IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amon Apolonio Vieira

04/2024

Outline

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

Executive Summary

The project utilized a systematic approach to analyze data from the SpaceX API. Data collection was achieved through Python's requests library, ensuring comprehensive data retrieval. Following data collection, a thorough data wrangling process was undertaken to address missing values and correct data types, enhancing the dataset's quality and reliability.

Exploratory Data Analysis (EDA) was conducted using various visualization techniques, providing valuable insights into the dataset. Interactive visual analytics were then performed using Folium and Plotly Dash, enabling the visualization of spatial and temporal patterns within the data in an interactive manner.

For predictive analysis, classification models including Logistic Regression, SVM, Decision Tree, and KNN were employed. This involved standardizing features, splitting data into training and testing sets, performing hyperparameter tuning, and evaluating model accuracy.

By employing these methodologies, the project aimed to gain a comprehensive understanding of SpaceX launch outcomes and provide valuable insights for future analysis and decision-making.

Introduction

Project Background and Context:

- The project focuses on analyzing data from the SpaceX API to gain insights into SpaceX launch outcomes. SpaceX, founded by Elon Musk in 2002, has become a prominent player in the aerospace industry, conducting numerous successful rocket launches and pioneering reusable rocket technology. Understanding the factors influencing launch success is crucial for improving mission planning, safety, and overall operational efficiency.

Problems to Address:

- Predictive Analysis: One of the key objectives is to develop and evaluate machine learning models to predict launch outcomes based on various factors such as payload mass, launch site, and launch date. This analysis can provide valuable insights into the likelihood of a successful launch and help optimize mission planning and resource allocation.
- Exploratory Data Analysis (EDA): Another objective is to perform EDA to uncover patterns and trends in the data that may influence launch outcomes. By visualizing the data and identifying correlations between different variables, we can gain a deeper understanding of the factors that contribute to launch success or failure.
- Overall, the project aims to leverage data analytics and machine learning techniques to enhance our understanding of SpaceX launch outcomes and contribute to the ongoing advancements in space exploration.

Section 1

Methodology

Methodology

Executive Summary

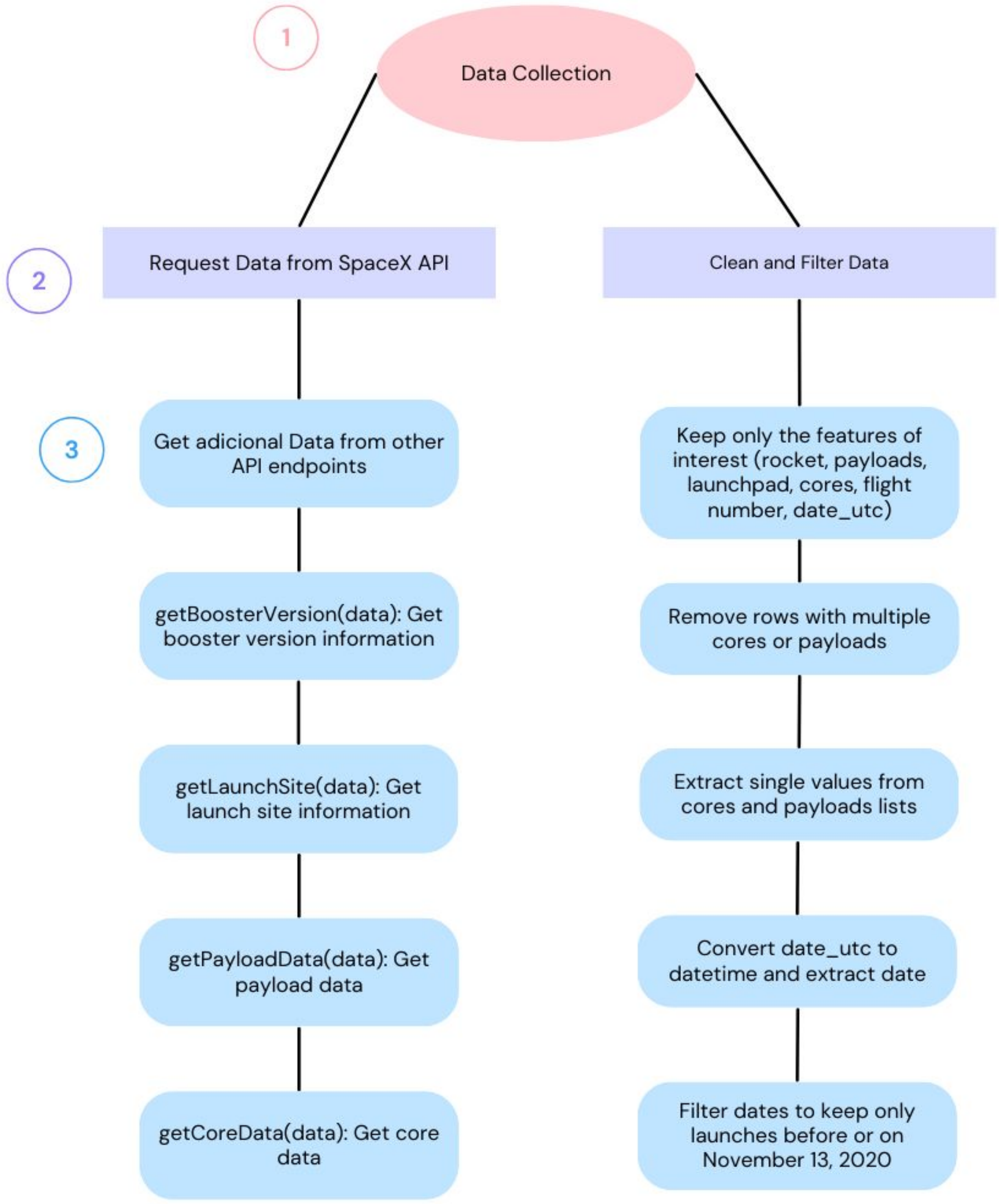
- Data collection methodology:
    - Data was collected from the SpaceX API using Python's requests library.
- Perform data wrangling
    - Missing values and incorrect data types were addressed.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - The code utilizes Logistic Regression, SVM, Decision Tree, and KNN classifiers. It standardizes features, splits data, performs hyperparameter tuning, and evaluates model accuracy.

Data Collection – SpaceX API

- Request Data
- Get adicional Data from other API endpoints
- Filter Data
- Clean Data
- Create DataFrame

Data Collection Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-spacex-data-collection-api.ipynb



1

Data Collection

2

Request Data from SpaceX API

3

Get adicional Data from other API endpoints

getBoosterVersion(data): Get booster version information

getLaunchSite(data): Get launch site information

getPayloadData(data): Get payload data

getCoreData(data): Get core data

Clean and Filter Data

Keep only the features of interest (rocket, payloads, launchpad, cores, flight number, date_utc)

Remove rows with multiple cores or payloads

Extract single values from cores and payloads lists

Convert date_utc to datetime and extract date

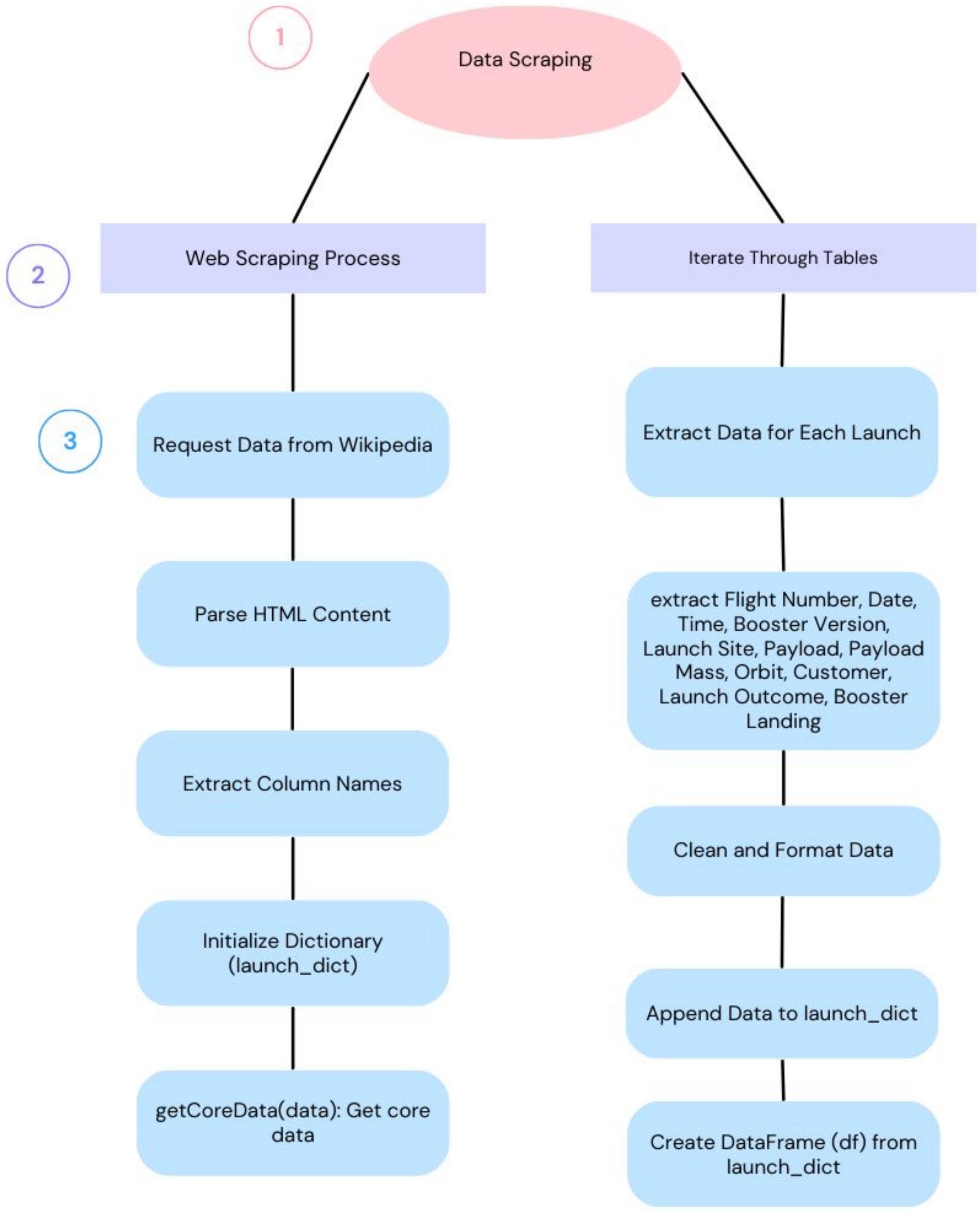
Filter dates to keep only launches before or on November 13, 2020

Data Collection - Scraping

- Request Data: Retrieving the HTML content of the Wikipedia page.
- Parse HTML: Using BeautifulSoup to parse and extract tables
- Extract Column Names: Cleaning column names.
- Iterate Through Tables: Processing each row for flight details.
- Clean and Format Data: Utilizing functions for data cleaning

Web Scraping Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-webscraping.ipynb



1 Data Scraping

2 Web Scraping Process

Iterate Through Tables

3 Request Data from Wikipedia

Parse HTML Content

Extract Column Names

Initialize Dictionary (launch_dict)

getCoreData(data): Get core data

Extract Data for Each Launch

extract Flight Number, Date, Time, Booster Version, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Booster Landing

