IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tacdin OZMEN

Outline

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

Executive Summary

- Objective: Analyze factors affecting Falcon 9 first-stage landing and build a classifier to predict landing success.
- Data: Combined SpaceX API and web-scraped dataset ($N = 90$, $M = 17$).
- Key EDA Findings:
  - Landing success becomes more frequent in later flights (experience/time effect).
  - Landing success rate varies by orbit type, suggesting orbit is a strong predictor.
- Model Result: Best model is Decision Tree (Train Accuracy = 0.87, Test Accuracy = 0.83).
- Deliverables: EDA (visual + SQL), Folium map, Plotly Dash dashboard, and classification model.

Introduction

- **Background:** First-stage landing is critical for reusability and cost reduction in Falcon 9 missions.
- **Problem Statement:** Identify key factors that influence landing outcome and predict landing success (Class).
- **Key Questions:**
  - How do launch site, orbit, and payload relate to landing success?
  - Does landing success improve over time?
  - What patterns can be validated using SQL analysis?
  - Which classification model performs best for predicting success?
- **Outputs:** Visual EDA, SQL-based EDA, interactive map (Folium), dashboard (Plotly Dash), and a predictive model.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API + web scraping to build the initial dataset.
- Perform data wrangling
  - Cleaning, type conversion, handling missing values, and creating the target label (Class)
- EDA: Visual exploration and SQL queries to identify patterns and key relationships.
- Interactive Analytics: Folium map for launch sites and Plotly Dash dashboard for filtering and visual analysis.
- Perform predictive analysis using classification models
  - Train/test split, feature preprocessing, and classification modeling to predict landing success.

Data Collection

- **Sources:** SpaceX REST API and Wikipedia (web scraping).
- **API Data:** Retrieved launch records (e.g., flight number, launch site, orbit, payload, landing outcome).
- **Scraping Data:** Extracted additional launch table information from Wikipedia and merged with API data.
- **Output:** Combined raw dataset prepared for cleaning and feature engineering.

Data Collection – SpaceX API

- Pulled launch data using SpaceX REST API (JSON).
- Main endpoint: /v4/launches (core launch records).
- Enriched data via: /v4/rockets, /v4/launchpads, /v4/payloads (lookup by IDs).
- Joined datasets to create the analysis table and derived the target Class label.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/1-jupyter-labs-spacex-data-collection-api.ipynb

Start →

GET /v4/launches →

Select needed fields →
GET /v4/rockets + GET /v4/launchpads + GET /v4/payloads →
Merge by IDs

(rocket, launchpad, payloads) →

Create Class label →

Export dataset (CSV/DataFrame)

Data Collection - Scraping

- Collected additional launch information using **web scraping** from Wikipedia tables.
- Parsed the HTML table into a structured DataFrame (rows = launches, columns = attributes).
- Cleaned and standardized column names and data types for merging with API data.
- Output: a scraped dataset used to enrich the main launch table.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/2-jupyter-labs-webscraping.ipynb

Start →

Open Wikipedia page (Falcon 9 launches table) →

Read HTML table →
Parse rows/columns →

Clean (rename columns, remove symbols, convert types) →
Create DataFrame →

Export scraped dataset

Data Wrangling

- Start → Load API + Scraped datasets → Merge/Join tables → Select relevant features → Handle missing values → Convert data types → Create Class label → Final cleaned dataset (90×17)
- Merged API and scraped datasets and selected the required features for analysis.
- Cleaned the data (handled missing values, fixed data types, removed inconsistencies).
- Created the target label Class (landing success/failure) for classification.
- Output: final cleaned dataset (90 × 17) used for EDA and modeling.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/3-labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Performed exploratory analysis using scatter plots, bar charts, and line charts.
- Visualized relationships between Flight Number, Payload Mass, Launch Site, and Orbit versus the landing outcome (Class).
- Identified trends over time and differences in success rates across categories.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/5-edadataviz.ipynb

EDA with SQL

- Used SQL queries to validate EDA findings and summarize launch performance.
- Queried launch sites, payload statistics, and landing outcomes (success vs failure).
- Identified patterns such as highest payload missions, outcome counts, and time-based rankings.
- SQL results were used to support visual insights and feature selection.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Built an interactive map using Folium to visualize launch site locations.
- Added markers and popups for each launch site and mapped landing outcomes (success/failure).
- Used distance/proximity visualization to analyze nearby infrastructure around selected sites.
- The map provides geographic context to support EDA findings.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/6-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Developed an interactive Plotly Dash dashboard for exploratory analysis.
- Included a launch site dropdown filter and a payload range slider for dynamic exploration.
- Visualized success metrics using pie charts (success count by site) and a scatter plot (payload vs outcome).
- The dashboard allows quick comparison of performance across sites and payload ranges.
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

- Built a classification pipeline to predict landing success (Class).
- Split the dataset into train/test sets and prepared features using encoding (and scaling if required).
- Trained multiple classifiers (e.g., Logistic Regression, SVM, KNN, Decision Tree) and used GridSearchCV for tuning.
- Evaluated models using accuracy and selected the best-performing model for final results.
- Start → Select features + target (Class) → Train/Test Split → Preprocess (One-Hot Encoding / Scaling if used) → Train models → Hyperparameter tuning (GridSearchCV) → Evaluate (Accuracy + Confusion Matrix) → Select best model
- https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/7-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

