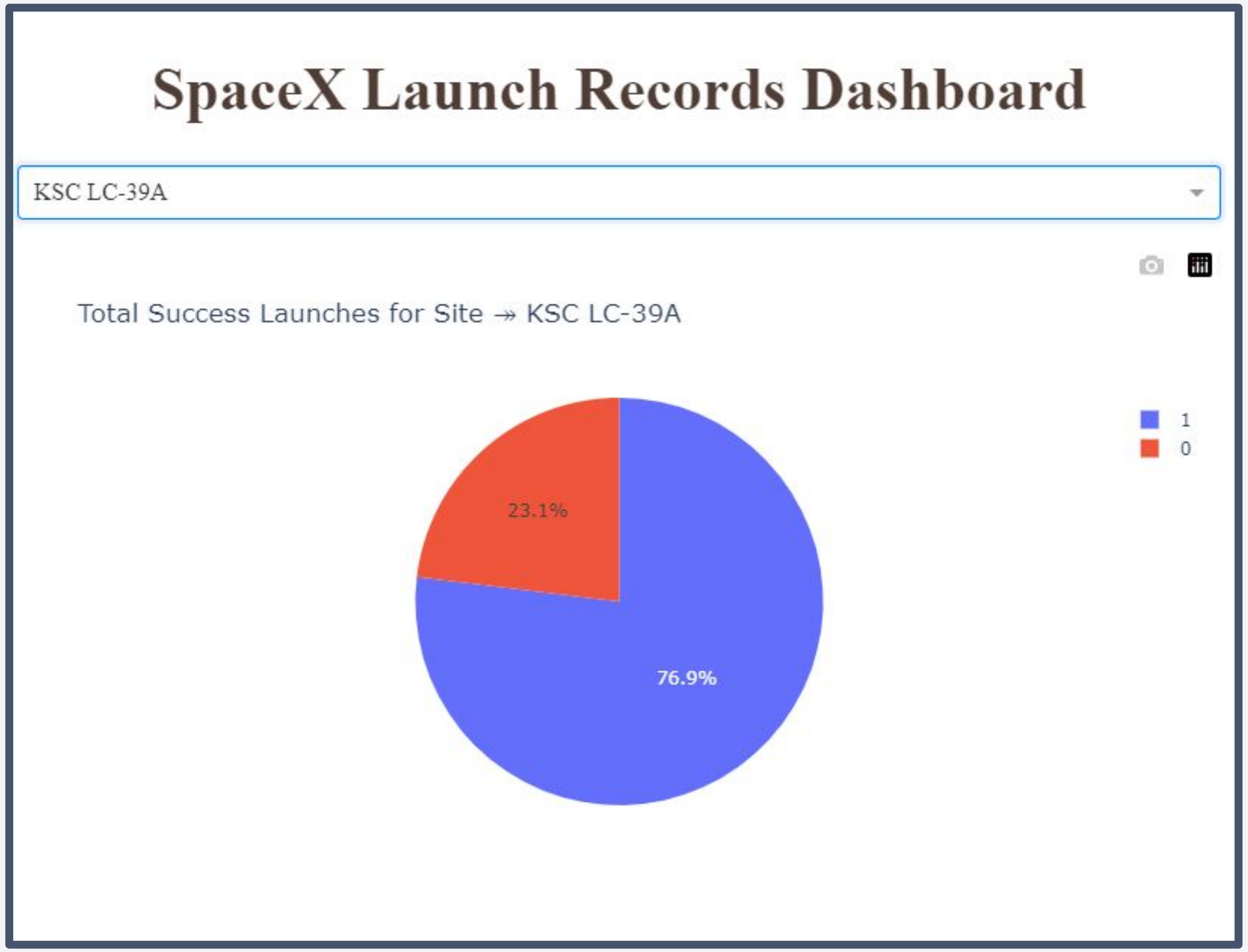

SpaceX Launch Records Dashboard

All Sites

Total Success Launches by All Sites

Dashboard highest launch rate

KSC LC-39A has the highest success rate among launch site:

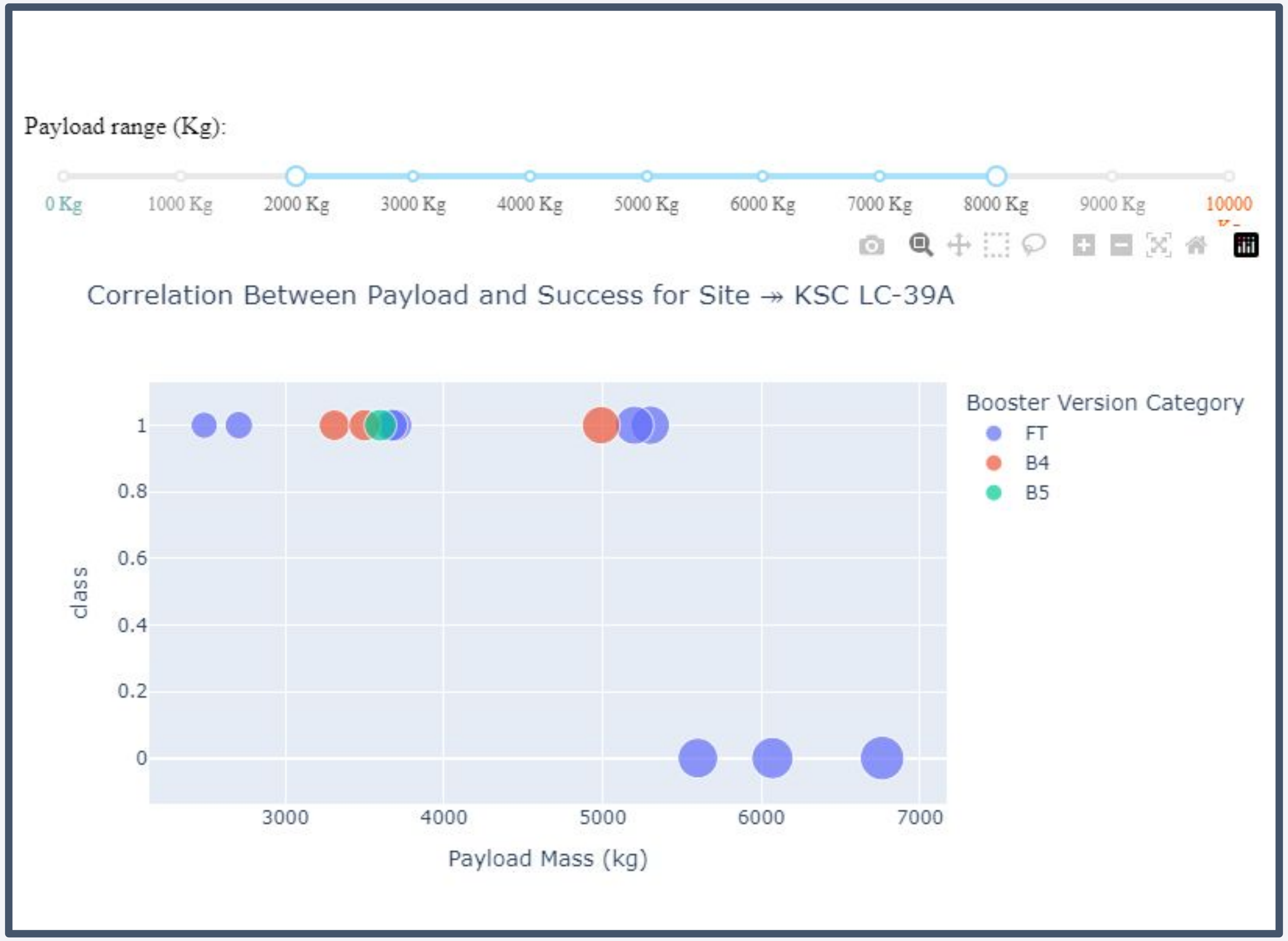

SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for Site → KSC LC-39A

