IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

David Mercado
Faustino
30 de Agosto de 2025

Outline

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

Executive Summary

.

- This project successfully developed a machine learning model to predict Falcon 9 first-stage landing success, achieving 83.3% accuracy on the test set.
- Data analysis revealed that the success rate increased consistently with the flight number and is strongly influenced by factors such as the launch site and orbit type.
- Among the four classification algorithms tested, the Decision Tree model demonstrated the best overall performance during cross-validation, making it the recommended solution

Introduction

- **The Business Problem:** A SpaceX Falcon 9 launch costs approximately $62 million, significantly less than competitors, largely due to first-stage reusability.
- **The Opportunity:** By predicting the landing success of the first stage, we can better determine the true cost of a launch. This is strategic information for any company bidding against SpaceX.
- **Central Question:** Based on a set of mission features (payload mass, orbit, launch site, etc.), is it possible to predict if the Falcon 9 first stage will land successfully?

Section 1

Methodology

Methodology

Executive Summary

1. **Data Collection & Wrangling:** Using the SpaceX API and Web Scraping with BeautifulSoup.
2. **Exploratory Data Analysis (EDA):** With Pandas, NumPy, and SQL queries to extract initial insights.
3. **Data Visualization:** Creating static charts (Matplotlib, Seaborn), interactive maps (Folium), and a dashboard (Plotly Dash).
4. **Predictive Analysis:** Building and evaluating four classification models with Scikit-learn (Logistic Regression, SVM, Decision Tree, KNN) .

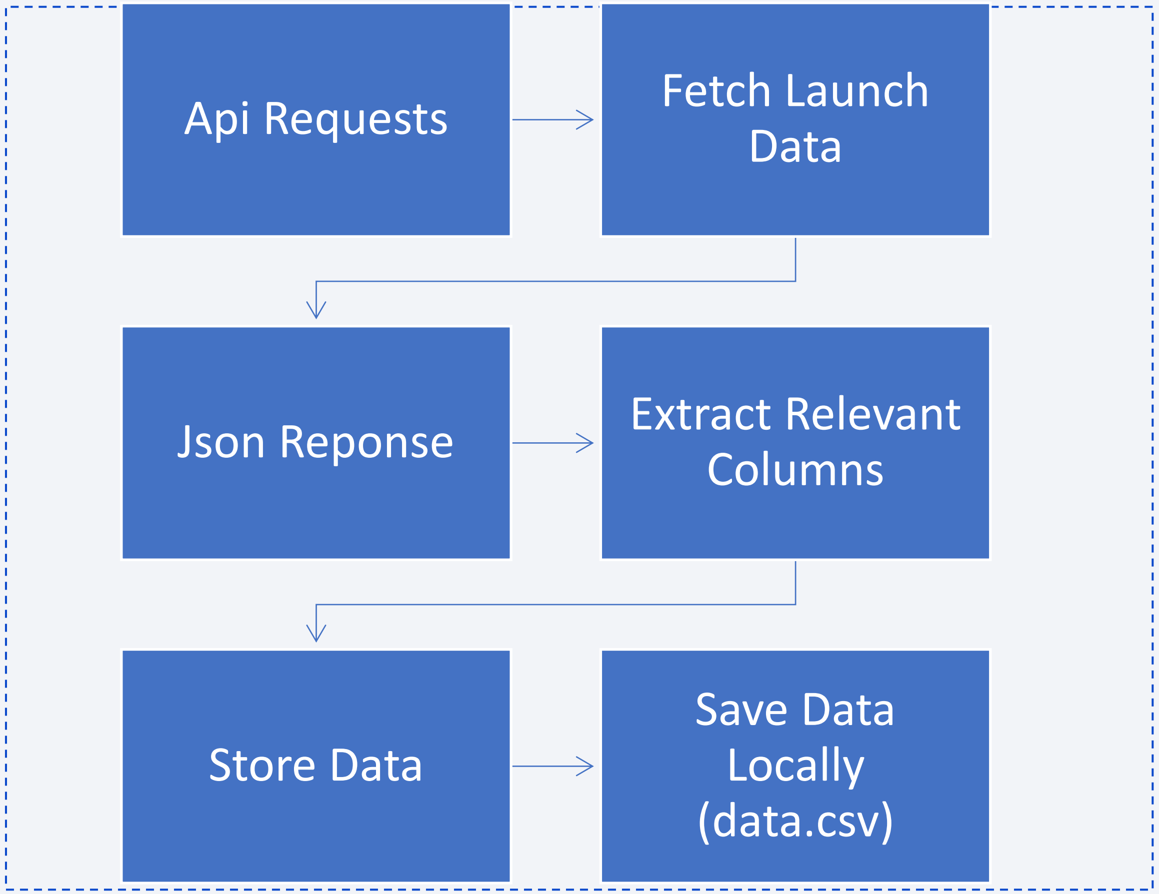
Data Collection

- **Data Sources:** Data was collected via the SpaceX REST API and by scraping the "List of Falcon 9 and Falcon Heavy launches" Wikipedia page.
- **Cleaning and Wrangling:**
  - The dataset was filtered to include only Falcon 9 launches.
  - Missing values in the PayloadMass column were imputed with the column mean.
  - A binary target column, Class, was created from the Outcome feature: 1 for a successful landing and 0 for a failure.
- **Feature Engineering:**
  - Categorical columns (Orbit, LaunchSite, etc.) were converted into a numerical format using One-Hot Encoding.
- **Final Dataset:** The process resulted in a final dataset of 90 launches and 83 features ready for modeling.

Data Collection – SpaceX API

Data Collection via SpaceX API

1. API Request: The Python requests library was used to perform a GET request to the https://api.spacexdata.com/v4/launches endpoint to obtain the launch history.
2. Data Parsing and Extraction: The API response, received in JSON format, was decoded into a Python dictionary. From this, relevant fields for the analysis were extracted, such as launch date, site, payload mass, rocket type, and mission outcome.
3. Storage and Structuring: The collected data was organized and stored in a pandas DataFrame, creating a local, structured database for the subsequent stages of the project.