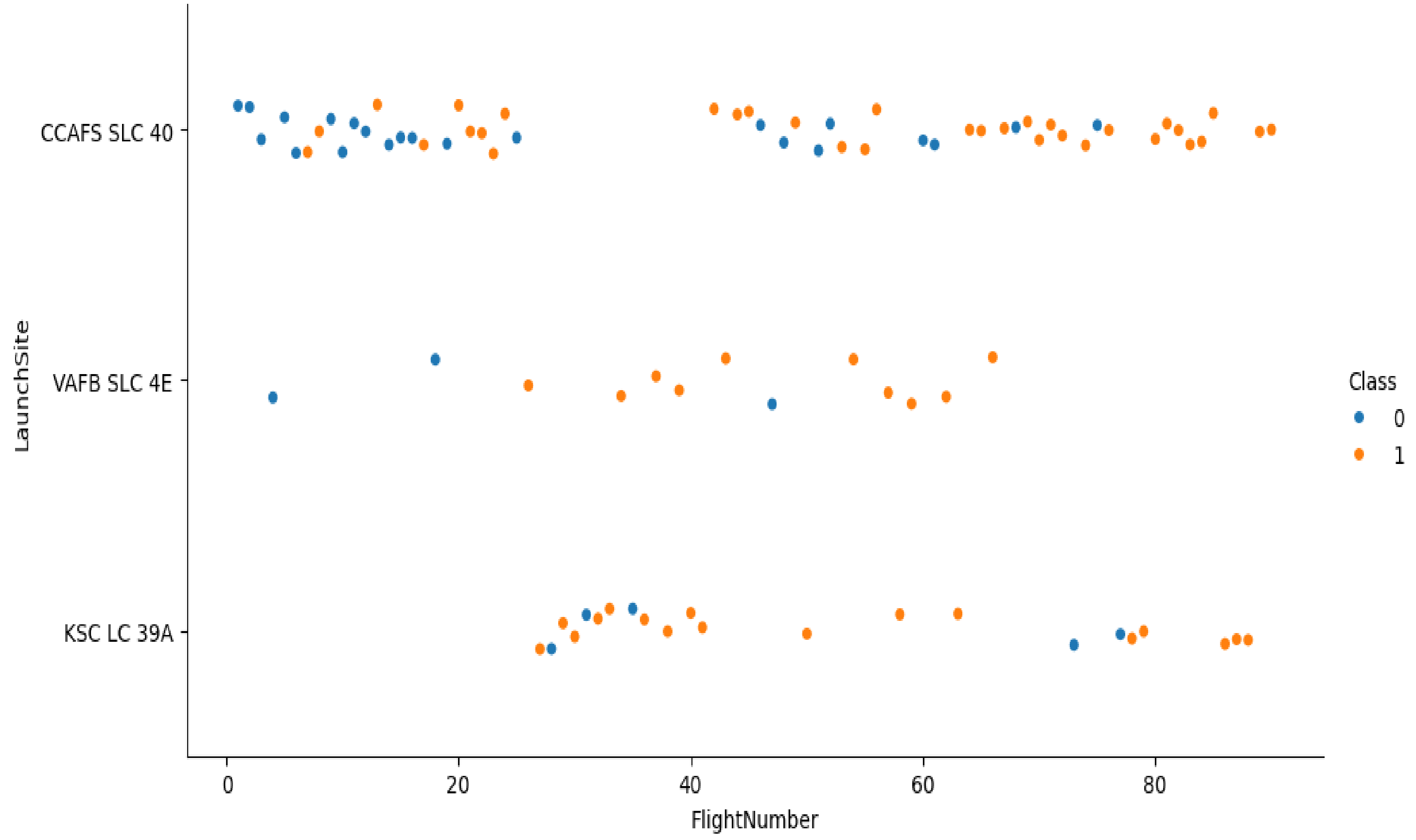
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

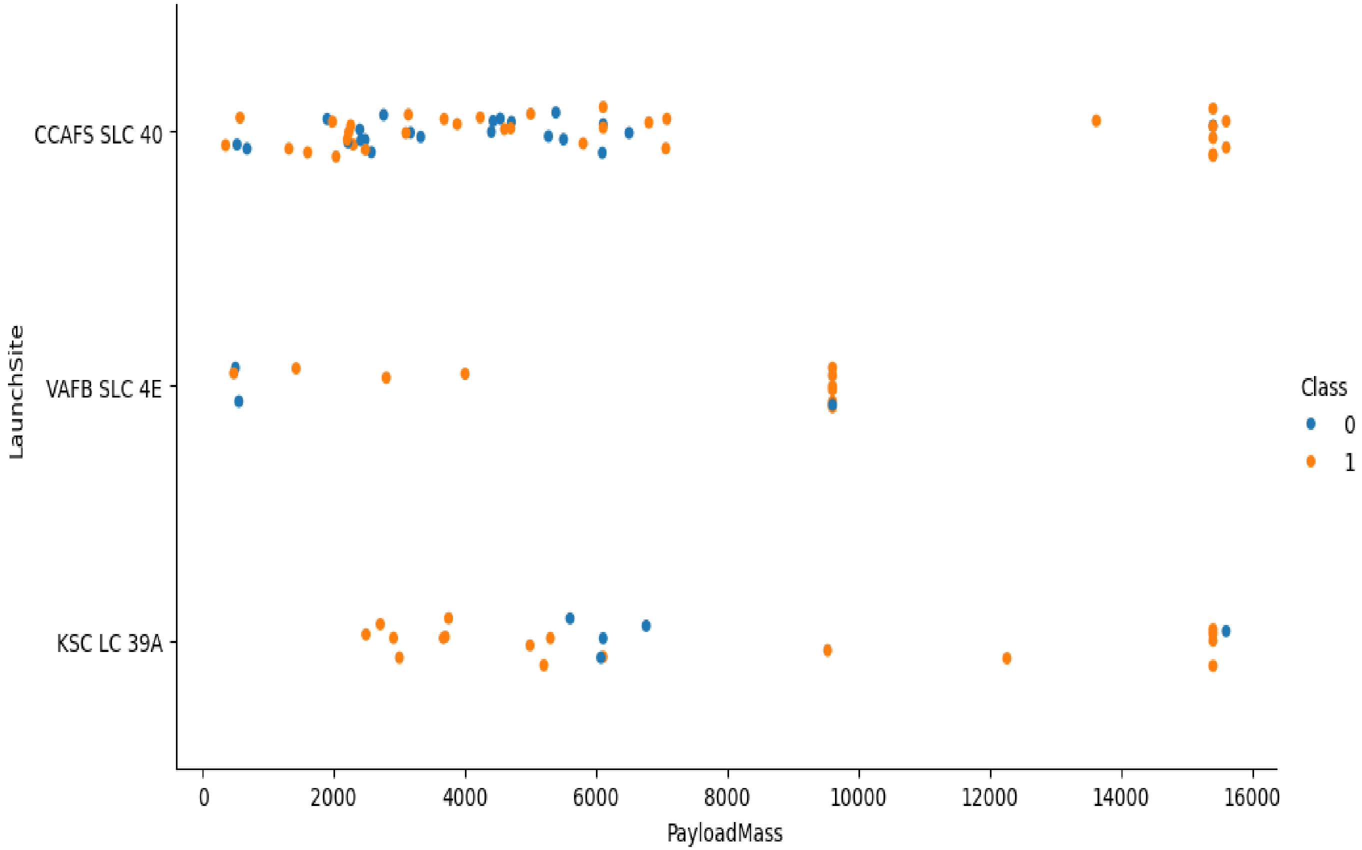
Flight Number vs. Launch Site

- Landing success (Class=1) becomes more frequent in later flights across all sites.
- CCAFS SLC 40 has the highest number of launches, so most successes are observed there.
- KSC LC 39A shows mostly successful outcomes after it starts appearing in the dataset.



