IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Josué Montila.

08/07/2025

# Outline

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

# Executive Summary

- In this capstone project, our objective is to develop a predictive model that can determine whether the **SpaceX Falcon 9 first stage** will successfully land or not, based on various factors related to the rocket's first stage.
- We begin with **data collection, cleaning, and exploratory data analysis (EDA)** using **SQL queries** and **data visualizations**.
- We also visualize the **launch sites on an interactive map using Folium**, and create an interactive **dashboard for data visualization** using **Plotly and Dash**.
- Through graphical analysis, we observe that **certain rocket features show strong relationships** with the success or failure of the landing.
- Multiple **machine learning models** are developed and evaluated to identify the one that most accurately predicts the **landing outcome**—whether it's a success or failure.

# Introduction

- SpaceX promotes the Falcon 9 rocket launches on its official website at a cost of $62 million per launch, whereas competitors often charge over $165 million. A significant portion of this cost efficiency comes from SpaceX's ability to reuse the rocket's first stage. As a result, being able to predict the success of the first stage landing can help estimate the overall launch cost. This insight can be valuable for other companies aiming to compete with SpaceX in the launch services market.

- The primary focus of our project is to determine whether the Falcon 9's first stage lands successfully or not.

- We are conducting a detailed data analysis to gather relevant and reliable data that will be used to predict the success of Falcon 9 landings.

- Our approach involves developing a machine learning classification model that examines various features and predicts the likelihood of a successful Falcon 9 first stage landing.

Section 1

# Methodology

# Methodology

Data Collection Approach:

- The data for this project is obtained through two primary methods: accessing the SpaceX API and web scraping data from Wikipedia.
- SpaceX offers several APIs that return JSON-formatted data, providing up-to-date details on their space missions and operations.
- Additionally, Wikipedia hosts comprehensive information on various space missions, which can be extracted through web scraping techniques to meet our data needs.

# Methodology (cont..)

Data Wrangling Process:

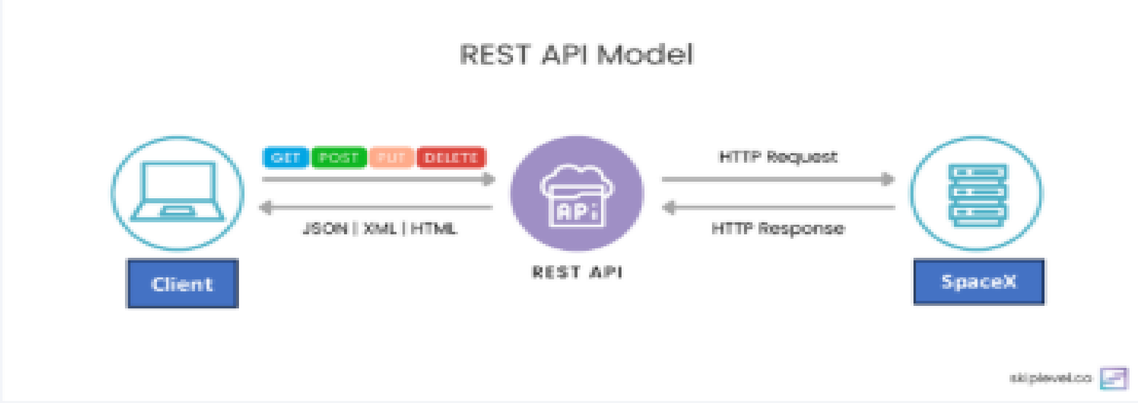
- The raw dataset contains missing entries, irrelevant values, and inconsistent data types.
- To clean the data, we either replace missing values with suitable alternatives or eliminate them, depending on the context and relevance.

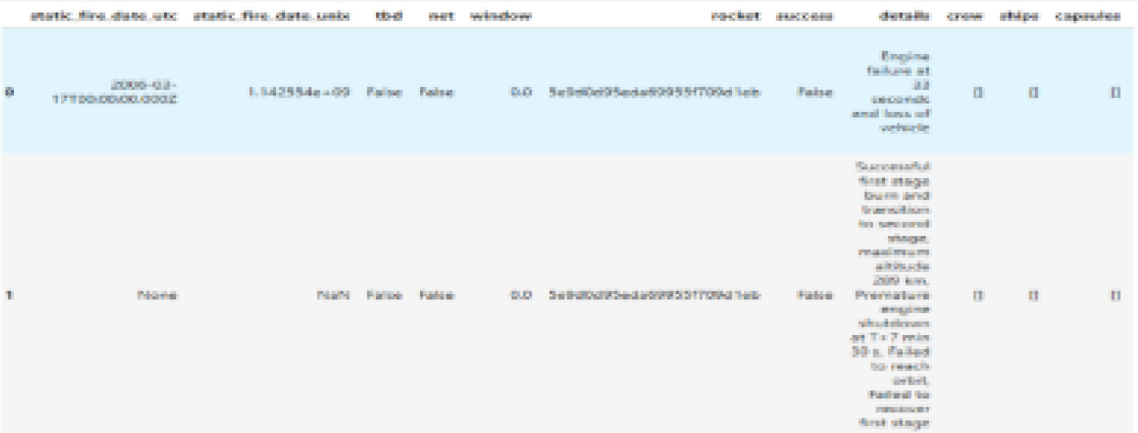
# Data Collection

- Data collection is the initial and crucial phase of any data analysis process.
- In this project, we are gathering information related to Falcon 9 first stage landings using two main sources:
- SpaceX APIs, and
- Web scraping from Wikipedia.
- SpaceX provides public APIs on its website that allow us to retrieve up-to-date data about rocket launches and various space missions.
- Wikipedia contains detailed and reliable records of space missions, making it a valuable source for scraping relevant data.

# Data Collection – SpaceX API

- We utilized the requests library in Python to send a GET request to the SpaceX API.
- The API responded with a 200 status code, indicating success, and returned the data in JSON format.
- This JSON data was then processed and normalized using Pandas, and transformed into a DataFrame for further analysis.
- The completed notebook for the SpaceX API data extraction is available on GitHub at:GitHub URL goes here)



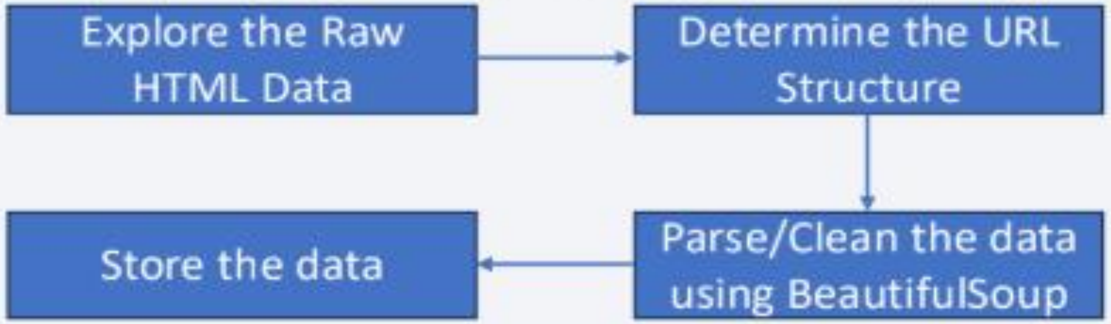


| | static_fire_date_utc | static_fire_date_unix | tbd | net | window | rocket | success | details | crew | ships | capsules |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2006-03-17T00:00:00.000Z | 1.142554e+09 | False | False | 0.0 | 5e9d0d95eda69955f709d1eb | False | Engine failure at 33 seconds and loss of vehicle | [] | [] | [] |
| 1 | None | NaN | False | False | 0.0 | 5e9d0d95eda69955f709d1eb | False | Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage | [] | [] | [] |

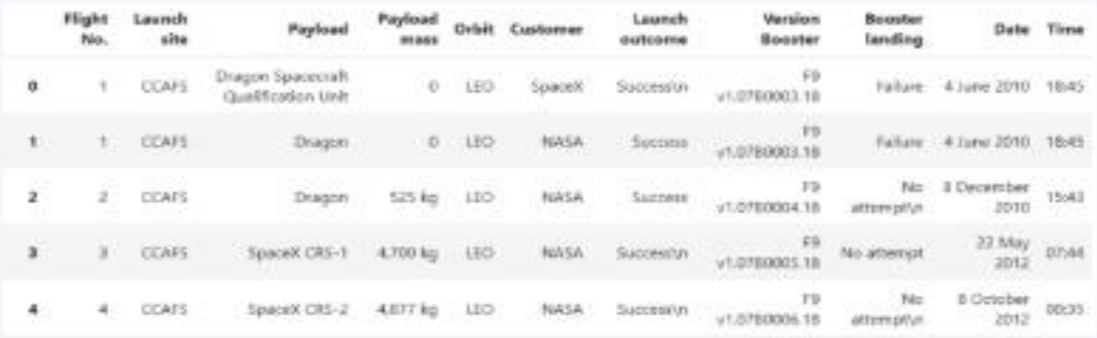
# Data Collection - Scraping

•We began by sending a **requests.get()** call to the **webpage URL** to retrieve its content.
•The website responded with the **HTML content**, which we read as plain text.
•We then used **BeautifulSoup** to **parse and structure the HTML**, allowing us to extract the required data in an organized manner.
•The full notebook for the web scraping process is available on GitHub at:
GitHub URL goes here)

Web Scrapping Structure :


Explore the Raw HTML Data
Determine the URL Structure
Parse/Clean the data using BeautifulSoup
Store the data

Web Scraped Data of Falcon9:

