IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Author: Prajit Bhalala
Date: July 20, 2025

Outline

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

Executive Summary

- This report details a comprehensive project aimed at predicting successful first-stage landings of the SpaceX Falcon 9 rocket.
- The ability to reuse the first stage is a cornerstone of SpaceX's cost-efficiency, making accurate landing prediction vital for competitive bidding.
- The project involved three key phases: data acquisition from the SpaceX API , thorough exploratory data analysis (EDA) and feature engineering, and the application of machine learning models.
- Key EDA insights revealed a strong upward trend in launch success rates over the years, alongside varying success probabilities across different launch sites and orbit types.
- In the machine learning phase, Logistic Regression, SVM, Decision Tree, and K-Nearest Neighbors (KNN) models were developed and optimized. These models all achieved an 83.33% accuracy on the test set, demonstrating their effectiveness in distinguishing between successful and unsuccessful landing outcomes.
- This analysis provides valuable predictive capabilities for future SpaceX missions and competitive market strategies.

Introduction

4.1 Problem Statement

The core problem addressed in this project is to accurately predict whether the first stage of the SpaceX Falcon 9 rocket will land successfully. This predictive capability is vital for assessing the cost-effectiveness of a launch, as the reusability of the first stage significantly reduces overall mission expenses compared to other providers. Such information can be leveraged by alternative companies bidding against SpaceX for rocket launch contracts.

4.2 Background

SpaceX has garnered global attention for its achievements in space exploration, notably its pioneering work in reusing the first stage of its Falcon 9 rockets. The ability to return the first stage from low-Earth orbit, first accomplished in December 2010, represents a monumental shift in space launch economics. While successful landings are ideal, some missions involve planned unsuccessful landings, often controlled landings in the ocean. Visual examples illustrate both the precision of successful landings and the outcomes of unsuccessful attempts, including crashes.

4.3 Objectives

This report outlines a systematic approach to build a machine learning pipeline for predicting Falcon 9 first-stage landing outcomes. The project's objectives are structured into three main phases:

- **Data Collection:** To gather comprehensive historical launch records for Falcon 9 and Falcon Heavy rockets from various online sources, including the SpaceX API and Wikipedia. This involves making HTTP requests and performing web scraping to extract relevant data.
- **Exploratory Data Analysis (EDA) & Feature Engineering:** To analyze the collected data to identify patterns, understand the relationships between different variables and landing success, and prepare the data for machine learning. This includes handling missing values, creating new features, and encoding categorical variables.
- **Machine Learning Prediction:** To develop and evaluate several classification models (Logistic Regression, SVM, Decision Tree, and KNN) capable of predicting the landing outcome based on the engineered features. The aim is to identify the best-performing model for this prediction task.

Section 1

Methodology

Methodology

Executive Summary

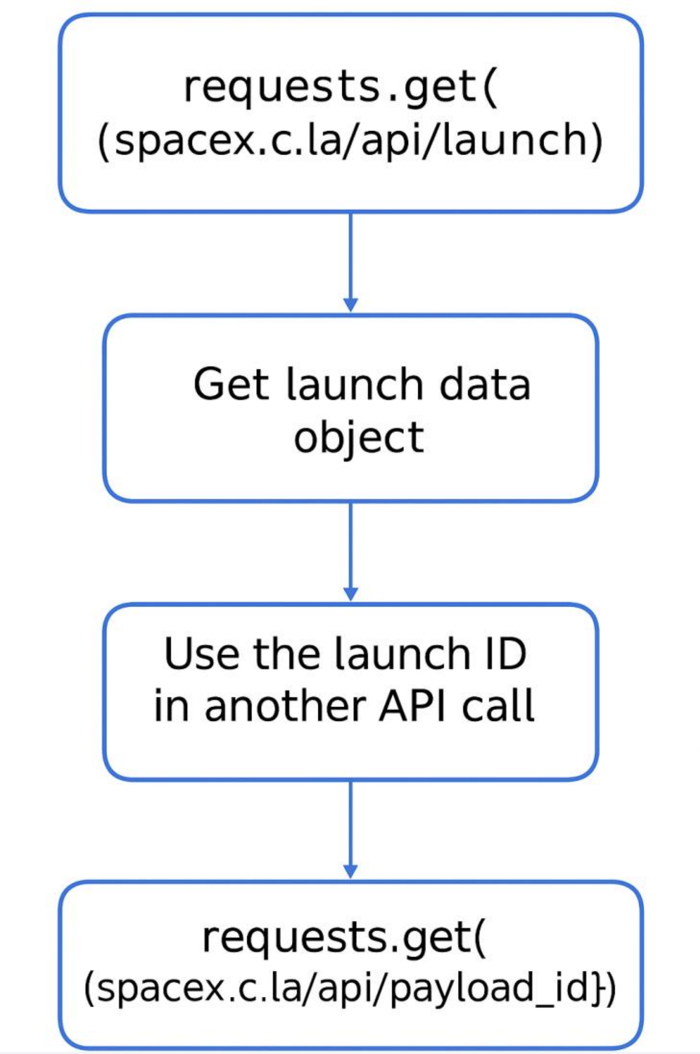
- Data collection methodology:
    - Describe how data was collected
- Perform data wrangling
    - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using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

Data Collection

- Comprehensive historical launch information was gathered from two primary sources:
- SpaceX API: Historical launch data was retrieved directly from the SpaceX API, specifically from the /v4/launches/past endpoint. Python's requests and pandas libraries were used to process the JSON responses into a DataFrame. Helper functions extracted key details like BoosterVersion, LaunchSite (including geo-coordinates), PayloadMass, Orbit, and various Core data attributes such as landing success, type, and reuse counts. The initial DataFrame was filtered to include only single-core, single-payload Falcon 9 launches within a specific date range (up to November 13, 2020), and missing PayloadMass values were imputed with the mean.
- Wikipedia Web Scraping: Additional historical launch records for Falcon 9 and Falcon Heavy were scraped from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches" (snapshot from June 9, 2021). requests and BeautifulSoup were employed for HTML parsing, with custom helper functions to accurately extract data from the complex HTML table structures.

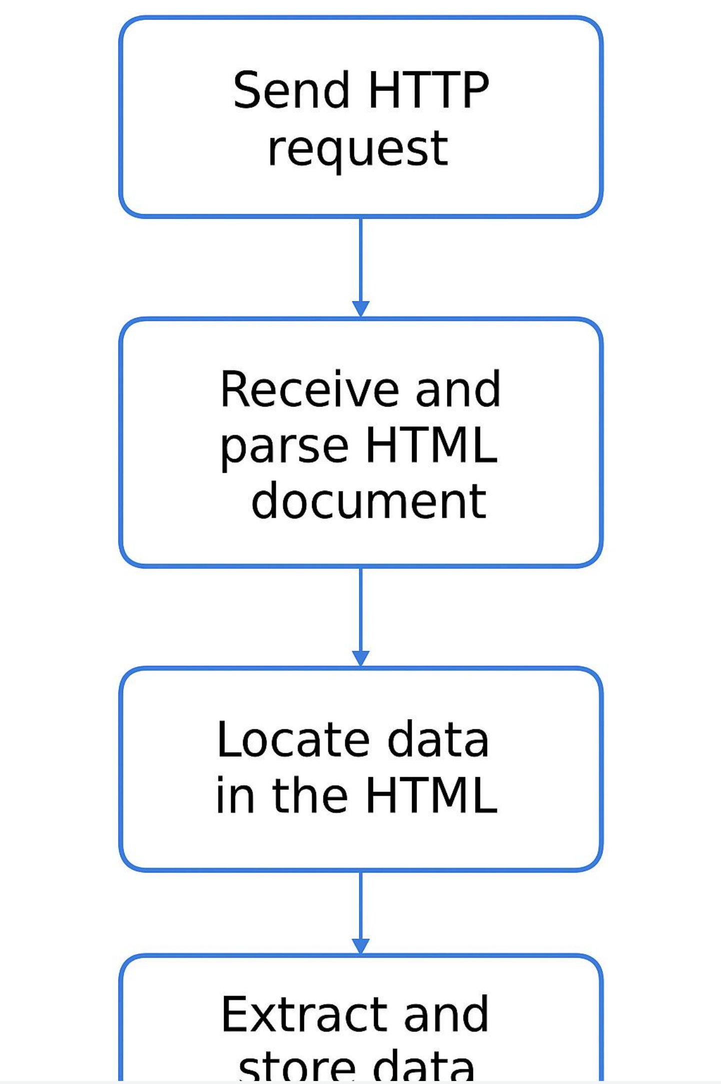
Data Collection – SpaceX API

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/3584e4968e3b9c2b1d6aca6ebecd99d27e846fa3/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/3584e4968e3b9c2b1d6aca6ebecd99d27e846fa3/jupyter-labs-webscraping.ipynb



Data Wrangling

- Once collected, the raw data underwent several critical transformations for machine learning:
- Handling Missing Values: NaN entries in LandingPad were retained, indicating no landing pad use, while missing PayloadMass values were imputed with their mean.
- Creation of Class Label: A binary Class variable (0 for unsuccessful, 1 for successful) was created by transforming the Outcome column (e.g., 'False ASDS', 'None None' mapped to 0; 'True ASDS' mapped to 1).
- Feature Selection & Encoding: Relevant features including FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, and Serial were selected. Categorical features (Orbit, LaunchSite, LandingPad, Serial) were then converted into numerical representations using One-Hot Encoding via pandas.get_dummies(). All resulting numeric columns were cast to float64 for consistency.

