IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

Summary of Methodologies

Data Collection: Used APIs and web scraping to gather SpaceX launch data and stored it in a structured format.

Data Wrangling: Processed the data to handle missing values, format inconsistencies, and extract relevant features such as payload mass, launch sites, and booster versions.

Exploratory Data Analysis (EDA): Visualized launch success rates by site, payload mass, and orbit types using scatter plots, pie charts, and success trends.

SQL Queries: Performed SQL operations to extract meaningful insights, including average payload mass, unique launch sites, and filtering by conditions.

Mapping: Created interactive maps to visualize the distribution of launch sites and their clustering.Machine Learning Models: Built classification models to predict launch success based on features like payload and orbit type, using techniques such as decision trees and logistic regression.

Summary of All Results

Exploratory Insights:The highest launch success rate was associated with certain orbits and payload ranges.KSC LC-39A had a high success rate as visualized in dashboards.

SQL Queries: Extracted key metrics such as the first successful ground landing date and the total payload mass carried by NASA (CRS) missions

Model Performance: The confusion matrix indicated a balanced performance, with the classification model achieving reasonable precision and recall.

Interactive Dashboard: Demonstrated real-time filtering and data exploration capabilities for better decision-making regarding launch successes and correlations.

Introduction

Project Background and Context

SpaceX, a leading aerospace manufacturer and space transportation company, has revolutionized the space industry by making rocket launches reusable and cost-effective. However, not all launches are successful, which leads to a need for thorough analysis.

This project aims to analyze historical launch data to understand patterns of success and failure and provide insights to improve future predictions of successful launches.

Problem to be solved

- Which launch sites have the highest success rates?
- How does the payload mass impact the success of launches? Which orbit types correlate with higher success rates?
- Can machine learning models accurately predict launch outcomes based on given features?
- What are the contributions of different customers (e.g., NASA) to successful missions?

Section 1

Methodology

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

Data was collected using **web scraping** from SpaceX's official launch records and public repositories.

Python libraries like BeautifulSoup and requests were used to extract relevant information.

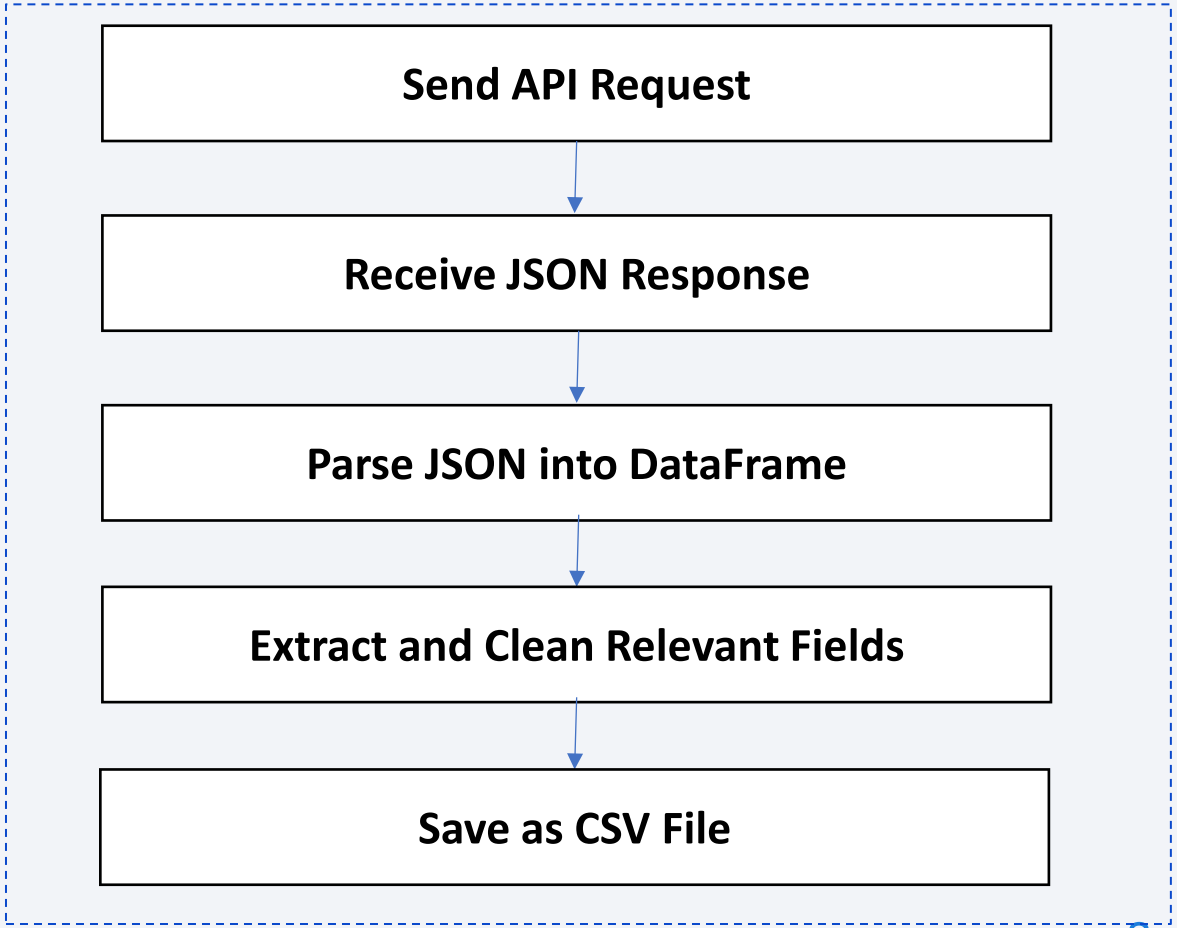
The data included key details such as launch site, booster version, payload mass, and mission outcomes (success or failure).

The scraped data was saved as a structured Pandas DataFrame for processing and cleaning.

Data Collection – SpaceX API

The endpoint URL of the SpaceX API (e.g., https://api.spacexdata.com/v4/launches).

https://github.com/macheam/IBM-DATASCIENCE



Used Python's requests library to send GET requests to retrieve the HTML page content

Data Collection - Scraping

Selected SpaceX official website or Wikipedia launch history page for data extraction.

Used Python's requests library to send GET requests to retrieve the HTML page content.

Utilized BeautifulSoup to parse the HTML response and extract relevant data (e.g., launch date, booster version, payload mass).

Identify Web Source

Send HTTP Request

Parse HTML Content

Clean and Structure Data

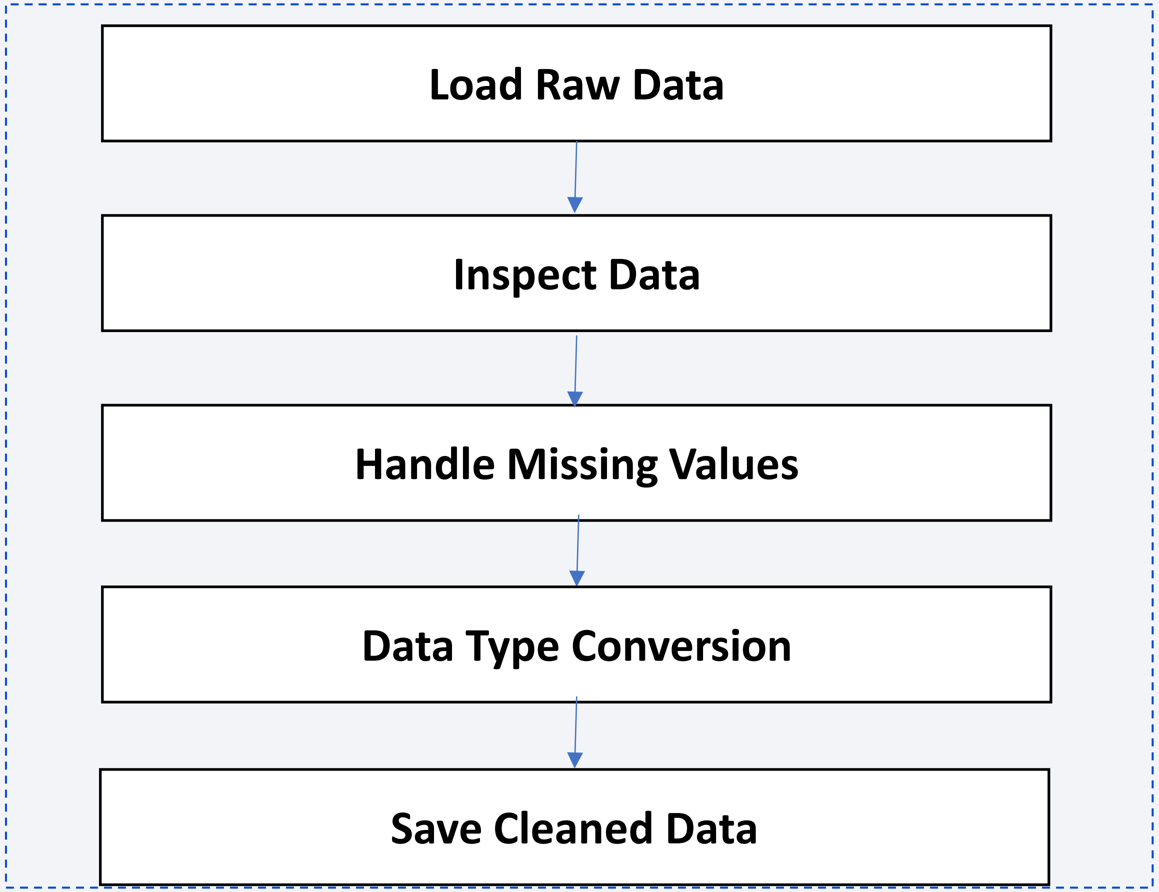
Save Processed Data

Data Wrangling

Selected SpaceX official website or Wikipedia launch history page for data extraction.

