IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jeff P

1/21/2024

Outline

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

Executive Summary

Summary of methodologies

The following methods were used to extract, consolidate, analyze and provide predictive analytics for this project.

- Data Extraction from the SpaceX API
- Data Extraction utilizing web page scraping from Wikipedia
- Data Wrangling and cleaning
- Data Analysis and Understanding with SQL
- Data Visualization with Folium and Dash
- Predictive Analytics with Machine Learning

Executive Summary

Summary of all results

- 67% overall success rate for first stage landings
- The success rate continues to improve over time
- The GTO orbit had the lowest success rate
- Payloadmass, orbit, launch site are some of the key features to predicting success
- We can predict future landing outcomes with 80-90 percent accuracy

Introduction

Project background and context

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 project looks to consolidate and analyze data from APIs and Websites to understand what factors correlate wit the outcome of the SpaceX rocket launches. Once the key features are determined and engineered, accurate predictions can be made about future ricket launches.

Problems and questions we are looking to answer

- What factors are important in a successful launch
- What machine learning models can help us better predict launch outcomes
- What is the success rate for future launches

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - Data was extracted from the SpaceX API and the Falcon9 Wikipedia page and saved to CSV files for further exploration
- **Perform data wrangling**
  - Datasets were processed using Python/Pandas data frame to clean and transform missing data and convert categorical variables to integer values using one hot encoding
- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - Data was loaded to a database for further analysis using SQL to better understand landing outcomes and get insights on potential input variables
- **Perform interactive visual analytics using Folium and Plotly Dash**
  - Visualization tools were used to better understand patterns and trends in the rocket launches
- **Perform predictive analysis using classification models**
  - 4 different models were utilized to find the most accurate tool to predict future landing outcomes

Data Collection

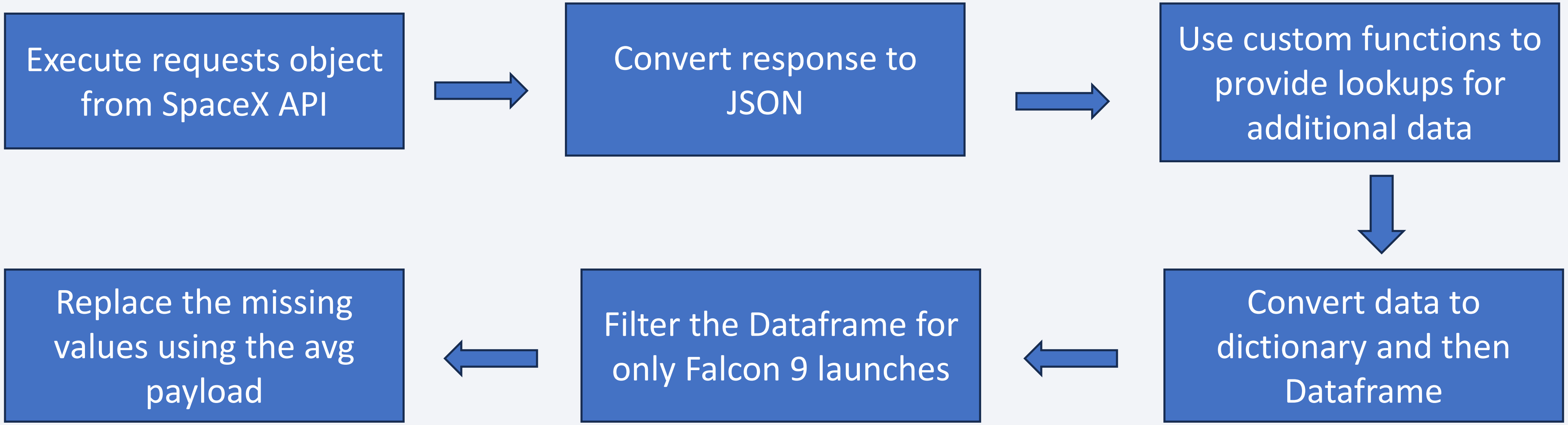
SpaceX API

The API extraction utilizes API requests to consolidate and join multiple datasets together by loading pandas data frames for cleansing and transformations

Wikipedia Web Scraping

The Wikipedia data collection utilizes the Beautiful Soup library to parse the site html and load a data frame for analysis

Data Collection – SpaceX API Workflow



Execute requests object from SpaceX API

Convert response to JSON

Use custom functions to provide lookups for additional data

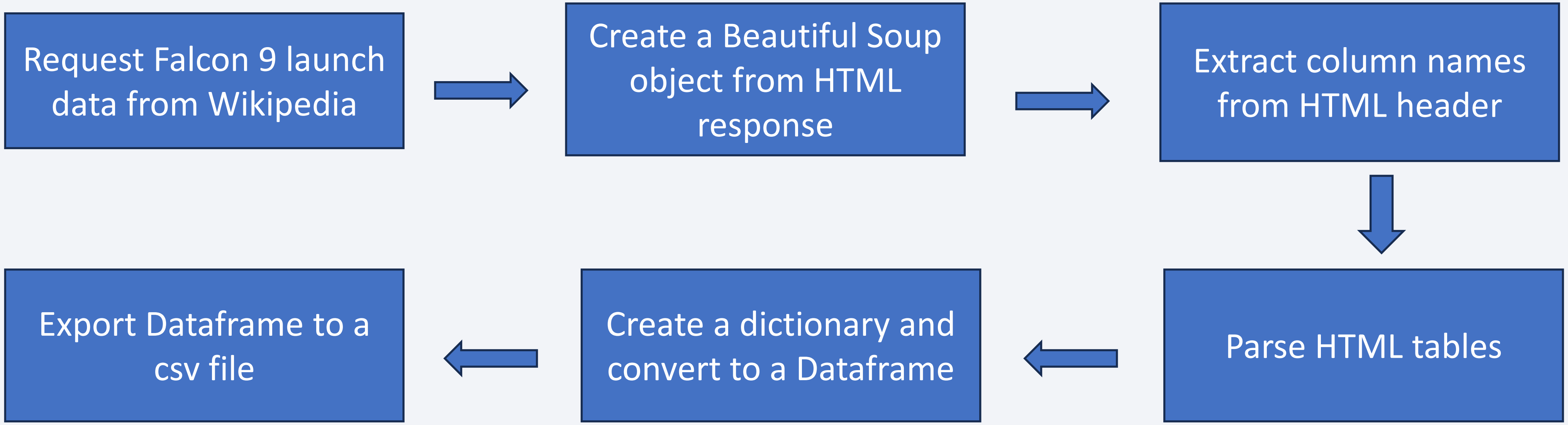
Convert data to dictionary and then Dataframe

Filter the Dataframe for only Falcon 9 launches

Replace the missing values using the avg payload

GitHub URL: [python/jupyter-labs-spacex-data-collection-api (1).ipynb at main · jpop78/python (github.com)]

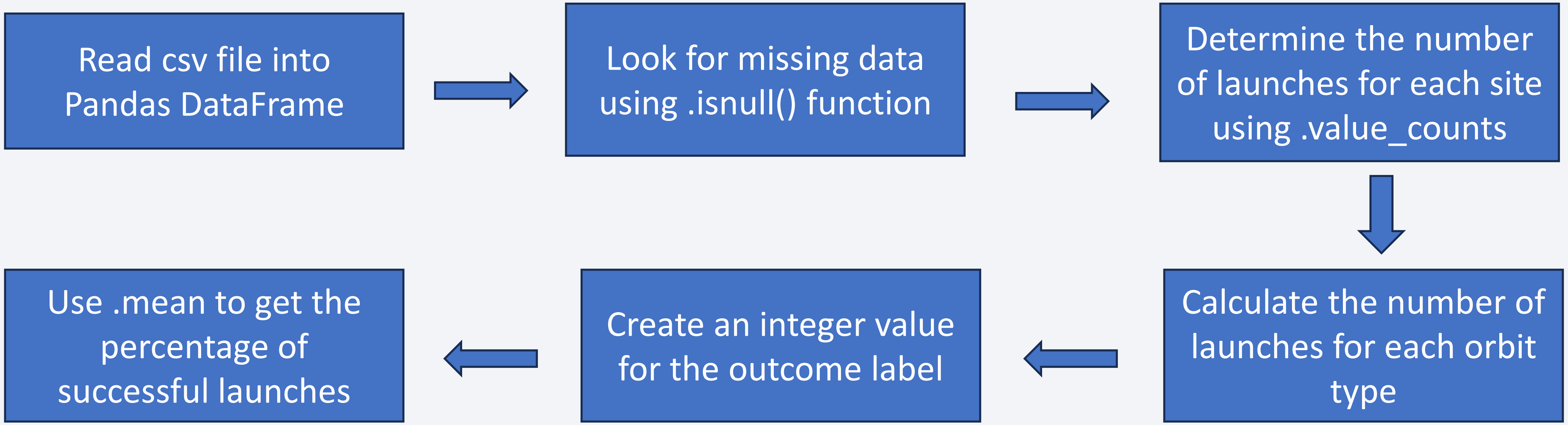
Data Collection – Scraping Workflow


Request Falcon 9 launch data from Wikipedia
Create a Beautiful Soup object from HTML response
Extract column names from HTML header
Parse HTML tables
Create a dictionary and convert to a Dataframe
Export Dataframe to a csv file

GitHub URL: python/jupyter-labs-webscraping.ipynb at main · jpop78/python (github.com)

Data Wrangling

Data Wrangling utilized python to analyze and transform the data using a Pandas Dataframe



Read csv file into Pandas DataFrame

Look for missing data using .isnull() function

Determine the number of launches for each site using .value_counts

Calculate the number of launches for each orbit type

Create an integer value for the outcome label

Use .mean to get the percentage of successful launches

GitHub URL: python/labs-jupyter-spacex-Data wrangling.ipynb at main · jpop78/python (github.com)

EDA with Data Visualization

Scatter Plots - Scatter plots were utilized to see which data points are potentially more correlated to specific landing outcomes

- Flight Number vs. Payload
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Payload vs. Orbit Type

Bar Charts – Used for comparing differences in categories

- Success Rate vs. Orbit

Line Chart – For understanding time series trends

- Success Yearly Trend

GitHub URL: python/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb at main · jpop78/python (github.com)

EDA with SQL

The following results were reported utilizing SQL after importing launch data to a database

- Names of unique launch sites
- A sample of records where launch site begins with 'CCA'
- The total payload mass carried by boosters launched by NASA
- The average payload mass carried by booster version F9 v1.1
- Date of first successful landing on ground pad
- Names of boosters which had success landing on drone ship with mass greater than 4,000 but less than 6,000
- The total number of successful and failed missions
- Names of booster versions which have carried the max payload
- Failed landing outcomes on drone ship, their booster version and launch site for the months in the year 2015
- The count of landing outcomes between 2010-06-04 and 2017-03-20 order in descending order

GitHub URL: python/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · jpop78/python (github.com)

Build an Interactive Map with Folium

- All launch sites were marked and added to the map
- Added a blue circle to indicate NASA Johnson Space Center
- Indicted the outcomes for each site, green for success, red for failed
- We calculated the distances between a launch site to its proximities.

Answered:

- Are launch sites near railways, highways or coastlines.
- Do launch sites keep certain distance away from cities.

GitHub URL: python/lab_jupyter_launch_site_location.jupyterlite.ipynb at main · jpop78/python (github.com)

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the launch success by launch site
- We a plotted scatter chart showing the relationship with Outcome and Payload Mass per booster version.

GitHub URL: python/spacex_dash_app.py at main · jpop78/python (github.com)

Predictive Analysis (Classification)

1- The launch data was first loaded to 2 data frames to get ready for modeling

2- Next, we converted the outcome column into a numpy array stored as a variable Y and scaled the input fields to the same scale

3- The train_test_split function was used to split that data into a training data set and a test set for validation

4-For each of the following classification models we used the GridSearchCV object to test the models with different parameter values to achieve the best accuracy.

Logistic Regression:

Support Vector Machine

Decision Tree

K-Nearest Neighbor

Accuracy was used as the metric for validating our models against the test data set.

We found the best performing classification model to be the decision tree.

