IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chris Van Ham
December 2024

Outline

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

Executive Summary

- Methods-
  - Data Wrangling: Cleaned and prepared data for analysis.
  - EDA and Visualizations: Identified trends and outliers using Python and SQL.
  - Interactive Tools: Built a Folium map and Plotly Dash dashboard for dynamic insights.
  - Predictive Analysis: Applied machine learning models to classify and predict outcomes.
- Results Key Insights:
  - Uncovered trends, correlations, and anomalies through EDA.
  - Geospatial Patterns: Mapped geographic clusters with Folium.
  - Dashboard Utility: Enabled real-time data exploration for stakeholders.
  - Predictive Accuracy: Achieved ~81-85% accuracy in classification models.
  - Actionable Insights: Synthesized findings into recommendations for stakeholders.

Introduction

- Our new company, SpaceY, is looking to enter the highly competitive space travel market. At present, SpaceX is the industry leader in space travel due to its relatively low-cost business practices, including salvaging and reusing many components of its rockets.
- This project answers the question of how SpaceY can compete with SpaceX by analyzing available data and information from SpaceX.
  - What key trends and relationships exist within the data?
  - How can stakeholders interact dynamically with the data to uncover insights specific to their needs?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from the links and information in the Hands-on lab exercises. This data included SpaceX API and web scrapping from SpaceX Wiki
- Perform data wrangling
  - Data wrangling involved extracting launch data from the SpaceX API and combining it with historical launch details from Wikipedia. The process included cleaning, standardizing, and merging datasets to create a unified, accurate dataset for analysis of SpaceX launches.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Methodology

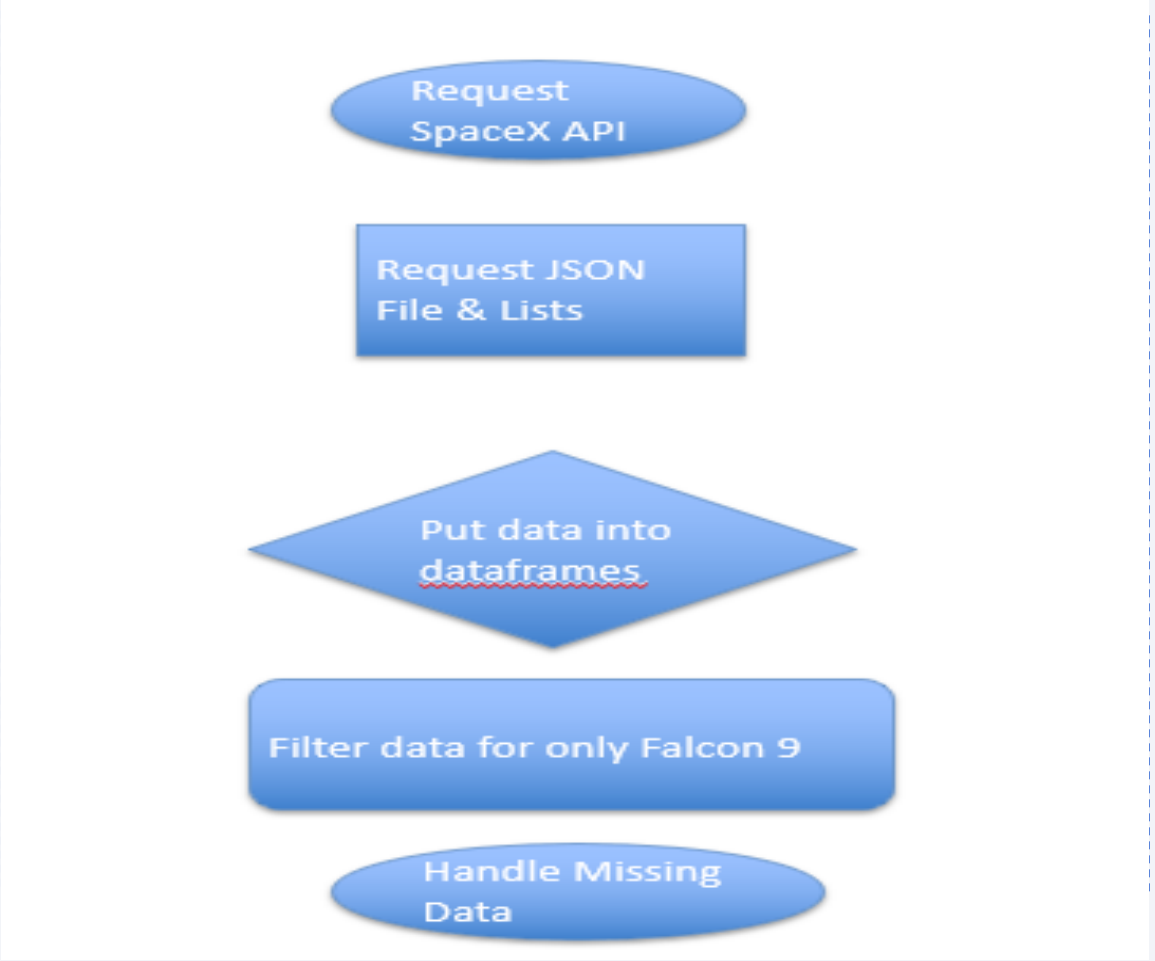
Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
    - Exploratory Data Analysis (EDA) was conducted on SpaceX launch data extracted from the SpaceX API and Wikipedia to identify key trends, patterns, and anomalies. Interactive visual analytics tools like Matplotlib and Seaborn were used to create dynamic and engaging visualizations, highlighting factors such as success rates and payload types. This methodology provided a deeper understanding of the data, enabling clear communication of insights through visually compelling slides.
- Perform interactive visuals using Folium and Plotly Dash
    - Interactive visual analytics were performed using Folium to create geospatial maps, showcasing the launch site locations and their geographical significance. Additionally, Plotly Dash was used to build a dynamic dashboard, enabling users to explore key metrics and trends interactively, such as payload success rates and launch outcomes.

Data Collection – SpaceX API

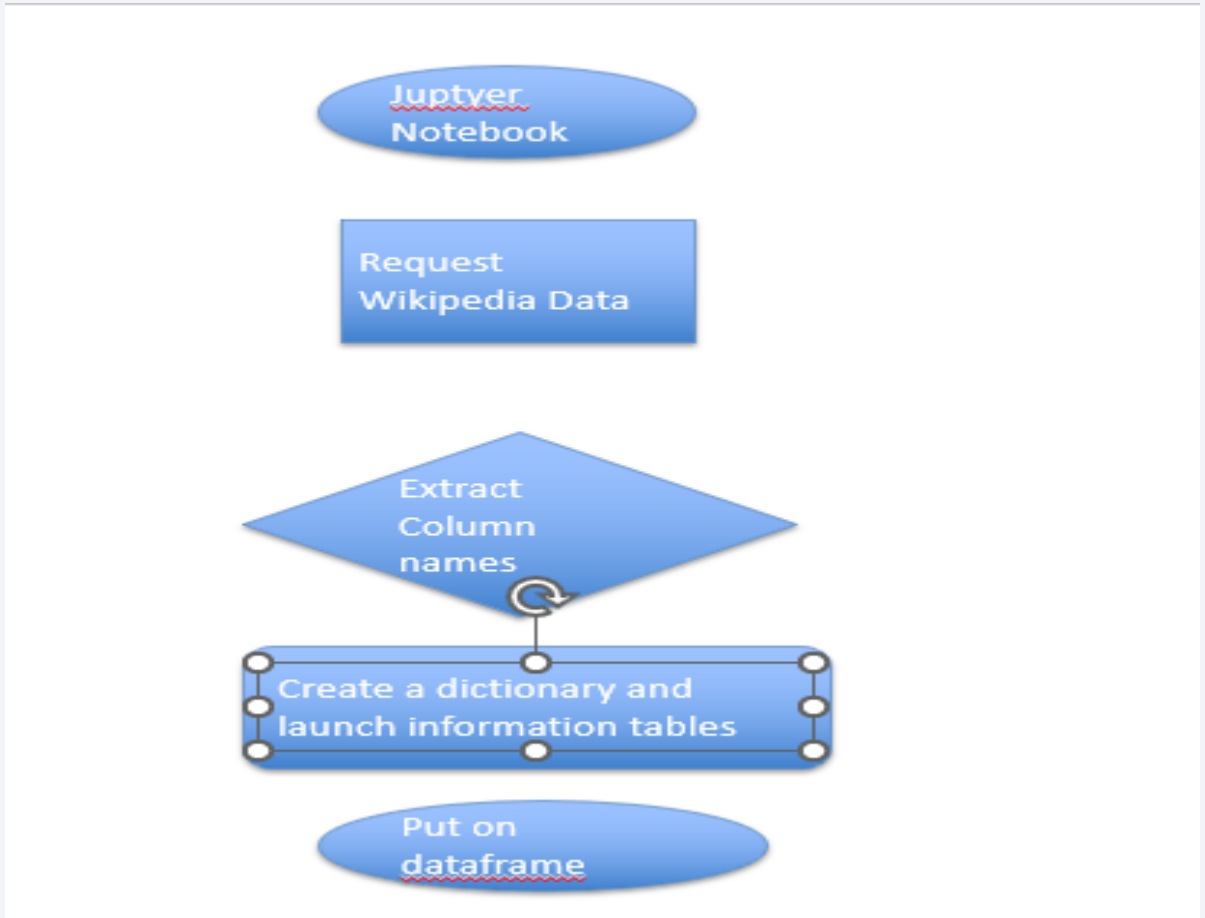
- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL



