IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Robert Malvin

November 30, 2023

https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project.git

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- SpaceX has a commanding lead in commercial space launches
  - SpaceX has a significant price advantage charging $62M per launch vs $165M competitors charge
  - A driver of their price advantage is the reuse of stage 1 of their rockets
- Key questions to better understand SpaceX reuse of stage 1
  - How often does SpaceX reuse stage 1
  - What are common characteristics of launches where stage 1 is landed safely for reuse
  - What are common characteristics of launches where stage 1 crashes
  - Create an algorithm that can be used to determine if stage 1 will land successfully

Section 1

Methodology

Methodology

Executive Summary

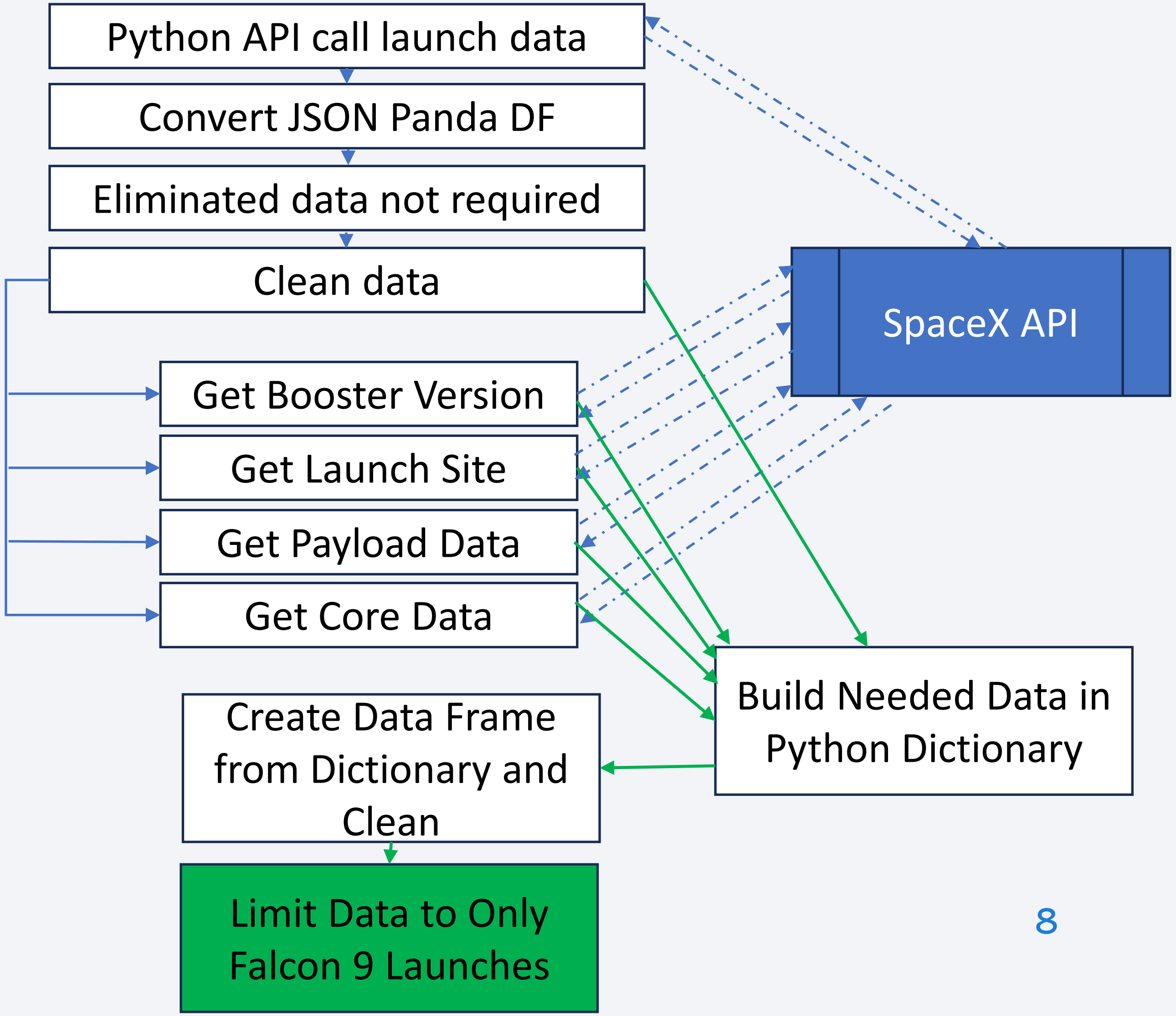
- Data collection methodology:
    - JSON download - SpaceX REST API: api.spacexdata.com/v4/
    - Web scraping - Wikipedia: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Perform data wrangling
    - Extract relevant data and transform into panda dataframe
    - Remove unusable data, correct datatypes, fix missing values
    - Classify each launch as success or failure
- Perform exploratory data analysis (EDA) using visualization and SQL
    - Explore data elements and relationships to success or failure of landing stage 1
        - Examples of elements explored: Payload, Orbit, Date (flightnumber), Launch Site
- Perform interactive visual analytics using Folium and Plotly Dash
    - Map launch sites and success/failure rates at each site
    - Build interactive dashboard to explore: Launch sites, Payload, Booster Version in relation to success/failure
- Perform predictive analysis using classification models
    - Build Logistic, Know Nearest Neighbor, Support Vector Machine, and Decision Tree
    - Use Grid Search to test various parameters and determine best model of each type
    - Evaluate the performance of best models against each other using accuracy and confusion matrix

Data Collection

- Data collection methodology:
  - JSON download - SpaceX REST API: api.spacexdata.com/v4/
  - Web scraping - Wikipedia: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

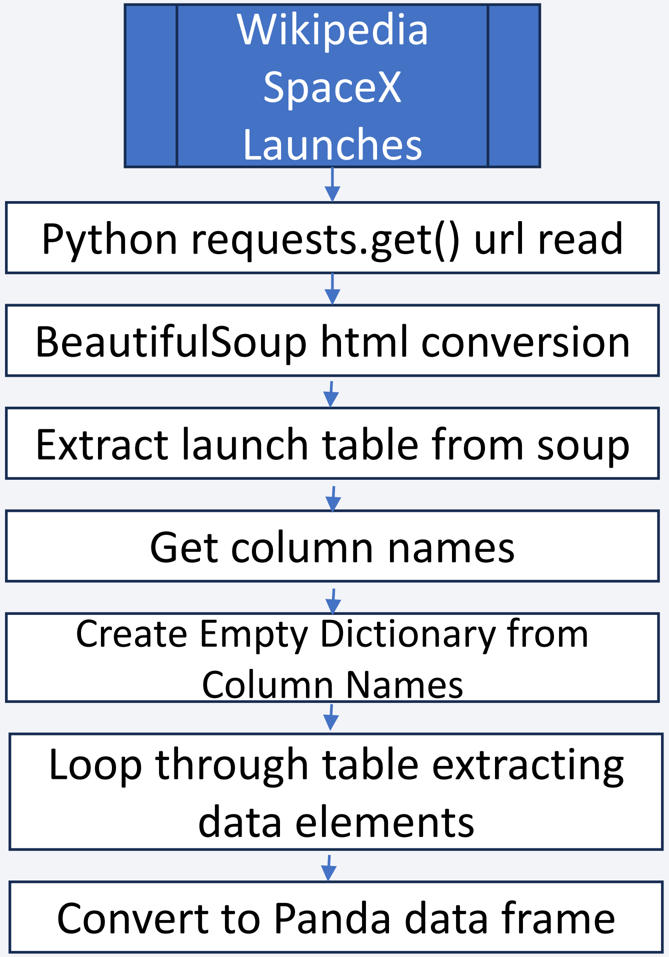
- SpaceX REST API call to get past launch data
- Convert API JSON response to Panda data frame
- Eliminate data not needed keeping: rocket, payloads, launchpad, cores, flight_number, date
- Cleanse cores, payload and date
- Using calls to SpaceX retrieve additional data: booster name, payload mass in kg, launchsite latitude and longitude, outcome of landing, type of landing, number of previous core uses, gridfins used, legs used, landing pad used, core version
- GitHub URL of the completed SpaceX API calls notebook: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/01%20jupyter-labs-spacex-data-collection-api.ipynb



Python API call launch data
Convert JSON Panda DF
Eliminated data not required
Clean data
SpaceX API
Get Booster Version
Get Launch Site
Get Payload Data
Get Core Data
Build Needed Data in Python Dictionary
Create Data Frame from Dictionary and Clean
Limit Data to Only Falcon 9 Launches

Data Collection - Scraping

- SpaceX REST API call to get past launch data
- Convert API JSON response to Panda data frame
- Eliminate data not needed keeping: rocket, payloads, launchpad, cores, flight_number, date
- Cleanse cores, payload and date
- Using calls to SpaceX retrieve additional data: booster name, payload mass in kg, launchsite latitude and longitude, outcome of landing, type of landing, number of previous core uses, gridfins used, legs used, landing pad used, core version
- GitHub URL of the completed Wikipedia scaping: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/02%20jupyter-labs-webscraping.ipynb



Wikipedia SpaceX Launches

Python requests.get() url read

BeautifulSoup html conversion

Extract launch table from soup

Get column names

Create Empty Dictionary from Column Names

Loop through table extracting data elements

Convert to Panda data frame

Data Wrangling

- Objectives
  - Exploratory Data Analysis
  - Determine Training Labels
- Steps
  - Evaluate percent of each variable with missing data
  - Review data types of each variable
  - Count of launches by: Launchsite, Orbit, Outcome
  - Create landing class (success/fail) based on Outcome
  - Calculate success rate
- Add the GitHub URL of your completed data wrangling related notebooks: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/03%20labs-jupyter-spacex-Data%20wrangling.ipynb

Key Findings

- 29% of records missing landing pad
- Many variables are type object will need to convert to dummy variables latter for modeling
- CCAFS SLC40 was site for 55 of 90 launches
- GTO and ISS (space station) were most common orbit
- 67% of landings were successful
- This was included in place of a flowchart… it seemed more useful

