IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marland H.
12/09/2024

Outline

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

Executive Summary

Summary of Methodologies

• **Data Collection & Wrangling**: Data was gathered using the SpaceX REST API and Wikipedia web scraping. It was cleaned and prepared by calculating launch frequencies, success rates, and filtering mission outcomes using Python and SQL.

• **Data Analysis & Visualization**: Bar and scatter plots were created for exploratory data analysis, along with dashboards using Plotly Dash and Folium maps for interactive geographical insights

Summary of Results

• Successfully identified key patterns in launch outcomes, including factors that affect success rates, payload distribution, and orbit selection trends.

• The analysis revealed a consistent improvement in mission success rates and highlighted correlations between payload mass, orbit types, and landing outcomes.

Introduction

- SpaceX is at the forefront of revolutionizing space travel with its innovative technologies and reusable rockets. Since its inception, SpaceX has carried out numerous launches, with varying degrees of success, setting the stage for deeper analysis into the factors that contribute to mission outcomes. As the company strives to improve operational efficiency and reliability, understanding the patterns and factors influencing successful launches has become a critical focus area.
-
- This project is centered on analyzing SpaceX's historical launch data to uncover key insights into launch outcomes, payload capacities, and the performance of reusable rocket boosters. By leveraging data-driven techniques, the project seeks to address the following questions:
- 1. What is the distribution of launch successes across different launch sites?
- 2. How do payload mass and booster versions correlate with mission success?
- 3. Can machine learning models accurately predict the success of future launches based on key parameters?
-
- The goal of this analysis is to provide actionable insights that can inform SpaceX's decision-making processes, support operational improvements, and contribute to the broader understanding of factors influencing rocket launches in the aerospace industry.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

Data was collected by is using SpaceX Rest API by making HTTP request to access the API"S endpoints, parsing the data, and storing it for analysis.

- Extracted key attributes from the raw JSON data, including launch site, payload mass, mission outcome, and booster version.

Perform data wrangling:

- Cleaned and formatted the data into a structured tabular format for analysis.
- Used Python libraries like Pandas to handle missing values, normalize data, and ensure consistency.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

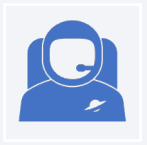
Perform predictive analysis using classification models:

- classification models to predict launch outcomes based on features like payload mass and booster version.
- Utilized logistic regression to analyze binary outcomes (success vs. failure) and evaluated model performance using metrics like accuracy and confusion matrices.

Data Collection Process and flowchart

1. **Wikipedia (HTML Table)**:

• You scraped launch data from a Wikipedia page listing Falcon 9 and Falcon Heavy launches using Python's BeautifulSoup library.

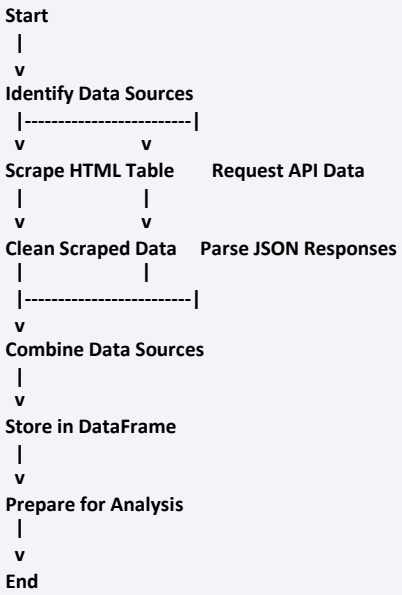


• The data extracted included details such as flight number, date, time, payload mass, launch site, booster version, orbit, customer, and launch outcome.

2. **SpaceX REST API**:

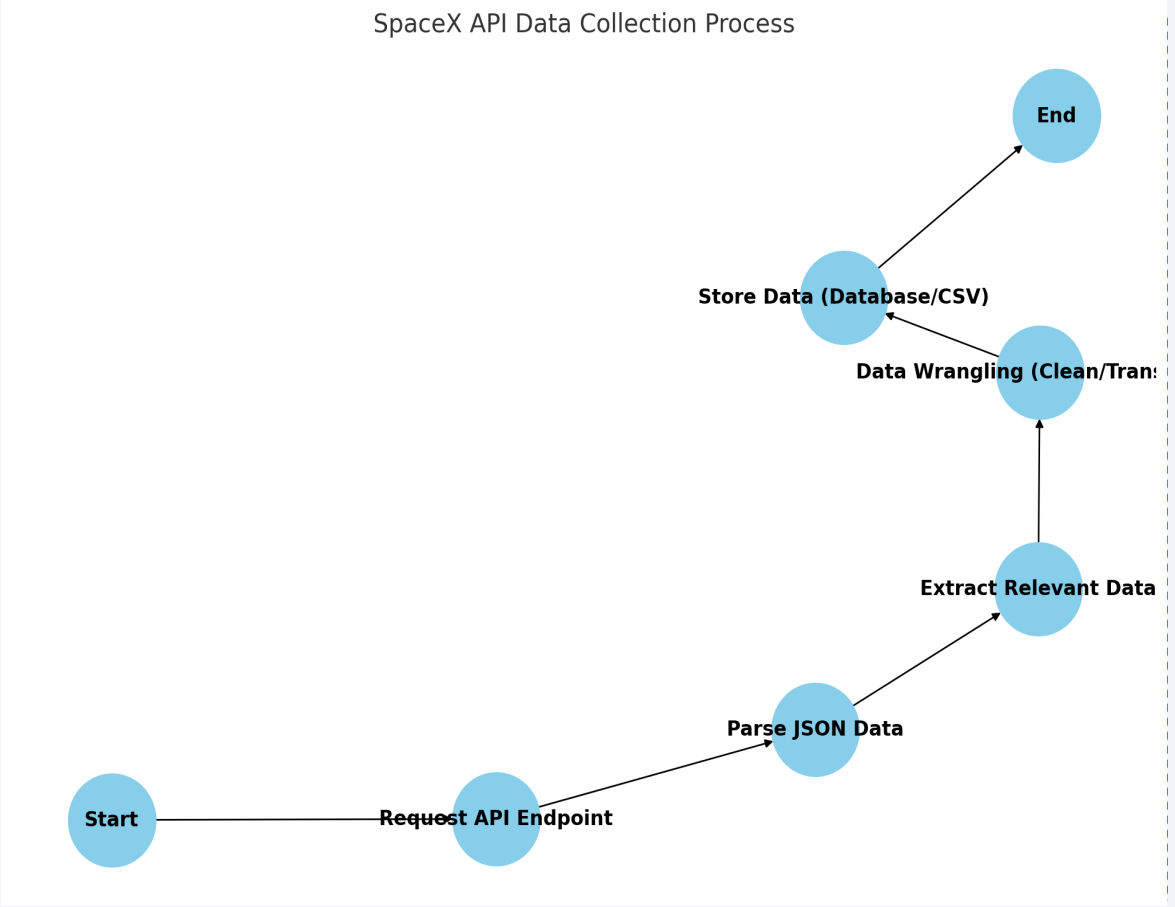
• You used the SpaceX API to retrieve structured data related to the company's launches.

• This source provided additional or supplementary information, such as booster version details, mission outcomes, and payload information



