IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abraham P
28-Aug-2025

Outline

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

Executive Summary

Methodologies Used

- Data Collection: Scraped Falcon 9 launch data from Wikipedia using requests and BeautifulSoup.
- Data Wrangling: Cleaned and structured launch records, handled missing values, and engineered features for modeling.
- EDA & Visualization: Explored launch success patterns using seaborn, matplotlib, and SQL queries; built interactive visuals with Plotly Dash.
- Geospatial Analysis: Mapped launch sites and proximity factors using Folium and geopy.
- Predictive Modeling: Tuned and evaluated four classifiers — Logistic Regression, SVM, Decision Tree, and KNN — using GridSearchCV and confusion matrices

Methodologies Used

Model	CV Accuracy	Test Accuracy	Best Parameters
Logistic Regression	84.6%	83.33%	C=0.01, penalty='l2'
SVM (Sigmoid Kernel)	84.8%	83.33%	kernel='sigmoid', C=1.0, gamma≈0.0316
KNN (p=1, k=10)	84.8%	83.33%	n_neighbors=10, p=1
Decision Tree	87.7%	61.11%	max_depth=16, splitter='random'

- Best Performing Models: Logistic Regression, SVM, and KNN — all consistent and generalizable.
- Dashboard: Built with Plotly Dash to visualize launch success interactively.
- Map: Folium-based interactive map showing launch site locations and proximity insights.

Introduction

Project Background & Context

- SpaceX Falcon 9 Launches represent a major shift in commercial spaceflight, with reusable boosters and frequent missions.
- The dataset, scraped from Wikipedia’s Falcon 9 launch records, includes launch dates, payloads, sites, booster versions, and landing outcomes.
- This project explores how data science can uncover patterns in launch success and booster recovery — two critical metrics for cost efficiency and mission reliability.

Problems to Solve

- What factors most influence launch success? - Is it payload mass, launch site, booster version, or orbit type?
- Can we predict launch outcomes using classification models? - How well do Logistic Regression, SVM, Decision Tree, and KNN perform?
- Are certain launch sites or booster versions more reliable? - Use geospatial mapping to visualize success clusters.
- How can we present these insights interactively? - Build a dashboard that allows users to explore launch patterns and model predictions.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Scraped Falcon 9 launch records from Wikipedia using requests and BeautifulSoup.
  - Parsed HTML tables to extract launch dates, payloads, booster versions, landing outcomes, and orbit types.
- Perform data wrangling
  - Cleaned inconsistent entries, handled missing values, and standardized categorical variables.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Applied four classification models: Logistic Regression, SVM (Sigmoid), Decision Tree, KNN
  - Used GridSearchCV for hyperparameter tuning
  - Evaluated models using: Cross-validation accuracy, Test accuracy and Confusion matrices for precision and recall

Data Collection

- Source: Wikipedia’s “List of Falcon 9 launches” page (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)
- Tools Used: requests, BeautifulSoup, pandas
- Method: Web scraping of HTML tables containing launch records

Data Extracted:

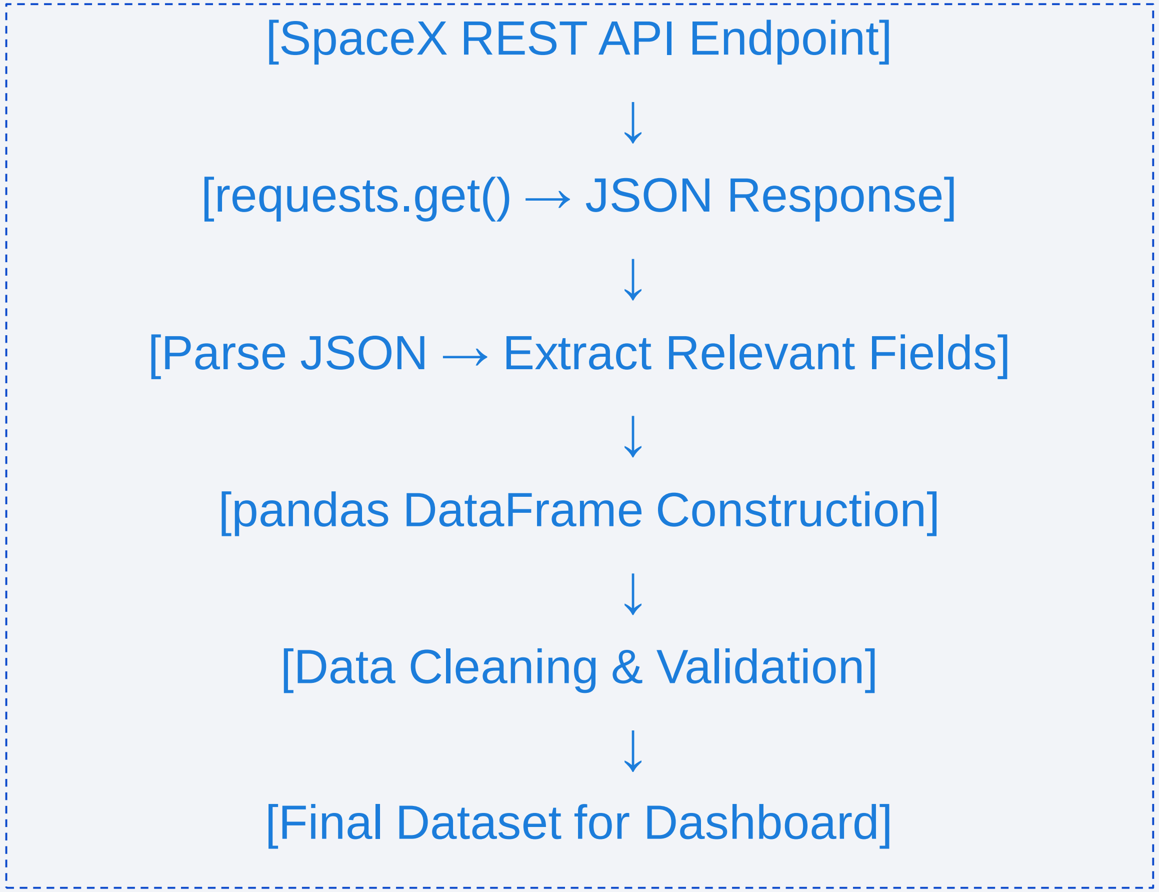
- ➢ Launch Date
- ➢ Launch Site
- ➢ Payload Mass
- ➢ Orbit Type
- ➢ Booster Version
- ➢ Landing Outcome

Data Collection – SpaceX API

- Source: SpaceX Public REST API (https://api.spacexdata.com/v4/launches)
- Tools Used: requests, json, pandas
- Method: RESTful API calls to retrieve structured launch data

Data Extracted:

- Launch ID, Name, Date
- Rocket Type
- Launch Site
- Payload Mass
- Success Status
- GitHub Notebook (Completed API Calls + Output):
- 🔗 View on GitHub



[SpaceX REST API Endpoint]

↓

[requests.get() → JSON Response]

↓

[Parse JSON → Extract Relevant Fields]

↓

[pandas DataFrame Construction]

↓

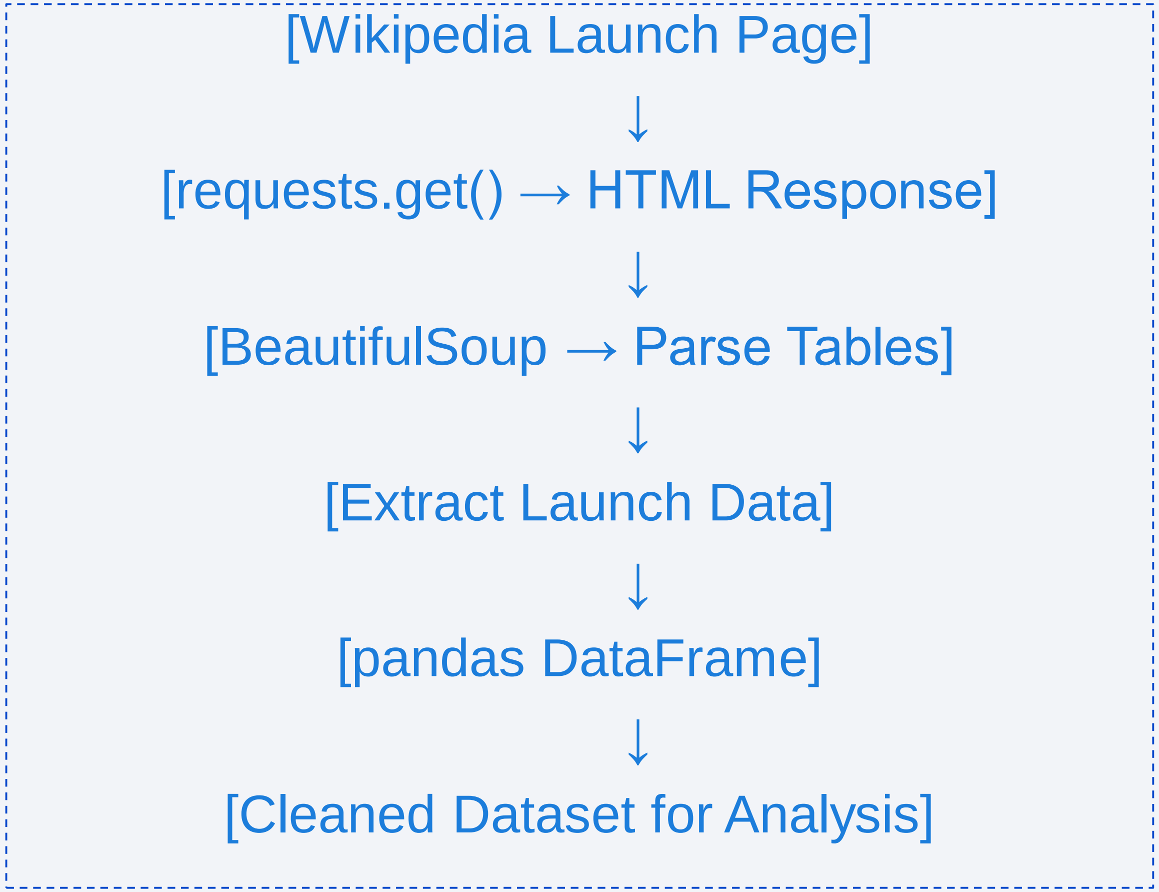
[Data Cleaning & Validation]

↓

[Final Dataset for Dashboard]

Data Collection - Scraping

- Source: Wikipedia’s “List of Falcon 9 launches” page (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)
- Tools Used: requests, BeautifulSoup, pandas
- Process:
- Sent HTTP GET request to retrieve HTML content
- Parsed HTML tables using BeautifulSoup
- Extracted launch records: date, site, payload, booster version, orbit, landing outcome
- Converted parsed data into a structured pandas DataFrame



[Wikipedia Launch Page]

↓

[requests.get() → HTML Response]

↓

[BeautifulSoup → Parse Tables]

↓

[Extract Launch Data]

↓

[pandas DataFrame]

↓

[Cleaned Dataset for Analysis]

GitHub Notebook (Web Scraping Code + Output):

- 🔗 View on GitHub

Data Wrangling

Initial Format: Raw HTML tables with inconsistent structure

Cleaning Steps:

- ➢ Removed null and irrelevant rows
- ➢ Standardized column names and formats (e.g., date, orbit, landing outcome)
- ➢ Converted categorical values to consistent labels (e.g., “Success” vs “Booster landed”)
- ➢ Extracted numeric features (e.g., payload mass, flight number)
- ➢ Handled missing values using imputation and logical inference

Feature Engineering:

- ➢ Created binary flags for landing success
- ➢ Derived booster version categories
- ➢ Added launch site proximity metrics (for later geospatial analysis)

GitHub Notebook (Data Wrangling Code + Output):

- 🔗 View on GitHub

EDA with Data Visualization

Methodologies Used

Chart Title	Purpose
Payload Mass vs. Flight Number (Class-colored)	To observe payload trends across missions and correlate with success/failure.
Flight Number vs. Launch Site (Landing Outcome)	To compare landing outcomes across different launch sites over time.
Payload Mass vs. Launch Site (Landing Outcome)	To analyze how payload weight and launch site influence landing success.
Success Rate by Orbit Type (Bar Chart)	To evaluate which orbit types are most reliable for successful missions.
Flight Number vs. Orbit Type (Landing Outcome)	To track how orbit choices evolved and affected landing outcomes.
Payload Mass vs. Orbit Type (Landing Outcome)	To assess how payload weight interacts with orbit type and mission success.
Yearly Launch Success Trend (Line Chart)	To visualize SpaceX's improvement in launch reliability over the years.