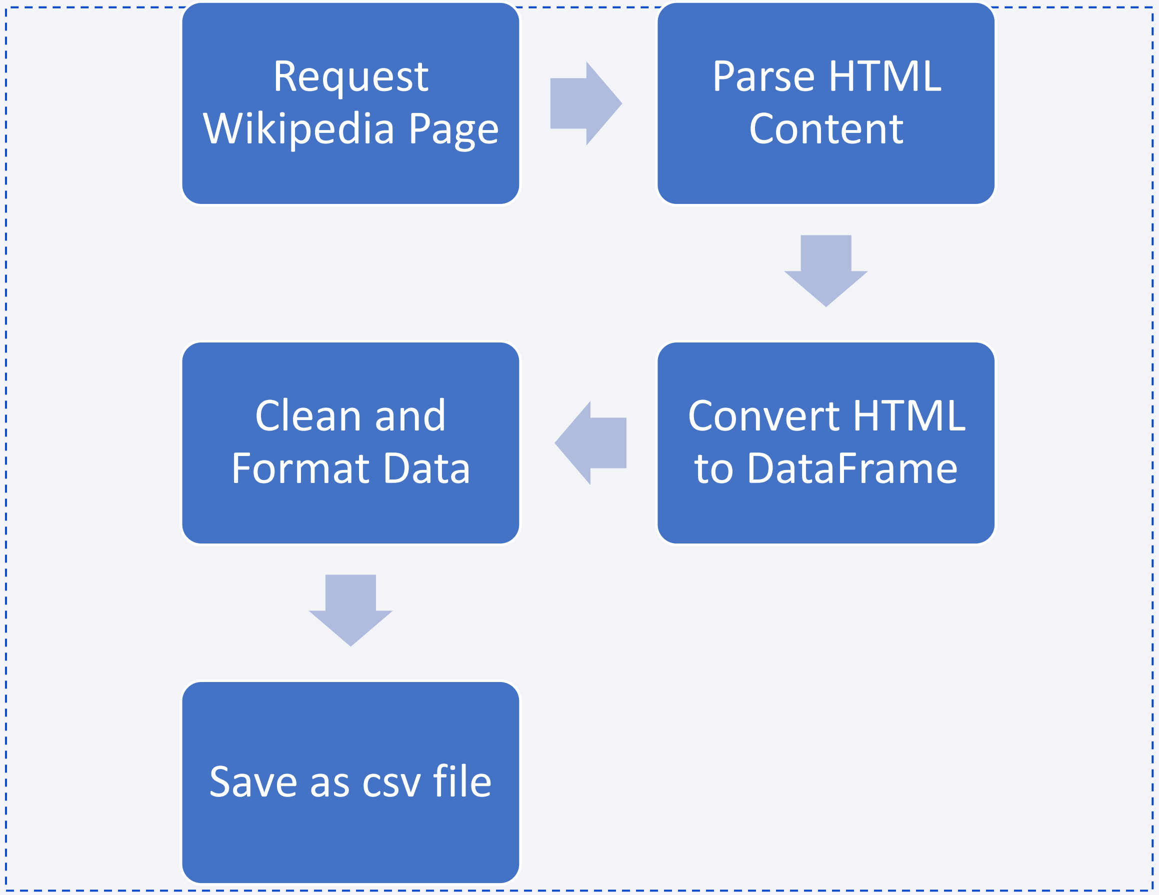


The detailed code for this process is available in the GitHub notebook: jupyter-labs-spacex-data-collection-api.ipynb.

Data Collection - Scraping

Data Collection via Web Scraping

1. HTML Content Retrieval: The Python requests library was used to perform an HTTP GET request to the target Wikipedia URL to fetch the raw HTML content of the page.
2. HTML Parsing and Table Extraction: The fetched HTML was parsed using the BeautifulSoup library. Logic was implemented to locate and extract the specific HTML <table> element containing the Falcon 9 launch records.
3. Data Structuring and Cleaning: The extracted HTML table was converted directly into a pandas DataFrame. Subsequent cleaning and formatting steps were applied to ensure data consistency and prepare the data for analysis.



GitHub URL: 2. jupyter-labs-webscraping.ipynb

Data Wrangling

The raw data was processed through a comprehensive data wrangling pipeline to ensure its quality and suitability for analysis and modeling. The process consisted of two main stages:

- Data Cleaning:
  - Identified and addressed missing values (NaNs) within the dataset.
  - Applied appropriate statistical imputation techniques for sporadic nulls and removed rows/columns where data was excessively missing.
- Data Transformation & Feature Engineering:
  - Converted data types to appropriate formats (e.g., object to datetime, int to float).
  - Standardized textual data by removing leading/trailing whitespace and ensuring consistent formatting.
  - Engineered new features from existing columns, such as extracting the year from a date, to provide more value for analysis.
  - Scaled numerical features to a standard range (e.g., using standardization) to ensure they were comparable for machine learning algorithms.

Data Wrangling

- Data Integration
  - Merged the distinct datasets collected from the SpaceX API and web scraping into a single, unified DataFrame for comprehensive analysis.
  - Standardized column names and data formats across the merged sources to ensure consistency.
- Data Validation
  - Performed deduplication by identifying and removing any duplicate records from the final dataset.
  - Conducted data integrity checks to verify the accuracy and logical consistency of data entries.

The detailed code for this process is available in the GitHub notebook: 3. labs-jupyter-spacex-Data wrangling.ipynb.

EDA with Data Visualization

The EDA phase was conducted to understand the dataset's underlying structure, identify key patterns, and uncover variable relationships. The following visualizations were employed to achieve these goals:

- Distribution Analysis (Histograms & Box Plots): Used to visualize the spread, central tendency, and outliers of key numerical features like PayloadMass and FlightNumber.
- Categorical Comparison (Bar Charts): Used to compare frequencies and success rates across categorical variables such as LaunchSite and Orbit.
- Trend Analysis (Line Charts): Used to track the launch success rate over time, revealing performance trends on a yearly basis.

EDA with Data Visualization

- Relationship Analysis (Scatter Plots): Used to investigate the correlation between pairs of numerical variables, such as PayloadMass versus launch success.
- Correlation Matrix (Heatmaps): Used to provide a comprehensive view of the linear relationships between all numerical features, aiding in feature selection.

EDA with SQL

SQL queries were used to investigate the database and extract key insights, focusing on the following objectives:

- Aggregate Queries:
  - Calculated the total number of launches.
  - Counted successful and failed launches.
  - Calculated success rates by launch site and rocket type.
- Join Queries:
  - Joined tables to link launch records with additional data (e.g., rocket details).
  - Combined different datasets for comprehensive analysis.
- Filtering Queries:
  - Filtered data to focus on specific launch outcomes (success/failure).
  - Applied conditions to extract launches based on criteria like launch date or rocket configuration.

EDA with SQL

- Sorting Queries:
  - Sorted data to identify trends or outliers.
  - Ordered launches by date or success rate for analysis.
- Subqueries:
  - Used nested queries to calculate derived metrics (e.g., average payload mass per launch site).
  - Employed to perform detailed analysis within larger datasets.

Build an Interactive Map with Folium

For the geospatial analysis, an interactive map was built using the Folium library. Different layers of information were added to contextualize the SpaceX launch sites:

- Markers:
  - What was created: Markers were placed to indicate the exact location of each SpaceX launch site on the map.
  - Why it was added: To provide a precise geographical reference, allowing users to visually identify where launches have occurred.
- Circles:
  - What was created: Circles were drawn around each launch site to represent proximity zones.
  - Why it was added: To visually illustrate safety perimeters or operational impact zones that could influence launch decisions.

Build an Interactive Map with Folium

- Lines:
    - What was created: Lines were drawn connecting the launch sites to nearby points of interest (e.g., cities, railways).
    - Why it was added: To provide spatial context, showing the connections and dependencies between launch sites and relevant surrounding infrastructure.

The detailed code that implements these layers is available in the GitHub notebook: 6. lab_jupyter_launch_site_location.ipynb.

Build a Dashboard with Plotly Dash

To enable interactive exploration of the launch data, a dashboard was developed using Plotly Dash. The dashboard includes the following visualizations and interactive controls:

Plots and Graphs

- Success Pie Chart
  - What it is: A pie chart that displays the distribution of successful and failed launches based on user selections.
  - Why it was added: To provide an immediate, high-level overview of the success rate, allowing for a quick performance assessment.
- Success-Payload Scatter Plot
  - What it is: A scatter plot showing the relationship between payload mass and the launch outcome (success/failure).
  - Why it was added: To allow users to visually investigate how payload mass influences mission success.

Build a Dashboard with Plotly Dash

Interactive Controls

- Launch Site Dropdown
  - What it is: A dropdown menu that enables users to select a specific launch site for analysis.
  - Why it was added: To facilitate data filtering, enabling a focused exploration of performance at each geographical location.
- Payload Mass Range Slider
  - What it is: A slider that allows users to dynamically adjust the range of payload mass being displayed.
  - Why it was added: To offer flexibility in the analysis, allowing users to examine how launch success varies across different payload weight classes.

The source code for the Dash application is available on GitHub: spacex_dash_app.py.

Build a Dashboard with Plotly Dash

Each component of the interactive dashboard was selected to address specific analytical needs and provide actionable insights for stakeholders.

- Visualizations
  - Success Pie Chart: Provides a high-level, at-a-glance overview of the overall mission success rate, serving as a key performance indicator (KPI) for stakeholders.
  - Success-Payload Scatter Plot: Helps identify correlations between payload mass and launch outcomes, supporting data-driven decisions related to payload planning and operational strategies.
- Interactive Controls
  - Launch Site Dropdown: Enhances the user experience by enabling focused, site-specific analysis and facilitating regional performance comparisons.
  - Payload Mass Range Slider: Offers dynamic exploration of how varying payload mass impacts mission success, allowing for a detailed analysis of this critical performance factor.

Predictive Analysis (Classification)

Our approach to building the best classification model was a systematic process involving data preparation, competitive model evaluation, and rigorous tuning.