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/3584e4968e3b9c2b1d6aca6ebecd99d27e846fa3/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Initial data inspection of 90 Falcon 9 launches revealed 5 missing PayloadMass values and 26 LandingPad missing values, with PayloadMass imputed by its mean, while LandingPad NaNs were retained as valid indicators of no pad use.
- Launch Site Analysis: Three unique launch sites were identified: CCAFS SLC 40 (55 launches), KSC LC 39A (22 launches), and VAFB SLC 4E (13 launches).
- Relationship between Flight Number and Launch Site: A scatter plot of FlightNumber vs. LaunchSite showed that increasing flight numbers correlated with higher landing success across different sites. Notably, VAFB-SLC 4E recorded no successful landings for heavy payloads (>10,000 kg).
- Relationship between Payload Mass and Launch Site: The PayloadMass vs. LaunchSite scatter plot further highlighted VAFB-SLC 4E's consistent unsuccessful landings for heavy payloads (>10,000 kg), while other sites showed success across varied payload masses.

- Orbit Type Analysis: Eleven unique orbit types were present, with GTO (27 launches) and ISS (21 launches) being the most frequent. A bar chart of success rates per orbit indicated 100% success for HEO, GEO, and ES-L1 (though with limited launches), and high rates for ISS and VLEO.
- Relationship between Flight Number and Orbit Type: Success in LEO orbits positively correlated with higher FlightNumber, whereas GTO orbits showed no clear relationship, with mixed outcomes across flight numbers.

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/3584e4968e3b9c2b1d6aca6ebecd99d27e846fa3/edadataviz.ipynb

EDA with SQL

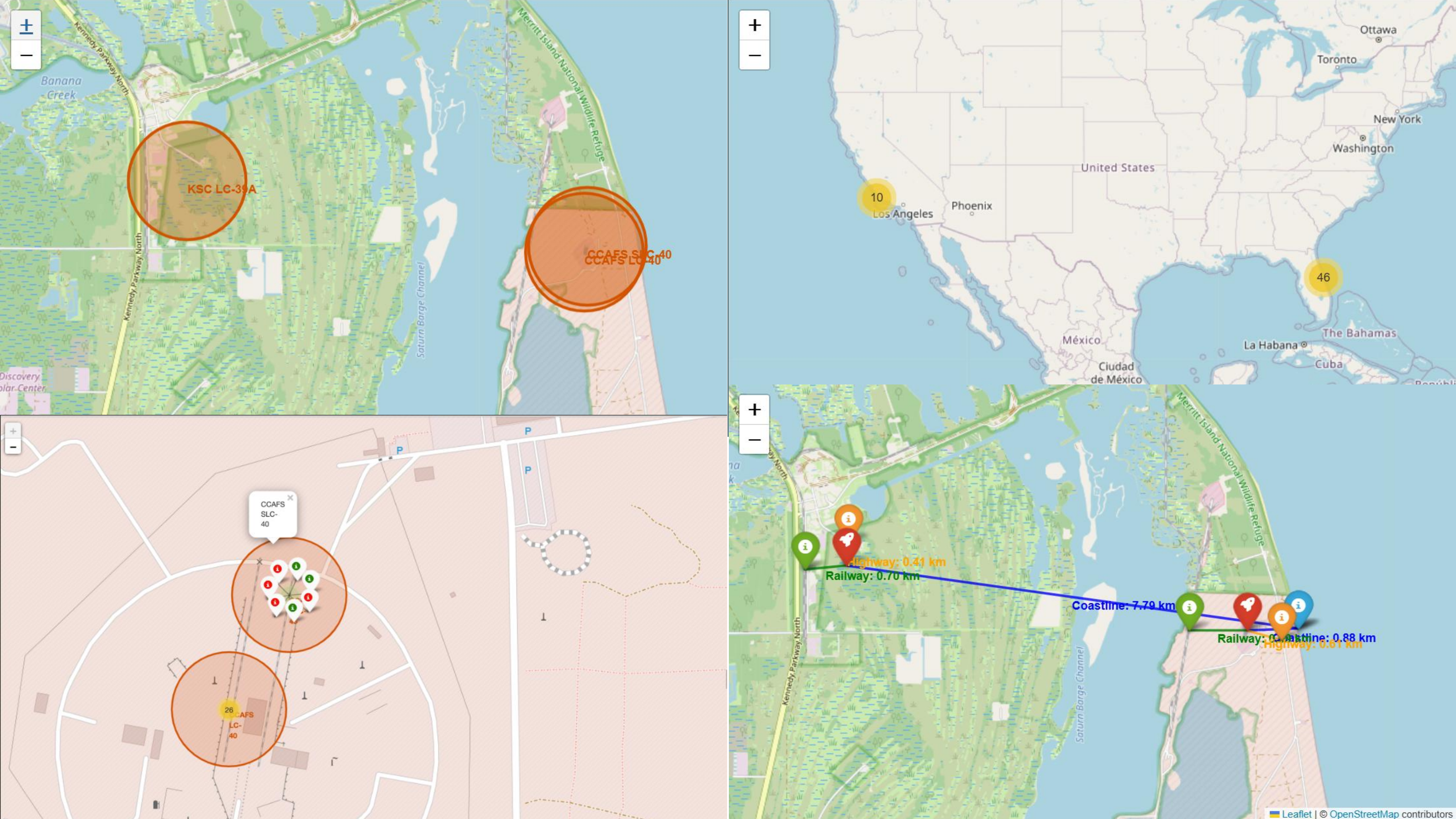
- Displayed unique launch sites.Retrieved 5 records for launch sites starting with 'CCA'.
- Calculated total payload mass carried by NASA (CRS) boosters.
- Determined average payload mass for F9 v1.1 booster version.Listed the date of the first successful ground pad landing.Identified booster versions with successful drone ship landings and specific payload mass.
- Listed total successful and failure mission outcomes.Ranked booster versions by maximum payload mass.
- Displayed records for failure landing outcomes in drone ship in 2015.Ranked landing outcomes between specific dates in descending order.

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/3584e4968e3b9c2b1d6aca6ebecd99d27e846fa3/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

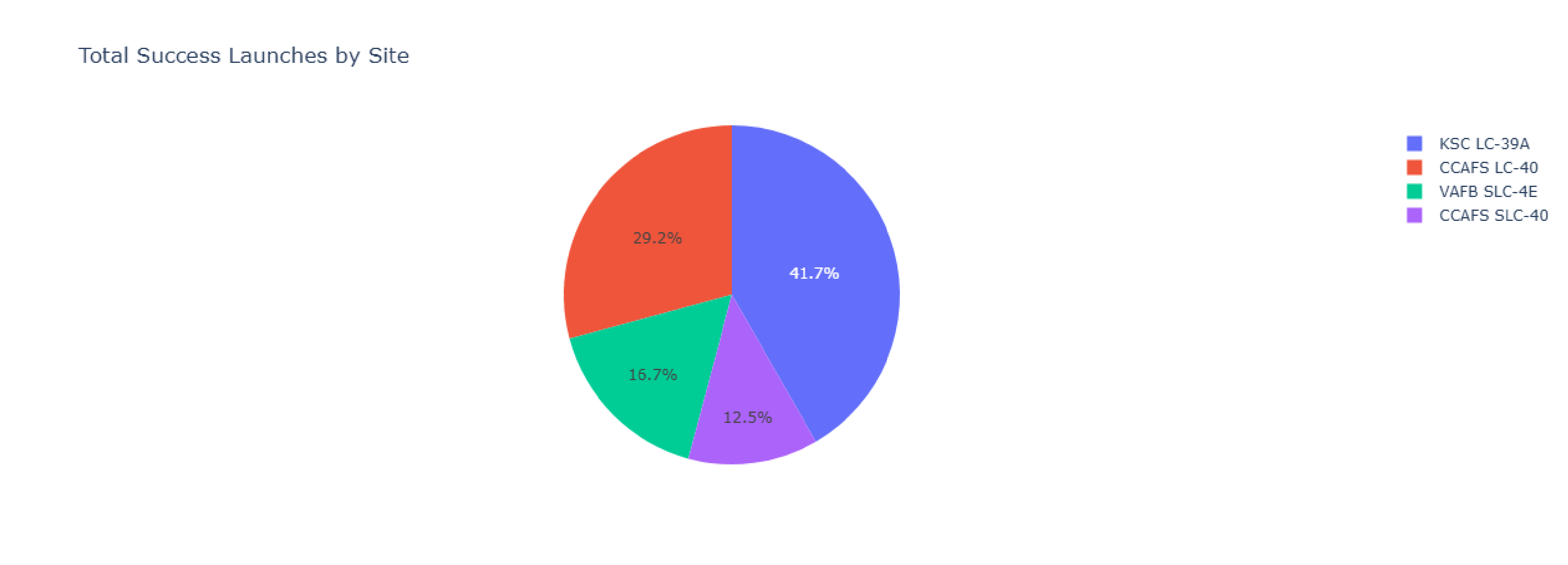
- To visually analyze the launch sites and their outcomes, a Folium map was created. Markers were added to pinpoint the exact locations of each launch site, providing geographical context. Additionally, circles were drawn around these markers to illustrate the successful payload deployment range from each site, helping to understand spatial success patterns.
- Markers were added to precisely locate each launch site on the map. Circles were used to visually represent the successful payload deployment range, indicating the reach of successful missions from those sites.

