SpaceY

Rocket Reuse Analysis

Mahmoud Ibrahim

01-09-2024

OUTLINE

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

Executive Summary

Summary of methodologies

1. **Data Collection:** Gathered data from SpaceX REST API and Web Scraping.
2. **Data Preparation:** Cleaned and explore data to identify any patterns that could define labels.
3. **Statistical Analysis:** Understand how the data distributed using SQLite.
4. **Data Visualization:** Standard plots and charts for visualizing the trends in the data.
5. **Interactive Visualization:** For interactives maps and dashboards using Folium and Dash Plotly.
6. **Machine Learning:** To predict launch success by developing appropriate models

Exploratory Data Analysis:

• Launch success has improved over time • KSC LC-39A has the highest success rate among landing sites

• Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate

Visualization/Analytics:

• Most launch sites are near the equator and all near the coast

Predictive Analytics:

• All models have similar performances on the test set. Decision tree model slightly outperforms

Introduction

With more access to space travel, commercial spaceflight has now arrived. Leading companies include SpaceX, Blue Origin, Virgin Galactic, and Rocket Lab. Of the group, the most successful ones include SpaceX: launching spacecraft to the International Space Station, establishing the Starlink constellation of satellite internet, and conducting manned space missions. What has been the main reason for its success is its reusability of the first stage of Falcon 9 rockets, which brought down their launch cost drastically.

The aim of this project is to gain better insights into the trends and features causing the success of space missions by analyzing launch data from SpaceX and building predictive models for identifying the success of rocket landings using historical data.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - Collecting data using SpaceX rest Api and other web scraping techniques
- **Perform data wrangling**
  - Cleaned the data, handle missing values and apply one-hot-encoding to prepare the date for the next stage
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - Develop appropriate classification models by choosing relevant algorithms, optimizing hyperparameters, and evaluating different metrics with regards to model performances such as accuracy, precision, and recall.

Data Collection

- The data was collected using various methods
- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

Key Phrases:

1-Request data: Made a request towards SpaceX API.

2-JSON Decoding: Decoded response content using JSON.

3-Custom Functions: Requested launch information using custom functions.

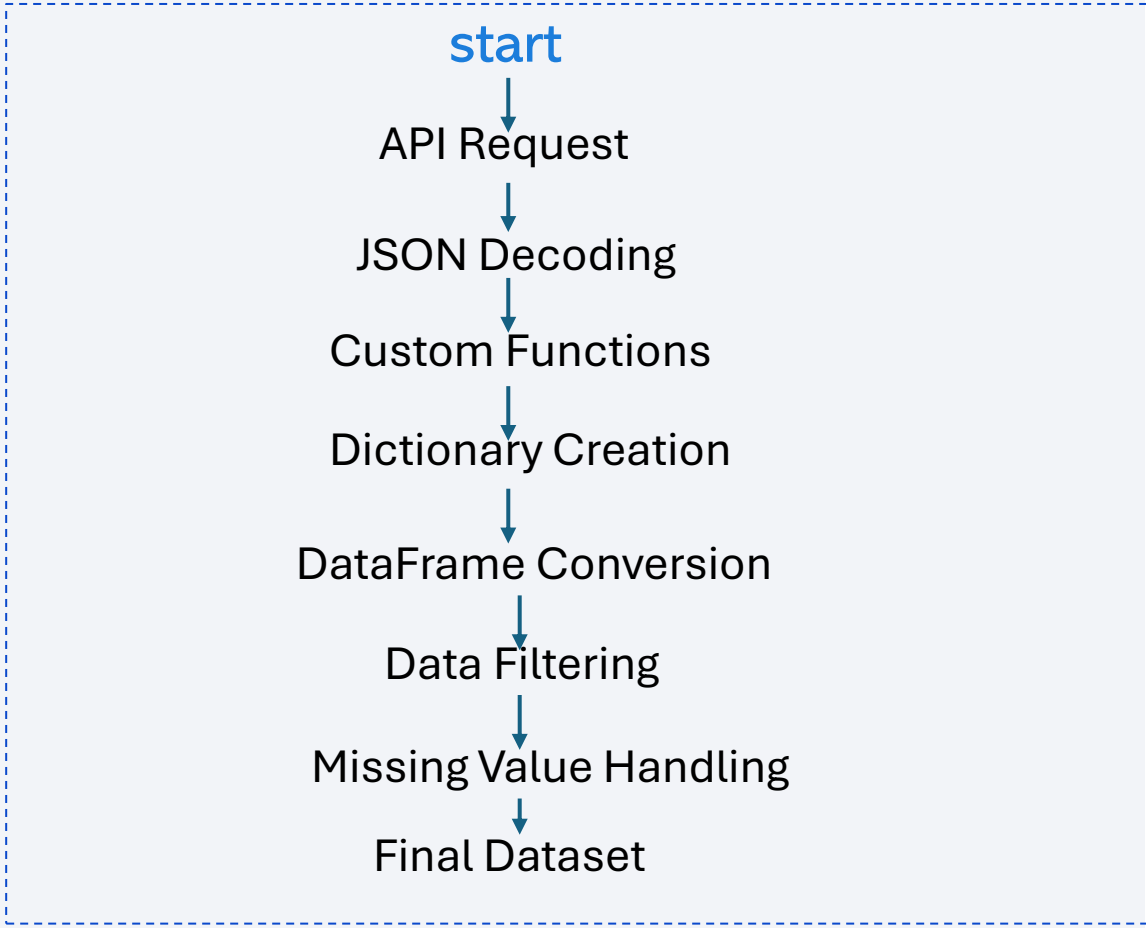
4-Dictionary Creation: Created a dictionary from the data.

5-DataFrame: Transformed the dictionary to pandas DataFrame.

6-Data Filtering: Filtered the DataFrame for Falcon 9 launches only.

7-Missing Value Handling: Used mean to fill empty in Payload Mass.

https://github.com/Mhdibrahim01/AppliedDS/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

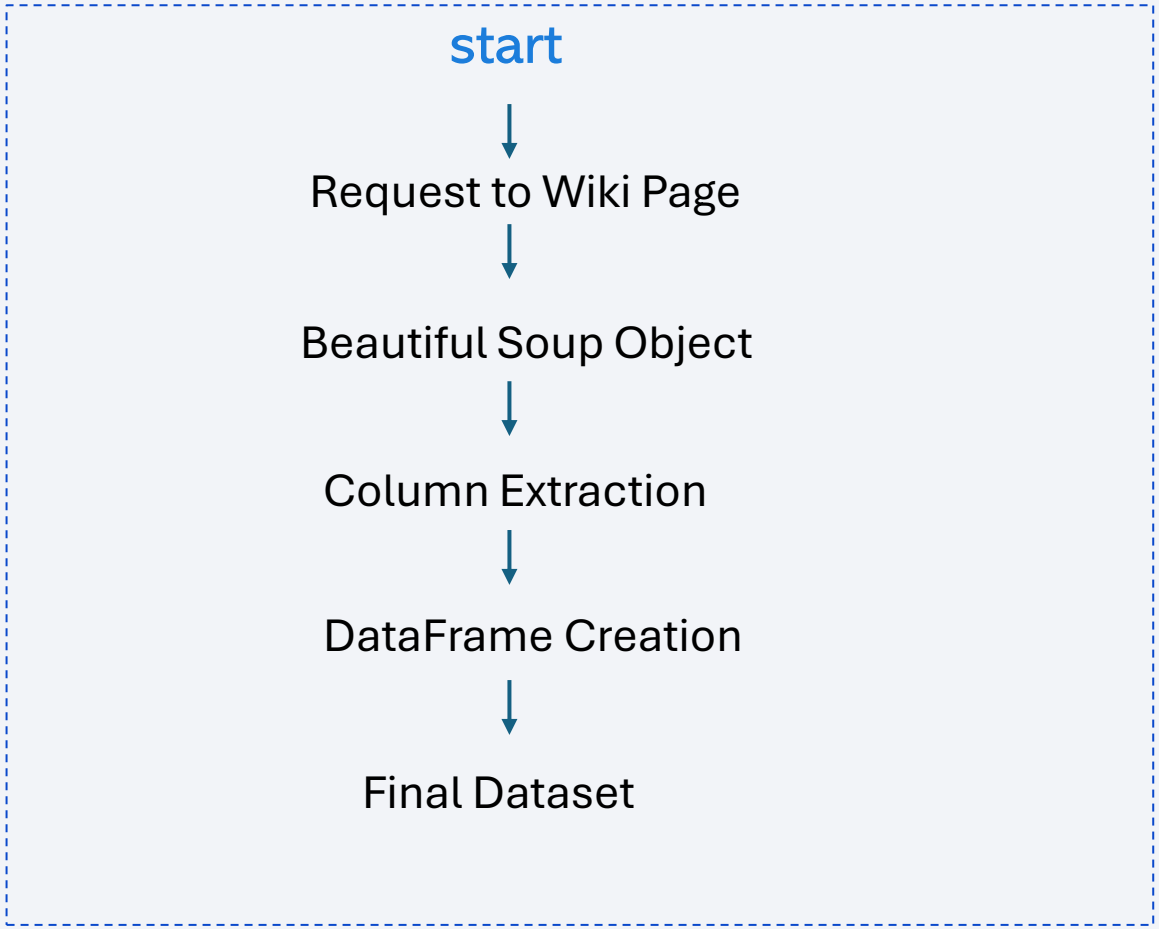


Data Collection - Scraping

Key Phrases:

1. **Request to Wiki Page:** Made a request to the Wikipedia page about Falcon 9 rocket launches.
2. **Beautiful Soup Object:** Created a Beautiful Soup object from the response.
3. **Column Extraction:** Extracted all column names from the HTML table headers.
4. **DataFrame Creation:** Parsed the HTML tables and created a DataFrame.

