IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kevin Castillo

March 20, 2025

Outline

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

Executive Summary

Methodologies

- Data Analysis & SQL: Extracted insights on launch success, payloads, and booster versions.
- Geospatial Mapping: Visualized launch sites, success markers, and proximity to infrastructure.
- Interactive Dashboard: Built with Plotly Dash to analyze launch trends dynamically.
- Machine Learning: Trained Logistic Regression, SVM, Decision Tree, and KNN to predict launch success.

Key Results

- KSC LC-39A had the highest success rate, followed by CCAFS LC-40.
- Mid-range payloads (4000-6000 kg) had better success rates, with certain boosters performing reliably.
- Logistic Regression and Decision Tree achieved 83.33% accuracy, making both the best models for predicting launch success.

Introduction

Background & Context

- SpaceX has revolutionized space travel with reusable rockets and frequent launches. Understanding factors influencing mission success is crucial for optimizing future launches. This project analyzes launch data, explores patterns, and applies machine learning to predict mission outcomes.

Key Problems & Questions

- Which launch sites have the highest success rates?
- How does payload mass impact launch success?
- Which booster versions perform best?
- Can machine learning accurately predict launch outcomes?

Goal: Identify trends, optimize launch strategies, and improve SpaceX's mission success.

Section 1

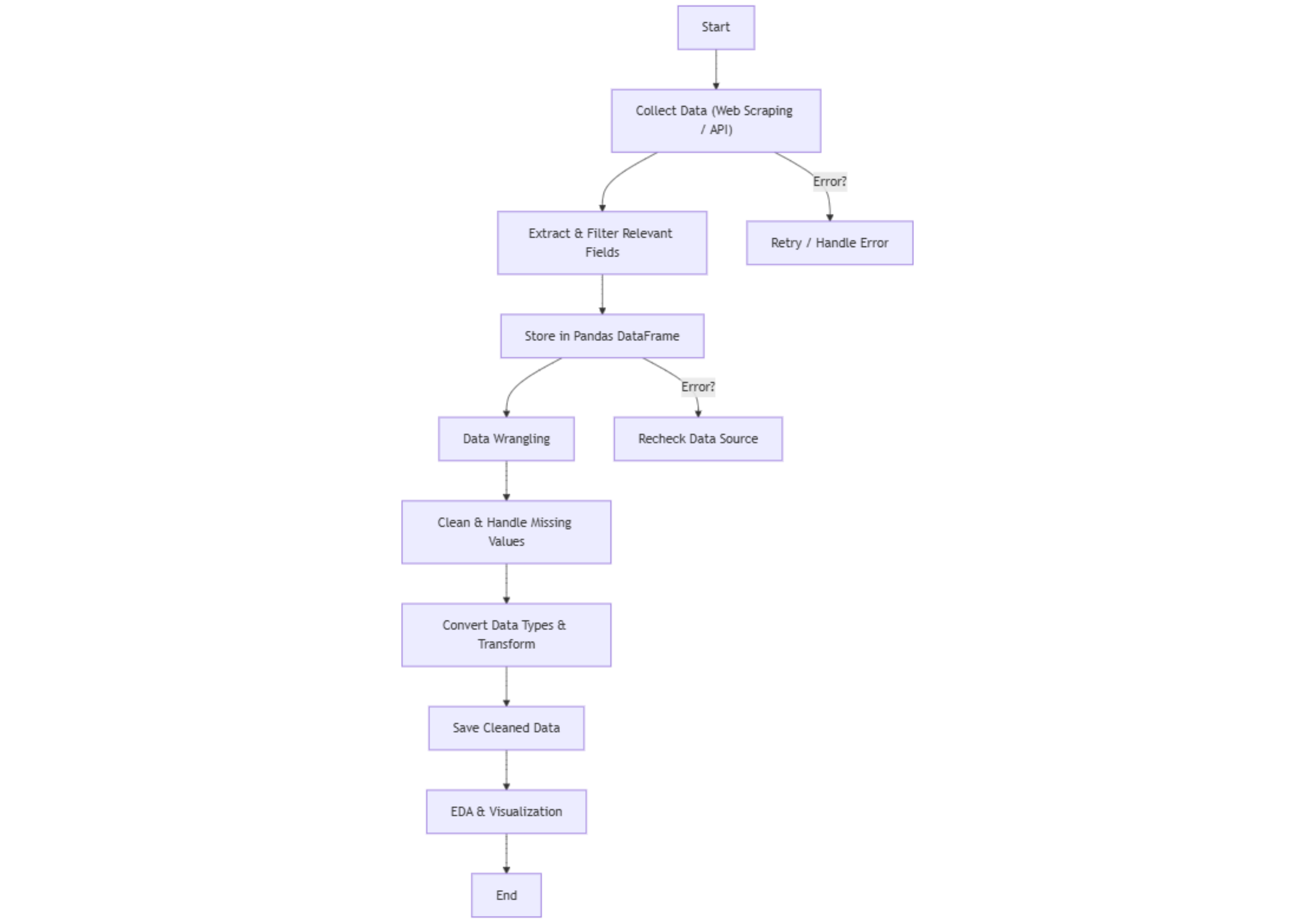
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The dataset was sourced from SpaceX launch records, containing details on launch sites, booster versions, payload mass, mission outcomes, and landing results.
- Perform data wrangling
  - Performed data wrangling to clean missing values, standardize formats, and filter relevant features for analysis. SQL queries were used for structured data exploration.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Implemented Logistic Regression, SVM, Decision Tree, and KNN to classify mission success, hyperparameter tuning and evaluation.

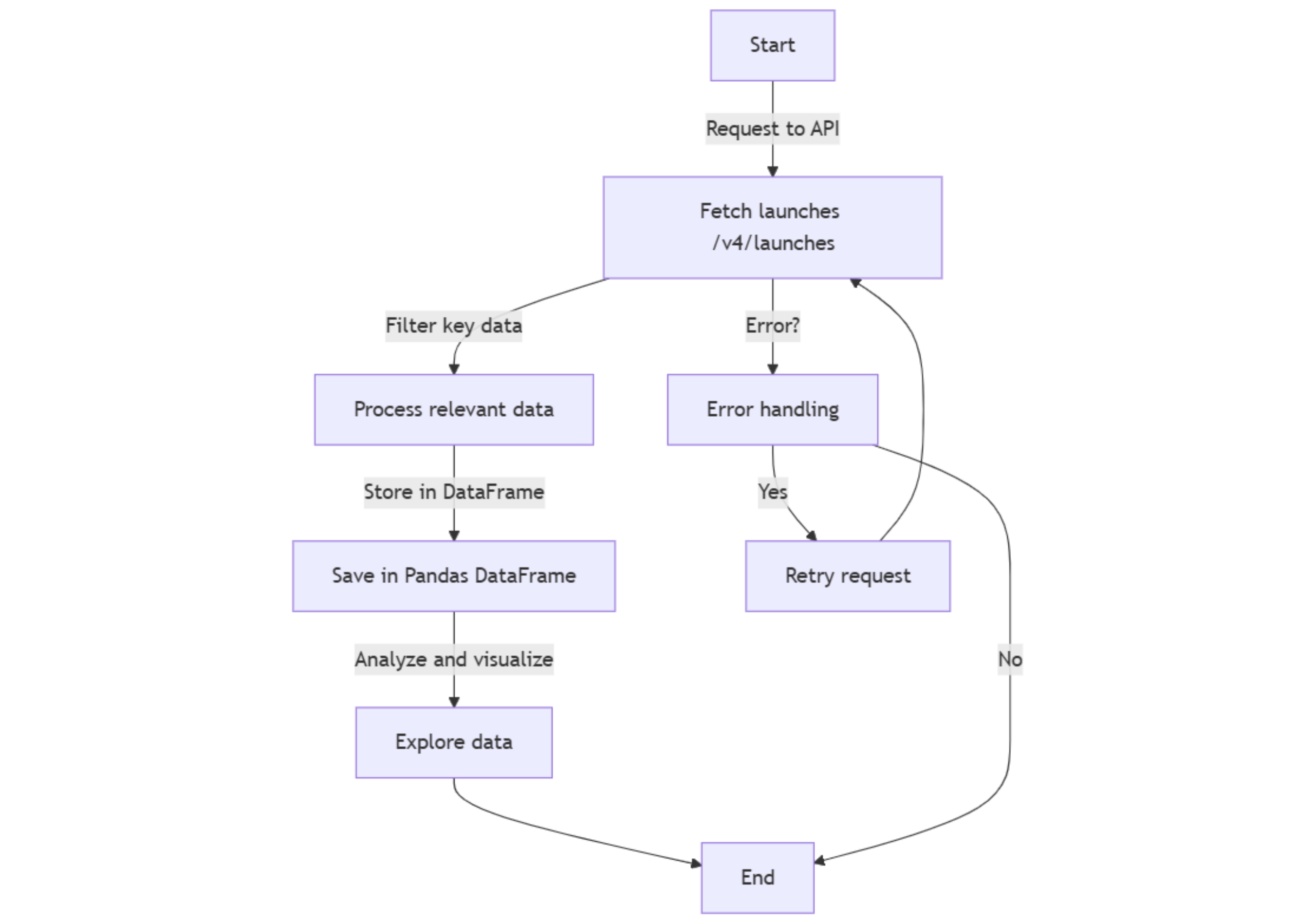
Data Collection


Start
Collect Data (Web Scraping / API)
Error?
Extract & Filter Relevant Fields
Retry / Handle Error
Store in Pandas DataFrame
Error?
Data Wrangling
Recheck Data Source
Clean & Handle Missing Values
Convert Data Types & Transform
Save Cleaned Data
EDA & Visualization
End

- Web Scraping: Extracted data from websites, parsed relevant fields, stored it in a Pandas DataFrame, applied cleaning steps, and performed exploratory data analysis (EDA).
- API Retrieval: Queried the SpaceX API to fetch launch data, filtered key fields, stored it, and analyzed it using Python visualization tools.
- Data Wrangling: Loaded raw data into a Pandas DataFrame, handled missing values, converted data types, applied transformations, and saved the cleaned dataset for further analysis.

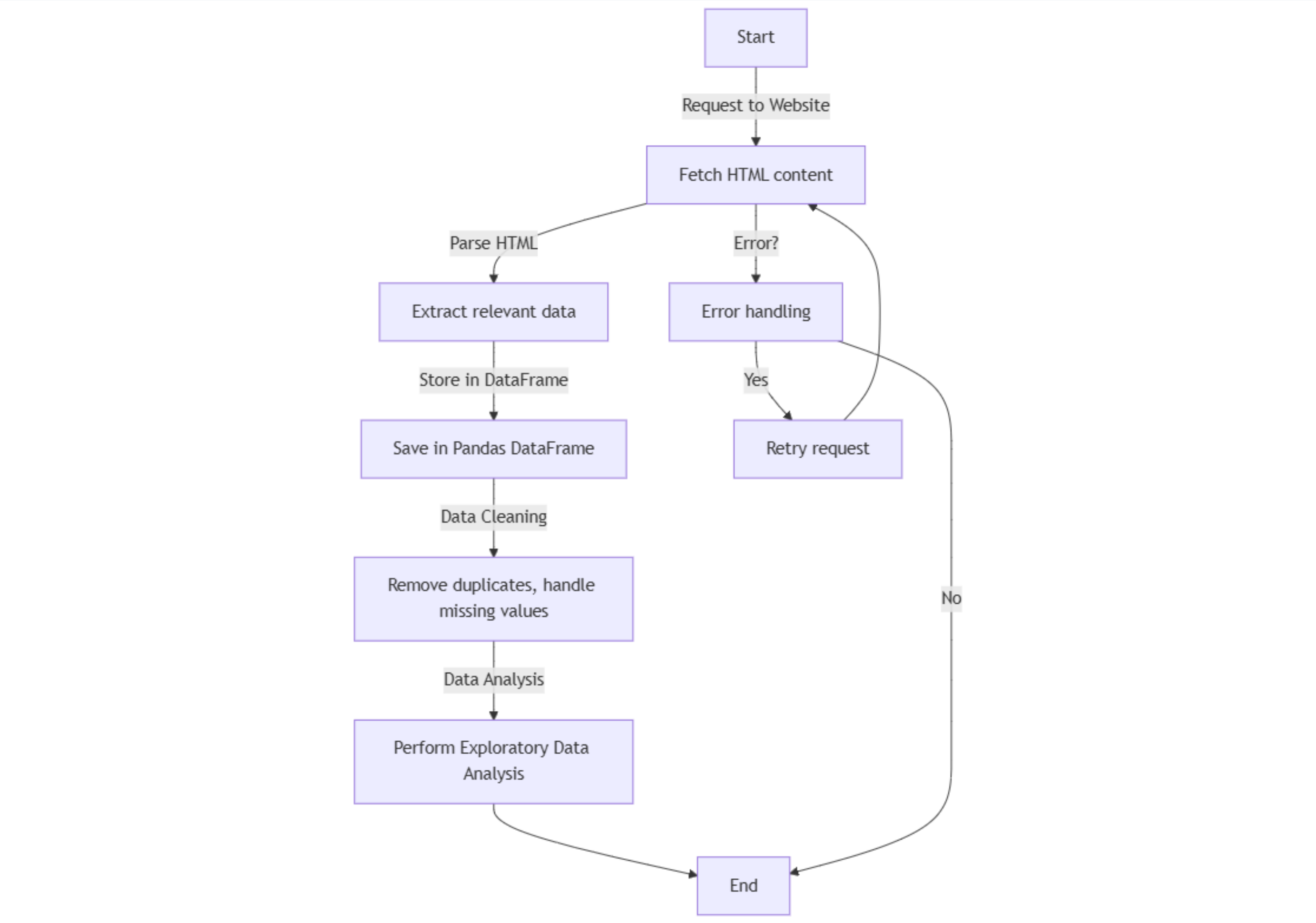
Data Collection – SpaceX API

- This flowchart illustrates the data collection process using the SpaceX API. The process begins with an API request to retrieve launch data, followed by filtering relevant information and storing it in a Pandas DataFrame. If an error occurs, a retry mechanism is triggered.
- Finally, the collected data is analyzed and visualized for further insights. We used this notebook: Data Collection API.



