IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sofiat Ajide
December 20th, 2024

Outline

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

Executive Summary

Summary of Methodologies: This project aimed to predict the success of Falcon 9 first-stage landings using a combination of data collection, exploratory data analysis (EDA), and machine learning techniques. Data was collected through SpaceX's API and web scraping from Wikipedia. We processed and cleaned the data using Python, including handling missing values and transforming it into a format suitable for analysis. We then applied SQL queries to extract insights from a PostgreSQL database. Finally, we built and evaluated various machine learning models, using a Decision Tree classifier, and fine-tuned them through hyperparameter optimization.

Summary of Results:

- Launch sites with higher flight volumes generally exhibited higher success rates.
- The success rate of launches increased from 2013 to 2020.
- Orbits such as ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.
- The KSC LC-39A site achieved the highest number of successful launches.
- The Decision Tree classifier emerged as the most effective model, outperforming others in predicting landing success.

Introduction

Project background and context

- SpaceX's Falcon 9 rocket reduces launch costs significantly through reusable first-stage technology.
- Falcon 9 launches cost approximately **$62 million**, compared to competitors charging upwards of **$165 million**.
- Reusability is key to cost savings, achieved by the successful landing of the first stage.
- Predicting the success of these landings can provide insights for cost estimation and competitive bidding.

Problems you want to find answers

1. What factors influence the successful landing of the Falcon 9 first stage?
2. Can we create a reliable predictive model for successful landings?
3. How can this information help in comparing costs between SpaceX and its competitors?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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 API

- Used **GET requests** to retrieve data from the SpaceX API.
- Response content was decoded into JSON format using .json() and converted to a **pandas dataframe** with .json_normalize().

2.Data Cleaning

- Checked for **missing values** and filled in gaps where necessary.
- Ensured the data was formatted and consistent for analysis.

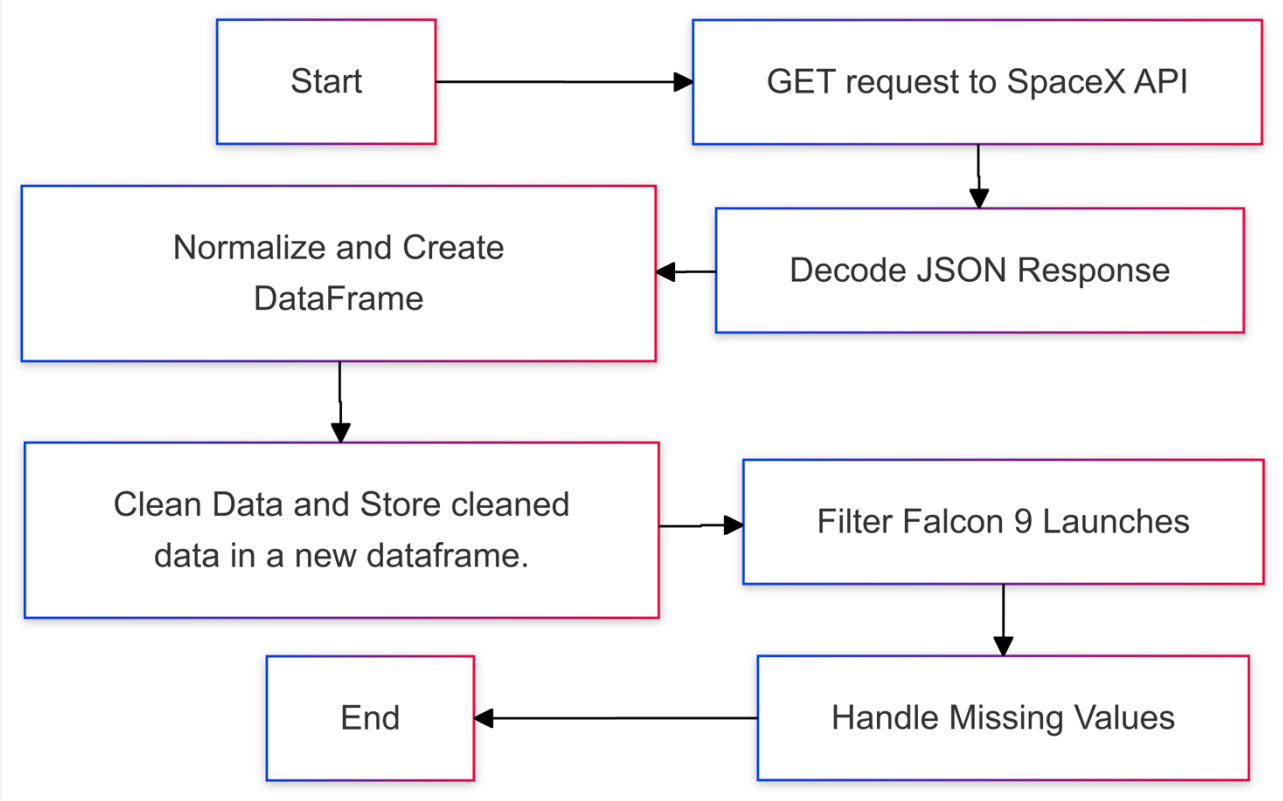
3.Web Scraping with BeautifulSoup

- Extracted Falcon 9 launch records from **Wikipedia**.
- Parsed HTML tables using **BeautifulSoup** and converted them to a pandas dataframe for future use.

Data Collection – SpaceX API

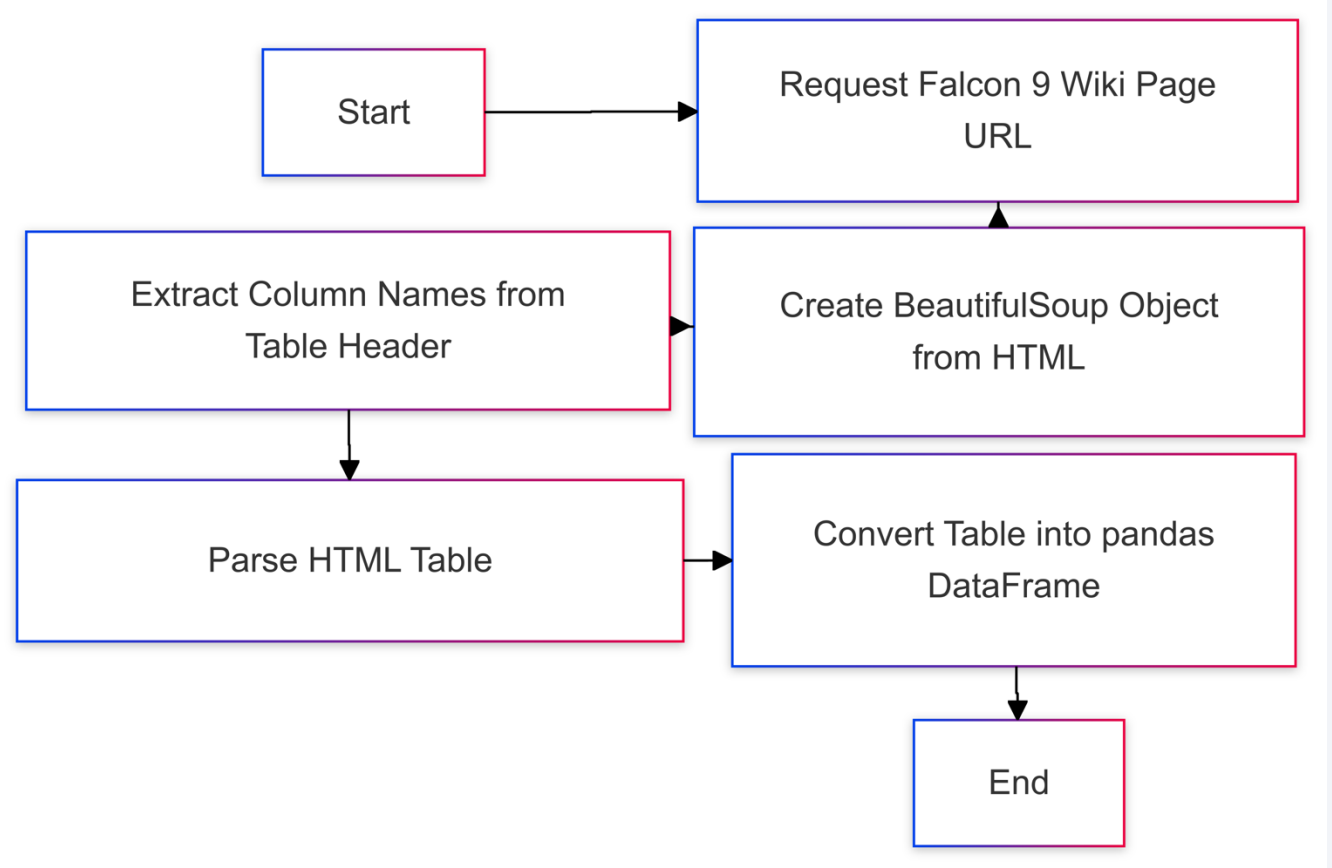
- We used a GET request to retrieve data from the SpaceX API. The retrieved data was cleaned, wrangled, and formatted, then stored in a pandas dataframe for further analysis.