Dashboard scatter

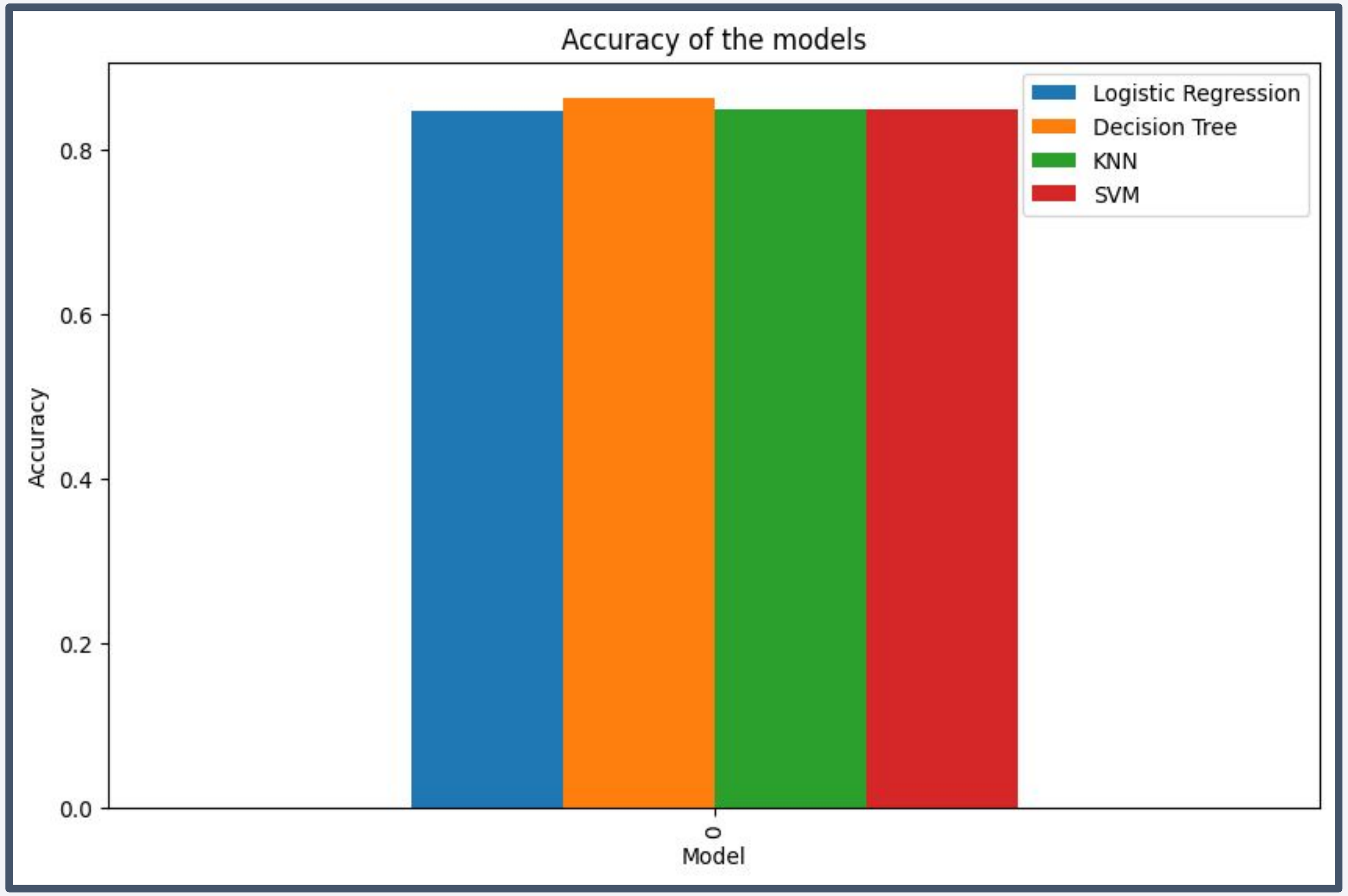
Scatter plot showing correlation between payload mass and success:



