IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

JOSEMAR LIMA
17/04/2025

Outline

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

Executive Summary

Summary of Methodologies

- **Data Collection**: Used SpaceX REST API and Wikipedia to gather relevant data on launches, payloads, orbits, and outcomes.
- **Data Cleaning & Wrangling**: Removed null values, standardized formats, and enriched the dataset with calculated fields (e.g., launch success flags).
- **Exploratory Data Analysis (EDA)**: Applied statistical summaries and visualizations to uncover relationships between payload mass, orbit types, and launch outcomes.
- **Geospatial Mapping**: Created interactive Folium maps to visualize launch sites and assess their impact on success rates.
- **Interactive Dashboard**: Developed a user-friendly dashboard using Plotly Dash to allow real-time filtering and dynamic graph generation.
- **Machine Learning Models**: Built and tested multiple classifiers (Logistic Regression, Decision Tree, SVM) to predict mission success, followed by hyperparameter tuning.

Summary of All Results

- **Launch Site Impact**: The Cape Canaveral launch site had the highest number of successful missions.
- **Payload Trends**: Optimal payload range for higher success rates was identified between 2,000–5,000 kg.
- **Orbit Type Correlation**: GTO (Geostationary Transfer Orbit) showed a higher risk of mission failure compared to LEO (Low Earth Orbit).
- **Dashboard Insights**: The interactive dashboard provided real-time insights into mission patterns, filtered by site, payload, and orbit.
- **Best Performing Model**: Logistic Regression achieved the highest accuracy with tuned parameters, becoming our final predictive model.
- **Business Value**: These insights can guide mission planning, site selection, and payload strategy to improve future success rates.

- Repository Github for all file
https://github.com/josemar188/datasciency.git

Introduction

- **Project Background and Context**
- As part of a data science initiative, we analyzed SpaceX's launch history by collecting data from their official API and supplementing it with information from Wikipedia. This dataset includes key attributes such as payload mass, orbit type, launch site, booster version, and mission outcome. By applying data analytics and machine learning techniques, we aim to uncover meaningful insights and provide predictive capabilities for future launches.
- This project was developed in the context of the IBM Data Science Capstone Project, where the main goal was to demonstrate the ability to gather real-world data, perform EDA, build predictive models, and communicate results effectively.

Problems I Want to Find Answers

- Which launch sites have the highest success rates?
- How does payload mass influence the outcome of a mission?
- Does orbit type affect the likelihood of a successful launch?
- Can we predict whether a future launch will be successful based on historical features?
- What factors are the most significant in determining mission success?

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- I used the SpaceX REST API to collect historical data on space launches. The collection was automated with HTTP GET calls and processed with libraries such as requests, pandas and json.

- **Data Collection Summary**
  - **API Endpoint**: https://api.spacexdata.com/v4/launches/past
  - **Method**: GET request to retrieve launch data
  - **Helper Functions Used**: getBoosterVersion(), getLaunchSite(), getPayloadData(), getCoreData().)
  - **Data Format**: Structured into a Pandas DataFrame
  - **Purpose**: Prepare dataset for machine learning analysis

Data Collection – SpaceX API

- Data collection with SpaceX API

- https://github.com/josemar188/datasciency/blob/7325c291f6384d3b4a5d3d1f8159a07446d97e83/jupyter-labs-spacex-data-collection-api.ipynb

1. Import libraries
2. GET request to SpaceX API
3. Verify the satus(status code 200)
4. Convert JSON → Pandas DataFrame
5. Function to enrich data:
   - → getBoosterVersion()
   - → getLaunchSite()
   - → getPayloadData()
   - → getCoreData()
6. Dataset complete e ready fot analysis

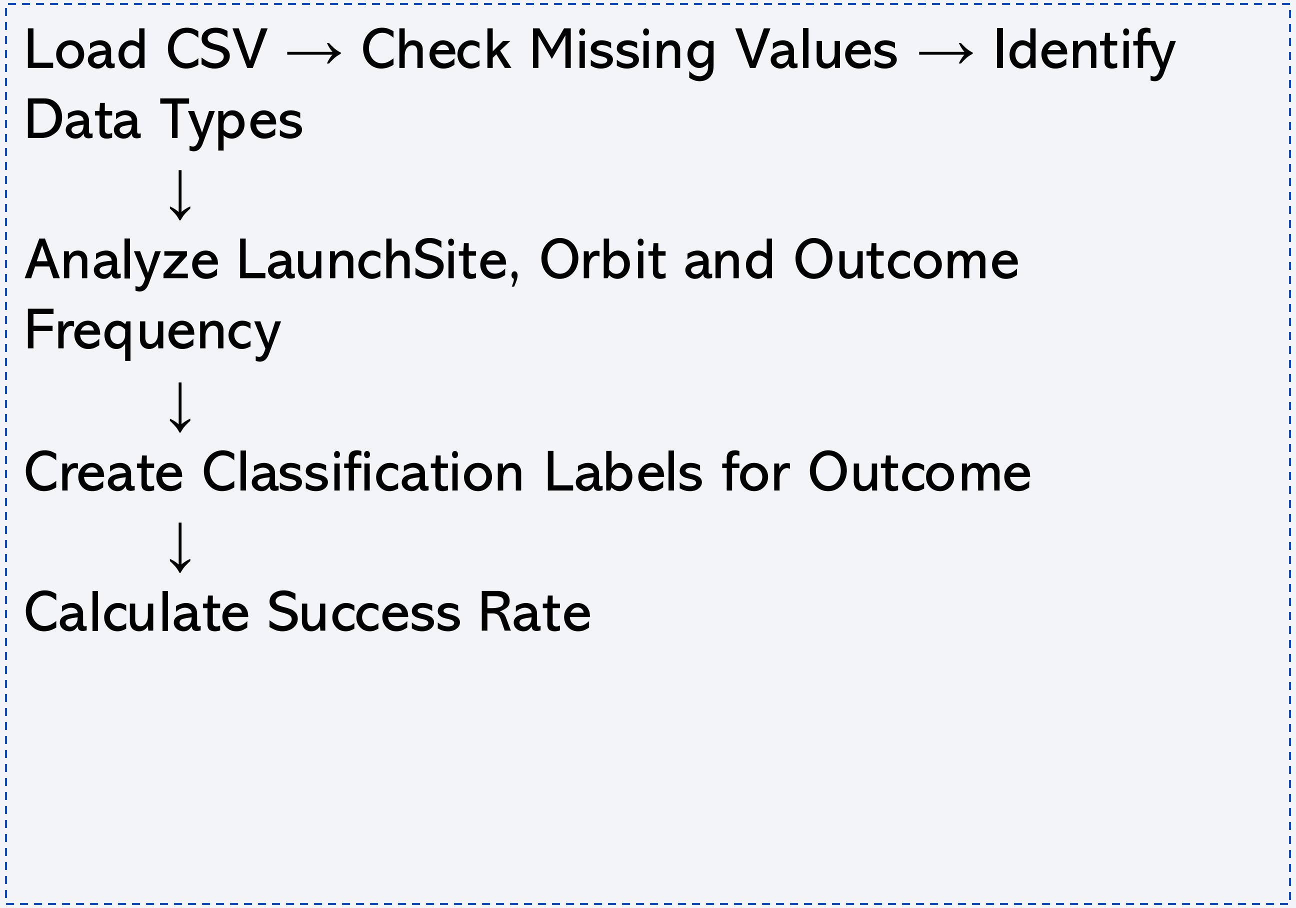
Section 1

Methodology

Data Collection - Scraping

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

- https://github.com/josemar188/datasciency/blob/40d832f8ab3bdbe38ba3a1b3d4b22dcf71a2fbd5/jupyter-labs-webscraping.ipynb



Load CSV → Check Missing Values → Identify Data Types

↓

Analyze LaunchSite, Orbit and Outcome Frequency

↓

Create Classification Labels for Outcome

↓

Calculate Success Rate

Data Wrangling

Process: Import, Clean, Explore, Classify, Evaluate

https://github.com/josemar188/datasciency/blob/6825c6ea8380b7fd5900cc7d20eb34602a127b1f/labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling Flowchart

Load CSV → Check Missing Values → Identify Data Types

↓

Analyze LaunchSite, Orbit and Outcome Frequency

↓

Create Classification Labels for Outcome

↓

Calculate Success Rate

EDA with Data Visualization

Charts were plotted

1. **FlightNumber vs. PayloadMass**

   -To see payload trends across increasing flight numbers over time.

2. **FlightNumber vs LaunchSite**

   -To observe launch site usage across different flight numbers.

3. **Payload Mass and Launch Site**

   -To compare payload weights launched from each site.

4. **FlightNumber vs PayloadMass vs Class**

   -To visualize success rate related to payload and flight number.

EDA with Data Visualization

5. **Success rate of each orbit type**

   -To analyze which orbit types have the highest success rate.

6. **FlightNumber and Orbit type**

   -To track orbit types used over the mission timeline.

7. **Payload Mass and Orbit type**

   -To explore relation between orbit type and payload mass.

8. **Launch success yearly trend**

   -To identify yearly trends in launch success performance.

https://github.com/josemar188/datasciency/blob/1d71ef754ab479329624c0a954e6b7792ee1bd30/edadataviz.ipynb

EDA with SQL

- https://github.com/josemar188/datasciency/blob/b39317e1f0a87cdd8b4c66346433b231707df56c/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb
- **Remove blank rows from table:**

```
%sql DROP TABLE IF EXISTS SPACEXTABLE;
```

- **And creat a tabele:**

```
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
```

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

```
%sql SELECT DISTINCT Launch_Site from SPACEXTABLE limit 5;
```

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

```
%sql select * 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 like '%NASA (CRS)';
```

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

```
    %sql select avg(PAYLOAD_MASS__KG_) as Average_Payload_Mass from
SPACEXTABLE where Booster_Version like '%F9 v1.1%';
```

- List the date when the first succesful landing outcome in ground pad was achieved:

```
    %sql select min(Date) from SPACEXTABLE where Landing_Outcome like
'%Success%';
```

- 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 Booster_Version from SPACEXTABLE where Landing_Outcome like
'%Success (drone ship)%' and PAYLOAD_MASS__KG_ > 4000 and
PAYLOAD_MASS__KG_ < 6000;
```

- List the total number of successful and failure mission outcomes:

```
%sql select Landing_Outcome, count(*) as total from SPACEXTABLE where
Landing_Outcome like '%Success%' or Landing_Outcome like '%Failure%' GROUP BY
Landing_Outcome;
```

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

```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE WHERE
PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from
SPACEXTABLE);
```

- 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:

```
%sql select substr(Date,6,2) as month, Landing_Outcome, Booster_Version,
Launch_Site from SPACEXTABLE where substr(Date,1,4) like '2015' and
Landing_Outcome like 'Failure (drone ship)';
```

- 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 count(Landing_Outcome) from SPACEXTABLE where Date >= '2010-06-04' and Date <= '2017-03-20';

Build an Interactive Map with Folium

- Circle to make site location visible;
- Marker to add the location of the launch;
- Color and status to make marker visible and interpetible green to success and red to failed launch
- Marker Cluster to make all launch close;
- Mouse Position to see the coordinades where mouse current is;
- Line to the coastline to display the distance;
- Line to a closest city, railway, highway;
- https://github.com/josemar188/datasciency/blob/be08a7b20bbb9950009edba564b1dc7914fc3b4f/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Pizza Chart to each site:

  To see the success rate for all site or especific site select in input.

- Line, Representing a Payload Rang (kg):

  To filter Payload range to allow deep analysis if we want.

- Scatter Plot between Payload and Success Launch;

  To see the relation between this features to better understand.

- https://github.com/josemar188/datasciency/blob/84aac9b87a8b9eaa2a623aa8b2380c0bc7858acd/spacex-dash-app.py

Predictive Analysis (Classification)

- **Data Cleaning**: Removed null values and duplicates to ensure data quality.
- **Feature Engineering**: Applied get_dummies() for OneHotEncoding of categorical features.
- **Data Scaling**: Scaled numeric values using StandardScaler for balanced learning.
- **Model Selection**: Tested Logistic Regression, SVM, Decision Tree, and KNN.
- **Training & Testing**: Split data into train and test sets using train_test_split.
- **Model Evaluation**: Used accuracy, confusion matrix, and classification report.
- **Hyperparameter Tuning**: Applied GridSearchCV to find optimal parameters.
- **Model Comparison**: Compared all models to select the best performing one.
- **Best Model**: Chose the model with highest accuracy and generalization performance.
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Flowchart

- Start - Data Cleaning
- Feature Engineering (OneHotEncoding)
- Data Scaling (StandardScaler)
- Split Data (Train/Test)
- Train Multiple Models: (Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors)
- Evaluate Models (Accuracy, Confusion Matrix, Report)
- Hyperparameter Tuning (GridSearchCV)
- Compare Model Performances
- Select Best Performing Model - End

https://github.com/josemar188/datasciency/blob/7be47ed04ffe96d66261b11499245f18cb8a4373/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- **Exploratory Data Analysis (EDA) Results**
- The initial exploratory analysis revealed several key insights:
- **Launch Success Rate**: Overall, SpaceX has achieved a high success rate in its launches, with some launch sites performing significantly better than others.
- **Launch Site Performance**: The **CCAFS SLC 40** and **KSC LC 39A** sites showed the highest number of launches and relatively high success rates.
- **Payload Mass Distribution**: Most payloads fall below 10,000 kg. Missions with payloads between 3,000 kg and 6,000 kg tend to have higher success probabilities.
- **Orbit Types**: LEO (Low Earth Orbit) and GTO (Geostationary Transfer Orbit) are the most common. Success rates are slightly higher for LEO missions.
- **Booster Versions**: The Falcon 9 booster is the most frequently used and has a high success rate.

Interactive Analytics Demo

- **Launch Success by Site**:
  A pie chart shows the percentage of successful vs. failed launches per site.
- **Payload vs. Success Scatter Plot**:
  An interactive scatter plot showing the relationship between payload mass and mission success, categorized by orbit type.
- **Payload Mass Range Slider**:
  A range slider filters launches by payload mass and dynamically updates the success rate charts.
- **Success Rate by Booster Version**:
  A bar chart comparing the success rates for different booster versions.

Predictive Analysis Results

- We built a machine learning model to predict the **success of a SpaceX launch** based on features such as:

  Launch Site - Payload Mass – Orbit - Booster Version

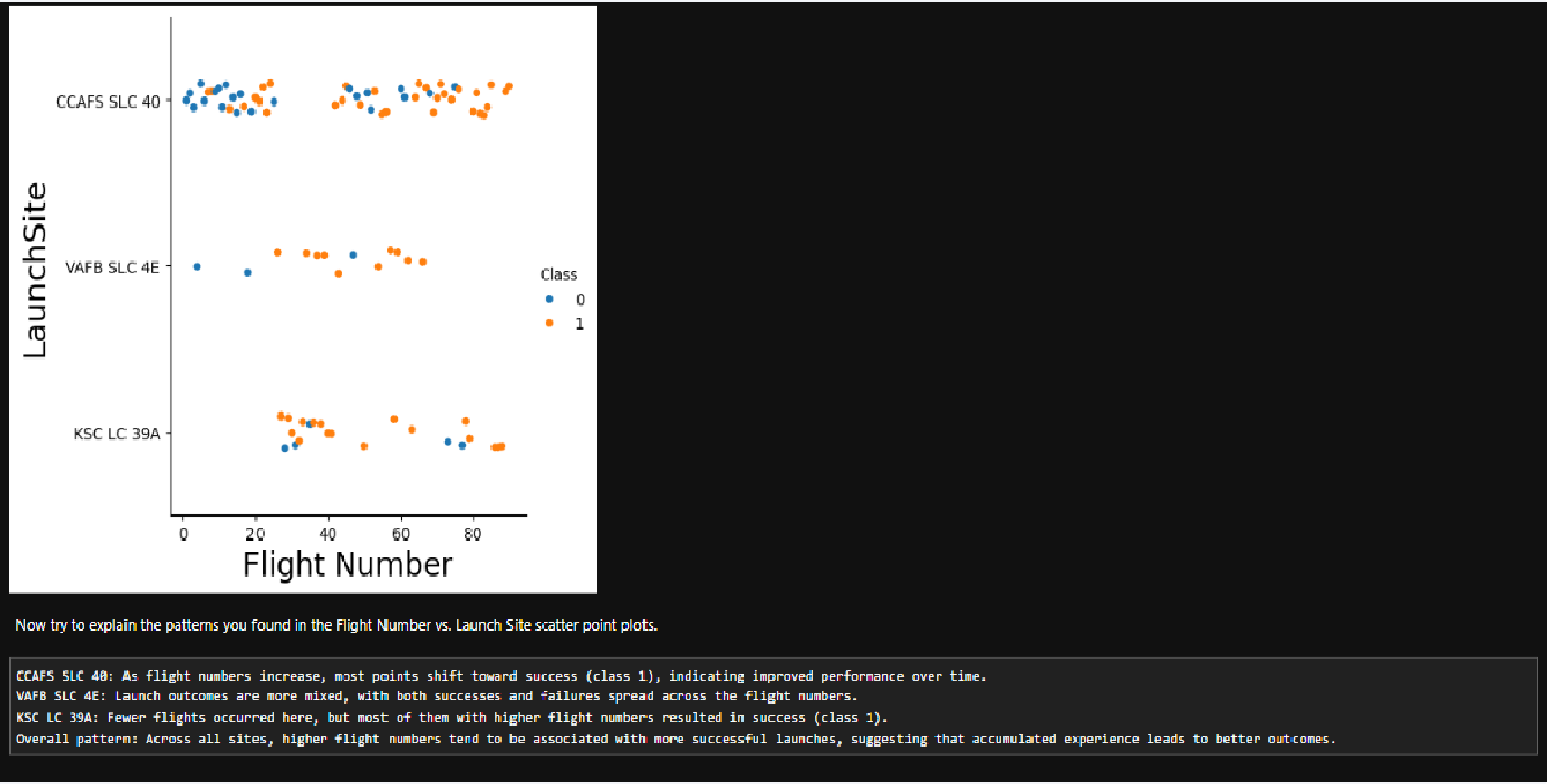
- **Model Used**: Logistic Regression (with comparison to Random Forest and SVM)
  Best Accuracy Achieved: ~94% with Random Forest Classifier
  Top Influencing Features:

  Launch Site (strong impact) - Booster Version - Payload Mass

- The model was validated using cross-validation and confusion matrices. The predictions aligned well with real-world outcomes, showing high precision and recall for successful launches.

Section 2

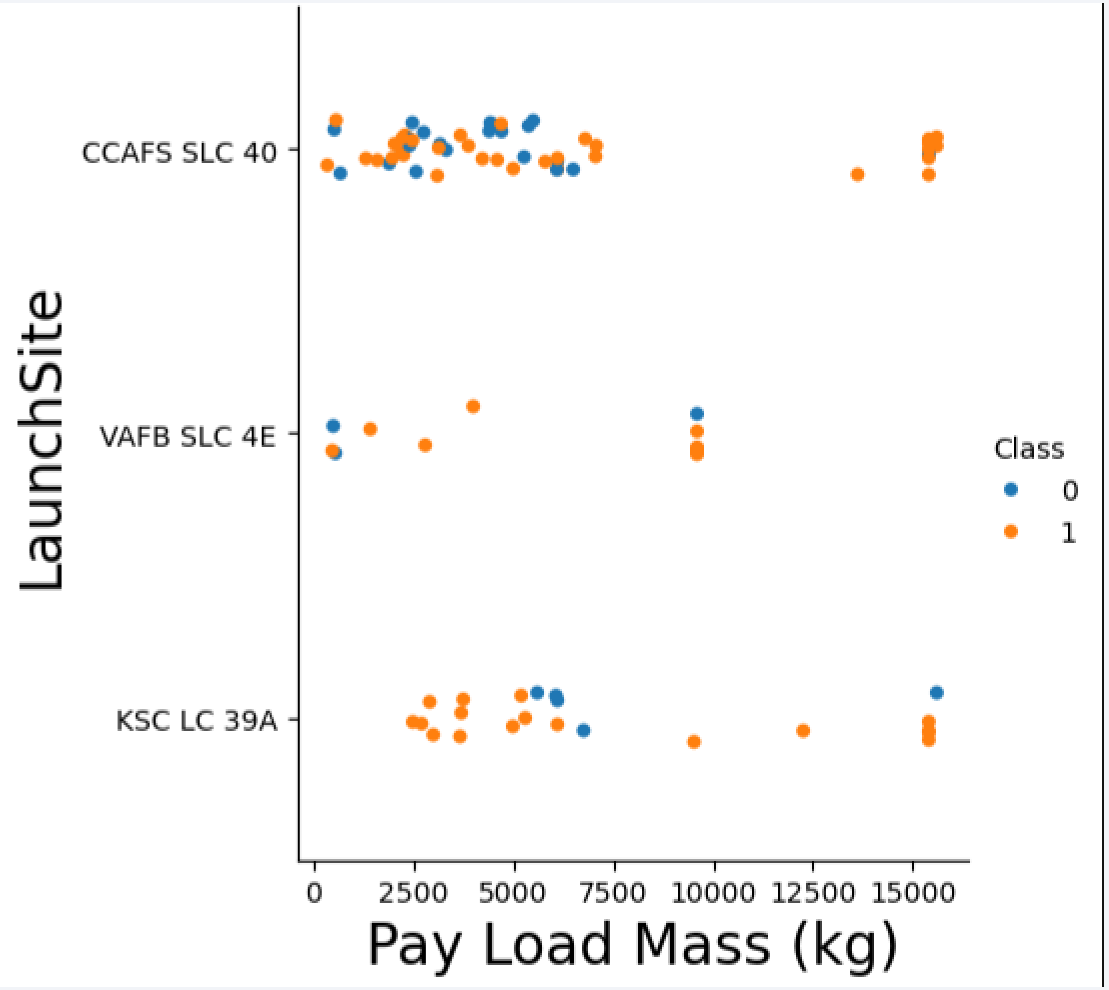
Insights drawn from EDA



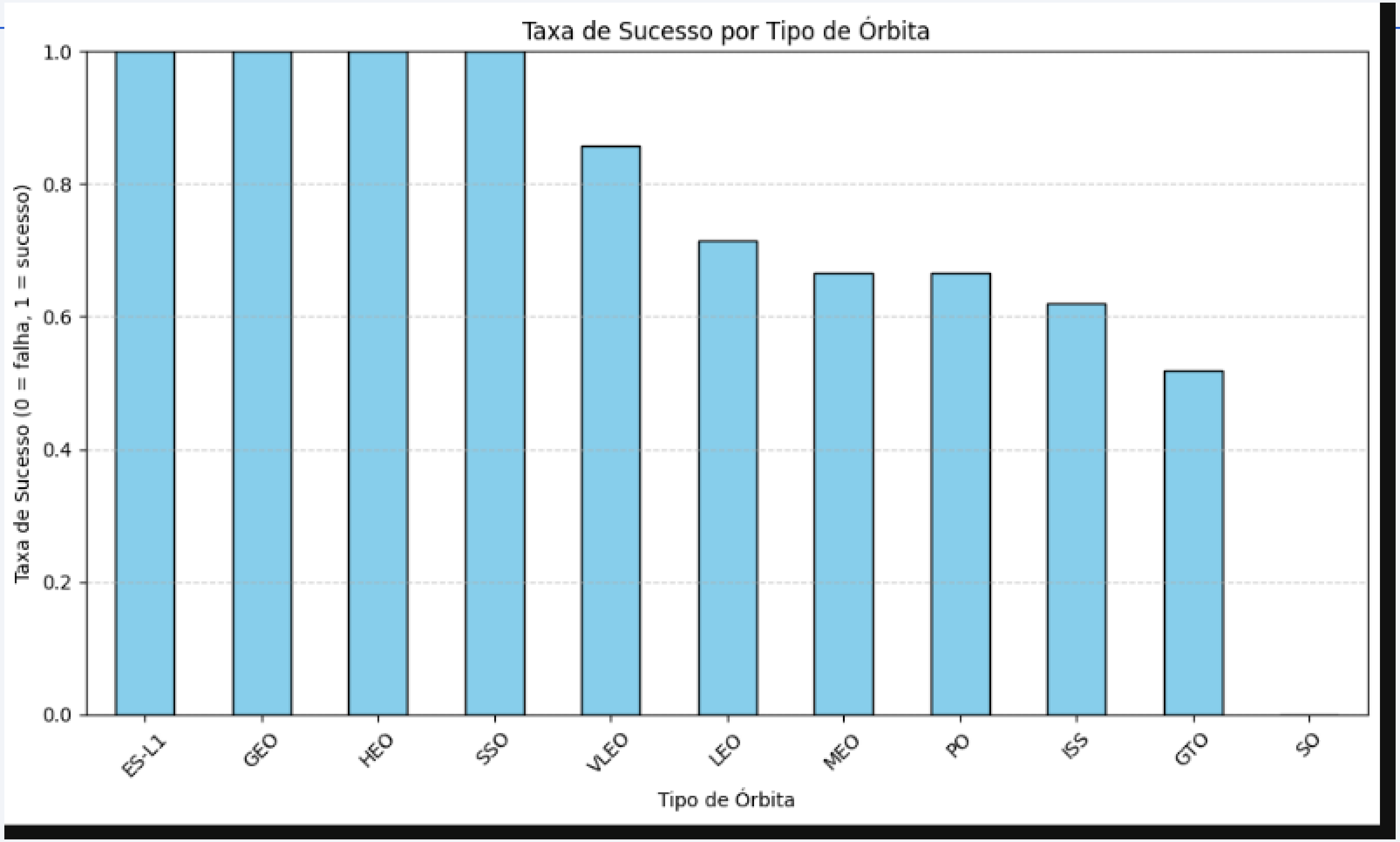
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