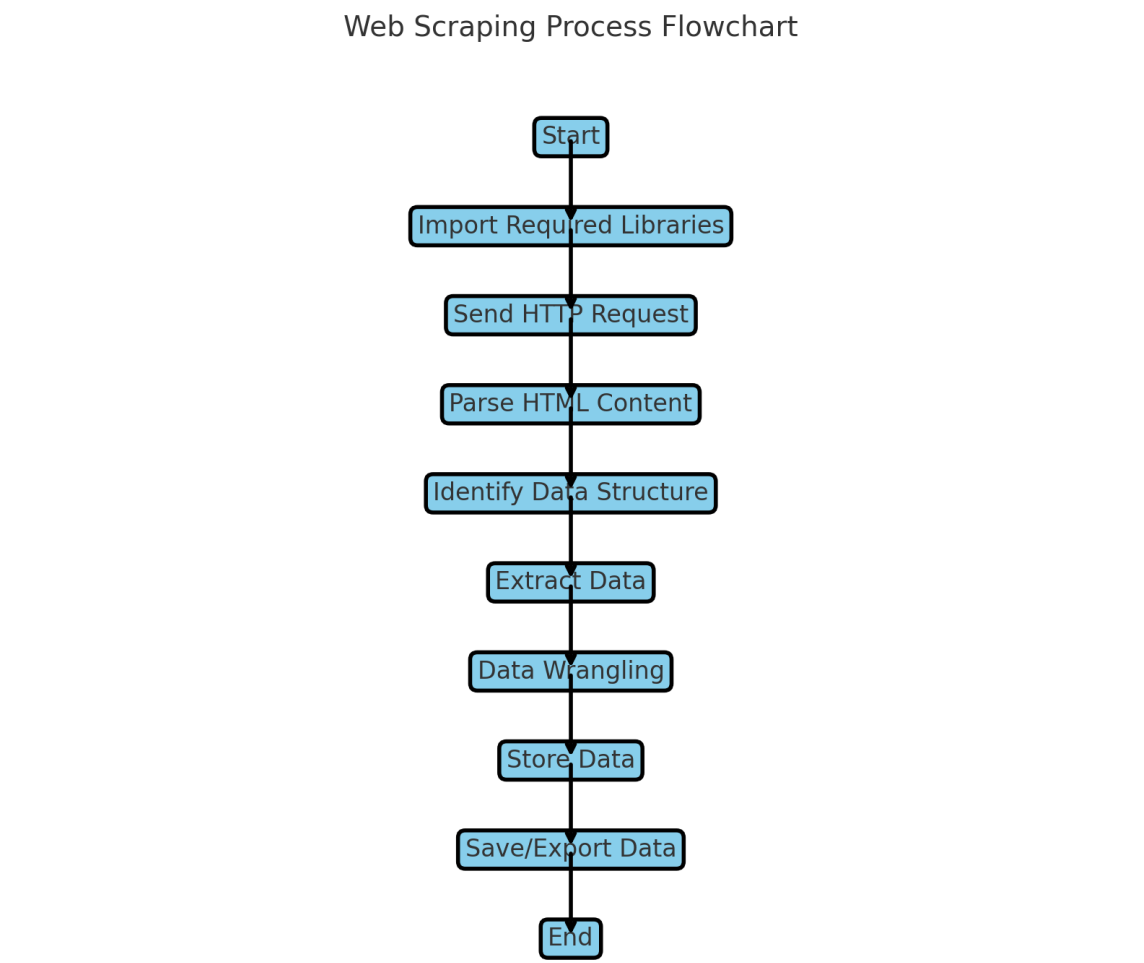
Data Collection – SpaceX API

https://github.com/Marland-Hamilton/super-robot/blob/main/jupyter-labs-spacex-data-collection-api%20(1).ipynb



Data Collection - Scraping

https://github.com/Marland-Hamilton/super-robot/blob/main/jupyter-labs-webscraping%20(1).ipynb



Data Wrangling Flowchart

- START
- ↓
- Import Required Packages
- - pandas, numpy, requests, matplotlib, etc.
- ↓
- Retrieve Data from SpaceX API
- - Parse JSON data into a Pandas DataFrame
- ↓
- Calculate Number of Launches
- - Use `value_counts()` on the `Launch Site` column
- ↓
- Calculate Number of Occurrences
- - Use `value_counts()` on `Orbit` and `Mission Outcome` columns
- ↓
- Create a Set of Outcomes
- - Extract unique mission outcomes using `set()`
- ↓
- Use For Loop for Landing Outcome
- - Iterate through mission outcomes to identify landing results
- ↓
- Set Outcomes for Failed Second Stage
- - Filter rows where the second stage was not successful
- ↓
- Create Landing Outcome Column
- - Assign "Success" or "Fail" to a new column based on conditions
- ↓
- Calculate Success Rate
- - Divide successful landings by total landings and store in a new column
- ↓
- Export Processed Data
- - Save to CSV or database for further analysis
- ↓
- END
- https://github.com/Marland-Hamilton/super-robot/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Exploratory data analysis (EDA) was conducted to uncover patterns and trends in the SpaceX launch data using various visualizations. Bar charts were utilized to compare the frequency of launches across different years, mission outcomes, and orbit types, providing a clear overview of categorical distributions. Scatter plots highlighted relationships between payload mass and success rates, offering insights into payload limits and their impact on mission outcomes. And lastly a line chart to show the positive uptrend of success over the years.These visualizations were chosen for their ability to effectively present categorical and relational data, simplifying the interpretation of key findings.

https://github.com/Marland-Hamilton/super-robot/blob/main/edadataviz.ipynb

EDA with SQL

Query to Retrieve All Launches:	• Selected all columns from the table containing SpaceX launch data to analyze the dataset comprehensively.	• **Filter by Launch Site**:	• Used WHERE clause to filter data for a specific launch site to focus on site-specific analyses.	• **Count Total Launches by Site**:
• Used COUNT() with GROUP BY on the Launch_Site column to calculate the total number of launches per site.	• **Calculate Success Rate**:	• Used CASE statements to categorize outcomes as "Success" or "Failure" and then calculated the ratio of successful launches to total launches.	• **Most Frequently Used Orbit**:	• Used GROUP BY Orbit with COUNT() to determine the most frequently used orbit for launches.
• **Filter by Date Range**:	• Used WHERE clause with conditions on the Date column to retrieve data for specific time periods.	• **Payload Analysis**:	• Queried MAX(), MIN(), and AVG() on the Payload_Mass column to calculate the maximum, minimum, and average payload mass for all launches.	• **Join Queries for Customer Insights**:
• Performed JOIN operations between launch data and customer data tables to analyze customer-specific launch records.	• **Landing Outcome Analysis**:	• Queried the Landing_Outcome column to count the number of successful and failed landings.	• **Distinct Mission Outcomes**:	• Used DISTINCT on the Mission_Outcome column to identify unique mission outcomes
		Let me know if you need more details on any specific query!		

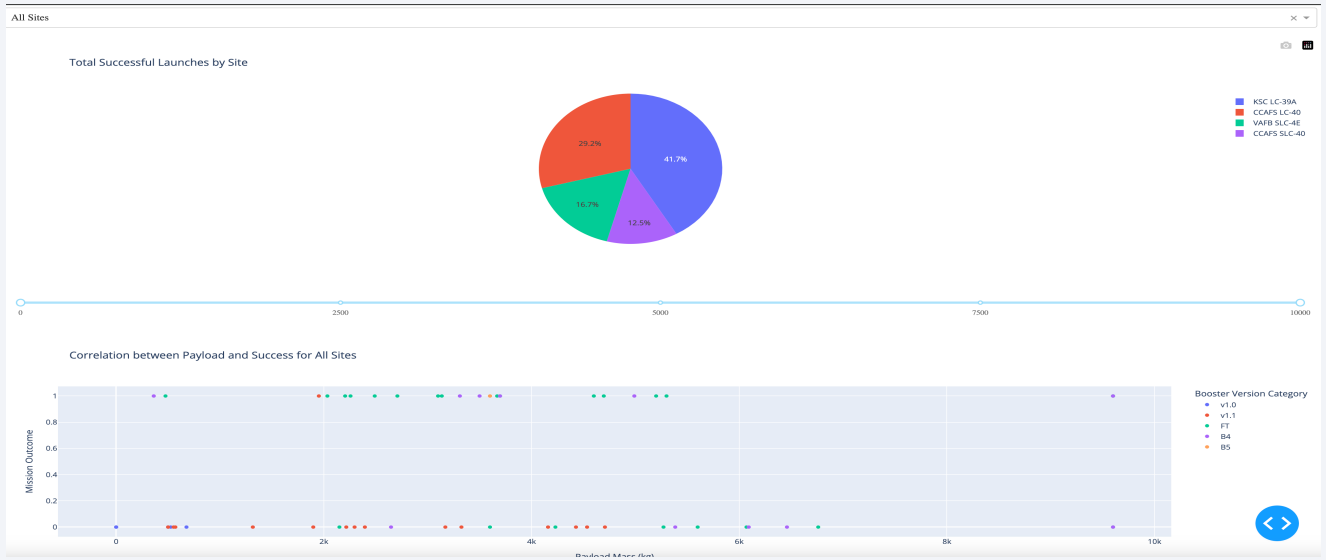
https://github.com/Marland-Hamilton/super-robot/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- the Folium map, I utilized markers to represent specific launch sites, including their coordinates and success status, to clearly identify and distinguish their locations. Circular overlays were added to highlight proximity zones around the launch sites, helping to analyze areas within a specified radius for safety and logistical purposes. Polylines were used to calculate and display distances between launch sites and key landmarks such as the nearest railroad, city, main highway, and coastline to evaluate accessibility and assess potential risks or constraints. These map features were chosen to visually convey spatial relationships and geographical distances, enabling better decision-making and understanding of the launch site's strategic positioning.

https://github.com/Marland-Hamilton/super-robot/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash



To build this Plotly Dashboard, I used a combination of plots, graphs, and interactions.
For the plots and graphs, I used a pie chart to display the “total successful launches by site’, which each segment contains a percentage of success of each launch or if a site is selected, gives the percentage between success and failed launches by the site. Slider(middle) which allows you filter and adjust the payload mas value. Scatter Plot which shows the correlation between payload and success sites. And for my interaction, I use dynamic filtering to explore subsets based on the payload mass, and categorical analysis using different colors to categorize different booster versions.

