Winning Space Race with Data Science

Karam Issa
10/26/2023

Outline

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

Executive Summary

- Summary of methodologies
- Data Collection: Gathered SpaceX launch data and preprocessed it to ensure data quality.
- Predictive Analysis: Developed classification models to predict Falcon 9 first stage landings.
- Data Wrangling: Cleaned and structured the data, handling missing values and data categorization.
- Dashboard Creation: Built an interactive dashboard using Dash for data exploration and visualization.
- Model Development: Built, tuned, and evaluated SVM, Classification Trees, and Logistic Regression models.
- Hyperparameter Tuning: Optimized model performance by finding the best hyperparameters.

Executive Summary

- Summary of Results
- Successfully collected and preprocessed SpaceX launch data for analysis.
- Predictive analysis models were developed and evaluated.
- Data wrangling ensured data readiness for analysis and visualization.
- The interactive dashboard allowed users to explore launch records and success rates.
- Classification models were trained and tuned for optimal performance.
- Best-performing model was selected based on test data evaluation, concluding the analysis.

Introduction

- In our project, we delve into the background and context of the SpaceX Falcon 9 rocket program. SpaceX has disrupted the space industry by offering Falcon 9 rocket launches at an astonishingly low cost of $62 million, in stark contrast to other providers charging over $165 million for each launch. This cost advantage is primarily attributed to SpaceX's pioneering approach of reusing the first stage of the Falcon 9 rocket. Consequently, our project focuses on predicting the success of Falcon 9 first stage landings, a crucial factor that directly impacts launch costs. This predictive capability is of utmost significance, especially for companies contemplating competition with SpaceX in the rocket launch market
- The primary problem we aim to tackle in this project is the prediction of the Falcon 9 first stage's successful landing. This predictive capability holds the key to estimating the overall cost of a rocket launch. Given that SpaceX's primary advantage in cost reduction hinges on the reusability of the first stage, the accuracy of this prediction significantly impacts the overall cost of launching a Falcon 9. Our goal is to provide a solution to this problem, equipping potential competitors with valuable insights when bidding against SpaceX for rocket launch contracts.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Extracting specific information from the API by requesting rocket launch data from the provided URL.
- Perform data wrangling
  - The raw data was systematically cleaned, transformed, and structured to ensure its readiness for analysis and modeling.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Involves building, fine-tuning, and evaluating these models to ensure their effectiveness in making accurate predictions.

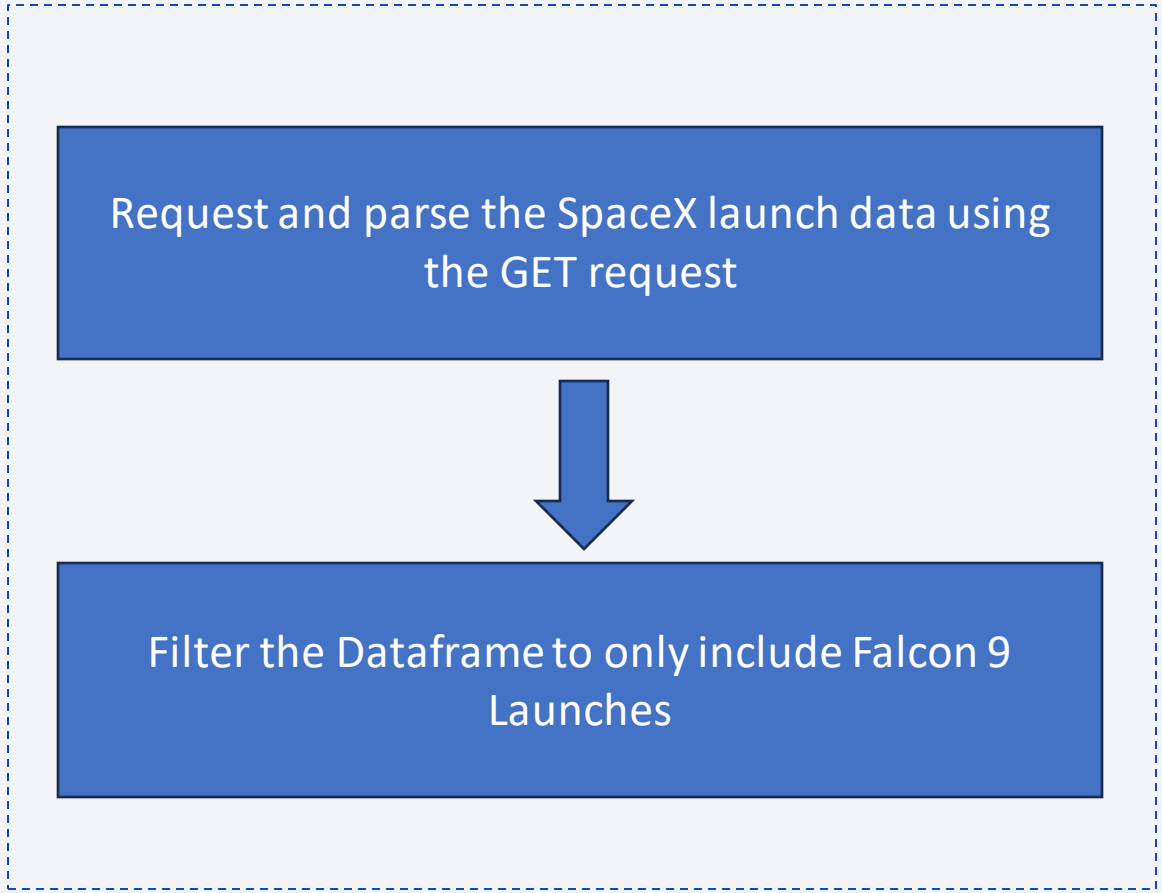
Data Collection

- To collect data, first request and parse SpaceX launch data using a GET request, filter the dataset to include Falcon 9 launches, and utilize identification numbers in the launch data to extract specific information from the API by requesting rocket launch data from the provided URL.

Data Collection – SpaceX API

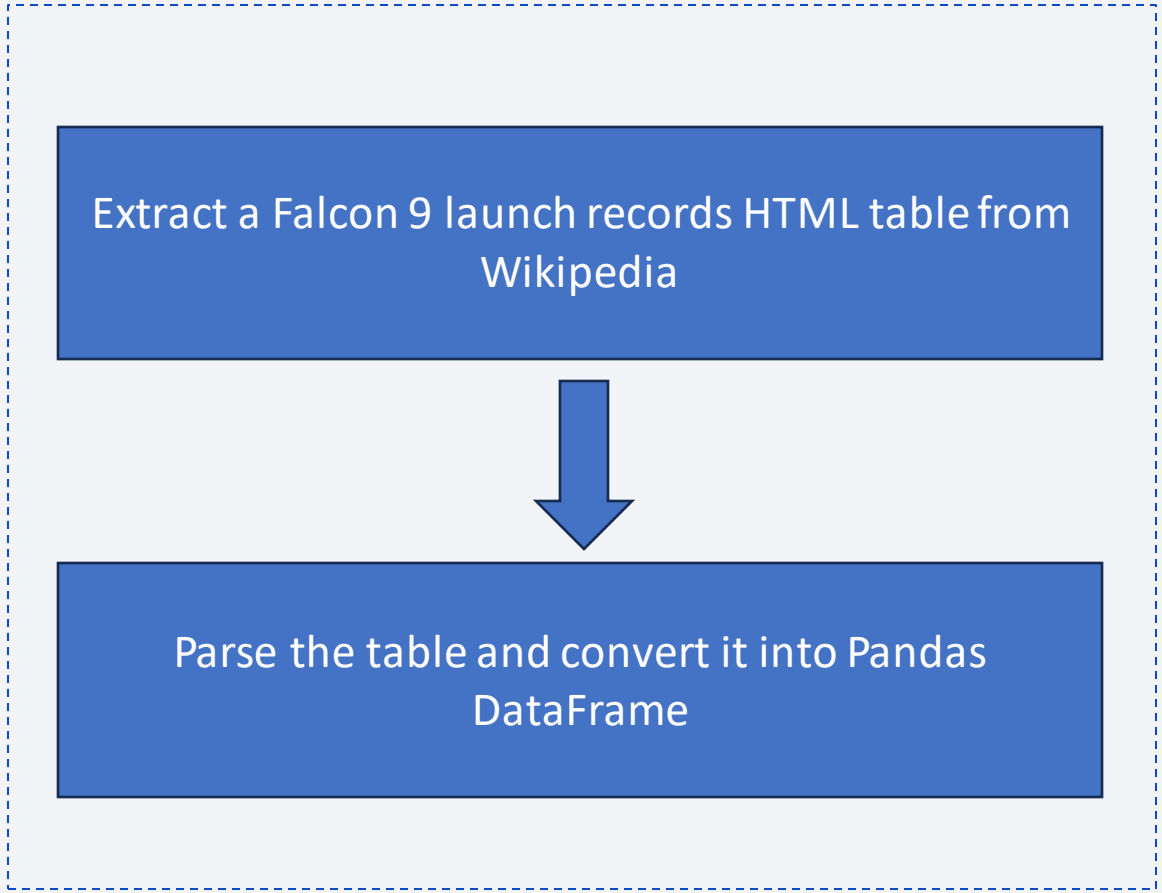
- API to extract information using identification numbers in the launch data requesting rocket launch data from SpaceX API with the following URL

- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



Data Collection - Scraping

- Web scrap Falcon 9 launch records with BeautifulSoup:
- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame
- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



Data Wrangling

- Exploratory Data Analysis: This involved an initial exploration of the dataset to understand its structure and characteristics.
- Determine Training Labels: The training labels for the predictive analysis were identified, which likely include information related to mission outcomes and landing outcomes.
- Missing Value Analysis: The percentage of missing values in each attribute was calculated, aiding in understanding data completeness and deciding how to handle missing data.
- Data Type Identification: Columns were categorized as either numerical or categorical, which is crucial for data modeling and analysis.
- Site-Based Analysis: The number of launches from each launch site was calculated, providing insights into launch distribution.
- Orbit Analysis: The number and occurrence of each orbit were determined, allowing for an understanding of the distribution of missions across different orbits.
- Mission Outcome Analysis: The number and occurrence of mission outcomes for the orbits were also calculated to assess mission success rates.
- Landing Outcome Label Creation: A new label, likely related to landing outcomes, was created from the "Outcome" column, which is crucial for predictive modeling.

- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display the average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome on a ground pad was achieved.
- List the names of the boosters with successful drone ship landings and a payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster versions that have carried the maximum payload mass.
- Display records showing the month names, failure landing outcomes on a drone ship, booster versions, and launch sites for the months in the year 2015.
- 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.
- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Markers:** Markers were added to represent each launch site on the map (Task 1). These markers help visually identify the exact locations of the launch sites, making it easy for users to spot them.
- **Circle Markers:** Circle markers were used to distinguish between successful and failed launches for each launch site (Task 2). Green circle markers were employed to indicate successful launches, while red circle markers represented failed launches. This visual differentiation allows for quick assessment of success rates at each site.
- **Lines:** Lines were drawn to connect the launch site with its proximities, demonstrating the calculated distances (Task 3). These lines visually illustrate the geographical relationship between the launch site and its nearby landmarks, providing valuable information on proximity.
- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- **1. Dropdown Menu for Launch Site Selection (Interaction):**
- A dropdown menu allows users to select specific launch sites, providing the flexibility to analyze launches for individual sites or all sites combined.
- **2. Pie Chart (Plot):**
- A pie chart displays the total count of successful launches, which is updated based on the site selection made in the dropdown menu.
- **Reason:** This chart offers an overview of launch success rates for all or specific sites, enabling quick comparison.
- **3. Payload Range Slider (Interaction):**
- A range slider permits users to select a payload range (in kilograms), defining the payload mass range of interest for analysis.
- **4. Scatter Plot (Plot):**
- A scatter plot presents the correlation between payload mass and launch success.
- The color is coded by Booster Version Category, helping identify any patterns based on booster versions.
- **Reason:** This plot provides insights into how payload mass influences launch success and allows for comparisons between different booster versions.
- **5. Callback Functions (Logic):**
- Callback functions update the pie chart and scatter plot based on user interactions, ensuring that the displayed data is relevant and dynamic.
- **Reason:** Callback functions create responsive and interactive dashboards, enhancing the user experience
- https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/SpaceXPlotly.py

Predictive Analysis (Classification)

1. Exploratory Data Analysis and Determination of Training Labels:

Conducted exploratory data analysis to understand the dataset's characteristics and distributions.

Identified suitable training labels for classification, which likely include success/failure or similar outcomes.

2. Creation of a 'Class' Column:

Created a 'Class' column in the dataset to represent the target variable for classification, typically denoting success (1) or failure (0) based on the determined training labels.

3. Data Standardization:

Standardized the data to ensure that all features have the same scale, which can improve the performance of certain machine learning algorithms.

4. Splitting Data into Training and Test Sets:

Split the dataset into training data and test data to facilitate model training and evaluation.

The training data is used to train the classification models, while the test data is reserved for model evaluation.

Predictive Analysis (Classification)

5. Hyperparameter Tuning:

Employed hyperparameter tuning for different classification algorithms, specifically Support Vector Machines (SVM), Classification Trees, and Logistic Regression.

Utilized techniques such as grid search or random search to find the best hyperparameters for each model.

6. Model Building and Evaluation:

Constructed classification models with varying hyperparameters for each of the selected algorithms.

Evaluated model performance using relevant evaluation metrics, such as accuracy, precision, recall, and F1-score.

Models were repeatedly assessed and adjusted to improve their performance.

7. Model Comparison:

Compared the performance of SVM, Classification Trees, and Logistic Regression using the test data.

The model with the highest performance on the test data, based on the chosen evaluation metric, was considered the best-performing classification model.

Predictive Analysis (Classification)

https://github.com/Karam-Issa/DataScience_CapstonePoject/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Find the method performs best:

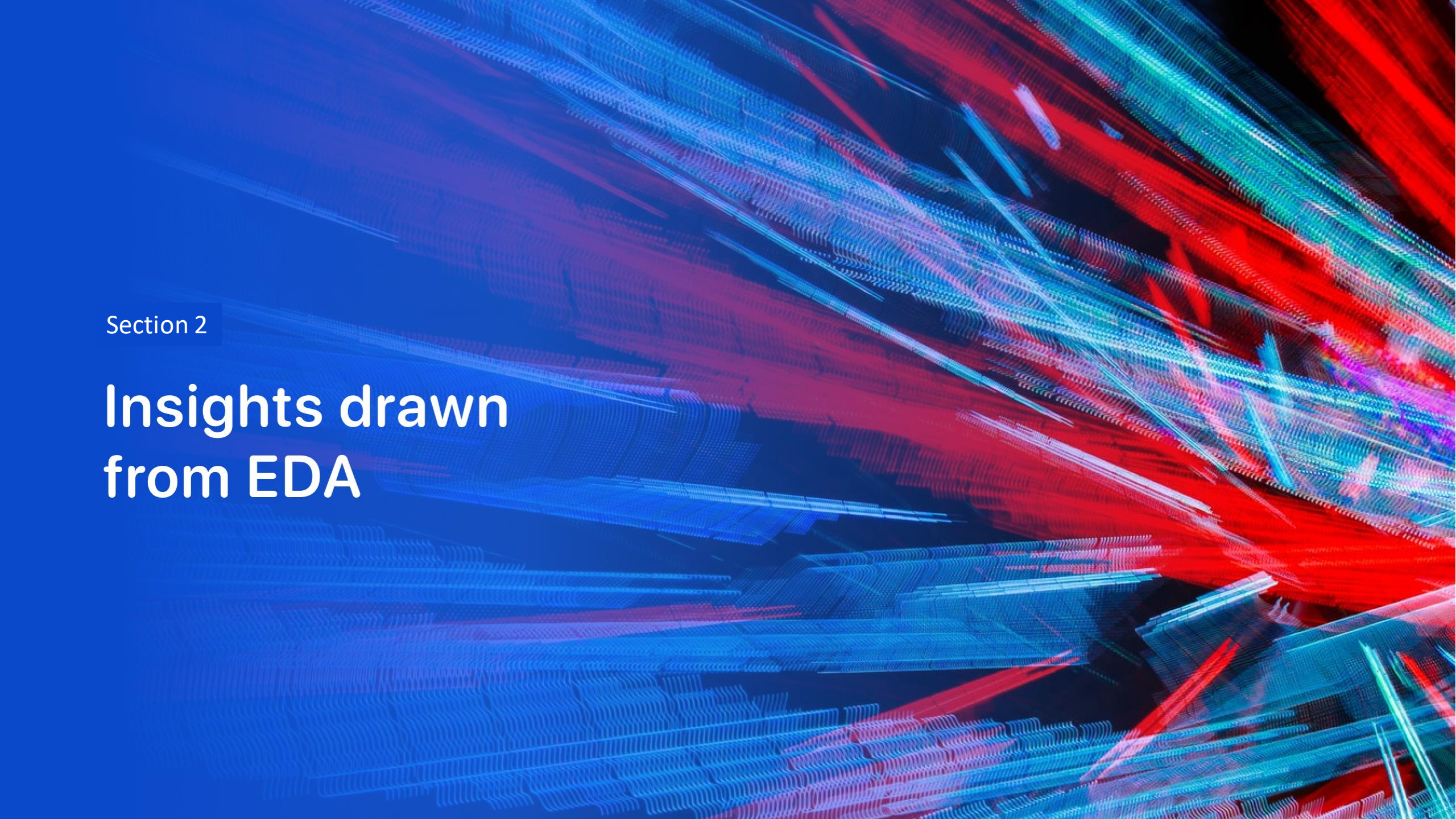
In [35]:

```
# Calculate the accuracy of each model on the test data
logreg_accuracy = logreg_cv.score(X_test, Y_test)
svm_accuracy = svm_cv.score(X_test, Y_test)
tree_accuracy = tree_cv.score(X_test, Y_test)
knn_accuracy = knn_cv.score(X_test, Y_test)

# Create a dictionary to store the accuracies
accuracies = {
    "Logistic Regression": logreg_accuracy,
    "SVM": svm_accuracy,
    "Decision Tree": tree_accuracy,
    "K-Nearest Neighbors": knn_accuracy
}

# Find the method with the highest accuracy
best_method = max(accuracies, key=accuracies.get)
best_accuracy = accuracies[best_method]

print(f"The best performing method is {best_method} with an accuracy of {best_accuracy:.2f}")
```

