# Winning Space Race with Data Science

Ektoras Delaportas
29/1/2025

# Outline

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

# Executive Summary

- Summary of methodologies: The project aimed to predict the first stage landing success of SpaceX's Falcon 9 rocket using a comprehensive data science pipeline. Raw data was sourced through API calls and web scraping from Wikipedia, followed by data wrangling to clean and prepare the dataset. Exploratory Data Analysis (EDA) was performed using Python, SQL queries, and advanced visualization techniques like Folium and Plotly Dash to uncover key insights. Feature engineering techniques were applied to structure data for predictive modeling. Finally, machine learning models such as SVM, classification trees, KNN and logistic regression were implemented and evaluated to determine the best-performing method.
- Summary of all results: The analysis revealed significant correlations between launch success rates and factors such as payload mass, orbit type, and launch site location. Interactive maps highlighted geographic patterns in launch outcomes, while machine learning models successfully classified first-stage landing outcomes. Among the tested methods, all the models demonstrated the same high predictive accuracy, providing a reliable approach to assess the feasibility of rocket reusability. These insights can optimize cost predictions for SpaceX launches and support competitive bidding by alternate aerospace companies.

# Introduction

- Project background and context: SpaceX's Falcon 9 rockets are revolutionizing the aerospace industry by significantly reducing launch costs through the reusability of their first stages. A Falcon 9 launch costs approximately $62 million, compared to $165 million for competitors. Determining the likelihood of a successful first-stage landing is crucial for understanding cost efficiency and enabling informed decision-making for both SpaceX and potential competitors in the aerospace market.
- Problems you want to find answers: This project aims to answer key questions, such as: What factors influence the success of a first-stage landing? Can predictive models reliably classify landing outcomes based on historical data? By addressing these questions, the study seeks to uncover actionable insights to optimize launch strategies and enable competitive bidding in the industry.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - API calls with requests and web scraping with BeautifulSoup.
- Perform data wrangling:
  - Preprocessing, cleaning, handling missing values and reformatting. Key features extraction and engineering.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models:
  - SVM, decision trees, KNN and logistic regression, GridSearchCV, and Accuracy with Score.

# Data Collection

- Data was collected from multiple sources, including SpaceX’s API and web scraping. API calls retrieved launch data directly from SpaceX, while BeautifulSoup was used to scrape Falcon 9 and Falcon Heavy launch records from Wikipedia. The collected data was cleaned and formatted to ensure consistency and usability.

# Data Collection – SpaceX API

- **Step 1:** Define helper functions that use the API to extract information from the launch data.
- **Step 2:** Use GET request to retrieve and parse the SpaceX launch data
- **Step 3:** Filter the dataframe to only include Falcon 9 launches
- **Step 4:** Dealing with Missing Values
- GitHub link for data collection with REST API

Send HTTP Request:
Initiate a request to Wikipedia requests.

↓

Receiving the Response: Receiving JSON-formatted data from the API.

↓

Parsing the Response: Extracting relevant data fields (e.g., launch date, success status).

↓

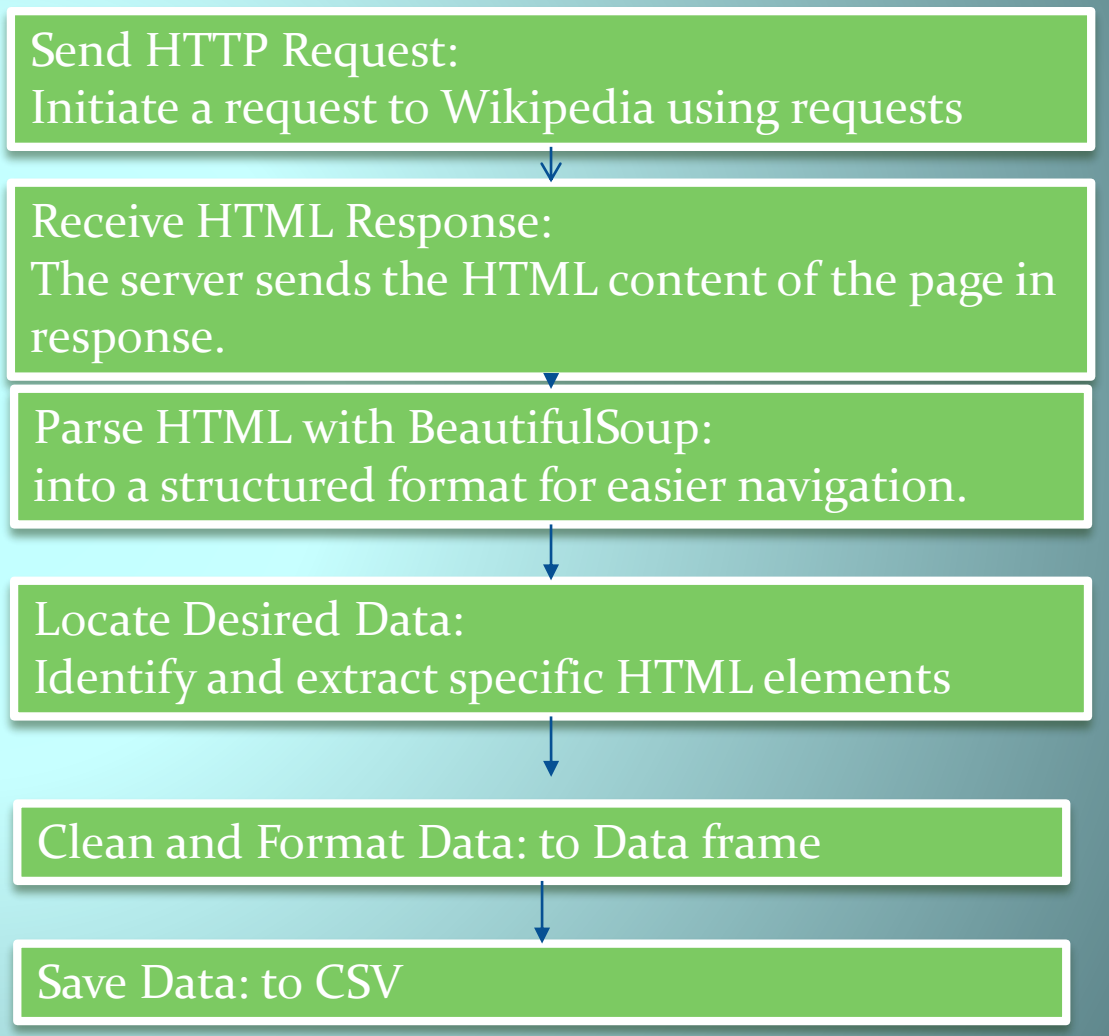
Data Cleaning and Formatting: Handling missing or inconsistent data and organizing it into a structured format.

↓

Storing the Data: Saving the cleaned data into a CSV file

# Data Collection - Scraping

- **Step 1:** Define helper functions to process web scraped HTML table
- **Step 2:** Use GET request to retrieve Falcon9 data and create BeautifulSoup from JSON response
- **Step 3:** Extract all variable names from the HTML table header
- **Step 4:** Create a data frame by parsing the launch HTML tables
- GitHub link for data collection with Webscraping



# Data Wrangling

- The raw data underwent preprocessing, including cleaning, handling missing values, and reformatting to align with analytical needs. Key features were extracted and engineered to facilitate accurate modeling and visualization:
- **Step 1:** Calculate the number of launches on each site
- **Step 2:** Calculate the number and occurrence of each orbit
- **Step 3:** Calculate the number and occurrence of mission outcome of the orbits
- **Step 4:** Create a landing outcome label from Outcome column
- GitHub link for Data Wrangling

# EDA with Data Visualization (1/2)

- EDA was conducted to uncover patterns, relationships, and trends in the data. This was achieved through the Data Visualization Seaborn, providing insights into key factors influencing rocket landing success, which guided the following Feature engineering process. Graphs:

i. **FlightNumber vs. PayloadMassand:** and overlay the outcome of the launch. We see that as the flight number increases, the first stage is more likely to land successfully.

ii. **FlightNumber vs LaunchSite:** and overlay the outcome of the launch. Again, the first stage is more likely to land, for each landing site, as the number of flights increases.

iii. **Payload Mass Vs. Launch Site:** and overlay the outcome of the launch. **S**ome sites are used only for a specific range of Payload Mass (e.g. VAFB <10K). Also, there are favored ranges of payload mass for each site (e.g. KSC : 2-4K)

iv. **Class Vs. Orbit:** some orbits have highest success rates (e.g. ES-L1, SSO, HEO, GEO)

