IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sourena Mohit Tabatabaie
January 5, 2026

Outline

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

Executive Summary

Goal: Predict whether a Falcon 9 first stage will land successfully, and identify operational factors (payload, orbit, launch site, etc.) correlated with success

What I did (pipeline):

- ✓ Collected SpaceX launch data via **SpaceX REST API** and **Wikipedia web scraping**
- ✓ Cleaned & wrangled into a structured dataset
- ✓ Performed EDA using **visualization + SQL**
- ✓ Built interactive analytics with **Folium map + Plotly Dash**
- ✓ Trained and evaluated multiple classification models (LogReg / SVM / Decision Tree / KNN) and selected best model

High-level outcome:

- ➢ Launch success improves over time (experience effect)
- ➢ Success patterns vary by **orbit**, **payload range**, and **launch site**
- ➢ Best-performing ML models reached **~0.833 test accuracy**

Introduction

- **Background**
  - Reusability of Falcon 9 first stage is key to lowering launch cost, so predicting landing success helps planning, safety, and cost estimation.

- **Problem statement**
  - Can we predict first-stage landing success using launch-level features?

- **Questions**
  1. Which features (payload, orbit, booster version, launch site, etc.) correlate with success?
  2. Are some launch sites geographically advantageous (coastline, infrastructure, distance to city, etc.)?
  3. What classification model best predicts landing success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collected launch-level data from the **SpaceX REST API** (launches, rockets, payloads, launchpads/cores) and joined them into one analysis table.
  - Performed **Wikipedia web scraping** (Falcon 9 launch records tables) to extract/validate mission and landing outcome information not directly available in the API.
- Perform data wrangling
  - Cleaned and standardized fields (dates, orbit names, launch site names, booster versions, outcome text).
  - Handled missing values (e.g., payload mass and incomplete outcome fields) and removed/filtered non-informative columns.
  - Created the binary target label class: 1 = successful landing/success, 0 = failure/unsuccessful landing (consistent definition used across analysis).
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built classification models using engineered features (payload mass + one-hot encoded categorical variables like orbit/site/booster) ; Trained and compared Logistic Regression, SVM, Decision Tree, KNN to predict class.

How classification models were built, tuned, and evaluated

- Split data into **train/test** sets; applied preprocessing (encoding + scaling when needed).
- Used **GridSearchCV** to tune hyperparameters for each model.
- Evaluated using **test accuracy** and **confusion matrix** (and optionally precision/recall/F1/ROC-AUC if included) and selected the best-performing model based on generalization performance.

Data Collection

Primary source: SpaceX public REST API (launches + linked objects)

Secondary source: Wikipedia tables (Falcon 9 launch records) via web scraping to enrich/validate outcomes

Goal: build one clean dataset to analyze factors affecting first-stage landing success (class)

SpaceX REST API (JSON)

↓

Extract launches + IDs (rocket / payload / launchpad / cores)

↓

Call linked endpoints to enrich launch records

↓

Wikipedia scraping (HTML tables) → landing outcome / mission details

↓

Merge + clean → final analysis dataset (CSV / DataFrame)

Data Collection – SpaceX API

Queried SpaceX API endpoints to retrieve launch records and related metadata:

- Launches (core launch info)
- Rockets (rocket/booster details)
- Payloads (payload mass, orbit)
- Launchpads (site name + coordinates)

Built a relational-style dataset by joining endpoint responses using IDs.

Link :Winning-Space-Race-With-Data-Science/Module_1/jupyter-labs-spacex-data-collection-api.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science

/v4/launches/past (or launches endpoint)

↓ (extract IDs: rocket, payloads[], launchpad, cores[])

/v4/rockets/{rocket_id} → booster/rocket attributes

/v4/payloads/{payload_id} → payload mass + orbit

/v4/launchpads/{launchpad_id}→ site name + latitude/longitude

↓

Merge all into one table (one row per launch)

↓

Export dataset for wrangling + EDA

Data Collection - Scraping

- Source: Wikipedia Falcon 9 launch records tables
- Tools: requests + BeautifulSoup (or pandas.read_html) → HTML tables → DataFrame
- Parsed and standardized: dates, launch site, orbit, payload, landing outcome
- Cleaned outcome strings and kept only analysis-relevant columns
- Exported results for merging with API dataset (spacex_web_scraped.csv)
- Link : [Winning-Space-Race-With-Data-Science/Module_1/jupyter-labs-webscraping.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science]()

Wikipedia page (HTML tables)

HTTP request (requests)

Parse tables (BeautifulSoup / pandas.read_html)

Clean & standardize columns (dates, orbit, outcomes)

Extract landing outcome + mission fields

Save as CSV (spacex_web_scraped.csv) → merge with API dataset

Data Wrangling

- Loaded datasets from API + scraping outputs and selected relevant features (payload mass, orbit, site, booster, outcomes).
- Cleaned missing/invalid values (e.g., payload mass) and fixed data types (dates/numerics/categories).
- Standardized categorical values (orbit/site/booster text normalization).
- Created target label class:
  - 1 = successful landing / mission success
  - 0 = failure / unsuccessful landing
- Prepared a modeling-ready dataset:
  - one-hot encode categorical fields (orbit/site/booster)
  - keep numeric payload mass and flight number
  - final table used in EDA + SQL + ML

Raw API + Scraped tables
↓
Select relevant columns + unify schema
↓
Clean missing values + type conversions
↓
Standardize categorical fields (orbit/site/booster)
↓
Create label: class (1=success, 0=failure)
↓
Export final dataset → EDA / SQL / Folium / Dash / ML

Link : [Winning-Space-Race-With-Data-Science/Module_1/labs-jupyter-spacex-Data wrangling.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science]

EDA with Data Visualization

Summarize what charts were plotted & why

- Flight Number vs Launch Site (scatter)
  - Used to check the learning curve over time and whether some sites reached high success earlier than others.
- Payload Mass vs Launch Site (scatter)
  - Used to see how payload distributions differ by site and whether success changes with payload range.
- Success Rate vs Orbit Type (bar/aggregated plot)
  - Used to compare mission difficulty by orbit and identify which orbits have consistently higher/lower success.
- Flight Number vs Orbit Type (scatter)
  - Used to examine how success evolves over time across different mission profiles (orbit categories).
- Payload Mass vs Orbit Type (scatter)
  - Used to analyze the interaction between payload constraints and orbit choice (payload-orbit tradeoff).
- Yearly Average Success Trend (line)
  - Used to summarize global improvement in success rate over years (operational maturity + reuse improvements).

Link:Winning-Space-Race-With-Data-Science/Module_2/edadataviz.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science

EDA with SQL

Summarize SQL queries performed

- Identify distinct values (e.g., launch sites, booster versions, orbit types) using SELECT DISTINCT.
- Filter by patterns (e.g., sites starting with CCA%) using LIKE.
- Aggregation on payload mass using SUM(payload_mass), AVG(payload_mass)
- Landing success logic using filters on Landing_Outcome (e.g., first successful landing date; successful drone-ship landings in a payload range).
- Group-by outcome counts (success vs failure totals; landing outcome frequencies) using GROUP BY.
- Find max payload missions per booster using MAX(payload_mass) and associated grouping.
- Time-based filtering (e.g., year 2015 failures; date range 2010-06-04 to 2017-03-20) using date comparisons and strftime().

Link:Winning-Space-Race-With-Data-Science/Module_2/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science

Build an Interactive Map with Folium

- folium.Map centered on a reference location with an appropriate zoom level.
- Launch site markers + circles: folium.Marker + folium.Circle for each launch site (with popup labels).
- Outcome markers: color-coded markers for each launch (success vs failure) using a marker cluster to reduce clutter.
- Proximity markers: markers for nearest coastline / highway / railway / city points.
- Distance lines: folium.PolyLine lines from launch site to proximity points.
- Distance calculation: computed distances (km) using the haversine formula.
- Why I added these objects
  - To visualize where launch sites are geographically and compare locations across the US.
  - To see whether launches cluster near the coast (safety corridor) and infrastructure (logistics).
  - To connect geography with performance by overlaying success/failure outcomes on the map.
  - To quantify "site advantages" using distance-to-infrastructure measurements.

Link:Winning-Space-Race-With-Data-Science/Module_3/lab_jupyter_launch_site_location.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science

Build a Dashboard with Plotly Dash

[Link.Winning-Space-Race-With-Data-Science/Module_3/Build an Interactive Dashboard with Plotly Dash.py at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science]

- plots/interactions I added
  - Launch Site dropdown: filter all visuals by selected site (or “ALL sites”).
  - Pie chart:
  - For “ALL sites”: distribution of successful launches by site.
  - For a selected site: Success vs Failure breakdown.
  - Payload range slider: filter points by payload mass range.
  - Scatter plot: Payload Mass vs Launch Outcome (class), colored by booster/version category (or other categorical feature).
- Why I added these plots/interactions
  - To let users explore relationships quickly without rerunning notebooks.
  - Pie chart gives a fast comparison of site performance and failure concentration.
  - Payload slider + scatter helps reveal how success varies by payload range and site.
  - Overall, the dashboard supports interactive hypothesis testing (site/orbit/payload effects).

Predictive Analysis (Classification)

- Feature preparation: selected predictive features (e.g., payload mass + categorical variables such as orbit, launch site, booster version) and created the target label class (1=success, 0=failure).
- Preprocessing: applied one-hot encoding for categorical variables; applied standardization for numeric features when required (especially for SVM/KNN).
- Train/Test split: split the dataset into training and testing sets with a fixed random seed for reproducibility.
- Model training: trained multiple classifiers: Logistic Regression, SVM, Decision Tree, KNN.
- Model tuning: used GridSearchCV (cross-validation) to tune hyperparameters for each model.
- Model evaluation: compared models using test accuracy and inspected confusion matrix for the best model to understand false positives/false negatives.
- Best model selection: selected the best performing (in your run, models achieved similar accuracy; choose one as final model and report its confusion matrix).

