IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amoon Austin
December 17, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection with Web Scraping
  - Data Collection through API
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- **Project background and context**
  - SpaceX promotes Falcon 9 rocket launches on its website with a price tag of $62 million, while other providers charge over $165 million per launch. The cost savings largely stem from SpaceX's ability to reuse the first stage. Thus, by predicting whether the first stage will successfully land, we can estimate the cost of the launch. This information could be valuable for other companies looking to compete with SpaceX for rocket launches. The objective of this project is to develop a machine learning model that predicts the successful landing of the first stage.
- **Problems you want to find answers**
  - What factors influence the success of a rocket landing?
    - The relationship between different features that contribute to the likelihood of a successful landing.
    - The operational conditions necessary to ensure a successful landing program.
  - What are the key characteristics that distinguish a successful landing from a failed one?
    - How do the interactions between various rocket variables affect the outcome of the landing?
    - What conditions are required to optimize SpaceX's landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through the SpaceX REST API and web scraping from Wikipedia.
- Perform data wrangling
  - Unnecessary columns were dropped, and one-hot encoding was applied to categorical features for 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
  - Developing, fine-tuning, and evaluating classification models.

Data Collection

- Data collection was performed by making a GET request to the SpaceX API, which provided rocket, launch, and payload information.
  - The API response was decoded as JSON using the .json() function, then converted into a pandas dataframe using .json_normalize().
  - The data was cleaned by checking for and filling in any missing values.

- Additionally, web scraping was conducted from Wikipedia to gather Falcon 9 launch records.
  - Using BeautifulSoup, the launch data was extracted from an HTML table, parsed, and converted into a pandas dataframe for further analysis.
  - The SpaceX API URL used was api.spacexdata.com/v4/, and the Wikipedia page URL was:https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922.

Data Collection – SpaceX API

1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON File

```
data = response.json()
data = pd.json_normalize(data)
```

3. Transform data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

4. Create dictionary with data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Export to file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

1. Getting Response from HTML

```
response = requests.get(static_url)
```

2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables

```
html_tables = soup.findAll('table')
```

4. Get column names

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

See notebook for the rest of code

7. Create dataframe from dictionary

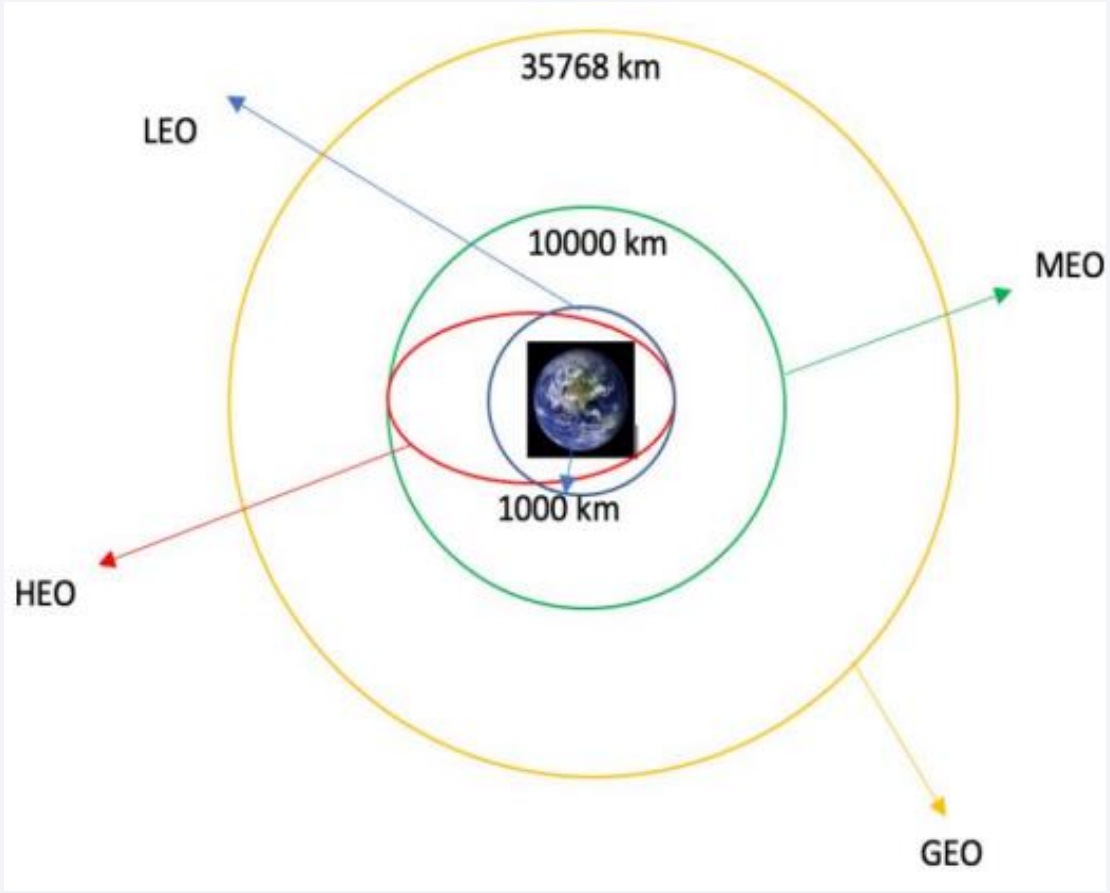
```
df=pd.DataFrame(launch_dict)
```

8. Export to file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

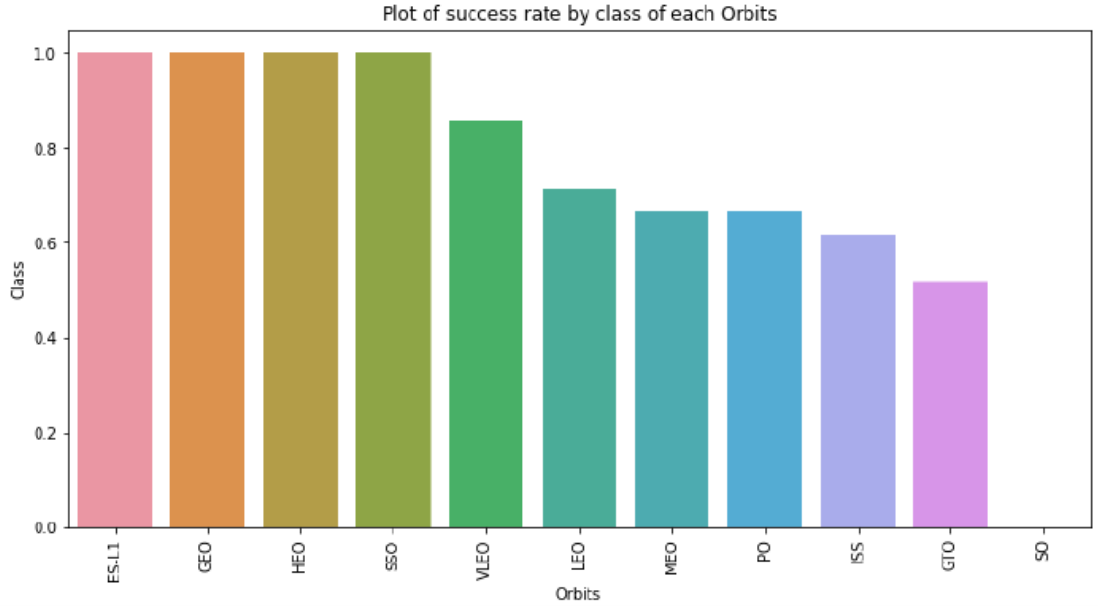
- Exploratory data analysis (EDA) was performed to determine the training labels.
- We analyzed the number of launches at each site and the frequency of each orbit.
- The landing outcome was derived from the outcome column, where "True Ocean," "True RTLS," and "True ASDS" indicated a successful mission, while "False Ocean," "False RTLS," and "False ASDS" signified failure.
- These string variables were transformed into categorical variables, with 1 representing success and 0 representing failure.
- The results were then exported to a CSV file.



