IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MohammadReza Mirafzal
June 8th 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard with Plotly Dash
  - Predictive analysis

- Summary of all results
  - EDA results
  - Interactive analysis
  - Predictive analysis

Introduction

- Project background and context
  - SpaceX offers Falcon 9 rocket launches at a competitive price of $62 million, in contrast to other providers whose launches can cost over $165 million. A significant portion of these cost savings stems from SpaceX's ability to reuse the first stage of the rocket.
- Problems you want to find answers
  - The primary objective of this project is to predict whether the first stage of the SpaceX Falcon 9 rocket will successfully land.

Section 1

Methodology

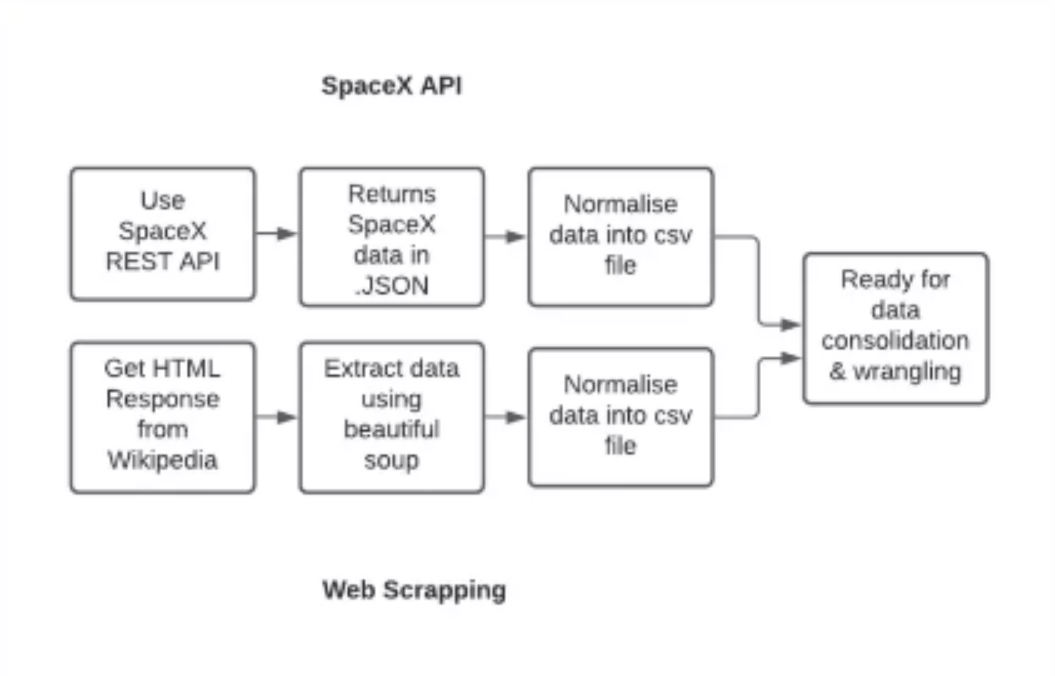
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Data scraping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for ML and data cleaning of null values and irrelevant columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Four different ML classification model have been built and evaluated comparatively.

Data Collection

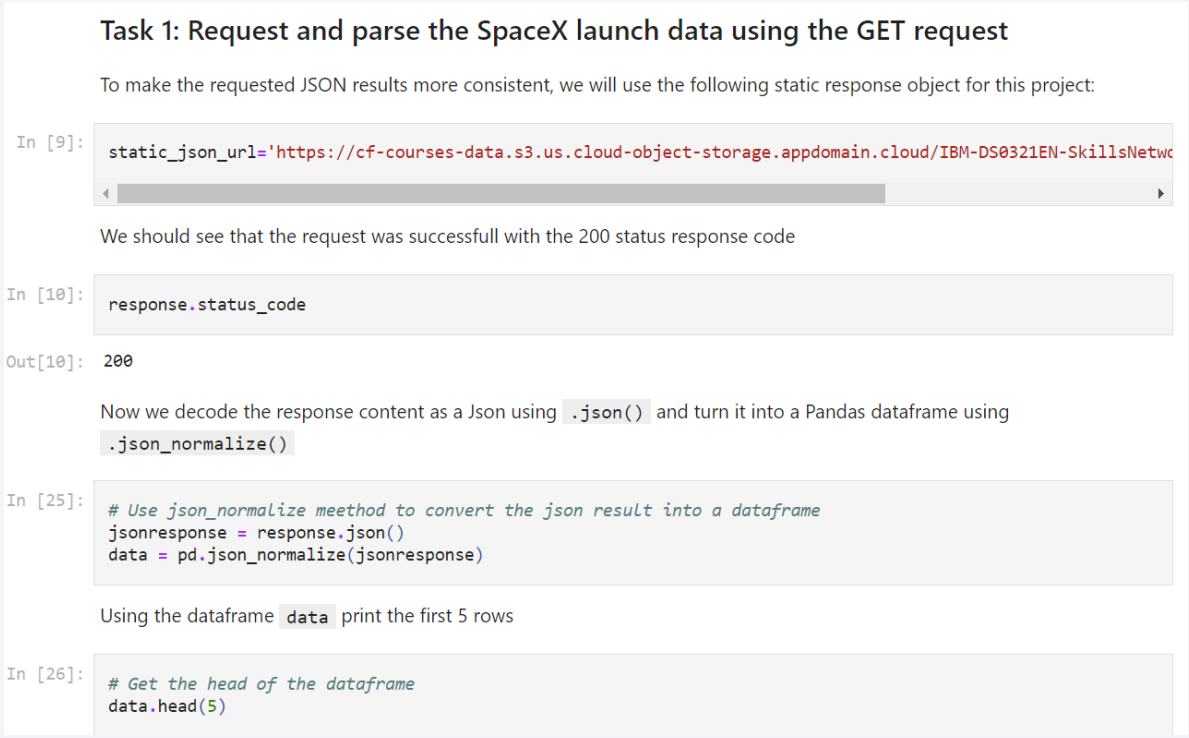
- **SpaceX Launch Data**: Collected via the SpaceX REST API, which provides detailed information on rocket launches, including the type of rocket utilized, payload details, launch parameters, landing specifications, and the final outcome of each launch.
- **API Details**: The SpaceX REST API can be accessed through the endpoint https://api.spacexdata.com/v4/.
- **Wikipedia Data Collection**: Utilized web scraping techniques on Wikipedia to gather additional Falcon 9 launch data, employing the BeautifulSoup library for parsing the HTML content.



Data Collection – SpaceX API

- We collected data via a GET request to the SpaceX API, then proceeded to clean, format, and perform basic data wrangling on the acquired information.
- Here is the link to the notebook: https://github.com/mamarexa/ibm_data_science_project/blob/main/01_SpaceX_Data_Collection_API.ipynb

Hey there! Just a heads up, you'll need to do a bit of scrolling through the Python code on GitHub to hit the juicy bits. Sorry about the novel-length text in there—I'm not entirely sure what inspired its epic length!