GitHub Notebook – EDA with Data Visualization

🔗 View on GitHub

EDA with SQL

Summary of SQL Queries Performed

- Task 1: Retrieved unique launch site names SELECT DISTINCT "Launch_Site" FROM SPACEXDATA
- Task 2: Displayed first 5 missions from launch sites starting with “CCA” SELECT * FROM SPACEXTABLE WHERE "LaunchSite" LIKE 'CCA%' LIMIT 5
- Task 3: Calculated total payload mass for NASA (CRS) missions SELECT SUM("PayloadMass") FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)'
- Task 4: Computed average payload mass for booster version F9 v1.1 SELECT AVG("PayloadMass") FROM SPACEXTABLE WHERE "BoosterVersion" = 'F9 v1.1'
- Task 5: Identified first successful ground pad landing date SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)' ➤ Result: 2015-12-22
- Task 6: Listed boosters with successful drone ship landings carrying 4000–6000 kg payloads SELECT DISTINCT "BoosterVersion" WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PayloadMass" BETWEEN 4000 AND 6000
- Task 7: Counted mission outcomes SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome"
- Task 8: Found booster version used for the heaviest payload SELECT DISTINCT "BoosterVersion" WHERE "PayloadMass" = (SELECT MAX("PayloadMass"))
- Task 9: Filtered 2015 drone ship failures by month, booster, and site SELECT substr("Date", 6, 2), "Landing_Outcome", "BoosterVersion", "LaunchSite" WHERE "Landing_Outcome" LIKE '%Failure%' AND "Landing_Outcome" LIKE '%drone ship%' AND substr("Date", 0, 5) = '2015'
- Task 10: Aggregated landing outcomes between 2010–2017 SELECT "Landing_Outcome", COUNT(*) WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome"

GitHub Notebook – EDA with SQL

🔗 [View on GitHub]

Build an Interactive Map with Folium

Summary of SQL Queries Performed

Map Objects Created

❑ Markers → Plotted each launch record with success/failure outcome → Used color-coded icons: green for success, red for failure → Added popups showing launch site and outcome

❑ Circle Markers → Highlighted fixed launch site locations → Used larger radius and labels for visual emphasis

❑ DivIcon Labels → Annotated proximity points (e.g., coastline, city) with distance values → Styled with HTML for clarity and contrast

❑ PolyLines → Drew lines between launch sites and proximity features (coastline, city, infrastructure) → Visualized spatial relationships and supported distance calculations

❑ Marker Clusters → Grouped overlapping launch records for cleaner visualization → Enabled interactive exploration of dense launch zones

Why These Objects Were Added

❑ To visualize launch outcomes spatially and interactively

❑ To highlight geographic patterns in launch success

❑ To analyze proximity factors like distance to coastlines, cities, and infrastructure

❑ To enhance storytelling in the dashboard with annotated visuals

❑ To support peer review with reproducible, interactive geospatial analysis

GitHub Notebook – Folium Interactive Mapping

🔗 View on GitHub

Build a Dashboard with Plotly Dash

Plots and Interactions Included

- Launch Site Dropdown (dcc.Dropdown)
- Allows users to select either All Sites or a specific launch site
- Dynamically updates both pie and scatter plots based on selection
- Success Pie Chart (dcc.Graph)
- Displays total successful launches across all sites
- If a specific site is selected, shows Success vs. Failure breakdown for that site
- Payload Range Slider (dcc.RangeSlider)
- Enables users to filter data by payload mass range
- Refines scatter plot to show only launches within selected payload interval
- Payload vs. Outcome Scatter Plot (dcc.Graph)
- Visualizes correlation between payload mass and launch success
- Color-coded by booster version category
- Includes hover tooltips for launch site and booster details

Purpose Behind These Features

- To explore launch performance across different SpaceX sites
- To analyze success patterns based on payload mass and booster versions
- To demonstrate interactive filtering using Dash callbacks
- To build a professional-grade dashboard for data storytelling and peer review
- To enhance user experience with intuitive controls and responsive visuals

GitHub Repository for Peer Review

🔗 View on GitHub

Predictive Analysis (Classification)

Model Development Summary

- Data Preparation
  - Cleaned and encoded categorical features (Orbit, LaunchSite, BoosterVersion)
  - Scaled numerical features using StandardScaler
  - Split into training and test sets (80/20)
- Model Training & Tuning
  - Trained multiple classifiers:
  - Logistic Regression (GridSearchCV, L2 regularization)
  - Support Vector Machine
  - K-Nearest Neighbors
  - Decision Tree (tuned with criterion='gini', max_depth=16, splitter='random')
- Evaluation Metrics
  - Accuracy, Precision, Recall, F1-Score
  - Confusion matrix visualizations
  - 10-fold cross-validation for robustness

Predictive Analysis (Classification)

Model Comparison

Model	Test Accuracy	False Positives	False Negatives
Logistic Regression	83.3%	3	0
SVM	83.3%	3	0
KNN	83.3%	3	0
Decision Tree	61.1%	4	3

- Final Selection
  - Best Performing Models: Logistic Regression, SVM, KNN
  - All achieved 83.3% test accuracy
  - Zero false negatives: correctly predicted every actual landing
  - Only 3 false positives: misclassified half of the “did not land” cases
  - Chosen for reliability, interpretability, and generalization

Predictive Analysis (Classification)

Flowchart of Classification Workflow

Data Preprocessing → Feature Engineering → Train/Test Split

↓

Model Training → Hyperparameter Tuning → Evaluation

↓

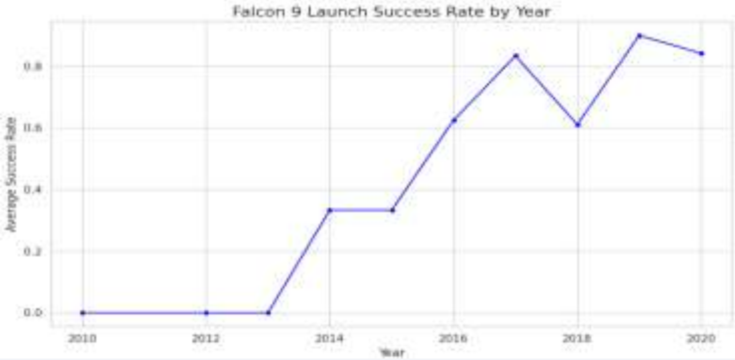
Confusion Matrix Analysis → Best Model Selection

GitHub Repository (Completed Lab & Codebase)

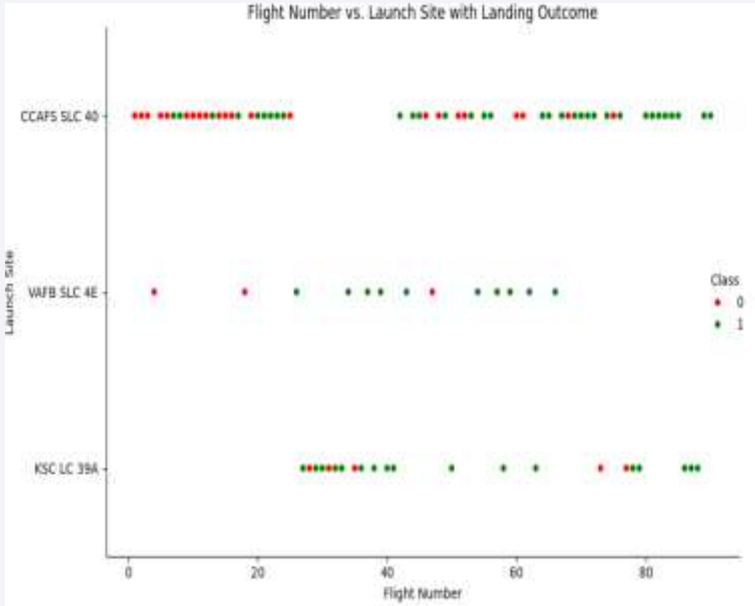
🔗 View on GitHub

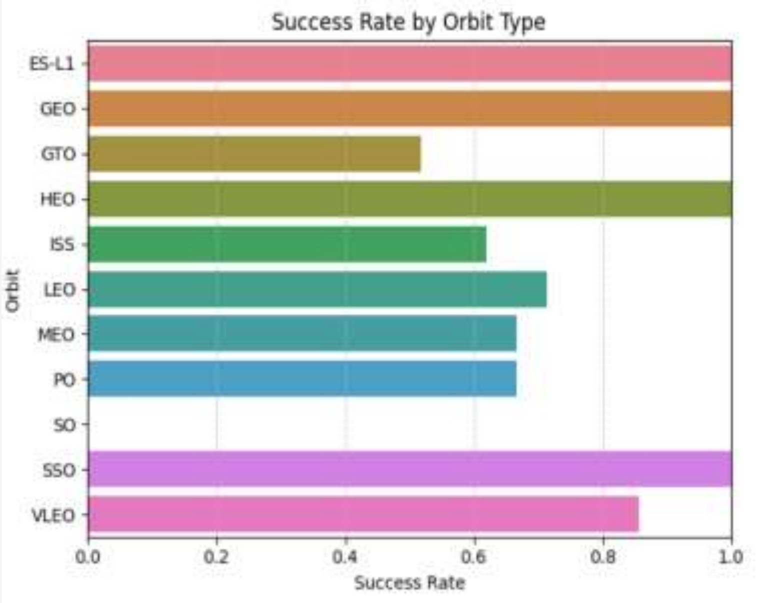
Results

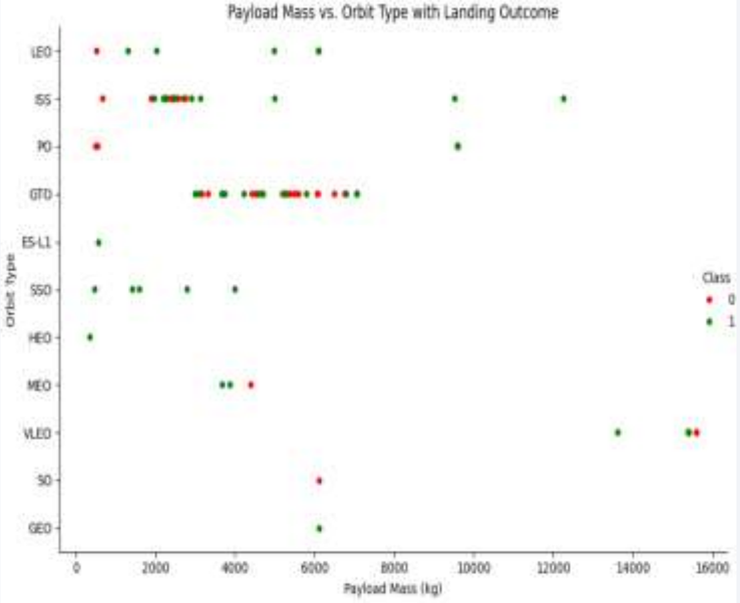
Exploratory Data Analysis Results



- Launch success rates vary significantly by launch site and booster version.
- Payload mass shows a nonlinear relationship with landing success.
- Orbit type influences landing probability — LEO missions tend to succeed more.
- Heatmaps and pairplots revealed strong correlations between payload, orbit and success.

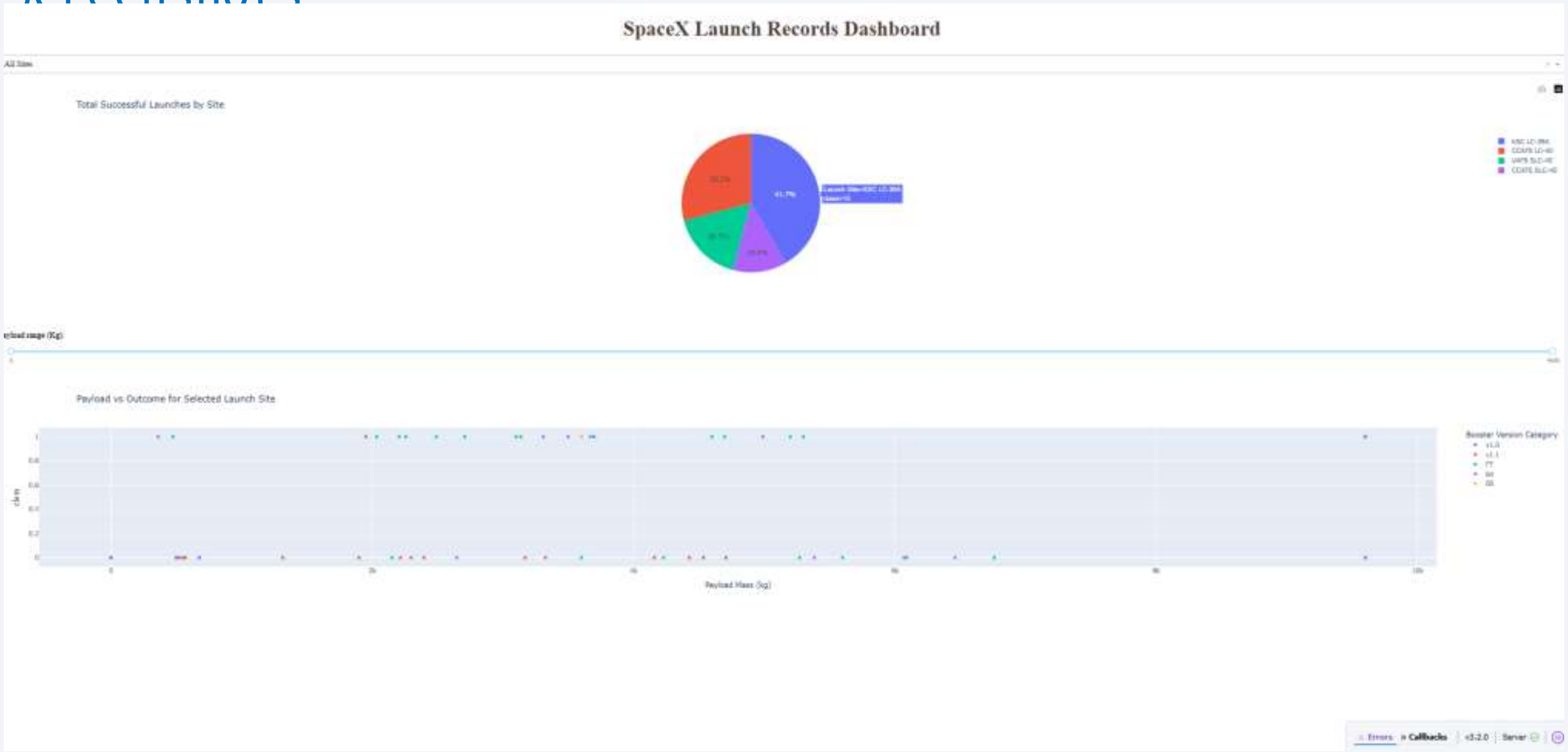






Results

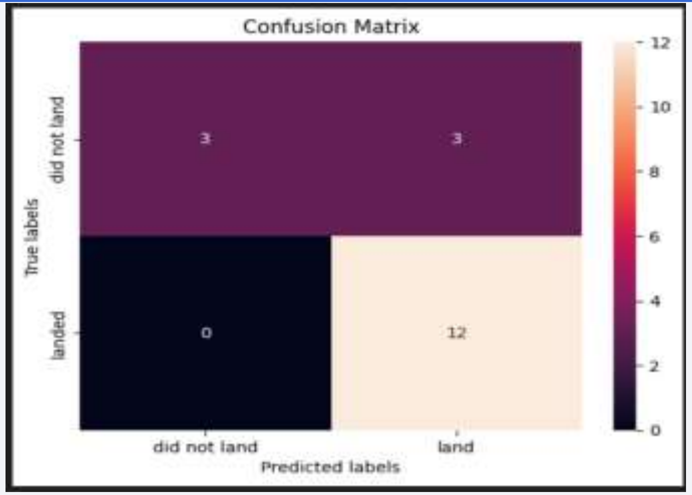
Interactive Analytics Demo in Screenshots

