IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rizwan Farooq Khan
14-September-2024

Outline

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

Executive Summary

Summary of Methodologies:

- Data collection and preparation
- Data wrangling and cleaning
- Exploratory Data Analysis (EDA) using data visualization techniques
- EDA using SQL queries
- Development of an interactive map with Folium
- Creation of an interactive dashboard using Plotly Dash
- Predictive analysis through classification models

Summary of Results:

- Findings from exploratory data analysis
- Screenshots showcasing the interactive analytics features
- Results from the predictive analysis

Introduction

Project background and context

SpaceX's success in the space industry is partly due to its cost-effective rocket launches. The Falcon 9 rocket, advertised at $62 million per launch, is significantly cheaper than competitors who charge upwards of $165 million. A major factor contributing to these lower costs is SpaceX's ability to reuse the first stage of the Falcon 9 rocket.

Our goal is to predict the success rate of the Falcon 9 first stage landing.

By accurately predicting whether the first stage will land successfully, we can estimate launch costs more precisely. This information is valuable for any competing company considering bidding against SpaceX for rocket launches.

Problems you want to find answers

What factors determine whether the rocket will land successfully?

How does the relationship between various rocket variables affect the likelihood of a successful landing?

What conditions must SpaceX meet to optimize performance and ensure the highest rocket landing success rate?

Section 1

Methodology

Methodology

Executive Summary:

Data Collection Methodology:

- Data was gathered from the **SpaceX REST API** and **web scraping** from Wikipedia.
- Data wrangling was performed, transforming the data for machine learning by applying **one-hot encoding** to categorical variables and removing irrelevant columns.

Exploratory Data Analysis (EDA):

- EDA was conducted using **visualization tools** and **SQL** to uncover patterns and relationships between variables.
- Various graphs, such as scatter plots and bar charts, were created to illustrate these relationships.

Interactive Visual Analytics:

- **Folium** was used to build interactive maps, and **Plotly Dash** was employed to create an interactive dashboard for enhanced data visualization.

Predictive Analysis:

- Predictive analysis was performed using **classification models** to predict rocket landings.
- The models were built, tuned, and evaluated to optimize performance, applying techniques such as **GridSearchCV** for hyperparameter tuning.

Data Collection

Data Collection Process:

The data collection process involved two primary methods:

SpaceX API:

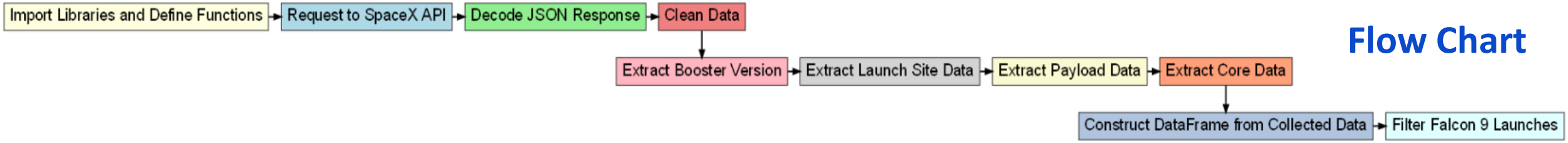
- **GET Request**: Data was collected by sending a GET request to the SpaceX API.
- **Response Parsing**: The response was decoded as JSON using the .json() function.
- **Data Normalization**: The JSON data was converted into a structured format by using the .json_normalize() method, transforming it into a pandas DataFrame for easier analysis.

Web Scraping (Wikipedia):

- **Web Scraping**: Historical Falcon 9 launch records were extracted using Python's BeautifulSoup library.
- **HTML Parsing**: The HTML tables containing launch information were parsed, extracting relevant data.
- **Data Transformation**: The extracted data was converted into a pandas DataFrame, ready for further analysis.

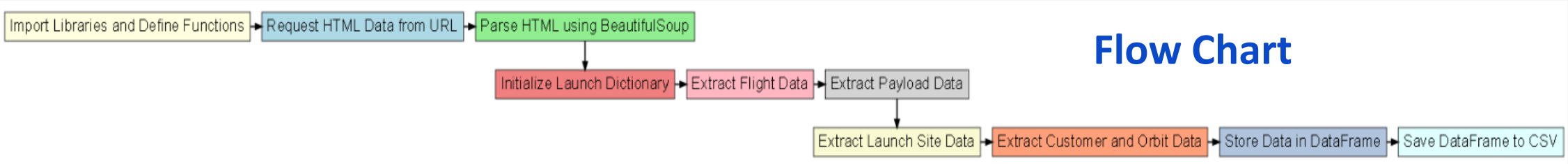
After collecting data from both sources, the datasets were cleaned, missing values were handled, and irrelevant fields were dropped. The final step involved combining both datasets into a unified structure for subsequent machine learning tasks.

Data Collection – SpaceX API



Below is brief explanation of SpaceX APIs Data collections.
Import Libraries and Define Functions: Set up the environment and define functions before making API requests.
Request to SpaceX API: Make the initial GET request to fetch the data.
Decode JSON Response: Decode the API response into JSON format.
Clean Data: Perform basic data cleaning and handle missing values.
Extract Booster Version: Retrieve information about the rocket boosters.
Extract Launch Site Data: Get data about launch sites, including location details.
Extract Payload Data: Extract details about payloads.
Extract Core Data: Gather information about rocket cores.
Construct DataFrame from Collected Data: Combine all the collected data into a Pandas DataFrame.
Filter Falcon 9 Launches: Filter the DataFrame to include only Falcon 9 launches.
Github Link: Below github URL contain notebook with code and results as well.
https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/spacex-data-collection-api.ipynb

Data Collection - Scraping



Key Phrases to Describe the Process:

HTTP Request: Use the requests.get() method to send an HTTP GET request to the static Wikipedia URL.Capture the response and check the status to ensure it's successful.

Parse HTML Content: Use the BeautifulSoup library to parse the HTML content of the Wikipedia page.Set the appropriate parser ('html.parser').

Extracting Data from Tables:Locate tables with the class "wikitable plainrowheaders collapsible".Loop through each table row to extract data using find_all() for table rows (<tr>) and cells (<td>).

Data Cleaning: Handle missing values and inconsistent formatting (e.g., None checks, reference links removal).Split date and time, clean strings, and ensure the extracted data matches the expected format.

Populating the Dictionary: Append values such as flight number, launch site, payload, payload mass, orbit, customer, launch outcome, and booster landing into a pre-defined dictionary.

Data Export: Convert the dictionary into a pandas DataFrame.Save the DataFrame as a CSV file for further analysis.

Github URL: Below github URL contain notebook with code and results as well.

https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/Data%20Collection-webscraping.ipynb

Data Wrangling

Loading the Dataset The SpaceX dataset was loaded from a CSV file using Pandas for data manipulation and analysis.

Exploratory Data Analysis (EDA): The dataset was examined to understand the distribution of launches, payloads, orbits, and outcomes.

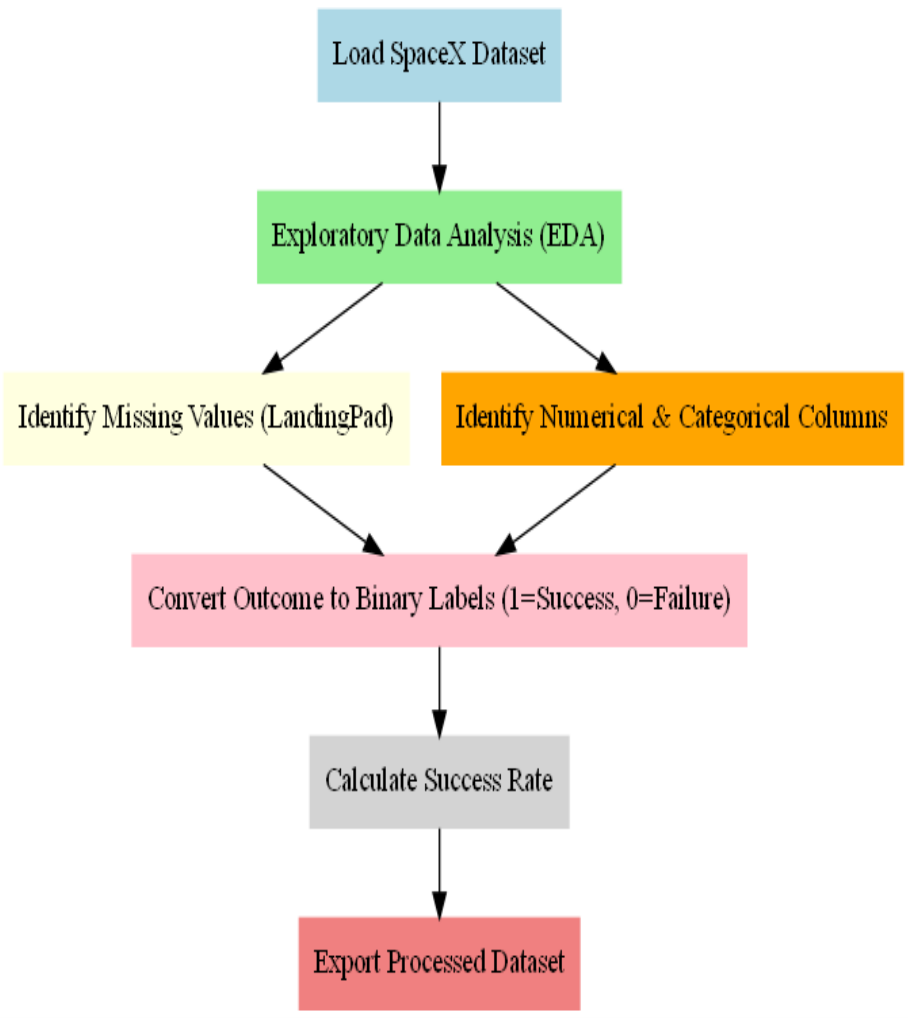
Handling Missing Values: The column LandingPad contained a significant amount of missing values **Identifying Key Attributes Categorical Columns:** Data types were categorized using df.dtypes().5. Determining Launch Success Mission Outcomes.

