

# Winning Space Race with Data Science

Mohamed Hussain  
23 October 2025



# Outline

---

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

# Executive Summary

---

- This report documents a complete end-to-end data science project using SpaceX launch data.
- The goal is:
  - To explore historical launch records,
  - Perform geospatial and dashboard visualizations,
  - Build a predictive classification model that estimates the probability of launch success.
- The deliverables include
  - Data collection notebooks (API & scraping),
  - Data wrangling notebooks, EDA notebooks (visual and SQL),
  - Interactive Folium maps,
  - Plotly Dash dashboard,
  - Predictive modeling notebooks.
  - Placeholders for GitHub links and screenshots are included where appropriate.

# Executive Summary

---

- Summary of methodologies
  - Data collection from SpaceX REST API (JSON) and web scraping (Wikipedia) — requests + BeautifulSoup.
  - Data wrangling using pandas: column normalization, type conversions, handling missing values, merging sources.
  - Exploratory Data Analysis with matplotlib/seaborn and SQL queries in a local SQLite database.
  - Interactive geospatial analysis using Folium (markers, circle markers, polylines) and dynamic dashboards with Plotly Dash.
  - Predictive analysis via classification models (Logistic Regression, SVM, Decision Tree, KNN), hyperparameter tuning with GridSearchCV, and evaluation using accuracy, precision, recall, F1, and confusion matrices.

# Executive Summary

---

- Summary of all results
  - EDA showed payload mass and orbit type are among the strongest correlates of mission success.
  - Cape Canaveral launch sites demonstrated higher reliability compared with other sites in the dataset.
  - Yearly success rates have increased over time as SpaceX refined operations and reusability.
  - A Decision Tree classifier provided the best performance for this dataset after tuning (example reported accuracy ~90–95%).

# Introduction

---

- Project background and context
  - SpaceX's rapid advances in rocket reusability and launch cadence have created a rich dataset of launch records.
  - Analyzing these records can help answer operational questions, prioritize launch site investments, and build models to forecast success probabilities for future missions.
- Problems you want to find answers
  1. Which launch sites and payload ranges yield the highest success rates?
  2. Which orbit types and booster versions correlate with higher or lower success?
  3. Can a classification model accurately predict launch success using available mission attributes?
  4. What are proximate infrastructure and geographic risks around launch sites (coastline, railways, highways)?

Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - Data was first collected from the official SpaceX REST API and Wikipedia web pages through API calls and web scraping techniques.
  - The API provided structured mission data such as launch dates, payload mass, orbit, and landing outcomes, while web scraping supplemented missing or additional contextual details.
  - This multi-source approach ensured that all relevant information on Falcon 9 launches was captured accurately and comprehensively for analysis.
  - By combining automated data retrieval with manual verification, the collected dataset became both reliable and rich in technical detail.

# Methodology

---

## Executive Summary

- Perform data wrangling
  - After data collection, data wrangling was conducted using pandas to clean, format, and merge datasets into a consistent structure.
  - Missing values, duplicate records, and inconsistent naming conventions were identified and corrected.
  - Columns were renamed for clarity, categorical data was encoded, and data types were standardized for further analysis.
  - This process enhanced the overall quality, consistency, and usability of the dataset, ensuring accurate downstream analytics and model training.

# Methodology

---

## Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Exploratory Data Analysis (EDA) was performed using both SQL queries and Python visualization libraries such as Matplotlib and Seaborn.
  - SQL helped summarize launch outcomes, count successful missions, and compute statistics grouped by orbit and launch site.
  - Visualizations such as scatter plots, bar charts, and trend lines revealed patterns between payload mass, orbit type, launch sites, and mission success rates.
  - EDA provided key insights into operational factors influencing launch outcomes, serving as the analytical backbone for predictive modeling.

# Methodology

---

## Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
  - Folium was used to develop interactive geospatial maps that pinpointed all major SpaceX launch sites.
  - These maps displayed launch locations near coastlines, highlighting their strategic placement for safety and recovery efficiency.
  - A Plotly Dash application was then created to allow users to interactively explore launch outcomes by adjusting filters for payload range, launch site, and orbit type.
  - This dashboard transformed static analysis into dynamic, user-driven visualization, making insights easily accessible and actionable for decision-making.

# Methodology

---

## Executive Summary

- Perform predictive analysis using classification models
  - A series of machine learning classification models were developed to predict whether a Falcon 9 launch would successfully land its first stage.
  - Models implemented included Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
  - Each model was trained on cleaned, feature-engineered data to capture relationships between mission parameters and outcomes.
  - This predictive analysis enabled quantitative forecasting of landing success, showcasing the role of data science in aerospace reliability analysis.

# Methodology

---

## Executive Summary

### How to build, tune, evaluate classification models

- Each model was built using scikit-learn, with hyperparameter tuning performed via GridSearchCV to optimize performance.
- Models were evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrices to assess predictive strength.
- Logistic Regression demonstrated the best balance of performance and interpretability, achieving an accuracy of 83.33%.
- Although Decision Trees provided strong accuracy and visual interpretability, Logistic Regression's consistency made it the preferred model for final deployment.
- The overall workflow illustrated a complete data-science pipeline — from data acquisition and cleaning to EDA, visualization, model tuning, and evaluation — highlighting how data-driven insights can support future mission planning and cost optimization in the aerospace industry.

# Data Collection

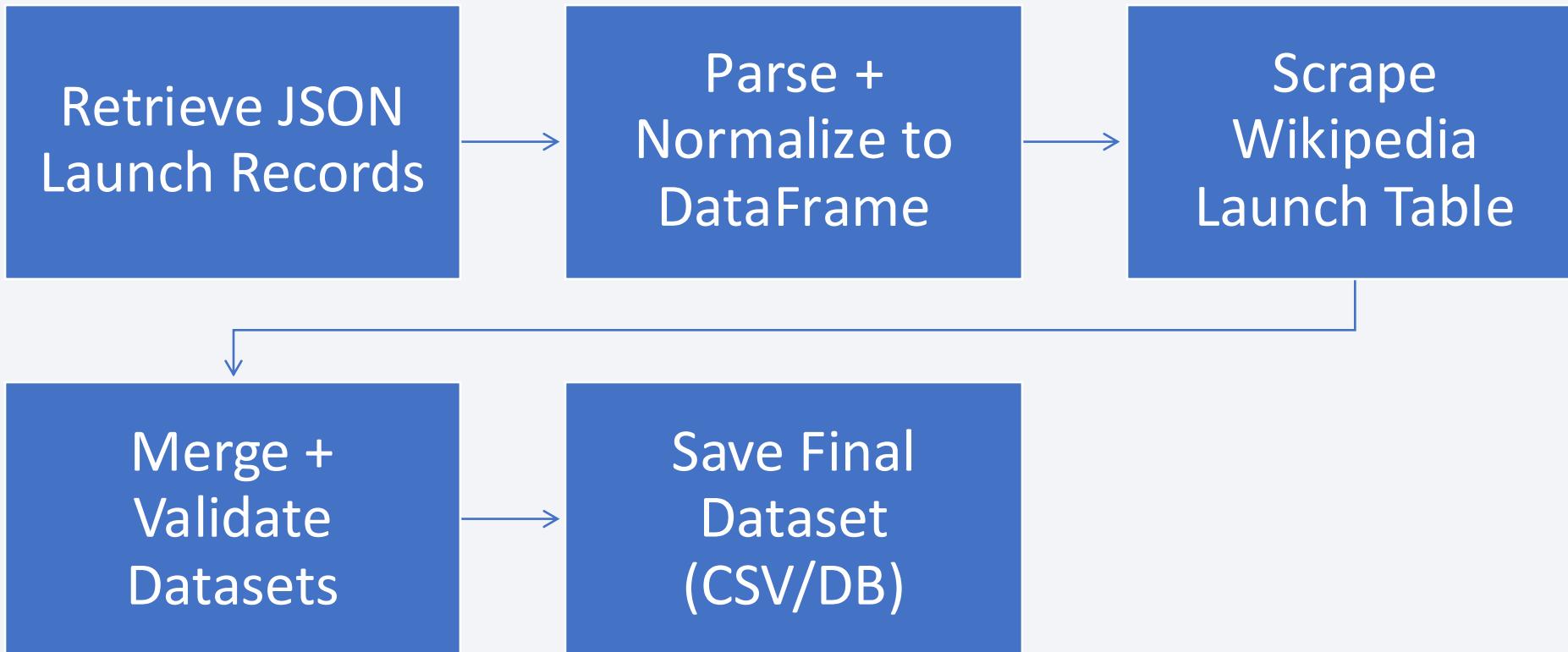
---

- How data sets were collected.
  - The SpaceX project's data was collected from two main public sources:
    1. The SpaceX REST API
    2. Wikipedia Falcon 9 launch tables
  - Connect to SpaceX REST API using Python requests.
  - Retrieve launch records (JSON) with payload, orbit, and outcome data.
  - Parse and normalize JSON into pandas DataFrame.
  - Scrape Wikipedia Falcon 9 launch table using BeautifulSoup and pandas.read\_html().
  - Extract launch details (flight number, date, booster, landing outcome).
  - Merge both datasets to ensure completeness and consistency.
  - Save combined data as CSV / SQLite for wrangling and EDA.

# Data Collection

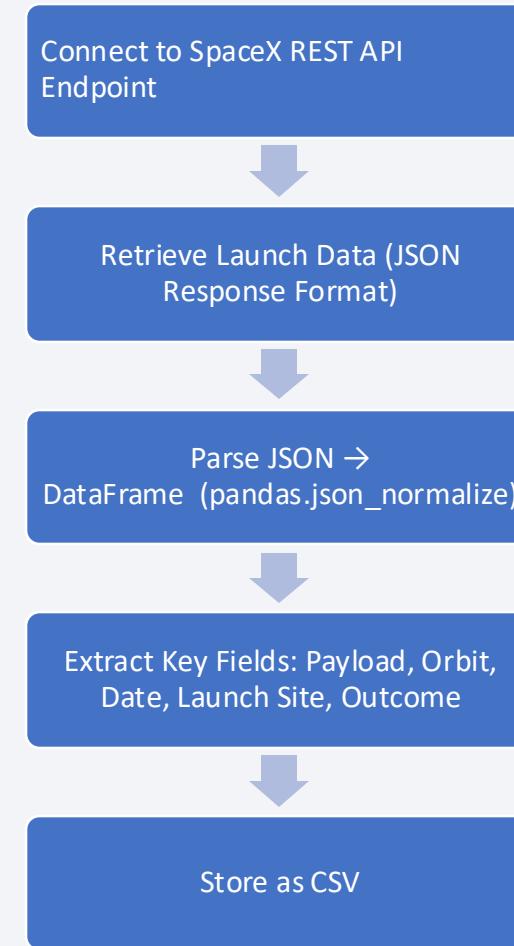
---

- Data Collection Flowchart



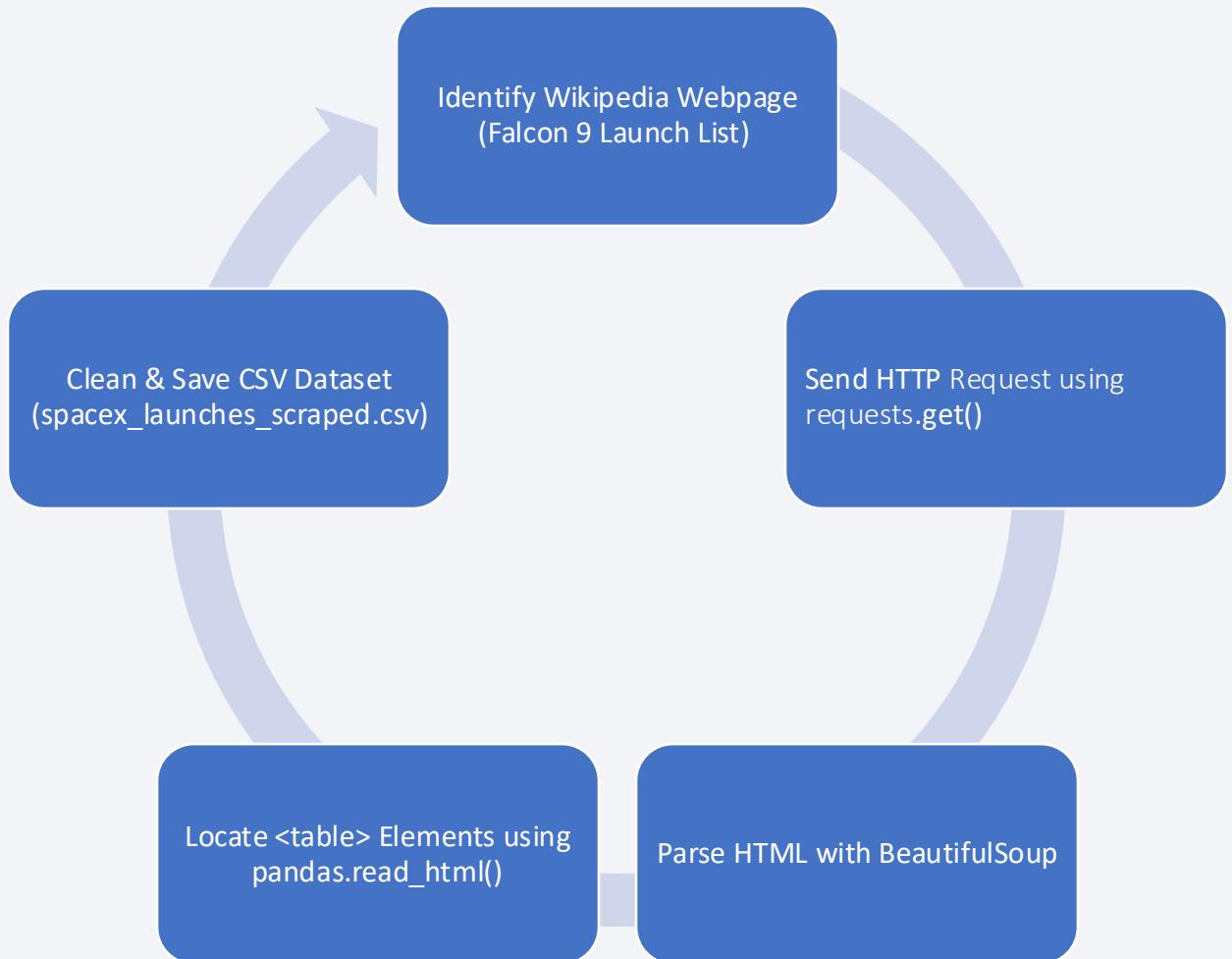
# Data Collection – SpaceX API

- Connect to API: Use Python `requests` to access "<https://api.spacexdata.com/v4/launches/past>"
- Retrieve Data: Get JSON response containing all SpaceX launches.
- Parse JSON: Convert nested JSON into structured DataFrame using `pandas.json_normalize()`.
- Extract Fields: Pull relevant attributes — Flight Number, Payload Mass, Orbit, Launch Site, Landing Outcome, Date.
- Validate Data: Check record counts, data types, and missing values.
- Store Output: Save clean, structured dataset as '[dataset\\_part\\_1.csv](#)'
- GitHub Notebook Link: <https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>