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/934a72175bc10dc958df37983af102461dfc5ea3/lab-jupyter-launch-site-location-v2.ipynb


Banana Creek
Kennedy Parkway North
KSC LC-39A
Merritt Island National Wildlife Refuge
CCAFS SLC-40
CCAFS LC-40
Saturn Barge Channel
Ottawa
Toronto
New York
Washington
United States
10
Los Angeles
Phoenix
46
The Bahamas
La Habana
Cuba
México
Ciudad de México
CCAFS SLC-40
26
Highway: 0.41 km
Railway: 0.70 km
Coastline: 7.79 km
Coastline: 0.88 km
Leaflet | © OpenStreetMap contributors

Build a Dashboard with Plotly Dash

- I added a Dash dashboard with a pie chart and a scatter plot. The pie chart visualizes launch success by site, with dropdown interaction to filter "All Sites" or specific locations. The scatter plot shows payload mass vs. outcome, using a range slider for payload filtering, and also interacts with the site dropdown to analyze specific sites or all.
- GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/934a72175bc10dc958df37983af102461dfc5ea3/spacex-dash-app.py



Predictive Analysis (Classification)

- Four classification models were trained and evaluated on standardized data, split into 80% training and 20% testing sets. Hyperparameter tuning was performed using GridSearchCV with 10-fold cross-validation.
- Logistic Regression: Best hyperparameters: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}. Validation Accuracy: 0.8464. Test Accuracy: 0.8333. The confusion matrix showed 12 true positives, 3 false positives, 3 false negatives, and 0 true negatives.
- Support Vector Machine (SVM): Best hyperparameters: {'C': 1.0, 'gamma': 0.0316, 'kernel': 'sigmoid'}. Validation Accuracy: 0.8482. Test Accuracy: 0.8333. Confusion matrix results were identical to Logistic Regression.
- Decision Tree Model: Best hyperparameters: {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}. Validation Accuracy: 0.8893. Test Accuracy: 0.8333. Confusion matrix results mirrored the other models.
- K-Nearest Neighbors (KNN) Model: Best hyperparameters: {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}. Validation Accuracy: 0.8482. Test Accuracy: 0.8333. Confusion matrix results were consistent across all models.

- Comparative Model Performance: All four models achieved an identical test accuracy of 83.33%. While the Decision Tree showed a higher validation accuracy, its test set performance aligned with the others, suggesting comparable predictive power on this dataset.

GitHub URL: https://github.com/Prajit1221/SpaceX-Falcon-9-First-Stage-Landing-Prediction/blob/934a72175bc10dc958df37983af102461dfc5ea3/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- The exploratory data analysis provided crucial insights into factors influencing Falcon 9 first-stage landing success. An increasing success rate over the years highlights SpaceX's continuous improvements in reusability.
- Launch site analysis revealed distinct patterns; VAFB SLC 4E, for instance, showed no successful heavy payload landings (>10,000 kg), suggesting specific operational limitations. Orbit type also proved significant, with LEO, ISS, HEO, GEO, and ES-L1 demonstrating high success rates, while GTO presented more complexity.
- In the machine learning phase, Logistic Regression, SVM, Decision Tree, and KNN models all achieved an identical test accuracy of 83.33%. Consistently, confusion matrices showed 12 true positives, 3 false positives, 3 false negatives, and 0 true negatives, indicating strong success prediction but some false positives/negatives.
- The Decision Tree model exhibited the highest validation accuracy (88.93%), suggesting potential for better generalization with more data. Overall, the models offer a robust framework, with EDA insights informing predictions.

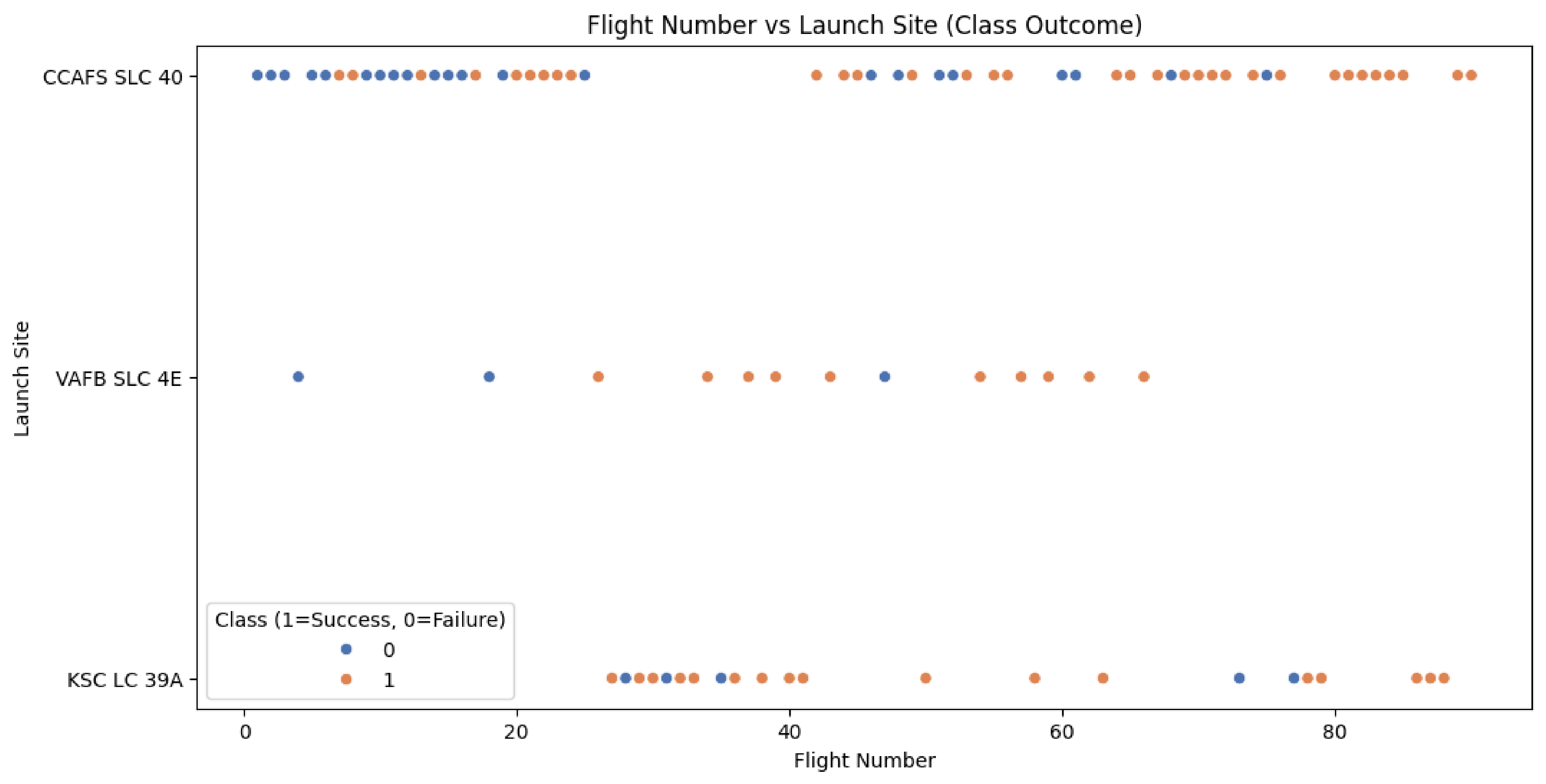
- This report successfully demonstrated a comprehensive approach to predicting SpaceX Falcon 9 first-stage landing outcomes, crucial for optimizing rocket reusability and launch costs.
- Through meticulous data collection from the SpaceX API and Wikipedia, a rich dataset was compiled. Exploratory data analysis revealed significant trends: SpaceX's landing success rate has shown a clear upward trajectory since 2013, and different launch sites and orbit types exhibit varying success patterns.
- Four machine learning models—Logistic Regression, SVM, Decision Tree, and KNN—were implemented, optimized, and evaluated. All achieved an identical test accuracy of 83.33%, demonstrating strong predictive capability for landing success. While test accuracies were uniform, the Decision Tree showed a promising higher cross-validation score.
- Future work includes increasing data volume, advanced feature engineering, and exploring deep learning for enhanced accuracy and interpretability. This project establishes a solid foundation for understanding and predicting Falcon 9 landing outcomes.

