IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<qilin zheng>
<07-29-2025>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Objective:

This project sought to analyze historical SpaceX Falcon 9 launch data to identify key factors influencing first-stage landing success and to build a machine learning model to predict these outcomes, ultimately aiming to reduce operational costs.

Methodology:

- An end-to-end data science **pipeline** was employed, beginning with data collection via the SpaceX API and web scraping. The data was then cleaned and explored using **SQL** queries and visual analytics with Matplotlib, Seaborn, and interactive maps with **Folium**. Finally, several classification models were trained and optimized using **GridSearchCV** to find the best predictor.

Key Results & Findings:

- Launch Site Impact: The KSC LC-39A launch site demonstrates the highest success rate (77.3%).
- Payload Influence: Payloads between 3000-4000 kg have the highest success rate (72.7%), while those in the 6000-9000 kg range show a significantly lower success rate.
- Booster Version: The Falcon 9 B5 booster version achieved a 100% success rate in the dataset, highlighting its reliability.
- Predictive Model: The best-performing classification model achieved a test accuracy of over 85%, providing a reliable tool for predicting launch outcomes.

Conclusion & Recommendation:

The analysis successfully identified launch site, payload mass, and booster version as critical determinants of landing success. The developed predictive model is recommended as a valuable decision-support tool for SpaceX to identify high-risk missions for further review.

Project Introduction: Predicting Falcon 9 Success

Project Background and Context :

- SpaceX has revolutionized the space industry by pioneering reusable rocket technology with its Falcon 9 rocket.
- The key to this revolution is the successful landing and recovery of the rocket's first stage, allowing it to be reused in future missions.
- While SpaceX advertises launches for ~$62 million, a fraction of its competitors' costs, this low cost is heavily dependent on first-stage reuse. A launch failure represents a significant financial loss and a setback to the launch schedule.

The Business Problem

- Each Falcon 9 launch carries a high financial stake. A failed first-stage landing negates the primary cost-saving advantage of reusability.
- Therefore, being able to predict the outcome of a first-stage landing before a launch is incredibly valuable. It allows SpaceX and its customers (like NASA) to assess risk, manage costs, and optimize launch parameters.

Project Objectives

- This project aims to answer a critical question: Can we predict if the Falcon 9 first stage will land successfully?
- To achieve this, we will analyze historical launch data to:
- Determine the key factors and variables that influence landing success.
- Build a robust machine learning classification model to predict the outcome (Success/Failure) of a landing attempt.

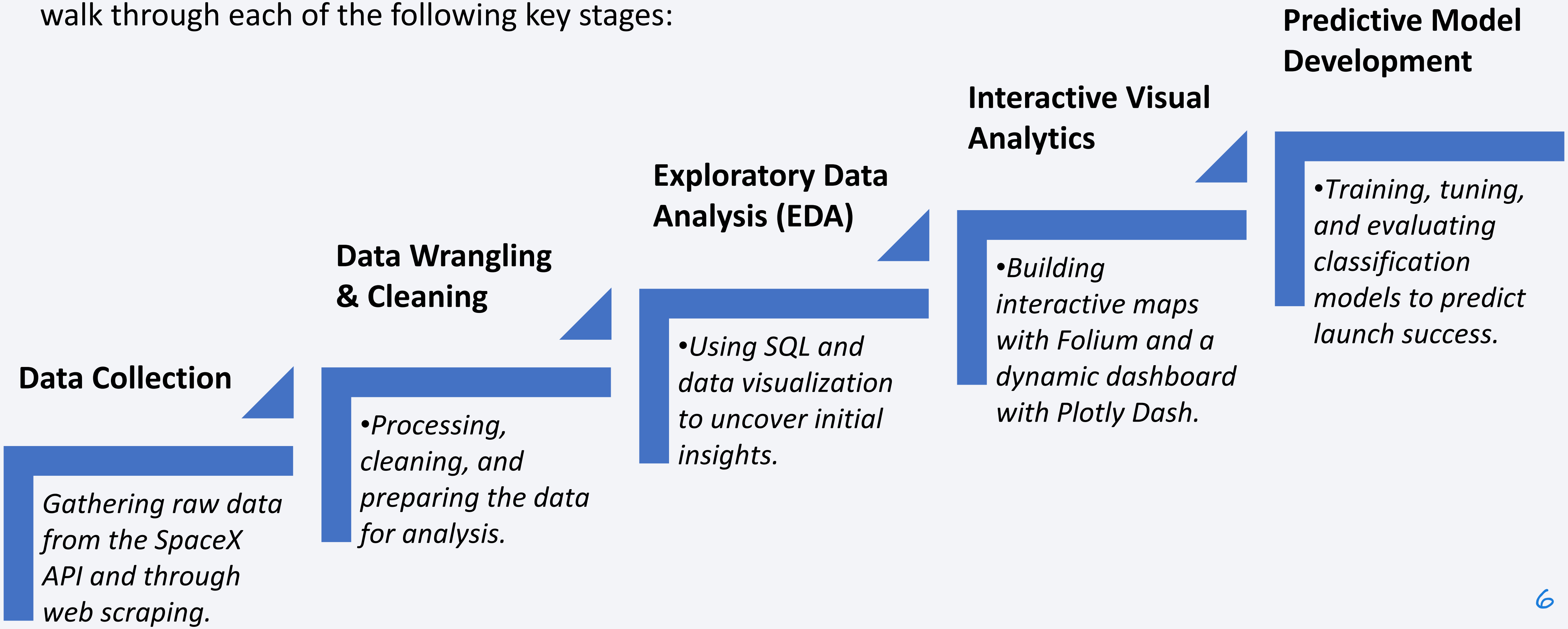
Section 1

Methodology

Project Methodology

Introduction:

- This section provides a detailed overview of the end-to-end data science process used in this project. We will walk through each of the following key stages:



Data Collection

Gathering raw data from the SpaceX API and through web scraping.

Data Wrangling & Cleaning

•Processing, cleaning, and preparing the data for analysis.

Exploratory Data Analysis (EDA)

•Using SQL and data visualization to uncover initial insights.

Interactive Visual Analytics

•Building interactive maps with Folium and a dynamic dashboard with Plotly Dash.

Predictive Model Development

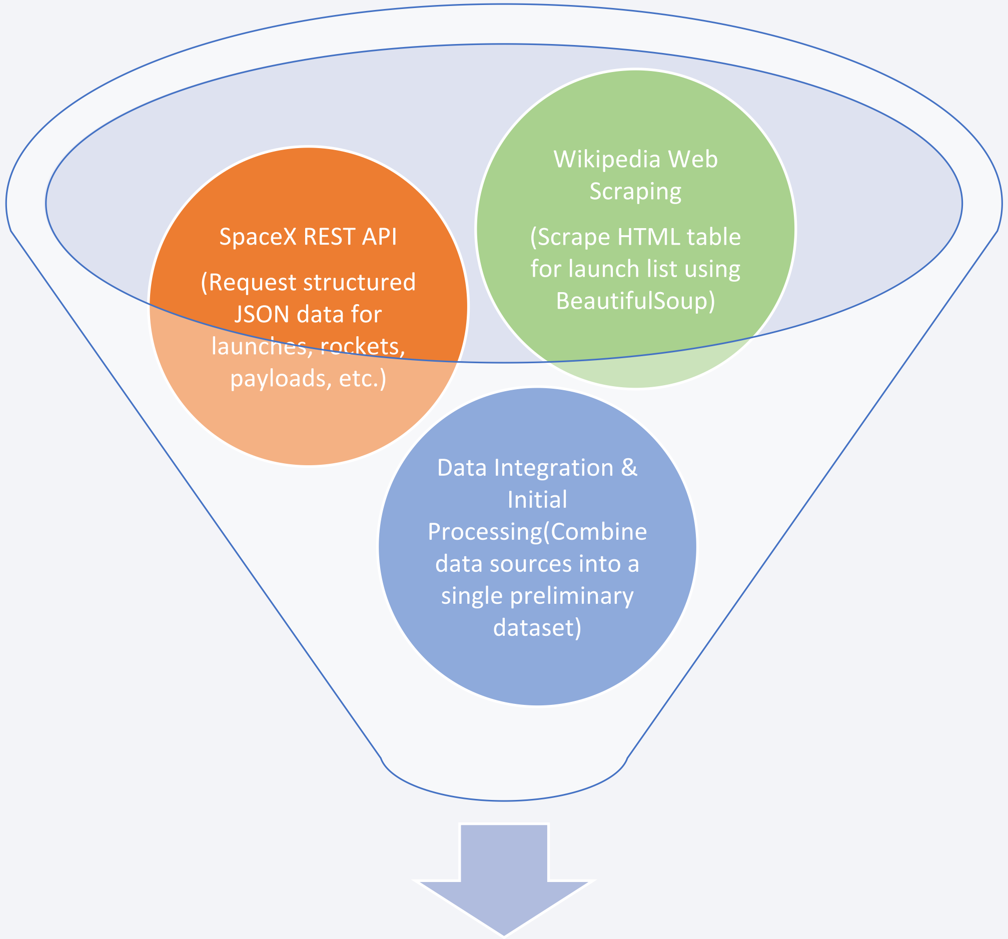
•Training, tuning, and evaluating classification models to predict launch success.

Data Collection: Overall Process

- **Objective:**

  The foundational step of this project was to gather comprehensive historical launch data. To ensure a robust and complete dataset, a dual-source approach was implemented:

  - **Primary Source:** The official SpaceX REST API for detailed, structured technical data.
  - **Secondary Source:** Web scraping of a Wikipedia page for supplementary launch records and to cross-reference information.


SpaceX REST API
(Request structured JSON data for launches, rockets, payloads, etc.)
Wikipedia Web Scraping
(Scrape HTML table for launch list using BeautifulSoup)
Data Integration & Initial Processing(Combine data sources into a single preliminary dataset)

Combined Raw Dataset

(Ready for Data Wrangling)

Data Collection – SpaceX API

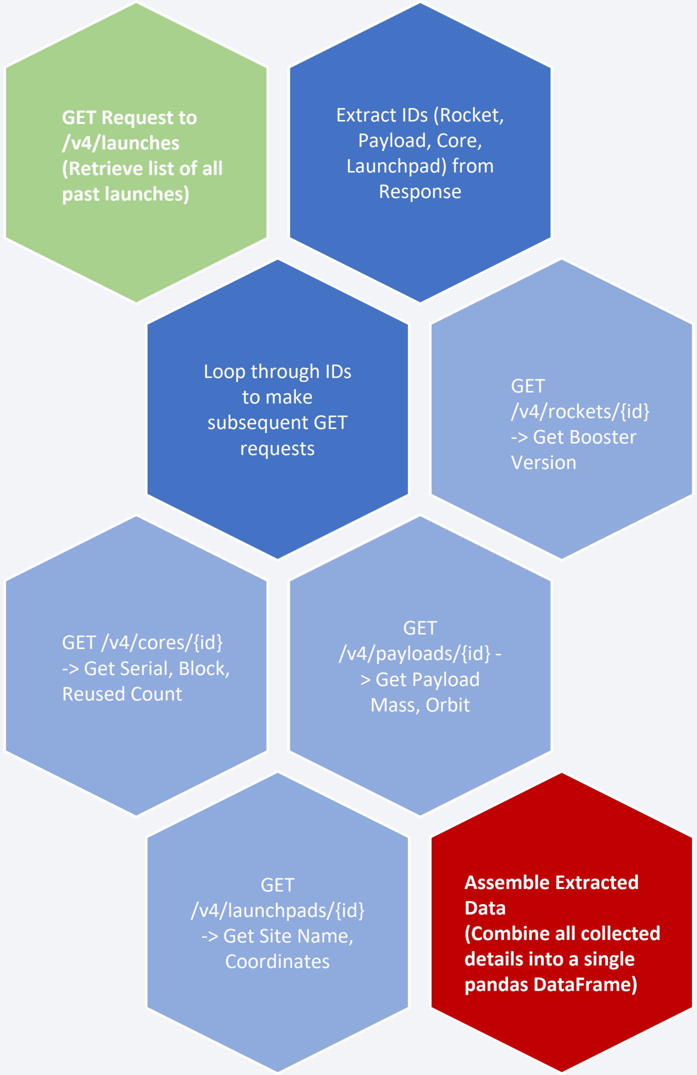
- **Process Overview:**
  The primary dataset was constructed by making a series of REST API calls to the SpaceX v4 endpoint.