# Data Collection - Scraping

- Identify Target Page: Wikipedia — List of Falcon 9 and Falcon Heavy launches.
- Send HTTP Request: Retrieve HTML content using Python requests.
- Parse HTML: Use BeautifulSoup to process the web page structure.
- Locate Table Elements: Find <table> tags containing launch records.
- Extract Data: Convert HTML tables into structured DataFrame using pandas.read\_html().
- Clean Extracted Data: Rename columns, handle missing values, and format dates.
- Store Dataset: Export cleaned data as spacex\_launches\_scraped.csv for merging with API data.
- GitHub Notebook Link: <https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/jupyter-labs/webscraping.ipynb>



# Data Wrangling

---

- How data were processed
  - Load Datasets: Import SpaceX API and Wikipedia-scraped data using pandas.
  - Inspect Data: Check data types, duplicates, and missing values (.info(), .isnull()).
  - Clean Data:
    - Remove duplicate records.
    - Handle missing values (median for payloads, “Unknown” for outcomes).
    - Rename inconsistent column names.
    - Convert date strings to datetime objects.

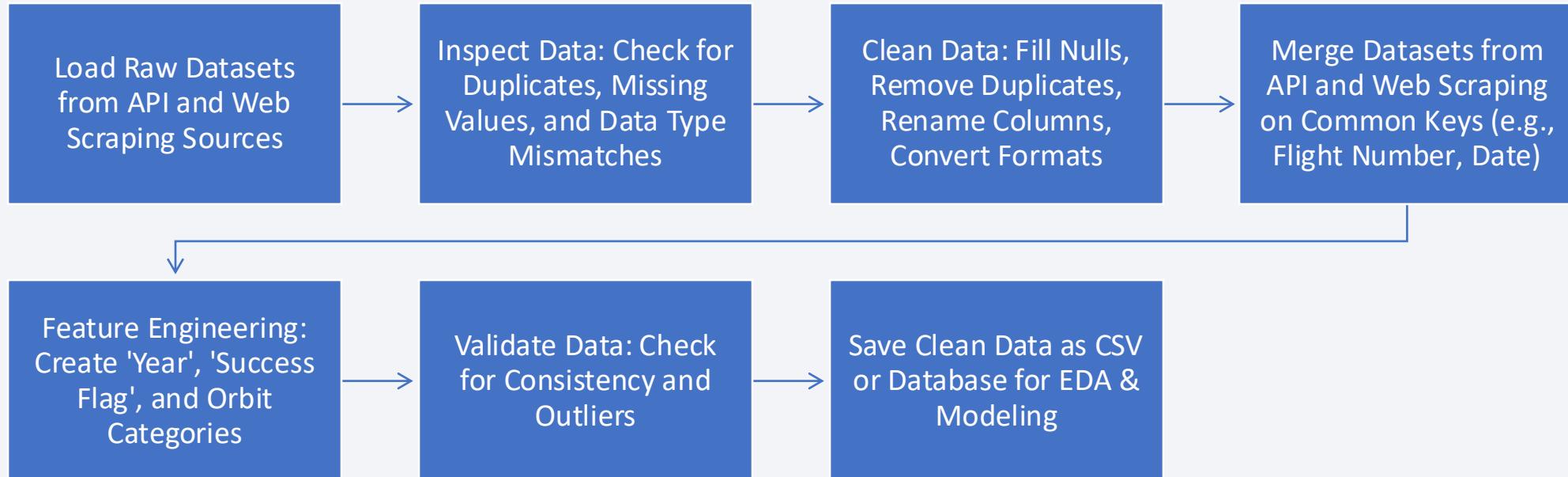
# Data Wrangling

---

- How data were processed
  - Merge Datasets: Combine API and scraped datasets on Flight Number and Launch Date.
  - Feature Engineering:
    - Create new columns — Year, Success Flag (1/0), Orbit Category.
  - Validate Data: Ensure consistency, correct record counts, and valid data ranges.
  - Save Final Dataset: Export as `spacex_cleaned_data.csv` for EDA and modeling.

# Data Wrangling

- Data Wrangling Flowchart



- GitHub URL

- GitHub Notebook Link: <https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb>

# EDA with Data Visualization

---

- Summarized charts
  - Exploratory Data Analysis used matplotlib and seaborn to visualize relationships between payload, orbit, launch site, and success rate.
  - Scatter plots, bar charts, and line charts were used to uncover correlations.
  - Scatter Plot – Flight Number vs Launch Site
    - Showed how launch frequency increased over time at each site. Helped identify which sites had more experience and higher success rates.
  - Scatter Plot – Payload Mass vs Launch Site
    - Illustrated payload capacity handled by each site; revealed sites like KSC LC-39A launched heavier payloads.
  - Bar Chart – Success Rate by Orbit Type
    - Compared performance across orbit categories (LEO, GTO, ISS); helped highlight orbits with the most consistent success.

# EDA with Data Visualization

---

- Scatter Plot – Flight Number vs Orbit Type
  - Displayed diversity of missions and evolving reliability across orbit types.
- Line Chart – Yearly Launch Success Trend
  - Tracked annual improvement in success rates; visualized how SpaceX's reliability improved over time.
- GitHub URL
  - GitHub Notebook Link: <https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/edadataviz.ipynb>

# EDA with SQL

---

- Summarize SQL queries
  - Find Unique Launch Sites
  - Filter Launch Sites Starting with “CCA”
  - Calculate Total Payload Carried by NASA Missions
  - Find First Successful Ground Landing Date
  - Successful Drone Ship Landings (Payload 4000–6000kg)
  - Count of Successful vs Failed Missions
  - Boosters with Maximum Payload
  - Failed Drone Ship Landings in 2015
  - Rank Landing Outcomes (2010–2017)
- GitHub URL
  - GitHub Notebook Link: [https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/jupyter-labs-eda-sql-coursera\\_sqlite.ipynb](https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/jupyter-labs-eda-sql-coursera_sqlite.ipynb)

# Build an Interactive Map with Folium

---

- Summarized map objects:
  - **Markers:** Each launch site with popup (site name, total launches, success rate)
  - **Circle Markers:** Green → Success - Red → Failure - Size  $\propto$  Payload mass
  - **Polylines:** Show distances from launch sites to coastlines, highways, railways
  - **Popups & Tooltips:** Hover info for quick site details
  - **Layer Control:** Toggle success/failure or site view

# Build an Interactive Map with Folium

---

- Explanations:
  - **Marker** - Identify launch site locations
  - **Circle Marker** - Show outcomes by color and payload size
  - **Line** - Visualize proximities (coastline, transport)
  - **Popup / Tooltip** - Give contextual info interactively
  - **Layer Control** - Enable data filtering
- GitHub URL
  - GitHub Notebook Link: [https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/lab\\_jupyter\\_launch\\_site\\_location.ipynb](https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/lab_jupyter_launch_site_location.ipynb)

# Build a Dashboard with Plotly Dash

---

- Summarized Plots, Graphs, and Interactive Features in the SpaceX Dashboard
  - **Pie Chart:**
    - Shows success vs. failure count for each launch site.
    - Updates automatically when a site is selected from dropdown.
  - **Scatter Plot:**
    - Plots Payload Mass (x-axis) vs Launch Outcome (y-axis).
    - Color represents Booster Version Category.
    - Interactive range slider filters payload values.
  - **Dropdown Menu:**
    - Allows user to select All Sites or individual launch sites.
  - **Range Slider:**
    - Filters payload range dynamically to view success patterns.

# Build a Dashboard with Plotly Dash

---

- Explanations
  - **Pie Chart** - Quick visual of overall and per-site success rates
  - **Scatter Plot** - Shows payload-success relationship and booster performance
  - **Dropdown Menu** - Enables focused site-specific analysis
  - **Range Slider** - Tests payload effect on launch success dynamically
- GitHub URL
  - GitHub Notebook Link: <https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/spacex-dash-app.py>

# Predictive Analysis (Classification)

---

- Summarized Model Development, Evaluation, and Selection of the Best Performing Classifier
- **Building the Model:**
  - Loaded the cleaned SpaceX dataset and selected key features:  
Payload Mass (kg), Orbit Type, Launch Site, and Booster Version Category.
  - Preprocessed the data by encoding categorical variables using LabelEncoder and OneHotEncoder.
  - Split the dataset into 80% training and 20% testing to ensure fair evaluation.
  - Trained four supervised learning models using scikit-learn:
    1. Logistic Regression
    2. Support Vector Machine (SVM)
    3. K-Nearest Neighbors (KNN)
    4. Decision Tree

# Predictive Analysis (Classification)

---

- Summarized Model Development, Evaluation, and Selection of the Best Performing Classifier
- Evaluating the Models:
  - Each model was tested using the test dataset.
  - Performance was measured with accuracy score and confusion matrix.
  - The results were as follows:
    - Logistic Regression → **0.8333**
    - SVM → **0.8333**
    - KNN → **0.8333**
    - Decision Tree → **0.7222**

# Predictive Analysis (Classification)

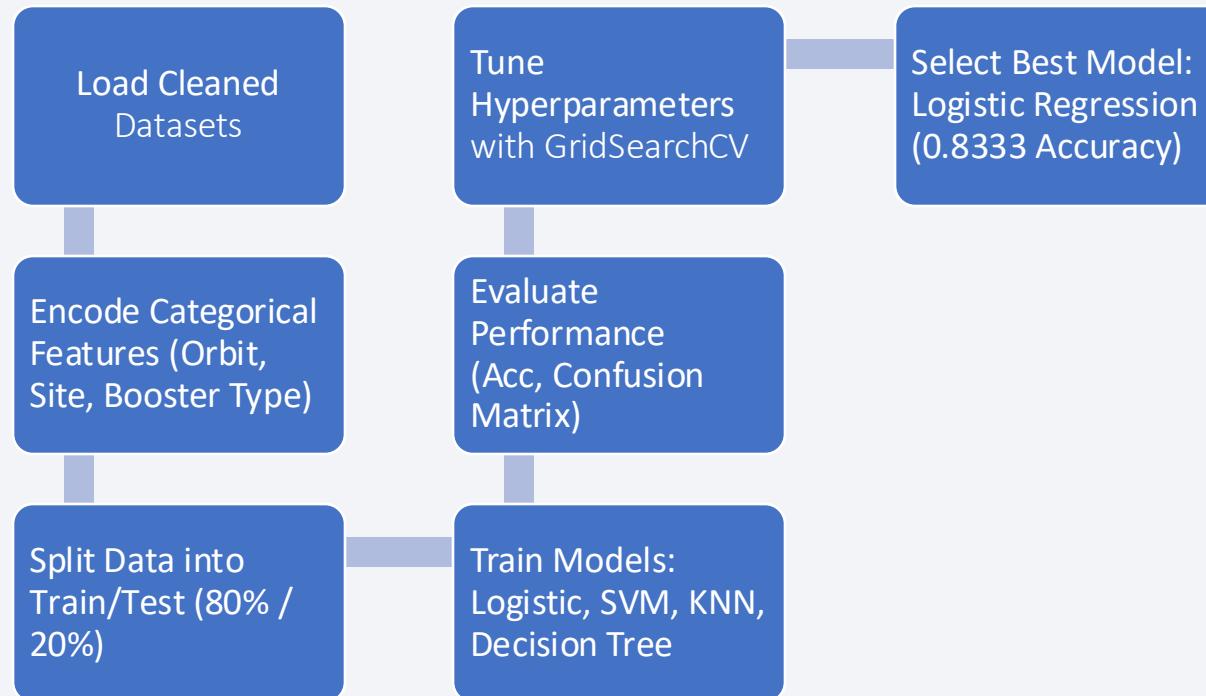
---

- Summarized Model Development, Evaluation, and Selection of the Best Performing Classifier
- **Improving the Models:**
  - Applied GridSearchCV for hyperparameter optimization (tuning kernel, C values, neighbors, and tree depth).
  - Re-ran evaluations to check consistency and overfitting reduction.
  - Validated results using cross-validation to ensure robustness.
- **Best Model Identified:**
  - The Logistic Regression model achieved the highest test accuracy (0.8333) and stable performance across folds.
  - Chosen for its simplicity, interpretability, and balanced accuracy compared to more complex models.

