IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Adam Kureshi

DD April 2025

Outline

1. Executive Summary
2. Introduction
3. Methodology
4. Results
5. Conclusion
6. Appendix

1. Executive Summary

Summary of methodologies

- This project focused on predicting the successful landing of the SpaceX Falcon 9 first stage, a critical determinant of mission cost-efficiency. Using real launch data, we performed end-to-end data science processes including:
    - Data collection using SpaceX REST API and CSV data
    - Web scraping and wrangling
    - Exploratory data analysis using Seaborn, Matplotlib, SQL
    - Interactive mapping with Folium
    - Interactive dashboard development with Plotly Dash
    - Predictive analysis using classification models (Logistic Regression, SVM, Decision Trees, KNN)
- The best-performing model was the Decision Tree Classifier with a validation accuracy of 88.75% and test accuracy of 83.33%.

2. Introduction

SpaceX aims to reduce launch costs by reusing its Falcon 9 rocket stages. Predicting whether a first-stage landing will be successful informs mission planning and cost estimates.

- This project seeks to answer:
  - Which variables influence successful landings?
  - Which launch sites and boosters are most reliable?
  - Can we build a model that predicts landing success?

Section 1

Methodology

3. Methodology

- **Data Collection:** SpaceX API, CSVs, web scraping
- **Data Wrangling:** Merging, cleaning, encoding, standardization
- **EDA & Visualization:** Seaborn, Matplotlib, SQL queries
- **Predictive Modeling:** 4 models tested using GridSearchCV

3. Methodology - Data Collection

Data Collection – SpaceX API

- Used REST API to extract launch data
- Added CSV for coordinates and landing success

3. Methodology - Data Collection

Overview of SpaceX API Collection

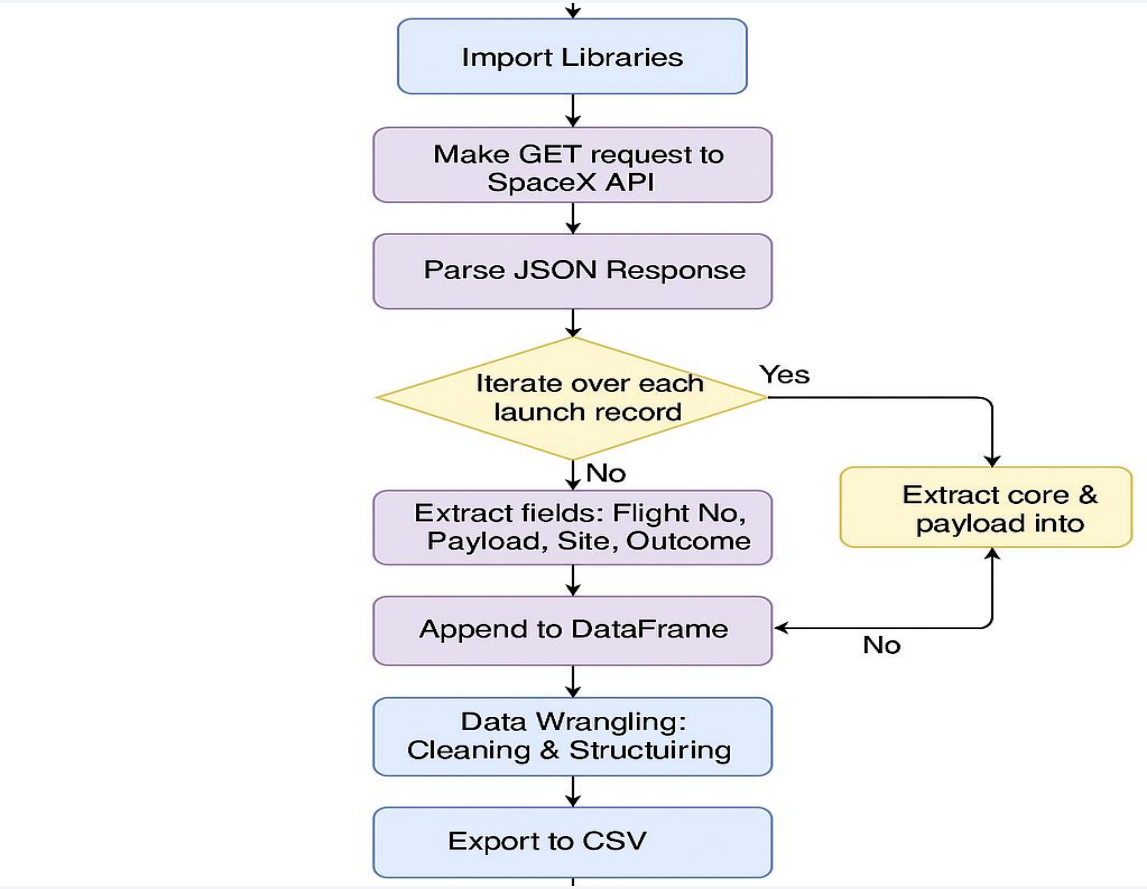
- Data sets were collected using the following steps:
  - Launch data collected via SpaceX API and CSV files
  - Launch site coordinates obtained via CSV with Lat, Long, and class fields
  - Booster features pre-encoded in a second dataset for machine learning
- The process has been elaborated on the next slide – including:
  - data collection process
  - key phrases
  - flowcharts

3. Methodology - Data Collection – SpaceX API

SpaceX API Collection Steps

- Using REST API calls from SpaceX, we extracted detailed JSON data about past Falcon 9 launches.
- We filtered data based on booster version and payloads, focusing only on Falcon 9 missions.
- The data was structured into a tabular format for further analytics and exported as a CSV."
- Key features like launch site, orbit type, payload mass, and landing success were extracted.

GitHub URL:
https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/1.jupyter-labs-spacex-data-collection-api.ipynb



3. Methodology - Data Collection

Data Scraping

- Scraped Wikipedia for Falcon 9 mission info using BeautifulSoup
- Parsed HTML table to extract mission-related attributes
- Saved structured dataset as CSV for reuse in later stage

3. Methodology - Data Collection

Overview of SpaceX API Collection

- Data sets were collected using the following steps:
  - Launch data collected via SpaceX API and CSV files
  - Launch site coordinates obtained via CSV with Lat, Long, and class fields
  - Booster features pre-encoded in a second dataset for machine learning
- The process has been elaborated on the next slide – including:
  - data collection process
  - key phrases
  - flowcharts

3. Methodology - Data Collection

Data Scraping Steps

1. Set Up Environment & Import Libraries → Install: beautifulsoup4, requests → requests, bs4, pandas, re, unicodedata
2. Define Utility Functions → Extract: date, booster version, status, mass
3. Send HTTP Request → requests.get(url)
4. Parse HTML → BeautifulSoup to navigate DOM
5. Identify Table → Find relevant HTML tables
6. Extract Rows & Structure Data → Use helper functions → Store in list/dictionary
7. Create DataFrame → pandas.DataFrame()
8. Export/Analyze → Save as CSV / analyze

GitHub Notebook:
https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/2.jupyter-labs-webscraping.ipynb

3. Methodology - Data Wrangling

Data Wrangling

- Cleaned nulls, encoded categorical variables
- Standardized numeric features using StandardScaler
- Created binary target column (Class)

3. Methodology - Data Collection

Overview of Data Wrangling

- Data preparation was carried out through the following steps:
  - Missing values were handled by cleaning or imputing null fields.
  - Categorical variables (e.g., Orbit, Launch Site, Landing Outcome) were one-hot encoded.
  - Numerical features like Flight Number, Payload Mass were standardized using StandardScaler.
  - A binary target column Class (1 = Landed, 0 = Not Landed) was created for machine learning tasks.
  - The processed dataset was saved and used for Exploratory Data Analysis and Modeling.
- The next slide presents:
  - Key wrangling techniques and logic
  - Code snippets demonstrating data transformation
  - A visual flow summarizing the wrangling process

3. Methodology - Data Wrangling

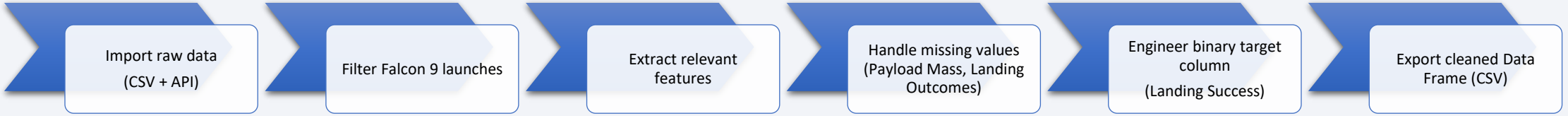
Overview of Data Wrangling Process

- We processed raw launch data from multiple sources (API & static CSV) to create a clean and structured dataset suitable for exploratory analysis and machine learning modeling.

Steps

- We harmonized inconsistent labels, standardized formats, and isolated Falcon 9 data for consistent downstream modeling.
- Landing outcomes were binarized to serve as our prediction target for classification.
- All categorical columns such as orbit type and booster version were prepared for one-hot encoding.
- The wrangling pipeline was designed to ensure reproducibility and eliminate noisy or sparse records.

GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/3.labs-jupyter-spacex-Data%20wrangling.ipynb



3. Methodology - EDA with SQL

EDA with SQL

- Queries run in SQLite to get:
- Launch site names
- Total/avg payload
- Success/Failure counts
- First successful ground landing

3. Methodology - EDA with SQL

Why SQL for EDA?

- SQL enables **efficient querying** of structured datasets.
- Allows precise **aggregation, filtering, and grouping** of launch metrics.
- Especially useful for **launch site analysis, payload filtering, and outcome breakdowns**

Steps

1. Launch Site Discovery: Queried unique launch sites to understand SpaceX’s global footprint.
2. Site-specific Records: Filtered launch records for Cape Canaveral (sites beginning with "CCA") for deeper location-based analysis.
3. Payload Mass by Customer: Aggregated payloads for major customers like NASA (CRS) to evaluate mission volumes.
4. Booster Version Performance: Computed average payload mass per booster version to assess efficiency.

3. Methodology - EDA with SQL (cont.)

Steps

5. First Ground Landing: Retrieved the earliest successful ground landing date using MIN() and filtering.
6. Drone Ship Success (Medium Payloads): Identified successful drone landings with payloads between 4000–6000 kg.
7. Mission Outcome by Site: Counted successes/failures by site to determine performance trends.
8. Top Payload Boosters: Used a subquery to list boosters that carried the maximum recorded payloads.
9. Drone Ship Failures in 2015: Isolated failed drone landings within the 2015 timeframe using string filtering.
10. Landing Outcome Rankings (2010–2017): Ranked landing outcomes across all records using GROUP BY and ORDER BY.

GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/4.jupyter-labs-eda-sql-coursera_sqllite.ipynb

3. Methodology - EDA with Data Visualization

EDA with Data Visualization

- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Success Rate by Orbit
- Time trend (2010–2020)

3. Methodology - EDA with Data Visualization

Purpose of EDA

- The goal of Exploratory Data Analysis (EDA) was to uncover patterns, assess relationships, and identify the most influential features that affect the first stage Falcon 9 landing success.