- The process involved retrieving an initial list of all launches and then using IDs from that data to make subsequent calls for more detailed information about rockets, payloads, and cores.

GitHub Link:

For a detailed walkthrough, see the notebook:

- [1_Data_Collection&Wrangling.ipynb]



GET Request to /v4/launches (Retrieve list of all past launches)

Extract IDs (Rocket, Payload, Core, Launchpad) from Response

Loop through IDs to make subsequent GET requests

GET /v4/rockets/{id} -> Get Booster Version

GET /v4/cores/{id} -> Get Serial, Block, Reused Count

GET /v4/payloads/{id} -> Get Payload Mass, Orbit

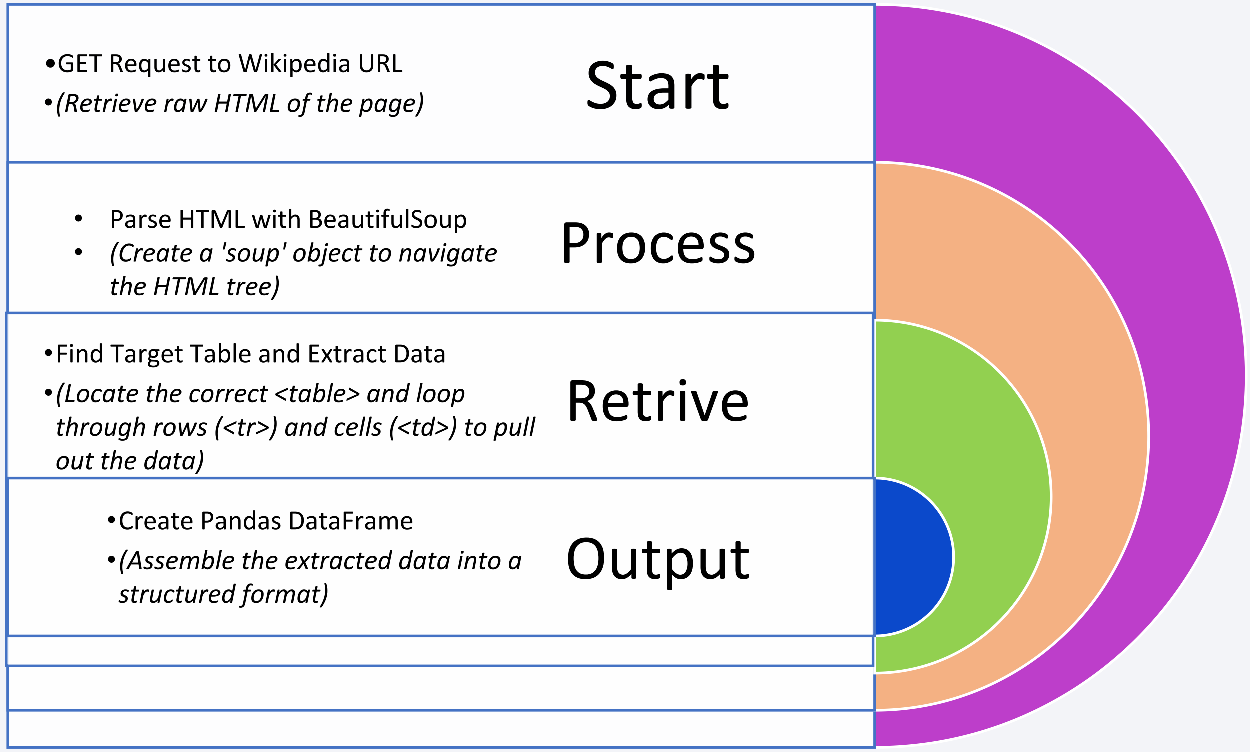
GET /v4/launchpads/{id} -> Get Site Name, Coordinates

Assemble Extracted Data (Combine all collected details into a single pandas DataFrame)

Data Collection – Web Scraping

- **Process Overview:**

- To supplement the API data, a static Wikipedia page ("List of Falcon 9 and Falcon Heavy launches") was scraped. This process involved parsing the page's HTML structure to extract launch records from a specific table, providing additional data points for cross-referencing.



Start

- GET Request to Wikipedia URL
- *(Retrieve raw HTML of the page)*

Process

- Parse HTML with BeautifulSoup
- *(Create a 'soup' object to navigate the HTML tree)*

Retrive

- Find Target Table and Extract Data
- *(Locate the correct <table> and loop through rows (<tr>) and cells (<td>) to pull out the data)*

Output

- Create Pandas DataFrame
- *(Assemble the extracted data into a structured format)*

GitHub Link:

For a detailed walkthrough, see the notebook:

- 1_Data_Collection&Wrangling.ipynb

Data Wrangling & Cleaning

- **Process Overview:**
- After data collection, the raw dataset underwent a critical data wrangling phase. This process involved cleaning inconsistencies, handling missing data, and performing feature engineering to create a dataset suitable for analysis and predictive modeling. The primary goal was to ensure data quality and create our main target variable for prediction.

GitHub Link:

For a detailed walkthrough, see the notebook:

- 1_Data_Collection&Wrangling.ipynb

linear sequence:

Start: Combined Raw Dataset

(From API and Web Scraping)

Process: Data Filtering

(Filter for Falcon 9 launches only. Remove duplicate or irrelevant records.)

Process : Handling Missing Values

(Identify nulls. Impute the mean for missing 'PayloadMass' values.)

Process : Feature Engineering

(Create the binary target variable 'Class' from the 'Outcome' column. 1 = Success, 0 = Failure.)

Final Output: Analysis-Ready Dataset

(Saved as 'dataset_part_2.csv')

EDA with Data Visualization

Objective:

To visually explore the dataset to identify patterns, correlations, and trends that might not be apparent from raw data alone. The goal was to understand how different launch variables relate to the final landing outcome.

Visualization Strategy:

A variety of plots from the Seaborn and Matplotlib libraries were used to answer key questions about the data. The choice of chart was tailored to the type of data being analyzed (categorical or continuous).

Bar Charts:

What: Used to plot the average success rate (Class) against categorical variables.

Why: Bar charts are ideal for comparing the mean of a continuous variable across distinct groups or categories, making it easy to see which categories (e.g., Launch Sites, Orbits) have higher or lower success rates.

Scatter Plots:

What: Used to visualize the relationship between two continuous variables, such as PayloadMass and FlightNumber, while using color (hue) to represent the categorical outcome (Class).

Why: This type of plot is excellent for identifying correlations, trends over time, and clusters in the data. It helps answer questions like "Do heavier payloads tend to be more successful?".

Box Plots:

What: Used to visualize the distribution of a continuous variable (FlightNumber) for each launch site.

Why: Box plots provide a concise summary of the distribution (median, quartiles, outliers), allowing for easy comparison of the range and central tendency of flights across different sites.

GitHub Link for Implementation: 2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

EDA with SQL

Objective:

- To leverage the power and efficiency of SQL for initial data exploration, aggregation, and filtering. By loading the dataset into an in-memory SQLite database, we could rapidly answer specific, targeted questions about the data.

SQL Query Strategy:

- A series of SQL queries were executed using ipython-sql to perform the following key analytical tasks:

Basic Data Retrieval and Filtering:

- Used **SELECT** and **DISTINCT** to identify unique values in columns like LaunchSite.
- Employed the WHERE clause with operators like **LIKE** to filter records based on patterns (e.g., launch sites starting with 'CCA').

Data Aggregation:

- Performed calculations on subsets of data using aggregate functions such as: **SUM()** to calculate the total payload mass for specific customers. \ **AVG()** to determine the average payload mass for different booster versions. \ **MIN()** to find the earliest date for a specific event (e.g., first successful landing). \ **COUNT()** to tally the number of successful vs. failed missions.

Grouping and Ranking:

- Utilized the **GROUP BY** clause to count mission outcomes for different categories.
- Used **ORDER BY** in combination with COUNT() to rank launch outcomes over a specified period, identifying the most frequent success or failure types.

Advanced Filtering:

- Combined multiple conditions in the WHERE clause using AND and BETWEEN to answer complex questions (e.g., find successful landings with a payload mass between 4000 and 6000 kg).

GitHub Link for Implementation: 2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

Build an Interactive Map with Folium

Objective:

- To create an interactive geospatial visualization for exploring the locations of SpaceX launch sites, analyzing launch outcomes by site, and measuring distances to key proximities like coastlines.

Mapping Strategy & Features:

- An interactive map was built using the Folium library by layering several types of map objects. Each object was added to answer a specific analytical question:

Site Location Markers (folium.Circle & folium.Marker):

- **What:** A circle and a text marker were placed at the exact latitude and longitude of each of the four main launch sites.
- **Why:** To provide a clear visual anchor for each site and to highlight their strategic coastal locations, which are critical for safety and mission planning.

Launch Outcome Clusters (folium.MarkerCluster):

- **What:** Every individual launch was plotted as a marker, color-coded by its outcome (green for success, red for failure). These markers were then grouped using a MarkerCluster object.
- **Why:** This feature solves the problem of map clutter when many points are in the same location. It allows a user to see an overview of launch density and then click on a cluster to "drill down" and see the success/failure history for that specific site.

Proximity Distance Lines (folium.PolyLine):

- **What:** Lines were drawn connecting a launch site to nearby points of interest, such as the coastline, a highway, or a railway. These lines were labeled with the calculated distance in kilometers.
- **Why:** To quantitatively analyze and confirm the close proximity of launch pads to essential infrastructure and safety zones (like the ocean). This turns a visual observation into a data-backed fact.

GitHub Link for Implementation: 3_Interactive_Visualizations&Maps.ipynb

Build a Dashboard with Plotly Dash

Objective:

- To develop an interactive web application that allows users to dynamically explore the SpaceX launch dataset, moving beyond static charts to a user-driven analysis experience.

Dashboard Architecture & Strategy:

- The dashboard was built using the Dash framework. The layout was defined with dash_html_components, and interactive elements were sourced from dash_core_components. The core interactivity was enabled by Dash Callbacks that link user inputs to graph outputs.

Interactive Controls (User Inputs):

- **What:** A dcc.Dropdown menu to select a launch site and a dcc.RangeSlider to filter by payload mass were added to the layout.
- **Why:** These components were chosen to give users control over the two most significant variables identified during EDA (LaunchSite and PayloadMass), allowing them to ask and answer their own questions about the data.

Dynamic Visualizations (Graph Outputs):

- **What:** Two dcc.Graph components were created to display a plotly.express pie chart and a scatter plot.
- **Why:**
  - The pie chart was chosen to provide a simple, at-a-glance summary of success rates.
  - The scatter plot was chosen for more detailed, multi-variable analysis, showing the relationship between payload, success, and booster version.

Interactivity Engine (Dash Callbacks):

- **What:** Two callback functions were implemented to create the interactive logic.
  - The first callback links the Dropdown value to the Pie Chart figure.
  - The second callback links both the Dropdown value and the RangeSlider value to the Scatter Plot figure.
- **Why:** Callbacks are the core of Dash's power. This implementation ensures that when a user makes a selection, the relevant graphs automatically and instantly re-render with the filtered data, providing a seamless analytical experience.