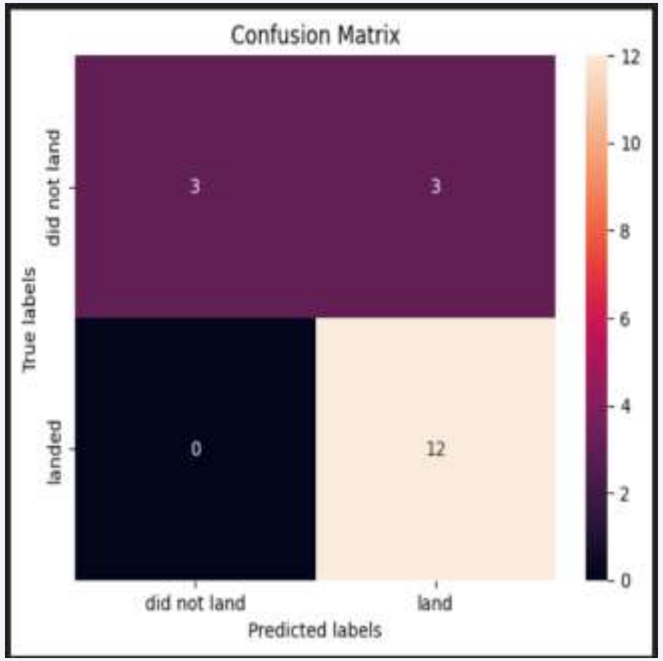


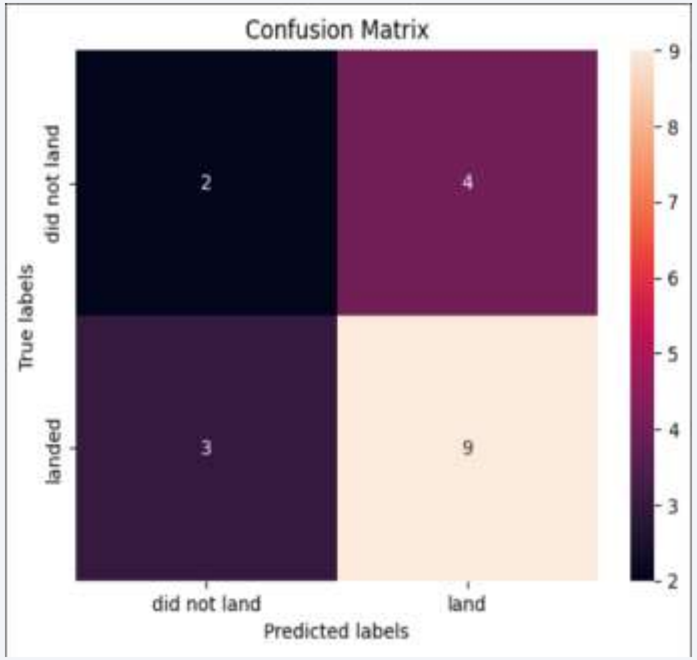
Results

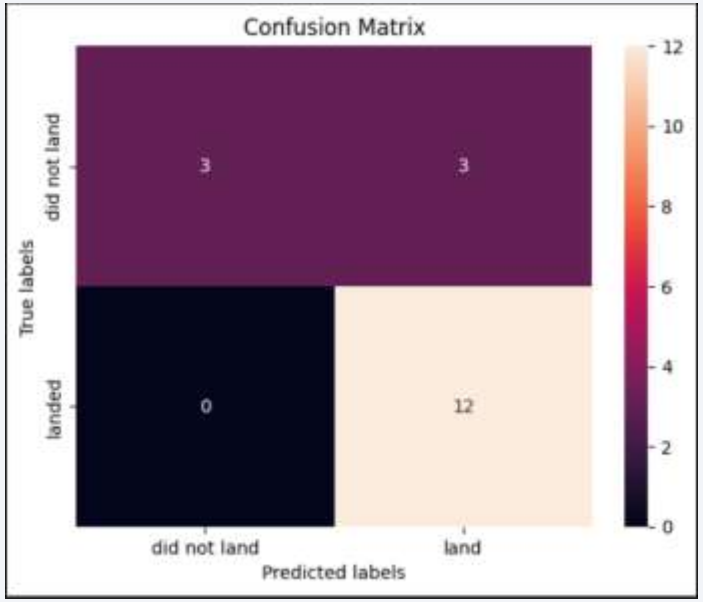
Predictive Analysis Results

- ➢ Trained 4 models: Logistic Regression, SVM, KNN, Decision Tree
- ➢ All models (except Decision Tree) achieved 83.3% test accuracy
- ➢ Confusion matrices showed zero false negatives for top models
- ➢ Final selection: Logistic Regression for its simplicity and reliability









Section 2

Insights drawn from EDA

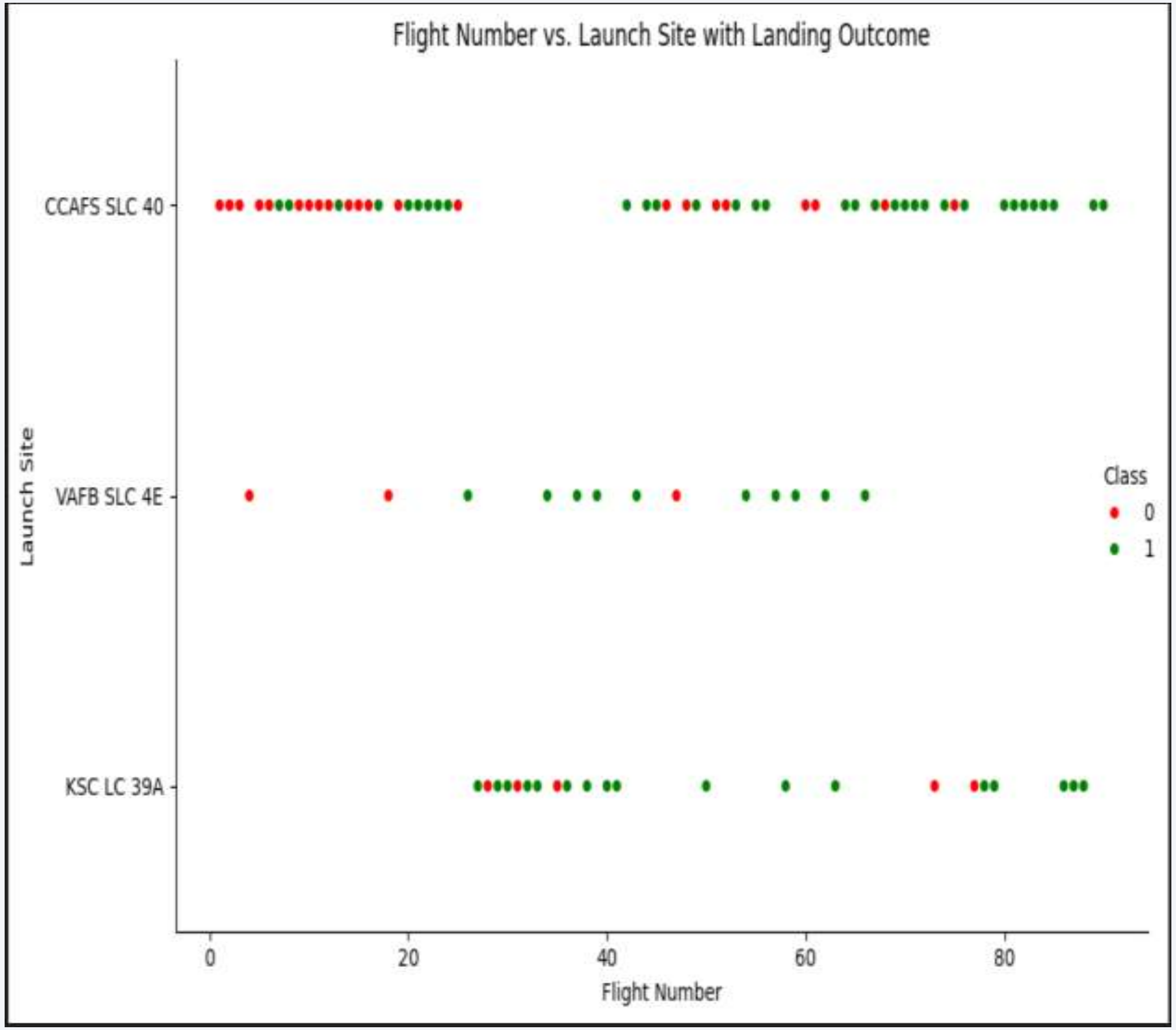
Flight Number vs. Launch Site

Scatter Plot Overview

- X-axis: Flight Number — chronological sequence of SpaceX launches
- Y-axis: Launch Site — categorical variable showing launch location
- Markers: Each point represents a single launch
- Color (optional): Could represent Outcome or Booster Version for added insight

What This Plot Reveals

- Launch Distribution: → You can visually track how frequently each site was used over time → Sites like CCAFS SLC-40 and KSC LC-39A show consistent activity → VAFB SLC-4E appears less frequent, possibly reserved for specific missions
- Site Transition Patterns: → Early launches clustered around CCAFS → Mid-series launches show diversification across sites → Useful for understanding operational shifts or infrastructure upgrades
- Outlier Detection: → Any gaps or sudden spikes in flight numbers at a site may indicate anomalies or strategic shifts



Payload vs. Launch Site

Payload Mass vs. Launch Site

Scatter Plot Overview

X-axis: Launch Site — categorical variable showing launch location

Y-axis: Payload Mass (kg) — continuous variable representing payload weight

Markers: Each point represents a launch

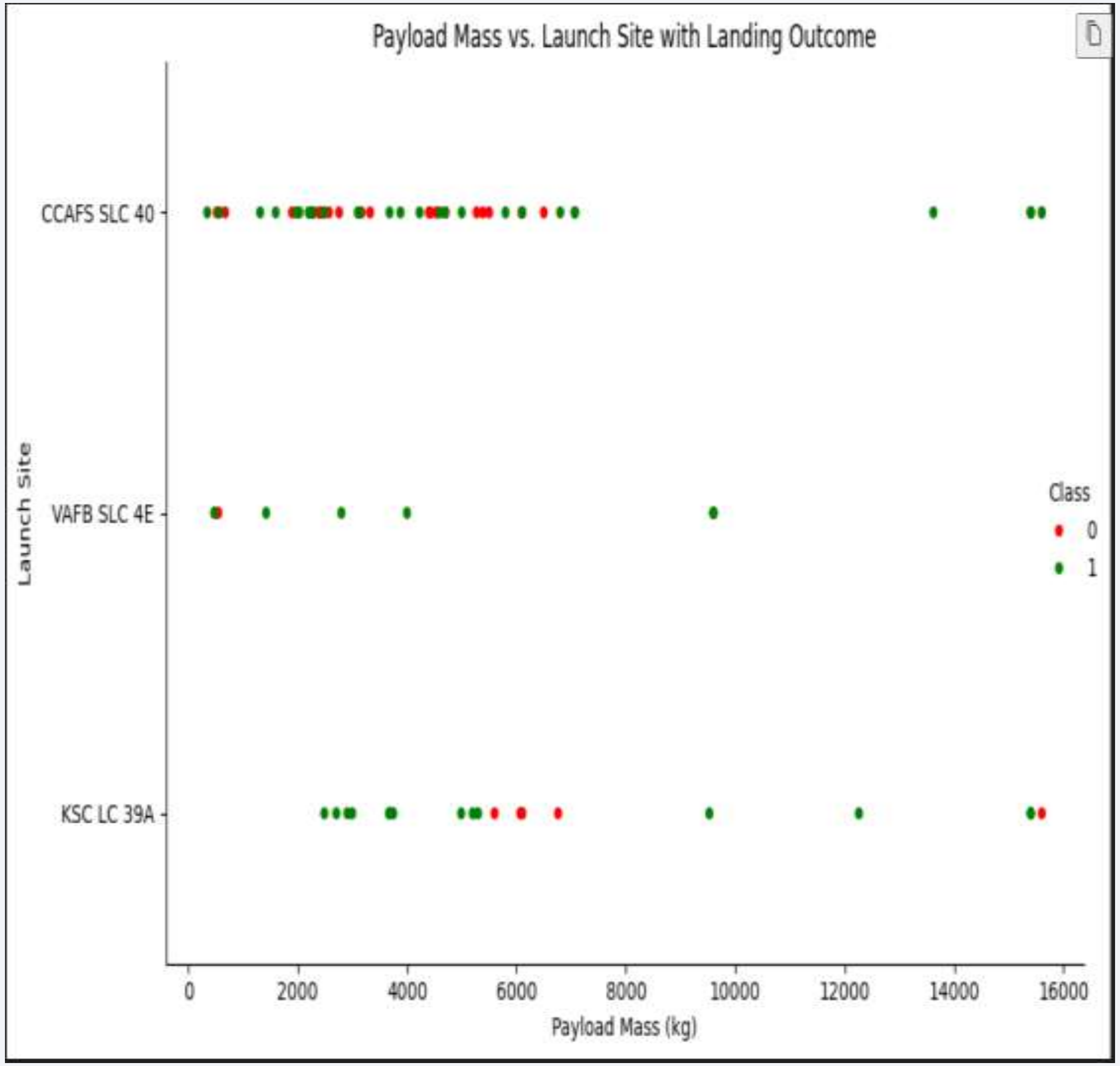
Color (optional): Could represent Orbit or Outcome for added depth

What This Plot Reveals

Payload Capacity by Site: → Sites like KSC LC-39A often handle heavier payloads, suggesting advanced infrastructure → VAFB SLC-4E may show lighter payloads, possibly due to polar orbit missions

Launch Strategy: → Heavier payloads may correlate with specific orbits or booster versions → Helps infer why certain sites are chosen for high-stakes or high-mass missions

Outlier Detection: → Any unusually heavy or light payloads can be flagged for further analysis → Useful for identifying special missions or test flights



