IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

CHARLES Gregory
December 19th, 2024

Outline

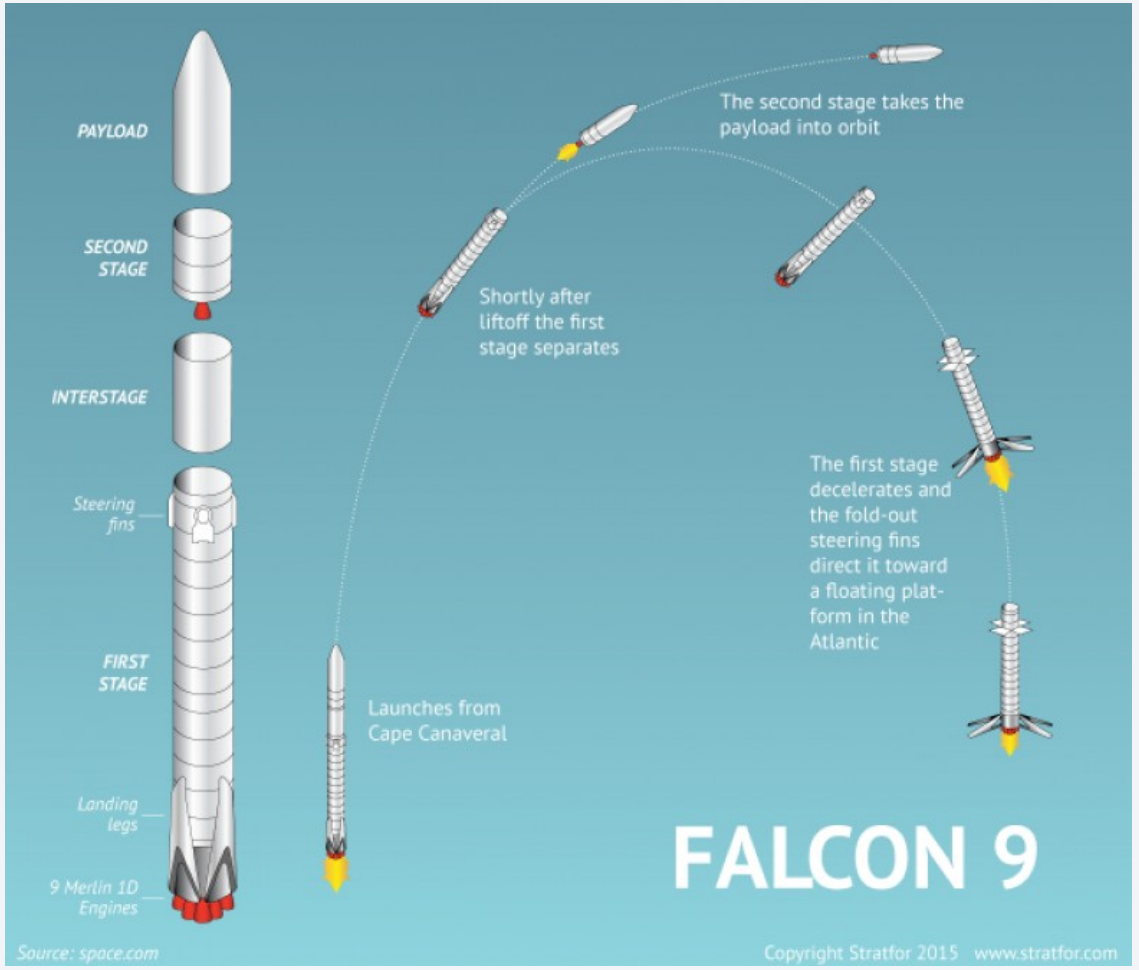
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- **Objective:**
  Predict Falcon 9 first-stage landing success to assess cost efficiency and competitiveness in the private space industry.
- **Summary of Methodologies:**
  Employed data collection (API, scraping), preprocessing (data wrangling), exploratory data analysis (EDA), interactive visualizations, and machine learning predictive modeling.
- **Summary of Results:**
  Achieved high predictive accuracy with actionable insights, supporting competitive decision-making for private space ventures.

Introduction

- The commercial space industry is expanding with companies like Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX.
- SpaceX leads the industry due to cost-efficient launches using reusable Falcon 9 rockets.
- **Problem Statement:**
  Can we predict the Falcon 9 first-stage landing success to estimate launch costs and gain a competitive edge?



Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

    Data was collected using the SpaceX API and Wikipedia web scraping

- Perform data wrangling

    Applied one-hot encoding to categorical features for model compatibility.

- 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

Data Collection Process:

- Collected data via GET requests to the SpaceX API.
- Decoded the JSON response using .json() and converted it to a pandas DataFrame with .json_normalize().
- Cleaned the data by checking for and filling missing values.
- Performed web scraping on Wikipedia for Falcon 9 launch records using BeautifulSoup.
- Extracted launch records from HTML tables, parsed them, and converted to a pandas DataFrame for further analysis.

Data Collection – SpaceX API

- **Objective:** Compile accurate datasets to analyze Falcon 9 launches.
- **Key Process:**
  - REST API Calls: Queried SpaceX's API for structured JSON data on launches.
  - Transformation: Converted JSON to Pandas DataFrames for analysis.
  - Supplemental Data: Used web scraping to fill data gaps (e.g., launch comments, extra payload details).
- **Github:**
  SpaceX Falcon 9 - Data Collection API Lab

[Install Libraries] → [Import Required Libraries] → [Send GET Request to SpaceX API]

→ [Parse JSON Response] → [Convert JSON to Pandas DataFrame] → [Filter Relevant Columns]

→ [Clean Data]:

- Remove rows with multiple cores or payloads
- Format date and restrict date range

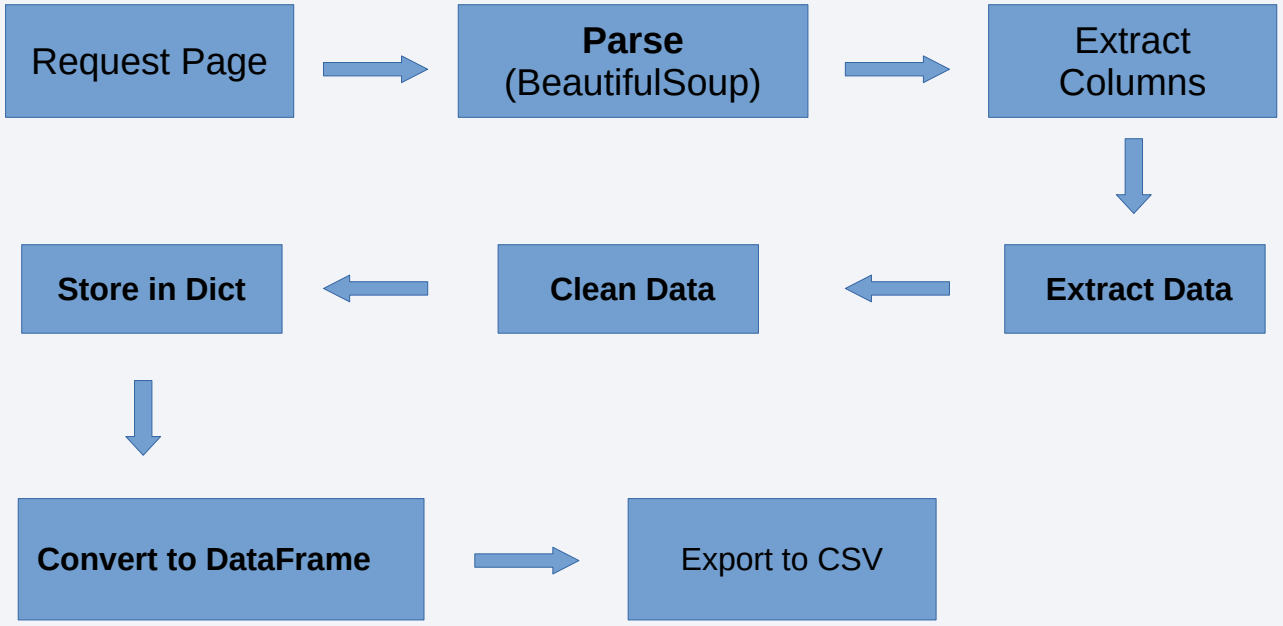
→ [Fetch Rocket Details] → [Fetch Payload Details] → [Fetch Launchpad Details]

→ [Fetch Core Details] → [Combine Data into Final Dictionary] → [Create Pandas DataFrame]→ [Filter Falcon 9 Launches]→ [End]

Data Collection - Scraping

- **HTML Parsing** - Extracting data by navigating the HTML structure using libraries like Beautiful Soup (Python).
- **HTTP Requests** - Sending requests to retrieve web pages or APIs using tools like requests

Github:
SpaceX Falcon 9 - Web Scraping



Data Wrangling

Key Phrases & Flowchart

1. ***Import and Load:*** Imported libraries (pandas, numpy) and loaded the dataset.
2. ***Handle Missing Data:*** Identified and calculated percentages of missing values.
3. ***Data Categorization:*** Classified columns into numerical and categorical types.
4. ***Exploratory Data Analysis (EDA):*** Counted launches per site and occurrences per orbit.
5. **Outcome Analysis:** Identified landing outcomes and created a binary classification - label (landing_class).
6. **Label Assignment:** Assigned 1 for successful landings and 0 for unsuccessful landings.
7. **Export Processed Data:** Saved the cleaned and labeled dataset for future use

Github

SpaceX Falcon 9 - Data Wrangling

EDA with Data Visualization

1) **Scatterplots:**

Purpose: To analyze relationships between key variables such as flight numbers, payload mass, launch sites, and orbit types.

Reason: Scatterplots are ideal for visualizing how two continuous variables interact and their influence on success rates.

2) **Bar Charts:**

Purpose: To compare categorical data, such as success rates across orbit types.