Clean and Format Data

Append Data to launch_dict

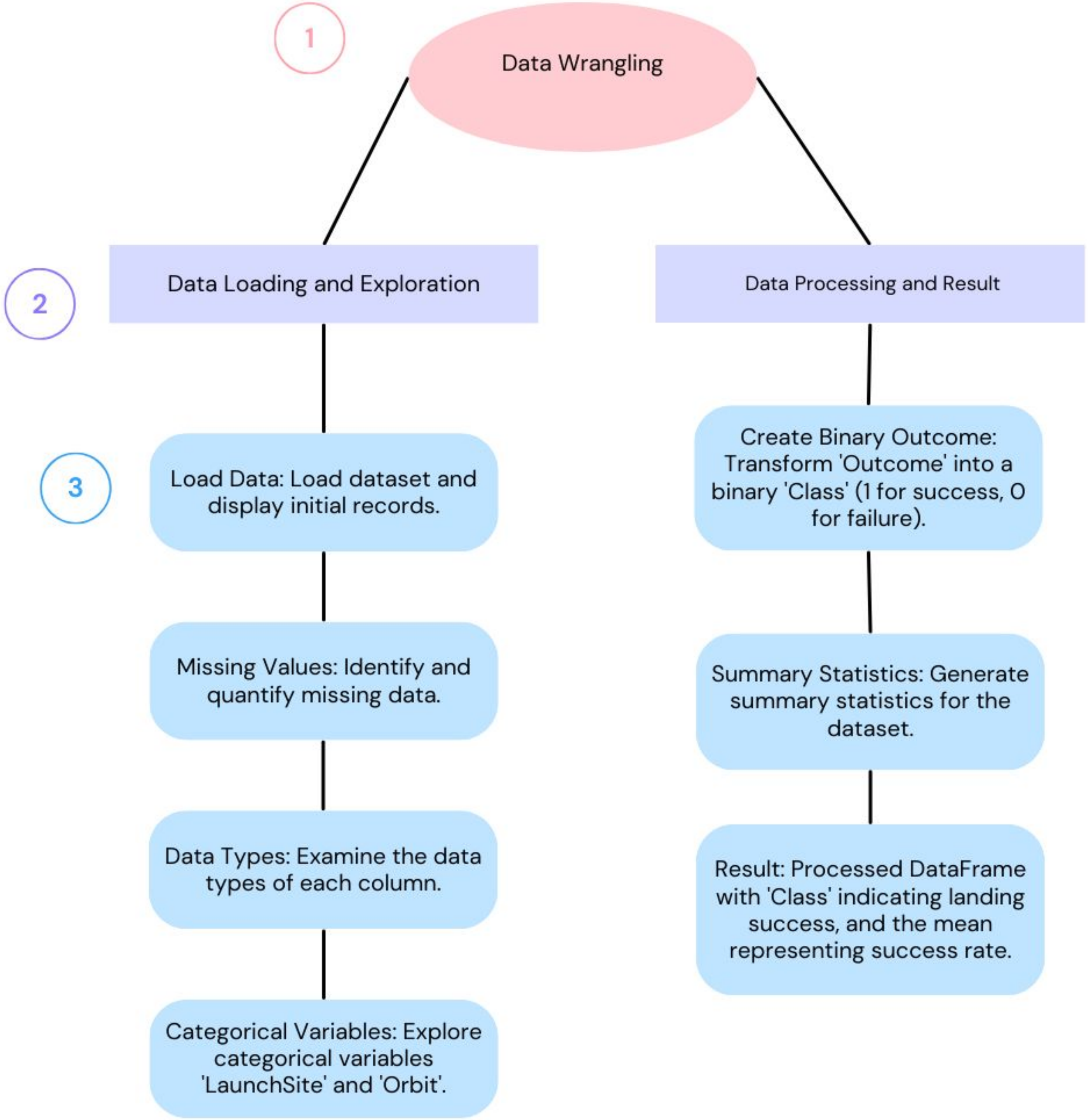
Create DataFrame (df) from launch_dict

Data Wrangling

- Loaded the dataset from a CSV file.
- Checked for missing values and calculated the percentage of missing values for each column.
- Checked the data types of each column.
- Explored the distribution of categorical variables.
- Created a binary variable 'Class' to indicate successful (1) or unsuccessful (0) landings.
- Generated summary statistics of the DataFrame.

Data Wrangling Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb



1
Data Wrangling
2
Data Loading and Exploration
Data Processing and Result
3
Load Data: Load dataset and display initial records.
Missing Values: Identify and quantify missing data.
Data Types: Examine the data types of each column.
Categorical Variables: Explore categorical variables 'LaunchSite' and 'Orbit'.
Create Binary Outcome: Transform 'Outcome' into a binary 'Class' (1 for success, 0 for failure).
Summary Statistics: Generate summary statistics for the dataset.
Result: Processed DataFrame with 'Class' indicating landing success, and the mean representing success rate.

EDA with Data Visualization

Charts Plotted and Reasons:

- FlightNumber vs. PayloadMass: Shows how flight number and payload mass relate to landing success, indicating heavier payloads may lead to less successful landings.
- FlightNumber vs. LaunchSite: Identifies any patterns between launch site and mission success.
- PayloadMass vs. LaunchSite: Illustrates the relationship between payload mass and launch site.
- Orbit Type vs. Success Rate: Displays the average success rate for each orbit type, indicating their impact on mission success.
- FlightNumber vs. Orbit Type: Reveals any patterns between flight number and orbit type.
- PayloadMass vs. Orbit Type: Shows how payload mass relates to orbit type and landing success.
- Yearly Average Success Rate: Visualizes the trend in launch success over time.

Data Visualization Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Summary of SQL queries

- Unique launch sites
- Records with launch sites starting with 'CCA'
- Total payload mass by NASA (CRS) boosters
- Average payload mass by booster version F9 v1.1
- Date of first successful ground pad landing
- Boosters with drone ship success and payload between 4000 and 6000
- Total successful and failure mission outcomes
- Booster versions with maximum payload mass
- Month names, drone ship failure landing outcomes, booster versions, and launch sites in 2015
- Count of landing outcomes between 2010-06-04 and 2017-03-20, ranked

Data Visualization with SQL Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Summary of map objects

- Added Circle objects around launch sites with popups showing site names.
- Used Marker objects with custom icons and popups for each launch site.
- Employed MarkerCluster for nearby site grouping.
- Integrated MousePosition for real-time latitude and longitude info.
- Calculated and visualized distances to the nearest coastline.
- Used PolyLine objects to connect sites to coastlines and significant locations.

These map objects were added to provide a comprehensive visual representation of launch sites and their surroundings, aiding in understanding their geographical context and relationships.

Interactive Map with Folium Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Success Pie Chart:

- Summary and Interactions: This pie chart shows the total successful launches count for all sites or for a specific site if selected. Users can select a launch site from the dropdown list to view the success vs. failure counts for that site.
- Purpose: Provides an overview of the success rates at different launch sites, helping stakeholders understand the performance of each site.

Success vs. Payload Scatter Chart:

- Summary and Interactions: This scatter chart displays the correlation between payload mass and launch success, with the ability to filter by payload range and launch site. Users can select a launch site from the dropdown list and adjust the payload range using a slider.
- Purpose: Helps understand how payload mass affects launch success, allowing analysis based on specific criteria like launch site and payload range.

Code for Dashboard with Plotly Dash

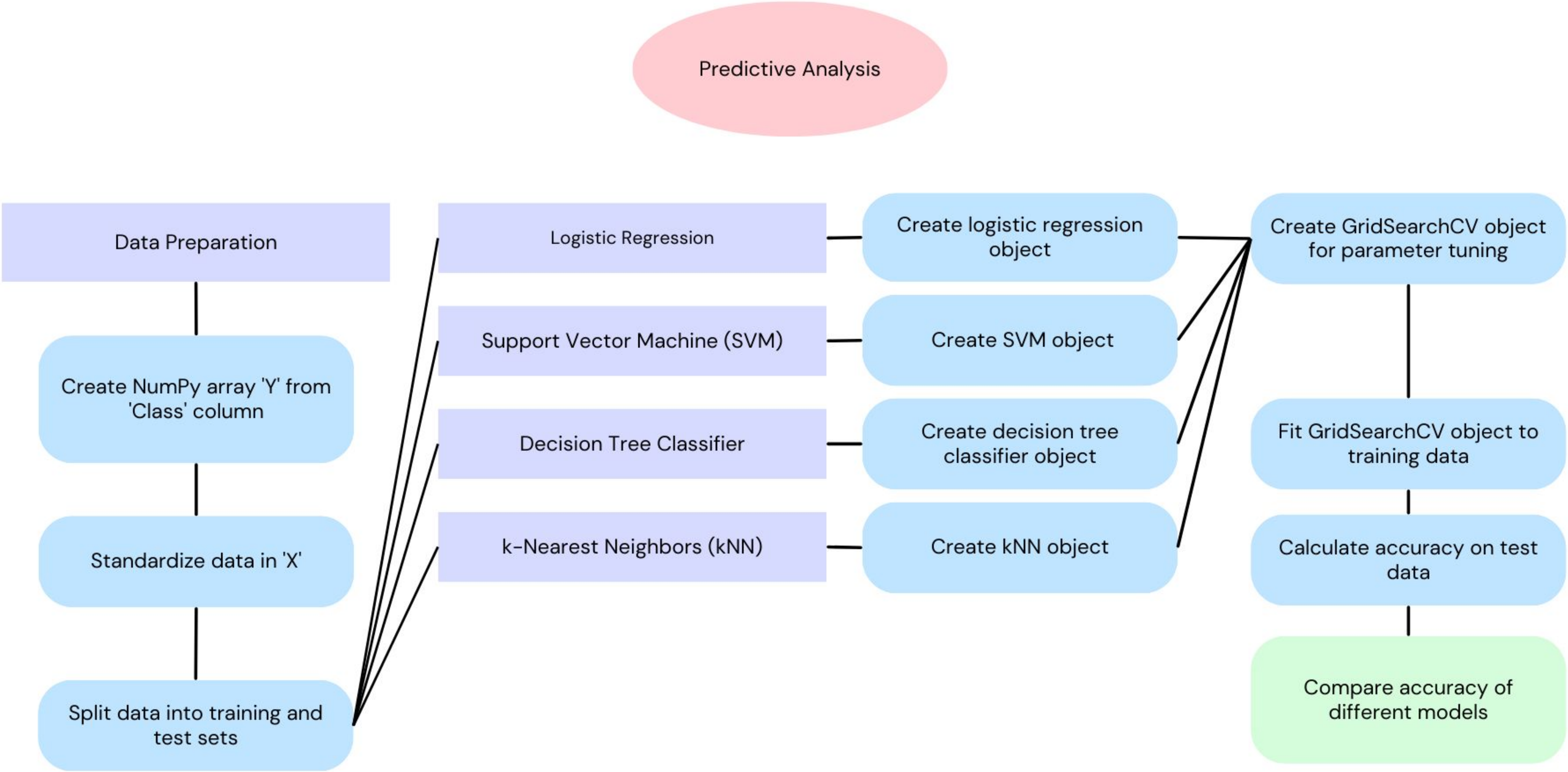
- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/dash_interactivity.py

Predictive Analysis (Classification)

- Data Preparation: Created NumPy arrays from the data and standardized them.
- Splitting Data: Split the data into training and test sets.
- Model Building:
- Used GridSearchCV to find the best parameters for each algorithm.
- Trained the models on the training data.
- Model Evaluation: Calculated the accuracy of each model on the test data.
- Model Comparison: Compared the accuracy of the models to find the best performer.
- Based on the results, the Decision Tree model had the highest accuracy of 0.8889, making it the best performing model for this classification task.

