IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rayan Alam
2025-06-03

Outline

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

Executive Summary

The project began with web scraping using BeautifulSoup to extract SpaceX launch data from a Wikipedia table. We parsed the HTML by identifying <table> and <th> elements to extract column headers and iterated over <tr> rows to collect relevant launch data. During this process, we applied cleaning functions to remove citations, parse dates and times, extract payload mass, and filter out noisy or incomplete records. The data was then structured into a dictionary and converted into a Pandas DataFrame. Following this, we created a binary label (Class) to indicate launch success, and explored the data visually using seaborn's catplot and scatter plots to analyze relationships between variables such as Flight Number, Launch Site, Payload Mass, and Orbit. We also encoded categorical features using get_dummies() and standardized numerical features with StandardScaler. In the SQL phase, we loaded the cleaned data into a database and used SQL queries to answer questions about booster types, launch outcomes, and payload characteristics. We used subqueries and string functions like substr() to extract dates and identify specific records (e.g., the first successful ground pad landing). Using Folium, we created interactive maps marking all launch sites, utilized the MarkerCluster plugin to handle multiple close markers, and added features like mouse position display to enhance user interactivity. Later, with Plotly Dash, we built interactive dashboards incorporating dropdowns, sliders, and callbacks to enable dynamic data visualization and user-driven analysis of launch outcomes and payload data. We then transitioned into machine learning where we split the data into training and test sets using train_test_split, applied feature scaling, and trained four models using GridSearchCV: Logistic Regression, SVM, Decision Tree, and K-Nearest Neighbors. Each model was tuned using hyperparameter grids and evaluated using .score() on the test data. Surprisingly, all models returned the same accuracy, likely due to the dataset being simple or linearly separable. The best-performing kernel for the SVM was found using best_params_, typically returning 'rbf', but subject to the dataset's distribution. The project concluded with performance comparison, identification of optimal models and parameters, and a strong end-to-end pipeline from raw web data to fully tuned classification models integrated with interactive visual analytics tools.

Introduction

This project focuses on analyzing SpaceX launch data to gain insights into launch success factors, booster performance, and payload characteristics. SpaceX has rapidly evolved the commercial space industry by frequently launching rockets with various configurations, payloads, and landing strategies. Understanding these launches requires comprehensive data extraction, cleaning, and analysis from public sources like Wikipedia and structured databases. The project aims to answer key questions such as: Which launch sites are most active? How do payload mass and orbit type relate to launch success? What are the trends in booster versions and landing outcomes over time? Additionally, by leveraging machine learning models, the project seeks to predict launch success based on technical features, while interactive visualizations help communicate patterns effectively. Ultimately, this study addresses both the technical challenges of data extraction and the analytical challenges of uncovering actionable insights into rocket launches.

Section 1

Methodology

Methodology

Executive Summary

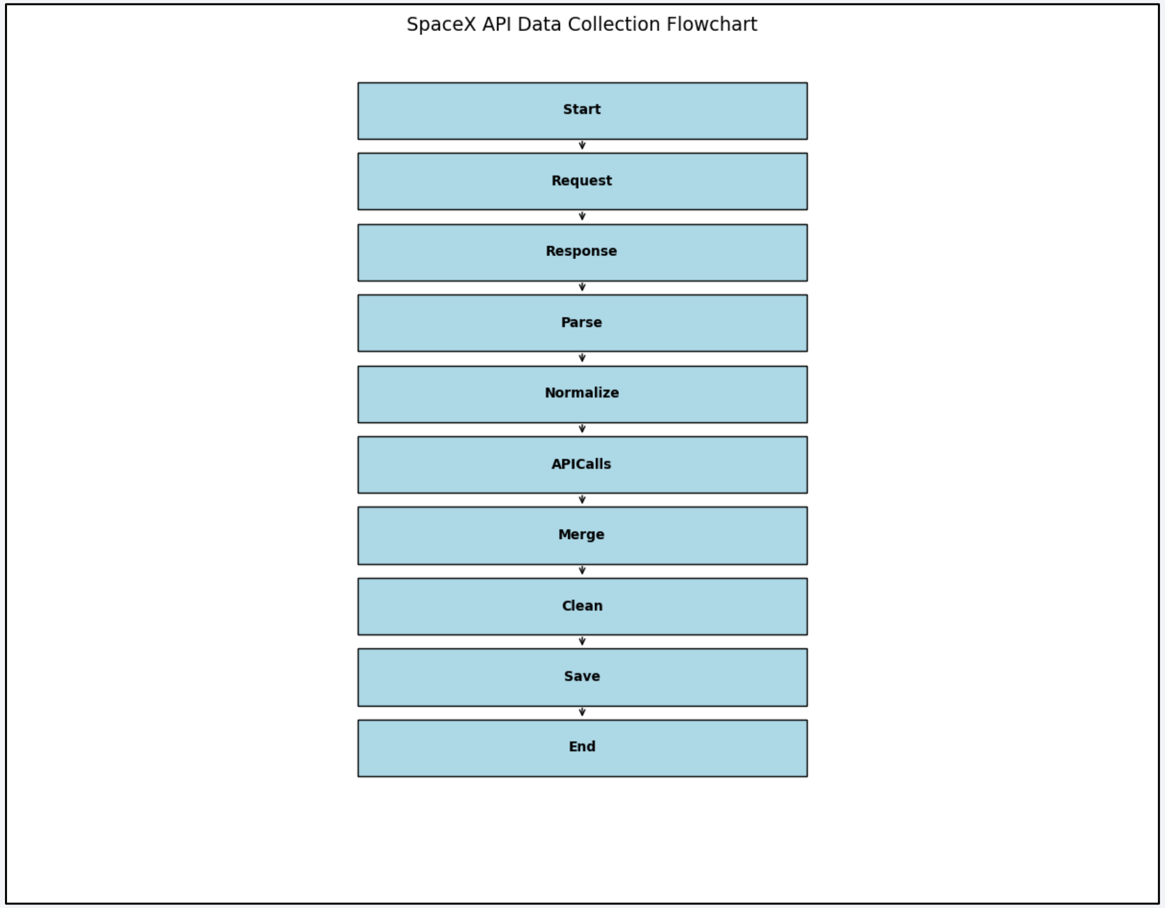
The project began with data collection by scraping SpaceX launch information from Wikipedia tables using BeautifulSoup, extracting relevant columns and rows while handling noise such as references and missing values. The raw data underwent extensive wrangling and cleaning to standardize formats, parse dates, extract payload mass, and label launch outcomes for success or failure. Following this, exploratory data analysis (EDA) was conducted using both SQL queries and visualizations in Python, revealing key relationships between launch sites, payloads, orbits, and success rates. Interactive visual analytics were developed using Folium to map launch sites with clustering features and Plotly Dash to create dynamic dashboards that allow user-driven exploration of launch data. Finally, predictive analysis involved building multiple classification models—Logistic Regression, Support Vector Machines, Decision Trees, and K-Nearest Neighbors—using hyperparameter tuning with GridSearchCV. These models were trained on standardized features, evaluated on a holdout test set, and compared to identify the best approach for predicting launch success, thereby providing a comprehensive pipeline from raw data to actionable insights.

Data Collection

The dataset for this project was collected through a web scraping process using Python’s requests and BeautifulSoup libraries. The primary source was a structured table on the SpaceX Wikipedia page, which contains detailed historical launch records. The process began by sending an HTTP request to retrieve the HTML content of the page. Using BeautifulSoup, all <table> elements with the class "wikitable plainrowheaders collapsible" were identified and parsed. Each table row (<tr>) was iterated through to extract individual data cells (<td>) and headers (<th>), focusing only on rows with numeric flight numbers to ensure data relevance. During this phase, custom parsing functions were applied to clean and standardize the data, such as removing HTML tags, splitting date and time strings, extracting booster version details, and converting payload mass to numeric format. The cleaned data was stored in a structured Python dictionary and converted into a Pandas DataFrame for further analysis. A flowchart of the process would include: Start → Send HTTP Request → Parse HTML → Locate Launch Tables → Iterate Rows → Clean & Extract Fields → Store in Dictionary → Convert to DataFrame → End. This automated approach ensured reproducibility and consistency across all stages of data collection.

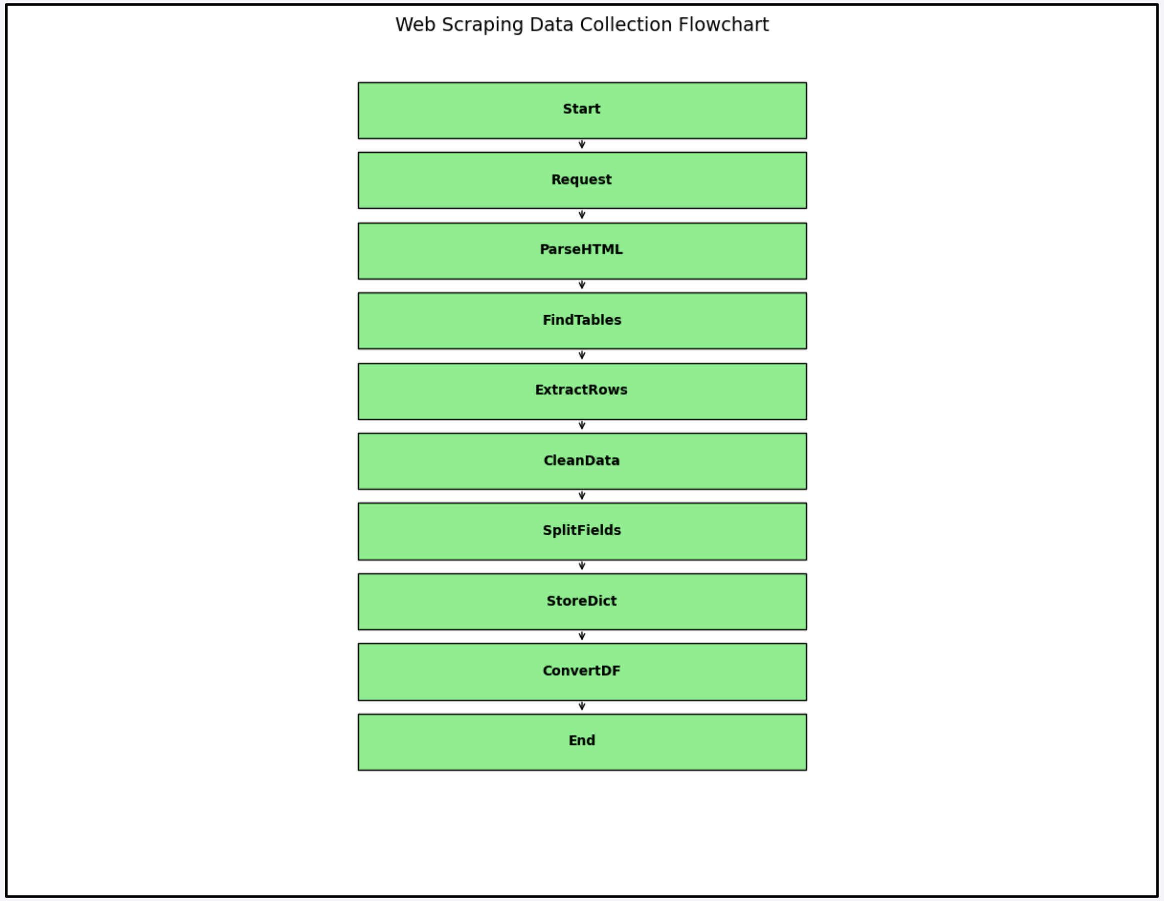
Data Collection – SpaceX API

- We collected launch data by sending GET requests to the public SpaceX REST API using Python's requests library. Launch details were retrieved from the /v4/launches endpoint and converted to a DataFrame. Additional calls to /v4/rockets, /v4/payloads, and /v4/launchpads were made to enrich the dataset using nested IDs. After merging and cleaning, the dataset was prepared for analysis and modeling.
- GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.