GitHub URL: [python/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · jpop78/python (github.com)]()

Results

Exploratory data analysis results

- Landing Outcomes have continued to improve over time
- There are strong correlations to the landing outcomes based on landing sites, orbit types, and payload mass

Interactive analytics

- The launch sites tend to be near the equator, close to the coast, and far enough from highways to prevent launch damage

Predictive analysis results

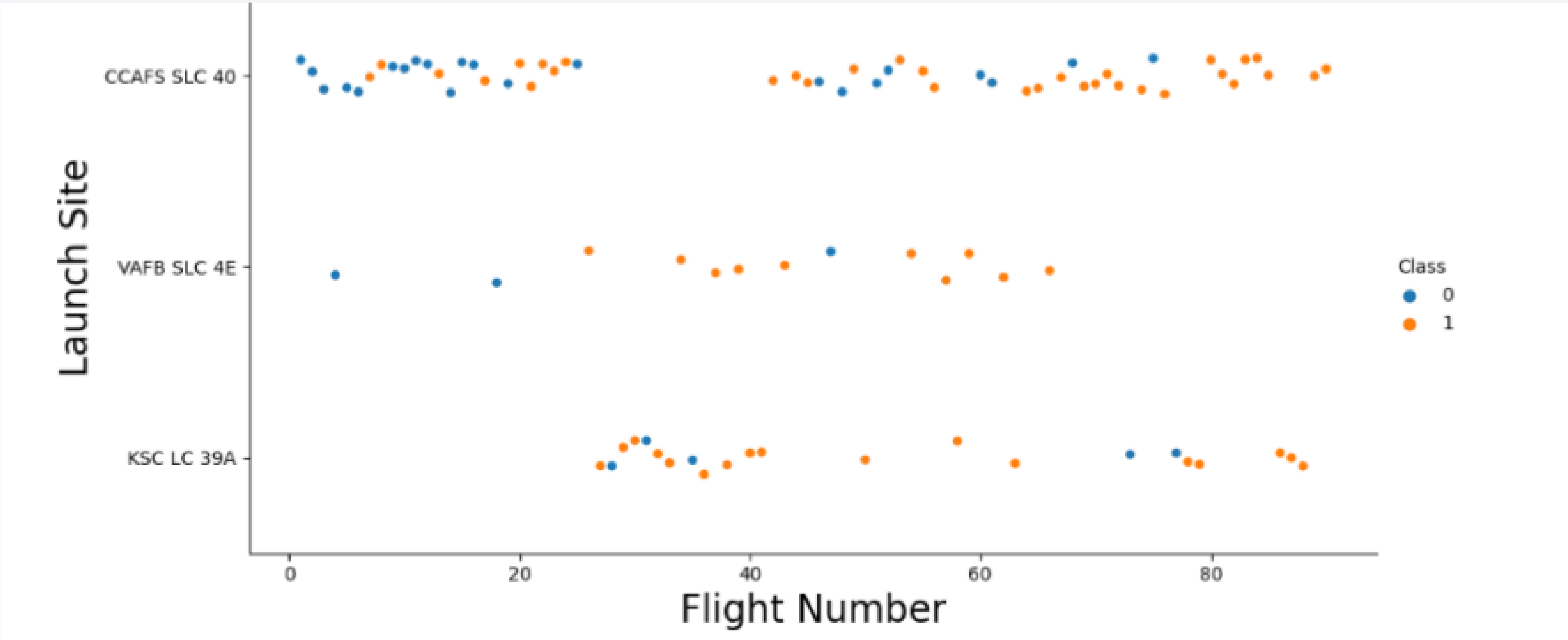
- The decision tree model produced 94% accuracy for predicting landing outcomes on test data

Section 2

Insights drawn from EDA

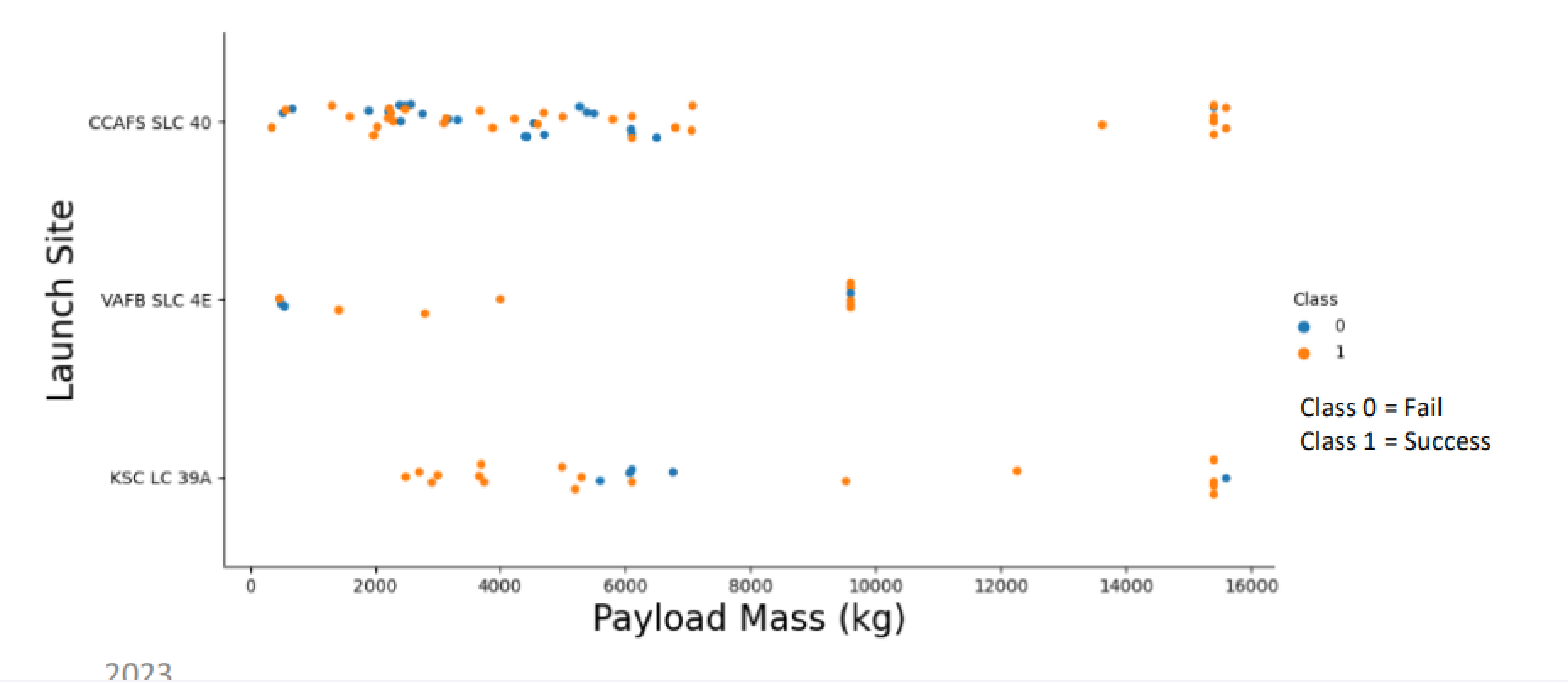
Flight Number vs. Launch Site

- The plot shows that later flights had a higher success rate based on a higher rate of orange plots compared to blue
- CCAFS SLC 40 had the highest number of launches by far and had a perfect success rate after the 80th launch



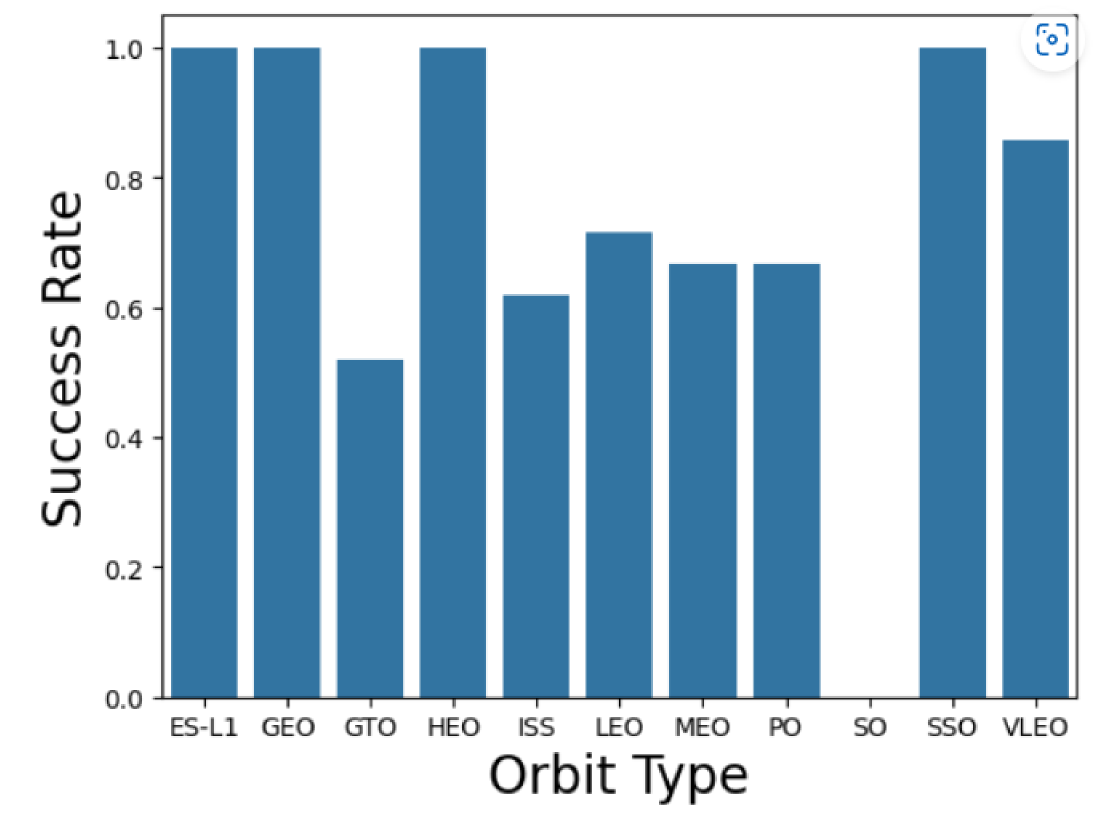
Payload vs. Launch Site

- Payloads higher than 8000kg have a very high success rate
- VAFB SKC 4E has not launched payloads > 10,000 kg
- Most failed launches occurred at a payload between 4,000 and 8,000 kg across all sites