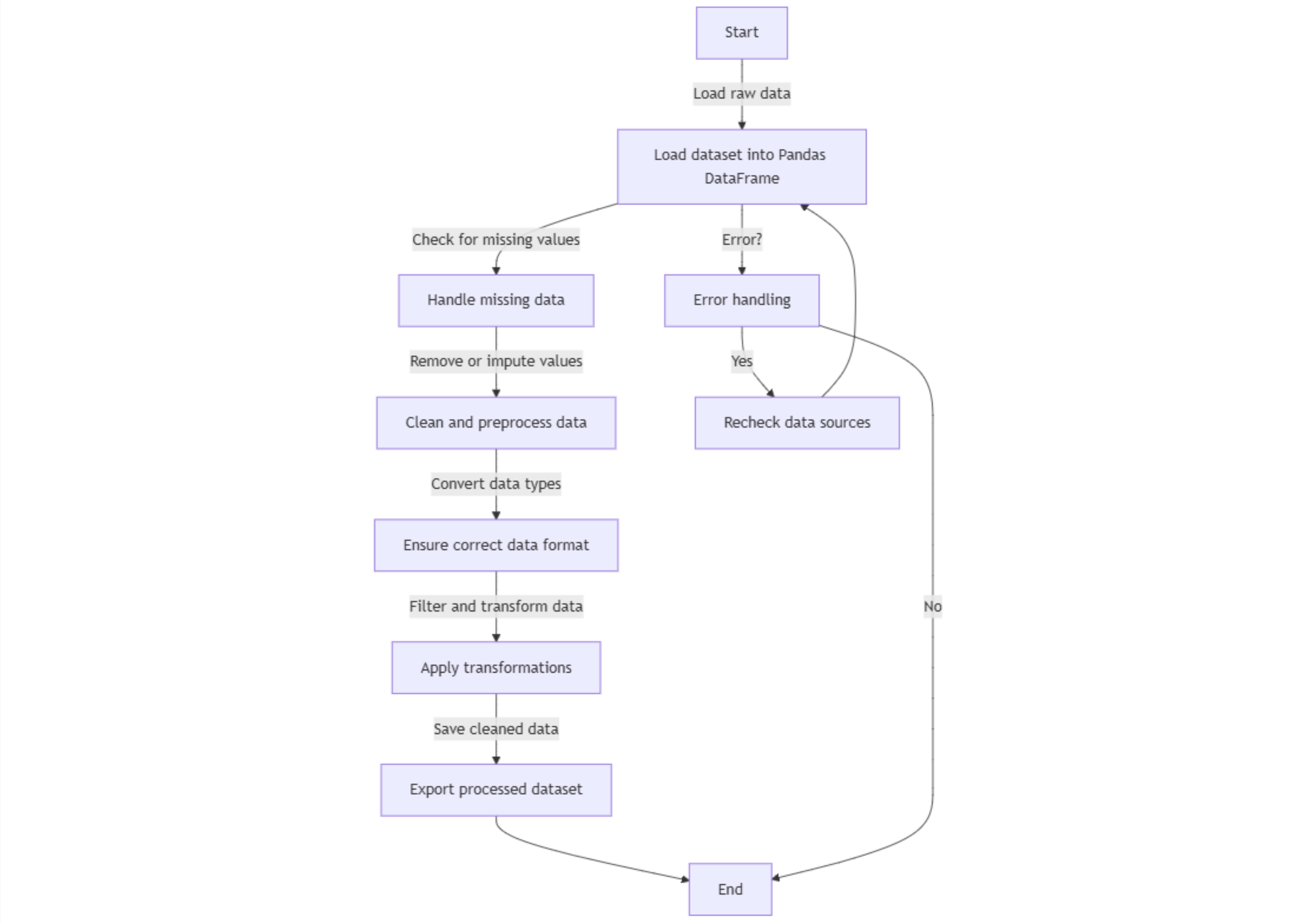
Data Collection - Scraping

- The flowchart outlines a web scraping process: fetch content, parse HTML, store and clean data in a DataFrame, and analyze insights. Errors are handled with a retry mechanism.
- We used this notebook: Data Collection Scraping.

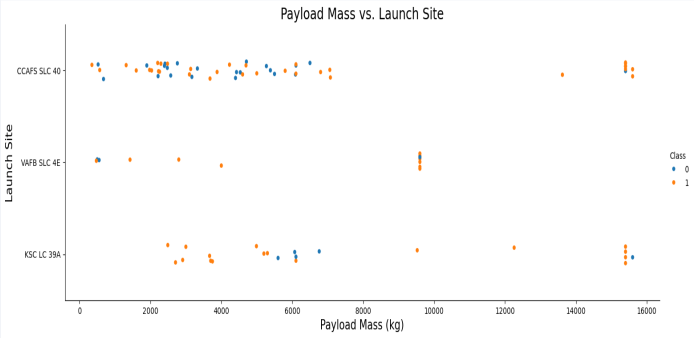


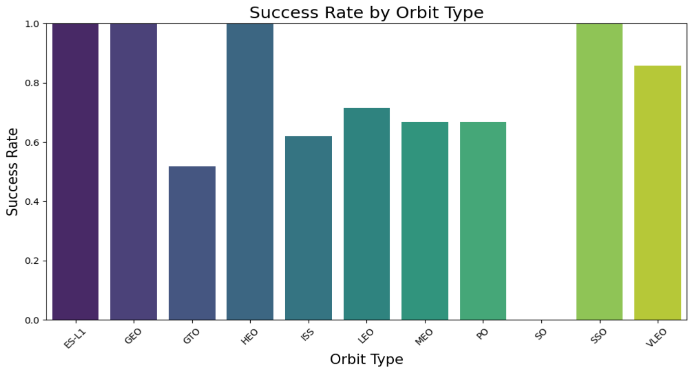
Data Wrangling

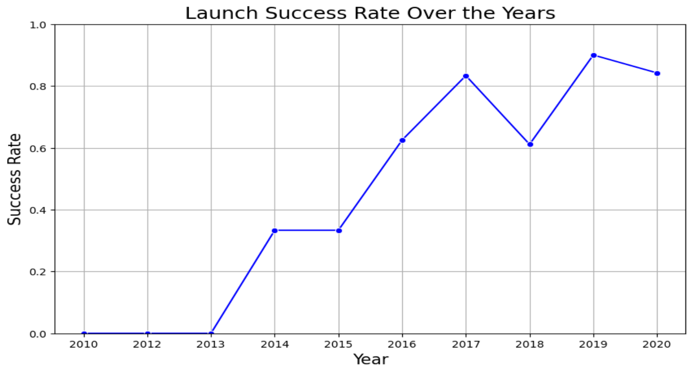
- This flowchart shows how SpaceX data is cleaned and prepared: load data, drop unnecessary columns, handle missing values, encode variables, normalize, merge datasets, and save for analysis.
- We did this process with this notebook: Data Wrangling



EDA with Data Visualization







- We used scatter plots, bar charts and line charts because they help us to: identify trends in launch success; understand the impact of payload mass, launch site, and orbit type; and see how SpaceX improved its technology over time.
- We completed EDA with data visualization notebook here: EDA with data visualization

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Created a new table (filtering out rows where Date is null)
  - Retrieved distinct launch sites (Extracted a list of unique launch sites)
  - Filtered launches from specific sites (Retrieved the first 5 launches from sites that start with CCA)
  - Summed payload mass for NASA (CRS) launches (Calculated the total payload mass for NASA (CRS) missions)
  - Calculated average payload mass for Falcon 9 v1.1 missions
- We completed EDA with SQL notebook here: EDA with SQL

Build an Interactive Map with Folium

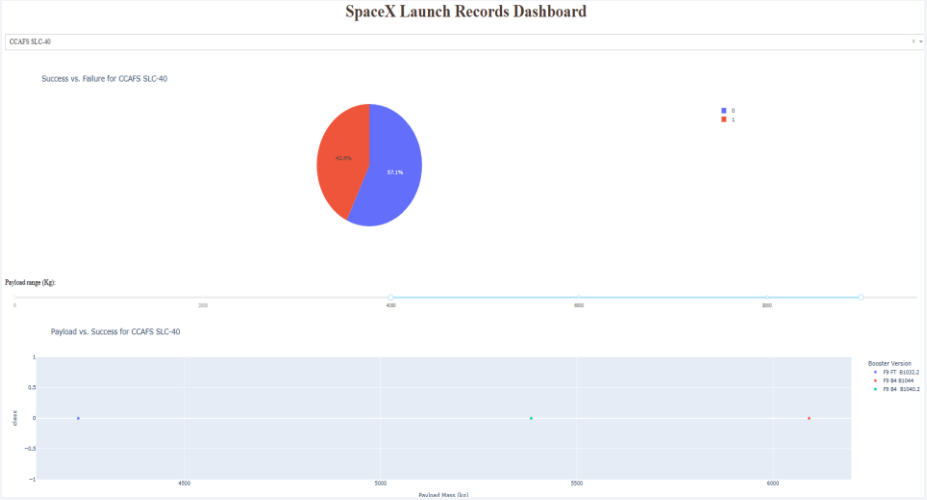
We used the following elements:

- Markers & Labels (folium.Marker)
  - Helped visualize key locations (launch sites, infrastructure, success/failures).
  - Allowed interactive popups showing distances to important locations.
- Circles (folium.Circle)
  - Provided visual emphasis on launch sites and key infrastructure.
- Lines (folium.PolyLine)
  - Allowed distance analysis between launch sites and coastal areas, cities, roads, and railways.
  - Helped answer proximity-related questions about infrastructure and safety considerations.

We completed interactive map with Folium map here: Interactive map

Build a Dashboard with Plotly Dash

- Pie Chart (plotly.express.pie)
  - Helps compare launch success rates across different sites.
  - Allows users to quickly analyze performance trends per launch site.
- Scatter Plot plotly.express.scatter)
  - Shows whether payload mass affects launch success.
  - Helps identify optimal payload ranges for successful missions.
- Dropdown Menu (dcc.Dropdown)
  - Provides an interactive way to filter launch site data.
  - Makes the dashboard user-friendly and dynamic.
- Range Slider (dcc.RangeSlider)
  - Helps focus analysis on specific payload sizes.
  - Allows users to observe launch success trends for different payloads.



SpaceX Launch Records Dashboard

CCAFS SLC-40

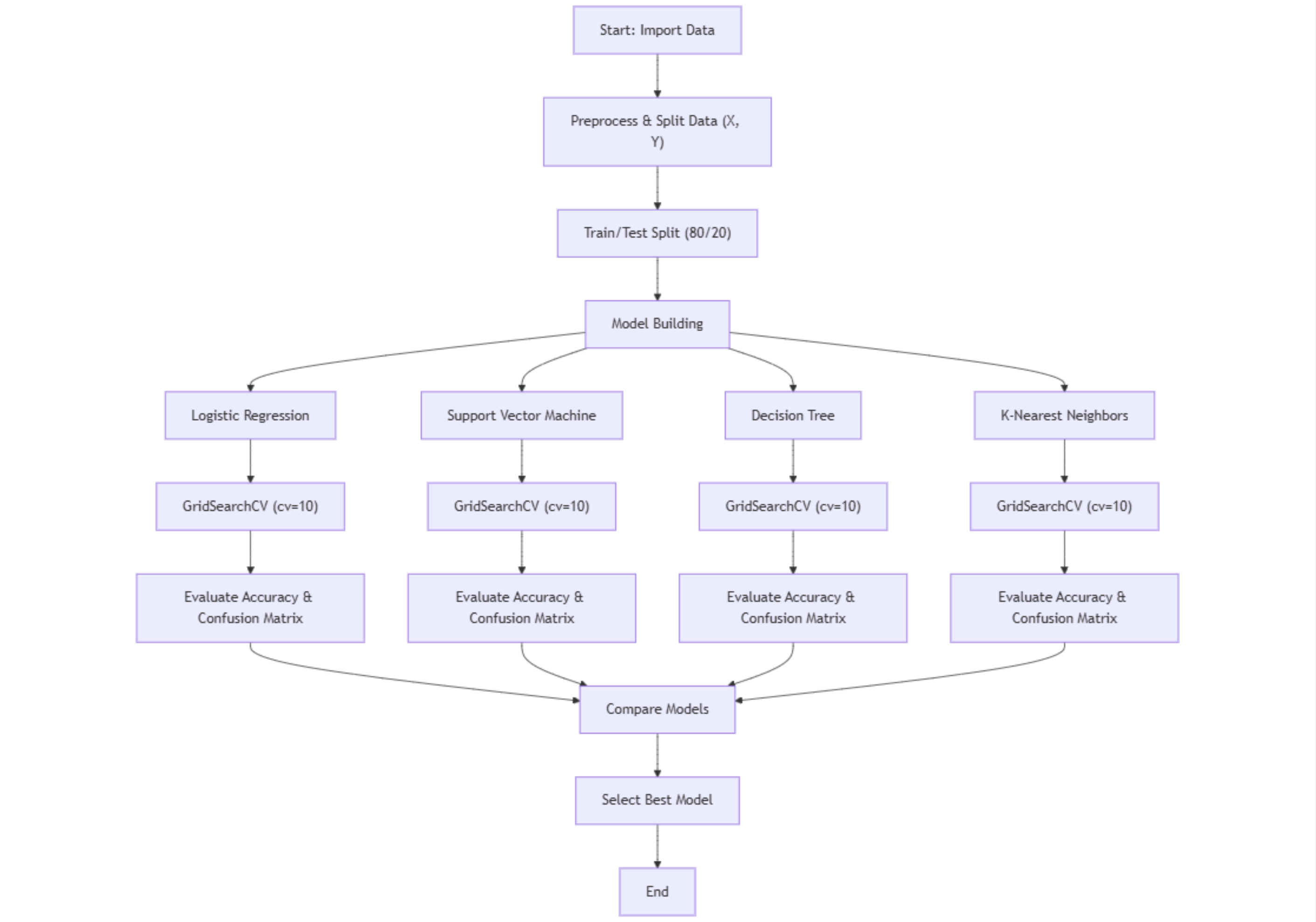
Payload range (Kg):