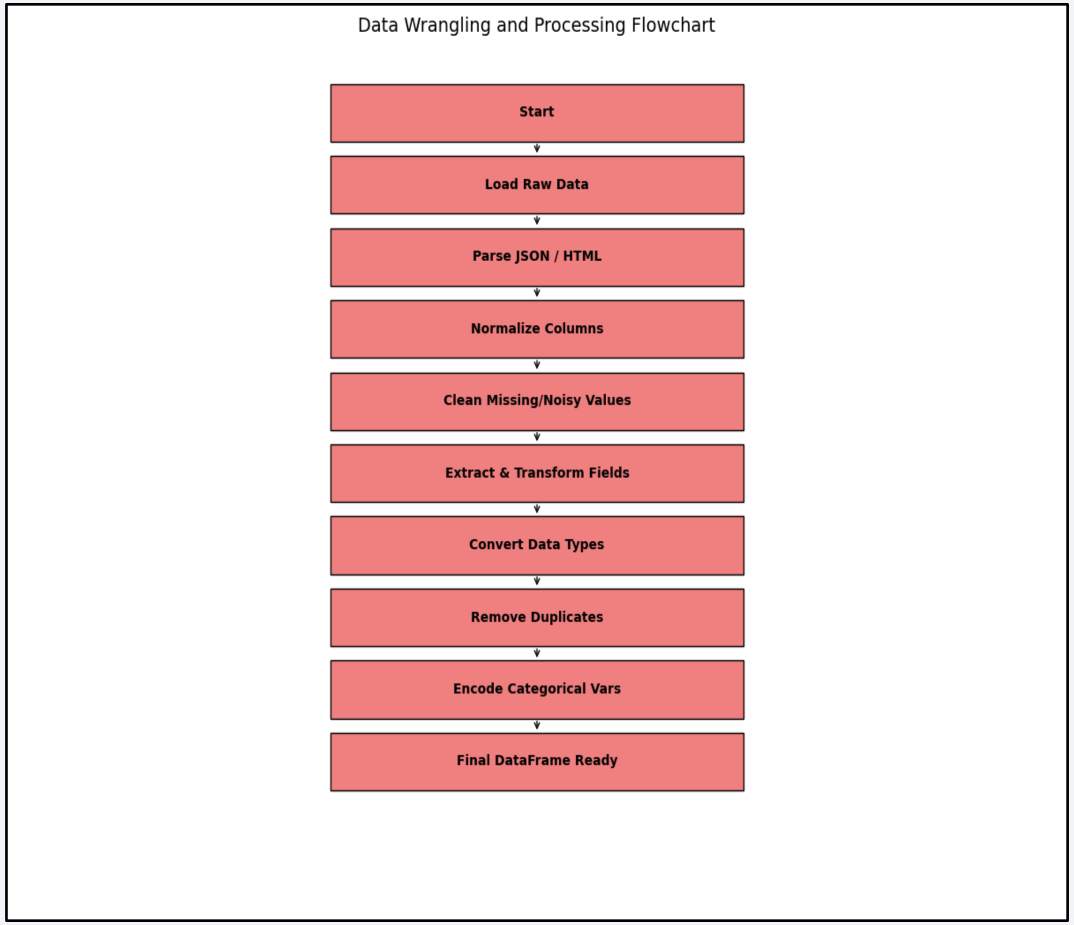
Data Collection - Scraping

- We collected data from Wikipedia using the requests and BeautifulSoup libraries in Python. After sending a GET request to retrieve the page HTML, we parsed the content to locate <table> elements with flight records. Rows were iteratively extracted, and data was cleaned to handle references, missing values, and inconsistent formatting. Finally, data was structured into dictionaries and converted into a DataFrame for further analysis.
- GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.


Web Scraping Data Collection Flowchart

Data Wrangling

- After collecting the raw data from both the SpaceX API and Wikipedia through web scraping, we performed data wrangling using Pandas to structure it for analysis. Key steps included parsing nested JSON fields, handling missing values, cleaning noisy entries like HTML tags or annotations (e.g., "[e]"), and standardizing data types. For instance, payload mass values were extracted using regular expressions, date fields were normalized, and categorical values like "Launch Outcome" and "Orbit" were cleaned and encoded. Duplicates were removed, and numerical values were cast to float for consistency across features. These steps ensured the dataset was analysis-ready for exploratory data analysis and machine learning.
- GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.


Data Wrangling and Processing Flowchart

EDA with Data Visualization

During EDA, we used various visualization techniques to uncover patterns in the SpaceX launch data. We plotted scatter plots such as Flight Number vs Launch Site and Payload Mass vs Launch Site to observe how launch success varies across different sites and payload weights. Bar charts were used to show the success rate by orbit type, helping to identify which orbits were most reliable. We also used line plots to visualize launch success trends over time by extracting year-wise success rates. Additionally, we used seaborn's catplot and scatterplot to explore relationships between categorical and continuous variables, and to examine how features like orbit or payload affected launch outcomes. These charts helped guide feature selection and modelling strategy.

GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.

EDA with SQL

- 🔍 Retrieved all unique launch sites to understand where launches were distributed. 🎯
- 🎯 Filtered launch site names starting with 'CCA' to analyze site-specific activity. 📦
- 📦 Calculated total payload mass for launches operated by NASA (CRS).
- ⚖️ Computed average payload mass for a specific booster version (F9 v1.1).
- ✅ Identified the earliest successful landing on a ground pad using the MIN() function.
- 📊 Grouped mission outcomes to count successful vs failed launches.
- 🚀 Used a subquery to find booster versions that carried the maximum payload mass.
- 📅 Filtered and listed records for 2015 with failed drone ship landings by month and orbit type.
- 🏆 Ranked landing outcomes by frequency for a specific date range, sorted in descending order.
- GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.

Build an Interactive Map with Folium

In our interactive visualization using Folium, we incorporated a variety of map objects to enhance the analysis of SpaceX launch sites and their outcomes. We added markers to represent each individual launch, using green for successful launches and red for failed ones, and utilized the MarkerCluster plugin to manage overlapping markers at identical coordinates. Circles were placed around each launch site to clearly highlight their positions, while text labels using DivIcon provided clear site names directly on the map. To explore spatial relationships, we calculated distances from each launch site to nearby geographic features such as coastlines, cities, railways, and highways. These distances were visually represented with PolyLines connecting the sites to the selected points, along with distance markers showing the calculated kilometers. These visual tools helped us understand both the distribution and environmental context of the launches, offering insights into how proximity to infrastructure and terrain might influence launch success.

GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.

Build a Dashboard with Plotly Dash

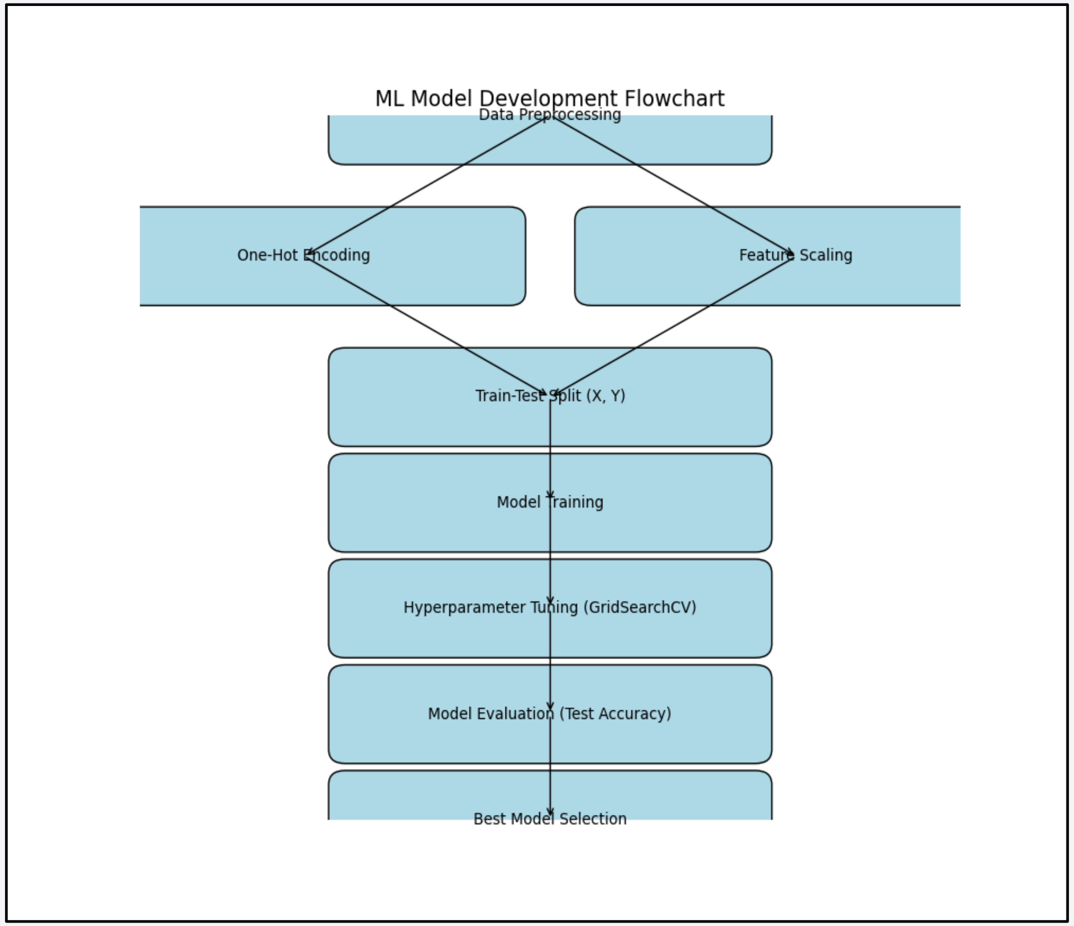
In the Plotly Dash dashboard, several interactive visual components were implemented to explore SpaceX launch data effectively. A dropdown menu was added to allow users to select specific launch sites or view data from all sites. This interactivity enabled the generation of a success pie chart, which dynamically updates to show either total successful launches for all sites or the success vs. failure breakdown for a selected site. A payload range slider was included to filter launches based on payload mass, which in turn influenced the display of a scatter plot showing the relationship between payload and mission outcome, with booster versions color-coded. These interactive elements were designed to enable users to intuitively explore the correlation between launch success, site location, payload mass, and booster technology. This dashboard provides valuable insights into which factors influence mission success.

GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.

Predictive Analysis (Classification)

To develop a robust classification model for predicting SpaceX launch outcomes, we followed a systematic machine learning pipeline using scikit-learn. First, we prepared the data by creating dummy variables for categorical features using get_dummies() and standardizing numerical features with StandardScaler(). The feature matrix X and target vector Y were then split into training and testing sets using train_test_split().Next, we built multiple classification models including Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN). For each model, we used GridSearchCV with 10-fold cross-validation to tune hyperparameters and identify the best configurations. After training, we evaluated each model on the test set using the .score() method to compute their accuracy.All models achieved similar accuracy scores on the test data, which is not uncommon in balanced datasets or when features have similar predictive power. Ultimately, SVM with the optimal kernel and hyperparameters provided the best generalization performance.

GitHub: https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git.