Steps

- We harmonized inconsistent labels, standardized formats, and isolated Falcon 9 data for consistent downstream modeling.
- Landing outcomes were binarized to serve as our prediction target for classification.
- All categorical columns such as orbit type and booster version were prepared for one-hot encoding.
- The wrangling pipeline was designed to ensure reproducibility and eliminate noisy or sparse records.

GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/5.edadataviz.ipynb

3. Methodology - Interactive Map (Folium)

Build an Interactive Map with Folium

- Circles + markers for launch sites
- Color-coded markers by landing outcome
- Distance to coastlines, highways calculated

3. Methodology - Build an Interactive Map (Folium)

Added Map Objects:

- Markers: Plotted each launch site and labeled them for spatial clarity.
- Circles: Highlighted launch zones with radius overlays.
- Color-coded Markers: Green for successful landings, red for failures.
- Polylines: Connected launch sites to nearby coastlines, highways, and railways to analyze proximity.

Purpose:

- To visualize geographic distribution of launch sites.
- To assess correlation between launch site success and surrounding infrastructure.

GitHub Notebook:

- https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/6.lab_jupyter_launch_site_location.ipynb

3. Methodology - Interactive Dashboard (Plotly Dash)

Build a Dashboard with Plotly Dash

- Dropdown for site → pie chart
- Payload slider → scatter plot (Payload vs. Outcome)
- Booster version visual comparison

3. Methodology - Build a Dashboard with Plotly Dash

Added Components:

- Dropdown Menu: Select launch site to filter charts.
- Pie Chart: Shows launch success distribution across all sites or selected site.
- Scatter Plot: Visualizes correlation between payload mass and launch outcome, color-coded by booster version.
- Payload Range Slider: Dynamically filters the scatter plot to view impact of payload mass on mission success.

Why These Components:

- Enables interactive exploration of SpaceX mission success by site, payload, and booster version.
- Helps reveal patterns in launch performance based on key mission variables.

GitHub Notebook:

- https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/7.spacex-dash-app.py

3. Methodology - Predictive Analysis

Predictive Analysis (Classification)

- Used GridSearchCV with 4 models:
    - Logistic Regression
    - SVM (Sigmoid)
    - Decision Tree (Best)
    - KNN
- Metrics captured: Accuracy, Best Params, False Positives

3. Methodology - Predictive Analysis

Classification Model Development Overview:

- Data Preprocessing: Standardized features using StandardScaler; target label (Class) extracted.
- Data Splitting: 80/20 train-test split using train_test_split.
- Model Selection & Tuning:
    - Logistic Regression: Tuned with GridSearchCV for C, penalty, solver.
    - SVM: Explored multiple kernels, C, and gamma values.
    - Decision Tree: Tuned criterion, splitter, depth, leaf samples.
    - KNN: Tuned number of neighbors and distance metrics.

3. Methodology - Predictive Analysis

Classification Model Development Overview (cont.):

Evaluation:

- Used accuracy_score and confusion_matrix for test evaluation.
- Best model: Decision Tree (Validation Accuracy ~ 88.75%), followed by SVM and Logistic Regression.

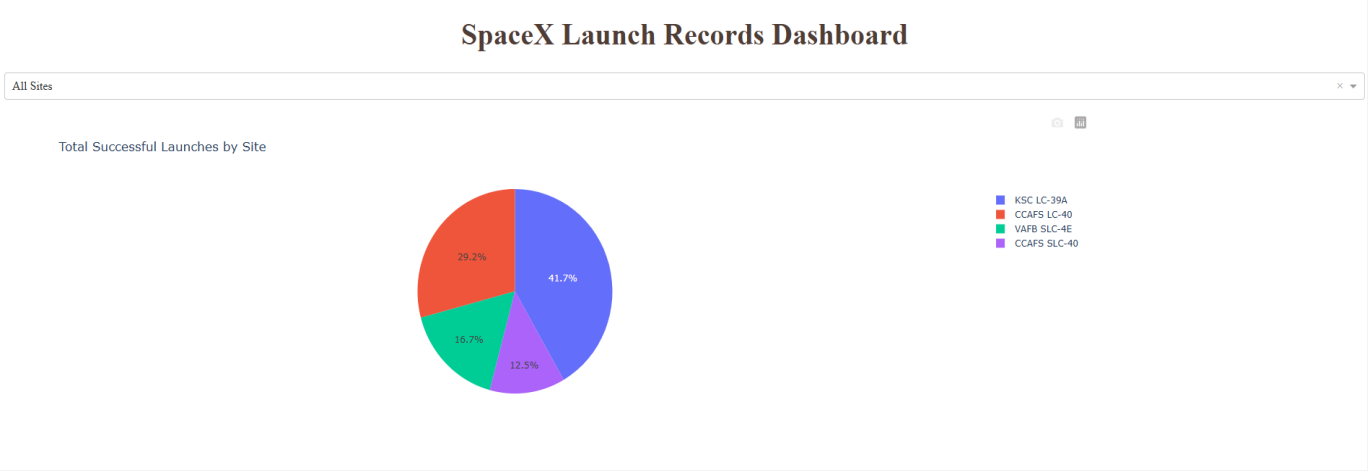
Why This Process:

- Ensures robustness via cross-validation and hyperparameter tuning.
- Provides explainable and measurable performance across classifiers.

GitHub Notebook:

- https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/8.SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

4. Results - Dashboard



Plotly Dashboard Overview

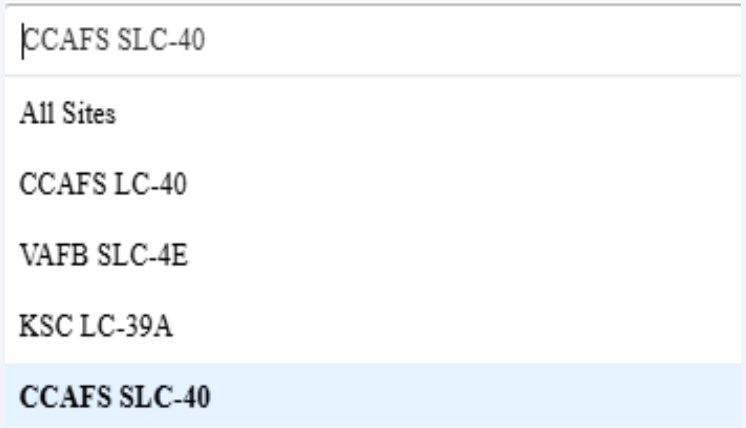
- **Description**
  - Dropdown view of pie chart
  - Slider effect on scatter plot
- **Notes**
  - The user can adjust the payload to the selected value to get the booster version for the highest success rate.

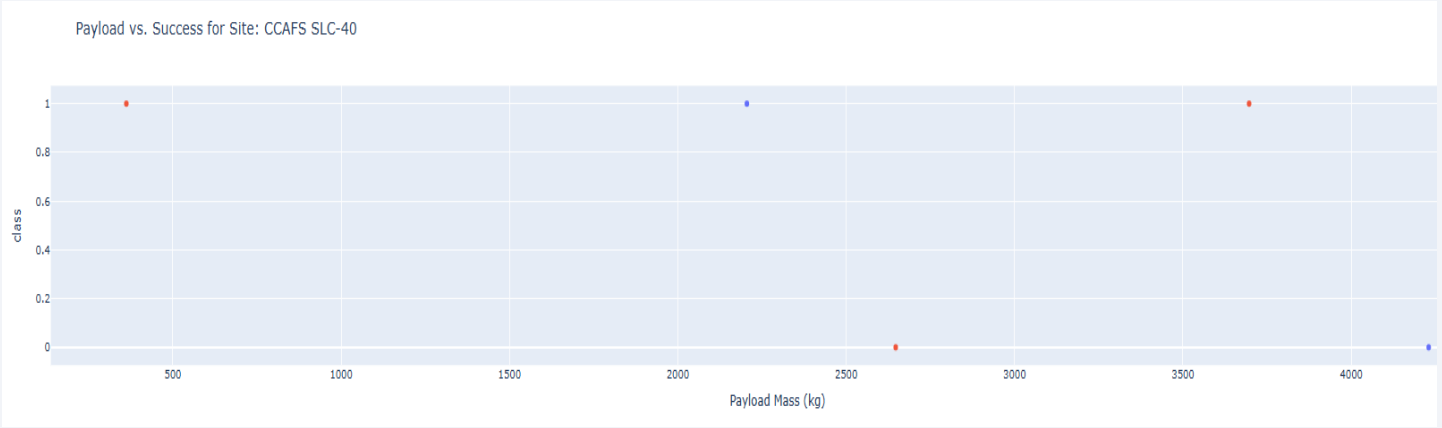
Results explained in next slide

4. Results - Dashboard

Interactive analytics demo in screenshots (Plotly Dashboard)

- **Use-case Scenarios:**
  - Selecting ‘CCAFS SLC-40’ in dropdown shows 42% success rate for recent launches.
  - Adjusting payload slider to [0–5000] shows success threshold with more success for Booster B4.