- GitHub URL to the completed SpaceX API notebook: https://github.com/sofiatajide/datasciencecoursera/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- We utilized web scraping with BeautifulSoup to extract Falcon 9 launch records. The extracted table was parsed and converted into a pandas dataframe for analysis.
- GitHub URL of the completed web scraping notebook: https://github.com/sofiatajide/datasciencecoursera/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



Data Wrangling

- We calculated the number of launches at each site and analyzed the frequency of each orbit type.
- We created the landing outcome label based on the "outcome" column and exported the results to a CSV file.
- GitHub URL of the completed web scraping notebook: https://github.com/sofiatajide/datasciencecoursera/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

We explored the data by visualizing various relationships, including:

- The correlation between flight number and launch site: To visualize if there is any pattern or correlation between the number of flights and different launch sites.
- The relationship between payload and launch site: To examine how payload sizes vary across different launch sites.
- The success rate of each orbit type: To visualize the success rate for each orbit type.
- The connection between flight number and orbit type: To explore the relationship between flight number and orbit type.
- The yearly trend of launch success rates: To track the launch success rate over time.
- GitHub URL of EDA with visualization: https://github.com/sofiatajide/datasciencecoursera/blob/main/edadataviz.ipynb

EDA with SQL

We performed exploratory data analysis (EDA) using SQL to gain insights from the data. Some of the queries we executed include:

- Identifying the unique launch sites in the space mission.
- Calculating the total payload mass carried by boosters launched by NASA (CRS).
- Finding the average payload mass carried by booster version F9 V1.1.
- Counting the total number of successful and failed mission outcomes.
- Analyzing the failed landing outcomes on the drone ship, along with their corresponding booster version and launch site names.

- GitHub URL of EDA with SQL: https://github.com/sofiatajide/datasciencecoursera/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Markers**: We placed markers on the map to represent the exact locations of each launch site.
  Purpose: To visually identify and interact with each site for detailed analysis.
- **Circles**: Color-coded circles were used to indicate launch outcomes: red for failures and green for successes.
  Purpose: To provide a clear visual distinction between successful and failed launches at each site.
- **Lines**: Lines were added to show relationships between launch sites and other geographical features.
  Purpose: To visualize the proximity of launch sites to railways, highways, coastlines, and cities.
- **Outcome Labeling**: We assigned binary labels to launch outcomes: **0** for failure and **1** for success.
  Purpose: To easily identify and categorize the launch performance.
- **Proximity Analysis**: We calculated the distances from launch sites to nearby geographical features such as railways, highways, and coastlines.
  Purpose: To address key questions about the geographic placement of launch sites.
- GitHub URL interactive map with Folium map:
  https://github.com/sofiatajide/datasciencecoursera/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Pie Charts**:
  We plotted pie charts to show the total number of launches by each site.
  Purpose: To give users a quick, visual breakdown of the launch frequency per site and help identify which sites are more active.
- **Scatter Plot**:
  We created a scatter plot to display the relationship between launch outcome and payload mass (in kg) for different booster versions.
  Purpose: To analyze how payload mass influences the launch outcome and to identify trends across different booster versions.

Interactive Features

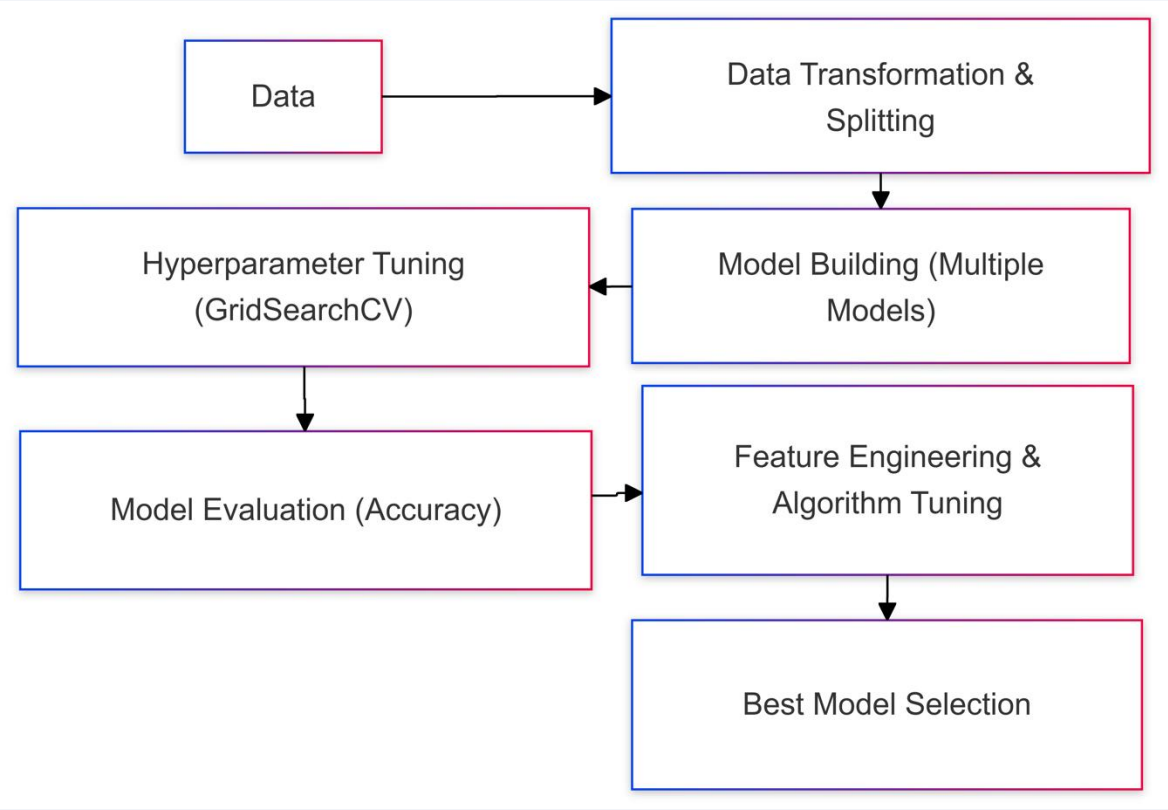
- **Interactivity**:
  Users can hover over the charts for detailed information, filter data by site or booster version, and explore the data dynamically. **Purpose**: To allow users to interact with the data, customize views, and gain deeper insights into specific aspects of the launches.
- GitHub URL of Plotly Dash lab:https://github.com/sofiatajide/datasciencecoursera/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- **Data Loading & Preprocessing**:
  - Loaded data using **NumPy** and **Pandas**.
  - Transformed the data into a suitable format for model training.
  - Split the dataset into **training** and **testing** sets.
- **Model Building**:
  - Built multiple machine learning models (e.g., Logistic Regression, Random Forest, etc.).
  - Tuned hyperparameters using **GridSearchCV** to optimize model performance.
- **Evaluation & Metric**:
  - Used **accuracy** as the evaluation metric to assess the model's performance.
- **Model Improvement**:
  - Applied **feature engineering** to enhance the dataset and improve model accuracy.
  - Performed **algorithm tuning** to fine-tune model parameters.
- **Best Model Selection**:
  - Identified the best performing classification model based on accuracy and other performance metrics.
- GitHub URL of predictive analysis lab: https://github.com/sofiatajide/datasciencecoursera/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)

