IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Waldemar Herter
27.09.2024

Outline

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

Executive Summary

- Summary of methodologies
  - Used APIs and web scraping to gather data, cleaned it, and explored it with visualizations. Built machine learning models for predictions.
- Summary of all results
  - The SVM model with the RBF kernel achieved the best performance

Introduction

- Project background and context
  - Focused on analyzing SpaceX launch data to predict future outcomes.

- Problems you want to find answers
  - What factors lead to successful launches?
  - How do different launch sites compare?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - Collected Data via API calls and web scraping
- Perform data wrangling
  - Cleaned and transformed data for analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Created scatter plots and bar graphs to visualize data
  - Analyzed launch success rates.

Methodology

Executive Summary

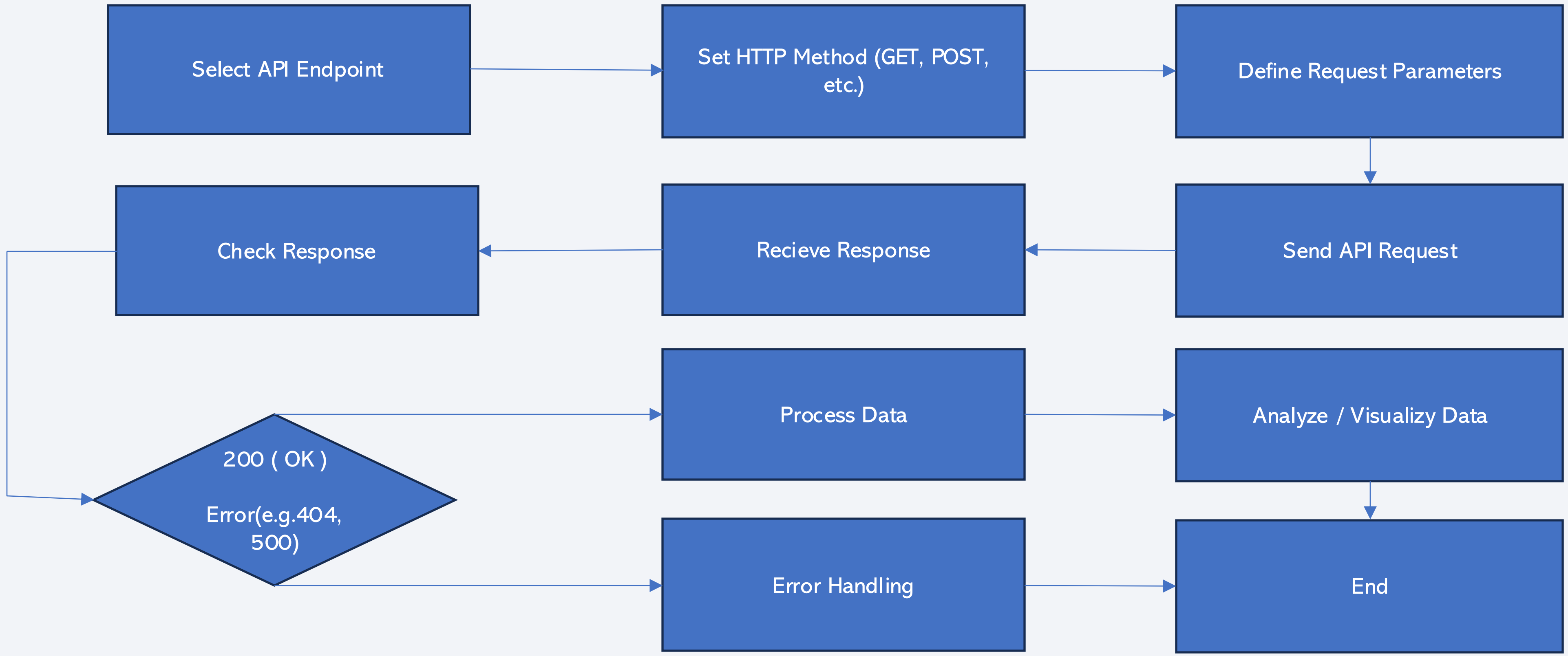
- Build an Interactive Map with Folium
  - Map Objects Created: Added markers for launch sites.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Created a dashboard with interactive visuals
- Perform predictive analysis using classification models
  - Developed and evaluated classification models

Data Collection

Methodology Overview

- **Data Collection**:
  - Collected via API calls & web scraping.
  - SpaceX API for launch data
  - Additional data via web scraping
- **Data Wrangling**:
  - Cleaned and processed for analysis
- **Exploratory Data Analysis (EDA)**:
  - Visualized data with scatter plots & bar graphs
  - SQL analysis on launch success rates
- **Interactive Tools**:
  - Built interactive maps using Folium
  - Created a dashboard with Plotly Dash
- **Predictive Analysis**:
  - Developed & evaluated classification models

Data Collection – SpaceX API



- [GitHub link to Jupyter Notbook]

Data Collection - Scraping

Request Web Page

- HTTP GET Request using `requests.get()`
- - URL: `https://en.wikipedia.org/...`
- - Response status check (`response.status_code`)

Parse HTML with BeautifulSoup

- Create a BeautifulSoup object (`soup = BeautifulSoup(response.text, 'html.parser')`)
- Find all tables using `soup.find_all('table')`

Extract Column Names

- Loop through `<th>` tags in the launch table
- Use `extract_column_from_header()` to clean and extract header names
- Store column names in `column_names` list

Parse Table Data

- Initialize `launch_dict` with column names as keys
- Loop through each row `<tr>` in the table
- Extract flight data, booster version, launch site, payload, etc.
- Store parsed values in corresponding `launch_dict` keys

Create Pandas DataFrame

- Convert `launch_dict` to DataFrame using `pd.DataFrame()`

- [GitHub link to Jupyter Notebook]

Data Wrangling

Data Collection	• Loaded SpaceX Falcon 9 dataset (CSV File) using pandas
Missing Values	• Identifiing missing values, notably in the KIndingPad column (approx 28%)
Data Types & EDA	• Categorical : LaunchSite, Orbit, Outcome, etc. • Numerical: Flightnumber, PayloadMass, etc. • Analyzed: LaunchSite, Orbit and Outcome distributions
Label Creation (Landing Success)	• Converted Outcome into binary Class (0:fail, 1:success)
Export	• Saved processed dataset with new labels to CSV

- [GitHub link to Jupyter Notebook]()

EDA with Data Visualization

Data Collection

- Loaded SpaceX Falcon 9 dataset (CSV File) using pandas

Missing Values

- Identifiing missing values, notably in the KlndingPad column (approx 28%)

Data Types & EDA

- Categorical : LaunchSite, Orbit, Outcome, etc.
- Numerical: Flightnumber, PayloadMass, etc.
- Analyzed: LaunchSite, Orbit and Outcome distributions

Label Creation (Landing Success)

- Converted Outcome into binary Class (0:fail, 1:success)

Export

- Saved processed dataset with new labels to CSV

- GitHub link to Jupyter Notebook

EDA with SQL

Flight Number vs. Payload Mass
- **Chart Type:** Catplot
- **Purpose:** To visualize the relationship between flight attempts and payload mass, and their impact on landing success.

Flight Number vs. Launch Site
- **Chart Type:** Catplot
- **Purpose:** To analyze how the number of flights correlates with different launch sites.

Payload Mass vs. Launch Site
- **Chart Type:** Scatterplot
- **Purpose:** To observe the relationship between payload mass and launch sites, highlighting patterns for heavy payloads.

Success Rate by Orbit Type
- **Chart Type:** Barplot
- **Purpose:** To determine which orbit types have the highest success rates.