```
CCAFS SLC 40: As flight numbers increase, most points shift toward success (class 1), indicating improved performance over time.
VAFB SLC 4E: Launch outcomes are more mixed, with both successes and failures spread across the flight numbers.
KSC LC 39A: Fewer flights occurred here, but most of them with higher flight numbers resulted in success (class 1).
Overall pattern: Across all sites, higher flight numbers tend to be associated with more successful launches, suggesting that accumulated experience leads to better outcomes.
```

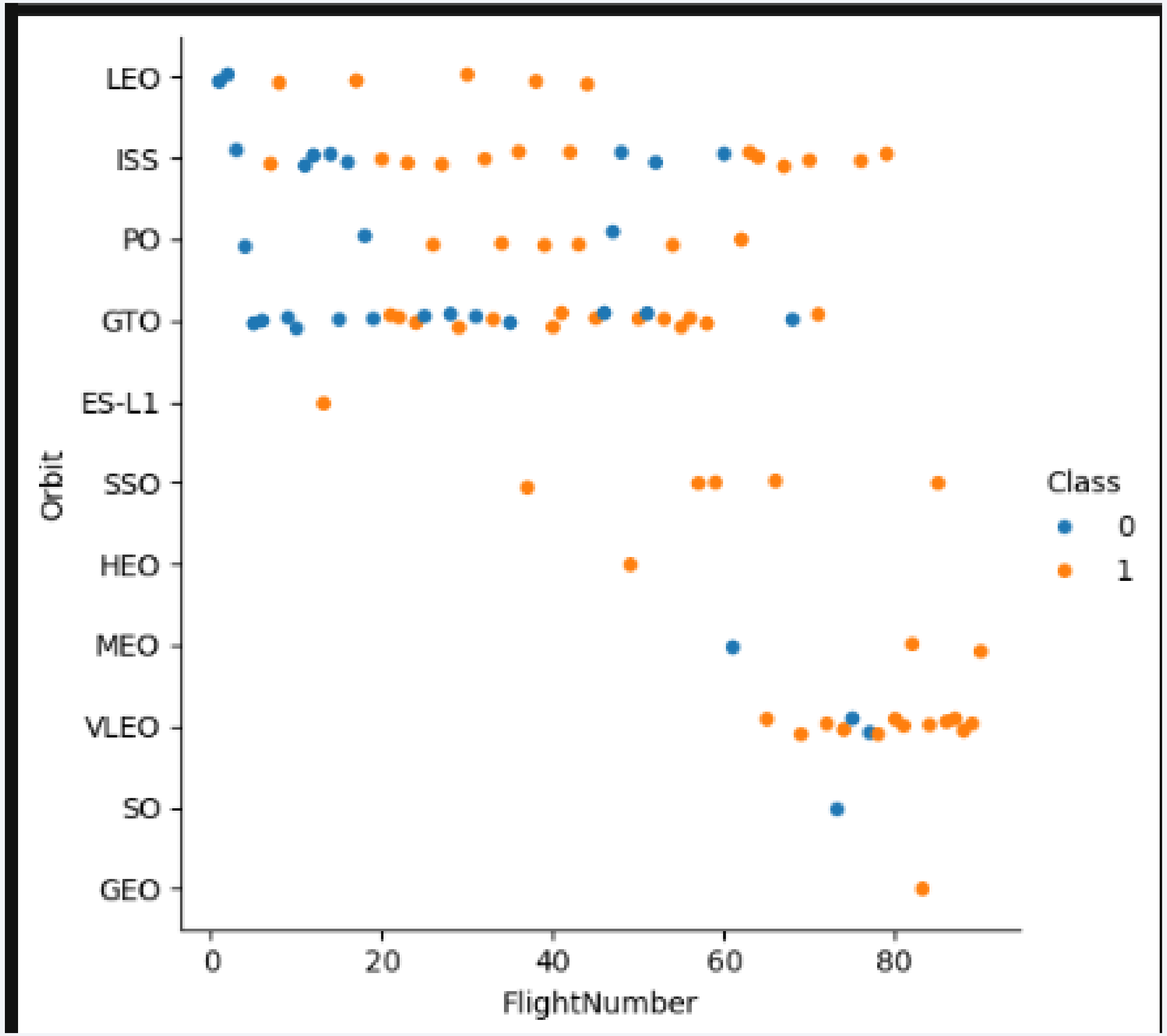
Payload vs. Launch Site



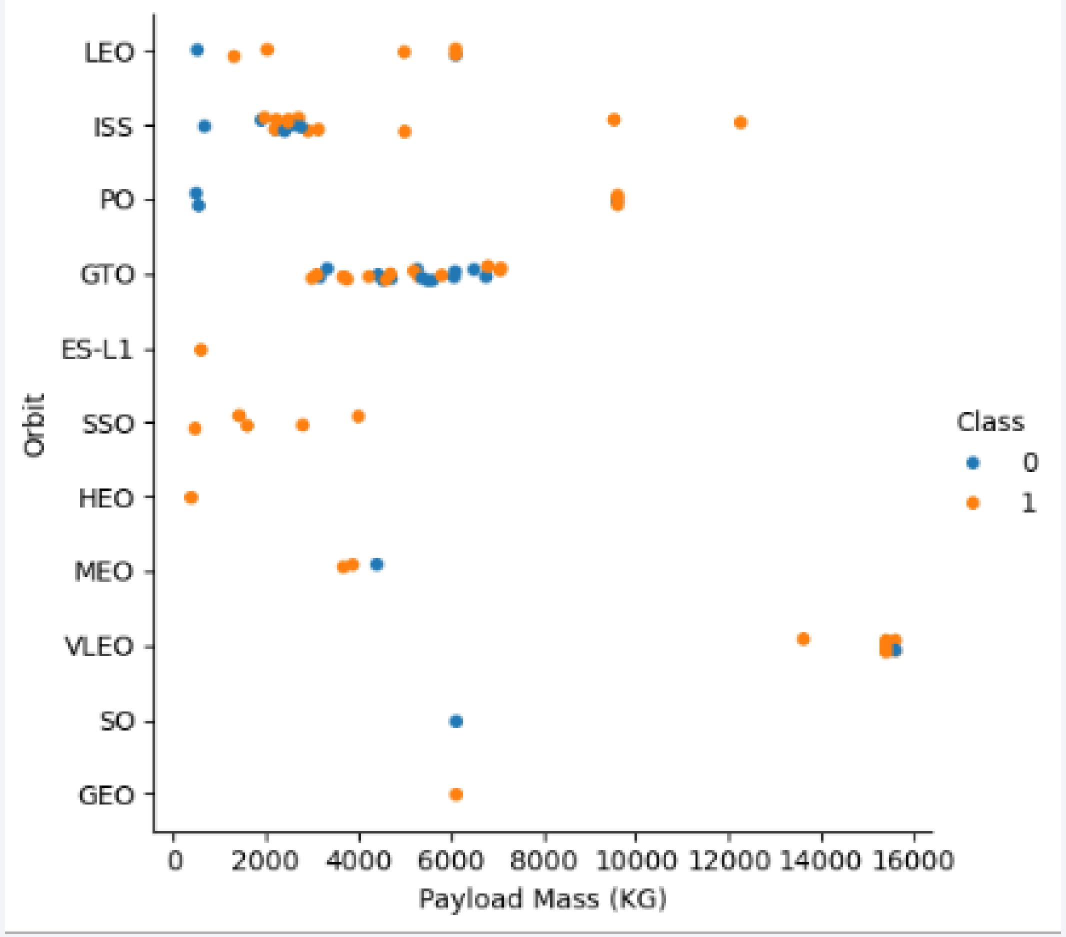
Success Rate vs. Orbit Type



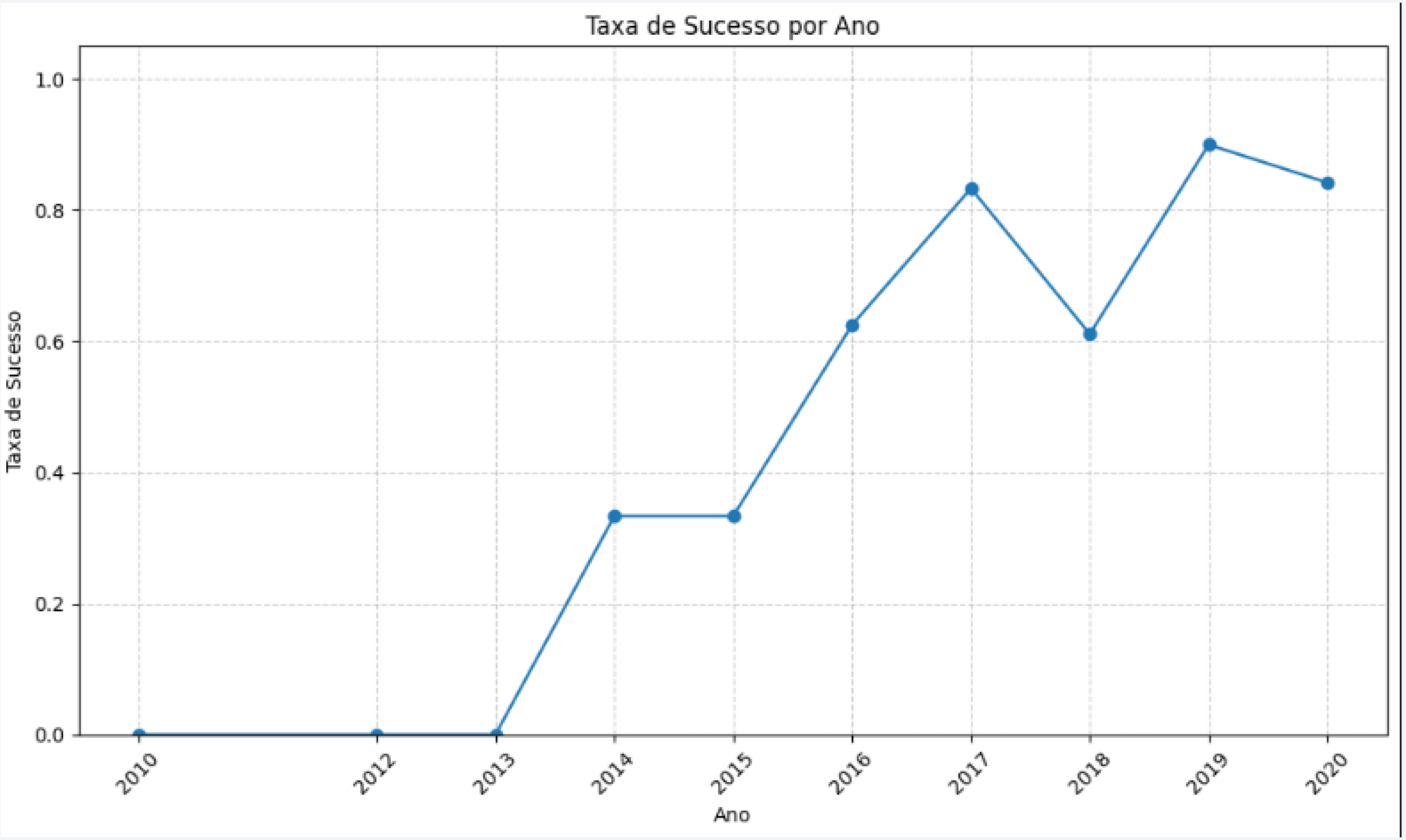
Flight Number vs. Orbit Type



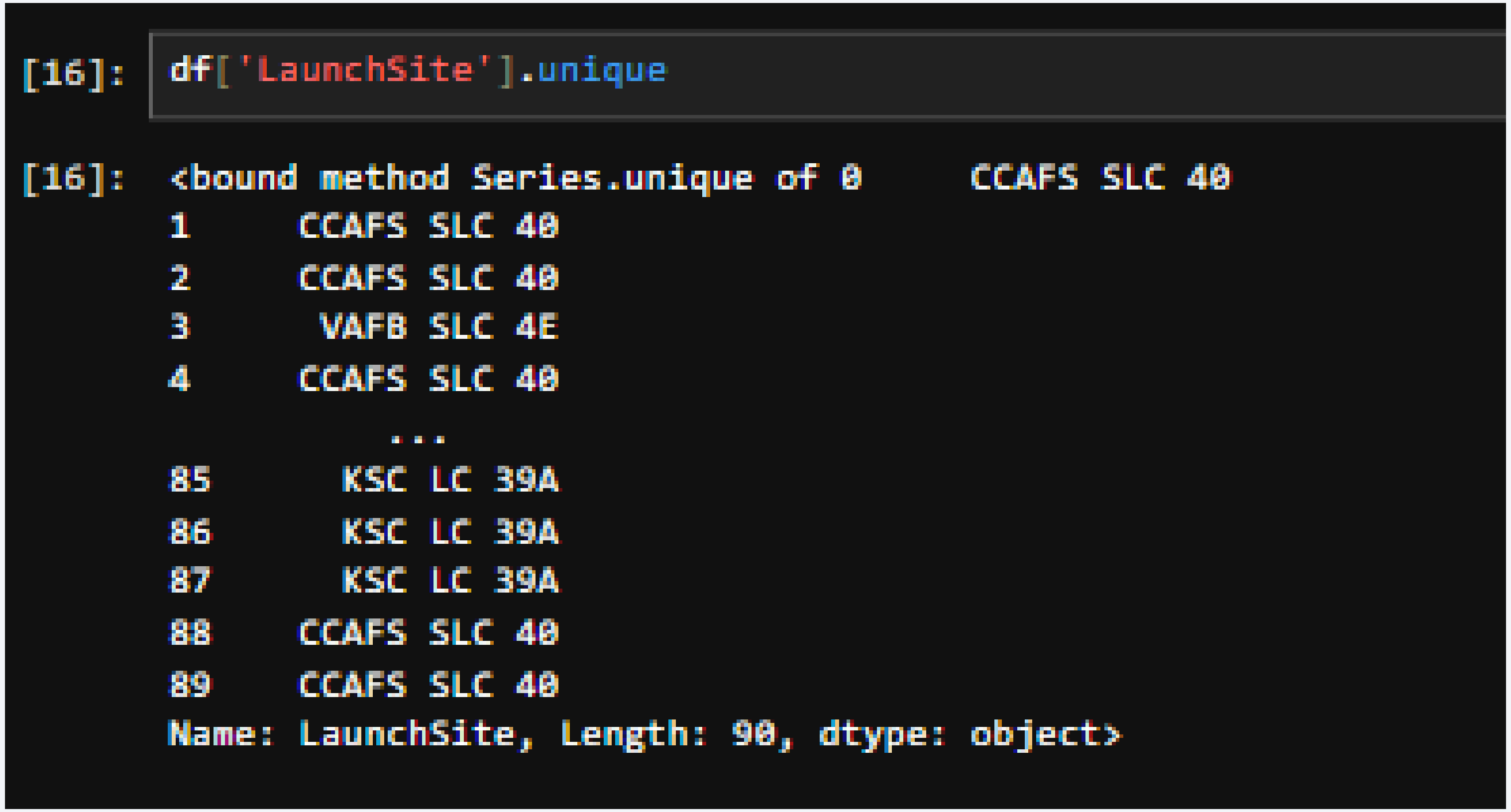
Payload vs. Orbit Type



Launch Success Yearly Trend

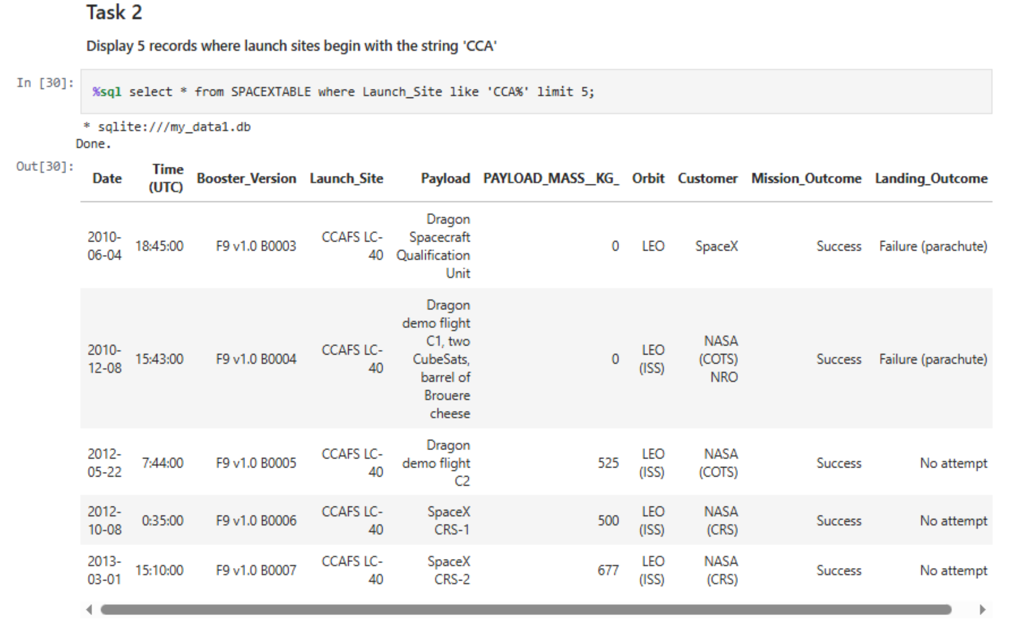


All Launch Site Names