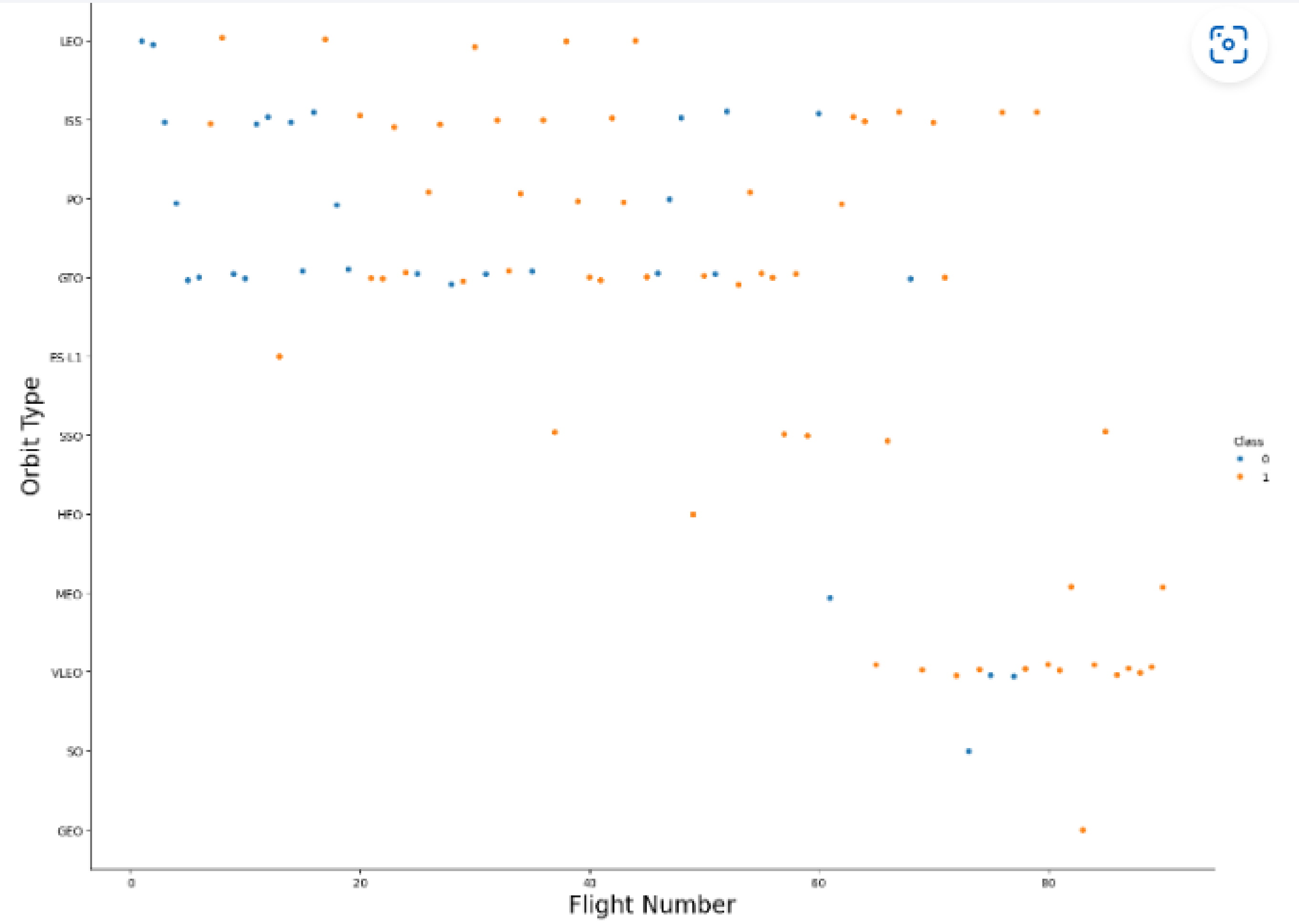
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO have a 100% success rate
- SO has a 0% success rate and the remaining are between 50-85%



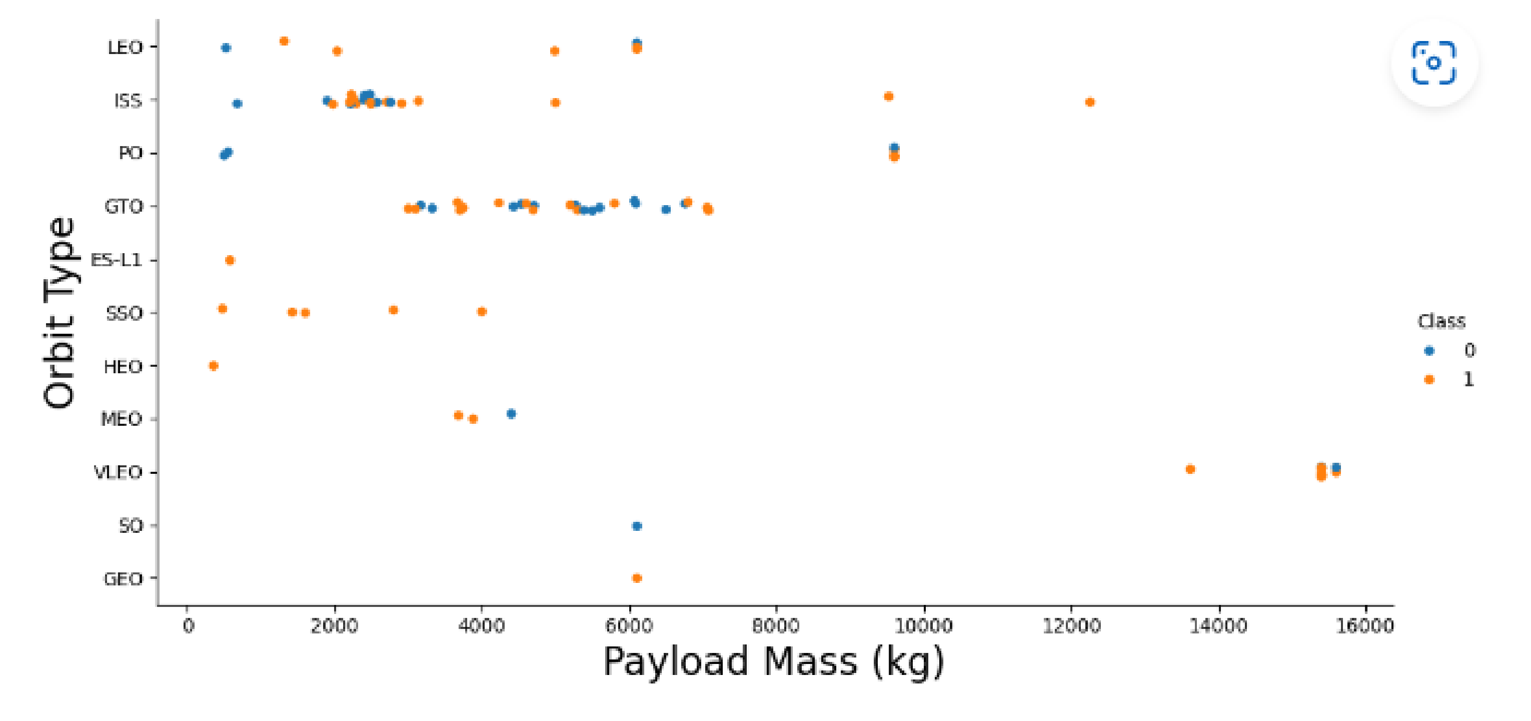
Flight Number vs. Orbit Type

- LEO orbit success rate improves over time
- The success rate for the GTO orbit does not change over time
- The VLEO orbit has a high success rate but did not start launching until after 60 flights



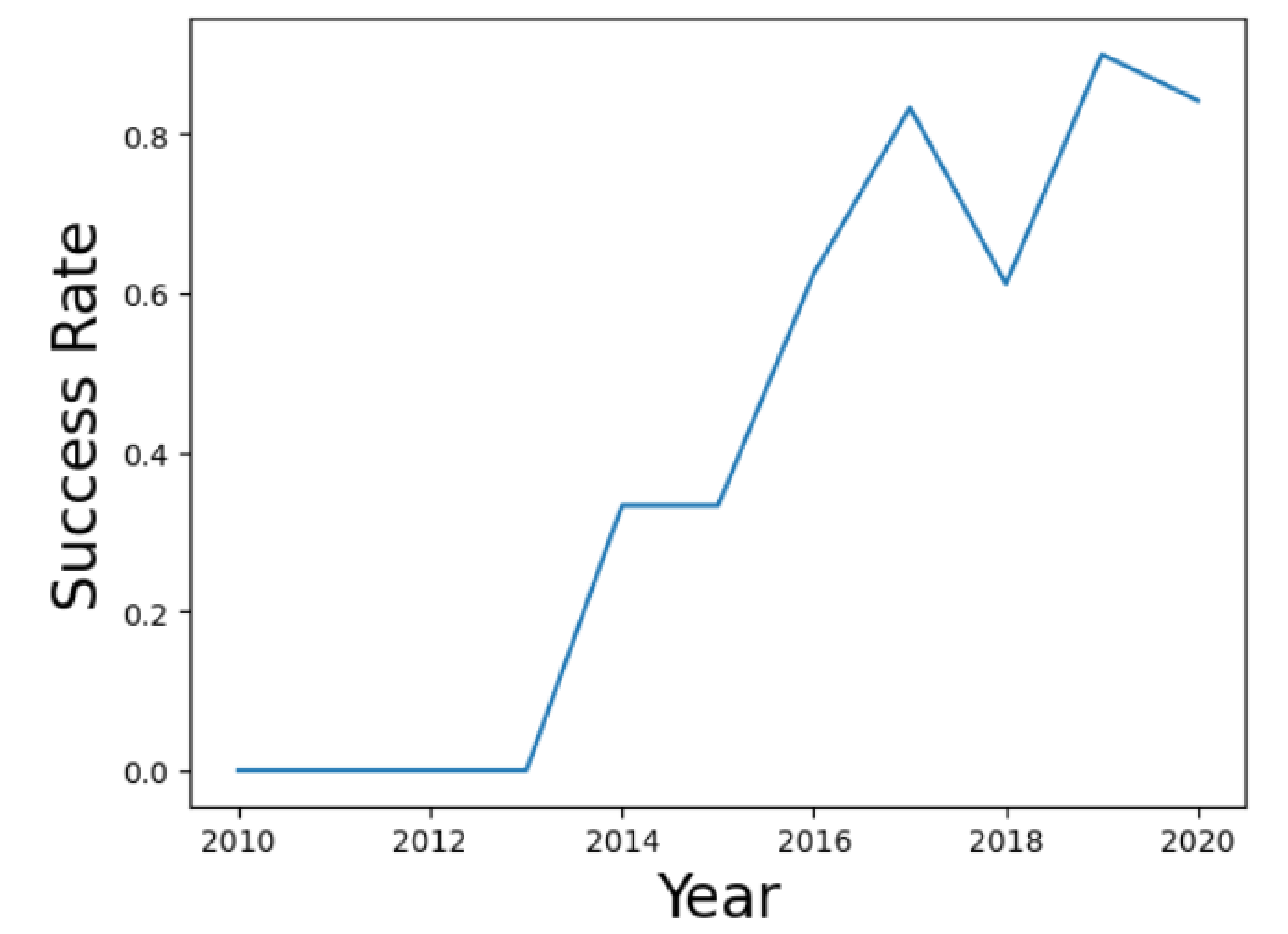
Payload vs. Orbit Type

- Polar, LEO and ISS orbits have a high success rate for higher payloads
- The GTO orbit has roughly a 50% success rate overall