Section 5

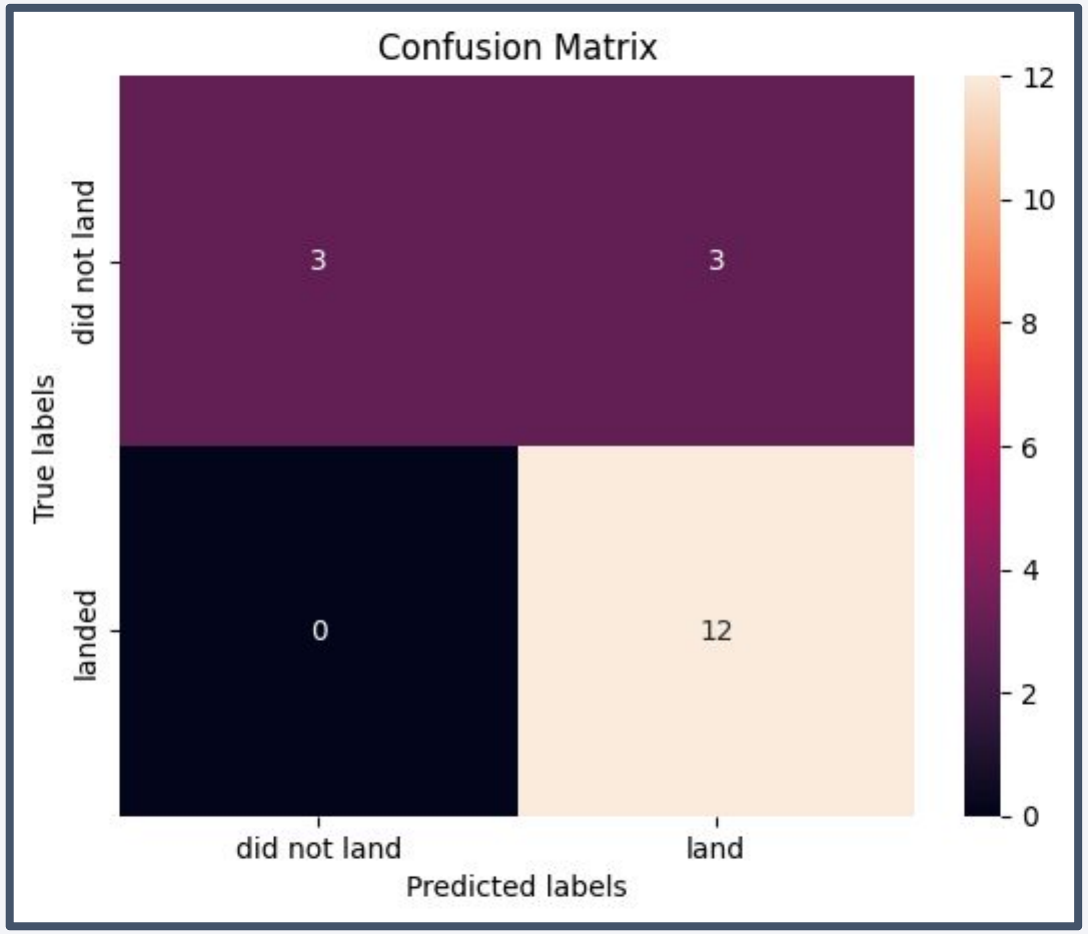
Predictive Analysis (Classification)

Classification Accuracy



Looking at this bar chart it is possible to see that all models tested performed almost exactly the same, except for decision tree which performed slightly better.

Confusion Matrix



- True positives = 12, bottom right corner
- False positives = 3, top left corner
- True negatives = 3, top right corner
- False negatives = 0, bottom left corner

Conclusions

- Launch sites are always close to the coast and most launch sites are close to the equator line.
- The ML model has a good accuracy when predicting the outcome of a mission.
- Launch success is always increasing over time.
- Higher payload masses generally means higher success rate.
- KSC LC-39A has the highest success rate among launch sites.

Appendix

- Github repo : https://github.com/awesomegab/IBM-DS-Capstone
- SpaceX API repo: https://github.com/r-spacex/SpaceX-API
- OK

Thank you!