Data Collection - Scraping

- Start: Fetch webpage content using requests. Parse HTML: Use BeautifulSoup. Extract Data: Iterate through table rows to extract and clean data. Structure Data: Organize extracted data into a dictionary. Convert to DataFrame: Transform the dictionary into a Pandas DataFrame. Export Data: Save the cleaned DataFrame as a CSV file.
- Add the GitHub URL



Data Wrangling

- Data taken from Wikipedia SpaceX Falcon launches.
- HTML Parsing with Beautiful Soup
- Data Extraction
- Data Cleaning
- Data Structuring
- Add the GitHub URL

EDA with Data Visualization

- Bar Chart: Shows the number of launches and the most active locations.
- Pie Chart: Visualizes the proportion of missions to different orbits. Highlights diversity in mission targets.
- Histogram: Shows the distribution of payload weights. Identifies trends in mission capacities.
- Scatter Plot: Explores relationships between payload mass and orbit type.
- Line Chart: Tracks yearly launch growth.
- Purpose: Simplifies data, reveals trends, and highlights SpaceX's operational efficiency.
- Add the GitHub URL

EDA with SQL

- Unique Launch Sites: Extracted distinct launch site names from the dataset.
- Launch Sites Starting with 'CCA': Filtered and displayed 5 records of launch sites beginning with 'CCA'.
- Total Payload Mass by NASA (CRS): Calculated the total payload mass for missions launched by NASA (CRS).
- Average Payload Mass for F9 v1.1: Computed the average payload mass carried by the booster version F9 v1.1.
- First Successful Ground Pad Landing: Retrieved the date of the first successful landing on a ground pad.
- Boosters with Max Payload Mass: Identified boosters that carried the maximum payload mass using a subquery.
- Failure Outcomes on Drone Ship (2015): Listed failures in drone ship landings in 2015, grouped by month.
- Rank Landing Outcomes (2010–2017):Ranked landing outcomes by count within a specific date range.
- Add the GitHub URL

Build an Interactive Map with Folium

- Markers were added at each launch site to identify their locations and provide details such as the site name, payload, and landing outcomes.
- Circle Markers represent the payload mass, with circle sizes scaled for better visualization, providing a quick visual comparison of payload weights across sites.
- Polylines show the connected repeated launch locations to visualize frequently used sites and their spatial relationships.
- Add the GitHub URL

Build a Dashboard with Plotly Dash

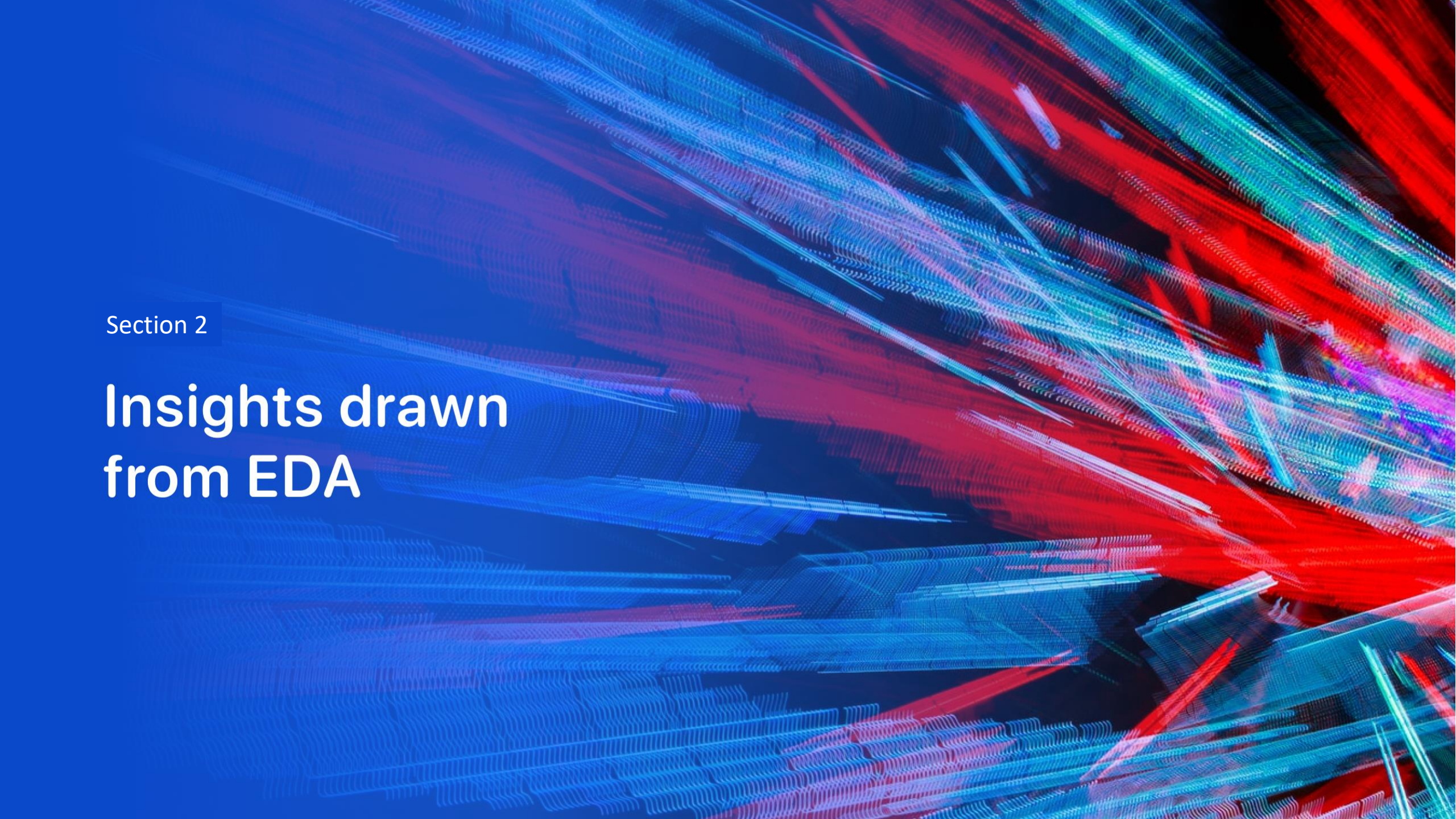
- A Pie Chart and a Scatterplot were used.
- The Pie Chart showed the distribution of successful landings from all launch sites. The scatterplot showed launch sites and payload mass from 0-10,000kg.

Predictive Analysis (Classification)

- The dataset (dataset_part_2.csv) was cleaned, and features were selected for modeling, with the Class column serving as the target variable.
- Preprocessing involved scaling numerical data and encoding categorical variables (e.g., from dataset_part_3.csv).
- Classification algorithms such as Logistic Regression, Random Forest, and Support Vector Machines were applied.
- Models were evaluated using accuracy, precision, recall, and F1 score metrics on a test dataset (20% split from the original).
- Hyperparameter tuning (e.g., grid search) was used to optimize model performance.