https://github.com/Mhdibrahim01/Applied-DS/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- **Key Phrases:**

1. **Exploratory Data Analysis (EDA):** Conducted EDA to understand the data.
2. **Missing Values:** Identified and calculated the percentage of missing values in each attribute.
3. **Data Types:** Identified numerical and categorical columns.
4. **Launch Count by Site:** Calculated the number of launches at each site.
5. **Orbit Occurrence:** Calculated the number and occurrence of each orbit.
6. **Mission Outcome Analysis:** Calculated the number and occurrence of mission outcomes for each orbit.
7. **Landing Outcome Label:** Created a landing outcome label from the Outcome column.

https://github.com/jonrfoss/IBM-Data-Science-Capstone-SpaceX/blob/main/03_SpaceX_Data_Wrangling.ipynb

EDA with Data Visualization

Flight Number vs Payload Mass (Scatter Plot):
Visualize the relationship between flight numbers and payload mass to identify any patterns or trends.
Flight Number vs Launch Site (Scatter Plot):
Examine how flight numbers are distributed across different launch sites.
Payload Mass (kg) vs. Launch Site (Scatter Plot):
Explore the distribution of payload mass for different launch sites to see if there are any site-specific trends.
Orbit vs Success Rate (Bar Chart):
Compare the success rates across different orbit types to determine which orbits are associated with higher success rates.
Flight Number vs Orbit (Scatter Plot):
Analyze how flight numbers correlate with different orbit types to identify any patterns.
Payload vs Orbit Type (Scatter Plot):
Investigate the relationship between payload size and orbit type to understand how payloads vary with different orbits.
Year vs Average Success Rate (Line Plot):
Track the average success rate over the years to observe trends and improvements in launch success rates.

https://github.com/Mhdibrahim01/Applied-DS/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Loaded the SpaceX data into SQLite.
- Displayed the unique launch site names in the space missions.
- Retrieved 5 records where launch sites begin with the string 'CCA’.
- Displayed the total payload mass carried by boosters launched by NASA (CRS).
- Calculated the average payload mass carried by the booster version F9 v1.1.
- Listed the date when the first successful landing outcome on a ground pad was achieved.
- Listed the names of boosters that succeeded on a drone ship with a payload mass between 4000 and 6000 kg.
- Counted the total number of successful and failed mission outcomes.
- Listed the booster versions that carried the maximum payload mass using a subquery.
- Displayed records showing the month names, failed landing outcomes on a drone ship, booster versions, and launch sites for the months in the year 2015.
- Ranked the count of landing outcomes (e.g., Failure on drone ship, Success on ground pad) between the dates 2010-06-04 and 2017-03-20 in descending order.

https://github.com/Mhdibrahim01/Applied-DS/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Mapping Launch Sites and Key Details

- **Marking Launch Sites:**
  - I started by placing markers and circles on the map to clearly show where each launch site is located. Adding labels made it easy to identify these spots at a glance.
- **Highlighting Launch Outcomes:**
  - To visualize which launches succeeded and which failed, I created clusters of markers for each site. I used green markers for successful launches and red for failures. This way, anyone looking at the map can instantly see the track record of each location.
- **Measuring Distances:**
  - I also wanted to understand how close the launch sites are to important landmarks. First, I marked the nearest coastline and drew a line from the launch site to this point, showing the exact distance. Then, I added markers for the closest city, railway, and highway, with lines connecting them to the launch site. This gives a complete picture of how accessible and strategically located each site is.

https://github.com/Mhdibrahim01/Applied-DS/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Dashboard Elements and Their Purpose

- **Dropdown for Launch Sites:**
  - I added a dropdown menu to allow users to select a specific launch site or view data for all sites. This gives users the flexibility to focus on a particular site or get an overview of all sites, making the dashboard more interactive and user-friendly.
- **Payload Slider:**
  - The payload slider enables users to choose a specific range of payload mass (in KG) to filter the data. This helps in analyzing how different payload sizes impact launch success or failure, allowing for a more tailored analysis based on user input.
- **Pie Chart for Success/Failure Rates:**
  - The pie chart visualizes the success and failure rates for the selected launch site. If "All Sites" is selected, it shows the success rate for each site. This chart provides a quick and clear understanding of the success rate distribution, making it easy to compare outcomes across different sites.
- **Scatter Plot for Payload vs. Success/Failure:**
  - I included a scatter plot to display the correlation between payload mass and launch outcomes (success or failure). This plot helps users see how payload size affects the likelihood of a successful launch, either for a specific site or across all sites, offering insights into the factors influencing launch success.

https://github.com/Mhdibrahim01/Applied-DS/blob/main/spacex.py

Predictive Analysis (Classification)

Model Development Process

- **Data Preparation:**
  - I started by creating a numpy array from the class column and assigned it to the target variable Y. Then, I standardized the feature columns in X to ensure consistency in the data.
- **Splitting the Data:**
  - Next, I split the data into training and test sets, allowing me to train the models on one part of the data and evaluate them on another.
- **Model Creation and Evaluation:**
  - I created logistic regression, support vector machine, decision tree classifier, and k-nearest neighbor models. For each model, I used GridSearchCV to find the best parameters, calculated the model score, identified the best parameter score, and evaluated the results using confusion matrices. I also plotted these results to visualize performance.
- **Finding the Best Model:**
  - After comparing the performance of all models, I found that the decision tree classifier slightly outperformed the others, making it the best-performing method in this case.

https://github.com/Mhdibrahim01/AppliedDS/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Exploratory Data Analysis

- **Launch Success Trends:** Launch success rates have shown improvement over time.
- **Top-Performing Site:** KSC LC-39A stands out with the highest success rate among all landing sites.
- **Orbit Success:** Certain orbits, such as ES-L1, GEO, HEO, and SSO, have achieved a 100% success rate.

Visual Analytics

- **Geographical Insights:** Most launch sites are strategically located near the equator and coastlines, minimizing risks to populated areas (city, highway, railway) while ensuring proximity to essential resources for launch operations.

Predictive Analytics

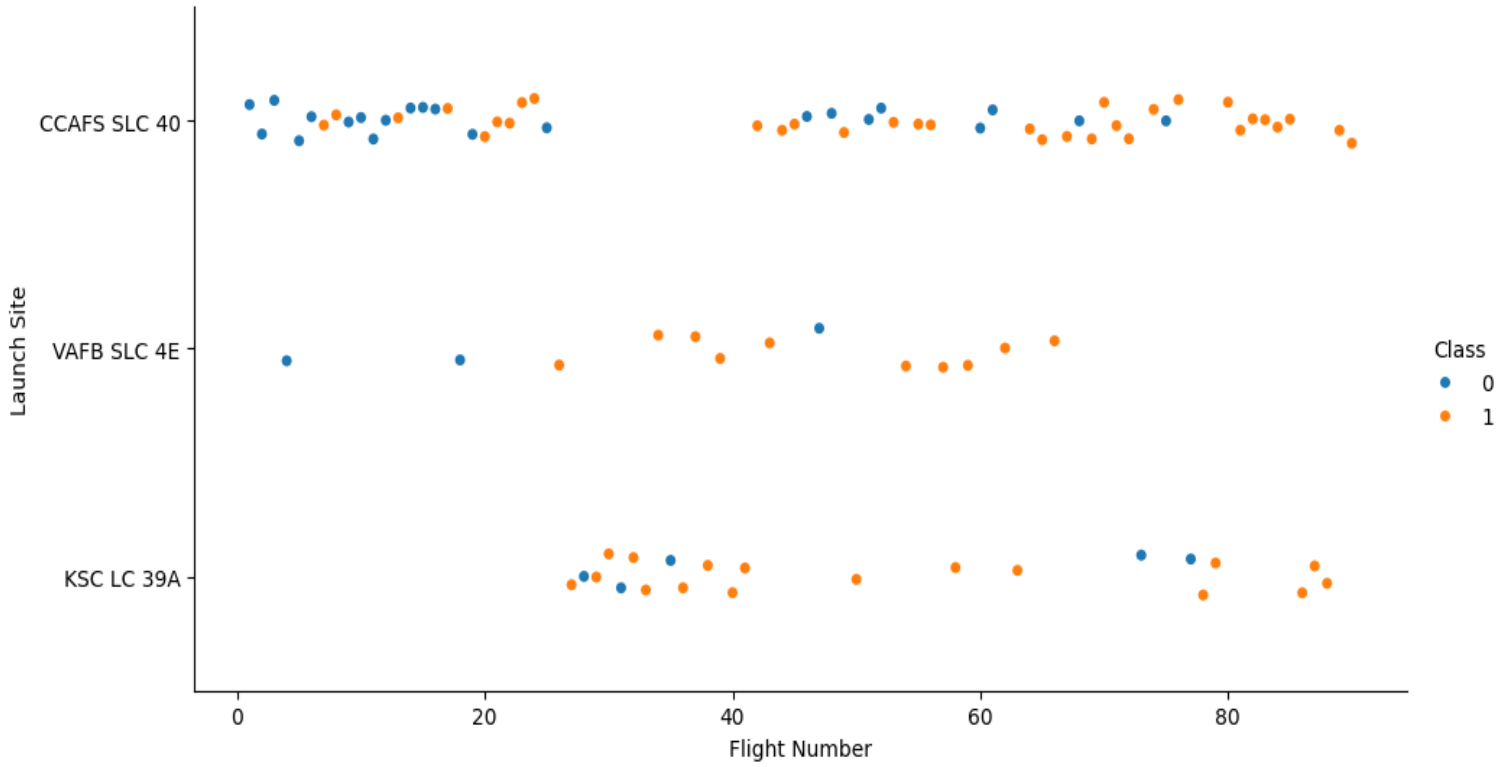
- **Best Model:** The Decision Tree model emerged as the most effective predictive model for the dataset.

