IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Akintunde Rockson
June 28, 2025

Outline

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

Executive Summary

The primary objective of this study is to analyze SpaceX Falcon 9 launch data and apply machine learning models to uncover the key factors influencing first-stage landing success. These insights are essential for SpaceY to remain competitive with SpaceX.

Methodologies

- **Data Collection:** Gathered relevant data through APIs and web scraping techniques.
- **Data Transformation:** Performed data wrangling to clean and structure the collected data for analysis.
- **Exploratory Data Analysis:** Leveraged SQL, Pandas, and visualization tools to explore and understand key patterns in the data.
- **Interactive Dashboard:** Developed an interactive dashboard using Plotly Dash and Folium to visualize launch outcomes and geographic data.
- **Predictive Modeling:** Built machine learning models to predict the likelihood of Falcon 9 first-stage landing success.

Executive Summary

Methodologies

- **Data Analysis:** Granular insights through comprehensive exploratory data analysis.
- **Data Visualizations and Interactive Dashboards:** Visual representations and interactive tools to facilitate deeper exploration of the data.
- **Predictive Model Analysis:** Evaluation of model performance and accuracy in forecasting Falcon 9 first-stage landing success

Introduction

Project Background

Recent advancements in private space travel have made the space industry more mainstream and accessible to the public. However, launch costs continue to pose a major barrier for emerging competitors. SpaceX holds a competitive advantage as a pioneer in first-stage rocket recovery and reuse technology. With launch costs around $62 million and the ability to reuse the first stage for future missions, SpaceX operates far more efficiently than many of its rivals, who spend over $165 million per launch. This cost advantage positions SpaceX as a dominant force in the evolving space industry.

Introduction

Problems

This research explores several key questions related to SpaceX Falcon 9 first-stage landings. It examines whether landing success can be reliably predicted, investigates the impact of variables such as launch site, payload mass, and booster version on landing outcomes, and analyzes correlations between various parameters and their associated success rates. By addressing these areas, the study aims to generate insights that may support other space agencies in making informed, strategic decisions as they seek to compete with SpaceX.

Section 1

Methodology

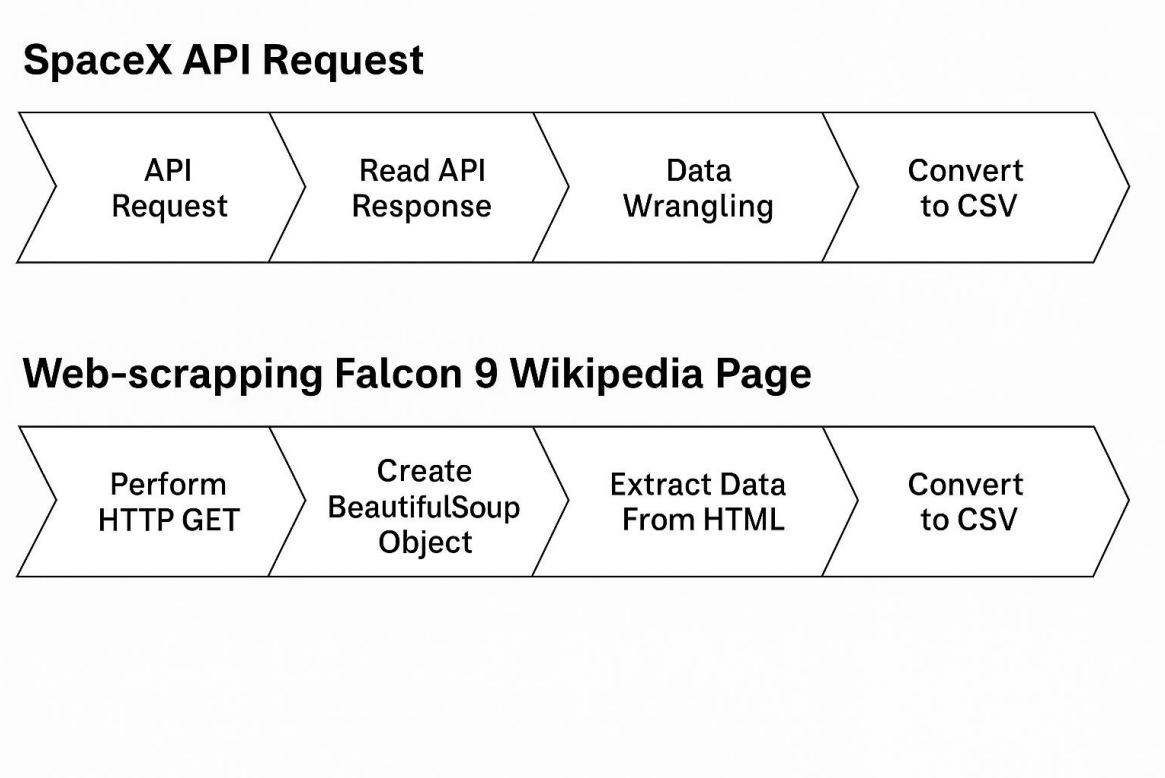
Methodology

Executive Summary

- Data collection methodology:
  - Collected data using the SpaceX API and by web scraping the Falcon 9 Wikipedia page.
- Perform data wrangling
  - Cleaned and structured the dataset through data wrangling, including converting mission outcomes into binary values (0 for failure, 1 for success).
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardized and transformed the data, then split it into training and test sets. Various classification algorithms including Logistic Regression, SVM, Decision Tree, and KNN were evaluated on the test data to determine the most effective model.

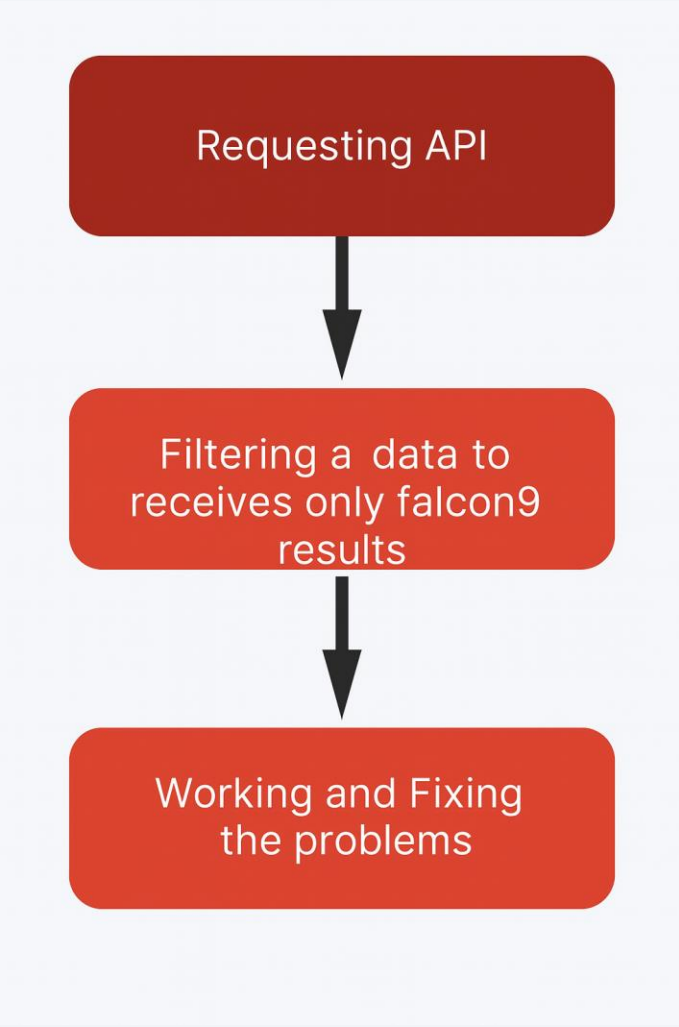
Data Collection

- **SpaceX REST API:** Collected detailed launch data such as rocket type, launch date, payload mass, launch outcome, launch site information (including coordinates), booster version for Falcon 9, and landing results.
- **Web Scraping:** Extracted supplementary Falcon 9 launch details from Wikipedia using BeautifulSoup, including launch date and time, booster version, payload specifications, orbit type, and other contextual information.



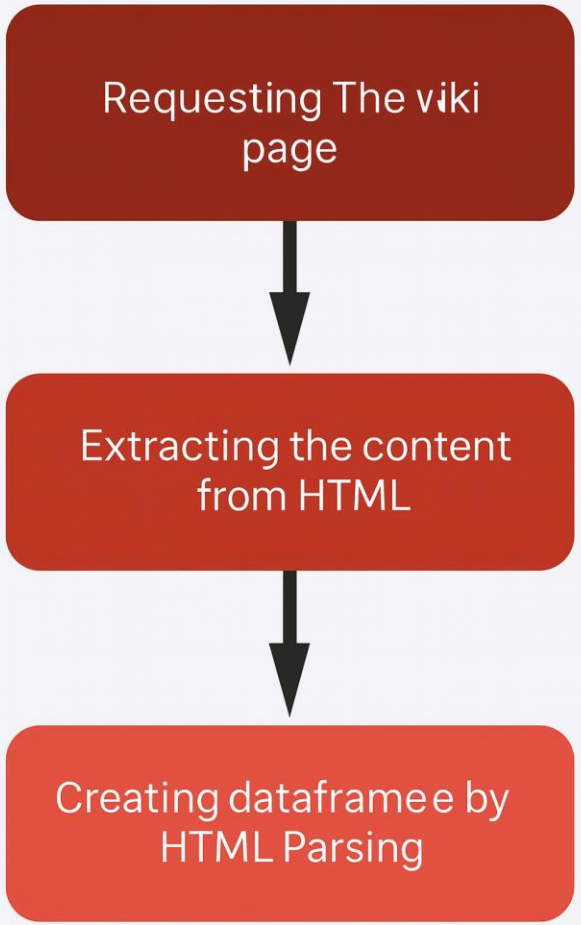
Data Collection – SpaceX API

- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/jupyter-labs-spacex-data-collection-api.ipynb



