IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

By Tue Thu
2026/01/09

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusions

Executive Summary

The objective of this project is twofold:

1. To evaluate the economic viability of each SpaceX launch and determine whether reusing the first-stage booster is necessary by training a machine learning model using publicly available data.
2. Based on the above results, make SpaceY more affordable to compete with SpaceX.

Methodologies: The project followed the full data science lifecycle, including:

- Data collection through API and web scraping
- Data Wrangling
- Exploratory Data Analysis (EDA)
- Interactive Visual Analytics and Dashboards with Folium and Plotly Dash
- Predictive Analysis (classification)

Results

Among the 4 tested algorithms—Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN)—the Logistic Regression model achieved the highest accuracy at 93% for the whole dataset

Introduction

- SpaceX has transformed the aerospace industry through the development of its reusable Falcon 9 launch system. It can cut launch costs from more than $165 million to roughly $62 million. This breakthrough in reusability delivers significant economic and technological benefits.

- Accurately predicting the landing success of the Falcon 9 first stage is crucial for mission planning and cost estimation, and it provides valuable insights for stakeholders involved in space operations, logistics, and investment.

- This project aims to develop a machine learning model that forecasts landing outcomes by gathering data from an API, specifically the SpaceX REST API. By applying the complete data science workflow—data acquisition, preprocessing, exploratory analysis, and predictive modeling—we identify the key drivers of successful landings and deliver a practical, data-driven tool for the aerospace sector.

Section 1

Methodology

Methodology

- Data collection methodology:

Data was collected through requests to the SpaceX API and additional web scraping from Wikipedia.

- Perform data wrangling

The collected data is saved in JSON form ad HTML tables

After that, Pandas and NumPy were used to clean, organize, and manage the dataset for further visualization and analysis

- Perform exploratory data analysis (EDA) using visualization and SQL.
- Perform interactive visual analytics using Folium and Plotly Dash.
- Perform predictive analysis using classification models

Several classification models were tested, with K-Nearest Neighbors (KNN) achieving the lowest performance at 77.8% accuracy.

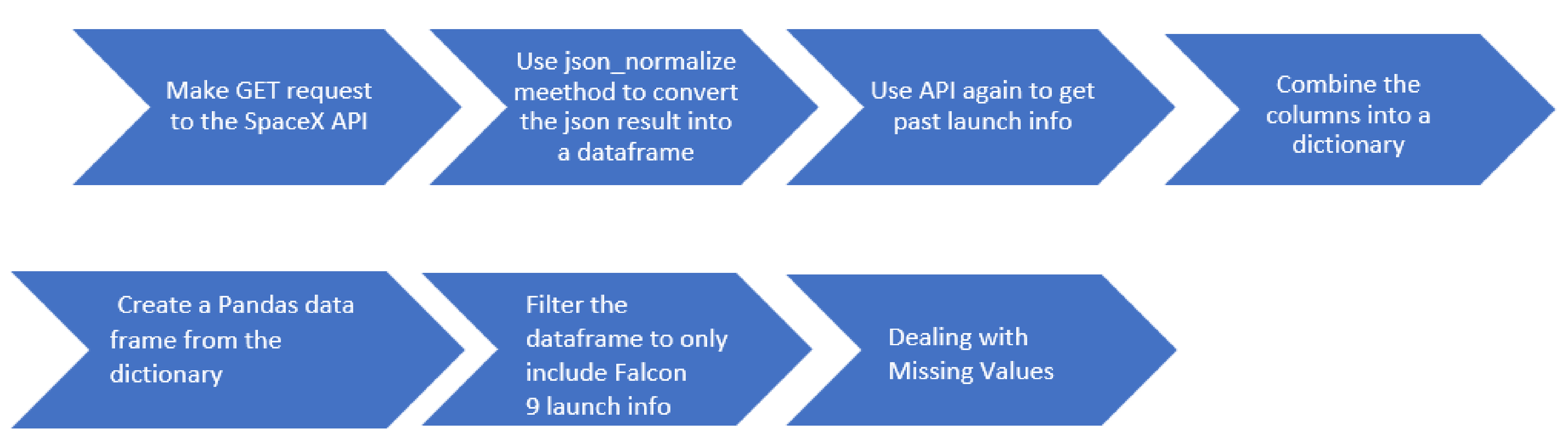
Data Collection

The data for the following final project was collected using two different techniques.

1. Request and parse the SpaceX launch data using the GET request

In this method, we only include Falcon 9 launches, use the mean for the for the PayloadMass then replace the missing data (np.nan values) with the mean we calculated.

Link to the notebook



Make GET request to the SpaceX API

Use json_normalize meethod to convert the json result into a dataframe

Use API again to get past launch info

Combine the columns into a dictionary

Create a Pandas data frame from the dictionary

Filter the dataframe to only include Falcon 9 launch info

Dealing with Missing Values

Data Collection – SpaceX API

2. Web scraping with BeautifulSoup

2.1. Request the Falcon9 Launch Wiki page from the below URL to extract and compile to the launch records HTML table
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

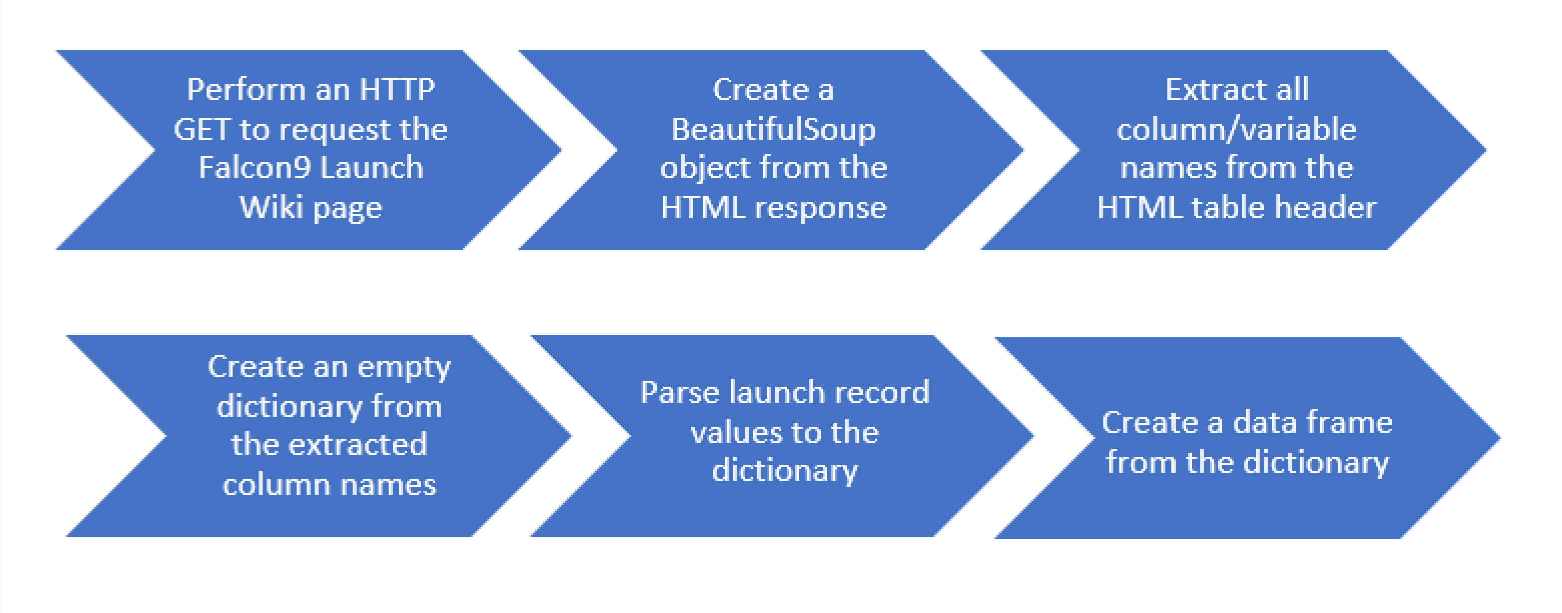
2.2. Extract (Parse) all column/variable names from the HTML table header and convert it into a Pandas data frame

The full step-by-step extraction procedure can be found at:
Link to the notebook

Data Collection - Scraping

In this project, we used BeautifulSoup to build a web archive of Falcon 9 launch logs and extracted the corresponding HTML table from Wikipedia. The table's columns were then scraped and processed using web-scraping techniques to construct a clean Pandas DataFrame, as shown below:

Link to the notebook



Perform an HTTP GET to request the Falcon9 Launch Wiki page

Create a BeautifulSoup object from the HTML response

Extract all column/variable names from the HTML table header

Create an empty dictionary from the extracted column names

Parse launch record values to the dictionary

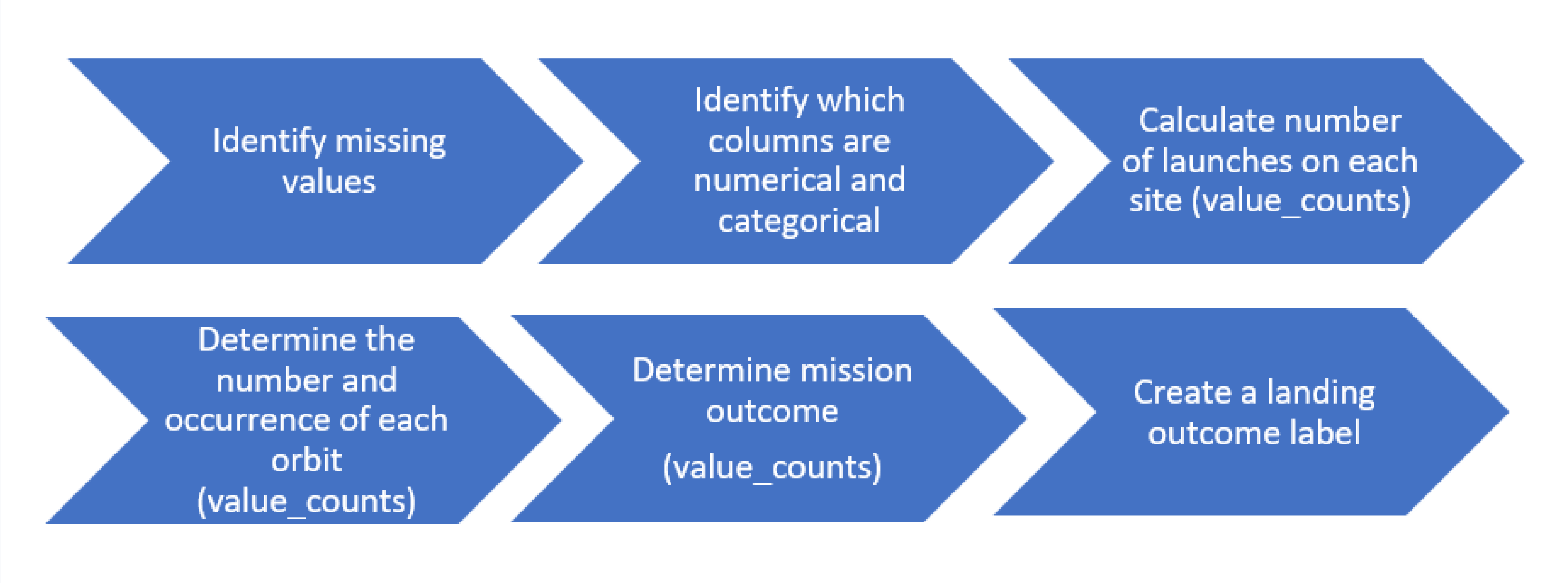
Create a data frame from the dictionary

Data Wrangling

We examined the dataset to identify data types and check for any missing, duplicate, or null values. This process followed the workflow illustrated in the flowchart below.

You can also review the full procedure at:

Link to the notebook


Identify missing values
Identify which columns are numerical and categorical
Calculate number of launches on each site (value_counts)
Determine the number and occurrence of each orbit (value_counts)
Determine mission outcome
(value_counts)
Create a landing outcome label

EDA with Data Visualization

For this analysis, we created three types of visualizations using Pandas, Matplotlib, and Seaborn:

1. **Scatter plots** to explore relationships between:

- Flight Number and Launch Site
- Payload Mass and Launch Site
- FlightNumber and Orbit type
- Payload Mass and Orbit type

2. **Bar charts** to reveal relationship between success rate of each orbit type

3. **Line charts** to identify trends over time such as launch success yearly trend

The complete analysis and all visualizations can be found here:
Link to the notebook

EDA with SQL

The **SPACEXTABLE** dataset was generated by filtering out records with zero or missing dates. From this table, we identified all unique launch sites, calculated the total payload delivered for NASA CRS missions, and computed the average payload for the Falcon 9 v1.1 variant. We also determined the date of the first successful landing, counted both successful and failed landing attempts, and identified the booster version that carried the heaviest payload. In addition, all pad landing failures that occurred in 2015 were extracted along with their relevant details.