# EDA with Data Visualization (2/2)

v. **FlightNumber vs Orbit**: some orbits are related to number of flights (e.g. LEO )

vi. **Payload Mass vs Orbit:** Polar,LEO and ISS have grater success rate for heavy payloads

vii. **Success Rate per Year:** since 2013 it kept rising till 2020

- **Feature Engineering:** Created dummy variables to categorical columns (Orbits, LaunchSite, LandingPad, and Serial)
- **Cast to float**
- **Save Data to CSV**
- GitHub link for EDA and data visualization

# EDA with SQL (1/2)

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- Date when the first successful landing outcome in ground pad was achieved.
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Total number of successful and failure mission outcomes

# EDA with SQL (2/2)

- **Names of the booster_versions which have carried the maximum payload mass.**
- **Records which display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.**
- **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.**
- GitHub link for EDA with SQL

# Build an Interactive Map with Folium

- *Circle* was added to highlight NASA Johnson Space Center's with a text label on a specific coordinate.
- *Circle* and *Marker* was added to highlight each launch site on the site map.
- *MarkerCluster* object was created and a *Marker* was added to each site for every launch, based on their Class, to mark their success or not.
- *Markers* was added to proximities of launch sites and calculated corresponding distances, to findout how close the launch is to populated areas for safety reasons, transportation facilitation etc. *Lines* were added to designate the distance between launch sites and these proximities.
- GitHub link for Folium map

# Build a Dashboard with Plotly Dash

I. Interactions:
   a. Drop down menu to choose launch site or all sites
   b. Range Slider to choose the range of Payload Mass Kg

II. Graphs:
   a. Interactive Pie chart to display Total Successful Launches by Site
   b. Interactive Scatter plot to display Correlation between Payload and success for chosen Sites, colored against the booster version and payloads

- These graphs show which sites, booster versions and payloads are more likely to succeed
- GitHub link for Plotly Dash lab

# Predictive Analysis (Classification)

- Machine learning pipelines were developed to classify the likelihood of first-stage landings. Models, including SVM, decision trees, KNN and logistic regression, were trained, hyperparameter-tuned, and evaluated for performance. The best-performing model was selected based on test data accuracy and reliability.

1. Create target variable (Class)
2. Standardize the data (StandardScaler)
3. Split the data into training and test data (train_test_split)
4. Models are trained and hyperparameters are selected (GridSearchCV)
5. Evaluate each model by Calculating the accuracy on the test data (score)
6. Plot confusion matrix based on real and predicted classes

- GitHub link for predictive analysis lab

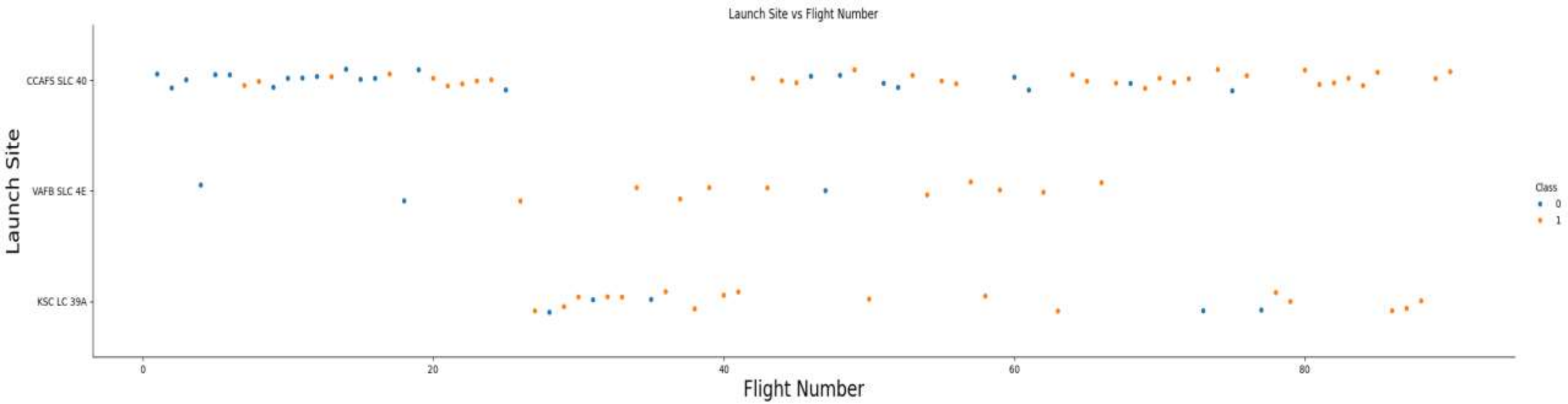
# Results

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

Section 2

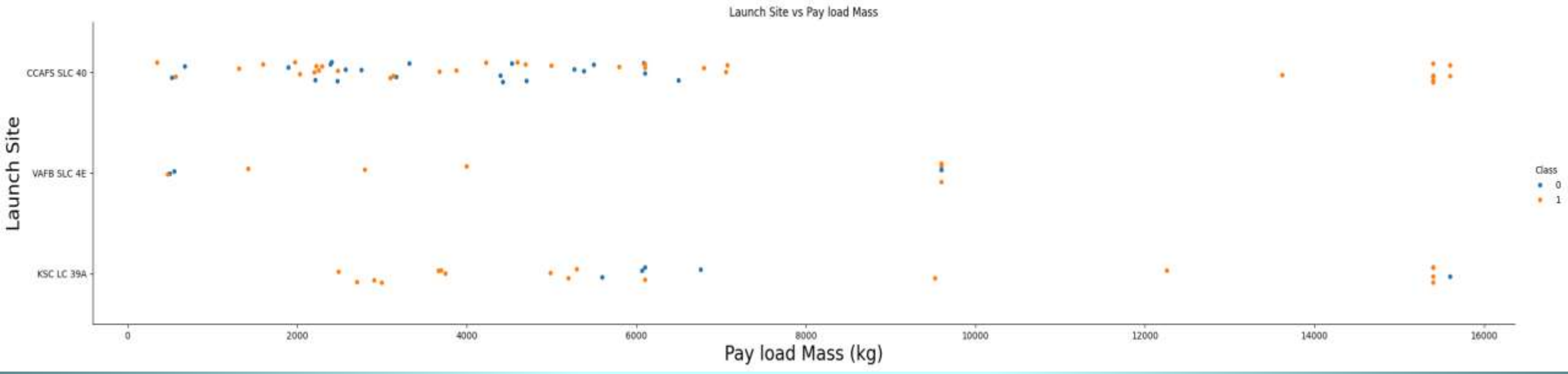
# Insights drawn from EDA

# Flight Number vs. Launch Site




- It is obvious that as the flight number increases the success rate for every launch site rises. This is due to the cumulative experience from previous launches and improved technology used as the time passes by.