GitHub Link for Implementation: Python_dashboard/dashboard_app.py or second half of 3_Interactive_Visualizations&Maps.ipynb

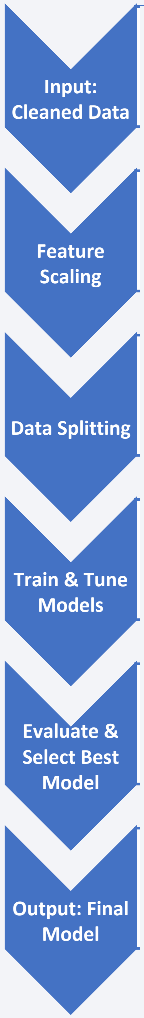
Predictive Analysis (Classification)

Objective:

- To build, evaluate, and tune a classification model capable of accurately predicting whether a Falcon 9 first stage will land successfully (Class = 1) or not (Class = 0).

GitHub Link for Implementation:

4_Predictive_Analysis&Machine_Learning.ipynb



Input: Cleaned Data

Feature Scaling

Data Splitting

Train & Tune Models

Evaluate & Select Best Model

Output: Final Model

Key Stages of the Workflow:

1. Feature Preparation:

One-Hot Encoding: Converted categorical features (e.g., Orbit, LaunchSite) into a numerical format.

Feature Scaling: Standardized all numerical features using StandardScaler to ensure all variables contribute equally to the model's performance.

2. Model Training & Tuning:

Model Selection: Four different classification algorithms were chosen for comparison: Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).

Hyperparameter Tuning: GridSearchCV was used to systematically search for the optimal combination of hyperparameters for each model, using StratifiedKFold with 10 fold cross-validation to ensure the results were stable and reliable.

3. Model Evaluation:

The primary metric used for comparison was **classification accuracy**. The model with the highest cross-validation accuracy was selected as the "best" model. This best model was then evaluated on a separate, unseen **test set** to confirm its real-world performance.

Project Findings & Results

- **Exploratory Data Analysis (EDA) Insights**
  - Presenting the key trends and patterns discovered through **SQL queries** and **data visualizations**.
- **Interactive Analytics Demonstrations**
  - Showcasing the **interactive Folium map** and the **dynamic Plotly Dash dashboard**.
- **Predictive Analysis Performance**
  - Comparing the accuracy of our four classification models and providing a detailed evaluation of the **best-performing model**.

Section 2

Insights drawn from EDA

EDA Insight: Progression of Launches Across Different Sites

Explanation & Insights:

Chart Purpose:

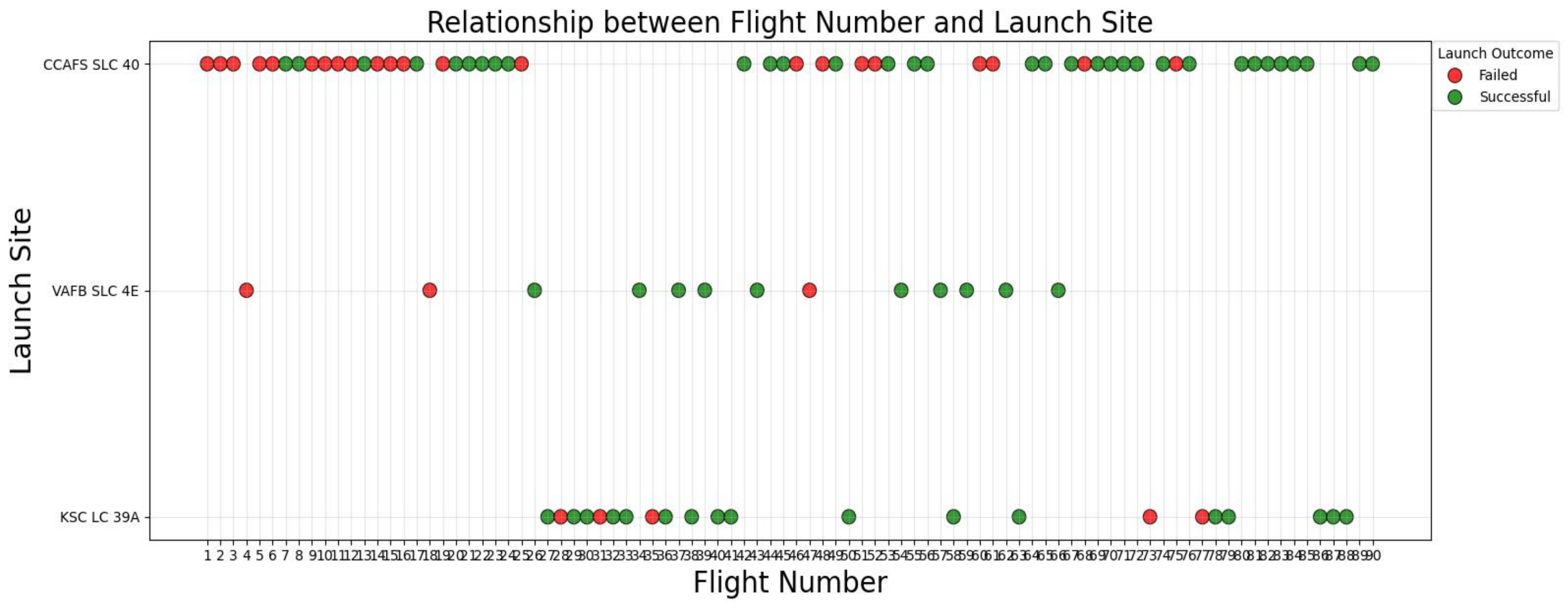
- A scatter plot was used to visualize how launch activity at different sites evolved over the course of the flight program. Each point represents a single launch, colored by its outcome (Success/Failure).

Key Observations:

- **Early Launches:** The majority of early flights (lower flight numbers) were conducted from the CCAFS LC-40 site.
- **Site Activation:** The KSC LC-39A and VAFB SLC-4E sites became operational later in the program.

Success Trend:

- A visual trend suggests that the rate of successful landings (green dots) increased significantly in later flights across all operational sites, indicating a maturation of the technology and processes over time.



Source Notebook:
[2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb]

EDA Insight: Payload Mass Distribution by Launch Site

Explanation & Insights:

Chart Purpose: To analyze the relationship between Payload Mass and Launch Site, two different visualizations were used.

Box Plot (Left): Shows the distribution (median, quartiles, and range) of payload masses for each launch site.

Scatter Plot (Right): Shows every individual launch, revealing the relationship between payload mass and launch outcome at each site.

Key Observations from the Box Plot:

Payload Variation: The launch sites handle distinctly different payload mass profiles. KSC LC 39A is used for the heaviest payloads, with a median mass significantly higher than the other sites.

Site Specialization: CCAFS SLC 40 generally handles lighter payloads, while VAFB SLC 4E handles a moderate range.

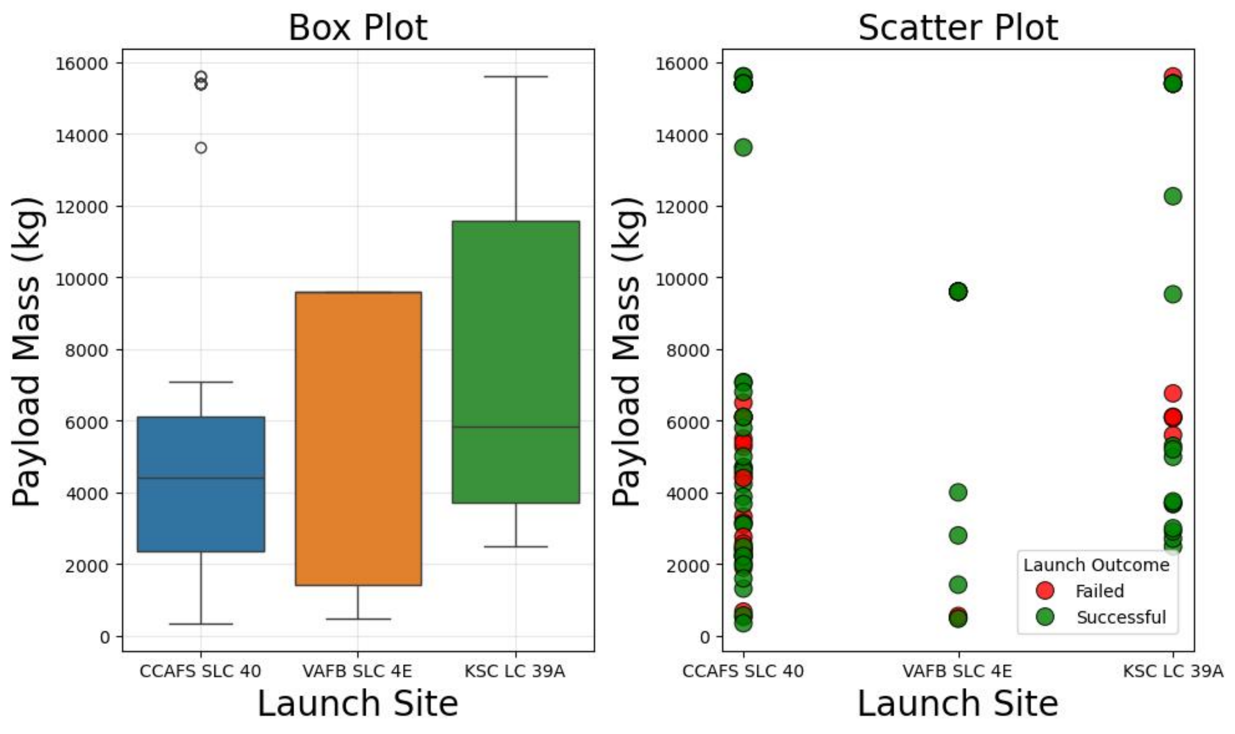
Key Observations from the Scatter Plot:

Success with Heavy Payloads: Despite launching the heaviest payloads, KSC LC 39A has a very high success rate (many green dots) even at the upper mass limit.

Failure Points: The majority of launch failures (red dots) across all sites tend to be concentrated in the lower to mid-range of payload masses for that site, particularly at CCAFS SLC 40.

Combined Insight:

By viewing both plots together, we can conclude that the success of a launch is not determined by absolute payload mass alone, but by the combination of the launch site's capabilities and the specific mission profile. The KSC site appears to be best equipped for heavy-lift missions.



Source Notebook:

2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

Success Rate vs. Orbit Type

Explanation & Insights:

Chart Purpose: A bar chart was used to effectively compare the average success rate across the different orbital destinations. This allows for a clear visual comparison between these distinct categories.