Predictive Analysis Notebook

- https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Predictive Analysis

Data Preparation

Create NumPy array 'Y' from 'Class' column

Standardize data in 'X'

Split data into training and test sets

Logistic Regression

Support Vector Machine (SVM)

Decision Tree Classifier

k-Nearest Neighbors (kNN)

Create logistic regression object

Create SVM object

Create decision tree classifier object

Create kNN object

Create GridSearchCV object for parameter tuning

Fit GridSearchCV object to training data

Calculate accuracy on test data

Compare accuracy of different models

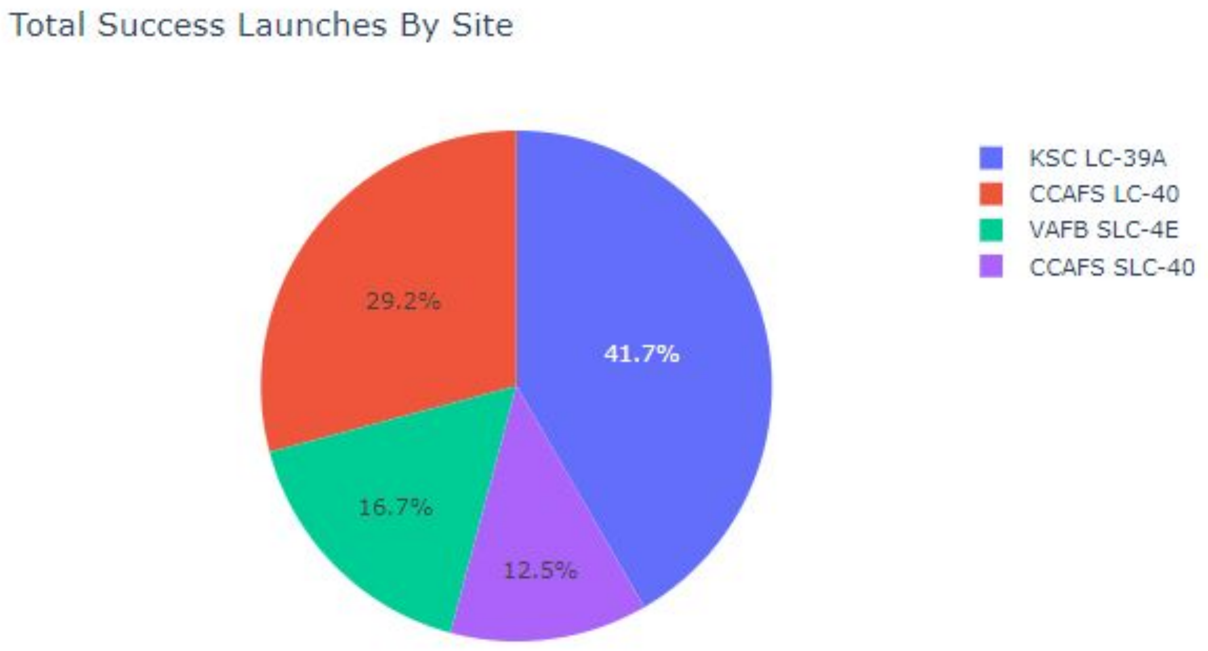
Results

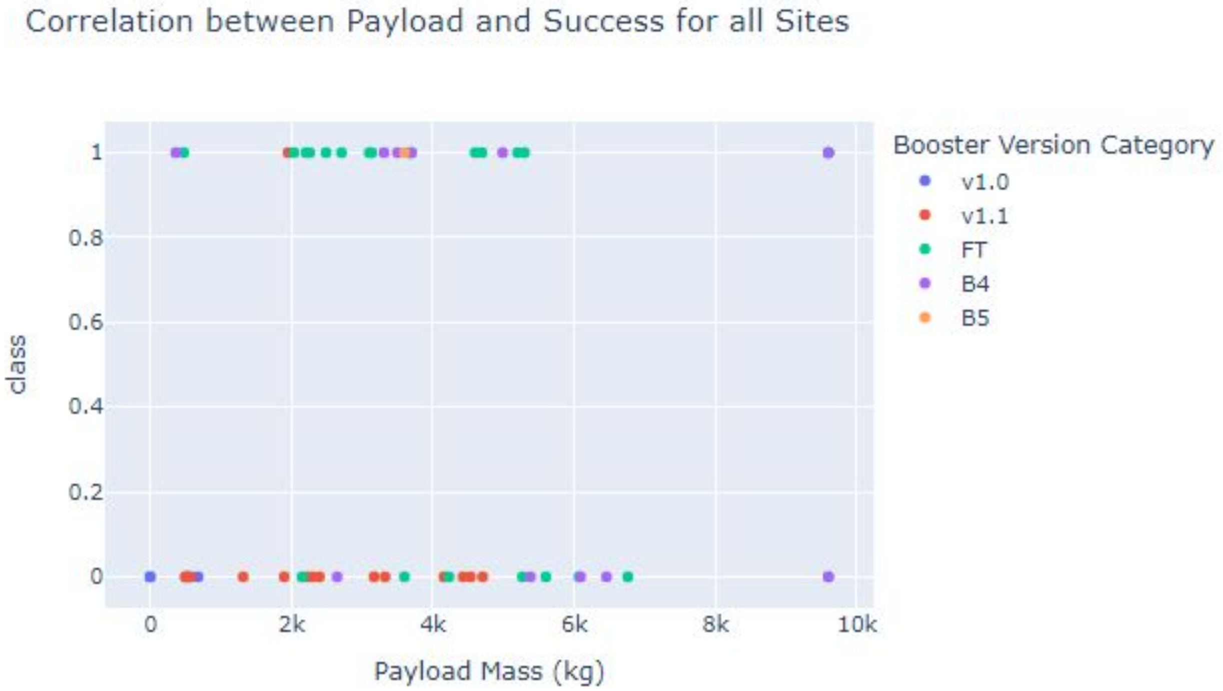
Exploratory data analysis results

- Overall, these EDA results provide valuable insights into the dataset, which can be used to further analyze and model the data for predicting launch success.

Predictive analysis results

- The predictive analysis results indicate that the Decision Tree model had the highest accuracy of 0.8889, making it the best performing model for the classification task. The other models, including Logistic Regression, Support Vector Machine, and k-Nearest Neighbors, also performed well with accuracies of 0.8333. These results suggest that the Decision Tree model is the most suitable for predicting the outcome of spaceX launches based on the given dataset.



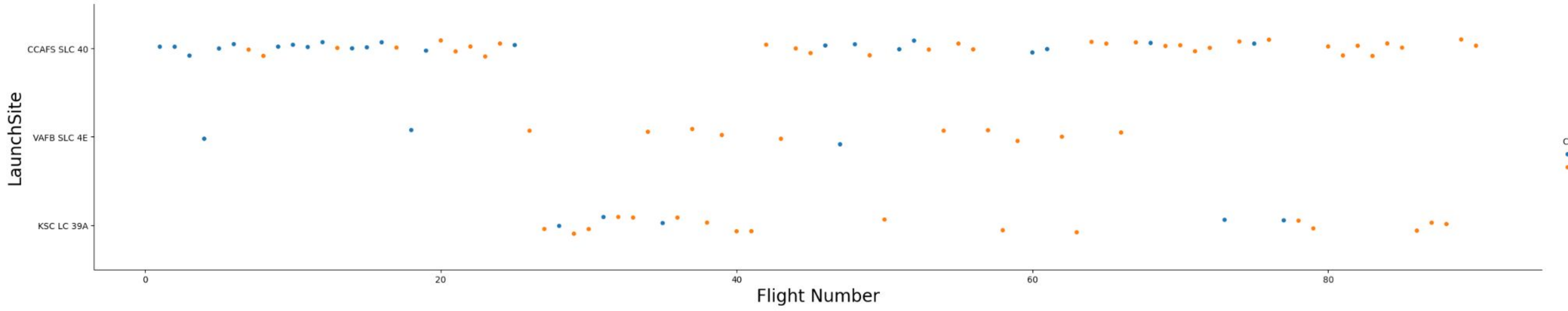


Section 2

Insights drawn from EDA

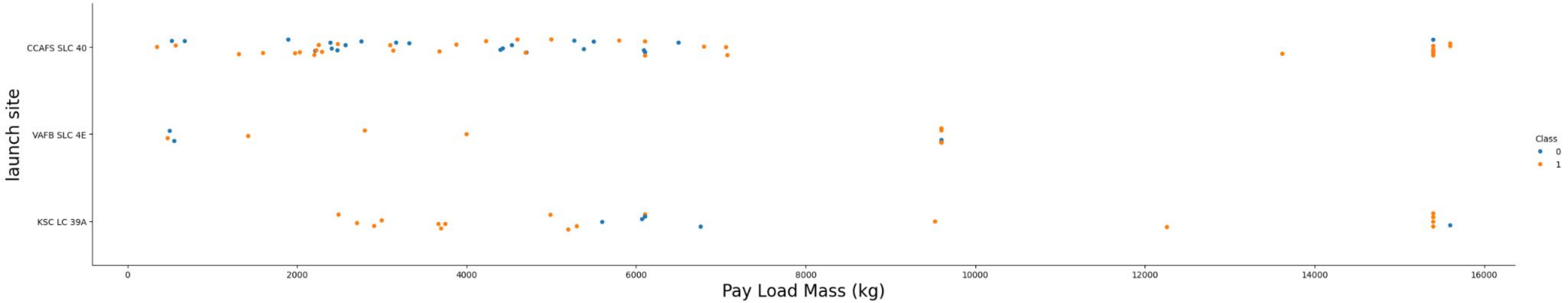
Flight Number vs. Launch Site

- The scatter plot visualizes the relationship between the Flight Number and the Launch Site, with the hue indicating whether the launch was a success or not. Each point on the plot represents a launch, with the x-axis showing the Flight Number and the y-axis showing the Launch Site. The hue distinguishes between successful and unsuccessful launches.



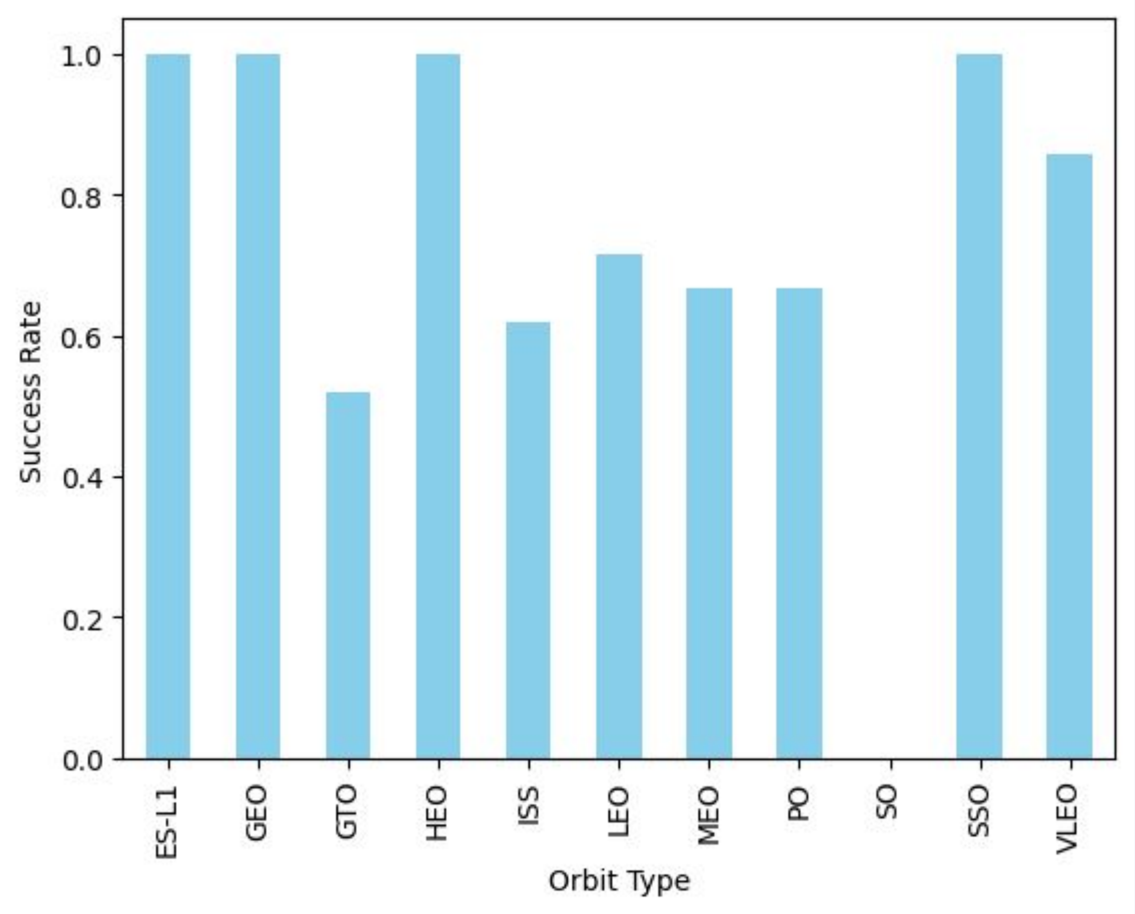
Payload vs. Launch Site

- This scatter plot visualizes the relationship between the Payload Mass and the Launch Site, with the hue indicating whether the launch was a success or not. Each point on the plot represents a launch, with the x-axis showing the Payload Mass (in kilograms) and the y-axis showing the Launch Site. The hue distinguishes between successful and unsuccessful launches.



