IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Hery Wibowo
July 24, 2024

# Outline

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

# Executive Summary


Methodologies


Results

**Data Collection SpaceX API**

Dataset: Falcon 9 launches, exported to dataset_part_1.csv

**Data Collection Scraping**

Extracted Columns, view of Falcon 9 and Falcon Heavy launch data from Wikipedia

**Data Wrangling**

Cleaned dataset saved with new Class column to dataset_part_2.csv

**EDA with SQL**

Get insight from queries SQL

**EDA with Data Visualization**

Get insight from visualization charts (Scatter, Barchart, Line)

**Build an Interactive Map with Folium**

Map display with the following insights: Launch Sites, Success Rates, Proximity Insights

**Build a Dashboard with Plotly Dash**

Dashboard display with piechart and scatter diagrams with insights: Site Analysis, Payload Analysis, Booster Analysis

**Predictive Analysis (Classification)**

Models: Best hyperparameters and performance metrics identified (SVM, Classification Trees, Logistic Regression, KNN). Conclusion: Select the model with the best overall performance

# Introduction

## Objective

- Predict if the Falcon 9 first stage will land successfully

## Context

- SpaceX's Falcon 9 rocket launches cost $62 million, while other providers charge $165 million or more per launch
- Savings are due to the reusability of the Falcon 9 first stage

## Purpose

- If we can predict whether the first stage will land successfully, we can estimate the cost of a launch
- This prediction can help other companies in their bid against SpaceX for rocket launches

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

## Data Collection API

- Fetch and parse SpaceX launch data from a JSON URL, filter relevant rows, and extract key features into a Pandas DataFrame. This DataFrame, focusing on Falcon 9 launches, includes payload details and core performance metrics.

## Data Collection Scraping

- Identify and access the Wikipedia page. Then, use BeautifulSoup to parse the HTML and locate the target table. Extract and clean the data, and convert it into a Pandas DataFrame for analysis

## Data Wrangling

- The data collection process involved defining the goal, sourcing a SpaceX dataset from a public URL, loading it with pandas, and performing initial exploration to identify missing values and data types. The data was cleaned by addressing missing values and converting outcome labels into binary success/failure indicators. Finally, the cleaned dataset was exported for analysis

# Data Collection – SpaceX API

**Key Steps:**

1. **Identify API Endpoints**: Determine relevant SpaceX API endpoints (e.g., `https://api.spacexdata.com/v4/launches`).
2. **HTTP Requests**: Use `requests` library to make GET requests to the API.
3. **Data Extraction**: Parse JSON responses to extract required data.
4. **Data Cleaning**: Clean and preprocess the data (e.g., handle missing values).
5. **Data Storage**: Save cleaned data to a CSV file for analysis.

```
START
   |
   v
Identify API Endpoints
   |
   v
Make GET Requests to API
   |
   v
Extract Data from JSON Responses
   |
   v
Clean and Preprocess Data
   |
   v
Store Data for Analysis
   |
   v
END
```

[Link Git Hub Data Collection-SpaceX API]

# Data Collection - Scraping

**Key Steps:**

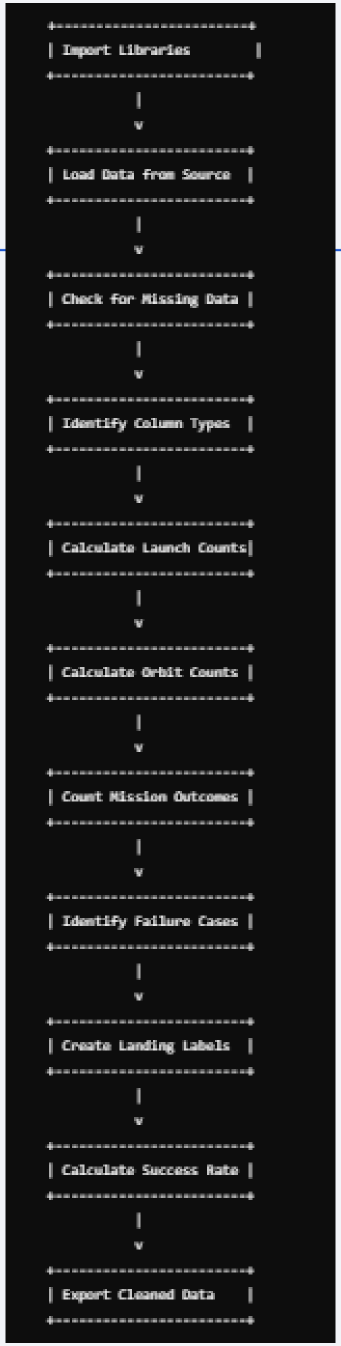
1. **Request HTML Page:** Use `requests.get()` to fetch the HTML from the Wikipedia URL.
2. **Parse HTML:** Create a `BeautifulSoup` object from the HTML content.
3. **Extract Table Data:**
    i. Locate and parse the target table.
    ii. Extract column names and rows.
4. **Store Data:** Store extracted data in a dictionary and convert it to a Pandas DataFrame.
5. **Export Data:** Save the DataFrame as a CSV file.

```
A[Start] --> B[Request HTML Page]
B --> C[Create BeautifulSoup Object]
C --> D[Extract Table Data]
D --> E[Store Data in DataFrame]
E --> F[Export to CSV]
F --> G[End]
```

[Link Git Hub Data Collection-Scraping]

# Data Wrangling

1. **Import Libraries & Load Data:** Load data using pandas.
2. **Check for Missing Values:** Find and list missing values in the dataset.
3. **Column Types:** Identify which columns are numerical and which are categorical.
4. **Launch Counts:** Count how many launches were done at each site.
5. **Orbit Counts:** Count how often each type of orbit was used.
6. **Outcome Counts:** Count different landing outcomes and identify failures.
7. **Create Labels:** Create a new column Class with 1 for successful landings and 0 for failures.
8. **Success Rate:** Calculate the proportion of successful landings.
9. **Export Data:** Save the cleaned data to a CSV file.




[Link Git Hub Data Wrangling]

# EDA with Data Visualization

1. **Flight Number vs. Payload Mass**
   - **Chart Type**: Scatter plot, **Purpose**: To show how the number of flights and payload mass affect landing success.
2. **Flight Number vs. Launch Site**
   - **Chart Type**: Scatter plot, **Purpose**: To see how the number of flights varies by launch site and its impact on success.
3. **Payload Mass vs. Launch Site**
   - **Chart Type**: Scatter plot, **Purpose**: To understand how payload mass affects landing success at different launch sites.
4. **Success Rate by Orbit Type**
   - **Chart Type**: Bar chart, **Purpose**: To compare the success rates of different orbit types.
5. **Flight Number vs. Orbit Type**
   - **Chart Type**: Scatter plot, **Purpose**: To see if the number of flights affects success rates for different orbits.
6. **Payload Mass vs. Orbit Type**
   - **Chart Type**: Scatter plot, **Purpose**: To check how payload mass impacts success rates across different orbits.
7. **Yearly Launch Success Trend**
   - **Chart Type**: Line chart, **Purpose**: To track changes in the success rate over the years.



Link Git Hub EDA Data Visualization

# EDA with SQL

1. **TAKS 1**: *%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;*
2. **TASK 2**: *%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;*
3. **TASK 3**: *%sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';*
4. **TASK 4**: *%sql SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';*
5. **TASK 5**: *%sql SELECT MIN("Date") AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';*
6. **TASK 6**: *%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass__kg_" > 4000 AND "Payload_Mass__kg_" < 6000;*
7. **TASK 7**: *%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY "Mission_Outcome";*
8. **TASK 8**: *%sql SELECT "Booster_Version", "Payload_Mass__kg_" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE);*
9. **TASK 9**: *%sql SELECT CASE substr("Date", 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April' WHEN '05' THEN 'May' WHEN '06' THEN 'June' WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September' WHEN '10' THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December' END AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr("Date", 1, 4) = '2015';*
10. **TASK10**: *%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;*



[Link Git Hub EDA with SQL]

# Build an Interactive Map with Folium

**Summary of Map Objects Added to the Folium Map**

- **Markers:**
  - Launch Sites: Name displayed.
  - Launch Outcomes: Color-coded (green for success, red for failure).
  - Proximity Points: Closest coastline, city, railway, and highway, with distances.
- **Circles:**
  - Around Launch Sites: 1000-meter radius for visual context.
- **Lines:**
  - Proximity Lines: Distances to coastlines, cities, railways, highways.
- **Marker Cluster:**
  - Launch Outcomes: Groups markers to reduce clutter.
- **Mouse Position:**
  - Displays latitude and longitude for coordinates.