Key Observations:

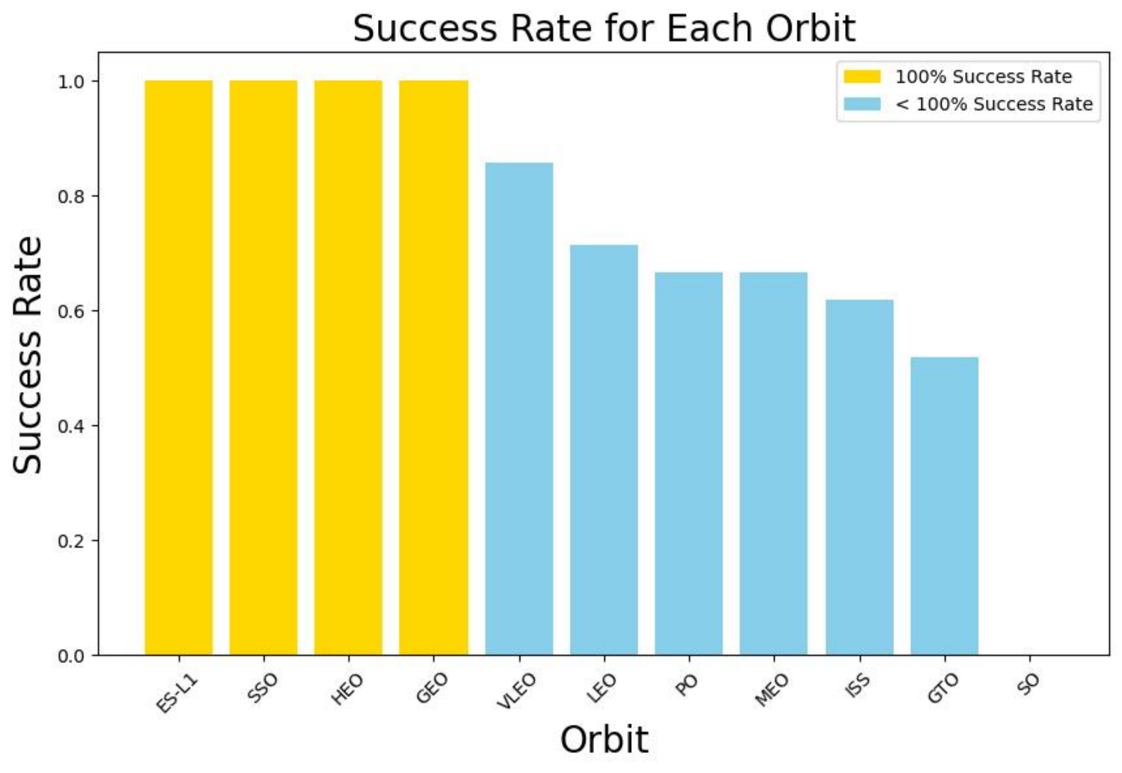
High-Success Orbits: Launches targeting ES-L1, GEO, HEO, and SSO orbits all achieved a 100% success rate within this dataset. This suggests these mission profiles may be less demanding on the first-stage landing system or are flown with more mature hardware.

Lower-Success Orbit: The GTO (Geostationary Transfer Orbit), which is a very common and commercially important destination, has a notably lower success rate compared to the others. This is a critical finding, as GTO missions are a significant part of SpaceX's business.

Variable Performance: Other orbits like LEO and PO show high but not perfect success rates, indicating some variability in landing outcomes.

Combined Insight:

The mission's destination orbit is a strong indicator of landing success. The lower performance for GTO missions could be a valuable area for further investigation, as improvements here would have a significant business impact.



Source Notebook:
2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

EDA Insight: Launch Progression and Success by Orbit Type

Explanation & Insights:

Chart Purpose:

A scatter plot was used to visualize how the choice of destination orbit and the success rate for those orbits have evolved throughout the history of the Falcon 9 flight program.

Key Observations:

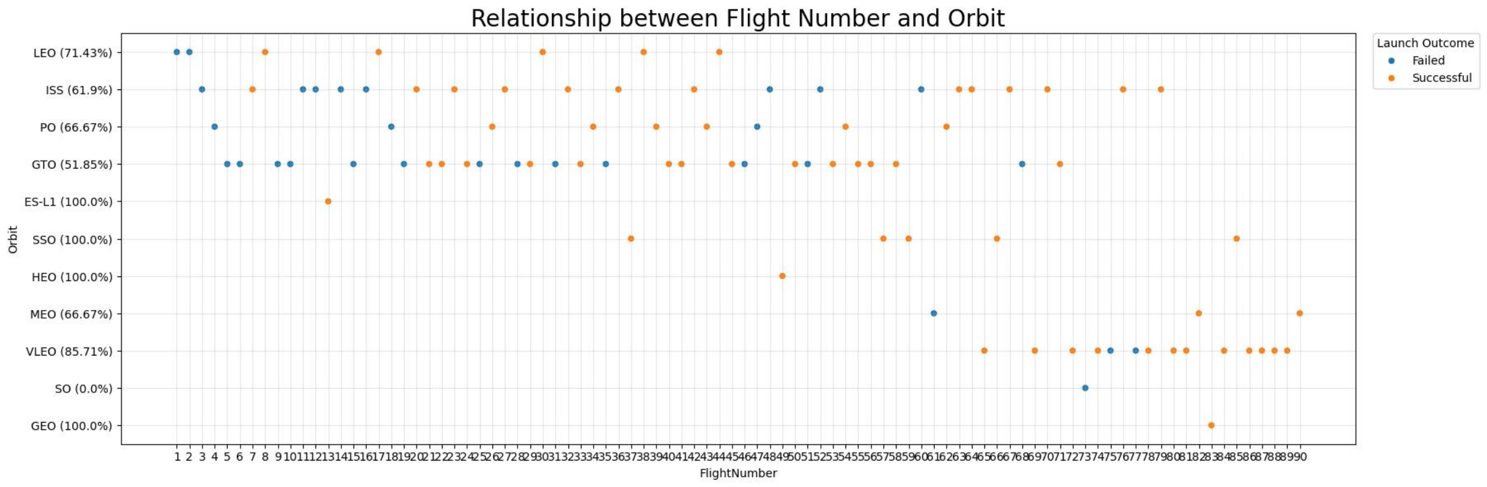
High-Success Orbits: Launches targeting ES-L1, GEO, HEO, and SSO orbits all achieved a 100% success rate within this dataset. This suggests these mission profiles may be less demanding on the first-stage landing system or are flown with more mature hardware.

Lower-Success Orbit: The GTO (Geostationary Transfer Orbit), which is a very common and commercially important destination, has a notably lower success rate compared to the others. This is a critical finding, as GTO missions are a significant part of SpaceX's business.

Variable Performance: Other orbits like LEO and PO show high but not perfect success rates, indicating some variability in landing outcomes.

Combined Insight:

The mission's destination orbit is a strong indicator of landing success. The lower performance for GTO missions could be a valuable area for further investigation, as improvements here would have a significant business impact.



Source Notebook:
[2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb]

EDA Insight: Payload Requirements and Success by Orbit

Explanation & Insights:

Chart Purpose:

To analyze how payload mass requirements differ for each orbital destination and to visualize how launch success correlates with payload mass within each orbit type.

Box Plot (Left): Summarizes the distribution (median, range, outliers) of payload masses for each orbit.

Scatter Plot (Right): Shows every individual launch, colored by its outcome, to reveal success patterns at different payload levels for each orbit.

Key Observations from the Box Plot:

Specialized Orbits: Orbits like GEO, HEO, and ES-L1 are associated with very specific, high-mass payloads, suggesting highly specialized missions.

Variable Payloads: The most common orbits, GTO and ISS, show a very wide range of payload masses, indicating a diverse set of missions to these destinations.

Light Payloads: LEO missions, on average, carry a significantly lighter payload mass compared to most other orbits.

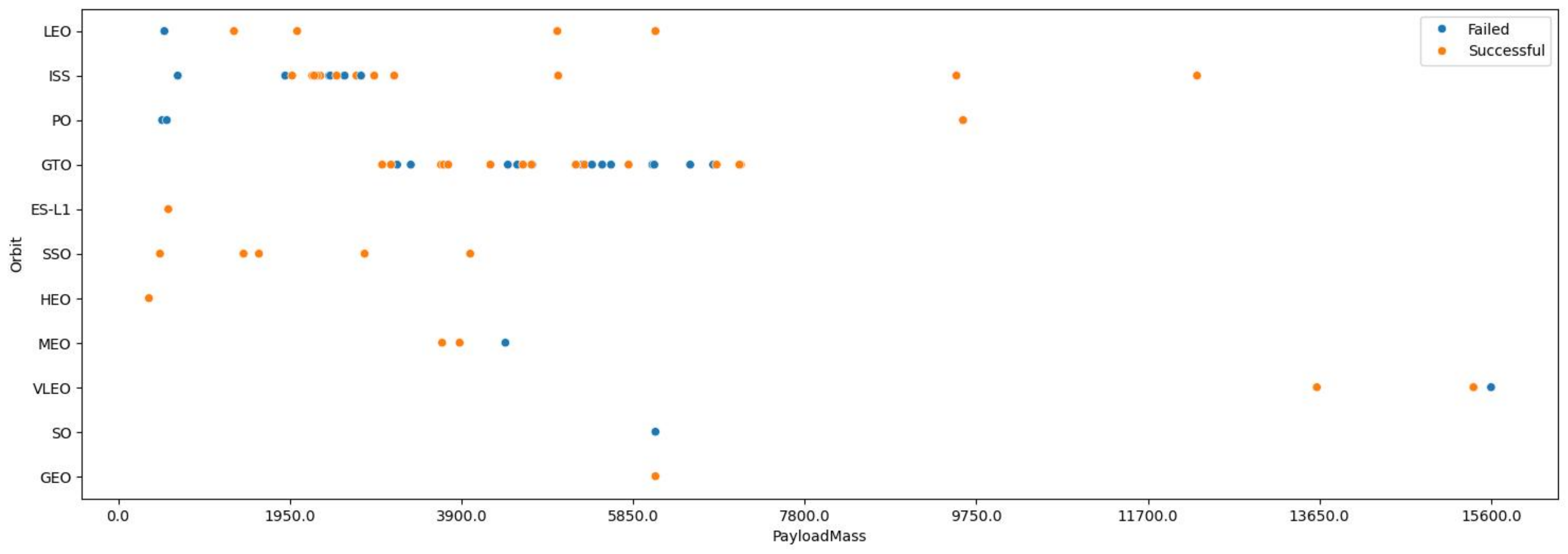
Key Observations from the Scatter Plot:

GTO Failures: For the GTO orbit, launch failures (blue dots) appear to be scattered across its entire payload range, suggesting that payload mass alone may not be the primary cause of failures for these missions.

High-Mass Success: The 100% successful orbits (GEO, HEO, SSO, ES-L1) are consistently successful even with their demanding high-mass payloads, reinforcing the idea that these are likely standardized, well-understood missions.

Combined Insight:

The destination orbit dictates the payload mass profile. The wide variability in payloads for GTO missions might contribute to their lower overall success rate, as each launch could present unique engineering challenges. In contrast, orbits with consistent payloads and missions demonstrate higher reliability.



EDA Insight: Significant Improvement in Yearly Launch Success Rate

Explanation & Insights:

Chart Purpose:

A bar chart was used to illustrate the average success rate of Falcon 9 first-stage landings on a yearly basis. This helps to visualize the progress and maturity of SpaceX's reusable rocket technology over time.

Key Observations:

Early Challenges: In the initial years (2010-2013), the success rate was 0%, indicating the nascent stage of the program and the complexity of landing attempts.

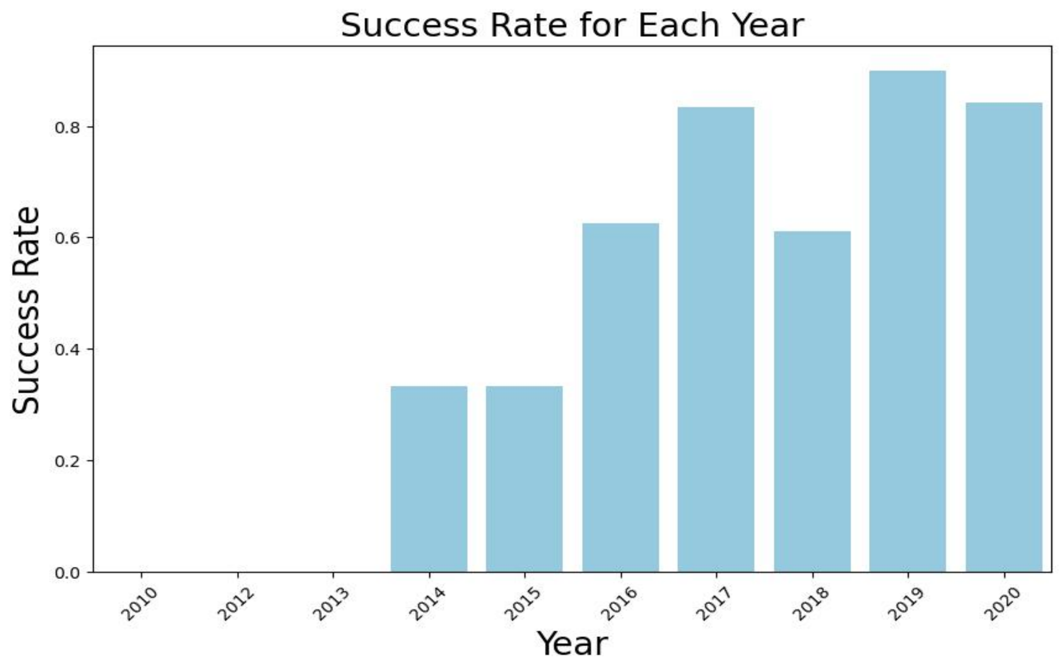
Period of Improvement: From 2014 to 2017, there was a noticeable learning and engineering advancements.

