IBM Developer
SKILLS NETWORK

SPACEX

Winning Space Race with Data Science

HAMMAD QASIM
28.12.2024

Outline

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

Executive Summary

Summary of methodologies

1. Data collection
2 Data wrangling
3. Exploratory Data Analysis with Data Visualization
4 Exploratory Data Analysis with SQL
5 Building an interactive map with Folium
6. Building a Dashboard with Plotly Dash
7. Predictive analysis (Classification)

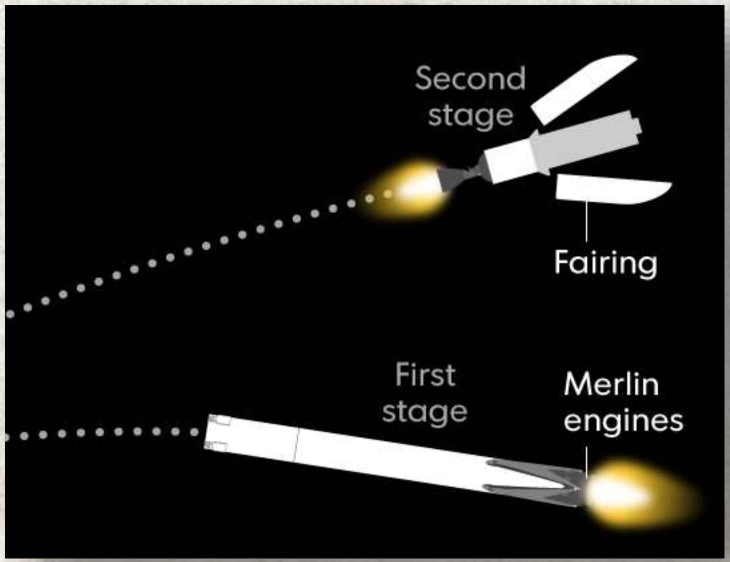
Summary of all results

- Exploratory Data Analysis results
- Interactive analytics demo (screenshots)
- Predictive data analysis results

Introduction

- Project background and context

SpaceX has transformed the commercial space industry by drastically reducing the cost of space travel. Unlike traditional providers, which charge upwards of $165 million per launch, SpaceX offers Falcon 9 rocket launches for $62 million. This affordability is made possible by their innovative reuse of the rocket’s first stage. The ability to predict whether the first stage will successfully land directly impacts the overall launch cost. In this project, we leverage public data and machine learning techniques to determine the likelihood of SpaceX reusing the first stage.



Introduction

Problems you want to find answers

- What factors influence the successful landing of the Falcon 9 rocket's first stage?
- How can we gather and preprocess relevant data from public sources about SpaceX launches?
- What machine learning models are most effective for predicting the reuse of the first stage?
- How can we evaluate the performance of different models to identify the optimal one?
- What insights can be derived from the data to inform Space Y's bidding strategy against SpaceX?
- How can we visualize the data and findings to effectively communicate results to stakeholders?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - SpaceX Rest API calls
  - Webscraping of Wikipedia
- Perform data wrangling
  - Data filtration
  - Filling missing data
  - One Hot Encoding (Predictive Analysis preparation)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- The data was collected using two methods:
  1. REST API ca l
  2. Webscraping
- The SpaceX REST API was accessed via HTTP GET requests to retrieve detailed launch data, including metrics such as launch success, payload mass, and orbit type.
- The JSON responses were parsed and structured into data tables for analysis.
- Additionally, historical launch records were gathered by scraping the Wikipedia page for Falcon 9 and Falcon Heavy launches.
- HTML content was parsed using web scraping libraries, with specific focus on extracting and cleaning tabular data, ensuring it was formatted consistently for further processing.

Data Collection - SpaceX API

- **API Endpoint**
  SpaceX REST API was used to retrieve launch data
- **Data Retrieval**
  Sent HTTP GET requests to SpaceX API endpoints
- **JSON Parsing**
  Extracted fields such as launch outcome, rocket type, payload mass, and orbit
- **Data Storage**
  Organized the collected data into structured tables or DataFrames for analysis

GitHub: SpaceX API calls notebook

Data Collection - SpaceX API

1. **Request Rocket Launch Data from SpaceX API**
   Initiate an API call to SpaceX to obtain data related to rocket launches
2. **Decode the JSON Response (`.json()`) and Convert It into a DataFrame (`.json_normalize()`)**
   Use the .json() method to parse the API response and employ `.json_normalize()` to flatten the nested JSON structure into a tabular format
3. **Extract Relevant Launch Information Using Custom Functions**
   Apply custom functions to retrieve specific details about the launches from the API response
4. **Organize the Data into a Dictionary**
   Structure the retrieved data into a dictionary, where the keys correspond to the data categories, and the values hold the respective launch details
5. **Create a DataFrame from the Dictionary**
   Convert the dictionary into a pandas DataFrame for easier manipulation and analysis
6. **Filter Data to Include Only Falcon 9 Launches**
   Apply filters to the DataFrame to select only the records related to Falcon 9 rocket launches
7. **Handle Missing Payload Mass Values by Replacing Them with the Column's Mean**
   Fill in any missing values in the Payload Mass column by using the calculated mean of the column to ensure consistency
8. **Export the Data to CSV**
   After cleaning and organizing the data, export the final dataset to a CSV file for further use or analysis

Data Collection - SpaceX API

Start

↓

Define Objectives

↓

Identify API Endpoint

↓

Set Up Tools (Python, Libraries)

↓

Make REST API Calls

↓

Process and Validate Data

↓

Store Data

↓

Visualize Data

↓

End

Data Collection - Scraping

- **Source Website**
  Historical data was gathered from Wikipedia
- **HTML Parsing**
  Used web scraping libraries like `BeautifulSoup` to parse the HTML content
- **Table Extraction**
  Extracted tabular data from the webpage containing launch details
- **Data Cleaning**
  Processed and formatted the raw data into consistent formats

[**GitHub:** Webscraping notebook]