Creating Training Labels: A new column, landing_class, was created, where the value is 1 if the landing was successful and 0 if unsuccessful.The labels will serve as the target for supervised machine learning models.

Calculating Launch Success Rate: The success rate of the landings was calculated by counting the occurrences of successful outcomes in the dataset.Success rate was determined as 66.67%, which reflects the overall effectiveness of the Falcon 9 launches.

Github URL: Below github URL contain notebook with code and results as well.

https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/spacex-Data%20wrangling.ipynb



EDA with Data Visualization

We plotted scatter plot for FlightNumber vs. PayloadMass and overlay the outcome of the launch. We saw that as the flight number increases, the first stage is more likely to land successfully. The payload mass also appears to be a factor; even with more massive payloads, the first stage often returns successfully.

Plotted scatter plot Visualize the relationship between Flight Number and Launch Site.

Visualize the relationship between Payload Mass and Launch Site. The payload mass also appears to be a factor; even with more massive payloads, the first stage often returns successfully.

Plotted a bar chart to visualize the relationship between success rate of each orbit type.

With scatter plot visualized the relationship between FlightNumber and Orbit type and observed that 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.

Visualized the relationship between Payload Mass and Orbit type it is observed that 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.

Plotted line graph to visualize the launch success yearly trend.Success rate increased from 2013 to 2020.

Github URL: Below github URL contains notebook with code and results as well.

https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/EDA-Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

While performing EDA with sql below EDA tasks performed.

Displayed the names of the unique launch sites in the space mission.

Displayed 5 records where launch sites begin with the string 'CCA'.

Displayed the total payload mass carried by boosters launched by NASA (CRS).

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

Listed the date when the first successful landing outcome on a ground pad was achieved.

Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

Listed the total number of successful and failure mission outcomes.

Listed the names of the booster versions which have carried the maximum payload mass.

Listed the records showing the month names, failure landing outcomes in drone ship, booster versions, and launch site for the months in the year 2015.

Ranked the count of landing outcomes between the dates 2010-06-04 and 2017-03-20, in descending order.

Github Link: Below github URL contain notebook with code and results as well.

https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/EDA%20with%20SQL-SQL%20Notebook.ipynb

Build an Interactive Map with Folium

We marked all launch sites on the Folium map.

We added map objects like markers, circles, and lines to indicate the success or failure of launches at each site.

Launch outcomes (failure or success) were assigned as 0 and 1, respectively.By examining color-labeled marker clusters, we identified launch sites with relatively high success rates.

We calculated the distances between a launch site and its nearby features.

We answered questions such as:

- Are launch sites located near railways, highways, or coastlines?
- Do launch sites maintain a certain distance from cities?

Github URL: Below github URL contain notebook with code and results as well.

https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone Project/blob/main/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

Created an interactive dashboard using Plotly Dash.

Displayed pie charts to show the total number of launches per site.

Plotted a scatter graph to visualize the relationship between mission outcomes and payload mass (kg) across different booster versions.

Github URL: Below github URL contain notebook with code and results as well.

https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/dashboard_app.py

Predictive Analysis (Classification)

Loaded the data using NumPy and pandas, followed by data transformation, and splitting into training and testing sets.

Developed various machine learning models and fine-tuned hyperparameters using GridSearchCV.

Utilized accuracy as the performance metric, enhancing the model through feature engineering and algorithm tuning.

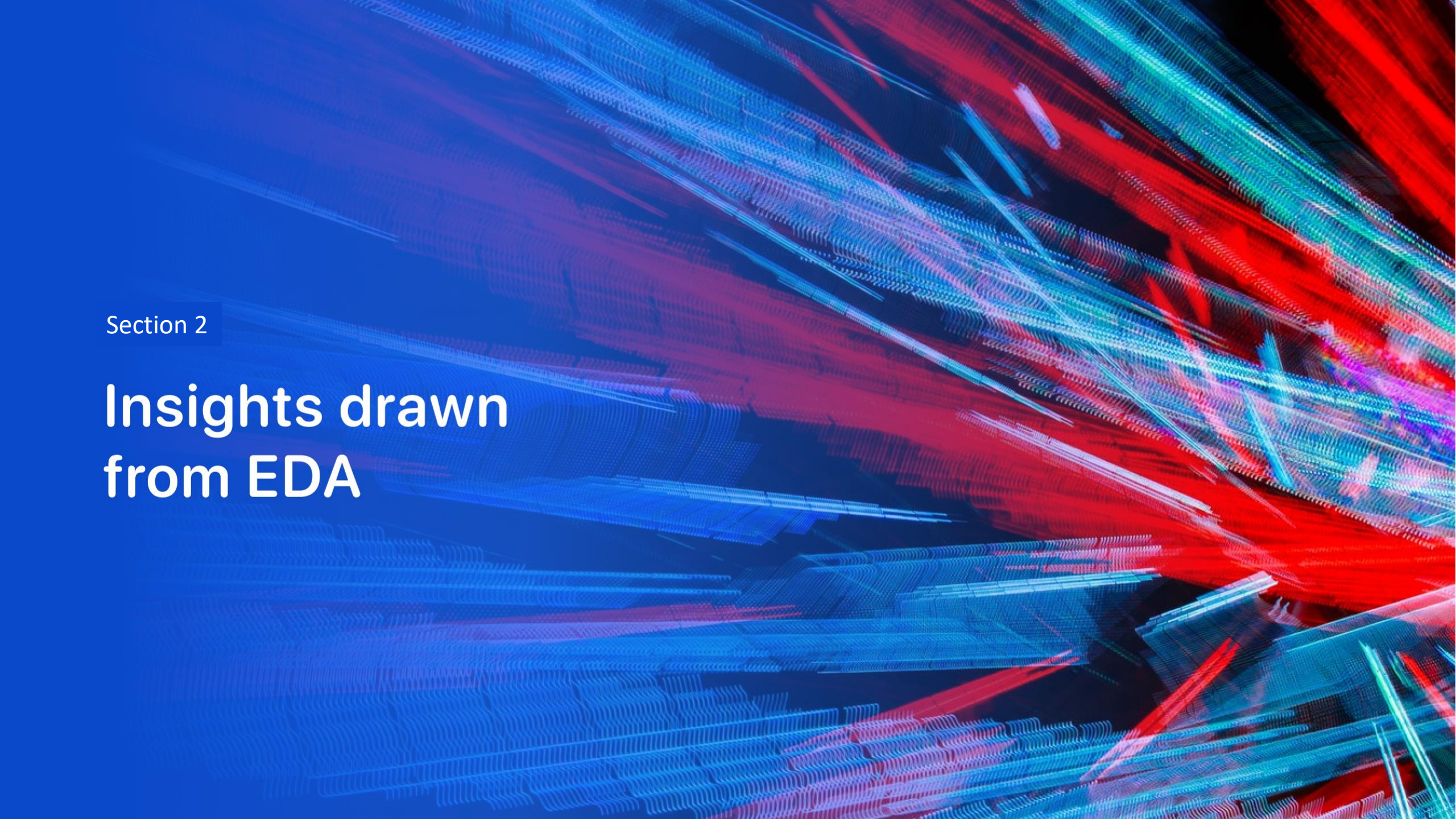
Identified the best-performing classification model.

Github URL: Below github URL contain notebook with code and results as well.