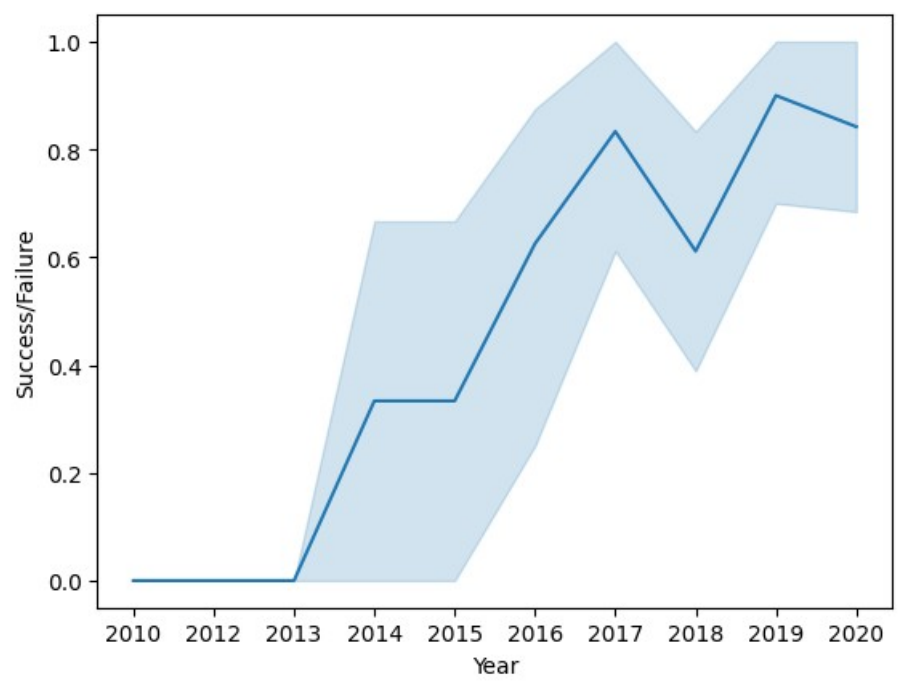
Reason: Bar charts effectively summarize and compare proportions or percentages.

3) **Line Chart:**

Purpose: To track trends over time, specifically the yearly success rate.

Reason: Line charts are well-suited for showing changes and trends across chronological data.

These charts were chosen to provide a comprehensive exploration of variable relationships, success patterns, and historical trends.



Github:
SpaceX Falcon 9 - EDA with Visualization

EDA with SQL

SQL queries help uncover deeper insights into the data. Key queries include:

- Unique launch sites
- Records for launch sites starting with "CCA%"
- Total payload mass of NASA-launched boosters
- Average payload mass for F9 v1.1 boosters
- Date of the first successful landing
- Boosters that successfully landed on drone ships with payloads between 4000-6000 kg
- Total successful and failed mission outcomes
- Boosters carrying the maximum payload mass
- Failed outcomes on drone ships, including booster version and launch site since 2005

Github: SpaceX Falcon 9 - SQL Notebook

Build an Interactive Map with Folium

Interactive Mapping with Folium in Jupyter Notebooks

- **Launch Site Visualization:**

Visualize launch sites on an interactive map, displaying both successful and failed launches for each site.

- **Geospatial Analysis:**

Calculate and display the distance from one of Florida's launch sites to nearby features such as the coastline, railroad, city, and highway.

Draw lines on the map to represent distances, providing a clear spatial understanding of proximity.

- **Interactive Exploration:**

The map enables interactive exploration of launch data and spatial relationships, helping users understand the geographical context of launch sites relative to key locations.

Note: The interactive maps created with Folium cannot be viewed directly on GitHub due to rendering limitations.

Github: SpaceX Falcon 9 - Launch Sites Locations Analysis with Folium

Build a Dashboard with Plotly Dash

Interactive Visualizations with Plotly Dash

- **Plot 1:** Successful Launches by Site (Pie Chart)

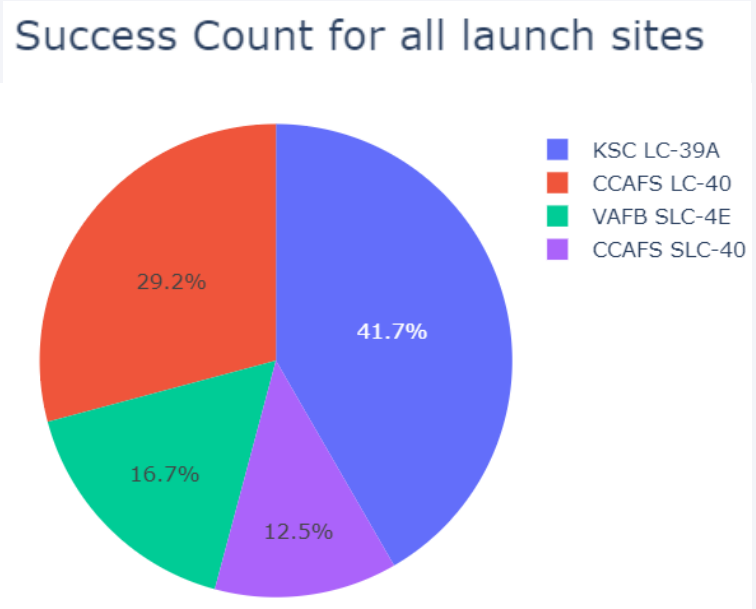
Interactive pie chart displaying the percentage of successful launches for each site.

Users can select individual sites to dynamically view the proportion of successes for that specific site.

- **Plot 2:** Payload vs. Successful Launches (Scatter Plot)

Visualizes the relationship between Payload (in kg) and the number of successful launches.

Interactive filtering options allow users to select specific sites and adjust the payload range to refine the analysis.



Github: SpaceX Falcon 9 - Launch Sites Locations Analysis with Folium

Predictive Analysis (Classification)

Model Comparison and Performance Evaluation

Models Built:

- Logistic Regression, Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbors (KNN)

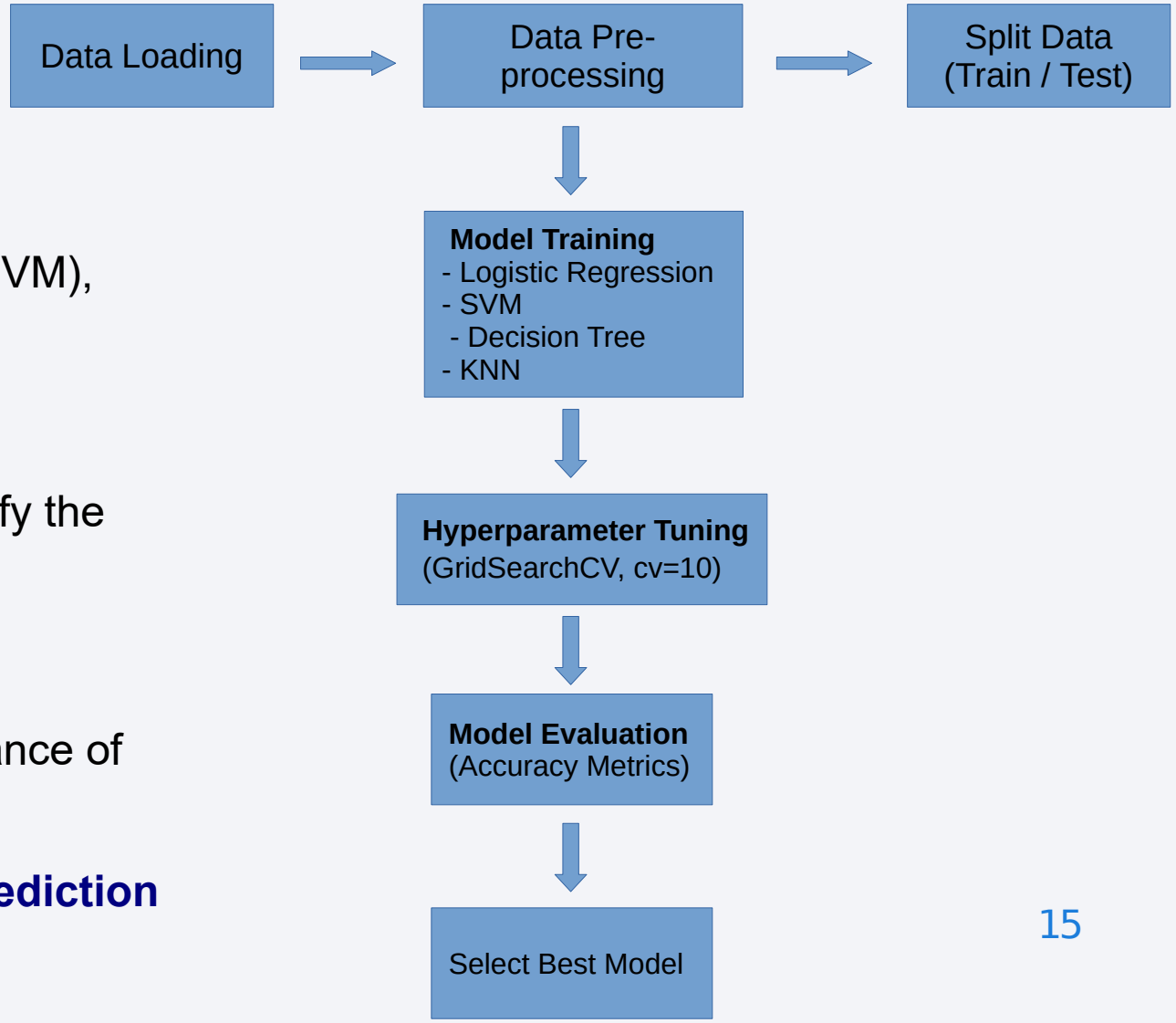
Hyperparameter Tuning:

- Grid Search with Cross-Validation used to identify the best hyperparameters for each model.