**Explanation**

- **Markers:** Identify and analyze launch sites and outcomes.
- **Circles:** Highlight and locate launch sites.
- **Lines:** Show distances to key geographical features.
- **Marker Cluster:** Manage multiple markers for readability.
- **Mouse Position:** Helps identify coordinates dynamically.

Link Git Hub Map with Folium

# Build a Dashboard with Plotly Dash

**Summary of Plots/Graphs and Interactions**

**1. Dropdown for Launch Sites:**
- **Purpose:** Filters data by launch site (or all sites).
- **Interaction:** Updates the pie chart and scatter plot based on the selected site.

**2. Pie Chart for Launch Success by Site:**
- **Purpose:** Shows successful launches per site or success rate for a selected site.
- **Interaction:** Changes with site selection from the dropdown.

**3. Range Slider for Payload Mass:**
- **Purpose:** Filters data by payload mass range.
- **Interaction:** Updates the scatter plot to reflect the specified range.

**4. Scatter Plot for Success vs. Payload Mass:**
- **Purpose:** Displays the relationship between payload mass and launch success.
- **Interaction:** Updates based on selected payload range and launch site.

**Explanation**

•**Dropdown:** Allows detailed examination of specific sites.
•**Pie Chart:** Visualizes proportions and success rates clearly.
•**Range Slider:** Helps analyze the impact of different payload masses.
•**Scatter Plot:** Shows correlation between payload mass and success, highlighting trends.

Link Git Hub Dashboard

# Predictive Analysis (Classification)

**Model Development Process:**

**Steps**

1. **Data Collection**: Web scrape Falcon 9 launch data.
2. **Data Cleaning**: Handle missing values and normalize data.
3. **Feature Engineering**: Create new features.
4. **Exploratory Data Analysis**: Visualize data relationships.
5. **Data Splitting**: Split data into training and testing sets.
6. **Model Selection**: Choose algorithms (Logistic Regression, Decision Trees, KNN, SVM).
7. **Model Training**: Train models on the training dataset.
8. **Performance Metrics**: Evaluate with Accuracy.
9. **Baseline Performance**: Establish baseline metrics.
10. **Hyperparameter Tuning**: Optimize parameters with GridSearchCV.
11. **Cross-Validation**: Ensure robustness with k-fold cross-validation.
12. **Feature Selection**: Use RFE for significant features.
13. **Final Model Selection**: Choose the best model.
14. **Model Validation**: Validate on test data.

# Predictive Analysis (Classification)

*Flowchart*

```
[Start]
  |
  v
[Data Collection] --> [Data Cleaning] --> [Feature Engineering] --> [EDA]
  |
  v
[Data Splitting] --> [Model Selection] --> [Model Training]
  |
  v
[Performance Metrics] --> [Baseline Performance]
  |
  v
[Hyperparameter Tuning] --> [Cross-Validation] --> [Feature Selection]
  |
  v
[Final Model Selection] --> [Model Validation]
  |
  v
[End]
```

[Link Git Hub Predictive Analysis (Classification)]

# Results

## EDA

- EDA SQL: Cover unique launch sites, initial 'CCA' launches, NASA CRS payload totals, F9 v1.1 average payload, earliest successful ground pad landing, successful drone ship landings with specific payloads, mission outcome counts, maximum payload boosters, and 2015 failed drone ship landings by month.
- EDA Visualization: The analysis shows that Falcon 9’s landing success improves with higher flight numbers, heavier payloads, and certain orbits, with a general trend of increasing success rates over the years.

## Interactive Analytics

- Mapped launch sites with circles and markers, highlighted success and failure of launches with color-coded markers using MarkerCluster, and calculated distances from launch sites to coastlines, cities, railways, and highways, visualizing these distances with lines on the map.
- Dashboard: Add a site dropdown, a pie chart callback for success rates, a payload range slider, and a scatter plot callback for payload vs. success. Analyze the pie chart for successful launches and success rates by site, and use the scatter plot to find payload ranges and booster versions linked to success.

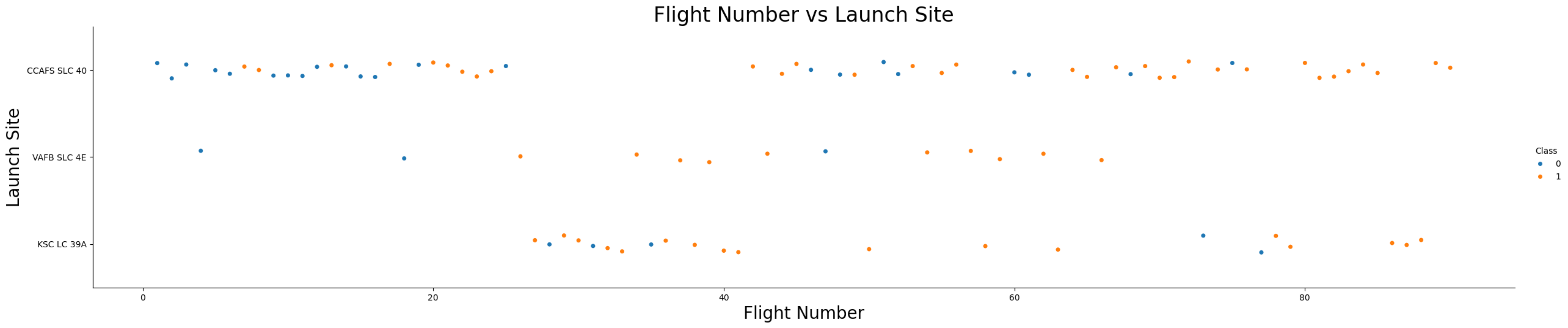
## Predictive Analysis

- "Classification Model Accuracy" shows that four classification models—Logistic Regression, SVM, Decision Tree, and K-Nearest Neighbors—each achieved an accuracy of 0.83. This means all models correctly classified 83% of the instances in the dataset, indicating they performed equally well on this dataset. Further evaluation with other metrics or cross-validation may provide additional insights into model performance.

Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site




**Explanation:**

•**X-axis (Flight Number)**: Represents the sequence of flights. A higher flight number indicates a later flight.

•**Y-axis (Launch Site)**: Indicates the launch site where the rocket was launched. Different sites are shown as distinct categories.

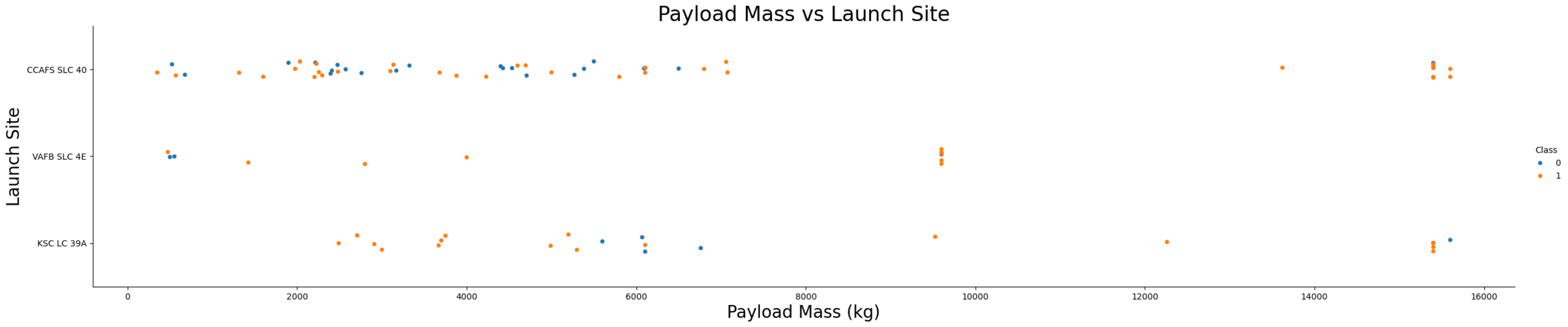
•**Color (Class)**: Represents the success (1) or failure (0) of the landing, with different colors indicating whether the landing was successful or not.