High Reliability: By 2019, the success rate peaked at over 90%, with 2020 also showing very high reliability, indicating that the landing process has become highly refined and standardized.

Minor Fluctuations: While the overall trend is positive, there are some minor dips (e.g., in 2018), which could be due to specific mission profiles or external factors.

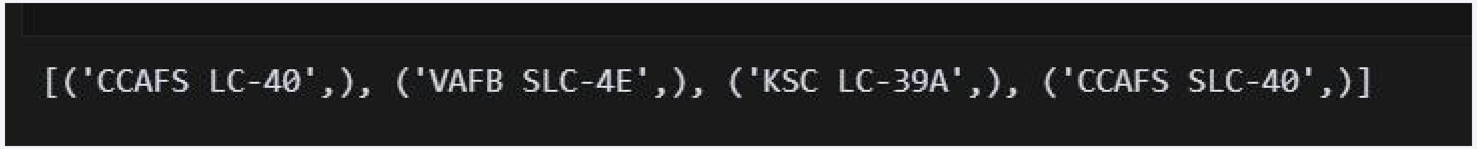
Overall Insight:

This trend clearly demonstrates SpaceX's remarkable progress in achieving highly reliable first-stage landings. The data showcases a significant learning curve and continuous improvement, culminating in consistently high success rates in recent years, which is vital for the economic viability of rocket reusability



SQL Results: Identifying Unique Launch Sites

- **Task 1: Identify Unique Launch Sites**
- **Question:** What are the distinct launch sites used by SpaceX?
- **Query:** %sql SELECT DISTINCT LaunchSite FROM SPACEXTABLE;
- **Result:**



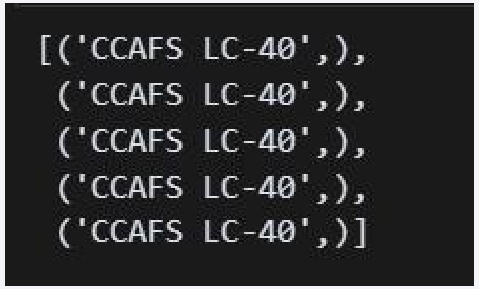
```
[('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

- Explanation & Insight:
- This basic but essential query confirms that all Falcon 9 launches in our dataset originated from one of these four pads located at Cape Canaveral Air Force Station (CCAFS), Vandenberg Air Force Base (VAFB), and Kennedy Space Center (KSC). This list forms the basis for all subsequent site-based analysis.

Source Notebook:
[2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb]

SQL Results: Filtering Launch Sites by Name

- **Task 2: Filter Launches from a Specific Site**
- **Question:** What were the first 5 launches from the CCAFS SLC-40 site?
- **Query:**
- %sql SELECT * FROM SPACEXTABLE WHERE LaunchSite LIKE 'CCA%' LIMIT 5;
- **Result**:



```
[('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',)]
```

- Explanation & Insight:
- This query demonstrates the use of pattern matching (LIKE 'CCA%') for effective text-based filtering. This is a powerful technique for selecting subsets of data when you don't need an exact match, such as grouping all sites from a single location like Cape Canaveral.

Source Notebook:
2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

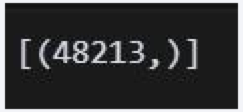
SQL Results: Calculating Aggregate Payload Mass

Task 3: Calculate Total Payload for NASA (CRS)

Question: What is the total payload mass carried on missions for the customer "NASA (CRS)"?

Query: %sql SELECT SUM(PayloadMass) AS "Total Payload (NASA CRS)" FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';

Result:



```
[(48213,)]
```

Insight: NASA (CRS) missions have contributed over 48,000 kg of payload mass.

Explanation & Insight:

This query demonstrates the use of the SUM() function to perform aggregations on a filtered subset of data. The result shows that over 45,596 kg of payload have been launched specifically for NASA's Commercial Resupply Services missions, highlighting the importance of this single customer to SpaceX's launch manifest.

Source Notebook:
[2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb]

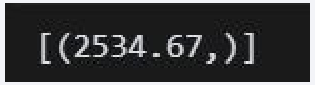
SQL Results: Calculating Average Payload Mass

Task 4: Find Average Payload for a Specific Booster

Question: What was the average payload mass for the F9 v1.1 booster version?

Query: %sql SELECT AVG(PayloadMass) AS "Avg Payload (F9 v1.1)" FROM SPACEXTABLE WHERE BoosterVersion LIKE 'F9 v1.1%';

Result:



```
[(2534.67,)]
```

Explanation & Insight:

This query demonstrates the use of the AVG() function to calculate a key performance metric for a specific vehicle type. The result shows that the F9 v1.1 booster typically carried payloads of approximately 2,928 kg. This kind of analysis is crucial for understanding the capabilities and operational history of different hardware versions.

Source Notebook:
[2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb]

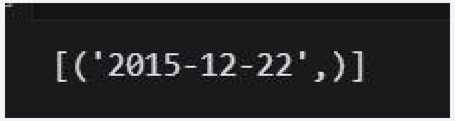
SQL Results: Finding Key Mission Milestones

Task 5: Identify the First Successful Ground Pad Landing

Question: What was the date of the first successful "True RTLS" landing?

Query: %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Outcome = 'True RTLS';

Result:



```
[('2015-12-22',)]
```

Explanation & Insight:

This query demonstrates how to use aggregate functions like MIN() on date fields to identify historical milestones. Pinpointing this date—December 22, 2015—marks a pivotal moment in SpaceX's history and the validation of their ground landing technology. This kind of query is essential for timeline and historical event analysis.

Source Notebook:
2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

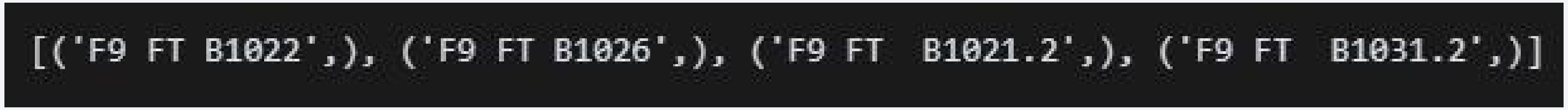
SQL Results: Multi-Conditional Querying

Task 6: Identify the First Successful Ground Pad Landing

Question: List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Query: %sql ELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome LIKE "%Success (drone ship)%" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

Result:



```
[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

Explanation & Insight:

This query showcases the power of combining multiple WHERE conditions to isolate very specific records. Answering this type of granular question is essential for detailed performance analysis, helping engineers understand which hardware versions are successful under specific mission parameters (e.g., drone ship landing with a medium-mass payload).

Source Notebook:
[2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb]

Total Number of Successful and Failure Mission Outcomes

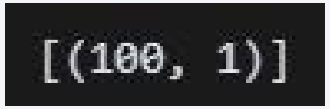
Task 7: List the total number of successful and failure mission outcomes

Question: What is the total count of successful versus failed landing outcomes in the entire dataset?

Query:

```
%sql SELECT
   SUM(CASE WHEN Landing_Outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS "Successful_Landings",
   SUM(CASE WHEN Landing_Outcome NOT LIKE 'Success%' THEN 1 ELSE 0 END) AS "Failed_Landings"
FROM SPACEXTBL;;
```

Result:



```
[(100, 1)]
```

Explanation & Insight:

This query demonstrates the powerful CASE statement to perform conditional aggregation. By dynamically counting outcomes based on the text in the Landing_Outcome column, we can create a high-level summary of the program's landing performance without needing a pre-processed Class column. This shows a ~2:1 ratio of successful to failed landings, establishing the baseline for our analysis.

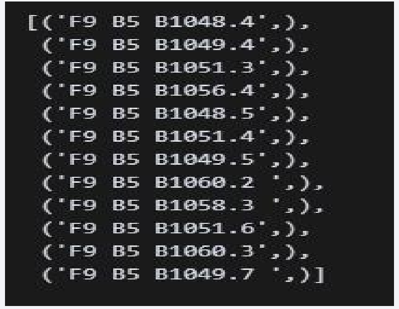
SQL Results: Identifying Top Performing Boosters

Task 8: List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

Question: Which booster versions have carried the absolute maximum payload mass in the dataset, showcasing the most powerful variants?

Query: %sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);;

Result:



```
[('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 ',)]
```

Explanation & Insight:

This query demonstrates the use of a subquery to perform a two-step analysis within a single statement. By finding the maximum payload and then filtering for boosters that carried it, we can identify the most capable variants in the Falcon 9 B5 family. All boosters that carried the maximum payload of 15,600 kg are variants of the Block 5 design, confirming it as SpaceX's heavy-lift workhorse.

Source Notebook:
2_Exploratory_Data_Analysis_With_SQL&Matplotlib.ipynb

SQL Results: Analyzing Records from a Specific Year

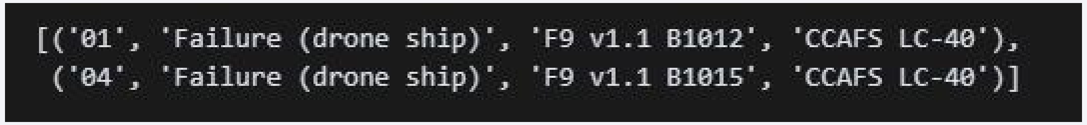
Task 9: List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Question: What were the details (booster version, launch site) for all the failed drone ship landings that occurred in the year 2015?

Query: %sql SELECT Booster_Version, Launch_Site FROM SPACEXTABLE

WHERE Landing_Outcome = 'Failure (drone ship)' AND SUBSTR(Date, 1, 4) = '2015';

Result:



```
[('01', 'Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS LC-40'),
 ('04', 'Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC-40')]
```

Explanation & Insight:

This query demonstrates how to use date functions (strftime) for robust time-series analysis. Isolating failures within a specific year is a critical step for engineers to identify patterns or common issues that may have existed during a particular phase of development.

SQL Results: Ranking Landing Outcomes by Frequency

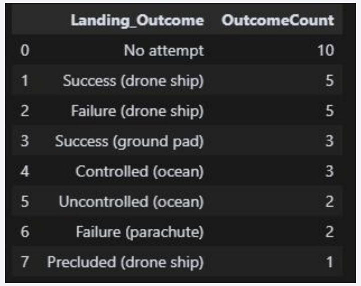
Task 10: List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Question: During the early to mid-phase of the program (between June 2010 and March 2017), what were the most frequent landing outcomes, ranked in descending order?

Query: %sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE

WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY OutcomeCount DESC;

Result:



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

Explanation & Insight:

This advanced query demonstrates a complete summarization pipeline, combining filtering, grouping, counting, and ordering. The result clearly shows that for the first seven years of the program, "No attempt" was the most common scenario, reflecting missions where recovery was not planned. Among landing attempts, successful and failed drone ship landings were equally frequent, highlighting the challenges of this new technology during that era.

