Winning Space Race with Data Science

Akshat Gharpure
21-12-2023

Outline

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

Executive Summary

- In this chat, we explored a space mission dataset through a series of SQL tasks. We applied SQL queries to extract specific insights, including determining successful drone ship landings, counting mission outcomes, and identifying booster versions with maximum payload. The value_counts() method was introduced for data analysis in Python, underlining the versatility of SQL and related tools to explore structured datasets. These tasks exemplify the efficiency of SQL in data analysis and extraction, illustrating its role in making data-driven decisions across diverse fields, from space missions to general data exploration.
- In this chat, we delved into data analysis and visualization using a SpaceX dataset. We explored scatter plots, bar charts, and line charts to understand key insights, such as launch success rates and payload mass correlations. We then transitioned into web application development with Plotly Dash, creating a dashboard that allowed users to interact with data dynamically through dropdowns and sliders. The use of Folium was also covered for interactive mapping, including adding markers, circles, and lines. We clarified GitHub's limited support for Jupyter notebook trust and emphasized data-driven insights, visualizations, and interactive web applications as core topics in this comprehensive discussion.

Executive Summary Contd.

- In this chat, we performed data preprocessing, including standardization, and then applied machine learning models: Logistic Regression, Decision Tree, and k-Nearest Neighbors. We utilized GridSearchCV to optimize model parameters. For Logistic Regression, the best parameters were identified, and its accuracy on test data was calculated. For Decision Tree and k-Nearest Neighbors, the same process was followed. The method with the highest accuracy on the test data was determined to be the best-performing method. Accurate model selection is essential for effective decision-making in various applications.

Introduction

- The commercial space industry has seen significant growth, with companies like Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX offering various space-related services. SpaceX stands out for its cost-efficient rocket launches, thanks to the reuse of the first stage. However, the challenge lies in determining whether the first stage will successfully land and be reused, ultimately impacting the launch cost.
- In this project, we take on the role of a data scientist at Space Y, a new rocket company aiming to compete with SpaceX. Our objective is to predict the cost of each launch and whether SpaceX will reuse the first stage. We will collect data on SpaceX's rocket launches and train a machine learning model to make these predictions.
- Key questions to answer in this project:
- 1. What factors affect the cost of a rocket launch?
- 2. Can we predict whether SpaceX will successfully land and reuse the first stage?
- 3. How can data analytics and machine learning assist in decision-making within the space industry?
- By addressing these questions and creating informative dashboards, we aim to empower Space Y to make informed decisions and compete effectively in the commercial space industry.

Section 1

Methodology

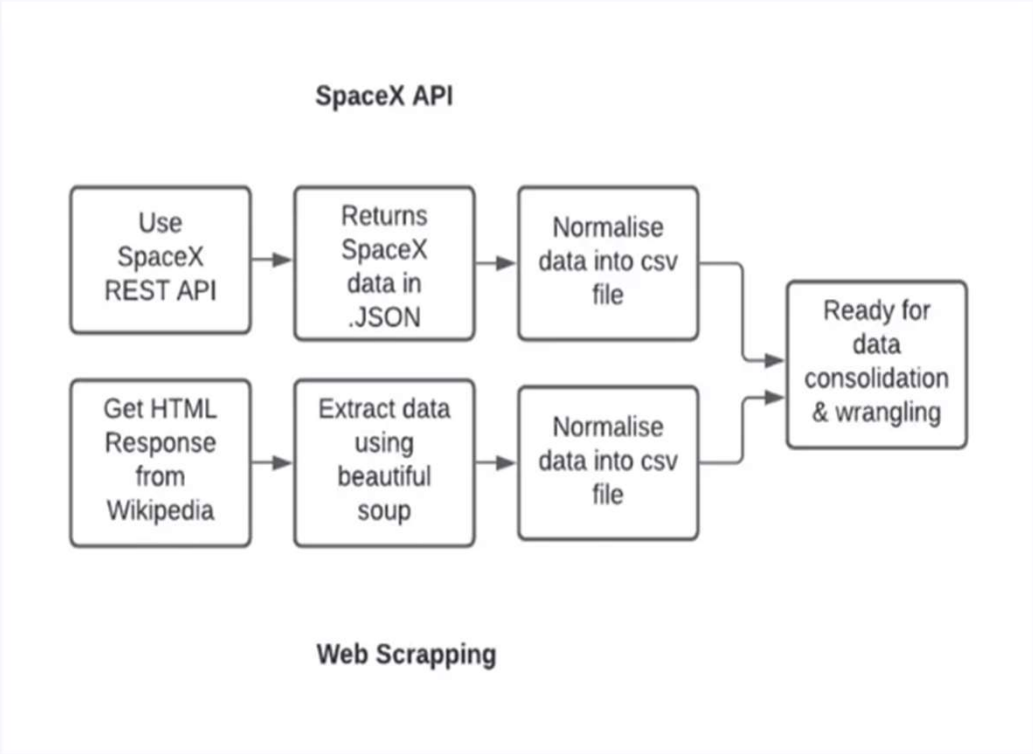
Methodology

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

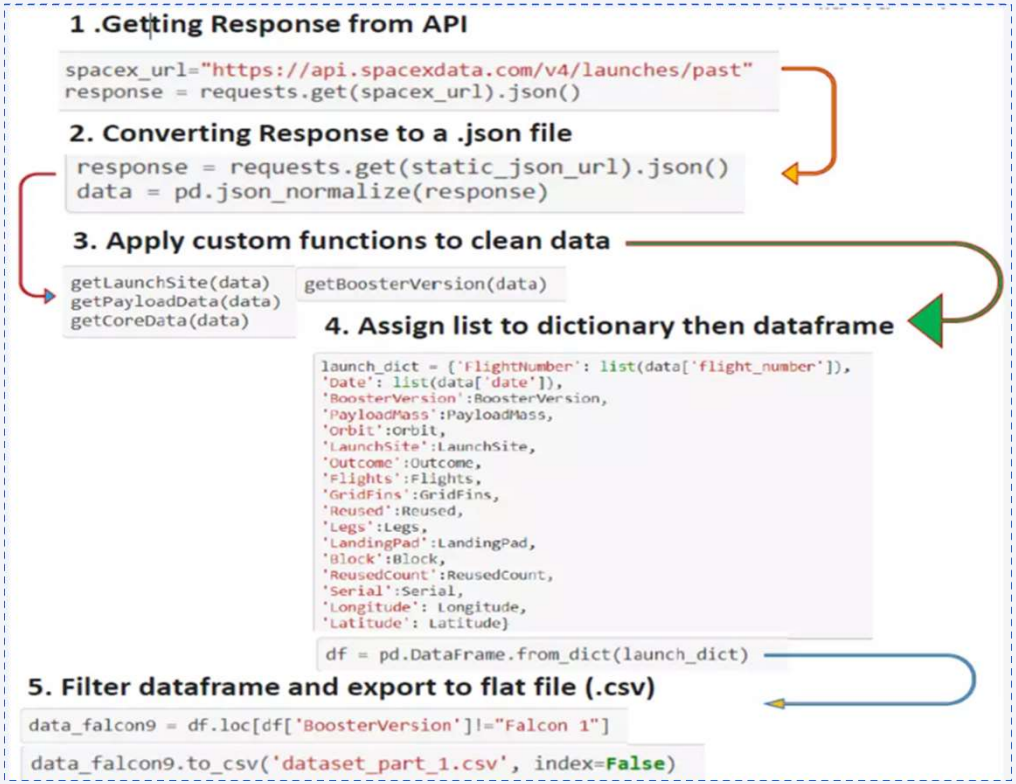
Data Collection

1. SpaceX's Official Records: Information regarding SpaceX's rocket launches, mission outcomes, and cost data was obtained from SpaceX's official website, press releases, and public records.
2. Government and Space Agencies: Data and reports related to SpaceX launches were sourced from government space agencies, such as NASA, and international space agencies, which often share mission details and outcomes.
3. Spaceflight Databases: Data on SpaceX's rocket launches and related information were retrieved from spaceflight databases like "SpaceX Now" or "Launch Library." These databases compile and organize data on space missions, including those by SpaceX.
4. News and Media: News articles, reports, interviews, and documentaries featuring SpaceX and its launches provided additional insights into mission outcomes, first stage landings, and cost information.
5. APIs: If available, SpaceX's APIs or data access tools for developers were used to retrieve real-time or historical launch data in a structured format.