Flight Number vs. Orbit Type
- **Chart Type:** Scatterplot
- **Purpose:** To explore how flight numbers relate to different orbit types and success.

Payload Mass vs. Orbit Type
- **Chart Type:** Scatterplot
- **Purpose:** To assess how payload mass affects success rates across various orbits.

Yearly Success Rate Trend
- **Chart Type:** Lineplot
- **Purpose:** To track the success rate of launches over time.

- GitHub link to Jupyter Notebook

Build an Interactive Map with Folium

Markers	Indicates the exact locations of SpaceX launch sites, helping identify where launches take place geographically
Circle Markers	Represents launch outcomes (green = successful, red = failed) to visualize the performance of each site
Popups	Provide additional informations about each site, such as name and launch statistics, for better context
Lines	Show distance between launch sites and key infrastructure (e.g. coastline, highway), illustrating proximity for logistics and safety analysis

- GitHub link to Jupyter Notebook

Build a Dashboard with Plotly Dash

Plots/Graphs Added:

- **Pie Chart (Success Rate)**:
  - Shows the total number of successful launches by launch site. When a specific site is selected, it displays a breakdown of successful vs. failed launches for that site.
- **Scatter Plot (Payload vs. Success)**:
  - Displays the relationship between payload mass and launch success. It helps identify any trends or correlations between the payload size and the success rate.

Interactions Added:

- **Launch Site Dropdown**:
  - Allows users to select a specific launch site or view data for all sites. This interaction helps in filtering the data and focusing on the performance of particular launch sites.
  - **Payload Range Slider**:
    Users can filter launches based on payload weight. This interaction helps examine how different payload sizes impact launch outcomes.

- GitHub link to Jupyter Notebook

Predictive Analysis (Classification)

Two CSV datasets containing flight details

- FlightNumber
- PayloadMass
- Landing Outcome (Class)

Data Preparation

- Loaded data and created target variable Y from the Class column
- Standardized features in X
- Split data into training (80%) and testing (20%) sets

Logistic Regression

- Hyperparameters tuned via GridSearchCV (Best C: 0.01)
- Validation accuracy: 84.6%
- Test accuracy: 83.3%
- Confusion matrix shows 12 True Positives, 3 False Positives

Support Vector Machine (SVM)

- Best kernel: sigmoid, C: 1.0
- Validation accuracy: 84.8%
- Test accuracy: 83.3%

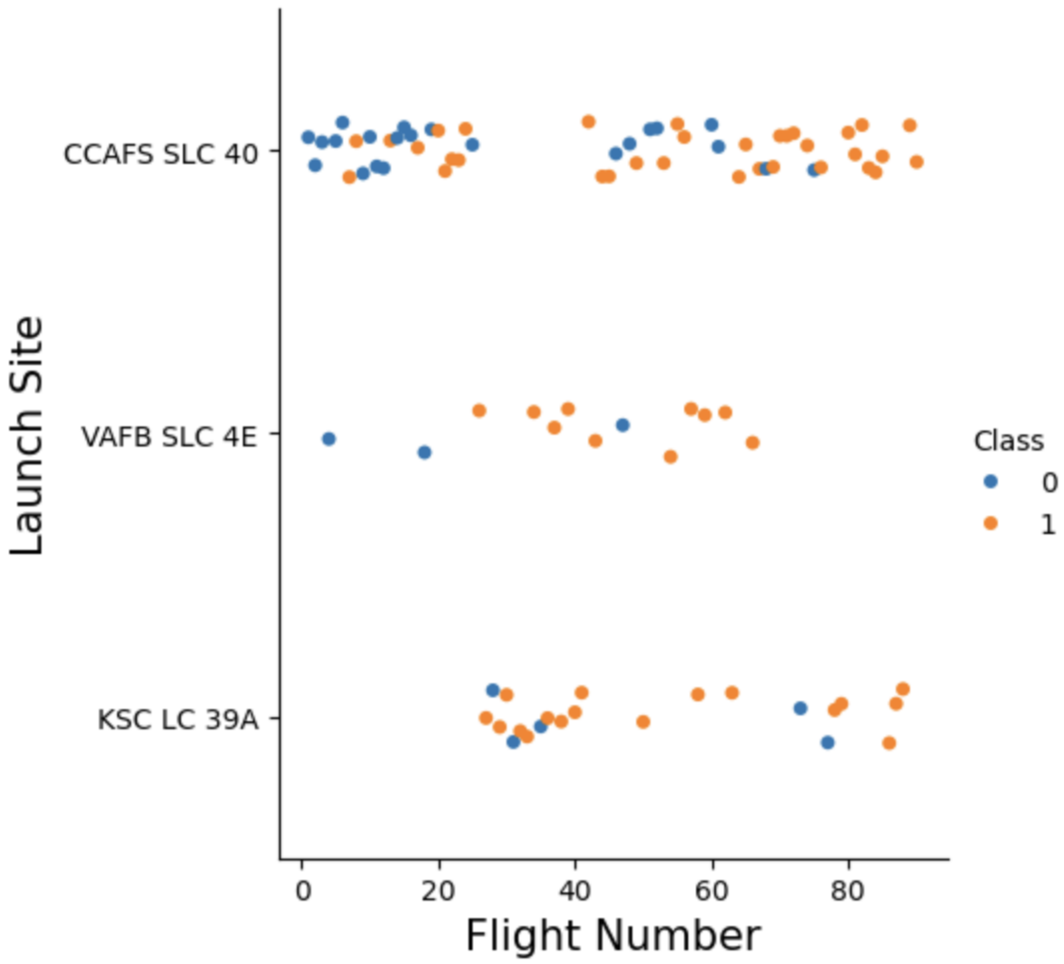
Results

- Exploratory data analysis results
  - Insights from data exploration
- Interactive analytics demo in screenshots
  - Demonstrations of dashboard features
- Predictive analysis results
  - Summary of model performance

