IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abdulrahman Almulhim

21/08/2024

Outline

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

Executive Summary

In this project, we successfully predicted the Falcon 9 first stage landing outcome with 94% accuracy. The process involved data collection through APIs, web scraping, and SQL, followed by data wrangling and EDA. We developed an interactive dashboard using Plotly Dash and Folium to visualize Falcon 9 landing success rates and locations on a map. After standardizing and splitting the data, we tested multiple machine learning algorithms, including Logistic Regression, SVM, Decision Tree, and KNN Classifiers, optimizing each with GridSearchCV to get the best hyperparameters for each ML algorithm. The ML algorithm that produced the highest prediction accuracy was the Decision Tree Classifier, which predicted 17 out of 18 outcomes correctly.

Introduction

Project Background and Context:

- The Falcon 9 is a partially reusable rocket designed by SpaceX, with the first stage responsible for boosting the rocket into orbit and attempting to land back on Earth.
- Predicting the success of these landings is crucial for enhancing reusability and reducing costs in space missions.

Problems to Solve:

- Can we predict if the Falcon 9 first stage will land successfully?
- What factors influence the success rate of these landings?
- How can we visualize and analyze the data to provide actionable insights for future missions?

Section 1

Methodology

Methodology

Executive Summary

1. Data Collection

- **Describe** how data was gathered from various sources.

2. Data Wrangling

- **Process** used to clean and prepare the data for analysis.

3. Exploratory Data Analysis (EDA)

- **Utilize** visualization techniques and SQL to explore the data.

4. Interactive Visual Analytics

- **Tools**: Used Folium and Plotly Dash to create interactive visualizations.

5. Predictive Analysis

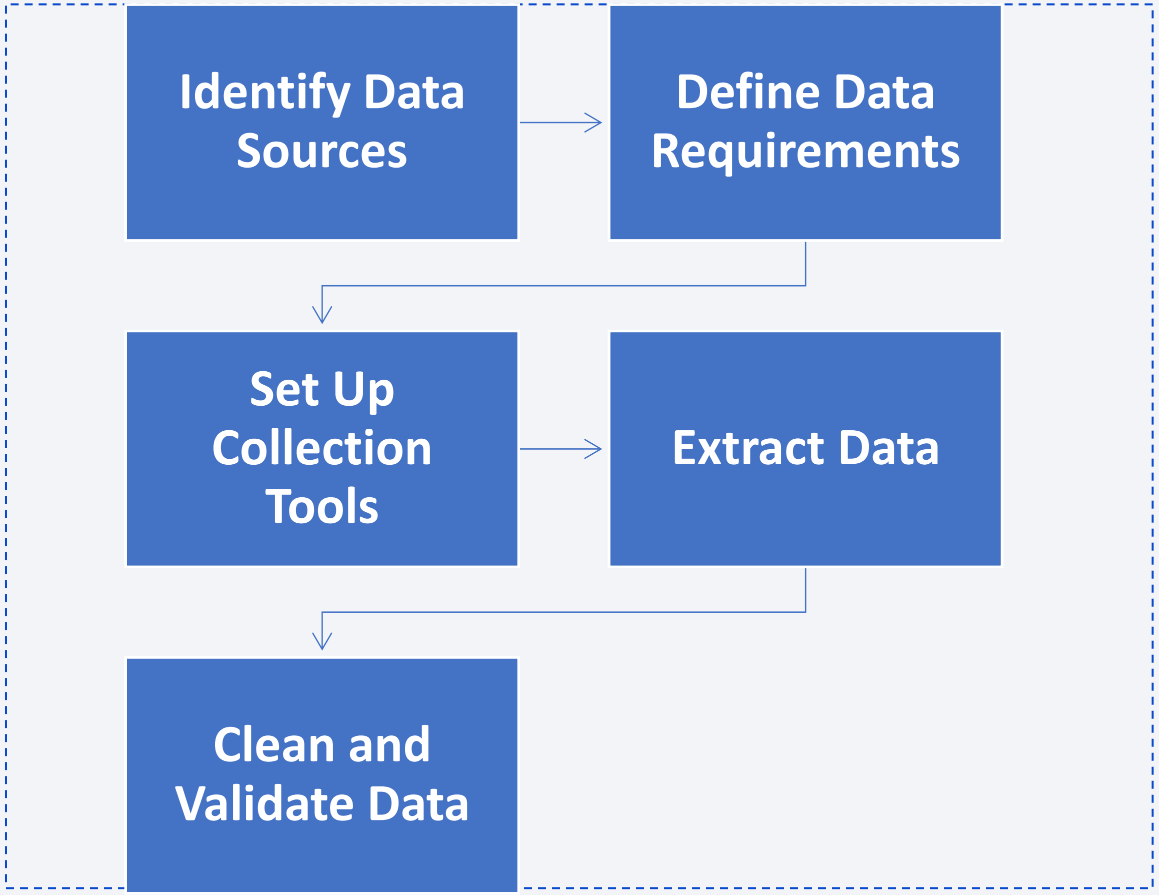
- **Modelling**: Steps to build, tune, and evaluate classification models.

Data Collection

- **Identify Data Sources:** Selected relevant APIs and databases specific to the project (e.g., SpaceX launch data from NASA API).
- **Define Data Requirements:** Identified key variables like launch site, payload, and outcome.
- **Set Up Collection Tools:** Utilized Python scripts and API calls to extract the data.
- **Extract Data:** Retrieved datasets directly from APIs and stored them in a structured format.
- **Clean and Validate Data:** Performed data wrangling to remove inconsistencies and ensure data integrity.

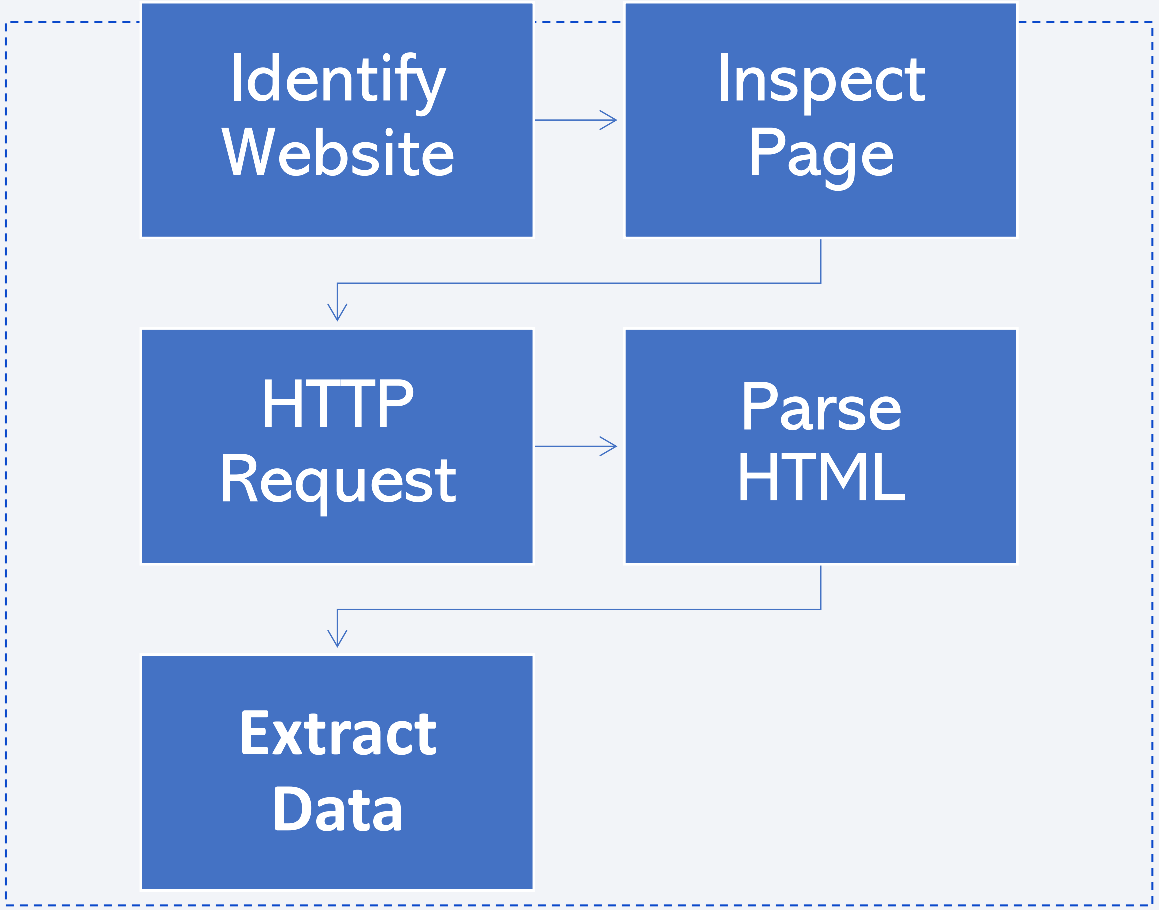
Data Collection – SpaceX API

- **Data Sources:** SpaceX REST API.
- **Data Requirements:** Launch details, rocket information, and mission outcomes.
- **Collection Tools:** Python script to fetch data from the SpaceX API.
- **Extract Data:** Run the script to gather data.
- **Clean and Validate Data:** Ensure data completeness and accuracy.
- https://github.com/Da7mMulhim/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



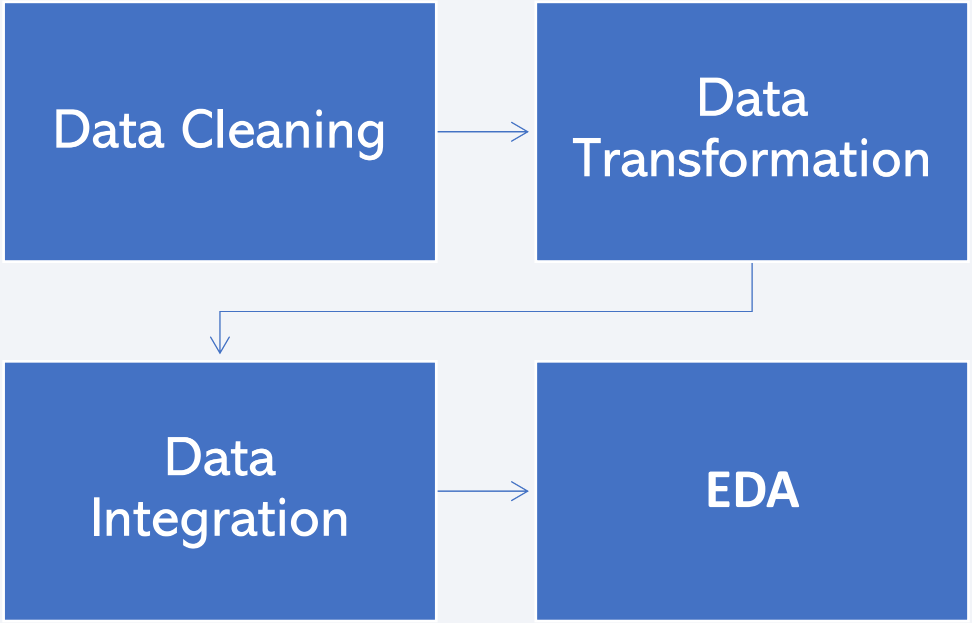
Data Collection - Scraping

- **Identify Website:** Determine the target website.
- **Inspect Page:** Use browser tools to inspect HTML elements.
- **HTTP Request:** Use requests to get HTML content.
- **Parse HTML**: Use BeautifulSoup to parse HTML.
- **Extract Data**: Extract data using methods like find or CSS selectors.
- https://github.com/Da7mMulhim/testrepo/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- **Data Cleaning:** Handle missing values, remove duplicates, correct inconsistencies.
- **Data Transformation:** Normalize, scale, encode categorical variables.
- **Data Integration:** Combine data from multiple sources.
- **Exploratory Data Analysis (EDA):** Summary statistics, graphical representations.
- https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-0a5f0868fc44ddd900af481256bedb9585478a3a



EDA with Data Visualization

- **Histograms:** Used to visualize numerical variable distribution..
- **Box Plots:** Used to show data distribution
- **Scatter Plots:** Used to examine relationships.
- **Bar Charts:** Used to compare categorical data.
- **Heatmaps:** Used to show magnitude using color.
- https://github.com/Da7mMulhim/testrepo/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- **Select Query:** Retrieves all data from a table.
- **Filtered Select:** Retrieves specific data based on a condition.
- **Join Query:** Combines rows from related tables.
- **Group By:** Groups data and provides summary.
- **Order By:** Sorts data in ascending or descending order.
- **Aggregate Functions:** Performs calculations on data.
- **Subquery:** Filters data using a nested query.
- https://github.com/Da7mMulhim/testrepo/blob/main/jupyter-labs-eda-sql-edx_sqllite.ipynb