Section 2

Insights drawn from EDA

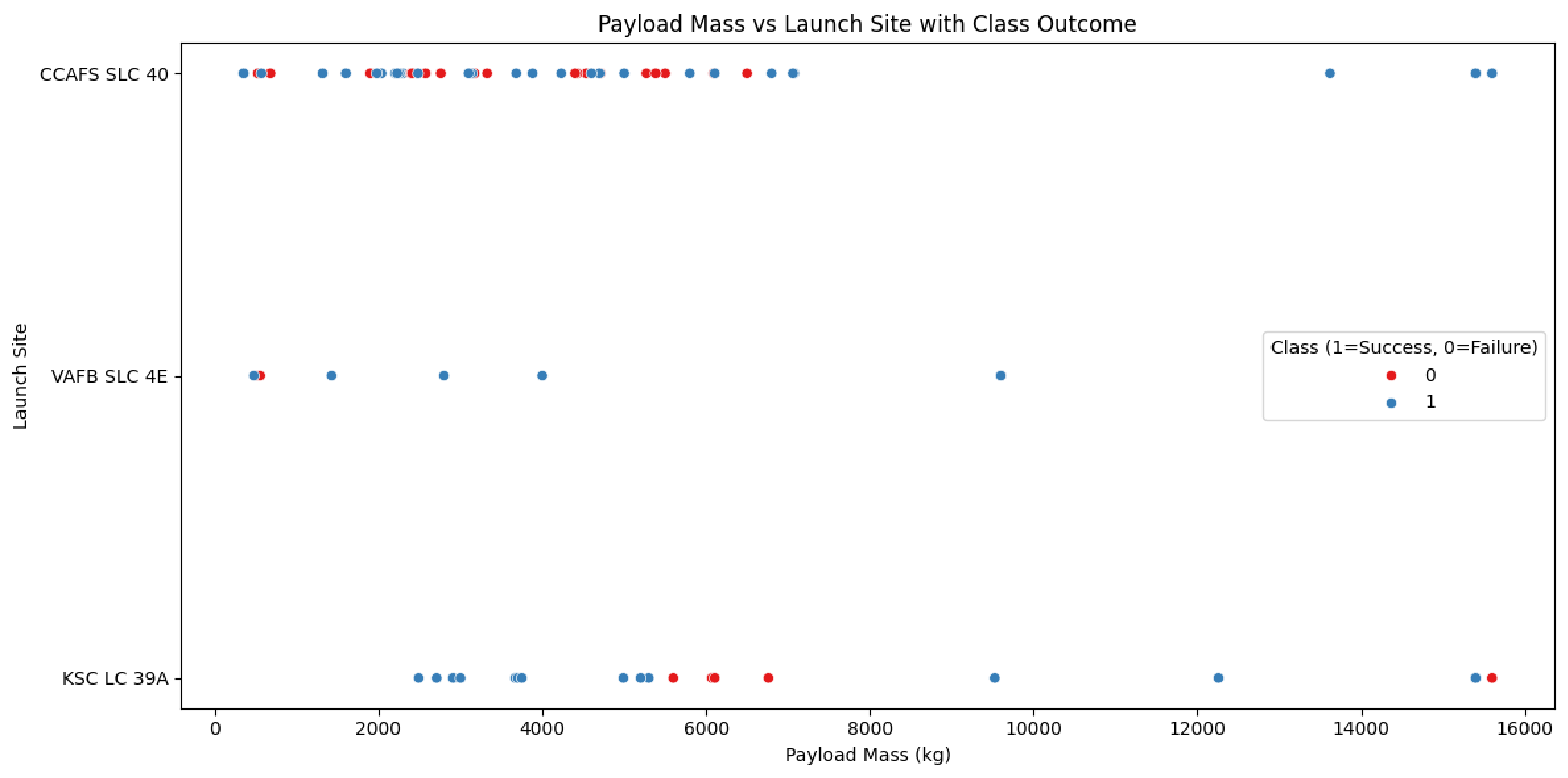
Flight Number vs. Launch Site

- This scatter plot reveals VAFB SLC 4E had only failures. CCAFS SLC 40 and KSC LC 39A show mixed outcomes, with more successes at higher Flight Numbers.



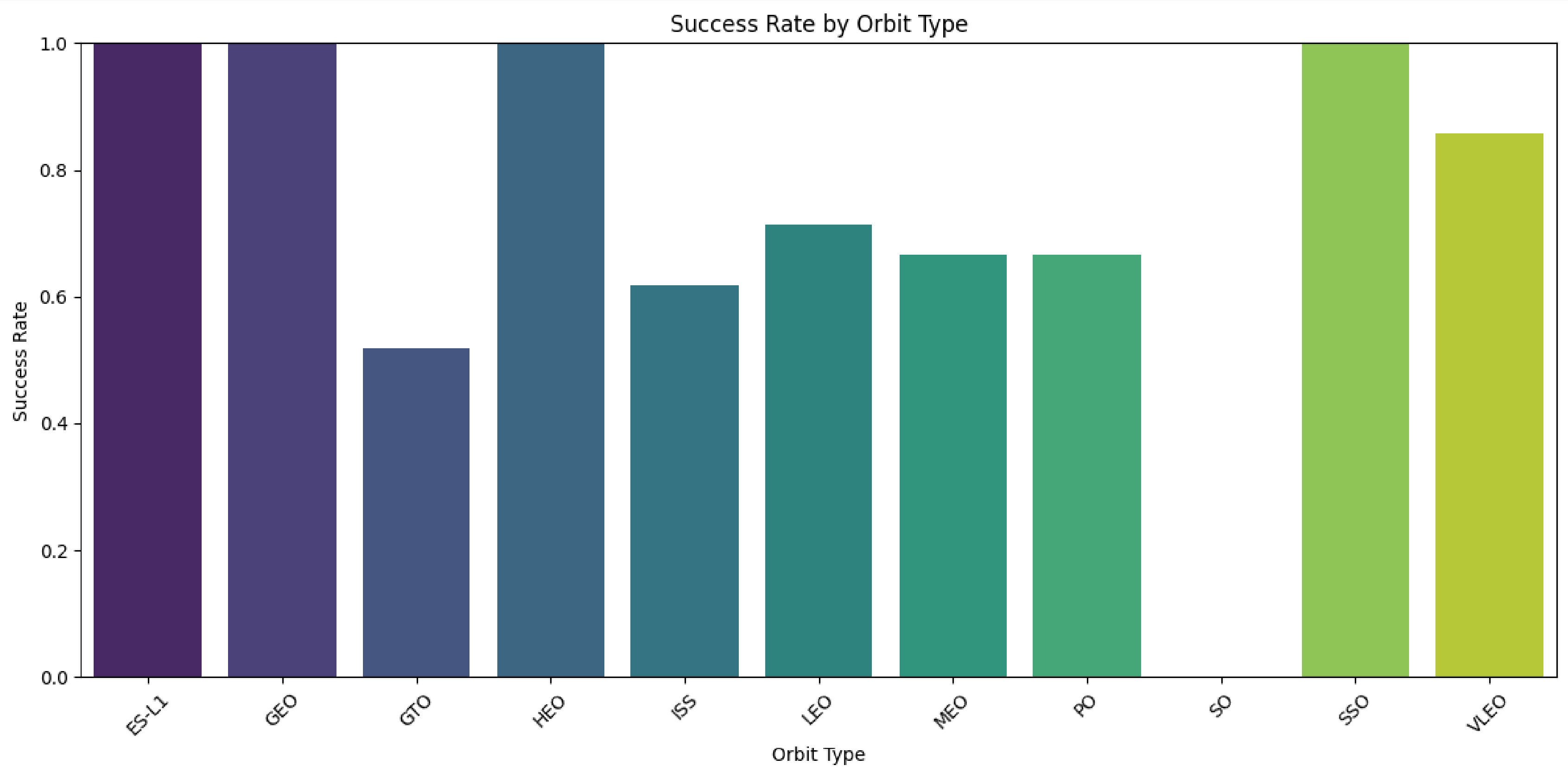
Payload vs. Launch Site

- This scatter plot shows that VAFB SLC 4E primarily had failures, while CCAFS SLC 40 and KSC LC 39A had mixed success across varying payload masses.



