IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Tianyang Wang
https://github.com/williamwang-ty/ibm-ds-capstone
27/05/2025

# Outline

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

# Executive Summary

- Collected data from the public SpaceX API and the SpaceX Wikipedia page. Created a label column named 'class' to categorize successful landings. Explored the data using SQL, visualizations, Folium maps, and dashboards. Selected relevant columns to be used as features. Converted all categorical variables to binary using one-hot encoding. Standardized the data and utilized GridSearchCV to identify the best parameters for machine learning models. Visualized the accuracy scores of all models.
- Developed four machine learning models: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All models yielded similar results with an accuracy rate of approximately 83.33%. Each model tended to over-predict successful landings. More data is required to improve model performance and accuracy.

# Introduction

- **Background: The Commercial Space Age**
  The commercial space industry is booming, with SpaceX leading through competitive pricing at $62 million per launch compared to the industry average of $165 million USD. This edge comes from their ability to recover and reuse the first stage of rockets. Space Y seeks to rival SpaceX by adopting similar recovery techniques.

- **Problem: Machine Learning for Stage 1 Recovery**
  Space Y has commissioned us to build a machine learning model to predict the successful recovery of their rocket's first stage, a key step to optimize operations and cut costs in competing with SpaceX.

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 two methods: API requests to the public SpaceX API and web scraping from a table on SpaceX's Wikipedia page.

The next slide presents the API data collection flowchart, followed by the flowchart for the web scraping process.

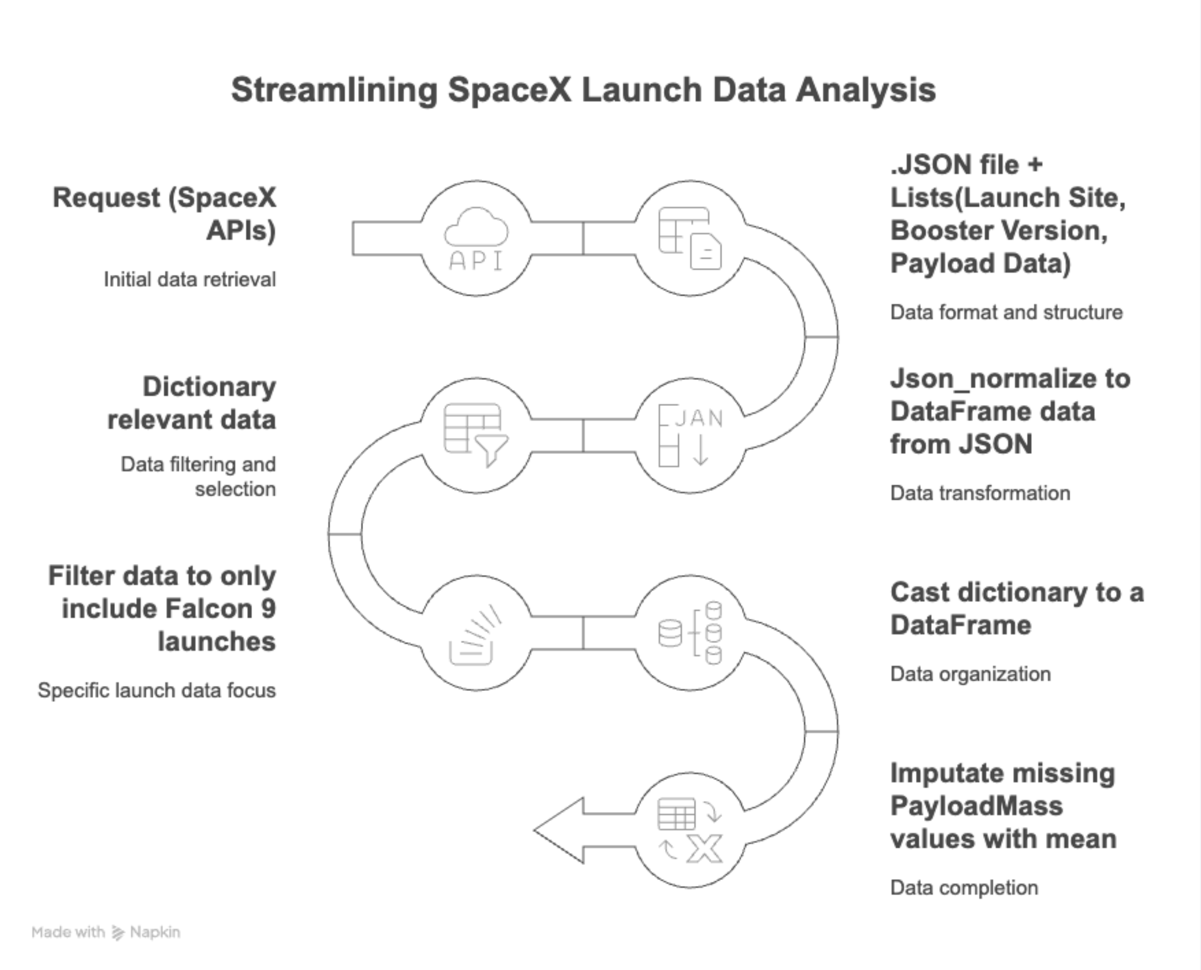
**SpaceX API Data Fields:**
FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

**Wikipedia Web Scraped Data Fields:**
Flight No., Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Booster Version, Booster Landing, Date, Time

# Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- GitHub URL of the completed SpaceX API calls notebook: [https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module1/Data%20Collection%20Api.ipynb](https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module1/Data%20Collection%20Api.ipynb)



**Streamlining SpaceX Launch Data Analysis**

**Request (SpaceX APIs)**
Initial data retrieval

**.JSON file + Lists(Launch Site, Booster Version, Payload Data)**
Data format and structure

**Json_normalize to DataFrame data from JSON**
Data transformation

**Dictionary relevant data**
Data filtering and selection

**Filter data to only include Falcon 9 launches**
Specific launch data focus

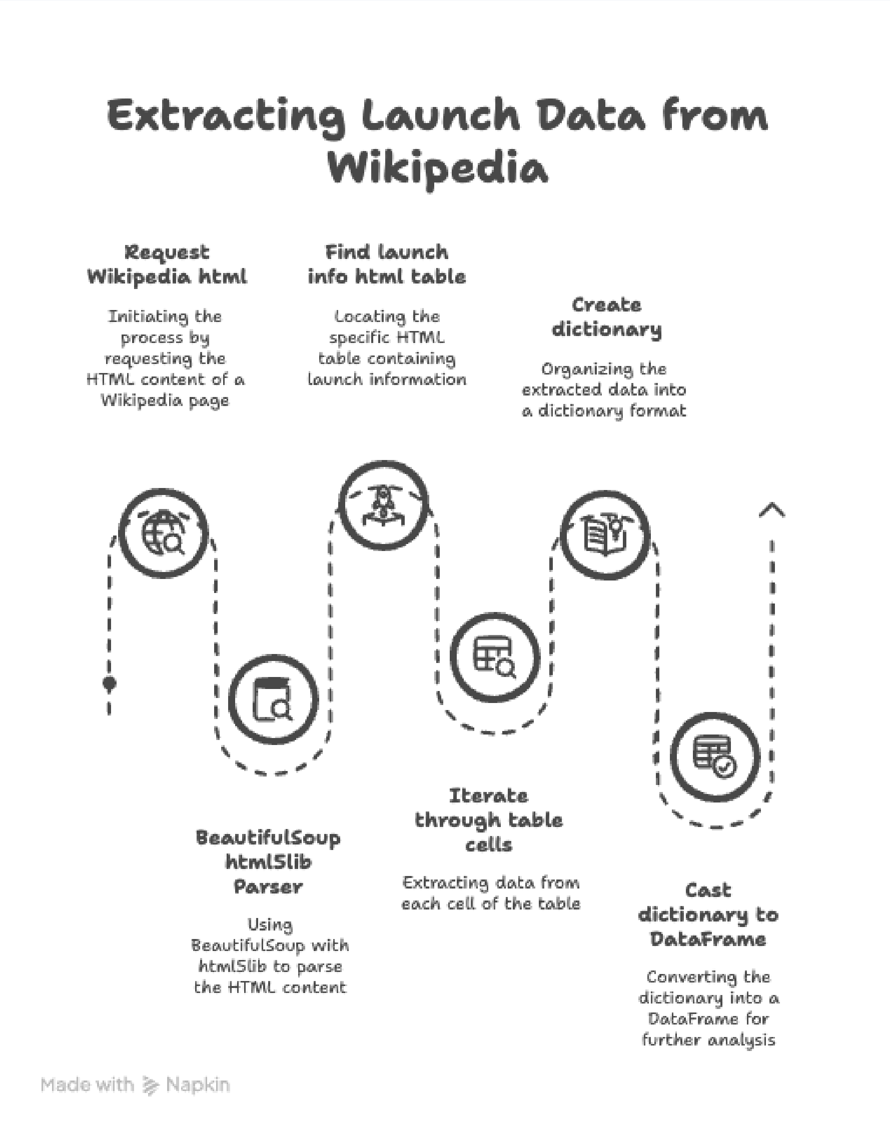
**Cast dictionary to a DataFrame**
Data organization

**Imputate missing PayloadMass values with mean**
Data completion

# Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- GitHub URL of the completed web scraping notebook: [https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module1/Data%20Collection%20with%20Web%20Scraping.ipynb](https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module1/Data%20Collection%20with%20Web%20Scraping.ipynb)



## Extracting Launch Data from Wikipedia

**Request Wikipedia html**

Initiating the process by requesting the HTML content of a Wikipedia page

**BeautifulSoup html5lib Parser**

Using BeautifulSoup with html5lib to parse the HTML content

**Find launch info html table**

Locating the specific HTML table containing launch information

**Iterate through table cells**

Extracting data from each cell of the table

**Create dictionary**

Organizing the extracted data into a dictionary format

**Cast dictionary to DataFrame**

Converting the dictionary into a DataFrame for further analysis

# Data Wrangling

Create a new training label column called **class**, where **successful landings = 1** and **failures = 0**.

The **Outcome** column includes both *Mission Outcome* and *Landing Location*.

**Label Mapping:**

- Assign **1** to: True ASDS, True RTLS, True Ocean
- Assign **0** to: None None, False ASDS, None ASDS, False Ocean, False RTLS

- GitHub URL:
- https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module1/Data%20wrangling.ipynb

# EDA with Data Visualization

- Exploratory Data Analysis (EDA) was performed on the following variables: **Flight Number, Payload Mass, Launch Site, Orbit, Class,** and **Year**.

- **Plots Used:**
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload Mass vs. Launch Site
  - Orbit vs. Success Rate
  - Flight Number vs. Orbit
  - Payload vs. Orbit
  - Yearly Success Trend

- Scatter plots, line charts, and bar plots were used to explore relationships between variables for potential use in the machine learning model.

- GitHub URL:
- https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module2/EDA%20with%20Visualization.ipynb

# EDA with SQL

- Loaded the dataset into an IBM DB2 database and queried it using SQL through Python.