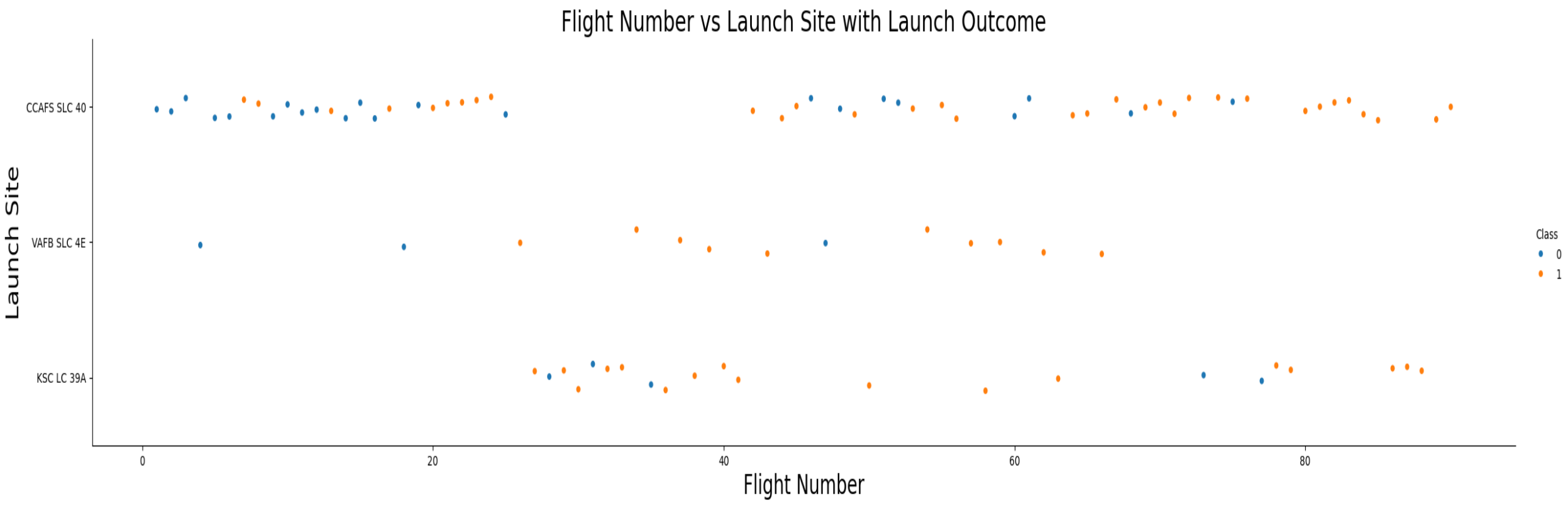
https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project/blob/main/SpaceX_Machine%20Learning%20Prediction.ipynb

Results

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

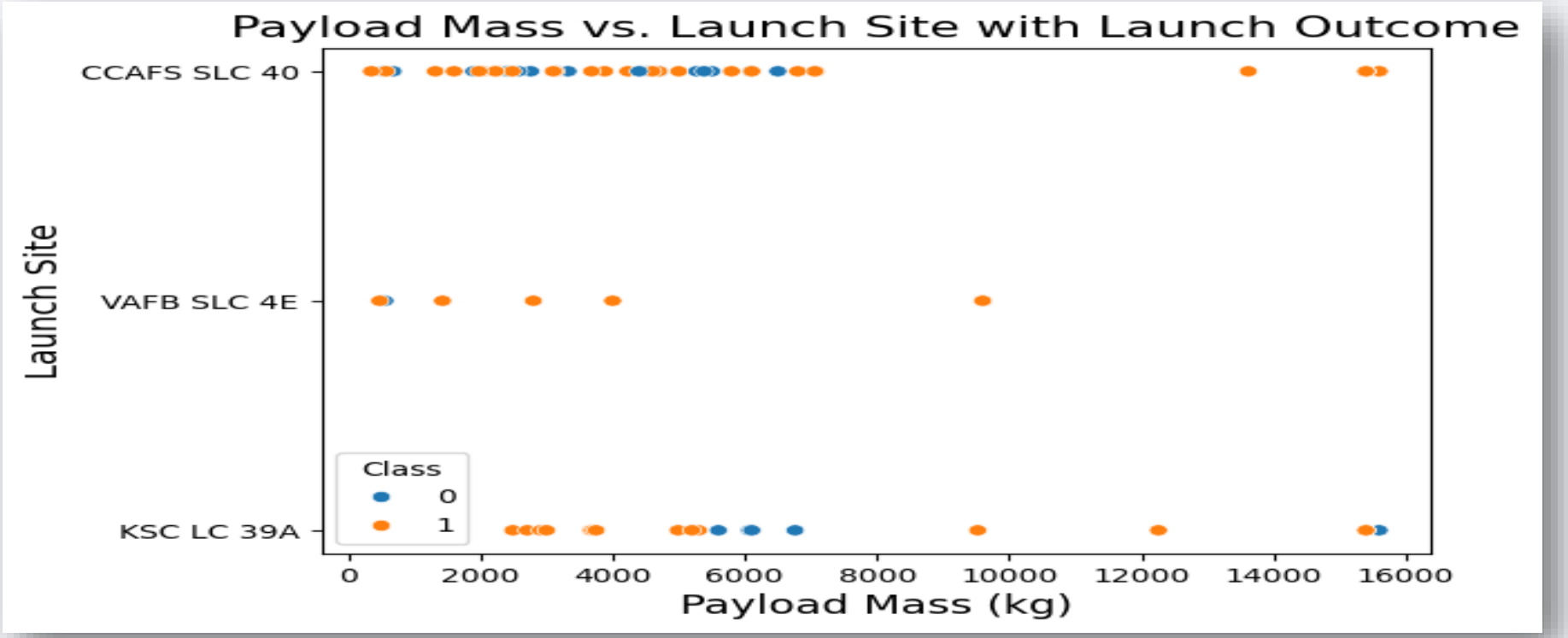

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



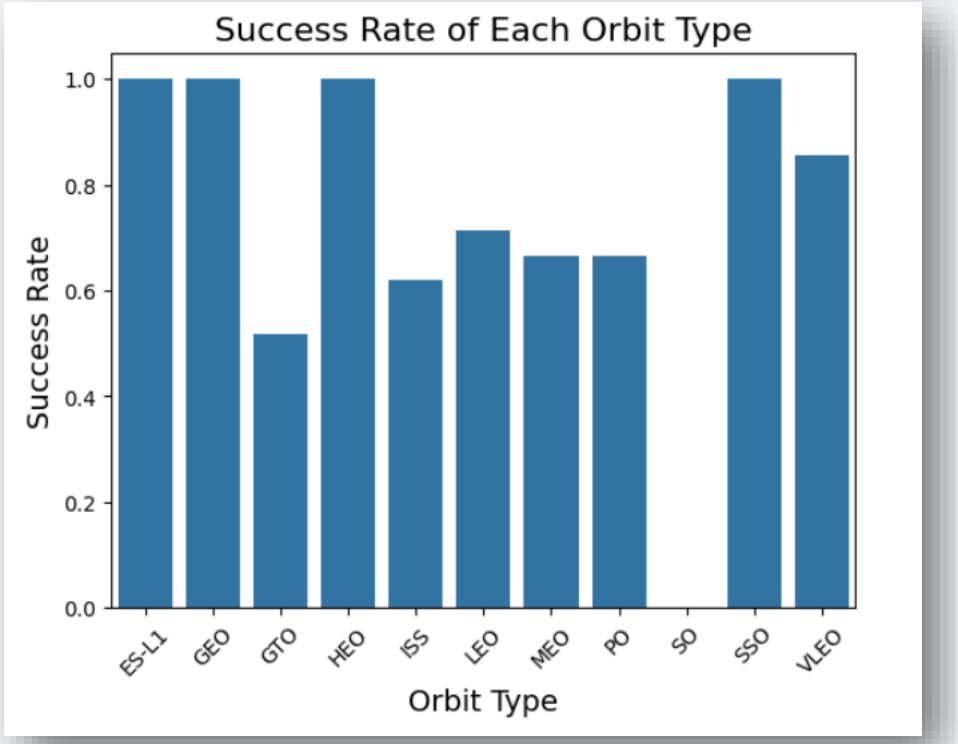
For Launch sites CCAFS SLC 40 and VAFB SLC 4E , as the flight numbers increased , # successful launches increase as well.