EDA with Data Visualization

- Visualizations used to understand data and data relationships
  - Scatterplot: Flight Number by Payload Mass and Class (success/failure)
  - Scatterplot: Flight Number by Launch Site and Class (success/failure)
  - Scatterplot: Payload Mass by Launch Site and Class (success/failure)
  - Bar chart: Success Rate by Orbit
  - Scatterplot: Flight Number by Orbit and Class (success/failure)
  - Line chart: Payload Mass by Orbit and Class (success/failure)
  - Scatterplot: Success Rate by Year
- Add the GitHub URL of your completed EDA with data visualization notebook: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/05%20jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
    - Unique list of Launch Sites
    - 5 records where launch site begins with ‘CCA’
    - Total sum of payloads for customer NASA (CRS)
    - Average payload for booster version F9 v1.1
    - Date of first successful landing
    - Boosters with success in drone ship and payload between 4,000 and 6,000
    - Mission outcome counts
    - Booster versions that have carried max payload
    - Month, booster version, and launch site of of drone ship failures in 2015
    - Landing outcomes between June 4, 2010 and March 20, 2017
- Add the GitHub URL of your completed EDA with SQL notebook: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/04%20jupyter-labs-eda-sql-coursera_sqllite%20lab%20environment.ipynb

Build an Interactive Map with Folium

- Mapped launch sites with circle markers
  - Visual understanding of where launches occur
- Added marker clusters to show red for failed landing and green for successful landing
  - Create and easy non cluttered visual of success/failure of landing from each launch site
- Mapped distance to coastline, nearest railroad, nearest highway and nearest city
  - Launch sites are near coastlines for safety reasons and railways for shipping logistics
  - Are farther from highways and cities for safety
- Plotted line with distance for each
- Add the GitHub URL of your completed interactive map with Folium map: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/06%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboard with user selections for:
  - All Launch Sites and Each Individual Launch Site
  - Payload range
- Pie Chart to visualize success rate
  - Across launch sites and for each launch site
- Scatterplot to visualize success rate for payload ranges by booster version
- Add the GitHub URL of your completed Plotly Dash lab: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/07%20spacex_dash_app.py

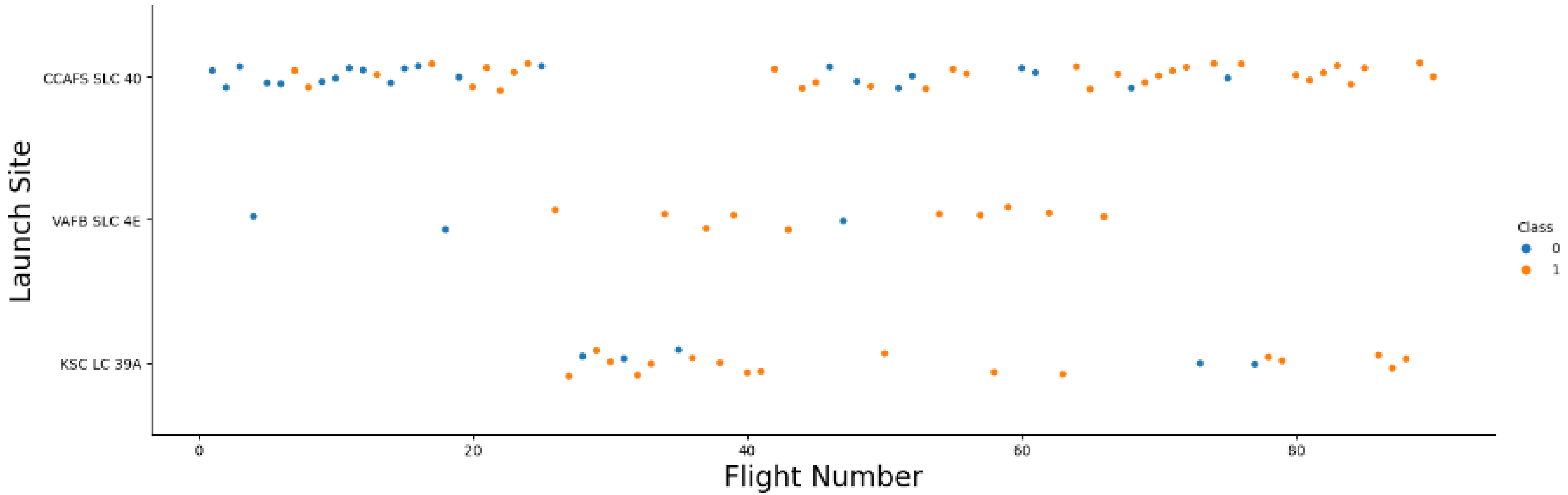
Predictive Analysis (Classification)

- Create a column for class
- Standardize data
- Split into training and test data
- Use GridSearchCV() to find best Hyperparameters for SVM, Classification Tree, Logistic Regression and KNN
- Evaluate which model performs the best
- Add the GitHub URL of your completed predictive analysis lab: https://github.com/rmalvin/IBM-Coursera-Data-Science-Capstone-Project/blob/main/08%20SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Section 2

Insights drawn from EDA

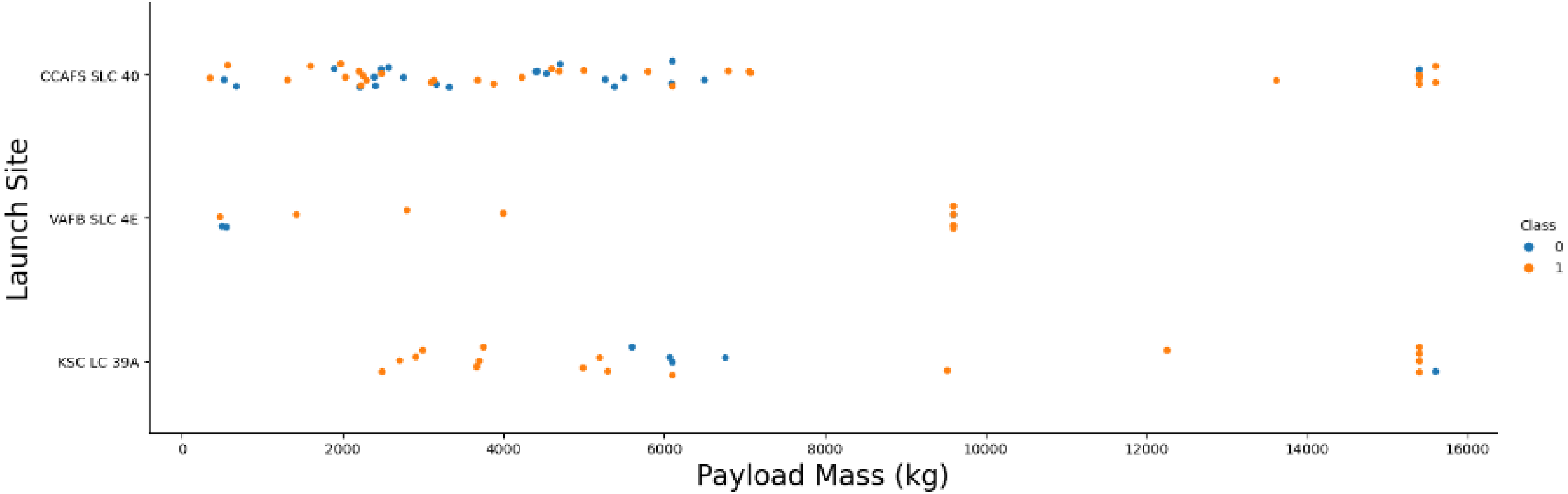
Flight Number vs Launch Site
Success rate increased over time at each launch site



- CCAFS SLC 40 has the most launches across all time and recent success rates are very high
- VAFB SLC 4E hasn’t been used as often or recently