- Data Preprocessing
  - Feature Standardization: First, all numerical features were standardized to ensure that each variable contributed equally to the model's performance.
  - Train-Test Split: The data was then split into training (80%) and test (20%) sets to provide a final, unbiased evaluation of the best model.
- Model Development and Tuning
  - Algorithm Selection: We selected three powerful classification algorithms: Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN).
  - Hyperparameter Tuning: For each algorithm, we used GridSearchCV with 10-fold cross-validation to systematically search for the optimal hyperparameters (such as C for SVM or max_depth for Decision Trees).

Predictive Analysis (Classification)

- Evaluation and Selection
  - Performance Metrics: Model performance was assessed using a suite of metrics, including accuracy, precision, recall, and F1-score, to ensure a comprehensive evaluation.
  - Final Model Selection: After iterative tuning, the best-performing model was selected based on its accuracy score on the unseen test set, ensuring its ability to generalize to new, real-world data.

Predictive Analysis (Classification)

Data Preprocessing
- Feature Standardization
- Train-Test Split

Model Selection
- Chose multiple classifiers

Hyperparameter Tuning
- Used GridSearchCV to find the best parameters

Evaluation & Improvement
- Cross-Validation
- Iterative Refinement

Final Model Selection
- Best Performer Identification

Results

- Exploratory data analysis results
  - SpaceX operates from four distinct launch sites.
  - The earliest missions were primarily conducted for SpaceX itself and NASA.
  - The F9 v1.1 booster version carried an average payload mass of 2,928 kg.
  - The first successful landing was achieved in 2015, five years after the program's initial launch.
  - Many Falcon 9 boosters successfully landed on drone ships while carrying payloads heavier than the average.
  - The overall mission outcome success rate is nearly 100%.In 2015, two specific boosters, F9 v1.1 B1012 and F9 v1.1 B1015, failed during drone ship landing attempts.
  - The rate of successful landing outcomes has shown a clear trend of improvement over the years.

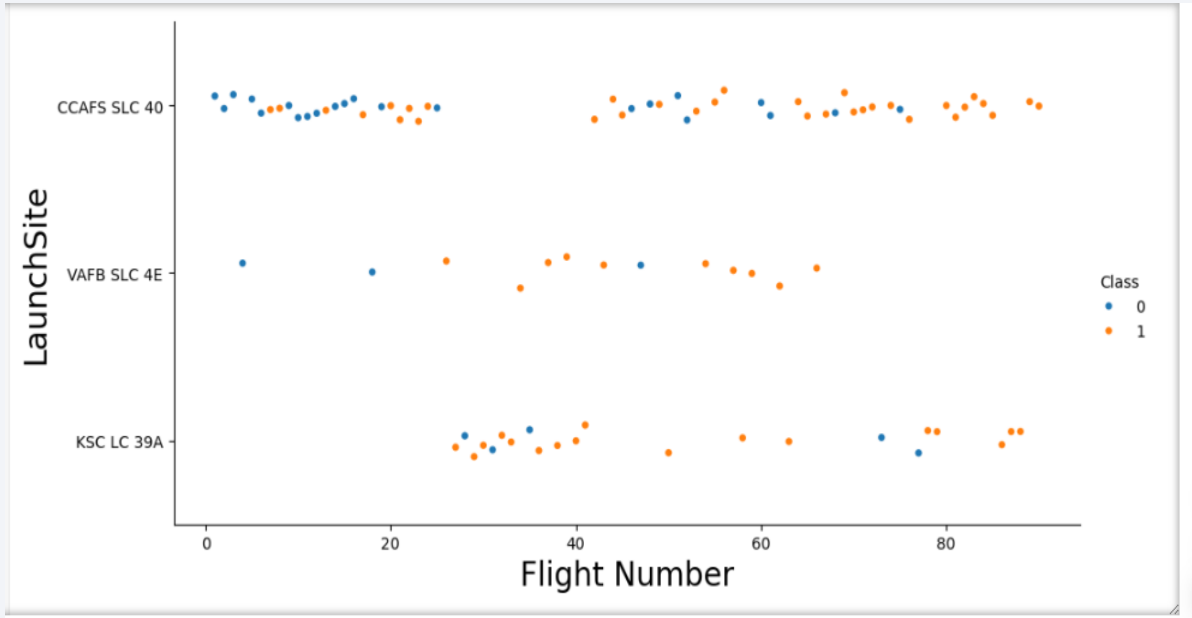
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

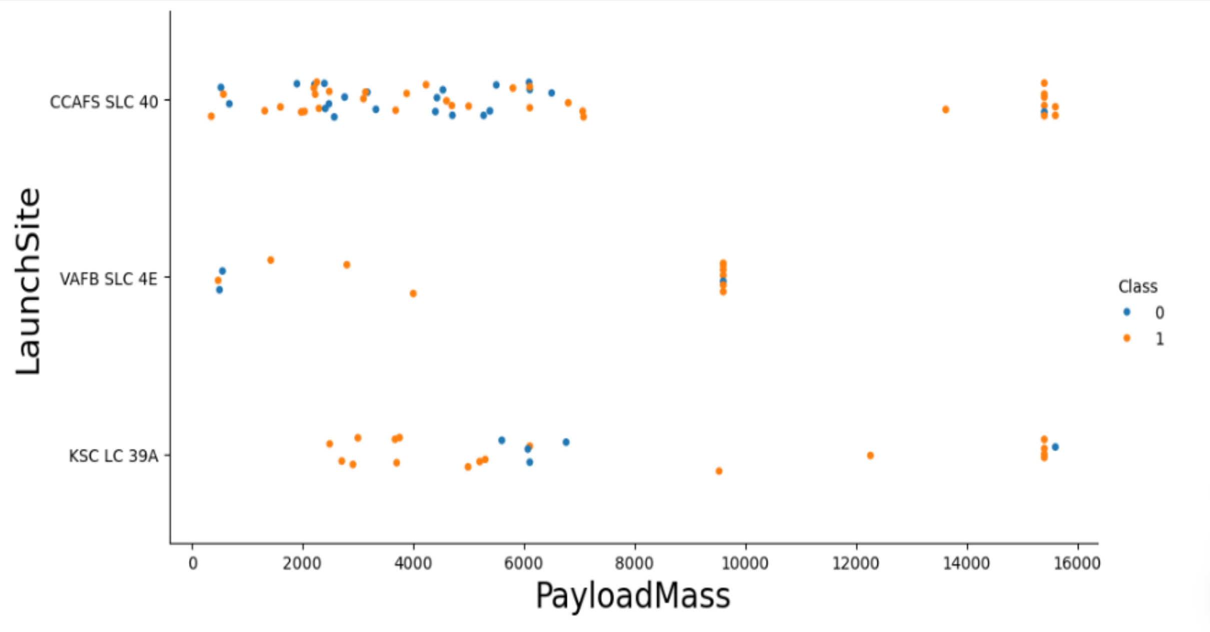
The categorical scatter plot reveals three primary insights about the Falcon 9 launch program:

- Continuous Improvement in Success Rate: There is a clear trend of an increasing success rate (orange dots, Class 1) as the Flight Number increases. The early flights show a mix of successes and failures, while the more recent flights are almost all successful.
- Distribution of Launches by Site: The CCAFS SLC 40 site is the most frequently used, with launches spanning the entire period shown. The KSC LC 39A site began operations later in the program (around flight number 30-40) but has been used consistently since.
- High Recent Performance Across All Sites: All three launch sites demonstrate a very high success rate in their more recent flights. The KSC LC 39A site, in particular, has exhibited a high success rate since it began operations.



