IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Suryansh Sharma>

<23 July 2025>

Outline

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

Executive Summary

This project presents a comprehensive analysis of SpaceX's Falcon 9 launch history, aimed at uncovering operational insights and predicting the likelihood of successful rocket landings.

The process began with **data collection** from SpaceX's publicly available APIs and datasets, capturing key launch attributes such as payload mass, launch sites, booster versions, orbit types, and landing outcomes. The collected data underwent **data wrangling**, where missing values were addressed, data types standardized, and irrelevant entries removed to ensure quality and consistency.

To explore underlying trends, two layers of **Exploratory Data Analysis (EDA)** were performed. The first involved **data visualization** using tools like matplotlib and seaborn, enabling clear identification of correlations—for instance, how booster version and launch site impact success rates. The second layer used **SQL-based EDA** to query launch data directly, allowing precise extraction of patterns over time, such as monthly failure trends and landing outcomes across different platforms (e.g., drone ship vs. ground pad).

Building on these insights, **predictive analysis** was conducted using various **machine learning algorithms**, including Logistic Regression, Decision Trees, K-Nearest Neighbors, and Support Vector Machines. These models were trained and tested to classify landing success, using features like payload mass, orbit type, and flight number. The best-performing model offers actionable predictive insights into mission planning and risk assessment for future Falcon 9 launches.

This end-to-end pipeline reflects the power of combining traditional analytics with modern ML techniques to inform decisions in a high-stakes aerospace domain.

Introduction

- Project background and context

SpaceX's Falcon 9 is a groundbreaking effort in reusable rocketry, aiming to reduce the cost of space access. With rich historical data from dozens of launches, analyzing these missions provides an opportunity to uncover patterns that influence landing success.

This project applies the full data science pipeline—data collection, wrangling, EDA (with both visualizations and SQL), and machine learning—to explore and predict the outcomes of Falcon 9 landings. The goal is to extract insights that can support better mission planning and operational efficiency.

- Problems we want to find answers

- **What launch and booster characteristics most strongly influence a successful landing?**
  By analyzing features such as launch site, payload mass, booster version, and orbit type, we aim to determine which factors correlate with successful outcomes.

- **How have landing outcomes varied over time and across different platforms (e.g., drone ship vs. ground pad)?**
  Using SQL and visualization tools, we examine monthly and yearly trends in mission success/failure.

- **Is it possible to predict the success of a Falcon 9 landing using historical mission data?**
  This problem is tackled with machine learning, using classification algorithms trained on past launch features to predict future landing outcomes.

- **What insights can be derived to support decisions for future launches and improve operational efficiency?**
  Beyond predictive modeling, the goal is to create data-informed recommendations to reduce mission risk and optimize resource allocation.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- The dataset used for this project was sourced from the publicly available SpaceX API and Kaggle repositories, which compile historical data of Falcon 9 launches. This includes information on launch dates, payload mass, launch sites, booster versions, landing outcomes, and orbit types.
- The data was acquired in CSV format and then imported into Python and an SQLite database for further analysis. By using multiple sources, the dataset ensures a comprehensive view of Falcon 9 missions, allowing both structured queries and advanced predictive modeling.

Perform data wrangling

- **Missing Data**: Handled missing values in key columns like Payload Mass (kg) by imputing or removing incomplete rows.
- **Feature Selection & Transformation**: Removed irrelevant columns (e.g. serial numbers), and extracted useful features such as Launch Site, Orbit, and Outcome.
- **Label Engineering**: Converted Outcome into a binary Class variable (1 for successful landing, 0 for failure).
- **Data Type Optimization**: Converted categorical and datetime columns into appropriate formats for analysis.
- **Export**: Final wrangled dataset was saved as dataset_part_2.csv for use in visualization and modeling.

Methodology

Perform exploratory data analysis (EDA) using visualization and SQL

EDA was performed using Seaborn, Matplotlib, and Pandas to uncover patterns in SpaceX's Falcon 9 missions:

- **Launch Success Trends**: Flight success rate improved over time, especially after 2013.
- **Flight Number vs. Payload**: Higher flight numbers showed improved success; heavier payloads often decreased landing success.
- **Launch Site & Orbit Impact**: Certain sites like KSC-LC-39A and orbits like LEO had higher success rates.
- **Orbit & Payload Patterns**: Orbits such as LEO, Polar, and ISS had more favorable outcomes with heavier payloads.
- **Yearly Trends**: Success rate showed an upward trend from 2013 to 2017 and beyond.

Methodology

Perform interactive visual analytics using Folium and Plotly Dash

- To explore Falcon 9 launch data interactively, we employed **two main tools**: Folium for geospatial mapping and Plotly Dash for dynamic dashboard visualizations.

Folium Mapping

- Used to plot launch sites on an interactive map.
- Markers display site names and successful launch info.
- Popups helped visualize location-specific patterns in mission outcomes.

Plotly Dash Dashboard

- Developed a responsive web app to allow users to explore:
- **Launch success rates** using **Pie Charts** (total successes by site or success vs. failure for a specific site).
- **Payload impact on success** via **Scatter Plots**, with color indicating booster versions.
- **Payload filtering** through a **Range Slider**, enabling users to narrow results by payload weight.
- Enabled **dropdown-based site selection** to view insights per site or across all missions.

Methodology

Perform predictive analysis using classification models

To predict the success of Falcon 9 launches, we applied **supervised machine learning techniques** focused on classification.

Model Building

- **Target Variable**: class (1 = Success, 0 = Failure)
- **Features Used**:
  - Payload Mass (kg)
  - Orbit (encoded)
  - Launch Site (encoded)
  - Booster Version Category (encoded)
- Performed **One-Hot Encoding** for categorical features.

Model Selection & Tuning

We implemented and compared multiple classifiers:

- **Logistic Regression**
- **Decision Tree**
- **Support Vector Machine (SVM)**
- **K-Nearest Neighbors (KNN)**

Tuning was done using:

- **GridSearchCV** for hyperparameter optimization (e.g., depth in Decision Trees, C & kernel in SVM)
- **Cross-validation** to reduce variance in performance evaluation.

Evaluation Metrics

We used the following metrics to evaluate and compare models:

- **Accuracy Score**
- **Precision, Recall, F1-Score**
- **Confusion Matrix**
- **ROC-AUC Curve** (for probabilistic interpretation)

Data Collection

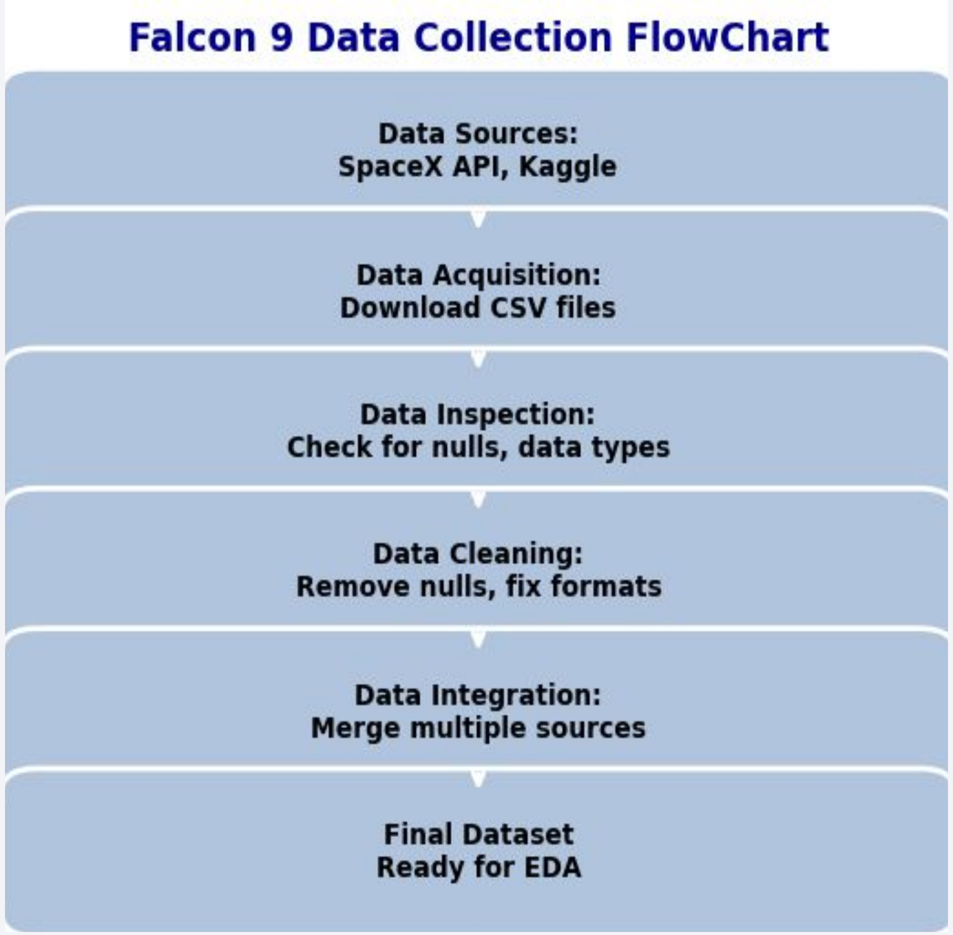
To analyze SpaceX Falcon 9 launches, we sourced and curated datasets using **web scraping**, **API extraction**, and **manual CSV compilation**. The focus was to gather **structured**, **relevant**, and **clean data** on past missions.

Data Sources Used:

- **SpaceX API** – for dynamic and updated launch data
- **SpaceX Launch Archives** (Wikipedia & official sites) – for historical mission metadata
- **Kaggle Falcon 9 Dataset** – for payloads, orbits, booster info, and success classification

Key Techniques Employed

- **API to JSON Parsing** using requests and json libraries
- **Web Scraping** with BeautifulSoup for HTML table extraction
- **Data Cleaning & Merging** in **Pandas**
- **Standardization** of formats (e.g., date, mass units)



Falcon 9 Data Collection FlowChart

Data Sources:
SpaceX API, Kaggle

Data Acquisition:
Download CSV files

Data Inspection:
Check for nulls, data types

Data Cleaning:
Remove nulls, fix formats

Data Integration:
Merge multiple sources

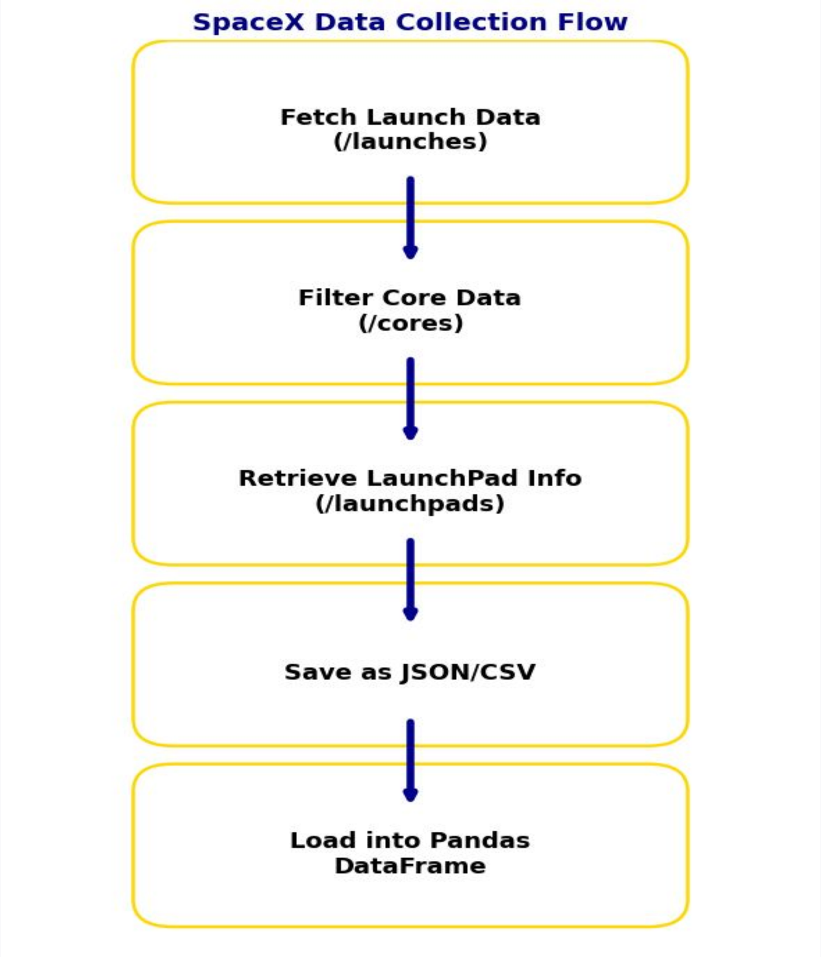
Final Dataset
Ready for EDA

Data Collection – SpaceX API

Key Phrases

- **Public API Used**: SpaceX API v4
- **Method**: RESTful GET requests
- **Format**: JSON
- **Tools**: Python, requests module, pandas for DataFrame conversion
- **Purpose**: To extract real-time and historical launch data including:
  - Launch site
  - Payload mass
  - Launch success/failure
  - Rocket type
  - Launch date

- https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

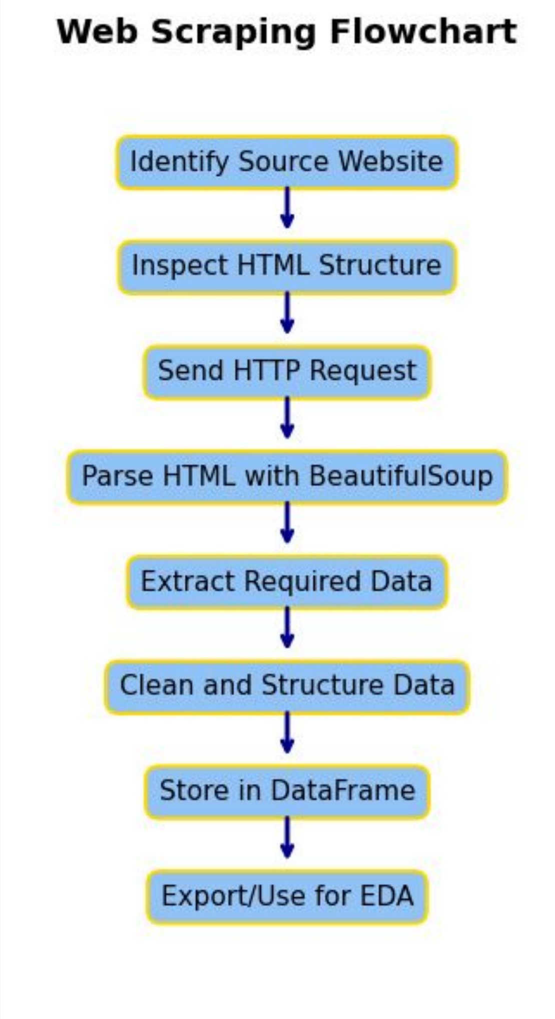

SpaceX Data Collection Flow
Fetch Launch Data (/launches)
Filter Core Data (/cores)
Retrieve LaunchPad Info (/launchpads)
Save as JSON/CSV
Load into Pandas DataFrame

Data Collection - Scraping

Key Phrases – Web Scraping Process for Falcon 9 Data

1. **Identify Source Website** – NASA Spaceflight & Wikipedia launch logs.
2. **Inspect HTML Structure** – Use browser dev tools to locate relevant elements.
3. **Send HTTP Requests** – Use requests to fetch page content.
4. **Parse HTML** – Use BeautifulSoup to extract data fields (e.g., mission name, launch date, payload, etc.).
5. **Clean Extracted Data** – Remove HTML tags, format dates, handle missing values.
6. **Convert to DataFrame** – Store structured data for analysis.
7. **Export/Append to CSV** – Save data locally or merge with API data.

