IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Agbaji Marcus Ifeanyi
18th September 2023

Table of content

Table of contents Contd.

Executive Summary

- **Project Overview:** This data science project focuses on predicting the successful landing of SpaceX Falcon 9 first stages. The objective was to develop a reliable model using various machine learning algorithms to determine whether the first stage of the Falcon 9 rocket would successfully land after a launch. Data for this analysis was collected through web scraping of the SpaceX website.
- **Methods and Techniques**: Several machine learning algorithms were employed to create predictive models, including Support Vector Machine (SVM), Logistic Regression, Linear Regression, and Decision Tree. These algorithms were selected for their suitability in handling classification tasks and regression analysis.
- **Key Findings**: After rigorous experimentation and evaluation, the Decision Tree algorithm emerged as the most effective model for predicting Falcon 9 first-stage landings. It demonstrated the highest accuracy and predictive power compared to other methods, making it the preferred choice for this specific problem.
- **Data collection:** Data collection involved web scraping techniques and the use of SpaceX API to gather relevant information from the SpaceX website. This process ensured that the model was trained on the most current and accurate data available, enhancing its predictive capabilities.
- **Conclusion:** The project's primary goal was to develop a predictive model capable of determining the outcome of Falcon 9 first-stage landings. It was determined through comprehensive analysis and experimentation that the Decision Tree algorithm outperformed other machine learning methods in this context. This model can be utilized to assist in assessing the likelihood of successful landings and may have practical applications in optimizing Falcon 9 missions.
- **Future Directions:** Future work may involve further fine-tuning the Decision Tree model, exploring ensemble methods, or incorporating additional features and data sources to enhance prediction accuracy. Additionally, real-time data integration and monitoring could be considered to provide timely insights into Falcon 9 missions.

Introduction

SpaceX, founded by Elon Musk, has revolutionized space exploration with its innovative technologies and ambitious missions. One critical aspect of SpaceX's operations is the reusability of rocket components, particularly the Falcon 9 first stage. The ability to successfully land and reuse this stage has significant implications for cost reduction and the sustainability of space exploration.

In this data science project, we delve into the intriguing challenge of predicting whether the Falcon 9 first stage will land successfully after launch. Leveraging the wealth of data available on the SpaceX website, we employed various machine-learning algorithms to construct predictive models for this mission-critical task.

Our exploration involved the application of Support Vector Machine (SVM), Logistic Regression, Linear Regression, and Decision Tree algorithms, each carefully chosen for their suitability in addressing classification and regression problems. Our primary objective was to develop a model that could reliably anticipate the outcome of Falcon 9 first-stage landings.

Data collection played a pivotal role in this endeavour. Through the utilization of web scraping techniques and the use of SpaceX API, we ensured that our models were trained on the most up-to-date and accurate data, providing a foundation for robust predictions.

This report presents our project's key findings, highlighting the superiority of the Decision Tree algorithm as the most effective model for predicting Falcon 9 first-stage landings. We conclude by discussing potential practical applications of this predictive capability and outlining avenues for future research and refinement, underscoring the significance of our work in advancing the goals of SpaceX and the broader field of space exploration.

Section 1

Methodology

Methodology

- **Executive Summary**

- **Data Collection Methodology:** This section elucidates the meticulous methodology employed for data collection and its transformation into a structured Pandas DataFrame.
- **Data Wrangling:** Within this segment, a comprehensive demonstration unfolds, showcasing the intricate process by which collected data was rigorously processed and standardized, rendering it amenable for further analytical exploration.
- **Exploratory Data Analysis (EDA) with Visualization and SQL:** This pivotal section unveils the multifaceted world of data exploration. Through adept utilization of SQL, the processed data is queried and harnessed for visual insights, thus laying the foundation for informed decision-making.
- **Interactive Visual Analytics with Folium and Plotly Dash:** This segment places emphasis on the dynamic realm of interactive visual analytics. Leveraging the capabilities of Folium and Plotly Dash, the presentation of insights pertaining to launch sites and mission outcomes is thoughtfully elucidated, enhancing the depth of understanding.
- **Predictive Analysis using Classification Models:** In the realm of predictive analysis, the data is systematically partitioned into training and testing datasets. Employing the precision of GridSearchCV, optimal hyperparameters are meticulously fine-tuned. Evaluation metrics encompass R2 score, Accuracy, and a comprehensive examination of the Confusion Matrix, ensuring a robust assessment of model performance

Data Collection Process

Initiating Data Retrieval: To gather data from the SpaceX API, we initiated a GET request using the appropriate endpoint URL. This request was sent to SpaceX servers to access the required information.

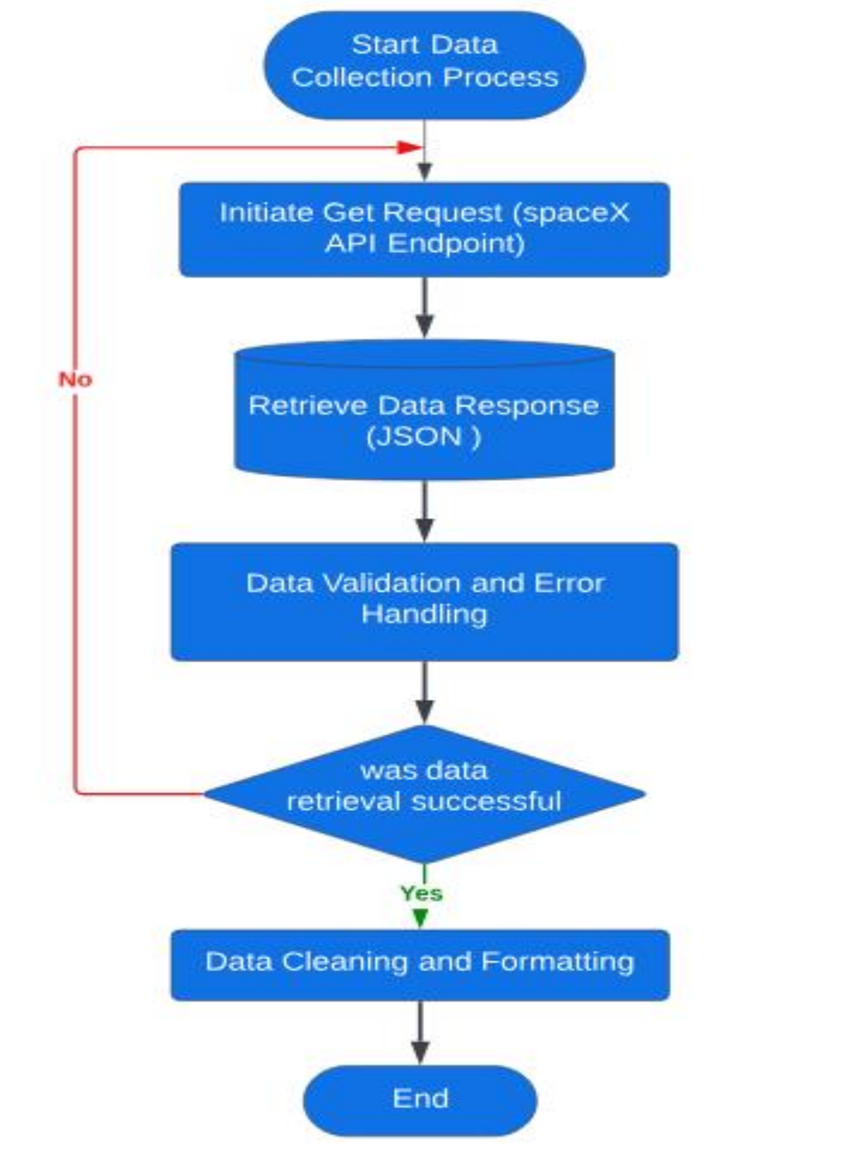
Request Parameters: We specified request parameters to filter and retrieve the specific data of interest. For example, we requested information related to Falcon 9 rocket launches and landings.

Data Retrieval: SpaceX's API responded to our GET request by providing the requested data in a structured format, typically in JSON (JavaScript Object Notation) or another machine-readable format.

Data Validation and Error Handling: We implemented error-handling mechanisms to address any issues that might arise during data retrieval. This included checking for response status codes and handling potential errors gracefully.

Data Cleaning and Formatting: Once the data was obtained, we conducted data wrangling to clean and format it. This involved:

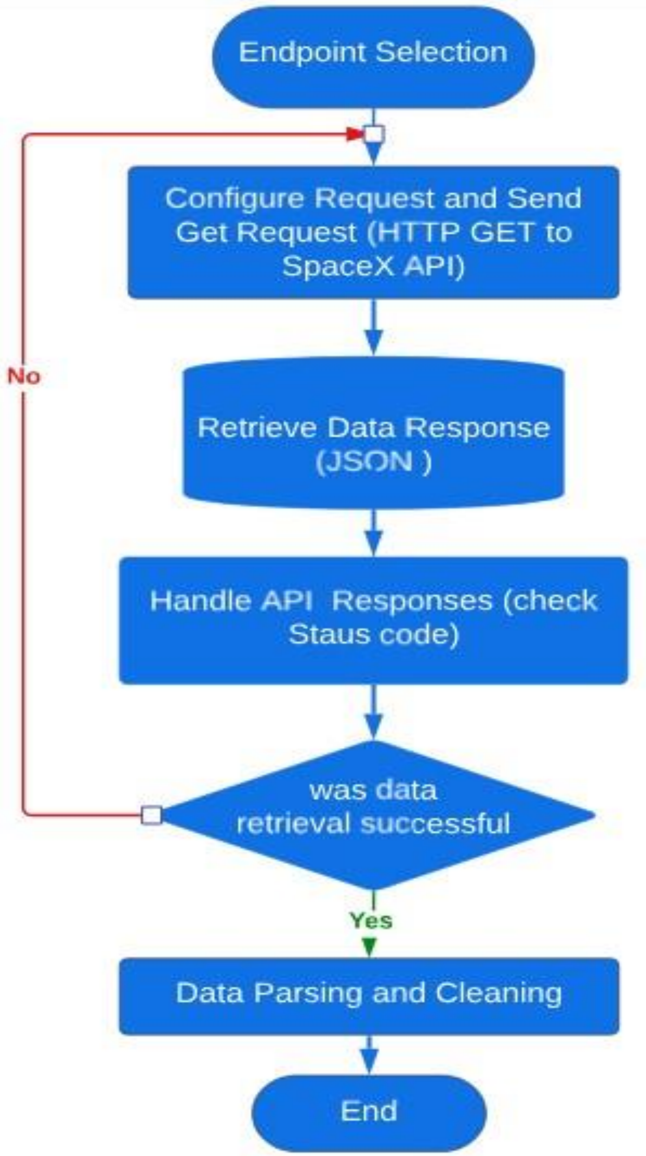
- ○ Removing duplicates, missing values, or irrelevant data points.
- ○ Renaming columns for clarity.
- ○ Converting data types if necessary.
- ○ Parsing date and time information into a standardized format.
- ○ Ensuring data consistency and accuracy.


Start Data Collection Process
Initiate Get Request (spaceX API Endpoint)
Retrieve Data Response (JSON)
Data Validation and Error Handling
was data retrieval successful
No
Yes
Data Cleaning and Formatting
End

Data Collection – SpaceX API

This Shows Data Collection Using SpaceX API

- **Step 1; Endpoint Selection:** We initiated the data collection process by selecting the appropriate REST API endpoints provided by SpaceX. These endpoints serve as access points to retrieve specific data related to SpaceX missions, rockets, and more.
- **Step 2; Request Configuration:** We configured our REST API requests by specifying request parameters, including query parameters, headers, and other relevant options. This allowed us to tailor the data retrieval to our specific needs.
- **Step 3; Sending GET Requests:** Utilizing HTTP GET requests, we sent requests to the selected SpaceX API endpoints. These GET requests were formulated with the specified parameters and were sent to the SpaceX servers for data retrieval.
- **Step 4; Handling Responses:** We implemented robust response handling mechanisms upon receiving responses from the SpaceX API. This included checking response status codes to ensure successful data retrieval.
- **Step 5; Data Parsing and Cleaning:** The data obtained from the API was typically in JSON format. We parsed this data to extract relevant information. Additionally, we conducted data cleaning by removing duplicates, handling missing values, and ensuring data consistency


Endpoint Selection
Configure Request and Send Get Request (HTTP GET to SpaceX API)
No
Retrieve Data Response (JSON)
Handle API Responses (check Staus code)
was data retrieval successful
Yes
Data Parsing and Cleaning
End

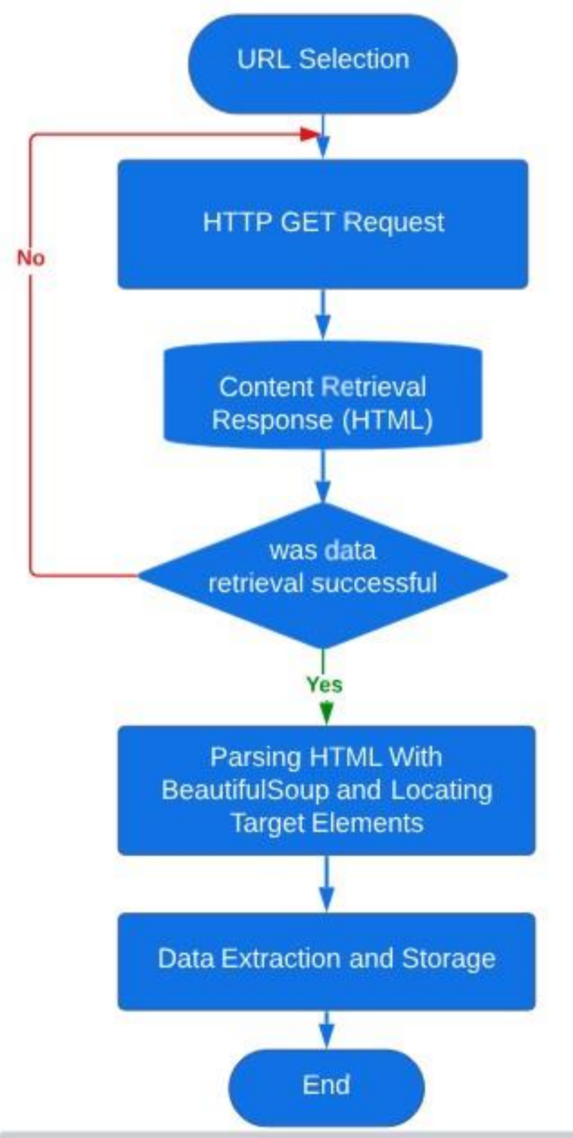
GitHub notebook for the data collection: https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Introduction: Web scraping is a technique used to extract structured data from websites. Below, we describe the key steps of our web scraping process using BeautifulSoup, a Python library for parsing HTML and XML documents.