Link:Winning-Space-Race-With-Data-Science/Module_4/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · Sourena-Mohit/Winning-Space-Race-With-Data-Science

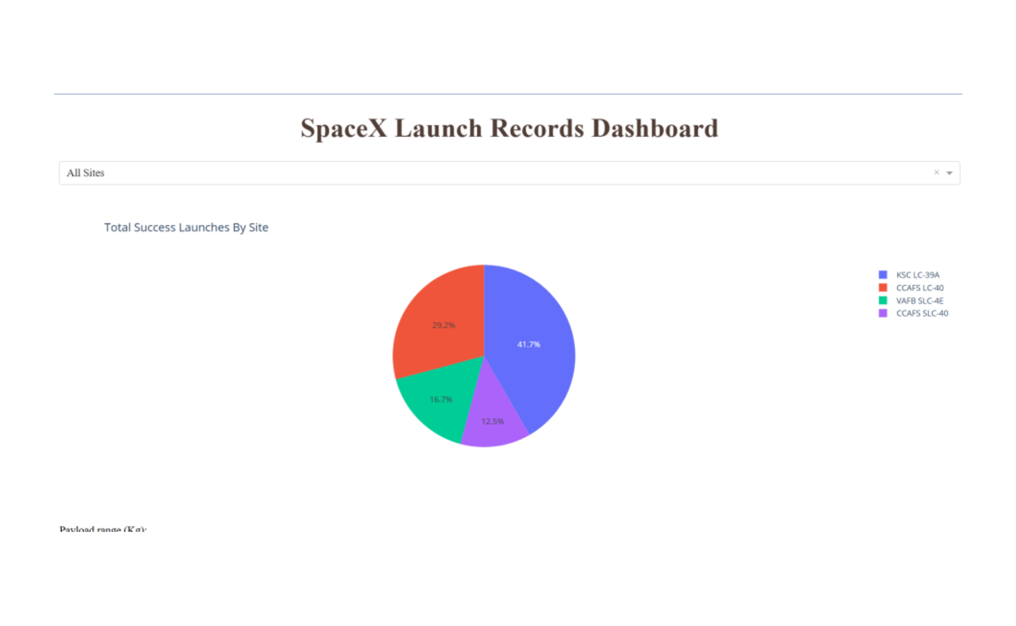
Input dataset (features + class label)
↓
Preprocess: one-hot encode + scale numeric features
↓
Train/Test split
↓
Train models (LR / SVM / DT / KNN)
↓
Hyperparameter tuning (GridSearchCV + CV)
↓
Evaluate (Accuracy + Confusion Matrix)
↓
Select best model + report final performance

Results

- Exploratory data analysis results
  - Visual EDA showed success improves with flight number / time and varies by orbit type, payload mass range, and launch site.
  - SQL EDA confirmed patterns through aggregation queries (distinct sites, payload stats, landing outcomes frequency, time filters).
- Interactive analytics demo
  - Folium: launch sites mapped + success/failure markers + proximity distance lines (coast/infrastructure).
  - Dash: interactive dropdown (site), pie chart (success/failure), payload range slider + scatter plot.
- Predictive analysis results
  - Compared LR/SVM/DT/KNN after preprocessing + GridSearchCV.
  - Reported model accuracy (and confusion matrix for the final chosen model).
  - Interpreted error types (false positives vs false negatives) to understand real-world impact.

```
Model Accuracy Comparison:
Logistic Regression: 0.8333333333333334
SVM: 0.8333333333333334
Decision Tree: 0.8333333333333334
KNN: 0.8333333333333334
```

Results

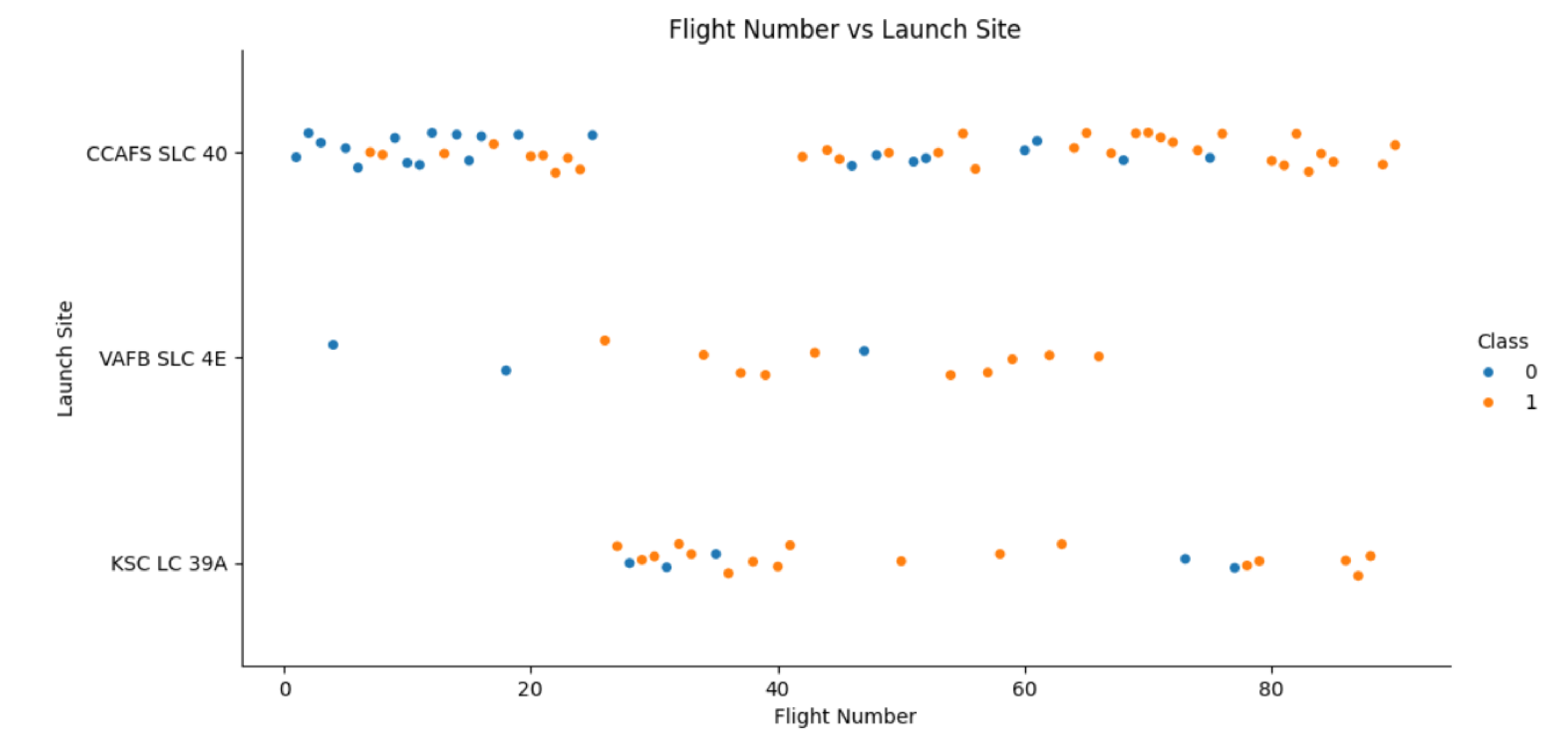

SpaceX Launch Records Dashboard
All Sites
Total Success Launches By Site
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40


Payload range (Kg):
0
2500
5000
7500
10000
Correlation between Payload and Success for all Sites
class
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT
B4
B5