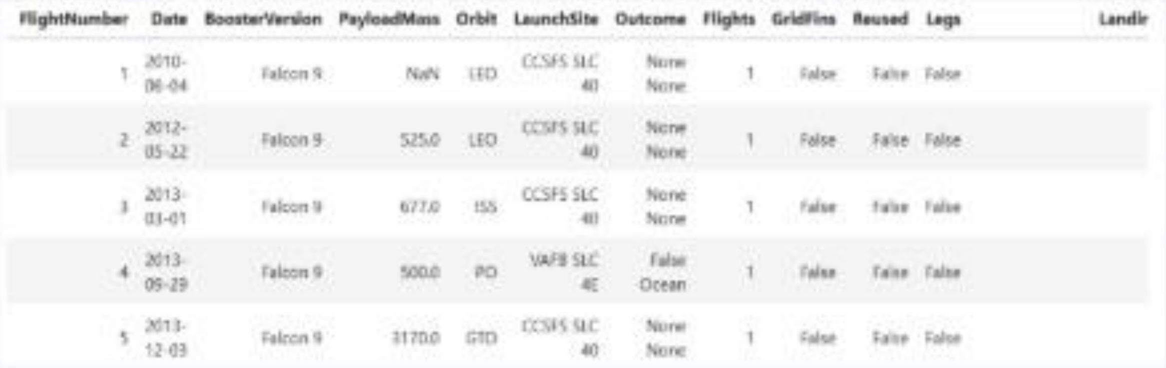


| | Flight No. | Launch site | Payload | Payload mass | Orbit | Customer | Launch outcome | Version Booster | Booster landing | Date | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | CCAFS | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success\n | F9 v1.07B0003.18 | Failure | 4 June 2010 | 18:45 |
| **1** | 1 | CCAFS | Dragon | 0 | LEO | NASA | Success | F9 v1.07B0003.18 | Failure | 4 June 2010 | 18:45 |
| **2** | 2 | CCAFS | Dragon | 525 kg | LEO | NASA | Success | F9 v1.07B0004.18 | No attempt\n | 8 December 2010 | 15:43 |
| **3** | 3 | CCAFS | SpaceX CRS-1 | 4,700 kg | LEO | NASA | Success\n | F9 v1.07B0005.18 | No attempt | 22 May 2012 | 07:44 |
| **4** | 4 | CCAFS | SpaceX CRS-2 | 4,877 kg | LEO | NASA | Success\n | F9 v1.07B0006.18 | No attempt\n | 8 October 2012 | 00:35 |

# Data Wrangling

- The API data included both Falcon 1 and Falcon 9 launches, so we filtered out Falcon 1 entries.
- We found missing values: 5 for Payload Mass and 26 for Landing Pad. Rows with missing Payload Mass were removed.
- Flight numbers were reset starting from 1.
- A new "Class" column was created: 1 for successful landings, 0 for failures, based on the "Outcome" column.
- GitHub URL of the data wrangling notebook: Your URL here

The **final DataFrame** consists of **90 rows and 17 columns**.
The columns include:
**FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude**, and **Latitude**.



| FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | Landir |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2010-06-04 | Falcon 9 | NaN | LEO | CCSFS SLC 40 | None None | 1 | False | False | False | |
| 2 | 2012-05-22 | Falcon 9 | 525.0 | LEO | CCSFS SLC 40 | None None | 1 | False | False | False | |
| 3 | 2013-03-01 | Falcon 9 | 677.0 | ISS | CCSFS SLC 40 | None None | 1 | False | False | False | |
| 4 | 2013-09-29 | Falcon 9 | 500.0 | PO | VAFB SLC 4E | False Ocean | 1 | False | False | False | |
| 5 | 2013-12-03 | Falcon 9 | 3170.0 | GTO | CCSFS SLC 40 | None None | 1 | False | False | False | |

# EDA with Data Visualization

During Exploratory Data Analysis (EDA) with visualizations, we explored relationships between various features in the dataset.

We used scatter plots and bar/line charts to identify potential correlations influencing launch outcomes.

Scatter plots were used to analyze how launch success is affected by combinations of:

Flight Number vs. Payload Mass

Flight Number vs. Launch Site

Payload Mass vs. Launch Site

Flight Number vs. Orbit Type

Payload Mass vs. Orbit Type

# EDA with Data Visualization (cont…)

We also used:

A bar chart to visualize success rate by Orbit Type

A line chart to observe success rate by Year

GitHub URL for the completed EDA and visualization notebook:URL

# EDA with SQL

- Data analysis was performed using SQL queries to identify patterns and insights.
- Data was stored in a SQLite3 database, and queries were executed using connection and cursor objects.
- We analyzed the maximum Payload Mass across different booster versions and customers.
- Mission outcomes were grouped to count successes and failures, and trends were explored over time and across payload mass ranges.
- The date of the first successful landing was also identified.
- GitHub URL for the completed SQL analysis notebook:URL

# Build an Interactive Map with Folium

- This part of the project focused on three key tasks:
    - Plotting all launch sites on a map
    - Marking successful and failed launches at each sit
    - Calculating distances from launch sites to nearby points of interest
- The map was built using folium.Map() with defined coordinates and zoom level.
- We added elements like circles and popups for each launch site using .add_child().
- Marker Clusters were used to display launch outcomes (success/failure).
- Distance markers and connecting lines were added to show proximity to nearby features.
- GitHub URL of completed interactive map with Folium map: [Insert URL]

# Build a Dashboard with Plotly Dash

- We developed a dashboard to visually present SpaceX launch data analytics.
- A dropdown menu allows selection of all launch sites or a specific site.
- Based on the selection, a pie chart displays the success rate for all or selected sites.
- A range slider lets users filter by Payload Mass.
- Using the selected site(s) and payload range, a scatter plot shows Payload Mass vs. Class across Booster Versions.
- The dashboard layout includes a heading followed by four components: Dropdown, Pie Chart, Range Slider, and Scatter Plot.
- GitHub URL for the completed dashboard: URL

# Predictive Analysis (Classification)

- We built four classification models to predict Falcon 9 landing outcomes as Success (1) or Failure (0) using various features:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K-Nearest Neighbors (KNN)
- The data was first scaled using StandardScaler, then split into training and testing sets (80/20).
- GridSearchCV was used to tune hyperparameters for each model.
- Finally, we compared model performance using the confusion matrix and accuracy score on the test set.
- GitHub URL for the completed machine learning analysis: URL

# Results

- Exploratory Data Analysis (EDA) Results:
- The first successful landing occurred on 2015-12-22.
- Boosters that successfully landed on a drone ship with payloads between 4000–6000 kg include:
- F9 FT B1022, B1026, B1021.2, and B1031.2.
- The F9 B5 booster series carried the maximum payload mass overall.
- The average payload mass for the F9 v1.1 booster is approximately 2535 kg.

# Results

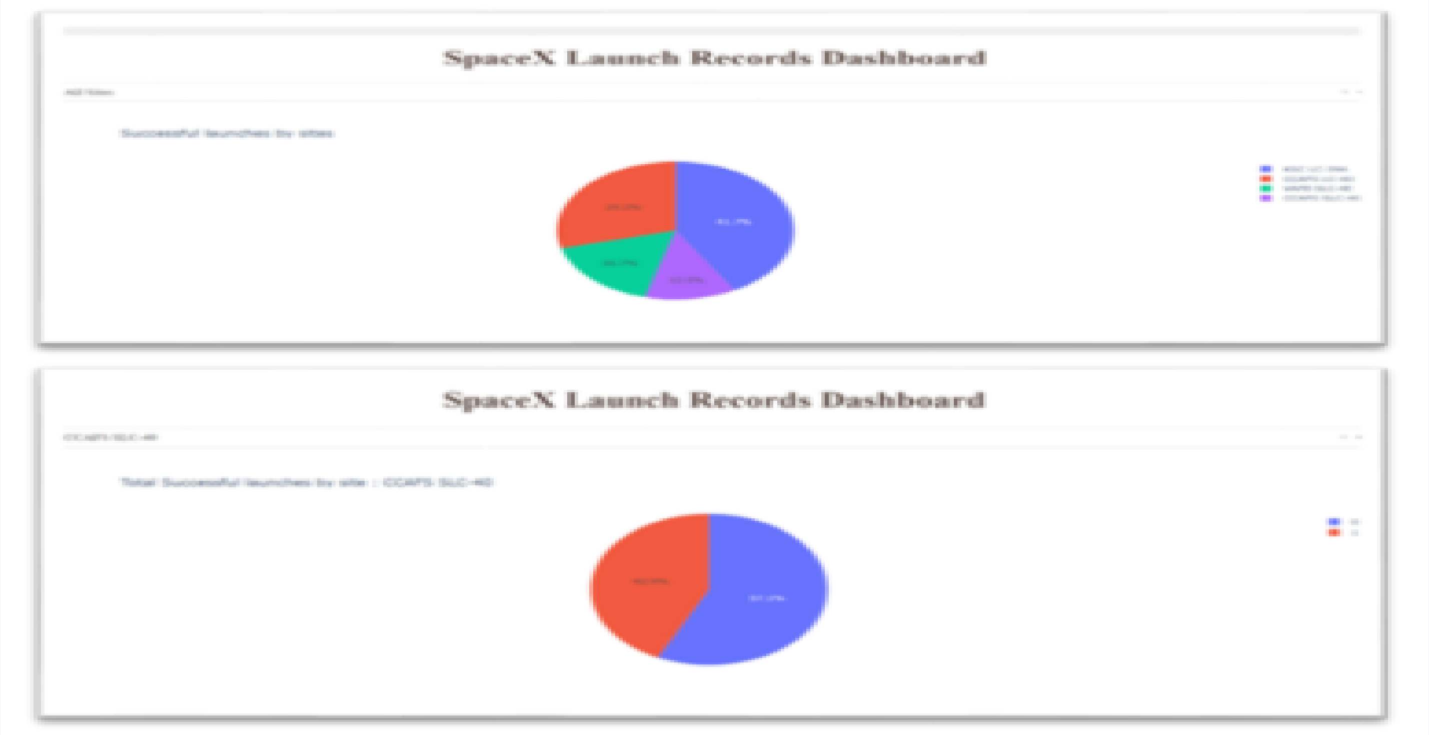
- Predictive Analysis Results:
- All models achieved a test accuracy of 83.33%, with similar recall, precision, and true positive rate.

- The Decision Tree Classifier showed the highest training accuracy (88.75%), possibly due to overfitting, while others averaged around 85%.

- Since the train-test accuracy gap is small, the Decision Tree may still be the most suitable model for classification.