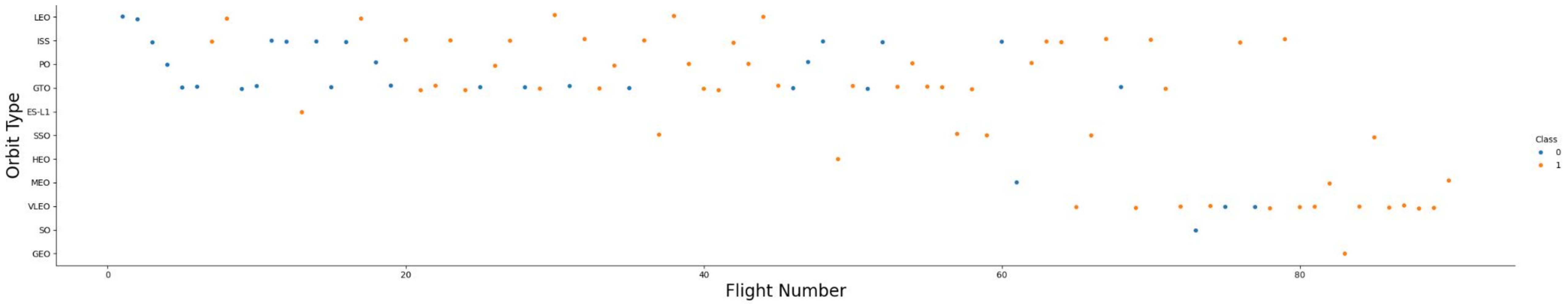
Success Rate vs. Orbit Type

- This bar plot shows the success rate of each orbit type, indicating the proportion of successful launches for each orbit. The x-axis represents the different orbit types, and the y-axis represents the success rate (ranging from 0 to 1, where 1 indicates 100% success).
- The height of each bar indicates the success rate for the corresponding orbit type. A higher bar indicates a higher success rate for that orbit type, while a lower bar indicates a lower success rate.



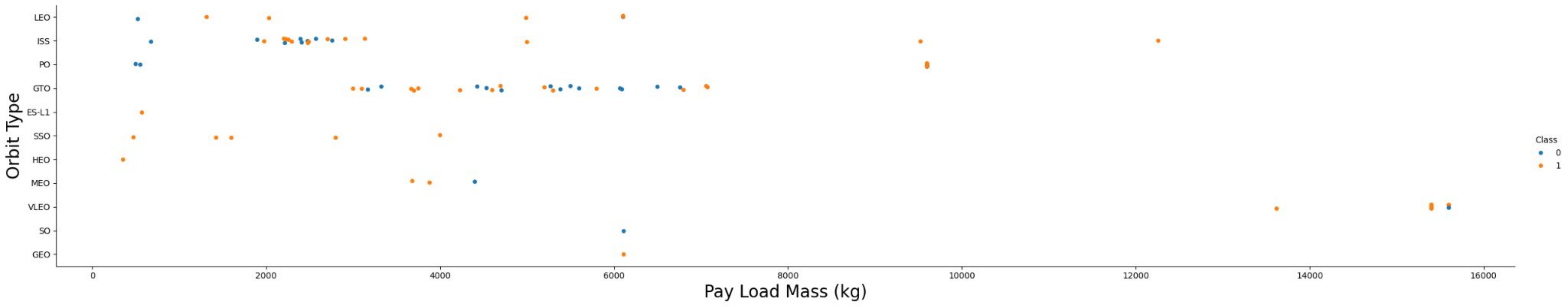
Flight Number vs. Orbit Type

- This scatter plot visualizes the relationship between the Flight Number and the Orbit Type, with the hue indicating whether the launch was a success or not. Each point on the plot represents a launch, with the x-axis showing the Flight Number and the y-axis showing the Orbit Type. The hue distinguishes between successful and unsuccessful launches.



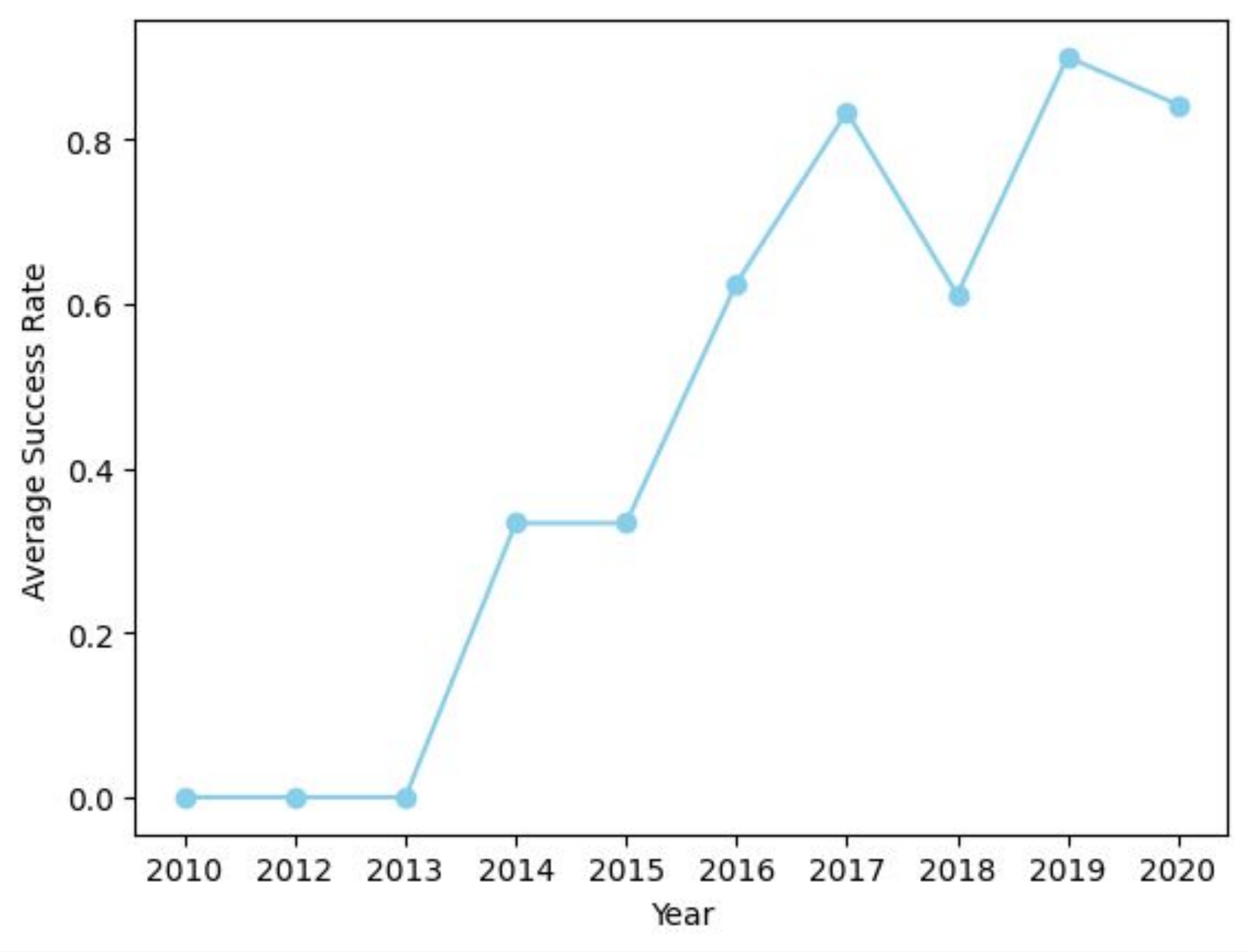
Payload vs. Orbit Type

- This scatter plot visualizes the relationship between the Payload Mass and the Orbit Type, with the hue indicating whether the launch was a success or not. Each point on the plot represents a launch, with the x-axis showing the Payload Mass (in kilograms) and the y-axis showing the Orbit Type. The hue distinguishes between successful and unsuccessful launches.



Launch Success Yearly Trend

This line plot visualizes the average launch success rate over the years. Each point on the plot represents a year, with the x-axis showing the year and the y-axis showing the average success rate. The average success rate is calculated by taking the mean of the success values (1 for success, 0 for failure) for each year.



All Launch Site Names

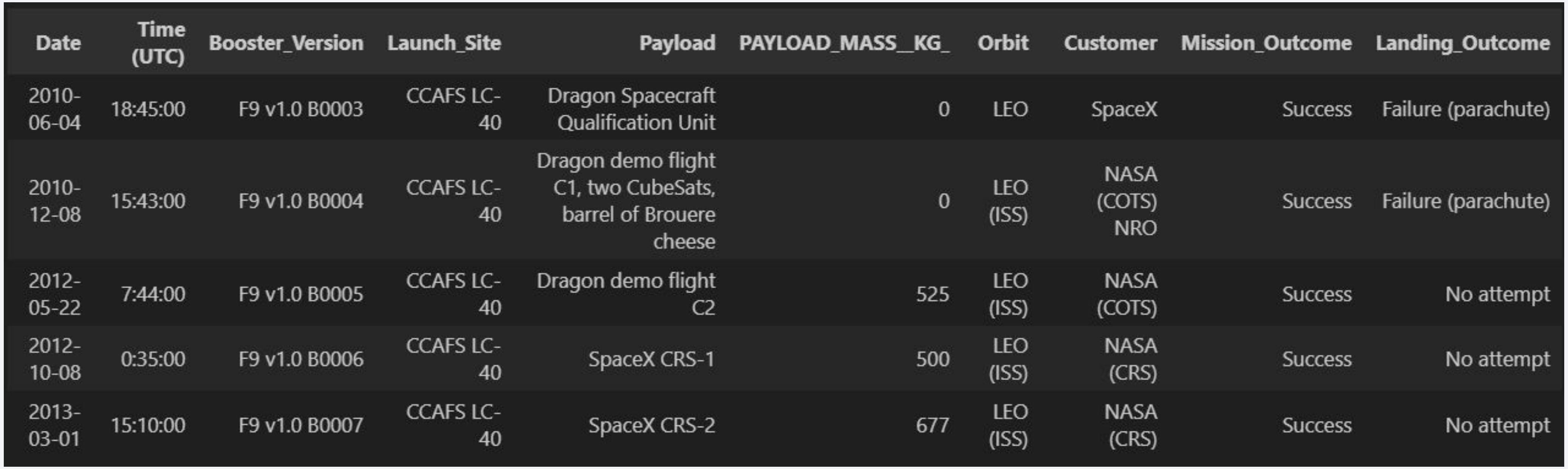
The query “***select DISTINCT Launch_Site from SPACEXTABLE***” retrieves the unique values in the 'Launch Site' column. The result of the query is a list of unique launch sites:

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

These are the unique launch sites from which SpaceX has conducted launches.

Launch Site Names Begin with 'CCA'

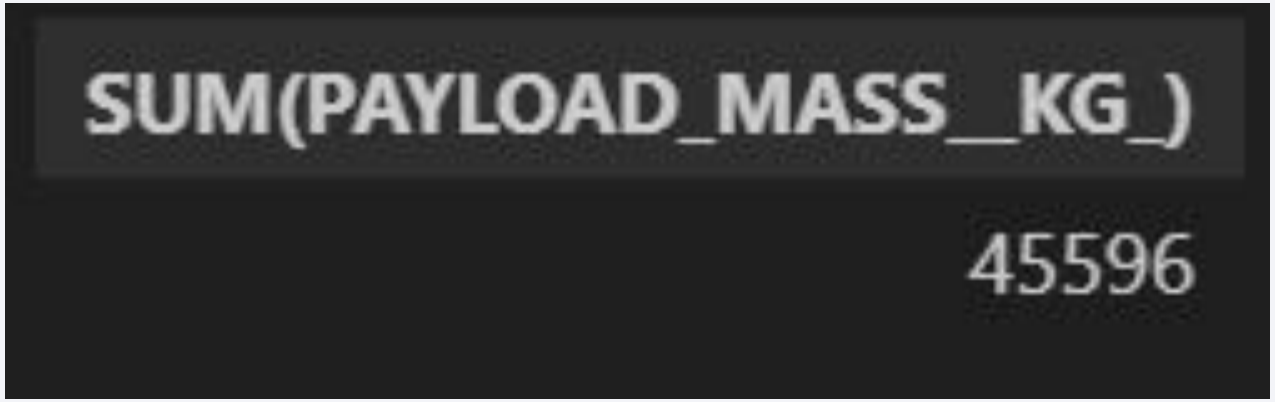
The query ***%sql select * from SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5*** retrieves the first 5 rows from the SPACEXTABLE where the Launch_Site column starts with 'CCA'. The result includes columns for Date, Time (UTC), Booster Version, Launch Site, Payload, Payload Mass (kg), Orbit, Customer, Mission Outcome, and Landing Outcome.