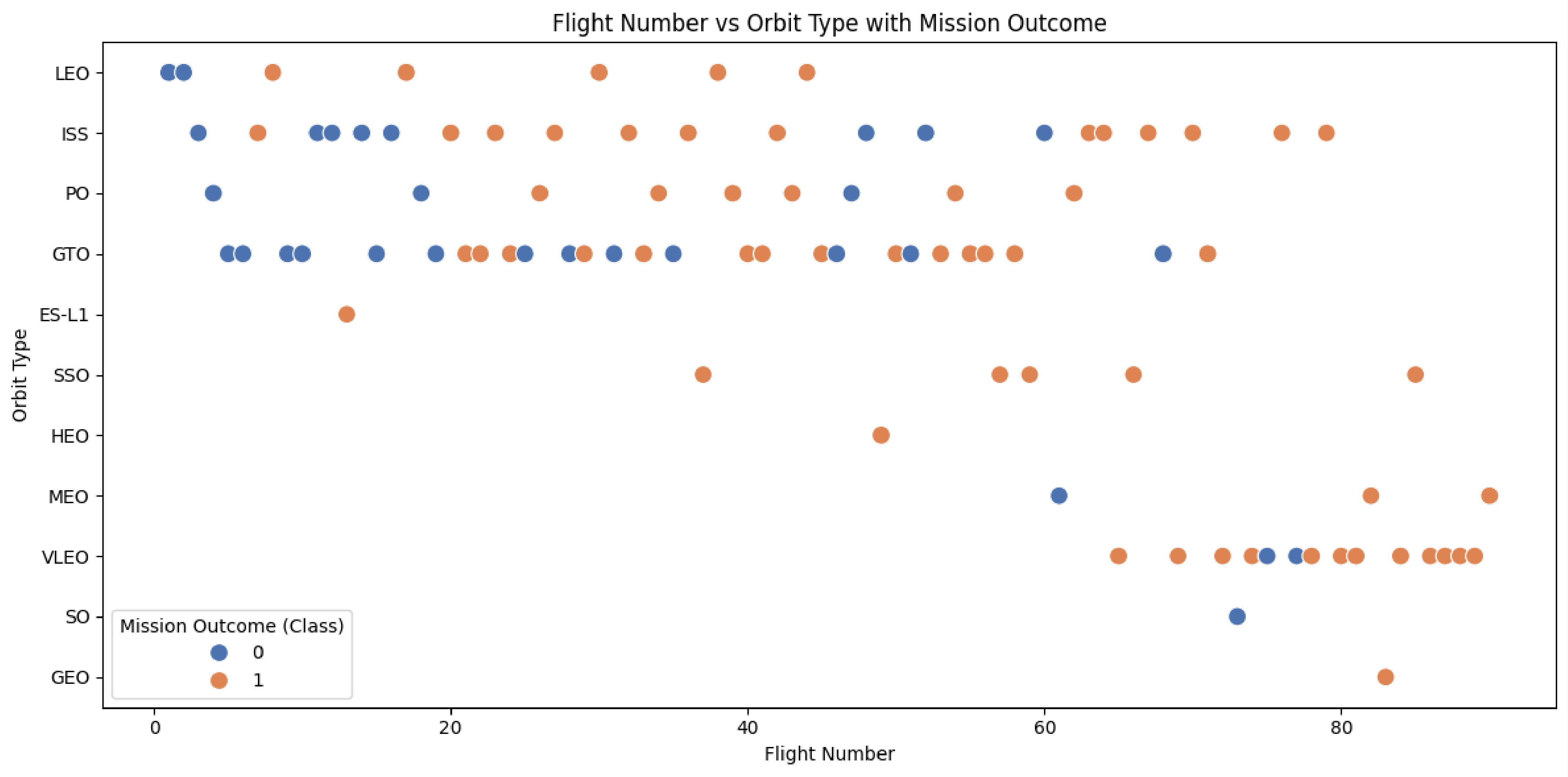
Success Rate vs. Orbit Type

- This bar chart shows Success Rate by Orbit Type. ES-L1, GEO, HEO, SSO, and VLEO orbits achieved 100% or high success rates, while GTO, ISS, LEO, MEO, and PO showed varied, generally lower, success.



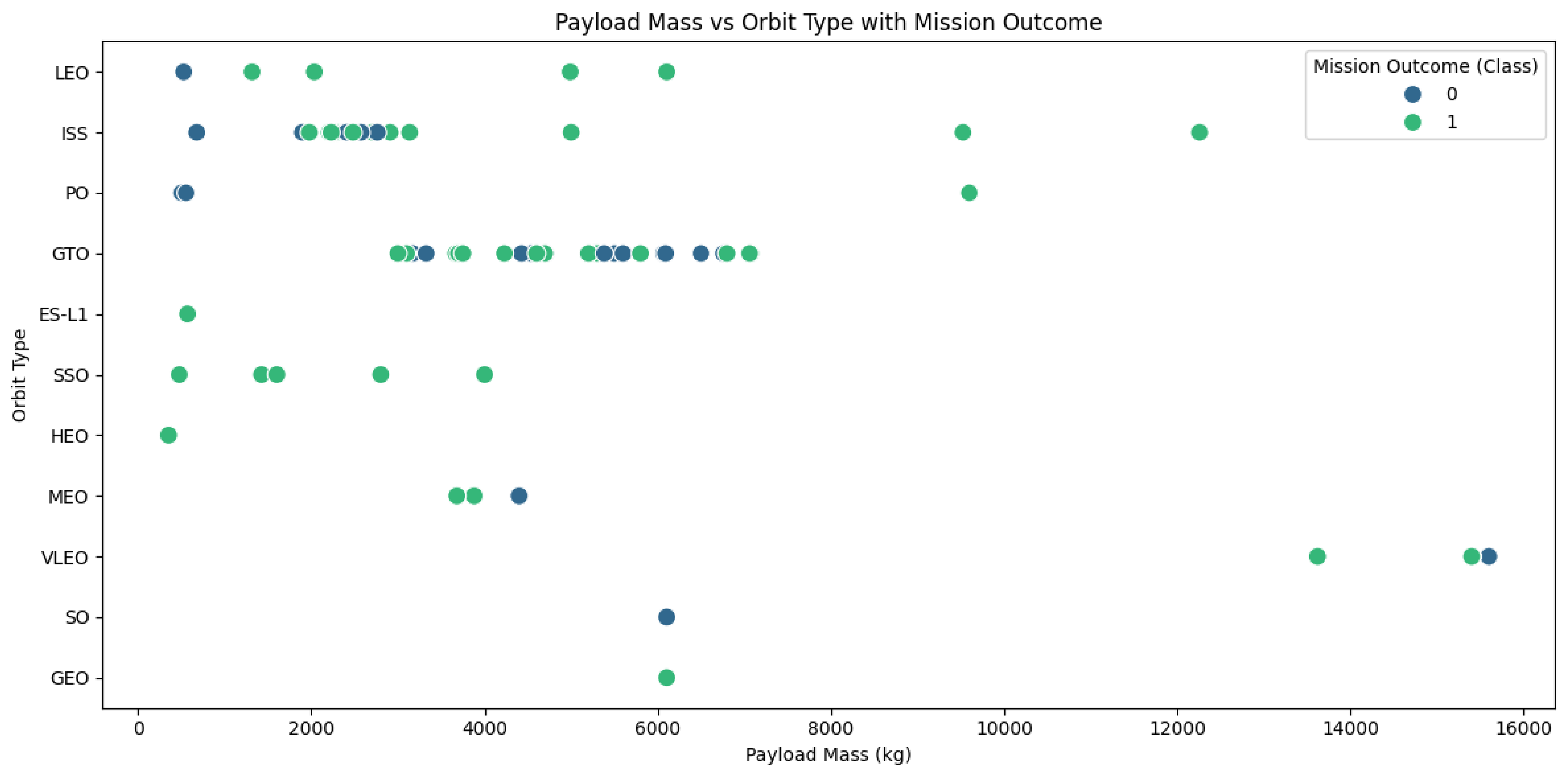
Flight Number vs. Orbit Type

- This scatter plot illustrates Flight Number vs. Orbit Type with Mission Outcome. LEO and ISS orbits show mixed outcomes across flight numbers, while higher flight numbers generally correlate with more successful outcomes for VLEO and SSO orbits.