Section 2

Insights drawn from EDA

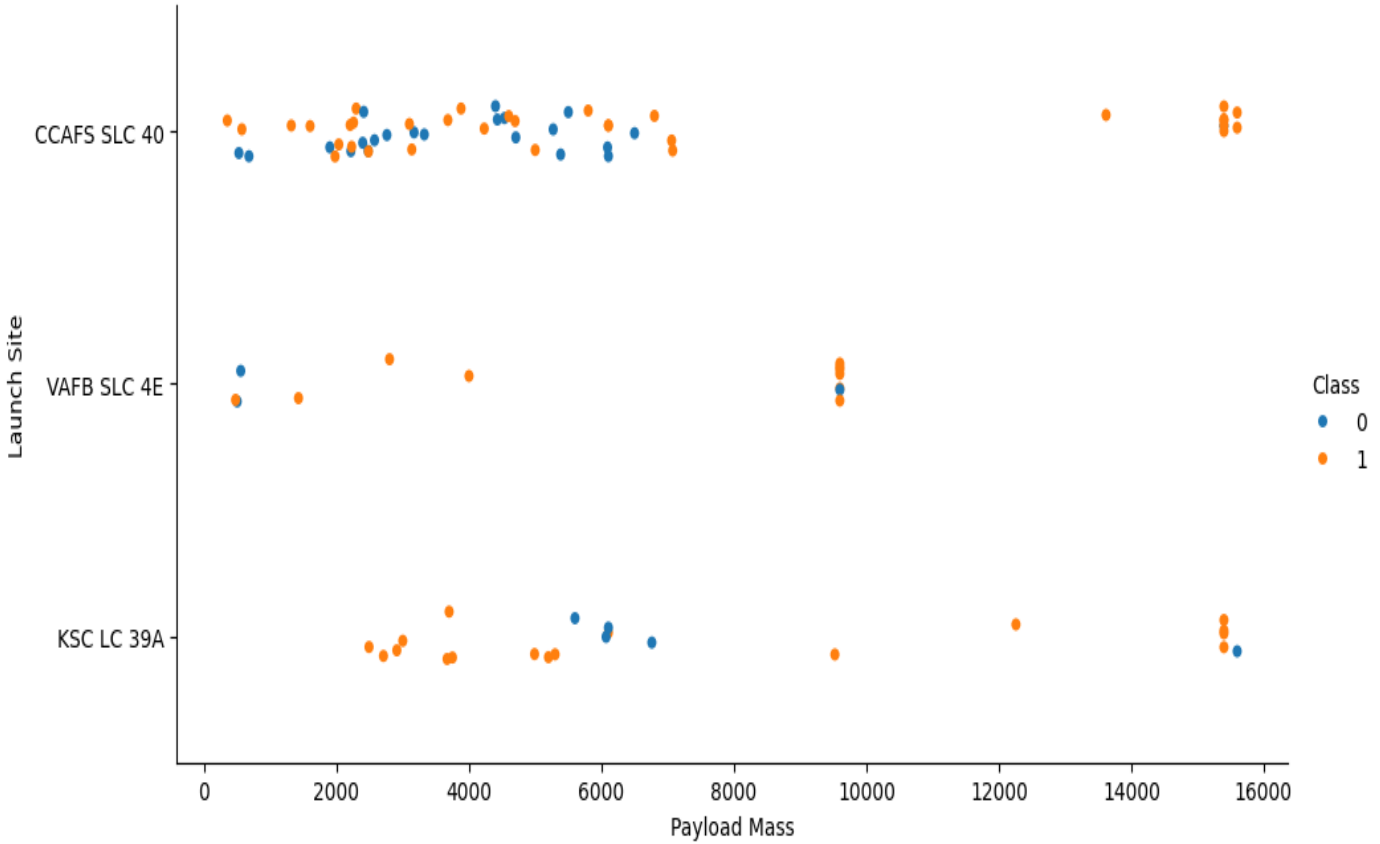
Flight Number vs. Launch Site

The scatter plot shows a clear trend: after around 40 flights, the success rate started to rise, indicating that newer launches are more successful. Also, among the three launch sites, KSC LC-39A shines with the highest success rate, highlighting its effectiveness and reliability for SpaceX.



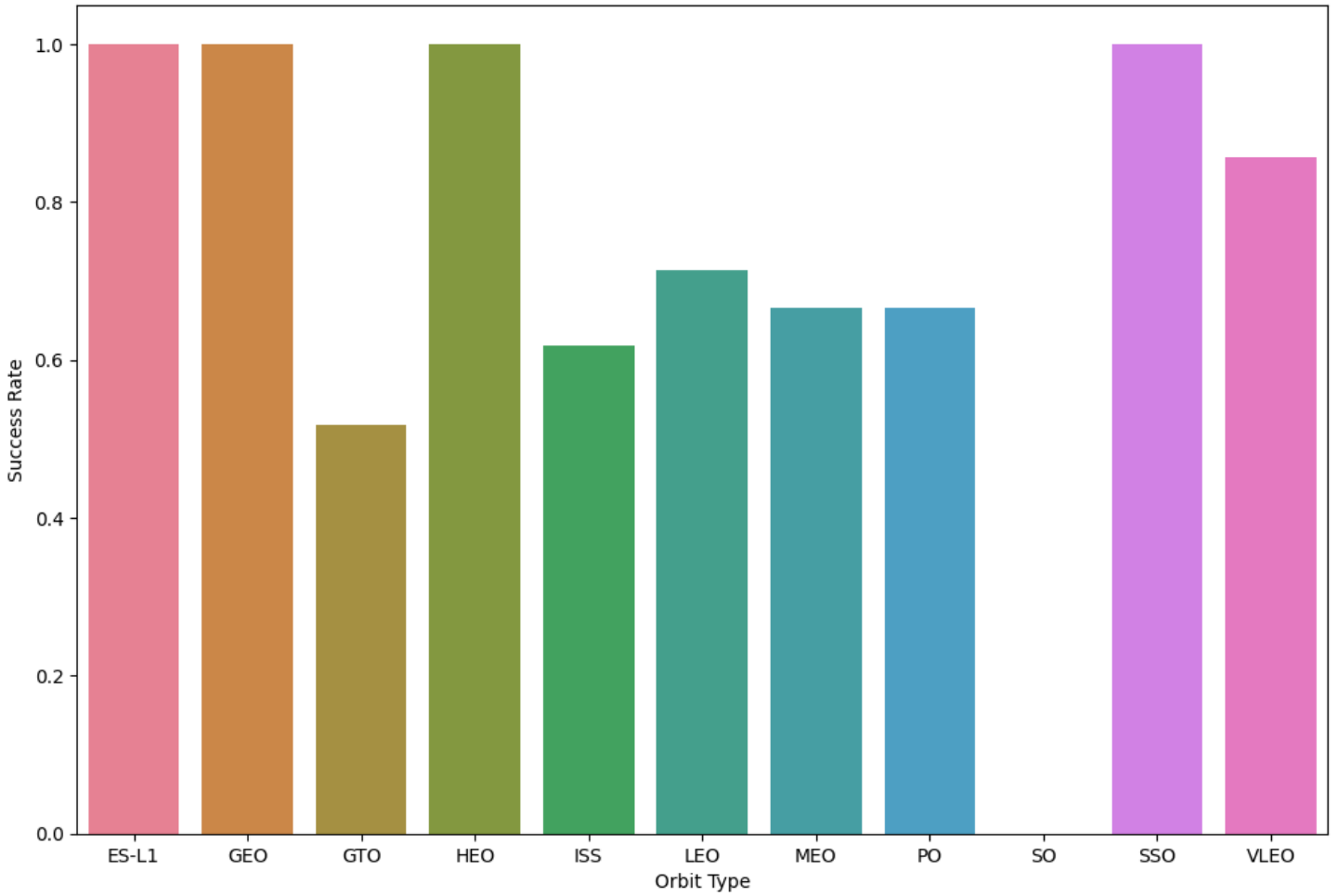
Payload vs. Launch Site

We observe from the scatter plot that CCAFS SLC 40 achieves a 100% success rate with payloads over 6,500 kg, suggesting that higher payloads are associated with higher success rates for this site. On the other hand, VAFB SLC 4E hasn't had any rocket launches with payloads exceeding 10,000 kg, so we can't assess its performance for higher payloads. Meanwhile, KSC LC-39A shows a higher success rate for payloads between 2,000 kg and 6,000 kg, indicating that this site performs better with lower payloads.