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

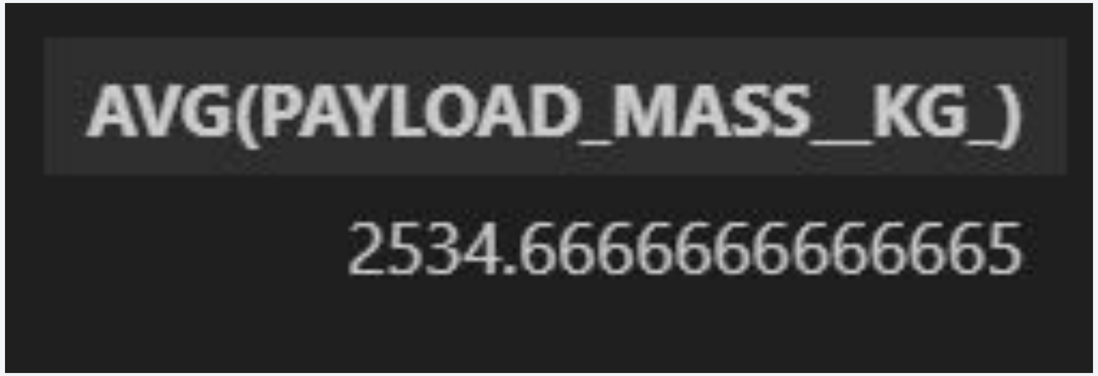
- The query ***%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Customer = 'NASA (CRS)'*** calculates the total payload mass (in kilograms) carried by boosters launched by NASA under the CRS (Commercial Resupply Services) program.



SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

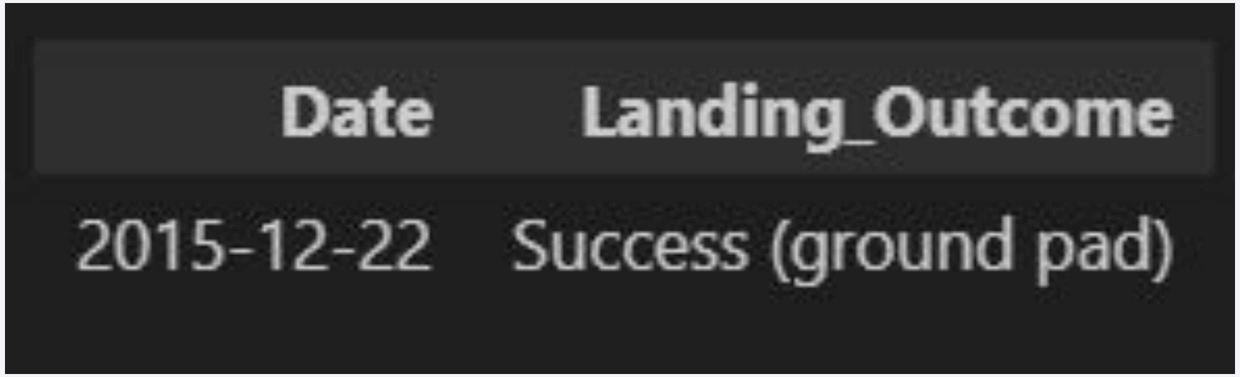
- The query ***%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'*** calculates the average payload mass (in kilograms) carried by boosters of version F9 v1.1.



AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

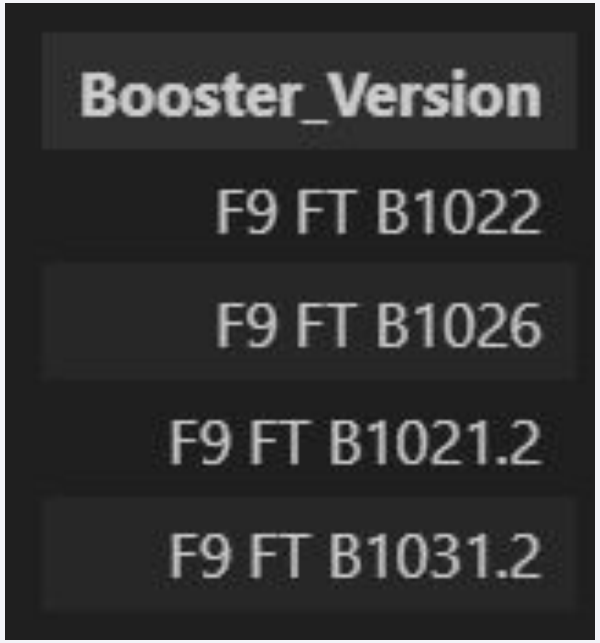
- The query ***%sql select Date, Landing_Outcome from SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)' ORDER BY Date LIMIT 1*** retrieves the date of the first successful landing outcome achieved on a ground pad.



Date	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

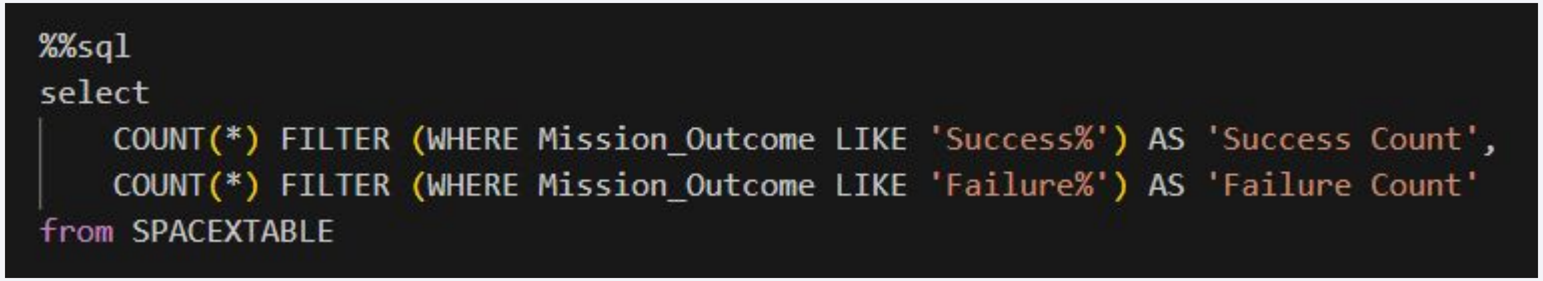
- The query ***%sql select Booster_Version from SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000*** retrieves the names of the boosters that have achieved success in landing on a drone ship and have a payload mass greater than 4000 kilograms but less than 6000 kilograms.



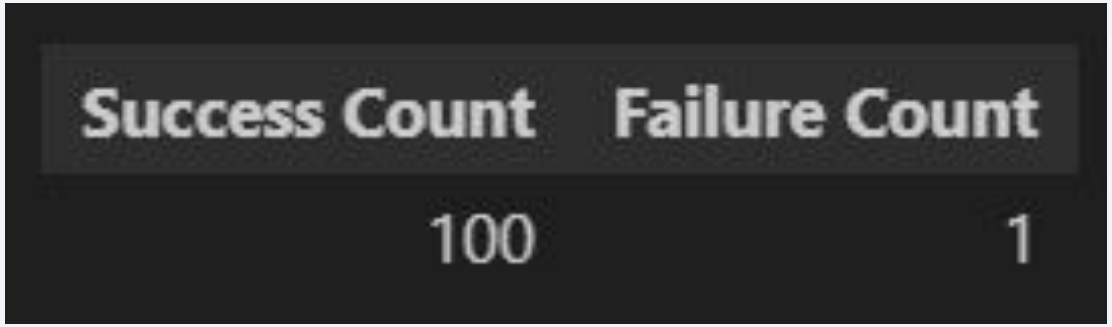
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The query uses filters to count the total number of successful and failed mission outcomes in the dataset.



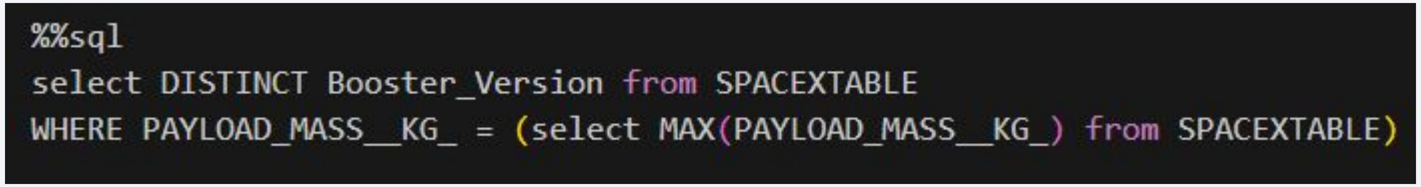
```
%%sql
select
    COUNT(*) FILTER (WHERE Mission_Outcome LIKE 'Success%') AS 'Success Count',
    COUNT(*) FILTER (WHERE Mission_Outcome LIKE 'Failure%') AS 'Failure Count'
from SPACEXTABLE
```



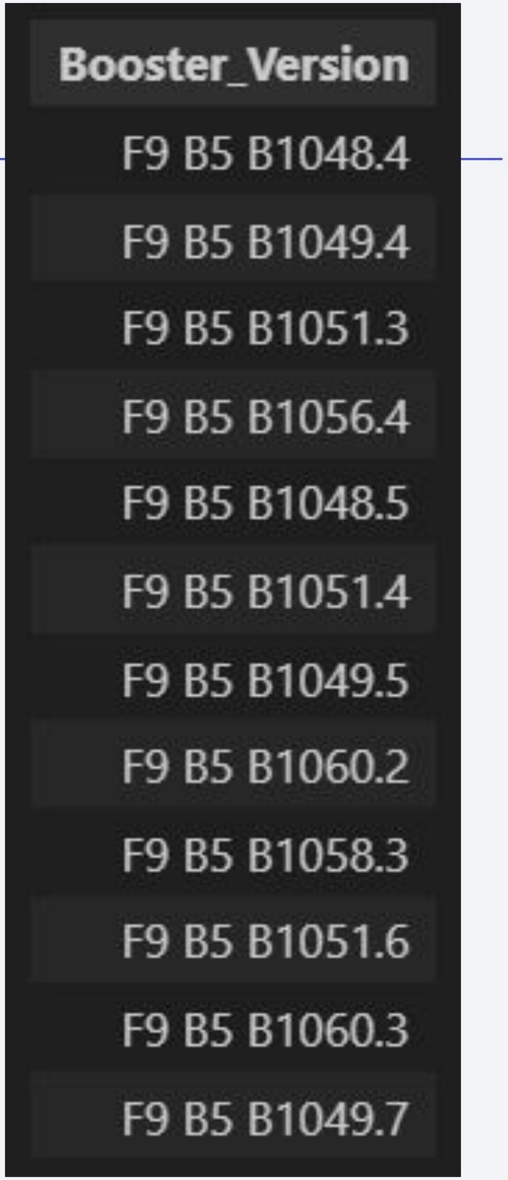
Success Count	Failure Count
100	1

Boosters Carried Maximum Payload

- The query uses a subquery to find the maximum payload mass in the dataset, and then retrieves the names of the booster versions that have carried this maximum payload mass.



