IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ahmed Noor
07/06/2024

Outline

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

Executive Summary

This report presents the findings of our data science project, which aimed to predict the success of Falcon 9 first stage landings.

Summary of methodologies

Methodologies: We employed a robust data science methodology, beginning with *data collection and cleaning,* followed by *exploratory data analysis*. We then proceeded to *feature engineering* and *selection*, where we identified the most relevant variables for our models. We utilized various machine learning algorithms, I including *Decision Trees, K-Nearest Neighbors (KNN), Support Vector Machines (SVM),* and *Logistic Regression (Logreg)* to build predictive models. These models were rigorously validated using techniques such as *cross-validation*

Summary of all results

- Our analysis revealed that the success of a mission depends on several factors, including the *launch site, orbit, and the number of previous launches. Knowledge gained from previous launches* contributes to transitioning from failure to success. Secondly, our models achieved an *accuracy of approximately 83% on test data*, demonstrating their effectiveness in predicting the outcome. Furthermore, the *Decision Tree Algorithm emerged as the most effective model* based on its superior accuracy score on training data.
- In terms of trends, the *success rate shows an upward trend over the decade*, suggesting that SpaceX has been improving its rocket launches, achieving higher success rates as time progresses.

Introduction

❖Project background and context

The focus of this project is to predict whether the Falcon 9 first stage will successfully land. SpaceX, on its website, states that the Falcon 9 rocket launch costs 62 million dollars, while other providers charge upwards of 165 million dollars per launch. The cost difference arises from SpaceX's ability to reuse the first stage. By determining the likelihood of successful landings, we can estimate launch costs. This information is valuable for any company aiming to compete with SpaceX in the rocket launch industry

❖Problems you want to find answers

1. *Identify key factors contributing to failed landings & successful landings?*
2. *Investigate relationships between Rocket variables and success/failure rates?*
3. *Determine the specific conditions that maximize SpaceX's landing success rate?*

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - ***SpaceX REST API:*** Employed to collect comprehensive data on rocket launches and landings, including details about the rocket, payload, launch, and landing specifications.
  - ***Web Scraping from Wikipedia:*** Utilized web scraping techniques to gather additional information from Wikipedia, ensuring a robust dataset
- Perform data wrangling
  - ***Dropping Unnecessary Columns****:* Cleaned the dataset by removing redundant and irrelevant columns to streamline the analysis process.
  - ***One Hot Encoding for Classification Models****:* Applied One Hot Encoding to categorical variables, enabling effective use in classification models.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - *Built, tuned, and evaluated* multiple classification models to ensure accuracy and reliability in predictions

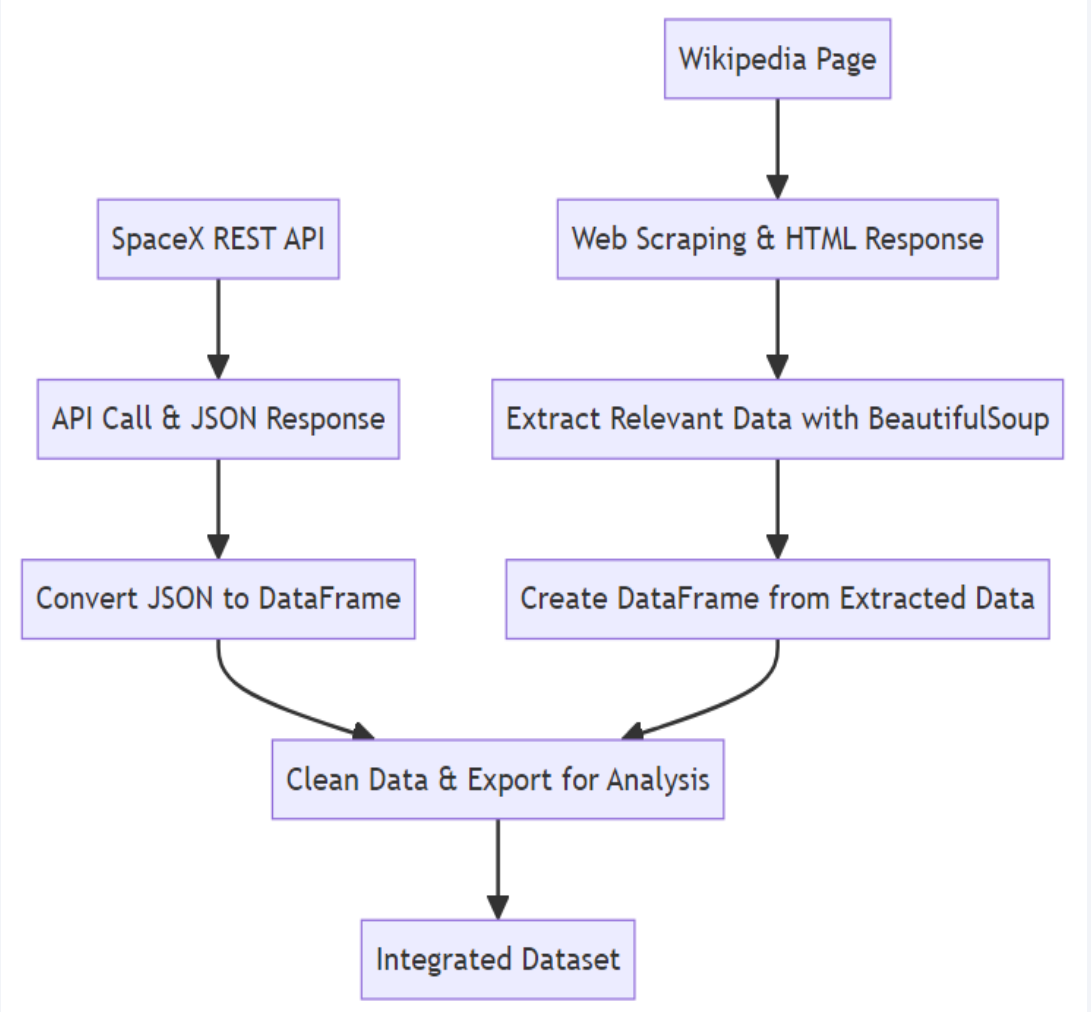
Data Collection

❖*SpaceX REST API*

- ***Objective:*** Collect comprehensive data on rocket launches, payload, and landing information
- Space X REST API URL is api.spacexdata.com/v4/

❖*Web Scraping from Wikipedia*

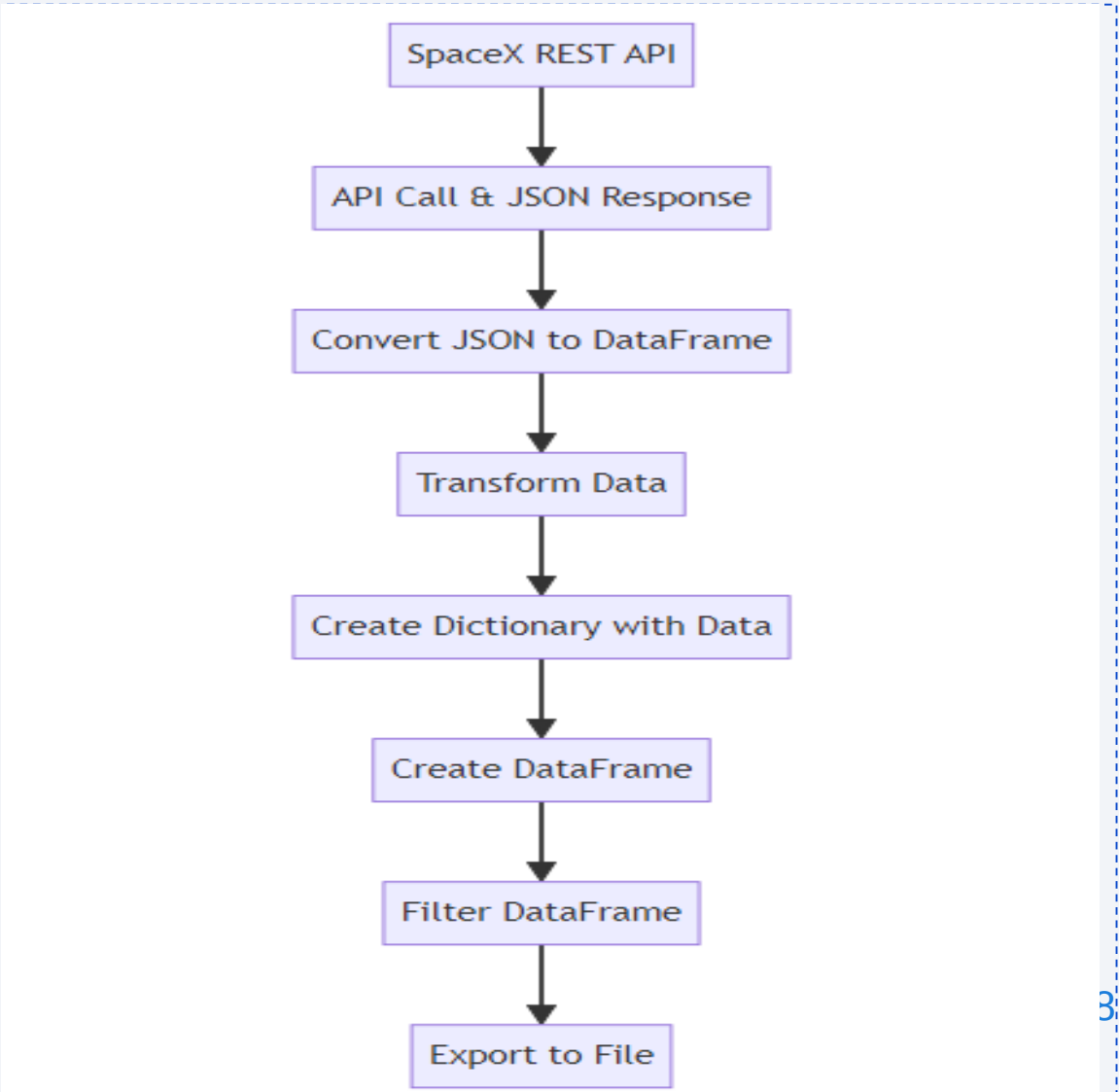
- ***Objective:*** Augment the dataset with additional launch, landing, and payload information
- **URL** is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