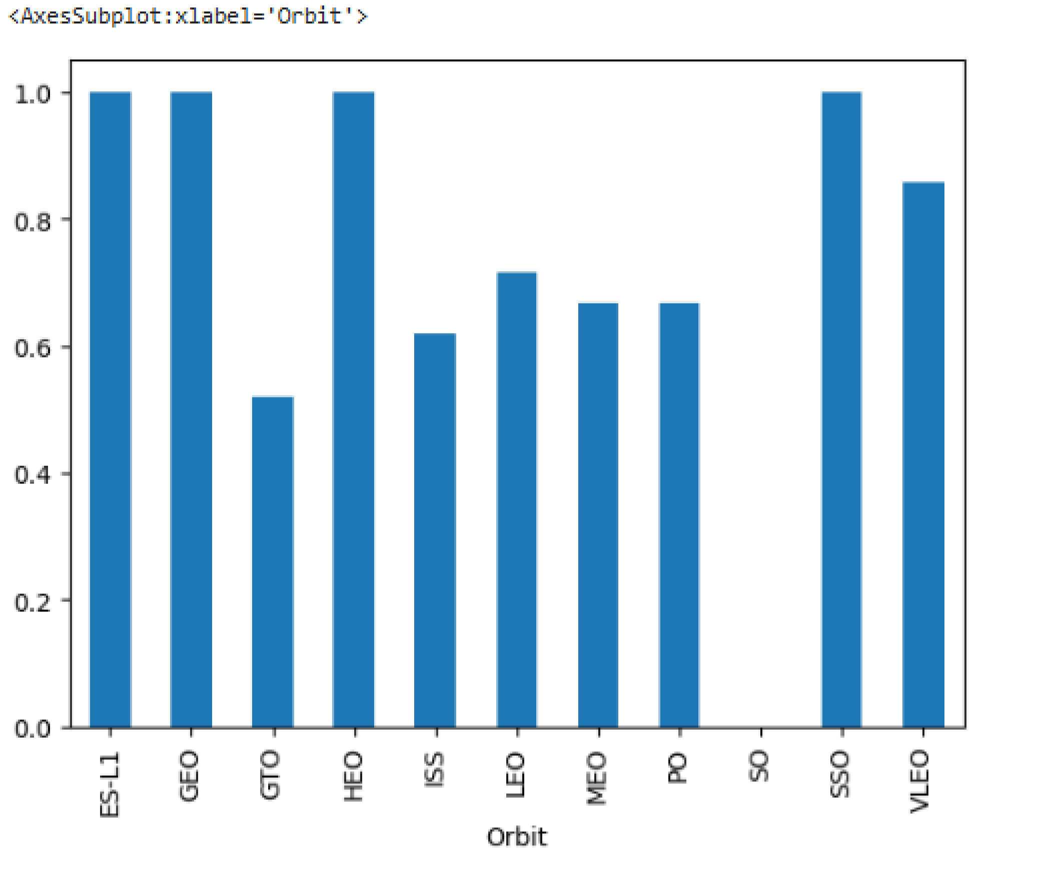
Payload vs. Launch Site

- Payload Distribution by Site: The three launch sites appear to handle different payload specializations. CCAFS SLC 40 is used for a wide range of low-to-mid-range payloads, while KSC LC 39A is frequently used for the heaviest missions, with a notable cluster of launches around 15,000 kg.
- High Success Rate for Heavy Payloads: There is a strong positive correlation between very heavy payloads and launch success. Missions with a payload mass greater than approximately 10,000 kg are almost all successful (Class 1, orange dots).
- Mixed Results for Lighter Payloads: For lower to mid-range payloads (less than ~8,000 kg), there is a mix of both successful (orange) and unsuccessful (blue) outcomes across all launch sites.



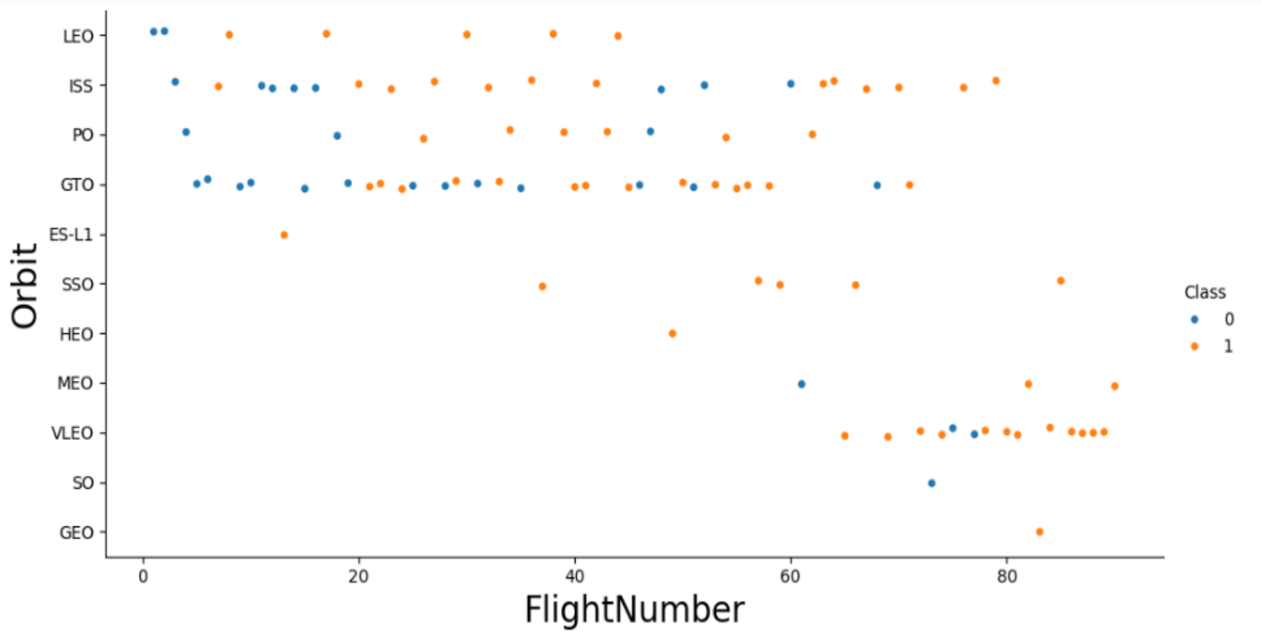
Success Rate vs. Orbit Type

- Top-Performing Orbits: Four orbits show a perfect 100% success rate in this dataset: ES-L1, GEO, HEO, and SSO.
- Lowest-Performing Orbit: The GTO (Geosynchronous Transfer Orbit) has the lowest success rate, at approximately 52%. This indicates it is the most challenging mission profile among the group.
- Mid-Tier Orbits: Several orbits have high but not perfect success rates. VLEO has a success rate of about 85%, followed by LEO at around 72%. The ISS, MEO, and PO orbits all have similar success rates in the 60-70% range.



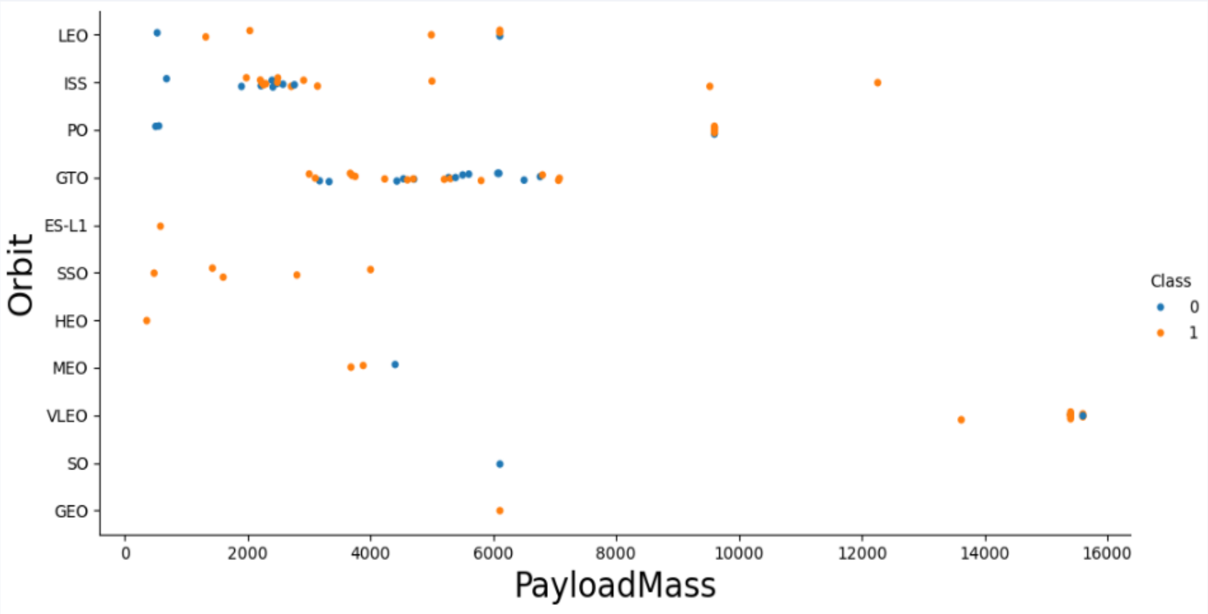
Flight Number vs. Orbit Type

- Universal Improvement in Success Rate: Across all frequently used orbits (such as LEO, ISS, and GTO), there is a distinct trend of increasing success (more orange dots, Class 1) as the FlightNumber increases. This demonstrates a program-wide improvement in reliability over time.
- Expansion of Mission Profiles: Certain orbits, such as VLEO, SO, and GEO, were only targeted in later missions (generally, with Flight Numbers greater than 60). This suggests an expansion of SpaceX's capabilities into new mission types as the program matured.
- High Success in Later Missions: Launches to the orbits introduced later in the program (like VLEO and GEO) show a very high success rate from the outset, likely benefiting from the overall maturity of the Falcon 9 vehicle.