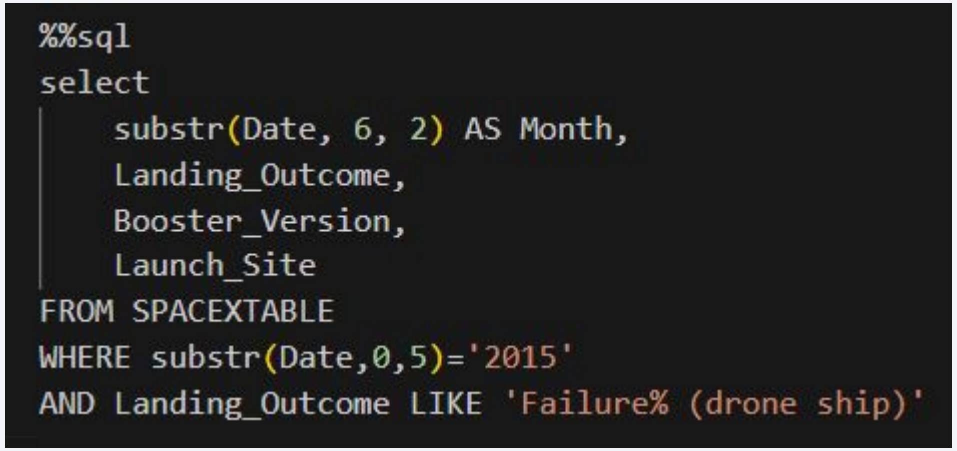
```
%%sql
select DISTINCT Booster_Version from SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```



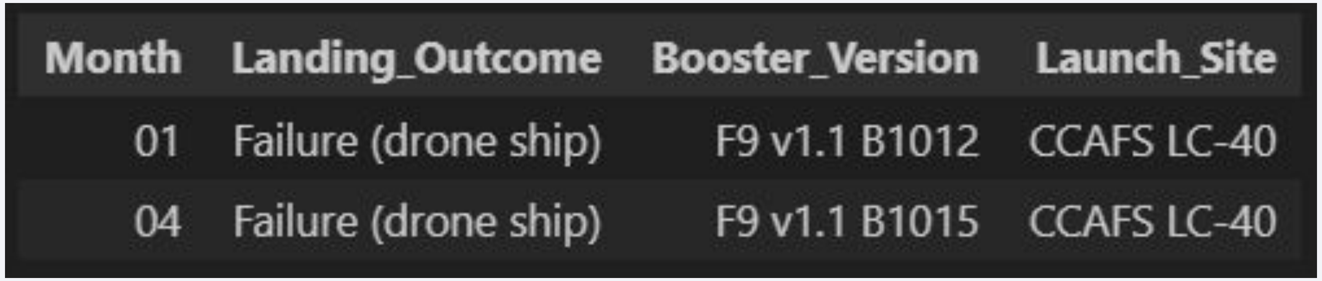
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- The query retrieves records for the months in the year 2015 that have a failure landing outcome on a drone ship. It displays the month names, failure landing outcomes, booster versions, and launch sites for these records.



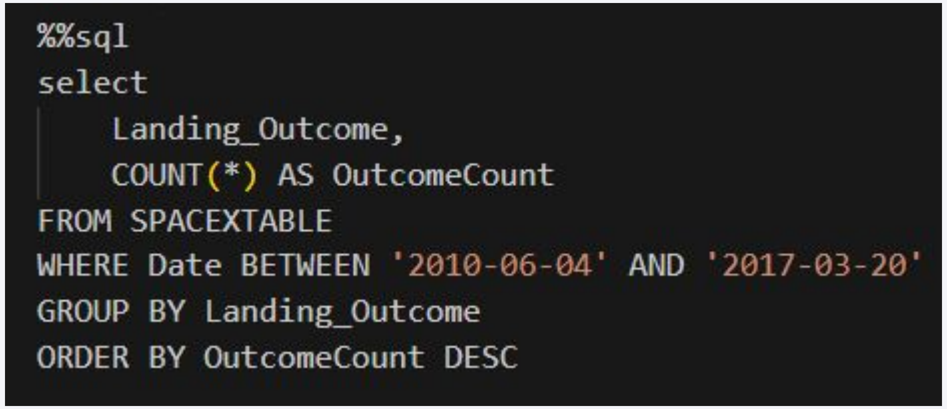
```
%%sql
select
    substr(Date, 6, 2) AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE substr(Date,0,5)='2015'
AND Landing_Outcome LIKE 'Failure% (drone ship)'
```



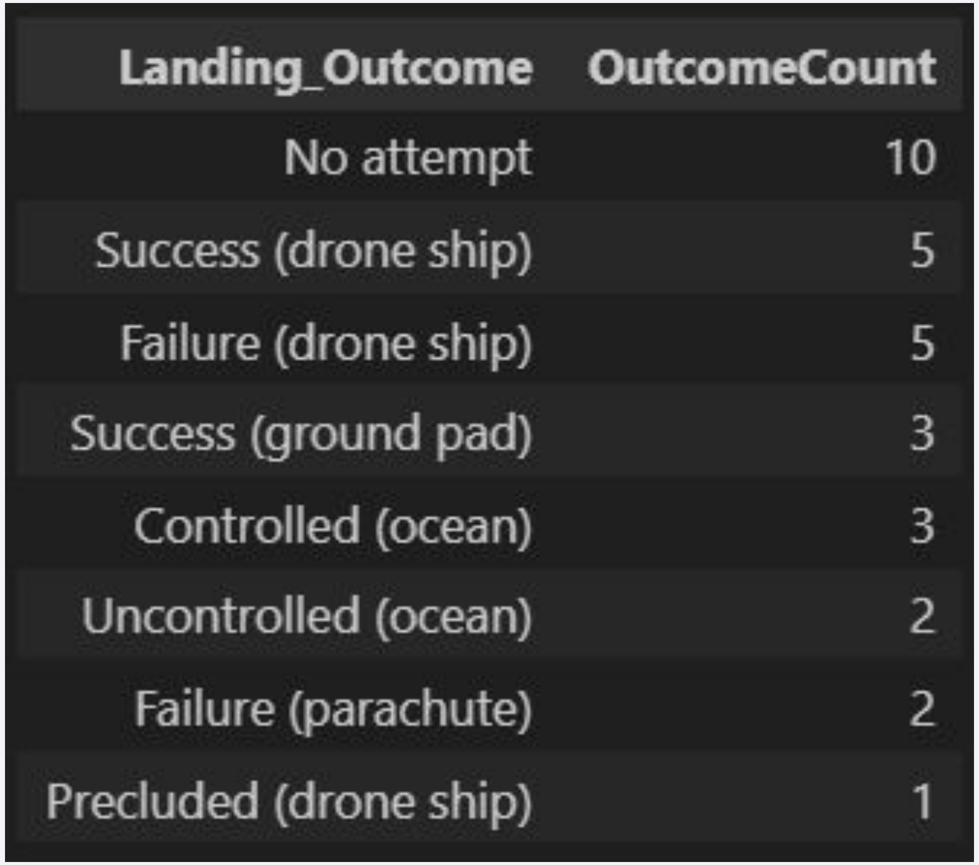
Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The query retrieves the count of different landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, and ranks them in descending order based on the count.