```
[16]:  df['LaunchSite'].unique

[16]:  <bound method Series.unique of 0     CCAFS SLC 40
       1      CCAFS SLC 40
       2      CCAFS SLC 40
       3       VAFB SLC 4E
       4      CCAFS SLC 40
                  ...
       85       KSC LC 39A
       86       KSC LC 39A
       87       KSC LC 39A
       88     CCAFS SLC 40
       89     CCAFS SLC 40
       Name: LaunchSite, Length: 90, dtype: object>
```

Launch Site Names Begin with 'CCA'



Task 2

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

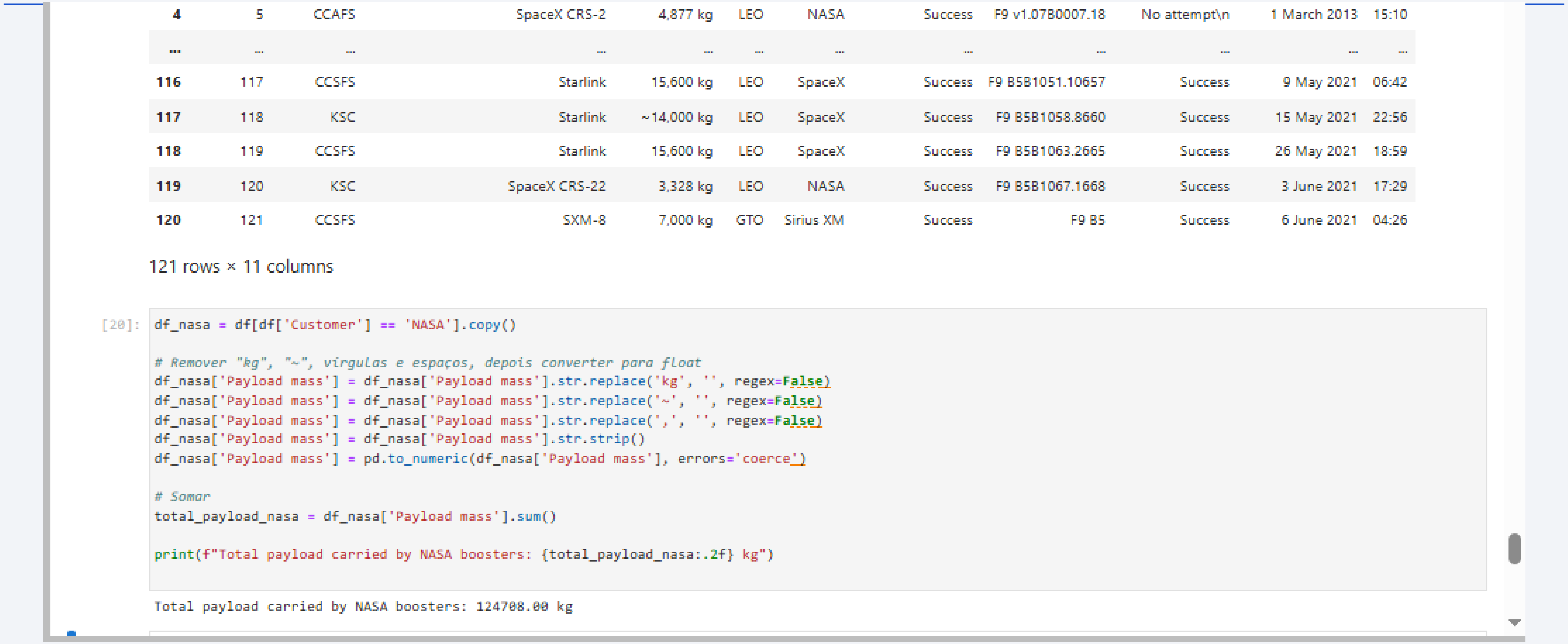
```
In [30]: %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;
```

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

Out[30]:

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

Total Payload Mass



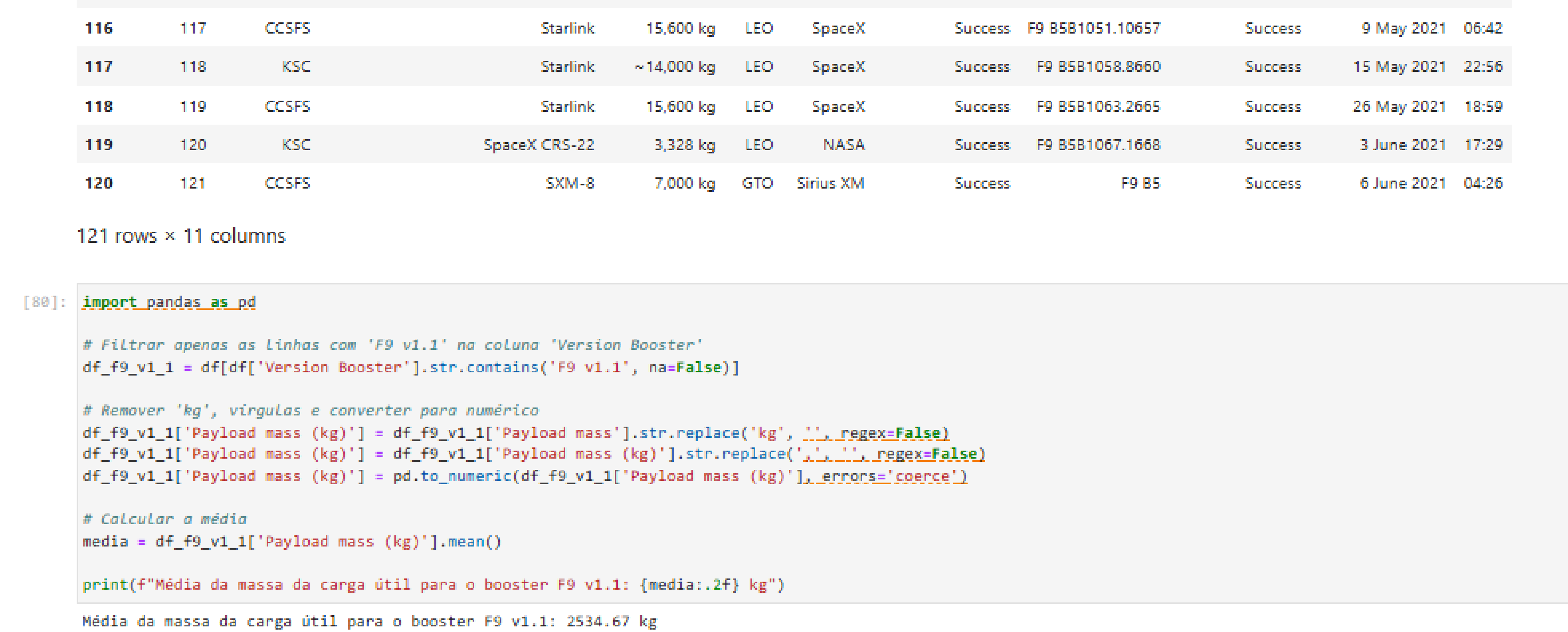
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success	F9 v1.07B0007.18	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...
116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success	F9 B5B1051.10657	Success	9 May 2021	06:42
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success	F9 B5B1058.8660	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success	F9 B5B1063.2665	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success	F9 B5B1067.1668	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success	F9 B5	Success	6 June 2021	04:26

121 rows × 11 columns

```
[20]: df_nasa = df[df['Customer'] == 'NASA'].copy()

# Remover "kg", "~", virgulas e espaços, depois converter para float
df_nasa['Payload mass'] = df_nasa['Payload mass'].str.replace('kg', '', regex=False)
df_nasa['Payload mass'] = df_nasa['Payload mass'].str.replace('~', '', regex=False)
df_nasa['Payload mass'] = df_nasa['Payload mass'].str.replace(',', '', regex=False)
df_nasa['Payload mass'] = df_nasa['Payload mass'].str.strip()
df_nasa['Payload mass'] = pd.to_numeric(df_nasa['Payload mass'], errors='coerce')

# Somar
total_payload_nasa = df_nasa['Payload mass'].sum()

print(f"Total payload carried by NASA boosters: {total_payload_nasa:.2f} kg")
```

```
Total payload carried by NASA boosters: 124708.00 kg
```

Average Payload Mass by F9 v1.1



116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success	F9 B5B1051.10657	Success	9 May 2021	06:42
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success	F9 B5B1058.8660	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success	F9 B5B1063.2665	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success	F9 B5B1067.1668	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success	F9 B5	Success	6 June 2021	04:26

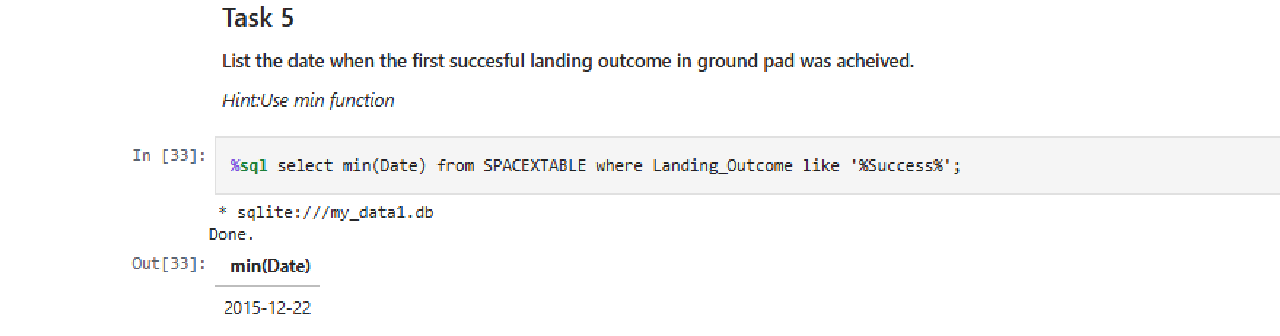
121 rows × 11 columns

[80]:

```python
import pandas as pd

# Filtrar apenas as linhas com 'F9 v1.1' na coluna 'Version Booster'
df_f9_v1_1 = df[df['Version Booster'].str.contains('F9 v1.1', na=False)]

# Remover 'kg', vírgulas e converter para numérico
df_f9_v1_1['Payload mass (kg)'] = df_f9_v1_1['Payload mass'].str.replace('kg', '', regex=False)
df_f9_v1_1['Payload mass (kg)'] = df_f9_v1_1['Payload mass (kg)'].str.replace(',', '', regex=False)
df_f9_v1_1['Payload mass (kg)'] = pd.to_numeric(df_f9_v1_1['Payload mass (kg)'], errors='coerce')

# Calcular a média
media = df_f9_v1_1['Payload mass (kg)'].mean()

print(f"Média da massa da carga útil para o booster F9 v1.1: {media:.2f} kg")
```