- Queries were used to explore:
  - Launch site names
  - Mission outcomes
  - Customer payload sizes
  - Booster versions
  - Landing outcomes

- GitHub URL
  - https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module2/EDA%20with%20SQL.ipynb

# Build an Interactive Map with Folium

- Folium maps show launch sites, landing outcomes (success/failure), and proximity to key locations: **railways, highways, coasts,** and **cities**.

- This helps explain launch site placement and visualizes landing success in relation to location.

- Github URL：
  - https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module3/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

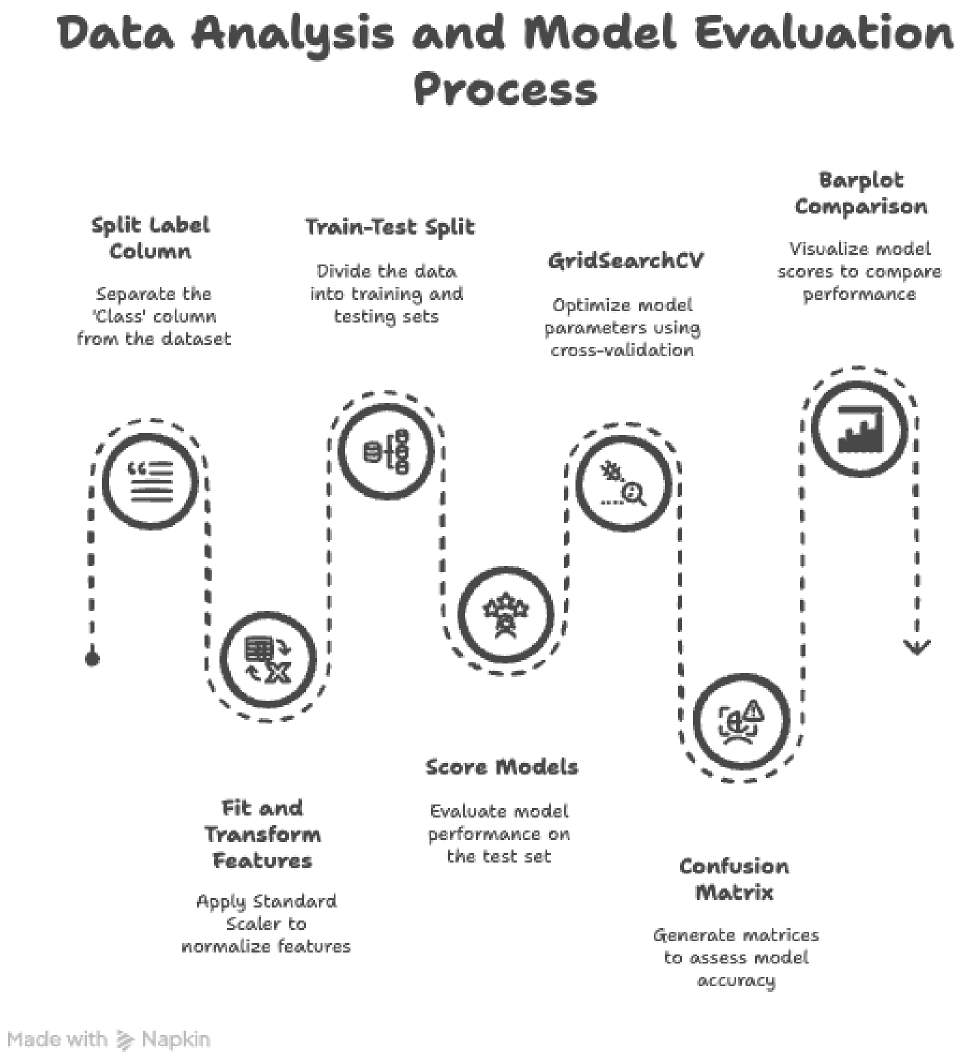
# Build a Dashboard with Plotly Dash

- The dashboard includes a **pie chart** and a **scatter plot**.
  - The **pie chart** shows the distribution of successful landings across all launch sites or by individual site.
  - The **scatter plot** filters by site (all or individual) and payload mass (0–10,000 kg slider).
  - The pie chart visualizes launch success rates, while the scatter plot reveals how success varies by launch site, payload mass, and booster version.

- GitHub URL:
  - https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module3/spacex_dash_app.py

# Predictive Analysis (Classification)

- GitHub URL
  - [https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module 4/Machine%20Learning%20Prediction.ipynb](https://github.com/williamwang-ty/ibm-ds-capstone/blob/main/Module4/Machine%20Learning%20Prediction.ipynb)



## Data Analysis and Model Evaluation Process

**Split Label Column**

Separate the 'Class' column from the dataset

**Fit and Transform Features**

Apply Standard Scaler to normalize features

**Train-Test Split**

Divide the data into training and testing sets

**Score Models**

Evaluate model performance on the test set

**GridSearchCV**

Optimize model parameters using cross-validation

**Confusion Matrix**

Generate matrices to assess model accuracy

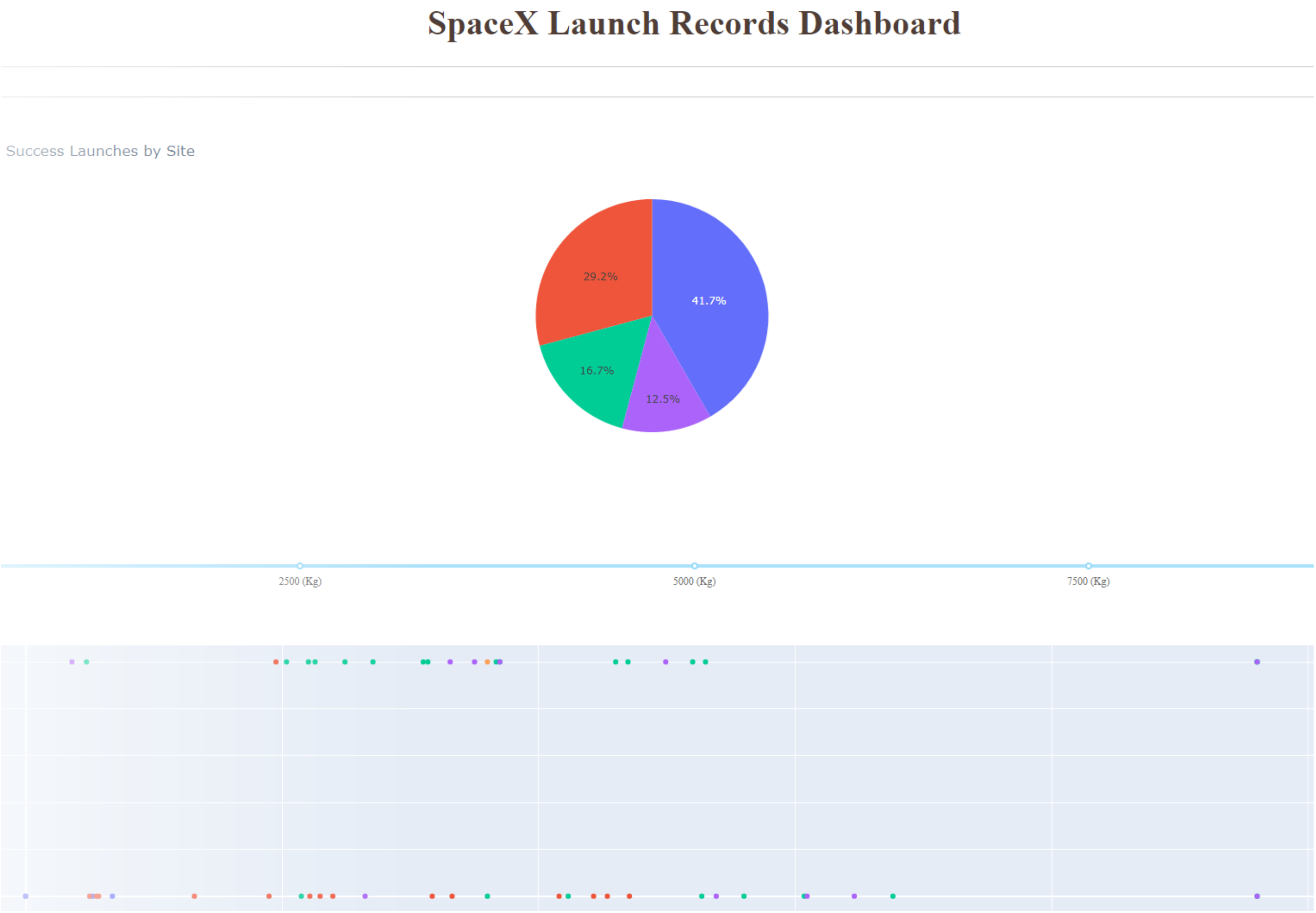
**Barplot Comparison**

Visualize model scores to compare performance

Made with Napkin

# Results

- This is a preview of the Plotly dashboard.
- The next slides will cover:
    - EDA with visualizations
    - EDA using SQL
    - Interactive Folium map
    - Model results with ~83% accuracy

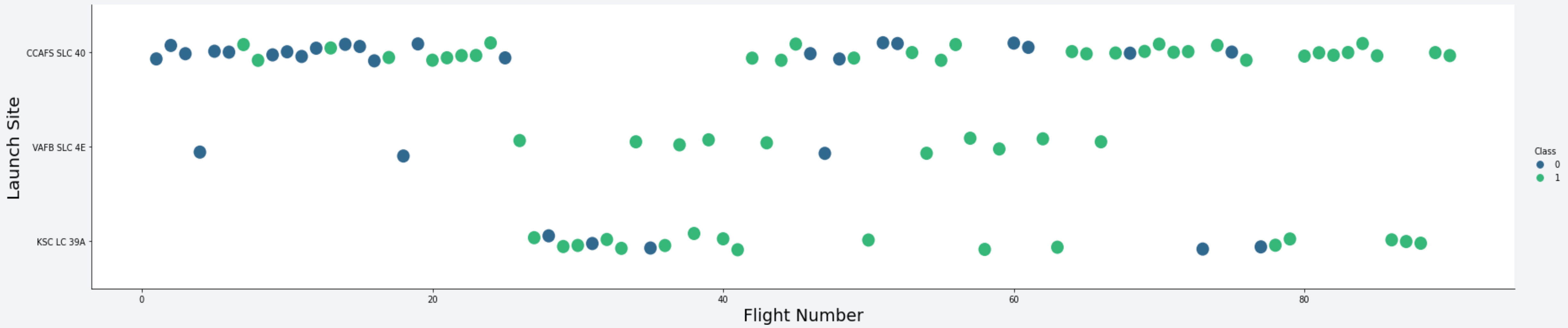


SpaceX Launch Records Dashboard

Section 2

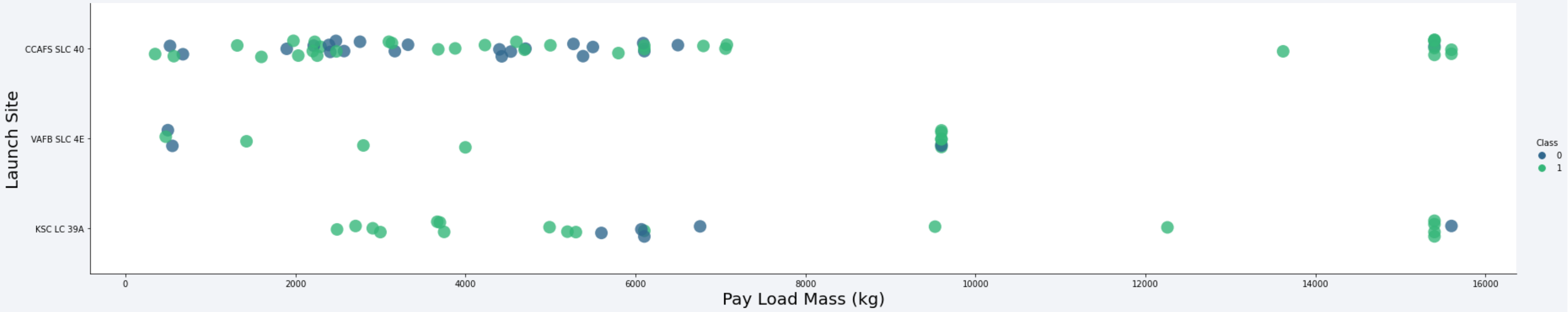
# Insights drawn from EDA

# Flight Number vs. Launch Site



- Green indicates successful launch; Purple indicates unsuccessful launch.
- The graph shows a rising success rate over time, with a major improvement around Flight 20.
- **CCAFS** is the primary launch site, handling the highest number of launches.

