IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ahmed Safwat Mohamed Taleb>
<7_30_2025>

Outline

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

Executive Summary

This project successfully demonstrates a complete data science pipeline applied to predict the success of SpaceX launches. It began with collecting and preparing launch data from reliable sources such as the SpaceX API and Launch Library. Through thorough data wrangling and exploratory data analysis, key factors influencing launch outcomes were identified and visualized. Several machine learning models—including Logistic Regression, Decision Trees, SVM, and KNN—were implemented and evaluated for accuracy, with hyperparameter tuning performed using GridSearchCV. The most effective model was deployed in a user-friendly web application built with Plotly Dash, allowing interactive exploration and prediction of launch success. Overall, the project showcases how machine learning can be practically applied to support decision-making in the aerospace industry.

Introduction

- Project background and context:
  The commercial space age has arrived, with private companies like SpaceX, Virgin Galactic, and Blue Origin pioneering space access. SpaceX stands out for its cost-effective Falcon 9 rocket launches, significantly reducing launch expenses by reusing the first stage of its rockets—a capability most competitors lack.
  In this project, as a data scientists at SpaceY, a new aerospace startup aiming to compete with SpaceX. Our mission is to explore the feasibility of predicting the reuse of Falcon 9's first stage using machine learning and available data.

- Problems you want to find answers:

  Can we predict whether the Falcon 9 first stage will land successfully?

  How does first stage recovery impact launch cost estimation?

  What launch factors influence the success or failure of first stage landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Used the SpaceX public API to retrieve launch records, including payload mass, orbit, launch site, booster version, and landing outcomes.
- Web scraping was performed from Wikipedia to complement the dataset with launch site details.

Data Collection – SpaceX API

- https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/jupyter_labs_spacex_data_collection_api_v2.ipynb

```
[Start]
   |
   v
[Install Required Libraries]
   |
   v
[Import Libraries and Set Pandas Options]
   |
   v
[Fetch Launch Data from SpaceX API]
   |
   v
[Define Functions to Collect Extra Details]
   |
   v
[Use API to Get Rocket, Launchpad, and Payload Info]
   |
   v
[Clean and Format Data into a DataFrame]
   |
   v
[Save or Display Final Dataset]
   |
   v
[End]
```

Data Collection - Scraping

- https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/jupyter_labs_webscraping.ipynb

```
[Start]
    |
    v
[Install Required Libraries]
    |
    v
[Import Libraries]
    |
    v
[Define Helper Functions for Data Extraction]
    |
    v
[Request Web Page Using requests.get()]
    |
    v
[Parse HTML with BeautifulSoup]
    |
    v
[Extract and Clean Table Data]
    |
    v
[Store Data in Pandas DataFrame]
    |
    v
[Save or Display Final Data]
    |
    v
[End]
```

Data Wrangling

- Describe how data were processed:
  - Cleaned and merged datasets, removed missing values, and standardized categorical variables.
  - Created new features such as binary landing success and one-hot encoding for categorical fields.
- https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/labs_jupyter_spacex_Data_wrangling_v2.ipynb

```
[Start]
   |
   v
[Import pandas, numpy]
   |
   v
[Load CSV from URL into DataFrame]
   |
   v
[Check for Missing Values]
   |
   v
[Inspect Column Data Types]
   |
   v
[Drop Irrelevant or Null Columns]
   |
   v
[Filter Valid Rows Only]
   |
   v
[Create New 'Class' Column (1=Success, 0=Failure)]
   |
   v
[Preview and Confirm Cleaned Data]
   |
   v
[End]
```

EDA with Data Visualization

Purpose	Relationship	Chart Type
Identify which launch sites contribute most to .successful landings	Launch Site vs. Success Count	**Bar Chart**
Explore if payload size & orbit type affect landing .success	Payload Mass vs. Success (per Orbit)	**Scatter Plot**
Analyze how success rates evolved over time by orbit .type	Flight Number vs. Success (per Orbit)	**Scatter Plot**
Identify which orbits are more challenging for .successful landings	Orbit Type vs. Success Rate	**Bar Chart**
.Assess if launch sites improved success over time	Flight Number vs. Success (per Site)	**Scatter Plot**
.Visualize overall landing success vs. failure distribution	Class (0/1) vs. Count	**Pie/Bar Chart**
.Observe payload trends and variations over time	Flight Number vs. Payload Mass	**Box Plot**
.Identify most frequently used booster versions	Booster Version vs. Count	**Histogram**

https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/jupyter_labs_eda_dataviz_v2.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - List the total number of successful and failure mission outcomes
  - List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
  - 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.
  - 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.
- https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/jupyter_labs_eda_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

Element Added to the Map	Purpose / Importance
`folium.Circle`	To visually highlight the exact launch site location using a circular marker
`folium.Marker` with `DivIcon`	To label each launch site directly on the map with its name
`folium.MarkerCluster`	To group and display individual launch results (success/failure) at each launch site
`folium.Marker` (Success/Failure)	To indicate each launch result with color (green = success, red = failure)
`folium.Marker` (Nearby Landmarks)	To mark nearby features such as city, railway, highway, or coastline
`folium.DivIcon` for Distance	To show the calculated distance between the launch site and the nearby landmark
`folium.PolyLine`	To draw lines connecting the launch site to nearby landmarks to show spatial relations
`MousePosition` Plugin	To display real-time coordinates as the mouse moves over the map

https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/lab_jupyter_launch_site_location_v2.ipynb

Build a Dashboard with Plotly Dash

Component	Description	Purpose / Importance
Dropdown Menu (`dcc.Dropdown`)	Allows user to select a specific launch site or view data from all sites.	Enables dynamic filtering of the data to explore site-specific success rates.
Pie Chart (`success-pie-chart`)	Updates based on the selected site. Shows: ▪ For "All Sites": Total successful launches by site. ▪ For specific site: Success vs. Failure count.	Helps visualize the performance of each site or how successful a specific site was.
Payload Range Slider (`dcc.RangeSlider`)	Lets user choose a payload mass range (min to max).	Enables filtering to analyze how launch success relates to payload mass.
Scatter Plot (`success-payload-scatter-chart`)	Updates based on selected launch site and payload range. Shows: Payload Mass vs. Launch Outcome, colored by Booster Version.	Helps identify trends between payload and success/failure and compare booster types.