# Results

- Interactive analytics demo :



SpaceX Launch Records Dashboard

All Sites

Successful launches by sites

SpaceX Launch Records Dashboard

Total Successful launches by site : CCAFS- SLC-40

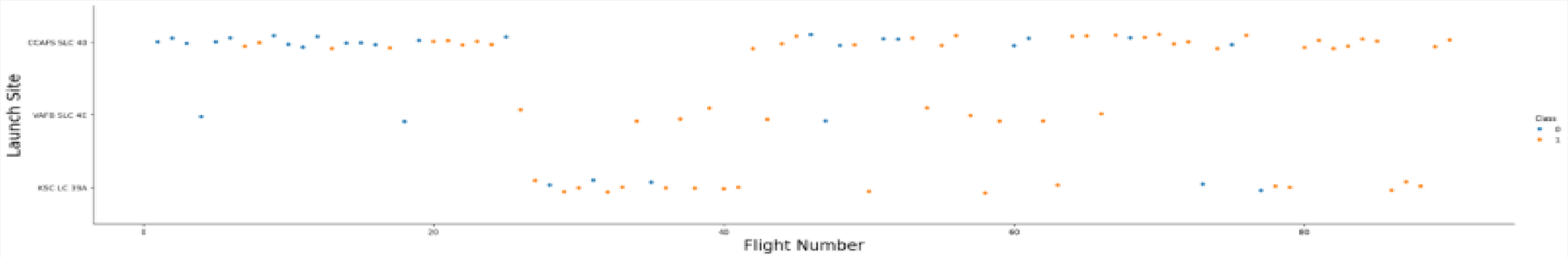
Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site

- At each launch site, a higher Flight Number generally indicates a greater chance of a successful landing.

- VAFB SLC 4E shows a high success rate even with fewer launches compared to other sites.

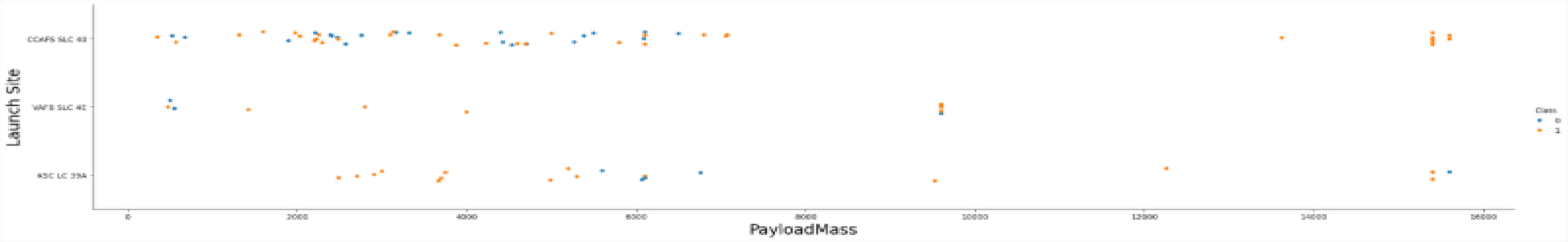


Scatter plot of Flight Number vs. Launch Site

# Payload vs. Launch Site

- At CCAFS SLC 40, a higher Payload Mass tends to increase the chances of a successful landing.

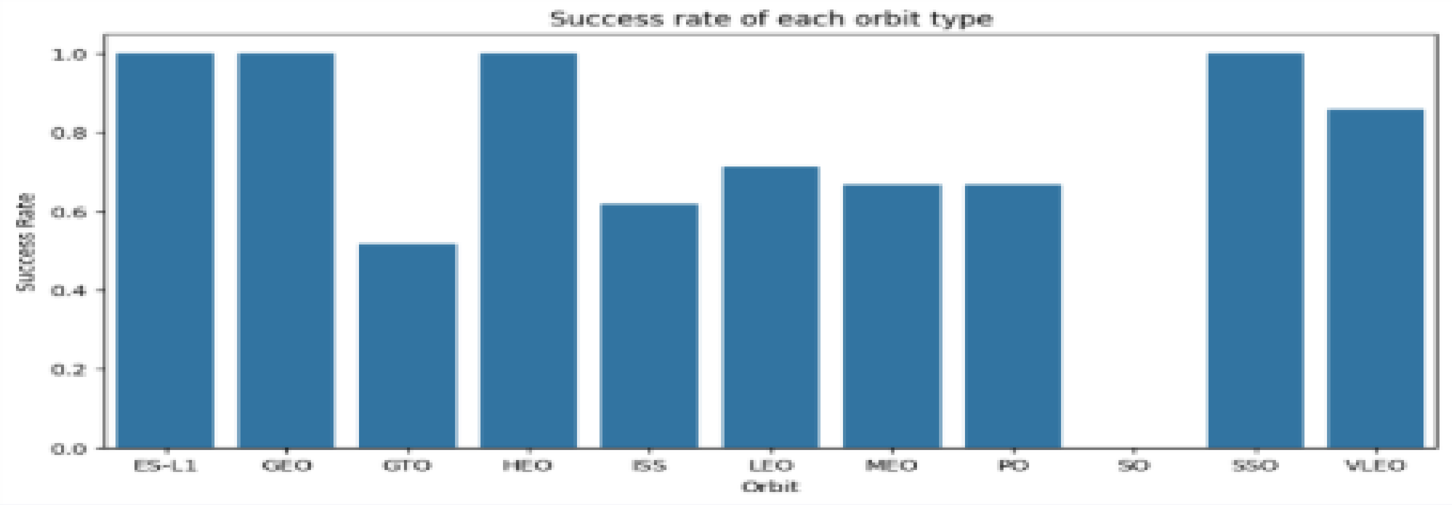
- However, at VAFB SLC 4E and KSC LC 39A, there is no clear correlation between Payload Mass and landing outcome.



Scatter plot of Payload Mass vs. Launch Site

# Success Rate vs. Orbit Type

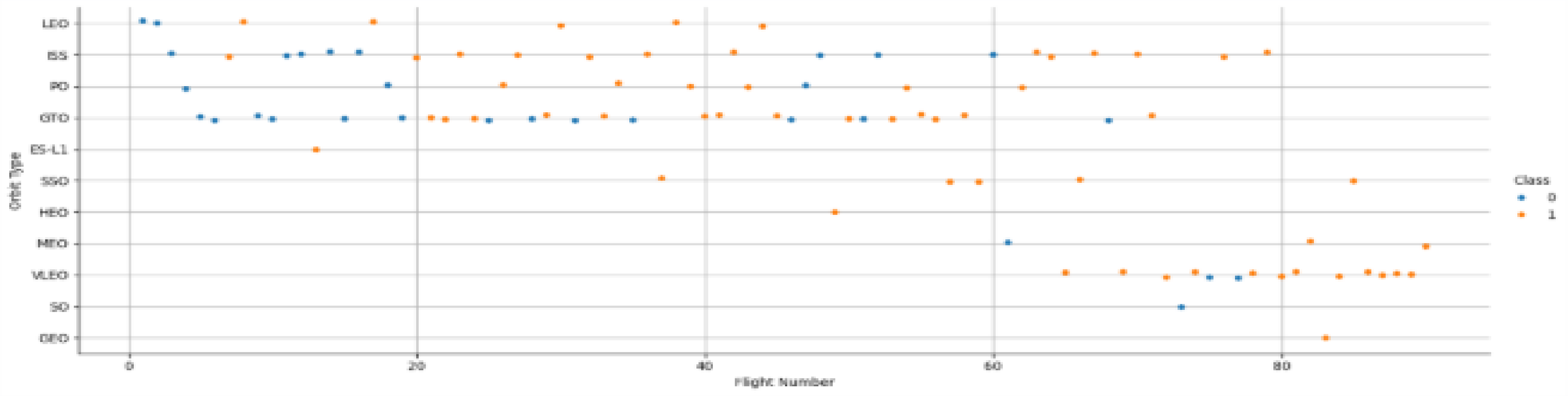
- Flights to ESL1, GEO, HEO, and SSO orbits have shown a 100% success rate in Falcon 9 first-stage landings, whereas flights to SO orbit have had no successful landings.
- Other orbits show a landing success rate of 50% or higher.



Bar plot of Success rate vs. Orbit type

# Flight Number vs. Orbit Type

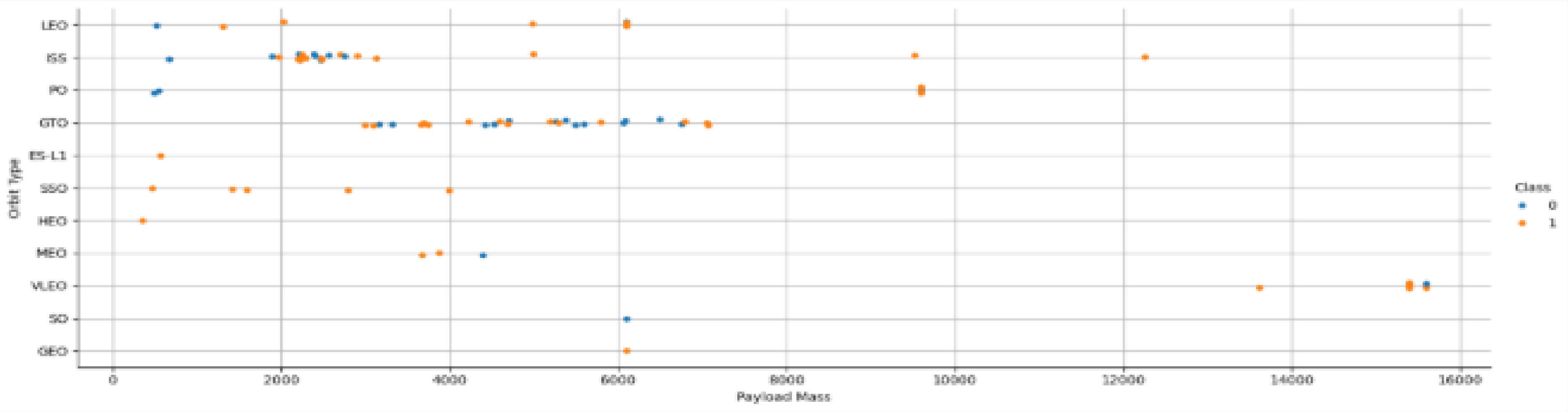
- Flights to LEO, VLEO, and MEO orbits show a correlation between the number of flights and successful first-stage landings of Falcon 9.
- In contrast, flights to other orbits show little to no correlation between flight count and landing success.