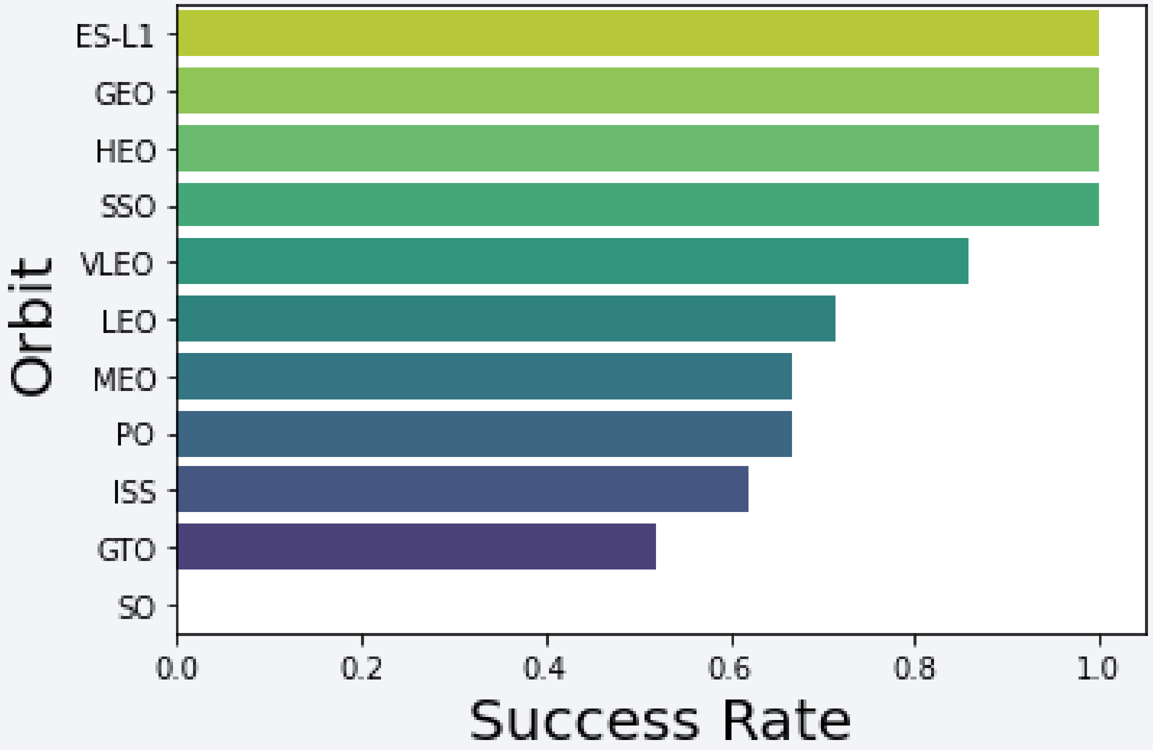
# Payload vs. Launch Site



Green indicates successful launch; Purple indicates unsuccessful launch.

Most payloads range between **0–6000 kg**, with variations across different launch sites.

# Success Rate vs. Orbit Type



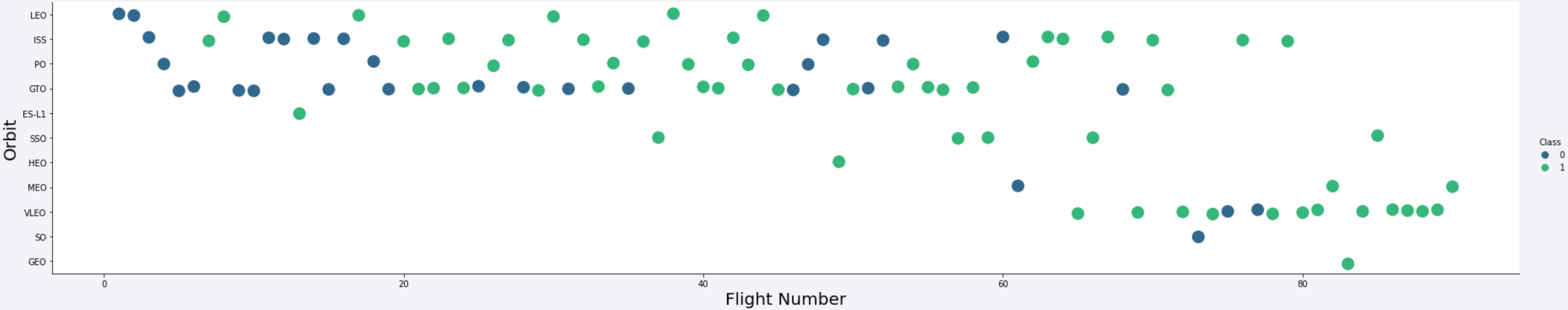
**ES-L1, GEO, HEO (1 each)** and **SSO (5)** have a **100% success rate**.

**VLEO (14)** shows good success with more attempts.

**SO (1)** has a **0% success rate**.

**GTO (27)** has the **largest sample** with about **50% success**.

# Flight Number vs. Orbit Type

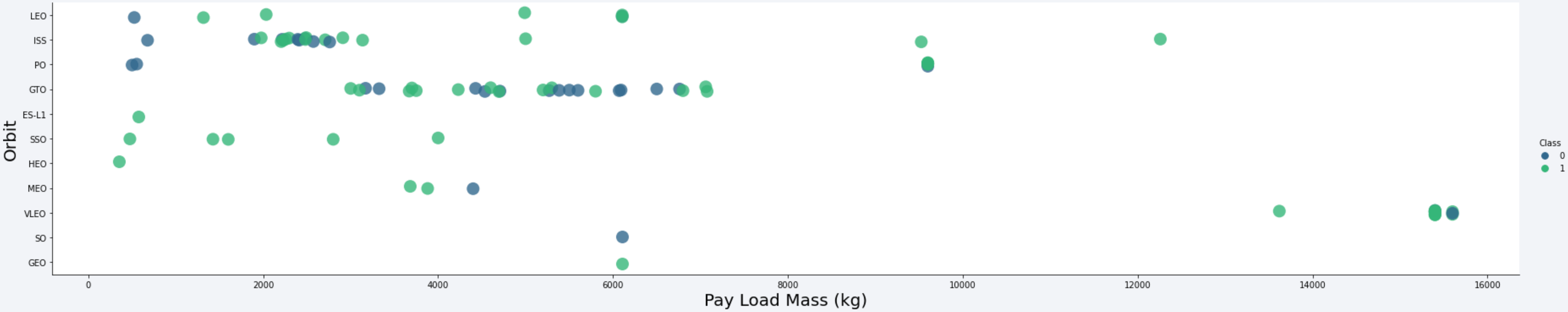


SpaceX has shifted its orbital preferences over time, with launch outcomes correlating to these changes.

The company initially favored LEO orbits with moderate success, then moved to various orbital types, and recently returned to VLEO launches.

Performance data suggests SpaceX achieves better results in lower orbits and Sun-synchronous orbits.

# Payload vs. Orbit Type



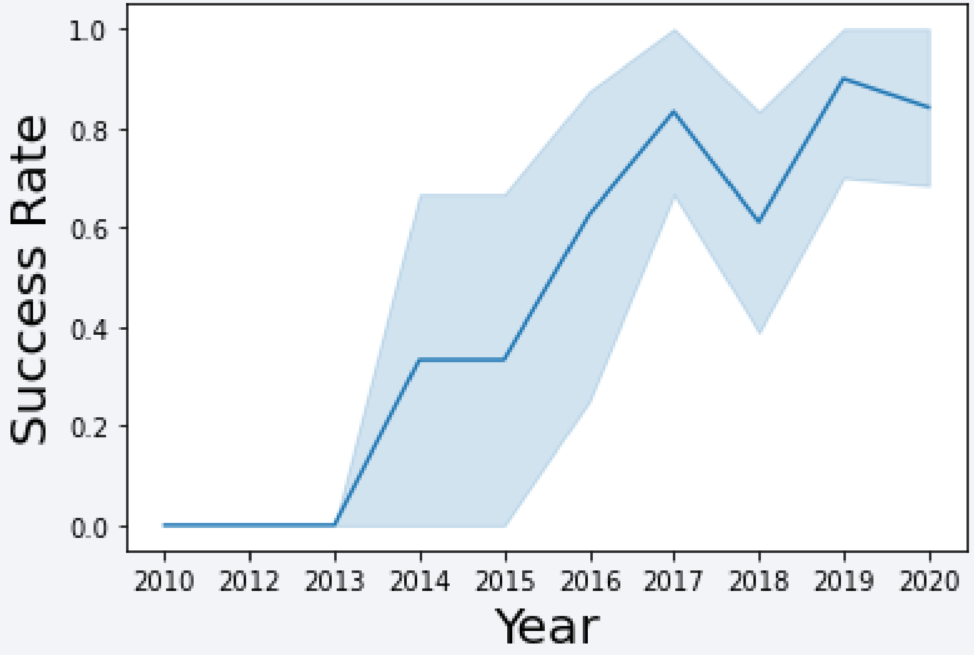
**Payload Mass and Orbit Correlation:**

LEO and SSO missions typically carry lighter payloads with relatively low mass values

VLEO missions, despite being highly successful, exclusively handle heavier payloads in the upper mass range

Clear correlation exists between orbital destination and payload mass requirements

