IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Samuel Agbonika
01/11/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- the get request of the SpaceX API was used to collect data, clean the requested data and did some basic data wrangling and formatting.

- This is the link to the notebook:

https://github.com/AgbaSparks/falcon9/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Get request for launch data using API

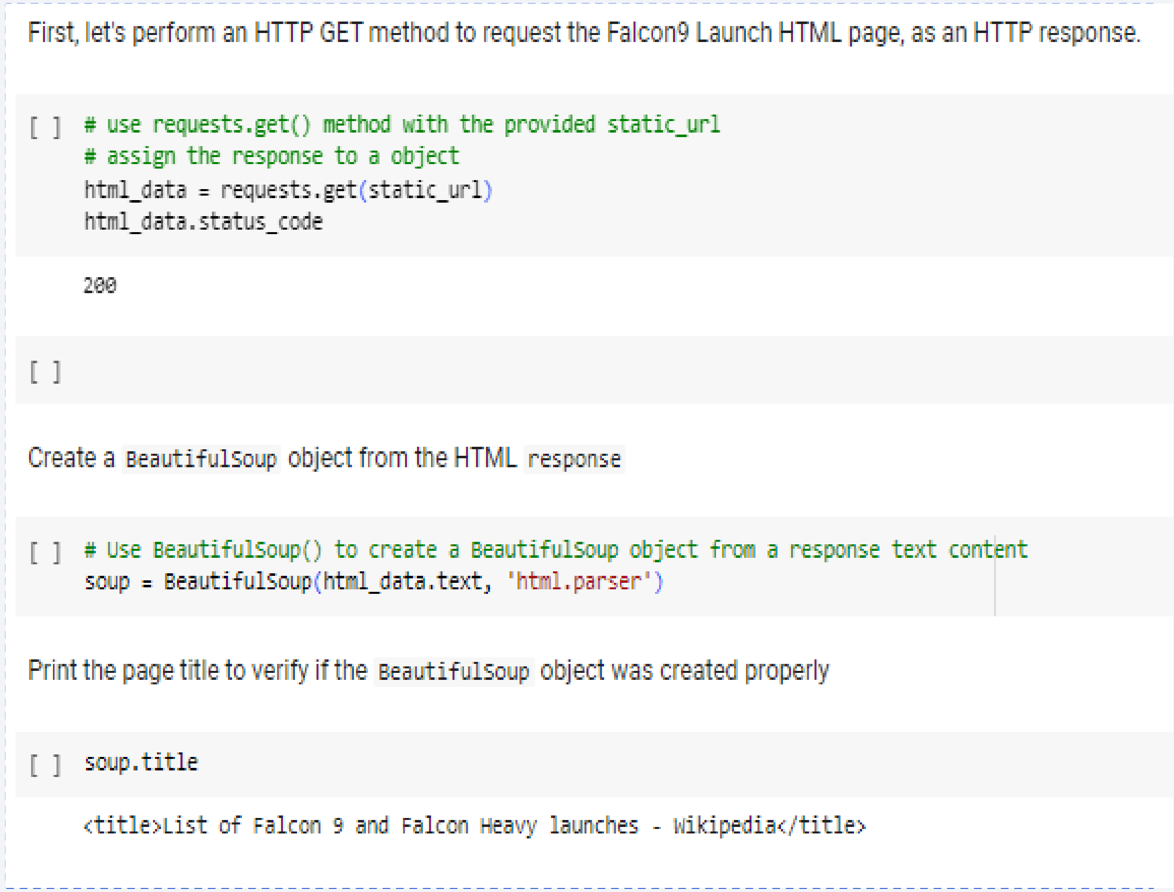
Use json_normalize to convert json to dataframe

Perform data cleaning and deal with missing values.

Data Collection - Scraping

- We applied web scrapping to Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- This is the link:

https://github.com/AgbaSparks/falcon9/blob/main/jupyter-labs-webscraping.ipynb



First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

    200
```

```
[ ]
```

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

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

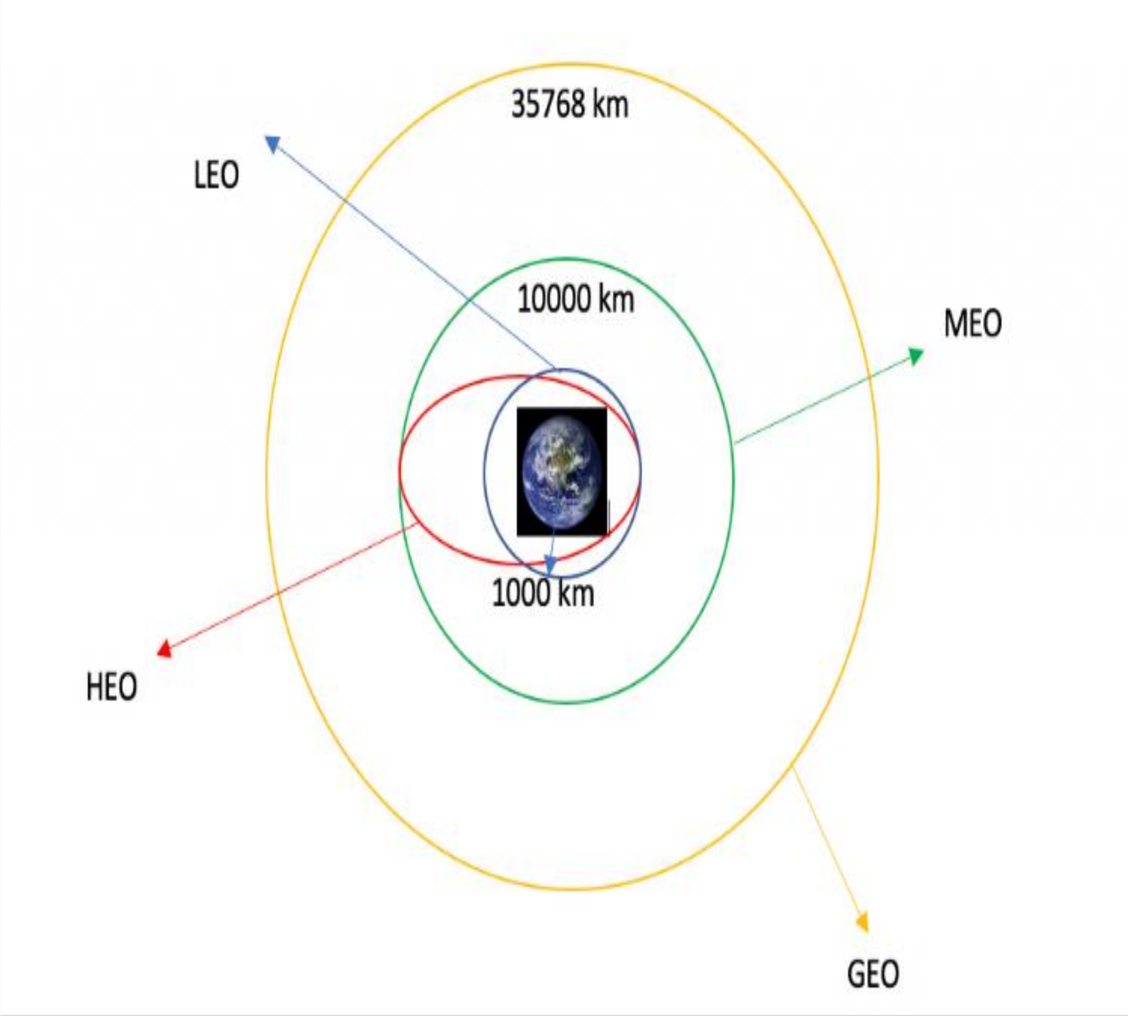
Print the page title to verify if the `BeautifulSoup` object was created properly

```
[ ] soup.title

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

Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- This is the link:

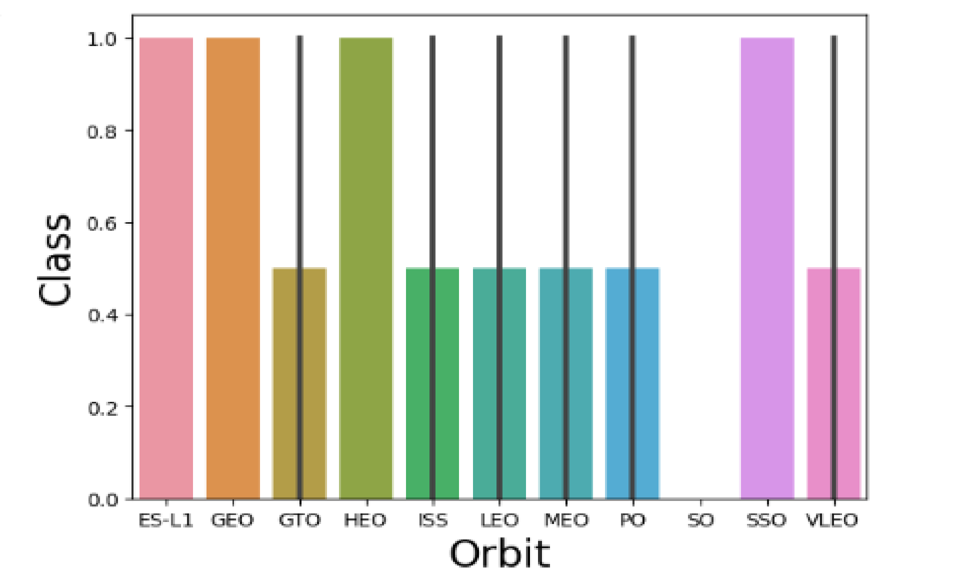
https://github.com/AgbaSparks/falcon9/blob/main/Data_Wrangling_Falcon9.ipynb

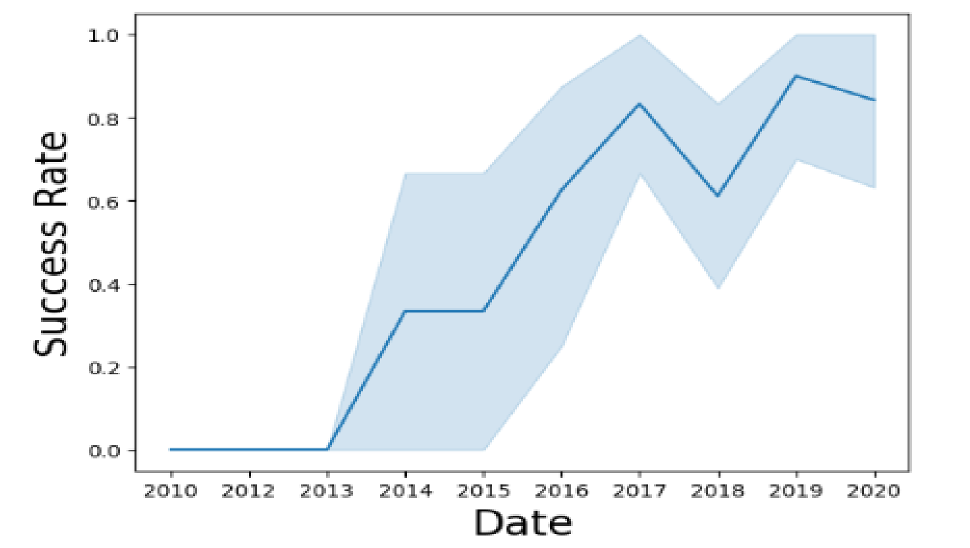


EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend. Using scatter plots, barplots, and area plots, we wanted to find out whether the Flight number and payload mass would affect landing outcomes.
- This is the notebook link:

https://github.com/AgbaSparks/falcon9/blob/main/EDA_with_Visualization.ipynb





EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names
  - The names of the booster_versions which have carried the maximum payload mass.
- https://github.com/AgbaSparks/falcon9/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- https://github.com/AgbaSparks/falcon9/blob/main/DashBoarding_with_Folium.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- This was done to show the success rates of specific launch site with regards to the payload mass.
- https://github.com/AgbaSparks/falcon9/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- https://github.com/AgbaSparks/falcon9/blob/main/08_SpaceX_Predictive_Analytics.ipynb