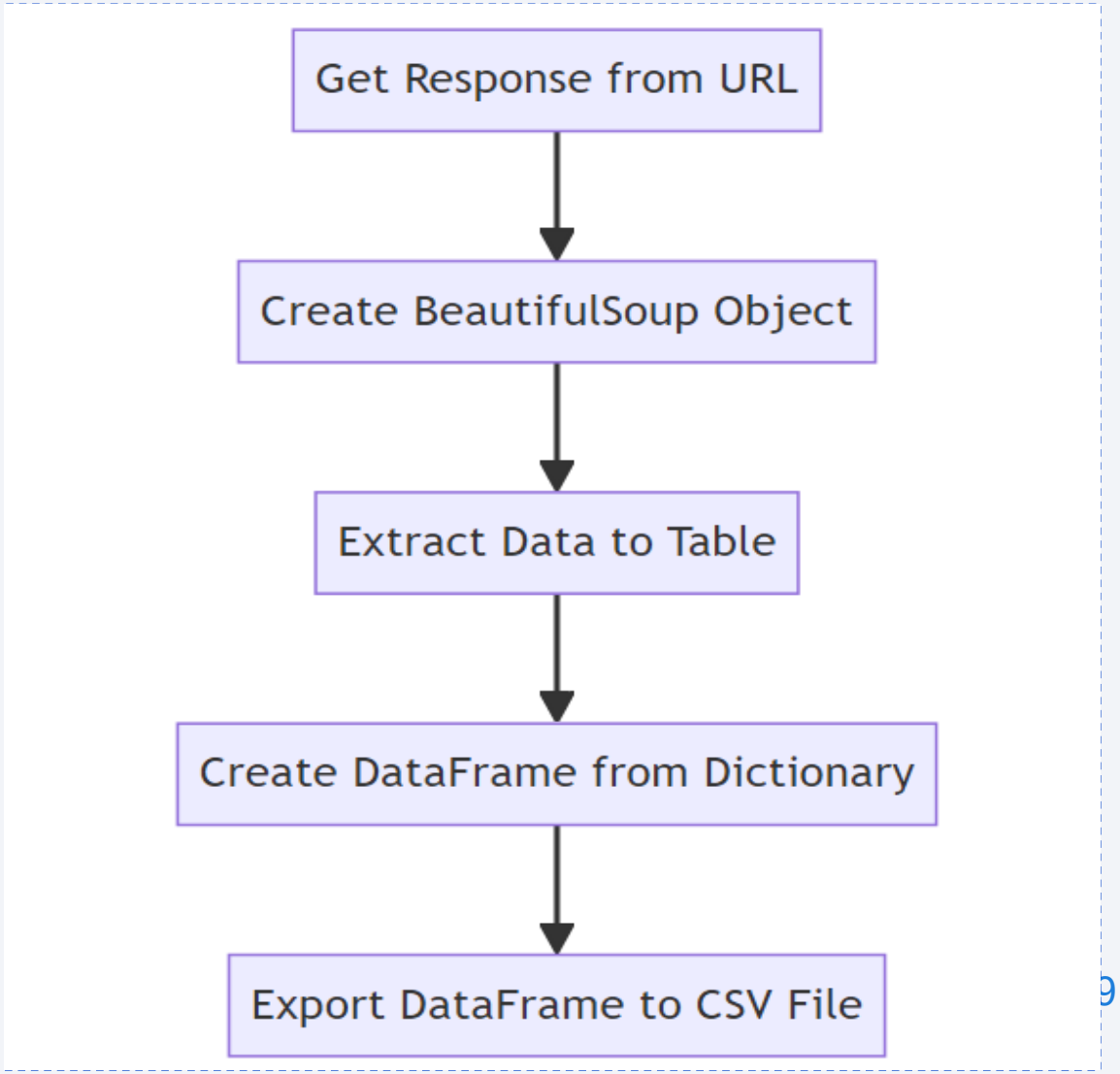
- *Make an API call* to the SpaceX REST API.
- *Convert the JSON response* after receiving into a structured format e.g a DataFrame.
- *Process and manipulate* the data as needed.
- *Organize the data into a dictionary* for further analysis.
- *Construct a DataFrame* from the processed data.
- *Apply filters* or conditions to extract specific subsets of data.
- *Save* the filtered data to a CSV file

Link to code



Data Collection - Scraping

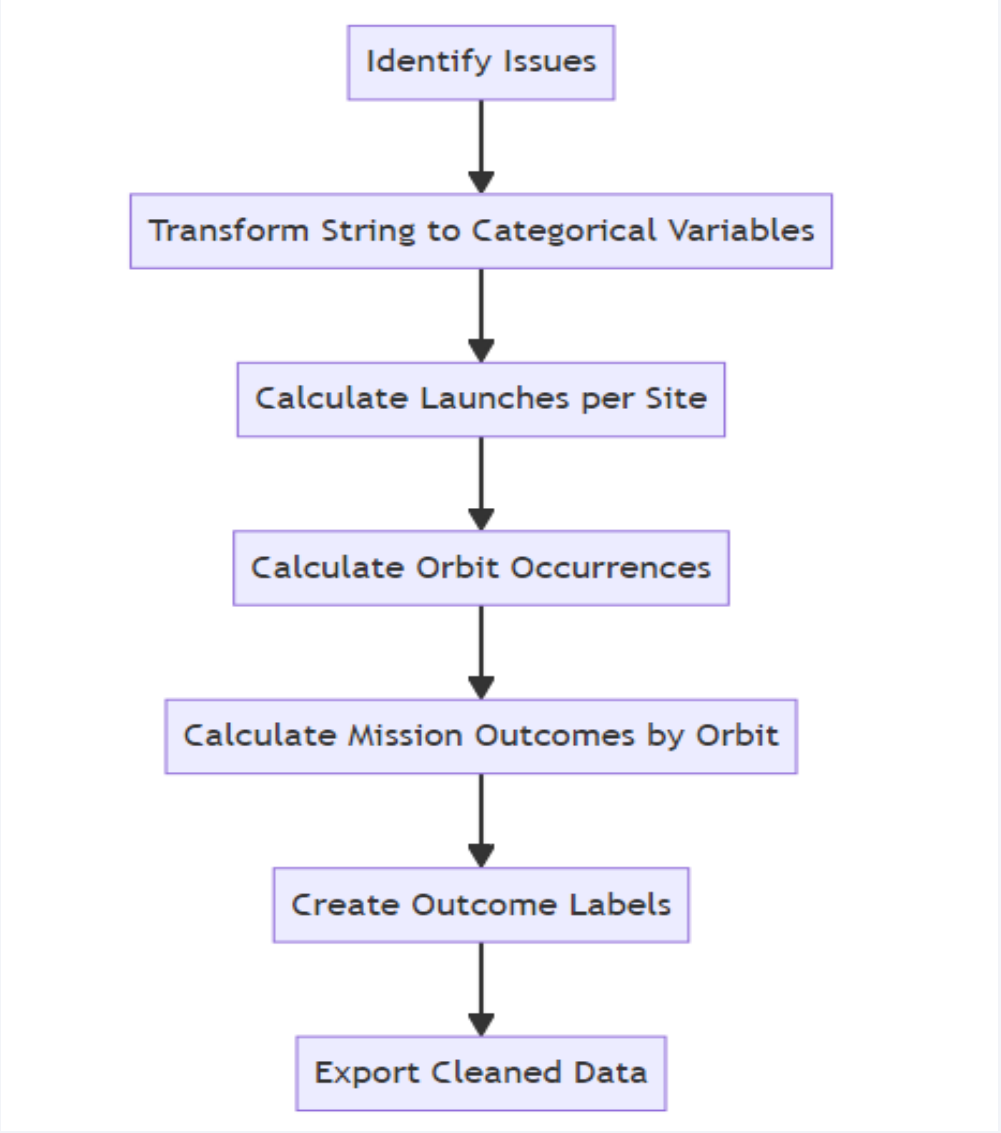
- *Send an HTTP* request to the Wikipedia page to receive the HTML response.
- *Use the BeautifulSoup library* to parse and then Extract relevant data from the HTML.
- *Identify the table and then extract* rows and columns from the table.
- *Create and then convert the dictionary* into a Pandas DataFrame.
- *Save the DataFrame* to a CSV file.



Link to code

Data Wrangling

- In the dataset, identify cases where the booster did not land successfully.
- Convert string variables into categorical variables:
  - *1 indicates mission success.*
  - *0 indicates mission failure*
- Use the *`value_counts()`* method to calculate the number of launches for each site.
- Calculate the *number and occurrence of each orbit* type.
- Determine the *number and occurrence of mission outcomes* per orbit type.
- Create a *new column `Class`* to label the outcome of each mission as *success (1) or failure (0).*
- Export the cleaned and processed data to a *CSV* file.



Link to code

EDA with Data Visualization

- Scatter Graph:

Scatter plots show relationship between variables. This relationship is called the correlation

Purpose:

To explore if there is a correlation between

- *Number of flights vs the payload mass.*
- *Flight Number vs. Launch Site.*
- *Payload vs. Launch Site.*
- *Orbit vs. Flight Number.*
- *Payload vs. Orbit Type.*
- *Orbit vs. Payload Mass.*

Link to code

- Bar Graph

Bar graphs show the relationship between numeric and categoric variables.

Purpose:

To compare the success rates of launches based on orbit types

- Line Graph

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data

Purpose:

To track the success rate of launches over time.