Build an Interactive Map with Folium

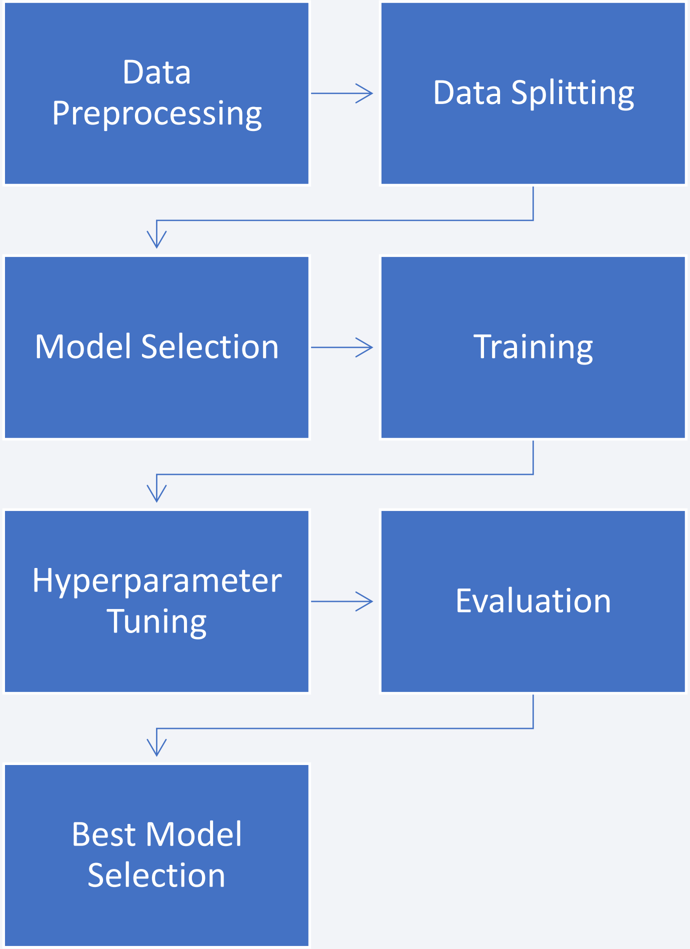
- **Markers:** Pinpoint specific locations with additional info.
- **Circles:** Represent areas around a point.
- **Circle Markers:** Highlight important points.
- **Polylines:** Visualize paths or connections.
- **Polygons:** Highlight specific areas.
- **Choropleth Maps:** Represent data patterns across regions.
- https://github.com/Da7mMulhim/testrepo/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- **Launch Success Pie Chart:** Visualize the success rate of SpaceX launches.
- **Payload vs. Launch Outcome Scatter Plot:** Show the relationship between payload mass and launch outcomes.
- **Launch Sites Success Bar Chart:** Compare the success rates of different launch sites.
- **Yearly Launch Trends Line Chart:** Display the number of launches per year to observe trends.
- **Launch Site Dropdown Filter:** Filter data by specific launch sites for detailed analysis.
- **Payload Range Slider:** Explore the impact of different payload masses on launch outcomes.
- **Year Range Slider:** Analyze trends over specific time periods.
- https://github.com/Da7mMulhim/testrepo/blob/main/spacex_dash_app.py

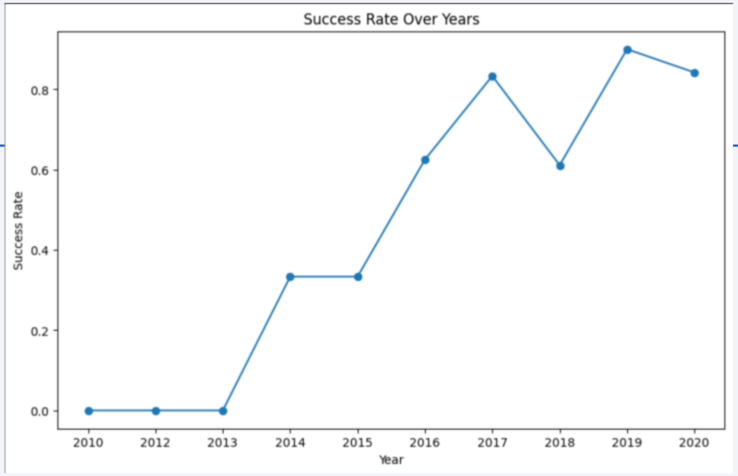
Predictive Analysis (Classification)

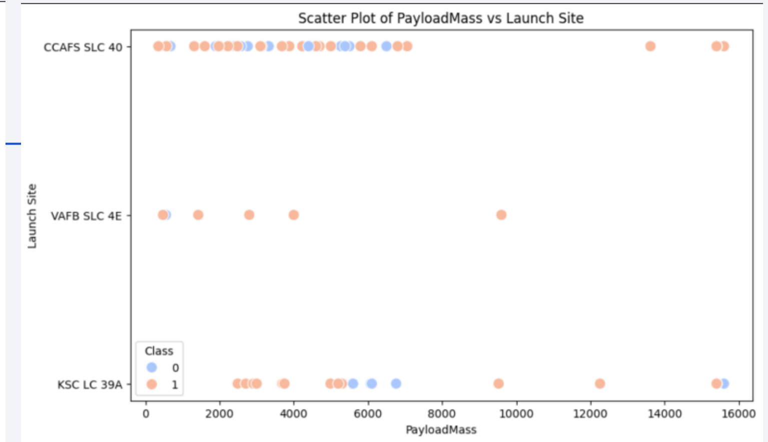
- **Data Preprocessing:** Clean, normalize, handle missing values
- **Data Splitting:** Train/test split
- **Model Selection:** Choose classification algorithms (e.g., Decision Tree, SVM)
- **Training:** Train models on training data
- **Hyperparameter Tuning:** Grid search, random search, cross-validation
- **Evaluation:** Using score method
- **Best Model Selection:** Compare performance metrics, select best model
- https://github.com/Da7mMulhim/testrepo/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

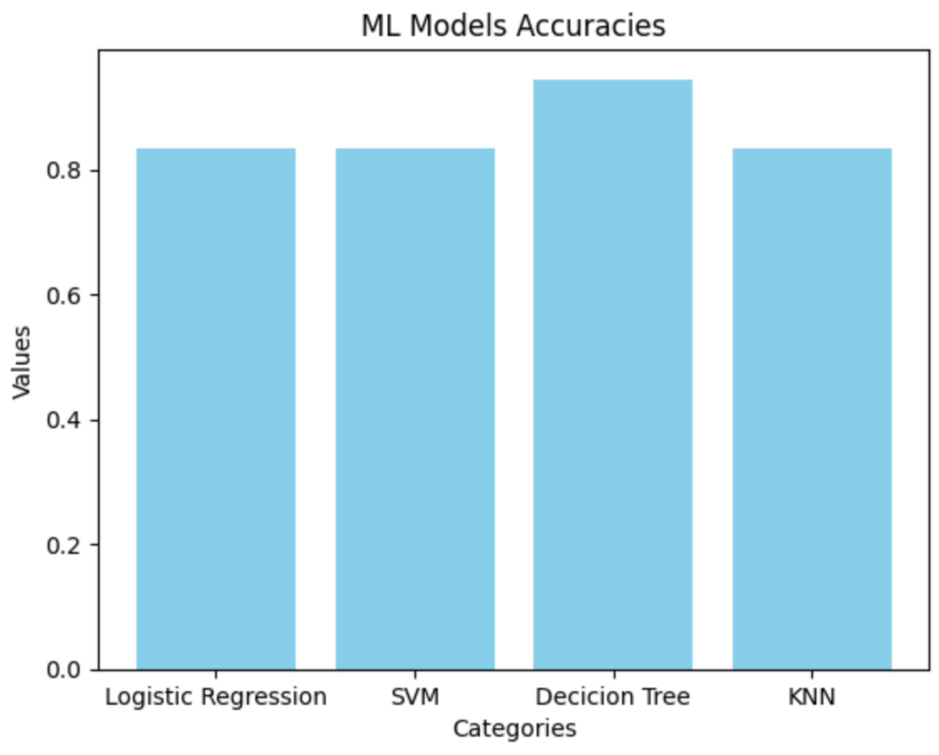


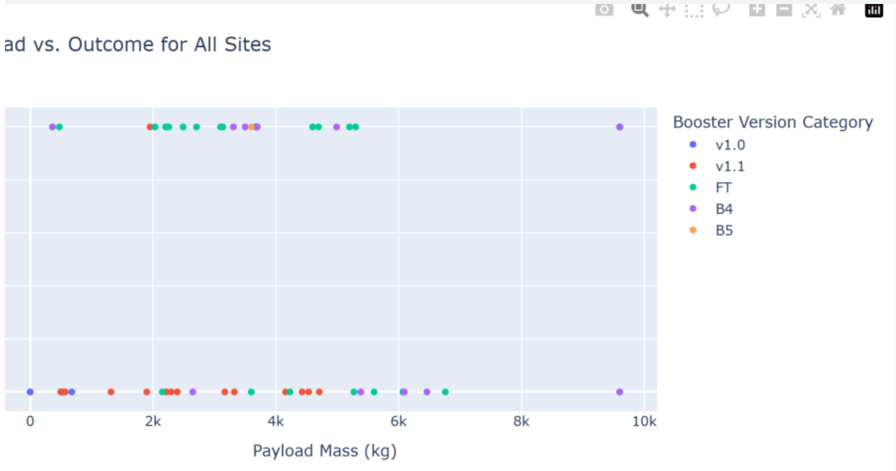
Results

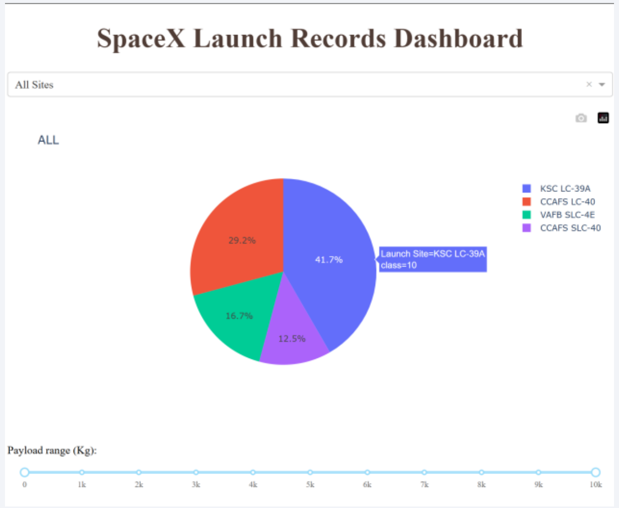
- Exploratory data analysis results
- Interactive analytics demo
- Predictive analysis results










SpaceX Launch Records Dashboard

All Sites

Payload range (Kg):

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

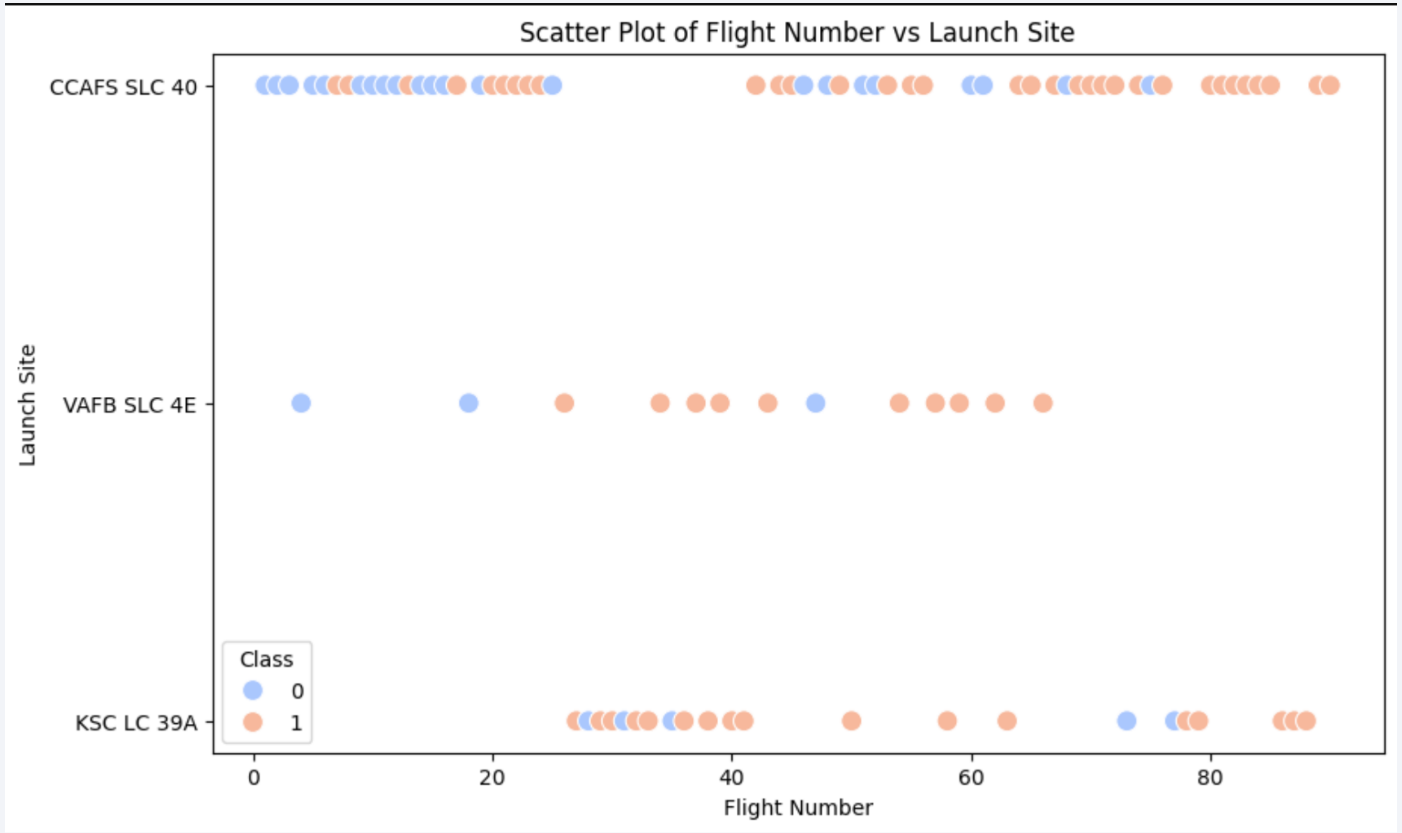


Chart: Displays the relationship between flight number and launch site, categorized by landing success.
Class Representation: Orange (Success), Blue (Failure).
Key Insights:
•**CCAFS SLC-40**: High number of flights with mixed success.
•**KSC LC-39A**: Increasing success with higher flight numbers.
•**VAFB SLC-4E**: showing balanced success and failure.