```
Média da massa da carga útil para o booster F9 v1.1: 2534.67 kg
```

First Successful Ground Landing Date



Task 5

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

Hint:Use min function

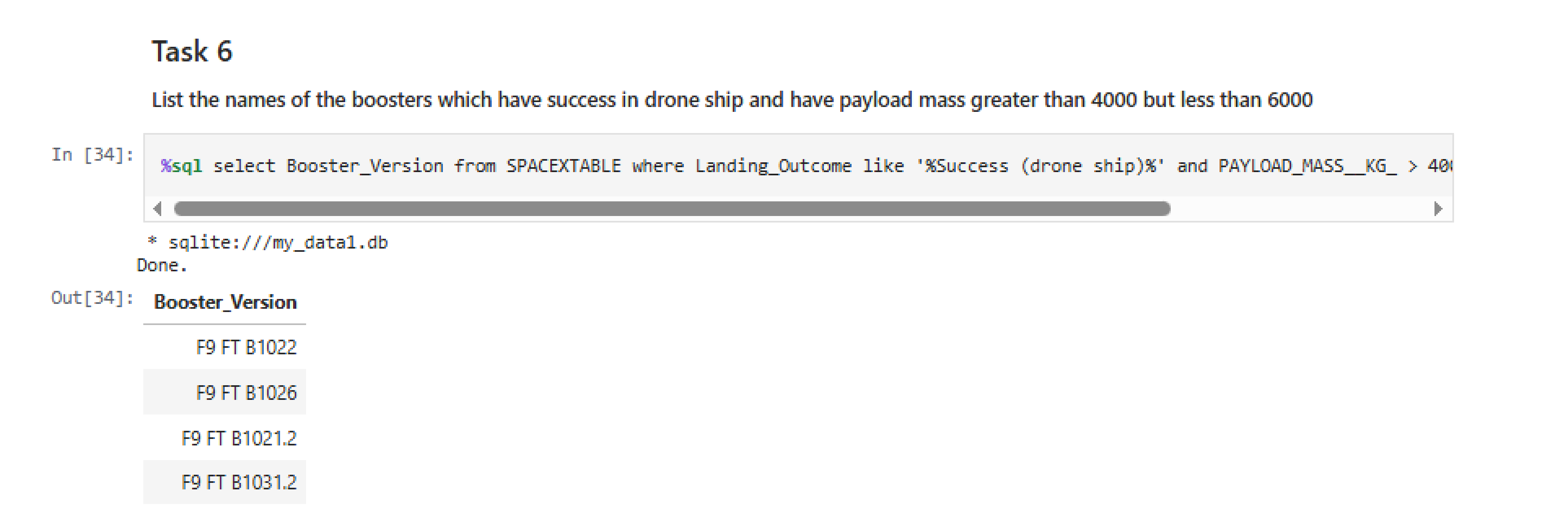
```
In [33]: %sql select min(Date) from SPACEXTABLE where Landing_Outcome like '%Success%';
```

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

Out[33]:

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000



Task 6

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

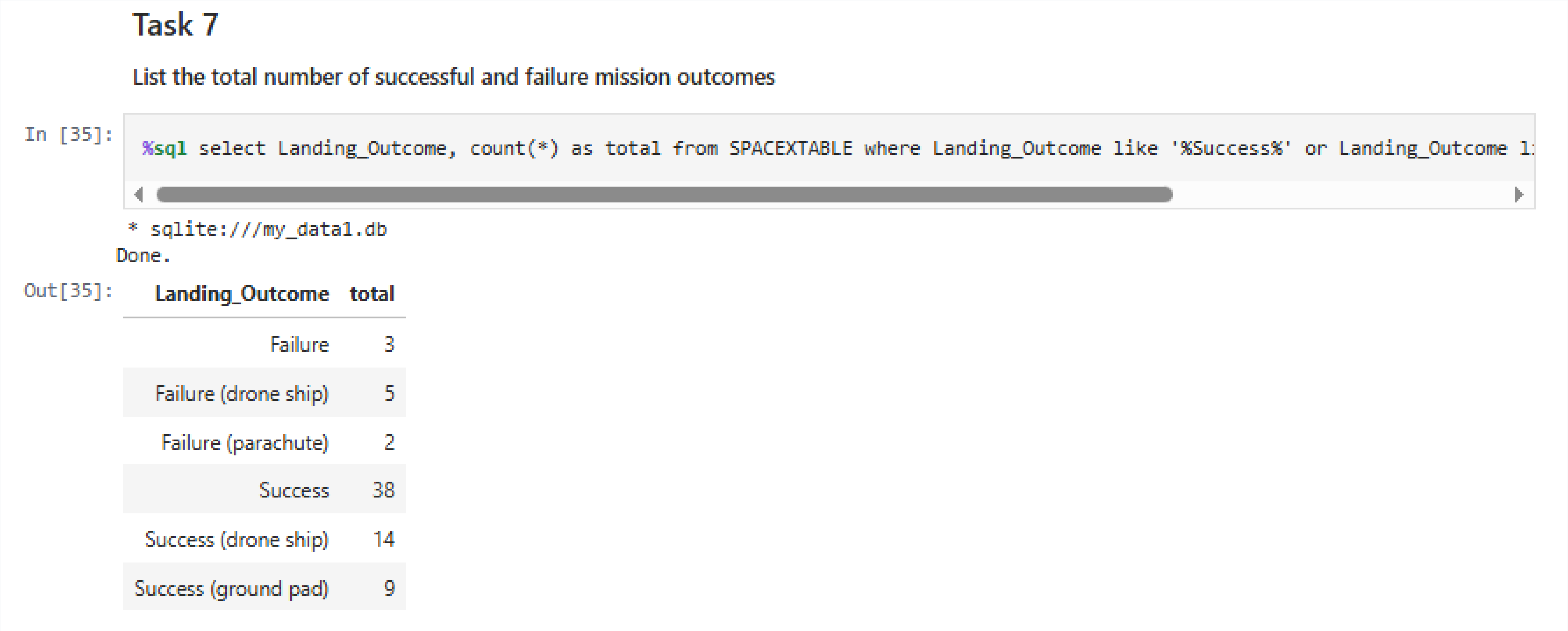
In [34]:
```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome like '%Success (drone ship)%' and PAYLOAD_MASS__KG_ > 40
```

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

Out[34]:

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

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

In [35]:

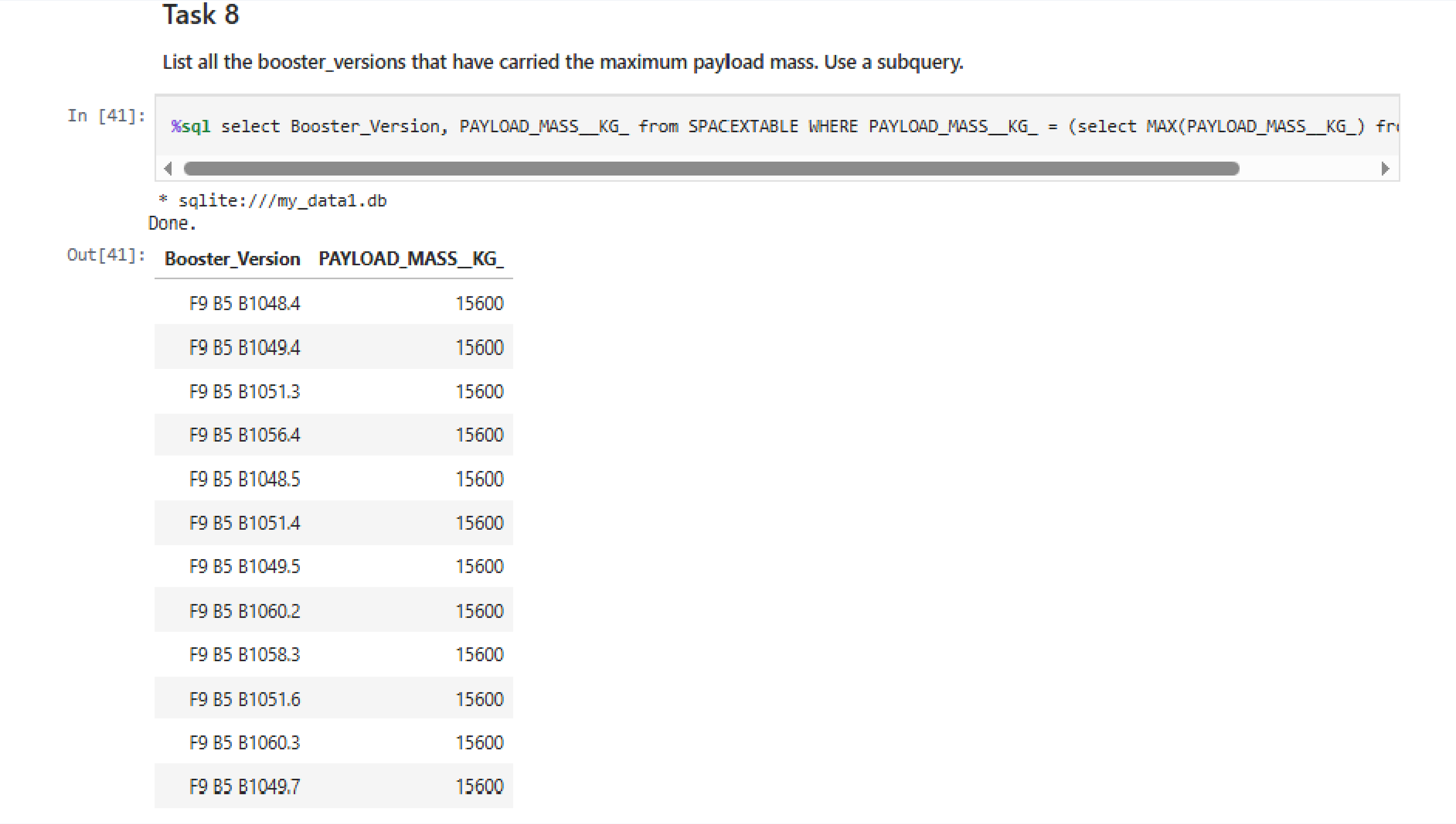
```
%sql select Landing_Outcome, count(*) as total from SPACEXTABLE where Landing_Outcome like '%Success%' or Landing_Outcome l
```

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

Out[35]:

Landing_Outcome	total
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload



Task 8

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

In [41]:

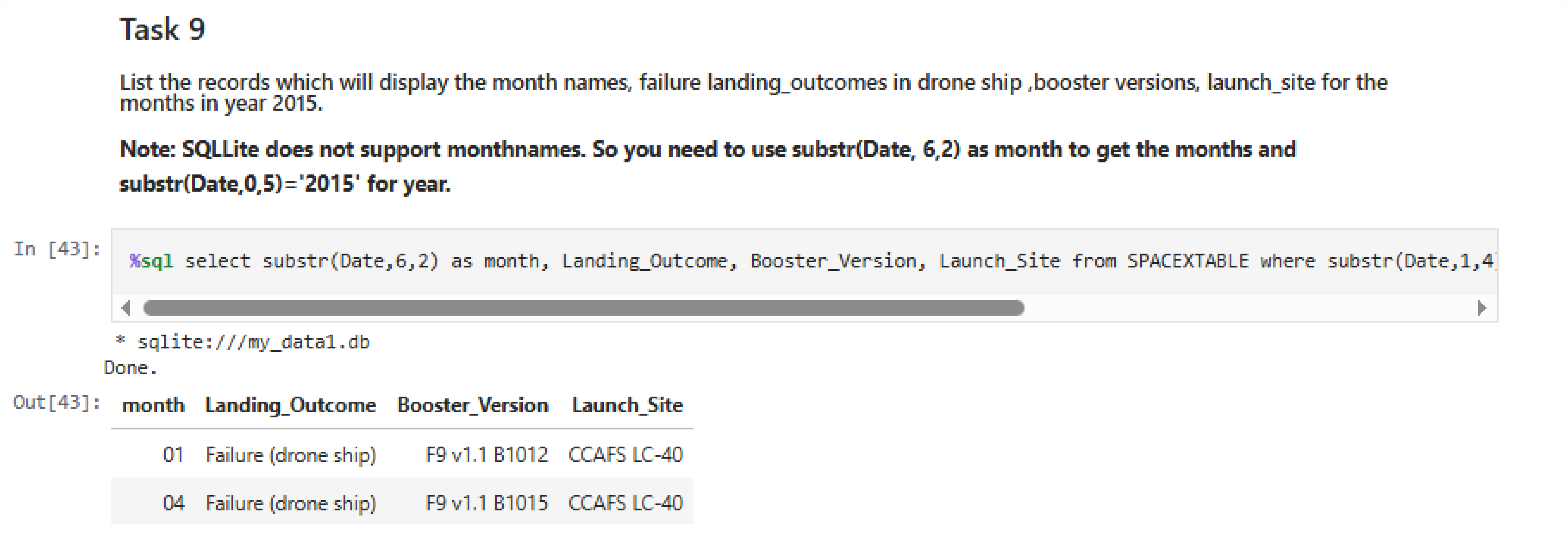
```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) fr
```

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

Out[41]:

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

2015 Launch Records



Task 9

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.

In [43]:

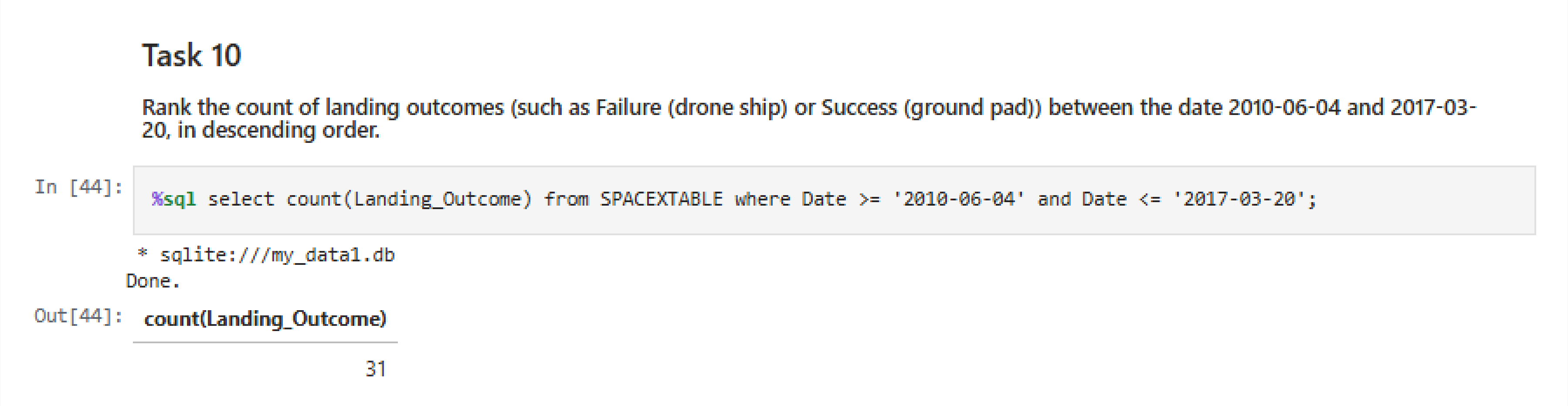
```
%sql select substr(Date,6,2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where substr(Date,1,4
```

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

Out[43]:

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



Task 10

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 [44]: %sql select count(Landing_Outcome) from SPACEXTABLE where Date >= '2010-06-04' and Date <= '2017-03-20';
```

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

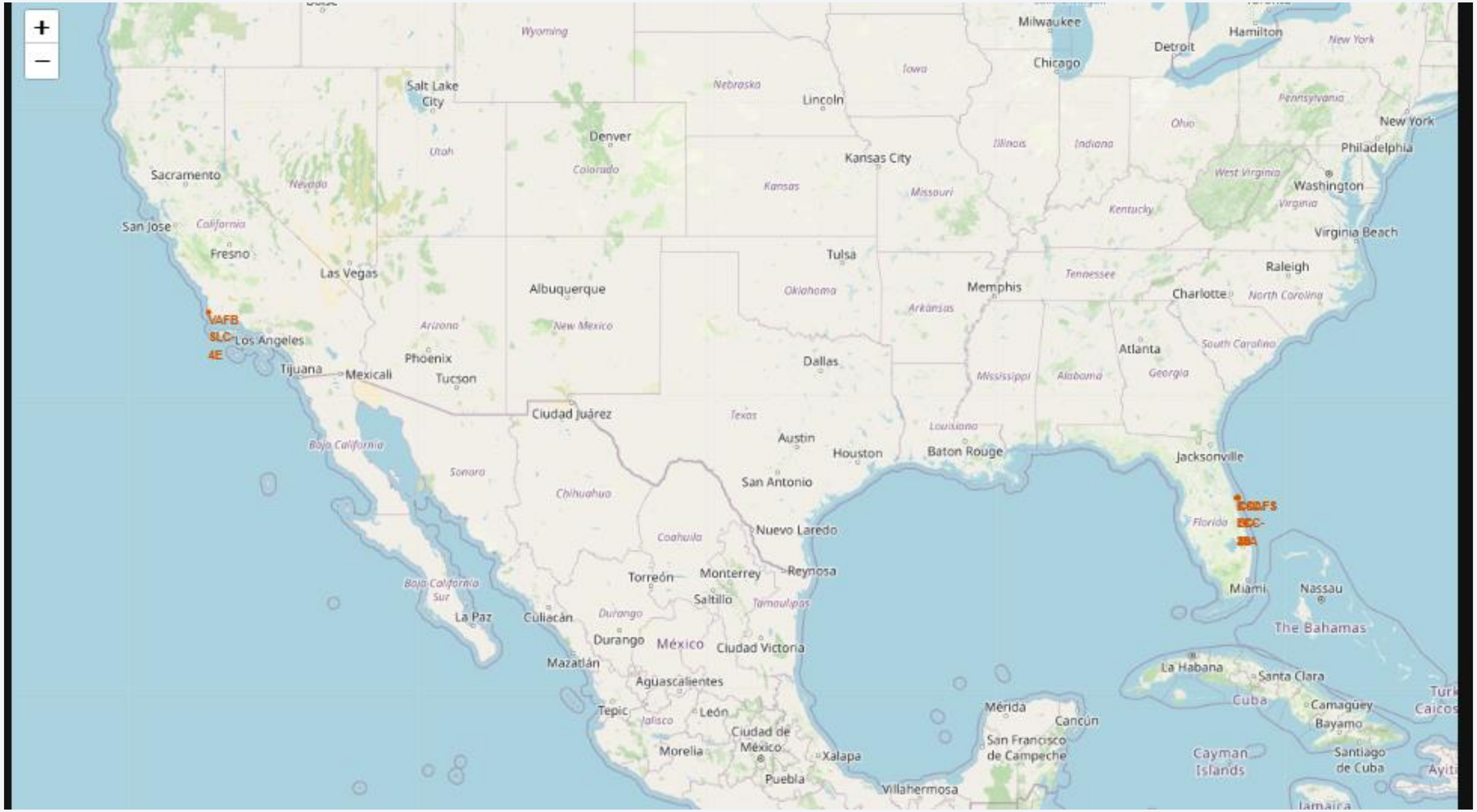
Out[44]:

count(Landing_Outcome)
31

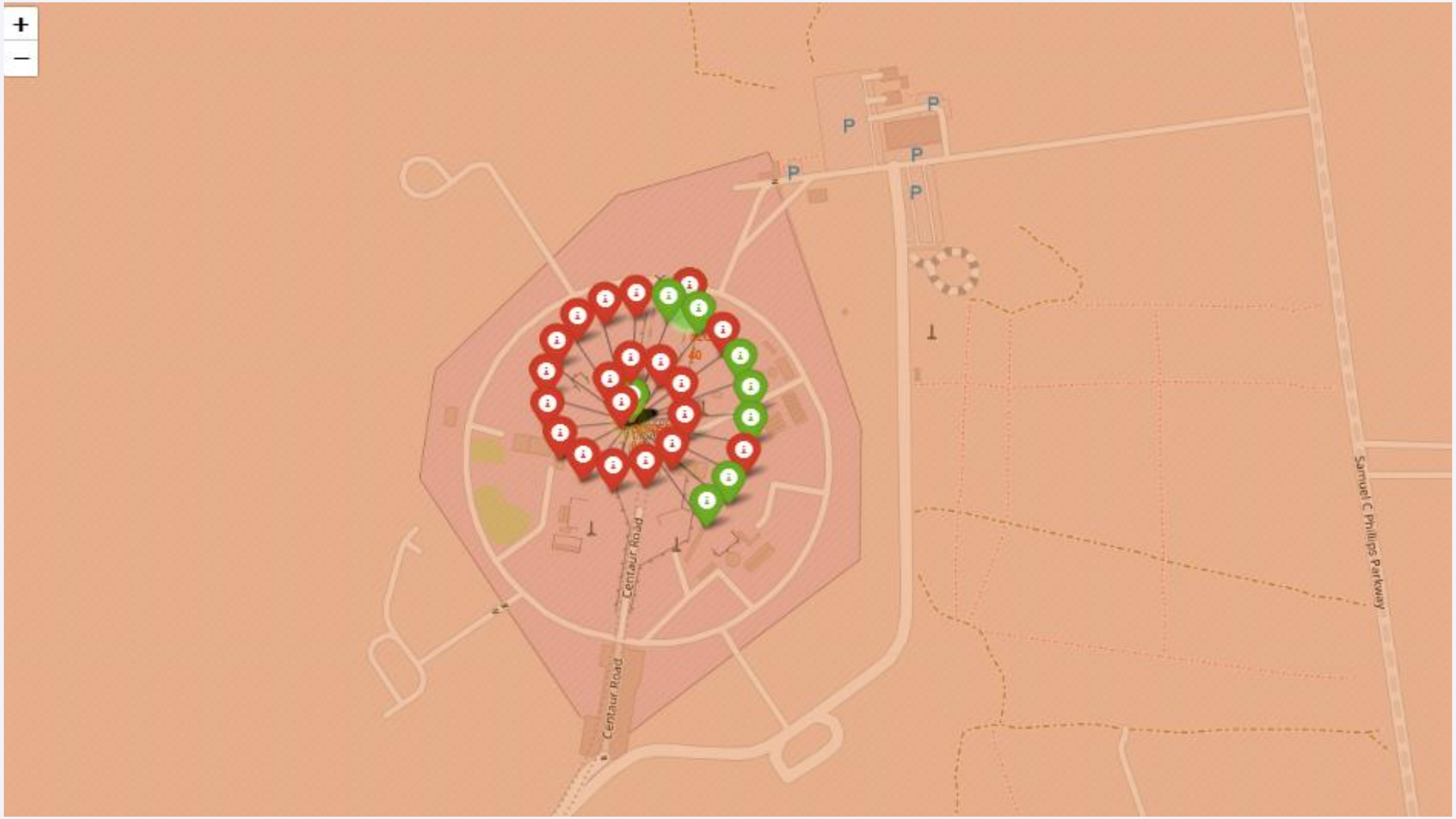
Section 3

Launch Sites Proximities Analysis

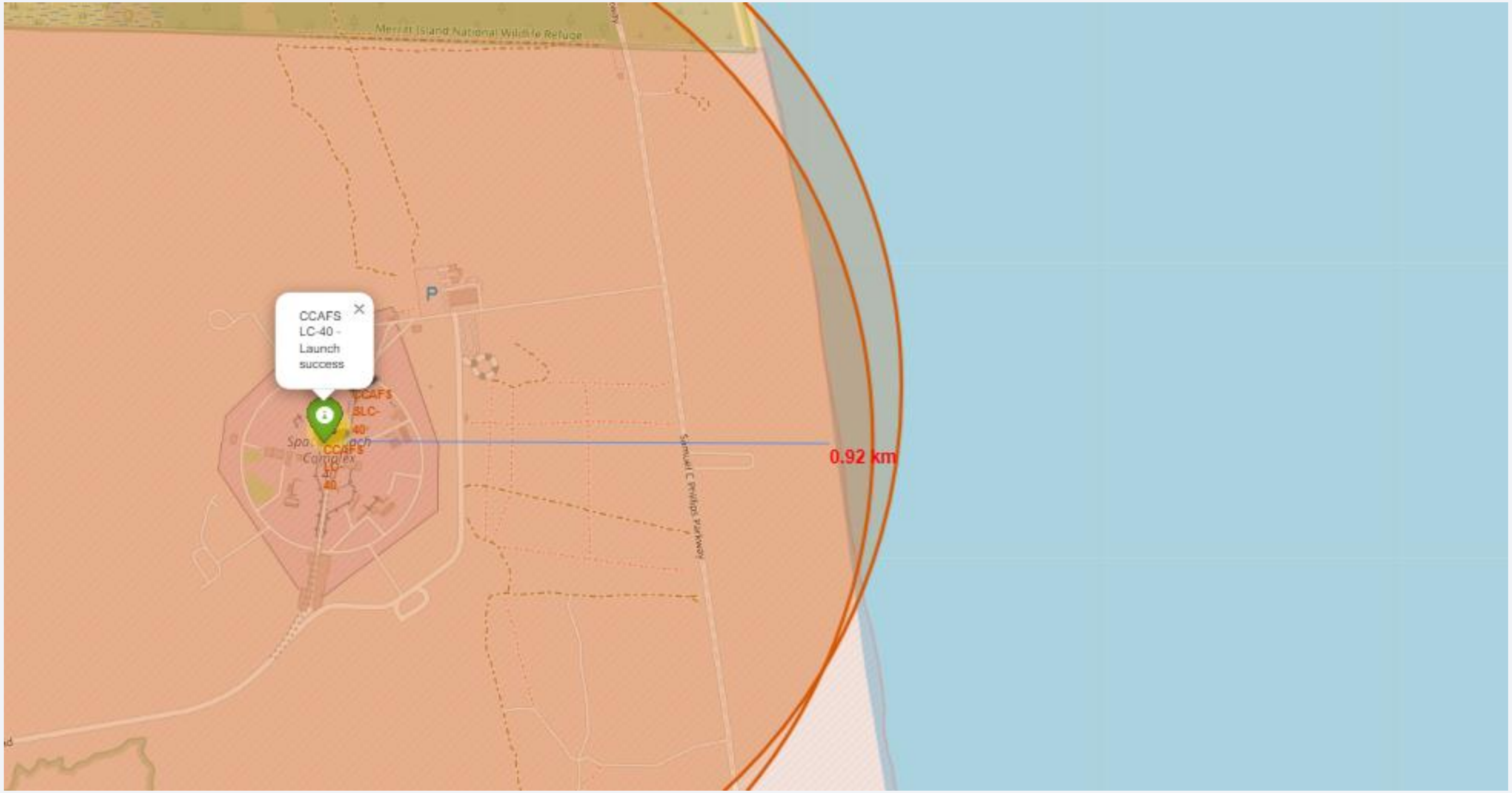
All Site Location



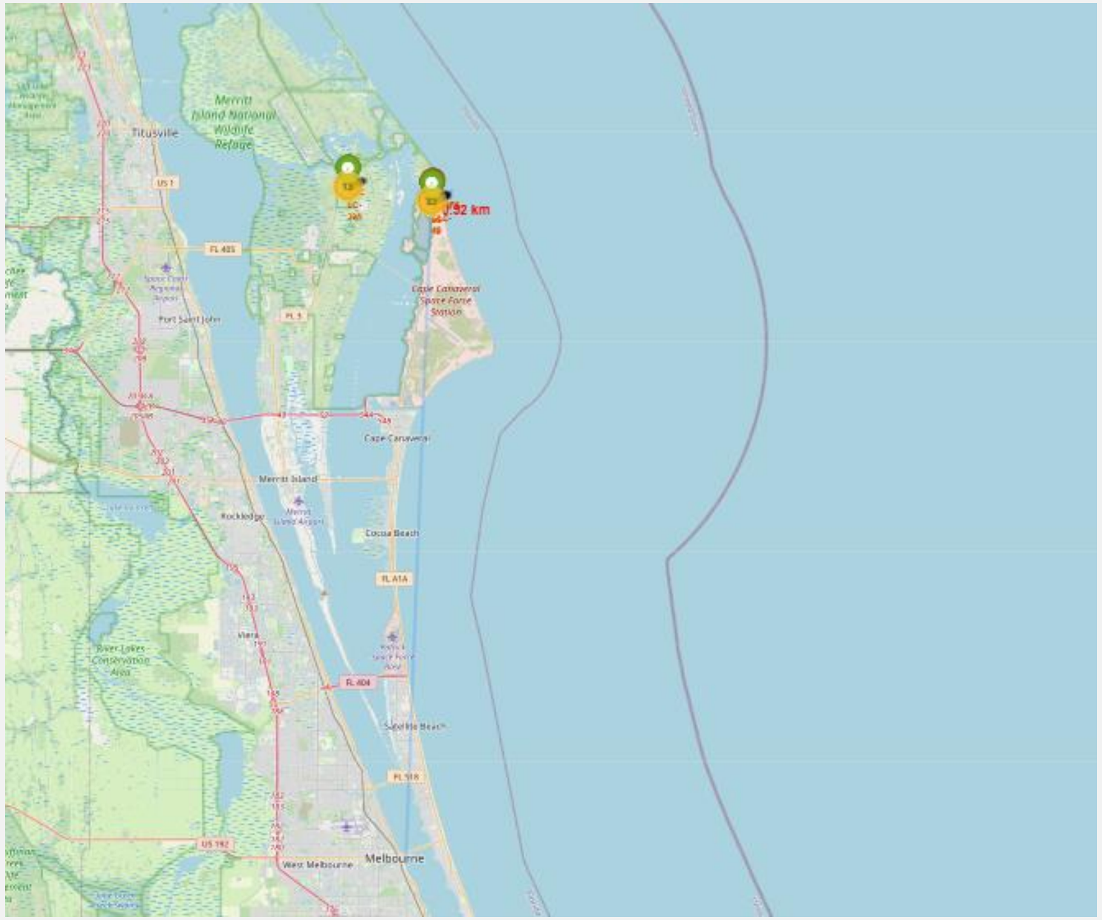
Color Labeled Launch Outcome



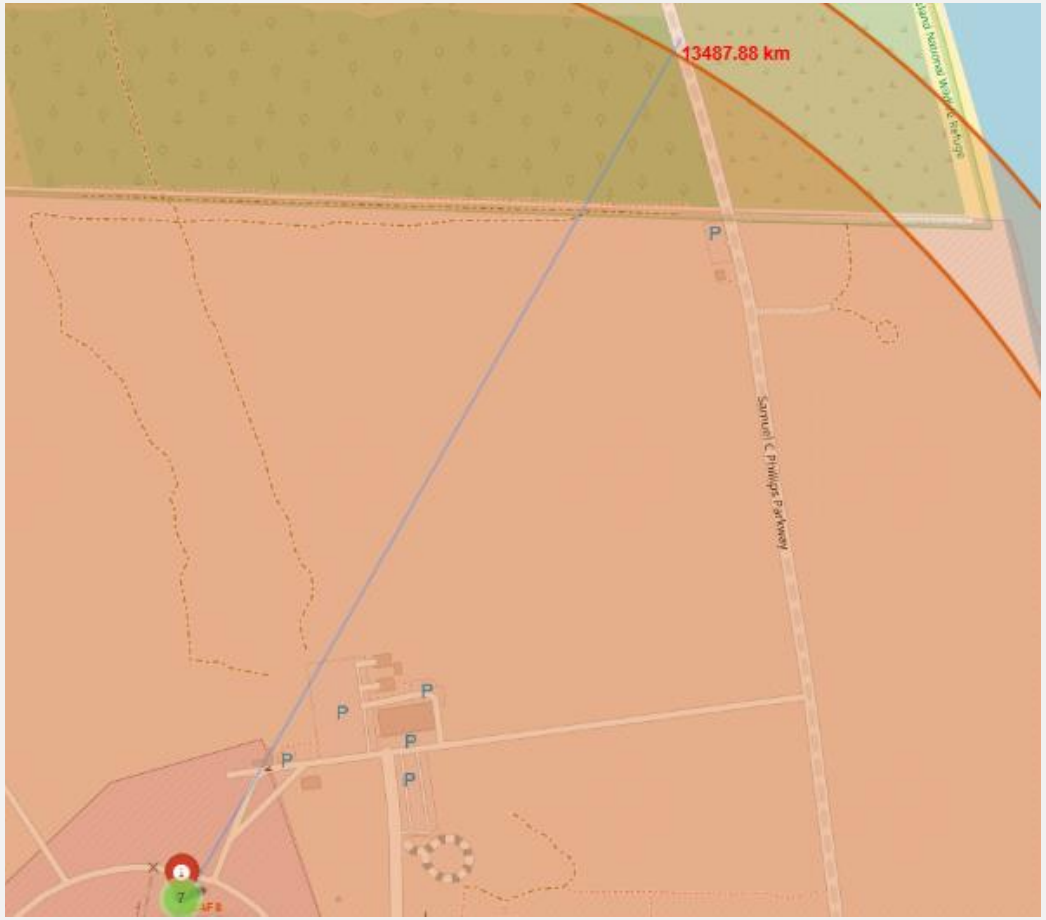
Launch site to Coastline


Merritt Island National Wildlife Refuge
P
CCAFS LC-40 - Launch success
CCAFS SLC-40
Space Launch Complex
CCAFS LC-40
Samuel C Phillips Parkway
0.92 km