- **Step 1: URL Selection**; our Journey began with the careful selection of the target webpage's URL. This URL directed us to the specific web page containing the data we aimed to retrieve.
- **Step 2: HTTP GET Request**; to access the webpage's content, we initiated the HTTP GET request to the chosen URL. This request allowed us to retrieve the HTML content of the webpage.
- **Step 3: HTML Content Retrieval;** with the get request successfully executed, we retrieved the HTML content by the web server. This HTML content formed the structural basis for our data collection efforts.
- **Step 4: Parsing HTML with BeautifulSoup**; we used the BeautifulSoup library to parse the HTML content. This involves creating a BeautifulSoup object and specifying the parser to be used (e.g., 'html.parser').
- **Step 5: Locate Target Elements;** A crucial step in our process was to identify and locate the HTML elements containing the data we intended to scrape. To achieve this, we employed various methods provided by BeautifulSoup, including searching by tag names, attributes, or classes.
- **Step 6: Data Extraction;** we pinpointed the target HTML elements, and we proceeded to extract the desired data. This involved accessing the text, attributes, and other properties of these elements, enabling us to retrieve the information of interest.
- **Step 7**: Data Storage: The data was processed and stored as a CSV file for further analysis.

GitHub URL of the completed web scraping notebook:
https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/jupyter-labs-webscraping.ipynb


URL Selection
HTTP GET Request
No
Content Retrieval Response (HTML)
was data retrieval successful
Yes
Parsing HTML With BeautifulSoup and Locating Target Elements
Data Extraction and Storage
End

Data Wrangling

- **Step 1: Data Loading;** Our data wrangling journey commenced with the loading of the scraped dataset. We used suitable Python libraries, such as pandas to read and import the data into our working environment.
- **Step 2: Data Inspection**; Initially examine the dataset to understand its structure and characteristics. Checked for the number of columns, data types, and presence of missing values.
- **Step 3: Data Cleaning**; Addressed data quality issues:
  - Removed duplicate records to ensure data integrity.
  - Handled missing values by imputing them, based on the specific column context.
  - Corrected inconsistencies in data entries, such as typos or formatting errors.
- **Step 4: Data Transformation**; Calculated the number of launches on each launch site by grouping the data based on the launch site information and computing the number and occurrence of each orbit type:
  - Grouped the data by orbit type.
  - Counted the occurrences of each orbit type.
  - Analysed mission outcomes per orbit type:
  - Counted the occurrences of each mission outcome within each orbit type.
  - Created a landing outcome label from the "Outcome" column by categorizing outcomes into specific labels (e.g., 'Successful', 'Failed').

GitHub URL for the completed data wrangling: https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

The following plots were done to find the insight hidden the available data:

1. Catplot: Payload Mass vs. Flight Number

- Purpose: This catplot serves as a vital tool for visualizing the evolution of payload mass over multiple SpaceX missions.
- Significance: By monitoring payload mass trends over time, we gain valuable insights into how payload capacity has evolved, which is critical for optimizing mission planning and logistics.

2. Catplot: Launch Site vs. Flight Number

- Purpose: The catplot depicting Launch Site vs. Flight Number reveals the choice of launch sites for different SpaceX missions.
- Significance: Understanding the historical preferences in launch site selection enables us to identify patterns and insights in SpaceX's mission planning and logistical strategies.

3. Bar Plot: Categorical Data

- Purpose: This bar plot succinctly summarizes key categorical data points, such as orbital type, by other relevant features.
- Significance: Bar plots provide a clear and concise overview of mission data, allowing for rapid identification of patterns and trends across various mission attributes.

EDA with Data Visualization

4. Scatterplot: Payload Mass vs. Launch Site

- ○ Purpose: Analysing payload mass variations in relation to launch site is critical for optimizing mission planning.
- ○ Significance: This scatterplot offers insights into how payload capacity varies across different launch sites, aiding in informed decisions related to payload selection and mission logistics.

5. Scatterplot with Hue (Class): Payload Mass vs. Orbit

- ○ the 'Class' variable (success/failure), we can identify patterns that shed light on the influence of payload mass and orbit on mission success, offering valuable insights into mission planning. Purpose: This visualization explores the relationships between payload mass, orbit, and mission outcomes.
- ○ Significance: By incorporating

6. Line Plot: Mission Outcomes Over Time

- ○ Purpose: This line plot chronicles SpaceX missions over time, categorizing them as either successes or failures.
- ○ Significance: Tracking mission outcomes across years allows us to gauge historical performance, recognize trends, and evaluate mission reliability, which is essential for data-driven decision-making and performance assessment.

These visualizations represent valuable tools for interpreting SpaceX's mission data, enabling data-driven decision-making and enhancing our understanding of historical performance and mission planning strategies.

GitHub URL for the completed EDA:

https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

To perform Exploratory Data Analysis with SQL the following query was used:

- **Display Unique Launch Sites:**

  Query: select distinct("Landing_Outcome") from SPACEXTABLE

- **Display Launch Site Starting with 'CCA’:**

  Query: select * from SPACEXTABLE WHERE LAUNCH_SITE LIKE '%CCA%' LIMIT 5

- **Total Payload Mass by NASA (CRS):**

  Query: select count(PAYLOAD_MASS__KG_) AS 'TOTAL_NASA_PAYLOAD’ FROM SPACEXTABLE WHERE customer like '%NASA%

- **Average Payload Mass for Booster Version F9 v1.1:**

  Query: select AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1’

- **Data of the First Successful Landing on Ground Pad:**

  Query: select min(Date) from SPACEXTABLE WHERE LANDING_OUTCOME LIKE 'SUCCESS%’

- **Boosters with Success in Drone ship and Payload Mass Ranges:**

  Query: select Booster_Version FROM SPACEXTABLE WHERE (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)AND LANDING_OUTCOME LIKE 'SUCCESS (DRONE%'

EDA with SQL

- **Total Number of Successful and Failure Mission Outcomes:**

  Query: SELECT COUNT(CASE WHEN landing_outcome like '%success%' THEN 1 ELSE NULL END) as success_count,
  COUNT(CASE WHEN landing_outcome like '%failure%' THEN 1 ELSE NULL END) as failure_count FROM SPACEXTABLE

- **Booster Versions with Maximum Payload Mass:**

  Query: SELECT Booster_version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
  WHERE payload_mass__kg_ = (select min(payload_mass__kg_) from SPACEXTABLE)

- **Records for Months Names, Failure(Drone Ship), Booster Versions and Launch Sites in 2015:**

  Query: SELECT SUBSTRING(Date,1,4) AS YEAR, landing_outcome, Booster_version, launch_site
  FROM SPACEXTABLE WHERE LANDING_OUTCOME LIKE '%FAILURE (D%’ AND YEAR = '2015’

- **Ranking Landing Outcomes between Specific Dates:**

  Query: SELECT landing_outcome, COUNT(landing_outcome) as Outcome_count,
  RANK() OVER (ORDER BY COUNT(landing_outcome) DESC) as Outcome_rank FROM SPACEXTABLE
  WHERE date >= '2010-06-04' AND date <= '2017-03-20’ GROUP BY landing_outcome ORDER BY Outcome_count DESC;

GitHub URL for the completed EDA with SQL:
https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

The following Map objects where used in the creation of the interactive map with Folium

- **Map Circle:**
  - **Purpose:** Created circles around various launch sites and NASA JSC.
  - **Significance:** Circles visually represent specific areas of interest, helping to define launch site boundaries and highlight the location of NASA JSC
- **Map Marker:**
  - **Purpose:** Placed markers at specific geographic locations.
  - **Significance:** Markers serve as visual indicators, pinpointing precise locations on the map and enhancing the user's ability to identify key points of interest.
- **MarkerCluster:**
  - **Purpose:** Clustered and marked launch sites based on mission outcomes (success/failure), using green and red markers respectively.
  - **Significance:** Clustering markers simplifies the visualization of mission outcomes, providing a clear overview of the distribution of successful and failed launches.
- **MousePosition:**
  - **Purpose**: Utilized the MousePosition object to display coordinates when hovering the mouse over a point on the map.
  - **Significance:** This feature enhances user experience by allowing quick access to geographic coordinates, facilitating precise location identification.
- **Folium Polyline:**
  - **Purpose:** Drew lines from launch sites to coastline points.
  - **Significance:** The polyline visually represents the path from launch sites to the coastline, providing context and helping users understand the trajectory of missions

GitHub URL for the completed interactive map:
https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

Summary of Plots/Graphs and Interactions:

1. **Pie Chart (Success Rate by Year):**
   - **Plot:** A pie chart displaying the success rate of all launch sites based on the inputted year.
   - **Interaction:** Users can select a specific year using an input or dropdown, and the pie chart dynamically updates.
2. **Scatter Plot (Success vs. Failure by Payload Range):**
   - **Plot:** A scatter plot illustrating the distribution of success and failure cases based on the inputted payload range and slide position.
   - **Interaction:** Users can interactively adjust the payload range and slide position, and the scatter plot updates in real-time.

Explanation of Plots/Graphs and Interactions:

1. **Pie Chart (Success Rate by Year):**
   - **Purpose:** The pie chart provides an overview of the success rates of all launch sites for a specific year.
   - **Significance:** It helps users quickly identify which launch sites had the highest success rates in a given year, aiding in historical performance assessment and decision-making for mission planning.
2. **Scatter Plot (Success vs. Failure by Payload Range):**
   - **Purpose:** The scatter plot offers a visual representation of success and failure cases based on payload range and slide position.
   - **Significance:** This interactive plot enables users to explore the relationship between payload range, slide position, and mission outcomes. It supports data-driven decision-making by allowing users to identify trends and correlations that may influence mission success or failure.

By incorporating these plots and interactions into the dashboard, we enhance user engagement and data exploration. Users can make informed decisions, analyse historical performance, and gain valuable insights into the factors affecting mission success within the context of launch sites and payload characteristics.

GitHub URL for the Completed Python File of Plotly Dash Lab:
https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Python_files/dash_final_project.py

Predictive Analysis (Classification)

Summary of Model Development Process:

Step 1: Data Preparation

- Load the dataset into the environment.
- Split the dataset into training and testing data sets.

Step 2: Model Selection and Parameter Tuning

- Choose algorithms: Logistic Regression, SVM, Decision Tree, and KNN.
- Define and set initial hyperparameters for each model.
- Create model objects for each selected algorithm.

Step 3: Cross-validation with Hyperparameter Tuning

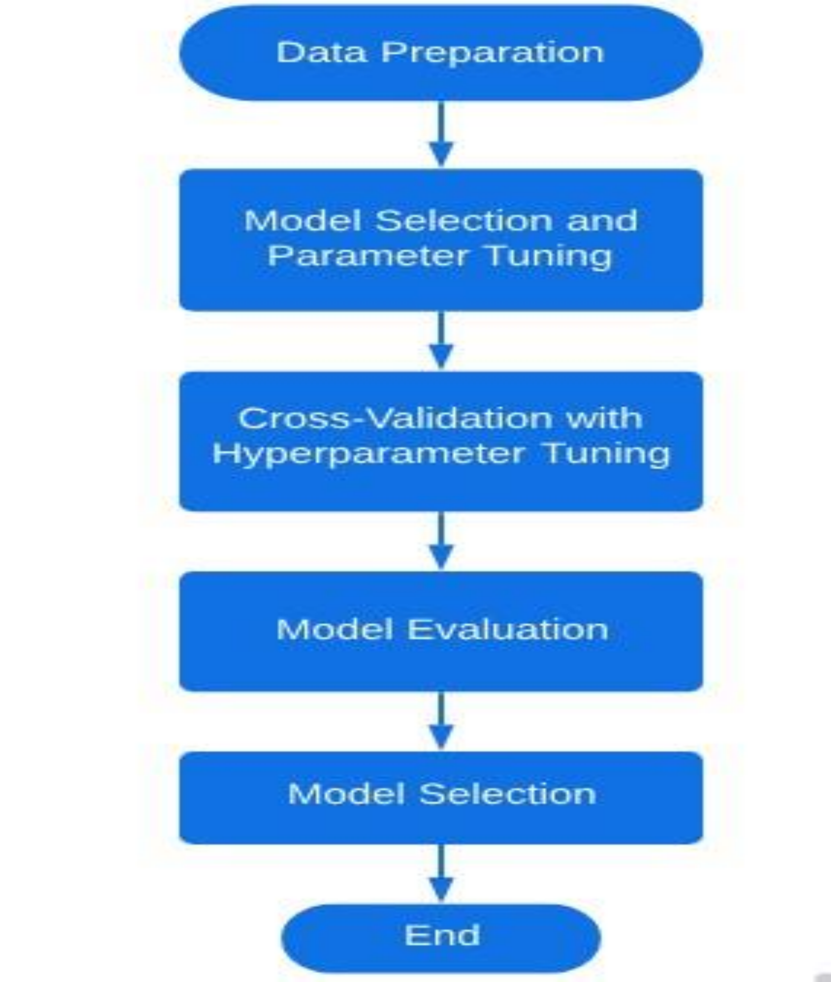
- Create a GridSearchCV object with 10-fold cross-validation for each model.
- Fit the GridSearchCV object to find the best hyperparameters for each model.

Step 4: Model Evaluation

- Evaluate each model on the testing dataset.
- Obtain and plot confusion matrices for model performance assessment.

Step 5: Model Selection

- Calculate the scores for each model.
- Choose the model with the highest score as the best-performing model.