- https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-



Data Wrangling

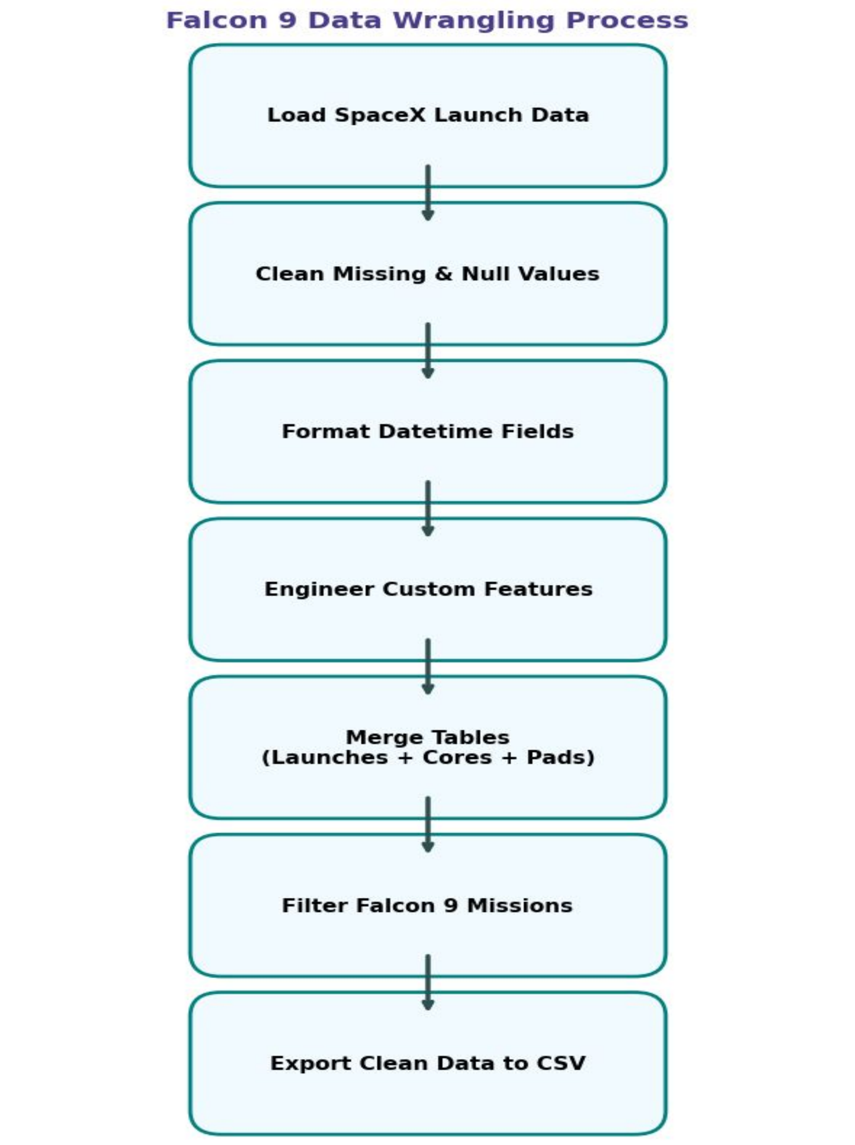
Data Wrangling Overview

Data wrangling involved transforming raw SpaceX data into a structured, analysis-ready format. The process included **cleaning**, **standardizing**, and **merging** data from various sources.

Key Phrases in the Wrangling Process:

- **Load JSON/CSV Files:** Using pandas, we loaded launch, core, and launchpad datasets into DataFrames.
- **Missing Value Handling:** Replaced or removed NaN, None, and null entries, especially in launch outcomes and rocket reuse metrics.
- **Datetime Standardization:** Converted all timestamp fields (e.g., launch_date_utc) to Python datetime objects.
- **Feature Engineering:** Created new features like:
    - mission_success (binary success/failure)
    - reused_booster (True/False)
    - landing_class (categorical label)
- **Data Type Conversion:** Categorical features (e.g., rocket names, orbit type) were encoded or converted using LabelEncoder.
- **Data Merging:** Merged launch data with cores and launchpads on keys like core_serial and site_id.
- **Filtering & Cleanup:** Filtered only **Falcon 9** missions and dropped irrelevant features like mission links or webcast URLs.

- https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-


Falcon 9 Data Wrangling Process

Load SpaceX Launch Data

Clean Missing & Null Values

Format Datetime Fields

Engineer Custom Features

Merge Tables
(Launches + Cores + Pads)

Filter Falcon 9 Missions

Export Clean Data to CSV

EDA with Data Visualization

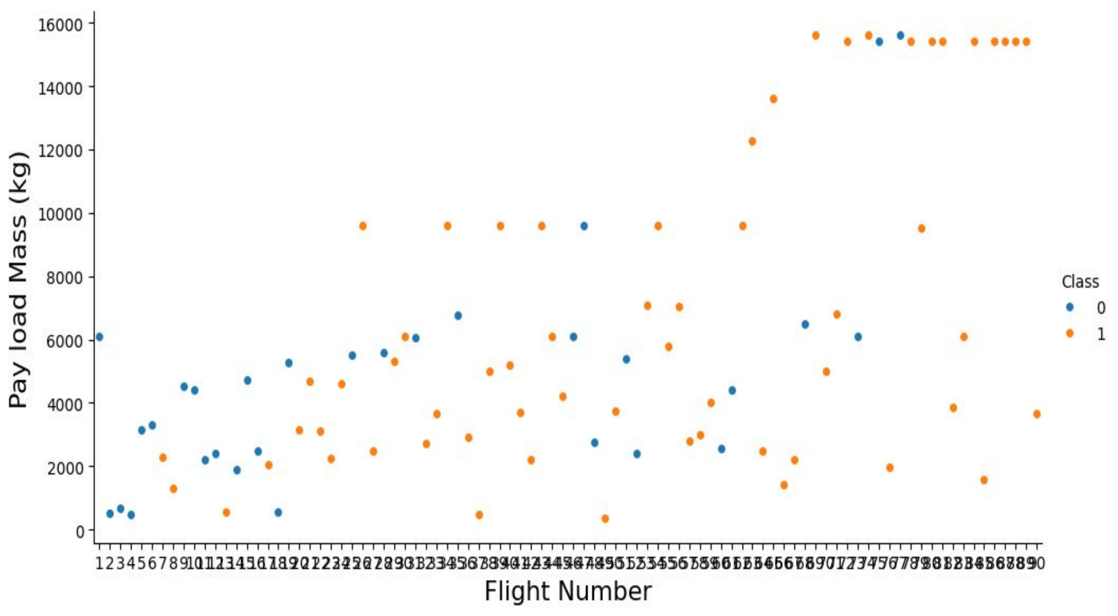
GitHub URL for completed labs = https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

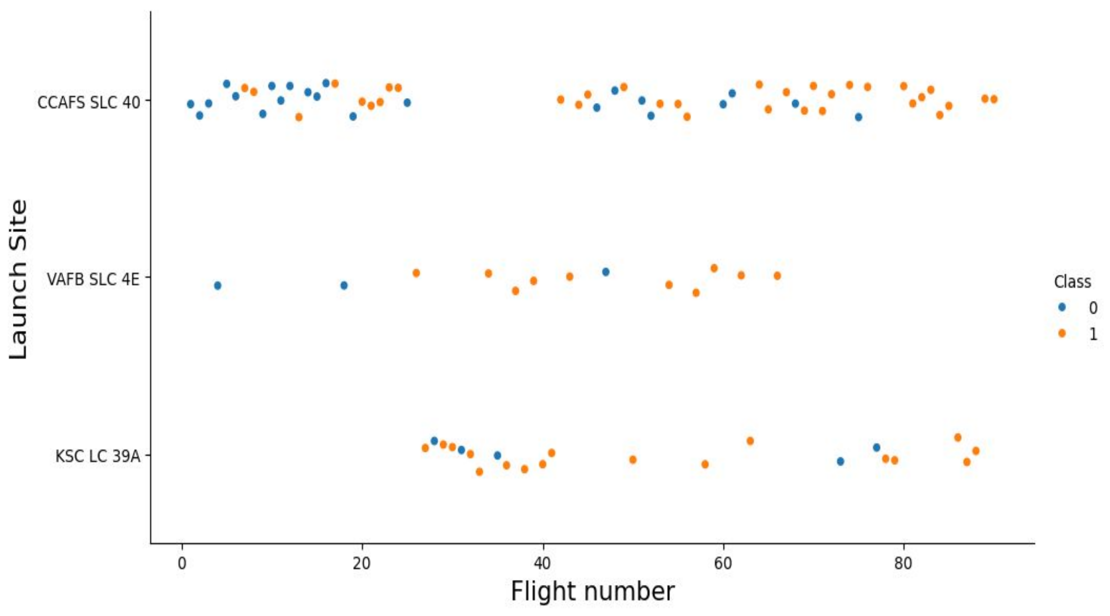
Flight Number vs. Payload Mass (sns.catplot)

- To observe whether mission count (experience) and payload size influence landing success.
- **Insight**: Higher flight numbers showed increased success, while heavier payloads reduced it.

Flight Number vs. Launch Site (sns.catplot)

- **Why**: To evaluate success distribution across launch sites over time.
- **Insight**: Some sites had more consistent success rates as experience increased.

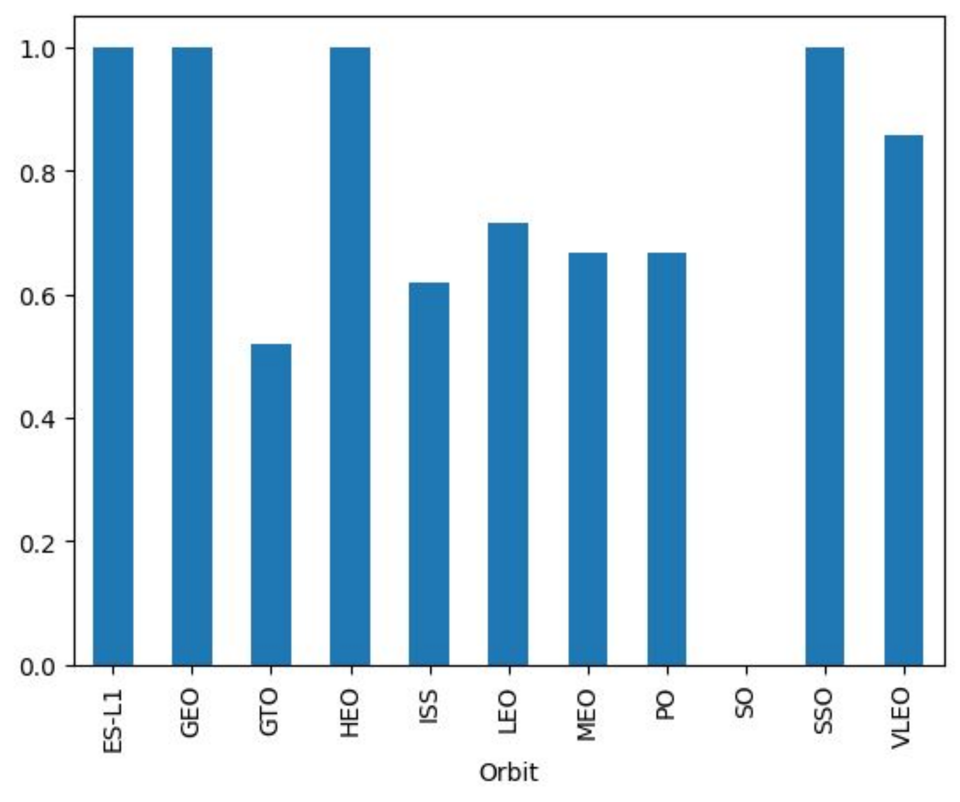




EDA with Data Visualization

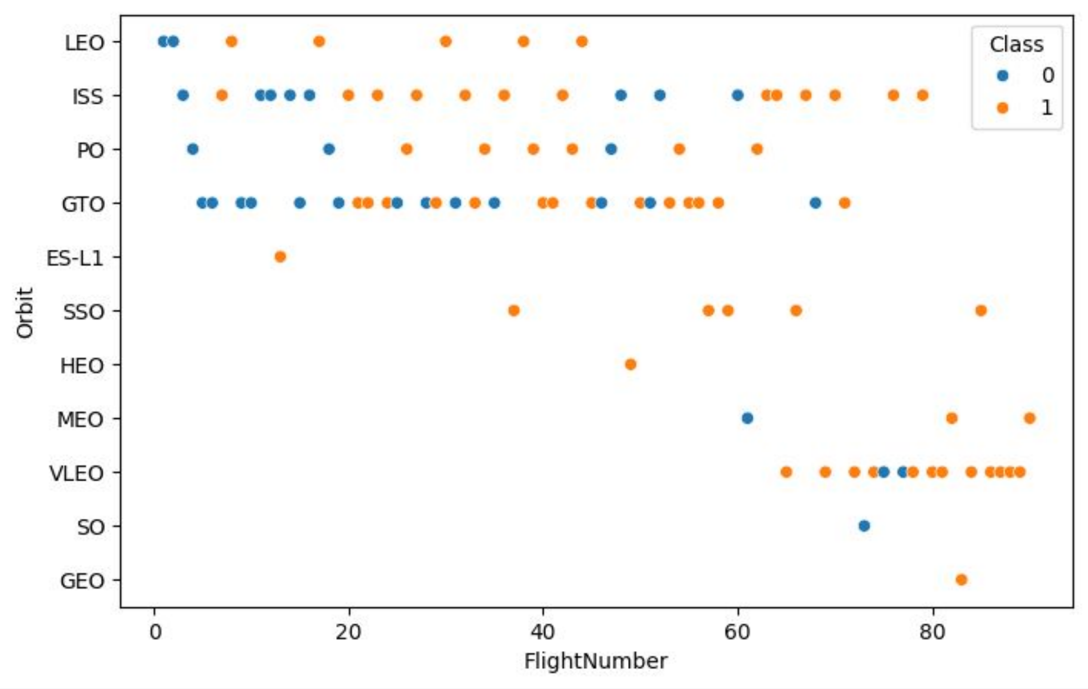
Orbit Type vs. Success Rate (bar plot)

- **Why**: To identify which orbits correlate with higher landing success.
- **Insight**: LEO and Polar orbits had the highest success rates.



Flight Number vs. Orbit Type (sns.scatterplot)

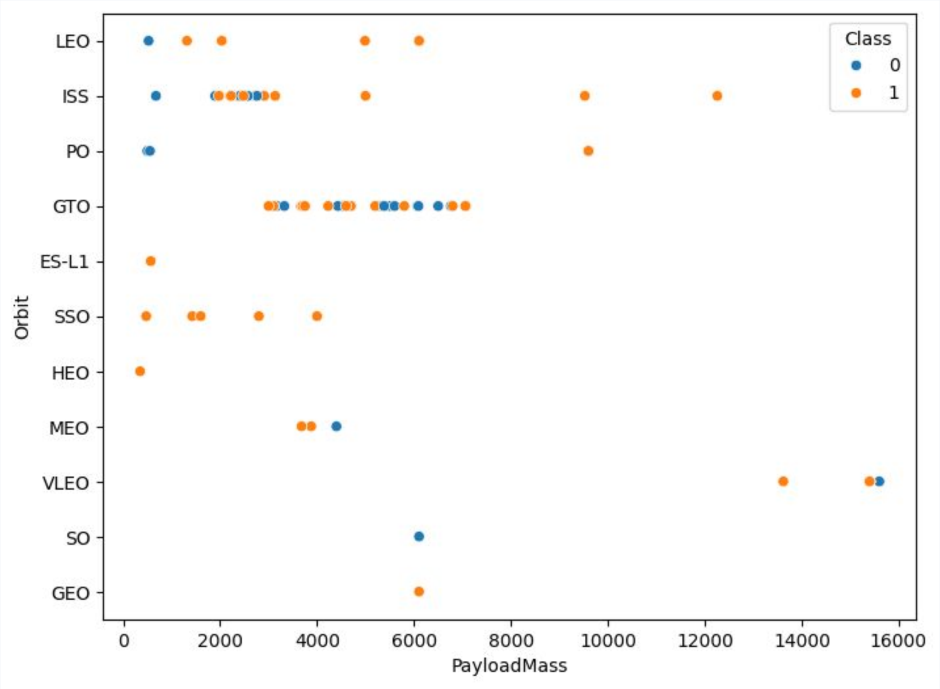
- **Why**: To check whether mission experience improved success across different orbits.
- **Insight**: More evident positive correlation in LEO; GTO remained variable.