Used Python's requests library to send GET requests to retrieve the HTML page content.

Utilized BeautifulSoup to parse the HTML response and extract relevant data (e.g., launch date, booster version, payload mass).



https://github.com/macheam/IBM-DATASCIENCE

EDA with Data Visualization

Pie Chart – Launch Success vs Failure:

- Why: To visualize the proportion of successful and failed launches.
- Purpose: Helps to quickly understand the overall success rate of launches at different sites.

Scatter Plot – Payload Mass vs Launch Outcome:

- Why :To analyze the relationship between payload mass and launch outcome. Purpose: Helps to quickly understand the overall success rate of launches at different sites.

Bar Chart (Optional for Comparison):

- To compare the number of launches across different launch sites.
- Purpose: Help Pinpoint the most active launch sites and their performance

EDA with SQL

SELECT Query:Retrieved specific columns such as Launch Site, Payload Mass (kg), Booster Version, and Class (success/failure).Example:SELECT Launch_Site, Payload_Mass, Booster_Version, Class FROM spacex_table;

WHERE Clause:Filtered records based on conditions (e.g., successful launches or specific sites).Example:SELECT * FROM spacex_table WHERE Launch_Site = 'KSC LC-39A' AND Class = 1;

GROUP BY and COUNT:Counted the number of successful and failed launches at each site.Example:SELECT Launch_Site, COUNT(Class) AS Total_Launches, SUM(Class) AS Success_Count FROM spacex_table GROUP BY Launch_Site;

ORDER BY: Ordered the results to display top-performing sites by success count.Example:SELECT Launch_Site, SUM(Class) AS Success_Count FROM spacex_table GROUP BY Launch_Site ORDER BY Success_Count DESC;

Build an Interactive Map with Folium

1. Map Objects Added:

Markers (Location Pins):

Purpose: To indicate the exact location of each SpaceX launch site.

Why: Helps visualize where launches occurred geographically.

Details: Each marker included information such as the launch site's name and launch success rate.

Circle Markers:

Purpose: To show an approximate region of interest around launch sites.

Why: Highlights the area and differentiates sites visually based on payload mass or other attributes.

Details: The color and radius of the circles varied based on metrics such as the total number of launches.

Popup Information:

Purpose: To display detailed information (e.g., Launch Site, Booster Version, and Class) when clicking on a marker.

Why: Provides additional insights without cluttering the map.

Lines/Polylines (if applicable):

Purpose: To draw connections or indicate paths related to launch trajectories.

Why: Visualizes flight paths for better interpretation of mission details.

Build a Dashboard with Plotly Dash

Dropdown Menu for Launch Sites:

- **Purpose:** Allows users to select a specific launch site or display data for all sites.
- **Why:** Enables dynamic exploration of data by filtering results based on launch site.

Range Slider for Payload Mass:

- **Purpose:** Allows users to filter launches by payload mass range.
- **Why:** Helps focus the analysis on specific payload mass ranges to detect patterns.

Hover and Click Interactions:

- **Purpose:** Provides additional launch details (e.g., booster version and payload mass) when hovering over or clicking on points in the scatter plot.
- **Why:** Enhances data exploration and provides detailed insights.

Predictive Analysis (Classification)

Flowchart:

Start → 2. Load and Preprocess Data → 3. Split Train/Test Sets

Select Models → 5. Cross-Validation Evaluation → 6. Hyperparameter Tuning

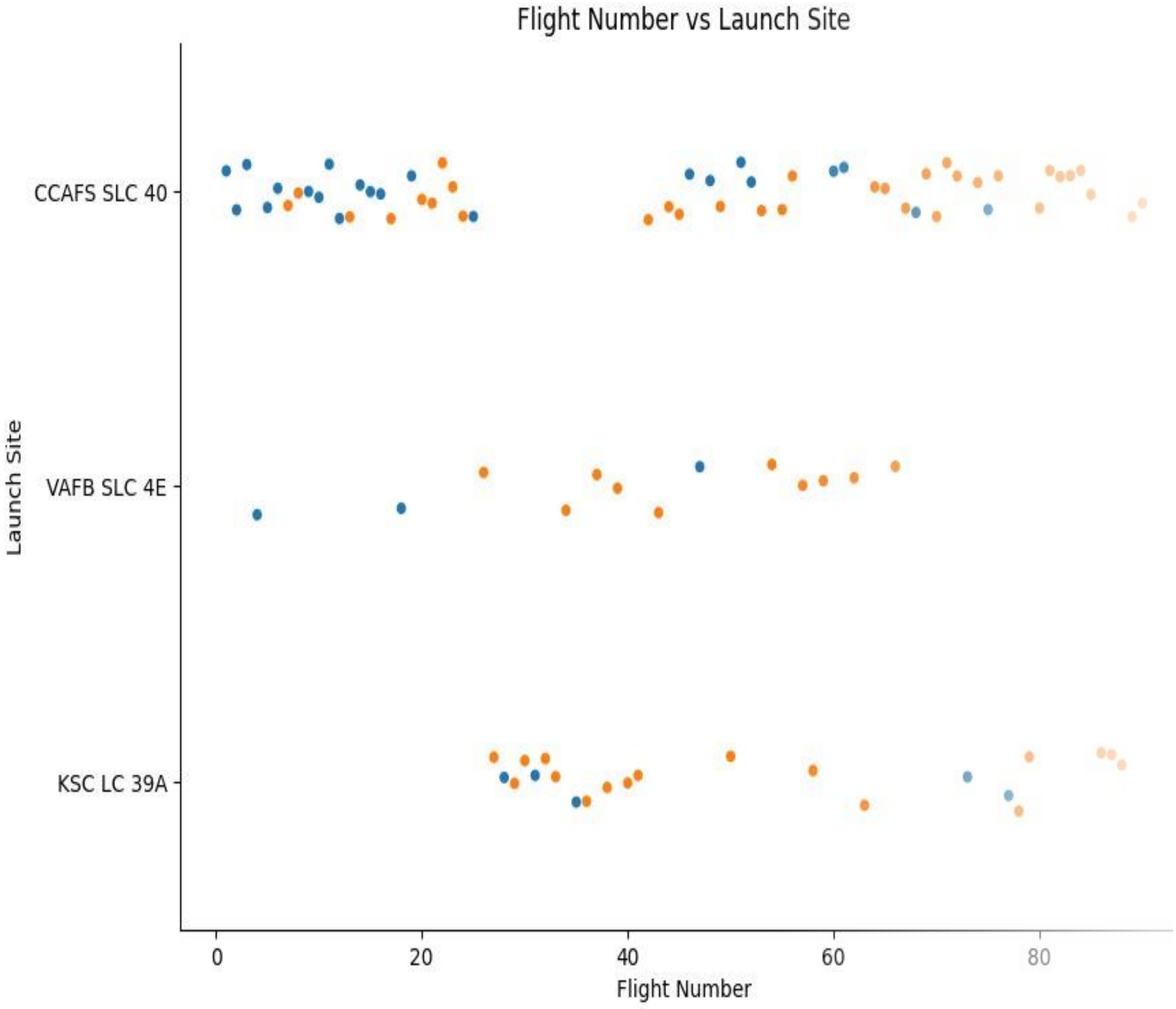
Choose Best Model → 8. Test Best Model → End