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwo
```

We should see that the request was successfull with the 200 status response code

In [10]:
```
response.status_code
```

Out[10]:
```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [25]:
```
# Use json_normalize meethod to convert the json result into a dataframe
jsonresponse = response.json()
data = pd.json_normalize(jsonresponse)
```

Using the dataframe `data` print the first 5 rows

In [26]:
```
# Get the head of the dataframe
data.head(5)
```

Data Collection - Scraping

- Web scraping was utilized to extract Falcon 9 launch records using BeautifulSoup. The data from the table was parsed and transformed into a pandas DataFrame.
- Here is the link to the notebook: https://github.com/mamarexa/ibm_data_science_project/blob/main/02_SpaceX_Web_Scraping.ipynb

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

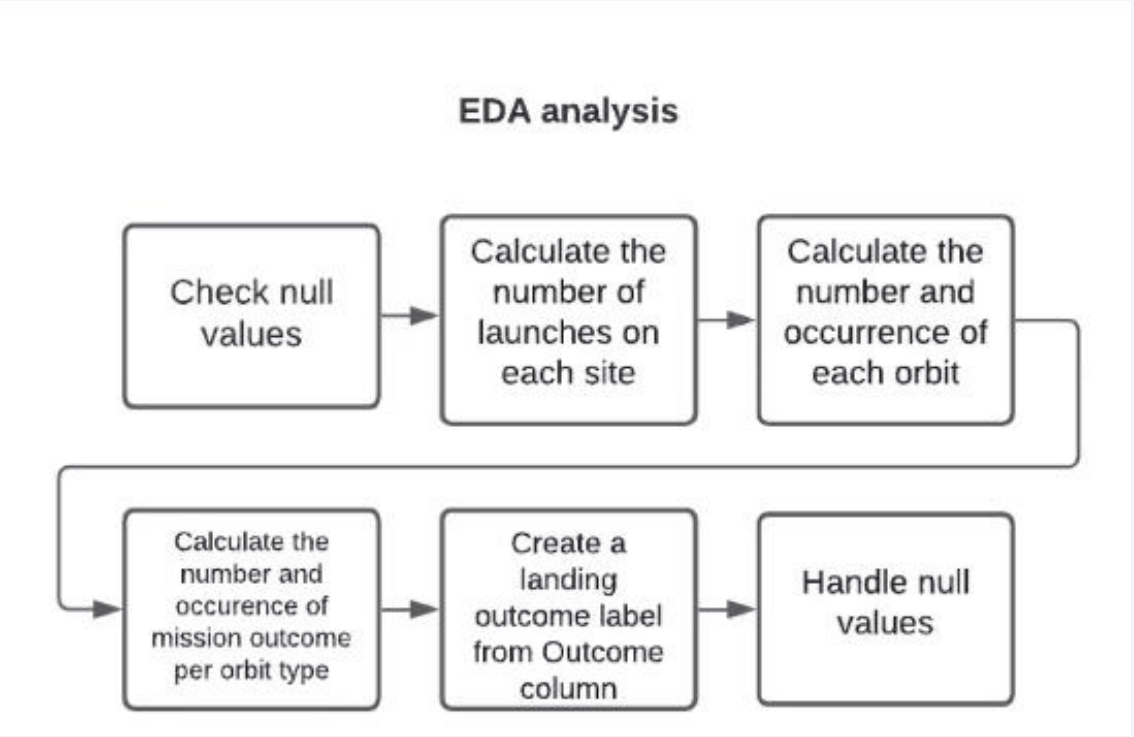
```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Data Wrangling

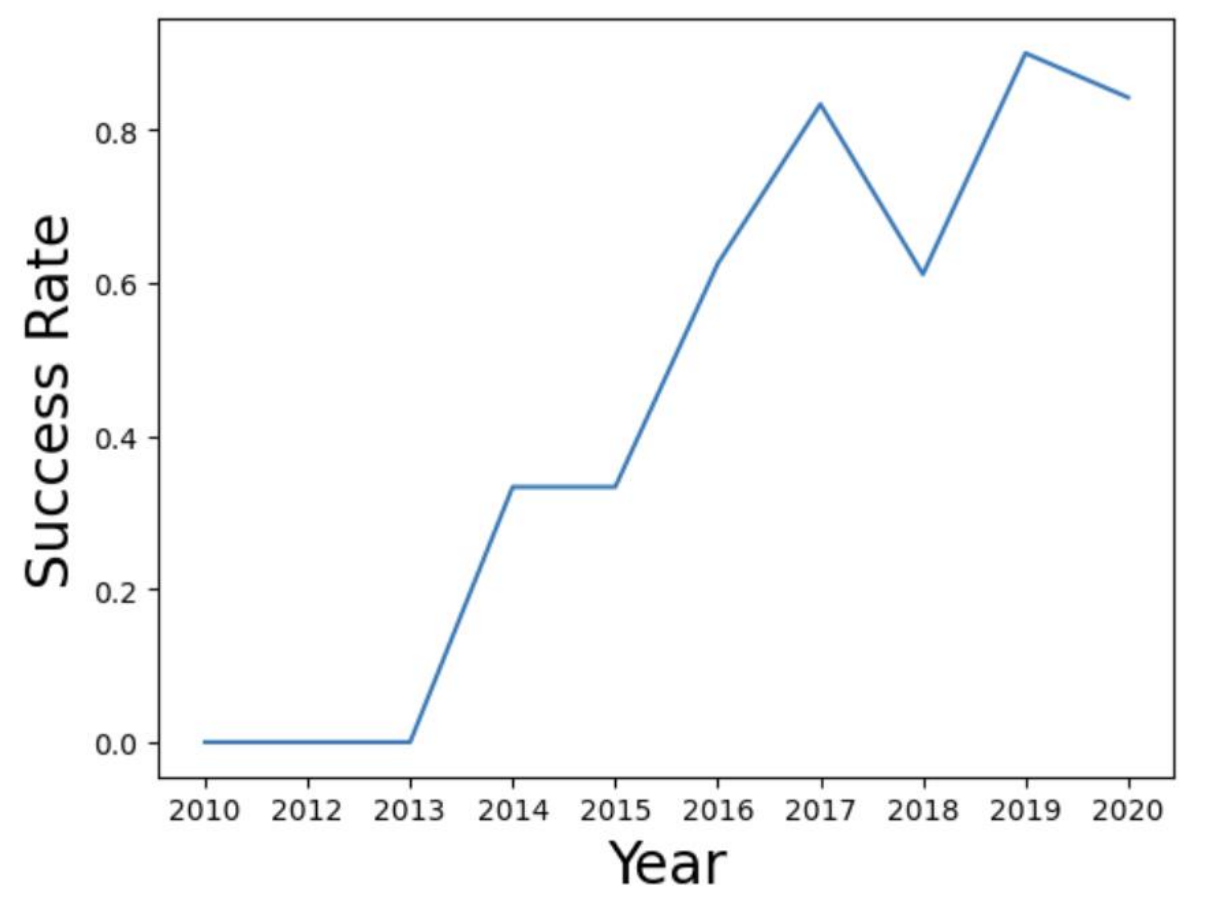
- Exploratory data analysis was conducted, and training labels were determined.
- The number of launches per site, along with the frequency of each orbit, was calculated.
- Additionally, landing outcome labels were derived from the outcome column, and the results were exported to a CSV file.
- Here is the link to the notebook: https://github.com/mamarexa/ibm_data_science_project/blob/main/03_SpaceX_Data_Wrangling.ipynb



EDA with Data Visualization

