IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ngan Tran

May 2025

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

- 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 of Falcon 9 will land, we can determine the cost of a launch. This information can be useful for a rival company of Space X.
- Problems we want to find answers:
  - What factors determine if the rocket will land successfully?
  - Which machine learning model would work best (have the highest accuracy) to predict the outcome of a Falcon 9 first stage landing from a future launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX launch data was gathered from the SpaceX REST API with the endpoint (or URL): api.spacexdata.com/v4/launches/past
  - Falcon 9 Launch data was collected by web scraping related Wiki pages: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Perform data wrangling
  - Perform some Exploratory Data Analysis (EDA) to find some patterns in the data, using the method value_counts() on some columns
  - Determine the landing outcome label for training supervised models.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash

Methodology (cont.)

Executive Summary

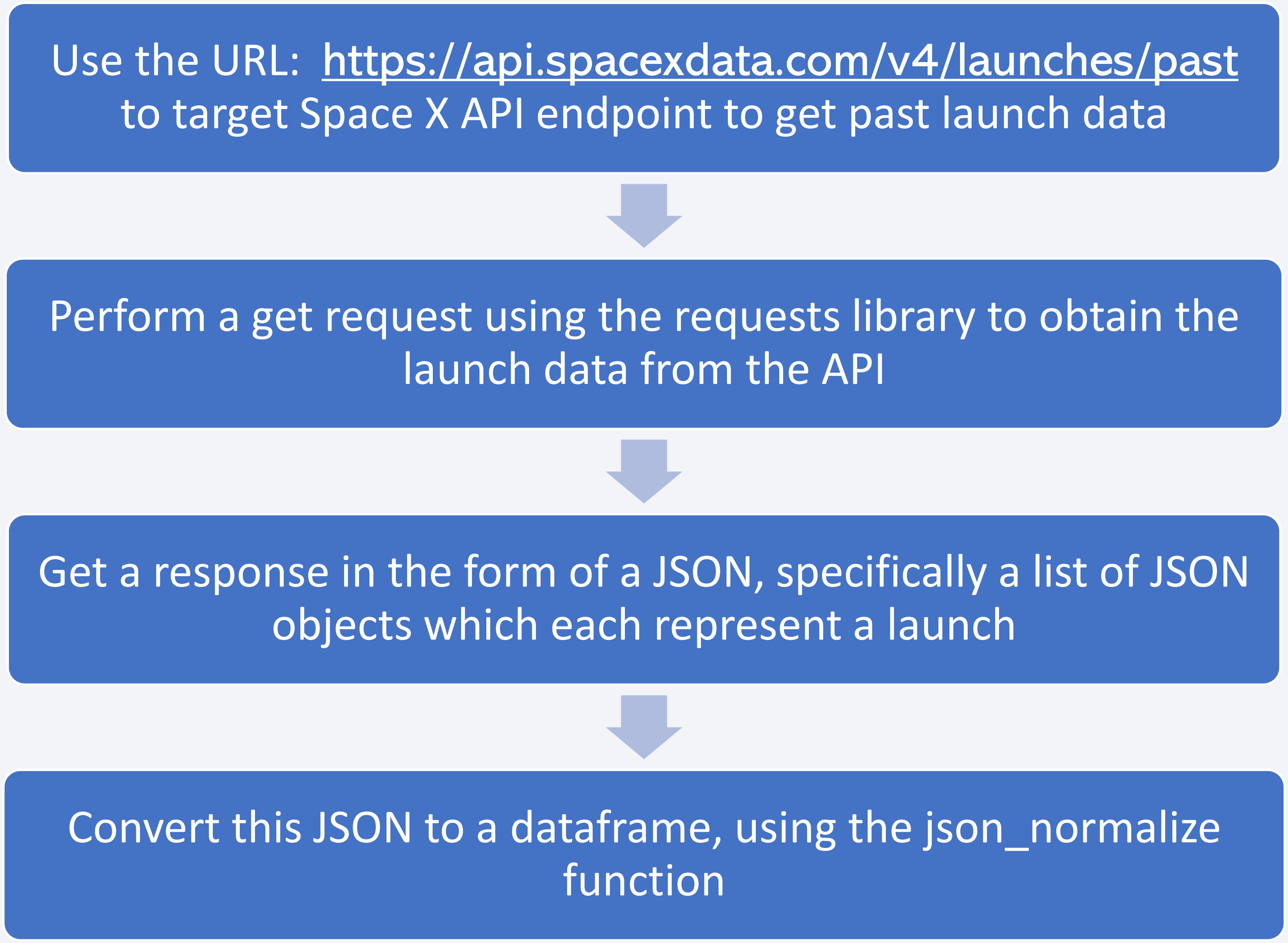
- Perform predictive analysis using classification models:
  - Perform exploratory Data Analysis and determine Training Labels:
    - Create the column ‘Class’ (training labels converted from landing outcomes)
    - Standardize the data
    - Split the data into training and test data using function train_test_split
    - Train the model and perform Grid Search to find the best hyperparameter for SVM, Classification Trees and Logistic Regression.
  - Find the method performs best using test data.

Data Collection

- SpaceX launch data was gathered from the SpaceX REST AP: https://api.spacexdata.com/v4/launches/past
- Falcon 9 Launch data was collected by web scraping related Wiki pages: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

- Using the Python Requests library to get data from the SpaceX API
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Use the URL: https://api.spacexdata.com/v4/launches/past to target Space X API endpoint to get past launch data

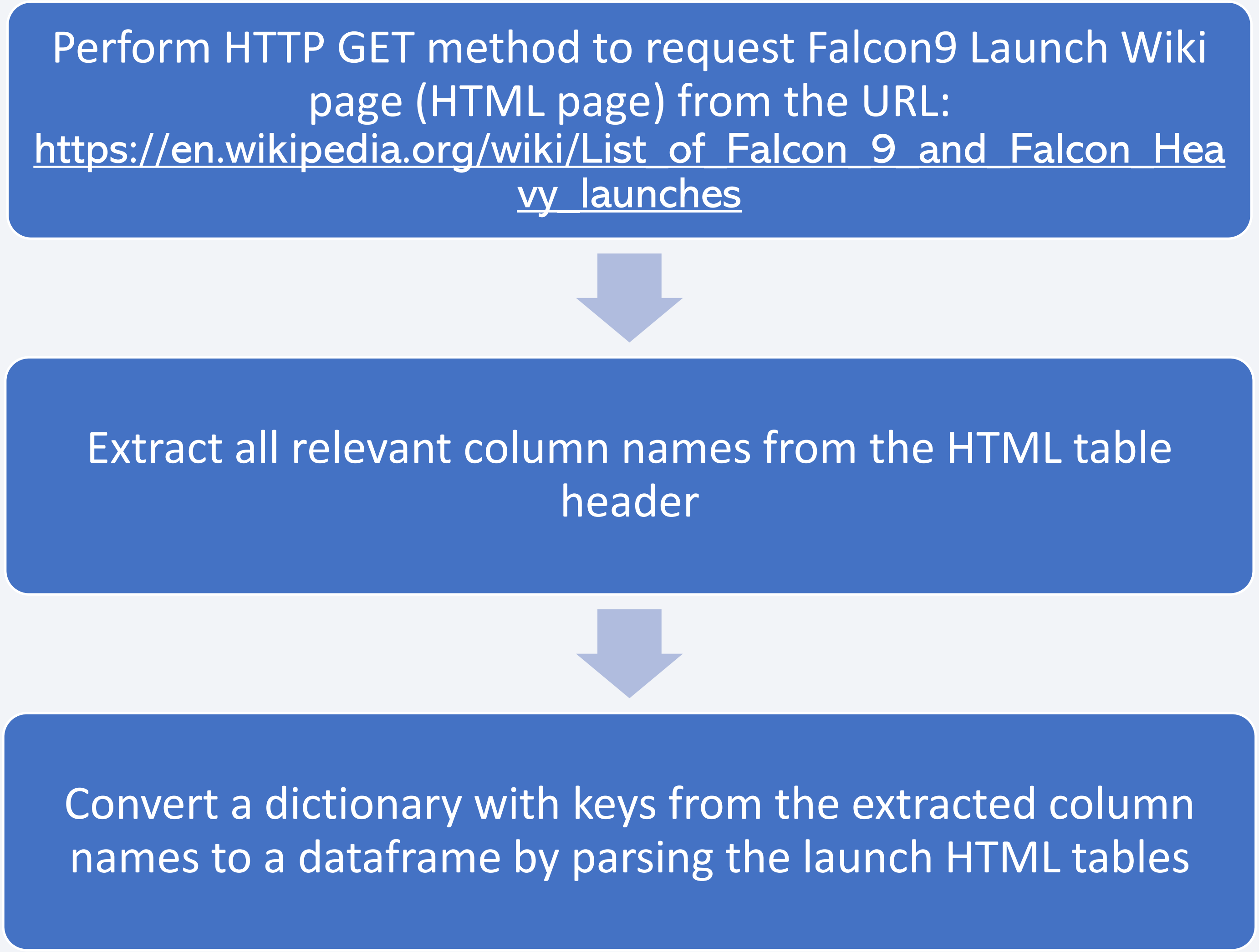
Perform a get request using the requests library to obtain the launch data from the API

Get a response in the form of a JSON, specifically a list of JSON objects which each represent a launch

Convert this JSON to a dataframe, using the json_normalize function

Data Collection – Scraping

- Using the Python BeautifulSoup package to web scrape HTML tables that contain valuable Falcon 9 launch records
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/jupyter-labs-webscraping.ipynb



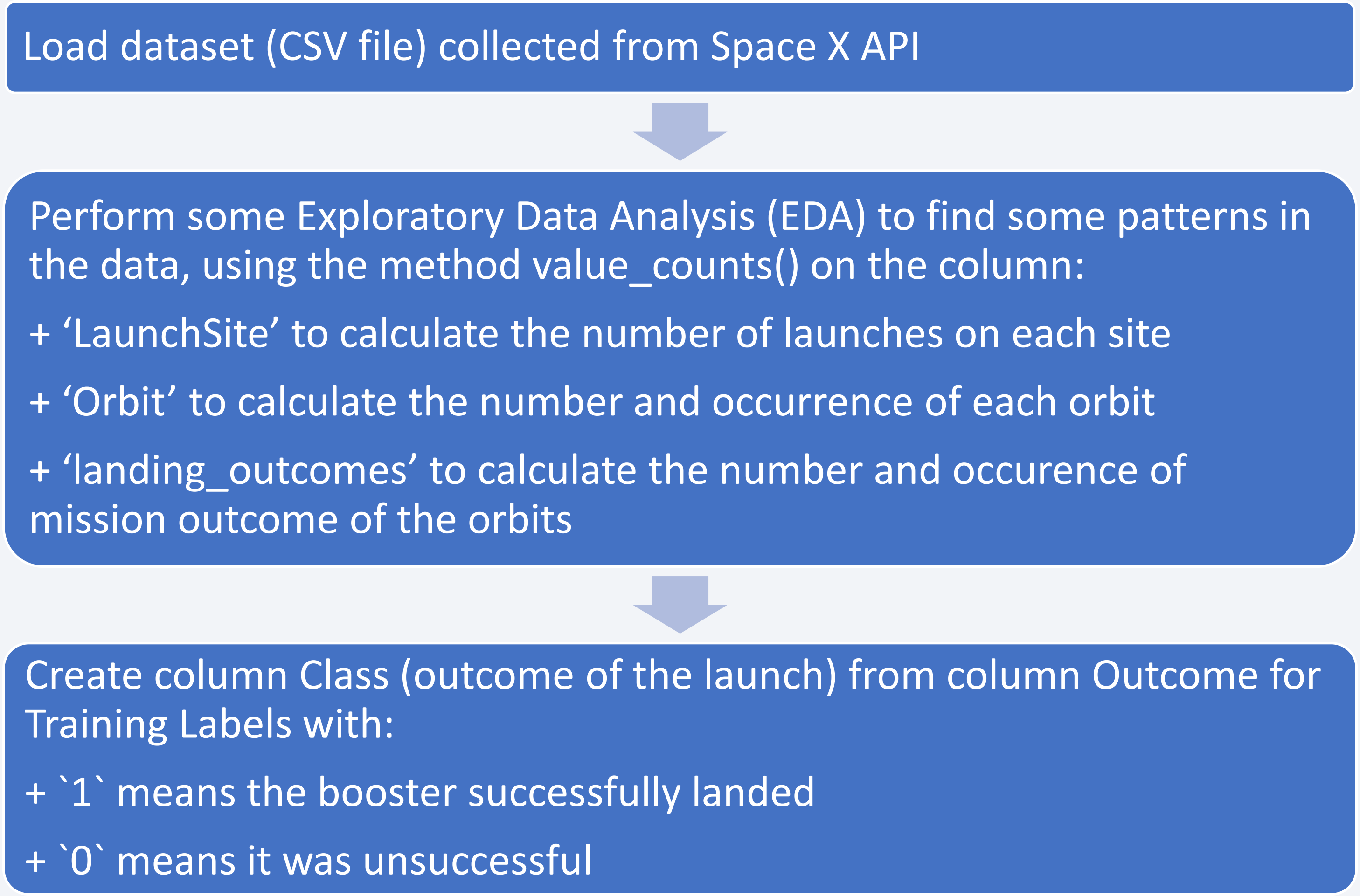
Perform HTTP GET method to request Falcon9 Launch Wiki page (HTML page) from the URL: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Extract all relevant column names from the HTML table header

Convert a dictionary with keys from the extracted column names to a dataframe by parsing the launch HTML tables

Data Wrangling

- Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



Load dataset (CSV file) collected from Space X API

Perform some Exploratory Data Analysis (EDA) to find some patterns in the data, using the method value_counts() on the column:

+ ‘LaunchSite’ to calculate the number of launches on each site

+ ‘Orbit’ to calculate the number and occurrence of each orbit

+ ‘landing_outcomes’ to calculate the number and occurence of mission outcome of the orbits

Create column Class (outcome of the launch) from column Outcome for Training Labels with:

+ `1` means the booster successfully landed

+ `0` means it was unsuccessful

EDA with Data Visualization

- The following charts were plotted:
    - Scatter point chart to see how the Flight Number and Payload Mass variables would affect the launch outcome (Class)
    - Scatter point chart to see how the Flight Number and Launch Site variables would affect the launch outcome (Class)
    - Scatter point chart to see how the Payload Mass and Launch Site variables would affect the launch outcome (Class)
    - Bar chart to check if there are any relationship between success rate (Class) and Orbit type
    - Scatter point chart to see how the Flight Number and Orbit type variables would affect the launch outcome (Class)
    - Scatter point chart to see how the Payload Mass and Orbit type variables would affect the launch outcome (Class)
    - Line chart to see the launch success (Class) trend by Year
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/edadataviz.ipynb

EDA with SQL

- The following SQL queries were performed:
  - SELECT DISTINCT Launch_Site FROM SPACEXTABLE to display the names of the unique launch sites in the space mission
  - SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5 to display 5 records where launch sites begin with the string 'CCA'
  - SELECT SUM(PAYLOAD_MASS__KG_) AS Total_PayloadMass FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)' to display the total payload mass carried by boosters launched by NASA (CRS)
  - SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_PayloadMass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1' to display average payload mass carried by booster version F9 v1.1
  - SELECT MIN(Date) AS FirstSuccessfull_landing_date FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)' to list the date when the first successful landing outcome in ground pad was achieved.
  - SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000 to list the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

EDA with SQL (cont.)