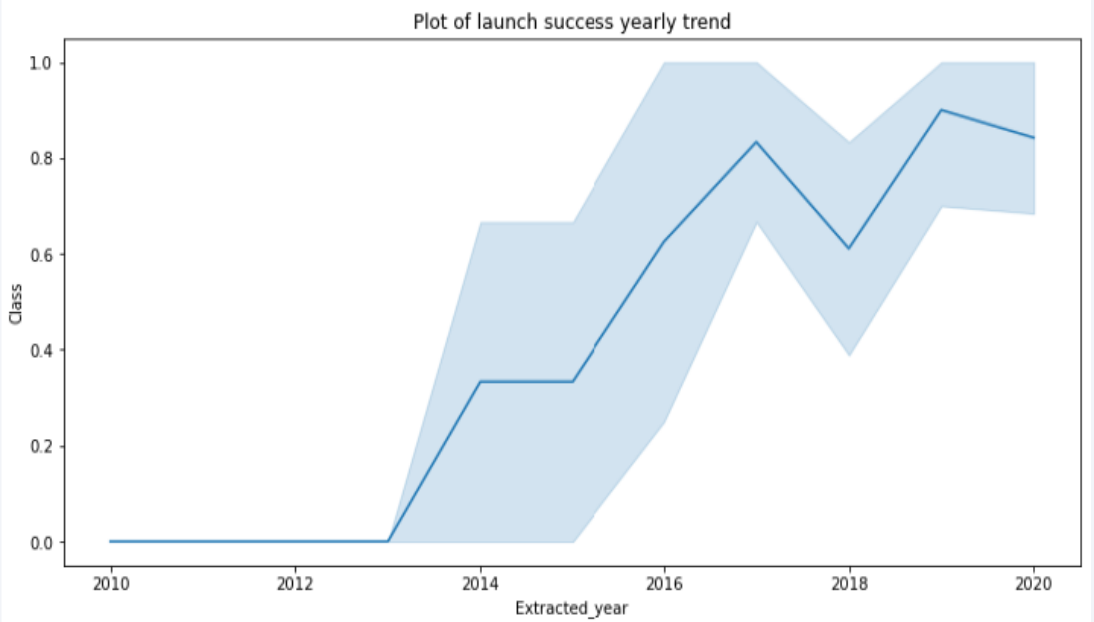
EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

Bar Graph (Success rate vs Orbit)



Line Graph (Success rate vs Year)



EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database directly within the Jupyter notebook and performed exploratory data analysis (EDA) using SQL queries to gain insights. Our SQL queries aimed to:
  - Display the unique launch sites in the space mission.
  - Display five records where launch sites begin with 'CCA'.
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display the average payload mass carried by booster version F9 v1.1.
  - List the date of the first successful landing outcome on a ground pad.
  - List the boosters that succeeded in drone ship landings with payload masses between 4000 and 6000.
  - List the total number of successful and failed mission outcomes.
  - List the booster versions that carried the maximum payload mass.
  - Display records of failure landing outcomes on drone ships in 2015, including month names, booster versions, and launch sites.
  - Rank the count of successful landing outcomes between April 6, 2010, and March 20, 2017, in descending order.

Build an Interactive Map with Folium

- We used Folium to create an interactive map centered on NASA Johnson Space Center in Houston, Texas. The map included various markers and map objects to visualize the launch sites and their success or failure rates. Specifically:
  - Red circles were placed at NASA Johnson Space Center and at each launch site's coordinates, labeled with their respective names.
  - Markers were color-coded to indicate landing outcomes: green for successful landings (class 1) and red for unsuccessful landings (class 0).We used marker clusters to group points and display different information for the same coordinates.
  - We calculated the distances from launch sites to nearby railways, highways, coastlines, and cities, plotting lines to represent these distances.
  - This mapping approach helped us identify which launch sites had relatively high success rates and understand their proximity to key locations, enhancing our analysis of the data.

Build a Dashboard with Plotly Dash

- We built an interactive dashboard using Plotly Dash, which includes several key components for data visualization and user interaction:
  - A dropdown menu allows users to select a specific launch site or view data for all launch sites.
  - A pie chart displays the total number of successful and failed launches for the selected site.
  - A range slider enables users to select a payload mass range.
  - A scatter plot shows the relationship between the landing outcome (success or failure) and the payload mass (Kg) for different booster versions.
  - The interactive elements help users explore and analyze the data more effectively.

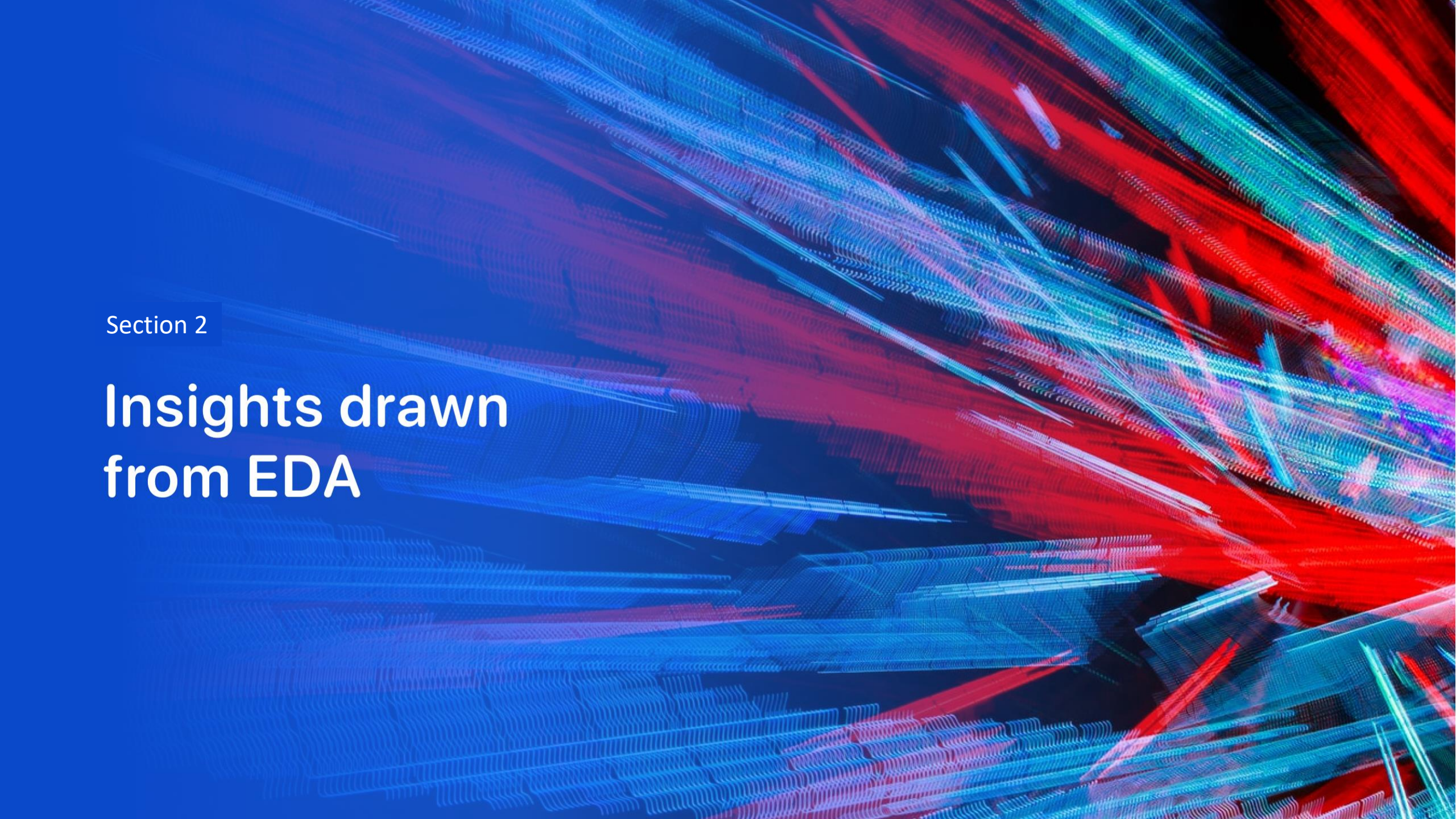
Link already shared for review.