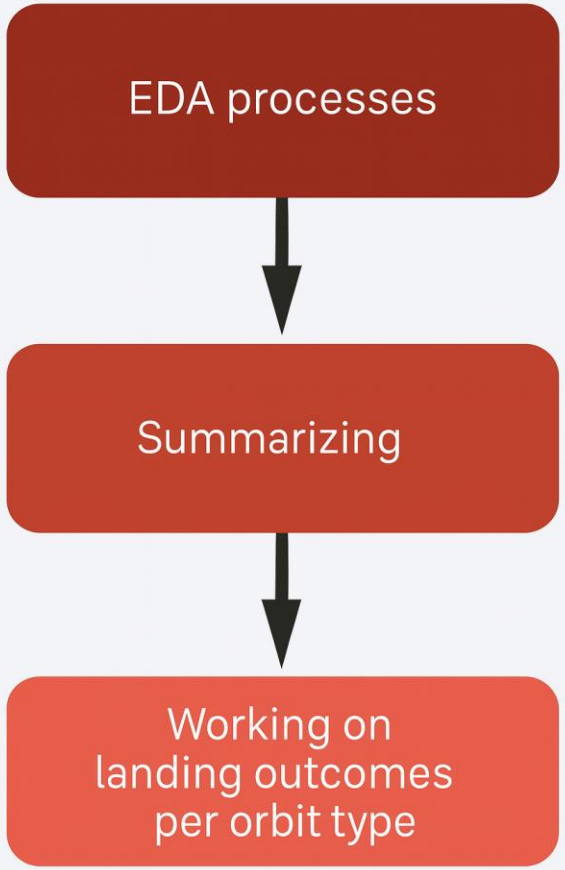
Data Collection - Scraping

- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/jupyter-labs-webscraping.ipynb



Data Wrangling

- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/labs-jupyter-spacex-Data wrangling.ipynb



EDA with Data Visualization

The notebook includes several visualizations to explore relationships within the Falcon 9 launch data:

- A scatter plot of Payload Mass vs. Flight Number was used to examine how payload size relates to launch success.
- Multiple categorical scatter plots (with Launch Site, Flight Number, and Payload Mass on different axes) were used to assess how location and mass impact launch outcomes.
- A bar chart showing average success rate by orbit type provided insight into which orbits correlate with higher success.
- Additional scatter plots analyzed how orbit type relates to Flight Number and Payload Mass.
- A line plot of success rate over time showed trends in launch performance.
- Github: http://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/edadataviz.ipynb

EDA with SQL

- Retrieved full data and specific columns from the SpaceX table to explore launch details.
- Grouped and counted mission outcomes to evaluate landing success rates.
- Aggregated payload mass by launch site to identify top-performing locations.
- Filtered for successful landings to analyze conditions contributing to success.
- Ranked launches by payload size to find the most significant missions.
- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

To visualize launch site locations and related spatial information, several map objects were added to the Folium map:

- **Markers**: Placed at launch sites to indicate exact coordinates and provide popup information. These help identify and label each launch location.
- **Circles**: Drawn around launch sites to highlight a specific area of interest, possibly to show range or proximity. They visually emphasize spatial coverage.
- **Lines**: Used to draw connections (e.g., from a launch site to a coordinate or path), helping to illustrate directions, trajectories, or distances.
- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

To build an interactive dashboard for exploring Falcon 9 launch data, the following plots and interactions were added:

- **Graph components (e.g., pie charts, bar charts):** Used to visualize launch outcomes and payload performance, helping users quickly identify success rates and trends across different categories.
- **Dropdown**: Allows users to select specific launch sites, dynamically filtering the data to focus on results from a chosen location.
- **RangeSlider**: Enables users to adjust the payload mass range, helping explore how launch success varies with payload size.
- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/plotly_dash.py

Predictive Analysis (Classification)

To build, evaluate, and improve the classification models, the following steps were taken:

- Prepared the dataset by scaling and splitting it into training and test sets.
- Built multiple classification models including Logistic Regression, SVM, K-Nearest Neighbors, and Decision Tree.
- Trained each model on the training set and evaluated their accuracy using test data.
- Compared model performance using accuracy scores and classification reports.
- Selected the best performing model based on evaluation metrics such as precision and accuracy.
- Github: https://github.com/TundeRockson/IBMDataScience/blob/main/Capstone/SpaceX_Machine Learning Prediction_Part_5.ipynb

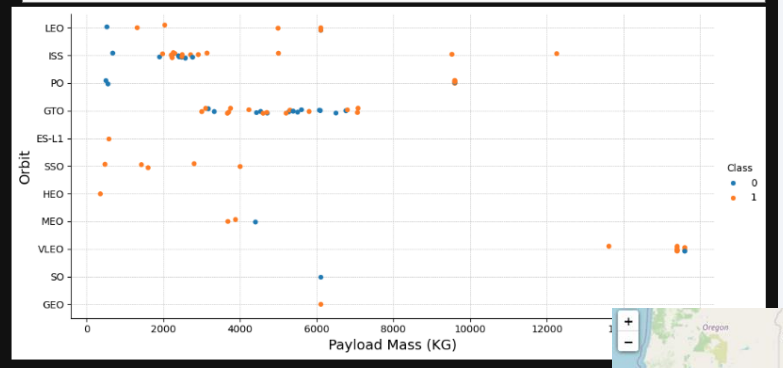
Data Preprocessing → Model Building → Model Training → Model Evaluation → Best Model Selection

Results

TASK 5: Visualize the relationship between Payload Mass and Orbit type

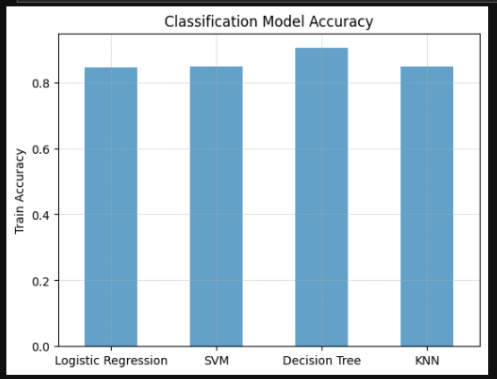
Similarly, we can plot the Payload Mass vs. Orbit scatter point charts to reveal the relationship between Payload Mass and Orbit type

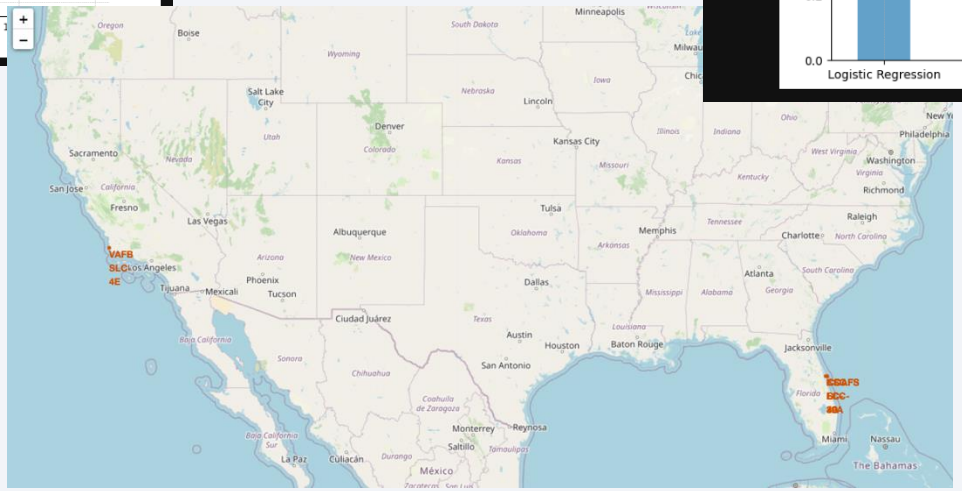
In [9]:
```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect=2)
plt.xlabel("Payload Mass (KG)",fontsize=14)
plt.ylabel("Orbit",fontsize=14)
plt.grid(True, linestyle='--', linewidth=0.4)
plt.show()
```