Payload vs. Launch Site

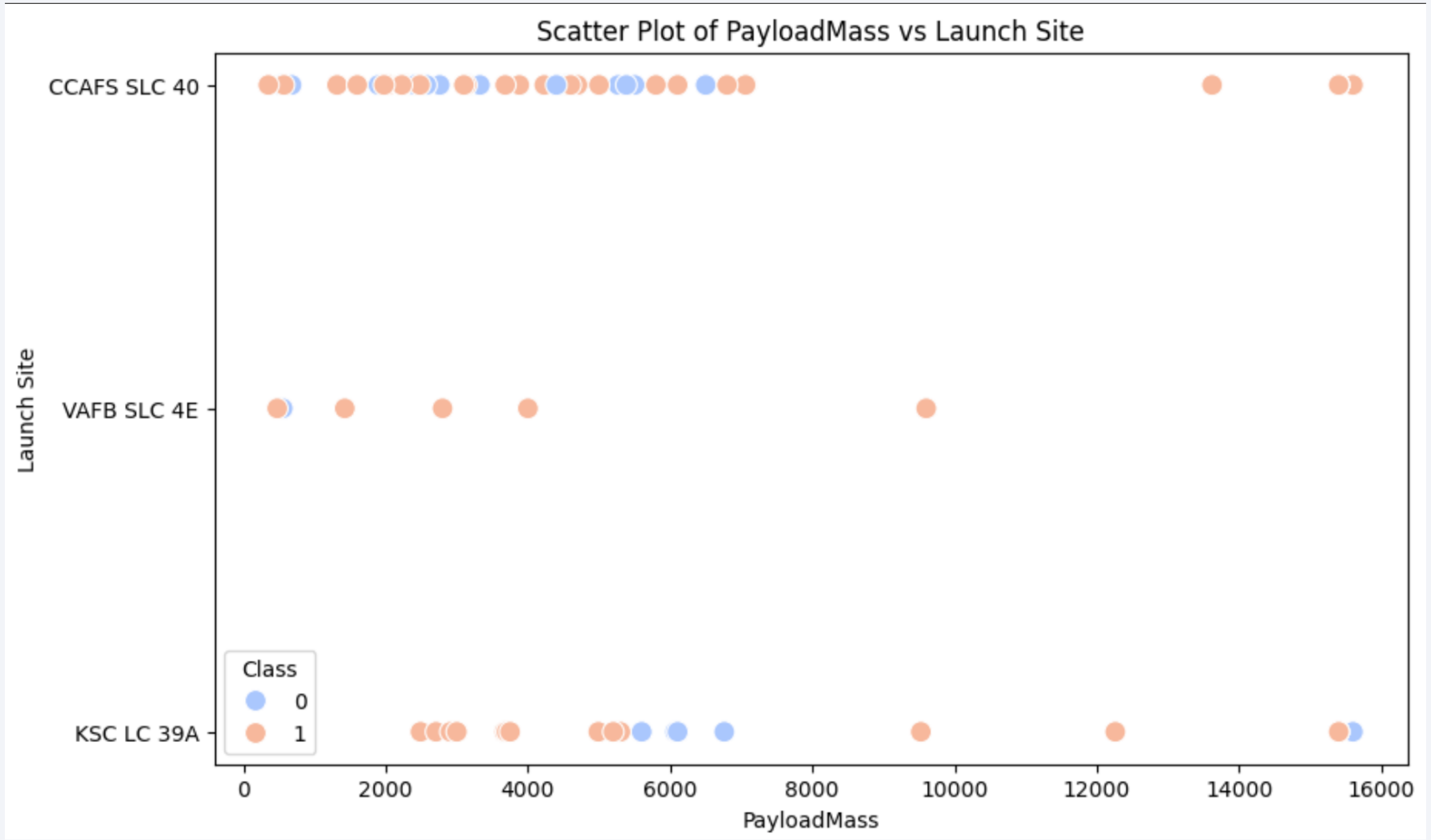


Chart: Displays the relationship between payload mass and launch site, categorized by landing success.
Class Representation: Orange (Success), Blue (Failure).
Key Insights:
•**CCAFS SLC-40**: Shows a variety of payloads with mixed success.
•**KSC LC-39A**: Handles a wider range of payloads with consistent success across different masses.
•**VAFB SLC-4E**: lower payloads, mostly successful.

Success Rate vs. Orbit Type

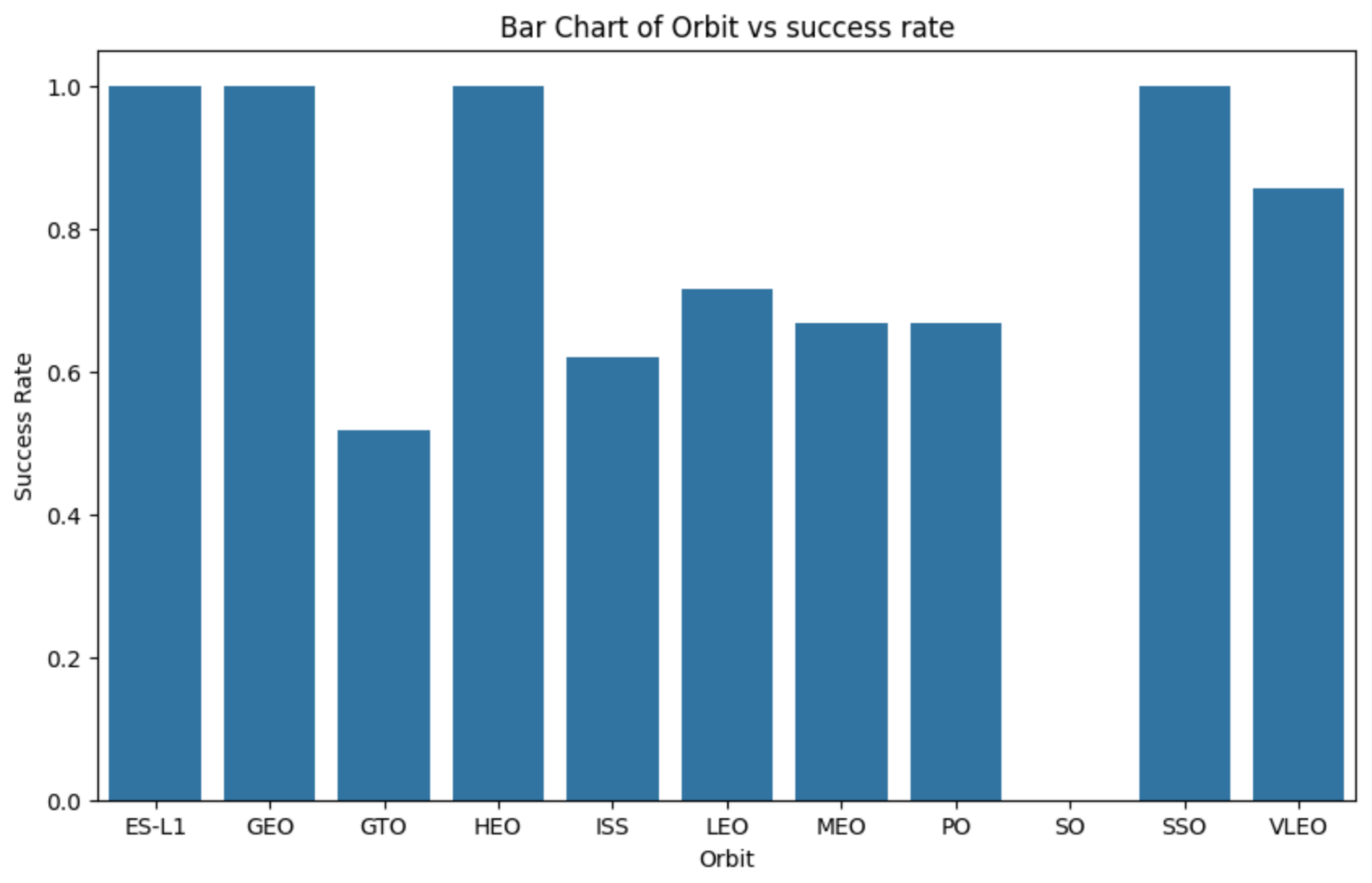


Chart: Shows success rate of Falcon 9 landings across various orbits.

Key Insights:

•**Highest Success**: ES-L1, GEO, LEO, SSO, and VLEO with a 100% success rate.

•**Moderate Success**: GTO, HEO, and PO orbits have varied success, with GTO being the lowest.

•**Implications**: Success varies significantly depending on the target orbit

Flight Number vs. Orbit Type

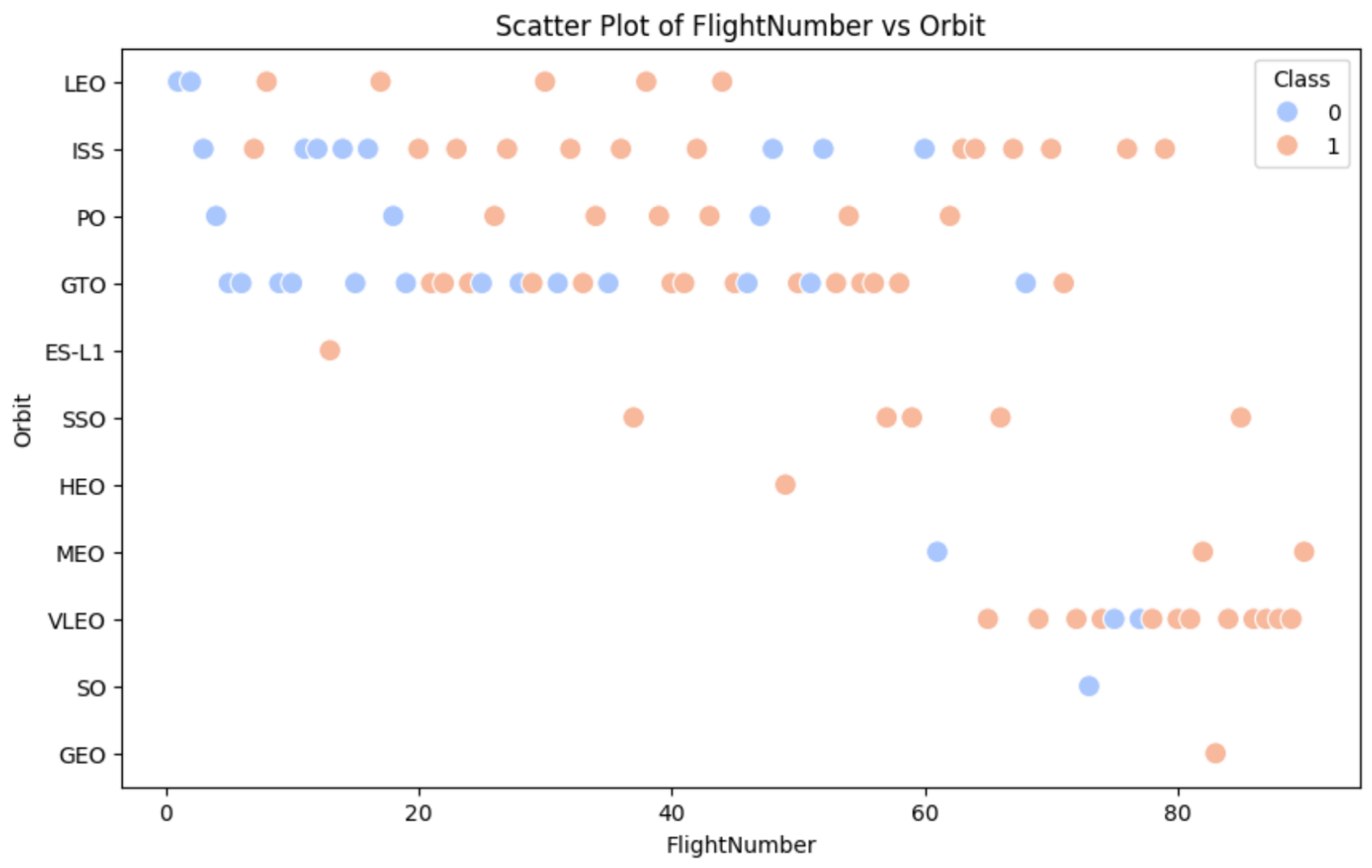


Chart: Shows the relationship between the flight number of Falcon 9 launches and the corresponding orbit types.

Key Insights:

•The ISS has a steady distribution of success and failure to land, reflecting consistent space station supply missions.

•SSO has few flights, but always successfully lands

•ES-L1, GEO, HEO, and SO are rarer, indicating specialized or less frequent missions.