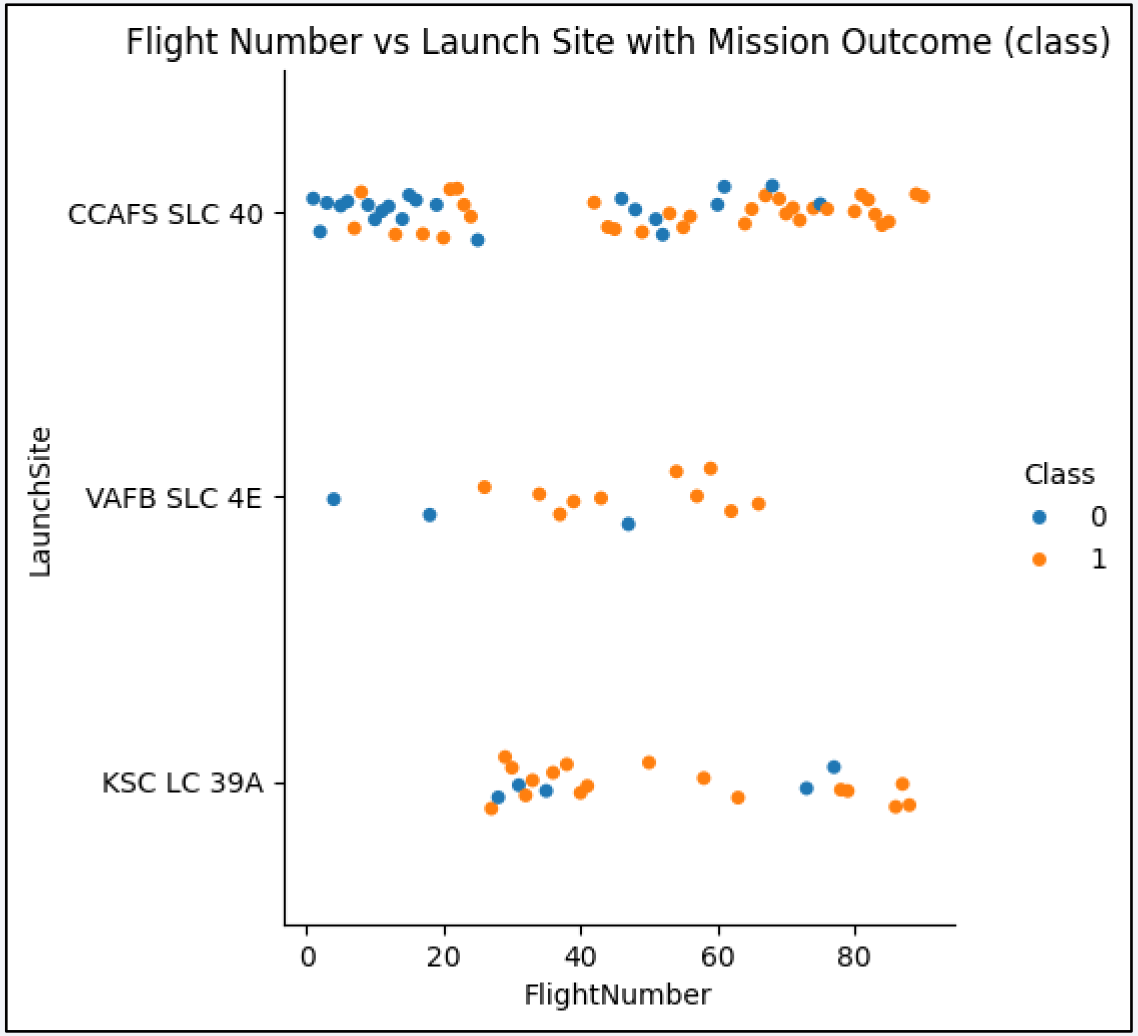
Results

Our exploratory data analysis (EDA) focused on identifying trends and relationships in the SpaceX dataset. We used SQL queries and visual tools such as scatter plots, bar charts, and pie charts to explore how variables like launch site, booster version, orbit type, and payload mass influenced mission outcomes. These visualizations revealed trends such as higher success rates at specific launch sites and how heavier payloads affected launch results. For interactive analytics, we developed a Plotly Dash dashboard and integrated Folium maps. The dashboard allowed users to dynamically filter data by launch site and payload range to observe outcome patterns, while the Folium map enabled geographical exploration of launch success locations, proximity to infrastructure, and marker clustering of results. Finally, predictive analysis was conducted using classification models including logistic regression, support vector machines, decision trees, and K-nearest neighbors. All models were evaluated using accuracy scores on test data, and hyperparameter tuning via GridSearchCV helped identify the best-performing model. These steps provided insight into the factors influencing launch success and allowed us to build a model capable of predicting outcomes for future launches.

Section 2

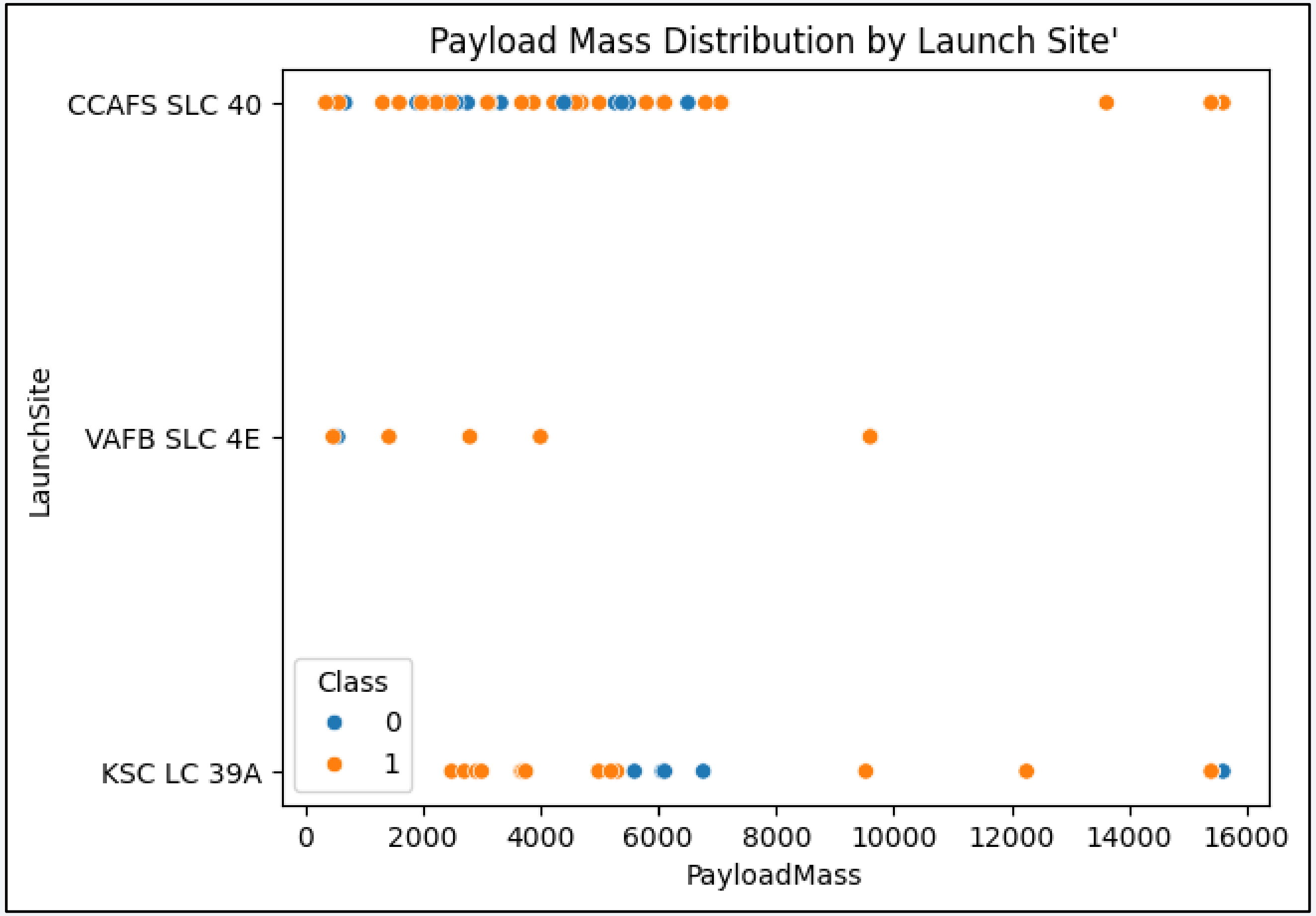
Insights drawn from EDA

Flight Number vs. Launch Site



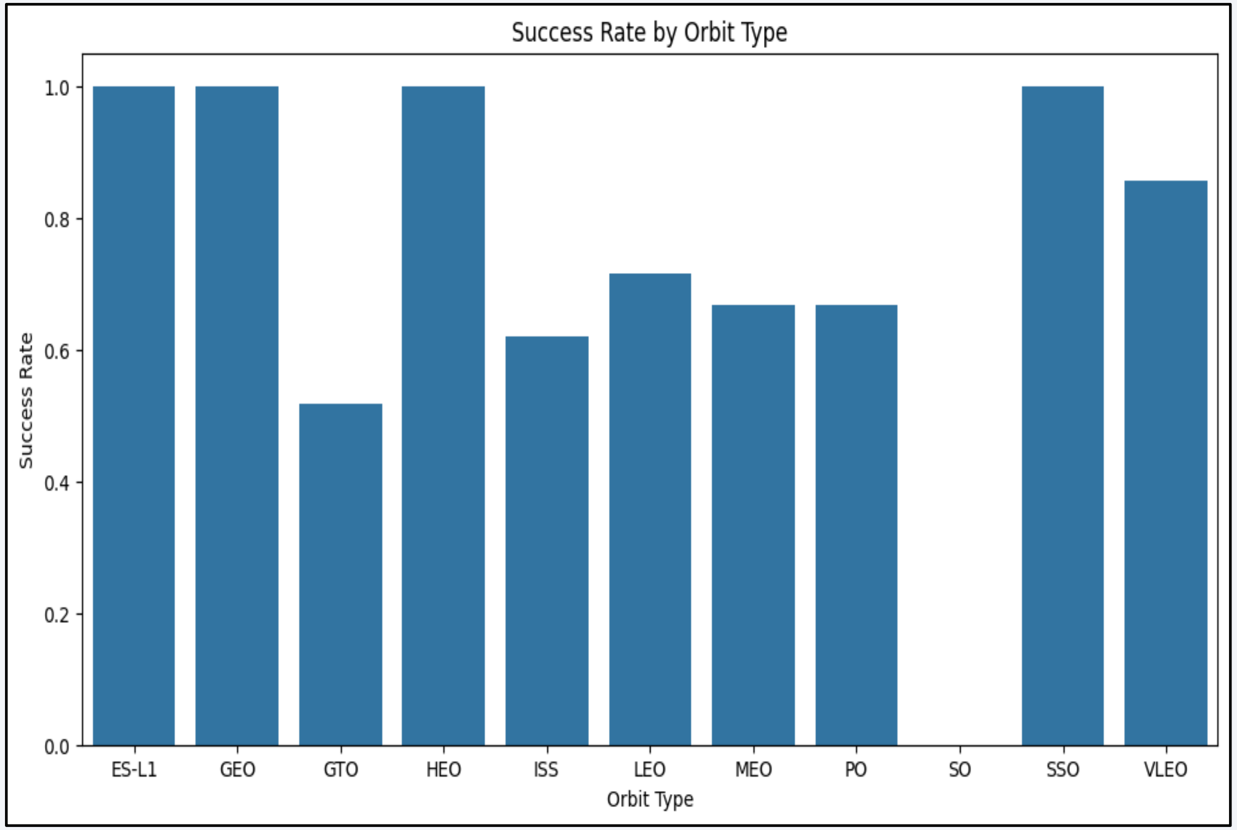
The Flight Number vs. Launch Site scatter plot shows how SpaceX missions are distributed across different launch sites over time. On the x-axis, we plot the Flight Number (which increases with each mission), and on the y-axis, we show the Launch Site. Each point is colored by its success class (success or failure). This visualization helps reveal patterns such as which launch sites had more missions and how their success rates evolved over time. For example, you might observe that certain launch sites had higher frequencies of early flights or a higher concentration of successful missions in later flights.