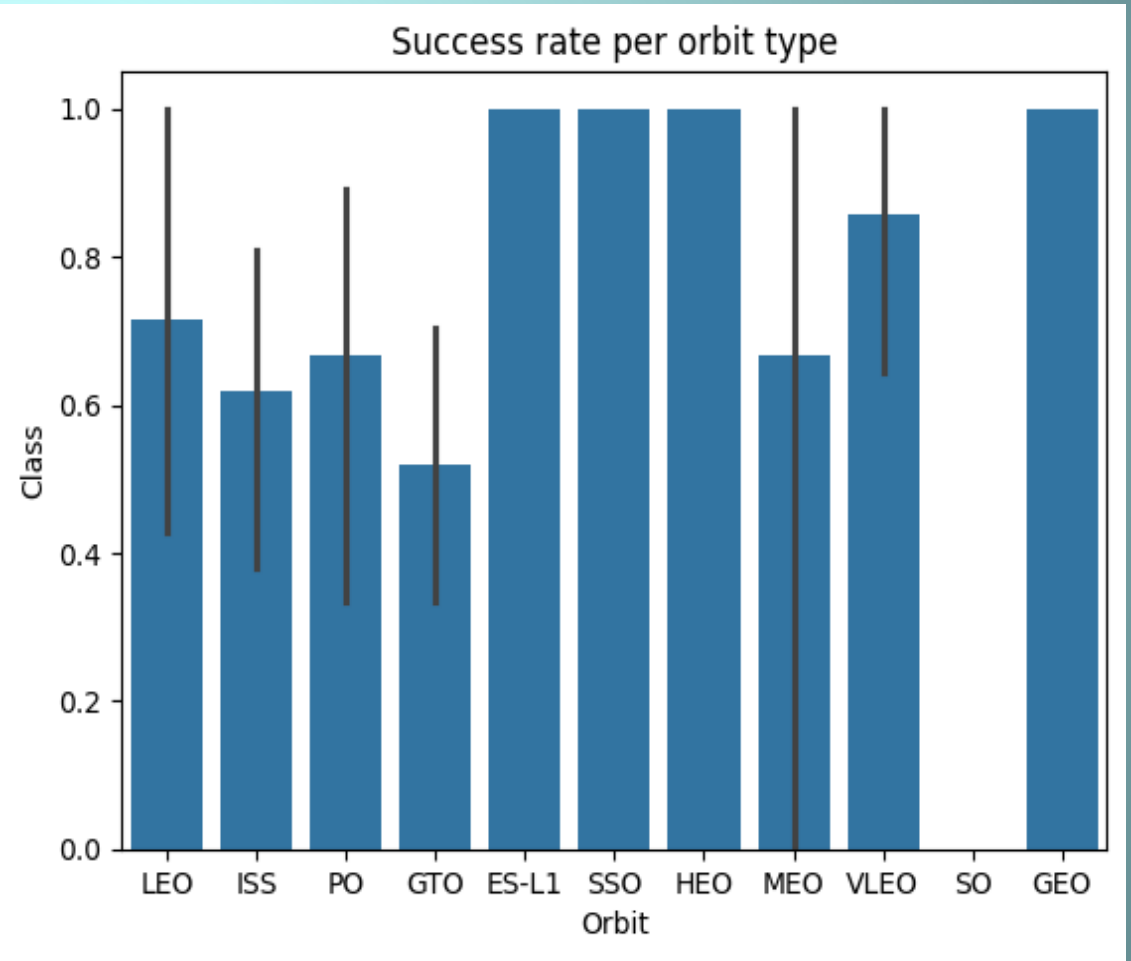
# Payload vs. Launch Site



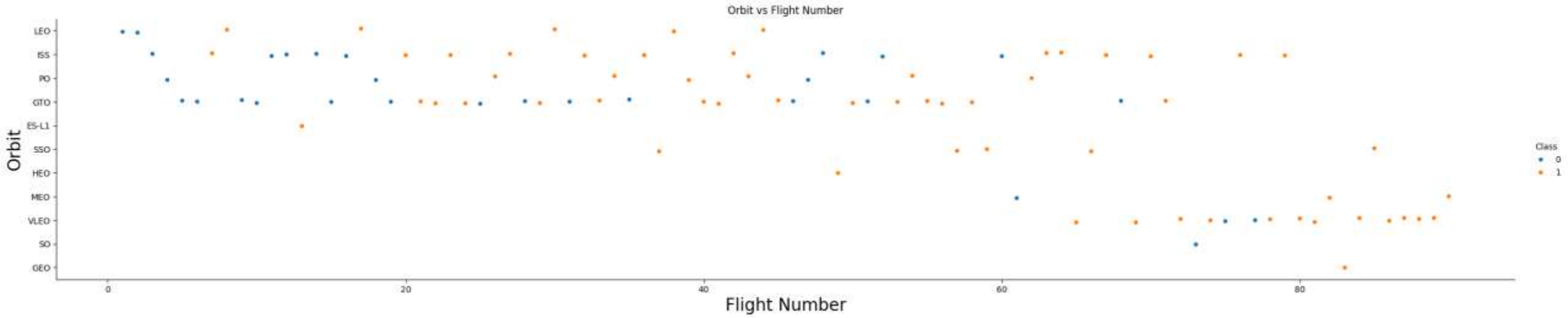

- As we see some sites are used only for a specific range of Payload Mass (e.g. VAFB <10K, CCAFS <8K & >14K).
- Also, there are favored ranges of payload mass for each rocket (e.g. KSC : 2-5K, VAFB: 1-4K, CCAFS>6K).

# Success Rate vs. Orbit Type

- Launches for specific orbits like ES-L1, SSO, HEO, GEO show greater success rate probably because of multiple previous attempts for these orbits. Also, strategic launch site selection for these orbit minimize the risks.

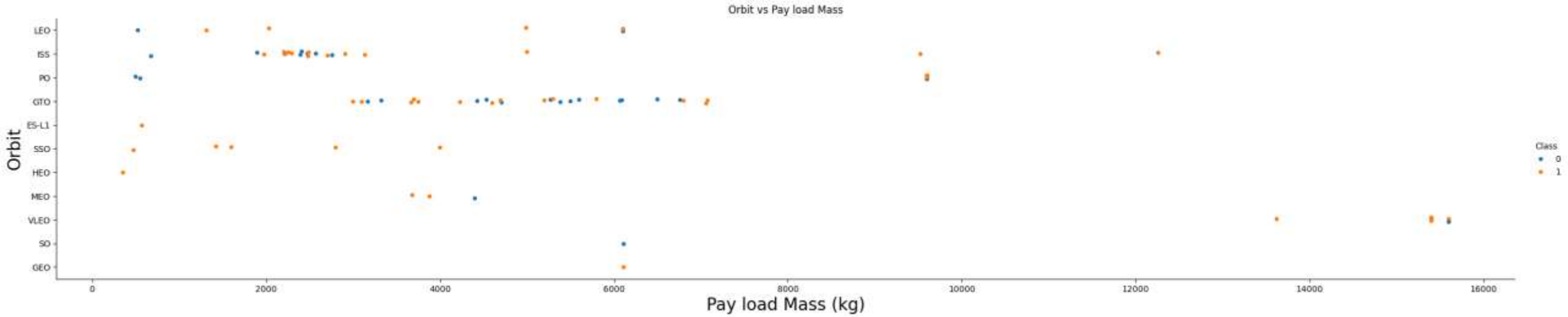


# Flight Number vs. Orbit Type




- Some orbits are related to number of flights: LEO shows a rising success rate as flights increase.
- The same goes for most of the flights.
- However, there are some exceptions like GTO where this is not apparent.

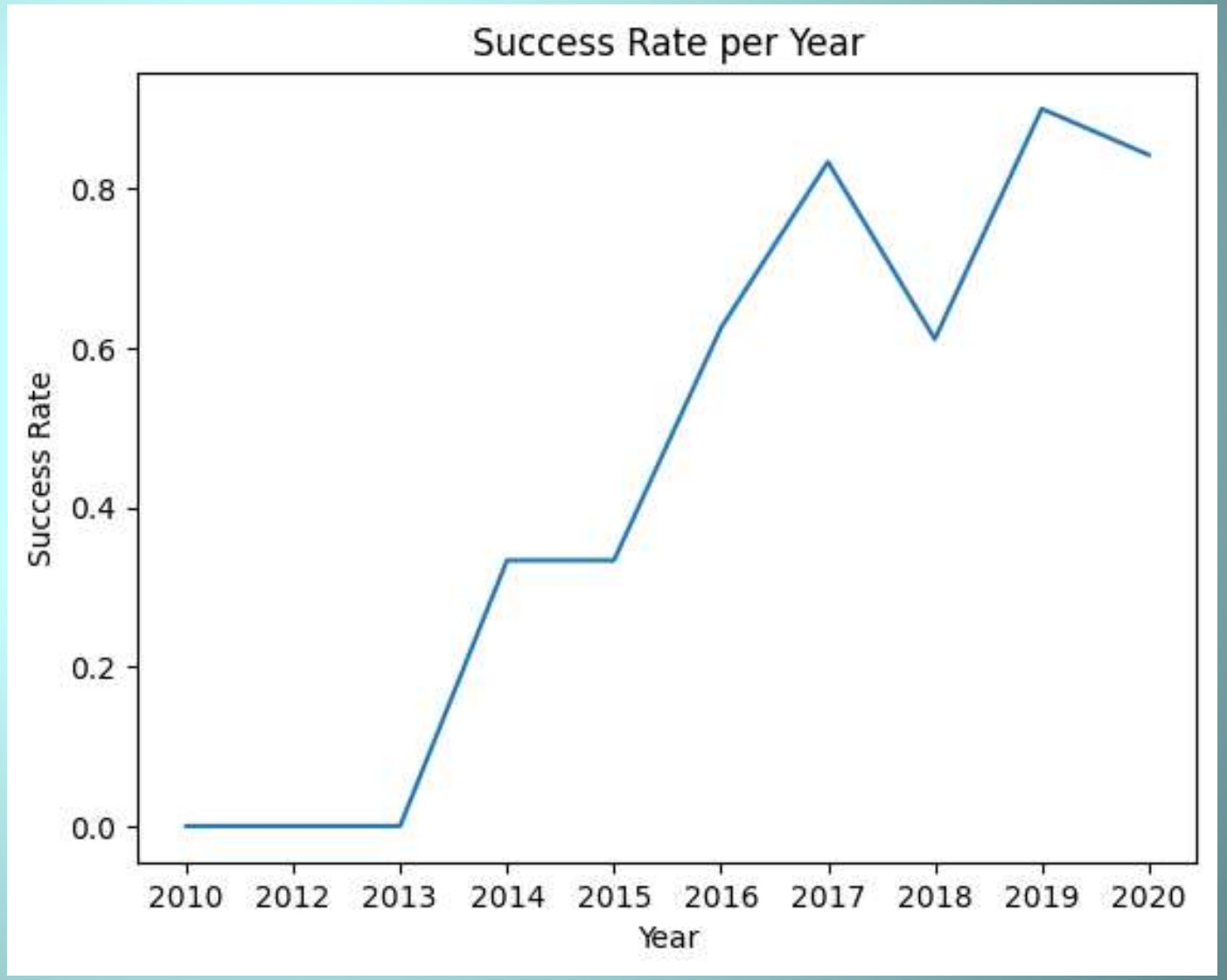
# Payload vs. Orbit Type




- Polar,LEO and ISS have grater success rate for heavy payloads.
- However, for GTO and others, this is not the case as it is difficult to distinguish between successful and unsuccessful landings, since both outcomes are present