Data Collection - Scraping

1. **Request Falcon 9 Launch Data from Wikipedia**
   Make an HTTP request to the Wikipedia page containing the Falcon 9 launch data to retrieve the page content
2. **Create a `BeautifulSoup` Object from the HTML Response**
   Use the `BeautifulSoup` library to parse the HTML response, creating a `BeautifulSoup` object for further processing
3. **Extract Column Names from the HTML Table Header**
   Parse the header of the HTML table to extract the column names, which will be used as labels for the data
4. **Collect Data by Parsing HTML Tables**
   Extract the relevant data from the HTML tables by navigating through the rows and columns to gather the launch details
5. **Construct Data into a Dictionary**
   Organize the extracted data into a dictionary format, where keys represent column names, and values represent the corresponding data entries
6. **Create a `DataFrame` from the Dictionary**
   Use the pandas library to convert the dictionary into a structured `DataFrame`, enabling easy manipulation and analysis
7. **Export Data to CSV**
   Once the data is organized and structured in a `DataFrame`, export it to a CSV file for storage or further analysis

Data Collection - Scraping

Start

↓

Define Objectives

↓

Identify Target Website

↓

Analyze Website Structure

↓

Set Up Tools (Python, Libraries)

↓

Write Scraping Code

↓

Process and Clean Data

↓

Store Data

↓

Ensure Compliance

↓

End

Data Wrangling

- The dataset includes various instances where the booster failed to land successfully. In some cases, a landing attempt was made but ended in failure due to an incident.
- These outcomes are generally converted into training labels, with "1" representing a successful landing and "0" indicating a failed landing.

1. **Exploratory Data Analysis**
   Begin by examining the dataset to understand its structure, identify patterns, and detect any inconsistencies or missing values
2. **Training Labels**
   Create training labels based on relevant outcomes, such as the success or failure of a mission, to facilitate machine learning model training
3. **Number of Launches per Site**
   Aggregate the data to calculate how many launches occurred at each launch site
4. **Orbit Type Occurrences**
   Count the number of launches for each orbit type to analyze the distribution and occurrence of each orbit
5. **Mission Outcome Analysis per Orbit Type**
   Determine the number and occurrence of mission outcomes (successful or failed) for each orbit type to explore any correlations
6. **Landing Outcome Label from Outcome Column**
   Derive a new label representing the landing outcome, based on the "Outcome" column, categorizing it as successful or unsuccessful
7. **Export to CSV**
   Once wrangling is complete, export the cleaned and structured data to a CSV file for further analysis or use

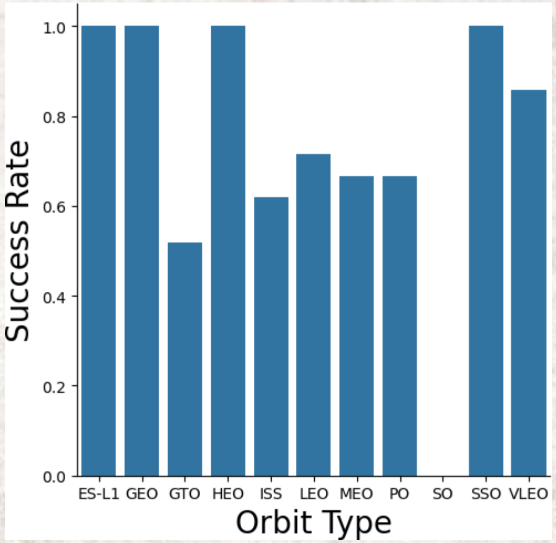
GitHub: Data wrangling notebook

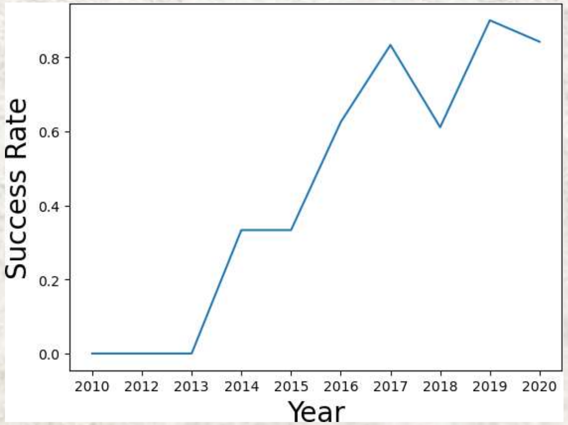
Data Wrangling

Start
↓
Import Raw Data
↓
Inspect Data Quality
↓
Handle Missing Data
↓
Standardize and Clean Data
↓
Transform and Enrich Data
↓
Filter and Organize Data
↓
Merge or Combine Datasets
↓
Validate and Save Data
↓
End

EDA with Data Visualization

- Charts
  - **Scatter point chart**
    Helps examine relationships and identify outliers
    - Flight Number vs. Payload Mass
    - Flight Number vs. Launch Site
    - Payload Mass vs. Launch Site
    - Flight Number vs. Orbit Type
  - **Bar chart**
    Uses rectangular bars to represent data magnitude of different categories or groups
    - Orbit Type vs. Success Rate
  - **Line chart**
    Dispaly trends over time
    - Launch Success Yearly Trend





[GitHub: EDA with Data Visualization notebook]

EDA with SQL

Performed SQL queries

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string ‘CCA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the records of failed landing outcomes in drone ship, their booster versions and launch site for the months in year 2015.
- Ranking 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

[**GitHub:** EDA with SQL notebook]()

Build an Interactive Map with Folium

Markers for All Launch Sites

- Placed a marker with a circle, a popup label, and a text label at the NASA Johnson Space Center, using its latitude and longitude as the starting point.
- Added markers with circles, popup labels, and text labels for all launch sites, displaying their geographic locations and highlighting their proximity to the Equator and nearby coasts.

Coloured Markers

- Used color-coded markers to represent launch outcomes:
  - Green for successful launches
  - Red for failures.
- These markers were clustered to help visualize which sites have higher success rates.

Distances from Launch Site

- Added colored lines to illustrate distances from CCAFS SLC-40 launch site to nearby features: railway, highway, coastline, and the closest city

GitHub: Interactive Map notebook

Build a Dashboard with Plotly Dash

Dropdown for Launch Sites

- Implemented a dropdown menu to allow users to select a specific launch site.

Pie Chart for Launch Success

- Created a pie chart displaying the total number of successful launches across all sites, as well as the breakdown of successes and failures for a selected launch site.

Payload Mass Range Slider

- Introduced a slider to let users adjust the range of payload mass.