EDA with Data Visualization

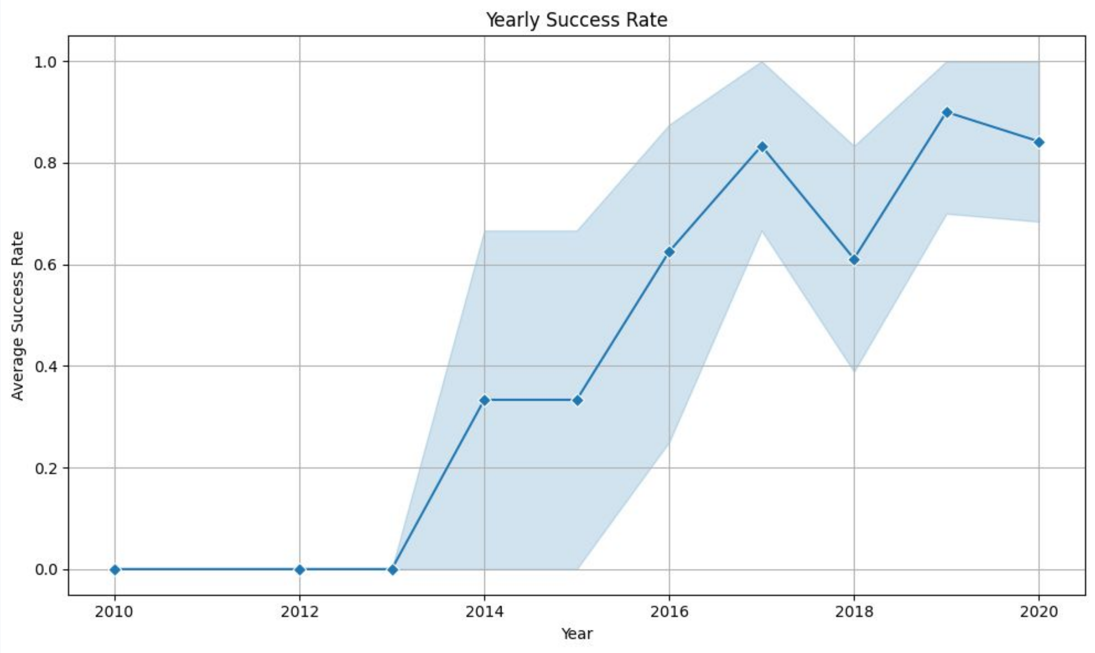
Payload Mass vs. Orbit Type (sns.scatterplot)

- **Why**: To examine if orbit type combined with payload size affects outcomes.
- **Insight**: Heavy payloads succeeded more in LEO/ISS, less clear in GTO.



Yearly Success Rate (line plot)

- **Why**: To track SpaceX's performance trend over time.
- **Insight**: Success rate steadily improved from 2013 to 2017, showing operational maturity.



EDA with SQL

GitHub URL for completed labs = https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

These queries provided structured insight, enabling data cleaning, feature creation, and informed visual exploration.

- **Total Number of Falcon 9 Launches**
  - Queried the count of all launches where rocket name was **'Falcon 9'**.
- **Successful Launches Count**
  - Retrieved the number of Falcon 9 missions with a **successful landing outcome**.
- **Launch Sites Used by SpaceX**
  - Extracted **distinct launch site names** from the database.
- **Average Payload Mass**
  - Calculated the **mean payload mass** across all Falcon 9 launches.
- **Max and Min Payload Mass by Customer**
  - Used GROUP BY and MAX/MIN to analyze **payload range per customer**.
- **Top Launch Sites by Mission Count**
  - Used GROUP BY and ORDER BY to identify **most frequently used launch sites**.
- **Year-wise Launch Success Count**
  - Extracted the **number of successful launches** per year using YEAR() function and aggregation.
- **Orbit Success Analysis**
  - Queried landing success rates **by orbit type** to explore mission outcome trends.
- **Joined Tables: Launches + Cores + Pads**
  - Combined launch data with core and launchpad tables using **JOIN statements** to enrich analysis.
- **Filtered GTO Orbit Missions**
  - Selected only those launches targeting **Geostationary Transfer Orbit (GTO)** for focused analysis.

Build an Interactive Map with Folium

GitHub URL for completed labs = https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

Findings These Map Objects Enabled:

- Most launch sites are **close to coastlines**, reducing risk to populated areas.
- Sites are **near highways and railways**, suggesting logistic accessibility.
- They **maintain distance from major cities**, likely for safety and airspace control.
- Launch outcomes varied by site—visible through clustered success/failure markers.
- **folium.Circle**
  - **Purpose**: Highlighted each launch site's exact geographic position.
  - **Why**: Helps visually identify launch site clusters and their proximity to coastlines or infrastructure.
- **folium.Marker**
  - **Purpose**: Placed a labeled marker on each launch site.
  - **Why**: Allows quick identification of launch sites like "CCAFS LC-40" or "VAFB SLC-4E" when exploring the map.
- **folium.Icon Markers for Launch Outcome**
  - **Purpose**: Color-coded success (green) and failure (red) outcomes.
  - **Why**: Enables visual pattern recognition of which sites perform better over time.
- **MarkerCluster**
  - **Purpose**: Grouped densely overlapping outcome markers at each site.
  - **Why**: Reduces clutter and improves readability for sites with many launches.
- **folium.PolyLine**
  - **Purpose**: Drew lines from launch sites to nearby features (e.g., coastlines, highways, cities).
  - **Why**: Helped visualize and measure distances to understand environmental and logistical factors.
- **folium.DivIcon**
  - **Purpose**: Displayed numeric distance values on the map.
  - **Why**: Made proximity measurements (like “1.27 KM to coast”) clear and readable without external references.
- **MousePosition Plugin**
  - **Purpose**: Displayed live lat/long coordinates under the mouse pointer.
  - **Why**: Essential for manually identifying coordinates of proximity features such as railways or roads.

Build a Dashboard with Plotly Dash

Dashboard Summary
https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

1. Pie Chart: Launch Success Distribution

- **Interaction Trigger:** Launch Site Dropdown (site-dropdown)
- **What it shows:** If **All Sites** is selected → displays total successful launches per site. If a **specific site** is selected → shows the **success vs failure** ratio at that site.
- **Why added:** Gives a clear overview of how each launch site performs and allows quick comparisons of success rates across locations.

3. Dropdown Selector: Launch Site

- **Purpose:**
  Allows users to explore data for **individual sites** or **all launch sites combined**.
- **Why added:**
  Empowers the user to drill down into specific data slices and tailor the dashboard view based on their focus.

2. Scatter Plot: Payload vs Launch Success

- **Interaction Triggers:**
  - Launch Site Dropdown (site-dropdown)
  - Payload Range Slider (payload-slider)
- **What it shows:** Relationship between **payload mass** and **mission outcome** (success or failure). Colored by **booster version category** for more detailed insight.**Why added:** Useful for identifying whether heavier payloads correlate with higher or lower success rates. The color categorization also reveals which booster models perform better.

4. Range Slider: Payload Mass (kg)

- **Purpose:**
  Lets users filter launches based on **payload mass range**.
- **Why added:**
  Helps in investigating performance trends with light vs heavy payloads, and ensures flexibility in exploring the dataset.

Predictive Analysis (Classification)

GitHub URL for completed labs = https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

Model Development Process Summary

1. Data Preparation

- Extracted relevant features (Payload Mass, Orbit, Launch Site, Booster Version, etc.).
- Encoded categorical variables using **One-Hot Encoding**.
- Normalized data with **StandardScaler**.
- Defined feature set X and target variable y.
- Split data into training and test sets (80:20 split).

2. Model Building

Trained the following classification models:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **Decision Tree Classifier**
- **K-Nearest Neighbors (KNN)**

3. Hyperparameter Tuning

Used **GridSearchCV** with **10-fold cross-validation** for each model:

- Tuned hyperparameters (e.g., C for Logistic Regression, k for KNN, max_depth for Decision Tree).
- Trained multiple configurations to find optimal combinations.

4. Model Evaluation

- Evaluated models using:
  - **Accuracy scores** on test set.
  - **Confusion matrices** to assess TP, FP, FN, TN.
  - **Classification report** (precision, recall, F1-score) if needed.

5. Final Model Selection

- Compared test accuracies and cross-validation scores.
- Selected the model with the **highest generalization performance**.
- Visualized the results using bar charts or tables for easier comparison.

Results

GitHub URL for completed labs = https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

Exploratory Data Analysis (EDA) Results

- **Dataset Overview**:
  - The dataset contains academic, behavioral, and demographic features of students.
  - Target variable: **Dropout status** (Binary classification).
- **Missing Values**:
  - Checked and handled missing/null values.
  - Imputation or removal strategies applied where necessary.
- **Class Distribution**:
  - Target classes were imbalanced to some extent; addressed during modeling via metrics awareness (e.g., precision/recall).
- **Univariate Analysis**:
  - Histograms and count plots revealed:
  - Skewed distributions in features like **grades**, **absenteeism**, and **parental education**.
  - Dropout students had higher absenteeism and lower academic scores.
- **Bivariate Analysis**:
  - Used boxplots, scatter plots, and grouped bar charts.
  - Strong correlation seen between **low grades**, **disciplinary issues**, and **dropout probability**.
- **Correlation Matrix**:
  - Highlighted key positive correlations:
  - Absenteeism ↑ → Dropout ↑
  - Grade Avg ↓ → Dropout ↑
  - Parent Engagement ↓ → Dropout ↑

Results

GitHub URL for completed labs = https://github.com/suryansh-spec/Falcon-9-first-stage-Applied-Data-Science-Capstone-

Exploratory Data Analysis (EDA) Results

Models Trained:

- **K-Nearest Neighbors (KNN)**
- **Support Vector Machine (SVM)**
- **Logistic Regression**
- **Decision Tree**

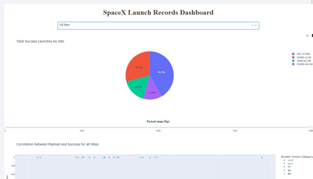
Evaluation Metrics:

- Used **Train Accuracy** and **Test Accuracy** to measure performance.
- F1-Score or Confusion Matrix are used.

Model	Train Accuracy	Test Accuracy
KNN	0.84464	0.94444
SVM	0.83210	0.88888
Logistic Regression	0.80351	0.94444
Decision Tree	0.85890	0.83333

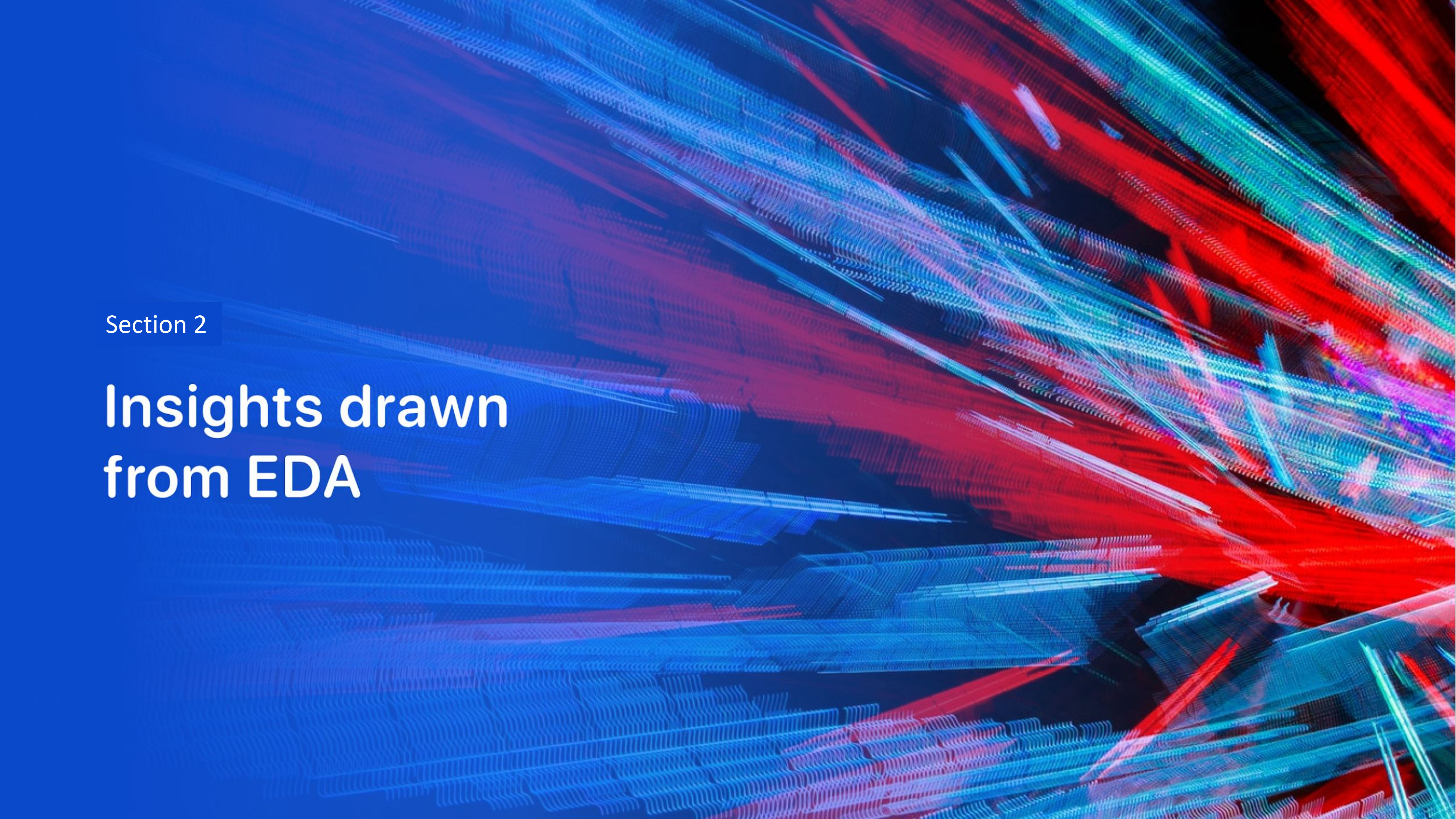
Best Performing Model:

- **KNN** provided the **best balance** between training and testing performance.
- **High test accuracy (0.9444)** with minimal overfitting made it the **top candidate**.