Predictive Analysis (Classification)

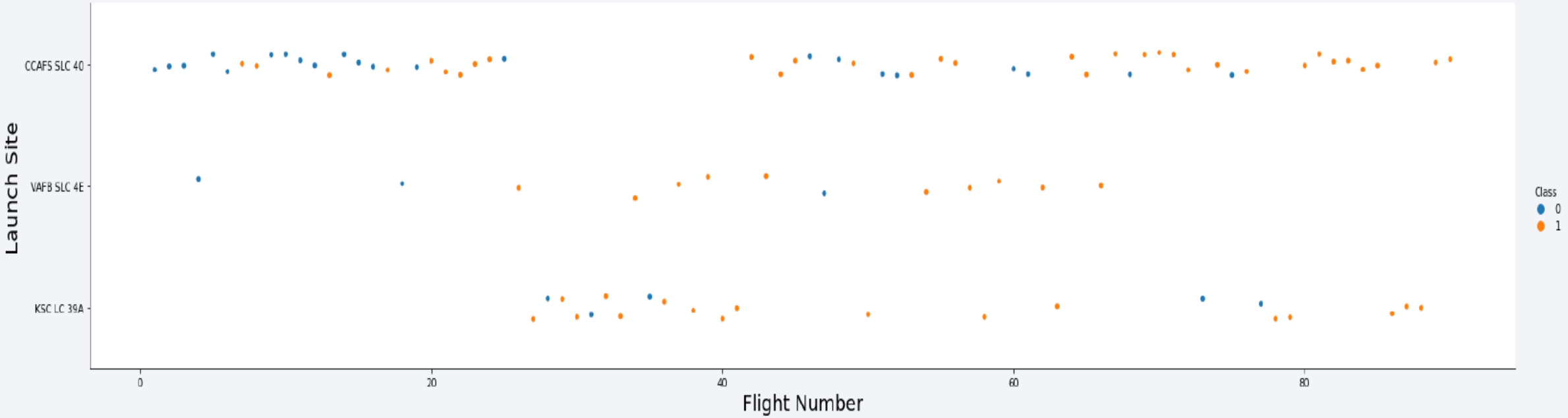
- Data Preparation:
  - Loaded the dataset using numpy and pandas.
  - Normalized the data.
  - Split the data into training and test sets.
- Model Preparation:
  - Selected various machine learning algorithms.
  - Set parameters for each algorithm using GridSearchCV.
  - Trained GridSearchCV models with the training dataset.
- Model Evaluation:
  - Identified the best hyperparameters for each model.
  - Computed the accuracy for each model using the test dataset.
  - Plotted a confusion matrix to visualize model performance.
- Model Comparison:
  - Compared models based on their accuracy.
  - Chose the model with the best accuracy.

Results

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

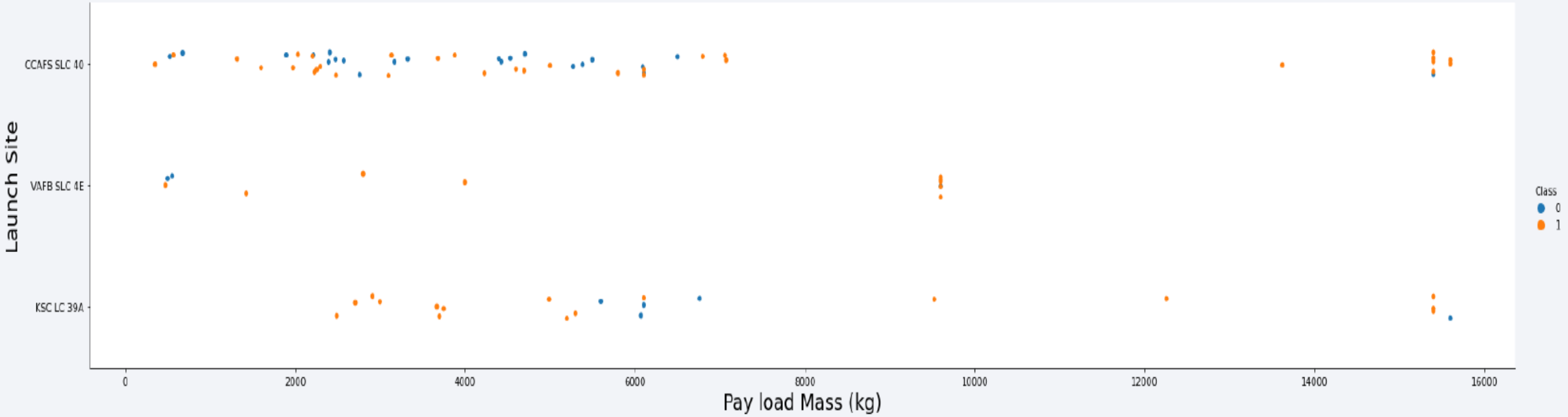

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



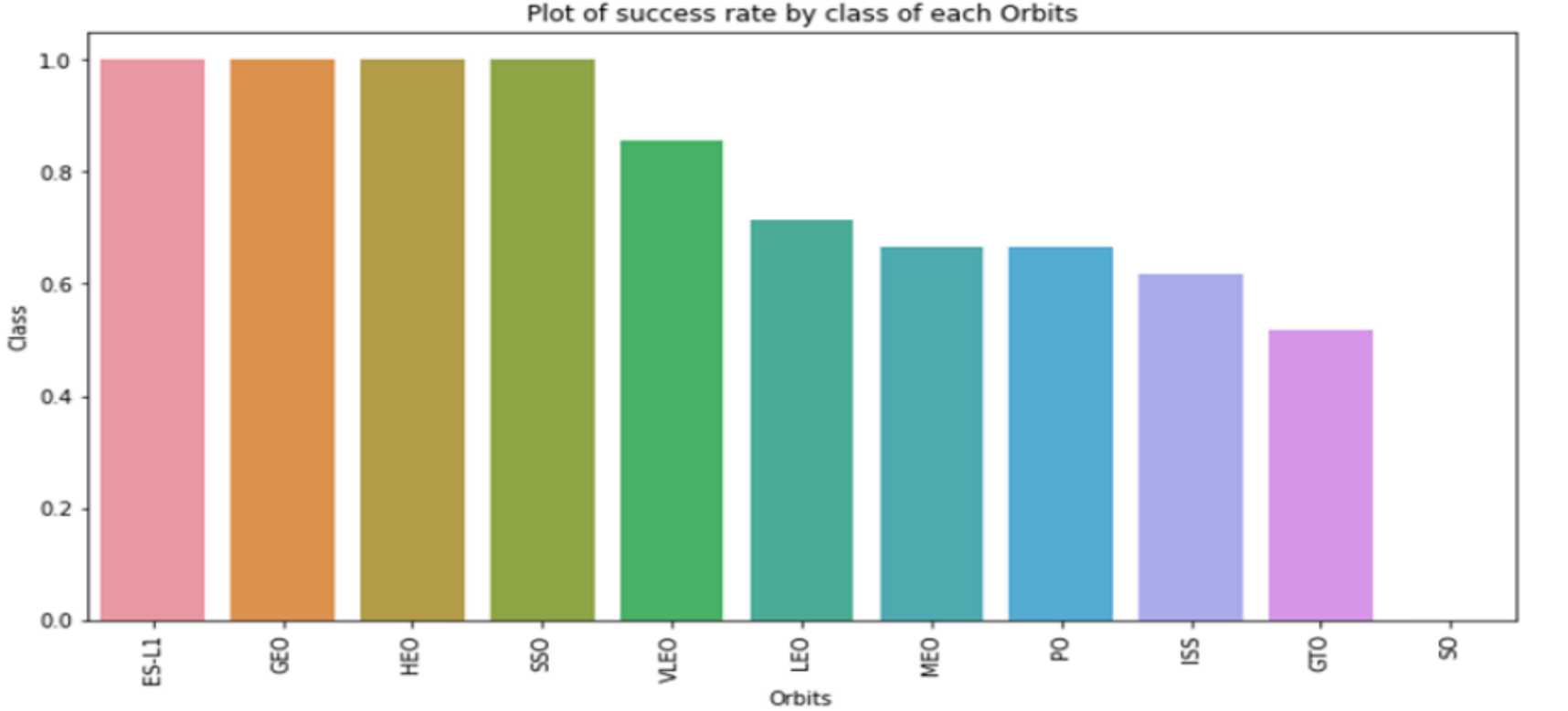
From the plots, we observed that the success rate at a launch site increases with the number of flights conducted at that site.