EDA with SQL

Here's the summary of the SQL queries performed

- **Launch Sites Beginning with 'CCA'**: This query displays records where the launch site starts with the string 'CCA'.
- **Unique Launch Sites**: *This query shows 5 unique launch site names in the space mission*
- **Average Payload Mass by NASA (CRS)**: *This query calculates and displays the average payload mass carried by boosters launched by NASA (CRS).*
- **Dates with Specific Criteria**: *This query lists dates when both successful landing outcomes occurred and payload mass was between 4000 and 6000.*
- **First Successful Drone Ship Landing**: *This query identifies the name of the booster associated with the first successful landing achieved on a drone ship at the ground pad.*
- **Total Booster Versions**: *This query counts the total number of booster versions that have carried out missions*
- **Mission Outcomes**: *This query provides items related to successful and failure mission outcomes.*
- **Monthly Records for 2015**: *This query lists records showing months, failure landing outcomes in drone ship, booster versions, and launch sites for the year 2015.*

Link to code

Build an Interactive Map with Folium

1. ***Centered Map***: *The folium map is centered on NASA Johnson Space Center at Houston, Texas. This is the central reference point for mapping launch sites.*

2. ***Red Circle at NASA Johnson Space Center***: *A red circle marker at NASA Johnson Space Center with a popup label showing its name. To identify the central location on the map.*

3. ***Red Circles for Launch Sites***: *Red circle markers at each launch site coordinate with labels showing the launch site names. To visually represent all launch sites on the map.*

4. ***Marker Cluster***: *Grouping of points in a cluster to display multiple pieces of information for the same coordinates. To prevent overlap and clutter.*

5. ***Markers for Landings***: *Green markers for successful landings and red markers for unsuccessful landings. To distinguish visually between successful and unsuccessful landings.*

6. ***Distance Markers and Lines***: *Markers showing distances between launch sites and key locations (railway, highway, coastway, city), with lines plotted between them. To understand the geographic relationship between launch sites and important infrastructure.*

By adding these objects, the map becomes a comprehensive tool for visualizing the locations of launch sites, their success rates, and their proximities to key infrastructures, enhancing understanding and analysis of the data.

Link to code

Build a Dashboard with Plotly Dash

1. **Dropdown:**
   - Allows the user to choose a specific launch site or view data for all launch sites.

   "dash_core_components.Dropdown"

2. **Pie Chart:**
   - Displays the total number of successful and failed launches for the selected launch site.

   "plotly.express.pie"

3. **RangeSlider:**
   - Allows the user to select a range of payload mass values.

   "dash_core_components.RangeSlider"

4. **Scatter Plot:**
   - Shows the relationship between launch success and payload mass.

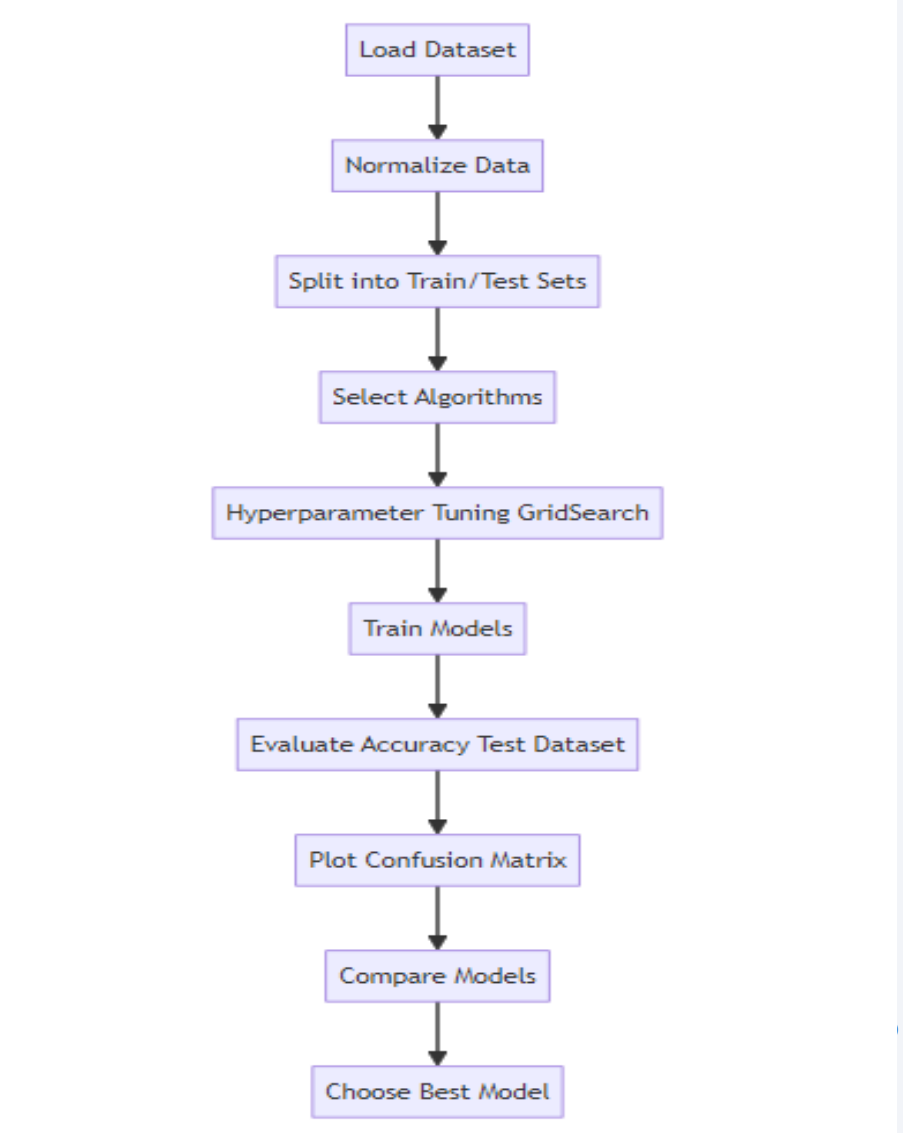
   "plotly.express.scatter"

These components provide a comprehensive and interactive way to explore and analyze launch data, making the dashboard a powerful tool for understanding the factors influencing launch success.

Link to code

Predictive Analysis (Classification)

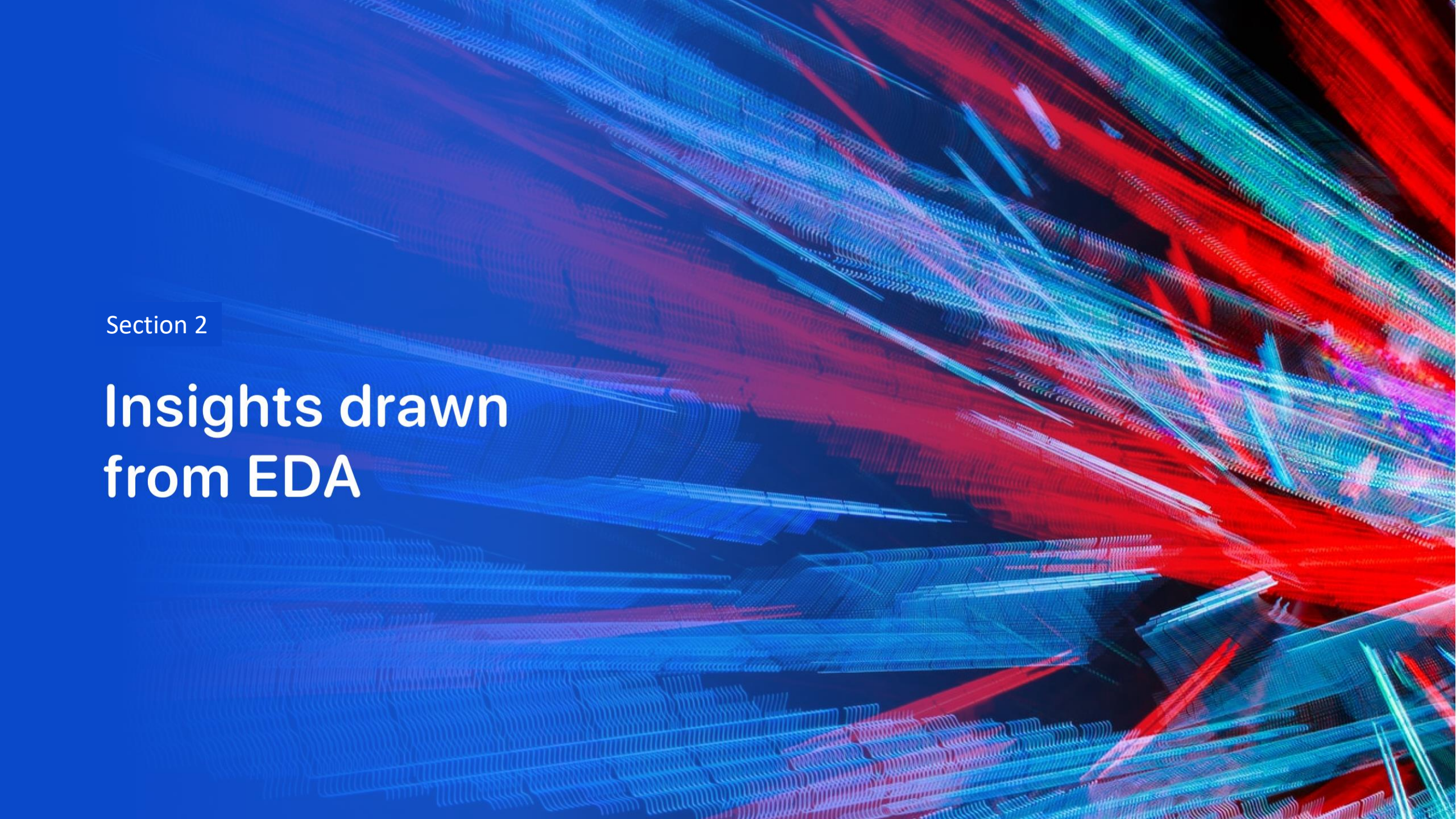
1. Data Preparation:
   - Import the dataset
   - Standardize the data to ensure uniformity.
   - Divide the dataset into training and test sets
2. Model Preparation:
   - Choose suitable machine learning algorithms for the classification task.
   - Define parameters for each algorithm using `GridSearchCV`.
3. Training Models:
   - Train the models with the training dataset using `GridSearchCV` to find the best parameters.
4. Model Evaluation:
   - Extract the best hyperparameters for each model.
   - Calculate the accuracy of each model using the test dataset
   - Plot the confusion matrix to visualize the performance
5. Model Comparison:
   - Assess models based on their accuracy and choose the model with the highest accuracy



