Winning Space Race with Data Science

António Figueiredo

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection, wrangling and formatting using SpaceX API and Web Scraping
  - Exploratory Data Analysis using, Pandas, Numpy and SQL
  - Data Visualization using Matplotlib and Seaborn
  - Interactive Visual Analytics with Folium and Dash
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics screenshots
  - Predictive Analytics result

Introduction

- **Project background and context**

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. This project is geared towards predicting the success of the first phase retrieval event, thereby offering predictive insights aimed at enhancing decision-making within the space industry

- Problems you want to find answers

The main question that we are trying to answer is, for a given set of features about a Falcon 9 rocket launch which include its payload mass, orbit type, launch site, and so on, will the first stage of the rocket land successfully? And, what operating conditions needs to be in place to ensure a successful landing program

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

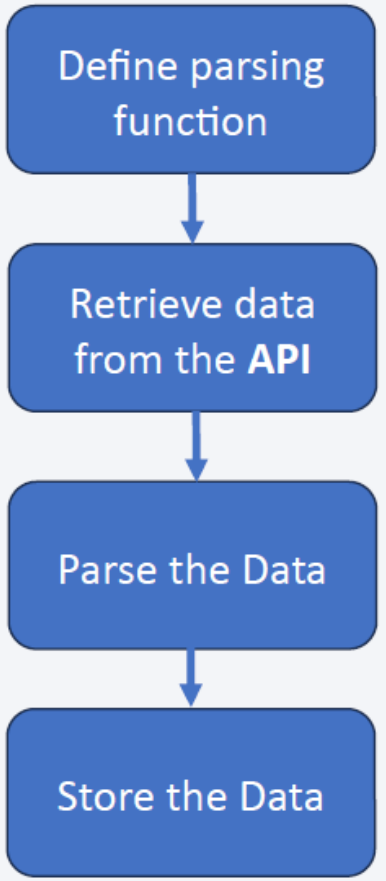
- Data was collected using a GET request to the SpaceX API.
- The response content was then decoded as JSON using the .json() function and converted into a pandas DataFrame with .json_normalize().
- Next, the data was cleaned, missing values were identified, and necessary imputations were made.
- Additionally, web scraping was performed on Wikipedia using BeautifulSoup to retrieve Falcon 9 launch records.
- The goal was to extract the launch records from an HTML table, parse the data, and convert it into a pandas DataFrame for further analysis.

Data Collection – SpaceX API

- To gather data, we sent a GET request to the SpaceX API, then processed and refined the retrieved information by cleaning, structuring, and formatting it for further analysis.

- The completed SpaceX notebook can be found on GitHub URL

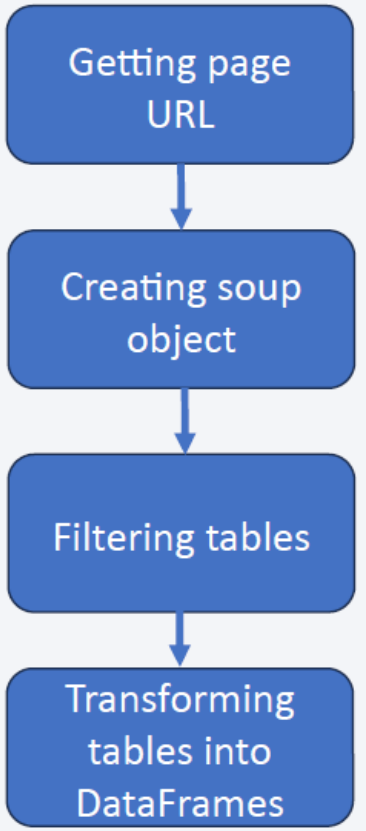
https://github.com/amgfigueiredo



Data Collection - Scraping

- Using BeautifulSoup, we extracted Falcon 9 launch records through web scraping.
- The retrieved data was then processed by parsing the table and transforming it into a structured pandas DataFrame.
- The completed SpaceX notebook can be found on GitHub URL

