Winning Space Race with Data Science

Diciembre 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data Collection through AP
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches at a cost of 62 million dollars per launch, significantly lower than the 165 million dollars or more charged by other providers. A major contributor to these savings is the reusability of the first stage of the Falcon 9 rocket.

By predicting whether the first stage will land successfully, we can estimate the overall launch cost. This insight is valuable for alternate companies aiming to compete with SpaceX for rocket launch contracts.

The objective of this project is to develop a machine learning pipeline capable of predicting the successful landing of the first stage of a Falcon 9 rocket.

- Problems you want to find answers

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

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

The project data was gathered using two primary methods:

1. SpaceX API

- Data was retrieved via HTTP GET requests and parsed from JSON format into a structured pandas DataFrame using .json_normalize().
- Missing values and inconsistencies were cleaned and handled to ensure quality.

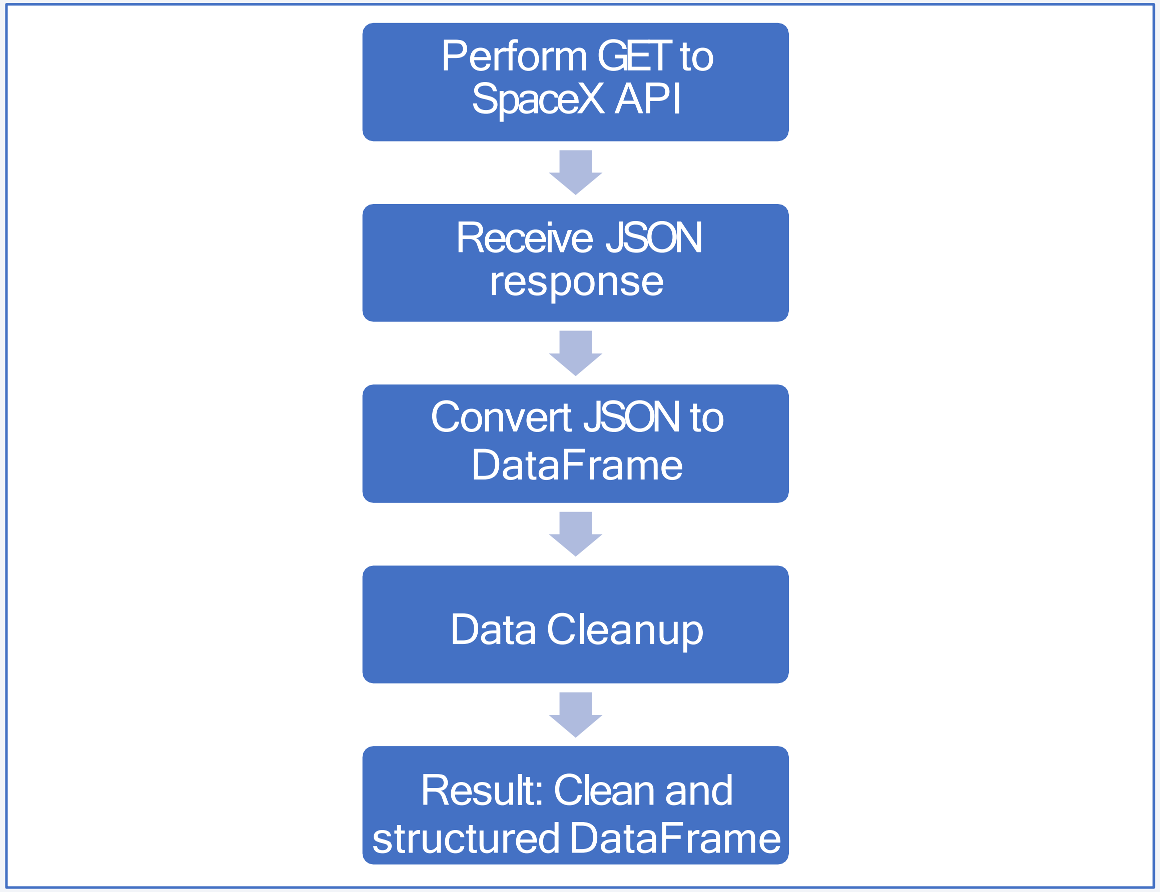
2. Web Scraping

- Additional launch records were scraped from Wikipedia using BeautifulSoup.
- HTML tables were extracted, parsed, and converted into pandas DataFrames for analysis.

The final dataset combines both sources, forming a clean and unified foundation for analysis and modeling.

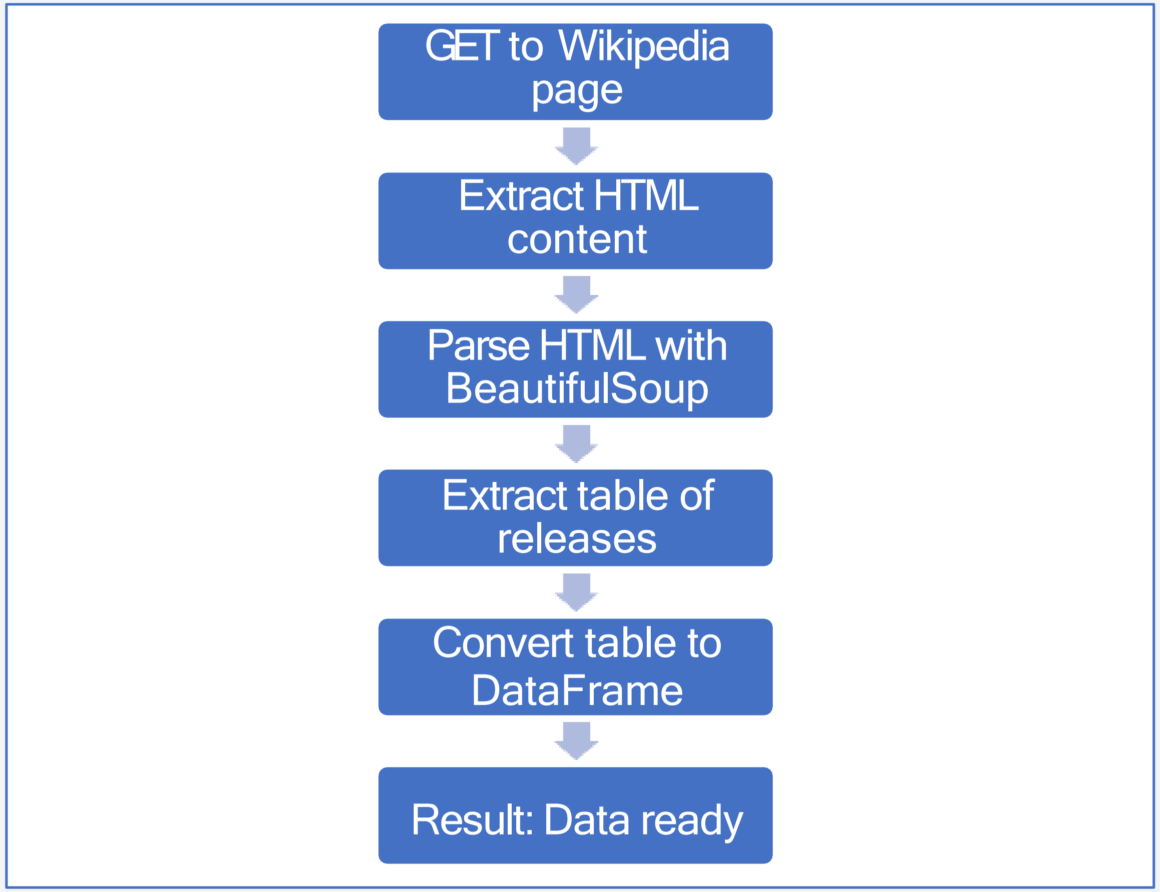
Data Collection - SpaceX API

- Data Retrieval:
  - Utilized an HTTP GET request to query the SpaceX REST API.
  - Collected launch data, performed basic wrangling and applied data formatting to structure the response.
- Data Processing:
  - Transformed the JSON response into a pandas DataFrame for further analysis.
  - Performed data cleaning to handle missing or inconsistent values.
- External Reference:
  - For full implementation and output, refer to the completed notebook on GitHub: [SpaceX API Data Collection Notebook]



Data Collection - Scraping

- Web Scrape Process:
  - Used HTTP GET requests to retrieve Falcon 9 launch records from a Wikipedia page.
  - Leveraged BeautifulSoup to parse the HTML content and extract the launch table.
- Data Transformation:
  - Extracted the launch table and converted it into a pandas DataFrame for analysis.
- External Reference:
  - The full implementation, including code and results, can be accessed here:
  - [Web Scraping Notebook on GitHub]()