Scatter plot of Flight Number vs. Orbit type

# Payload vs. Orbit Type

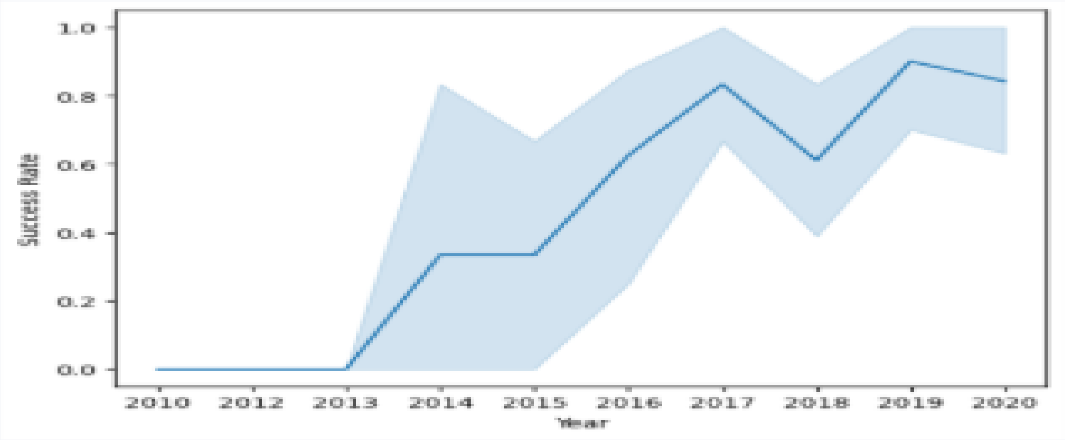
- Flights to LEO, ISS, and PO orbits show that higher Payload Mass is associated with increased success in first-stage landings.
- Flights to other orbits do not display a clear relationship between Payload Mass and landing success.



Scatter plot of Payload Mass vs. Orbit type

# Launch Success Yearly Trend

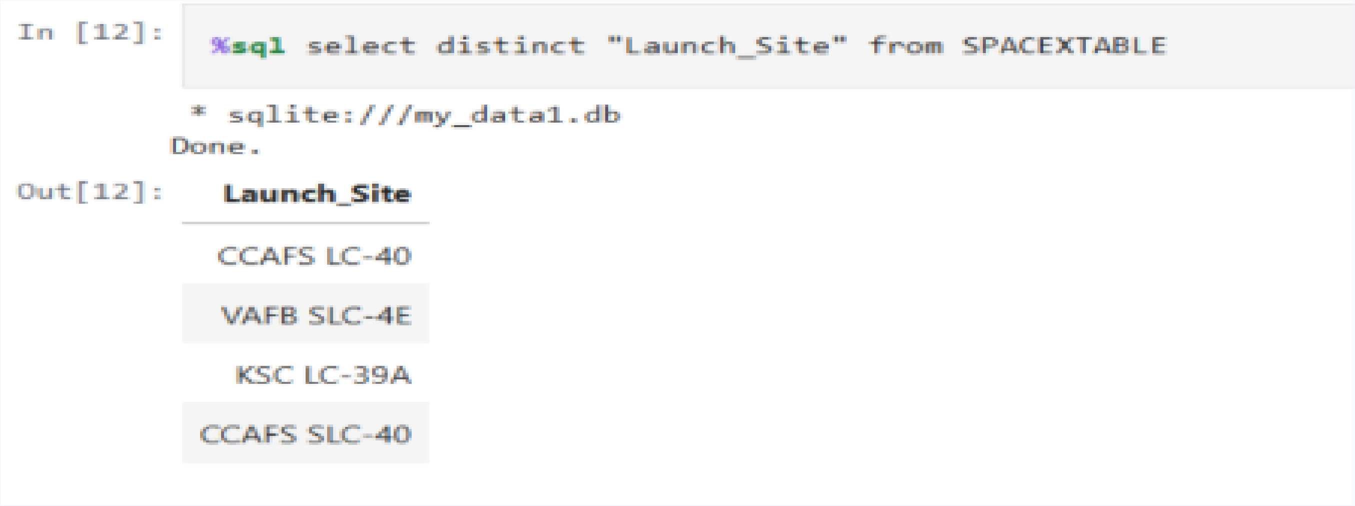
- The graph indicates a steady rise in success rate starting from 2013.
- There is a slight dip in success rate between 2017 and 2018.
- After 2018, the success rate increases again, continuing up to 2020.



Line plot of Year vs. Success Rate

# All Launch Site Names

- • We used Distinct method to find unique Launch Site names from a
- SPACEXTABLE table using SQL query as :
- %sql select distinct "Launch_Site" from SPACEXTABLE

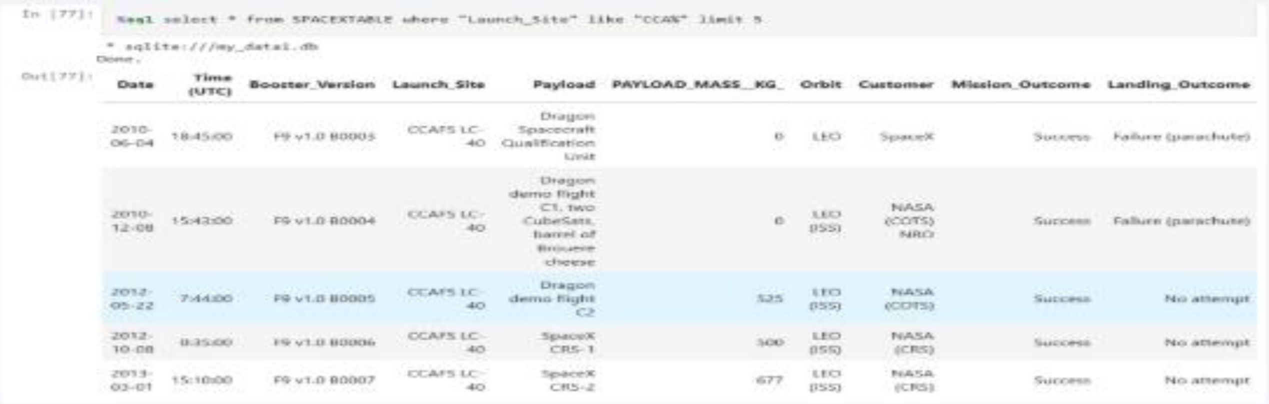


```
In [12]:  %sql select distinct "Launch_Site" from SPACEXTABLE

           * sqlite:///my_data1.db
          Done.
Out[12]:
```

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

# Launch Site Names Begin with 'CCA'

-  Launch Sites with names begin “CCA” are queried using “like” clause on launch site column by passing value as “CCA%”.
- •Limit 5 clause is used to get only 5 records.
- %sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5

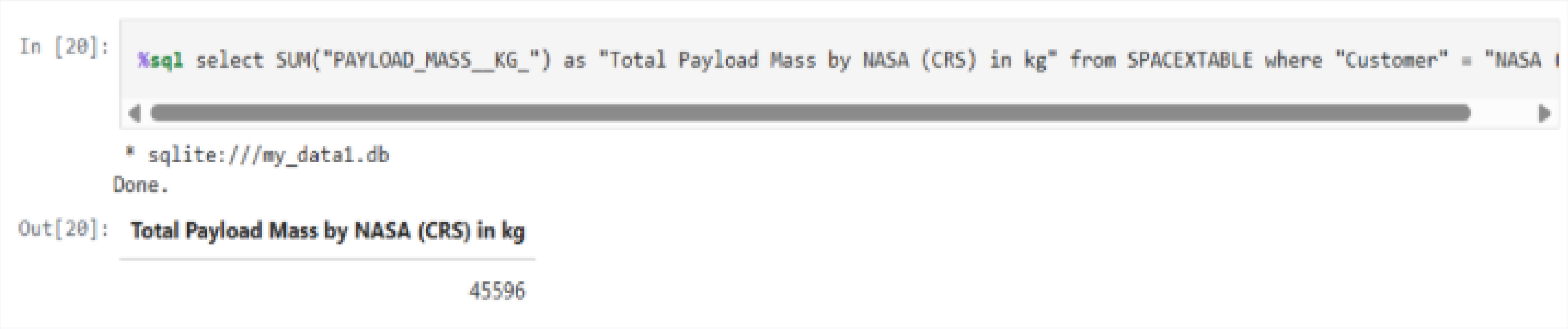


```
In [77]:  %sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5
 * sqlite:///my_data1.db
Done.
Out[77]:
```

| 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

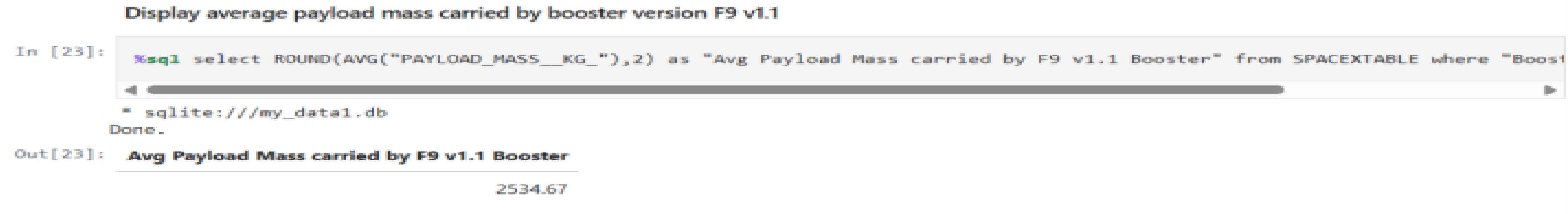
- The total payload mass for the customer "NASA (CRS)" was calculated using the SUM() function on the "Payload_Mass__kg_" column with a WHERE clause to filter by customer.
- The total payload mass for NASA (CRS) amounts to 45,596 kg.%sql select SUM("PAYLOAD_MASS__KG_") as "Total Payload Mass by NASA (CRS) in kg" from
- SPACEXTABLE where "Customer" = "NASA (CRS)"



```
In [20]: %sql select SUM("PAYLOAD_MASS__KG_") as "Total Payload Mass by NASA (CRS) in kg" from SPACEXTABLE where "Customer" = "NASA (

 * sqlite:///my_data1.db
Done.

Out[20]: Total Payload Mass by NASA (CRS) in kg
                                         45596
```

# Average Payload Mass by F9 v1.1

- The average payload mass for boosters starting with F9 v1.1 is 2535 kg.
- This was calculated using the AVG() function.
- A WHERE clause with a LIKE condition was used to filter booster versions that begin with "F9 v1.1".
- %sql select ROUND(AVG("PAYLOAD_MASS__KG_"),2) as "Avg Payload Mass carried by F9 v1.1
- Booster" from SPACEXTABLE where "Booster_Version" like "F9 v1.1%“



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

```
In [23]: %sql select ROUND(AVG("PAYLOAD_MASS__KG_"),2) as "Avg Payload Mass carried by F9 v1.1 Booster" from SPACEXTABLE where "Boos
```