# Launch Success Yearly Trend

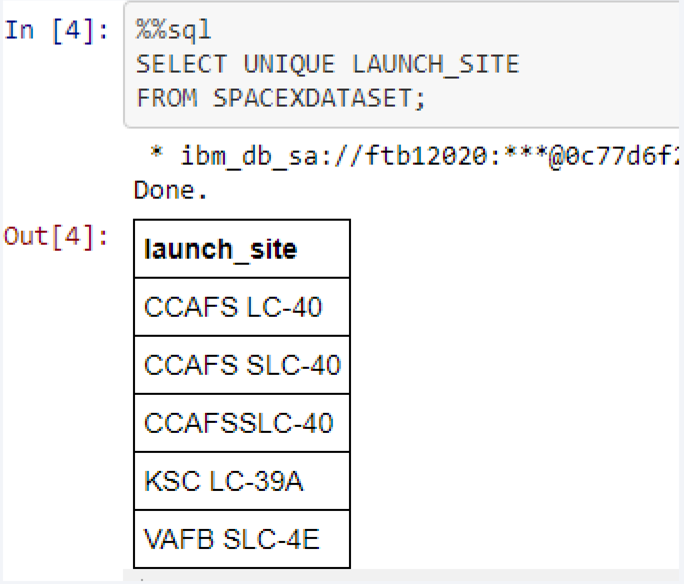


**Launch Success Trend:**

Success rates have consistently improved since 2013, with only a minor decline in 2018

Recent years show approximately 80% success rate, indicating strong operational maturity

# All Launch Site Names



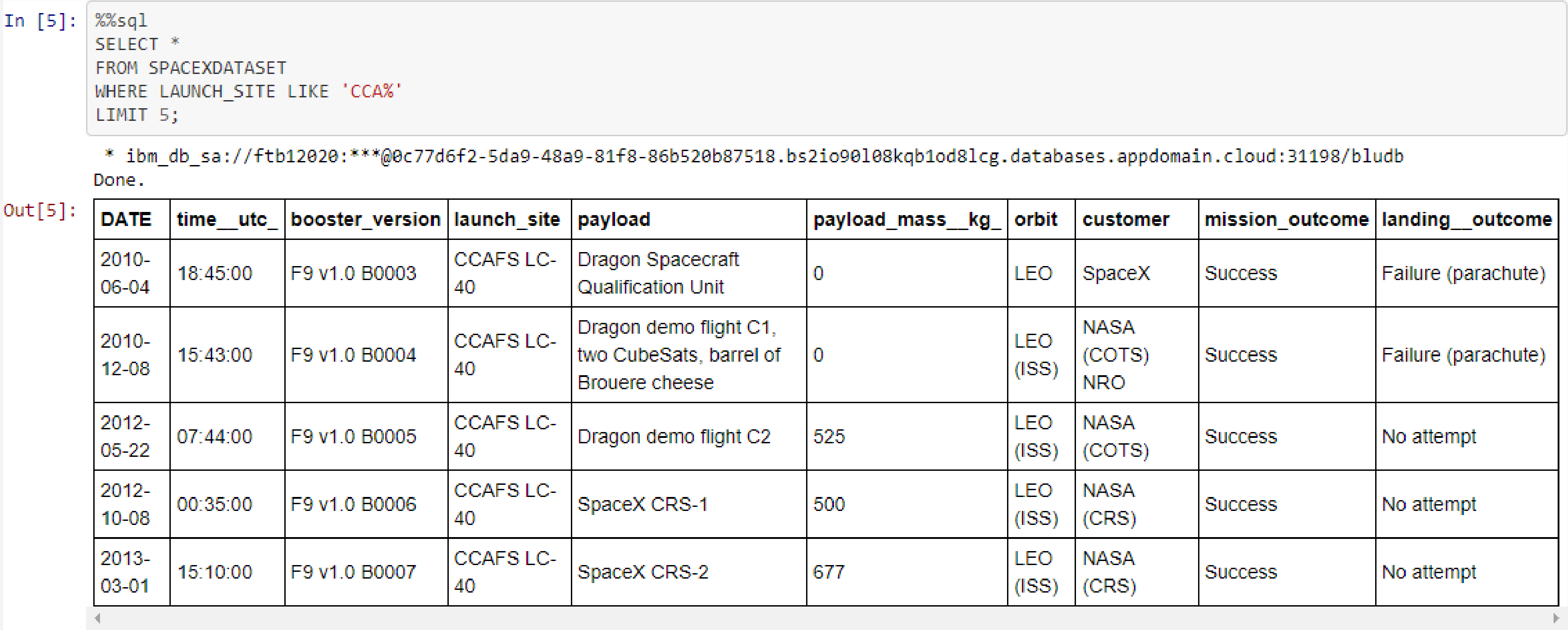
```
In [4]: %%sql
        SELECT UNIQUE LAUNCH_SITE
        FROM SPACEXDATASET;

         * ibm_db_sa://ftb12020:***@0c77d6f
        Done.
Out[4]:
```

| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS SLC-40 |
| CCAFSSLC-40 |
| KSC LC-39A |
| VAFB SLC-4E |

- Database contains inconsistent naming for the same launch sites due to data entry errors
- CCAFS SLC-40, CCAFSSLC-40, and CCAFS LC-40 all represent the same Cape Canaveral launch facility
- Only 3 unique launch sites exist: CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E
- Data standardization needed to consolidate duplicate entries under consistent naming convention

# Launch Site Names Begin with 'CCA'



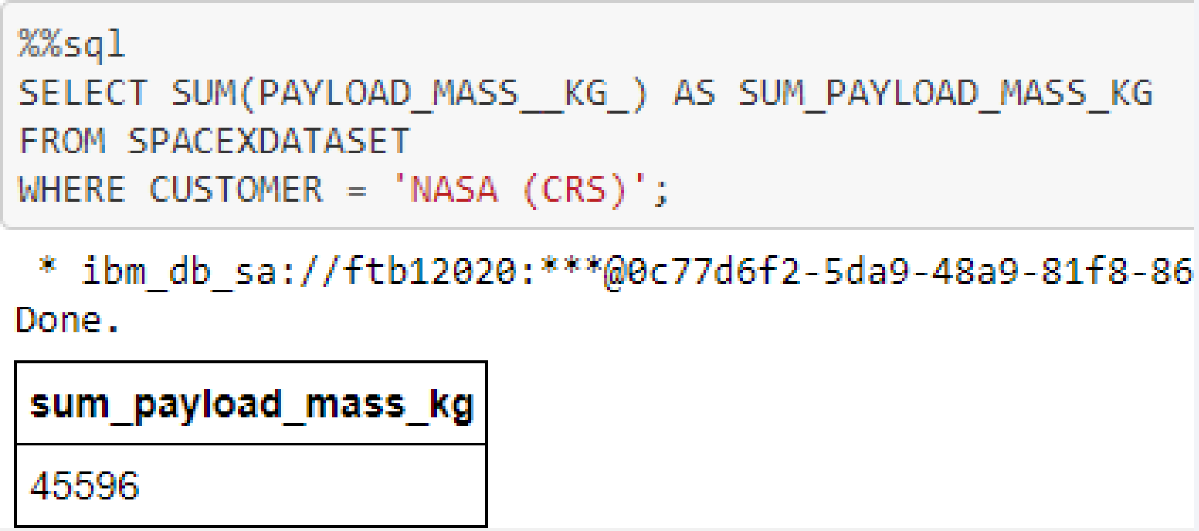
```
In [5]: %%sql
        SELECT *
        FROM SPACEXDATASET
        WHERE LAUNCH_SITE LIKE 'CCA%'
        LIMIT 5;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[5]:

| 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 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 00:35:00 | 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



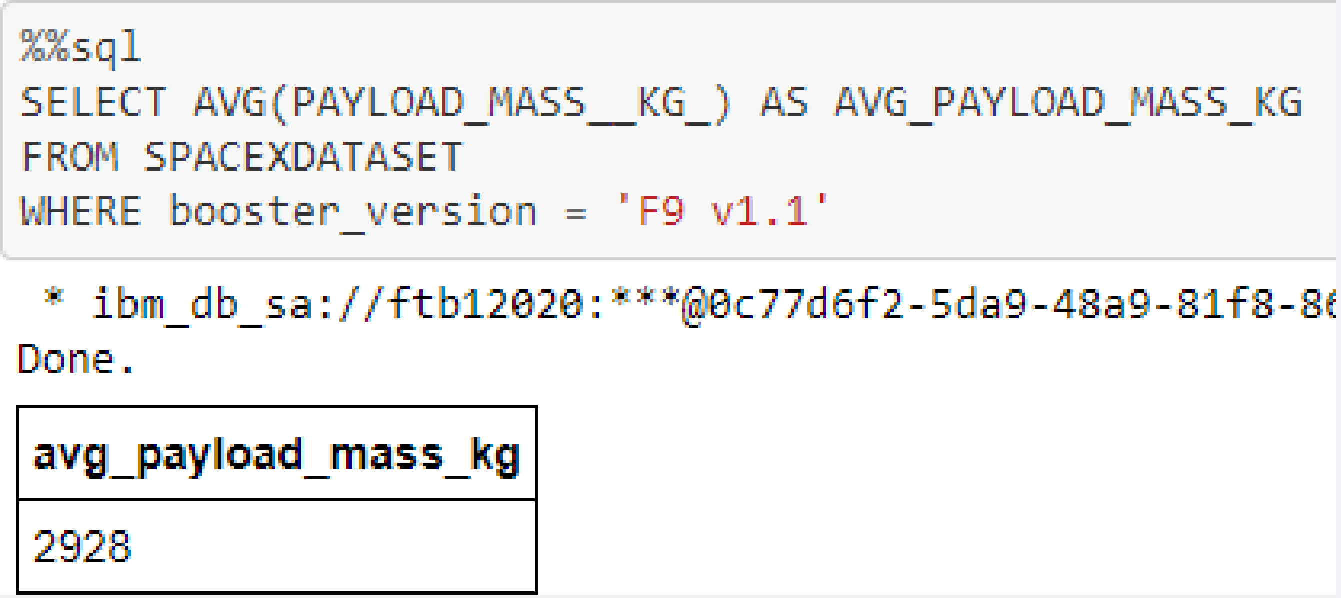
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

| sum_payload_mass_kg |
|---|
| 45596 |

- Query calculates total payload mass in kilograms for all NASA-contracted missions
- CRS (Commercial Resupply Services) missions specifically deliver cargo to the International Space Station
- NASA serves as the primary customer for ISS resupply operations through SpaceX's commercial launch services

# Average Payload Mass by F9 v1.1

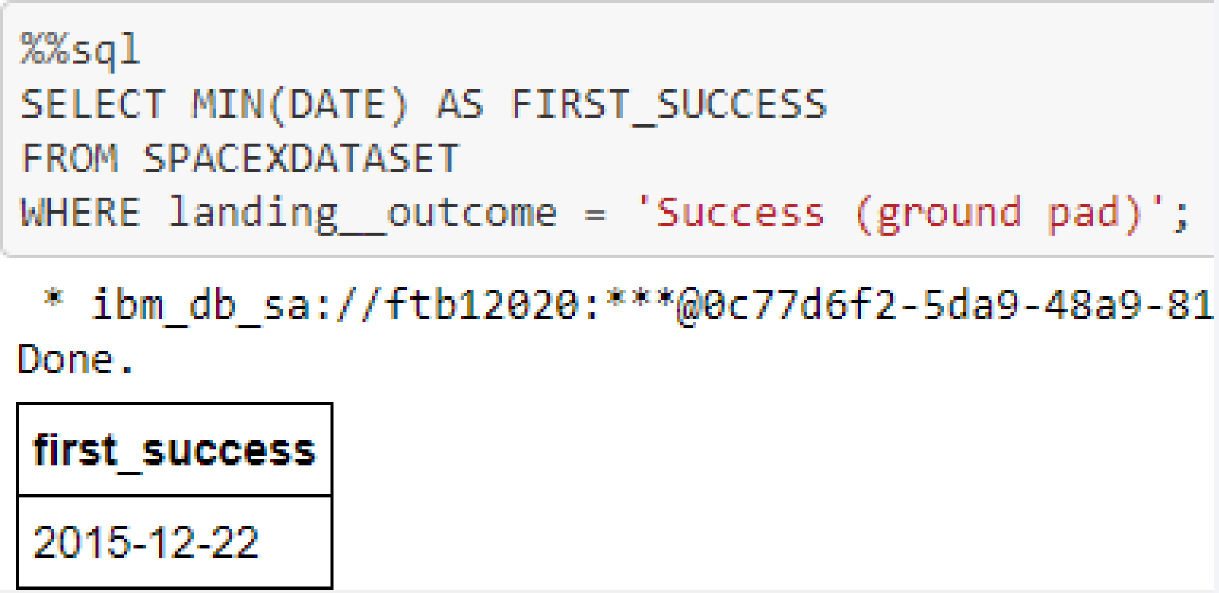