- SELECT COUNT(Mission_Outcome) AS SuccessOutcome FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Success%’ to list the total number of successful and failure mission outcomes
- SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) ORDER BY Booster_Version to list all the booster_versions that have carried the maximum payload mass, using a subquery.
- SELECT Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Failure (drone ship)’ AND Date BETWEEN '2015-01-01' AND '2015-12-31’ to 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.
- SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20’ GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC to 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.

- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

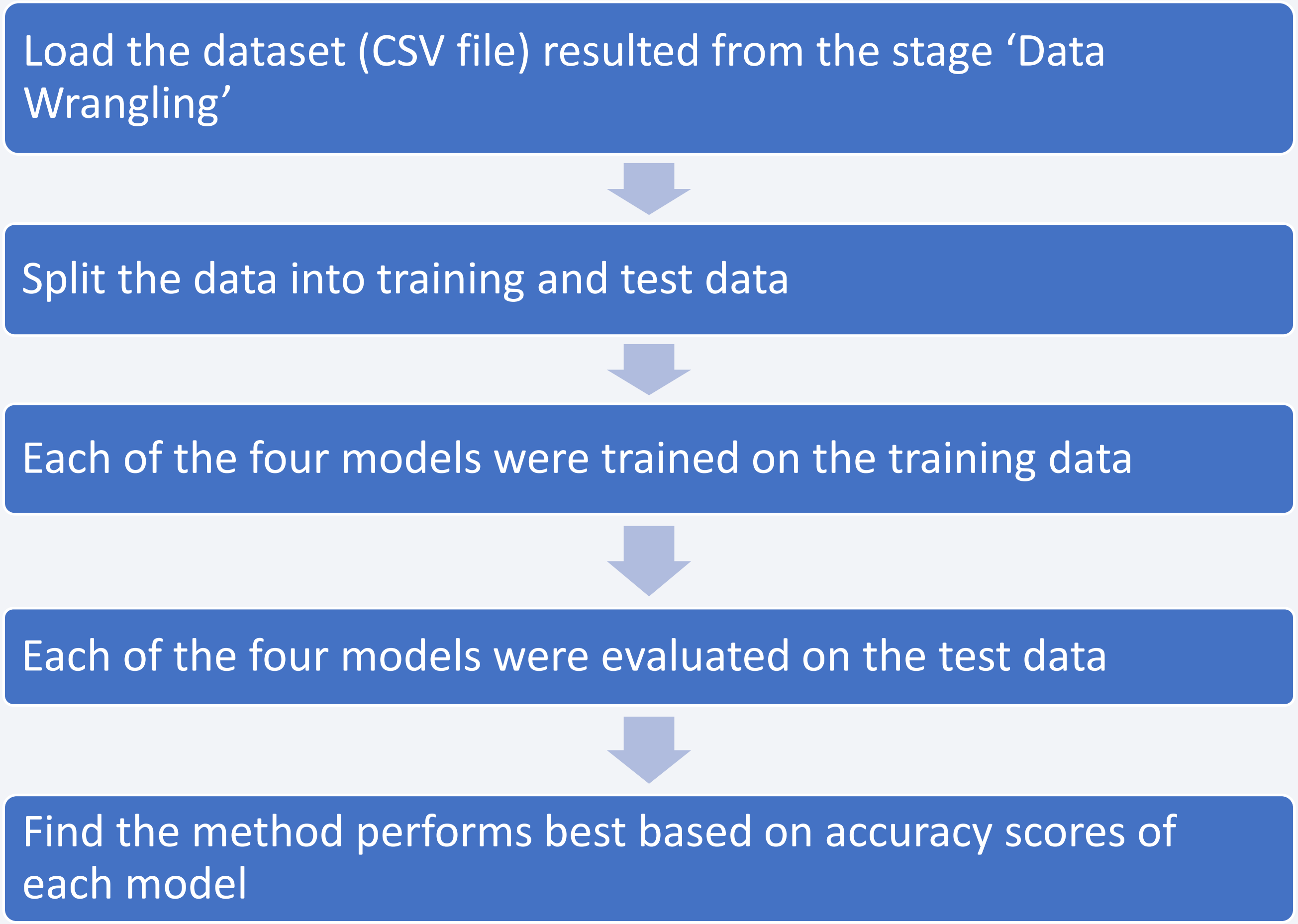
- Map objects were created and added to a folium map:
  - Circles and markers for each launch site to explore the launch sites on the map
  - Color-labeled markers for the launch outcomes for each site to easily identify which launch sites have relatively high success rates
  - Lines between a launch site to the selected coastline point, a closest city, railway, highway to see if launch sites are in close proximity to coastline, railway, highway and if launch sites keep certain distance away from cities
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Plots/graphs and interactions you have added to a dashboard:
  - Dropdown list to enable Launch Site selection (All sites or a specific launch site)
  - Pie chart to show the total successful launches for all sites. If a specific launch site was selected, show the Success vs. Failed counts for the site.
  - Range slider to select payload range
  - Scatter chart to show the correlation between payload and launch success
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

- We split the data into training and test data using the function train_test_split
- Four machine learning models were trained on the training dataset: Logistic Regression, SVM (support vector machine), Decision Tree and KNN (k-Nearest Neighbors)
- Find the best Hyperparameters for each model using the function GridSearchCV
- Find the method performs best based on accuracy scores of each model.
- GitHub URL: https://github.com/ngantran-1/IBM-Data-Science-Course/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Load the dataset (CSV file) resulted from the stage ‘Data Wrangling’