We completed Plotly Dash lab with this app: Plotly Dash

Predictive Analysis (Classification)

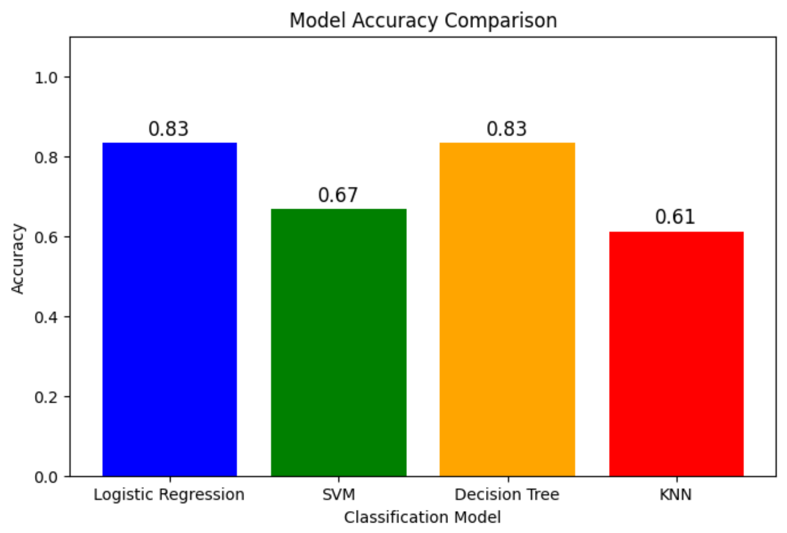
- Data Preparation
  - Imported and preprocessed the dataset.
  - Split the dataset into training and test sets.
- Model Building & Evaluation
  - Built and trained Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN) classifiers.
  - Find optimal hyperparameters for each model.
- Performance Evaluation
  - Evaluated each model using accuracy on the test..
  - Plotted confusion matrices.
- Model Comparison and Selection
  - Compared the test accuracies of all models.
  - Selected the model with the highest test accuracy as the best performer.

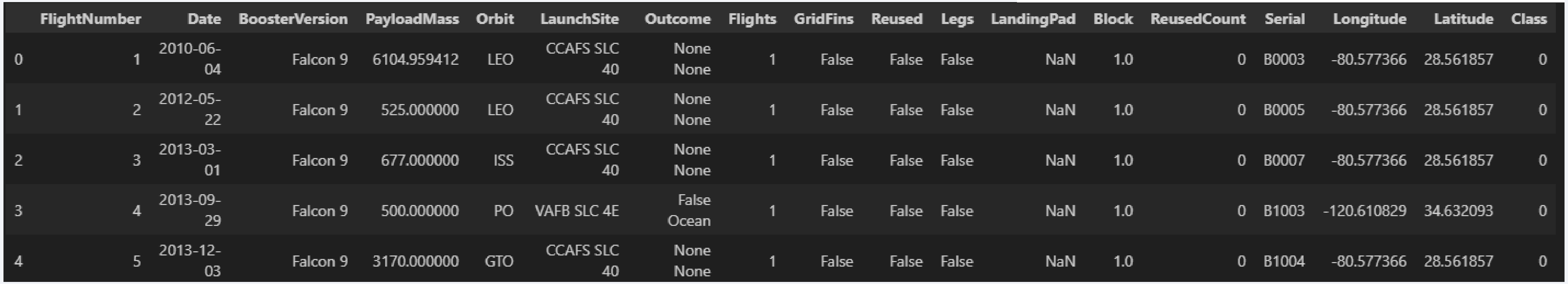
We completed predictive analysis lab here: [predictive analysis]()



Results

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



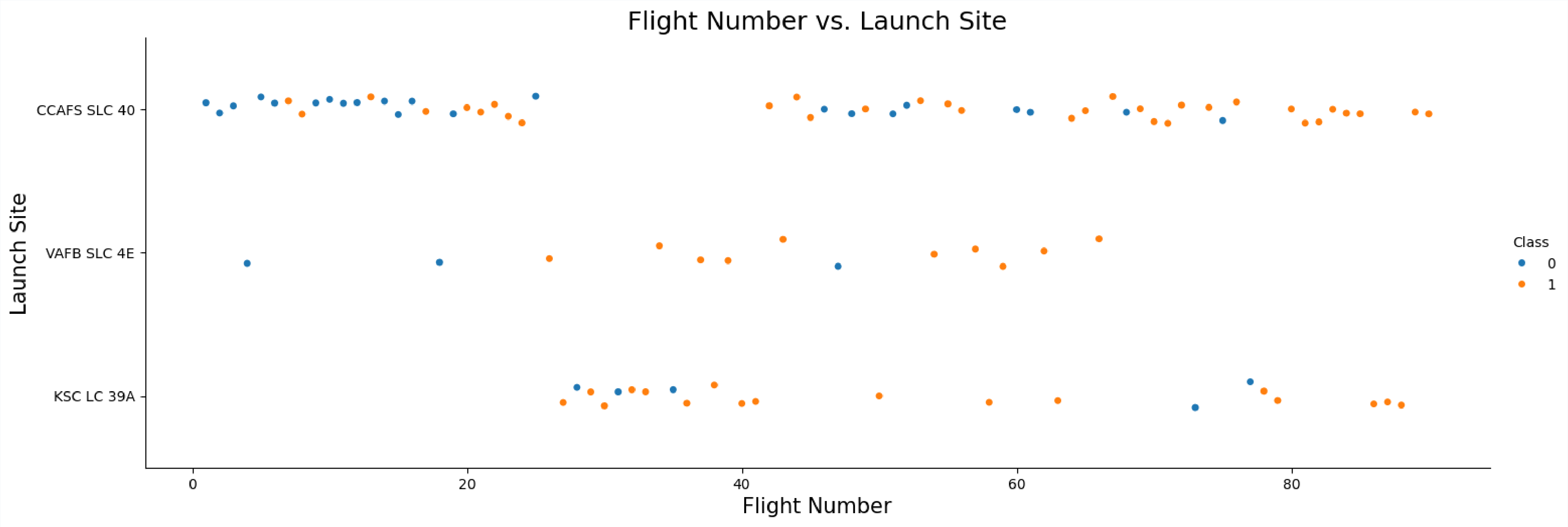


	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

Section 2

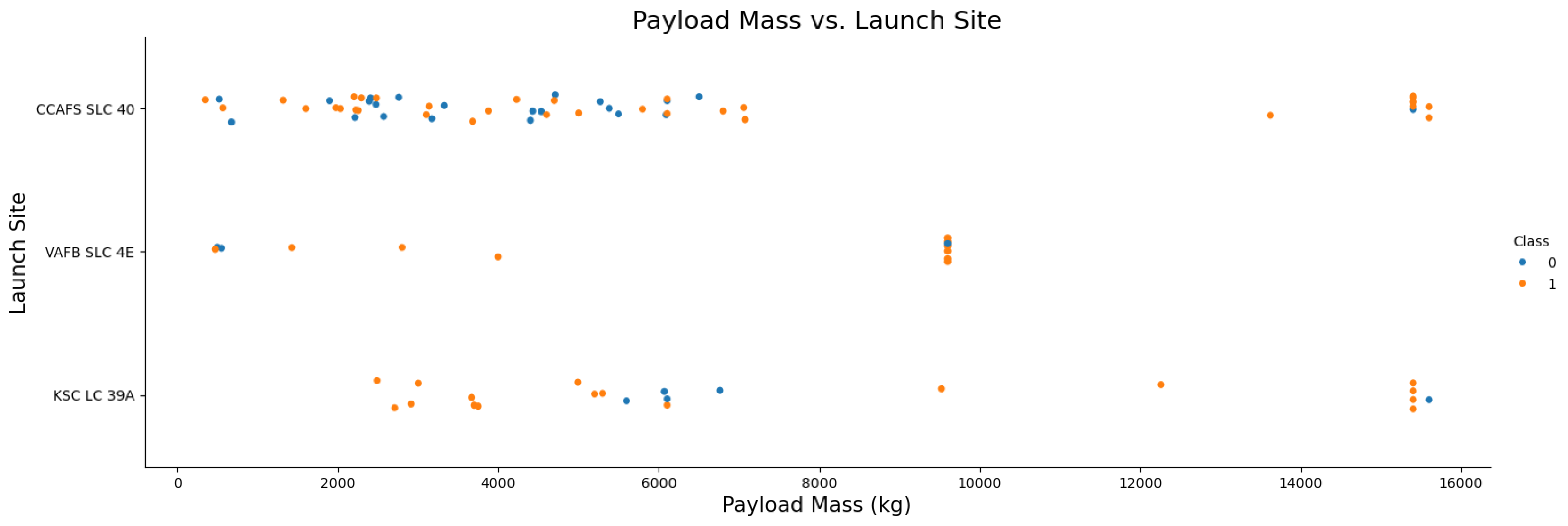
Insights drawn from EDA

Flight Number vs. Launch Site



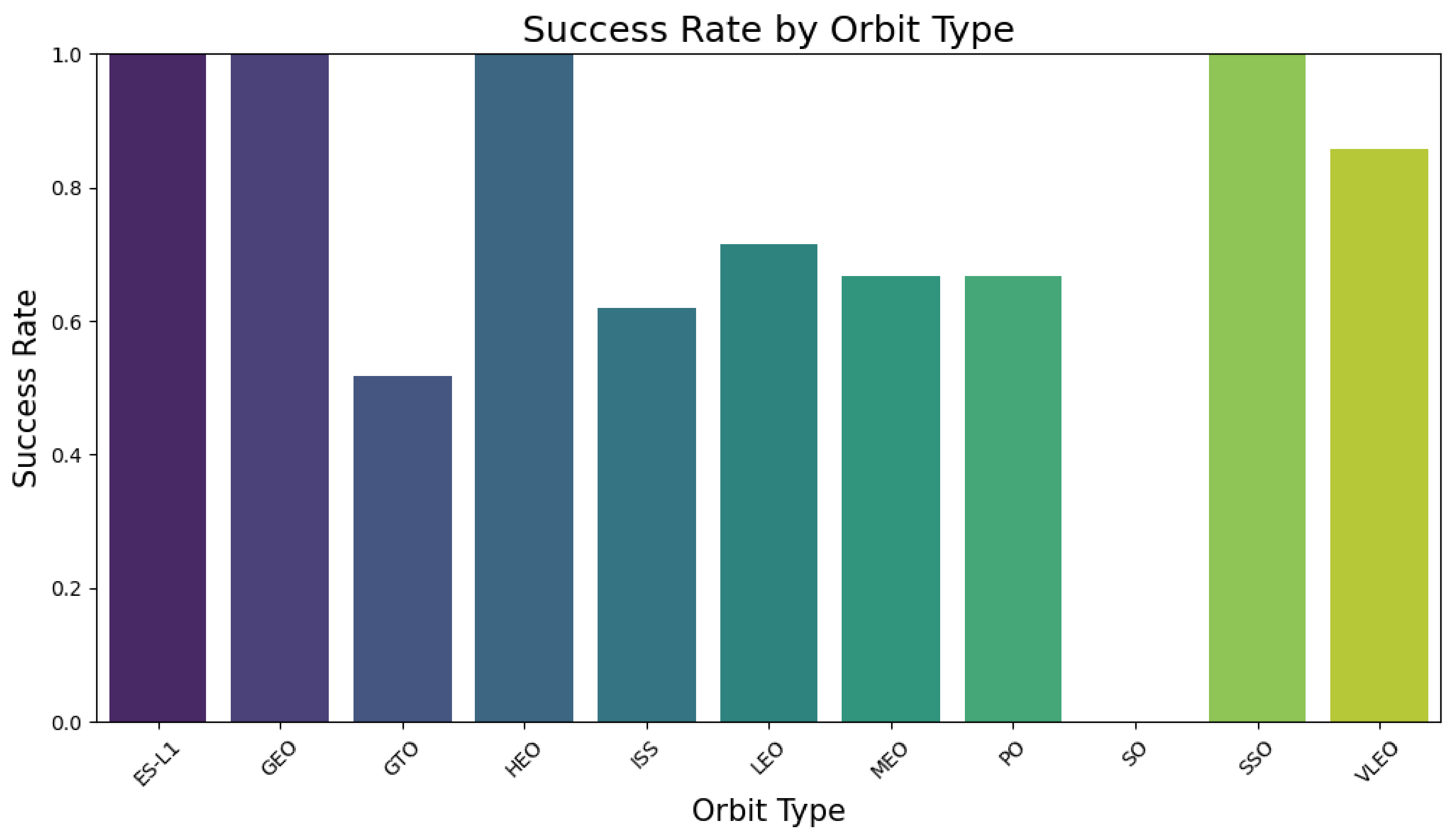
- As the number of flights increases, the success rate at some sites also appears to improve, especially at KSC LC 39A and CCAFS SLC 40.

