IBM Developer
SKILLS NETWORK

WINNING THE SPACE RACE WITH DATA SCIENCE

Dirk Hartog
March 10, 2023

Outline

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

Executive Summary

- **Summary of methodologies**

- **Summary of all results**

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.

Our primary purpose of this data analysis will be to
predict if the Falcon 9 first stage will land successfully.

SECTION 1:

METHODOLOGY

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using a request to the SpaceX API
  - Cleaned the requested data
- Perform data wrangling
  - Converted the outcomes of a booster landing into training labels for
- 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

- Data was collected by using a request call to SpaceX API
- Data was also collected through web scraping to extract Falcon 9 launch records HTML table from Wikipedia
    - The html was parsed table and converted into a Pandas dataframe

Data Collection – SpaceX API

Step 1

Imported relevant libraries:

- requests
- pandas
- numpy
- datetime

Step 2

Defined a series of helper functions that use the API to extract information using identification numbers in the launch data.

Step 3

Did an initial clean up of the data that included taking a subset of the data

- Launchpad
- Rocket
- Payload
- Cores
- Flight number
- Date_utc

Step 4

The helper functions were ran which extracted relevant information from our initial data set to create new columns for later use in our analysis and model building

Step 5

We wrangled the data filtering only the data from Falcon - 9 rockets

We dealt with missing values in Payload Mass by replacing them with the average Payload Mass

Reference notebook:

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/e5ee47d4d8bab4c737b1a488a92903ee1d3010b0/Data%20collection%20API.ipynb

Data Collection - Scraping

Step 1

Installed and imported relevant packages and libraries :

- sys
- requests
- BeautifulSoup
- re
- unicodedata
- pandas

Step 2

Used the request.get() method on the static url and created a BeautifulSoup object from the response.

Used the soup.find_all() function to find the relevant table

Step 3

We extracted the column name from the header of the wikipedia table and appended that to a list called column names.

This list was used to create an empty dictionary with the column names as keys and an empty list as values to be filled.

Step 4

The HTML tables were parsed using for loops and conditional statements to extract the information and append launch dictionary.

Step 5

Finally, the launch dictionary was converted to a data frame for exploratory data analysis

Reference notebook:

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/e5ee47d4d8b

Data Wrangling

Step 1

Imported the necessary libraries to perform initial data analysis and to determine the training labels we will use in our predictions models.

Loaded our dataframe using pd.read_csv()

Step 2

Performed an initial exploratory data analysis.

- Identified and calculated the percentage of the missing values in each attribute
- Identified which columns are numerical and categorical
- Calculated the number of launches on each site
- Created a landing outcome label from Outcome column.

Step 3

The landing outcome label will represent the classification variable that represents the outcome of each launch.

If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully.

Reference notebook:
https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/e5ee47d4d8bab4c737b1a488a92903ee1d3010b0/Data%20Wrangling.ipynb

EDA with Data Visualization

- Summary of charts
  - The Seaborn library was used for visualizations
    - Catplots were created and each point on the plot was classified by it’s class (launch success or failure)
      - Flight number vs. Payload Mass
      - Launch site vs. Flight number
      - Launch site vs. Payload Mass
      - Flight number vs. Orbit
      - Orbit vs. Payload Mass
    - A Barplot was used to visualize the average success rate based on Orbit
    - A Lineplot was created to look at the average success rate from 2010 to 2020

Reference notebook

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/e5ee47d4d8bab4c737b1a488a92903ee1d3010b0/EDA_dataviz.ipynb

EDA with SQL

- SQL Queries
  - Identified the unique launch sites
  - Displayed 5 records where launch site started with ‘CCA’
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - The date of the first successful landing outcome in ground pad was achieved.
  - The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - The total number of successful and failure mission outcomes
  - The names of the booster versions which have carried the maximum payload mass.
  - Display 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.

Reference notebook

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/e5ee47d4d8bab4c737b1a488a92903ee1d3010b0/EDA_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Objects were added to quickly identify the locations of launch sites, success/failures of each launch based on the site and proximity to cities, railways, and highways.
  - Marked Nasa Johnson Space Center and all launch sites using folium.Circle, folium.Popup and folium.Marker.
  - Marked success and failures at each site by creating a MarkerCluster for each launch site.
  - Marked the distance of the closest city, railway, highway, and coastline for each launch site.

Reference notebook

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/36a69e53f690911cdeaf5137fd6205913a9bba92/Lunch_Site_Location.ipynb

Build a Dashboard with Plotly Dash

- A dashboard was created to visualize percentage of successful and failed launches based on launch site and Payload mass.
- Features of the dashboard include…
  - A drop down menu to choose each launch site separately or info for all sites
  - An interactive slider to change what points were visualized in the scatter plot of Payload Mass vs. Class (success/failure)
    - Each point was also classified based on booster version
- We wanted to have a clear understanding of the success/failure rates at each site and the parameters that influenced the outcomes (payload mass, F9 booster version)

Reference code

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/de761cfa9bf1cde9e48dfbccb09ab94174881874/spacex_dash_app.py

Predictive Analysis (Classification)

- We built a model to predict the outcome of the first stage of the launch.
  - Attempted to find the best Hyperparameter for SVM, Classification Trees and Logistic Regression

Step 1

The data was standardized and a numpy array was created using the class column.

The function train_test_split to split the data X and Y into training and test data

Step 2

A GridSearchCV and a logistic regression object was created to fit the training data.

An accuracy score was calculated on the testing data

A confusion matrix was generated to define the performance of a classification algorithm

Step 3

The previous step was carried out to determine the accuracy of the test data using methods…
Support Vector Machine
Decision Classification tree
KNN classification

Practically all these algorithms give the same result

Reference notebook

https://github.com/D-hartog/IBM-Applied-Data-Sci-Capstone/blob/de761cfa9bf1cde9e48dfbccb09ab94174881874/Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

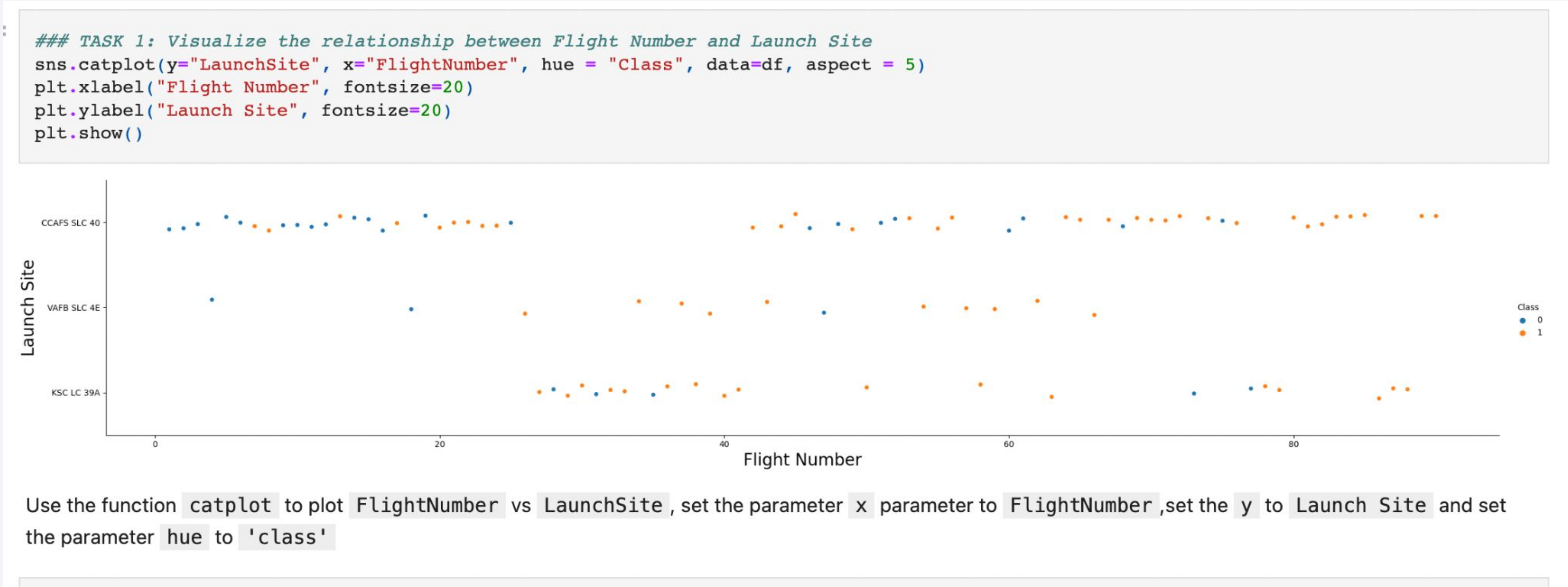
Results

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

SECTION 2:

INSIGHTS FROM EXPLORATORY DATA ANALYSIS

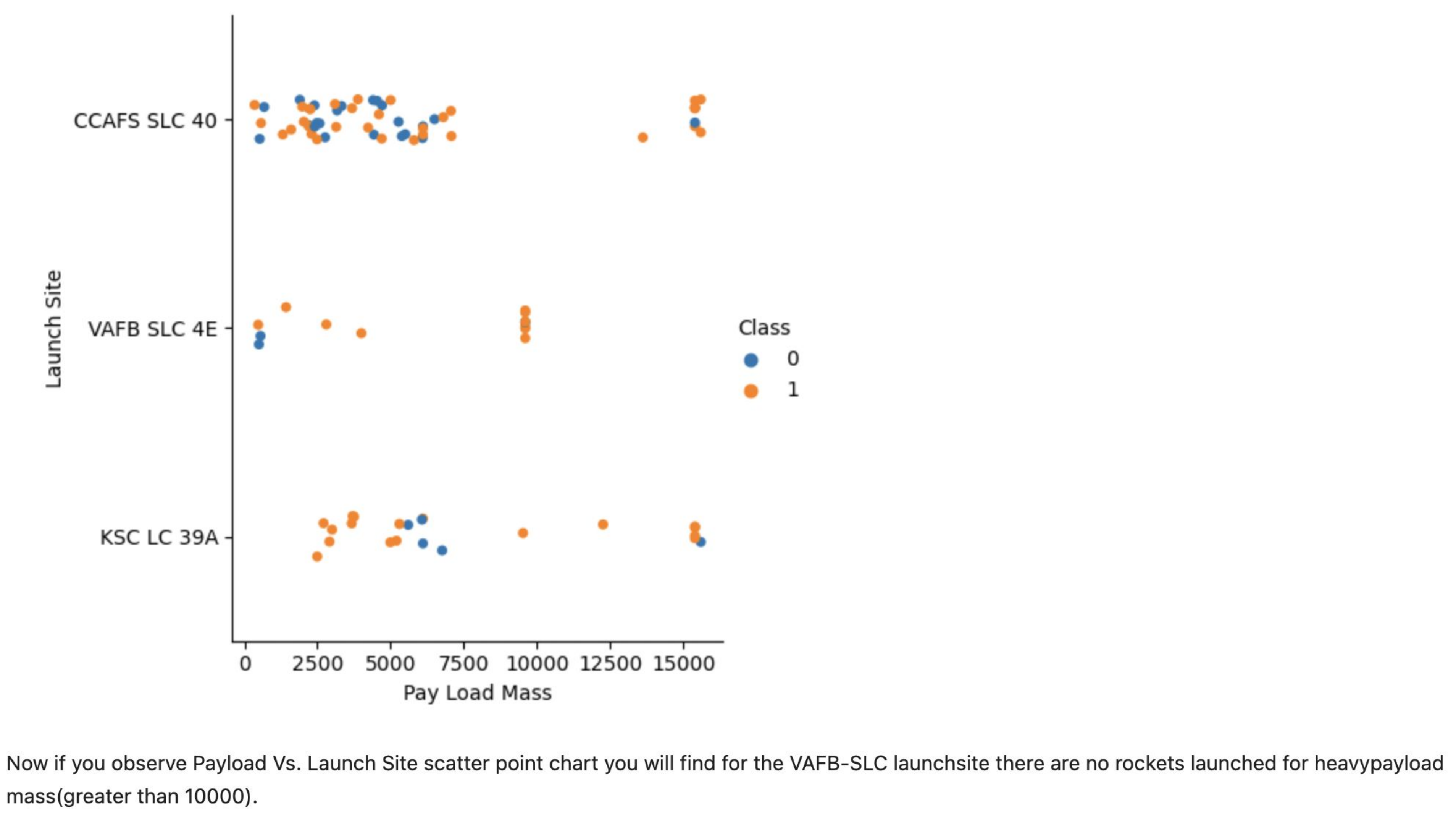
Flight Number vs. Launch Site



```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue = "Class", data=df, aspect = 5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```

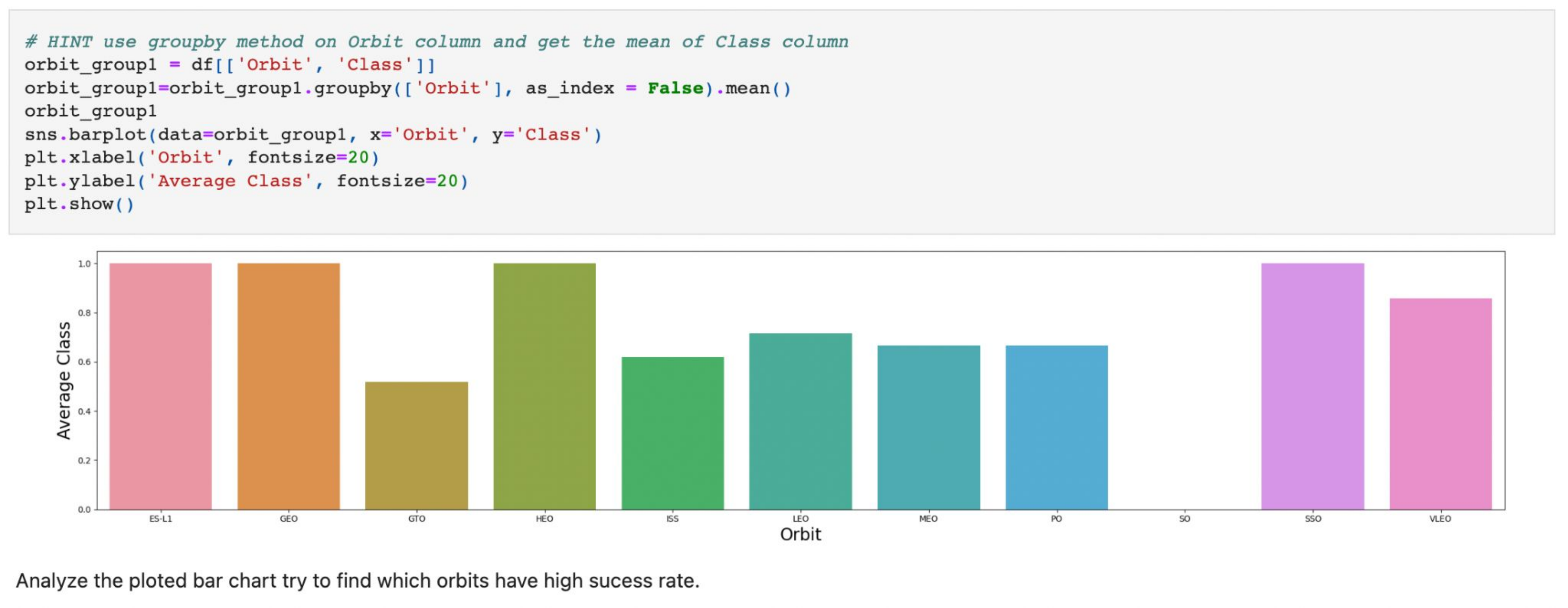
Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

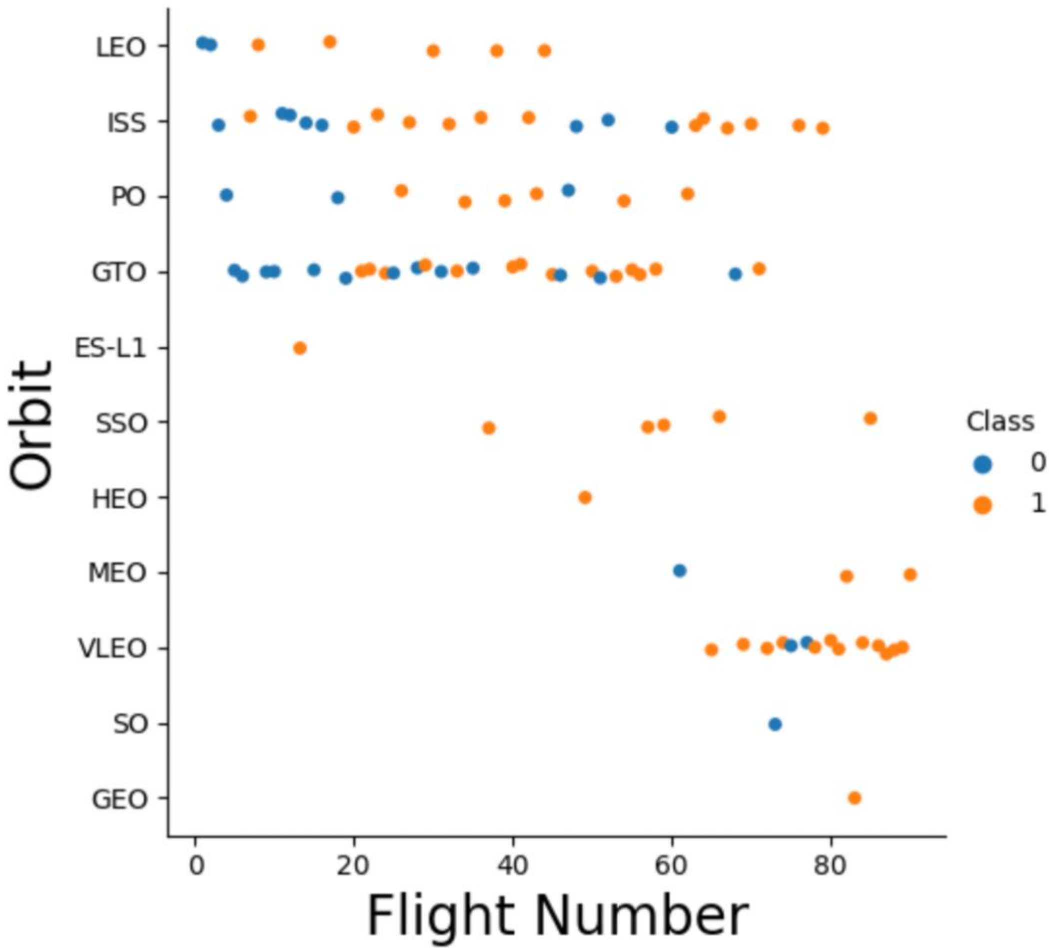
Success Rate vs. Orbit Type