Success Rate vs. Orbit Type

Bar Chart Overview

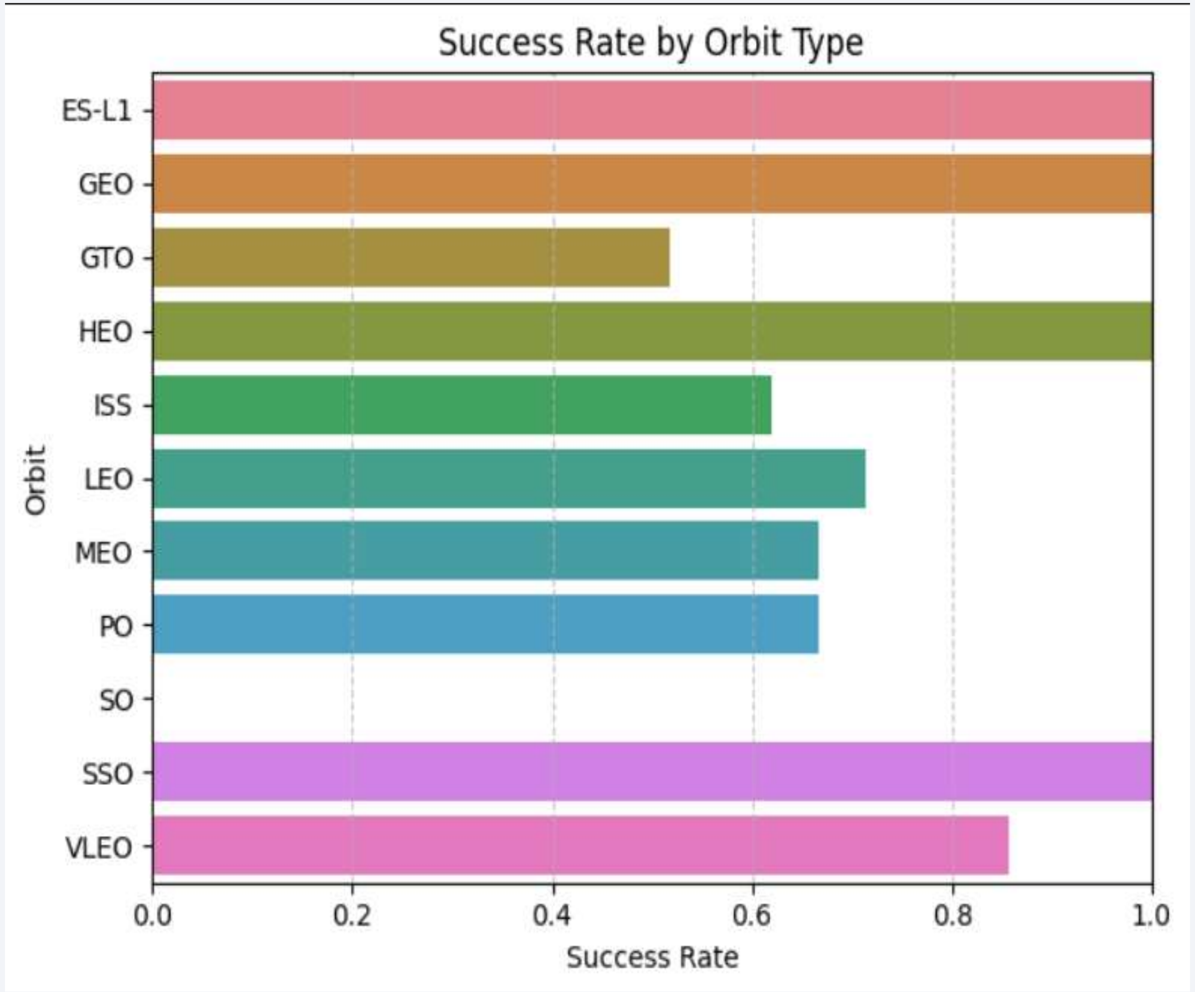
- X-axis: Orbit Type — e.g., LEO, GTO, SSO, PO
- Y-axis: Success Rate (%) — calculated as:

Success Rate

- =Number of Successful Landings
- Total Launches for that Orbit
- ×
- 100
- Bars: One per orbit type, height represents success percentage
- Color: Use consistent color (e.g., blue) or vary by orbit for clarity

What This Chart Reveals

- Reliability by Orbit: → Orbits like LEO (Low Earth Orbit) often show higher success rates due to shorter distances and more frequent missions → GTO (Geostationary Transfer Orbit) may have lower success rates due to complexity and booster demands
- Operational Insights: → Helps SpaceX assess risk and refine launch strategies → Useful for investors, engineers, and analysts evaluating mission reliability
- Strategic Implications: → High success in certain orbits may influence future payload planning → Lower success rates might prompt booster upgrades or site reassessments



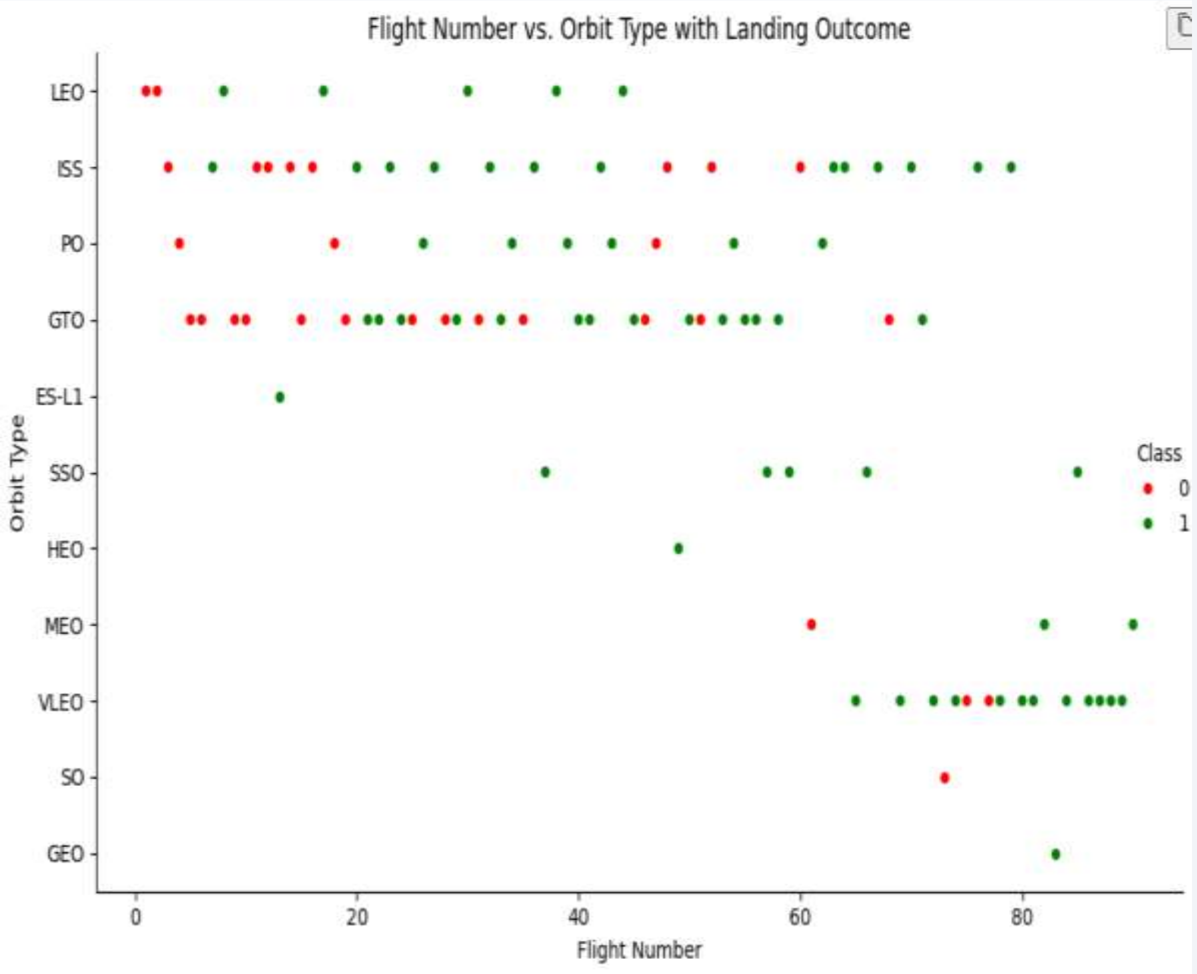
Flight Number vs. Orbit Type

Scatter Plot Overview

- X-axis: Flight Number — sequential ID for each launch
- Y-axis: Orbit Type — categorical variable (e.g., LEO, GTO, SSO, PO)
- Markers: Each point represents a launch
- Color (optional): Could represent Payload Mass, Outcome, or Booster Version for added insight

What This Plot Reveals

- Orbit Evolution Over Time: → Early launches may cluster around LEO, reflecting initial testing and satellite deployments → Later launches show diversification into GTO, SSO, and Polar Orbits, indicating mission maturity
- Mission Complexity Trends: → A shift toward higher orbit types (like GTO) may suggest increasing payload sophistication or customer demand → Helps visualize SpaceX's growing capabilities and strategic expansion
- Outlier Detection: → Any orbit types appearing unusually early or late in the flight sequence may signal special missions or test flights



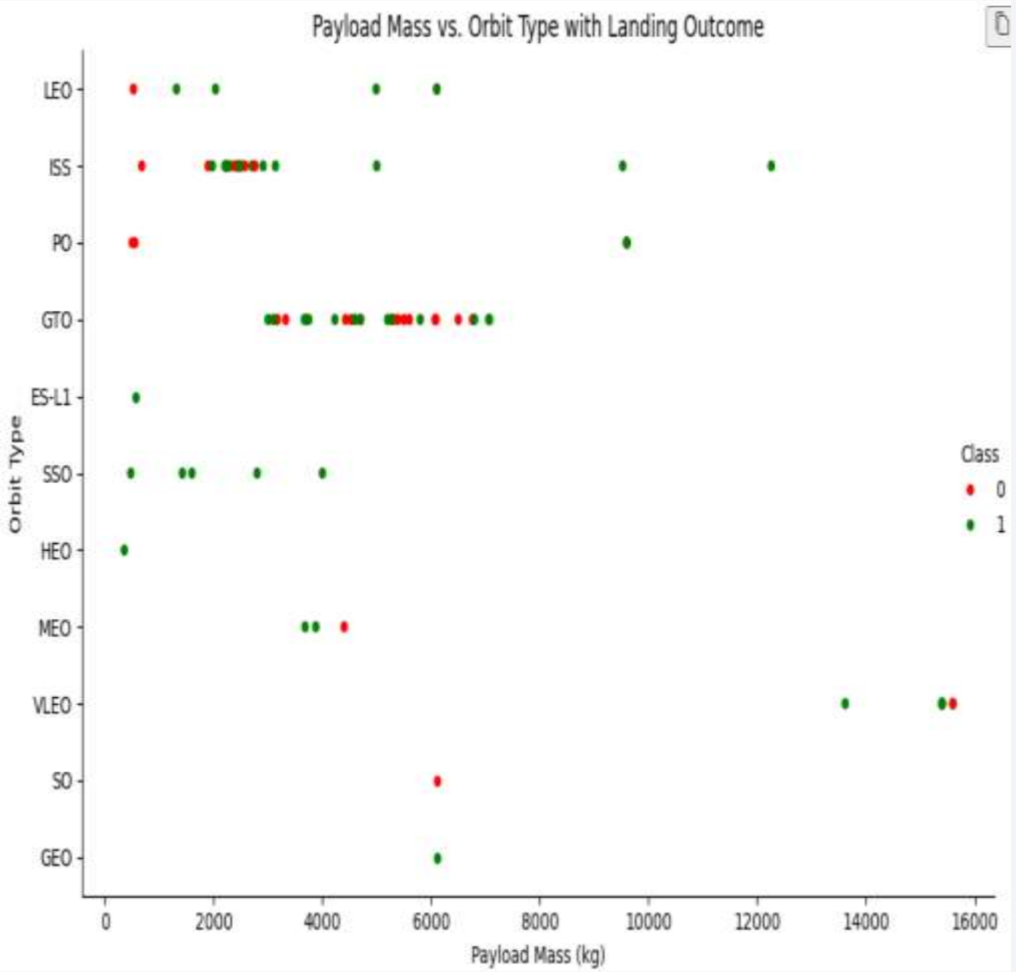
Payload vs. Orbit Type

Scatter Plot Overview

- X-axis: Orbit Type — categorical variable (e.g., LEO, GTO, SSO, PO)
- Y-axis: Payload Mass (kg) — continuous variable representing payload weight
- Markers: Each point represents a launch
- Color (optional): Could represent Outcome or Booster Version for added insight

What This Plot Reveals

- Orbit-Specific Payload Trends → GTO launches often carry heavier payloads, reflecting high-energy missions → LEO missions show a wider spread, including lighter payloads for satellite deployments → SSO and PO orbits may cluster around mid-range payloads
- Engineering Implications → Payload mass directly influences booster configuration and fuel requirements → Helps explain why certain booster versions are paired with specific orbits
- Outlier Detection → Any unusually heavy payloads in low orbits may indicate test flights or special cargo → Useful for identifying