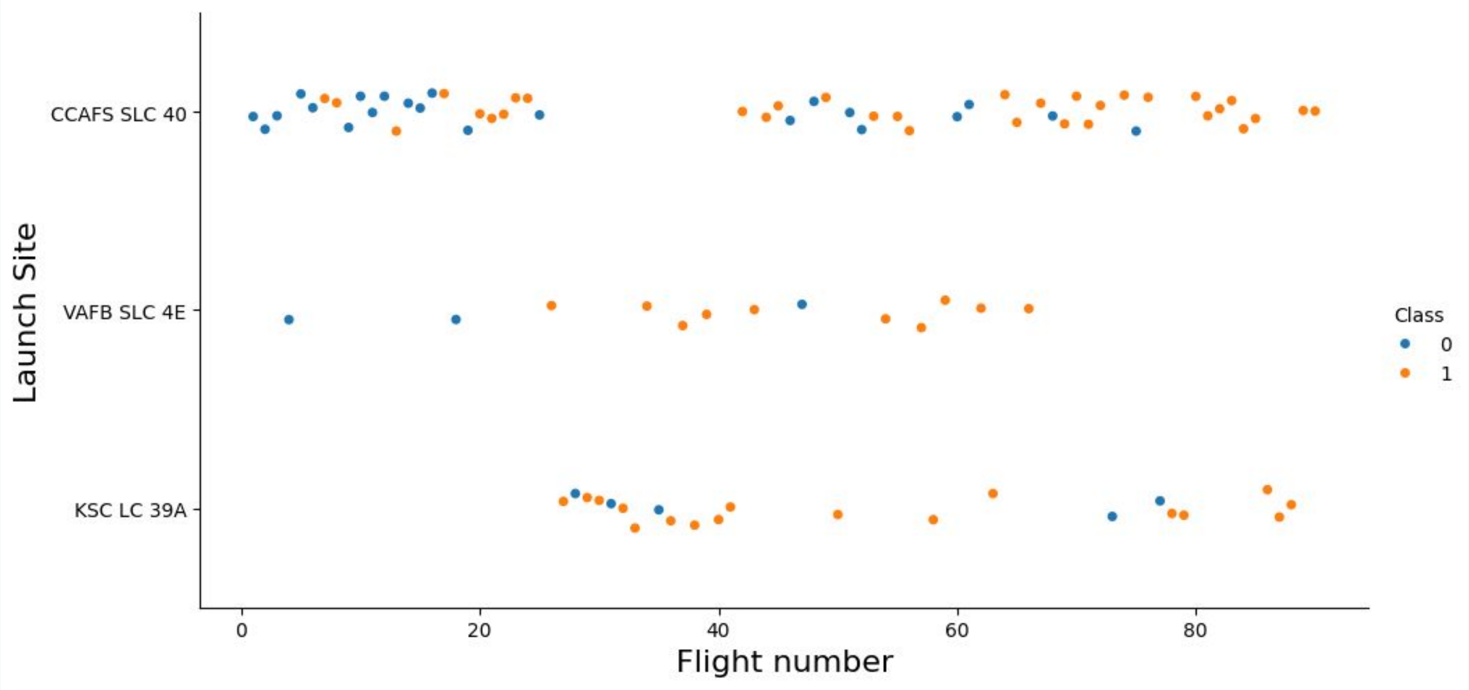


Section 2

Insights drawn from EDA

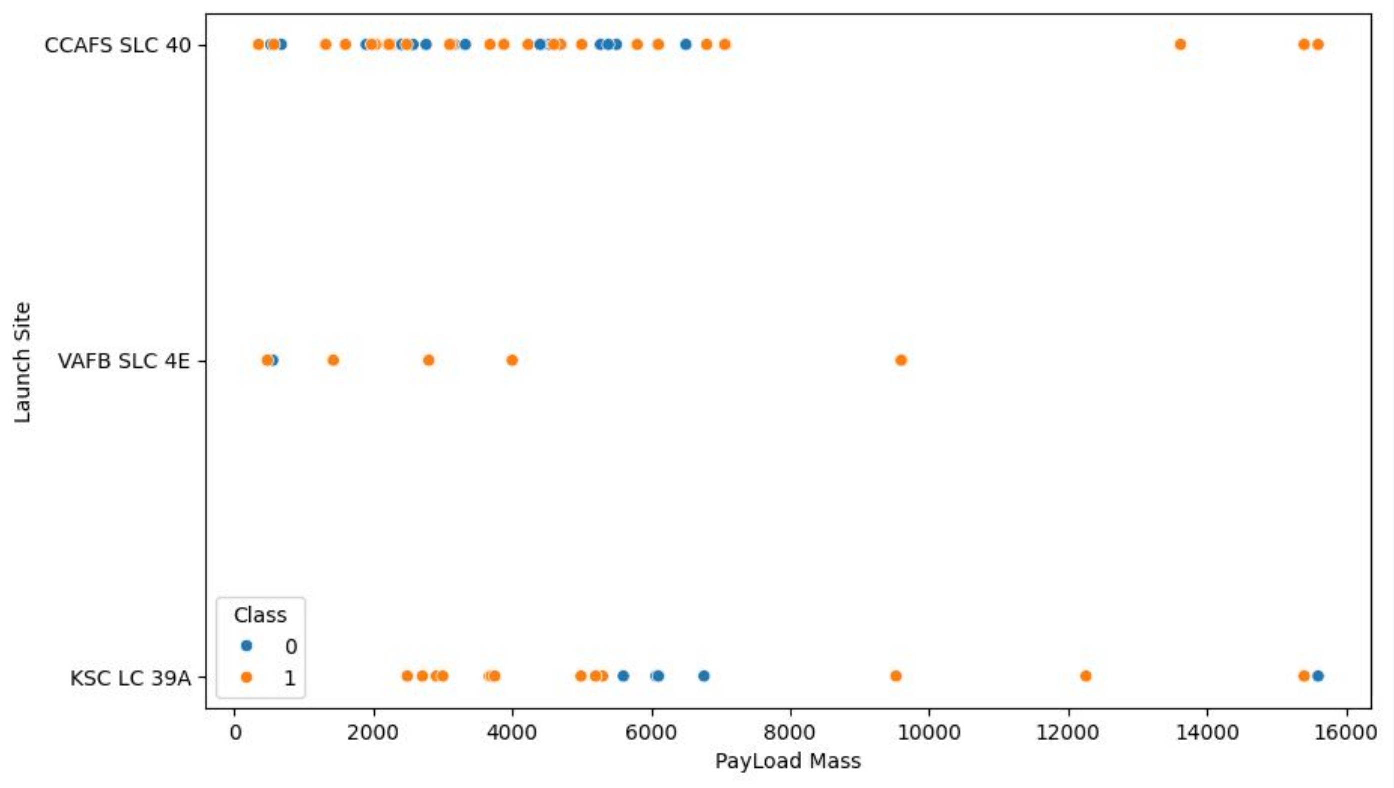


Flight Number vs. Launch Site



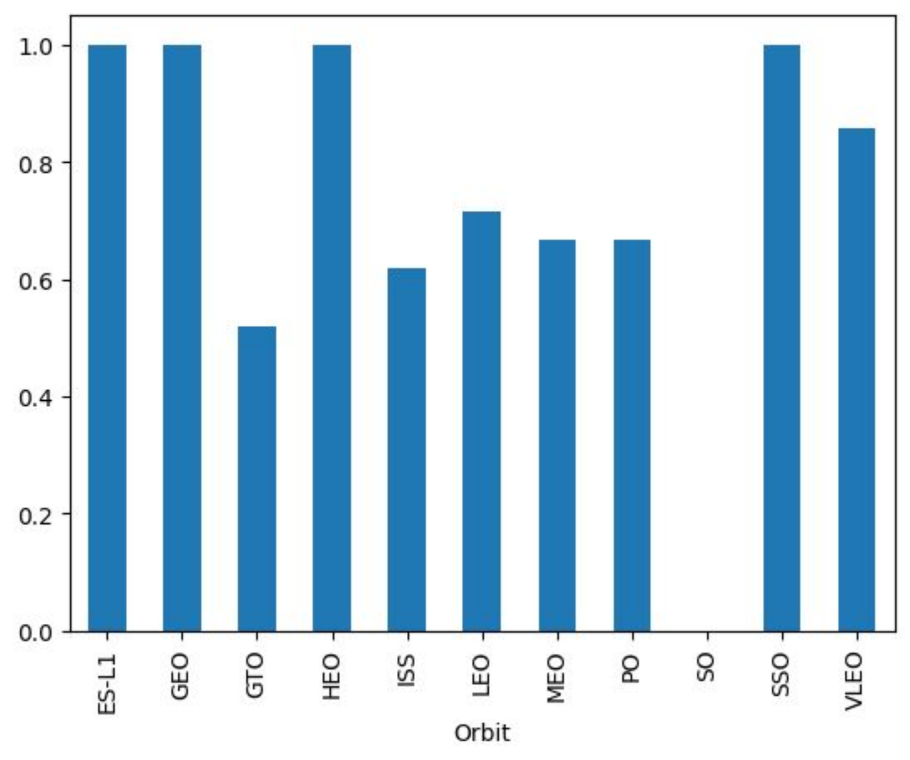
- To evaluate success distribution across launch sites over time.
- **Insight**: Some sites had more consistent success rates as experience increased.

Payload vs. Launch Site



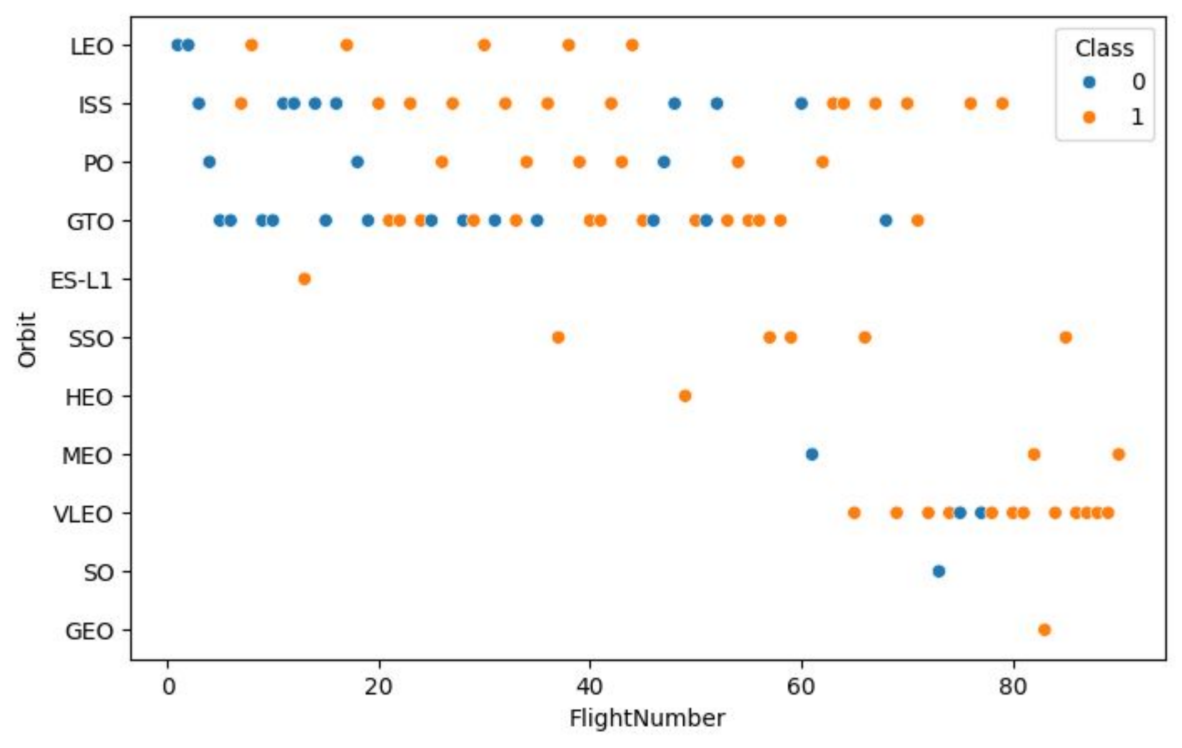
- If you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type



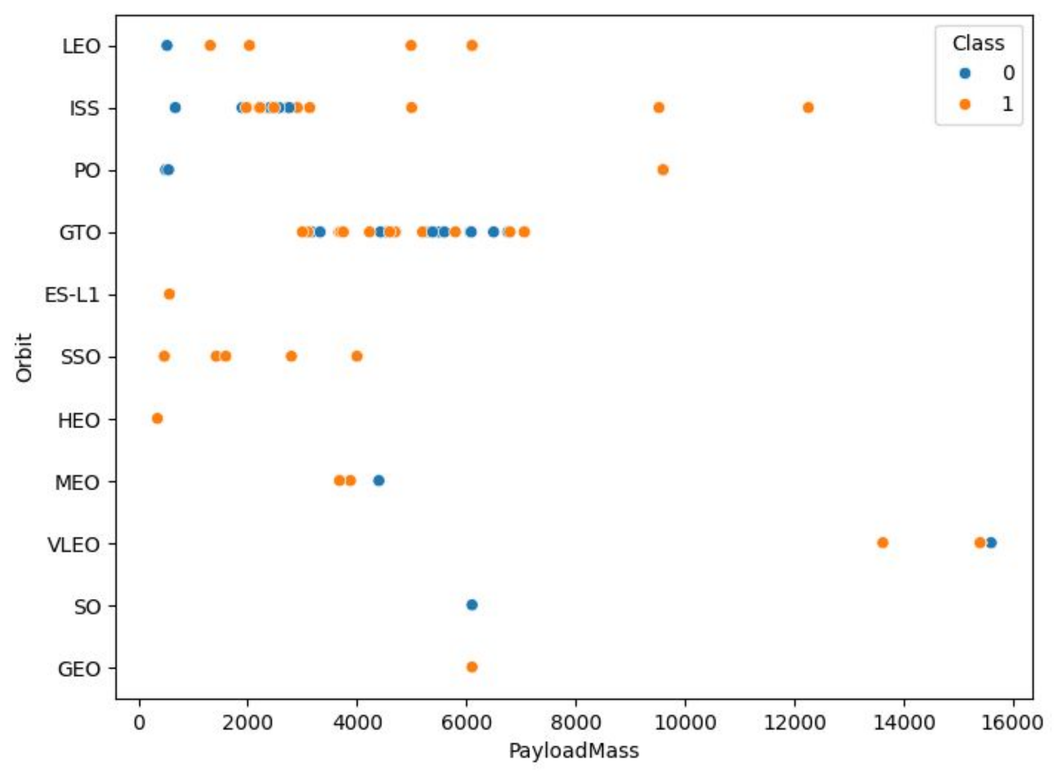
- To identify which orbits correlate with higher landing success.
- **Insight**: LEO and Polar orbits had the highest success rates.

Flight Number vs. Orbit Type



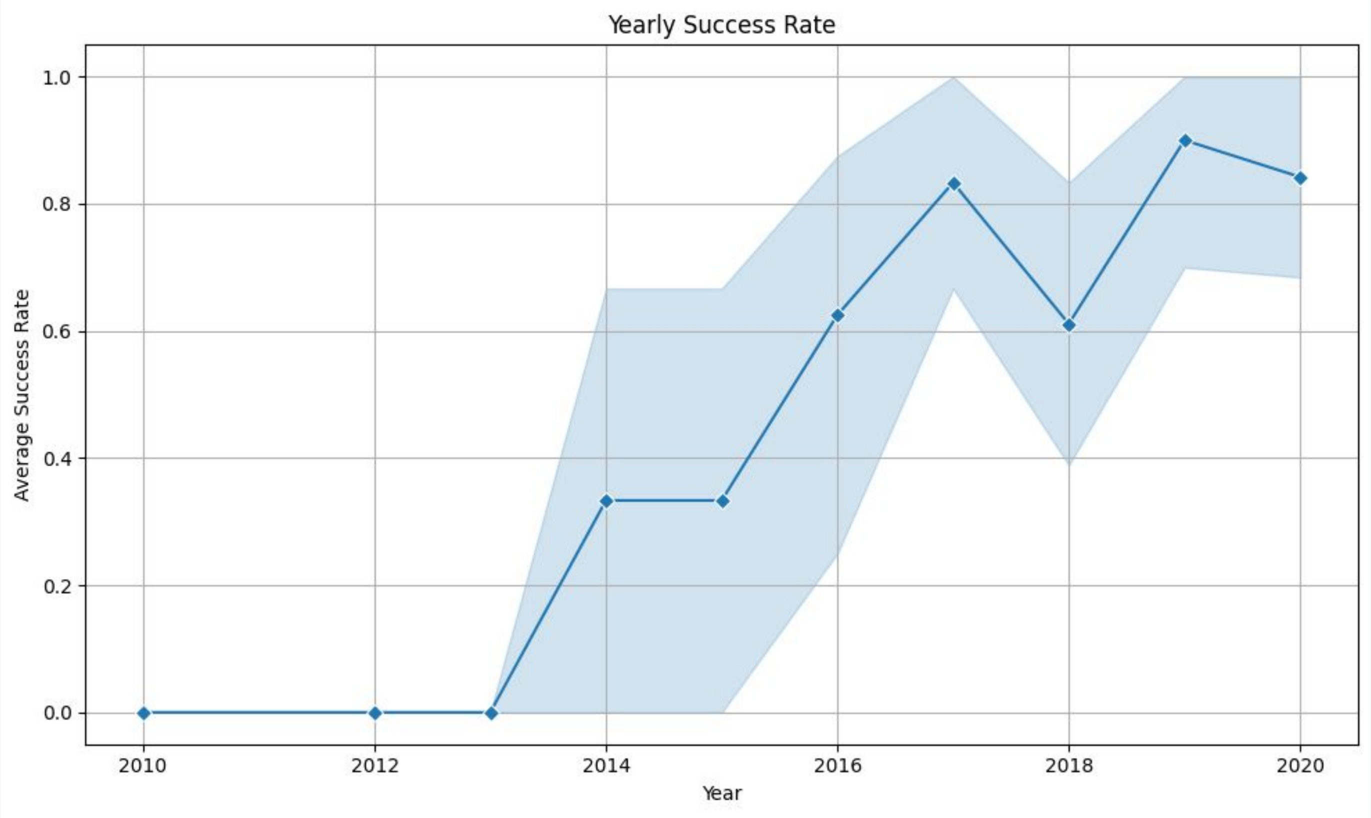
- To check whether mission experience improved success across different orbits.
  - **Insight**: More evident positive correlation in LEO; GTO remained variable.

Payload vs. Orbit Type



- To examine if orbit type combined with payload size affects outcomes.
- **Insight**: Heavy payloads succeeded more in LEO/ISS, less clear in GTO.

Launch Success Yearly Trend



- To track SpaceX's performance trend over time.
- **Insight**: Success rate steadily improved from 2013 to 2017, showing operational maturity.