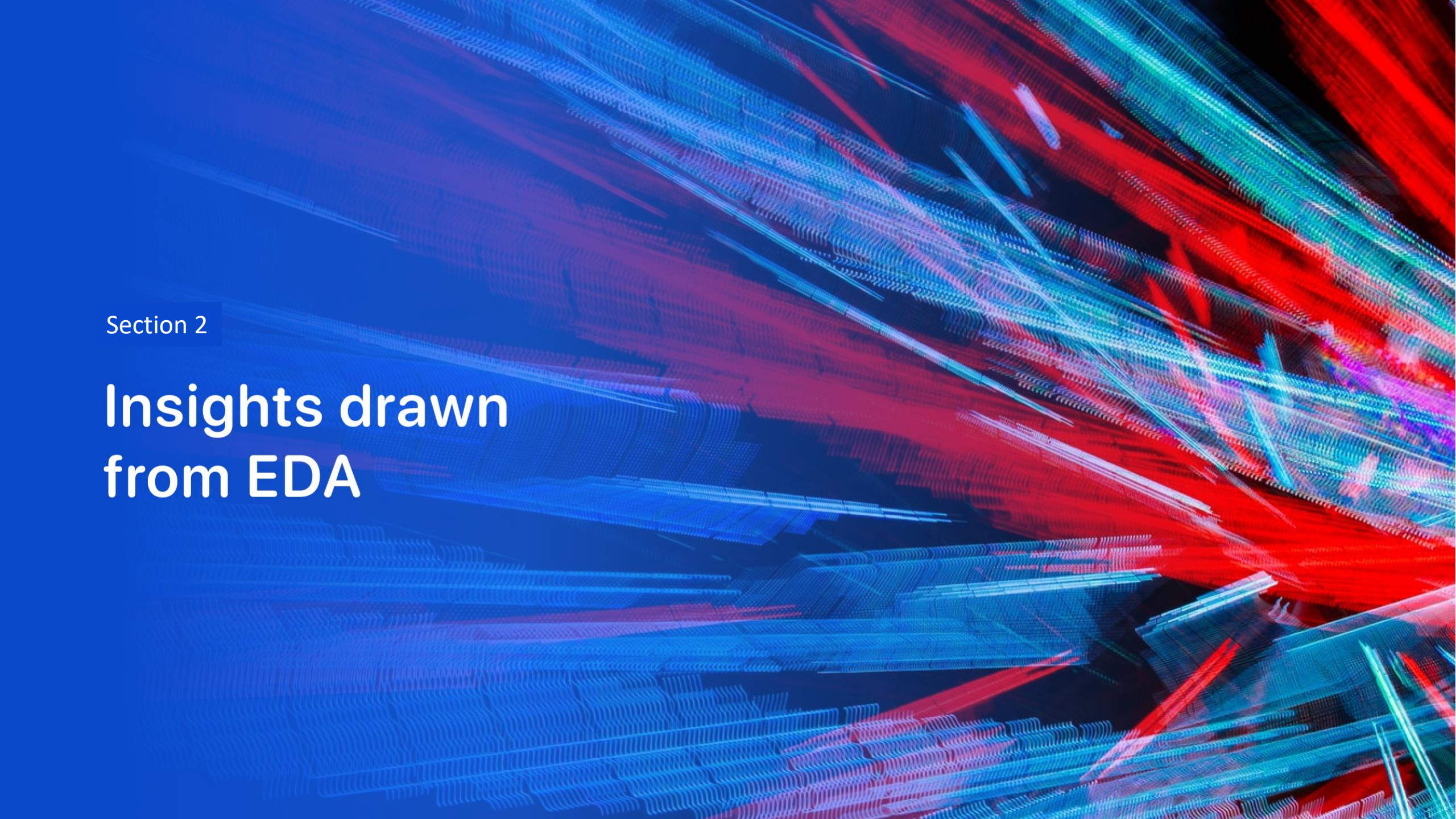




4. Results - Predictive Models

Tabular Summary of Predictive Model Results

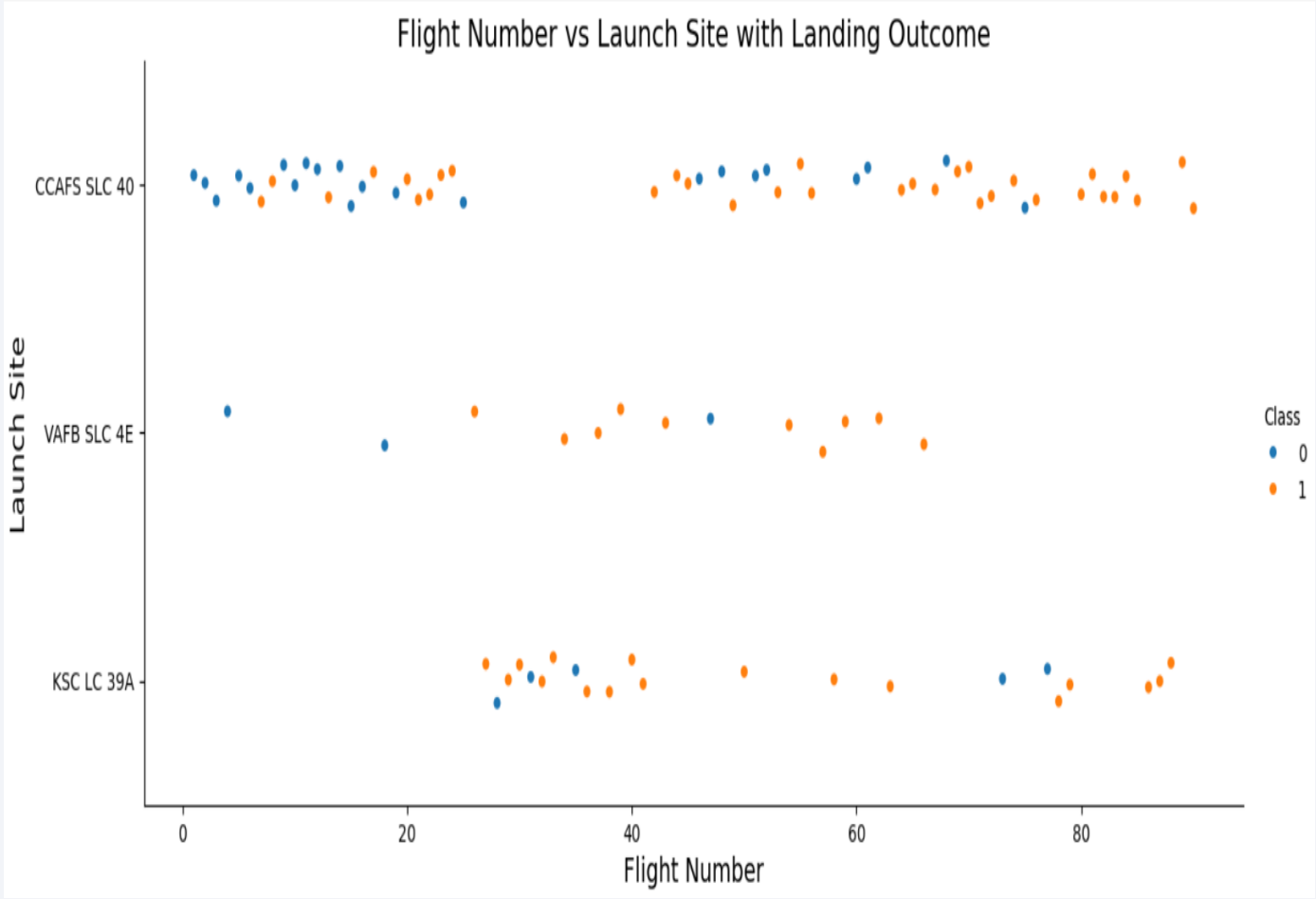
Model	Accuracy	Best Params	False Positives
Logistic Regression	83.30%	C=1, penalty='l2'	3
SVM	83.30%	kernel='sigmoid', gamma=0.03	2
Decision Tree	83.30%	entropy, depth=16, sqrt split	1
KNN	83.30%	n_neighbors=10, p=1	3


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

Scatter plot of Flight Number vs. Launch Site

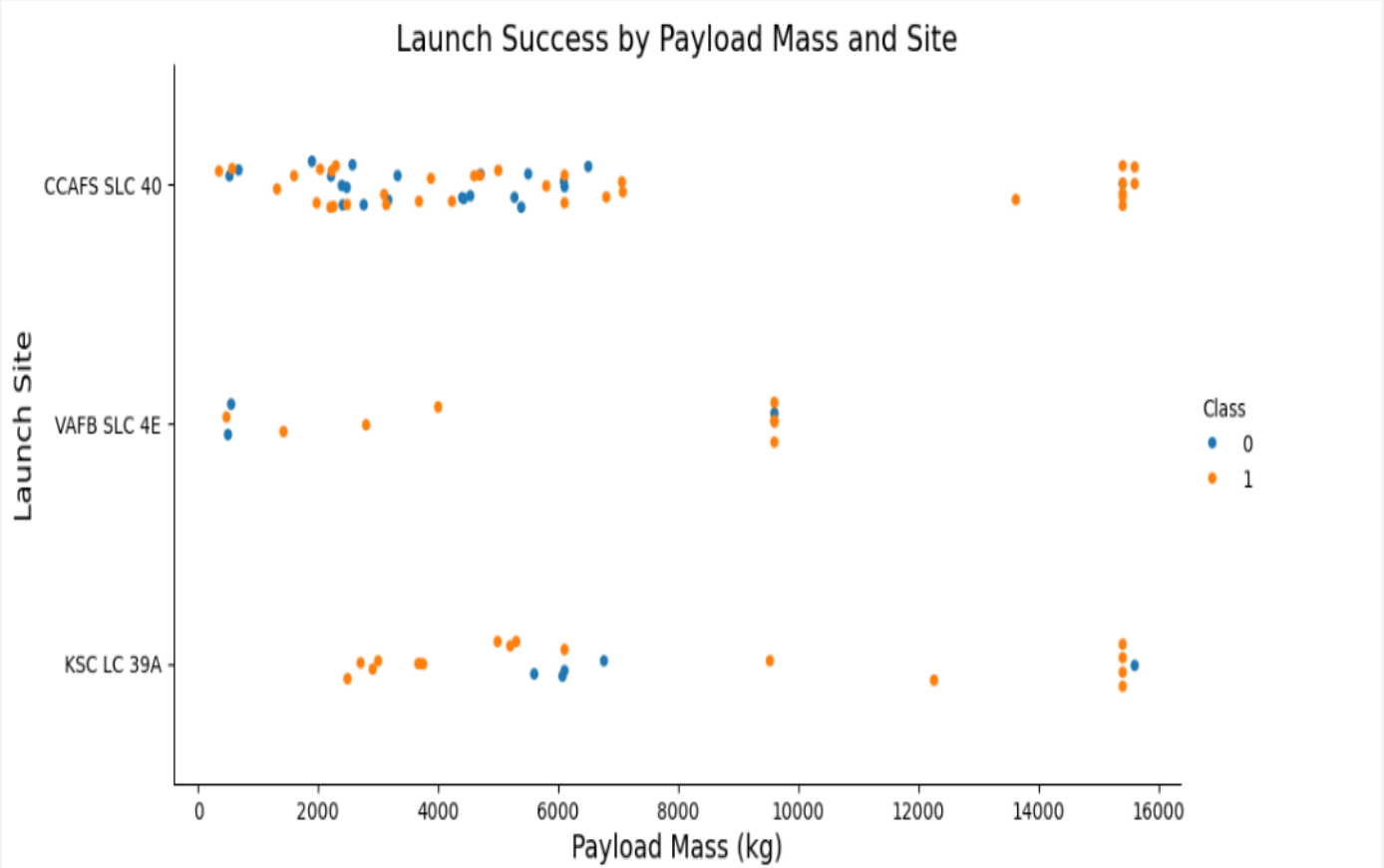
- CCAFS SLC 40 has the highest number of launches, showing a wide range of flight numbers from early to late missions.
- KSC LC 39A and VAFB SLC 4E started launching more in mid-to-later missions.
- Over time, success rates increase, especially evident at CCAFS.