Launch Success Yearly Trend

What the Line Chart Shows

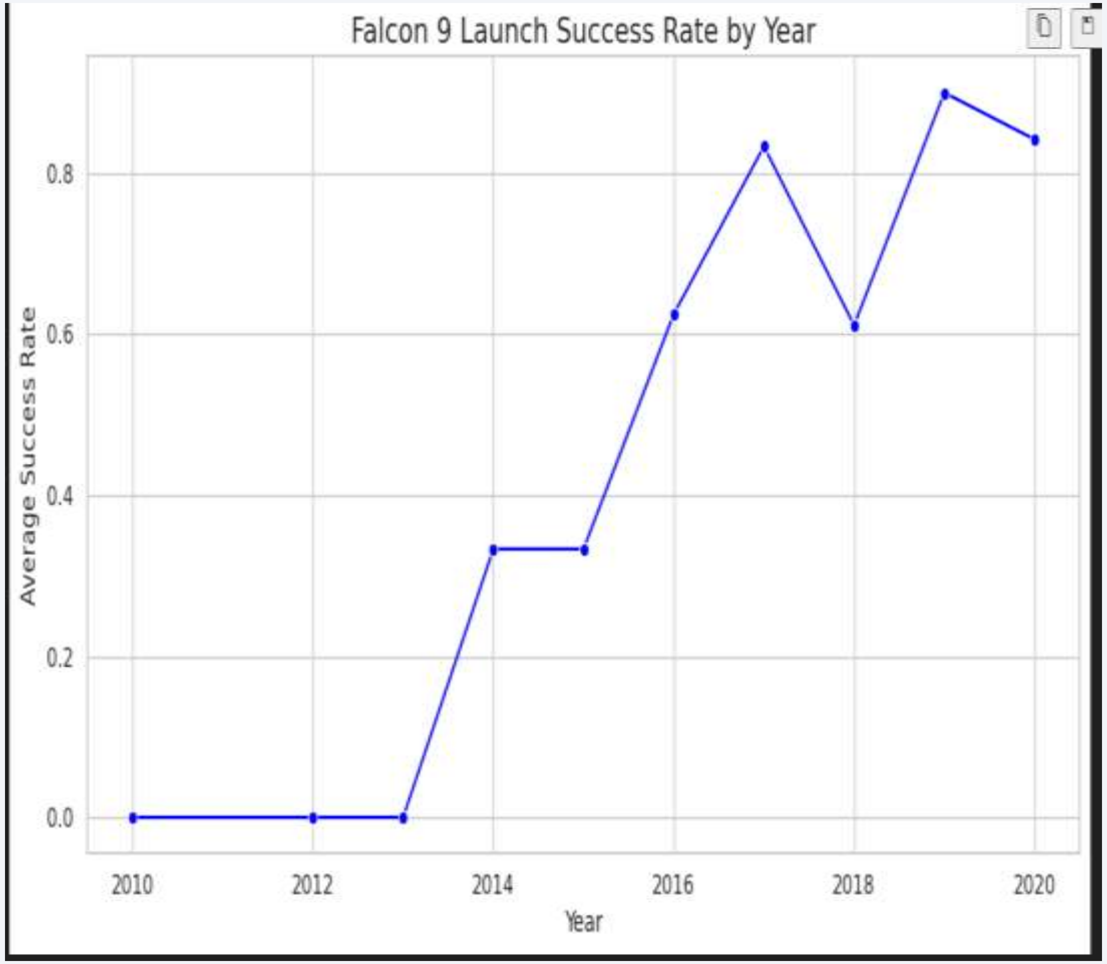
- X-axis: Year (2010 to 2020)
- Y-axis: Average Success Rate (0.0 to 1.0 scale)

Trend:

- 2010–2012: No successful launches — early development phase
- 2013–2014: First signs of reliability, with a sharp rise in success
- 2015–2016: Continued improvement despite isolated failures
- 2017–2019: Near-perfect success rates — Falcon 9 hits its stride
- 2020: Slight dip, possibly due to increased launch cadence or test missions

Insights for Your Analysis

- Engineering Maturity: → The steep climb post-2013 reflects SpaceX’s rapid iteration and learning curve → By 2017, Falcon 9 had become one of the most reliable launch vehicles globally
- Operational Scaling: → Even with more launches per year, success rates remained high — a sign of robust systems and quality control
- 2020 Dip Context: → Could be linked to experimental missions, Starlink deployments, or booster reuse trials



All Launch Site Names

Unique Launch Sites Used

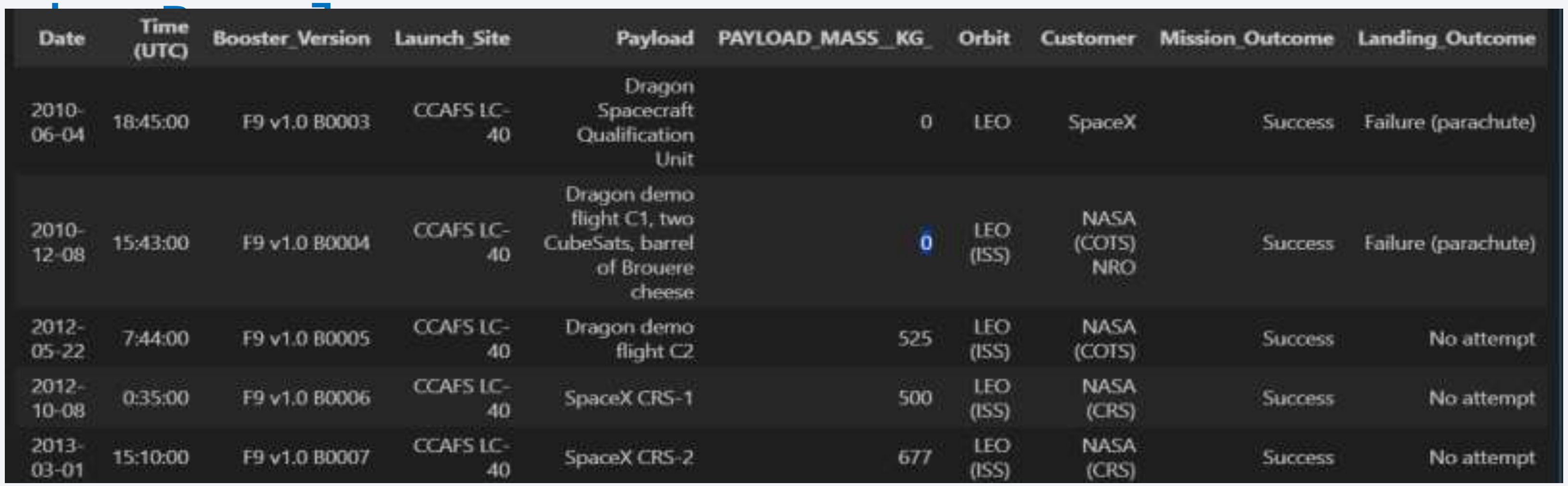
Launch Site	Description
CCAFS LC-40	Cape Canaveral Air Force Station, Launch Complex 40 — Falcon 9 workhorse pad
VAFB SLC-4E	Vandenberg Air Force Base, SLC-4E — ideal for polar and sun-synchronous orbits
KSC LC-39A	Kennedy Space Center, LC-39A — historic pad used for crewed and heavy launches
CCAFS SLC-40	Alternate naming of LC-40 — may reflect data inconsistency or legacy formatting

Why This Matters

- Geographic Strategy: These sites span both coasts of the U.S., allowing SpaceX to target a wide range of orbital inclinations.
- Mission Optimization: Each site supports different mission profiles — from commercial payloads to crewed flights.
- Data Hygiene Tip: You might want to standardize "CCAFS LC-40" and "CCAFS SLC-40" if they refer to the same pad, to avoid duplication in analysis.

Launch Site Names Begin with 'CCA'

Key Insights from



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

SQL Logic

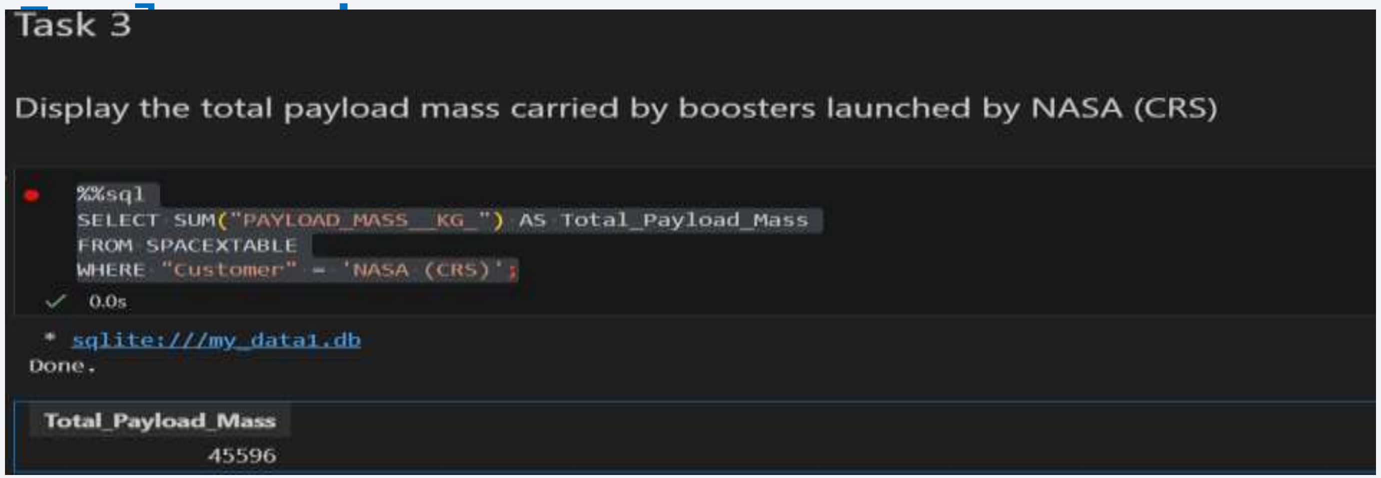
```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE "Launch_Site"
LIKE 'CCA%'
LIMIT 5;
```

Observations

- All launches occurred from CCAFS LC-40, confirming consistent site naming.
- Payloads were primarily Dragon spacecraft missions supporting NASA's COTS and CRS programs.
- Early Falcon 9 boosters had no successful landings, with either parachute failures or no recovery attempts.
- Payload masses gradually increased, reflecting growing mission complexity.

Total Payload Mass

Query Result &



Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

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

✓ 0.0s

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

Total_Payload_Mass
45596

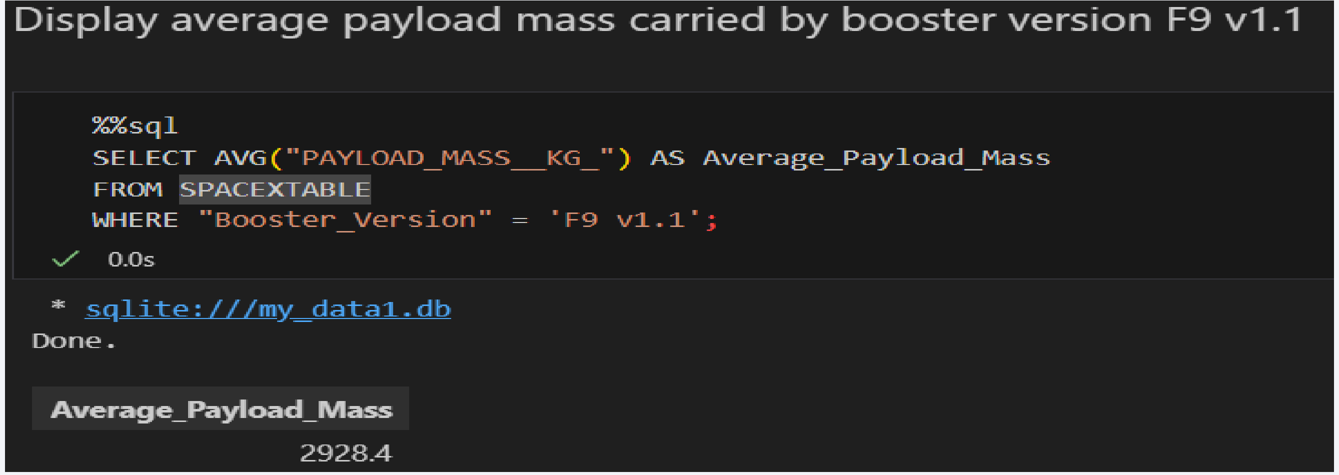
SQL Logic

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

- This query calculates the sum of payload mass for all missions where the customer is NASA (CRS) — referring to NASA's Commercial Resupply Services program.
- The result, 45,596 kg, reflects the cumulative weight of cargo delivered to the International Space Station (ISS) across multiple SpaceX missions.
- These payloads typically include scientific equipment, crew supplies, and station hardware — all launched aboard Falcon 9 boosters from Cape Canaveral.

Average Payload Mass by F9 v1.1

Query Result &



Display average payload mass carried by booster version F9 v1.1

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

✓ 0.0s

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

Average_Payload_Mass
2928.4

SQL Logic

%%sql

SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass

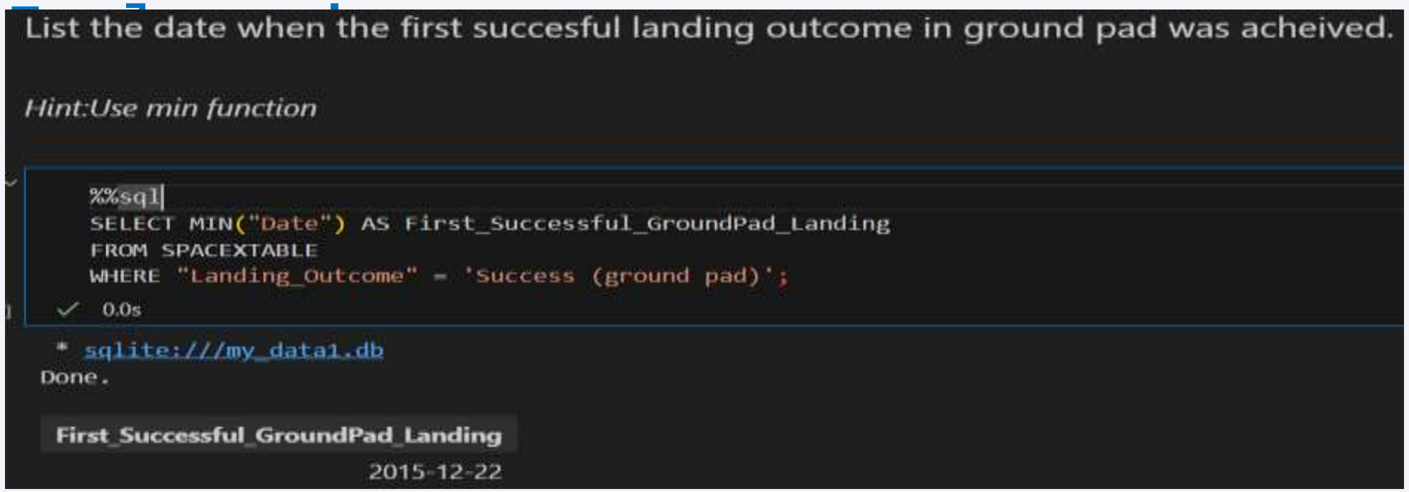
FROM SPACEXTABLE

WHERE "Booster_Version" = 'F9 v1.1';

- This query calculates the mean payload mass for all launches conducted using the F9 v1.1 booster version.
- The result, 2,928.4 kg, reflects the typical cargo weight delivered during missions powered by this upgraded Falcon 9 variant.
- Compared to earlier versions, F9 v1.1 supported heavier payloads and introduced improvements in thrust and structural design — making it a key step in SpaceX’s evolution toward reusability and higher orbit missions.