Results

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

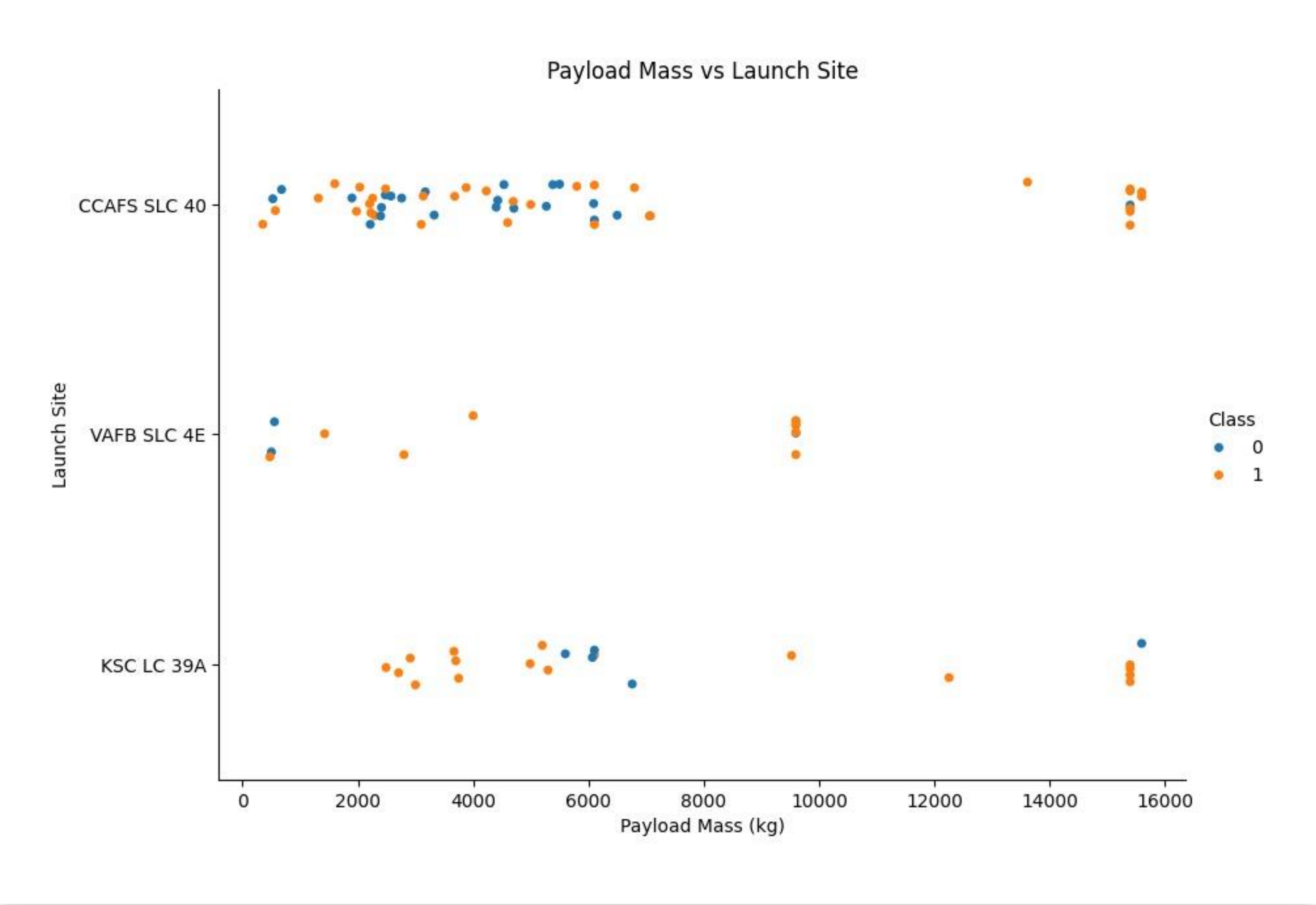
Section 2

Insights drawn from EDA



- **Orange dots (Class 1):** Successful launches.
- **Blue dots (Class 0):** Failed launches.

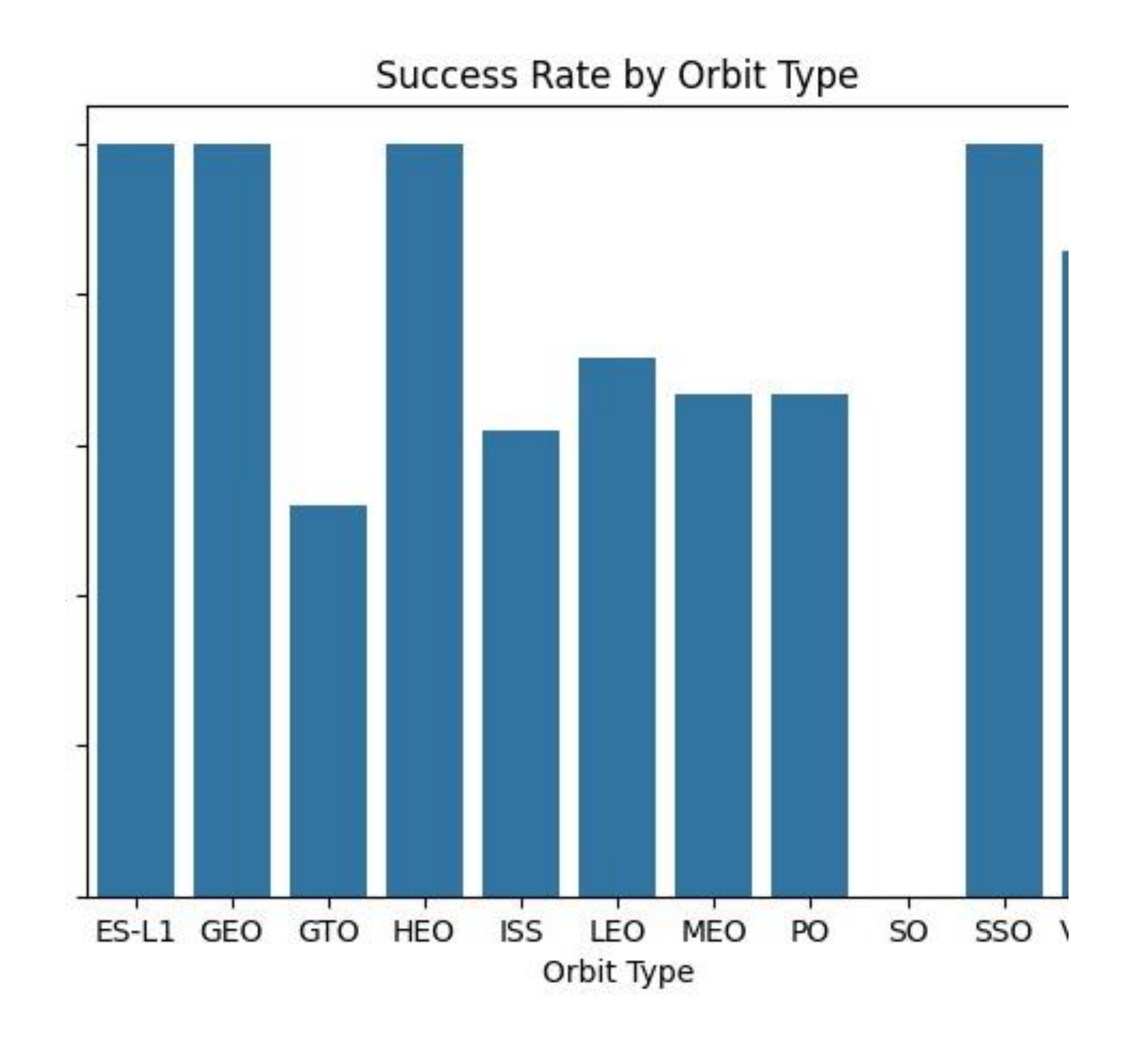
- The plot helps compare how many launches happened at each site and shows that success rates generally improved over time, especially at sites like KSC LC 39A and CCAFS SLC 40.



Payload vs. Launch Site

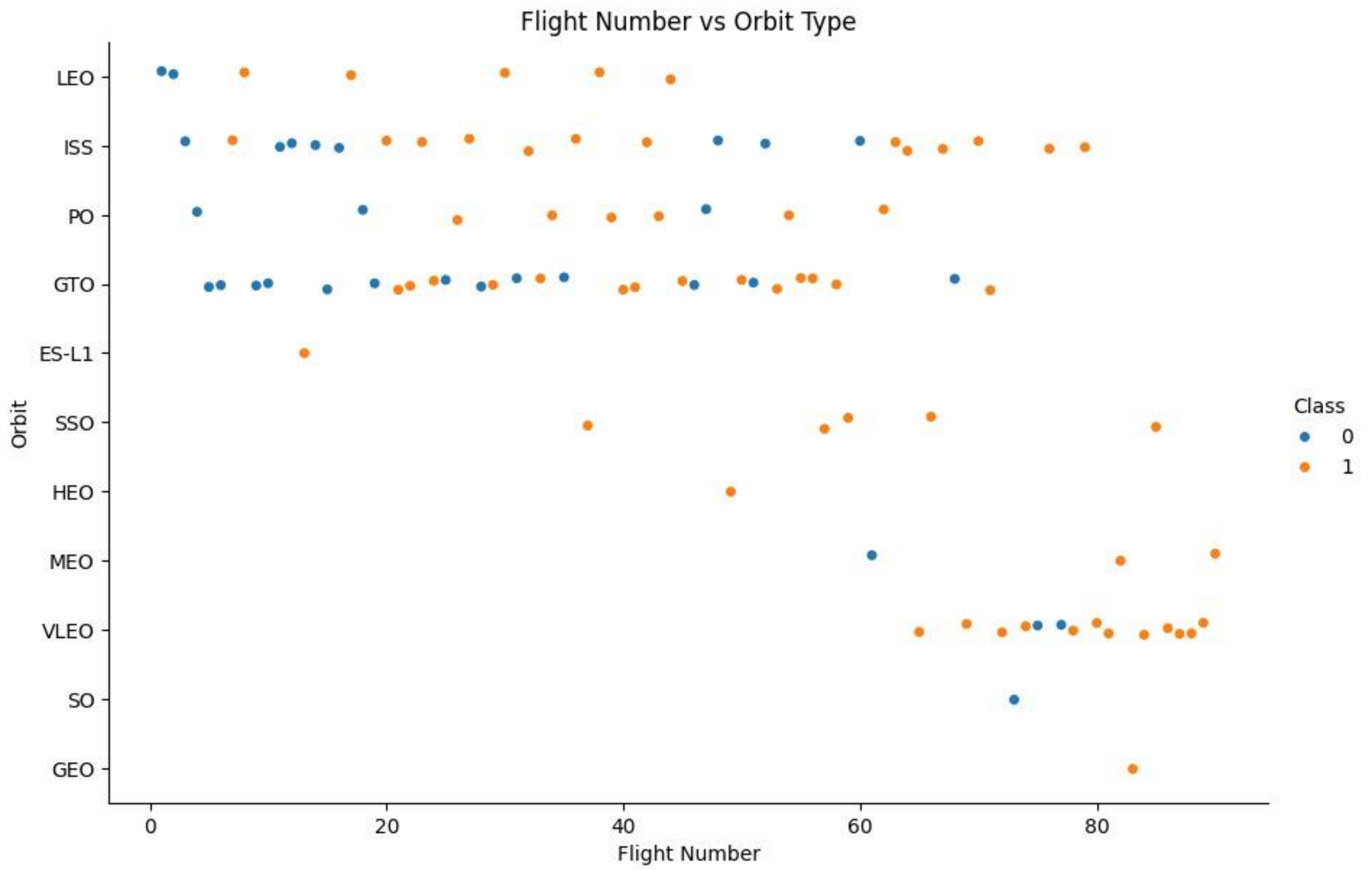
Success Rate vs. Orbit Type

- The bar chart shows **success rates** for different **orbit types**.
- **High success:** ES-L1, GEO, GTO, SSO (near 100%).
- **Moderate success:** LEO, ISS, MEO (~70-80%).
- **Low success:** HEO (much lower).
- Overall, SpaceX performs reliably across most orbit types, with some variability in elliptical orbits like HEO.