Payload vs. Launch Site

Scatter plot of Payload vs. Launch Site

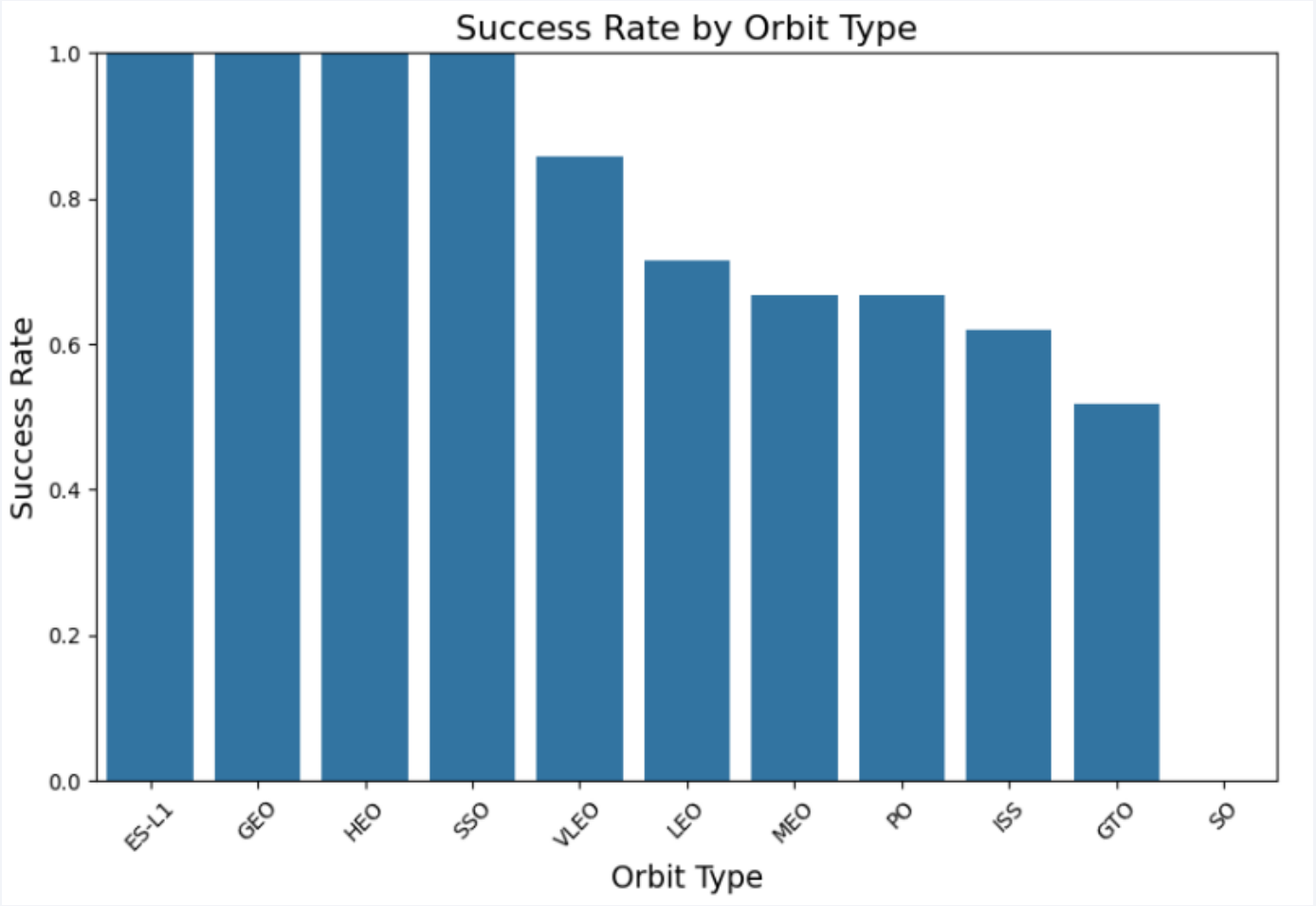
- KSC LC 39A handles higher average payloads (mean ~7606 kg), possibly due to heavier missions.
- VAFB SLC 4E shows a tight range of high-payload missions but fewer launches overall.
- CCAFS SLC 40 is more diverse, showing success in both low and high payload ranges (up to 15600 kg).



Success Rate vs. Orbit Type

Bar-chart for the success rate of each orbit type

- Perfect Success (100%): ES-L1, GEO, HEO, and SSO — reliable but infrequent missions.
- High Success (~85%): VLEO — lower altitude aids recovery.
- Moderate Success (66–71%): LEO, MEO, PO — common orbits with varying challenges.
- Lower Success (~52–62%): ISS and GTO — complex or high-payload missions.
- Zero Success: SO orbit — no successful landings recorded.



Flight Number vs. Orbit Type

Scatter point of Flight number vs. Orbit type

Patterns Observed:

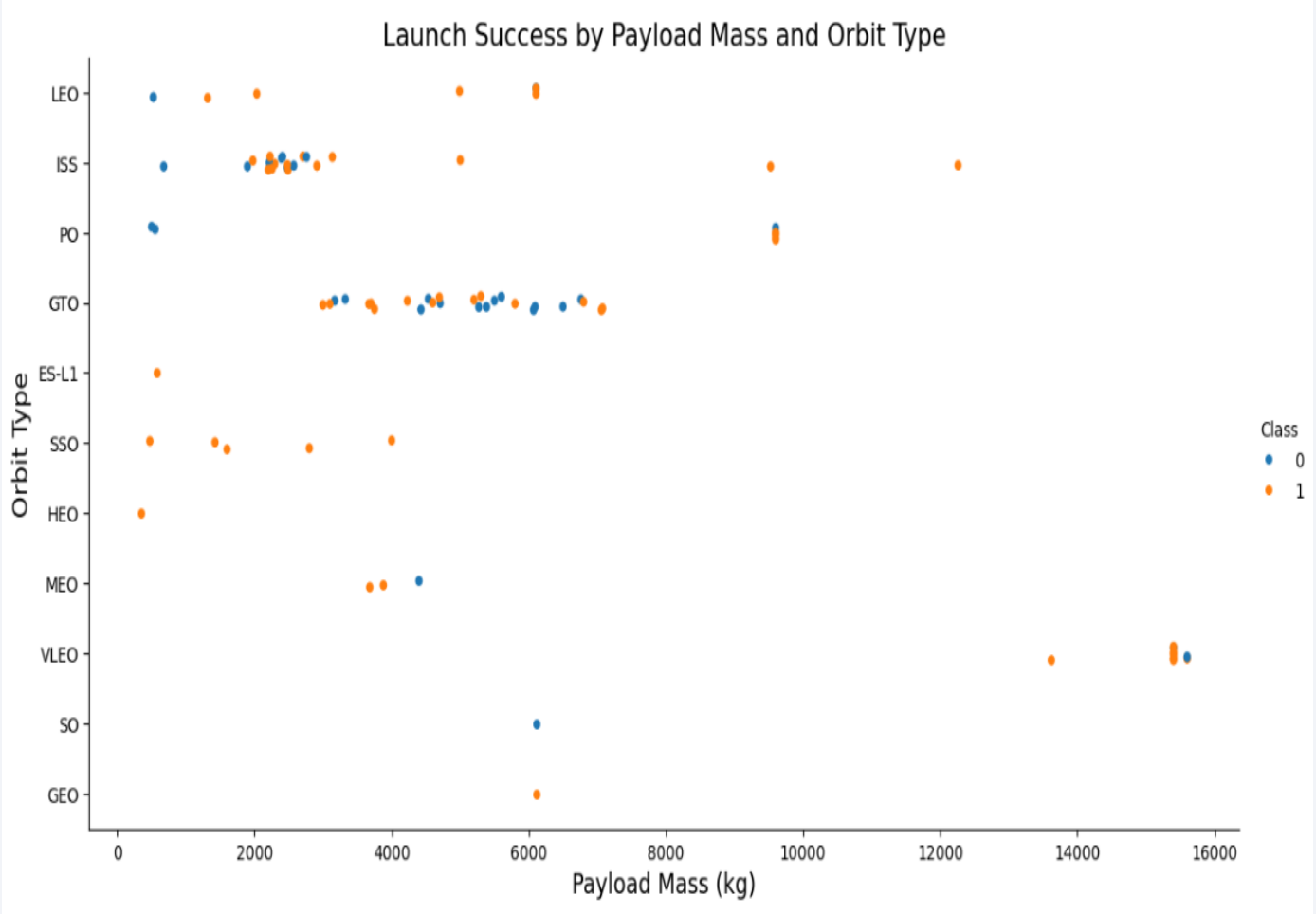
- LEO & ISS: Frequent early launches with growing success over time.
- GTO & VLEO: Mixed outcomes in mid-to-late flights — reflect challenges at higher altitudes.
- ES-L1, GEO, HEO: Rare but 100% success — show precise, targeted missions.

Trend: Higher flight numbers correlate with improved success across most orbits.

Payload vs. Orbit Type

Scatter point of payload vs. orbit type

- ***GTO Orbits:*** Show the widest payload range (~3000–7000+ kg) with mixed success—highlighting challenges of heavier payloads at high altitudes.
- ***VLEO & LEO Orbits:*** Cluster at higher payloads (~6000+ kg) with generally good success rates, particularly in VLEO.
- ***ISS Orbits:*** Spread across moderate payloads (~1000–6000 kg) with both successes and failures, reflecting complexity of docking/precision.
- ***SSO, ES-L1, HEO, GEO:*** Typically involve low-to-mid payloads, and each shows perfect success — suggesting they are highly controlled, low-frequency missions.