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



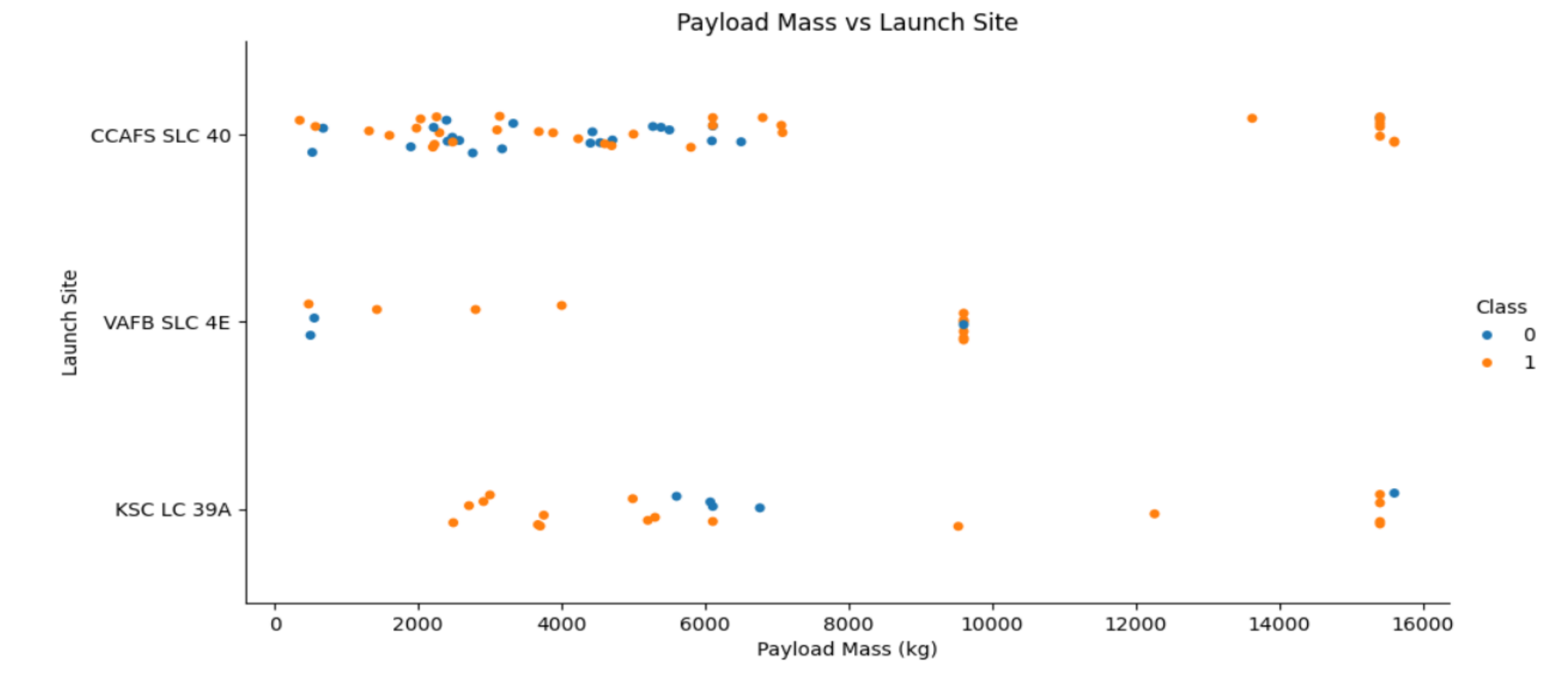
Flight Number vs. Launch Site

The scatter plot shows that early SpaceX launches were more likely to fail landing, especially at CCAFS SLC 40. As flight number increased, successful landings became more common. Launches from KSC LC 39A, which appear later in time, show a higher success rate, while VAFB SLC 4E has fewer launches and less consistent landing success.

Overall Patterns Observed

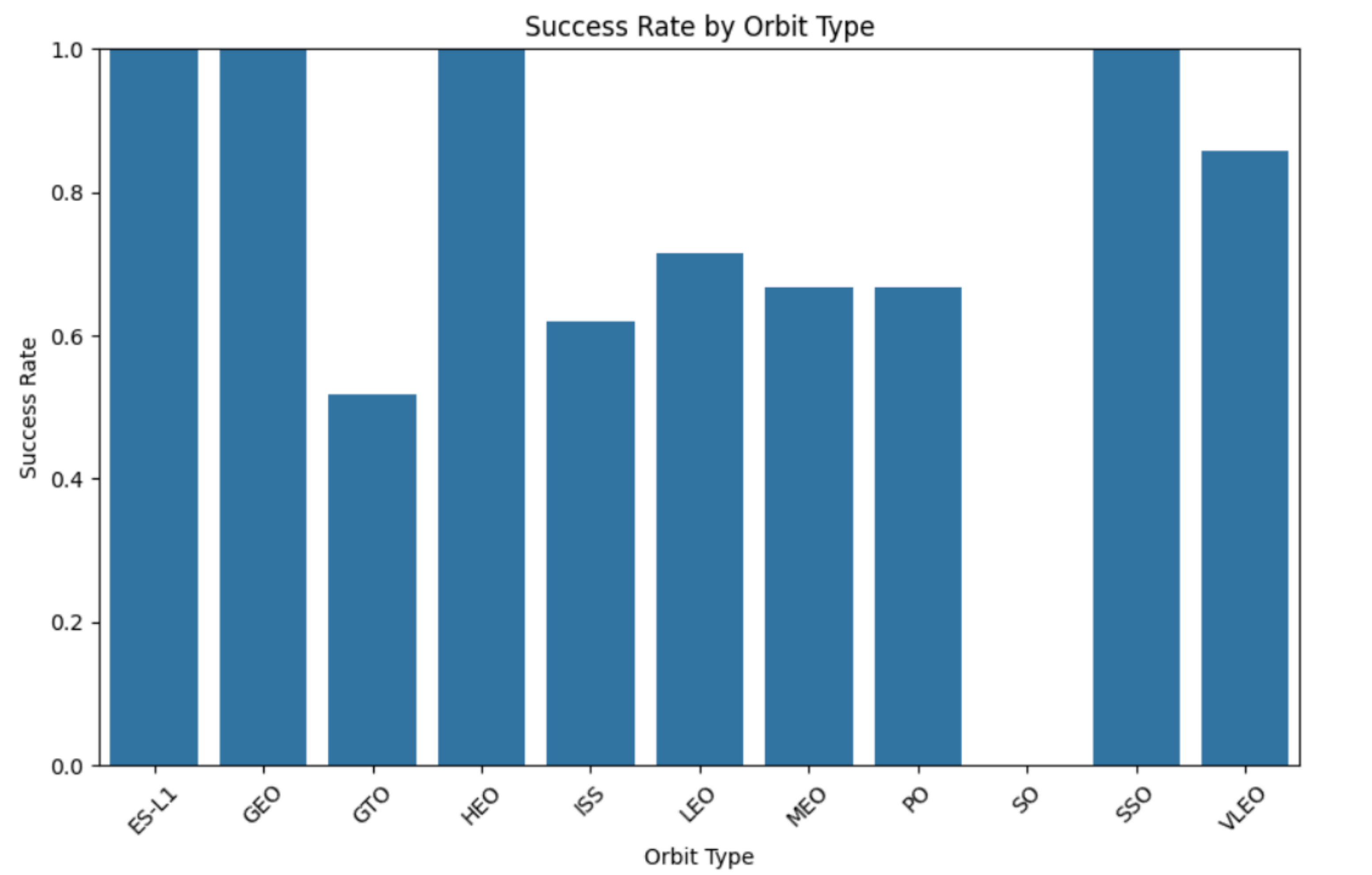
- Landing success increases with flight number
  - Early missions → more failures
  - Later missions → mostly successful
- Launch site matters
  - CCAFS SLC 40 shows clear improvement over time
  - KSC LC 39A has a high success rate from the start
  - VAFB SLC 4E shows mixed results
- Experience plays a major role As SpaceX gained experience, landing reliability improved across sites

Payload vs. Launch Site



Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



	Orbit	Class
0	ES-L1	1.000000
1	GEO	1.000000
2	GTO	0.518519
3	HEO	1.000000
4	ISS	0.619048
5	LEO	0.714286
6	MEO	0.666667
7	PO	0.666667
8	SO	0.000000
9	SSO	1.000000
10	VLEO	0.857143

Success rate varies significantly by orbit; **GTO has the lowest success (~0.52)** while several orbits appear **near 100%**. This suggests **orbit type is a key explanatory feature** and should be included in the predictive model, with attention to **sample size per orbit**.

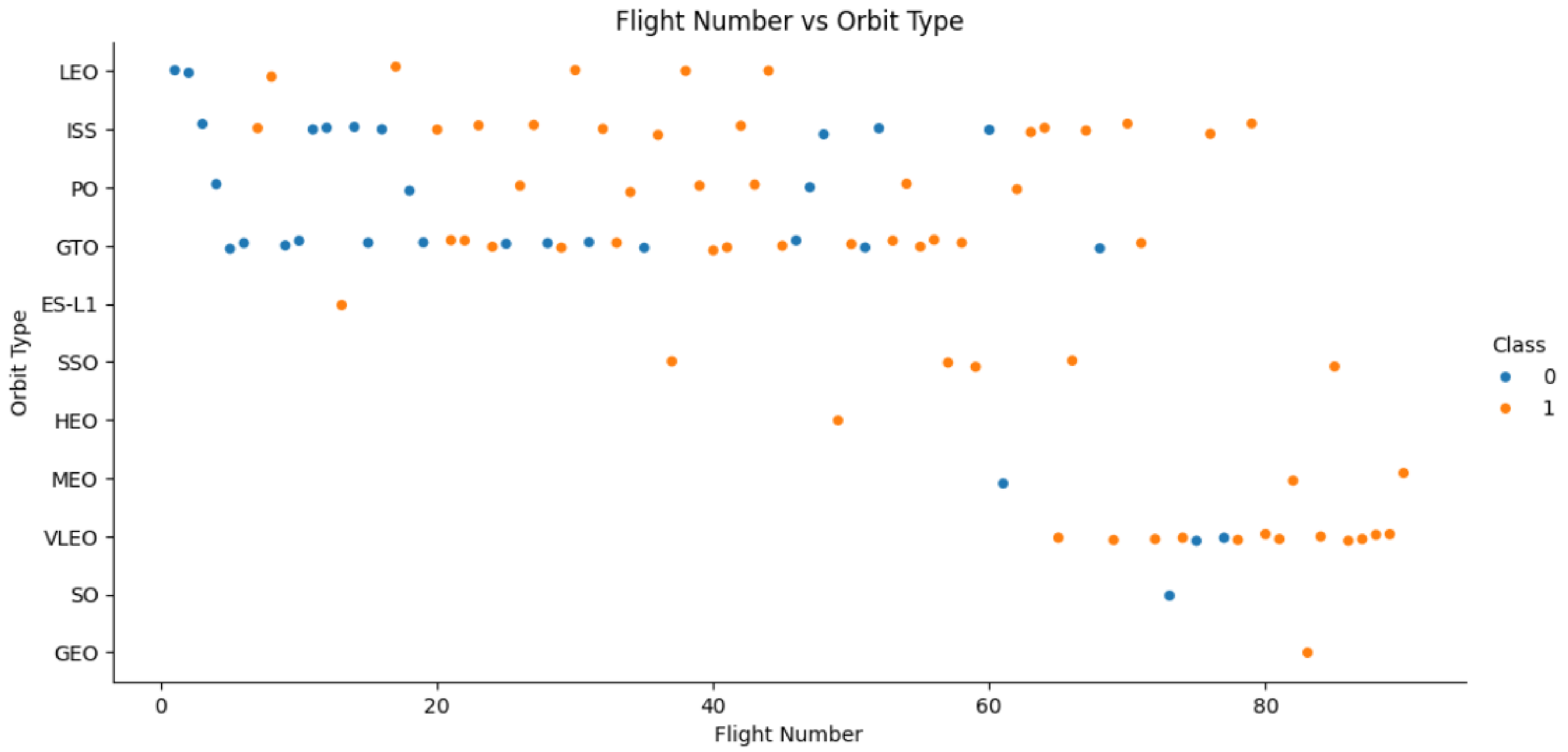
Success Rate vs. Orbit Type

Orbit type is a strong predictor of landing success because it reflects mission profile and difficulty:

- GTO missions typically require more energy (higher velocity / longer burns), which can leave less fuel margin for controlled booster recovery → lower landing success rate.
- LEO/ISS/MEO/PO tend to be more “routine” with more manageable recovery profiles → medium success rates.
- SSO / polar-style missions can still be very successful, especially as operations mature (and in your dataset it appears as 1.0).