# Launch Success Yearly Trend

- The success rate is rising from 2013 onwards, with a small drop in 2018.
- Experience and advances in technology play the leading role here



# All Launch Site Names

- I use DISTINCT here in order to take all the unique names

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

# Launch Site Names Begin with 'CCA'

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landi |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failur |
| 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 | Failur |
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | |

I use WHERE clause with LIKE to locate the exact name

# Total Payload Mass

| TOTAL_PAYLOAD_MASS |
| --- |
| 45596 |

- TOTAL_PAYLOAD_MASS = 45596
- SUM of Payload Mass is used together with a
- WHERE clause to locate NASA

# Average Payload Mass by F9 v1.1

| AVG_PAYLOAD_MASS |
| --- |
| 2535.0 |

- AVG_PAYLOAD_MASS = 2535.0
- AVG Payload Mass is used with
- WHERE clause and LIKE to find F9 v1.1

# First Successful Ground Landing Date

| FIRST_GROUND_LAND |
| --- |
| 2015-12-22 |

- MIN Date along with
- WHERE clause to locate 'Success (ground pad)'

# Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE clause to locate 'Success (drone ship)‘
- AND  4000<Payload Mass<6000

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

# Total Number of Successful and Failure Mission Outcomes

| Success | Failure |
|---|---|
| 100 | 1 |

- One Subquery for each outcome with
- WHERE clause and LIKE to distinguish between outcomes

# Boosters Carried Maximum Payload

- Subquery in WHERE clause
- SELECT MAX Payload Mass

| Booster_Version | Max_Mass |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

| Month_Name | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

- CASE with substr(Date, 6, 2) to replace each number of month with it's name
- WHERE clause to locate 2015
- AND LIKE to find Failure (drone ship)

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

- WHERE clause with BETWEEN '2010-06-04' AND '2017-03-20‘ to get range of dates
- GROUP BY Landing outcome to group all unique outcomes together
- and ORDER BY COUNT Landing outcome
- DESC to take the descending order

| Landing_Outcome | No_Outcomes |
|---|---|
| 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

# Launch sites Map

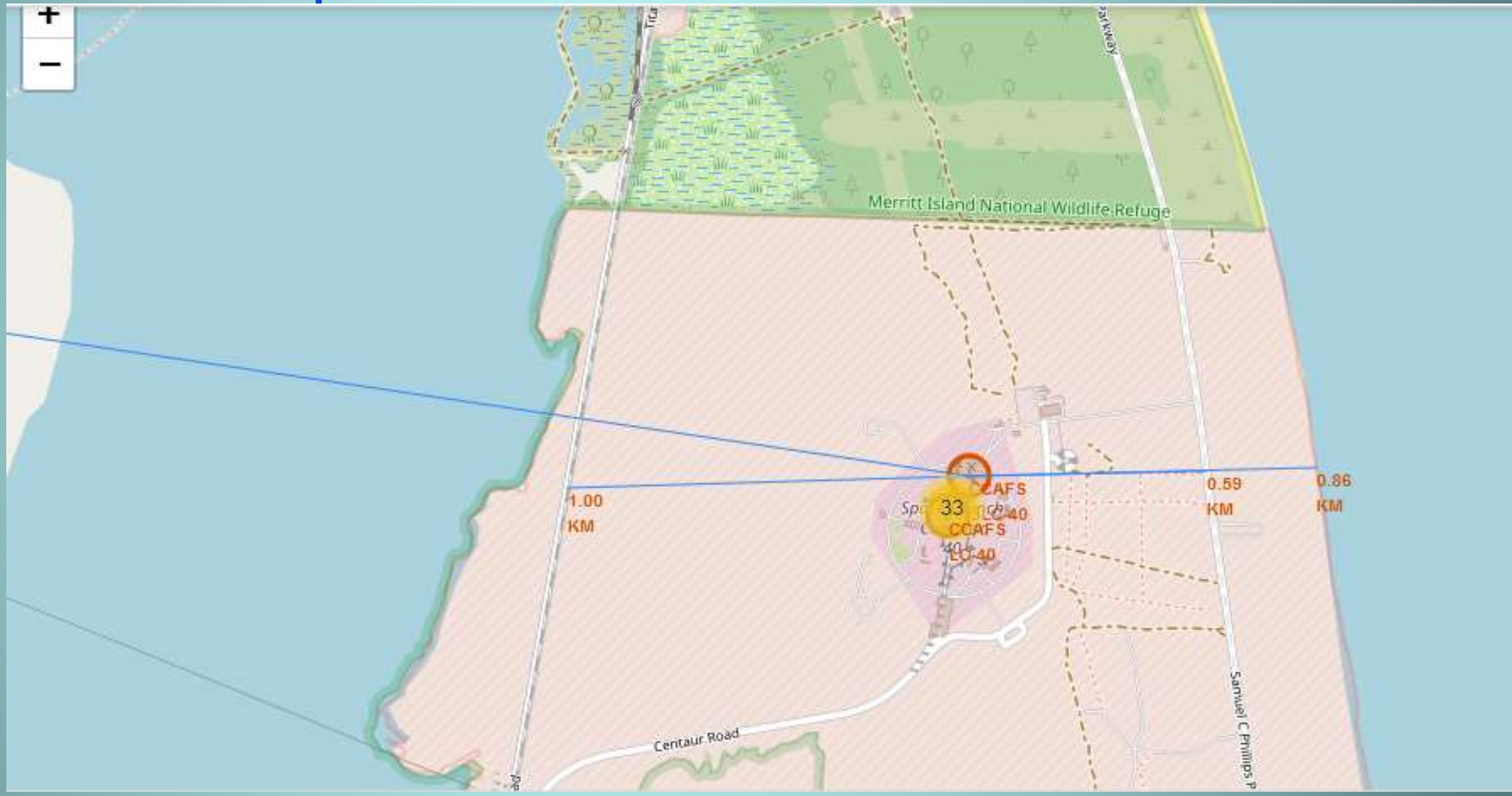
- 4 Launch sites
- All launch sites are in proximity to the Equator line, because this is convenient for the launch due to greater angular momentum there, from earth's rotation.
- all launch sites are in close proximity to the coast, because of reduced risk in comparison to towns, transportation of large components by sea, plenty of water if explode on site etc

# Launch Outcome Map

- CCAF S SLC-40 Launch site
- Red: failure
- Green: success
- This way it is to visually identify sites with many failures like this one