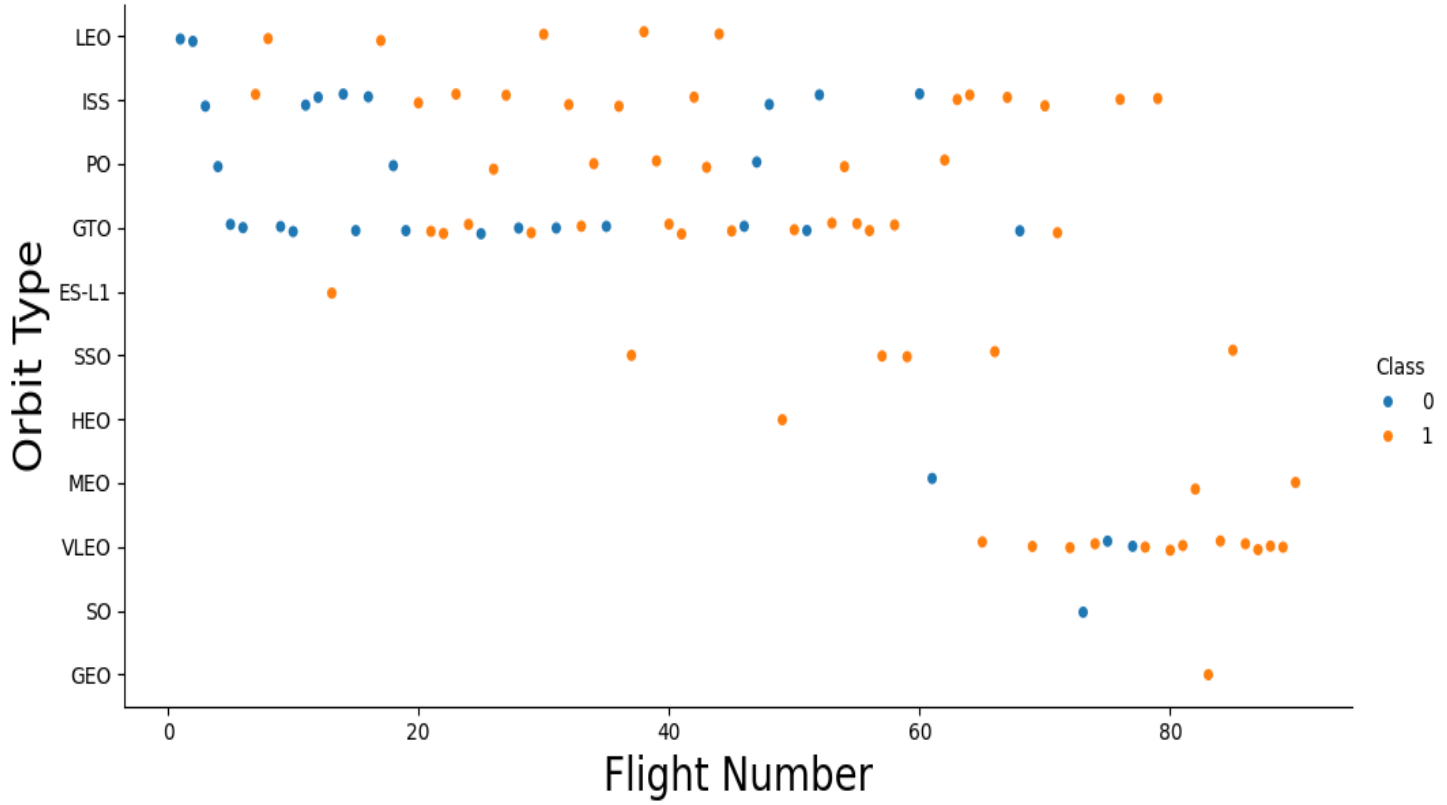
Success Rate vs. Orbit Type

The bar chart reveals that orbits ES-L1, GEO, HEO, and SSO all have a 100% success rate, highlighting their reliability. Additionally, it's noticeable that no rockets were launched to the SO orbit, so we don’t have data on its success rate



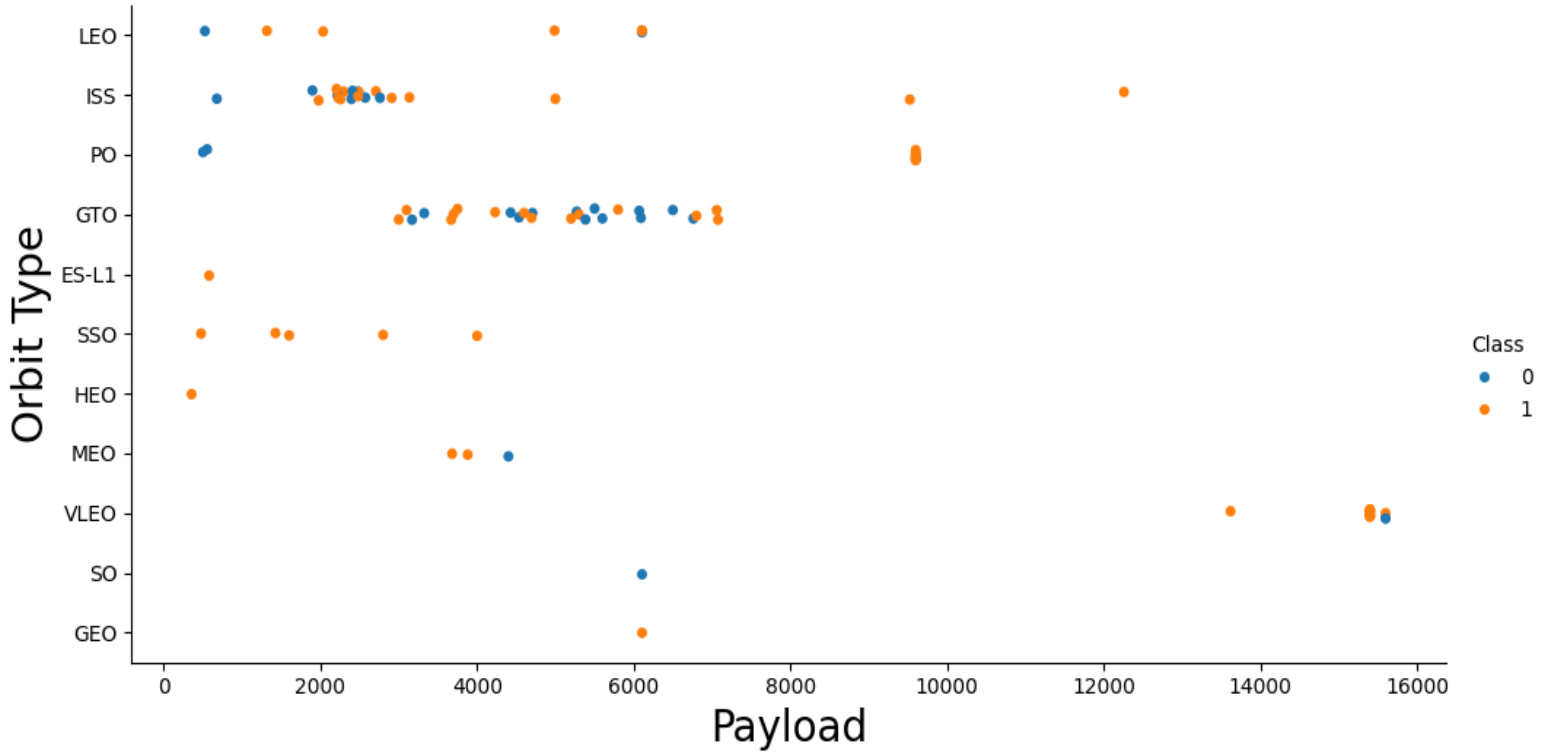
Flight Number vs. Orbit Type

We see that the success rate tends to increase with the number of flights. For instance, the VLEO orbit shows a 100% success rate after 80 flights. However, for the GTO orbit, the success rate doesn't seem to follow a clear trend, as it doesn't appear to relate to the number of flights