Split the data into training and test data

Each of the four models were trained on the training data

Each of the four models were evaluated on the test data

Find the method performs best based on accuracy scores of each model

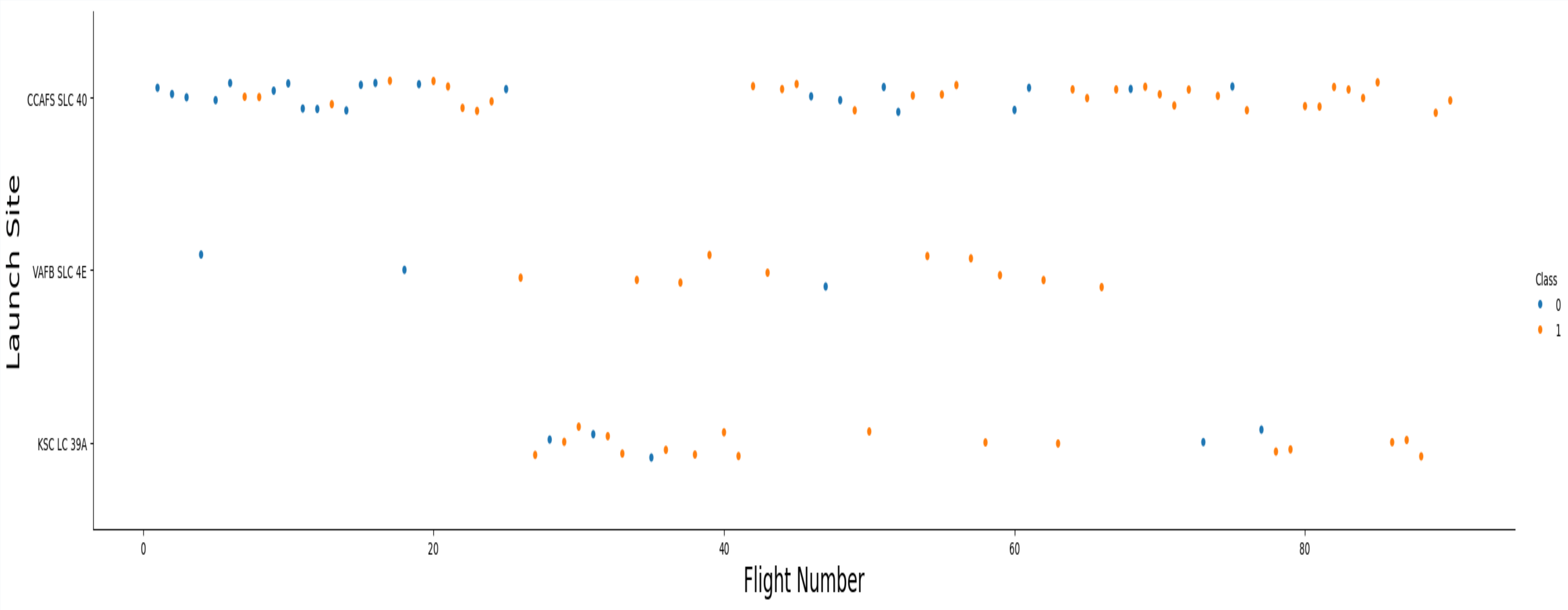
Results

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

Section 2

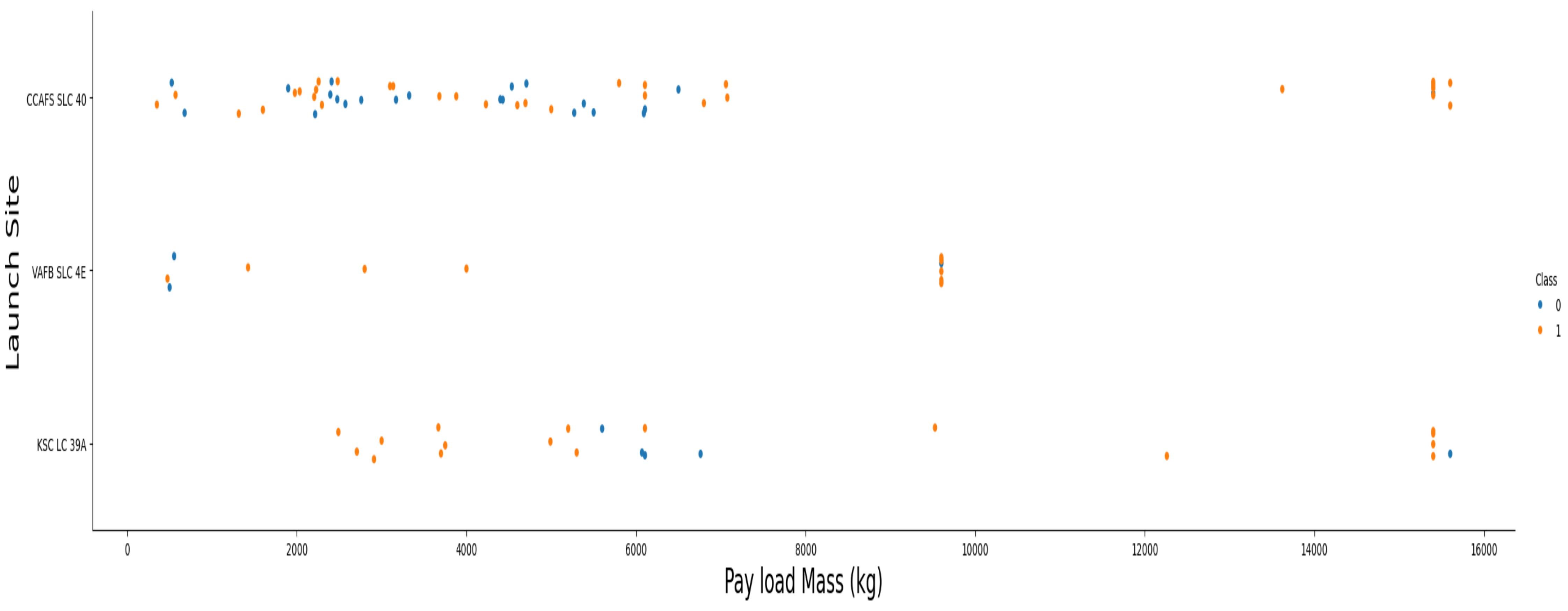
Insights drawn from EDA

Flight Number vs. Launch Site



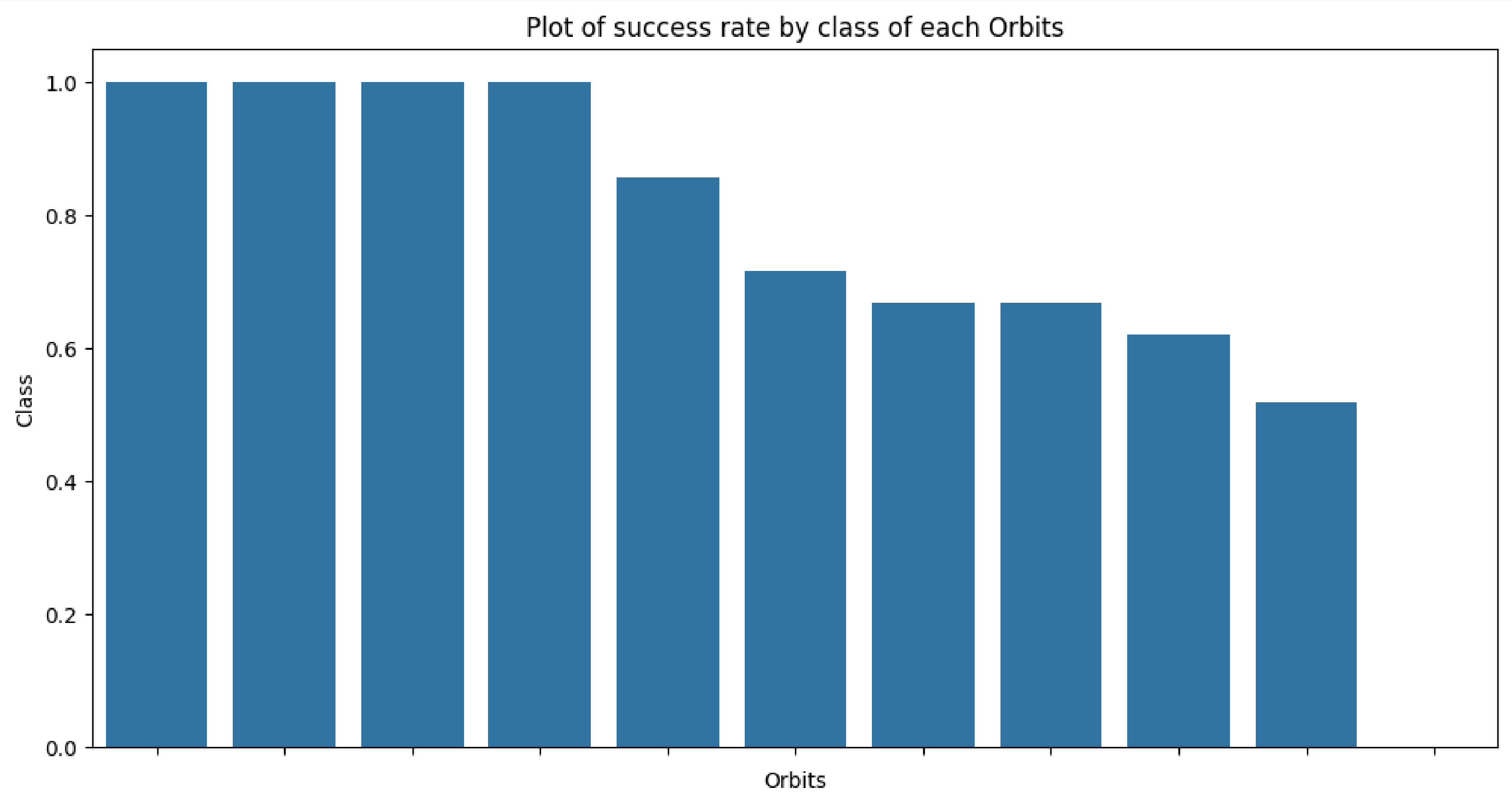
We see that as the flight number increases, the first stage is more likely to land successfully (Class 1).

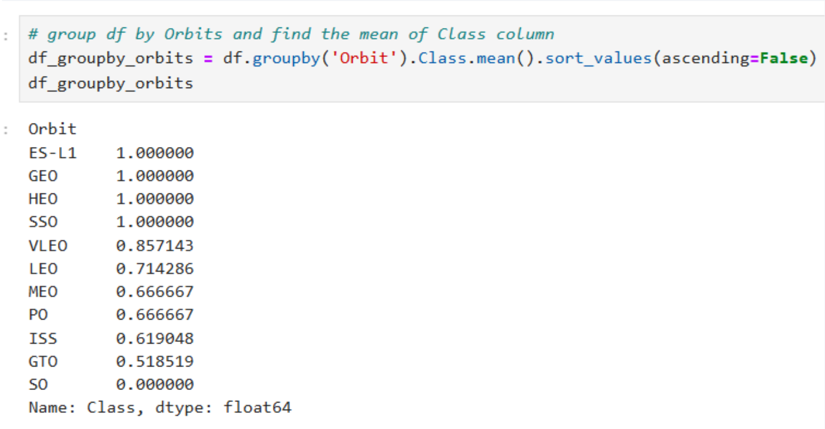
Payload vs. Launch Site



We see that for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000)

Success Rate vs. Orbit Type



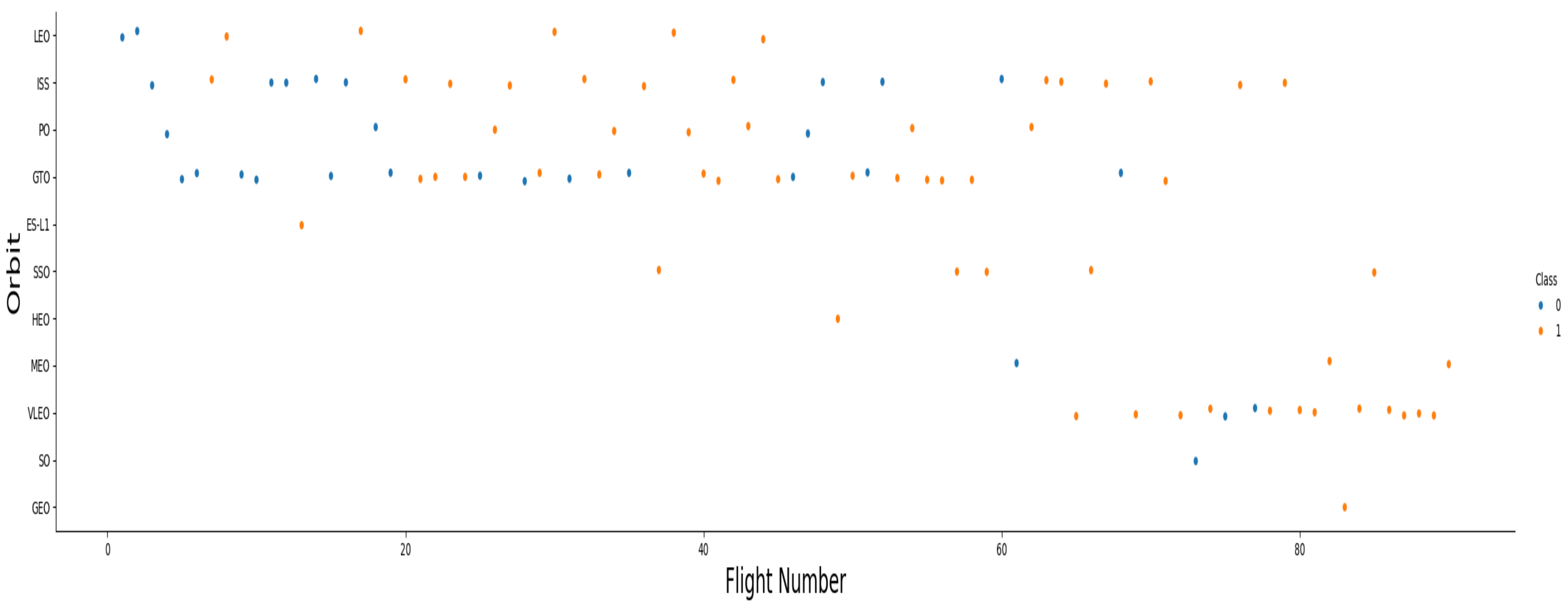


```
# group df by Orbits and find the mean of Class column
df_groupby_orbits = df.groupby('Orbit').Class.mean().sort_values(ascending=False)
df_groupby_orbits
```

```
Orbit
ES-L1    1.000000
GEO      1.000000
HEO      1.000000
SSO      1.000000
VLEO     0.857143
LEO      0.714286
MEO      0.666667
PO       0.666667
ISS      0.619048
GTO      0.518519
SO       0.000000
Name: Class, dtype: float64
```

We identify Orbits ES-L1, GEO, HEO and SSO have the highest success rates, while SO has the lowest success rate.

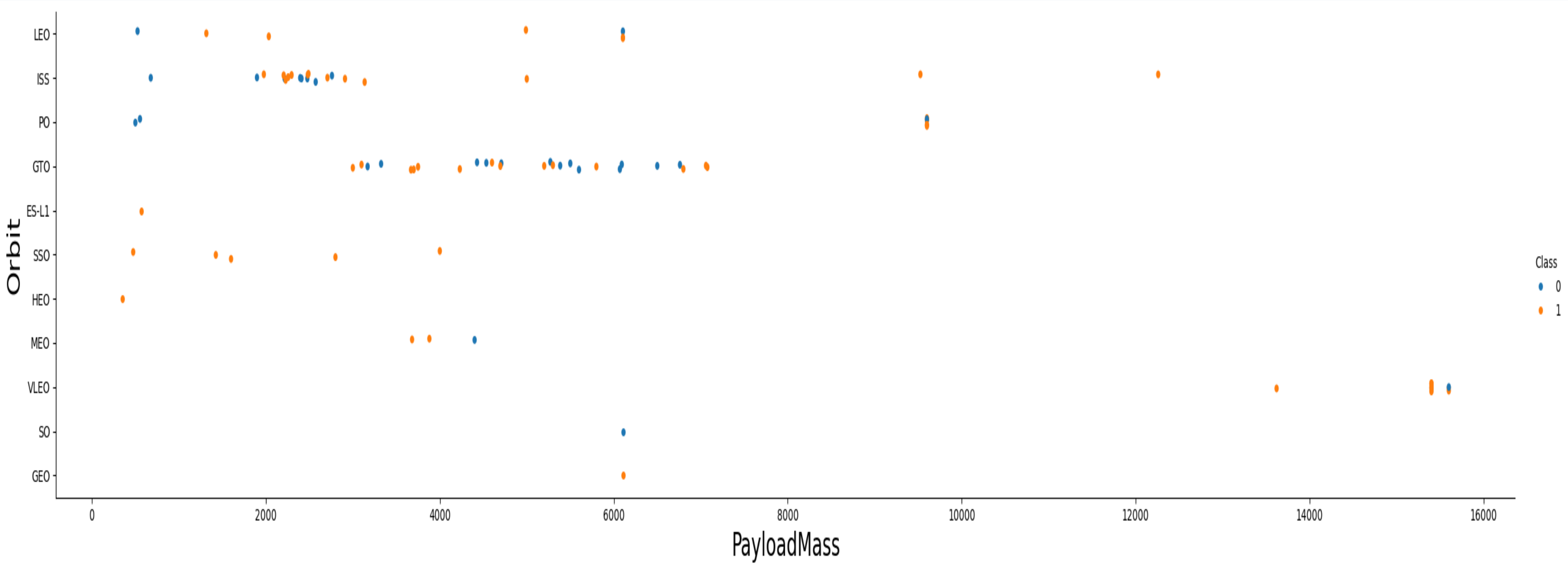
Flight Number vs. Orbit Type



We observe that in the LEO orbit success seems to be related to the number of flights.

Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

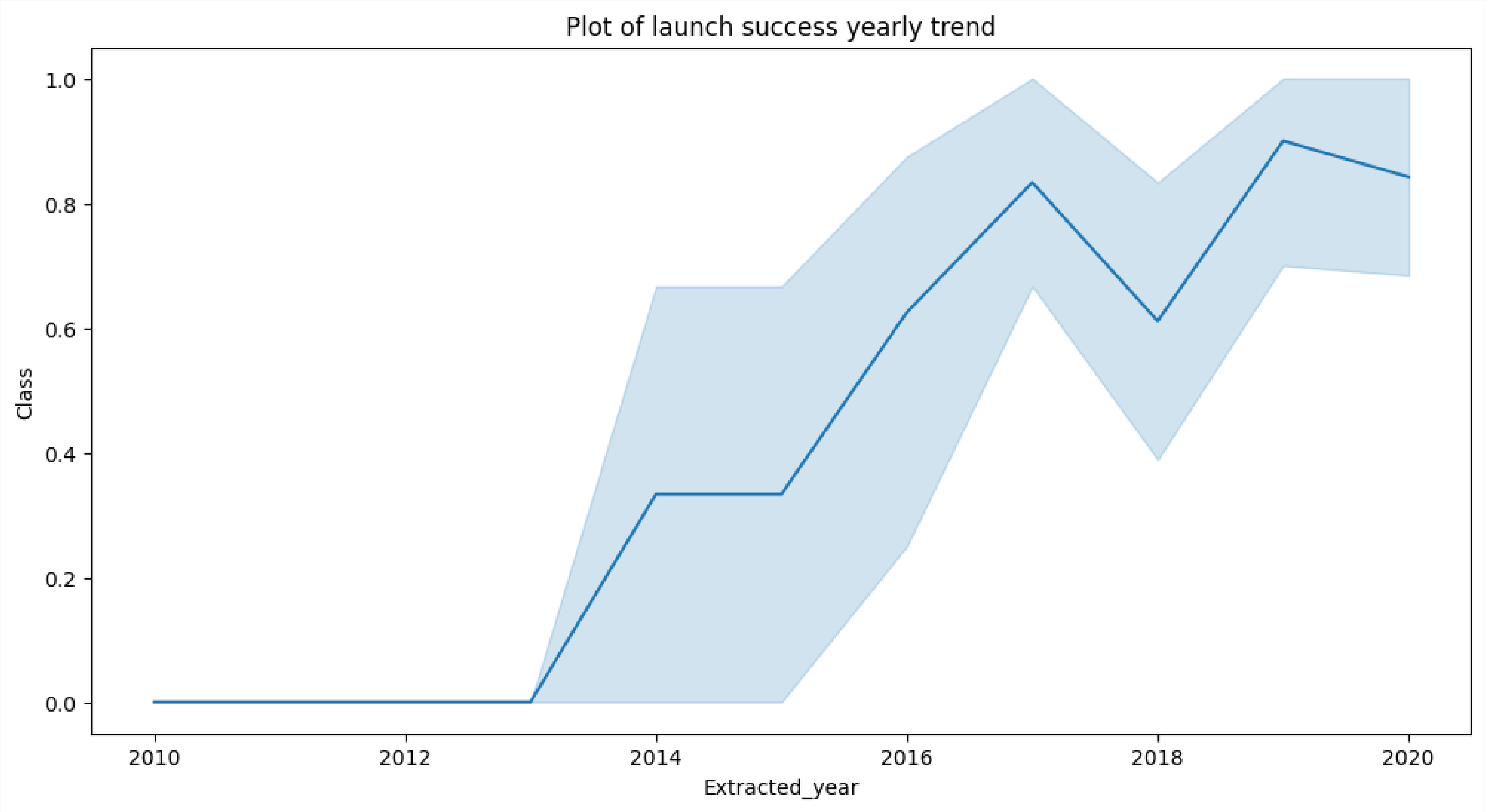
Payload vs. Orbit Type



With heavy payloads, the successful landing rate is high for Orbits Polar, LEO and ISS.