Payload vs. Launch Site



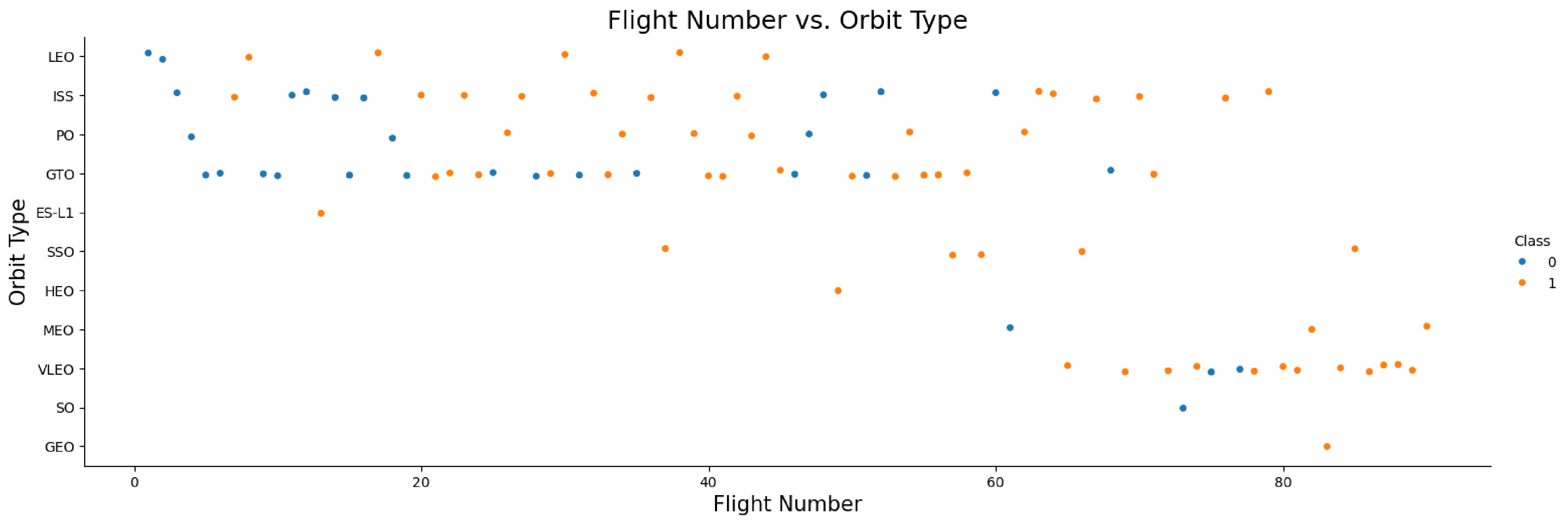
- For heavier loads (>10,000 kg), the success rate is higher, especially in KSC LC 39A.
- For lighter loads (<6000 kg), there are more failures, especially in CCAFS SLC 40.

Success Rate vs. Orbit Type



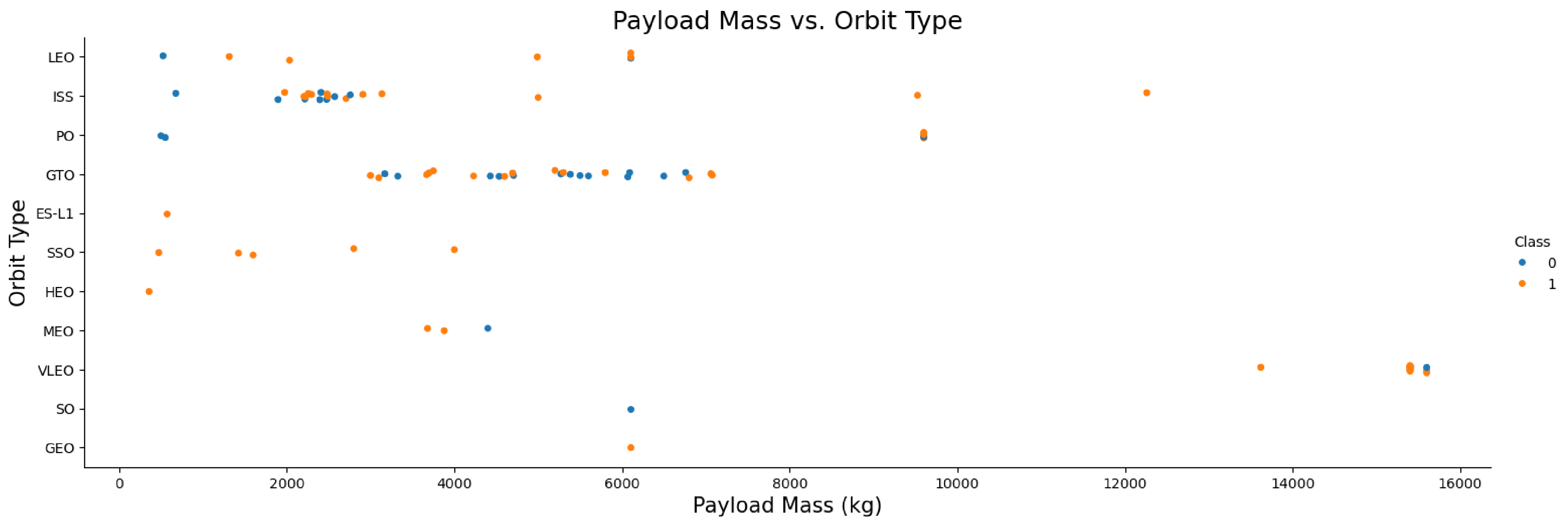
- Transfer orbits (GTO) have more failures, possibly because of the difficulty of the maneuver.
- Low altitude orbits (LEO, MEO, ISS) have intermediate success rates.
- Specific orbits such as ES-L1, GEO and SSO have 100% success rates, suggesting that launches to these missions have been well planned and executed.

Flight Number vs. Orbit Type



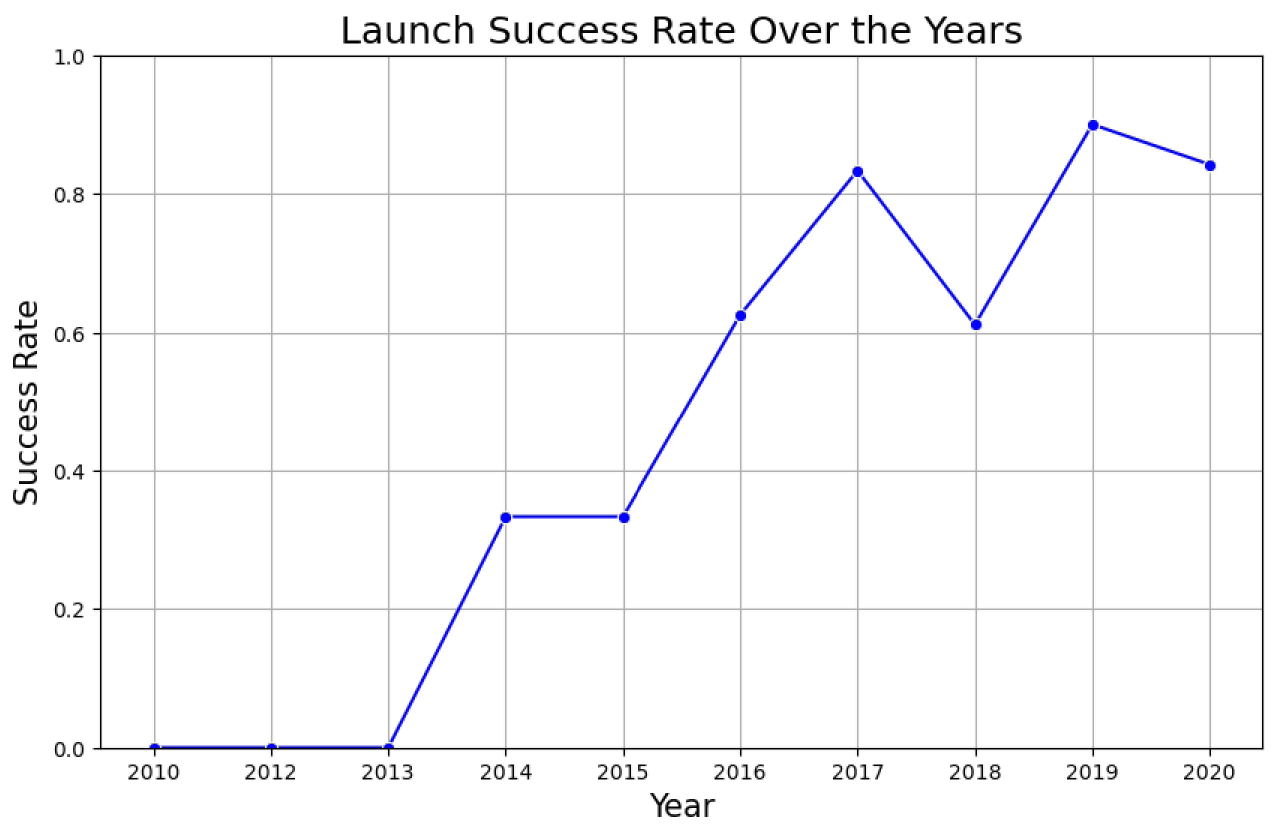
- Launch success improves with experience and number of flights.
- Low orbits (LEO, ISS) have more attempts and better results over time.
- Orbits like GTO continue to have failures even with more flights, suggesting they are more complex.

Payload vs. Orbit Type



- Low orbits (LEO, ISS) are more reliable for loads of different sizes.
- Transfer orbits (GTO) have more failures, especially with medium loads.
- Heavier loads tend to have more risks, especially in GTO and MEO.

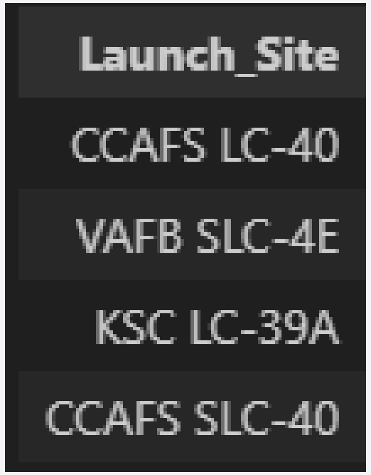
Launch Success Yearly Trend



- SpaceX has significantly improved its launch success rate over the years, overcoming early failures.
- By 2017-2020, launches became highly reliable, showing advancements in reusability and landing precision.
- The slight drop in 2018 might indicate specific mission challenges, but overall, the trend remains positive.

All Launch Site Names

- Find the names of the unique launch sites

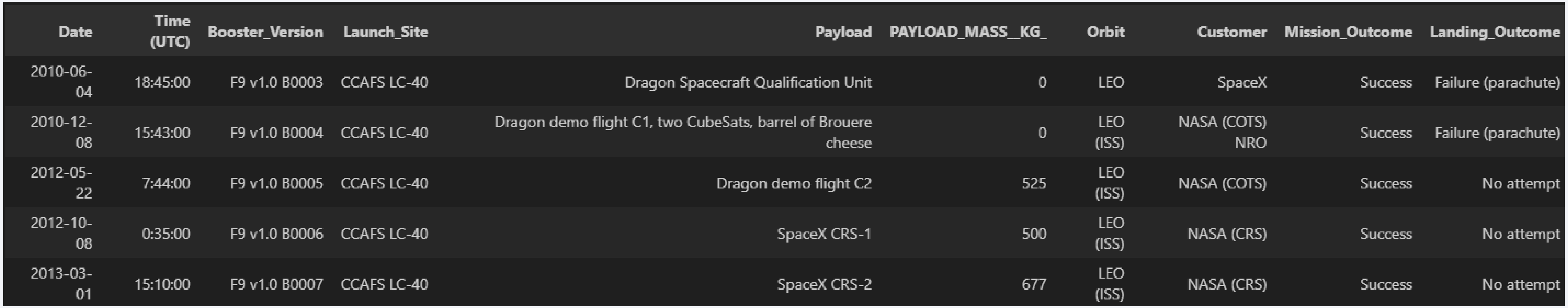


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- These sites represent the different locations used for SpaceX missions. This query helps identify all available launch sites without duplicates, ensuring clarity in the dataset.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

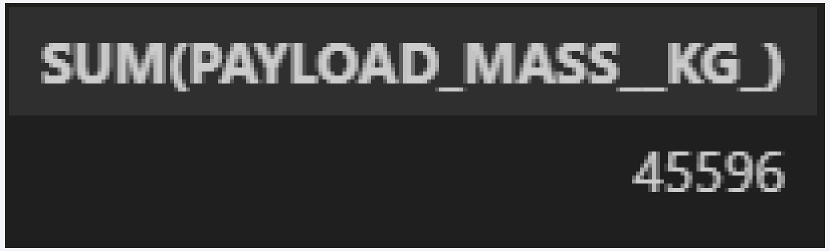


Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight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

- The Mission_Outcome is "Success" in all cases, but the Landing_Outcome varies (some failures and no attempts). This query helps filter launches from specific locations (starting with 'CCA'), allowing for targeted analysis of launch outcomes and payloads.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

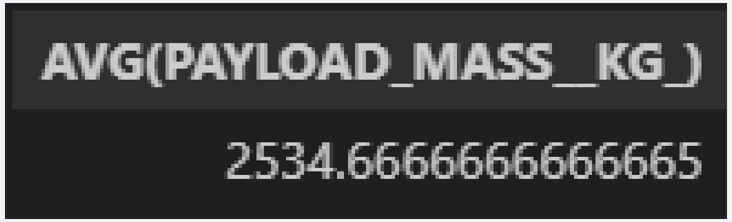


SUM(PAYLOAD_MASS__KG_)
45596

- By filtering for NASA (CRS) launches, this query provides insights into the total payload mass delivered, which is useful for understanding the scale of SpaceX's support for NASA's resupply missions.