```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

| avg_payload_mass_kg |
|---|
| 2928 |

# First Successful Ground Landing Date



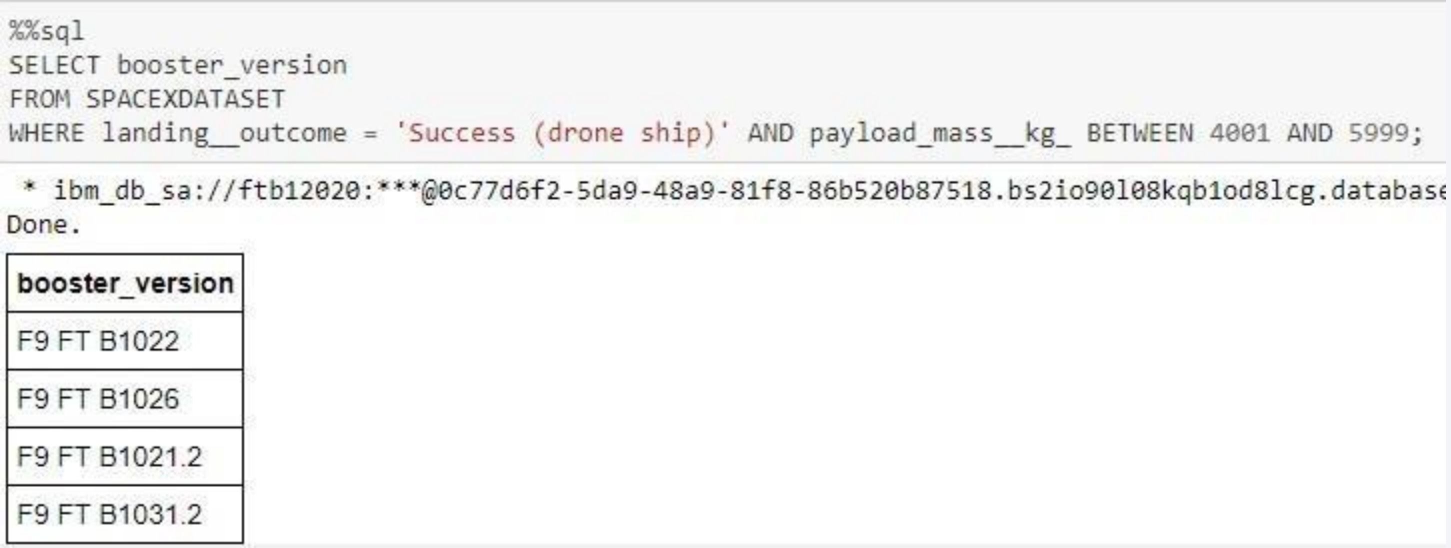
```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

| first_success |
|---|
| 2015-12-22 |

- Query identifies the first successful ground pad landing occurred in late 2015
- Earlier successful landings began in 2014, but these were likely drone ship or other recovery methods
- Ground pad landings represented a significant technological advancement in SpaceX's reusability program

# Successful Drone Ship Landing with Payload between 4000 and 6000



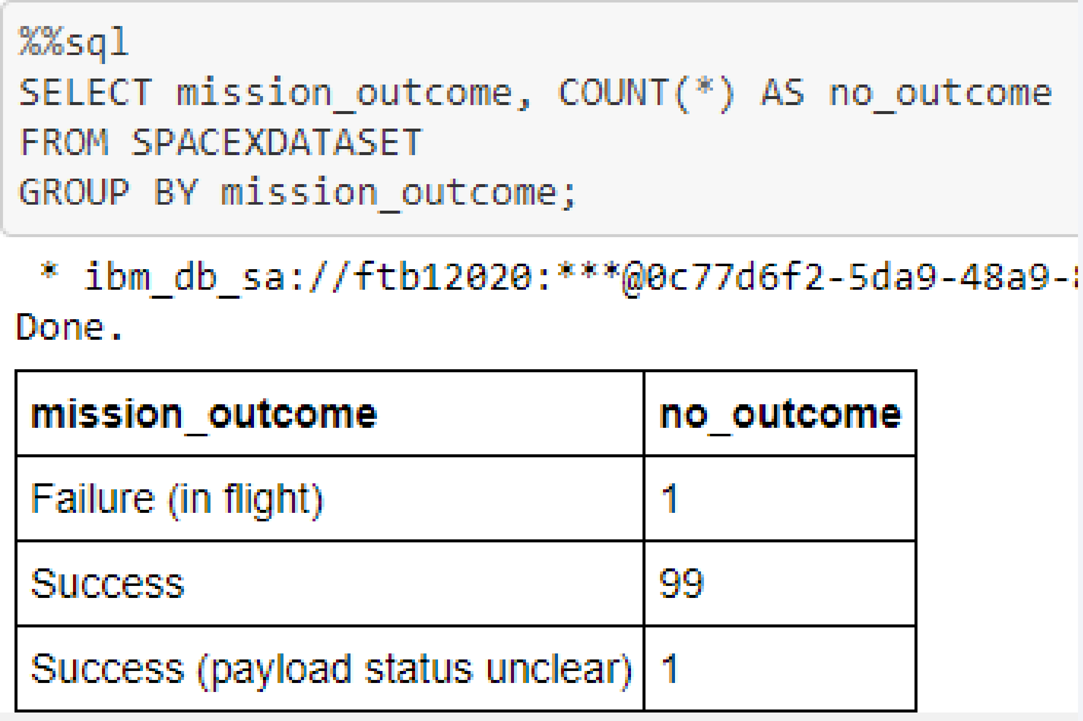
```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
Done.
```

| booster_version |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

- Query identifies four specific booster versions that achieved successful drone ship landings
- These missions carried payloads between 4,000-6,000 kg (exclusive range)
- Demonstrates successful recovery capabilities across multiple booster iterations within a specific payload mass window

# Total Number of Successful and Failure Mission Outcomes



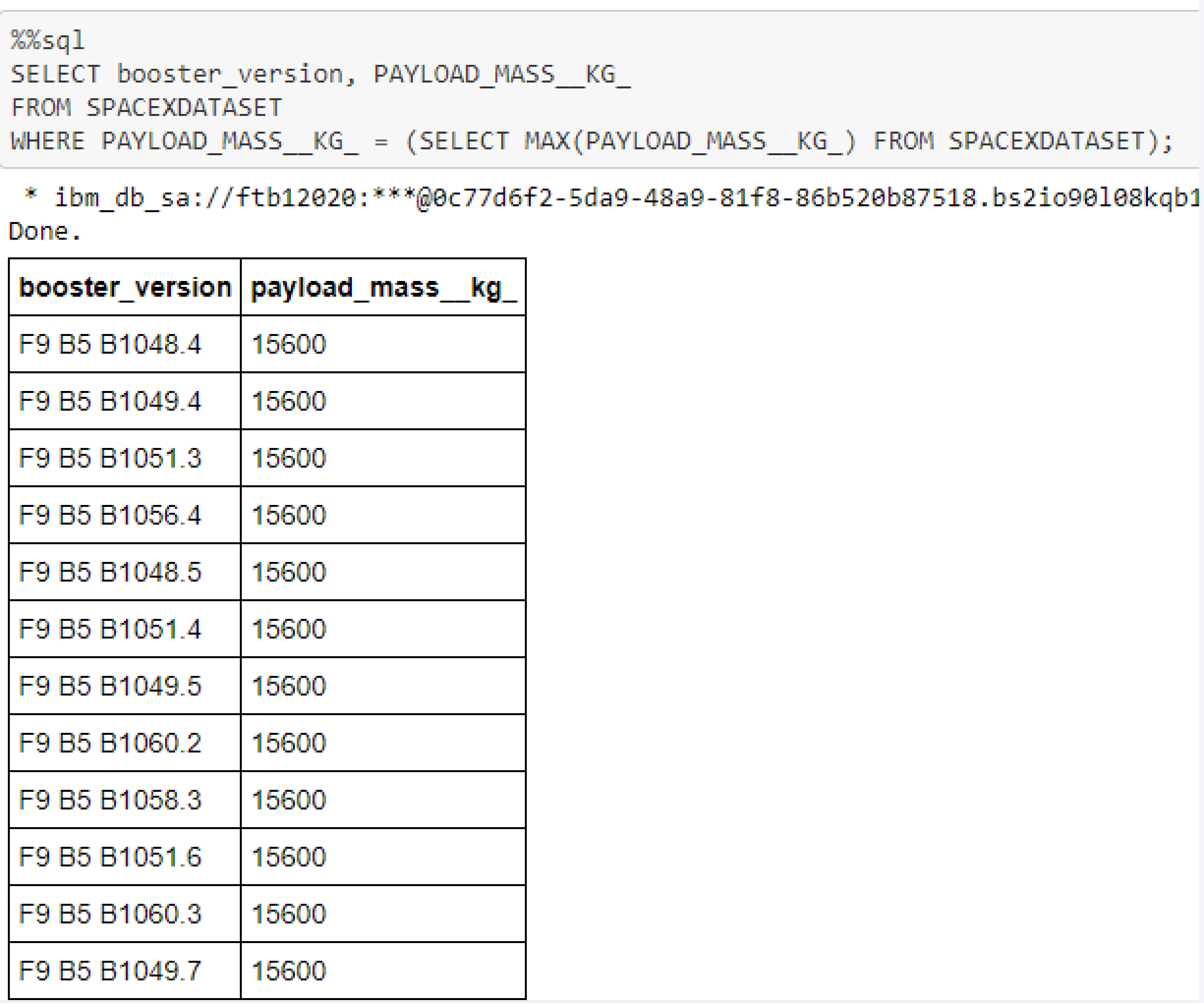
```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.
```

| mission_outcome | no_outcome |
|---|---|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

- Query shows SpaceX achieves mission success nearly 99% of the time across all launches
- Most landing failures appear to be intentional, indicating mission success despite recovery attempts
- Database contains one launch with unclear payload status and one in-flight failure
- High success rate demonstrates SpaceX's operational reliability in primary mission objectives

# Boosters Carried Maximum Payload



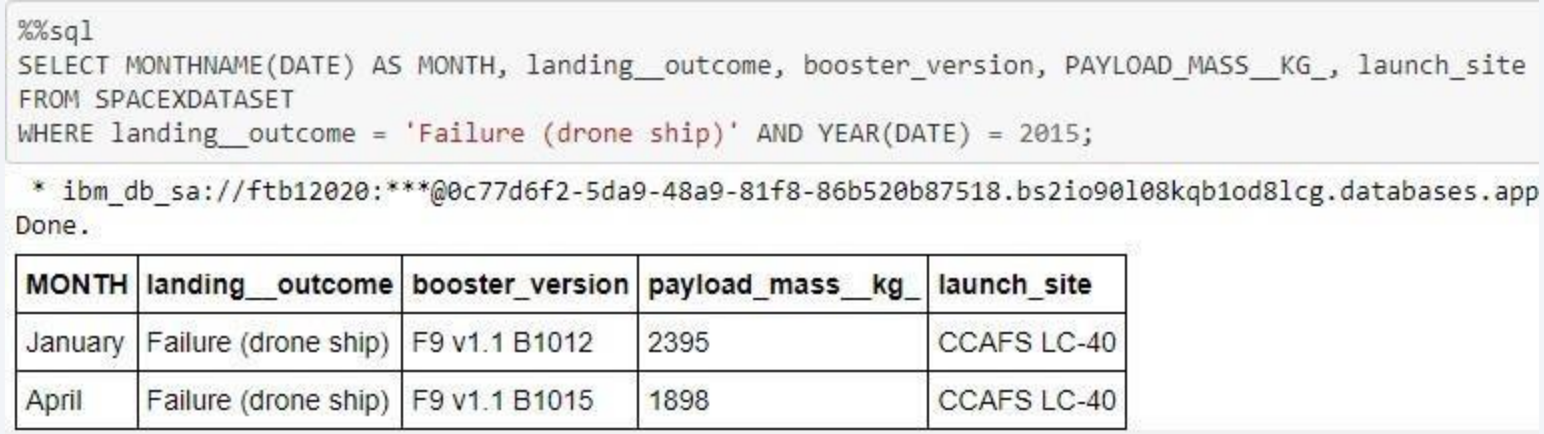
```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
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 |