Payload vs. Orbit Type

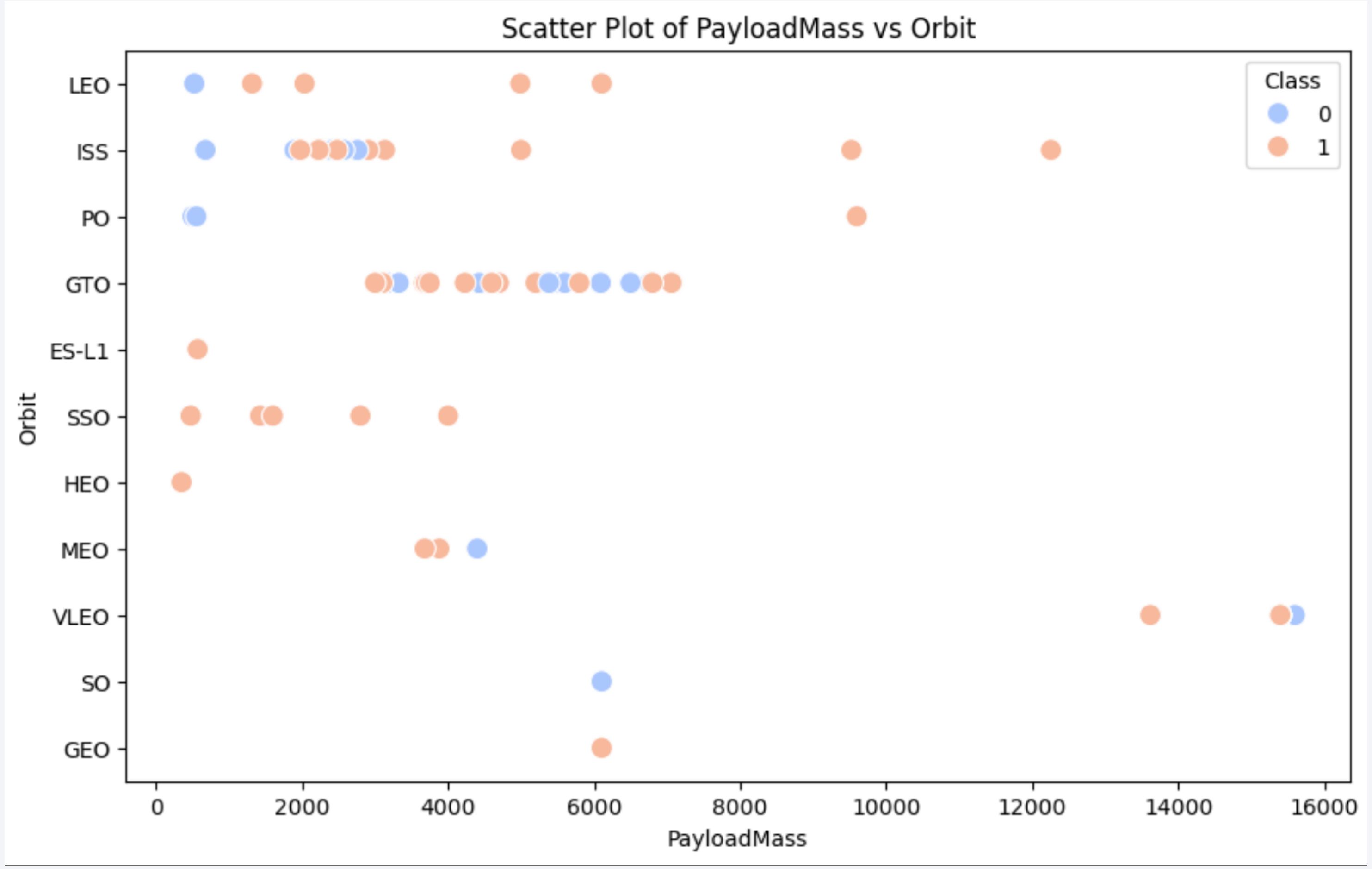


Chart: The chart visualizes payload mass distribution across different orbit types to identify patterns

Key Insights:

•VLEO has the highest payloads with successful outcomes.

•GTO has the highest number of occurrences with payloads predominantly between 2,500 and 8,000 kg, reflecting its common use for medium to large payloads.

•ISS shows high success rates across a wide range of payloads.

•SSO consistently has a higher success rate with smaller payloads.

Launch Success Yearly Trend

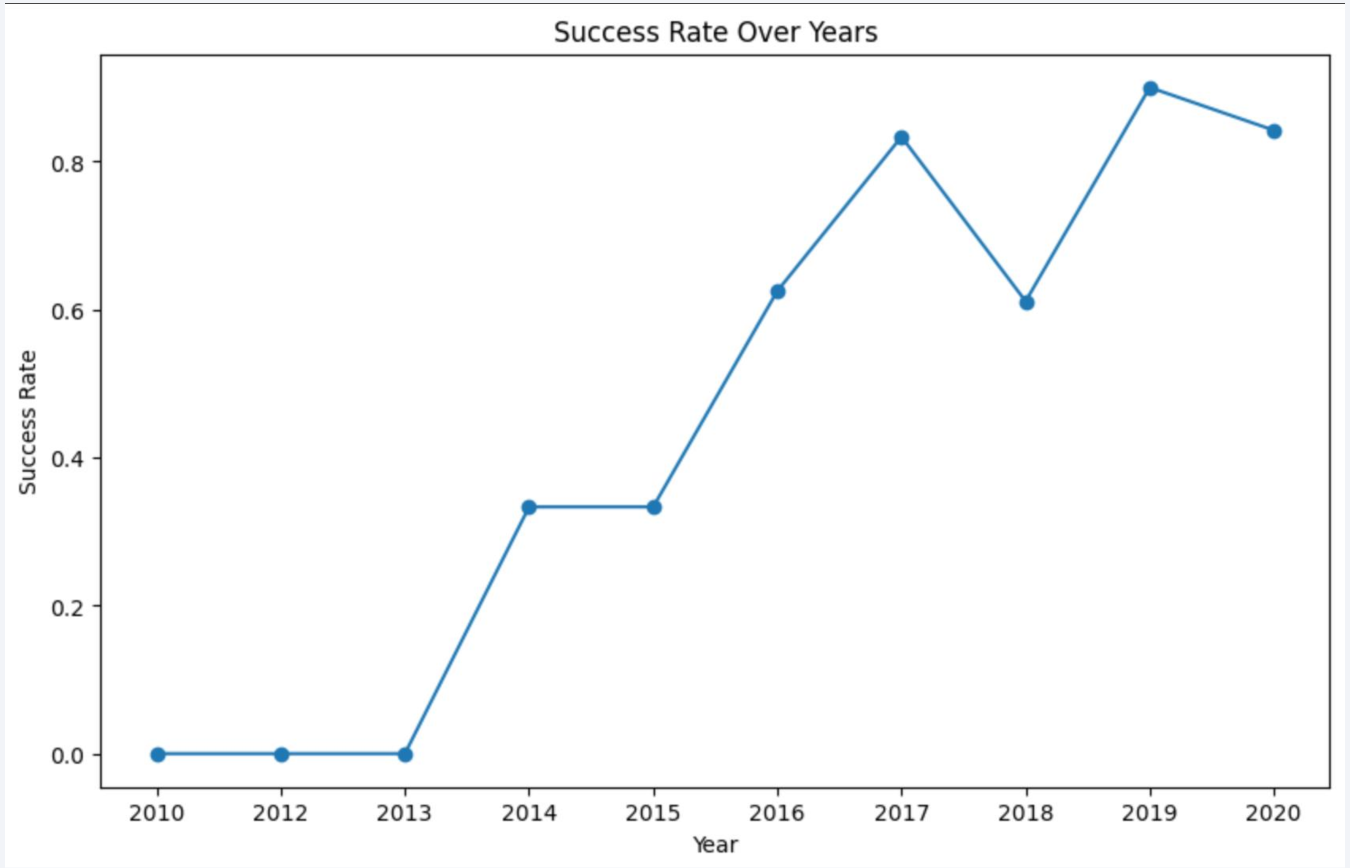
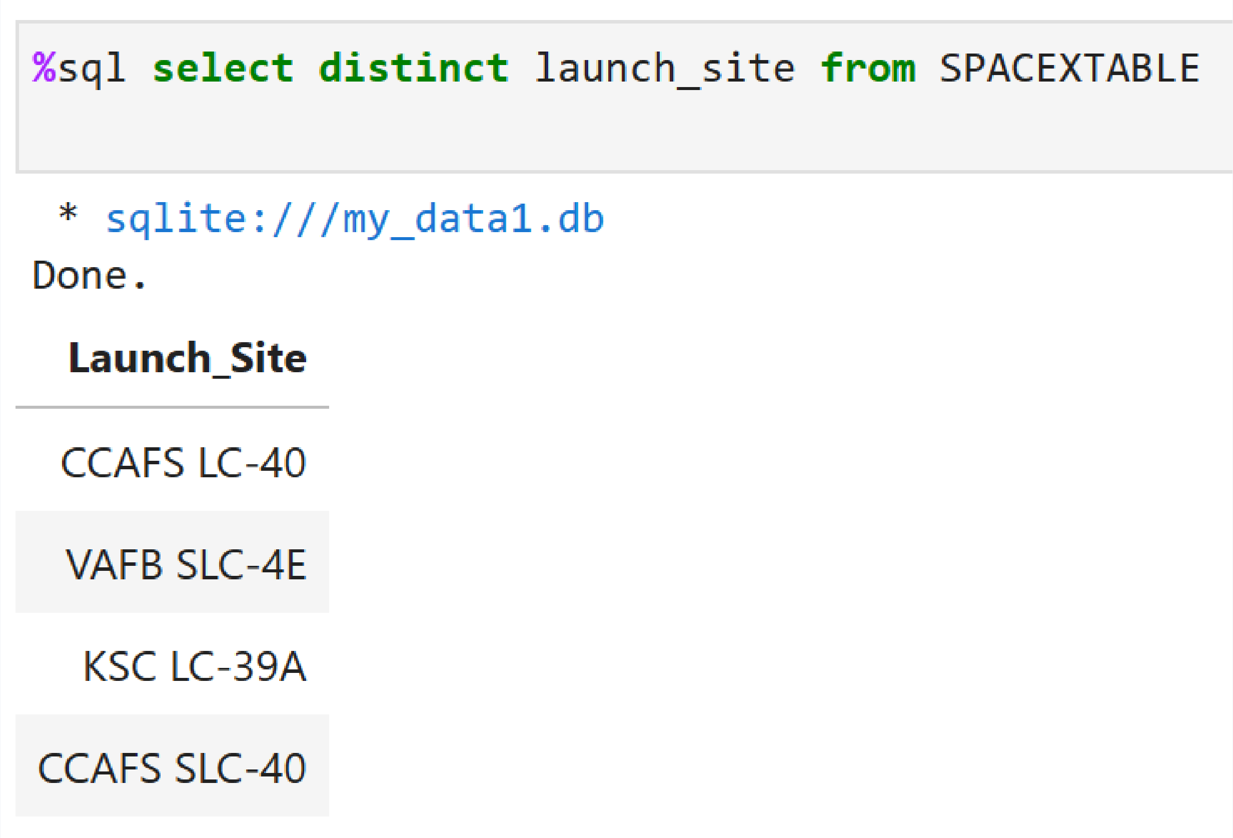


Chart: The line chart tracks the yearly average success rate to assess trends and changes in success rates over time.

Key Insights:

- Overall trend shows a positive growth in success rates over the years.
- There was a notable improvement from 2015 (0.33) to 2016 (0.63).
- The highest success rate was in 2019 at 0.90.
- A slight decrease in success rate is observed in 2020 compared to 2019.

All Launch Site Names



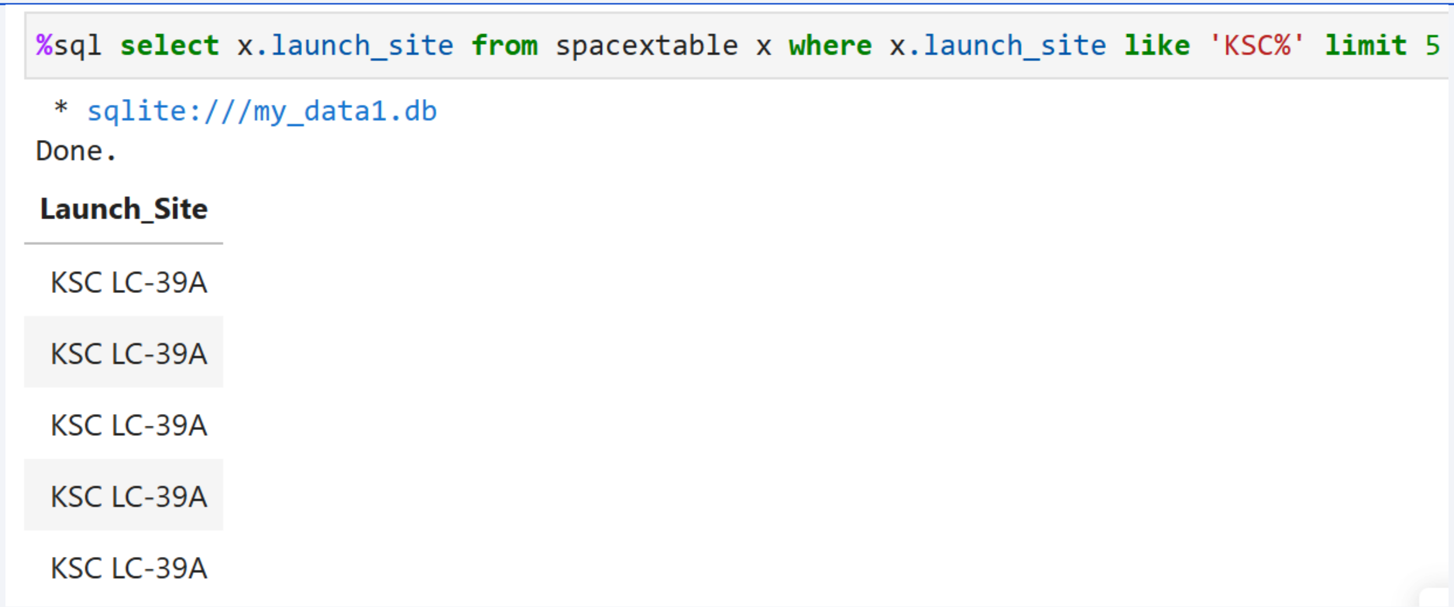
```
%sql select distinct launch_site from SPACEXTABLE
```

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

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

- The query uses SELECT DISTINCT to retrieve only unique launch site names.
- There are a total of 4 distinct launch sites in the results.