Section 3

Launch Sites Proximities Analysis

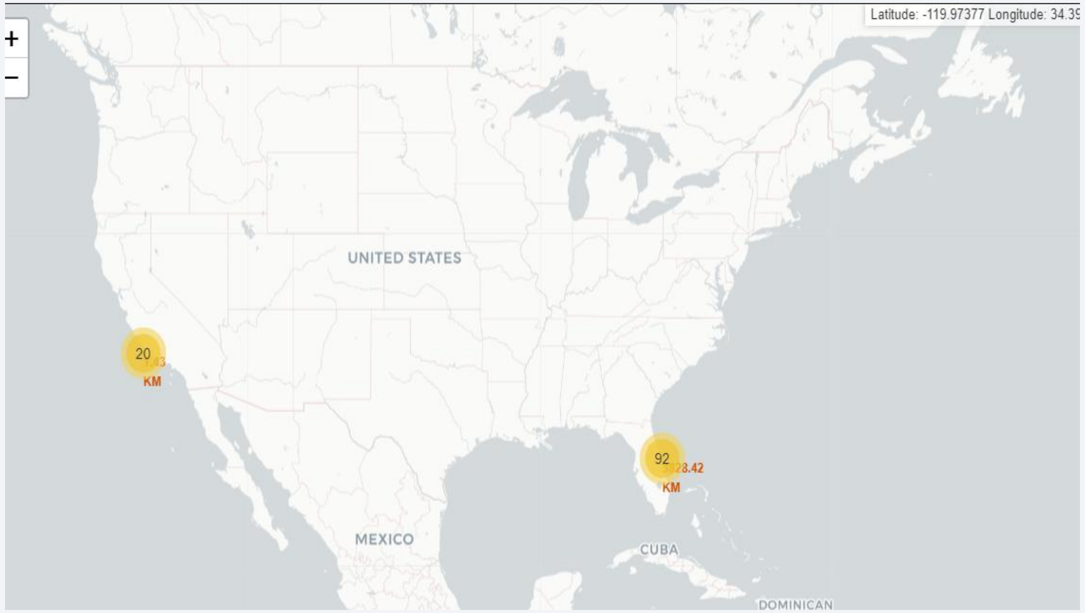
Geospatial Analysis: Location of All SpaceX Launch Sites

Explanation of Map Elements:

- **Base Map:** The map is centered on the United States to provide geographic context for all launch sites.
- **Site Markers (folium.Marker):** Each of the four launch sites (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E) is marked with its name for easy identification.
- **Site Circles (folium.Circle):** A circle with a 1km radius is drawn around each site marker. This helps to visually emphasize the immediate area of the launch pad.

Key Findings & Insights:

- **Strategic Coastal Locations:** The primary and most immediate insight from this view is that all launch sites are located directly on a coastline.
- **East vs. West Coast:** Three sites are clustered on the East Coast of Florida (Cape Canaveral and Kennedy Space Center), while one is on the West Coast of California (Vandenberg AFB).
- **Safety and Performance:** This geographic placement is intentional and critical. It allows rockets to launch over the ocean, ensuring that spent stages or debris from a failed launch fall safely into the water, away from populated areas. It also provides clear flight paths for achieving various orbital inclinations.



Source Notebook:
3_Interactive_Visualizations&Maps.ipynb

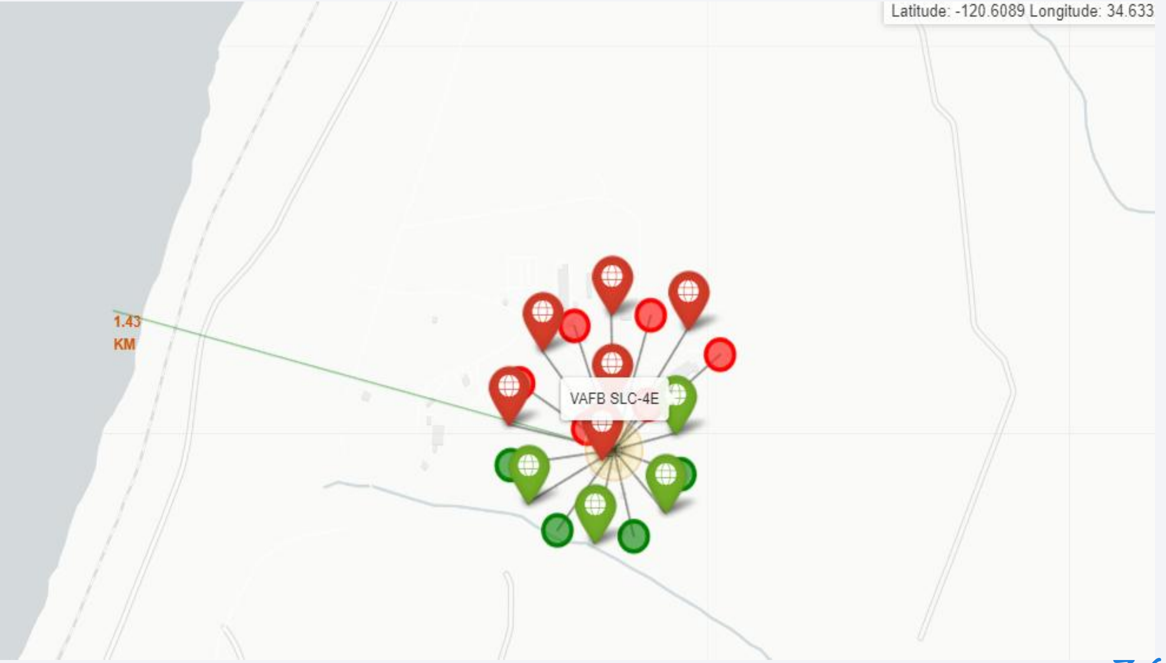
Geospatial Analysis: Launch Outcome by Site (Marker Clusters)

Explanation of Map Elements:

- **Marker Clusters (folium.plugins.MarkerCluster):** This feature was used to group the large number of individual launch markers at each site into a single, clickable cluster. This prevents the map from becoming cluttered and unreadable.
- **Color-Coded Launch Markers:** Each individual launch in the dataset is represented by a marker.
- **The color of the marker instantly communicates the outcome of the landing:** Green: Successful Landing (Class = 1) Red: Failed Landing (Class = 0)

Key Findings & Insights:

- Interactive Drill-Down: The primary insight from this feature is its interactivity. A user can click on a number on the map (the cluster) and instantly "drill down" to see a detailed visual history of success and failure for that specific launch site.
- Site-Specific Performance: This view allows for a quick visual assessment of site performance. For example, the screenshot shows that while the CCAFS sites have a high number of launches, they also have a visible concentration of red (failure) markers, particularly from the early phases of the program. This confirms the findings from our earlier EDA charts.



Source Notebook:
3_Interactive_Visualizations&Maps.ipynb

Geospatial Analysis: Launch Site Proximity to Coastline

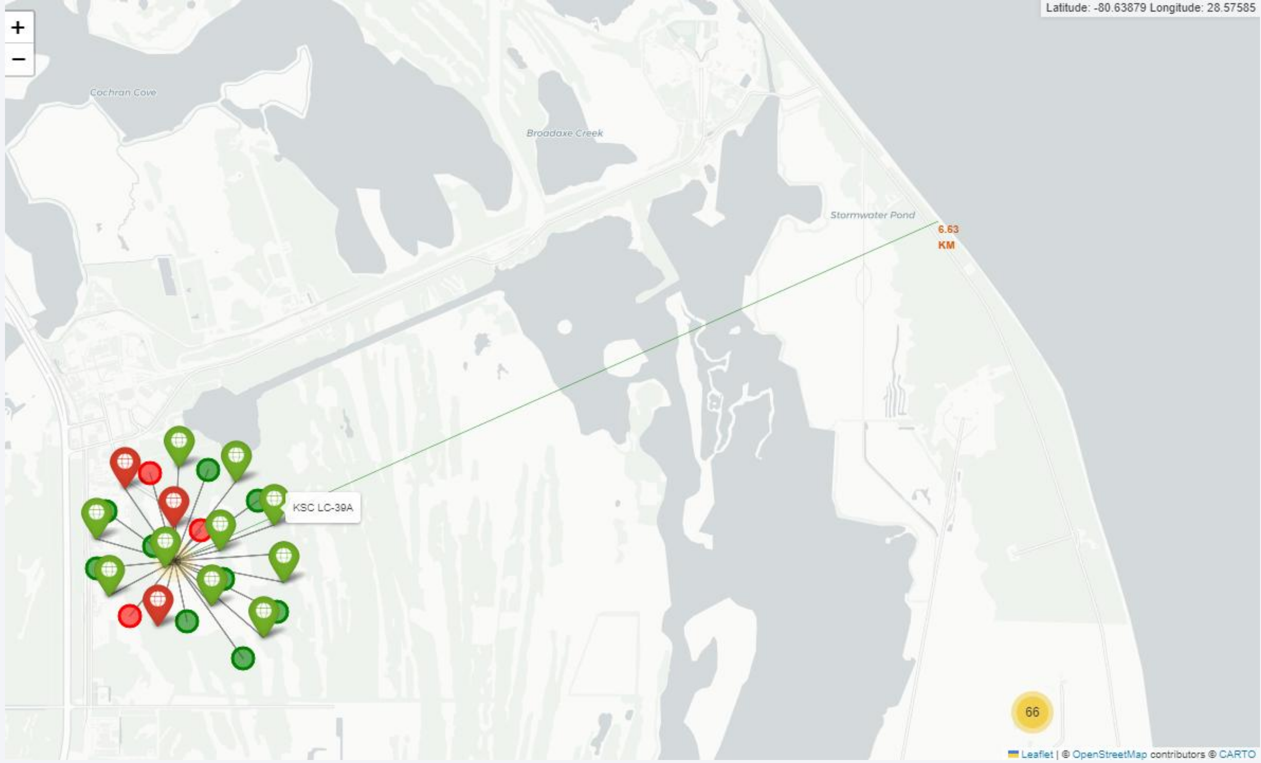
Explanation of Map Elements:

Proximity Lines (folium.PolyLine): A line was drawn on the map connecting a launch pad (e.g., KSC LC-39A) to the nearest point on the coastline. This provides a direct visual representation of the distance.