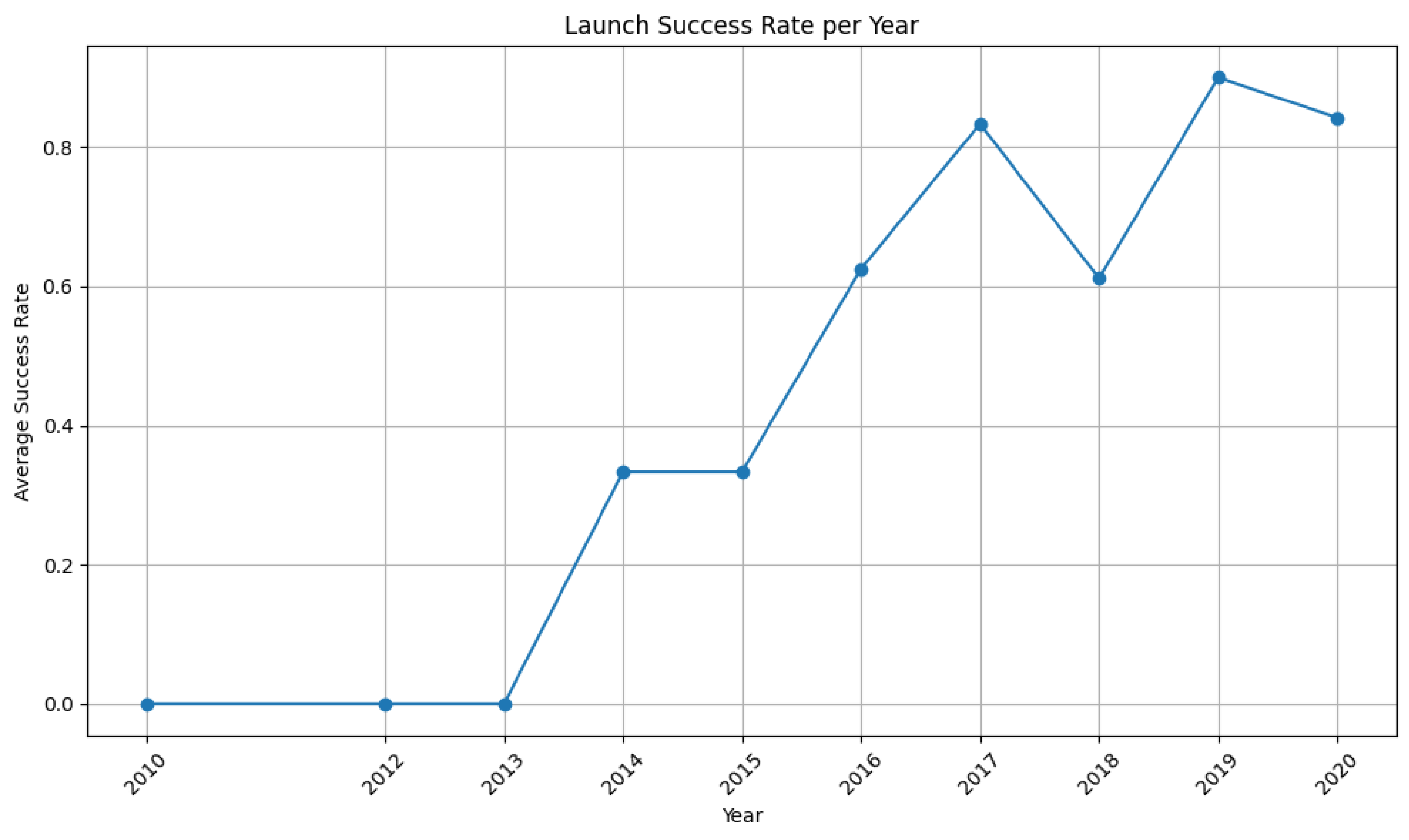
Payload vs. Orbit Type

- This scatter plot displays Payload Mass vs. Orbit Type with Mission Outcome. Successful outcomes (Class 1) are observed across various payload masses and orbit types, with a notable clustering of successes for high Payload Mass in VLEO.



Launch Success Yearly Trend

- This line chart shows the Launch Success Rate per Year. The success rate was zero until 2013, then significantly increased, reaching a peak in 2019 before slightly declining in 2020.



All Launch Site Names

- Query: SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
- Explanation: This query is designed to retrieve all unique values from the "Launch_Site" column in the SPACEXTABLE.
- The DISTINCT keyword ensures that only unique site names are returned, eliminating any duplicates.

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

Launch Site Names Begin with 'CCA'

- Query: SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
- Explanation: This query selects all columns (*) from SPACEXTABLE. The WHERE "Launch_Site" LIKE 'CCA%' clause filters the results to include only those rows where the "Launch_Site" column starts with the characters 'CCA'.
- The % is a wildcard character representing any sequence of zero or more characters. LIMIT 5 restricts the output to the first 5 matching records found.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2

Total Payload Mass

- Query: SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%’;
- Explanation: This query calculates the sum of all values in the "Payload_Mass__kg_" column. It renames the sum to Total_Payload_Mass using AS.
- The WHERE "Customer" LIKE '%NASA (CRS)%' clause filters records where the "Customer" column contains the substring 'NASA (CRS)'.

Total_Payload_Mass
48213

Average Payload Mass by F9 v1.1

- Query: SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1’;
- Explanation: This query computes the average of the "Payload_Mass__kg_" for all entries. The result is aliased as Average_Payload_Mass.
- The WHERE "Booster_Version" = 'F9 v1.1' clause ensures that only records corresponding to the 'F9 v1.1' booster version are included in the average calculation.

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

- Query: SELECT MIN(Date) AS First_Successful_Ground_Pad_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)’;
- Explanation: This query finds the earliest Date from the SPACEXTABLE.
- It filters records where the "Landing_Outcome" is exactly 'Success (ground pad)', and MIN(Date) then returns the minimum (earliest) date among these successful ground pad landings.

First_Successful_Ground_Pad_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query: SELECT Booster_Version FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass__kg_" > 4000 AND "Payload_Mass__kg_" < 6000;
- Explanation: This query retrieves the Booster_Version for records that meet two conditions: the "Landing_Outcome" must be 'Success (drone ship)', AND the "Payload_Mass__kg_" must be between 4000 and 6000 kg (exclusive of 4000 and 6000).

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

Total Number of Successful and Failure Mission Outcomes

- Query: SELECT "Mission_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE GROUP BY "Mission_Outcome";
- Explanation: This query counts the occurrences of each unique "Mission_Outcome" in the SPACEXTABLE. The GROUP BY "Mission_Outcome" clause groups rows that have the same mission outcome, and COUNT(*) then tallies the number of rows within each group. The count is aliased as Outcome_Count.

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

Boosters Carried Maximum Payload

- Query: SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
- Explanation: This query uses a subquery to first determine the MAX("Payload_Mass__kg_") from the entire SPACEXTABLE. The outer query then selects the Booster_Version and PAYLOAD_MASS__KG_ for all rows where the PAYLOAD_MASS__KG_ is equal to this maximum value, effectively listing all booster versions that carried the heaviest payload.

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

2015 Launch Records

- Query: SELECT substr(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Failure%' AND "Landing_Outcome" LIKE '%drone ship%' AND substr(Date, 0, 5) = '2015’;
- Explanation: This query extracts the month from the Date column using substr(Date, 6, 2) (starting at position 6 for 2 characters, e.g., 'MM') and aliases it as Month. It selects Landing_Outcome, Booster_Version, and Launch_Site.
- The WHERE clause applies multiple filters: the Landing_Outcome must contain both 'Failure' and 'drone ship' (using LIKE '%Failure%' and LIKE '%drone ship%'), and the year extracted from the Date (using substr(Date, 0, 5)) must be '2015'.

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

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

- Query: SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Outcome_Count DESC;
- Explanation: This query counts the occurrences of each unique Landing_Outcome. It filters records to include only those where the Date falls within the specified range ('2010-06-04' and '2017-03-20'). The results are grouped by Landing_Outcome and then ordered in DESCending order based on their Outcome_Count.

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

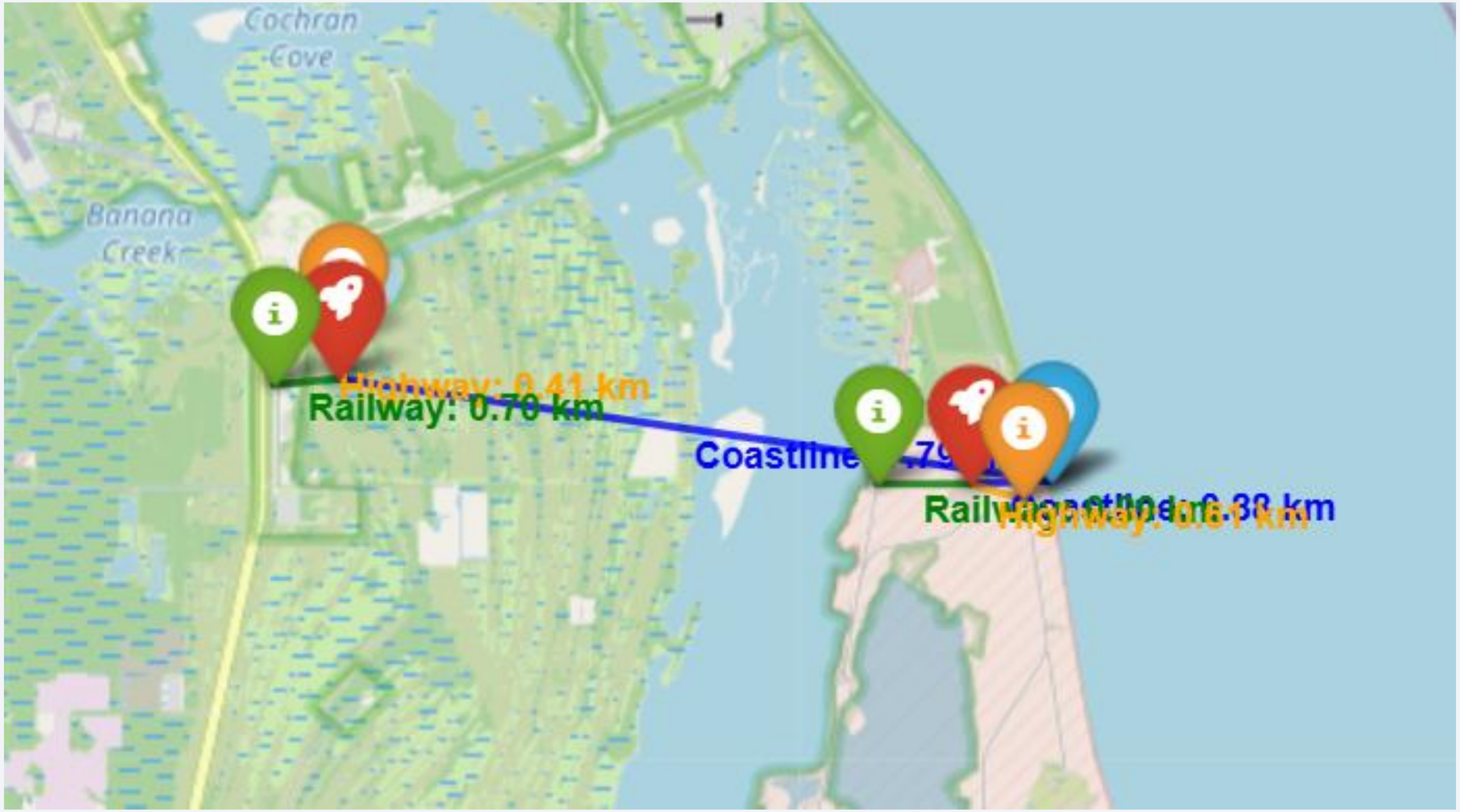
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