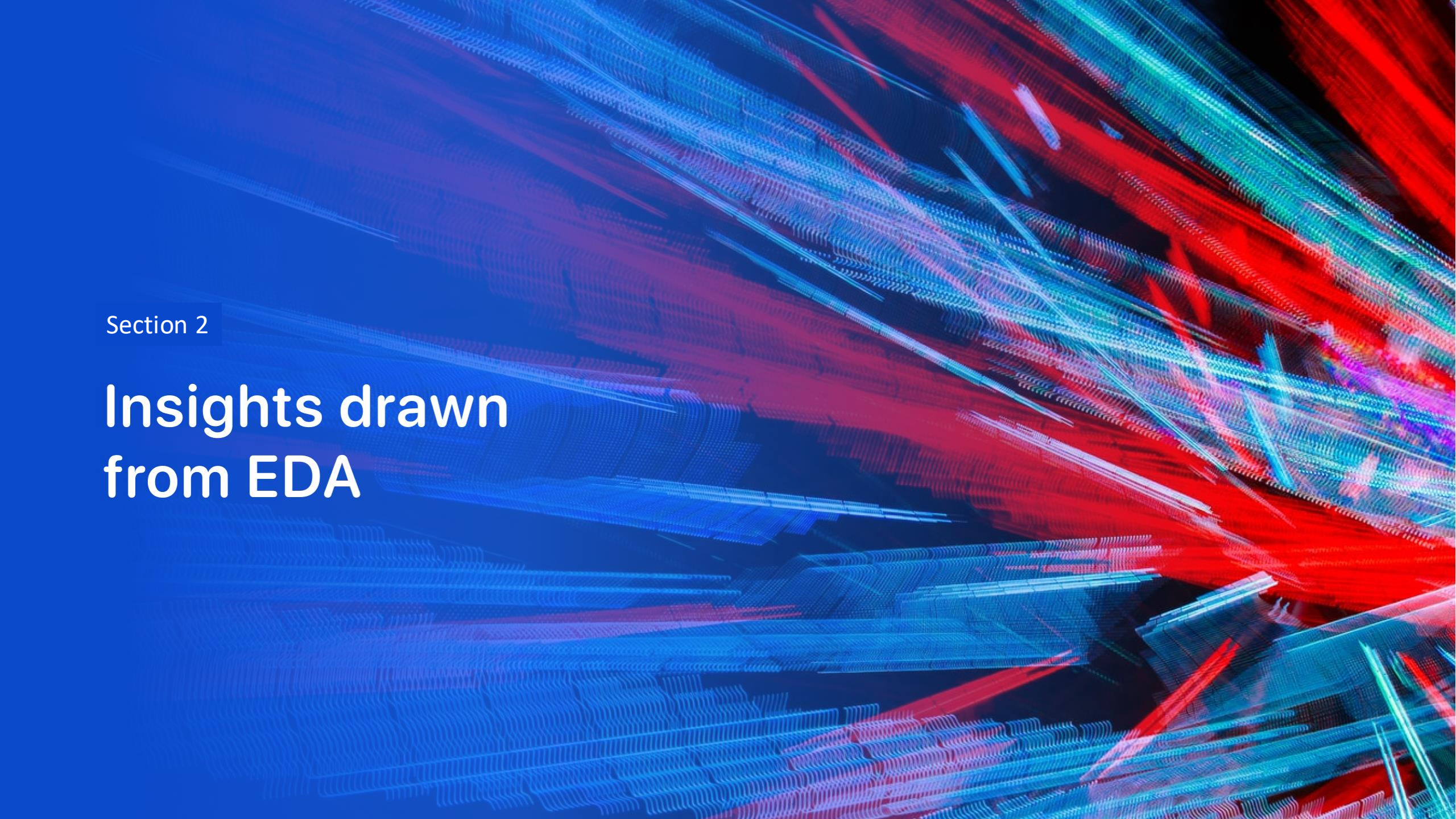
- Flow Chart



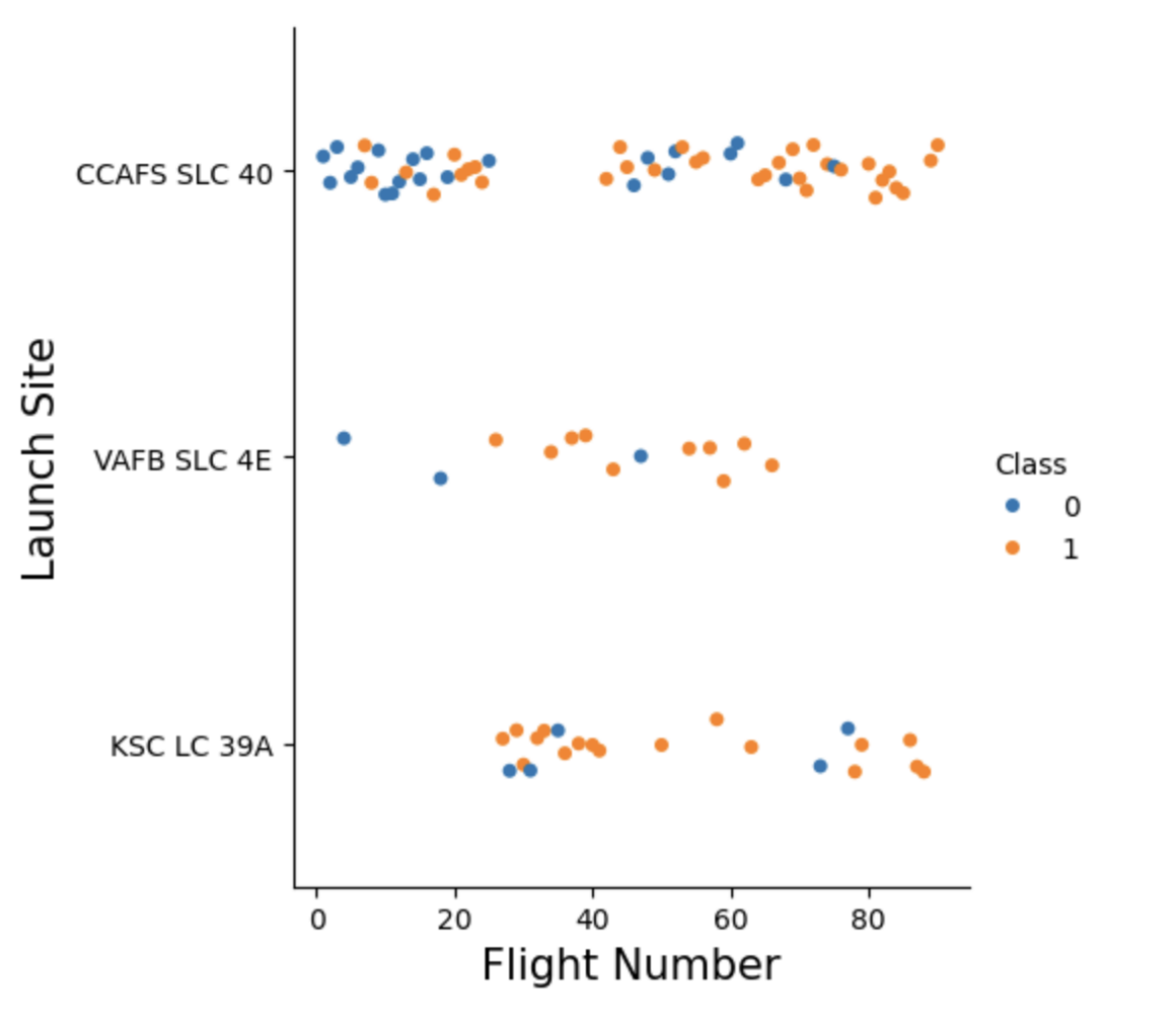
- GitHub URL of predictive analysis lab: https://github.com/sofiatajide/datasciencecoursera/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