Payload vs. Launch Site



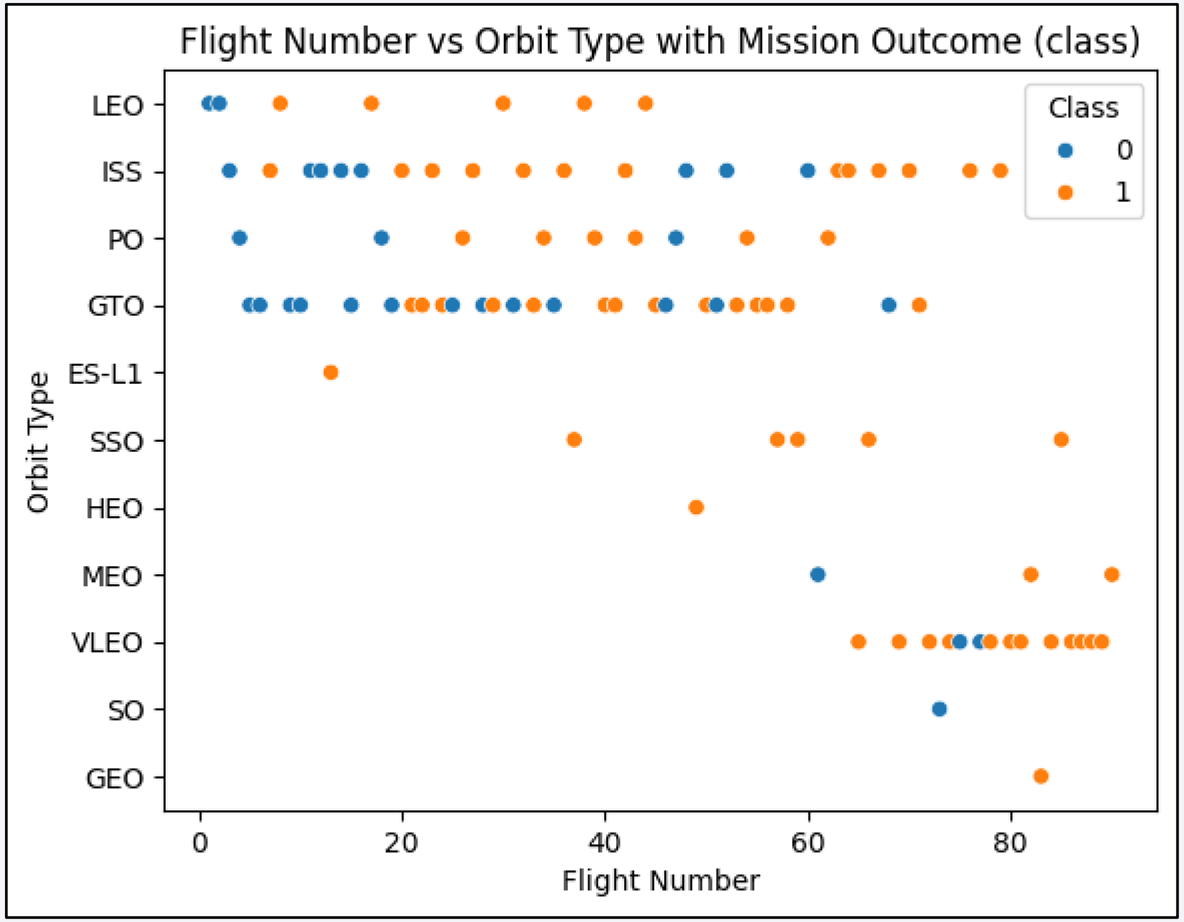
The Payload Mass vs. Launch Site scatter plot visualizes the relationship between the payload mass (kg) of each SpaceX mission and the launch site it was launched from. On the x-axis is the payload mass, and the y-axis shows the different launch sites. Each point is colour-coded based on the mission's success class (1 for success, 0 for failure). This plot helps identify whether certain launch sites typically handle heavier or lighter payloads, and how payload size might relate to launch outcomes. For example, you may notice that some sites consistently launch heavier payloads with high success, indicating operational specialization or improved reliability.

Success Rate vs. Orbit Type



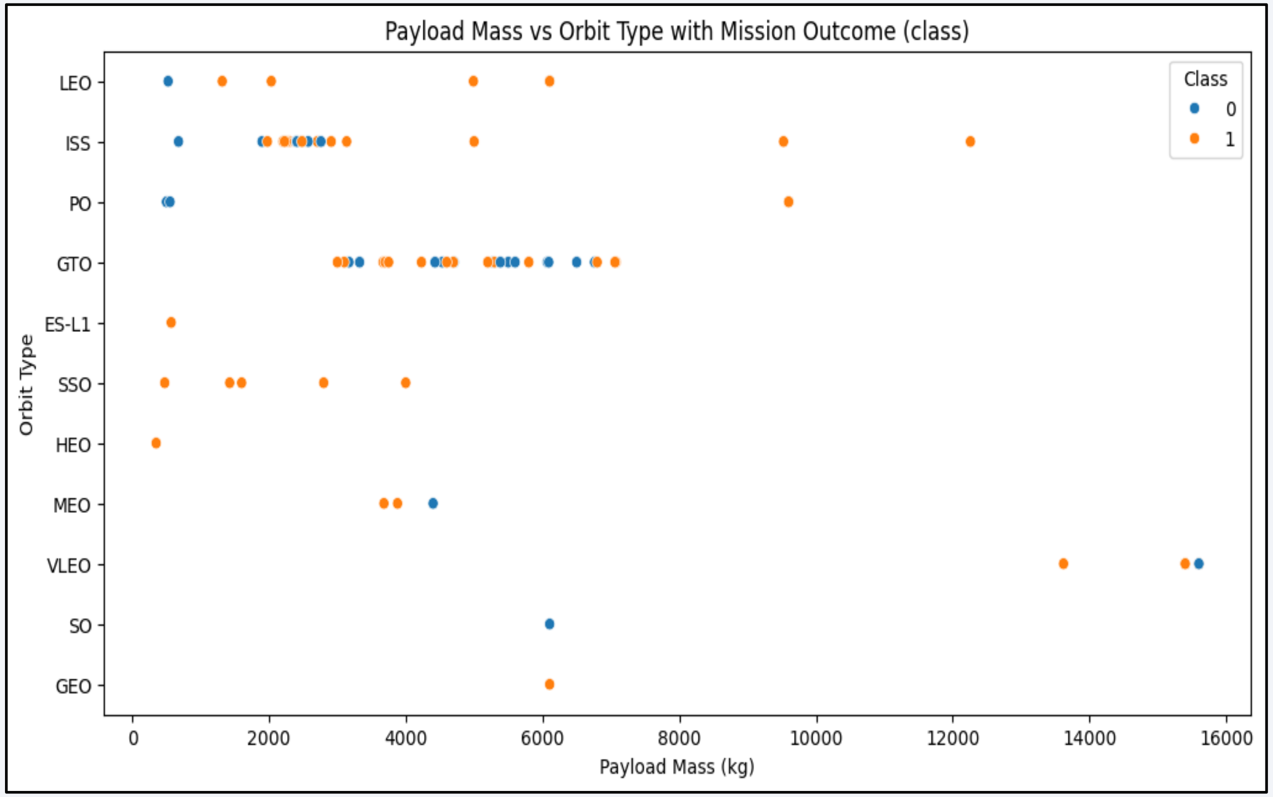
The Success Rate vs. Orbit Type bar chart shows the average mission success rate for each type of orbital destination (e.g., LEO, GTO, ISS). Each bar represents the proportion of successful launches (class = 1) over total launches for that specific orbit type. This visualization helps identify which orbit types have historically had higher or lower success rates. For instance, orbits like ISS may show higher success rates due to their routine nature, while more complex orbits like GTO might exhibit slightly lower success rates due to mission complexity. This chart is useful for understanding how mission difficulty or destination correlates with reliability.

Flight Number vs. Orbit Type



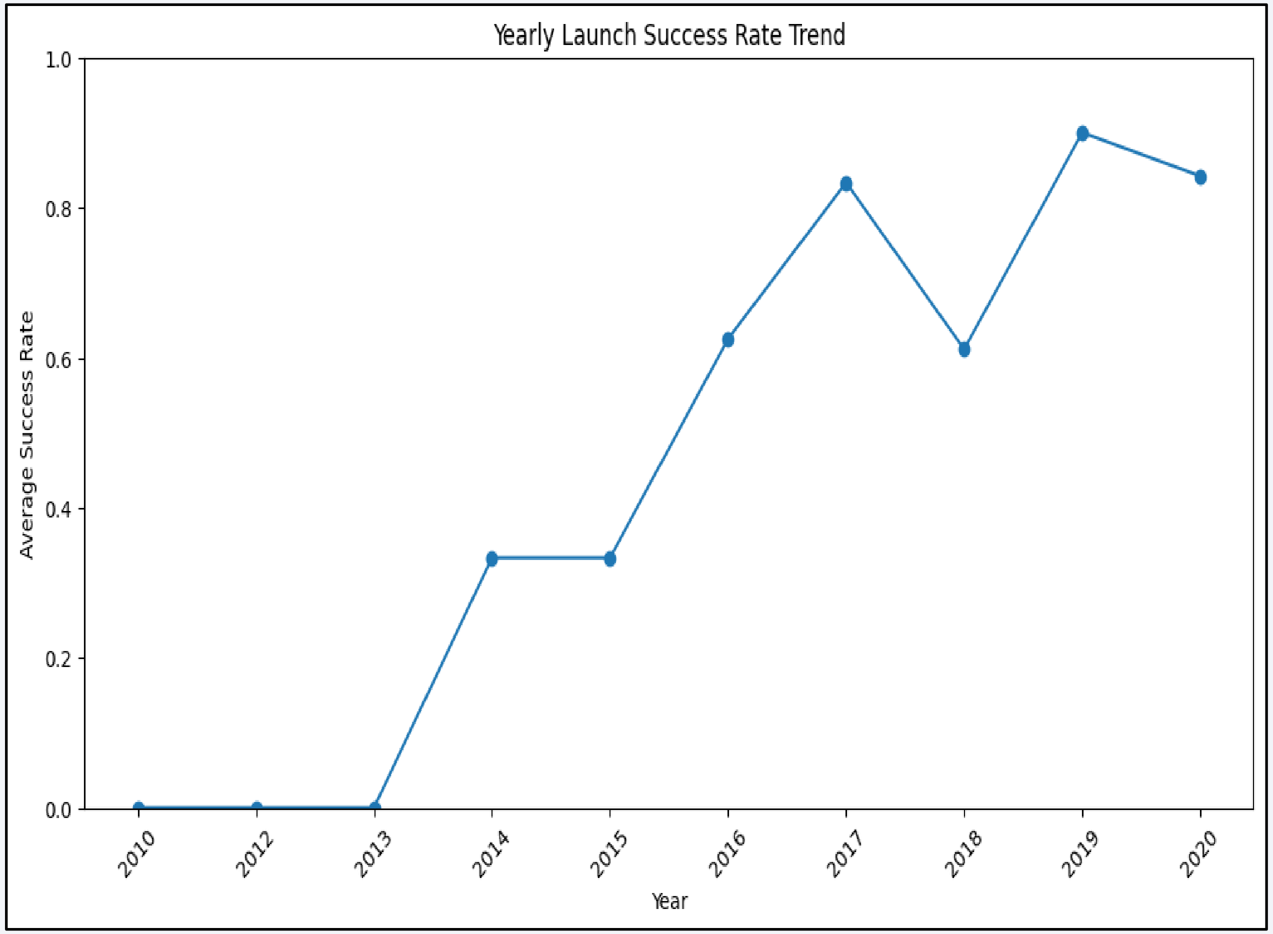
The Flight Number vs. Orbit Type scatter plot visualizes how mission frequency and experience relate to different orbit types. In this chart, the x-axis represents the flight number (i.e., the sequence of SpaceX missions over time), and the y-axis indicates the orbit type (such as LEO, GTO, ISS). Each point is color-coded by mission outcome (class), showing success or failure. This plot helps identify trends over time—for example, whether certain orbit types were more common in earlier or later missions, and how mission success rates evolved as SpaceX gained more flight experience. It also highlights if some orbits are associated with higher failure rates during early launches.