- https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/spacex_dash_app.py

Predictive Analysis (Classification)

Action
Data Loading & Exploration
Feature Engineering
Data Standardization
Train-Test Split
Model Training with GridSearchCV
Model Evaluation (Accuracy & Confusion Matrix)
Model Comparison & Selection

```
           |  Load & Explore Data  |
           +-----------+-----------+
                       v
        +-----------------------------+
        |  Extract Features (X, Y)    |
        +-----------------------------+
                       v
        +-----------------------------+
        |  Standardize Features       |
        |  (StandardScaler)           |
        +-----------------------------+
                       v
        +-----------------------------+
        |  Train-Test Split (80/20)   |
        +-----------------------------+
                       v
   +-----------------+------------------+-------------------+-----------------+
   |                 |                  |                   |                 |
   v                 v                  v                   v                 v
Logistic          SVM (GridCV)    Decision Tree (CV)    KNN (GridCV)      ... (more)
Regression        + tuning        + tuning              + tuning
+ tuning          parameters      parameters            parameters
                       v
        +------------------------------------------+
        |  Evaluate: Accuracy & Confusion Matrix   |
        +------------------------------------------+
                       v
        +-----------------------------+
        | Compare Models & Select     |
        | Best Performing Classifier|
        +-----------------------------+
```

https://github.com/AhmedSafwatMohamed/IBM_Project/blob/main/scripts-and-notebooks/SpaceX_Machine_Learning_Prediction_Part_5_v1.ipynb

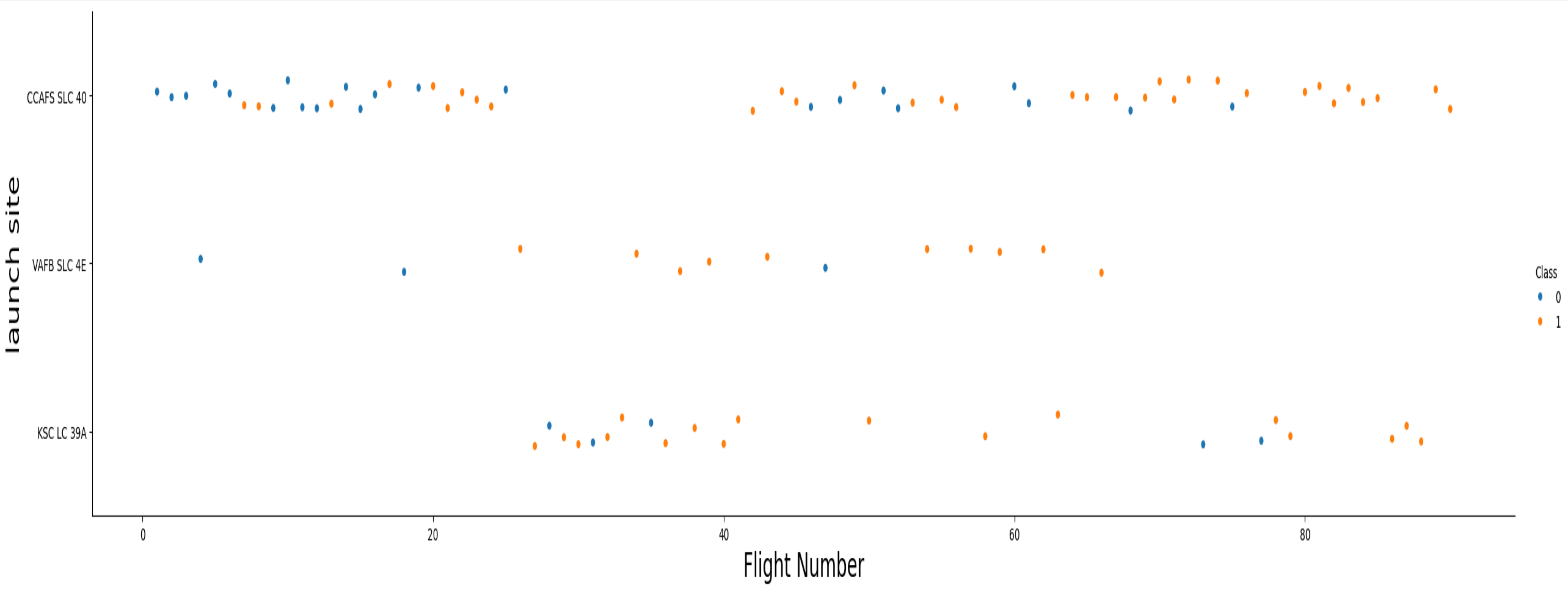
Results

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

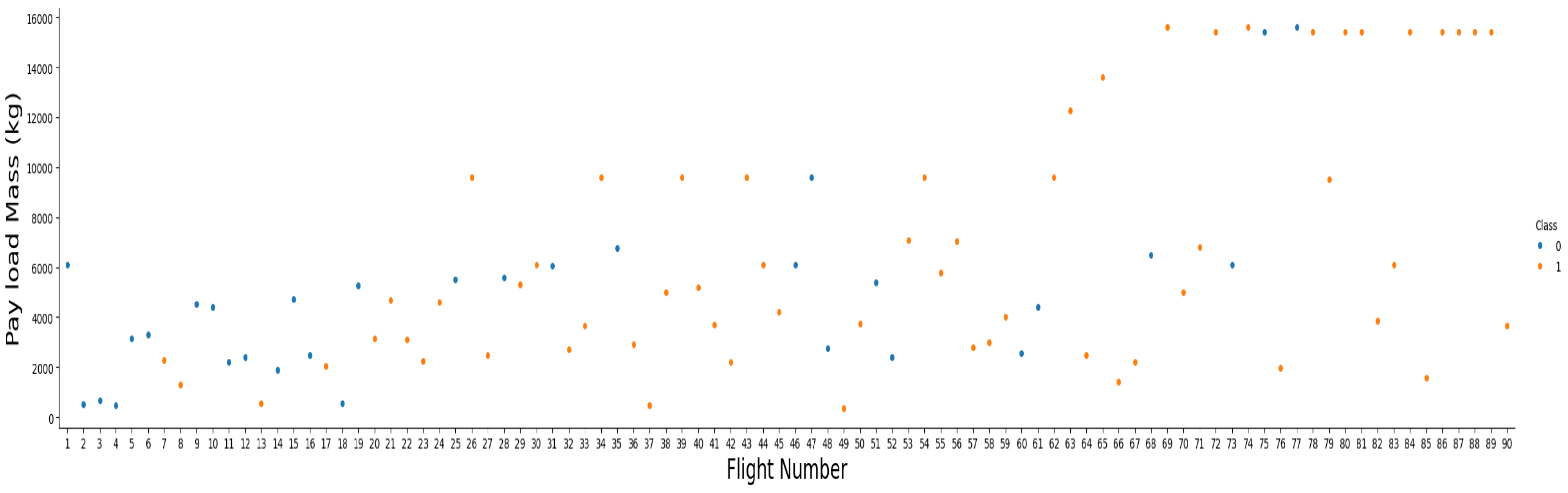
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