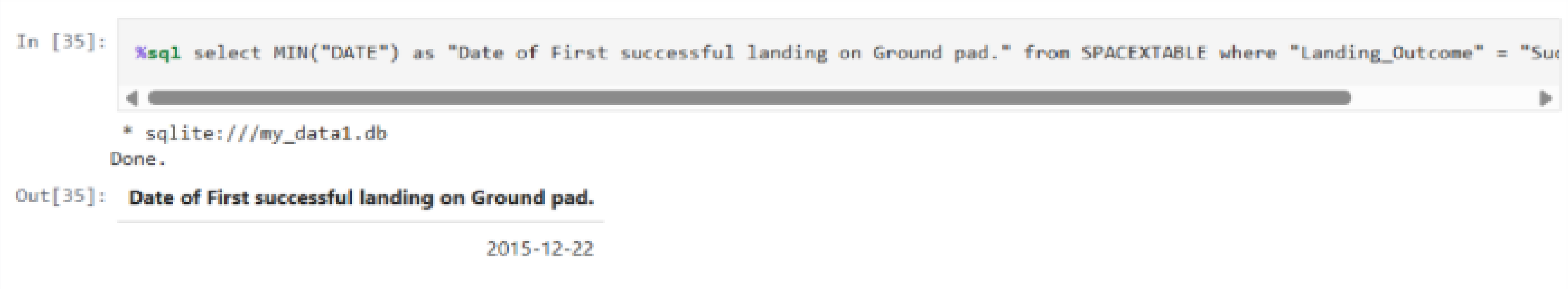
```
 * sqlite:///my_data1.db
Done.
```

Out[23]:

| Avg Payload Mass carried by F9 v1.1 Booster |
|---:|
| 2534.67 |

# First Successful Ground Landing Date

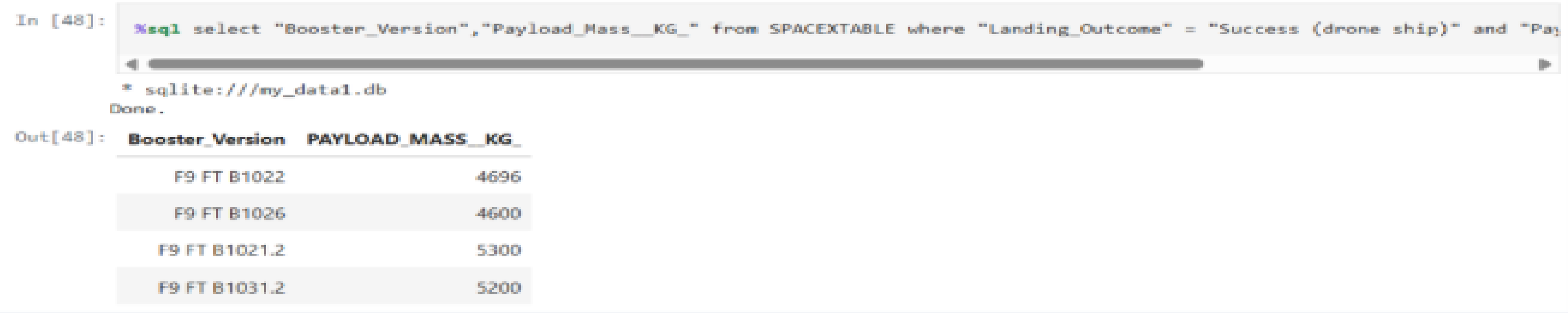
- The first successful ground landing occurred on December 22, 2015.
- The MIN() function was used to identify the earliest landing date.
- A WHERE clause filtered the data to include only landings with the outcome "Success (ground pad)".
- %sql select MIN("DATE") as "Date of First successful landing on Ground pad." from
- SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)“.



```
In [35]:  %sql select MIN("DATE") as "Date of First successful landing on Ground pad." from SPACEXTABLE where "Landing_Outcome" = "Suc

           * sqlite:///my_data1.db
          Done.
Out[35]:  Date of First successful landing on Ground pad.
                                                2015-12-22
```

# Successful Drone Ship Landing with Payload between 4000 and 6000

- Successful Drone Ship Landings with Payload between 4000 and 6000 are : F9 FT
- B1022 , F9 FT B1036 , F9 FT B1021.2 , F9 FT B1031.2 .
- WHERE clause is used with and clause to select the Landing Outcome as “Success
- (drone ship)” and Payload Range as 4000 to 6000 kg. For Range “between .. and ..”
- clause is used.
- %sql select "Booster_Version","Payload_Mass__KG_" from SPACEXTABLE where "Landing_Outcome" = "Success (drone
- ship)" and "Payload_mass__Kg_" between 4000 and 6000



```
In [48]:  %sql select "Booster_Version","Payload_Mass__KG_" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and "Pa
```

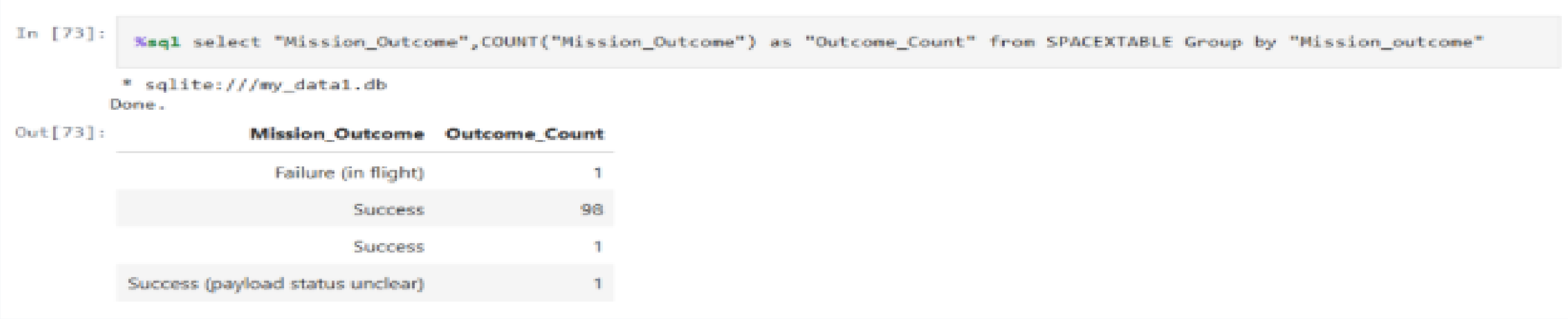
```
 * sqlite:///my_data1.db
Done.
```

Out[48]:

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 FT B1022 | 4696 |
| F9 FT B1026 | 4600 |
| F9 FT B1021.2 | 5300 |
| F9 FT B1031.2 | 5200 |

# Total Number of Successful and Failure Mission Outcomes

- There are total 100 successful missions and 1 Failure in flight mission
- outcomes in data.
- Group by clause is used to group “Mission Outcome” and COUNT() method is
- used to get count of Outcomes.
- %sql select "Mission_Outcome",COUNT("Mission_Outcome") as "Outcome_Count" from SPACEXTABLE Group by "Mission_outcome"



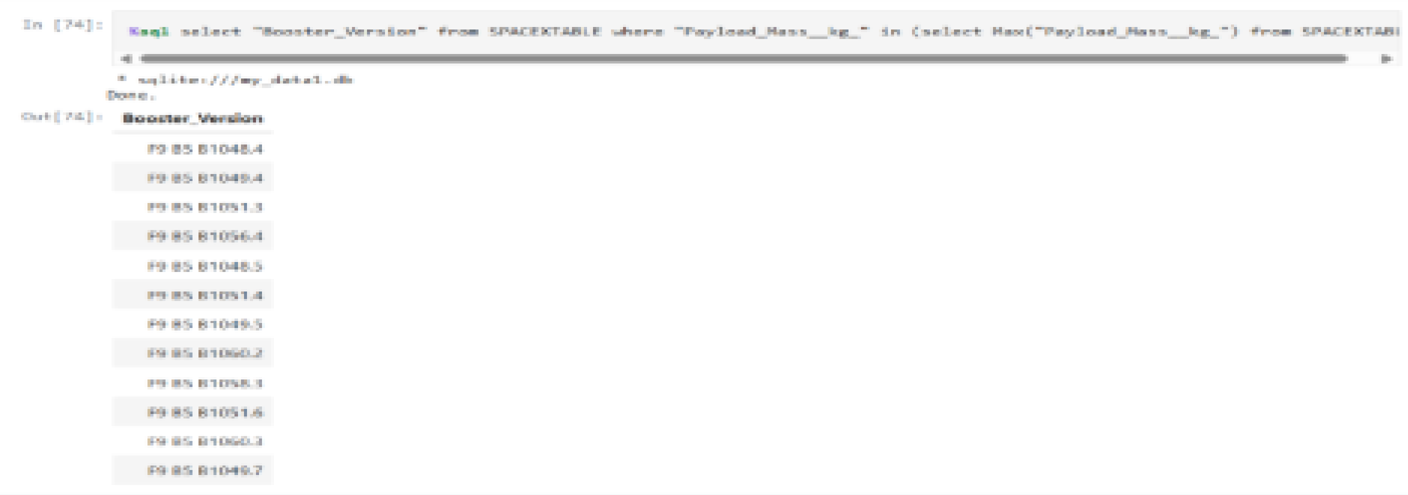
```
In [73]: %sql select "Mission_Outcome",COUNT("Mission_Outcome") as "Outcome_Count" from SPACEXTABLE Group by "Mission_outcome"

 * sqlite:///my_data1.db
Done.
Out[73]:
```

| Mission_Outcome | Outcome_Count |
|---:|---:|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

- The boosters that carried the maximum payload belong to the F9 B5 series.
- A subquery was used to find the maximum value in the "Payload_Mass_kg_" column.
- The corresponding "Booster_Version" values were then selected to identify the specific boosters.