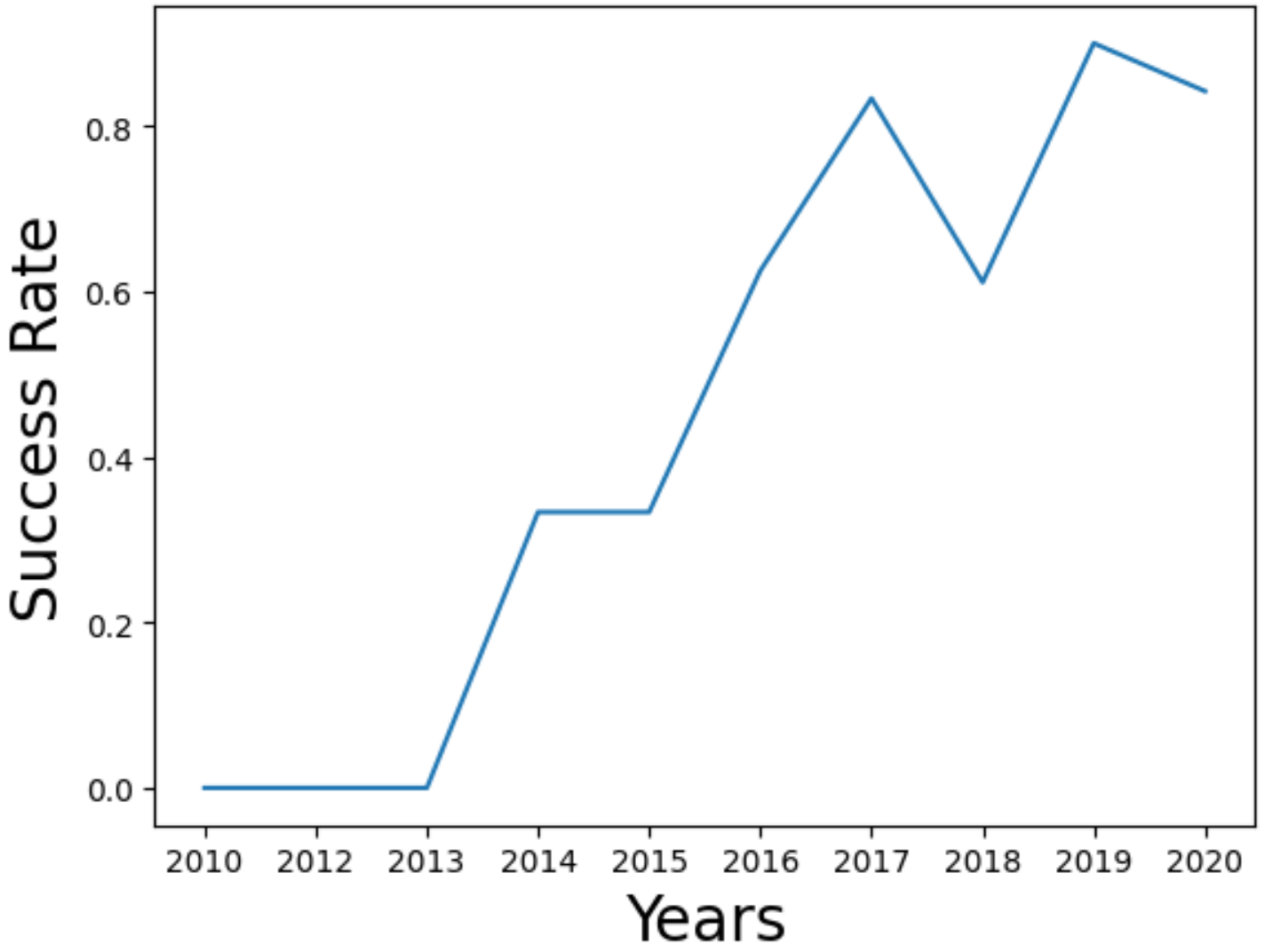
Payload vs. Orbit Type

We observe that the SSO orbit has a 100% success rate with low payloads, but we lack data for high payloads, so we can't determine its success rate in those cases. On the other hand, the ISS orbit shows a high success rate for payloads above 2,500 kg, indicating that it performs well with heavier payloads.



Launch Success Yearly Trend

- We observe that the success rate steadily increased from 2013 to 2017, with stability in 2014. However, after 2017, there was a slight decrease, followed by an increase again in 2018, showing a rebound in success.



All Launch Site Names

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

```
%sql select DISTINCT "Launch_Site" from SPACEXTABLE
```

Python

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

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

This query retrieves the unique names of the launch sites from the SpaceX data. The result shows that there are three distinct launch sites: CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE where Launch_Site LIKE 'CCA%' limit 5
```

✓ 0.0s Python

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

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

This query selects the first five records from the SPACEXTABLE where the Launch Site name begins with "CCA". The LIKE 'CCA%' pattern matches any launch site names that start with "CCA", and the LIMIT 5 clause restricts the output to just five rows.

Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)'
```

Python

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

sum(PAYLOAD_MASS__KG_)
45596

This query calculates the total payload mass (in kilograms) carried by boosters launched for NASA's Commercial Resupply Services (CRS) missions. The SUM(PAYLOAD_MASS__KG_) function adds up all the values in the PAYLOAD_MASS__KG_ column where the Customer is 'NASA (CRS)'

Average Payload Mass by F9 v1.1

This query finds the average payload mass for all launches that used the F9 v1.1 booster

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

```
%sql select avg(PAYLOAD_MASS__KG_) as 'Average Payload' from SPACEXTBL where Booster_Version='F9 v1.1'
```

✓ 0.0s Python

* sqlite:///my_data1.db
Done.

Average Payload
2928.4

First Successful Ground Landing Date

This query finds the earliest date when a landing was successfully completed on a ground pad.

```
%sql select min(Date) from SPACEXTBL where Landing_Outcome='Success (ground pad)'
```

[12] ✓ 0.0s Python

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

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql select Booster_Version from SPACEXTBL where Landing_Outcome='Success (drone ship)' and
PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000
```

Python

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

Booster_Version
F9 FT B1029.1
F9 FT B1036.1
F9 B4 B1041.1

This query retrieves the booster versions for launches that landed successfully on a drone ship, with a payload mass between 4,000 kg and 6,000 kg.

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, COUNT(*) from SPACEXTBL GROUP BY Mission_Outcome
```

[14] ✓ 0.0s Python

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

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

This query counts the number of launches for each mission outcome and groups the results by the Mission Outcome column. This will show the total count of each type of mission outcome, such as 'Success' or 'Failure'

Boosters Carried Maximum Payload

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_= (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

[51] Python

```
 * 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

This query retrieves the booster versions used for launches with the maximum payload mass recorded in the dataset. The subquery finds the highest payload mass, and the main query selects the booster versions corresponding to this maximum payload

2015 Launch Records

```
%%sql select substr(Date,6,2) as month,Booster_Version, Launch_Site,Landing_Outcome from SPACEXTBL
where substr(Date,0,5)='2015' and Landing_Outcome='Failure (drone ship)'
```

[15] ✓ 0.0s Python

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

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

This query selects the month, booster version, launch site, and landing outcome for all records from the year 2015 where the landing outcome was a failure on a drone ship. The SUBSTR function extracts the month from the Date column and filters the results accordingly.

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