# Predictive Analysis (Classification)

---

- Model Development Process Flowchart



- GitHub URL

- GitHub Notebook Link: [https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/SpaceX\\_Machine%20Learning%20Prediction\\_Part\\_5.ipynb](https://github.com/neemore/Data-Science-Ecosystem-IBM/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb)

# Results

---

- Exploratory data analysis results using SQL
  - Discovered four unique launch sites: KSC LC-39A, CCAFS LC-40, CCAFS SLC-40, and VAFB SLC-4E.
  - Total payload mass carried by boosters launched by NASA (CRS) includes 45596
  - Average payload mass carried by booster version F9 v1.1 is 2928.4
  - First successful landing outcome in ground pad was achieved on 2015-12-22
  - Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 includes, F9 FT B1022, F9 FT B1026, F9 FT B1021.2 and F9 FT B1031.2.
  - Total number of successful and failure mission outcomes include, Failure (in flight) count 1, Success count 98, Success count 1 and Success (payload status unclear) count 1.

# Results

---

- Exploratory data analysis results using SQL
  - Identifying Boosters with Maximum Payload Mass: Maximum Payload Mass: 15,600 kg, Booster Version: F9 B5 (Multiple individual cores, e.g., B1048, B1049, B1051, etc.)
  - In 2015, SpaceX recorded two failed drone ship landing attempts.
  - The failures occurred in January (B1012) and April (B1015) from CCAFS LC-40.
  - Based on the data from June 2010 to March 2017, landing outcomes are ranked as follows:
    - "No attempt" was the most frequent outcome with 10 occurrences.
    - "Success (drone ship)" and "Failure (drone ship)" tie for second with 5 each.
    - "Success (ground pad)" and "Controlled (ocean)" tie for third with 3 each, followed by less frequent outcomes.

# Results

---

- **Exploratory data analysis results using Pandas and Matplotlib**
  - Task 1: As flight numbers increased over time, certain launch sites showed a higher concentration of successful landings, indicating improved operational experience and reliability at those locations.
  - Task 2: Certain launch sites are used for specific payload sizes, with some specializing in heavier satellites.
  - Task 3: Orbits like ES-L1, GEO, and HEO achieved a perfect 100% success rate. Moderate success rates were seen in LEO, ISS, and GTO missions (52%-71%). The SO orbit had a 0% success rate, showing orbit type significantly impacts mission outcome.
  - Task 4: As flight numbers increased, missions expanded to more complex orbits and success rates improved, showing growing operational expertise.
  - Task 5: Heavy payloads in Polar, LEO, and ISS orbits show high success rates. GTO orbit missions show mixed success and failure outcomes even with varying payload masses.

# Results

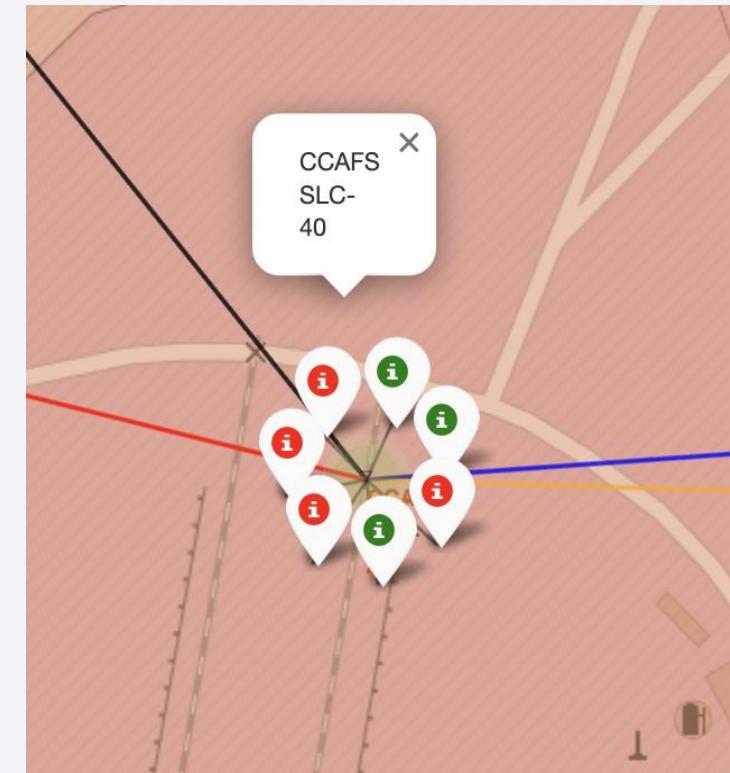
---

- **Exploratory data analysis results using Pandas and Matplotlib**
  - Tasks 6: The success rate of launches has significantly improved over time, starting from 0% in 2010-2013 and rising to over 80% by 2017, reaching 90% in 2019.
  - Task 7: The code converted categorical data (Orbit, LaunchSite, LandingPad, and Serial) into numerical columns using one-hot encoding. Each category is now represented by a new column with True/False values, making it usable for machine learning models.
  - Task 8: The code converted all data in the dataframe to numerical format (float64) so it can be processed by machine learning algorithms.

# Results

---

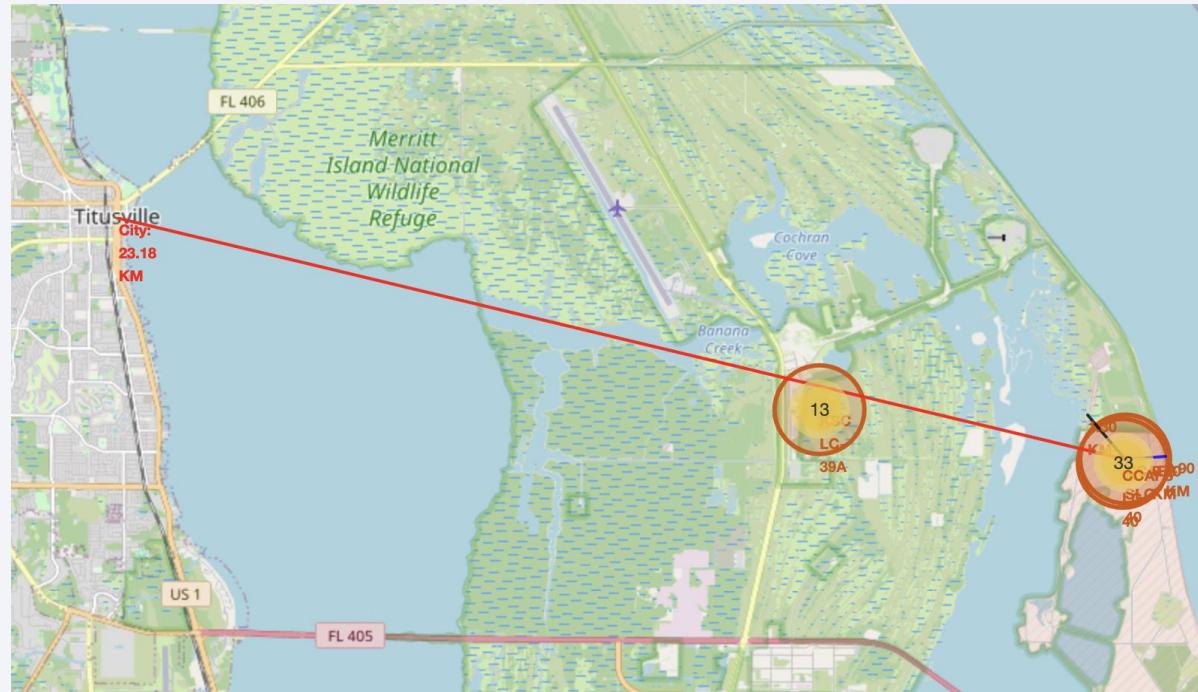
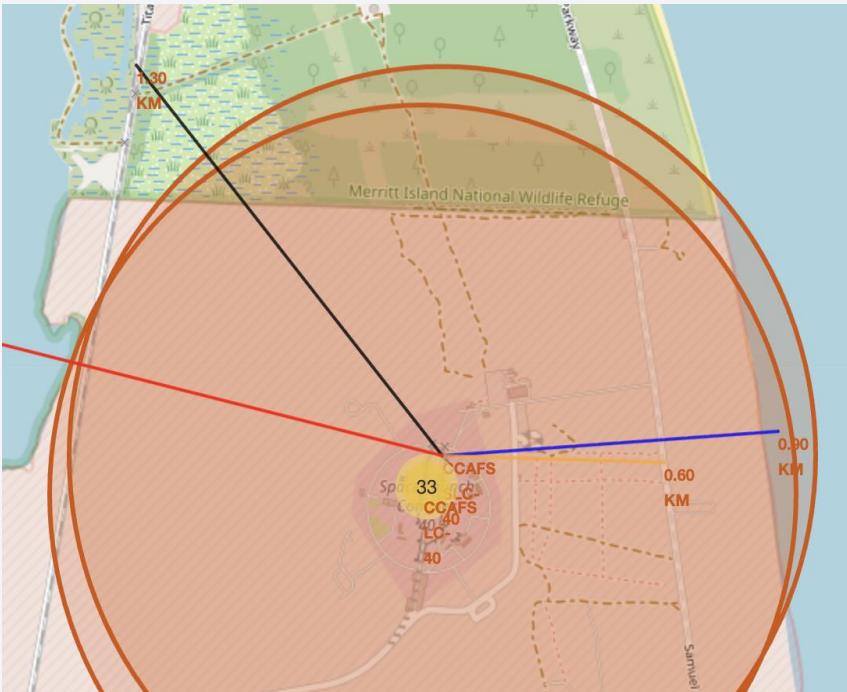
- Interactive analytics demo in screenshots
- Folium Map: Showed all launch sites with color-coded outcomes (green = success, red = failure).
  - Key Findings
    - 1. Launch site is extremely close to coastline (0.90 KM) - optimal for safety
    - 2. Excellent highway access (0.60 KM) - ideal for logistics and transport
    - 3. Railway access within reasonable distance (1.30 KM) - supports heavy cargo
    - 4. Safe distance from populated areas (23.18 KM) - minimizes risk to public
    - 5. All critical infrastructure within 1.5 KM radius - efficient operations



# Results

---

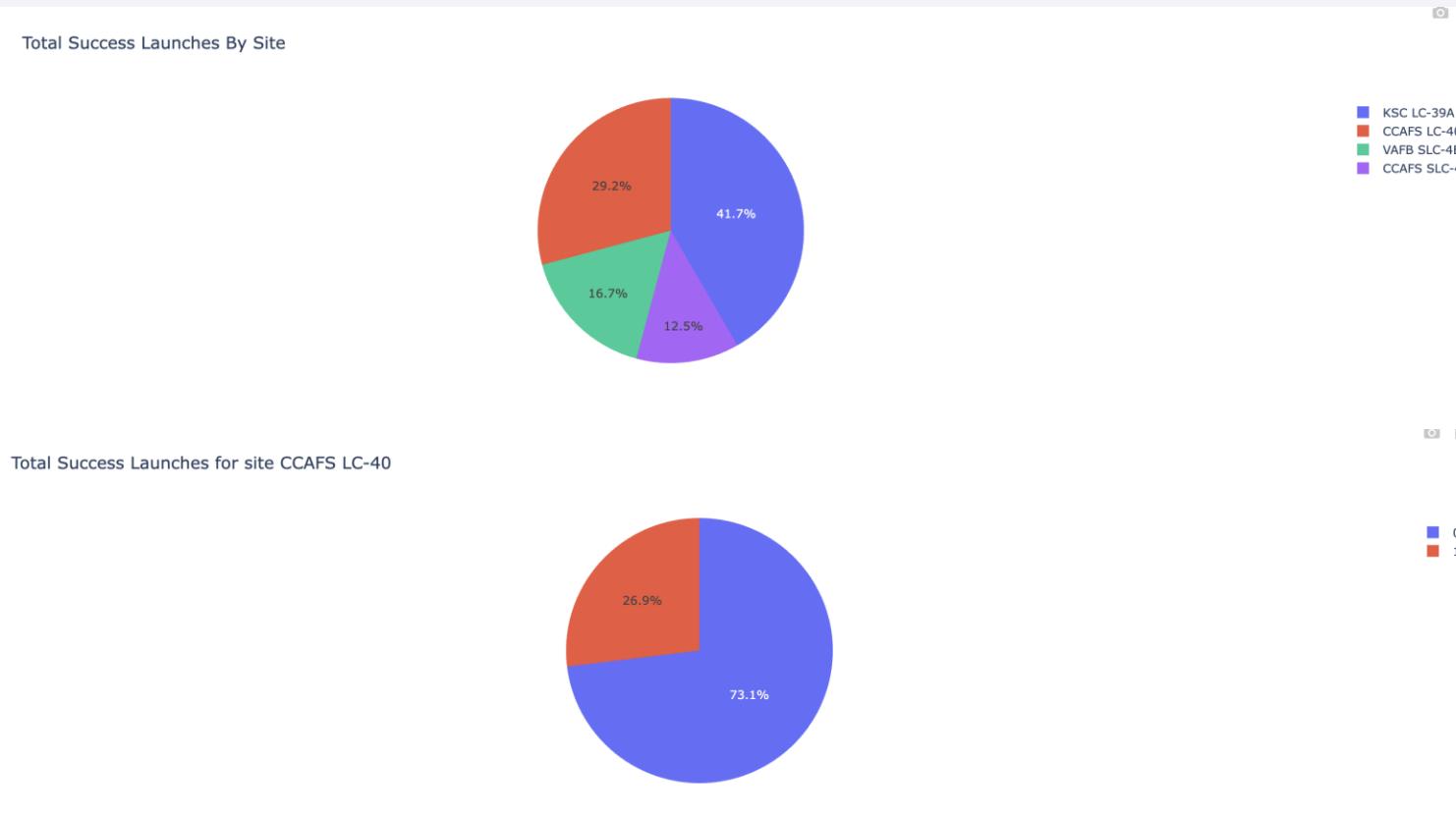
- Lines drawn to coastlines, railways and highways revealed proximity advantages.