Results

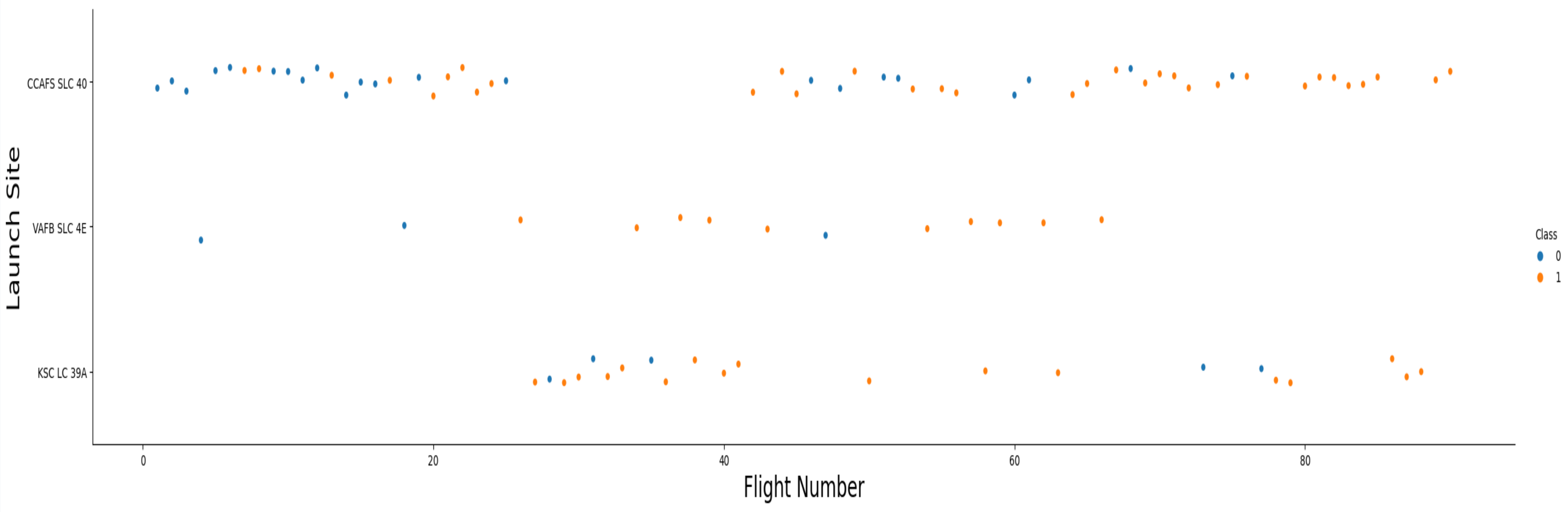
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

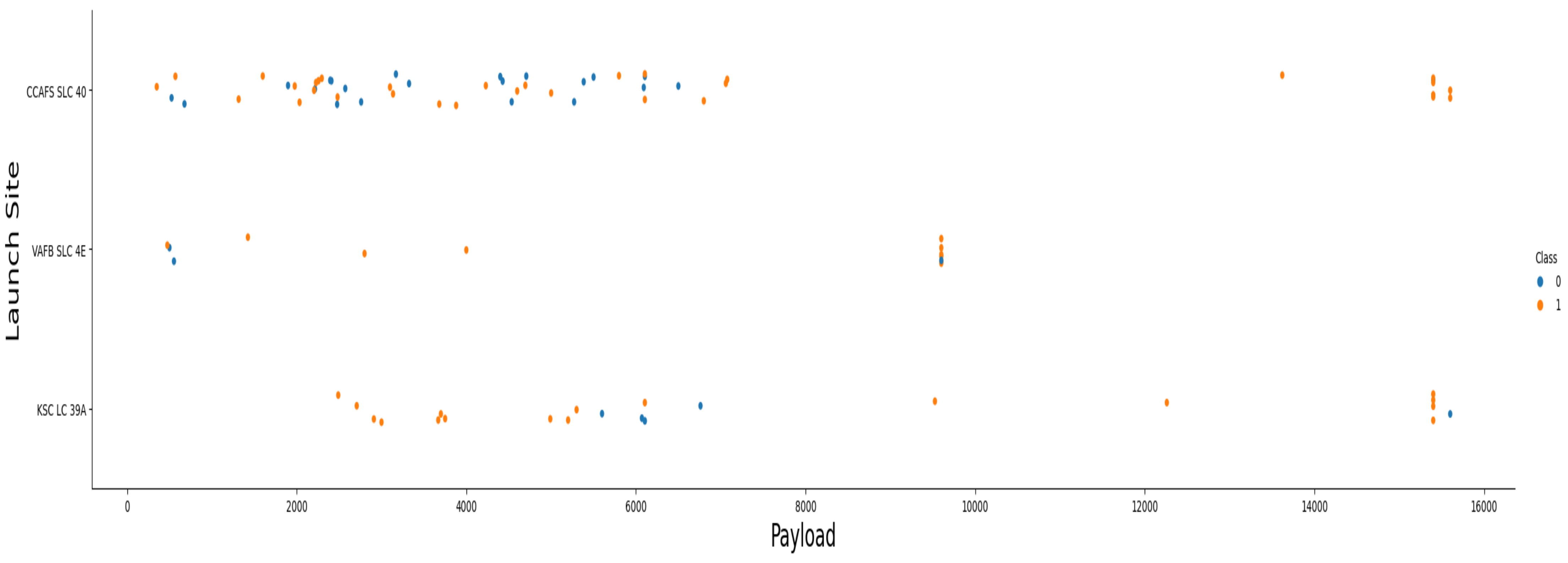
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



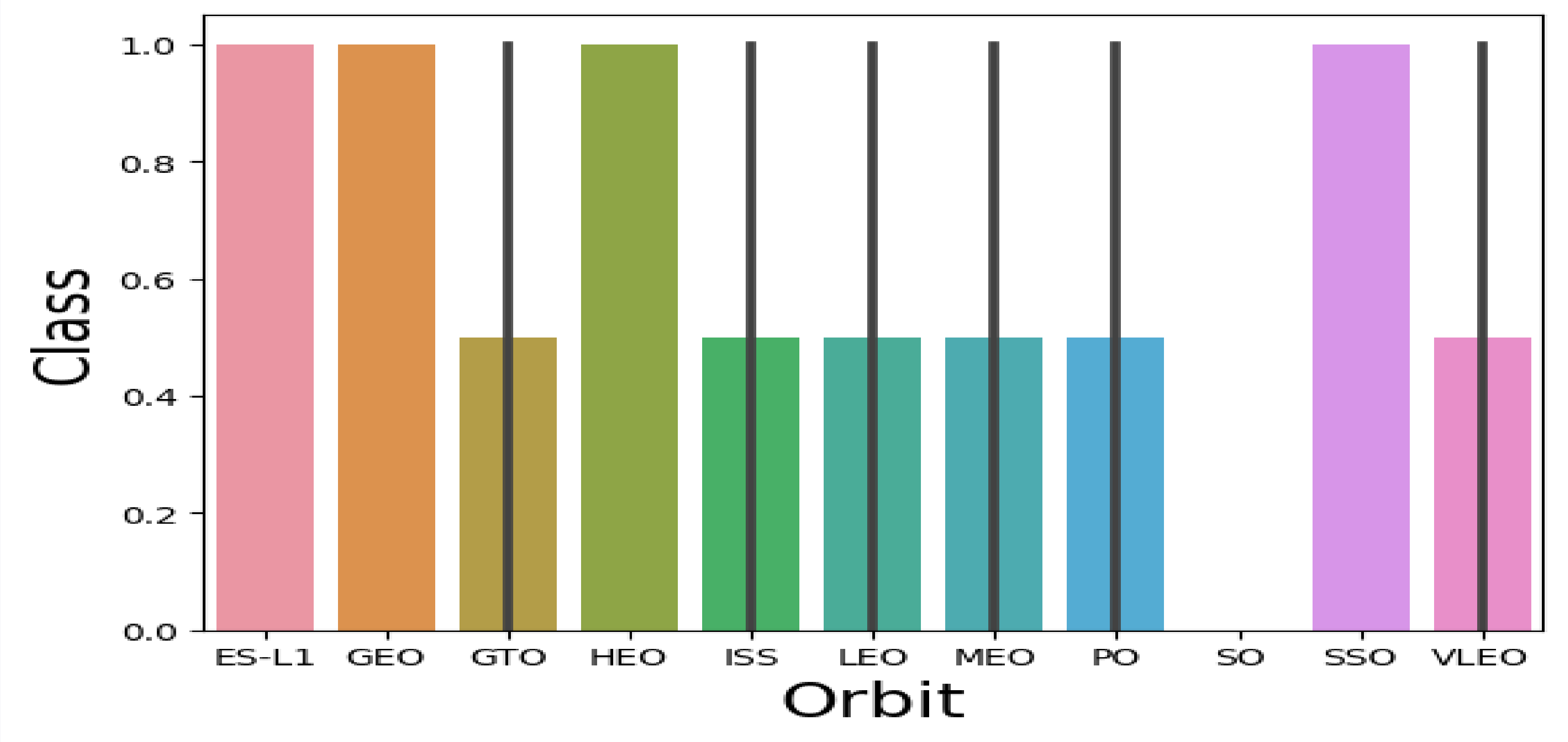
Payload vs. Launch Site

- The greater the payload mass for site CCAFS SLC 40, the higher the success rate for the rocket.