These combination give clear insight, detailed exploration, user engagement, and multifaceted analysis to make the dashboard both informative and user- friendly.

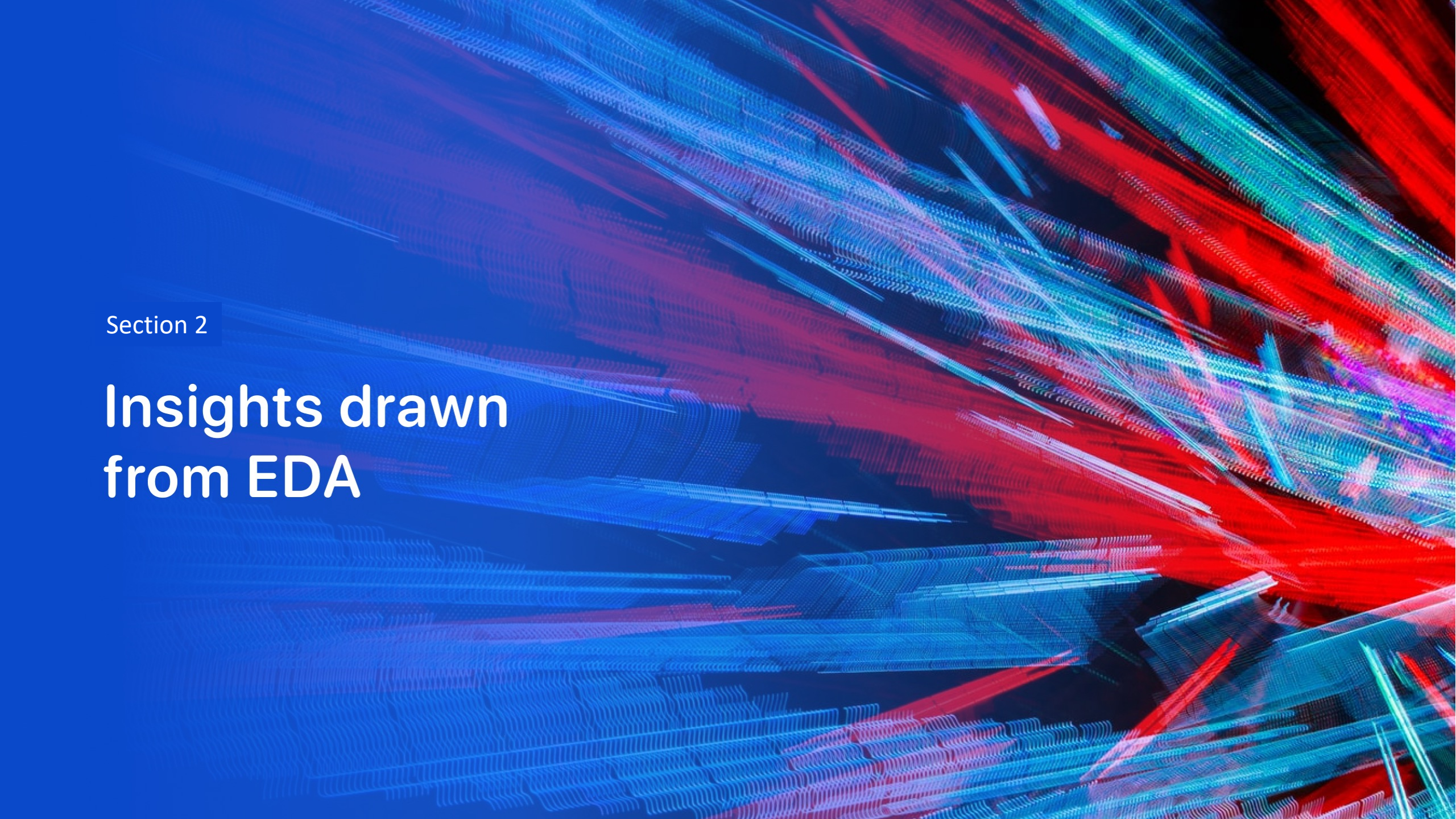
https://github.com/Marland-Hamilton/super-robot/blob/main/python3.11%20spacex_dash_app.py

Predictive Analysis (Classification)

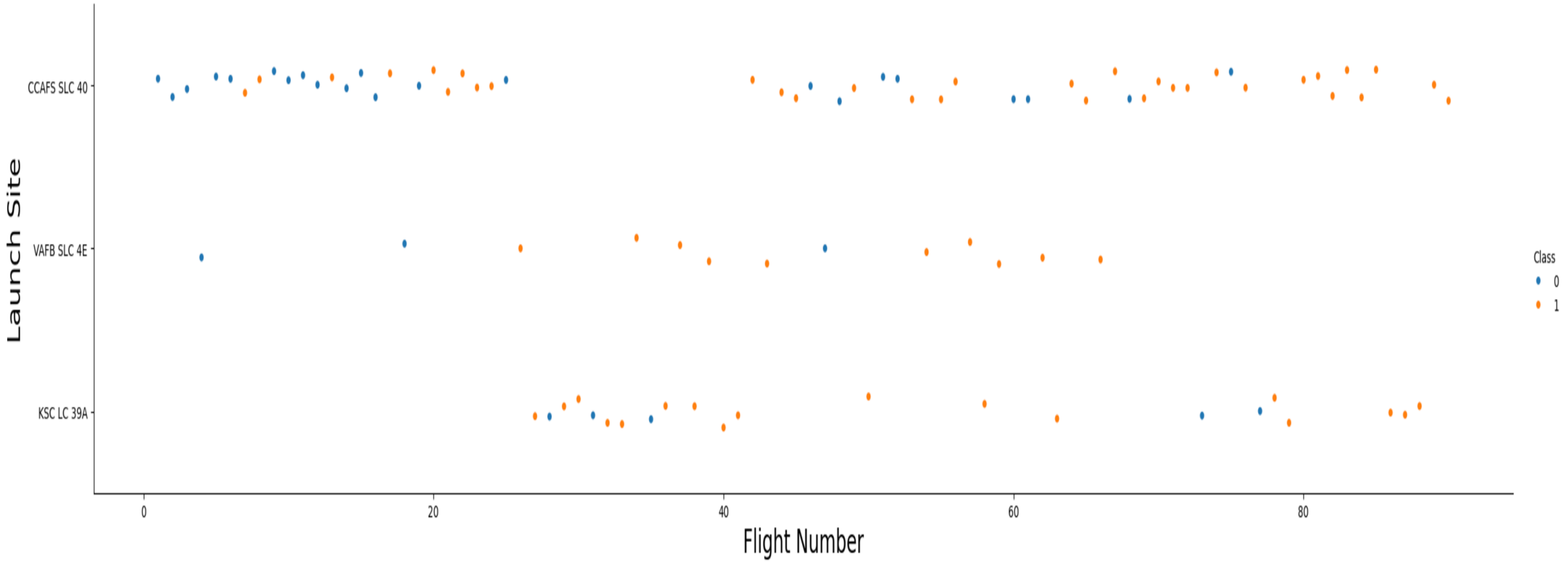
- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