Payload vs. Launch Site



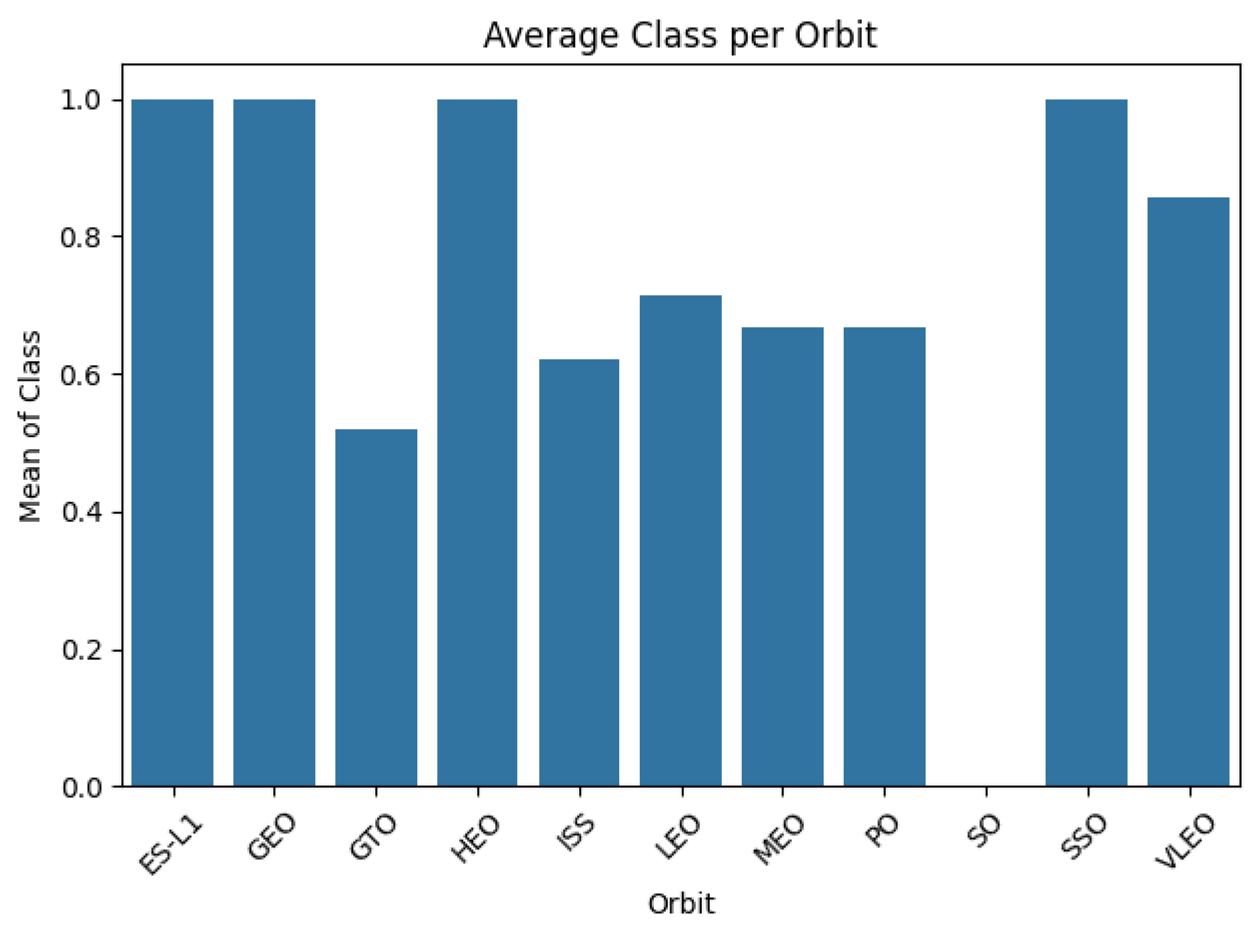
We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

Success Rate vs. Orbit Type

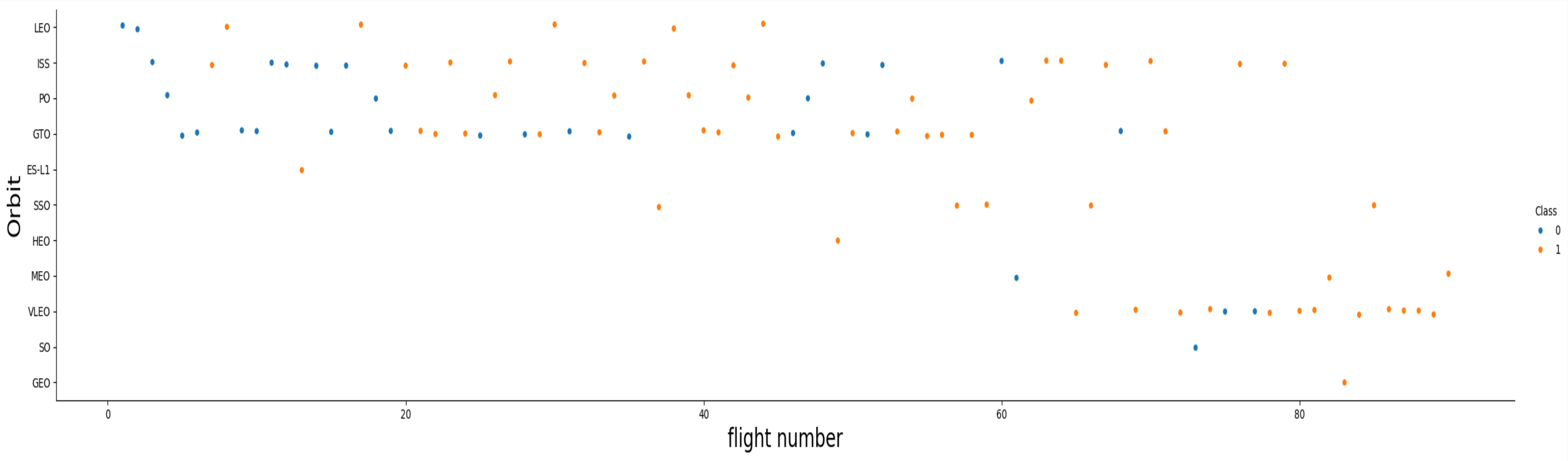
It seems like :

ES-11
GEO
HEO
SSO

Have higher success rate than at other orbit types.