**Observations:**

•**Patterns**: We can observe which launch sites have more successful or unsuccessful landings as the number of flights increases.

•**Success and Failure**: Different colors will help to see how success rates vary across different launch sites over time.

# Payload vs. Launch Site




**Explanation:**

•**X-axis (Payload Mass)**: Represents the mass of the payload being launched.

•**Y-axis (Launch Site)**: Indicates the launch site where the rocket was launched.

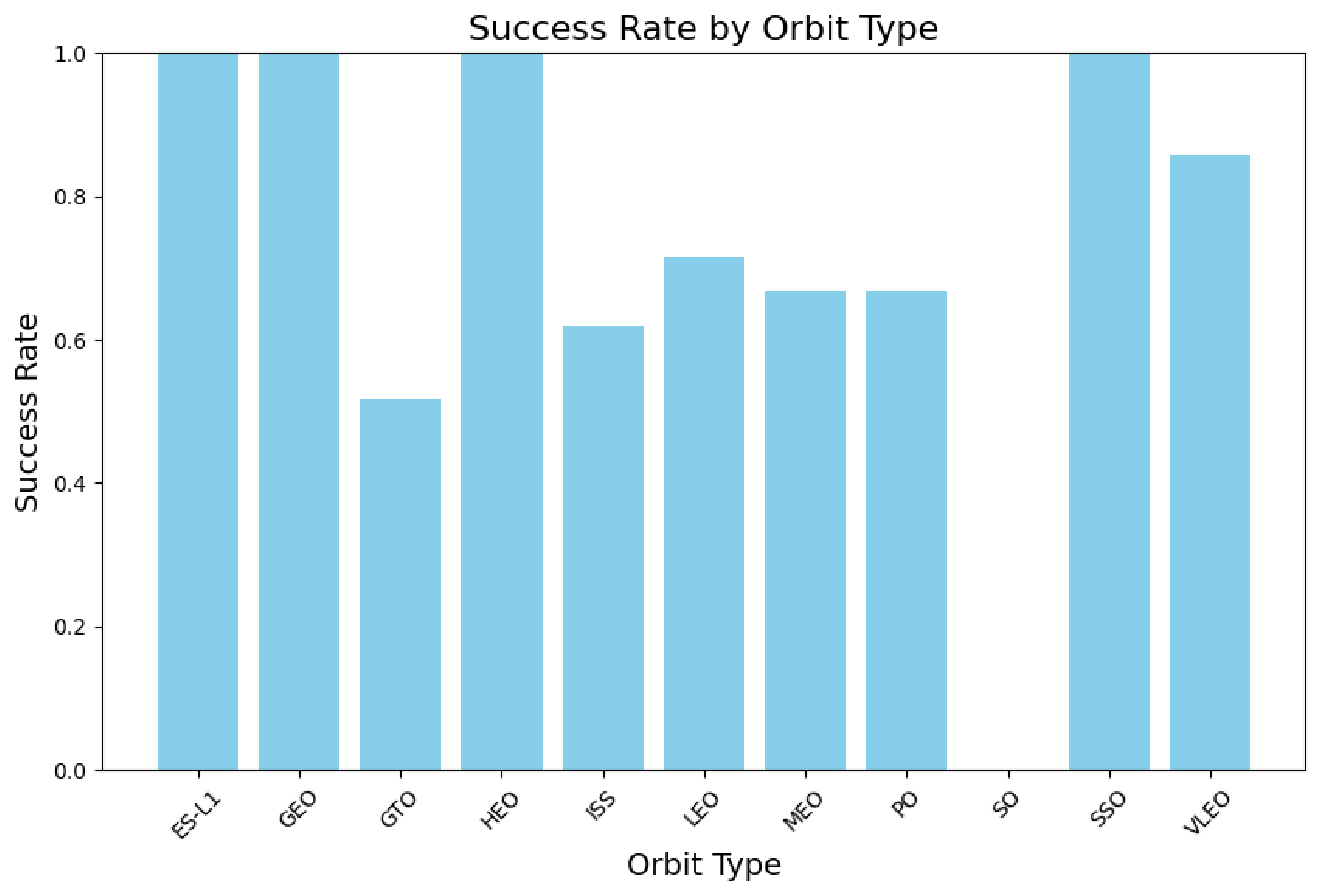
•**Color (Class)**: Shows whether the landing was successful (1) or not (0).

**Observations**:

•The plot reveals how payload mass varies across different launch sites.

•We can see which launch sites handle larger payloads and how this affects the landing success.

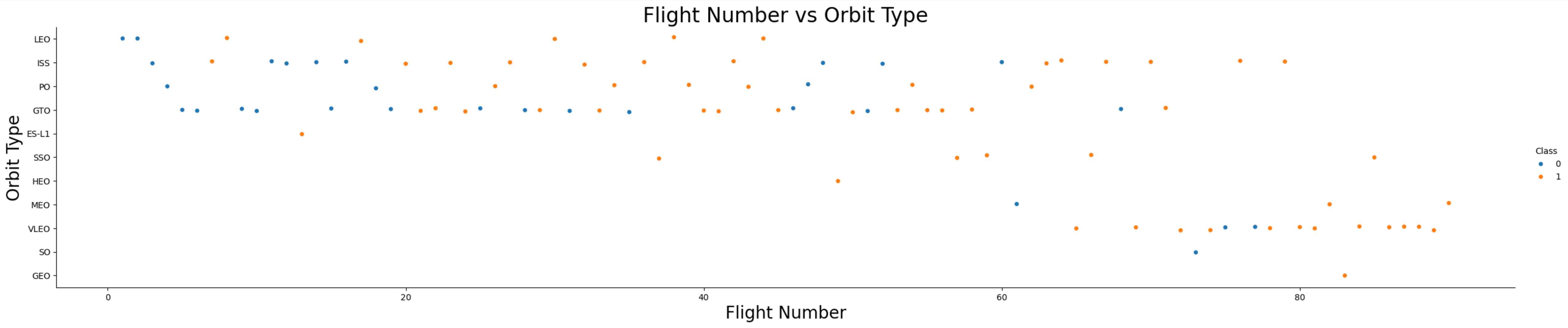
# Success Rate vs. Orbit Type




**Explanation:**
•**X-axis (Orbit Type)**: Represents different types of orbits.
•**Y-axis (Success Rate)**: Shows the average success rate of landings for each orbit type.

**Observations**:
•The bar chart displays which orbit types have higher or lower success rates.
•This helps identify how orbit type affects the likelihood of a successful landing.

# Flight Number vs. Orbit Type




**Explanation:**

•**X-axis (Flight Number)**: Represents the sequence of flights.

•**Y-axis (Orbit Type)**: Indicates the type of orbit for each flight.

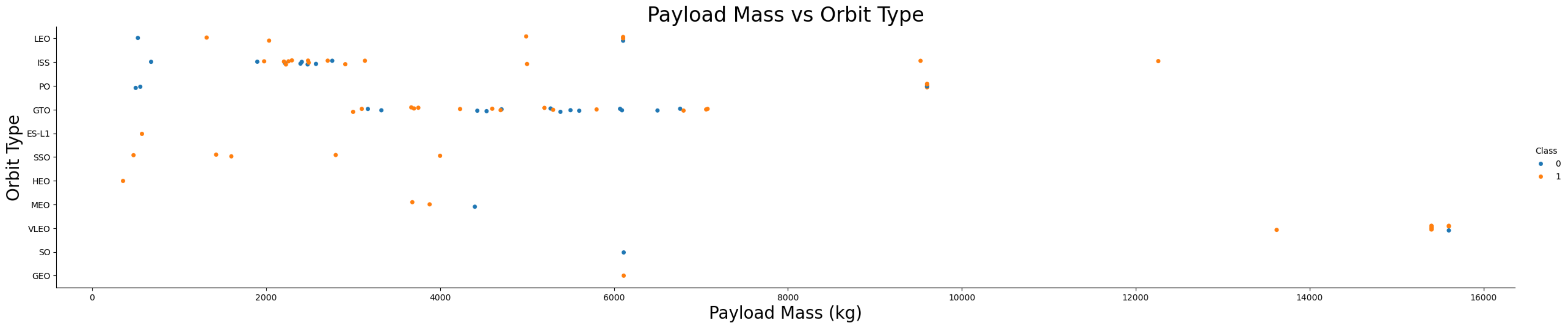
•**Color (Class)**: Shows whether the landing was successful (1) or not (0).