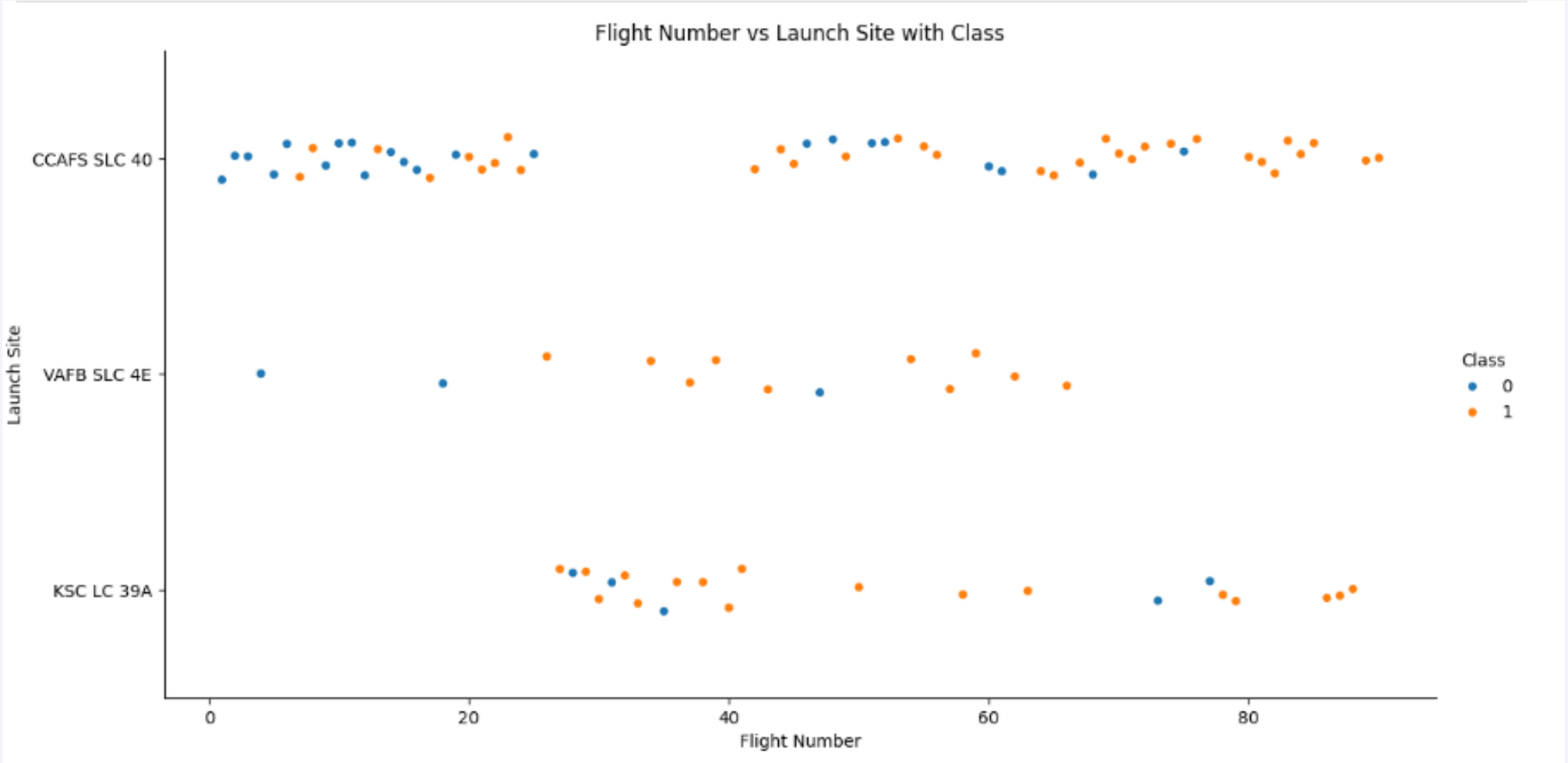
```
The best performing method is Logistic Regression with an accuracy of 0.83
```