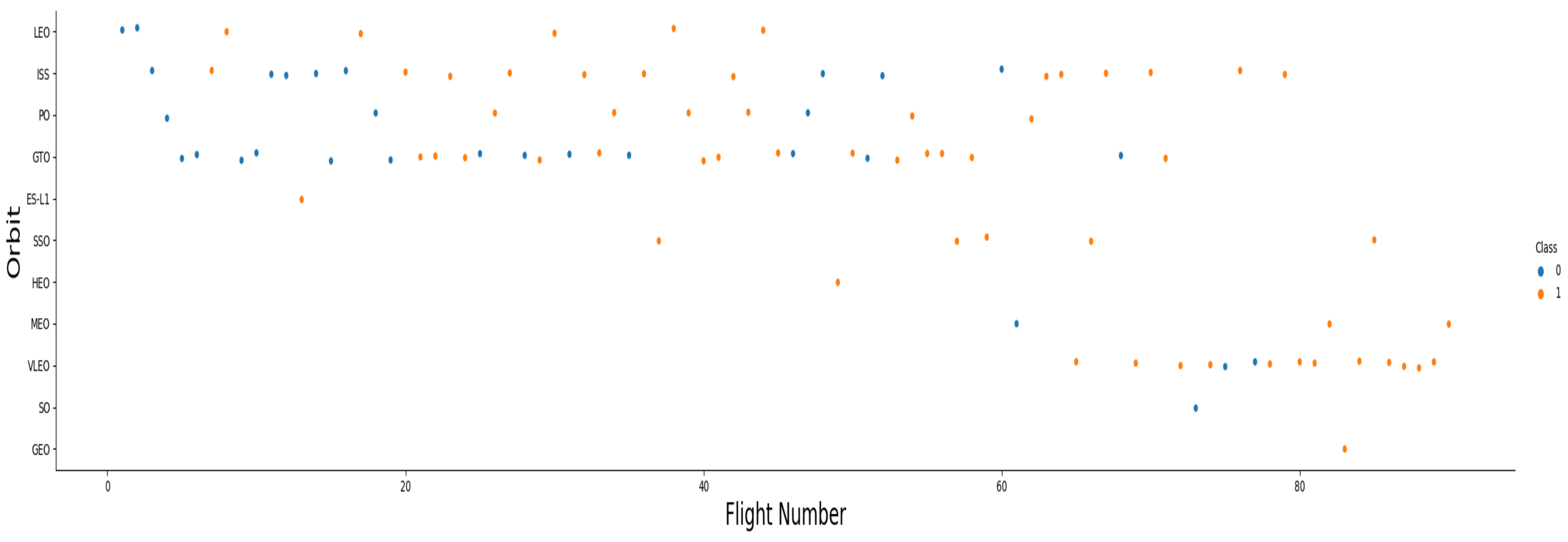
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO had the most success rate.



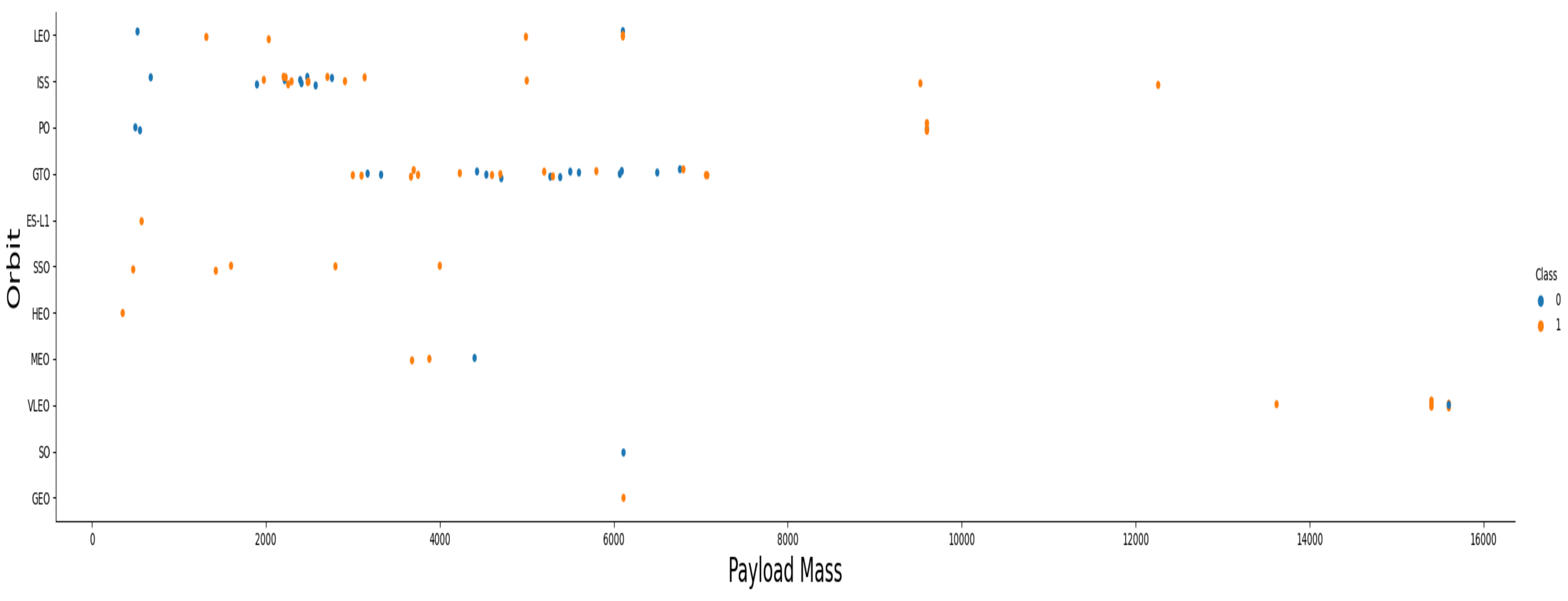
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



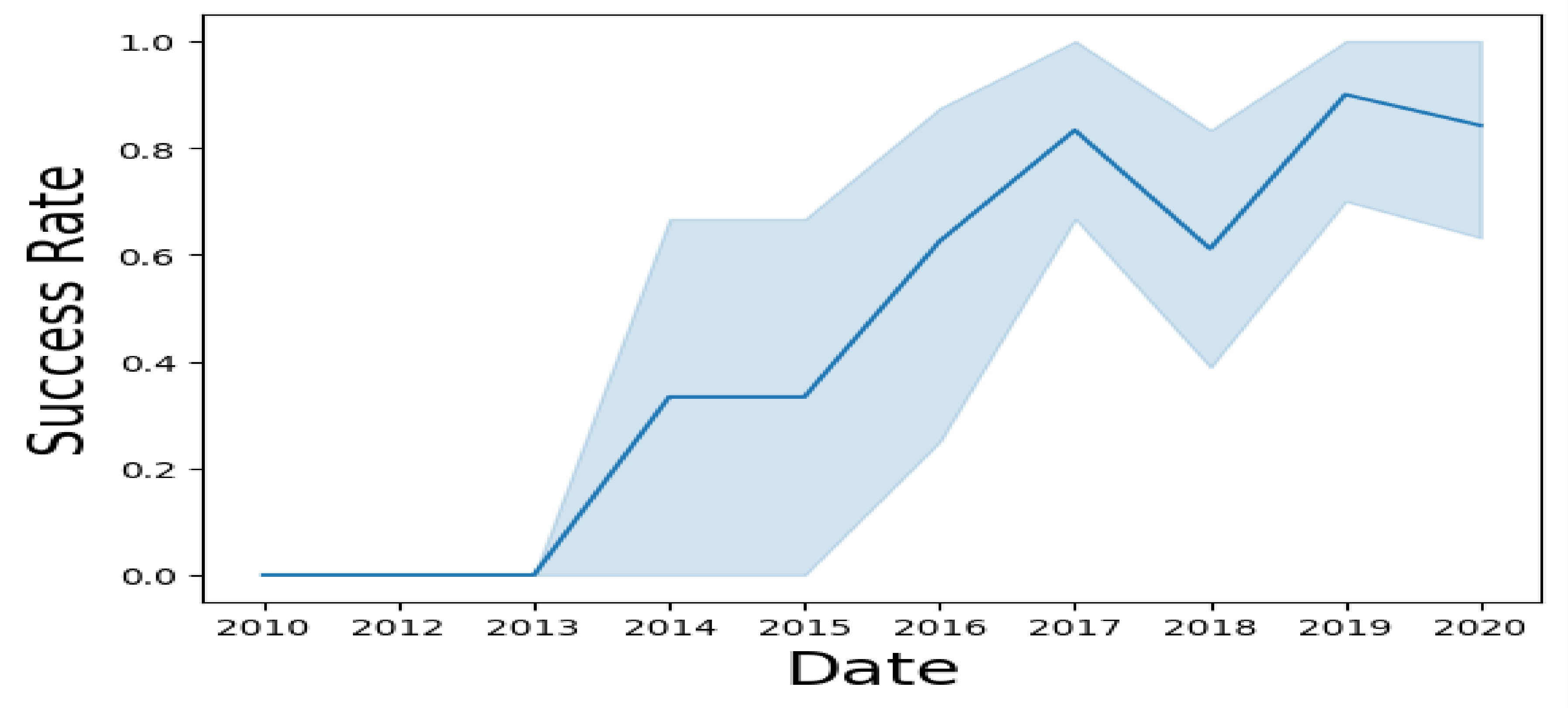
Payload vs. Orbit Type

- It was observed that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



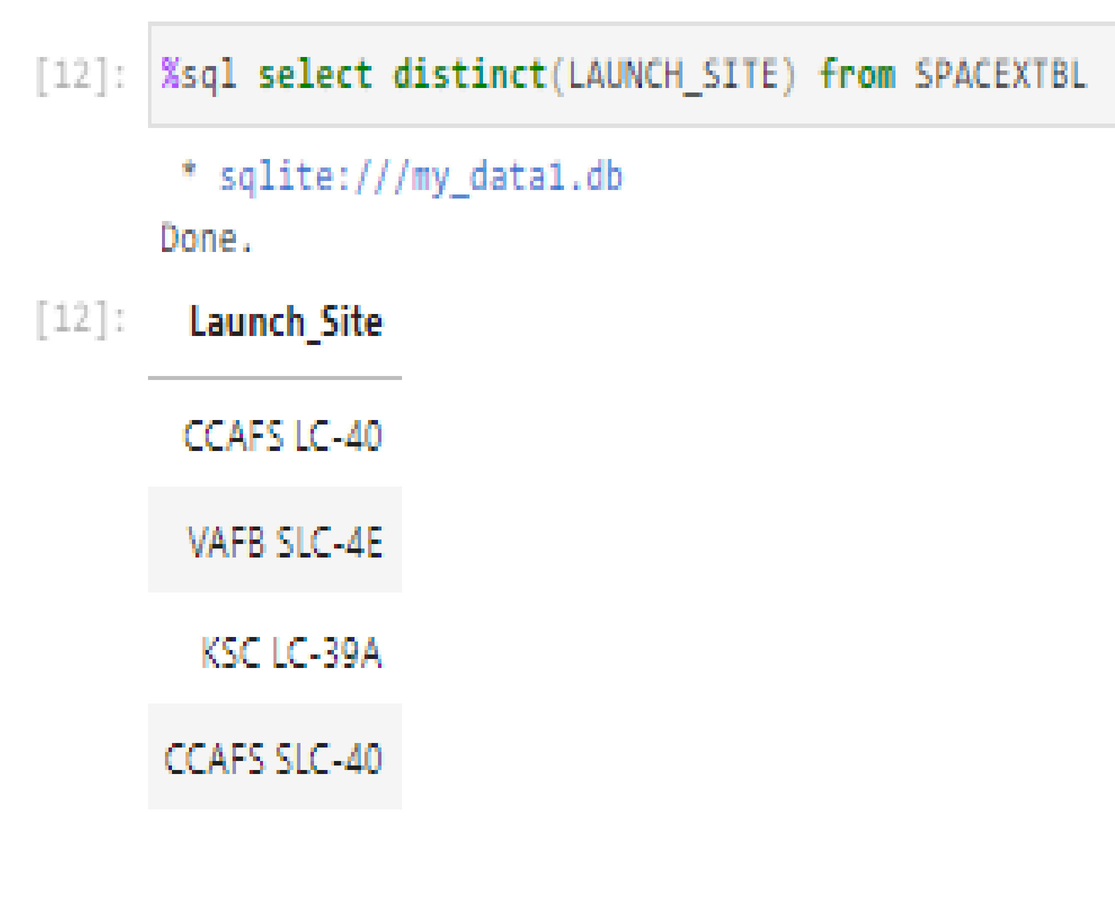
Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