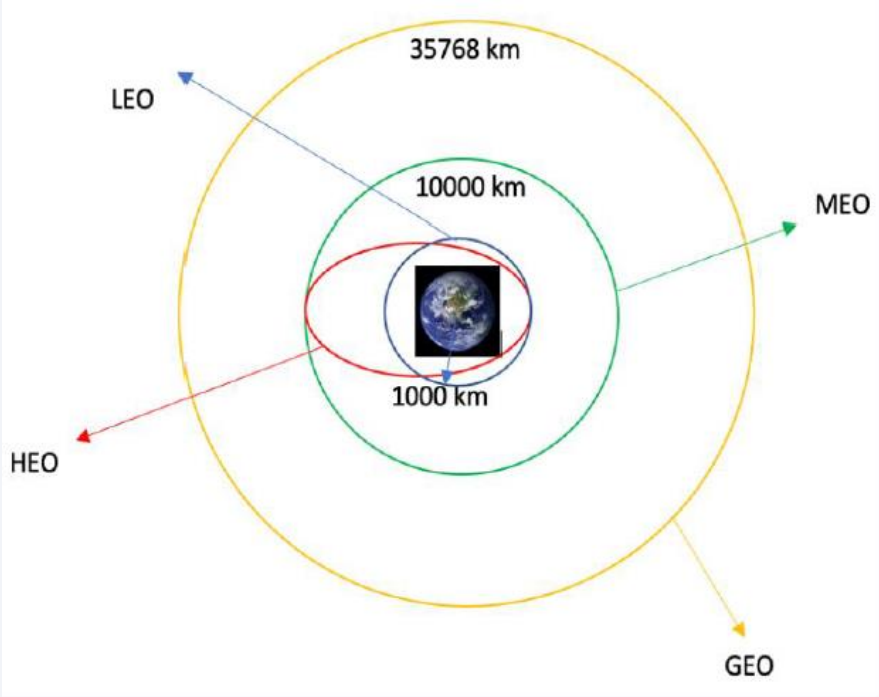
https://github.com/amgfigueiredo



Data Wrangling

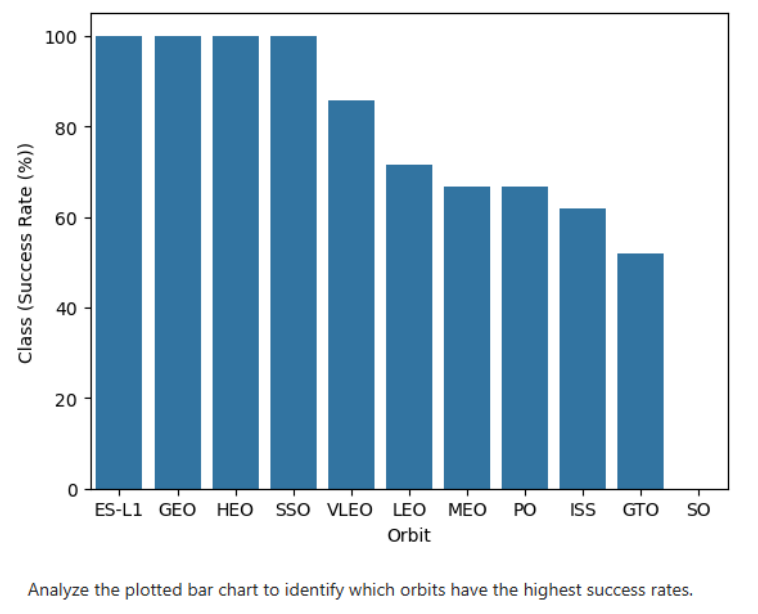
- We conducted exploratory data analysis to define the training labels.
- The analysis included counting the number of launches per site and examining the frequency of different orbit types.
- Additionally, we derived the landing outcome label from the outcome column and saved the processed data as a CSV file.
- The completed SpaceX notebook can be found on GitHub URL