Section 2

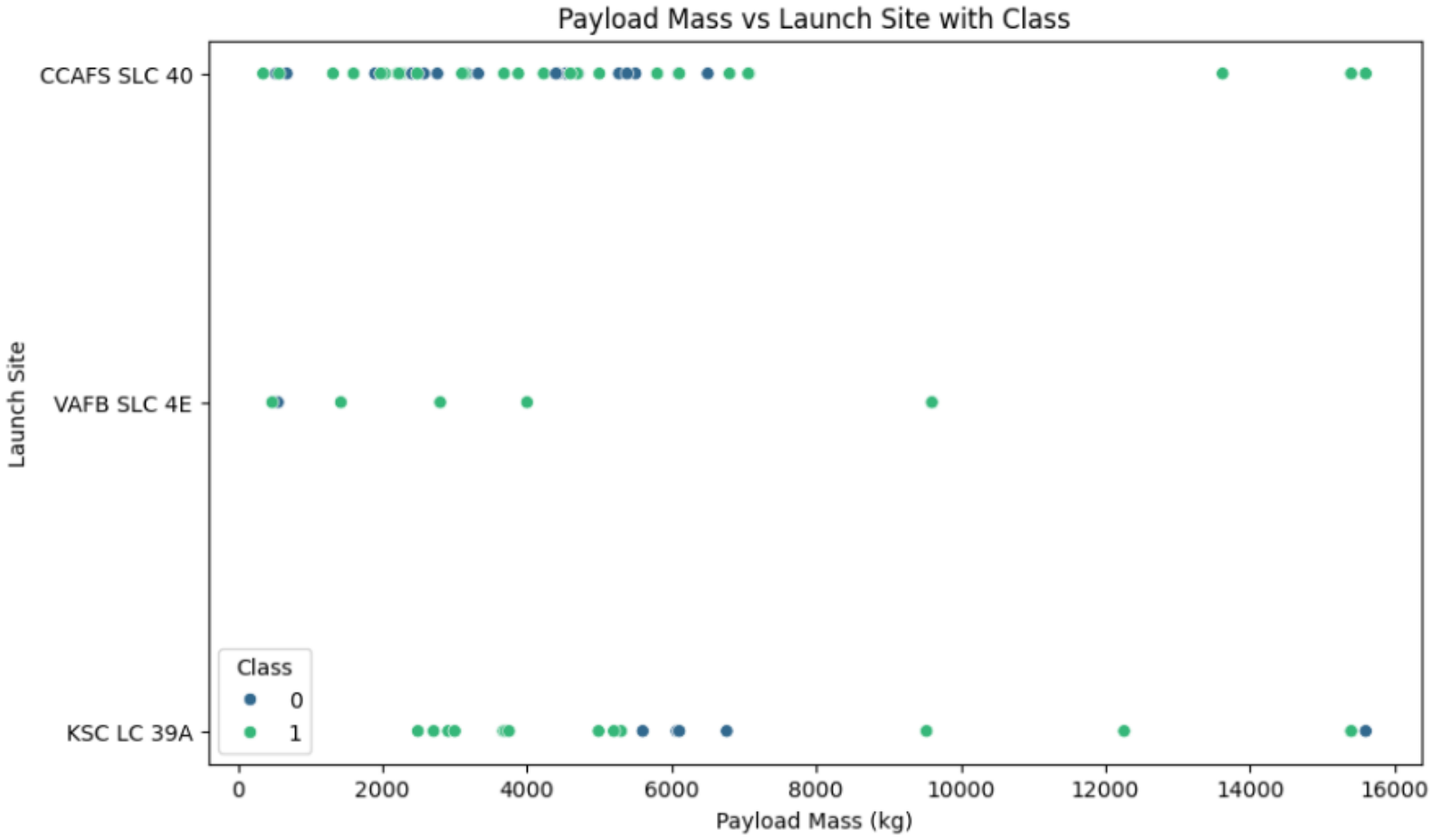
Insights drawn from EDA



Flight Number vs. Launch Site

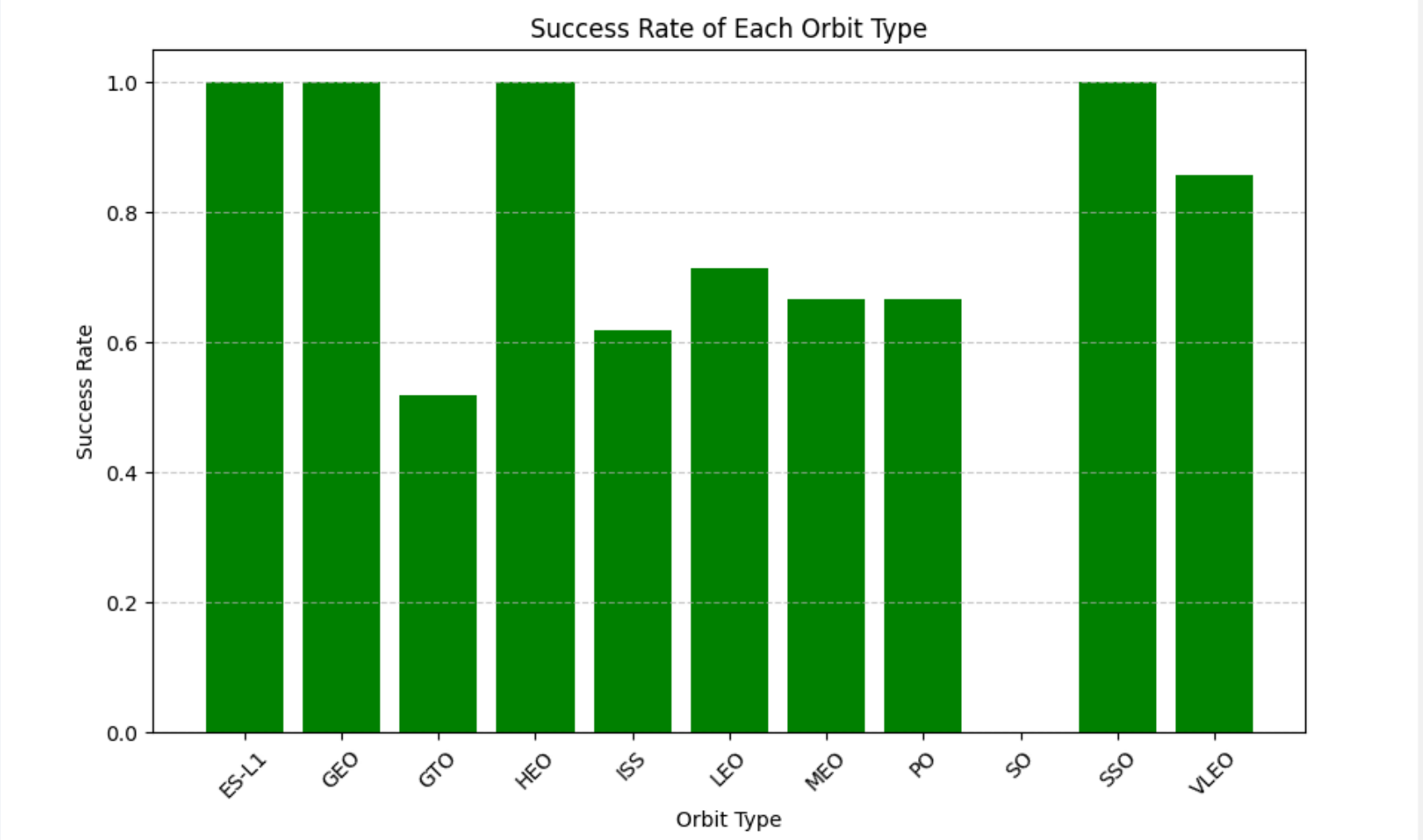


Payload vs. Launch Site

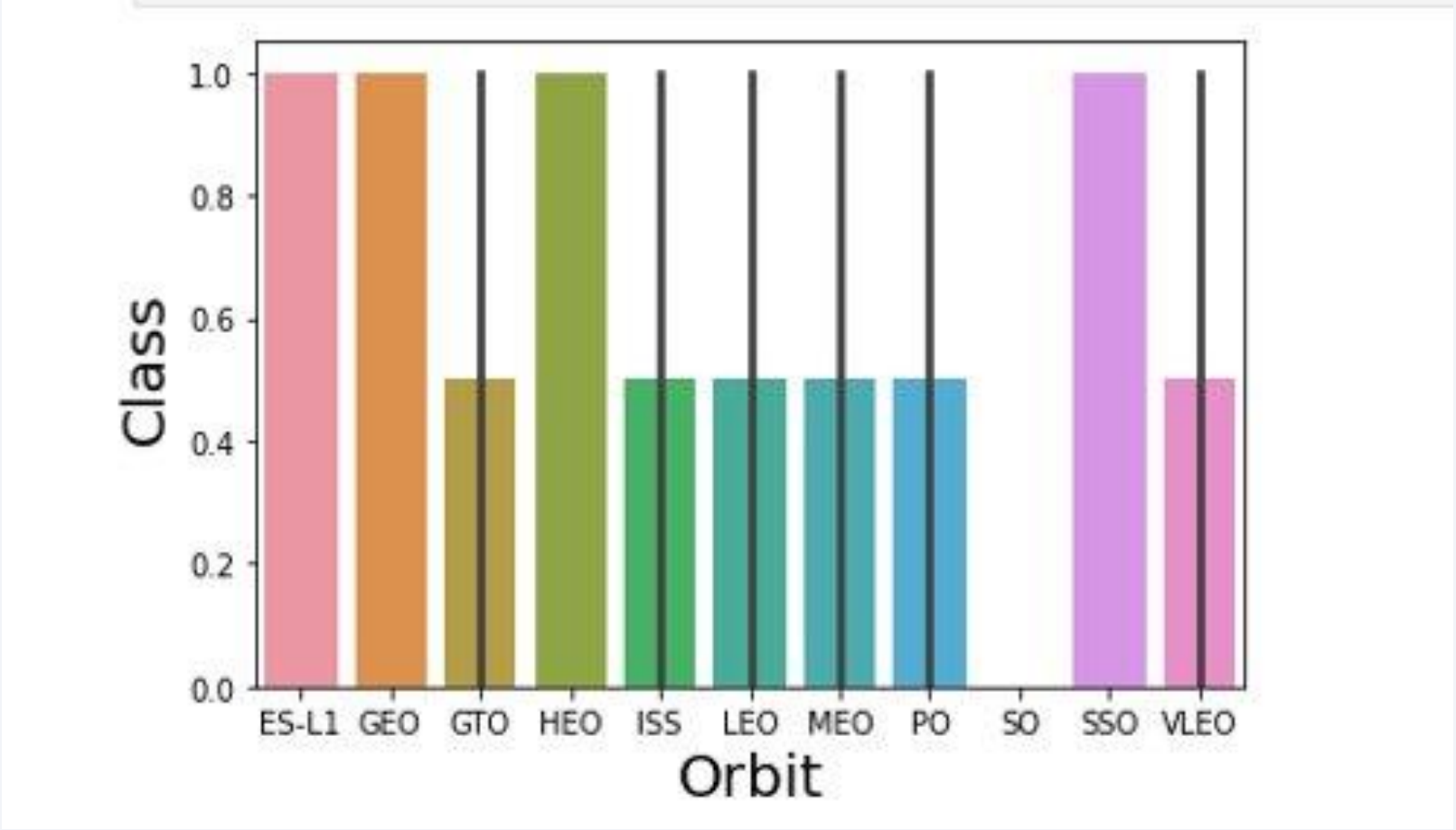


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

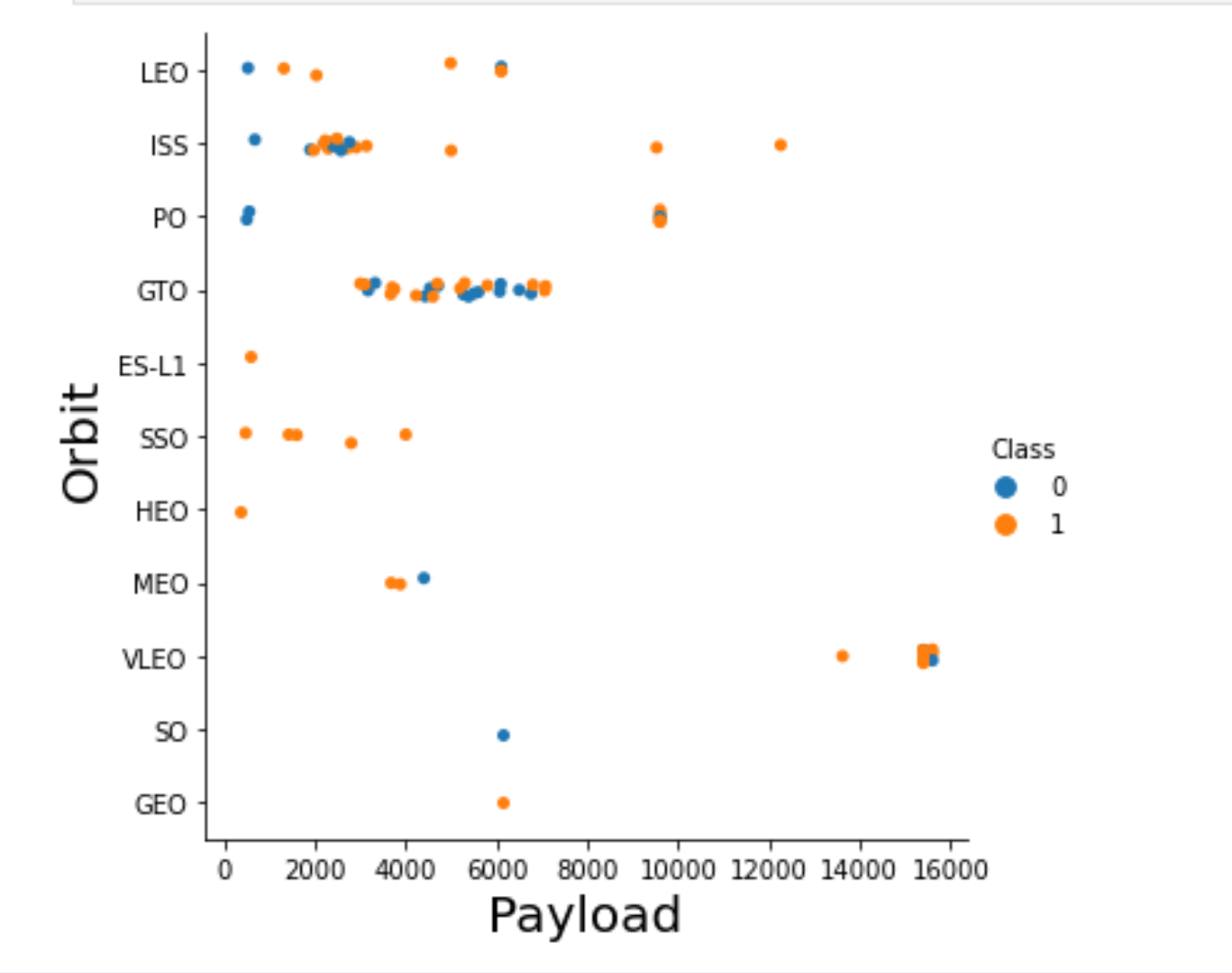
Success Rate vs. Orbit Type



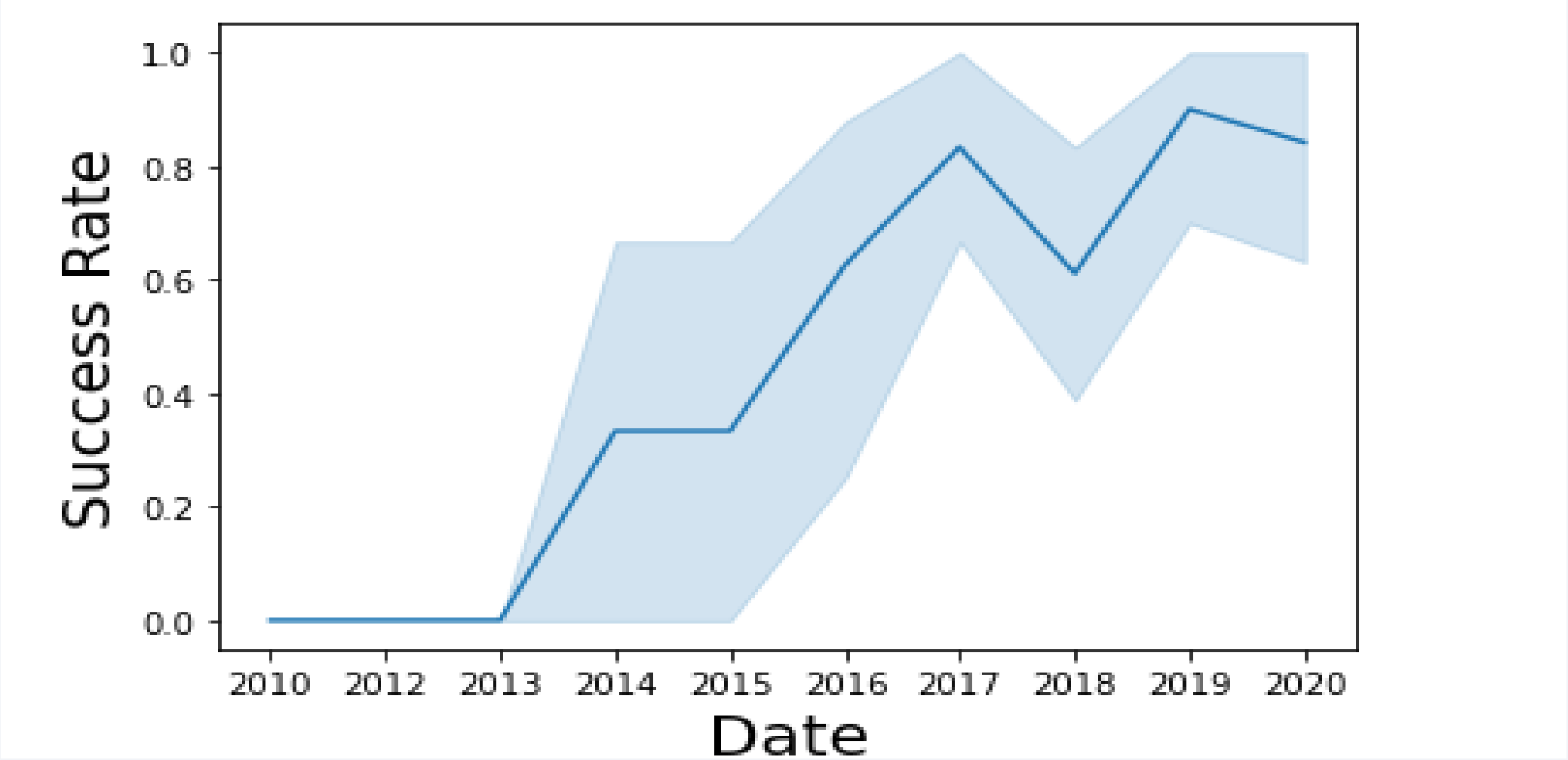
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

In [11]:
```
%sql select Distinct(launch_site) from SpaceXTable
```

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

In [13]:
```
%sql select * from spacextable where launch_site like 'CCA%' limit 5
```

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

Out[13]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landin
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight	525	LEO (ISS)	NASA (COTS)	Success	

Total Payload Mass

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

In [18]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SpaceXtable WHERE Customer = 'NASA (CRS)';
```

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

Out[18]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

Task 4

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

In [20]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM spacextable WHERE Booster_Version = 'F9 v1.1';
```

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

Out[20]:

Average_Payload_Mass
2928.4

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 [21]:
```
%sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM spacextable WHERE Mission_Outcome = 'Success' AND Landing_Outcor
```

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

Out[21]:

First_Successful_Landing_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 [24]:
```
%sql SELECT DISTINCT Booster_Version FROM SpaceXtable WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ :
```

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

Out[24]:

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 [25]:
```
%sql SELECT Mission_Outcome, COUNT(*) AS Total FROM spacextable WHERE Mission_Outcome IN ('Success', 'Failure') GROUP BY Mi
```

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

Out[25]:

Mission_Outcome	Total
Success	98

Boosters Carried Maximum Payload

Task 8

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

In [26]:
```
%sql SELECT Booster_Version FROM spacextable WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacextable);
```

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

Out[26]:

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

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.

```
%sql SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM spacextable WHERE substr(Date, (
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
10	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

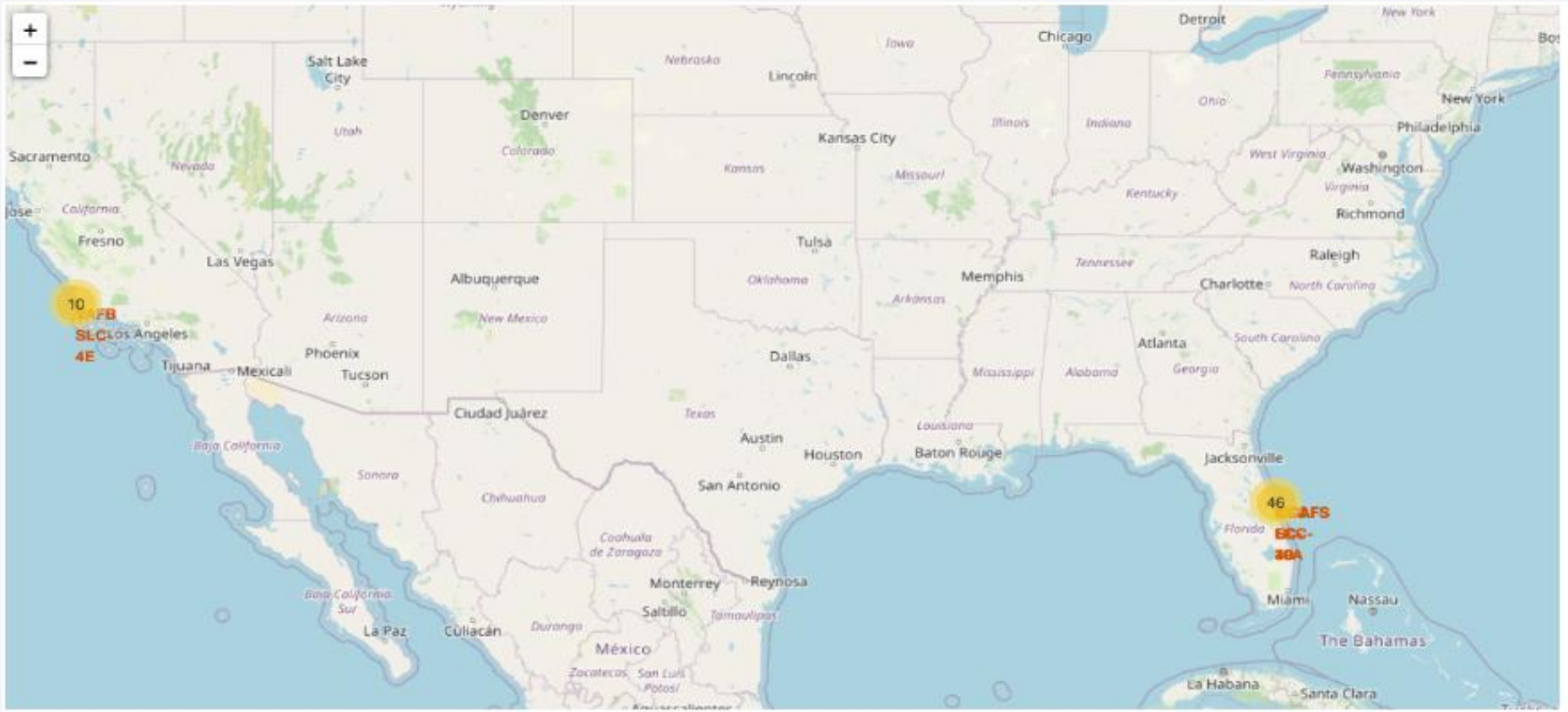
Task 10

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.