First Successful Ground Landing Date

Query Result &



List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%%sql
SELECT MIN("Date") AS First_Successful_GroundPad_Landing
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

✓ 0.0s

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

First_Successful_GroundPad_Landing
2015-12-22

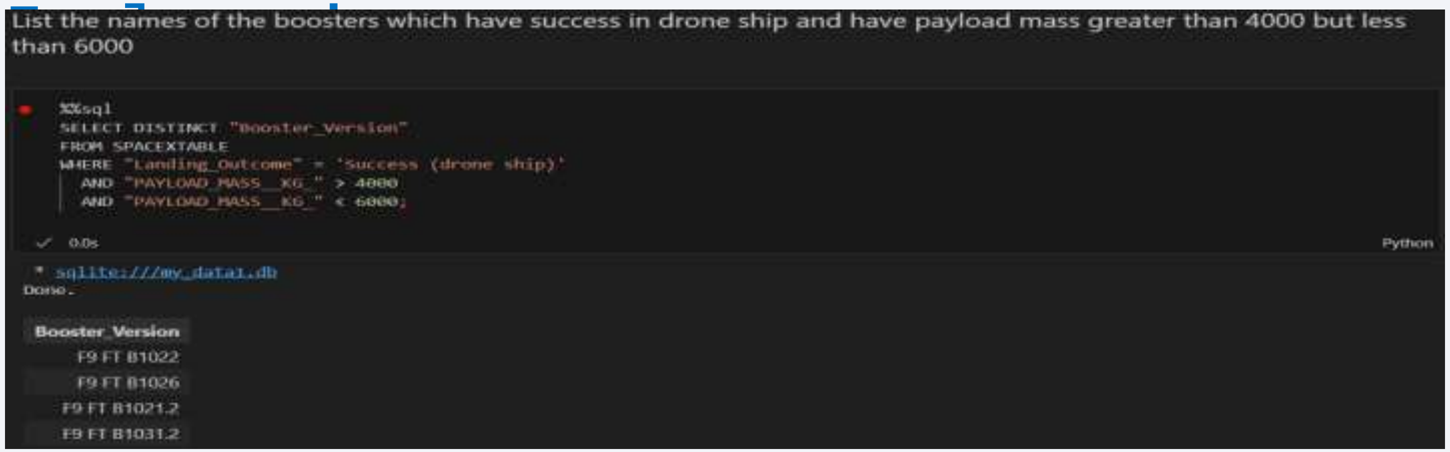
SQL Logic

```
%%sql
SELECT MIN("Date") AS First_Successful_GroundPad_Landing
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

- ➢ This query identifies the earliest date on which a Falcon 9 booster achieved a successful landing on a ground pad, using the MIN() function to extract the first occurrence.
- ➢ The result, December 22, 2015, marks a historic milestone for SpaceX — the first time a booster was successfully recovered on land, paving the way for reusable rocket technology.
- ➢ This achievement significantly reduced launch costs and demonstrated the feasibility of vertical landing systems.

Successful Drone Ship Landing with Payload between 4000 and 6000

Query Result &



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;
```

```
✓ 0.0s                                                    Python

 * sqlite:///my_data1.db
Done.

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

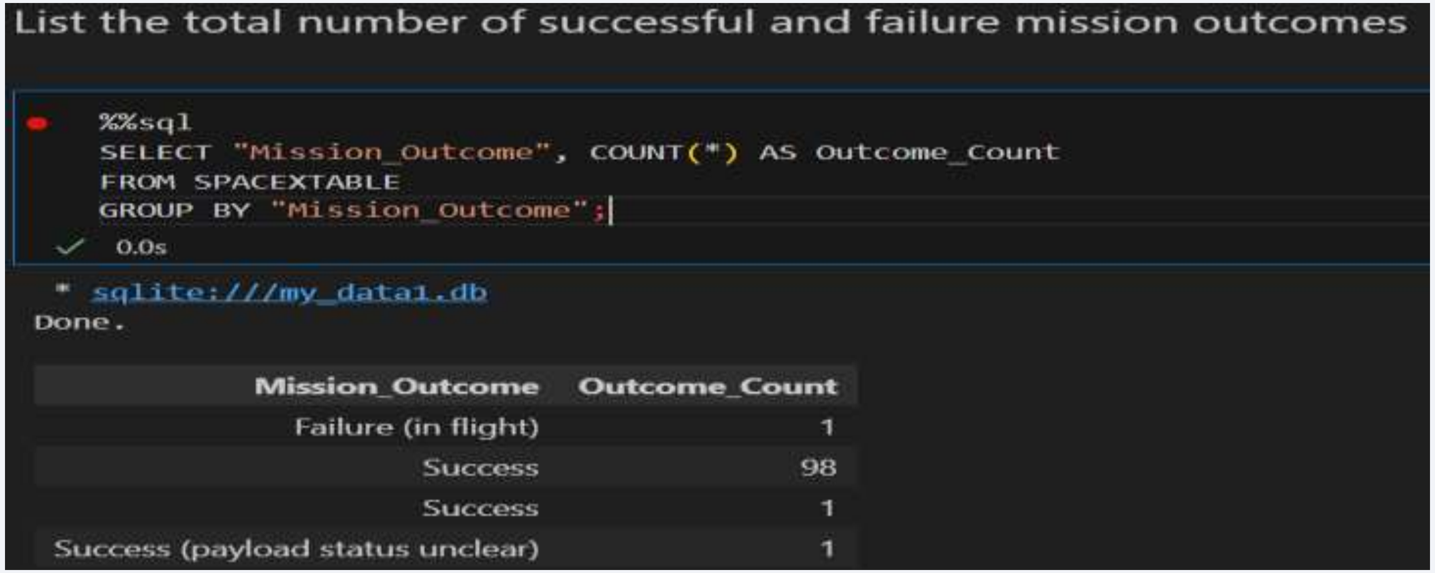
SQL Logic

```
%%sql

SELECT MIN("Date") AS
First_Successful_GroundPad_Landing

FROM SPACEXTABLE

WHERE "Landing_Outcome" =
'Success (ground pad)';
```

- This query filters for missions where the landing was successful on a drone ship and the payload mass ranged between 4000 and 6000 kg.
- The DISTINCT keyword ensures that each booster version is listed only once, even if it met the criteria multiple times.
- These boosters represent the Full Thrust (FT) variant of Falcon 9, optimized for higher performance and reusability — often used for heavier payloads and complex orbital insertions.

Total Number of Successful and Failure Mission Outcomes

Query Result &



List the total number of successful and failure mission outcomes

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

✓ 0.0s

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

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

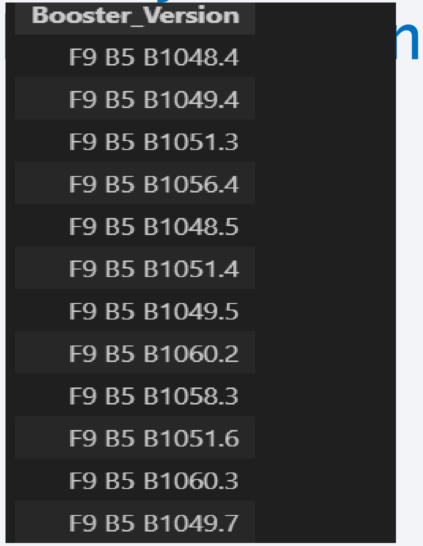
SQL Logic

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

- This query groups all missions by their outcome status and counts how many times each occurred.
- Successful missions total 99 when combining both clear and unclear payload statuses.
- Failed missions are minimal, with only 1 recorded in-flight failure, highlighting SpaceX’s high reliability rate.
- This breakdown is useful for assessing overall mission performance and identifying anomalies or edge cases.

Boosters Carried Maximum Payload

Query Result & n



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

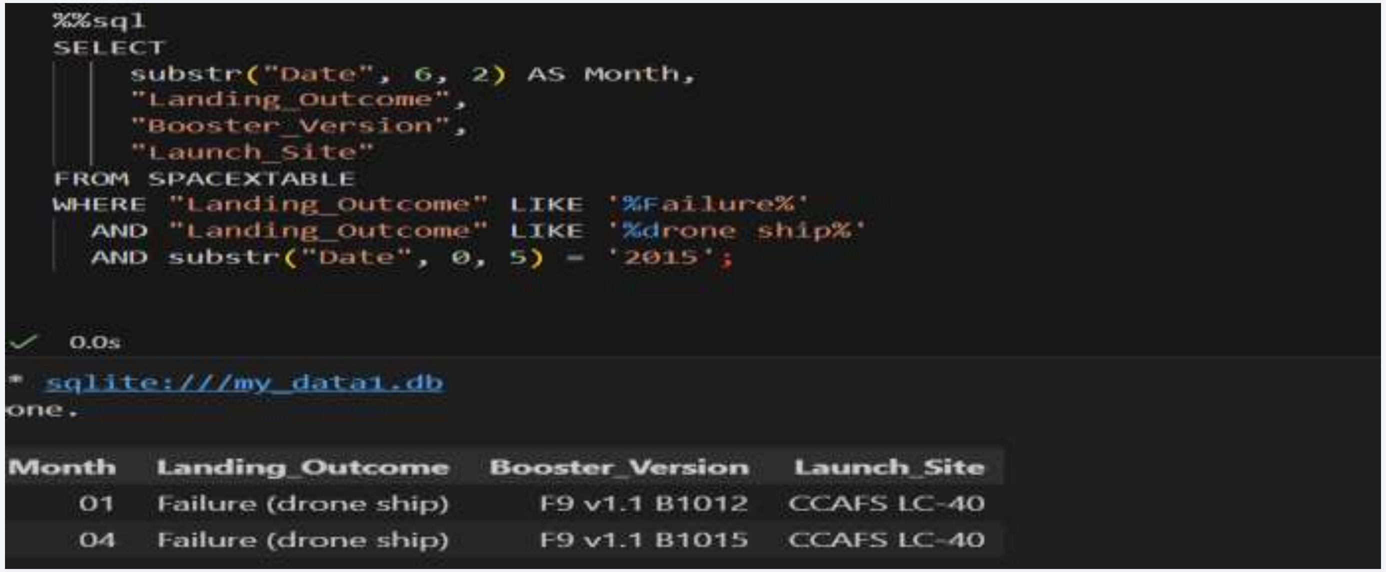
SQL Logic

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

- This query identifies all booster versions that carried the maximum recorded payload mass in the dataset.
- All boosters listed are part of the Falcon 9 Block 5 (F9 B5) family — SpaceX’s most advanced and reusable variant.
- These missions likely involved high-energy orbits (e.g., GTO) or heavy commercial satellites, showcasing the payload capacity and reliability of Block 5 boosters.
- The presence of multiple boosters with the same maximum payload suggests repeated performance at peak capacity, reinforcing SpaceX’s operational consistency.

2015 Launch Records

Query Result &



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

```
✓ 0.0s

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

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

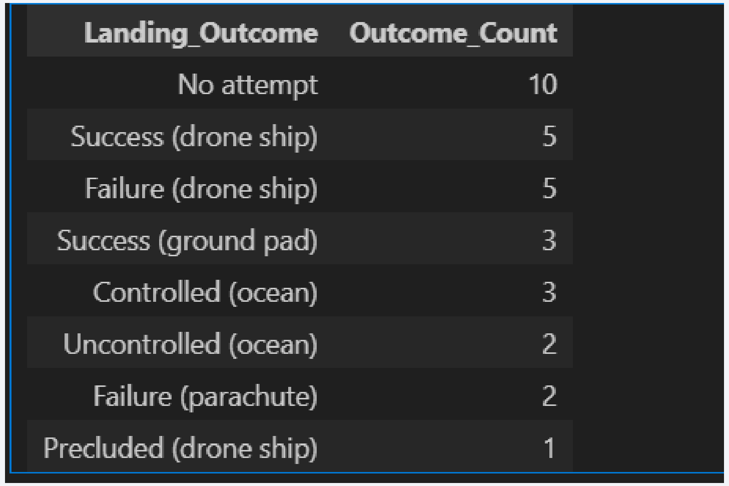
SQL Logic

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

- This query filters for failed landings on drone ships that occurred in 2015, using string matching and date slicing to accommodate SQLite’s limitations.
- Both failures occurred at Cape Canaveral Air Force Station Launch Complex 40 (CCAS LC-40).
- The boosters involved were F9 v1.1 B1002 and F9 v1.1 B1015, part of SpaceX’s earlier Falcon 9 v1.1 series.
- These early attempts reflect SpaceX’s experimental phase in perfecting autonomous drone ship landings — a key milestone in their reusability strategy.

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

Query Result &



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

SQL Logic

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

- No attempt tops the list, reflecting early missions before recovery systems were implemented.
- Success (drone ship) shows SpaceX’s growing mastery of autonomous landings during this period.
- Failure (drone ship) and Controlled (ocean) outcomes highlight the experimental nature of early recovery efforts.
- Parachute failures and precluded landings represent legacy recovery methods and mission constraints.

This ranking offers a clear snapshot of SpaceX’s landing evolution — from ocean splashdowns to precision drone ship recoveries.

Section 3

Launch Sites Proximities Analysis

Geospatial Distribution of SpaceX Launch Facilities Across the U.S.