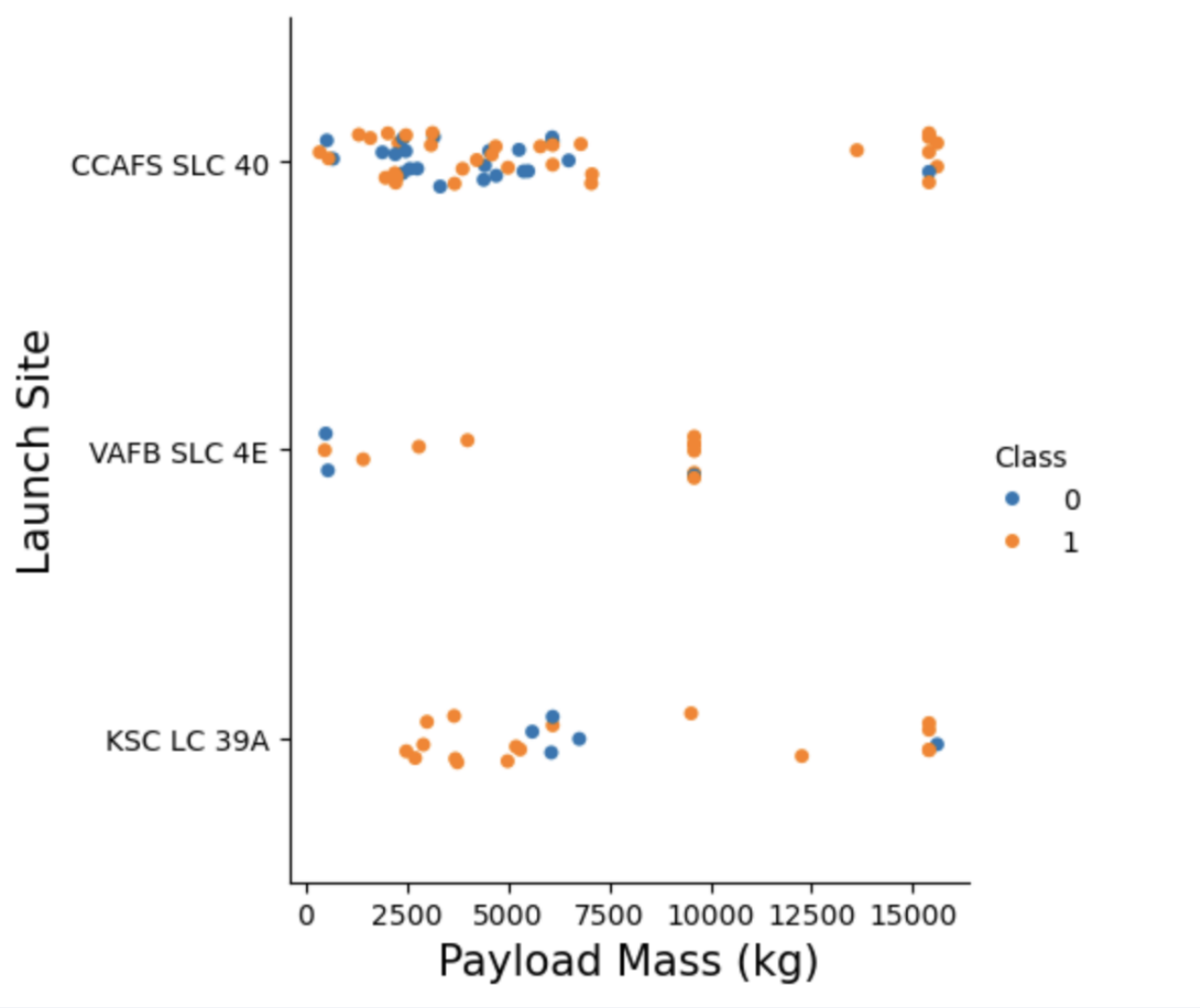

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



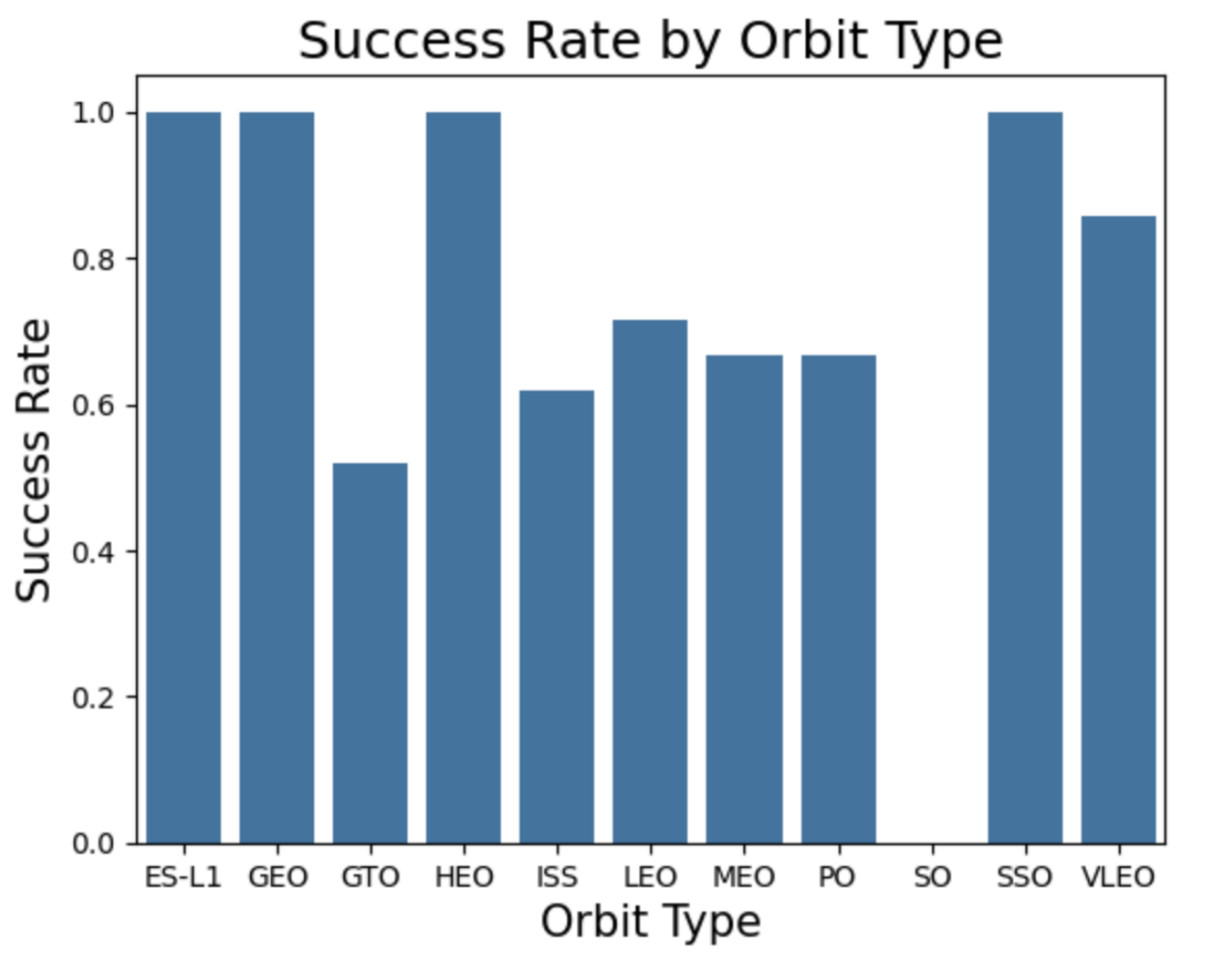
The plot reveals that a higher number of flights at a launch site is associated with a greater success rate at that site.

Payload vs. Launch Site



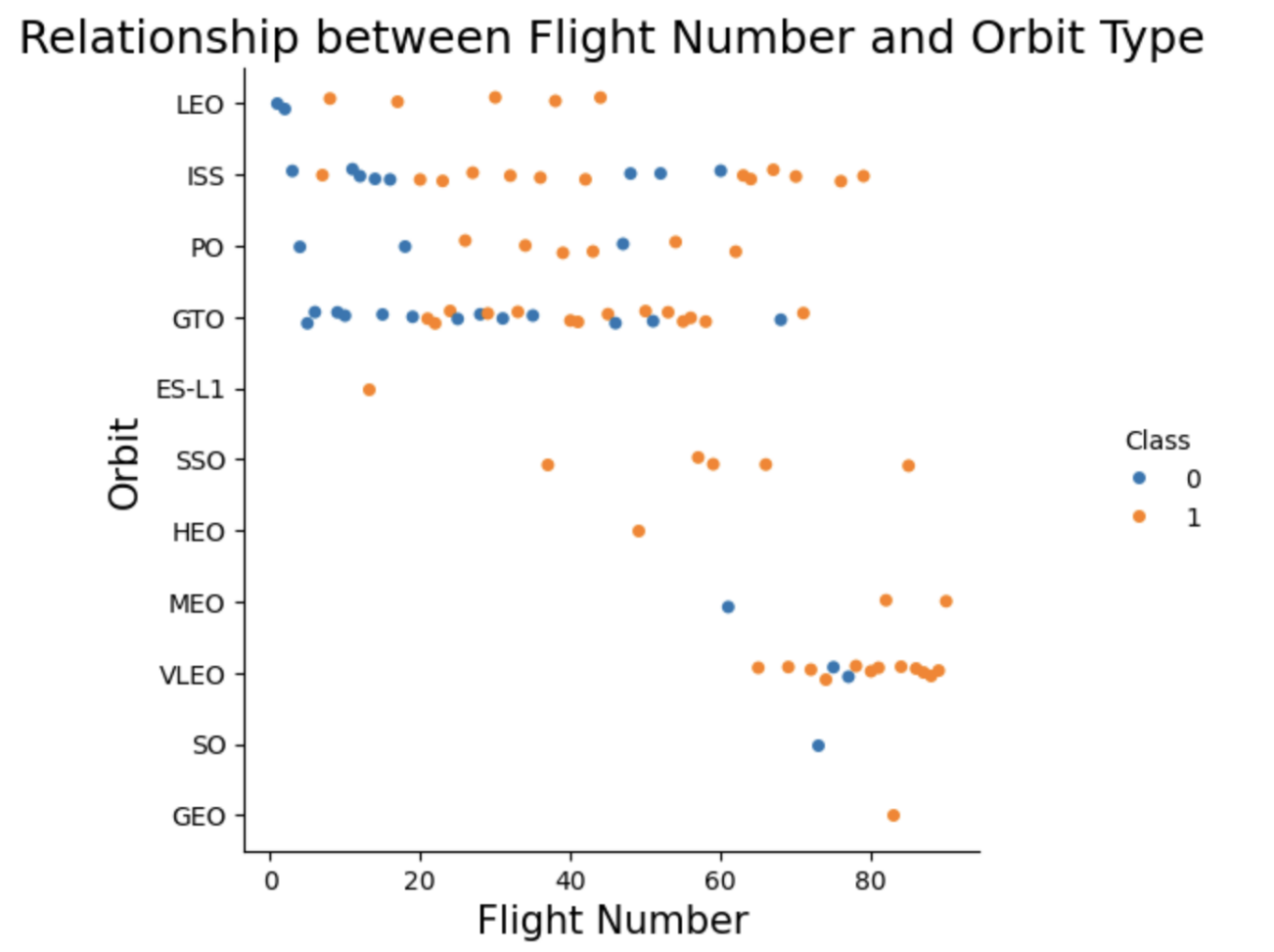
The plot shows that as the payload mass increases for the launch site CCAFS SLC 40, the success rate of the rocket also increases.

Success Rate vs. Orbit Type



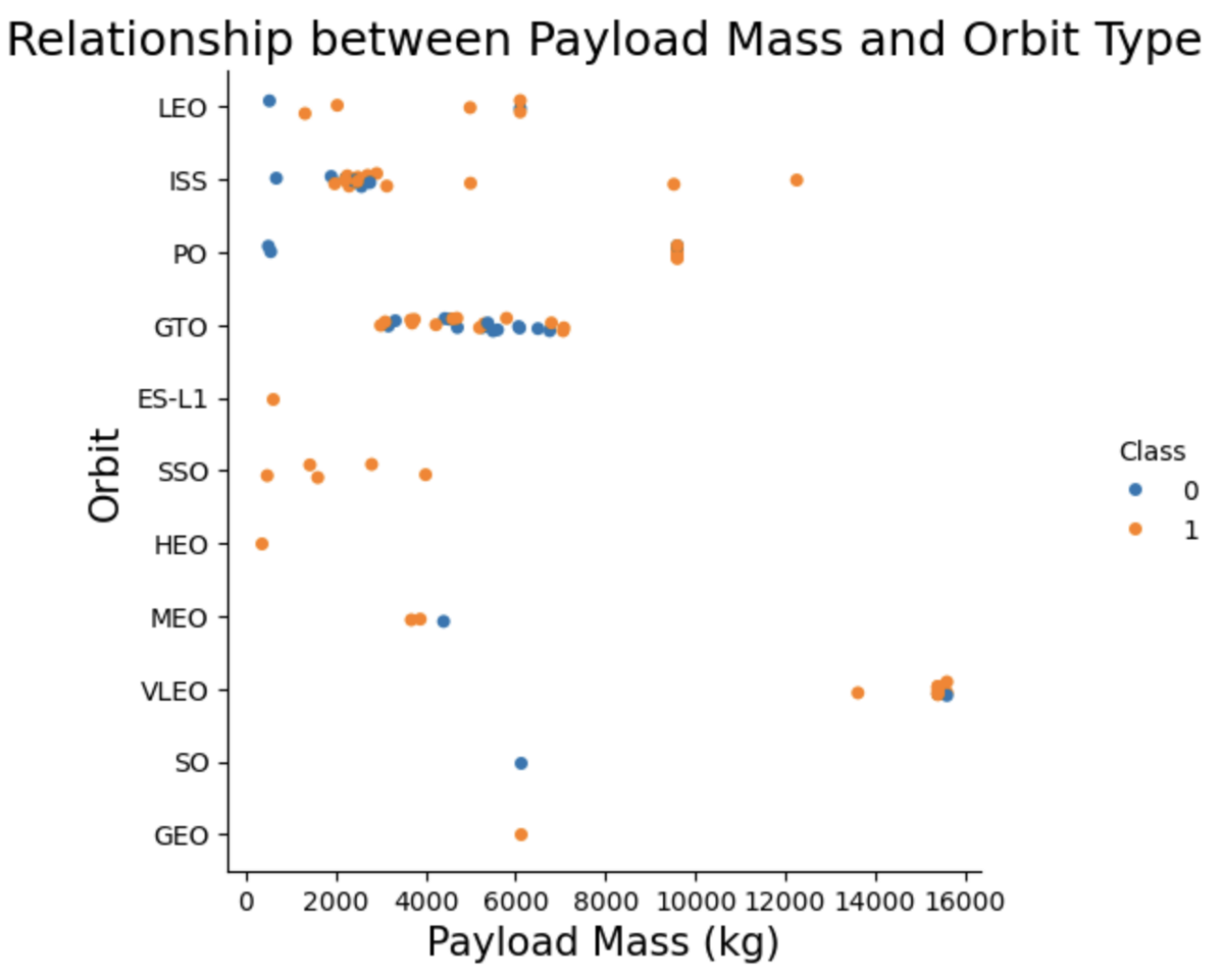
The plot indicates that the orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.