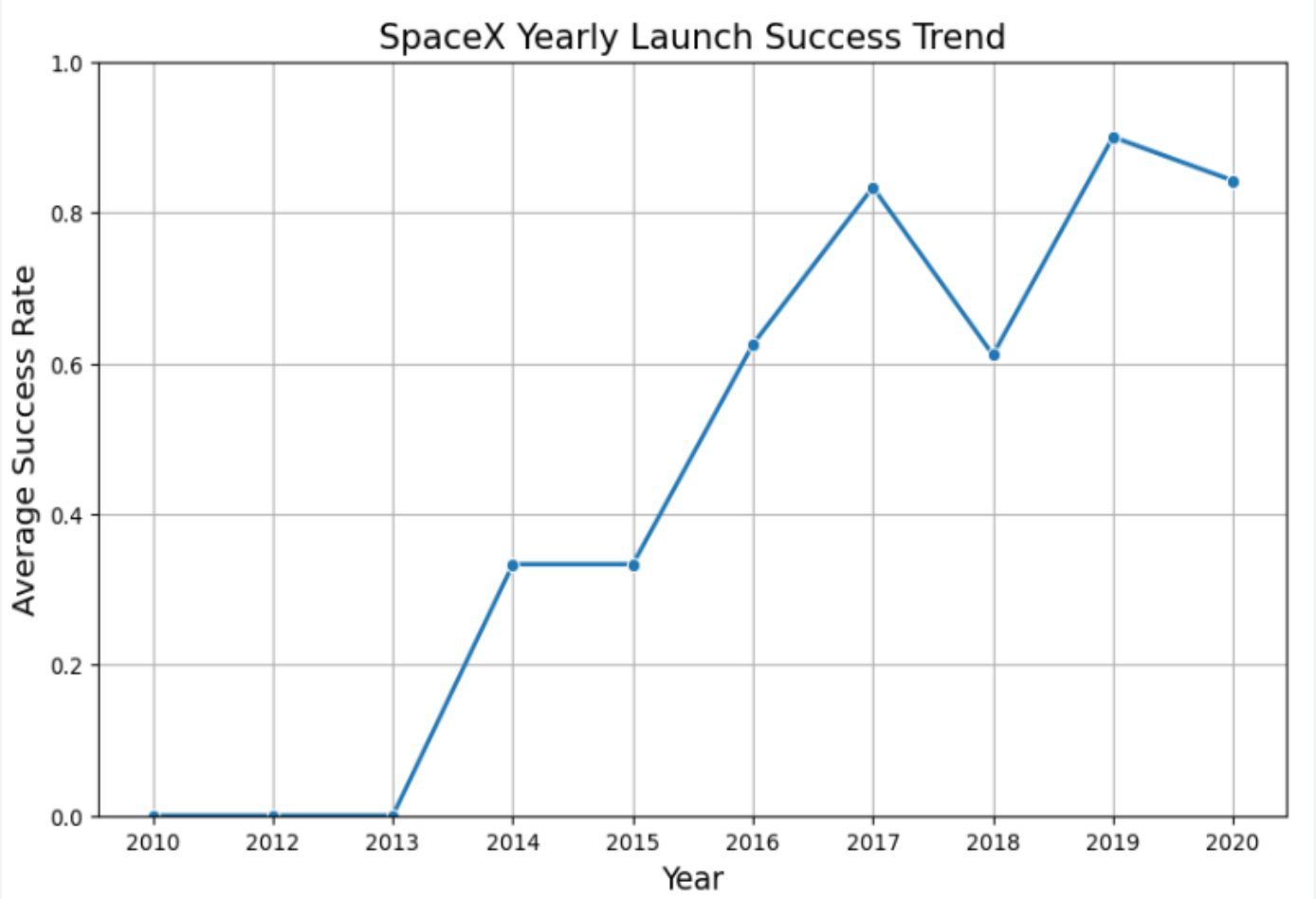


Launch Success Yearly Trend

Line-chart of yearly average success rate

- **Trend:** Clear improvement in launch success rates over time.
- **Early years (2010–2014):** Low to moderate success rates — initial phase of SpaceX learning.
- **Post-2015:** Significant rise in success rates, reaching consistently high performance (above 80%).

Insight: As flight numbers and experience grew, SpaceX drastically improved reliability.



All Launch Site Names

Names of the unique launch sites

- SpaceX primarily operated from four major launch sites across

1. ***Cape Canaveral***,
2. ***Kennedy Space Center***, and
3. ***Vandenberg Air Force Base*** during the period analyzed.

Task 1

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

In [11]:
```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`

- **What was done:**
  Queried the SpaceX dataset to extract the **first 5 launches** at **Cape Canaveral sites**.
- **What was found:**
  All 5 records are from **"CCAFS LC-40"**, one of the most frequently used SpaceX pads at Cape Canaveral.

Insight: Cape Canaveral served as a **major operational base** for early Falcon 9 launches.

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
In [12]: %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

Out[12]:

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

Total payload carried by boosters from NASA

- **What was done:** Summed the payload mass for all launches where NASA was involved as a customer (including missions labeled as "NASA (CRS)", "NASA (COTS)", etc.).
- **What was found:** Boosters carried a total of 45,596 kilograms of payload for NASA missions.

Insight: NASA represents a key client for SpaceX — significant payload mass reflects important cargo deliveries (e.g., ISS resupply).

Task 3

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

In [13]:
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

Out[13]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

Calculating the average payload mass carried by booster version F9 v1.1

- **What was done:** Calculated the average payload mass for all flights using the booster version F9 v1.1.
- **What was found:** The average payload mass was approximately 2928.4 kilograms for the F9 v1.1 booster version.

Insight: The F9 v1.1 booster was primarily used for medium-lift missions, making it a transitional model before the introduction of the more powerful F9 Full Thrust versions.

Task 4

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

In [14]:
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

Out[14]:

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

Selecting the dates of the first successful landing outcome on ground pad

- **Explanation:**
  - We use the MIN() SQL function to find the earliest (smallest) date value.
  - We filter the data where the "Landing_Outcome" is exactly 'Success (ground pad)', meaning a successful vertical landing on land (not ocean or drone ship).
  - This gives us the first ever successful ground landing achieved by SpaceX from the available dataset.
- **Importance:**
  - SpaceX's first successful ground landing was a historic milestone demonstrating full mission reusability [1].
  - It marked a breakthrough toward dramatically lowering launch costs [2].

Task 5

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

Hint:Use min function

```
In [15]: %sql SELECT MIN(Date) AS First_Successful_Ground_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
 * sqlite:///my_data1.db
Done.
Out[15]: First_Successful_Ground_Landing
                                2015-12-22
```


Sources

1. https://spaceflightnow.com/2015/12/23/spacex-rocket-landing-applauded-but-experts-say-implications-tbd/
2. https://www.space.com/31420-spacex-rocket-landing-success.html
3. https://en.wikipedia.org/wiki/SpaceX_reusable_launch_system_development_program

Successful Drone Ship Landing (Payload 4000 & 6000)

Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

- **What was done:** Filtered flights where the landing was a success on a drone ship and the payload mass was between 4000 kg and 6000 kg.
- **What was found:** The following booster versions successfully landed on drone ships while carrying medium-weight payloads:
    - **1)** F9 FT B1022; **2)** F9 FT B1026; **3)** F9 FT B1021.2; **4)** F9 FT B1031.2

Insight: These boosters were among the early successful recoveries, proving SpaceX's ability to reliably recover rockets after medium-weight missions.

Out[16]:

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

Total Number of Successful & Failure Mission Outcomes

Calculating the total number of successful and failure mission outcomes

- **What was done:** Grouped all records by the *Mission_Outcome* field and counted how many missions fell into each outcome type.
- **What was found:**
  - Success: 98 missions
  - Failure (in flight): 1 mission
  - Success (payload status unclear): 1 mission

Insight: SpaceX achieved a very high mission success rate overall. Only 1 outright failure was recorded, and 1 mission had a partial success (payload uncertainty).

Task 7

List the total number of successful and failure mission outcomes

In [17]:
```
%sql SELECT "Mission_Outcome", COUNT(*) AS Total FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

Out[17]:

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

Boosters Carried Maximum Payload

List of names of the booster which have carried the maximum payload mass

QUERY: %sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);

- **What was done:** Identified the booster versions that carried the heaviest payload mass recorded in the dataset (15600 kg).
- **What was found:** Multiple F9 Block 5 boosters (e.g., B1048, B1049, B1051, B1056, B1060, B1058) consistently achieved this maximum payload capacity.

Insight: SpaceX's F9 Block 5 series is the most capable version, demonstrating high reusability and heavy-lift performance.

2015 Launch Records

List of failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

- **What was done:** Filtered SpaceX launches for 2015 where the landing on drone ships failed.
- **What was found:**
  - Two failures occurred in 2015.
  - Both failures involved Falcon 9 v1.1 booster versions (B1012 and B1015) launched from CCAFS LC-40 (Cape Canaveral Air Force Station Launch Complex 40).

Insight: Early drone ship landing attempts in 2015 faced challenges, as SpaceX was still refining precision landing techniques.

Ranking Landing Outcomes (2010-06-04 & 2017-03-20)

Ranked count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

QUERY: %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;

- **What was done:** Ranked all landing outcomes between 2010-06-04 and 2017-03-20 by their frequency, in descending order.

- **What was found:**
  - **Most common outcome:** "No attempt" (10 launches) — SpaceX initially didn't attempt landings in early missions.
  - **Equal success and failure:** 5 "Success (drone ship)" and 5 "Failure (drone ship)", reflecting the experimental nature of drone ship landings.
- Ground pad successes: 3 "Success (ground pad)" — a key milestone towards reusable rockets.

Insight: Early Falcon 9 missions focused more on successful launches rather than landings, and gradually incorporated drone ship and ground pad recoveries.

Section 3

Launch Sites Proximities Analysis

Site Locations Visualized using Folium Map

Screenshots of the generated folium map including all launch sites' location markers on the global map

Key Elements Mapped:

- *Launch Site Markers:*
  - Each SpaceX launch site (e.g., CCAFS LC-40, KSC LC-39A, VAFB SLC-4E) is represented using marker pins.
  - Clicking a marker shows a popup displaying the launch site name.
- *Launch Site Distribution:*
  - Eastern Florida cluster (CCAFS + KSC) — ideal due to Earth's rotation advantage for eastward launches.
  - California site (VAFB) supports polar orbit missions needing different launch trajectories.

Launch Outcomes Visualized using Folium

Exploration of the folium map to show the color-labeled launch outcomes on the map

Marker Clusters for Launch Outcomes:

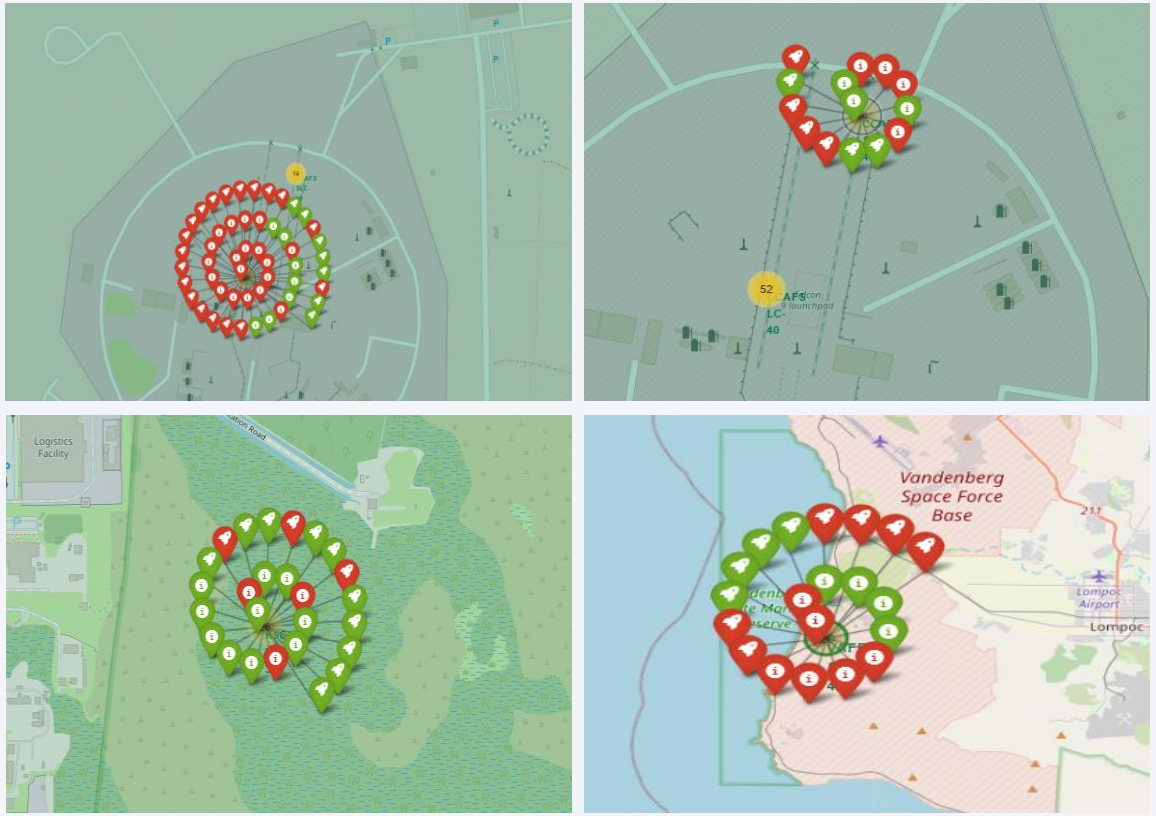
- Each launch (success or failure) is **color-coded**:
  - **Green** marker for successful landings.
  - **Red** marker for failed landings.