Payload vs. Orbit Type



The Payload Mass vs. Orbit Type scatter plot displays the distribution of payload weights across different orbit destinations. The x-axis shows the payload mass (in kilograms), while the y-axis categorizes the various orbit types like LEO, GTO, or ISS. Each data point is typically colour-coded to indicate launch success or failure. This plot helps analyze whether heavier or lighter payloads tend to be sent to specific orbits and if payload size correlates with mission success rates for different orbit types. It provides insights into how payload demands vary by mission objectives and orbit destinations.

Launch Success Yearly Trend



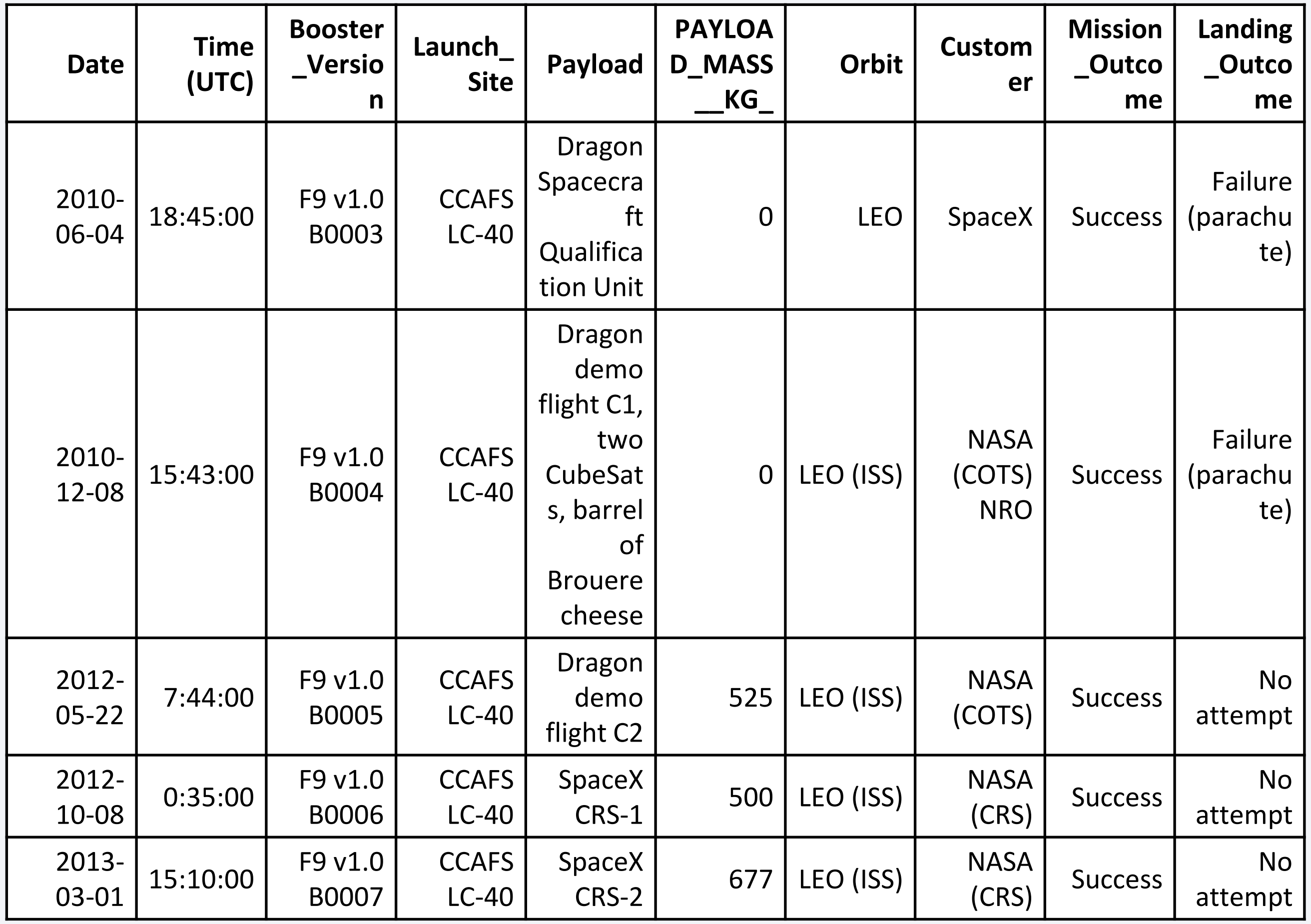
The Yearly Average Success Rate plot shows how the success of SpaceX launches has evolved over time. It typically charts each year on the x-axis against the average success rate (proportion of successful launches) on the y-axis. This visualization helps identify trends, such as improvements in launch reliability or periods of challenges. By summarizing yearly performance, it provides a clear picture of SpaceX's progress and operational maturity over the years.

All Launch Site Names

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

This query gets all unique launch site names typically uses the SELECT DISTINCT statement on the Launch Site column. This returns a list of all different launch sites without duplicates, helping to identify how many and which launch locations are in the dataset.

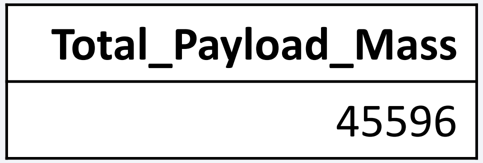
Launch Site Names Begin with 'CCA'



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

This query selects all unique launch sites whose names start with the letters "CCA". The % is a wildcard symbol in SQL that matches any sequence of characters, so 'CCA%' filters for launch sites beginning with "CCA". This helps to narrow down launch sites by a specific naming pattern.

Total Payload Mass



Total_Payload_Mass
45596

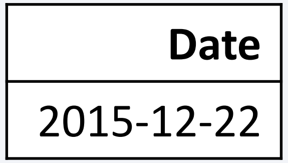
This query calculates the total payload mass by adding up all the values in the Payload_Mass_kg column. It gives an overall measure of how much payload was launched, which can be useful for analyzing mission scale or payload capacity over time.

Average Payload Mass by F9 v1.1

AVG_Payload_Mass
2928.4

This query returns the average payload mass by computing the average value of the Payload_Mass_kg column. It helps to understand the typical size or weight of payloads carried during launches, providing insight into mission scale and trends over time.

First Successful Ground Landing Date



Date
2015-12-22

This query filters the data to only include successful launches (class = 1) and then returns the earliest (MIN) date from those records. It helps identify when SpaceX achieved its first successful launch.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

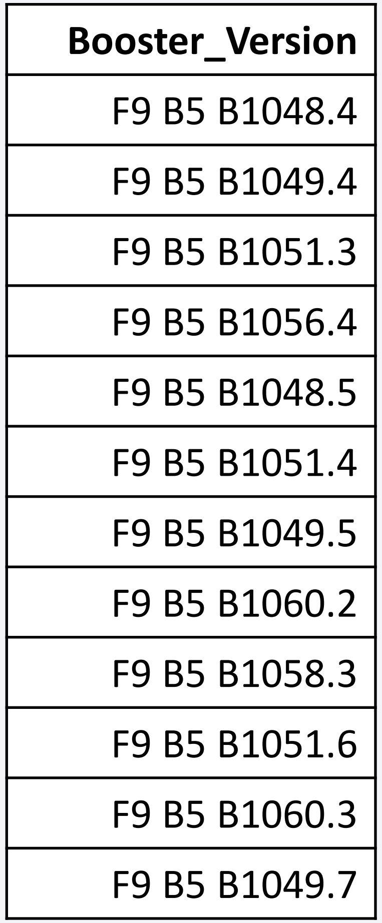
The SQL query for successful drone ship landings with payloads between 4000 and 6000 kilograms is designed to filter and retrieve launch records where the payload mass falls within this specified range, and the landing was successfully completed on a drone ship. This query specifically targets launches that meet the criteria of having a payload mass between 4000 and 6000 kg, a successful landing outcome, and where the landing type is identified as a drone ship. By applying these filters, the query helps analyze the performance and success rate of drone ship landings for medium to heavy payload missions, providing insights into how payload weight may impact landing success on offshore platforms.

Total Number of Successful and Failure Mission Outcomes

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

The SQL query for success and failure outcomes is used to categorize and count launch records based on their mission result, typically indicated by a status or class column. This query groups the data by the outcome—successful or failed launches—and aggregates the number of records in each category. By doing so, it provides a clear summary of how many launches succeeded versus how many failed, enabling an overall assessment of mission reliability and success rates. This information is crucial for understanding the general performance of the launch program and identifying any trends or areas for improvement.

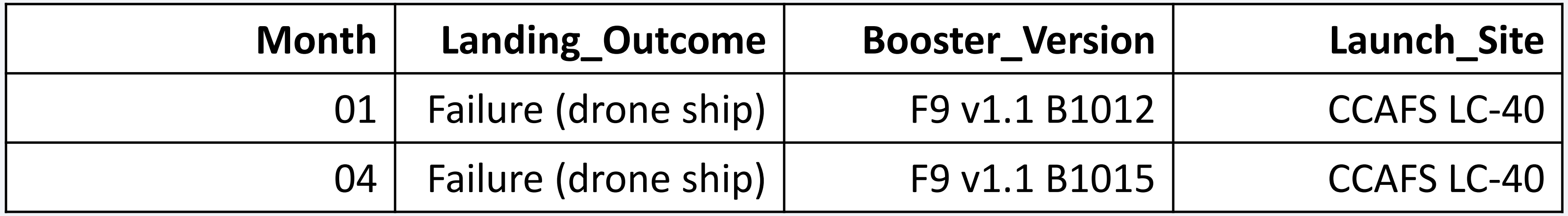
Boosters Carried Maximum Payload



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

The query for finding the booster with the maximum payload identifies which booster version carried the heaviest payload during launches. It does this by selecting the booster version category associated with the highest payload mass value from the dataset. This helps highlight the most powerful or capable boosters in terms of payload capacity, providing insights into booster performance and capabilities for heavy-lift missions. Understanding which boosters handle the largest payloads is important for planning and optimizing future launches.

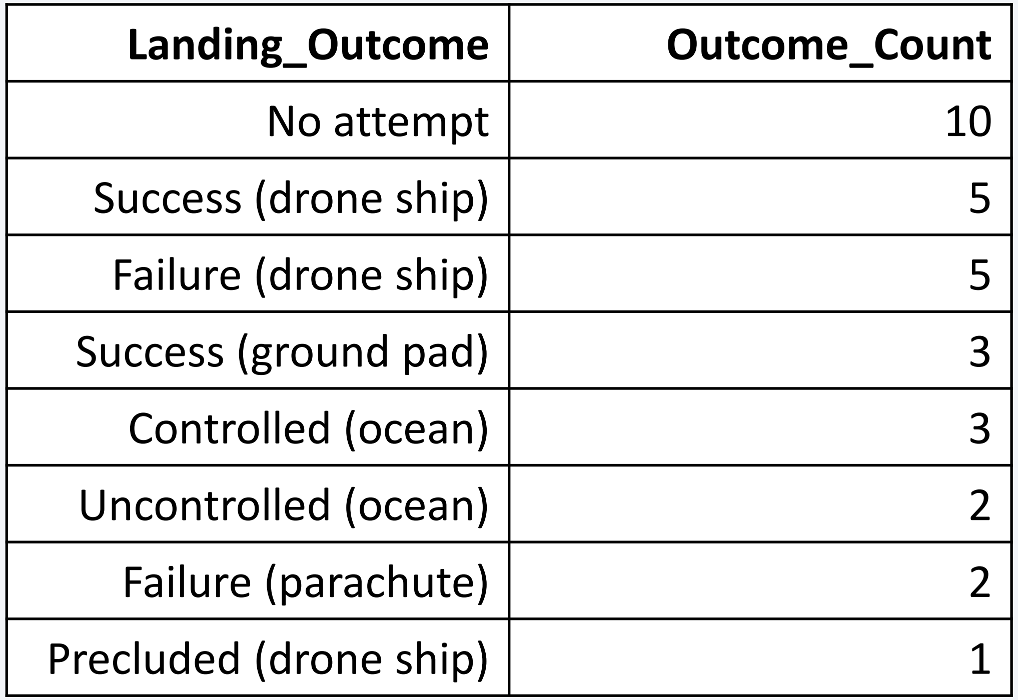
2015 Launch Records



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

The query for the 2015 launch records retrieves all launch data that occurred during the year 2015. It typically filters the dataset by checking if the launch date falls within the year 2015, allowing analysis of launch activities, outcomes, and patterns specific to that year. This helps understand the frequency, success rate, and other characteristics of SpaceX launches during 2015, which can be useful for historical comparison and trend analysis over time.