- Query identifies booster versions that carried the maximum payload mass of 15,600 kg
- All heavy payload missions used F9 Block 5 boosters with B10xx.x designations
- Strong correlation exists between booster version capabilities and payload mass capacity
- Block 5 variants demonstrate superior performance for maximum weight missions

# 2015 Launch Records



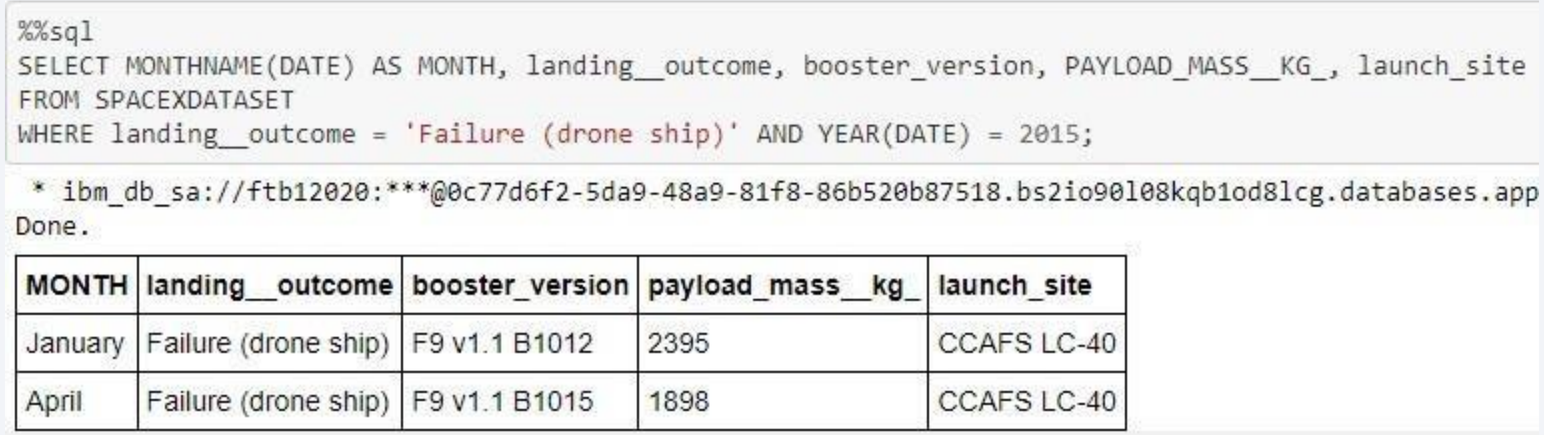
```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

| MONTH | landing__outcome | booster_version | payload_mass__kg_ | launch_site |
|---|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | 2395 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | 1898 | CCAFS LC-40 |

- Query identifies two specific 2015 launches where Stage 1 failed to land successfully on drone ships
- Results include complete mission details: launch month, landing outcome, booster version, payload mass, and launch site
- These failures represent early attempts at autonomous drone ship recovery technology development

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



```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

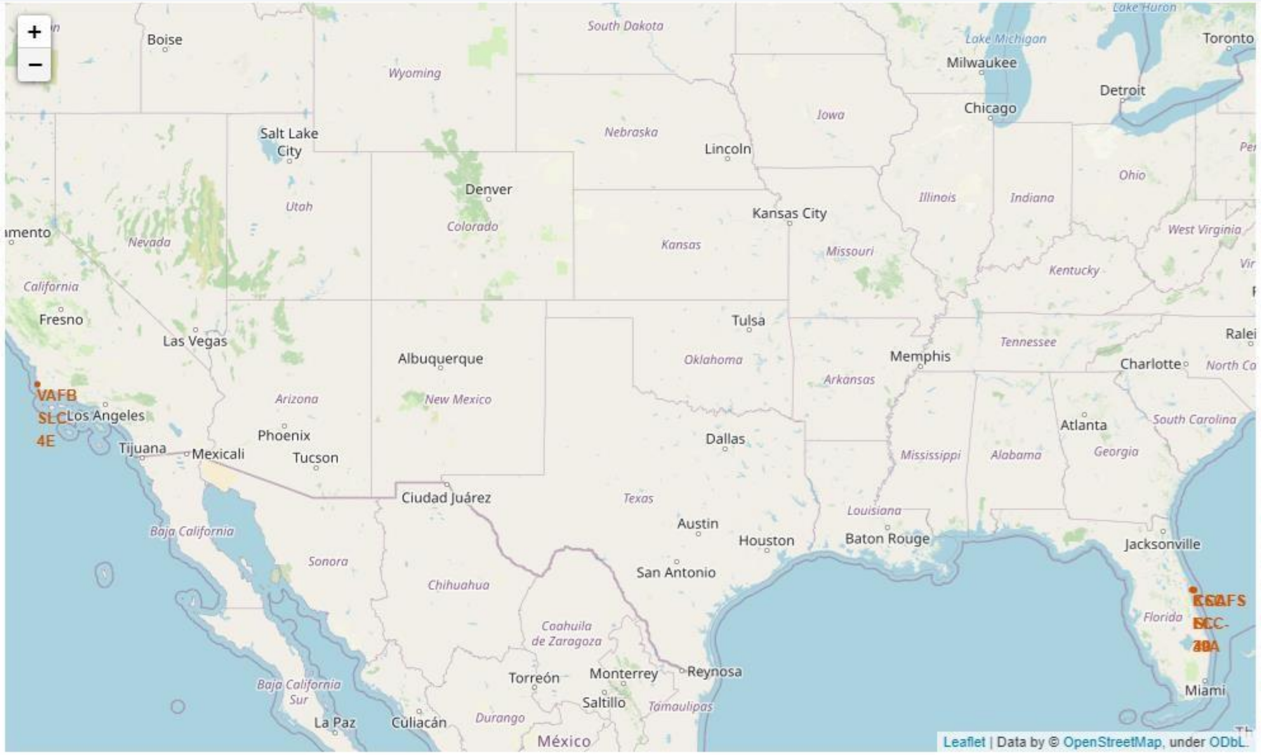
| MONTH | landing__outcome | booster_version | payload_mass__kg_ | launch_site |
|---|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | 2395 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | 1898 | CCAFS LC-40 |

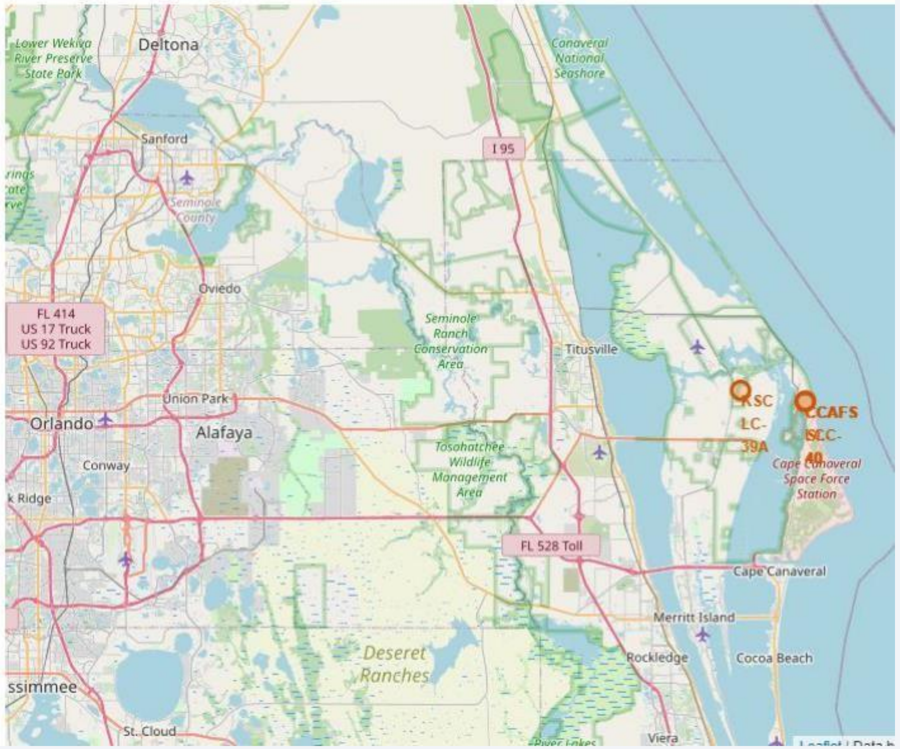
- Query covers successful landings between June 4, 2010 and March 20, 2017 (inclusive)
- Two successful landing types achieved: drone ship landings and ground pad landings
- Total of 8 successful recoveries accomplished during this developmental timeframe
- Represents SpaceX's early mastery of reusable rocket recovery technology