All Launch Site Names

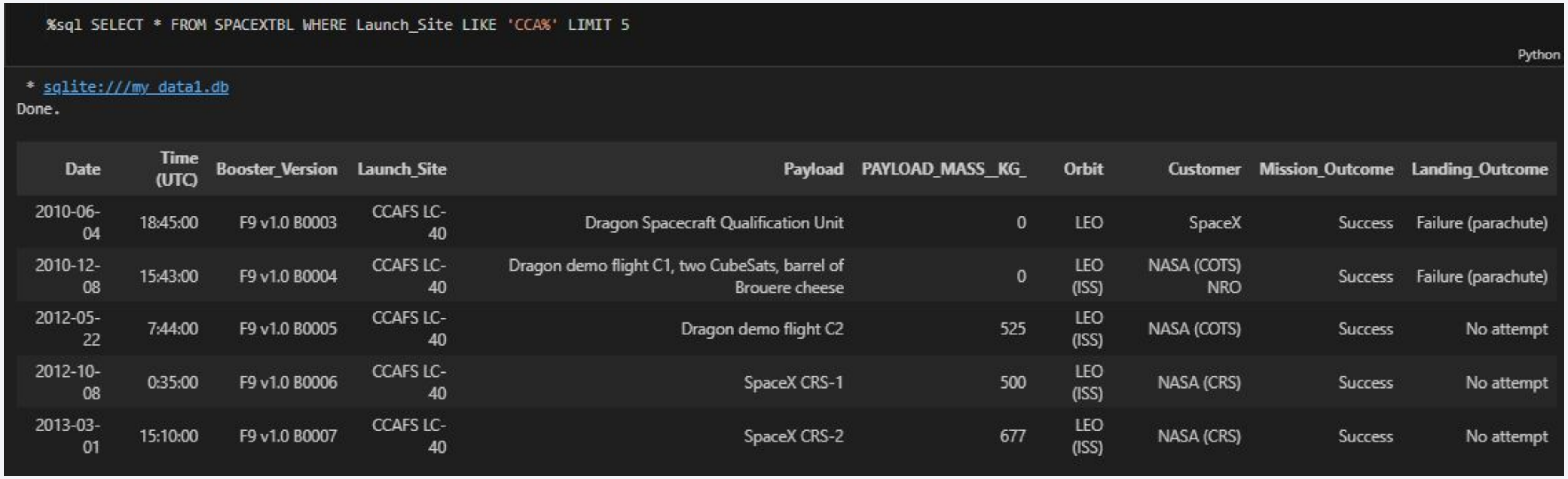
- **All Launch Sites = ['CCAFS LC-40' 'VAFB SLC-4E' 'KSC LC-39A' 'CCAFS SLC-40']**

- The unique() function retrieves **distinct values** from the 'Launch Site' column.
- This helps us understand **how many different sites** SpaceX has used for launches in the dataset.

Each site corresponds to a **physical location** SpaceX used to launch rockets, such as:

- **CCAFS LC-40** – Cape Canaveral Air Force Station
- **VAFB SLC-4E** – Vandenberg Air Force Base
- **KSC LC-39A** – Kennedy Space Center

Launch Site Names Begin with 'CCA'



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

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

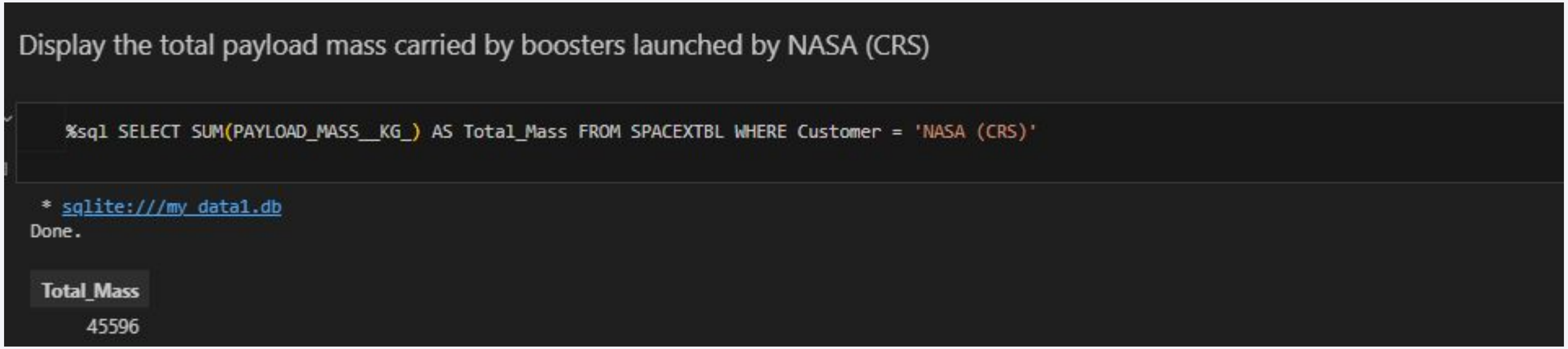
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

By filtering for "CCA"-based launch sites, we can **analyze launch patterns**, **success rates**, and **payload capabilities** specific to that site.

It's also useful for location-based analysis like:

- How successful were launches from CCA compared to VAFB (California)?
- Which orbits are most common for CCA?
- Are payloads heavier on average from this site?

Total Payload Mass



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

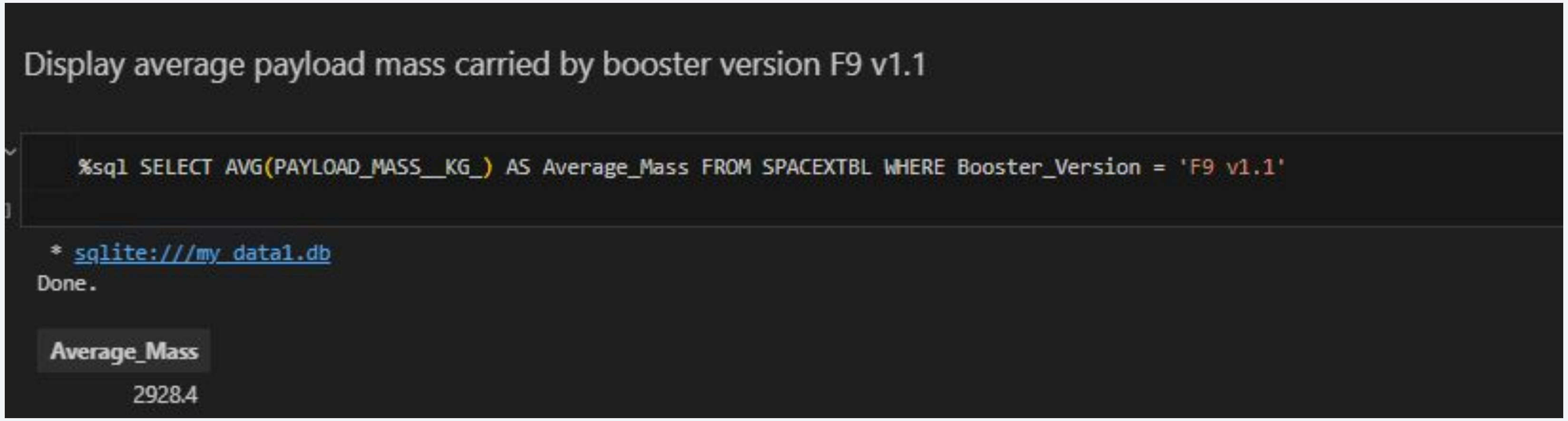
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

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

Total_Mass
45596

- **Total payload mass : 45596 kg**
- **Measuring impact of partnerships:**
  It gives insight into how much NASA has relied on SpaceX to transport materials, satellites, and experiments to space.
- **Understanding launch priorities:**
  ISS missions require precision and safety — and knowing that these made up a significant payload share adds confidence to SpaceX’s reliability.
- **Payload capacity insights:**
  By analyzing total weights delivered, we understand how much SpaceX's boosters are trusted with heavy loads for scientific and humanitarian missions.

Average Payload Mass by F9 v1.1



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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Mass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

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

Average_Mass
2928.4

Average_Mass : 2928.4 kg

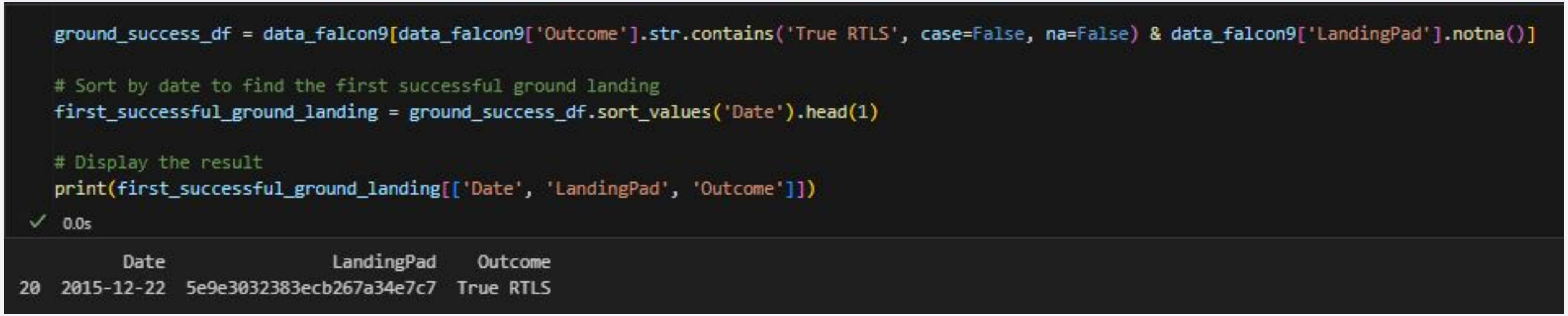
Why this matters:
The **F9 v1.1 (Falcon 9 version 1.1)** was a major upgrade over earlier Falcon 9 models. It had **enhanced engines, longer fuel tanks**, and better overall performance.

What we're doing:
We're:

1. Filtering the dataset to only include missions launched using "F9 v1.1".
2. Calculating the **mean** of the PayloadMass column in that filtered dataset.

Why it's useful:
This average tells us how much cargo this booster model typically carried into space. It's useful for comparing different booster versions and tracking SpaceX's engineering improvements over time.

First Successful Ground Landing Date



```
ground_success_df = data_falcon9[data_falcon9['Outcome'].str.contains('True RTLS', case=False, na=False) & data_falcon9['LandingPad'].notna()]

# Sort by date to find the first successful ground landing
first_successful_ground_landing = ground_success_df.sort_values('Date').head(1)

# Display the result
print(first_successful_ground_landing[['Date', 'LandingPad', 'Outcome']])
```

✓ 0.0s

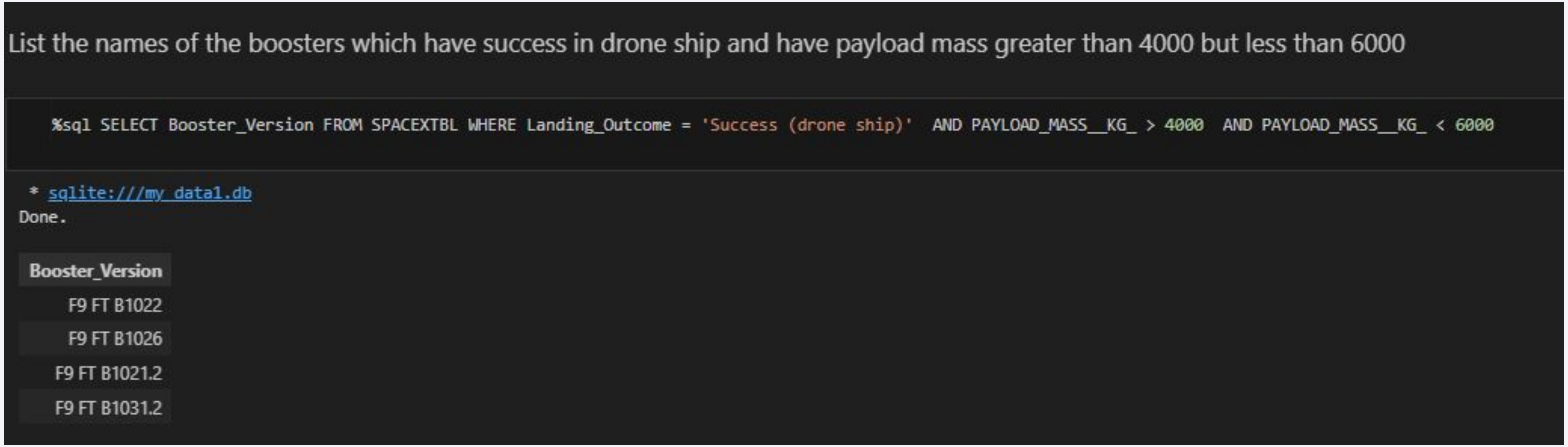
```
          Date                LandingPad    Outcome
20  2015-12-22  5e9e3032383ecb267a34e7c7  True RTLS
```

This date—**December 22, 2015**—marks a **historic breakthrough** for spaceflight:

It was the **first time SpaceX landed a Falcon 9 booster back on solid ground**, rather than letting it fall into the ocean. This proved reusability was not just a dream but a working strategy, reducing launch costs and boosting mission frequency.

It's a turning point in space tech, comparable to when the Wright brothers first landed and re-used a plane.

Successful Drone Ship Landing with Payload between 4000 and 6000



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

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

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

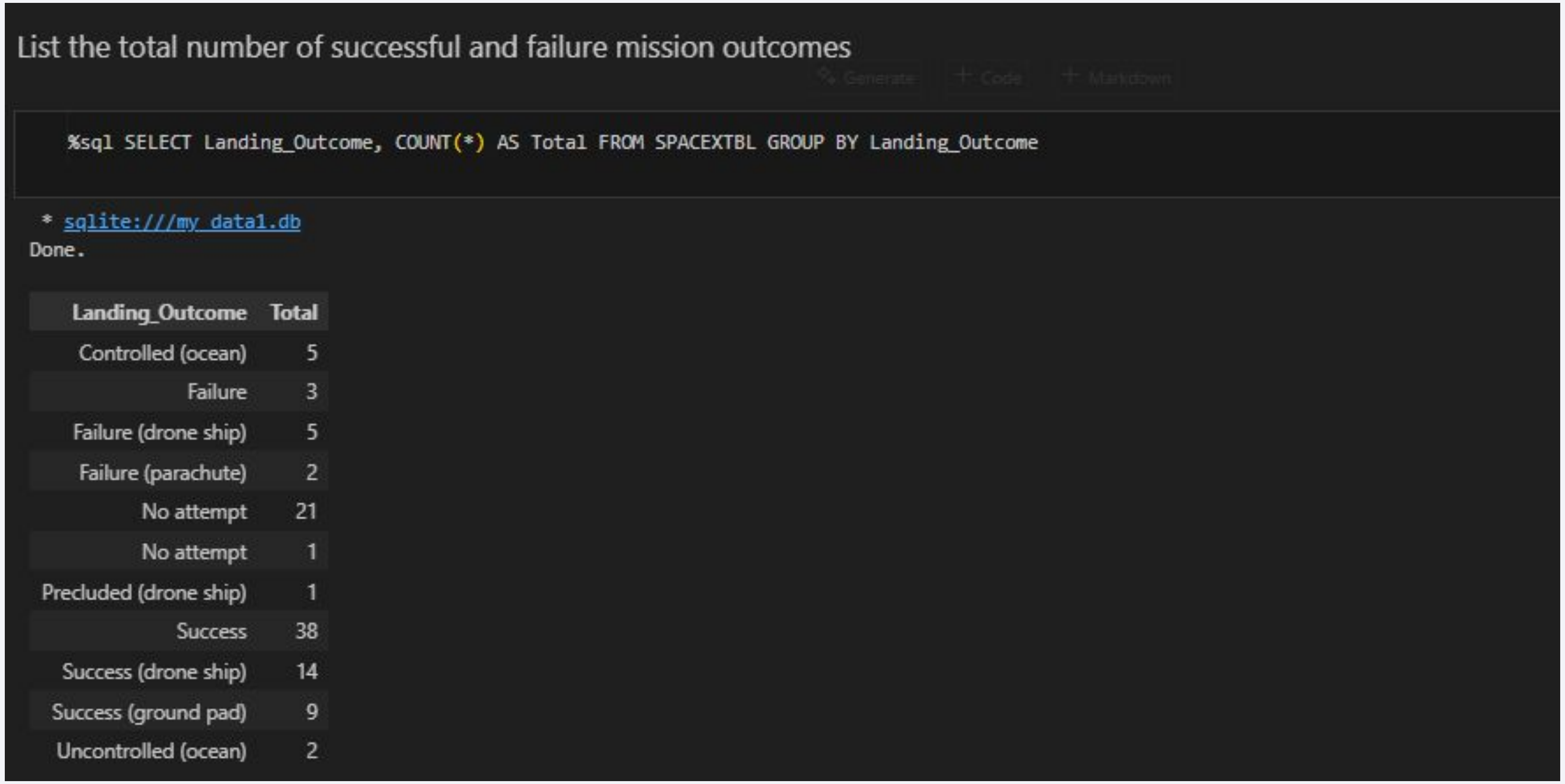
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

This query retrieves the **Booster_Version** of all Falcon 9 launches from the SPACEXTBL table where:

- The **landing was successful** on a **drone ship**.
- The **payload mass** was between **4000 and 6000 kg**.