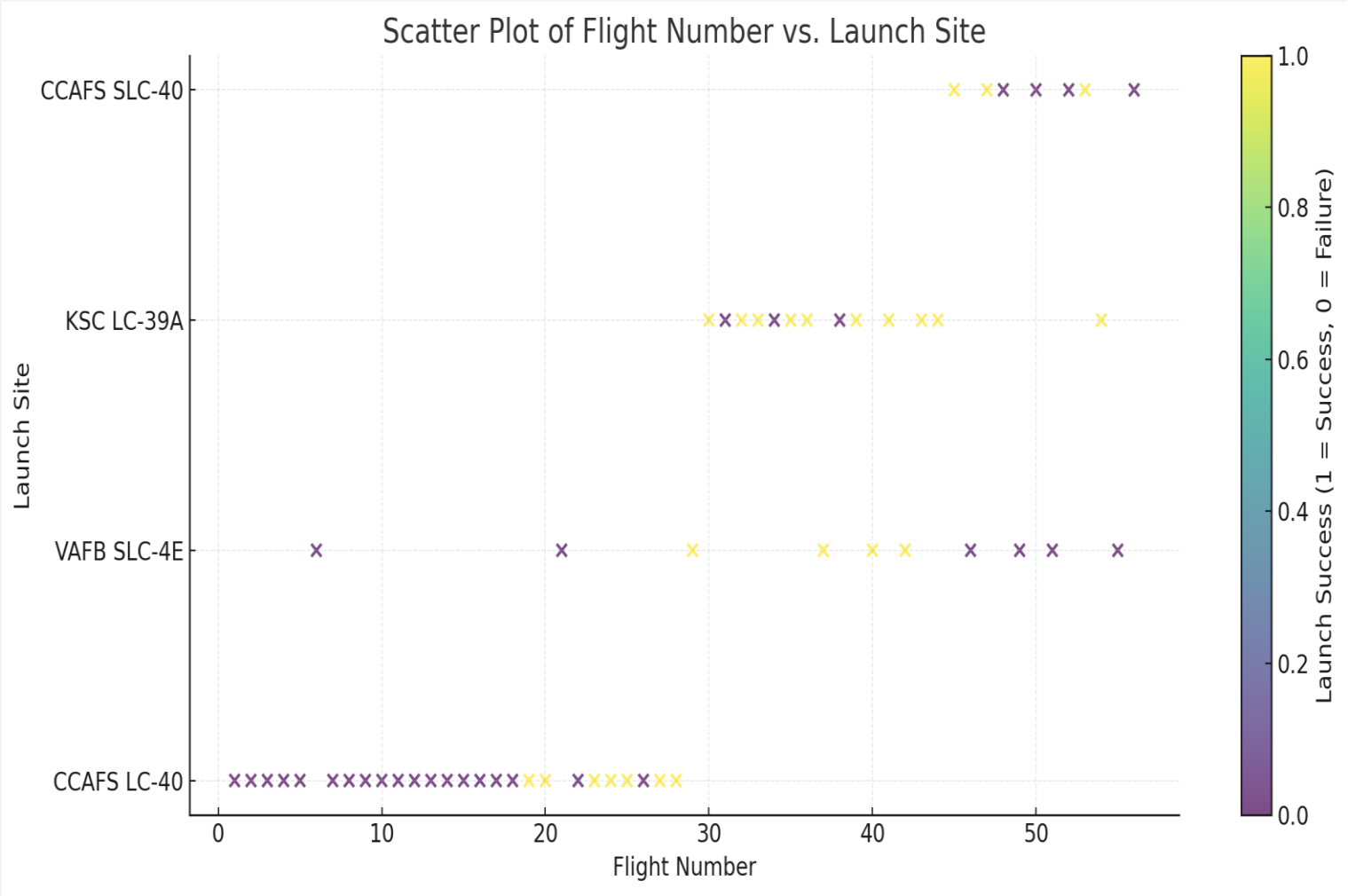
Results

- **Flowchart (Key Phrases):**
- **Data Preprocessing → Feature Engineering → Model Training (Logistic Regression, Random Forest, SVM) → Evaluation (Metrics and Confusion Matrix) → Hyperparameter Tuning → Best Model Selection.**


Section 2
Insights drawn from EDA

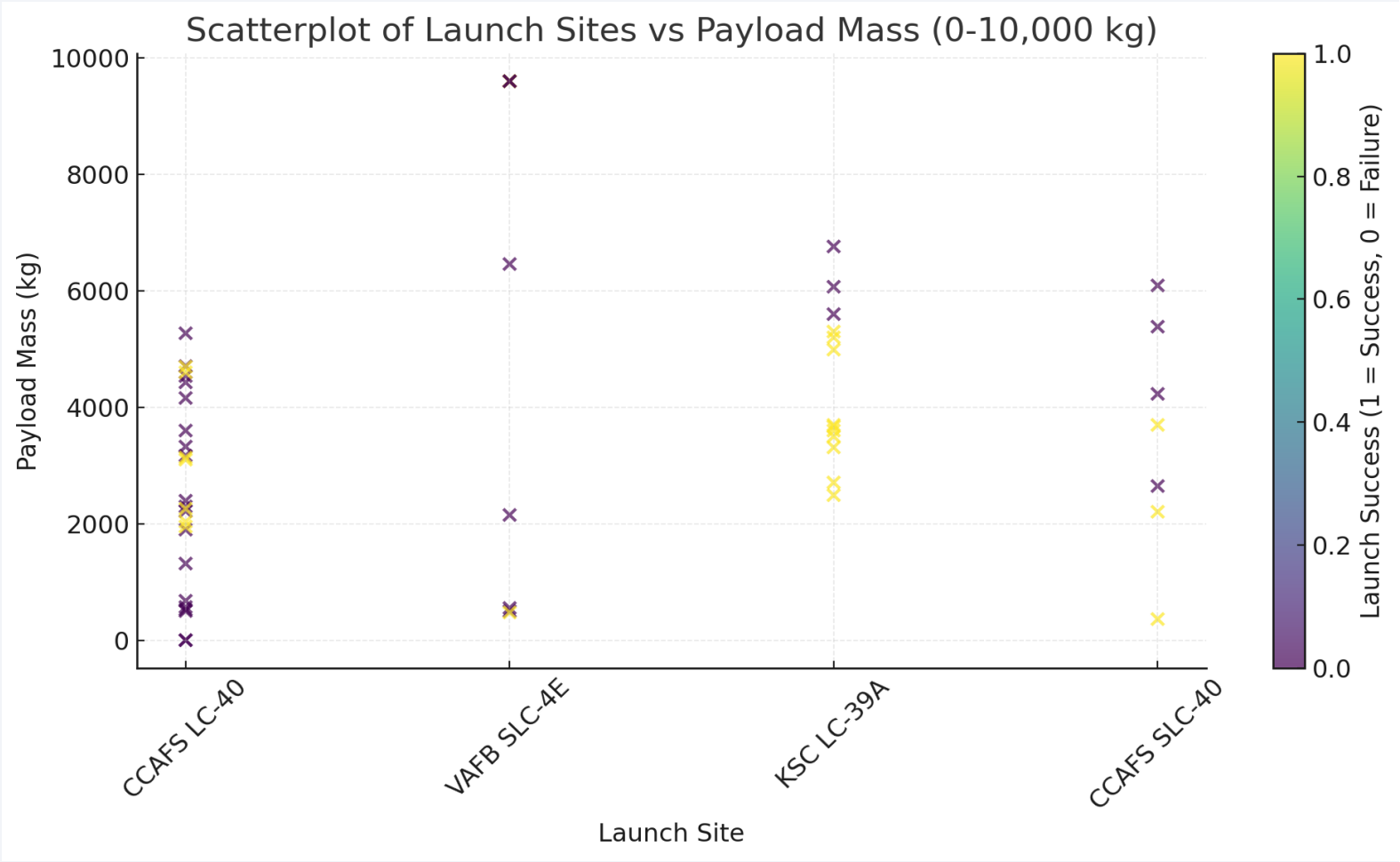
Flight Number vs. Launch Site

- The scatter plot to the right displays the relationship between flight numbers and launch sites. The x-axis represents the flight numbers, and the y-axis indicates the launch sites. Each point is colored based on the success of the launch (1 for success, 0 for failure), as shown in the color bar.



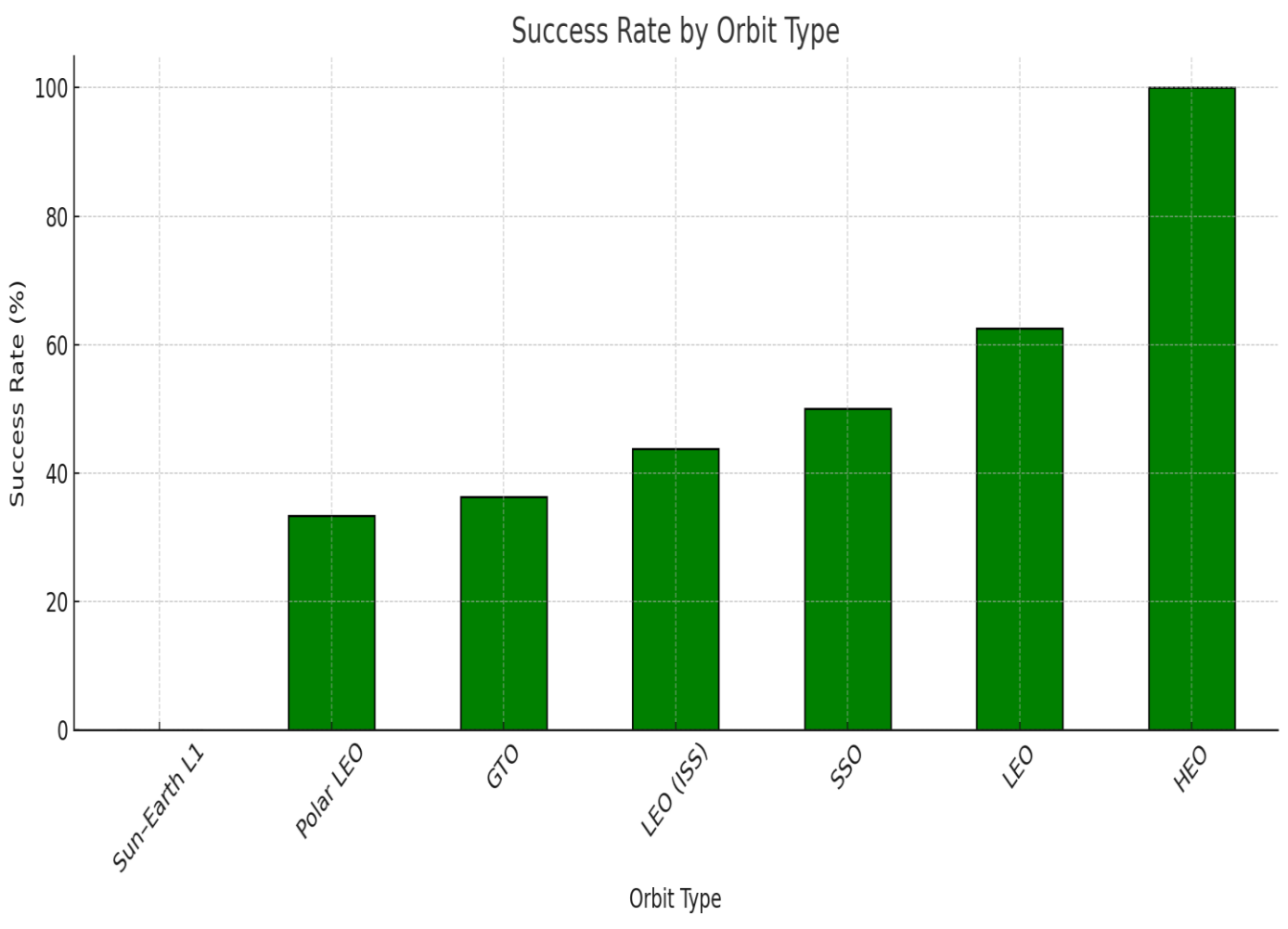
Payload vs. Launch Site

- Here is the scatterplot showing launch sites on the x-axis and payload mass (0–10,000 kg) on the y-axis. The color of the points indicates the success or failure of the launches, with the color bar representing the launch success classification.