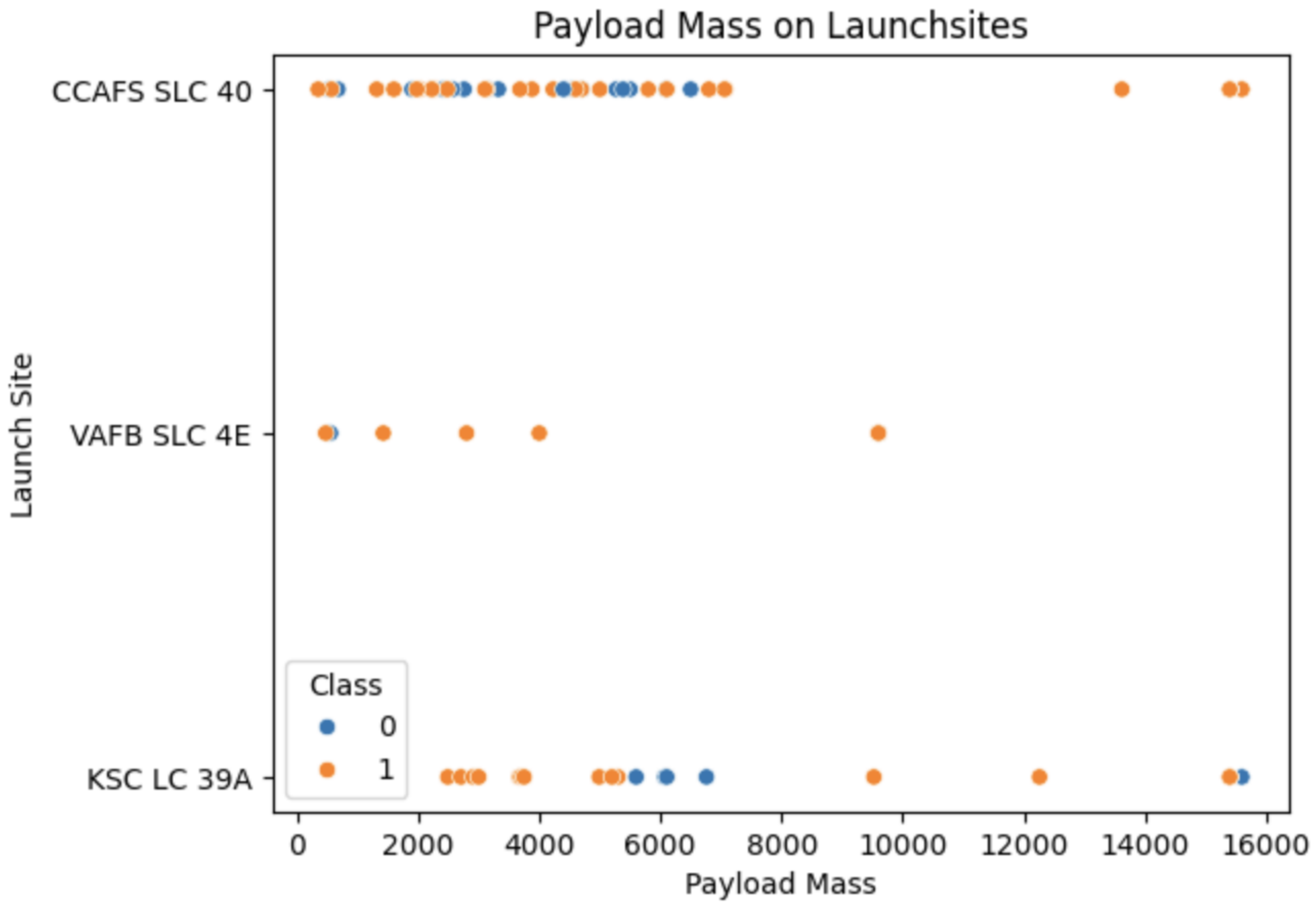
Section 2

Insights drawn from EDA

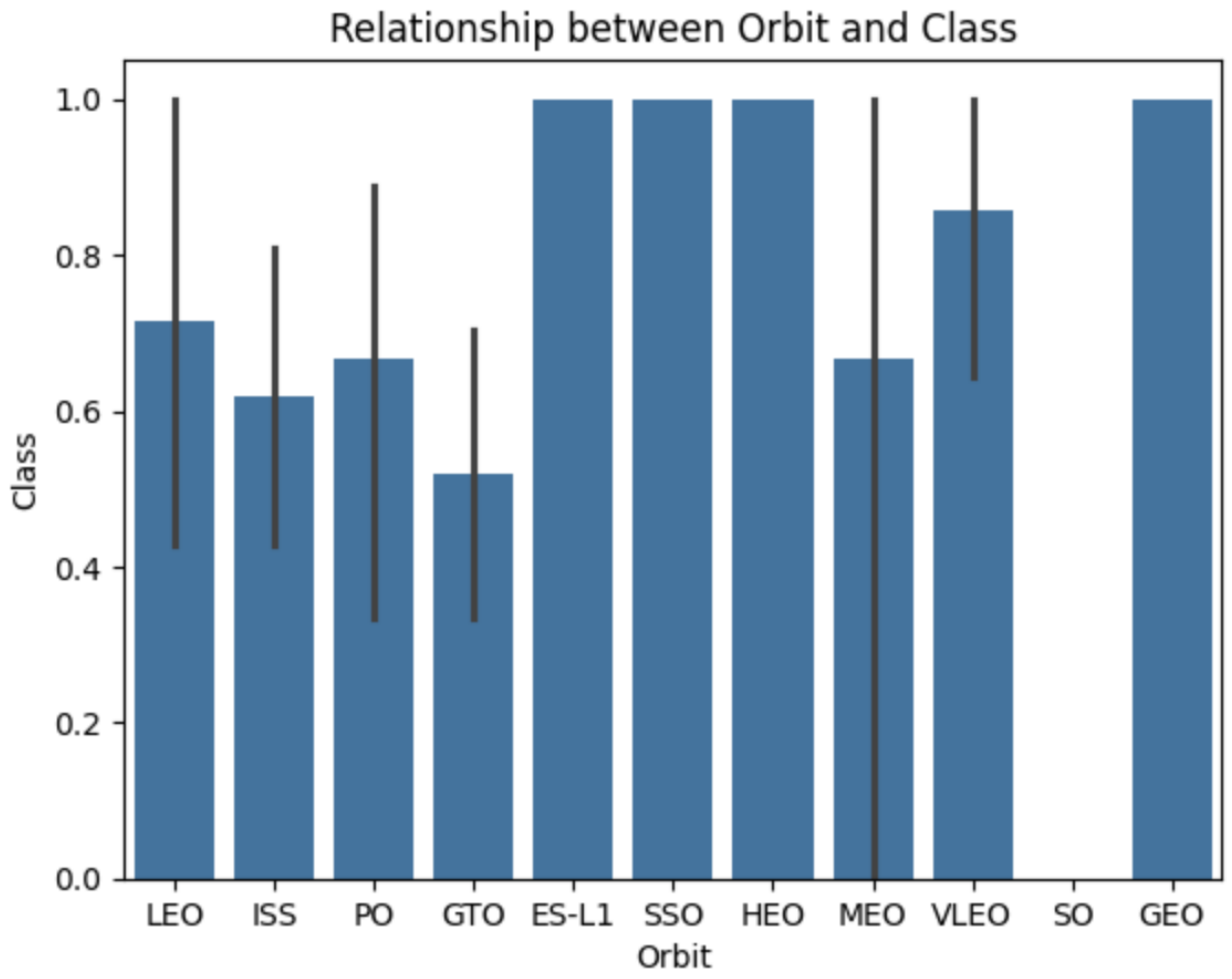
Flight Number vs. Launch Site



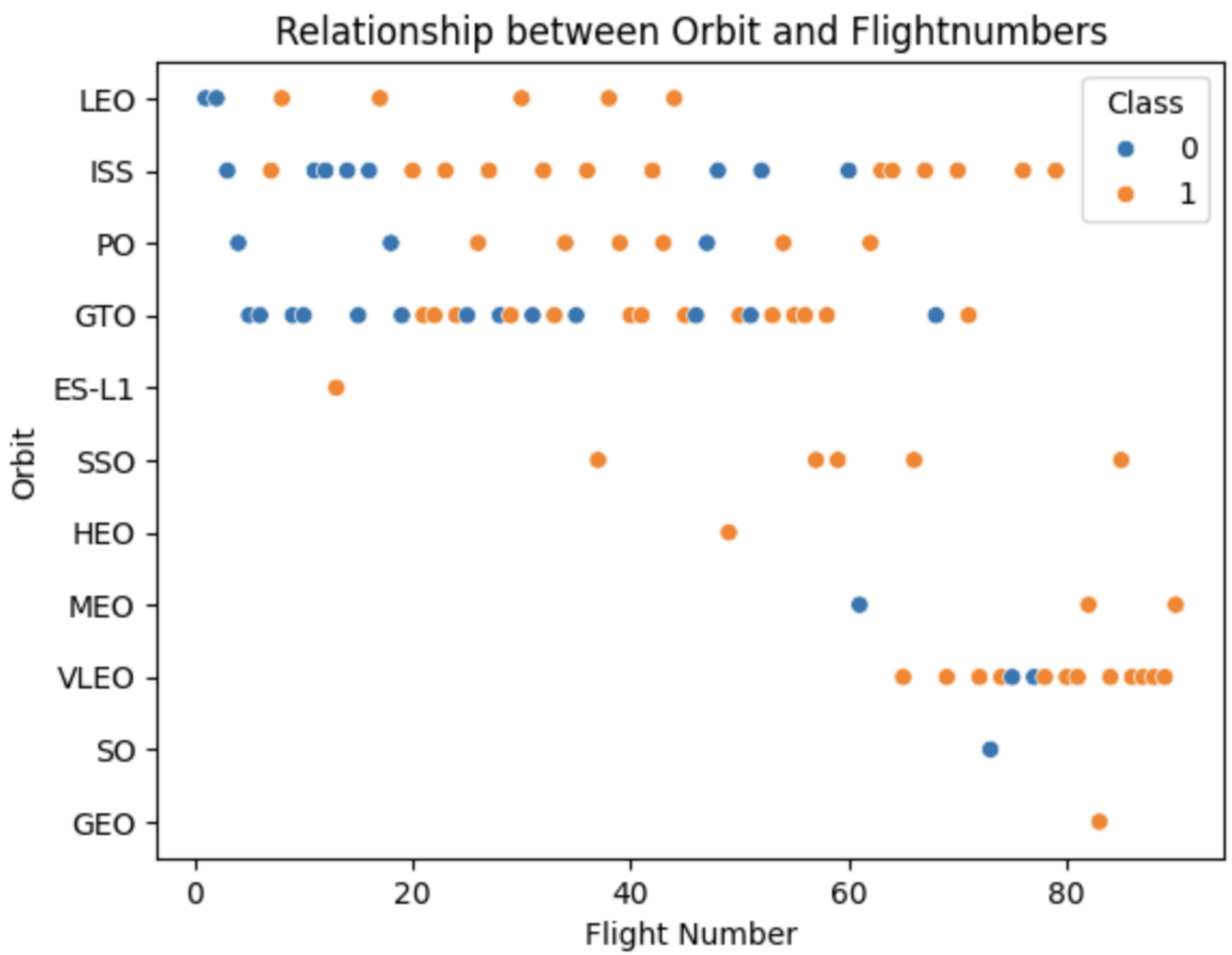
Payload vs. Launch Site



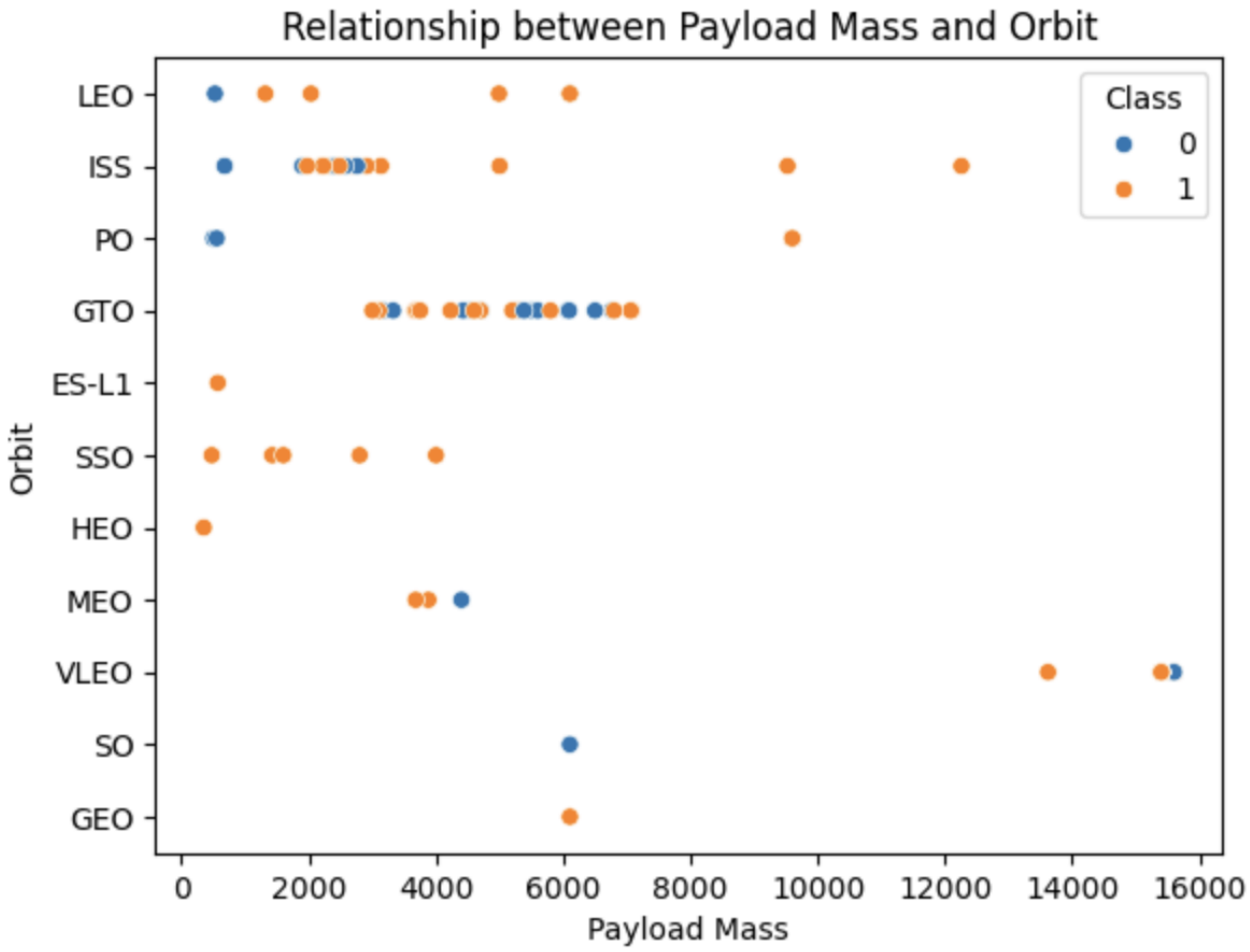
Success Rate vs. Orbit Type



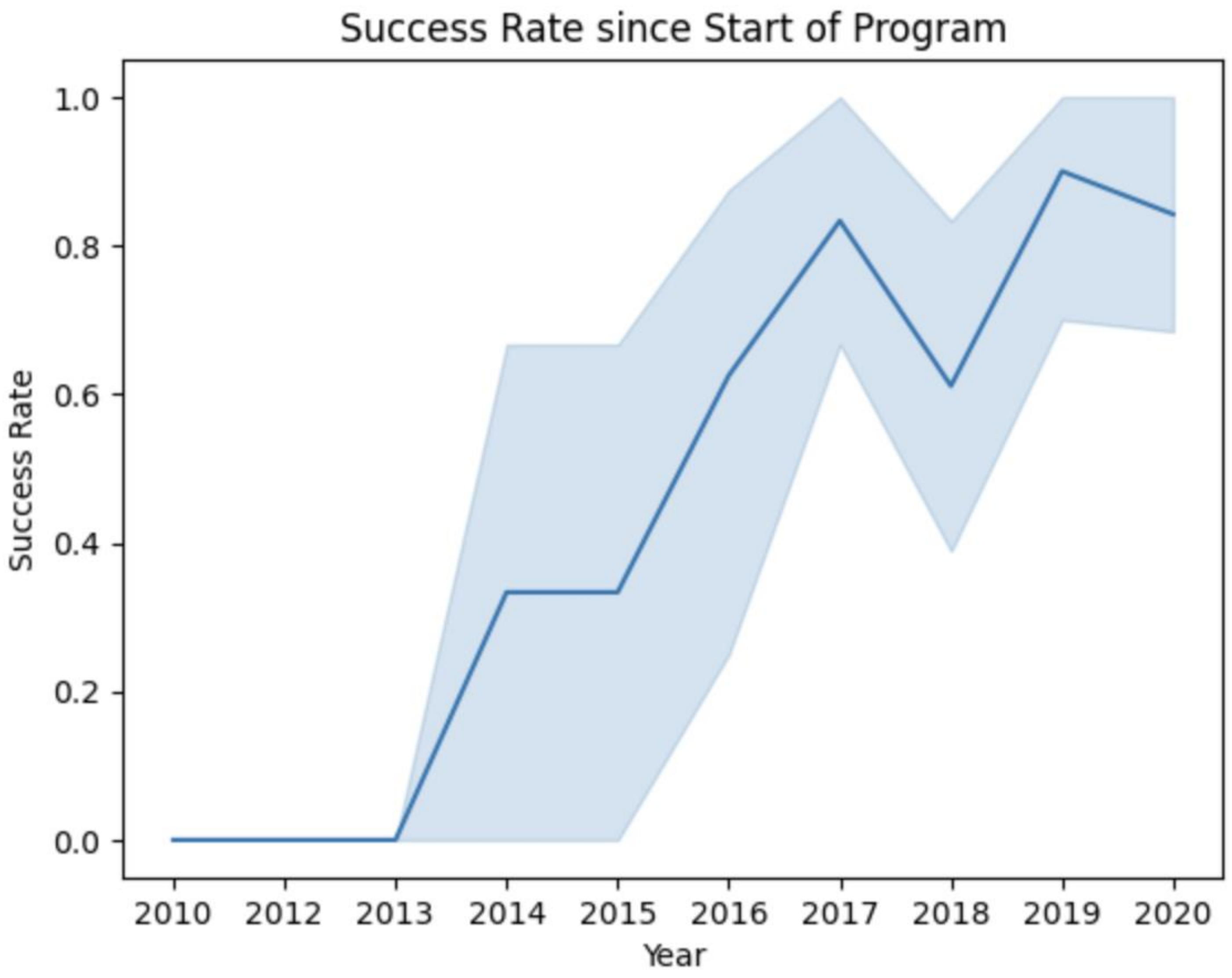
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
%sql select distinct Launch_Site from SPACEXTABLE
```

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

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

This approach gives a unique list of launch sites

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

This SQL query retrieves the first five records from the SPACEXTABLE where the **Launch Site** starts with "CCA." It helps you focus on specific data, making it easier to analyze without being overwhelmed by too much information.

Launch Site Names Begin with 'CCA'