Payload vs. Launch Site



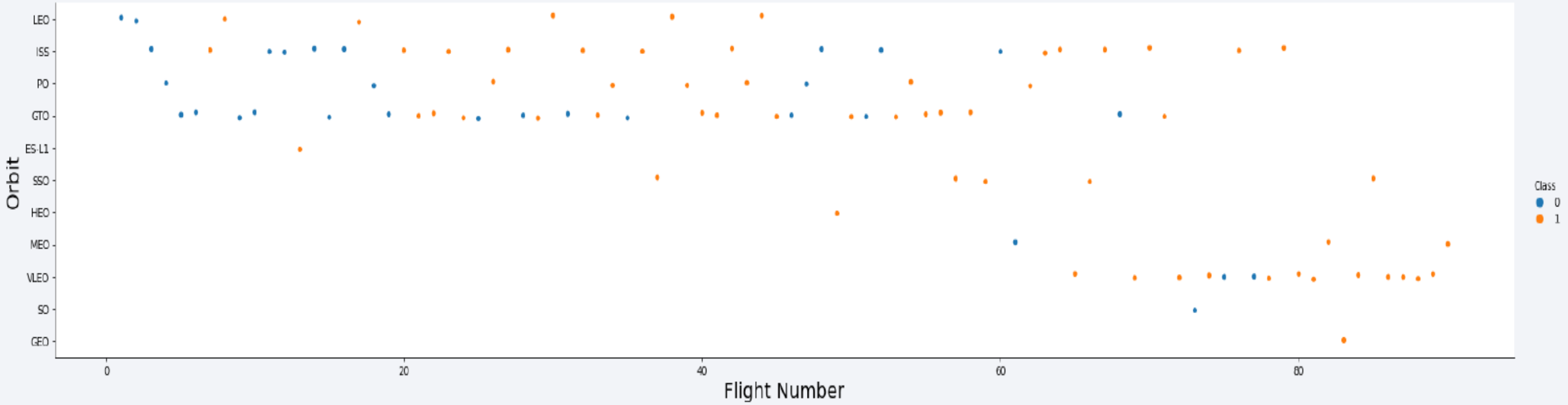
The success rate of a landing can depend on the payload mass. For example, at launch site CCAFS SLC 40, a greater payload mass tends to correlate with a higher success rate. However, if the payload is too heavy, it can lead to a failed landing.

Success Rate vs. Orbit Type



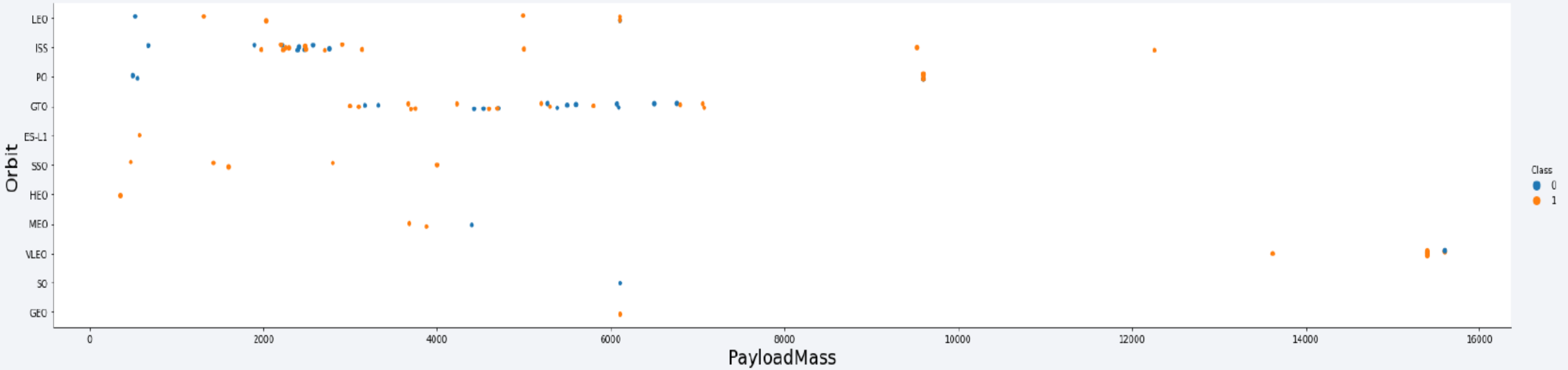
From the plot, we can observe that the orbit types ES L1, GEO, HEO, and SSO have the highest success rates.