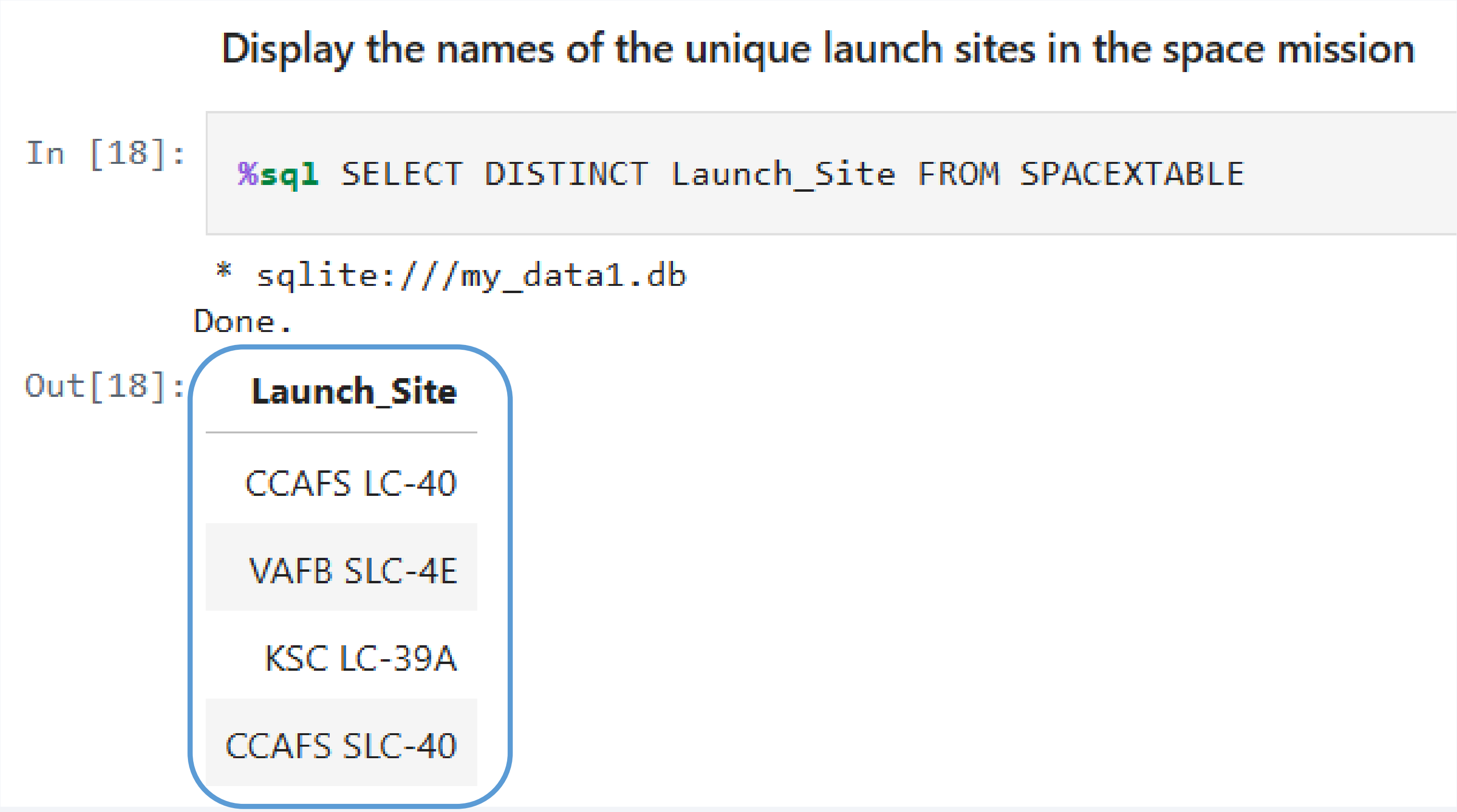
However, for Orbit GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



We observe that the success rate since 2014 was increasing till 2020, except 2018.

All Launch Site Names



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

```
In [18]:  %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

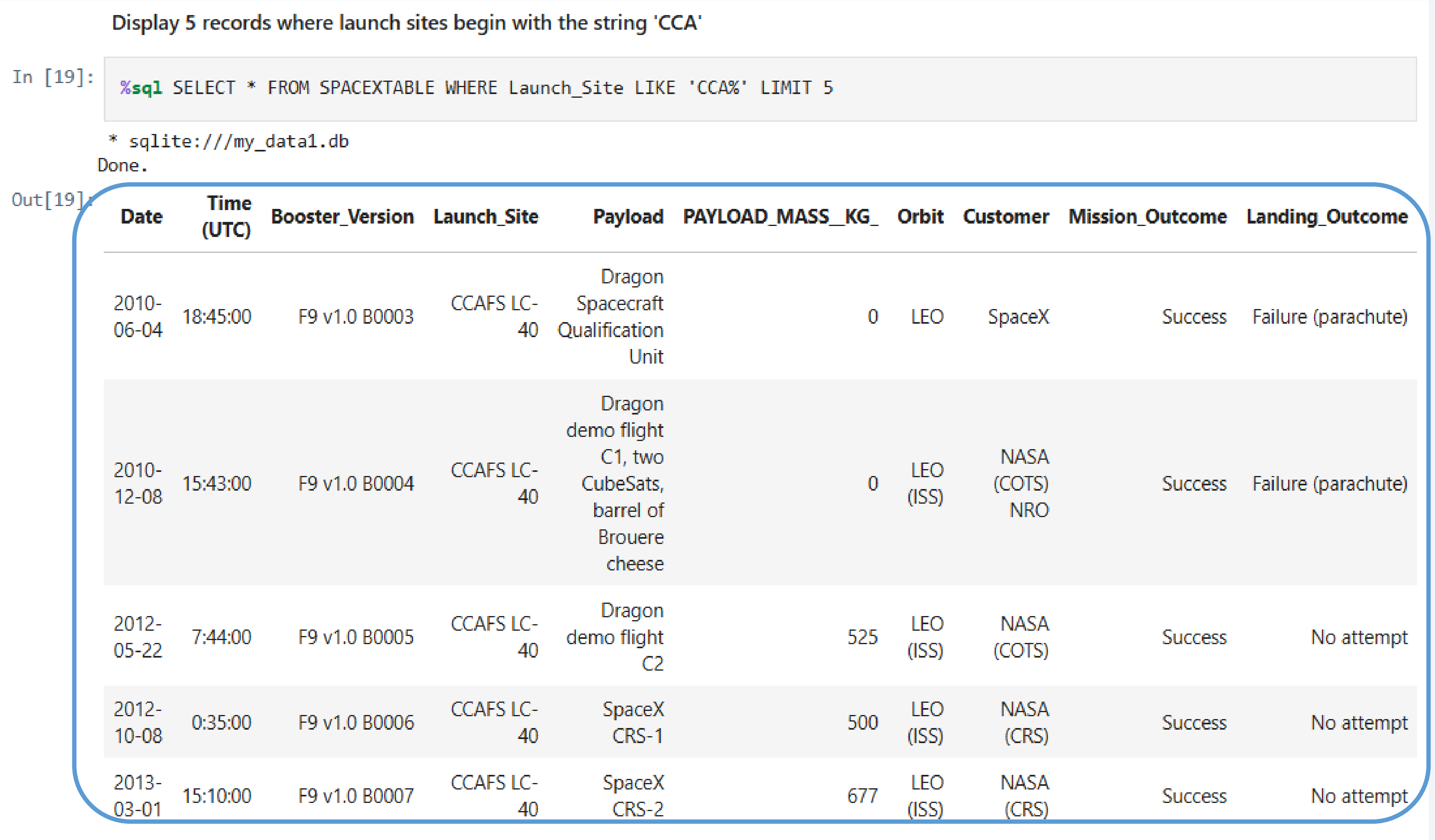
```
 * sqlite:///my_data1.db
Done.
```

Out[18]:

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

Explanation: There are 4 unique launch sites.

Launch Site Names Begin with 'CCA'



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

```
In [19]:  %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

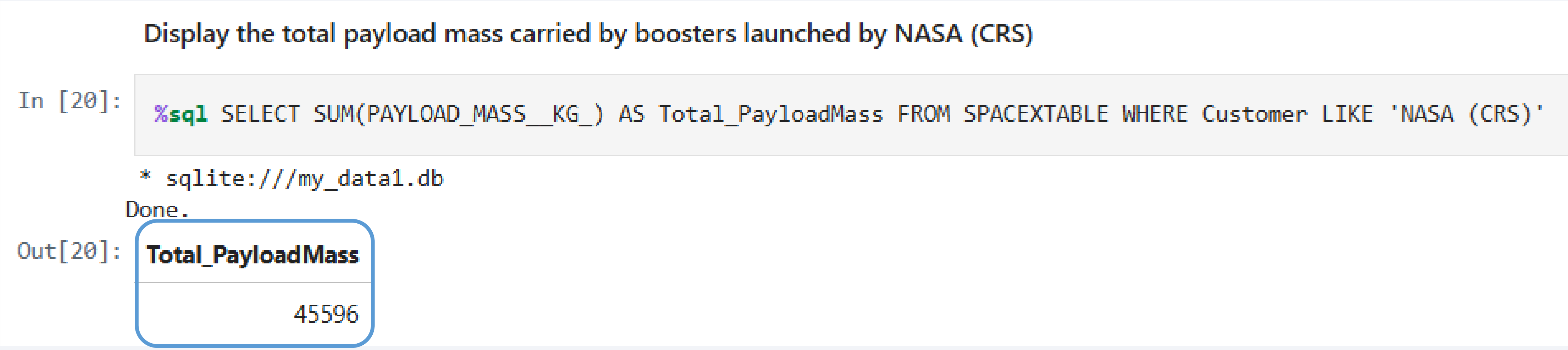
```
 * sqlite:///my_data1.db
Done.
```

Out[19]:

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

Explanation: The result shows 5 records of 'CCAFS LC-40' Launch Site

Total Payload Mass



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

```
In [20]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_PayloadMass FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)'
```

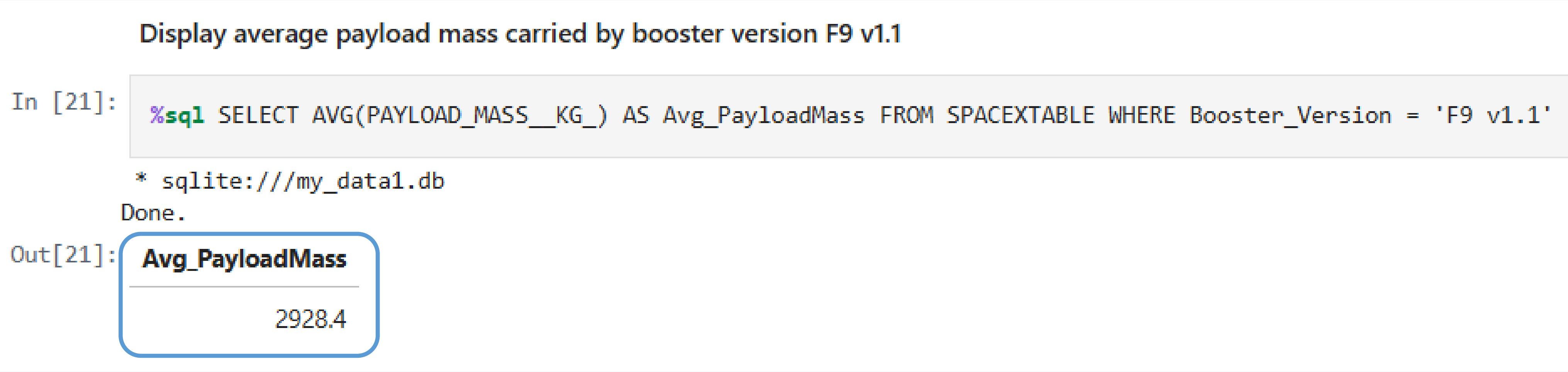
```
 * sqlite:///my_data1.db
Done.
```

Out[20]:

Total_PayloadMass
45596

Explanation: Total Payload Mass carried by boosters launched by NASA (CRS) is 45596

Average Payload Mass by F9 v1.1



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

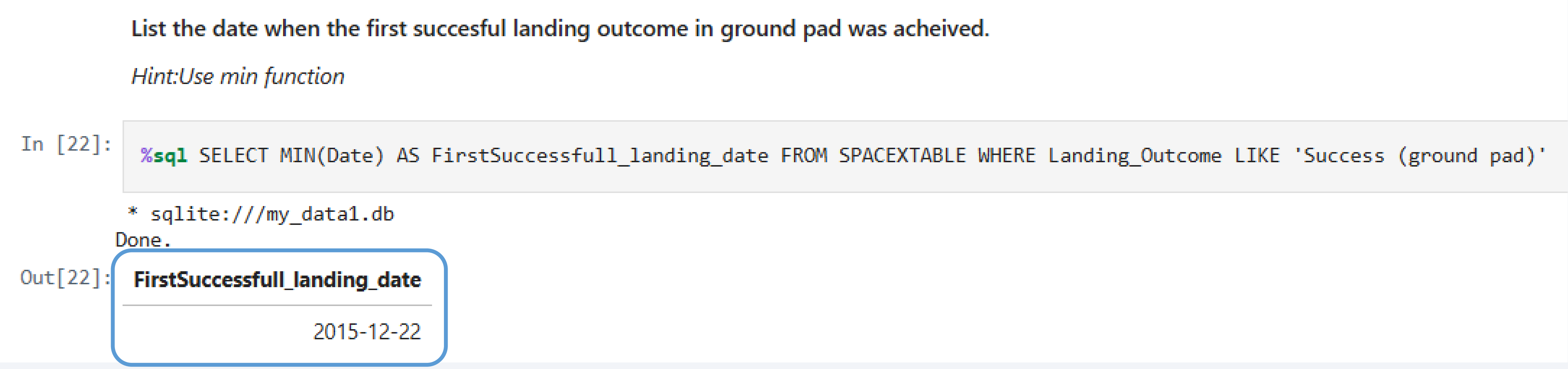
```
In [21]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_PayloadMass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'

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

Out[21]:

Avg_PayloadMass
2928.4

Explanation: Average Payload Mass carried by booster version F9 v1.1 is 2928.4

First Successful Ground Landing Date



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

Hint:Use min function