Payload vs. Launch Site



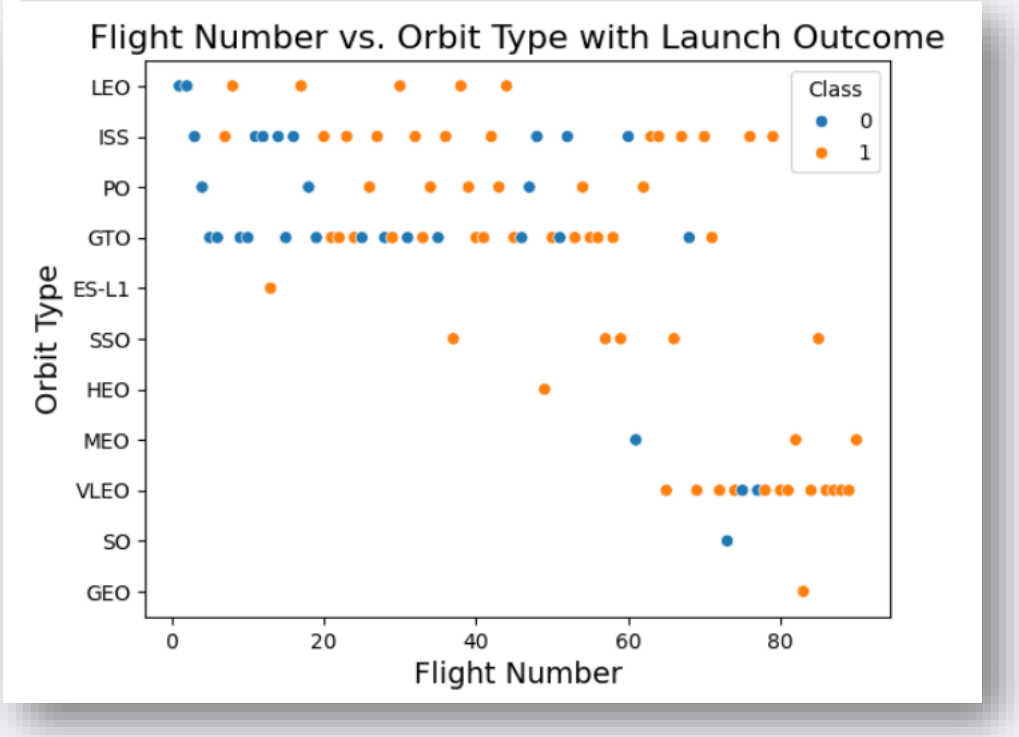
For Launch site CCAFS SLC 40 positive outcomes increased as payload mass increase while for site VAFB SLC 4E it has no impact.

Success Rate vs. Orbit Type



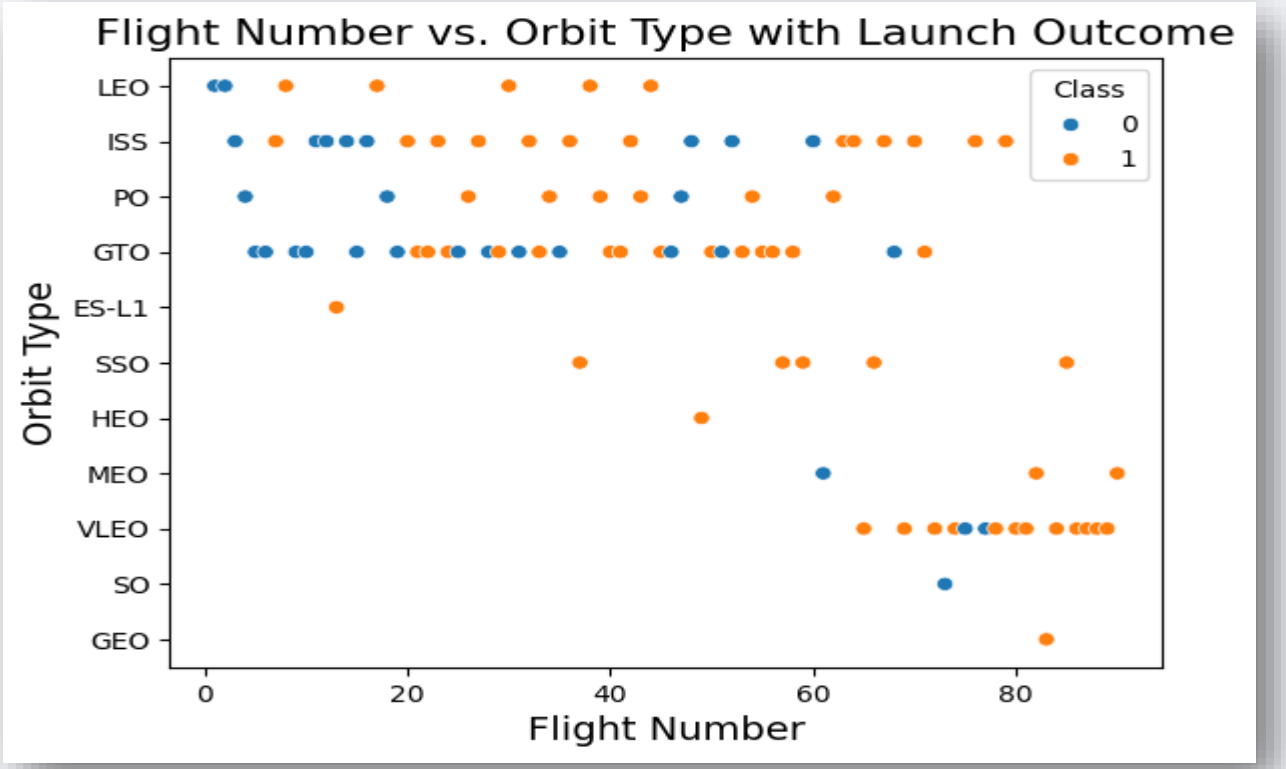
Orbits ES-L1,GEO,HEO and SSO has the highest success rate.

Flight Number vs. Orbit Type



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.

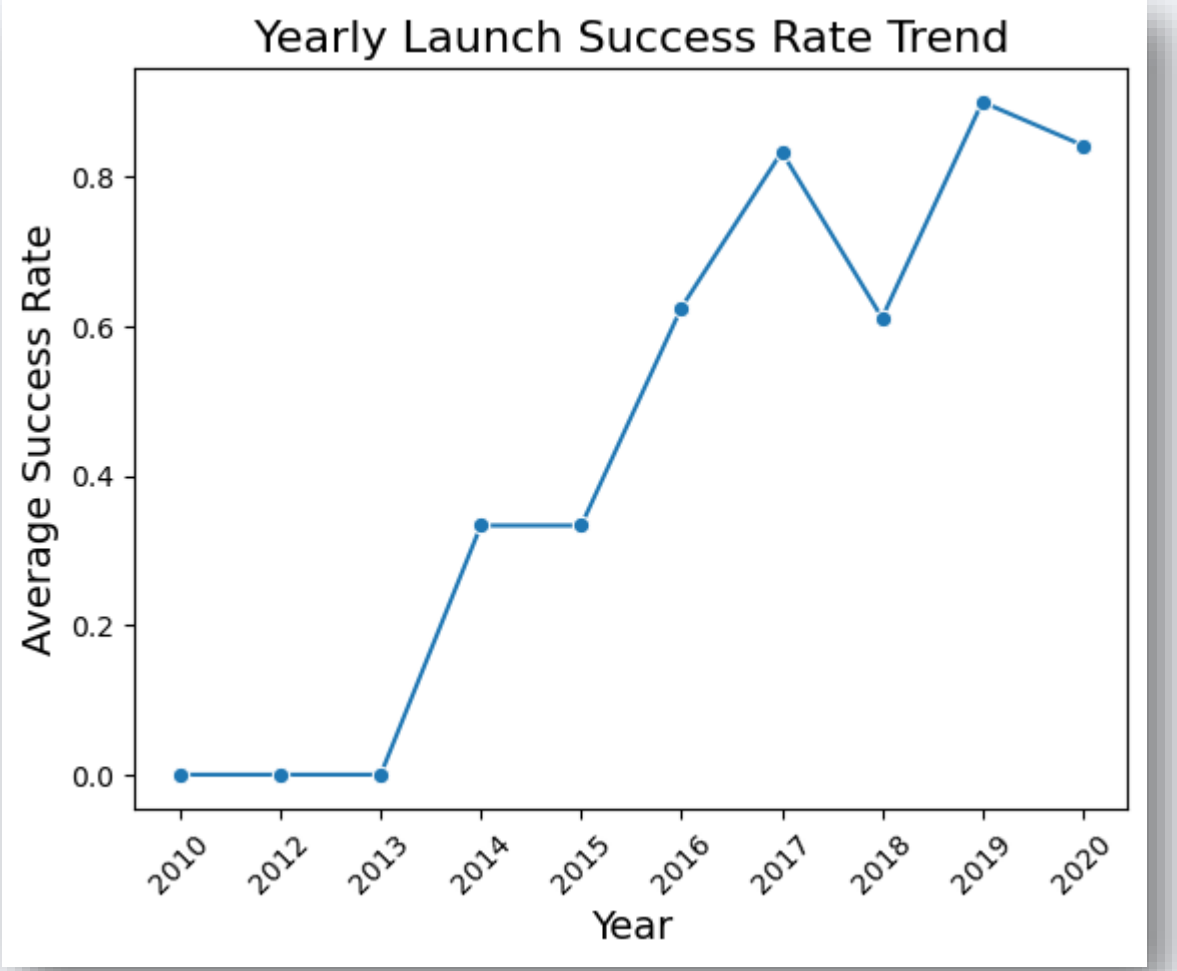
Payload vs. Orbit Type



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



Success rate kept increasing since 2013 till 2020.

All Launch Site Names

Unique launch site names are displayed as output of the query.

The query performs distinct count operation on Launch_Site column and find unique sites.