Data Preparation
Model Selection and Parameter Tuning
Cross-Validation with Hyperparameter Tuning
Model Evaluation
Model Selection
End

GitHub URL of the completed predictive analysis:
https://github.com/MarcusIfeanyi/Data_science_Coursera/blob/main/Notebooks/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1)-checkpoint.ipynb

Results

Results are analyzed from the following

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

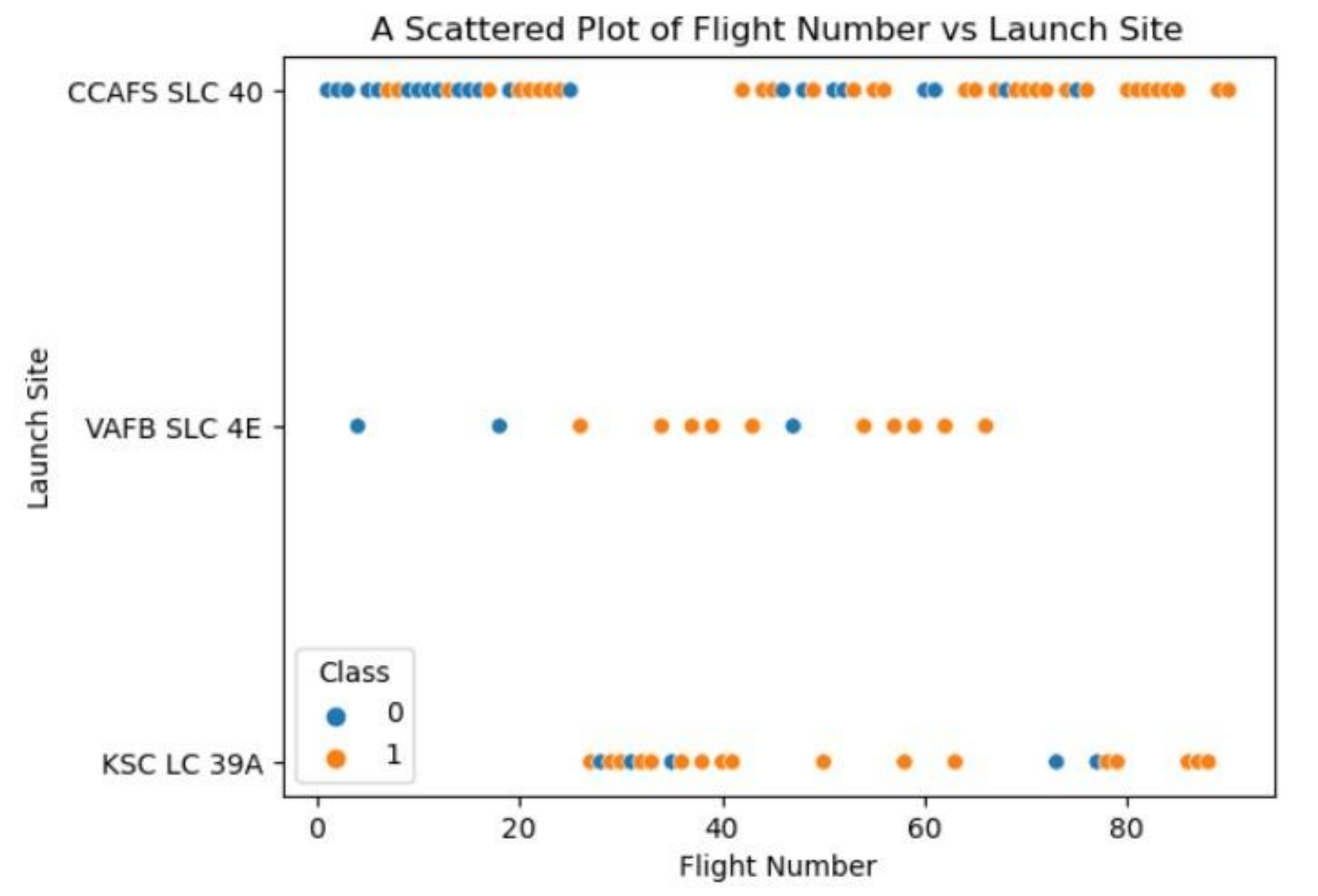
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

OBSERVATIONS AND EXPLANATIONS

In the graph analysis, it is discerned that Launch Site CCADS SLC 40 exhibits the highest frequency of flight occurrences. The delineation between the success of launched Rocket first stage landing, represented in orange, and failure, depicted in blue, is distinctly visualized. Notably, upon visual inspection of the graph, it is indicative that Launch Site KSC LC 39A demonstrates a comparatively elevated success rate in contrast to the other launch sites. Furthermore, there is a discernible increment in the success rate within Launch Site CCAFS SLC 40, which aligns with an observable rise in flight numbers above 80.

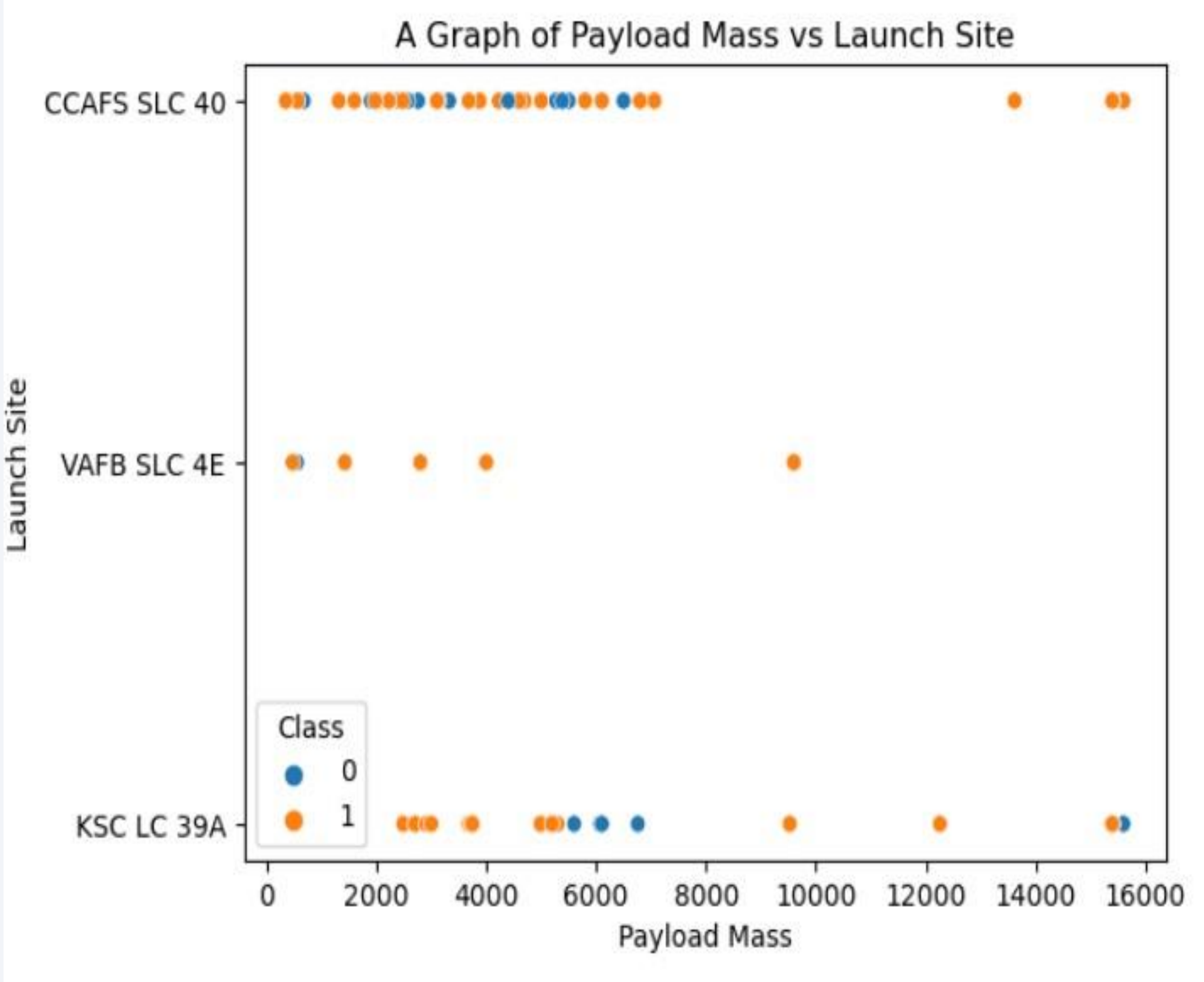


Payload vs. Launch Site

OBSERVATION AND EXPLANATION

In the course of our graph analysis, a discernible pattern emerges. It is observed that, with a lower payload mass, Launch Site KSC LC 39 exhibits a notably higher rate of success of launched Rocket first stage landing, distinctly highlighted in orange, in comparison to other launch sites. Simultaneously, failures are visually depicted in blue.

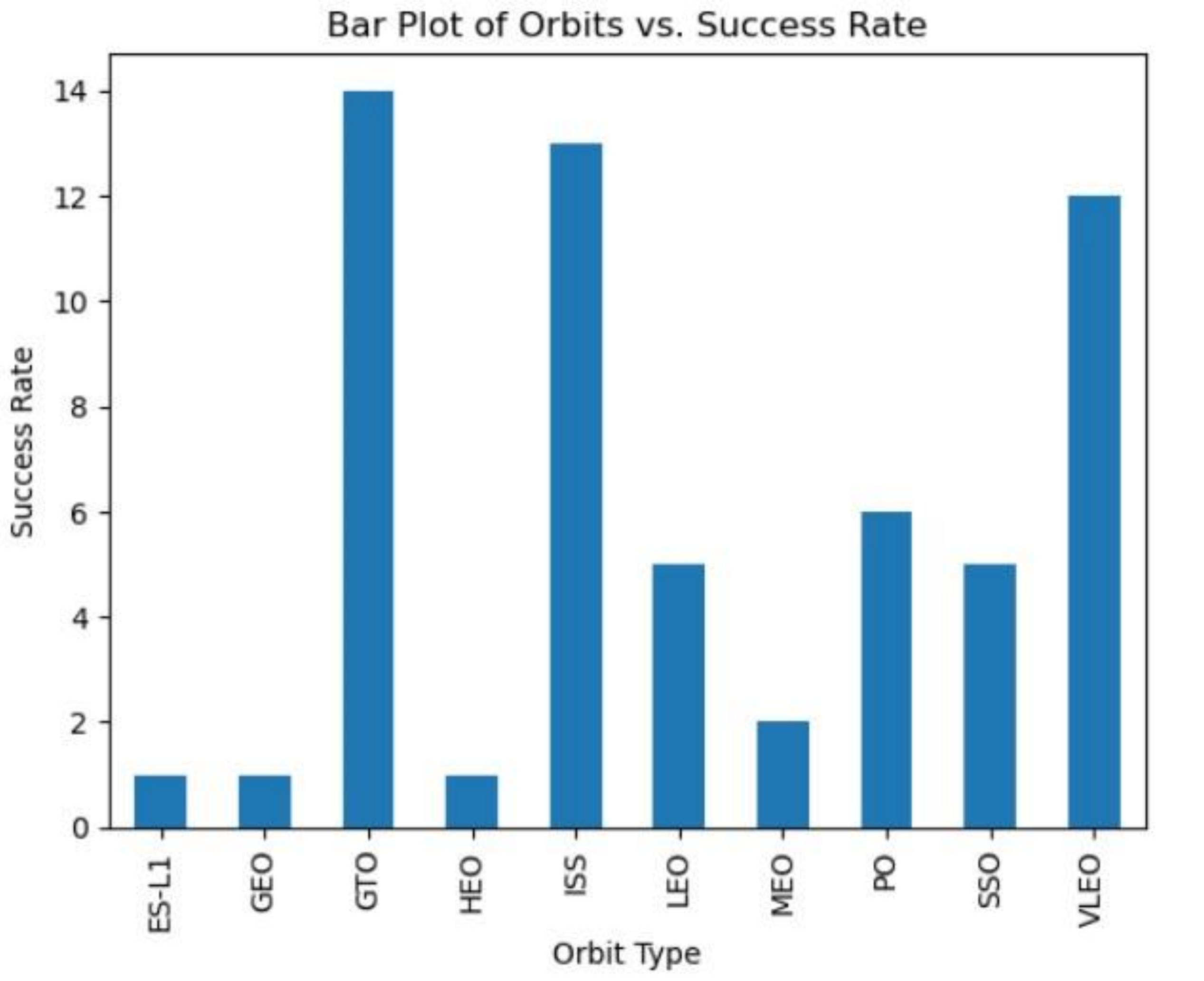
Remarkably, Site VAFB SLC 4E boasts the highest success rate as the payload gradually increases, peaking at payloads up to 10,000 kilograms. Beyond this threshold, a shift is evident, with Launch Site CCADS SLC 40 demonstrating the most favourable success rates for payloads exceeding 10,000 kilograms.



Success Rate vs. Orbit Type

OBSERVATION AND EXPLANATION

This bar chart provides a comprehensive overview of orbital destinations alongside the respective success rates pertaining to first-stage landing pad recoveries. Our analysis reveals that missions targeting Geostationary Transfer Orbit (GTO) exhibit the highest success rate in successfully recovering their first stage, followed closely by those destined for the International Space Station (ISS) orbit and Very Low Earth Orbit (VLEO). Conversely, missions to Earth-Moon Lagrange Point 1 (ES-L1), Geosynchronous Equatorial Orbit (GEO), and High Earth Orbit (HEO) exhibit relatively lower rates of first-stage landing success