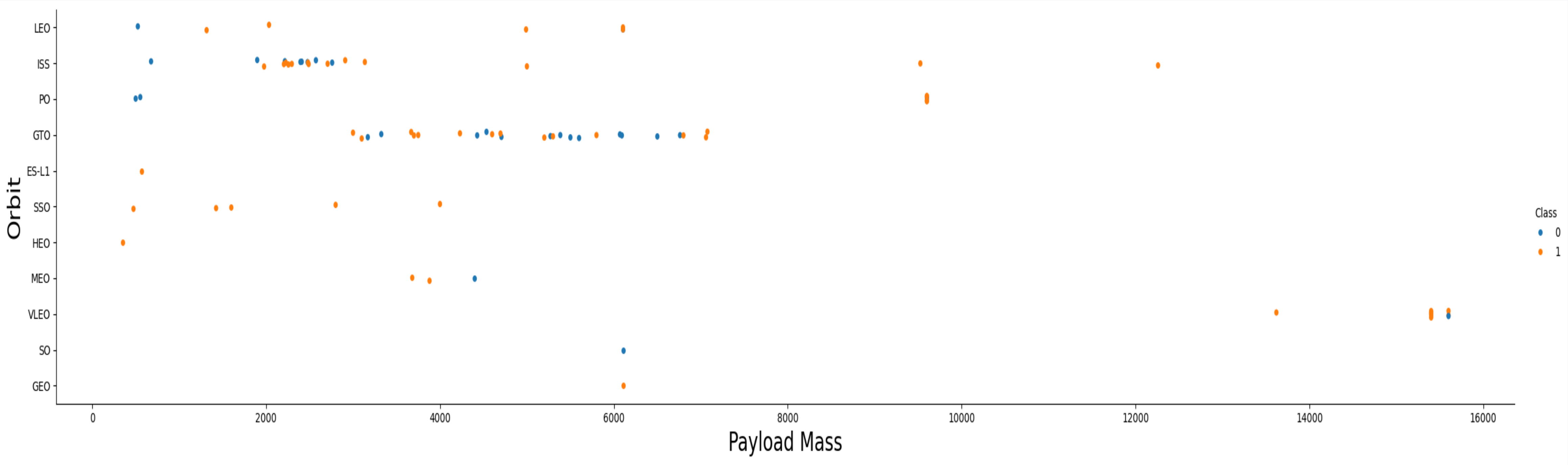


Flight Number vs. Orbit Type



in the LEO orbit we see the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



It seems With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

All Launch Site Names

It is a menu contains the launch sites appear in dataset which are for sites

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

Launch Site Names Begin 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

Between 2010–2013, SpaceX launched five Falcon 9 missions from CCAFS LC-40. All launches succeeded, but early booster recoveries failed. Most payloads were NASA missions to the ISS with small to medium mass (0–677 kg) targeting LEO.

Total Payload Mass

This value represents the combined payload mass of all recorded SpaceX missions in the dataset. It reflects the total weight of cargo delivered to various orbits, mainly in support of missions like ISS resupply and satellite deployments.

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

This indicates that, on average, each SpaceX mission carried approximately **2.9 metric tons** of payload into space, based on the data analyzed.

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

Tish date marks **SpaceX's first successful rocket landing**, a major milestone in spaceflight history. It demonstrated that Falcon 9's first stage could return and land vertically, enabling future reuse and significantly lowering the cost of space missions.

First_Successful_Landing_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

These are Falcon 9 Full Thrust (F9 FT) booster versions used in successful missions. The ".2" suffix (e.g., B1021.2) indicates that the booster was reused, demonstrating SpaceX’s capability for rocket reusability — a key goal in reducing launch costs.

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

Total Number of Successful and Failure Mission Outcomes

The majority of SpaceX missions have been successful, with 98 clearly marked as "Success" and two more labeled similarly. Only one mission failed during flight, and another had an unclear payload status despite launch success — highlighting SpaceX's strong reliability record overall.

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

Boosters Carried Maximum Payload

These boosters consistently delivered the maximum payload mass, showcasing the Falcon 9 Block 5's enhanced capacity and efficiency in handling heavy payload missions.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

These results highlight SpaceX's early challenges with booster recovery during the development of Falcon 9 v1.1, particularly with drone ship landings in the early stages of reusability testing.

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

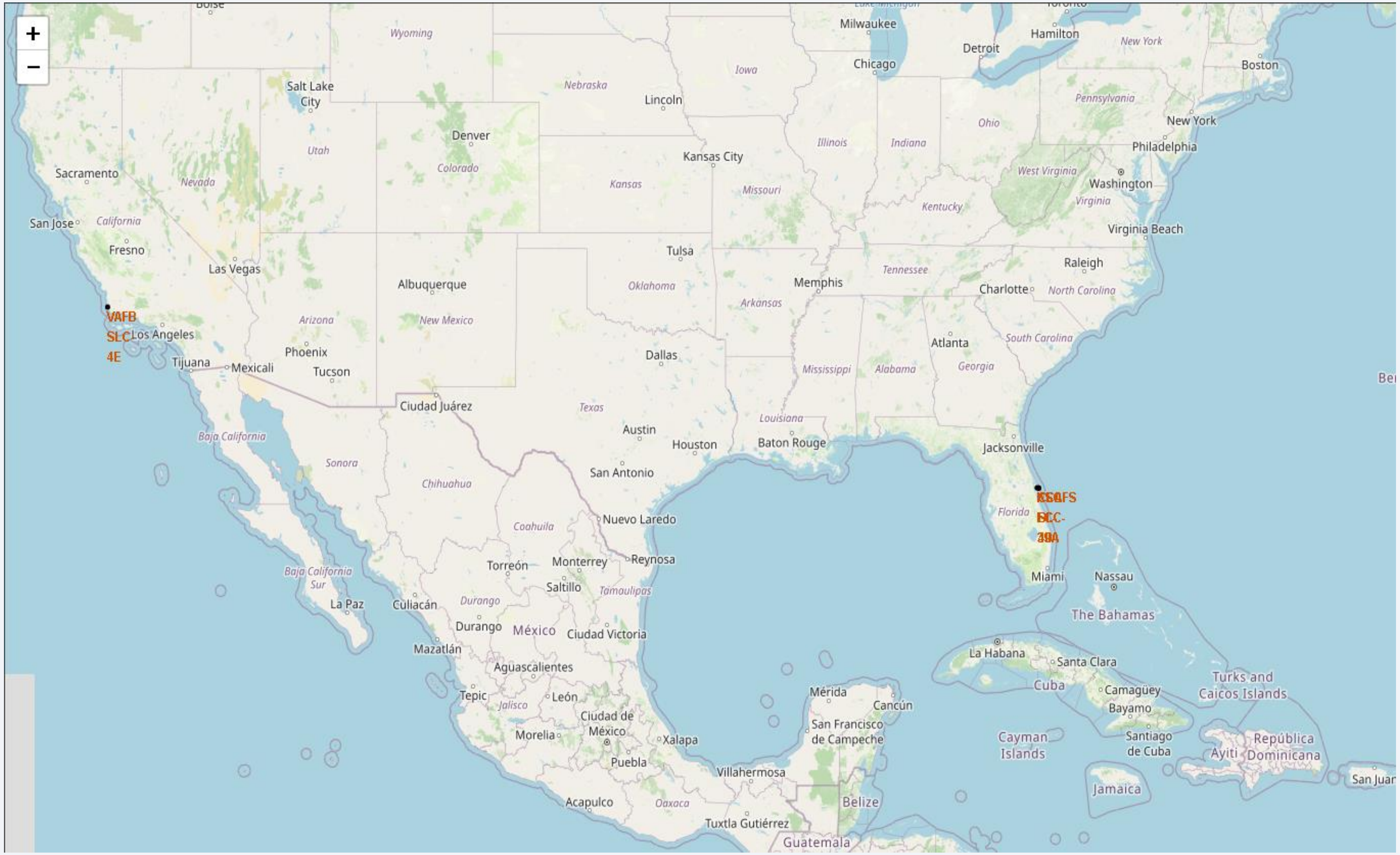
The data reflects SpaceX's experimentation with different landing strategies and their evolution over time toward more consistent success.