```
%%sql select Landing_Outcome,count(*) as count from SPACEXTBL
where Date > '2010-06-04' and Date <'2017-03-20'
group by Landing_Outcome order by count desc
```

[18] ✓ 0.0s Python

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

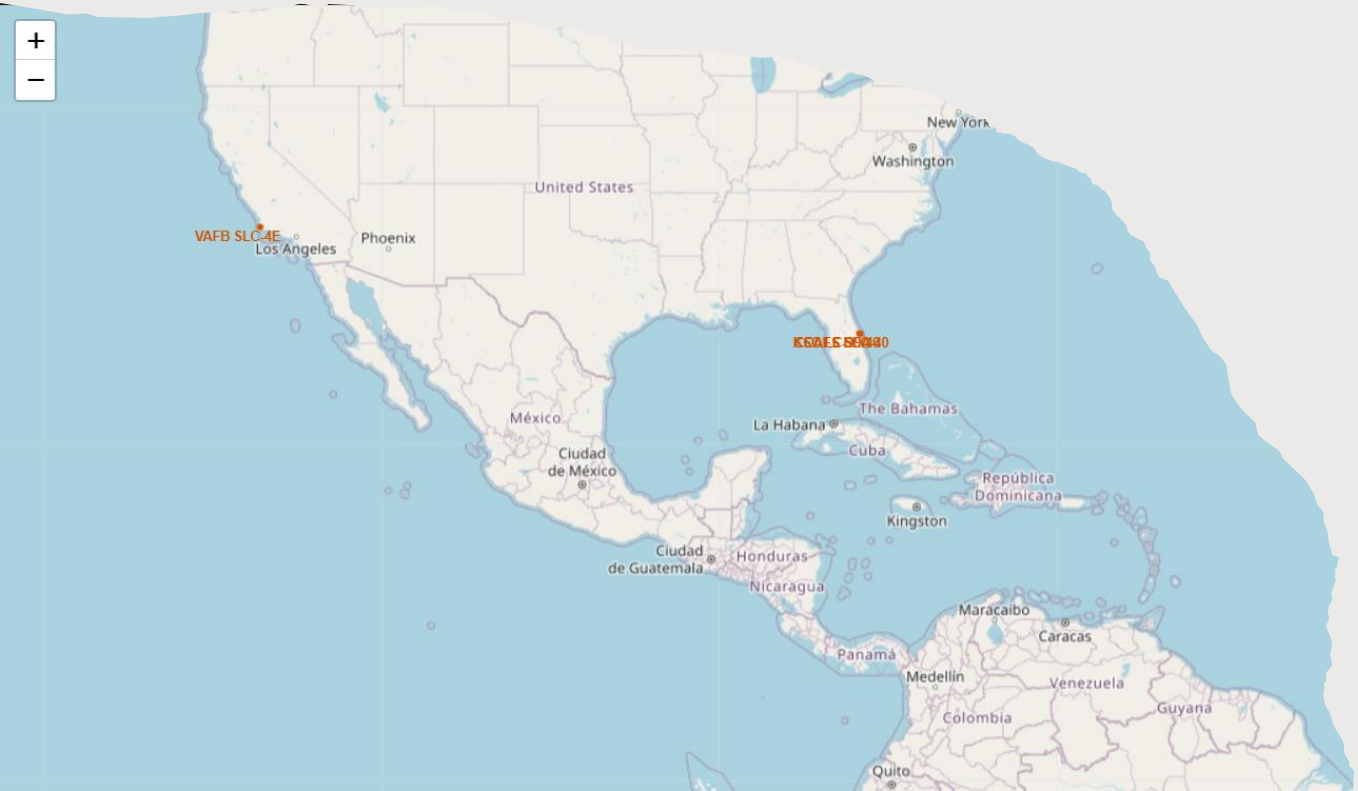
Landing_Outcome	count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

This query counts the number of launches for each landing outcome within the date range from June 4, 2010, to March 20, 2017. The results are grouped by Landing Outcome and ordered by the count in descending order, showing the most frequent outcomes first

Section 3

Launch Sites Proximities Analysis

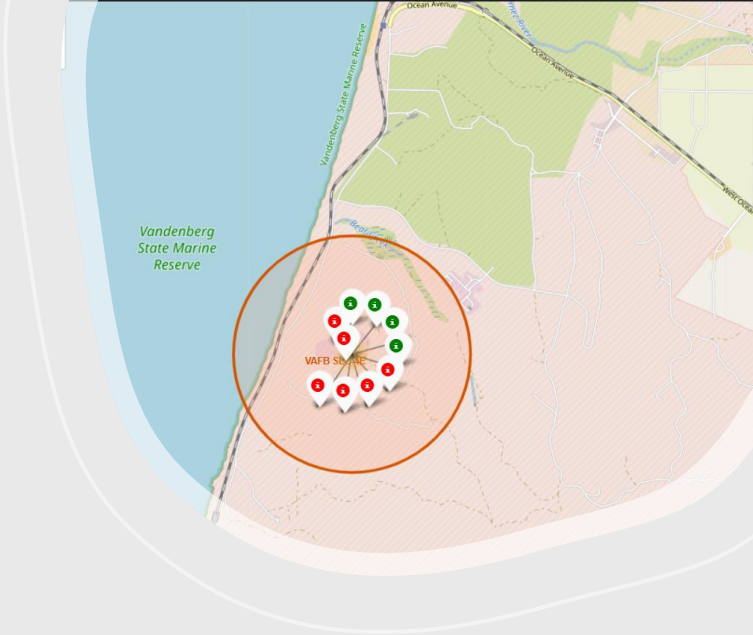
SpaceX Launch Sites

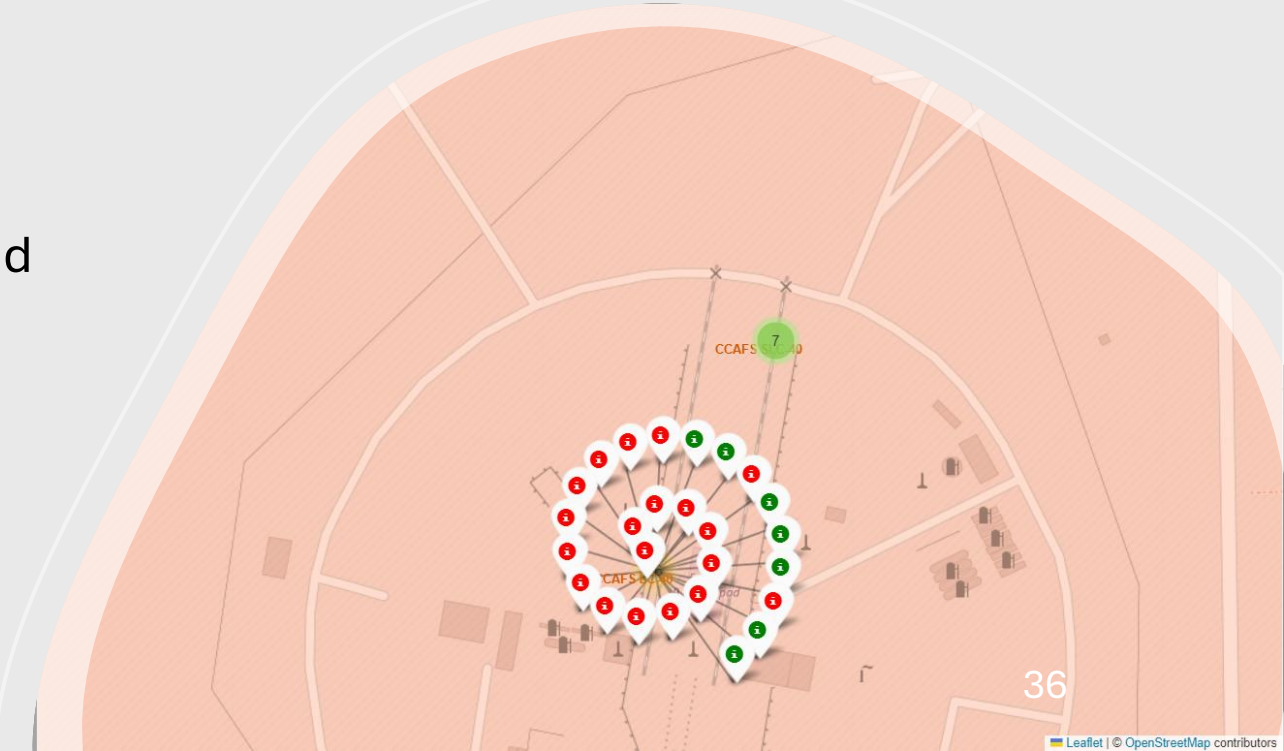


- This map highlights SpaceX's key launch sites:
- VAFB SLC-4E
- CCAFS SLC-40
- KSC LC-39A
- Launch sites are located near the coast for safety.

SpaceX Launch Outcomes

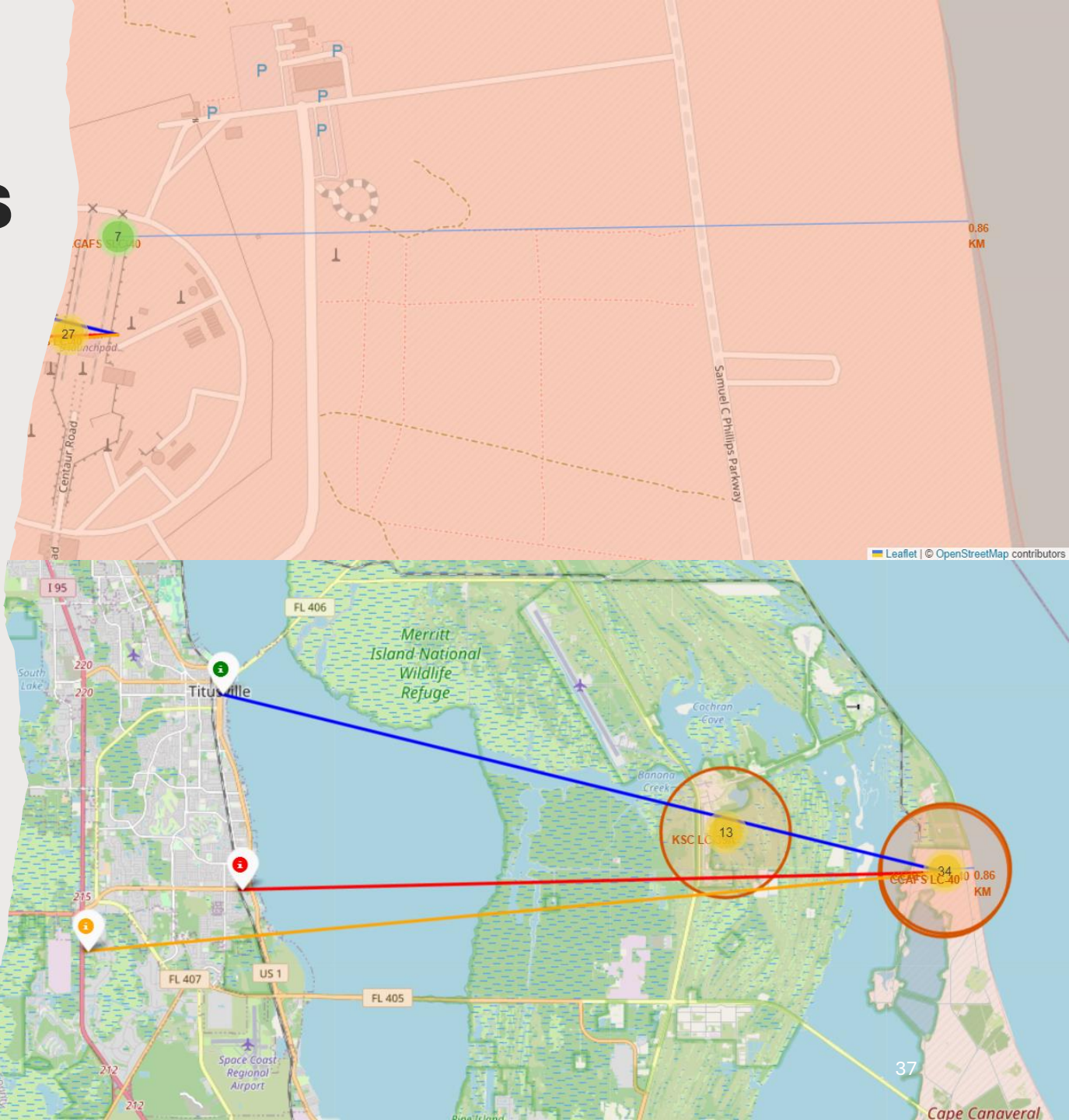
- This map visualizes the outcomes of SpaceX launches, color-coded by success and failure:
- Green markers represent successful landings.
- Red markers represent failed landings.
- The success rate is high at most sites, especially at KSC LC-39A.
- Analyzing the distribution helps understand the reliability of different launch sites.

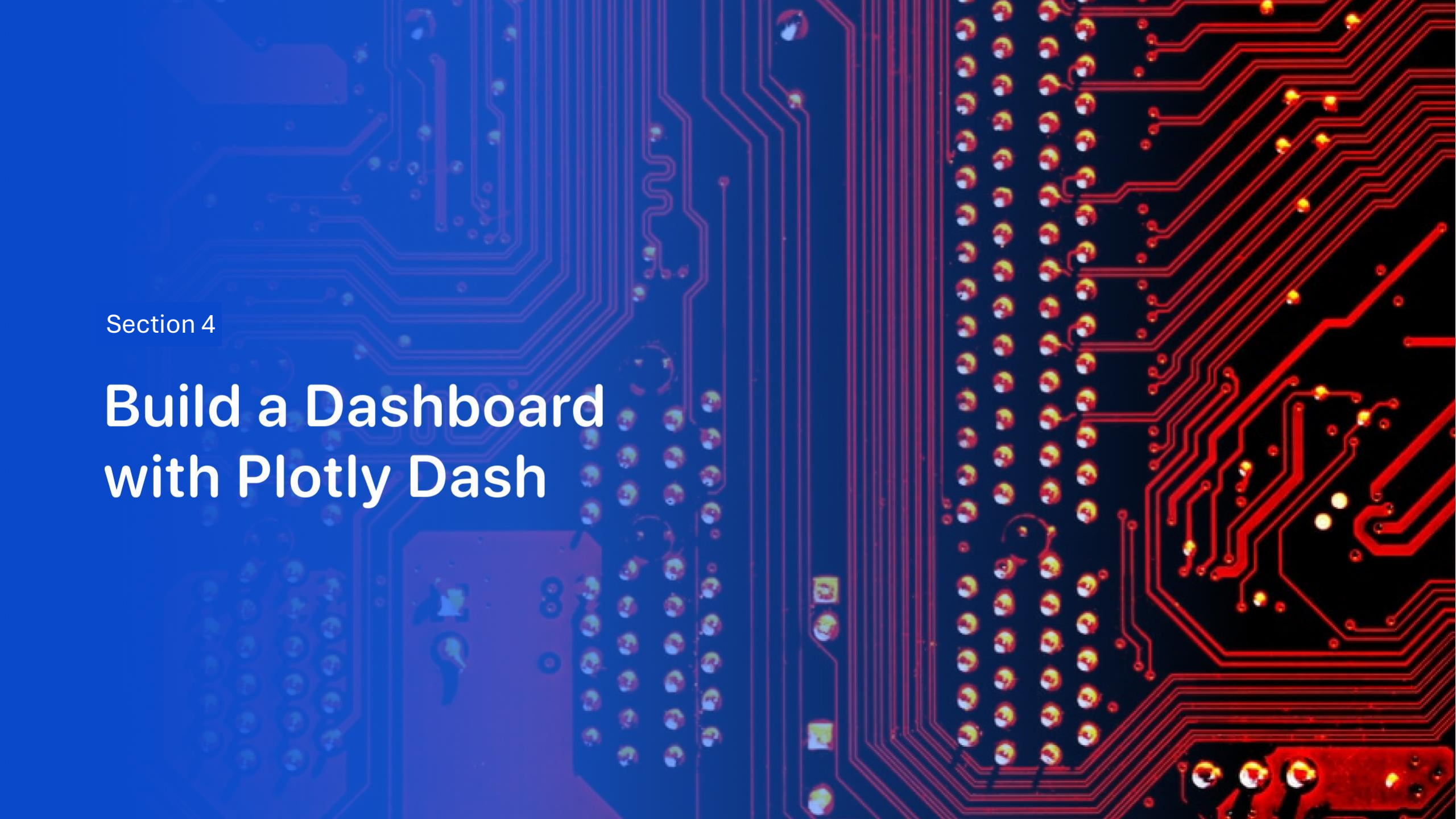