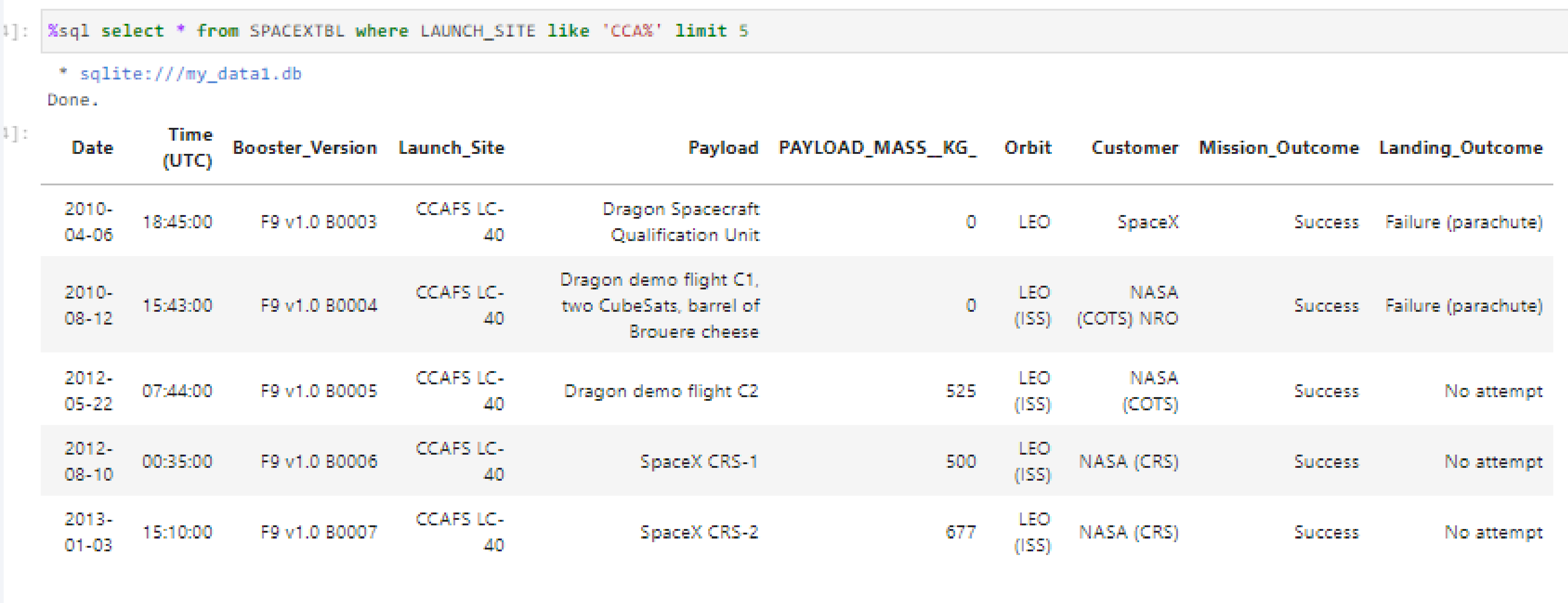
- The key word **DISTINCT** was used to show only unique launch sites from the SpaceX data.



```
[12]: %sql select distinct(LAUNCH_SITE) from SPACEXTBL

 * sqlite:///my_data1.db
Done.

[12]: Launch_Site
      CCAFS LC-40
      VAFB SLC-4E
      KSC LC-39A
      CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

- With the key word “like”, we used the query above to display 5 records where launch sites begin with `CCA`

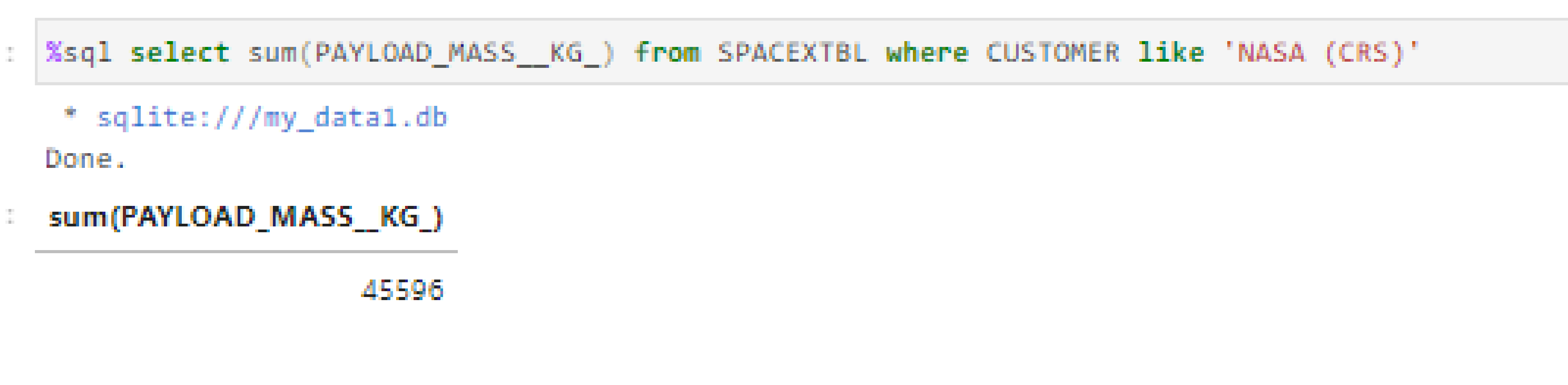


```
1]: %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5

     * sqlite:///my_data1.db
    Done.
1]:
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	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 calculated the total payload carried by boosters from NASA as 45596 using the query below



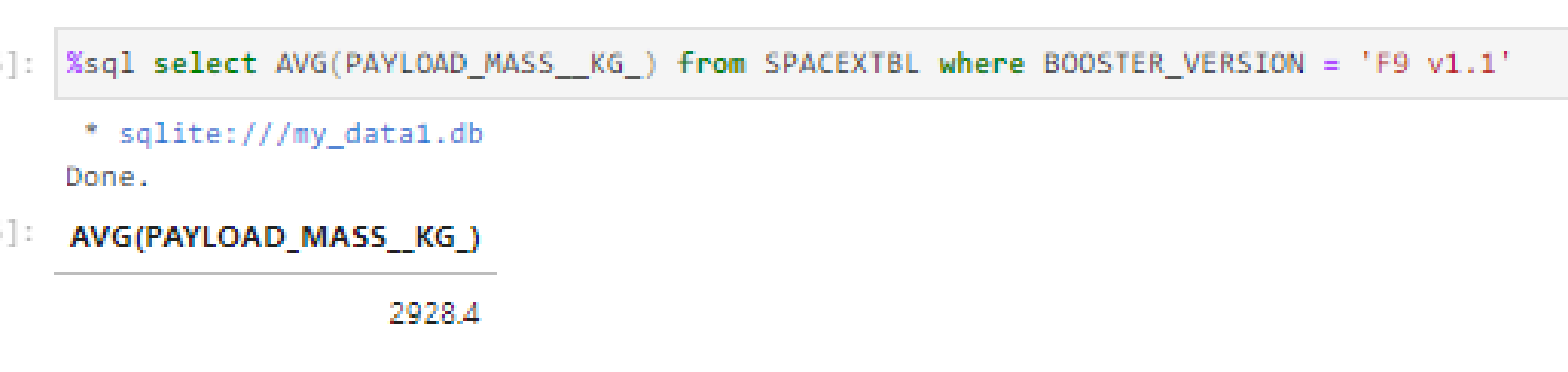
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER like 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4



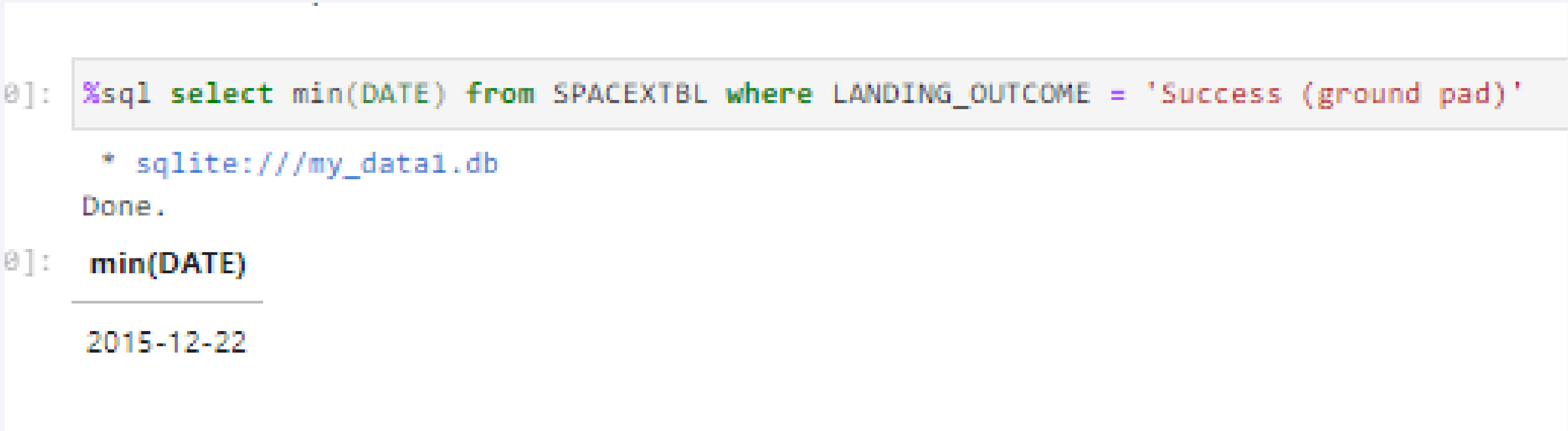
```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

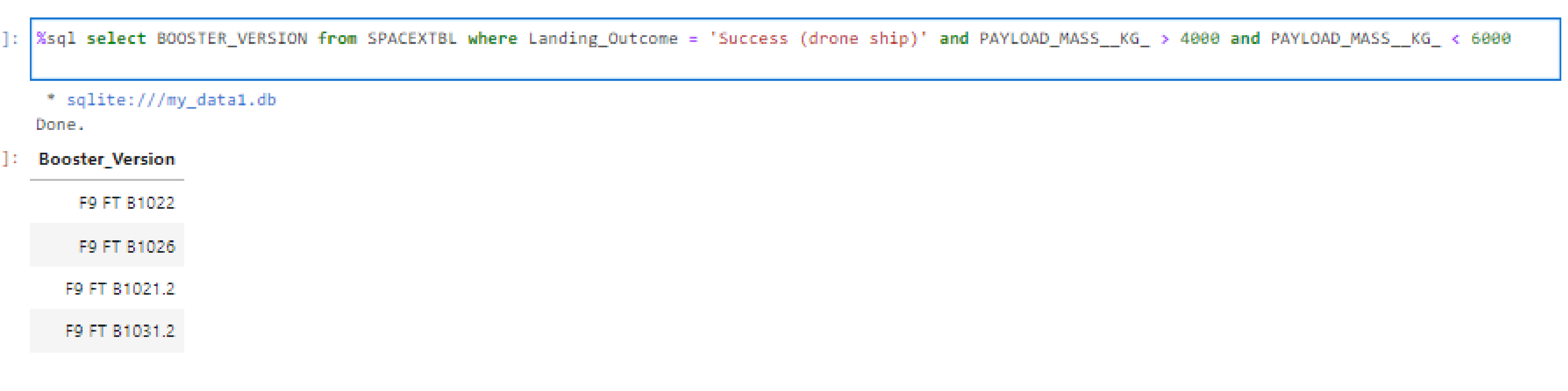
- It was observed that the date of the first successful landing outcome on ground pad was 22nd December 2015



```
0]: %sql select min(DATE) from SPACEXTBL where LANDING_OUTCOME = 'Success (ground pad)'

     * sqlite:///my_data1.db
    Done.