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

Section 3

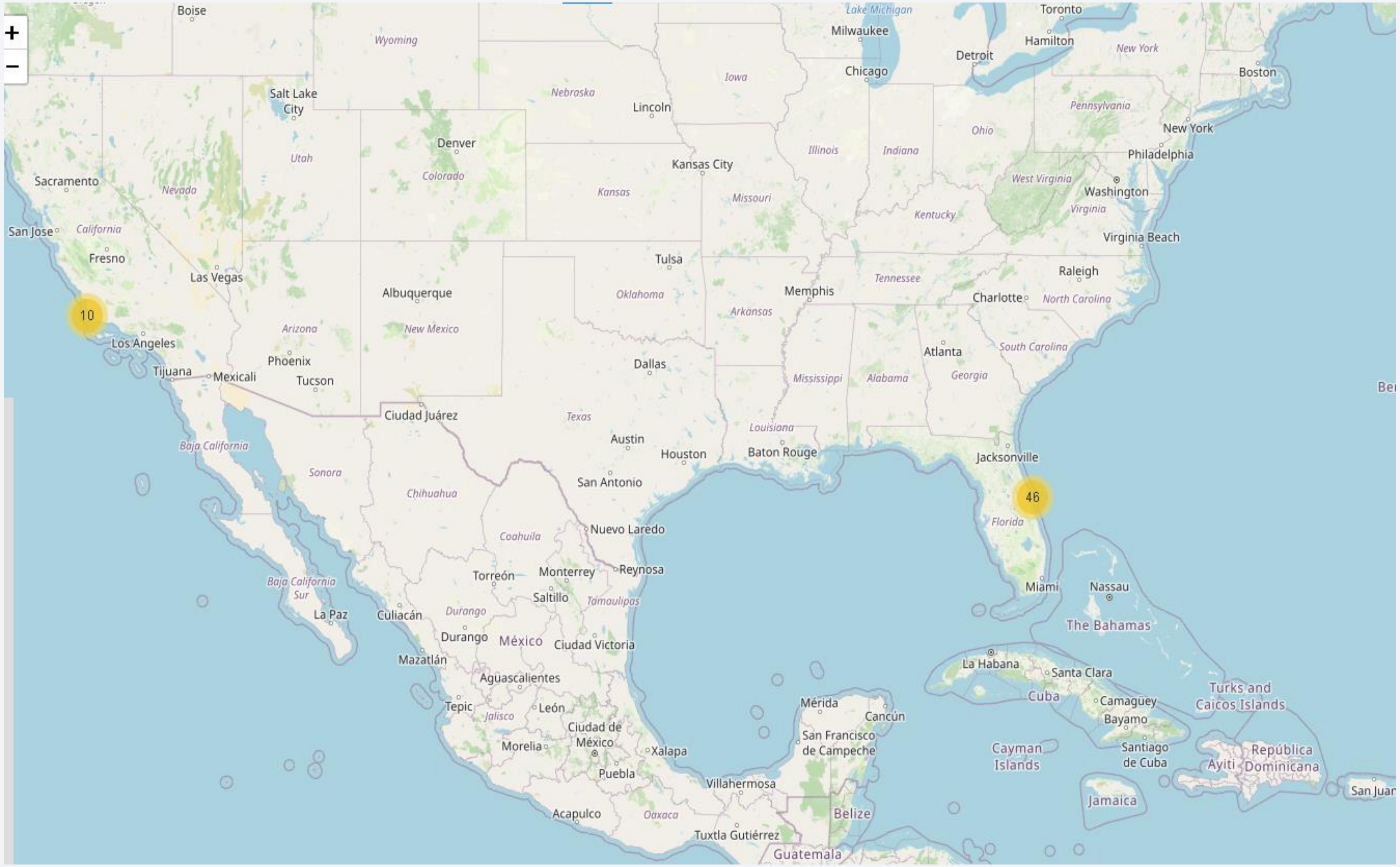
Launch Sites Proximities Analysis

Global View of All Launch Sites

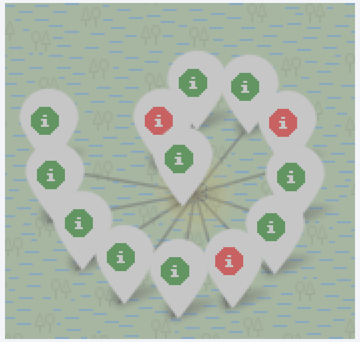


The map shows all SpaceX launch sites located near the U.S. coastlines, mainly in Florida and California. Their coastal positions reduce risk to populated areas during launches. Markers with site names help identify each location easily.