```
In [22]: %sql SELECT MIN(Date) AS FirstSuccessfull_landing_date FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)'
```

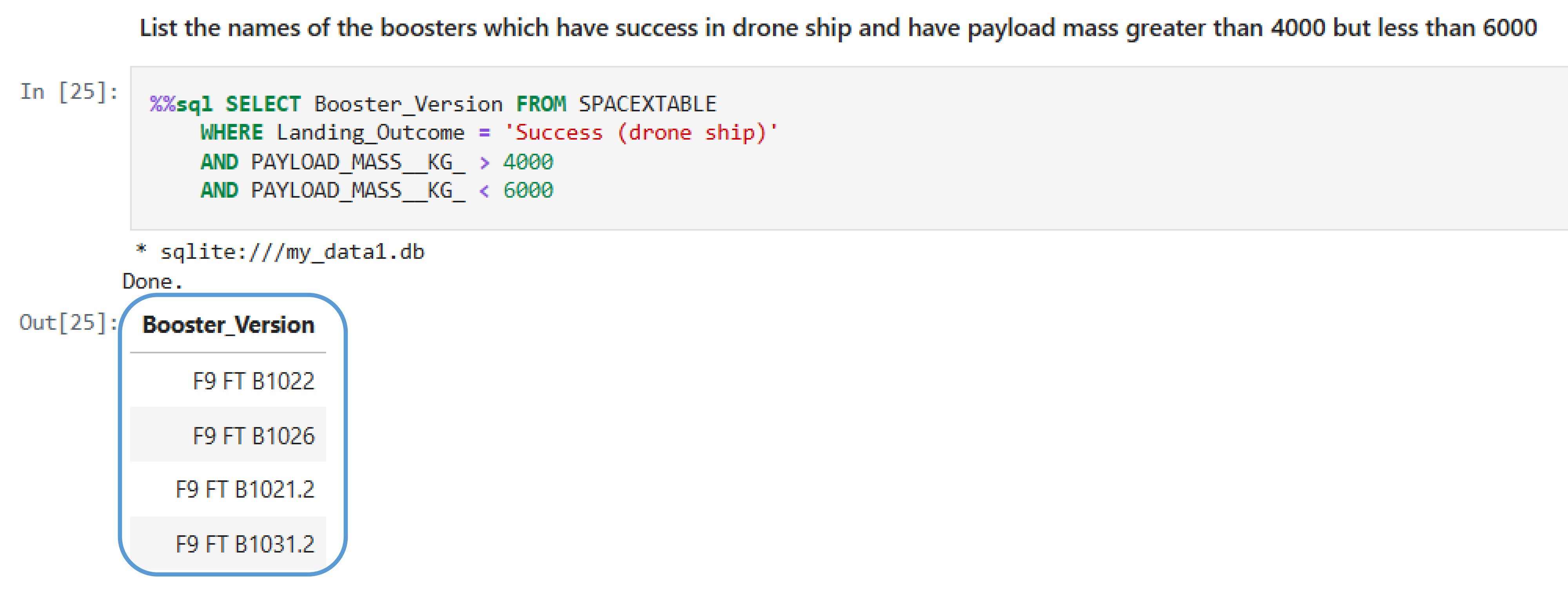
```
 * sqlite:///my_data1.db
Done.
```

Out[22]:

FirstSuccessfull_landing_date
2015-12-22

Explanation: The date when the first successful landing outcome in ground pad is 22-Dec-2015.

Successful Drone Ship Landing with Payload between 4000 and 6000



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

In [25]:

```
%%sql SELECT Booster_Version FROM SPACEXTABLE
    WHERE Landing_Outcome = 'Success (drone ship)'
    AND PAYLOAD_MASS__KG_ > 4000
    AND PAYLOAD_MASS__KG_ < 6000
```

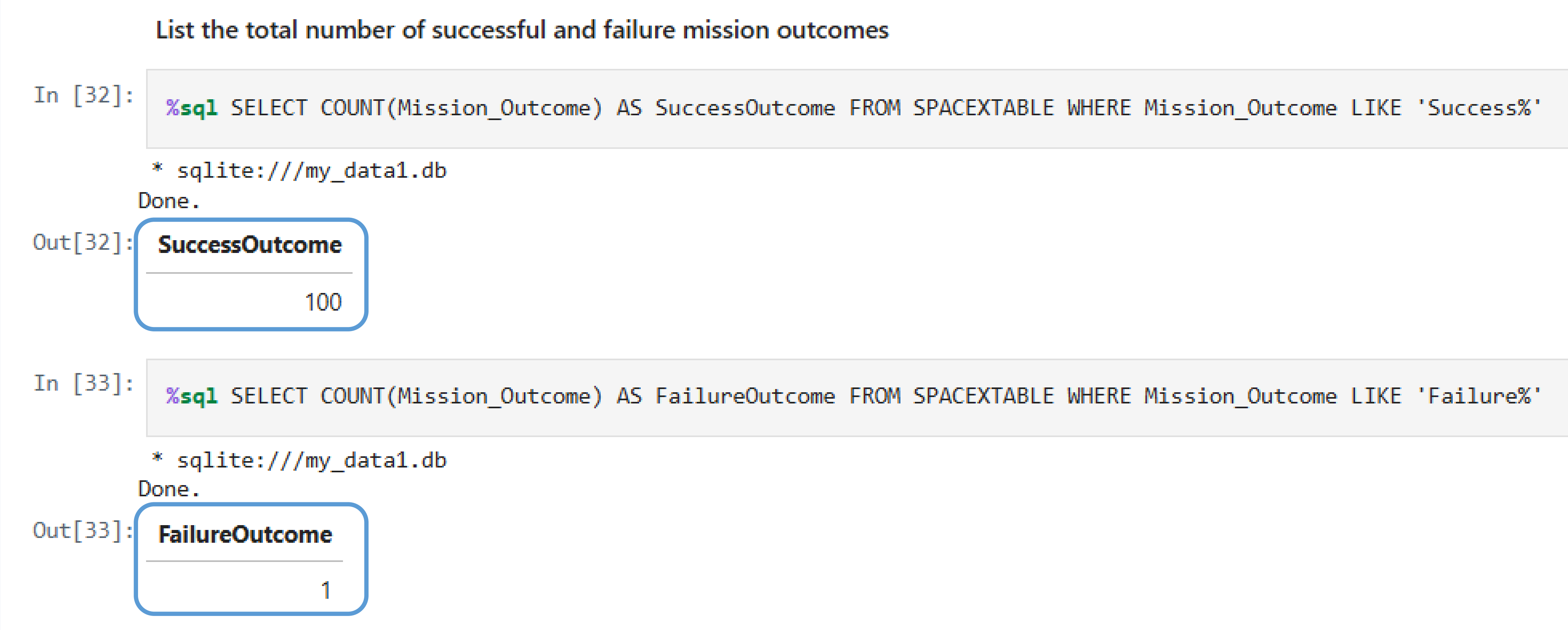
```
 * sqlite:///my_data1.db
Done.
```

Out[25]:

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

Explanation: There are 4 booster versions 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



List the total number of successful and failure mission outcomes

```
In [32]: %sql SELECT COUNT(Mission_Outcome) AS SuccessOutcome FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Success%'
```

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

Out[32]:

SuccessOutcome
100

```
In [33]: %sql SELECT COUNT(Mission_Outcome) AS FailureOutcome FROM SPACEXTABLE WHERE Mission_Outcome LIKE 'Failure%'
```

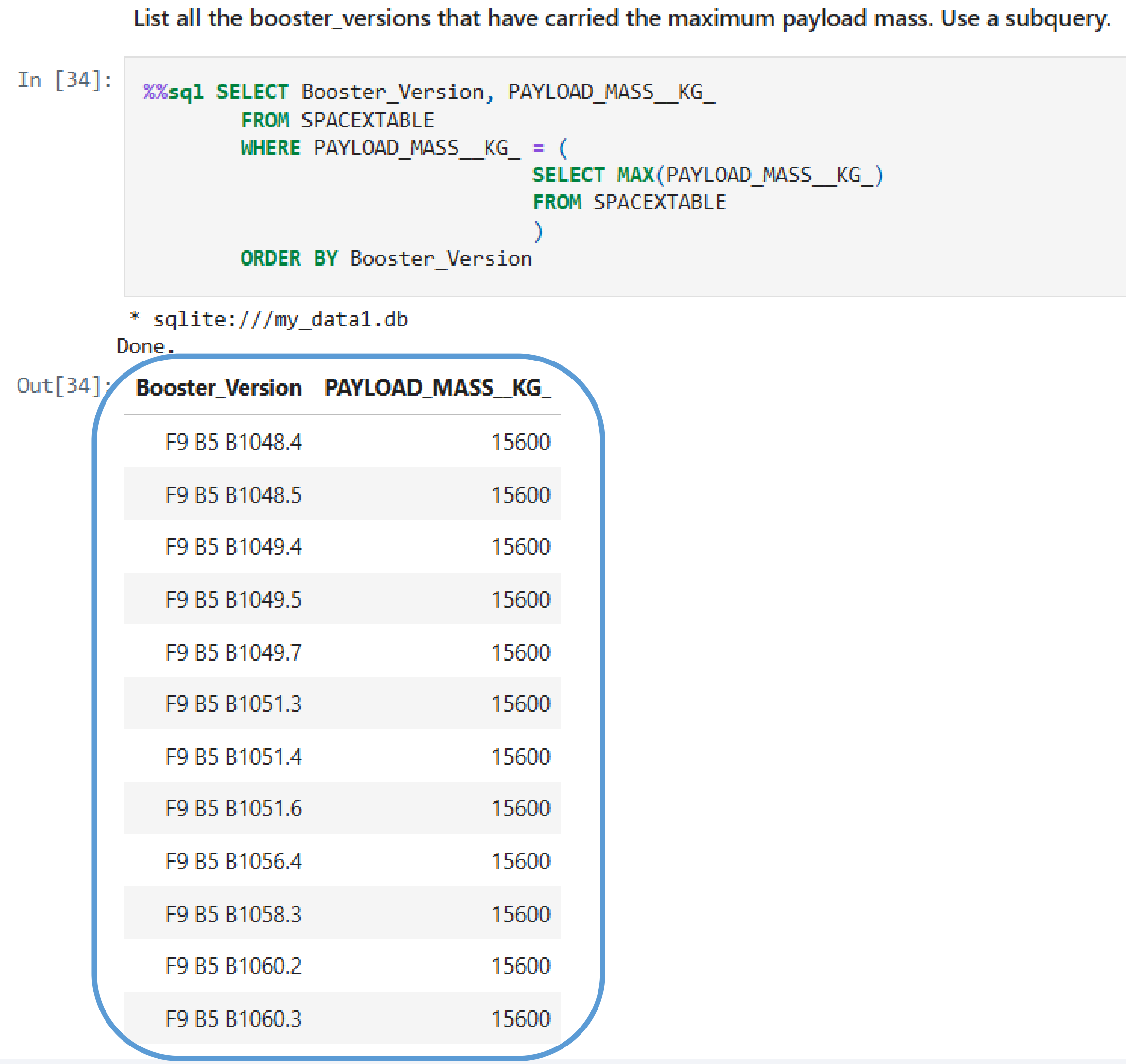
```
 * sqlite:///my_data1.db
Done.
```

Out[33]:

FailureOutcome
1

Explanation: Total number of successful mission outcomes is 100 and Total number of failure mission outcomes is 1 only.

Boosters Carried Maximum Payload



List all the booster_versions that have carried the maximum payload mass. Use a subquery.

```
In [34]: %%sql SELECT Booster_Version, PAYLOAD_MASS__KG_
                 FROM SPACEXTABLE
                 WHERE PAYLOAD_MASS__KG_ = (
                                         SELECT MAX(PAYLOAD_MASS__KG_)
                                         FROM SPACEXTABLE
                                         )
                 ORDER BY Booster_Version
```

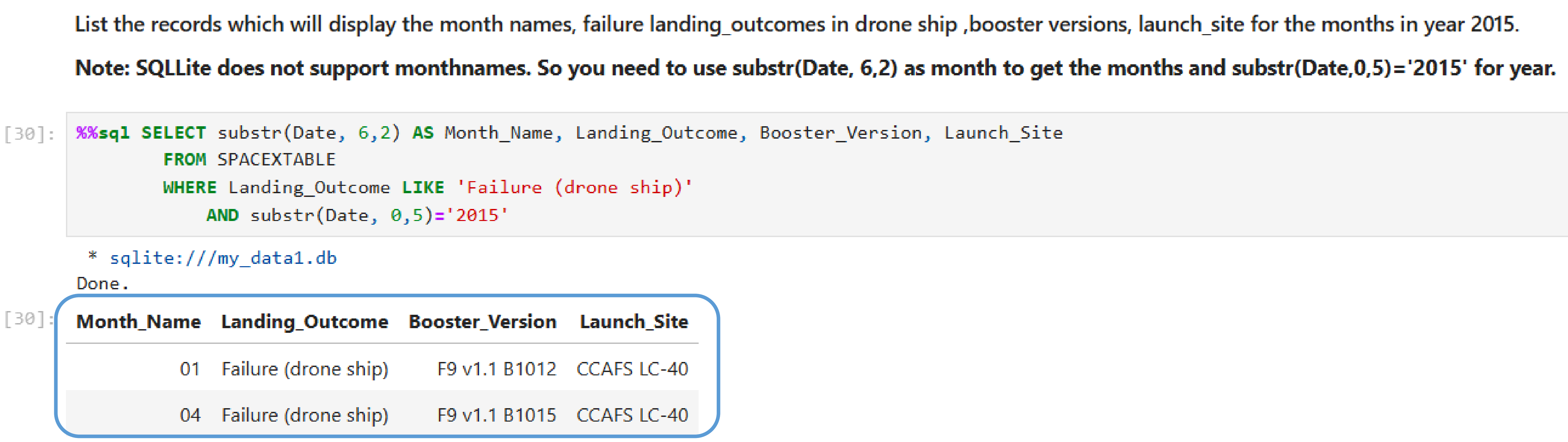
```
 * sqlite:///my_data1.db
Done.
```

Out[34]:

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

Explanation: The result shows the names of the booster which have carried the maximum payload mass (15600 kg)

2015 Launch Records



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.