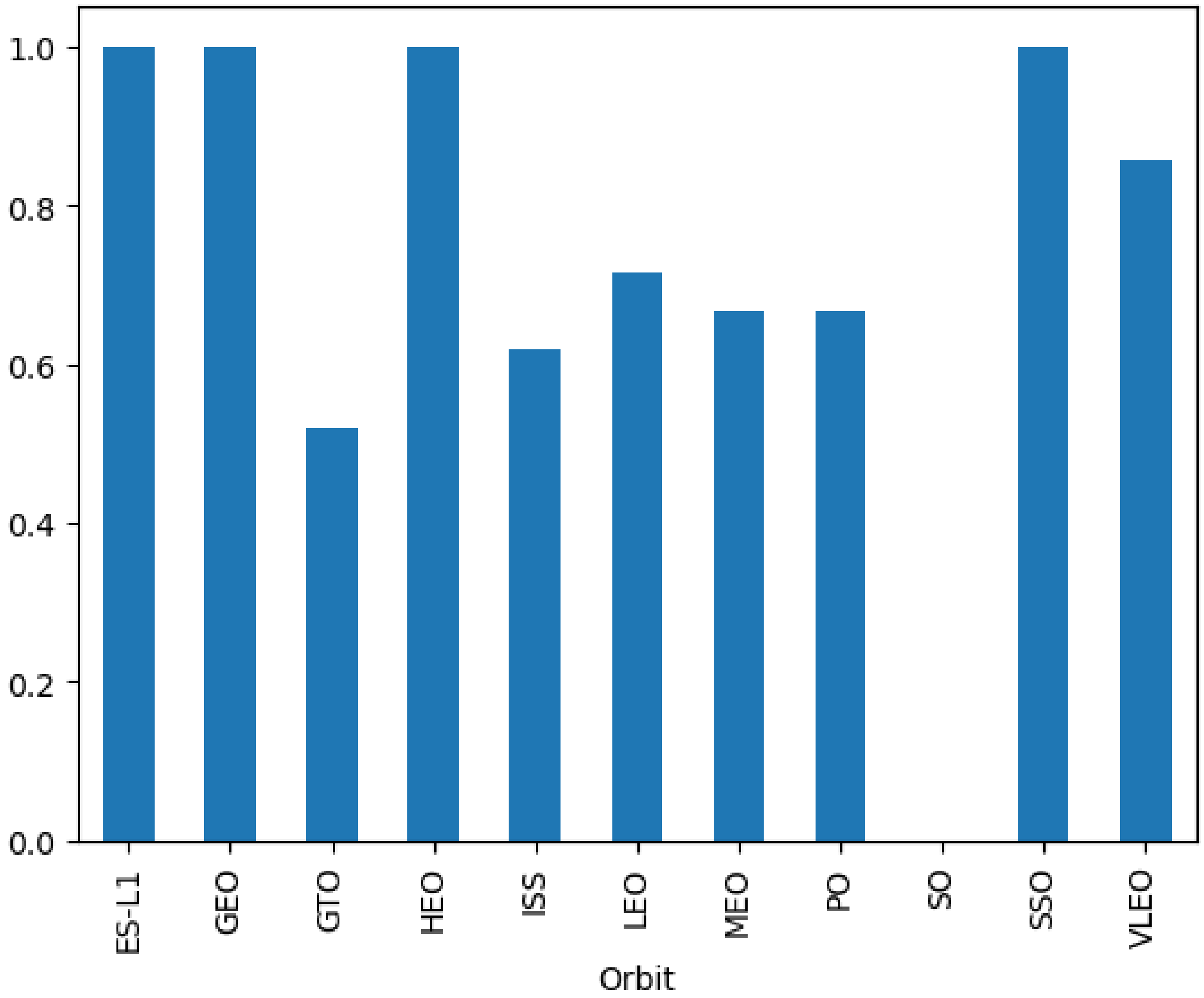
Payload vs. Launch Site

- Successful landings occur across a wide payload range at CCAFS and KSC.
- VAFB has fewer launches, but includes successful outcomes at mid payload masses.
- Payload mass alone does not fully separate success/failure; launch site context matters.



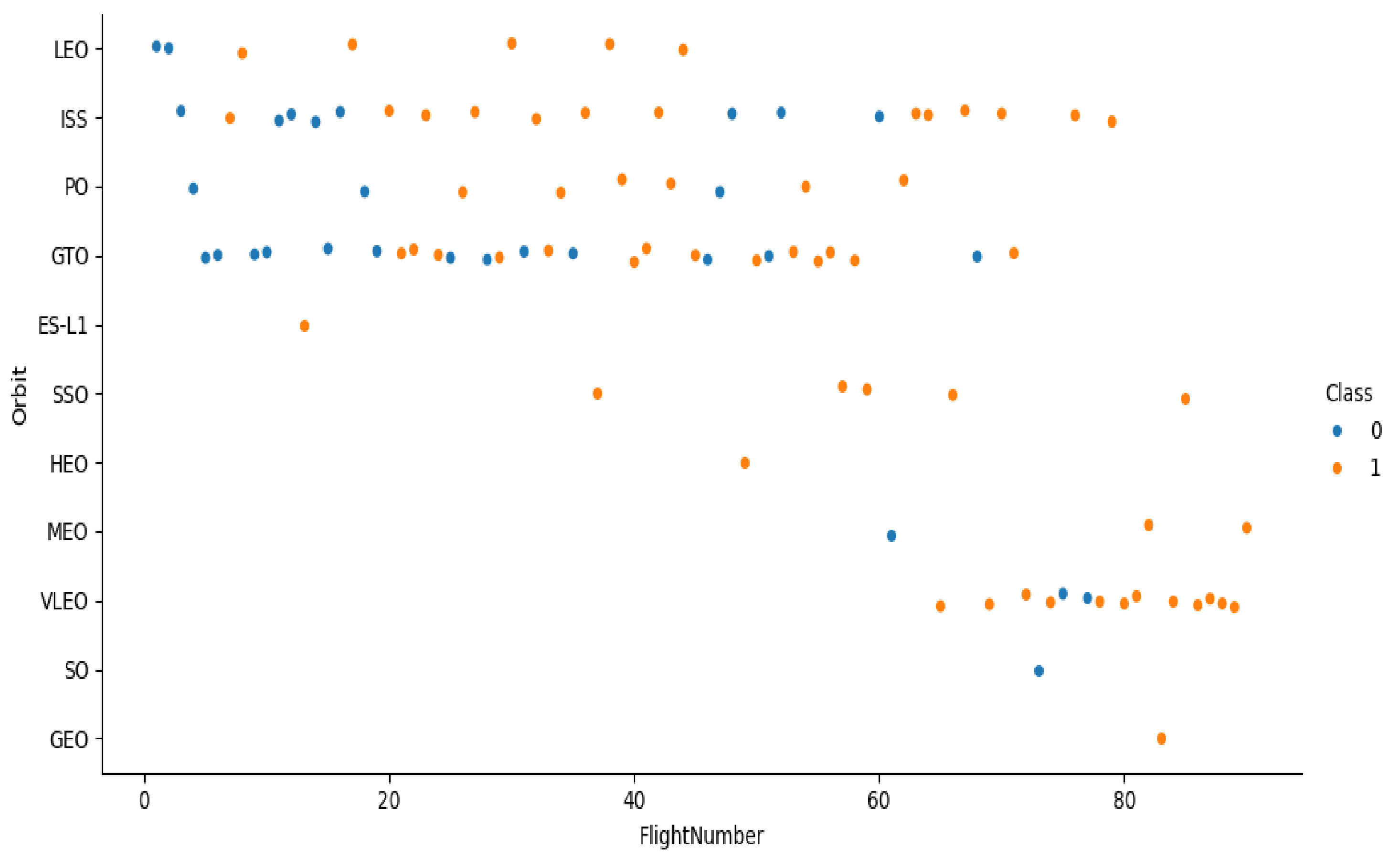
Success Rate vs. Orbit Type

- Landing success rate varies by orbit type, indicating orbit is an important predictor.
- Some orbits show consistently higher success rates than others.
- This supports using Orbit as a key feature in the classification model.