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

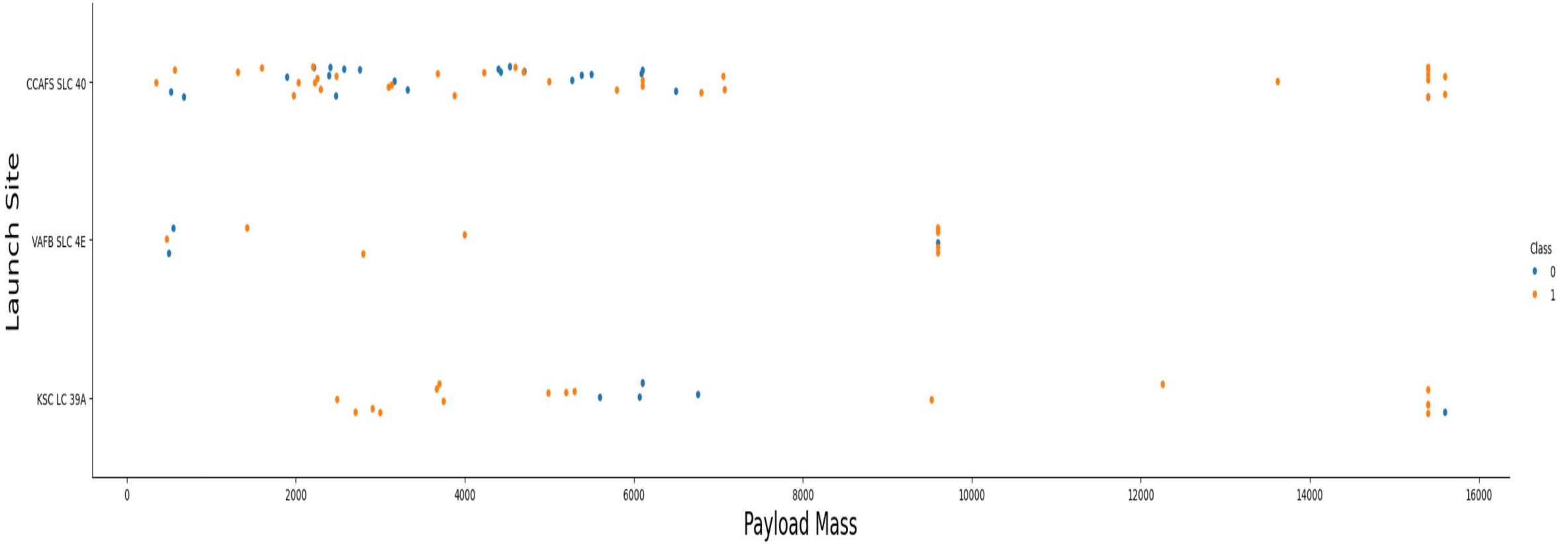

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



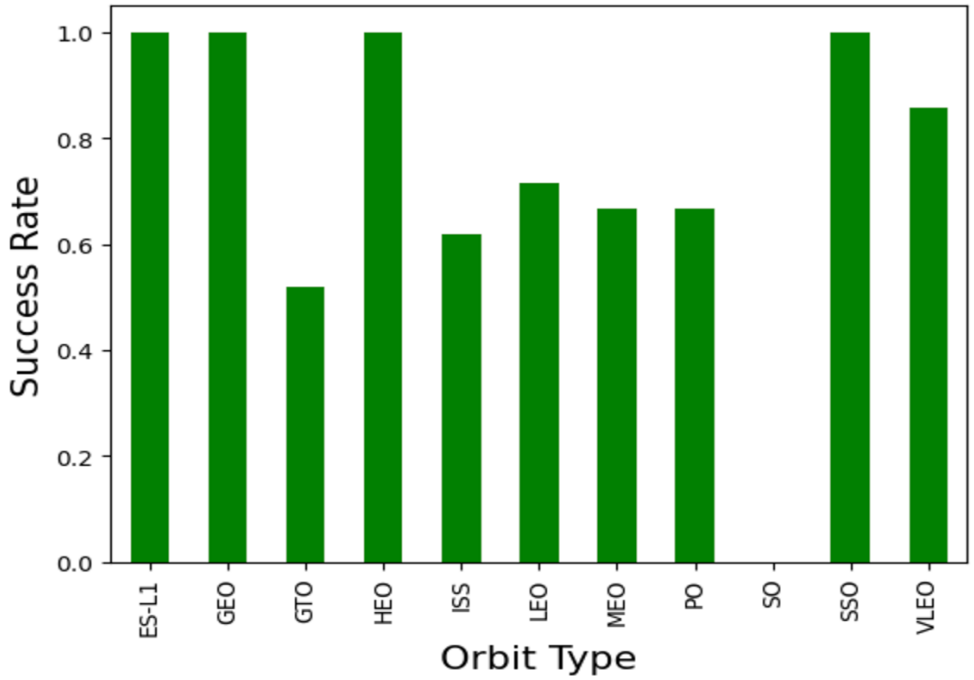
Based on the scatter plot for each station: Launch site CCAFS 40 had a good mixture of success and failures but, as the number of flights increased, the success rate improved. As for launch site VAFB SLC 4E, even though this site have fewer data points it does have a higher success rate than failures. Launch site KSC LC 39A had a higher propertion of successful flights number that came later than the other flights.

Payload vs. Launch Site

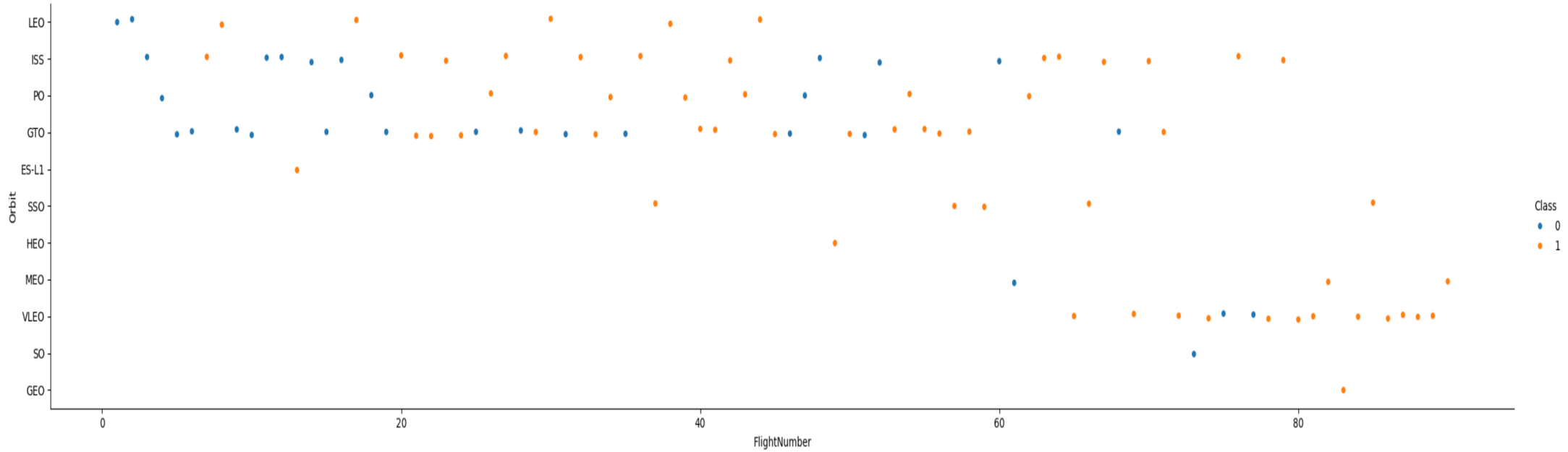


Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavypayload mass(greater than 10000). CCAFS-SLC-40 lauch site heavy payload mass estimate bewteen 100 to around 7500, there a mixture of failure and success flights but once the payload mass got greater than 1300, the rockets had a a majorty success rate.And if you observe launch site KSC-LC-39A Payload mass around the estimate 2200 - 5600 were in the success class with minimum unsuccessful flights being around 6000 - 7000.

Success Rate vs. Orbit Type



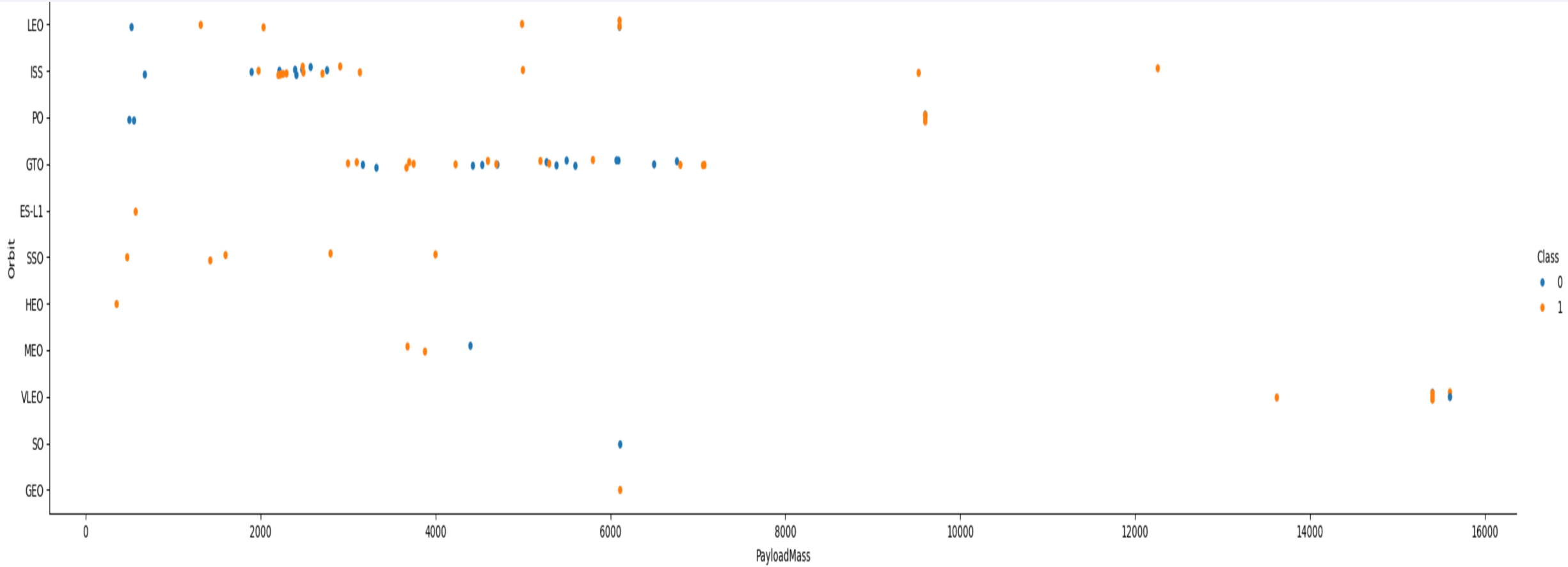
When analyzing the Success Rate VS Orbit Type, the orbit type that had the most success rate includes ES-L1, GEO, HEO, AND SSO.