Flight Number vs. Orbit Type



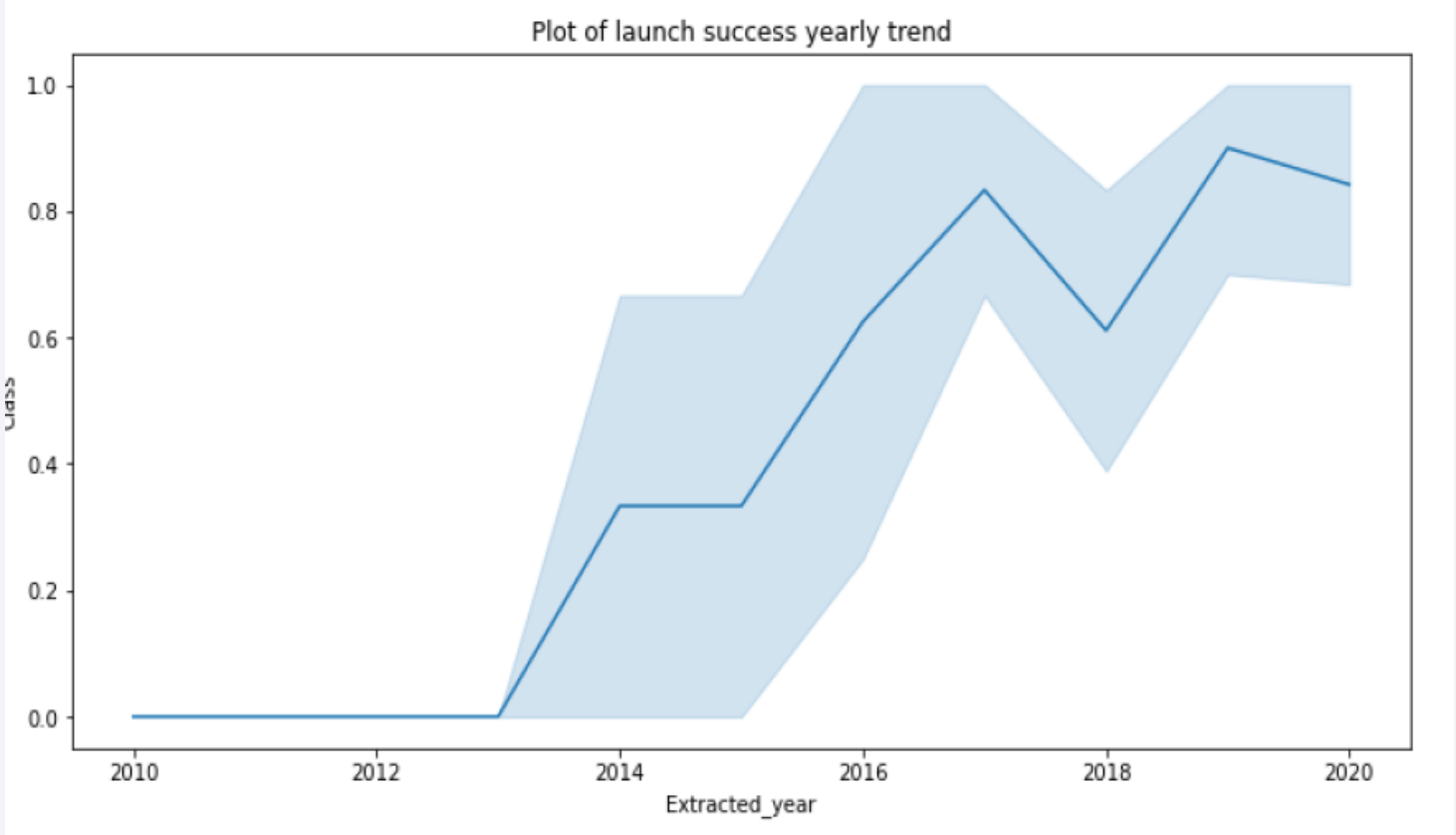
The plot shows the relationship between Flight Number and Orbit Type. We observe that for the LEO orbit, the success rate increases with the number of flights. However, for orbits like GTO, there is no clear relationship between the success rate and the number of flights. It can be inferred that the high success rates of orbits like SSO and HEO may be influenced by the knowledge gained from previous launches in other orbit types.

Payload vs. Orbit Type



The payload weight significantly impacts the success rate of launches in certain orbits. Heavier payloads tend to improve the success rate for LEO, PO, and ISS orbits. Conversely, for GTO orbits, reducing the payload weight can enhance the likelihood of a successful launch.

Launch Success Yearly Trend



Since 2013, we have observed a continuous increase in the success rate of SpaceX rocket launches, reaching its peak by 2020.

All Launch Site Names

- We used the keyword **DISTINCT** in the query to filter out duplicate launch sites and display only unique ones from the SpaceX data.

```
[10]: # Execute the SQL query to get unique launch sites
      result = %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;

      # Display the result
      result
```

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

[10]:

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

Launch Site Names Begin with 'CCA'

- We used the query with a **WHERE** clause followed by a **LIKE** clause to filter launch sites containing the substring "CCA." The **LIMIT 5** ensures that only 5 records are displayed after the filtering.

[11]:
```
# SQL query to select 5 records where Launch_Site begins with 'CCA'
result_task2 = %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;

# Display the result
result_task2
```

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

[11]:

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

- We used the query to calculate the total payload mass carried by boosters for NASA (CRS), which amounted to **45,596**.

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

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- We used the query to calculate the average payload mass carried by booster version F9 v1.1, which was **2,928.4**.

[13]:
```
# SQL query to calculate the average payload mass carried by booster version F9 v1.1
result_task4 = %sql SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';

# Display the result
result_task4
```

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

[13]:

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

- Using this query, we identified the oldest successful landing, which occurred on December 22, 2015. The **WHERE** clause filters the dataset to include only successful landings, and the **MIN** function selects the record with the earliest date.

[14]:
```
# SQL query to find the date of the first successful landing outcome on a ground pad
result_task5 = %sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';

# Display the result
result_task5
```

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

[14]:

First_Successful_Landing_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** and **AND** clauses to filter the dataset for boosters that successfully landed on the drone ship, with a payload mass between 4,000 and 6,000 kg.

[15]:
```
# SQL query to list the names of the boosters with successful drone ship landing and payload mass between 4000 and 6000 kg
result_task6 = %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass__kg_" > 4000 AND "Payload_Mass__kg_" < 6000;

# Display the result
result_task6
```

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

[15]:

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 the **WHERE** clause with the **LIKE** operator and a wildcard ('%') to filter mission outcomes as either a success or failure. The query includes subqueries to count successful and unsuccessful missions, with the COUNT function tallying the filtered records.

In [16]:
```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Boosters Carried Maximum Payload

- We used a subquery with the **SELECT** function in the **WHERE** clause to filter for the booster that carried the heaviest payload. The main query then returns the unique booster version that carried this maximum payload.

```
[17]: # SQL query to list the names of the booster versions which have carried the maximum payload mass using a subquery
result_task8 = %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE);