0]:  min(DATE)
     2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Booster versions F9 FT B1022, F9 FT B1026,F9 FT B1021.2 ,F9 FT B1031.2 all had successful drone ship landings



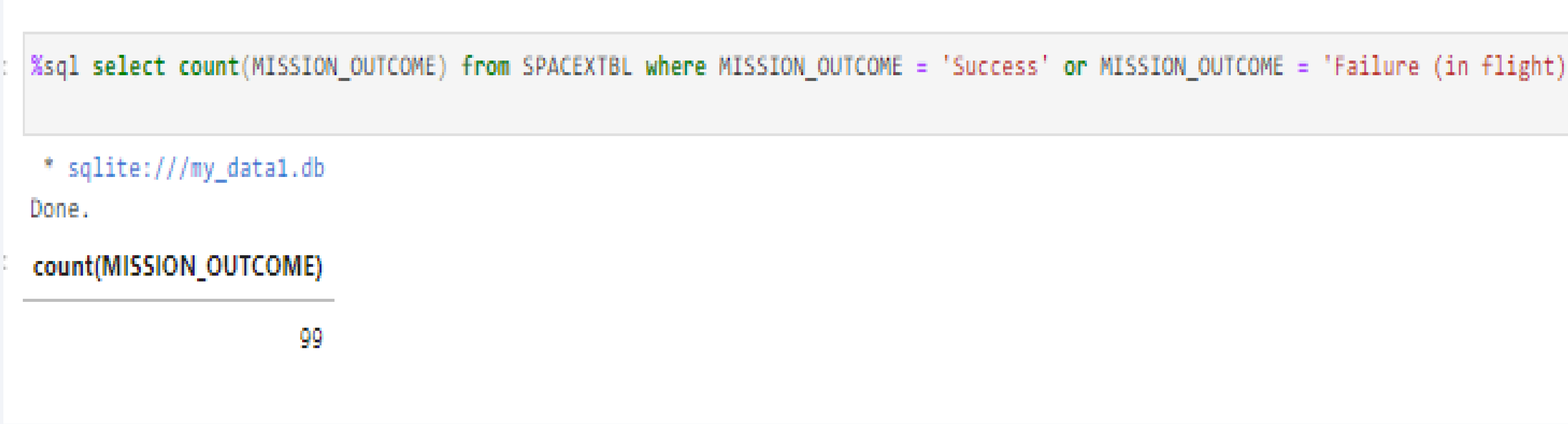
```
]: %sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000

 * 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

- We used keyword ‘or’ to filter for **WHERE** Mission Outcome was a success or a failure.



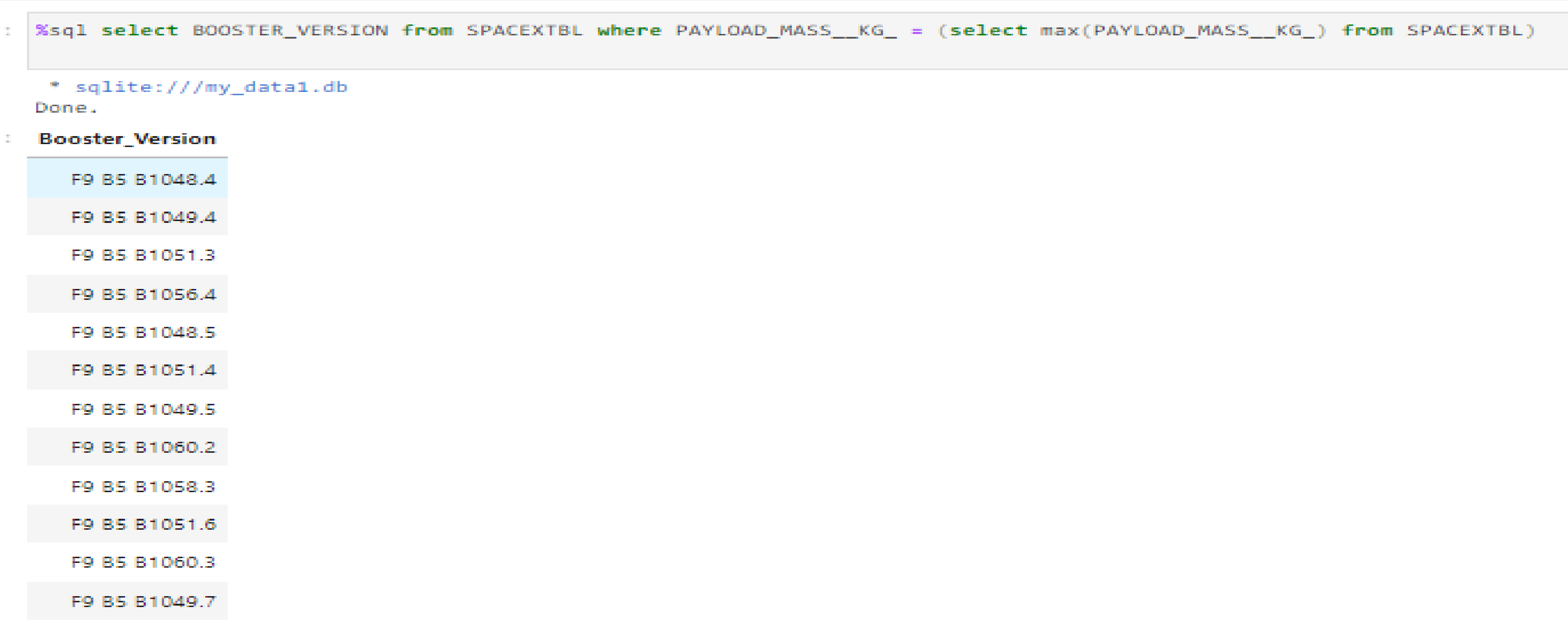
```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)
```

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

count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.



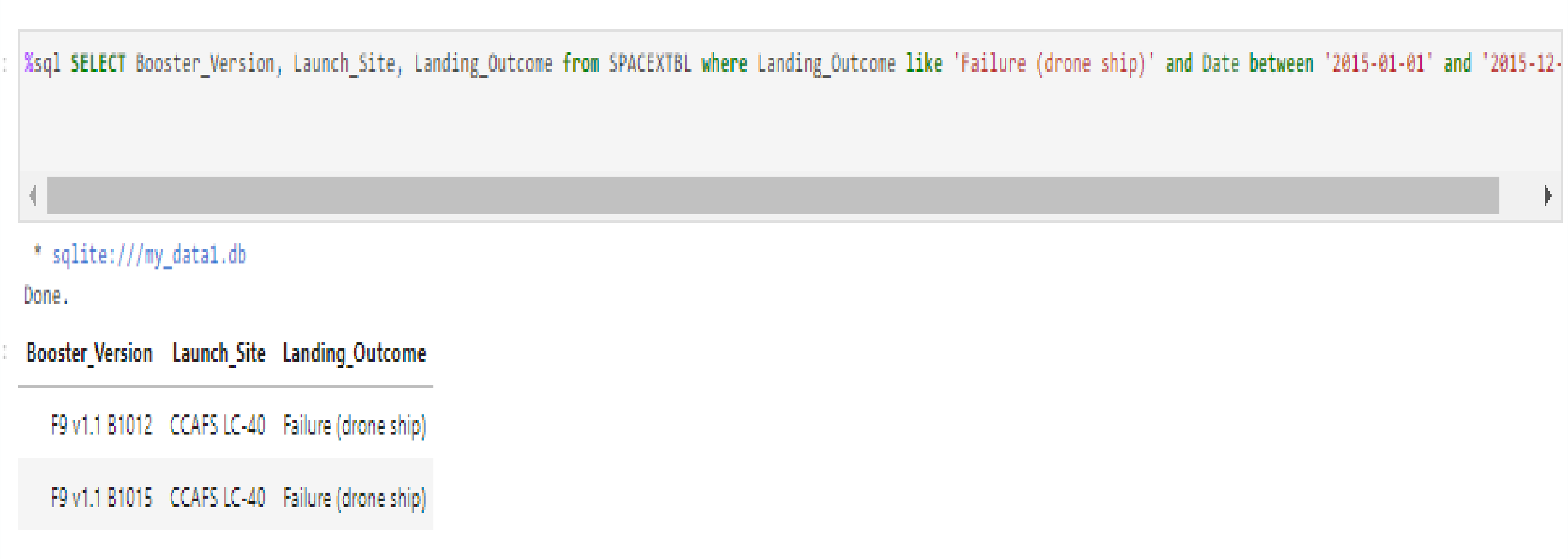
```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * 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