**Observations**:

•The plot helps to visualize if there is a pattern or relationship between the flight number and the orbit type.

•It shows how different orbit types are distributed across various flight numbers and how success rates vary across these flights.

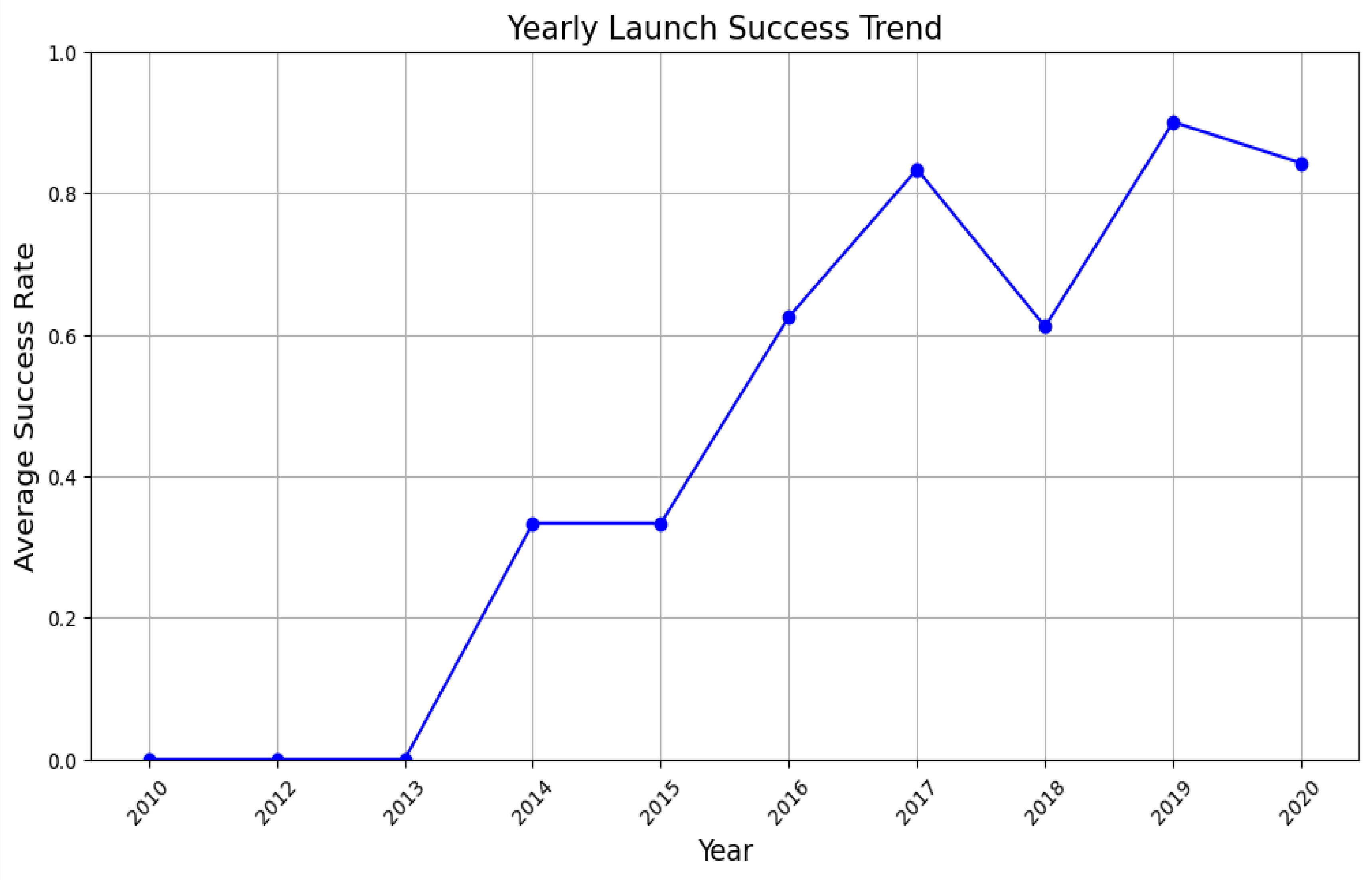
# Payload vs. Orbit Type




**Explanation:**
•**X-axis (Payload Mass)**: Represents the mass of the payload being launched.
•**Y-axis (Orbit Type)**: Indicates the type of orbit for each payload.
•**Color (Class)**: Shows whether the landing was successful (1) or no (0)t.

**Observations**:
•The plot helps visualize how different payload masses are associated with various orbit types.
•It shows if there is a pattern in payload mass distribution across different orbit types and how success rates may vary with payload mass in each orbit.

# Launch Success Yearly Trend




**Explanation:**

•**X-axis (Year)**: Represents the years from the dataset.

•**Y-axis (Average Success Rate)**: Shows the average success rate of Falcon 9 landings for each year.

•**Line**: Tracks the trend of the success rate over the years.

**Observations**:

•The line chart helps visualize how the success rate of Falcon 9 landings has changed over time.

•It shows whether there has been an improvement or decline in the success rate over the years, providing insights into the reliability and performance of the landings.

# All Launch Site Names

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

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

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

The unique launch sites are:

**1.CCAFS LC-40**
**2.VAFB SLC-4E**
**3.KSC LC-39A**
**4.CCAFS SLC-40**

These results indicate the distinct locations from which SpaceX has launched its missions, and they reflect the different sites used for various launches over the years.

# Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTABLE 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 |

**Explanation of the Result**

These records show launches that took place at sites starting with CCA, specifically CCAFS LC-40. Each record includes details about the launch date and time, booster version, launch site, payload, payload mass, orbit, customer, mission outcome, and landing outcome. This provides insight into the specific missions that were launched from CCAFS LC-40 and their respective outcomes.

# Total Payload Mass

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

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

| Total_Payload_Mass |
|---|
| 45596 |

**Explanation of the Result**

The result shows that the total payload mass carried by boosters for NASA (CRS) is 45,596 kilograms. This information is specific to the Commercial Resupply Services (CRS) missions, where NASA contracts private companies, like SpaceX, to deliver cargo to the International Space Station (ISS). The total payload mass reflects the weight of the cargo that has been delivered to the ISS under these contracts.

# Average Payload Mass by F9 v1.1

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

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

| Average_Payload_Mass |
|---|
| 2928.4 |

**Explanation of the Result**
The result shows that the average payload mass carried by the booster version F9 v1.1 is 2,928.4 kilograms. This provides insight into the typical payload capacity for the F9 v1.1 booster version, reflecting its performance and the type of missions it was used for.

# First Successful Ground Landing Date

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

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

| First_Successful_Landing_Date |
|---|
| 2015-12-22 |

The first successful landing outcome on a ground pad occurred on **December 22, 2015**. This indicates the earliest date when SpaceX successfully landed a Falcon 9 rocket's first stage on a ground pad. This milestone is crucial for reducing the cost of launches, as it demonstrates the reusability of the rockets.

# Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass__kg_" > 4(
```

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

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

**Result Explanation:**

- **F9 FT B1022**
- **F9 FT B1026**
- **F9 FT B1021.2**
- **F9 FT B1031.2**

These boosters successfully landed on a drone ship and carried payloads with a mass between 4000 and 6000 kg. This indicates that these specific booster versions have proven their capability to carry substantial payloads and successfully land on a drone ship, showcasing their reusability and reliability.

# Total Number of Successful and Failure Mission Outcomes

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

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

| Mission_Outcome | Total_Count |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

**Result Explanation:**

•**Success**: There are 98 missions with a successful outcome.

•**Success (payload status unclear)**: There is 1 mission where the mission was successful but the payload status is unclear.

•**Failure (in flight)**: There is 1 mission that failed during flight.

This result shows that out of all the recorded missions, the vast majority (98) were successful, with only a couple of exceptions. This highlights SpaceX's high success rate in their missions.

# Boosters Carried Maximum Payload

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

```
%sql SELECT "Booster_Version", "Payload_Mass__kg_" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__k
```

```
 * 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 |

**Explanation of Results:**

- The Booster_Version column lists the names of the boosters.
- The Payload_Mass__kg_ column shows the maximum payload mass (15600 kg) carried by these boosters.