Below are some SQL queries that have been performed:

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

```
%%sql SELECT Launch_Site FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)’;
```

The full, step-by-step process can be found here:
Link to the notebook

Build an Interactive Map with Folium

The map was constructed using several key Folium components:

- folium.Map for generating the interactive base map
- folium.Circle to highlight specific locations with informative circular overlays
- folium.Marker to pinpoint launch sites
- folium.MarkerCluster to group nearby markers and reduce visual clutter
- MousePosition to display real-time geographic coordinates as the cursor moves
- folium.DivIcon to place custom HTML/CSS-styled text directly on the map
- folium.PolyLine to draw connecting lines between locations
- folium.FeatureGroup to organize related map elements into manageable layers

Link to the notebook

Build a Dashboard with Plotly Dash

The Dash web application was built using the following components:

- **dcc.Dropdown** to select a launch site and display the corresponding success-versus-failure distribution in a pie chart
- **dcc.RangeSlider** to filter payload values and examine how payload size relates to launch success
- **dcc.Graph** to render both the pie chart and scatter plot
- **Dash callbacks** to enable dynamic, interactive updates between all interface elements

The visualizations included:

- **Pie Chart**, illustrating the proportion of successful and failed launches for each site
- **Scatter Plot**, showing the relationship between payload mass and launch outcome

Predictive Analysis (Classification)

We conducted the following tasks to predict if the first stage will land or not.

- Load the dataframe
- Create a NumPy array from the column Class in data
- Standardize the data in X then reassign it to the variable X
- Use the function train_test_split to split the data X and Y into training and test data
- Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10
- Seach the best Hyperparameters for Logistic Regression, SVM, Decision Tree and KNN classifiers
- Calculate the accuracy of each model and find out the best model

The complete process is available at:
Link to the notebook

Results

- **Exploratory Data Analysis Results:**

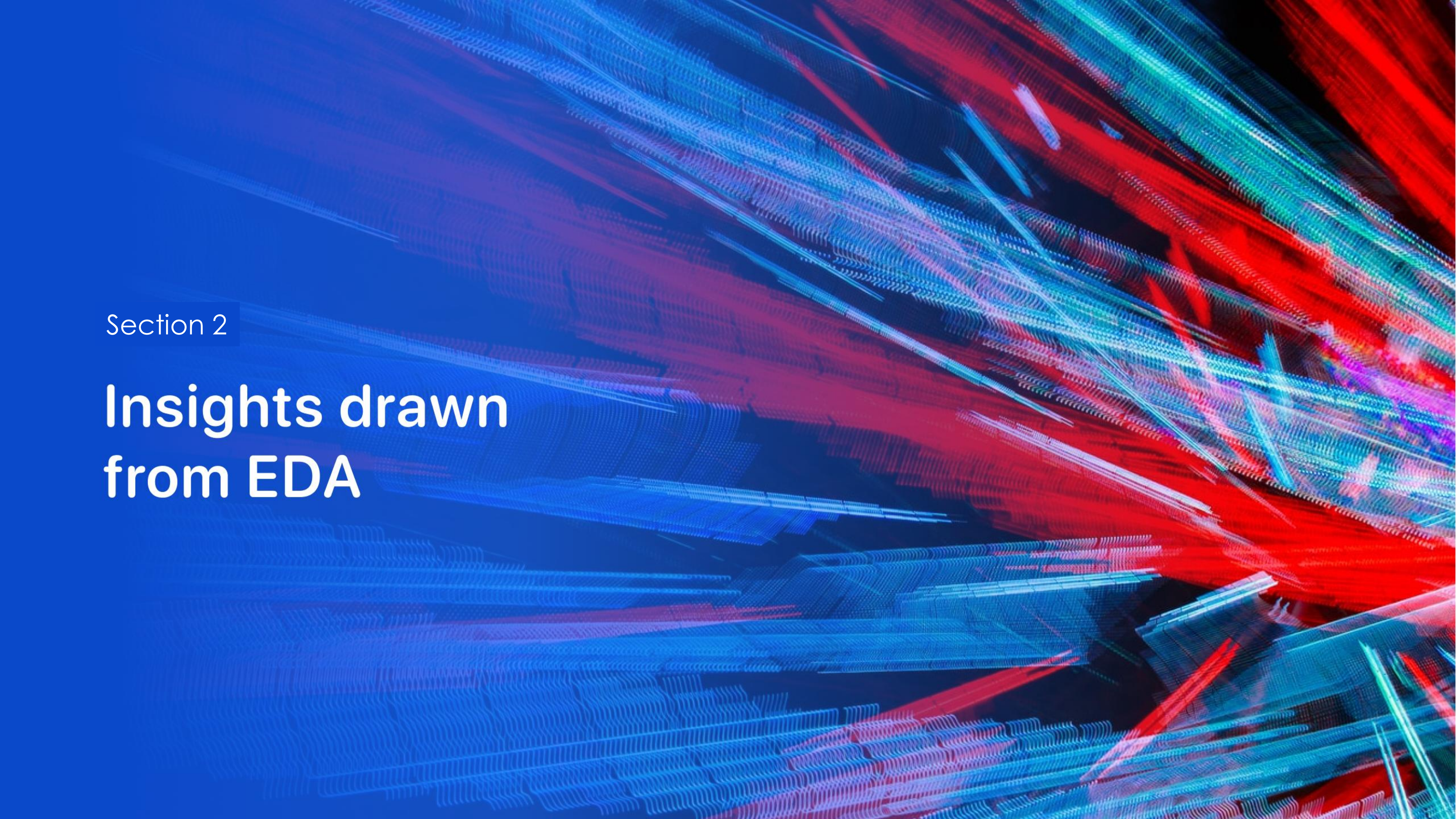
The analysis confirmed that several key parameters have a direct impact on whether a landing succeeds or fails.

- **Interactive Analysis (Screenshots):**

The visual exploration clearly showed how the choice of launch site influences the overall mission success rate.

- **Predictive Analysis Results:**

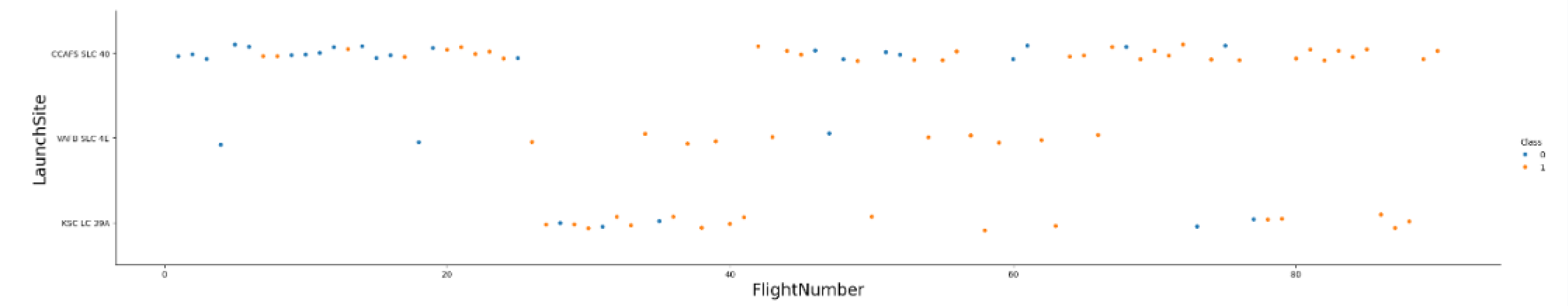
A Logistic Regression model was developed, achieving an accuracy of 93% in predicting the success of SpaceX first-stage landings.



Section 2

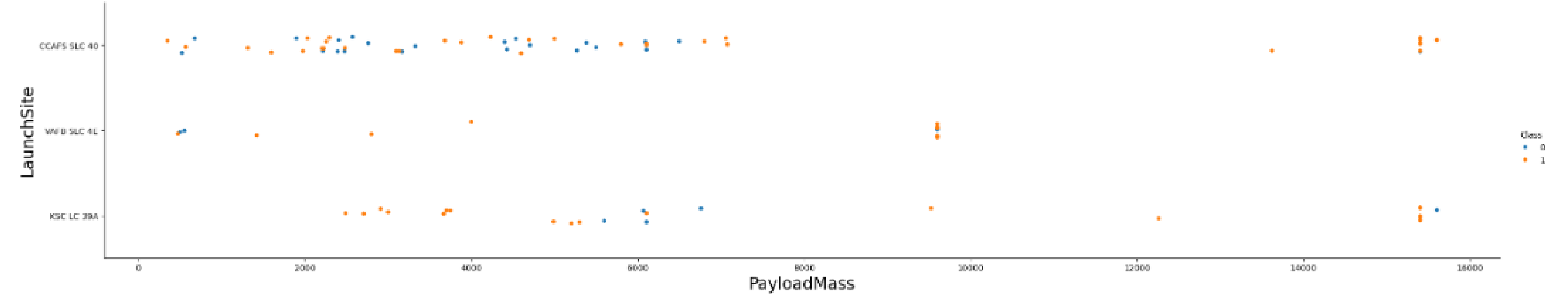
Insights drawn from EDA

Flight Number vs. Launch Site



The visualization reveals a clear trend: as flight numbers increase, success rates improve across all launch sites.
CCAFS SLC-40 stands out with significantly higher success rates compared to VAFB SLC-4E and KSC LC-39A.

Payload vs. Launch Site

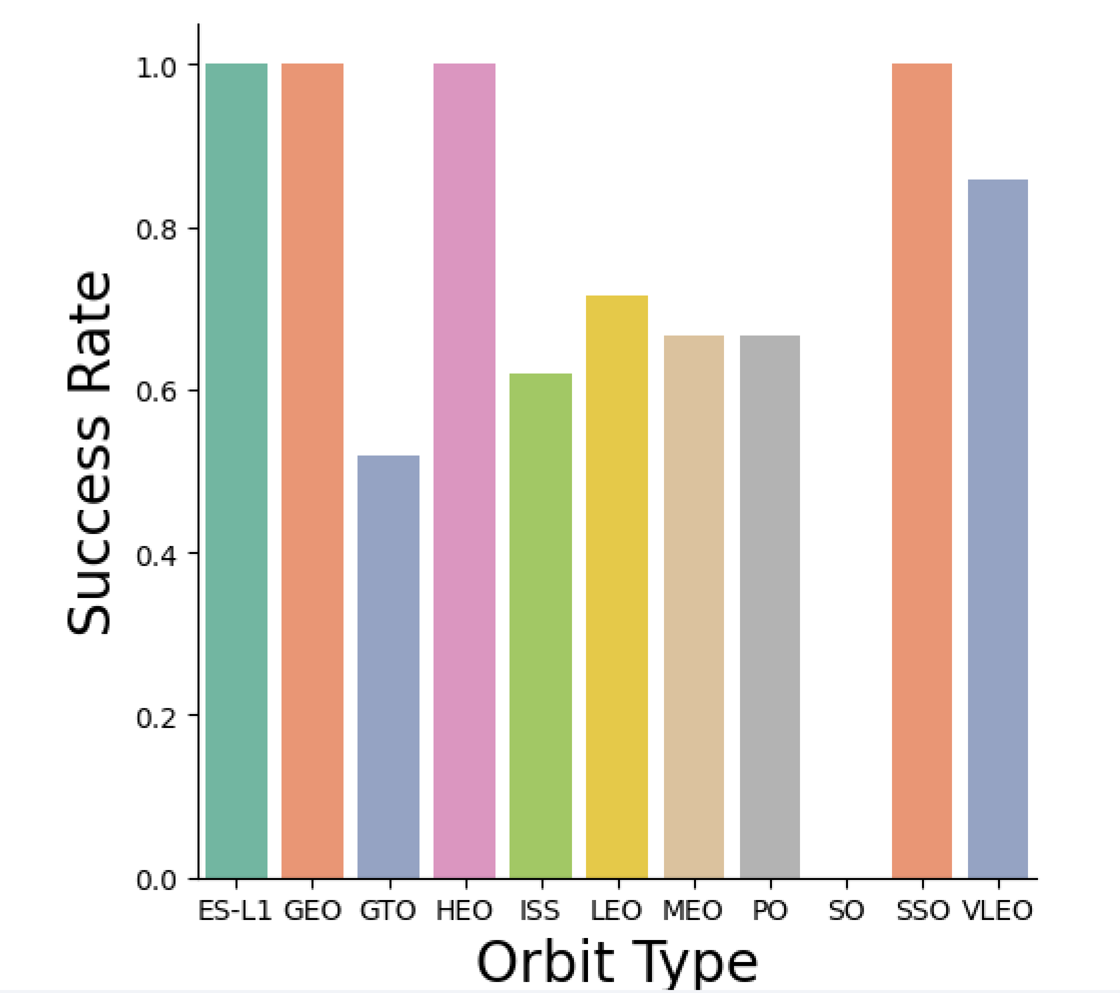


The graph highlights the differences in payload mass across the three launch sites.