Launch Site Names Begin with 'KSC'



```
%sql select x.launch_site from spacextable x where x.launch_site like 'KSC%' limit 5
```

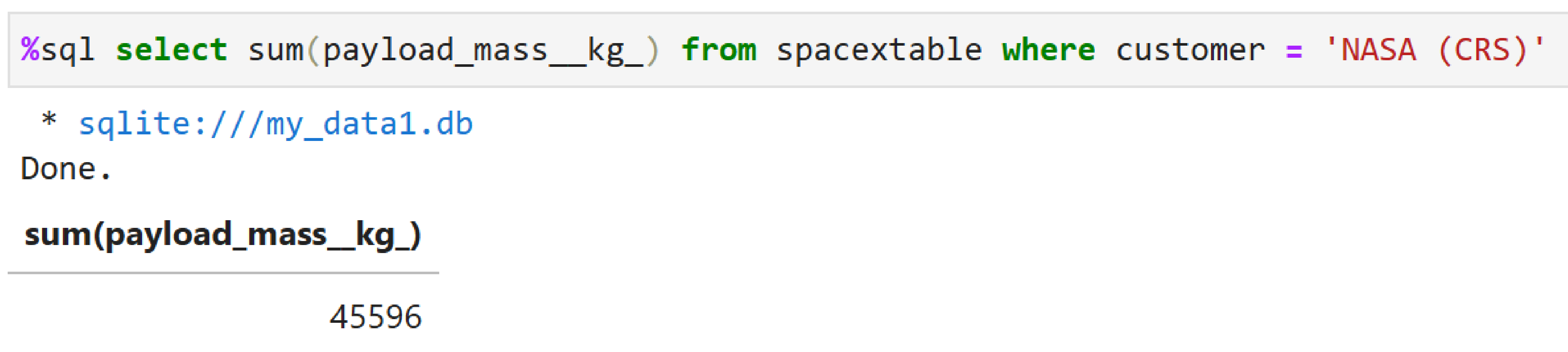
```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
KSC LC-39A
KSC LC-39A
KSC LC-39A
KSC LC-39A
KSC LC-39A

- **SELECT:** to retrieve specific columns from the database.
- **WHERE:** to filter records based on certain conditions.
- **LIKE:** to match string provided.
- **$:** in the LIKE clause indicates a placeholder for a string pattern, allowing dynamic matching of strings that start with the specified pattern.
- **LIMIT 5:** to limit the results to 5 records.

•It includes a WHERE clause to filter records based on a specific customer.

Total Payload Mass



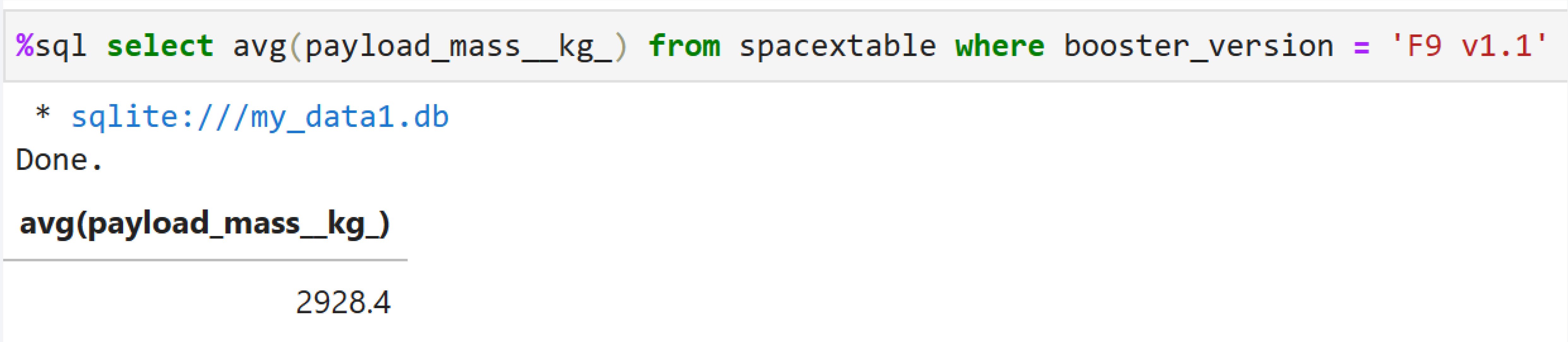
```
%sql select sum(payload_mass__kg_) from spacextable where customer = 'NASA (CRS)'
```

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

sum(payload_mass__kg_)
45596

- **SUM:** to calculate the total payload mass.
- **WHERE:** to filter records based on a specific customer.

Average Payload Mass by F9 v1.1



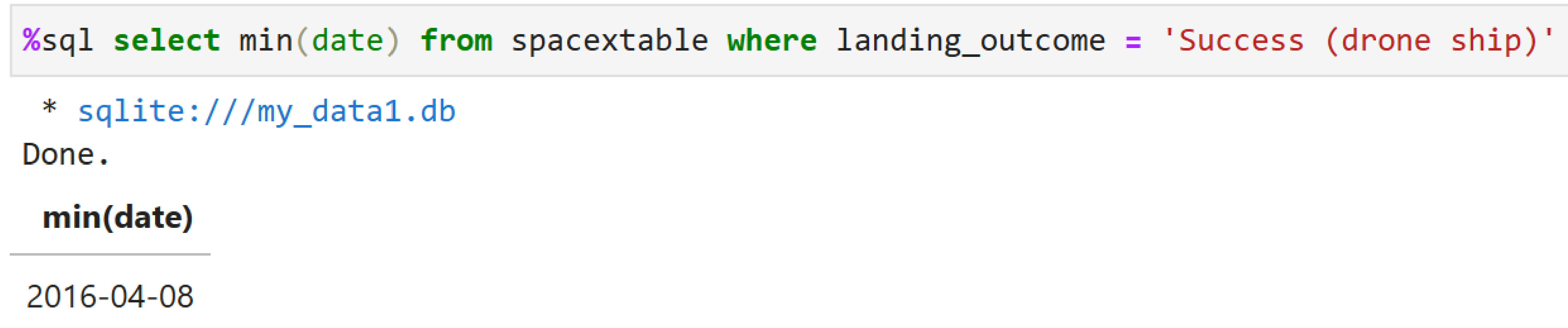
```
%sql select avg(payload_mass__kg_) from spacextable where booster_version = 'F9 v1.1'
```

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

avg(payload_mass__kg_)
2928.4

- **SELECT AVG:** to calculate the average payload mass from the spacex table.
- **WHERE:** to filter records where the booster_version is 'F9 v1.1'.

First Successful Ground Landing Date



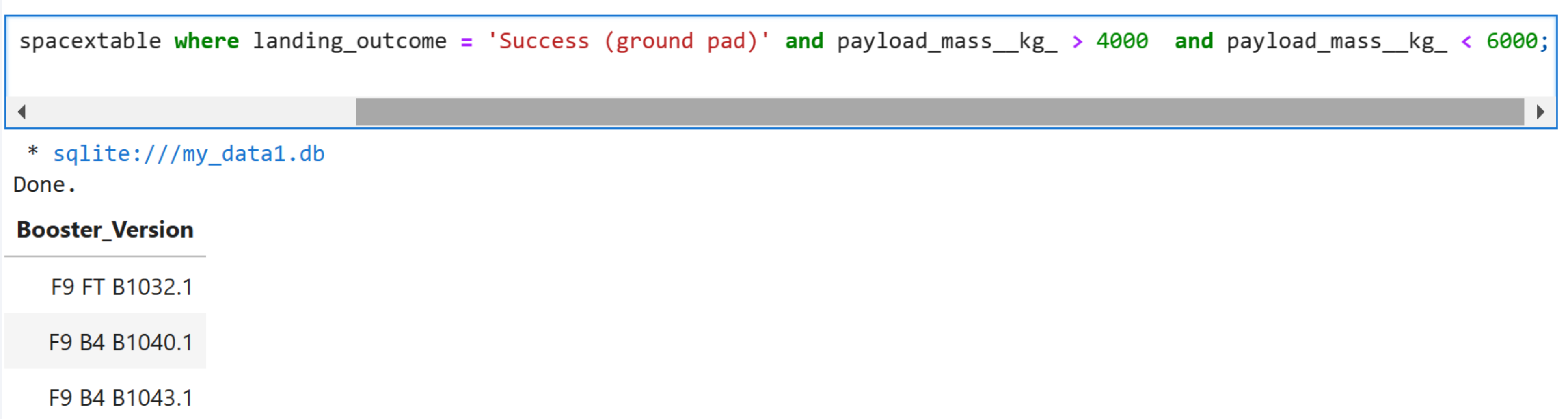
```
%sql select min(date) from spacextable where landing_outcome = 'Success (drone ship)'
```

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

min(date)
2016-04-08

- **SELECT MIN:** to find the earliest date from the spacex table.
- **WHERE:** clause to filter records where the landing_outcome is 'Success (drone ship)'.

Successful Drone Ship Landing with Payload between 4000 and 6000



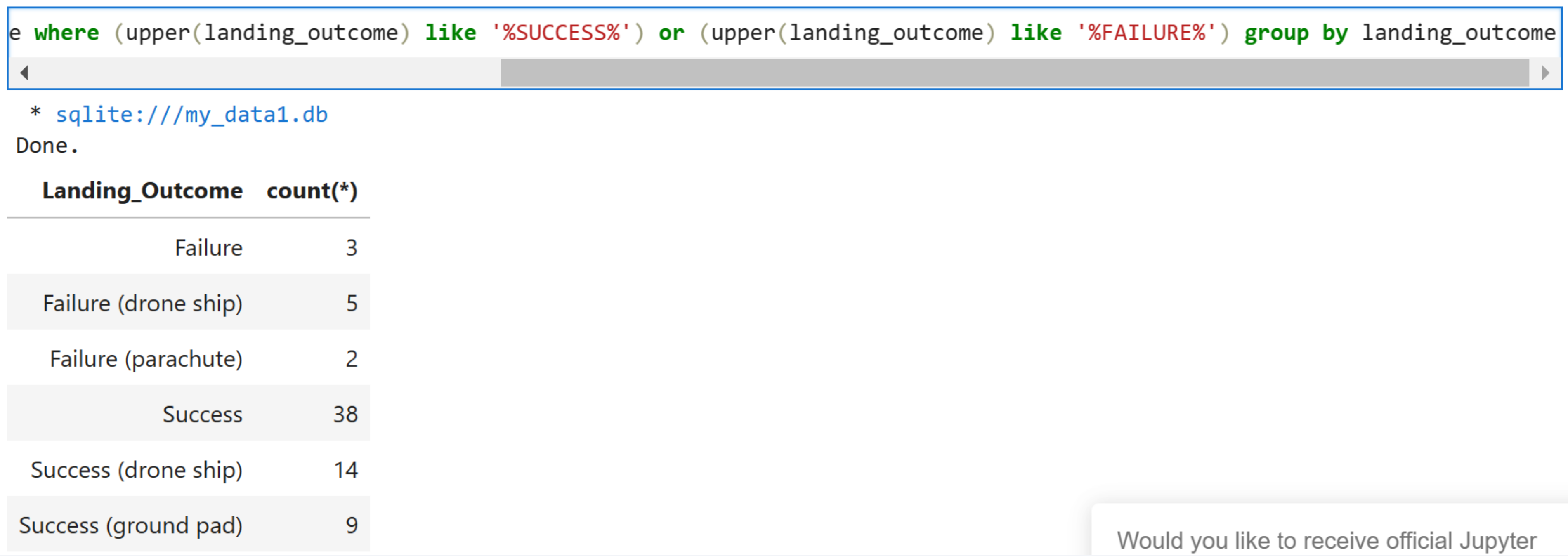
```
spacextable where landing_outcome = 'Success (ground pad)' and payload_mass__kg_ > 4000  and payload_mass__kg_ < 6000;
```

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

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

- **SELECT:** to retrieve the booster_version from the spacex table.
- **WHERE:** to filter records with a landing_outcome of 'Success (ground pad)’.
- It further filters records where payload_mass__kg_ is between 4,000 and 6,000 kg.

Total Number of Successful and Failure Mission Outcomes



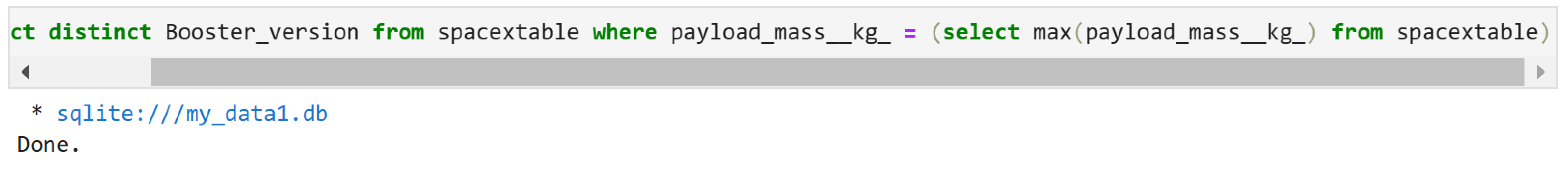
```
e where (upper(landing_outcome) like '%SUCCESS%') or (upper(landing_outcome) like '%FAILURE%') group by landing_outcome
```

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

Landing_Outcome	count(*)
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

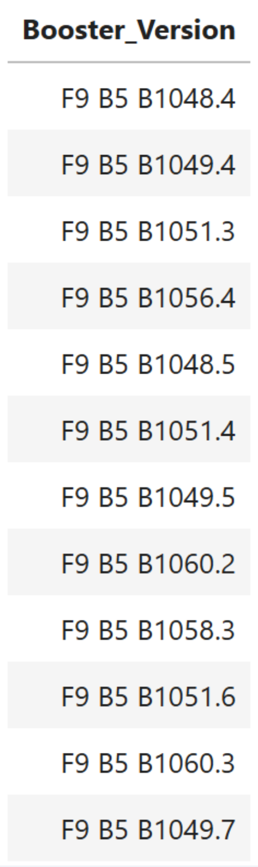
- **SELECT:** to retrieve landing_outcome and the count of records for each outcome from the spacex table.
- **WHERE:** clause to filter records where landing_outcome contains 'SUCCESS' or 'FAILURE', ignoring case.
- **GROUP BY:** To group the results by landing_outcome and count occurrences for each distinct outcome.

Boosters Carried Maximum Payload