```python
# HINT use groupby method on Orbit column and get the mean of Class column
orbit_group1 = df[['Orbit', 'Class']]
orbit_group1=orbit_group1.groupby(['Orbit'], as_index = False).mean()
orbit_group1
sns.barplot(data=orbit_group1, x='Orbit', y='Class')
plt.xlabel('Orbit', fontsize=20)
plt.ylabel('Average Class', fontsize=20)
plt.show()
```

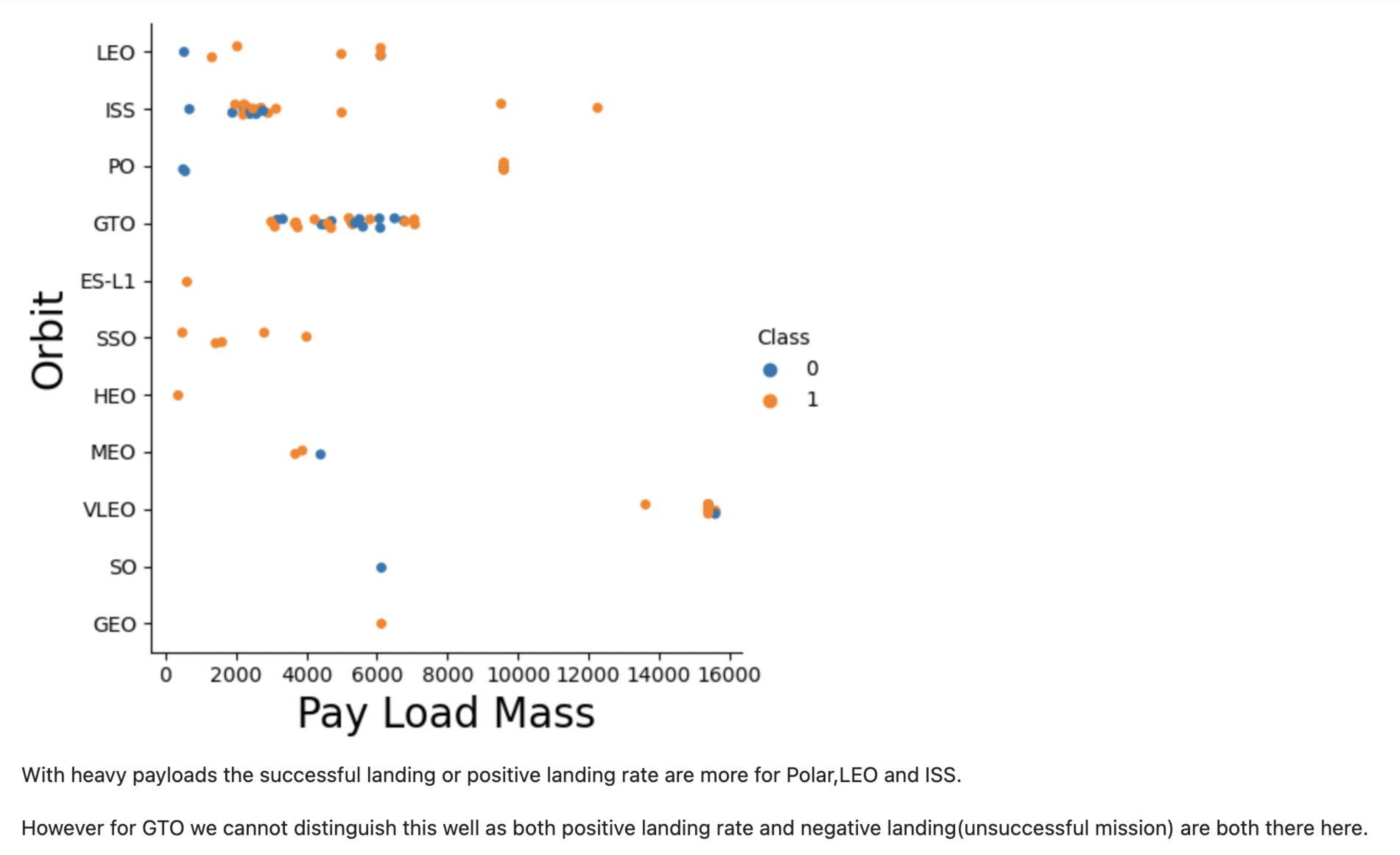
Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

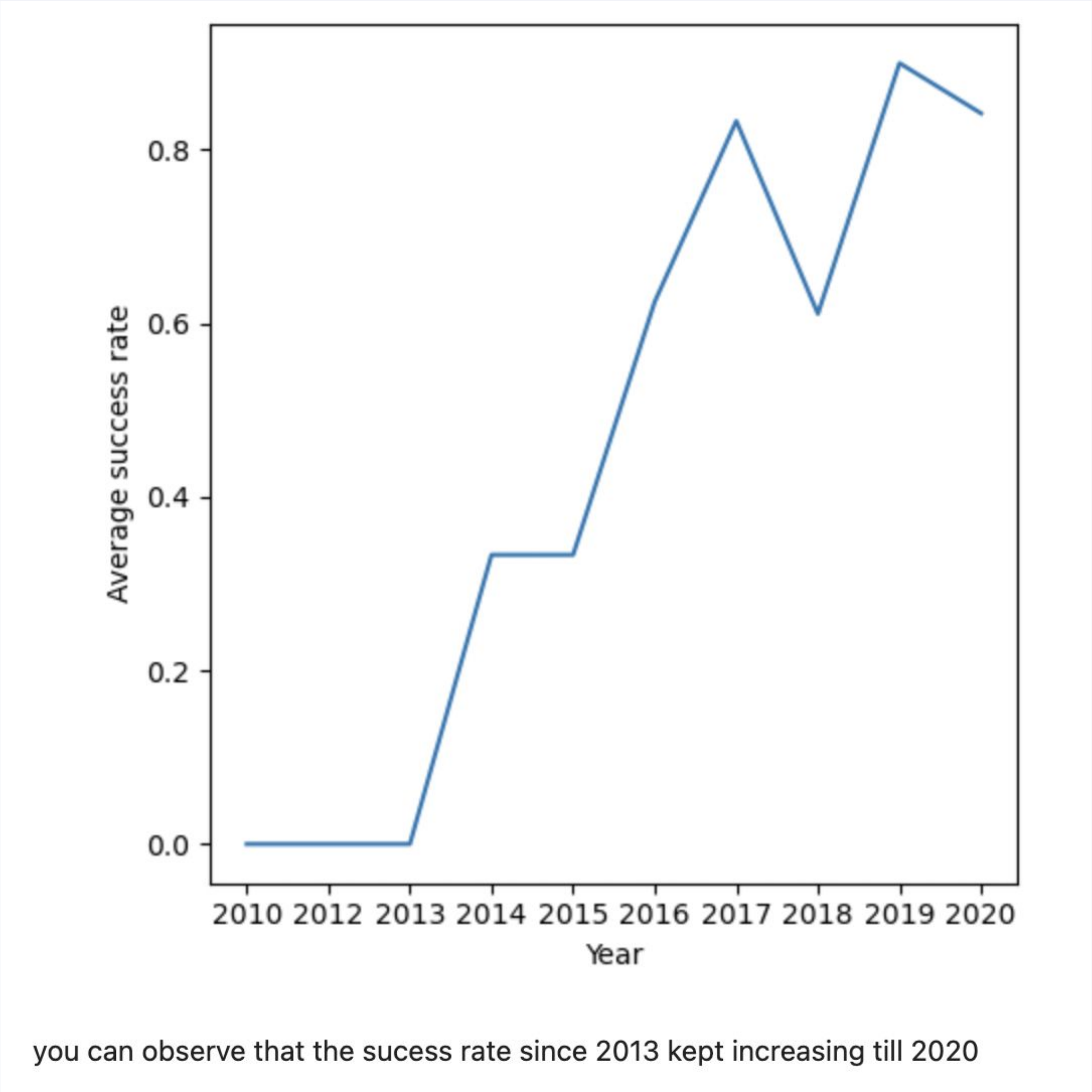
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

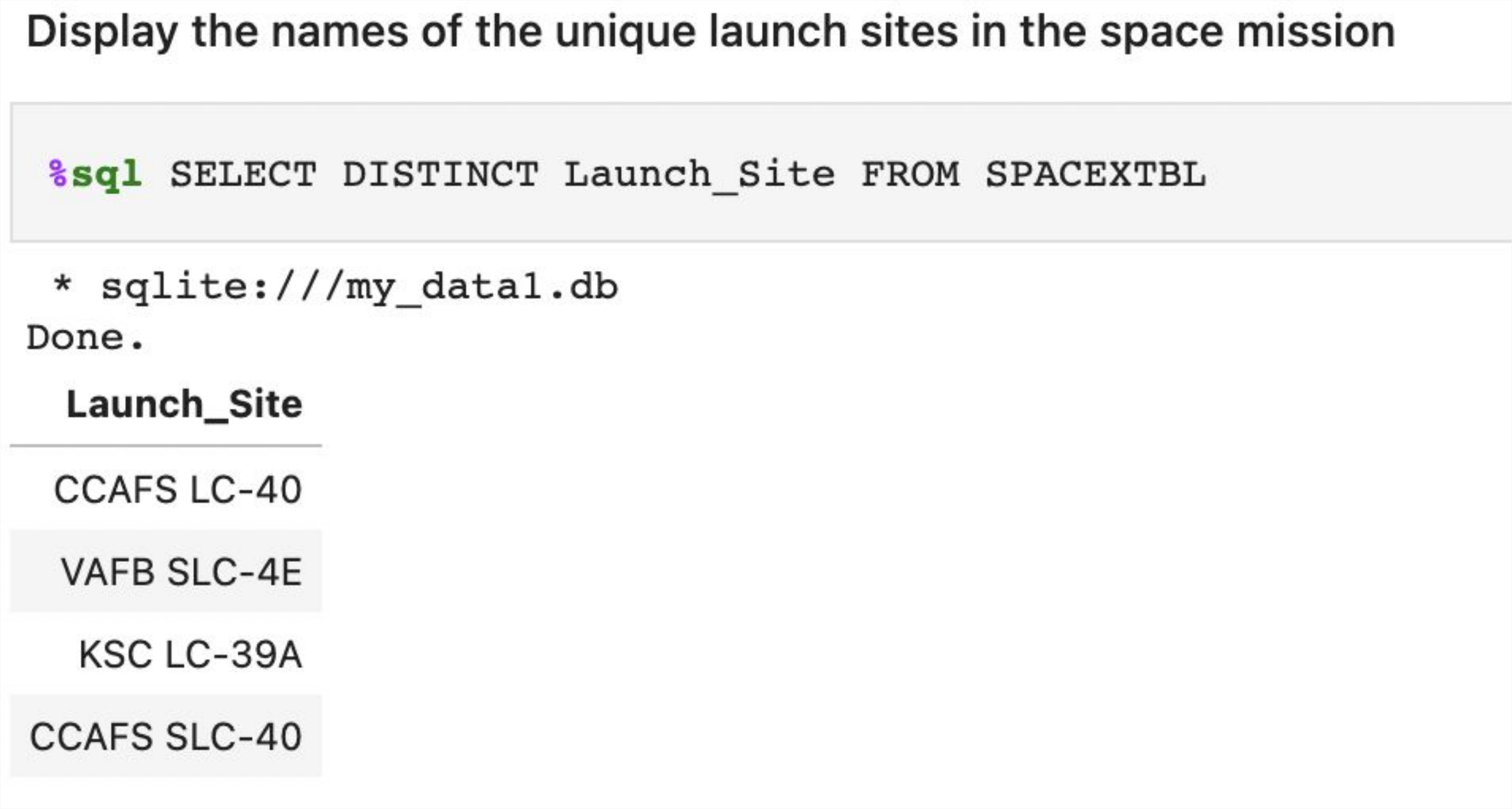
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names



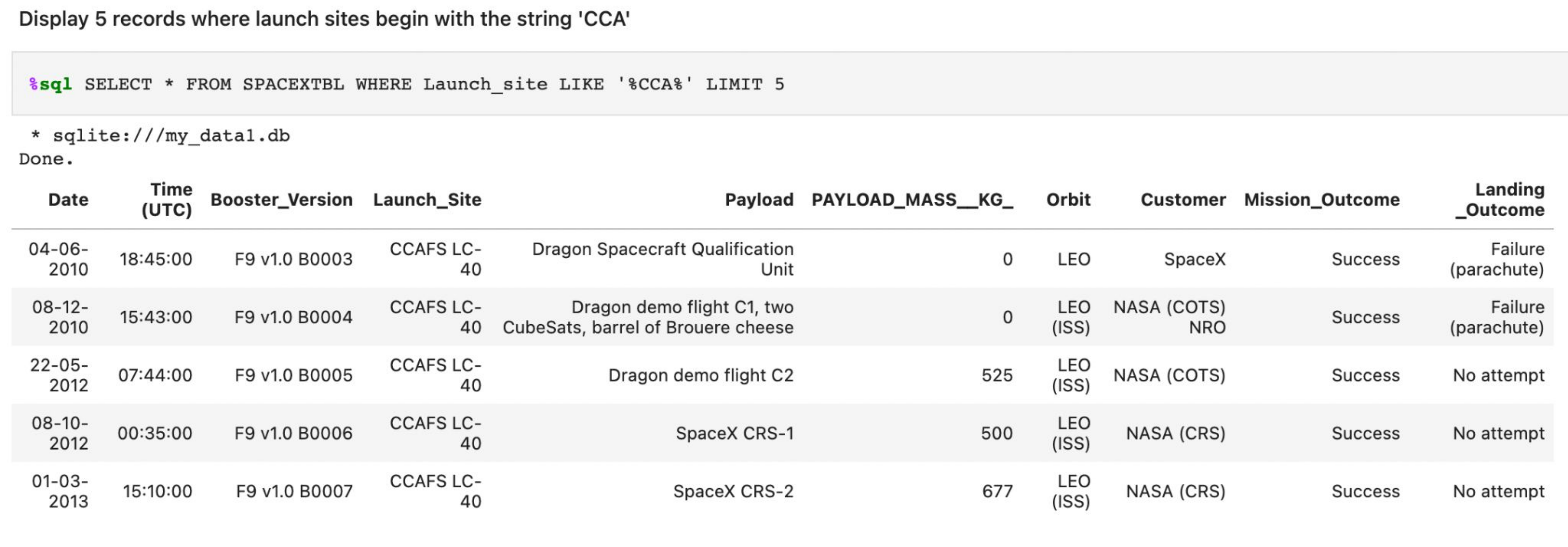
Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