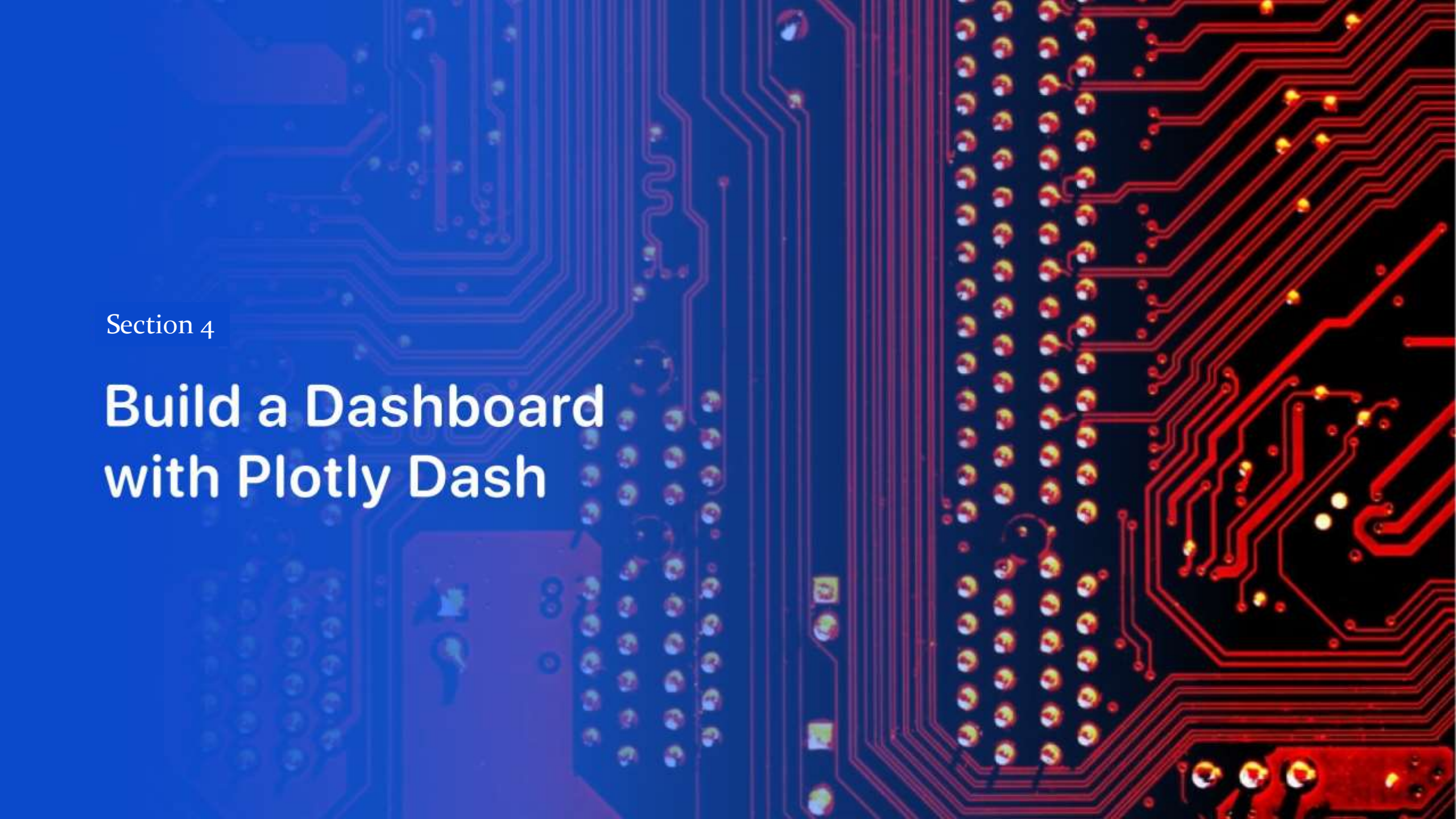
# Launch site Map and Proximities

- Distances to closest coastline, highway and railway, cities
- Close proximity to railways and highways provides easier transportation of material and personnel
- close proximity to coastline provides reduced risk in comparison to towns, transportation of large components by sea, plenty of water if explode on site etc
- Greater distance from cities for safety reasons in order be far away from highly populated areas