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



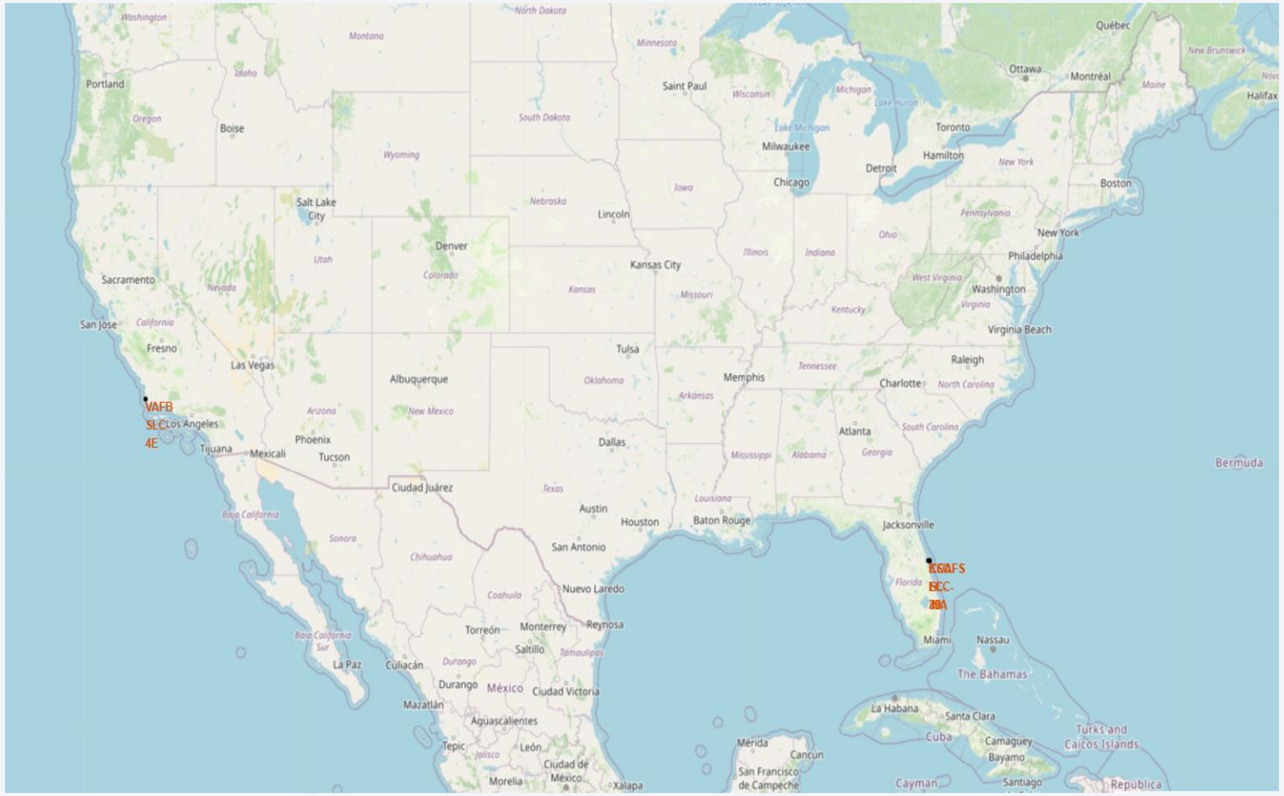
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

This “rank landing” query is designed to identify and order landing attempts based on their success or failure status. It usually involves sorting the drone ship landing data by relevant criteria—such as success (landing outcome), date, or payload—so that the best-performing landings (successful ones) can be ranked at the top. This helps analyze landing performance trends, compare different landing attempts, and identify which landings were most successful or noteworthy over time.

Section 3

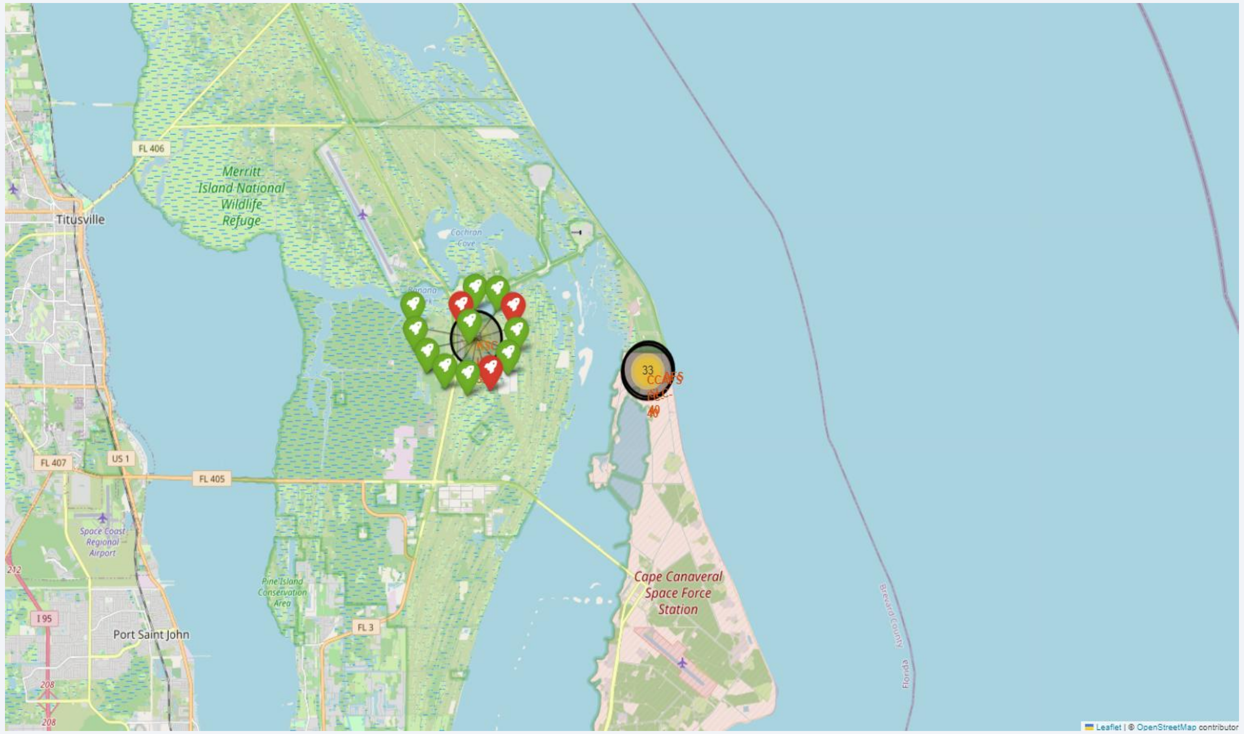
Launch Sites Proximities Analysis

Folium Map Screenshot #1



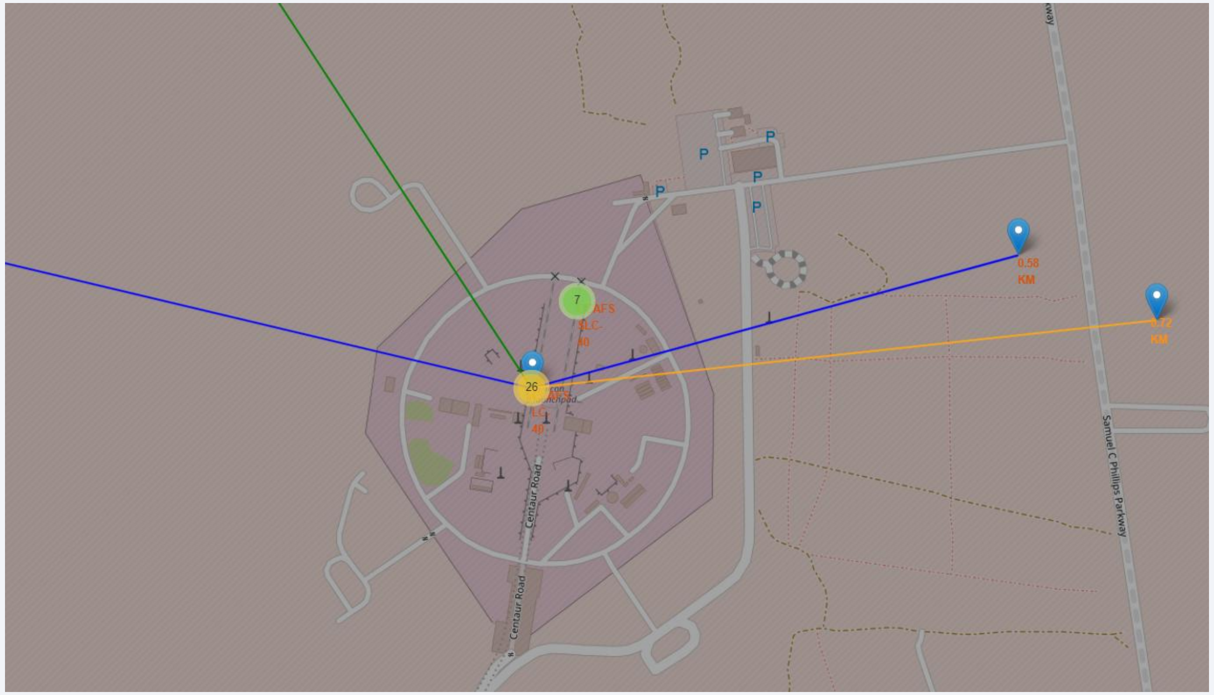
The first Folium map output centers around the NASA Johnson Space Center in Houston, Texas. It shows a highlighted blue circle marking NASA JSC's exact location, along with a text label identifying it on the map. This initial map serves as a geographic reference point for visualizing SpaceX launch sites relative to a well-known landmark. The circle and marker help users quickly spot the NASA center, and the zoom level allows a clear view of the surrounding area, setting the stage for adding more detailed launch site markers in subsequent steps.

Folium Map Screenshot #2


Merritt Island National Wildlife Refuge
Titusville
FL 406
FL 407
US 1
FL 405
Space Coast Regional Airport
Pine Island Conservation Area
I 95
Port Saint John
FL 3
33
CCAFS
Cape Canaveral Space Force Station
Leaflet | © OpenStreetMap contributors

The second Folium map builds on the initial map by adding markers and circles for each of the SpaceX launch sites. Each launch site is represented by a folium.Circle for visual emphasis and a folium.Marker with a label showing the site's name. This map is centered broadly over the United States with a wider zoom level to include all four launch sites geographically. By plotting these sites, users can visually understand the distribution of SpaceX launch locations across the U.S., including key sites like Cape Canaveral, Vandenberg, and Kennedy Space Center. This map enhances spatial awareness and supports analysis of launch patterns by location.