```
]:
SELECT Landing_Outcome, COUNT(*) AS Count
FROM spacextable
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
AND (Landing_Outcome LIKE 'Failure% (drone ship)' OR Landing_Outcome LIKE 'Success% (ground pad)') GROUP BY Landing_Outcome
```

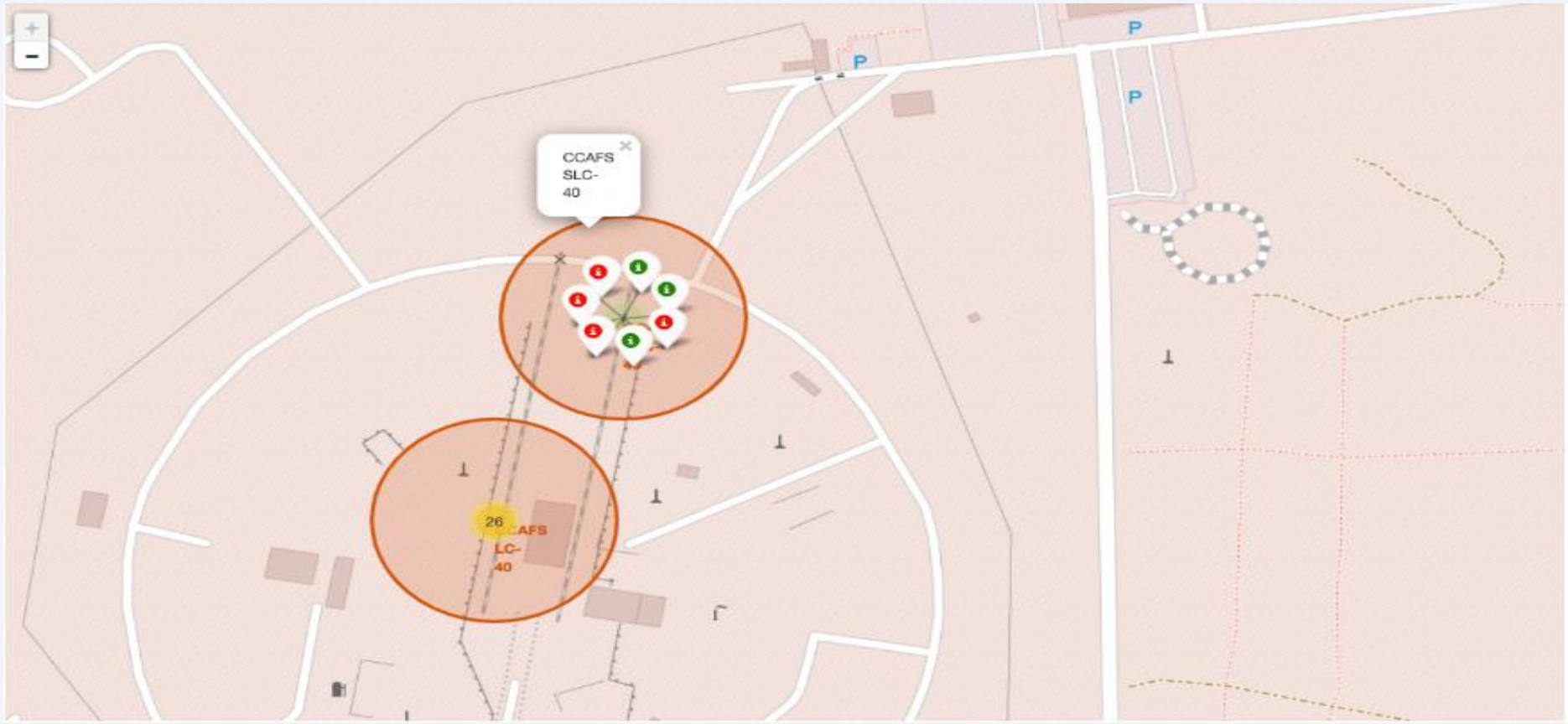

Section 3
Launch Sites
Proximities Analysis

Launch Sites



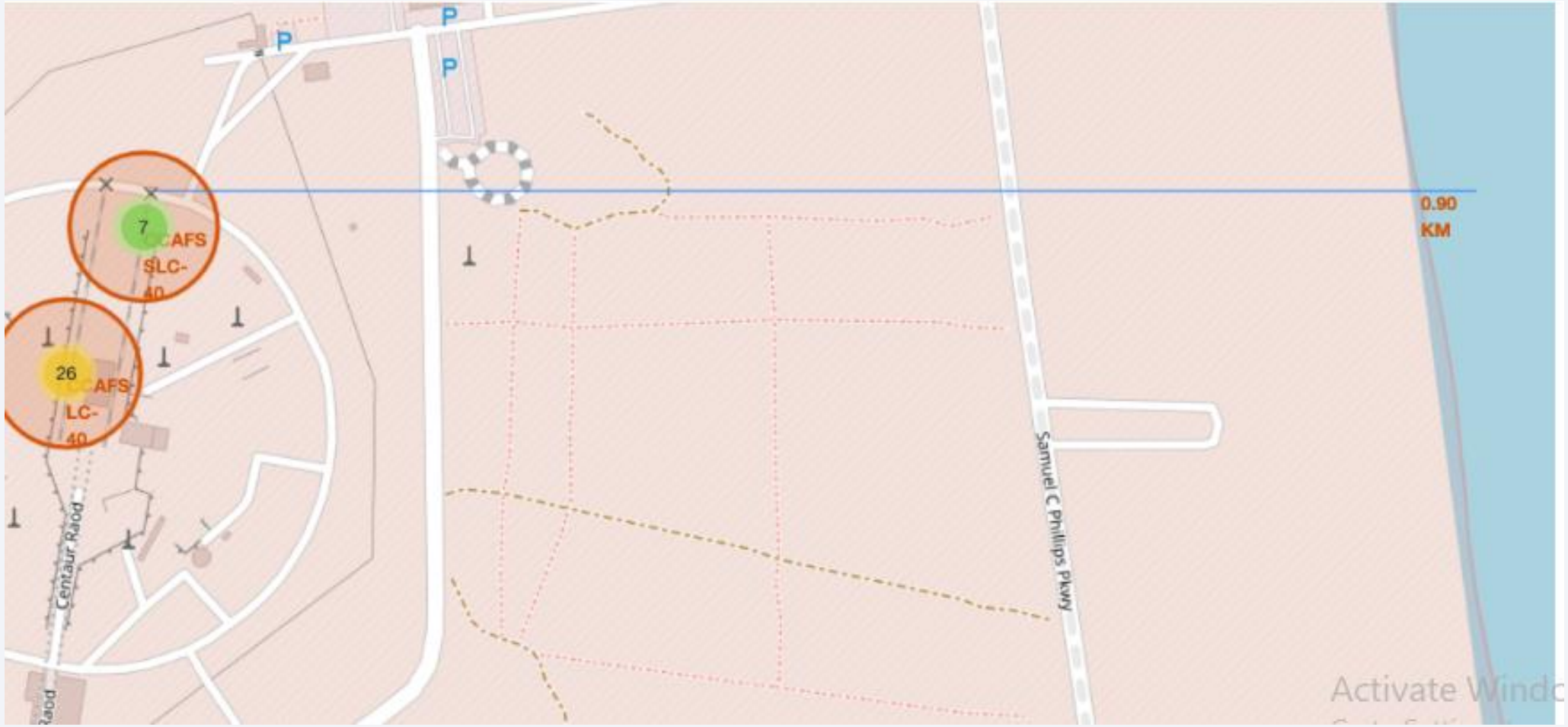
- All sites are close to the coast

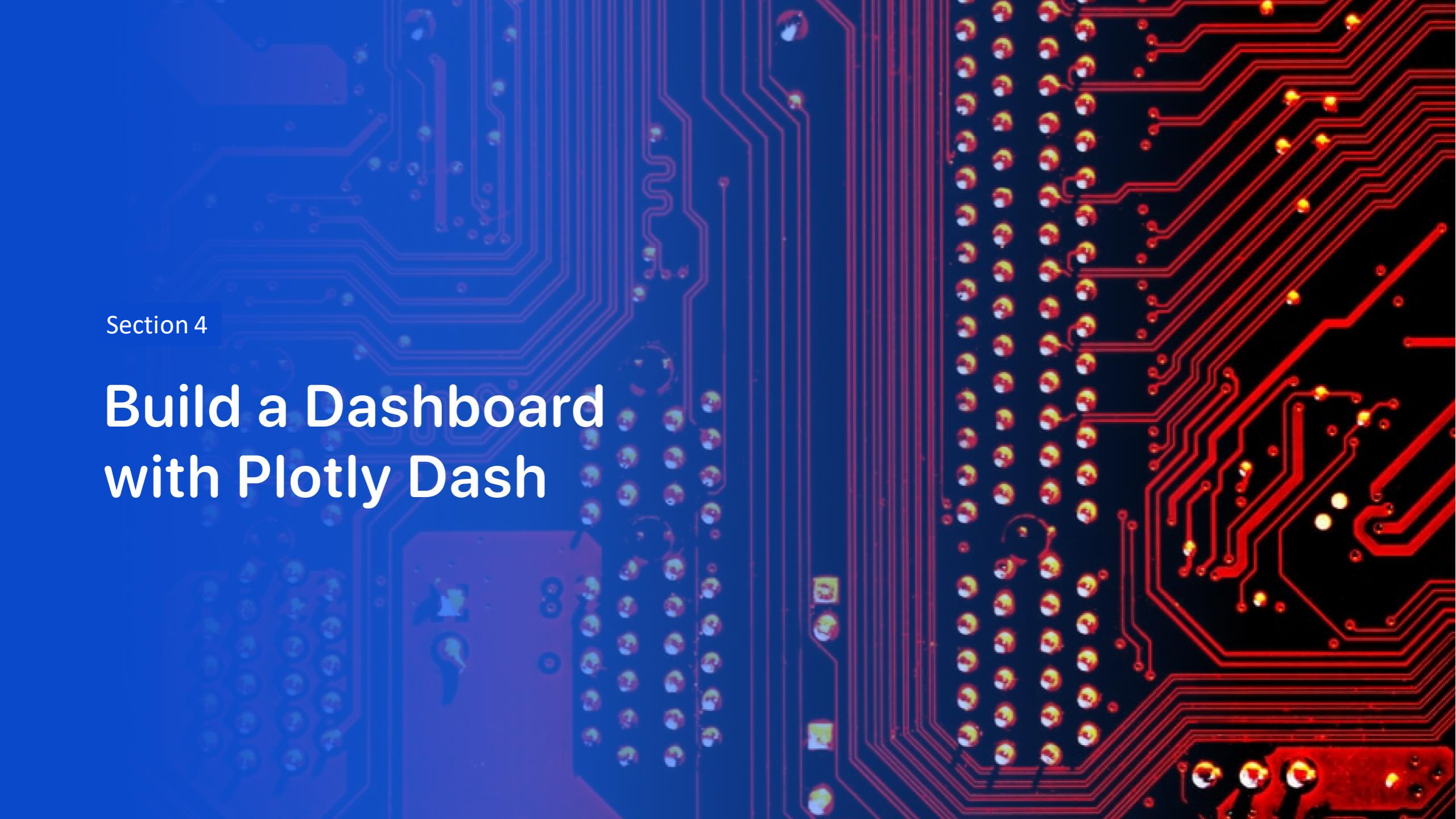
MarkerCluster



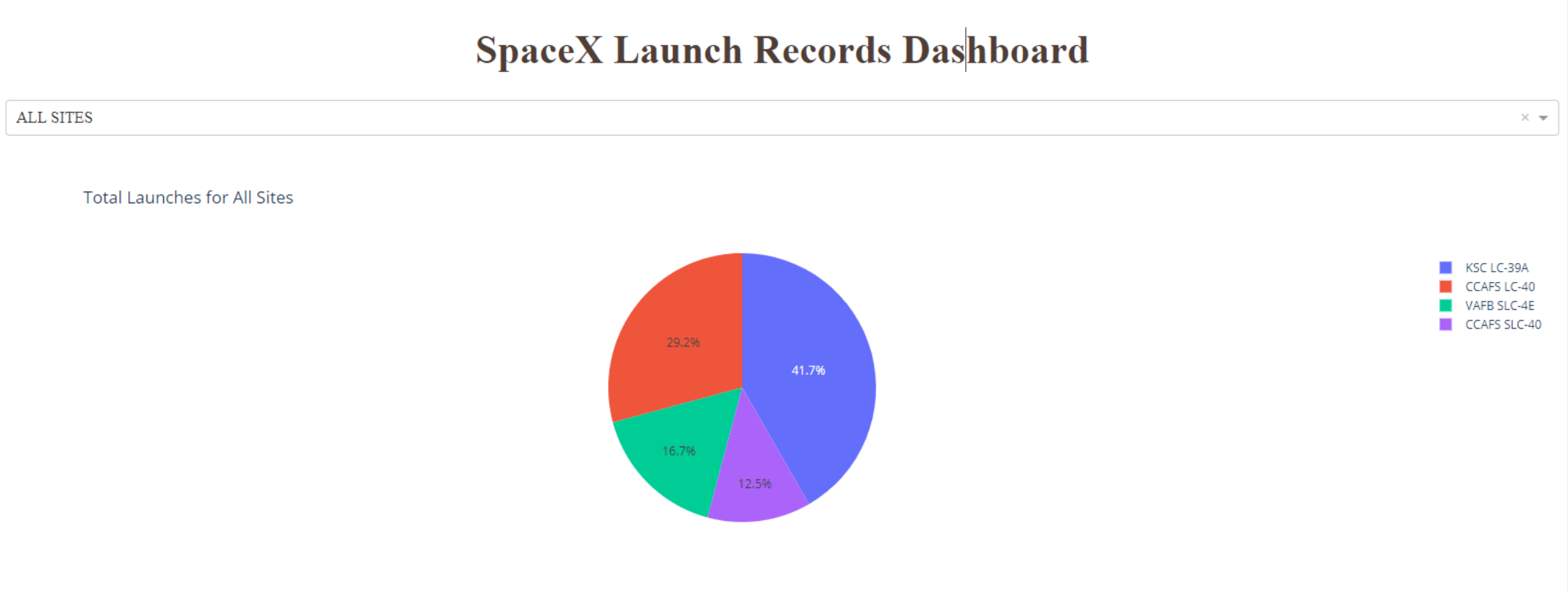
Markers indicating if successful launch(Green) and failed launch (RED)

PolyLine between a launch site to the selected coastline point



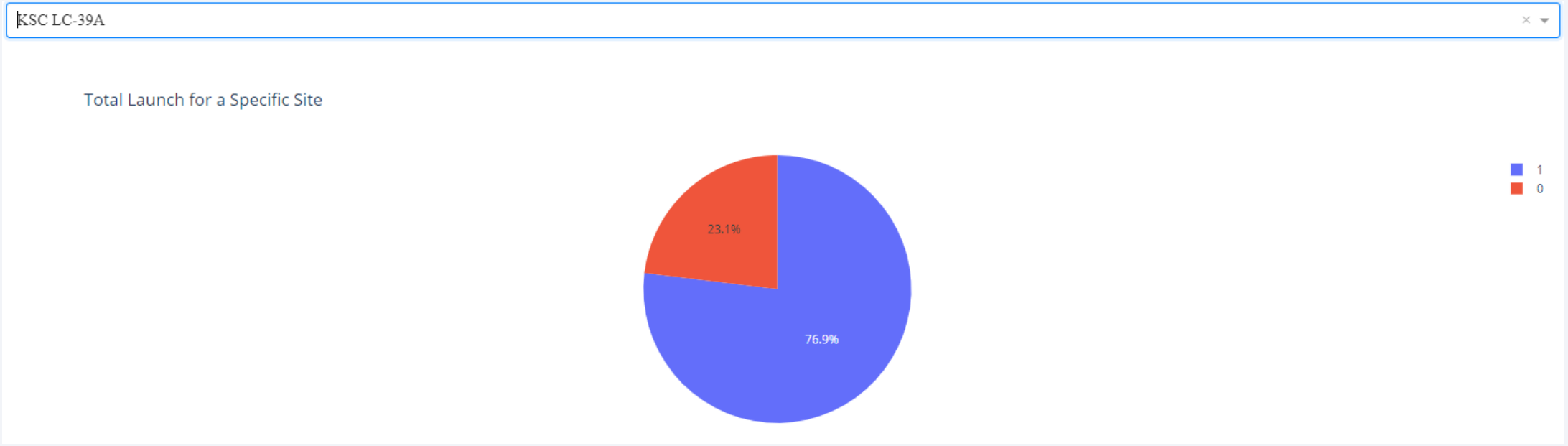