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like "NASA%"
```

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

sum(PAYLOAD_MASS__KG_)
99980

The SQL query computes the total payload mass (in kilograms) carried by all launches where the customer is **NASA** in the **SPACEXTABLE**. This helps to understand how much payload NASA has launched using the provided data.

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) From SPACEXTABLE where booster_version like "F9 v1.1%"
```

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

The SQL query calculates the average payload mass (in kilograms) for all launches associated with the **F9 v1.1** booster version in the **SPACEXTABLE**. This provides insights into the payload capacity of that specific booster model.

First Successful Ground Landing Date

```
%sql select date from SPACEXTABLE where Landing_Outcome == "Success" order by Date asc
```

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

Date
2018-07-22

The SQL query retrieves the launch dates from the **SPACEXTABLE**

where the landing outcome was a **success**. The results are sorted in

ascending order by date, allowing us to see all successful landings chronologically.

Successful Drone Ship Landing with Payload between 4000 and 6000

%sql select Booster_Version from SPACEXTABLE where Landing_Outcome like "%drone ship%"
and PAYLOAD_MASS__KG_ < 6000 and PAYLOAD_MASS__KG_ > 4000 and Landing_Outcome like "Success%"

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

This SQL query selects the **Booster_Version** from the **SPACEXTABLE** where the landing outcome was a **success** on a **drone ship**. It filters for launches with a **payload mass** between **4000 kg** and **6000 kg**. This allows us to identify which booster versions successfully landed on a drone ship with specific payload criteria

Total Number of Successful and Failure Mission Outcomes