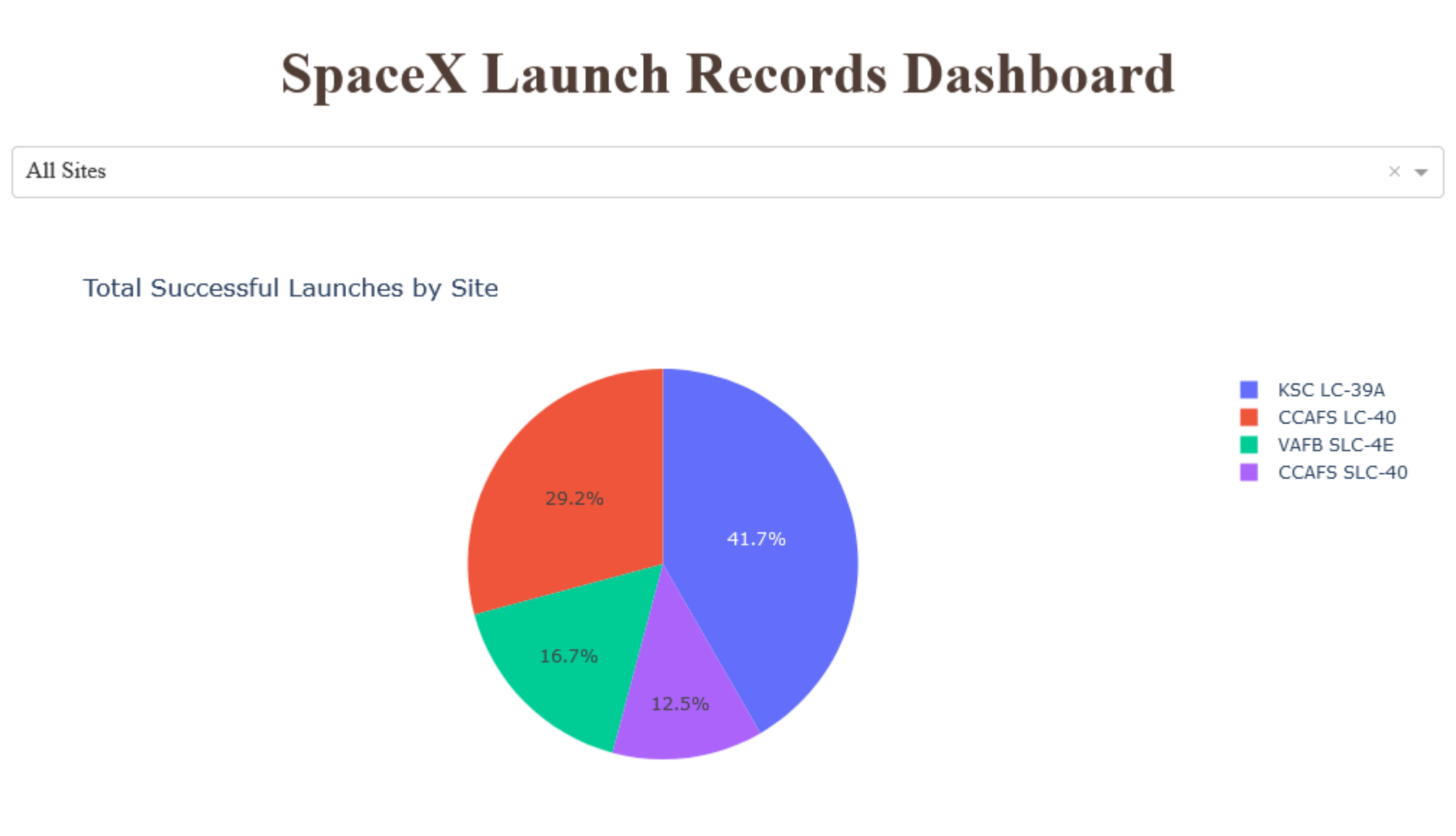
Section 4

# Build a Dashboard with Plotly Dash



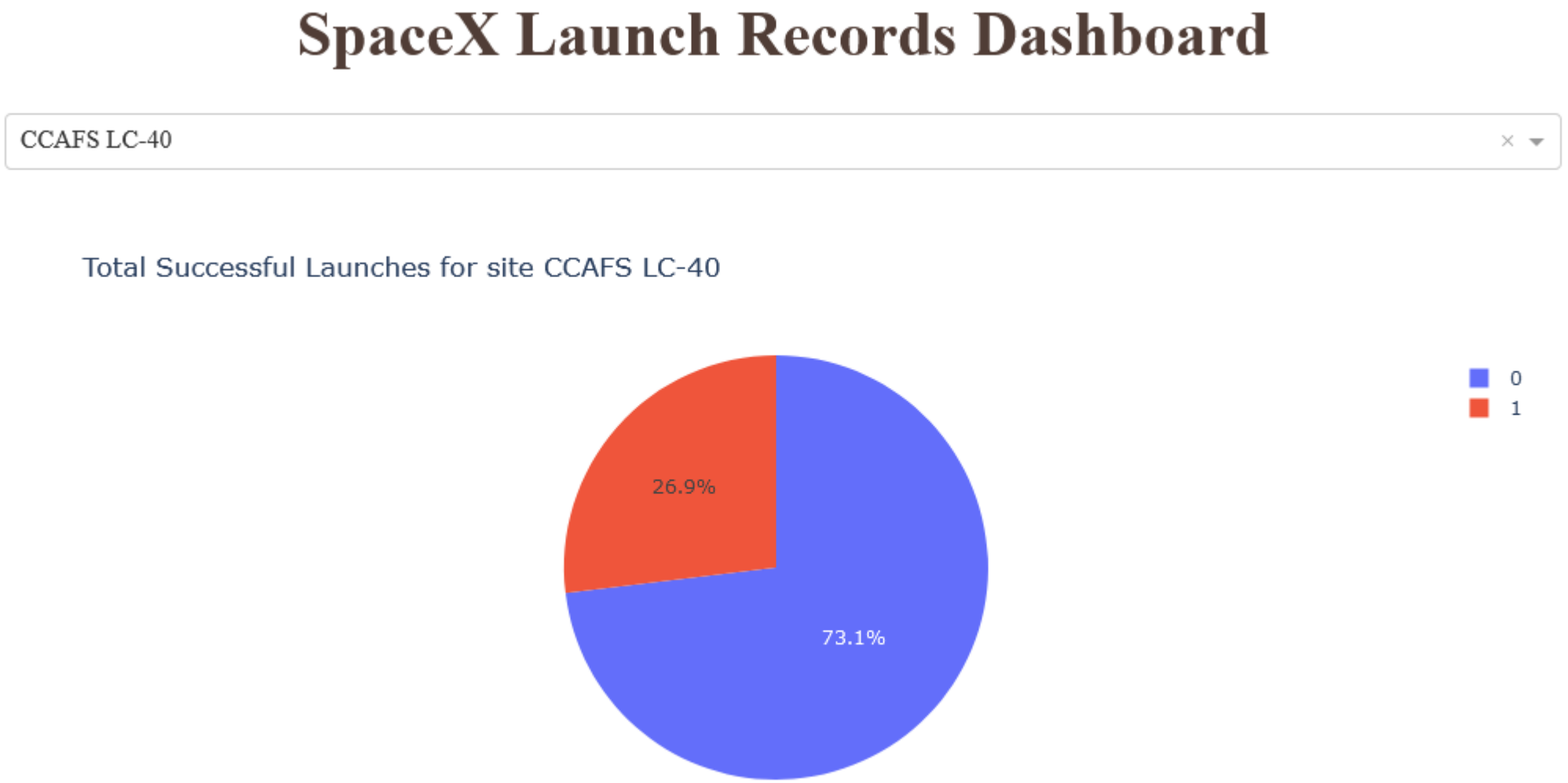
# Launch Success for All Sites

- This is a pie chart of the launch success for all sites for Falcon 9
- KSC LC-39A is the site with the most successful launches

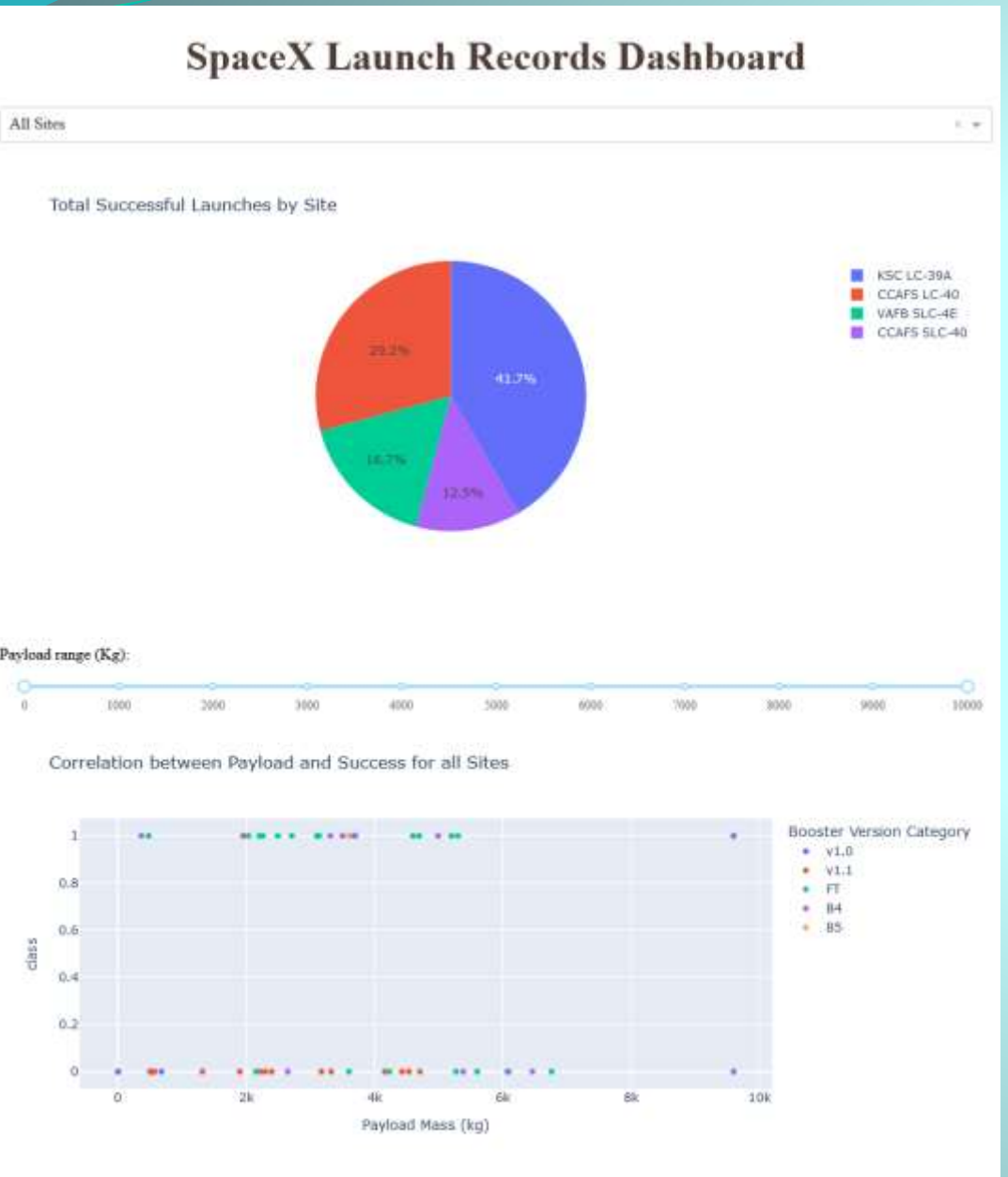


# Launch Success Rate CCAFS-40

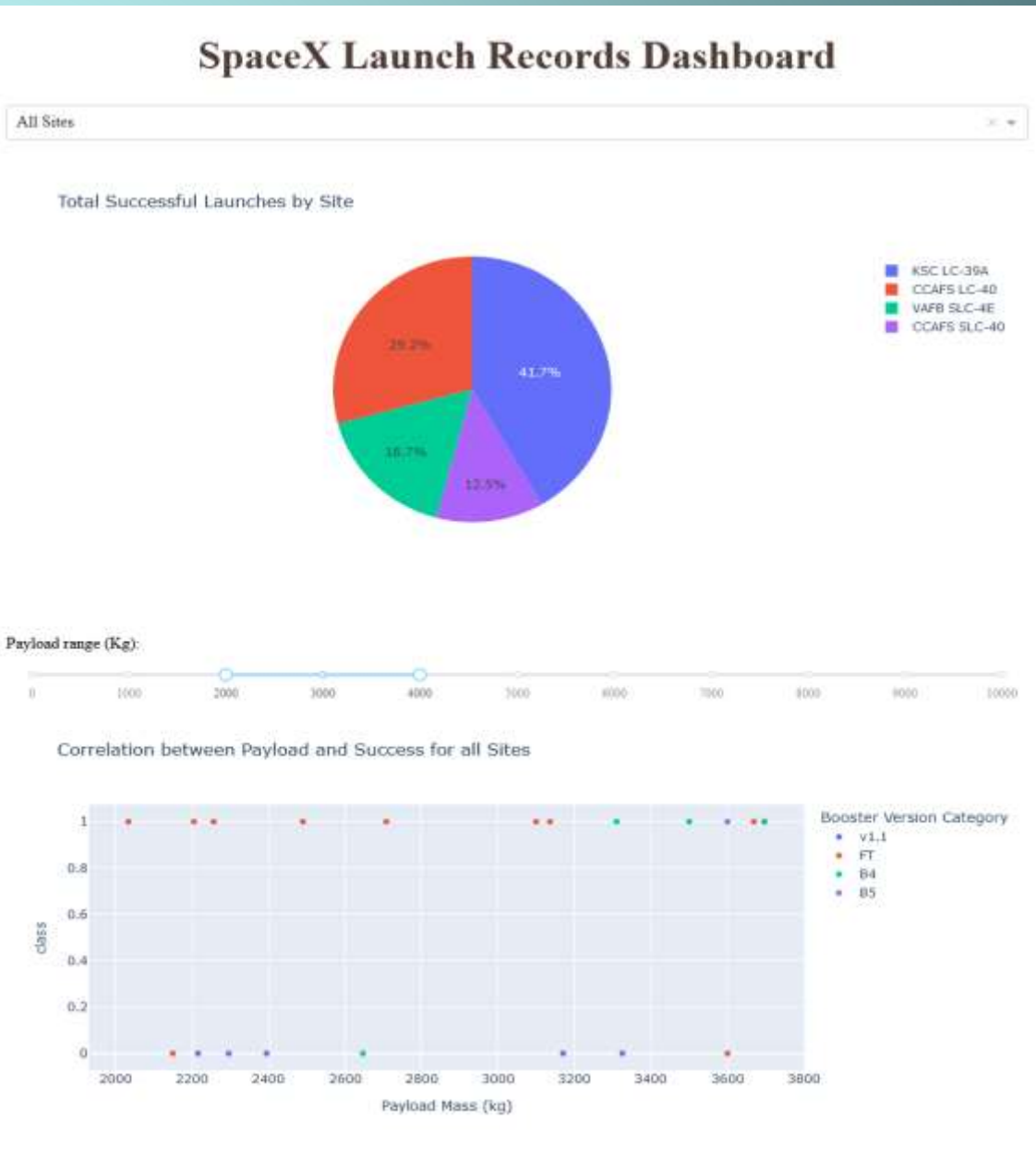
- CCAFS-40 has the highest launch success rate
- 3 out of 4 launches are successful
- GitHub link for Dash app