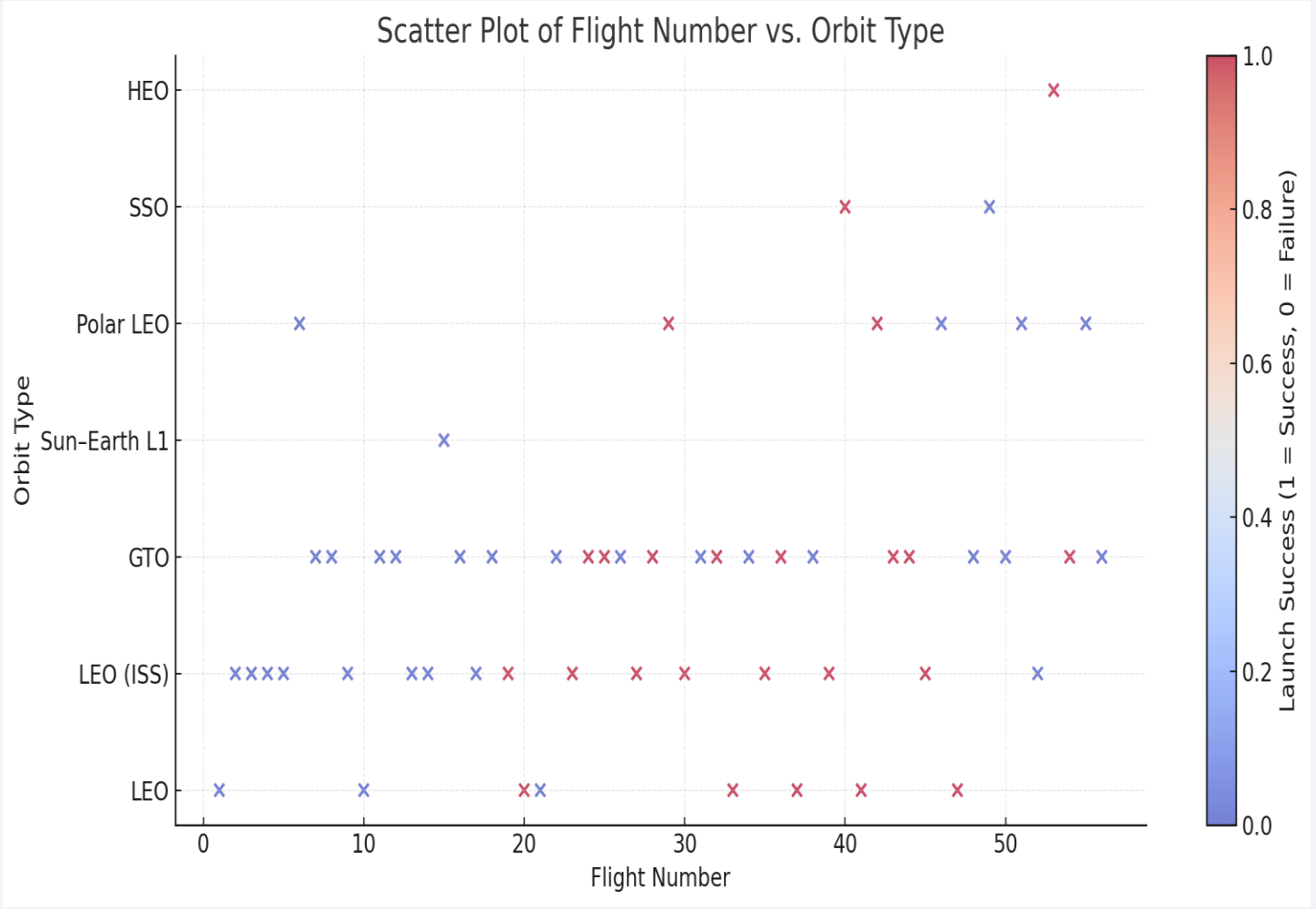
Success Rate vs. Orbit Type

- The bar chart shows the success rate for each orbit type. Each bar represents the percentage of successful launches for a specific orbit type, highlighting the relative reliability of launches targeting different orbits.



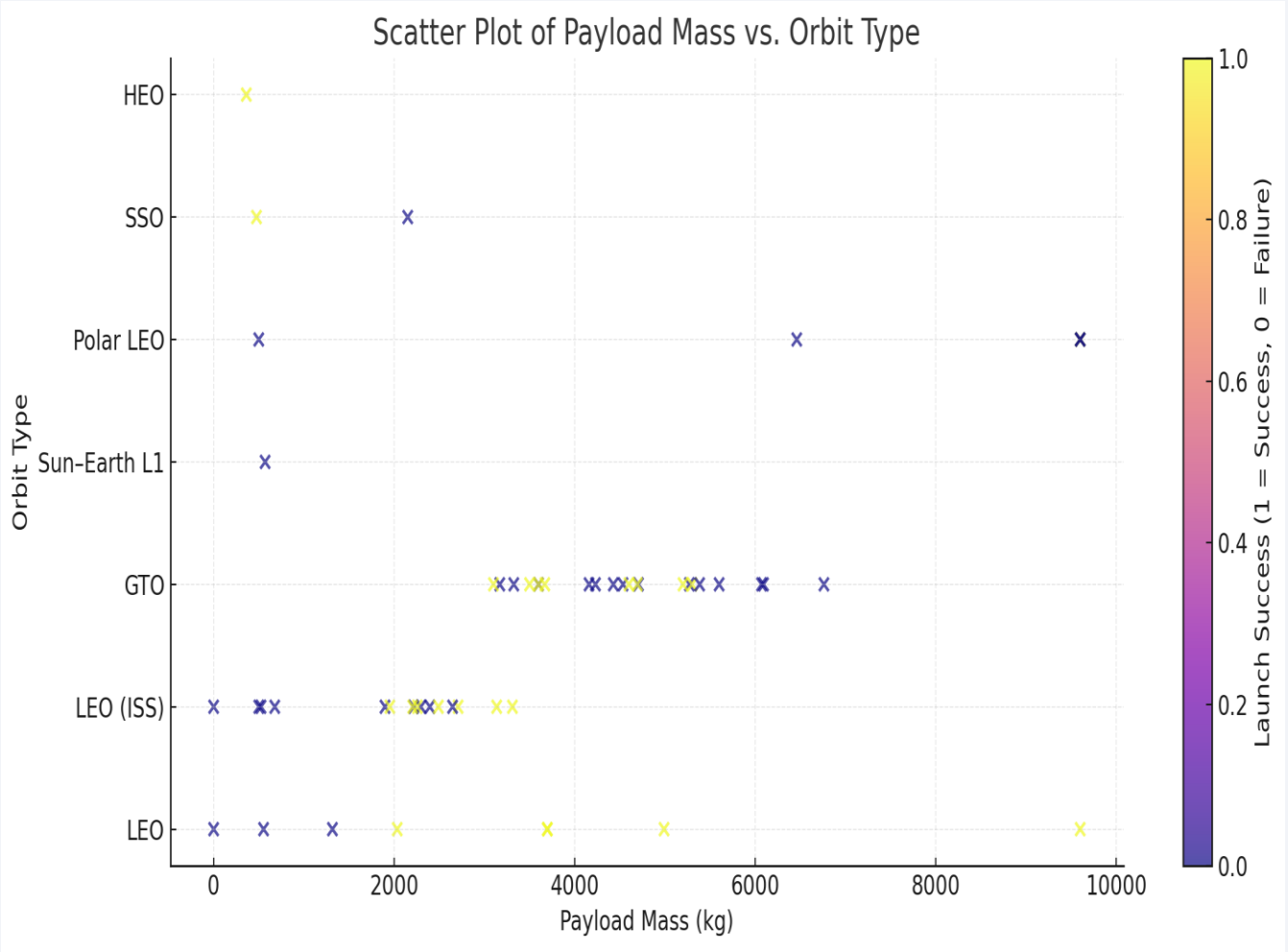
Flight Number vs. Orbit Type

- The scatter plot shows the relationship between flight numbers and orbit types. Each point represents a flight, with the x-axis showing the flight number and the y-axis indicating the orbit type. The color of the points corresponds to launch success (1 for success, 0 for failure), as indicated by the color bar. This visualization helps identify patterns or trends in the success of launches for different orbits over time.