Flight Number vs. Orbit Type



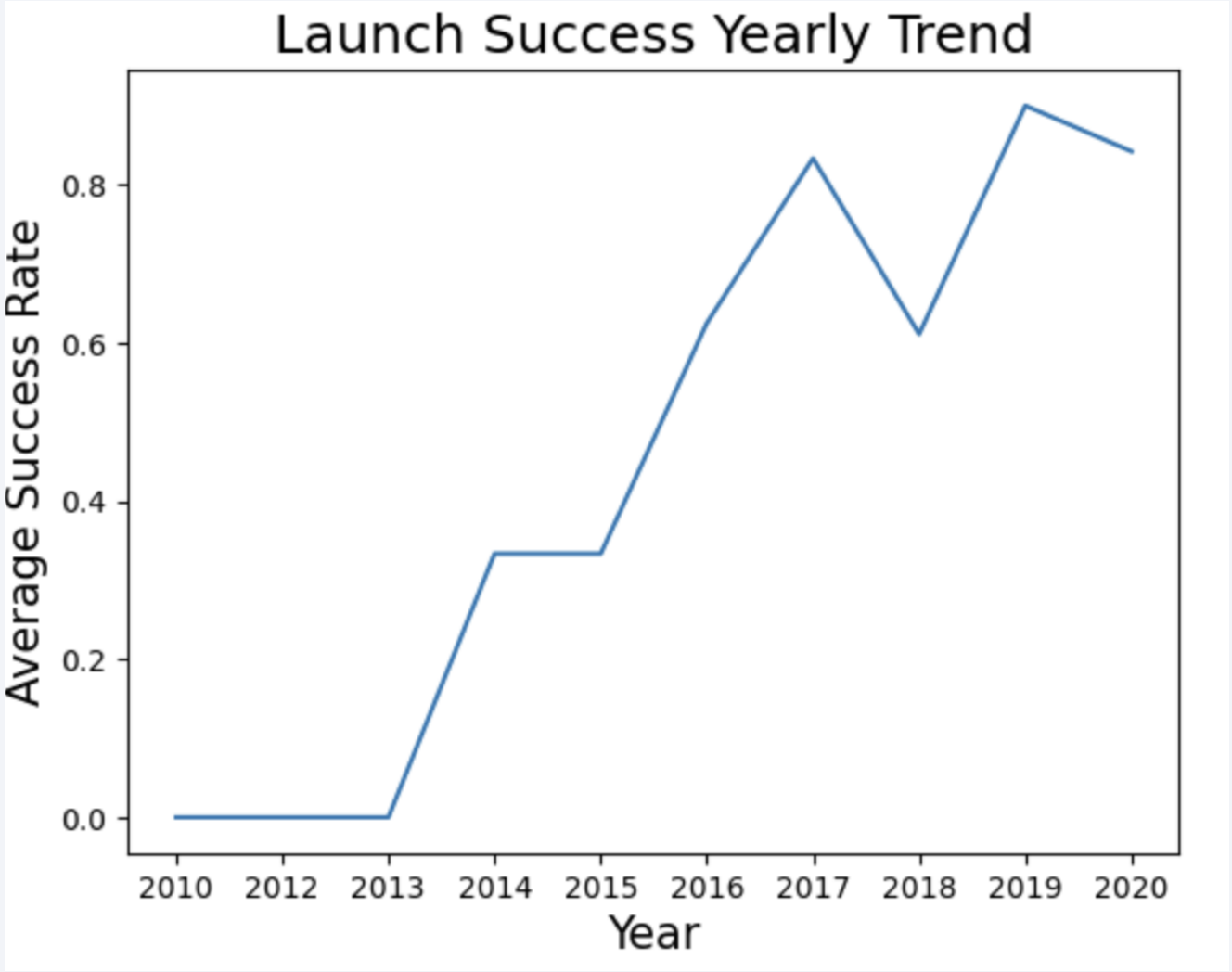
The plot below shows that in the LEO orbit, the success rate increases with the number of flights, while in the GTO orbit, there is no clear relationship between flight number and success.

Payload vs. Orbit Type



We can observe that for heavier payloads, successful landings are more frequent in the PO, LEO, and ISS orbits.

Launch Success Yearly Trend



The plot shows that the success rate has steadily increased from 2013 to 2020.

All Launch Site Names

- We used the keyword **DISTINCT** to display only the unique launch sites from the SpaceX data.

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where the launch sites start with 'CCA'.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

We calculated the total payload carried by boosters from NASA, which amounts to 45,596, using the query below.

```
%%sql
SELECT SUM(payload_mass__kg_) as total_payload FROM SPACEXTBL WHERE customer = 'NASA (CRS)';
```

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

total_payload
45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by the booster version F9 v1.1 using the query below.

```
%%sql
SELECT AVG(payload_mass__kg_) as average_payload FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%';
```

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

average_payload
2534.6666666666665

First Successful Ground Landing Date

We used **DISTINCT** to identify the value representing successful ground landings, and then applied the **MIN** function to find the date of the first successful landing outcome on the ground pad.

```
%%sql
select distinct Landing_Outcome from SPACEXTBL
```

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

Landing_Outcome
Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure
No attempt

```
%%sql
select min(date) from SPACEXTBL where landing_outcome = 'Success (ground pad)'
```

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

min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the **WHERE** clause to filter for boosters that have successfully landed on a drone ship and applied the **AND** condition to select boosters with a payload mass greater than 4000 but less than 6000.

```
%%sql
select booster_version, payload_mass__kg_ from SPACEXTBL
where landing_outcome = 'Success (drone ship)' and 4000 < payload_mass__kg_ and payload_mass__kg_ < 6000
group by booster_version, payload_mass__kg_
```

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

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

Total Number of Successful and Failure Mission Outcomes

We used the wildcard '%' in the LIKE clause to filter for mission outcomes where the result was either a success or a failure.

```
%%sql
select mission_outcome, count(mission_outcome) as total_nr
from SPACEXTBL
group by mission_outcome
```

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

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

Boosters Carried Maximum Payload

- We determined the booster that carried the maximum payload by using a subquery in the **WHERE** clause along with the **MAX()** function.

```
%%sql
SELECT DISTINCT booster_version
FROM SPACEXTBL
WHERE payload_mass__kg_ = (
    SELECT max(payload_mass__kg_)
    FROM SPACEXTBL
)
```

```
 * 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

2015 Launch Records

We used a combination of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes on the drone ship, along with their corresponding booster versions and launch site names for the year 2015.

```
%%sql
SELECT substr(Date, 6, 2) as month,
       landing_outcome,
       booster_version,
       launch_site
FROM SPACEXTBL
WHERE landing_outcome = 'Failure (drone ship)'
      AND substr(Date, 1, 4) = '2015'
GROUP BY month, landing_outcome, booster_version, launch_site;
```

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

month	Landing_Outcome	Booster_Version	Launch_Site
01	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

We used **GROUP BY** and **ORDER BY** to rank the count of landing outcomes (such as Failure on drone ship or Success on ground pad) between 2016-06-04 and 2010-03-20 in descending order.