```
ct distinct Booster_version from spacextable where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextable)
```

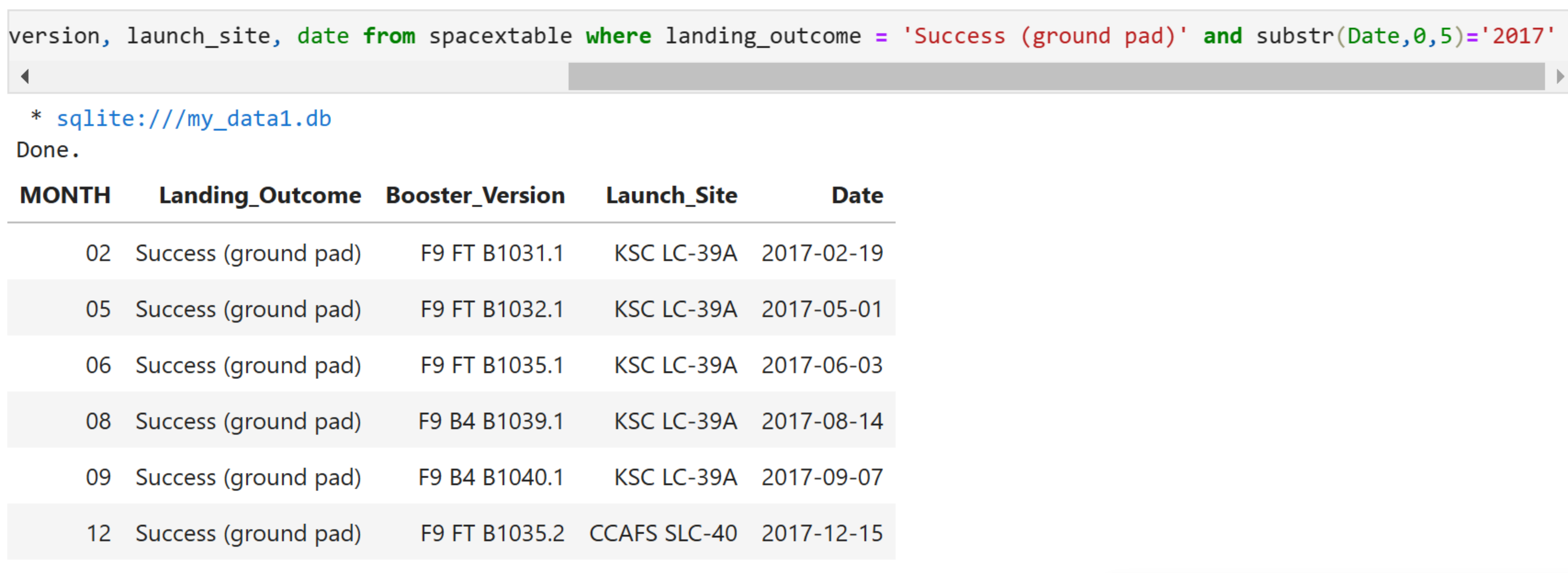
```
 * sqlite:///my_data1.db
Done.
```



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

- **SELECT DISTINCT:** to retrieve unique Booster_version values from the spacex table.
- **WHERE:** to filter records where payload_mass__kg_ is equal to the maximum payload mass found in the spacex table.
- **nested subquery:** to find the maximum payload_mass__kg_ from the spacextable, which is then used to filter the main query results.

2017 Launch Records



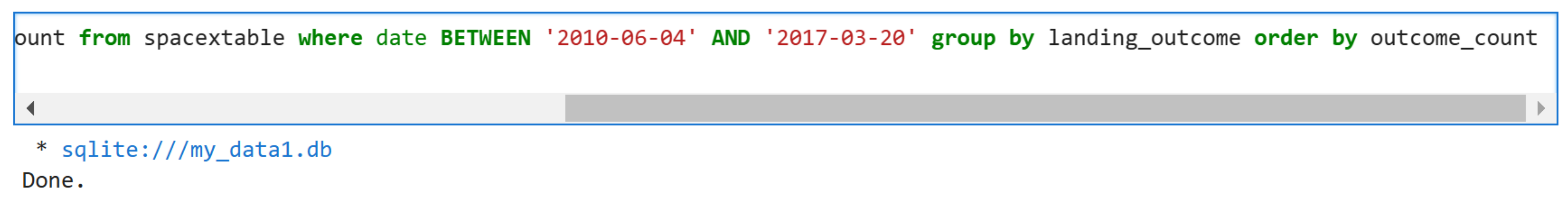
```
version, launch_site, date from spacextable where landing_outcome = 'Success (ground pad)' and substr(Date,0,5)='2017'
```

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

MONTH	Landing_Outcome	Booster_Version	Launch_Site	Date
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A	2017-02-19
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A	2017-05-01
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A	2017-06-03
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A	2017-08-14
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A	2017-09-07
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40	2017-12-15

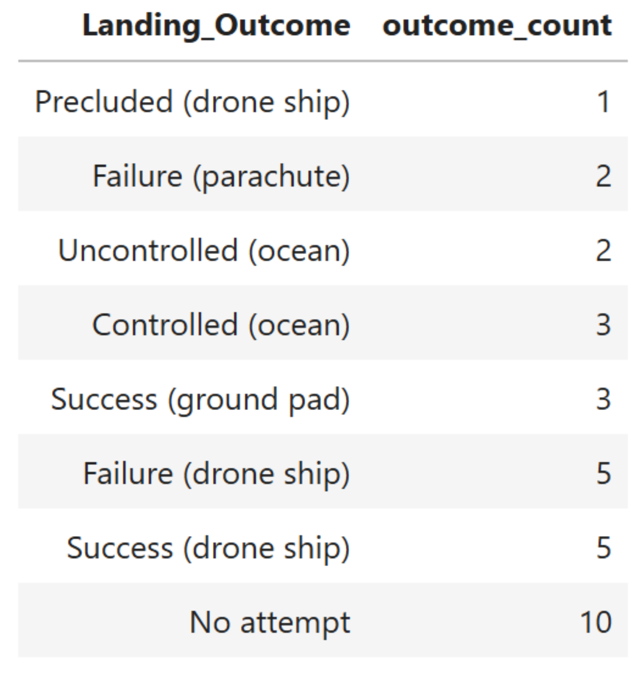
- **SELECT substr(Date, 6, 2) AS MONTH:** Extracts the month from Date.
- **WHERE:** To filter for successful ground pad landings.
- AND substr(Date, 1, 4) = '2017': Filters for the year 2017.

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