```
[30]: %%sql SELECT substr(Date, 6,2) AS Month_Name, Landing_Outcome, Booster_Version, Launch_Site
              FROM SPACEXTABLE
              WHERE Landing_Outcome LIKE 'Failure (drone ship)'
                  AND substr(Date, 0,5)='2015'
```

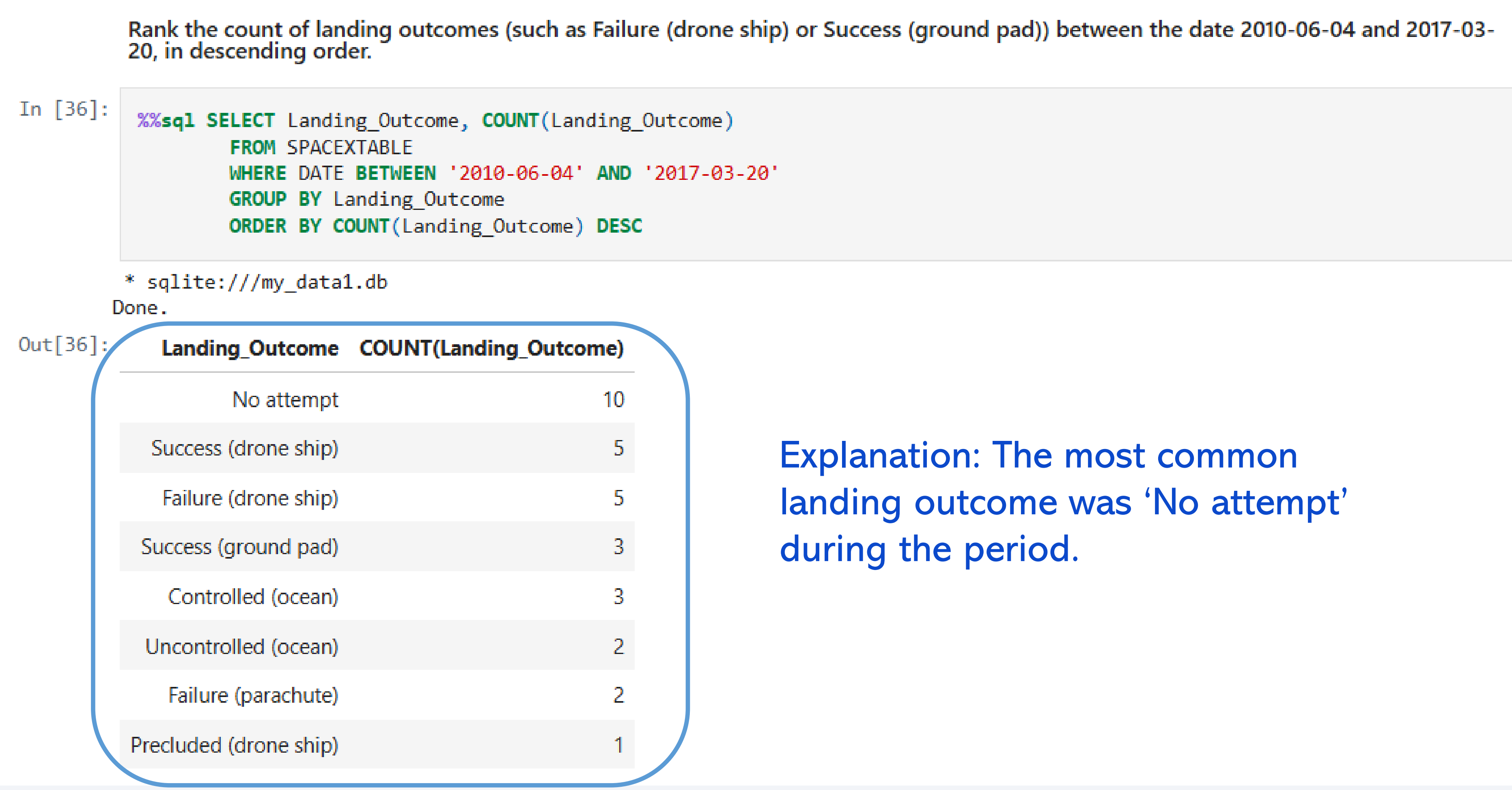
```
 * sqlite:///my_data1.db
Done.
```

[30]:

Month_Name	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

Explanation: There are 2 records of failed landing outcomes in drone ship in year 2015. One occurs in January and the other in April. Both launch site names in the records are CCAFS LC-40.

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



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.