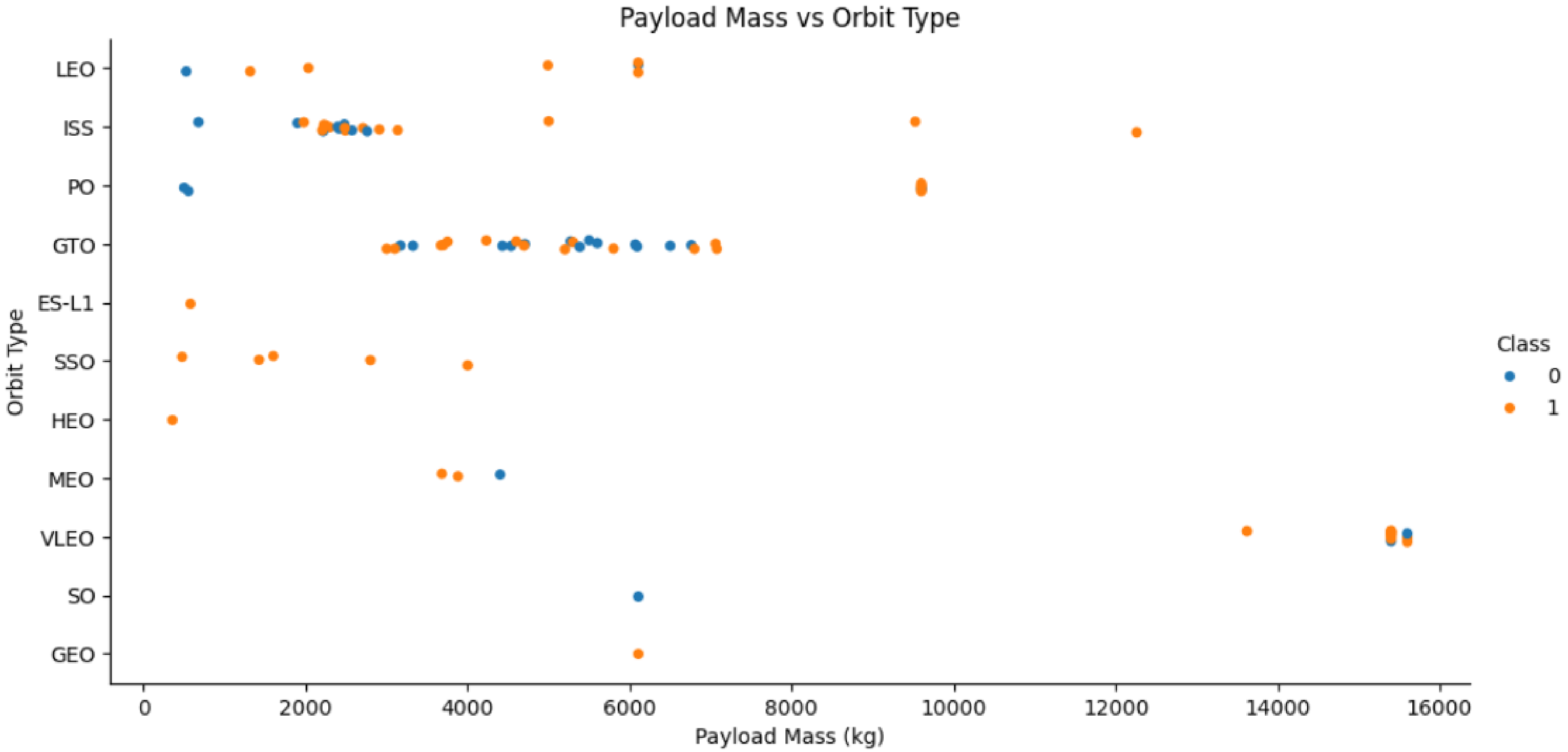
Some orbit categories may have small sample sizes, so a 1.00 or 0.00 success rate might be driven by few launches rather than a guaranteed pattern.

Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

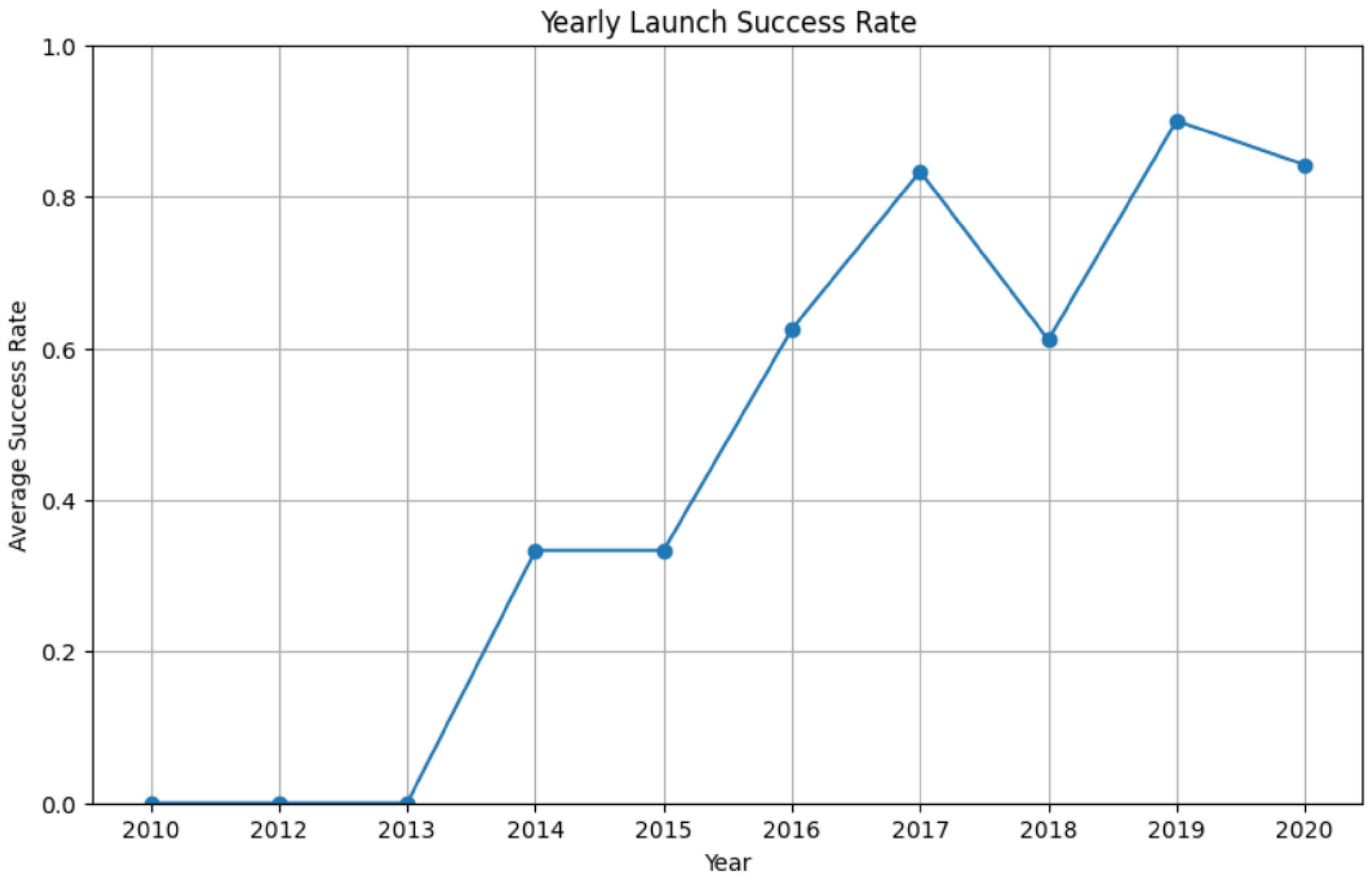
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

Task 1

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

```
%%sql
SELECT DISTINCT "Launch_Site"
FROM SPACEXTABLE;
```

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

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

This SQL query uses SELECT DISTINCT to list all unique launch sites in the dataset (SPACEXTABLE). The result shows there are four launch sites used in these missions: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E.

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

Total Payload Mass

In [34]:

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass
FROM SPACEXTABLE
WHERE "Customer" LIKE '%NASA%';
```

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

Out[34]:

Total_Payload_Mass
107010

This query calculates the total payload mass (kg) for missions where the Customer contains "NASA". Using SUM(PAYLOAD_MASS_KG_) with a WHERE Customer LIKE '%NASA%' filter, it shows NASA-related launches in this dataset carried 107,010 kg of payload in total.

Average Payload Mass by F9 v1.1

```
%%sql
SELECT
    AVG(COALESCE("PAYLOAD_MASS__KG_", 0)) AS Avg_Payload_Mass_KG
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

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

Avg_Payload_Mass_KG
2928.4

This query computes the average payload mass (kg) for launches using booster version “F9 v1.1”. It applies COALESCE(PAYLOAD_MASS_KG_, 0) to treat missing payload values as 0 so they don’t break the average calculation. The result shows an average payload of ≈ 2,928.4 kg for F9 v1.1 missions in this dataset.