Display the names of the unique launch sites in the space mission

In [13]:
```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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

Out[13]:

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

Launch Site Names Begin with 'CCA'

Below query filter the records starting with CCA% in Launch_Site and then displays first 5 records using limit 5. % sign after CCA means anything can be after it , it should search the words starting with CCA.

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS) is 45596.

The query sums the values in PAYLOAD_MASS_KG column to give total mass.

Display the total payload mass carried by boosters launched by NASA (CRS)

In [16]:
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

Out[16]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1 is 2928.4.

The query averages the values of column PAYLOAD_MASS_KG to give avg mass.

Display average payload mass carried by booster version F9 v1.1

In [17]:
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS average_payload_mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

Out[17]:

average_payload_mass
2928.4

First Successful Ground Landing Date

The dates of the first successful landing outcome on ground pad is 22-15-2015.

Present your query result with a short explanation here

None

In [21]:
```
%sql SELECT MIN("Date") AS first_successful_landing_date  FROM SPACEXTABLE  WHERE "Landing_Outcome" = 'Success (ground pad)'
```

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

Out[21]:

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [22]:
```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)'   AND "PAYLOAD_MASS
```

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

Out[22]:

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

Total Number of Successful and Failure Mission Outcomes

We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

```
%sql SELECT "Mission_Outcome", COUNT(*) AS total FROM SPACEXTABLE WHERE "Mission_Outcome" LIKE 'Success%' OR "Mission_Outcome" LIKE 'Failure%' GROUP BY "Mission_Outcome";
```

[28]:
```
%sql SELECT "Mission_Outcome", COUNT(*) AS total FROM SPACEXTABLE WHERE "Mission_Outcome" LIKE 'Success%' OR "Mission_Outcome" LIKE 'Failure%' GROUP BY "
```

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

[28]:

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

Boosters Carried Maximum Payload

We found the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

```
[24]: %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
```

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

[24]:

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

2015 Launch Records

We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql SELECT substr("Date", 6, 2) AS month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM
SPACEXTABLE WHERE substr("Date", 1, 4) = '2015' AND "Landing_Outcome" LIKE '%drone ship%' AND
"Landing_Outcome" LIKE '%Failure%';
```

Done.

[30]:

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

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

We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
SELECT "Landing_Outcome", COUNT(*) AS count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY count DESC;
```

Done.

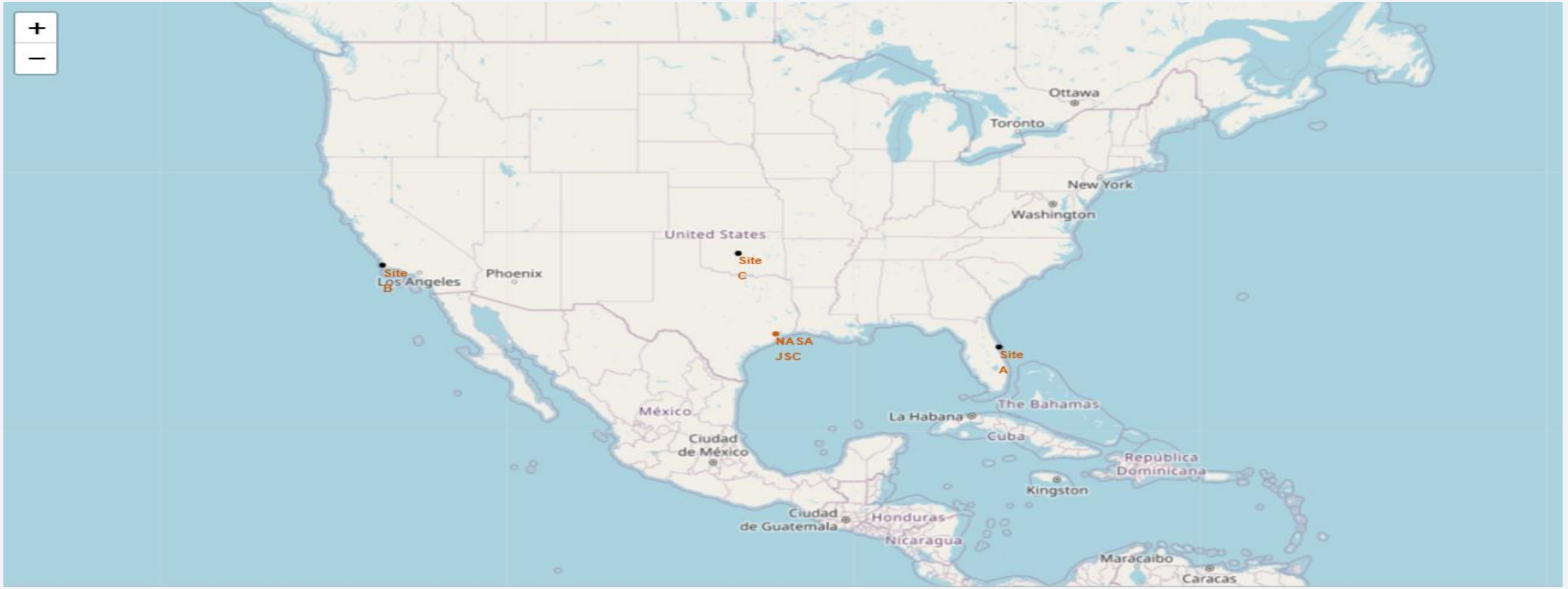
[26]:

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

Section 3

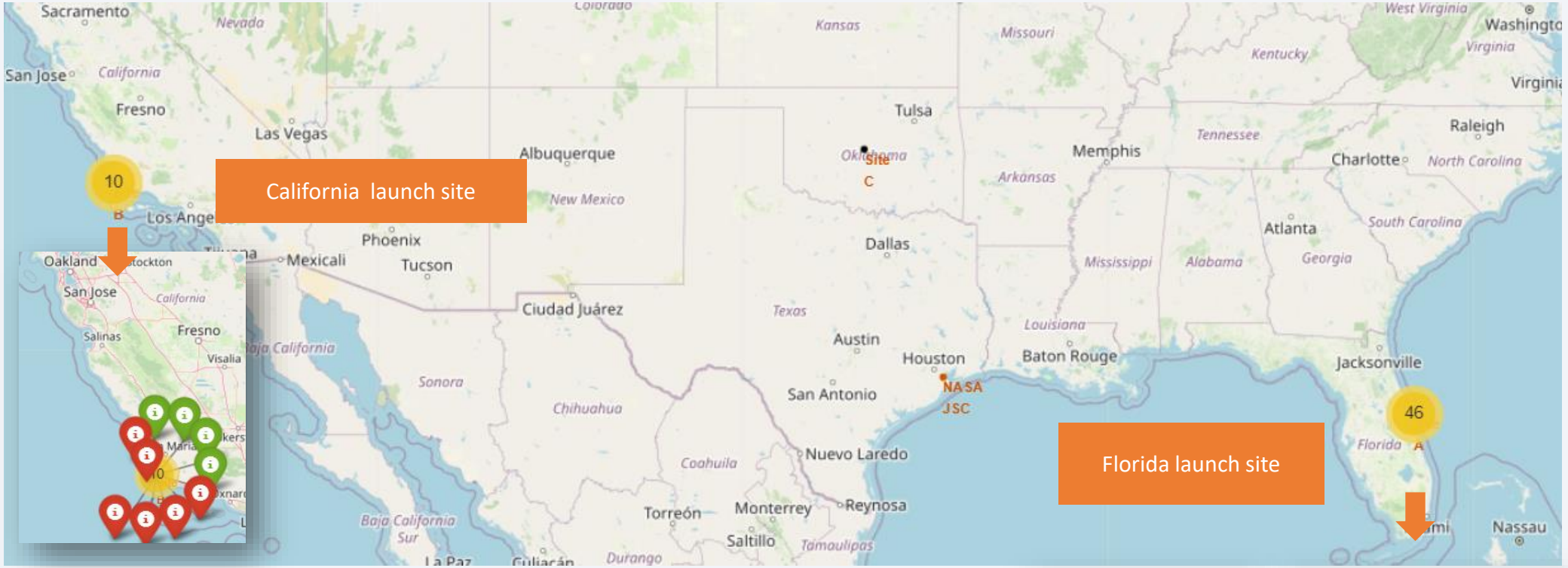
Launch Sites Proximities Analysis

All Launch Sites Map

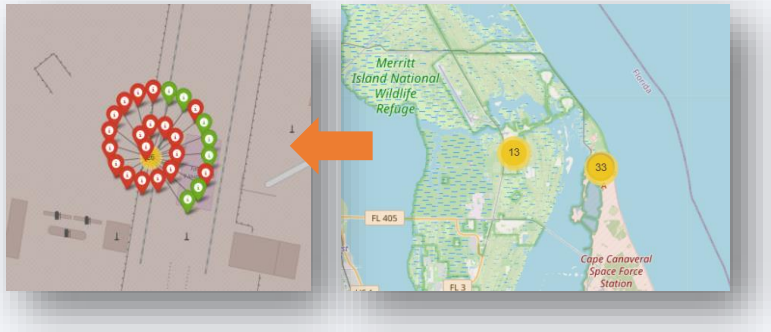


We can visualize all launch site locations on t map. he

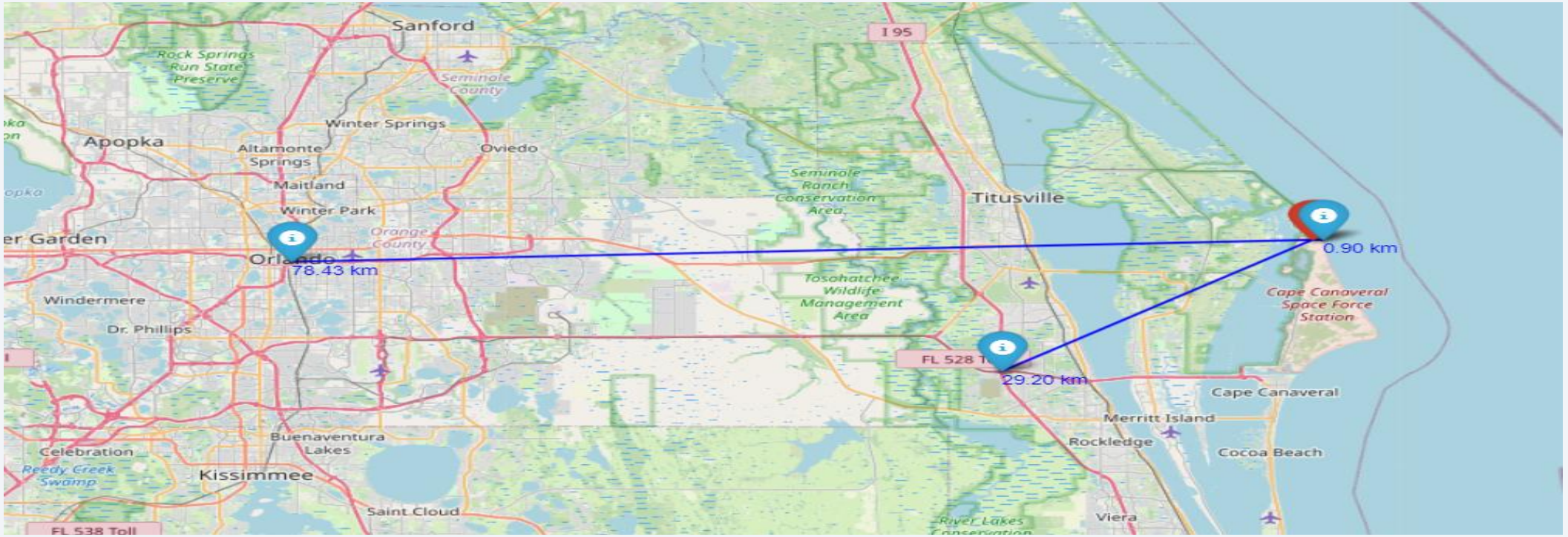
Markers showing launch sites with color labels



Green markers show successful landing and red markers show failed landings.

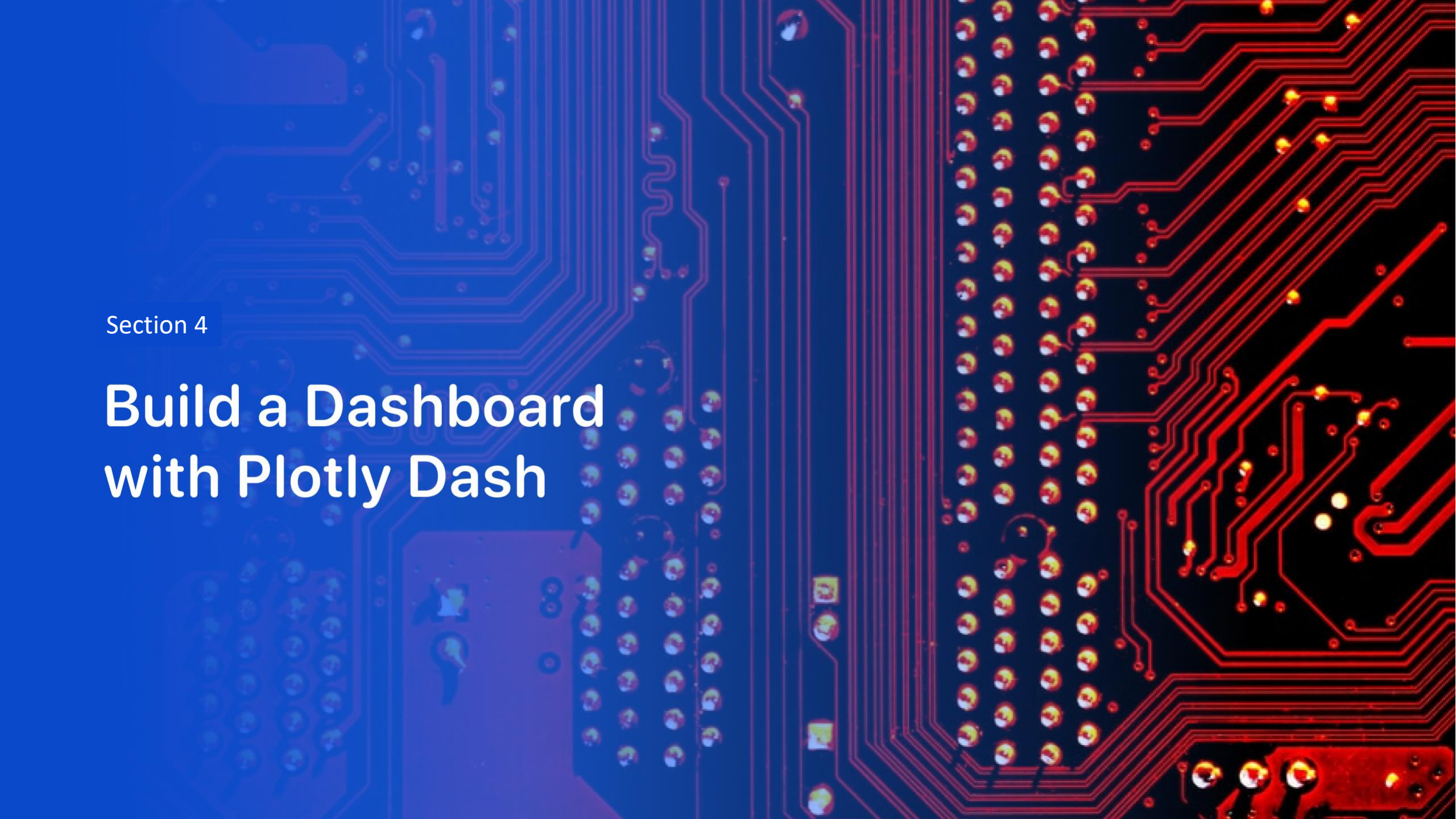


Launch Site distance to landmarks

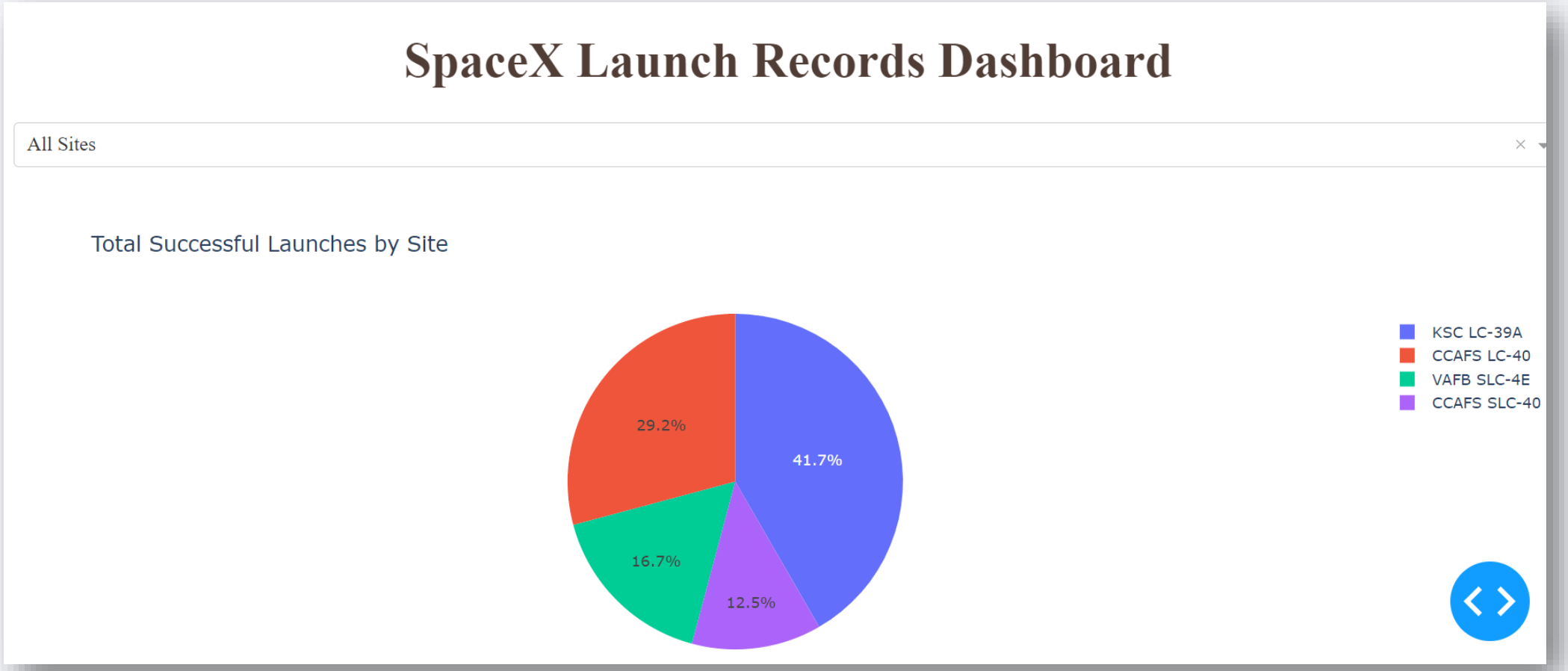


Observations

- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

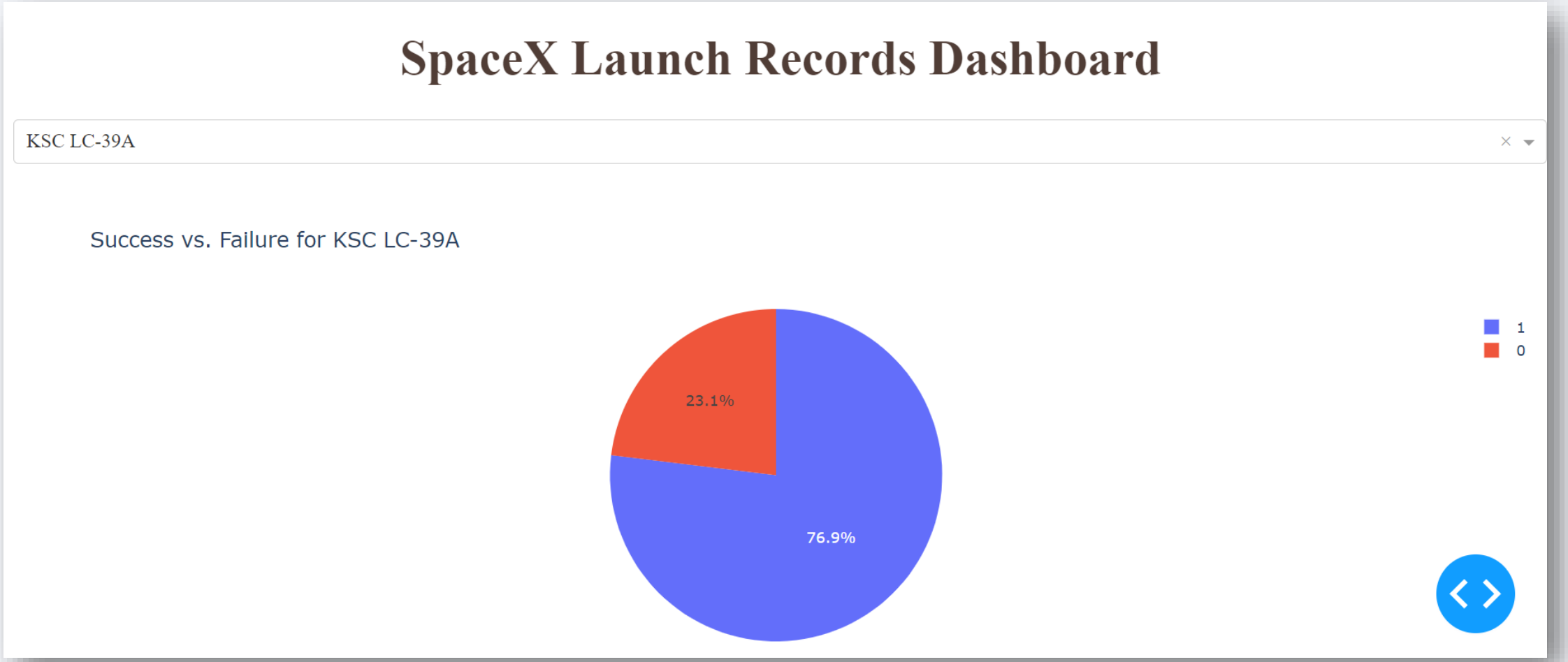

Section 4
Build a Dashboard with Plotly Dash

Success percentage achieved by each launch site



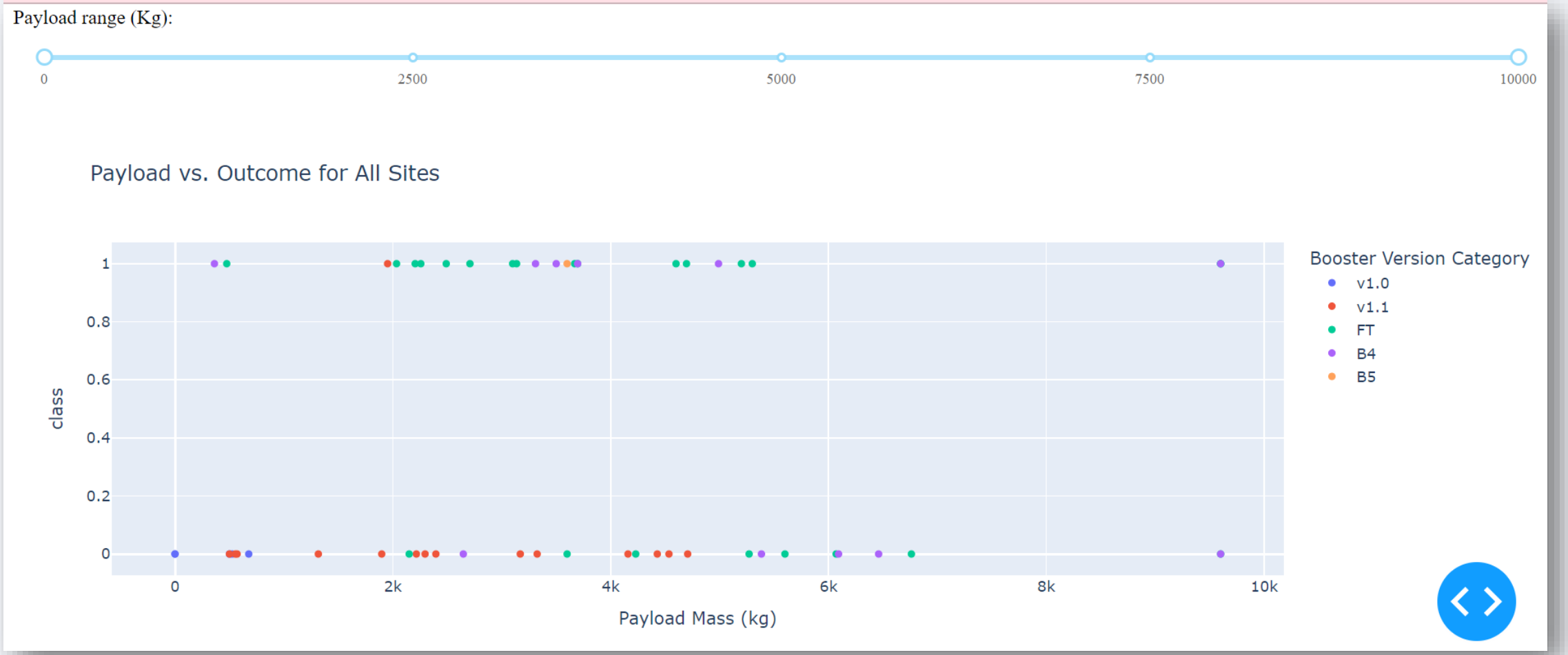
KSC LC-39A had the most successful launches

Launch site with the highest launch success ratio



KSC LC-39A had 76.9% success rate and 23.1% failure rate.

Payload vs Launch Outcome Scatterplot



Success rate of low weighted payloads is higher than the success rate of high weighted payloads.

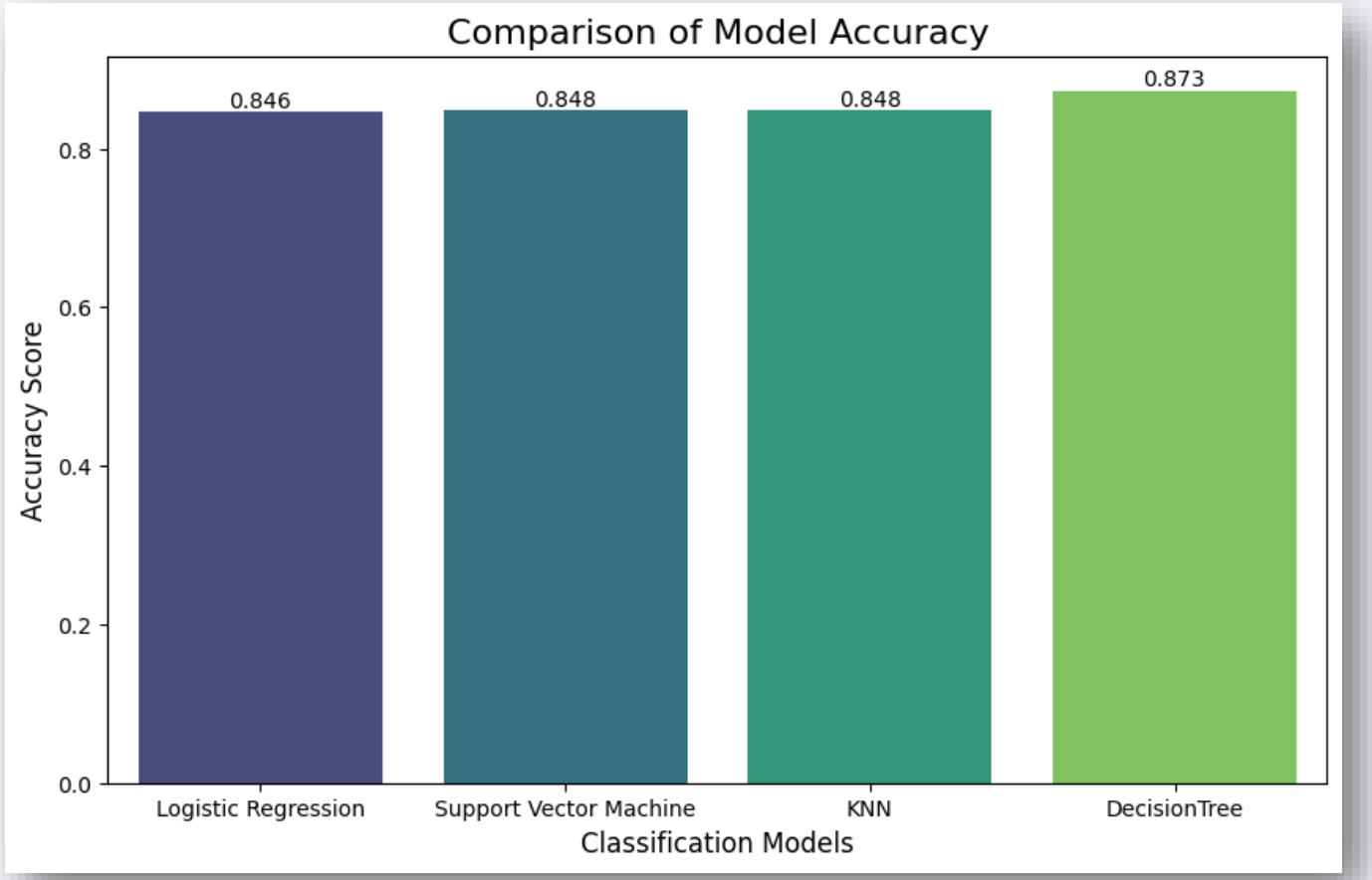
Section 5

Predictive Analysis (Classification)

Classification Accuracy