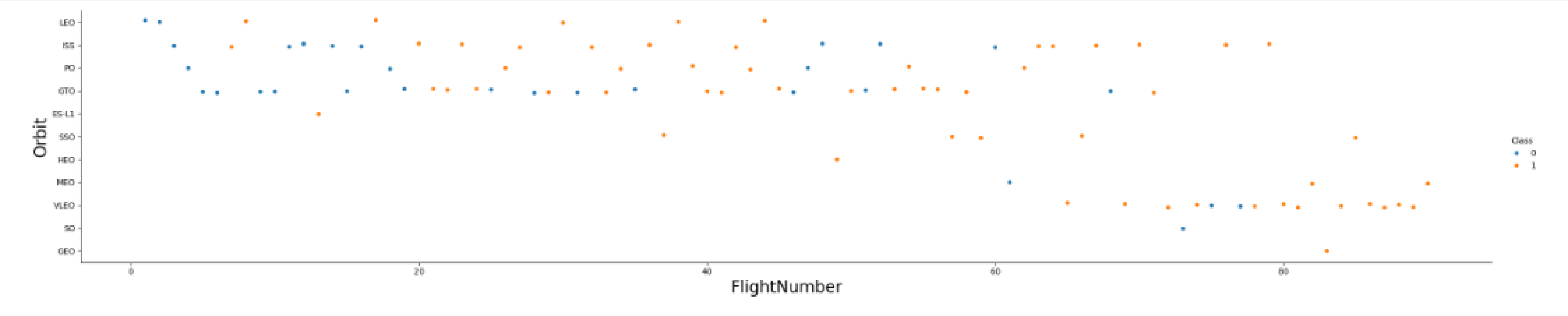
- CCAFS SLC-40 has launched rockets carrying payloads below 7,000 kg and above 13,000 kg, with no launches recorded in the intermediate range.
- VAFB SLC-4E has only launched rockets with payload masses under 10,000 kg.
- KSC LC-39A has exclusively launched rockets with payloads greater than 2,500 kg.

Success Rate vs. Orbit Type

The data indicates that the chosen orbit has a strong influence on mission success, with certain orbits performing considerably better than the rest such as ES-L1, GEO, HEO and SSO.

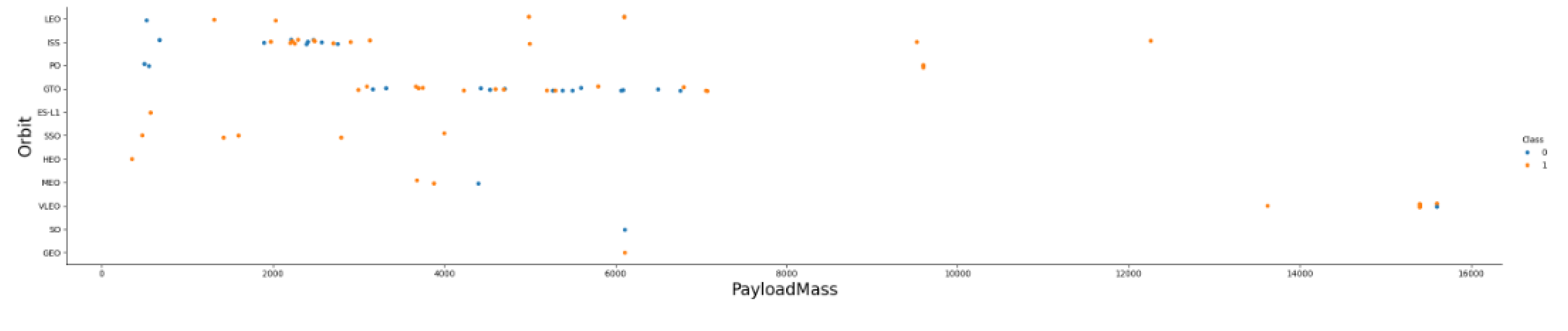


Flight Number vs. Orbit Type



- The visualization reveals a clear pattern: as the number of flights increases, success rates rise across all orbits. Notably, after about 40 launches, the proportion of successful landings improves by more than 50%.
- GTO and ISS missions show more failures than successful landings when compared with other orbits.
- In contrast, SSO stands out with a perfect success rate—although this is based on only four launches, so the sample size is limited.

Payload vs. Orbit Type

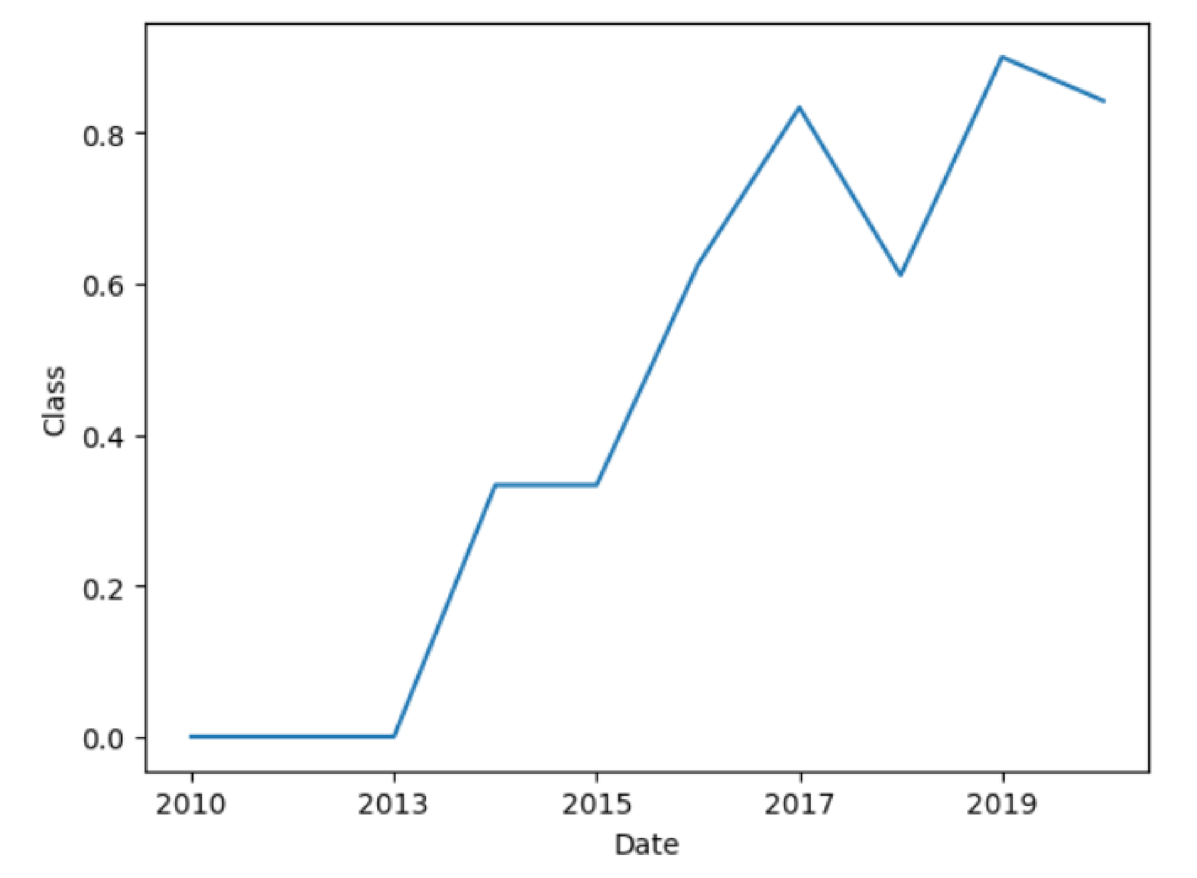


- A large portion of the payload masses across different orbits falls below 8,000 kg.
- The ISS orbit shows consistently higher success rates across its range of payload masses.
- In contrast, GTO missions display roughly a 50% success rate regardless of payload mass.

Launch Success Yearly Trend

This graph showcases SpaceX's remarkable improvement in rocket reusability:

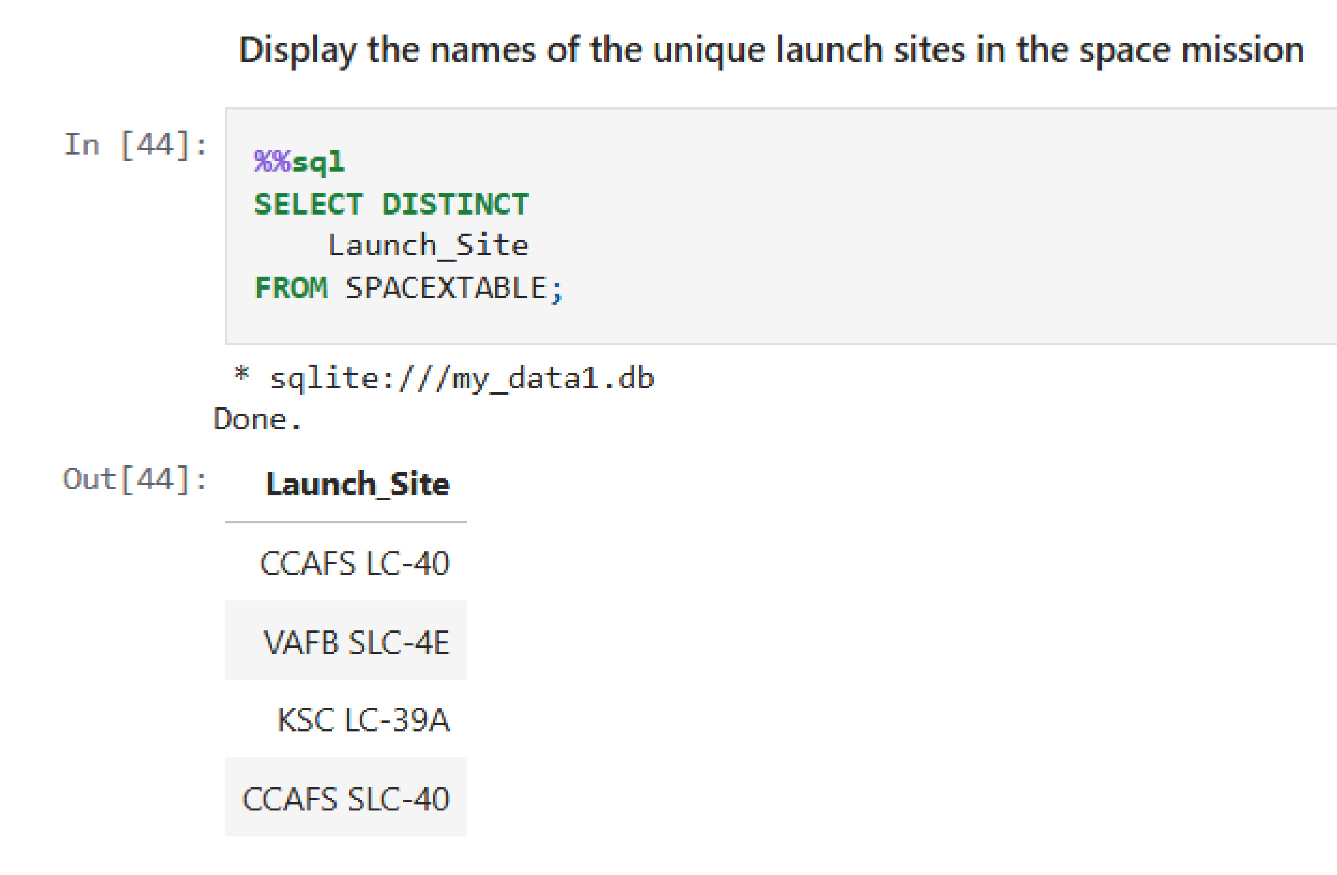
- Early attempts had a 0% success rate, but by 2017 the company surpassed—and maintained—an 85% success rate.
- The success rate from 2013 kept increasing until 2020
- The trend demonstrates ongoing learning, stronger engineering, and continuous innovation.



All Launch Site Names

This query retrieves and visualizes the four launch locations contained in our dataset, which are:

1. CCAFS LC-40
2. VAFB SLC-4E
3. KSC LC-39A
4. CCAFS SLC-40



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

In [44]:

```
%%sql
SELECT DISTINCT
    Launch_Site
FROM SPACEXTABLE;
```

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

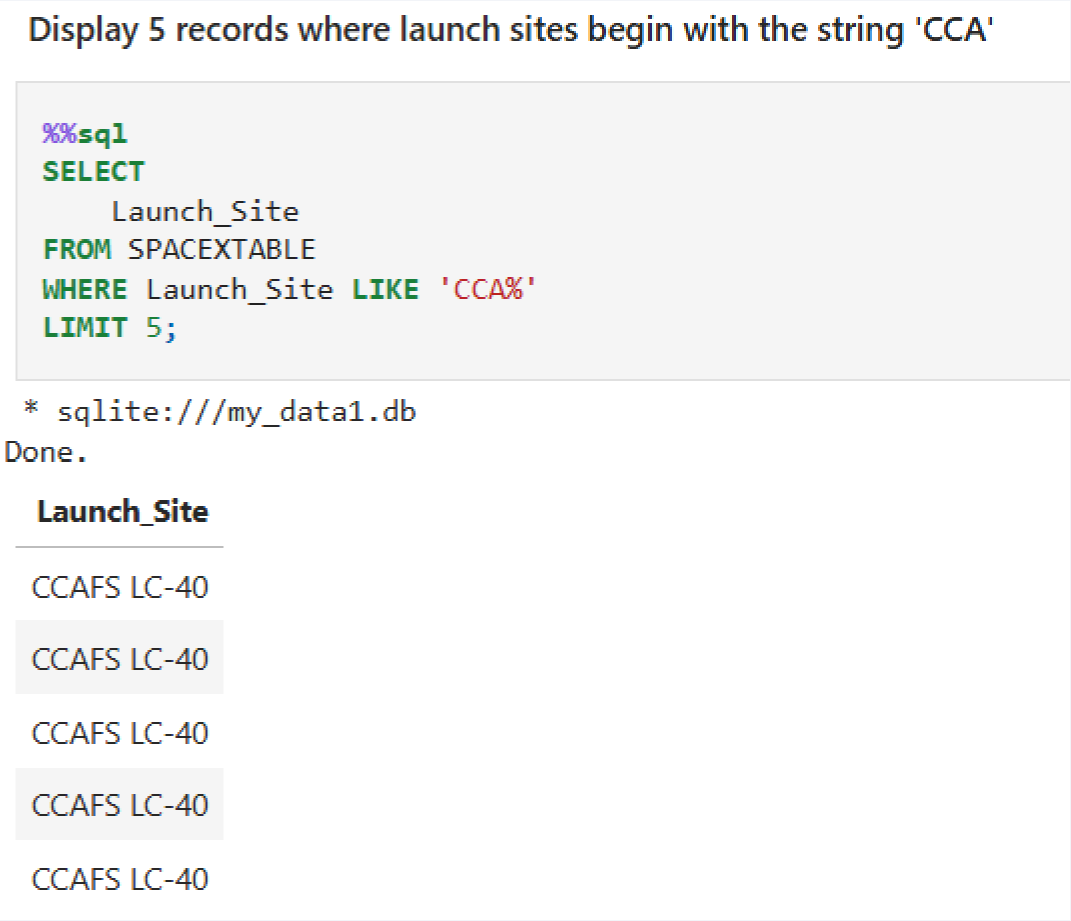
Out[44]:

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

Launch Site Names Begin with 'CCA'

This query allowed us to view the first five records for the CCAFS LC-40 launch site.