```
 * 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'



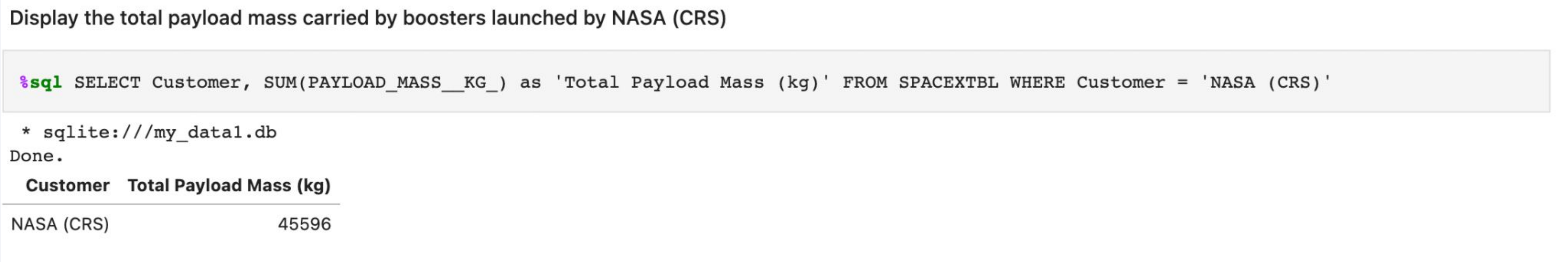
Display 5 records where launch sites begin with the string 'CCA'

```
%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
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



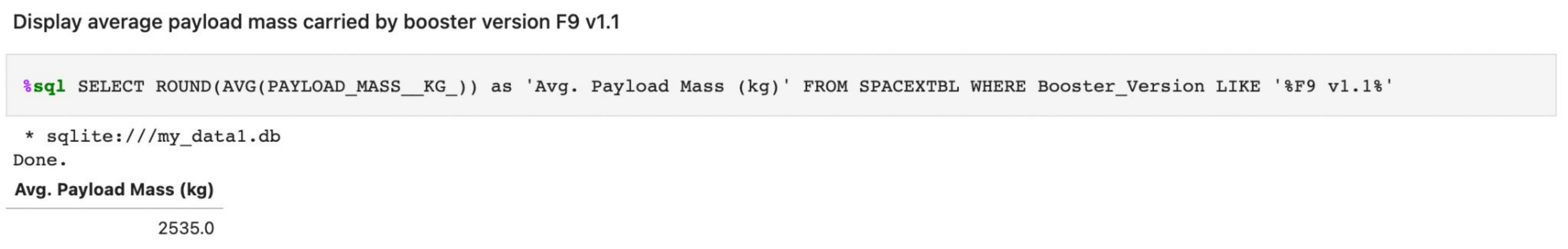
Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT Customer, SUM(PAYLOAD_MASS__KG_) as 'Total Payload Mass (kg)' FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

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

Customer	Total Payload Mass (kg)
NASA (CRS)	45596

Average Payload Mass by F9 v1.1



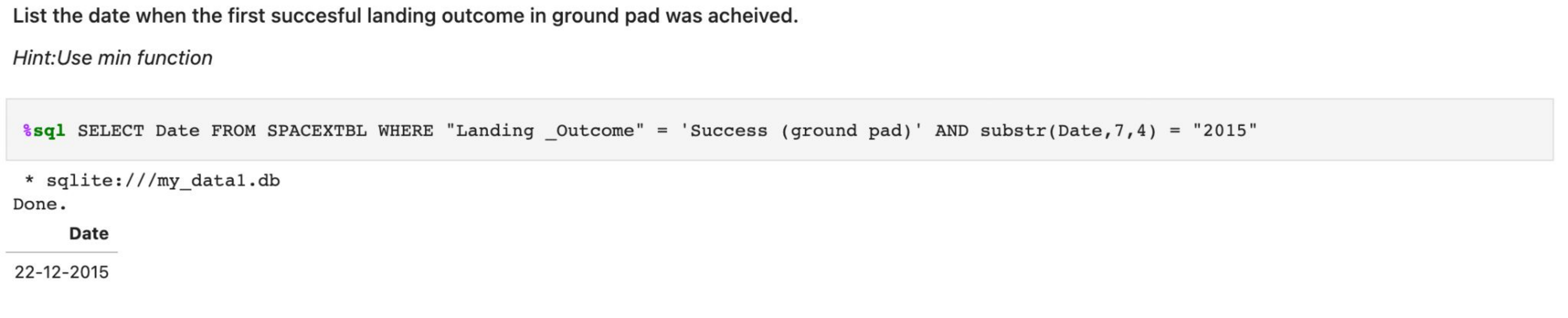
Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT ROUND(AVG(PAYLOAD_MASS__KG_)) as 'Avg. Payload Mass (kg)' FROM SPACEXTBL WHERE Booster_Version LIKE '%F9 v1.1%'
```

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

Avg. Payload Mass (kg)
2535.0

First Successful Ground Landing Date



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

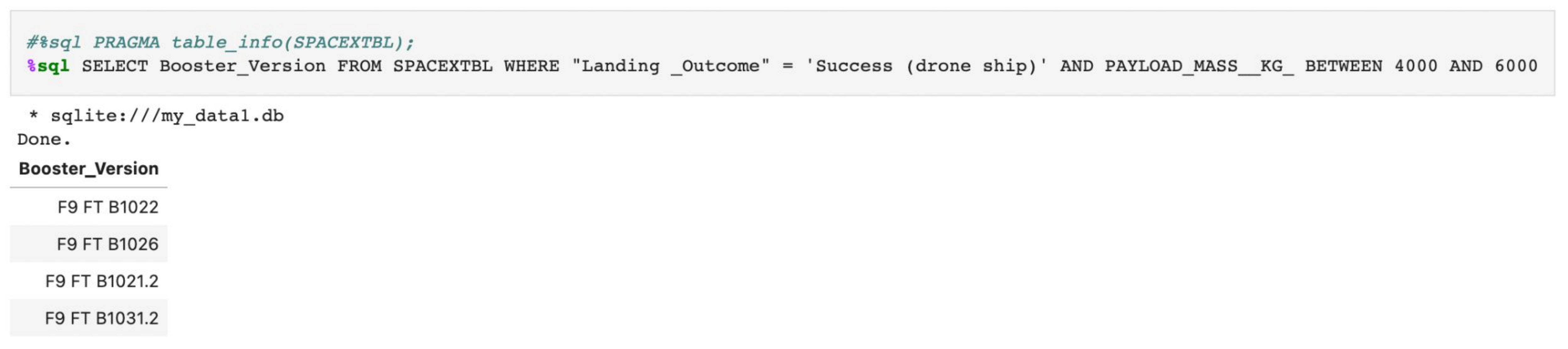
Hint:Use min function

```
%sql SELECT Date FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)' AND substr(Date,7,4) = "2015"
```