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success. In summary, the earlier flights tends to have been a failure whereas later flights progress in ore successful flights.

Flight Number vs. Orbit Type

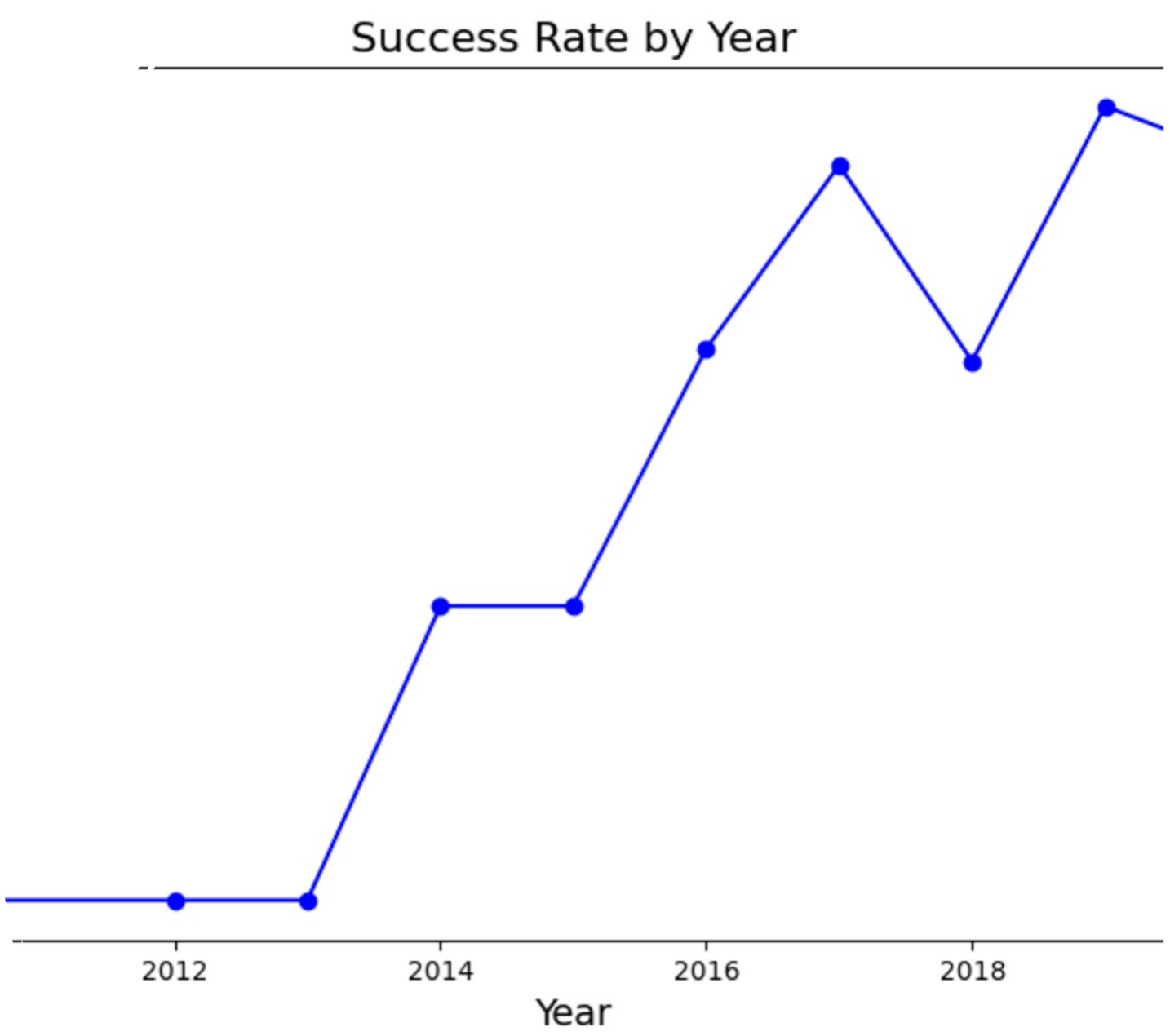
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



Since 2013, the Success rate yearly has been in a upward trend, showing positive progression.

```
[12]: %sql select Distinct "Launch_site"  from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

[12]:

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

All Launch Site Names

When finding all the launch sites, using distinct will filter out the different types used.

Launch Site Names Begin with 'CCA'

```
[13]: %sql select * from SPACEXTBL where "Launch_site" like 'CCA%' limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

[13]:

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

- To find launch sites that begin with ‘CCA’ , using the LIKE operator will search for a specific pattern.

Total Payload Mass

```
[14]: %sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTBL WHERE "Customer" = "NASA (CRS)"
 * sqlite:///my_data1.db
Done.
```

[14]:

SUM("PAYLOAD_MASS__KG_")
45596

To find the Total of NASA's CRS , using the aggregate sum gave me the total payload mass kg

Average Payload Mass by F9 v1.1

```
[15]: %sql select AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" = "F9 v1.1"
 * sqlite:///my_data1.db
Done.
[15]: AVG("PAYLOAD_MASS__KG_")
                        2928.4
```

- By using basic aggregate function “AVG” and what booster version “f9 v1.1” gave me the solution to find the average.

First Successful Ground Landing Date

```
[16]: %sql select *, MIN("DATE") FROM SPACEXTBL where "Landing_Outcome" = "Success"
```

```
 * sqlite:///my_data1.db
Done.
```

[16]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome	MIN("DATE")
2018-07-22	5:50:00	F9 B5B1047.1	CCAFS SLC-40	Telstar 19V	7075	GTO	Telesat	Success	Success	2018-07-22

- This query shows how finding the earliest date that had a successful landing outcome. I used the MIN function from the date column and added a condition In the WHERE clause to show only the “Success” outcomes which in resulted in my findings.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[19]: %sql select "Boosters" , * from SPACEXTBL WHERE "PAYLOAD_MASS__KG_" > '4000' AND "PAYLOAD_MASS__KG_" < '6000' AND "Landing_Outcome" = 'Success (drone ship)'
```

```
 * sqlite:///my_data1.db
Done.
```

[19]:

"Boosters"	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
Boosters	2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
Boosters	2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
Boosters	2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
Boosters	2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

- In this query selects all columns from the SPACEXTBL table where the payload mass is between 4000 and 6000 kg and the landing outcome is "Success (drone ship)". By applying these filters, it retrieves only those launches that meet the specified payload criteria and were successfully landed on a drone ship. The resulting data provides insights into launches with mid-range payloads and successful drone ship landings.

Total Number of Successful and Failure Mission Outcomes

```
[18]: %sql select "Mission_Outcome", count(*) from SPACEXTBL group by "Mission_Outcome"
```

```
 * sqlite:///my_data1.db
Done.
```

[18]:

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

- In this query, I used the COUNT and Group By function to aggregate the data for the conditions where we specifically sort the number of mission outcome by the count each has.

Boosters Carried Maximum Payload

```
[19]: %sql select "Booster_Version" from SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
 * sqlite:///my_data1.db
Done.
```

[19]:

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

- In this this query, you are finding the **Booster Version** that corresponds to the **maximum payload mass** (PAYLOAD_MASS__KG_) in the SPACEXTBL table.

2015 Launch Records

```
%sql select SUBSTR("date", 6, 2) as month, "Landing_Outcome", "Booster_Outcome","Launch_site" from SPACEXTBL WHERE "Landing_Outcome" = 'Failure (drone ship)' and SUBSTR("DATE",0,5) = '2015'
```

```
 * sqlite:///my_data1.db
Done.
```

month	Landing_Outcome	"Booster_Outcome"	Launch_Site
01	Failure (drone ship)	Booster_Outcome	CCAFS LC-40
04	Failure (drone ship)	Booster_Outcome	CCAFS LC-40

- The query retrieves the months and locations of failed drone ships landings in 2015.

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

```
%sql select "Landing_Outcome", count(*) from SPACEXTBL WHERE "DATE" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" order by count(*) desc
```

```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count(*)
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

- The query provides a summary of landing outcomes within the specified time frame showing the most to least landing outcome.