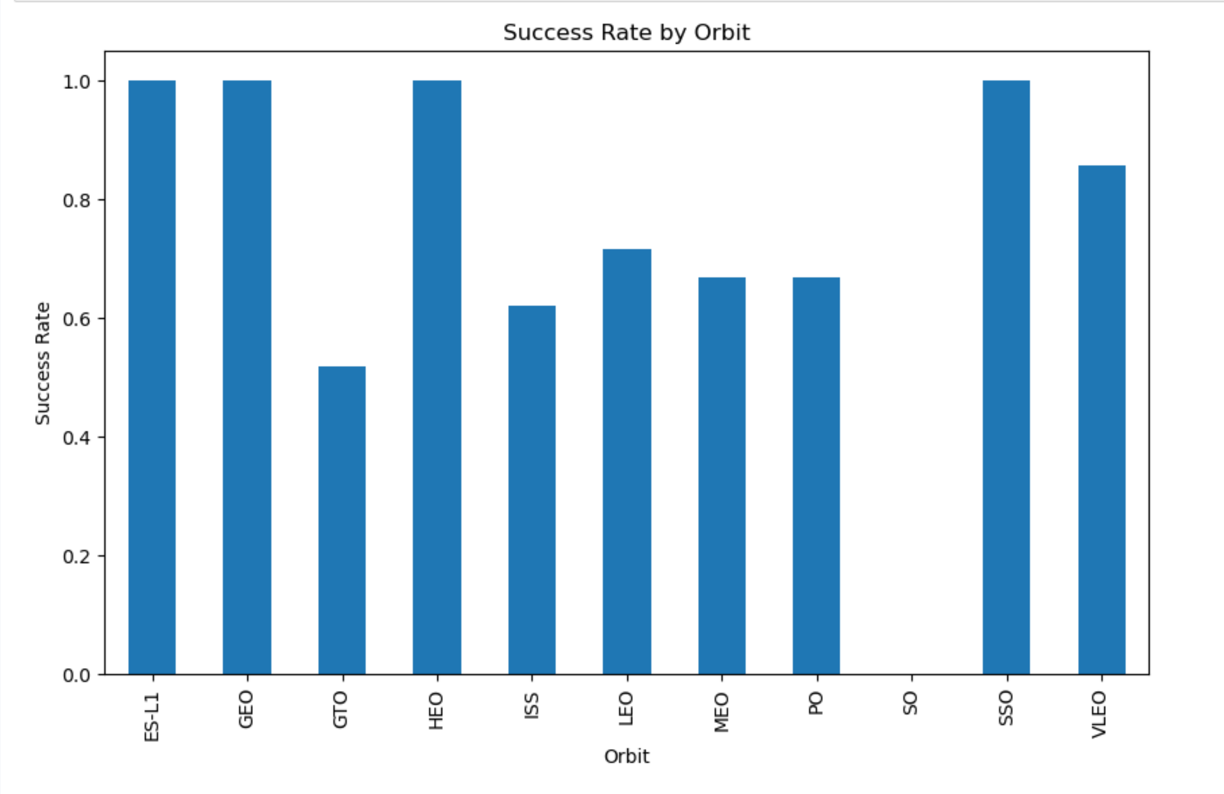
Payload vs. Launch Site

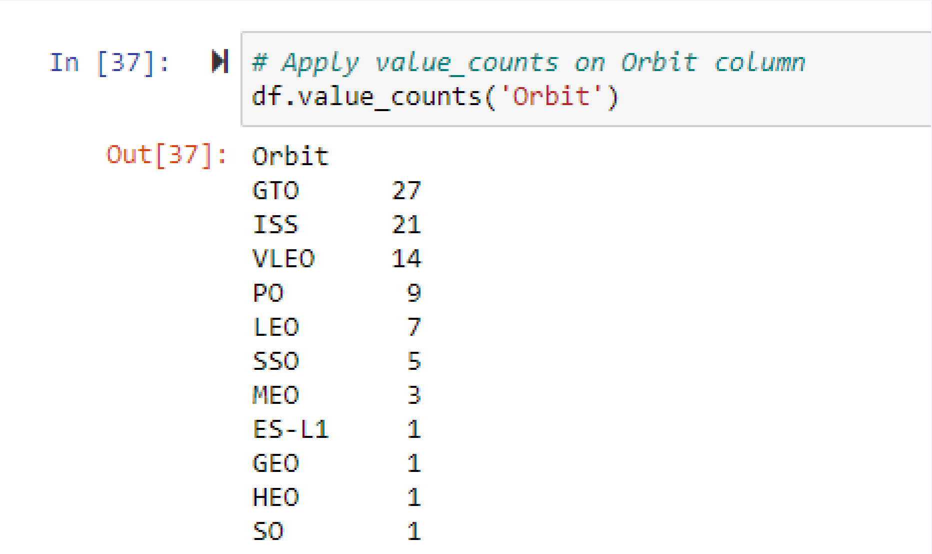
VAFB SLC 4E isn’t used for the largest payloads



- Payloads over 8,000 kg have high success rates

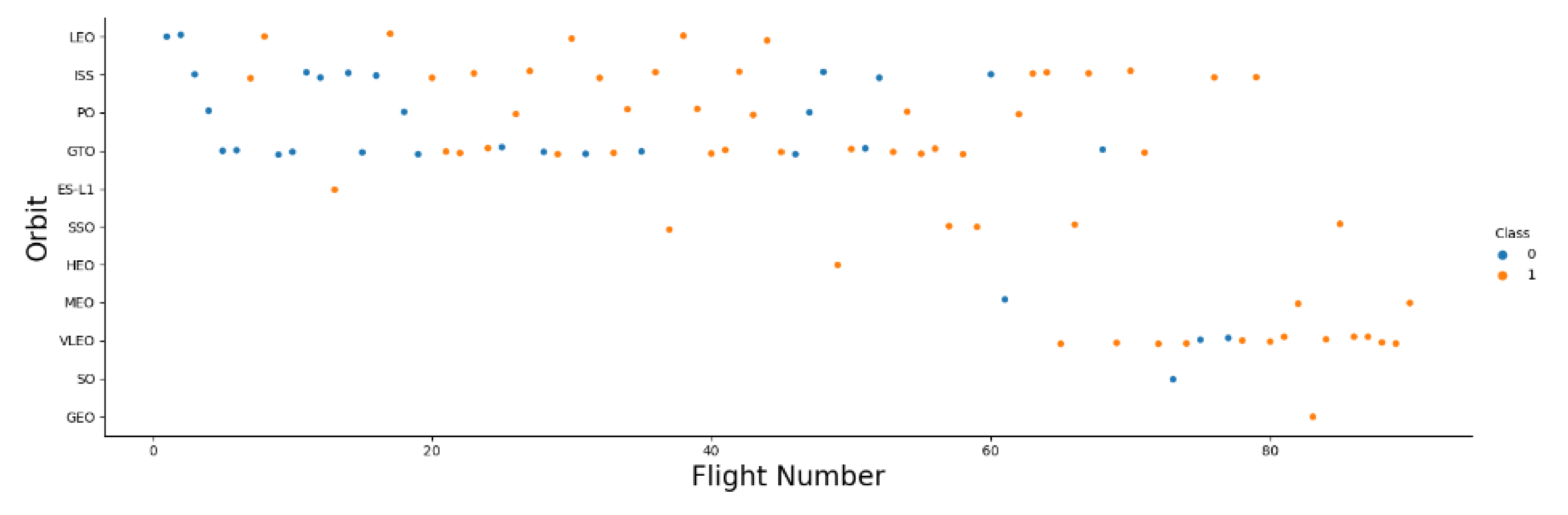
Success Rate vs. Orbit Type


Success Rate by Orbit


```
In [37]:  # Apply value_counts on Orbit column
          df.value_counts('Orbit')

Out[37]:  Orbit
          GTO      27
          ISS      21
          VLEO     14
          PO        9
          LEO       7
          SSO       5
          MEO       3
          ES-L1     1
          GEO       1
          HEO       1
          SO        1
```

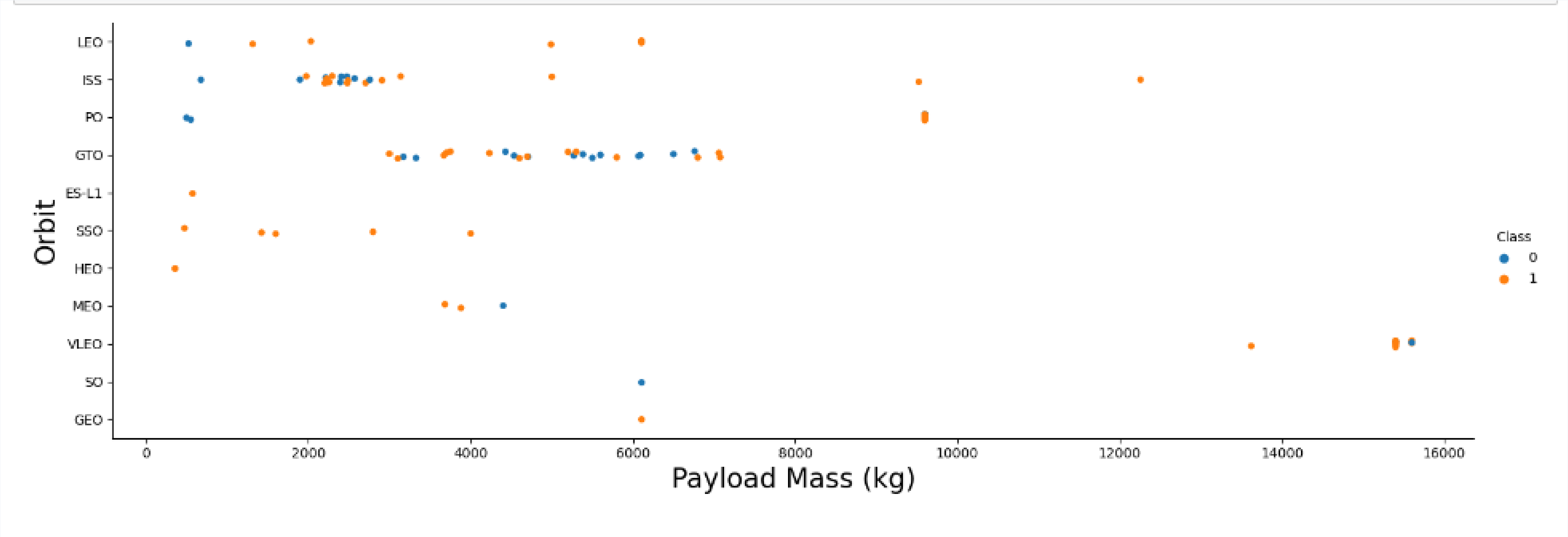
- Orbits with high success rates have very few launches

Flight Number vs. Orbit Type



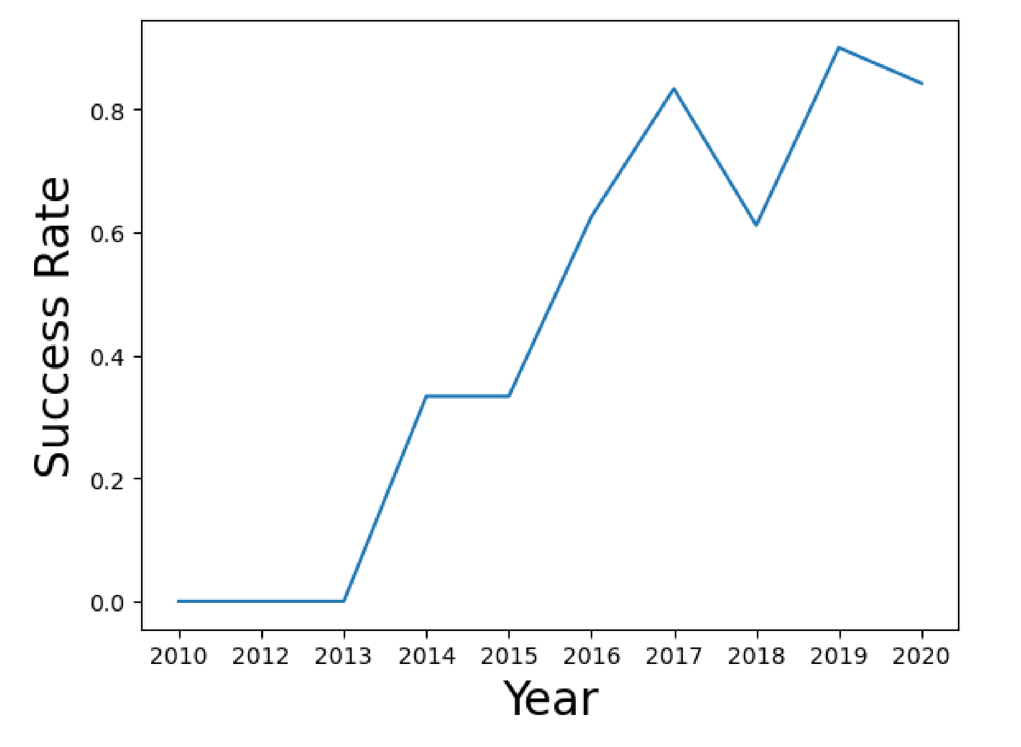
- Orbit VLEO is only part of more recent launches
- ISS orbit has been part of early and recent launches
- Success or failure in the GTO orbit doesn't appear to change with more recent flights