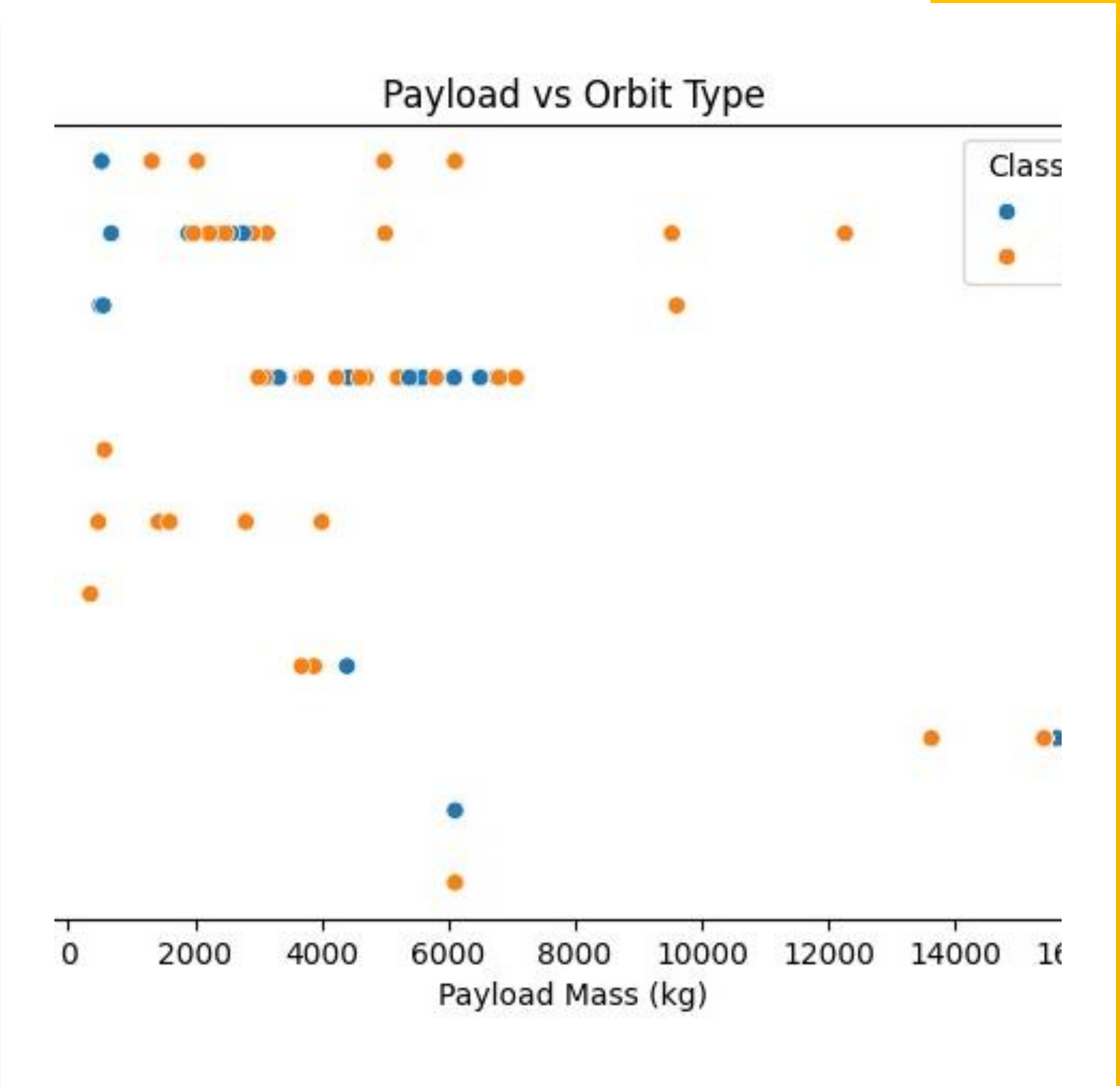
Flight Number vs. Orbit Type

- SpaceX’s performance has improved across most orbits over time, with fewer failures in later flights.Orbits like LEO, ISS, and GTO have more launches, indicating they are common mission types.Some orbits, such as HEO, continue to show variability in success.



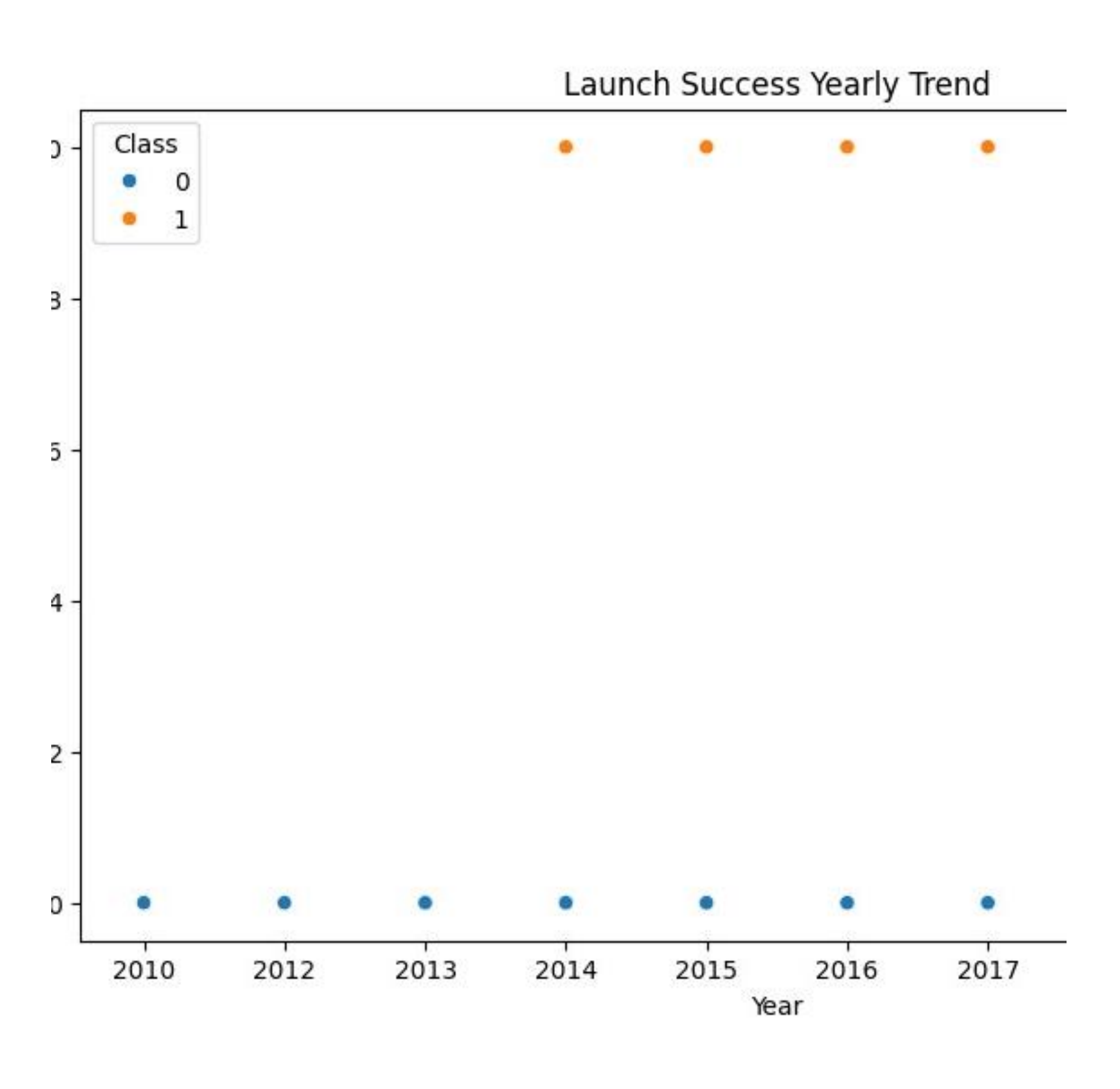
Payload vs. Orbit Type

- The scatter plot shows the relationship between **payload mass** and **orbit type**:
- **LEO and ISS:** High success rates across various payload masses.
- **GTO:** Successful launches for moderate payloads (4,000-6,000 kg).
- **Heavy payloads (~15,000 kg):** Successfully launched to orbits like GEO and VLEO.
- **HEO:** More failures regardless of payload mass.
- Overall, SpaceX handles a wide range of payloads effectively, except for elliptical orbits like HEO.



Launch Success Yearly Trend

- The plot shows SpaceX's **yearly trend of launch outcomes** from **2010 to 2020**:
- **Orange dots (1):** Successful launches.
- **Blue dots (0):** Failed launches.
- **Key Insights:**
- Failures were more common in the early years (2010–2015).
- From **2017 onward**, launches are consistently successful, indicating significant improvements in reliability over time.
- **Conclusion:** SpaceX has steadily improved its launch success rate, achieving near-perfect performance in recent years.



All Launch Site Names

- Find the names of the unique launch sites

  CCAFS SLC 40' 'VAFB SLC 4E' 'KSC LC 39A

- The SQL query retrieves the unique launch site names from the SPACEXTBL table using SELECT DISTINCT.