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

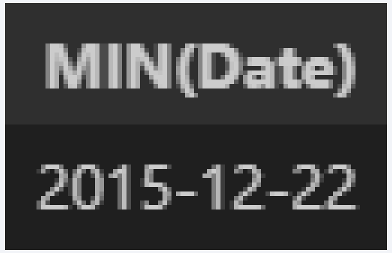


AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- This query is useful for comparing payload capacities across different Falcon 9 versions and understanding how much mass F9 v1.1 typically carried in SpaceX missions, all this through the average.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

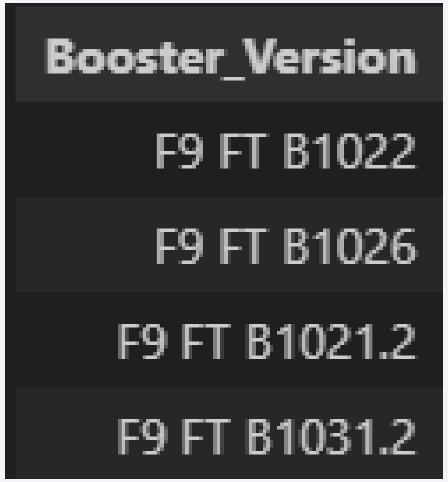


MIN(Date)
2015-12-22

- This milestone was crucial in SpaceX's reusability strategy, reducing launch costs significantly. This query helps identify the breakthrough moment when SpaceX achieved its first successful booster landing on solid ground, marking a significant advancement in rocket reusability.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

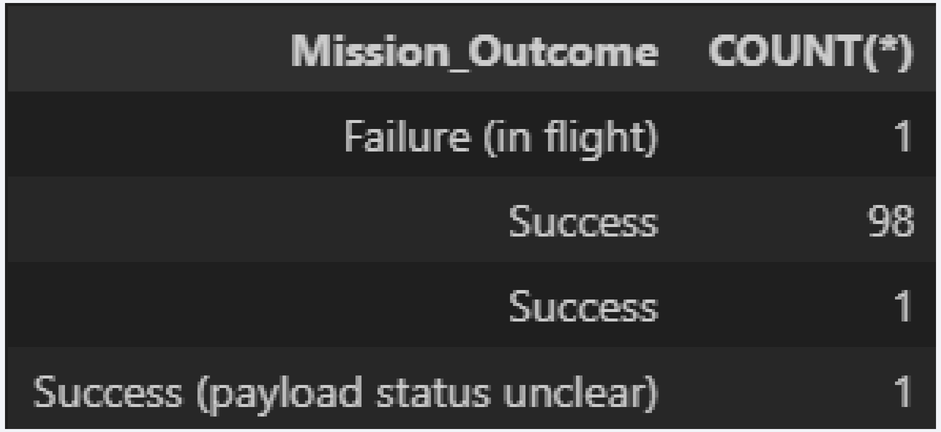


Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- These boosters were capable of carrying medium-range payloads and successfully executing drone ship landings. This query helps identify specific boosters that achieved successful drone ship landings while carrying moderate payloads, offering insights into SpaceX’s reusability and landing strategies.

Total Number of Successful and Failure Mission Outcomes

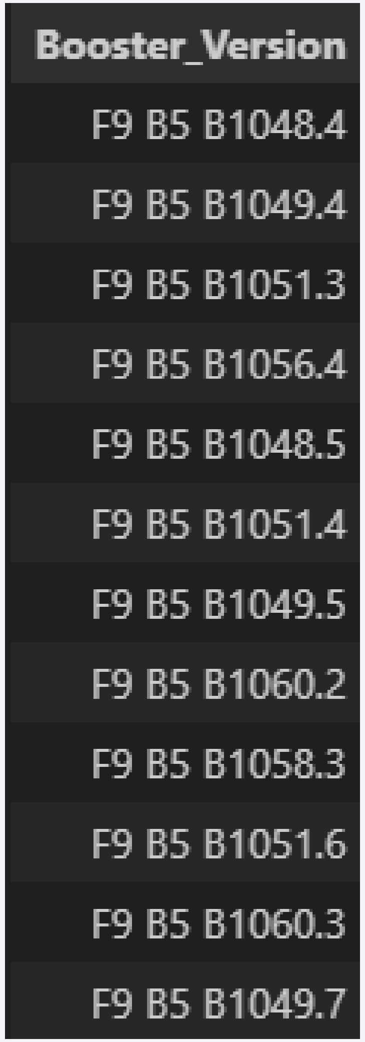
- Calculate the total number of successful and failure mission outcomes



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

- This query helps analyze SpaceX's overall mission success rate, showing that almost all missions were successful with very few failures. It also highlights possible inconsistencies in mission outcome labeling that may require data cleaning.

Boosters Carried Maximum Payload

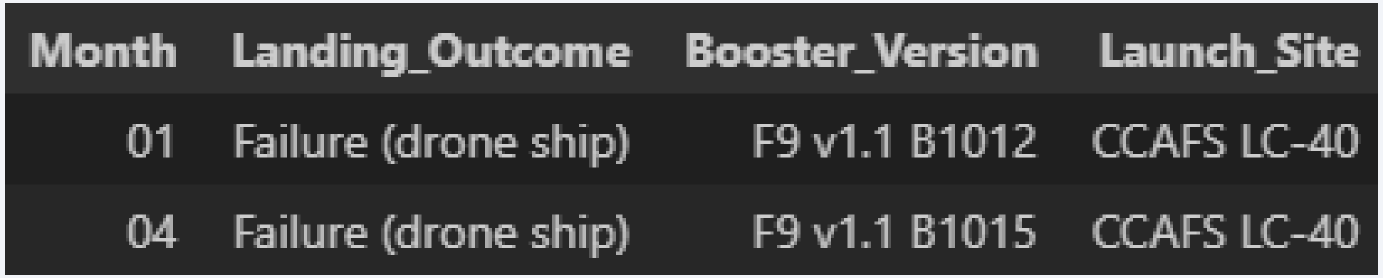


Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- The result shows multiple booster versions that have achieved this payload capacity, indicating multiple launches with the same maximum payload.
- This query helps identify the most capable boosters used by SpaceX in terms of payload capacity, which is crucial for analyzing booster performance and reusability.

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

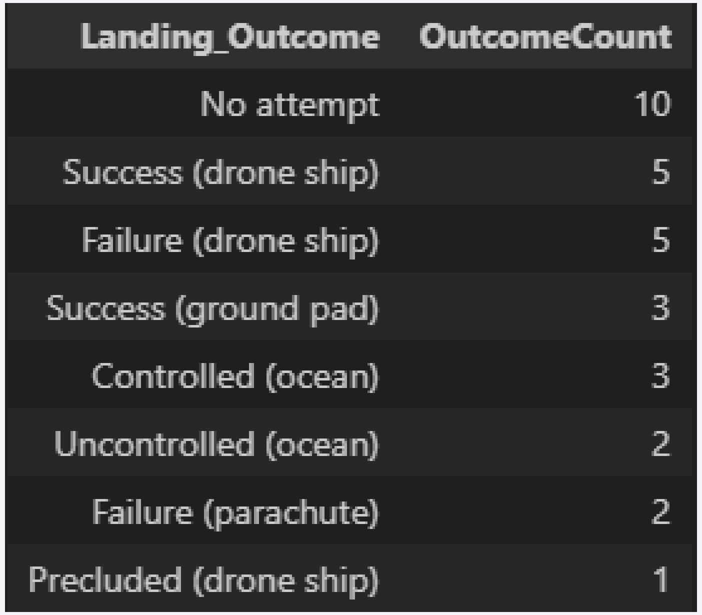


Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- The results show that in January and April 2015, there were failed drone ship landings for F9 v1.1 boosters B1012 and B1015 at CCAFS LC-40. This query helps in analyzing failure trends by month and identifying which boosters and launch sites were involved in unsuccessful landings on drone ships in 2015.

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
- The most frequent outcome was "No attempt" (10 launches), indicating that many missions did not include a landing attempt.
- This query helps evaluate how often different landing outcomes occurred within a specific timeframe. The results indicate that many missions did not attempt landings, and that drone ship landings were more common but had an equal success-failure ratio.

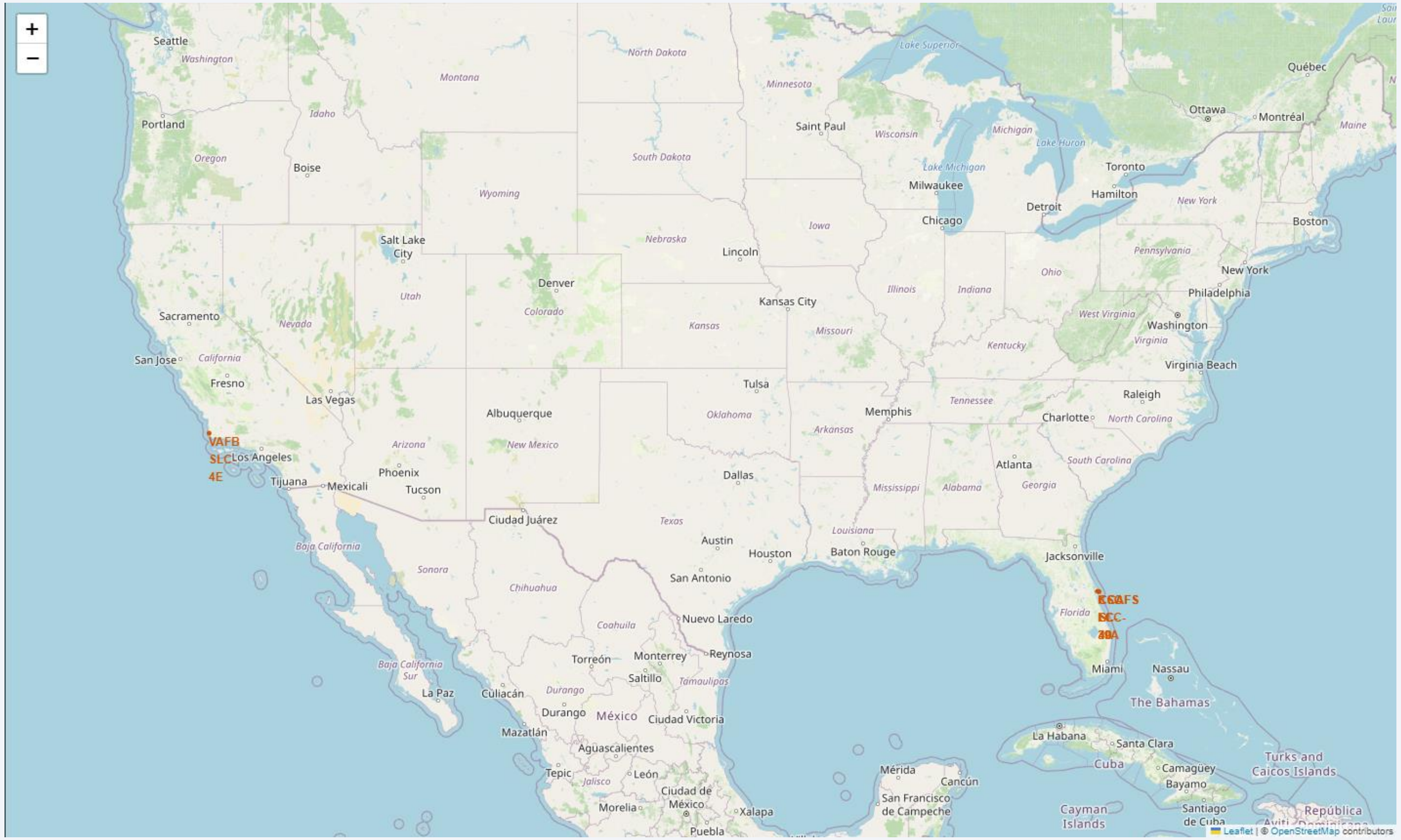


Landing_Outcome	OutcomeCount
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