Payload vs. Orbit Type



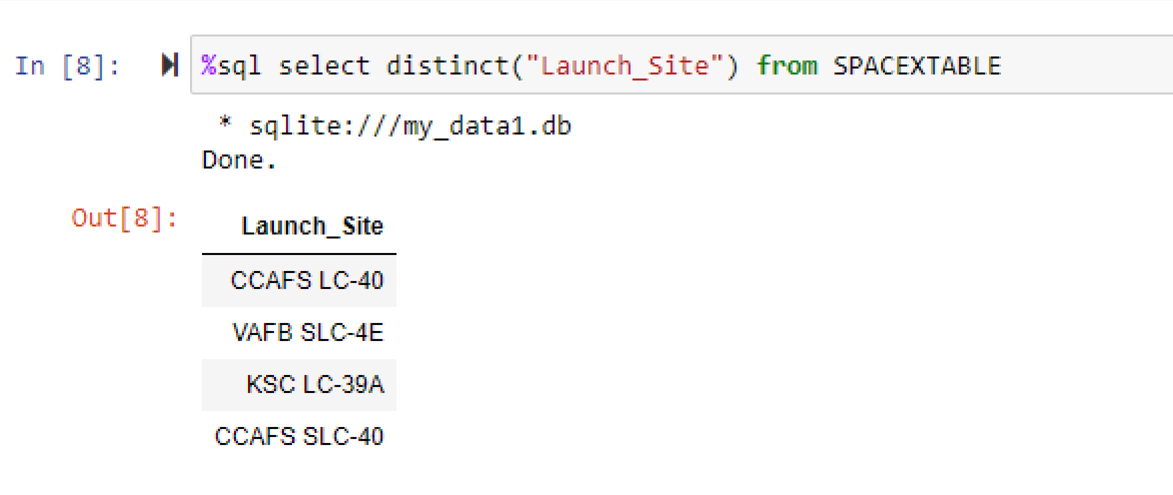
- Very heavy payloads seem to have mostly successful landings

Launch Success Yearly Trend



- Success has increased over time

All Launch Site Names



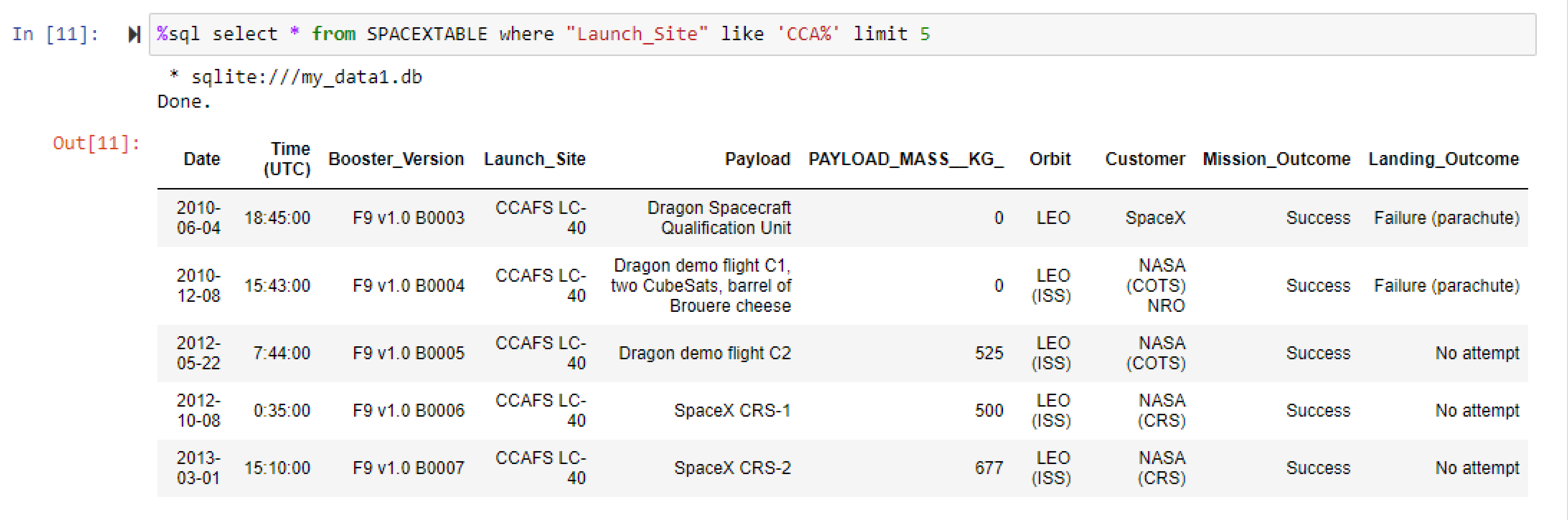
```
In [8]:  %sql select distinct("Launch_Site") from SPACEXTABLE

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

Out[8]:

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

- Used distinct to get list of launch sites

Launch Site Names Begin with 'CCA'



In [11]:

```
%sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
```

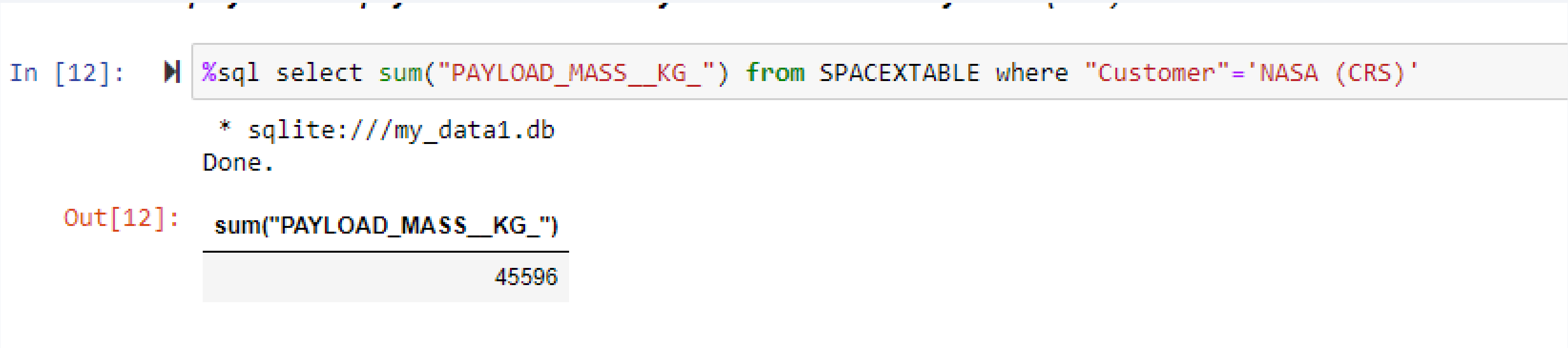
```
 * sqlite:///my_data1.db
Done.
```

Out[11]:

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

- Used like with '%' wildcard and limit

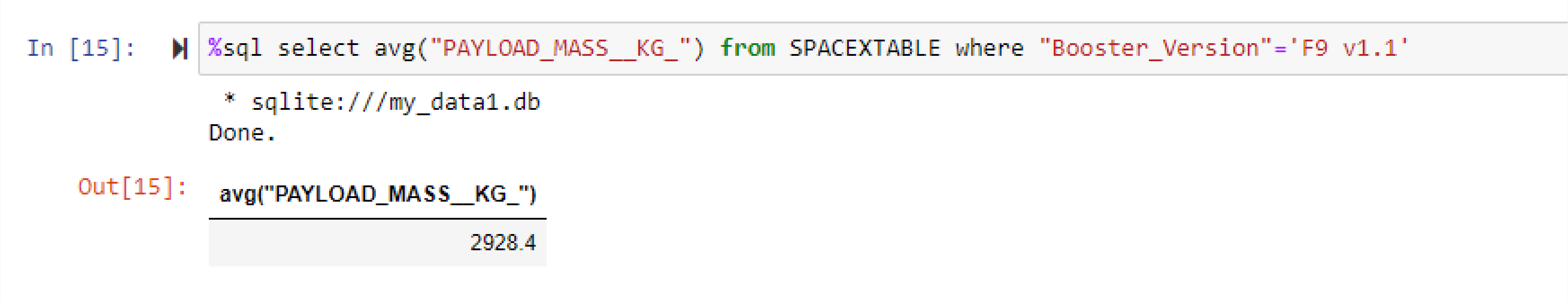
Total Payload Mass



```
In [12]:  %sql select sum("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Customer"='NASA (CRS)'

           * sqlite:///my_data1.db
          Done.

Out[12]:  sum("PAYLOAD_MASS__KG_")
                             45596
```

- Sum of Payload for specific customer value

Average Payload Mass by F9 v1.1



In [15]:

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version"='F9 v1.1'
```

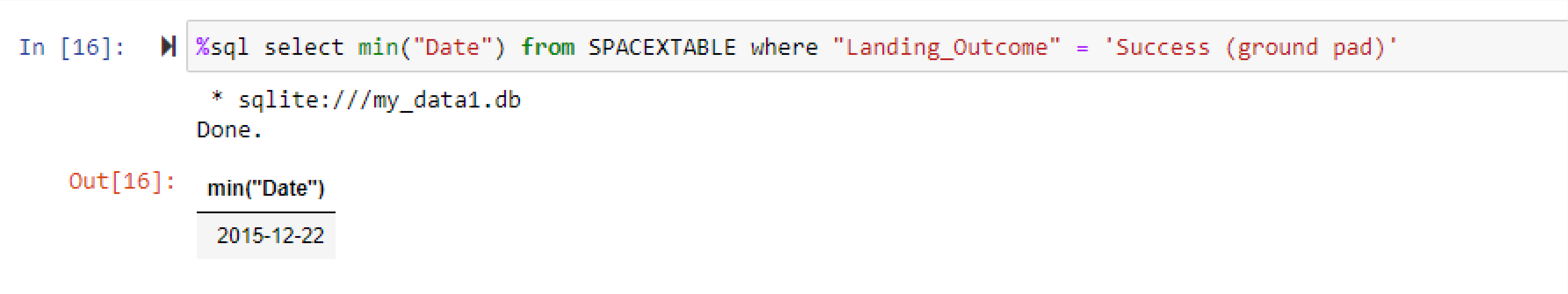
```
 * sqlite:///my_data1.db
Done.
```

Out[15]:

avg("PAYLOAD_MASS__KG_")
2928.4

- Average payload for specific booster version

First Successful Ground Landing Date



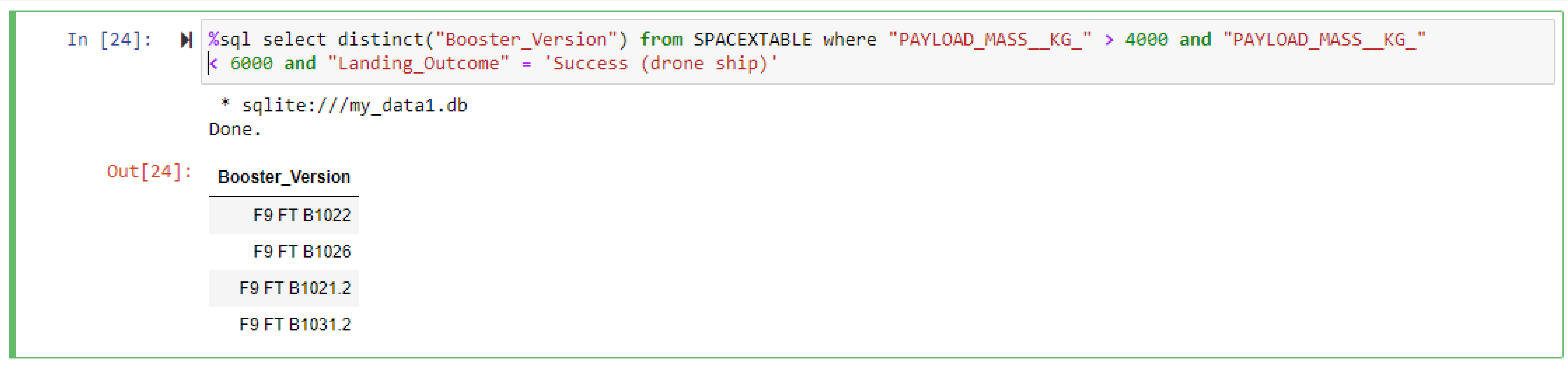
```
In [16]:  %sql select min("Date") from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)'

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

Out[16]:

min("Date")
2015-12-22

- Min function used on Date for specific landing outcome

Successful Drone Ship Landing with Payload between 4000 and 6000



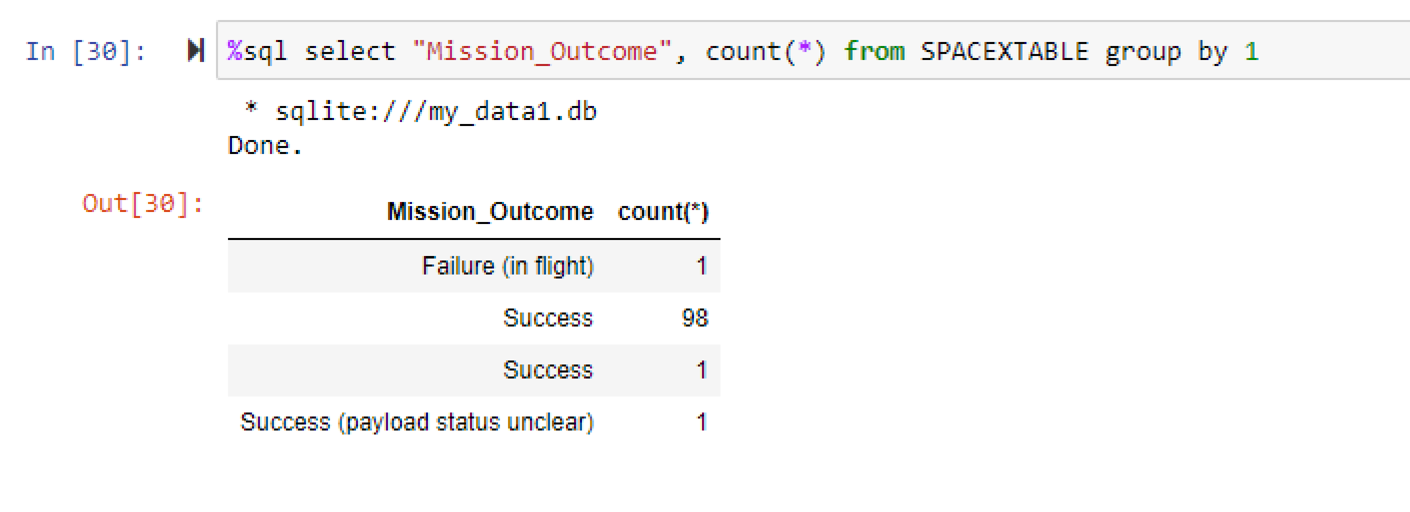
```
In [24]:  %sql select distinct("Booster_Version") from SPACEXTABLE where "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_"
          < 6000 and "Landing_Outcome" = 'Success (drone ship)'

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

Out[24]:

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

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes



```
In [30]:  %sql select "Mission_Outcome", count(*) from SPACEXTABLE group by 1
```

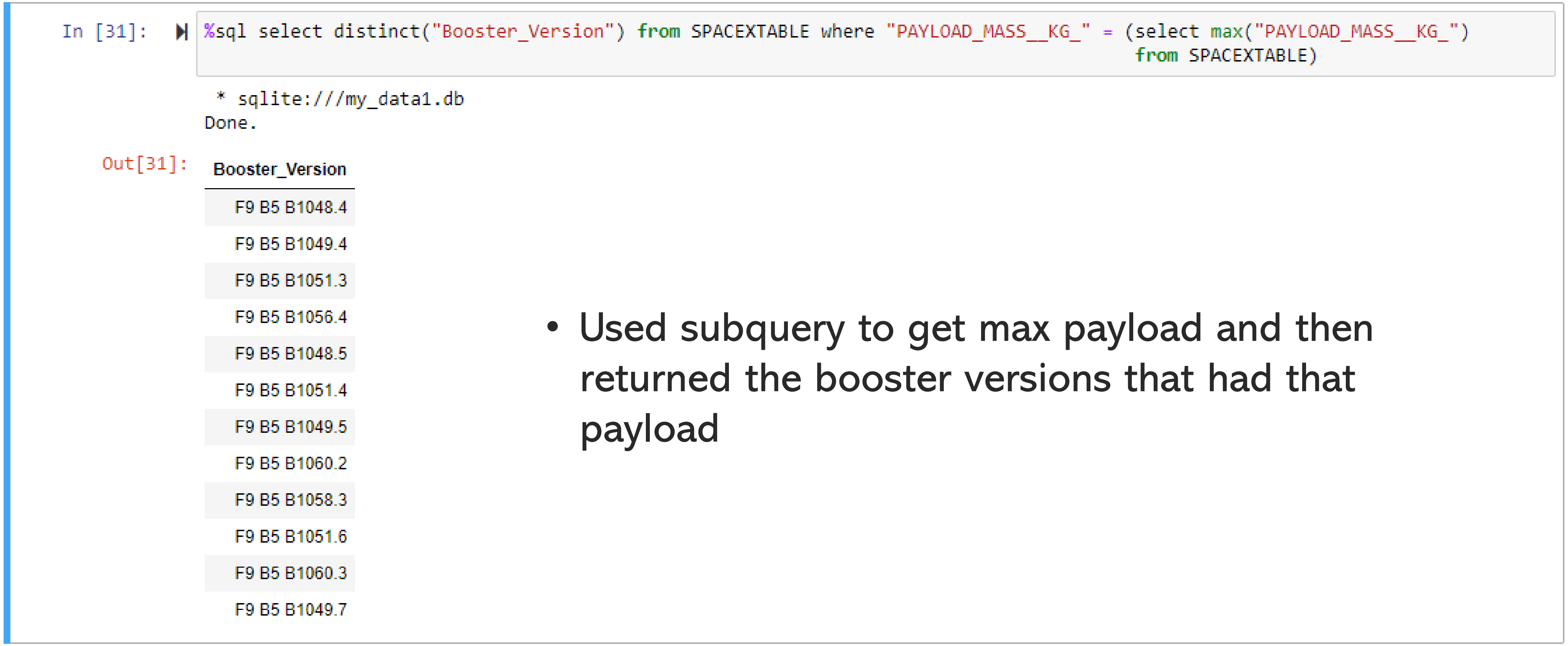
```
 * sqlite:///my_data1.db
Done.
```

Out[30]:

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

- 101 successful missions and only 1 failed mission

Boosters Carried Maximum Payload



In [31]:

```
%sql select distinct("Booster_Version") from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_")
                                                                                          from SPACEXTABLE)
```

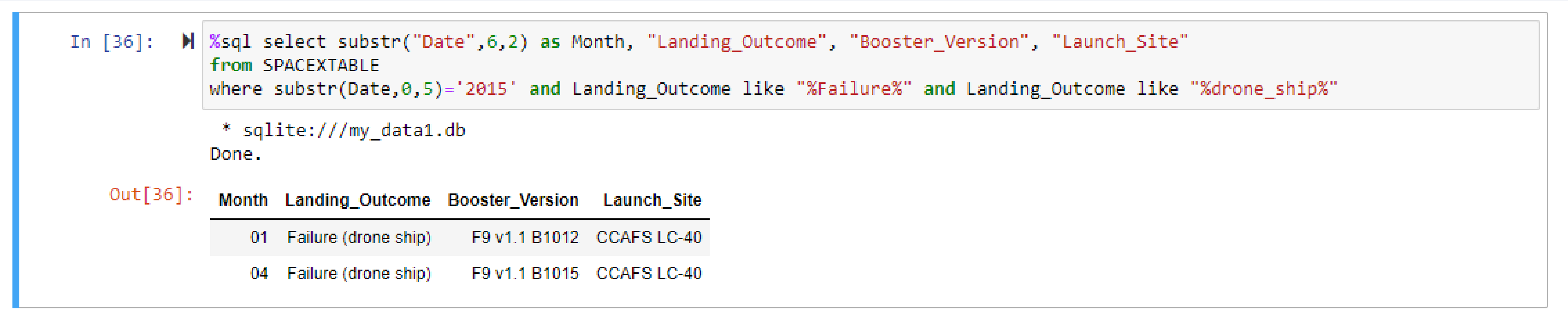
```
 * sqlite:///my_data1.db
Done.
```

Out[31]:

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

- Used subquery to get max payload and then returned the booster versions that had that payload