```
%%sql
select landing_outcome, count(landing_outcome) as total_nr
from SPACEXTBL
where date between '2010-06-04' and '2017-03-20'
group by landing_outcome
order by total_nr desc
```

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

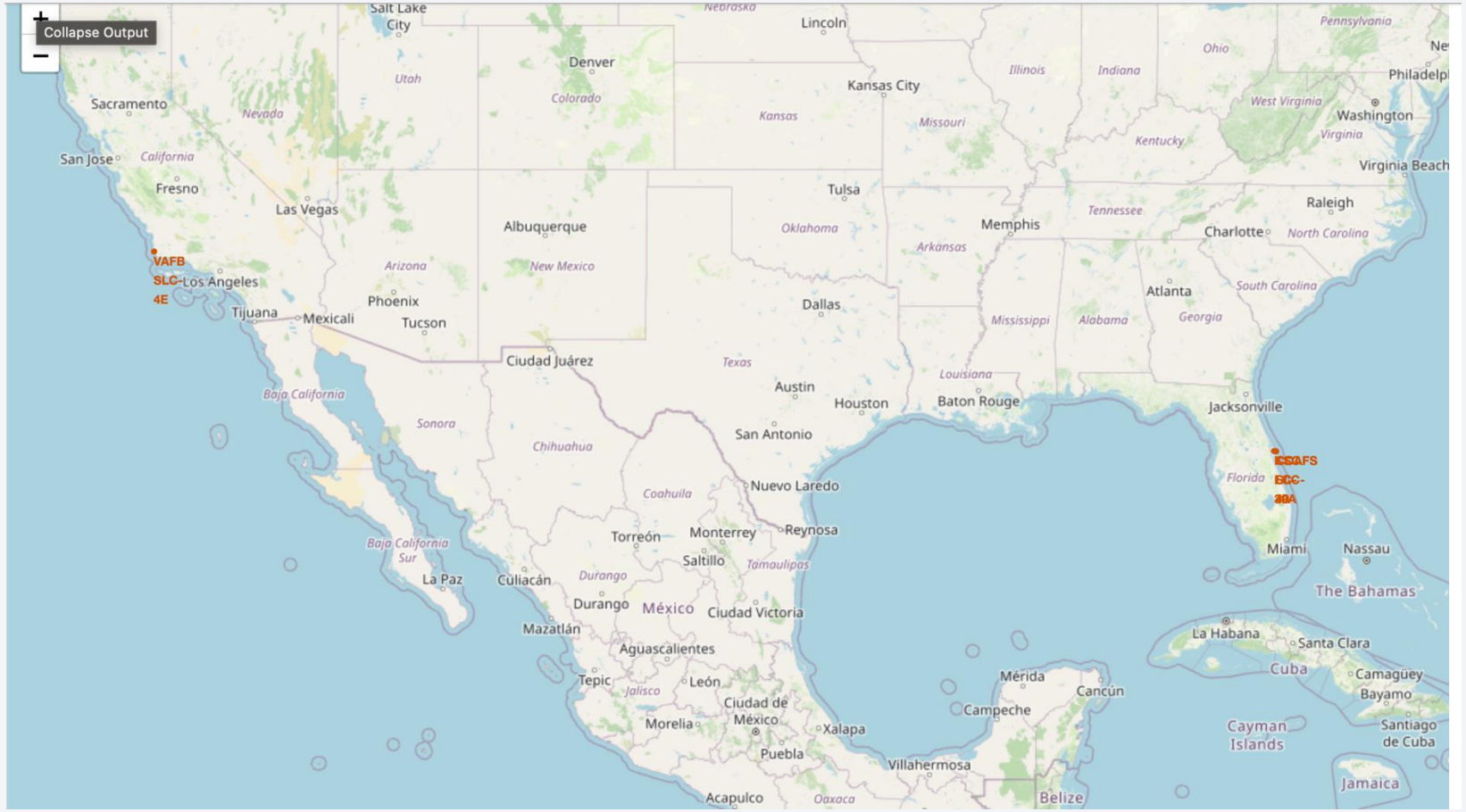
Landing_Outcome	total_nr
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

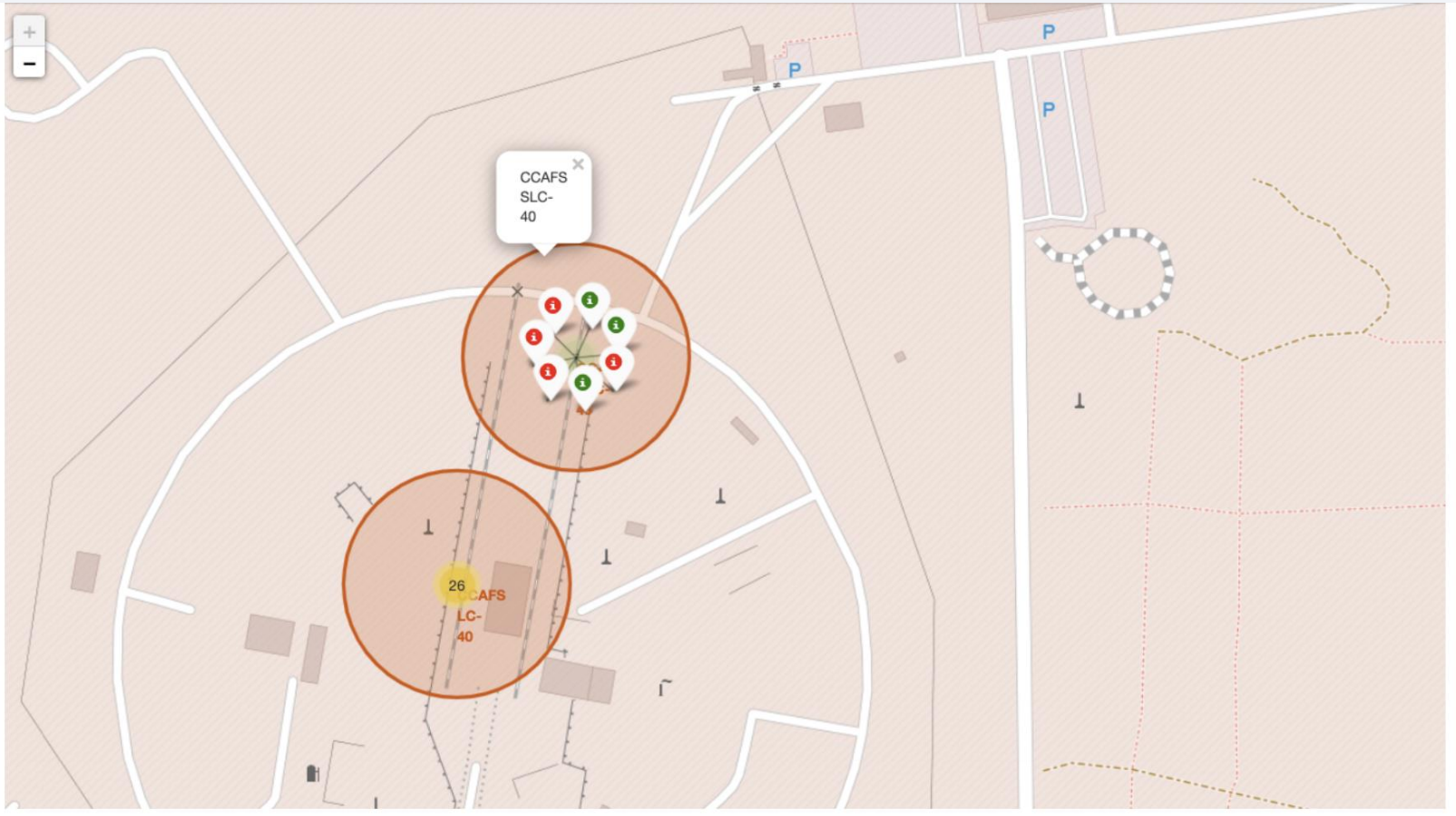
Visualisation of SpaceX Launch Sites on a Map

All SpaceX launch sites are located on the US coasts, specifically in Florida and California.



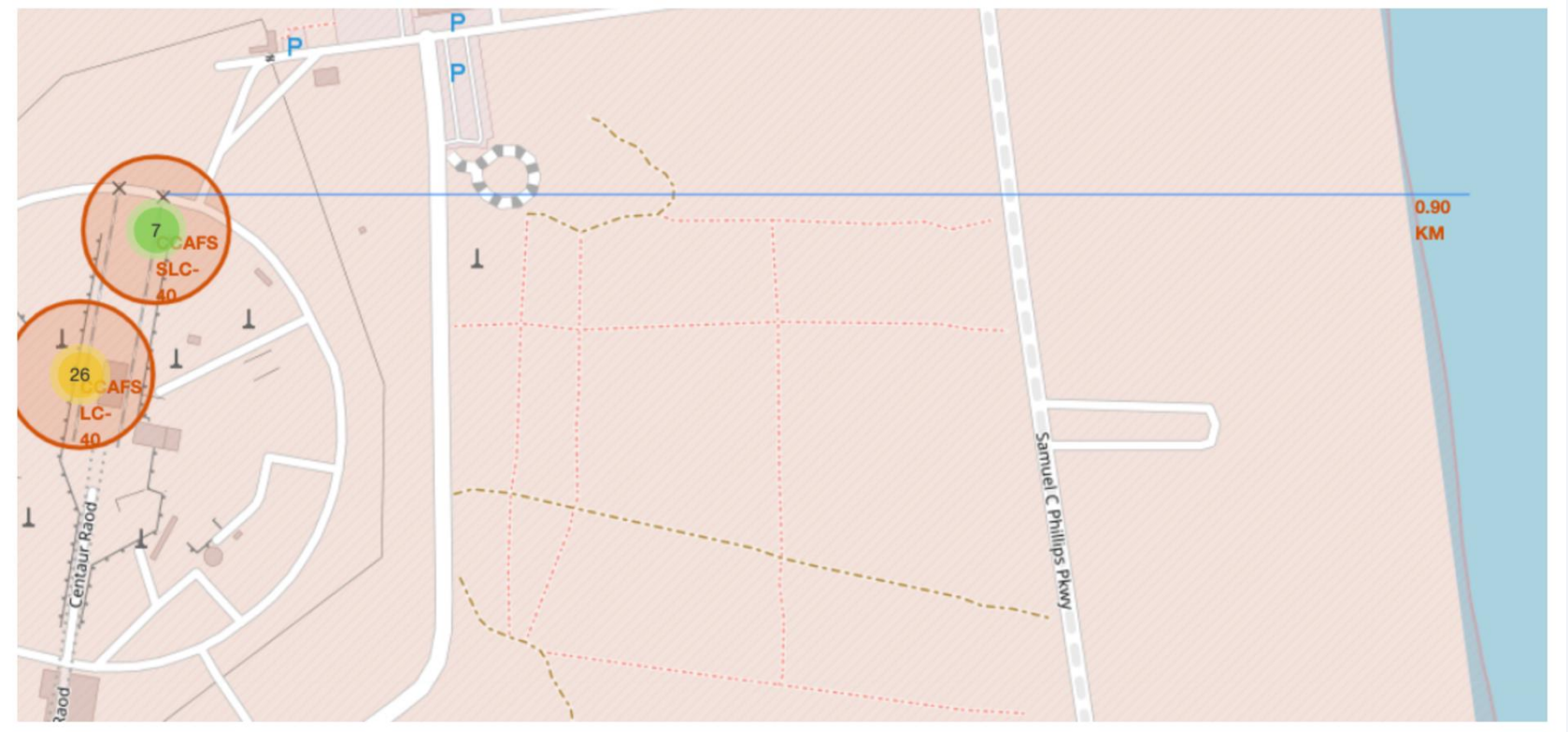
SpaceX Launch Sites: Success and Failure Markers

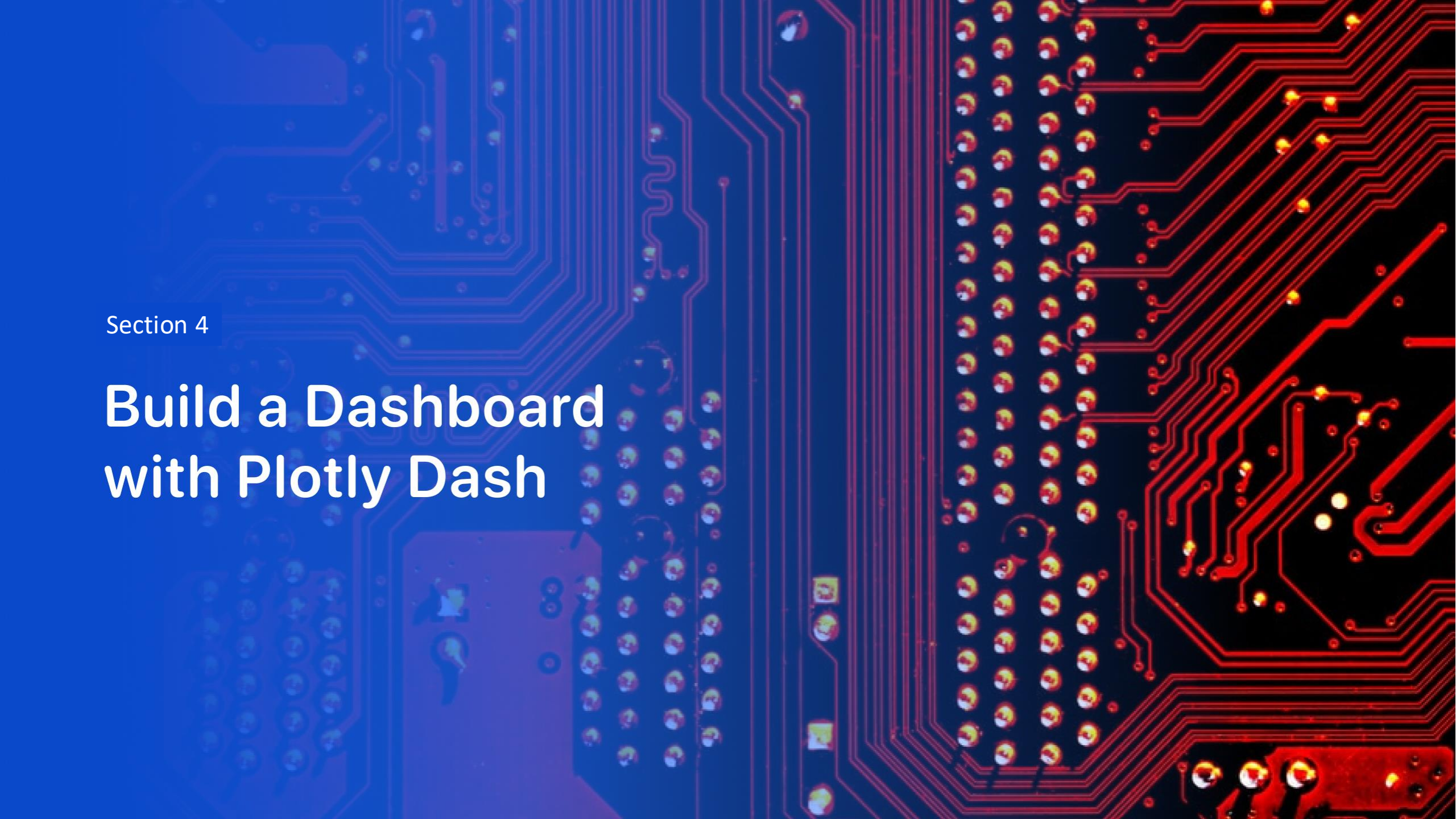
Green Marker: Successful launches, Red Marker: Failed launches



Proximity of Launch Sites to Railways and Highways

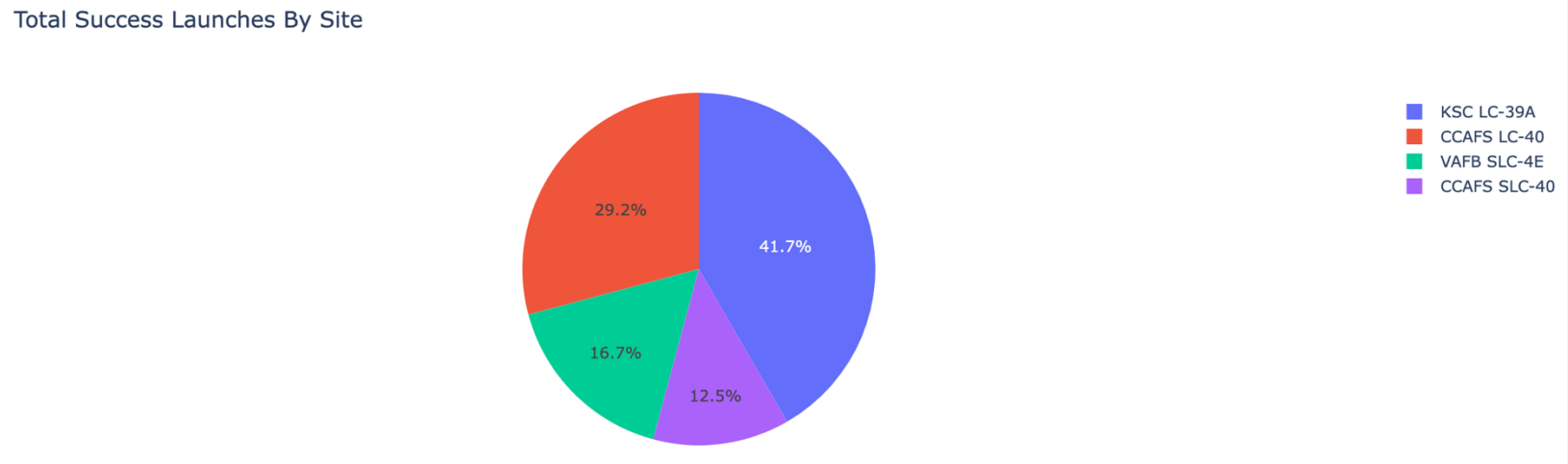
Launch site are relatively close to railway and highway for transport reasons.




Section 4
Build a Dashboard with Plotly Dash

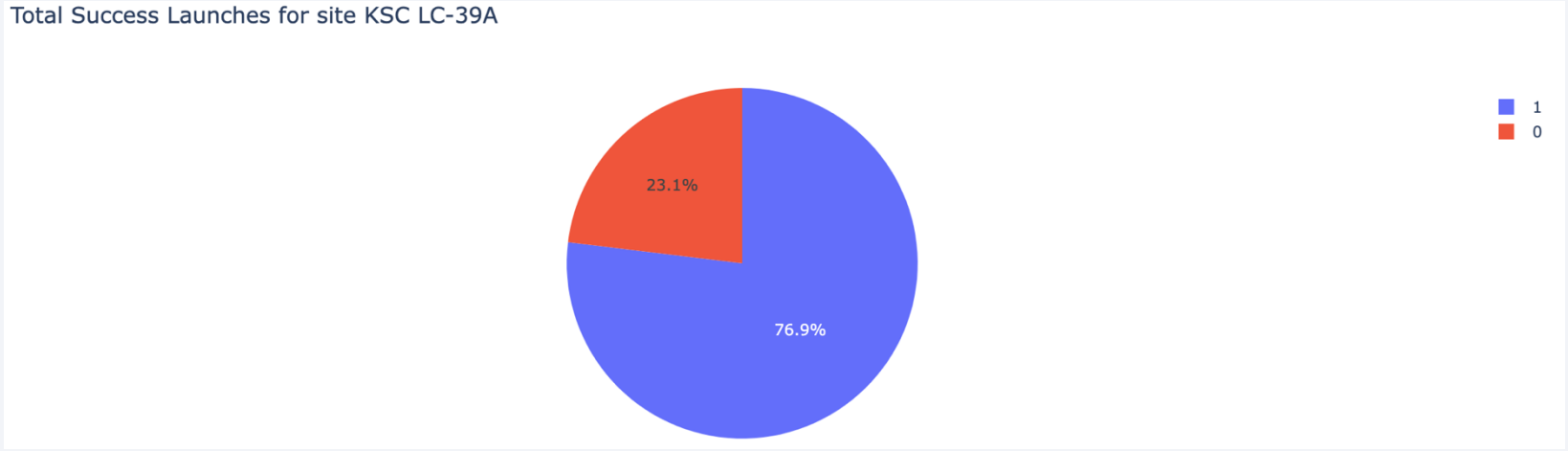
Launch Site with the Highest Launch Success Ratio