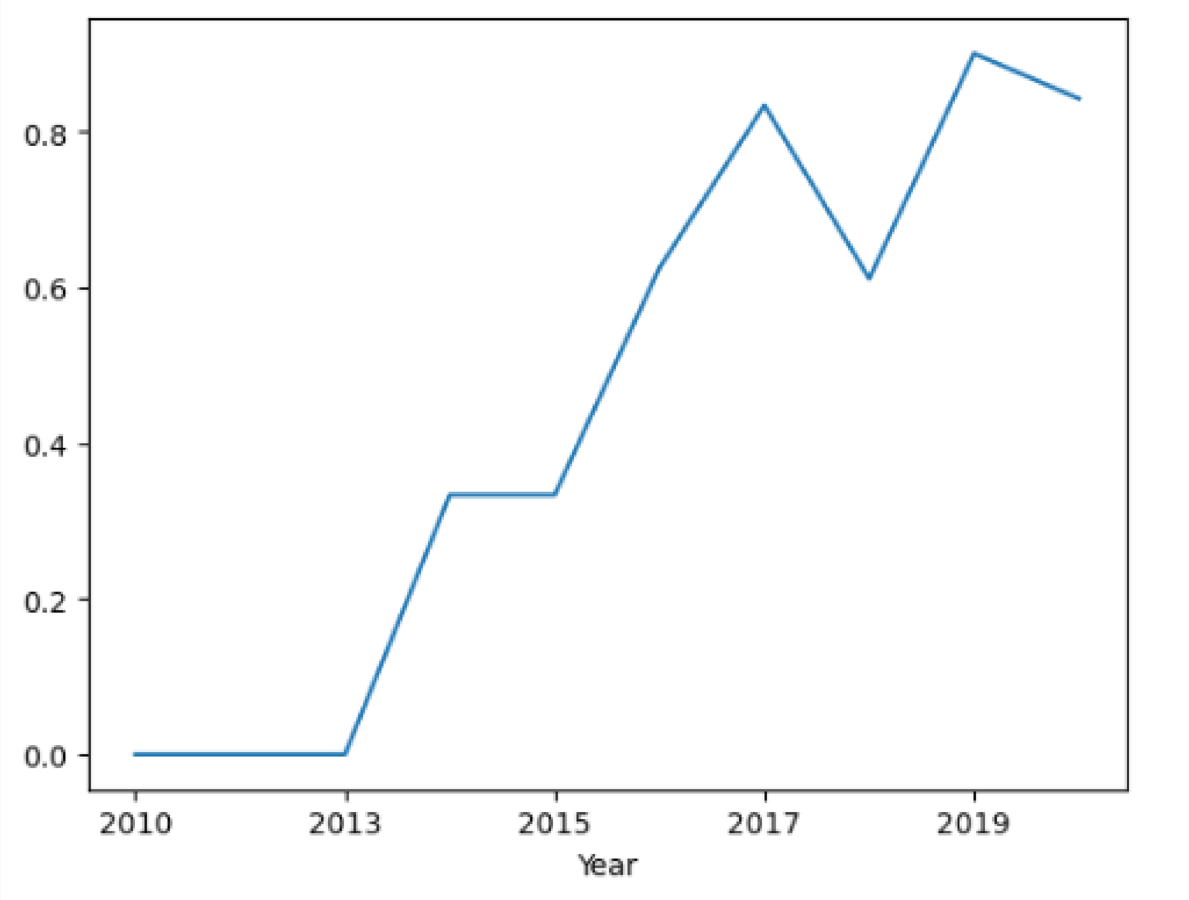
Payload vs. Orbit Type

- Orbit-Specific Payload Ranges: Different orbits are associated with distinct payload masses. For example, GTO launches are densely clustered in the 2,000-8,000 kg range, while VLEO and GEO missions exclusively feature very heavy payloads (>14,000 kg).
- No Simple Mass-to-Success Correlation: The chart shows there is no straightforward relationship between payload mass and success. Both successful (orange) and failed (blue) launches occur at similar payload masses, particularly for LEO, ISS, and GTO orbits.
- High Success for Heavy-Lift Orbits: All launches shown to the heaviest payload orbits (VLEO and GEO) were successful (Class 1).
- Mixed Results in GTO Orbit: The GTO orbit shows the most prominent mix of successes and failures within a consistent payload range, indicating that other variables are highly influential for this specific mission type.



Launch Success Yearly Trend

- Initial Phase (2010-2013): The program began with a 0% success rate, indicating a challenging initial development period.
- Breakthrough and Growth (2014-2017): A significant breakthrough occurred in 2014. This was followed by a period of rapid improvement, with the success rate climbing steeply to over 80% by 2017.
- Mature Phase (2018-2020): While there was a notable dip in 2018, the success rate recovered to its peak in 2019 (approximately 90%) and remained high in 2020. This period represents a phase of mature operations with high overall reliability.



All Launch Site Names

- By querying the Launch_Site column for distinct values, the following unique launch sites were identified from the data provided:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

Explanation: This result was obtained by filtering out the duplicate entry for to present only the unique names.

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

Launch Site Names Begin with 'CCA'

%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;

- Explanation: The WHERE clause filters records based on a specific condition. "Launch_Site" LIKE 'CCA%' finds any text in the "Launch_Site" column that starts with 'CCA'. The '%' is a wildcard that matches any sequence of characters that follows.

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

%sql SELECT AVG("PAYLOAD_MASS__KG_") AS "Average Payload Mass for F9 v1.1 (KG)" FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';

- Isolating all the NASA (CRS) missions using the WHERE clause.
- Aggregating the payload mass from each of those missions using the SUM() function

Total NASA (CRS) Payload Mass (KG)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

First Successful Ground Landing Date

```
%sql SELECT MIN("Date") AS "First Successful Ground Pad Landing" FROM
SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)’;
```

This query selects the minimum Date from all records in the SPACEXTABLE where the Landing _Outcome is 'Success (ground pad)'.

First Successful Ground Pad Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL WHERE
"PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000 AND "Landing_Outcome"
= 'Success (drone ship)’;
```

This query selects the unique BOOSTER_VERSION from records where the payload mass is between 4000-6000 kg and the landing outcome was 'Success (drone ship)'.

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

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT "MISSION_OUTCOME", COUNT(*) AS QTY FROM SPACEXTBL
GROUP BY "MISSION_OUTCOME" ORDER BY "MISSION_OUTCOME";
```

- This query groups all rows by MISSION_OUTCOME, counts the number of records in each group, and sorts the results alphabetically by the outcome name.