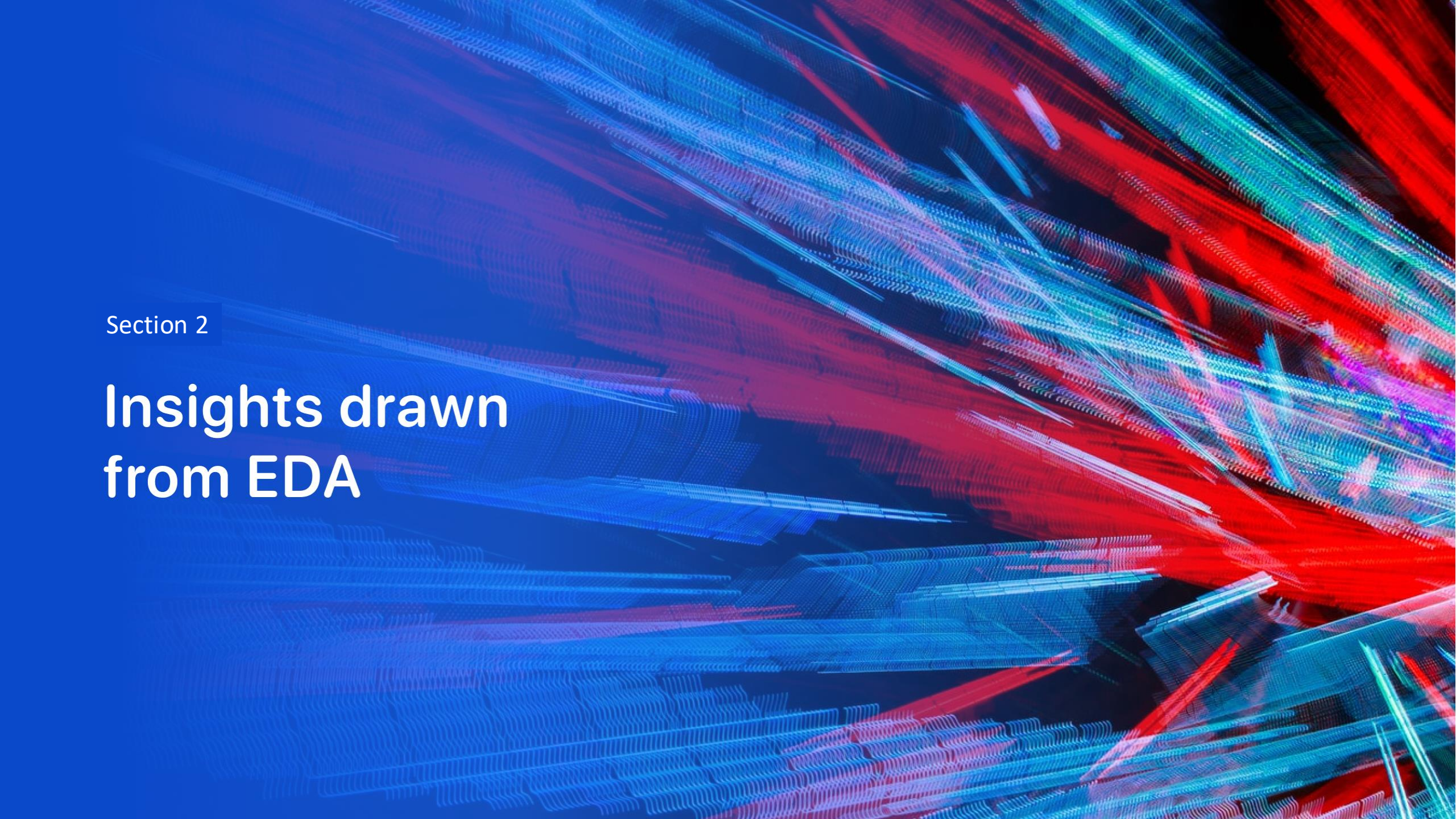


Find the method performs best:

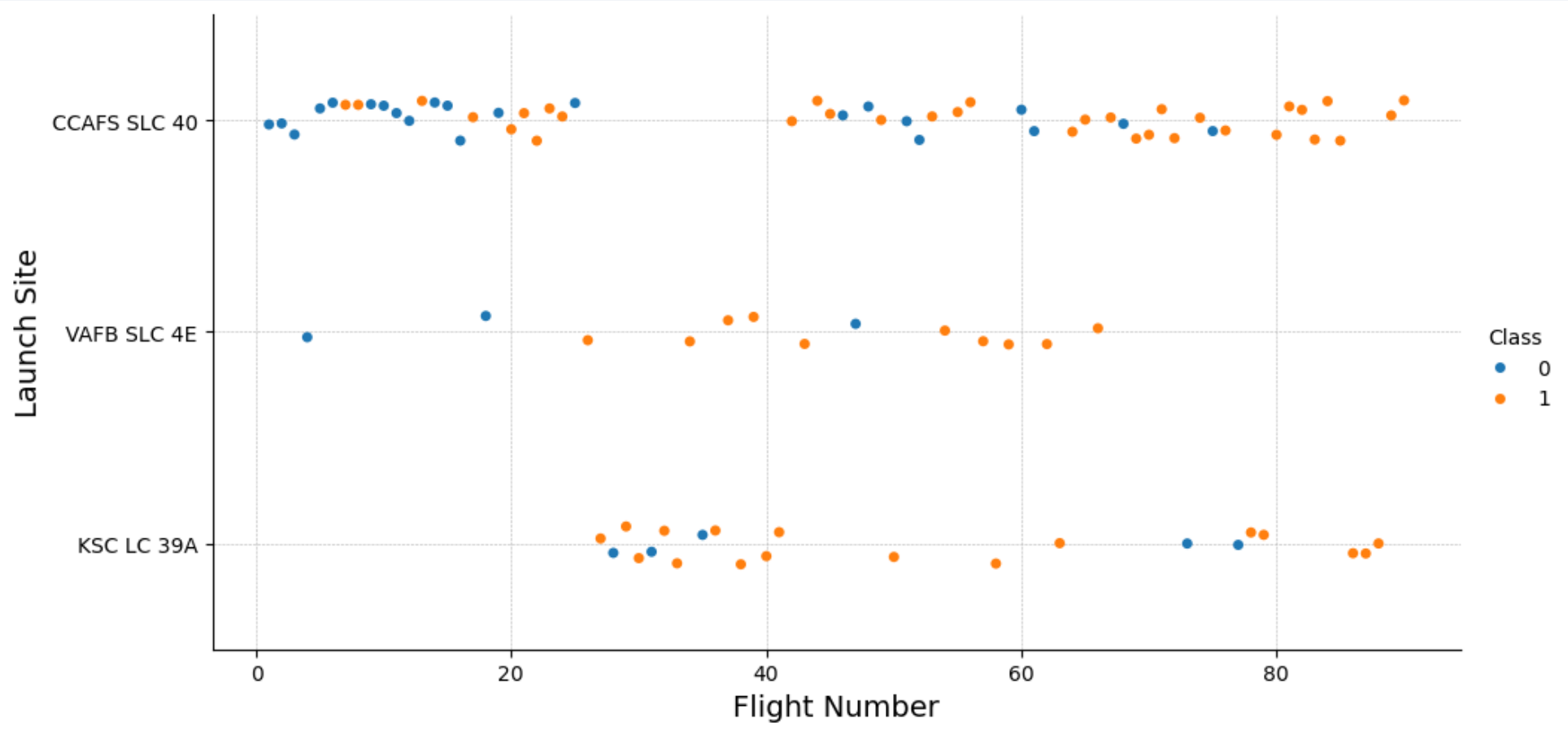
In [65]:
```
Report = pd.DataFrame({
    'algorithm': ['Logistic Regression', 'SVM','Decision Tree','KNN'],
    'train accuracy': [logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_],
    'test accuracy': [lr_acc, svm_acc, tree_acc, knn_acc]
})
Report.plot(
    kind='bar',
    x='algorithm',
    y='train accuracy',
    alpha=0.7,
    rot=0,
    ylabel='Train Accuracy', xlabel='', legend=False,
    title='Classification Model Accuracy')
plt.grid(True, linestyle='--', linewidth=0.4)
plt.show()
```