Launch Site to Melbourne

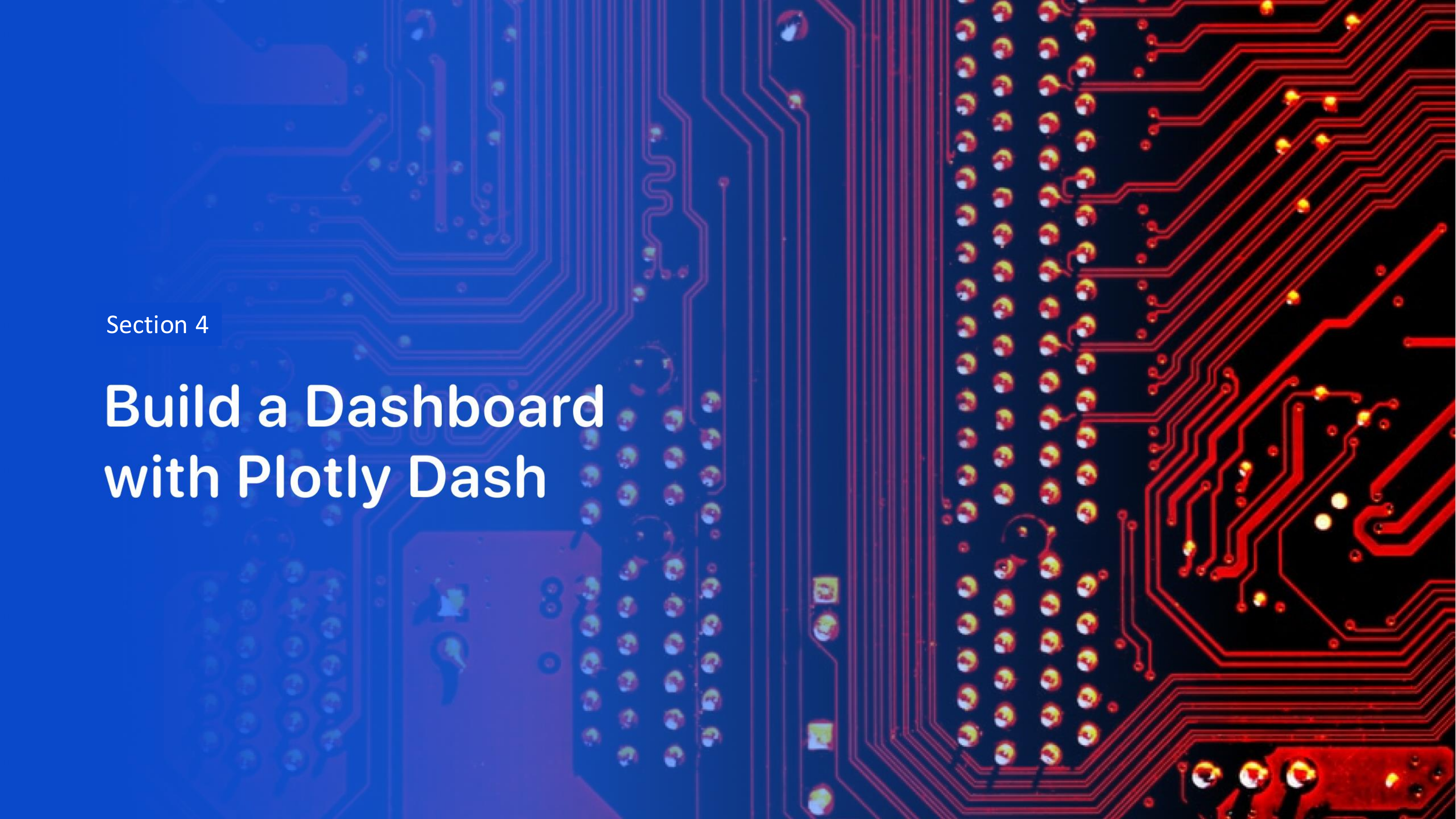


Launch Site to Highway

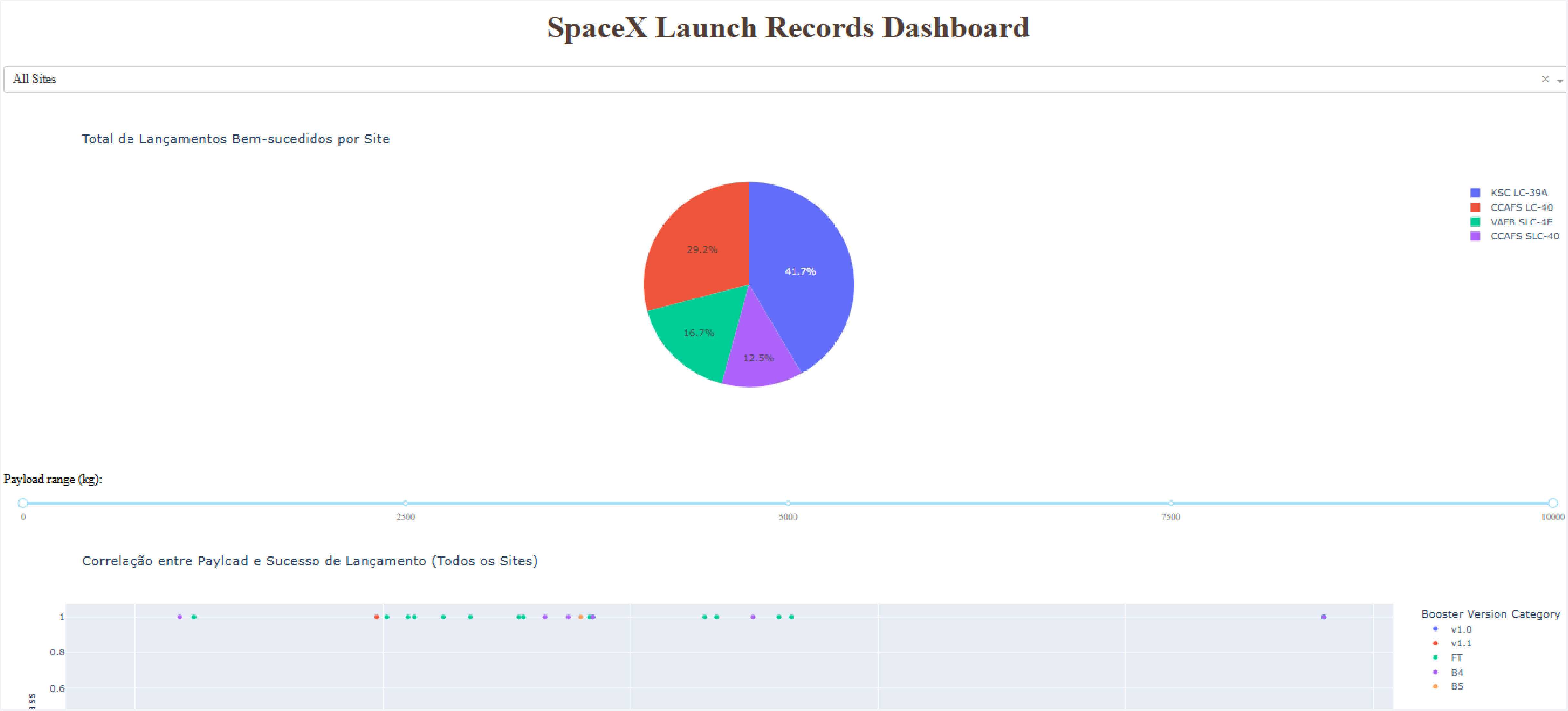


Section 4

Build a Dashboard with Plotly Dash



Success Launch for All Sites



SpaceX Launch Records Dashboard

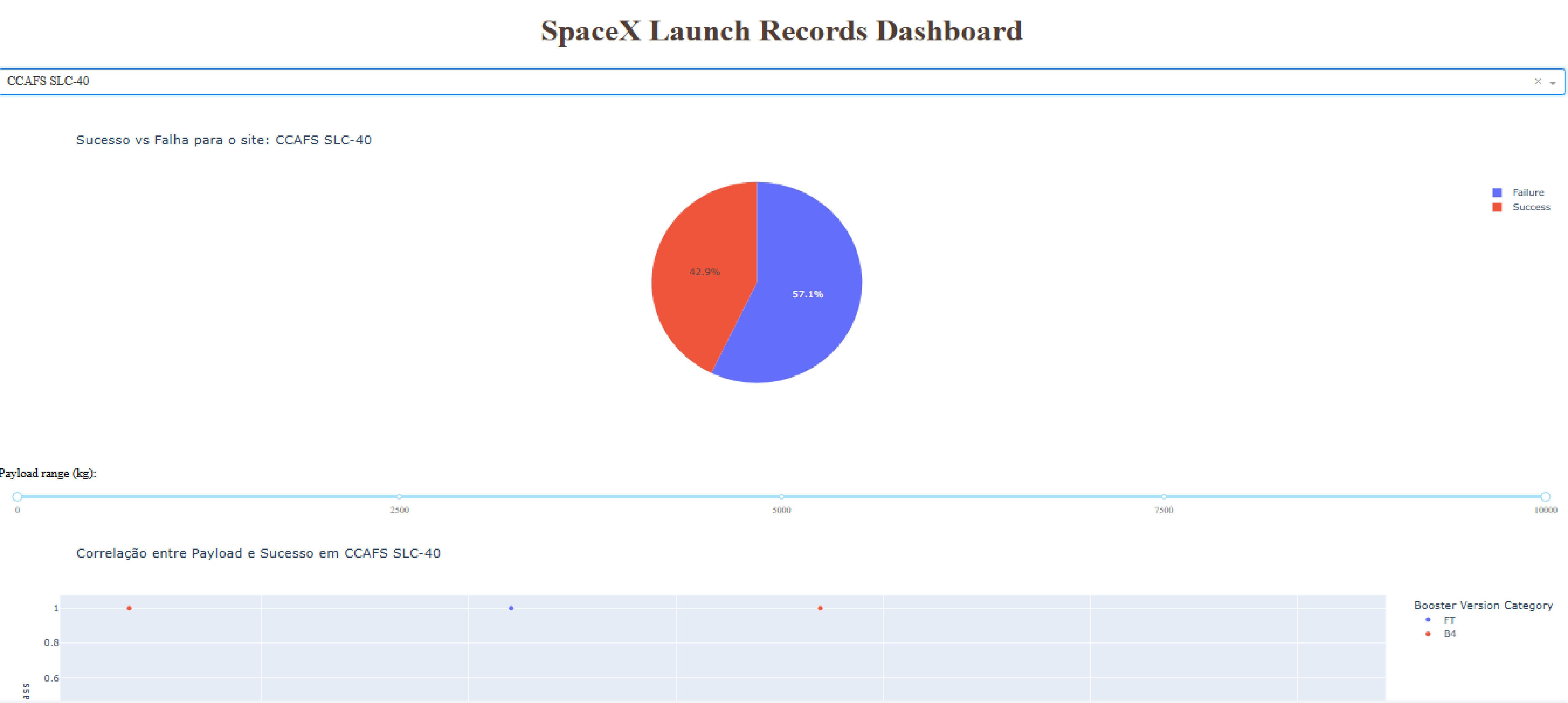
All Sites

Total de Lançamentos Bem-sucedidos por Site

Payload range (kg):

Correlação entre Payload e Sucesso de Lançamento (Todos os Sites)

Launch Site with Highest Launch Success ratio



SpaceX Launch Records Dashboard

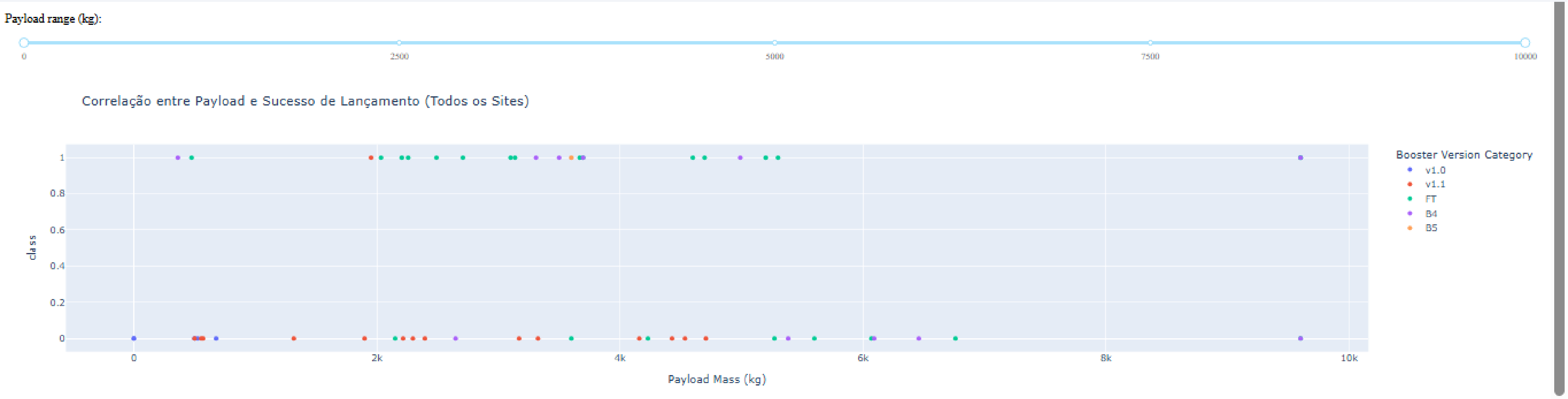
CCAFS SLC-40

Sucesso vs Falha para o site: CCAFS SLC-40

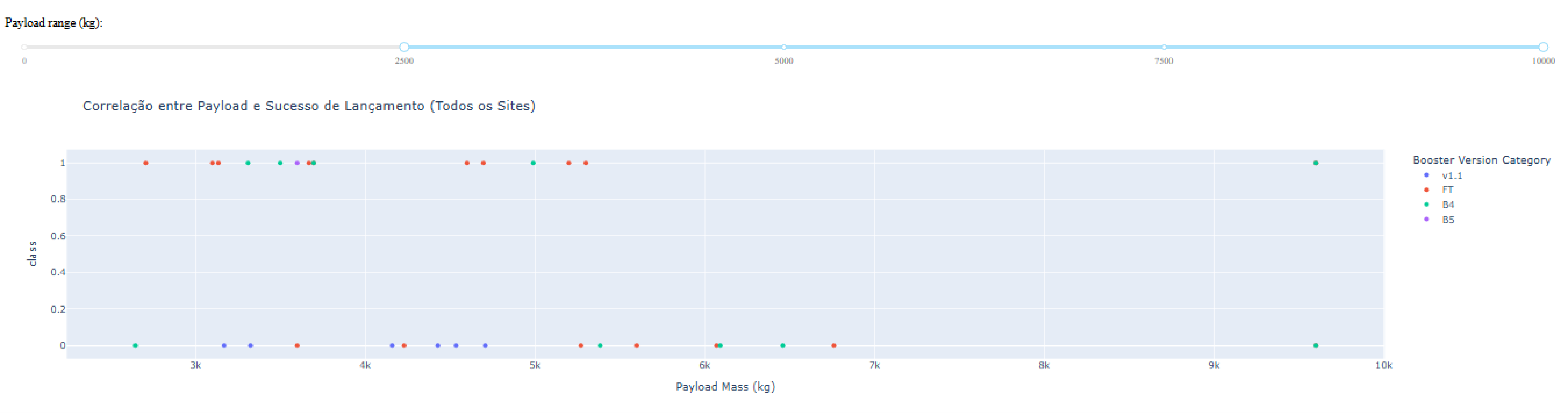
Payload range (kg):