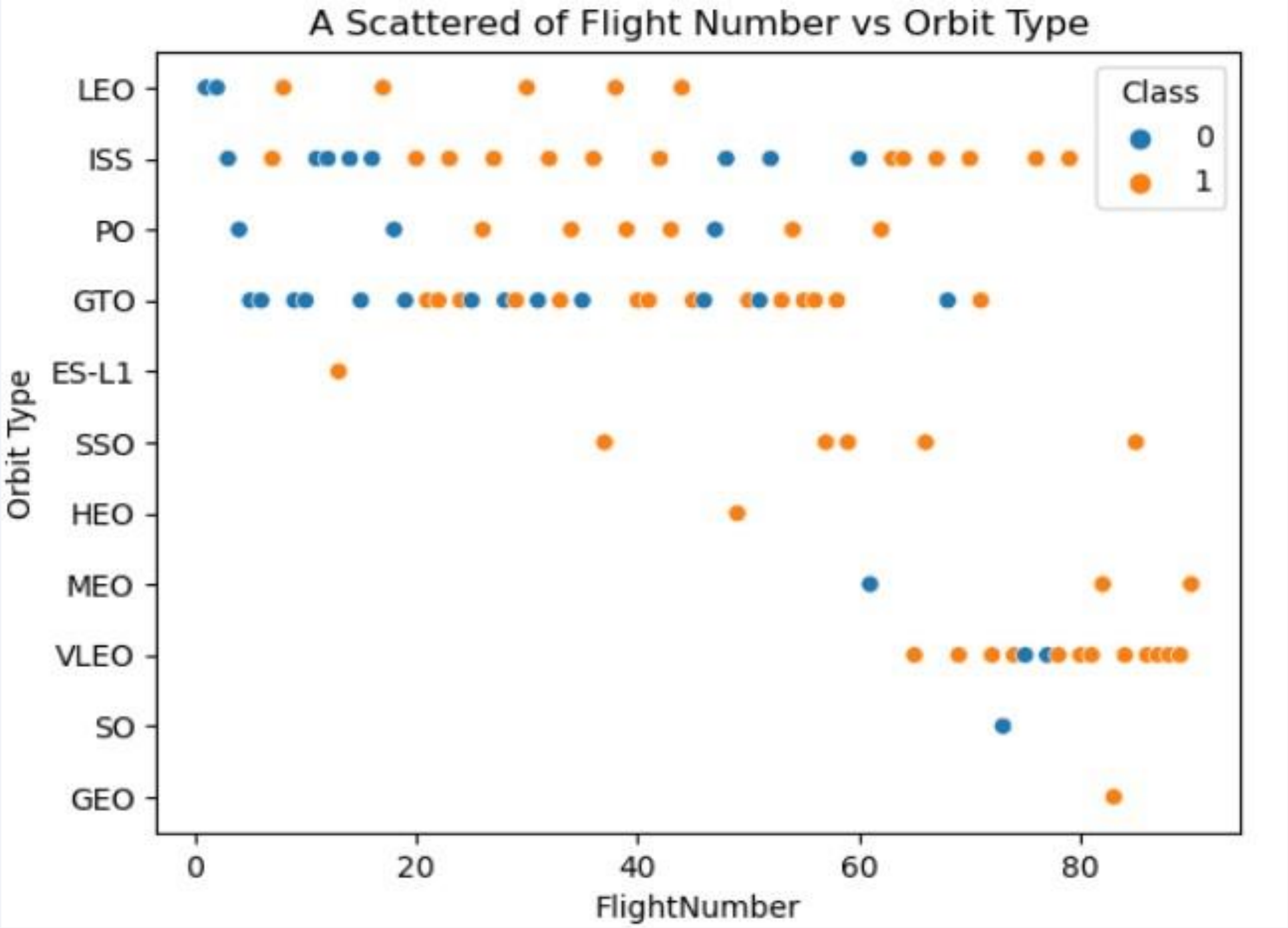


Flight Number vs. Orbit Type

OBSERVATION AND EXPLANATION

Through careful observation and analysis, it becomes apparent that missions directed toward achieving Sun-Synchronous Orbit (SSO) exhibit an exceptional track record of success of the launched Rocket first stage landing, as illustrated prominently in the visual representation through orange markings. Notably, no instances of failure, denoted by the absence of blue markings, have been recorded for this orbital destination.

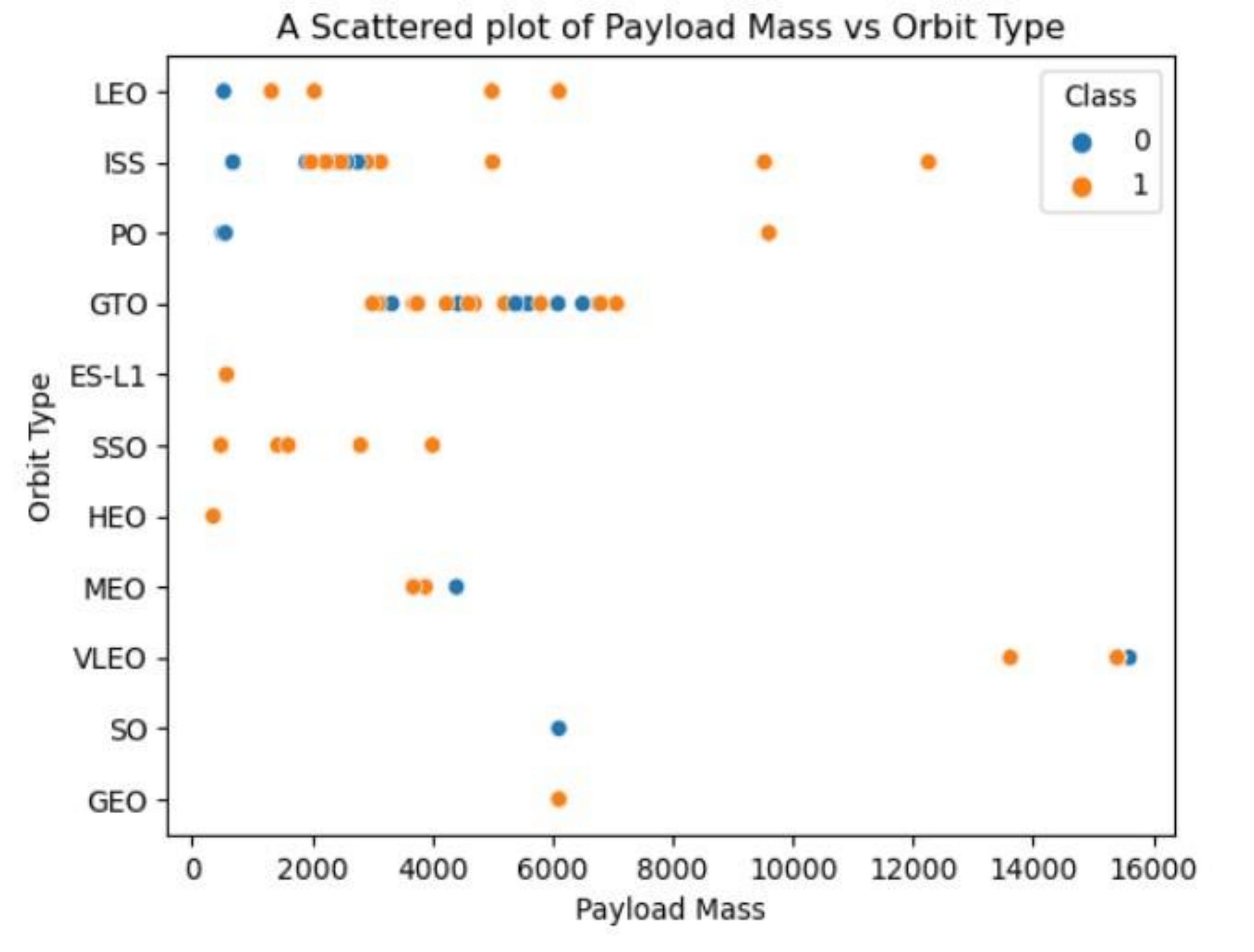
In a similar vein, for missions intended to reach Low Earth Orbit (LEO), there is a discernible increase in success rates beyond a flight number threshold of 10, with the trend continuing upwards. However, it is essential to highlight that this particular success trend is more pronounced for missions targeting Very Low Earth Orbit (VLEO), as success rates become significantly more substantial with an increase in flight numbers, particularly surpassing a threshold of 78 flights to VLEO orbits.



Flight Number vs. Orbit Type

EXPLANATIONS AND OBSERVATION

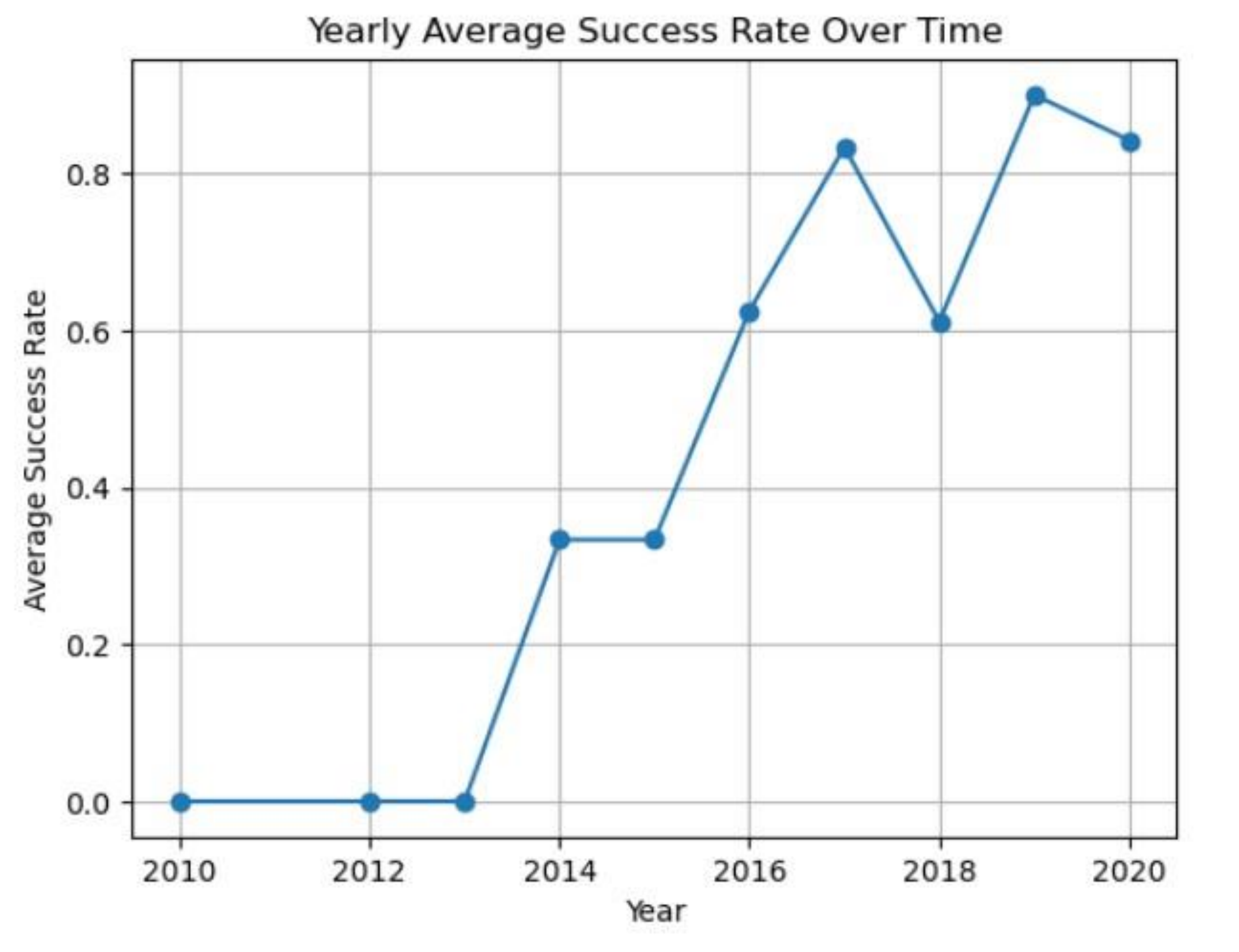
This graphical representation presents a scatter plot depicting the relationship between payload mass and orbit type. Upon careful analysis, a noteworthy observation comes to the forefront: In the context of first-stage landing success, the Sun-Synchronous Orbit (SSO) stands out with a remarkable 100% success rate for payloads below 4000 kilograms. Conversely, the International Space Station (ISS) exhibits superior first-stage landing success rates for payloads exceeding 4000 kilograms



Launch Success Yearly Trend

EXPLANATIONS AND OBSERVATION

This graphical representation provides an insightful perspective on the Yearly Average Success Rate over Time. Upon meticulous analysis, a distinctive trend emerges. In the period spanning from 2010 to 2012, the average success rate remains consistently at 0. However, a notable inflection point occurs in 2013, marking a significant and sustained ascent in the success rate. This upward trajectory persists, experiencing a minor dip in 2018, before culminating in its most substantial increase in the year 2019.



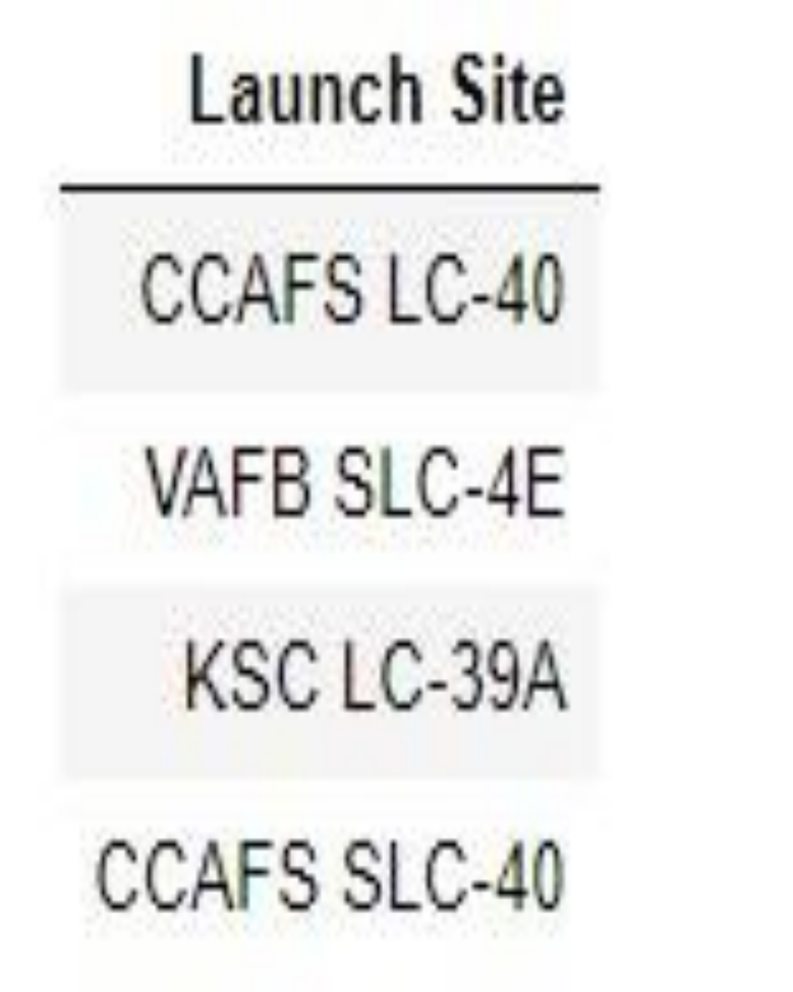
All Launch Site Names

EXPLANATION

The displayed image presents the query results derived from the execution of the SQL statement

Query: `SELECT DISTINCT(Launch_site) as 'Launch Site’
FROM SPACEXTABLE`.

This query effectively showcases all the distinct launch sites utilized by SpaceX. Employing the **'DISTINCT'** statement, the query isolates and reveals the unique attributes within the 'Launch Site' column. This data holds significance as it provides a comprehensive inventory of launch site locations, facilitating a deeper understanding of SpaceX's operational footprint and aiding in strategic decision-making.



Launch Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Query Result:

The query executed is as follows:

- SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE '%CCA%' LIMIT 5

This query returns a limited set of records (up to 5) from the "**SPACEXTABLE"** where the "**LAUNCH_SITE**" column contains the string "**CCA"** anywhere in its value.