https://github.com/amgfigueiredo

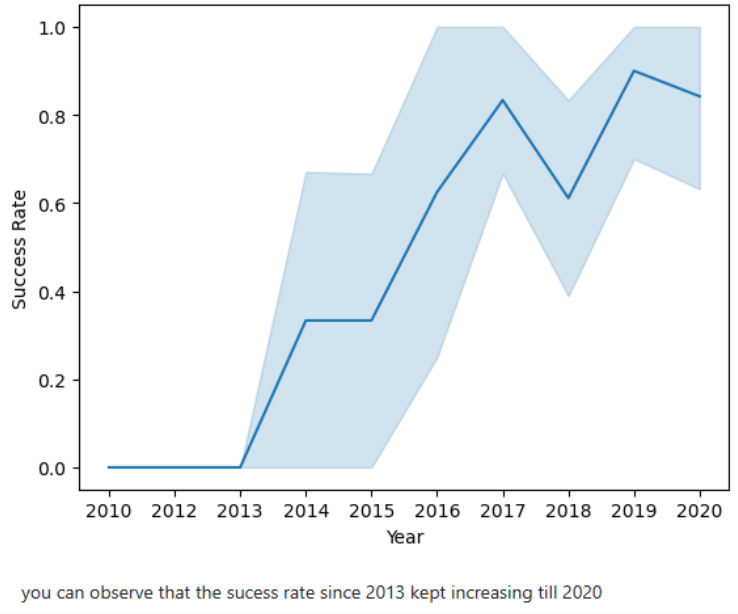


EDA with Data Visualization

- We analyzed the data by creating visual representations to examine patterns, including the correlation between flight number and launch site, payload and launch location, success rates across different orbit types, as well as the relationship between flight number, orbit type, and launch success.



Analyze the plotted bar chart to identify which orbits have the highest success rates.



you can observe that the sucess rate since 2013 kept increasing till 2020

The completed SpaceX notebook can be found on GitHub URL

https://github.com/amgfigueiredo

EDA with SQL

- Using SQL, we had performed many queries to get better understanding of the dataset for instance:

- The names of unique launch sites in the space mission.
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- The total number of successful and failure mission outcomes
- The failed landing outcomes in drone ship, their booster version and launch site names.

- The completed SpaceX notebook can be found on GitHub URL

https://github.com/amgfigueiredo

Build an Interactive Map with Folium

- We plotted all launch sites on a Folium map and incorporated visual elements like markers, circles, and lines to indicate the success or failure of launches at each location.
- Launch outcomes were categorized into two classes: 0 for failure and 1 for success.
- By utilizing color-coded marker clusters, we identified launch sites with higher success rates.
- Additionally, we measured the distances between each launch site and nearby locations, addressing key questions related to proximity and site performance.

- The completed SpaceX notebook can be found on GitHub URL

https://github.com/amgfigueiredo

Build a Dashboard with Plotly Dash

- We developed an interactive dashboard using Plotly Dash.
- Pie charts were created to visualize the total number of launches at various sites.
- A scatter plot was generated to analyze the correlation between payload mass (kg) and launch outcome across different booster versions.

- The completed SpaceX notebook can be found on GitHub URL

https://github.com/amgfigueiredo

Predictive Analysis (Classification)