First Successful Ground Landing Date

```
%%sql
SELECT MIN("Date") AS First_Success_Date
FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE 'Success%';
```

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

First_Success_Date
2015-12-22

This query finds the earliest date when a landing outcome was recorded as successful. It filters rows where Landing_Outcome starts with "Success" and returns the minimum Date. The first successful landing in this dataset occurred on 2015-12-22.

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
SELECT DISTINCT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

This query lists the unique booster versions that achieved “Success (drone ship)” landings and carried a payload mass between 4,000 and 6,000 kg. The result shows four boosters meeting these conditions: F9 FT B1022, F9 FT B1026, F9 FT B1021.2, and F9 FT B1031.2.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT "Mission_Outcome", COUNT(*) AS Count
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

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

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

This query groups the dataset by Mission_Outcome and counts how many launches fall into each outcome category. It shows missions are overwhelmingly successful (98 labeled “Success”), with very few failures (1 “Failure (in flight)”). The extra “Success” variants (e.g., “Success (payload status unclear)”) appear as separate rows because the text labels are slightly different.

Boosters Carried Maximum Payload

```
%%sql
SELECT DISTINCT "Booster_Version"
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
);
```

```
 * 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 finds the maximum payload mass in the dataset (using a subquery with MAX(PAYLOAD_MASS__KG_)) and then returns the distinct booster versions that carried a payload equal to that maximum value. The output shows multiple Falcon 9 Block 5 (F9 B5) boosters achieved the dataset's highest-payload missions (e.g., B1048.x, B1049.x, B1051.x, B1056.4, etc.).

2015 Launch Records

```
%%sql
SELECT
    substr("Date", 6, 2) AS Month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE substr("Date", 0, 5) = '2015'
  AND "Landing_Outcome" = 'Failure (drone ship)';
```

```
 * 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

This query filters the dataset to only 2015 launches (substr(Date,0,5)='2015') where the landing outcome was “Failure (drone ship)”. It also extracts the month from the date and returns the booster version and launch site for those failures. The results show two drone-ship landing failures in January (01) and April (04) of 2015, both from CCAFS LC-40 (boosters F9 v1.1 B1012 and F9 v1.1 B1015).

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

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

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

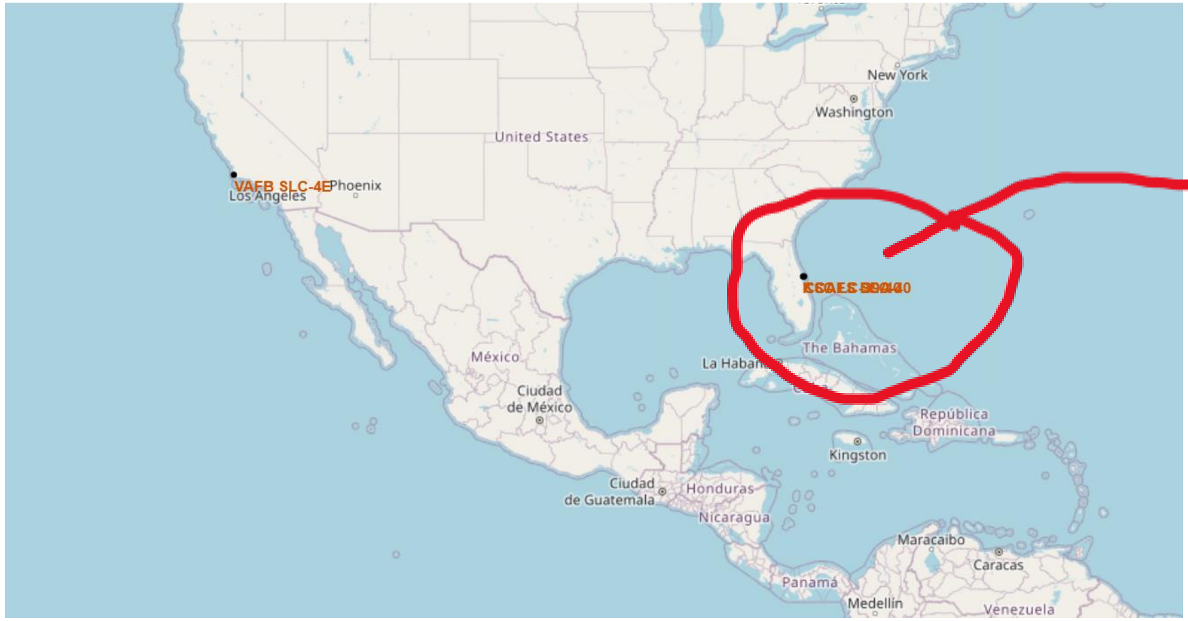
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

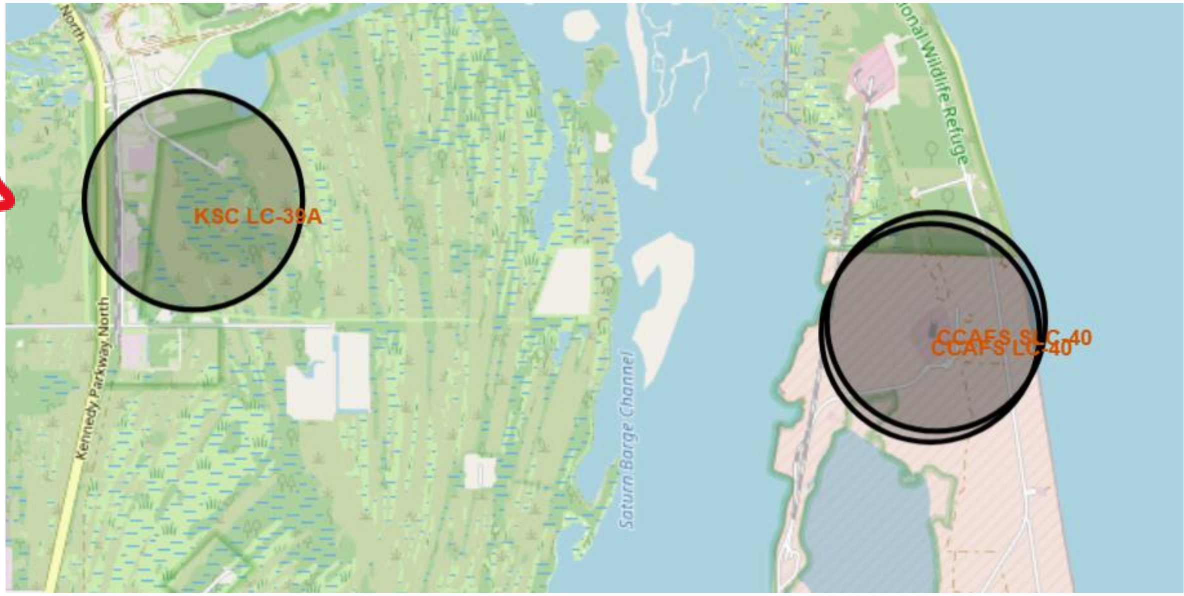
This query counts how often each landing outcome occurred between 2010-06-04 and 2017-03-20, then sorts the outcomes from most to least frequent. The results show “No attempt” is the most common outcome (10), followed by drone-ship landings with equal successes and failures (5 each). Ground pad successes appear less frequently (3), and a smaller number of missions ended in controlled/uncontrolled ocean landings or parachute failures highlighting that early missions often did not attempt landing, and recovery outcomes became more common later in the timeline.