1. CCAFS LC-40
2. CCAFS LC-40
3. CCAFS LC-40
4. CCAFS LC-40
5. CCAFS LC-40



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

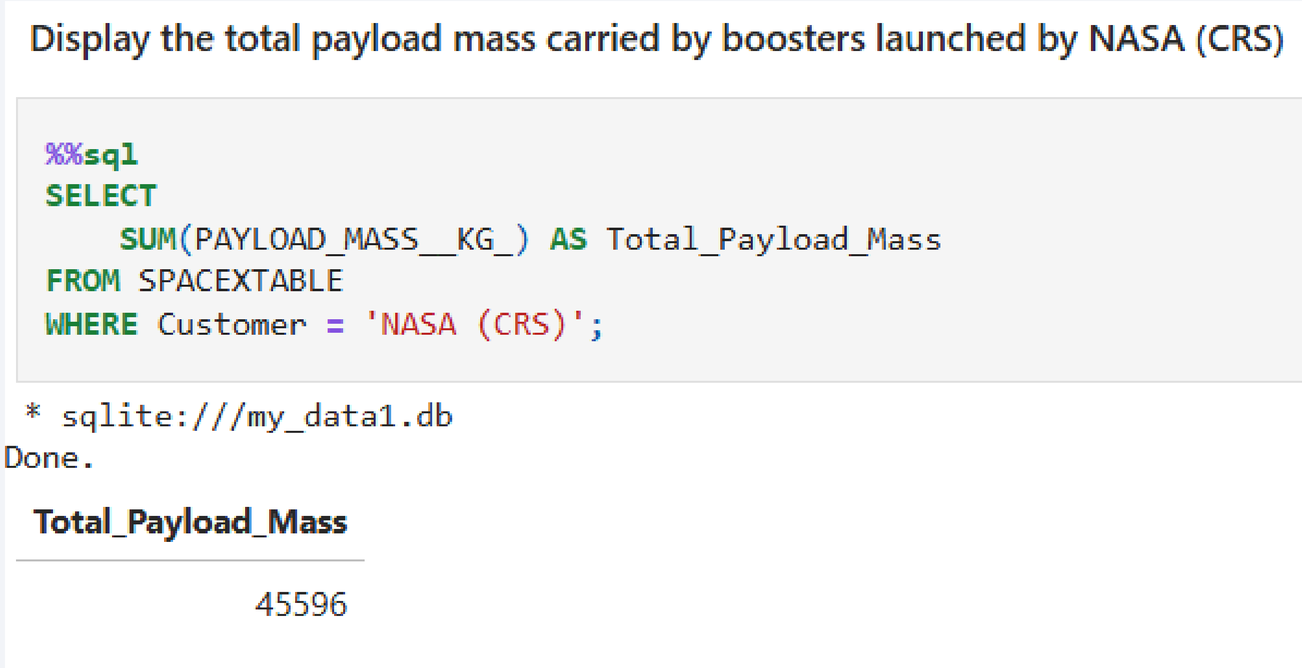
```
%%sql
SELECT
    Launch_Site
FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

The SQL query displayed in the image returns the total payload weight across all missions, yielding a value of 45,596 kg.



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

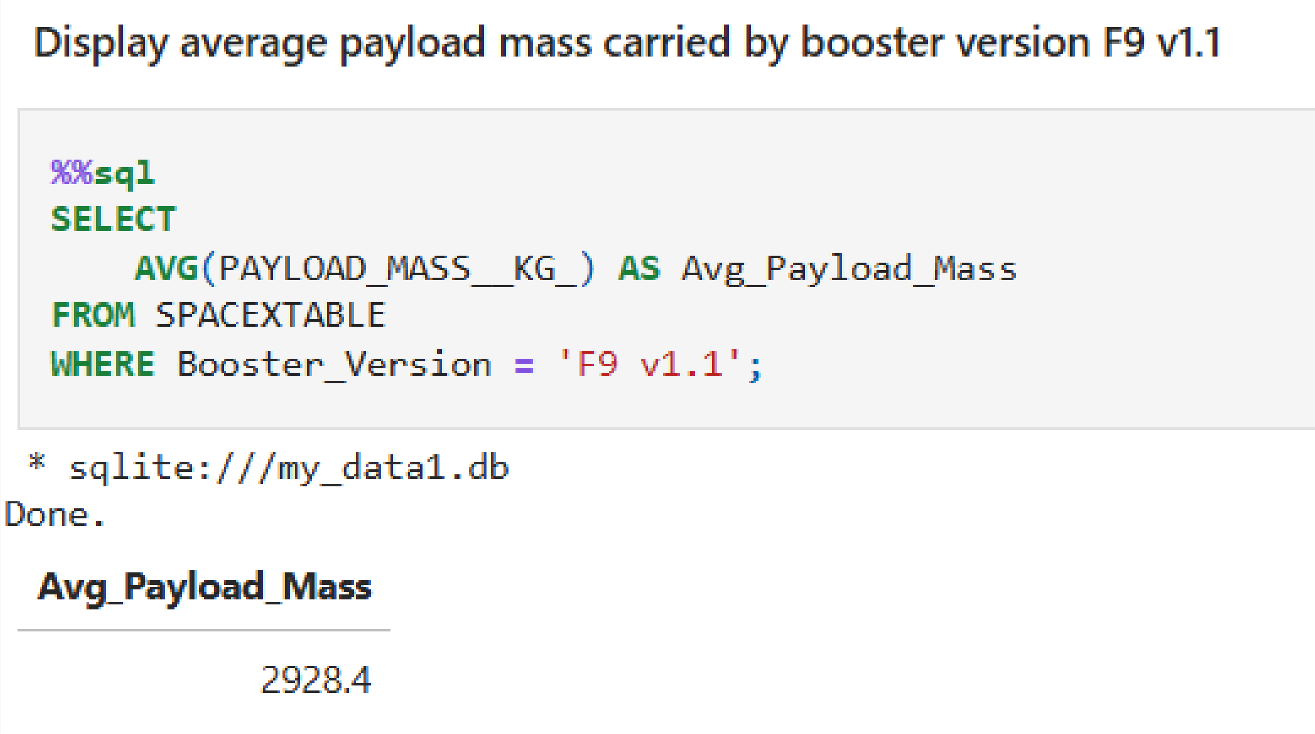
```
%%sql
SELECT
    SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

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

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

The SQL query also provides the average payload weight per launch, which is 2928.4 kg.



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

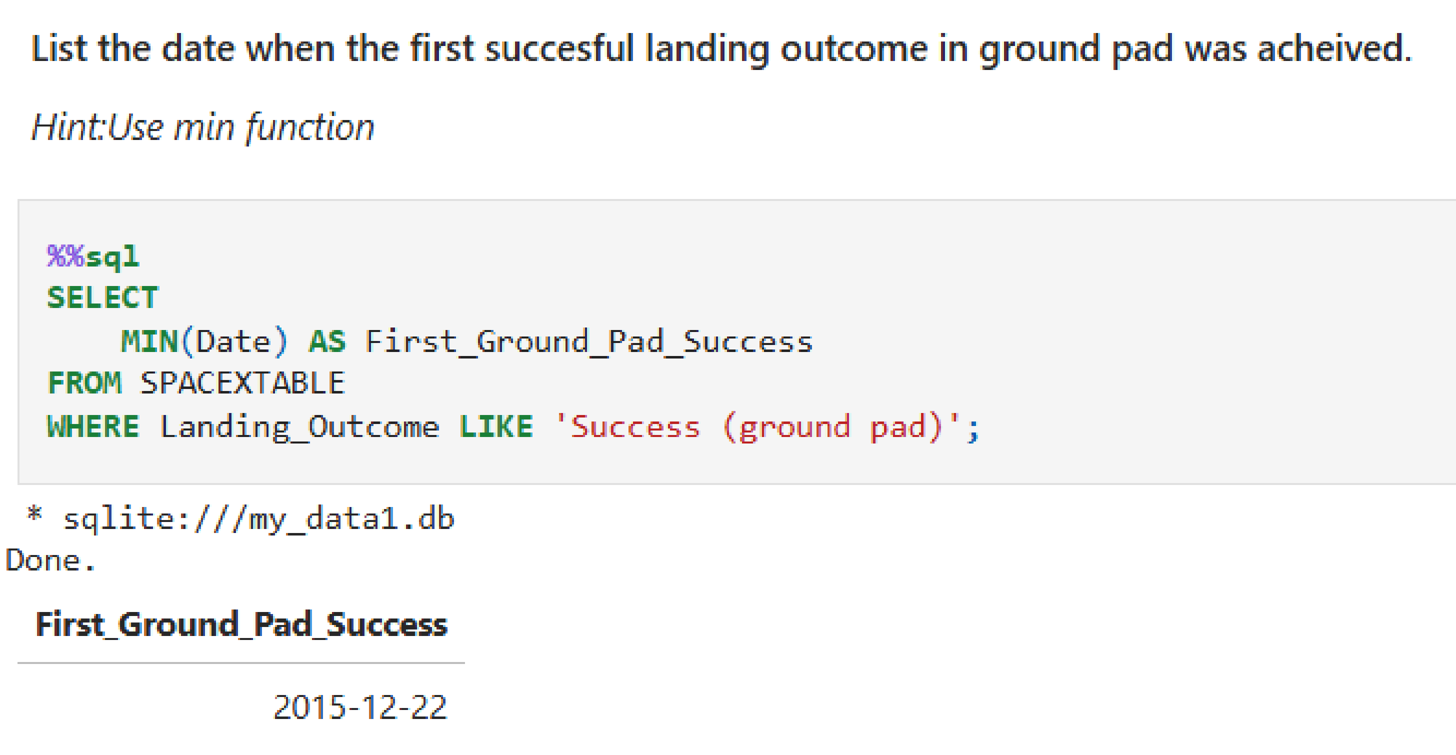
```
%%sql
SELECT
    AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_Mass
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';
```

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

Avg_Payload_Mass
2928.4

First Successful Ground Landing Date

From the SQL query displayed, we found that the earliest successful landing on the platform took place on 2015-12-22.