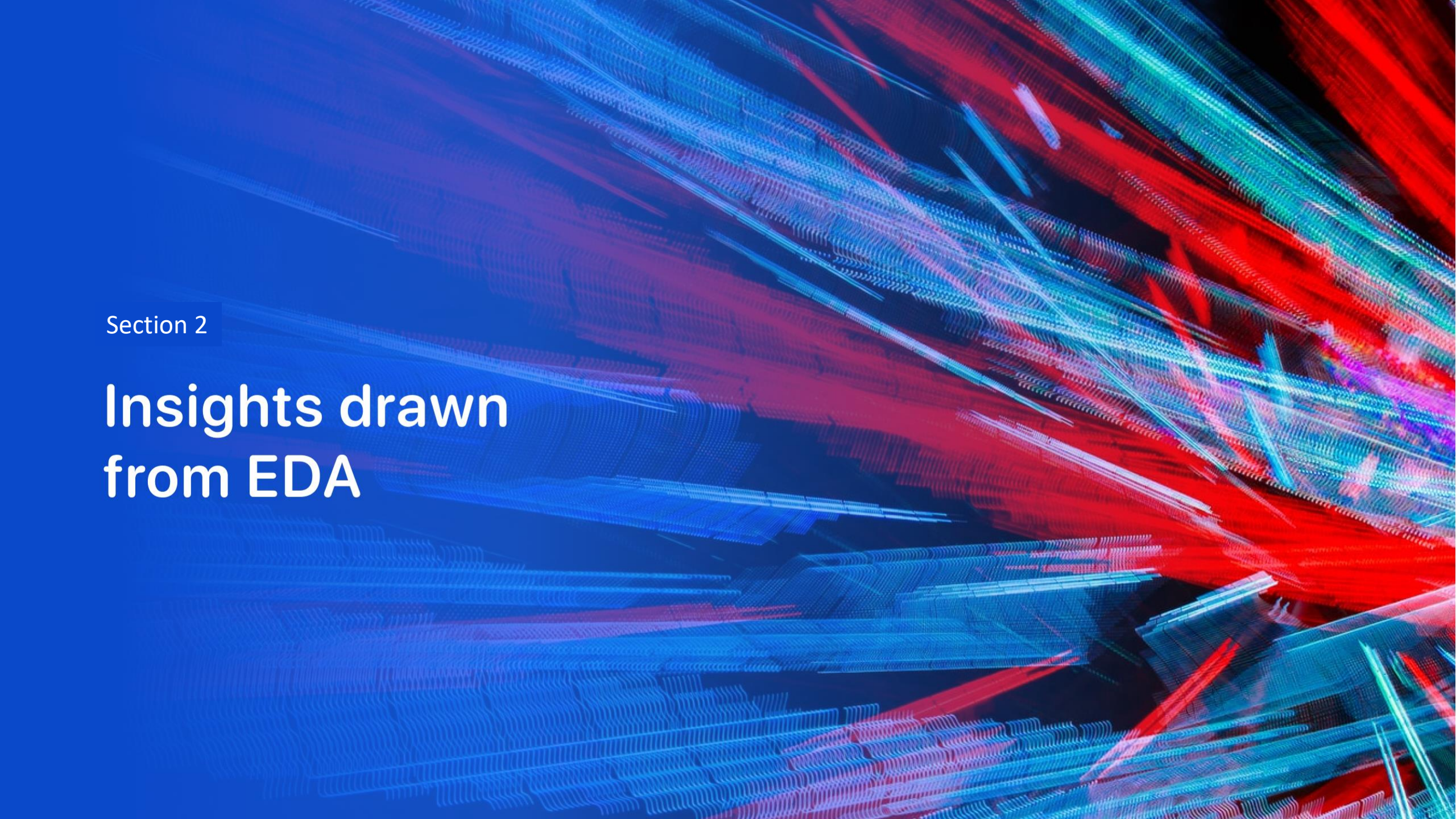
- We utilized NumPy and pandas to load and preprocess the data, followed by splitting it into training and testing sets.
- Various machine learning models were developed, and hyperparameters were optimized using GridSearchCV.
- Accuracy was chosen as the evaluation metric, and model performance was enhanced through feature engineering and fine-tuning.
- Ultimately, we identified the best-performing classification model.
- The completed SpaceX notebook can be found on GitHub URL