This range typically includes **mid-weight satellite deployments**, often for communication or GPS purposes.

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

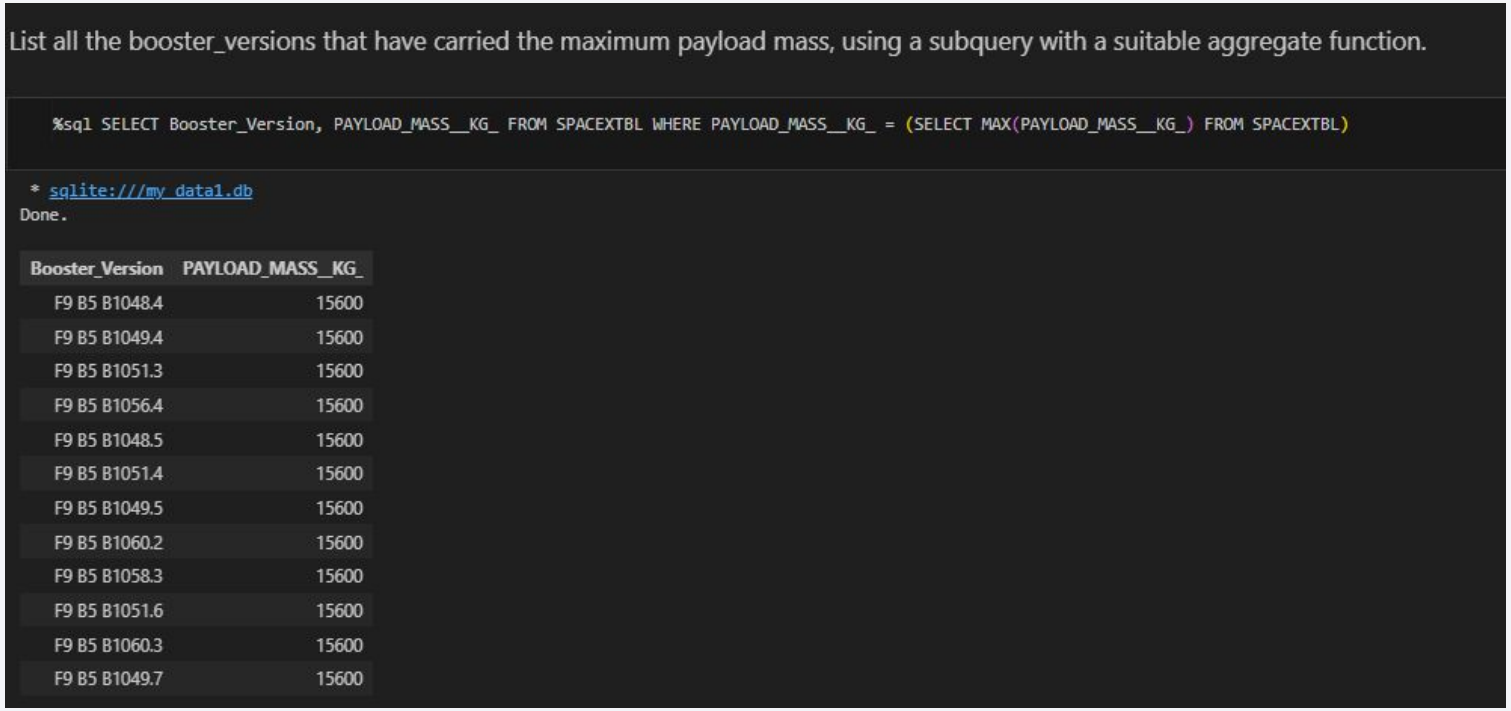
```
%sql SELECT Landing_Outcome, COUNT(*) AS Total FROM SPACEXTBL GROUP BY Landing_Outcome
```

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

Landing_Outcome	Total
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

- Helps assess **SpaceX's performance** over time.
- A high count of "Success" indicates **mission reliability**, while failure counts help identify **risk patterns or areas to improve**.

Boosters Carried Maximum Payload



List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

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

```
 * 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

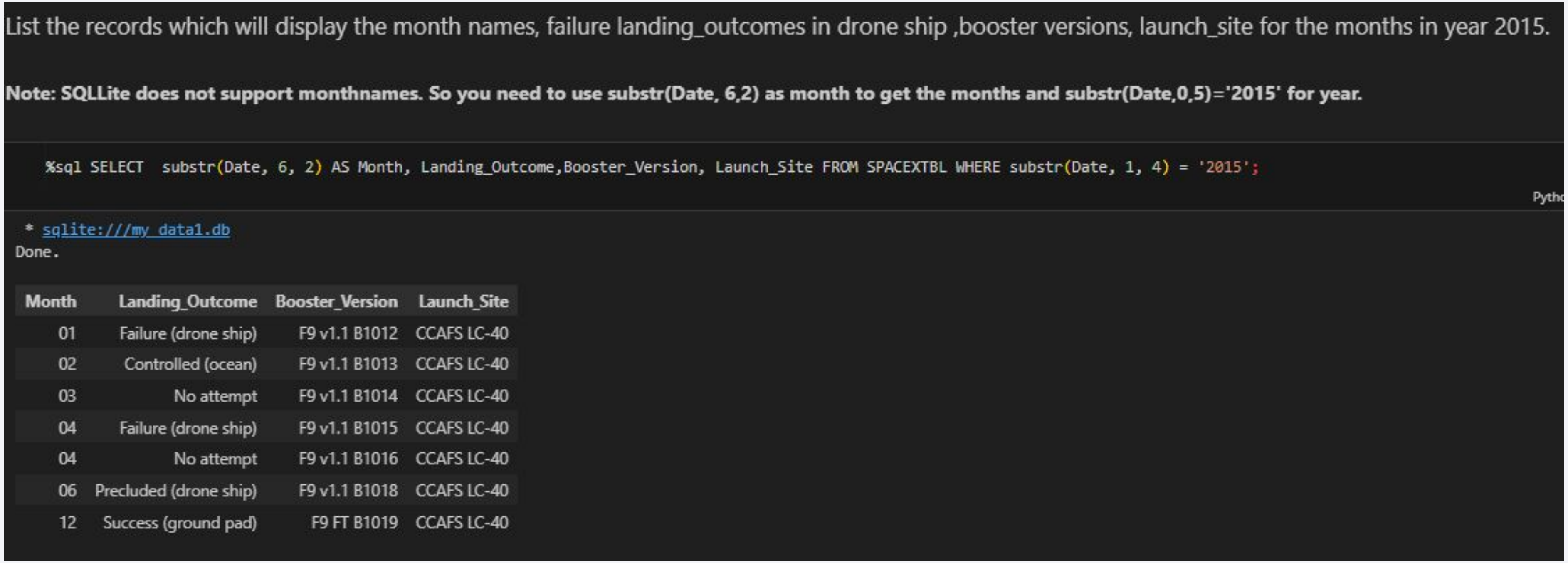
- The **inner query**:
  SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL

  fetches the **maximum payload mass** ever carried.

The **outer query**:

- Selects the **Booster Version** and its corresponding **Payload Mass** from all missions that match this **maximum payload**.
- If more than one booster carried the same maximum payload, it will list them all.

2015 Launch Records



List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

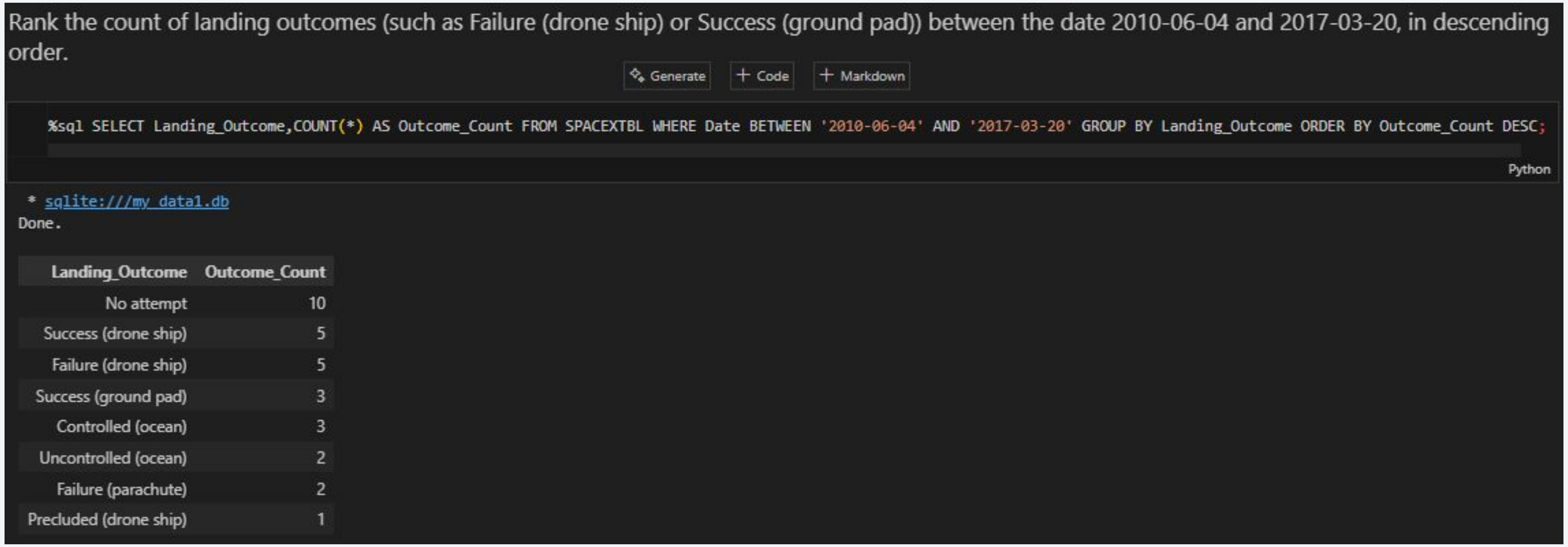
```
%sql SELECT  substr(Date, 6, 2) AS Month, Landing_Outcome,Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date, 1, 4) = '2015';
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

- Extracts the **month** from the Date column by taking characters 6 and 7 (substr(Date, 6, 2)), labeling this as Month.
- Selects the Landing_Outcome, Booster_Version, and Launch_Site columns.
- Filters the records to only include launches that happened in the year **2015** (substr(Date, 1, 4) = '2015').

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



Rank the 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.

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

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

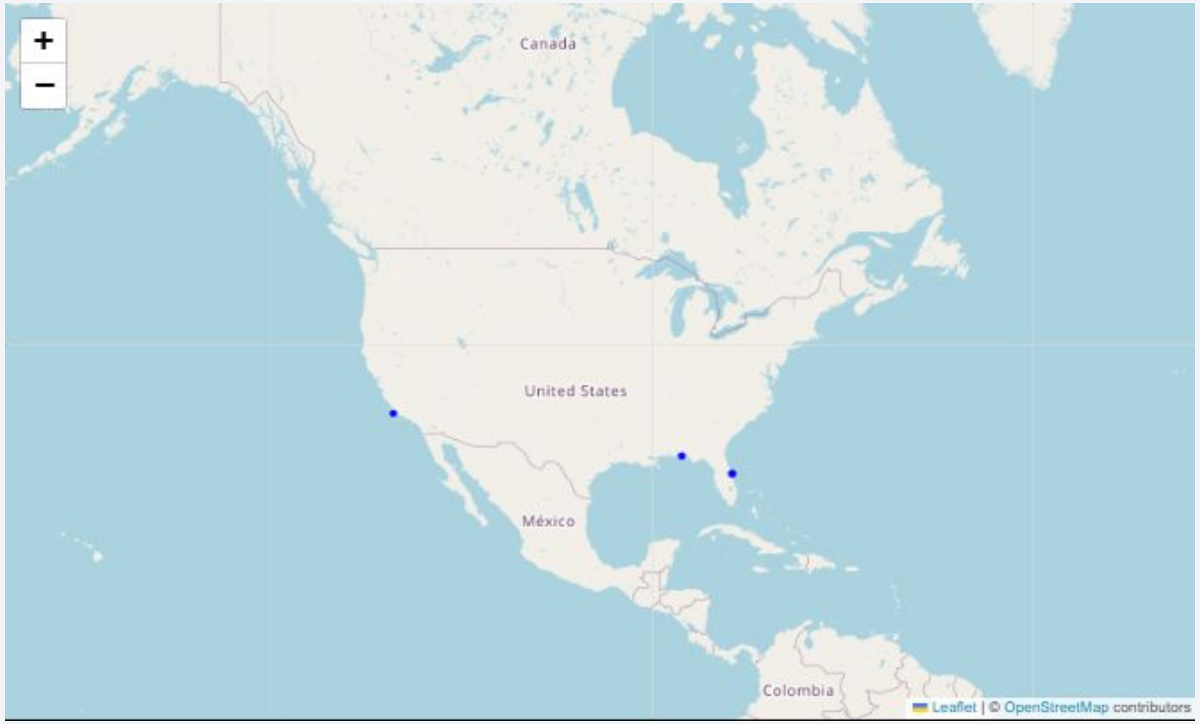
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- Counts how many times each **Landing_Outcome** appears in the data.
- Considers only launches between **June 4, 2010** (SpaceX's first launch) and **March 20, 2017**.
- Groups the data by Landing_Outcome to get counts per outcome.
- Orders the result by the count in descending order — so the most frequent outcomes appear at the top.

Section 3

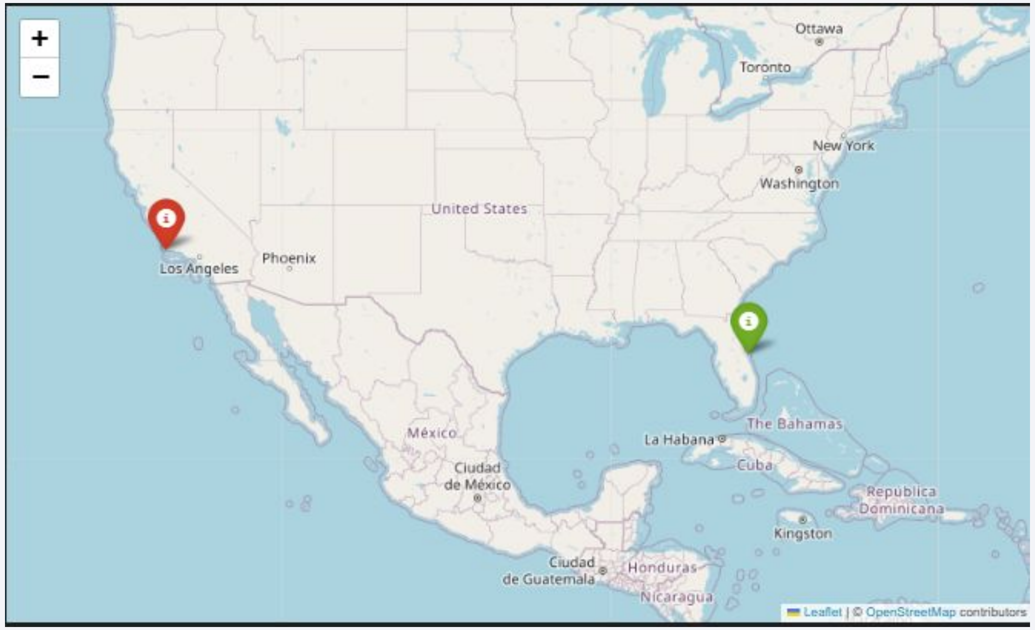
Launch Sites Proximities Analysis

All launch sites' locations markers Globally



- **Concentration on U.S. coasts:** Most launch pads are coastal, optimizing rocket safety and access to desired orbits.
- **Florida sites (CCAFS and KSC):** Primary launch points for many missions due to proximity to the equator (more efficient orbits).
- **California (VAFB):** Used mainly for polar orbits because of its latitude and coastal position.
- **Boca Chica, Texas:** Emerging site for Starship launches, showing SpaceX's expanding footprint.