Distance to Proximities

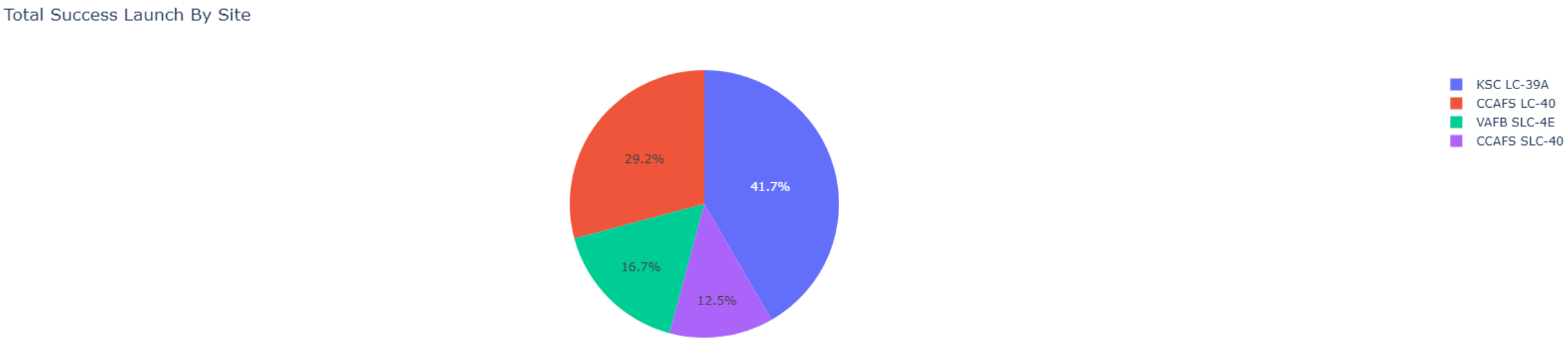
- The following map shows one selected SpaceX launch site and its proximity to key infrastructure:
- Coastline, Railway, Highway: The overground distance from the launch site, calculated to the nearest coastline, railway, and highway is shown.
- These distances are marked on the map, showing that the launch site had a great strategic position.
- The launch site had an appropriate location for the purposes of safety and efficiency during operations, standing far enough from any populated area and close to routes of transport.




Section 4
Build a Dashboard with Plotly Dash

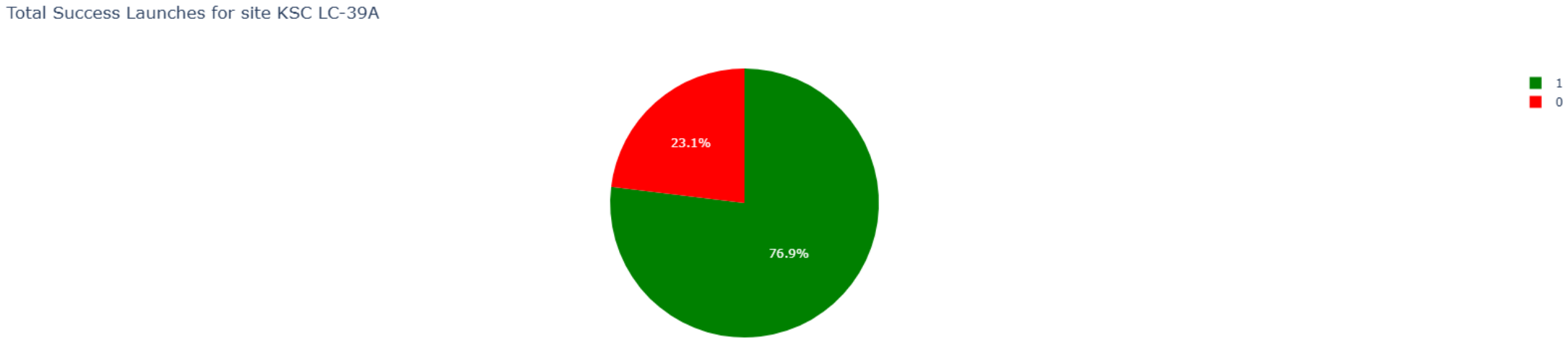
Launch Success Rates by Site

- **KSC LC-39A**: Leads with 41.7% of successful launches.
- **CCAFS LC-40**: Follows at 29.2%.
- **VAFB SLC-4E**: Contributes 16.7%.
- **CCAFS SLC-40**: Accounts for 12.5%.
- KSC LC-39A has the highest success rate among all launch sites.



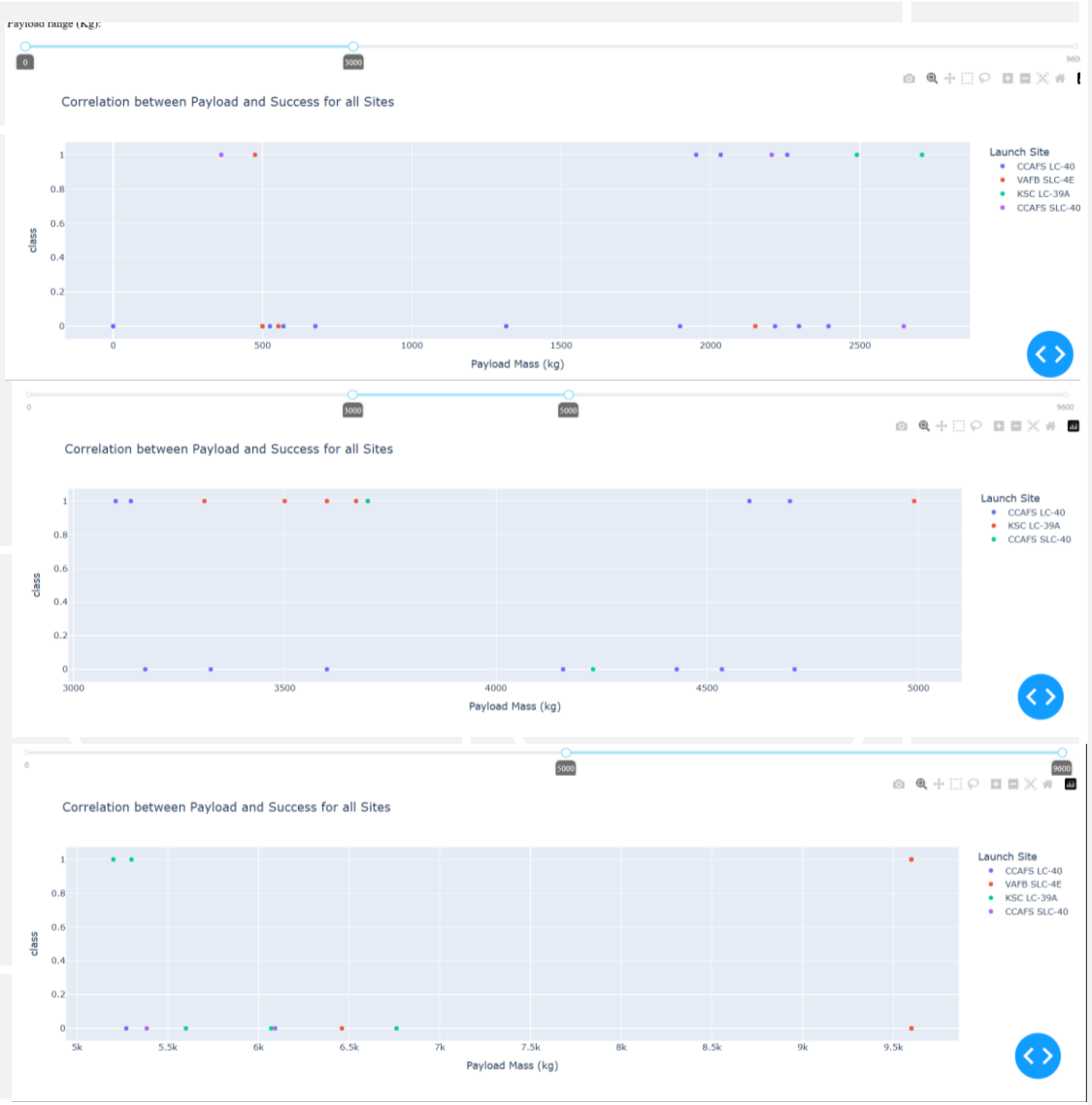
Launch Success Ratio for KSC LC-39A

- **High Success Rate:** KSC LC-39A has a strong success rate of 76.9%, indicating a high level of reliability.
- **Lower Failure Rate:** The failure rate is 23.1%, showing that while there have been some failures, they are less frequent.



Payload vs. Launch Outcome Analysis (0 to 10,000 kg)