%sql select "Booster_Version" from SPACEXTABLE where "Payload_Mass__kg_" in (select Max("Payload_Mass__kg_") from SPACEXTABLE)



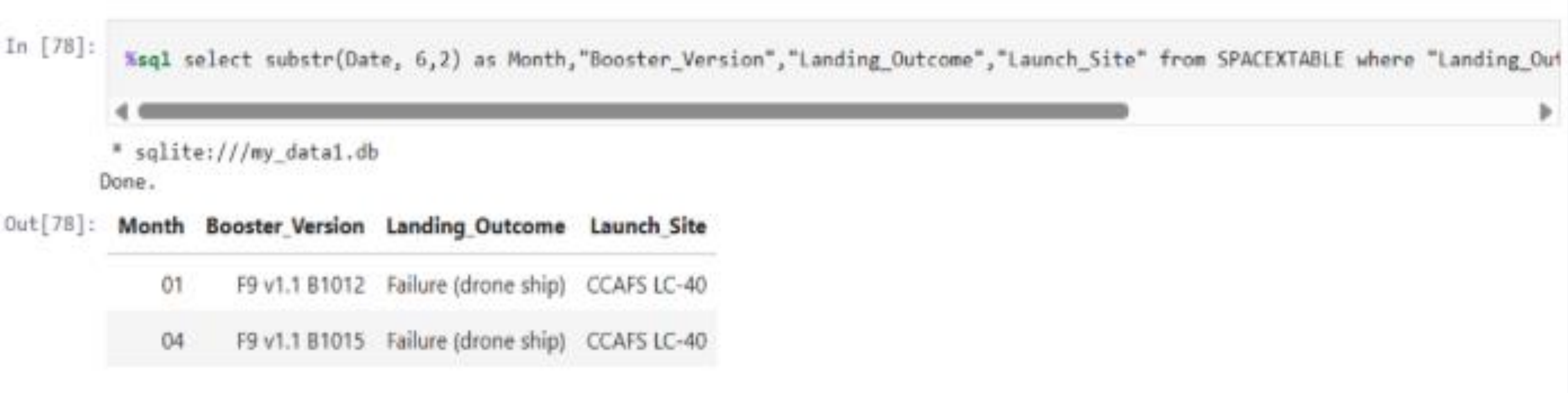
```
In [74]: %sql select "Booster_Version" from SPACEXTABLE where "Payload_Mass__kg_" in (select Max("Payload_Mass__kg_") from SPACEXTABI

 * sqlite:///my_data1.db
Done.
Out[74]: 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

- We used the substr() function to extract the month from the Date column and applied a LIKE clause to filter records from the year 2015.
- A WHERE clause was used to select rows where the Landing Outcome was "Failure (drone ship)".

%sql select substr(Date, 6,2) as Month,"Booster_Version","Landing_Outcome","Launch_Site" from

SPACEXTABLE where "Landing_Outcome"="Failure (drone ship)" and Date like "2015%"



In [78]:

```
%sql select substr(Date, 6,2) as Month,"Booster_Version","Landing_Outcome","Launch_Site" from SPACEXTABLE where "Landing_Out
```

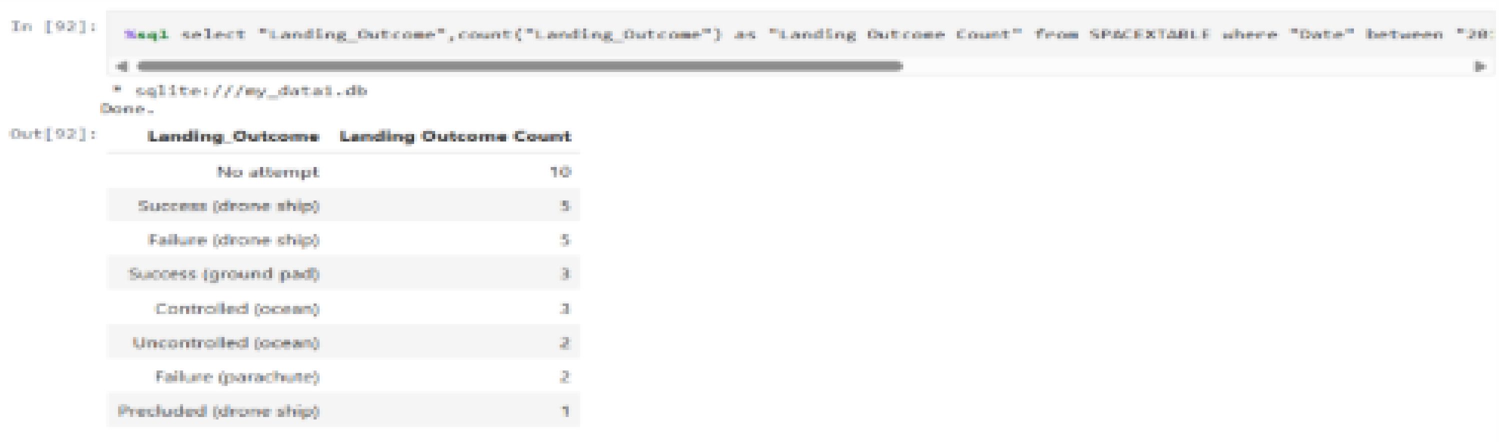
```
 * sqlite:///my_data1.db
Done.
```

Out[78]:

| Month | Booster_Version | Landing_Outcome | Launch_Site |
|---|---|---|---|
| 01 | F9 v1.1 B1012 | Failure (drone ship) | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | Failure (drone ship) | CCAFS LC-40 |

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

- The **date range** was filtered between **2010-06-04** and **2017-03-20** using a **WHERE clause**.
- **Landing outcomes** were grouped using the **GROUP BY clause**, and the **COUNT() function** was used to count each outcome type.
- An **ORDER BY clause** with **DESC** was applied to sort the outcomes in **descending order by count**.
- %sql select "Landing_Outcome",count("Landing_Outcome") as "Landing Outcome Count" from SPACEXTABLE where "Date“ between "2010-06-04" and "2017-03-20" group by "Landing_Outcome" order by count("Landing_Outcome") desc