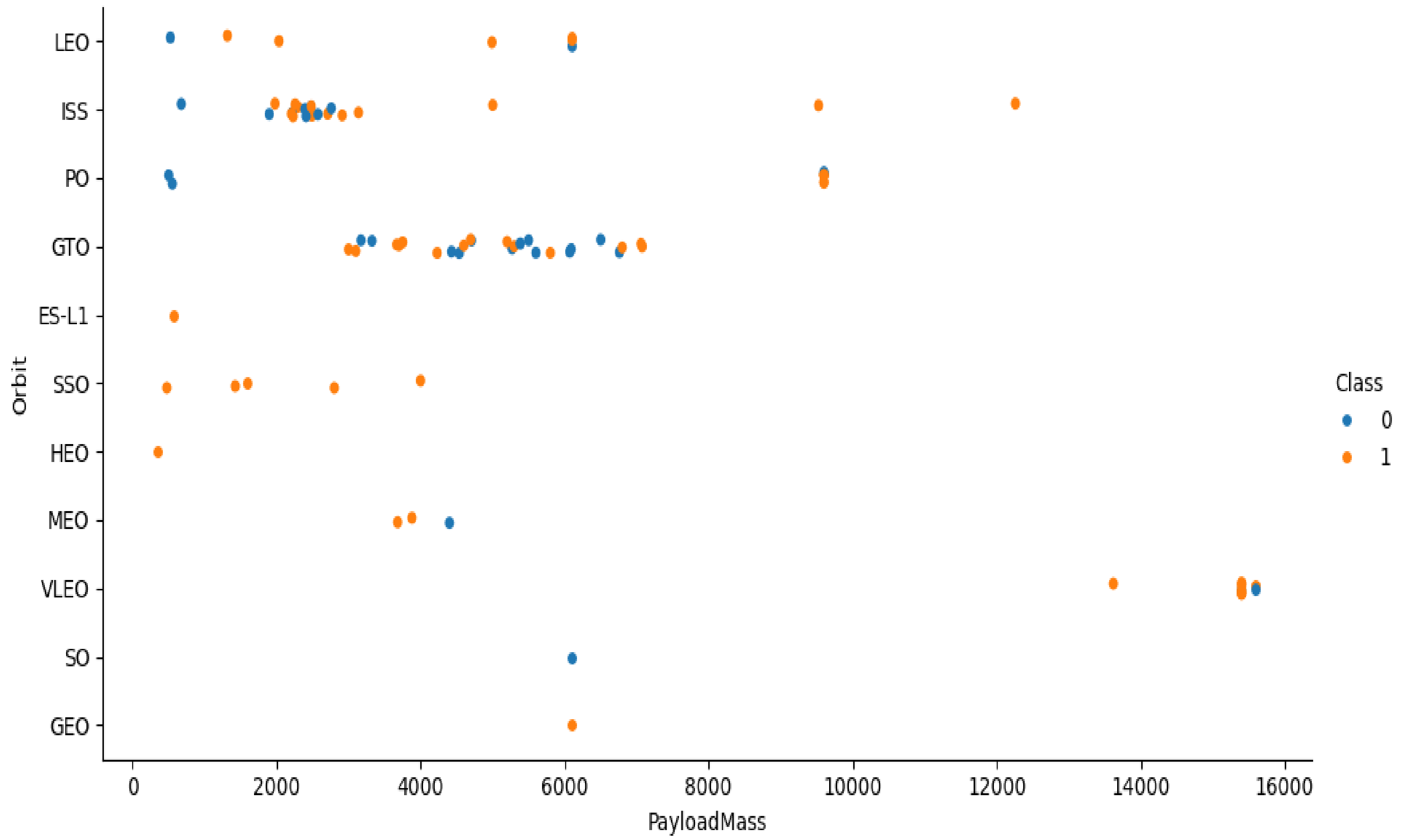
Flight Number vs. Orbit Type

- Successful outcomes (Class=1) appear more frequently at higher flight numbers across multiple orbit types.
- Certain orbits show a higher concentration of successes compared to others.
- This suggests both mission experience (time) and orbit type influence landing success.



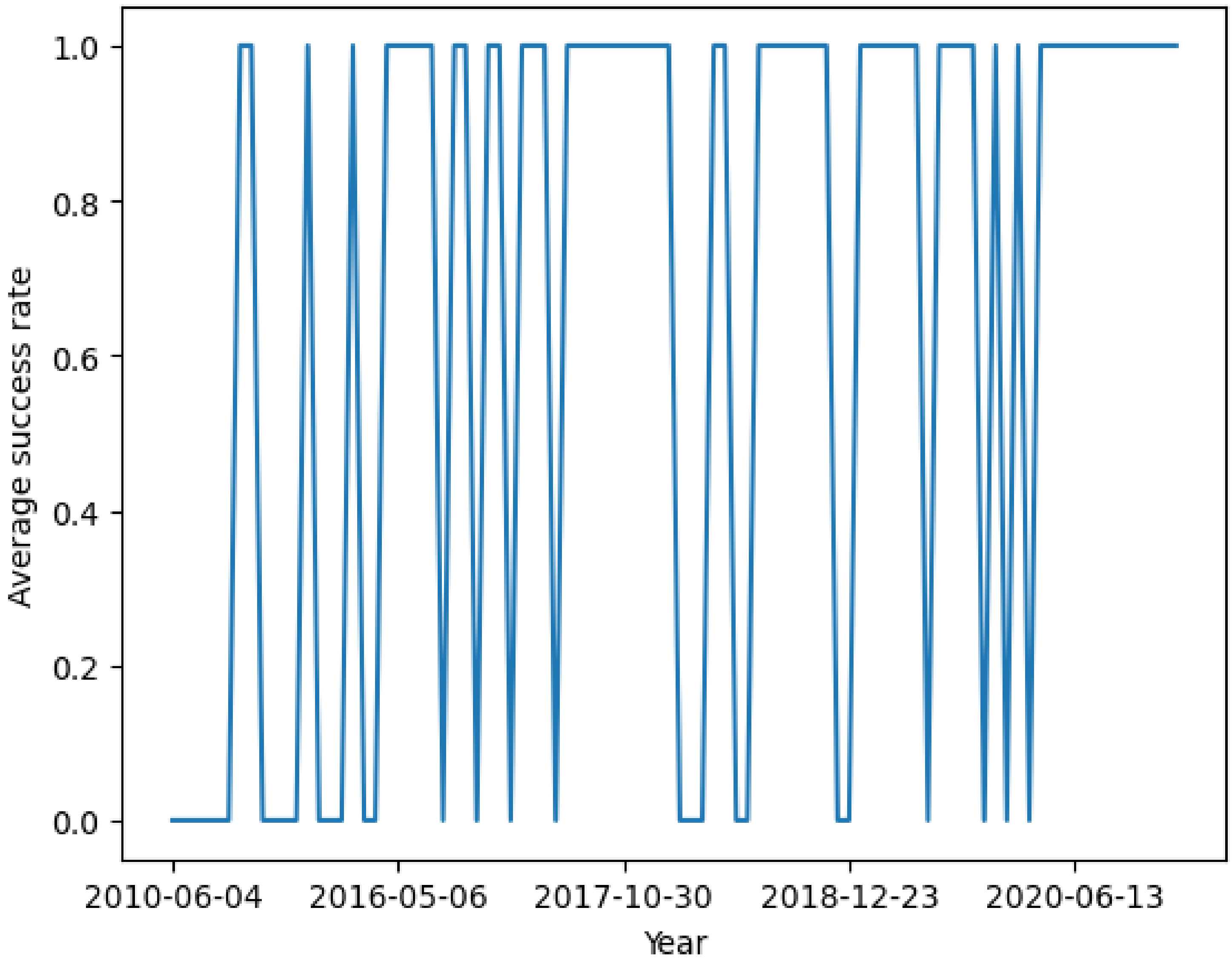
Payload vs. Orbit Type

- Success patterns differ by orbit across payload ranges, showing an interaction between orbit and payload.
- Some orbits achieve high success even at higher payload masses.
- Payload mass alone is not sufficient; orbit type provides key separation for outcomes.