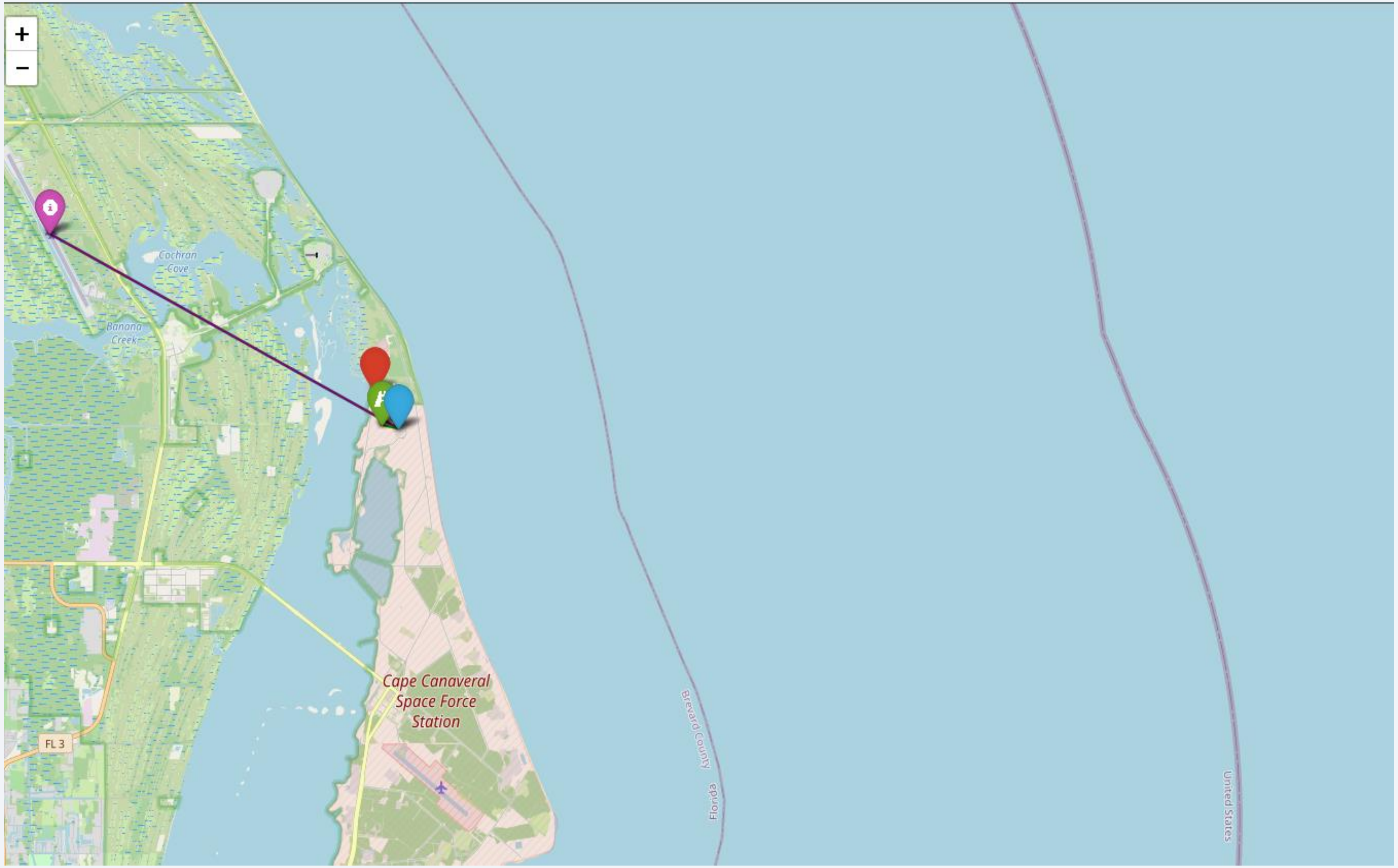
Launch Outcome Visualization by Site



Green markers show successful launches, red ones show failures. Most launches were successful, especially at certain sites, indicating higher reliability there.



Proximity of Launch Site to Key Infrastructure

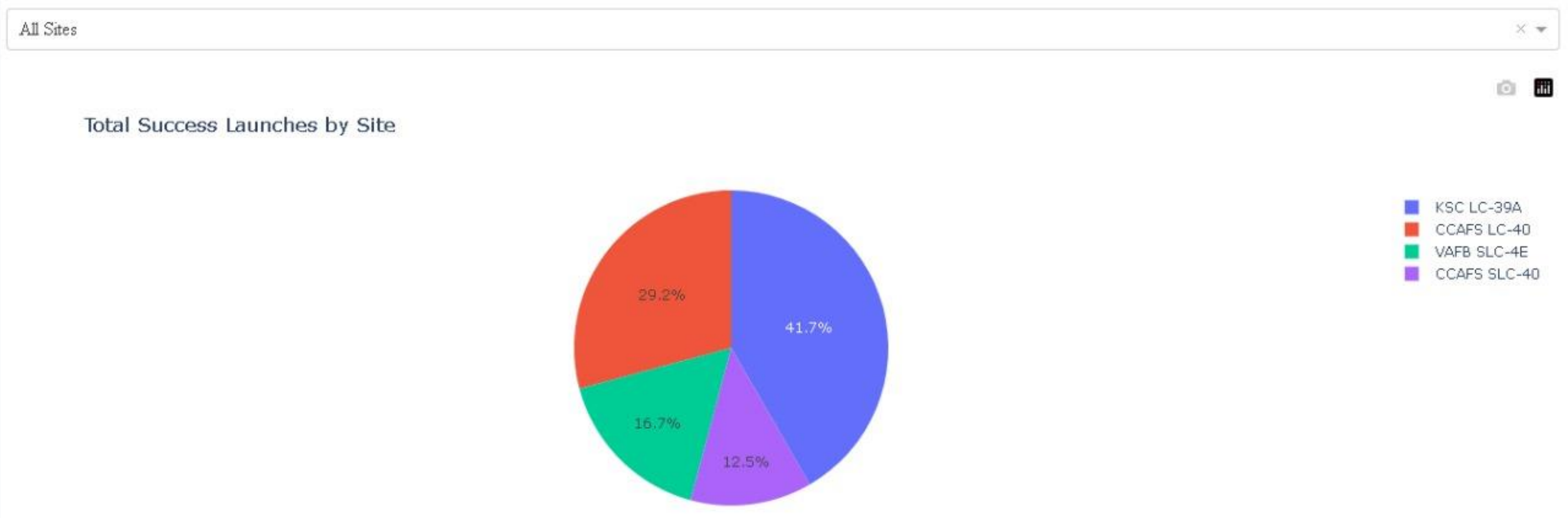


The map highlights how close the launch site is to key infrastructure, which supports efficient transport and operations.