Performance Evaluation:

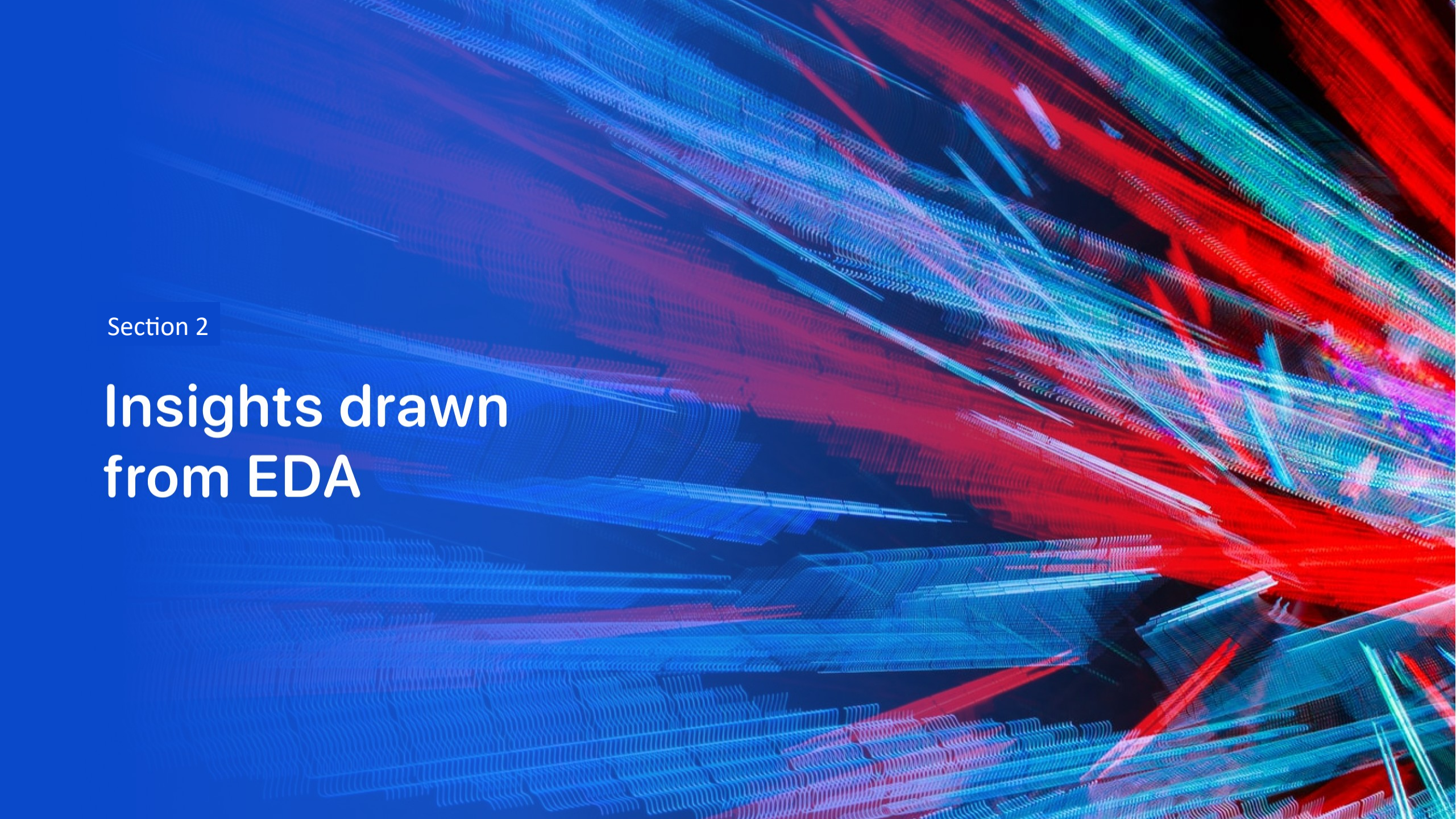
- Accuracy metrics used to compare the performance of each model and select the best-performing one.

Github: Space X Falcon 9 - Machine Learning Prediction



Results

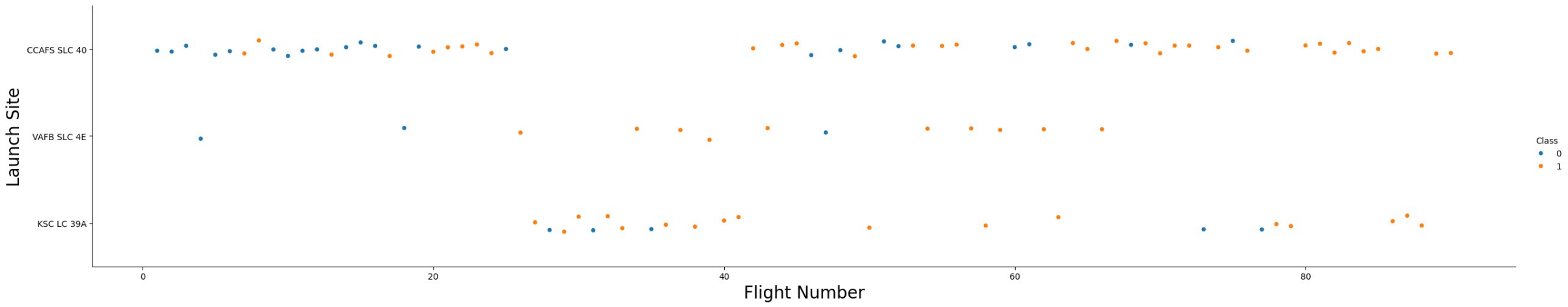
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

Based on the plot, it is observed that there is a positive correlation between the flight volume at a launch site and the corresponding success rate.

Specifically, an increase in the number of flights at a launch site appears to be associated with a higher success rate.



Payload vs. Launch Site

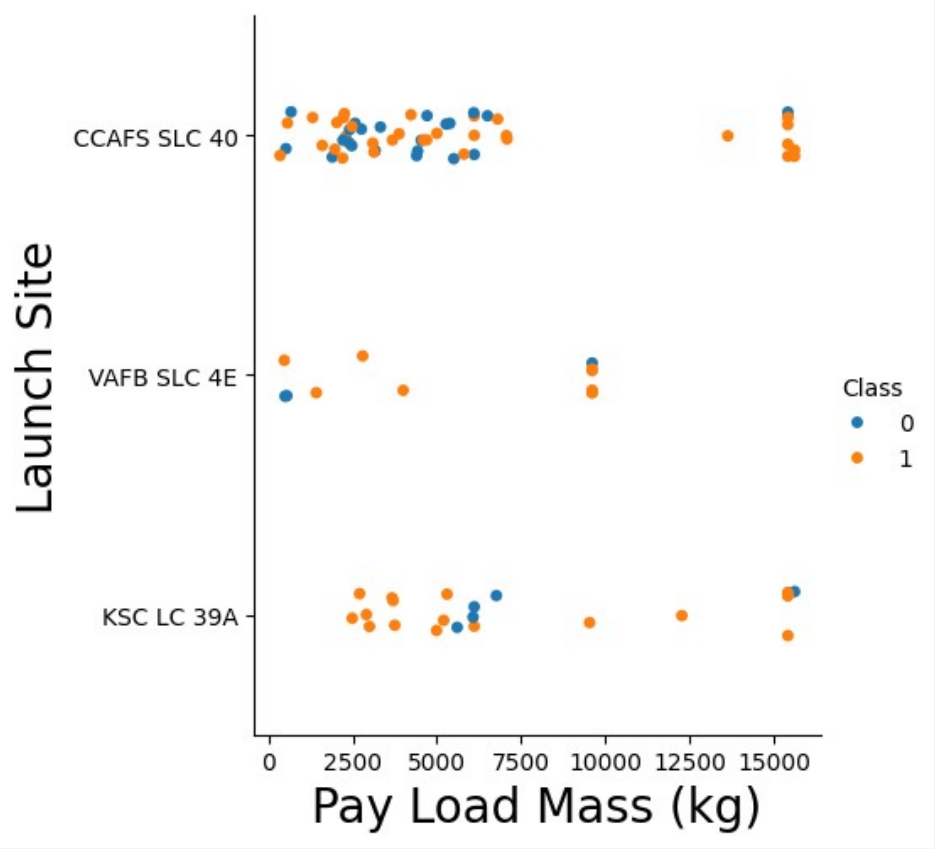
The analysis of the scatter plot depicting Payload vs. Launch Site provides insights into the distribution of payload capacities across various launch sites and their relationship to successful landings:

- **VAFB-SLC Launch Site:**
  - This site does not handle heavy payloads (payload mass > 10,000 kg).
  - All payloads launched here are ≤10,000 kg.

- **CCAFS SLC Launch Site:**
  - This site supports payloads in two distinct ranges: up to 7,500 kg and 12,500–15,000 kg.
  - There is no data for payloads in the intermediate range (7,500–12,500 kg).

- **KSC LC 39A Launch Site:**
  - This site handles a broader range of payloads, spanning from 2,500 kg to slightly over 15,000 kg, showing the highest payload diversity among the three sites.

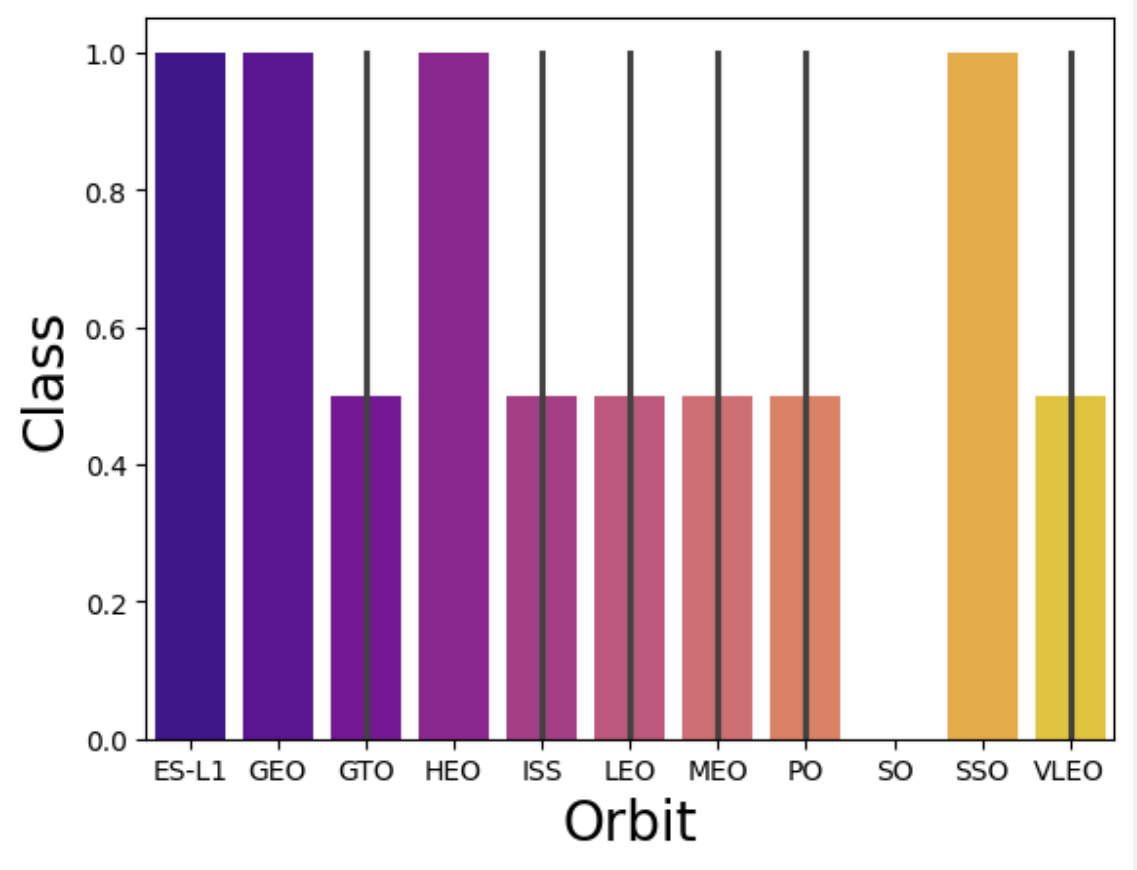
Payload vs. Landing Success:
- The scatter plot suggests a positive correlation between payload mass and landing success.
- As payload mass increases, the likelihood of a successful landing also tends to increase.