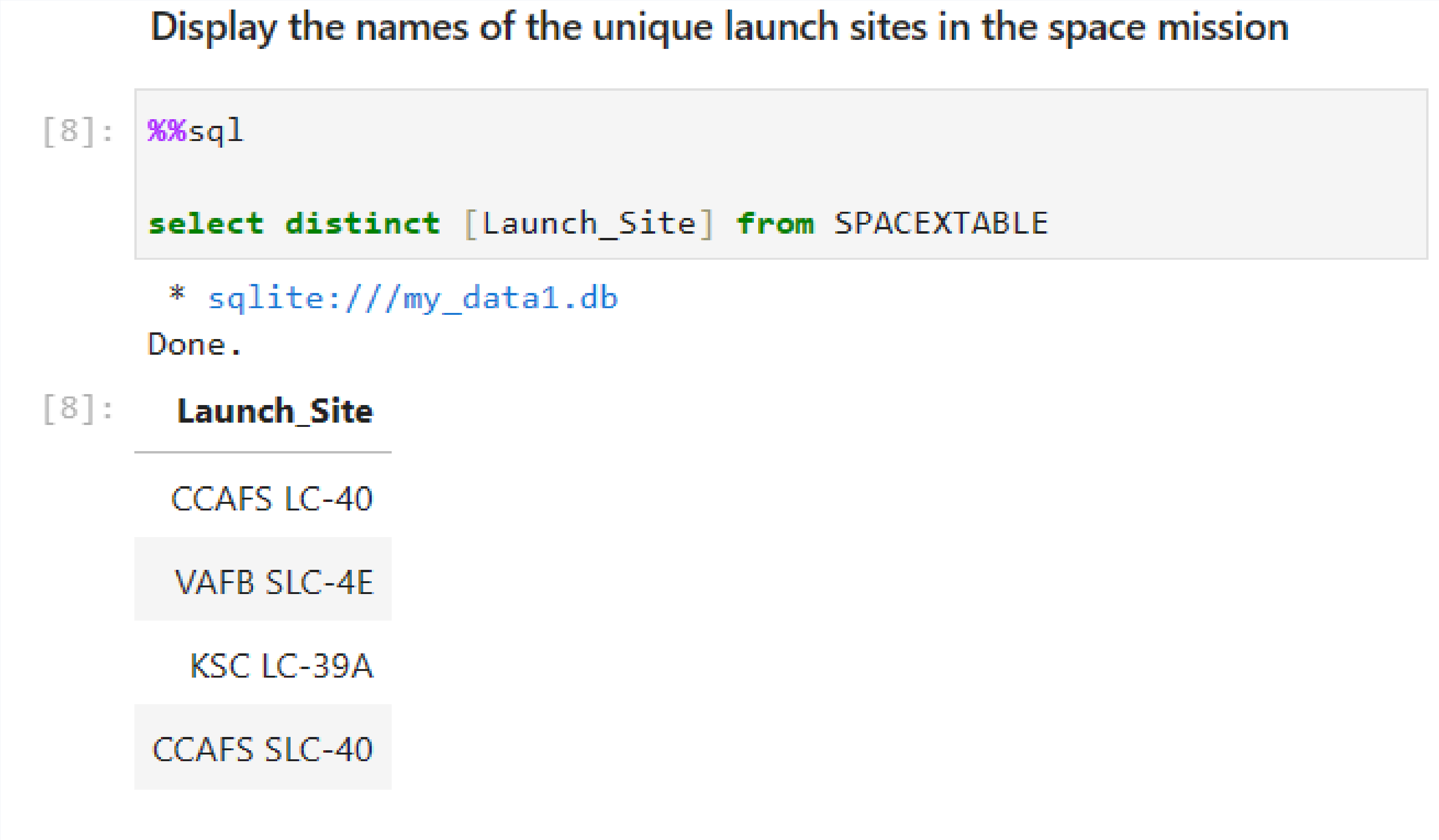
Launch Success Yearly Trend

- The time series graph shows that overall, the success rate is improving over time with a significant increase from 2015 to 2017



All Launch Site Names

- This query gets the distinct list of Launch Sites



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

[8]:
```
%%sql

select distinct [Launch_Site] from SPACEXTABLE
```

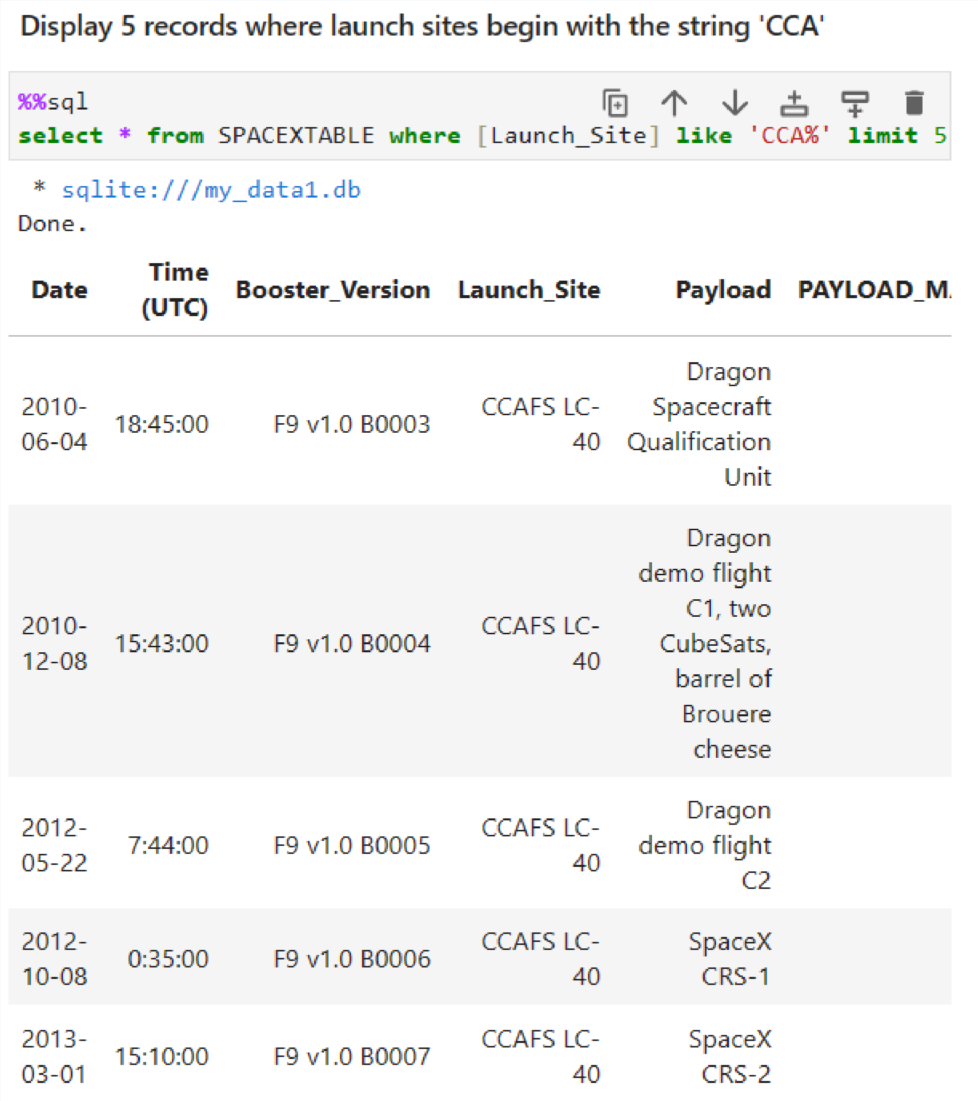
```
 * sqlite:///my_data1.db
Done.
```

[8]:

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

Launch Sites Begining with 'CCA'

- This query limits the results to 5 and displays the launch sites that have the string ‘CCA’ in the name.



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

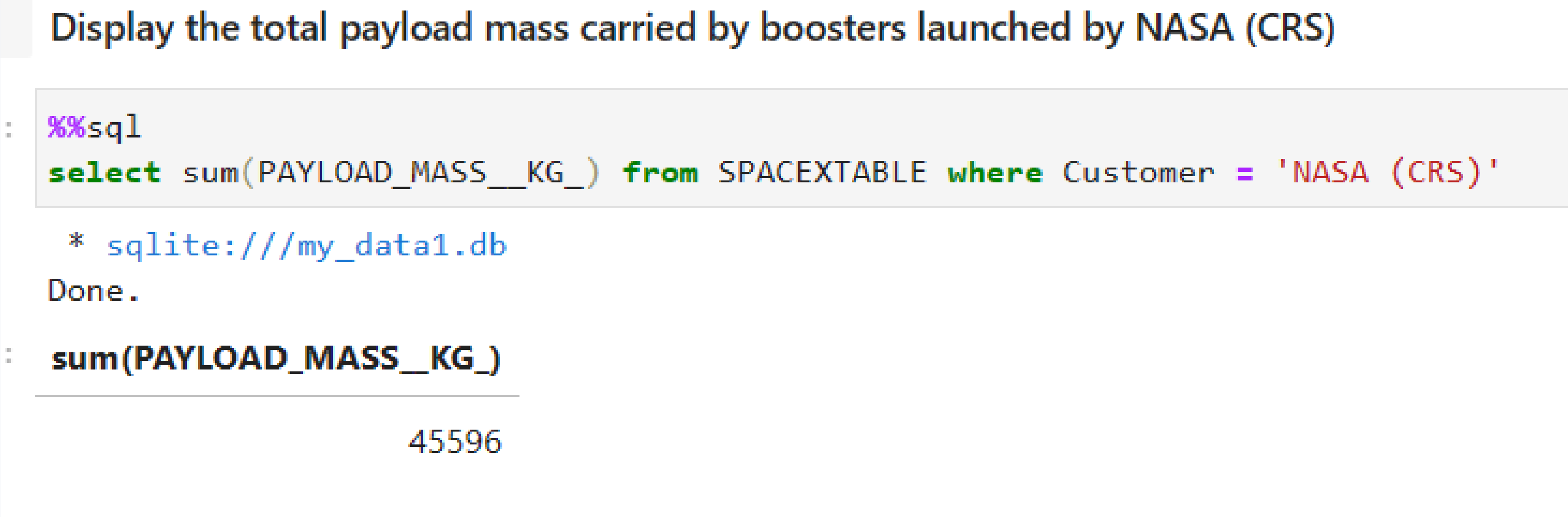
```
%%sql
select * from SPACEXTABLE where [Launch_Site] like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_M
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
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

- This query gets the sum of the pay load mass for all launches where the customer was NASA (CRS)



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

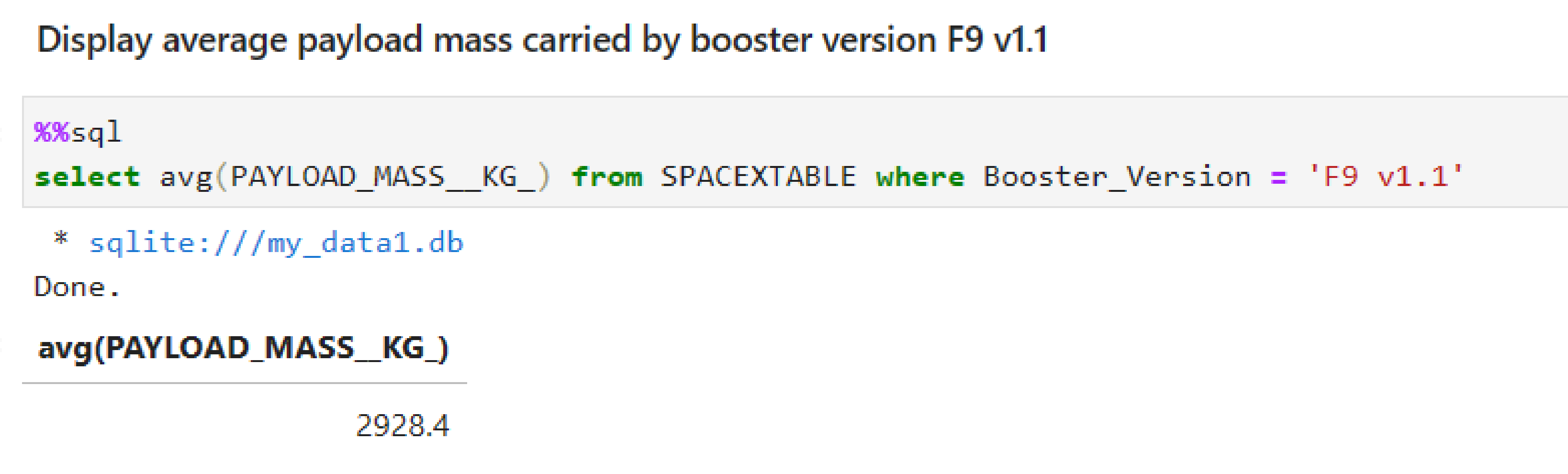
```
%%sql
select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- This query gets the average payload mass where the booster version equals ‘F9 v1.1’