Explanation:

The purpose of this query is to retrieve a subset of data from the "**SPACEXTABLE**" related to SpaceX launches. Specifically, it focuses on records where the launch site name contains the substring "**CCA."** The use of the `**LIKE**` operator with the `%` wildcard characters allows for a flexible search, matching any launch site name that includes "**CCA**" regardless of its position within the name.

The inclusion of the `**LIMIT 5**` clause ensures that the query returns only the first 5 matching records, providing a concise and manageable dataset for analysis or exploration.

Total Payload Mass

Query Result:

The query executed is as follows:

Query:

```
SELECT COUNT(PAYLOAD_MASS__KG_) AS 'TOTAL_NASA_PAYLOAD’
  FROM SPACEXTABLE
  WHERE customer LIKE '%NASA%';
```

Explanation:

This query serves to calculate and retrieve specific information from the "SPACEXTABLE" database table, focusing on payloads associated with NASA as the customer. Here's a breakdown of the query components:

- **`SELECT COUNT(PAYLOAD_MASS__KG_) AS 'TOTAL_NASA_PAYLOAD'`:** In this part of the query, we are using the `COUNT` function to tally the number of rows where the customer name contains the substring "NASA." The result of this count is aliased as 'TOTAL_NASA_PAYLOAD' for clarity.
- **`FROM SPACEXTABLE`: Specifies the source table, which is "SPACEXTABLE"** in this case.
- `WHERE customer LIKE '%NASA%'`: This condition filters the records to include only those where the 'customer' column contains the string "NASA" anywhere within its value. The `%` wildcard characters in the `LIKE` clause allow for flexible matching.

The query will return a single value, which represents the total count of payloads associated with NASA as the customer in the dataset.

Average Payload Mass by F9 v1.1

Query Result:

The query executed is as follows:

Query:

```
SELECT AVG(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
    WHERE Booster_Version LIKE 'F9 v1.1';
```

Explanation:

This query is designed to calculate and retrieve a specific piece of information from the **"SPACEXTABLE"** database table, focusing on the average payload mass for launches associated with the booster version **"F9 v1.1."** Here's a breakdown of the query components:

- **`SELECT AVG(PAYLOAD_MASS__KG_)`:** In this part of the query, we are using the **`AVG`** function to calculate the average payload mass (in kilograms) for all records that match the specified criteria.
- **`FROM SPACEXTABLE`:** Specifies the source table, which is **"SPACEXTABLE"** in this case.
- **`WHERE Booster_Version LIKE 'F9 v1.1'`:** This condition filters the records to include only those where the 'Booster_Version' column exactly matches the value **'F9 v1.1'.** The **`LIKE`** clause is used with an exact match (no wildcards) in this case.

The query will return a single numeric value, which represents the average payload mass for launches associated with the specified booster version.

First Successful Ground Landing Date

Query Result:

The following SQL query was executed:

Query:
```
SELECT MIN(Date)
    FROM SPACEXTABLE
    WHERE LANDING_OUTCOME LIKE 'SUCCESS%';
```

Explanation:

This SQL query is designed to retrieve the minimum (earliest) date from the "SPACEXTABLE" database table for records where the 'LANDING_OUTCOME' column begins with the string 'SUCCESS'. Here's a breakdown of the query components:

- **`SELECT MIN(Date)`:** In this part of the query, the `MIN` function is used to find the minimum date value from the 'Date' column.
- **`FROM SPACEXTABLE`:** Specifies the source table as "SPACEXTABLE".
- **`WHERE LANDING_OUTCOME LIKE 'SUCCESS%'`:** This condition filters the records to include only those where the 'LANDING_OUTCOME' column starts with the word 'SUCCESS'. The `%` wildcard character is used to match any text that follows the word 'SUCCESS'.

The query will return a single date value, representing the earliest date on which a successful landing outcome occurred in the dataset.

Successful Drone Ship Landing with Payload between 4000 and 6000

Query Result:

The query executed is as follows:

Query: SELECT Booster_Version

FROM SPACEXTABLE

WHERE (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)

AND LANDING_OUTCOME LIKE 'SUCCESS (DRONE%’;

Explanation:

This query returns a list of booster versions from the "SPACEXTABLE" database table where two conditions are met:

1. **`(PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)`:** This condition filters the records to include only those with a payload mass between 4000 and 6000 kilograms.
2. **`LANDING_OUTCOME LIKE 'SUCCESS (DRONE%'`:** This condition further refines the selection by including only records where the 'LANDING_OUTCOME' column starts with the string 'SUCCESS (DRONE'.

The combined effect of these conditions is that the query returns a list of booster versions associated with successful drone ship landings for payloads in the specified mass range.

The query will return a single date value, representing the earliest date on which a successful landing outcome occurred in the dataset.

Total Number of Successful and Failure Mission Outcomes

Query Result:

The query executed is as follows:

```
SELECT
  COUNT(CASE WHEN landing_outcome like '%success%' THEN 1 ELSE NULL END) as success_count,
  COUNT(CASE WHEN landing_outcome like '%failure%' THEN 1 ELSE NULL END) as failure_count
FROM SPACEXTABLE;
```

Explanation:

This query is designed to calculate and return two distinct counts based on specific conditions applied to the 'landing_outcome' column in the "SPACEXTABLE" database table.

- **`COUNT(CASE WHEN landing_outcome like '%success%' THEN 1 ELSE NULL END) as success_count`:** This part of the query counts the number of records where the 'landing_outcome' column contains the string 'success' (case-insensitive). It uses a `CASE` expression to assign a value of 1 for each record that matches the condition, and then counts the total number of 1s. The result is aliased as 'success_count.'
- **`COUNT(CASE WHEN landing_outcome like '%failure%' THEN 1 ELSE NULL END) as failure_count`:** Similarly, this part of the query counts the number of records where the 'landing_outcome' column contains the string 'failure' (case-insensitive). It uses a `CASE` expression to assign a value of 1 for each record that matches the condition, and then counts the total number of 1s. The result is aliased as 'failure_count.'

The query returns two values: 'success_count' representing the count of successful landings and 'failure_count' representing the count of failed landings in the dataset.

Boosters Carried Maximum Payload

Query Result:

The query executed is as follows:

```
    SELECT Booster_version, PAYLOAD_MASS__KG_
    FROM SPACEXTABLE
    WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTABLE);
```

Explanation:

This query aims to retrieve specific information from the "SPACEXTABLE" database table based on two key components:

- **`SELECT Booster_version, PAYLOAD_MASS__KG_`:** This part of the query specifies the columns that will be included in the result. It selects the 'Booster_version' and 'PAYLOAD_MASS__KG_' columns.
- **`FROM SPACEXTABLE`:** Specifies the source table as "SPACEXTABLE."
- **`WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTABLE)`:** The condition in this part of the query filters the records to include only those where the 'payload_mass__kg_' column is equal to the maximum payload mass found in the same table. It accomplishes this by using a subquery `(SELECT MAX(payload_mass__kg_) FROM SPACEXTABLE)` to determine the maximum payload mass in the entire dataset.

The result of the query will provide the 'Booster_version' and 'PAYLOAD_MASS__KG_' values associated with the record(s) that have the maximum payload mass in the dataset.

2015 Launch Records

Query Result:

The query executed is as follows:

```
SELECT SUBSTRING(Date,1,4) AS YEAR, landing_outcome, Booster_version, launch_site
        FROM SPACEXTABLE
        WHERE LANDING_OUTCOME LIKE '%FAILURE (D%'
        AND YEAR = '2015’;
```

Explanation:

This query is designed to retrieve specific information from the "SPACEXTABLE" database table based on several conditions:

- **`SELECT SUBSTRING(Date,1,4) AS YEAR, landing_outcome, Booster_version, launch_site`:** This part of the query selects and renames columns for the result. It extracts the first four characters of the 'Date' column as 'YEAR' and includes columns for 'landing_outcome,' 'Booster_version,' and 'launch_site' in the output.
- **`FROM SPACEXTABLE`:** Specifies the source table as "SPACEXTABLE."
- **`WHERE LANDING_OUTCOME LIKE '%FAILURE (D%'`:** This condition filters the records to include only those where the 'LANDING_OUTCOME' column contains the substring 'FAILURE (D%' (case-insensitive).
- **`AND YEAR = '2015'`:** This condition further refines the selection by including only records where the 'YEAR' column (derived from the 'Date' column) matches the value '2015'.

The result of the query will provide a subset of records from the "SPACEXTABLE" table, showing the year, landing outcome, booster version, and launch site for SpaceX launches that match the specified criteria in 2015

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

Query Result:

The query executed is as follows:

```
SELECT
   landing_outcome, COUNT(landing_outcome) as Outcome_count,
   RANK() OVER (ORDER BY COUNT(landing_outcome) DESC) as Outcome_rank
FROM SPACEXTABLE WHERE date >= '2010-06-04' AND date <= '2017-03-20’
    GROUP BY landing_outcome ORDER BY Outcome_count DESC;
```

Explanation:

This SQL query is intended to analyse and rank the outcomes of SpaceX landings within a specific date range, which is from June 4, 2010, to March 20, 2017. Here's a breakdown of the query components:

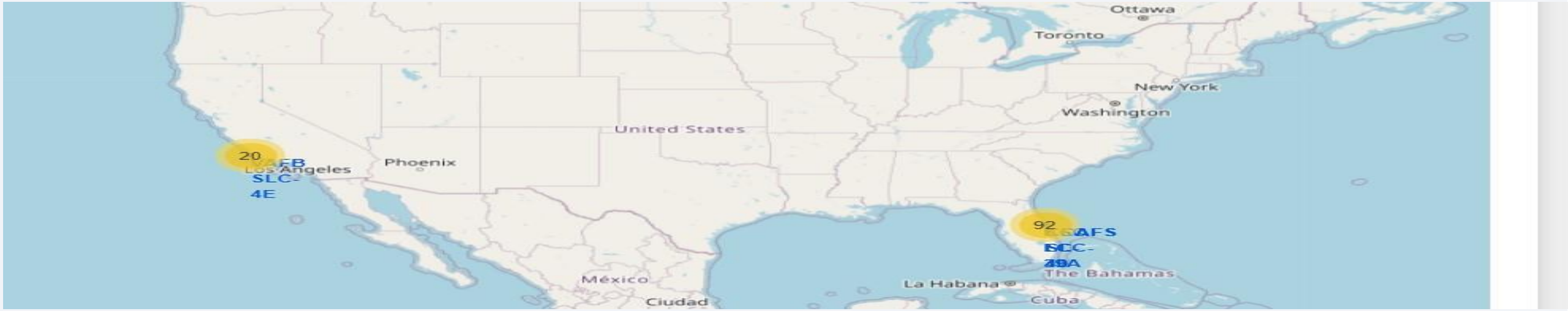
- **`SELECT landing_outcome, COUNT(landing_outcome) as Outcome_count`:** In this part of the query, it selects the 'landing_outcome' column and calculates the count of each unique landing outcome within the specified date range. The result is aliased as 'Outcome_count' for clarity.
- **`RANK() OVER (ORDER BY COUNT(landing_outcome) DESC) as Outcome_rank`:** This part of the query utilizes the `RANK()` window function to rank the landing outcomes based on the count in descending order (`DESC`). The result is aliased as 'Outcome_rank,' indicating the rank of each outcome based on its frequency.
- **`FROM SPACEXTABLE`:** Specifies the source table as "SPACEXTABLE."
- **`WHERE date >= '2010-06-04' AND date <= '2017-03-20'`:** This condition filters the records to include only those with a 'date' within the specified date range.
- **`GROUP BY landing_outcome`:** Groups the records by the 'landing_outcome' column to perform the count and ranking for each unique landing outcome.
- **`ORDER BY Outcome_count DESC`:** Orders the result set by 'Outcome_count' in descending order, ensuring that the landing outcomes with the highest counts are listed first.

The query returns a result set that includes the landing outcomes, their respective counts, and their ranks based on the count within the specified date range.

Section 3

Launch Sites Proximities Analysis

Launch Site Locations Analysis with Folium



The following are the elements and potential findings driven by the above folium map:

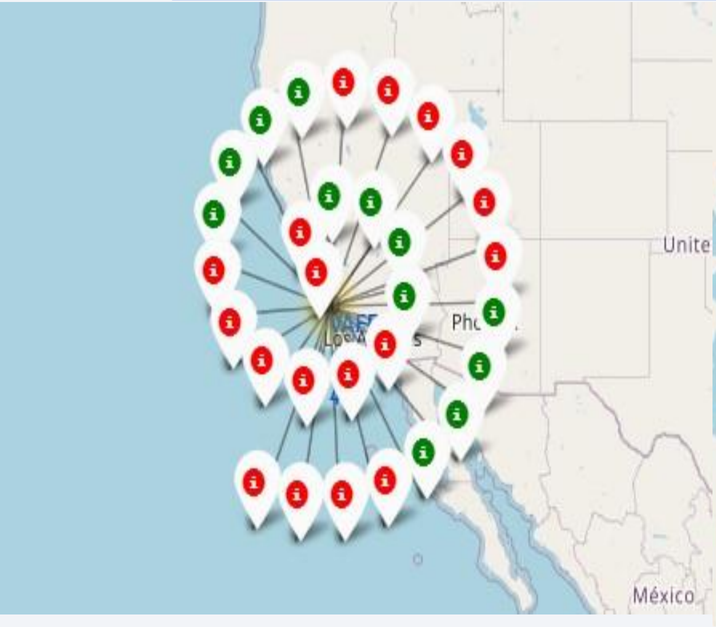
- **Launch Site Markers:** Each launch site would be represented by a marker on the map, typically pinpointing the exact location of the launch facility.
- **Clustered Markers:** To improve the map's readability, markers can be clustered together when they are close to each other.
- **Pop-up Information**: Clicking on a marker can trigger a pop-up window with additional information about the launch site. This information might include the launch site's name, coordinates, historical significance, and notable missions.