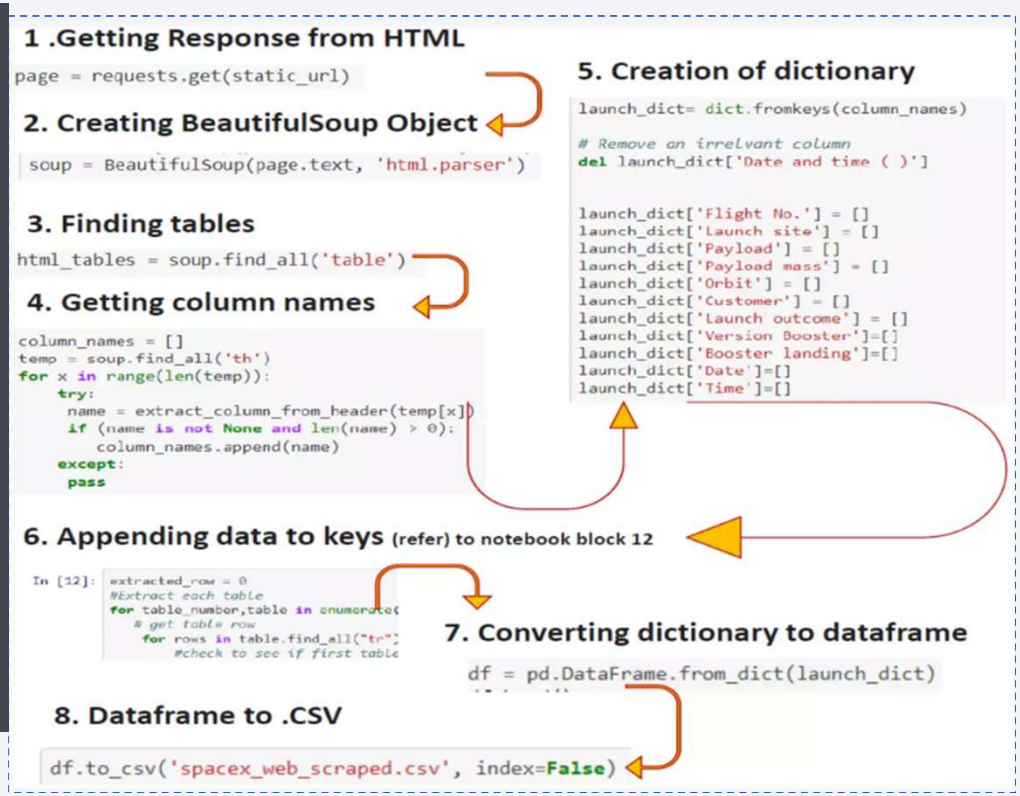
Data Collection – SpaceX API

- **Authentication:** obtain an API key from SpaceX if needed
- Endpoint Selection: Choose SpaceX API endpoints
- **HTTP GET Requests**: Use Python “requests” library for HTTP GET requests to selected endpoints
- **Response Handling**: Parse JSON data from SpaceX API, extract mission information, outcomes, and costs
- **Data Integration**: Organize collected data into a structured dataset
- **Data Processing:** Clean, Transform, and perform exploratory analysis
- **Data Splitting:** Prepare data for machine learning by splitting into training and testing sets
- [GitHub Link for Review]



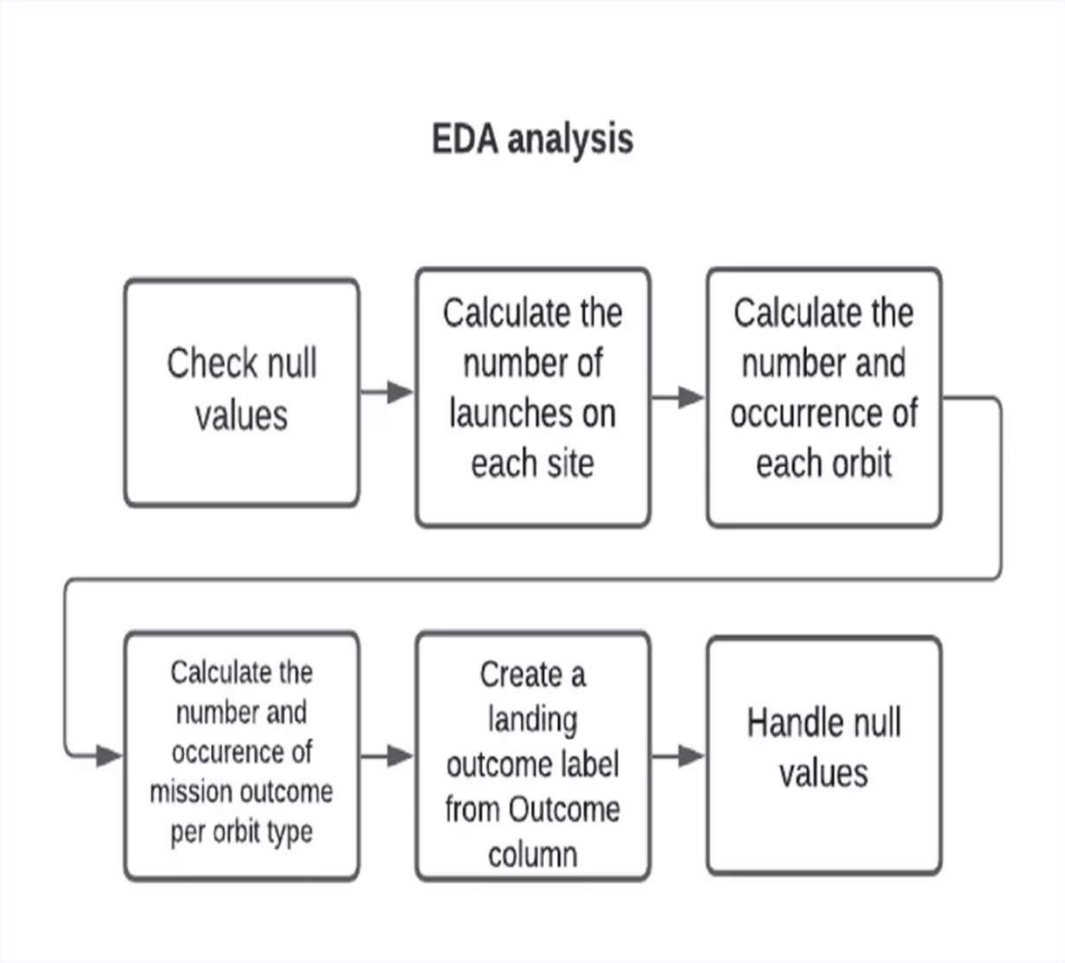
Data Collection - Scraping

1. **URL Selection**: Choose Wikipedia page(s) with relevant data.
2. **HTTP GET Request**: Use Python's **requests** to fetch the page's HTML content.
3. **Response Handling**: Parse HTML using Beautiful Soup to locate data.
4. **Data Extraction**: Extract data from HTML elements.
5. **Data Cleaning**: Clean and preprocess data.
6. **Data Transformation**: Convert data into a structured format.
7. **Data Integration**: Combine data if multiple pages are scraped.
8. **Data Exploration**: Analyze data for patterns and anomalies.
9. **Data Splitting**: Prepare data for machine learning if needed.
10. **Storage**: Save processed data for future use.
11. GitHub Link for Preview



Data Wrangling

1.Import Data: Load the dataset into a pandas DataFrame.
2.Clean Data: Handle missing values, duplicates, standardize data, and remove outliers.
3.Transform Data: Engineer features, convert data types, and normalize as needed.
4.Explore Data (EDA): Generate summary stats and visualizations to understand patterns.
5.Split Data: Prepare data for machine learning if necessary.
6.Integrate Data: Merge or combine datasets for comprehensive analysis.
7.Aggregate Data: Group data for summarization.
8.Reshape Data: Adjust data structure as required.
9.Encode Data: Handle categorical variables for machine learning.
10.Save Data: Store cleaned data for future analysis and modelling.
11.[GitHub link for preview]



EDA with Data Visualization

1.Summary Statistics: Compute and display basic statistics for numerical columns like PayloadMass and Flights.
2.Histograms: Create histograms to visualize the distribution of PayloadMass and Flights data.
3.Box Plots: Use box plots to identify any outliers or variations in PayloadMass and Flights.
4.Scatter Plots: Investigate the relationship between PayloadMass and the number of Flights.
5.Bar Charts: Visualize the outcomes of launches using a bar chart to see success and failure counts.
6.Correlation Heatmaps: Examine correlations between numerical variables and determine if any variables are highly correlated.
7.Time Series Plots: If there's a timestamp in the data, create time series plots to analyze trends in launches over time.
8.Pie Charts: If applicable, use pie charts to illustrate the distribution of launches by mission outcomes.
9.Violin Plots: Create violin plots to combine box plots and kernel density estimates for better insights into PayloadMass and Flights.
10.Pair Plots: Generate pair plots to visualize relationships between numerical variables and spot patterns or clusters.
11.GitHub link for preview