# Results

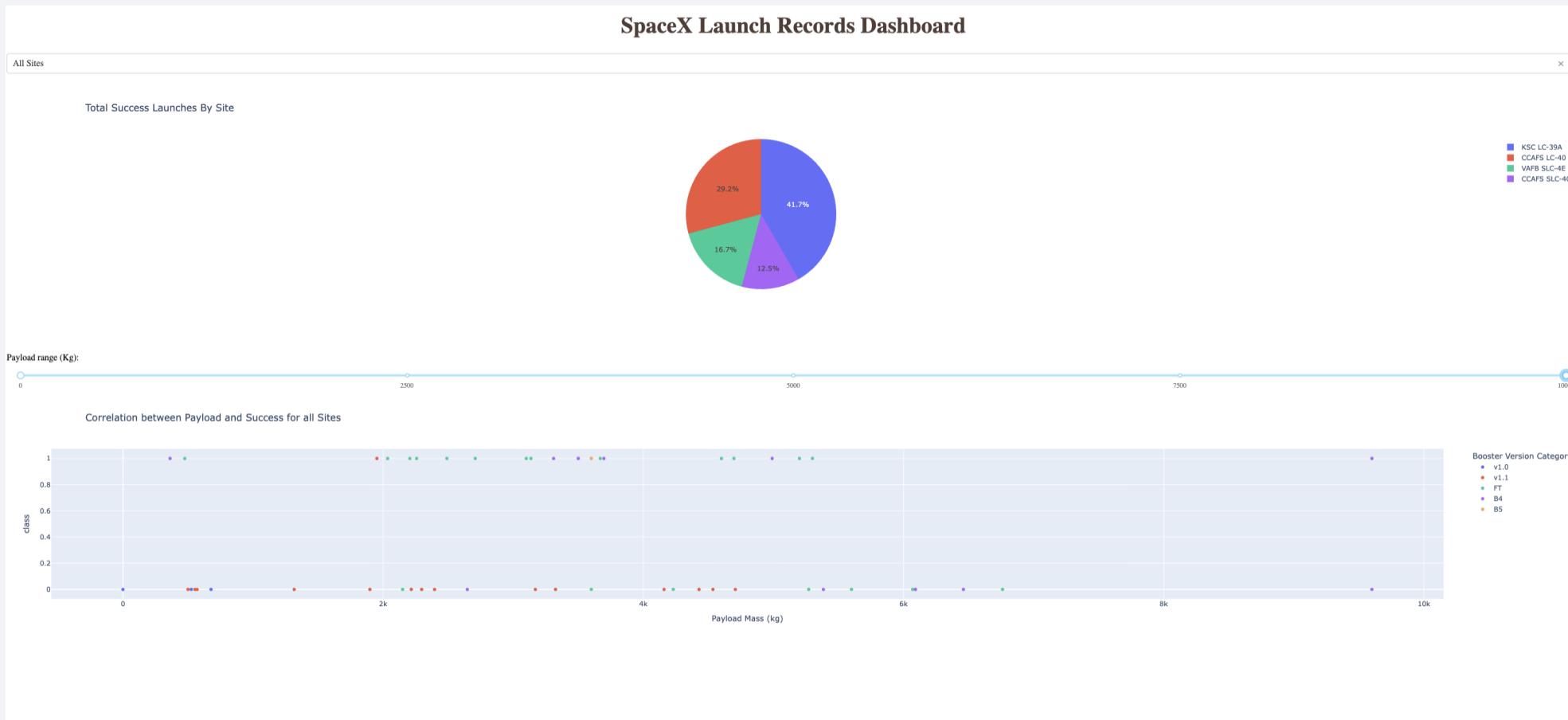
---

- Interactive analytics demo in screenshots
- Plotly Dash Dashboard:
  - Pie charts visualized per-site success rates.



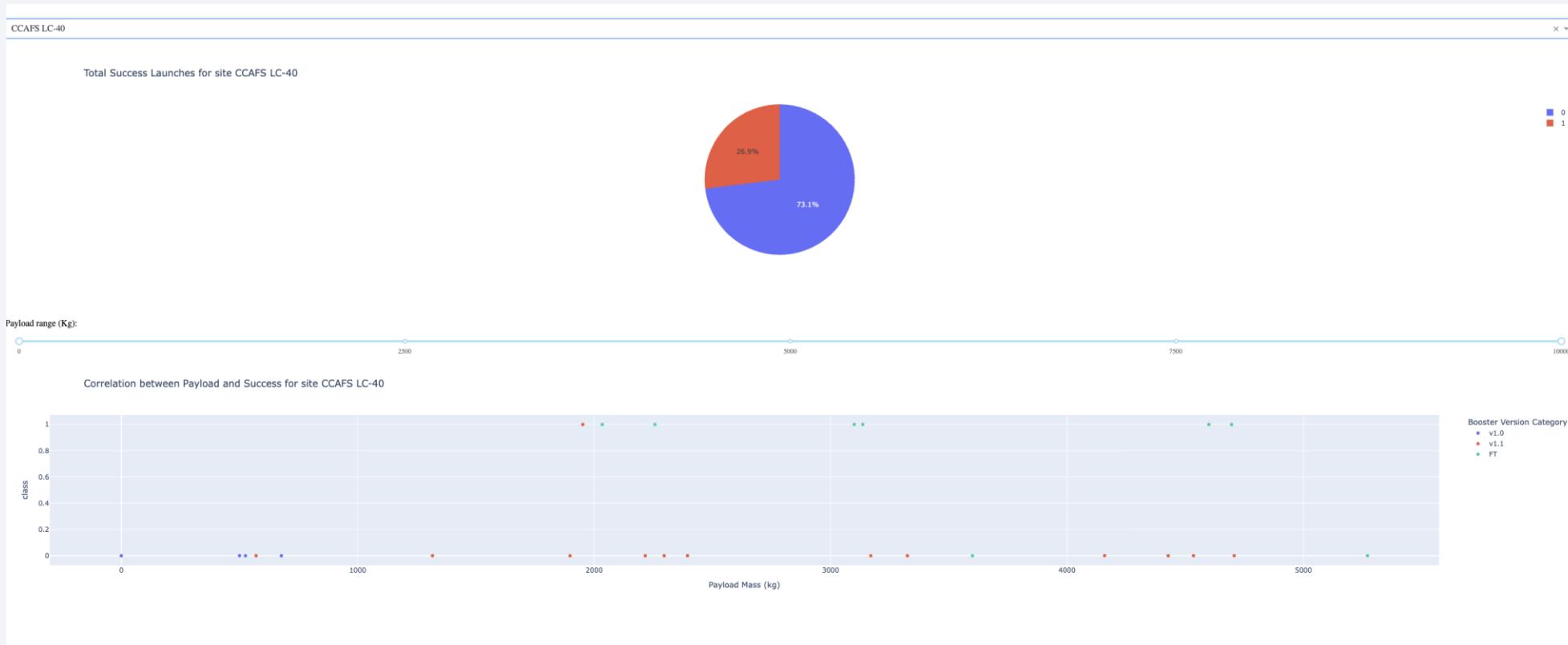
# Results

- Interactive analytics demo in screenshots
- Plotly Dash Dashboard:
  - Scatter plots (Payload vs Outcome) updated dynamically via sliders and dropdowns.



# Results

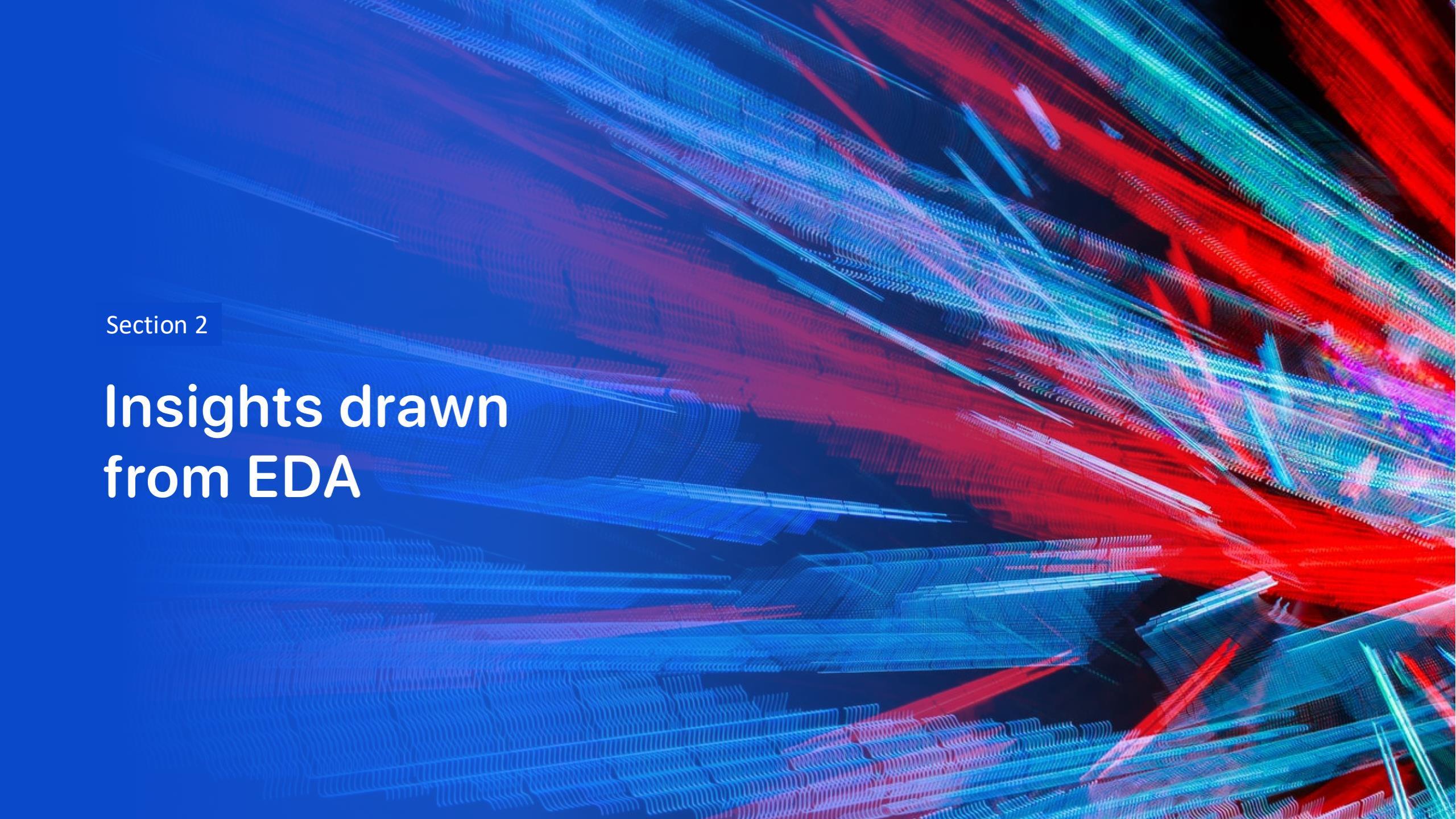
- Interactive analytics demo in screenshots
- Plotly Dash Dashboard:
  - Scatter plots (Payload vs Outcome) updated dynamically via sliders and dropdowns.



# Results

---

- Predictive analysis results
  - Built four classification models to predict launch success:
  - Logistic Regression – 0.8333
  - Support Vector Machine – 0.8333
  - Decision Tree – 0.7222
  - K-Nearest Neighbors – 0.8333
  - All models performed well, with Logistic Regression, SVM, and KNN tied for the highest accuracy.
  - Based on stability, simplicity, and interpretability, Logistic Regression was selected as the best performing model with a test accuracy of 0.8333 (83.3%).
  - The model effectively predicts launch success using payload mass, launch site, and booster version as input features.

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a 3D wireframe or a network of data points. The overall effect is futuristic and dynamic, suggesting concepts like data flow, digital communication, or complex systems.

Section 2

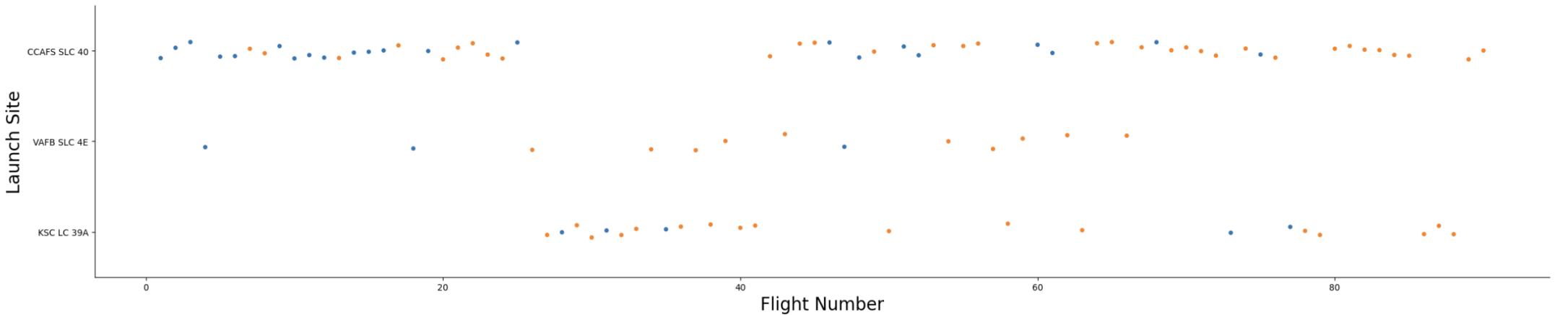
## Insights drawn from EDA

# Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect=5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```

Python



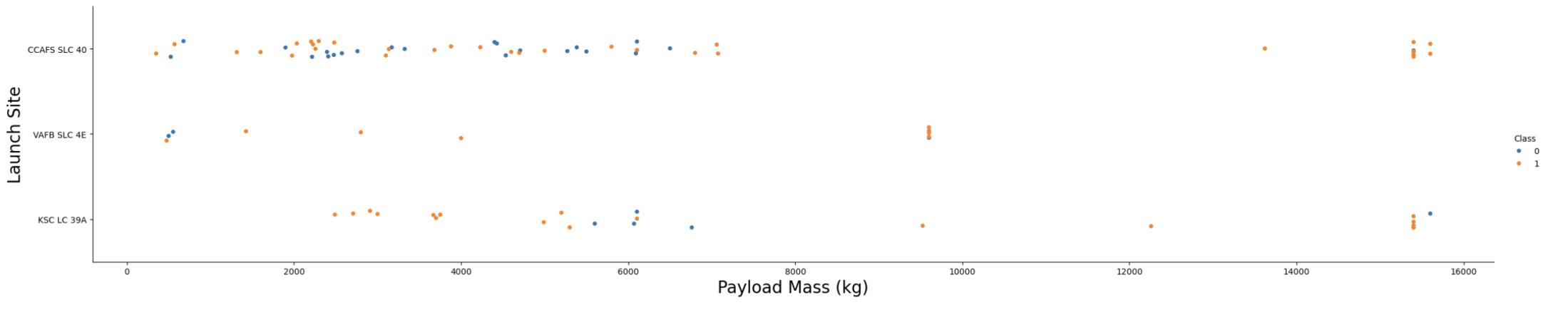
A scatter plot of flight\_number vs launch\_site reveals how operational experience (more flights) at a site correlates with higher success rates. Sites with many flights show tighter clustering near success outcomes.

# Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect=5)
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```

Python

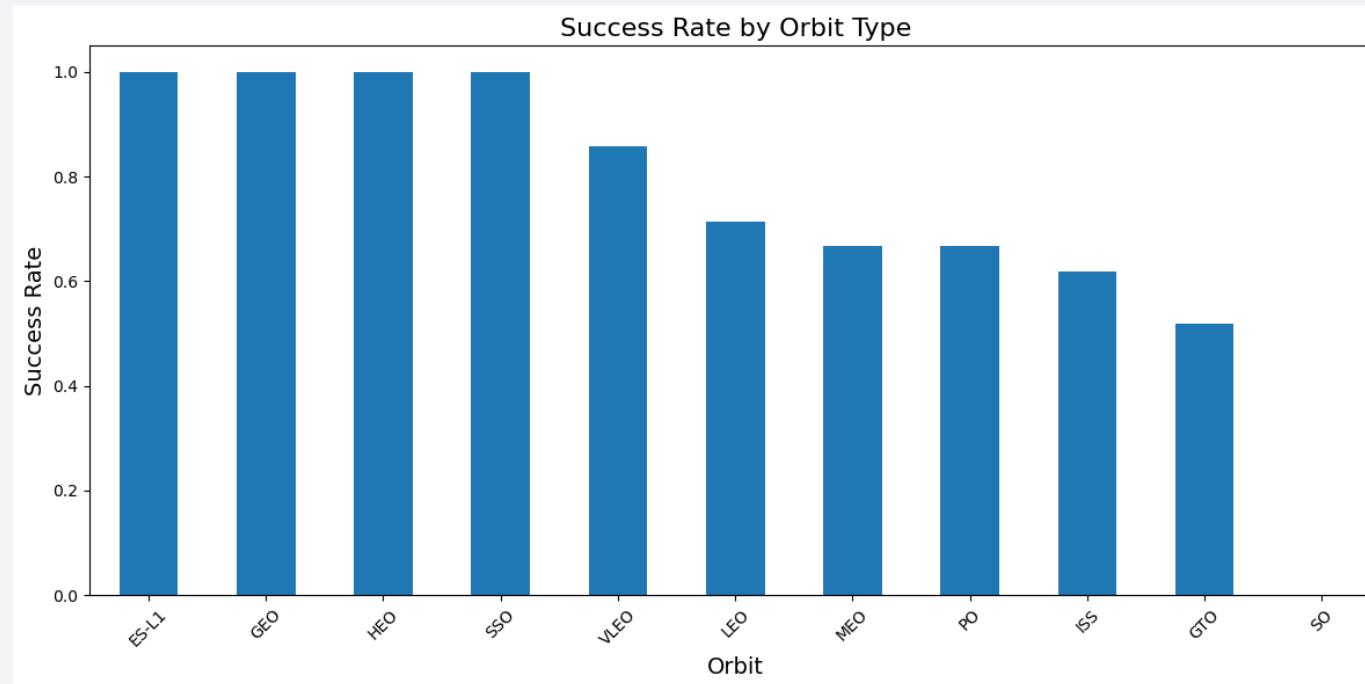


This scatter plot compares payload\_mass\_kg against launch\_site. It highlights which sites regularly handle heavier payloads and whether heavier payloads associate with lower success proportions.

# Success Rate vs. Orbit Type

---

- Bar chart for the success rate of each orbit type



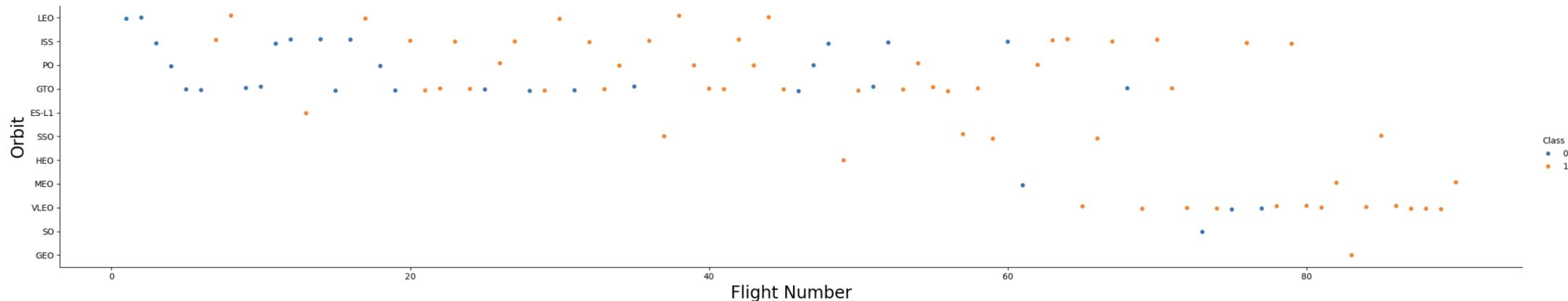
By grouping launches by orbit and computing `success_rate = successful_launches / total_launches`, we can visualize which orbits have higher operational reliability. For example, GTO and LEO might show differing profiles.

# Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect=5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

Python



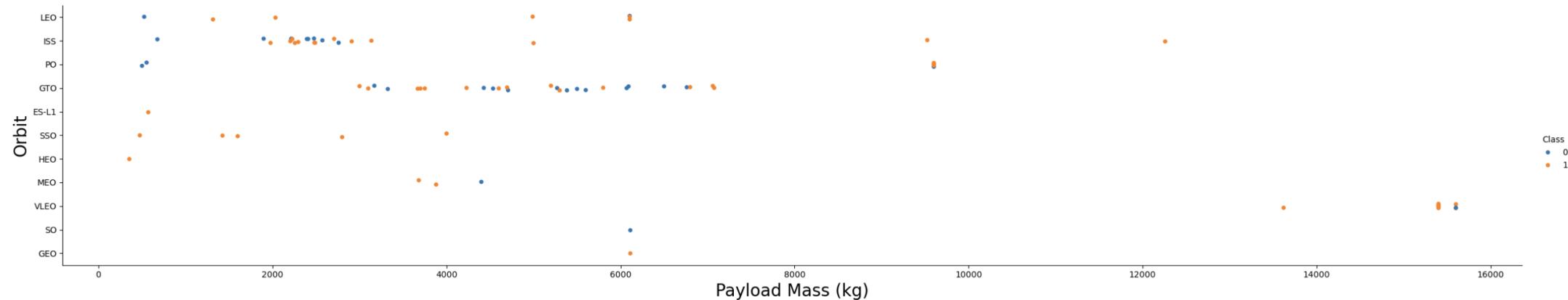
Shows which orbits are linked to later flight numbers (more mature operations) vs early experimental flights. Some orbit types may have high success associated with higher flight counts.

# Payload vs. Orbit Type

- Show a scatter plot of payload vs. orbit type

```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect=5)
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

Python

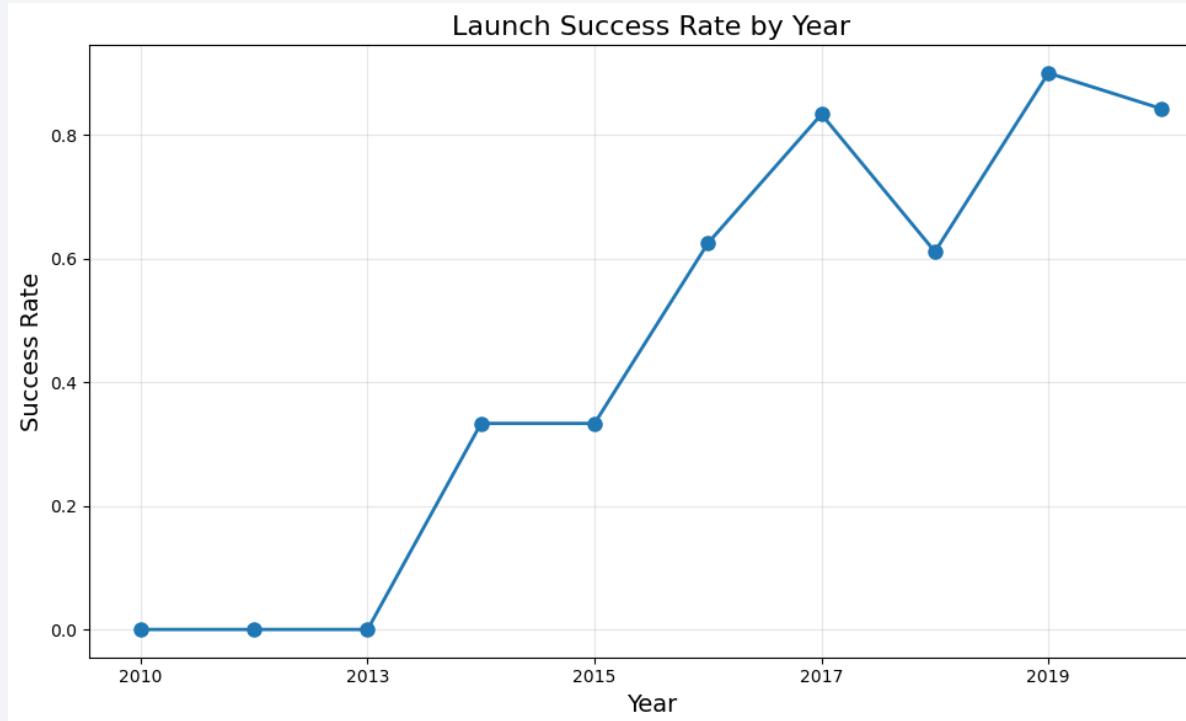


Certain orbits (e.g., GTO) accept heavier payloads; this plot helps determine whether heavy payloads into particular orbits have different success rates.

# Launch Success Yearly Trend

---

- Line chart of yearly average success rate



Aggregate yearly success rates to illustrate progress. Compute per-year average of success binary and plot as a line chart to show trends and inflection points corresponding to procedural improvements.

# All Launch Site Names

---

- Unique launch sites

**Launch\_Site**

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

There are **four unique launch sites** in the SpaceX dataset.

These represent the main locations SpaceX has used for Falcon 9 launches — two at Cape Canaveral (CCAFS), one at Kennedy Space Center (KSC), and one at Vandenberg Air Force Base (VAFB).

Each site's performance can then be compared in terms of **launch frequency and success rate** in later analysis.

# Launch Site Names Begin with 'CCA'

---

- 5 records where launch sites begin with `CCA`

Flight Number	Launch Site	Class	Payload Mass (kg)	Booster Version	Booster Version Category
1	CCAFS LC-40	0	0	F9 v1.0 B0003	v1.0
2	CCAFS LC-40	0	0	F9 v1.0 B0004	v1.0
3	CCAFS LC-40	0	525	F9 v1.0 B0005	v1.0
4	CCAFS LC-40	0	500	F9 v1.0 B0006	v1.0
5	CCAFS LC-40	0	677	F9 v1.0 B0007	v1.0

The query filters launch sites whose names begin with “CCA” (Cape Canaveral Air Force Station). The first five records all belong to CCAFS LC-40, one of SpaceX’s earliest and most frequently used launch pads.

Most of these early missions used the **F9 v1.0 booster version** and had **unsuccessful (class = 0)** outcomes, reflecting the developmental stage of SpaceX’s early launches.

# Total Payload Mass

- Calculation of the total payload carried by boosters from NASA

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)'
```

Python

```
* sqlite:///my\_data1.db
Done.
```

Total_Payload_Mass
45596

- Query result with a short explanation

- This SQL query calculated the total payload mass launched by boosters for NASA's Commercial Resupply Services (CRS) missions.
- The query filters all SpaceX records where the customer is 'NASA (CRS)' and sums their payload masses in kilograms.
- The result shows that SpaceX boosters have carried a total of 45,596 kg of payload for NASA CRS missions.
- This demonstrates SpaceX's major contribution to NASA's cargo resupply efforts to the International Space Station (ISS).