Success Rate vs. Orbit Type

Based on the plot the highest success rates are:

- ES-L1,
- GEO,
- HEO,
- SSO



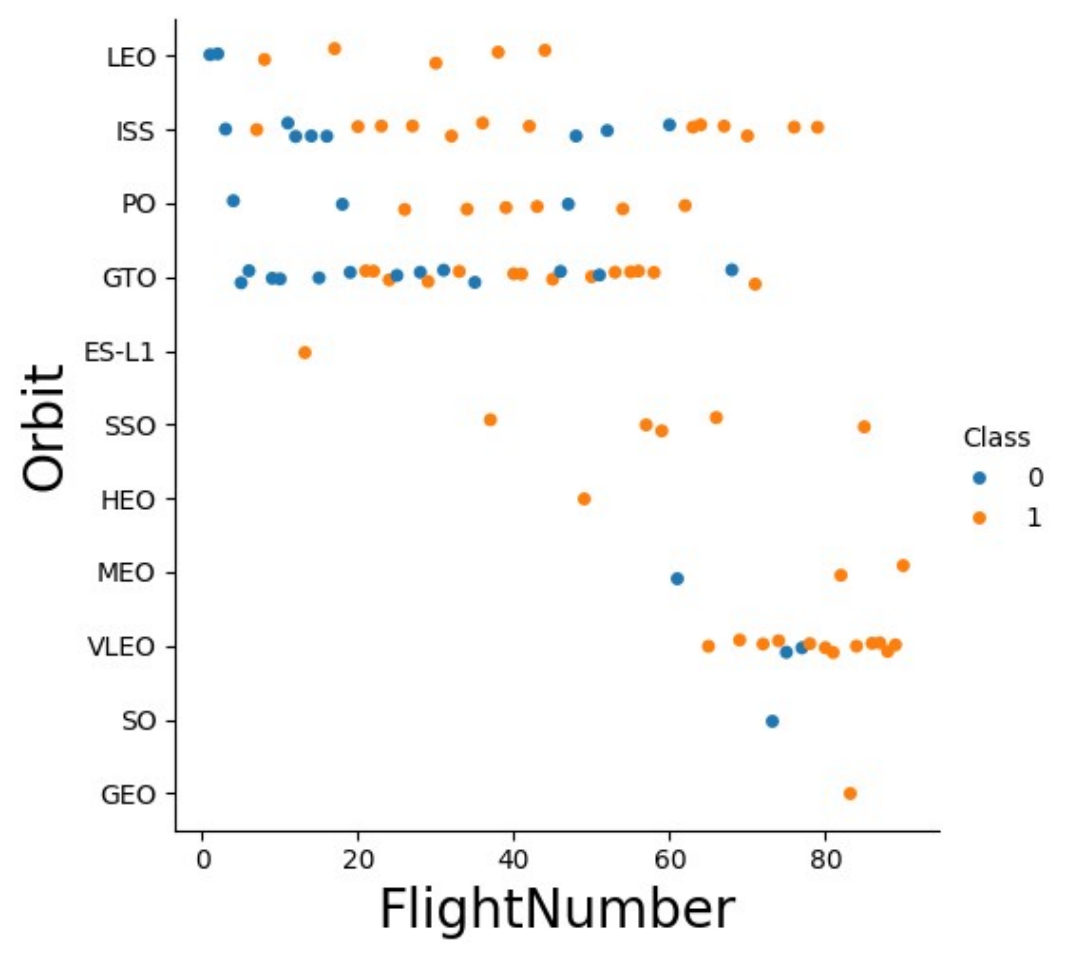
Flight Number vs. Orbit Type

Success rates correlate with the number of flights:

- LEO

no such relationship:

- GTO

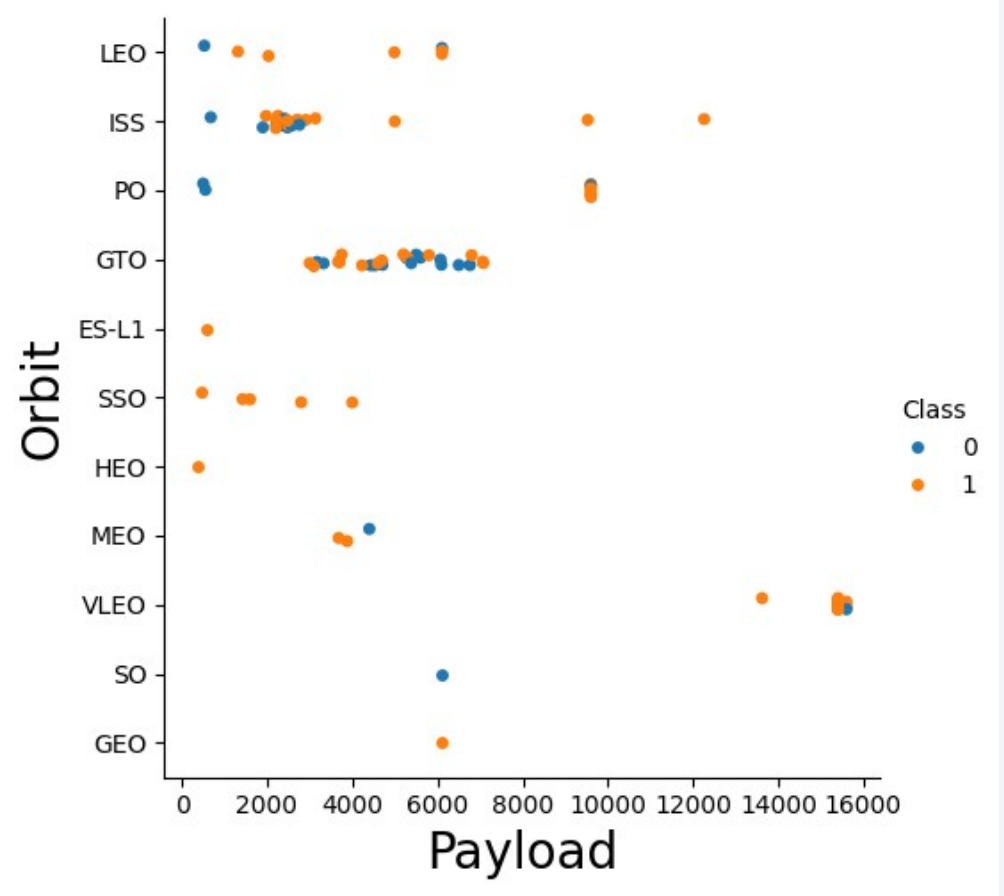


Payload vs. Orbit Type

Heavy payloads with
higher success rates:

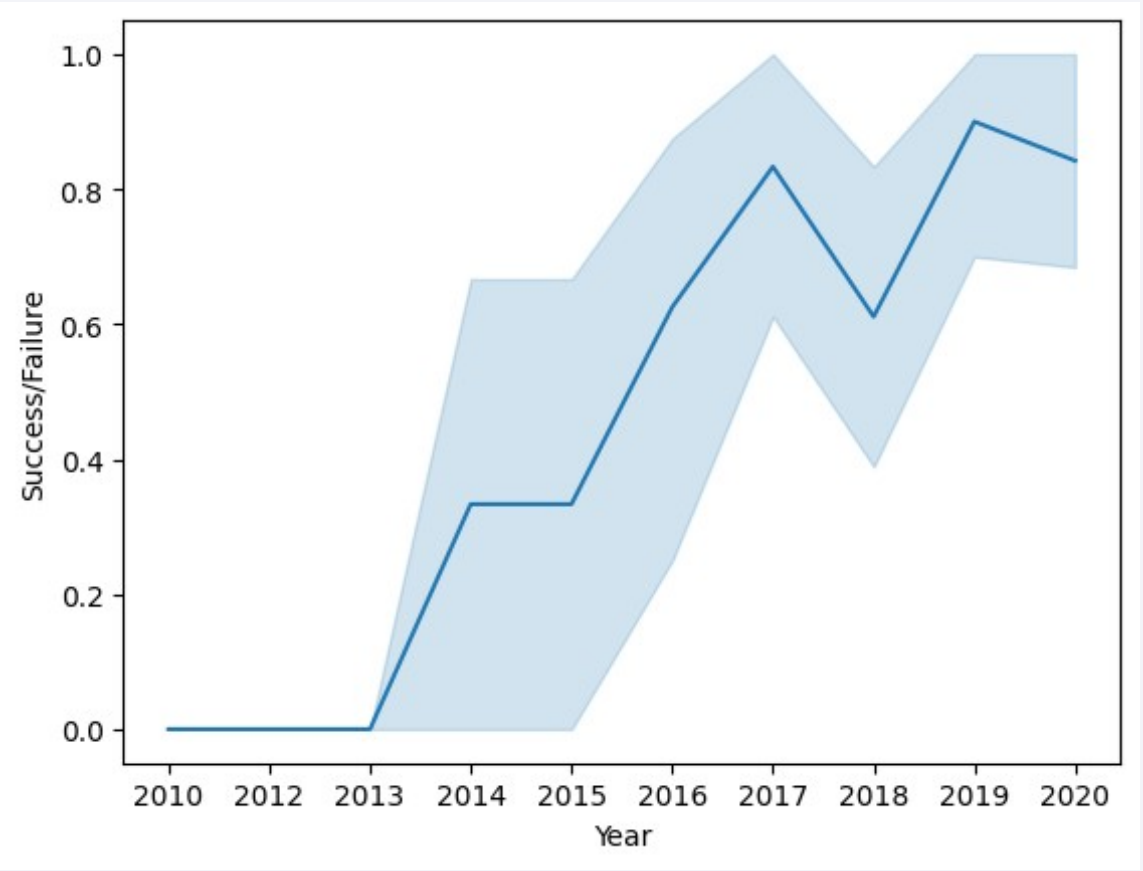
- Polar,
- LEO,
- ISS orbits.