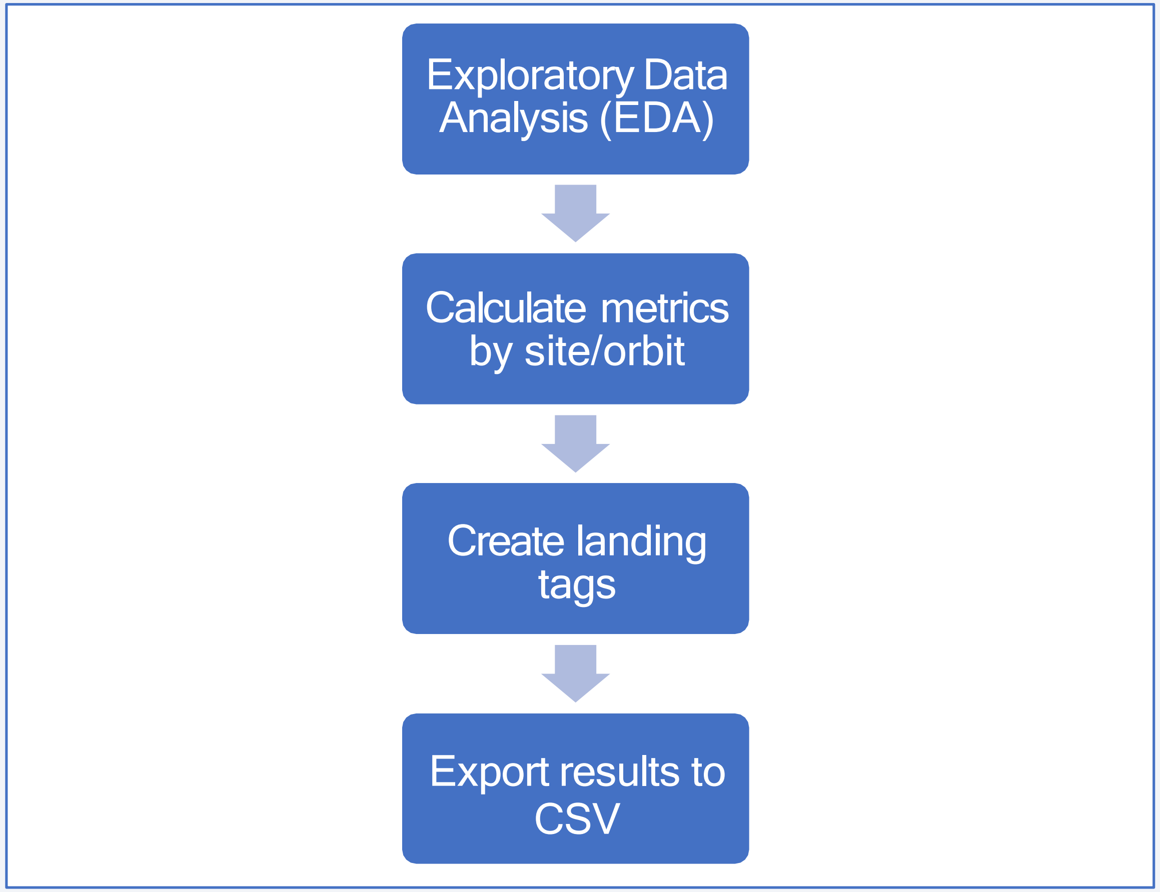


Data Wrangling

- Key Steps of Data Wrangling Process:
  - Exploratory Data Analysis (EDA): Identified patterns and key insights in the dataset.
  - Key Metrics Summary: Calculated launches per site and occurrences of orbit types.
  - Outcome Label Creation: Created landing outcome labels from the *Outcome* column.
  - Export Process: Saved the cleaned and transformed data as a CSV file for further analysis.

- External Reference:

- For full implementation and outputs, refer to: [GitHub Notebook - Data Wrangling]



EDA with Data Visualization

- **Charts and Purpose**
  - **Flight Number vs Launch Site**: Understand the relationship between flights and launch sites (Cat plot).
  - **Payload Mass vs Launch Site**: Analyze payload mass variations and mission success (Scatter plot).
  - **Success Rate by Orbit Type**: Determine orbit types with higher success rates (Bar chart).
  - **Flight Number vs Orbit Type**: Identify flight distribution and success rates by orbit (Scatter plot).
  - **Payload Mass vs Orbit Type**: Study payload mass across orbit types (Scatter plot with colors).
  - **Yearly Trend of Launch Success**: Observe trends in launch success over time (Line chart).
- **External Reference**
  - GitHub Notebook - EDA with Data Visualization

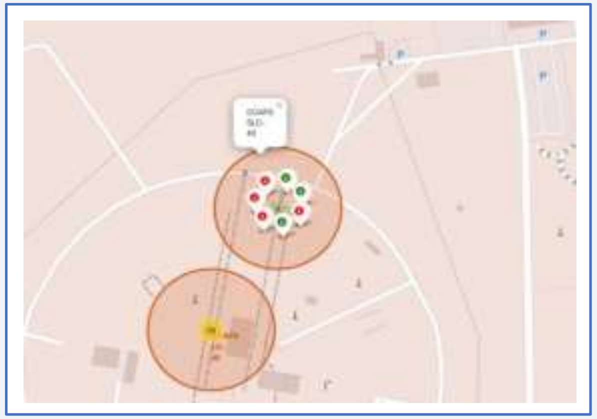
EDA with SQL

Key SQL Queries

- **Unique Launch Sites: Retrieve distinct launch site names**
- **Filtered Records: Launch sites starting with 'CCA'.**
- **Payload Analysis**:
  - Total payload mass for NASA (CRS).
  - Average payload mass for 'F9 v1.1' rockets.
- **Landing Outcomes**
  - First successful ground landing date.
  - Count of successful and failed landings.
- **Performance Analysis**
  - Boosters with maximum payload.
  - Drone landings with payload between 4000-6000 kg.
  - Failed drone landings in 2015.
- **External Reference**
- The complete notebook can be accessed here: [EDA with SQL Notebook on GitHub]

Build an Interactive Map with Folium

- **Markers**: Indicate key points such as launch sites and display success or failure outcomes using color-coded markers (green for success, red for failure).
- **Circles**: Highlight specific areas, such as launch site locations, for better visualization of geographic positions.
- **Marker Clusters**: Group multiple markers at the same coordinates, such as multiple launches at a single site, to simplify the map.
- **Lines**: Represent distances between launch sites and nearby coastal points, helping analyze proximity and location relationships.
- **External Reference**: [GitHub Link]

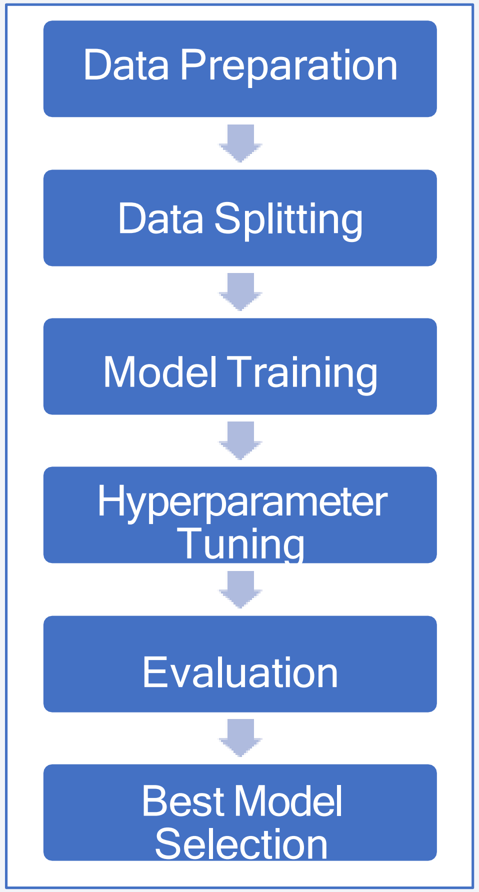


Build a Dashboard with Plotly Dash

- **Pie Charts**: Display the total number of launches by specific launch sites to visualize launch frequency and site activity.
- **Scatter Plot**: Illustrates the relationship between Outcome (success/failure) and Payload Mass (Kg) for different booster versions, helping identify patterns or trends in payload performance.
- **Interactions**: The dashboard allows interactive exploration of the data, enabling users to filter by launch sites, booster versions, or payload mass ranges for deeper insights.
- **External Reference**: GitHub Link

Predictive Analysis (Classification)

- **Data Preparation:**
  - Extracted the target variable Class and standardized the features X using StandardScaler.
- **Data Splitting:**
  - Split data into training and testing sets (80/20) using train_test_split.
- **Model Training and Hyperparameter Tuning:**
  - Logistic Regression: Tuned C, penalty, and solver (accuracy: 84.6%).
  - SVM: Tuned kernel, C, and gamma (accuracy: 84.8%).
  - Decision Tree: Optimized criterion, max_depth, and min_samples_leaf (accuracy: 77.8%).
  - KNN: Tuned n_neighbors, algorithm, and p (accuracy: 84.8%).
- **Model Evaluation:**
  - Confusion matrices and test accuracy scores were analyzed to evaluate model performance.
- **Best Performing Model:**
  - Logistic Regression provided the best balance of accuracy at 83%.
- External Reference: [GitHub Link]