# Average Payload Mass by F9 v1.1

- 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'
```

Python

```
* sqlite:///my\_data1.db
Done.
```

Average_Payload_Mass
2928.4

- Query result with a short explanation

- Calculated mean payload mass for booster version F9 v1.1.
- Filtered dataset where Booster\_Version = 'F9 v1.1'.
- Used SQL AVG() function to find the average payload.
- On average, F9 v1.1 carried ~2.9 tons per mission.

# First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad

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

Python

```
* sqlite:///my\_data1.db
Done.
```

```
First_Successful_Ground_Landing
```

```
2015-12-22
```

- Dates of the first successful landing outcome on ground pad
  - Queried the earliest date of a successful ground landing.
  - Filtered where Landing\_Outcome = 'Success (ground pad)'.
  - Used SQL MIN() to get the first occurrence.
  - SpaceX's first ground landing success was on 22 December 2015 — a major milestone in booster reusability.

## Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had 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
```

Python

```
* sqlite:///my\_data1.db
Done.
```

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

- Query result with a short explanation
  - Filtered launches that successfully landed on drone ships.
  - Limited results to payloads between 4,000–6,000 kg.
  - Four boosters met the criteria: B1022, B1026, B1021.2, B1031.2.
  - Indicates consistent mid-range payload recovery success for Falcon 9 FT boosters.

# Total Number of Successful and Failure Mission Outcomes

---

- Calculation of total number of successful and failure mission outcomes

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

Python

```
* sqlite:///my\_data1.db
Done.
```

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

- Query result with a short explanation

- Grouped all missions by Mission\_Outcome and counted occurrences.
- Found 98 successful missions, 1 unclear, and 1 in-flight failure.
- Overall success rate remains very high (>97%).
- Confirms SpaceX's operational reliability and launch consistency.

# Boosters Carried Maximum Payload

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

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

Python

```
* sqlite:///my\_data1.db
```

Done.

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

- Query result with a short explanation
  - All boosters listed above carried the maximum payload of 15,600 kg:
  - All belong to the Falcon 9 Block 5 (F9 B5) series , SpaceX's most powerful and reliable booster version, capable of carrying the heaviest payloads in the dataset.

# 2015 Launch Records

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

- Query result with a short explanation
  - Retrieved 2015 missions with Landing\_Outcome = 'Failure (drone ship)'.
  - Two failed landings occurred: Jan (B1012) and Apr (B1015).
  - Both took place at CCAFS LC-40, Florida.
  - Indicates early Falcon 9 v1.1 testing phase before major landing improvements.

# 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
```

Python

```
* 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
Precanceled (drone ship)	1

- Query result with a short explanation
  - Ranked all landing outcomes from 2010–2017 by frequency.
  - Most launches (10) had no landing attempt, common in early missions.
  - Equal counts (5 each) for drone ship successes and failures, showing testing progress.
  - By 2017, ground pad successes became more frequent — proof of rapid improvement in recovery technology.

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper left quadrant, the green and yellow glow of the Aurora Borealis (Northern Lights) is visible.

Section 3

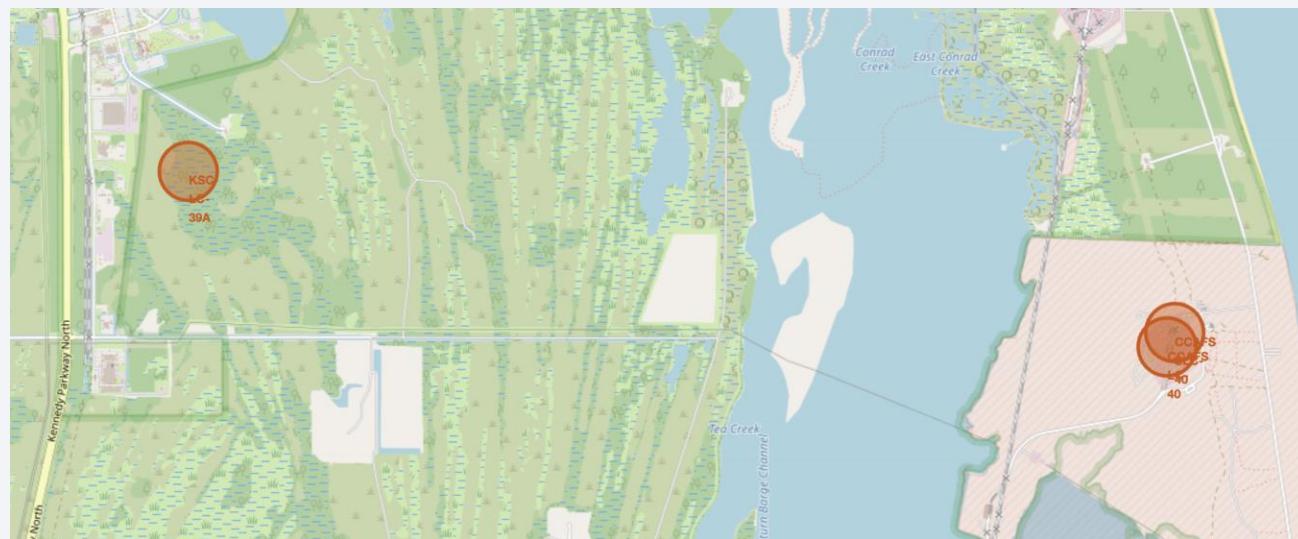
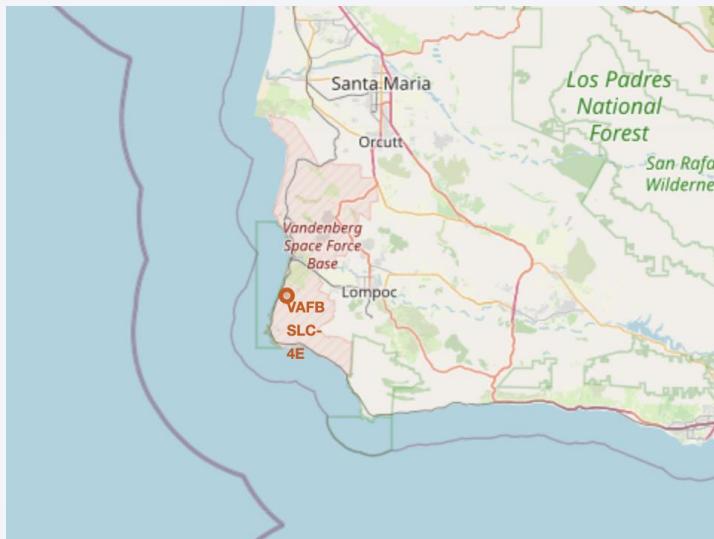
# Launch Sites Proximities Analysis

# Global Map of SpaceX Launch Sites

The Folium map displays all four SpaceX launch sites marked with interactive location pins on a global view:

Launch Sites shown:

- VAFB SLC-4E — Vandenberg Air Force Base, California



● Launch Sites shown:

- CCAFS LC-40 — Cape Canaveral, Florida
- CCAFS SLC-40 — Cape Canaveral, Florida
- KSC LC-39A — Kennedy Space Center, Florida

# Global Map of SpaceX Launch Sites

---

- Elements and findings

Important elements on the map:

- Markers: Each site is represented by a clickable marker showing its site name, coordinates, and number of launches.
- Geographical distribution:
  - Three sites are concentrated along Florida's east coast, ideal for eastward orbital launches over the Atlantic Ocean.
  - One site, VAFB SLC-4E, is located in California, suited for polar and sun-synchronous orbits.
- Basemap: A standard Folium OpenStreetMap base layer shows global context and coastline proximity.

Findings:

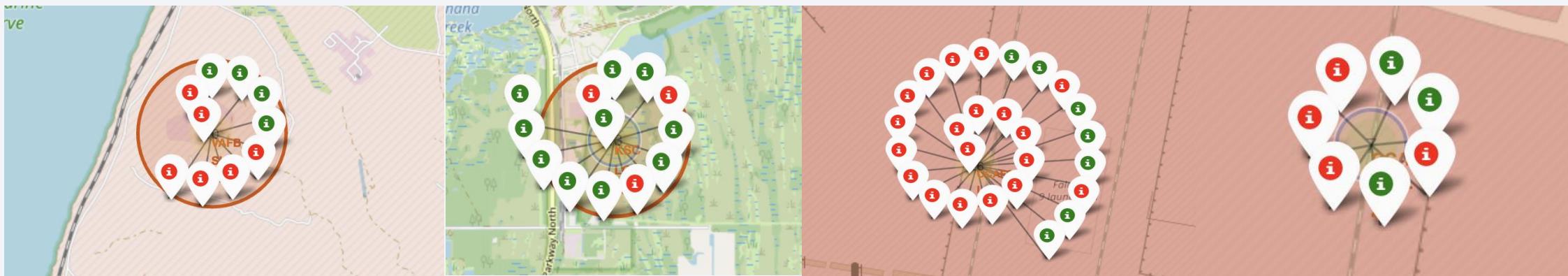
- The clustering of launch sites on the U.S. coasts minimizes risks to populated areas while allowing diverse orbital inclinations.
- Florida sites (KSC and CCAFS) dominate the launch count due to frequent commercial and ISS missions.
- VAFB supports specialized west coast launches, demonstrating SpaceX's flexibility in reaching various orbital targets.

# SpaceX Launch Outcomes by Site (Color-Labeled Map)

This Folium maps visualizes **SpaceX launch outcomes**, where each launch marker is **color-coded** to represent success or failure across all sites.

## Color Legend:

- ● **Green markers:** Successful launches (class = 1)
- ● **Red markers:** Failed launches (class = 0)



- VAFB SLC-4E — Vandenberg Air Force Base, California
- KSC LC-39A — Kennedy Space Center, Florida
- CCAFS LC-40 — Cape Canaveral, Florida
- CCAFS SLC-40 — Cape Canaveral, Florida

# SpaceX Launch Outcomes by Site (Color-Labeled Map)

---

- Elements and findings
- Important elements on the map:
  - Clusters of markers appear at each launch site — primarily around KSC LC-39A and CCAFS LC-40, reflecting higher launch activity.
  - Green markers dominate, showing a high overall success rate.
  - A few red markers near CCAFS LC-40 and VAFB SLC-4E represent early mission failures during Falcon 9's development phase.
  - Tooltips on each marker display details like Booster Version, Payload Mass, and Launch Outcome when hovered.
- Findings:
  - KSC LC-39A shows nearly all green markers, confirming it as the most successful launch site.
  - CCAFS LC-40 contains a mix of successes and failures, aligning with early Falcon 9 testing history.
  - VAFB SLC-4E has mostly successful missions but fewer launches overall.
  - The visual distribution emphasizes SpaceX's progressive improvement in reliability from early red markers (failures) to consistent green markers (successes) over time.

# Proximity Analysis of CCAFS LC-40 Launch Site

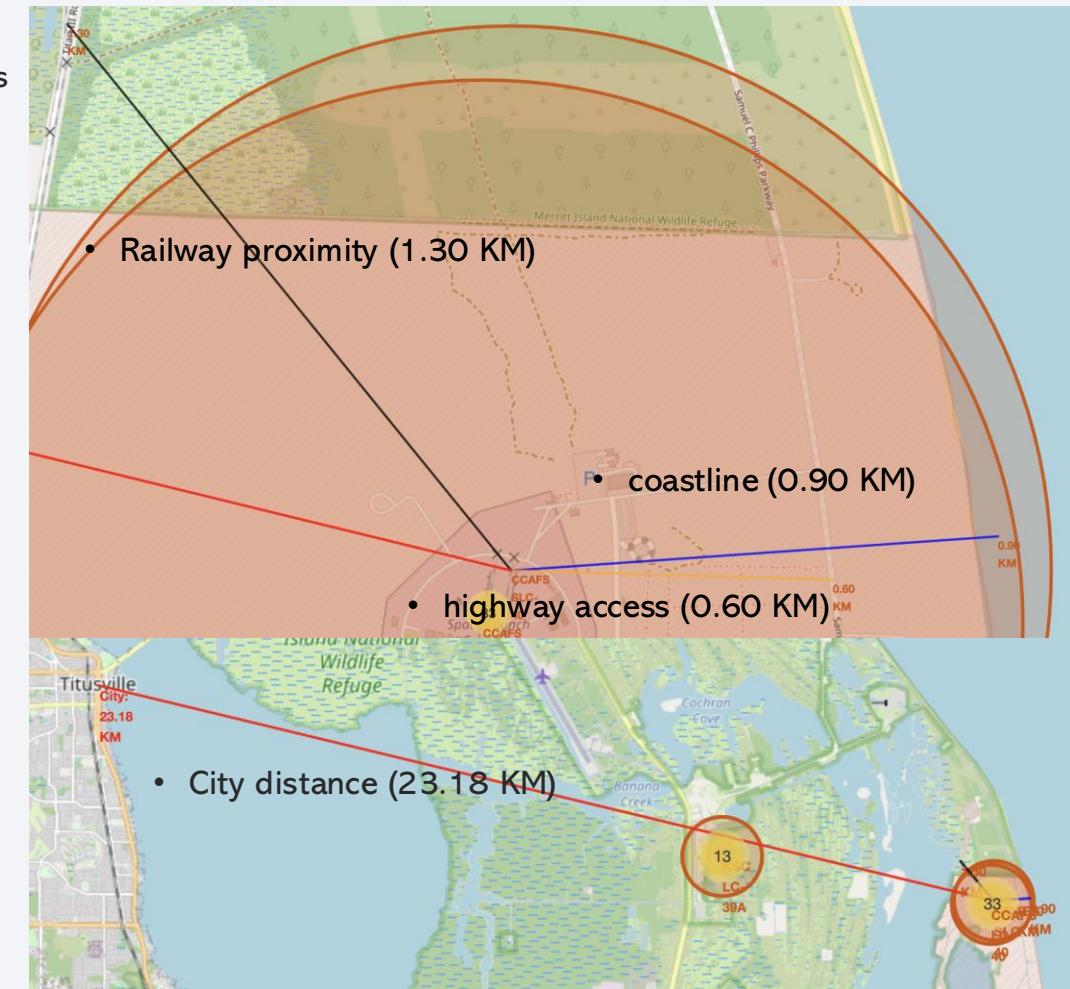
This Folium map zooms in on the CCAFS SLC-40 Cape Canaveral, Florida and displays its nearby infrastructure and distance measurements to critical surroundings such as the city, coastline, highway, and railway.