However, in GTO orbit, the data shows a mix of both successful and unsuccessful landings, making it difficult to establish a clear trend.



Launch Success Yearly Trend

The success rate steadily increased from 2013 to 2017, with a brief dip in 2018, but overall showed an upward trend, particularly after 2015.



All Launch Site Names

The function **distinct()** in this query is used to return only unique values from the **LAUNCH_SITE** column in the **SPACEXTBL** table, eliminating any duplicates.

```
In [10]:
 %sql select distinct(LAUNCH_SITE) from SPACEXTBL
 * 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** function in this query filters the results to show rows where the **LAUNCH_SITE** column starts with "**CCA**". The **LIMIT 5** restricts the output to the first 5 matching records.

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * 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

Total Payload Mass

The **SUM()** function calculates the total payload mass (in kilograms) for rows where the **CUSTOMER** is '**NASA (CRS)**' in the **SPACEXTBL** table.

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

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

The **AVG()** function calculates the average payload mass (in kilograms) for rows where the **BOOSTER_VERSION** is '**F9 v1.1**' in the **SPACEXTBL** table.

```
In [13]:
 %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
Out[13]:
```

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

The **MIN()** function returns the earliest date (**DATE**) of a successful ground pad landing, as indicated by the **Landing_Outcome** being '**Success (ground pad)**', from the **SPACEXTBL** table.

```
%sql SELECT min(DATE) AS "First successful ground pad landing" FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

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

First successful ground pad landing
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 P
```

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

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

This query retrieves the **BOOSTER_VERSION** for rows where the **Landing_Outcome** is **'Success (drone ship)'** and the **PAYLOAD_MASS__KG_** is between **4000** and **6000** kilograms in the **SPACEXTBL** table.

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT 'Success' AS "Outcome",
    count(*) AS "Count" FROM SPACEXTBL
    WHERE Landing_Outcome LIKE 'Success%'
    UNION ALL
    SELECT 'Failure' AS "Outcome",
    count(*) AS "Count" FROM SPACEXTBL
    WHERE Landing_Outcome NOT LIKE 'Success%'
    UNION ALL
    SELECT 'All' AS "Outcome",
    count(*) AS "Count" FROM SPACEXTBL;
```

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

Outcome	Count
Success	61
Failure	40
All	101

This query counts and summarizes the number of landing outcomes in the **SPACEXTBL** table:

1) It first counts rows where **Landing_Outcome** starts with "**Success**" and labels the result as "**Success**".

2) Then, it counts rows where **Landing_Outcome** does **not** start with "**Success**" and labels the result as "**Failure**".

3) Finally, it counts all rows in the table and labels the result as "All".

UNION ALL combines these results into one output showing the counts for "**Success**", "**Failure**", and "**All**" outcomes.

Boosters Carried Maximum Payload

```
%%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

This query retrieves the unique **booster_version** values from the **SPACEXTBL** table where the **PAYLOAD_MASS__KG_** is equal to the maximum payload mass in the table.

The subquery (**SELECT max(PAYLOAD_MASS__KG_) FROM SPACEXTBL**) finds the highest payload mass, and the main query selects the booster_version(s) corresponding to that value.

2015 Launch Records

This query retrieves the **booster_version**, **launch_site**, **Landing_Outcome**, and the **month** (extracted from the **DATE** column) for rows where the **Landing_Outcome** is '**Failure (drone ship)**' and the **year** in the **DATE** column is **2015**. The **substr(Date, 0, 5)** function extracts the **year** from the **DATE** column, and **substr(Date, 6, 2)** extracts the **month**.

```
%%sql SELECT booster_version, launch_site, Landing_Outcome, substr(Date, 6,2) as month
    FROM SPACEXTBL
    WHERE Landing_Outcome = 'Failure (drone ship)'
    AND substr(Date,0,5)='2015'
```

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

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

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

```
%%sql SELECT Landing_Outcome, COUNT(*) AS "Count"
    FROM SPACEXTBL
    WHERE DATE BETWEEN '2010-06-04' and '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY Count DESC
    ;
```

```
 * 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
Failure (parachute)	2
Precluded (drone ship)	1

This query counts the number of occurrences of each **Landing_Outcome** in the **SPACEXTBL** table within the date range from '**2010-06-04**' to '**2017-03-20**'.

It **groups** the results **by Landing_Outcome**, orders them by the count in descending order (**ORDER BY Count DESC**), and labels the count as "**Count**".

Section 3

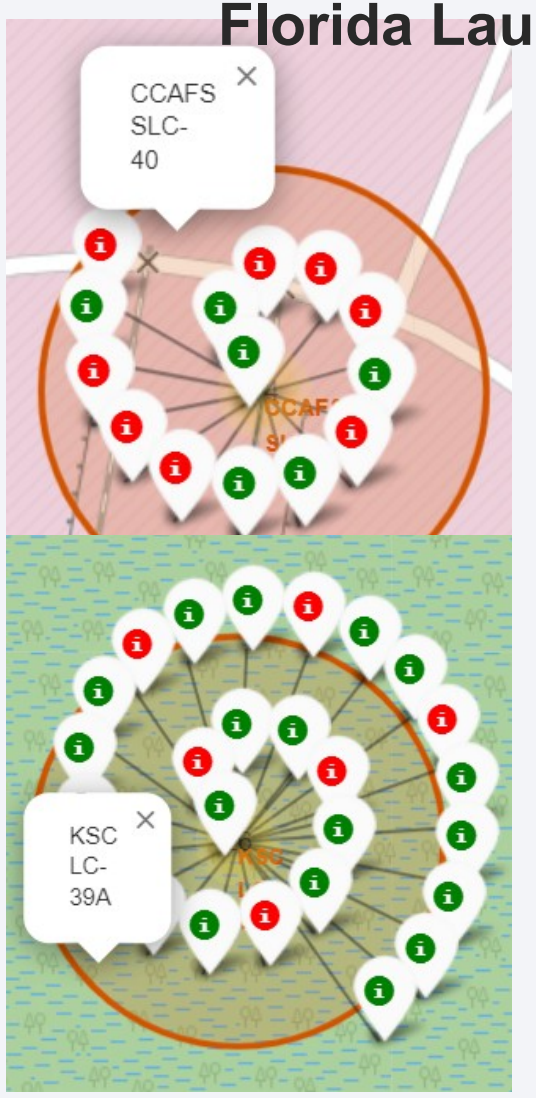
Launch Sites Proximities Analysis

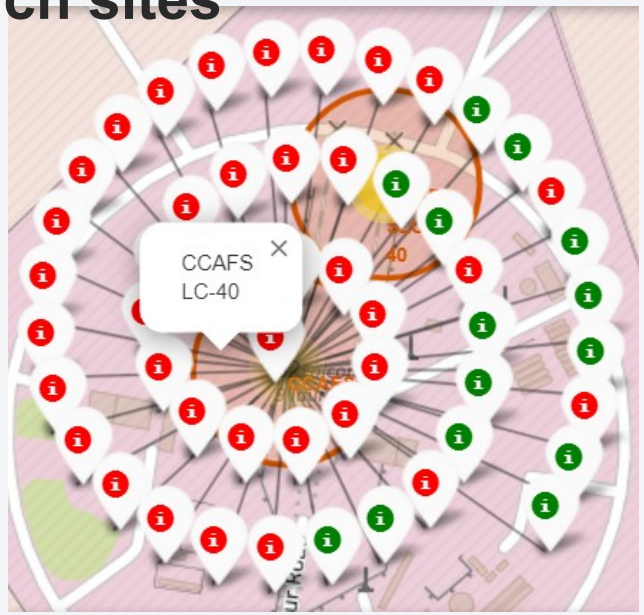
Global Map of All Launch Sites

The map shows that all SpaceX launch sites are located within the United States, with 3 sites situated in Florida and 1 in California .