```
ount from spacextable where date BETWEEN '2010-06-04' AND '2017-03-20' group by landing_outcome order by outcome_count
```

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



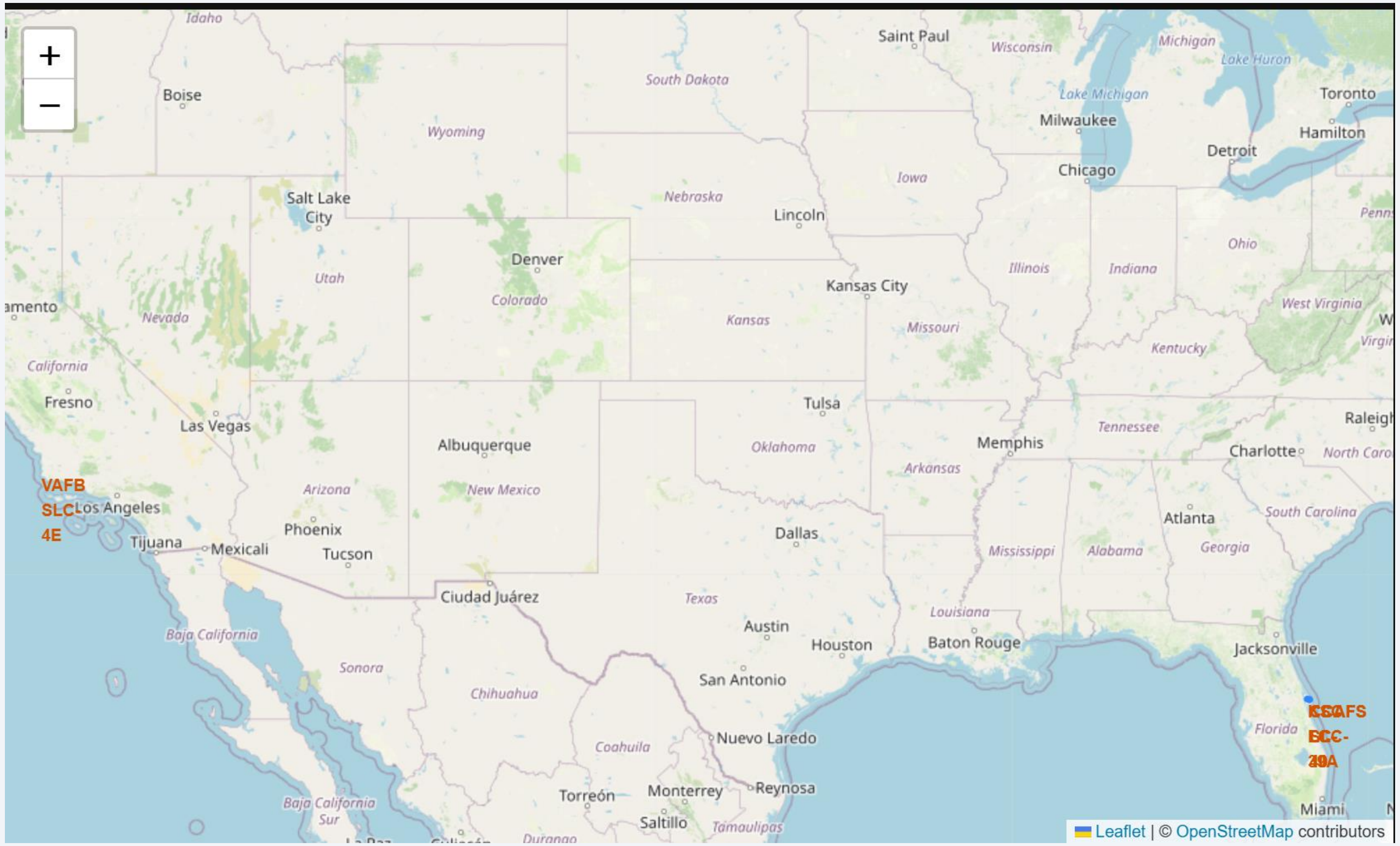
Landing_Outcome	outcome_count
Precluded (drone ship)	1
Failure (parachute)	2
Uncontrolled (ocean)	2
Controlled (ocean)	3
Success (ground pad)	3
Failure (drone ship)	5
Success (drone ship)	5
No attempt	10

- SELECT landing_outcome, count(landing_outcome) AS outcome_count: Counts occurrences of each landing_outcome and labels it as outcome_count.
- FROM spacex table: Retrieves data from the table.
- WHERE date BETWEEN '2010-06-04' AND '2017-03-20’:
- Filters records within the specified date range.
- GROUP BY landing_outcome: Groups results by landing_outcome.
- ORDER BY outcome_count: Orders the results by the count of outcomes.

Section 3

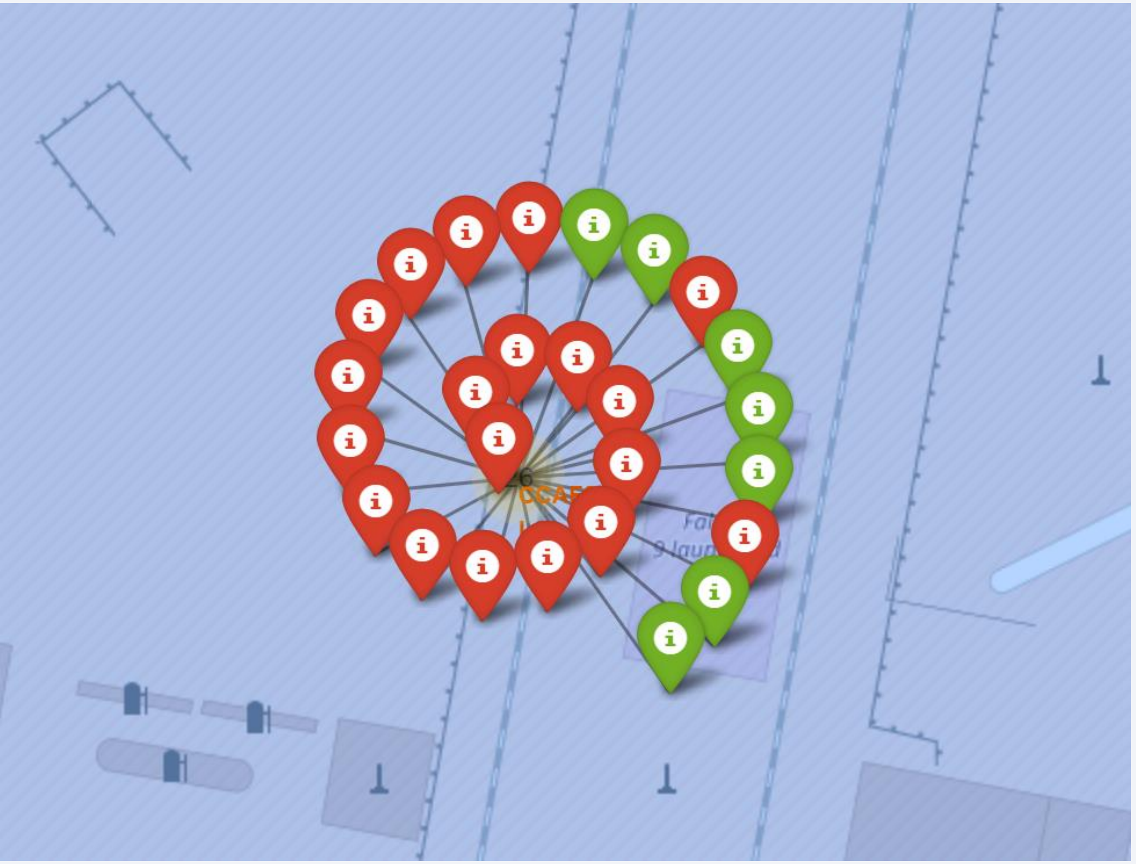
Launch Sites Proximities Analysis

Launchsites locations on map



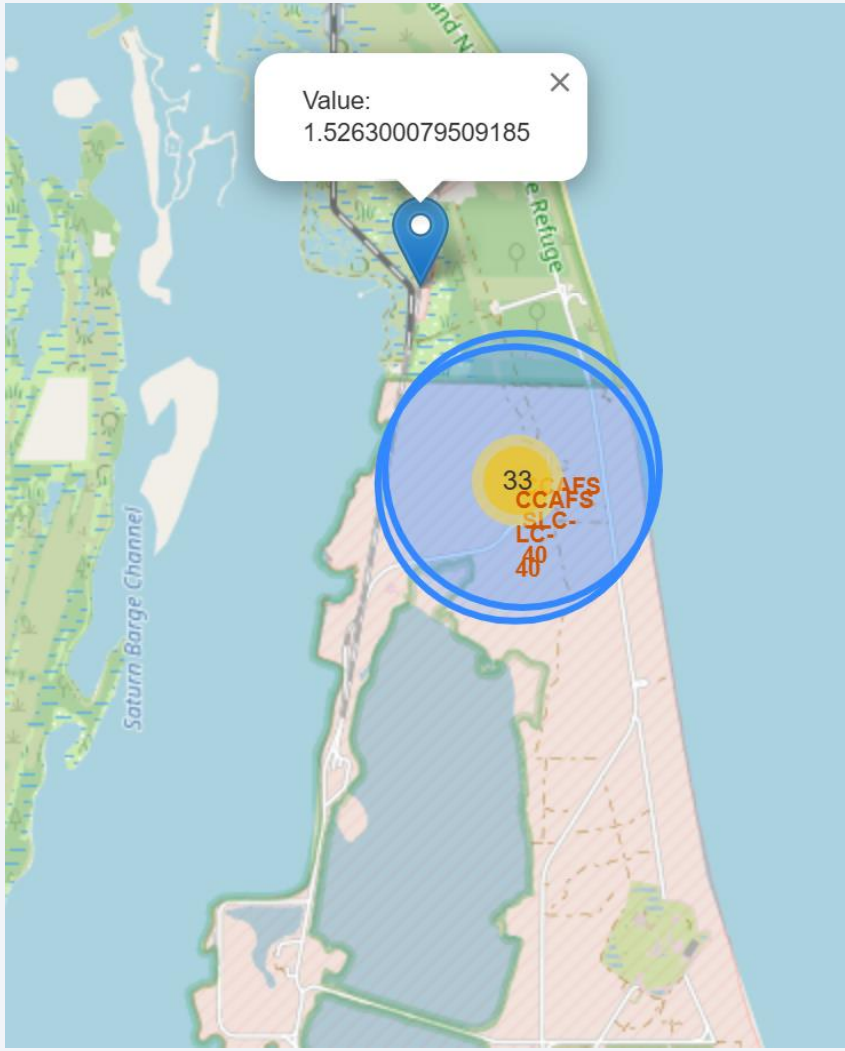
- The map shows the locations of all the launch sites
- Some locations are located in the far south east of USA, while other locations are located in the far south west
- **folium.Map():** Initializes the map with a starting location and zoom level.
- **folium.Circle():** Adds a circle marker at specific coordinates, with options like radius and fill.
- **folium.Popup():** Attaches a popup label to the circle marker.
- **folium.map.Marker():** Adds a marker at the specified coordinates with a custom HTML label.
- **site_map.add_child():** Adds the circle and marker elements to the map.

CCAFS LC-40 Launches

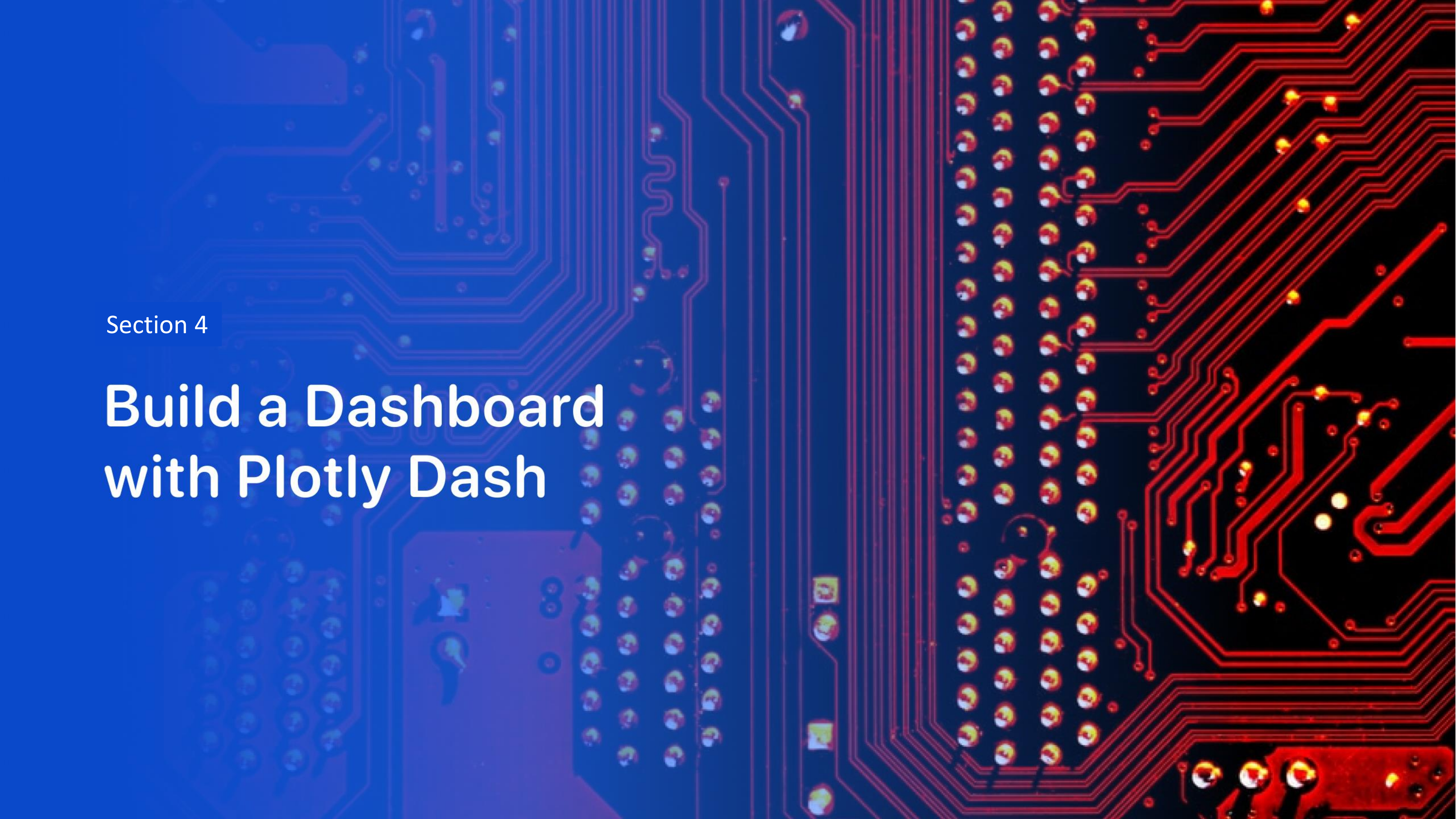


- This screenshot shows a specific launch site (CCAFS LC-40), highlighting all the previous launches with different colors depending on the success of the mission
- **marker_cluster:** An instance of MarkerCluster that groups multiple markers together for a cleaner map visualization.
- **folium.Icon:** Customizes the marker's icon appearance, using color to represent different outcomes (e.g., success or failure).
- **marker_cluster.add_child(marker):** Adds each marker to the marker_cluster instead of directly to the map, allowing for cluster visualization.

Distance from CCAFS-SLC-40 to Titan III Road



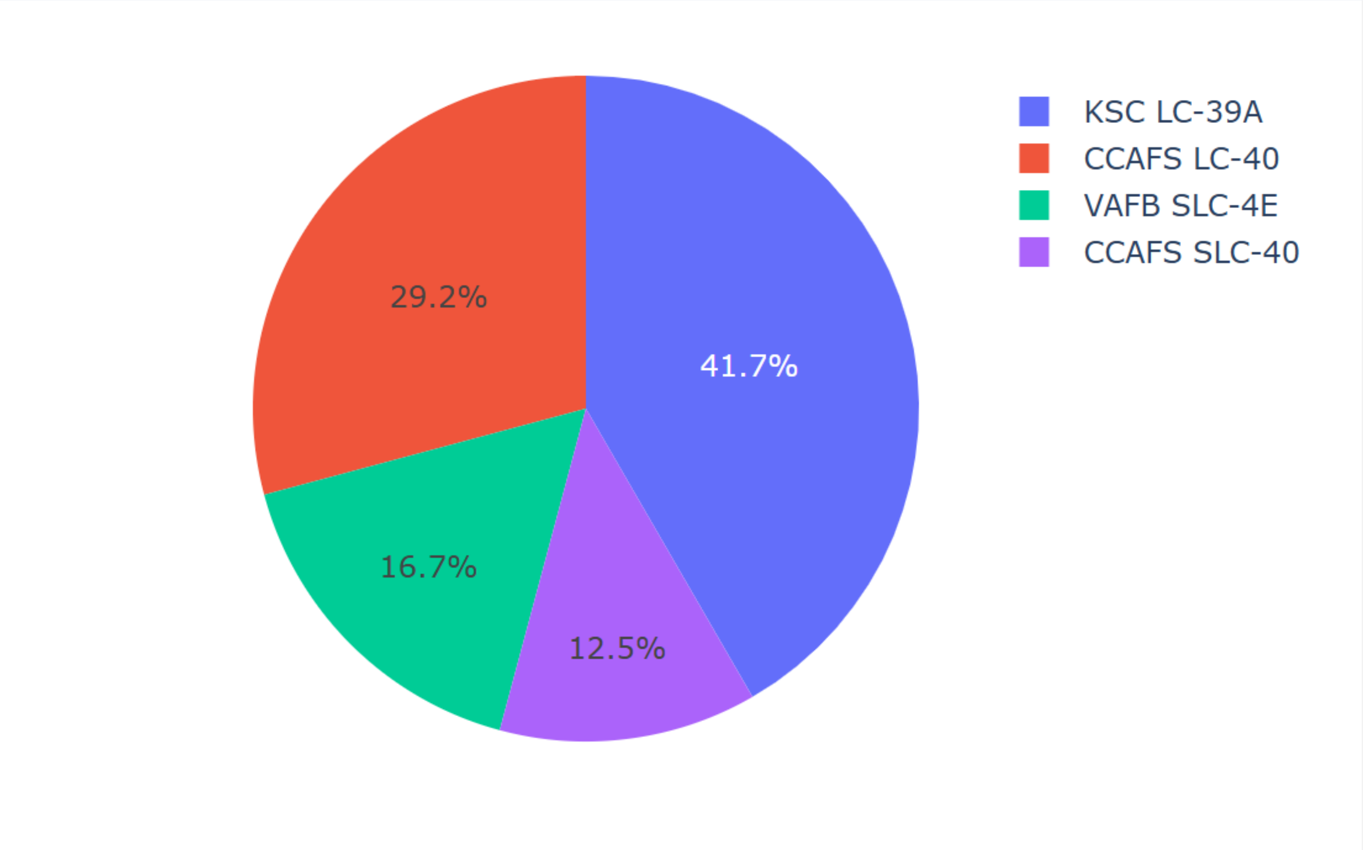
- The screenshot shows Launchsite CCAFS-SLC-40 and another point on Titan III road, new the railway, and the popout from the point shows the distance between them
- **folium.Marker:** Creates a marker on the map at the specified coordinates, which in this case represents the closest coastline point.