Scatter Plot of Payload Mass and Success Rate by Booster Version

- Designed a scatter plot to visualize the relationship between payload mass and launch success, categorized by different booster versions.

GitHub: Plotly Dashboard code

Predictive Analysis (Classification)

1. **Create a NumPy Array from the "Class" Column**
   Extract the "Class" column from the dataset and convert it into a NumPy array for model training.
2. **Standardize the Data Using `StandardScaler`**
   Standardize the dataset to ensure that all features have the same scale, improving model performance.
   Apply `StandardScaler` to fit and transform the data.
3. **Split the Data into Training and Testing Sets Using `train_test_split`**
   Split the dataset into training and testing subsets to evaluate the models' performance on unseen data
4. **Create a `GridSearchCV` Object**
   Set up a `GridSearchCV` object with 10-fold cross-validation to identify the best parameters for the models
5. **Apply `GridSearchCV` on Logistic Regression (LogReg), SVM, Decision Tree, and KNN Models**
   Use `GridSearchCV` to optimize hyperparameters for Logistic Regression, Support Vector Machine (SVM), Decision Tree, and k-Nearest Neighbors (KNN) models
6. **Calculate Model Accuracy Using `.score()` Method on Test Data**
   After training the models, assess their accuracy by evaluating their performance on the test data using the `.score()` method
7. **Examine the Confusion Matrix for Each Model**
   Evaluate the models' performance by examining the confusion matrix to identify true positives, false positives, true negatives, and false negatives
8. **Determine the Best Performing Model**
   Compare the models' accuracy and confusion matrices to identify which model performs best for the given data

GitHub: Predictive Analysis notebook

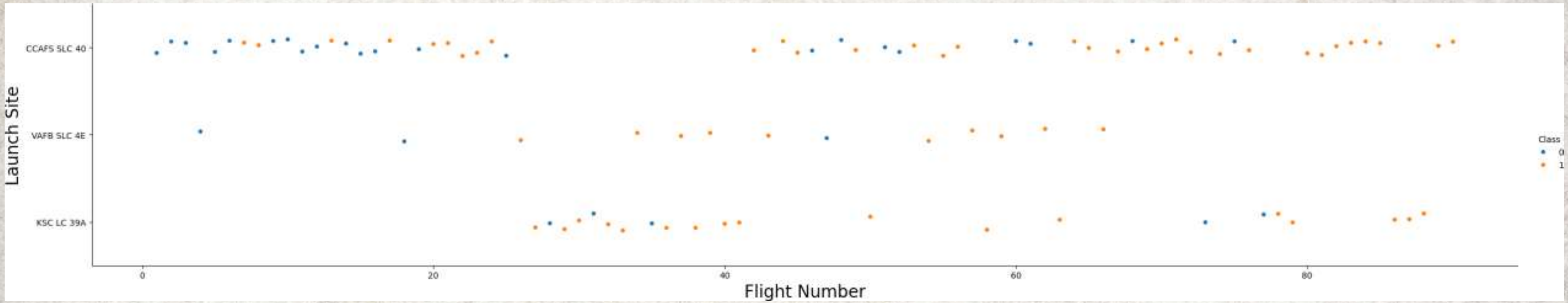
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

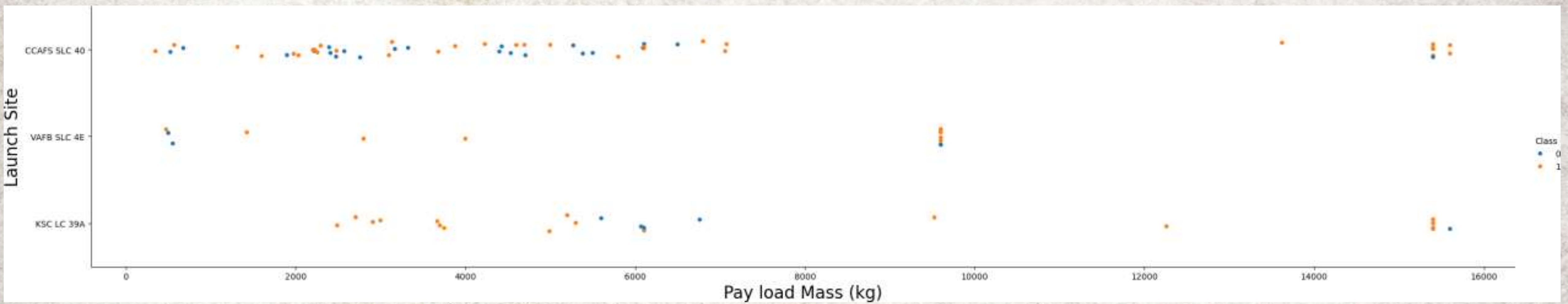
Flight Number vs. Launch Site



Explanation

- Most of the intial launches failed while almost all of the latest ones succeded
- Most of the launches were conducted from CCAFS SLC 40 launch site
- Launch sites VAFB SLC 4E and KSC LC 39A have higher success rates compared to CCAFS SLC 40
- Data indicates that probability of sucessful launch increases with every new launch