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

Date
22-12-2015

Successful Drone Ship Landing with Payload between 4000 and 6000

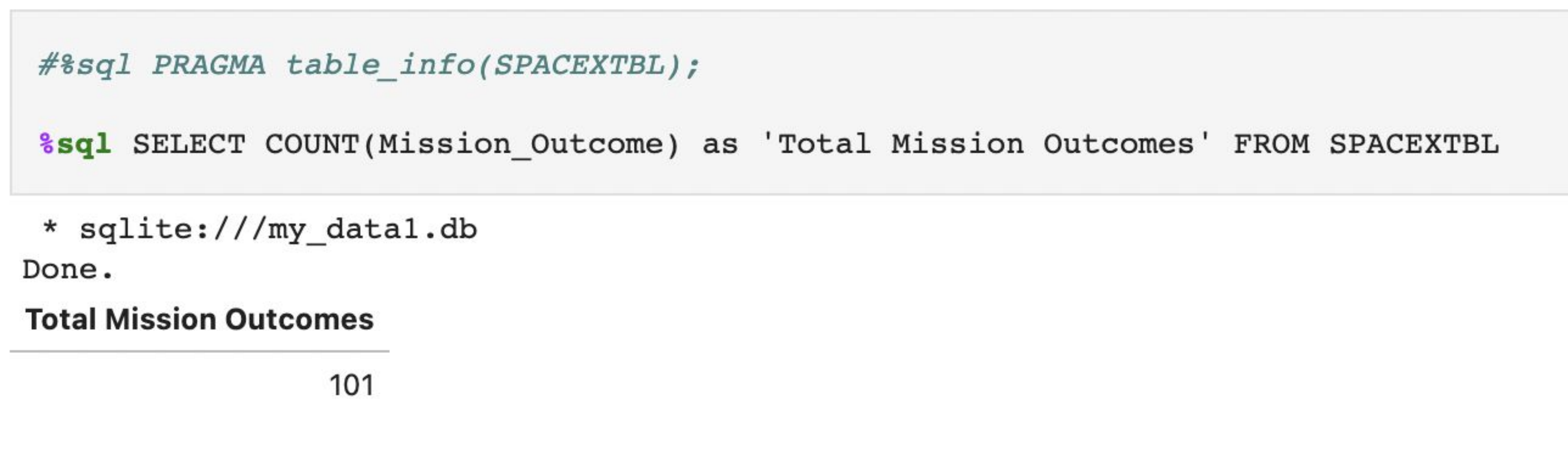


```
#%sql PRAGMA table_info(SPACEXTBL);
%sql SELECT Booster_Version FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes



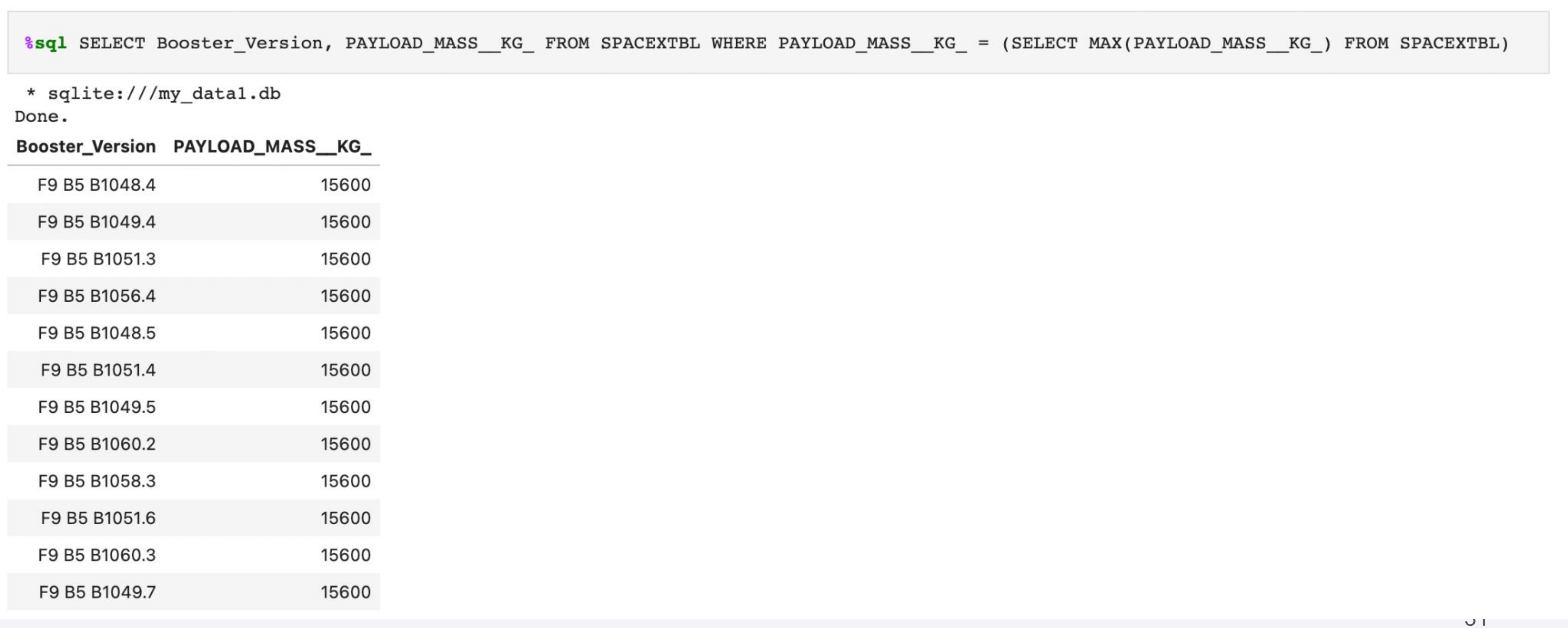
```
#%sql PRAGMA table_info(SPACEXTBL);

%sql SELECT COUNT(Mission_Outcome) as 'Total Mission Outcomes' FROM SPACEXTBL
```

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

Total Mission Outcomes
101

Boosters Carried Maximum Payload

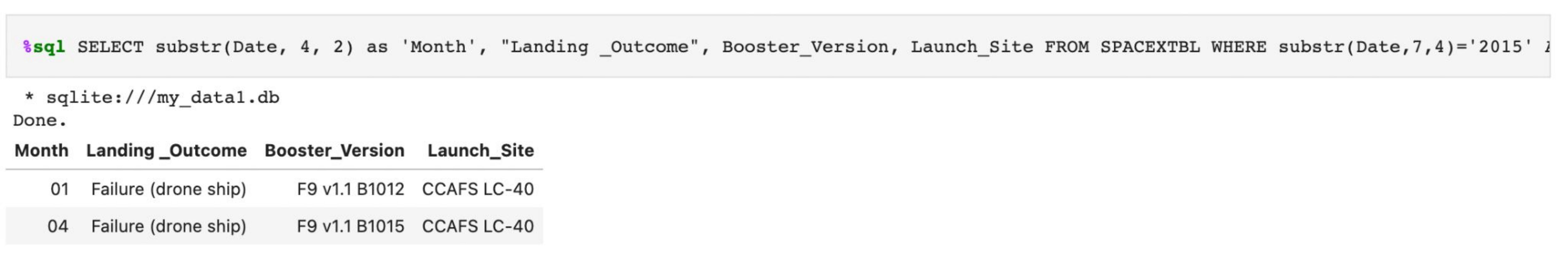


```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

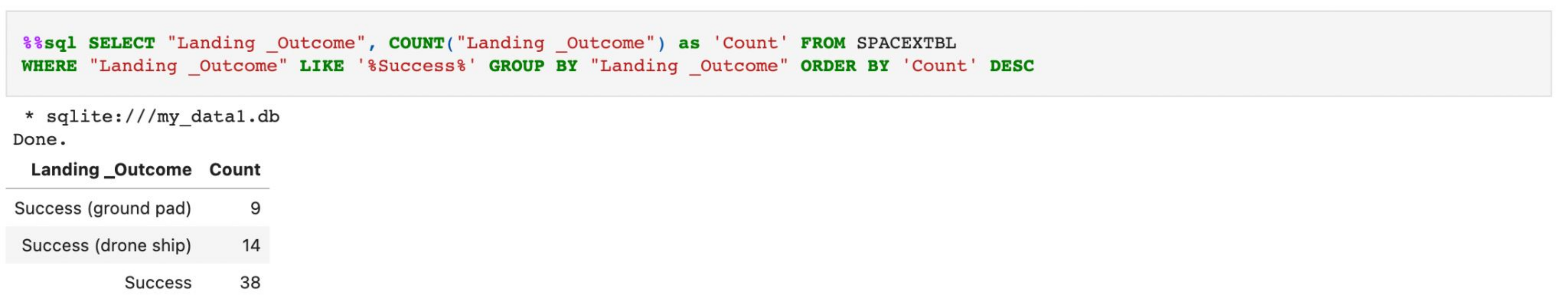


```
%sql SELECT substr(Date, 4, 2) as 'Month', "Landing _Outcome", Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015'
```

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

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

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



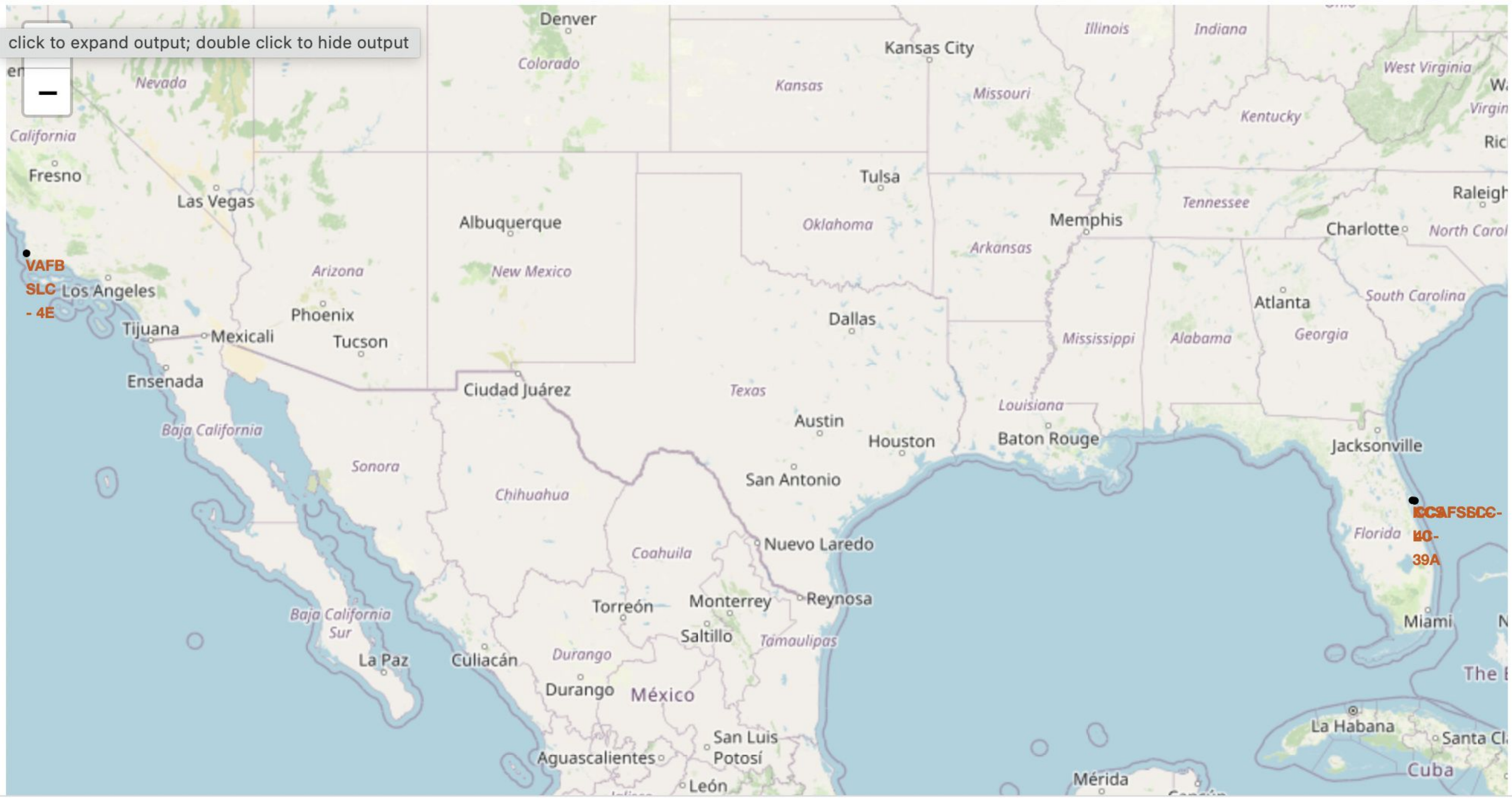
```
%%sql SELECT "Landing _Outcome", COUNT("Landing _Outcome") as 'Count' FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE '%Success%' GROUP BY "Landing _Outcome" ORDER BY 'Count' DESC
```