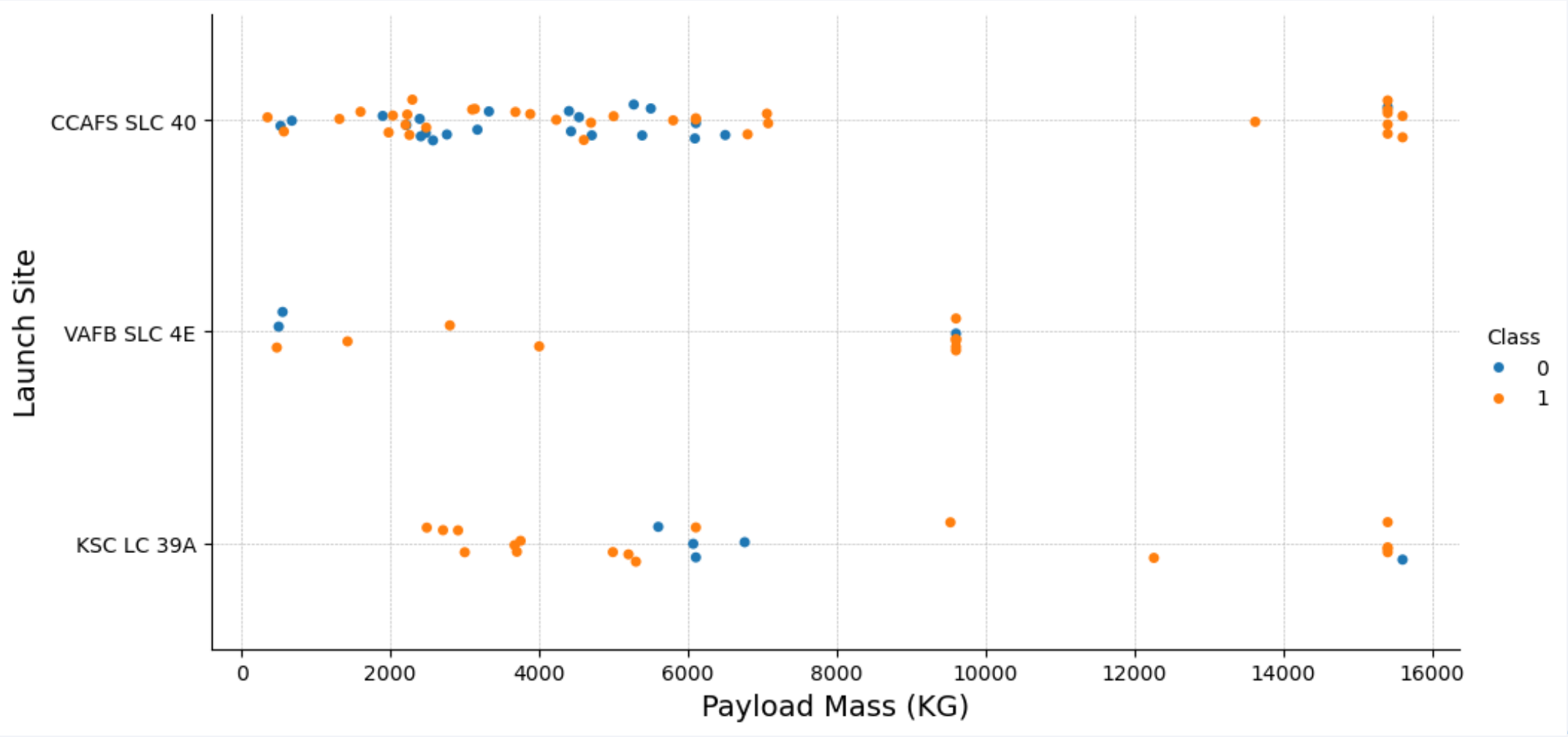



Section 2
Insights drawn from EDA

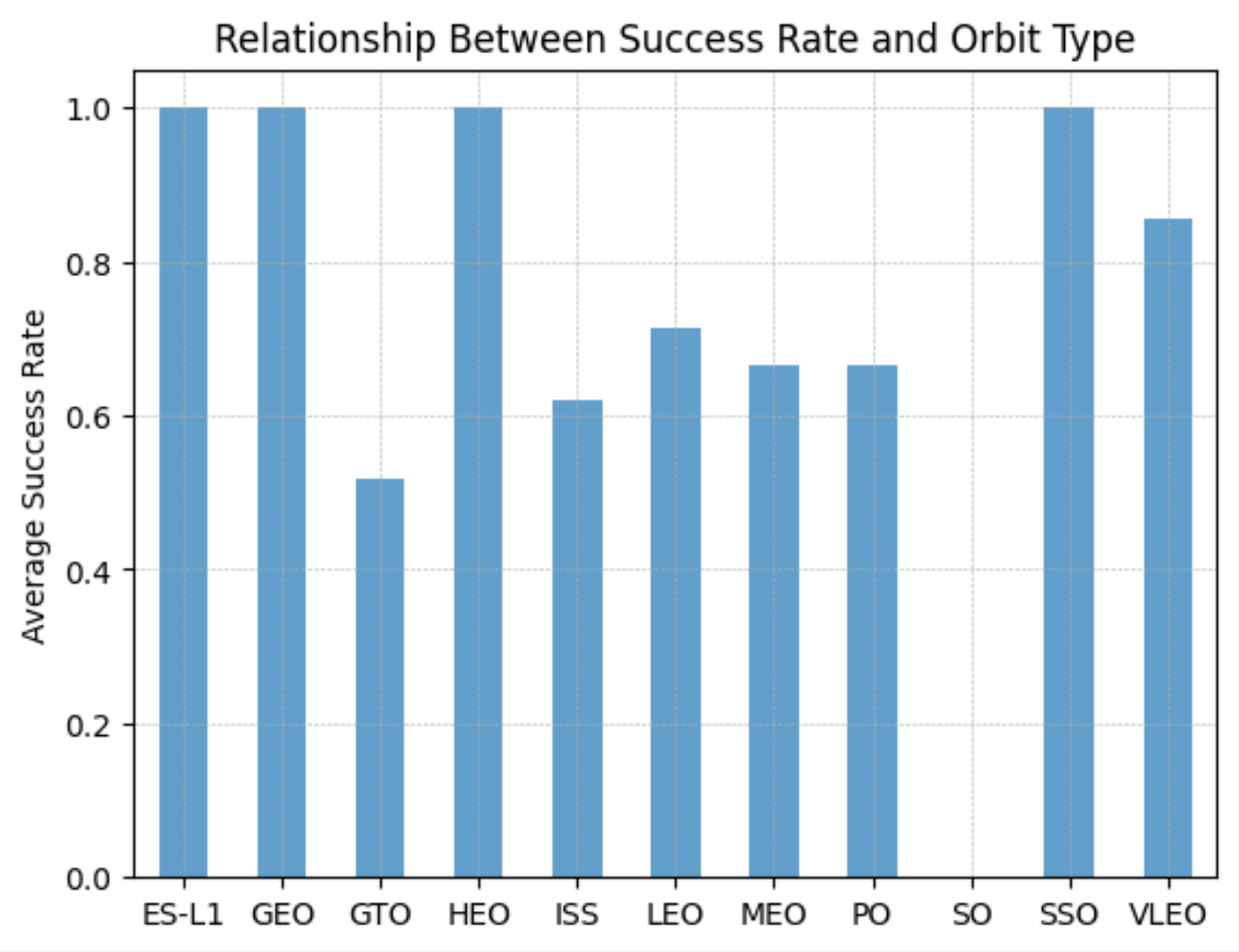
Flight Number vs. Launch Site



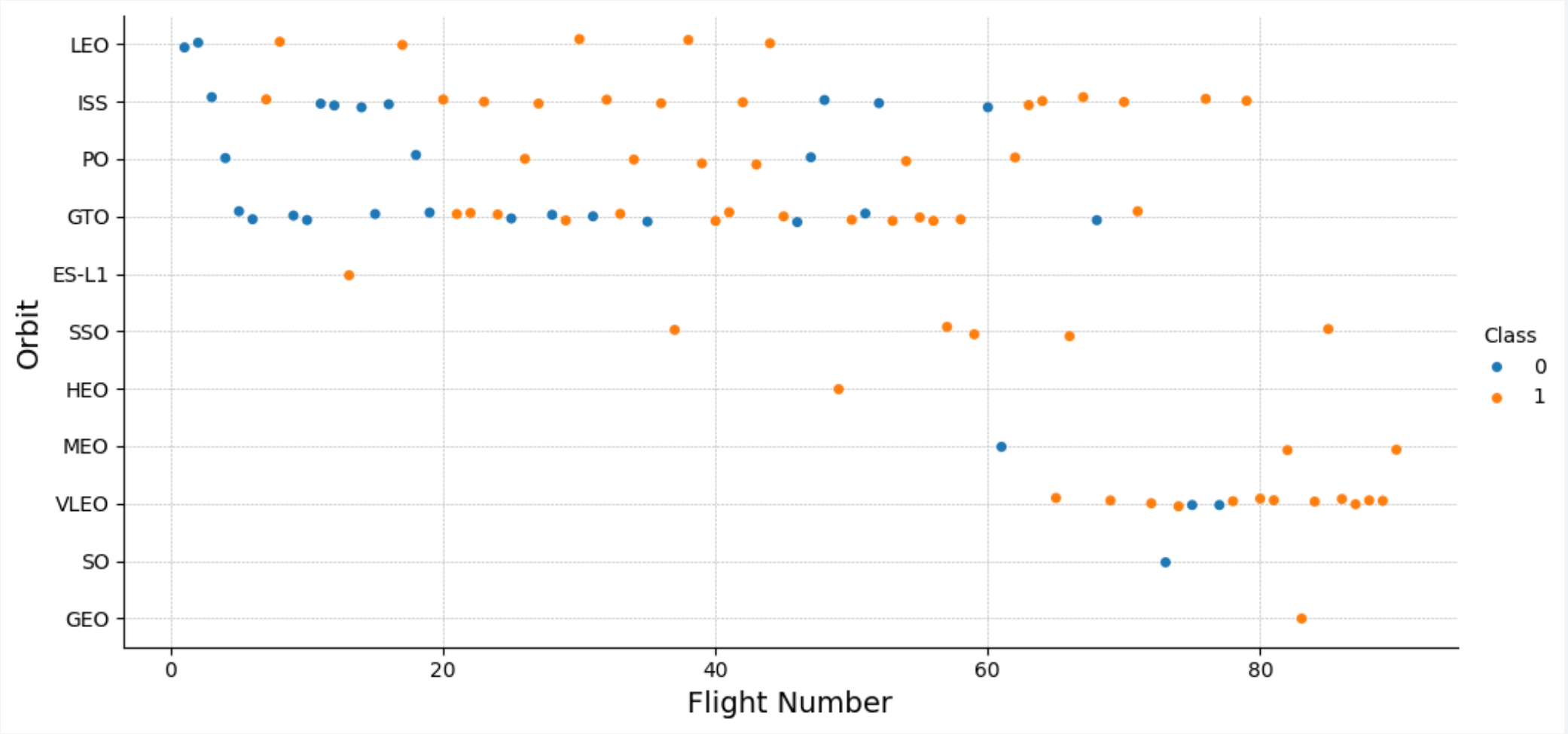
Payload vs. Launch Site



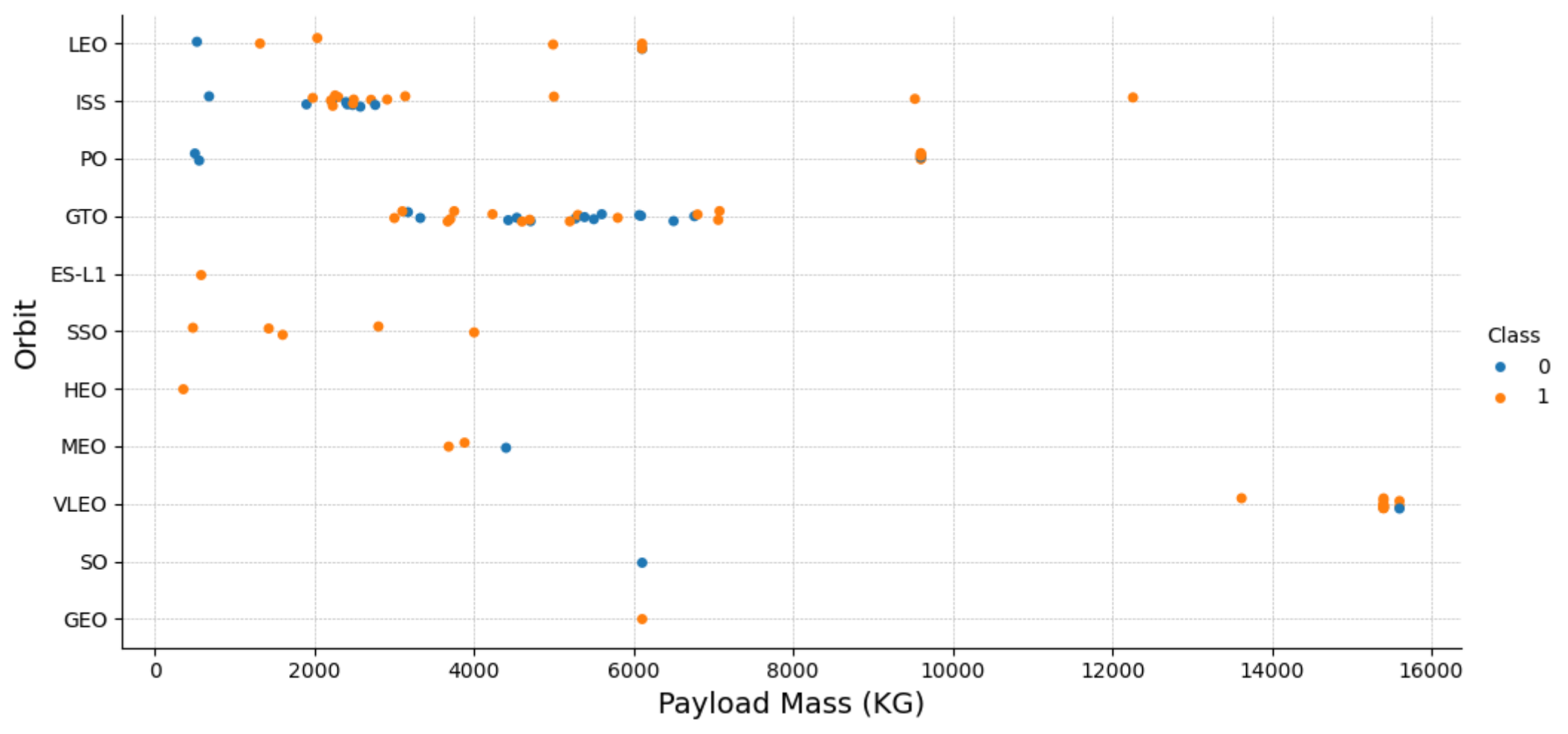
Success Rate vs. Orbit Type



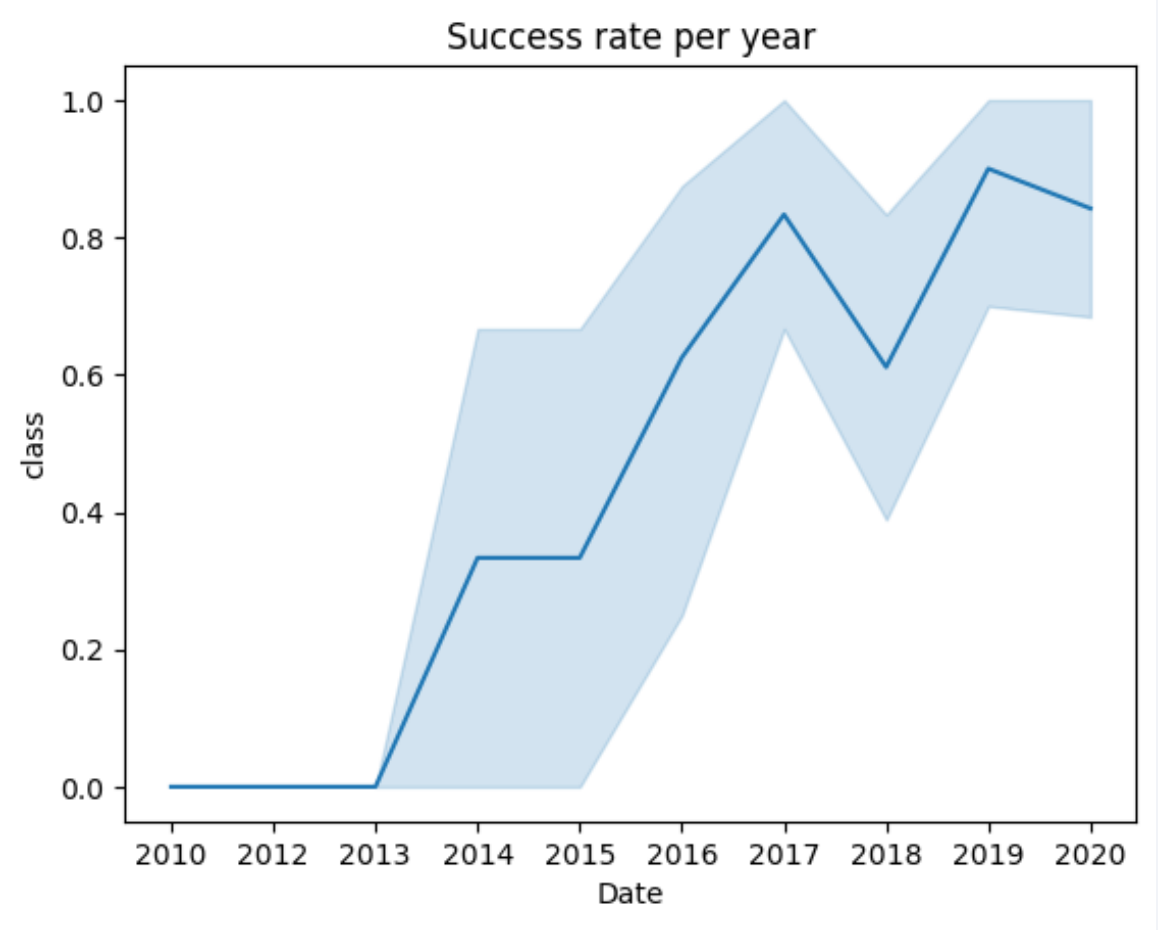
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

- **DISTINCT** is used to return only the unique values from the "Launch_Site" column.
- The query results revealed a total of 4 unique launch sites.

```
In [10]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
 * sqlite:///my_data1.db
Done.
Out[10]:
```

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

Launch Site Names Begin with 'CCA'

- The **LIKE** keyword with the format %CCA% filters records where the "Launch_Site" column contains the substring "CCA".
- The **LIMIT 5** clause restricts the output to only 5 records.

```
In [11]: %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "%CCA%" LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- **SUM** aggregates the values in the “PAYLOAD_MASS__KG_” column to calculate the total payload mass.
- The **WHERE** clause filters the data to include only records where the customer is “NASA (CRS)”.

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)";
 * sqlite:///my_data1.db
Done.
 total_payload_mass
              45596
```

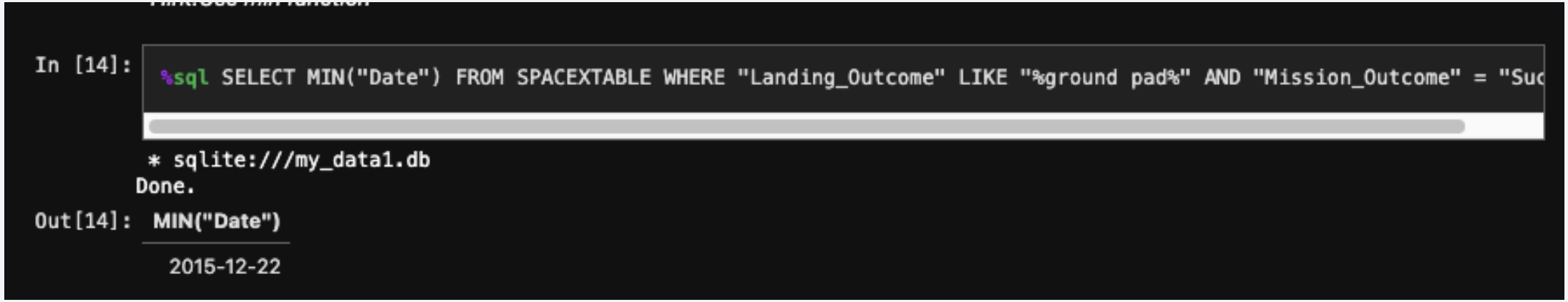
Average Payload Mass by F9 v1.1