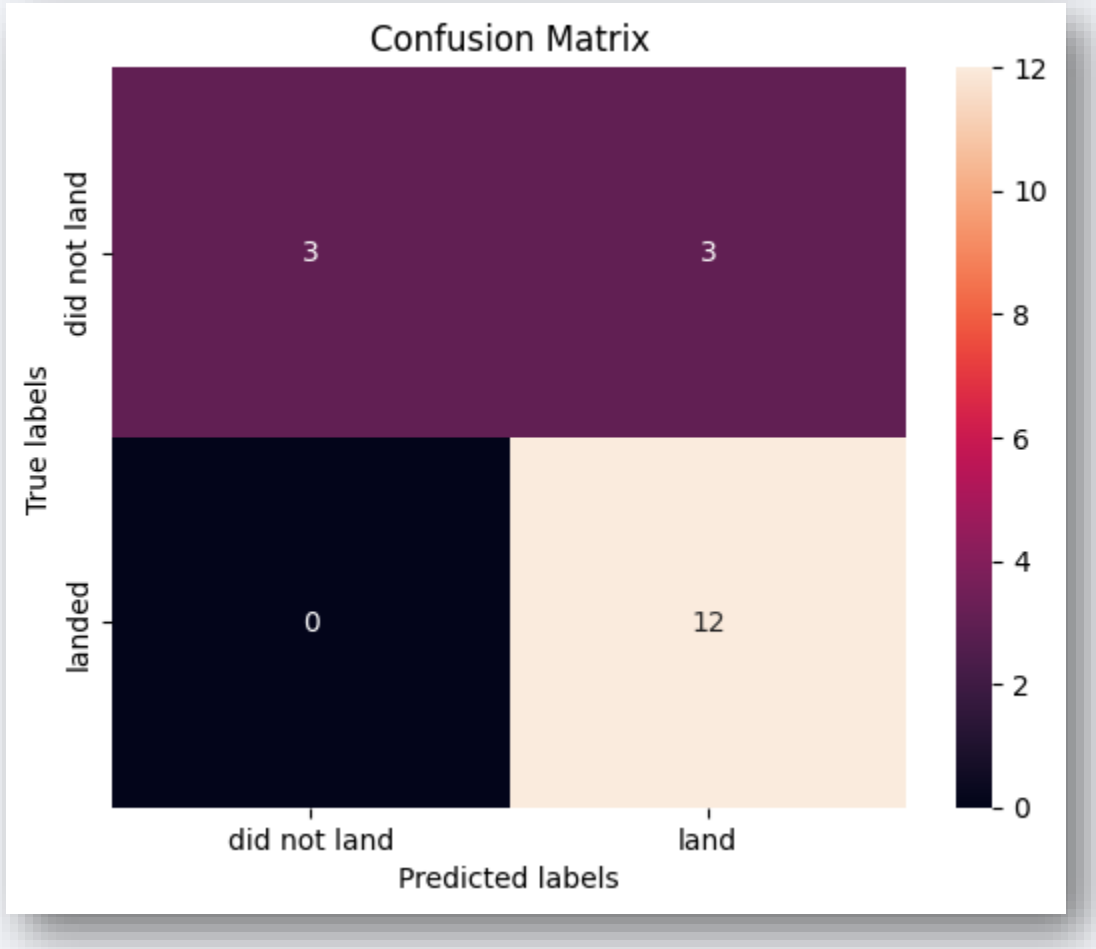
The bar chart shows classification models and corresponding accuracy scores.

Decision tree model has the highest classification accuracy



Confusion Matrix

The confusion matrix for the decision tree classifier indicates that the model effectively differentiates between classes. However, it has a significant issue with false positives, where unsuccessful landings are incorrectly classified as successful.



Conclusions

Below are the key findings:

A higher number of flights at a launch site is associated with an increased success rate at that site.

The success rate of launches began to rise from 2013 and continued to improve through 2020.

The orbits ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.

Among all launch sites, KSC LC-39A achieved the most successful launches.

The Decision Tree classifier emerged as the most effective machine learning algorithm for predicting launch success.

Innovative Insights

Impact of Launch Site Experience: The correlation between the number of flights and success rates at launch sites suggests that experience and infrastructure improvements at a site may contribute to better performance. Highlighting this could support the idea that investing in a launch site's capabilities and learning from past missions leads to higher success rates.

Temporal Trends in Success Rates: The increase in success rates from 2013 to 2020 could indicate advancements in technology, improved operational procedures, or better training. This trend might be tied to specific technological innovations or strategic changes made during this period.

Orbit-Specific Success Rates: The high success rates for specific orbits (e.g., ES-L1, GEO, HEO, SSO, VLEO) could provide insights into which types of missions are currently the most reliable. This information can be valuable for companies planning new missions or considering which orbits offer the most reliable outcomes.

Site-Specific Performance Analysis: KSC LC-39A's leading performance could be analyzed in detail to understand what factors contribute to its success. This might include infrastructure quality, the frequency of launches, or specific practices that could be replicated at other sites.

Machine Learning Model Insights: The Decision Tree classifier's superior performance suggests that this model effectively captures the complexities of launch success. Exploring why the Decision Tree performs better compared to other models could reveal important features or interactions in the data that influence launch success.

Predictive Value of Landing Success: Given that the success of the first stage landing is a significant cost factor for SpaceX, understanding how predictive features (such as weather, payload, or launch site characteristics) impact landing success could provide strategic advantages for cost estimation and competitive bidding.

Appendix

- The project with all notebook files can be found on below github link.
- https://github.com/rizwanfarooq780/IBM-Data-Science-Capstone-Project

Thank you!