Potential Findings:

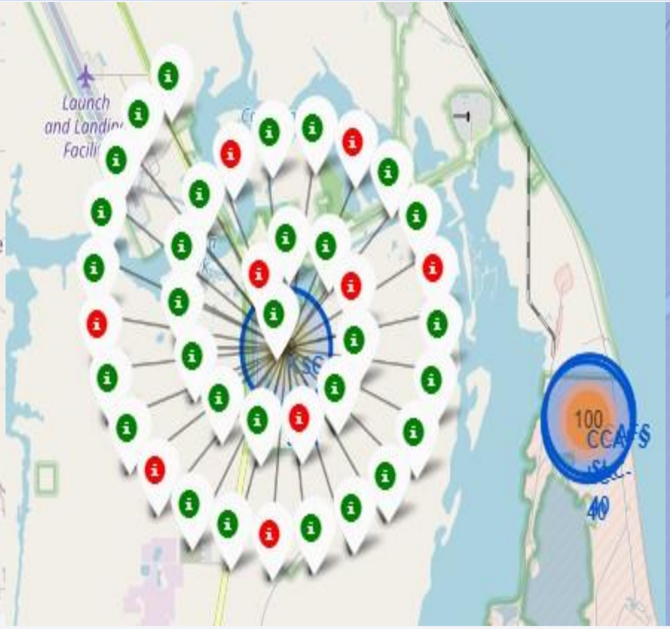
Mapping launch sites using Folium is a powerful tool for visualizing their spatial distribution and exploring patterns. It provides geographical context and insight into proximity to coastlines, water bodies, and populated areas, aiding safety and logistics.

From the screenshot above it is observed apart from launch site VAFB SLC-4E, other launch sites are densely in close proximity.

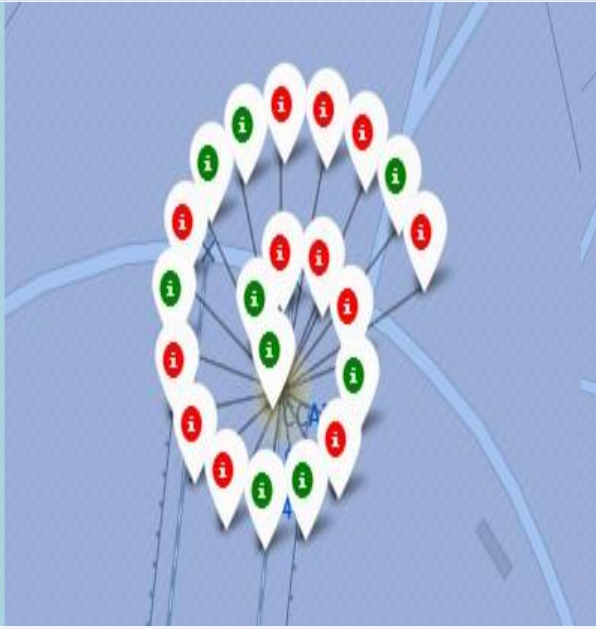
Launch Sites Successes and Failure using Folium



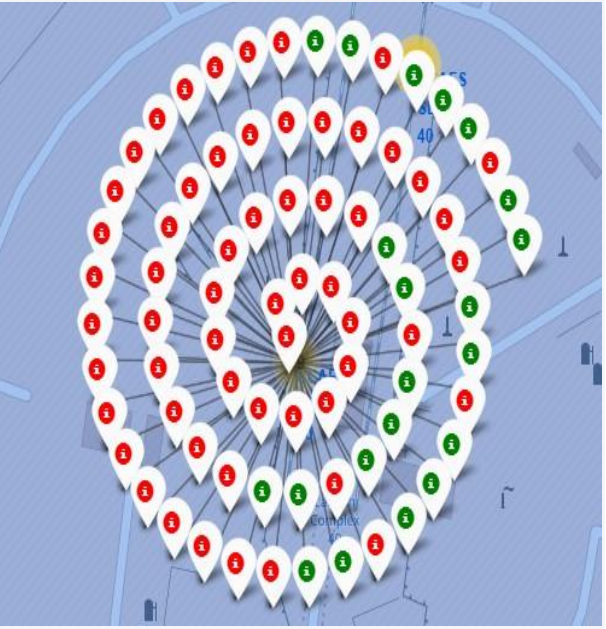
VAFB SLC 4E



KSC LC 39A



CCAFS SLC 40



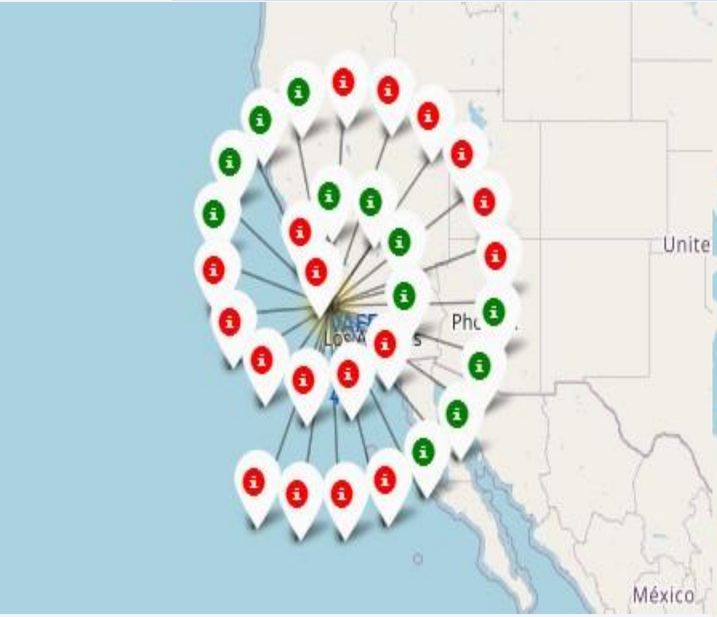
CCAFS LC-40

The provided screenshots above depict the outcomes of first-stage landings, with successes marked in green and failures in red. These visual representations were generated through the utilization of Folium map clusters and Folium markers, which enabled the grouping of launch sites and the differentiation of their success and failure records.

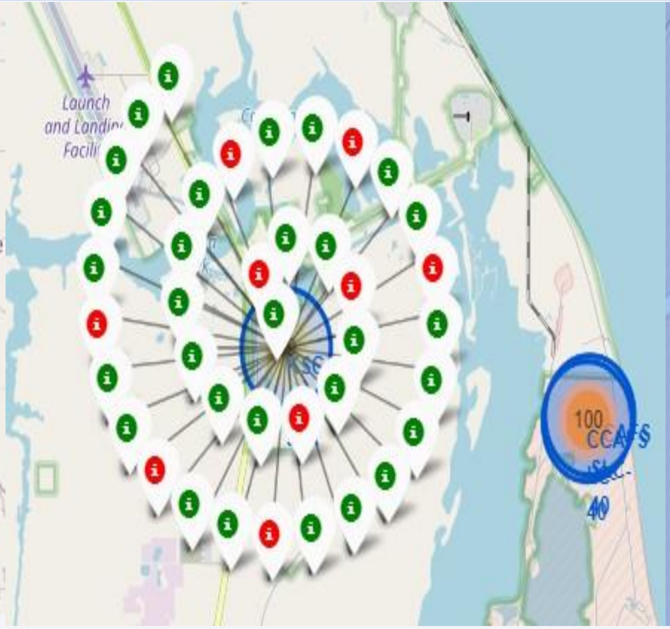
Key Findings:

Upon closer examination of the visual data, several noteworthy observations have emerged:

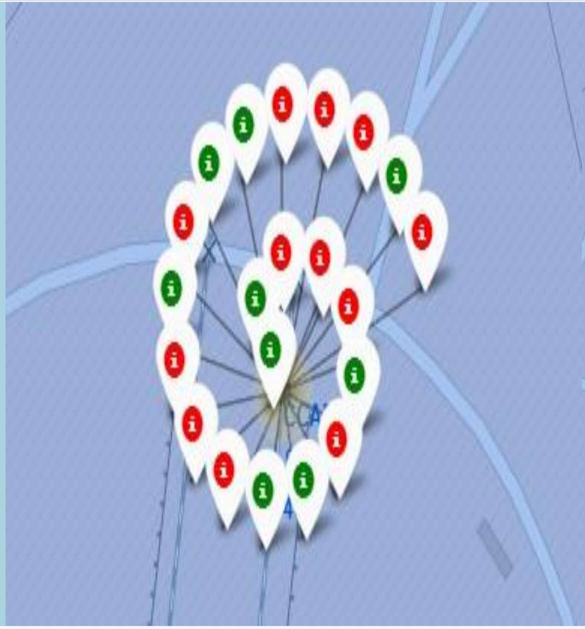
Launch Sites Successes and Failure using Folium



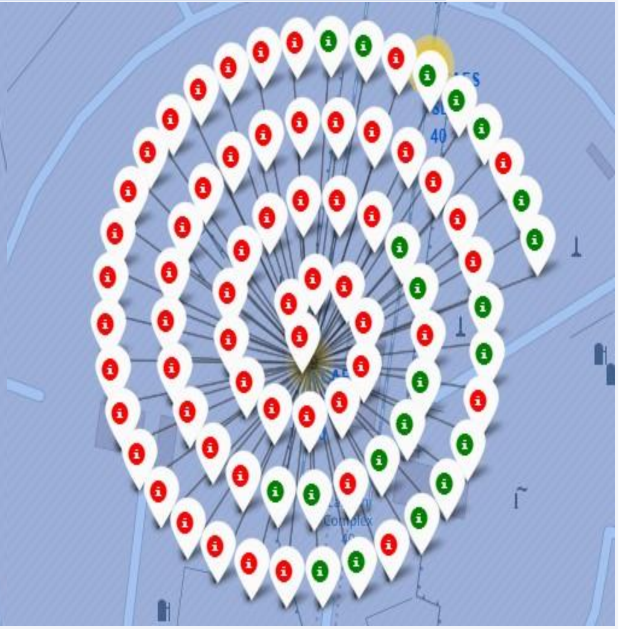
VAFB SLC 4E



KSC LC 39A



CCAFS SLC 40

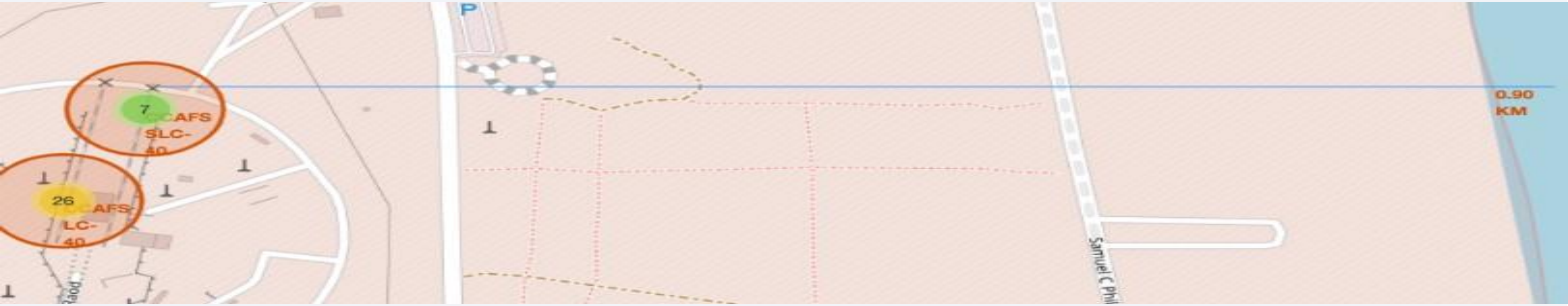


CCAFS LC-40

- **Launch Site Analysis:**
  - KSC LC 39A emerges as the launch site with the highest number of successful first-stage landings, denoted by the green markers.
  - Conversely, CCAFS SLC 40, despite having a substantial number of launches, exhibits the highest count of failures in first-stage landings, indicated by the red markers.
  - The other launch sites (VAFB SLC 4E AND CCAFS SLC 40) are also shown to depict their successes and failures in green and red markers respectively

These findings shed light on the performance of different launch sites in terms of their success rates for first-stage landings, offering valuable insights into their operational outcomes.

Proximity Analysis using Folium

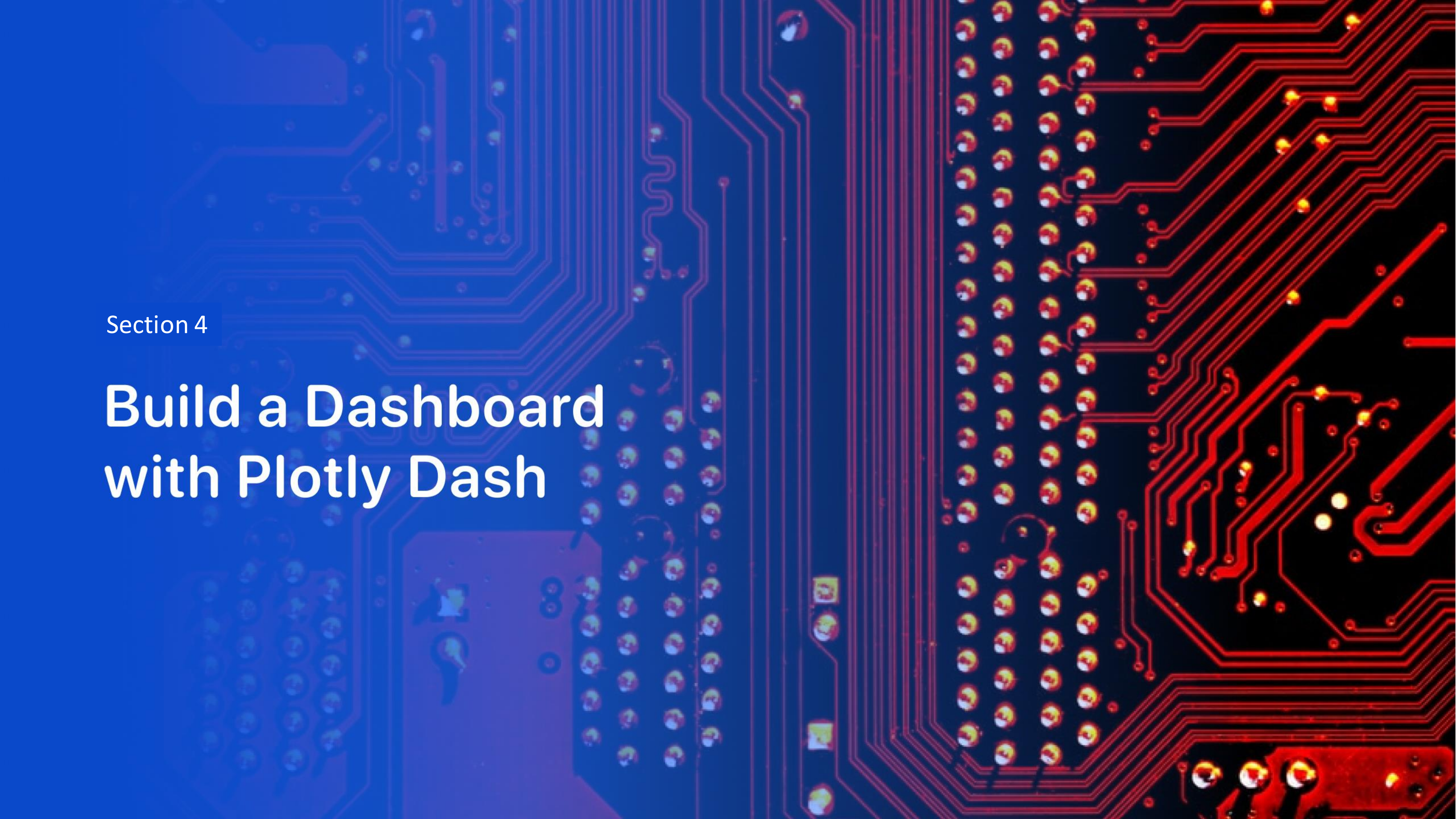


Key Components:

- **Mouse Position:** The Folium map incorporates mouse position functionality, which facilitates the identification of latitude and longitude coordinates when the mouse pointer is placed over a specific location on the map.
- **Calculate Distance Function:** An essential feature of the map is the "calculate distance" function, which serves the purpose of measuring the geographical distance between two distinct locations marked on the map.
- **Folium Map Polyline:** The map employs Folium's polyline feature, enabling the visualization of a line segment that denotes the distance between two designated points on the map.