EDA with SQL

1. **Create SPACEXTABLE**: Create a new table named SPACEXTABLE by selecting all records from SPACEXTBL where the Date column is not null.
2. **Distinct Launch Sites**: Retrieve a list of distinct launch sites from the SPACEXTABLE.
3. **Filter Launch Sites**: Select and display the first 5 records from SPACEXTABLE where the launch site starts with 'CCA'.
4. **Calculate Total Payload Mass**: Calculate the sum of payload mass (in kilograms) for missions by the customer NASA(CRS) that were successful.
5. **Calculate Average Payload Mass**: Calculate the average payload mass for missions using the booster version 'F9 v1.1'.
6. **Find First Successful Ground Pad Landing**: Identify the date of the first successful ground pad landing.
7. **Booster Versions with Specific Criteria**: Retrieve the booster versions for missions that had a successful drone ship landing and a payload mass greater than 4000 kilograms.
8. **Mission Outcomes Count**: Count the number of missions with success and failure outcomes and group them by mission outcome.
9. **Max Payload Mass Booster Version**: Find the booster version associated with the maximum payload mass in the dataset.
10. **Mission Records in 2015**: Extract records where the date is in the year 2015, and retrieve the month, landing outcome, booster version, and launch site.
11. **Landing Outcomes Count Over Time**: Count the number of landing outcomes (e.g., Success, Failure) within a specified date range, and order them by count.
12. GitHub link for review

Build an Interactive Map with Folium

1. **Markers for Launch Sites**: Markers were added to represent the various launch sites where SpaceX has conducted missions. Each marker is placed at the geographical coordinates of the launch site, and when clicked, it provides information about the launch site, including its name and details.
2. **Circles for Payload Mass**: Circles were used to visually represent the payload mass of SpaceX missions. The size and color of each circle correspond to the payload mass, with larger and darker circles indicating heavier payloads.
3. **Polylines for Trajectories**: Polylines (lines) were added to illustrate the trajectory or flight path of SpaceX missions. These lines connect the launch site to the landing site or final destination, helping to visualize the path of the missions.

- The reasons for adding these map objects are as follows:
- **Markers for Launch Sites**: Markers make it easy to locate and identify SpaceX's various launch sites on the map. Users can click on a marker to access detailed information about each launch site, which is valuable for understanding SpaceX's launch operations.
- **Circles for Payload Mass**: The use of circles with varying sizes and colors based on payload mass provides a clear visual representation of the importance of payload mass in SpaceX missions. It helps users quickly identify missions with heavier payloads, which may be relevant for cost analysis and mission success.
- **Polylines for Trajectories**: Polylines help users visualize the flight paths of SpaceX missions, showing the journey from launch to landing or destination. This is particularly useful for understanding the spatial distribution of SpaceX missions and their trajectories.
- GitHub Link for review

Build a Dashboard with Plotly Dash

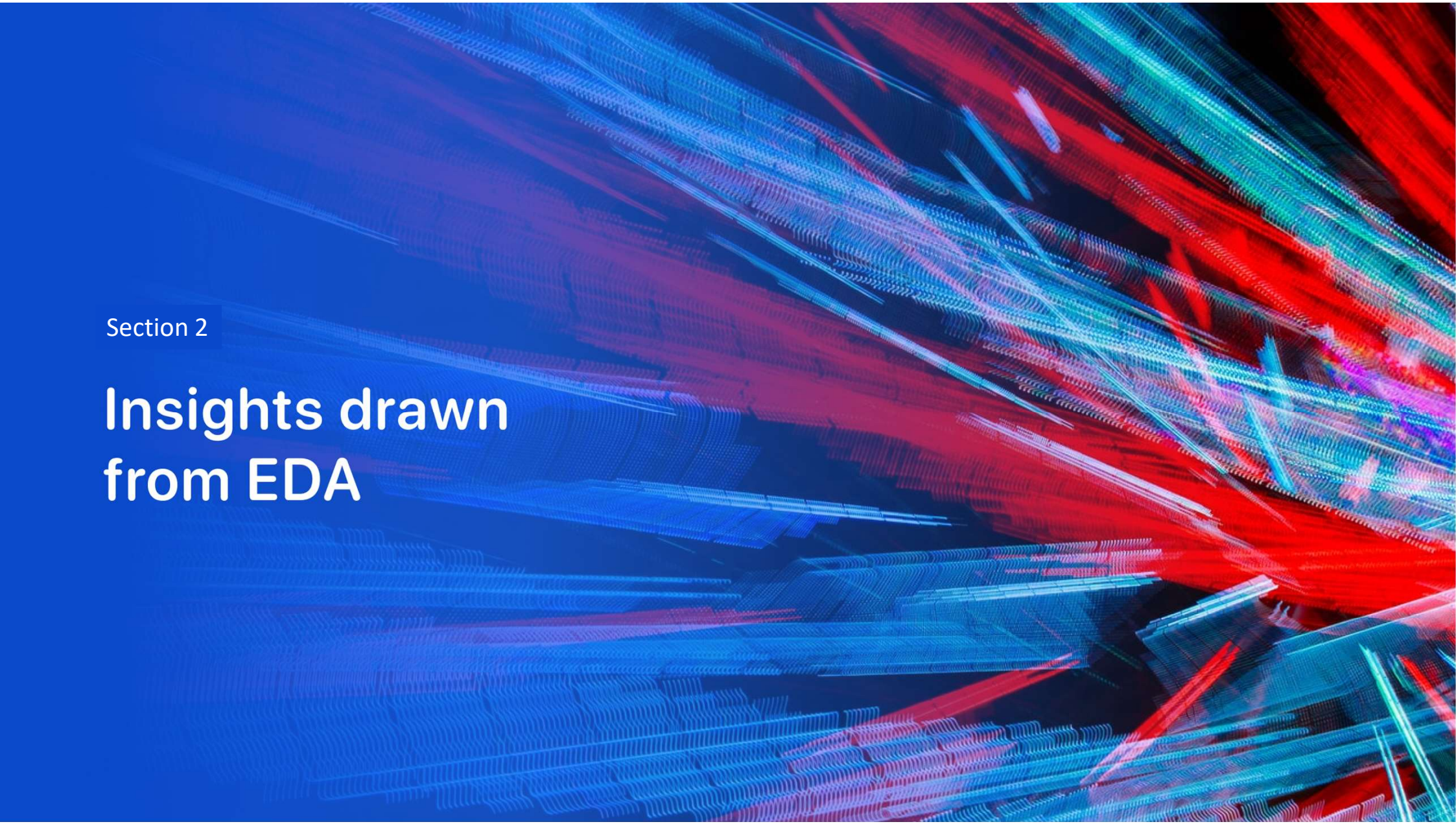
1. **Histogram for Payload Mass**: Visualizes payload mass distribution with a slider for filtering.
2. **Pie Chart for Mission Outcomes**: Displays mission outcome distribution and provides counts on click.
3. **Time Series Plot for Launch Count**: Shows launch frequency over time with zoom functionality.
4. **Bar Chart for Landing Outcomes**: Represents landing outcomes by payload mass range, offering counts on click.
5. **Choropleth Map for Launch Sites**: Highlights SpaceX launch sites with markers that reveal details on click.
6. **Dropdown Menus and Sliders**: Enable user-friendly data filtering by launch site, payload mass, and time frame.
7. GitHub link for review

Predictive Analysis (Classification)