This result shows that multiple boosters, all variants of the F9 B5 model, have carried the maximum payload mass of 15600 kg. This indicates the consistent capability of the F9 B5 booster version to handle the highest payload capacity recorded in the dataset.

# 2015 Launch Records

**%sql** SELECT CASE substr("Date", 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April' WHEN '05' THEN 'May' WHEN '06' THEN 'June' WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September' WHEN '10' THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December' END AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr("Date", 1, 4) = '2015';

```
%sql SELECT CASE substr("Date", 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' TH
```

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

| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

**Explanation of Results:**

•The Booster_Version column lists the versions of the boosters that had failed landing outcomes on drone ships.
•The Launch_Site column shows the launch sites from which these missions were launched.
•The Landing_Outcome column indicates that the landing outcome was a failure on a drone ship.

In this result, we see that all failed landing outcomes on drone ships in 2015 involved the F9 v1.1 booster version, and all launches were from the CCSFS SLC 40 launch site. This indicates a specific trend during that year and helps in understanding the reliability and performance of the booster versions and launch sites during that period.

# 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;

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

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

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

**Explanation of Results:**

•The Landing_Outcome column lists the different landing outcomes, such as Failure (drone ship), Success (ground pad), etc.

•The Count column shows the number of times each landing outcome occurred between June 4, 2010, and March 20, 2017.

•The results are ordered in descending order, with the most frequent landing outcomes appearing at the top.

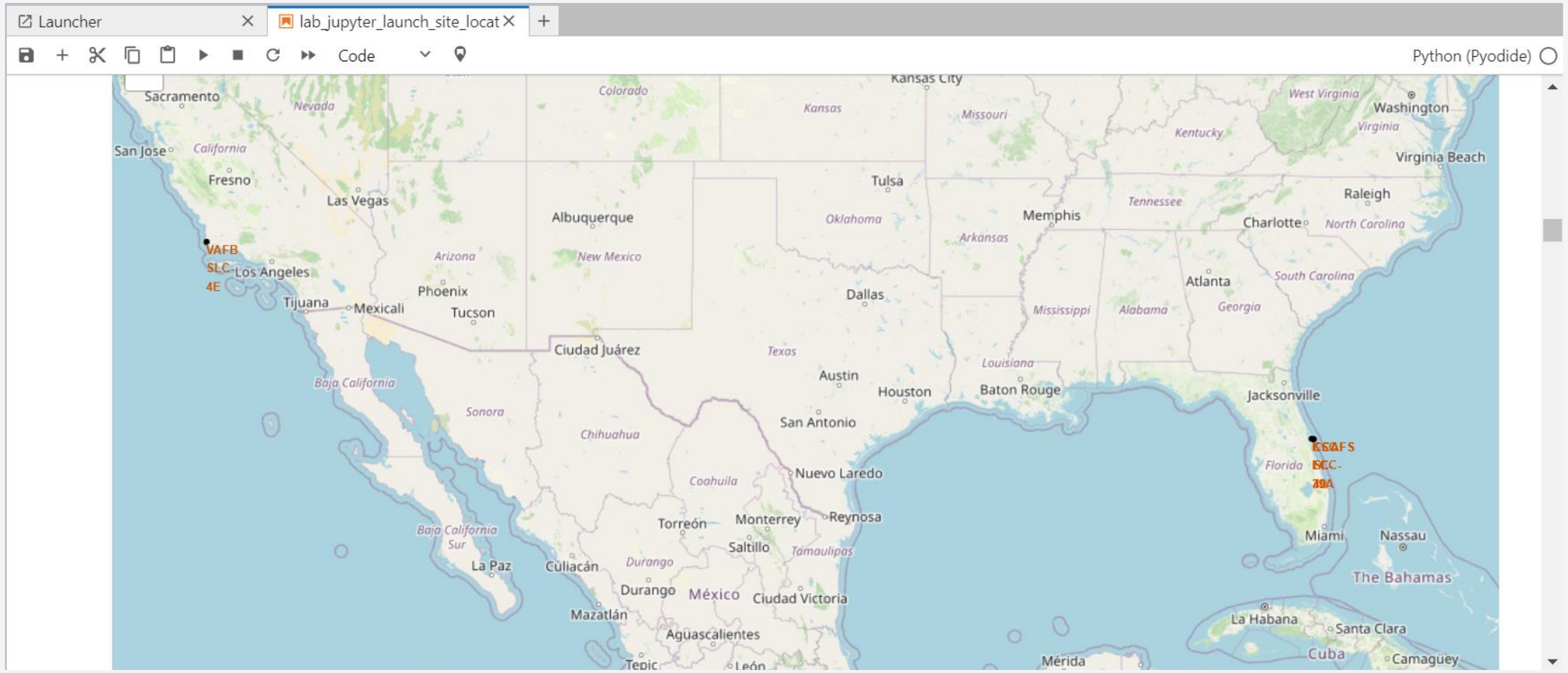
This ranking provides insights into the success and failure rates of different landing attempts over the specified period, highlighting which outcomes were most common and potentially identifying areas for improvement or focus in SpaceX's landing strategies.

Section 3

# Launch Sites Proximities Analysis

# <Folium Map Screenshot 1>

**Title for Folium Map Screenshot 1:**
"Map of SpaceX Launch Sites with Location Markers“




**Key Elements and Findings:**

**1.Launch Site Markers:**

- Circles and markers show locations of SpaceX launch sites like CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E.

**2.Center Location:**

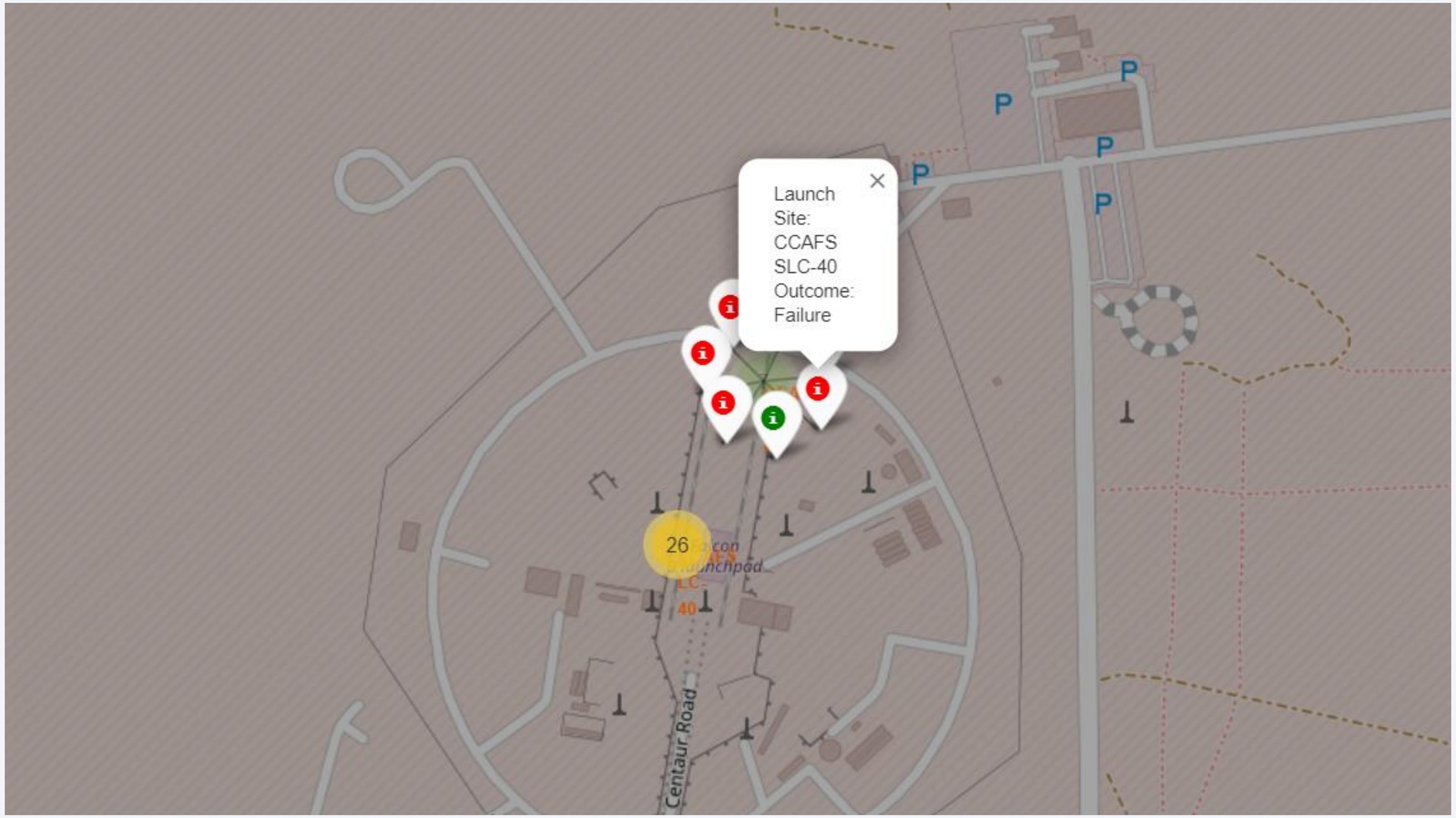
- The map is centered on NASA Johnson Space Center for reference.