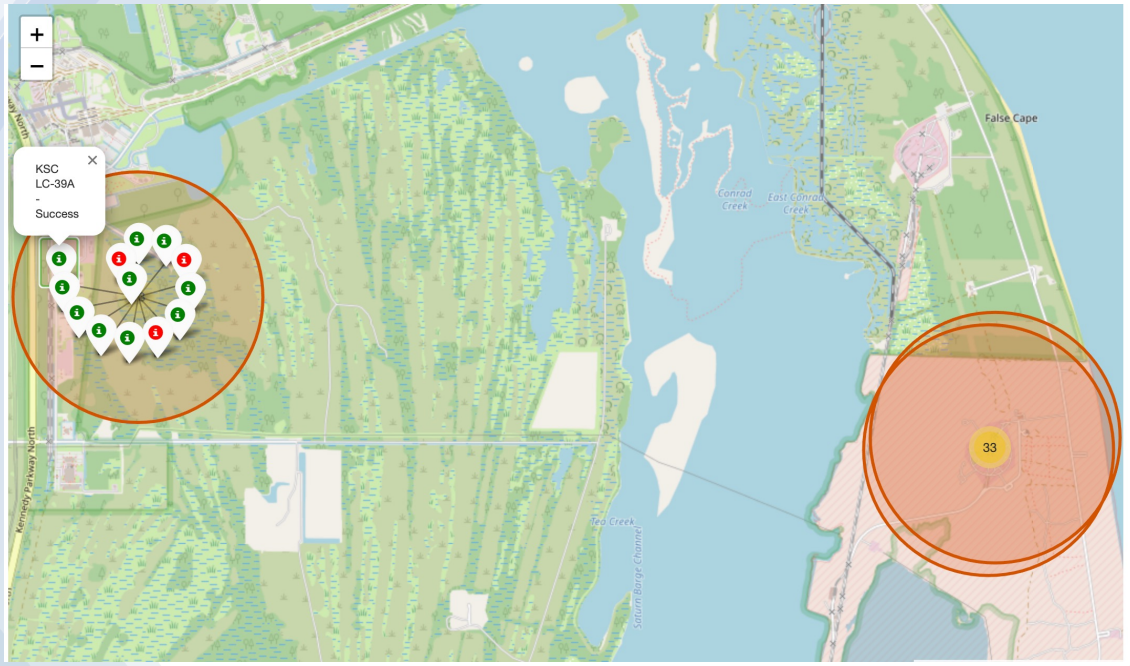

Section 3
Launch Sites
Proximities Analysis

Location of SpaceX Launch sites

- Base on the location of the launch site are located on each coast near a body of water due to a number of reasons such as for safety , ease of recovery, environmental impact.

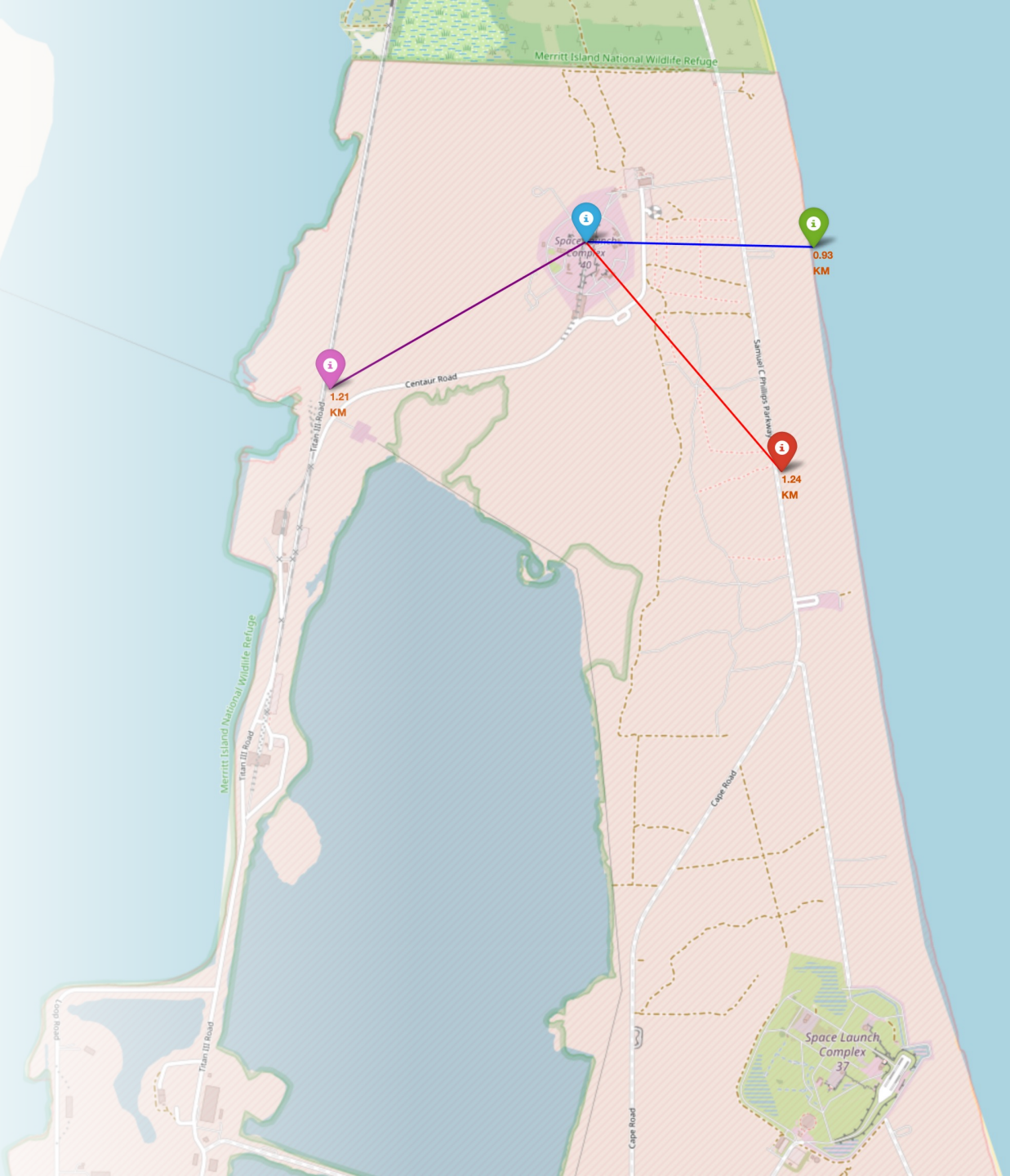
Colored Markers Based on Launch Outcomes