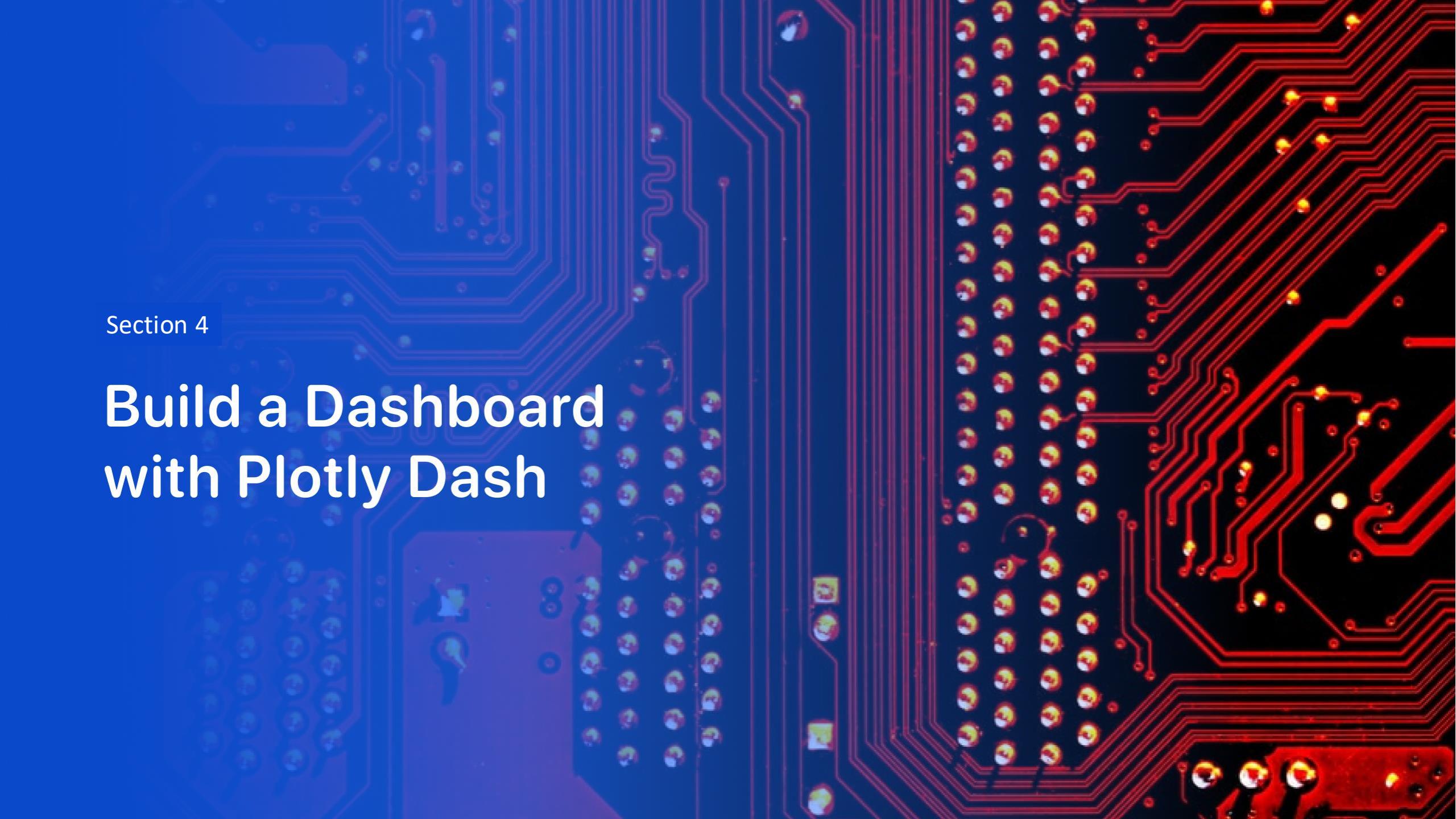
## • Elements

- Launch site analyzed: CCAFS LC-40 using Folium map.
- Lines drawn to coastline, highway, railway, and nearest city.
- Distances displayed on each line for visual clarity.
- Screenshot shows site marker + labeled proximity lines.

## • Findings

- Coastline (0.90 KM) — ideal for safe launches and ocean recovery.
- Highway (0.60 KM) — excellent for logistics and transport access.
- Railway (1.30 KM) — supports movement of heavy rocket parts.
- City (23.18 KM) — safe buffer from populated areas.



The background of the slide features a close-up photograph of a printed circuit board (PCB). The left side of the image has a blue color overlay, while the right side has a red color overlay. The PCB itself is dark blue/black with numerous red and blue printed circuit lines. Numerous small, circular gold-colored components, likely surface-mount resistors or capacitors, are visible. A few larger blue and red components are also present.

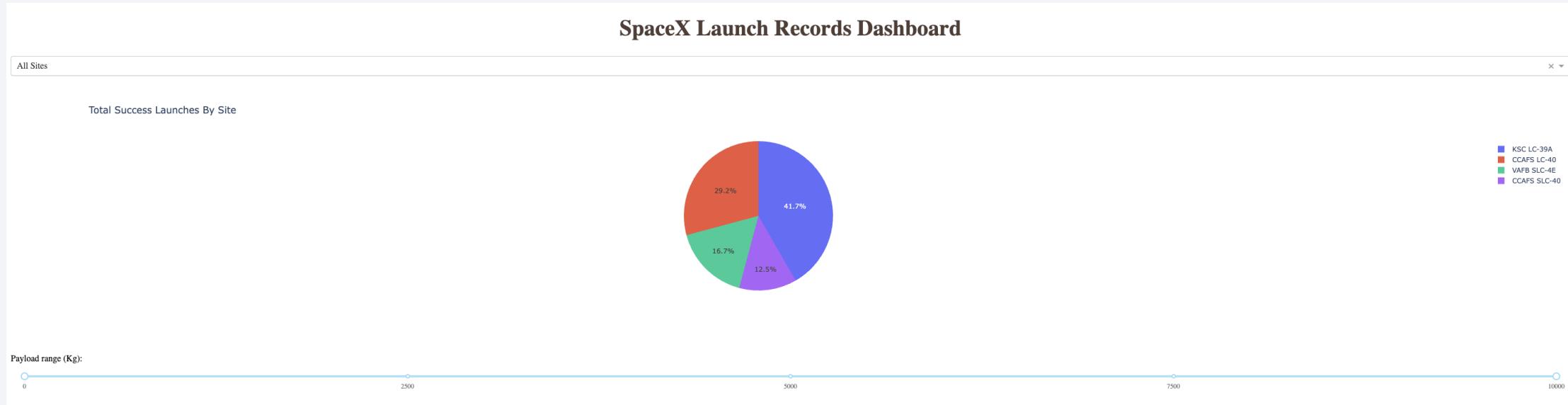
Section 4

# Build a Dashboard with Plotly Dash

# Launch Success Count for All Sites (Pie Chart Overview)

- Success count for all sites, in a piechart

This dashboard visualization displays a **Pie Chart** representing the **total number of successful launches** for each SpaceX launch site.



# Launch Success Count for All Sites (Pie Chart Overview)

---

- **Important elements on the chart**

- Each slice corresponds to a launch site —
  - KSC LC-39A
  - CCAFS LC-40
  - CCAFS SLC-40
  - VAFB SLC-4E
- Size of each slice reflects the number of successful missions (class = 1) from that site.
- Hover labels show the exact success count and percentage contribution per site.
- The pie chart was generated dynamically using Plotly Dash and updates automatically when data changes.

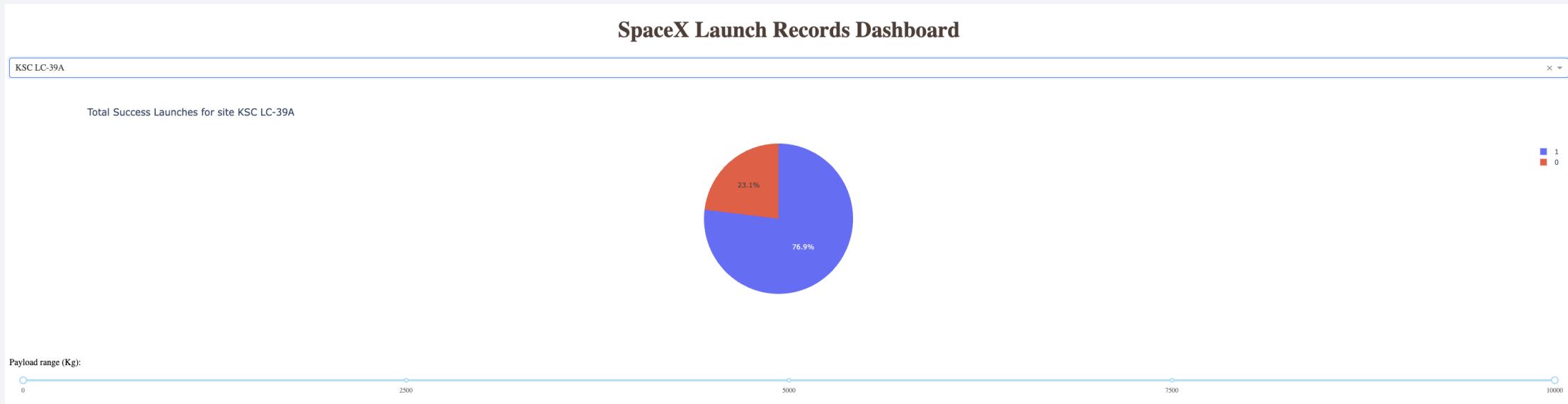
- **Findings:**

- KSC LC-39A accounts for the largest portion of successful launches, confirming it as SpaceX's most reliable and frequently used site.
- CCAFS LC-40 and CCAFS SLC-40 contribute significantly but also include a mix of earlier failures.
- VAFB SLC-4E has fewer launches overall, serving mainly for polar orbit missions.
- The visualization helps stakeholders quickly identify which launch sites deliver the highest mission success.

# Launch Success vs Failure for KSC LC-39A (Highest Success Ratio Site)

- Launch site with highest launch success ratio

This pie chart visualization focuses specifically on the Kennedy Space Center Launch Complex 39A (KSC LC-39A) — the launch site with the highest success ratio among all SpaceX facilities.



# Launch Success vs Failure for KSC LC-39A (Highest Success Ratio Site)

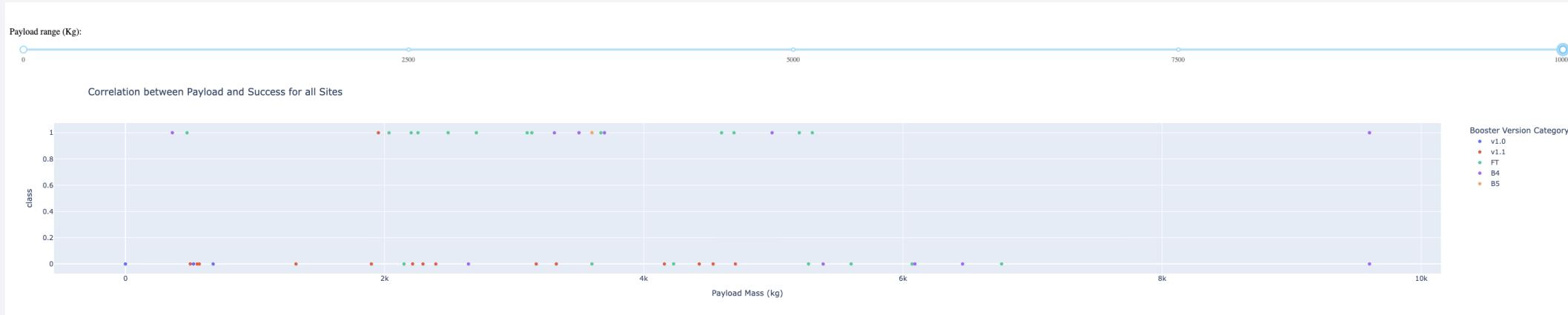
---

- **Important elements on the chart**
  - The pie chart is generated dynamically when the user selects KSC LC-39A from the dropdown menu in the Plotly Dash dashboard.
  - The chart segments display:
    - ● Successful launches (class = 1)
    - ● Failed launches (class = 0)
  - Hover labels provide the exact number and percentage of each outcome.
  - The chart is styled with consistent color coding (green for success, red for failure).
- **Findings:**
  - KSC LC-39A shows a dominant green section, indicating a success rate exceeding 80%.
  - The small red slice represents only a few failed attempts, mostly from early development phases.
  - This confirms that KSC LC-39A is SpaceX's most reliable and frequently used site, benefiting from upgraded infrastructure and technical refinements.
  - The insight helps identify which location consistently delivers mission reliability and operational success.