Notable Insights:

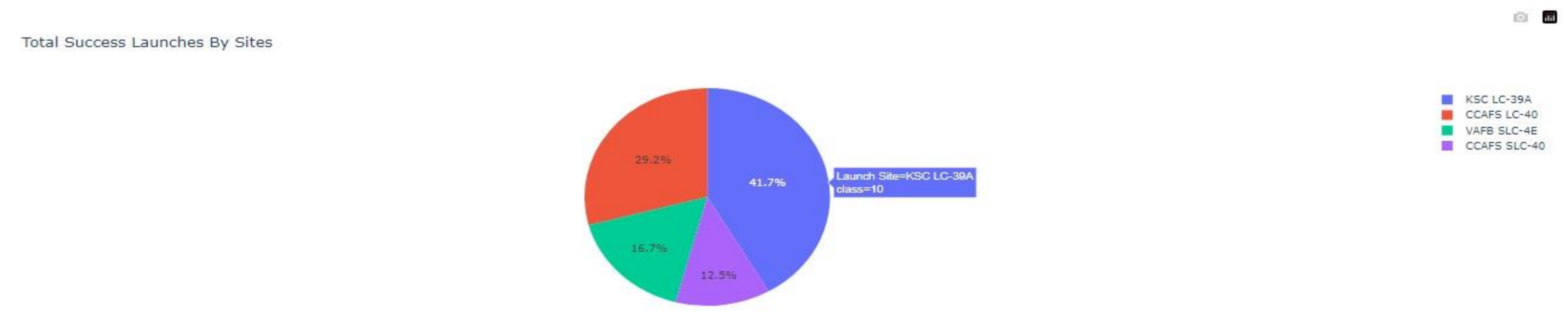
A comprehensive analysis of the launch sites reveals a strategic positioning strategy that prioritizes proximity to coastlines. This deliberate choice ensures the execution of safe launch trajectories over expansive bodies of water. Such proximity to the coast plays a pivotal role in guaranteeing the safety and controlled progression of rocket launches, particularly during critical phases such as booster separation and ascent. Furthermore, this geographical advantage grants launch missions the flexibility to attain a diverse range of orbital inclinations and azimuths, aligning with the distinctive requirements of various missions.



Section 4

Build a Dashboard with Plotly Dash

Dashboard of Total Successes By Site



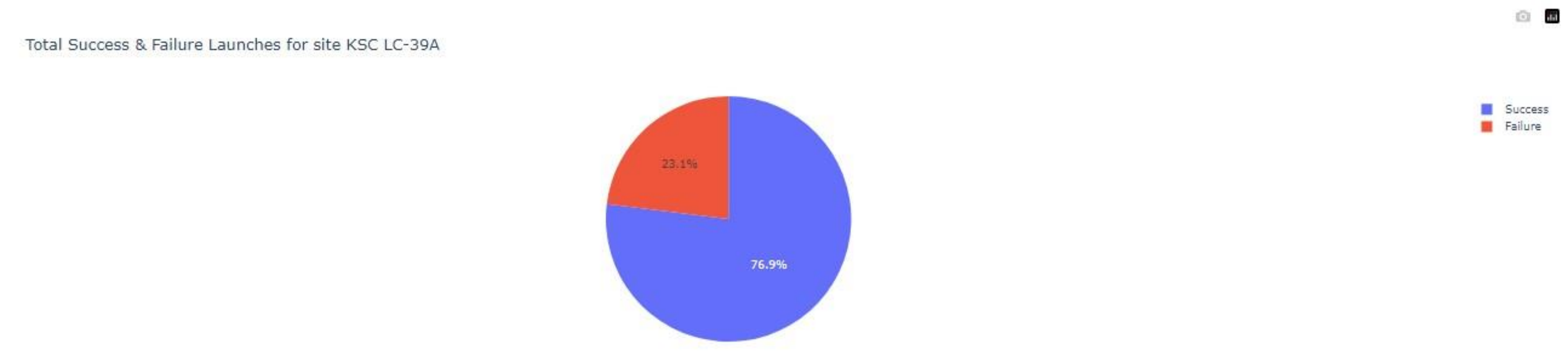
The presented pie chart illustrates the cumulative count of successful launches across various launch sites.

Important Elements and Findings on the Screenshot:

- **Pie Chart**: The primary visualization element, the pie chart, conveys a clear overview of the distribution of successful launches among different launch sites. Each segment of the chart represents a specific launch site, with the size of the segment proportionate to the number of successful launches at that site.
- **KSC LC-39A Dominance**: The most striking finding is the dominance of the KSC LC-39A launch site in terms of successful launches. This is evident from the noticeably larger portion of the pie chart attributed to this site, signifying its exceptional performance in first-stage landings.
- **Launch Site Identification:** Launch site labels accompanying each chart segment facilitate the identification of individual launch sites and their corresponding success counts. This feature enhances the comprehensibility of the chart.

In summary, the pie chart visually represents the distribution of successful launches among different launch sites, with KSC LC-39A emerging as the leader in terms of first-stage landing successes. This visualization serves as a valuable tool for understanding the success rates of various launch sites within the context of the space launch program.

Dashboard of Launch Site with Maximum Success



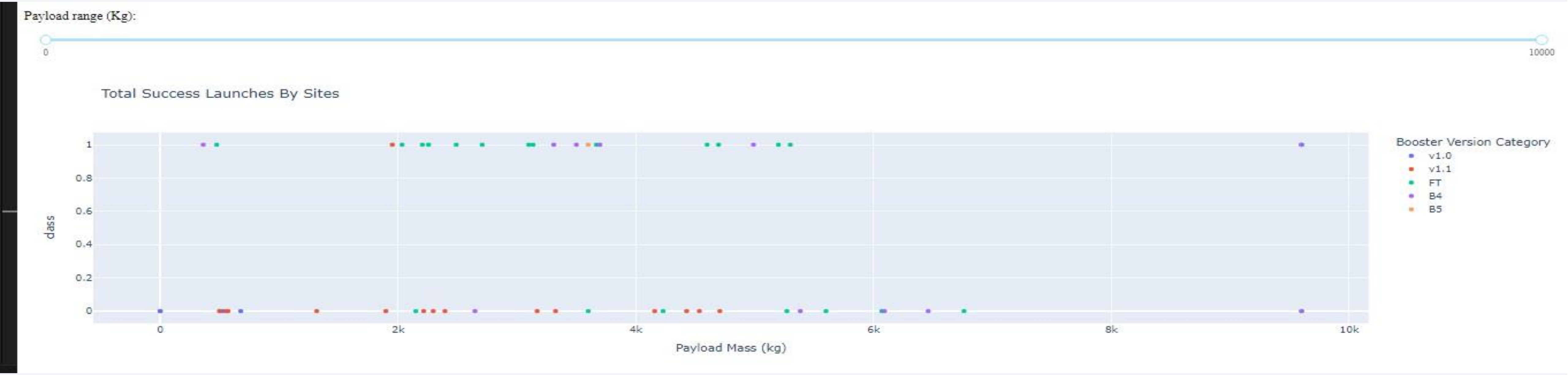
The presented screenshot features a pie chart illustrating the outcomes of first-stage landings at Launch Site KSC LC-39.

Important Elements and Findings on the Screenshot:

- **Pie Chart:** The central visualization element is the pie chart, which effectively conveys the distribution of first-stage landing outcomes. Within the chart, two distinct segments are visible, each representing either a successful (highlighted in purple) or failed (typically indicated by red) landing.
- **KSC LC-39 Pre-eminence**: A significant and noteworthy discovery is the unparalleled success of Launch Site KSC LC-39 in terms of first-stage landings. The prominent purple segment within the pie chart vividly depicts this exceptional performance, firmly establishing its position as the leader.
- **Outcome Identification:** The pie chart further allows viewers to easily discern between successful and failed landings through colour differentiation. This aids in quickly assessing the site's performance.

The screenshot offers a succinct visual summary of the outcomes of first-stage landings at Launch Site KSC LC-39, with the dominant purple colouration signifying its outstanding success. This graphical representation serves as a valuable tool for understanding the launch site's exemplary record in achieving successful first-stage landings within the broader context of space launch operations.

Payload vs. Launch Outcome Dashboard with Booster Version Insights

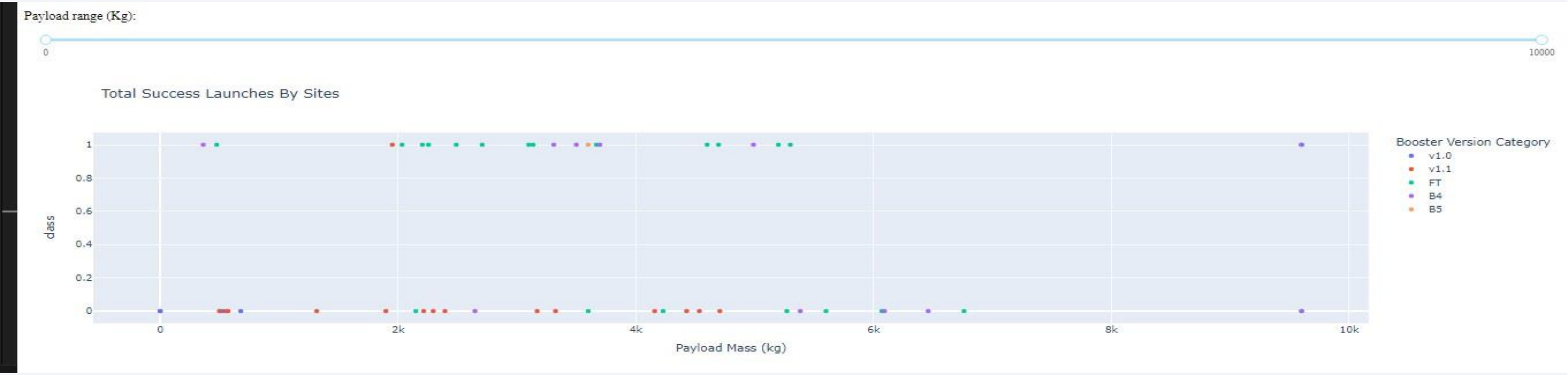


The provided screenshot presents a comprehensive data visualization dashboard that includes a scatter plot featuring Payload vs. Launch Outcome (Class) with colour differentiation based on the Booster Version Category. Additionally, it offers an interactive range slider to adjust the payload mass, allowing for deeper insights into the outcomes of first-stage landings, contingent upon the payload's mass.

Key Elements and Findings on the Screenshot:

- **Scatter Plot:** The central element of the dashboard is the scatter plot, which effectively depicts the relationship between Payload Mass and Launch Outcome. Each data point on the plot represents a specific payload, with colour distinctions denoting the respective Booster Version Category. This visualization aids in discerning patterns and trends related to payload mass and launch succes s.
- **Range Slider:** An essential feature is the range slider, prominently positioned within the dashboard. This interactive tool empowers users to dynamically adjust the payload mass range. By manipulating the range, users can delve into the impact of payload mas s on the distribution of launch outcomes.

Payload vs. Launch Outcome Dashboard with Booster Version Insights



- **Noteworthy Findings:**

Upon careful examination of the screenshot, several notable findings emerge:

- **Booster Version FT:** It is evident that Booster Version FT exhibits the highest number of successful first-stage landings within the payload mass range of 500 to 5000 kilograms. This is vividly portrayed through the distinctive green colouration on the scatter plot.
- **Payload Mass Range:** The scatter plot conveys valuable insights into the relationship between payload mass and launch success. Users can observe how the success rate varies across different payload mass ranges, identifying specific mass intervals associated with higher or lower success rates.

In conclusion, this dashboard serves as a powerful analytical tool for exploring the influence of payload mass on launch outcomes, with a focus on booster versions. The visualization facilitates the identification of payload mass ranges and booster versions linked to the highest success rates, offering valuable insights for decision-making and performance evaluation in the context of space launch missions.