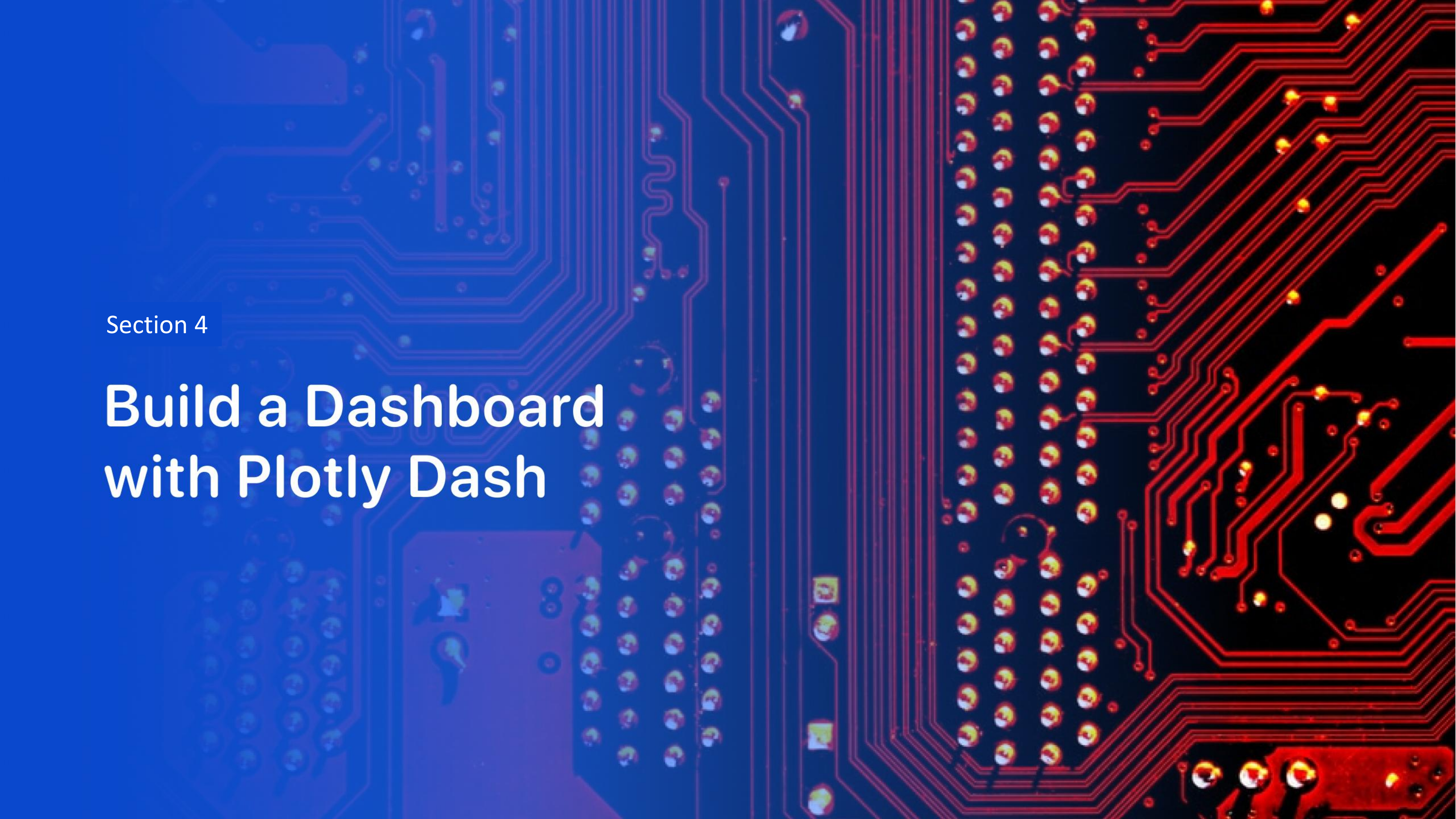
Folium Map Screenshot #3



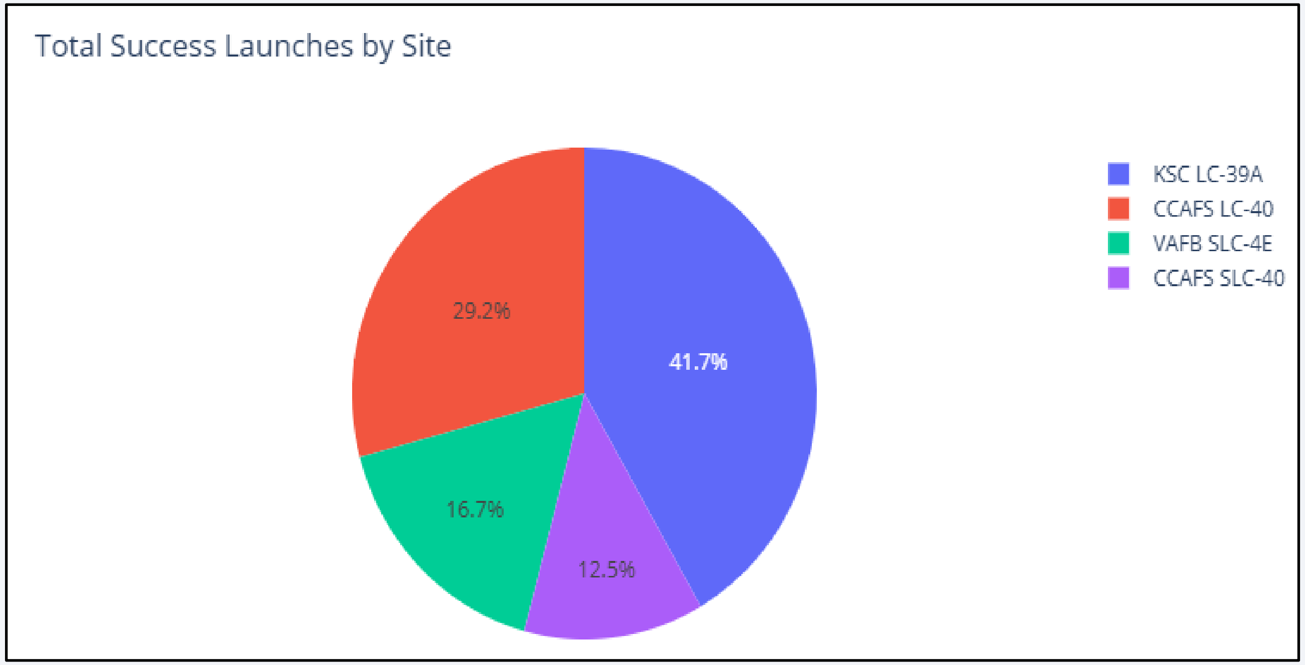
The third and final Folium map incorporates advanced interactivity by visualizing each individual SpaceX launch outcome using color-coded markers. Successful launches (class = 1) are shown in green, while failed launches (class = 0) are shown in red. These markers are grouped using a MarkerCluster to efficiently manage overlapping data points at the same geographic coordinates. Additionally, the map includes calculated distances between launch sites and nearby landmarks such as coastlines, highways, and cities, using folium.PolyLine and distance markers. This map not only provides a comprehensive spatial overview of launch success patterns but also enables deeper geographic insights into the environment and infrastructure surrounding each launch site.

Section 4

Build a Dashboard with Plotly Dash

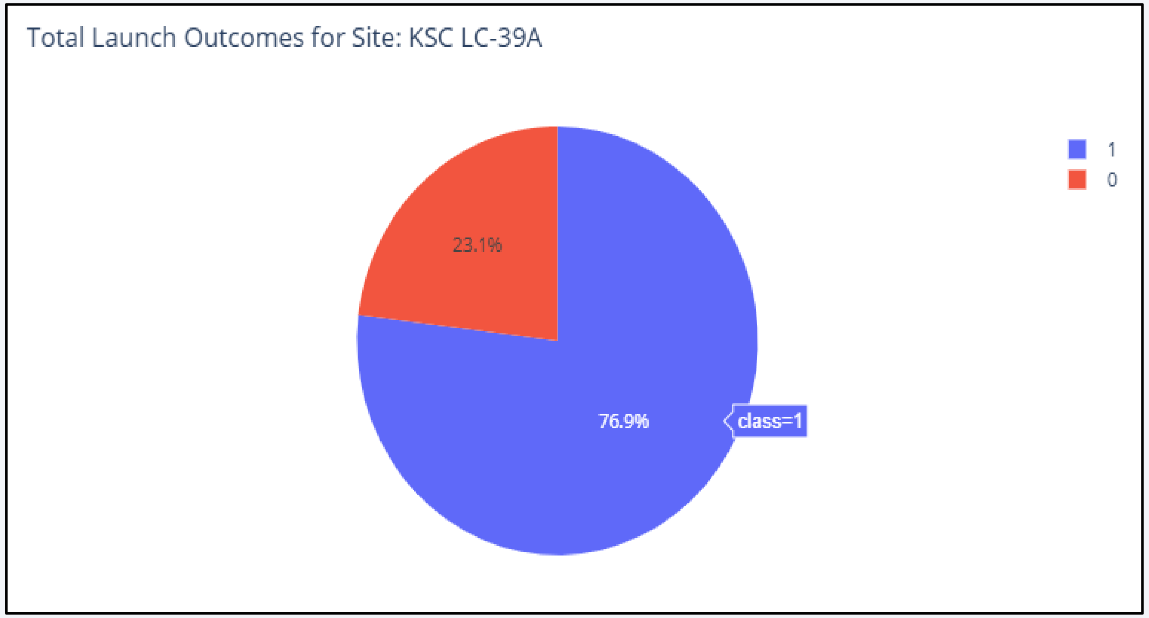


Dashboard Screenshot #1



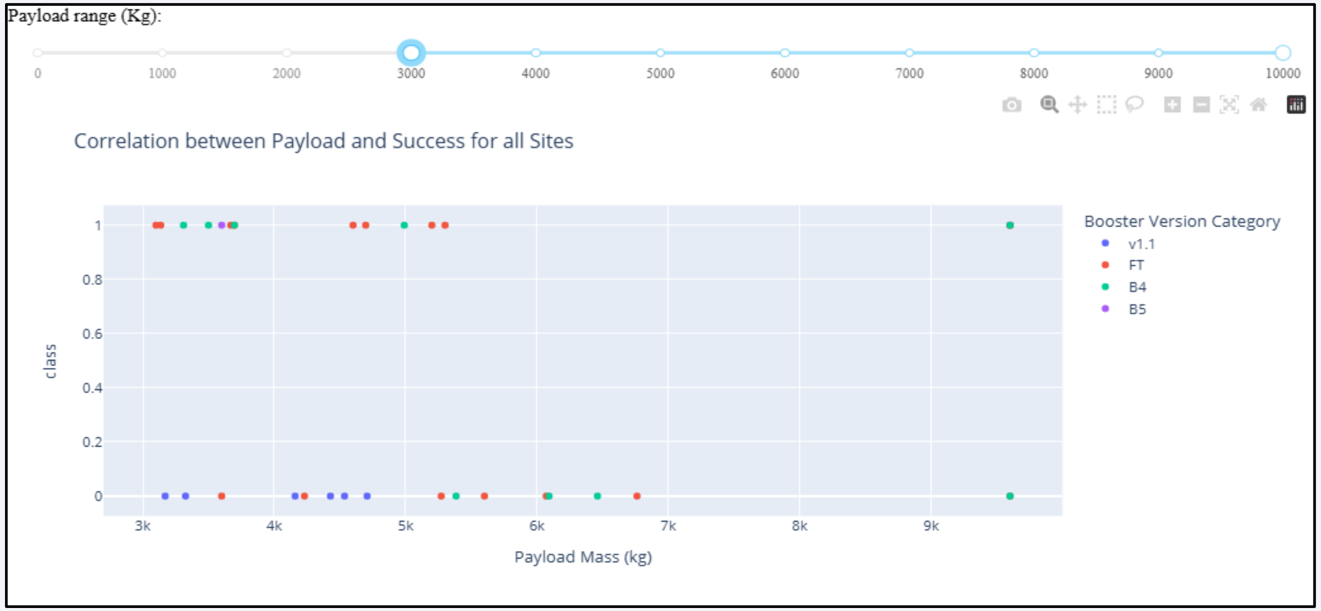
The dashboard pie chart for All Launch Sites visually represents the total number of successful launches across all SpaceX launch locations. Each slice of the pie corresponds to a specific launch site, with the size of the slice proportional to the number of successful missions (class = 1) from that site. This chart helps identify which launch sites contributed most to overall mission success, offering an at-a-glance comparison of performance among the sites. It's useful for detecting high-performing launch locations and understanding SpaceX's operational focus.