Section 3

# Launch Sites Proximities Analysis

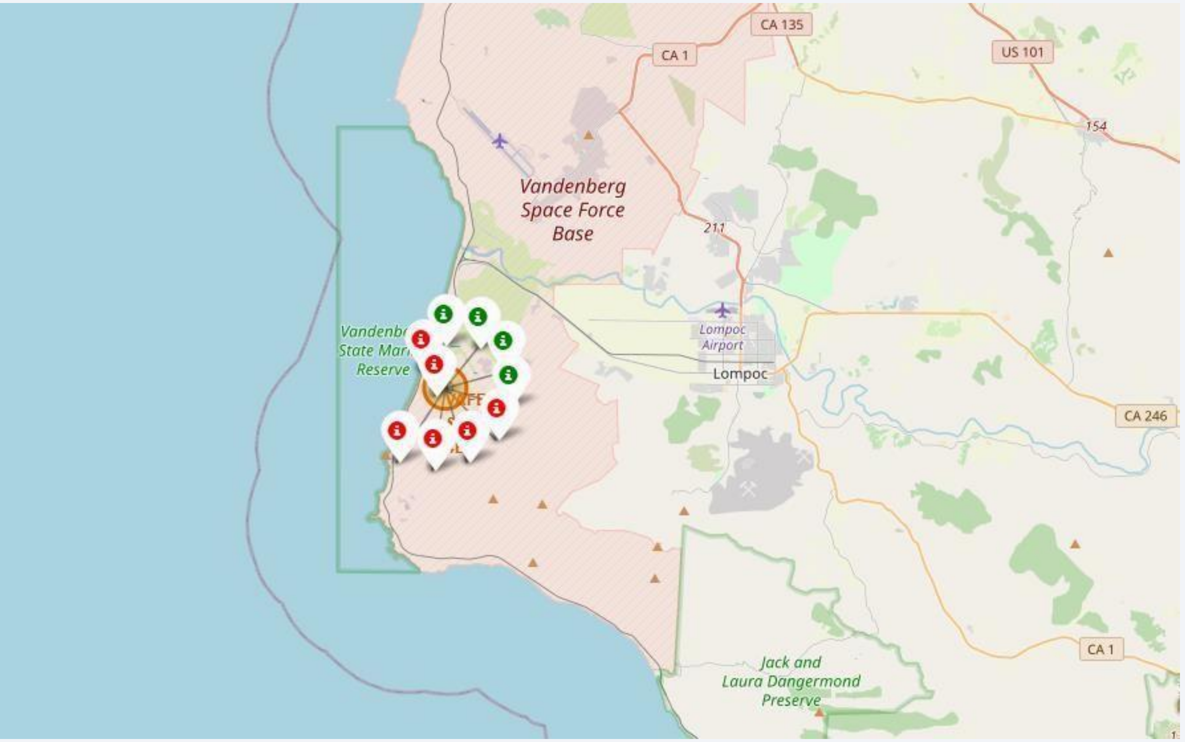
# Lauch Site Locations





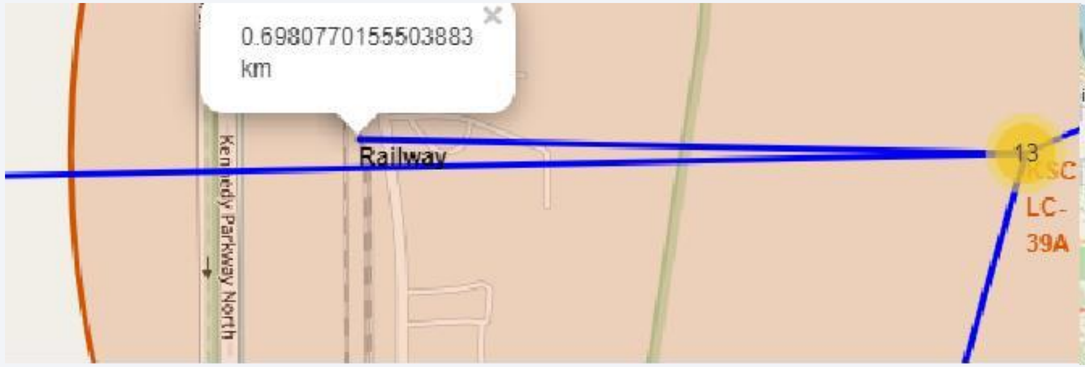
The left map displays all U.S. launch sites relative to the country's outline.
The right map zooms in on Florida's two closely located launch sites.
All sites are coastal.

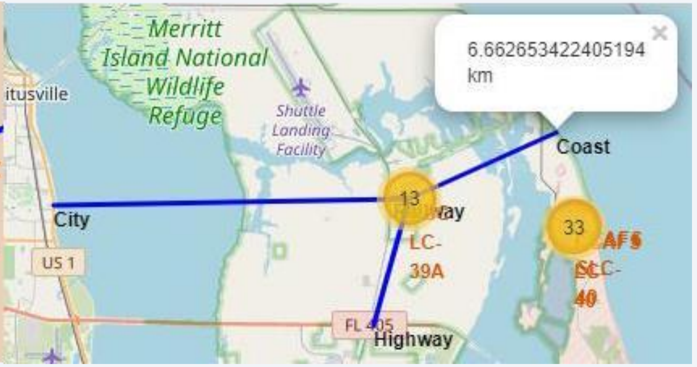
# Launch Markers

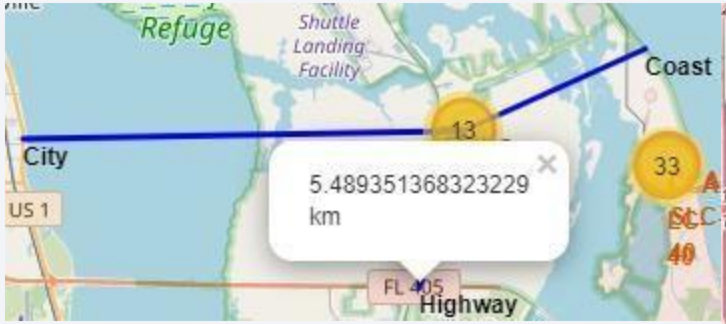


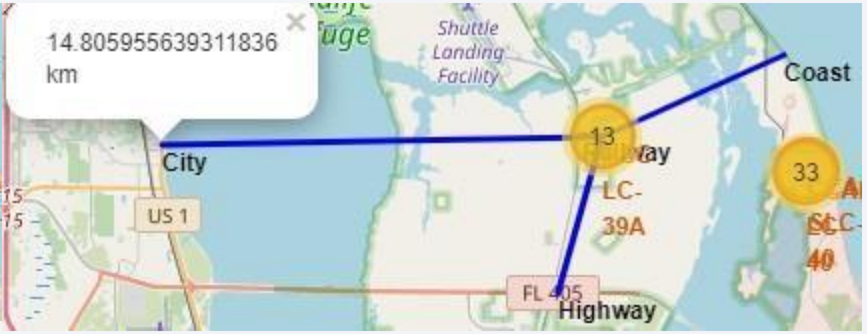
Folium map clusters, when clicked, show successful landings (green icons) and failed landings (red icons).
For example, VAFB SLC-4E displays 4 successful and 6 failed landings.

# Key Location Proximity



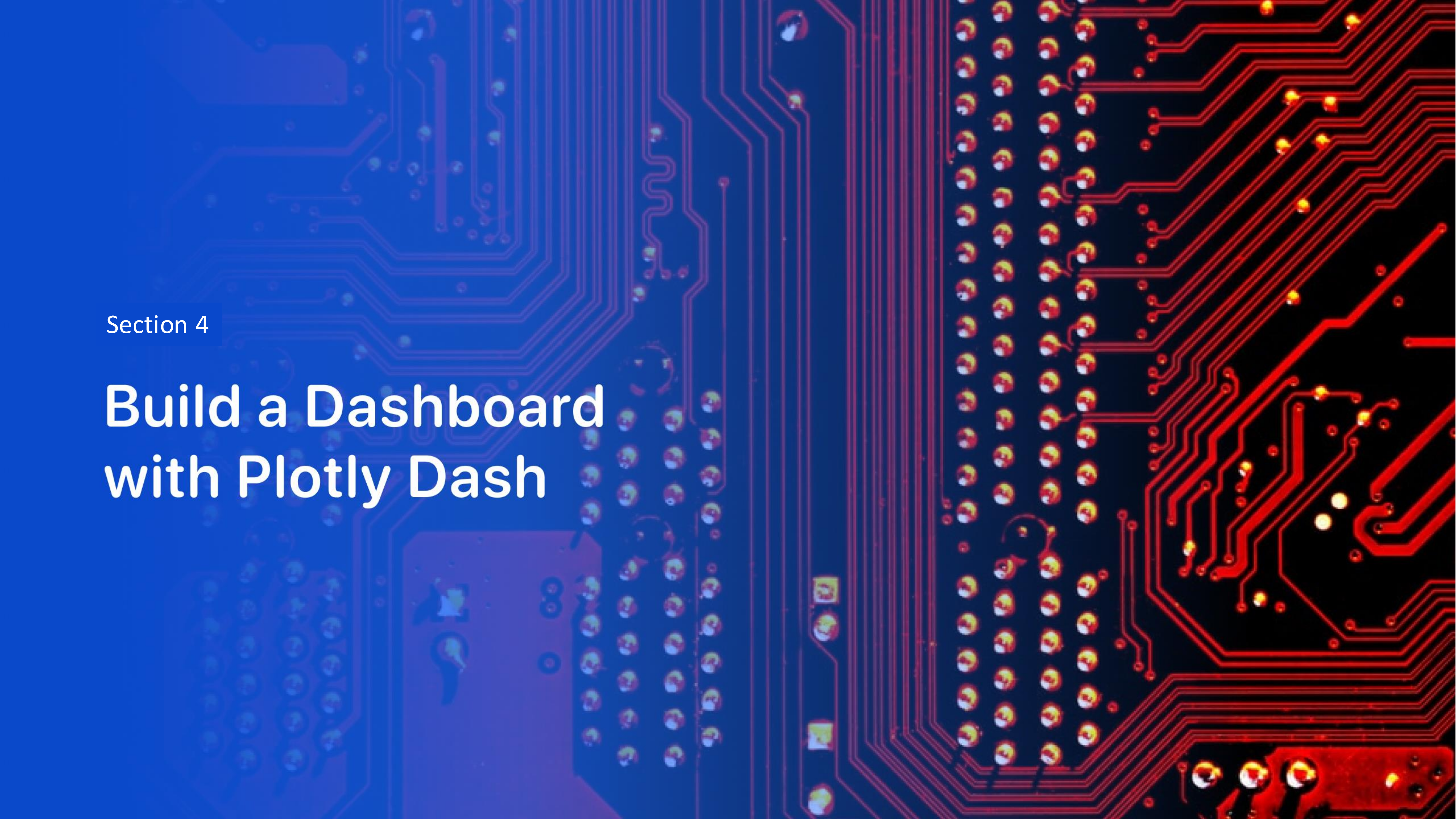






Using KSC LC-39A as an example, launch sites are strategically located near railways and highways for efficient transport of large components and personnel.