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015



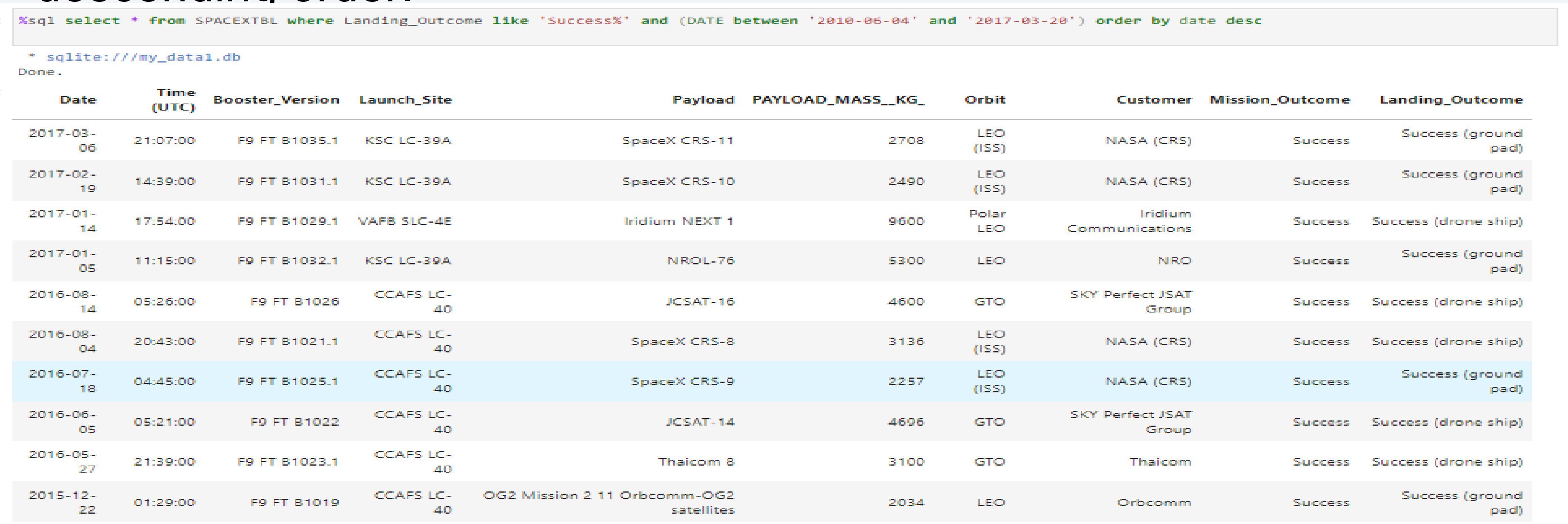
```
%sql SELECT Booster_Version, Launch_Site, Landing_Outcome from SPACEXTBL where Landing_Outcome like 'Failure (drone ship)' and Date between '2015-01-01' and '2015-12-
```