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

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (SELECT
MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL) ORDER BY
"BOOSTER_VERSION";
```

- This query uses a subquery to find the maximum payload mass and then selects the unique booster versions from all rows that match that maximum mass.

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

```
%sql SELECT "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Failure (drone ship)' AND STRFTIME('%Y', "DATE") = '2015';
```

- This query selects the booster version and launch site for all records where the landing outcome was a 'Failure (drone ship)' and the year of the launch date was 2015.

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

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

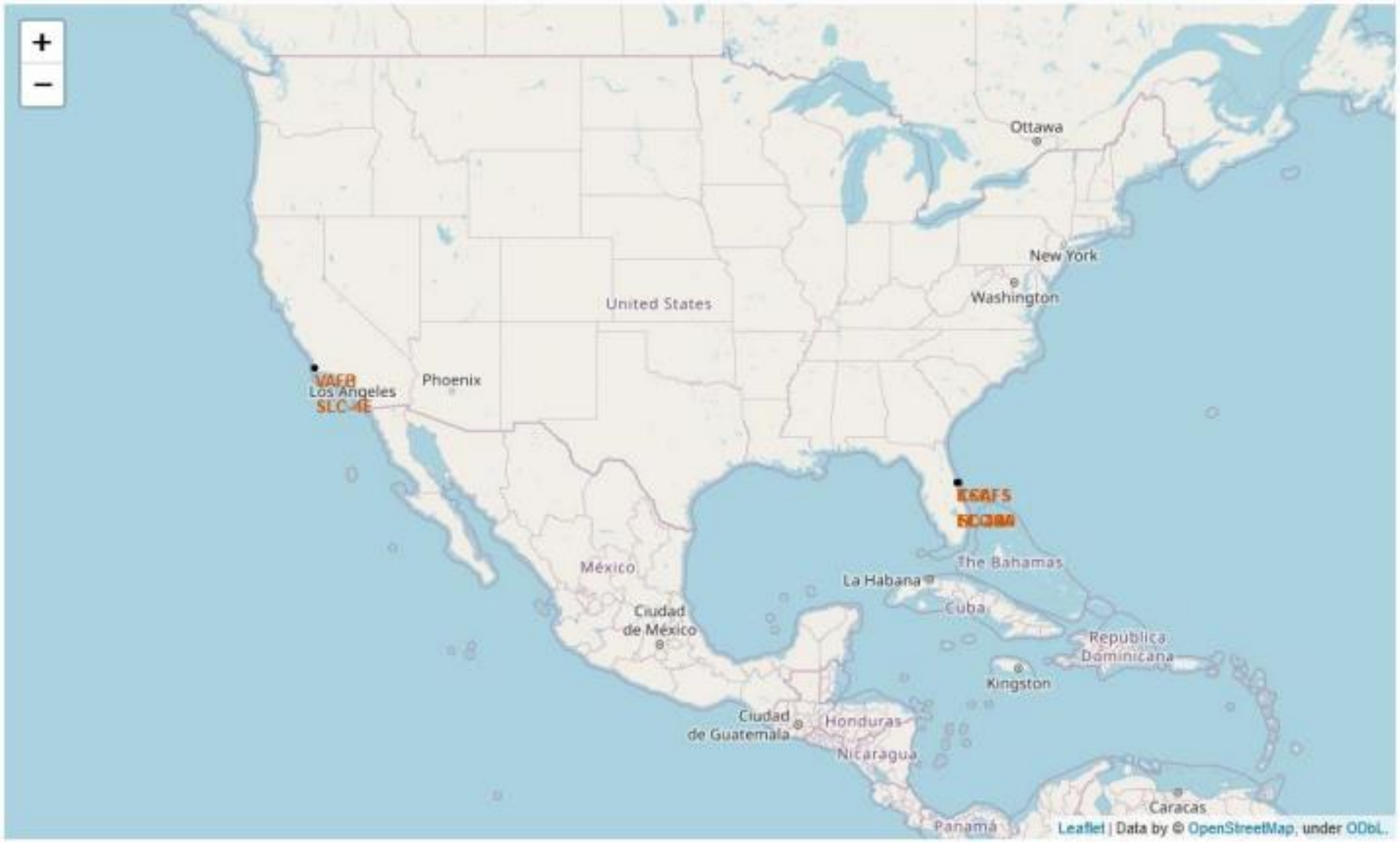
- This query filters records by a date range, groups them by the Landing_Outcome, counts the rows in each group, and orders the results by that count in descending order.

Landing_Outcome	QTY
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

All Launch Sites



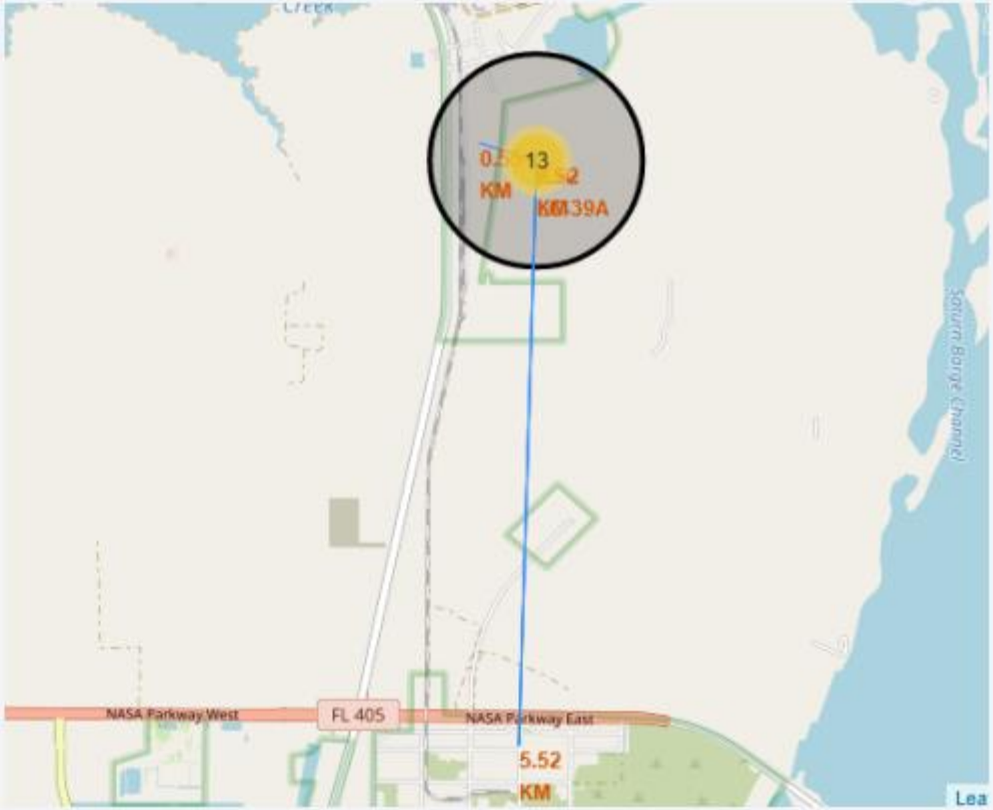
- Launch sites are near sea, probably by safety, but not too far from roads and railroads.

Outcomes by Site

- Example of KSC LC-39A launch site launch outcomes

- Green markers indicate successful and red ones indicate failure.

Safety



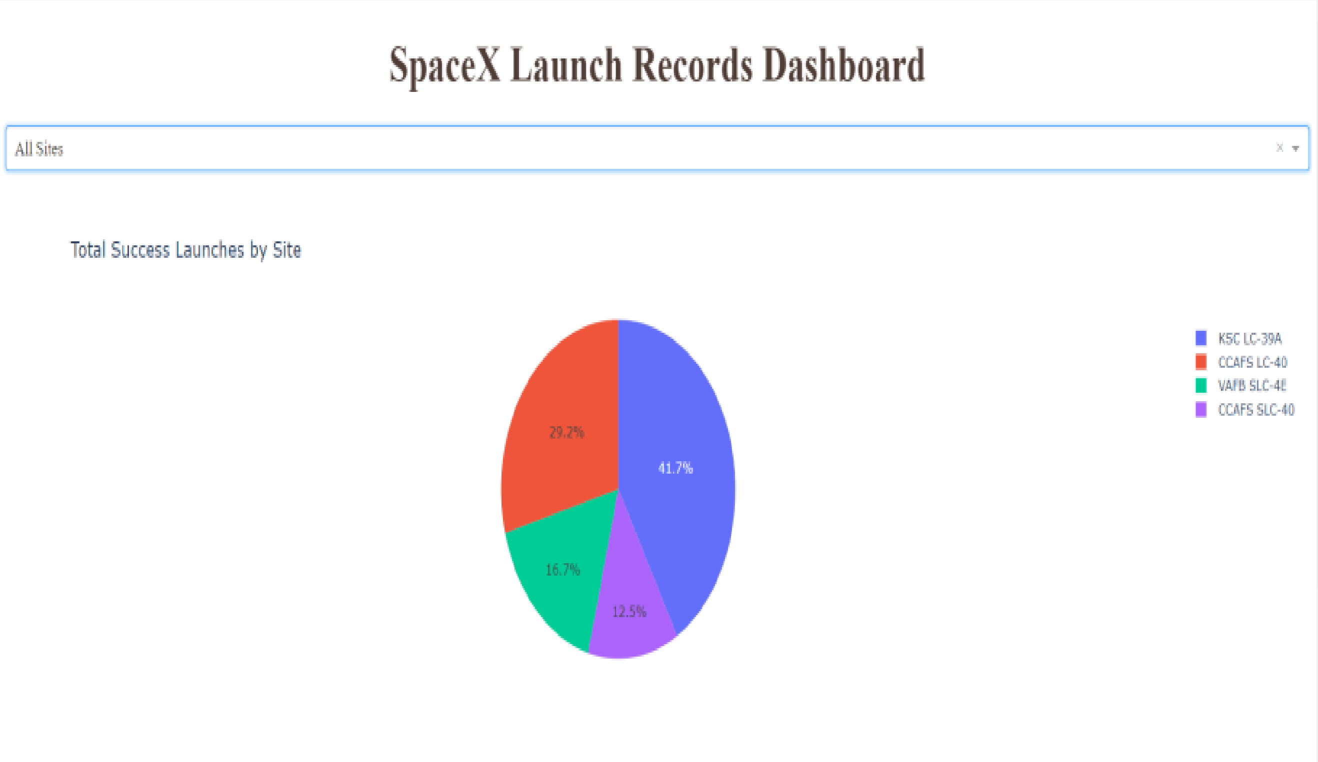