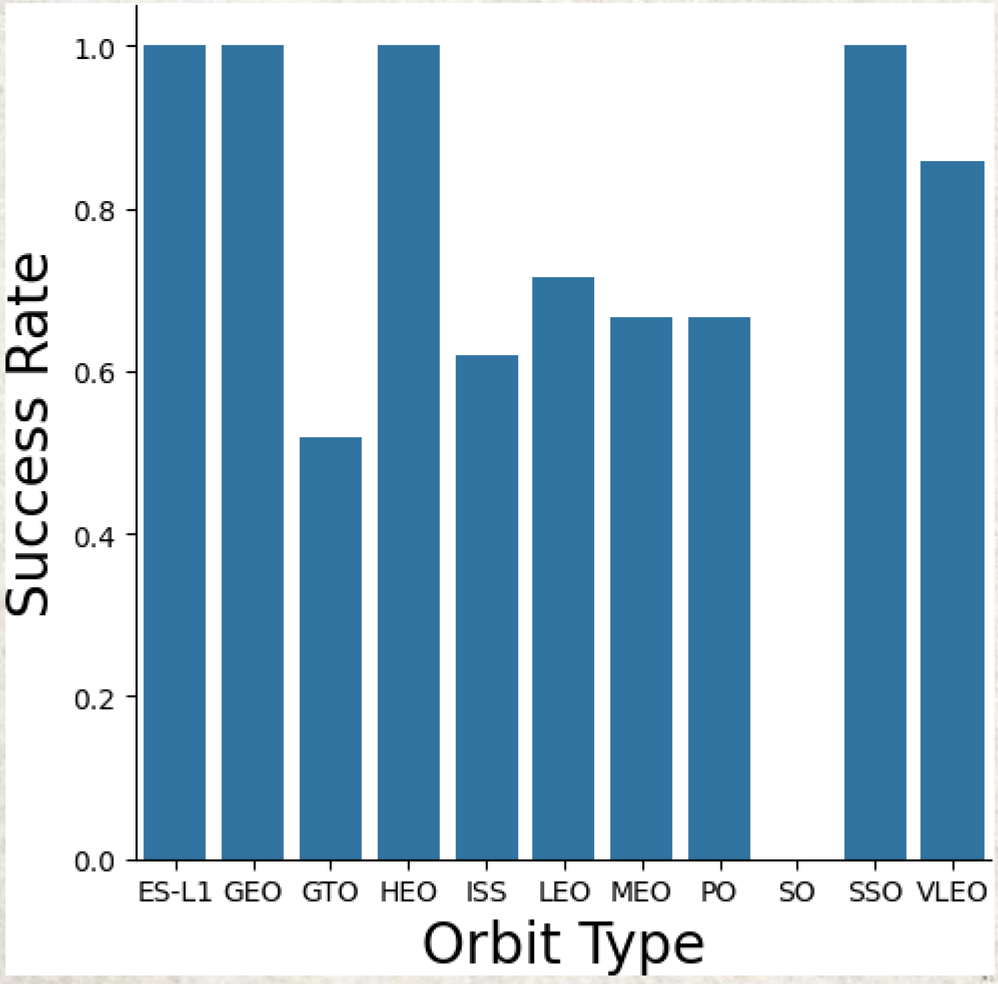
Payload vs. Launch Site



Explanation

- Payload mass plays a significant role in launch success, with higher payload masses generally associated with increased success rates for all launch sites
- The majority of launches carrying payloads exceeding 7000 kg achieved success
- For payloads under 5500 kg, KSC LC-39A stands out with a perfect 100% success rate

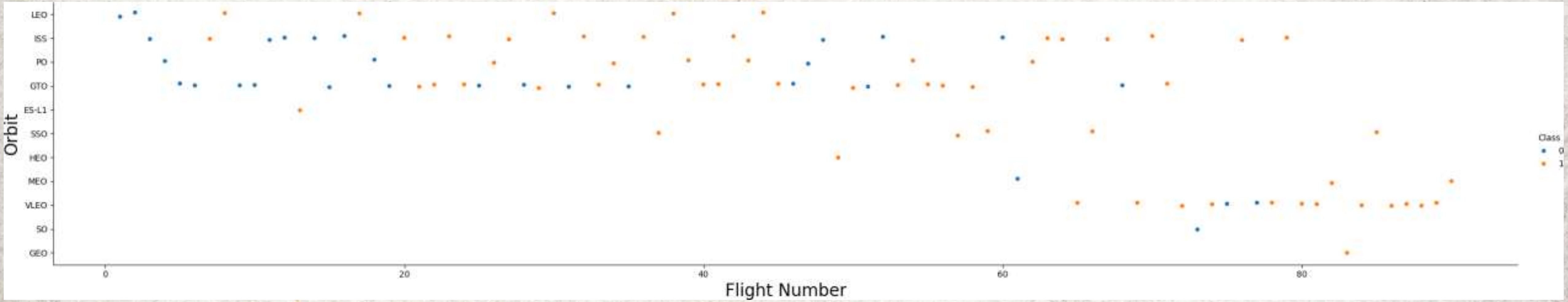
Success Rate vs. Orbit Type



Explanation

- **100% success rate**
  ES-L1, GEO, HEO and SSO
- **Variable success rate**
  GTO, ISS, LEO, MEO, PO and VLEO
- **0% sucess rate**
  SO