```
%%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
```



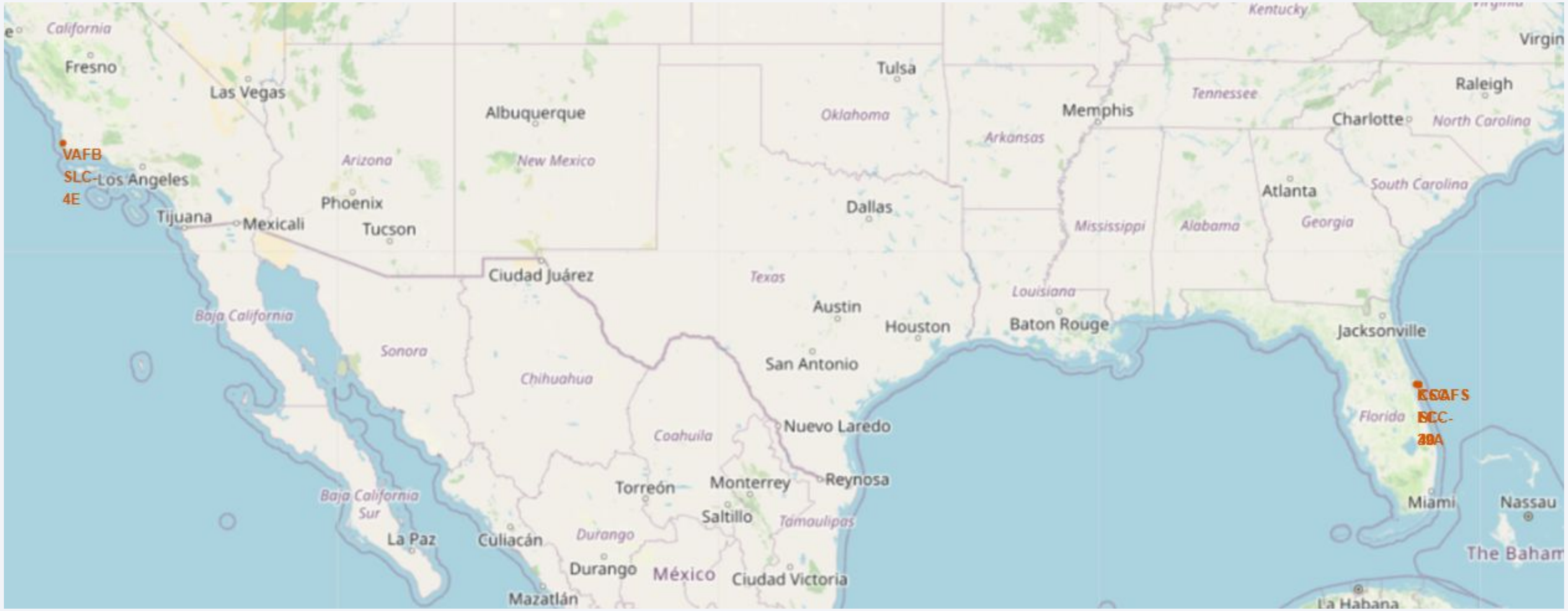
Landing_Outcome	OutcomeCount
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

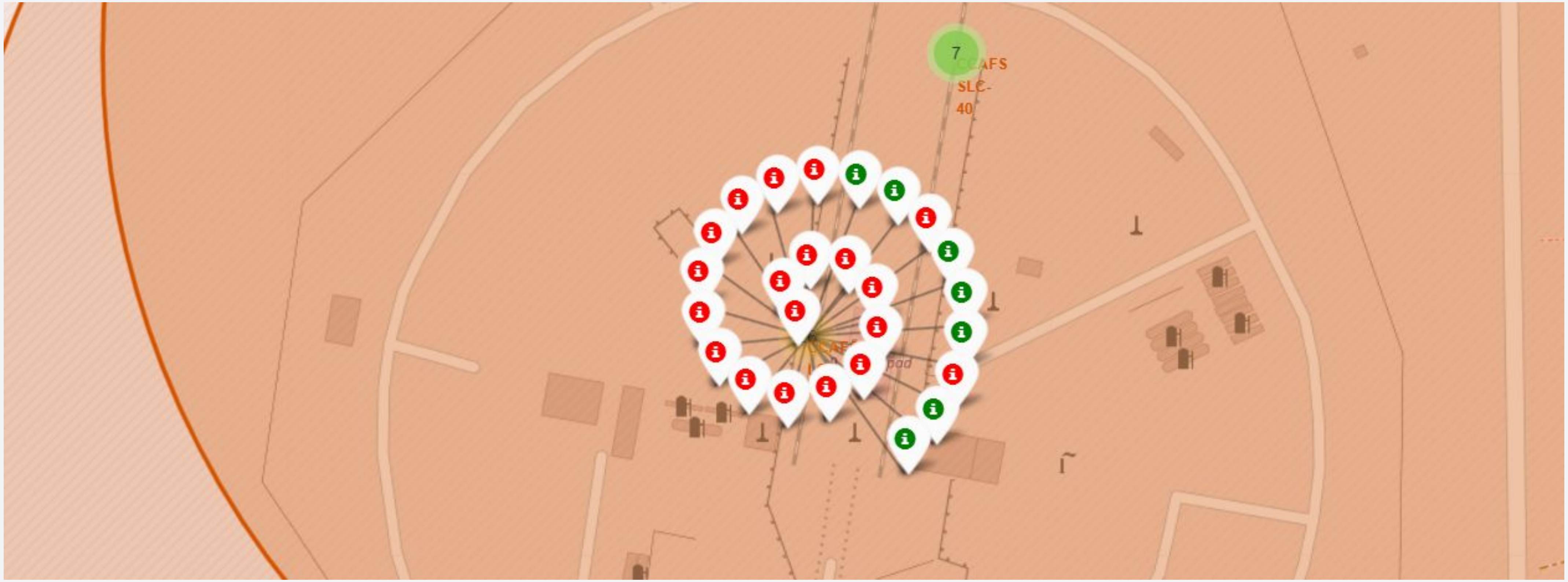
SpaceX Launch Sites

- The map displays the locations of SpaceX launch sites as circles on a map. Each circle represents a launch site, with the circle's size indicating the area around the launch site.



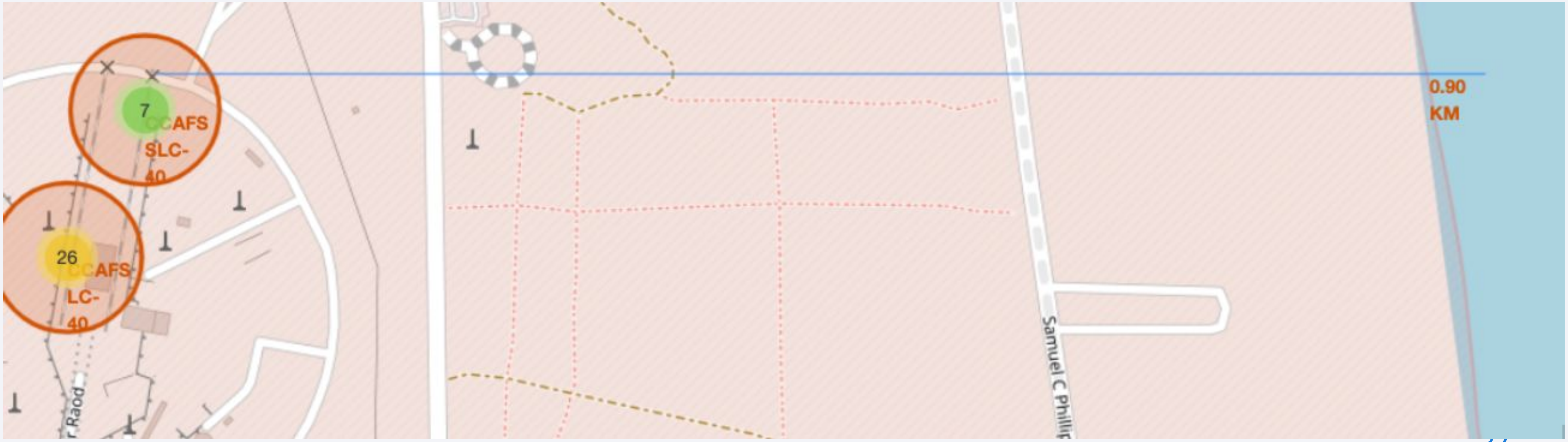
SpaceX Launch Sites with Success/Failure launch markers

- The map shows the locations of SpaceX launch sites, with markers indicating the outcomes of each launch. Green markers represent successful launches, while red markers represent failed launches.



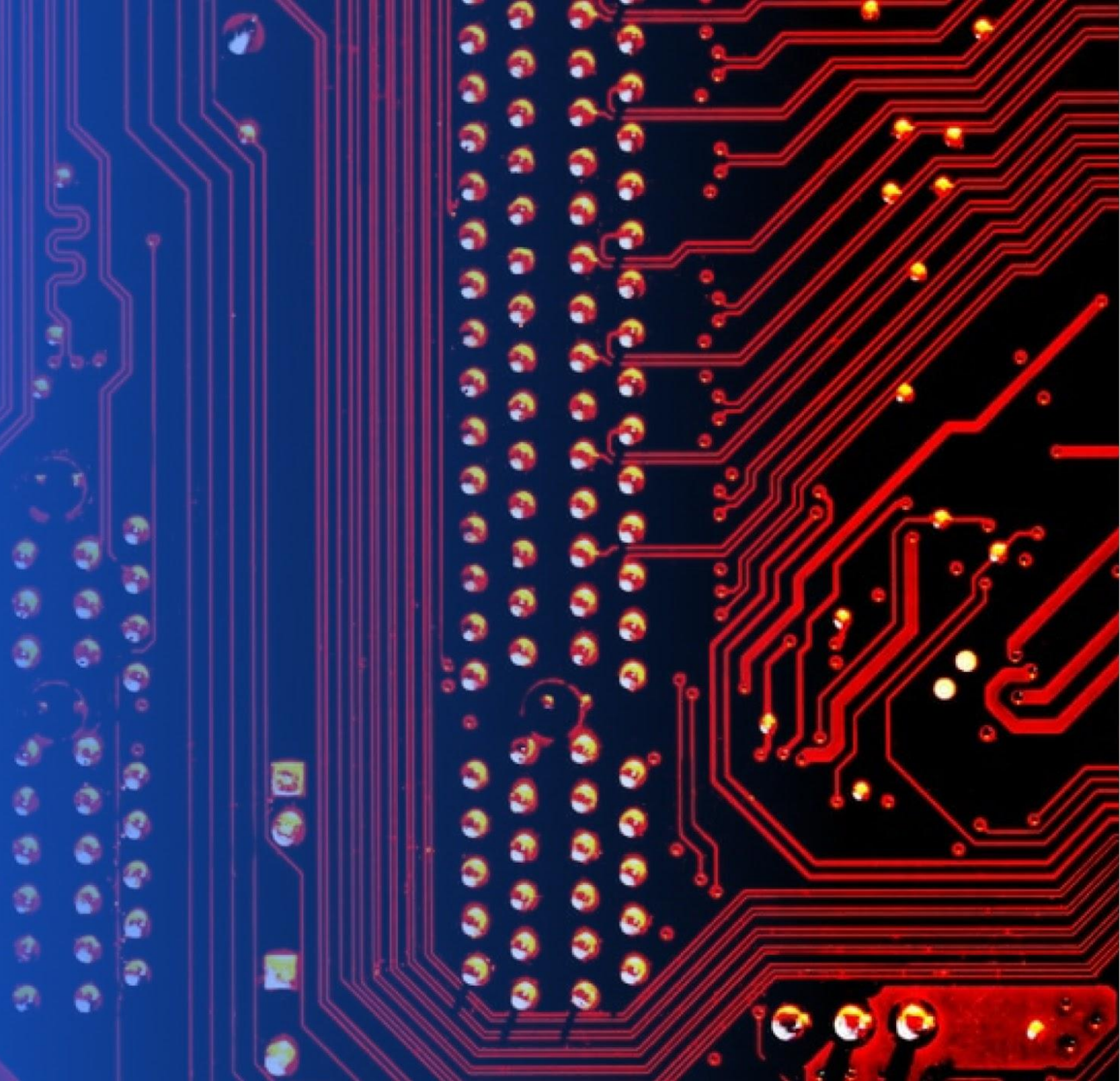
SpaceX Launch Site and Closest Coastline with Distance Measurement

- The map shows the location of a SpaceX launch site and the closest coastline point, along with the distance between them. The coastline point is marked with a marker displaying the distance in kilometers from the launch site.



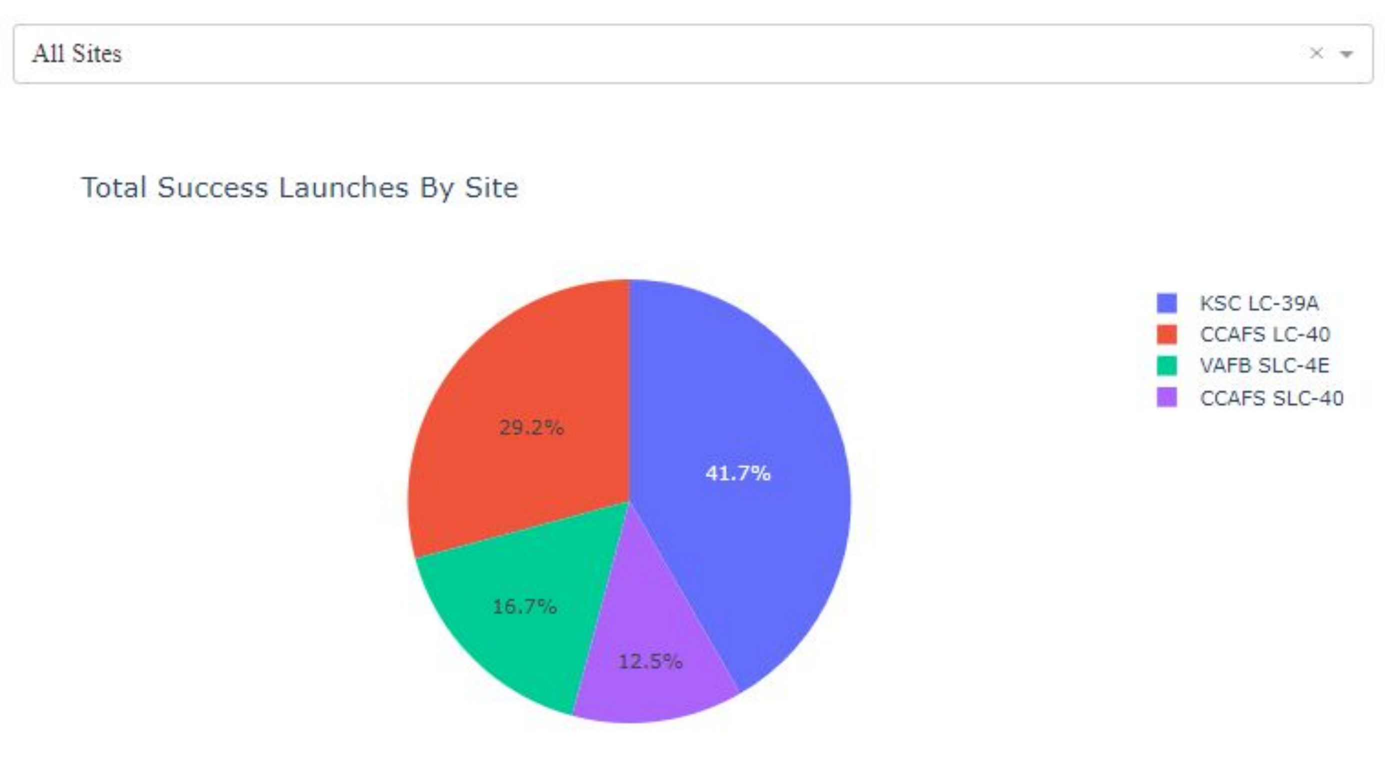
Section 4

Build a Dashboard with Plotly Dash



Distribution of Successful Launches Across SpaceX Launch Sites

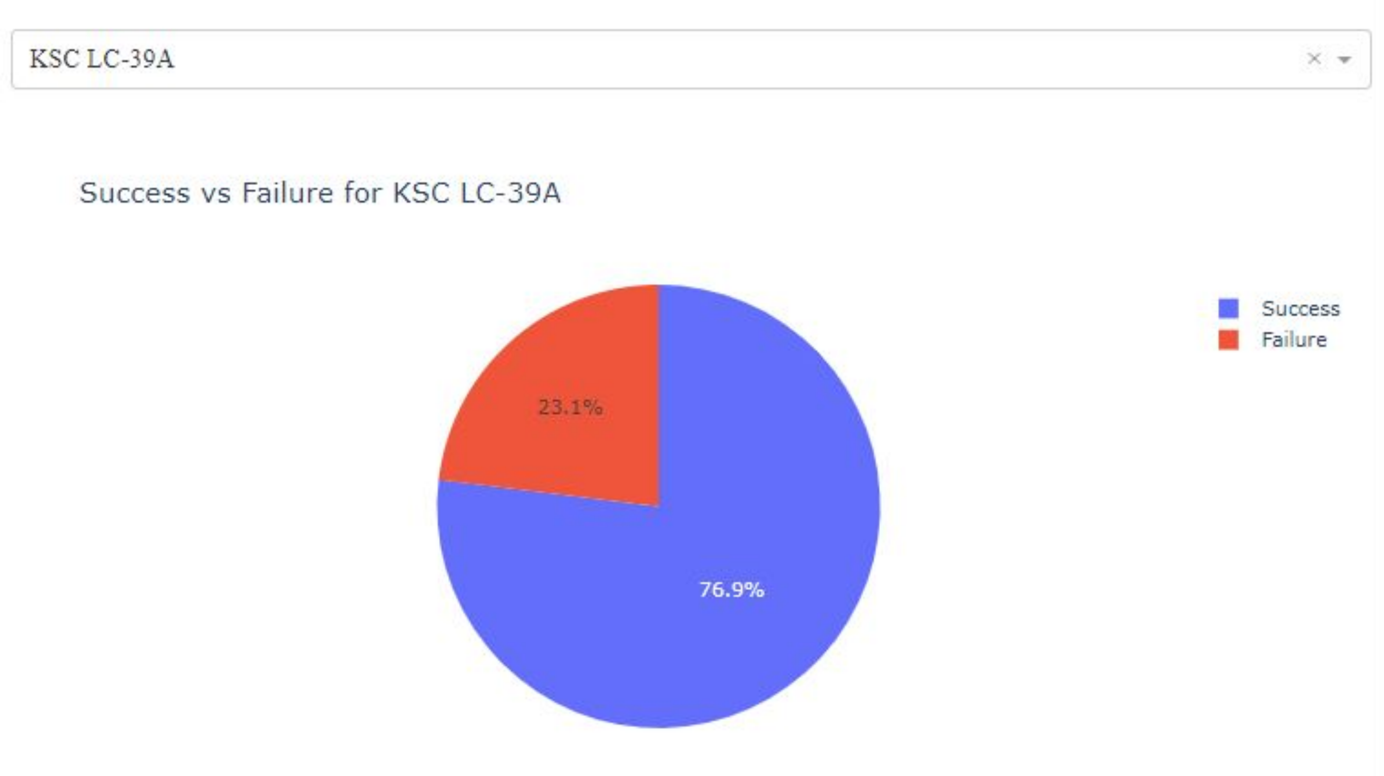
- The pie chart displays the distribution of successful launches across all SpaceX launch sites. Each segment of the pie represents a launch site, with the size of the segment indicating the proportion of successful launches at that site relative to the total number of successful launches. This visualization provides an overview of the success rates at different launch sites



All Sites

Success vs. Failure Launches at KSC LC-39A

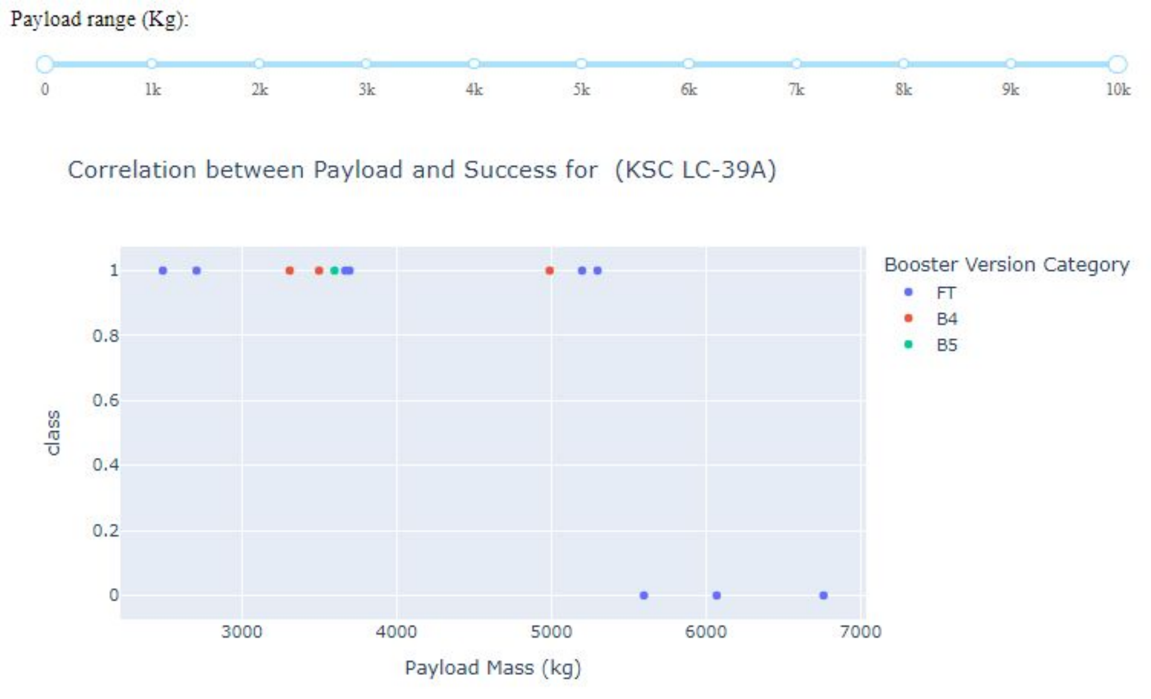
- The pie chart displays the proportion of successful and failed launches at the Kennedy Space Center Launch Complex 39A (KSC LC-39A). Each segment of the pie represents either a successful or failed launch, illustrating the success rate of missions at this specific launch site. This visualization provides insights into the performance of KSC LC-39A in terms of successful mission outcomes.



KSC LC-39A

Correlation between Payload Mass and Launch Outcome for SpaceX Launches

- The scatter plot shows the correlation between the payload mass and the launch outcome (success or failure) for SpaceX launches at all sites. Each point on the plot represents a launch, with the x-axis indicating the payload mass in kilograms and the y-axis indicating the launch outcome. The range slider allows you to select a specific range of payload masses to analyze. This visualization helps to understand if there is any relationship between the payload mass and the success of a launch.

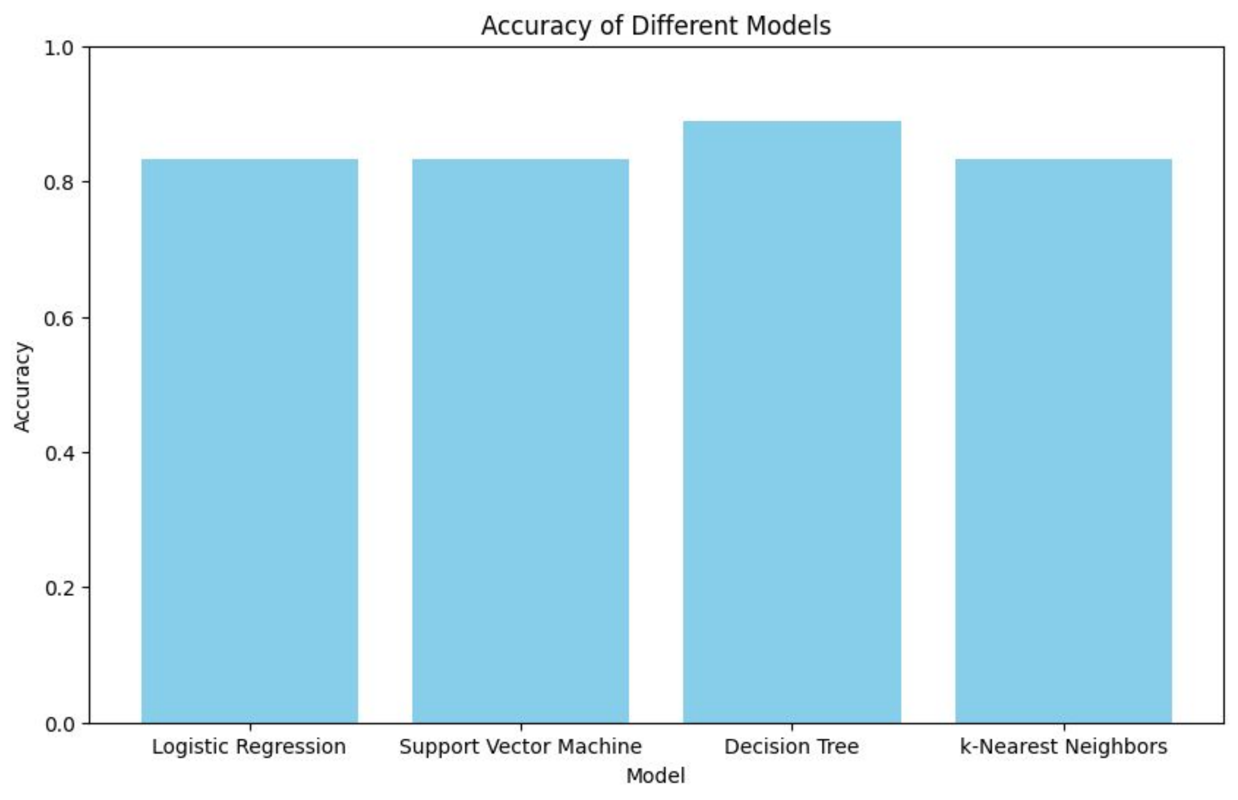


Section 5

Predictive Analysis (Classification)

Classification Accuracy