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

Hint:Use min function

```
%%sql
SELECT
    MIN(Date) AS First_Ground_Pad_Success
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

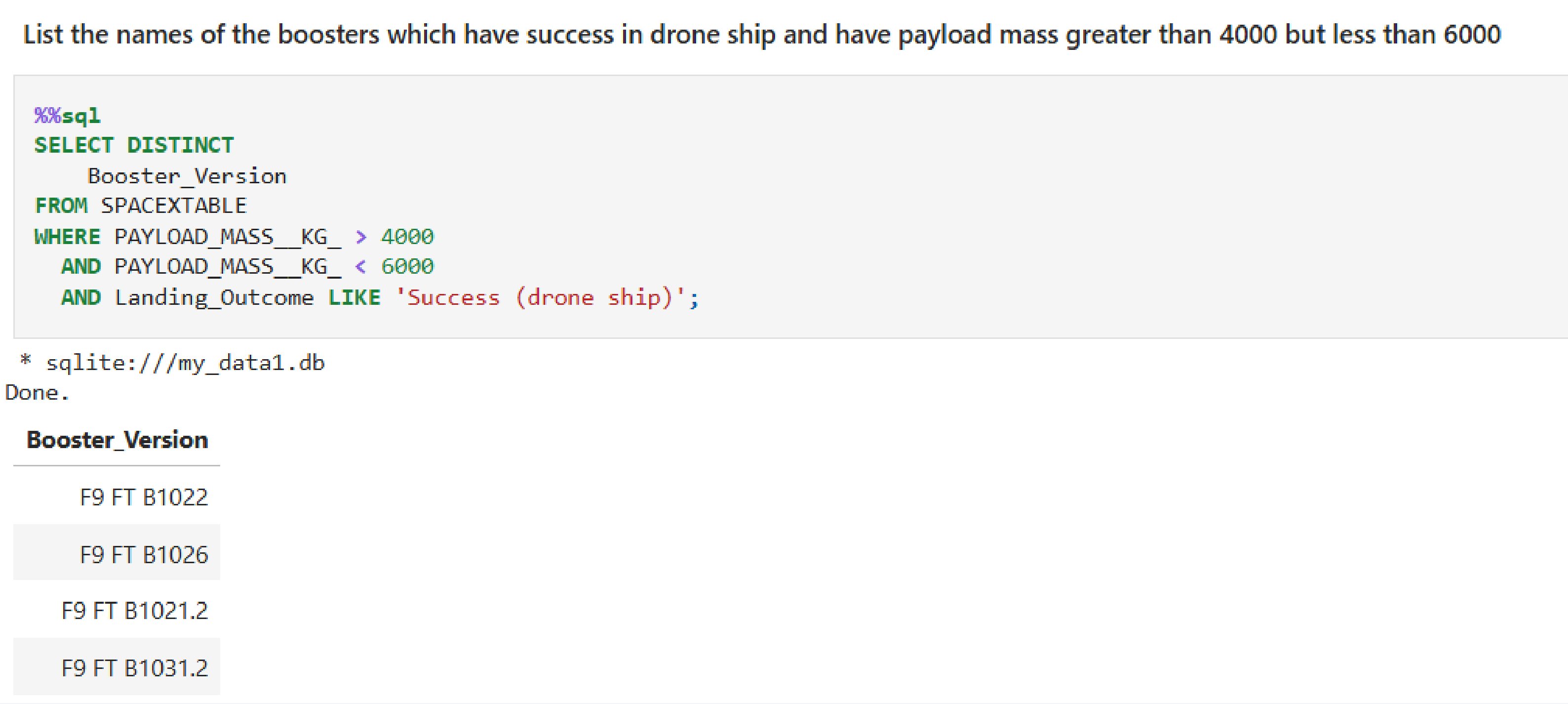
```
 * sqlite:///my_data1.db
Done.
```

First_Ground_Pad_Success
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

This SQL query allows us to retrieve all successful missions carrying payloads in the 4,000–6,000 kg range.

1. F9 FT B1022
2. F9 FT B1026
3. F9 FT B1021.2
4. F9 FT B1031.2



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

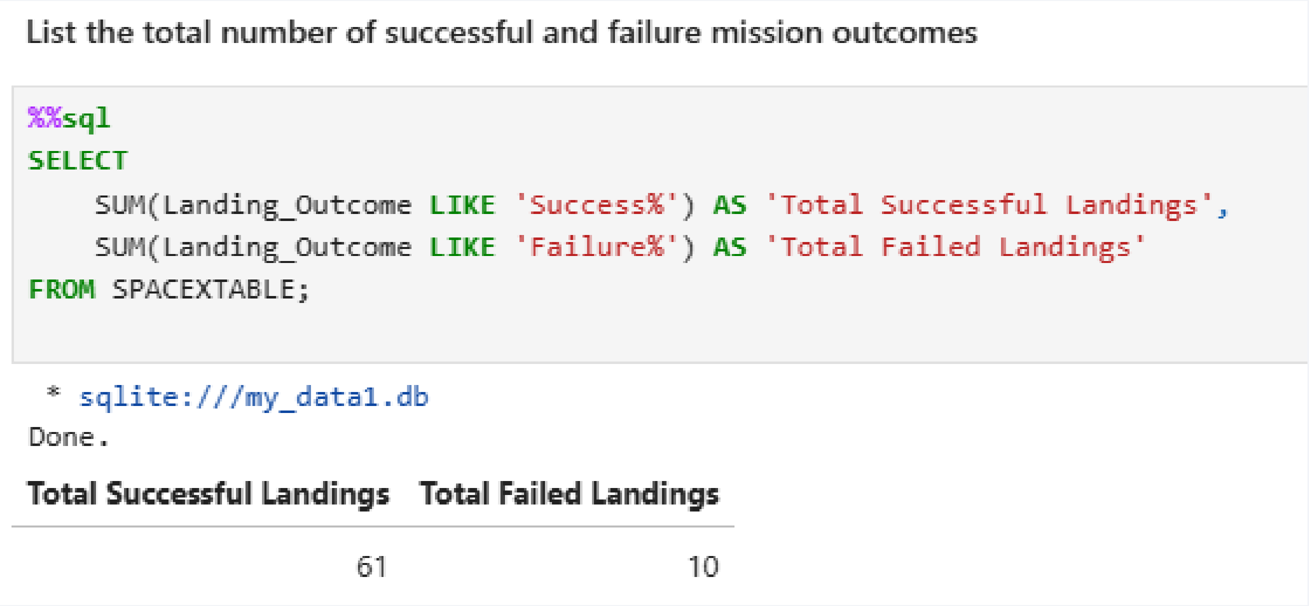
```
%%sql
SELECT DISTINCT
    Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ > 4000
  AND PAYLOAD_MASS__KG_ < 6000
  AND Landing_Outcome LIKE 'Success (drone ship)';
```

```
 * 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

The results of the query indicate that the outcome of missions were:
- 61 success
-10 failures



List the total number of successful and failure mission outcomes

```
%%sql
SELECT
    SUM(Landing_Outcome LIKE 'Success%') AS 'Total Successful Landings',
    SUM(Landing_Outcome LIKE 'Failure%') AS 'Total Failed Landings'
FROM SPACEXTABLE;
```

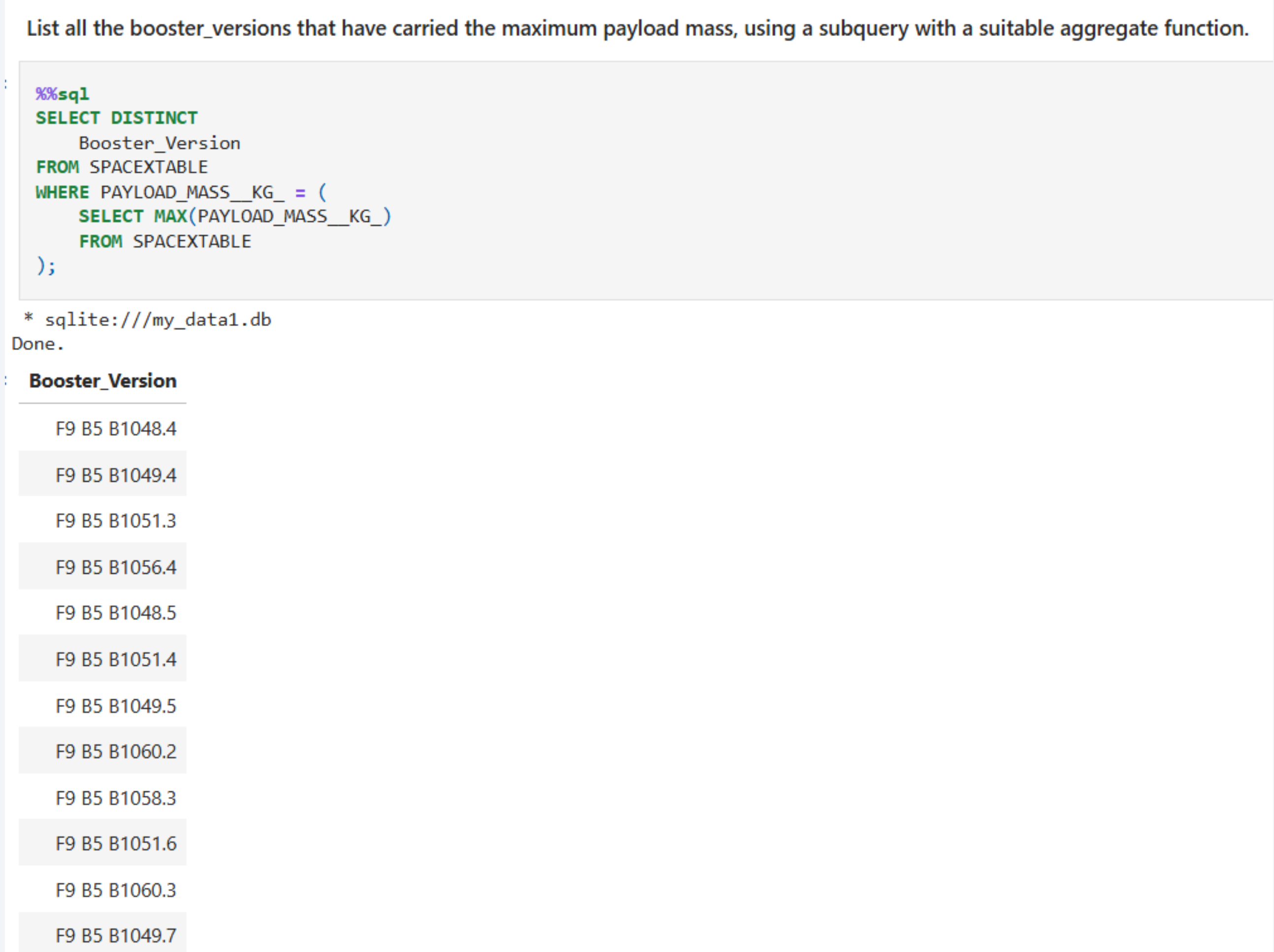
```
 * sqlite:///my_data1.db
Done.
```

Total Successful Landings	Total Failed Landings
61	10

Boosters Carried Maximum Payload

As shown in the image below, this SQL query allowed us to list the 12 launches that carried their maximum payload.

1. F9 B5 B1048.4
2. F9 B5 B1049.4
3. F9 B5 B1051.3
4. F9 B5 B1056.4
5. F9 B5 B1048.5
6. F9 B5 B1051.4
7. F9 B5 B1049.5
8. F9 B5 B1060.2
9. F9 B5 B1058.3
10. F9 B5 B1051.6
11. F9 B5 B1060.3
12. F9 B5 B1049.7