- The data was explored through visualizations that illustrated the relationships between various parameters. These included the connection between flight number and launch site, payload and launch site, the success rate of each orbit type, flight number and orbit type, as well as the yearly trend in launch success.

https://github.com/mamarexa/ibm_data_science_project/blob/main/05_SpaceX_EDA_Data_Viz.ipynb



EDA with SQL

- SQL queries executed include:
  - Retrieving the names of unique launch sites in space missions.
  - Displaying 5 records of launch sites that start with the string CCA'.
  - Showing the total payload mass carried by boosters for NASA's CRS missions.
  - Calculating the average payload mass carried by the Falcon 9 version 1.1 booster.
  - Identifying the date when a successful landing on a drone ship was first achieved.
  - Listing the names of boosters that successfully landed on the ground pad carrying a payload greater than 4000 but less than 6000.
  - Enumerating the total counts of successful and failed mission outcomes.
  - Revealing the names of booster versions that carried the maximum payload mass.
  - Displaying records that detail the month names, successful landings on the ground pad, and associated booster versions and launch sites for each month in 2015.
  - Ranking the count of successful landing outcomes between June 4, 2010, and March 20, 2017, in descending order.

https://github.com/mamarexa/ibm_data_science_project/blob/main/04_SpaceX_EDA_SQL.ipynb

Build an Interactive Map with Folium