# Orbit Vs Payload All Sites




- All sites with Range Slider for Payload Range.
- The payload range 2000-4000 has the highest success rate
- The range 6000-7000 has the lowest success rate.
- Booster version FT has the largest success rate.



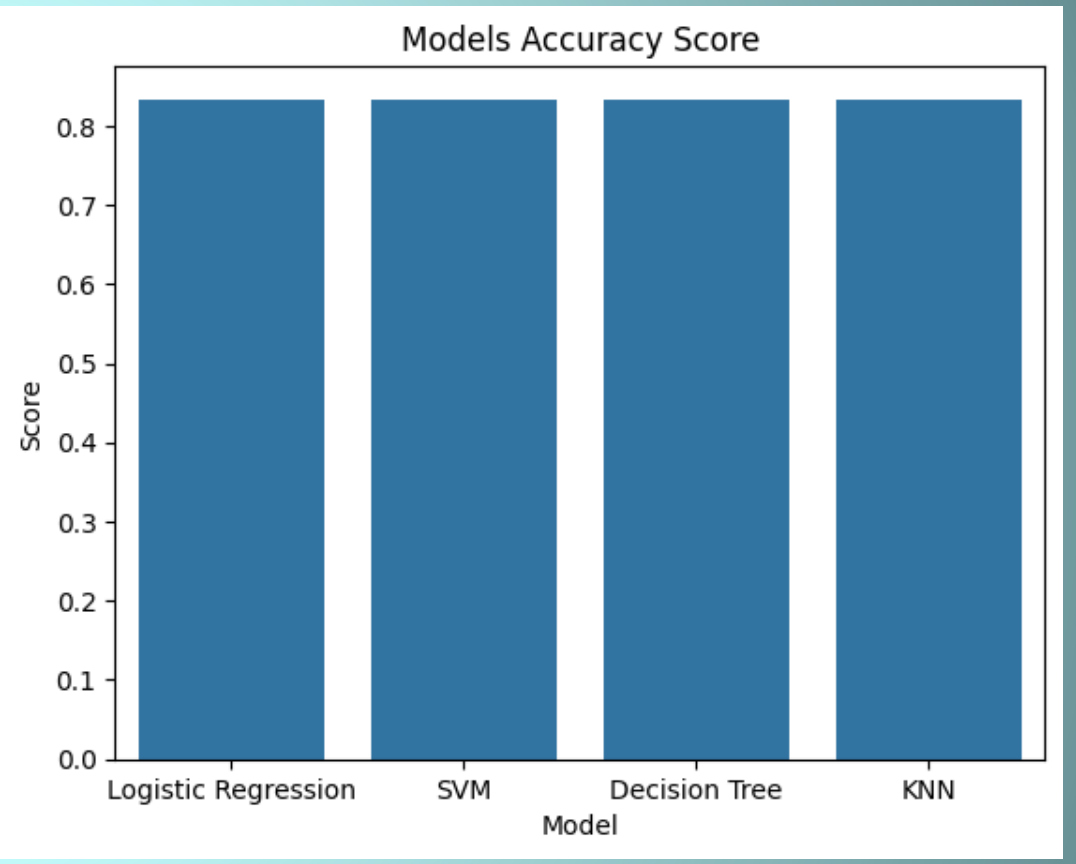
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- All 4 models tested had the same accuracy = 83.3%

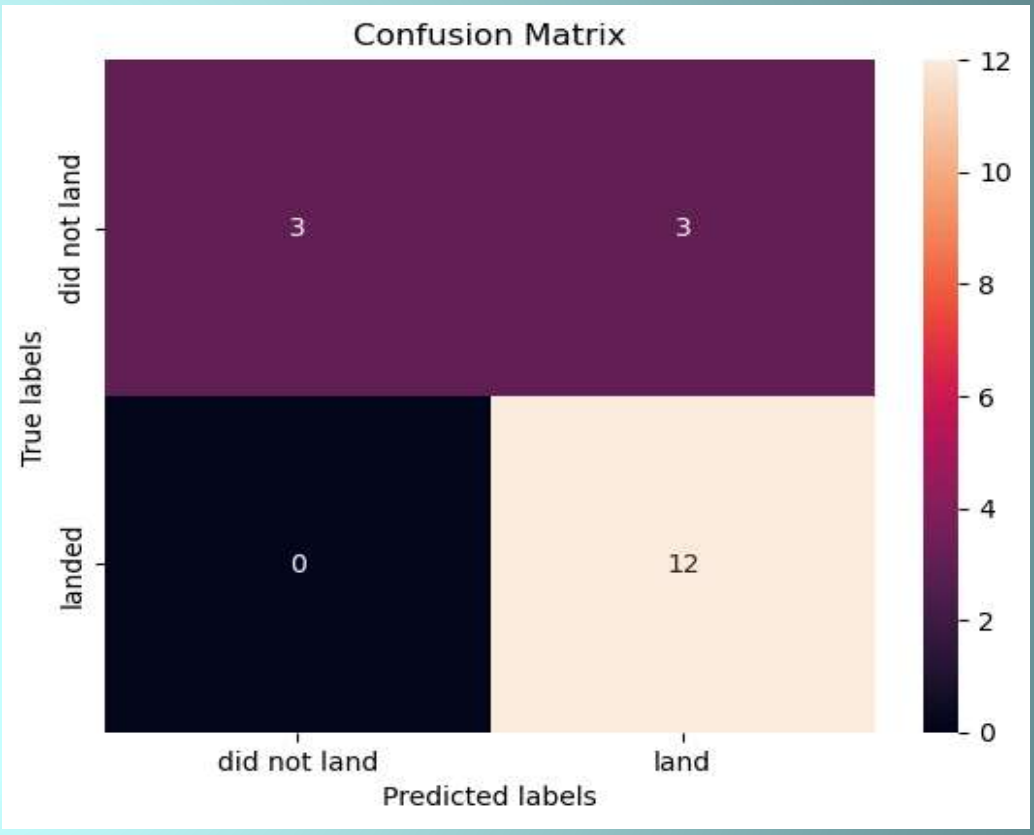
| | Model | Score |
|---|---|---|
| 0 | Logistic Regression | 0.833333 |
| 1 | SVM | 0.833333 |
| 2 | Decision Tree | 0.833333 |
| 3 | KNN | 0.833333 |



# Confusion Matrix

- All models had the same performance and produced the same confusion matrix
- TP : 12
- TN : 3
- FP : 3
- FN : 0
- While Accuracy and precision is high,
- Recall and F1-score are low

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.50 | 0.67 | 6 |
| 1 | 0.80 | 1.00 | 0.89 | 12 |
| accuracy | | | 0.83 | 18 |
| macro avg | 0.90 | 0.75 | 0.78 | 18 |
| weighted avg | 0.87 | 0.83 | 0.81 | 18 |



# Conclusions

## Model Performance:

- All four models achieved an accuracy of 83.3%, indicating good overall classification.
- Precision is high, meaning the model correctly identifies landings most of the time.
- However, recall is low, suggesting some actual landings are misclassified.