Launch Site Names Begin with 'CCA'

- The SQL query retrieves 5 records from the SPACEXTBL table where the launch site name starts with "CCA"
- (e.g., CCAFS LC-40).Key Details:All 5 records correspond to launches from CCAFS LC-40.The payloads include missions such as "Dragon Spacecraft Qualification Unit" and "SpaceX CRS-2".The Mission Outcomes show that all launches were successful, but landing attempts varied (e.g., "No attempt", "Failure (parachute)").

Task 2

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

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

[32] ✓ 0.0s Python

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

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 by F9 v1.1

- The SQL query calculates the **total payload mass (in kg)** carried by boosters for launches by **NASA (CRS)**.
- **Result:**
- The **total payload mass** launched for **NASA (CRS)** missions is **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)';
```

[33] ✓ 0.0s

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

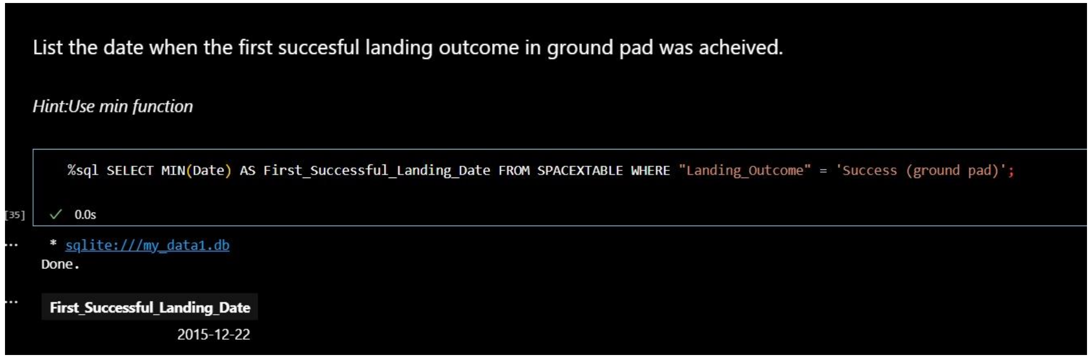
Total_Payload_Mass
45596

Average Payload Mass

- The SQL query calculates the **average payload mass (in kg)** carried by the **booster version F9 v1.1**.
- **Result:**
- The **average payload mass** for **F9 v1.1** is **2928.4 kg**.
- This shows that the F9 v1.1 booster typically carries around **2.9 tons** per launch.

First Successful Ground Landing Date

- The SQL query retrieves the earliest date when a successful landing on a ground pad occurred using the MIN function.Result:
- The first successful ground pad landing happened on 2015-12-22.This shows that SpaceX achieved its first ground pad landing success in December 2015.

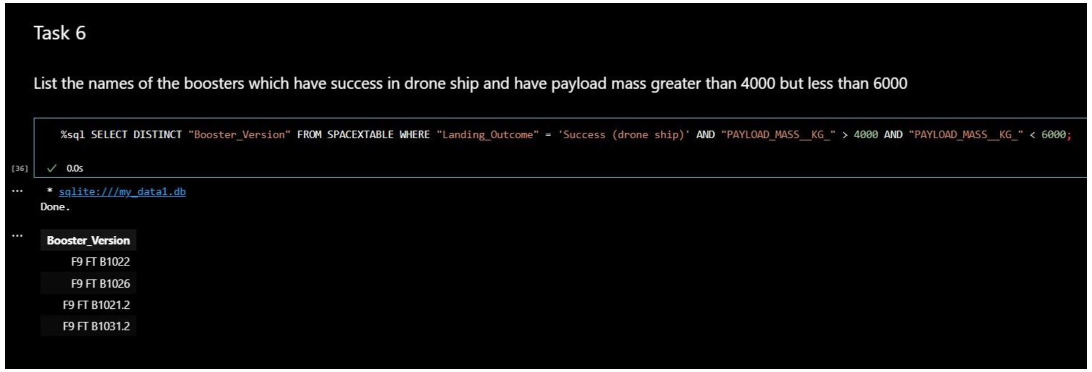


Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL query lists **booster versions** that had a **successful landing** on a **drone ship** and carried a **payload mass** between **4,000 kg and 6,000 kg**.
- **Result:**
  The following boosters met the criteria:

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

- This shows that specific boosters successfully carried medium payloads and landed on drone ships.



Total Number of Successful and Failure Mission Outcomes

- The data provides insight into SpaceX's various landing outcomes, showing consistent success with ground pads and drone ships.

List the total number of successful and failure mission outcomes

```
%sql SELECT "Landing_Outcome", COUNT(*) AS "Total_Count" FROM SPACEXTABLE GROUP BY
```

[37] ✓ 0.0s

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

Landing_Outcome	Total_Count
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

Boosters Carried Maximum Payload

- The SQL query lists the **booster versions** that carried the **maximum payload mass** using a **subquery** to find the maximum payload value.
- **Result:**
- The maximum payload mass is **15,600 kg**.
- Several booster versions, such as **F9 B5 B1048.4**, **F9 B5 B1051.3**, and **F9 B5 B1049.7**, carried this maximum payload.

List the names of the booster_versions which 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);
```

[38] ✓ 0.0s

```
 * 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

The SQL query extracts and displays records for the year 2015 were Landing outcome Failure on a drone ship..

Failures occurred in **January** and **April 2015**, showing early challenges with at-sea landings.

List the records which will display the month names, failure landing_outcomes in drone ship ,booster vers

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and su

```
%%sql
SELECT
    CASE
        WHEN substr(Date, 6, 2) = '01' THEN 'January'
        WHEN substr(Date, 6, 2) = '02' THEN 'February'
        WHEN substr(Date, 6, 2) = '03' THEN 'March'
        WHEN substr(Date, 6, 2) = '04' THEN 'April'
        WHEN substr(Date, 6, 2) = '05' THEN 'May'
        WHEN substr(Date, 6, 2) = '06' THEN 'June'
        WHEN substr(Date, 6, 2) = '07' THEN 'July'
        WHEN substr(Date, 6, 2) = '08' THEN 'August'
        WHEN substr(Date, 6, 2) = '09' THEN 'September'
        WHEN substr(Date, 6, 2) = '10' THEN 'October'
        WHEN substr(Date, 6, 2) = '11' THEN 'November'
        WHEN substr(Date, 6, 2) = '12' THEN 'December'
    END AS MonthName,
    "Landing_Outcome",
    "BoosterVersion",
    "LaunchSite"
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015'
AND "Landing_Outcome" LIKE '%Failure%'
AND "Landing_Outcome" LIKE '%drone ship%';
```

[39] ✓ 0.0s

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

MonthName	Landing_Outcome	"BoosterVersion"	"LaunchSite"
January	Failure (drone ship)	BoosterVersion	LaunchSite
April	Failure (drone ship)	BoosterVersion	LaunchSite

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

- This SQL query retrieves and ranks the **count of different landing outcomes** between the dates **2010-06-04** and **2017-03-20**, ordered in descending order of the count.

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 BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY "Outcome_Count" DESC;
```

✓ 0.0s

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

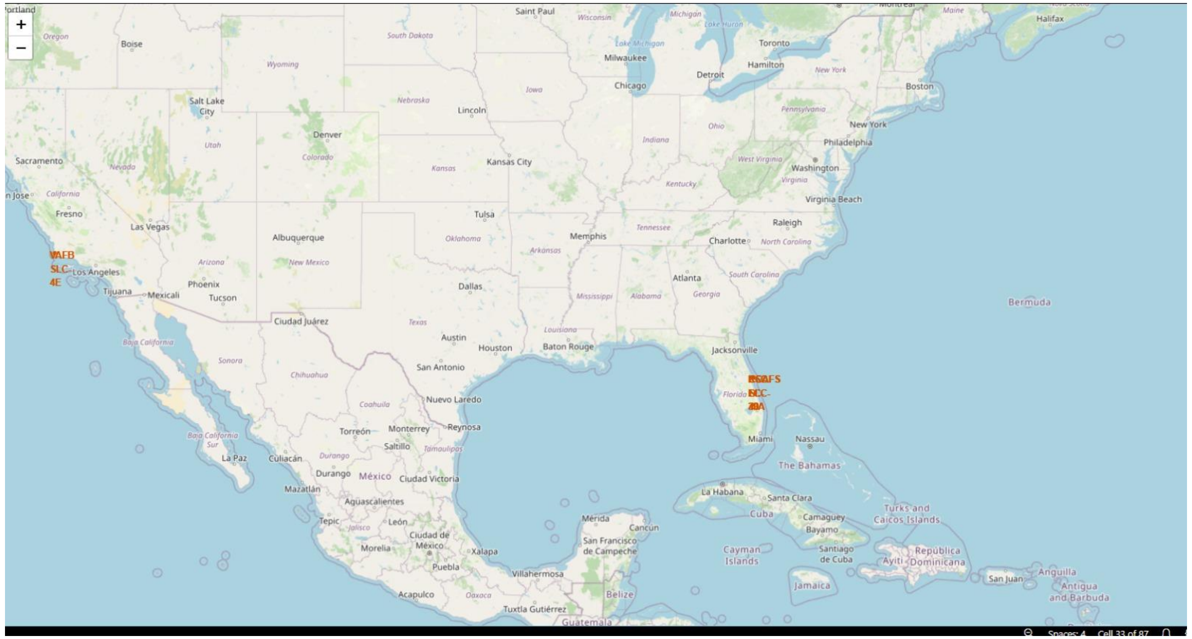
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