2015 Launch Records



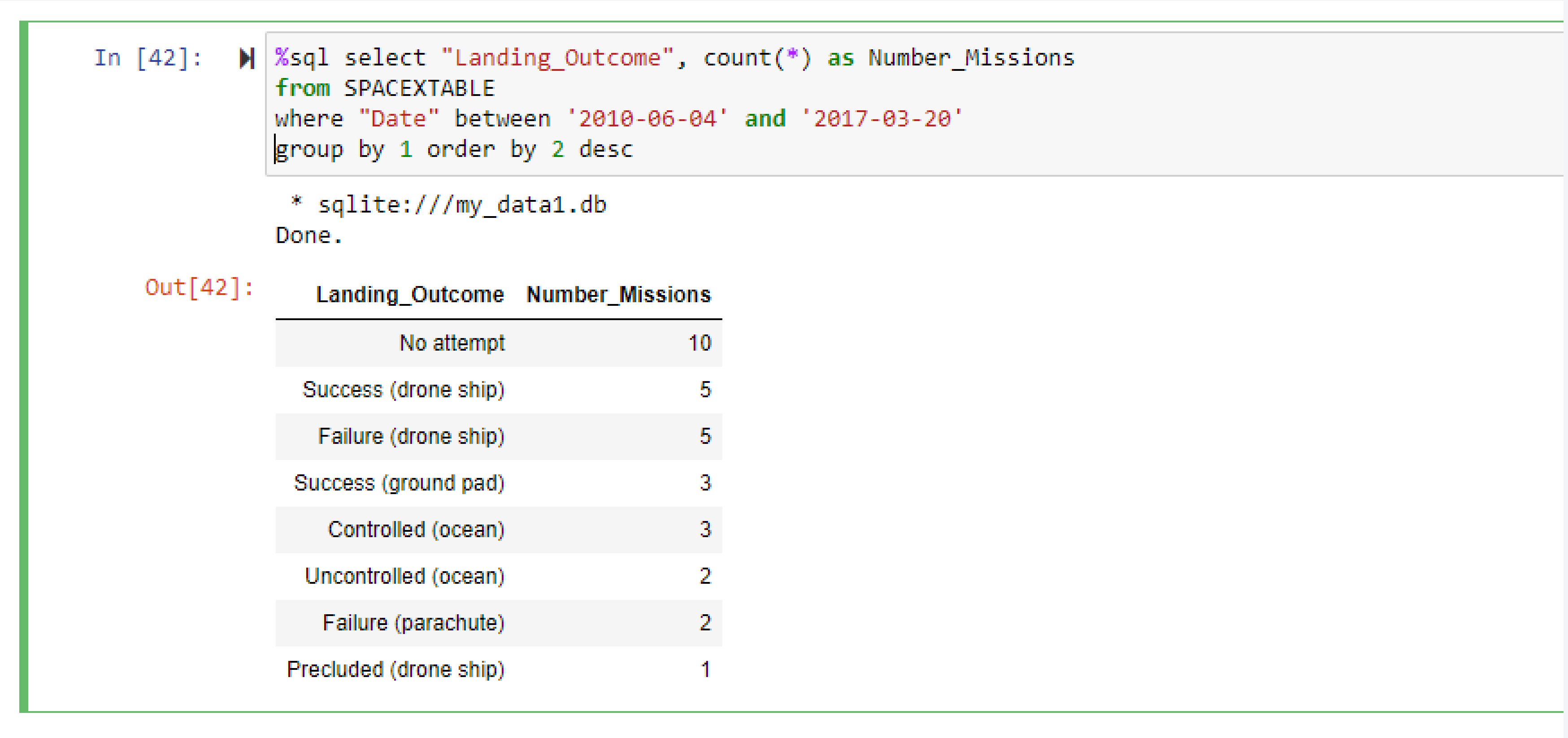
```
In [36]:  %sql select substr("Date",6,2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
          from SPACEXTABLE
          where substr(Date,0,5)='2015' and Landing_Outcome like "%Failure%" and Landing_Outcome like "%drone_ship%"

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

Out[36]:

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

- 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 [42]:

```
%sql select "Landing_Outcome", count(*) as Number_Missions
from SPACEXTABLE
where "Date" between '2010-06-04' and '2017-03-20'
group by 1 order by 2 desc
```

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

Out[42]:

Landing_Outcome	Number_Missions
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

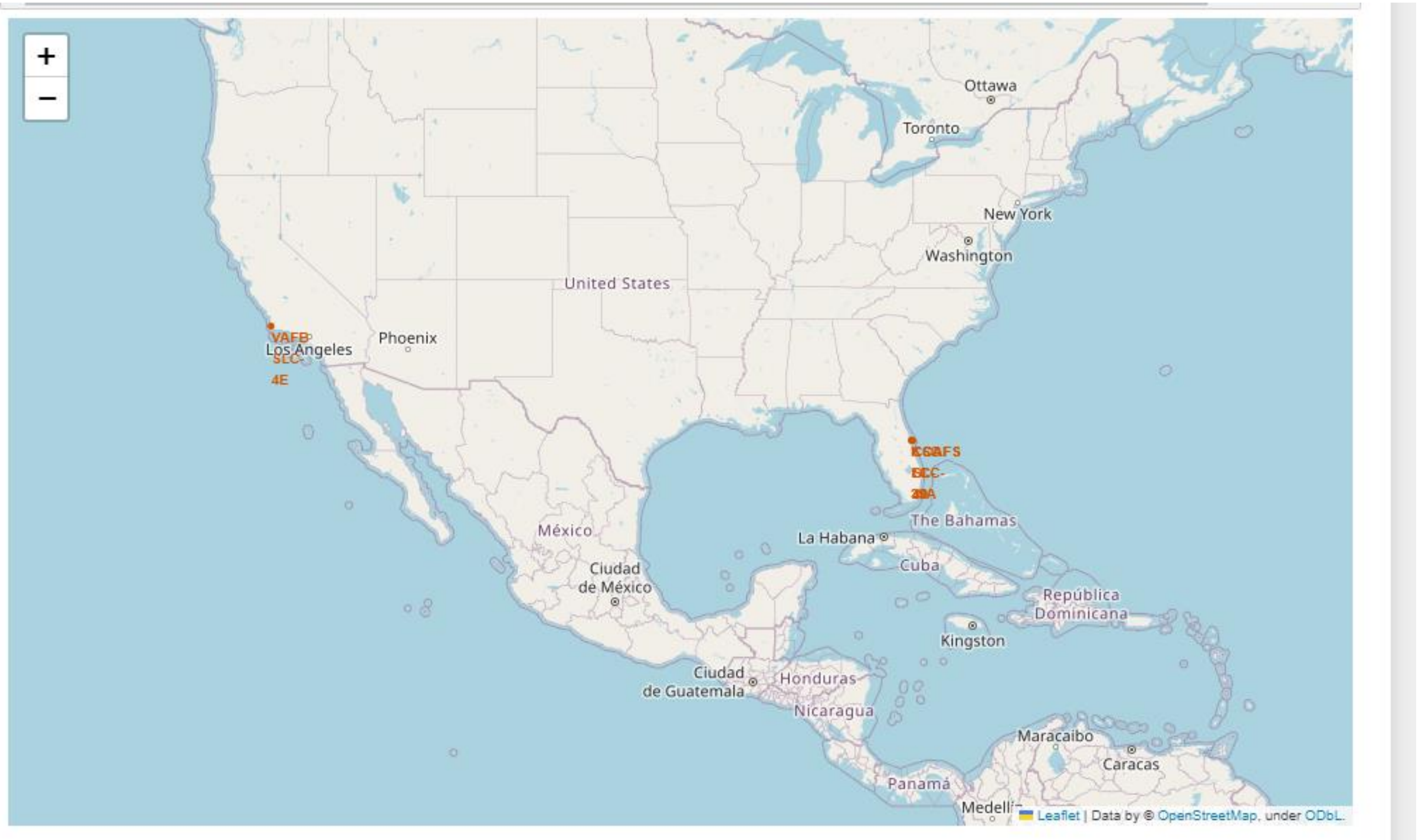
- Count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Section 3

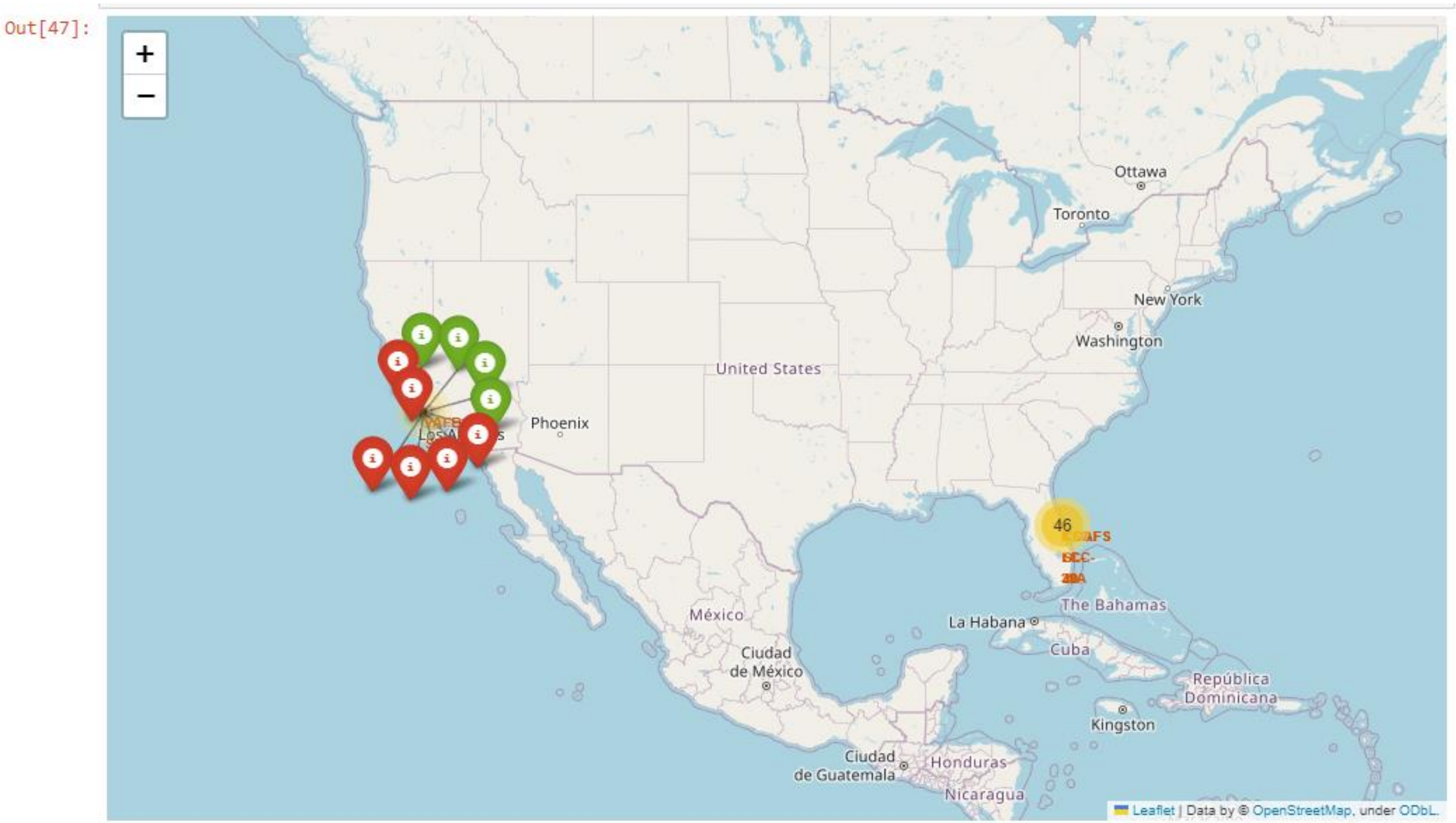
Launch Sites Proximities Analysis

SpaceX Launch Sites

Out[45]:

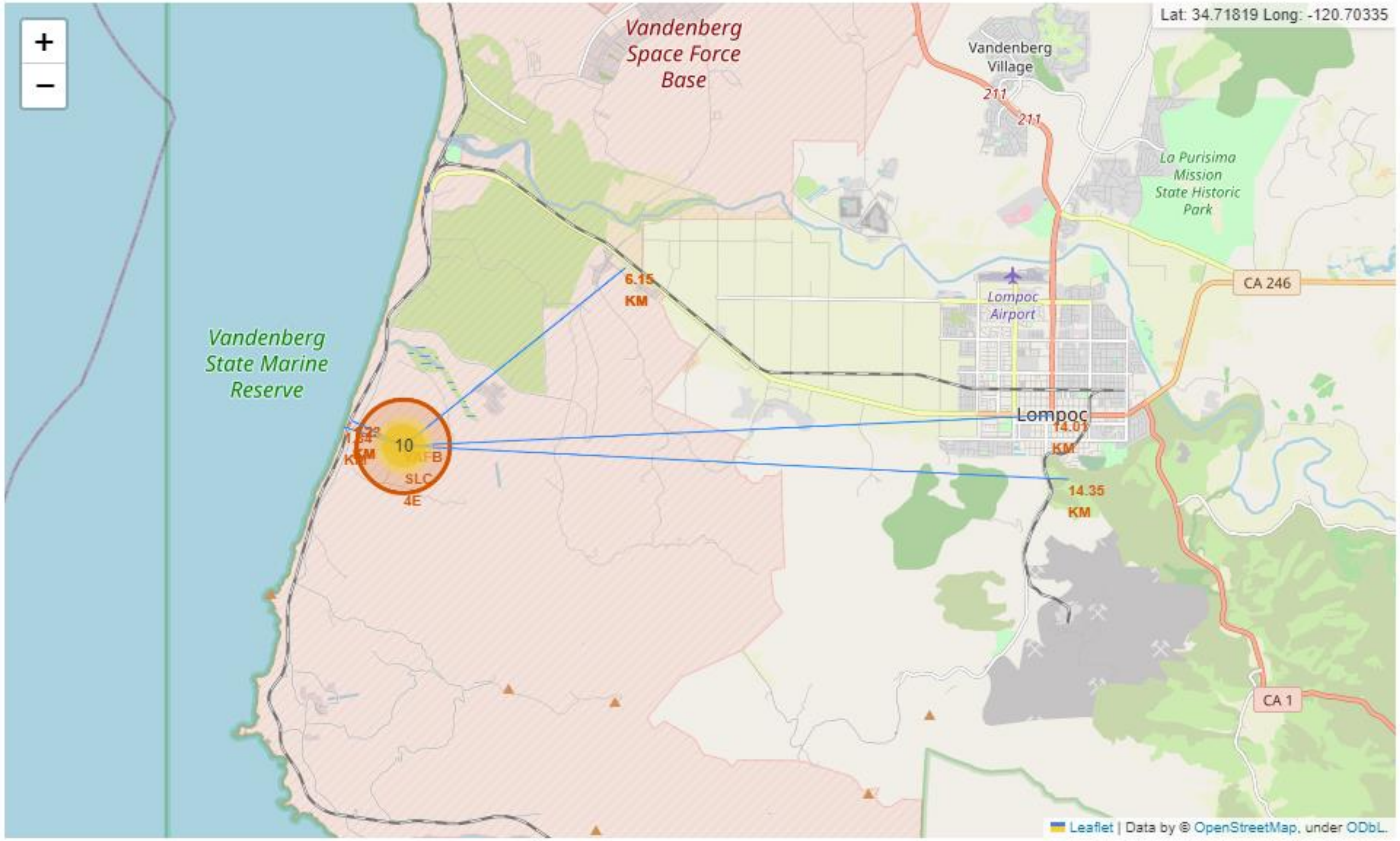


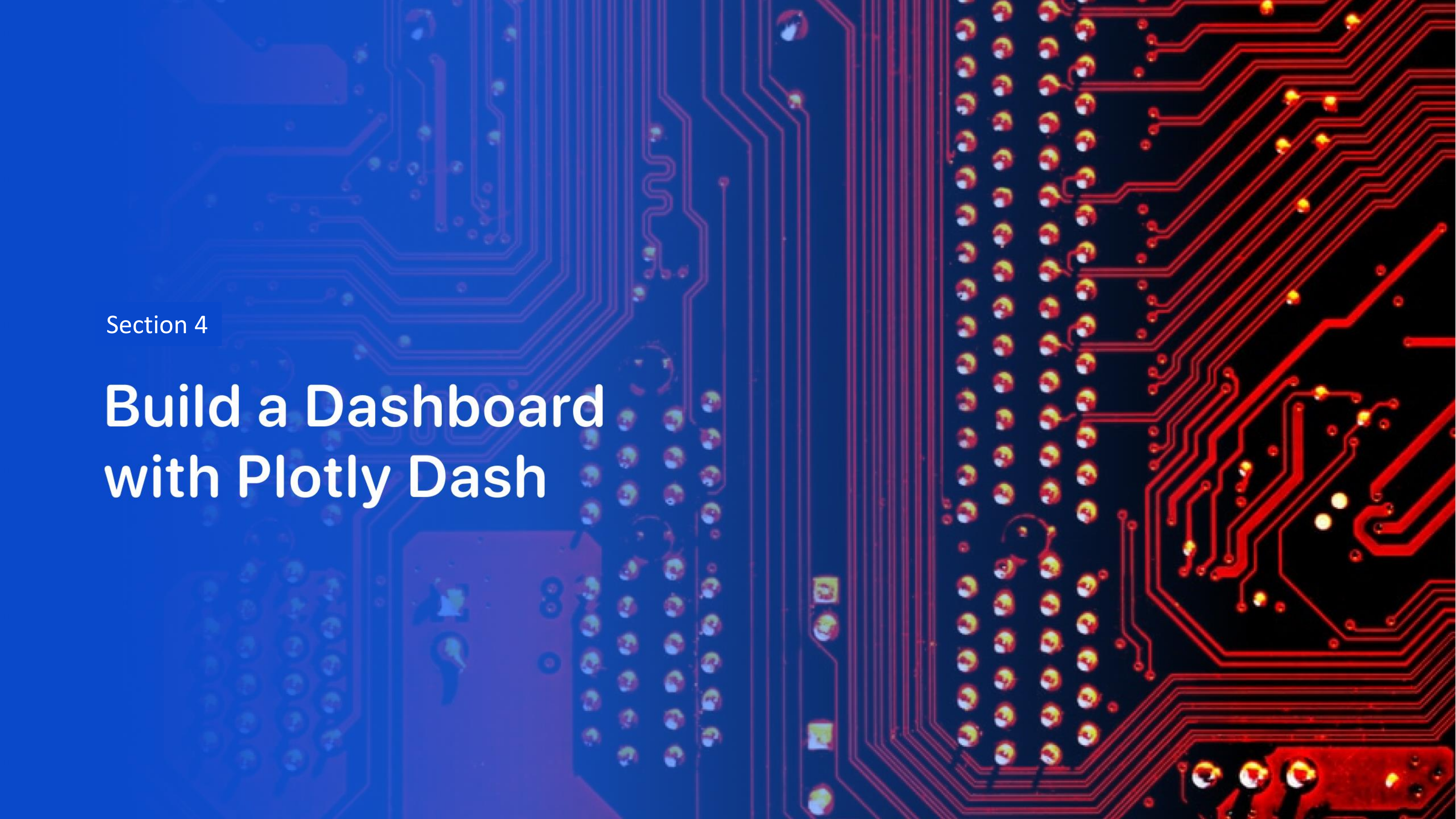
West Coast Launch Succuss/Failure Marker Cluster



West Coast Site Surrounding

Out[42]:

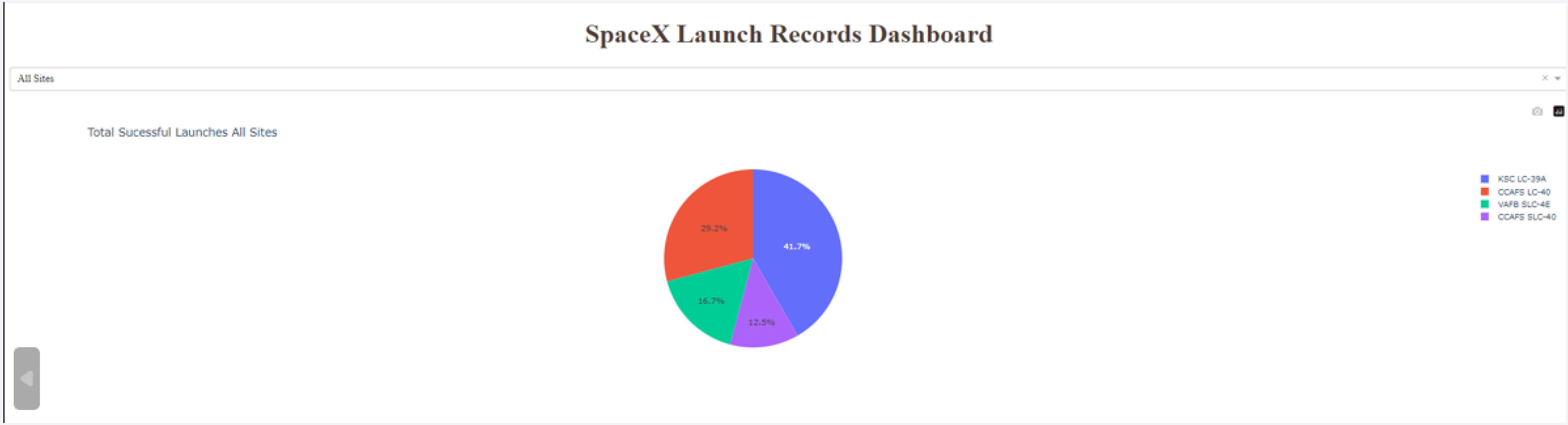

+
−
Lat: 34.71819 Long: -120.70335
Vandenberg Space Force Base
Vandenberg Village
211
211
La Purisima Mission State Historic Park
Vandenberg State Marine Reserve
6.15 KM
Lompoc Airport
CA 246
Lompoc
14.01 KM
10
SLC 4E
14.35 KM
CA 1
Leaflet | Data by © OpenStreetMap, under ODbL.



Section 4

Build a Dashboard with Plotly Dash

Landing Success by Launch Site

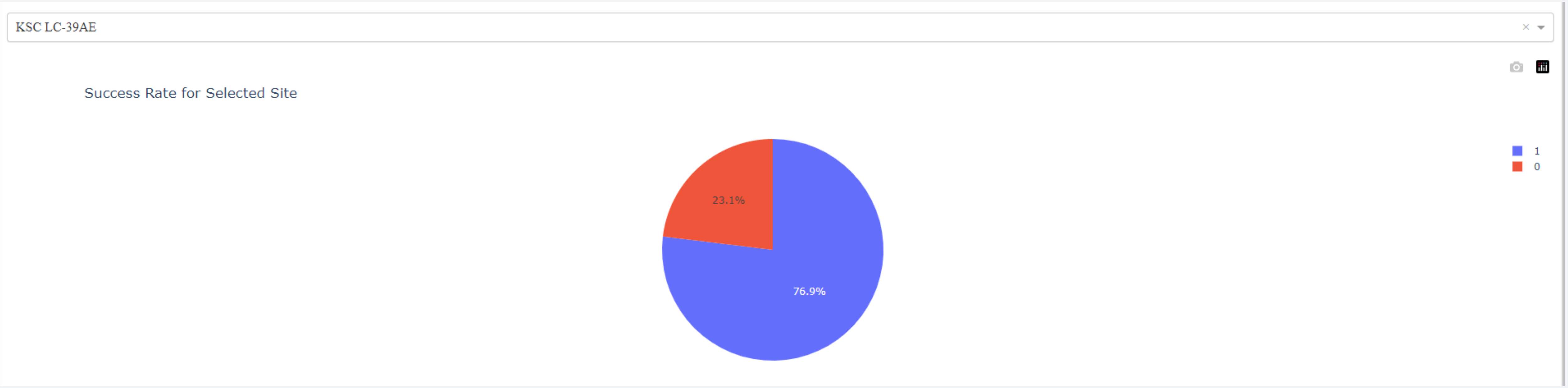


SpaceX Launch Records Dashboard

All Sites

- KSC LC 39A and CCAFS LC-40 have the highest landing sucesses

KSC LC-39AE has 77% success rate