**3.Proximity Insights:**

- Launch sites are mostly near the coast, which is strategic for safety and efficiency. Sites are not strictly near the equator but are positioned to optimize launch conditions.

# <Folium Map Screenshot 2>

**Title for Folium Map Screenshot 2:**
"Launch Outcomes Map with Color-Labeled Markers"




**Key Elements and Findings:**

**1.Color-Coding:**

- Green markers for successful launches, red for failed ones.

**2.Marker Clusters:**

- Clusters simplify the view for frequent launches.

**3.Findings:**

- Shows success rates by site, with color distribution indicating performance trends.

# <Folium Map Screenshot 3>

**Title for Folium Map Screenshot 3:**
"Launch Site Proximity with Distance Markers“

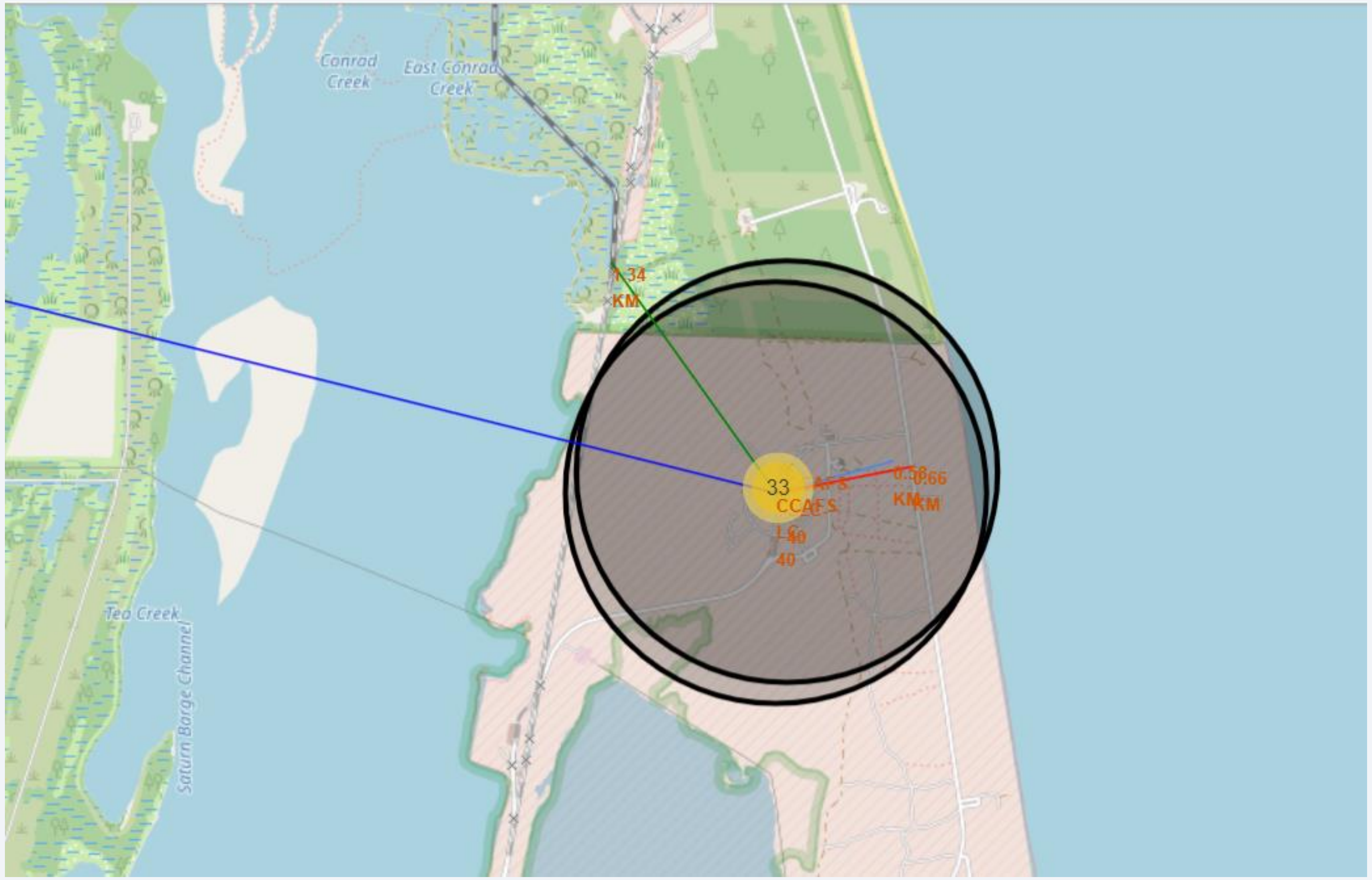
**Key Elements and Findings:**

**1.Distance Markers:**
- Shows distances from the launch site to railways, highways, etc.

**2.Lines and Markers:**
- Lines connect the launch site to these features, with distances displayed on markers.

**Findings:**

**1.Proximity Analysis:**
- Indicates how close the launch site is to critical infrastructure.



Section 4

# Build a Dashboard with Plotly Dash

# <Dashboard Screenshot 1>

**Explanation of Important Elements and Findings**

**1. Title:** Clearly indicates the chart represents successful launches across all sites.

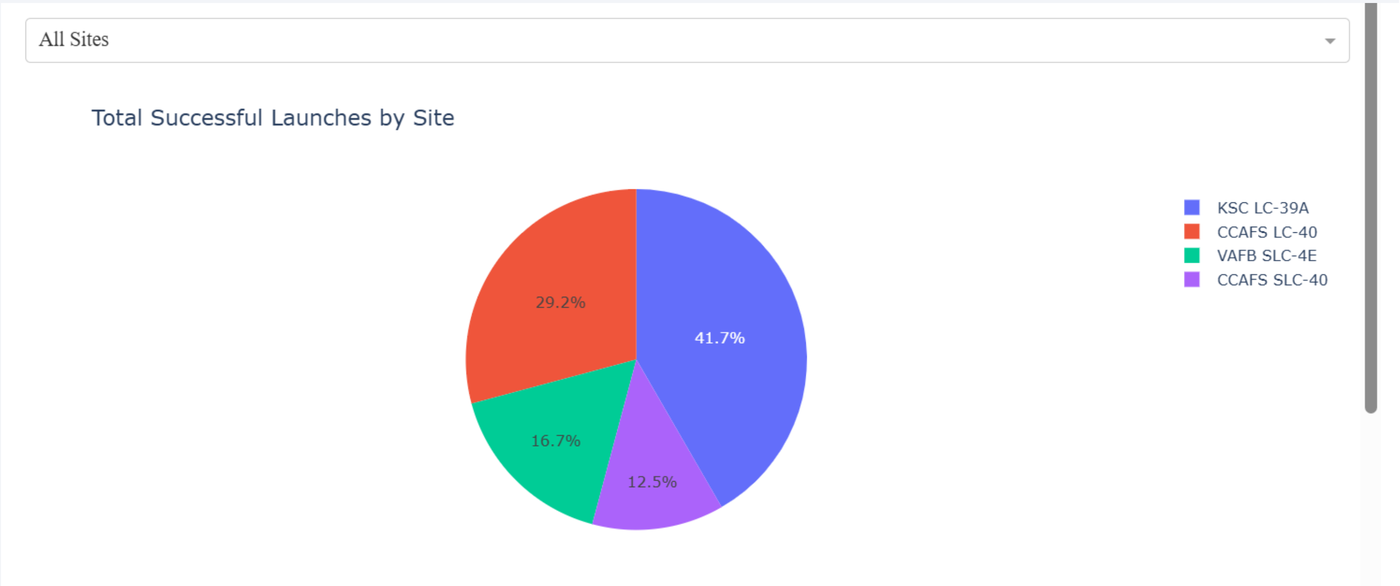
**2. Pie Chart Slices:** Each slice represents a launch site with size proportional to successful launches.