- **AVG** calculates the average value from the “PAYLOAD_MASS_KG” column.
- The **WHERE** clause filters the rows to include only those where the booster version contains “F9 v1.1”.

```
In [13]: %sql SELECT AVG("PAYLOAD_MASS__KG_") AS average_payload_mass FROM SPACEXTABLE WHERE "Booster_version" LIKE "%F9 v
 * sqlite:///my_data1.db
Done.
Out[13]: average_payload_mass
         2534.6666666666665
```

First Successful Ground Landing Date

- The **MIN** function retrieves the earliest (oldest) date from the “Date” column.
- The **WHERE** clause filters the rows to include only those where the “Landing_Outcome” is equal to “Success” and occurred on a group pad.



Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Present your query result with a short explanation here

In [15]:
```
%%sql
SELECT DISTINCT "Booster_Version"
FROM SPACEXTABLE WHERE
    "Landing_Outcome" LIKE "%drone ship%" AND
    "Mission_Outcome" LIKE "%Success%" AND
    "PAYLOAD_MASS__KG_" > 4000 AND
    "PAYLOAD_MASS__KG_" < 6000;
```

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

Out[15]:

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- This approach allowed us to calculate the frequency of each unique value or outcome by applying conditional logic within the query.

In [16]:
```
%%sql
SELECT
    COUNT(CASE WHEN "Mission_Outcome" LIKE "%Success%" THEN 1 END) AS total_number_success,
    COUNT(CASE WHEN "Mission_Outcome" LIKE "%Failure%" THEN 1 END) AS total_number_failure
FROM SPACEXTABLE;
```