Results

- **Exploratory Data Analysis (EDA) Results:**
  - Insights from the dataset were derived using visualizations and statistics, highlighting key patterns and trends.
- **Interactive Analytics Demo:**
  - Screenshots showcase interactive plots and graphs developed for dynamic exploration of data relationships.
- **Predictive Analysis Results:**
  - Performance of models: Logistic Regression, SVM, Decision Tree, and KNN.
  - Confusion matrices and accuracy scores were analyzed.
  - Logistic Regression provided the best balance with 83% accuracy

Section 2

Insights drawn from EDA

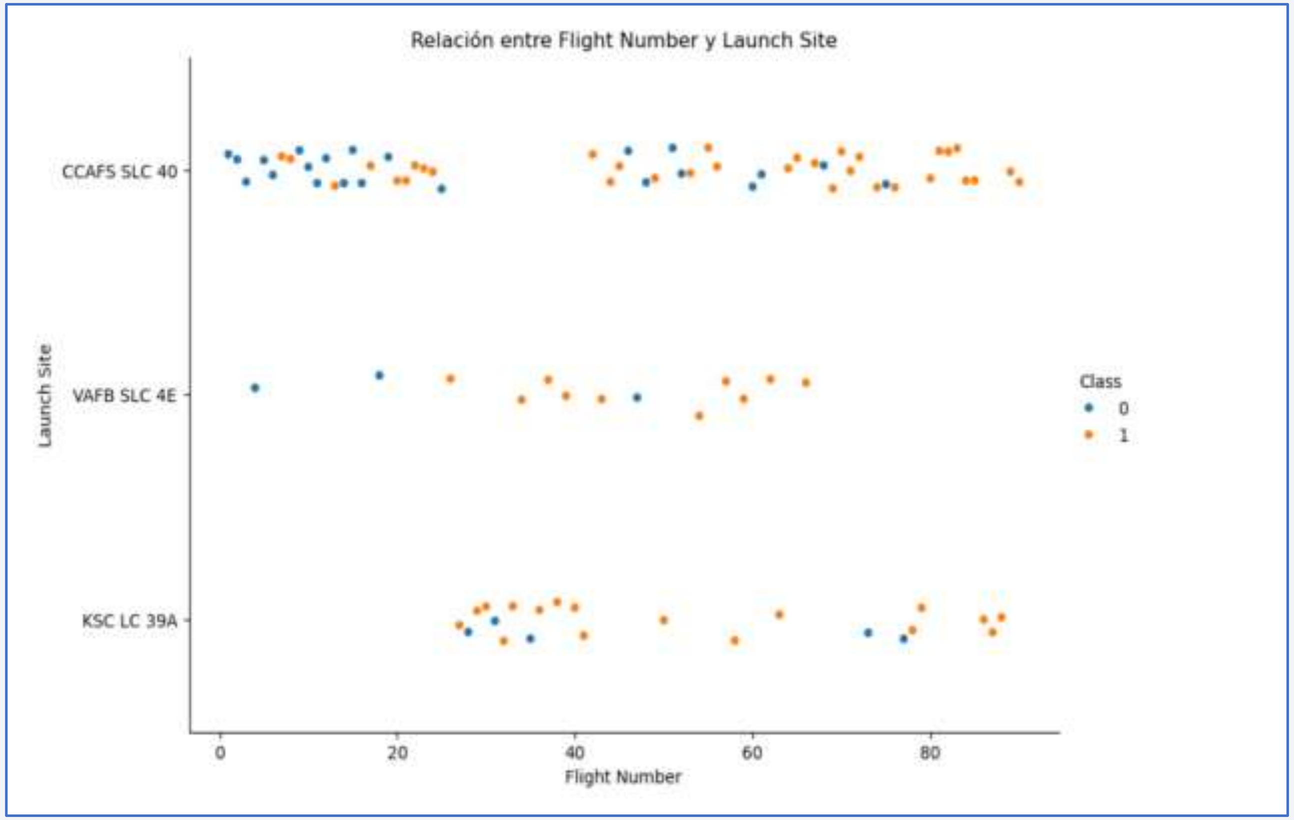
Flight Number vs. Launch Site

Graph Explanation:

- The scatter plot shows the relationship between Flight Number and Launch Site.

Key Insight:

- From the plot, it can be observed that the higher the number of flights at a launch site, the greater the success rate.
- For example, sites with more flight numbers, such as CCAFS SLC 40 and KSC LC 39A, have a higher concentration of successful launches (represented in orange).
- This suggests that experience and operational frequency at a launch site positively impact the likelihood of launch success.



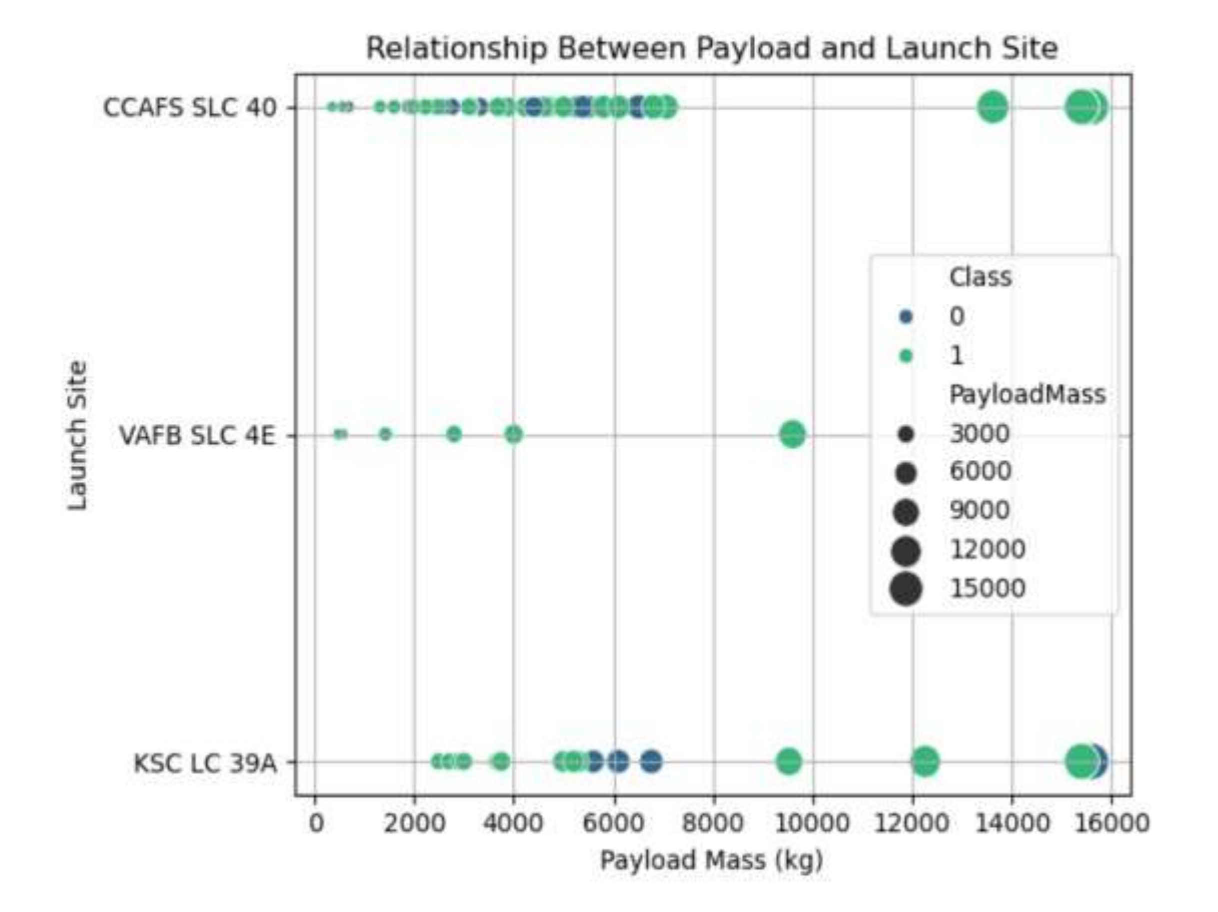
Payload vs. Launch Site

Graph Explanation:

- The scatter plot visualizes the relationship between Payload Mass (kg) and Launch Site.

Key Insight:

- From the plot, it is evident that heavier payloads tend to be associated with higher success rates (green points representing successful launches).
- Launch sites like KSC LC 39A and CCAFS SLC 40 can accommodate heavier payloads, and these sites have a higher success rate for payloads up to 15,000 kg.
- VAFB SLC 4E shows limited launches with relatively lower payload mass.
- This indicates that the payload capacity at certain launch sites influences success outcomes.