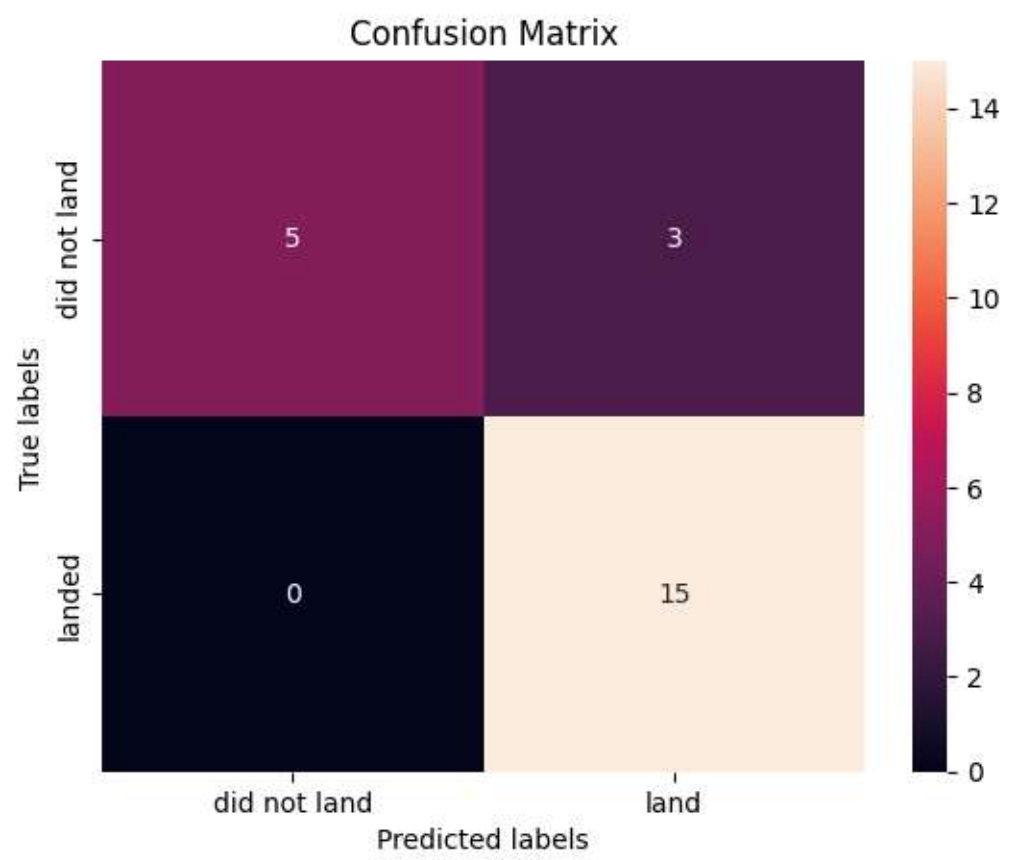
## Key Observations:

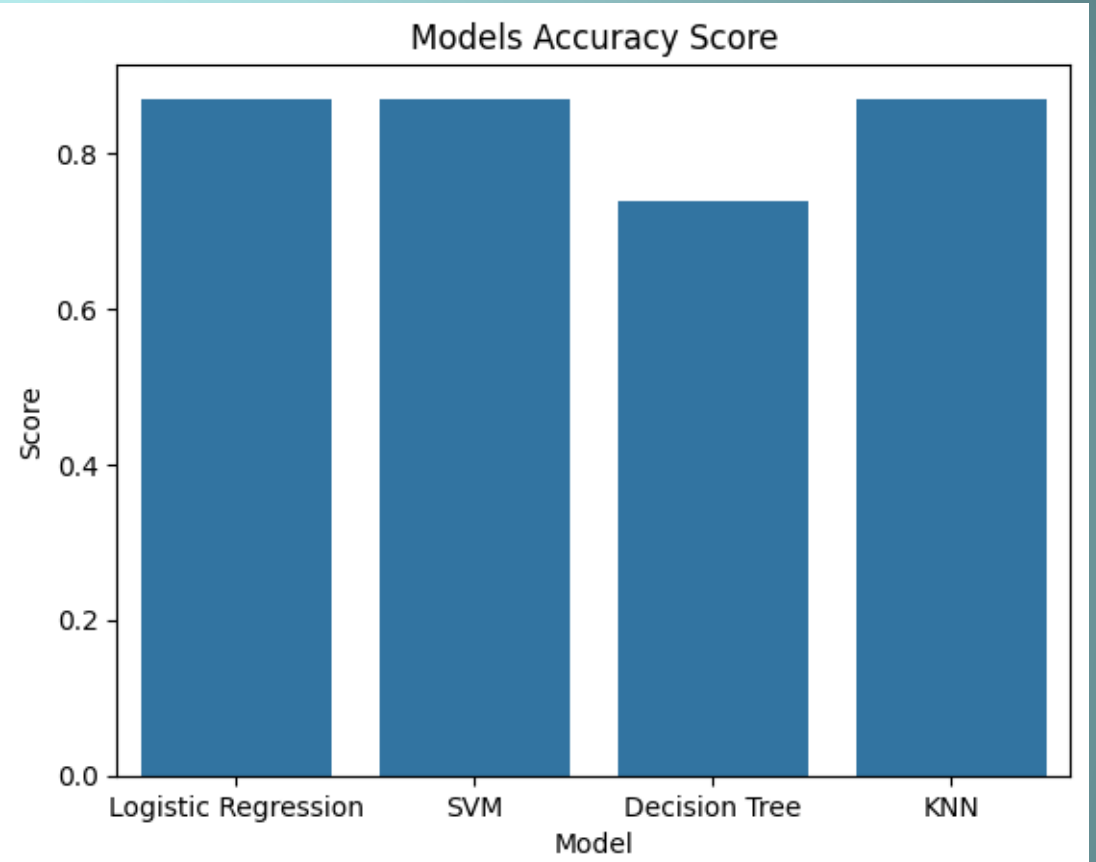
- False Positives (FP = 3): Some “did not land” cases were misclassified as “landed”.
- F1-score is moderate, reflecting a trade-off between precision and recall.
- This misclassification could impact decision-making in real-world applications.

## Future Improvements:

- Optimize model hyperparameters to improve recall while maintaining precision.
- Explore alternative classifiers or use ensemble methods for better performance.
- Feature engineering could enhance model input quality.
- Adjust classification thresholds to reduce false positives for more reliable predictions.

# Bonus Insight



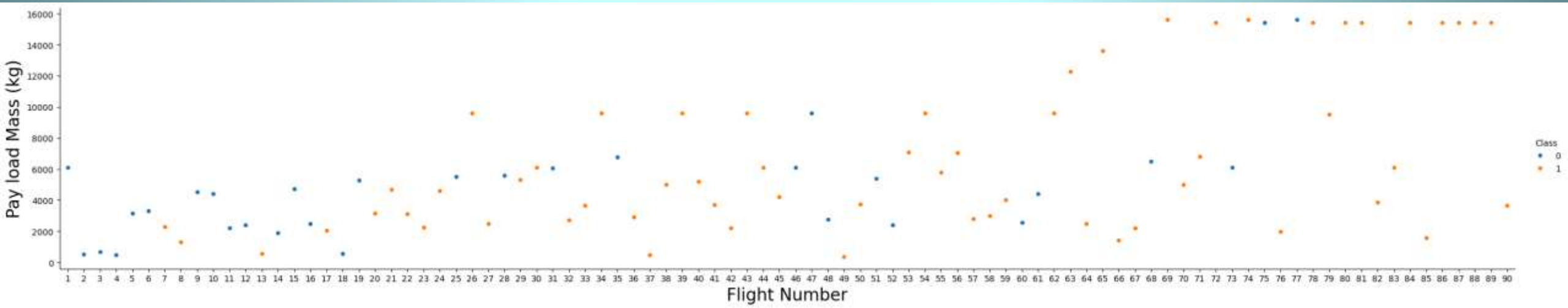

**Model Performance:**

- By optimizing the data split for training and testing (test_size=0.25), we can easily achieve a better accuracy.
- Accuracy ~ 87%, for all models except Decision Tree
- Recall, precision and F1-score are also improved

| | Model | Score |
|---|---|---|
| 0 | Logistic Regression | 0.869565 |
| 1 | SVM | 0.869565 |
| 2 | Decision Tree | 0.739130 |
| 3 | KNN | 0.869565 |

# Appendix

- Payload Vs Flight Number
- For a specific payload the success rate increases with flight number



# Thank you!