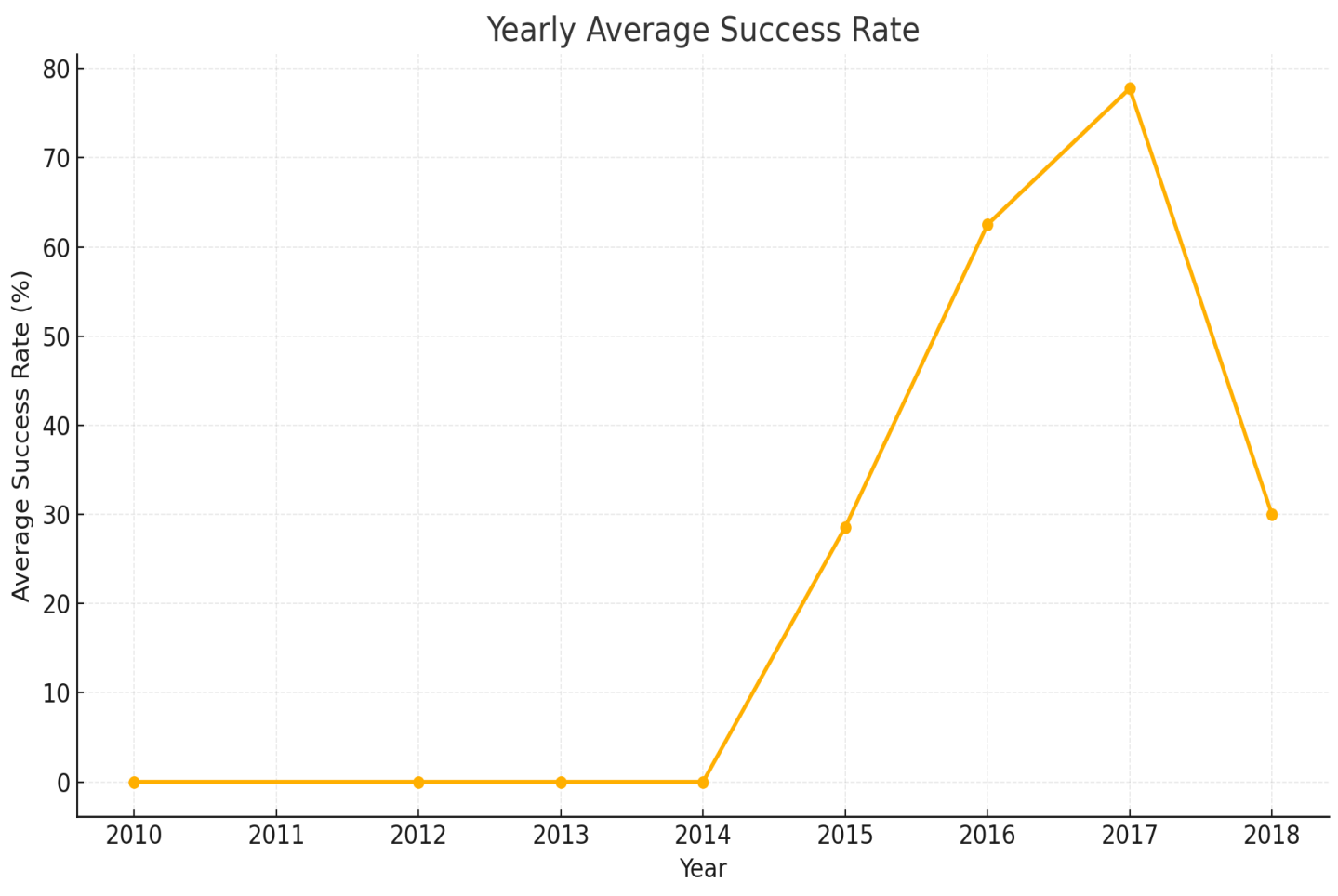
Payload vs. Orbit Type

- The scatter plot illustrates the relationship between payload mass and orbit type. The x-axis represents the payload mass in kilograms, and the y-axis indicates the orbit type. Each point is colored based on the launch success classification (1 for success, 0 for failure), as shown by the color bar.



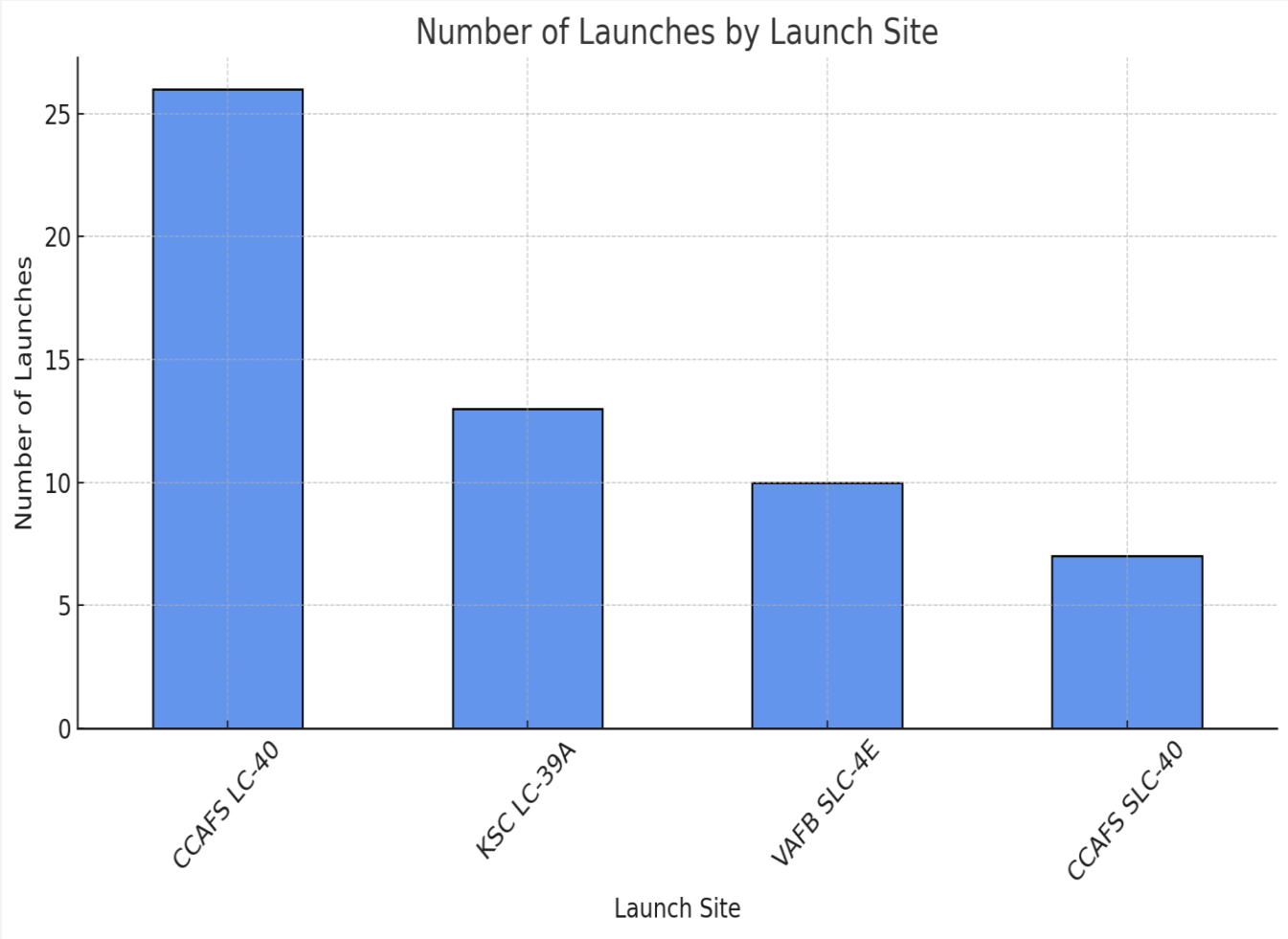
Launch Success Yearly Trend

- Here is the line chart showing the yearly average success rate of launches. The x-axis represents the year, and the y-axis displays the average success rate (as a percentage) for each year.



All Launch Site Names

- The bar chart shows the number of launches at each launch site.



Launch Site Names Begin with 'CCA'

- CCAFS LC-40
- CCAFS SLC-40
- These two launch sites appear multiple times with CCAFS LC -40 having the most launches and CCAFS SLC -40 having the fewest.