```
%sql select Landing_Outcome as "Landing Outcome", count(*) as amount from SPACEXTABLE group by Landing_Outcome
```

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

Landing Outcome	amount
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

This SQL query counts the occurrences of each **Landing Outcome** in the **SPACEXTABLE**, grouping the results by outcome type.

It shows how many launches resulted in success, failure, and other outcomes.

Boosters Carried Maximum Payload

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

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

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

This SQL query retrieves the **Booster Version** and **Payload Mass** from the **SPACEXTABLE** for the launch with the maximum payload mass.

It uses a subquery to find the highest payload mass and returns the corresponding booster version.

2015 Launch Records

Monat	Booster_Version	Launch_Site
Januar	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

This SQL query selects the **month** (in German), **Booster Version**, and **Launch Site** from the **SPACEXTABLE** for all launches in **2015** that had a landing outcome of **failure** on a **drone ship**. The month is derived from the date by using a CASE statement to convert the month number into its corresponding name in German.

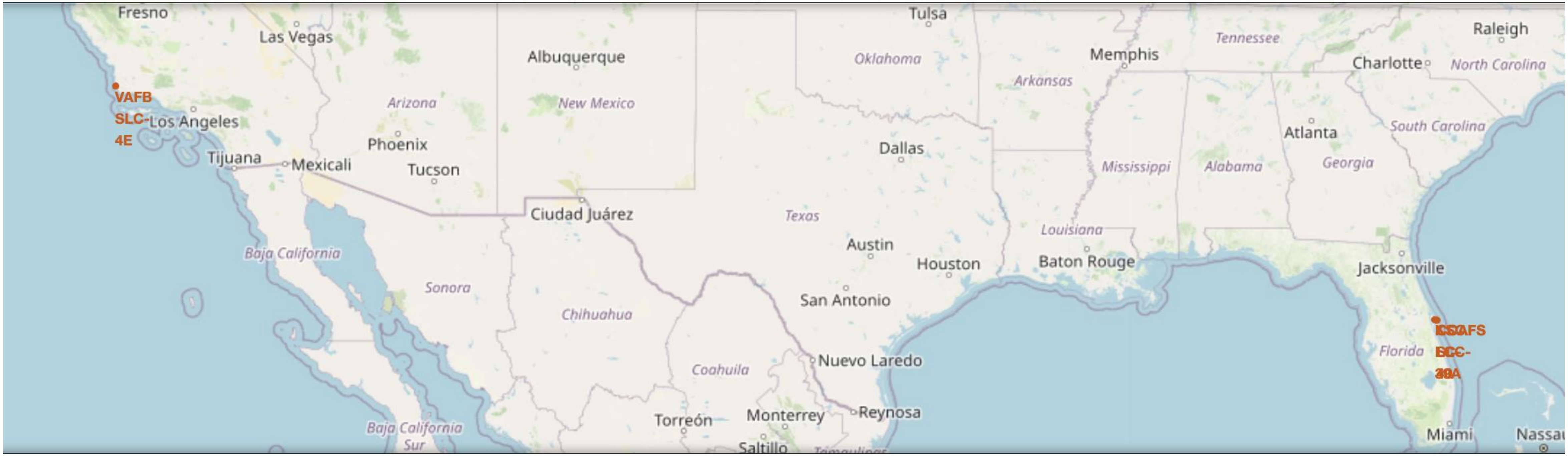