List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

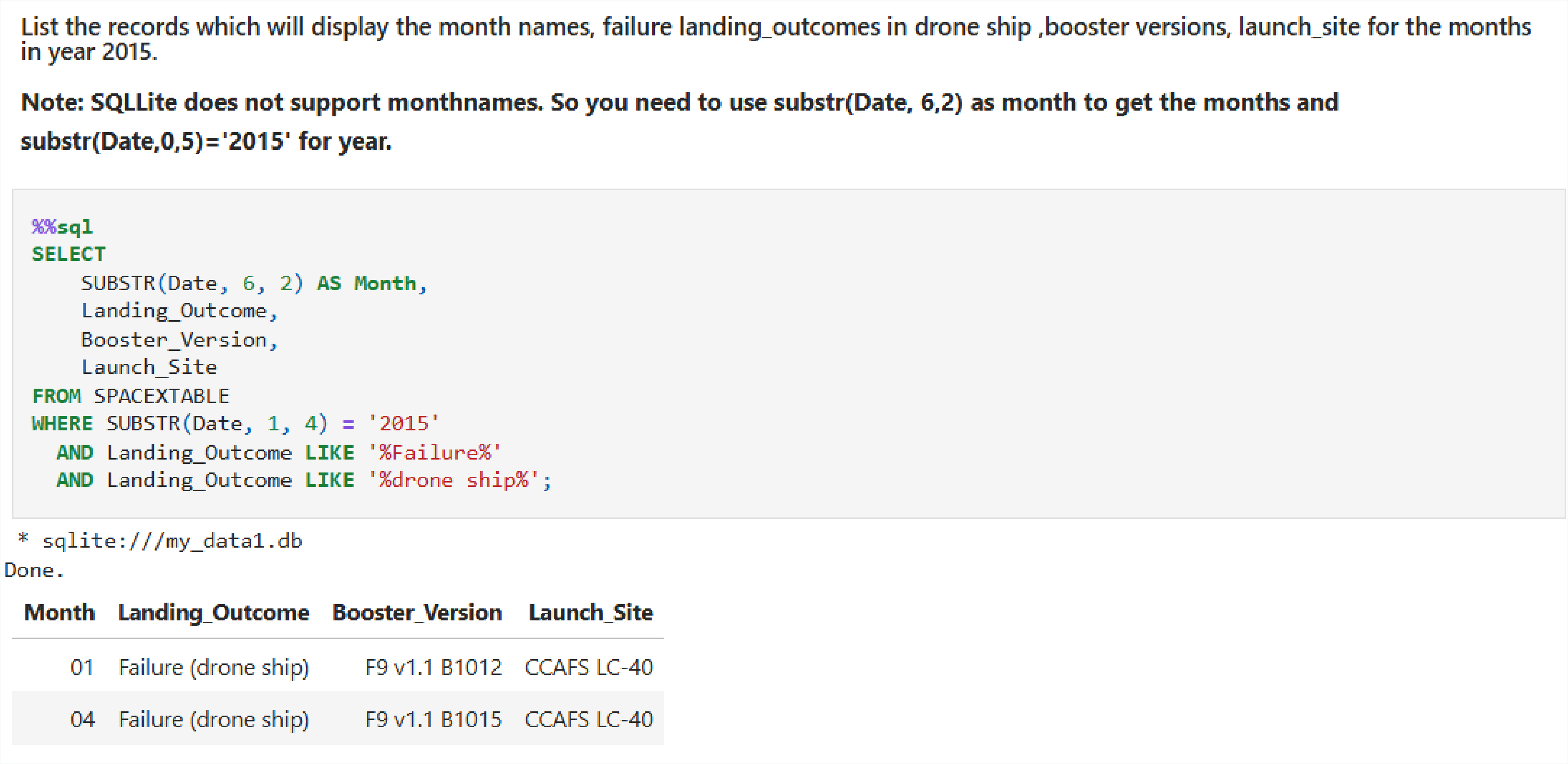
```
%%sql
SELECT DISTINCT
    Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
);
```

```
 * 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

The data shows that just two missions failed this year, both launched from the same site and separated by a three-month interval.



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.

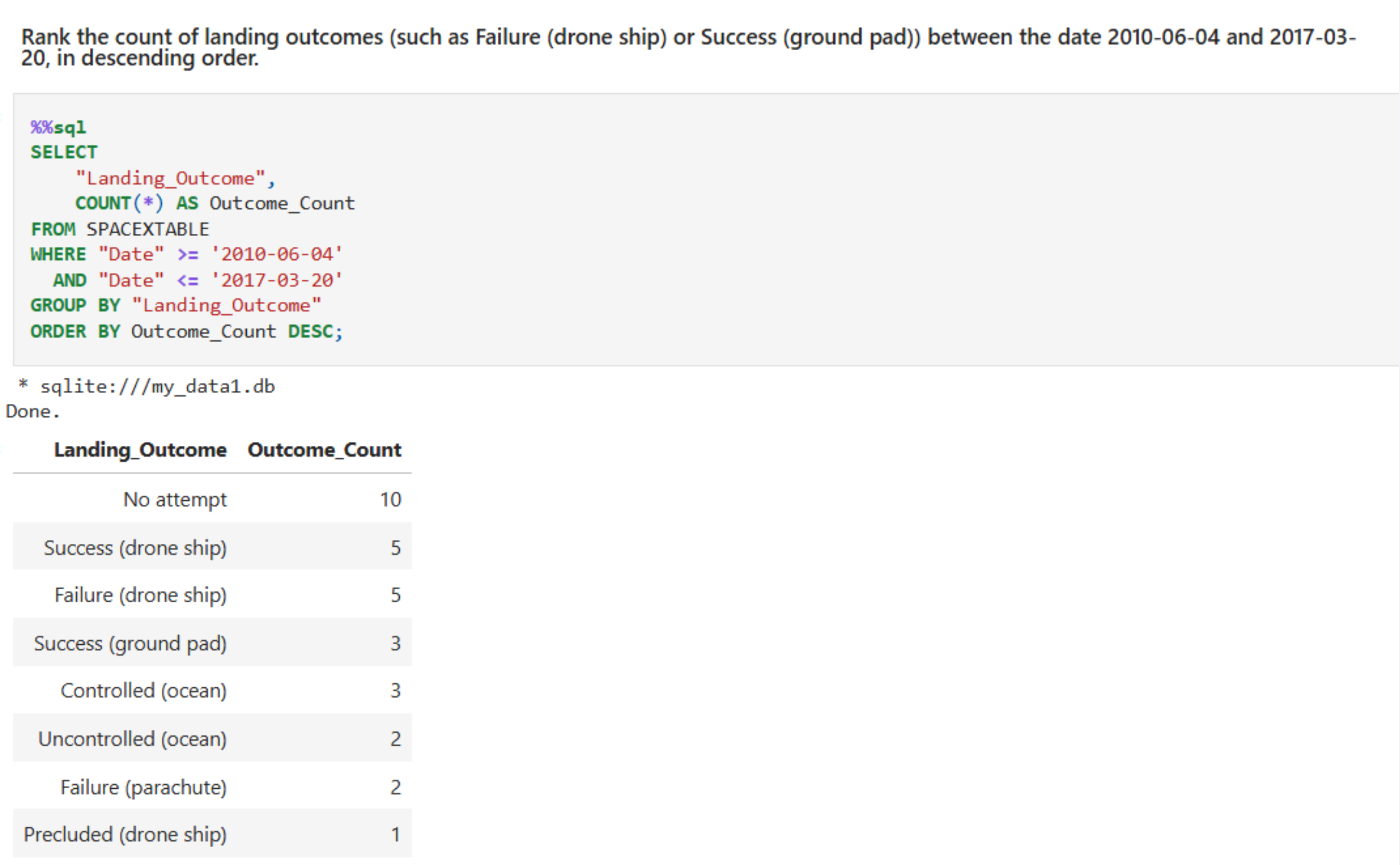
```
%%sql
SELECT
    SUBSTR(Date, 6, 2) AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE SUBSTR(Date, 1, 4) = '2015'
  AND Landing_Outcome LIKE '%Failure%'
  AND Landing_Outcome LIKE '%drone ship%';
```

```
 * 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

Finally, we summarize the number of positive outcomes by sorting them in descending order according to landing outcome type (e.g., failure on drone ship vs. success on ground platform) for missions launched 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.

```
%%sql
SELECT
    "Landing_Outcome",
    COUNT(*) AS Outcome_Count
FROM SPACEXTABLE
WHERE "Date" >= '2010-06-04'
  AND "Date" <= '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
```

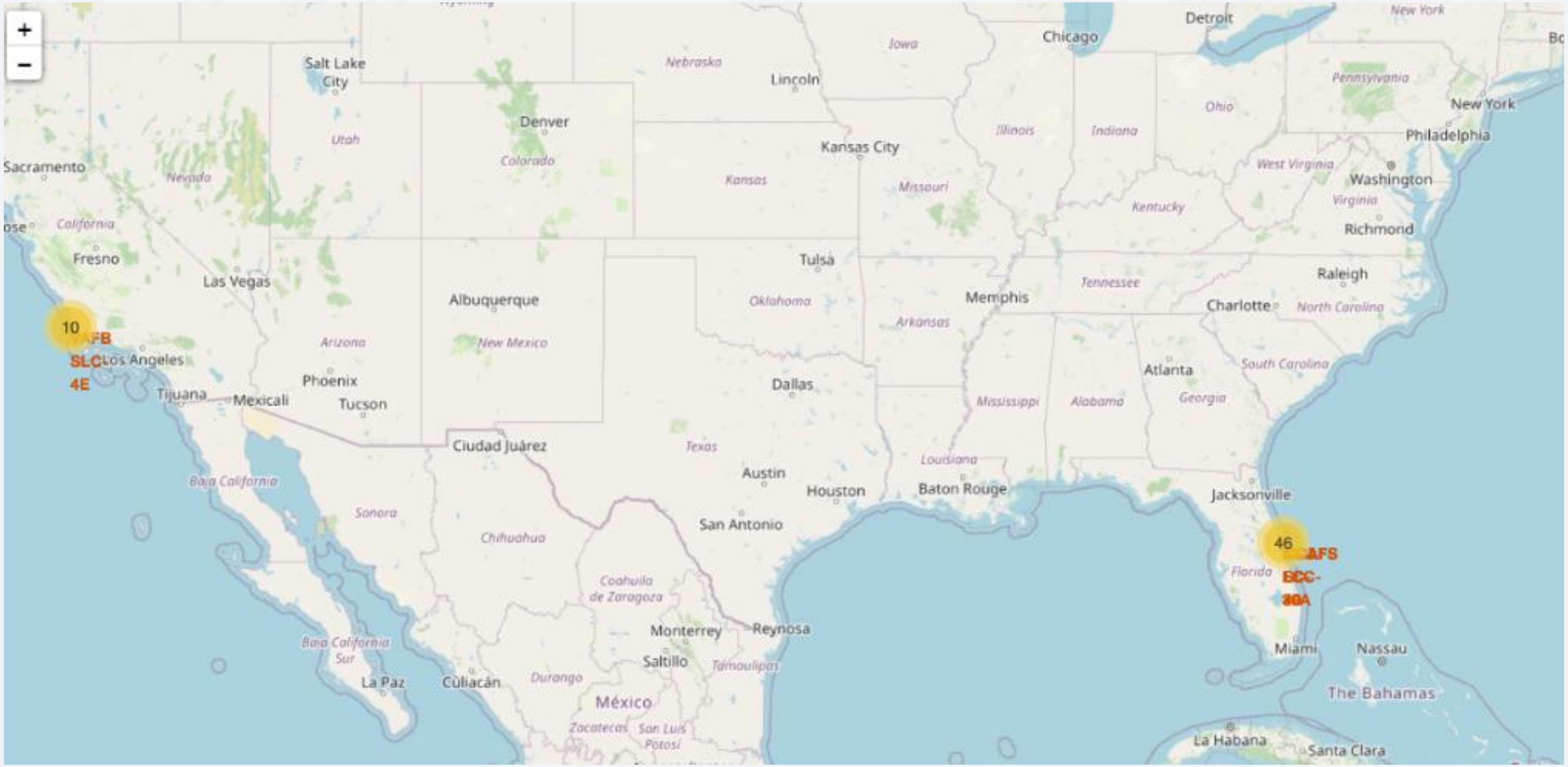
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Outcome_Count
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

Section 3

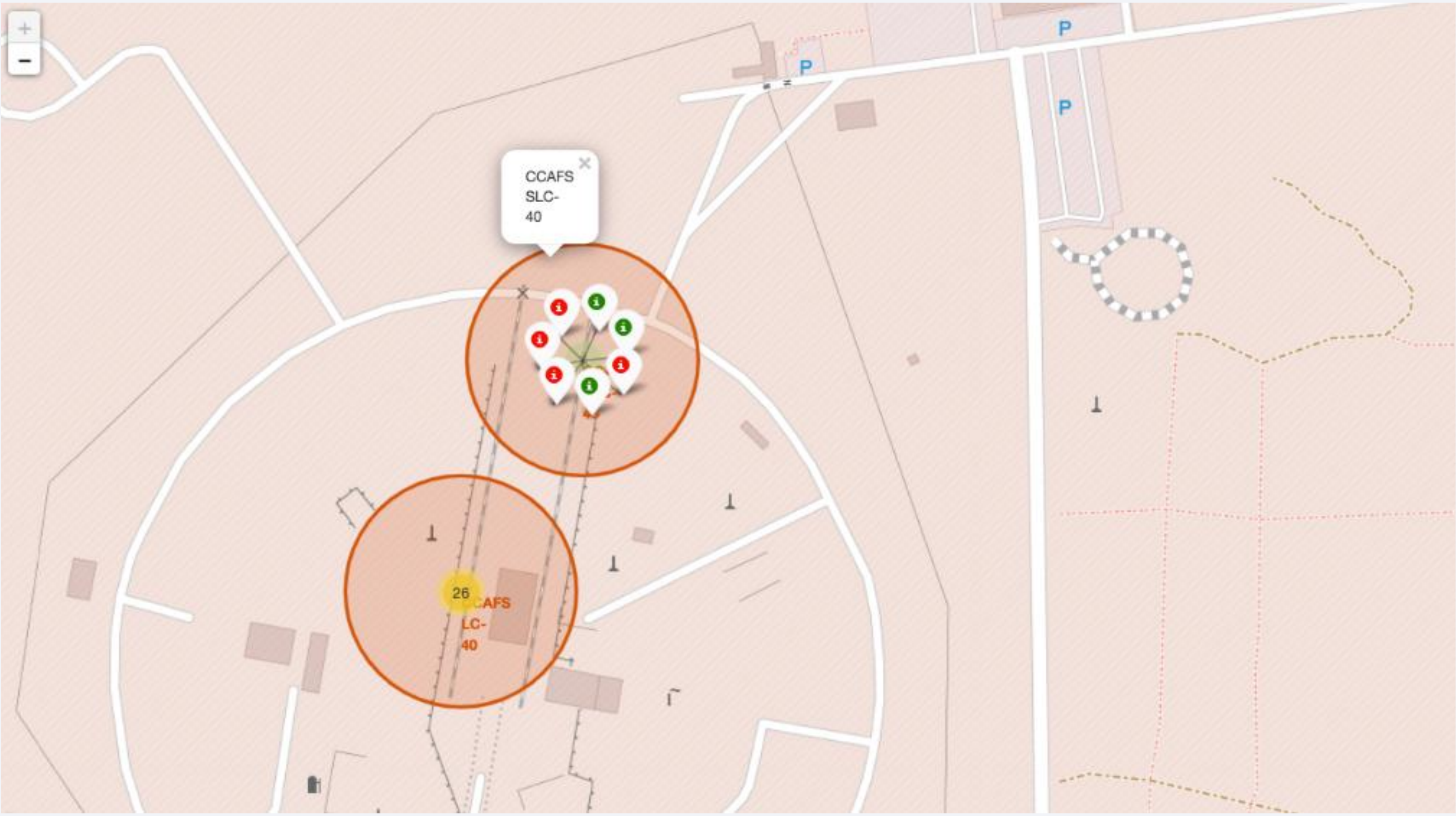
Launch Sites Proximities Analysis

1- All launch sites with Folium

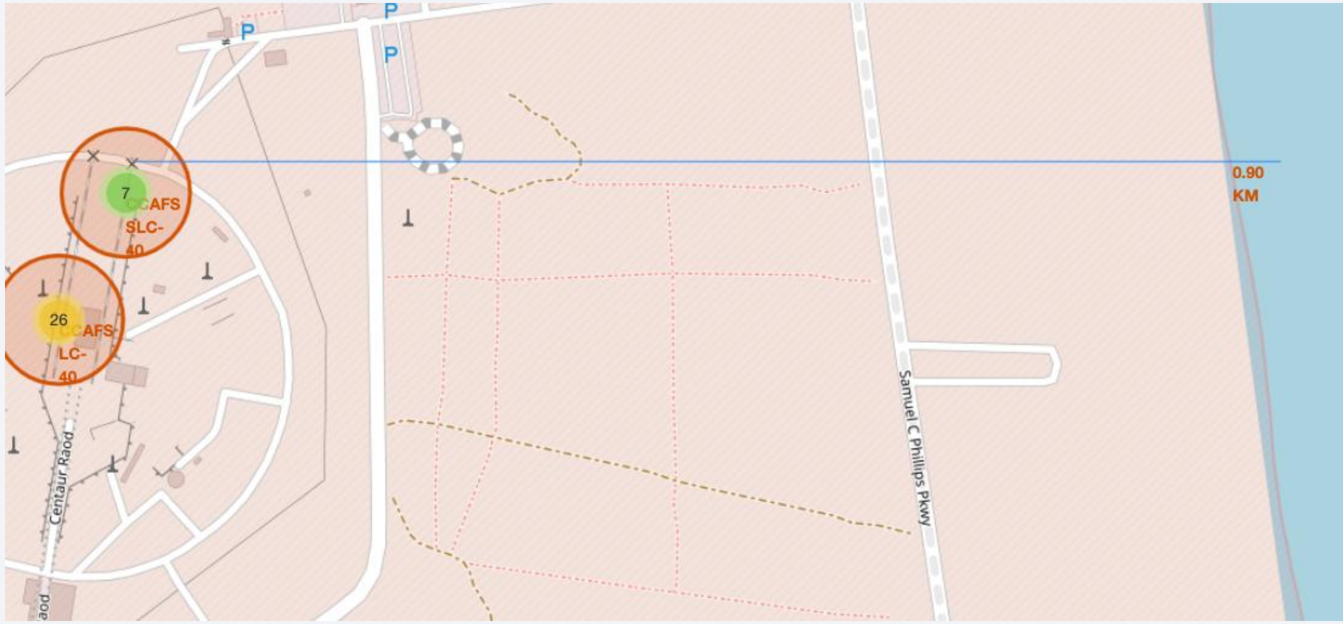


2- The launch results labeled by colors.