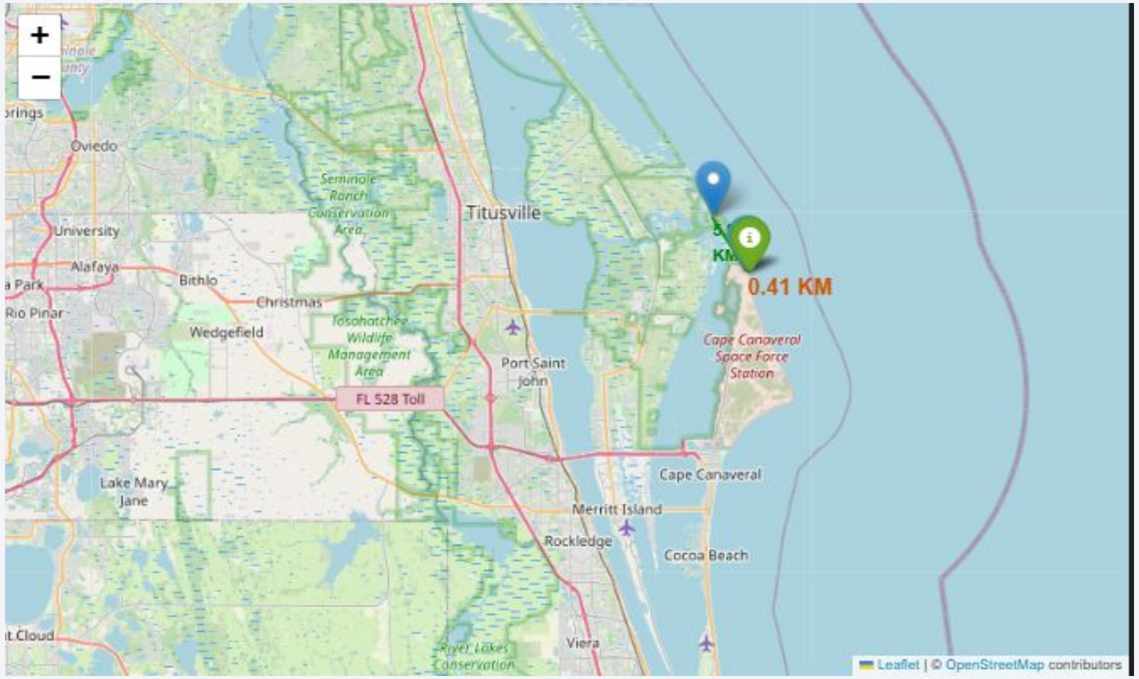
Color- Labelled launch outcomes



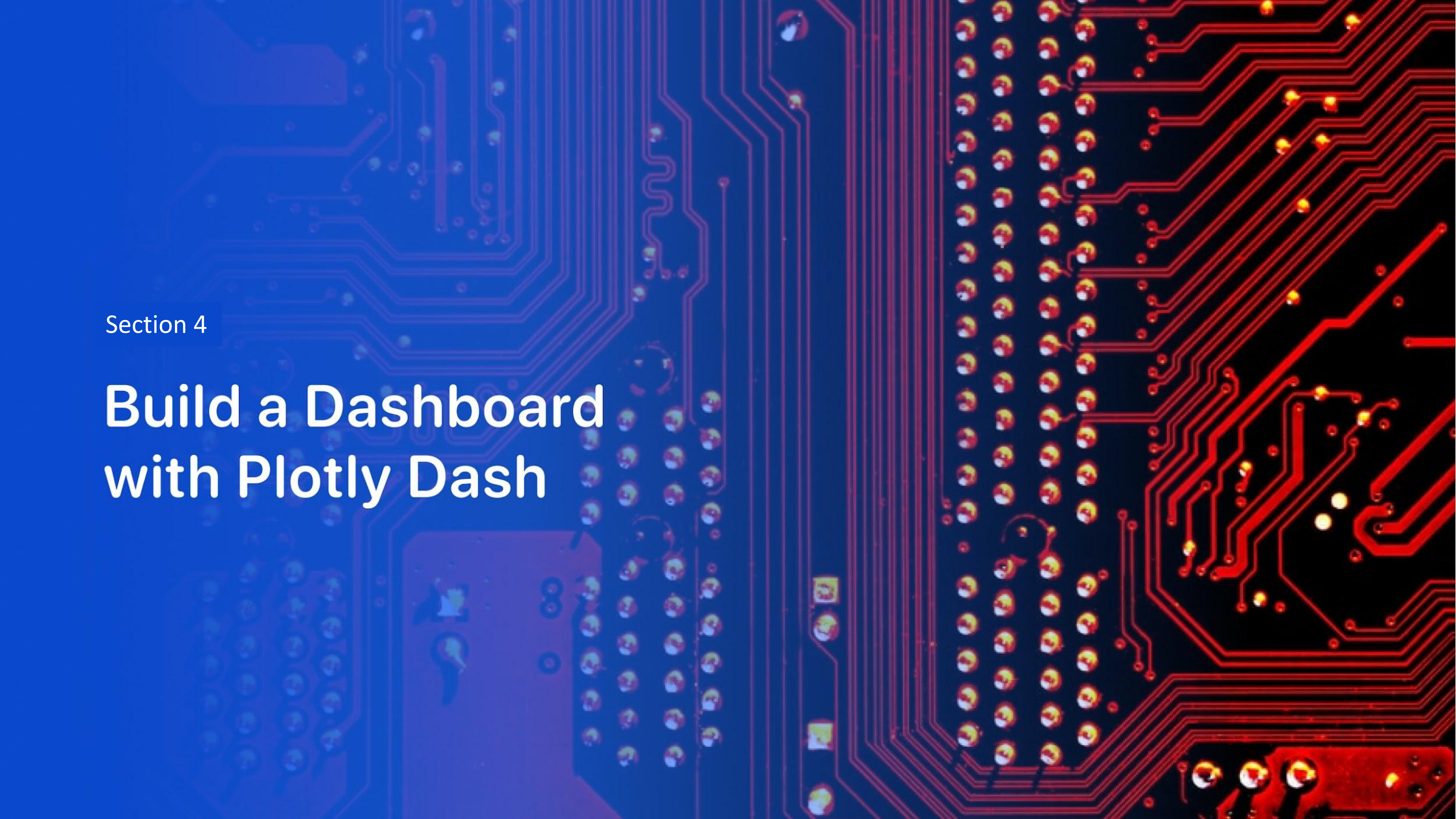
This visualization is more than just pretty dots on a map. It’s a strategic tool that highlights:

- SpaceX’s operational footprint.
- Performance trends across locations.
- Helps engineers and analysts target improvements where failures cluster.
- And gives stakeholders a clear, intuitive picture of launch success geographically.

Launch site distance with closes proximity



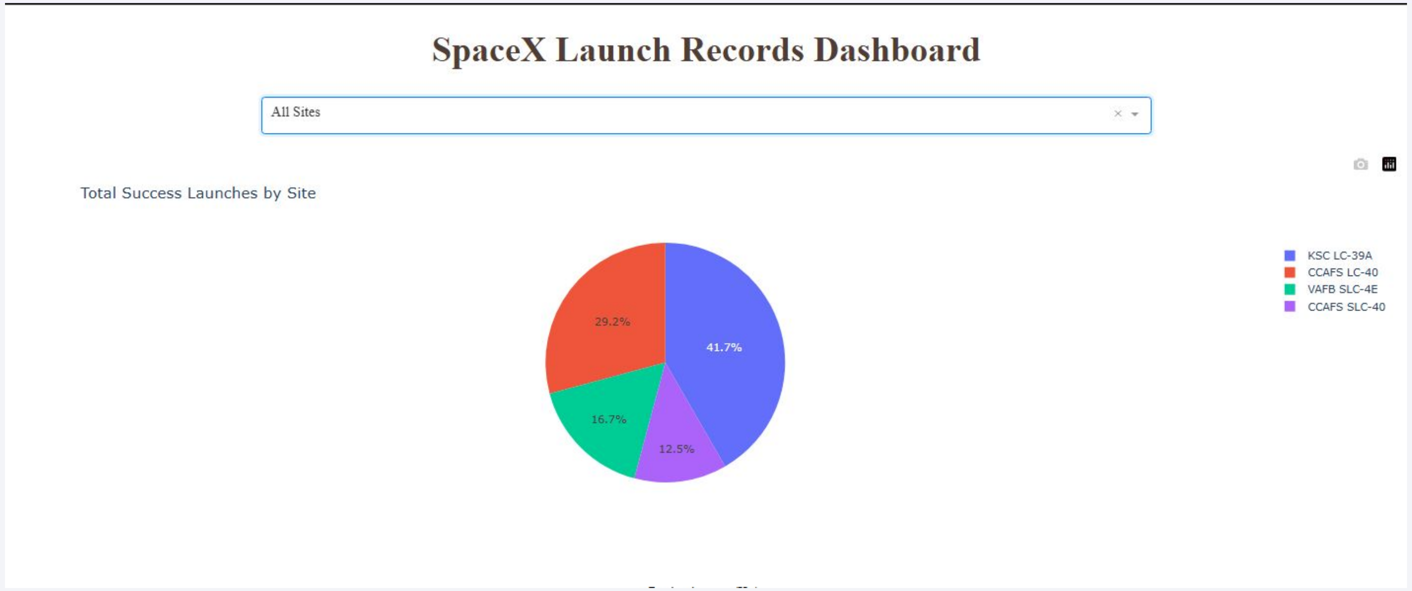
Safety Analysis: Knowing distances to infrastructure and natural features helps in emergency planning and hazard analysis.
Logistics Planning: Understanding how close transport routes are can optimize supply chain and personnel movement.
Environmental Impact: Proximity to coastline may affect launch trajectory planning and risk of debris over water.
Stakeholder Communication: Clear, annotated maps give project managers and regulators transparent insight into launch site context.



Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites



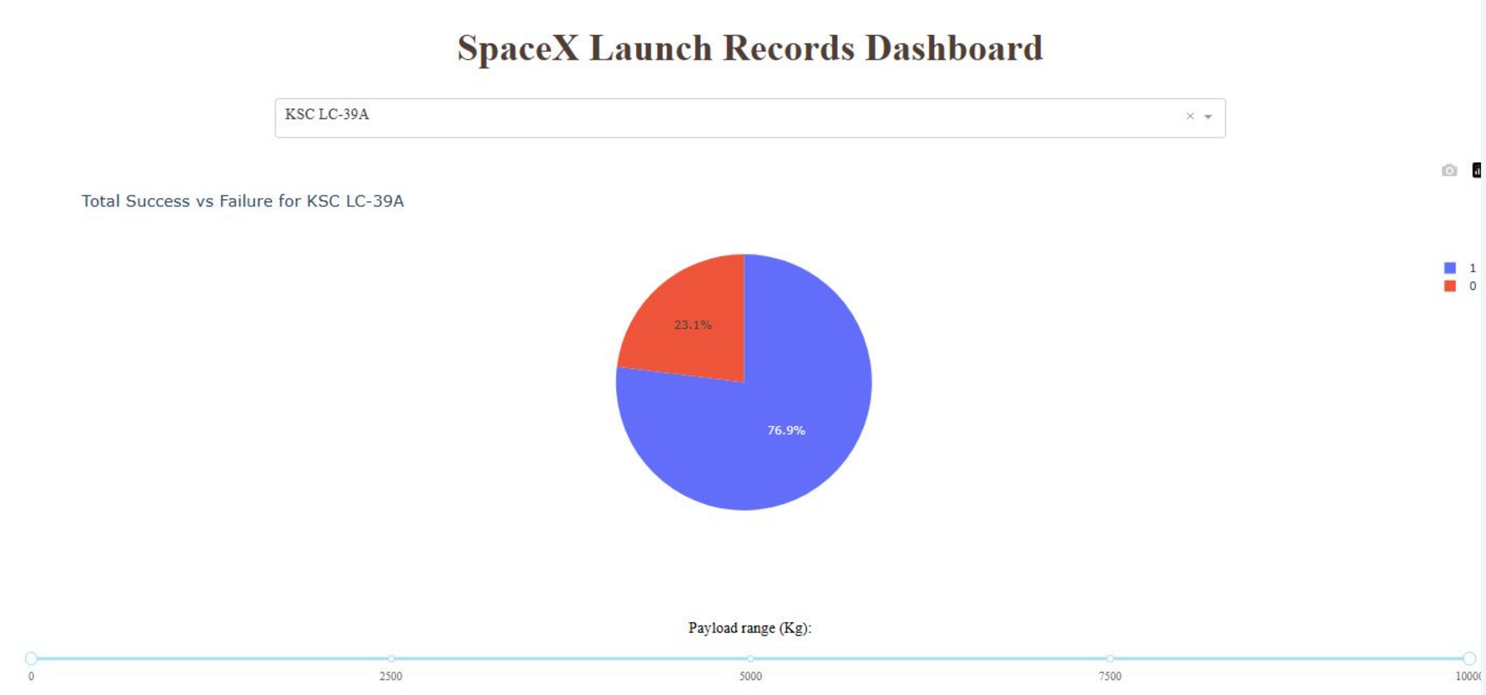
SpaceX Launch Records Dashboard

All Sites

Total Success Launches by Site

- **KSC LC-39A** dominates with over 40% of the successful launches, indicating it’s the primary hub for SpaceX’s successful missions.
- **CCAFS LC-40** is the next most active site with almost 30%, showing strong operational presence.
- **VAFB SLC-4E** and **CCAFS SLC-40** are less frequent but still contribute significantly.
- This distribution highlights the geographic and operational focus of SpaceX’s launch activities.

Piechart for the launch site with highest launch success ratio

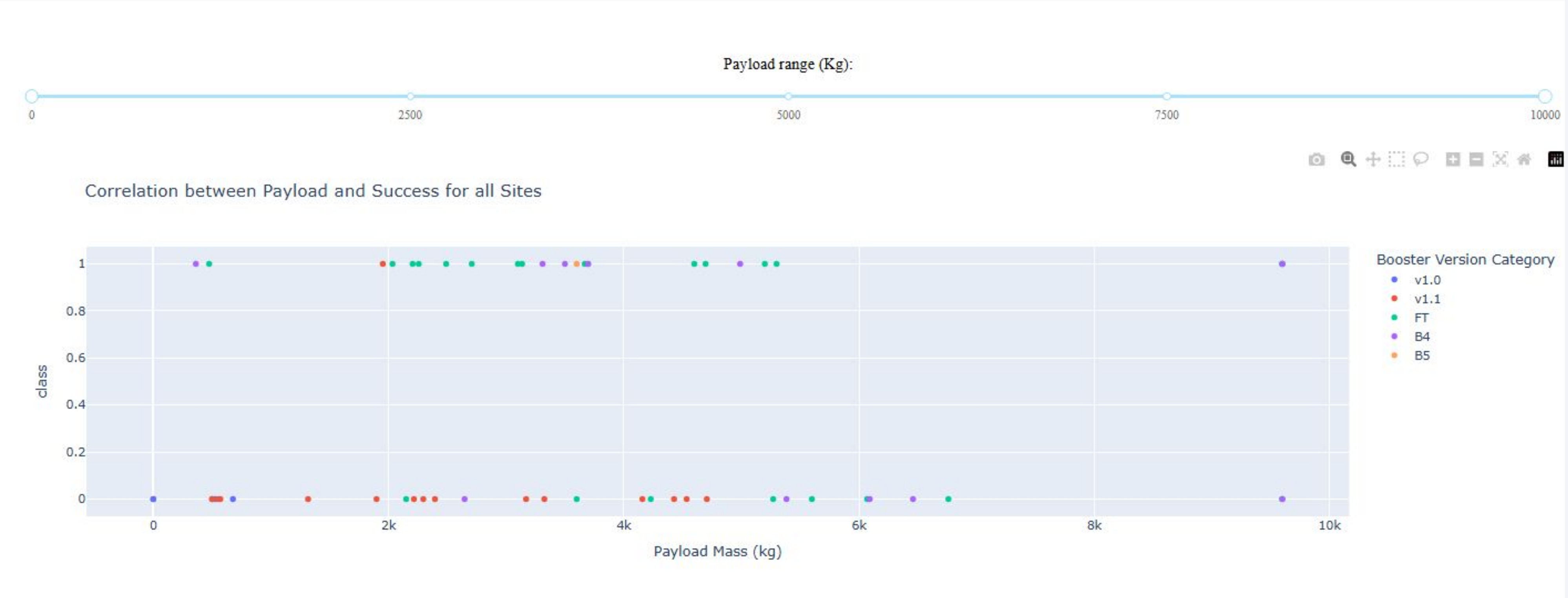


SpaceX Launch Records Dashboard

KSC LC-39A

1. **High Success Ratio:**
   - With **76.9% success**, KSC LC-39A leads in reliability. This site is known for handling high-profile missions including Falcon Heavy and Crew Dragon, which aligns with its high success stats.
2. **Failure Rate:**
   - 23.1% failure isn't negligible and could be associated with earlier experimental missions or weather-related issues. It's useful for historical comparison and improvement tracking.
3. **Operational Significance:**
   - The high success ratio supports KSC LC-39A's reputation as SpaceX's flagship pad, often used for missions with NASA partnerships or human-rated launches.