- These objects were used to mark the geographical locations of launch sites and to indicate the success or failure of launches at each site.
- This visualization aids in quickly assessing which locations have higher success rates and understanding the spatial distribution of launch sites relative to important geographic features like railways, highways, and coastlines.

https://github.com/mamarexa/ibm_data_science_project/blob/main/06_SpaceX_Interactive_Visual_Analytics_Folium.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches by a certain sites.
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

https://github.com/mamarexa/ibm_data_science_project/blob/main/07_SpaceX_Interactive_Visual_Analytics_Plotly.py

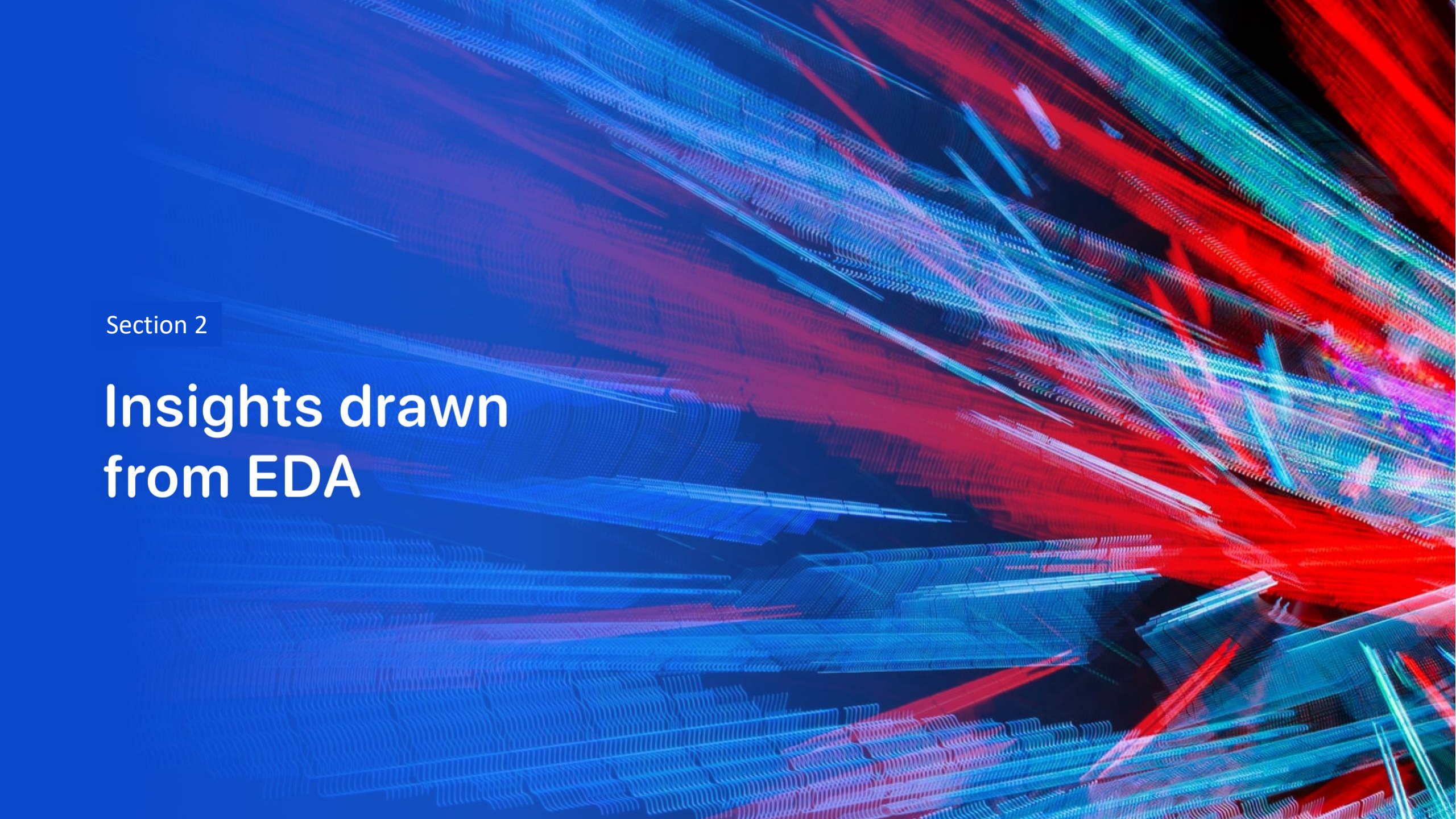
Predictive Analysis (Classification)

- The development process for the classification model in the notebook involves building, evaluating, improving, and identifying the best-performing model through several steps:
  - Model Building: Multiple classification models including SVM, KNN, and Logistic Regression were implemented to predict the successful landing of SpaceX Falcon 9 first stages.
  - Model Evaluation: Each model was evaluated based on accuracy and Area Under the Curve (AUC) metrics, using confusion matrices to further assess performance.
  - Model Improvement: Based on initial assessments, models were fine-tuned and optimized to enhance predictive performance.
  - Best Model Selection: Among the models, the one with the highest accuracy and AUC was selected as the best performer.