- **Green** Marker: **Successful** launch(class=1)
- **Red** Marker: **Failed** launch(class=0)



3- Launch location and distance to surrounding areas (cities, railways, etc.)



Launch sites are located nearby:

- **Railways** for transporting large components and hazardous materials.
- **Coastlines** so debris falls into the ocean and launches avoid populated areas.
- **Highways** to move oversized rocket parts and equipment.
- **The equator** to benefit from Earth's rotational speed and improve launch efficiency.

They are **far from cities** to reduce risks from failures, noise, shockwaves, explosive fuels, and airspace disruption.

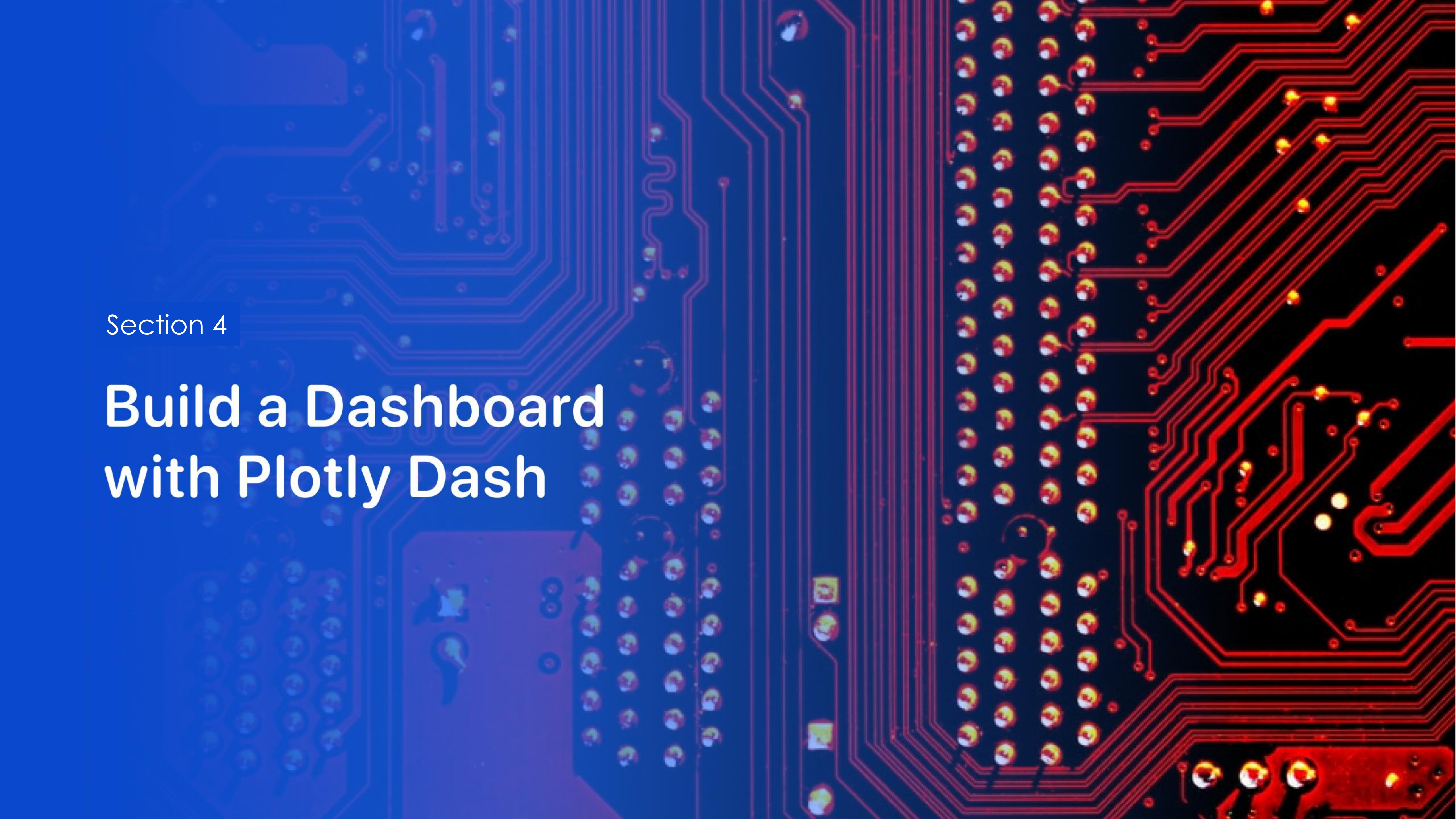
Analysis result

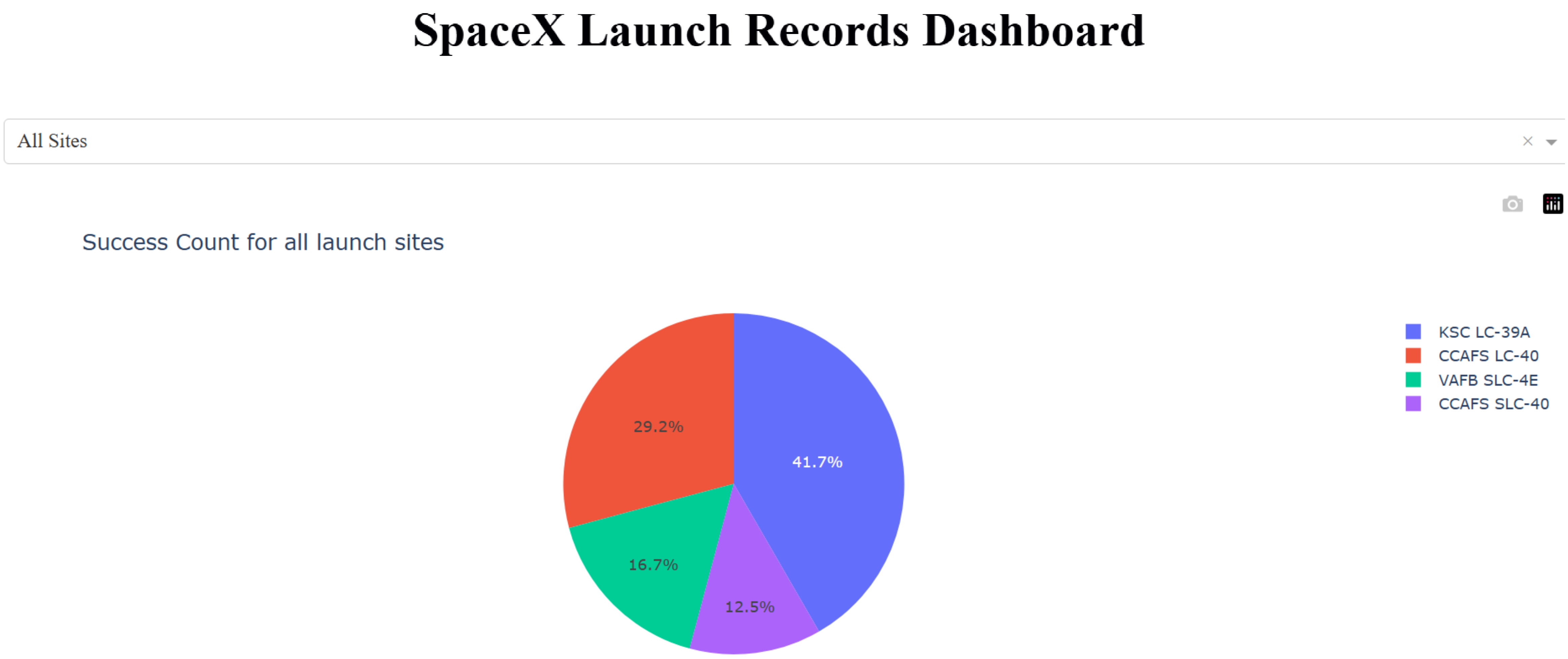
- The first image shows the geographic distribution of the launch sites on a global map.
- The second image highlights, using colored labels, the number of successful launches (in green) and failed launches (in red), along with their corresponding locations.
- We can also observe that all launch sites are positioned near coastlines and railway infrastructure, while remaining distant from major cities and highways.

References:
Link to the notebook

Section 4

Build a Dashboard with Plotly Dash



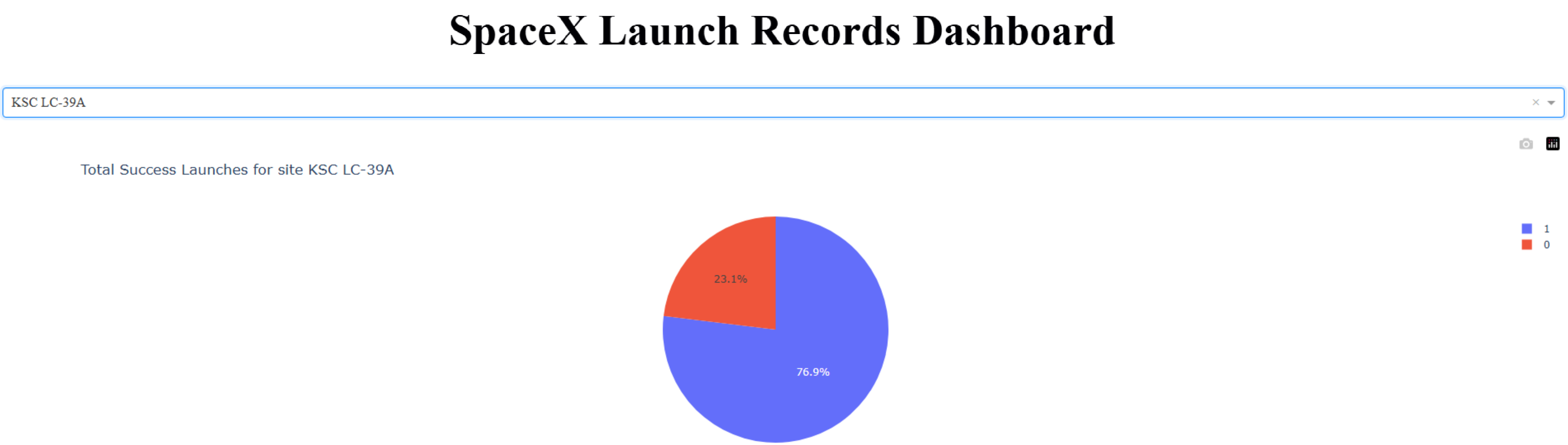


SpaceX Launch Records Dashboard

All Sites

Success Count for all launch sites

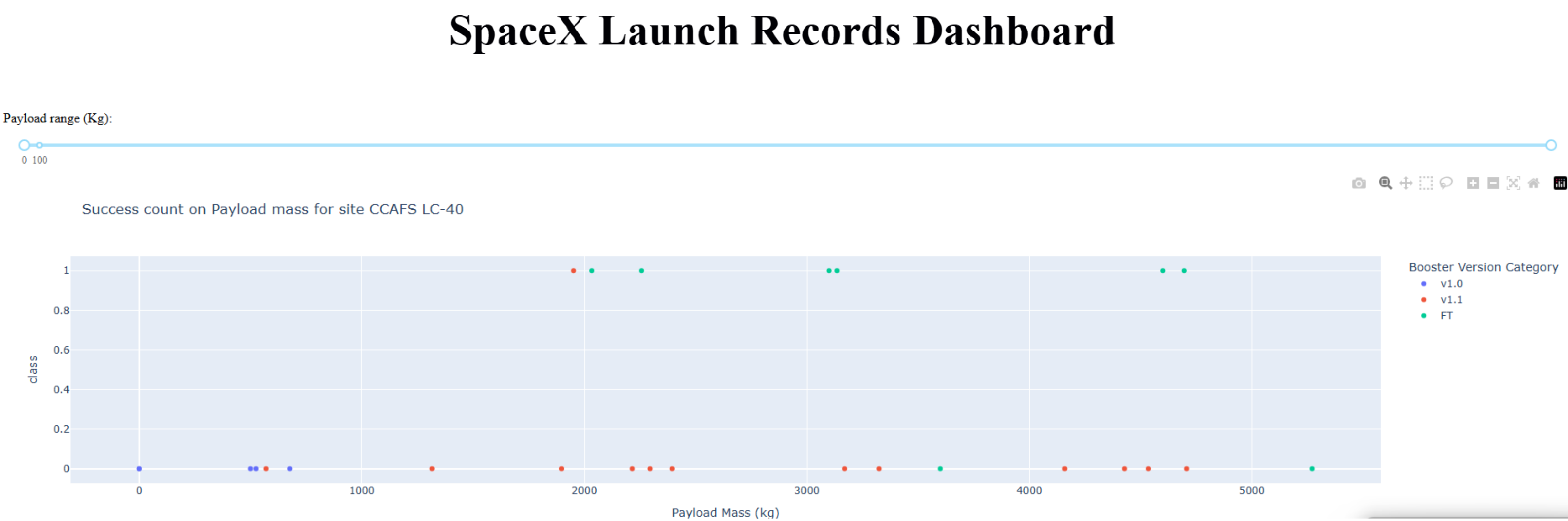
We observe that KSC LC-39A leads with a 41.7% success rate, while CCAFS SLC-40 has the lowest performance at just 12.5%.