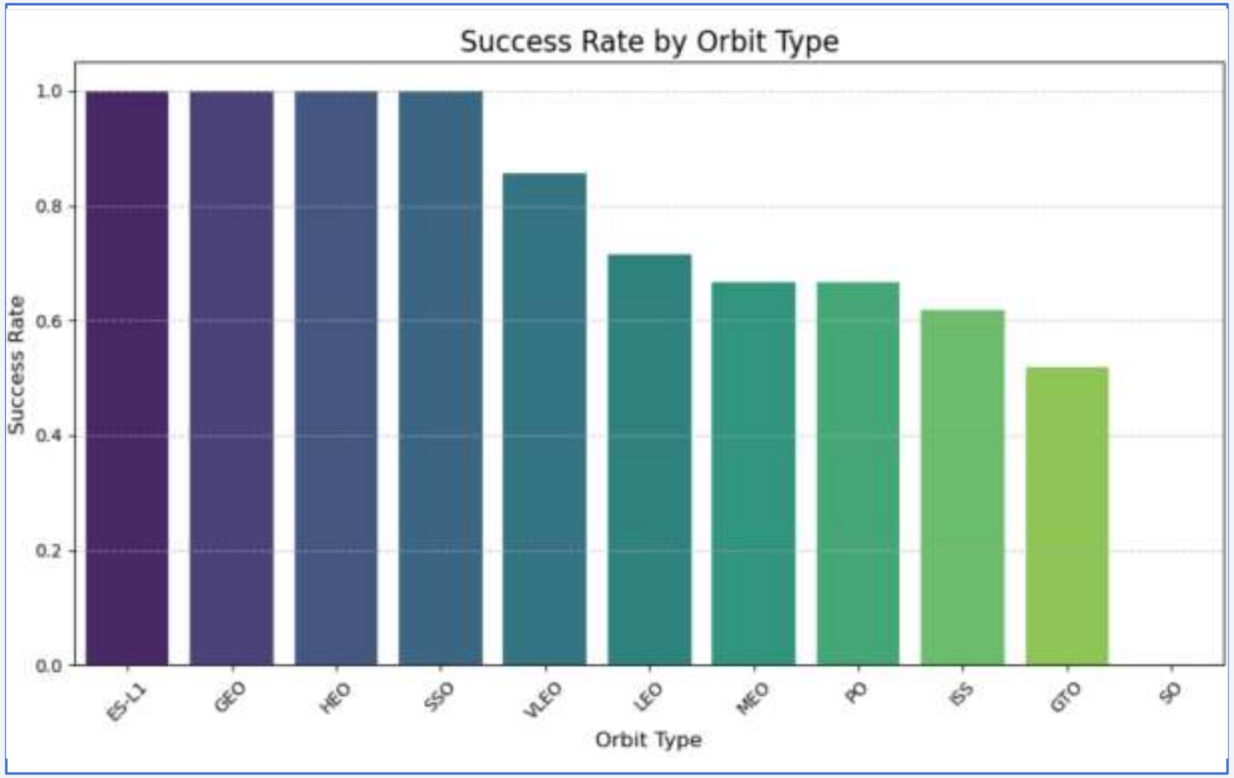
Success Rate vs. Orbit Type

Graph Explanation:

- The bar chart shows the success rate for different orbit types.

Key Insights:

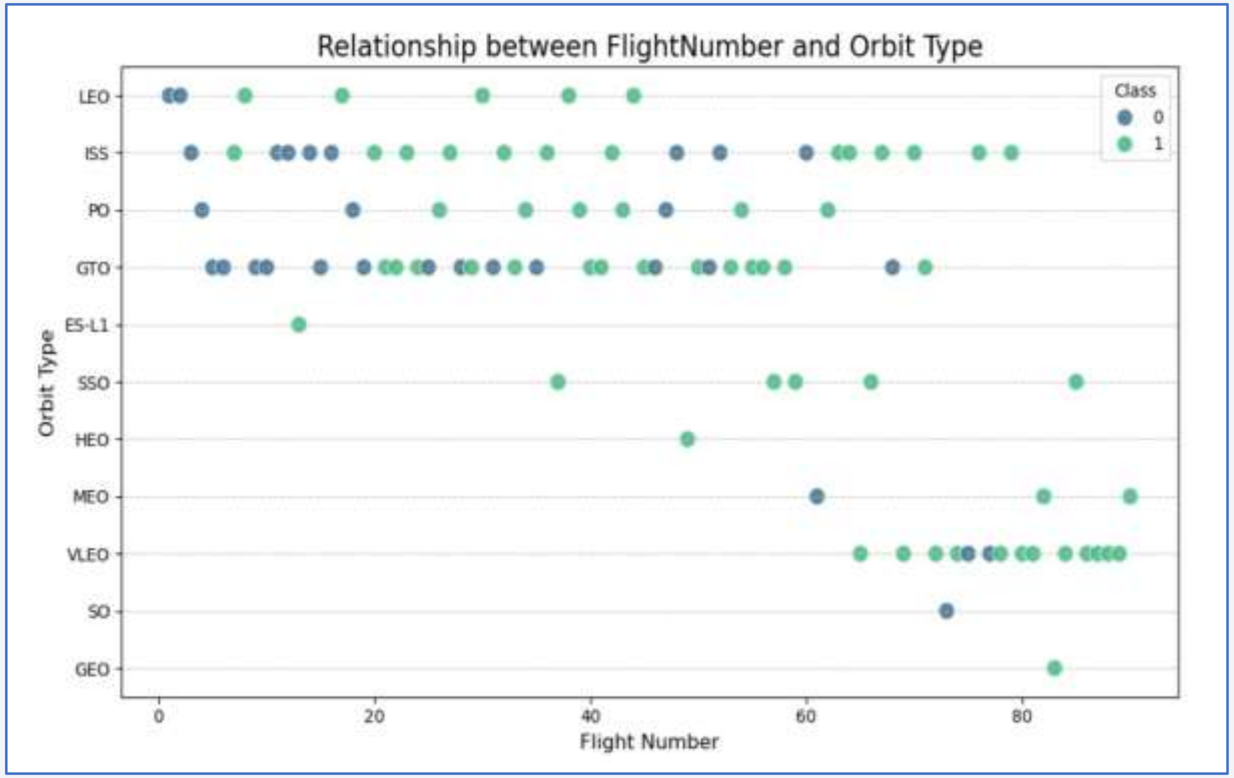
- ES-L1, GEO, HEO, and SSO have a 100% success rate.
- VLEO and LEO also show high success rates above 80%.
- GTO has the lowest success rate (~50%), highlighting challenges for this orbit.



Flight Number vs. Orbit Type

Graph Explanation:

- The scatter plot shows the relationship between Flight Number and Orbit Type. Each point represents a flight, with the success class indicated by color:
  - Green: Success (Class = 1)
  - Blue: Failure (Class = 0)



Key Insights:

- Higher flight numbers are linked to increased success rates, especially in LEO and ISS orbits.
- GTO orbit shows mixed outcomes, reflecting greater challenges.