```
In [36]: %%sql SELECT Landing_Outcome, COUNT(Landing_Outcome)
                 FROM SPACEXTABLE
                 WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
                 GROUP BY Landing_Outcome
                 ORDER BY COUNT(Landing_Outcome) DESC
```

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

Out[36]:

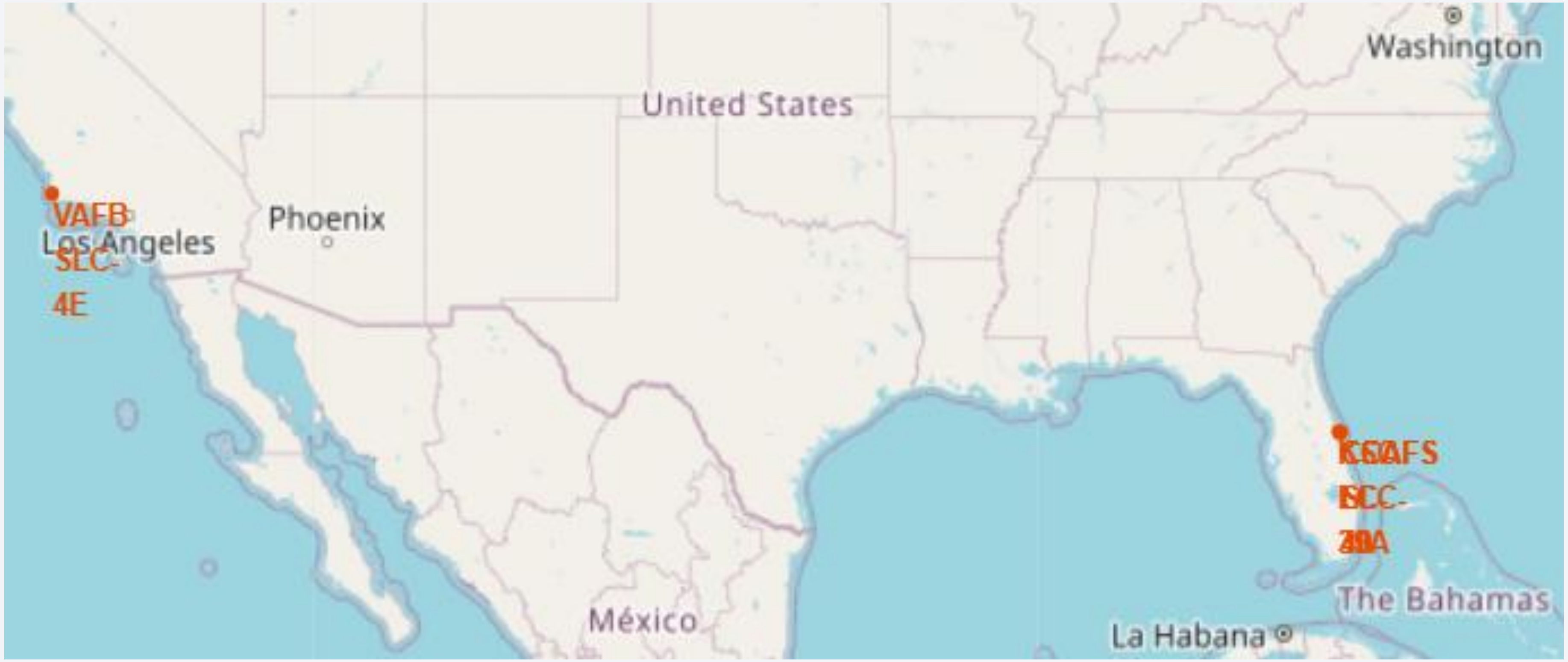
Landing_Outcome	COUNT(Landing_Outcome)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Explanation: The most common landing outcome was ‘No attempt’ during the period.

Section 3

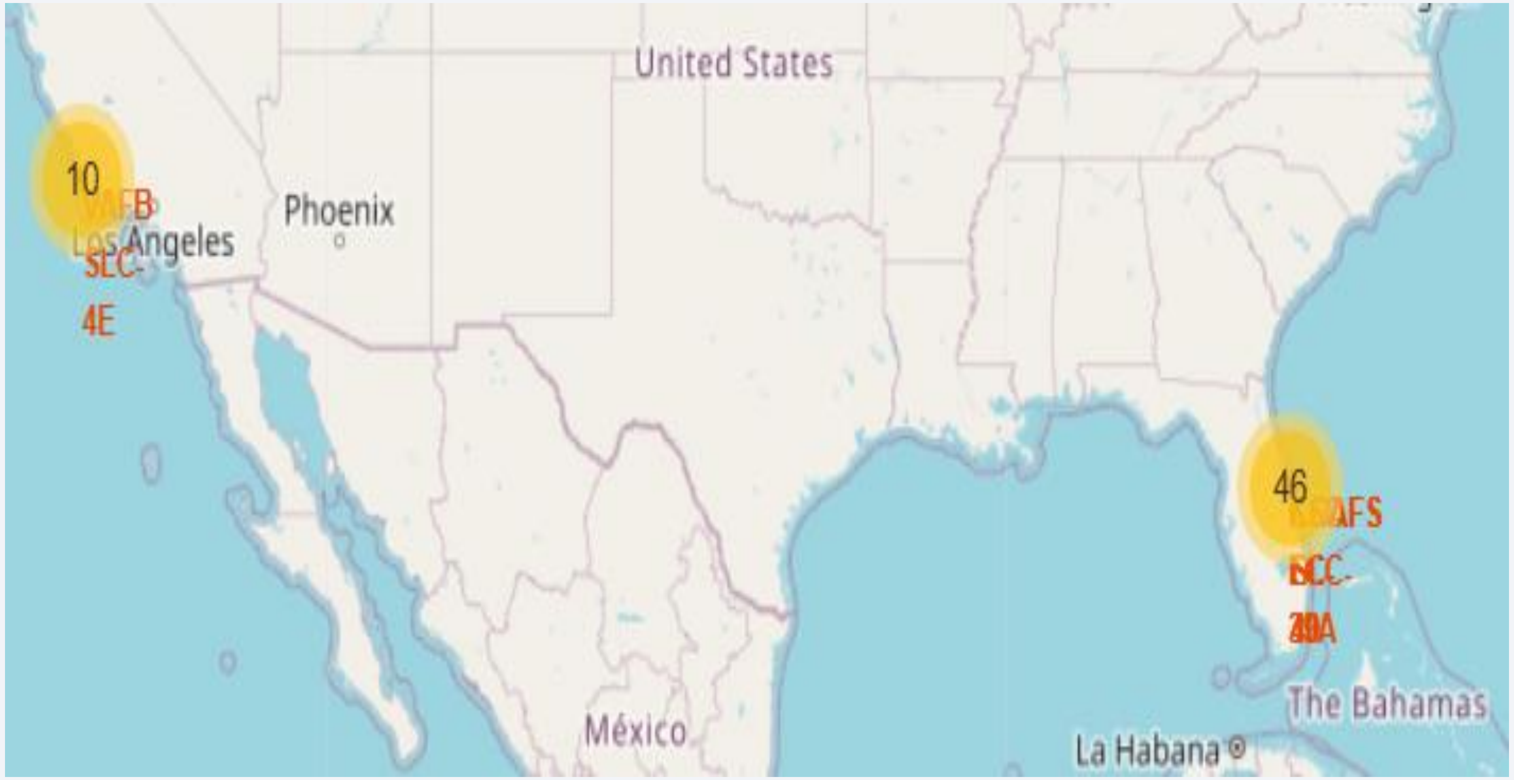
Launch Sites Proximities Analysis

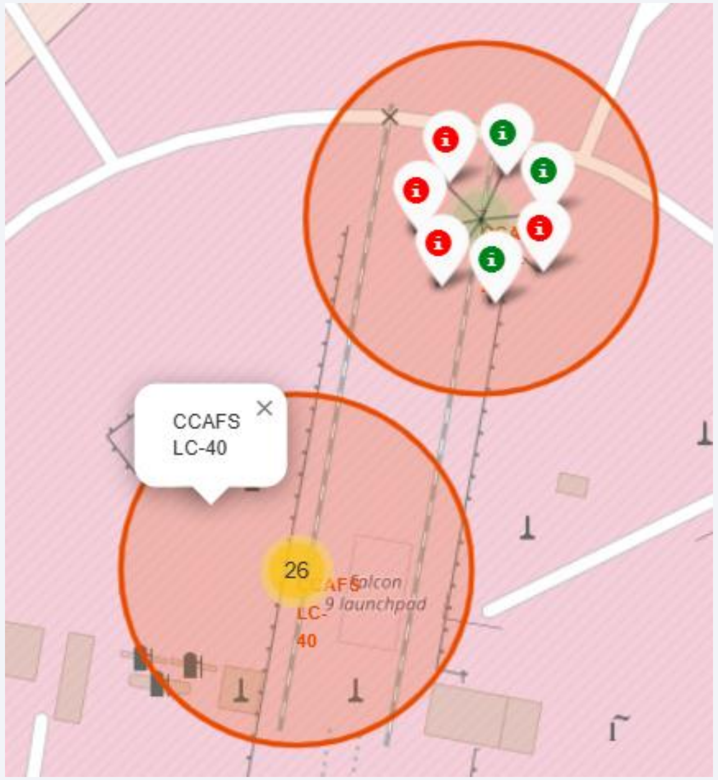
Launch Site Locations on the Map



- All launch sites are not in proximity to the Equator line.
- All launch sites are in very close proximity to the coast.

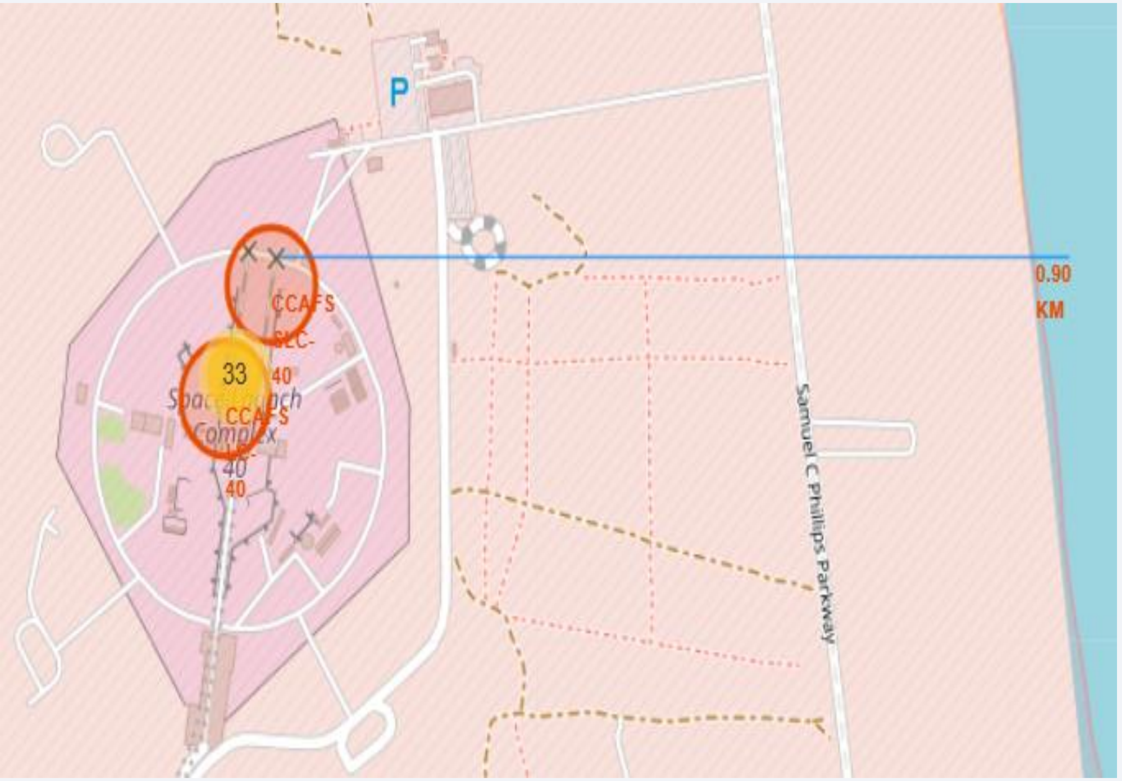
Launch Outcomes on the map





From the color-labeled launch outcomes on the map, we identified the CCAFS LC-40 launch site have relatively high success rates.

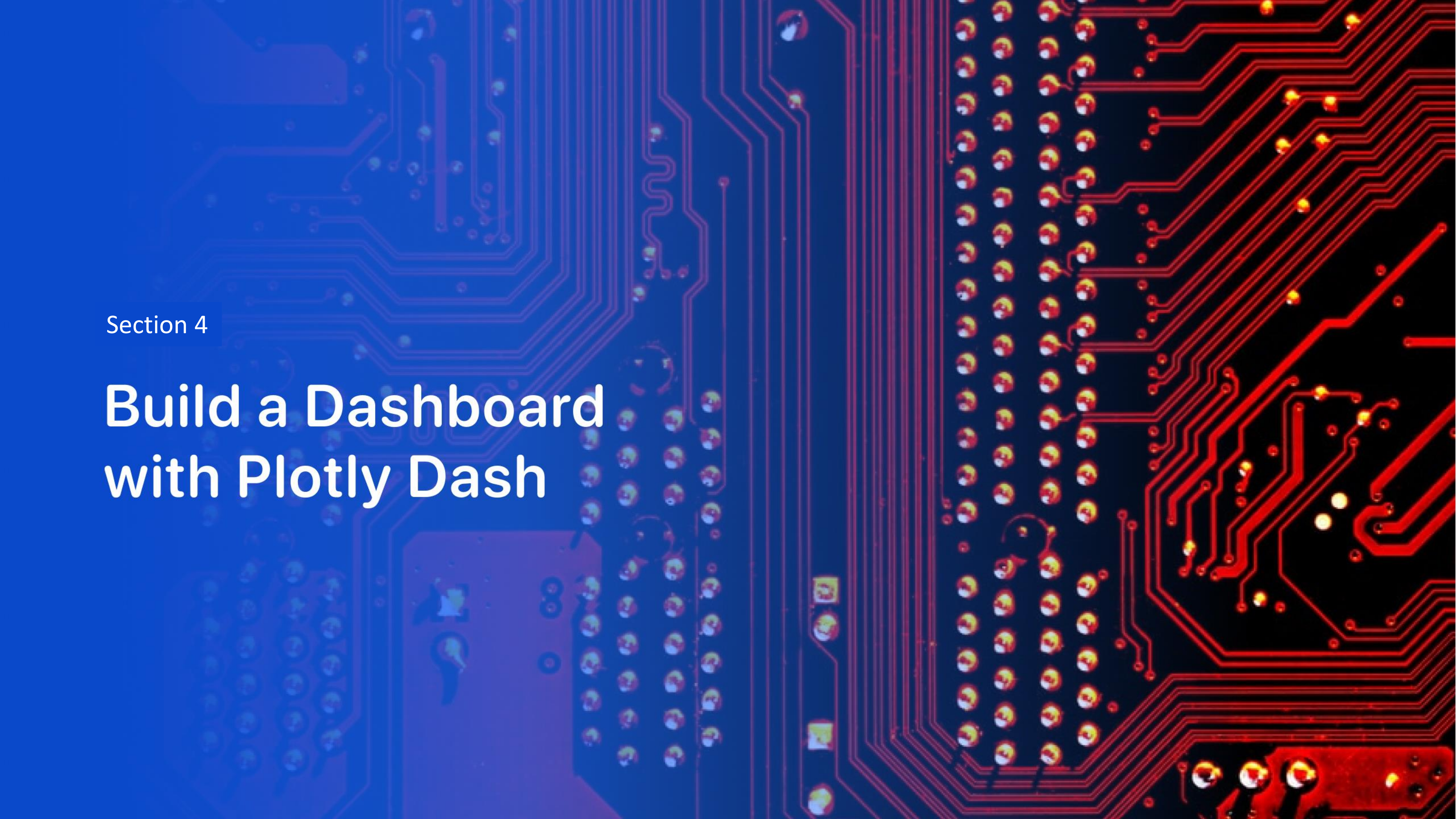
Proximity of CCAFS SLC-40 launch site to coastline, city



CCAFS SLC-40 launch site is in very close proximity to coastline. Specifically, the distance from the launch site to the nearest coastline is 0.9 KM only.

Church Street
78.45 KM
83
10A
11B
11A

CCAFS SLC-40 launch site is not in proximity to the City. Specifically, the distance from the launch site to the Florida City is 78.45 KM.

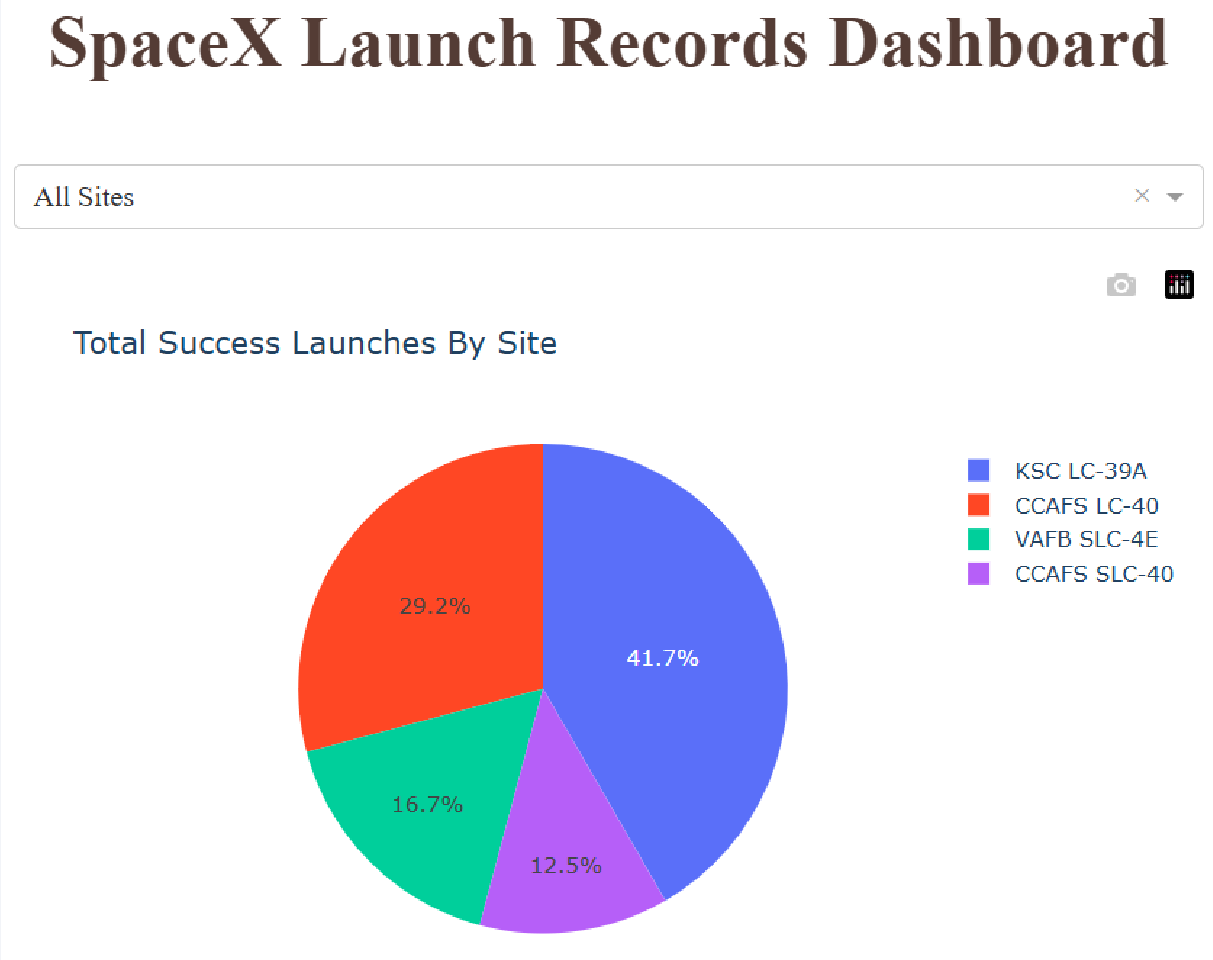


Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

- KSC LC-39A site has the most success launch, counted 41.7% in all sites.
- CCAFS SLC-40 has the least success launch, counted 12.5% in all sites.



SpaceX Launch Records Dashboard

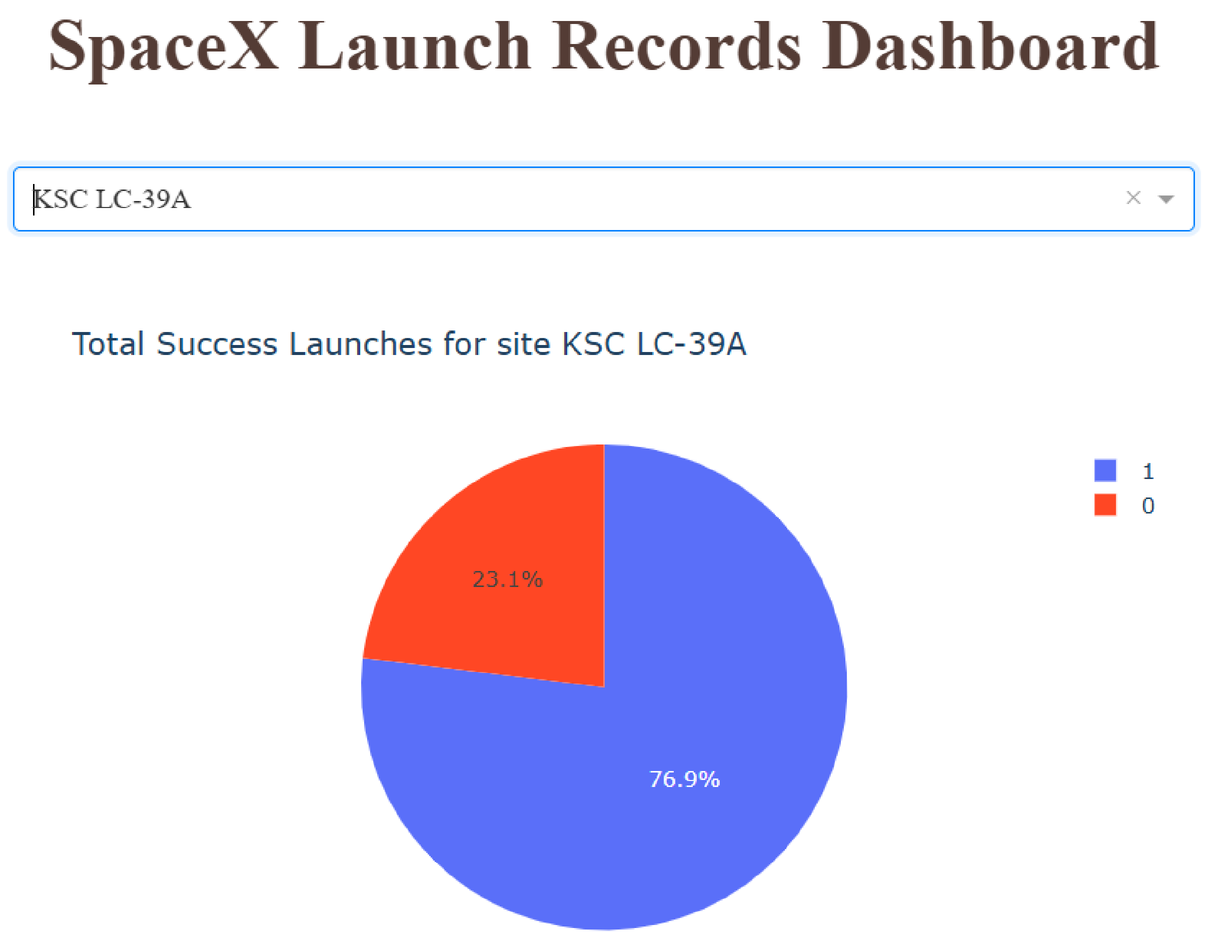
All Sites

Launch site with highest launch success ratio