Total Payload Mass

- The total payload mass carried by boosters for NASA is 107,010 kg

```
data = pd.read_csv(file_path)

# Filter data for rows where the customer is NASA and calculate the total payload mass
nasa_payload_mass = data[data['Customer'].str.contains('NASA', na=False)]['PAYLOAD_MASS__K

# Display the result
print(f"Total payload mass carried by boosters for NASA: {nasa_payload_mass} kg")
```

Average Payload Mass by F9 v1.1

- The average payload mass carried by the booster version F9 v1.1 is 2928.4 kg.

```
data = pd.read_csv(file_path)

# Filter data for rows where the booster version is 'F9 v1.1'
f9_v1_1_data = data[data['Booster_Version'] == 'F9 v1.1']

# Calculate the average payload mass for 'F9 v1.1'
average_payload_mass_f9_v1_1 = f9_v1_1_data['PAYLOAD_MASS__KG_'].mean()

# Display the result
print(f"The average payload mass carried by booster version F9 v1.1 is: {average_payload_m
```

First Successful Ground Landing Date

- December 22, 2015.

```
data = pd.read_csv(file_path)

# Filter for successful landings on a ground pad
successful_ground_pad = data[
    (data['Landing_Outcome'].str.contains('Success', na=False)) &
    (data['Landing_Outcome'].str.contains('ground pad', na=False))
]

# Find the date of the first successful landing on a ground pad
first_successful_ground_pad_date = successful_ground_pad['Date'].min()

# Display the result
print(f"The first successful landing on a ground pad occurred on: {first_successful_ground
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
# Filter data for successful landings on a drone ship and payload mass in the specified ra
successful_drone_ship = data[
    (data['Landing_Outcome'].str.contains('Success', na=False)) &
    (data['Landing_Outcome'].str.contains('drone ship', na=False)) &
    (data['PAYLOAD_MASS__KG_'] > 4000) &
    (data['PAYLOAD_MASS__KG_'] < 6000)
]

# Extract the names of boosters
successful_boosters = successful_drone_ship['Booster_Version'].unique()

# Display the result
print("Boosters that successfully landed on a drone ship with payload mass between 4000 ar
```

Total Number of Successful and Failure Mission Outcomes

- Success: 98
- Failure (in flight): 1
- Success (payload status unclear): 1
- Present your query result with a short explanation here

```
data = pd.read_csv(file_path)

# Calculate the total number of successful and failed mission outcomes
mission_outcome_counts = data['Mission_Outcome'].value_counts()

# Display the result
print("Mission Outcomes:")
print(mission_outcome_counts)
```

Boosters Carried Maximum Payload

- The boosters that have carried the maximum payload mass are: 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

```
data = pd.read_csv(file_path)

# Find the maximum payload mass
max_payload_mass = data['PAYLOAD_MASS__KG_'].max()

# Filter data to find the boosters that carried the maximum payload mass
boosters_with_max_payload = data[data['PAYLOAD_MASS__KG_'] == max_payload_mass]['Booster_\

# Display the result
print("Boosters that carried the maximum payload mass:")
print(boosters_with_max_payload)
```

2015 Launch Records

- In the year 2015, the failed landing outcomes on a drone ship, along with their booster versions and launch site names, are as follows:
  - Booster Version: F9 v1.1 B1012 Launch Site: CCAFS LC-40 Landing Outcome: Failure (drone ship)
  - Booster Version: F9 v1.1 B1015 Launch Site: CCAFS LC-40 Landing Outcome: Failure (drone ship)

```
data = pd.read_csv(file_path)

# Filter data for failed landings on drone ships in the year 2015
failed_drone_ship_2015 = data[
    (data['Landing_Outcome'].str.contains('Failure', na=False)) &
    (data['Landing_Outcome'].str.contains('drone ship', na=False)) &
    (pd.to_datetime(data['Date']).dt.year == 2015)
]

# Select the relevant columns: Booster Version, Launch Site, and Landing Outcome
failed_drone_ship_2015_info = failed_drone_ship_2015[['Booster_Version', 'Launch_Site', 'L

# Display the result
print("Failed landing outcomes on drone ships in 2015:")
```

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

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

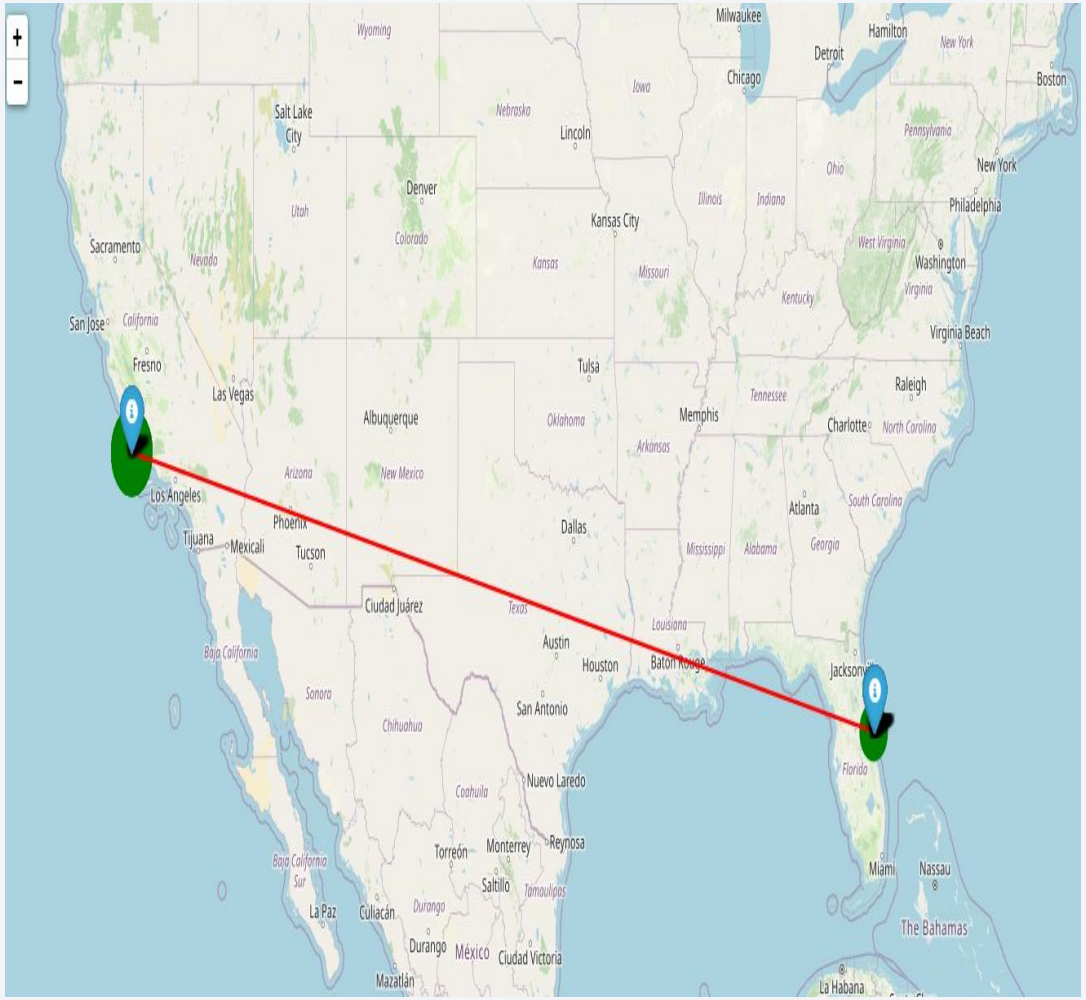
```
# Convert 'Date' column to datetime format
data['Date'] = pd.to_datetime(data['Date'])

# Filter data for the date range between 2010-06-04 and 2017-03-20
filtered_data = data[(data['Date'] >= '2010-06-04') & (data['Date'] <= '2017-03-20')]

# Rank the count of landing outcomes in descending order
landing_outcome_counts = filtered_data['Landing_Outcome'].value_counts()

# Display the result
print("Ranked count of landing outcomes between 2010-06-04 and 2017-03-20:")
print(landing_outcome_counts)
```

Section 3

Launch Sites Proximities Analysis

Launch Site Markers

- This map shows different, successful launch sites.