**3. Labels and Legends:** Identifies launch sites and their associated colors.

**4. Percentage/Count Display:** Shows the count or percentage of successful launches for each site.

**Key Findings:**

- **Distribution of Successes:** Easily compare success rates of different launch sites.
- **Site Comparison:** Identify most and least successful sites at a glance.
- **Overall Success Rates:** Assess performance and reliability of launch sites.




**Launch Success Count for All Sites**

# <Dashboard Screenshot 2>

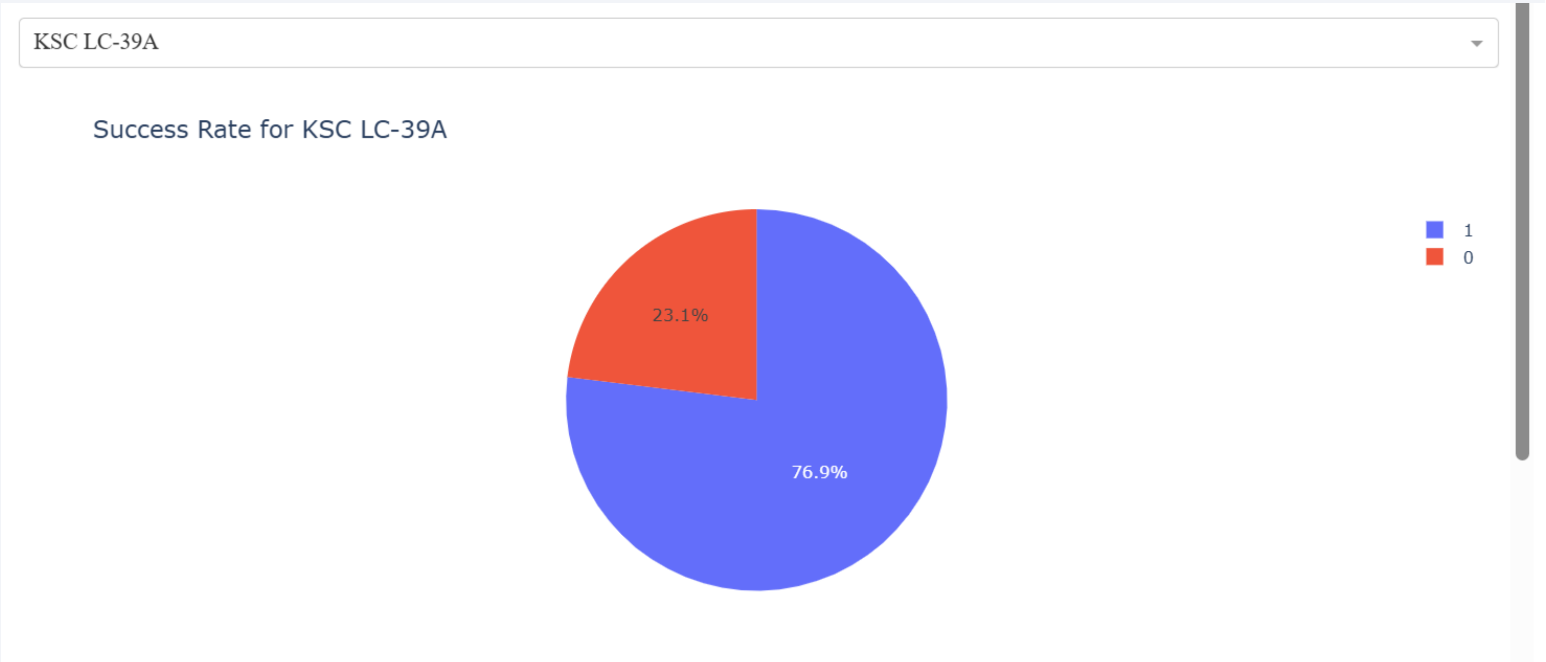
**Key Points from the Pie Chart**

**1. Title**: Reflects the most successful launch site's success rate.

**2. Sections**: Shows proportions of successful vs. unsuccessful launches.

**3. Color Scheme**: Differentiates between success and failure.

**4. Insight**: Provides the success ratio at the site, highlighting its performance.




**Launch Success Distribution for the Most Successful Launch Site**

# <Dashboard Screenshot 3>

**Screenshots**

**1. Run the App**: Generate scatter plots by adjusting the payload range slider.

**2. Capture Screenshots**: Take screenshots for different payload ranges.

**Key Points**

**1. Title**: Describes the scatter plot of payload vs. launch success.

**2. Scatter Plot**:

1. **X-Axis**: Payload mass.
2. **Y-Axis**: Launch success (1 = success, 0 = failure).
3. **Color**: Different booster versions/ Launch Site.

**3. Findings**:

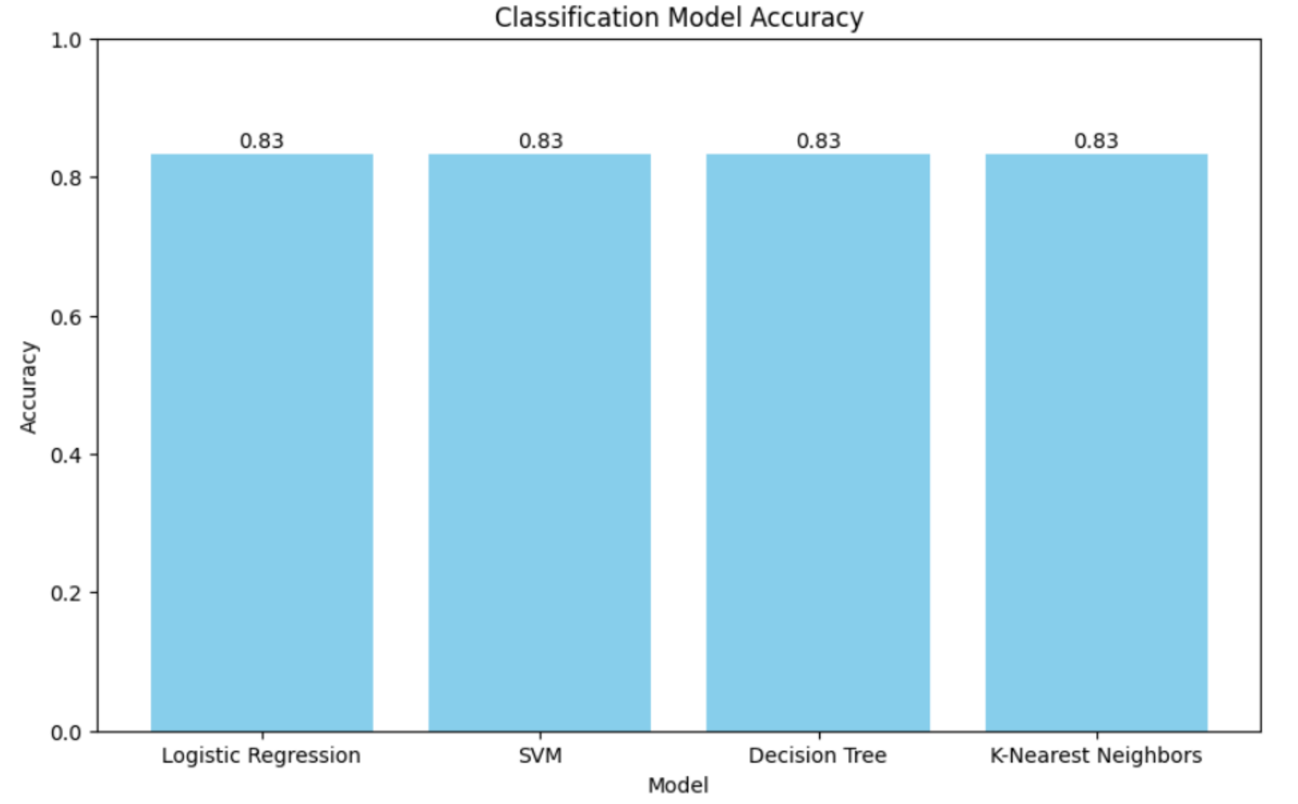
1. **Payload Ranges**: Determine which payload ranges have higher success rates.
2. **Booster Versions**: Identify which boosters are linked to higher success rates for different payloads.