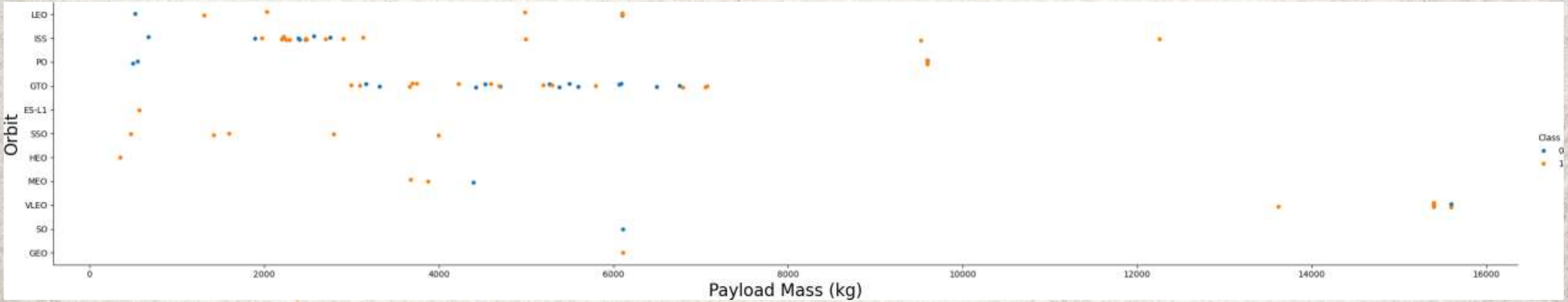
Flight Number vs. Orbit Type



Explanation

- Higher orbits tend to be selected with increasing flight number

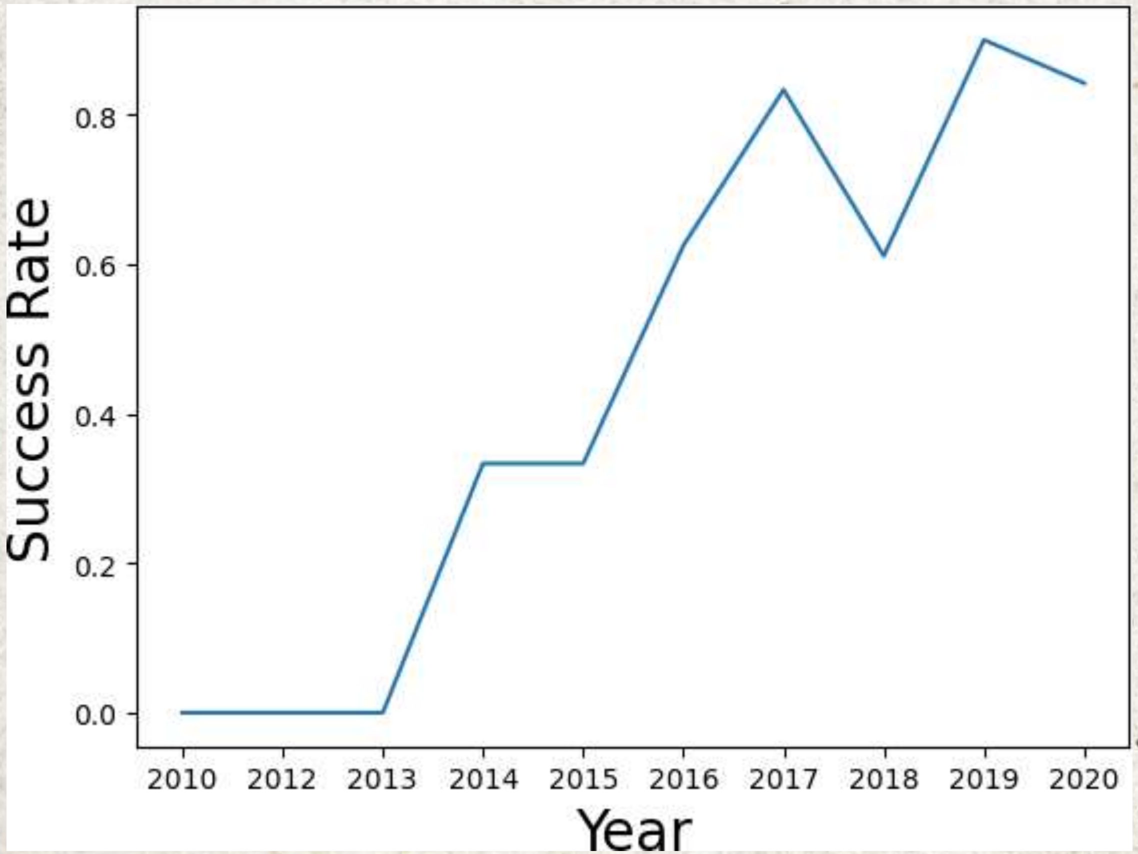
Payload vs. Orbit Type



Explanation

- Higher orbits tend to be selected for heavier payloads

Launch Success Yearly Trend



Explanation

- Success rate increased significantly between 2010 and 2020
- Small decreases occured in 2018 and 2020

All Launch Site Names

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

```
%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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string '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

Total Payload Mass

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

```
%sql SELECT sum(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = "NASA (CRS)"
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE "F9 v1.1%"
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

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

Hint:Use min function

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%Success%"
```

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

MIN(Date)
2015-12-22

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 DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome  = "Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

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

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

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

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql SELECT Booster_Version, PAYLOAD
FROM (
    SELECT Booster_Version, SUM(PAYLOAD_MASS__KG_) AS PAYLOAD
    FROM SPACEXTABLE
    GROUP BY Booster_Version
) AS Subquery
WHERE PAYLOAD = (
    SELECT MAX(SUM_PAYLOAD)
    FROM (
        SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD
        FROM SPACEXTABLE
        GROUP BY Booster_Version
    ) AS InnerSubquery
);
```