Launch Success Yearly Trend

- Overall landing success improves over time, with a higher frequency of successes in later years.
- Earlier years contain more failures, while later years show more consistent successful landings.
- This indicates an operational learning effect over time.



All Launch Site Names

- Launch sites: CCAFS SLC 40, KSC LC 39A, VAFB SLC 4E.
- This query lists all distinct launch site names in the dataset.
- The output confirms the launch sites used for subsequent EDA and comparisons.

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

Launch Site Names Begin with 'CCA'

- This query filters launch sites whose names begin with "CCA".
- The result isolates CCAFS-related launches for focused analysis of that site group.

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

- This query calculates the total payload mass carried for NASA missions.
- The result summarizes NASA's contribution to the overall payload delivered in the dataset.

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- This query computes the average payload mass for missions flown with Falcon 9 v1.1.
- The output provides a baseline payload level for this rocket configuration.

avg_payload_mass
2928.4

First Successful Ground Landing Date

- This query identifies the first date when a successful ground landing occurred.
- The result marks an important milestone in landing capability over time.

first_success_groundpad_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query filters missions with successful drone ship landings and payload mass between 4000 and 6000.
- The output highlights successful missions within this payload range.

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

Total Number of Successful and Failure Mission Outcomes

- This query counts the total number of successful and failed landing outcomes.
- The result provides the overall class distribution used in model evaluation.

outcome_group	total
Failure	40
Success	61

Boosters Carried Maximum Payload

- This query identifies the booster(s) that carried the maximum payload mass in the dataset.
- The output links the heaviest payload missions to their booster configuration.

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

- This query retrieves 2015 launch records with a failed drone ship landing outcome.
- The result lists the related booster and launch site details for those missions.

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

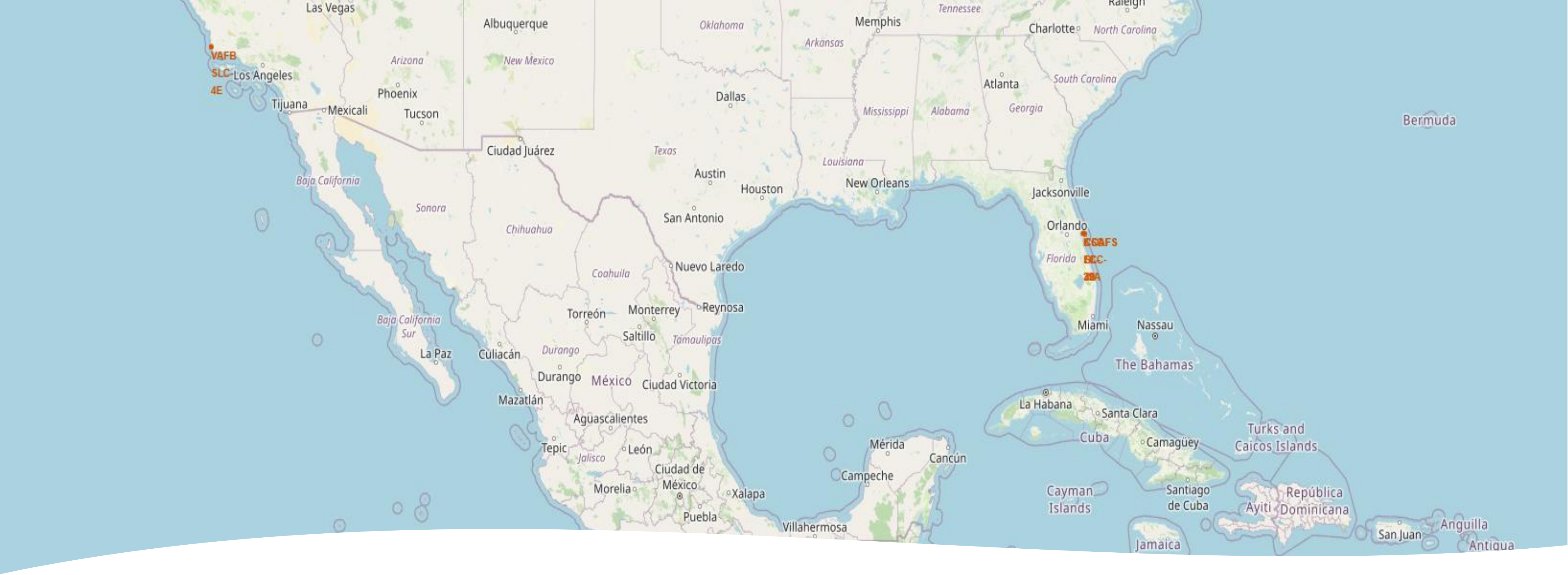
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- This query ranks landing outcomes between the specified dates using a window function.
- The output helps compare missions within the selected time period by outcome and performance.