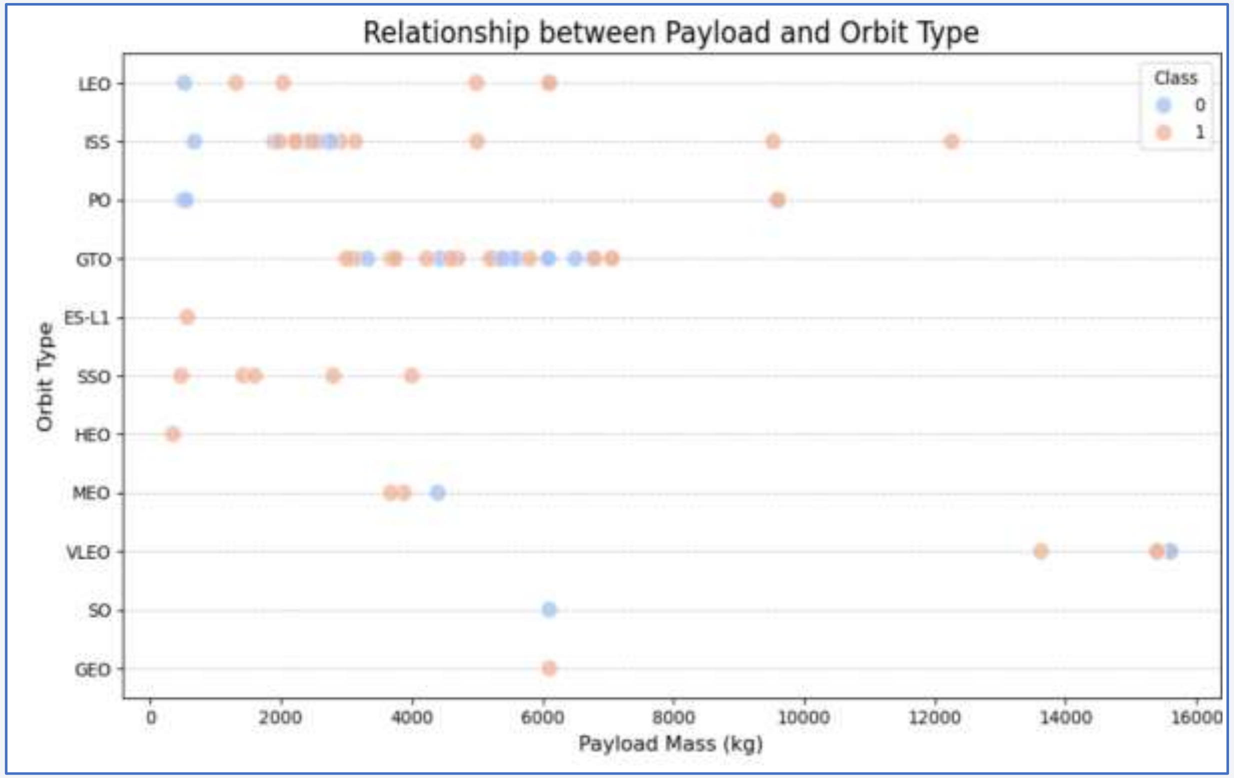
Payload vs. Orbit Type

Graph Explanation:

- The scatter plot shows the relationship between Payload Mass and Orbit Type. Successes (Class 1) are in orange, and failures (Class 0) in blue.

Key Insights:

- LEO supports a wide payload range with higher success rates.
- GTO has mixed results, especially with heavier payloads.
- Lower payloads generally show higher success across all orbit types.



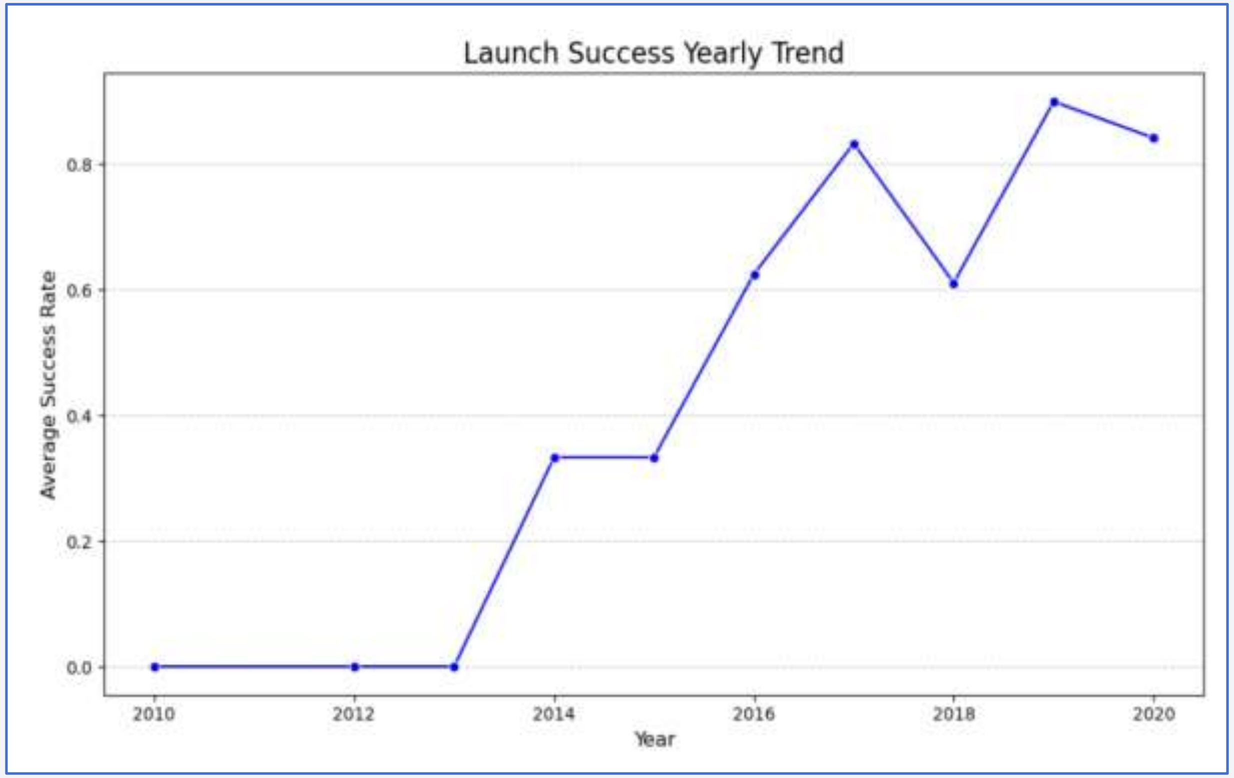
Launch Success Yearly Trend

Graph Explanation:

- This line chart illustrates the yearly average success rate of launches over time, showing how success has improved significantly.

Key Insights:

- Between 2010 and 2013, success rates were consistently low.
- From 2014 onward, there was a steady upward trend, reflecting advancements in technology and processes.
- By 2019, the success rate reached its peak, indicating significant reliability improvements in launch operations.
- Although there is a slight drop in 2020, the overall trend remains positive.



All Launch Site Names

SQL Query Explanation:

- The query uses the DISTINCT keyword to retrieve only the unique launch site names from the SpaceX_Missions table. This avoids duplicate entries in the result.

Query Result:

- The unique launch sites identified are:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A

Key Insight:

- These launch sites represent the primary locations where SpaceX missions have been conducted.

```python
query_1 = """
SELECT DISTINCT "Launch_Site"
FROM SpaceX_Missions;
"""
result_1 = pd.read_sql_query(query_1, con)
print(result_1)
```

```
    Launch_Site
0   CCAFS LC-40
1   VAFB SLC-4E
2    KSC LC-39A
3  CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

Query Explanation:

- We used the LIKE operator with 'CCA%' to find 5 launch site records starting with 'CCA', limiting the results to the first 5 rows.

Results:

- All records belong to the CCAFS LC-40 launch site, showing details like Date, Payload, Orbit, and Mission Outcome.

```
query_2 = """
SELECT *
FROM SpaceX_Missions
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
"""
result_2 = pd.read_sql_query(query_2, con)
result_2. head()
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

We used the SUM function to calculate the total payload mass for missions where the customer is NASA (CRS). The total payload carried by boosters is 45,596 kg, as shown in the query result.

```python
query_3 = """
SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass
FROM SpaceX_Missions
WHERE "Customer" LIKE 'NASA (CRS)';
"""
result_3 = pd.read_sql_query(query_3, con)
result_3.head()
```

	Total_Payload_Mass
0	45596

Average Payload Mass by F9 v1.1

We used the AVG function to calculate the average payload mass for missions where the booster version is F9 v1.1. The result is 2928.4 kg, as shown in the query result.

```
query_4 = """
SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass
FROM SpaceX_Missions
WHERE "Booster_Version" = 'F9 v1.1';
"""
result_4 = pd.read_sql_query(query_4, con)
result_4.head()
```

	Average_Payload_Mass
0	2928.4

First Successful Ground Landing Date

We used the MIN function to find the earliest date where the Landing Outcome was a Success. The first successful landing outcome occurred on July 22, 2018, as shown in the query result

```
query_5 = """
SELECT MIN("Date") AS First_Successful_Landing
FROM SpaceX_Missions
WHERE "Landing_Outcome" = 'Success';
"""
result_5 = pd.read_sql_query(query_5, con)
result_5.head()
```

	First_Successful_Landing
0	2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We applied a WHERE clause to filter for boosters with successful drone ship landings and used the AND condition to restrict the payload mass between 4000 kg and 6000 kg. The query returned the following booster versions:

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

```
query_6 = """
SELECT DISTINCT "Booster_Version"
FROM SpaceX_Missions
WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;
"""
result_6 = pd.read_sql_query(query_6, con)
result_6.head()
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

We calculated the total number of successful and failed mission outcomes. Out of all missions, there were 61 successes and 10 failures, as shown below.