Section 3

Launch Sites Proximities Analysis

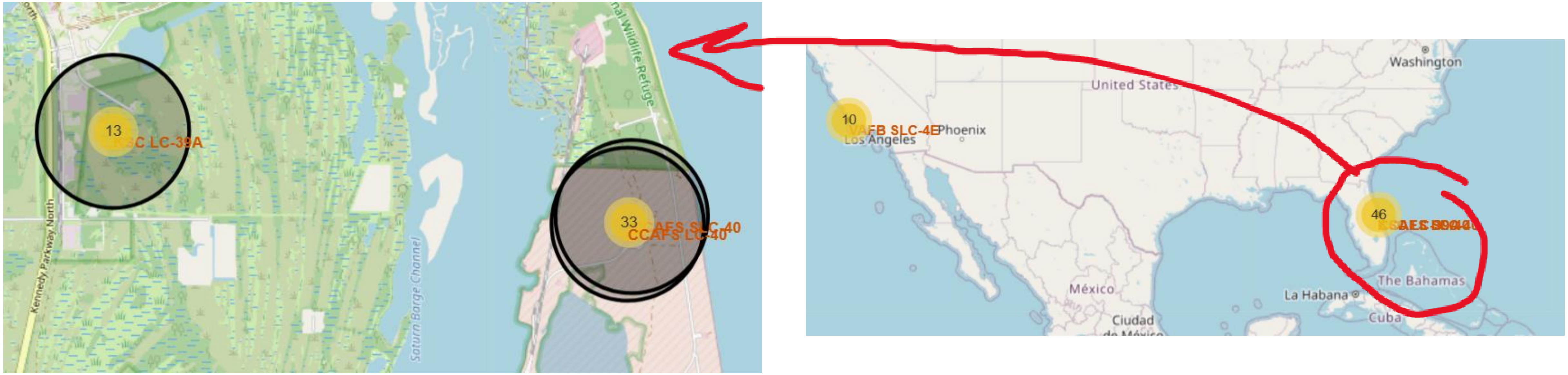
<Folium Map Screenshot 1>



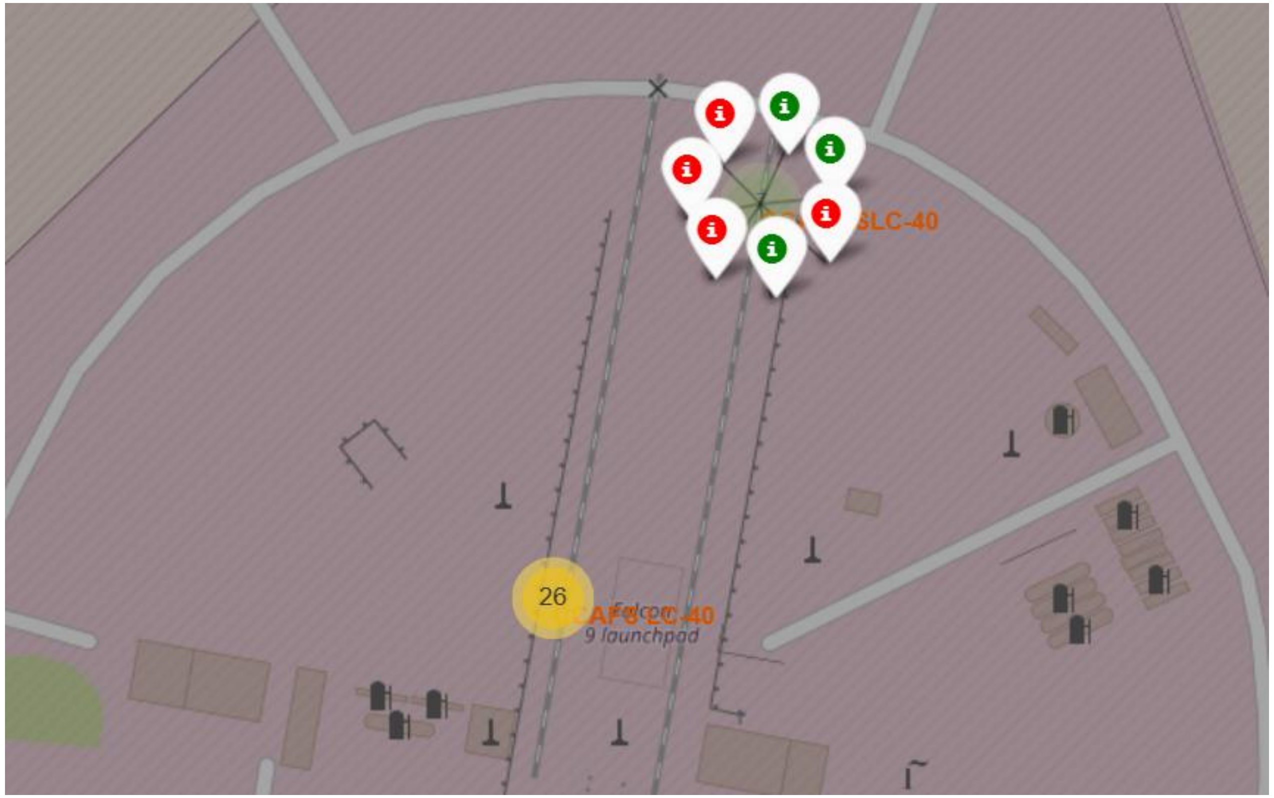


<Folium Map Screenshot 1>

Are all launch sites in proximity to the Equator line?
Answer: No.
Explanation (simple):
The Equator is at 0° latitude.
SpaceX launch sites are located roughly between 28°N and 35°N. This means they are far north of the Equator, not close to it.
Why this still works:
Launching closer to the Equator is helpful, but not mandatory.
These locations are chosen as a trade-off between:
Benefiting from Earth’s rotation
Safety
Existing infrastructure
Airspace and ocean access
Are all launch sites in very close proximity to the coast? Answer: Yes. Explanation (simple): All marked launch sites are located right next to the ocean. Examples: CCAFS & KSC → Atlantic Ocean VAFB → Pacific Ocean Why SpaceX prefers coastal sites: Safer disposal of rocket stages Easier recovery of boosters Fewer risks to populated areas Flexible launch trajectories over open water