https://github.com/mamarexa/ibm_data_science_project/blob/main/08_SpaceX_Machine8Learning_Prediction.ipynb

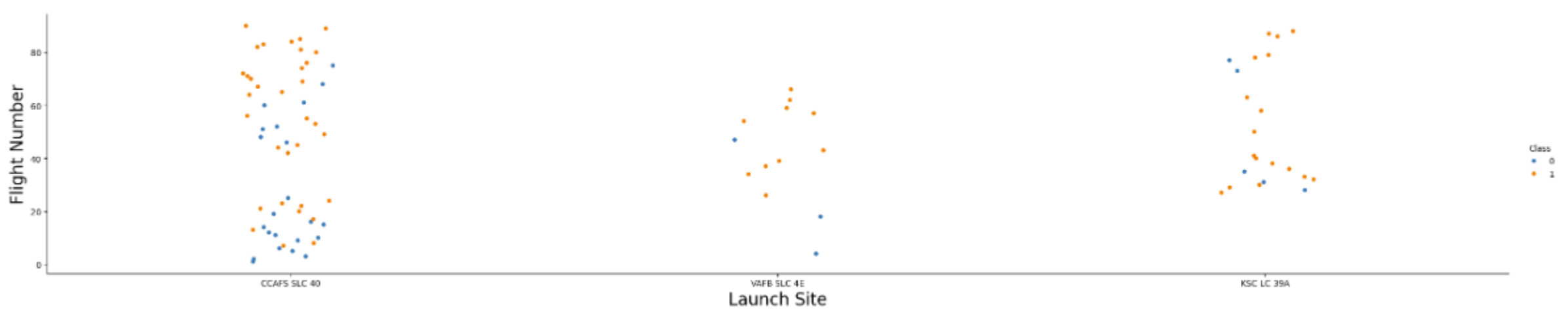
Results

- The SVM, KNN, and Logistic Regression models excel in prediction accuracy for this dataset.
- Payloads with lower weights tend to perform better than their heavier counterparts.
- The success rates of SpaceX launches increase over time as they refine their launch processes.
- Kennedy Space Center Launch Complex 39A has recorded the highest number of successful launches compared to other sites.
- The orbits GEO, HEO, SSO, and ES L1 are noted for having the highest success rates.


Section 2
Insights drawn from EDA

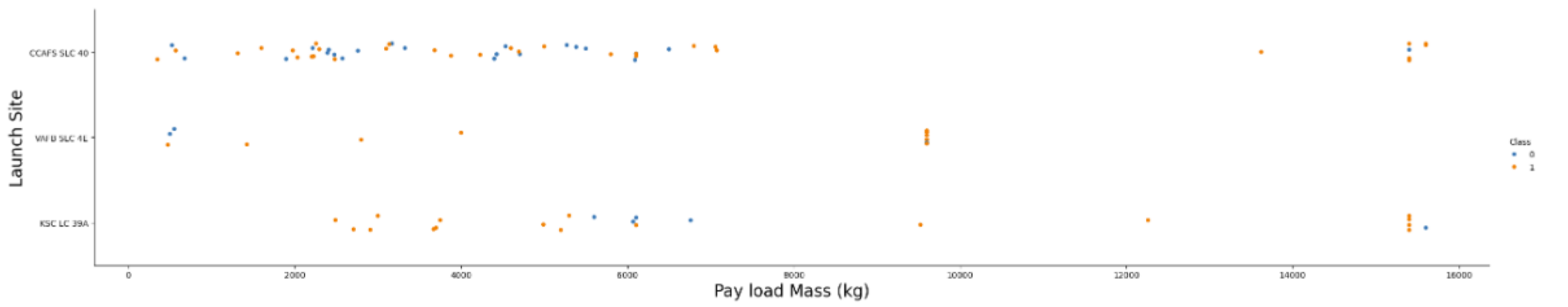
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



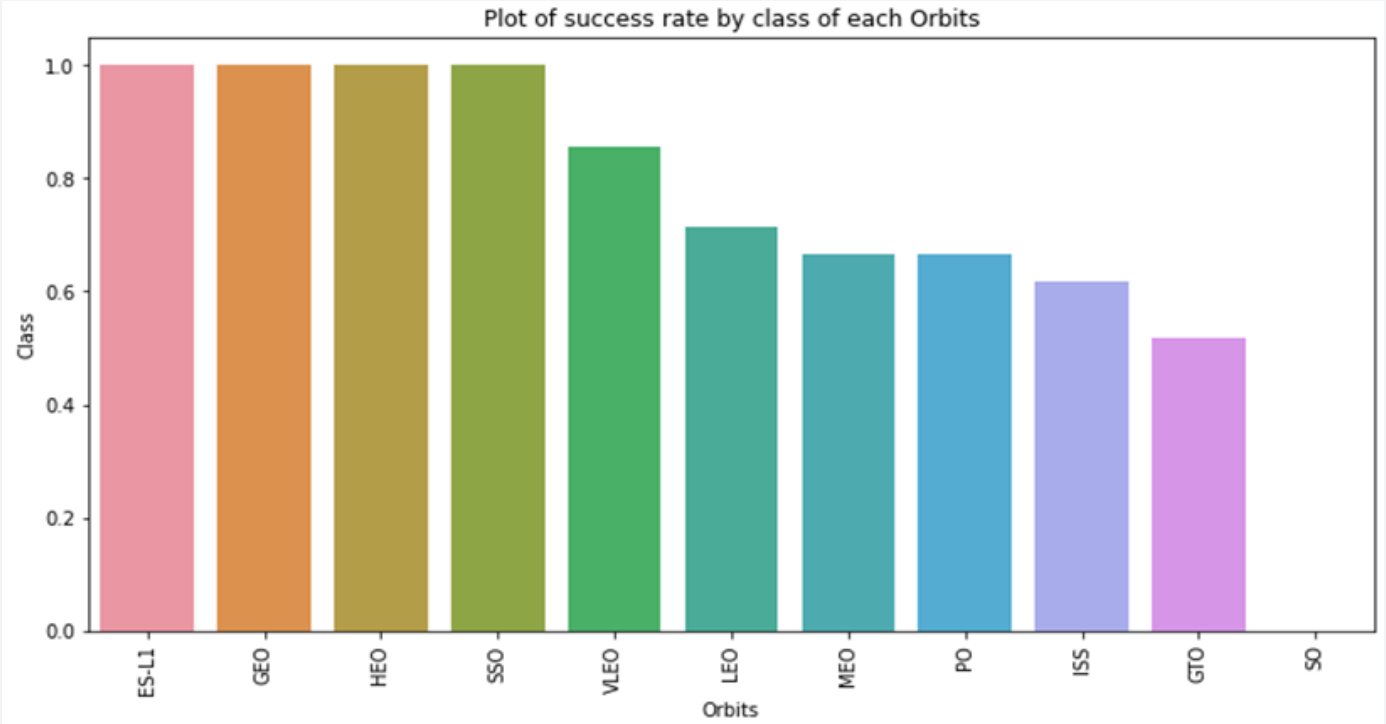
Payload vs. Launch Site

- Most payloads with lower mass have been launched from Cape Canaveral Air Force Station Launch Complex 40 (CCAFS SLC 40).



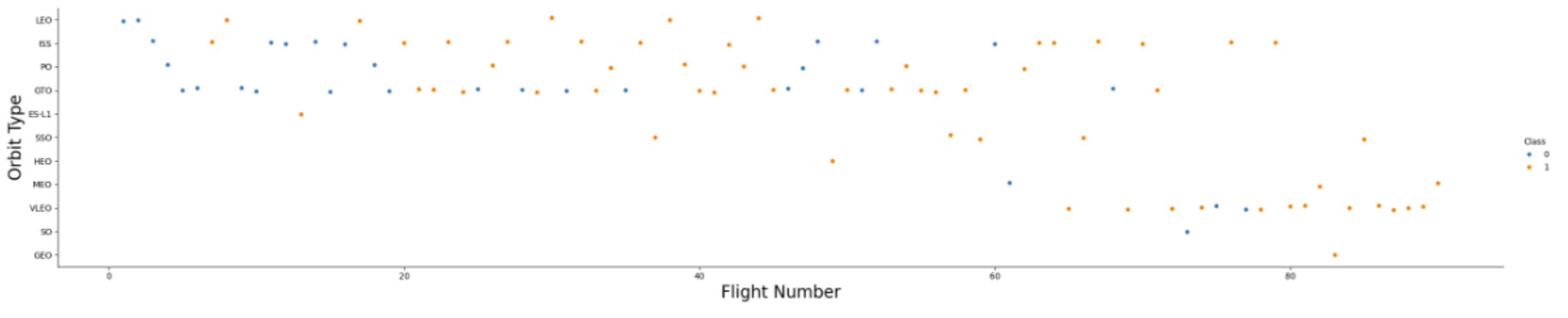
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



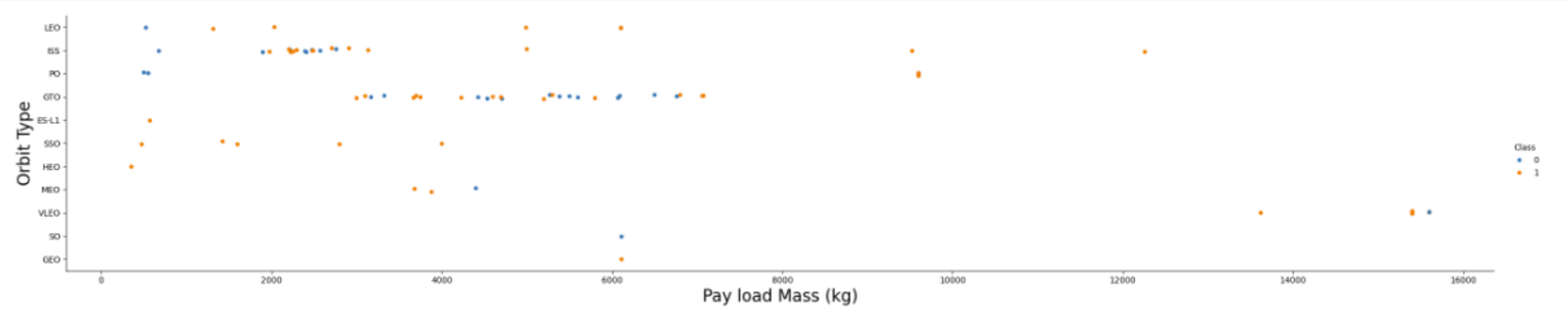
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



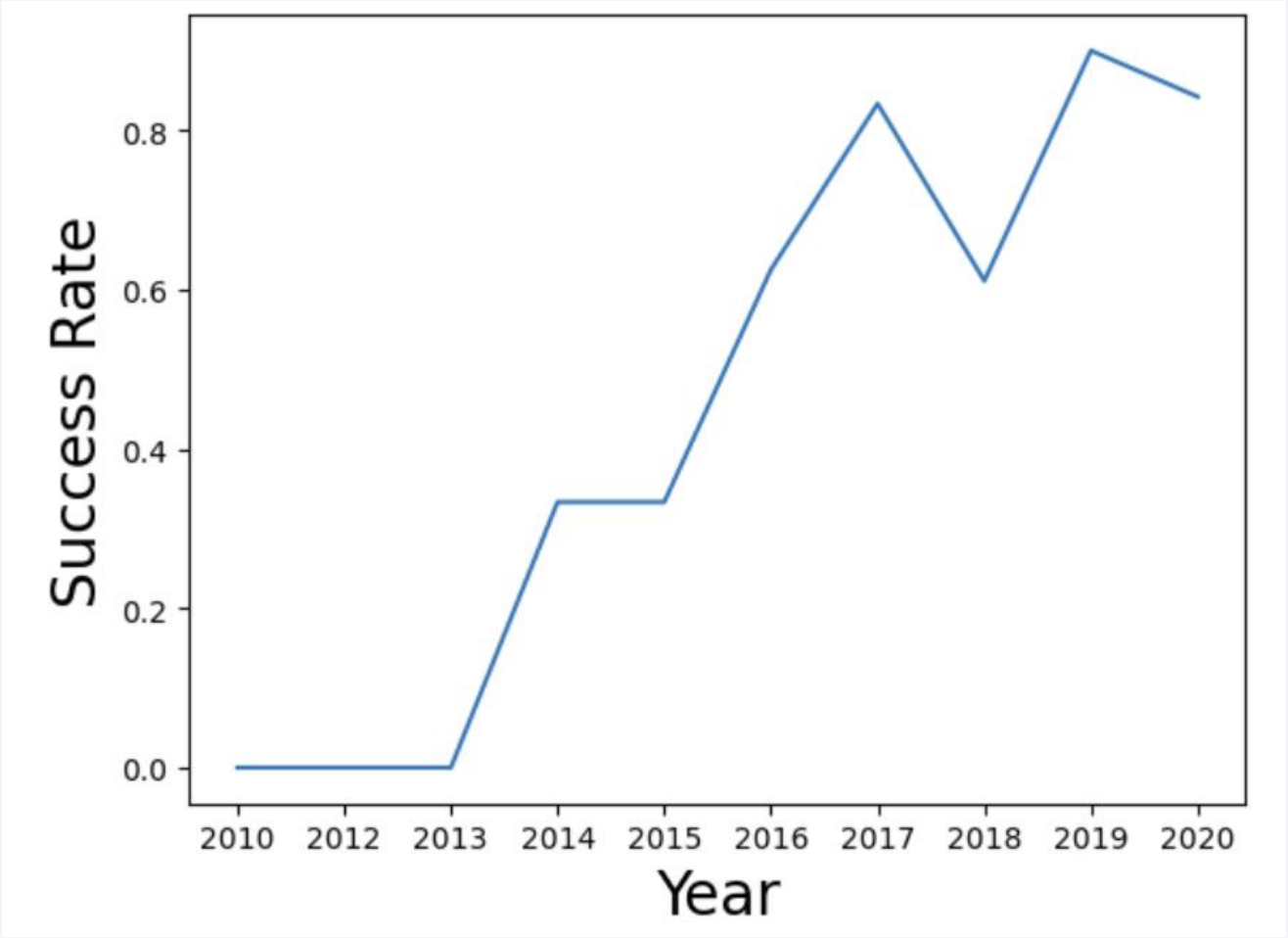
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- Using the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%sql select distinct "LAUNCH_SITE" from SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

- Using the query below to display 5 records where launch sites begin with `CCA`

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