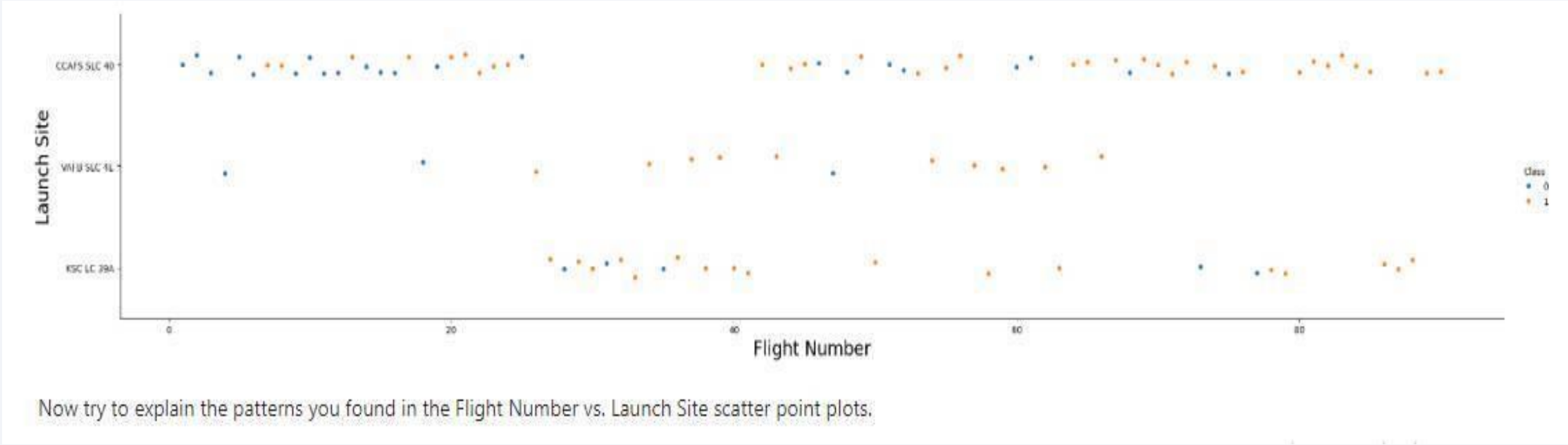
Link to code

Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



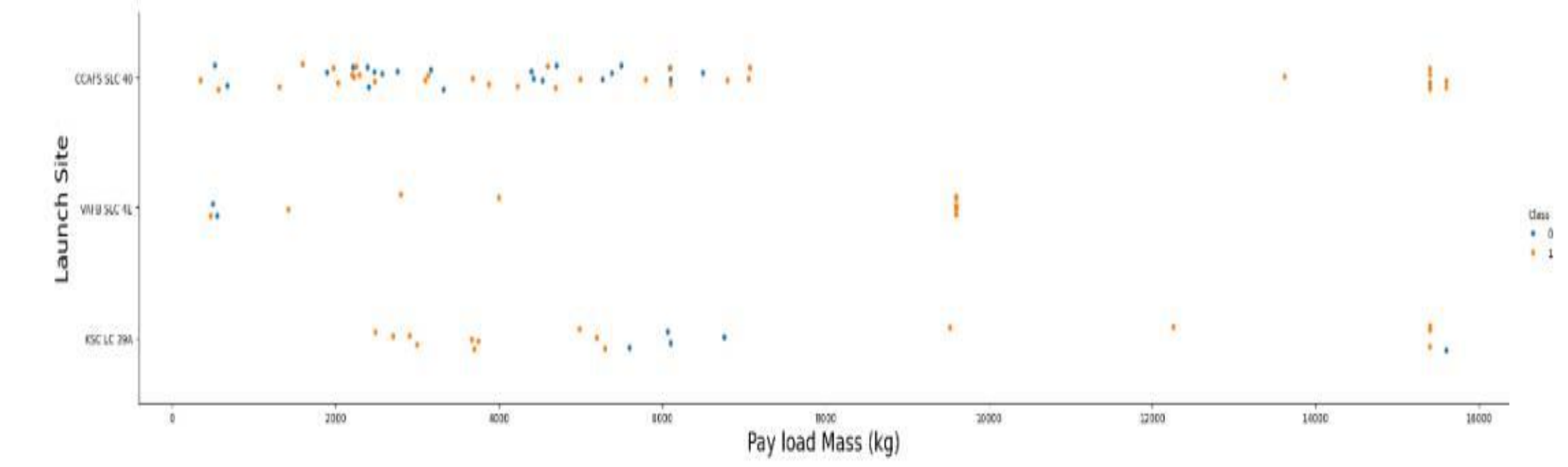
Explanation:

- *Improved Success Rates Over Time*: Across all launch sites, success rates have improved, likely due to increased experience and technological advancements.

- *Efficient Launch Sites*: VAFB SLC 4E and KSC LC-39A demonstrate high efficiency with fewer failures.

This analysis highlights the significant improvements in launch success over time and the relative efficiency of different launch sites.

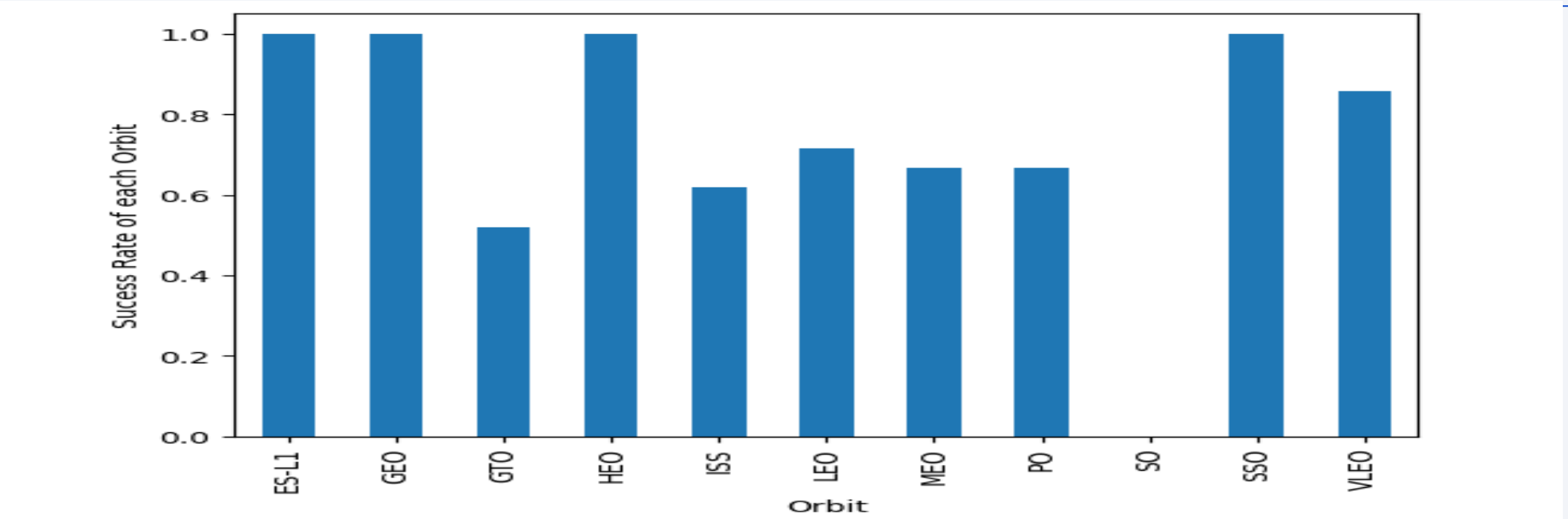
Payload vs. Launch Site



Explanation:

- *In summary, CCAFS SLC 40 and KSC LC-39A are active sites with successful launches across various payload masses. VAFB SLC 4E, though less frequent, also maintains a high success rate. The overall trend suggests that SpaceX has achieved reliability across different payload masses at these launch sites*

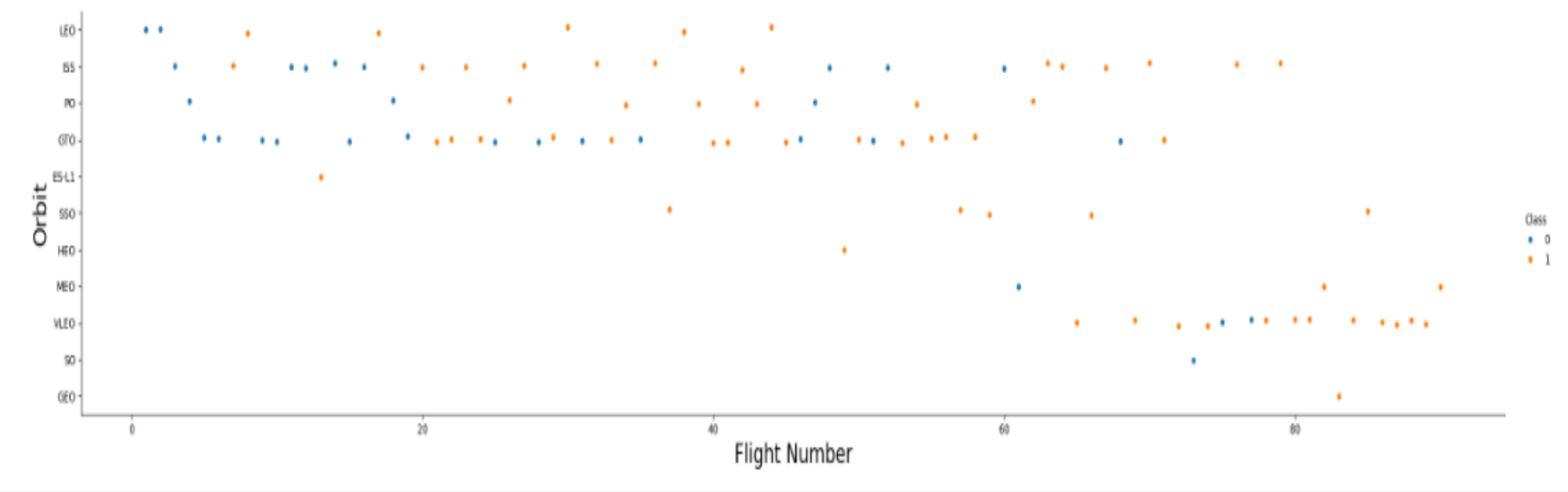
Success Rate vs. Orbit Type



Explanation:

- In summary, ES-L1, GEO, and VLEO orbits have the highest success rates, while ISS and LEO orbits are slightly lower. Understanding these success rates helps in planning future space missions effectively

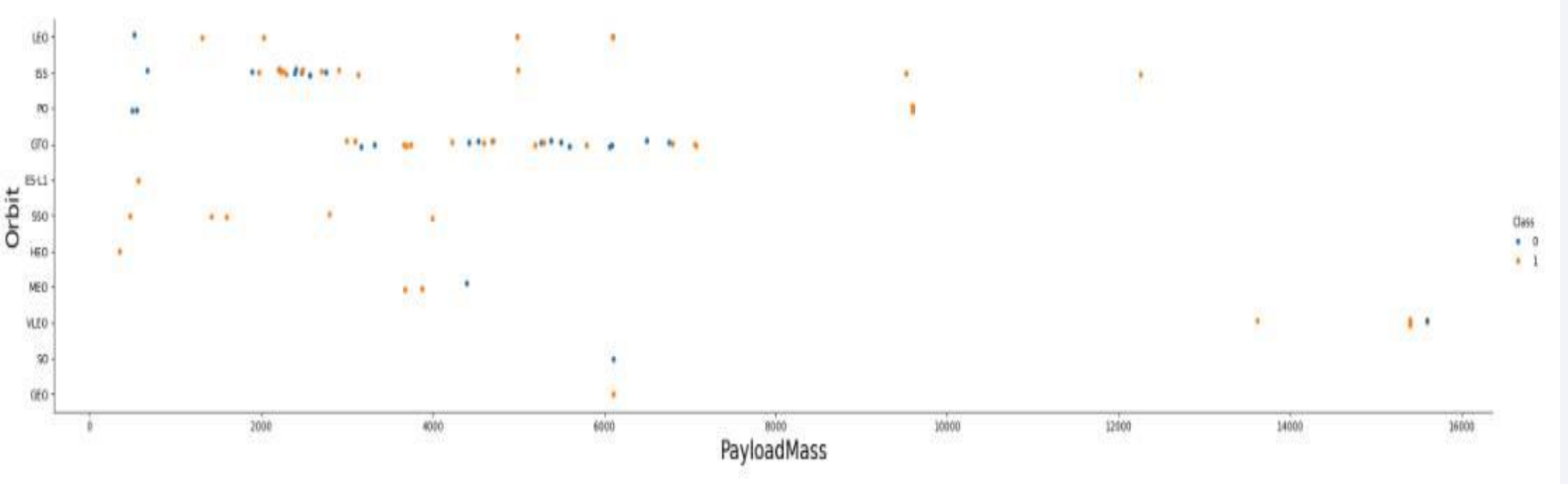
Flight Number vs. Orbit Type



Explanation:

- In summary, ES-L1, GEO, and VLEO orbits have the highest success rates, while ISS and LEO orbits are slightly lower. Understanding these success rates helps in planning future space missions effectively

Payload vs. Orbit Type

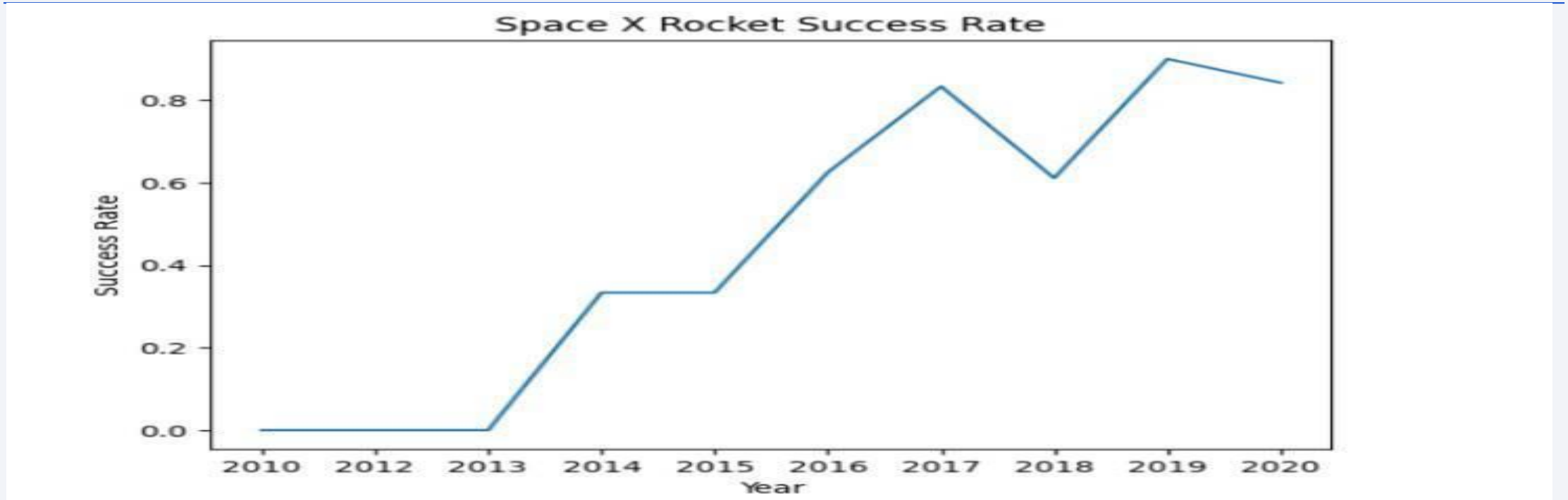


Explanation:

- Successful flights tend to cluster around certain orbit types, especially in the medium and high payload mass ranges.

- Unsuccessful flights lack a clear pattern and are more dispersed

Launch Success Yearly Trend



Explanation:

-**Overall Trend**: The success rate shows an upward trend over the decade. This suggests that Space X has been improving its rocket launches, achieving higher success rates as time progresses

In summary, Space X's rocket success rate has generally improved over the decade, with occasional setbacks but consistent progress

All Launch Site Names

Explanation:

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE and output only unique launch sites .

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

```
[9]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

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

[9]:

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

[10]:
```
%sql SELECT * \
FROM SPACEXTABLE \
WHERE "Launch_Site" LIKE 'CCA%' \
LIMIT 5
```

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

[10]:

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

Explanation:

The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering.

Total Payload Mass

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

[11]:
```
%%sql
SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass
FROM SPACEXTABLE
WHERE "Customer" = 'NASA (CRS)';
```

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

[11]:

Total_Payload_Mass
45596

Explanation:

This query returns the sum of all payload masses where the customer is NASA (CRS)

Average Payload Mass by F9 v1.1

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

[12]:
```
%%sql
SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

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

[12]:

Average_Payload_Mass
2928.4

Explanation:

This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

[13]:
```
%%sql
SELECT MIN("Date") AS First_Successful_Landing
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

[13]:

First_Successful_Landing
2015-12-22

Explanation:

With this query, we select the oldest successful landing. The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date..

Successful Drone Ship Landing with Payload between 4000 and 6000

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

[14]:
```
%%sql SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "Payload_Mass__kg_" > 4000
  AND "Payload_Mass__kg_" < 6000;
```

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

[14]:

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

Explanation:

This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[31]:
```
%%sql SELECT (SELECT COUNT("MISSION_OUTCOME")
FROM SPACEXTBL
WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS,
(SELECT COUNT("MISSION_OUTCOME")
FROM SPACEXTBL
WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

[31]:

SUCCESS	FAILURE
100	1

Explanation:

With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

Explanation:

We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[45]: %%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL
      WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

[45]:

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

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[46]:
```
%%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" = 'Failure (drone ship)';
```

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

[46]:

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

Explanation:

This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

[47]:
```
%%sql SELECT
  "Landing_Outcome",
  COUNT("Landing_Outcome") AS Outcome_Count,
  RANK() OVER (ORDER BY COUNT("Landing_Outcome") DESC) AS Rank
FROM
  SPACEXTABLE
WHERE
  Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
  "Landing_Outcome"
ORDER BY
  Outcome_Count DESC;
```

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

[47]:

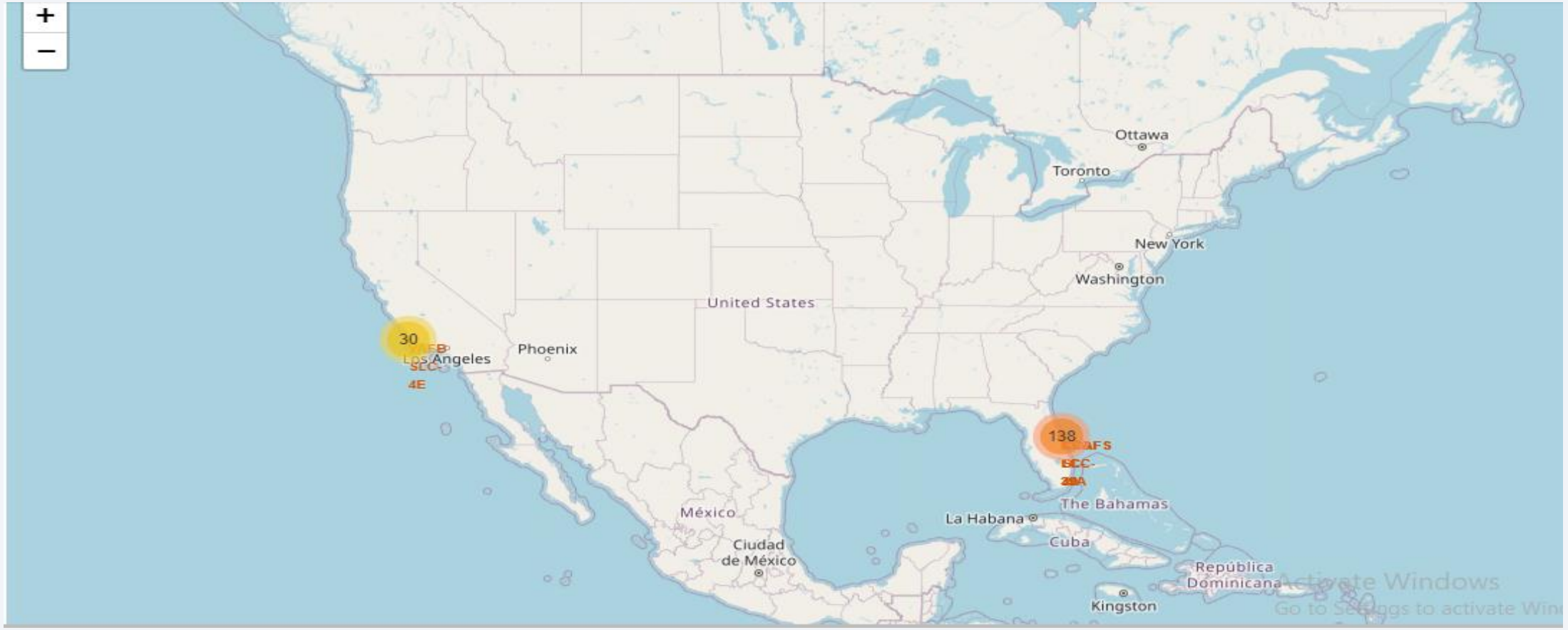
Landing_Outcome	Outcome_Count	Rank
No attempt	10	1
Success (drone ship)	5	2
Failure (drone ship)	5	2
Success (ground pad)	3	4
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Precluded (drone ship)	1	8

Explanation: This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Section 3

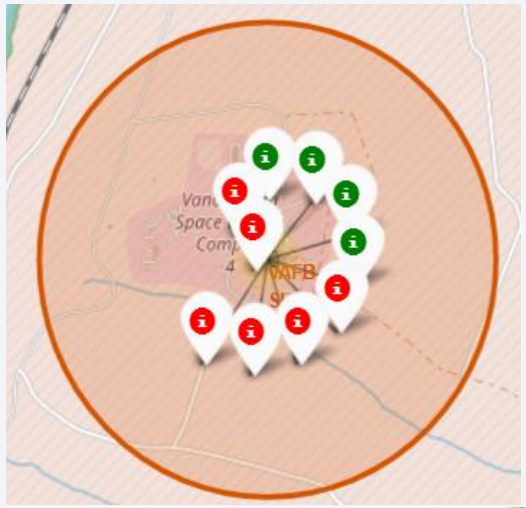
Launch Sites Proximities Analysis

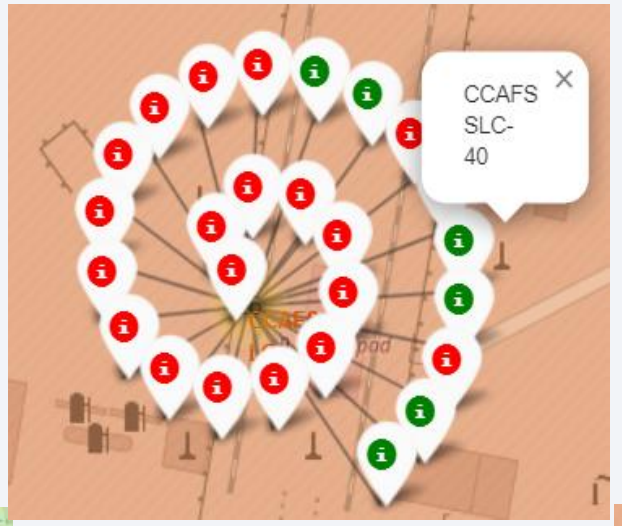
Folium Map – Ground stations

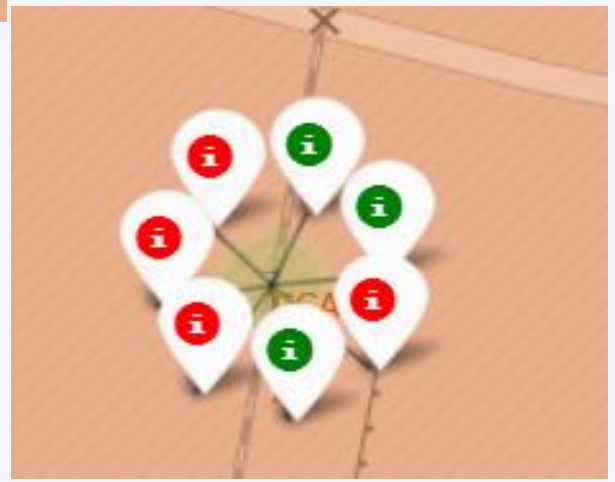


- *From the above, we can clearly see that all of the Space X launch sites are located on the coast of the United States*