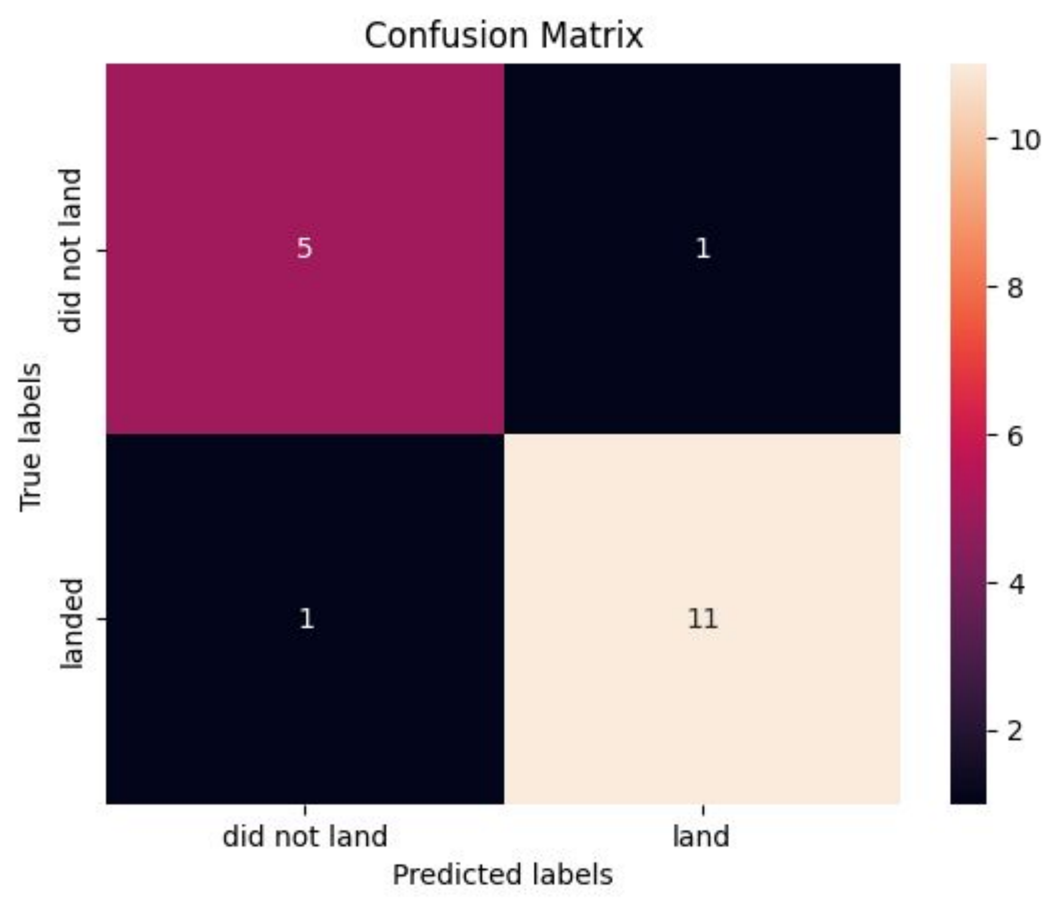
- Based on the results shown in the bar plot, the classification model with the highest classification accuracy is the Decision Tree. The Decision Tree model achieved an accuracy of approximately 0.89, which is slightly higher than the accuracy achieved by the other models. This indicates that the Decision Tree model performed the best among the models compared in terms of accurately predicting the classes in the dataset.



Confusion Matrix

Based on the confusion matrix:

- The model correctly predicted that 11 launches would land and that 5 launches would not land.
- However, the model incorrectly predicted that 1 launch would land when it did not, and that 1 launch would not land when it did.
- Overall, the model seems to perform reasonably well, correctly predicting the outcomes for the majority of launches. However, the misclassifications indicate that there is still room for improvement. Further analysis and possibly model refinement may be needed to improve its accuracy.



Conclusions

- Data Collection and Wrangling: Data was successfully collected from the SpaceX API and underwent thorough wrangling to address missing values and incorrect data types, ensuring the dataset's quality.
- Exploratory Data Analysis (EDA): EDA revealed interesting insights into SpaceX launch outcomes, including the distribution of successful and failed launches across different launch sites and orbits. Visualizations provided a clear understanding of launch success rates and trends over time.
- Predictive Analysis: The project utilized various classification models, including Logistic Regression, SVM, Decision Tree, and KNN, to predict launch outcomes. These models were evaluated based on their accuracy, with the Decision Tree model showing the highest accuracy among them.
- Importance of Launch Site: The choice of launch site appears to significantly impact launch success, as evidenced by the analysis of success rates at different sites.
- Need for Further Refinement: While the models showed decent accuracy, there is room for further refinement and improvement, particularly in distinguishing between the two classes (landed and did not land).
- Overall Insights: The project has provided valuable insights into SpaceX launch outcomes, highlighting the importance of data analytics and machine learning in optimizing mission planning and enhancing operational efficiency in the aerospace industry.

Appendix

- All the code used in this project, including data collection from the SpaceX API, data wrangling, exploratory data analysis, interactive visual analytics, and predictive analysis using classification models, can be found in the following

- GitHub repository: https://github.com/AmonApolonio/IBM-Data-Science-Professional-Certificate/tree/main/Applied%20Data%20Science%20Capstone

Thank you!