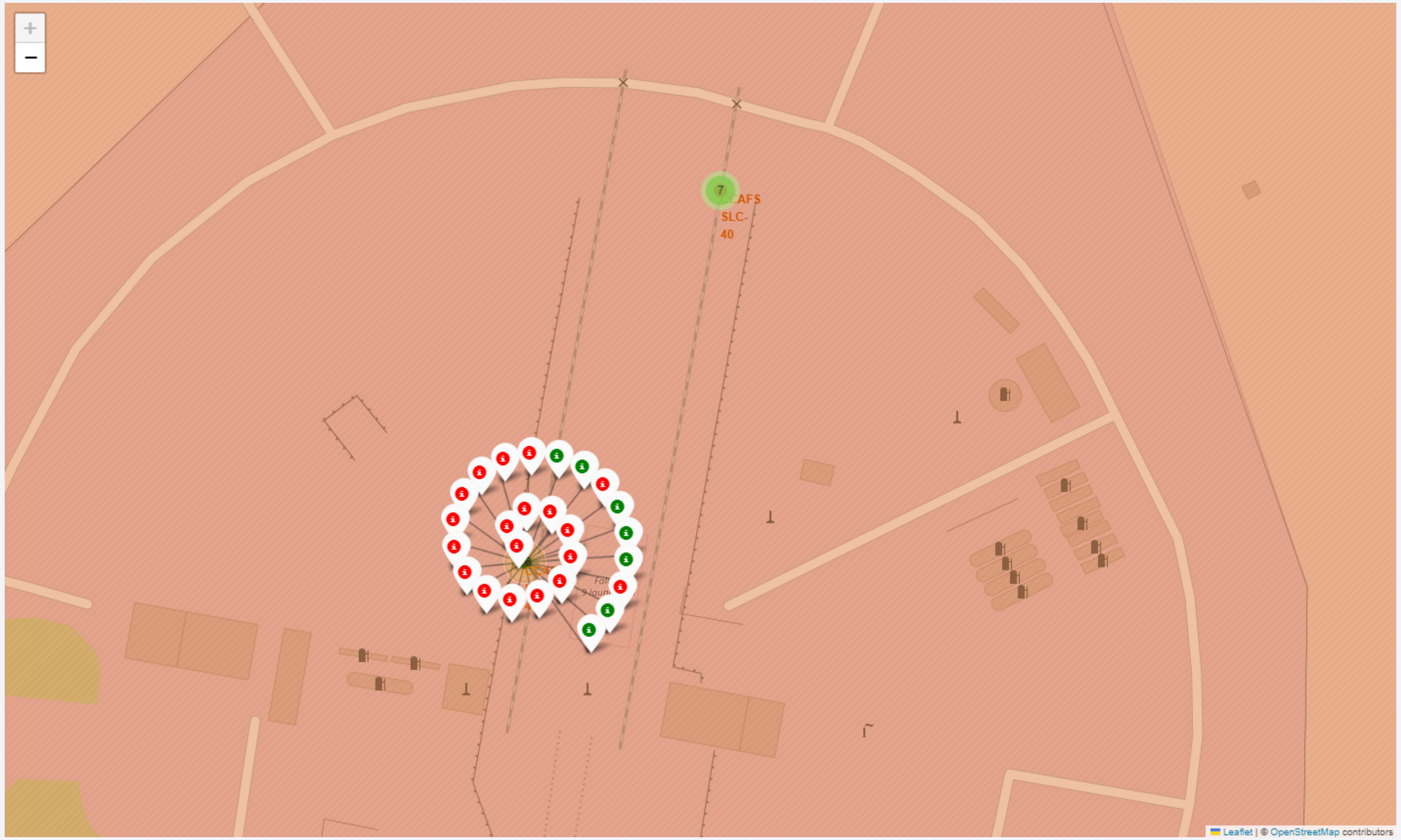
Launch Sites Proximities Analysis

Global Launches Sites Map



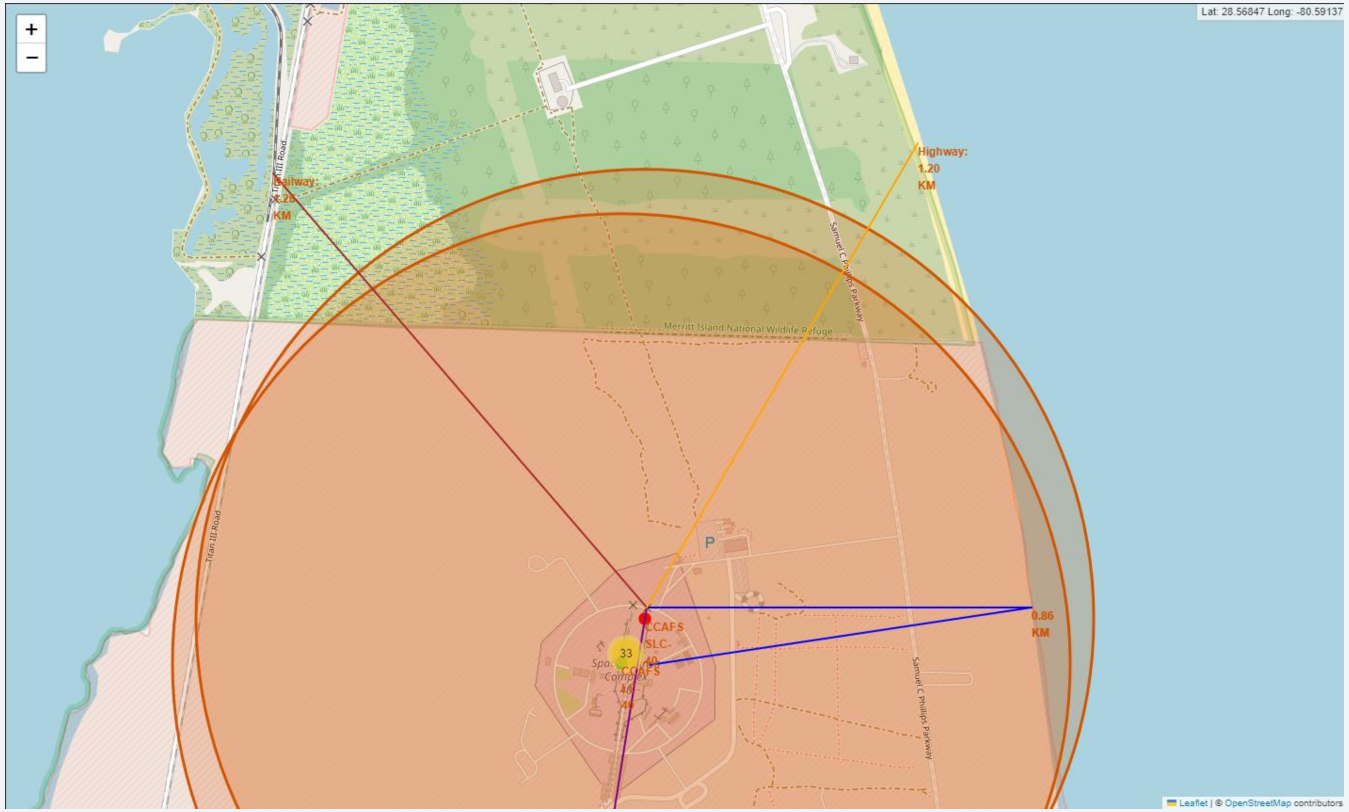
- Launch sites closer to the equator are beneficial for geostationary and equatorial orbit launches because the Earth's rotation provides an additional velocity boost. However, polar orbit missions (which require high inclination orbits) benefit from launch sites farther from the equator, like VAFB SLC-4E in California. Launching near the coast reduces risk to populated areas because rockets launch over open ocean rather than land. This also allows for easier booster recovery, especially for drone ship landings in the ocean.

Succes Identifier Map

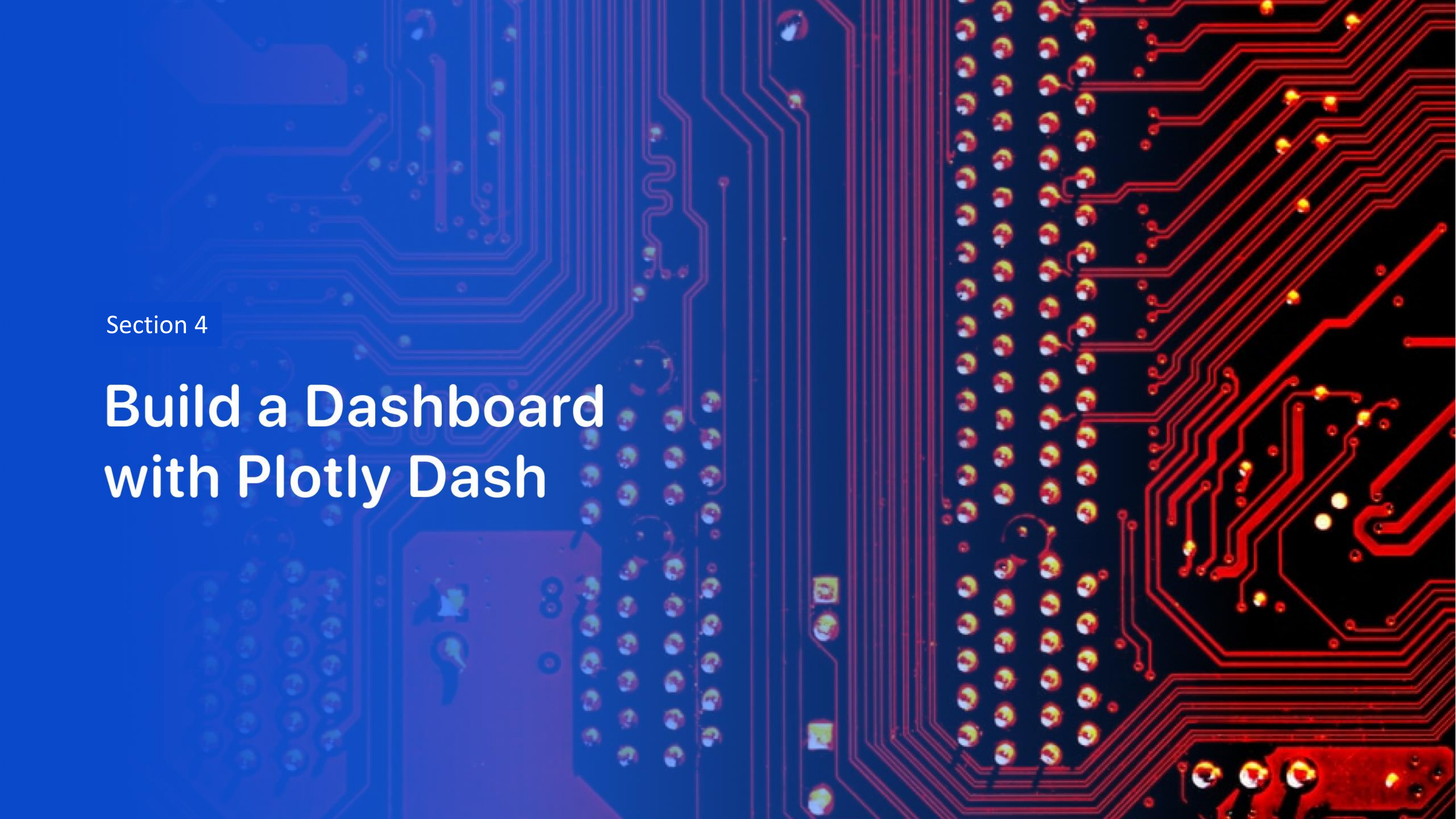


- Most launches at CCAFS SLC-40 resulted in failures (indicated by the large number of red markers).
- The clustering reveals a high frequency of launches at this location.
- The visualization helps identify trends in launch success and failure rates, providing insights for further analysis.

Roads, Highways, Cities zones map


+
−
Lat: 28.56847 Long: -80.59137
Railway: 1.28 KM
Highway: 1.20 KM
0.86 KM
CCAFS SLC-40
CCAFS 40
33
Titan III Road
Samuel C Phillips Parkway
Merritt Island National Wildlife Refuge
P
Leaflet | © OpenStreetMap contributors

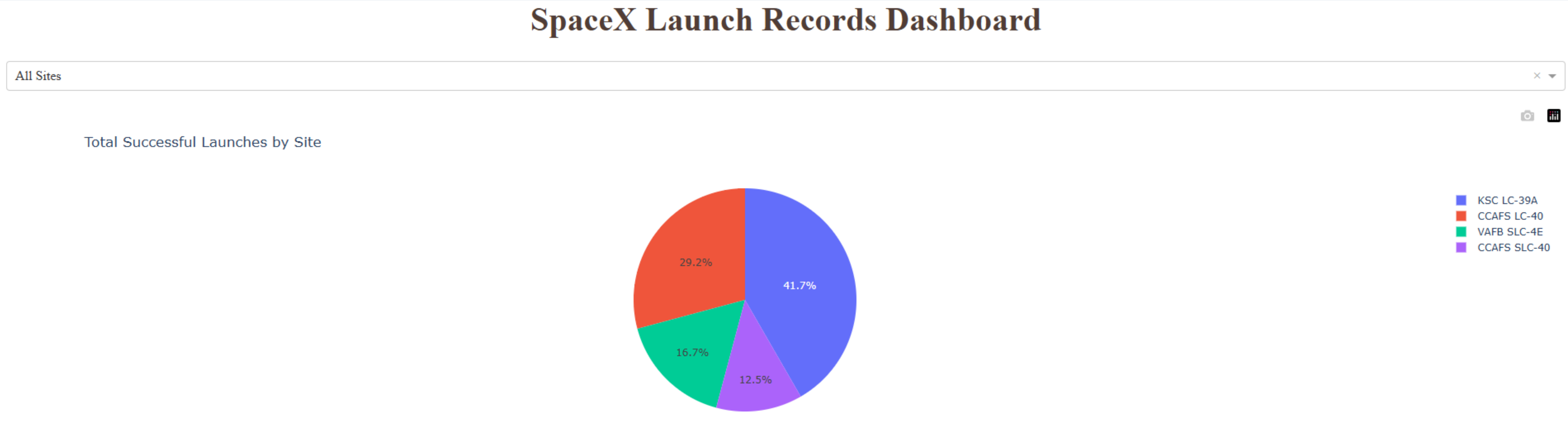
- Launch sites are relatively close to railways, enabling efficient transport of heavy equipment and fuel. Sites are very near highways, ensuring smooth logistics and workforce access. Launch sites are extremely close to the ocean, reducing risks. Sites are kept far from cities to enhance safety, minimize disruptions, and allow ample space for operations.