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

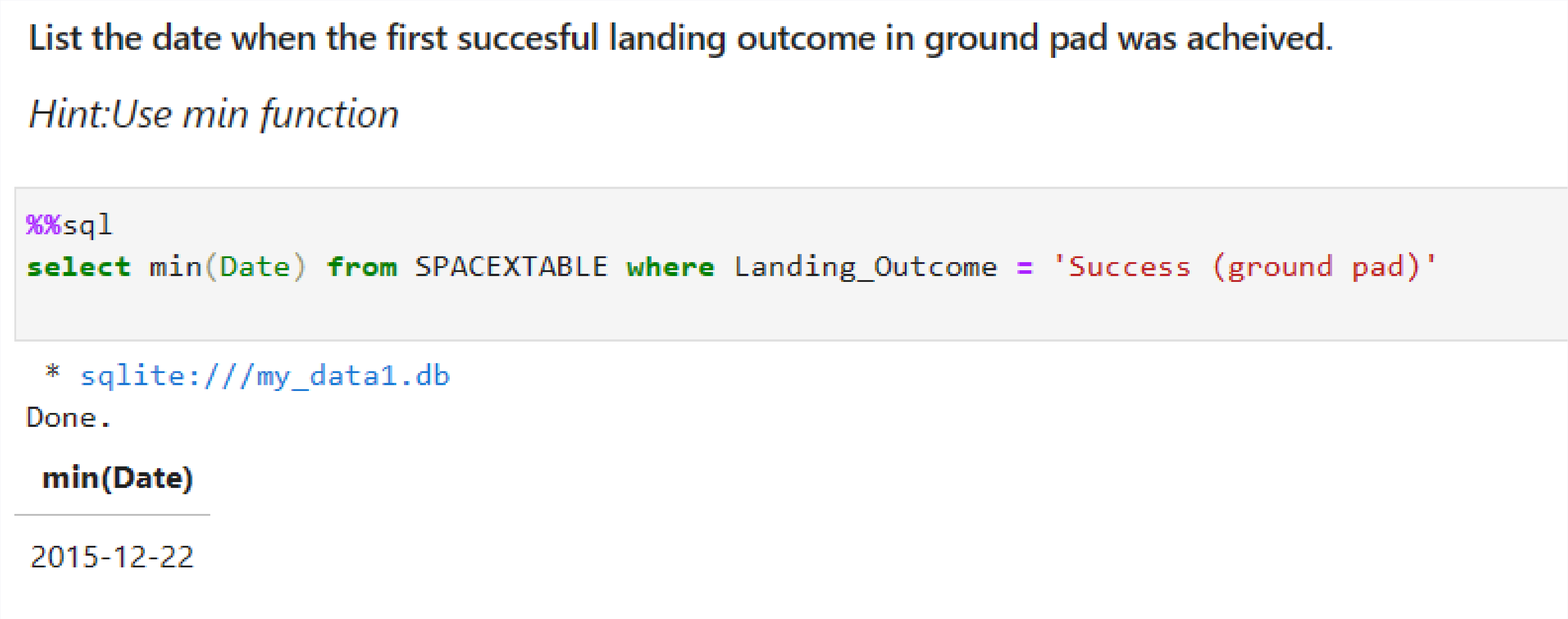
```
%%sql
select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

This query uses the min function to get the earliest date for a success outcome for ground pad



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

Hint:Use min function

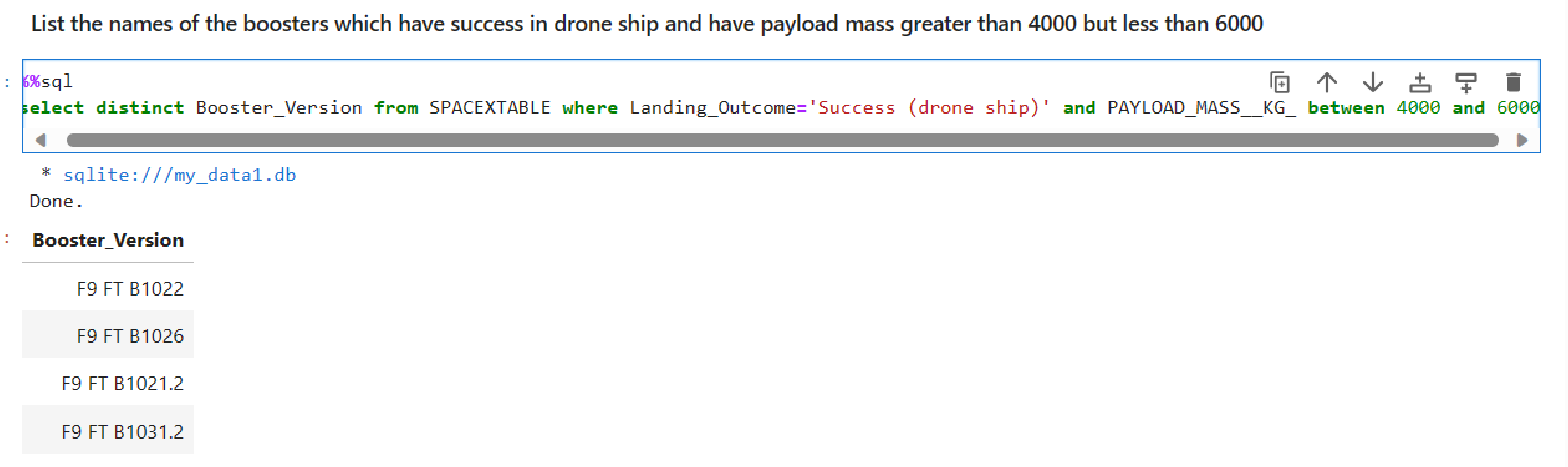
```
%%sql
select min(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

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

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query lists the booster versions with a success on drone ship and payload between 4000 and 6000 kg



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

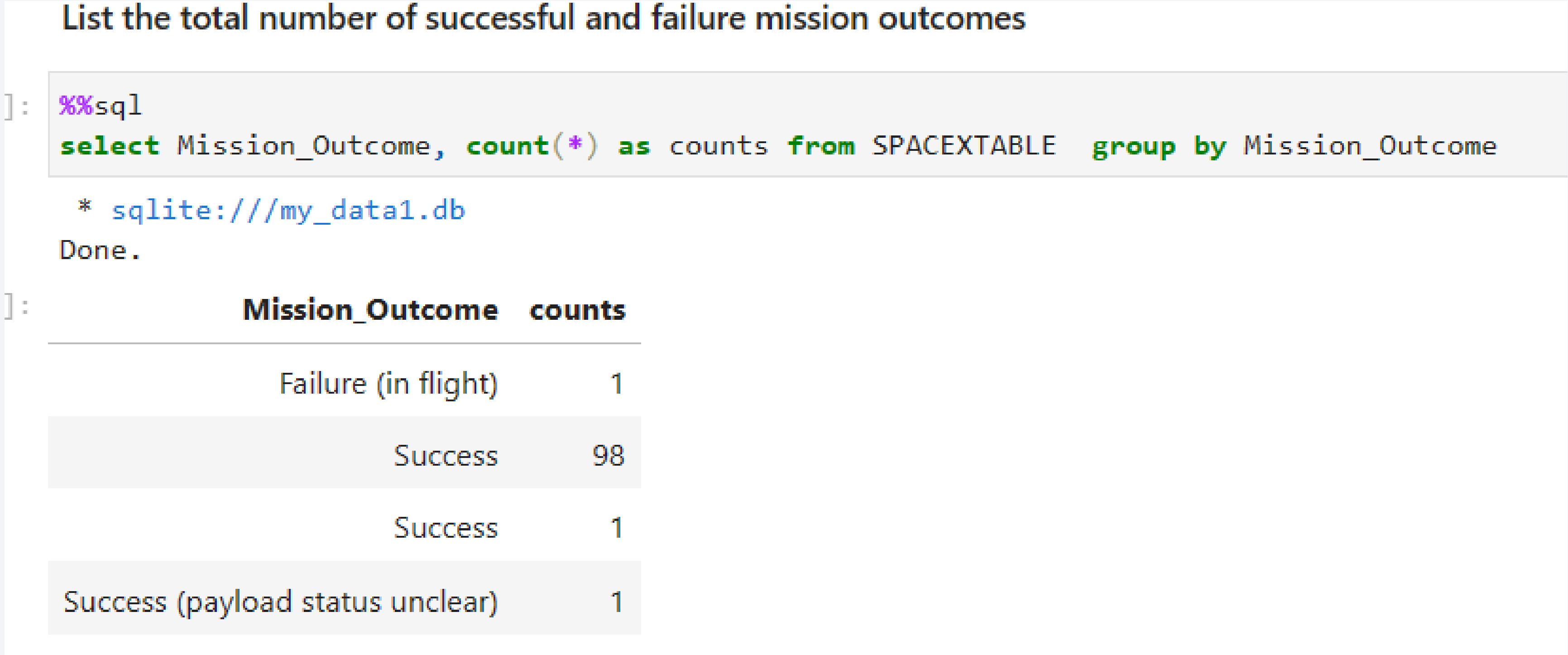
```
%%sql
select distinct Booster_Version from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

```
 * 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

- This query lists the counts of the outcomes by grouping by the mission_outcome column