<Folium Map Screenshot 2>

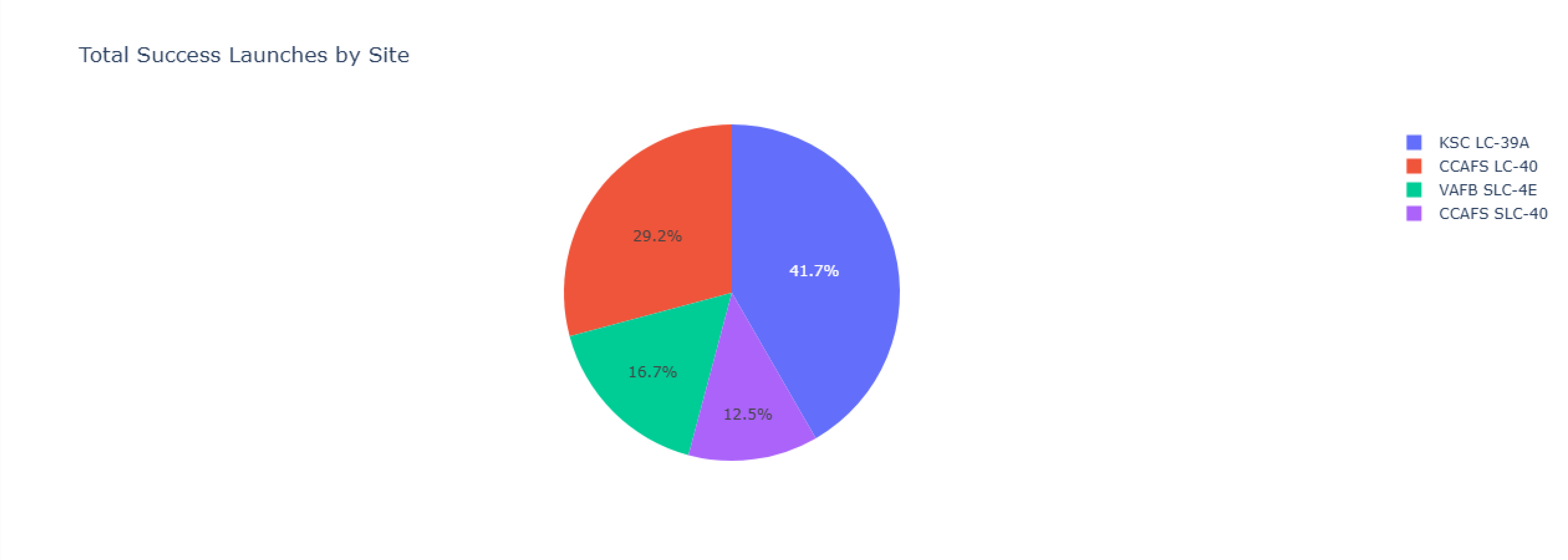
<Folium Map Screenshot 3>



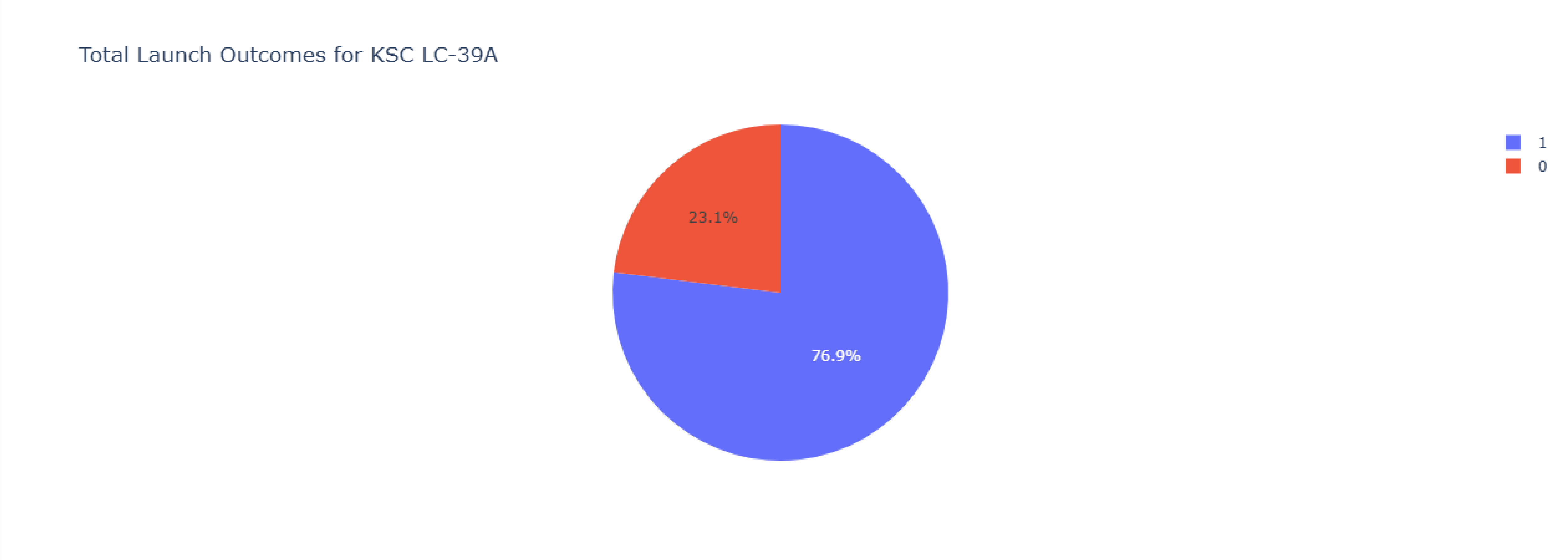
Section 4

Build a Dashboard with Plotly Dash

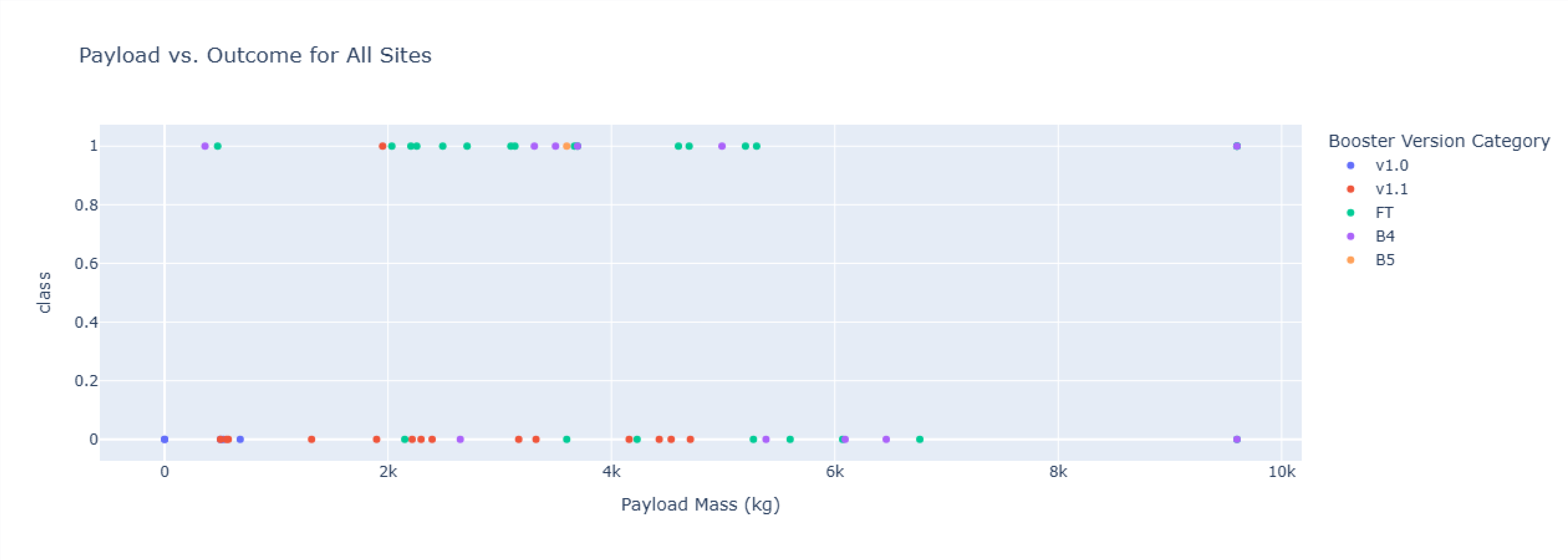
<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