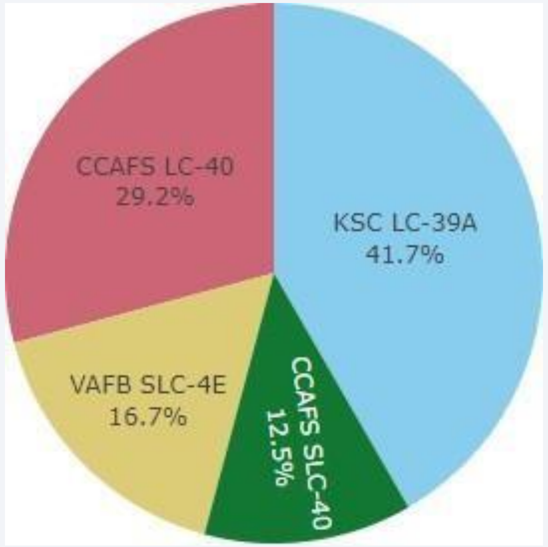
They are also positioned close to coasts and farther from cities, allowing failed launches to land safely in the sea, minimizing risks to populated areas.



Section 4

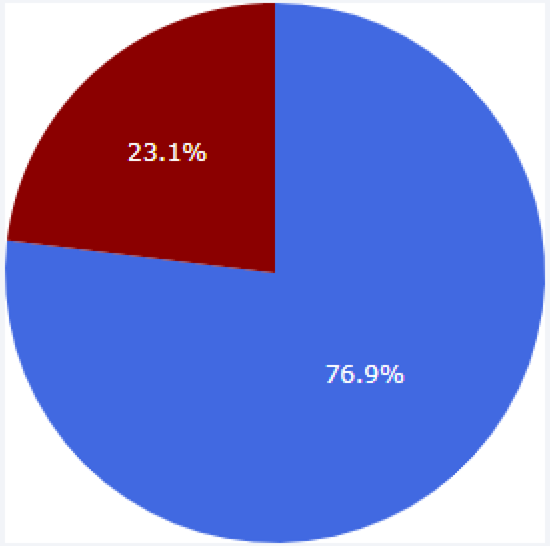
# Build a Dashboard with Plotly Dash

# Successfully Launched Across All Sites



Here's a breakdown of successful landing distribution by launch site:
CCAFS SLC-40 and KSC have an equal number of successful landings.
Most successful landings at CCAFS occurred before its name change from LC-40.
VAFB has the fewest successful landings, possibly due to a smaller sample size and the increased difficulty of west coast launches.

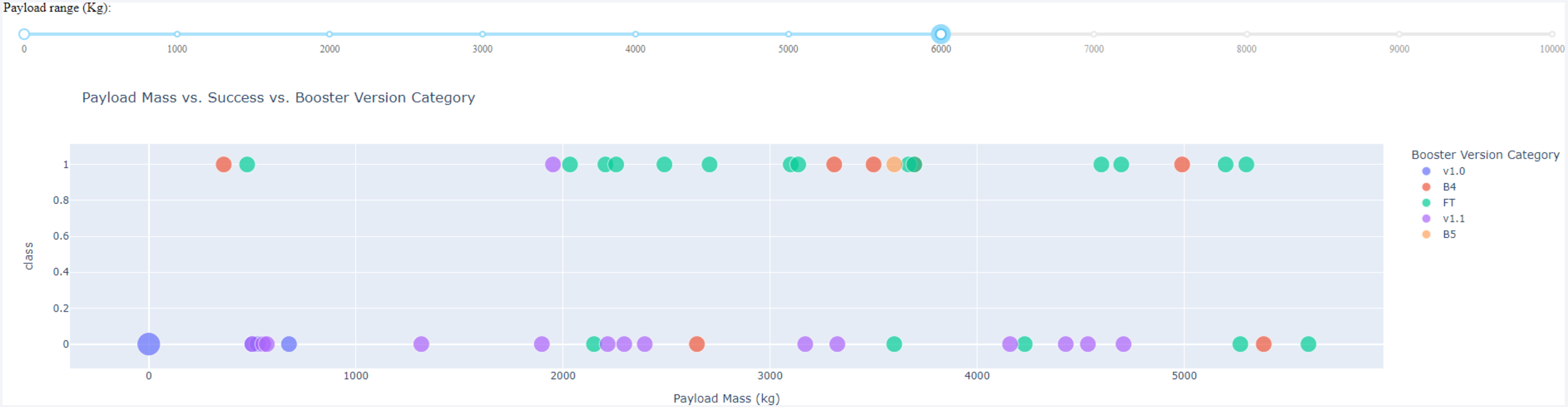
# Highest Launch Success Rate Site



KSC LC-39A Success Rate (blue=success)

KSC LC-39A: highest success rate (10 successful, 3 failed landings).

# Payload Mass & Success by Booster Version

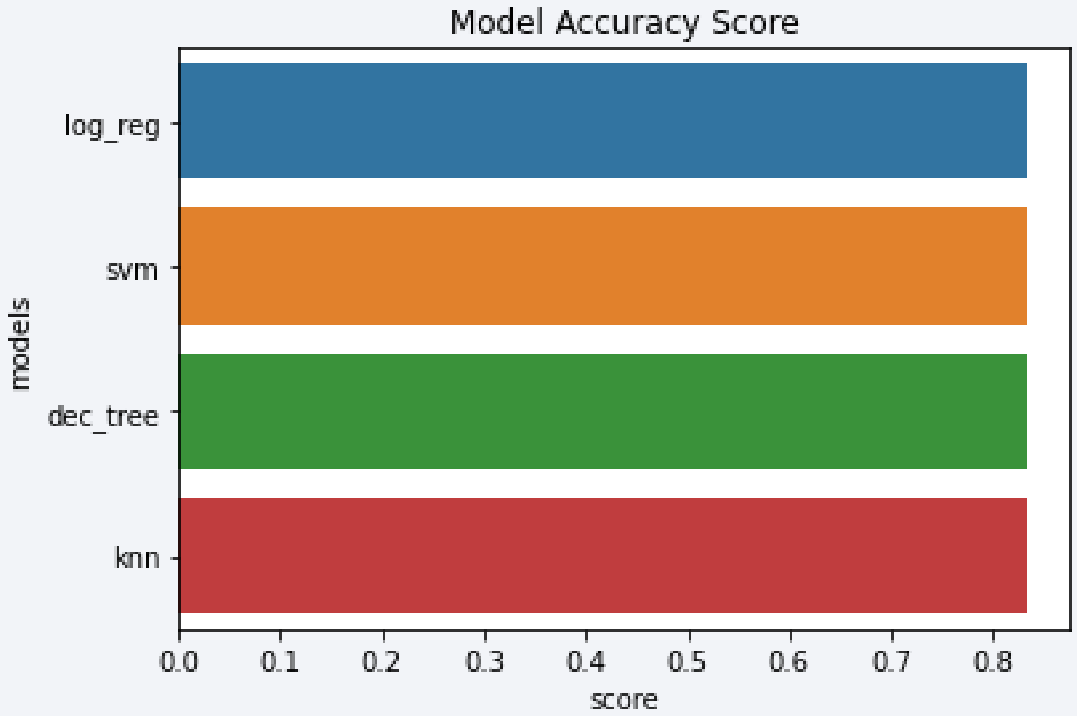


Key observations from the Plotly dashboard:
The payload range selector is set to 0-10,000, but the maximum payload is 15,600.
The scatter plot indicates successful landings with '1' and failures with '0'.
Booster version categories are represented by color and launch count by point size.
In the 0-6000 kg payload range, there are two failed landings with zero kg payloads.

Section 5

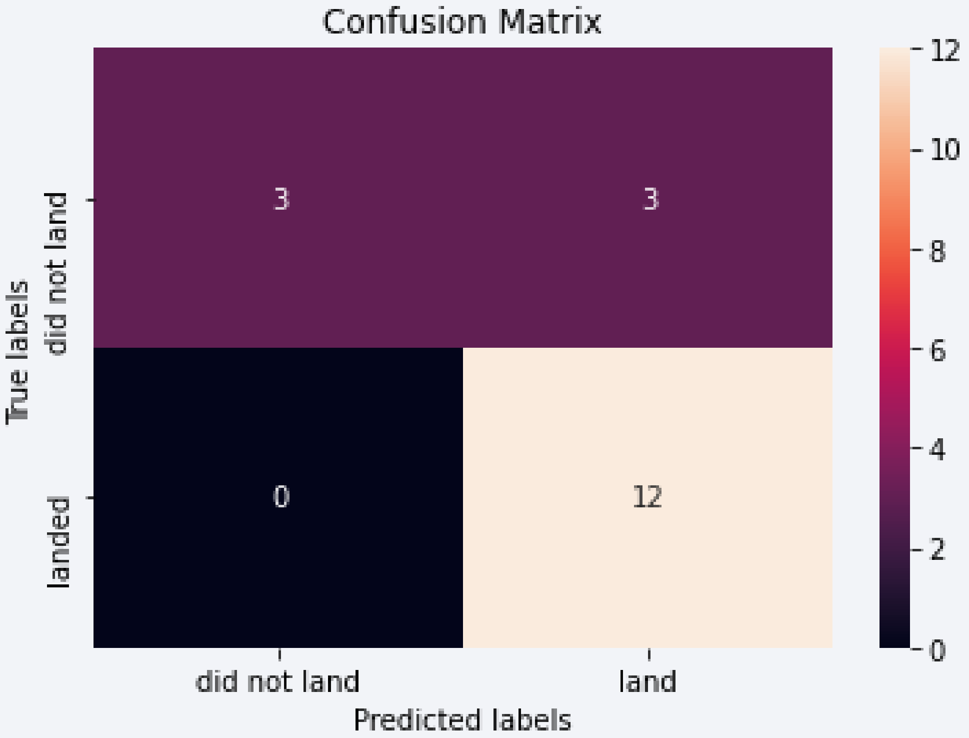
# Predictive Analysis (Classification)

# Classification Accuracy



The provided models demonstrated nearly identical performance on the test set, achieving an accuracy of 83.33%. However, it is crucial to acknowledge the limited size of the test set, comprising only 18 samples. This small sample size can lead to significant variability in accuracy, as observed with the Decision Tree Classifier model across multiple runs. Consequently, a larger dataset is likely necessary to definitively identify the optimal model.

# Confusion Matrix



Correct predictions lie on the top-left to bottom-right diagonal.

Since all models achieved identical test set results, they produced a single confusion matrix with the following key figures:
**True Positives:** 12 (Successful landings correctly predicted)
**True Negatives:** 3 (Unsuccessful landings correctly predicted)
**False Positives:** 3 (Unsuccessful landings *incorrectly* predicted as successful)
The presence of these 3 false positives highlights a clear tendency for the models to over-predict successful landings.

# Conclusions

- We developed a machine learning model for **Space Y** to predict SpaceX Stage 1 landing success, aiming to enhance competitiveness and potentially save **~$100 million USD** per launch.

- **Project Highlights:**
  - **Data:** Sourced from the SpaceX API and Wikipedia.
  - **Process:** Involved data labeling, DB2 storage, and dashboard visualization.
  - **Result:** Achieved an **83.33% accurate** prediction model.

- **Value & Application:**
  - This model enables **Space Y** to assess landing risks *before* launch, facilitating more cost-effective decisions.

- **Future Recommendations:**
  - Due to the limited test set size, we recommend **collecting more data** to further refine the model and improve accuracy.

# Appendix

- GitHub URL：
    - https://github.com/williamwang-ty/ibm-ds-capstone

Thank you!