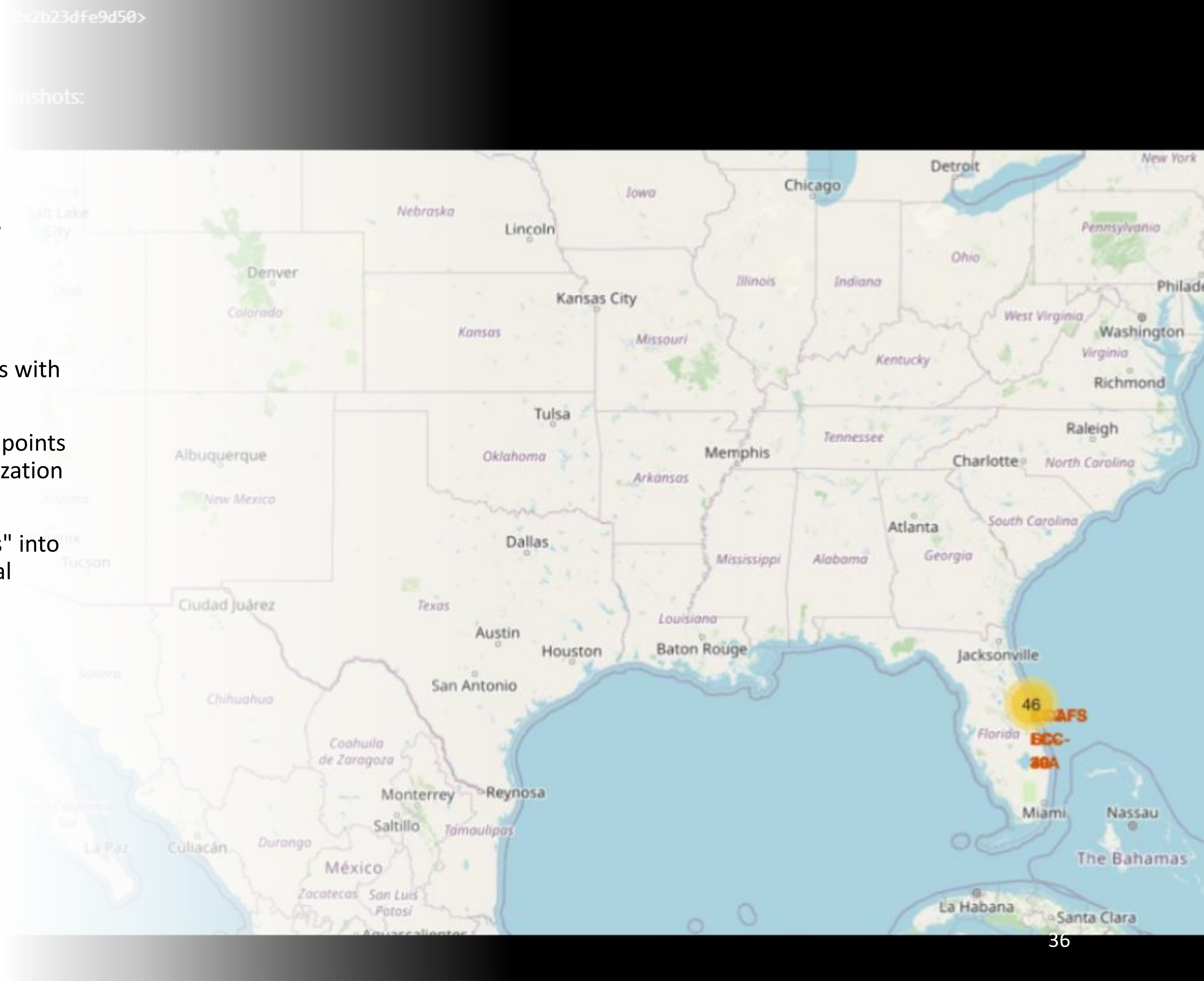
<Folium Map Screenshot 1>

- This map shows the locations of SpaceX launch sites in the U.S.:
- **VAFB SLC 4E** (Vandenberg Air Force Base, California) is marked on the west coast.
- **CCAFS LC-40** and **KSC LC-39A** (both in Florida) are marked on the east coast



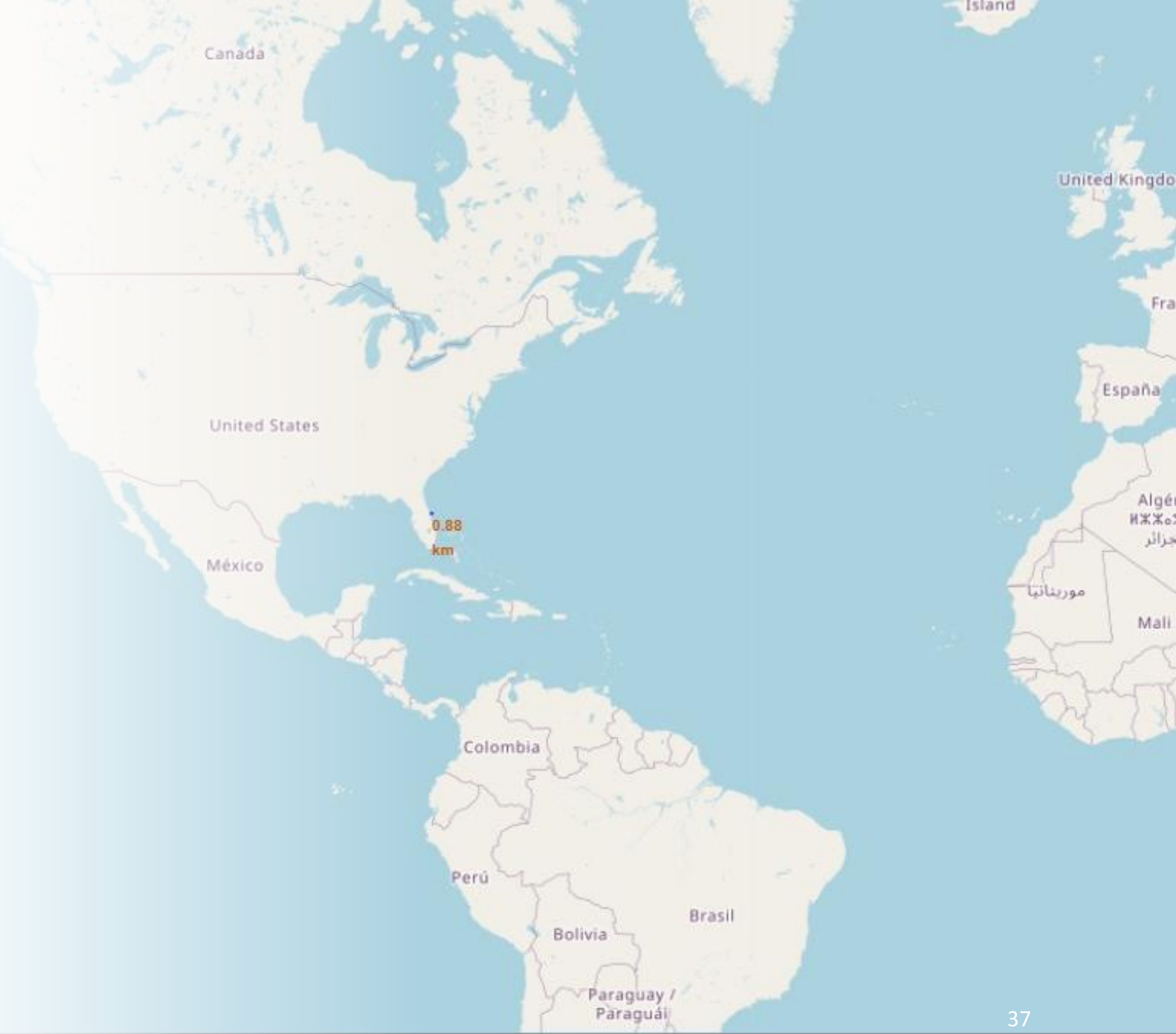
<Folium Map Screenshot 2>

- This map displays SpaceX launch sites with **marker clusters**:
- **Cluster markers** group multiple data points into a single marker for better visualization when zoomed out.
- When zoomed in, the cluster "breaks" into individual markers showing the actual locations.