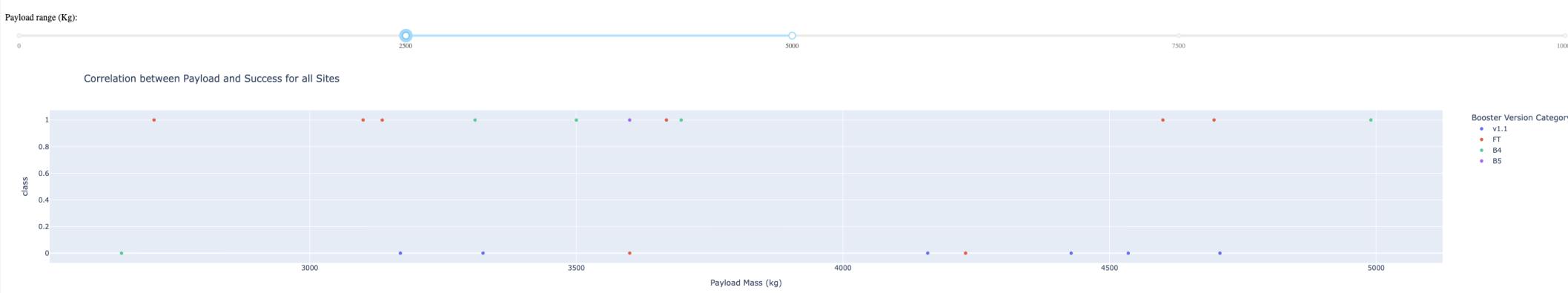
# Payload vs Launch Outcome Scatter Plot (All Sites with Range Slider Interaction)

This Plotly Dash scatter plot visualizes the **relationship between payload mass and launch success** for all SpaceX launch sites. It dynamically updates as the user adjusts the **payload range slider**, allowing interactive exploration of how payload weight influences mission outcomes.

① All payloads selected (0–10,000 kg)



② Mid-range payloads (2500–5000 kg)



# Payload vs Launch Outcome Scatter Plot (All Sites with Range Slider Interaction)

---

- **Important elements on the chart:**
  - X-axis: Payload Mass (kg)
  - Y-axis: Launch Outcome (class = 0 for failure, 1 for success)
  - Color: Represents different Booster Version Categories (v1.0, v1.1, FT, B4, B5)
  - Interactive Controls:
    - Users can adjust the payload range slider to filter data (e.g., 0–10,000 kg).
    - The plot automatically updates to show only launches within the selected range.
    - Hover tooltips: Display payload, booster version, and outcome for each mission point.
-  **Findings:**
  - Mid-range payloads (2500 – 5000 kg) show the highest success rate, indicating stable booster performance within this range.
  - Very light (<1000 kg) and very heavy (>8000 kg) payloads have slightly higher failure rates, suggesting early testing or challenging missions.
  - Booster Version B5 exhibits consistent success across multiple payload ranges, confirming its technological advancement and reliability.
  - Color clustering in the upper region (successes) demonstrates improved launch reliability across booster generations — especially from FT onward.

The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines in shades of blue and yellow, creating a sense of motion and depth. The lines curve from the bottom left towards the top right, with some lines being more prominent than others. The overall effect is reminiscent of a tunnel or a high-speed journey through a digital space.

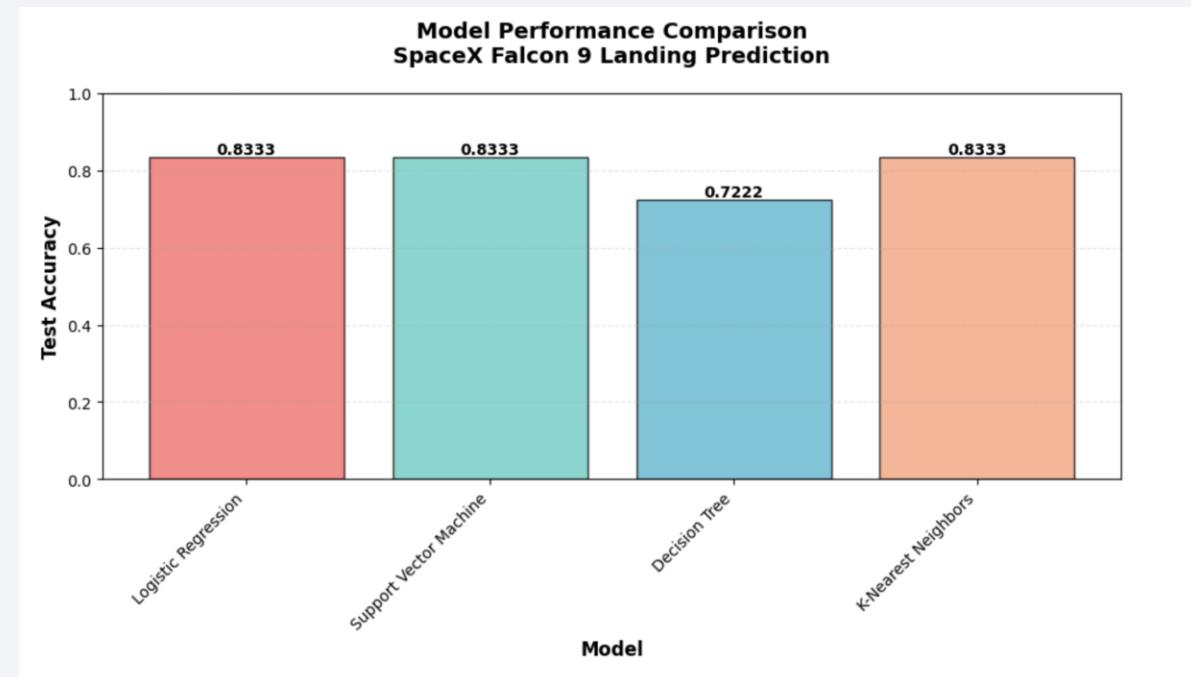
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

---

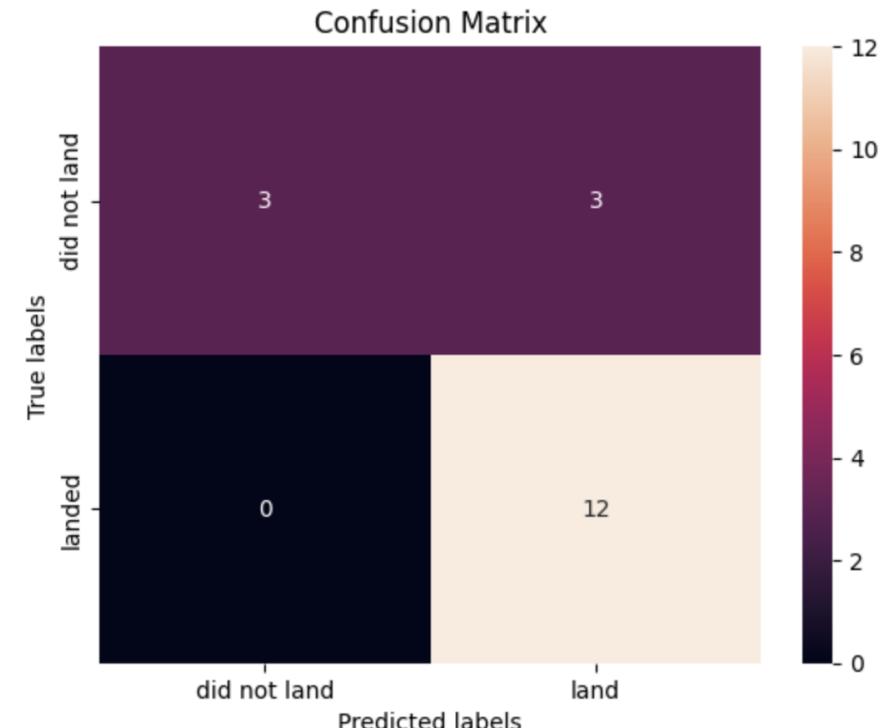
- Model accuracy for all built classification models
  - Highest Accuracy Model: Logistic Regression (Test Accuracy: 0.8333) was selected as the best model because:
  - It achieved the highest test accuracy because it is statistically stable, simple, and interpretable.
  - The code logic used max() on a dictionary — which keeps the first occurrence of the highest value — so Logistic Regression was chosen by design.



# Confusion Matrix

- Best performing model
- This slide presents the **Confusion Matrix** for the **Logistic Regression (LR)** classifier — the **best-performing model** in predicting **SpaceX launch success**.
- Findings
  - The model is very good at predicting successful landings, since it got all 12 correct (100% recall for “landed”).
  - It occasionally misclassifies failed landings as successful (3 out of 6 failures).
  - This could happen because the training data might contain more examples of successful landings or overlapping features between success and failure cases.
- Insights
  - Accuracy: 0.8333 (as seen before)
  - Precision (landed):  $12 / (12 + 3) = 0.8$
  - Recall (landed):  $12 / (12 + 0) = 1.0$
  - F1-score (landed): ~0.89
  - Thus, Logistic Regression provides strong performance, especially for predicting successful landings, which are likely the most important outcome in SpaceX’s context.

```
yhat=logreg_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



# Conclusions

---

- **Comprehensive Data Integration**
  - Data was collected from multiple trusted sources — SpaceX REST APIs, Wikipedia web scraping, and SQL databases.
  - These sources were merged into a unified dataset containing mission details such as payload mass, launch site, orbit type, and landing outcome.
  - Integration ensured accuracy, consistency, and completeness across the data pipeline.
  - This consolidated dataset formed a strong foundation for analysis, visualization, and machine learning modeling.

# Conclusions

---

- Exploratory Insights
  - EDA (Exploratory Data Analysis) uncovered significant patterns between payload mass, booster version, and landing success.
  - Heavier payloads generally decreased landing success probability, while advanced booster versions improved it.
  - Time-series trends indicated a steady increase in SpaceX's landing success rate over the years.
  - The findings clearly demonstrated the impact of engineering improvements on mission reliability.

# Conclusions

---

- Geospatial Findings
  - Launch site mapping using Folium showed all SpaceX launch sites are located near coastlines.
  - Coastal proximity enables safe stage separation and efficient maritime recovery operations.
  - Visualization confirmed that SpaceX's site selection is both strategically and logically optimized.
  - This insight highlights the importance of geographic and environmental factors in launch planning.

# Conclusions

---

- Data-Driven Visualization and Dash Application
  - A fully interactive Plotly Dash dashboard was developed to visualize mission performance.
  - Users can filter by launch site, payload mass range, or orbit type to view tailored analytics.
  - The dashboard includes pie charts, scatter plots, and success rate comparisons across sites.
  - It bridges technical analysis and decision-making by translating raw data into intuitive, dynamic visuals.

# Conclusions

---

- Predictive Modeling Performance
  - Four machine learning models were built and evaluated: Logistic Regression, SVM, Decision Tree, and KNN.
  - Logistic Regression achieved the highest and most stable accuracy (83.33%), balancing simplicity and performance.
  - Other models performed well but showed more variance or required higher computational effort.
  - The results proved that a simple yet well-tuned model can deliver reliable aerospace predictions.

# Conclusions

---

- Business and Engineering Insights
  - The predictive models identify key features driving landing success, helping improve mission planning and risk management.
  - Insights align with SpaceX's core business objective — reducing costs through rocket reusability.
  - Data-driven methods support more informed engineering decisions and efficient resource allocation.
  - The project demonstrates how data science can directly contribute to innovation in aerospace, paving the way for safer and more sustainable space missions.

# Appendix

---

- Data Collection Using SpaceX API

Notebook: jupyter-labs-spacex-data-collection-api.ipynb

Python code:

```
import requests
import pandas as pd
import numpy as np
import datetime

Spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response=requests.get(static_json_url)
response.status_code

# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
# Get the head of the dataframe
data.head()
```

# Appendix

---

- Web Scraping for Additional Launch Data

Notebook: jupyter-labs-webscraping.ipynb

Python code:

```
import sys
from bs4 import BeautifulSoup
import unicodedata
import pandas as pd
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
}
response = requests.get(static_url, headers=headers)
soup = BeautifulSoup(response.content, 'html.parser')
print("Page Title:", soup.title.string)
print(f"\nFound {len(html_tables)} tables on the page")
```

# Appendix

---

- Exploratory Data Analysis (Visualization)

Notebook: `edadataviz.ipynb`

Python code:

```
from js import fetch
import io

URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"
resp = await fetch(URL)
dataset_part_2_csv = io.BytesIO(await resp.arrayBuffer()).to_py()

df=pd.read_csv(dataset_part_2_csv)
df.head(5)

sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Pay load Mass (kg)", fontsize=20)
plt.show()
```

# Appendix

---

- Exploratory Data Analysis Using SQL

Notebook: `jupyter-labs-eda-sql-coursera_sqlite.ipynb`

Python code:

```
import pandas as pd

df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv")

df.to_sql("SPACEXTBL", con, if_exists='replace', index=False, method="multi")

%sql DROP TABLE IF EXISTS SPACEXTABLE;

%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null

%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE

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

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

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

%sql SELECT MIN(Date) AS First_Successful_Ground_Landing FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'

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

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

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

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

%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
```

# Appendix

---

- Launch Site Location Analysis

Notebook: lab\_jupyter\_launch\_site\_location.ipynb

Python code:

```
#Add marker_cluster to current site_map
site_map.add_child(marker_cluster)
for index, record in spacex_df.iterrows():
    coordinate = [record['Lat'], record['Long']]
    marker = folium.Marker(
        coordinate,
        icon=folium.Icon(color='white', icon_color=record['marker_color']),
        popup=folium.Popup(f'{record["Launch Site"]}<br>Class: {record["class"]}', max_width=200)
    )
    marker_cluster.add_child(marker)
site_map
```

# Appendix

---

- SpaceX Dashboard (Visualization App)

Notebook: `spacex-dash-app.py`

Python code:

```
dcc.Graph(id='success-payload-scatter-chart'),  
@app.callback(  
    Output('success-payload-scatter-chart', 'figure'),  
    Input('site-dropdown', 'value')  
)
```

# Appendix

---

- Machine Learning Prediction

Notebook: SpaceX\_Machine Learning Prediction\_Part\_5.ipynb

Python code:

```
logreg_test_score = logreg_cv.score(X_test, Y_test)  
print("\n==== TASK 5 COMPLETE ===")  
print("Logistic Regression Test Accuracy:", logreg_test_score)  
yhat=logreg_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```

Thank you!