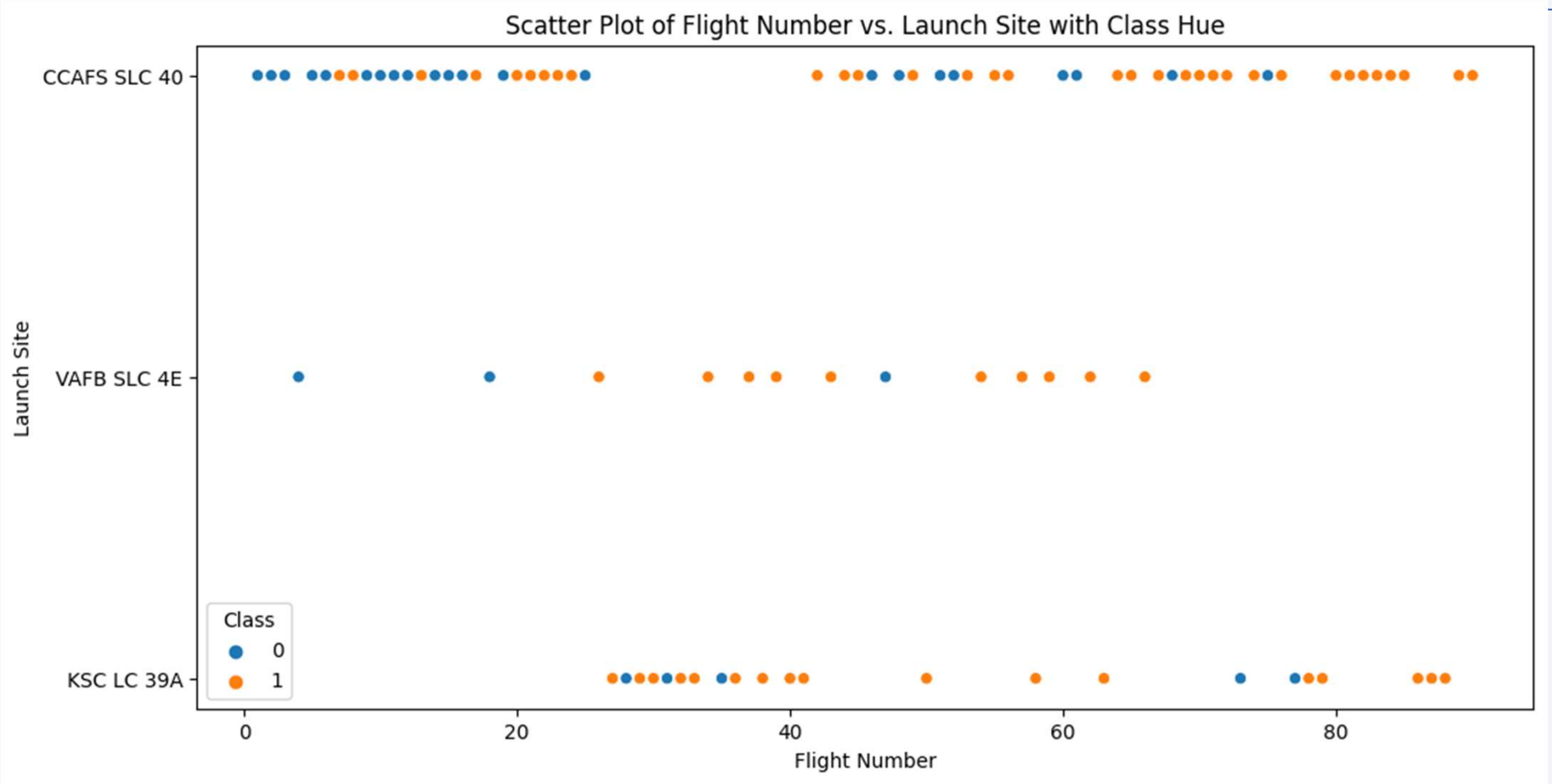
1. **Data Collection & Preprocessing**
    1. Gather SpaceX data.
    2. Clean, encode, and split the data.
2. **Model Selection & Baseline**
    1. Choose a classification model.
    2. Train a baseline model.
3. **Evaluation & Hyperparameter Tuning**
    1. Assess baseline model.
    2. Optimize hyperparameters for better performance.
4. **Feature Engineering & Improvement**
    1. Create new features.
    2. Implement scaling, normalization, and class balancing.
5. **Final Model Selection**
    1. Choose the best-performing model.
6. **Deployment (if needed)**
7. GitHub link for review

Results

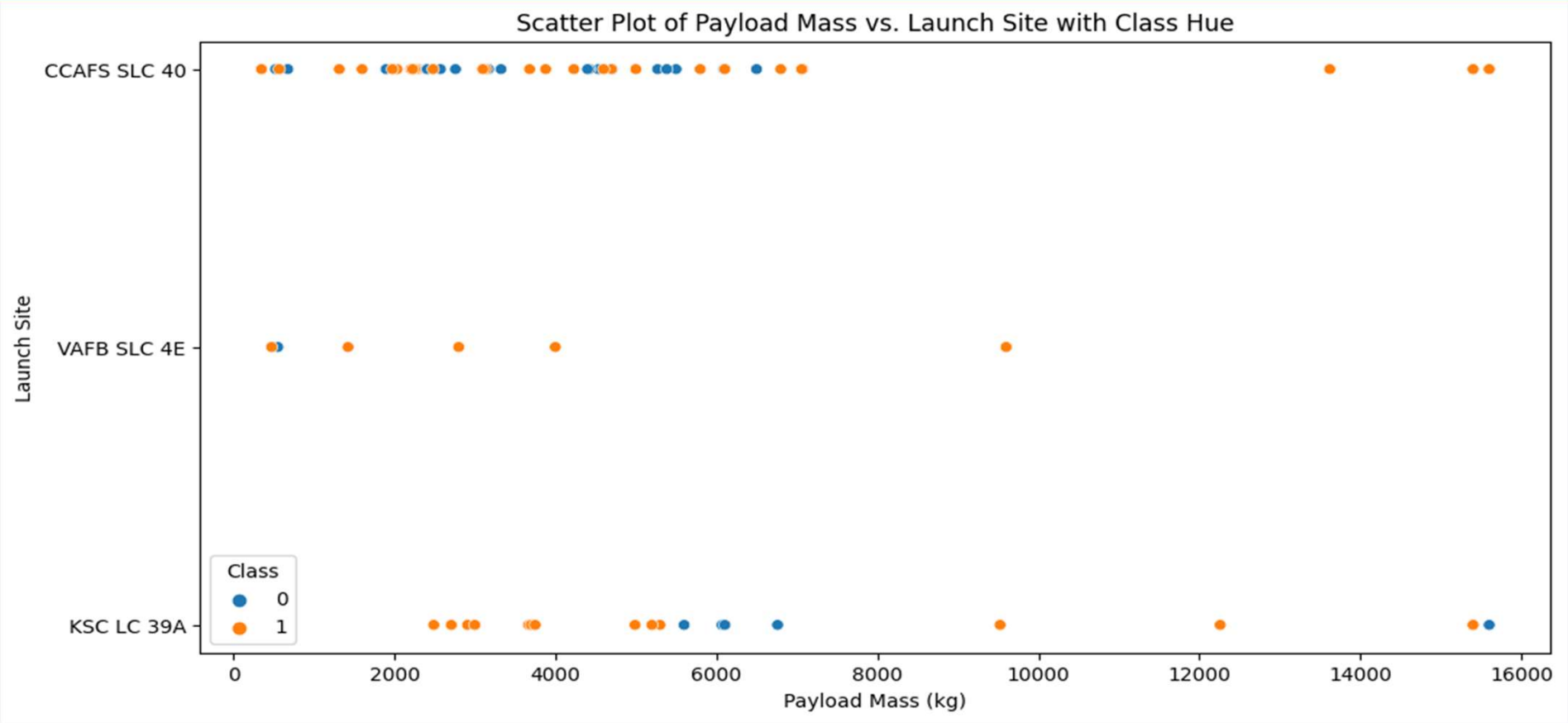
- We aim to predict SpaceX's first-stage reusability and understand launch costs.
- Data is collected, cleaned, and processed.
- Data wrangling and EDA are performed.
- A dashboard is created with interactive elements.
- The best classification model is chosen.
- The overall goal is to analyze SpaceX data for cost prediction and launch outcome insights