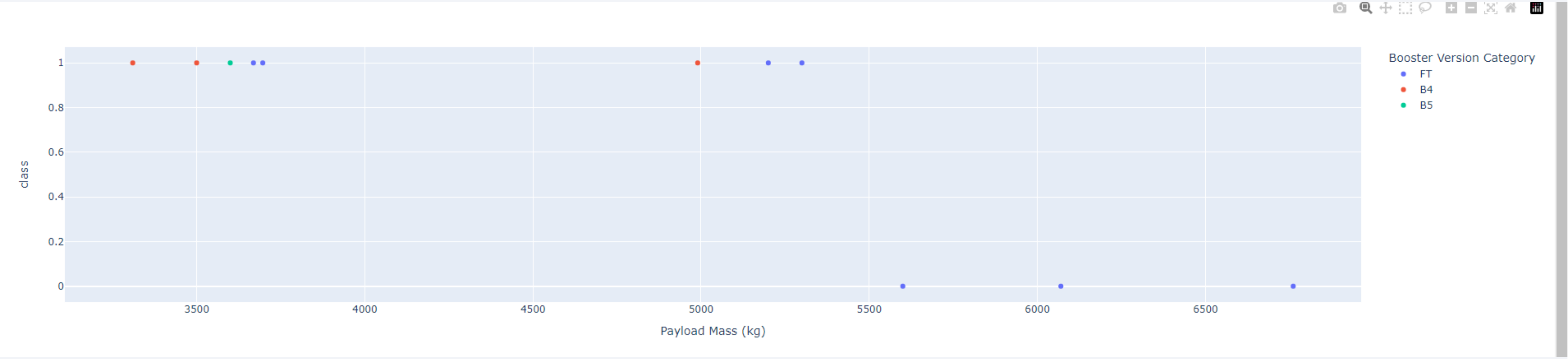


KSC LC-39AE

Success Rate for Selected Site

- KSC LC-39AE has the highest success rate at 77%

Booster Version FT only Successful Landing for Small Payloads

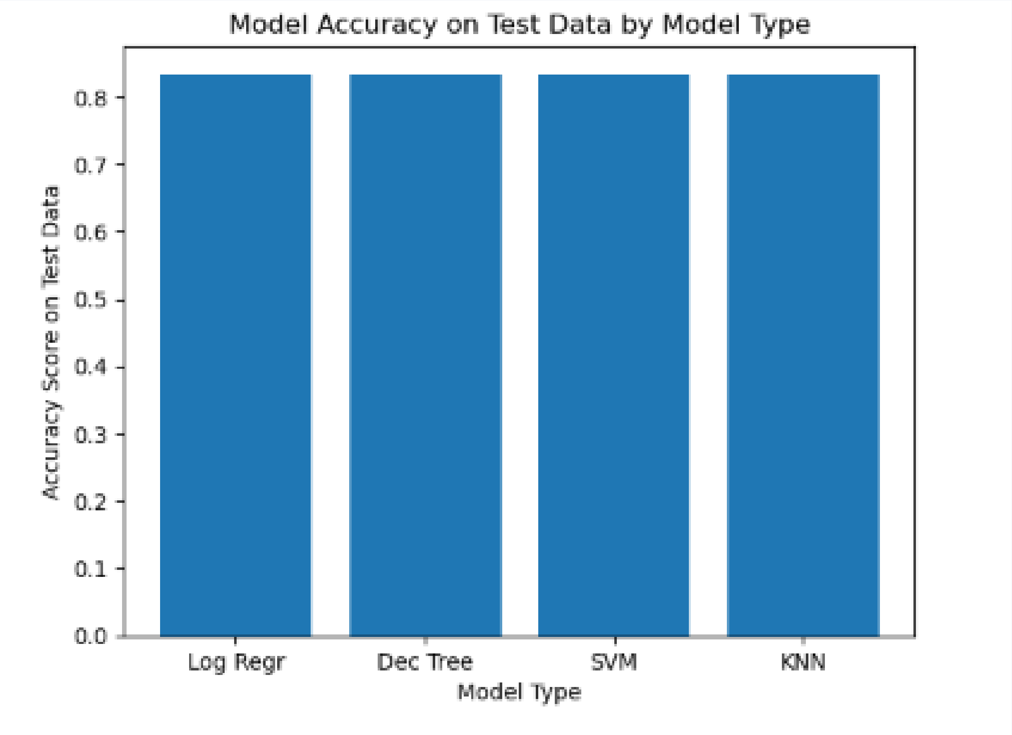


- The FT booster version with large payloads (over 5k) have a zero success rate

Section 5

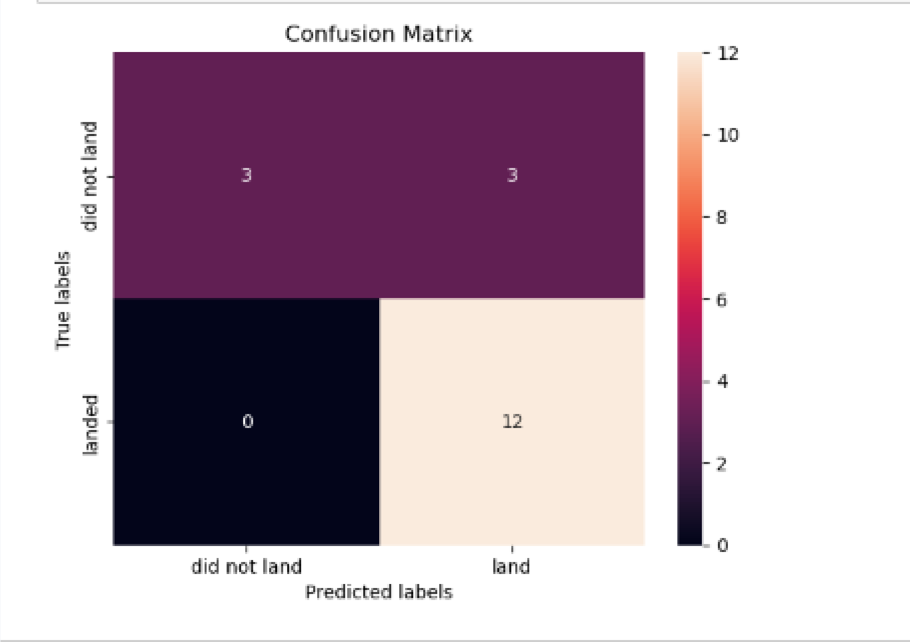
Predictive Analysis (Classification)

Classification Accuracy



- There was no discernible difference in classification accuracy of the different model types
- All performed well identifying successful landings however they were only 50% accurate on identifying failed landings
- The small size of the date set likely contributed to this

Confusion Matrix



- The best model of each model type returned the same confusion matrix
- They all predicted all to the successful landings
- They all were only 50% correct in predicting failed landings

Conclusions

- The models were able to accurately predict successful landings but had a problem with false positives only getting failed landings correct half the time
- The 83% accuracy rate suggests a reasonably good model that could be used
- Time was probably the most impactful variable suggesting that constant improvements in technology and technique drive the success rate

Thank you!