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

Landing _Outcome	Count
Success (ground pad)	9
Success (drone ship)	14
Success	38

SECTION 3:

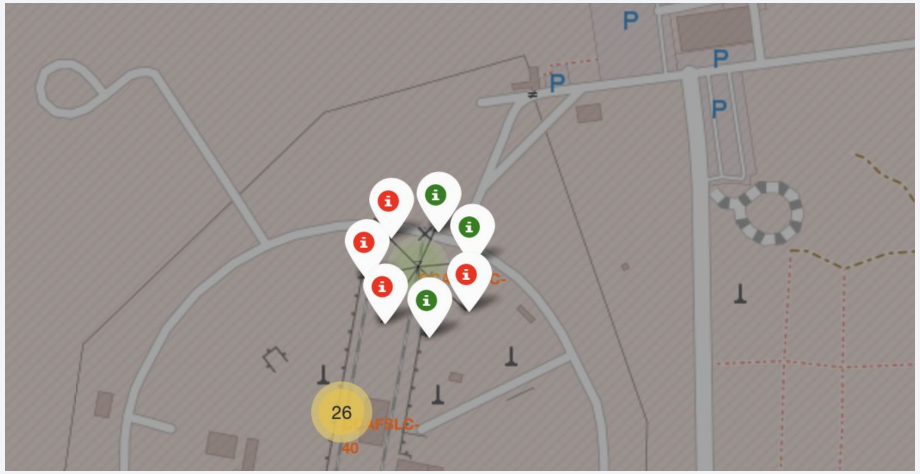
LAUNCH SITE PROXIMITIES

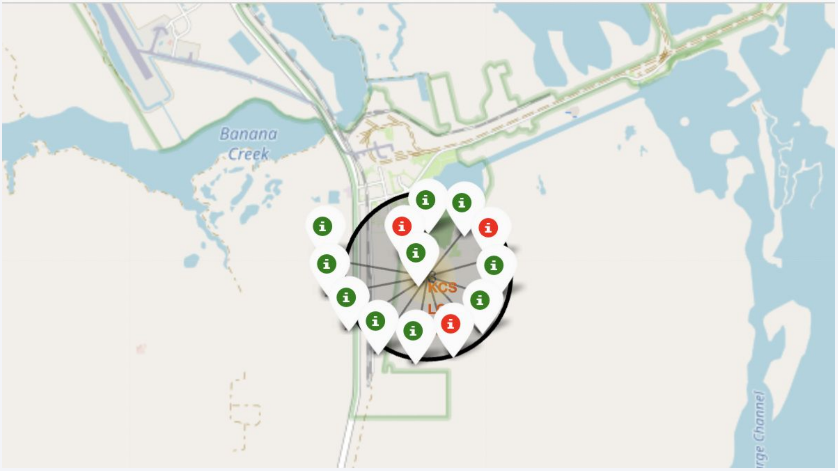


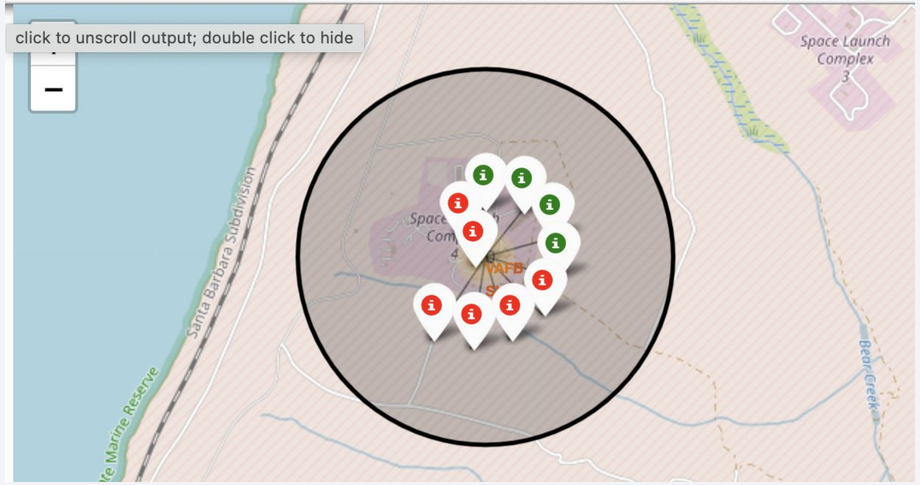
click to expand output; double click to hide output