List the total number of successful and failure mission outcomes

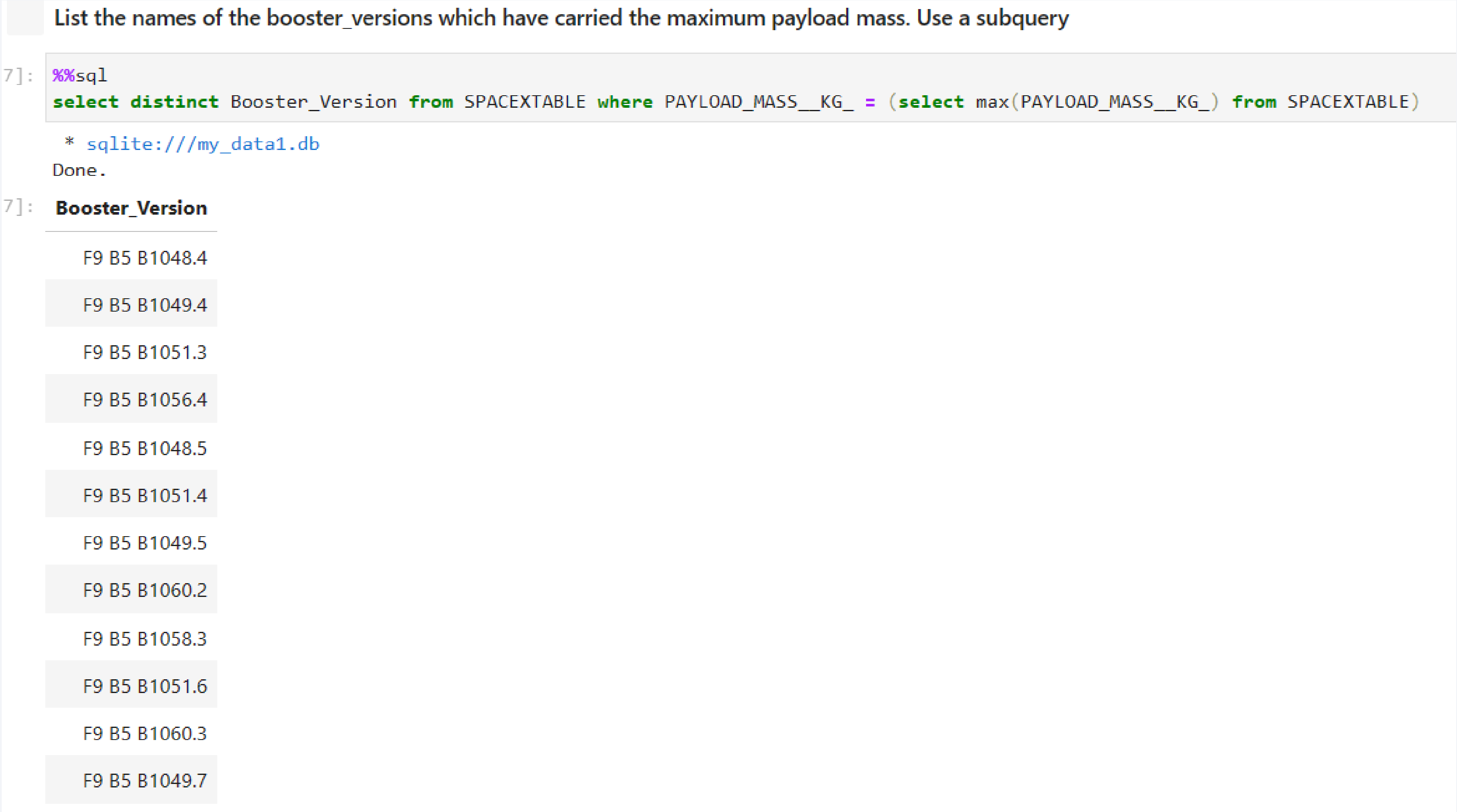
```
%%sql
select Mission_Outcome, count(*) as counts from SPACEXTABLE  group by Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

- This query gets the list of booster versions that had a payload mass equal to the max payload mass using a sub query to get the maximum payload.



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

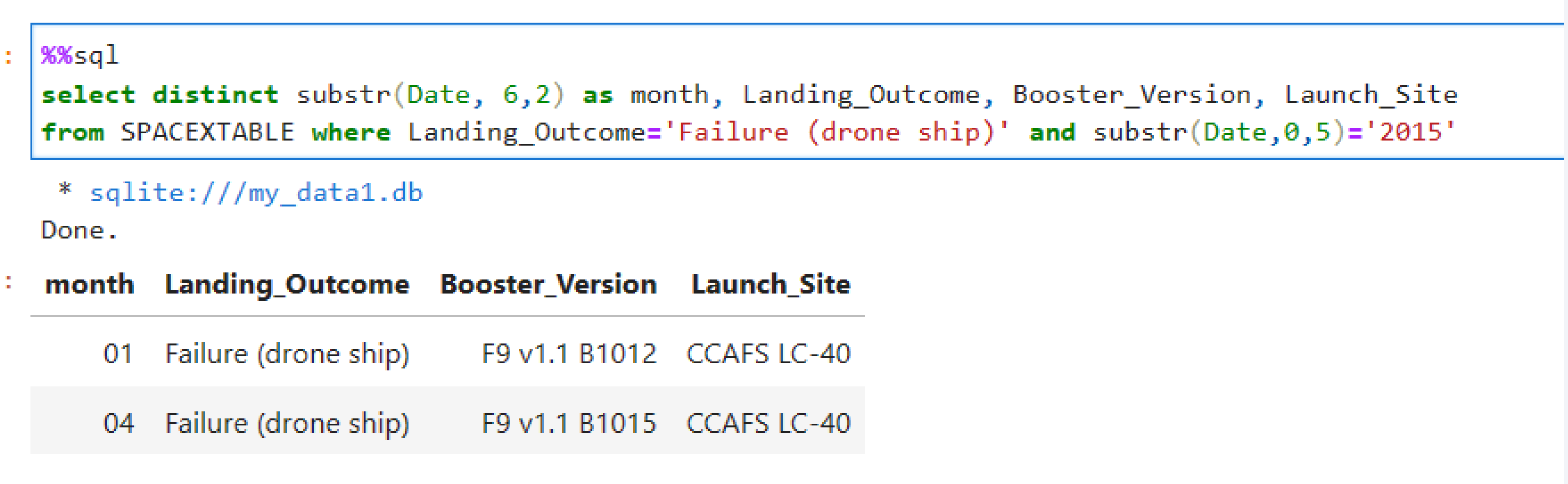
```
%%sql
select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

```
 * 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

- This query uses the substring function to get the month for failed drone ship landings in the year 2015



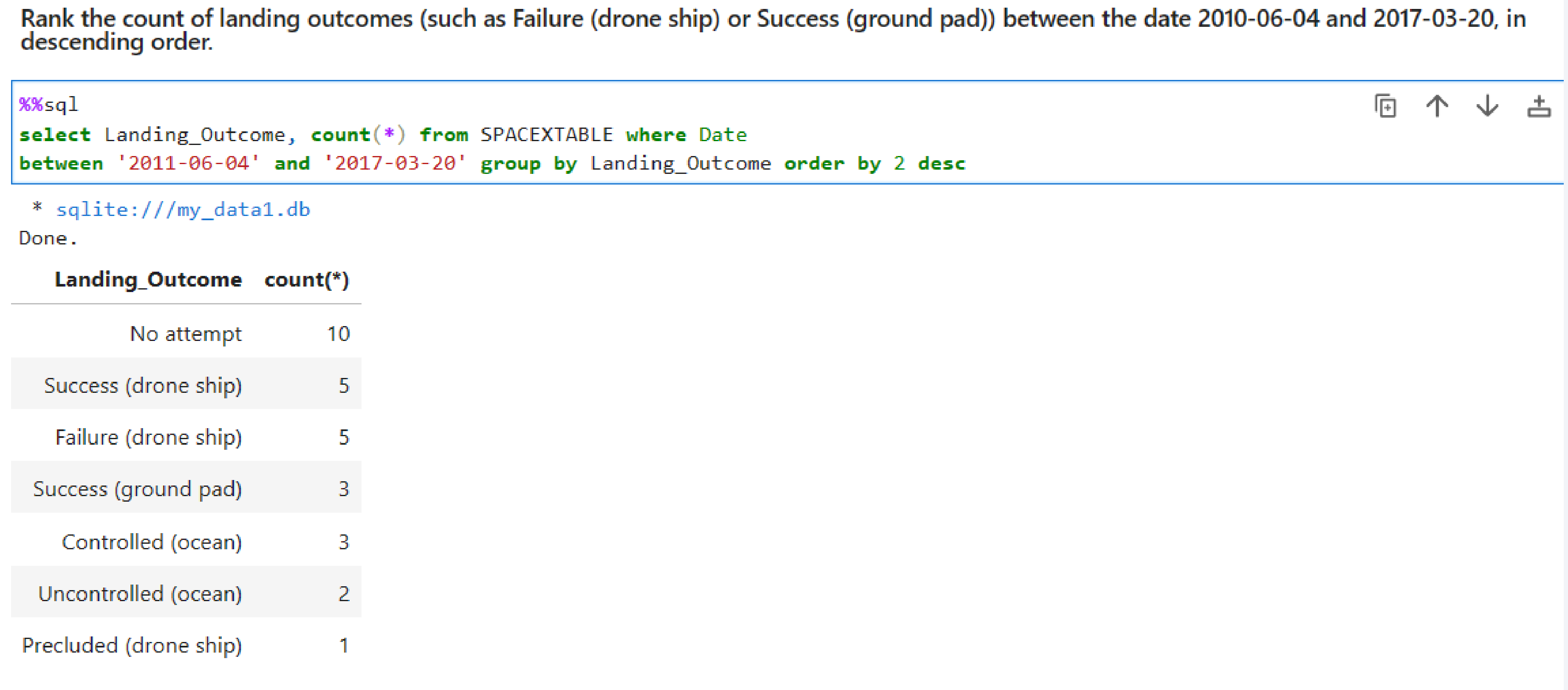
```
%%sql
select distinct substr(Date, 6,2) as month, Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTABLE where Landing_Outcome='Failure (drone ship)' and substr(Date,0,5)='2015'
```

```
 * 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

- This query gets the counts of launches per landing outcome for a date range and ranks them in descending order