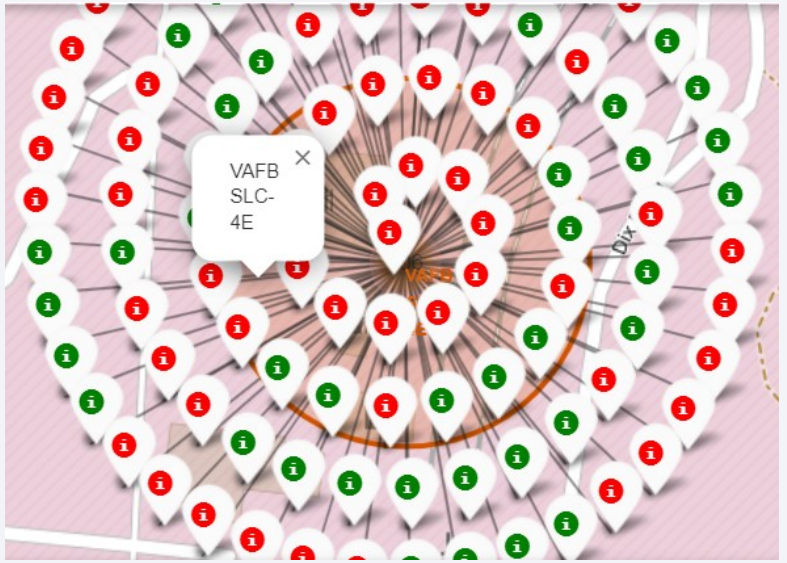
Mapping Success and Failure of Launches by Site

Florida Launch sites



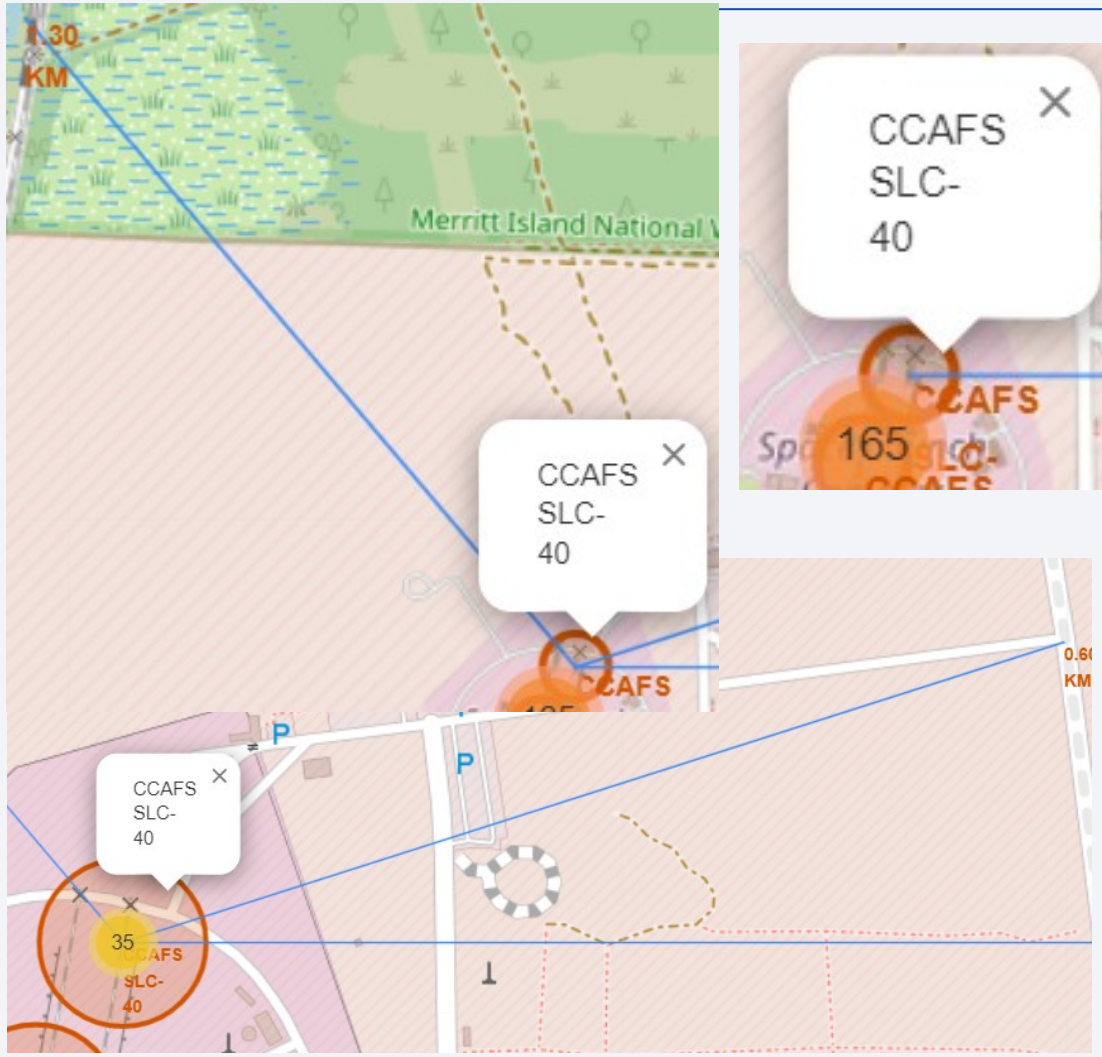


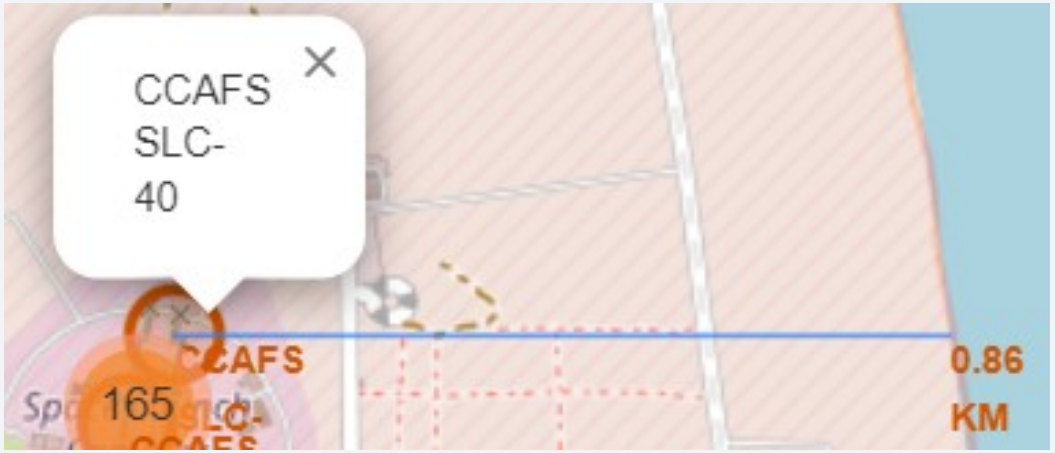
California Launch Sites



Green markers are used for successful launches,
while **red** markers represent failed launches,
visually distinguishing the outcomes across different sites.