<Folium Map Screenshot 2>

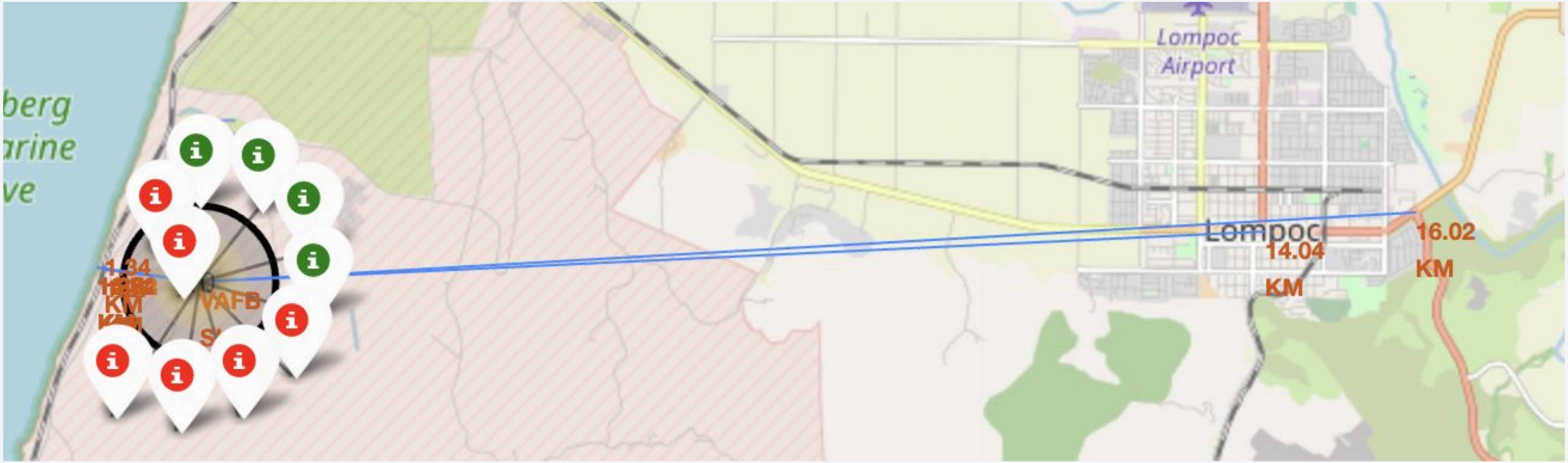
7
AFSSLC-
40
Road

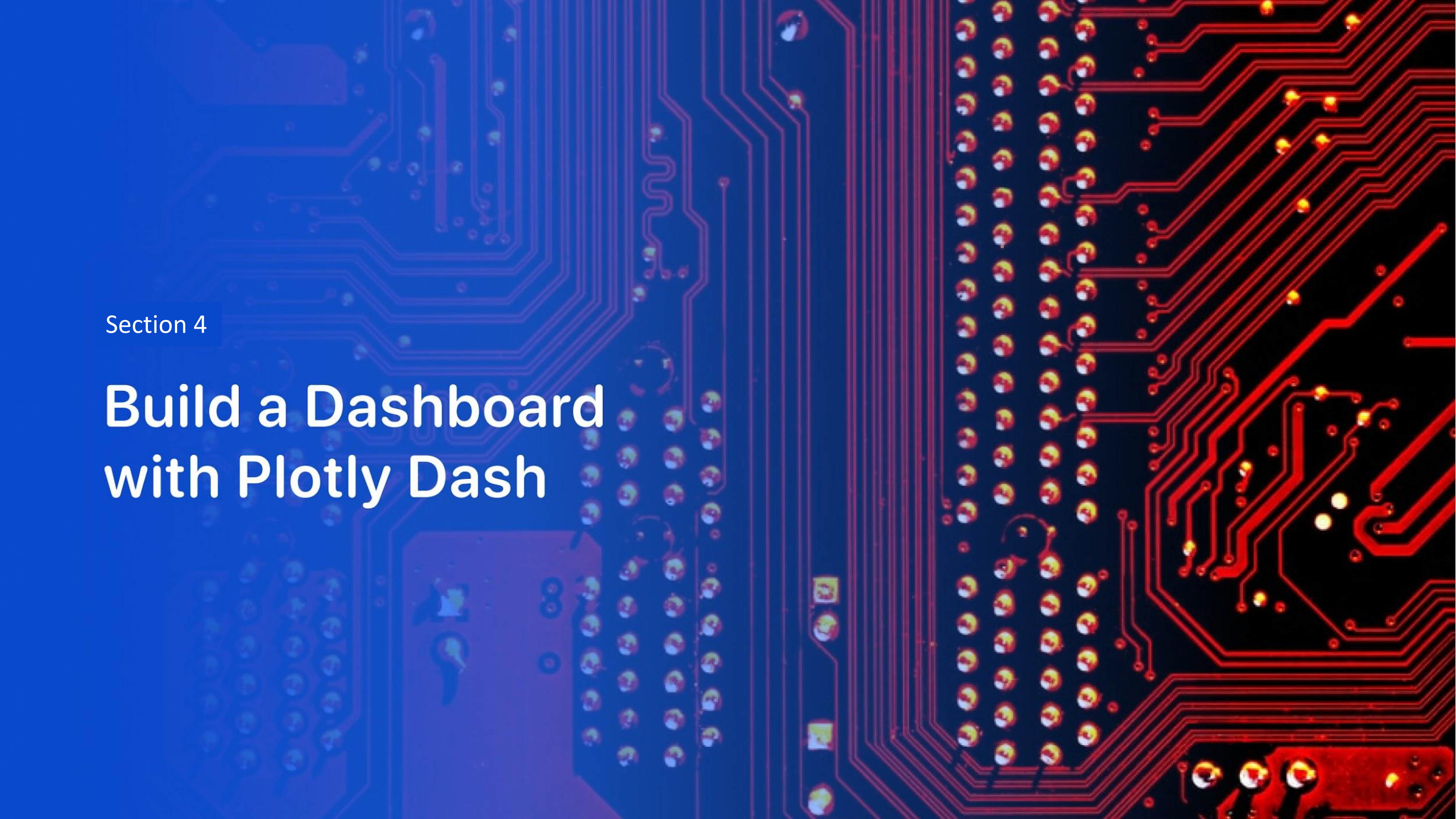





click to unscroll output; double click to hide

<Folium Map Screenshot 3>

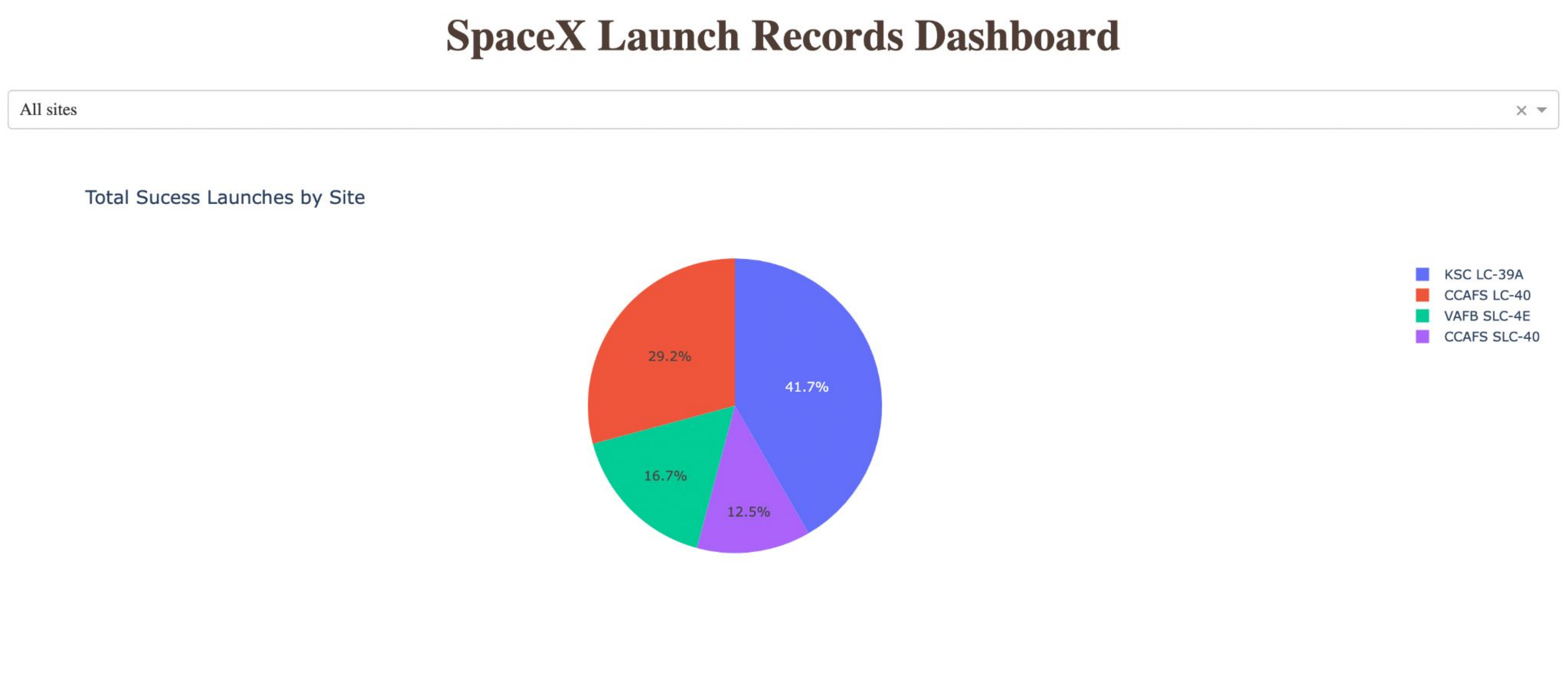

berg
arine
ve
Lompoc
Airport
Lompoc
14.04
KM
16.02
KM
1.34
KM
VAFB



Section 4

Build a Dashboard with Plotly Dash

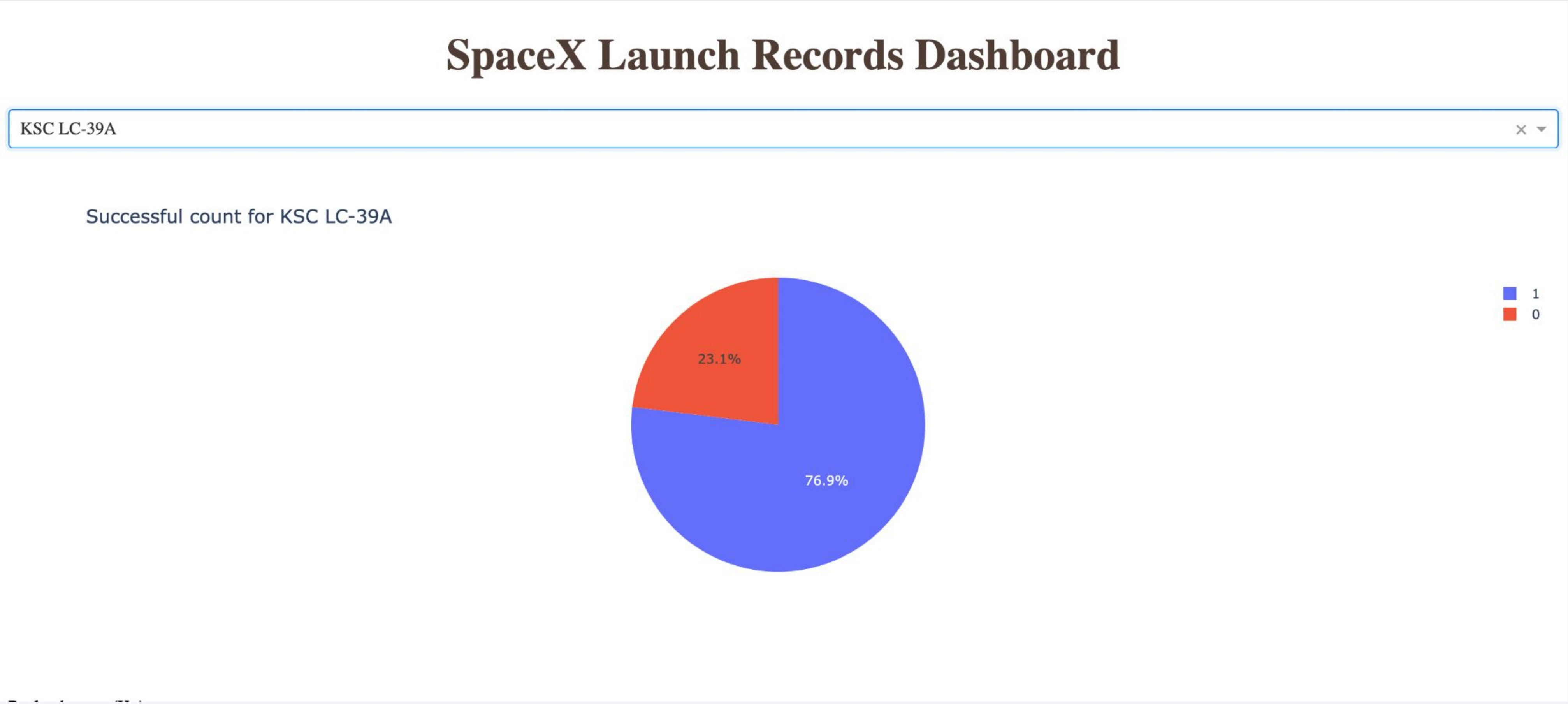
<Dashboard Screenshot 1>



SpaceX Launch Records Dashboard

All sites