Section 4

Build a Dashboard with Plotly Dash

All launches sites Dashboard

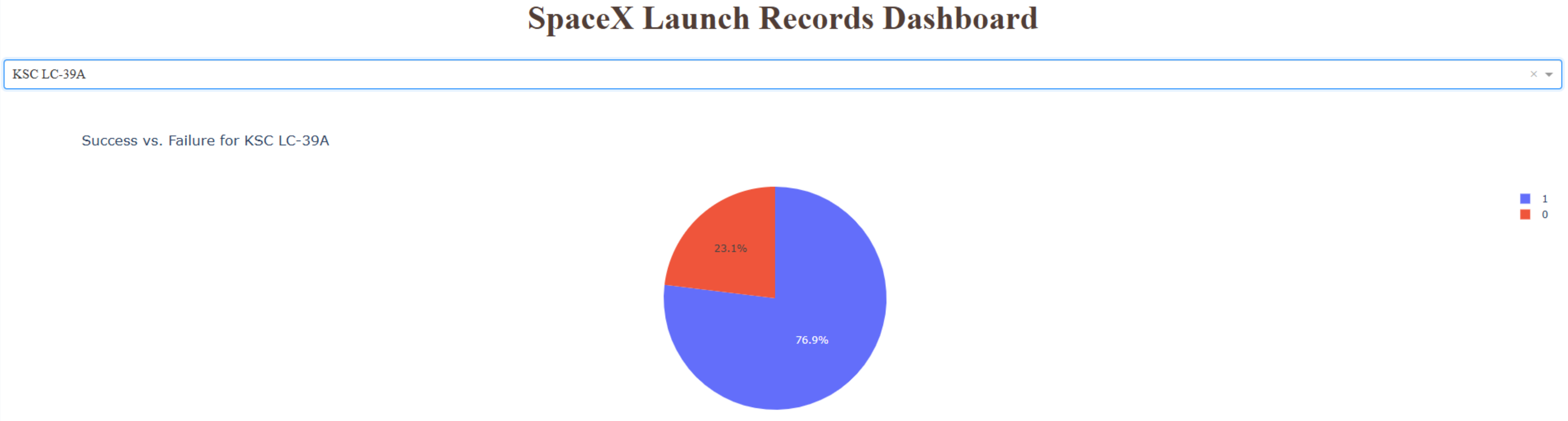


SpaceX Launch Records Dashboard

All Sites

- The pie chart visualizes the distribution of successful SpaceX launches across different launch sites, with a dropdown allowing site-specific filtering. Each segment represents a launch site, with percentage labels indicating their contribution to total successes. KSC LC-39A leads with the highest success rate (41.7%), followed by CCAFS LC-40 (29.2%), VAFB SLC-4E (16.7%), and CCAFS SLC-40 (12.5%. The legend helps identify each site by color. The findings suggest that KSC LC-39A is the most reliable or frequently used site, while other sites have varying levels of success, providing insights into SpaceX's most effective launch locations.

Most successful site dashboard

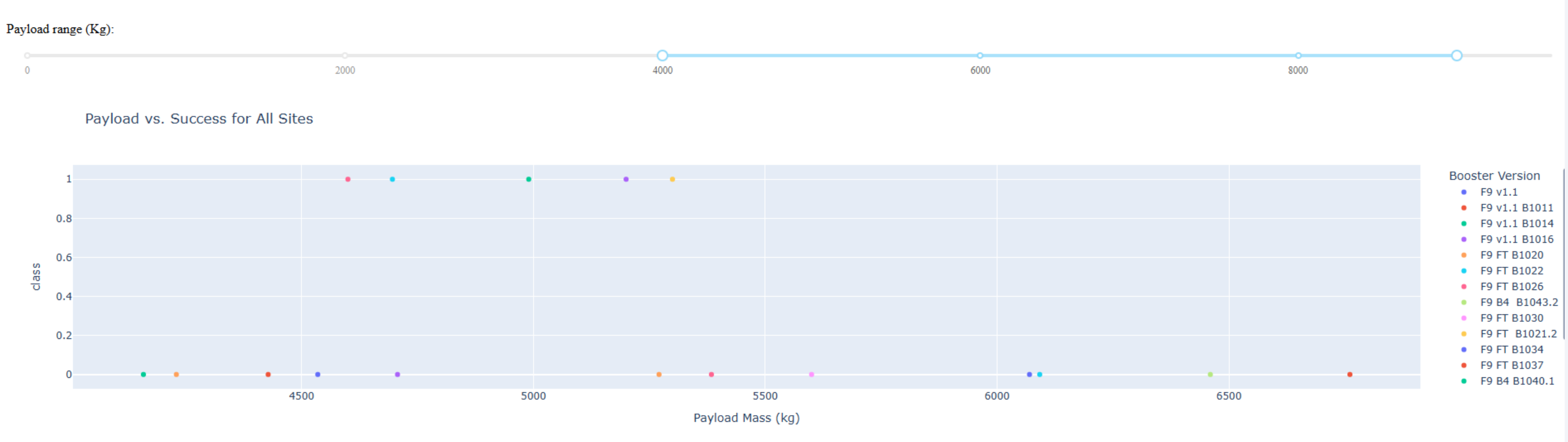


SpaceX Launch Records Dashboard

KSC LC-39A

- This SpaceX Launch Records Dashboard visualizes the success vs. failure rate for the selected launch site (KSC LC-39A). The dropdown menu at the top allows users to select different launch sites for comparison. The pie chart represents the proportion of successful (blue) and failed (red) launches at KSC LC-39A, with 76.9% successes and 23.1% failures. The legend on the right clarifies that 1 represents success and 0 represents failure. The chart highlights that KSC LC-39A has a high success rate, making it one of the most reliable launch sites.

All sites Payload vs Success chart (4000 - 5500)

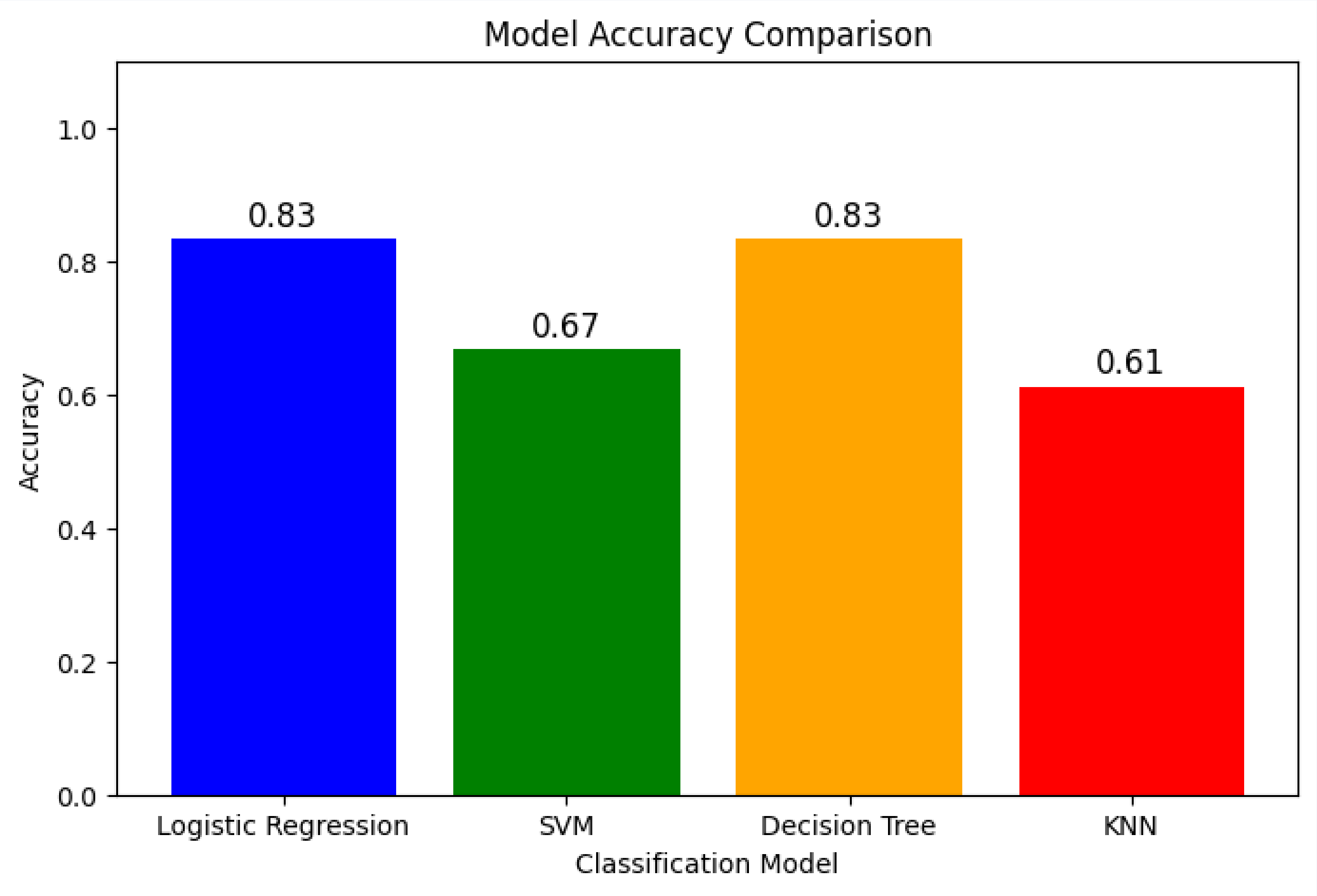


- The scatter plot shows the relationship between payload mass and launch success, with a payload range slider for filtering. The x-axis represents payload mass (kg), and the y-axis (class) indicates success (1) or failure (0). Each color-coded point represents a booster version, identified in the legend. Key findings suggest that payloads between 4000-6000 kg have higher success rates, with boosters like F9 FT B1022 and F9 FT B1026 performing well. Failures are more scattered, indicating challenges with heavier payloads. This visualization helps identify optimal boosters for different payload capacities.

Section 5

Predictive Analysis (Classification)