Clustered markers help avoid map overcrowding in high-traffic sites like CCAFS.

Insights

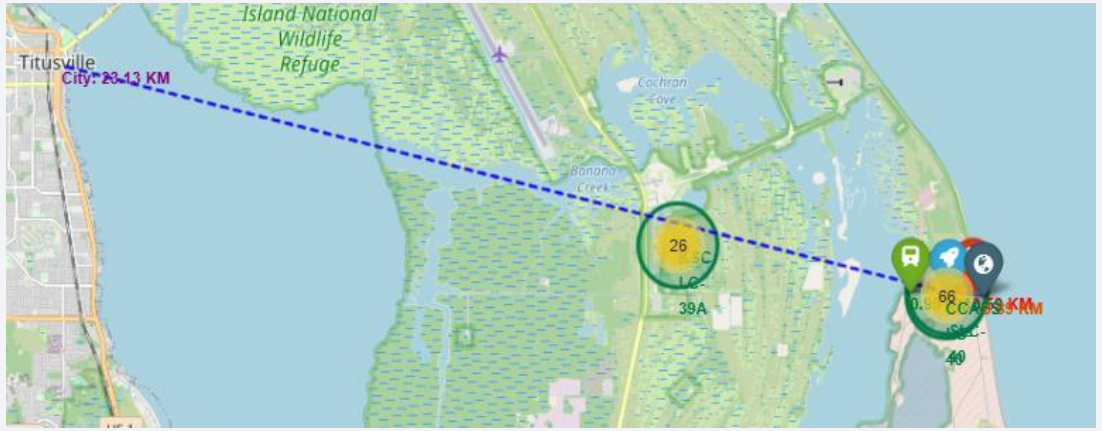
Landing Recovery:

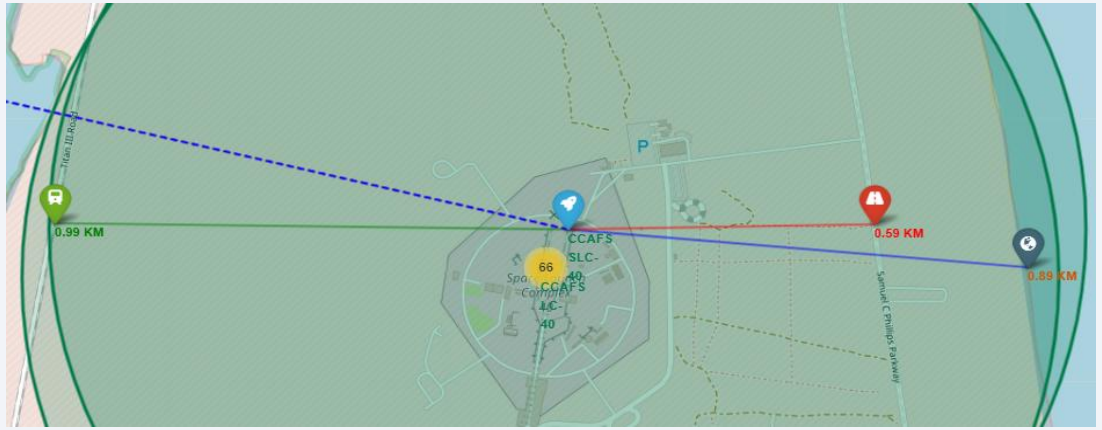
- Many failed landings (marked red) cluster around the **early flights** and decrease over time.
- Successful landing rates (green) increase with later launches, suggesting operational improvements.

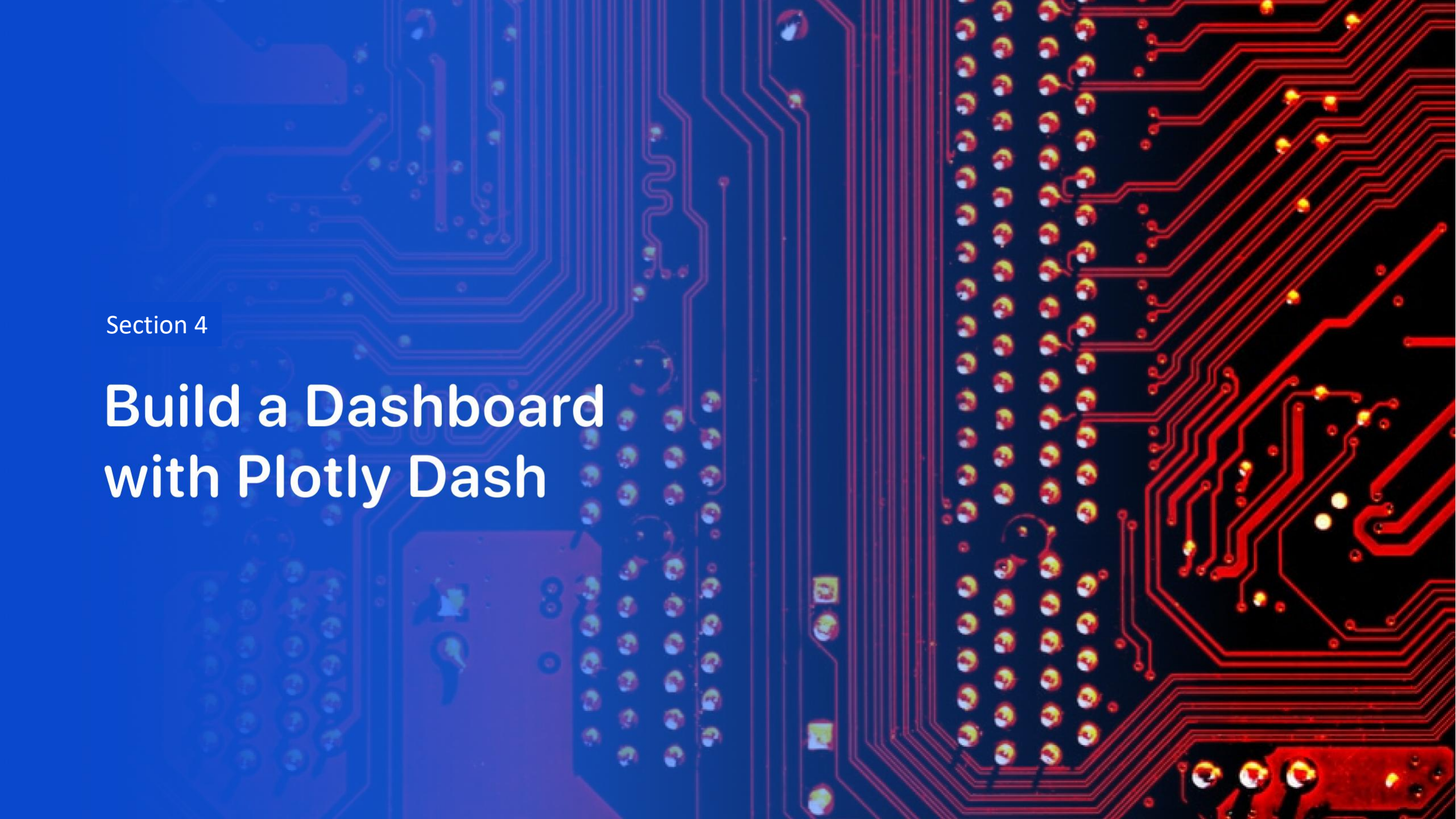


Infrastructural Proximities Visualized using Folium

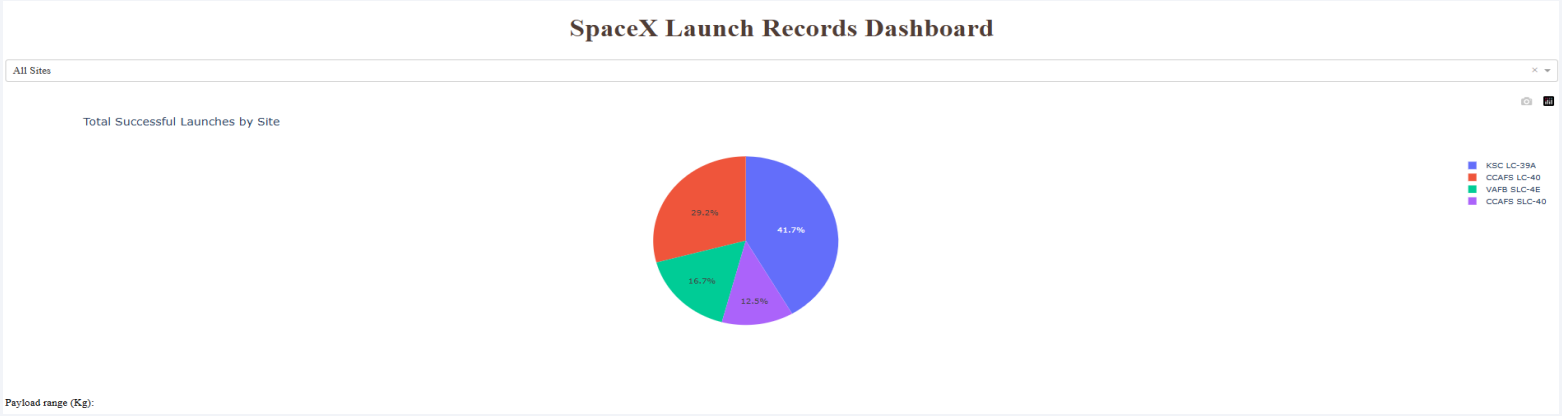
- Exploration of generated folium map and selected launch site to its proximities such as **railway**, **highway**, **coastline**, with distance **calculated and displayed**
- **Coastal Proximity**: All major launch sites are very close to coastlines (~1–5 km) — minimizing risk to populated areas during launch/landing.
- **Infrastructure Support:** Most launch sites are near highways and railways, aiding logistics for transporting rockets and heavy payloads.
- **Nearest City:** The nearest city from launch site CCAFS SLC-40 was ~ 23km






Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches by Site – Pie Chart Overview



This pie chart visualizes the proportion of total successful Falcon 9 launches across different SpaceX launch sites.

Key Elements:

- ***Slices:*** Each slice represents a different launch site.
- ***Size of Slice:*** Proportional to the number of successful launches recorded at that site.
- ***Color Coding:*** Different launch sites are distinguished with unique colors.