https://github.com/amgfigueiredo

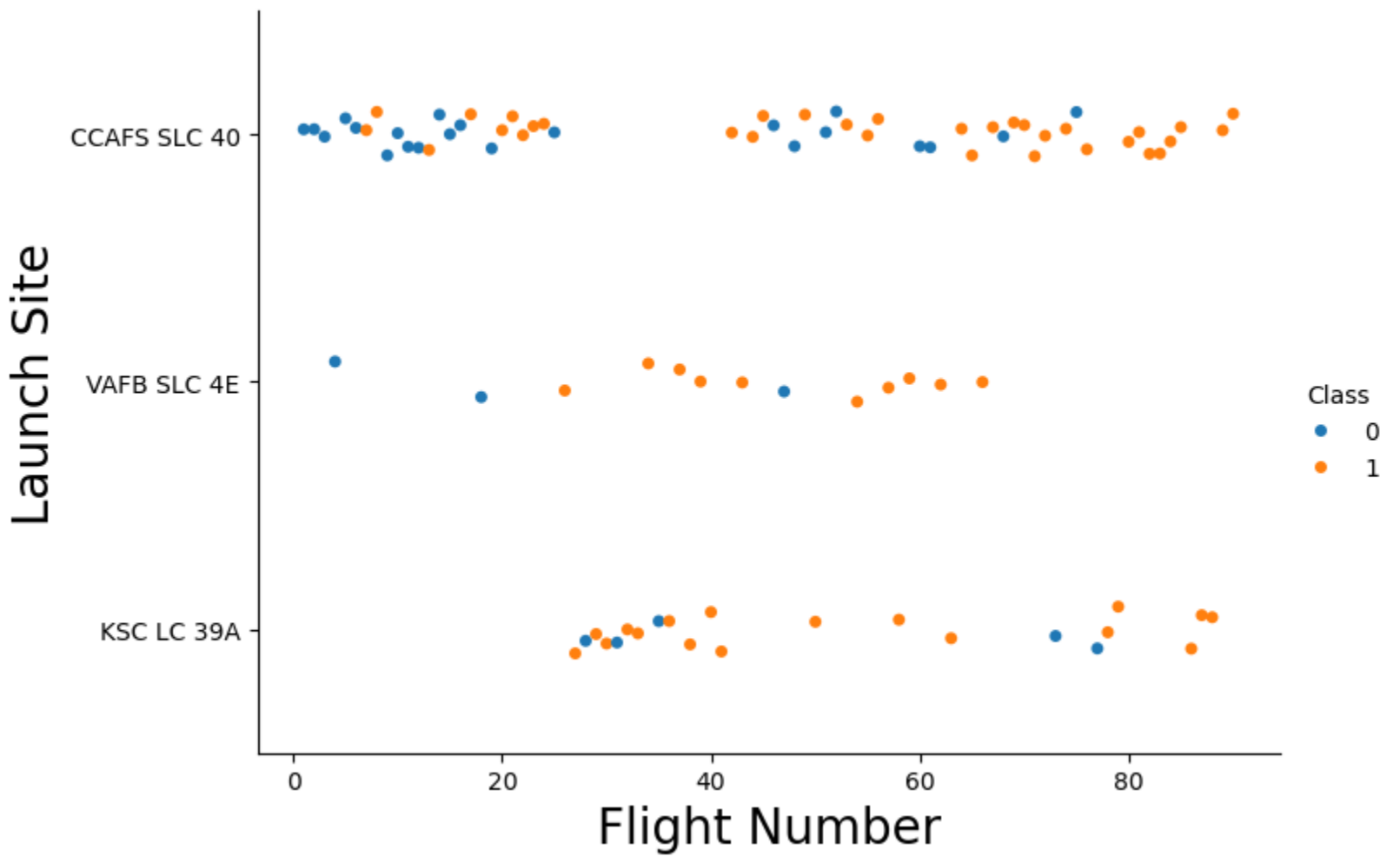
Results

- The results are split into 5 sections:
    - SQL (EDA with SQL)
    - Matplotlib and Seaborn (EDA with Visualization)
    - Folium
    - Dash
    - Predictive Analysis

In all of the graphs that follow, class 0 represents a failed launch outcome while class 1 represents a successful launch outcome.