<Dashboard Screenshot 3>

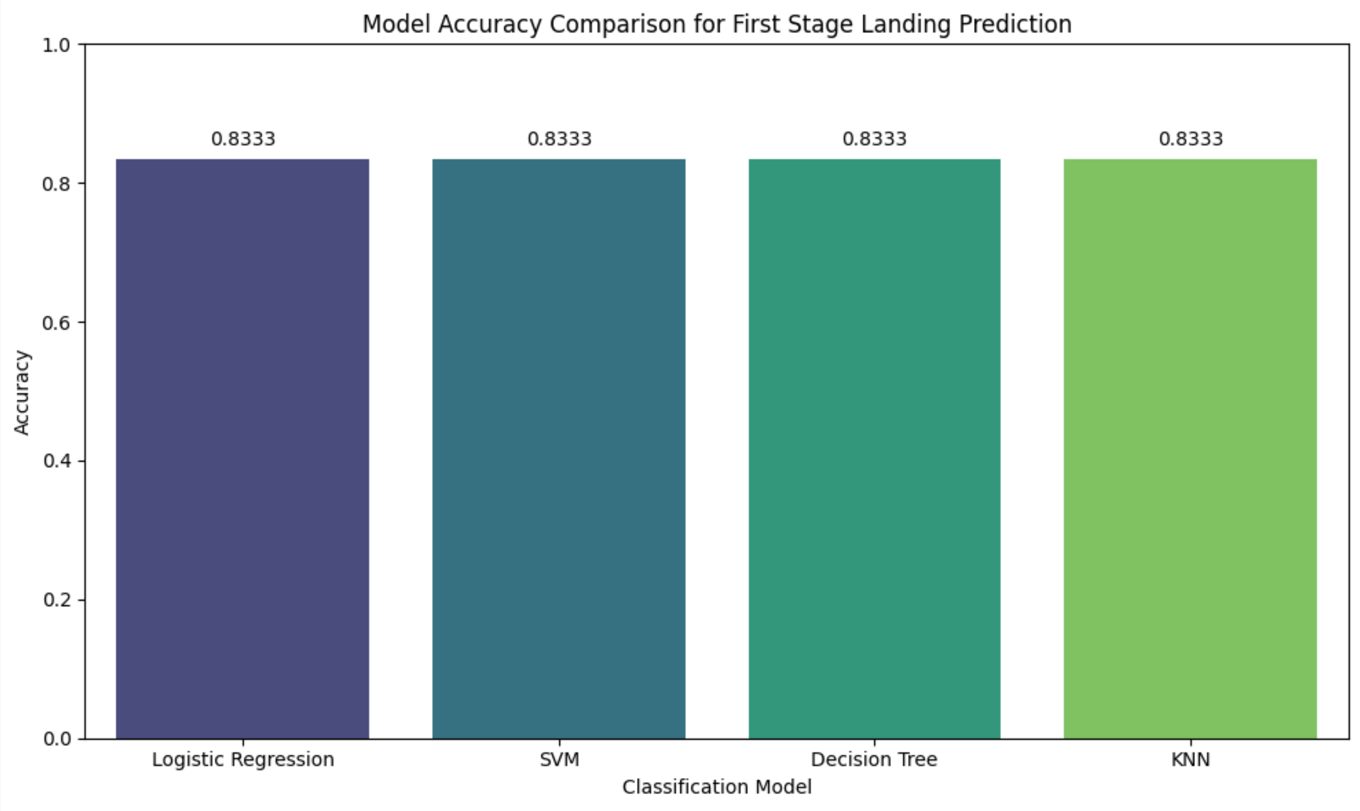


Section 5

Predictive Analysis (Classification)

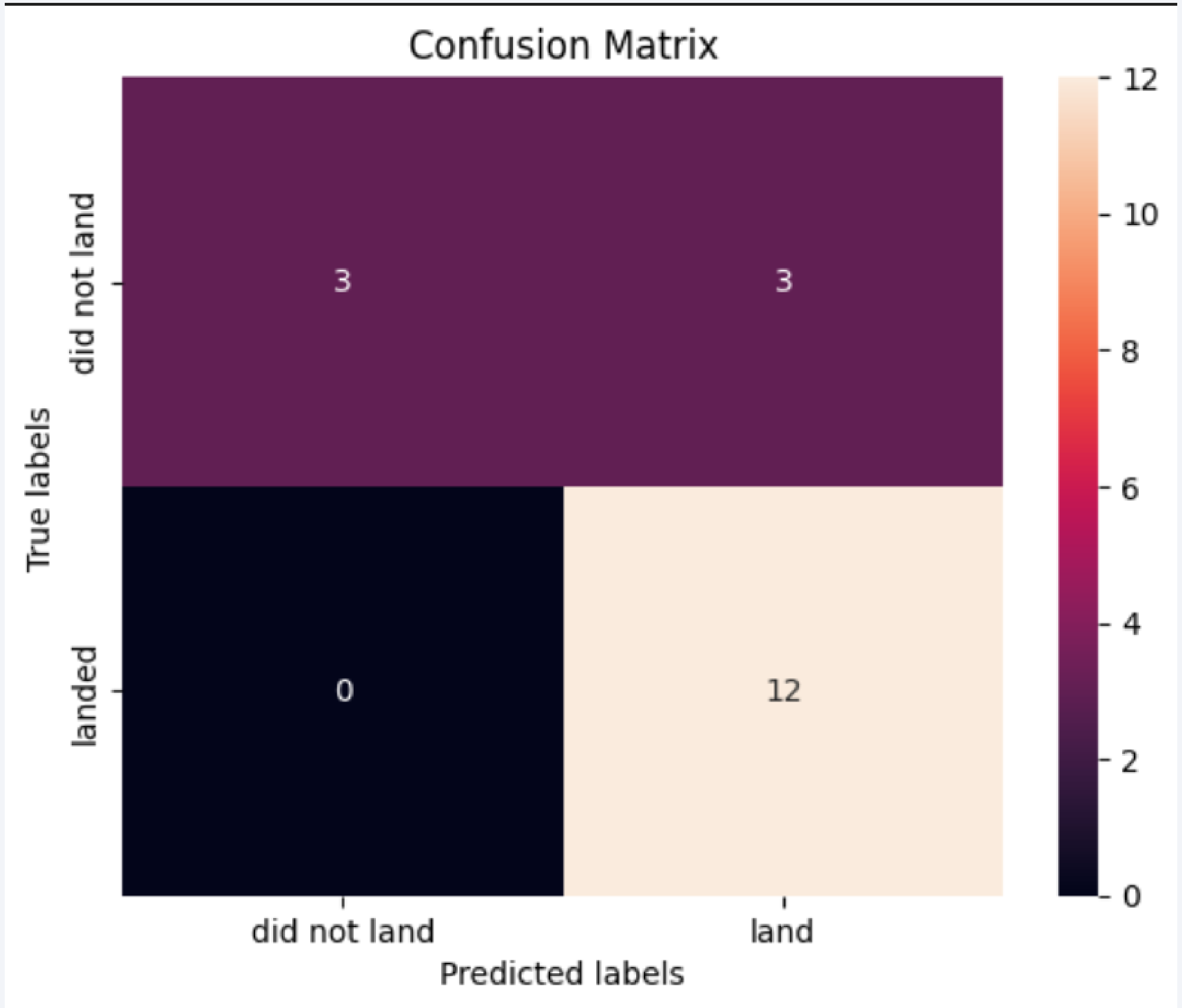
Classification Accuracy

- Based on the machine learning model evaluation, Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN) all achieved an identical test accuracy of 0.8333 (83.33%). Therefore, according to the test accuracy, all these models performed equally well.



Confusion Matrix

- A confusion matrix summarizes a classification model's performance. It shows correctly predicted landings (True Positives) versus incorrect predictions (False Positives/Negatives), revealing specific error types.



Conclusions

- This report successfully demonstrated a comprehensive approach to predicting SpaceX Falcon 9 first-stage landing outcomes, crucial for optimizing rocket reusability and launch costs.
- Through meticulous data collection from the SpaceX API and Wikipedia, a rich dataset was compiled. Exploratory data analysis revealed significant trends: SpaceX's landing success rate has shown a clear upward trajectory since 2013, and different launch sites and orbit types exhibit varying success patterns.
- Four machine learning models—Logistic Regression, SVM, Decision Tree, and KNN—were implemented, optimized, and evaluated. All achieved an identical test accuracy of 83.33%, demonstrating strong predictive capability for landing success. While test accuracies were uniform, the Decision Tree showed a promising higher cross-validation score.
- Future work includes increasing data volume, advanced feature engineering, and exploring deep learning for enhanced accuracy and interpretability. This project establishes a solid foundation for understanding and predicting Falcon 9 landing outcomes.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!