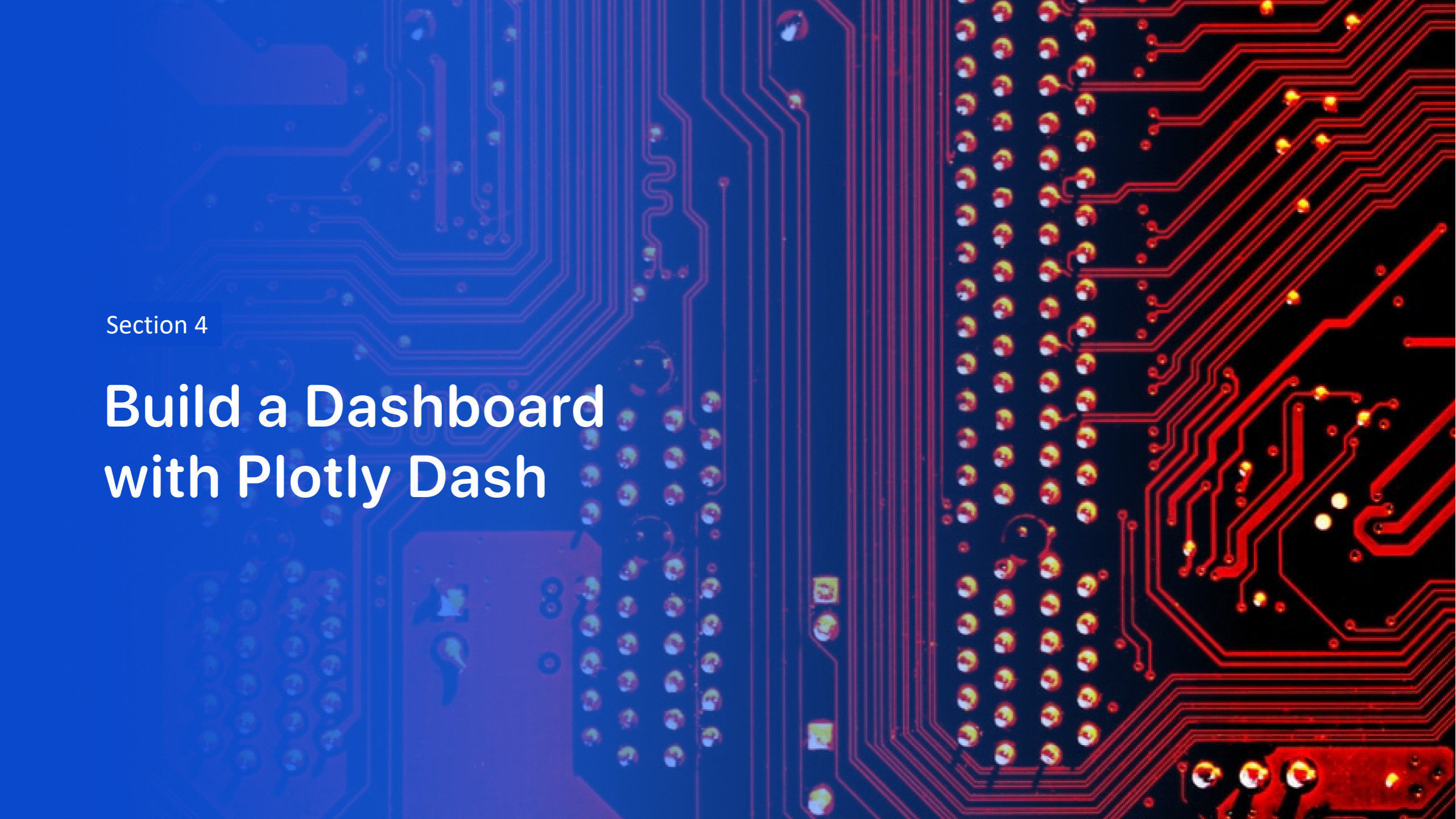
- For the launch results at the SpaceX's sites, green and red color markers are used for categorize results. For a successful launch the green marker is used, and for a failure launch, we used the red. Above at site KSC-LC-39A, you can see that the site site had majority of green markers which equates to the flights being a success.

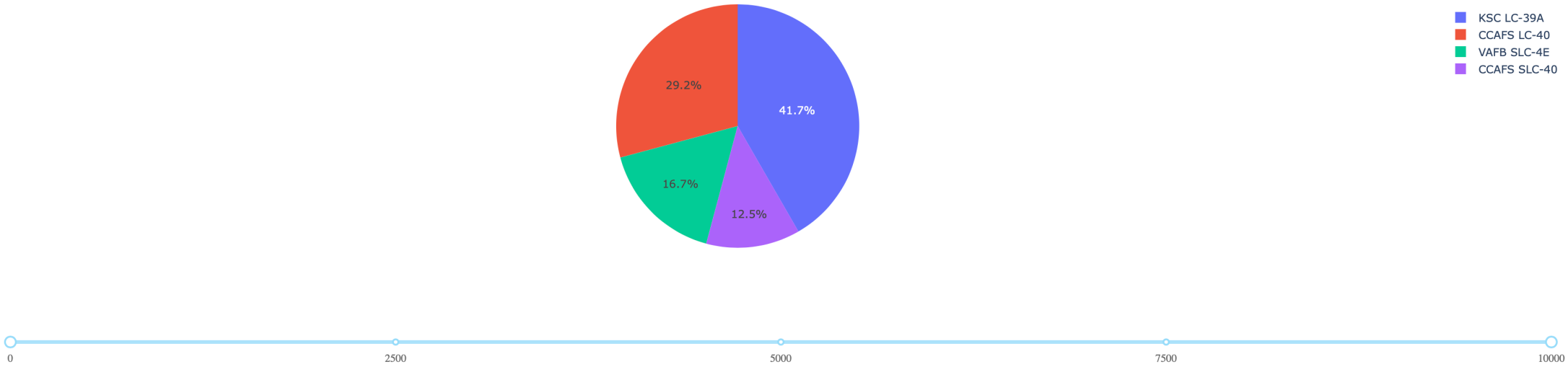


Closest Railway, highway, and coastline to Launch Site

- For the nearest railway, highway, and coastline to the launch site, the railway(purple) is around 1.21km away from the site, the Samuel C Philips Pkwy(red) is around 1.24 km away, and lastly the coastline(green) is approximately .93km away from the launch site.



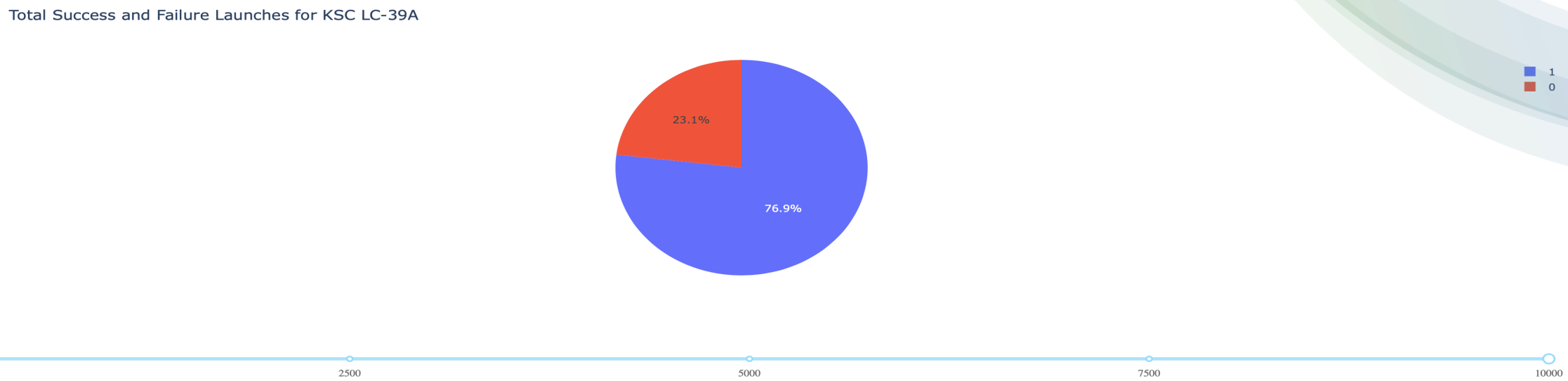

Section 4
Build a Dashboard with Plotly Dash



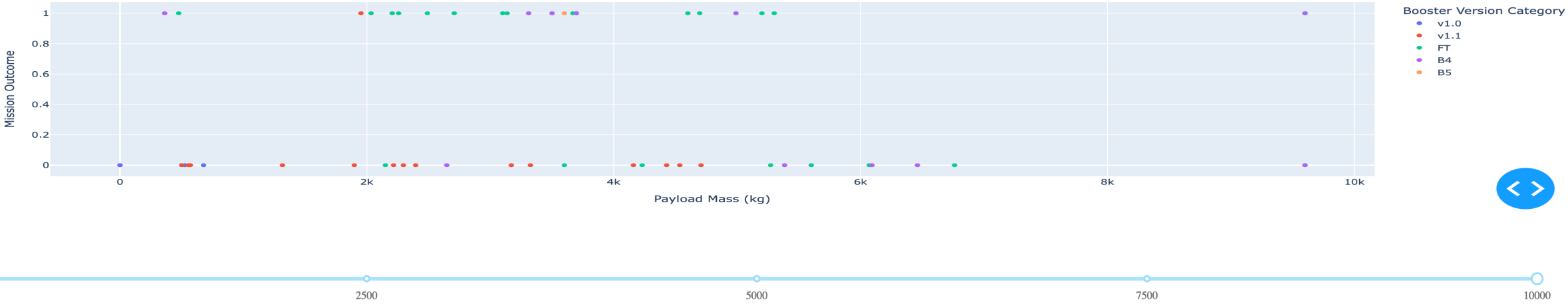
Success percentage by Launch Site

- The highest success launch site out of the site with 41.7% belonged to KSC-LC-39A , whereas launch site CCAAFS-SLC-40 had the least with only a 12.5% success rate.

Highest Launch to failure ratio based on site



- Based on the pie chart and out of all the launch sites, KSC-LC-39A had the best total success to failure ratio with the percentage of 76.9% success and only 23.1% in failures.