The chart visually indicates that the KSC LC-39A site is the most successful launch site, followed by CCAFS LC-40, VAFB SLC-4E, and CCAFS SLC-40. This distribution helps to understand the reliability and frequency of successful launches at different sites.



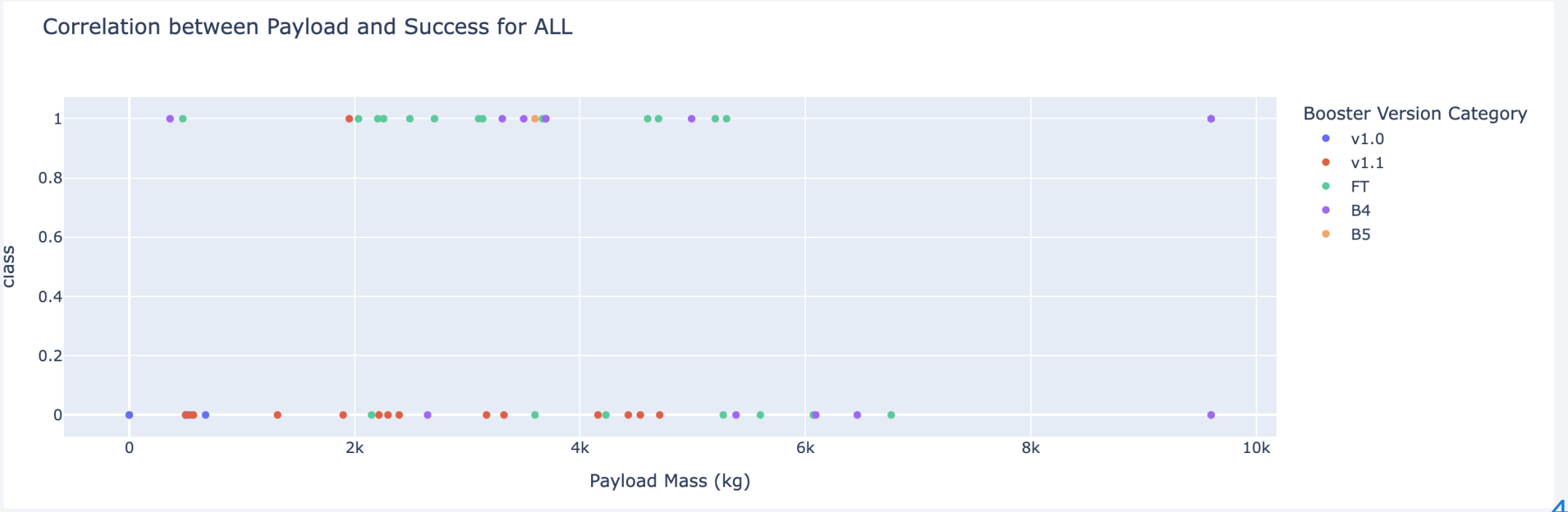
Pie chart showing the Launch site with the highest launch success ratio

The pie chart visually emphasizes that KSC LC-39A is a highly successful launch site, with nearly 77% of its launches being successful. This success rate makes KSC LC-39A a key player in SpaceX's launch operations, though improvements can still be made to minimize the failure rate.



Correlation between Payload and Success for ALL

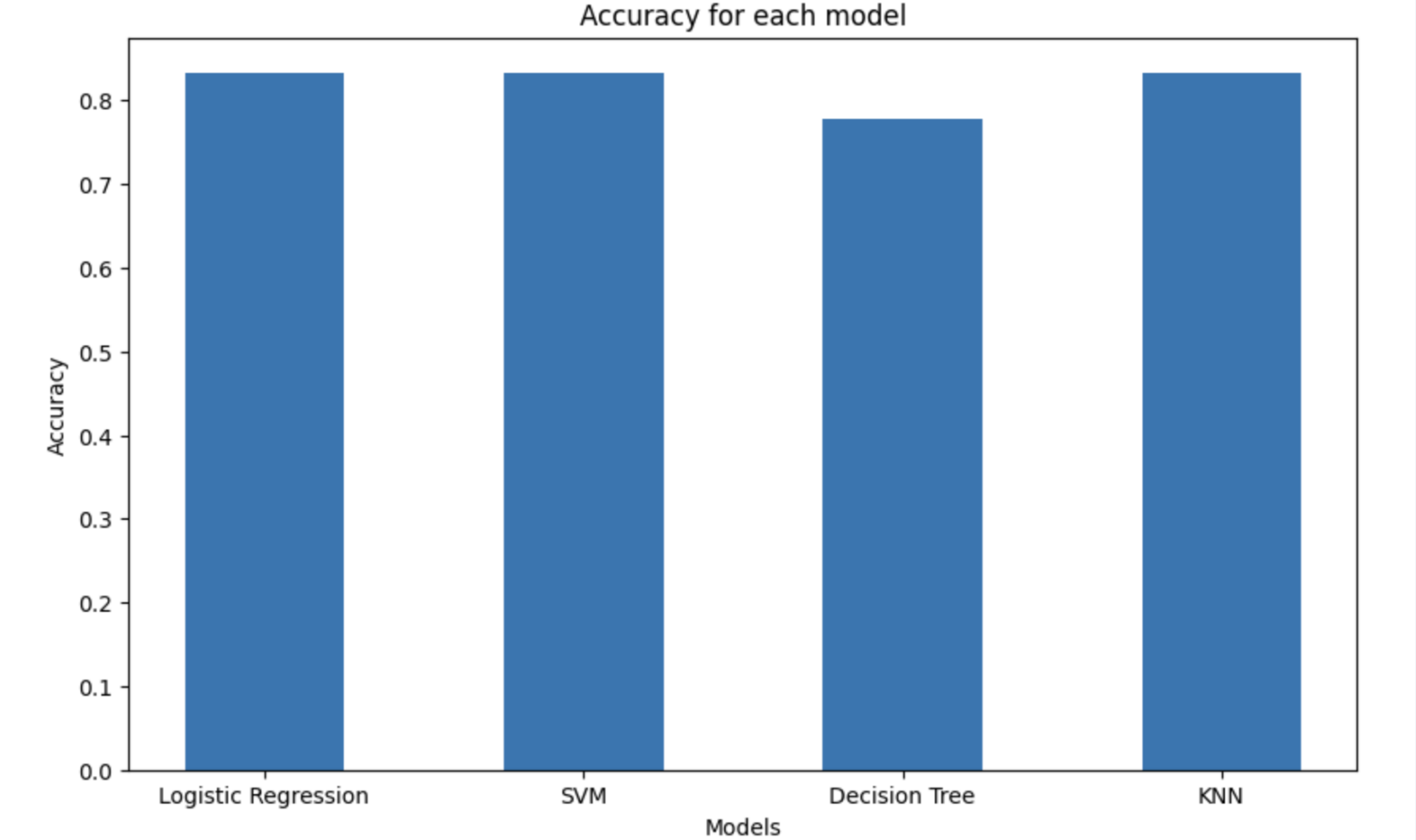
The scatter plot suggests that mission success does not depend heavily on the payload mass or the booster version used. Instead, other factors might play a more significant role in determining mission outcomes.



Section 5

Predictive Analysis (Classification)

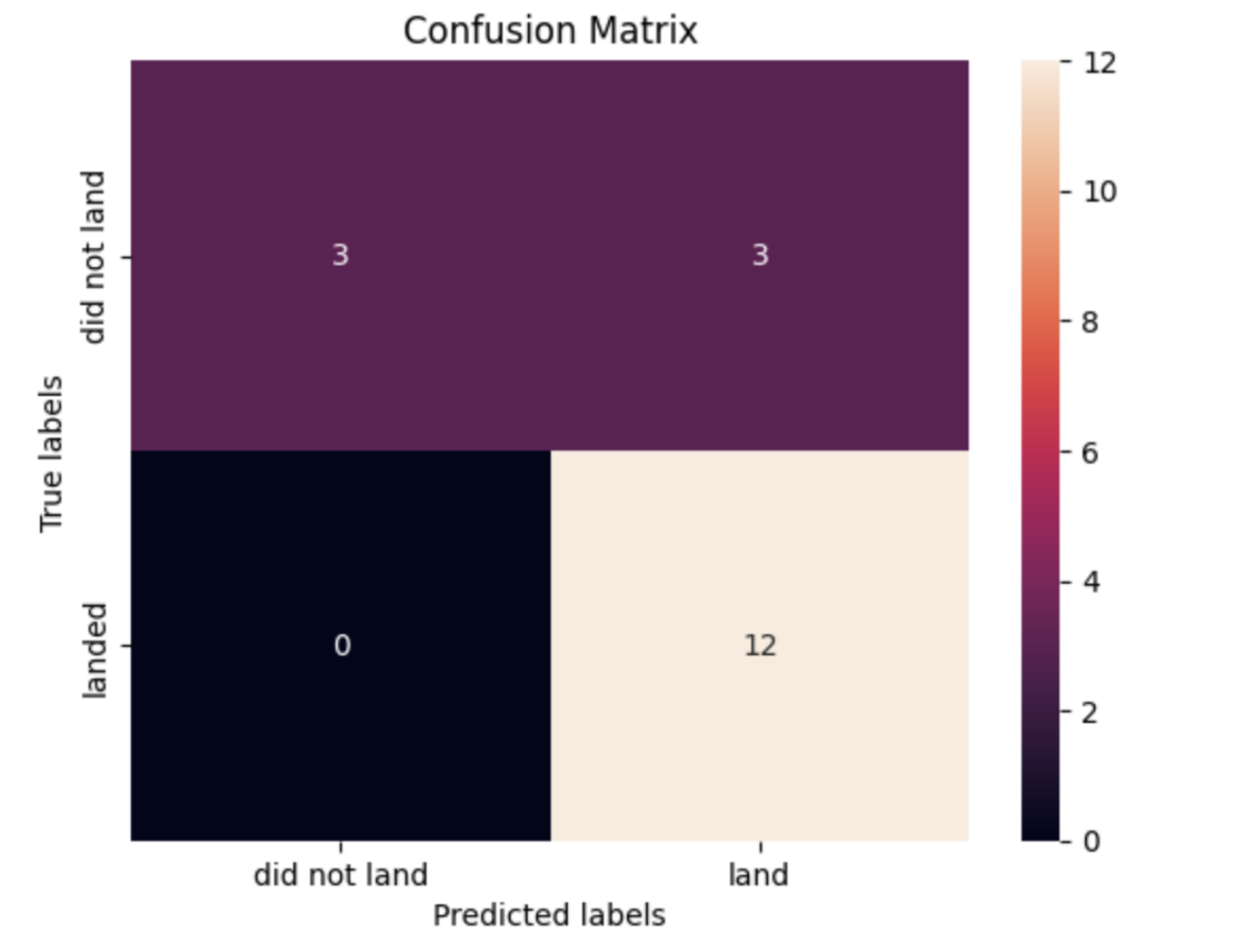
Classification Accuracy



It is evident that the model with the highest classification accuracy is **KNN**.

The y-axis represents accuracy, ranging from 0.0 to 1.0, and the x-axis lists the models. All four models have similar accuracy, with Logistic Regression, SVM, and KNN showing slightly higher accuracy than the Decision Tree. Among these, KNN stands out with the highest accuracy.

Confusion Matrix



The confusion matrix for the decision tree classifier demonstrates that the model can differentiate between the various classes. However, the primary issue is the presence of false positives, where unsuccessful landings are incorrectly classified as successful.

Conclusions

We can conclude that:

- The higher the number of flights at a launch site, the greater the success rate at that site.
- The launch success rate steadily increased from 2013 to 2020.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.
- KSC LC-39A had the most successful launches among all the sites.
- The Decision Tree classifier is the most effective machine learning algorithm for this task.

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!