# Display the result
result_task8
```

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

[17]:

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 combination of the **WHERE, LIKE, AND**, and **BETWEEN** conditions to filter for failed landing outcomes on the drone ship, including booster versions, launch site names, and landing dates from the year 2015.

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [18]:
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We selected landing outcomes and counted their occurrences from the data, filtering for landing outcomes between **2010-06-04** and **2017-03-20** using the **WHERE** clause.
- We then used the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to arrange them in descending order.

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

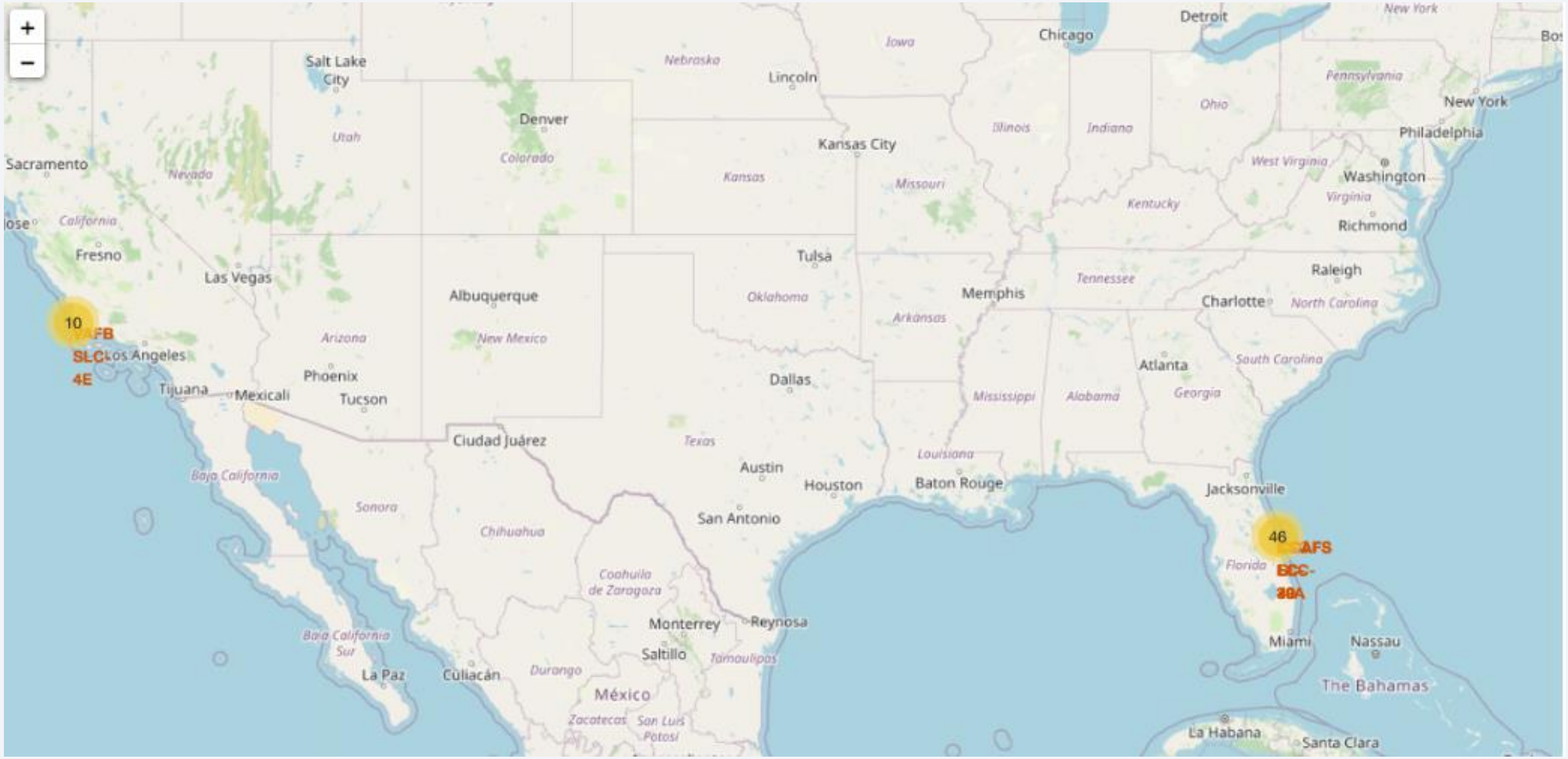
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

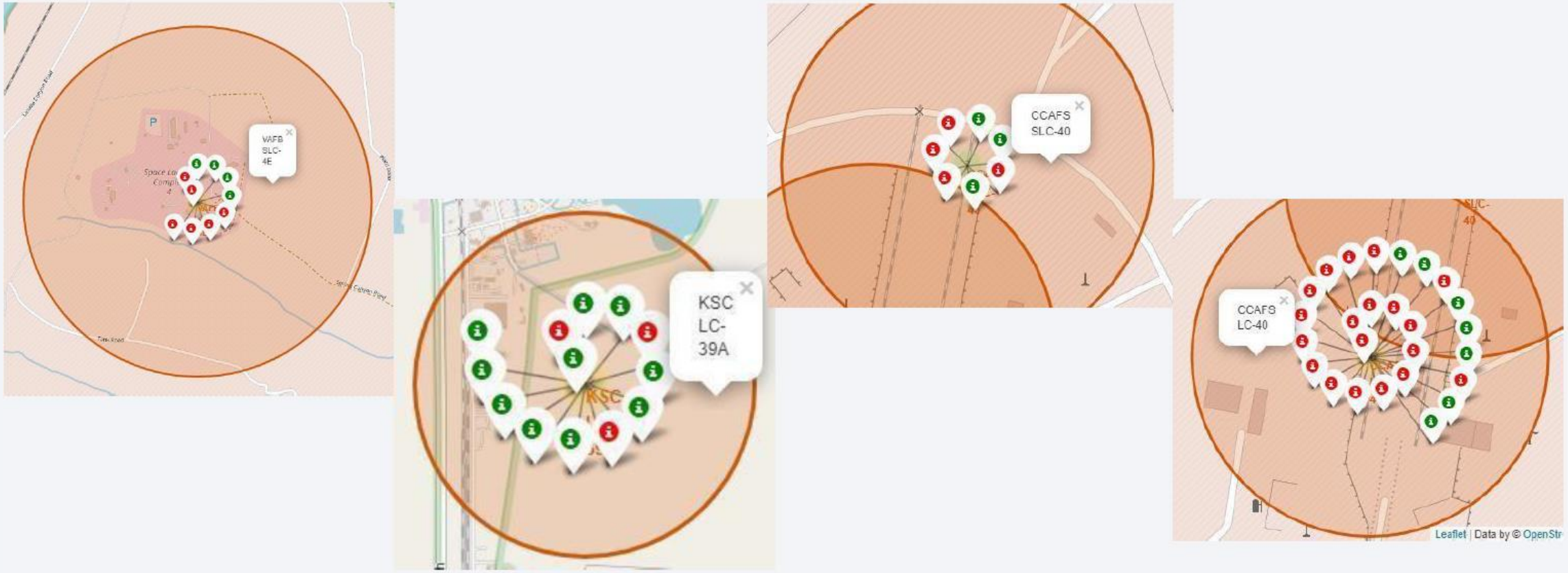
Folium Map – Ground stations

- We observe that SpaceX launch sites are situated along the coast of the United States.

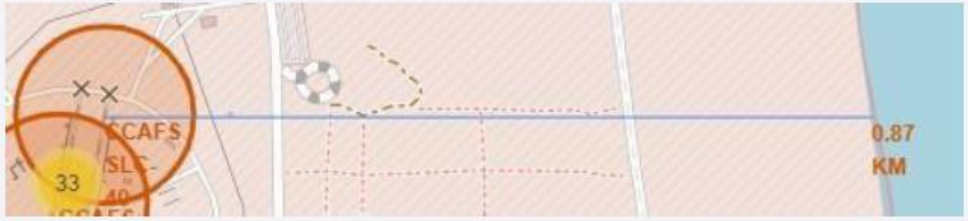


Folium Map – Color Labeled Markers

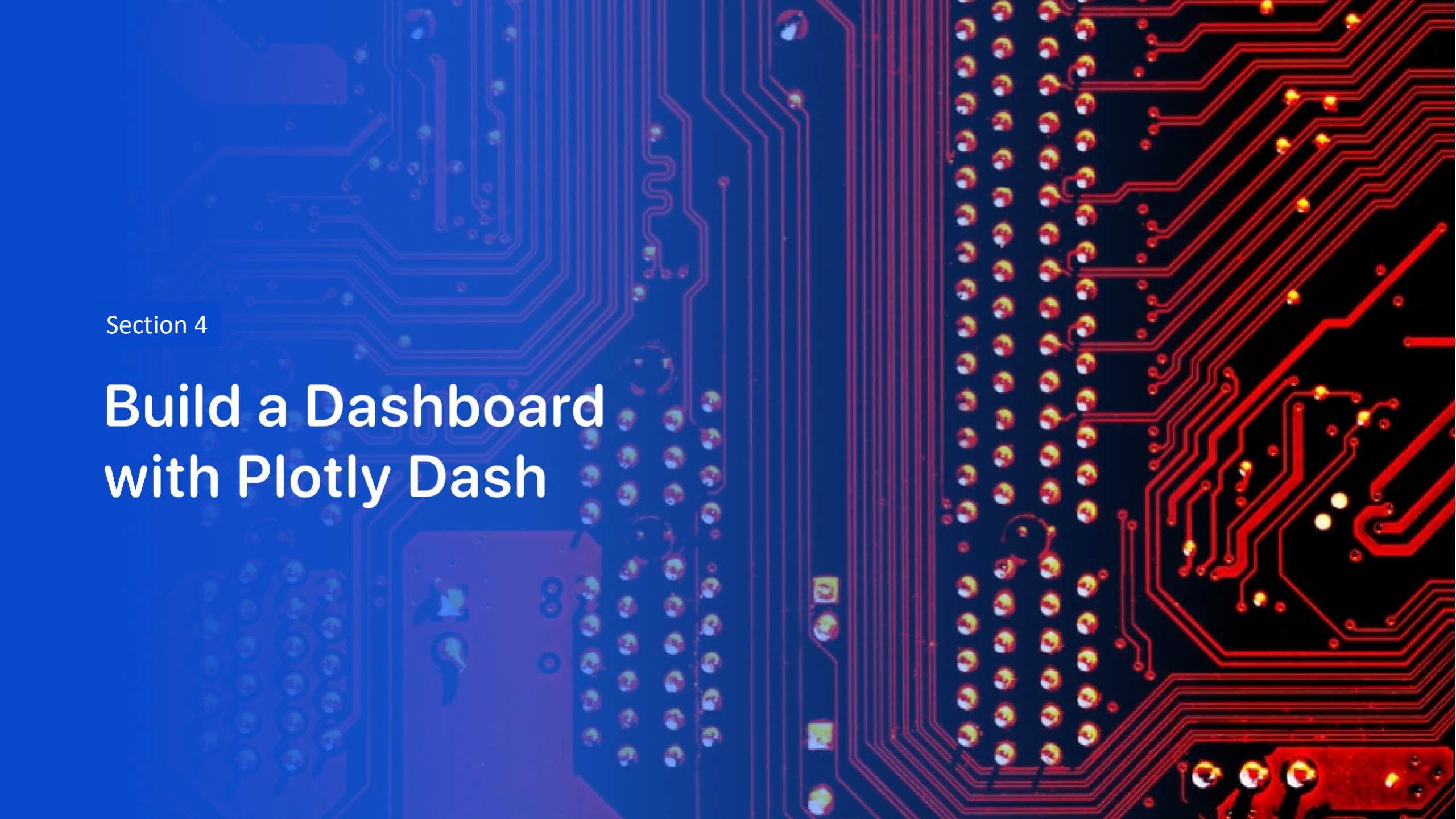