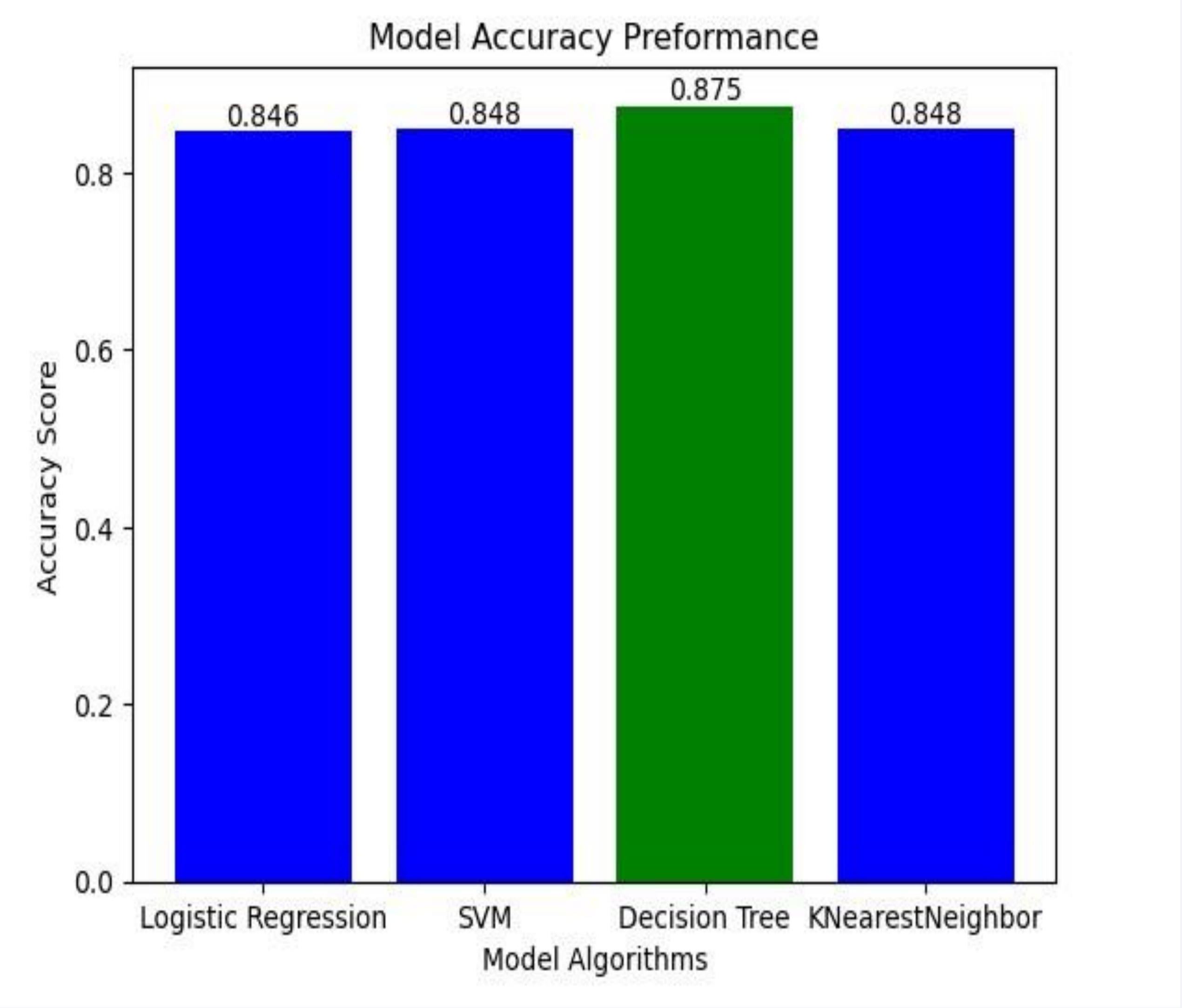
Section 5

Predictive Analysis (Classification)

Classification Accuracies

Explanations:

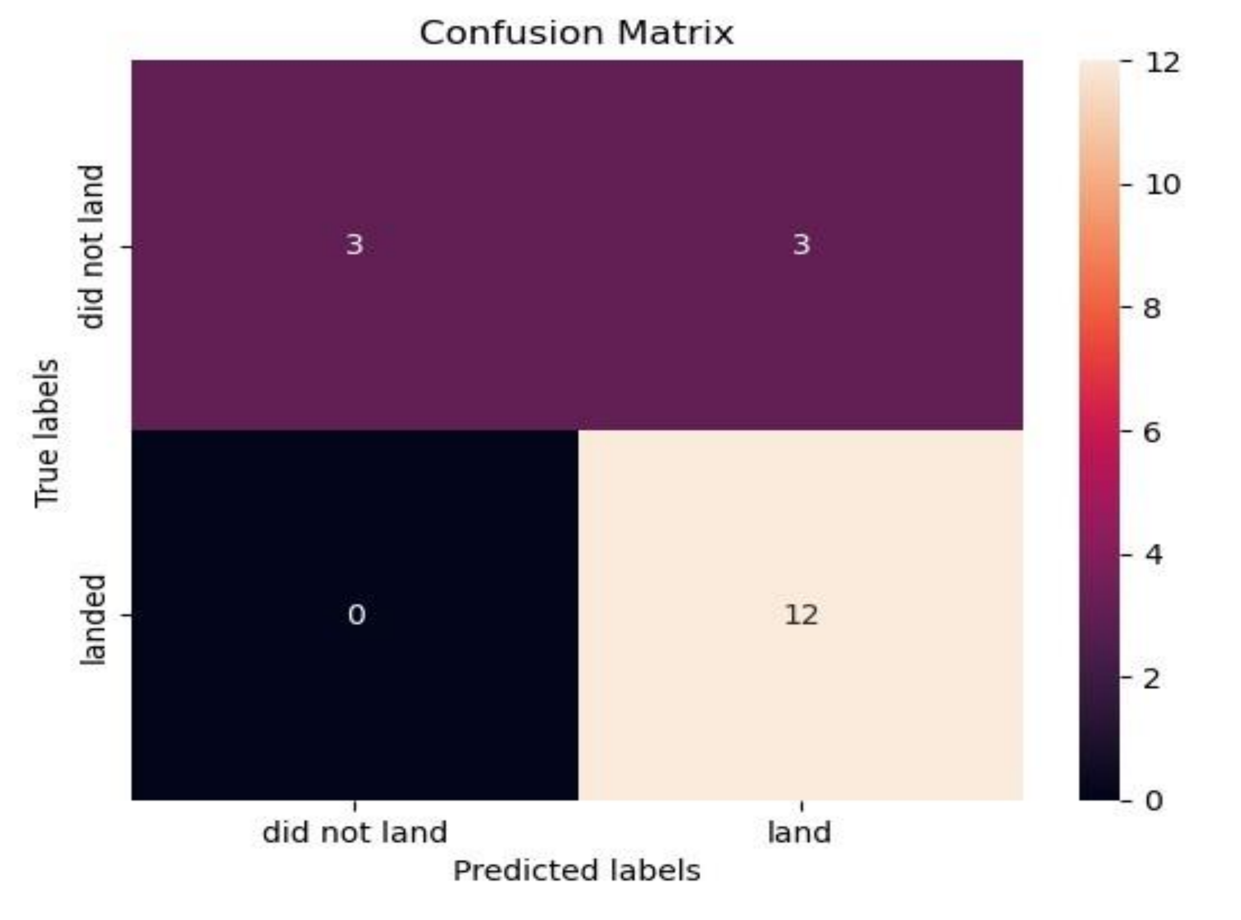
The bar plot illustrates the various model algorithms employed in training and testing a predictive model for first stage landing outcomes. Upon careful examination of the plot, a discernible trend emerges, with the Decision Tree training algorithm notably outperforming others in terms of accuracy. This exceptional performance is visually denoted by the green coloration, signifying an impressive accuracy score of 0.875.



Confusion Matrix of Best Model

Explanations:

- The image presents the Confusion Matrix of the Decision Tree algorithm-trained model, which has emerged as the top-performing model due to its exceptional accuracy in predicting first-stage landing outcomes for SpaceX missions. Within the Confusion Matrix, critical values are discerned: True Positives are recorded as 3, False Positives as 3, False Negatives as 0, and True Negatives as 9. These metrics provide invaluable insights into the model's performance and its capacity to correctly classify landing outcomes.



Conclusions

In this data science project, our primary focus was on predicting the successful landing of SpaceX missions. Through a comprehensive analysis encompassing various stages, we've unearthed valuable insights and contributed to a deeper understanding of the factors influencing mission outcomes.

- **Data Collection and Wrangling**:

  Our journey commenced with data collection, where we leveraged web scraping techniques and the SpaceX API to acquire the necessary data. Subsequently, we undertook rigorous data wrangling, ensuring the data was clean and structured for analysis.

- **Exploratory Data Analysis (EDA):**

  Employing SQL, we delved into exploratory data analysis, shedding light on critical aspects of the SpaceX missions. Our SQL-based analysis empowered us to glean meaningful insights and perform targeted investigations.

- **Geospatial Analysis:**

  To provide a geographical context, we harnessed the power of Folium to create an interactive map. This map unveiled the spatial distribution of launch sites, their proximity to coastlines, and the landing outcomes at each site. This visual tool proved invaluable for understanding the geographic nuances of the missions.

- **Data Visualization:**

  Our journey was punctuated by a variety of data visualization techniques, including scatter plots, bar plots, pie charts, and line plots. These visualizations played a pivotal role in illuminating hidden trends and patterns within the collected data.

- **Dashboard Creation:**

  For a consolidated view of the data, we designed a dashboard that featured a pie chart illustrating the success of individual and collective launch sites. Furthermore, we incorporated a scatter plot with a payload range slider, enhancing our understanding of the booster versions' performance across various payload ranges.

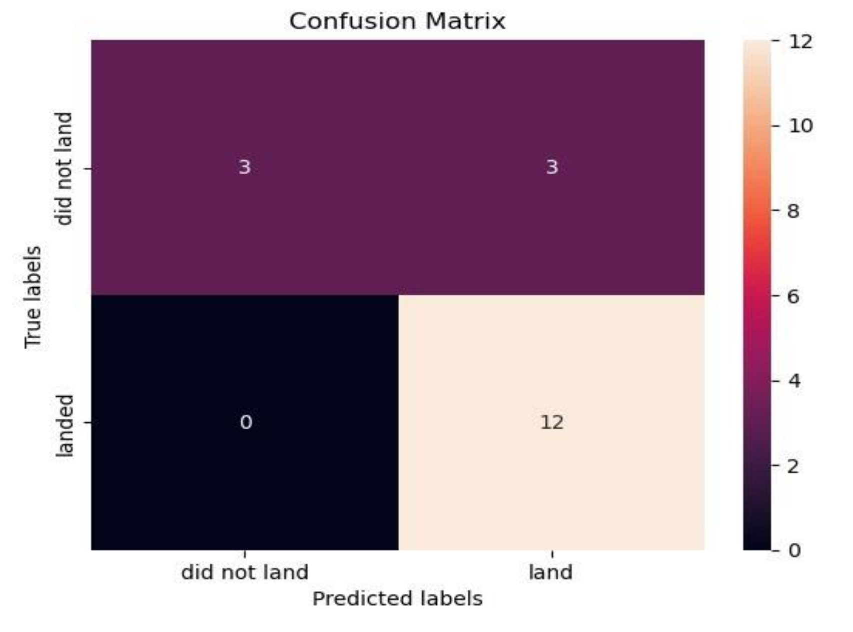
- **Predictive Analysis:**

  The pinnacle of our project involved predictive analysis. We trained and tested multiple machine learning models, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression, and Decision-Tree models. Through rigorous evaluation, the Decision-Tree model emerged as the clear winner, boasting remarkable accuracy in predicting landing outcomes.

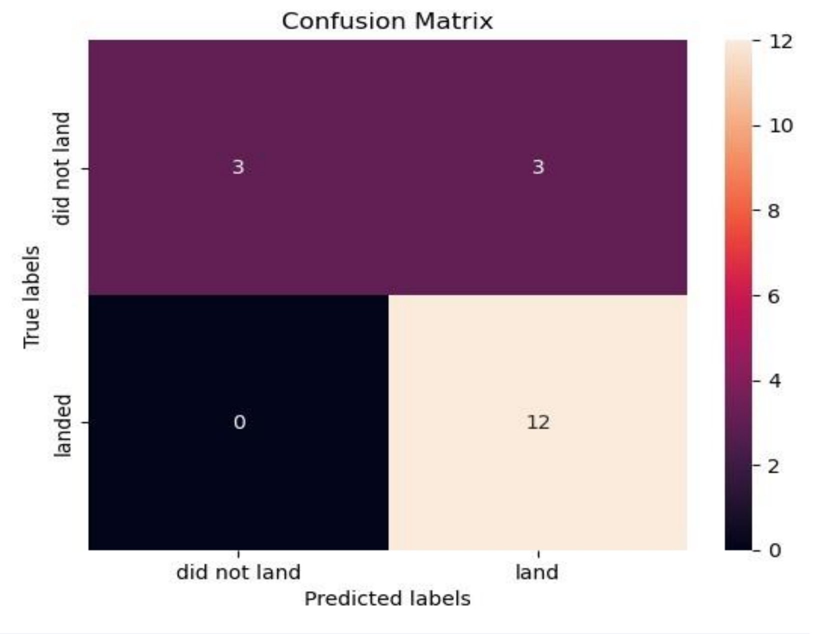
In summary, our comprehensive analysis journey has provided valuable insights into SpaceX missions. From data collection and exploration to geospatial analysis and predictive modelling, we've armed ourselves with a deeper understanding of the factors influencing successful landings. This project showcases the power of data science in unravelling the complexities of real-world scenarios and making informed predictions for the future.

Appendix

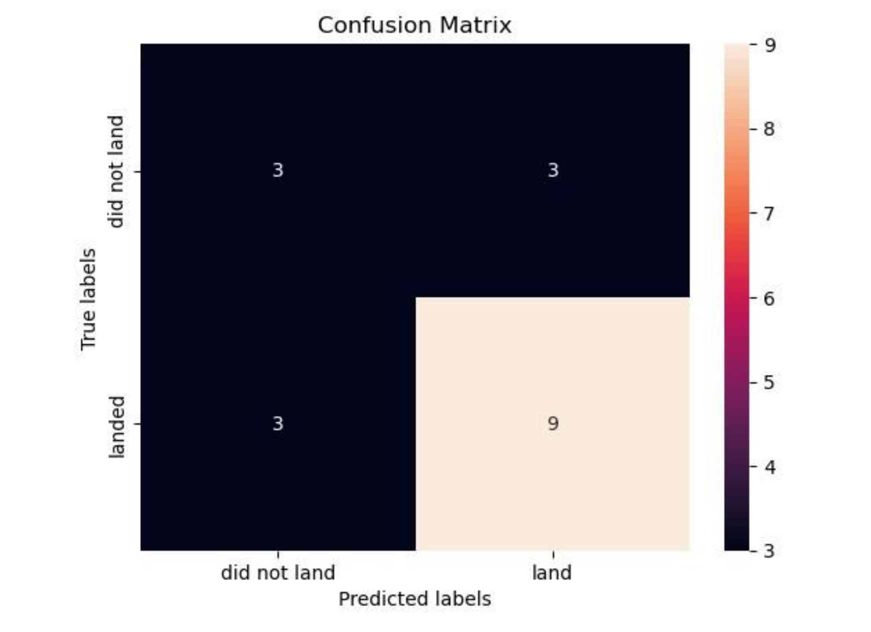
Confusion matrix of the other models



KNN



Support-Vector-Machine(SVM)



Logistic Regression

GitHub Repository of all Notebook used: https://github.com/MarcusIfeanyi/Data_science_Coursera/tree/main/Notebooks

GitHub Repository of all python code snippets:
https://github.com/MarcusIfeanyi/Data_science_Coursera/tree/main/Python_files

Thank you!