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

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

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 strftime('%m', Date) AS Month, Mission_Outcome, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Date LIKE "2015-%" AND Landing_Outcome = "Failure (drone ship)"
```

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

Month	Mission_Outcome	Landing_Outcome	Booster_Version	Launch_Site
01	Success	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Success	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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 SPACEXTABLE
WHERE Date > '2010-06-04' AND Date < '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
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

SpaceX Launch Sites Locations Markers

- Most launch sites are located near the equator due to the Earth's rotational speed. At the equator, the surface of the Earth moves at approximately 1,670 km/hour. This means any spacecraft launched from this region already carries that rotational speed, thanks to inertia. This initial velocity helps the spacecraft achieve the necessary speed to remain in orbit.
- Additionally, launch sites are typically situated close to the coast. Launching rockets over the ocean reduces the risk of debris or potential explosions affecting populated areas, ensuring greater safety.

Colour-labeled Launch Markers

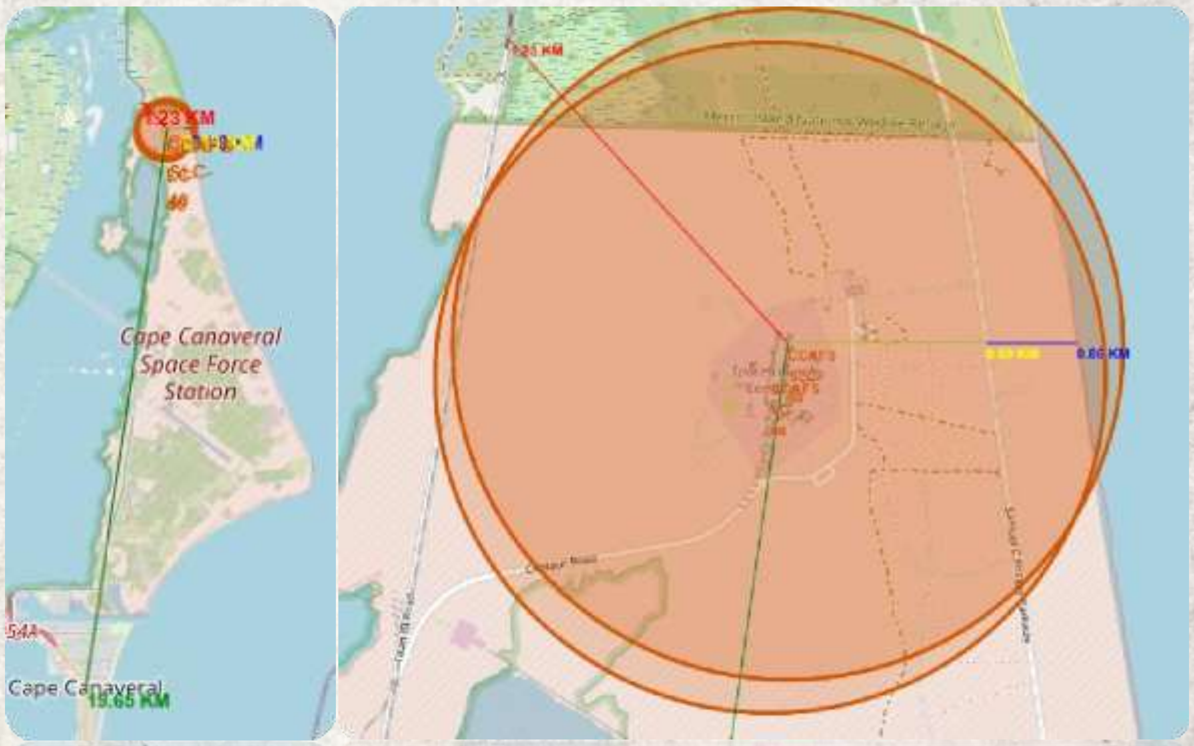
Explanation

• Launch Site	KSC LC-39A
• Successful launch	**Green**
• Failed launch	**Red**
• Success ratio	76.9%

Distance from the launch site CCAFS SLC-40 to its proximities

Explanation

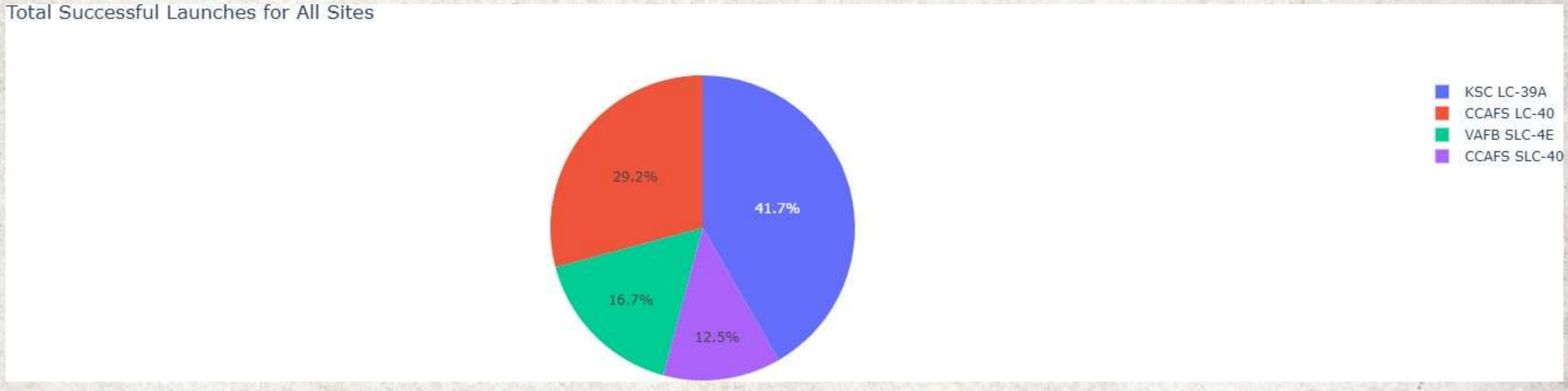
- Launch Site CCAFS SLC-40
- From the visual analysis of the launch site CCAFS SLC-40
    - it's relative close to railway (1.23 km)
    - it's relative close to highway (0.59 km)
    - it's relative close to coastline (0.56 km)
- The launch site CCAFS SLC-40 is also relative close to city Cape Canaveral (19.25 km), however it should be sufficient to ensure their relative safety.



Section 4

Build a Dashboard with Plotly Dash

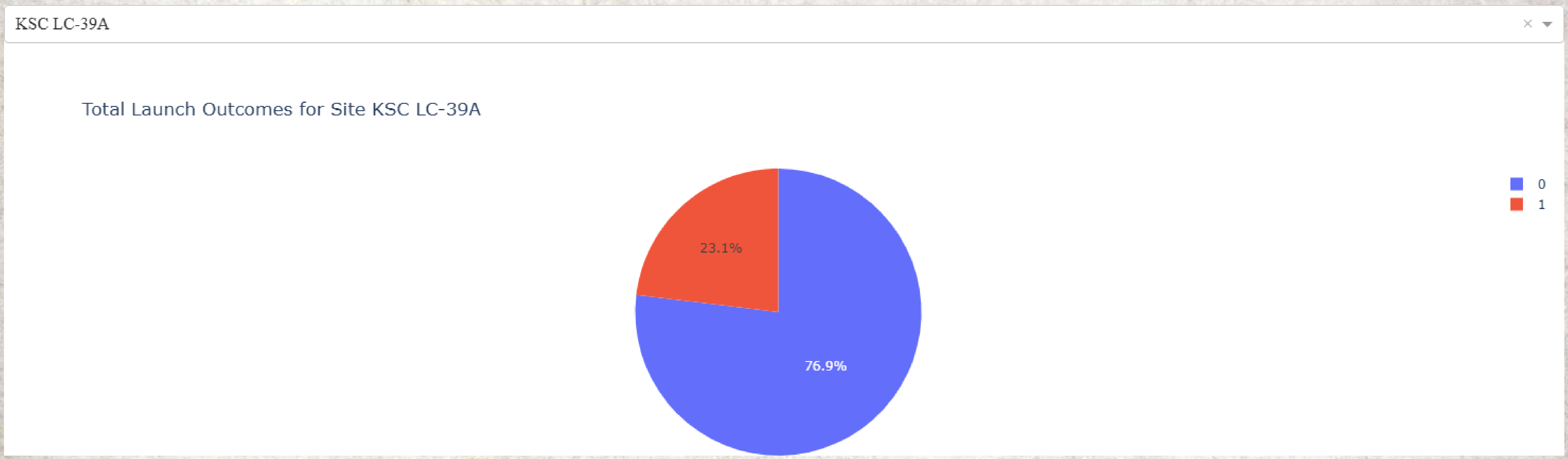
Total Success Launches by Site



Explanation

- Launch site KSC LC-39A has the most successful launches
- Launch site CCAFS SLC-40 has the fewest successful launches

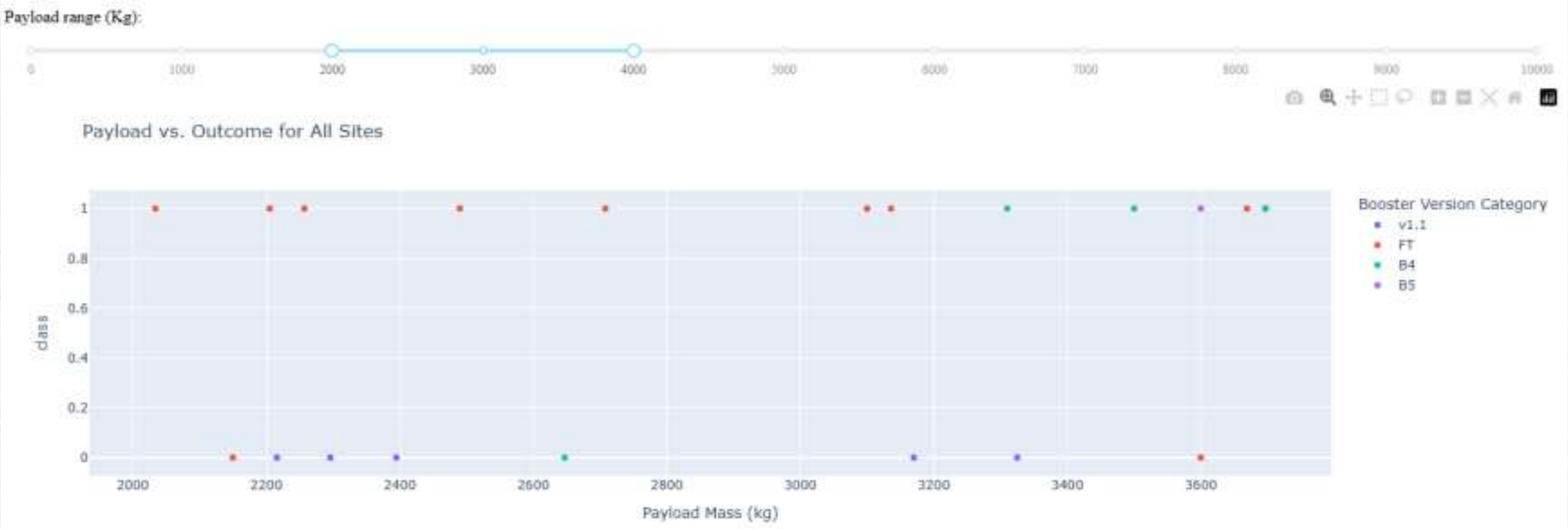
Launch Site with Highest Launch Success Ratio



Explanation

- Launch site KSC LC-39A has the highest success ratio (10 successful, 3 failed).

Payload vs. Launch Outcome Scatter Plot for All Sites



Explanation