Important Findings:

- KSC LC-39A and CCAFS SLC-40 dominate the success records, reflecting SpaceX's preference and operational success at these sites.
- VAFB SLC-4E has comparatively fewer launches but contributes meaningfully to successful missions on the West Coast.
- The pie chart immediately highlights operational hotspots for SpaceX, useful for understanding resource allocation and regional mission frequency.

Success vs. Failure at the Best Performing Launch Site

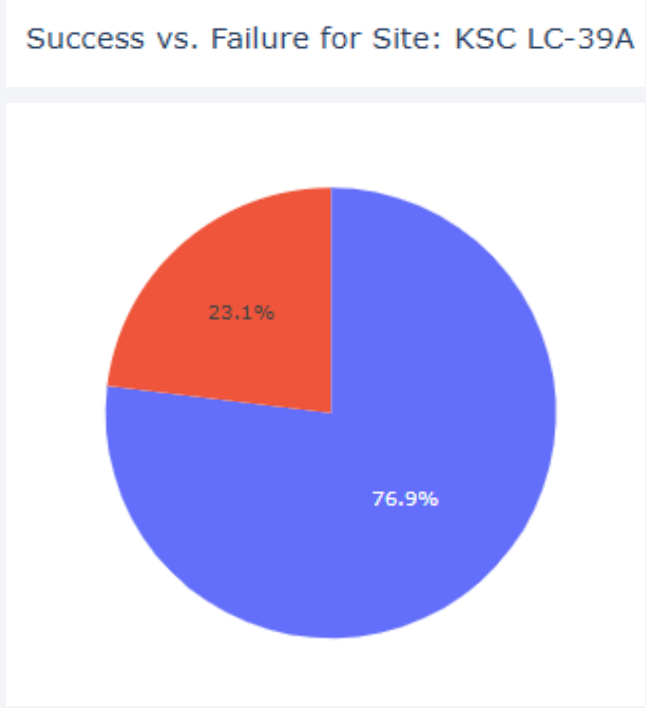
This pie chart shows the distribution of successful vs. failed launches specifically for the launch site with the highest launch success rate.

Key Elements:

- ***Two Segments:***
  - One segment represents successful landings (Class = 1)
  - The other segment represents failed landings (Class = 0).
- ***Color Coding:***
  - Typically green for success and red for failure (or another clear distinction)
  - Percentage/Count Labels: Shown either inside or on hover, revealing exact success vs failure ratio.

Important Findings:

- The selected launch site (likely KSC LC-39A based on past analysis) demonstrates a very high success rate — success slices dominate the chart.
- Only a small fraction of launches failed, indicating operational maturity and reliability improvements over time.
- Highlights that specific launch infrastructure and operational conditions play a crucial role in mission success.



Payload Mass vs. Launch Outcome Across All Sites

The scatter plot visualizes how payload mass (kg) correlates with launch success (Class) across all SpaceX launch sites.. The range slider allows selecting different payload mass ranges to dynamically filter the view.

Key Elements: X-axis: Payload Mass (kg); Y-axis: Launch Outcome (0 = Failure, 1 = Success)

- **Color:** Booster Version Category (each version shown in a different color)
- **Interaction:** Payload range selection dynamically updates the scatter distribution.

Important Findings:

- Lower to mid payloads (0–6000 kg) show higher concentration of successful launches.
- Very heavy payloads (above ~7000 kg) display more variability — some successes, some failures.

Booster Versions:

- Newer versions (like Falcon 9 Block 5) show higher reliability even at higher payloads.
- Older versions are clustered towards lighter payloads and have lower success rates.

Payload Sweet Spot:

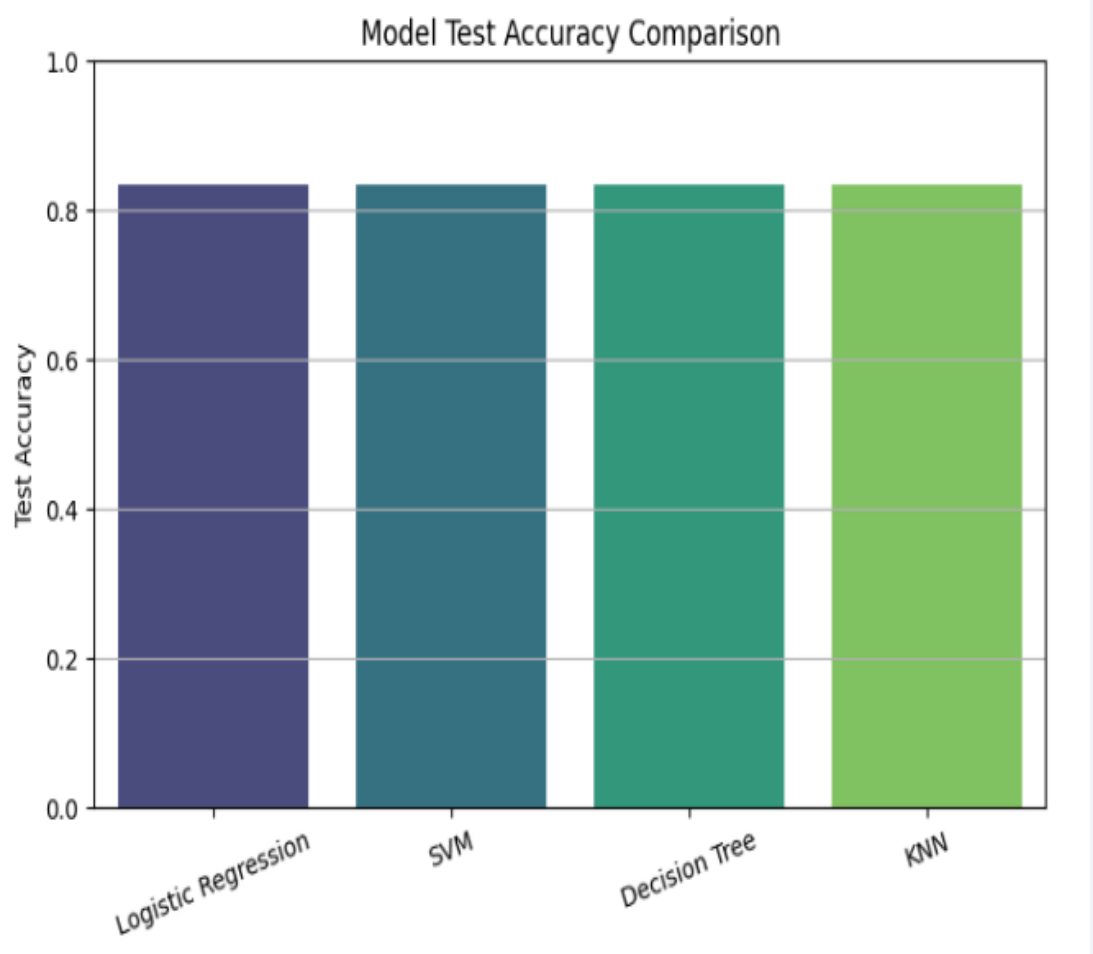
- Payloads between 2000–6000 kg generally have the highest success rates across boosters.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All four models — **Logistic Regression, SVM, Decision Tree, and KNN** — achieved comparable test accuracies of approximately **83.33%**.
- While none significantly outperformed the others on the test set, the **Decision Tree Classifier** stood out with the highest cross-validation accuracy of **~88.75%** during hyperparameter tuning.
- This stronger validation performance makes it the **preferred model** when balancing both generalization and test reliability.

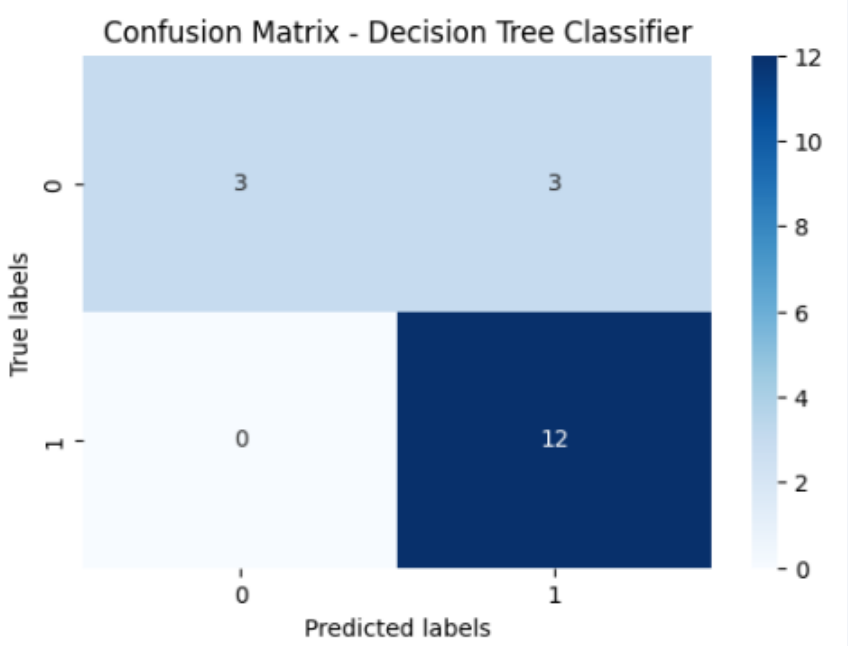


Confusion Matrix

Confusion Matrix Interpretation – Decision Tree Classifier

Explanation:

- ***True Positives (12):*** The model correctly predicted **12 landings** (actual landings classified as landings).
- ***True Negatives (3):*** The model correctly predicted **3 non-landings** (actual failures classified as failures).
- ***False Positives (3):*** The model incorrectly predicted **3 cases as landings** when they were actually not landed.
- ***False Negatives (0):*** No actual landings were missed — the model never predicted a landing failure incorrectly.



Classification Report - Decision Tree Classifier

	precision	recall	f1-score	support
Not Landed	1.00	0.50	0.67	6
Landed	0.80	1.00	0.89	12
accuracy			0.83	18
macro avg	0.90	0.75	0.78	18
weighted avg	0.87	0.83	0.81	18

Conclusions

- **Landing success prediction is feasible:** Through careful feature engineering, EDA, and machine learning modeling, SpaceX Falcon 9 first-stage landings could be reliably predicted with ~83% accuracy.
- **Decision Tree Classifier recommended:** Despite all models achieving similar test set performance, the Decision Tree demonstrated the highest cross-validation accuracy (~88.75%), indicating better generalization.
- **Operational insights gained:** EDA revealed that certain launch sites (e.g., KSC LC-39A), payload ranges (2000–6000 kg), and newer booster versions (Block 5) correlate with higher success rates.
- **SpaceX's learning curve evident:** Launch success improved significantly over time (especially post-2015), emphasizing the impact of cumulative flight experience and technological iteration.
- **Scope for further improvement:** Future models could integrate more temporal, environmental (e.g., weather), and mission complexity features to further enhance predictive accuracy.