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

Out[16]:

total_number_success	total_number_failure
100	1

Boosters Carried Maximum Payload

- The subquery uses the **MAX** function on the “PAYLOAD_MASS__KG_” column to determine the highest payload mass.
- The main query retrieves the corresponding “Booster_Version” where the payload mass equals this maximum value (15600 kg).

```
In [17]: %%sql
         SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") F
          * sqlite:///my_data1.db
         Done.
Out[17]: 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

- The query selects “Landing_Outcome”, “Booster_Version”, “Launch_Site”, and “Month_name” columns.
- It filters records where “Landing_Outcome” contains “drone ship”, “Mission_Outcome” contains “Failure”, and the “Date” falls within the year 2015.

In [18]:
```
%%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 "Month_Name",
    "Booster_Version",
    "Launch_Site",
    "Mission_Outcome",
    "Landing_Outcome"
FROM SPACEXTABLE
WHERE
    "Landing_Outcome" LIKE "%drone ship%" AND
    "Mission_Outcome" LIKE "%Failure%" AND
    substr("Date", 0, 5) = '2015';
```

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

Out[18]:

Month_Name	Booster_Version	Launch_Site	Mission_Outcome	Landing_Outcome
June	F9 v1.1 B1018	CCAFS LC-40	Failure (in flight)	Precluded (drone ship)

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

- The GROUP BY clause organizes the “Landing_Outcome” column into distinct groups.
- The **BETWEEN** clause filters records with dates from “2010-06-04” to “2017-03-20”.
- The **ORDER BY** clause sorts the grouped counts in descending order.
- The output is a ranked list of landing outcomes based on frequency within the specified date range.

In [19]:
```
%%sql
SELECT
    "Landing_Outcome",
    COUNT(*) AS "Count"
FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY "Count" DESC;
```

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

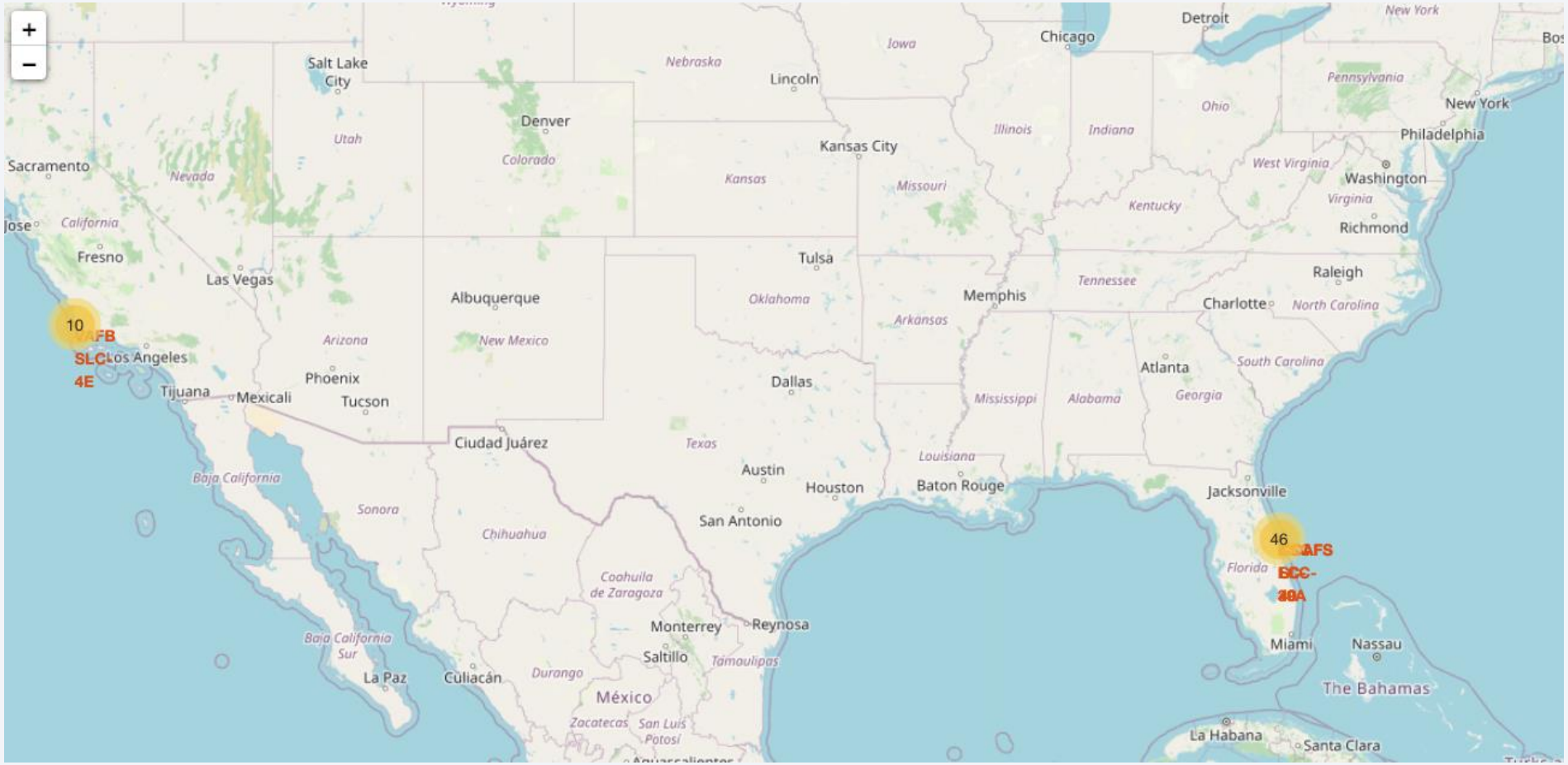
Out[19]:

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

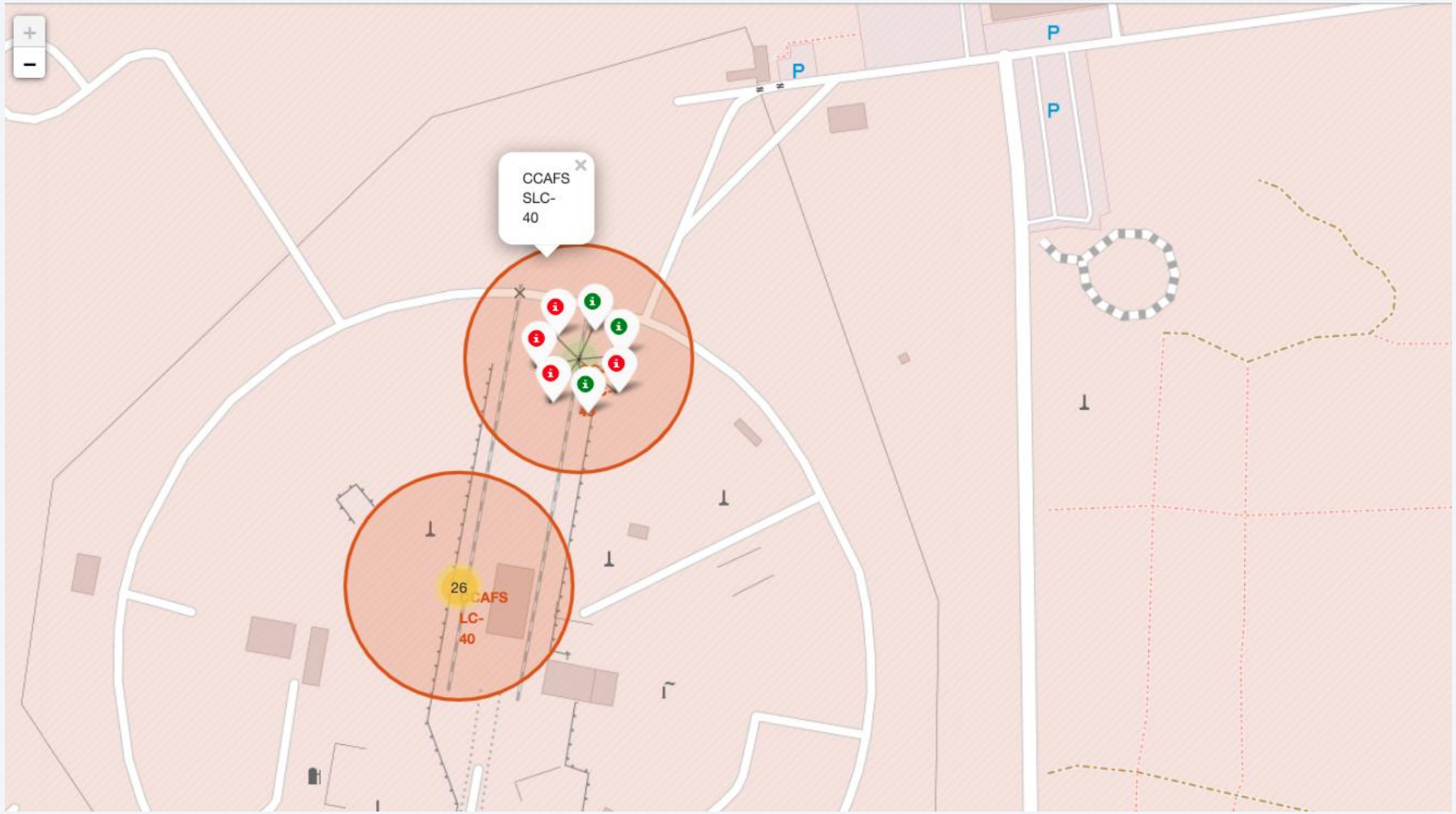
Section 3

Launch Sites Proximities Analysis

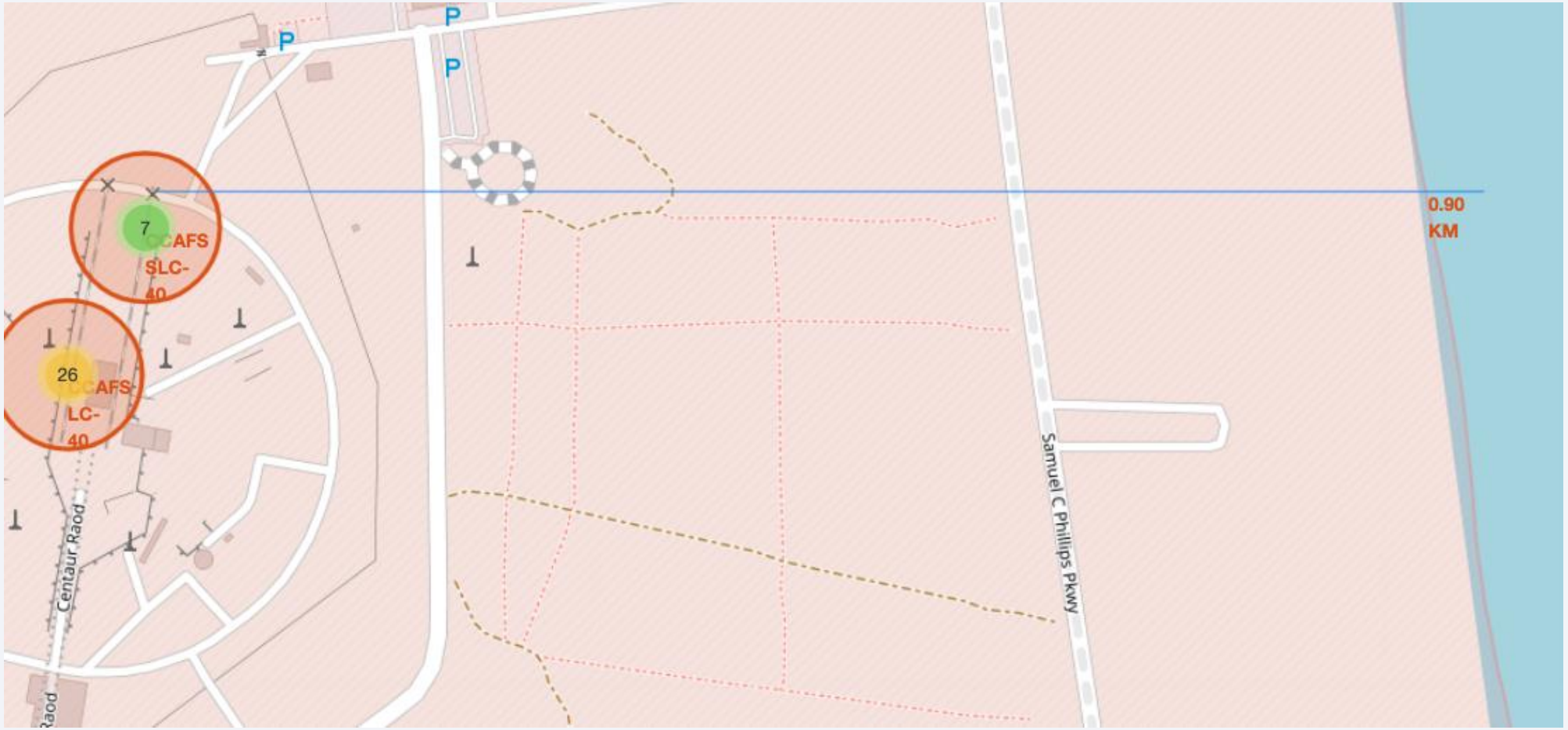
SpaceX Falcon9 - Launch Sites Map

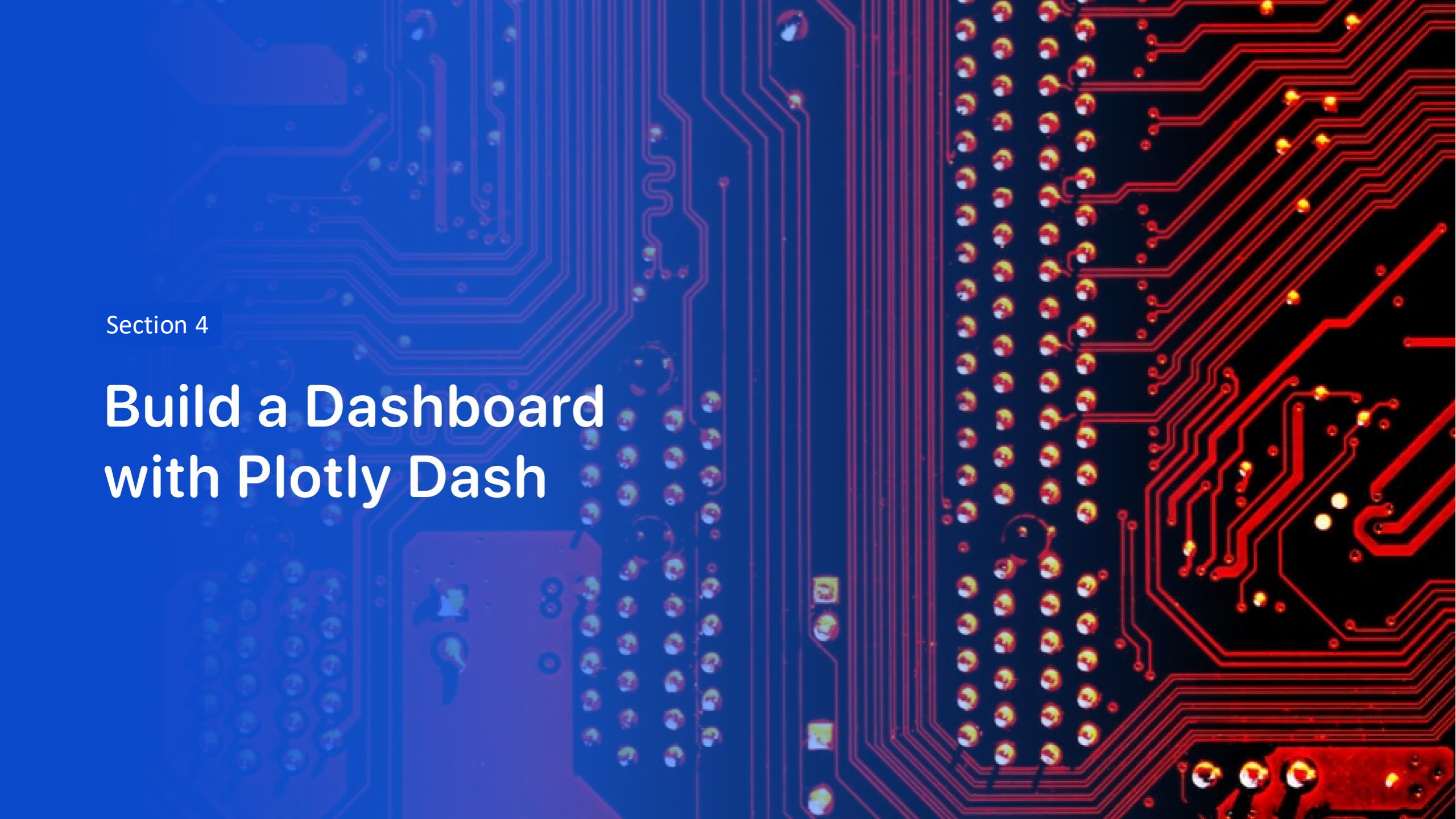


SpaceX Falcon9 – Launch Outcomes by Site



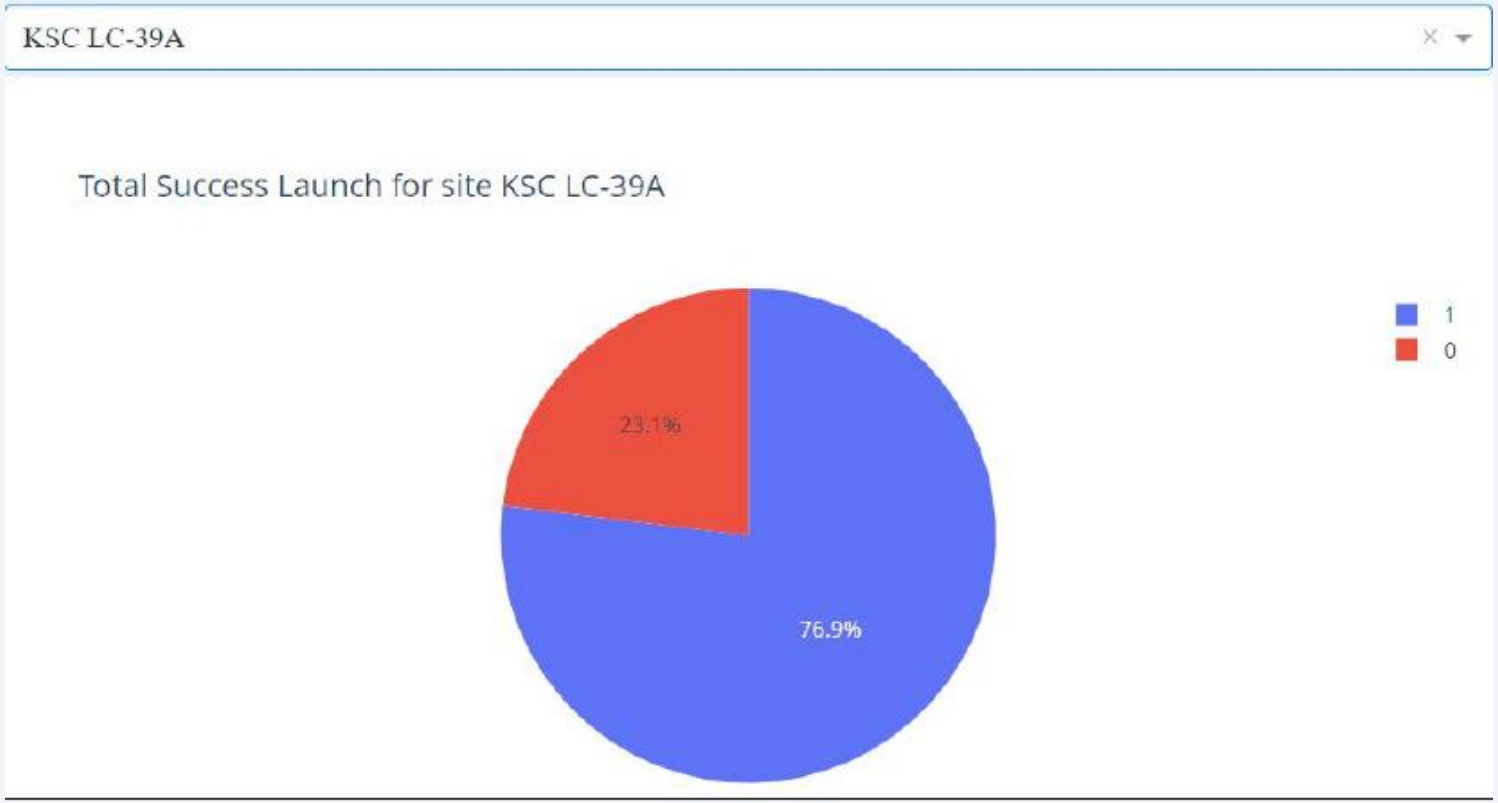