Landing_Outcome	cnt
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

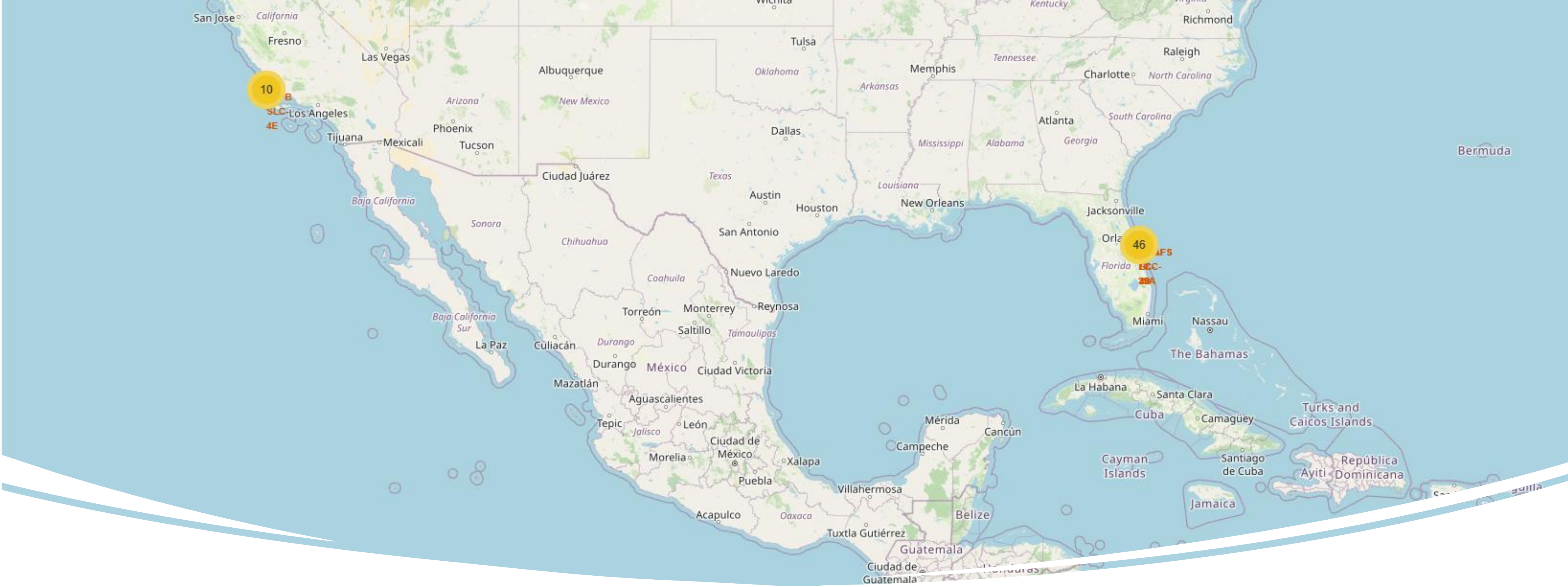
Section 3

Launch Sites Proximities Analysis



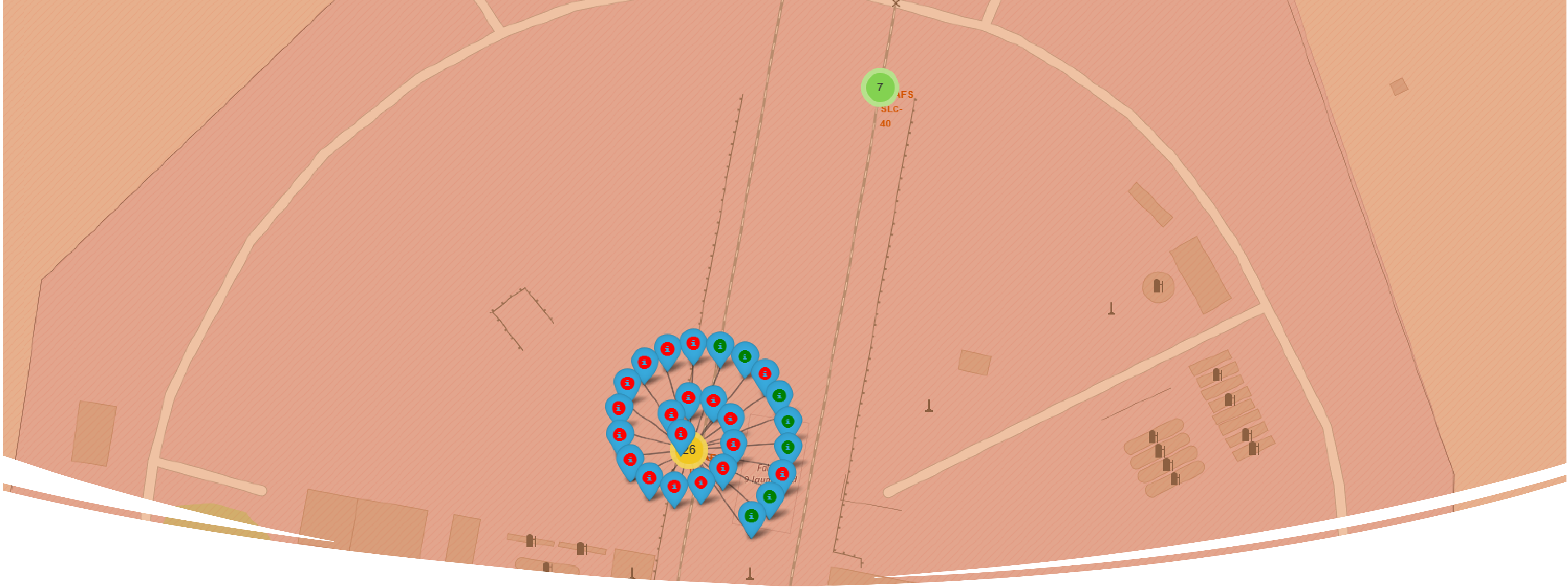
<Folium Map Screenshot 1>

- This map shows the geographic locations of all Falcon 9 launch sites using Folium markers.
- Each marker represents a launch site and provides location context for further analysis.



<Folium Map Screenshot 2>

- This map visualizes landing outcomes by location (success vs failure) using colored markers.
- It helps compare how outcomes are distributed across different launch sites.

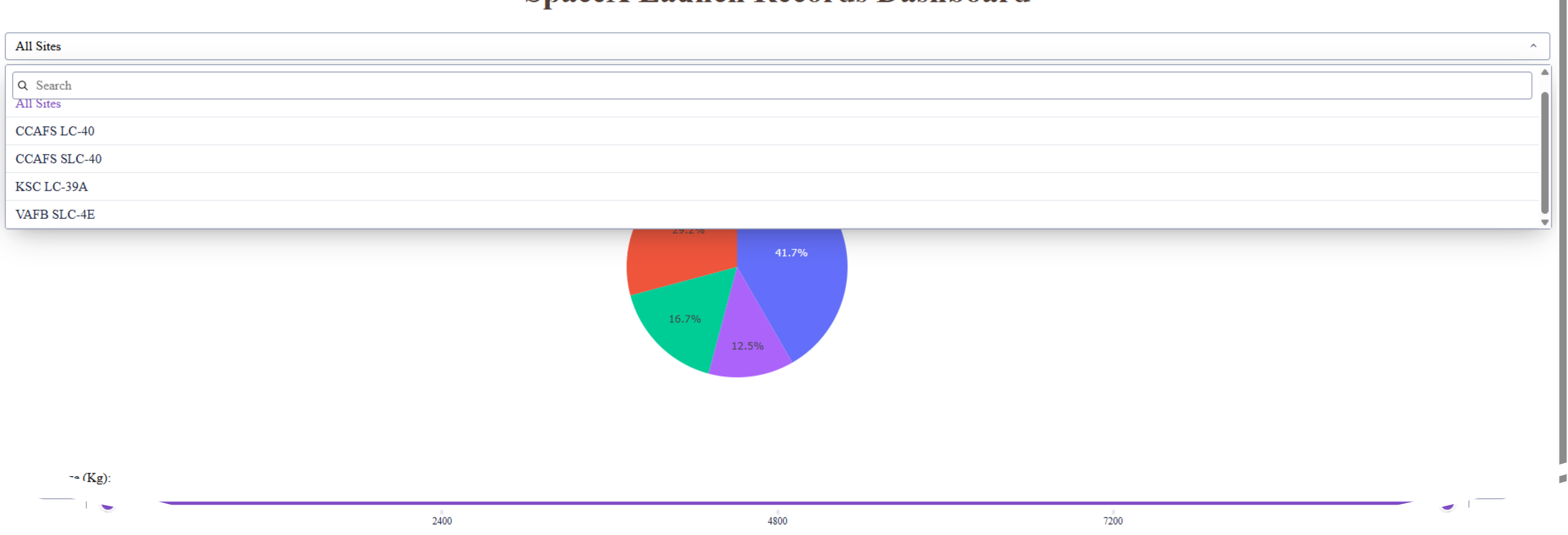


<Folium Map Screenshot 3>

- This map performs a proximity analysis around a selected launch site (e.g., coastline, highway, railway).
- Distances are visualized to provide geographic context for site selection and operations.