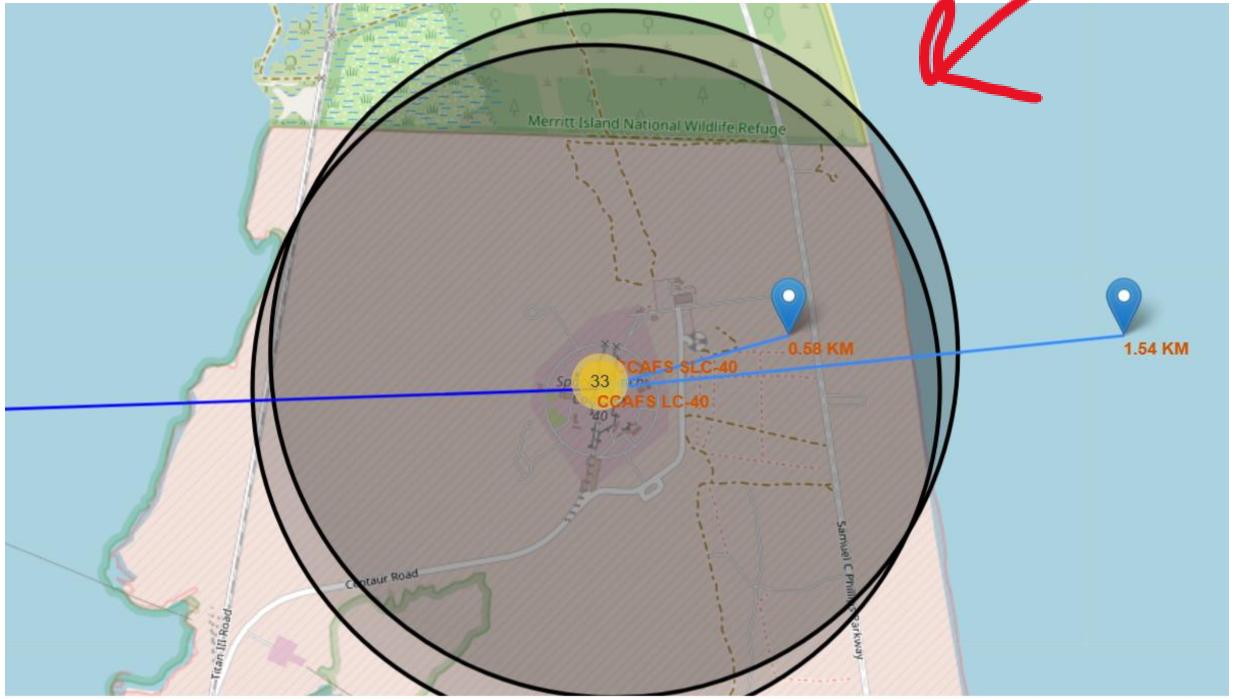


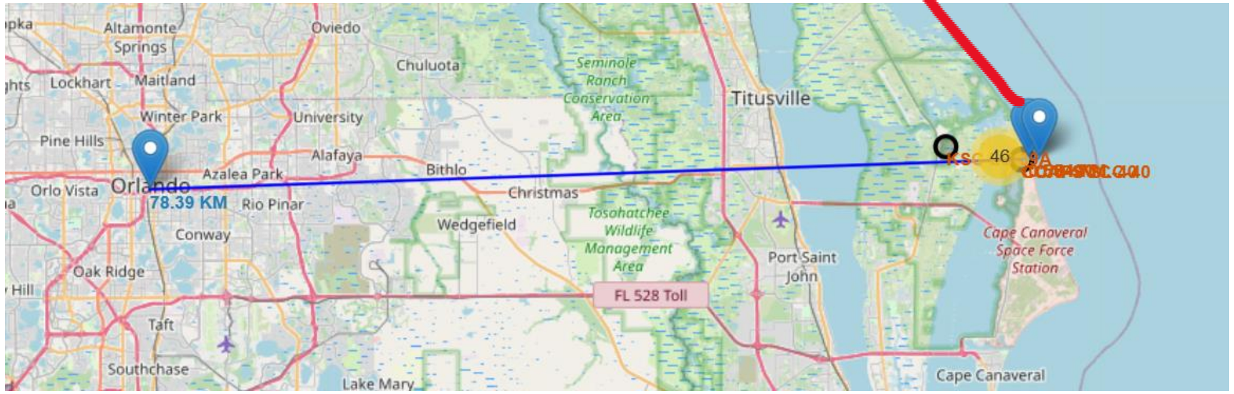
<Folium Map Screenshot 2>



<Folium Map Screenshot 2>

<Folium Map Screenshot 3>

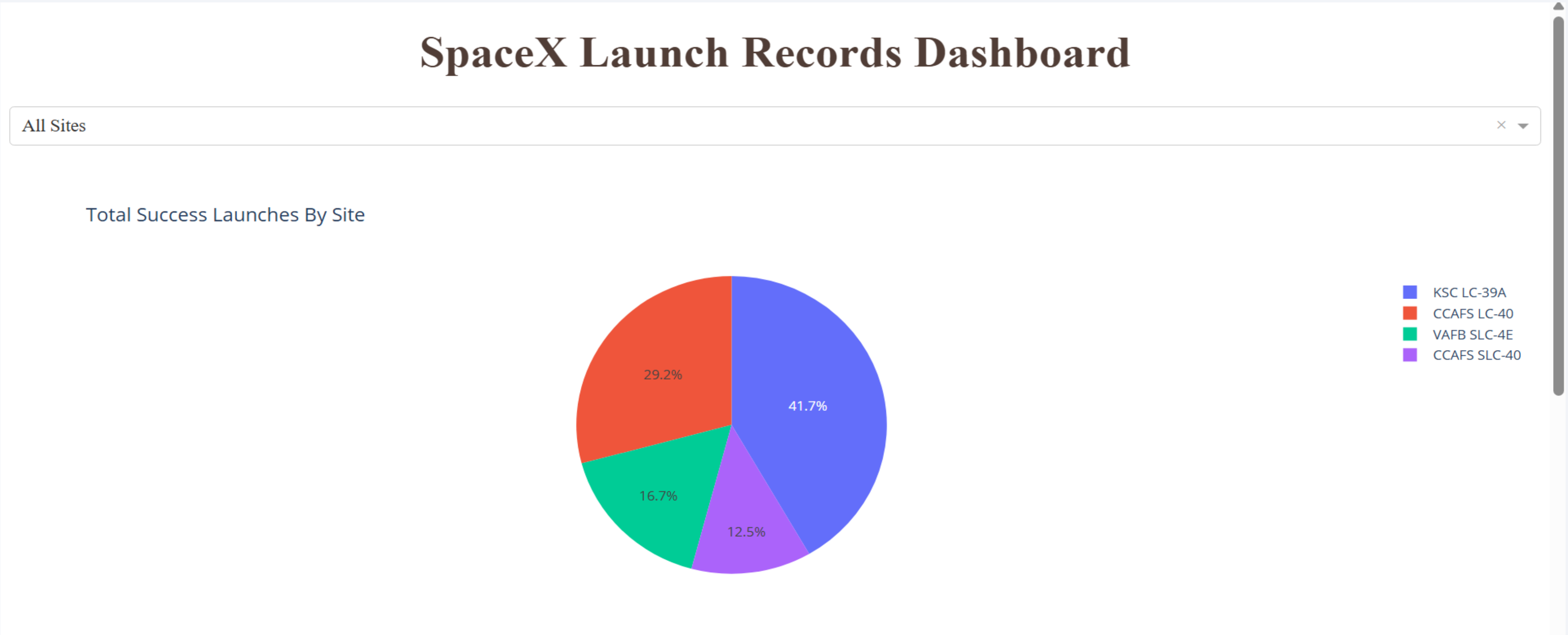




Section 4

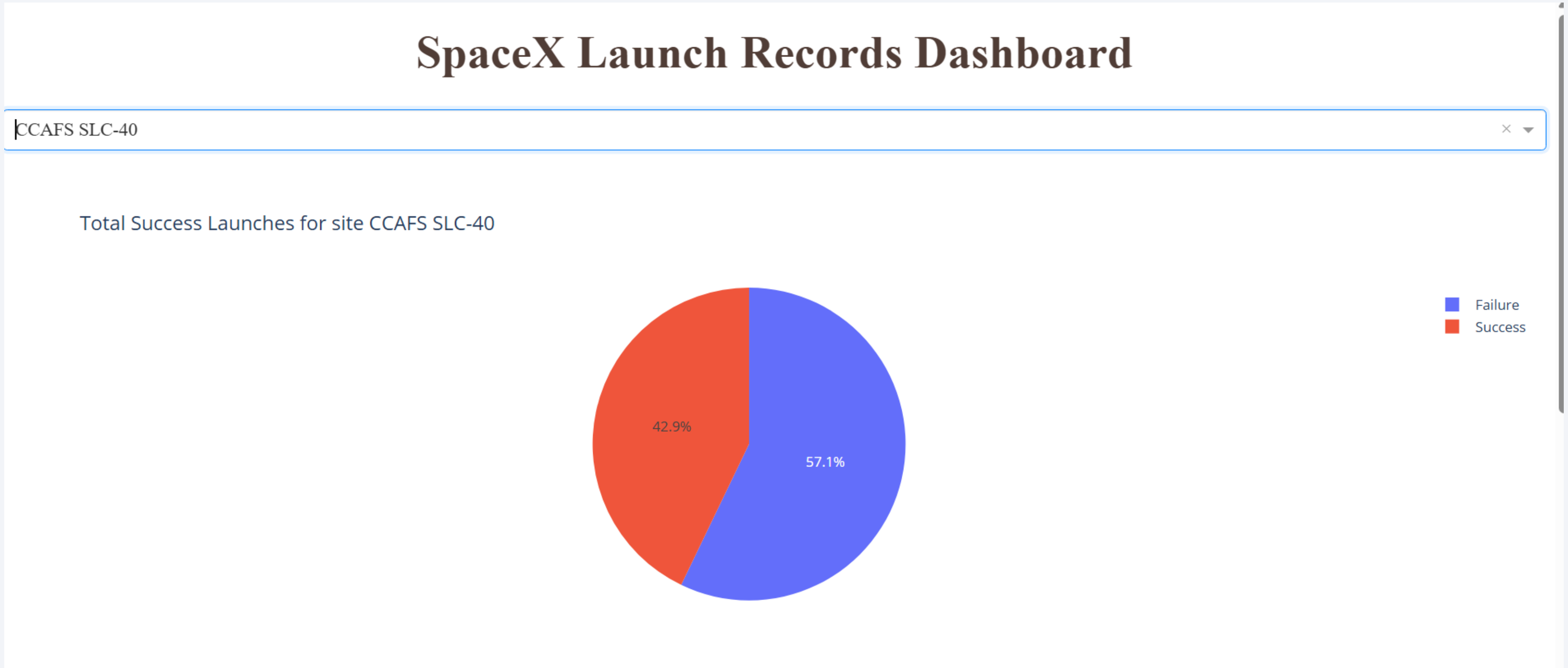
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



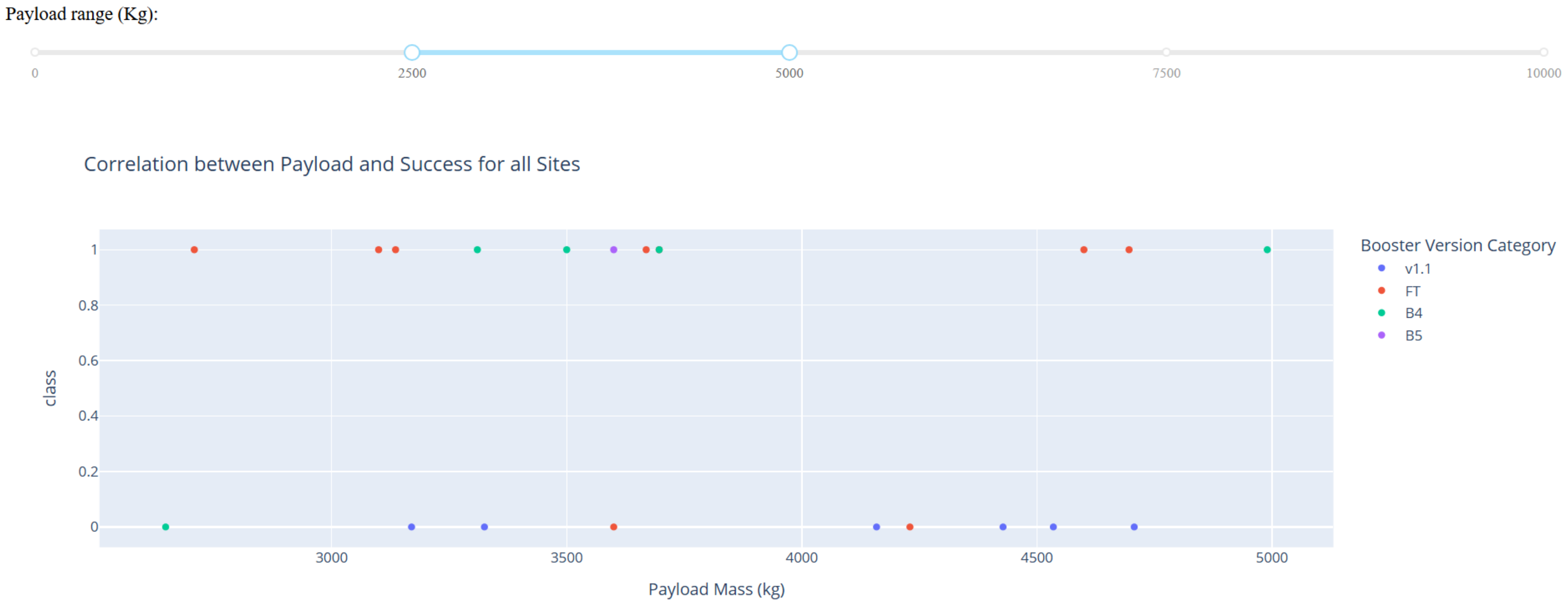
This dashboard view (set to All Sites) shows a pie chart of the distribution of successful launches by launch site. The largest share of successes comes from KSC LC-39A (~41.7%), followed by CCAFS LC-40 (~29.2%), while VAFB SLC-4E (~16.7%) and CCAFS SLC-40 (~12.5%) contribute smaller portions. Since the chart counts successes only (not total attempts), it highlights which sites produced the most successful launches in the dataset, but it does not by itself represent the success rate at each site.

<Dashboard Screenshot 2>



When filtering the dashboard to the selected launch site (shown here), the pie chart displays the success vs. failure breakdown for that site. In this example, CCAFS SLC-40 has more failures (57.1%) than successes (42.9%), indicating a lower historical success rate at this site compared with others and showing why it does not rank among the best-performing sites when evaluating landing outcomes.