Section 4

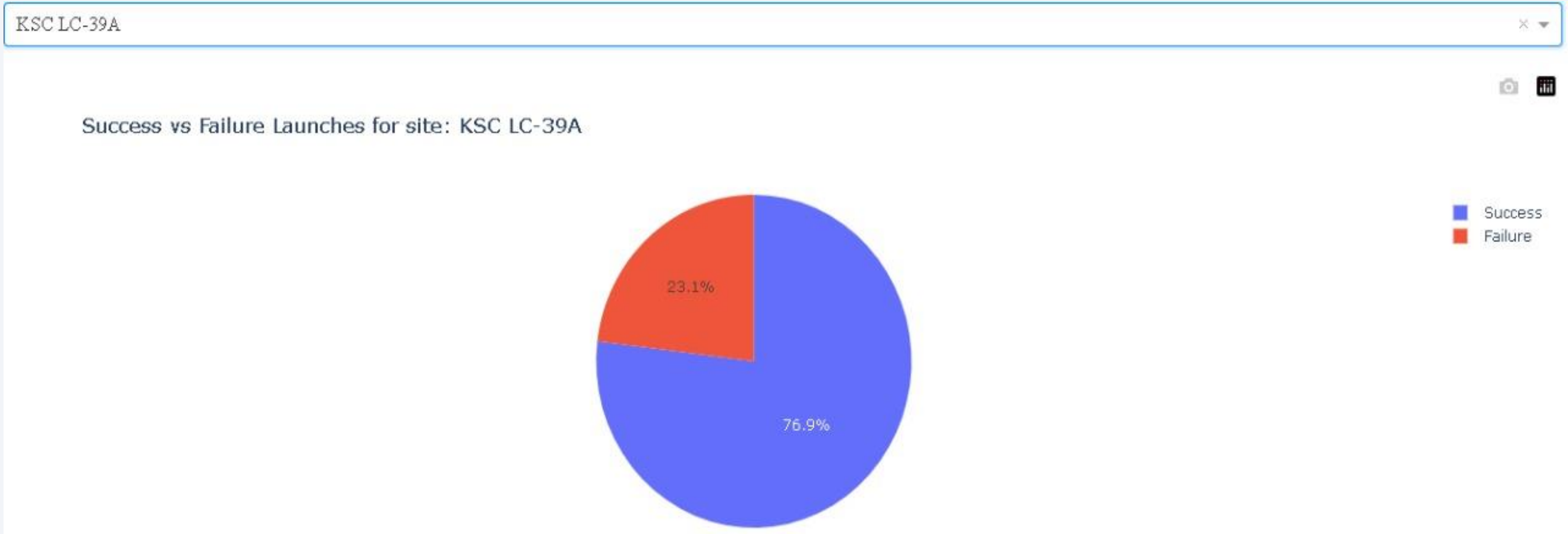
Build a Dashboard with Plotly Dash

Success Rate by Launch Site



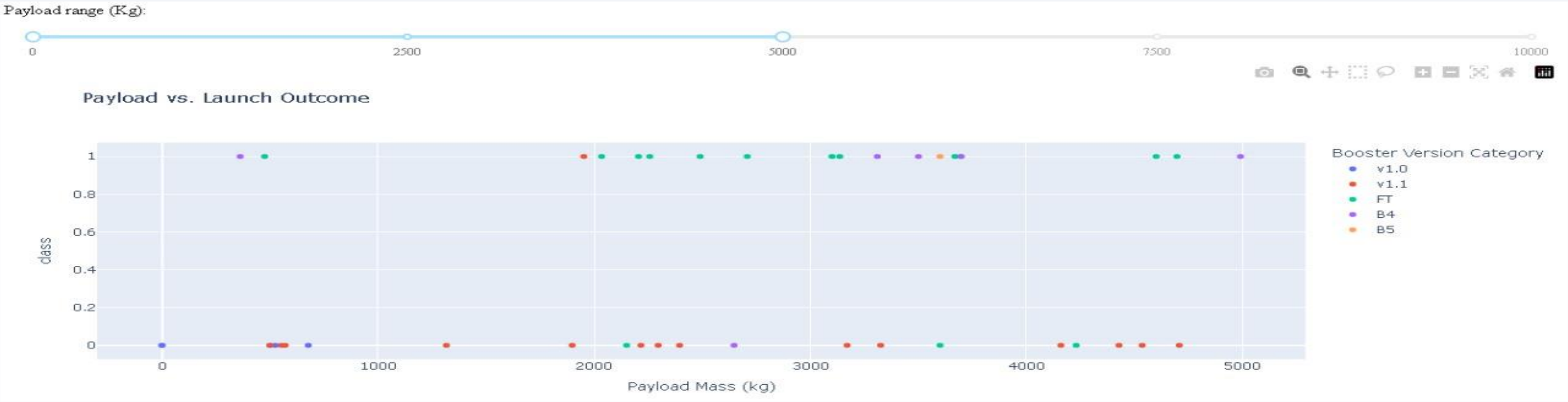
It seems that KSC LC-39A has the highest success rate 41.7% different around 12% from the next highest rate CCAFS LC-40

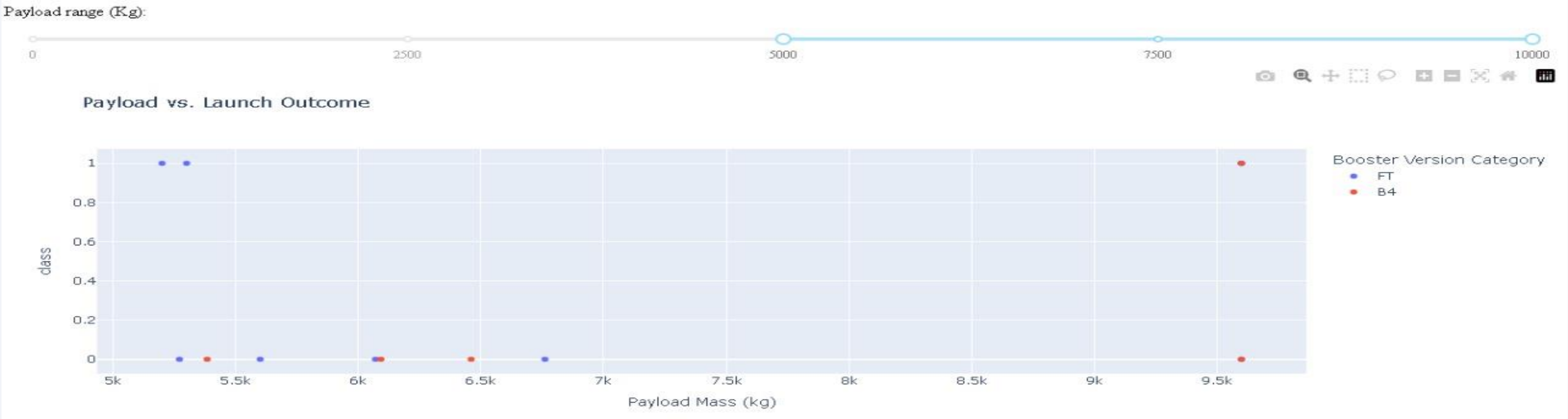
The highest Launch site success VS fail rate