Appendix

Python Code Snippets

- GridSearchCV hyperparameter tuning
  - *Logistic Regression (A)*
  - *SVM (B)*
  - *Decision Tree (C)*
  - *KNN (D)*

A

```
# Create Logistic Regression object
lr = LogisticRegression()

# Apply GridSearchCV
logreg_cv = GridSearchCV(lr, parameters, cv=10)

# Fit model on training data
logreg_cv.fit(X_train, Y_train)
```

B

```
[18]: # Define hyperparameters grid
parameters = {
    'kernel': ('linear', 'rbf', 'poly', 'sigmoid'),
    'C': np.logspace(-3, 3, 5),
    'gamma': np.logspace(-3, 3, 5)
}

# Create Support Vector Machine model
svm = SVC()
```

```
[19]: # Create GridSearchCV object with 10-fold cross validation
svm_cv = GridSearchCV(svm, parameters, cv=10)

# Fit to training data
svm_cv.fit(X_train, Y_train)
```

C

```
[23]: # Define parameter grid
parameters = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [2*n for n in range(1,10)],
    'max_features': ['sqrt'],  # 'auto' causes error; using 'sqrt'
    'min_samples_leaf': [1, 2, 4],
    'min_samples_split': [2, 5, 10]
}

# Create Decision Tree Classifier object
tree = DecisionTreeClassifier()
```

```
[24]: # Create GridSearchCV object
tree_cv = GridSearchCV(tree, parameters, cv=10)

# Fit on training data
tree_cv.fit(X_train, Y_train)
```

D

```
[28]: # Define parameter grid
parameters = {
    'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
    'p': [1, 2]  # p=1 (Manhattan), p=2 (Euclidean)
}

# Create KNN classifier object
KNN = KNeighborsClassifier()
```

```
[29]: # Perform GridSearchCV
knn_cv = GridSearchCV(KNN, parameters, cv=10)
knn_cv.fit(X_train, Y_train)
```

Appendix

Python Code Snippets

- Evaluate Model function (Confusion Matrix + Classification Report Plotter)

Defining Evaluate Function for All Models

```
def evaluate_model(y_true, y_pred, model_name="Model"):
    """
    Evaluates a classification model using a confusion matrix and classification report.
    """
    from sklearn.metrics import confusion_matrix, classification_report
    import seaborn as sns
    import matplotlib.pyplot as plt

    # Plot Confusion Matrixb
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(6,4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
    plt.title(f'Confusion Matrix - {model_name}', fontsize=14)
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

    # Print Classification Report
    print(f"Classification Report - {model_name}")
    print(classification_report(y_true, y_pred, target_names=['Not Landed', 'Landed']))
```

Appendix

SQL Queries

- Top payload boosters

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

- Drone ship landings

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

- Failed landings in 2015

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

[19] Python

- Launch site discovery, etc.

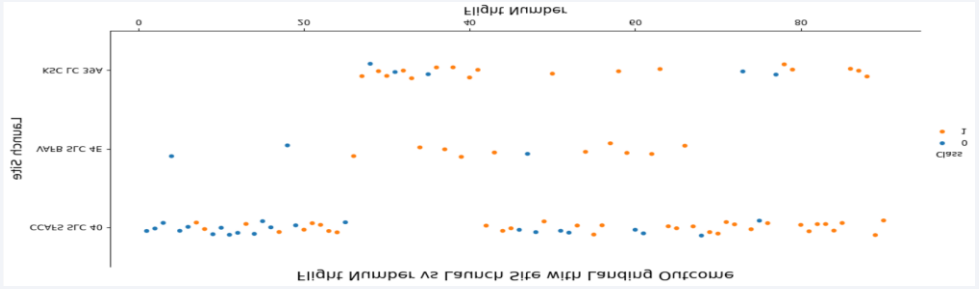
```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

[11] Python

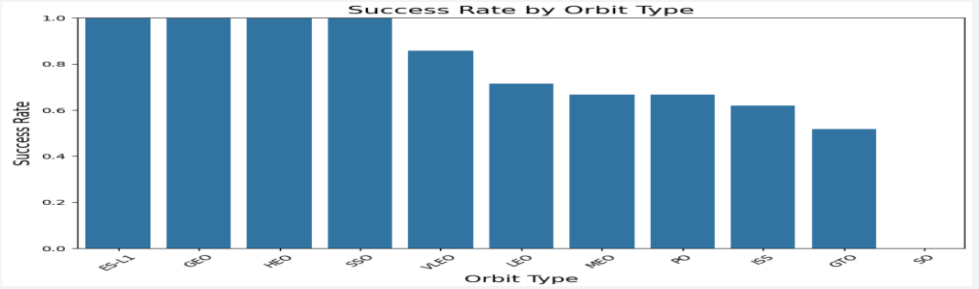
Appendix

Charts & Visualizations

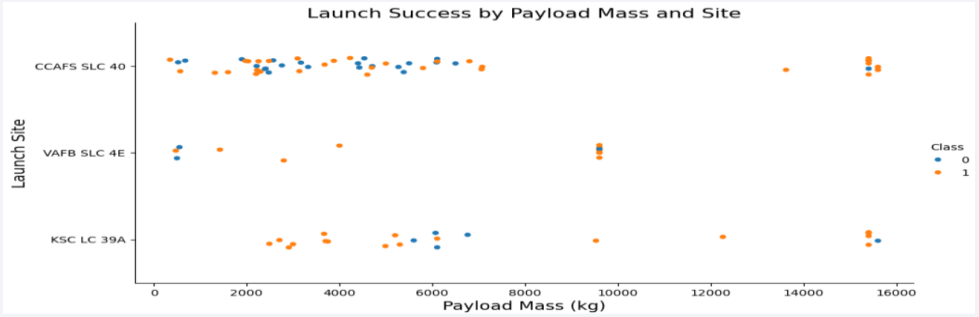
- Flight Number vs Launch Site scatter plot



- Success Rate by Orbit (bar chart)



- Payload vs Launch Site scatter plot

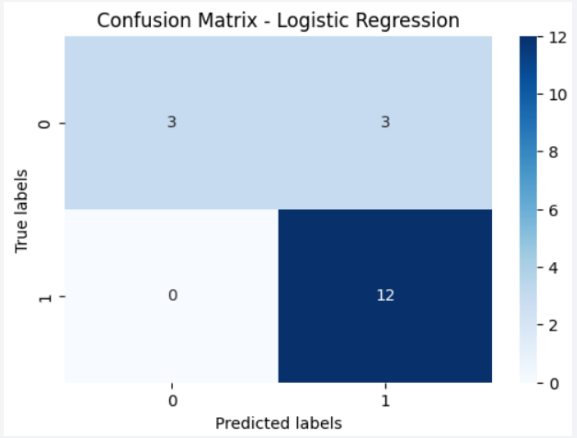


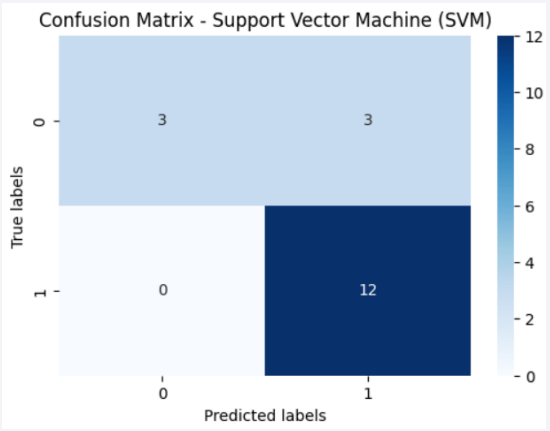
- Launch Success Yearly Trend (line chart)

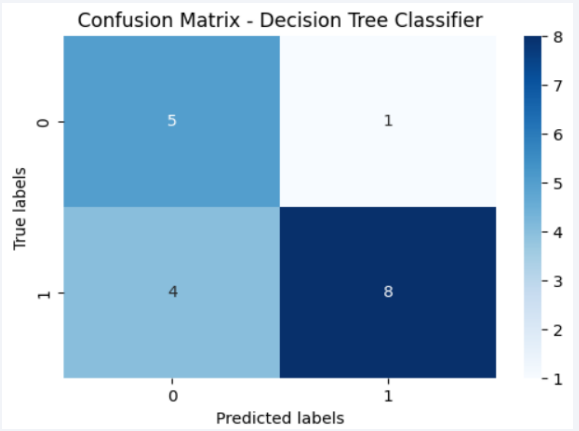
Appendix

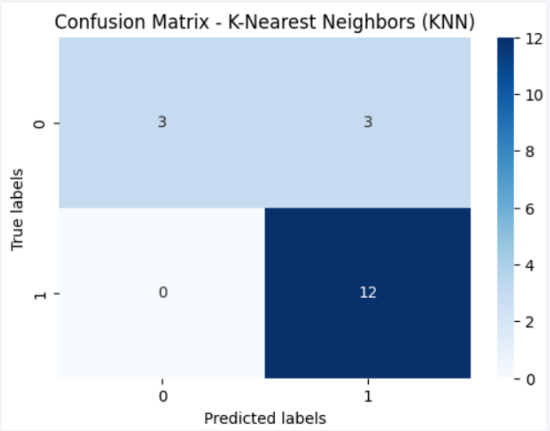
Charts & Visualizations

- Confusion matrices of all models









Appendix

Notebook References

1. Data Collection

- Notebook: SpaceX API Scraping + CSV Integration
- Description: Retrieved launch records using SpaceX REST API and integrated external CSV datasets.
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/1.jupyter-labs-spacex-data-collection-api.ipynb

2. Data Wrangling

- Notebook: Data Cleaning and Feature Engineering
- Description: Cleaned, merged, and transformed datasets into a structured format ready for EDA and ML.
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/3.labs-jupyter-spacex-Data%20wrangling.ipynb

Appendix

Notebook References (cont.)

3. Exploratory Data Analysis (EDA)

- Notebook:
- EDA with SQL: Queried SpaceX database for mission outcomes and payload insights.
- EDA with Visualization: Generated scatter plots, bar charts, and line charts to explore success patterns.
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/4.jupyter-labs-eda-sql-coursera_sqllite.ipynb
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/5.edadataviz.ipynb

4. Interactive Folium Map

- Notebook: Global Launch Sites Mapping
- Description: Created an interactive map marking launch sites, proximity analysis with folium.
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/6.lab_jupyter_launch_site_location.ipynb

Appendix

Notebook References (cont.)

5. Plotly Dash Application

- Code Block: SpaceX Launch Dashboard
- Description: Built an interactive dashboard to visualize launch success by site and payload range.
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/7.spacex-dash-app.py

6. Machine Learning Model Development

- Notebook: Falcon 9 Landing Prediction Models
- Description: Developed, tuned, and evaluated Logistic Regression, SVM, Decision Tree, and KNN models.
- GitHub Link: https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/blob/main/8.SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Appendix

GitHub Repository Link

- https://github.com/AdamkKureshi/IBM_DS_Cap_Stone/tree/main

Thank you!