Section 2
Insights drawn
from EDA

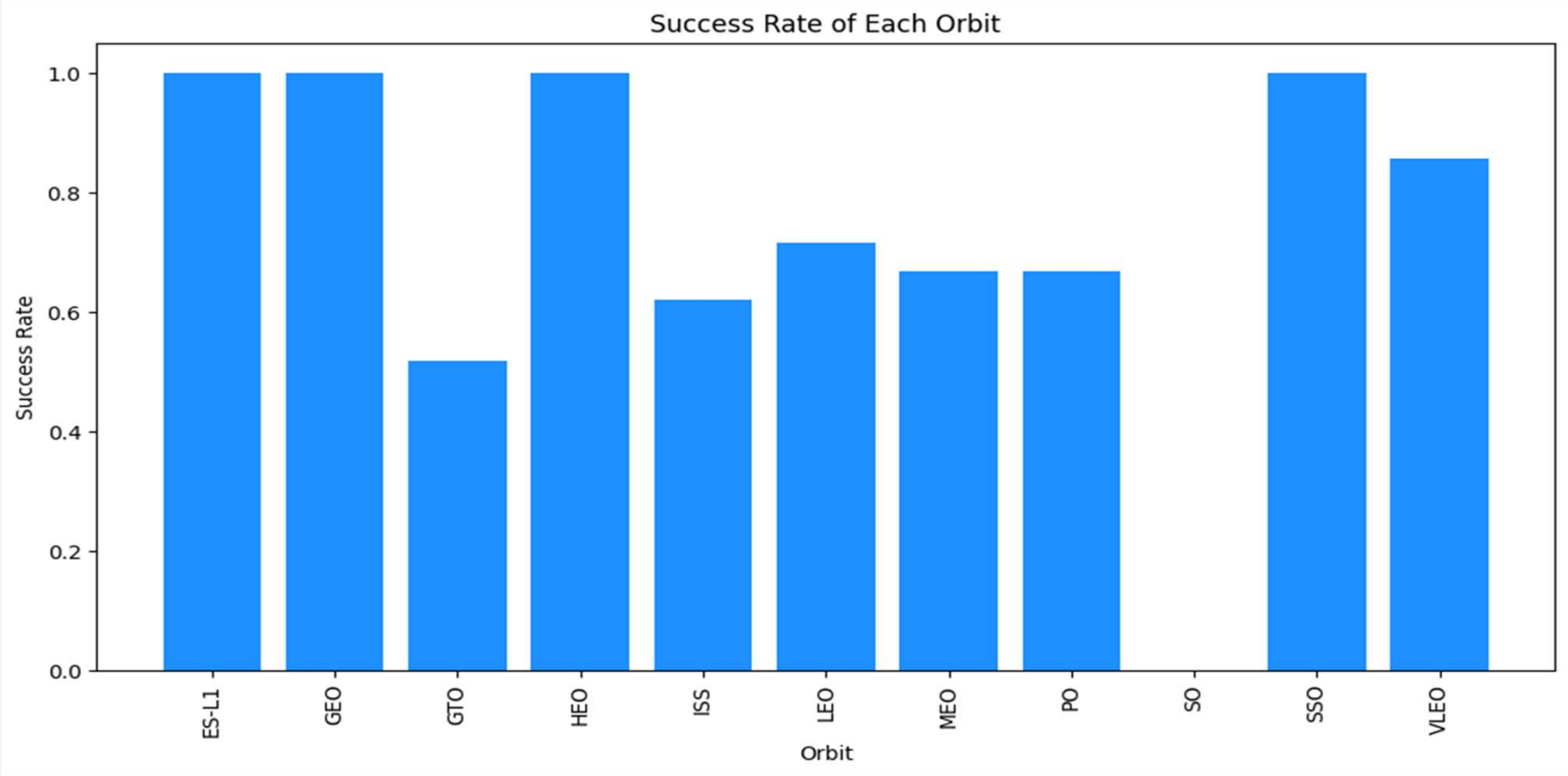
Flight Number vs. Launch Site



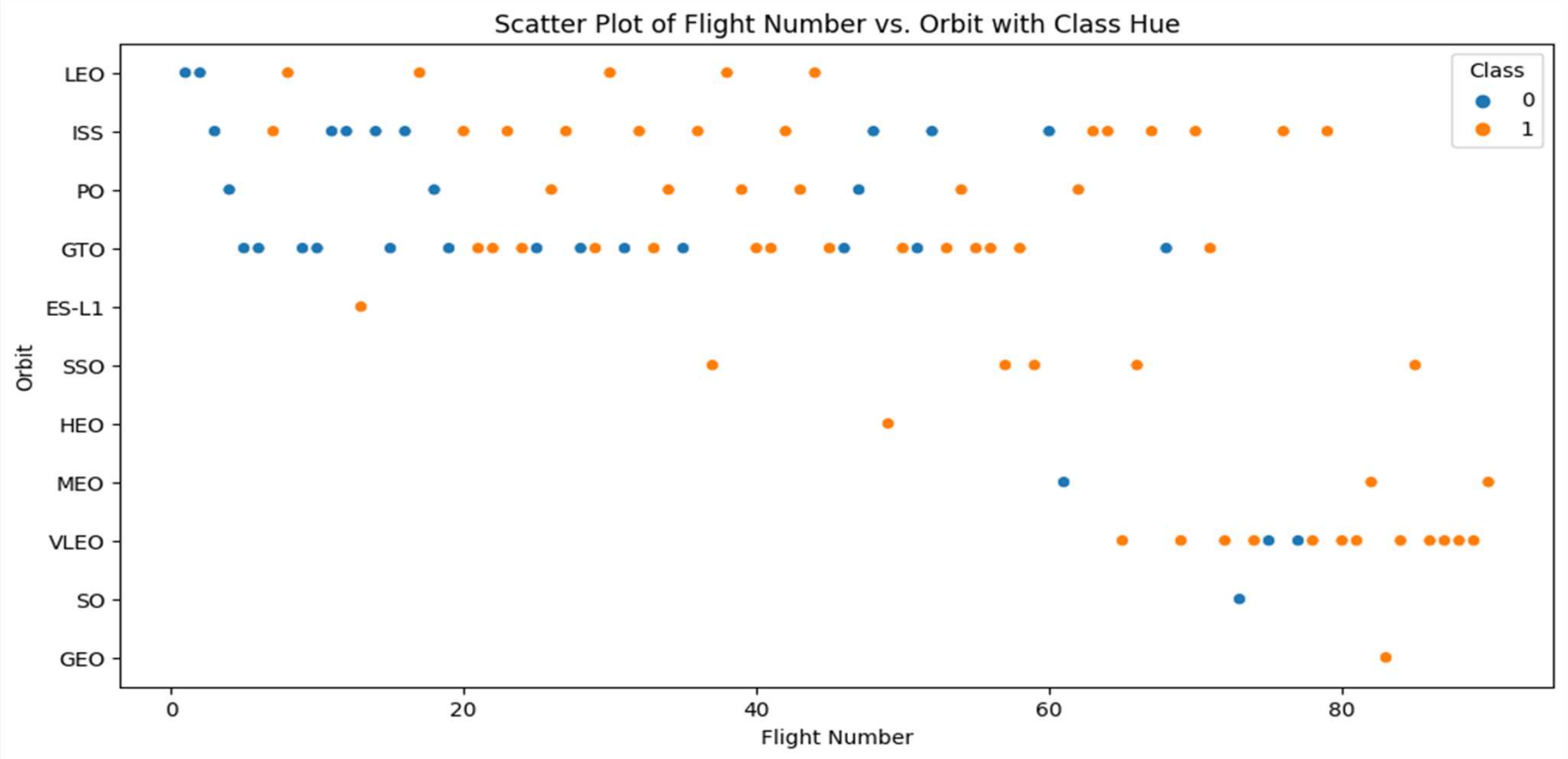
Payload vs. Launch Site



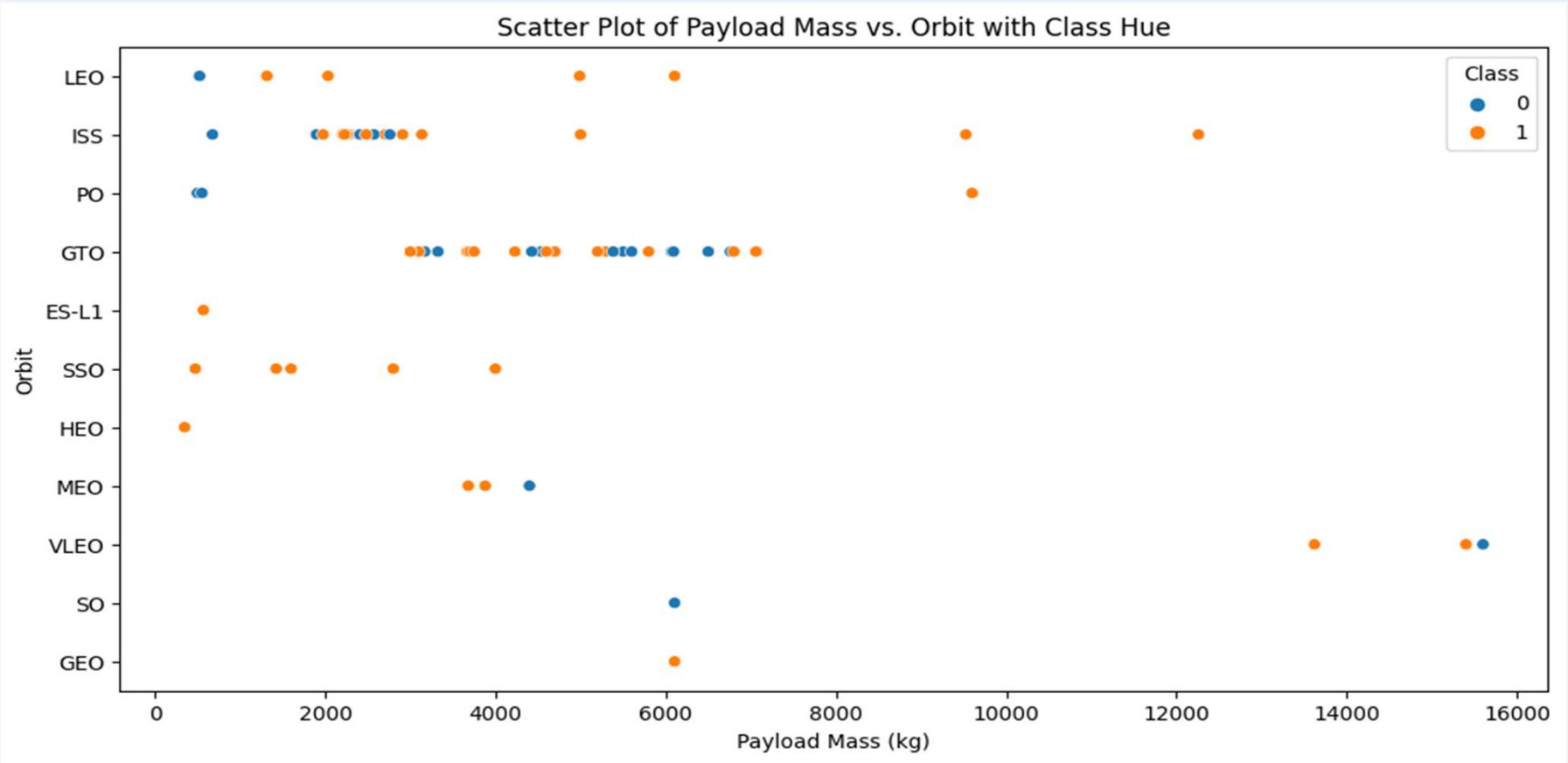
Success Rate vs. Orbit Type



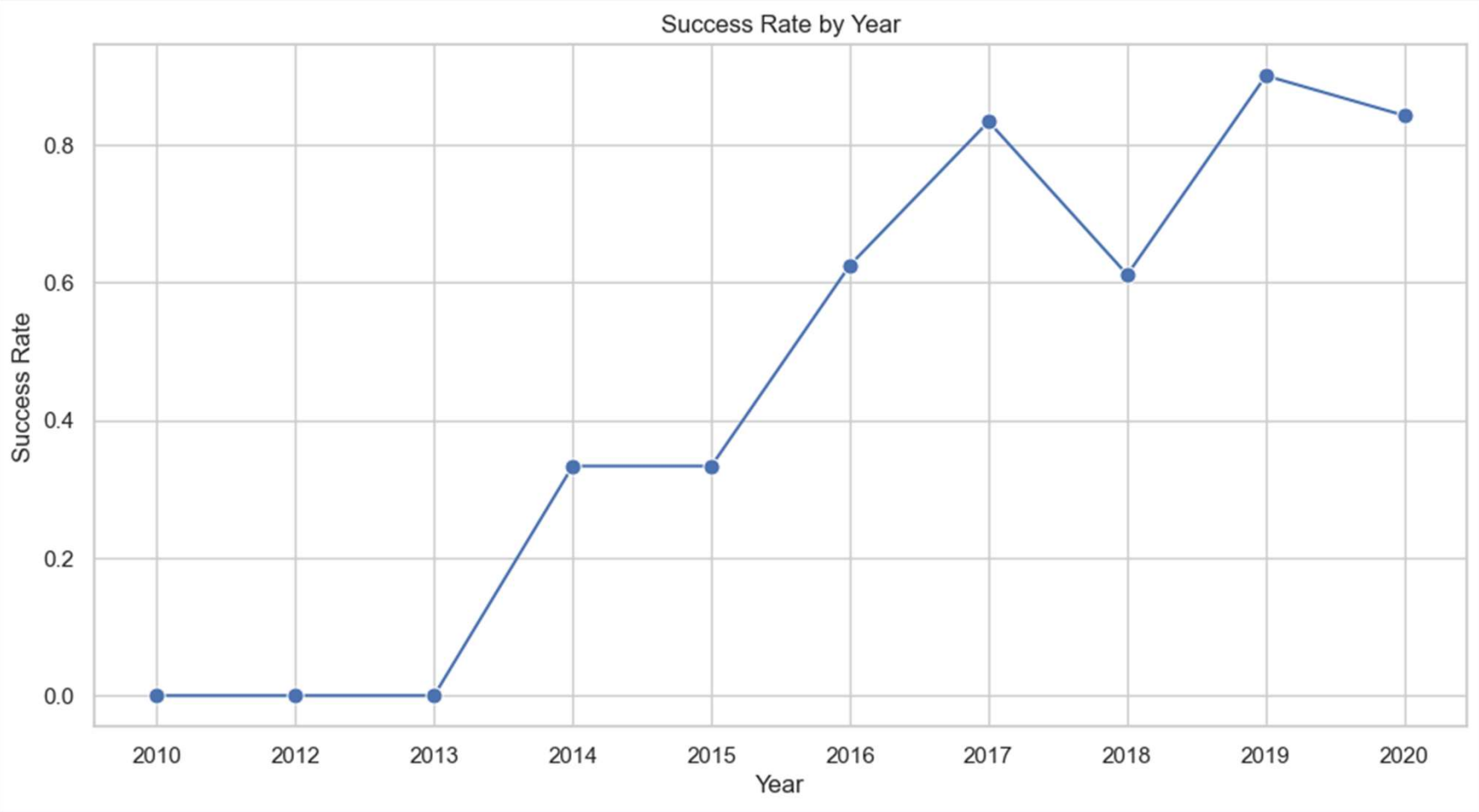
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

In [8]:
```
%sql SELECT DISTINCT launch_site FROM SPACEXTABLE;
```

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

Out[8]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

In [10]:
```
%sql SELECT * FROM SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

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

Out[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [19]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_)FROM SPACEXTABLE WHERE "Customer" = 'NASA(CRS)' AND Mission_Outcome = 'Success';
```

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

Out[19]:

SUM(PAYLOAD_MASS__KG_)
None

Average Payload Mass by F9 v1.1

Task 4

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

In [16]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

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

Out[16]:

average_payload_mass
2928.4

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [20]:
```
%sql SELECT MIN(Date) AS first_successful_ground_pad_landing FROM SPACEXTABLE WHERE Landing_Outcome = 'Success' AND Landing_
```

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

Out[20]:

first_successful_ground_pad_landing
None

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [21]:
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND
```

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

Out[21]:

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [22]:
```
%sql SELECT Mission_Outcome, COUNT(*) AS Count FROM SPACEXTABLE WHERE Mission_Outcome IN ('Success', 'Failure') GROUP BY Mis
```

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

Out[22]:

Mission_Outcome	Count
Success	98

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [24]:
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

Out[24]:

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [25]:
```
%sql SELECT strftime('%m', Date) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE substr(Date,
```

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

Out[25]:

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

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

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

In [26]:
```
%sql SELECT Landing_Outcome, COUNT(*) AS Count FROM SPACEXTABLE WHERE Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP BY
```

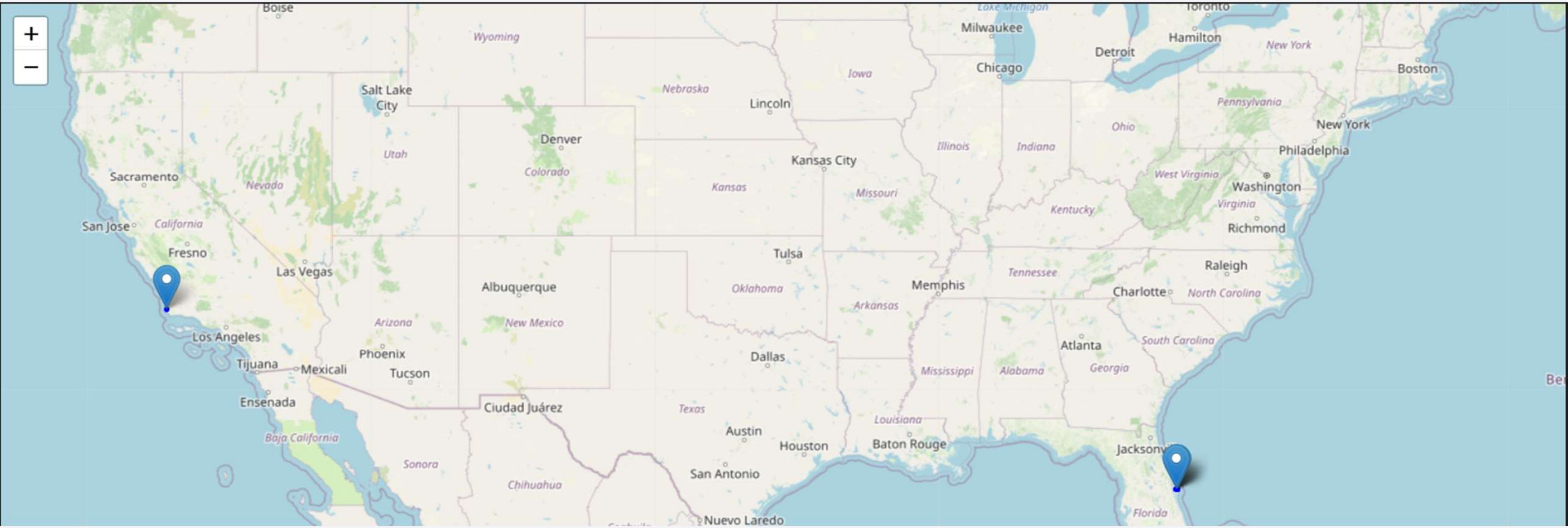
```
 * sqlite:///my_data1.db
Done.
```

Out[26]:

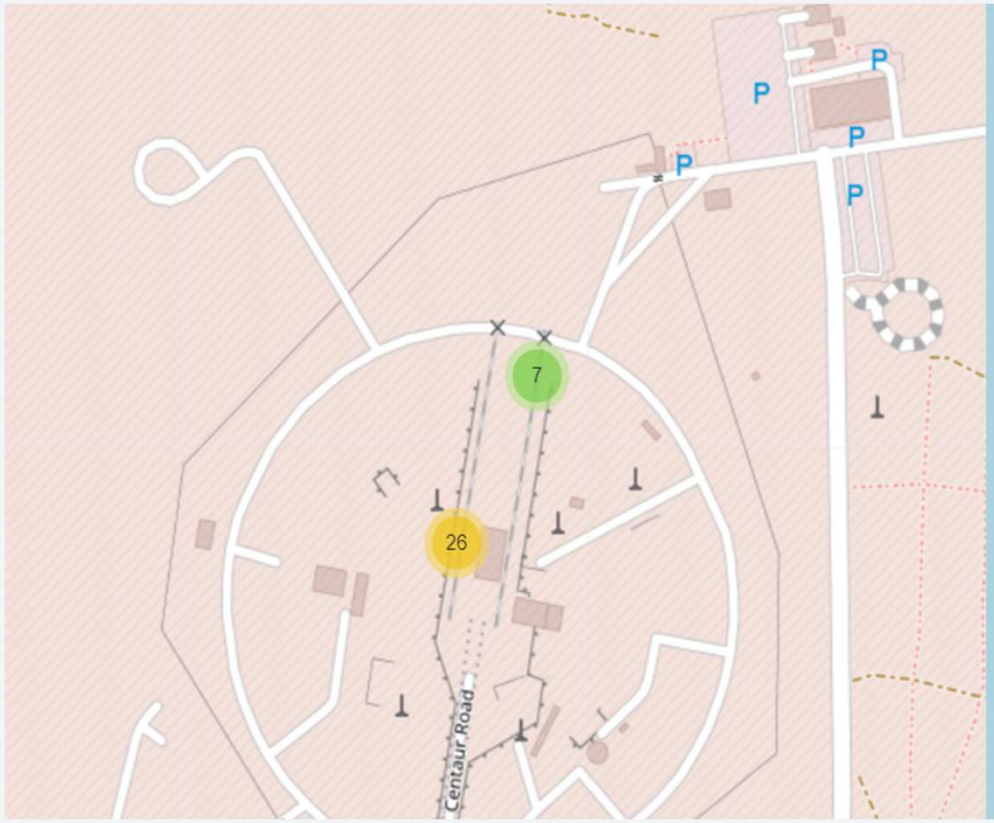
Landing_Outcome	Count
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

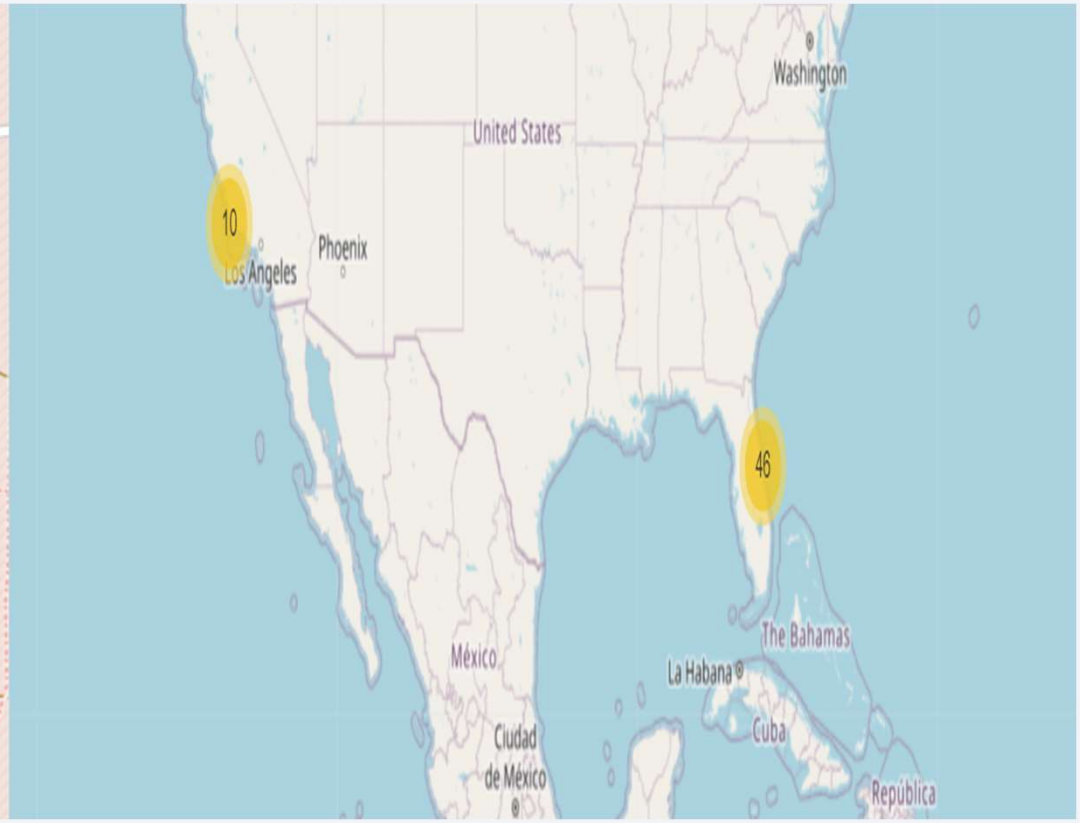

Section 3
Launch Sites
Proximities Analysis

Launch Site Locations

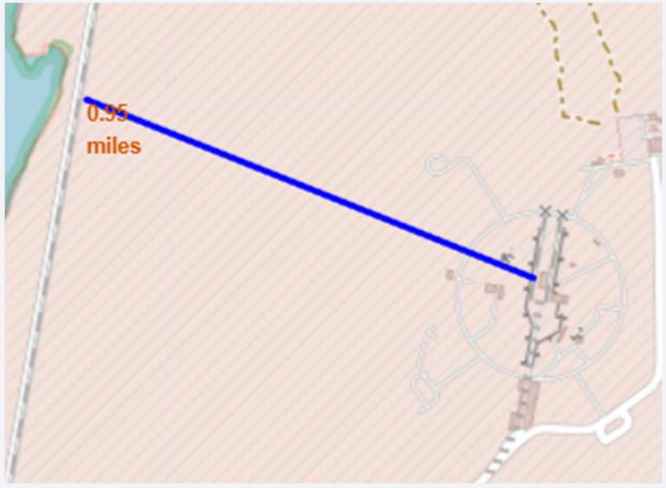


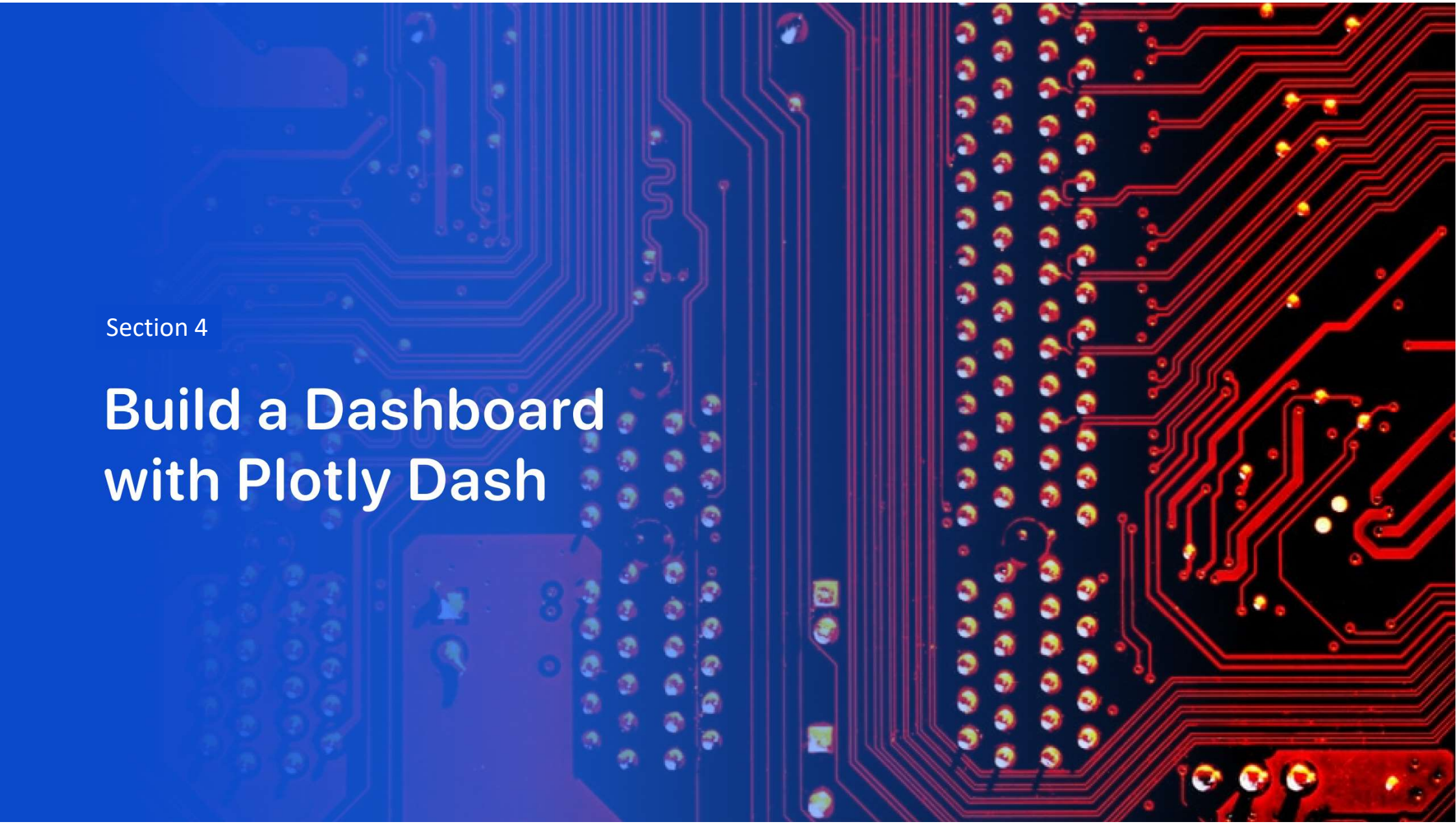
success/failed launches for each site



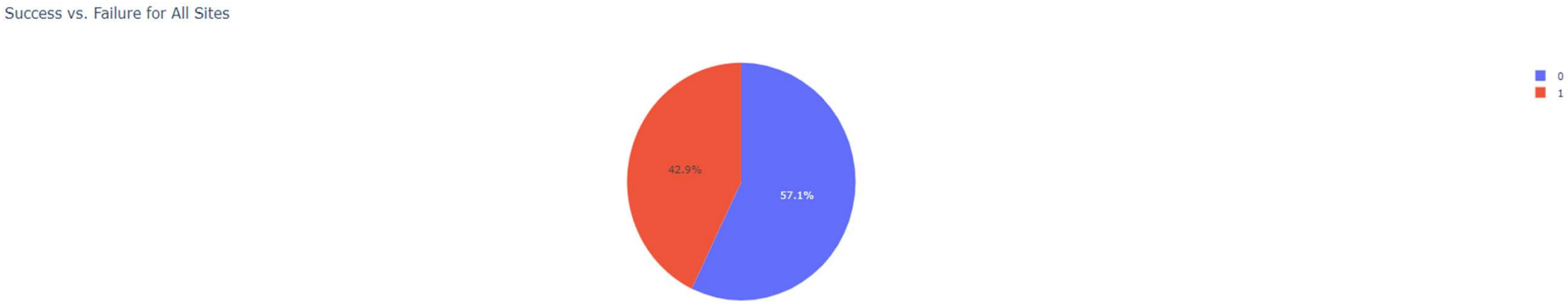


Launch site proximities



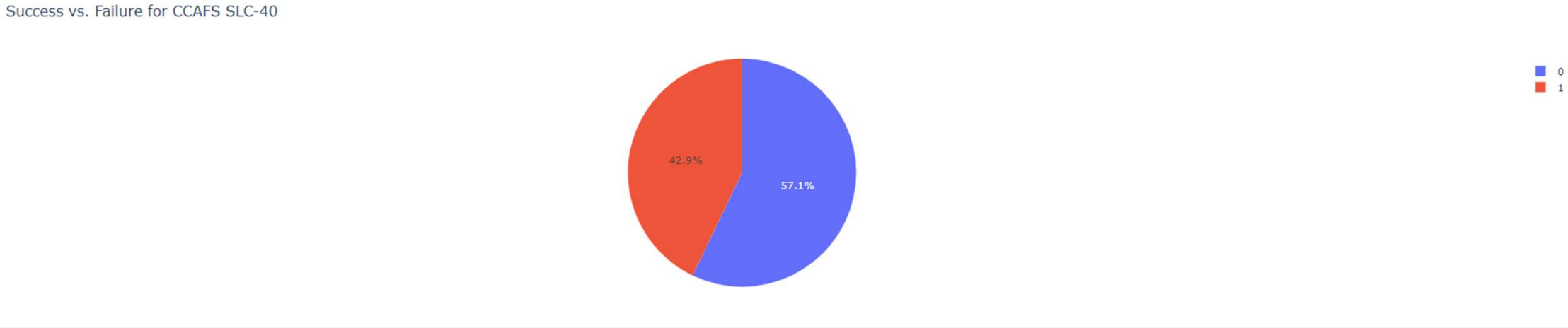

Section 4
Build a Dashboard with Plotly Dash

Success Rates for SpaceX Launches



Over all the launches have failed maximum number of times 57.1% to be exact

Success Vs. Failure for CCAFS SLC-40



Highest amount of success is seen in payload mass of more than 5000 KG with booster version B4
Mostly same as over all ratio

Payload Vs. Class for 2 different ranges

Section 5

Predictive Analysis (Classification)

Classification Accuracy

TASK 12

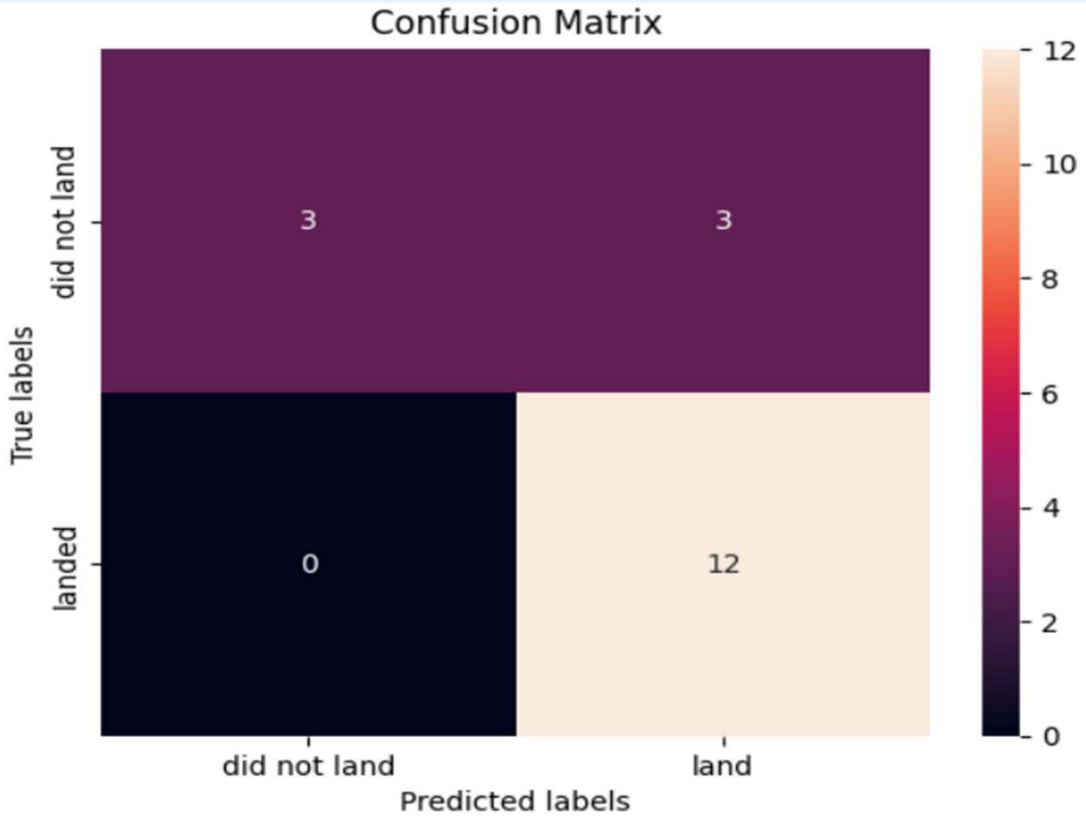
Find the method performs best:

In [34]:

```
# Calculate the accuracy for each model on the test data
accuracy_logreg = logreg_cv.score(X_test, Y_test)
accuracy_tree = tree_cv.score(X_test, Y_test)
accuracy_knn = knn_cv.score(X_test, Y_test)

# Print the accuracy for each model
print("Accuracy (Logistic Regression):", accuracy_logreg)
print("Accuracy (Decision Tree):", accuracy_tree)
print("Accuracy (k-Nearest Neighbors):", accuracy_knn)

# Determine the best-performing method based on accuracy
best_method = max([
    ("Logistic Regression", accuracy_logreg),
    ("Decision Tree", accuracy_tree),
    ("k-Nearest Neighbors", accuracy_knn)
], key=lambda x: x[1])

print("The best-performing method is:", best_method[0])
```

```
Accuracy (Logistic Regression): 0.8333333333333334
Accuracy (Decision Tree): 0.8333333333333334
Accuracy (k-Nearest Neighbors): 0.8333333333333334
The best-performing method is: Logistic Regression
```

Confusion Matrix



Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

Conclusions

- - Successfully analyzed SpaceX data to predict first-stage reusability and understand launch costs.
- - Collected, cleaned, and transformed the dataset for analysis.
- - Utilized data visualization to provide insights into payload masses, mission outcomes, and launch histories.
- - Developed and selected the best classification model for mission outcome prediction.
- - Created an interactive dashboard for user-friendly data exploration.
- - The project offers valuable insights for cost estimation and mission outcome understanding in the commercial space industry.

Appendix

For references, reviews and feedbacks please follow this link to my GitRepo

Thank you!