```
query_7 = """
SELECT
    CASE
        WHEN "Landing_Outcome" LIKE 'Success%' THEN 'Success'
        WHEN "Landing_Outcome" LIKE 'Failure%' THEN 'Failure'
        ELSE 'Other'
    END AS Mission_Status,
    COUNT(*) AS Total_Count
FROM SpaceX_Missions
GROUP BY Mission_Status
ORDER BY Total_Count DESC;
"""

result_total = pd.read_sql_query(query_total, con)
result_total.head()
```

	Mission_Status	Total_Count
0	Success	61
1	Other	30
2	Failure	10

Boosters Carried Maximum Payload

We identified the booster versions that carried the maximum payload mass of 15,600 kg using a subquery. The query compares the payload mass of each record to the maximum payload mass in the dataset and retrieves the relevant booster names. The boosters that achieved this are listed below

```
query_8 = """
SELECT "Booster_Version", "PAYLOAD_MASS__KG_"
FROM SpaceX_Missions
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SpaceX_Missions
)
ORDER BY "Booster_Version";
"""
result_8 = pd.read_sql_query(query_8, con)
result_8
```

	Booster_Version	PAYLOAD_MASS__KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

We retrieved the records of failed drone ship landings in 2015. The query uses the SUBSTR function to extract the month and year from the date and classify them into their respective month names. The records show two failed landings in January and April at the CCAFS LC-40 launch site, as summarized below

```
query_9 = """
SELECT
    CASE
        WHEN SUBSTR("Date", 6, 2) = '01' THEN 'January'
        WHEN SUBSTR("Date", 6, 2) = '02' THEN 'February'
        WHEN SUBSTR("Date", 6, 2) = '03' THEN 'March'
        WHEN SUBSTR("Date", 6, 2) = '04' THEN 'April'
        WHEN SUBSTR("Date", 6, 2) = '05' THEN 'May'
        WHEN SUBSTR("Date", 6, 2) = '06' THEN 'June'
        WHEN SUBSTR("Date", 6, 2) = '07' THEN 'July'
        WHEN SUBSTR("Date", 6, 2) = '08' THEN 'August'
        WHEN SUBSTR("Date", 6, 2) = '09' THEN 'September'
        WHEN SUBSTR("Date", 6, 2) = '10' THEN 'October'
        WHEN SUBSTR("Date", 6, 2) = '11' THEN 'November'
        WHEN SUBSTR("Date", 6, 2) = '12' THEN 'December'
    END AS Month_Name,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM
    SpaceX_Missions
WHERE
    "Landing_Outcome" LIKE '%Failure (drone ship)%'
    AND SUBSTR("Date", 1, 4) = '2015';
"""
result_9 = pd.read_sql_query(query_9, con)
result_9
```

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

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

We retrieved the Landing Outcomes and their corresponding counts from the dataset.

- The WHERE clause was used to filter records between 2010-06-04 and 2017-03-20.
- The GROUP BY clause grouped the data by landing outcome.
- The ORDER BY clause sorted the landing outcomes in descending order of their counts.
- This approach provided a clear ranking of the different landing outcomes within the specified date range.