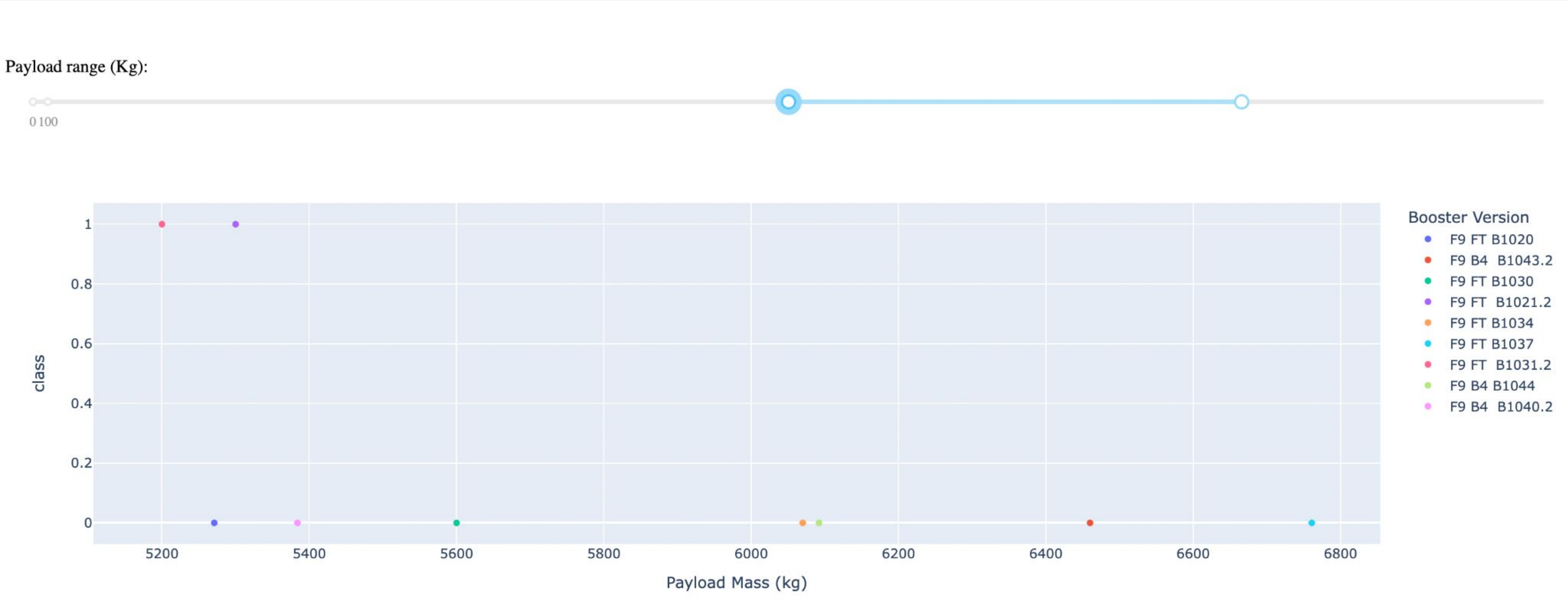
Launch site with highest launch success ratio



SpaceX Launch Records Dashboard

KSC LC-39A

Payload vs. Launch Outcome for all sites (Payload mass between 5,200 - 6,800 (kg)



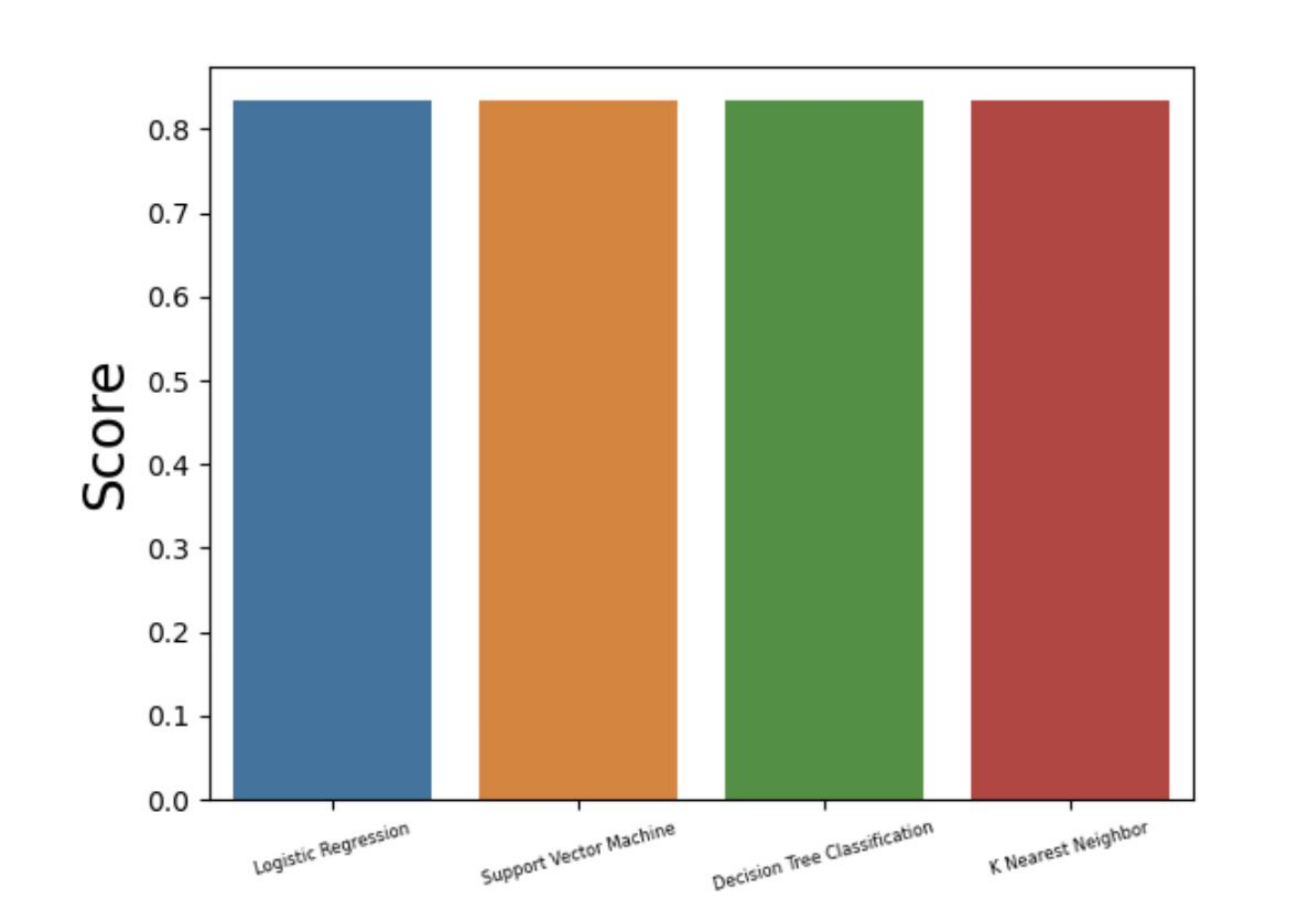
SECTION 5:

PREDICTIVE ANALYSIS (CLASSIFICATION)

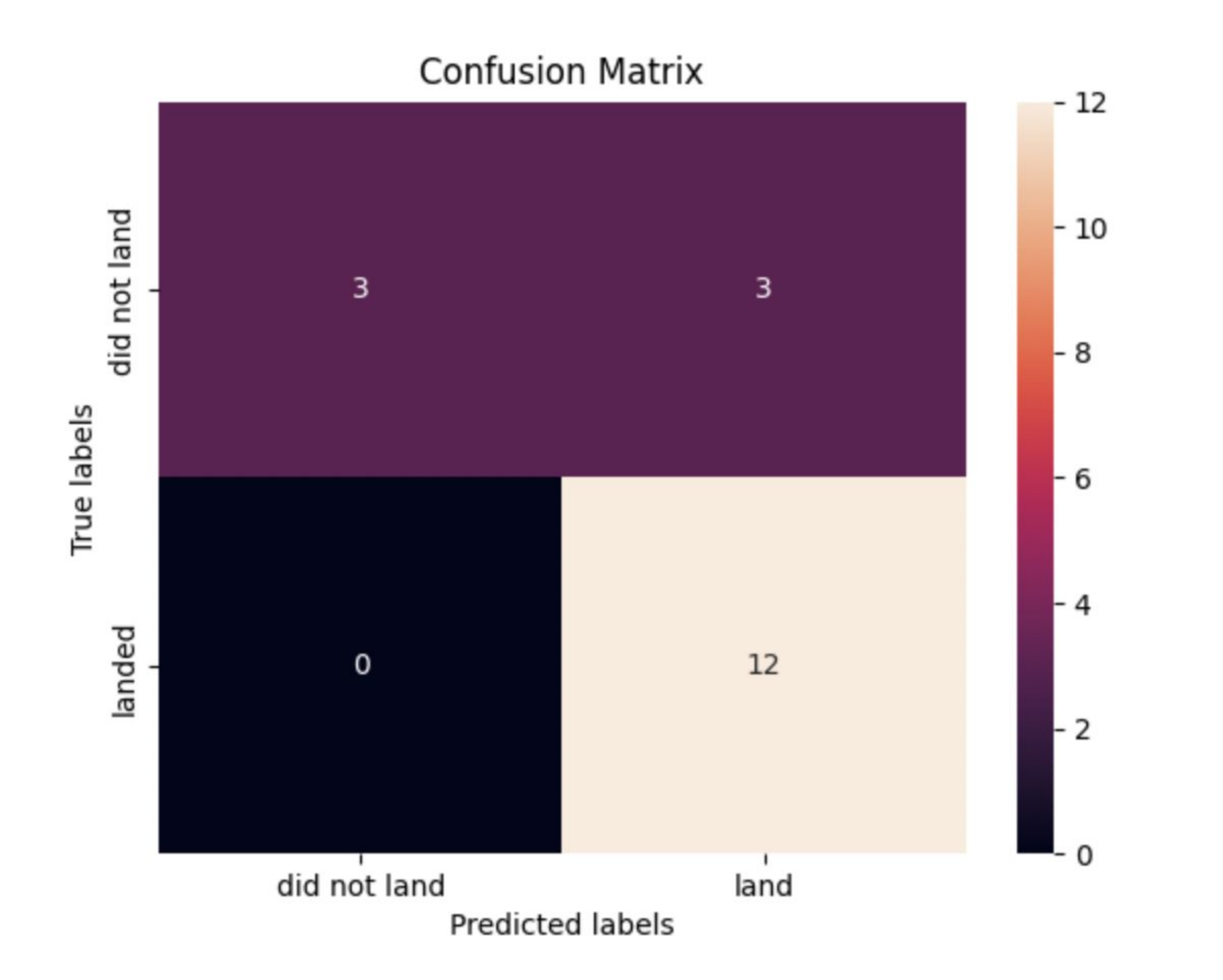
Classification Accuracy

Practically all these algorithms give the same result.

Accuracy score: 0.833



Confusion Matrix



THANK YOU!