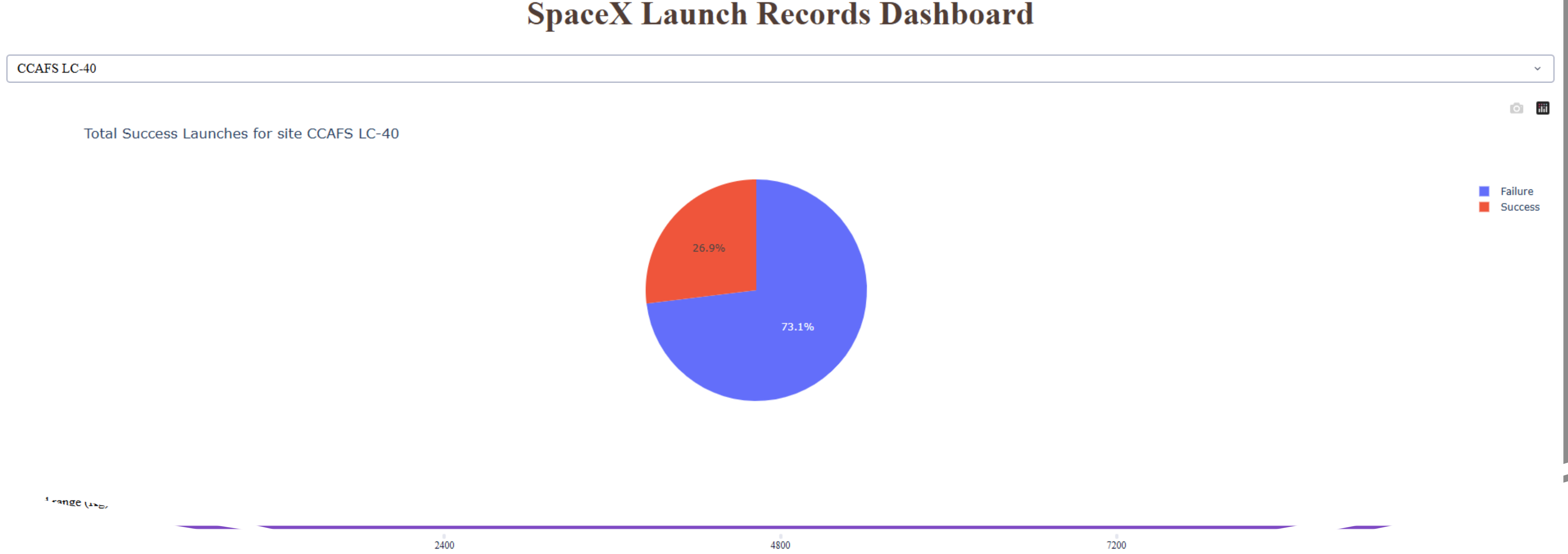
Section 4

Build a Dashboard with Plotly Dash



<Dashboard Screenshot 1>

- This dashboard view shows the overall distribution of landing outcomes for all launch sites.
- The pie chart summarizes total success vs failure counts across the entire dataset.

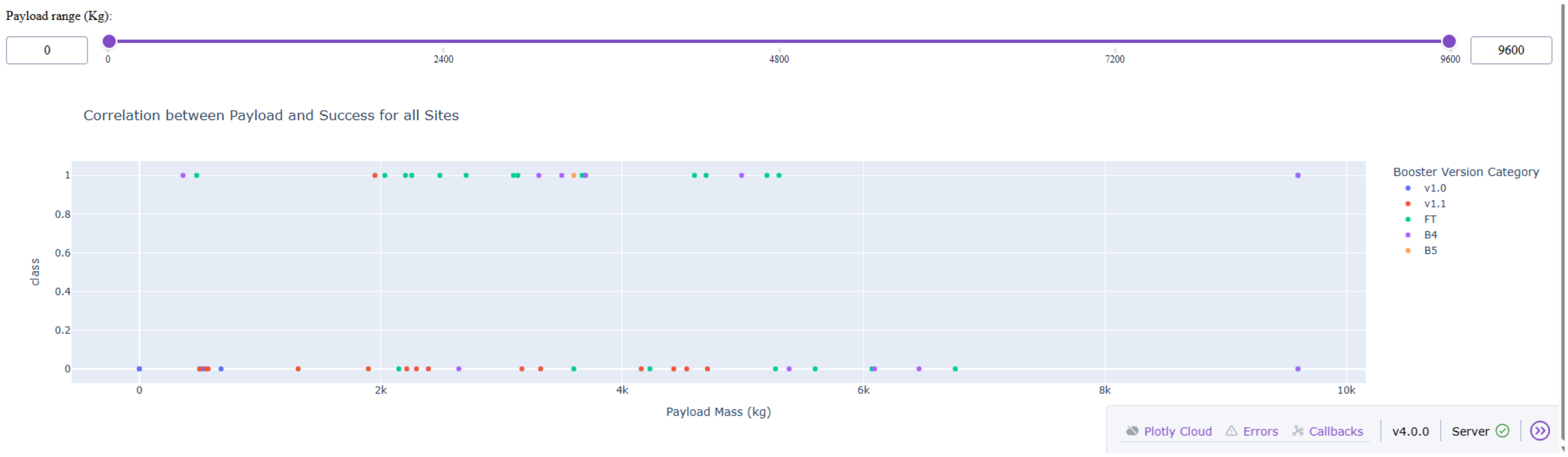


<Dashboard Screenshot 2>

- This view filters the dashboard to a single launch site and displays its landing outcome distribution.
- The pie chart enables quick comparison of success vs failure for the selected site.

<Dashboard Screenshot 3>

- The scatter plot shows the relationship between payload mass and landing outcome (Class) for the selected site(s).
- Adjusting the payload range slider updates the plot to focus on specific payload intervals and compare success patterns.