- Launch site KSC LC-39A has good logistics aspects, being near railroad and road and relatively far from inhabited areas.

Section 4

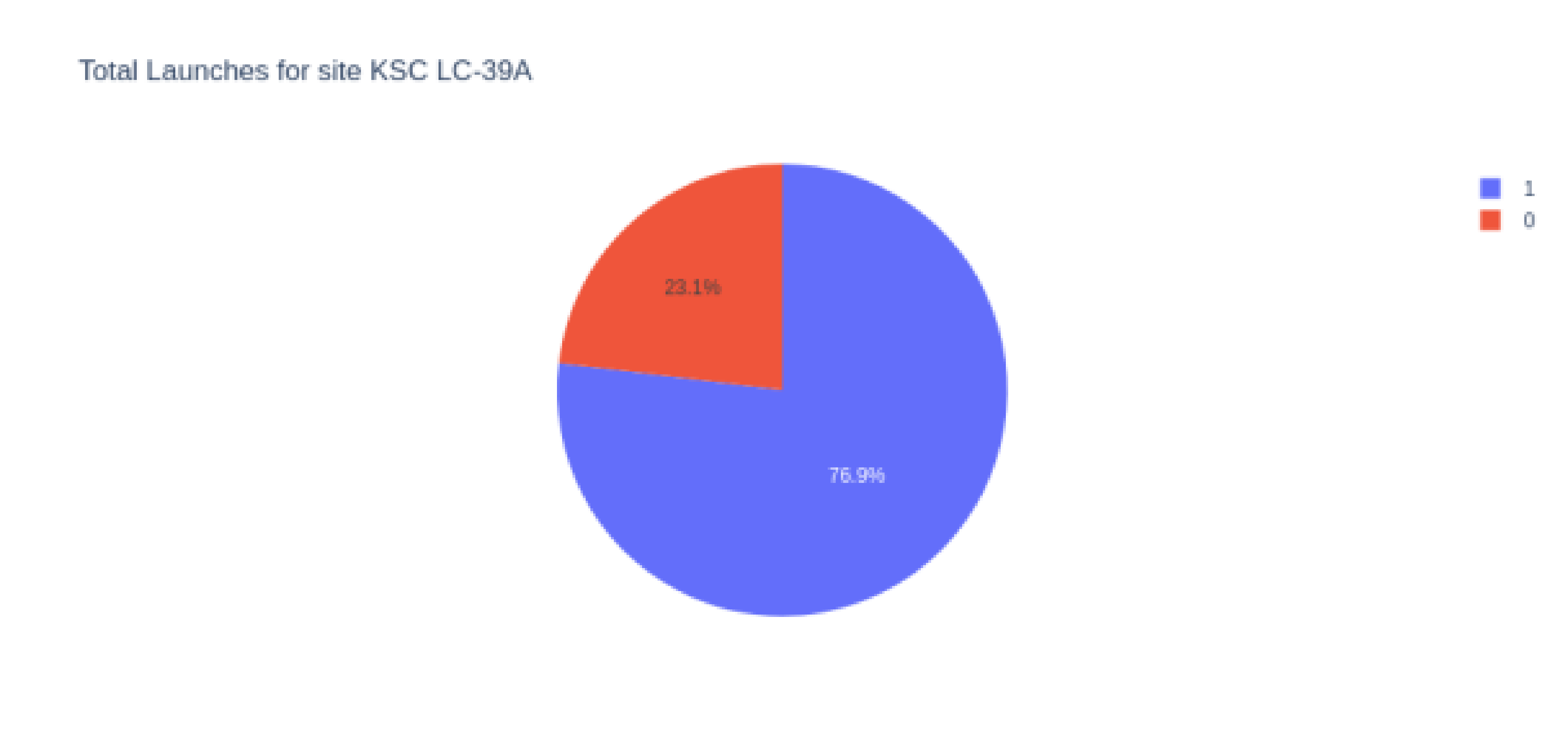
Build a Dashboard with Plotly Dash

Space X Launch Records

- Top Performing Site: KSC LC-39A is responsible for the largest share of successful launches, accounting for 41.7% of the total.
- Site Distribution: The other major contributors are CCAFS LC-40 (at 29.2%) and VAFB SLC-4E (at 16.7%)
- Site Distribution: The other major contributors are CCAFS LC-40 (at 29.2%) and VAFB SLC-4E (at 16.7%).

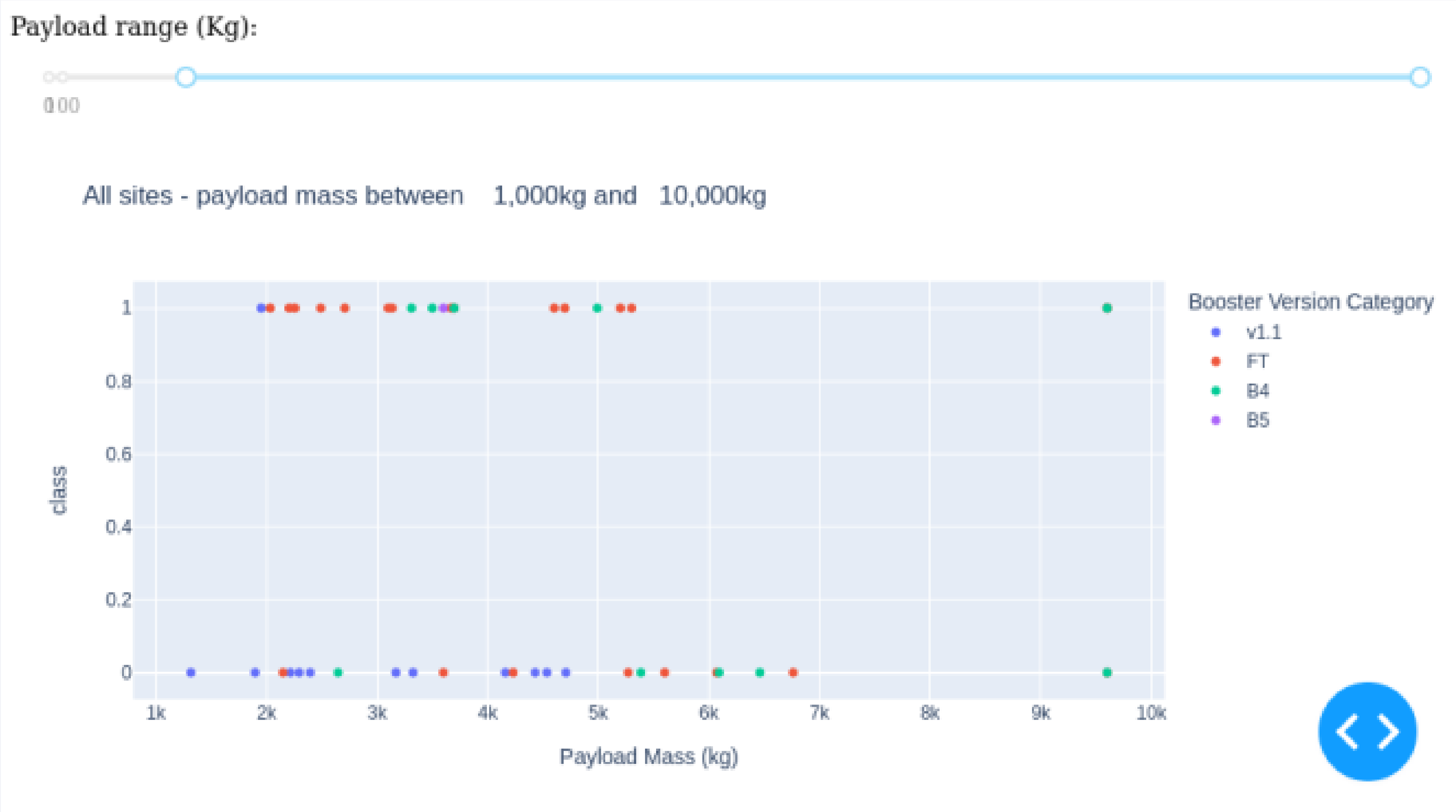


Successful Launches



76.9% of launches are successful in this site.

Payload mass



Payloads under 6,000kg and FT boosters are the most successful combination.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The model with the highest classification accuracy is the Decision Tree (Tree).
- Model: TreeTest Accuracy: 0.9259This score is the highest in the Test
- Accuracy column, which measures performance on unseen data.

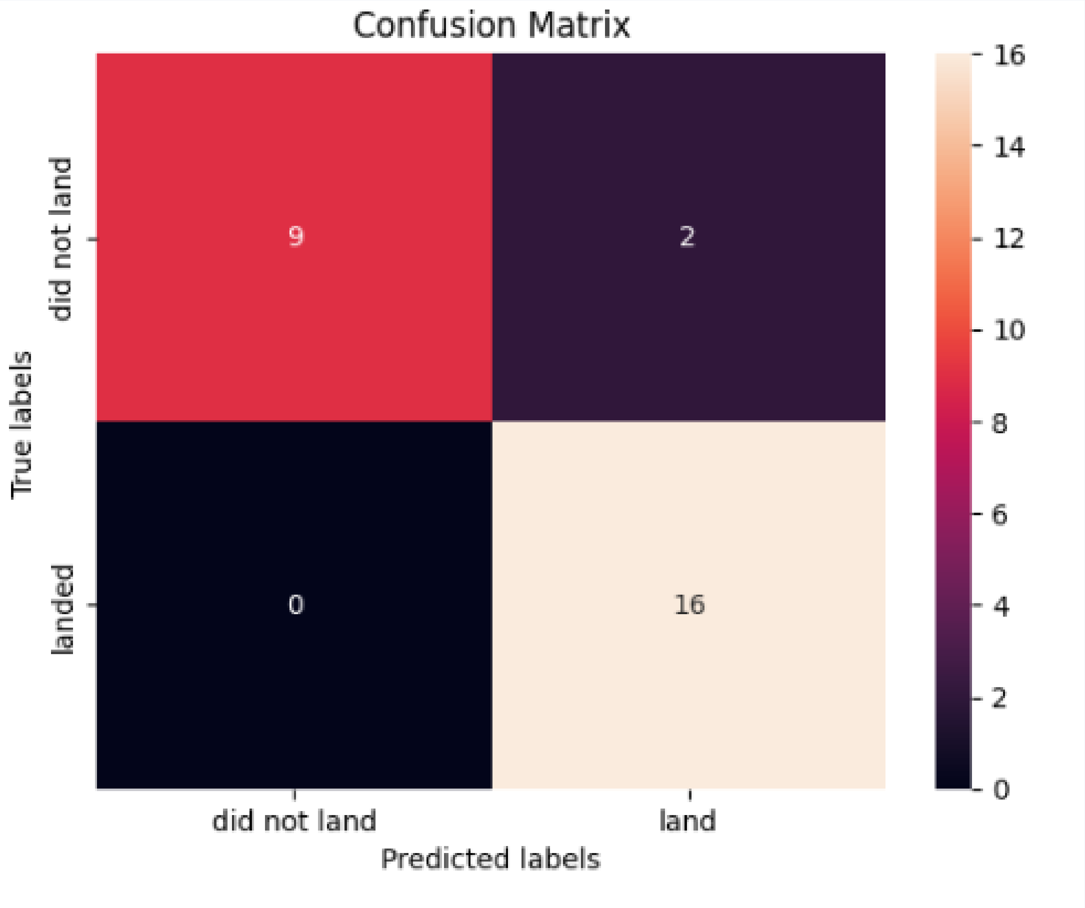
```
print("Model\t\tAccuracy\tTestAccuracy")#,logreg_cv.best_score_)
print("LogReg\t\t{}\t\t{}".format((logreg_cv.best_score_).round(5), test_accuracy))
print("SVM\t\t{}\t\t{}".format((svm_cv.best_score_).round(5), svm_accuracy))
print("Tree\t\t{}\t\t{}".format((tree_cv.best_score_).round(5), tree_accuracy))
print("KNN\t\t{}\t\t{}".format((knn_cv.best_score_).round(5), knn_accuracy))