Section 4
Build a Dashboard with Plotly Dash

Launch Site Distribution

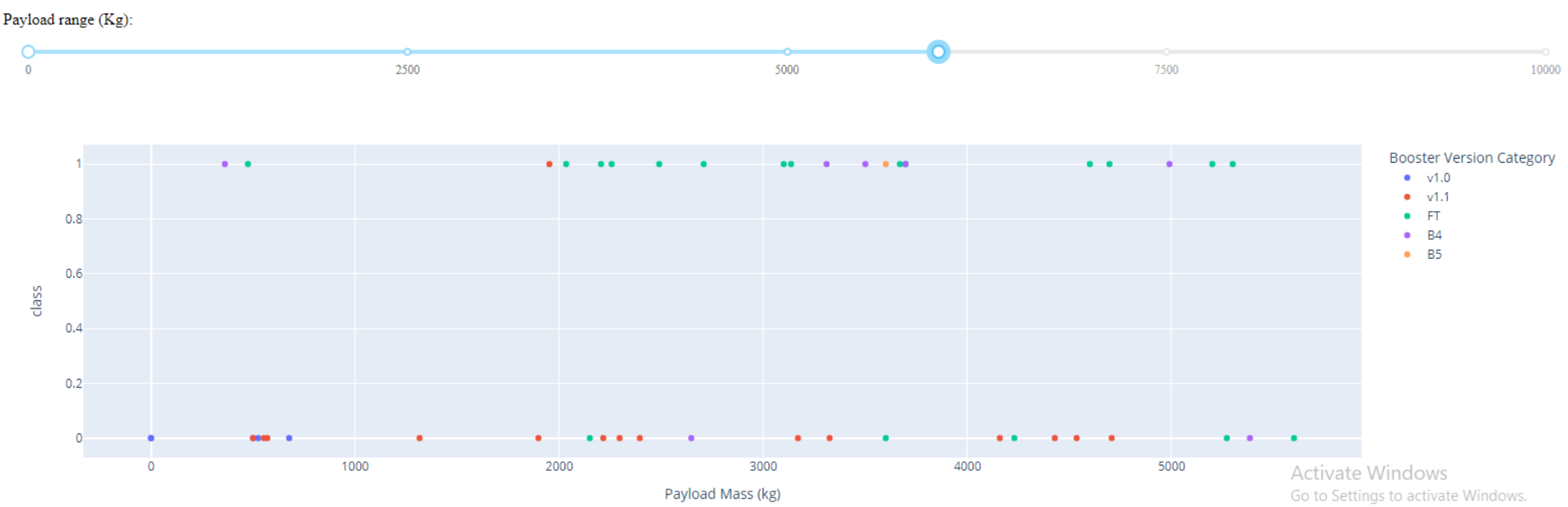


KSC LC-39A has the highest percentage of launches at 41.7%.

Highest launch Success



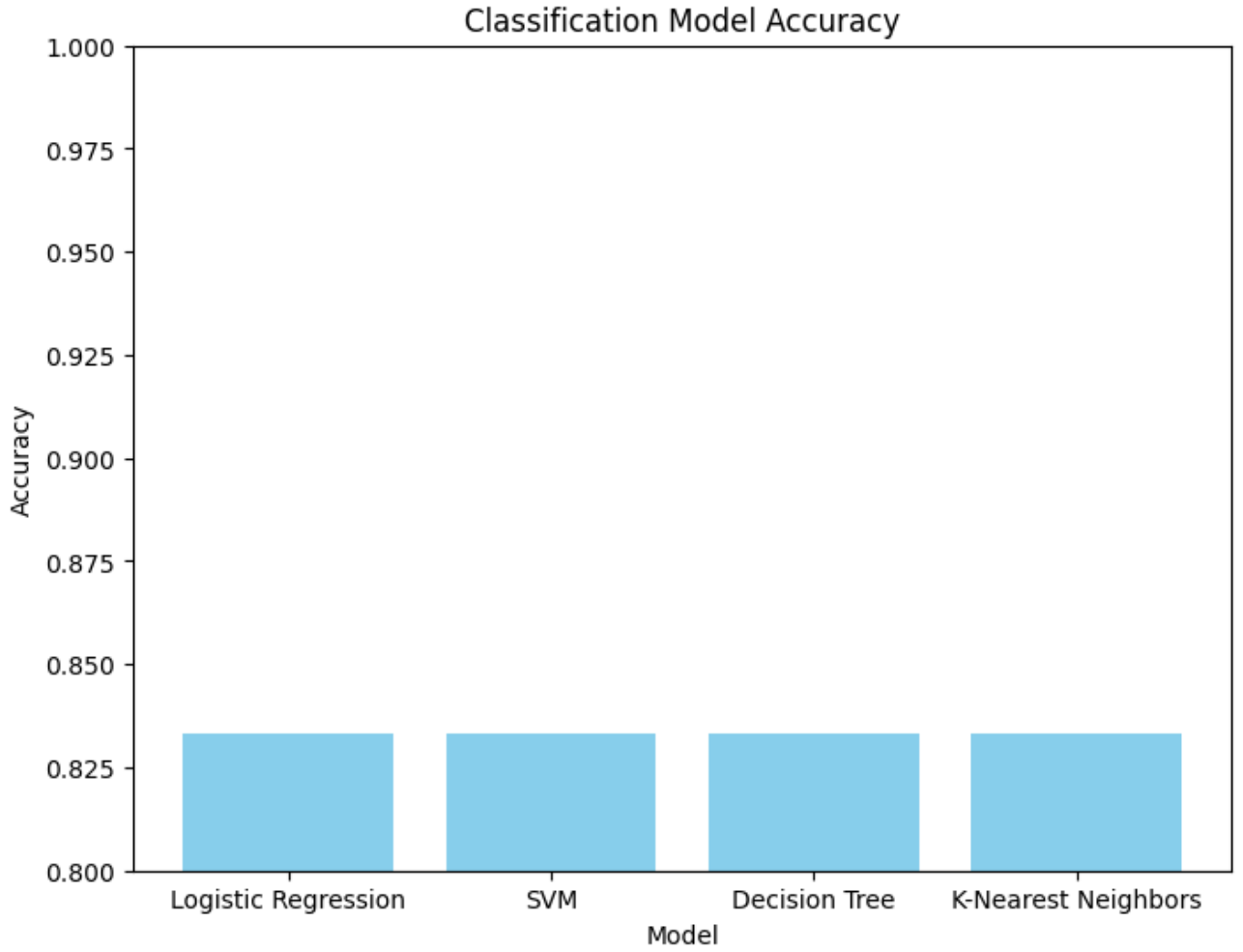
Payload vs. Launch Outcome scatter plot



Section 5

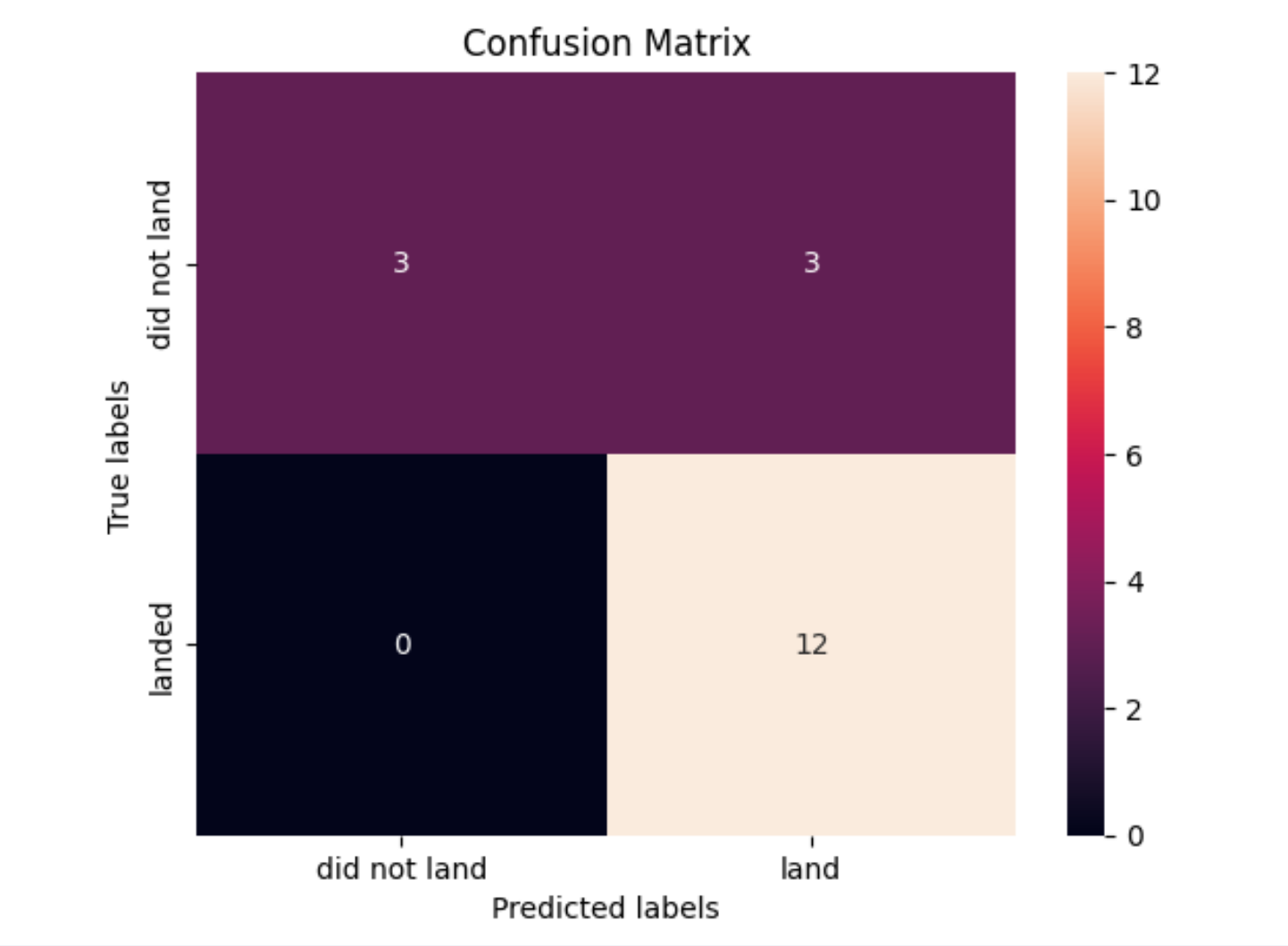
Predictive Analysis (Classification)

Classification Accuracy



The best performing model is Logistic Regression with an accuracy of 0.83

Confusion Matrix



Confusion Matrix

- The model correctly predicted "did not land" in 3 instances.
- The model correctly predicted "landed" in 12 instances.
- The model made 3 incorrect predictions, where it mistakenly predicted "did not land" when it was actually "landed" (false negatives).
- The model did not make any incorrect predictions where it predicted "landed" when it should have been "did not land" (false positives).

Conclusions

- Data collection and preprocessing are crucial for accurate analysis.
- Predictive analysis helps estimate launch costs in the rocket industry.
- Data wrangling involves cleaning and structuring data.
- We created an interactive dashboard to visualize SpaceX data.
- We discussed building, evaluating, and selecting classification models.
- Confusion matrices assess model performance.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!