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

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

This SQL query counts the **Landing Outcomes** in **SPACEXTABLE** for launches from **June 4, 2010**, to **March 20, 2017**, grouping the results by outcome and sorting them in descending order by count.

Section 3

Launch Sites Proximities Analysis



Global Launch Sites of SpaceX

- **Markers**:

  Each launch site is indicated by colored circles, with popups showing their names.

- **Geographic Distribution**:

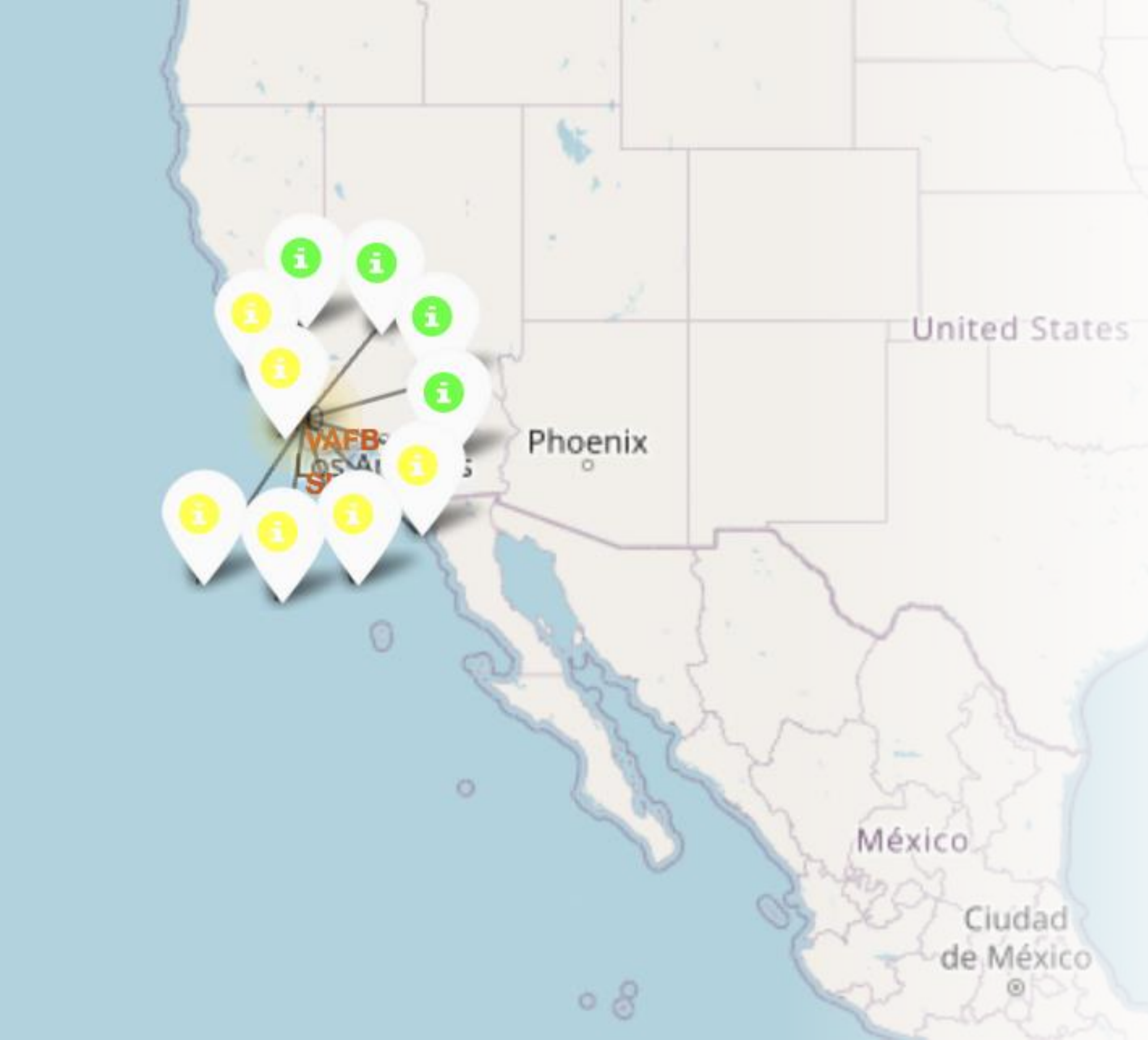
  Locations include NASA Johnson Space Center, Cape Canaveral, Vandenberg SFB, and Guiana Space Centre.

- **Interactivity**:

  Users can zoom and click markers for more information.

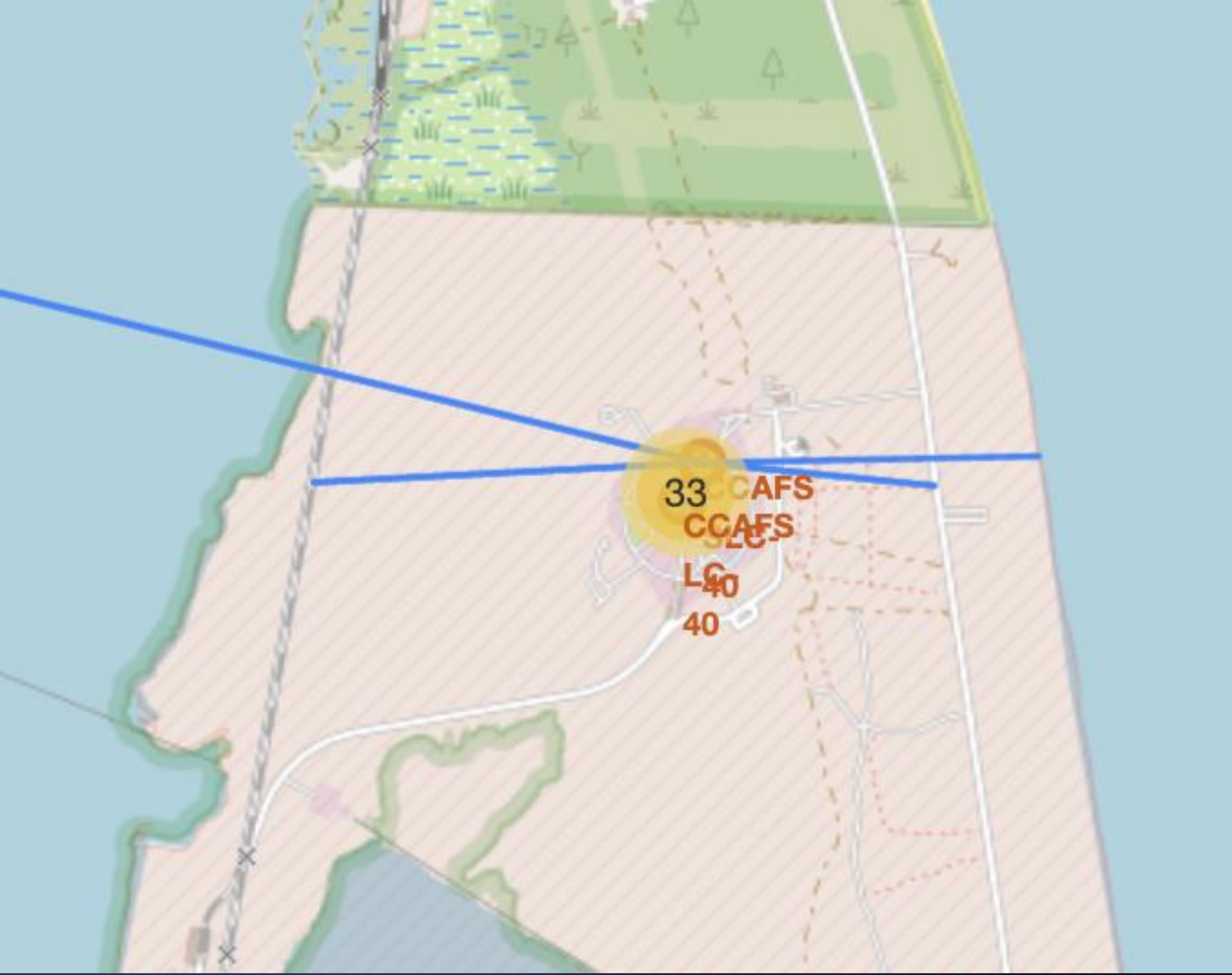
- **Strategic Locations**:

  Highlights SpaceX's use of various global sites based on mission needs.



SpaceX Launch Success and Failure Locations

- **Marker Clusters**:
  - The map features clustered markers representing launch sites. This helps in visualizing areas with multiple launches.
- **Color Coding**:
- Markers are color-coded based on the outcome of the launches:
  - **Green**: Successful launches.
  - **Yellow**: Failed launches.
- **Geographic Spread**
- The map showcases the global distribution of SpaceX launches, highlighting strategic locations.
- **Random Jitter**:
- Small random offsets are applied to marker positions to prevent overlapping, ensuring all markers are visible.
- **Interactive Elements**:
- Users can zoom in and out, click on markers for details, and explore launch success rates across different locations.



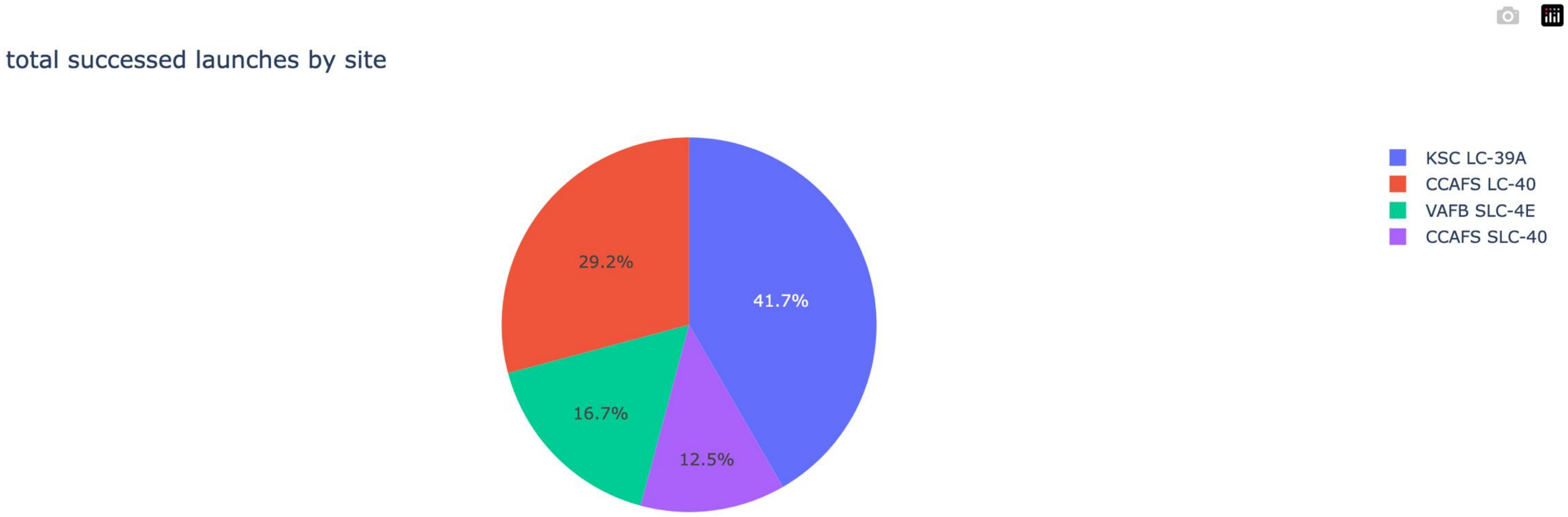
Proximity of Launch Sites to Key Landmarks

- **Titusville**
  - Closest city (28.61° N, -80.81° W) providing essential infrastructure.
- **Nearest Railway**
  - Coordinates (28.56° N, -80.59° W) support transport of materials.
- **Nearest Highway**
  - Coordinates (28.56° N, -80.57° W) facilitate vehicle access to the launch site.
- **Distance Calculation**
  - Measures distance from the launch site to each landmark.
- **Distance Markers**
  - Display distances on the map for quick reference.

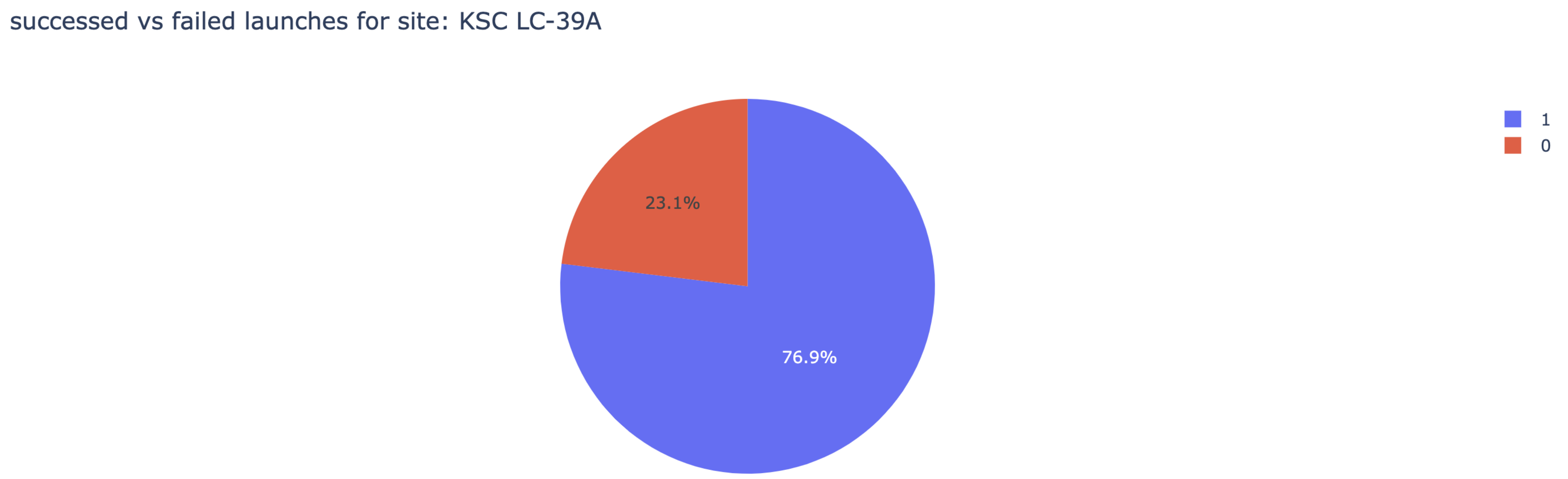
Section 4

Build a Dashboard with Plotly Dash

Total successed launches by site



successed vs failed launches for site: KSC LC-39A

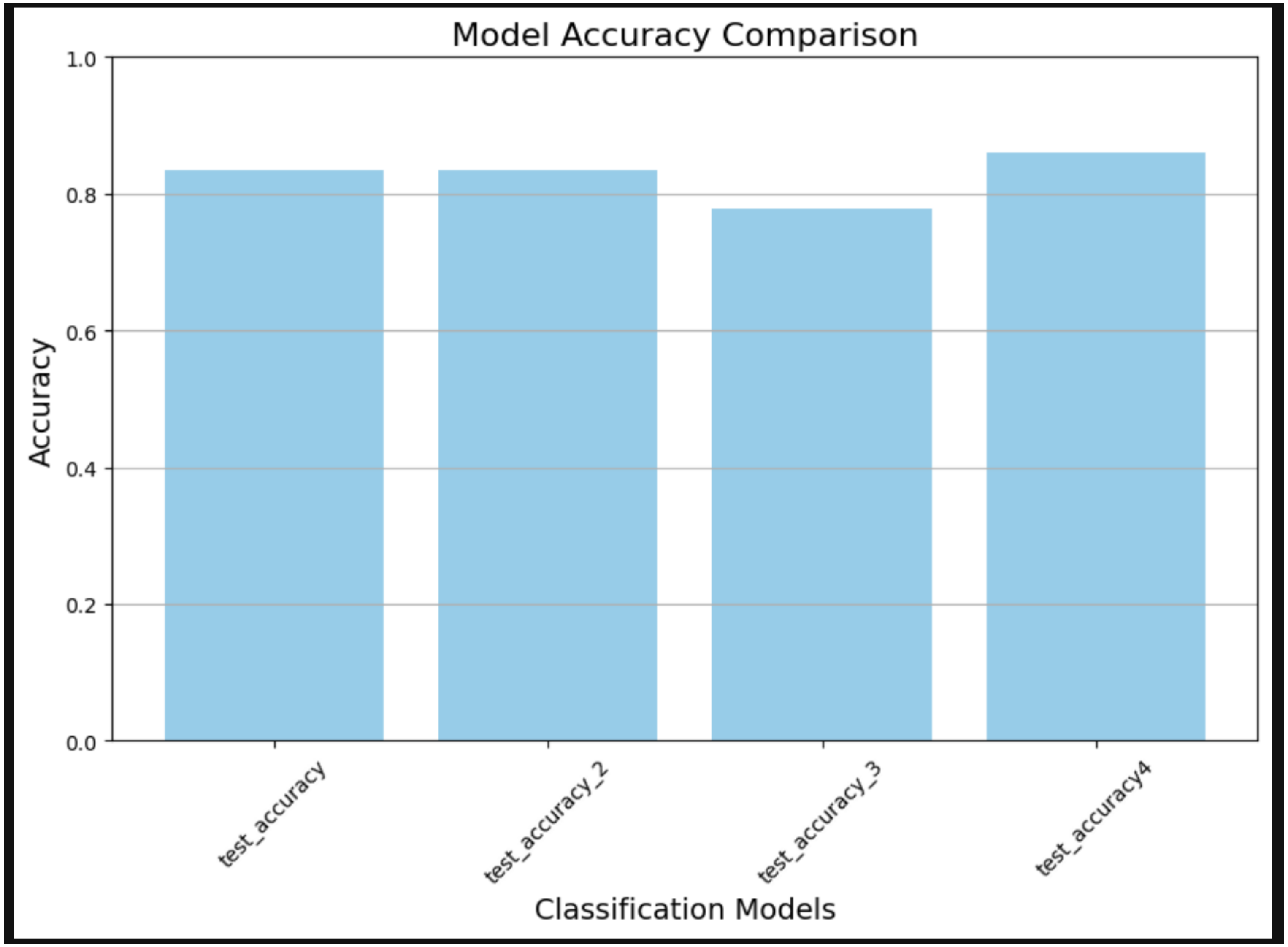


Payload vs outcome fo all sites with Payload Mass range from 2000kg to 7000kg

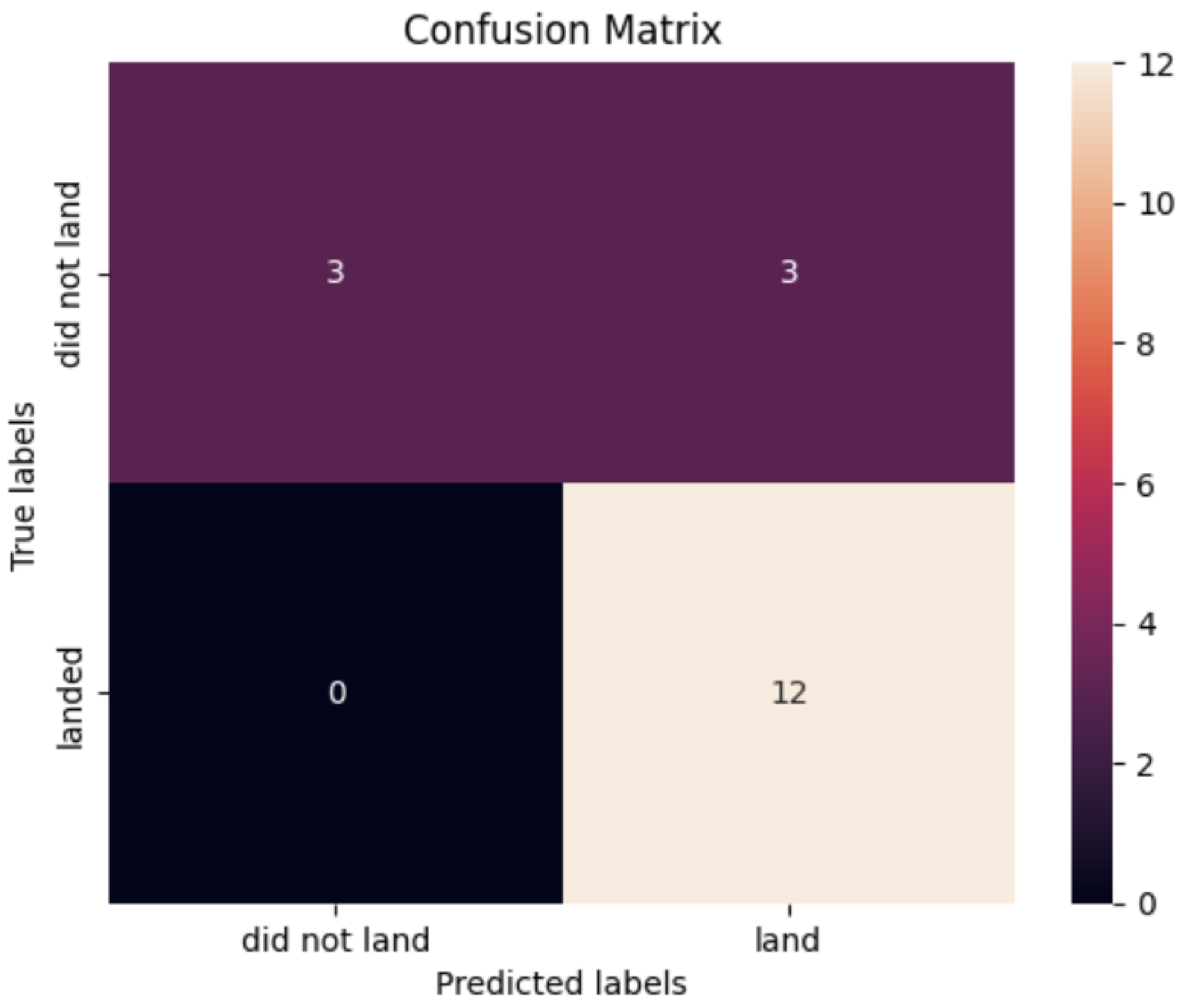
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- SVM with RBF kernel was the most accurate
- Visualizations revealed key trends
- More analysis can enhance model accuracy
- Future work should focus on deeper data insights

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!