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

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

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.



```
%sql select * from SPACEXTBL where Landing_Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

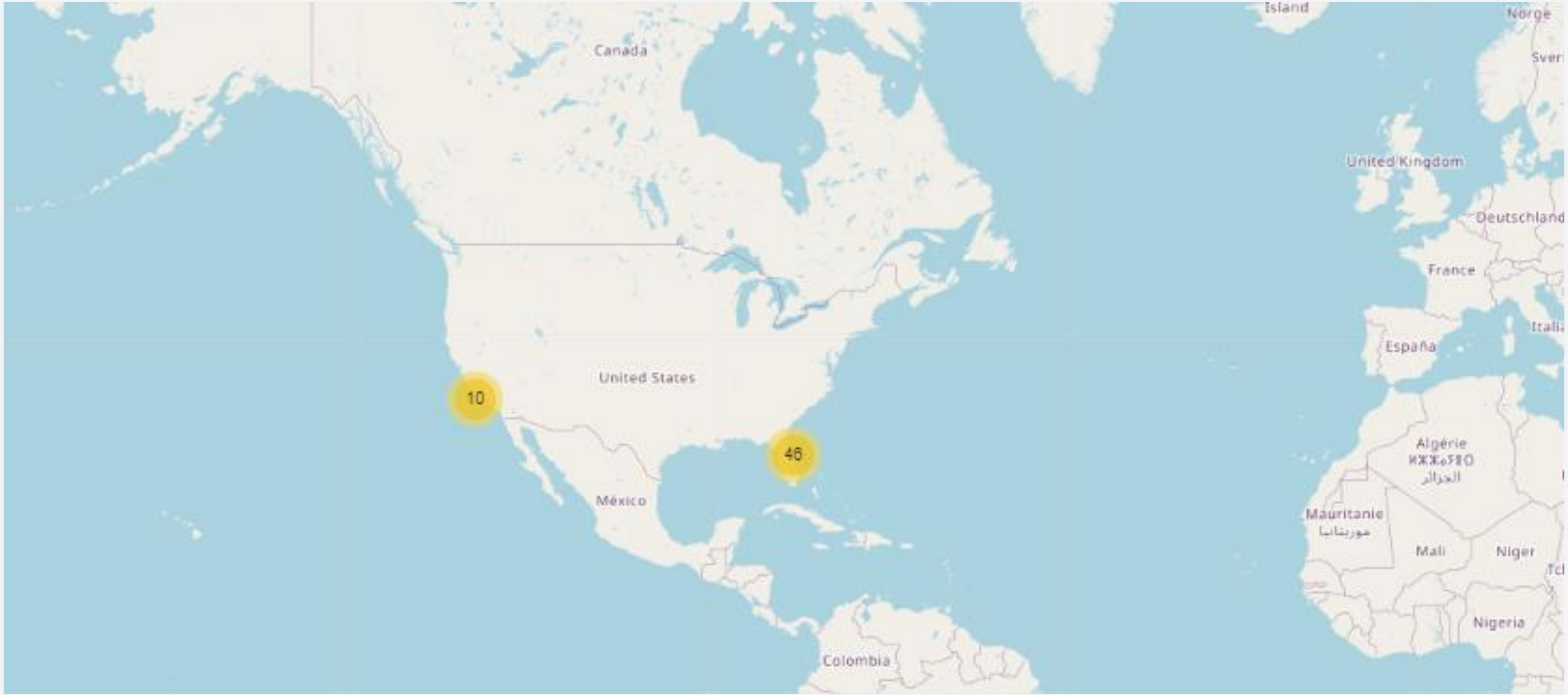
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-03-06	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-04	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-06-05	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

Launch Sites Proximities Analysis

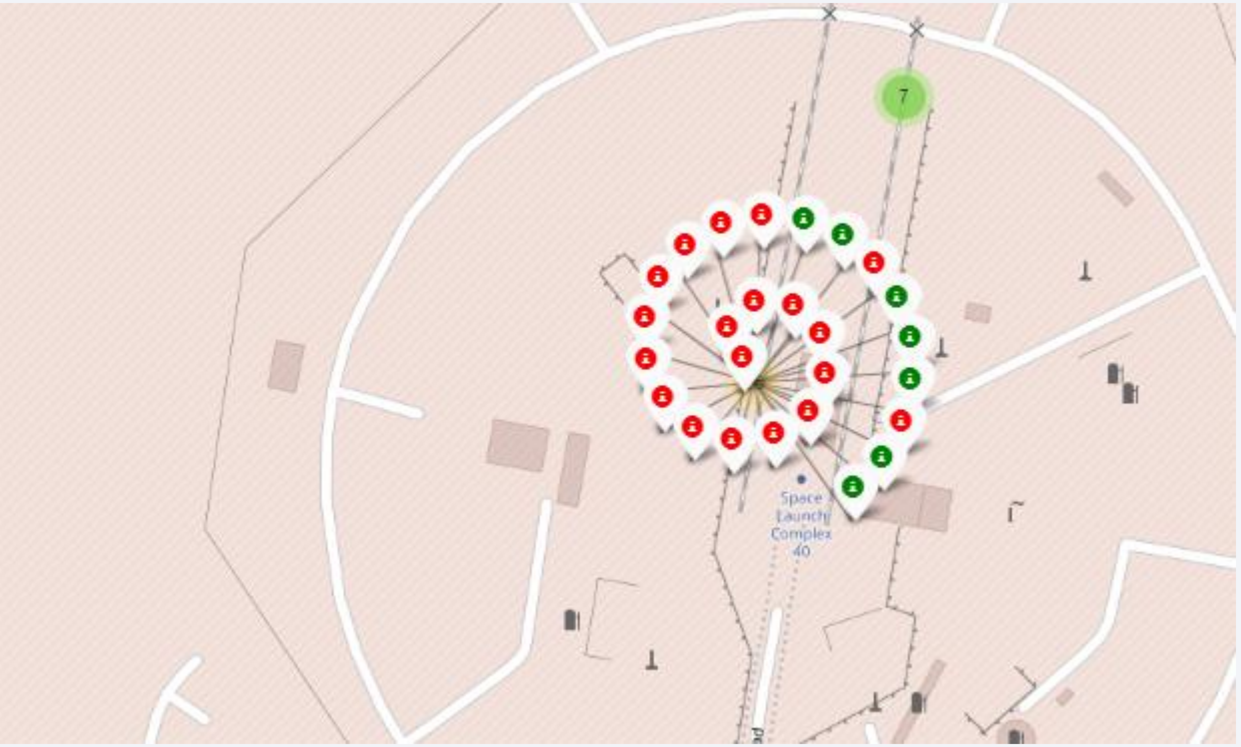
All Launch sites global marker



- We can see that all the launch sites are in the coastlines of Florida and California.

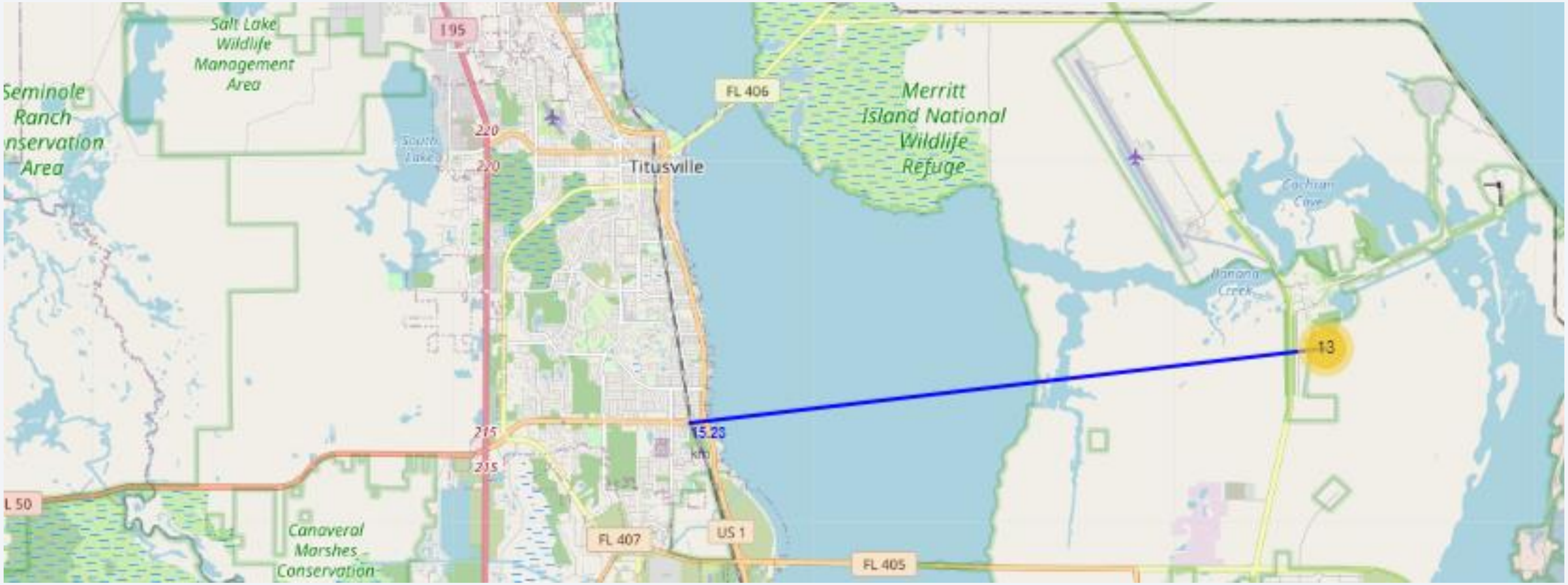
Launch sites with color markers

- Green markers show successful landing while red markers show unsuccessful landings.



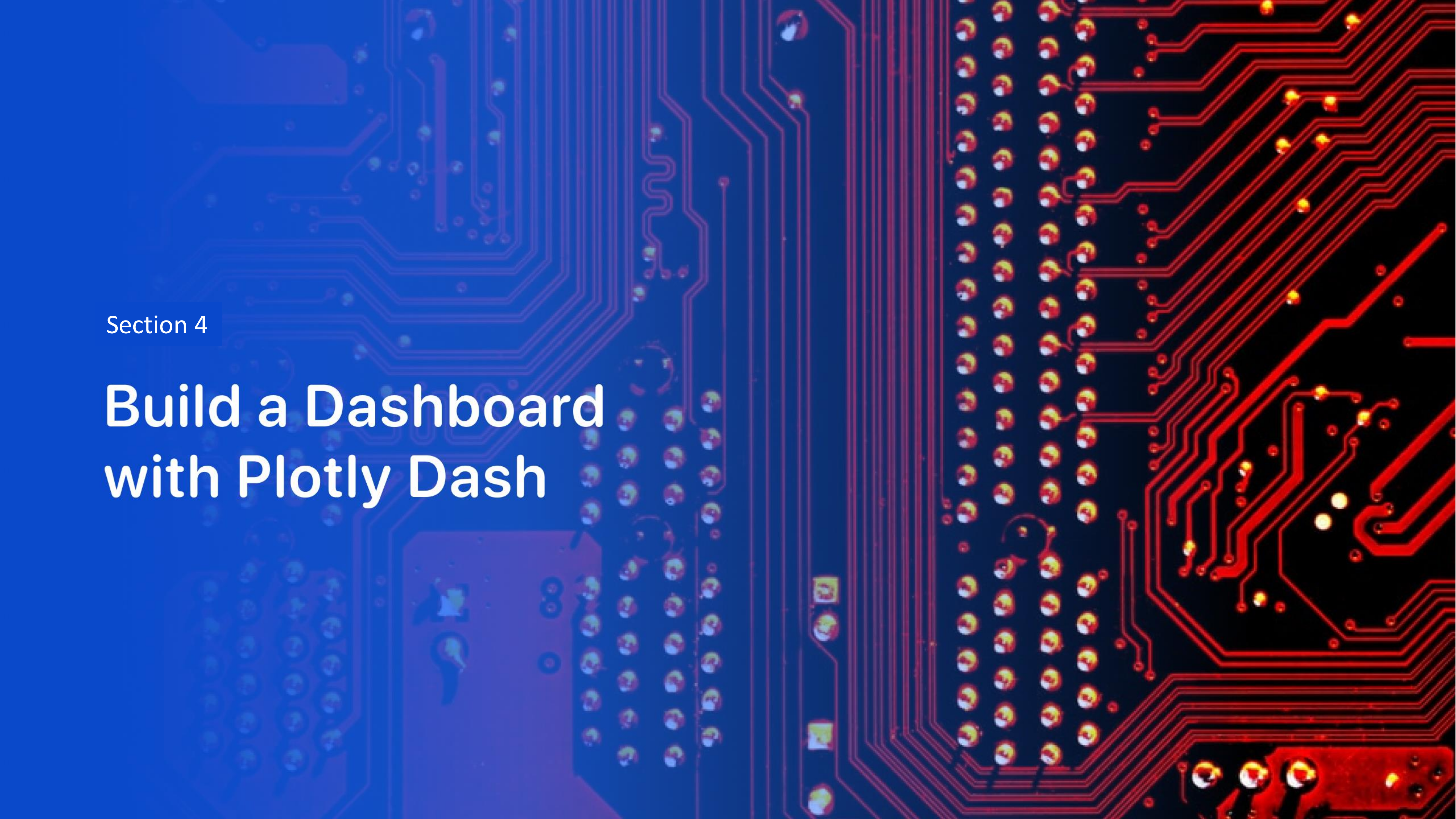
7
26
Space Launch Complex 40

Launch site proximity from landmarks

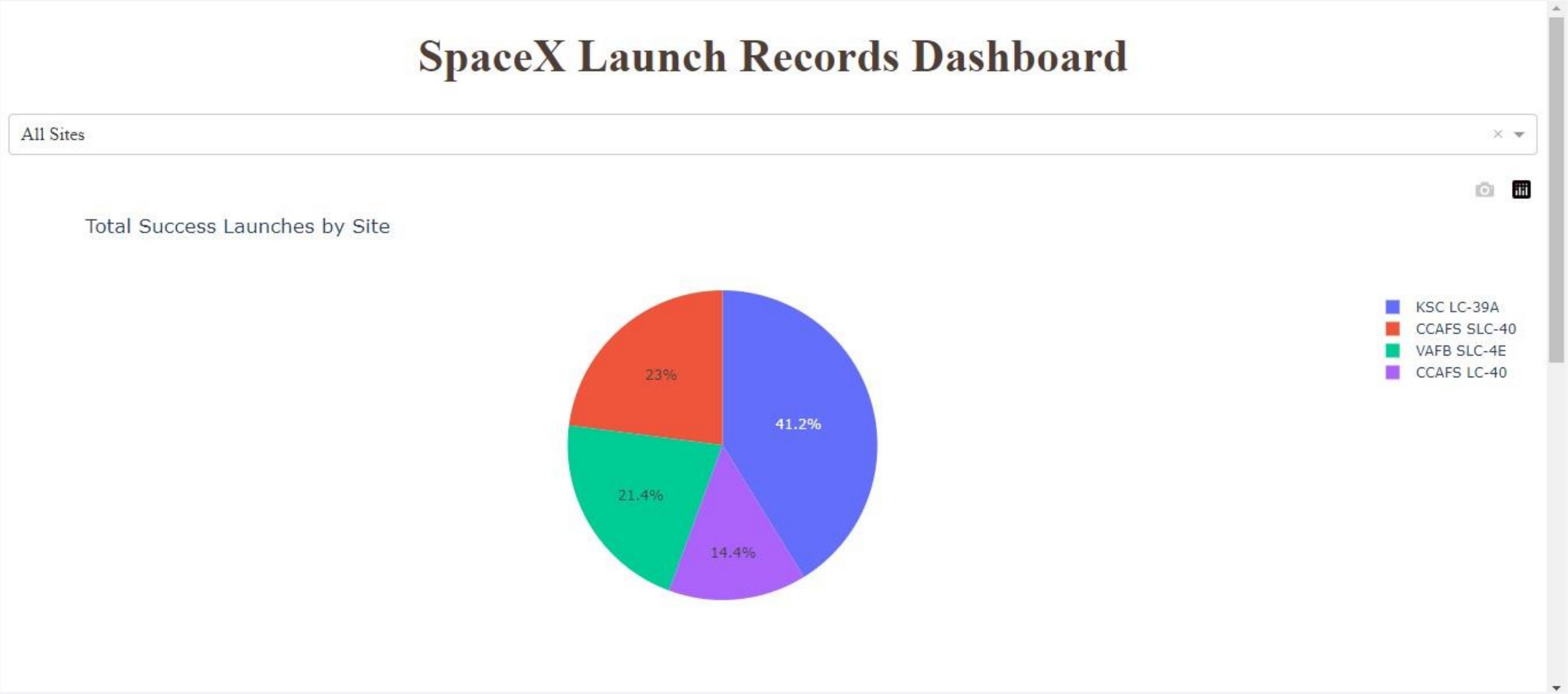

Salt Lake Wildlife Management Area
Seminole Ranch nservation Area
195
220
South Lake
Titusville
FL 406
Merritt Island National Wildlife Refuge
Cochran Cove
Banana Creek
13
15.23 km
215
L 50
Canaveral Marshes Conservation
FL 407
US 1
FL 405

Section 4

Build a Dashboard with Plotly Dash



Pie chart showing the success percentage achieved by each launch site



SpaceX Launch Records Dashboard

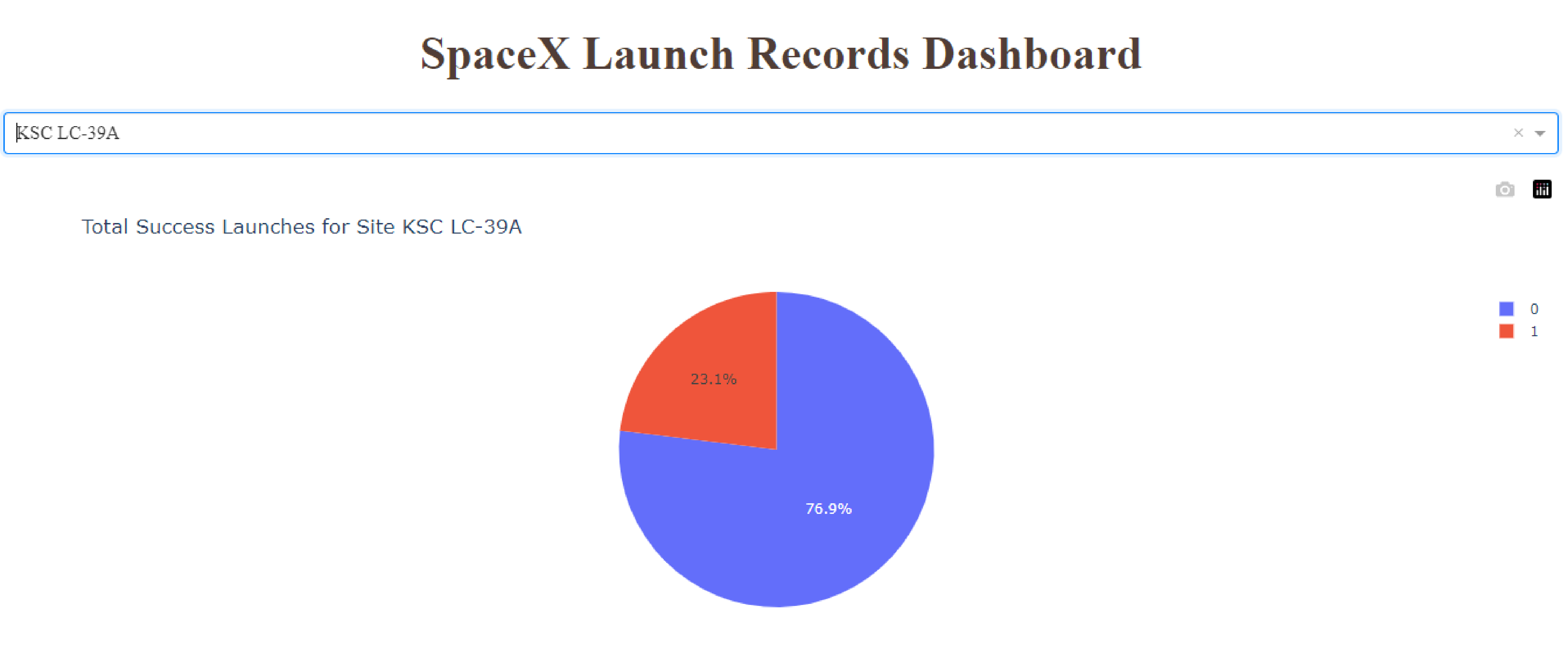
All Sites