Correlation between Payload Mass and Mission Outcome

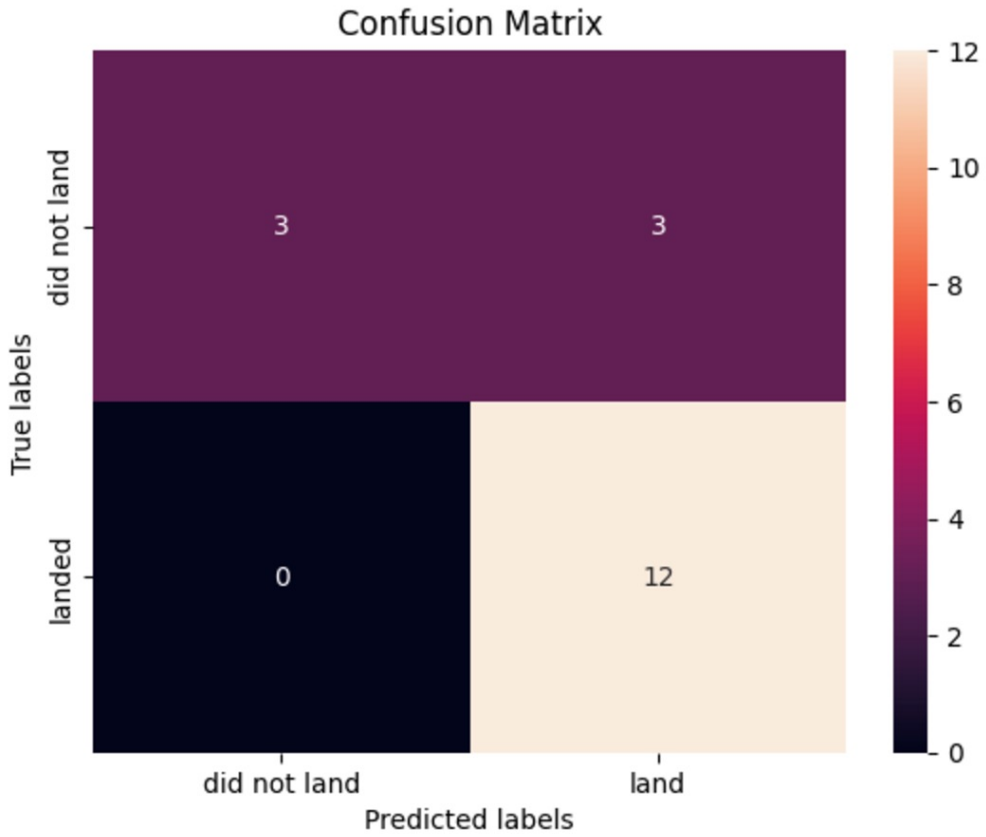
- Based on data, the payload mass the had the largest success rate appears to be in the range of 2000 to 6000 kg.
- Based on the data, you can visually see that the F9 booster has the highest success rates, as their marker is predominantly at the top.

Section 5

Predictive Analysis (Classification)

Confusion Matrix

- • The model performs well overall, achieving high recall (it captures all rockets that landed) and decent precision (some false positives are present).
- • However, it predicts **“landed”** too frequently, leading to **3 false positives.**
- • There are **no false negatives**, which means no rockets that actually landed were missed by the model.



Conclusions

- **Conclusion Points**
- 1. **Launch Frequencies by Orbit Type**
- • The analysis revealed the most frequently targeted orbit types for SpaceX launches, providing insight into the focus areas of their missions. This helps identify SpaceX's strategic priorities and the demand for specific orbit categories.
- 2. **Mission Outcomes**
- • The study highlighted the distribution of mission outcomes, identifying success rates and common reasons for failures. This information is crucial for understanding SpaceX's overall reliability and areas for improvement in mission planning.
- 3. **Payload Mass and Success Rates**
- • The scatter plot analysis demonstrated a potential relationship between payload mass and mission success. Heavier payloads may show a correlation with lower success rates, indicating a challenge for larger missions.
- 4. **Trends Over Time**
- • Temporal analysis using bar charts showed trends in successful launches over time, indicating technological advancements and operational improvements in SpaceX's processes.

- These findings provide actionable insights into SpaceX's operations, mission planning, and success metrics, paving the way for continuous improvement and strategic planning.

Appendix

Python Code Snippets

- **Data Wrangling**: Code used to calculate launch success rate and process landing outcomes.
- • **Visualization**: Scripts to create bar charts, scatter plots, and generate Folium maps.

SQL Queries

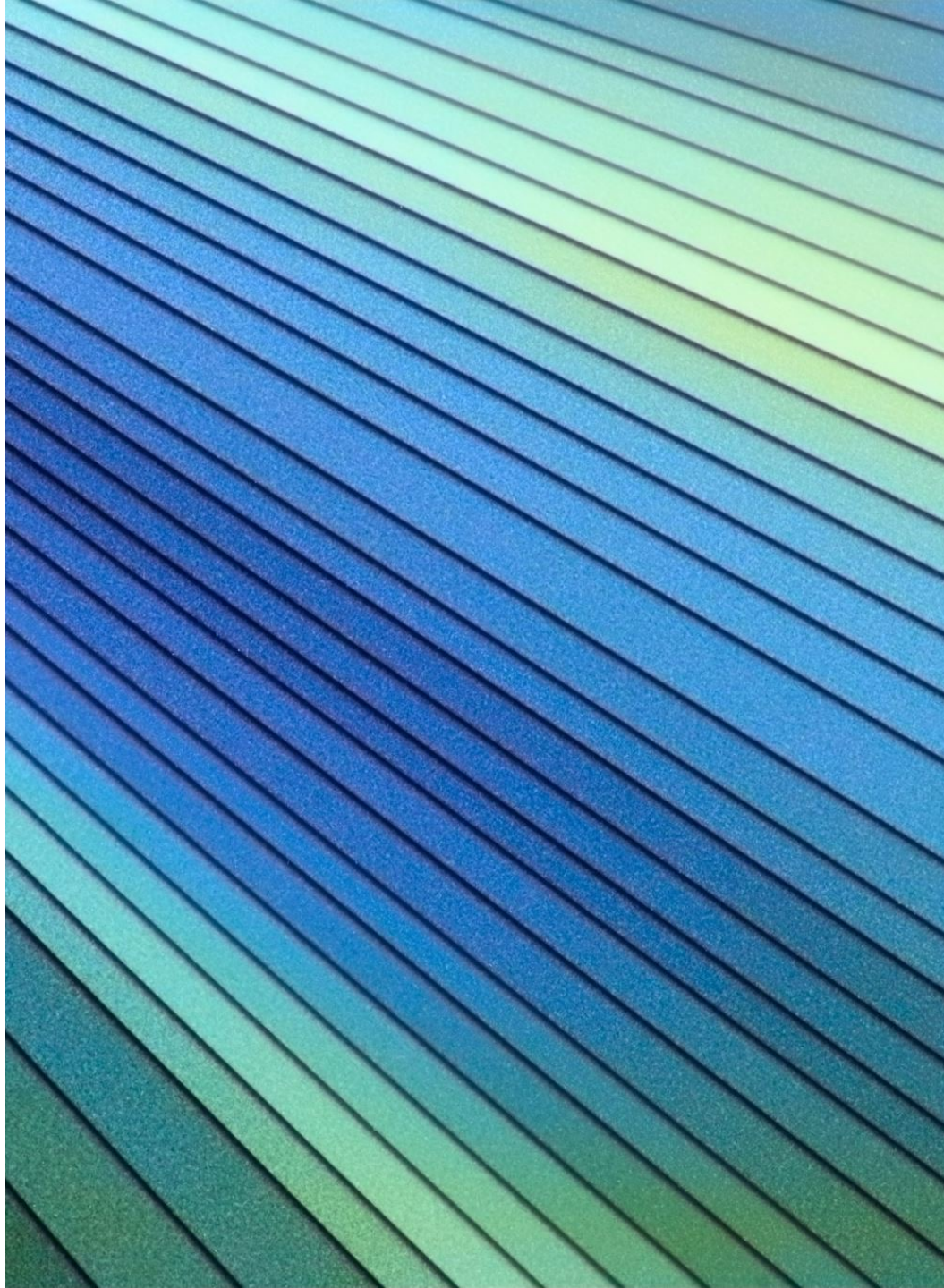
- Query to count total launches and calculate success rate by orbit type.
- Query to extract mission outcomes and categorize landing success.

Data Visualizations

- **Bar Chart**: Showcased mission outcomes to identify success trends.
- **Scatter Plot**: Analyzed relationships between payload mass and success rates.
- **Folium Map**: Visualized launch sites and calculated distances to landmarks like railroads, highways, and coastlines.

Other Assets

- Dataset: Cleaned and processed SpaceX API data used for all analyses.
- Notebook Outputs: Intermediate calculations and visualizations created during exploratory data analysis.



Thank you!