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(*) from SPACEXTABLE where Date
between '2011-06-04' and '2017-03-20' group by Landing_Outcome order by 2 desc
```

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

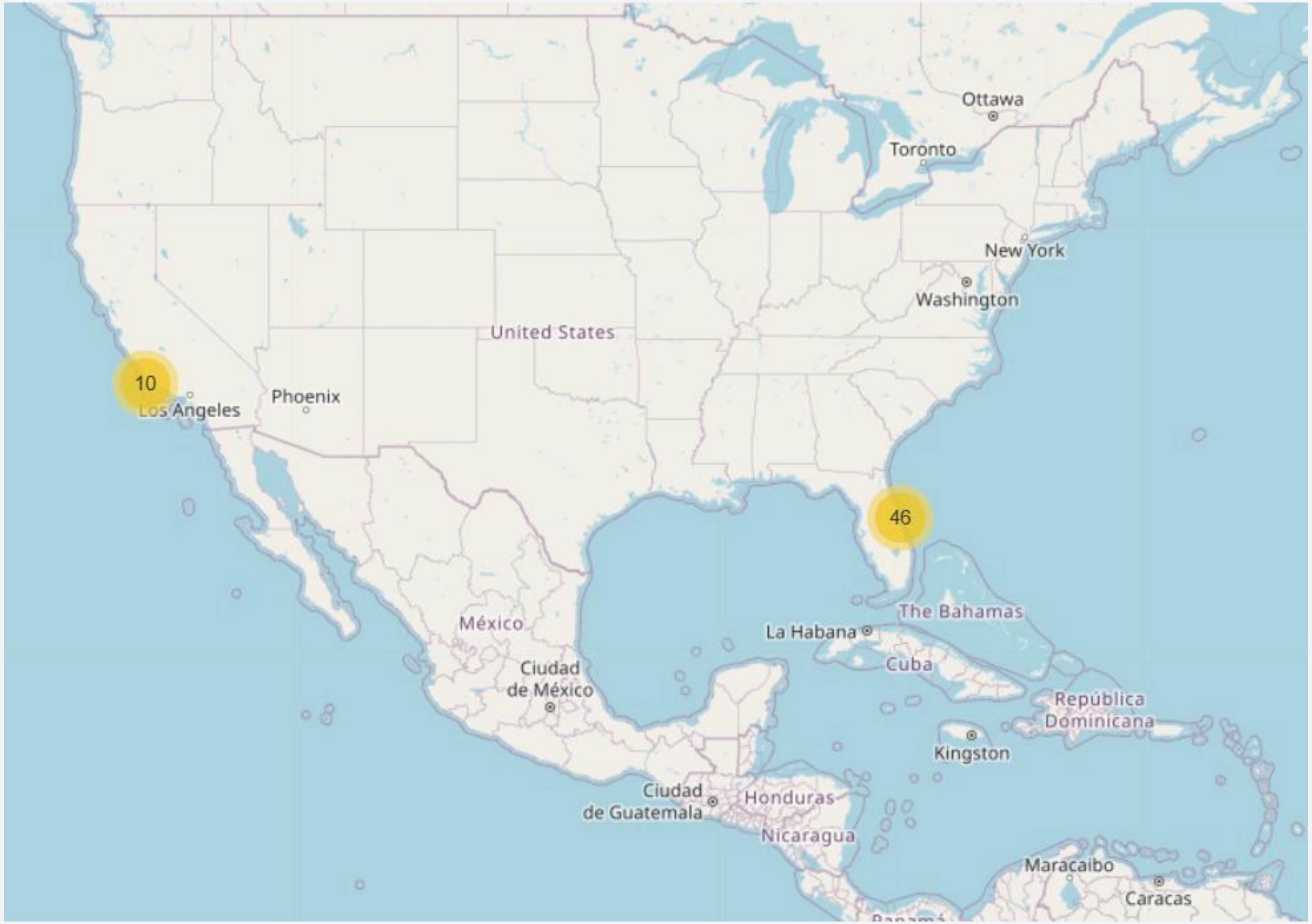
Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

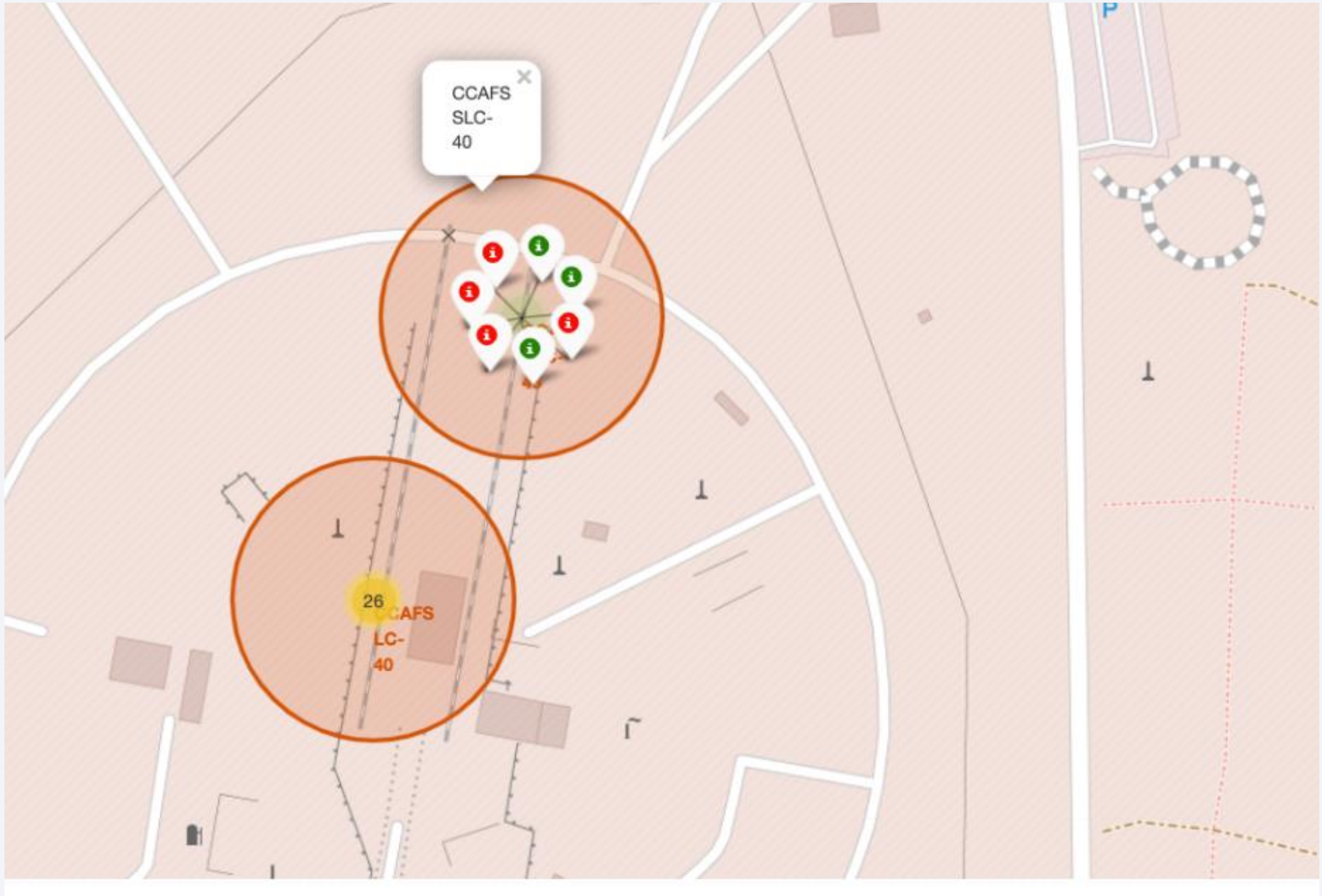
SpaceX Launch Sites

- The map shows the launch sites on the east and west coasts in the United States



SpaceX Launch Outcomes

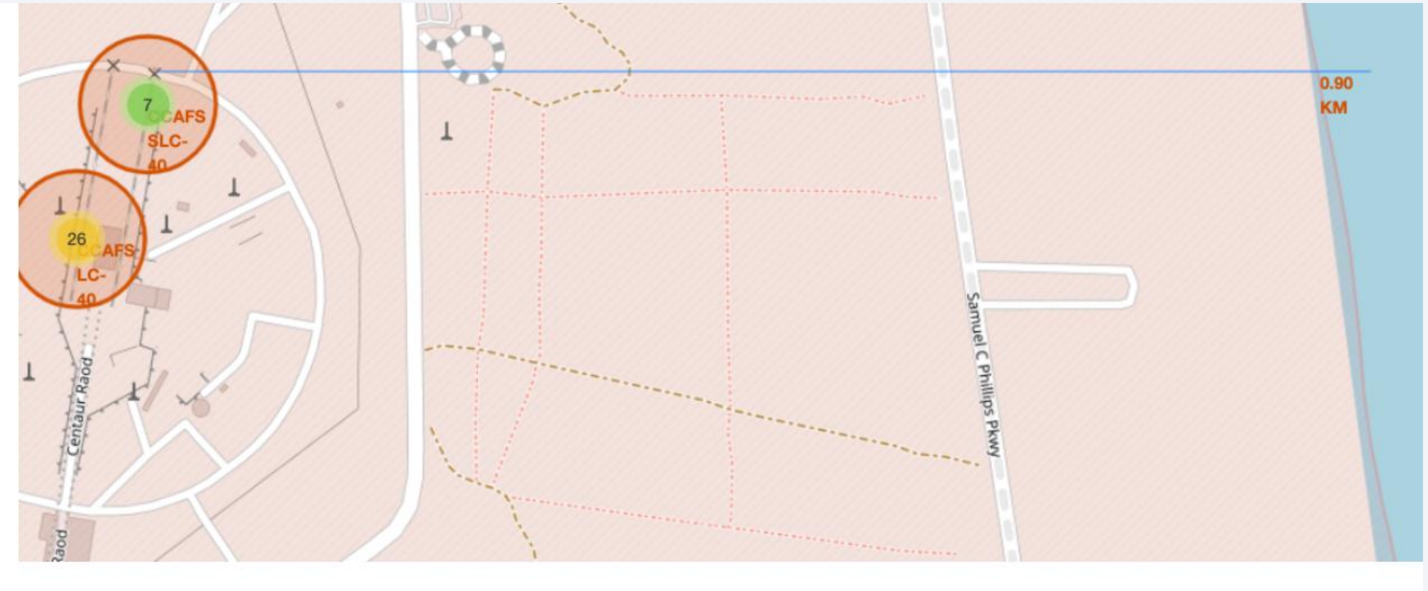
- As displayed, markers allow the viewer to see the success versus failure outcomes for a given launch site to get an idea of the success rate



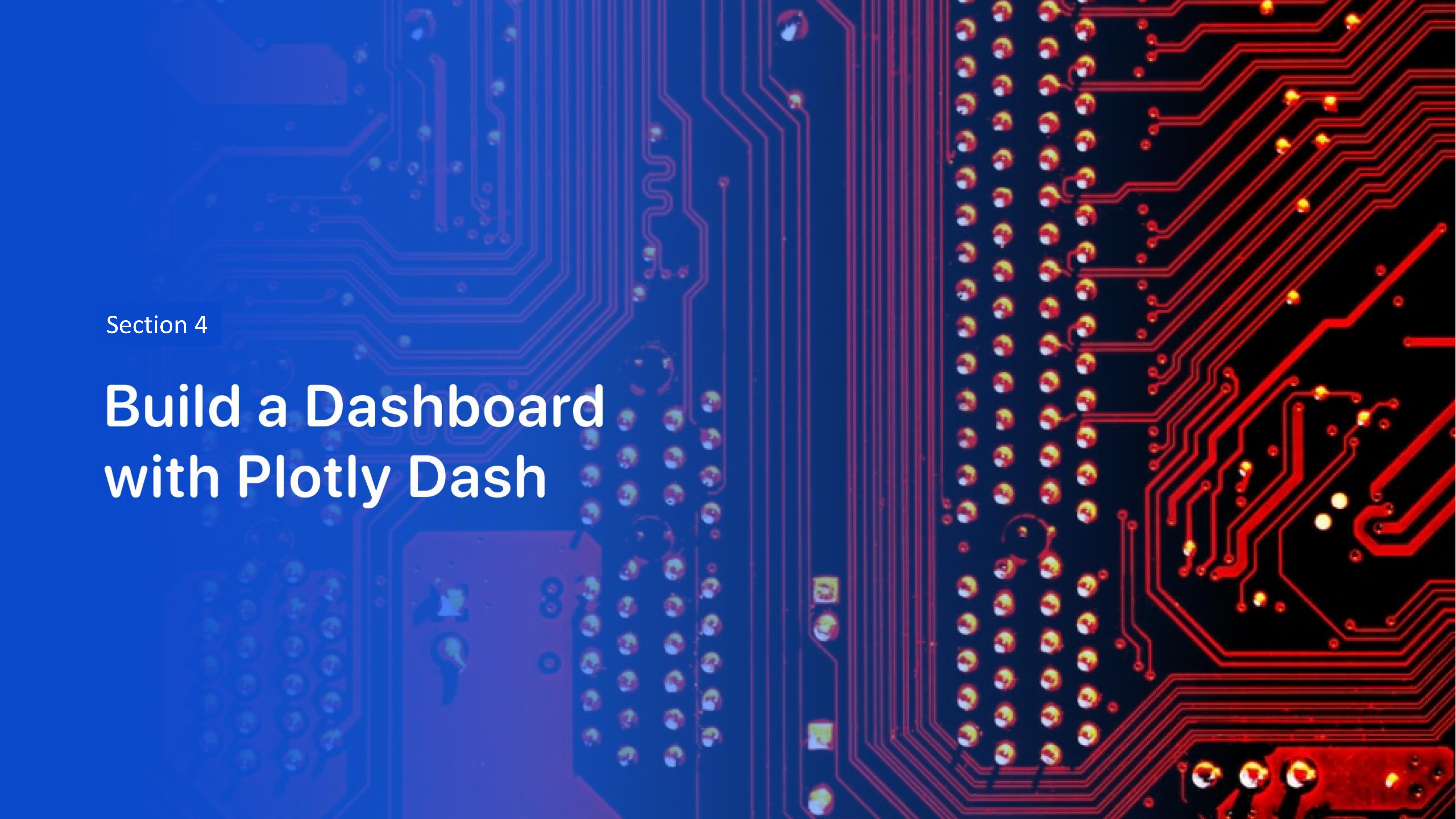
Distance To Proximities

NASA Railroad
Samuel C Phillips Pkwy
Melbourne

- The folium maps allow you to determine distances to railroads, highways, cities and coastlines to see they are far enough away to prevent damage during launches



center

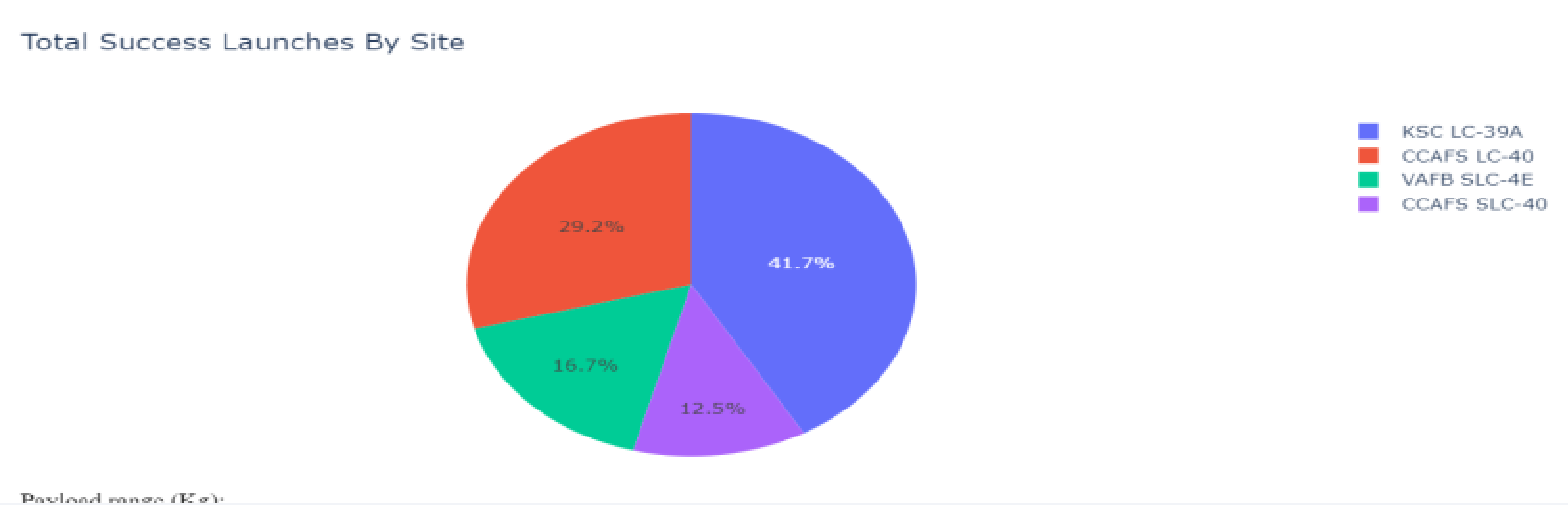


Section 4

Build a Dashboard with Plotly Dash

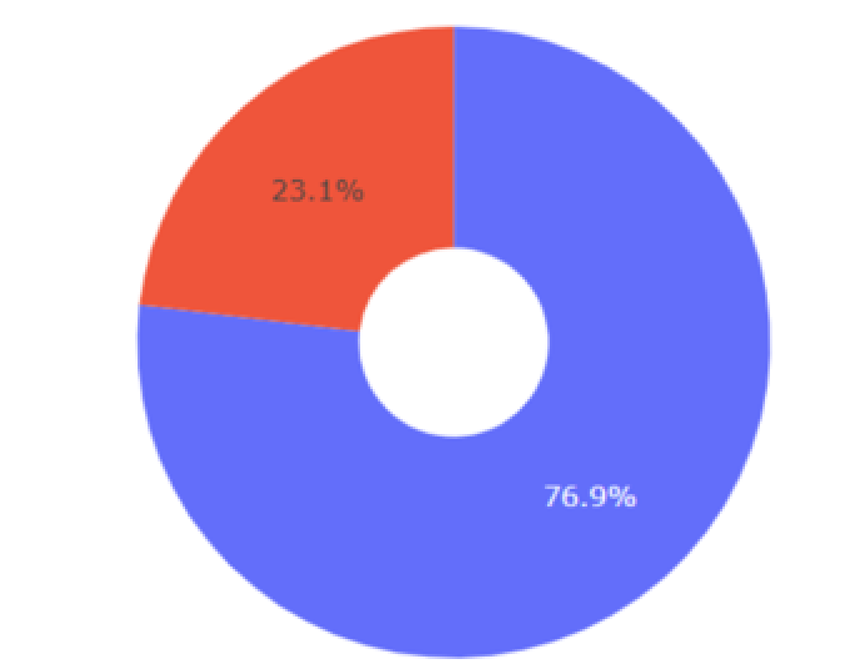
Launch Site Success

This pie chart shows how the different launch sites contributed to the overall successful outcomes



Most Successful Launch Site

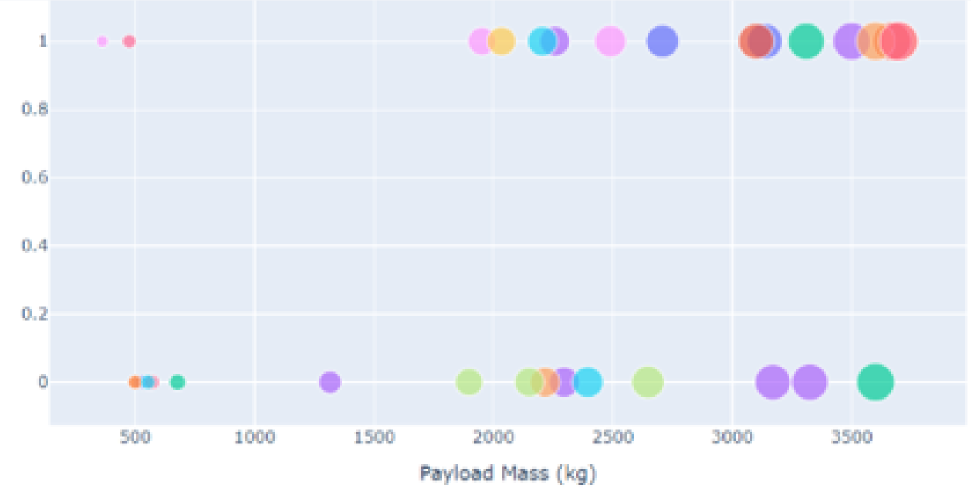