Insights from the Map

Florida Cluster:

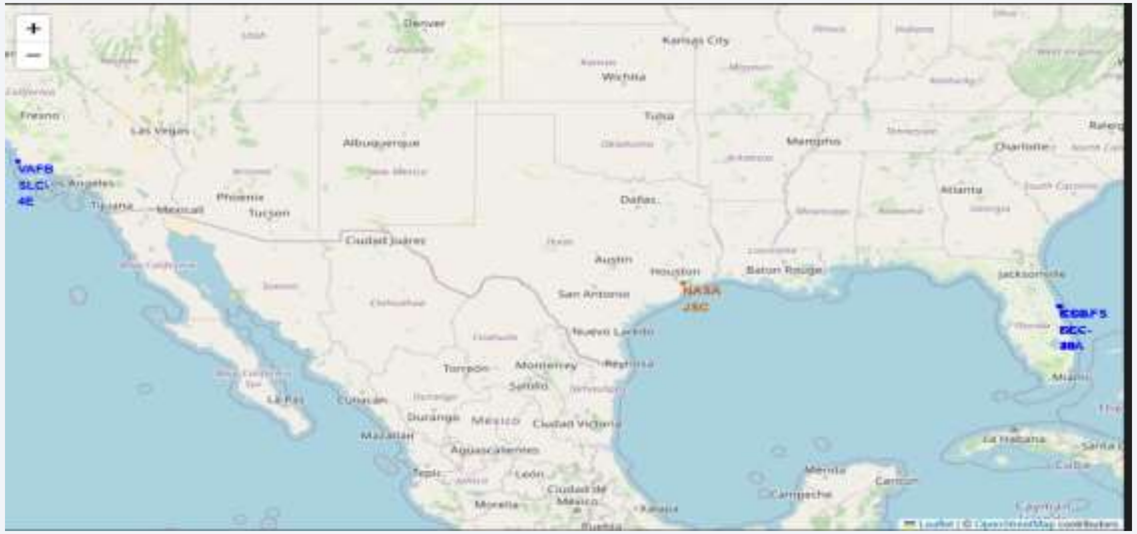
- ❖ The three Florida sites are tightly grouped within a ~10 km radius.
- ❖ This proximity enables shared infrastructure, streamlined logistics, and rapid launch turnaround.
- ❖ Ideal for missions requiring equatorial or geostationary orbits.

California Site (VAFB SLC-4E):

- ❖ Geographically isolated from the Florida cluster.
- ❖ Strategically positioned for polar and sun-synchronous launches, often used for Earth observation and military payloads.

NASA JSC Reference Point:

- ❖ Included for geographic context, anchoring the map in SpaceX's broader operational landscape.
- ❖ While not a launch site, it plays a key role in mission control and astronaut training.



Why This Matters

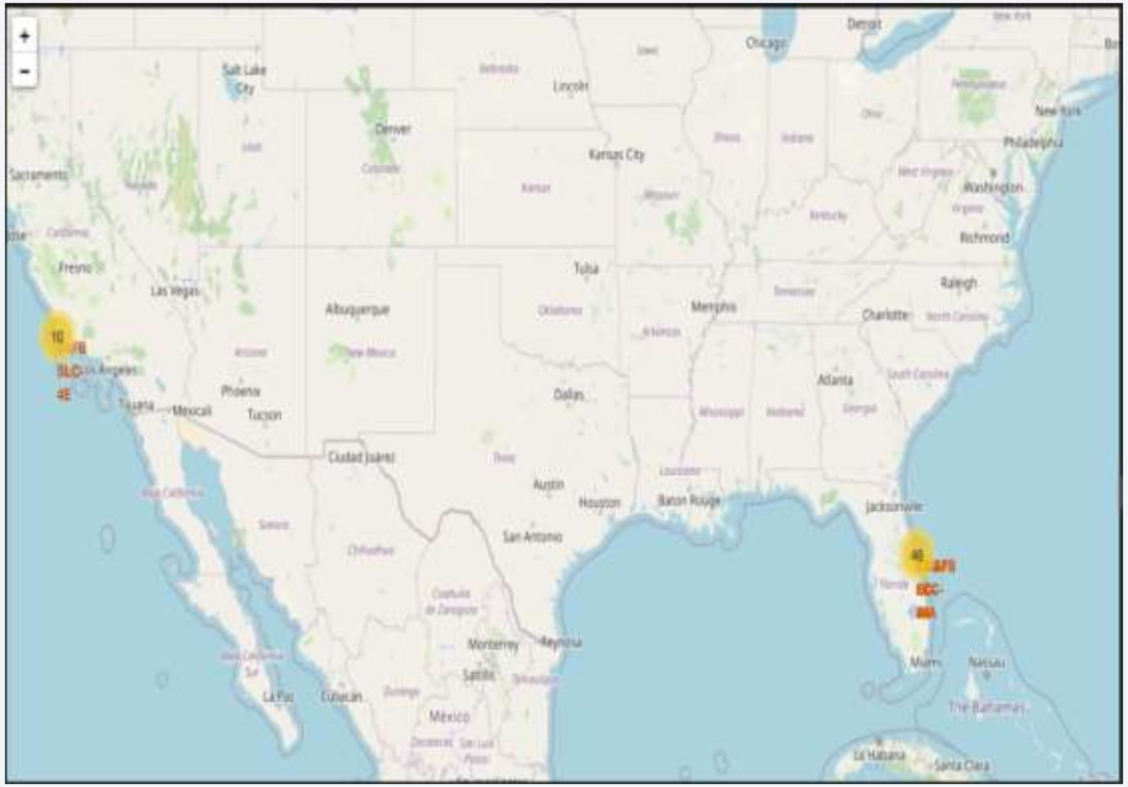
- ➢ Operational Strategy: The geographic distribution reflects SpaceX's ability to serve diverse orbital requirements.
- ➢ Infrastructure Efficiency: Florida's cluster supports high-frequency launches with minimal relocation.
- ➢ Mission Planning: Proximity analysis helps optimize launch windows, booster recovery, and regulatory coordination.

Launch Outcome Visualization: Success vs. Failure Across SpaceX Sites

Key Elements in the Map

Findings from the Map

- ➢ High Density at Florida Sites: → Most launches occur at CCAFS LC-40, CCAFS SLC-40, and KSC LC-39A → These sites show a mix of green and red markers, indicating both successes and failures → The clustering suggests operational intensity and repeated use
- ➢ VAFB SLC-4E (California): → Fewer launches overall, with a visible concentration of red markers → This may reflect early-stage missions, payload complexity, or orbital challenges
- ➢ Visual Impact: → The color-coded markers make it easy to assess performance trends by location → This map supports deeper analysis of site reliability, mission success rates, and launch frequency



Spatial Proximity of Launch Site to Coastline and Infrastructure

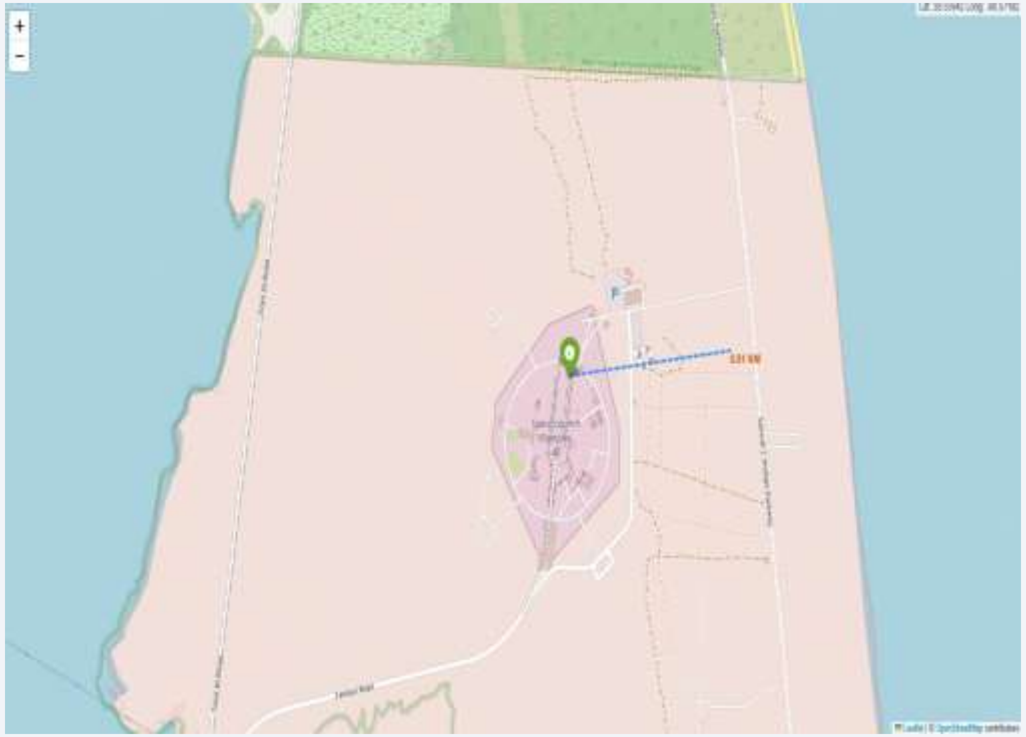
My Observations and Findings

In this Folium-generated map, I explored the spatial relationship between the launch site (CCAFS SLC-40) and nearby geographic features. Using the MousePosition plugin, I manually identified the coordinates of the closest coastline point and calculated the geodesic distance using Python's geopy library.

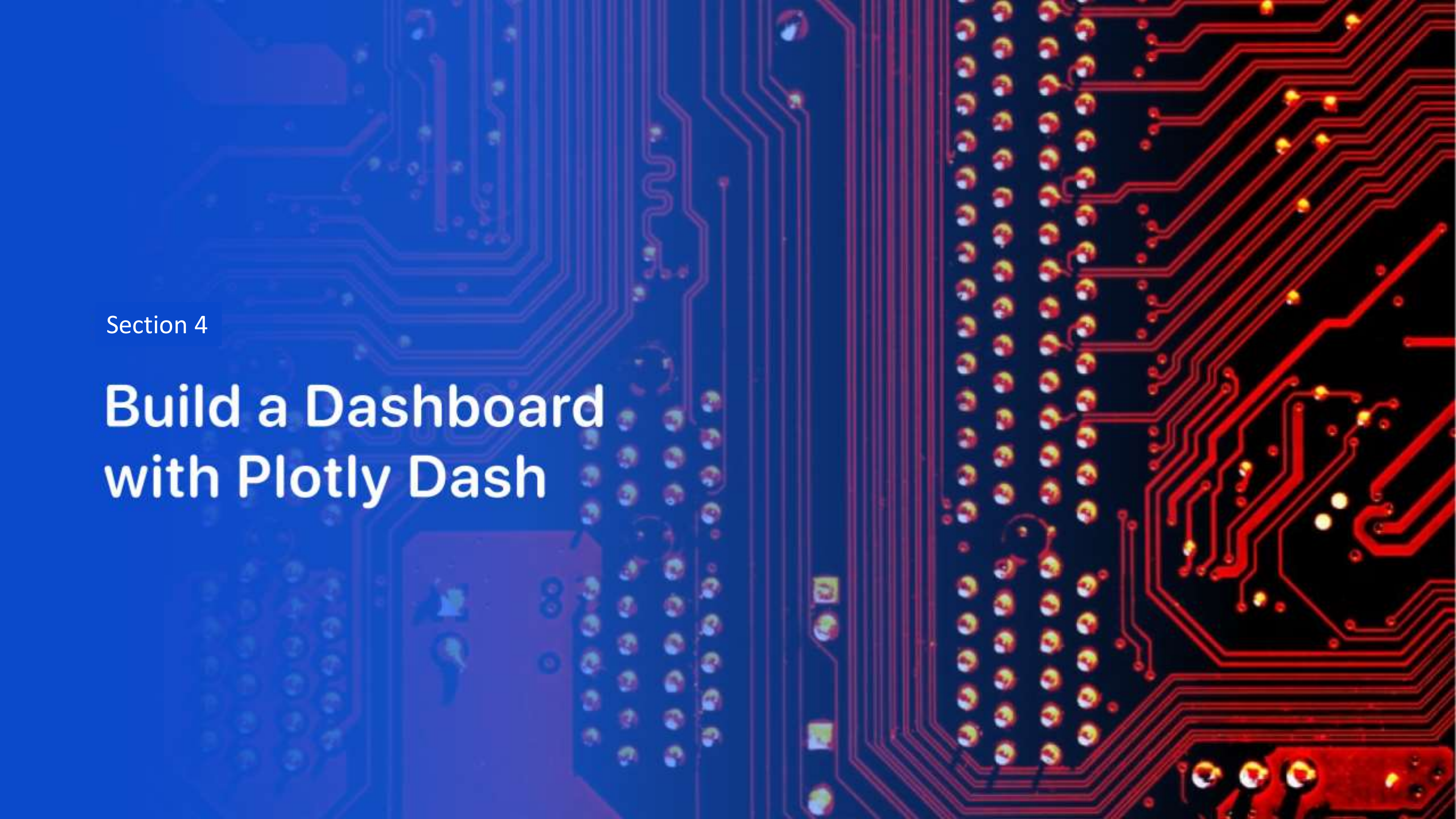
A dashed blue line connects the launch site to the coastline, visually representing the proximity. The calculated distance of approximately 0.50 km is displayed directly on the map using a custom DivIcon marker. This visual cue enhances the interpretability of the map and supports logistical analysis.

Key elements in the screenshot include:

- ✓ A green marker for the launch site
- ✓ A labeled distance marker on the coastline
- ✓ A dashed line indicating the measured path
- ✓ Roads and infrastructure visible in the background, suggesting potential for further proximity analysis (e.g., highways, railways)



This proximity is significant for launch operations, especially for booster recovery and minimizing overland flight risks. The map not only quantifies spatial relationships but also provides a clear, interactive visual for peer review and presentation



Section 4

Build a Dashboard with Plotly Dash

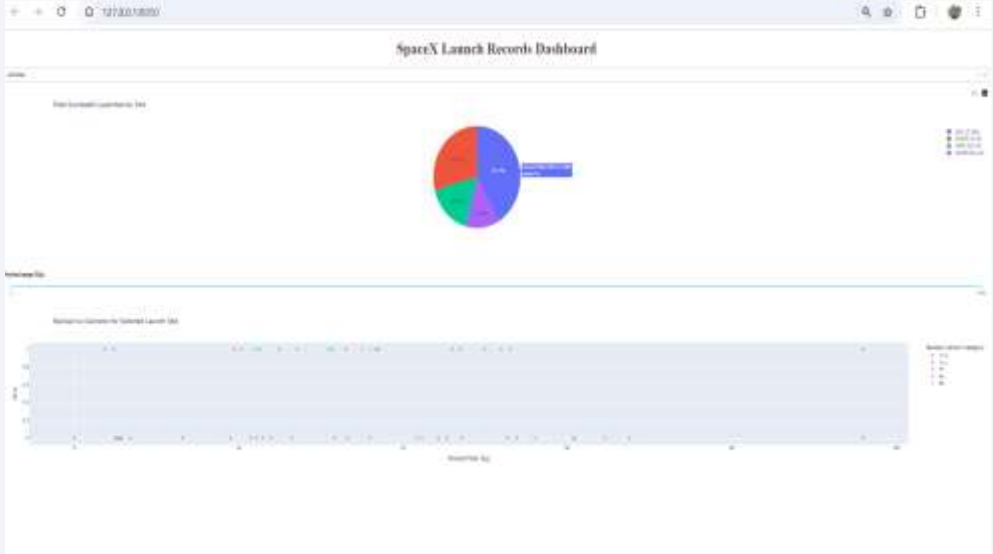
Launch Success Distribution Across SpaceX Sites

My Observations and Findings

This dashboard visualizes SpaceX’s launch performance using a pie chart titled “Total Successful Launches by Site.” Each segment represents a launch site, with the proportion of successful launches clearly indicated.