```
In [92]: %sql select "Landing_Outcome",count("Landing_Outcome") as "Landing Outcome Count" from SPACEXTABLE where "Date" between "20
```

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

Out[92]:

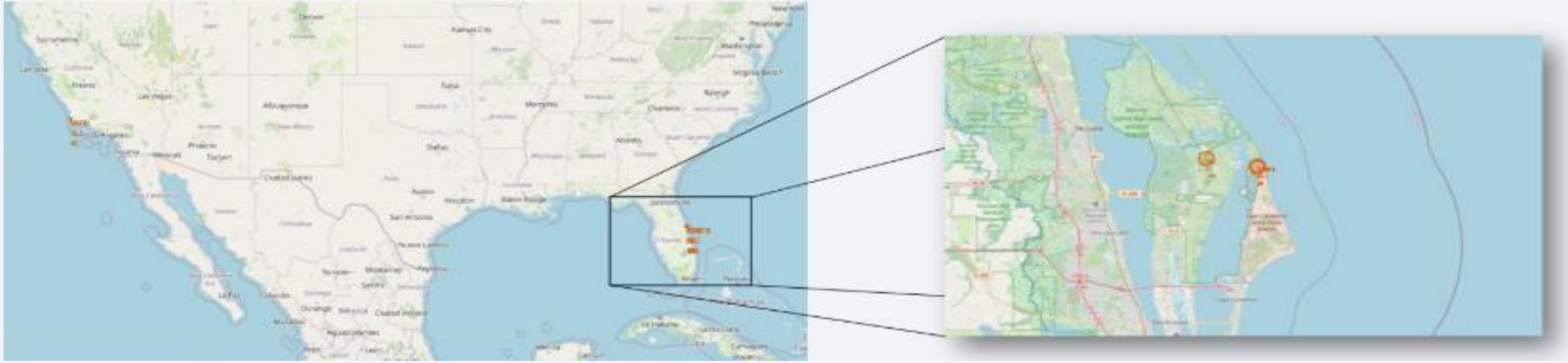
| Landing_Outcome | Landing Outcome Count |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

Section 3

# Launch Sites Proximities Analysis

# Locations of all Launch Sites on World Map

- There are four launch sites in total — three located on the U.S. East Coast and one on the West Coast.
- A 1 km radius circle was drawn around each launch site, along with a marker displaying the site name in a popup.



# Success and Failed Launches for each site on Map

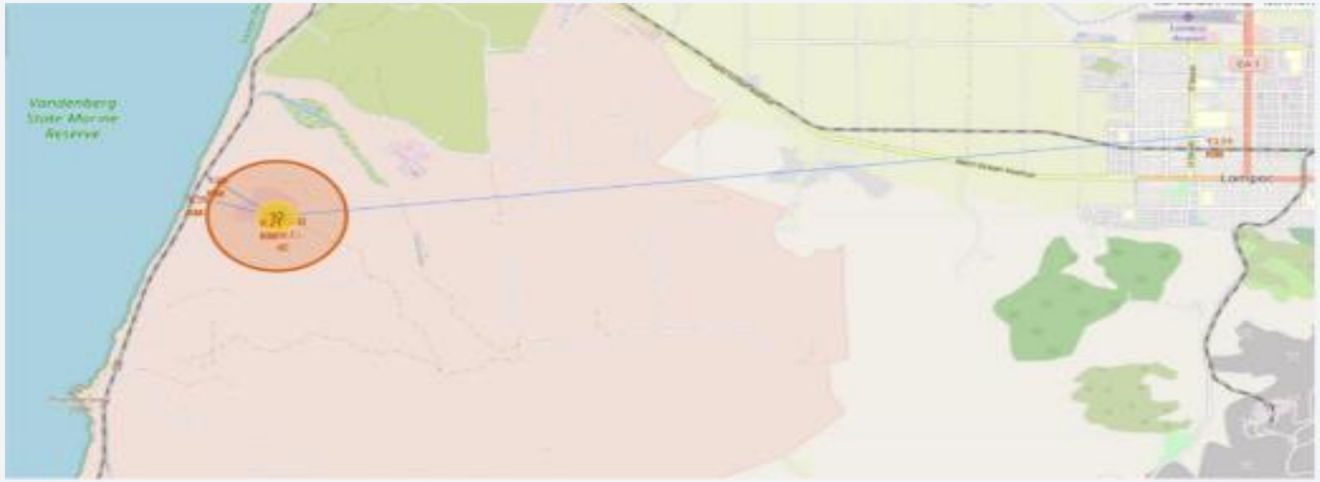
- Green markers represent successful launches, while red markers indicate failures, and are placed on the map for each launch site.
- A Marker Cluster is used to group markers that are located at the same or nearby positions.
- Clicking on a cluster reveals the individual success (green) and failure (red) icons.

# Mapping proximities from Launch Sites

- Distances to the nearest features—such as the coastline, railway, highway, and city—were calculated using their coordinates and those of the launch sites.
- A distance marker was placed at each point to display the distance from the launch site.
- Lines were drawn to visually connect each launch site with its corresponding nearby location.

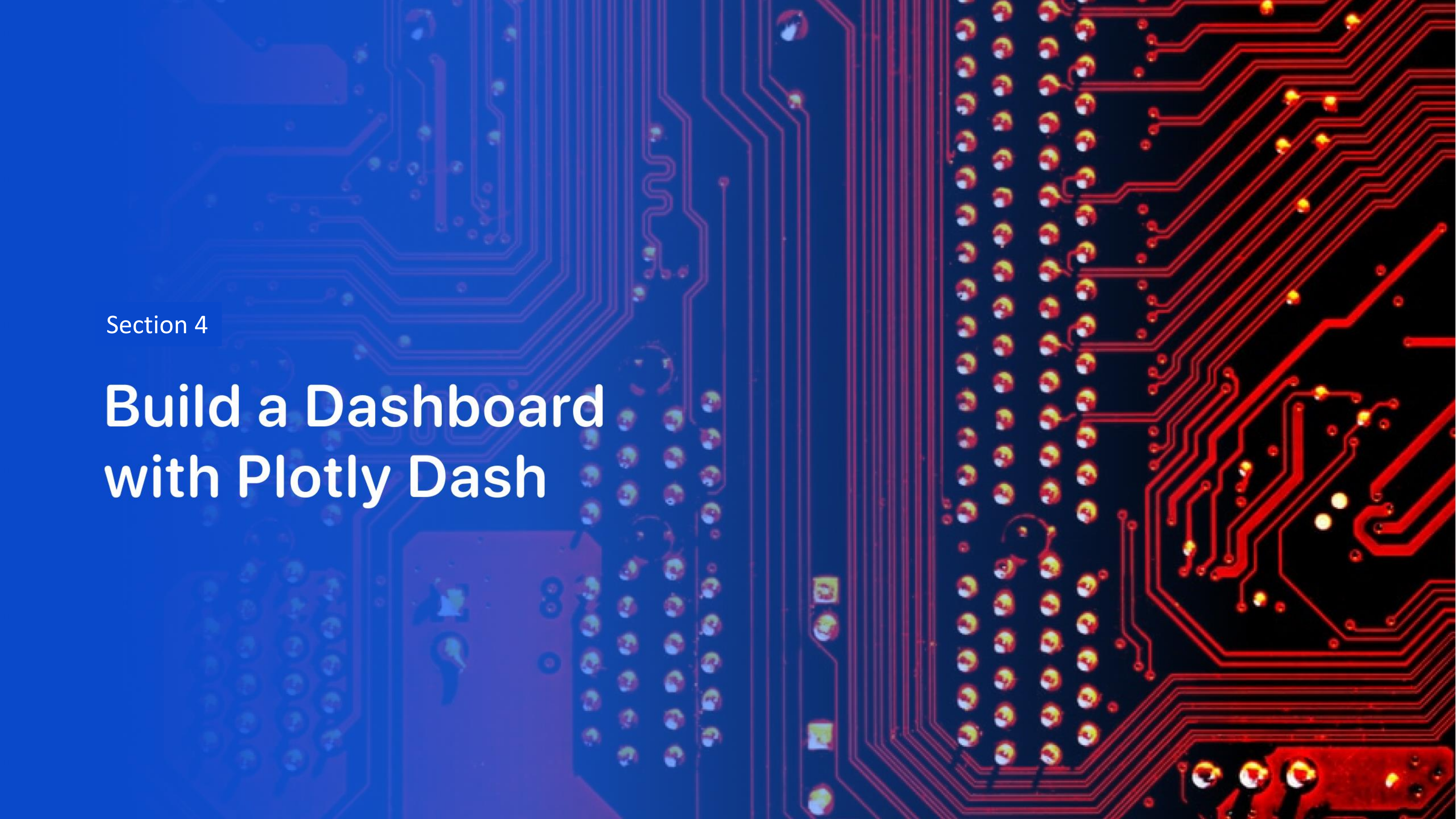
Distance from Launch Site:

- Coast line : 1.35 km
- Railway line : 1.25 km
- Highway road : 0.21 km
- City : 13.51 km



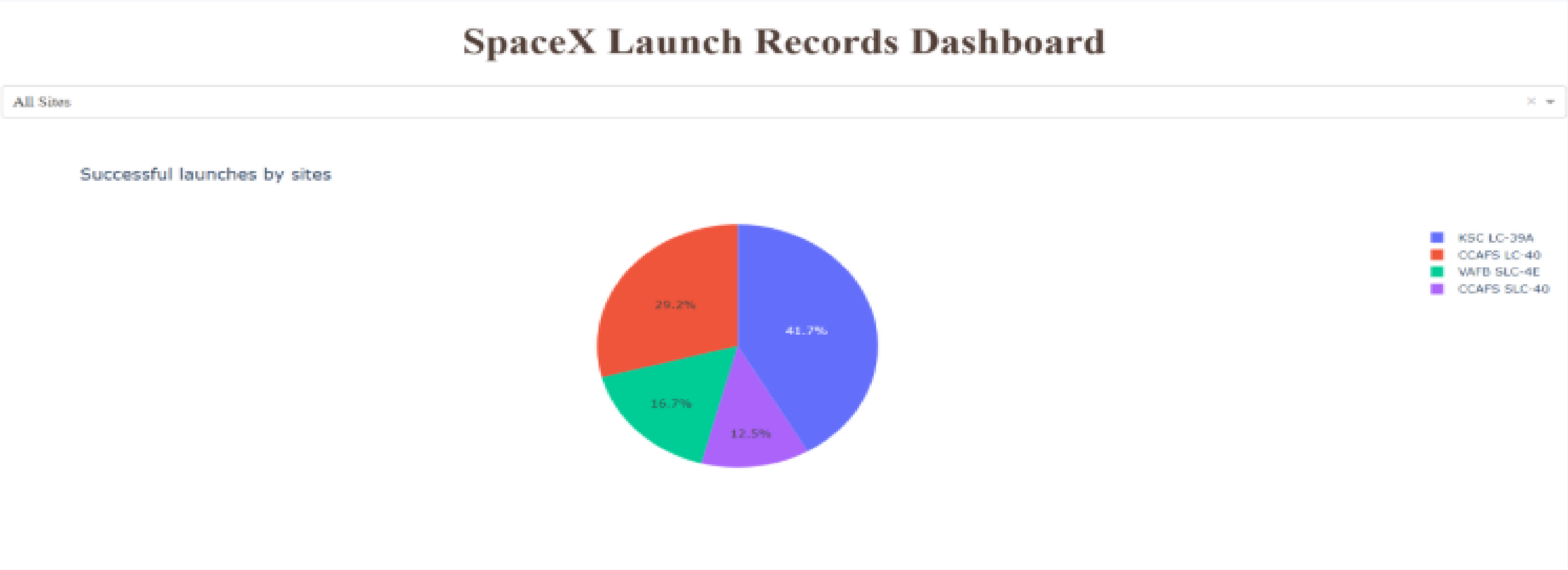
Section 4

# Build a Dashboard with Plotly Dash



# Pie Chart of Launch Success Count for All Sites

- KSC LC -39A Launch Site has most successful Launches while CCAFS SLC-40 has least successful launches of Falcon9.



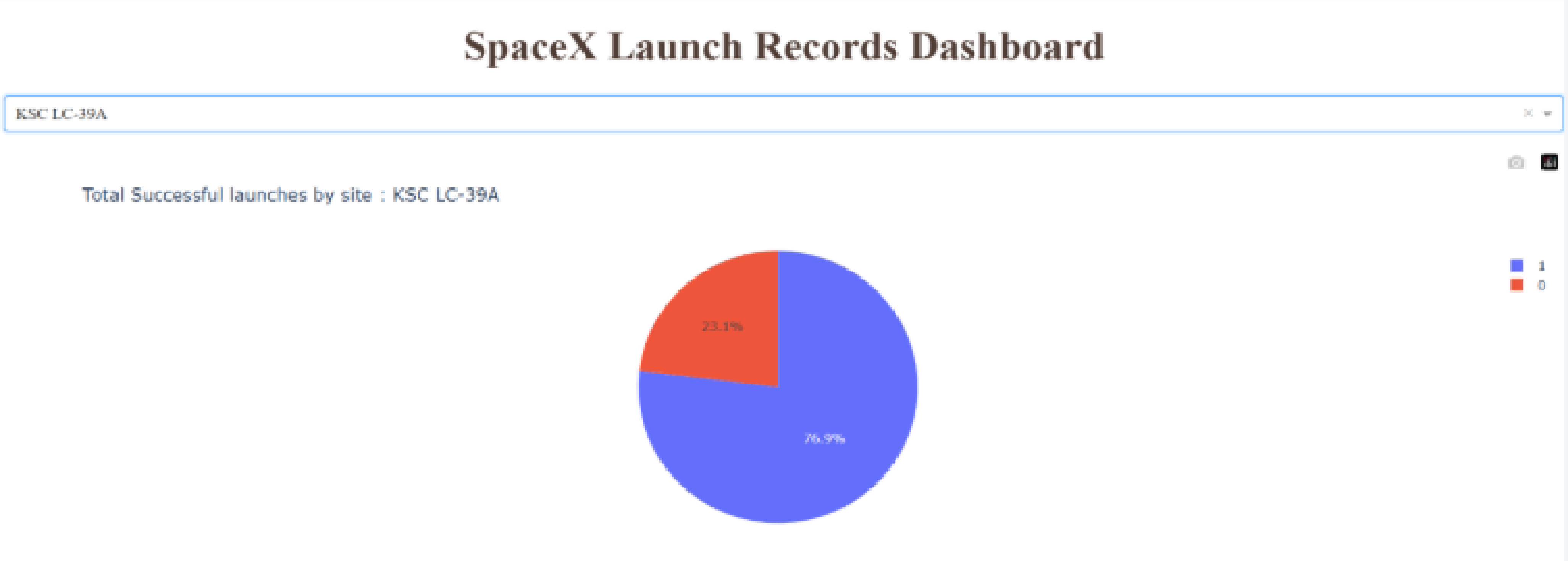
SpaceX Launch Records Dashboard

All Sites

Successful launches by sites

# Pie Chart of Lauch Site with Highest Lauch Success Rate

- KSC LC-39A has highest launch success rate with 76.9% of Success in Launching flights.

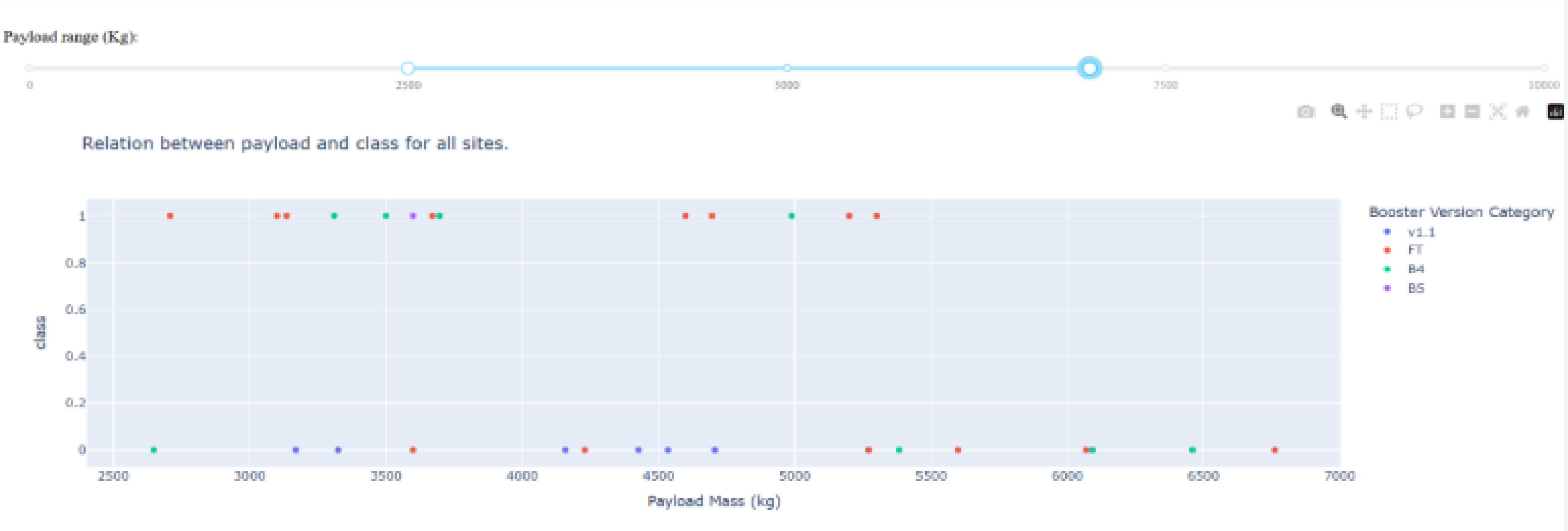


SpaceX Launch Records Dashboard

KSC LC-39A

# Scatter Plot of Success rate in Payload Mass range 2500kg to 7000kg

- **No clear correlation** is observed between **Payload Mass** and **Success Rate** within the **2500 to 7000 kg range**.

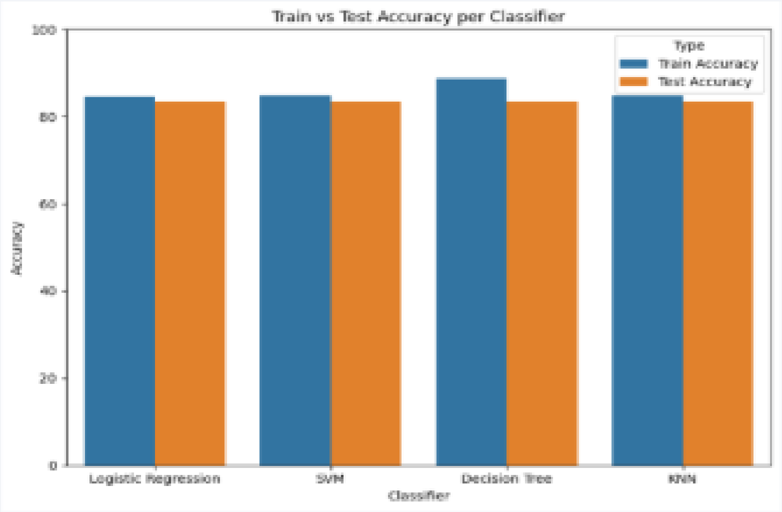


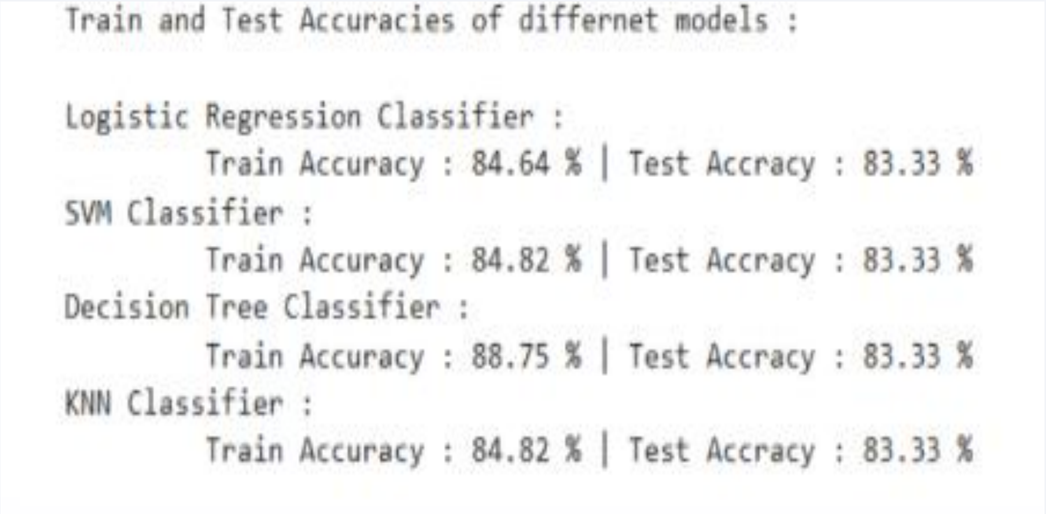
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- All four classification models achieved the same test accuracy of 83.33%.
- The Decision Tree Classifier showed the highest training accuracy at 88.75%, possibly due to overfitting, while the other models had training accuracies around 85%.
- Since the difference between training and test accuracy is minimal, the Decision Tree Classifier may be the best choice for classification.

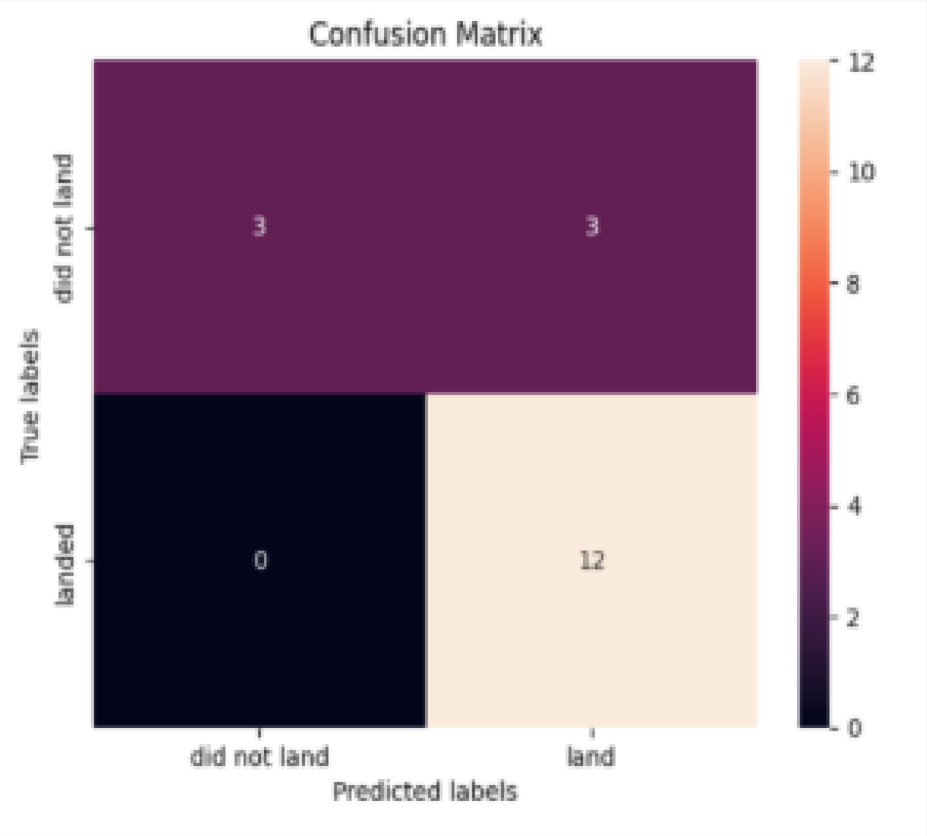