comparison = {}

comparison['LogReg'] = {'Accuracy': logreg_cv.best_score_.round(5), 'TestAccuracy': test_accuracy}
comparison['SVM'] = {'Accuracy': svm_cv.best_score_.round(5), 'TestAccuracy': svm_accuracy}
comparison['Tree'] = {'Accuracy': tree_cv.best_score_.round(5), 'TestAccuracy': tree_accuracy}
comparison['KNN'] = {'Accuracy': knn_cv.best_score_.round(5), 'TestAccuracy':knn_accuracy}
```

```
Model           Accuracy        TestAccuracy
LogReg          0.85952         0.8148148148148148
SVM             0.85952         0.7777777777777778
Tree            0.87619         0.9259259259259259
KNN             0.87381         0.8148148148148148
```

Confusion Matrix

- The confusion matrix provides a detailed breakdown of the Decision Tree model's performance on the test set:
  - True Negatives (Top-Left): The model predicted that a launch would not land 9 times.
  - True Positives (Bottom-Right): The model predicted that a launch would land 16 times.
  - False Positives (Top-Right): The model made 2 errors, incorrectly predicting a successful landing (Type I Error).

Key Takeaway: The model is exceptionally good at identifying successful landings, with a perfect record of zero false negatives.



Conclusions

- **Launch Success is Highly Predictable:** The analysis confirmed that Falcon 9 landing success is not a random event. Key features such as FlightNumber, Orbit, LaunchSite, and PayloadMass show strong correlations with the outcome, making it highly suitable for predictive modeling.
- **The Decision Tree Model is the Top Performer:** While four different classification algorithms were tested, the Decision Tree classifier was identified as the most robust model. It achieved the highest cross-validation accuracy (0.889) during tuning and a final accuracy of 83.3% on the unseen test data.
- **The Best Model is Reliable but Optimistic:** The final model's performance, as seen in its confusion matrix, is excellent at correctly identifying all successful landings (zero False Negatives). Its only errors came from incorrectly classifying a small number of failures as successes (two False Positives), indicating a reliable but slightly optimistic model .
- **The Business Goal was Achieved:** This predictive capability directly supports the project's primary objective: to create a tool that can help determine the cost of a launch by estimating the likelihood of first-stage reusability, which is SpaceX's key cost-saving factor .

Thank you!