Payload vs. Launch Outcome scatter plot for all sites



Booster FT and B5 are the most **reliable across all payloads**.

Payloads in the **2000–6000 kg** range are the **sweet spot** for success.

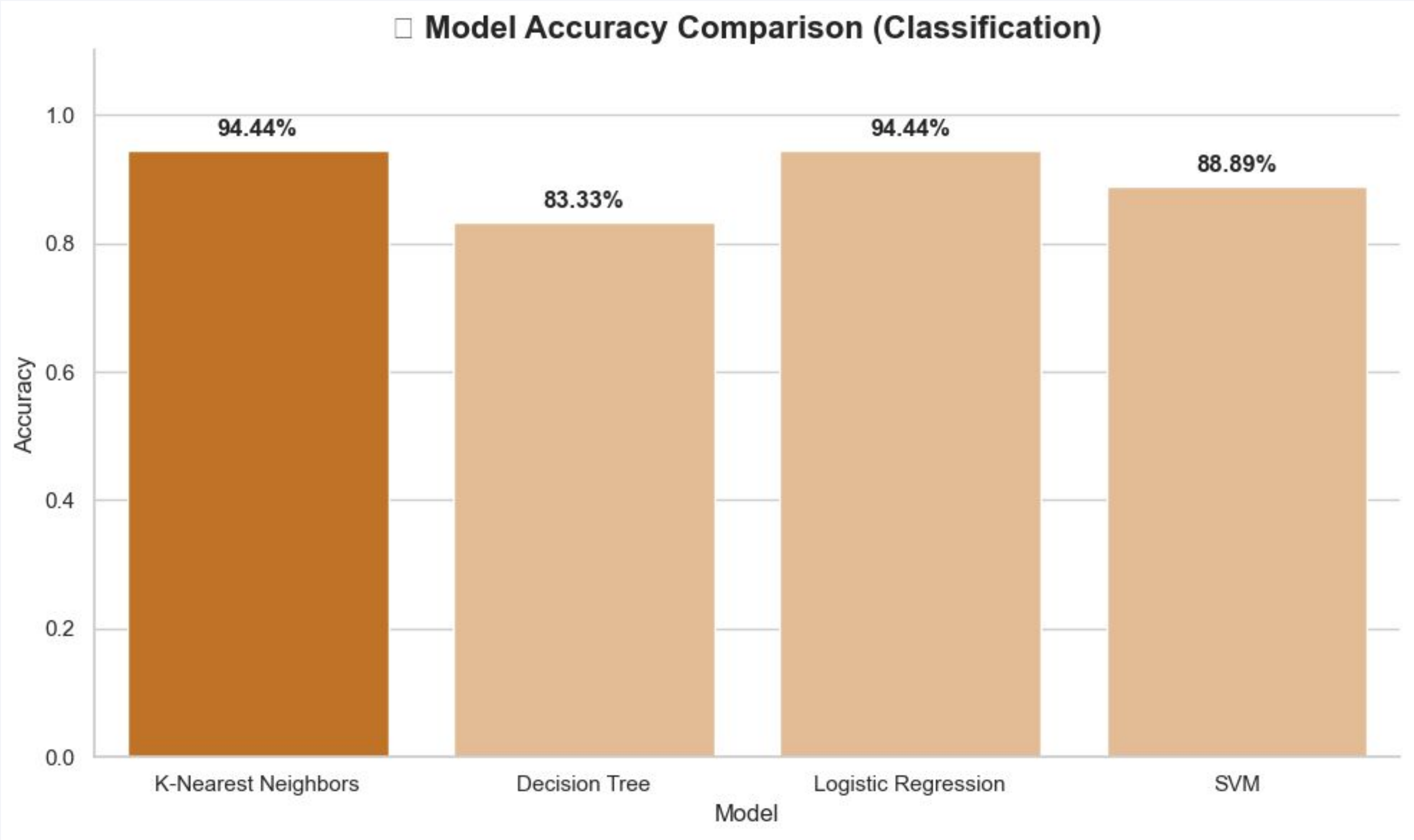
Early boosters like v1.0 and v1.1 were more failure-prone, especially with light and heavy payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

K Nearest Neighbors and Logistic Regression shows an equal accuracy of 94.44%



Confusion Matrix

Interpretation

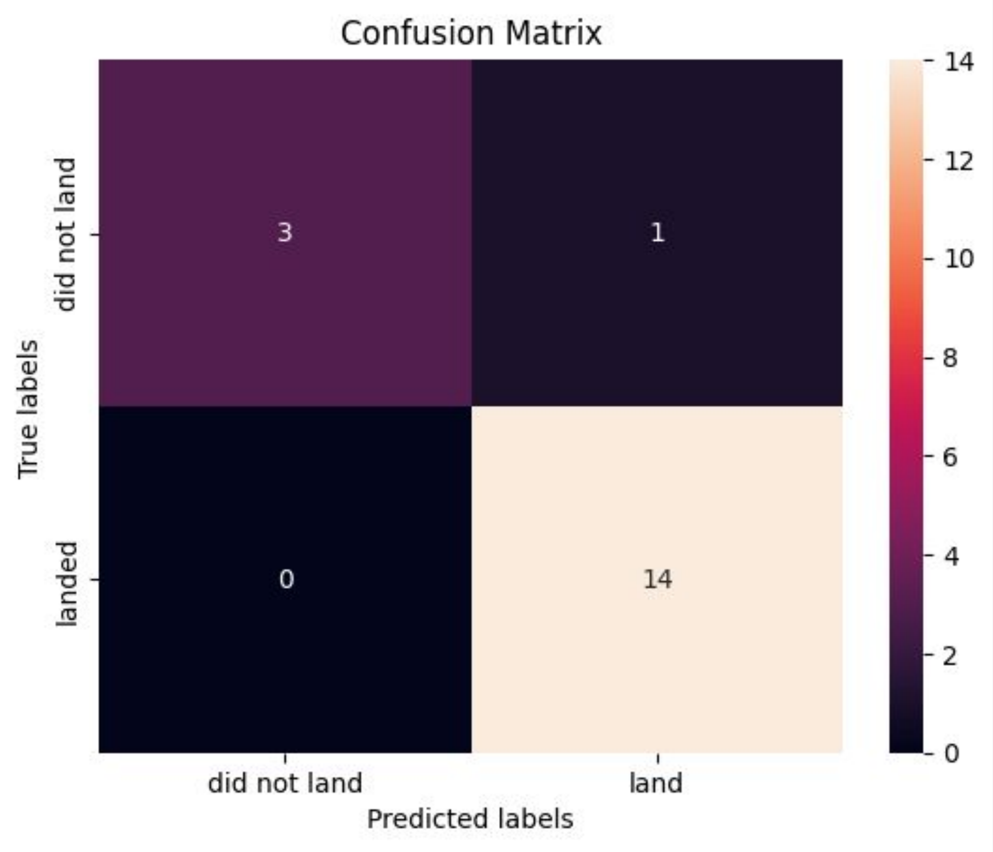
- **True Positives (TP = 14)**: The model correctly predicted 14 landings.
- **True Negatives (TN = 3)**: It correctly identified 3 cases that did not land.
- **False Positives (FP = 1)**: One case was predicted as "landed" but didn’t.
- **False Negatives (FN = 0)**: No landings were missed — nice!

What this means:

- The model is **highly reliable in detecting successful landings** (100% sensitivity or recall for “landed”).

- Only **1 mistake** was made — falsely predicting a failed landing as successful.

- **Precision, Recall, F1-score**, and **overall accuracy** would all be high, confirming strong performance.



Conclusions

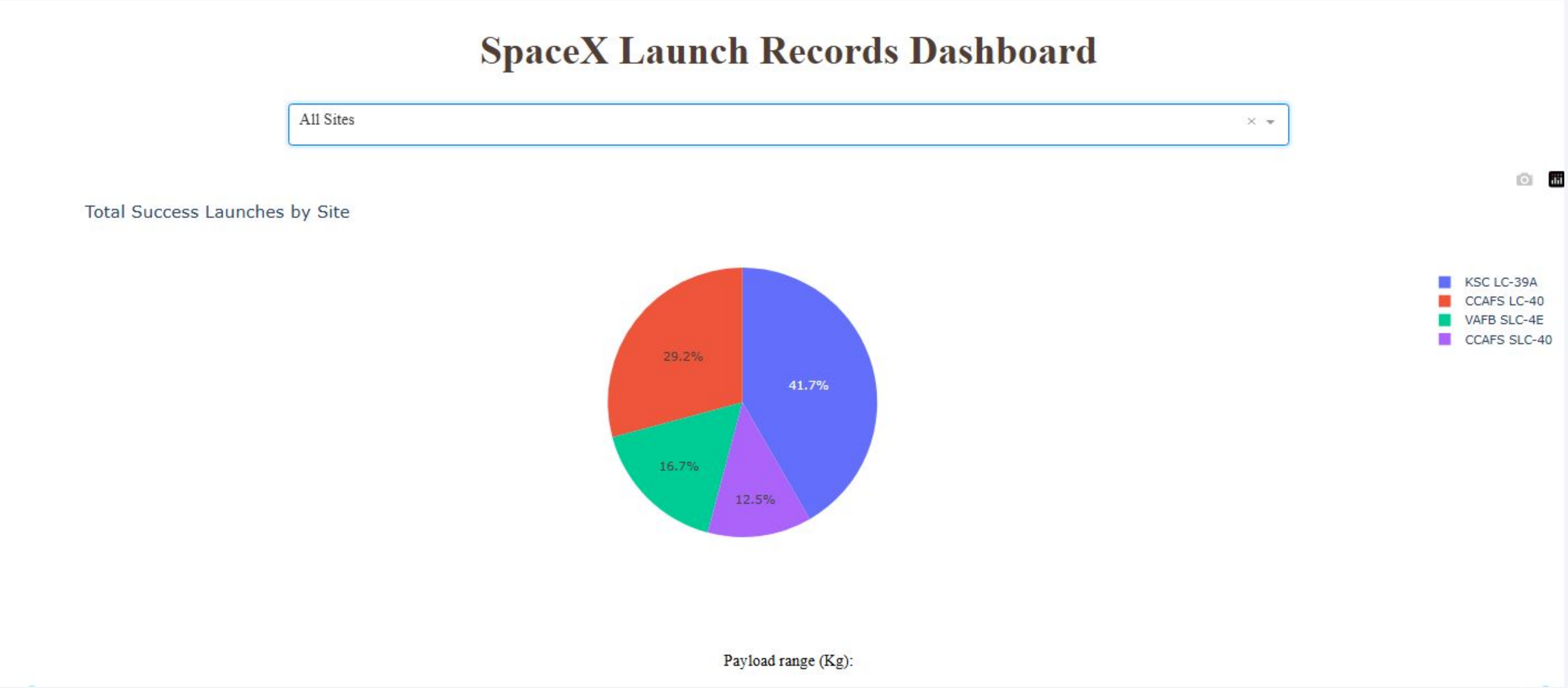
- **Multiple Classification Models Built**
 Four supervised learning models were implemented — K-Nearest Neighbors, Decision Tree, Logistic Regression, and SVM — to predict SpaceX launch outcomes based on mission features.

- **KNN and Logistic Regression Performed Best**
 Both models achieved the highest accuracy of **94.44%**, demonstrating strong predictive capability, especially in correctly identifying successful launches.

- **Confusion Matrix Validated Model Precision**
 The confusion matrix of the best model showed **minimal misclassifications**, especially **no false negatives**, highlighting excellent model sensitivity for detecting successful landings.

- **Model Selection Backed by Performance Metrics**
 The final model selection was made based on **accuracy, interpretability**, and **confusion matrix insights**, ensuring both reliable predictions and clarity for decision-making.

Appendix



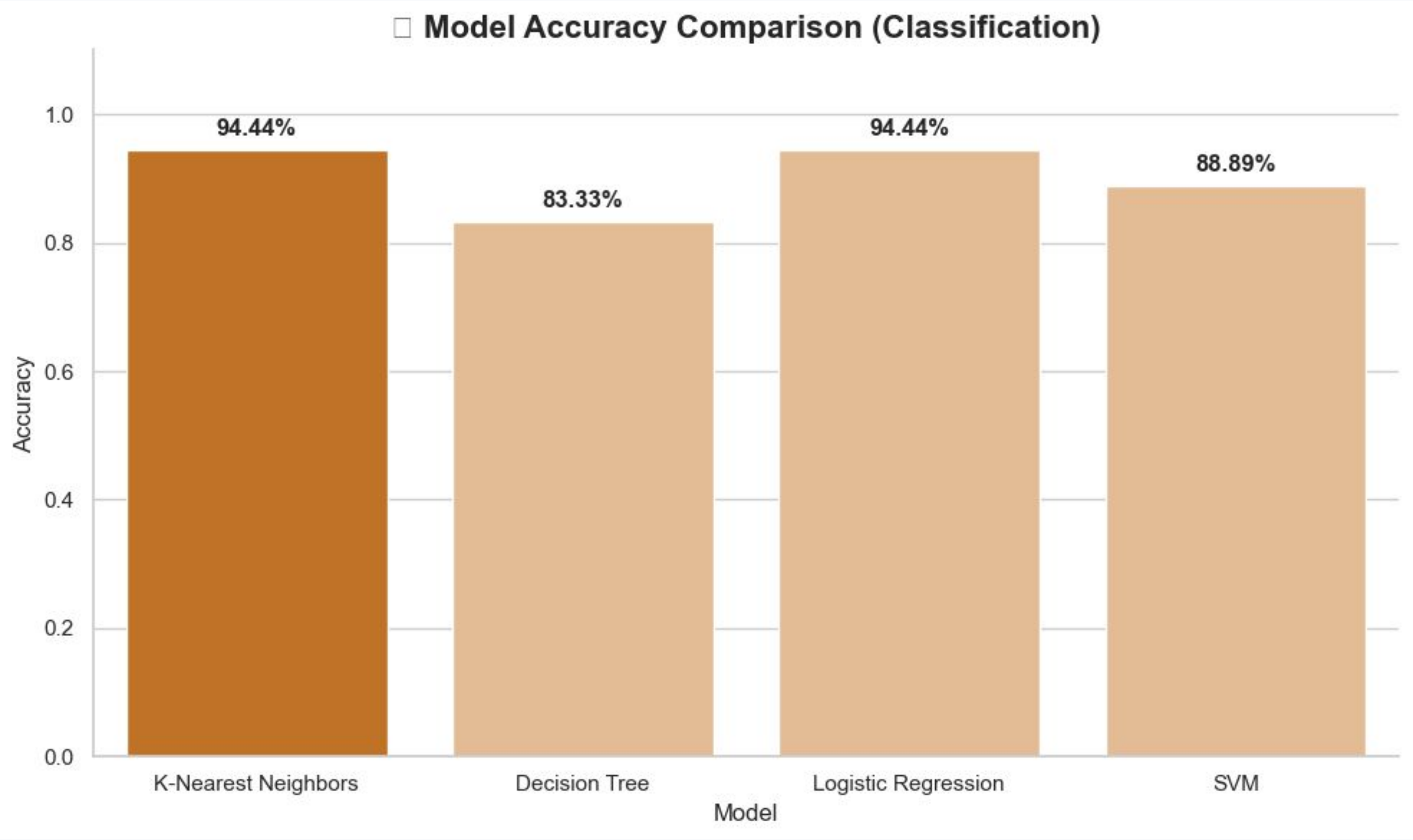
SpaceX Launch Records Dashboard

All Sites

Total Success Launches by Site

Payload range (Kg):

Appendix


□ Model Accuracy Comparison (Classification)
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
94.44%
83.33%
94.44%
88.89%
K-Nearest Neighbors
Decision Tree
Logistic Regression
SVM
Model

Thank you!