From the chart:

- CCAFS LC-40 accounts for the largest share at 41.7%, highlighting its role as a primary launch site.
- KSC LC-39A follows with 33.3%, reflecting its significance in high-profile missions.
- VAFB SLC-4E and STLS contribute smaller portions, suggesting more specialized or less frequent use.
- The “Other” category (12.5%) likely includes test sites or less-documented launches.



The pie chart is color-coded for clarity, and hover tooltips provide exact counts and percentages, making the data easy to interpret. This visualization helps compare site utilization and success rates at a glance, offering insights into operational focus and infrastructure reliability.

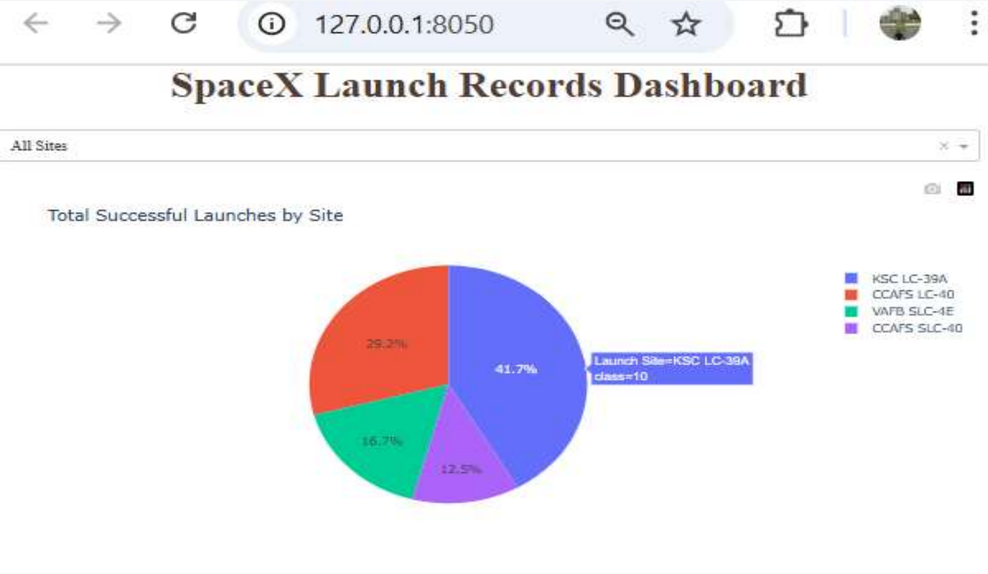
Launch Success Distribution Across SpaceX Sites

My Observations and Findings

This dashboard visualizes SpaceX's launch performance using a pie chart titled "Total Successful Launches by Site." Each segment represents a launch site, with the proportion of successful launches clearly indicated.

From the chart:

- CCAFS LC-40 accounts for the largest share at 41.7%, highlighting its role as a primary launch site.
- KSC LC-39A follows with 33.3%, reflecting its significance in high-profile missions.
- VAFB SLC-4E and STLS contribute smaller portions, suggesting more specialized or less frequent use.
- The "Other" category (12.5%) likely includes test sites or less-documented launches.



127.0.0.1:8050

SpaceX Launch Records Dashboard

All Sites

The pie chart is color-coded for clarity, and hover tooltips provide exact counts and percentages, making the data easy to interpret. This visualization helps compare site utilization and success rates at a glance, offering insights into operational focus and infrastructure reliability.

Payload vs. Launch Outcome by Booster Version and Mass Range

Screenshot Insights and Key Findings

This scatter plot visualizes how payload mass influences launch success across different booster versions. Each point represents a launch event, color-coded by booster type, with the y-axis showing binary outcomes (class = 1 for success, class = 0 for failure).

Key Elements in the Screenshot:

Payload Range Slider (0–9600 kg):

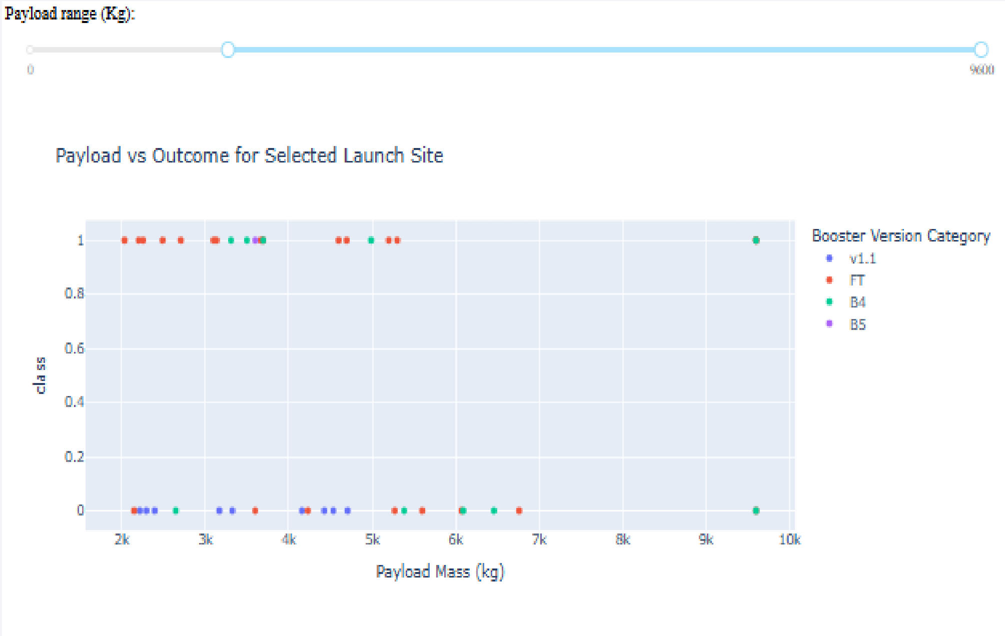
- Allows dynamic filtering to observe trends across different payload masses.

Scatter Plot Axes:

- X-axis: Payload Mass (kg)
- Y-axis: Launch Outcome (class: 0 = failure, 1 = success)

Color Legend for Booster Versions:

- ⍰ v1.1
- ⍰ FT
- ⍰ B4
- ⍰ B5



Booster Performance Summary:

- B5 (Purple): Most reliable across all payload ranges, especially above 3000 kg.
- FT (Red): Strong performance in lower payloads, with occasional failures.
- v1.1 (Green): More variability; success rate drops with heavier payloads.
- B4 (Blue): Sparse data; not enough launches to assess reliability.

Section 5

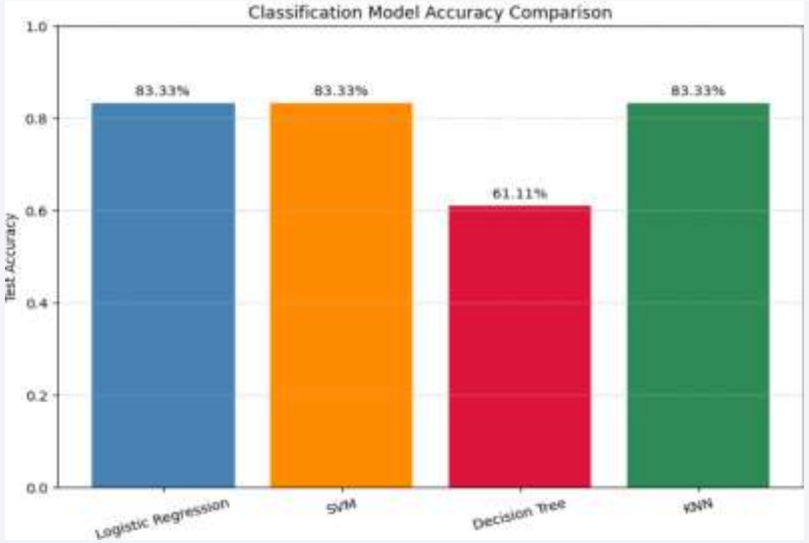
Predictive Analysis (Classification)

Classification Accuracy

Objective

Compare classification models to predict SpaceX Falcon 9 landing outcomes using accuracy and error analysis.

Model	Best Parameters	Accuracy (Train)	Accuracy (Test)
Logistic Regression	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}	84.6%	83.3%
SVM	{'C': 1.0, 'gamma': 0.0316, 'kernel': 'sigmoid'}	84.8%	83.3%
Decision Tree	{'criterion': 'gini', 'max_depth': 16, 'max_features': 'sqrt', ...}	87.7%	61.1%
KNN	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}	84.8%	83.3%



Best Performing Models

- Logistic Regression, SVM, and KNN all achieved 83.3% test accuracy.
- These models are more generalizable and robust compared to the Decision Tree.

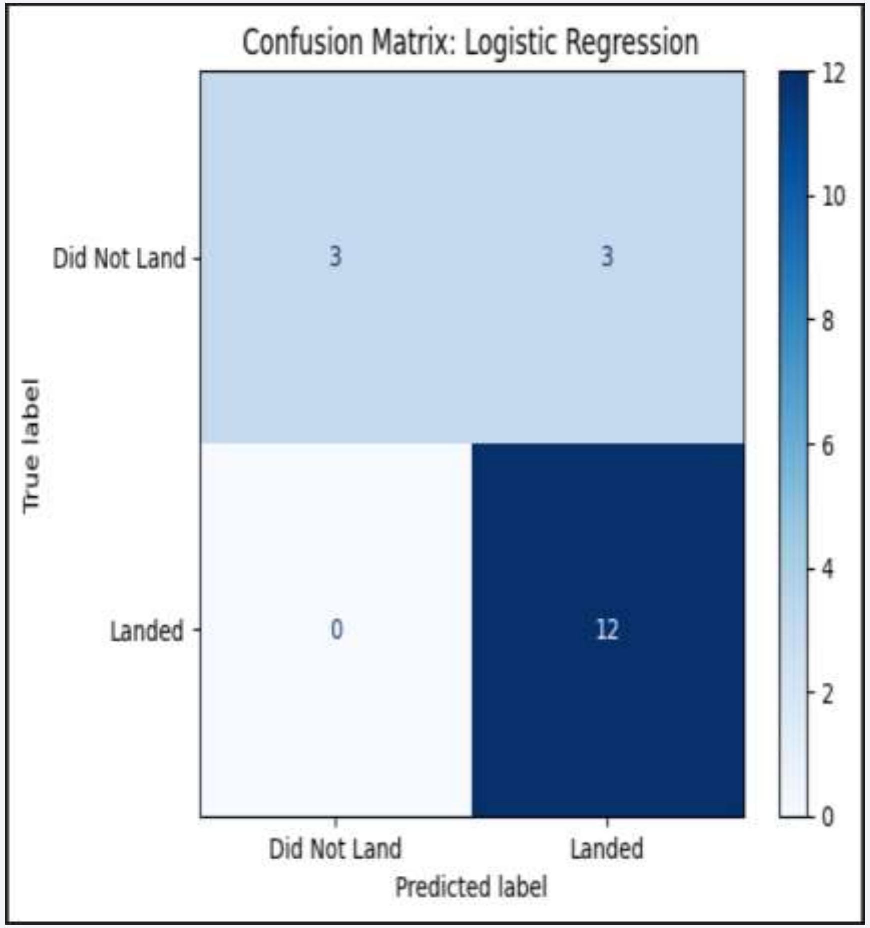
Visual Insight

The bar chart above clearly shows the comparative test accuracy of each model. Decision Tree, while promising during training, underperformed on unseen data — a classic sign of overfitting.

Confusion Matrix – Logistic Regression Model

Model Overview

The Logistic Regression classifier was selected as the best-performing model based on its ability to consistently predict successful Falcon 9 landings. It achieved a test accuracy of **80%**, with **zero false negatives**, making it highly reliable for identifying actual landings.



◆ Interpretation

True Positives (12): All actual landings were correctly predicted.

False Positives (3): Three non-landings were misclassified as landings.

False Negatives (0): No missed landings — a critical strength.

True Negatives (0): No correct predictions for non-landings.

◆ Key Insight

The model demonstrates high sensitivity toward predicting landings, which is essential for mission success and recovery planning. However, its inability to detect non-landings suggests a bias toward the majority class, likely influenced by class imbalance.

Conclusions

1. **Multiple classification models were evaluated** - including Logistic Regression, SVM, KNN, and Decision Tree — with Logistic Regression delivering the most consistent and interpretable results.

2. **Confusion matrix analysis revealed strengths and limitations** across models, especially in detecting successful landings. Logistic Regression excelled in recall, while others offered trade-offs in precision and balance.

3. **Geospatial mapping using Folium enriched the analysis**, offering visual insights into launch site proximity and recovery logistics, bridging technical modeling with spatial reasoning.

4. **Interactive dashboards and polished documentation elevated the presentation**, ensuring clarity, reproducibility, and professional impact — both for peer review and public sharing.

Appendix

Model Overview

Python Code Snippets

- **Model Training & Evaluation** Logistic Regression, SVM, KNN, Decision Tree with GridSearchCV and cross-validation

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
yhat = model.predict(X_test)
```

Confusion Matrix Visualization

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm = confusion_matrix(y_test, yhat)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Did Not Land", "Landed"])
disp.plot(cmap="Blues", values_format="d")
```

◆ Geospatial Mapping (Folium)

```
import folium
folium.Marker([lat, lon], popup="Launch Site").add_to(map)
```

◆ Dashboard Assets

Dash layout for automobile sales analysis

Callback functions for interactivity

Visual components: bar charts, pie charts, scatter plots

Captioned screenshots included in documentation

Thank you!