Folium map – Color Labeled Markers

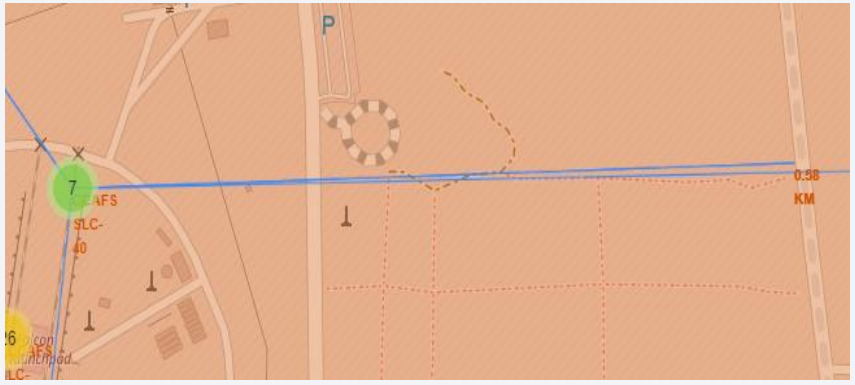


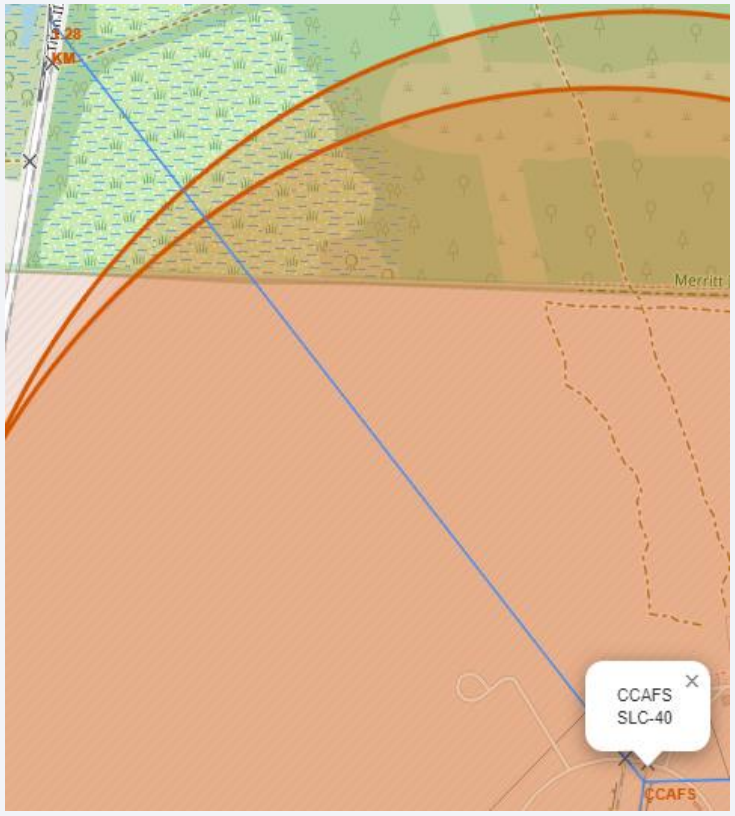


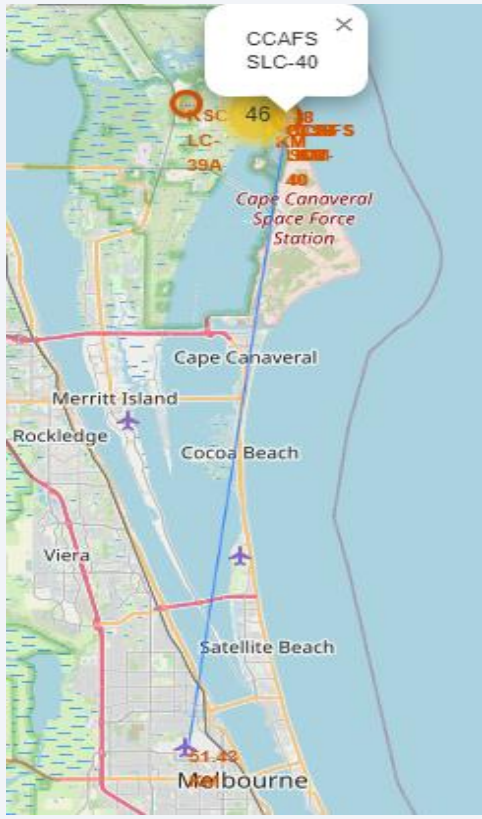


- *Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.*

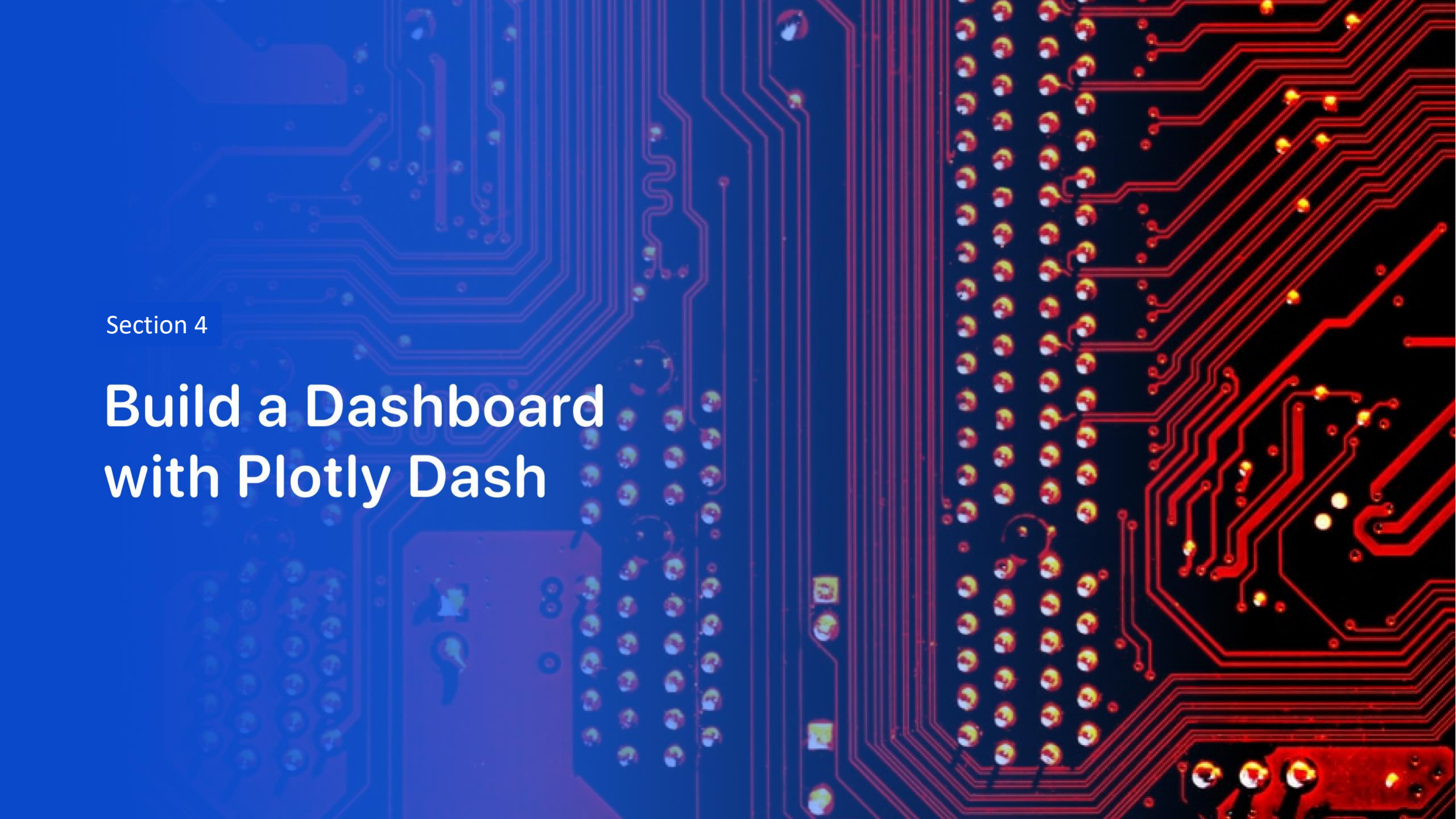
Folium Map – Distances between CCAFS SLC-40 and its proximities



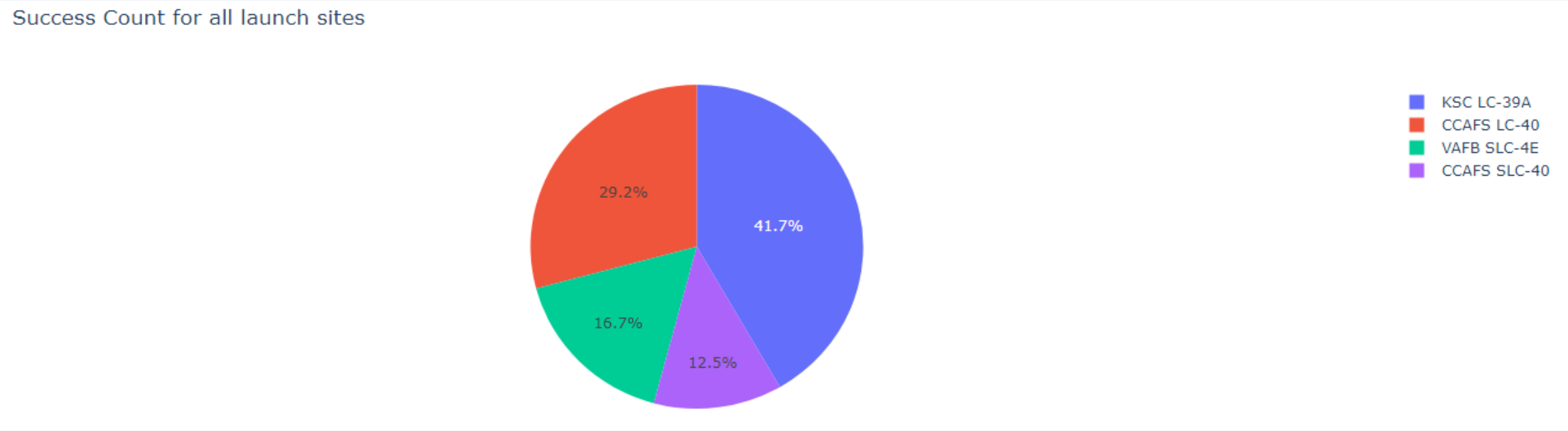




Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

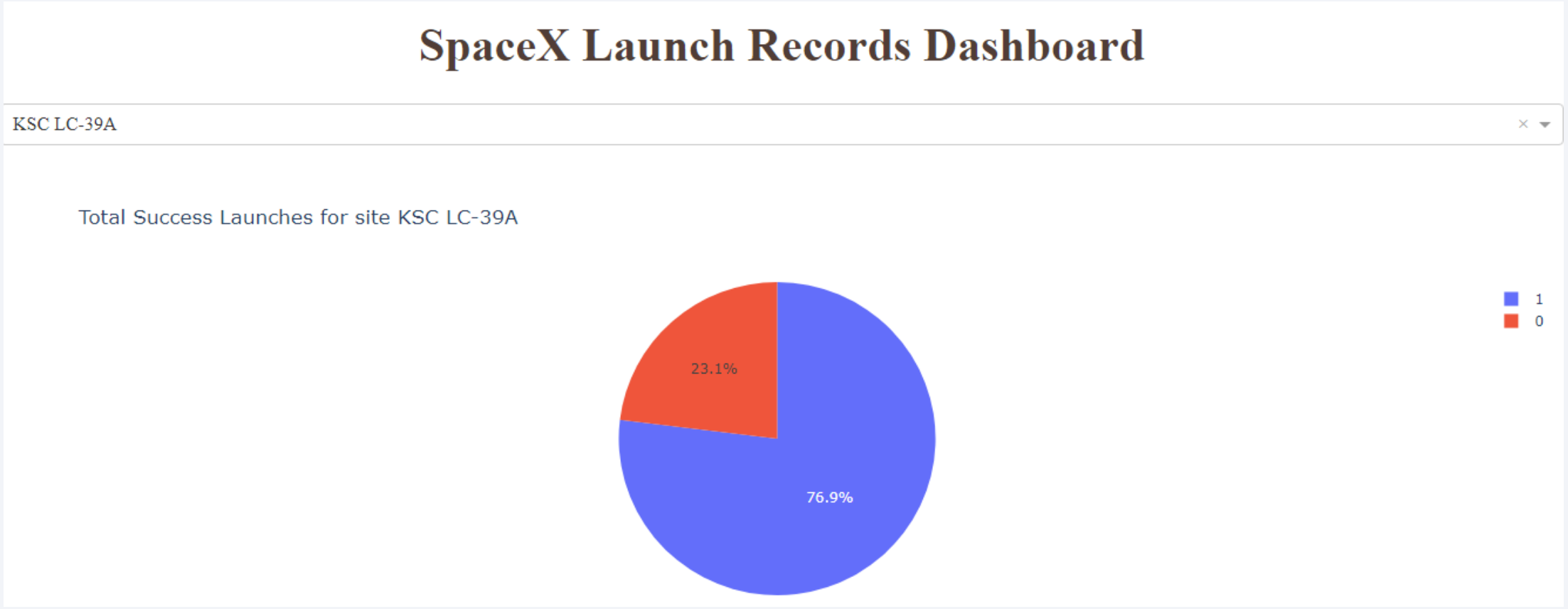

Section 4
Build a Dashboard with Plotly Dash

Dashboard - Total Success for all Sites



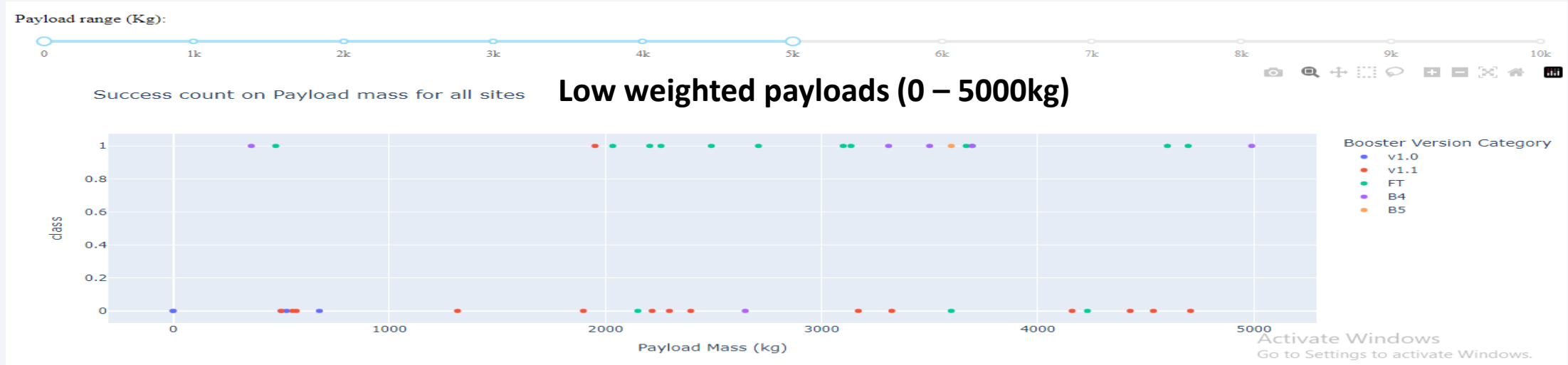
- We see that KSC LC-39A launch site has the highest successful rate of launches