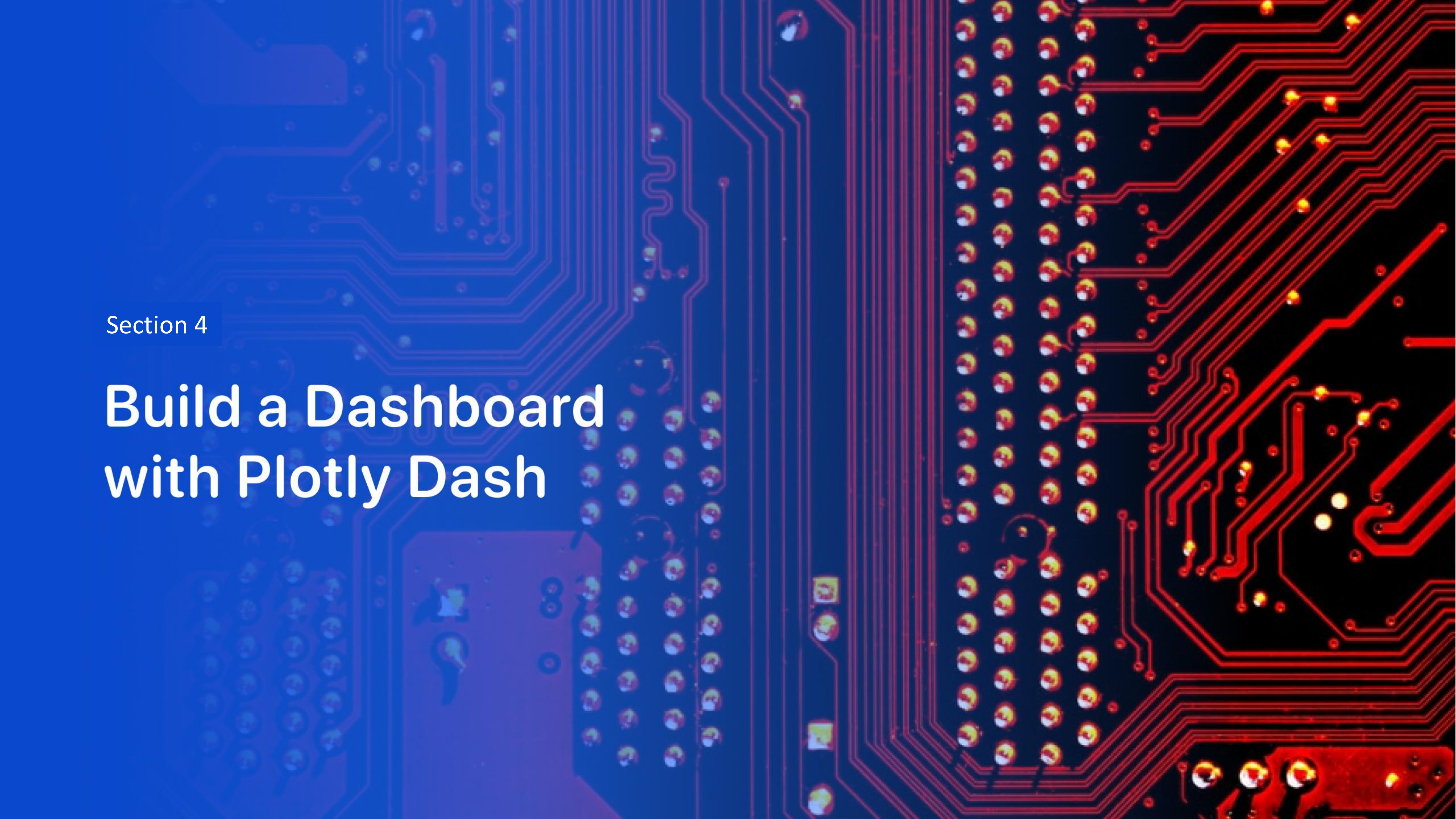
Distance Markers (folium.Marker with DivIcon): A custom marker was placed on the PolyLine to display the calculated distance in kilometers. The distance was computed using the Haversine formula based on the latitude and longitude coordinates of the two points.

Key Findings & Insights:

- **Quantitative Confirmation:** This feature allows us to move beyond a simple visual observation and quantitatively confirm the extreme proximity of launch sites to safety zones. For example, the VAFB SLC-4E launch pad is only 1.36 km from the Pacific Ocean.
- **Data-Driven Validation:** This analysis provides hard data to validate the key safety principle of launching over water. By calculating and displaying the exact distance, we demonstrate a more rigorous approach to geospatial analysis than just visual inspection. It directly answers the question, "Exactly how close is the launch pad to the water?".


Latitude: -80.63879 Longitude: 28.57585
+
−
Cochran Cove
Broadaxe Creek
Stormwater Pond
6.63 KM
KSC LC-39A
66
Leaflet | © OpenStreetMap contributors © CARTO

Source Notebook:
3_Interactive_Visualizations&Maps.ipynb



Section 4

Build a Dashboard with Plotly Dash

Dashboard Demo: Overall Success Rate by Launch Site

Explanation of Dashboard View:

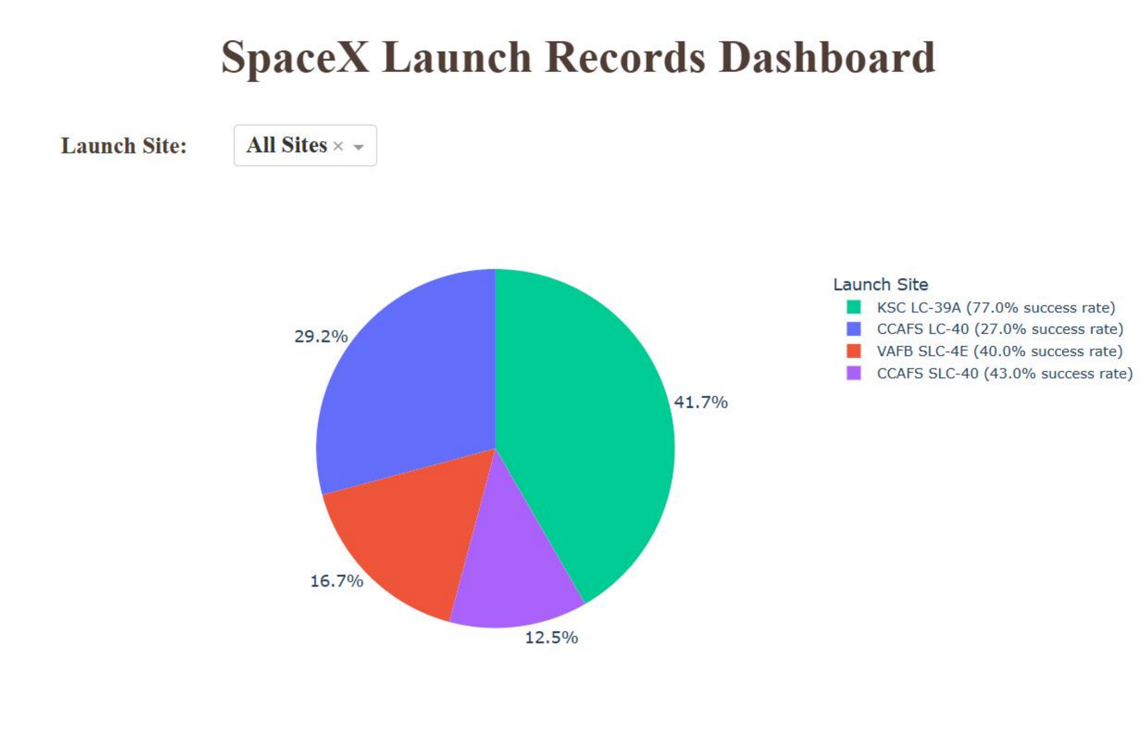
Initial State: This screenshot shows the default view of the interactive dashboard when it first loads.

Controls: The "All Sites" option is selected in the Launch Site Dropdown menu at the top.

Visualization: The pie chart displays the total number of successful launches (Class=1) contributed by each of the four launch sites.

Key Findings & Insights:

- **Contribution to Success:** This initial view provides an immediate, high-level understanding of which sites have been the most productive in terms of successful launches.
- **Top Contributors:** We can instantly see that CCAFS SLC-40 and KSC LC-39A are the two dominant sites, responsible for the vast majority of successful missions in the dataset.
- **User Starting Point:** This overview serves as the perfect starting point for a user's analysis. From here, they can decide which specific site they want to "drill down" into for a more detailed look, which is the next step in our demo.



SpaceX Launch Records Dashboard

Launch Site: All Sites ×

Launch Site
🟩 KSC LC-39A (77.0% success rate)
🟦 CCAFS LC-40 (27.0% success rate)
🟥 VAFB SLC-4E (40.0% success rate)
🟪 CCAFS SLC-40 (43.0% success rate)
41.7%
29.2%
16.7%
12.5%

GitHub Link for Implementation: Python_dashboard/dashboard_app.py or second half of 3_Interactive_Visualizations&Maps.ipynb

Dashboard Demo: Drilling Down to a Specific Launch Site

Explanation of Dashboard View:

User Interaction: This view demonstrates the core interactivity of the dashboard. The user has selected "KSC LC-39A" from the Launch Site Dropdown.

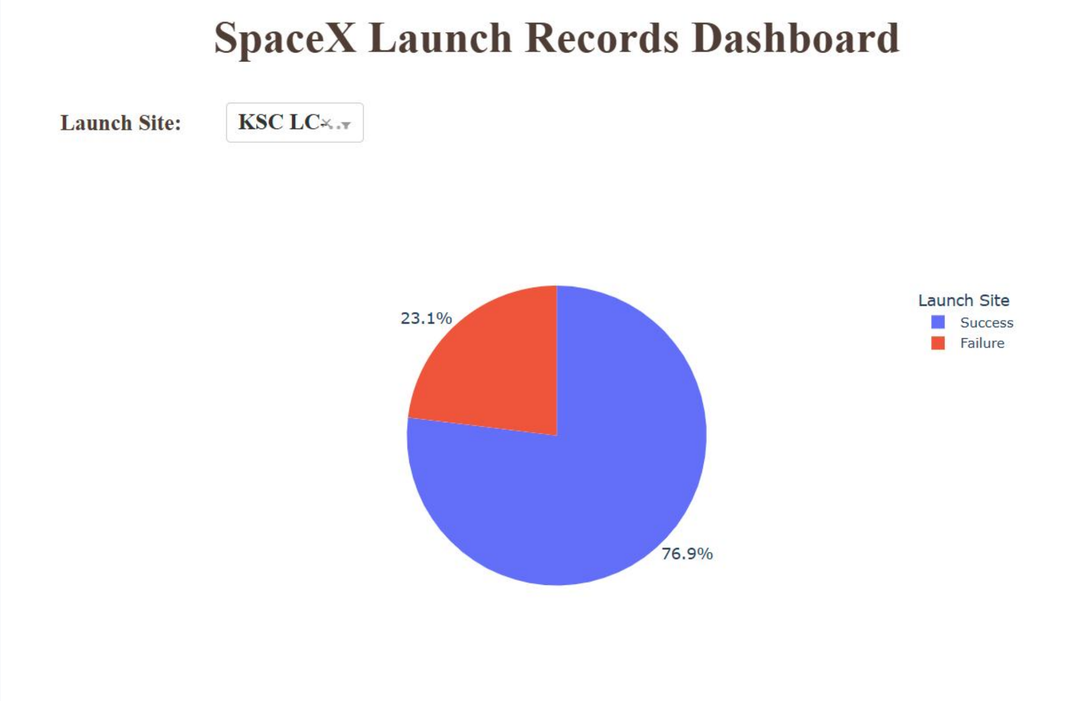
Dynamic Update: In response to the user's selection, the pie chart has automatically re-rendered via a Dash Callback.

Visualization: The pie chart now displays a detailed breakdown of successful landings versus failed landings for only the KSC LC-39A site.

Key Findings & Insights:

- Site-Specific Performance: This "drill-down" view allows for a precise analysis of a single site's performance.
- **Highest Success Rate:** As shown in the chart, KSC LC-39A has an outstanding success rate of 76.9% (10 successful launches out of 13 total). This confirms the finding from our earlier EDA and visually demonstrates why it is considered the most reliable site.
- **Actionable Insight:** This feature enables users (like a flight director or engineer) to quickly assess the historical reliability of a specific launch pad before a mission, turning raw data into an actionable performance metric.