- **Low Payloads (0 to 3,000 kg):**
- **Failure Rate:** Slightly higher.
- **Observation:** Failure rates are a bit higher, but still relatively low.
- **Medium Payloads (3,001 to 5,000 kg):**
- **Success Rate:** Higher.
- **Details:** 10 successful launches vs. 8 failures. This range has the best success rate.
- **Heavy Payloads (5,001 to 10,000 kg):**
- **Failure Rate:** Much higher.
- **Details:** 3 successful launches vs. 8 failures. Heavier payloads have a significantly higher failure rate.

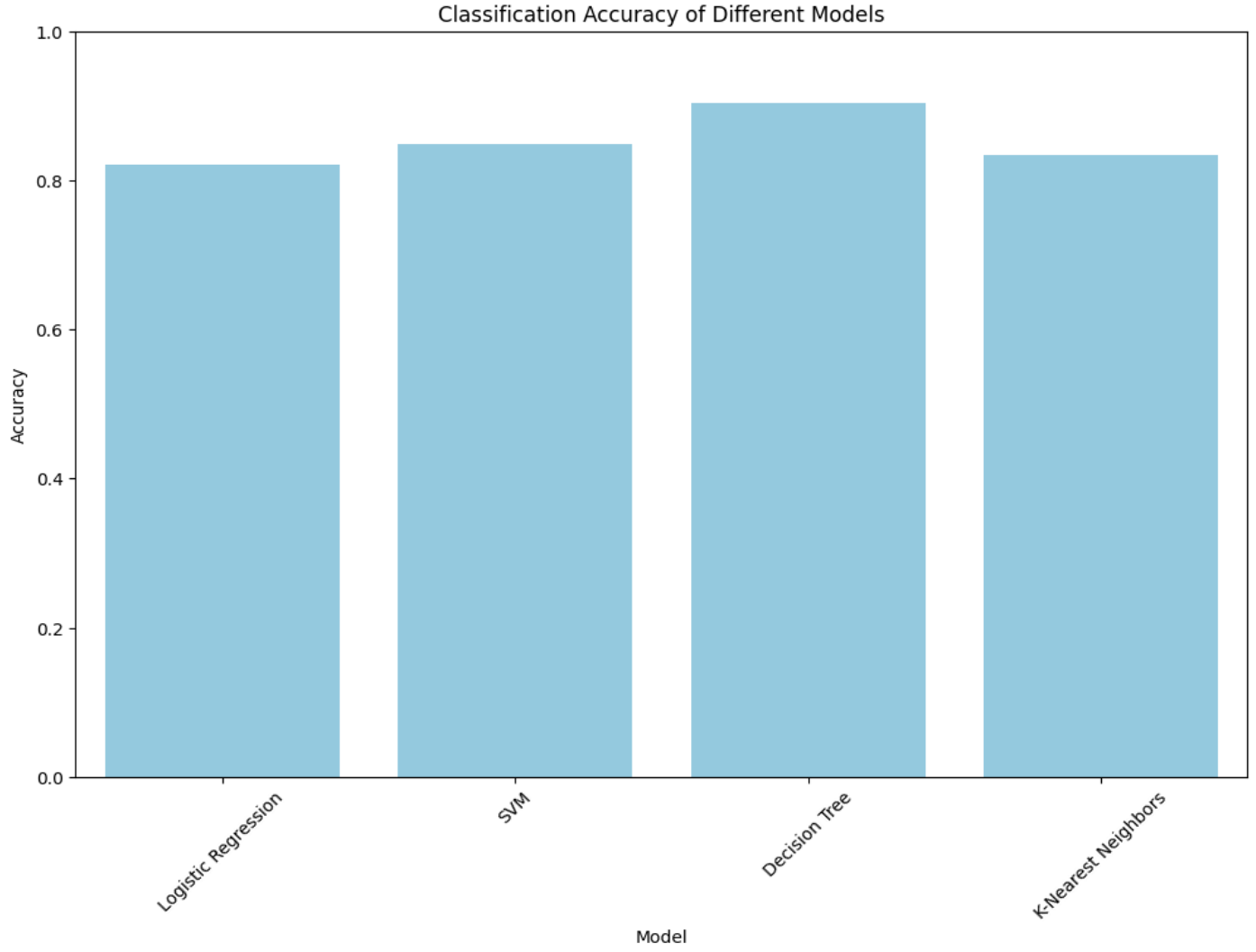


Section 5

Predictive Analysis (Classification)

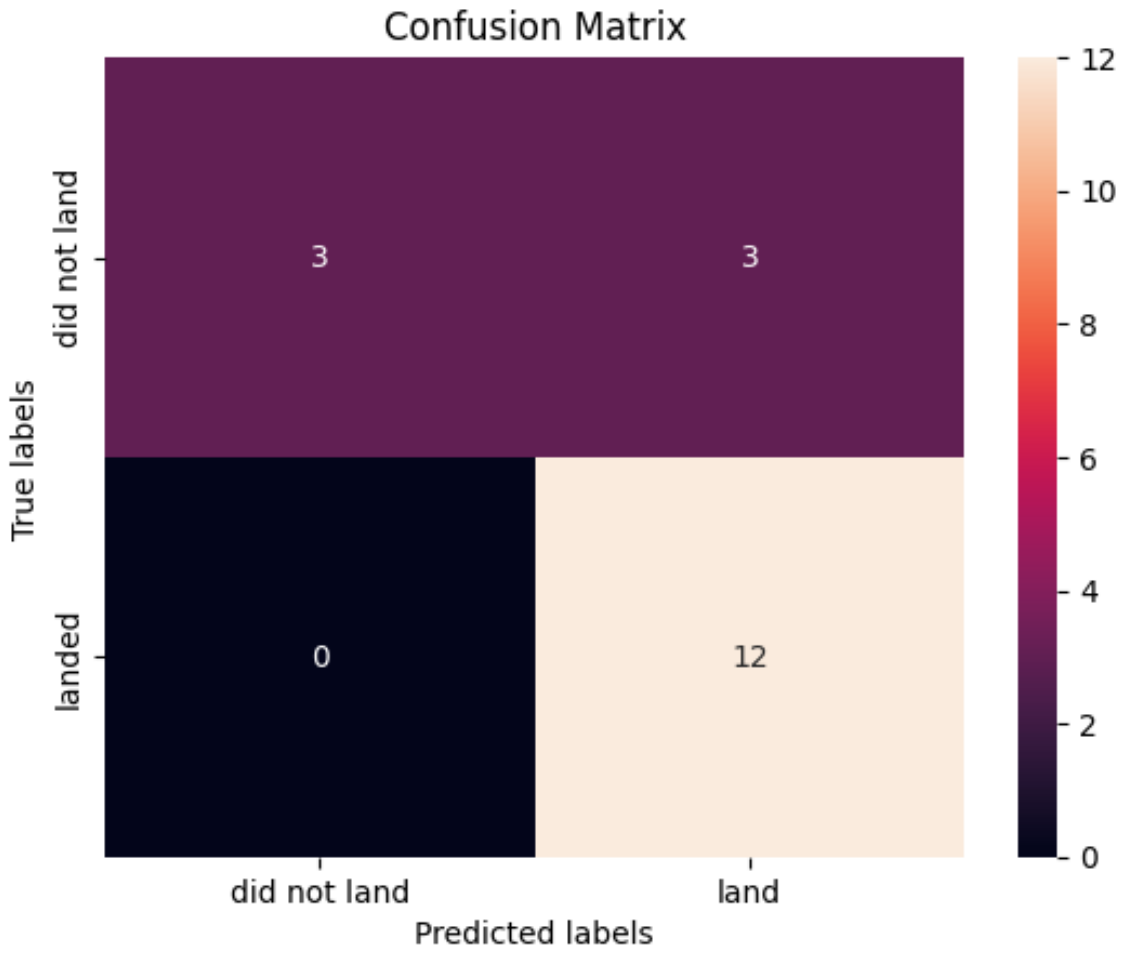
Classification Accuracy

- Logistic Regression, SVM, Decision Tree, and K-Nearest Neighbors have similar accuracy.
- The Decision Tree model has slightly higher accuracy compared to the others.



Confusion Matrix

- **True Positives (TP):** 12 (Correctly predicted positive cases)
- **False Positives (FP):** 3 (Incorrectly predicted as positive)
- **True Negatives (TN):** 3 (Correctly predicted negative cases)
- **False Negatives (FN):** 0 (No missed positive cases)
- **High Accuracy:** The model correctly identified all actual positives with no false negatives.
- **Overall Performance:** Excellent, with a few false positives but no false negatives.



Conclusion

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!