When selecting specific launch site to see its launch success ratio, we found that KSC LC-39A site has the highest launch success ratio (76.9%).

Note:

- Class 1 in blue means success launch
- Class 0 in red means failed launch

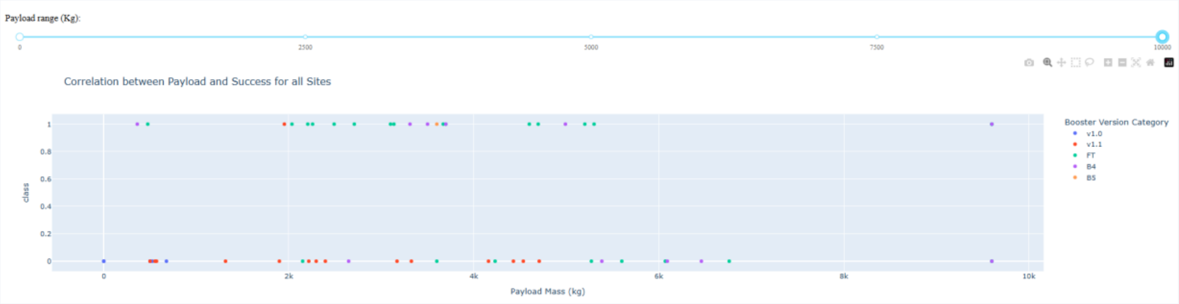


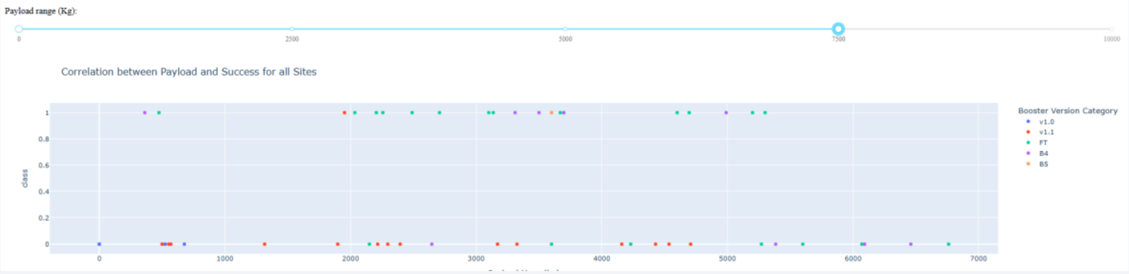
SpaceX Launch Records Dashboard

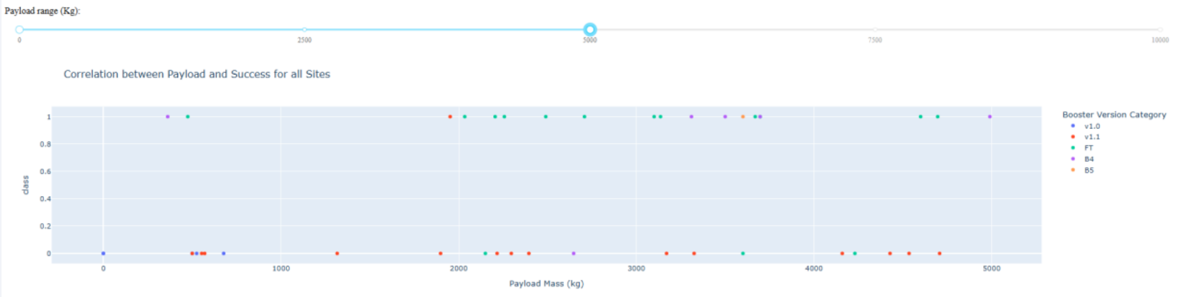
KSC LC-39A

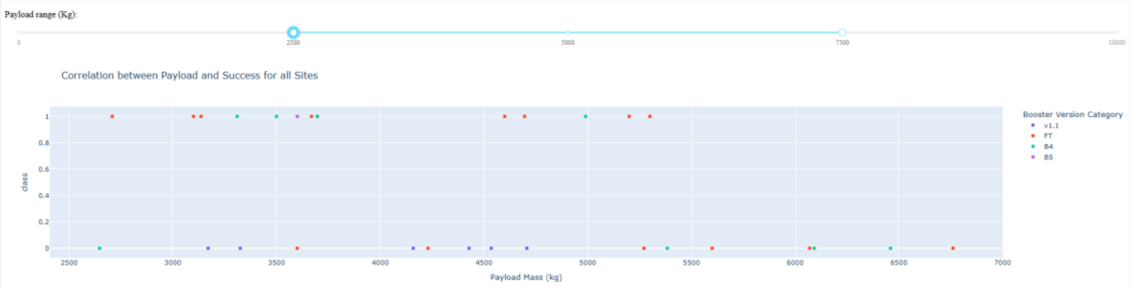
Total Success Launches for site KSC LC-39A

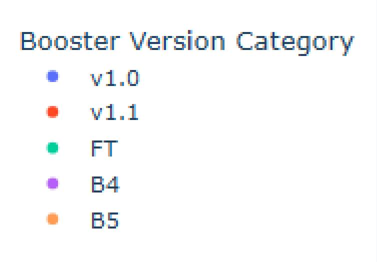
Payload vs. Launch Outcome











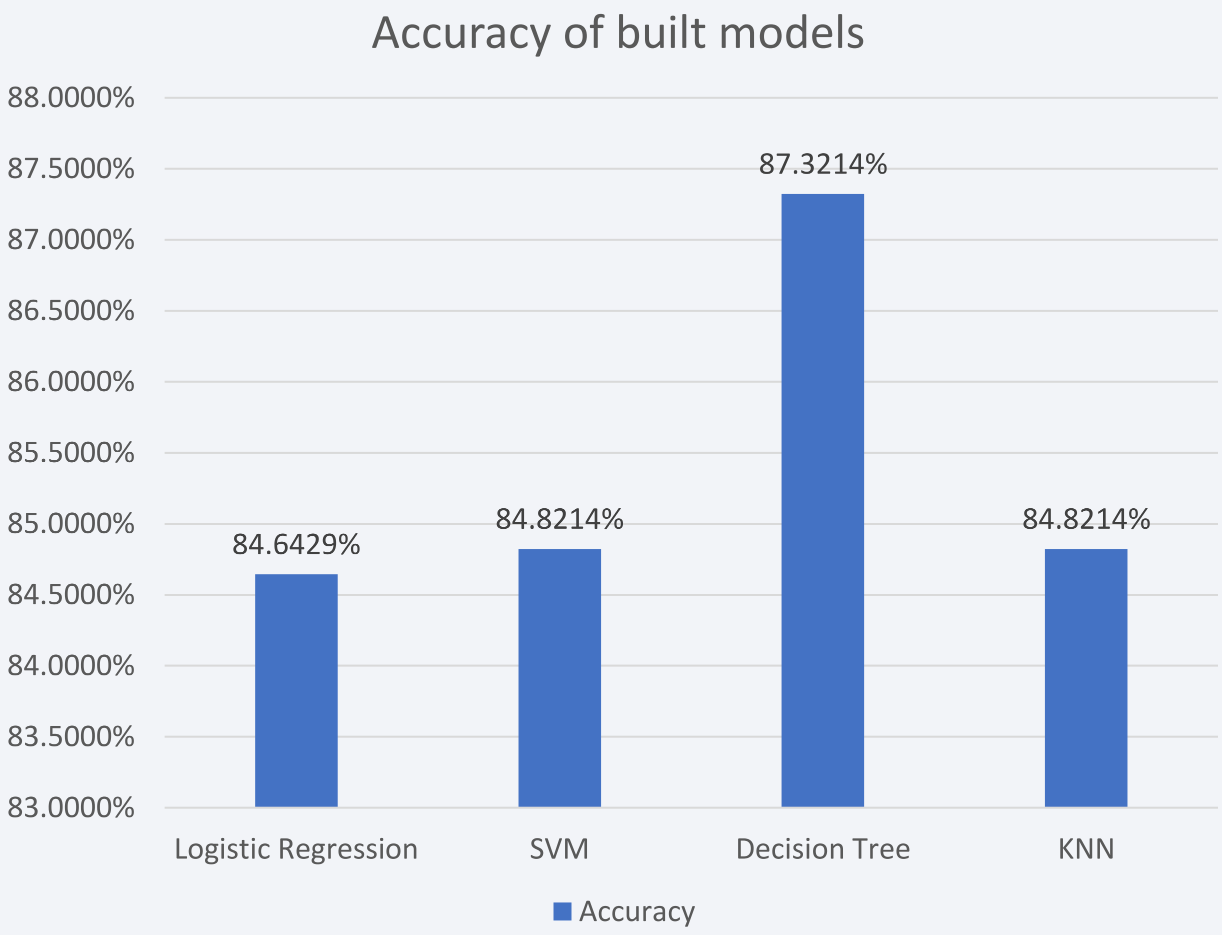
When selecting different payload range for all sites, we found that:

- Payload range from 0 – 7500 has the largest success rate
- Booster Version FT has the largest success rate, while Booster Version v1.1 has the most failed launch outcome

Section 5

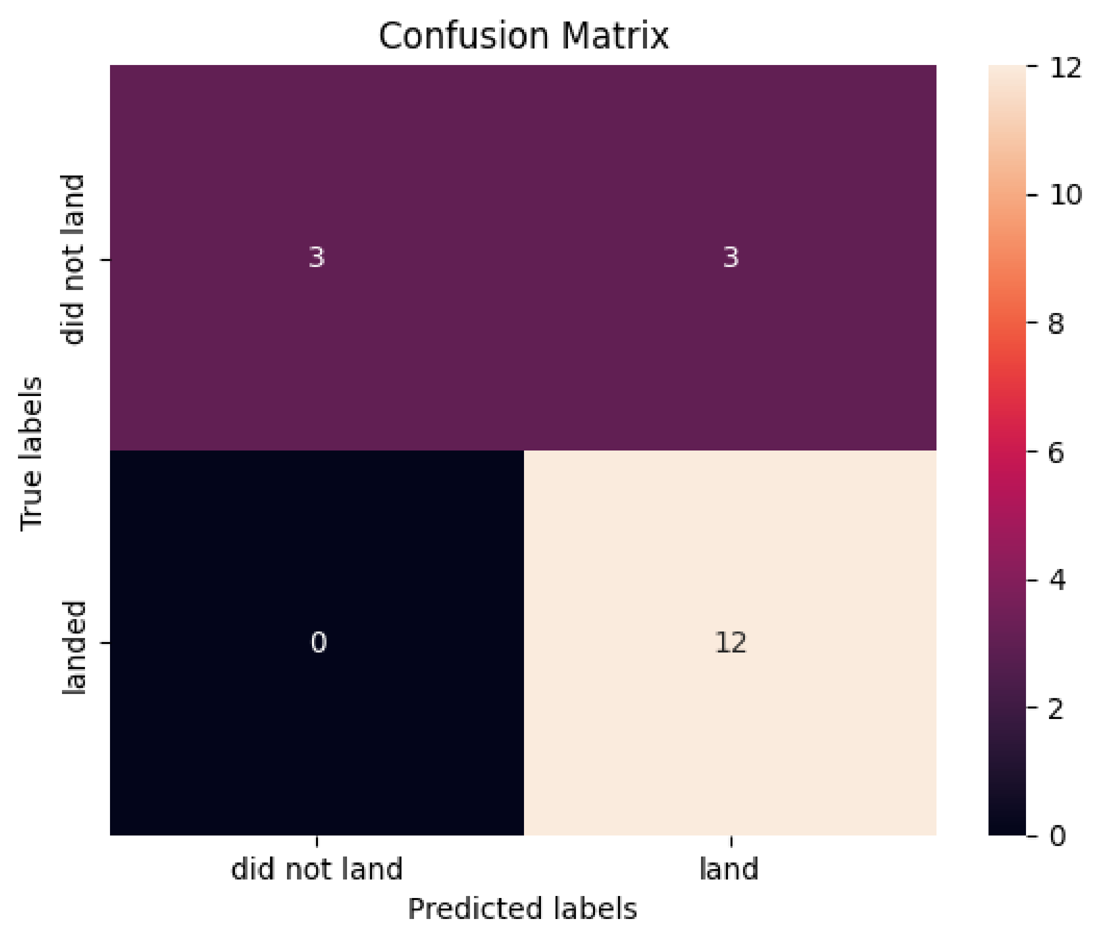
Predictive Analysis (Classification)

Classification Accuracy



As the bar chart, Decision Tree model has the highest classification accuracy (87.3214 %)

Confusion Matrix of the Best Model



- The Decision Tree model never predict incorrectly 'did not land'.
- However it sometimes predicts 'land' when it actually didn't (model predicted 'land' 12 instances, but it actually 'did not land' 3 instances).

Conclusions

- As the flight number increases, the first stage is more likely to land successfully.
- Payload range from 0 – 7500 has the largest success rate.
- Booster Version FT has the largest success rate.
- Orbits ES-L1, GEO, HEO and SSO has the most success rate.
- The launch success rate since 2014 was increasing till 2020, except 2018.
- All launch sites are in very close proximity to the coast.
- CCAFS LC-40 launch site have relatively high success rates.
- The Decision tree classifier is the best machine learning model for this task.

Thank you!