<Folium Map Screenshot 3>

- This map visualizes the **distance measurement** between two selected points:
- The **0.88 km** label indicates the calculated distance between the SpaceX launch site marker and a specific point.
- It demonstrates the use of a distance-measuring tool in interactive maps, useful for determining proximity between locations such as launch sites, landing zones, or observation points

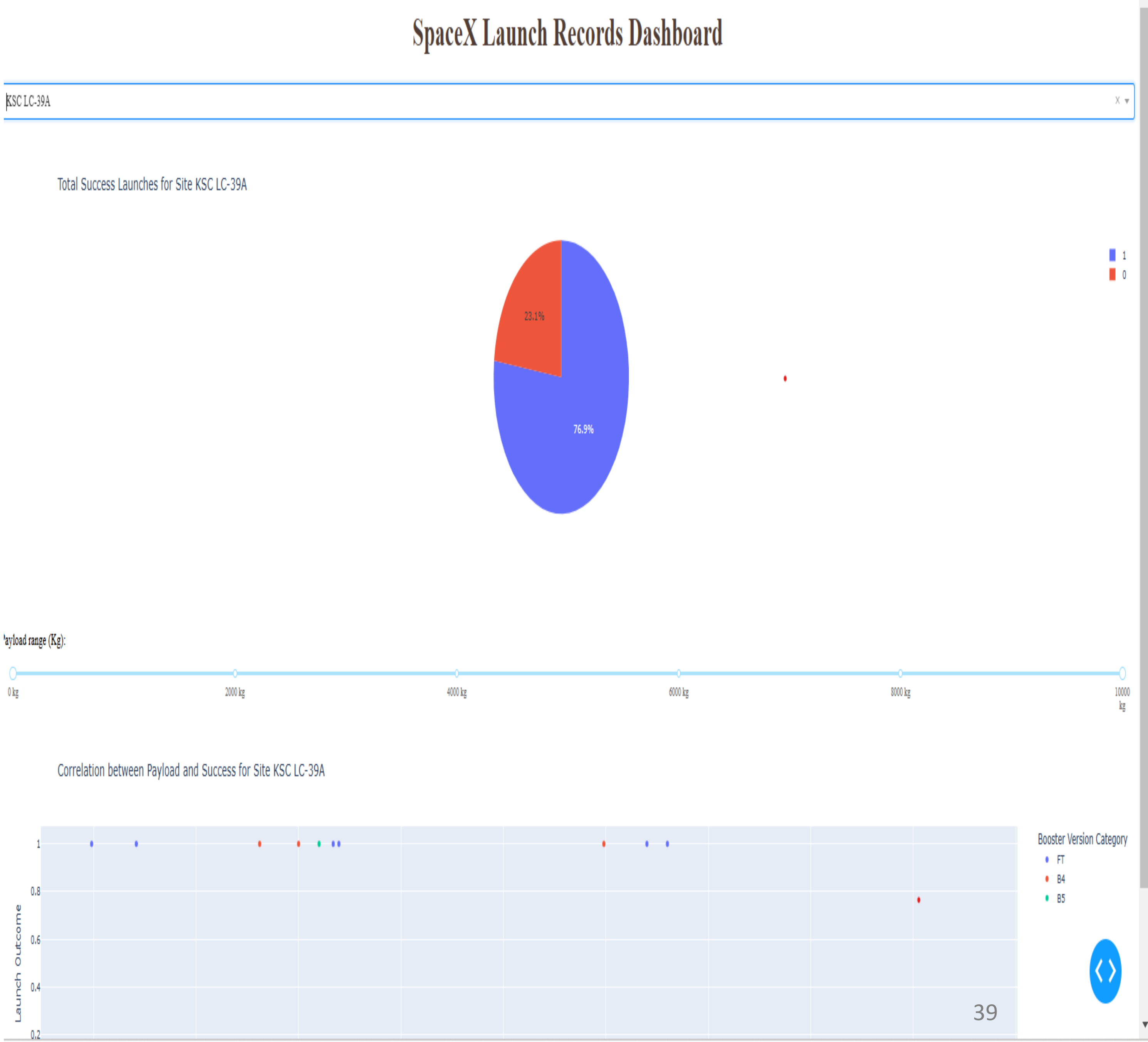


Section 4

Build a Dashboard with Plotly Dash

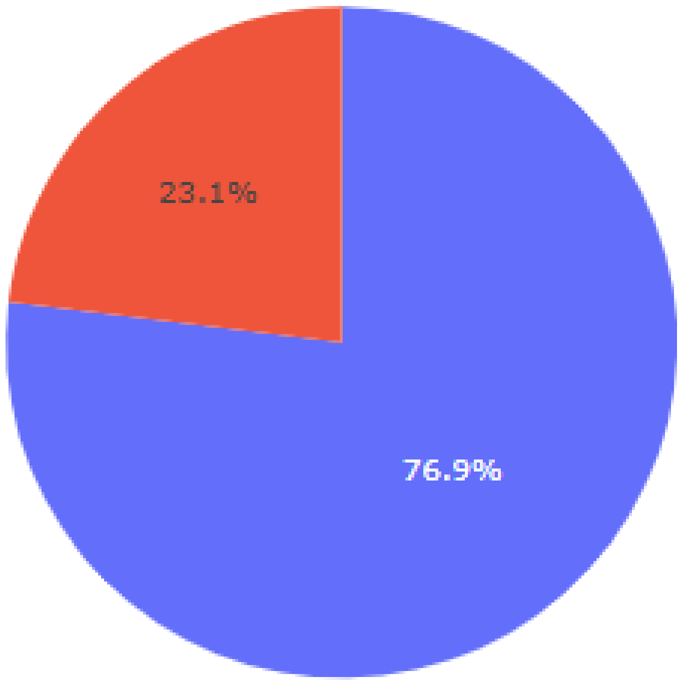
SpaceXLaunch Dashboard

- This dashboard shows SpaceX launch performance:
- **Dropdown:** Selects launch sites to view data.
- **Pie Chart:** Shows the success and failure rates.
- **Payload Slider:** Filters launches based on payload mass.
- **Scatter Plot:** Displays the correlation between payload mass and launch success, with booster versions color-coded.



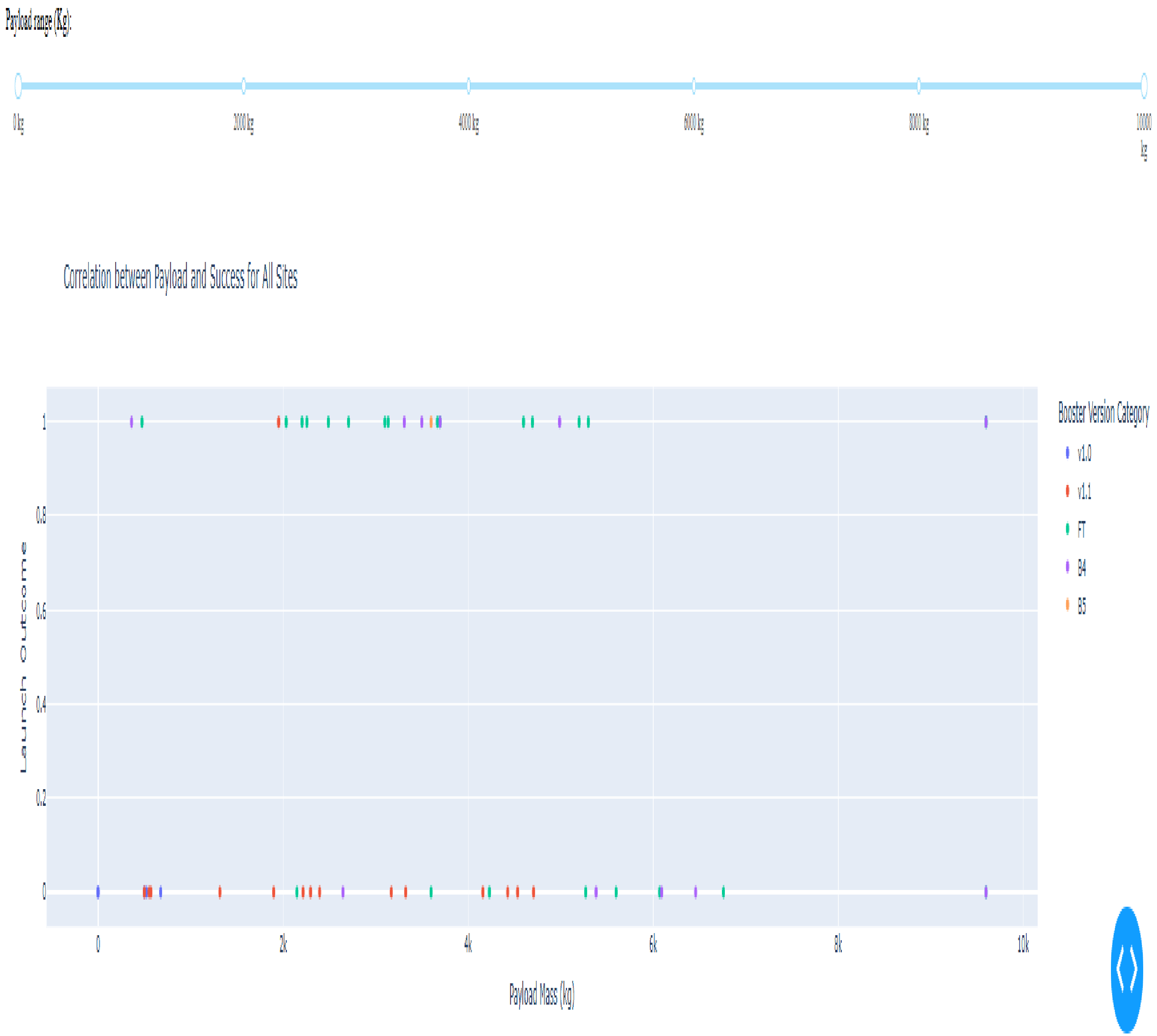
Highest Success Lauch ratio

- This pie chart illustrates the proportion of successful and failed launches at the selected launch site:
- **Blue (76.9%):** Successful launches.
- **Red (23.1%):** Failed launches.
- It highlights that the majority of launches at this site were successful.



<Dashboard Screenshot 3>

- This scatter plot visualizes the correlation between payload mass (in kg) and launch outcomes across all launch sites. Here’s a brief breakdown:
- The **x-axis** represents the payload mass (kg).
- The **y-axis** shows the launch outcome (0 for failure and 1 for success).
- Dots represent individual launches, color-coded by booster version categories.