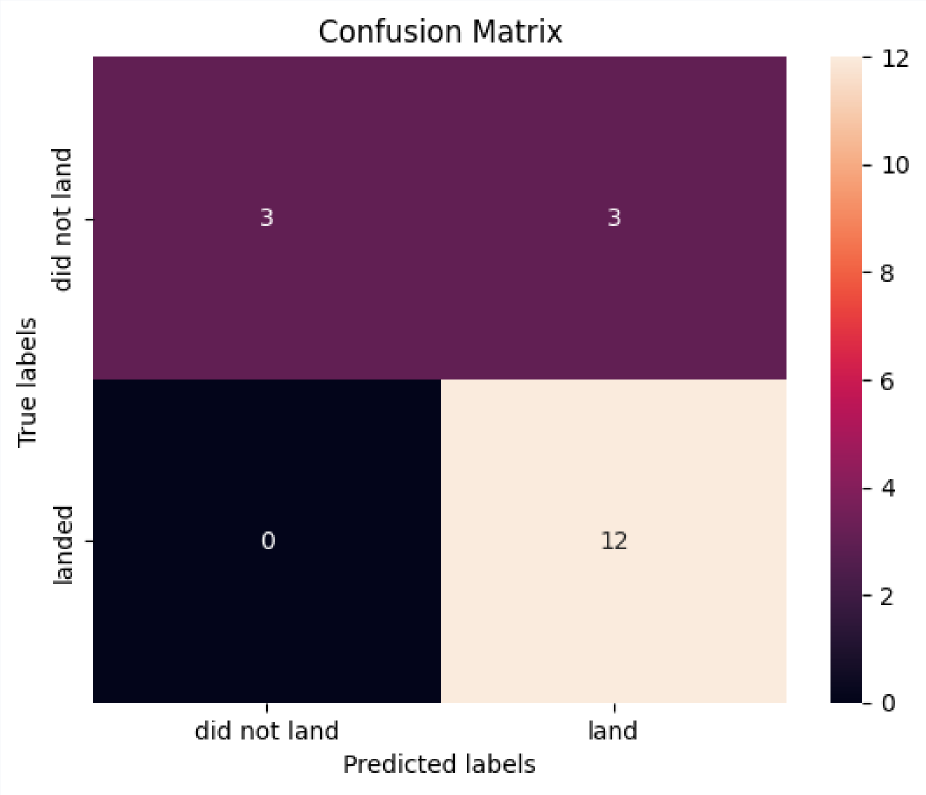
Classification Accuracy



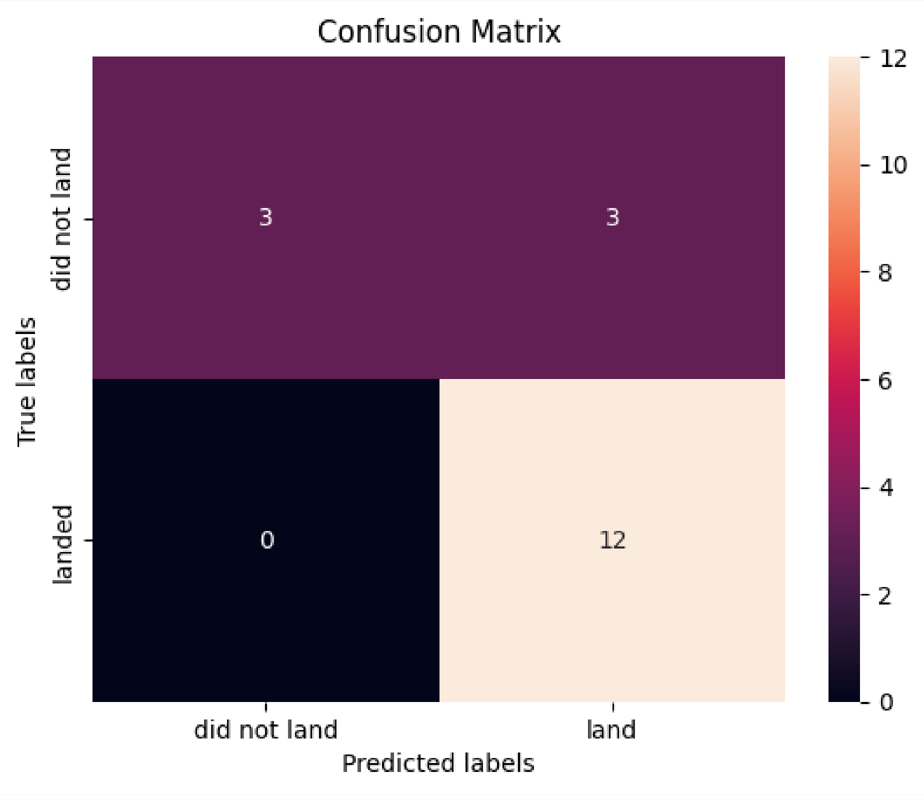
- The model that has the highest classification accuracy is Logistic Regression and Decision Tree.

Confusion Matrix

Logistic Regression



Decision Tree Classifier



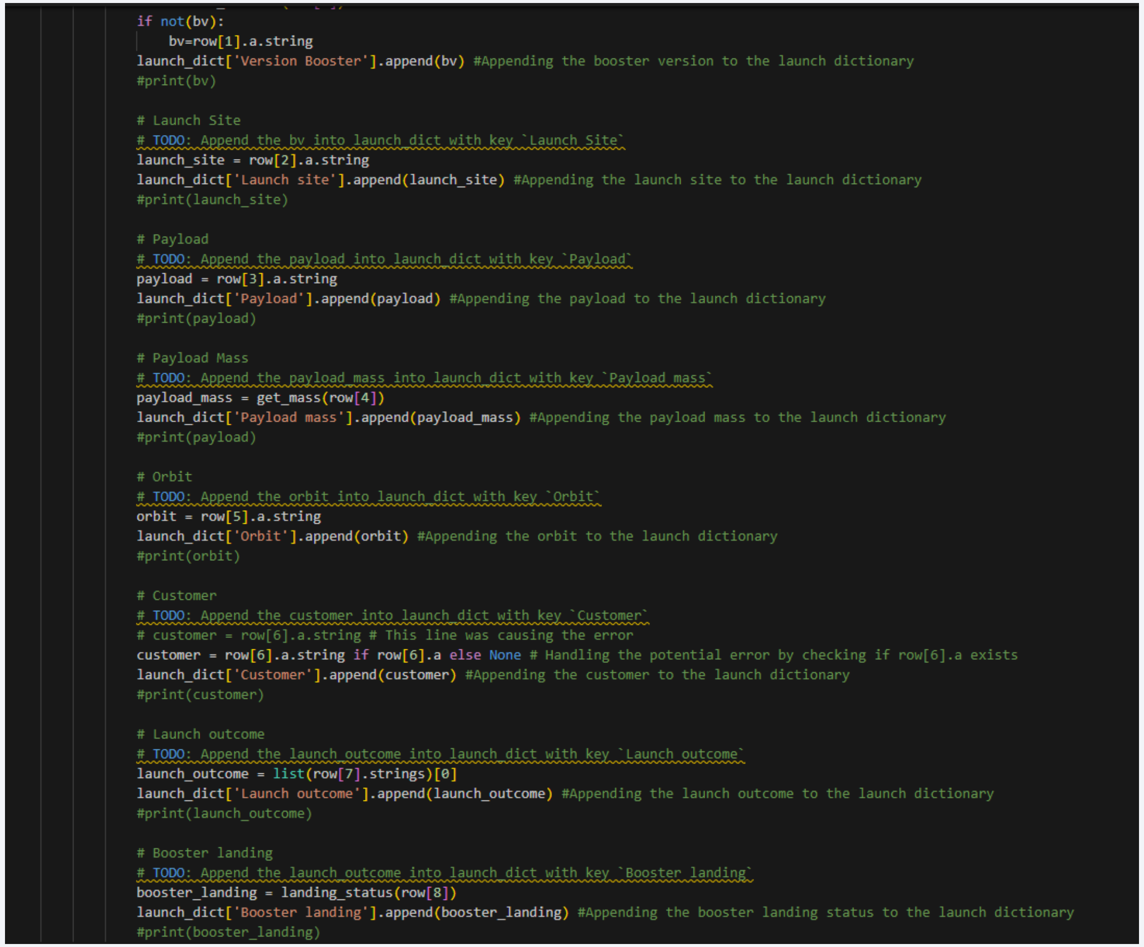
- Both models, Logistic Regression and Decision Tree have the best accuracy. That's in my case, knowing that decision tree has a random built.

Conclusions

- Data Collection
  - The project successfully combined data from web scraping and API sources, enabling comprehensive access to SpaceX launch records for further analysis.
- Data Exploration & Visualization
  - Using SQL queries, Folium maps, and Plotly charts, we uncovered key insights such as launch site success rates, payload impact, and booster reliability.
- Interactive Dashboard
  - The Plotly Dash application provided an intuitive way to explore launch performance by site and payload range, supporting dynamic, user-driven analysis.
- Predictive Modeling
  - Among the classification models built, SVM achieved the highest accuracy (100%), proving most effective at predicting launch success based on historical data.

Appendix

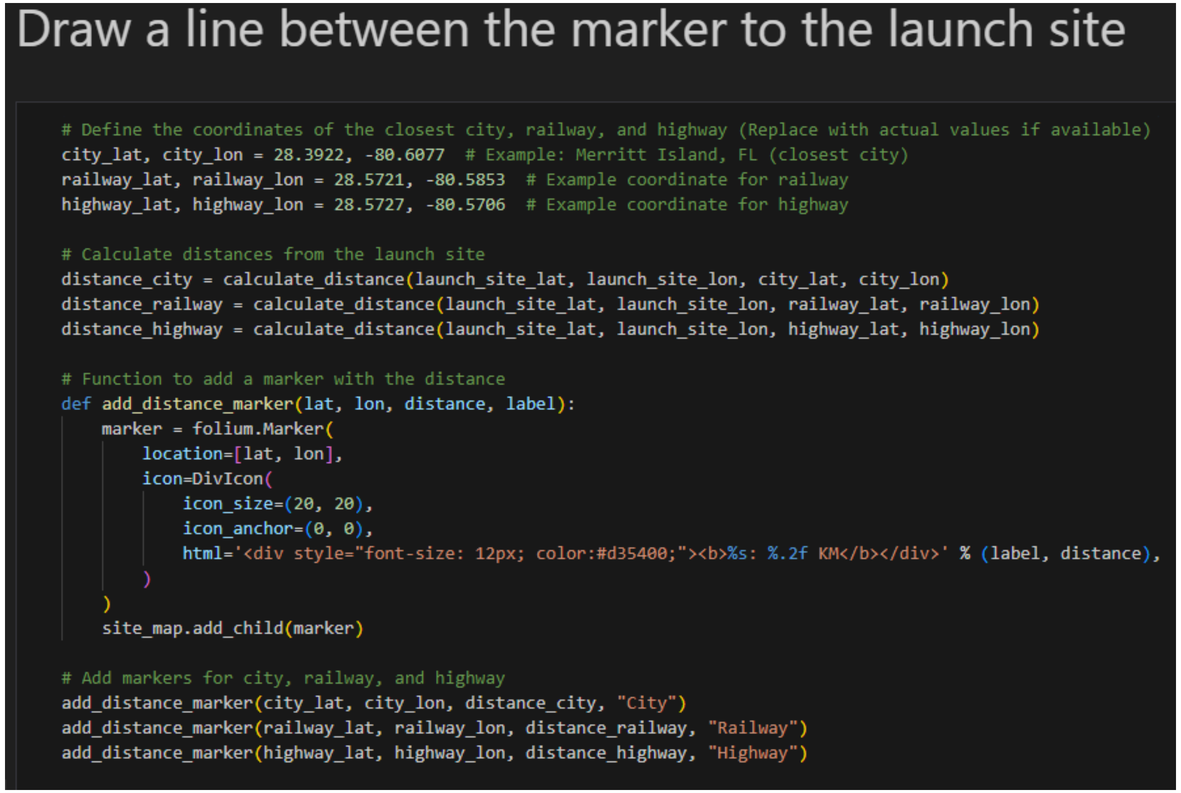
- Web scraping



```
if not(bv):
    bv=row[1].a.string
launch_dict['Version Booster'].append(bv) #Appending the booster version to the launch dictionary
#print(bv)

# Launch Site
# TODO: Append the bv into launch_dict with key `Launch Site`
launch_site = row[2].a.string
launch_dict['Launch site'].append(launch_site) #Appending the launch site to the launch dictionary
#print(launch_site)

# Payload
# TODO: Append the payload into launch_dict with key `Payload`
payload = row[3].a.string
launch_dict['Payload'].append(payload) #Appending the payload to the launch dictionary
#print(payload)

# Payload Mass
# TODO: Append the payload_mass into launch_dict with key `Payload mass`
payload_mass = get_mass(row[4])
launch_dict['Payload mass'].append(payload_mass) #Appending the payload mass to the launch dictionary
#print(payload)

# Orbit
# TODO: Append the orbit into launch_dict with key `Orbit`
orbit = row[5].a.string
launch_dict['Orbit'].append(orbit) #Appending the orbit to the launch dictionary
#print(orbit)

# Customer
# TODO: Append the customer into launch_dict with key `Customer`
# customer = row[6].a.string # This line was causing the error
customer = row[6].a.string if row[6].a else None # Handling the potential error by checking if row[6].a exists
launch_dict['Customer'].append(customer) #Appending the customer to the launch dictionary
#print(customer)

# Launch outcome
# TODO: Append the launch_outcome into launch_dict with key `Launch outcome`
launch_outcome = list(row[7].strings)[0]
launch_dict['Launch outcome'].append(launch_outcome) #Appending the launch outcome to the launch dictionary
#print(launch_outcome)

# Booster landing
# TODO: Append the launch_outcome into launch_dict with key `Booster landing`
booster_landing = landing_status(row[8])
launch_dict['Booster landing'].append(booster_landing) #Appending the booster landing status to the launch dictionary
#print(booster_landing)
```

Appendix

- Drawing lines with interactive map



Draw a line between the marker to the launch site

```python
# Define the coordinates of the closest city, railway, and highway (Replace with actual values if available)
city_lat, city_lon = 28.3922, -80.6077  # Example: Merritt Island, FL (closest city)
railway_lat, railway_lon = 28.5721, -80.5853  # Example coordinate for railway
highway_lat, highway_lon = 28.5727, -80.5706  # Example coordinate for highway

# Calculate distances from the launch site
distance_city = calculate_distance(launch_site_lat, launch_site_lon, city_lat, city_lon)
distance_railway = calculate_distance(launch_site_lat, launch_site_lon, railway_lat, railway_lon)
distance_highway = calculate_distance(launch_site_lat, launch_site_lon, highway_lat, highway_lon)

# Function to add a marker with the distance
def add_distance_marker(lat, lon, distance, label):
    marker = folium.Marker(
        location=[lat, lon],
        icon=DivIcon(
            icon_size=(20, 20),
            icon_anchor=(0, 0),
            html='<div style="font-size: 12px; color:#d35400;"><b>%s: %.2f KM</b></div>' % (label, distance),
        )
    )
    site_map.add_child(marker)

# Add markers for city, railway, and highway
add_distance_marker(city_lat, city_lon, distance_city, "City")
add_distance_marker(railway_lat, railway_lon, distance_railway, "Railway")
add_distance_marker(highway_lat, highway_lon, distance_highway, "Highway")
```

Thank you!