```
%sql select * from SPACEXTBL where "LAUNCH_SITE" like 'CCA%' limit 5;
```

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

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

```
%sql select sum("PAYLOAD_MASS__KG_") from SPACEXTBL where Customer like 'NASA%';
```

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

sum("PAYLOAD_MASS__KG_")
99980

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2534.

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

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTBL where "Booster_Version" like "F9 v1.1%";
```

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

avg("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015.

```
%sql select MIN(Date) from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)';
```

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

MIN(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.

```
%sql select * from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)' and "PAYLOAD_MASS__KG_" betwee
```

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

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

Total Number of Successful and Failure Mission Outcomes

- We query below to filter where mission outcome was a success or a failure.

```
%sql select "Mission_Outcome", COUNT(*) as total_number from SPACEXTBL group by "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

```
%sql select "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select MAX("PAYLOAD_MASS__KG_") fr
```

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

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

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

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

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

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

- 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 2017-03-20.
- 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.

Section 3

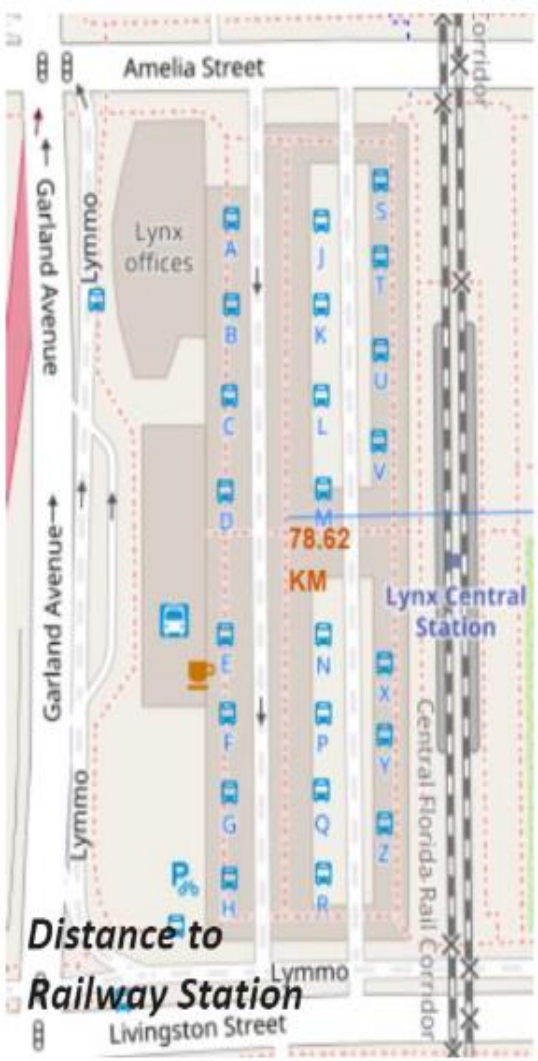
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

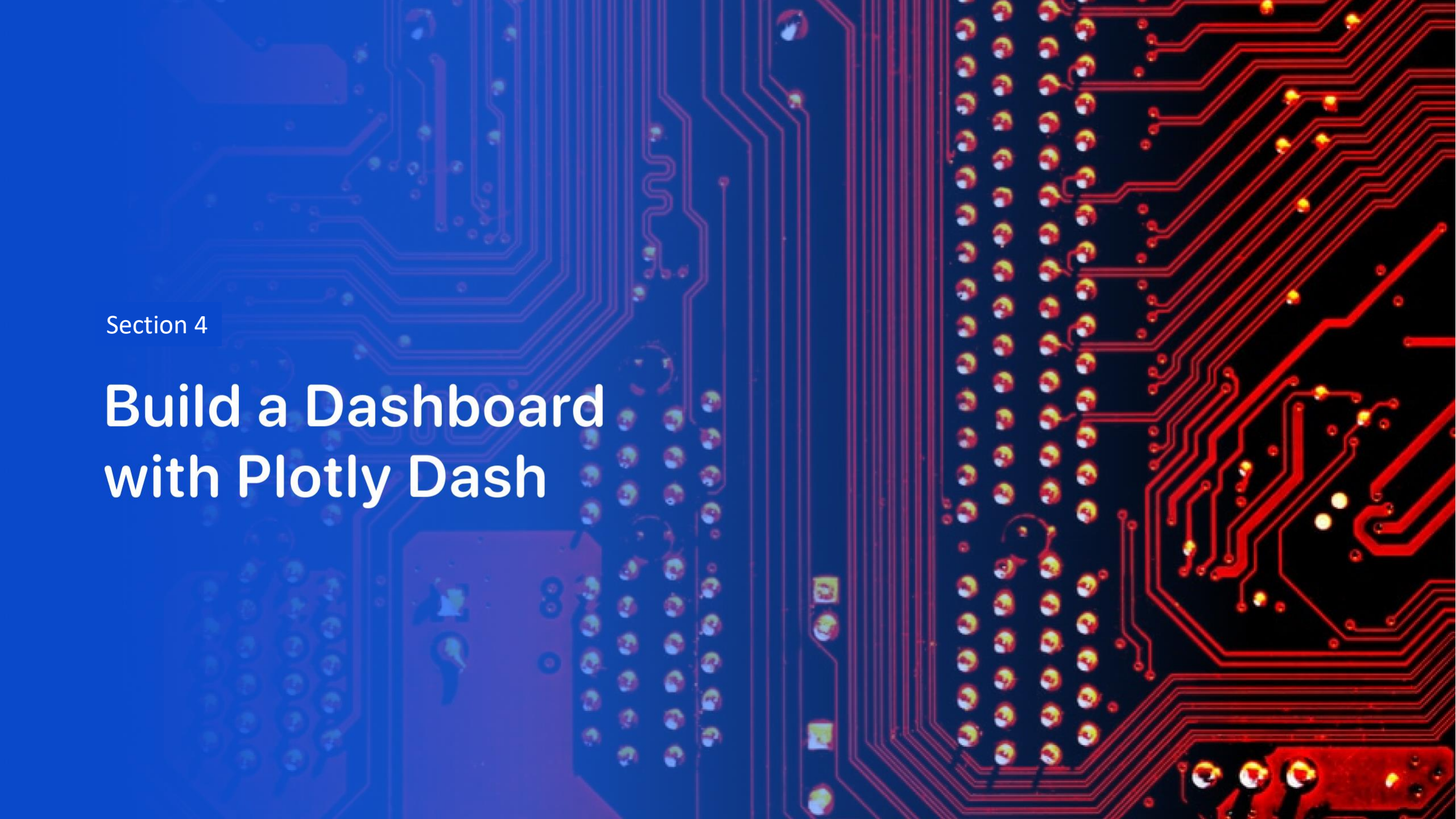
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

<Folium Map Screenshot 2>

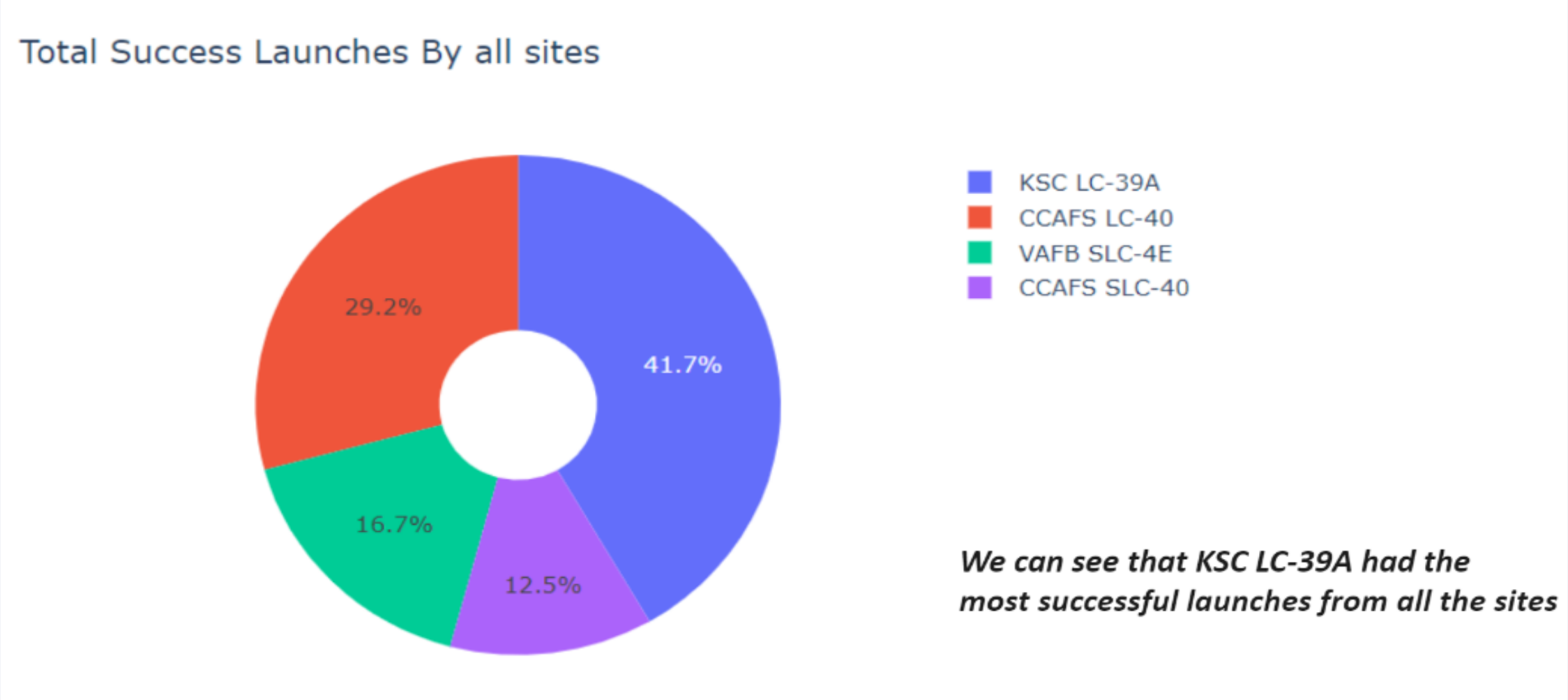
<Folium Map Screenshot 3>



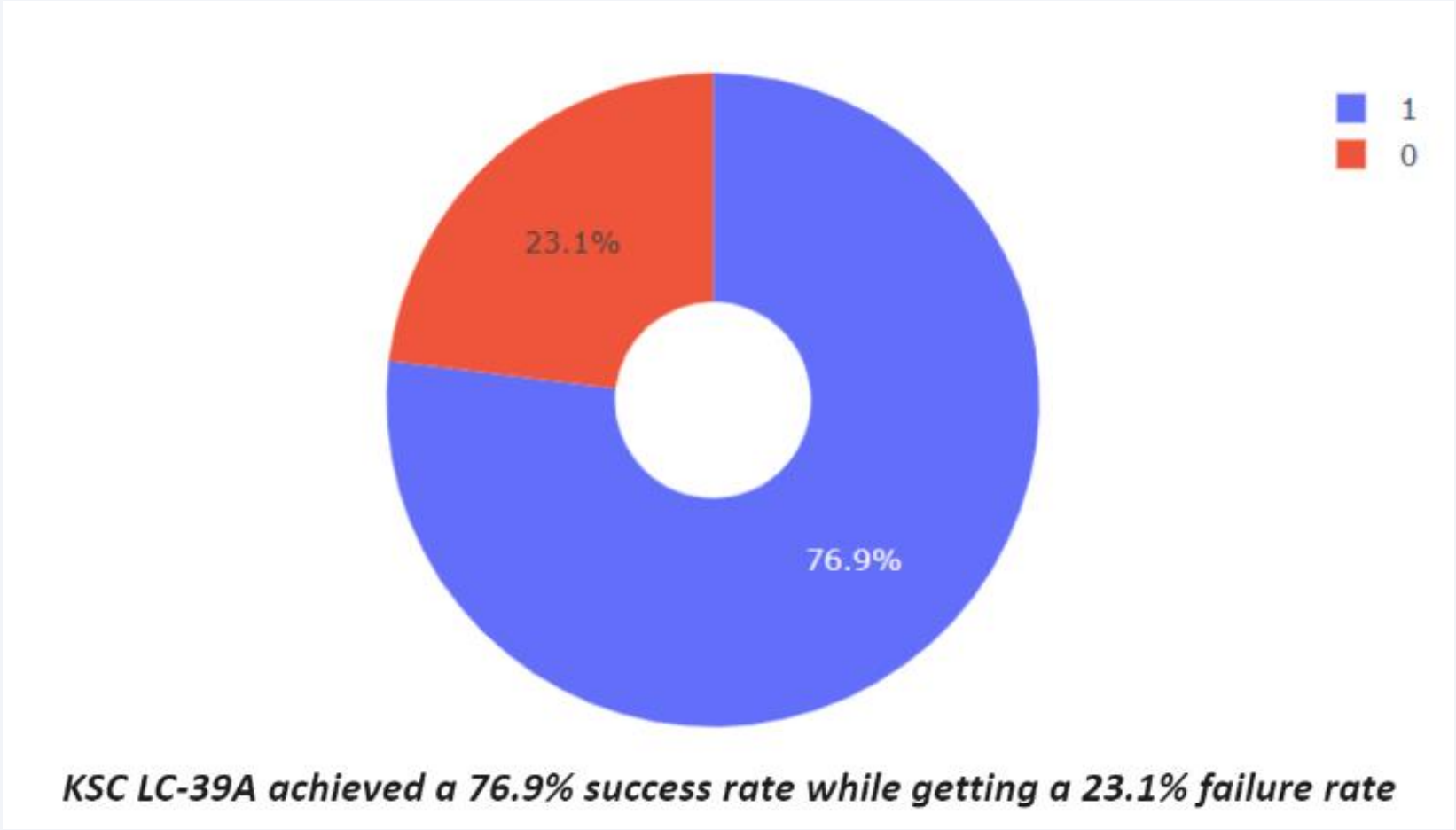
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

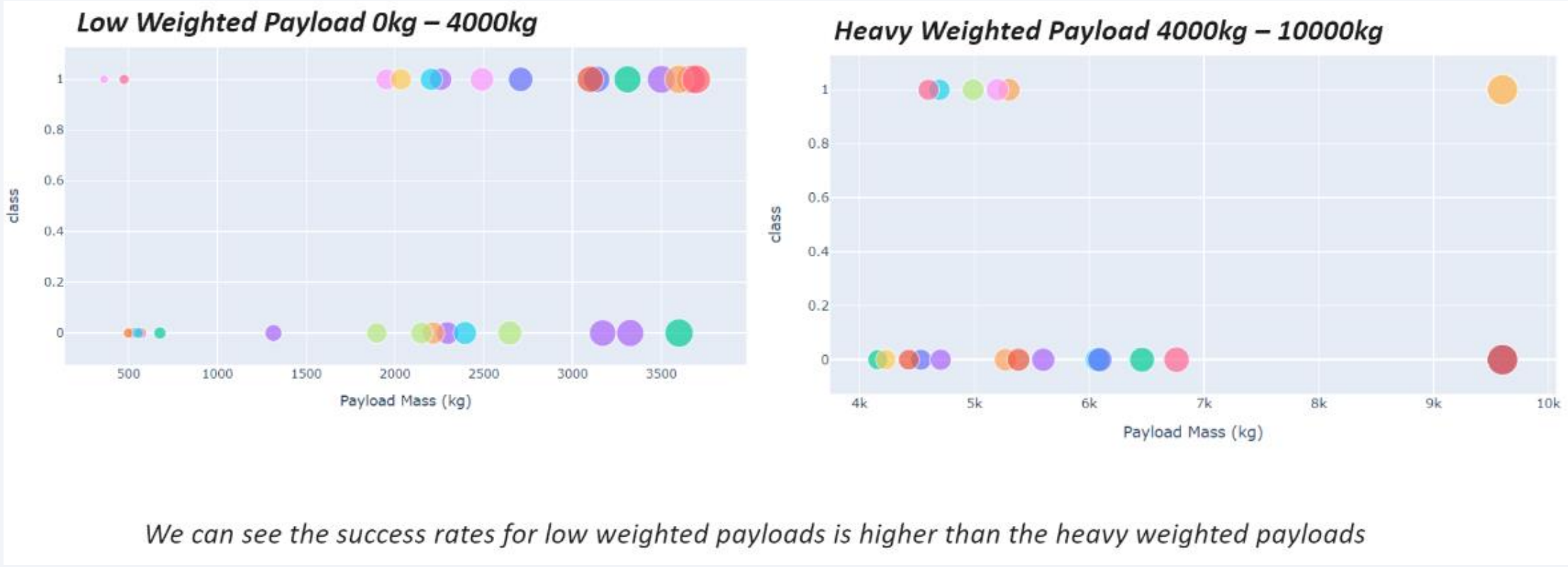


<Dashboard Screenshot 2>



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Dashboard Screenshot 3>



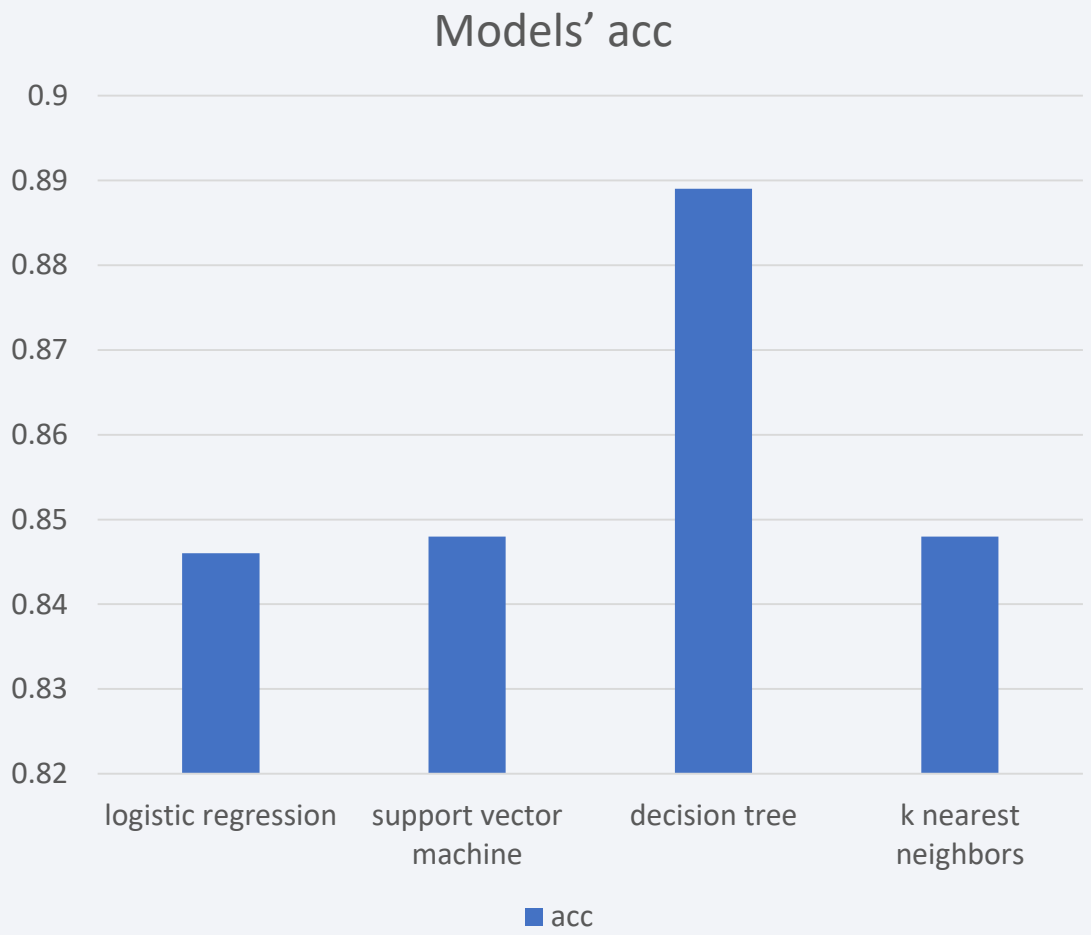
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

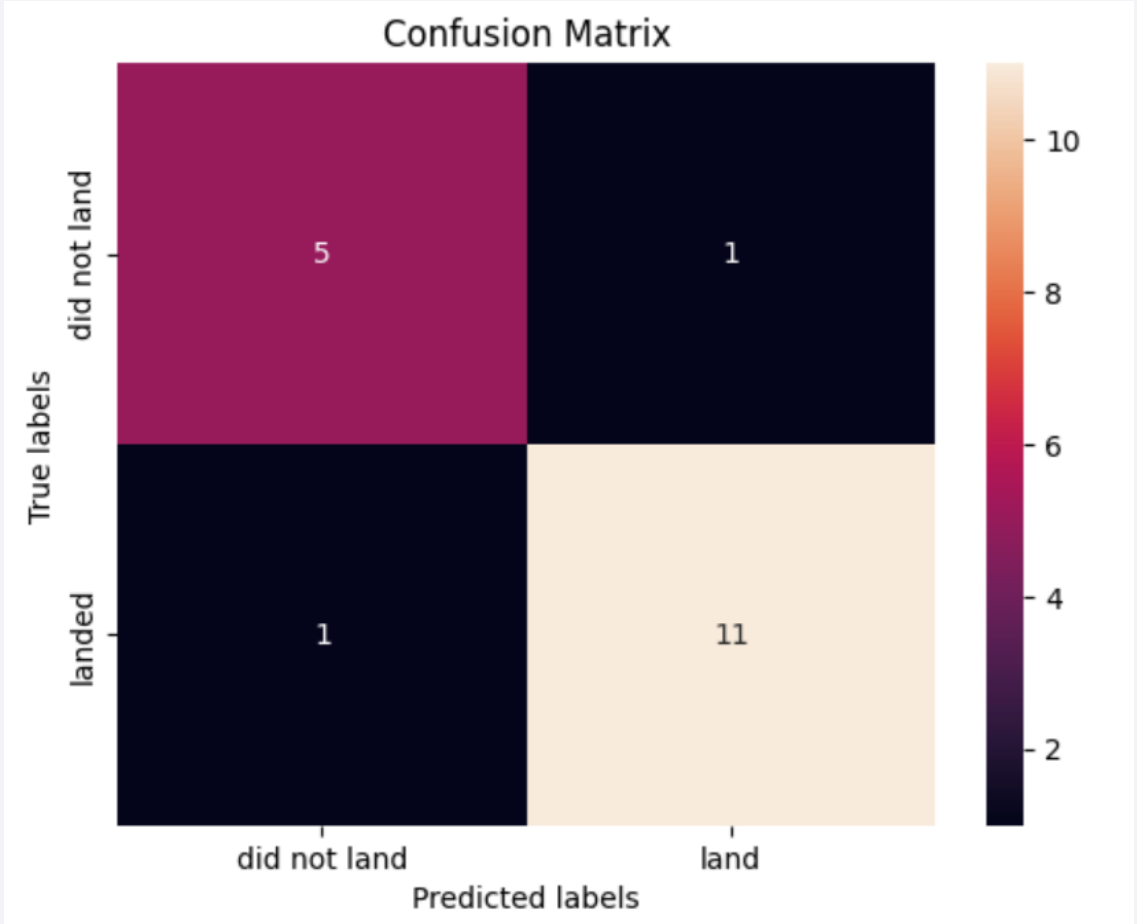
Classification Accuracy

- Based on the chart, decision tree model has the best accuracy with almost 0.89 accuracy.



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.
- There are only two predictions were actually wrongly predicted.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this problem.

Thank you!