- The green marker indicates successful launches, while the red marker represents unsuccessful launches.



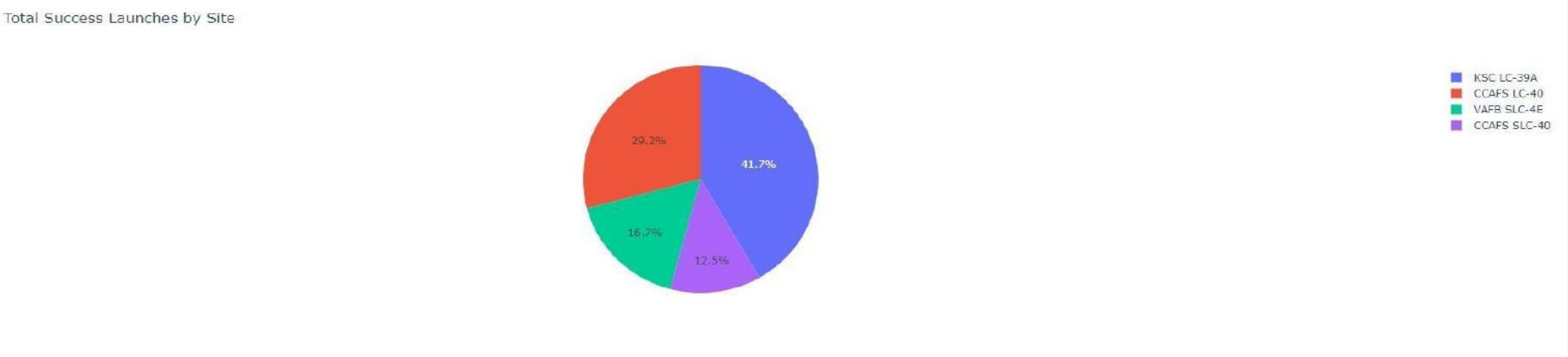
Folium Map – Distances between CCAFS SLC-40 and its proximities



- 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

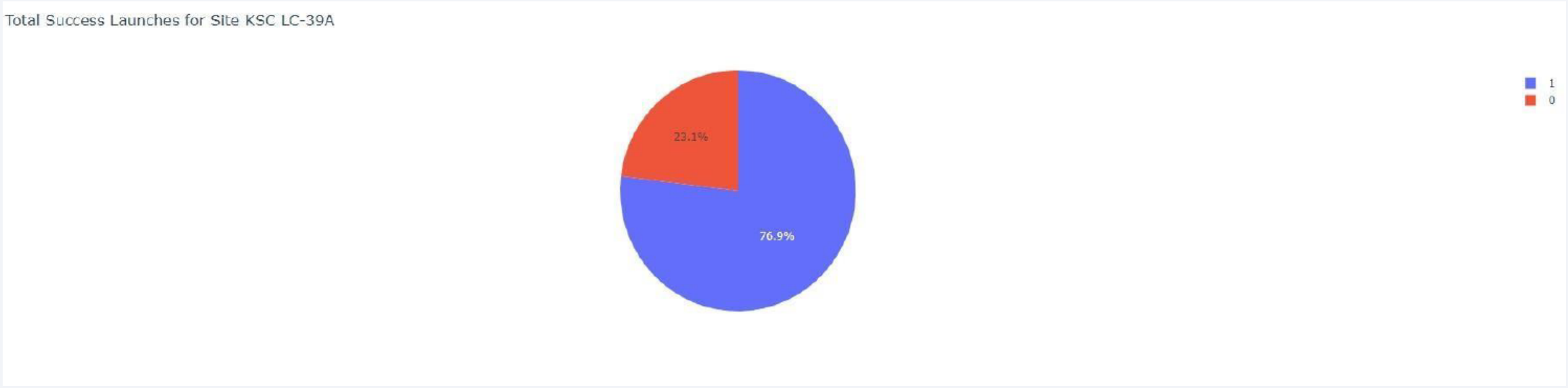

Section 4
Build a Dashboard with Plotly Dash

Success Rate of Launches by Site



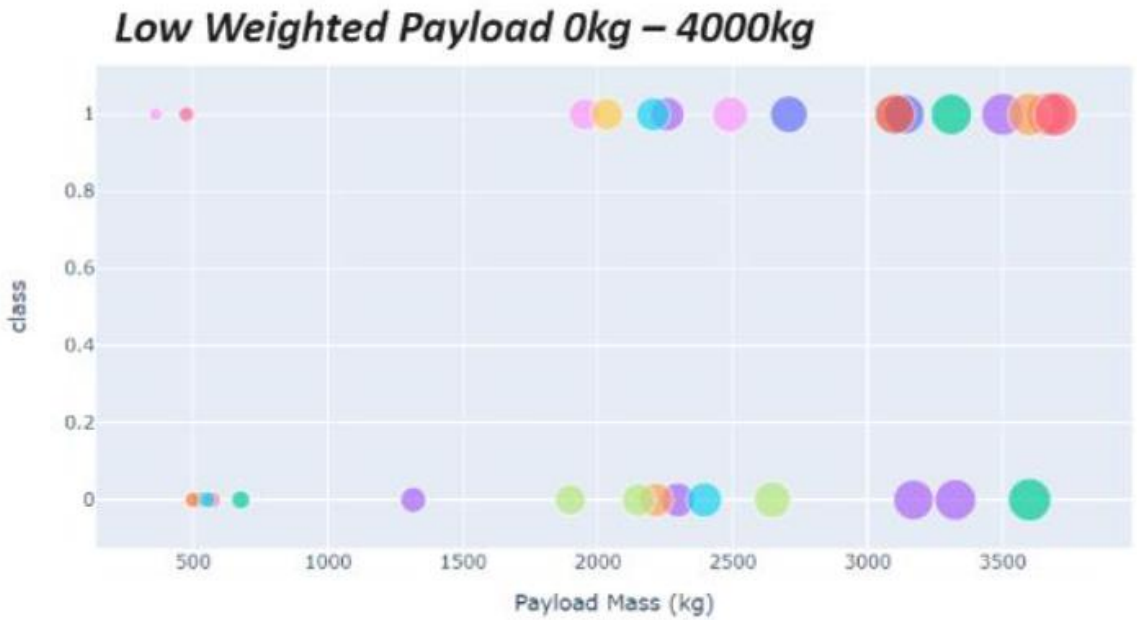
KSC LC-39A has the best success rate of launches, followed by CCAFS LC-40.

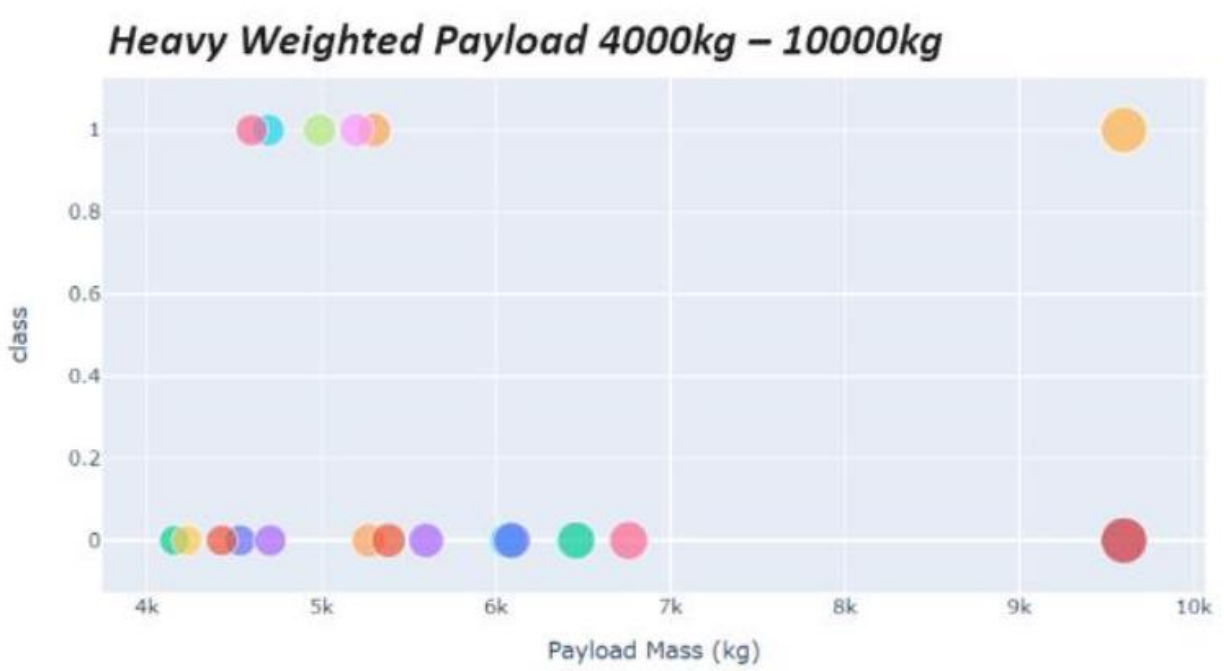
Launch Site with the Highest Launch Success Ratio



We observe that KSC LC 39 A has a success rate of 76.9%, with a failure rate of 23.1%.

Scatter Plot of Payload vs. Launch Outcome for All Sites, with Payload Range Selected in the Slider





Payloads with lower weights tend to have a higher success rate compared to heavier payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest accuracy for classification.

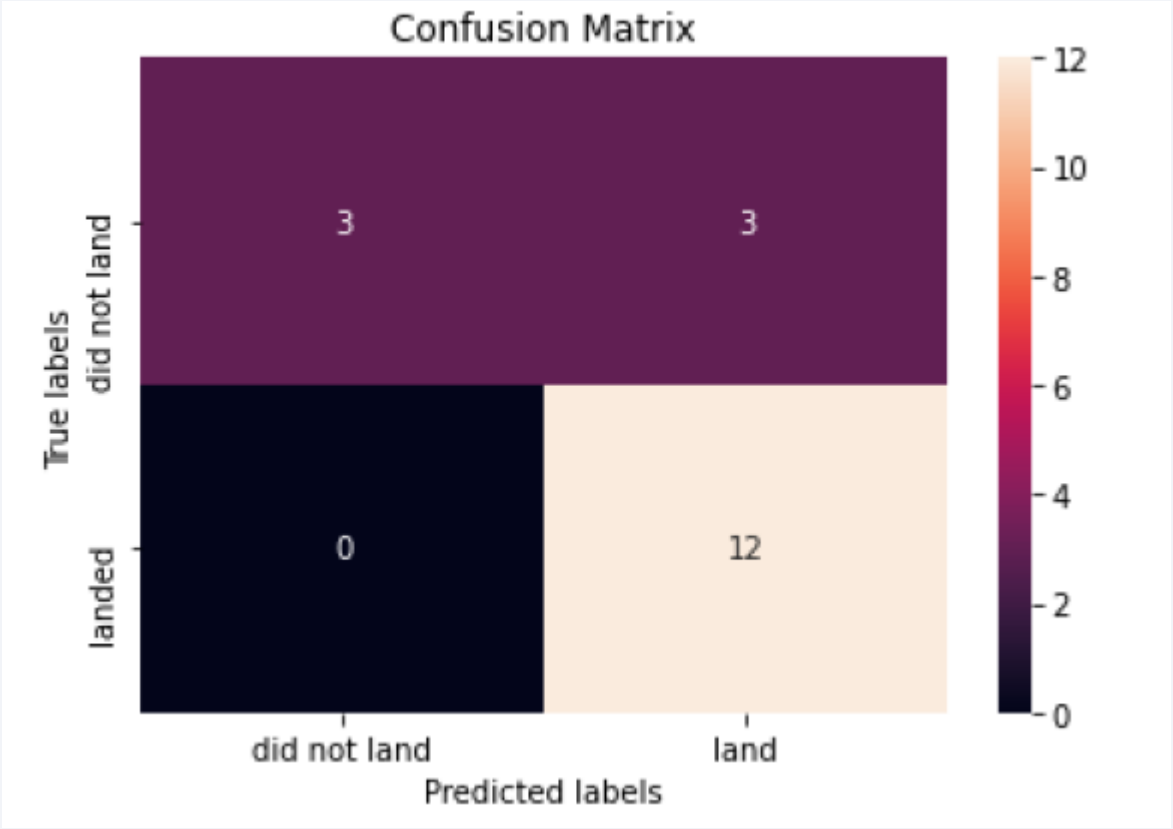
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier indicates that the model can differentiate between the various classes. However, a significant issue is the occurrence of false positives, where unsuccessful landings are incorrectly classified as successful.



Conclusions

- The success of a mission can be attributed to various factors, including the launch site, the orbit, and notably the number of previous launches, suggesting that knowledge gained between launches contributes to increased success rates. Key observations include:
  - The orbits with the highest success rates are GEO, HEO, SSO, and ES L1.
  - Payload mass significantly influences mission success, with lighter payloads generally performing better than heavier ones.
  - The number of flights at a launch site correlates with a higher success rate, as seen with KSC LC 39A, which has the highest success rate among launch sites.
  - The overall launch success rate has been increasing since 2013.
  - The decision tree classifier was selected as the best machine learning model for this task due to its superior training accuracy, even though the test accuracy across all models was similar.
- However, the current data does not explain why some launch sites perform better than others. Further investigation with additional data, such as atmospheric conditions, might be required to understand these differences.

Thank you!