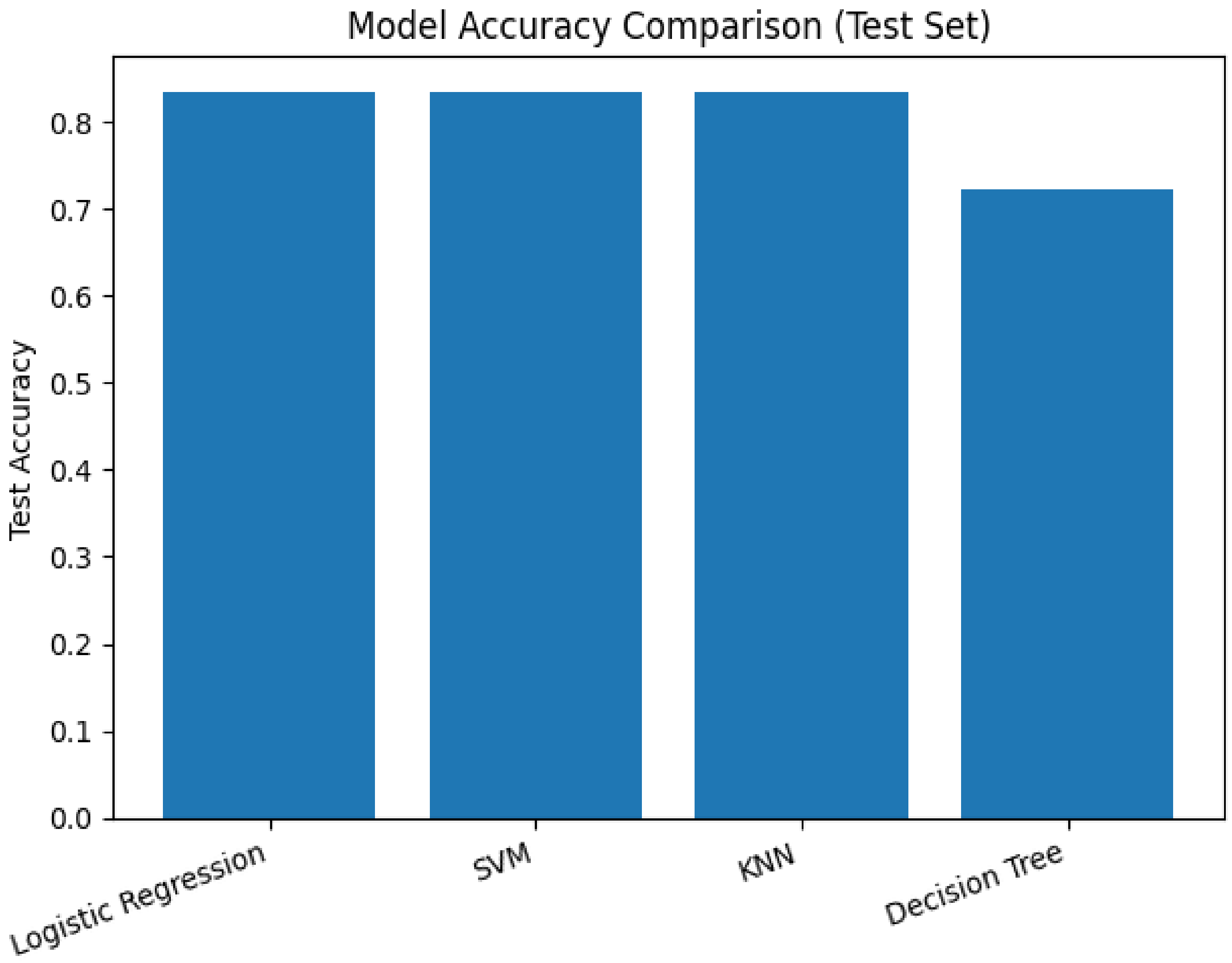


Section 5

Predictive Analysis (Classification)

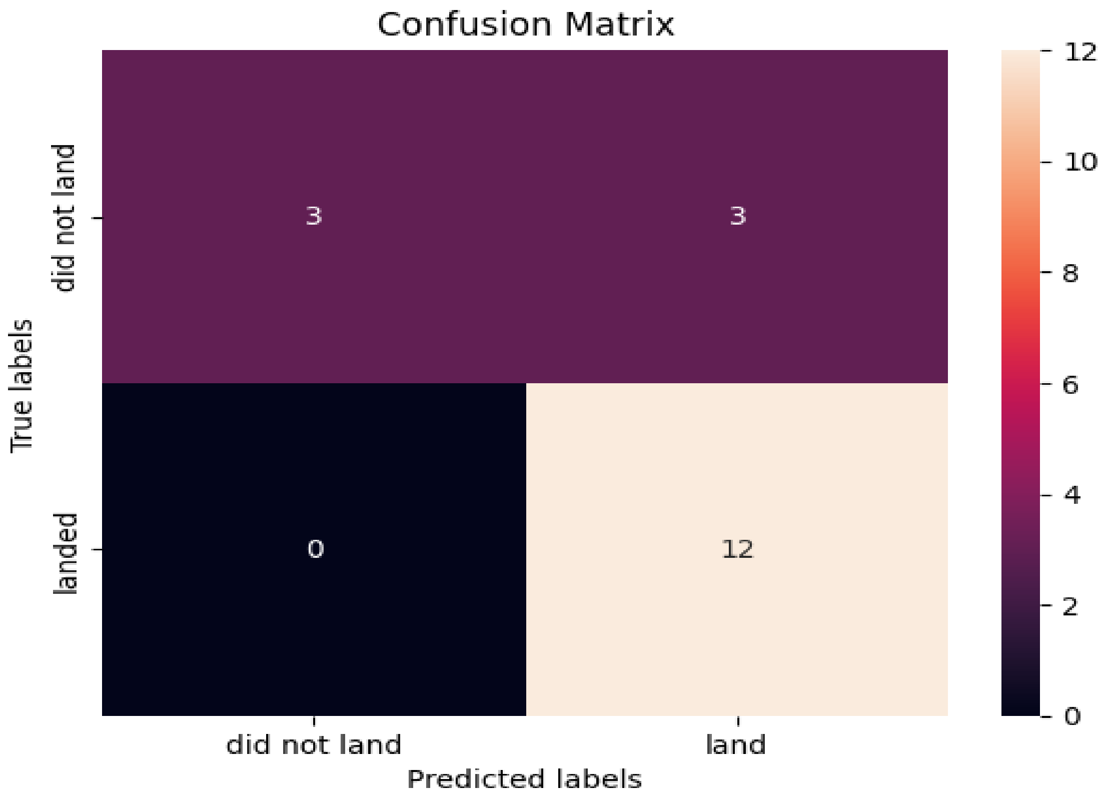
Classification Accuracy

- Compared tuned classifiers (Logistic Regression, SVM, Decision Tree, KNN) using test accuracy.
- All models achieved the same performance on the test set (Test Accuracy = 0.83).



Confusion Matrix

- Confusion matrix for the selected classifier on the test set.
- Most errors are false positives, where a non-landing is predicted as a landing.



Conclusions

- Landing success increases in later flights, indicating a clear improvement over time.
- Success rate differs across orbit types, so Orbit is an important predictive feature.
- Interactive Folium maps and the Dash dashboard support quick exploration by site and payload range.
- Classification models achieved ~0.83 test accuracy, and the confusion matrix shows remaining errors are mainly false positives.

Appendix

- SpaceX API Calls:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/1-jupyter-labs-spacex-data-collection-api.ipynb
- Web Scraping:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/2-jupyter-labs-webscraping.ipynb
- Data Wrangling:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/3-labs-jupyter-spacex-Data%20wrangling.ipynb
- EDA with SQL:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/4-jupyter-labs-eda-sql-coursera_sqllite.ipynb
- EDA Visualization:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/5-edadataviz.ipynb
- Folium Map:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/6-lab_jupyter_launch_site_location.ipynb
- Machine Learning Prediction:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/7-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb
- Plotly Dash App:
  - https://github.com/tcdn7/17-IBM-Test-Repo/blob/main/spacex-dash-app.py

Thank you!