Distances from Launch Sites to Nearby Locations

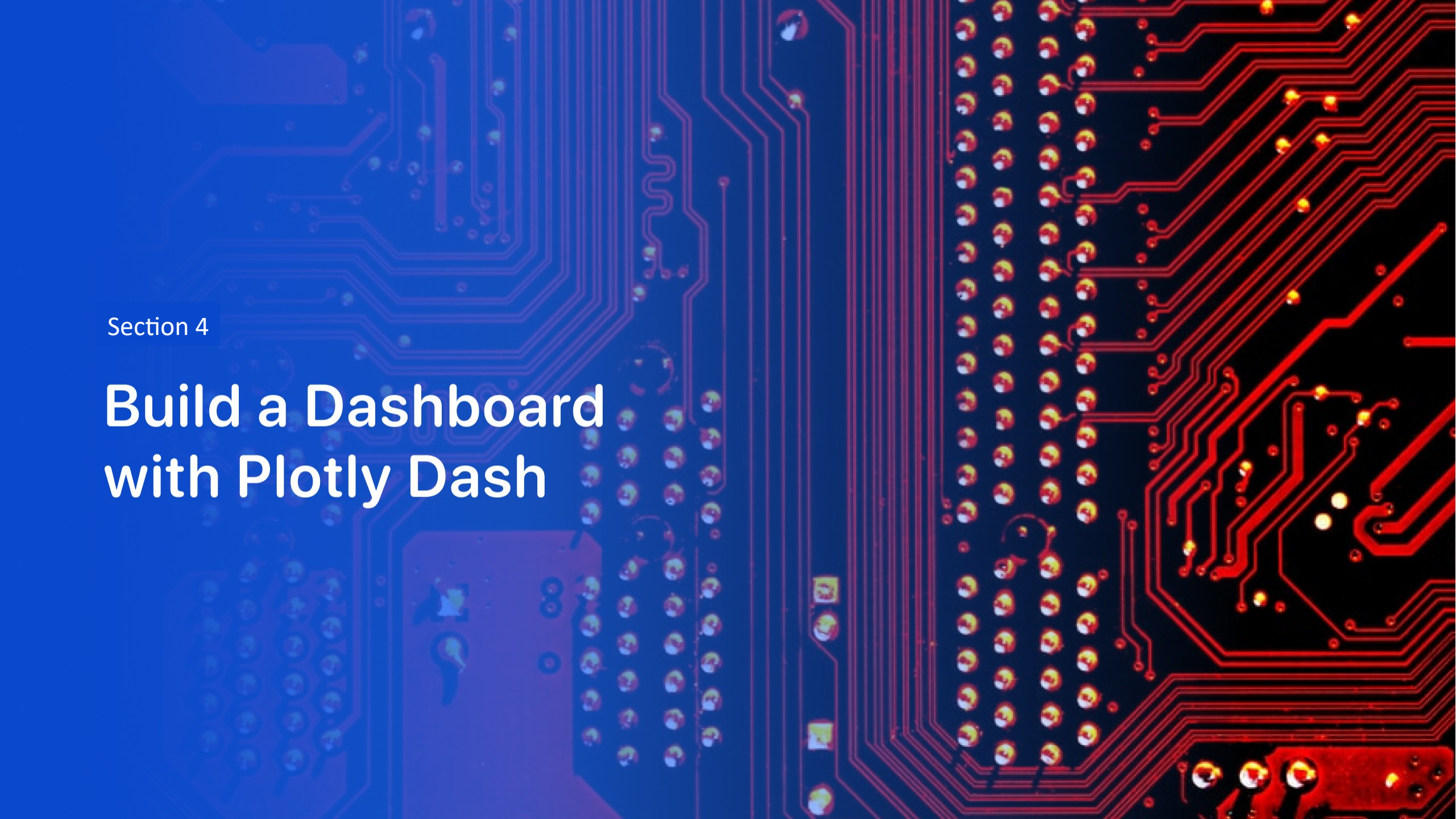




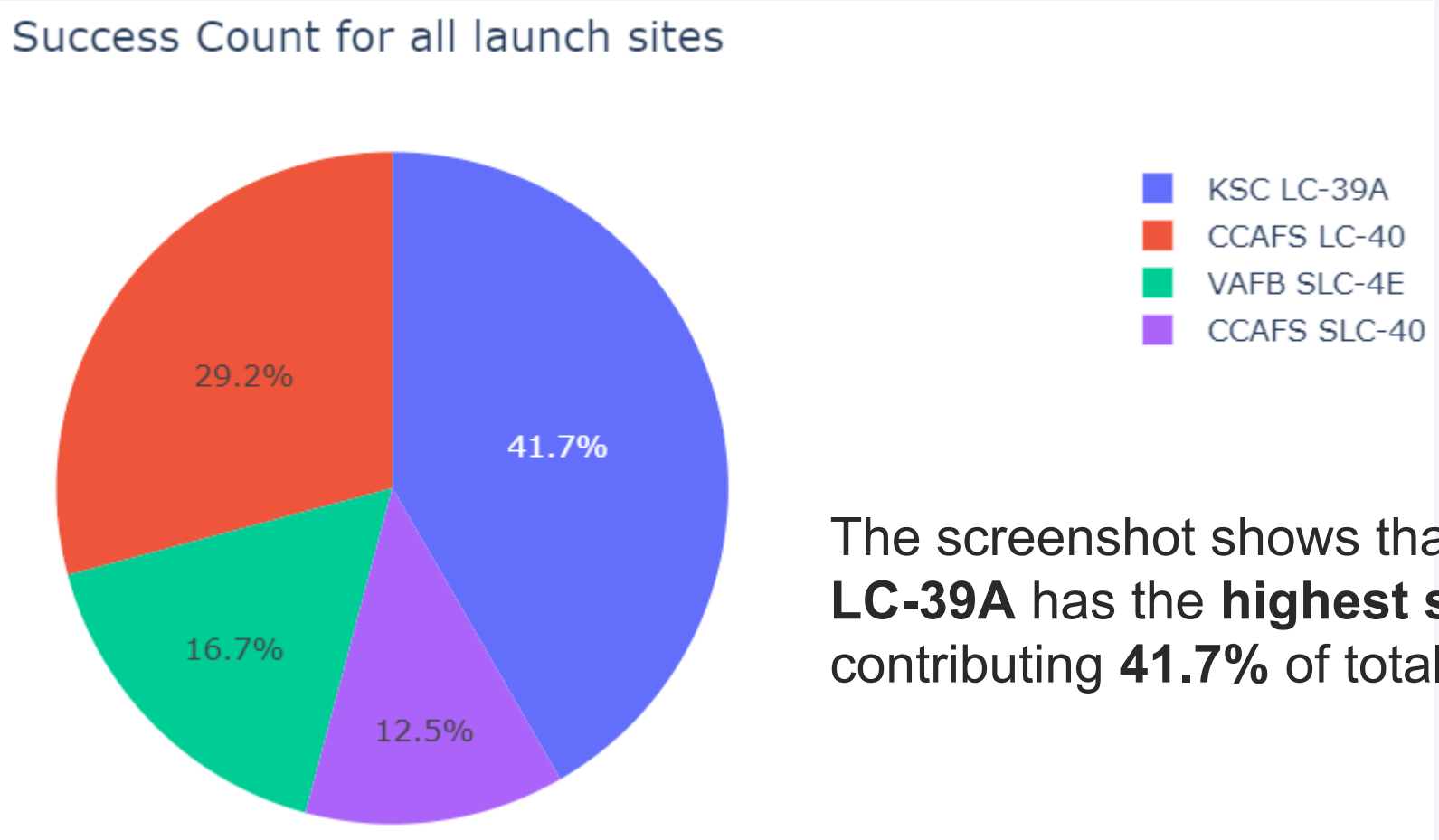
The analysis reveals the following distances from the CCAFS SLC-40 launch site to key proximities:

- The closest coastline is 0.86 km away.
- The nearest highway, Samuel C Phillips Parkway, is located 0.60 km from the site.
- The nearest railroad, NASA Railroad, is 1.30 km away.
- The nearest city, Melbourne, is situated 53.43 km from the launch site.

These distances highlight the proximity of the launch site to critical infrastructure, with the coastline and highway being relatively close, while the nearest city is significantly farther.

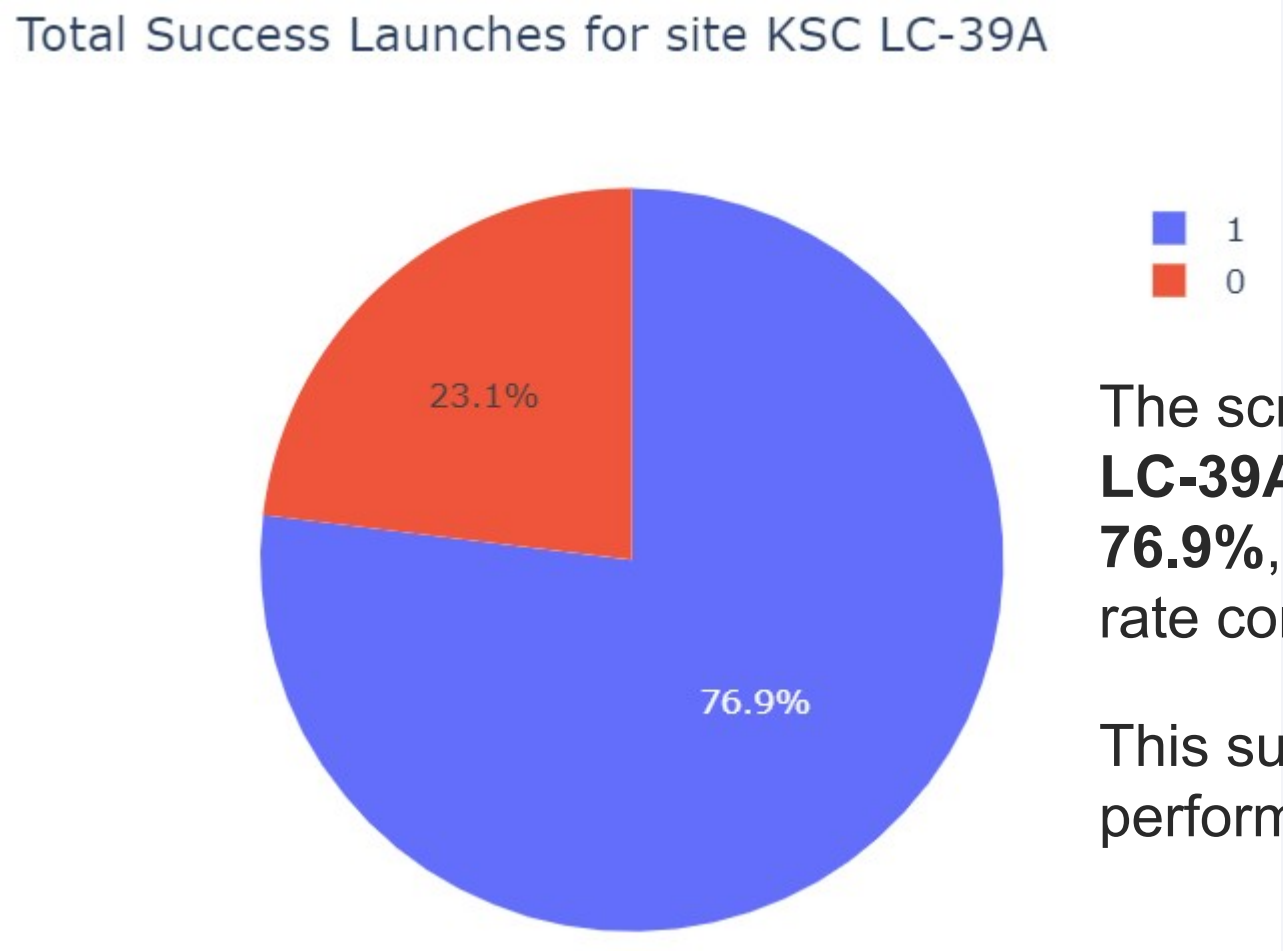

Section 4
Build a Dashboard with Plotly Dash

Launch Site Success Percentages



The screenshot shows that launch site **KSC LC-39A** has the **highest success rate**, contributing **41.7%** of total successful launches.