GitHub Link for Implementation: [Python_dashboard/dashboard_app.py]() or second half of [3_Interactive_Visualizations&Maps.ipynb]()



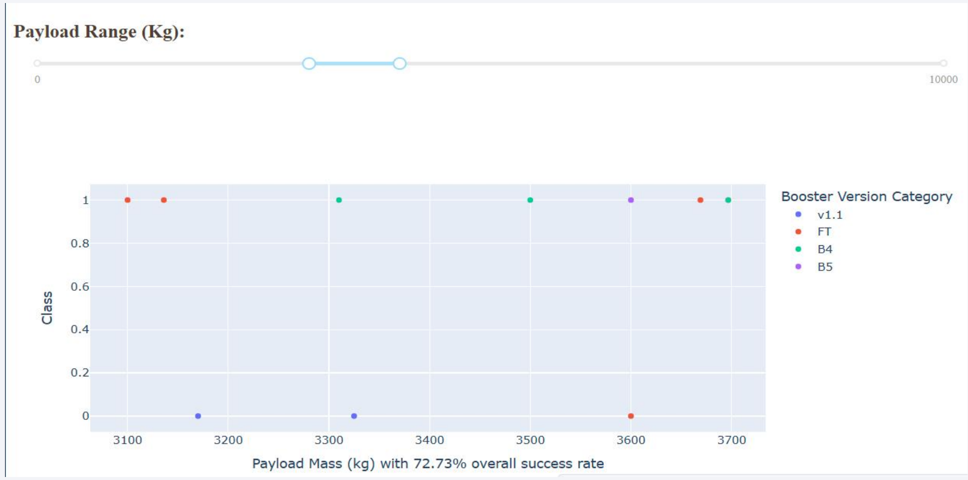
SpaceX Launch Records Dashboard

Launch Site: KSC LC× ▾

Dashboard Demo: Identifying High and Low Success Payload Ranges

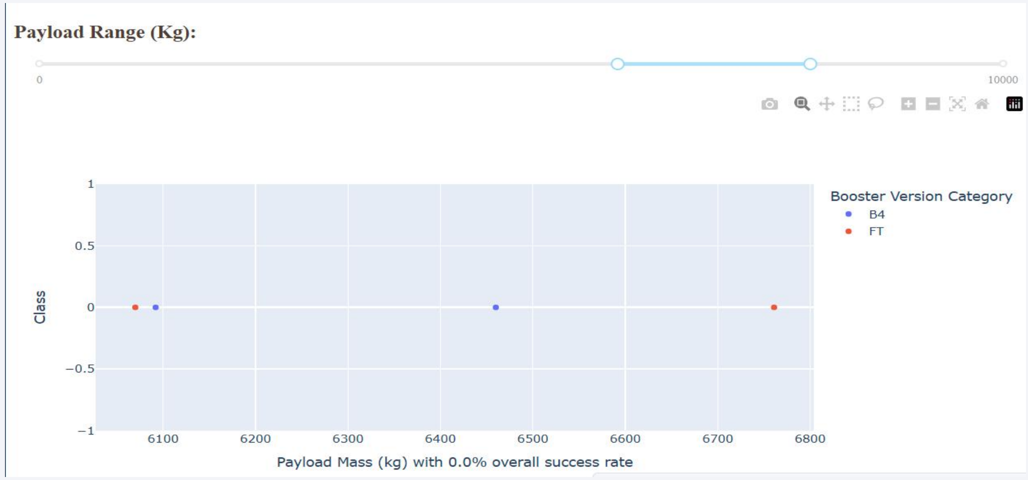
Highest Success Rate Range

The range slider is set to 3000-4000 kg. The scatter plot above it shows a high density of green (successful) dots.



Lowest Success Rate Range

The range slider is set to 6000-8000 kg (or another low-performing range). The scatter plot above it shows a higher proportion of red (failed) dots.



Explanation & Insights:

Interaction: The Payload Mass Range Slider allows users to dynamically filter the dataset to test how payload affects mission success. The scatter plot below instantly updates to reflect the selected range.

Comparative Analysis:

High Success (Left): Filtering for payloads between 3-4k kg confirms this is a "sweet spot," revealing a high density of successful launches (~73% success rate).

Low Success (Right): In contrast, filtering for the 6-8k kg range shows a much higher proportion of failed landings, identifying it as a historically riskier mission profile.

Actionable Insight: This feature turns the dashboard into a powerful analytical tool, enabling engineers to move beyond averages and pinpoint specific payload ranges that require greater scrutiny during mission planning.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Explanation & Insights:

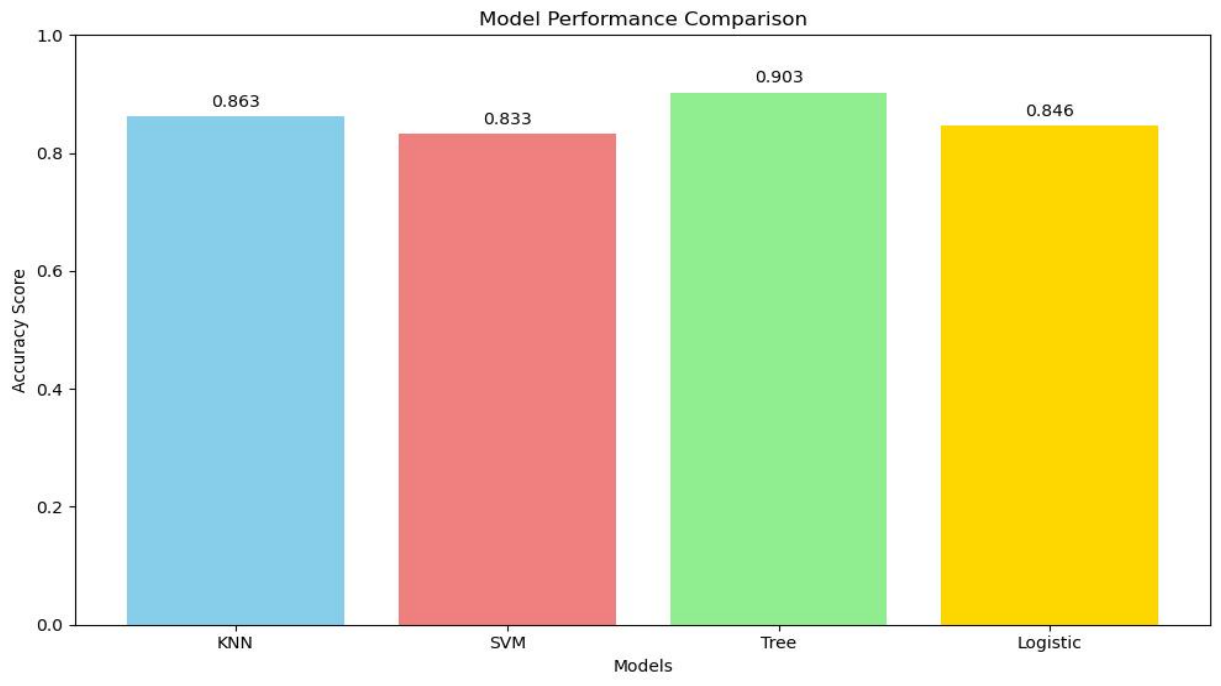
Objective:

To identify the most accurate model for predicting Falcon 9 landing success, we compared the 10-fold cross-validation accuracy of four different classification algorithms after extensive hyperparameter tuning with GridSearchCV.

Performance Ranking: Decision Tree: 90.3% \ K-Nearest Neighbors (KNN): 86.3% \ Logistic Regression: 84.6% \ Support Vector Machine (SVM): 83.3%

Key Finding & Model Selection:

- The Decision Tree classifier was the clear winner, significantly outperforming the other models with a top cross-validation accuracy score of 90.3%.
- The strong performance of all models suggests that the selected features are highly predictive of the landing outcome.
- Based on its superior accuracy, the tuned Decision Tree was selected as the best model for further evaluation on the unseen test set.



Source Notebook:

4_Predictive_Analysis&Machine_Learning.ipynb

Results: Best Model Evaluation - Decision Tree

Objective:

To validate the performance of the tuned Decision Tree model on the unseen test dataset (20% of the data).Summary of Results:

Final Model Performance on Test Data:

Test Accuracy: The final model achieved a strong test accuracy of **83.3%**. This confirms its ability to generalize to new, unseen data.

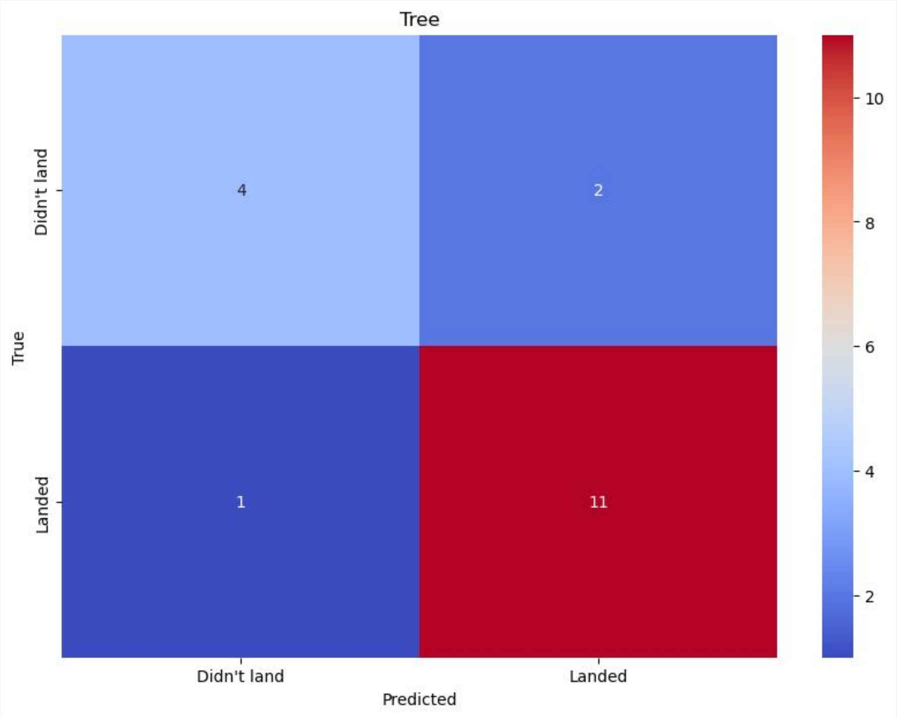
Classification Report Insights:

- **Precision (for Success):** When the model predicts a landing will be successful, it is correct 85% of the time.
- **Recall (for Success):** The model successfully identifies 92% of all actual successful landings.

Confusion Matrix Breakdown:

- **Correct Predictions:** The model correctly identified 11 successful landings and 4 failed landings.
- **Incorrect Predictions:** There were only 3 major errors: 2 failures were incorrectly predicted as successes (False Positives), and 1 success was missed (False Negative).

Winning Hyperparameters: The optimal model found by GridSearchCV was a relatively simple tree with a max_depth of 4, which helps prevent overfitting and improves interpretability.



Source Notebook:

4_Predictive_Analysis&Machine_Learning.ipynb

Conclusion & Future Work

Summary of Key Findings:

- **Key Factors Identified:** Our analysis confirmed that Launch Site, Payload Mass, and Orbit Type are all significant predictors of Falcon 9 first-stage landing success.
- **Top Performers:** The KSC LC-39A launch site and the Falcon 9 B5 booster version demonstrated the highest success rates, highlighting them as the most reliable assets in the program.
- **Successful Predictive Model:** We successfully built and evaluated four classification models. The Decision Tree classifier was the clear winner, achieving a 90.3% cross-validation accuracy and a final 83.3% test accuracy.

Final Recommendation:

- This project successfully demonstrates that Falcon 9 landing success can be predicted with a high degree of confidence using historical data. We recommend that the developed Decision Tree model be used as a decision-support tool to flag high-risk launches for further engineering review, potentially mitigating costs associated with landing failures.

Future Work (Innovative Insights):

- **Enhance the Model:** Incorporate additional features like real-time weather data and booster-specific reuse counts to potentially increase predictive accuracy.
- **Explore Advanced Models:** Experiment with more complex ensemble methods like Random Forest or Gradient Boosting (XGBoost).
- **Operationalize the Model:** Deploy the final model via a live API or web application, allowing engineers to get real-time success probability scores for upcoming missions.

Thank you!