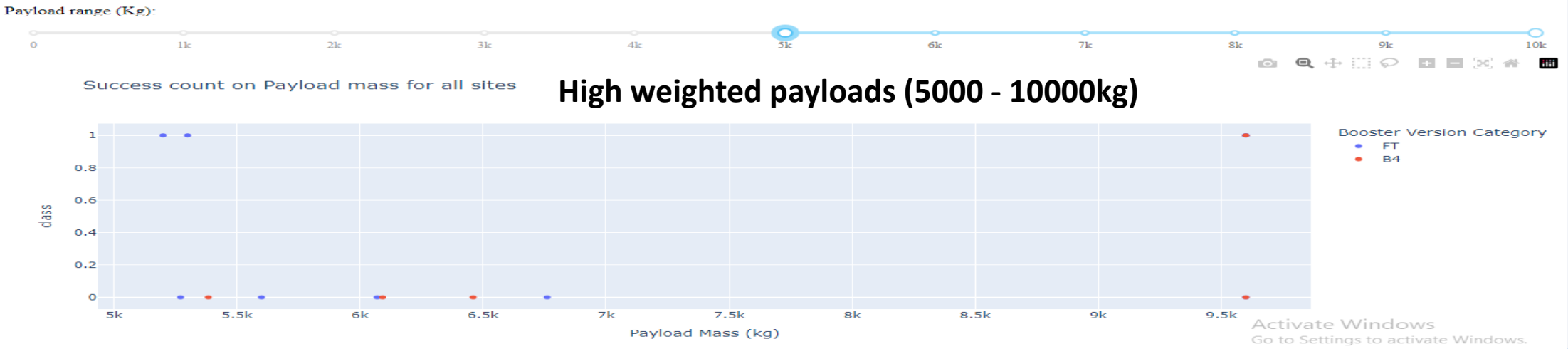
Dashboard - Launch Site With Highest Launch Success Ratio



We see that KSC LC-39A has achieved a 76.9% success rate while a 23.1% failure rate.

Payload vs. Launch Outcome scatter plot for all sites



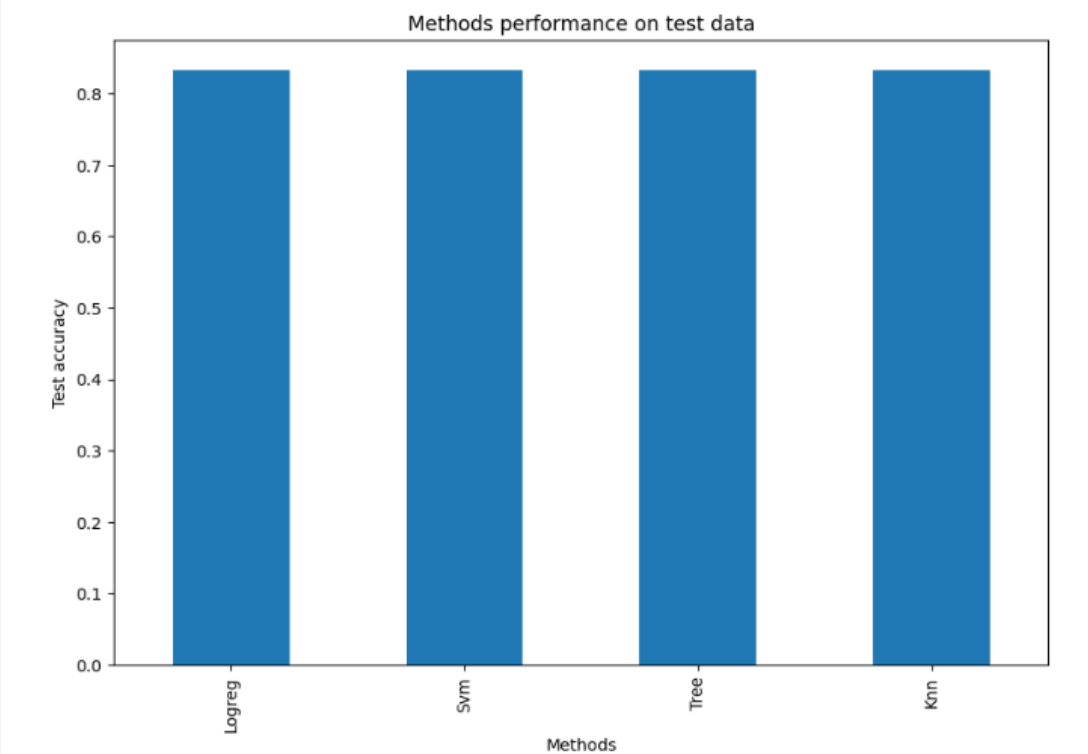


Low weighted payloads have a better success rate than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy



For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree

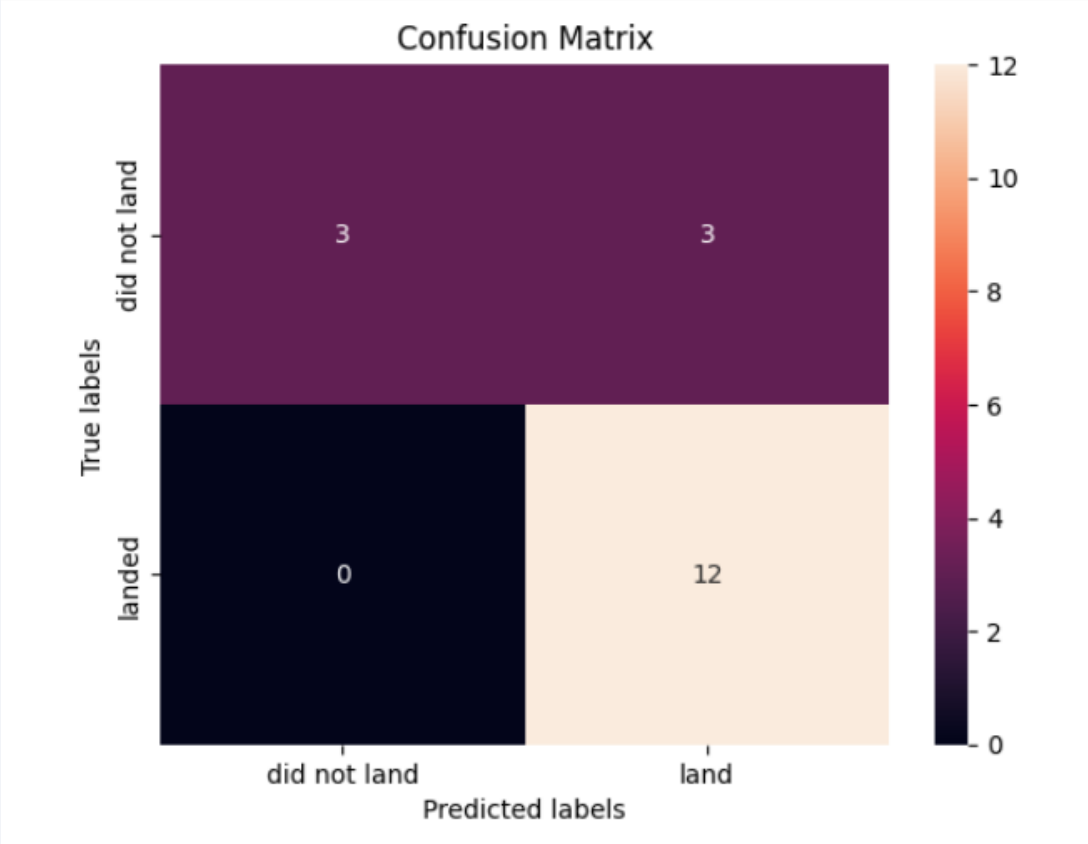
	Accuracy Train	Accuracy Test
Tree	0.875000	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

Confusion Matrix

Explanation:

As the test accuracy of all the models are equal, so the confusion matrices of all the models are also identical. The main problem of these models are false positives.

Decision Tree



Conclusions

- **Mission Success Factors:**
  - *The success of a mission depends on several factors, including the launch site, orbit, and the number of previous launches. Knowledge gained from previous launches contributes to transitioning from failure to success.*
- **Orbit Success Rates:**
  - *Orbits with the highest success rates include GEO (Geostationary Earth Orbit), HEO (Highly Elliptical Orbit), SSO (Sun-Synchronous Orbit), and ES-L1 (Earth-Sun Lagrange Point 1).*
- **Payload Mass Considerations:**
  - *Depending on the orbit, payload mass plays a role in mission success. Some orbits require light payloads, while others can accommodate heavier ones. Generally, lower-weight payloads perform better.*
- **Launch Site Variability:**
  - *The dataset doesn't explain why certain launch sites (e.g., KSC LC-39A) perform better than others. Obtaining additional atmospheric or relevant data could help address this question.*
- **Model Selection:**
  - *Despite identical test accuracies across models, we choose the Decision Tree Algorithm as the best model due to its superior train accuracy.*

Thank you!