**Payload Mass vs. Launch Success for All Sites**

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

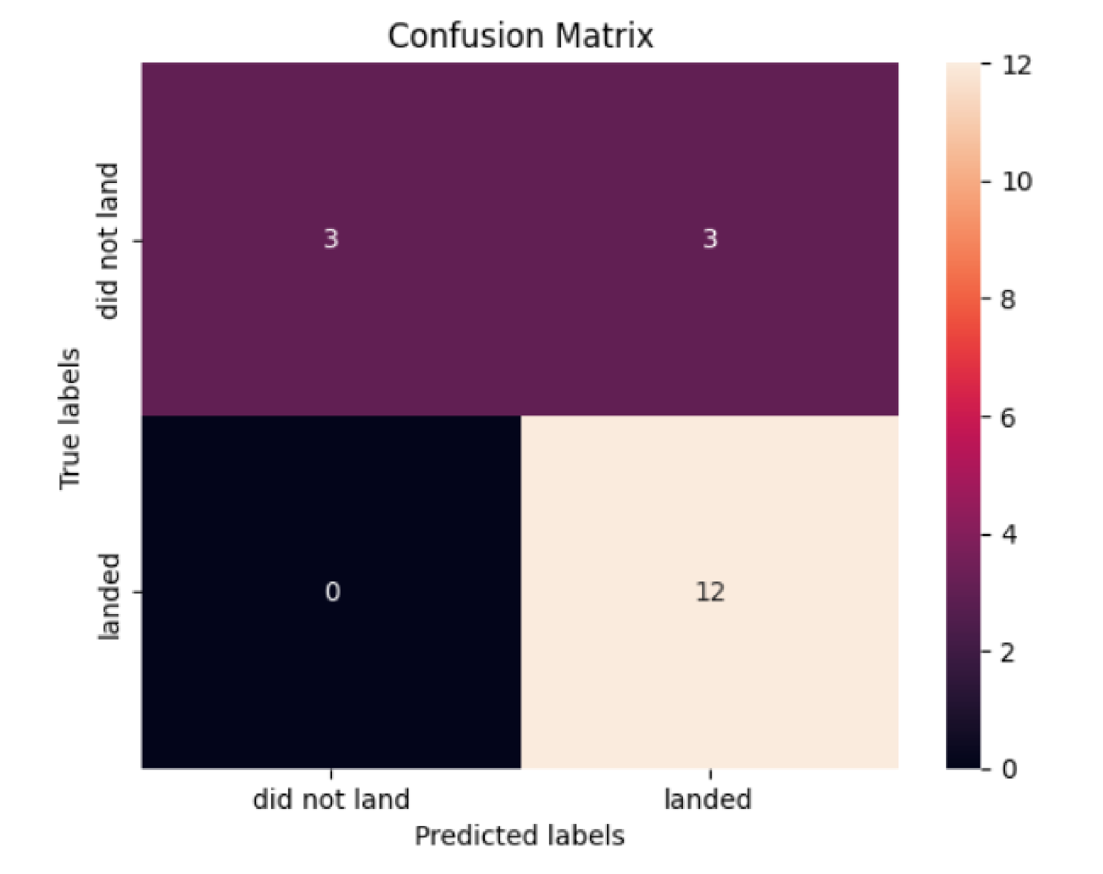



Each model has an accuracy of 0.83 both Logistic Regression , SVM, Decision Tree, K-Nearest Neighbors

# Confusion Matrix

The confusion matrix shows the performance of a classification model in predicting whether a SpaceX Falcon 9 first stage landing attempt was successful ("landed") or not ("did not land"). Explanation:

1. **True Positive (TP):** The model predicted "landed," and the actual label was "landed“. Value: 12 (bottom right cell)
2. **True Negative (TN):** The model predicted "did not land," and the actual label was "did not land“. Value: 3 (top left cell)
3. **False Positive (FP):** The model predicted "landed," but the actual label was "did not land“. Value: 3 (top right cell)
4. **False Negative (FN):** The model predicted "did not land," but the actual label was "landed“. Value: 0 (bottom left cell)



# Conclusions

•**Model Performance**: All four models (Logistic Regression, SVM, Decision Tree, and KNN) have identical performance in terms of accuracy, each scoring 0.83

•**Model Selection**: Since the accuracy is the same across all models, other factors such as model complexity, interpretability, training time, and application context should be considered when choosing a model for deployment

•**Further Analysis**: Additional metrics such as precision, recall, F1-score, and ROC-AUC could provide more insight into the model performance. It might also be useful to perform cross-validation to ensure the robustness of the models.

Overall, the chart indicates that any of the four models could be a viable choice based solely on accuracy.

# Appendix

**Dataset:**
1. https://api.spacexdata.com/v4/rockets/
2. https://api.spacexdata.com/v4/launchpads/
3. https://api.spacexdata.com/v4/payloads/
4. https://api.spacexdata.com/v4/cores/
5. https://api.spacexdata.com/v4/launches/past
6. https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
7. https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv
8. https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv
9. https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv
10. https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv
11. https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv
12. https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv

# Appendix

**Import Libraries**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing, model_selection,
linear_model, svm, tree, metrics

def plot_confusion_matrix(y, y_predict):
    cm = metrics.confusion_matrix(y, y_predict)
    sns.heatmap(cm, annot=True)
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()
```

**Evaluate Models**

```
for model, name in [(grid_lr, "Logistic Regression"),
(grid_svm, "SVM"), (grid_tree, "Decision Tree"),
(grid_knn, "KNN")]: y_pred = model.predict(X_test)
print(f"{name}\n{metrics.classification_report(Y_tes
t, y_pred)}") plot_confusion_matrix(Y_test, y_pred)
```

**Load and Preprocess Data**

```
data = pd.read_csv("dataset_part_2.csv")
X = pd.read_csv("dataset_part_3.csv")
Y = data['Class'].to_numpy()
X_standardized = preprocessing.StandardScaler().fit_transform(X)
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X_standardized,
Y, test_size=0.2, random_state=42)
```

**Train Models**

**Logistic Regression**

```
grid_lr = model_selection.GridSearchCV(linear_model.LogisticRegression(), {'C':
[0.01, 0.1, 1, 10, 100]}, cv=10) grid_lr.fit(X_train, Y_train)
```

**Support Vector Machine**

```
grid_svm = model_selection.GridSearchCV(svm.SVC(), {'C': [0.01, 0.1, 1, 10, 100],
'kernel': ['linear', 'rbf']}, cv=10) grid_svm.fit(X_train, Y_train)
```

**Decision Tree**

```
grid_tree = model_selection.GridSearchCV(tree.DecisionTreeClassifier(),
{'max_depth': [3, 5, 7, 9, 11]}, cv=10) grid_tree.fit(X_train, Y_train)
```

**K-Nearest Neighbors**

```
grid_knn = model_selection.GridSearchCV(neighbors.KNeighborsClassifier(),
{'n_neighbors': [3, 5, 7, 9, 11]}, cv=10) grid_knn.fit(X_train, Y_train)
```

# Thank you!