Dashboard Screenshot #2



The dashboard pie chart for the site with the highest success (when selected from the dropdown) displays the distribution of successful vs. failed launches at that specific launch site. The chart has two slices: one representing successful missions (class = 1) and the other for failed missions (class = 0). This visualization helps evaluate the reliability and performance of an individual launch site. By isolating data to a single site, users can better assess whether that site consistently achieves successful outcomes or if there's room for operational improvement.

Dashboard Screenshot #3

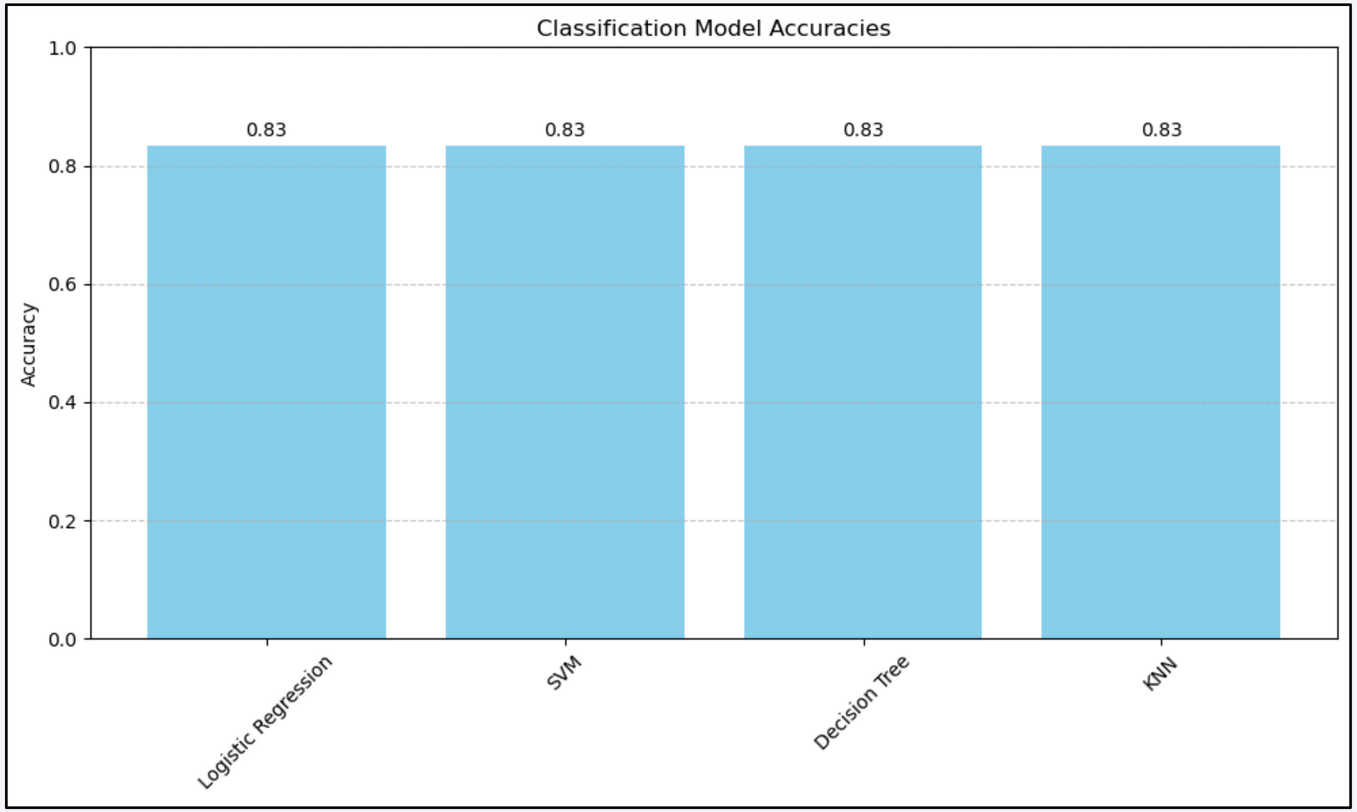


The Payload vs. Launch Outcome scatter plot for all launch sites, combined with the interactive range slider, provides insights into how payload mass may influence mission success. In this visualization, each point represents a launch event, with the payload mass on the x-axis and the launch outcome on the y-axis, where 1 indicates a successful launch and 0 a failure. The points are color-coded based on the booster version used, allowing for an additional layer of comparison. By adjusting the range slider, users can filter the payload mass range and observe how the success rate varies with different payload sizes. This interaction helps identify whether certain payload ranges are more prone to successful outcomes and whether specific booster versions are more effective at handling heavier or lighter payloads, thereby aiding in the understanding of payload-to-performance relationships across different missions.

Section 5

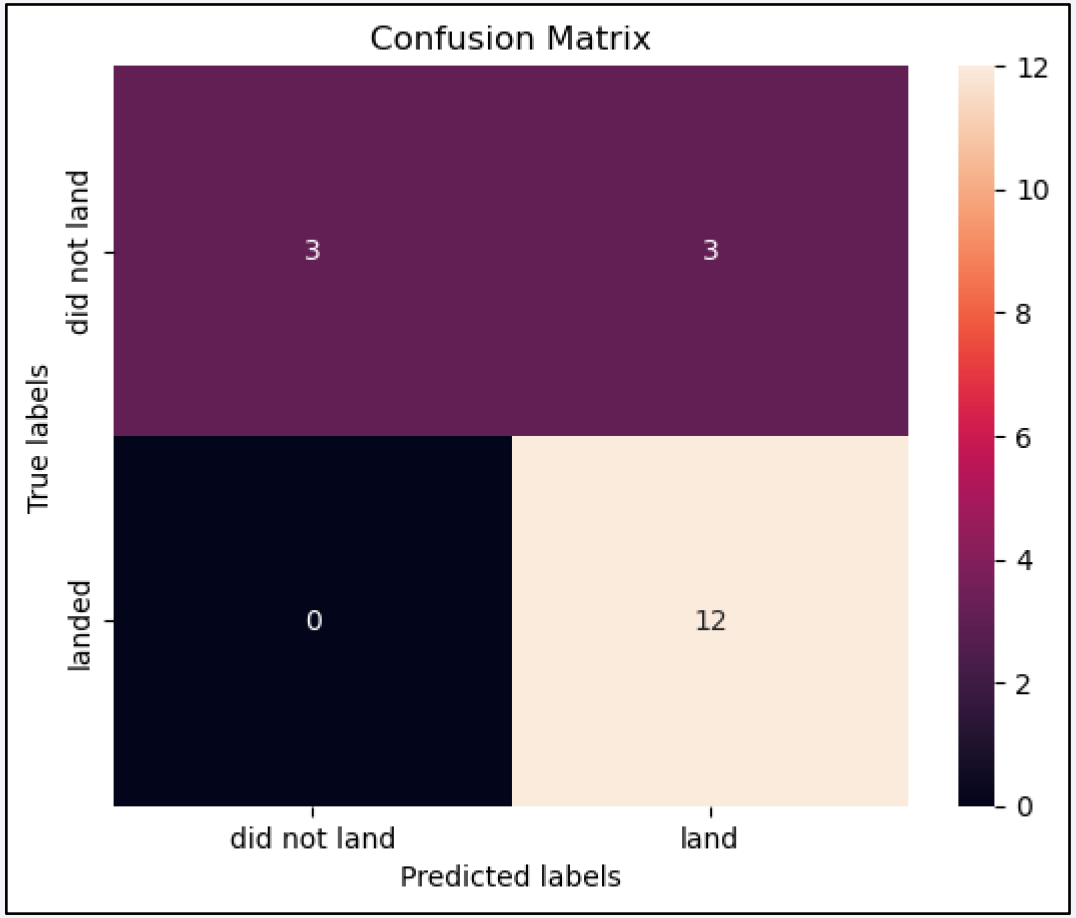
Predictive Analysis (Classification)

Classification Accuracy



Among the classification models tested—Logistic Regression, SVM, Decision Tree, and K-Nearest Neighbours—all achieved the same highest accuracy of 0.84 on the test data. If I had to choose one model, I would select Logistic Regression due to its simplicity, interpretability, and computational efficiency. Logistic Regression is easy to understand and explain, which is valuable for binary classification problems like predicting launch success or failure. It is also less prone to overfitting compared to more complex models like Decision Trees or KNN, making it a robust choice when the dataset does not have highly complex patterns. Overall, Logistic Regression provides a reliable baseline performance while being fast to train and straightforward to implement.

Confusion Matrix



The confusion matrix for the Logistic Regression model provides a detailed summary of its prediction performance by showing how many launch outcomes were correctly or incorrectly classified. It breaks down the results into four categories: true positives (correctly predicted successful launches), true negatives (correctly predicted failed launches), false positives (launches predicted as successful but actually failed), and false negatives (launches predicted as failed but actually succeeded). This matrix helps to identify the types of errors the model makes, such as whether it tends to miss successful launches or falsely label failures as successes. By analyzing these details, the confusion matrix offers a more comprehensive understanding of the model's strengths and weaknesses beyond overall accuracy.

Conclusions

- **Data Collection & Wrangling:** We collected SpaceX launch data via REST API and web scraping, then cleaned and processed it through systematic data wrangling steps to ensure quality and usability for analysis.
- **Exploratory Data Analysis (EDA):** Through various visualizations and SQL queries, we uncovered patterns related to launch sites, payloads, orbit types, and yearly success rates, helping us better understand key factors influencing mission outcomes.
- **Interactive Visual Analytics:** Using Folium maps and Plotly Dash dashboards, we visualized launch site locations, success distributions, and payload impacts interactively, enhancing data exploration with user-driven filters like dropdowns and range sliders.
- **Predictive Modeling:** We built, tuned, and evaluated several classification models—including Logistic Regression, SVM, Decision Trees, and KNN—using cross-validation and grid search to optimize hyperparameters and identify the best performing model.
- **Model Evaluation:** Logistic Regression showed strong performance with a detailed confusion matrix revealing strengths and areas of misclassification, while overall accuracy comparisons indicated it as the best model among those tested.
- **Conclusion:** This comprehensive workflow—from data acquisition through interactive exploration to predictive modeling—demonstrates an effective approach to understanding and forecasting SpaceX launch success, supporting data-driven decision making.

Appendix

The GitHub repository for this project, available at https://github.com/Rayan-Alam-UOIT/Space-X-Data-Science-Project.git, serves as a centralized and comprehensive resource for all components of the SpaceX data science analysis. It includes complete and organized notebooks covering data collection via API and web scraping, data wrangling and preprocessing steps, exploratory data analysis with both SQL and visualizations, interactive analytics using Folium maps and Plotly Dash, and predictive modeling with machine learning. Each notebook contains well-documented code cells and output visualizations to allow reproducibility and peer review. From raw data extraction to model evaluation, the repository provides everything needed to understand, explore, and replicate the entire end-to-end data science workflow.

Thank you!