Top Launch Site by Success Rate

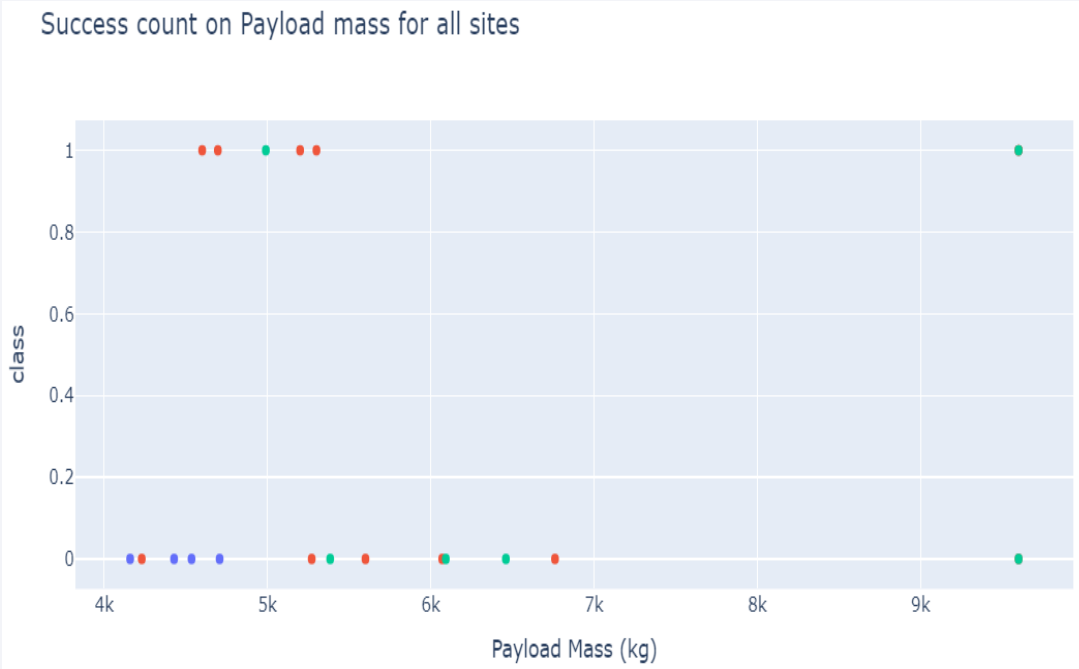


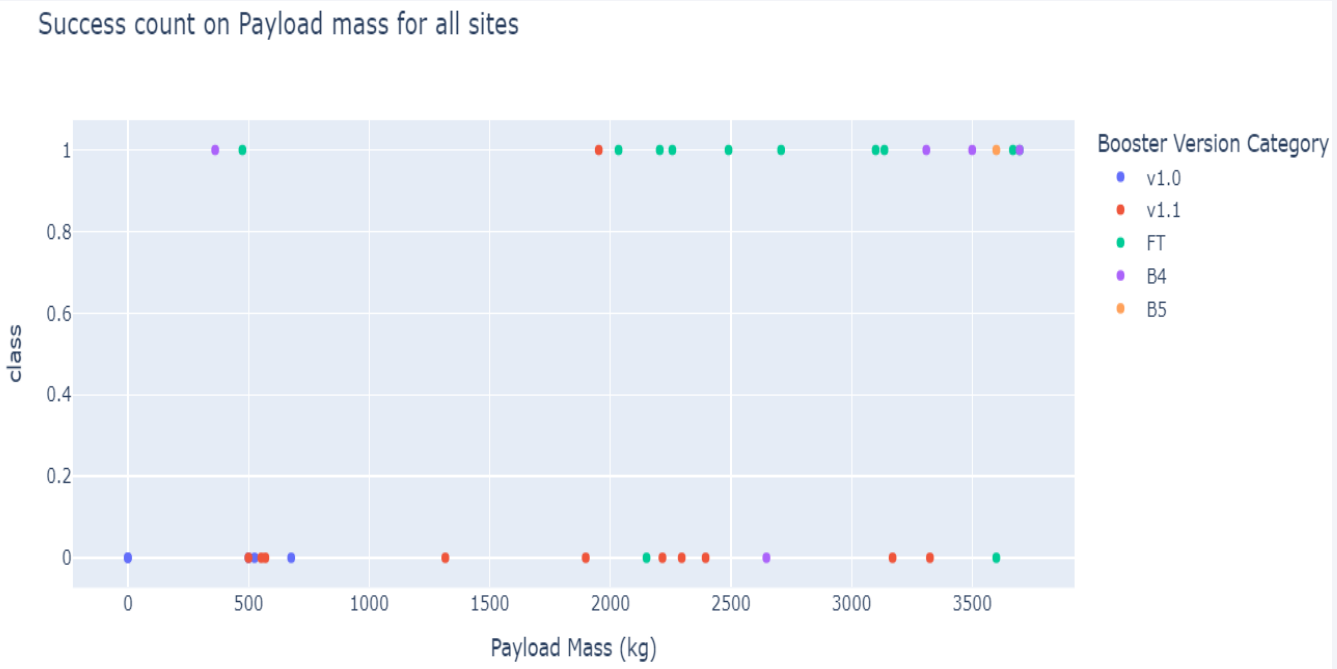
The screenshot shows that launch site **KSC LC-39A** has a **launch success rate of 76.9%**, indicating a relatively high success rate compared to other sites.

This suggests that KSC LC-39A has performed well in terms of mission success.

Success Rate by Booster Payload for F9 Versions

Low-weight F9 boosters show higher success rates compared to heavier ones, with the FT booster leading the pack.





Section 5

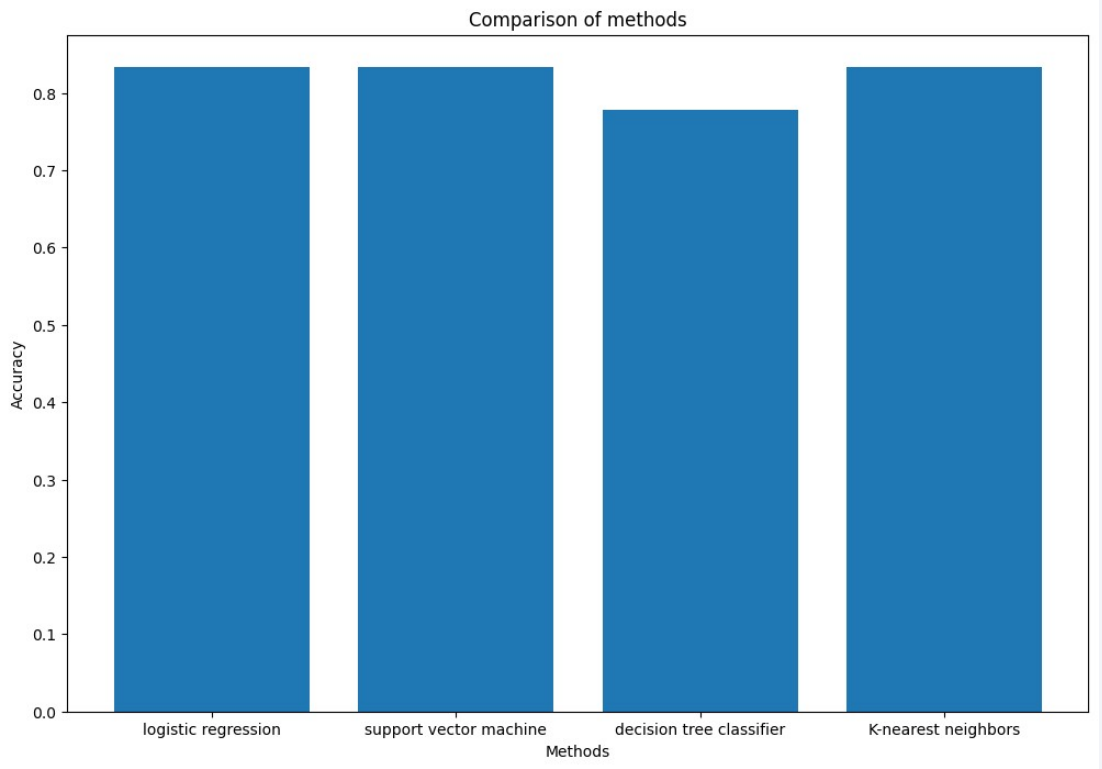
Predictive Analysis (Classification)

Classification Accuracy

Although KNN, logistic regression, and Support Vector Machine (SVM) all yield the same accuracy of 0.833, KNN demonstrates slightly higher training accuracy compared to SVM and Logic Regression.

On the other hand, the decision tree, despite having a high training accuracy of 0.89, shows the lowest test accuracy of the four models, at 0.77.

This suggests that the decision tree may be overfitting the training data, leading to a lower generalization performance on the test set.



Confusion Matrix

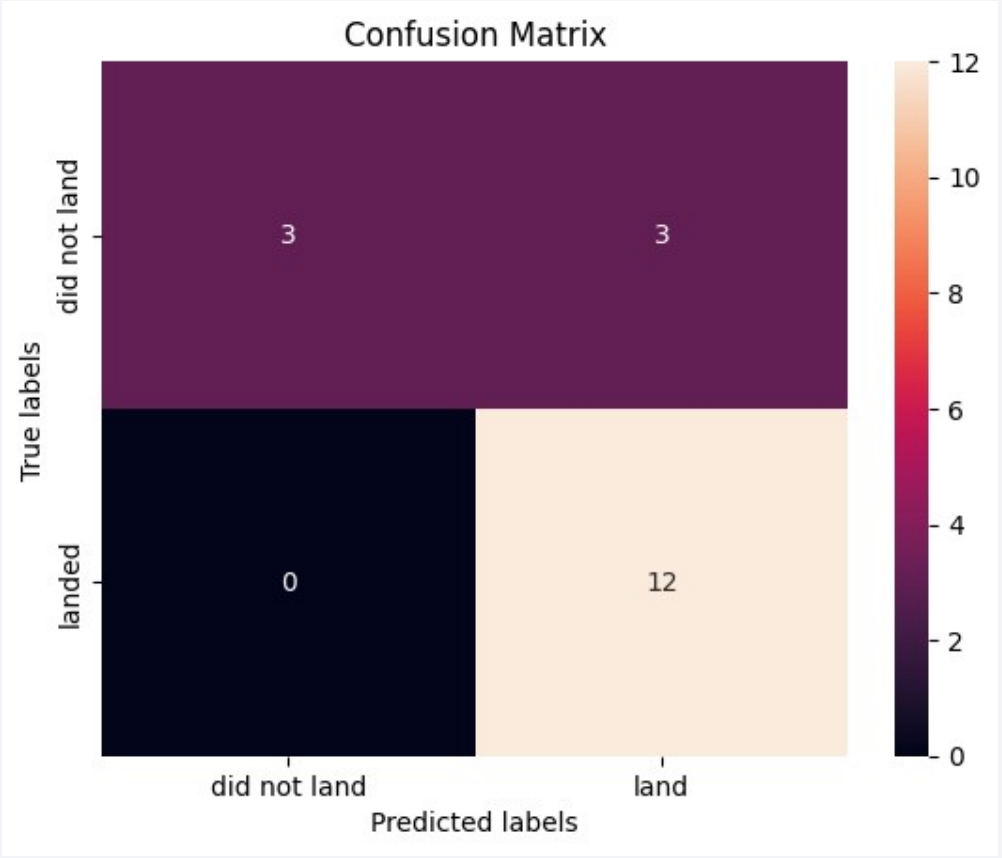
Examining the confusion matrix, we observe that the KNN model is generally effective at distinguishing between the different classes, though it faces issues with false positives.

Overview:

- **True Positive (TP)**: 12 (The true label is "landed," and the model correctly predicts "landed.")
- **False Positive (FP)**: 3 (The true label is not "landed," but the model incorrectly predicts "landed.")

The primary issue here is the occurrence of false positives, where the model mistakenly classifies instances as "landed" when they are not.

This suggests the model may be slightly over-predicting the "landed" class.



Conclusions

In conclusion, the analysis indicates that:

- A higher number of flights at a launch site correlates with an increased success rate.
- The launch success rate steadily improved from 2013 to 2020.
- Orbits such as ES-L1, GEO, HEO, SSO, and LEO exhibited the highest success rates.
- KSC LC-39A emerged as the site with the most successful launches.
- Among the machine learning models tested, the KNN classifier proved to be the most effective for predicting launch success.
- Low weighted F9 FT booster has greater launch success.

These findings provide valuable insights into the factors influencing launch success and the performance of different sites and orbits.

Appendix

All code, visuals, and notebooks from this analy

GitHub- SpaceX Data Science Project

This analysis was completed as part of the IBM Data Science Professional Certificate by Gregory Charles.

IBM - Data Science Professional Certificate

Thank you!