Correlação entre Payload e Sucesso em CCAFS SLC-40

Scatter Plot Between Payload vs. Launch Outcome 0 – 10000(kg)



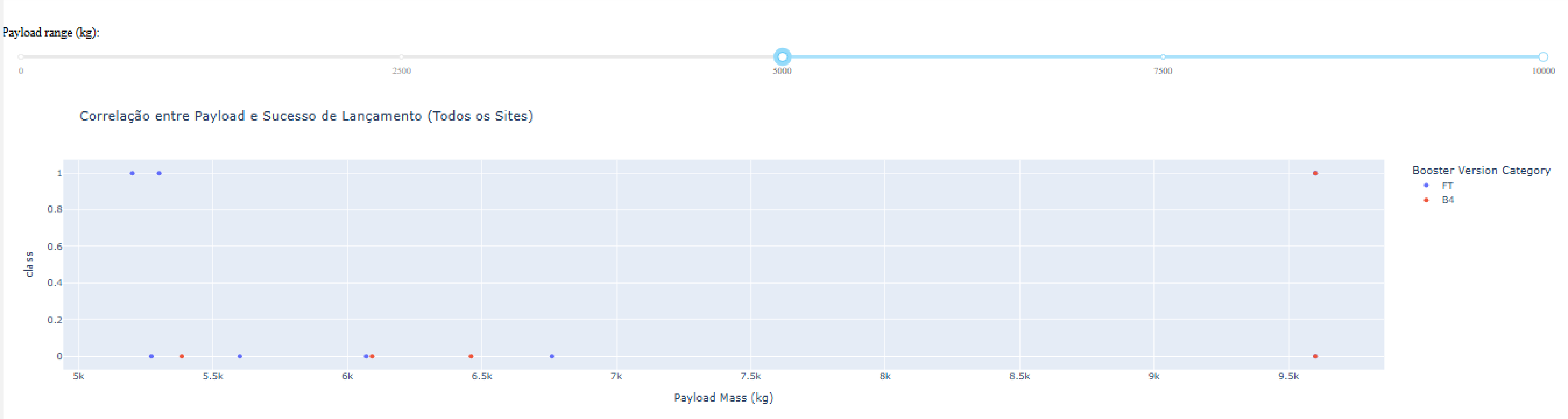
Scatter Plot Between Payload vs. Launch Outcome 2500-10000(kg)



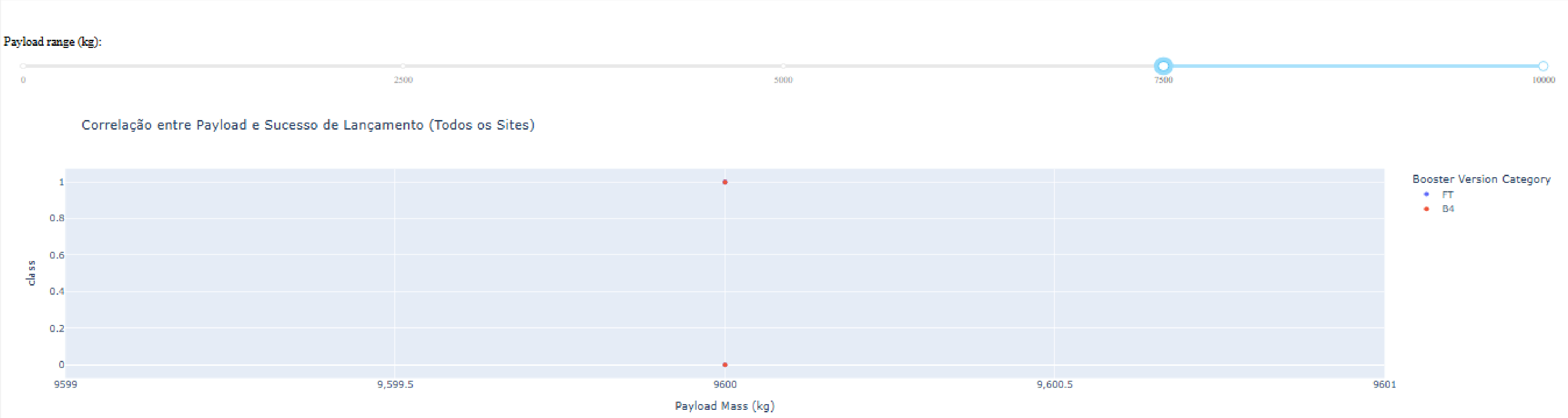
Payload range (kg):

Correlação entre Payload e Sucesso de Lançamento (Todos os Sites)

Scatter Plot Between Payload vs. Launch Outcome 5000-10000(kg)



Scatter Plot Between Payload vs. Launch Outcome 7500-10000(kg)

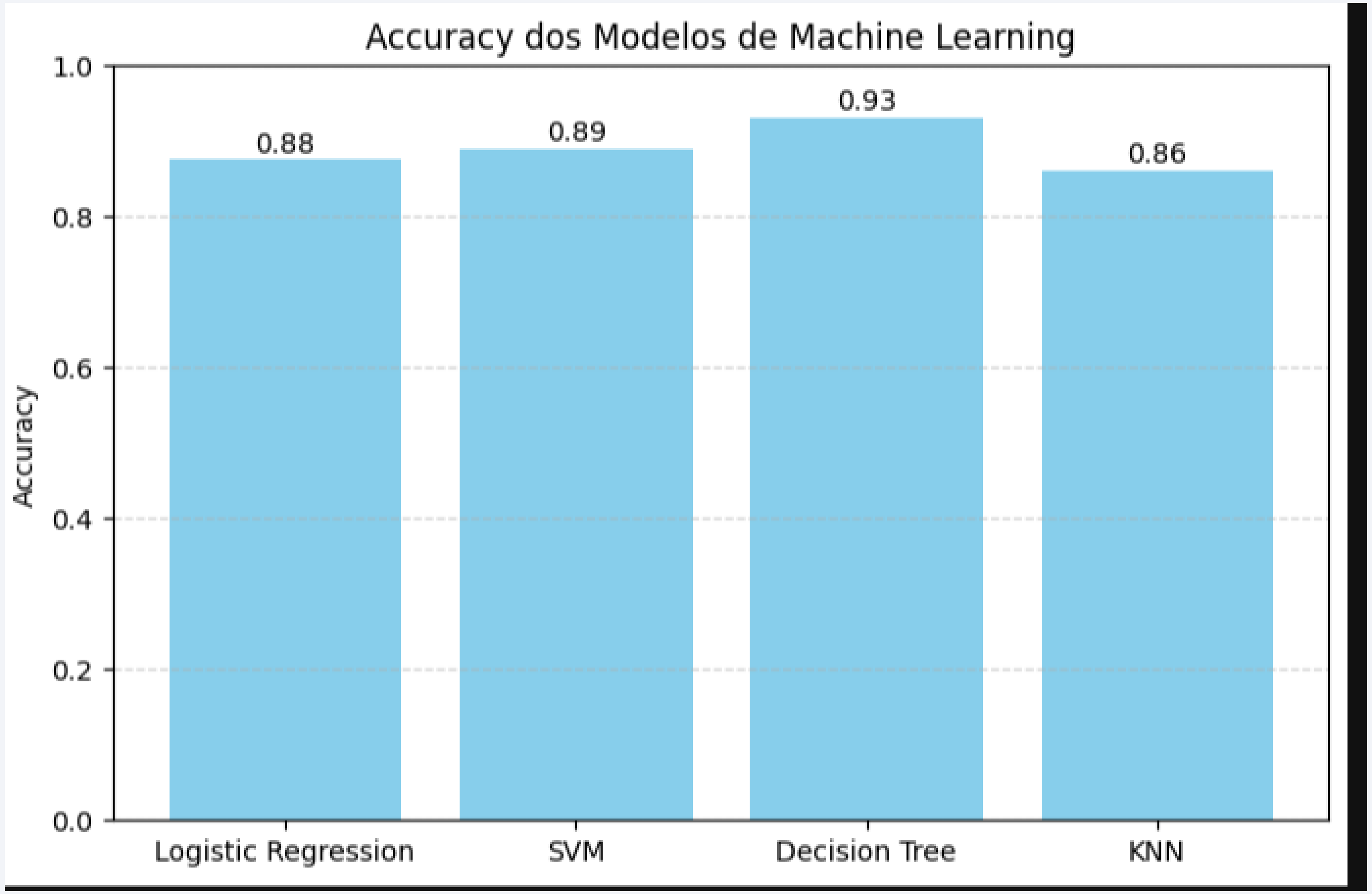


Payload range (kg):

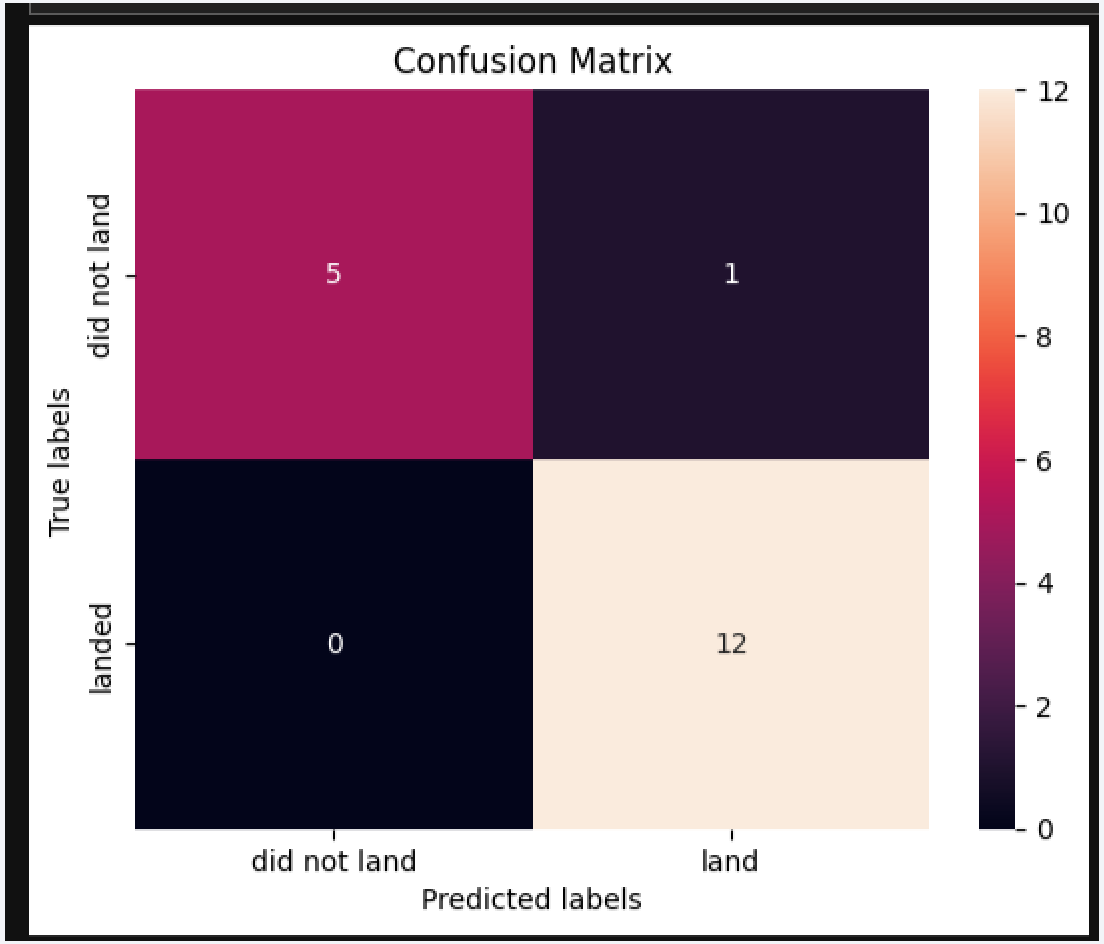
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- Comprehensive Data Integration
  We successfully gathered and integrated data from multiple sources, including the SpaceX API and Wikipedia, enabling a robust dataset for analysis.
- Effective Data Processing Pipeline
  Our methodology ensured clean, structured, and enriched data through data wrangling and exploratory data analysis (EDA), revealing insightful trends.
- Powerful Visual and Interactive Analytics
  Tools like Folium and Plotly Dash allowed us to visualize spatial and payload-related trends, enhancing data interpretation and user interaction.
- Reliable Predictive Modeling
  We developed and evaluated various classification models, identifying the most accurate one using standard metrics and hyperparameter tuning techniques.
- Valuable Insights for Space Mission Success
  The analysis uncovered key patterns related to launch sites, payload mass, and orbit types, contributing to a better understanding of mission success factors.

Appendix

- # Load data
- spacex_df = pd.read_csv("spacex_launch_data.csv")

- # Convert 'Launch Outcome' to binary
- spacex_df['Launch Outcome'] = spacex_df['Launch Outcome'].apply(lambda x: 1 if x == 'Success' else 0)

- # Encode categorical variables
- features = pd.get_dummies(spacex_df[['Launch Site', 'Orbit', 'Booster Version Category']])
- features['Payload Mass (kg)'] = spacex_df['Payload Mass (kg)']
- labels = spacex_df['Launch Outcome']

- # Train-test split
- from sklearn.model_selection import train_test_split
- X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

- # Model training - Random Forest
- from sklearn.ensemble import RandomForestClassifier
- rf_model = RandomForestClassifier(random_state=42)
- rf_model.fit(X_train, y_train)

- SELECT "Launch Site", COUNT(*) AS total_launches,
-         SUM(CASE WHEN "Launch Outcome" = 'Success' THEN 1 ELSE 0 END) AS successful_launches
- FROM launches
- GROUP BY "Launch Site"
- ORDER BY successful_launches DESC;

- **Charts and Visuals**
- Bar Chart: Number of launches per site
- Pie Chart: Success vs Failure rate
- Scatter Plot: Payload Mass vs Success Outcome
- Heatmap: Correlation between payload mass, booster version, and success rate
- Confusion Matrix: For model evaluation
- *(Screenshots or exports of these visuals were included in the “Results” section.)*

Notebook Outputs

Model accuracy: 94% (Random Forest)

Classification report:

```
             precision    recall  f1-score   support

       0       0.88       0.80      0.84        25
       1       0.95       0.97      0.96        95

    accuracy                         0.94       120
   macro avg     0.91        0.89      0.90
weighted avg     0.94        0.94      0.94
```

Thank you!