```
query_10 = """
SELECT
    "Landing_Outcome",
    COUNT(*) AS Outcome_Count
FROM
    SpaceX_Missions
WHERE
    "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    "Landing_Outcome"
ORDER BY
    Outcome_Count DESC;
"""
result_10 = pd.read_sql_query(query_10, con)
result_10
```

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

Section 3

Launch Sites Proximities Analysis

SpaceX Launch Site Locations in the United States

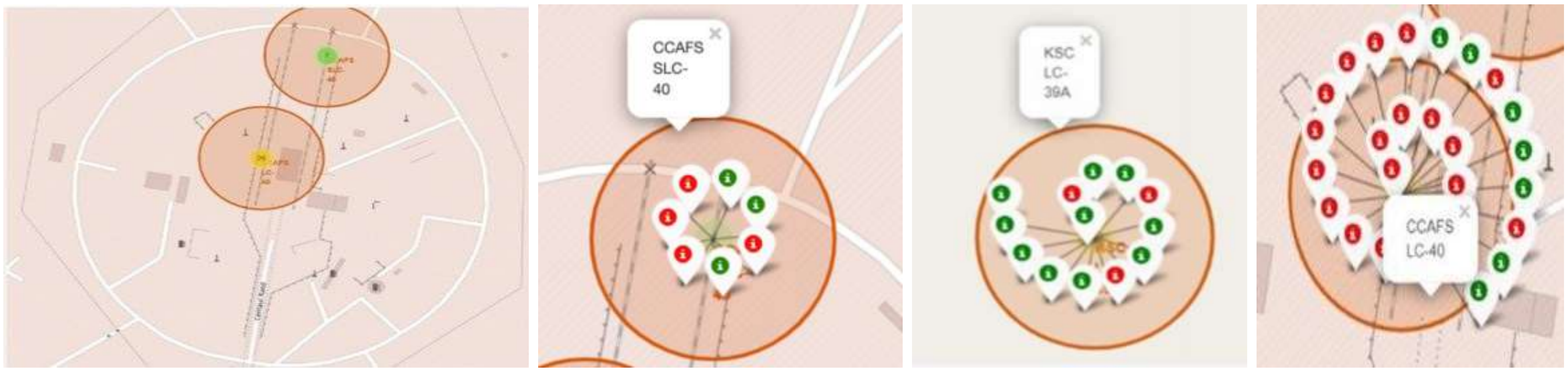
The map displays the locations of SpaceX launch sites in the United States:

- VAFB SLC-4E (Vandenberg, California)
- KSC LC-39A (Kennedy Space Center, Florida)
- CCAFS LC-40 (Cape Canaveral, Florida).
- These sites are strategically located to support various launch missions due to their geographic positioning and technical capabilities.

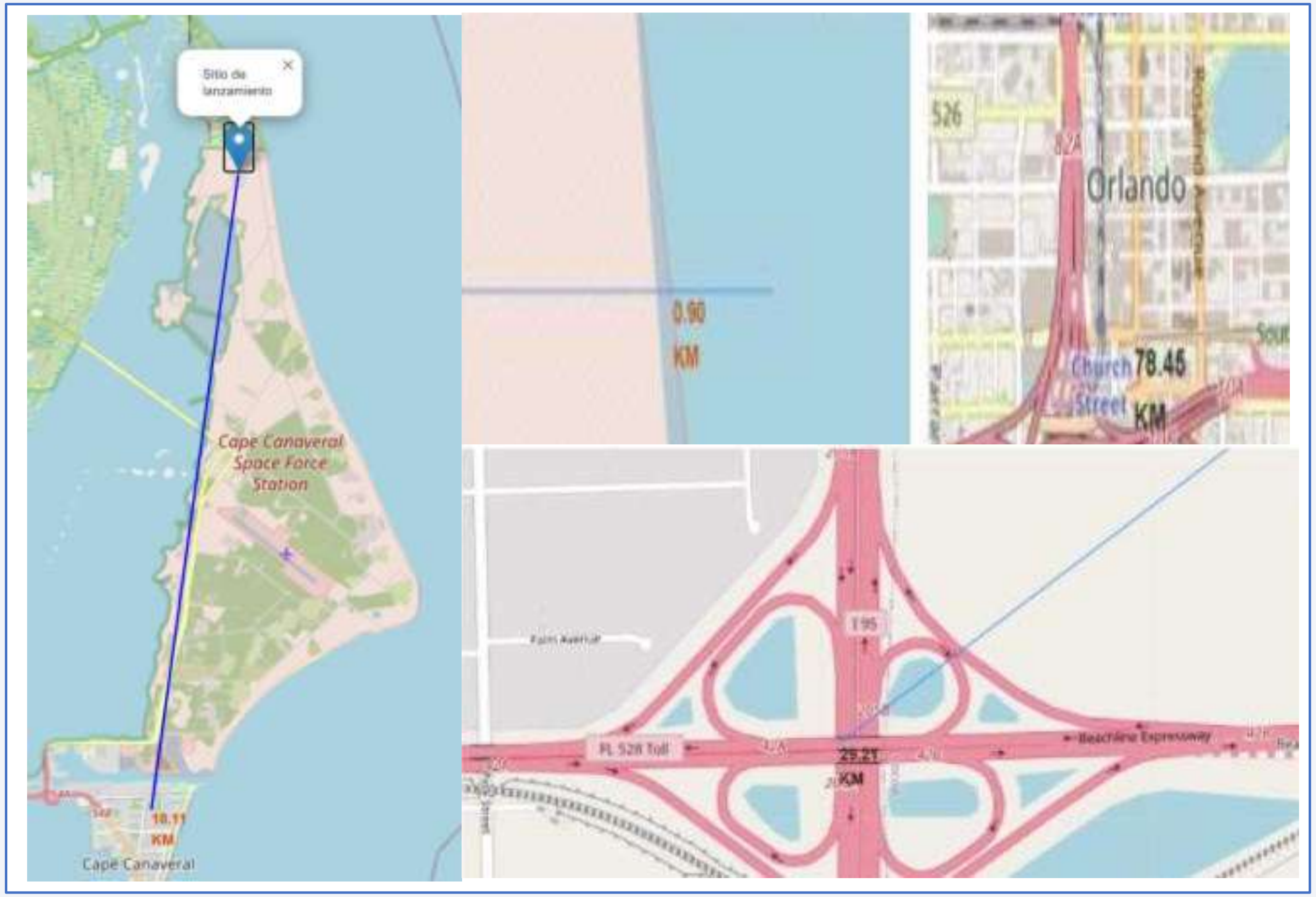
SpaceX Launch Outcomes: Successes and Failures by Site

The map shows SpaceX launch outcomes using color-coded markers:

- Green for successful launches.
- Red for failed launches.
- Florida Launch Sites (CCAFS LC-40, KSC LC-39A) show a mix of successes and failures with dense activity.
- California Launch Site (VAFB SLC-4E) also has both outcomes but fewer launches.



Launch Site Proximity Analysis: Cape Canaveral



The map highlights Cape Canaveral Space Force Station and its proximities:

- Railway and highway networks are displayed for logistical access.
- Coastline proximity shows launch sites' strategic placement near the ocean.

Measured distances include:

- 18.11 KM to Cape Canaveral.
- 0.90 KM to the coastline.
- Key intersections near FL 528 Toll and I-95 highway.

Key Findings:

- The location offers optimal connectivity to transportation networks and the coastline, enabling efficient logistics for launches and safety during takeoffs.

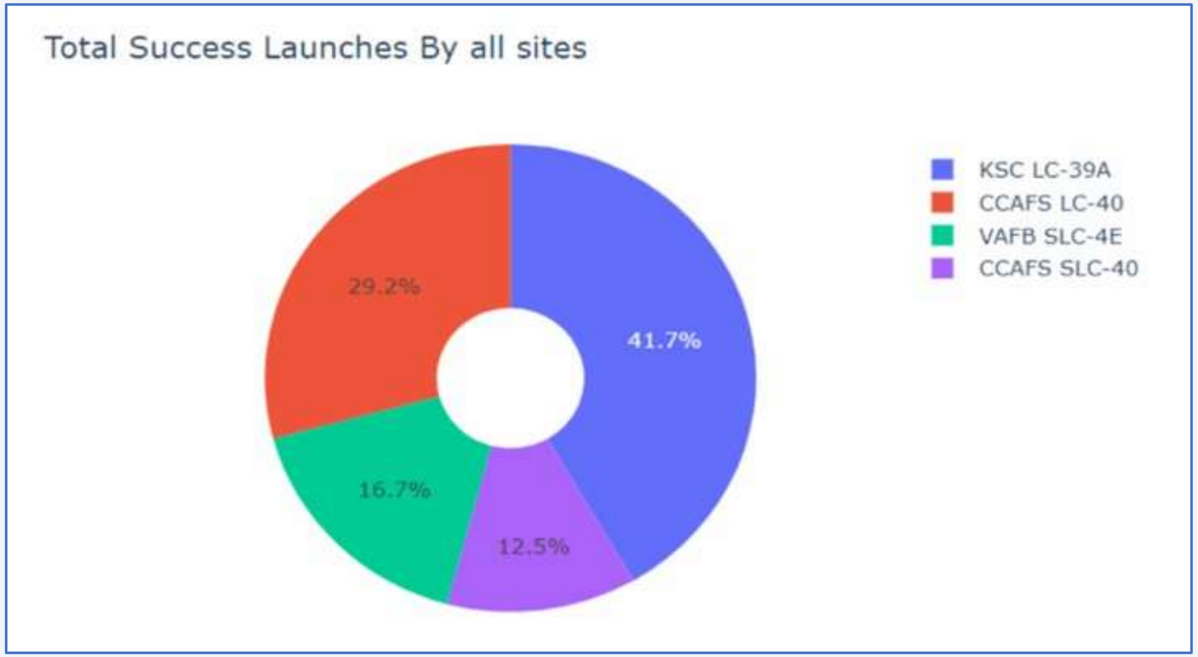
Section 4

Build a Dashboard with Plotly Dash

Launch Success Distribution Across All Sites

The pie chart illustrates the distribution of successful launches across SpaceX's launch sites:

- KSC LC-39A (Blue): 41.7% of total successful launches, the highest contribution.
- CCAFS LC-40 (Red): 29.2% of successful launches.
- VAFB SLC-4E (Green): 16.7% of successful launches.
- CCAFS SLC-40 (Purple): 12.5% of successful launches.

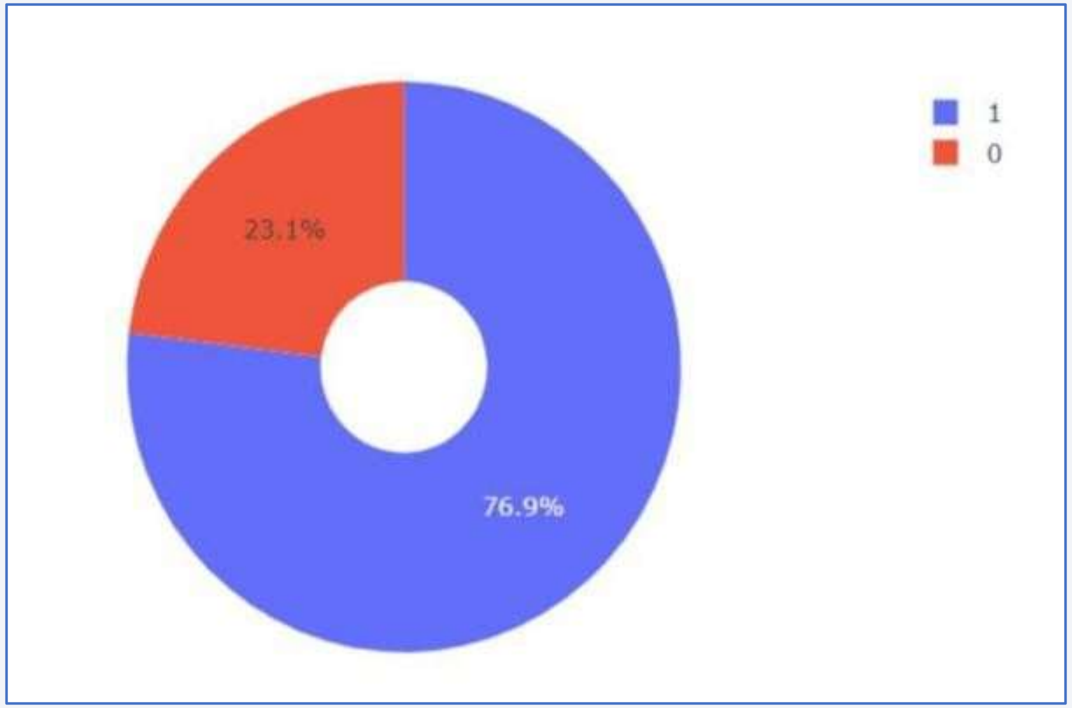


Key Findings:

- KSC LC-39A plays a dominant role in successful launches, contributing nearly half of the total.

KSC LC-39A: Launch Success and Failure Ratio

The pie chart represents the **success and failure rates** for the launch site **KSC LC-39A**, which achieved the **highest launch success ratio** among all SpaceX launch sites.