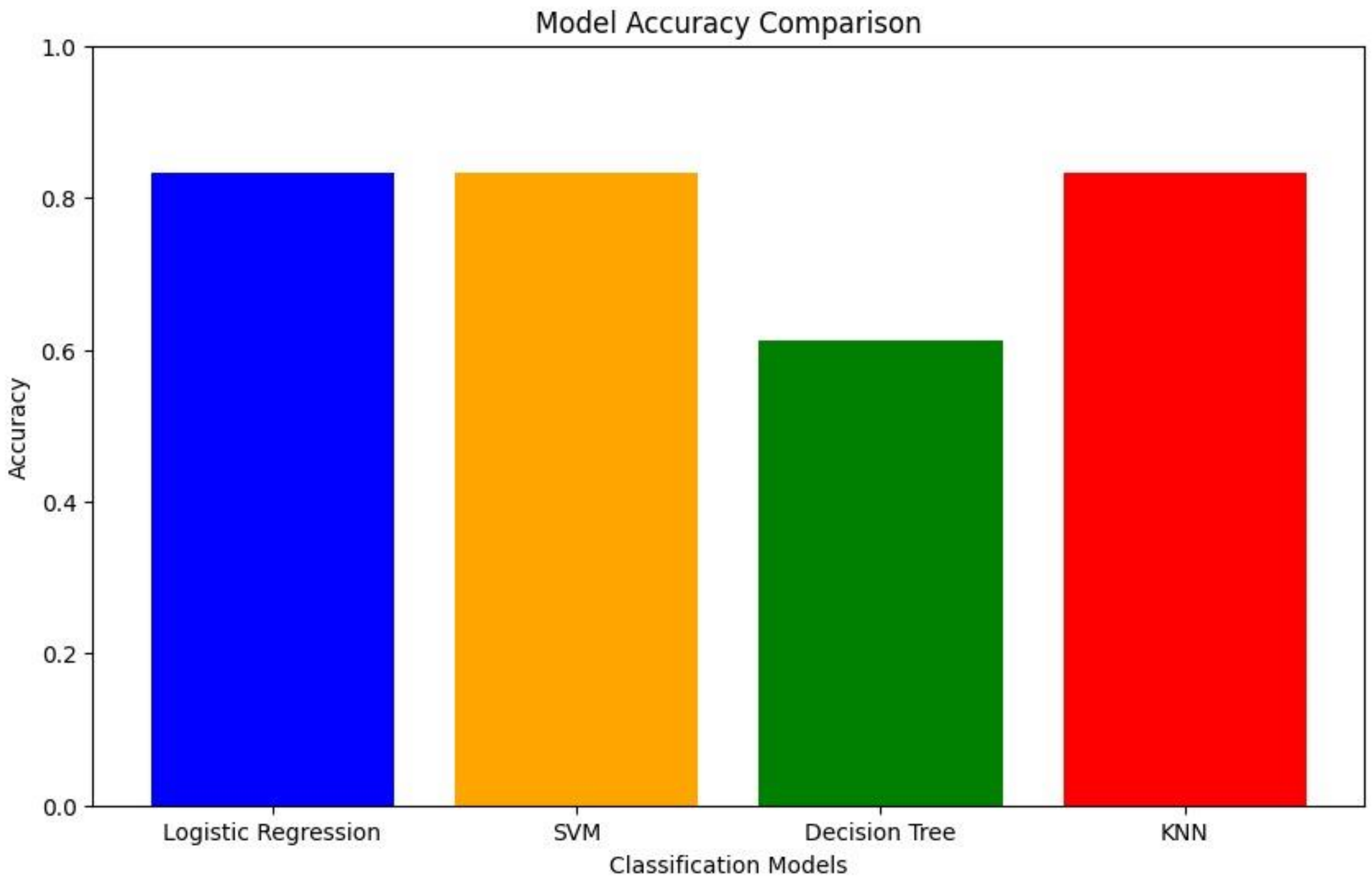


Section 5

Predictive Analysis (Classification)

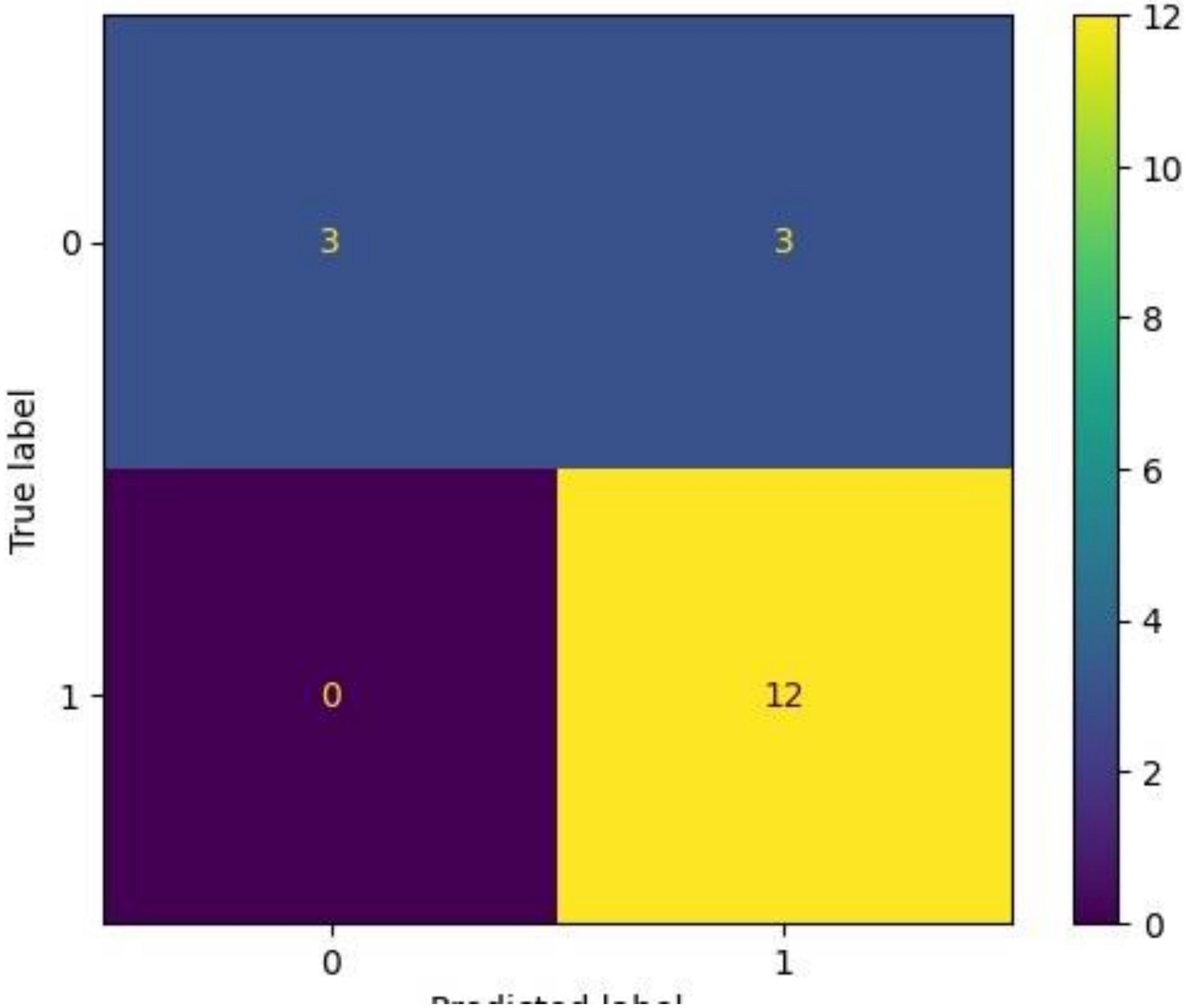
Classification Accuracy

- SUPPORT VECTOR MACHINE HAS THE HIGHEST ACCURACY



Confusion Matrix - SVM

- This is a **confusion matrix** used to evaluate the performance of a classification model. Here's a brief summary of its interpretation:
- **True Positives (1, 1)**: 12 — Correctly predicted as "1" (success).
- **True Negatives (0, 0)**: 3 — Correctly predicted as "0" (failure).
- **False Positives (0, 1)**: 3 — Incorrectly predicted as "1" when the actual value is "0".
- **False Negatives (1, 0)**: 0 — Incorrectly predicted as "0" when the actual value is "1".



Conclusions

Data Exploration and Visualization:
You successfully explored SpaceX launch data by visualizing trends, including success rates across different launch sites, orbit types, and payload masses. Interactive dashboards and correlation plots provided insights into the relationships between payload and launch outcomes.

SQL Queries for Data Analysis:
SQL queries were used to extract meaningful insights, such as the total payloads carried, successful and failed mission outcomes, and identifying boosters with specific landing outcomes. Subqueries and aggregations were effectively applied to refine results.

Model Development and Evaluation:
A classification model was built and evaluated using a confusion matrix. The model showed good predictive performance with a high success rate and minimal errors, as indicated by the distribution of true and false predictions.

Interactive Mapping and Dashboards:
You used

Appendix

- Relevant Assets from the ProjectPython Code Snippets:Collected SpaceX data using REST API with Python's requests library.Cleaned and prepared data using pandas for analysis.Visualized data with matplotlib, plotly, and seaborn.Built a machine learning model (logistic regression) using scikit-learn for launch success predictions.SQL Queries:Queried databases to extract relevant records such as payload mass, booster versions, and mission outcomes.Summarized mission outcomes by computing average payload mass and total successful launches.Charts and Graphs:Pie charts: Showed the proportion of successes vs failures at each launch site.Scatter plots: Highlighted the relationship between payload mass and success.Confusion matrix: Evaluated the model’s performance visually.Notebook Outputs:Interactive maps: Visualized launch site locations using Folium.Model results: Displayed classification reports and accuracy scores for predictive analysis.Datasets:Cleaned and transformed datasets exported for reproducibility.Filtered subsets by launch site, date range, and payload mass.

Thank you!