<Dashboard Screenshot 3>

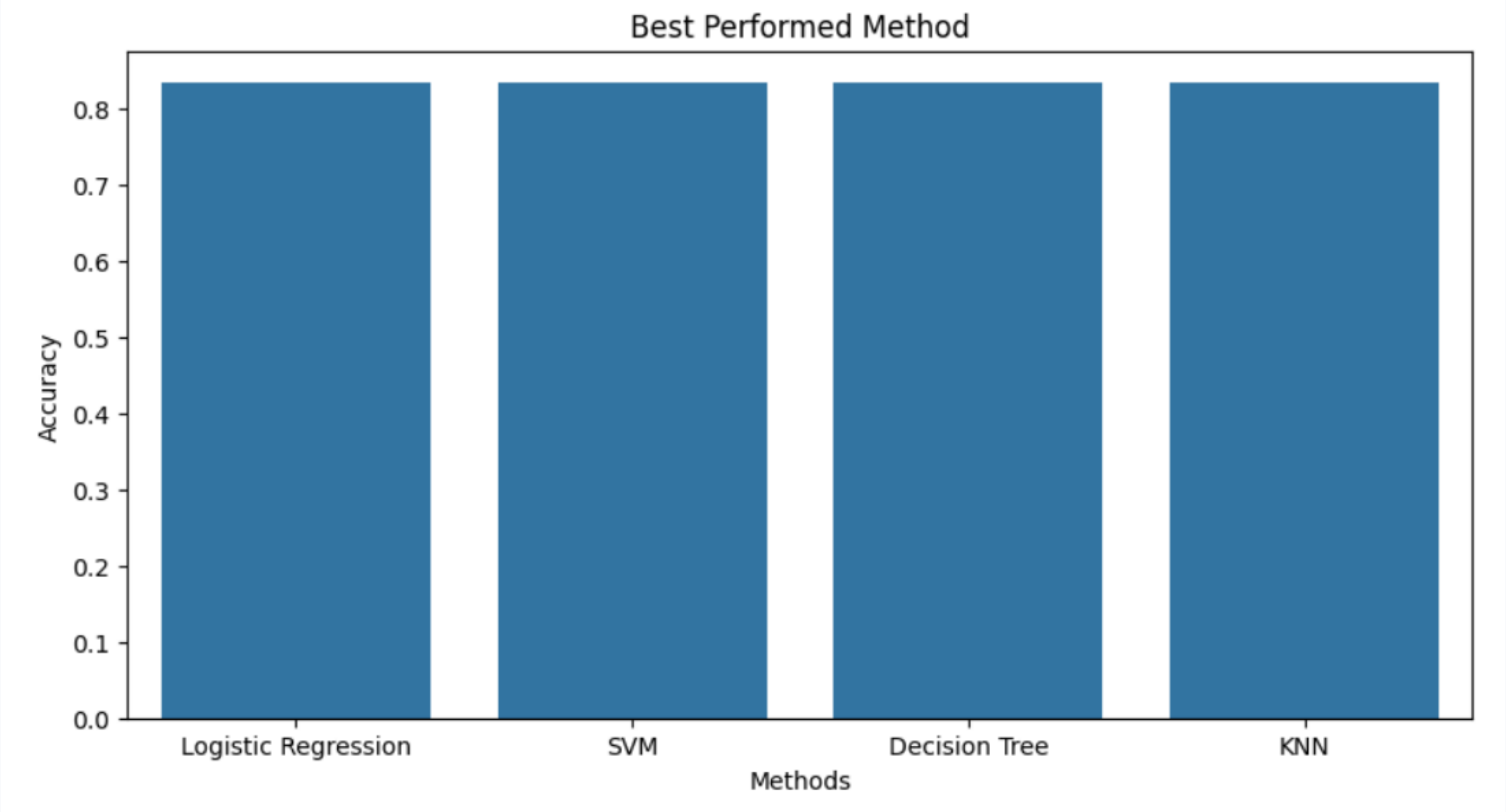


These two screenshots show the Dash scatter plot of Payload Mass (x-axis) vs Launch Outcome class (y-axis) for All Sites, colored by Booster Version Category, with the payload range slider used to filter which launches appear. In the full-range view (0–10,000 kg), most successful outcomes (class=1) cluster in the mid-payload region (roughly ~2,000–5,500 kg), while failures (class=0) are more visible at lower/mid payloads for older boosters (notably v1.0/v1.1). In the filtered view (~2,500–5,000 kg), the points show a clearer pattern: FT/B4/B5 missions in this mid-payload range are predominantly successful, whereas v1.1 includes several failures in the same range suggesting that newer booster versions (FT/B4/B5) achieved higher success rates and that the dashboard helps isolate payload windows where success is strongest.

Section 5

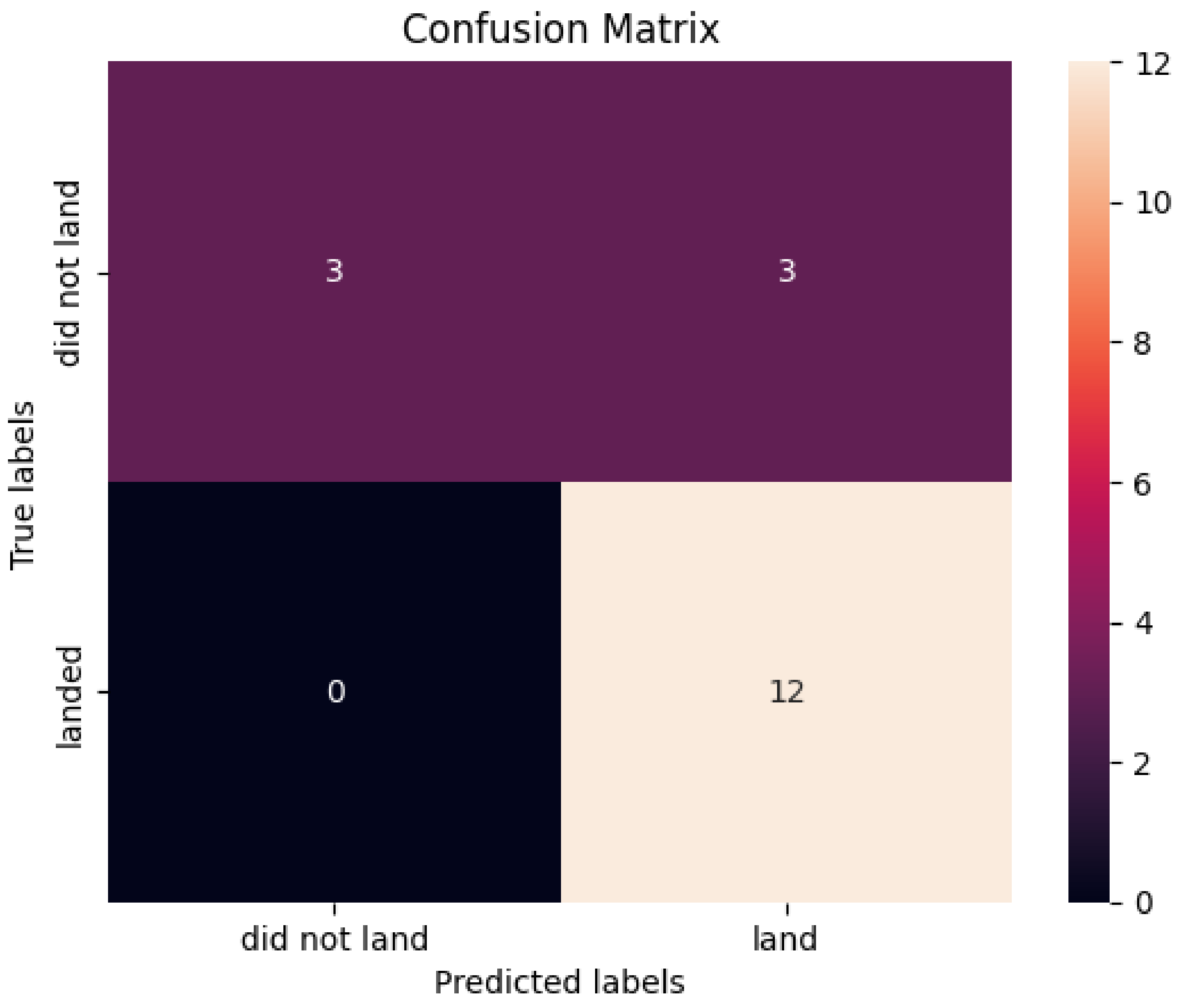
Predictive Analysis (Classification)

Classification Accuracy



From the bar chart, **all four models (Logistic Regression, SVM, Decision Tree, and KNN) achieve essentially the same highest test accuracy (~0.83)**. So there isn't a single unique "best" model here — **they are tied for highest classification accuracy** (≈ 0.833).

Confusion Matrix



Conclusions

- All tested classifiers (Logistic Regression, SVM, Decision Tree, KNN) achieved a very similar test accuracy (~0.83), meaning the dataset's predictive signal is captured consistently across different model families.
- Because performance is tied, the best choice is the simplest + most interpretable model (typically Logistic Regression) unless you have a specific reason to prefer another model.
- The similar scores suggest either (1) the dataset is relatively small, (2) features have limited complexity, or (3) the problem is close to a performance "ceiling" with the available variables.
- The model results confirm that features like orbit, payload, launch site, and booster version are useful predictors of landing success, but accuracy can likely be improved with more data and additional features (like weather, booster reuse count, mission energy/trajectory).

Appendix

- All relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that have created during this project are available on my GitHub
- Link to my GitHub : Sourena-Mohit/Winning-Space-Race-With-Data-Science: Applied Data Science Capstone Project for IBM Data Science Professional Certificate

Thank you!