SpaceX Falcon 9 – Launch Site Proximity Distance Map



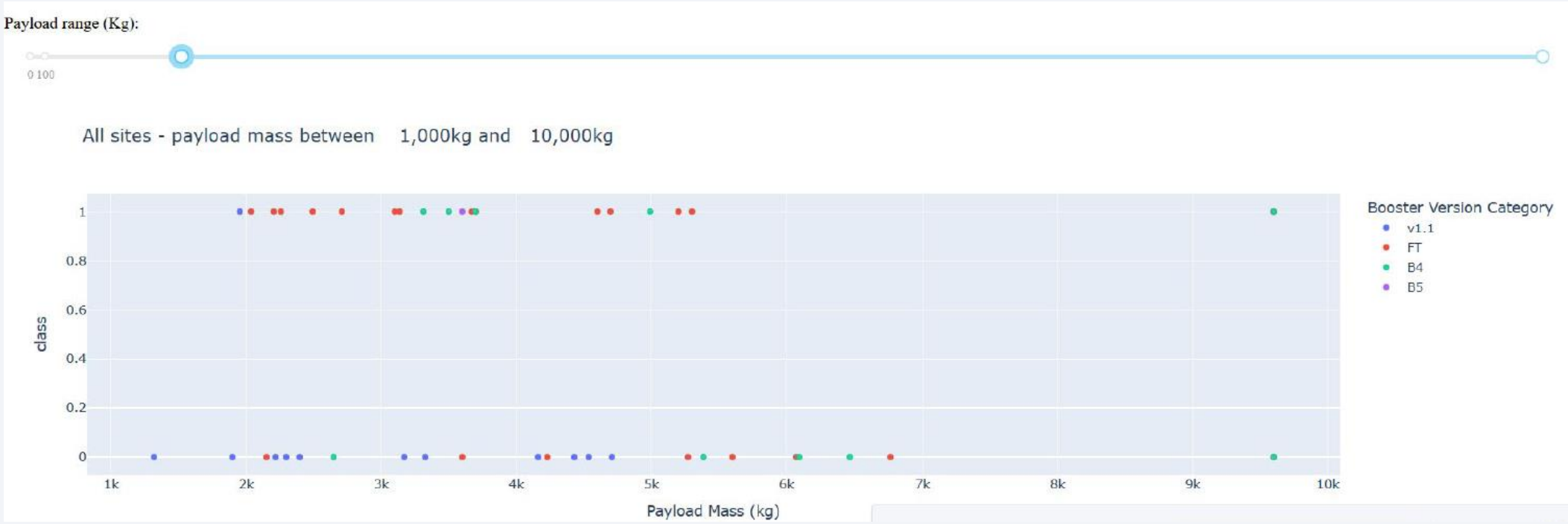

Section 4
Build a Dashboard with Plotly Dash

Launch Success Counts

Launch Site with Highest Launch Success Ratio



Payload vs. Launch Outcome Scatter Plot

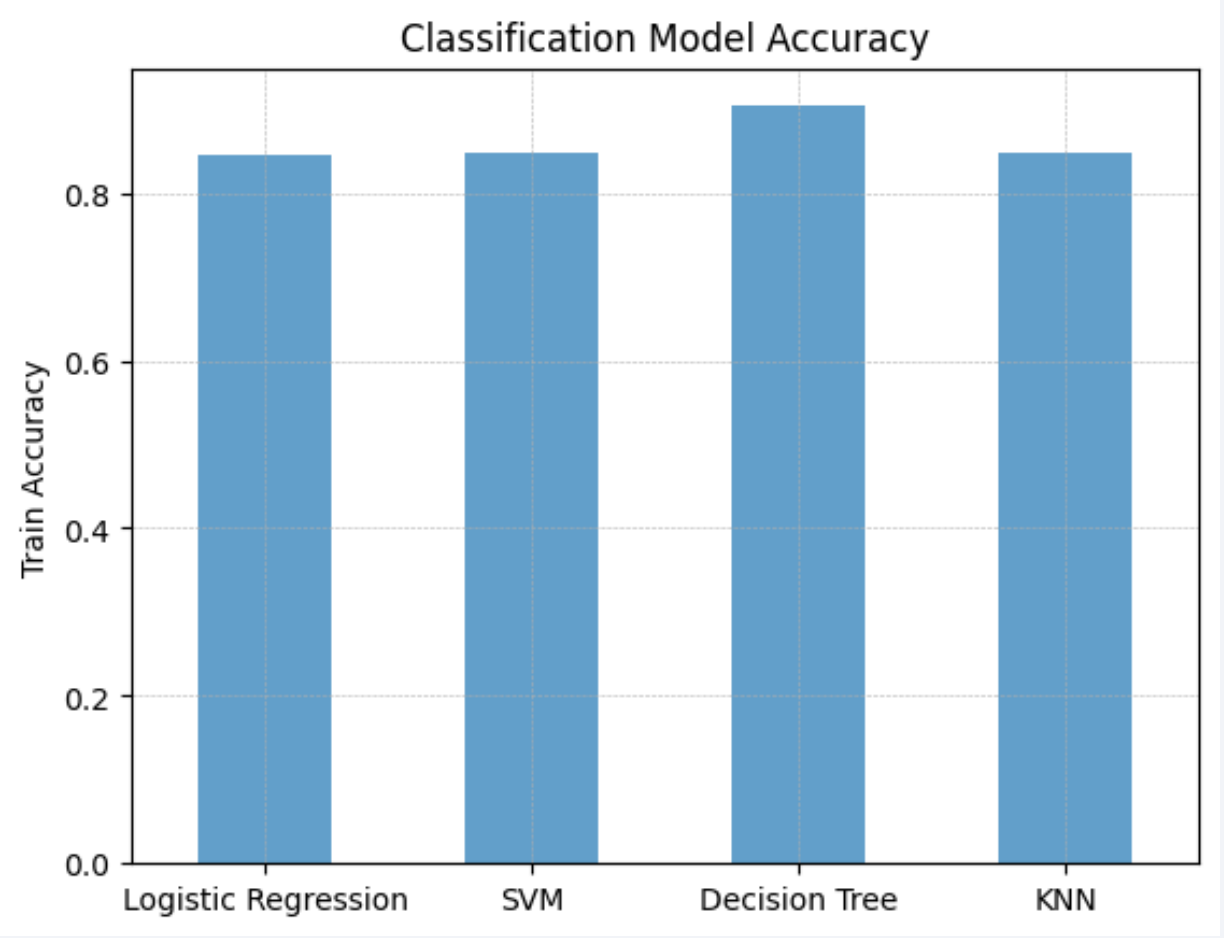


Section 5

Predictive Analysis (Classification)

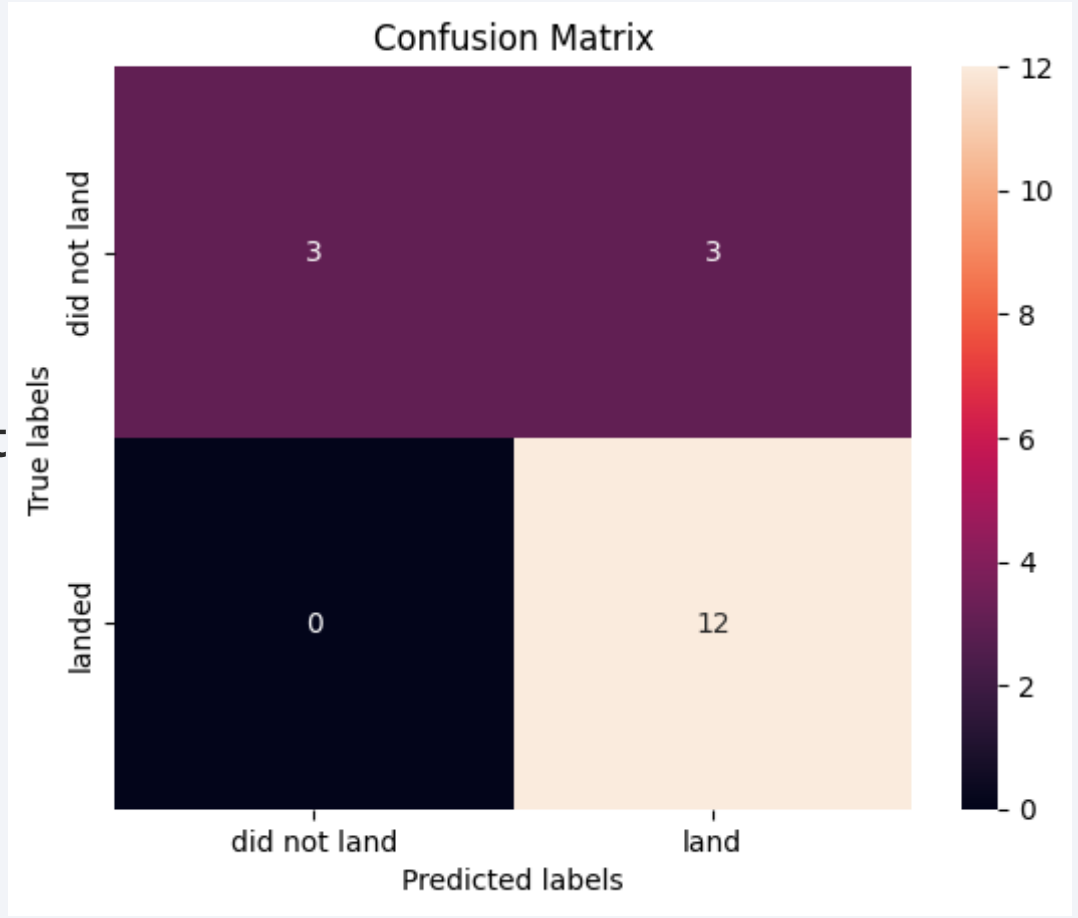
Classification Accuracy

- Decision Tree has the highest classification accuracy



Confusion Matrix

- The classifier made 18 predictions in total, consisting of 12 true positives, 3 true negatives, and 3 false positives. Overall, it achieved an accuracy of approximately 83%, with a misclassification rate of about 16.5%.



Conclusions

- The likelihood of first stage landing success improves as the number of flights increases.
- While higher payload mass generally aligns with better success rates, no strong correlation exists between the two.
- Launches with payloads over 7000 KG tend to be less risky.
- The decision tree model achieved higher classification performance compared to other machine learning models.
- Launch success rates rose by around 80% between 2013 and 2020.

Thank you!