Key Findings:

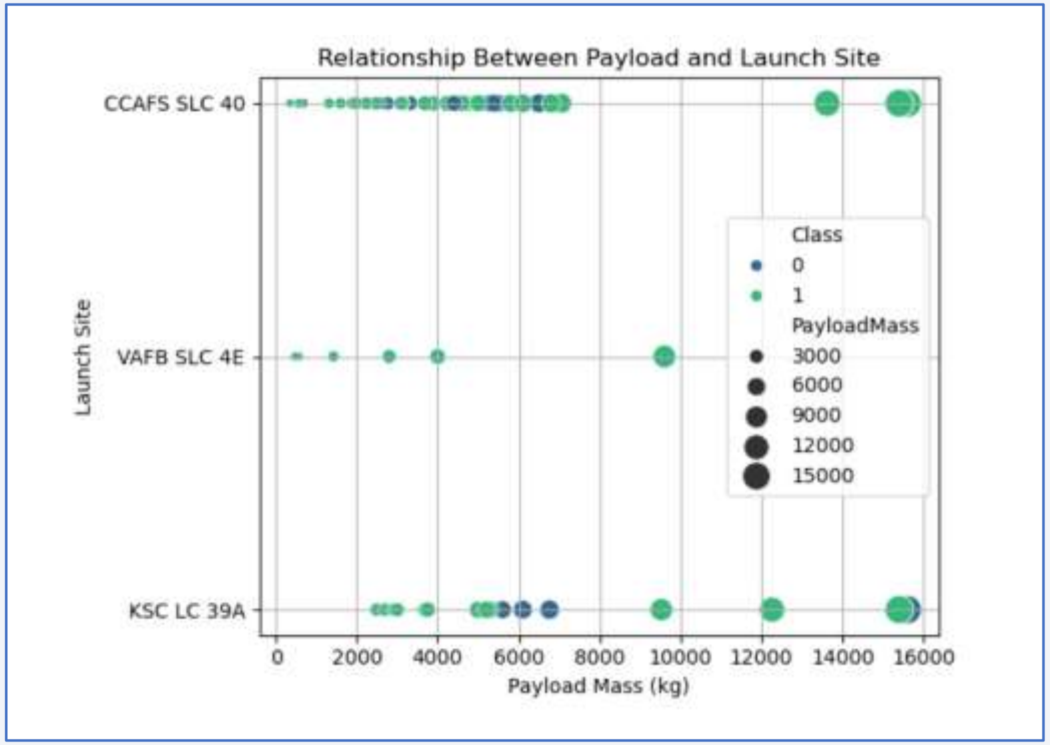
- **Success Rate** (Blue): **76.9%** of launches were successful.
- **Failure Rate** (Red): **23.1%** of launches resulted in failure.
- This highlights KSC LC-39A's **consistent performance** as the leading site for successful missions, reinforcing its strategic importance in SpaceX's operations.

Payload vs. Launch Outcome Across All Sites

The scatter plot illustrates the relationship between payload mass (kg) and launch outcomes at three launch sites: CCAFS SLC-40, VAFB SLC-4E, and KSC LC-39A.

Key Findings:

- Green markers indicate successful launches; blue markers indicate failures.
- CCAFS SLC-40 managed the largest payloads (up to 16,000 kg) with a high success rate.
- VAFB SLC-4E shows successes for payloads under 10,000 kg.
- KSC LC-39A has mixed results for payloads between 6,000 kg and 16,000 kg.
- The analysis highlights CCAFS SLC-40's superior performance, particularly for heavier payloads.



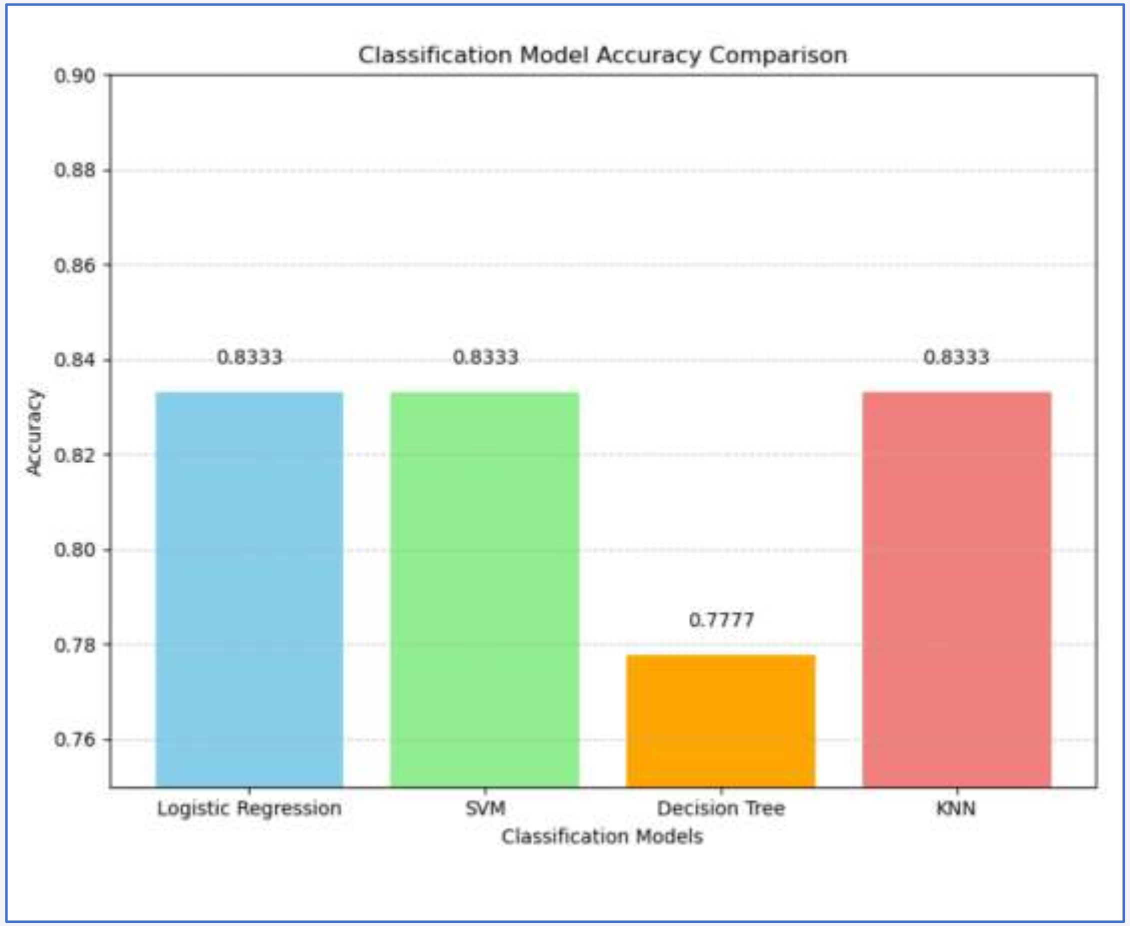
Section 5

Predictive Analysis (Classification)

Classification Accuracy

The bar chart compares the accuracy of the built classification models:

- **Logistic Regression**, **SVM**, and **KNN** achieved the highest accuracy of **83.33%**.
- The **Decision Tree** model had the lowest accuracy at **77.77%**.



Key Findings:

- Logistic Regression, SVM, and KNN performed equally well, while Decision Tree showed lower accuracy.

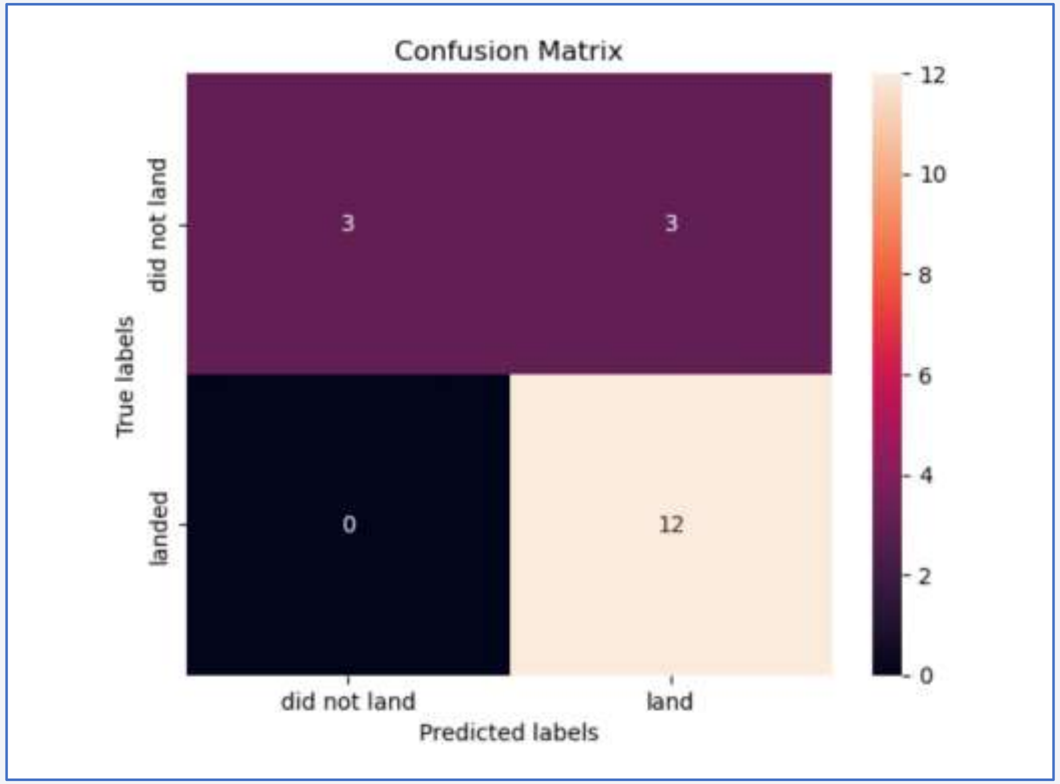
Confusion Matrix

The confusion matrix shows the performance of the **Logistic Regression** model:

- **True Positives (landed correctly)**: 12
- **True Negatives (did not land correctly)**: 3
- **False Positives (incorrectly predicted as landed)**: 3
- **False Negatives**: 0

Summary:

- The model correctly predicts all "landed" instances (no false negatives) but has a slight overestimation with 3 false positives. Overall, it performs well with an **83% accuracy**, making it the best-performing model.



Conclusions

1. Launch Site Insights:
    - Key launch sites: CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E.
    - CCAFS SLC-40 handles the heaviest payloads with consistent success.
2. Success:
    - KSC LC-39A achieved the highest success rate at 76.9%.
3. Model Performance:
    - Logistic Regression is the best-performing model with an accuracy of 83.3%.
4. Confusion Matrix:
    - The model effectively predicts "landed" outcomes with 12 true positives and minimal errors.

Key Insight

- The analysis highlights the role of launch sites and payload capacity in mission success, with Logistic Regression providing reliable predictions for launch outcomes.

Thank you!