- KSC LC-39A was the most successful launch site at almost 77 percent success.



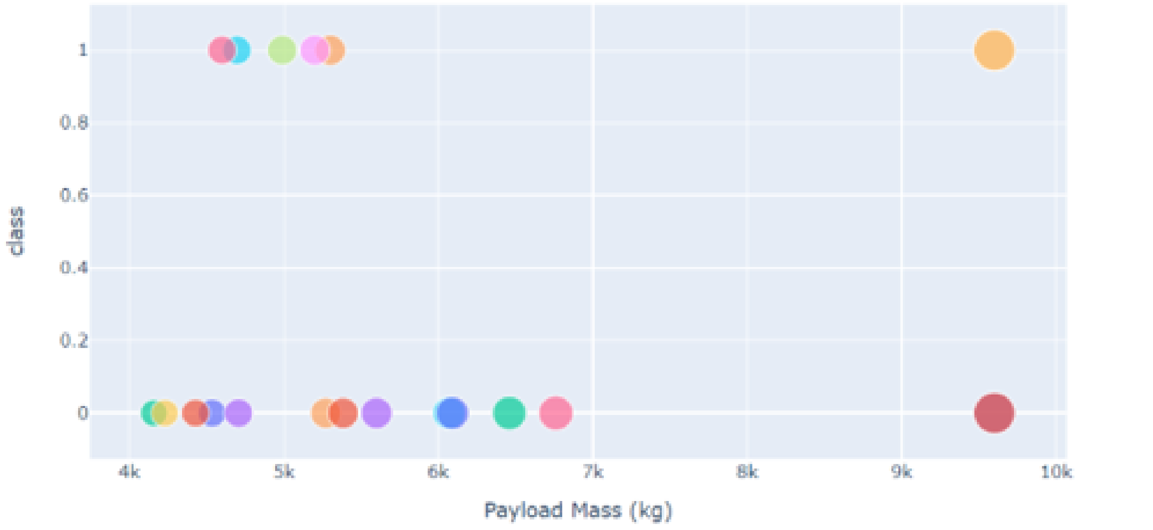
Payload vs Launch Outcome

As we can see the lighter payloads have a higher success rate overall

Light Payload



Heavy Payloads

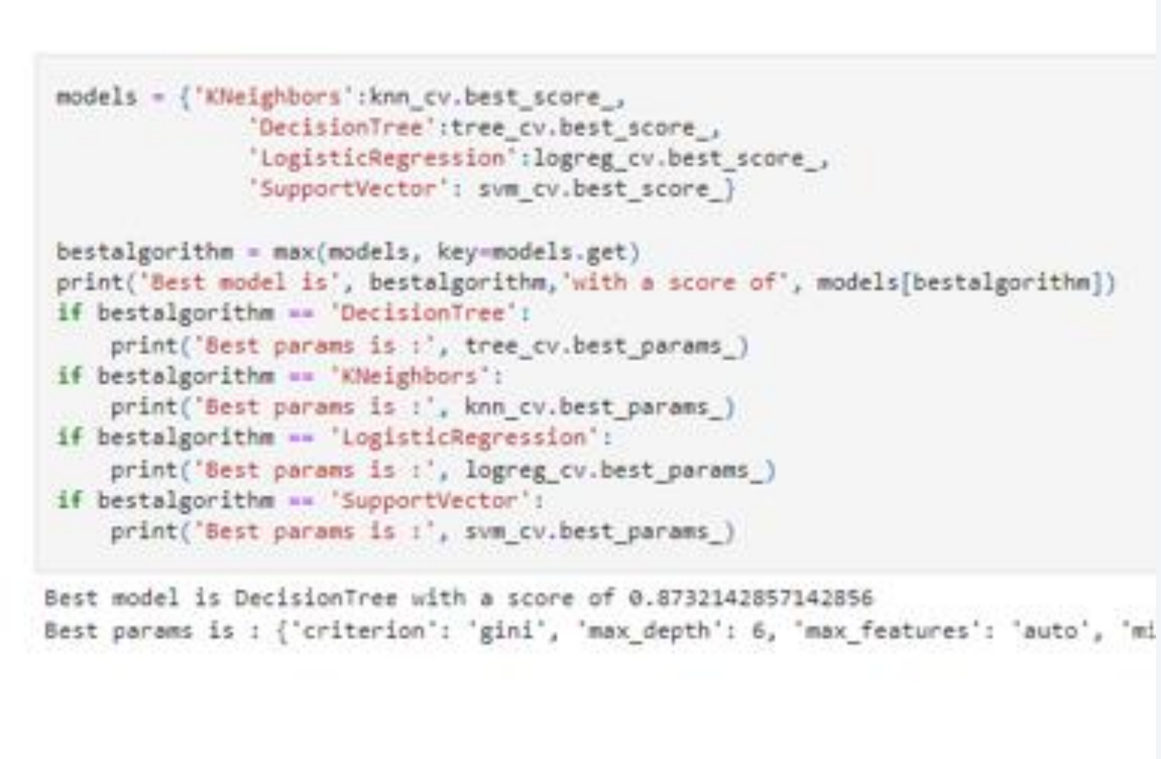


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree had the best accuracy score



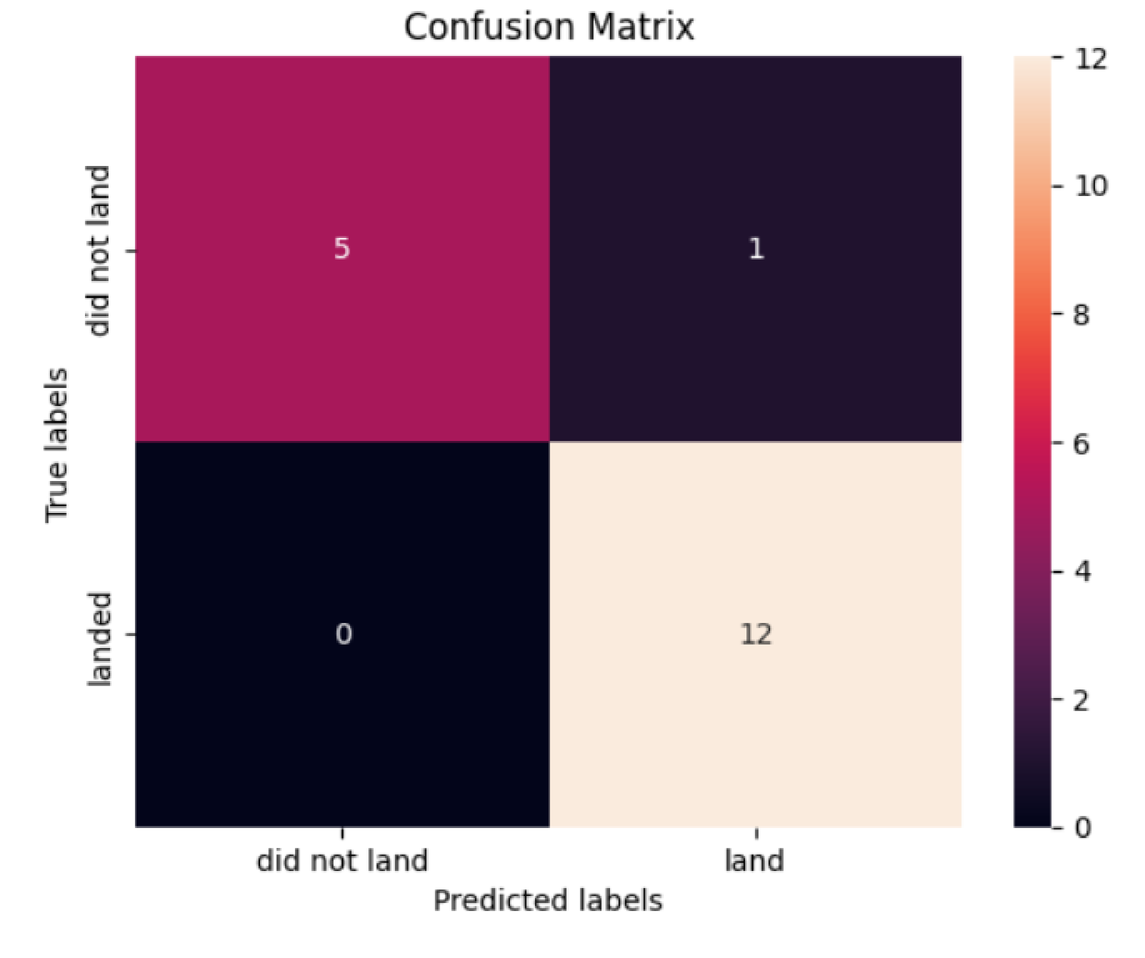
```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

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 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'mi
```

Confusion Matrix

- The Decision Tree model provided the best performance of all the models only incorrectly predicting 1 instance.



Conclusions

In concluding our research, we have found the following:

- The launches have a higher success rate over time indicting the launch teams are learning from prior launches
- There are specific orbits that have a higher success rate and should be studied in more detail.
- KSC LC-39A had the most successful lunches and more research should be done with launch site data
- All the classification models do well at predicting the outcomes, however the Decision Tree performs the best and should be used going forward.

Thank you!