Total Success Launches by Site

KSC LC 39A is the launch site with the highest success launches

Pie chart showing the Launch site with the highest launch success ratio

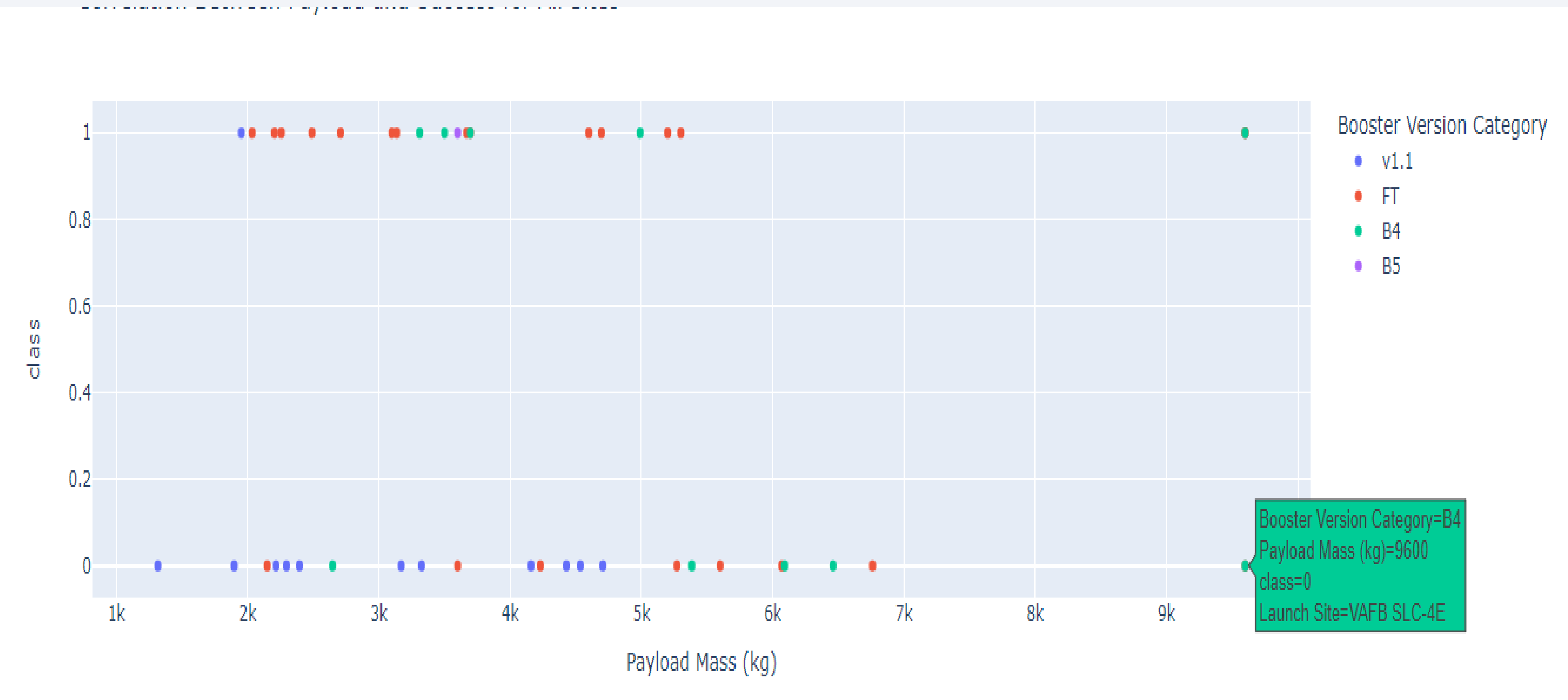
- KSC LC-39A had a 76.9% success rate while only having a failure rate of 23.1%



SpaceX Launch Records Dashboard

KSC LC-39A

Scatter plot of Payload vs Launch Outcome, with different payload selected in the range slider

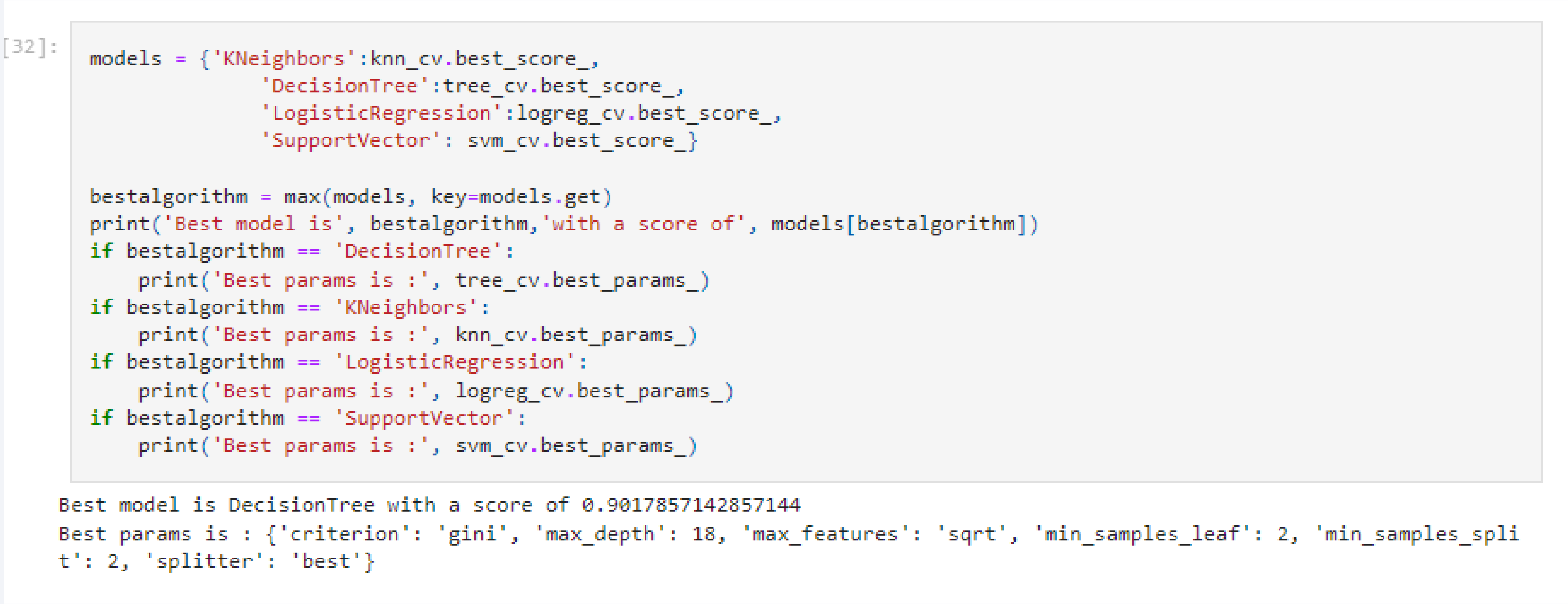


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the best model with a score of 0.901785

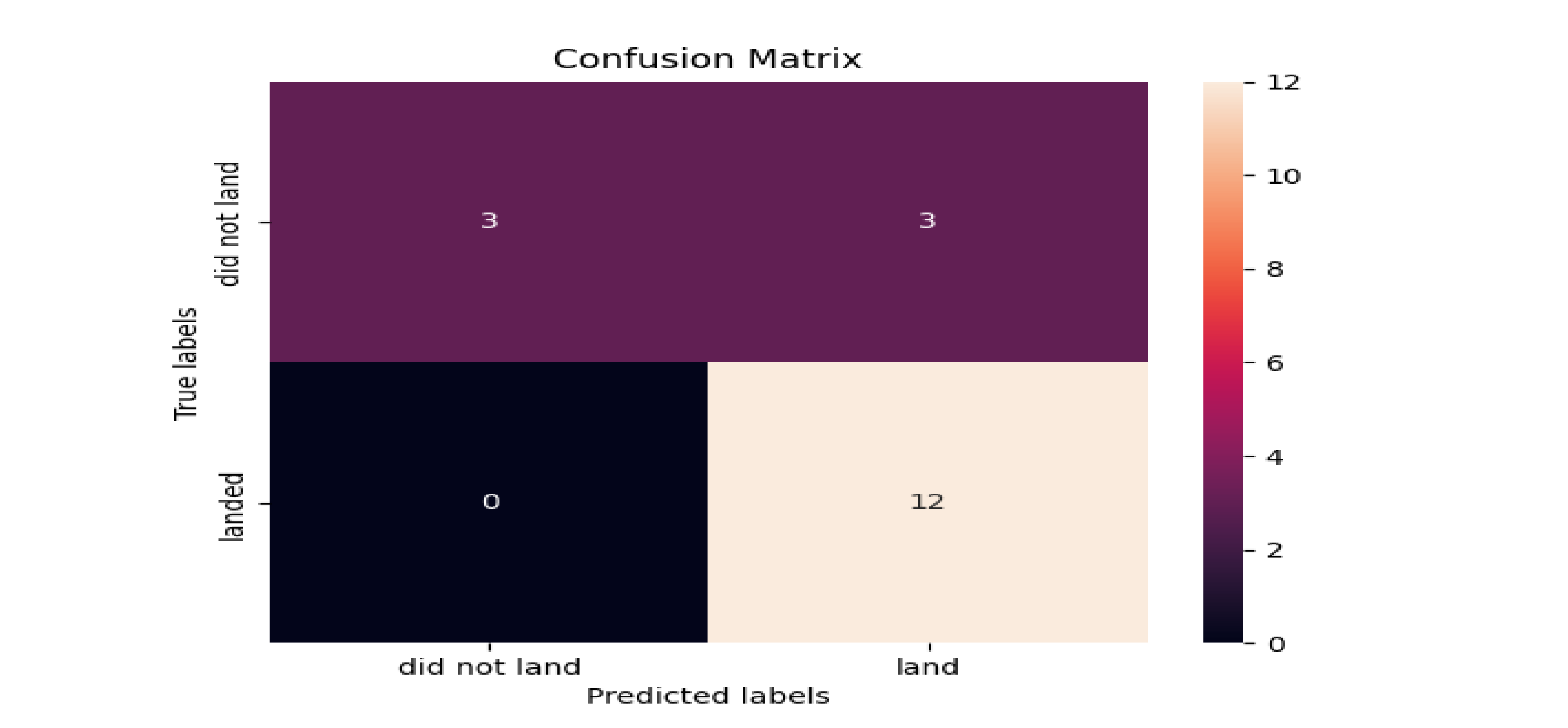


```
[32]:
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.9017857142857144
Best params is : {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_spli
t': 2, 'splitter': 'best'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

Our conclusions are as follows:

- Launch success rate started to increase in 2013 till 2020.
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Appendix

- https://github.com/AgbaSparks/falcon9/blob/main/08_SpaceX_Predictive_Analytics.ipynb
- https://github.com/AgbaSparks/falcon9/blob/main/DashBoarding_with_Folium.ipynb
- https://github.com/AgbaSparks/falcon9/blob/main/Data_Wrangling_Falcon9.ipynb
- https://github.com/AgbaSparks/falcon9/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb
- https://github.com/AgbaSparks/falcon9/blob/main/spacex_dash_app.py
- https://github.com/AgbaSparks/falcon9/blob/main/EDA_with_Visualization.ipynb

Thank you!