```
Train and Test Accuracies of differnet models :

Logistic Regression Classifier :
        Train Accuracy : 84.64 % | Test Accracy : 83.33 %
SVM Classifier :
        Train Accuracy : 84.82 % | Test Accracy : 83.33 %
Decision Tree Classifier :
        Train Accuracy : 88.75 % | Test Accracy : 83.33 %
KNN Classifier :
        Train Accuracy : 84.82 % | Test Accracy : 83.33 %
```

# Confusion Matrix

- The Decision Tree Classifier is the best model for predicting Falcon 9 first-stage landings.
- Although it has 3 false positives, which is not ideal, it still performs better than the other models.
- The model achieves a 100% true positive rate, with a recall of 100% and precision of 80% for successful landing predictions.
- It also has the highest training accuracy at 88.75% and a test accuracy of 83.33%.



# Conclusions

- This project aims to predict whether the first stage of a Falcon 9 launch will land successfully, helping to estimate the launch cost.
- Various features of a Falcon 9 launch, such as payload mass and orbit type, may influence the mission outcome.
- Multiple machine learning algorithms were used to analyze historical Falcon 9 launch data and build predictive models.
- Among the four algorithms tested, the decision tree model showed the best performance in predicting launch outcomes.

# Appendix

- L1_SpaceX_Data_Collection_using_API.ipynb
- L2_SpaceX_Data_Collection_using_W
- L3_SpaceX_Data_Wrangling.ipynb
- L4_SpaceX_EDA_with_SQL_(sqlite3).ipynb
- L5_SpaceX_EDA_with_Visualization.ip
- L6_SpaceX_Visual_Analytics_with_F
- L7_SpaceX_App_Dashboard_using_
- L8_SpaceX_Prediction_Analysis_(ML

# Thank you!