KSC LC-39A has the highest success VS fail launches 76.9%

Payload Mass effect on number of success launches



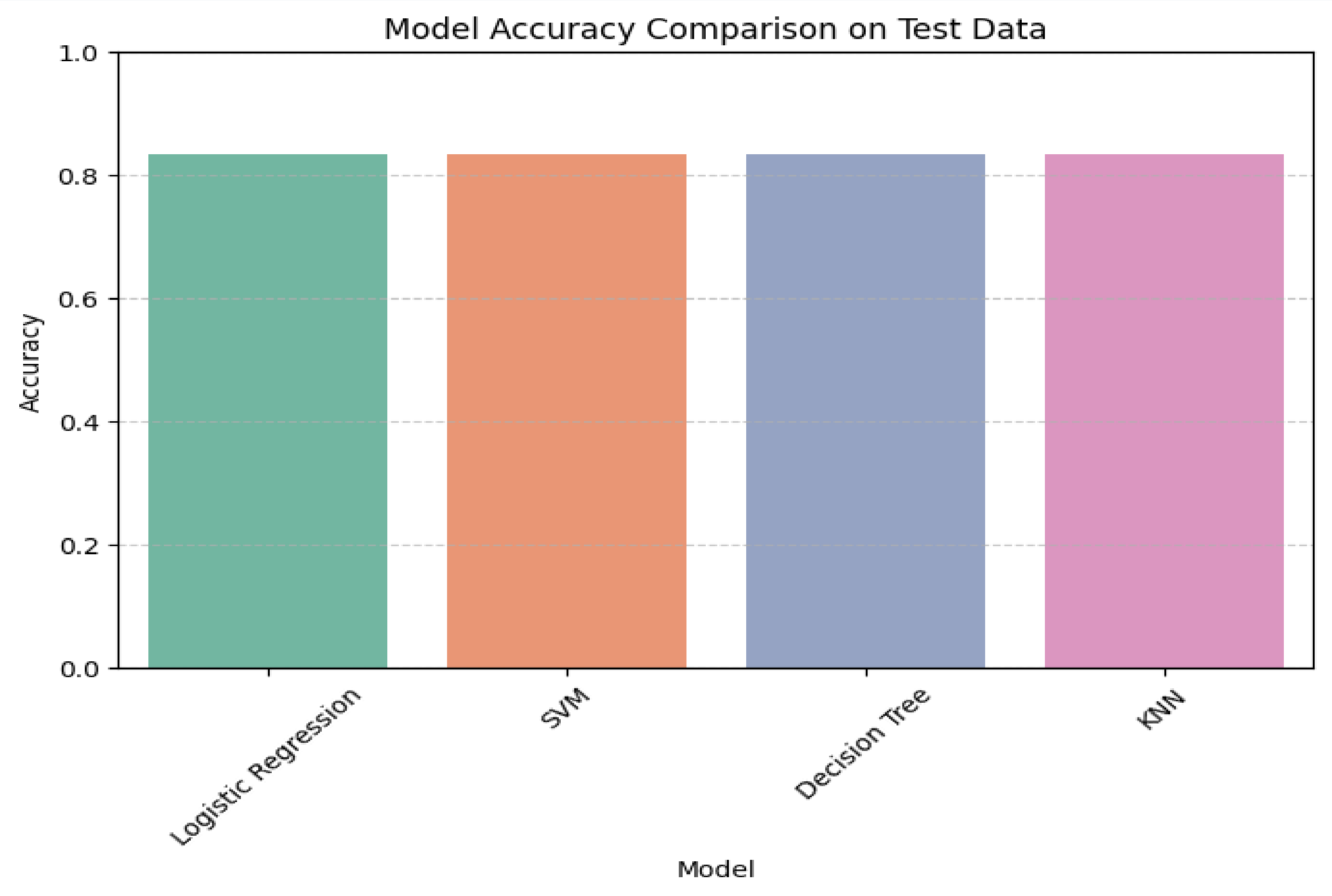


Section 5

Predictive Analysis (Classification)

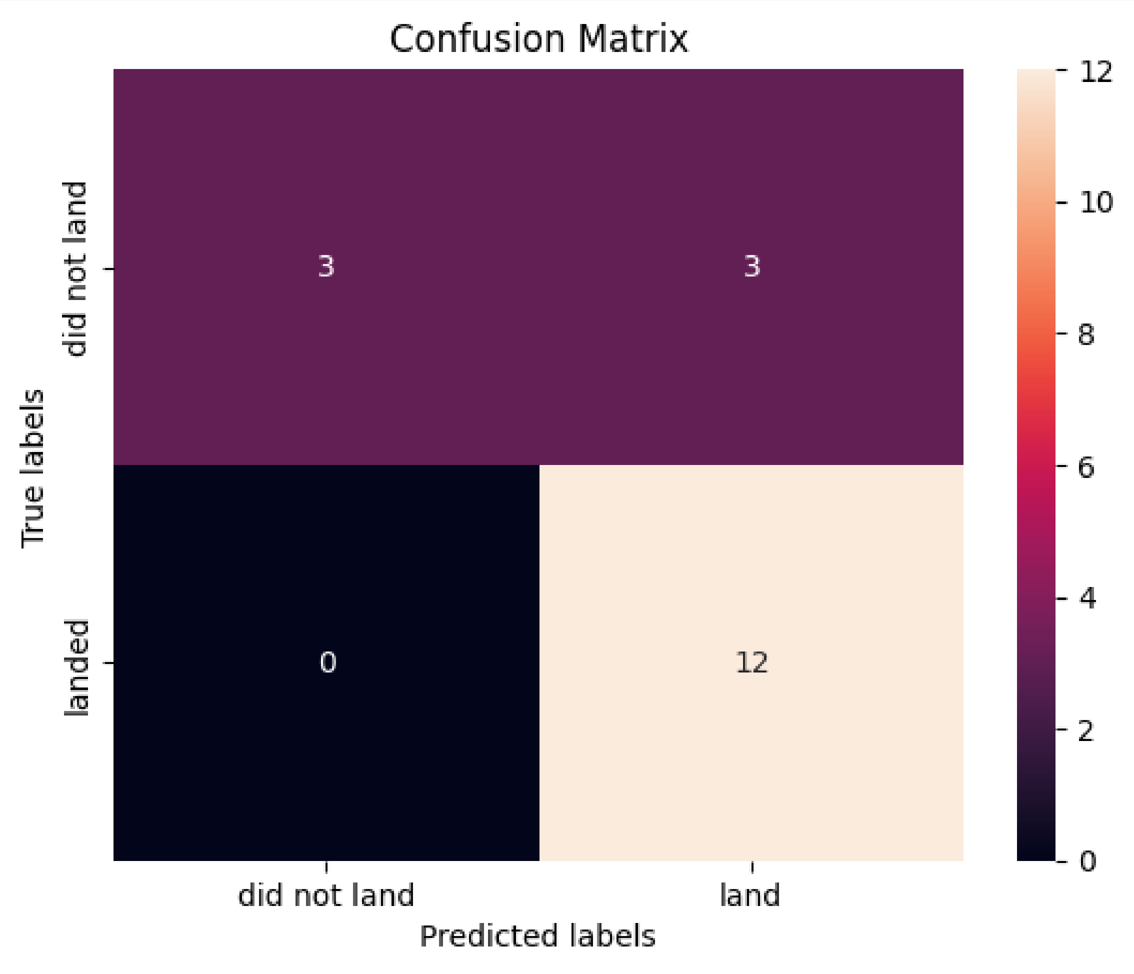
Classification Accuracy

✅ The best model is **Logistic Regression** with accuracy of **0.83** on test data.



Confusion Matrix

The confusion matrix shows the performance of the best machine learning model on the test dataset. It correctly predicted *12 landings* (true positives) and *3 non-landings* (true negatives). However, it also made *3 false positive errors—predicting that the rocket would land when it actually did not. There were **no false negatives*, meaning the model did not miss any actual landings. This indicates the model is very good at identifying successful landings, but slightly overestimates them.



Conclusions

• ✅ Successfully collected and consolidated launch data from various sources, including SpaceX API and Launch Library, ensuring a rich dataset for analysis.
• 📊 Performed comprehensive **data wrangling** using Pandas and NumPy to clean, filter, and prepare the dataset by handling missing values and irrelevant fields.
• 📍 Conducted **exploratory data analysis (EDA)** using Matplotlib, Seaborn, and Plotly to uncover patterns and visualize relationships between features like launch site, payload mass, and success rate.
• 🤖 Implemented and evaluated multiple **machine learning models**, including:

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

  to predict the success of SpaceX launches.

• 🧠 Used **GridSearchCV** to optimize hyperparameters and improve model performance, selecting the most accurate model for deployment.
The best model is **Logistic Regression** with accuracy of **0.83** on test data.
• 🖥 Developed an interactive web application using **Plotly Dash** to allow users to explore the data and view real-time predictions in an intuitive interface.

Appendix

https://github.com/AhmedSafwatMohamed/IBM_Project

Thank you!