- **popup:** Adds a popup to the marker that displays the distance between the coastline point and the launch site.
- **marker_cluster.add_child(distance_marker):** Adds the newly created marker to the marker_cluster, grouping it with other markers on the map.



Section 4

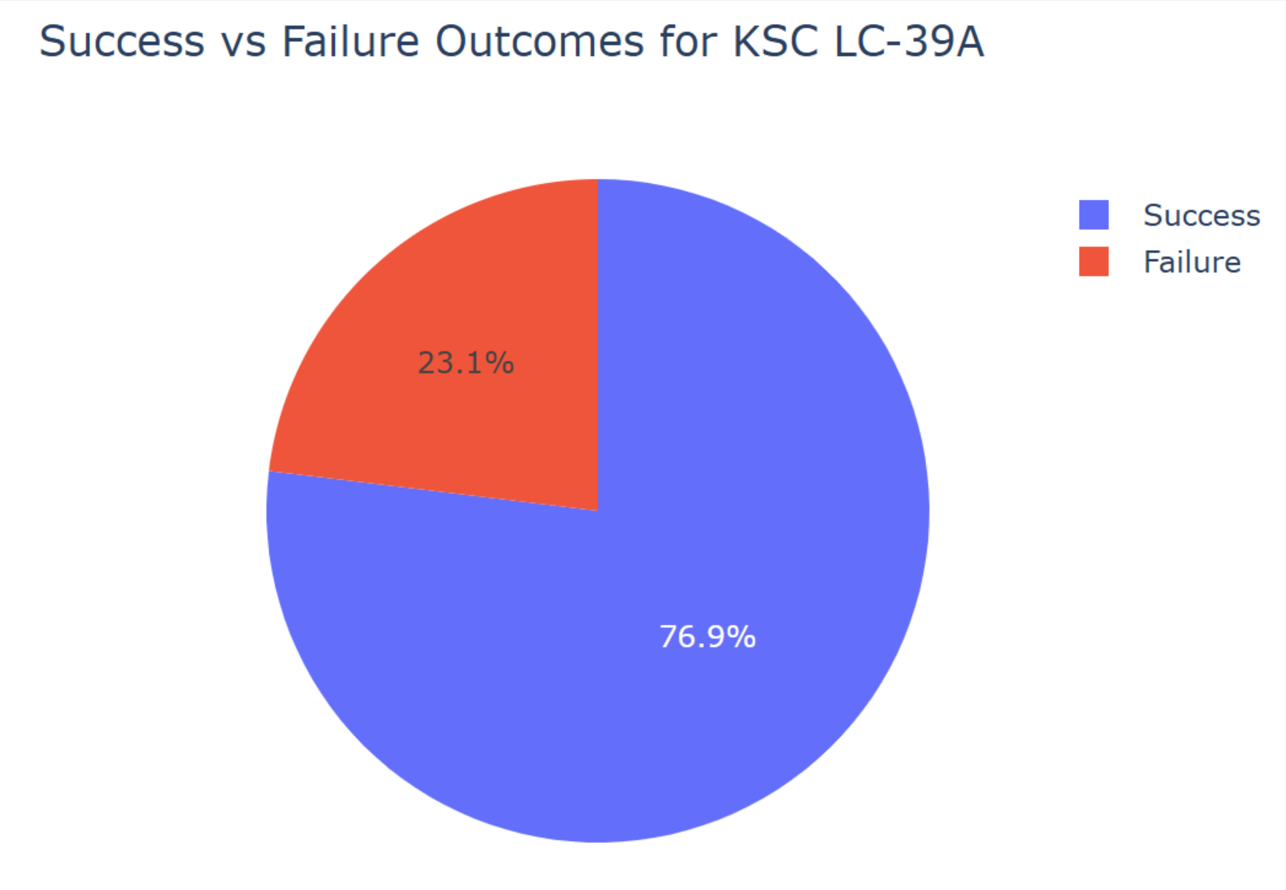
Build a Dashboard with Plotly Dash

Launch Distribution by Site



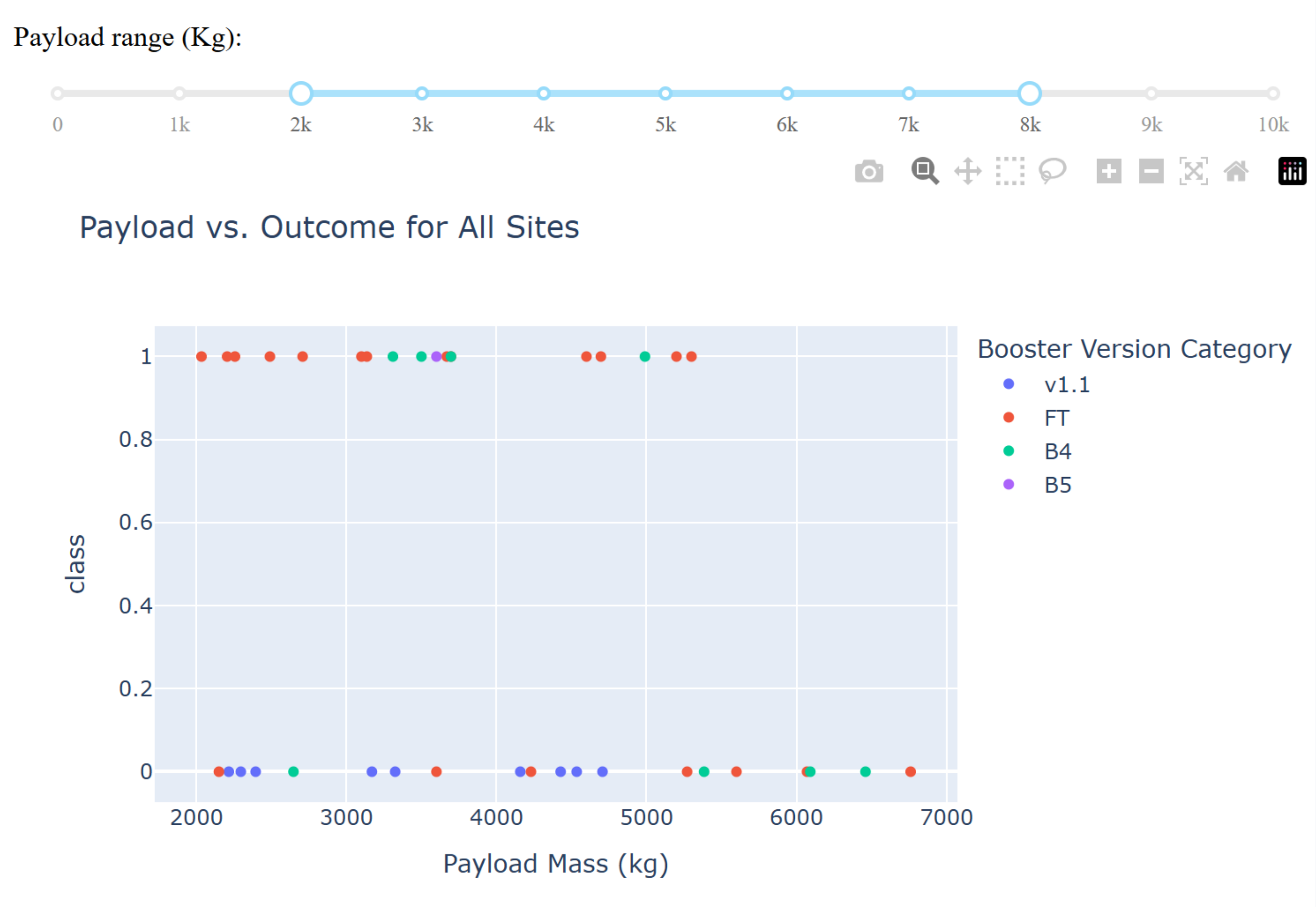
- **Piechart:** Illustrates how launches are distributed across different sites.
- **Site Representation:** Each segment represents the proportion of total launches attributed to each site.
- **Dominant Sites:** Site CCAFS SLC-40 is the site that held the most launches

Success vs Failure Outcomes for KSC LC-39A



- **Pie chart:** Illustrates the proportion of successful versus failed launches at KSC LC-39A
- **High Success Rate:** A significant portion of the launches at KSC LC-39A are successful.

Payload vs Outcome for All Sites

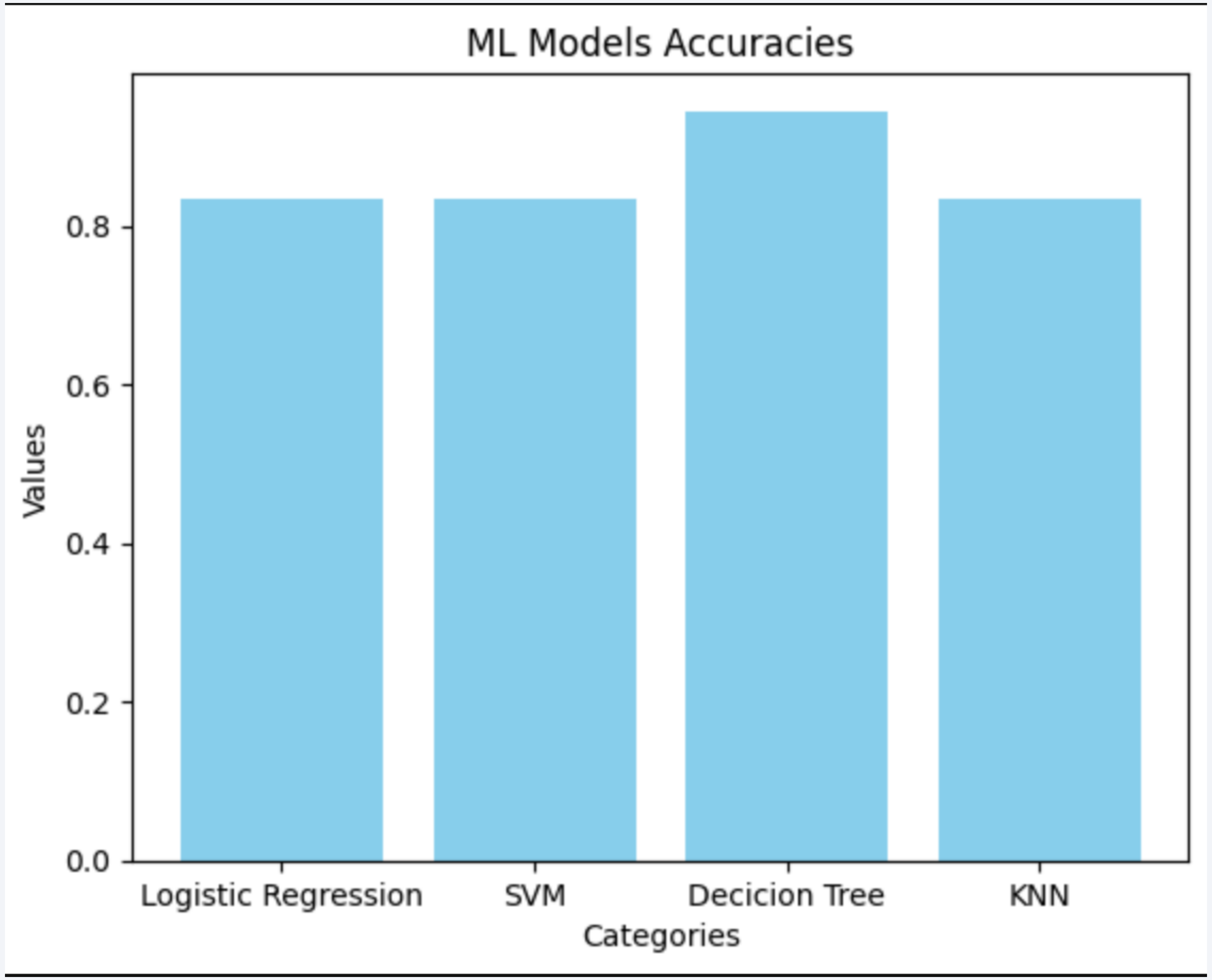


- **Chart:** Shows the relationship between payload mass and launch success, highlighting variations across different booster versions.
- Booster version FT has the highest success rate in the payload range.
- Booster version v1.1 has the lowest success rate in the payload range

Section 5

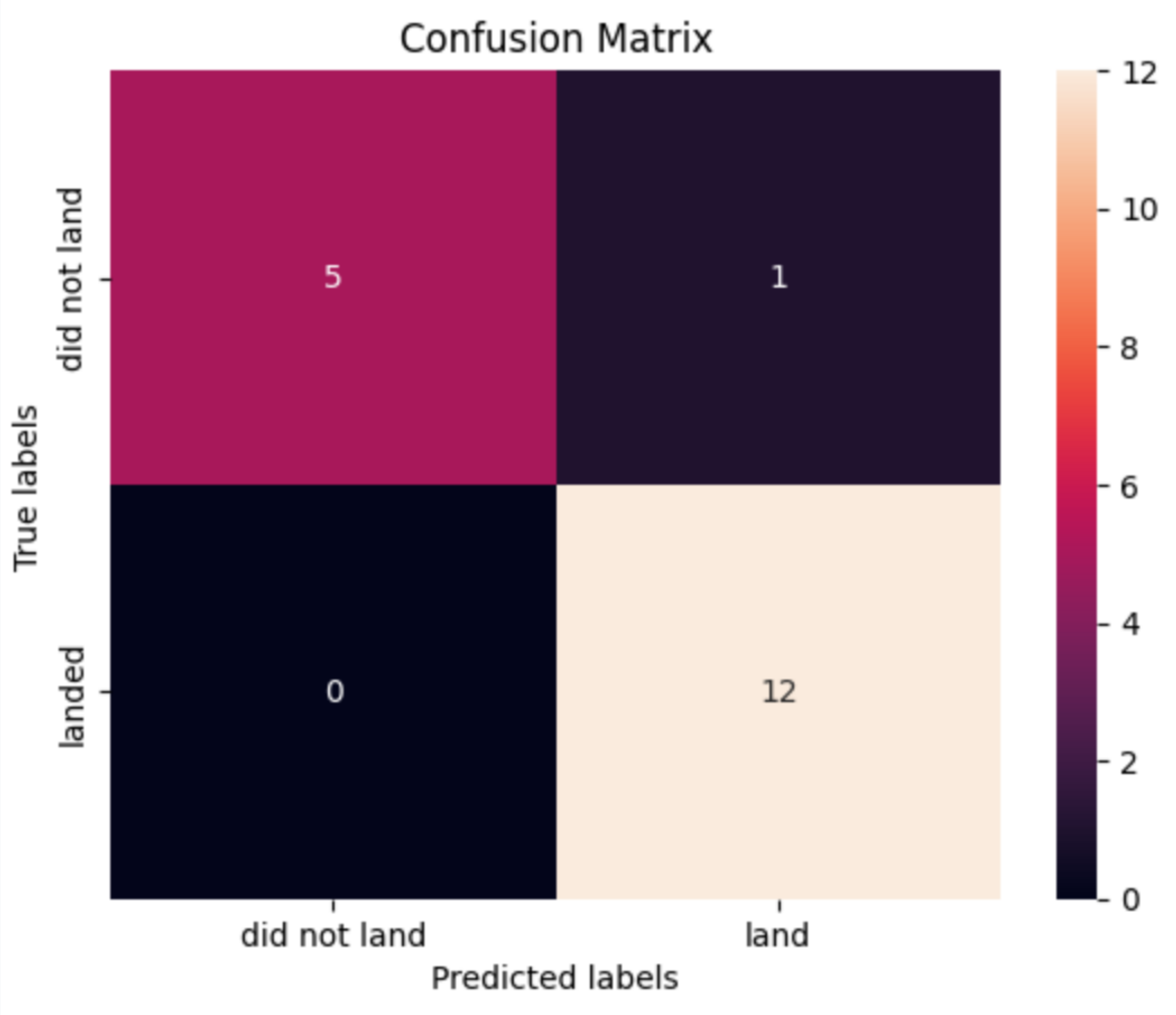
Predictive Analysis (Classification)

Classification Accuracy



- The Decision Tree Classifier has the best accuracy of 94%

Confusion Matrix



•**High True Positive Rate:** The model accurately predicted "Landed" 12 times, showing strong performance in identifying successful landings.
•**Perfect True Negative Rate:** The model correctly predicted "Didn't Land" in all 5 relevant cases, with no false negatives, suggesting reliable identification of failed landings.
•**No False Negatives:** The absence of false negatives (0 cases) implies the model never missed a successful landing, which is a positive aspect of its predictive accuracy.
•**Overall Accuracy:** The model demonstrates solid overall accuracy with only a few errors, primarily in overpredicting landings.

Conclusions

- **Recap of Objectives:** Summarized the key goals and the problems targeted by the project.
- **Methodology:** Implemented a robust data-driven approach combining advanced techniques.
- **Findings:** Identified key patterns and trends that can impact decision-making.
- **Impact:** Provided actionable insights that are immediately applicable to real-world scenarios.

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!