SpaceX Launch Records Dashboard

KSC LC-39A

The pie chart for launch site KSC LC-39A shows a launch success rate of nearly 77%, with 10 successful and only 3 failed landings.



SpaceX Launch Records Dashboard

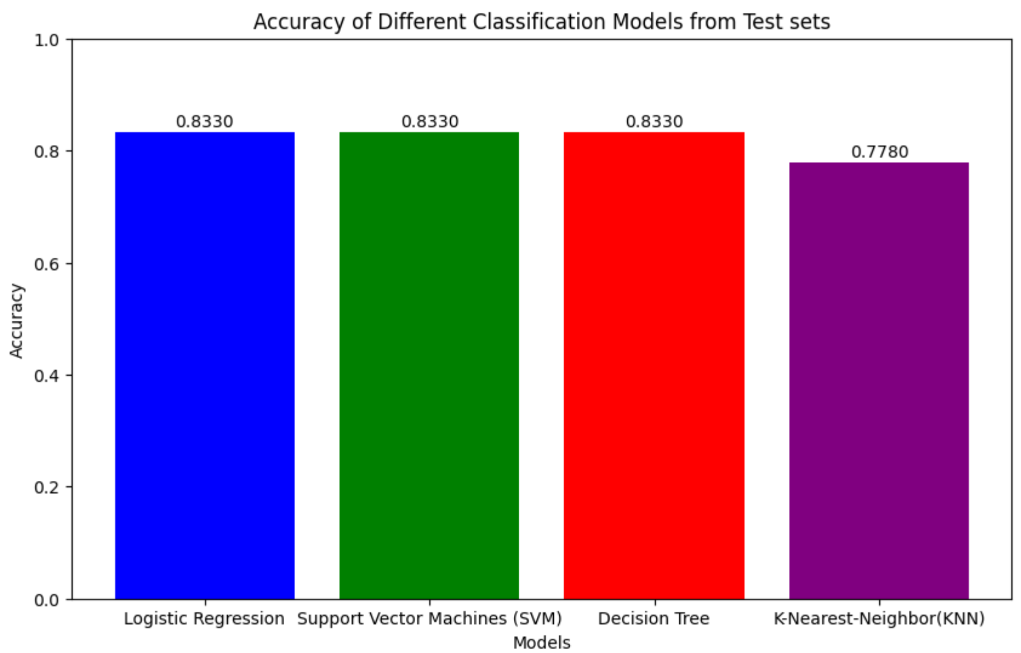
Payload range (Kg):

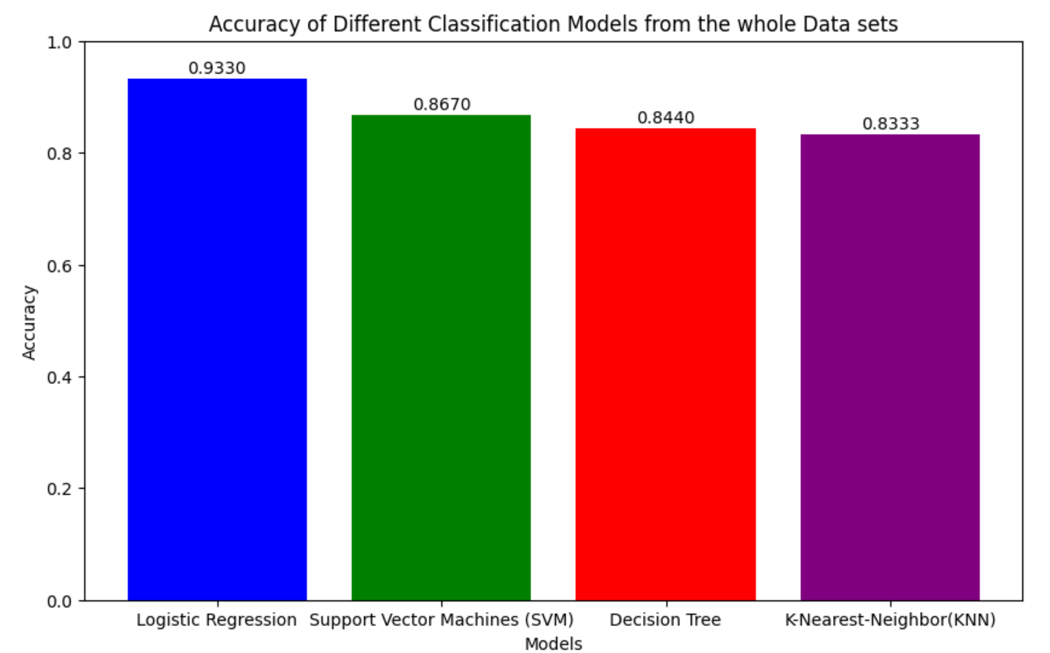
The data does not indicate a strong link between payload weight and success rate.
Successes are more frequent in the "FT" and "B5" versions, whereas "v1.0" and "v1.1" show higher failure counts.
Even very heavy payloads—up to 10,000 kg—have been launched successfully.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

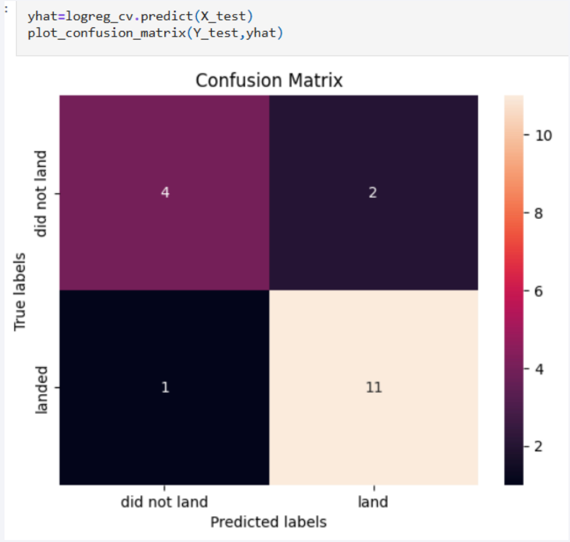




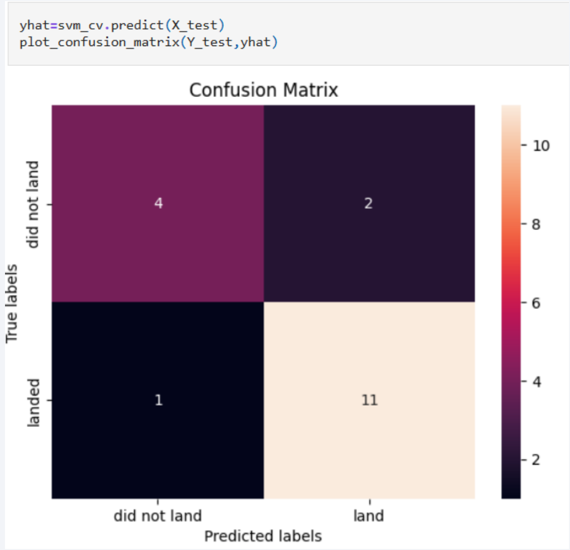
From the test set, we can see that Logistic Regression, Support Vector Machines (SVM), and Decision Tree all achieve similar performance at around 83% accuracy, while the K-Nearest Neighbor (KNN) model trails slightly with 78%.

However, when evaluating the full dataset, the differences become more pronounced. Logistic Regression performs the best with an accuracy of 93%, followed by SVM at 87%, the Decision Tree at 84%, and KNN at 83%.

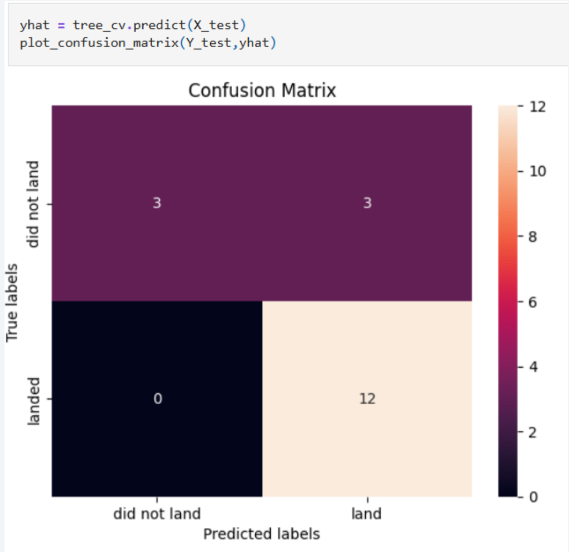
Confusion Matrix



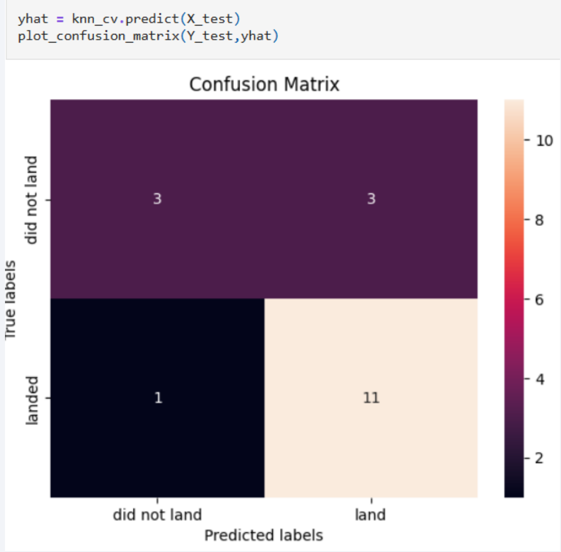
```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Logistic Regression
-Correctly predicted 11 out of 12 landings and 4 out of 5 non-landings.
-Accuracy: 83%
→Best overall performance, with the most balanced results across both classes.

Support Vector Machines (SVMs)
-Predicted 11 out of 12 landings correctly, but only 2 out of 6 non-landings.
-Accuracy: 83%
→Same accuracy performance with Logistic Regression

Decision Tree
-Perfectly predicted all 12 landings, but misclassified 3 non-landings.
-Accuracy: 83%
→High recall for landings, but lower precision due to under-predicting “did not land".

K-Nearest-Neighbor
-Same performance as Decision Tree: 11 correct landings, 3 correct non-landings.
-Accuracy: 77.8%
→Matches Logistic Regression and SVMs in accuracy but shows more confusion in non-landed cases.

Conclusions

- Regarding launch sites, KSC LC-39A shows the highest success rate, particularly for missions carrying payloads above 2,500 kg. Several factors may contribute to this performance, such as its proximity to the equator or major transportation infrastructure, though these influences would require further investigation.

- In terms of orbit types, ES-L1, GEO, HEO, and SSO demonstrate significantly higher mission success rates compared with other orbits.

- The scatter plots indicate positive relationships between the number of flights, launch site, orbit type, and the likelihood of a successful landing. Payload mass, however, does not show a clear correlation with mission success.

- The steady increase in success rates from 2013 to 2020 reflects continuous learning, engineering improvements, and ongoing innovation, with the first successful landing recorded on 2015-12-22. Across the dataset, there were 61 successful missions and only 10 failures.

- Using interactive visual analytics with Folium, we can observe geographical patterns: launch sites located near railways, coastlines, highways, or closer to the equator tend to operate more efficiently. At the same time, sites are intentionally placed far from major cities to reduce risks related to failures, noise, shockwaves, explosive fuels, and airspace disruption.

- From the machine learning perspective, models such as Logistic Regression, Support Vector Machines (SVM), and Decision Trees achieve around 83% accuracy, confirming that reusable rocket landings can be predicted reliably using supervised learning. The K-Nearest Neighbor (KNN) method performs slightly worse in comparison.

- Overall, this project highlights the strong potential of AI in real-world aerospace applications. It demonstrates how data science can analyze current conditions, identify the key factors influencing SpaceX's launch success, and support more informed decision-making in the reusable rocket industry.

Thank you!