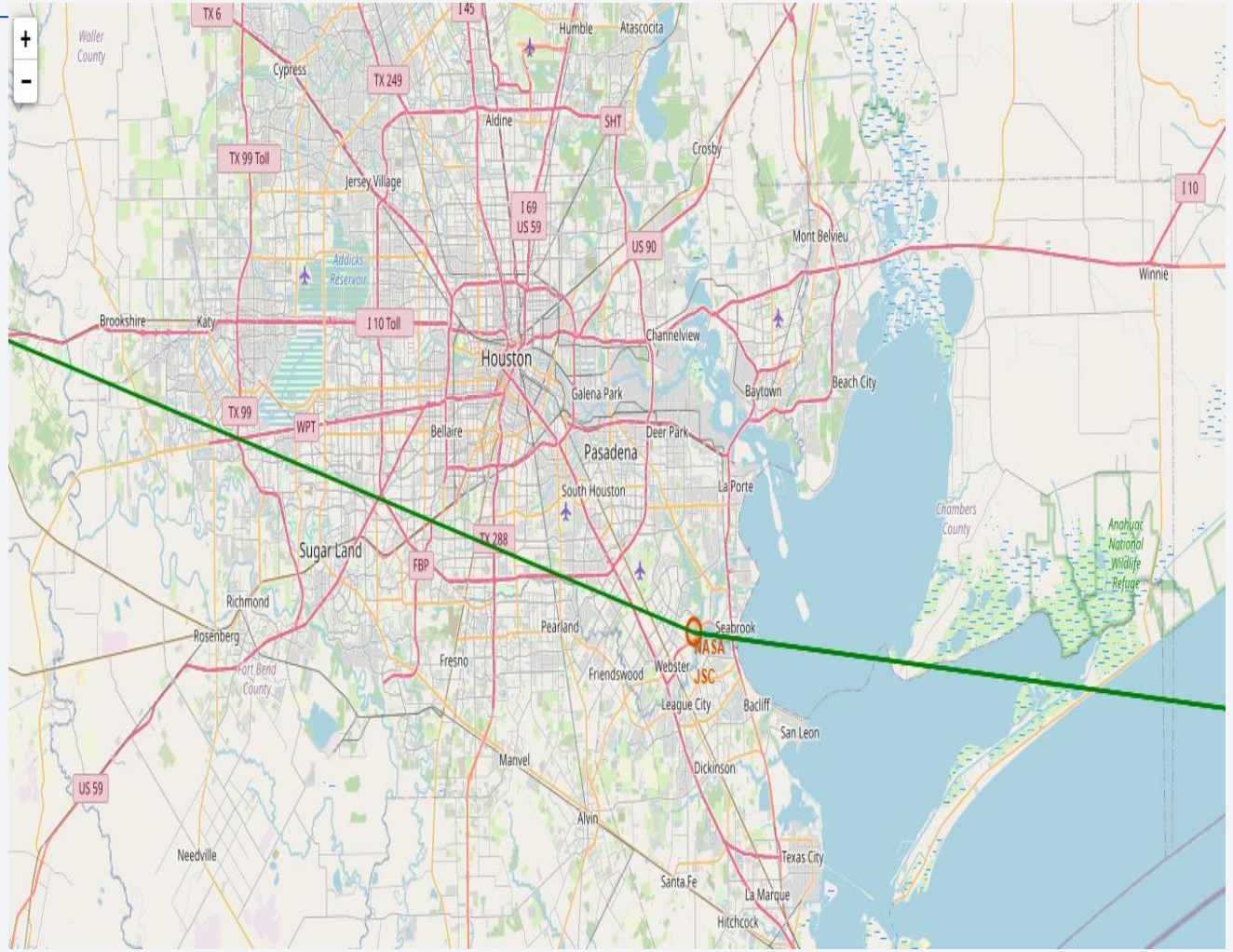


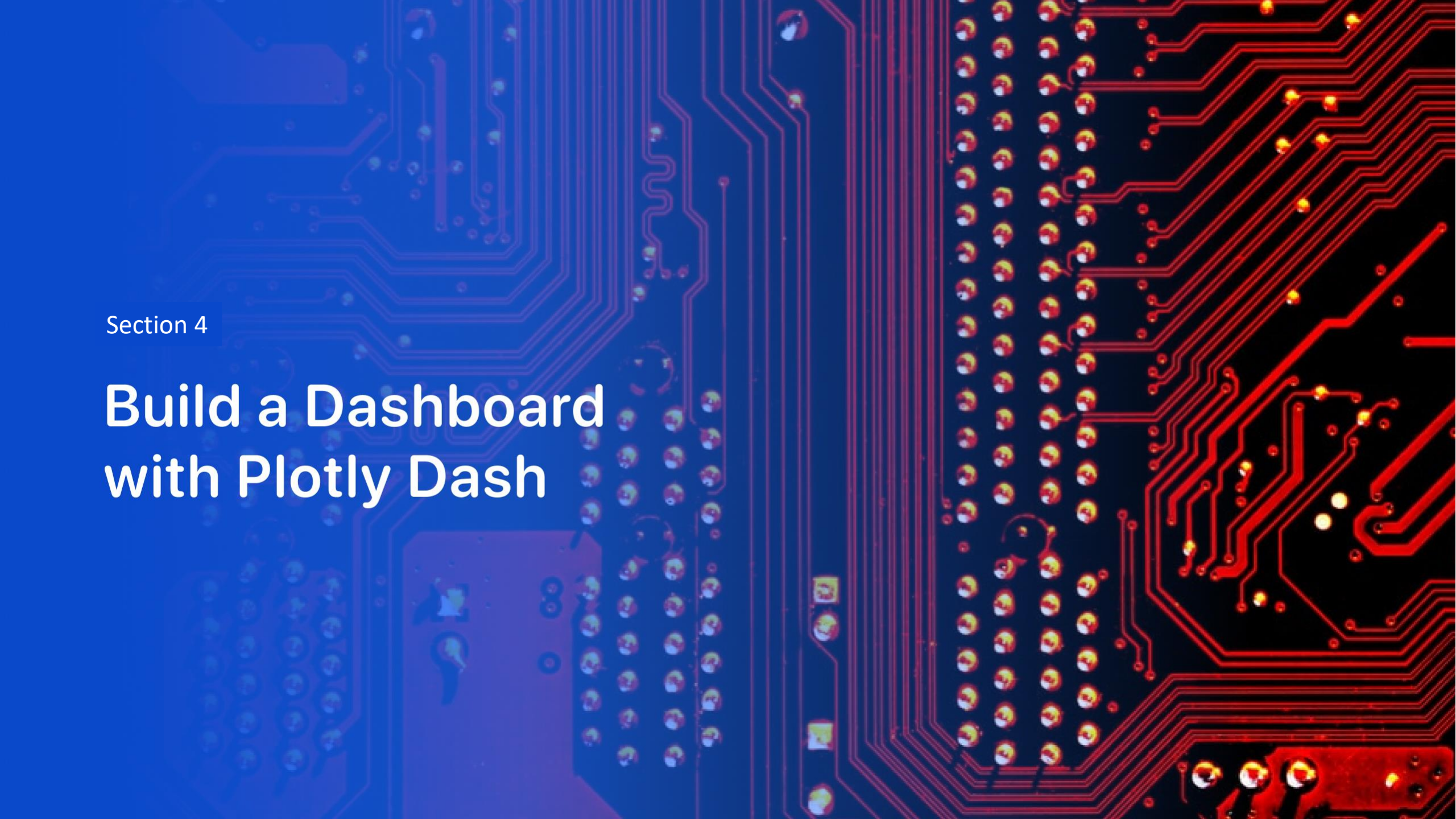
<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

Key Location Map

- This map shows key locations related to a launch site.



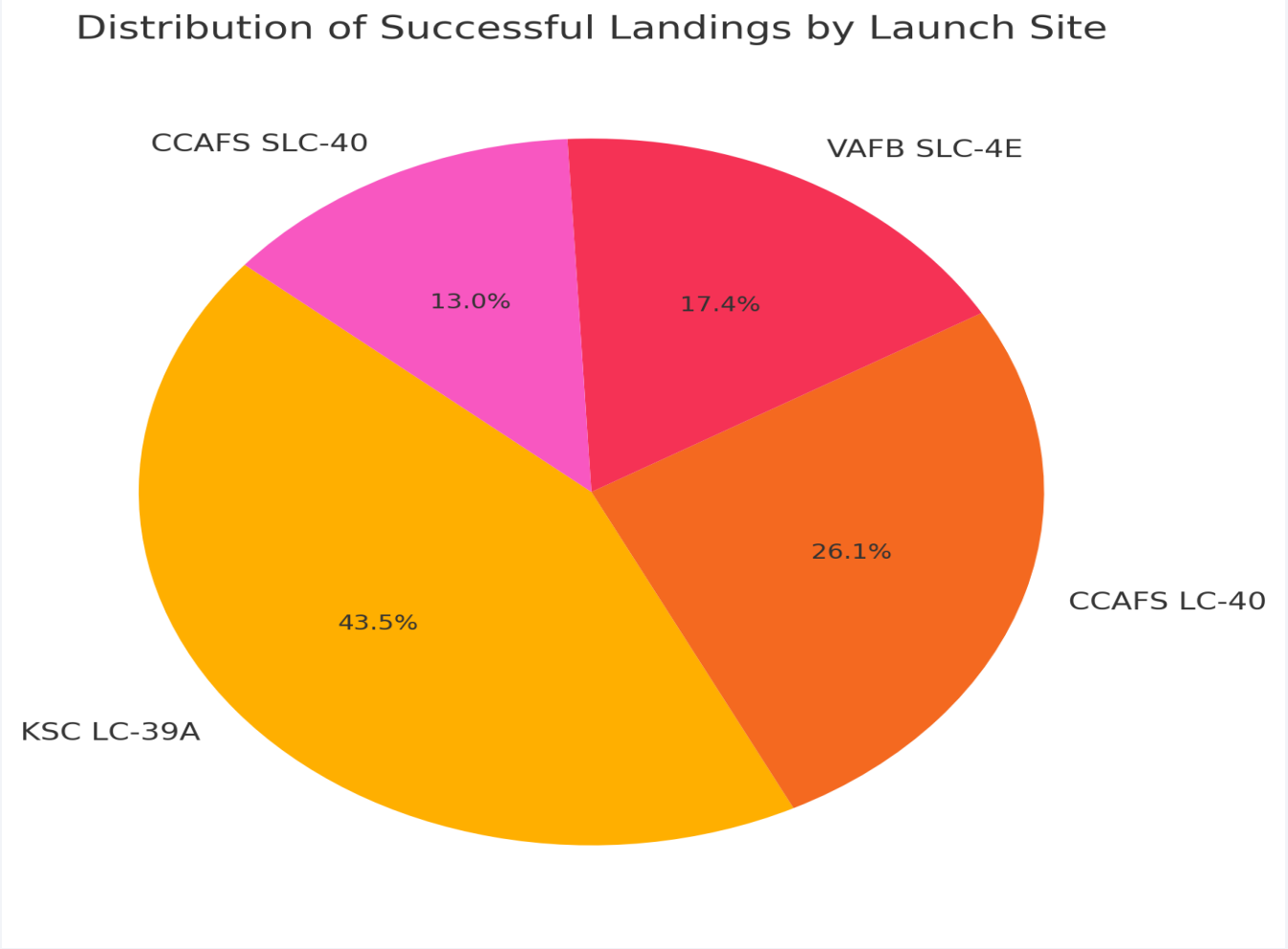

Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

Pie Chart of Successful Landings by Launch Sites

- KSC LC -39A had the highest percentage of successful landings by a wide margin.
- CCAFS SLC -40 had the lowest percentage of successful landings.