- Payloads between 2000 and 4000 kg have the highest success rate

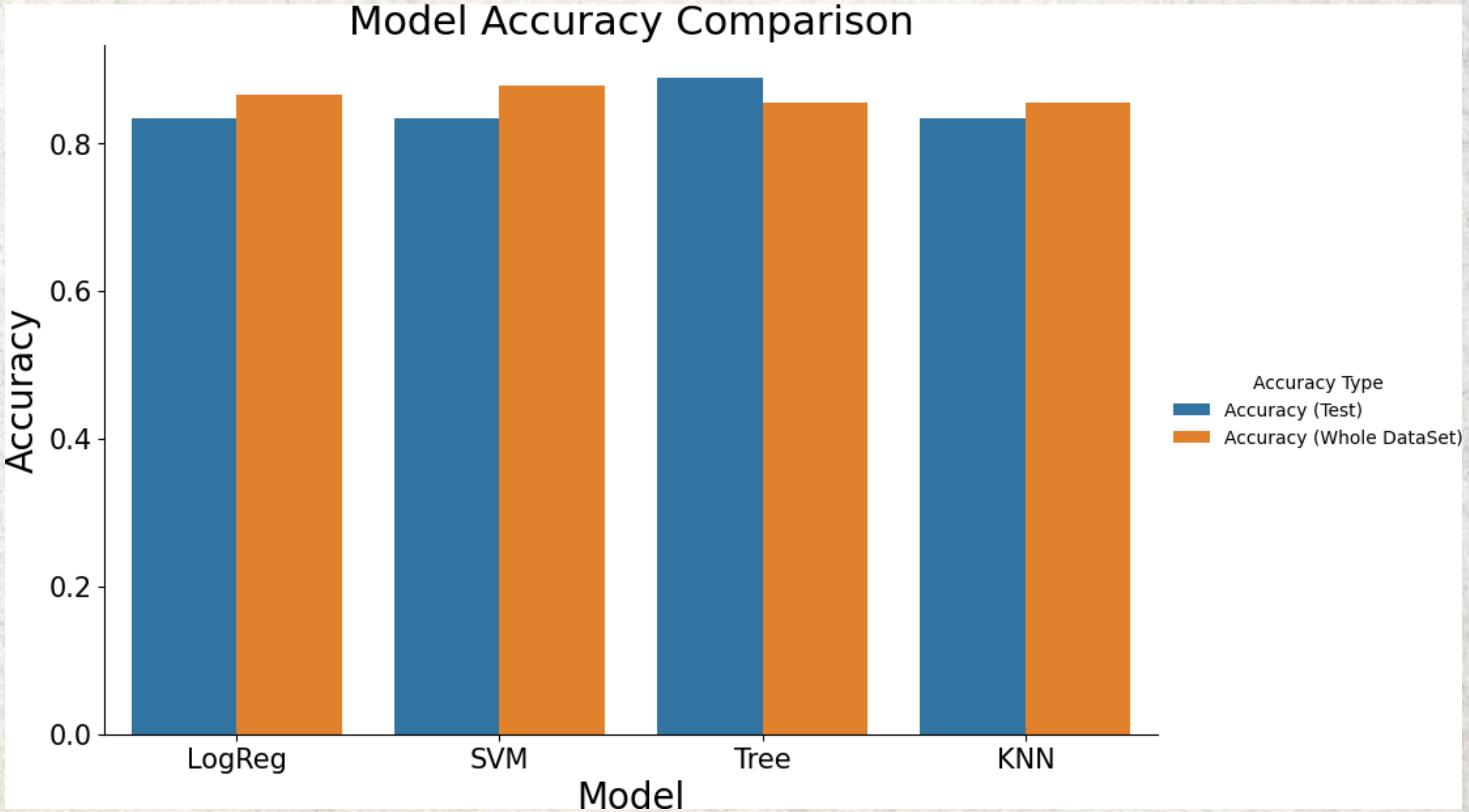
Section 5

Predictive Analysis (Classification)

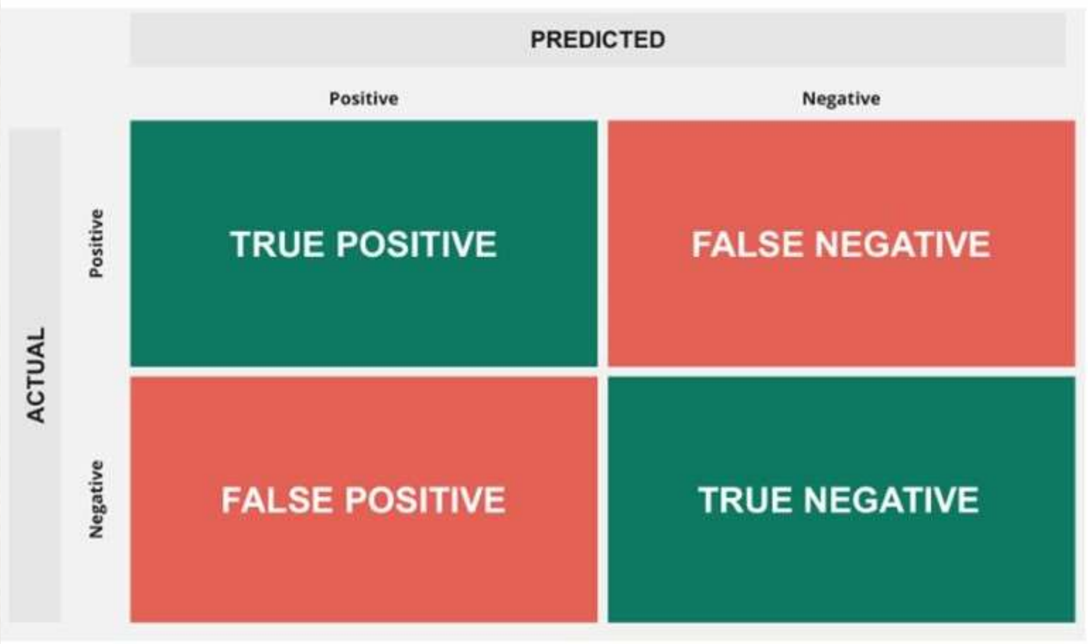
Classification Accuracy

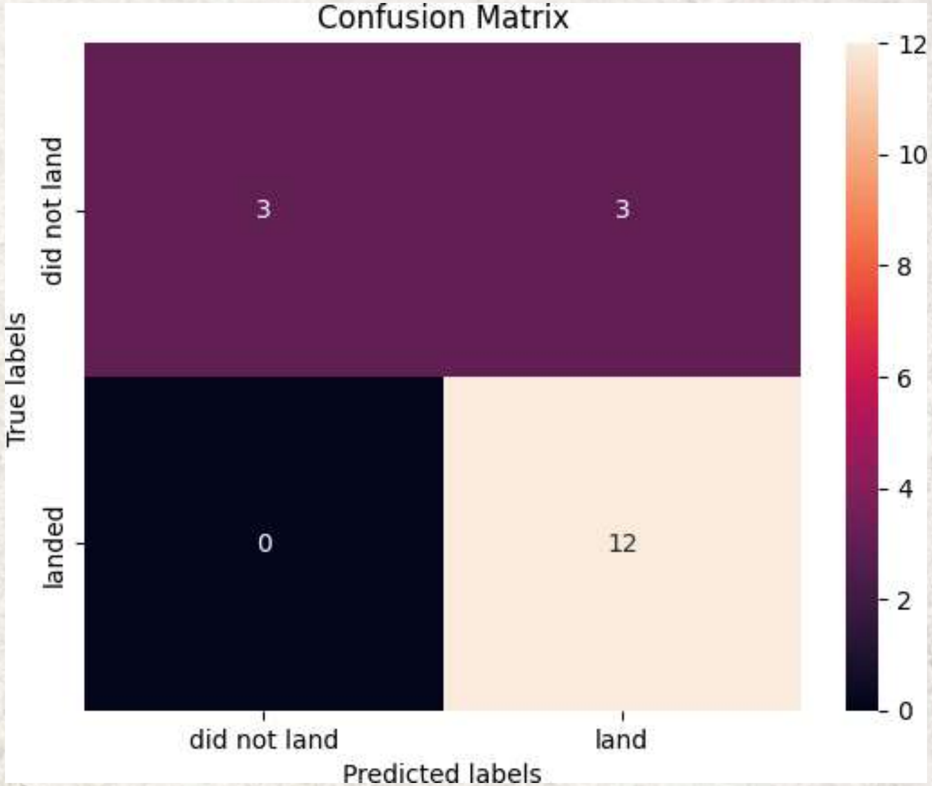
Explanation

- All models have comparable accuracy for both the test set and the entire data set.
- SVM had the highest accuracy for the entire data set, which may indicate that it is the best model of all four.



Confusion Matrix





Explanation

- A confusion matrix is a table used to evaluate the performance of a classification model by showing the counts of true positives, true negatives, false positives, and false negatives for each class.
- The model has problems with False Postive predictions (A: *did not land*, P: *land*)

Conclusions

- Launches with a low payload mass tend to perform better than those with larger payloads.
- Most launch sites are located near the Equator and in close proximity to the coast.
- The success rate of launches has been improving over the years.
- KSC LC-39A stands out with the highest success rate among all the launch sites.
- Orbits with 100% success rate:
  - ES-L1
  - GEO
  - HEO
  - SSO
- The SVM model proved to be the most effective algorithm for this dataset.

Thank you!