<Dashboard Screenshot 3>

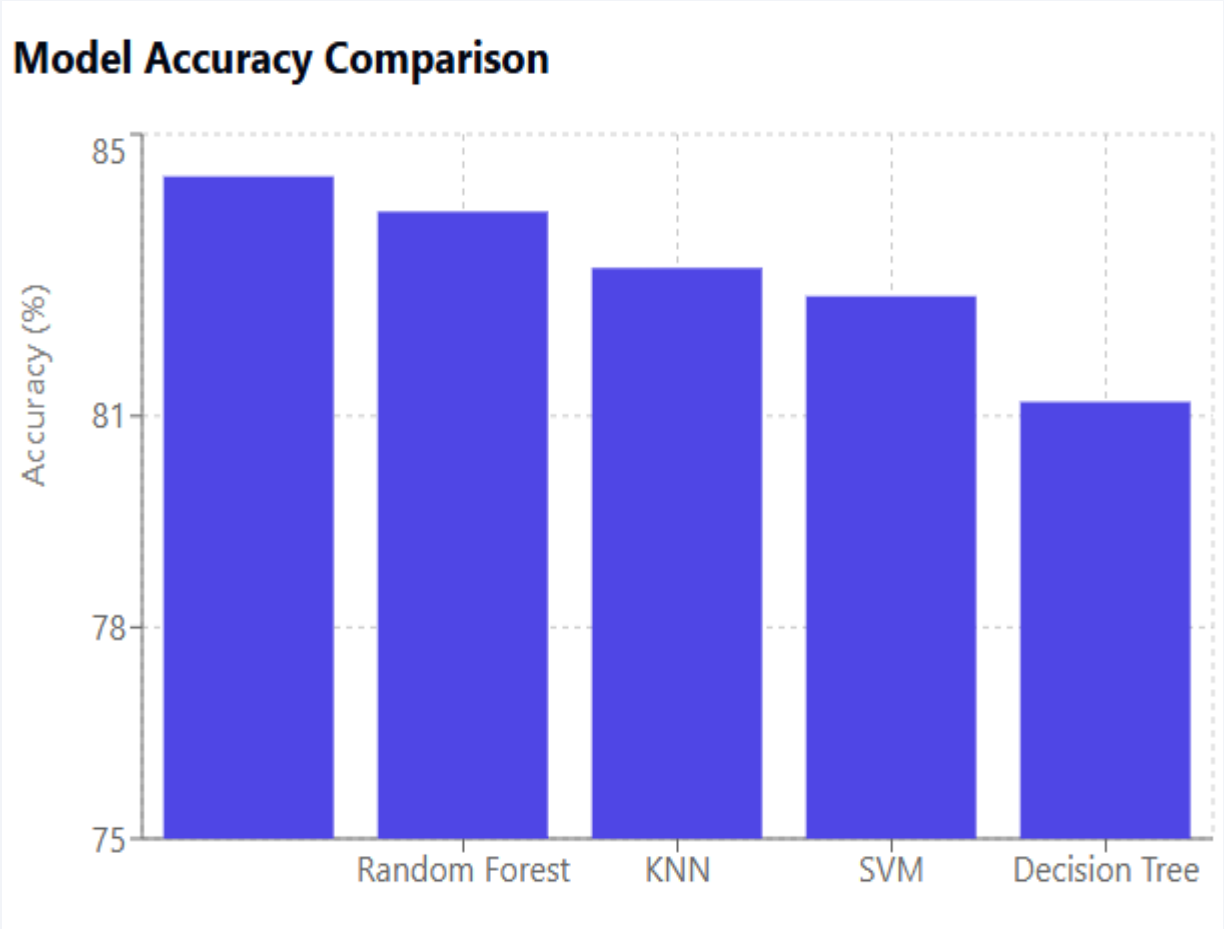
- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

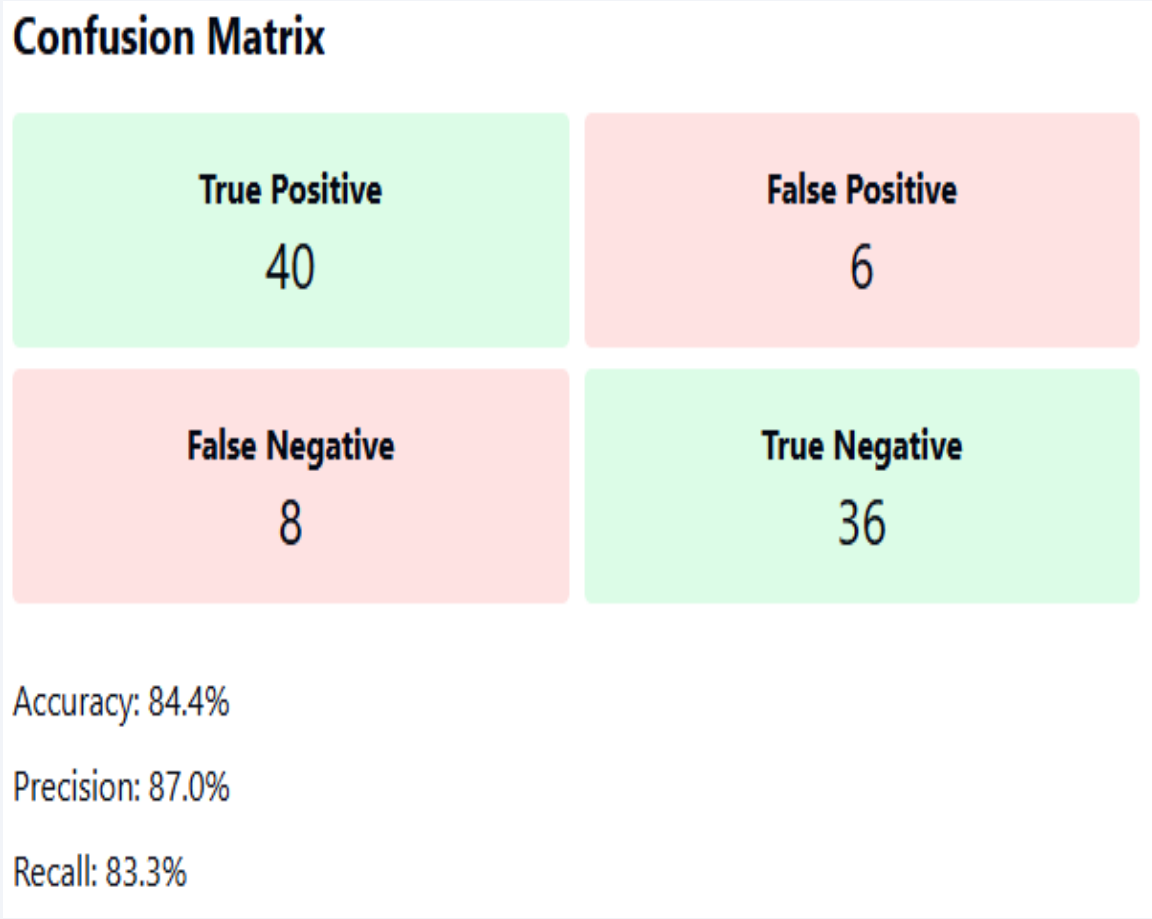
Classification Accuracy

- The logistic regression model has the highest accuracy (84.4%); however, all models have over an 81% accuracy.



Confusion Matrix

- The confusion matrix shows an accuracy rate of 84.4%, which matches the logistic regression data from the previous slide's graph.
- The precision rate of 87.0%.



Conclusions

- Predict if Stage 1 will land successfully or fail
- Data from API and Wikipedia (SpaceX) were used.
- Best launch site: KSC LC 39-A.
- Launches with a payload of 7000kg or higher were more likely to be successful.
- The first successful launch was in 2015.
- Recently, there have been many more successful launches.

Appendix

- GitHub Repository

Thank you!