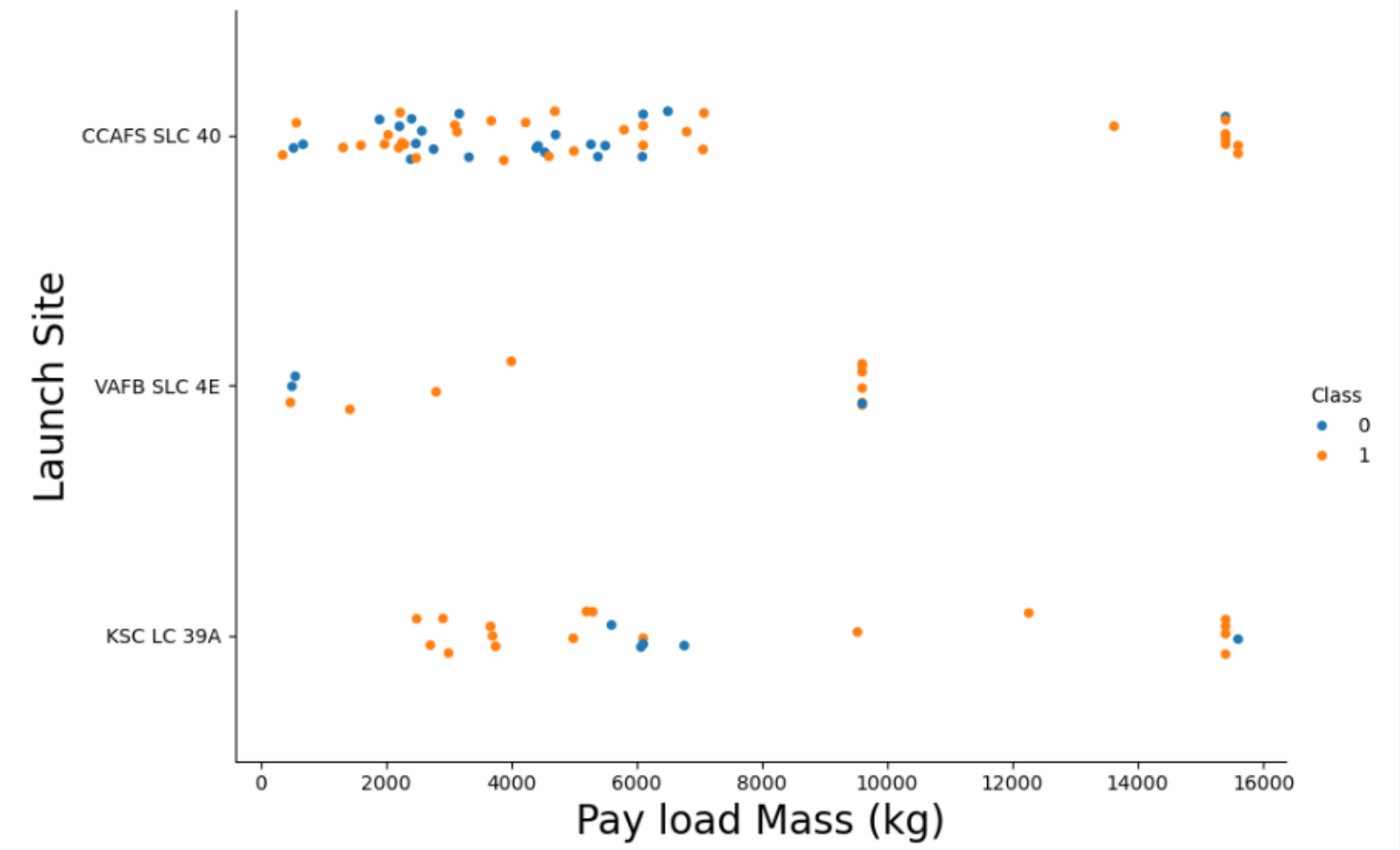

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



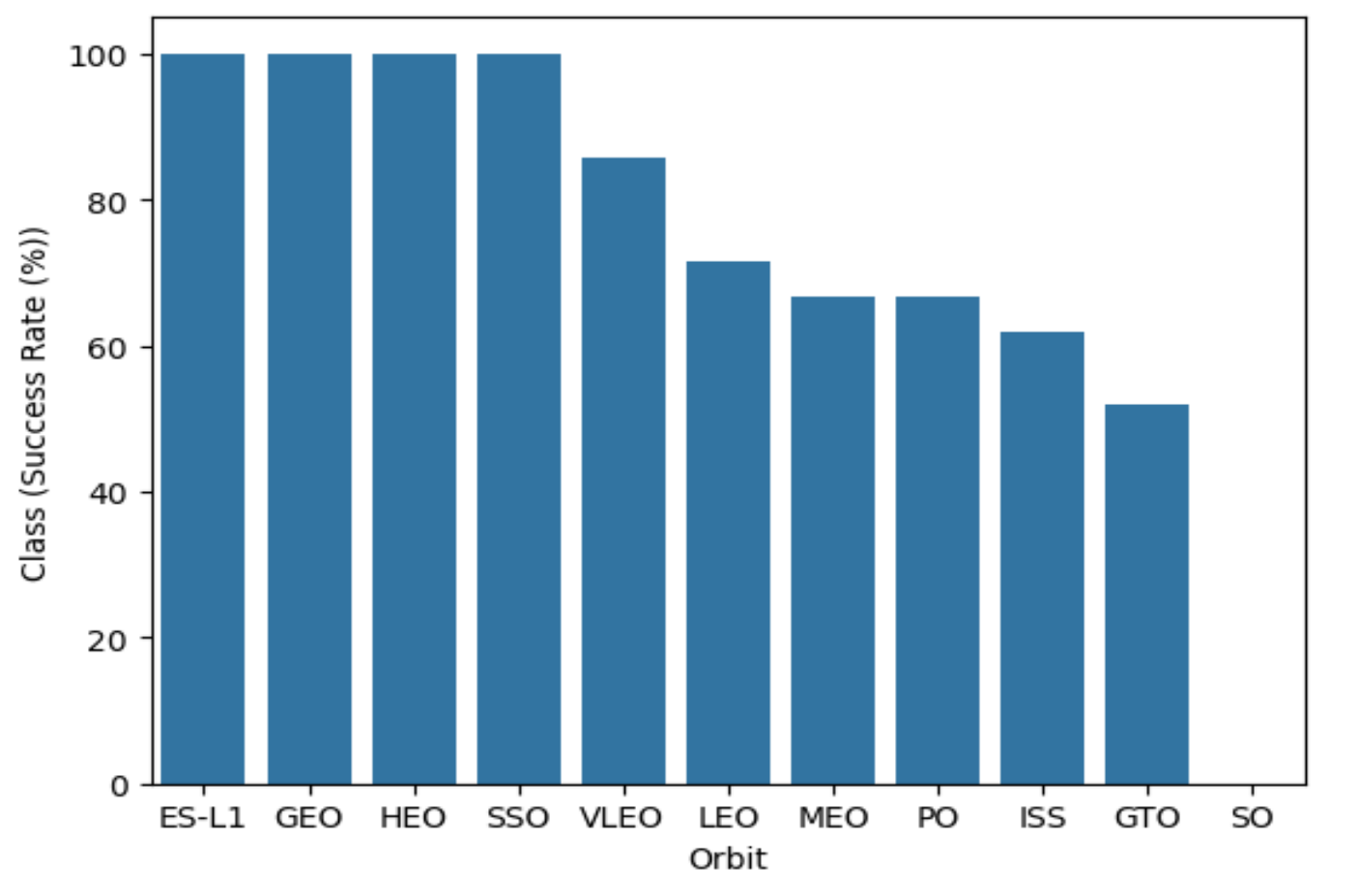
The plot revealed a positive correlation between the number of flights at a launch site and its success rate, indicating that sites with more launches tend to have higher success rates.

Payload vs. Launch Site



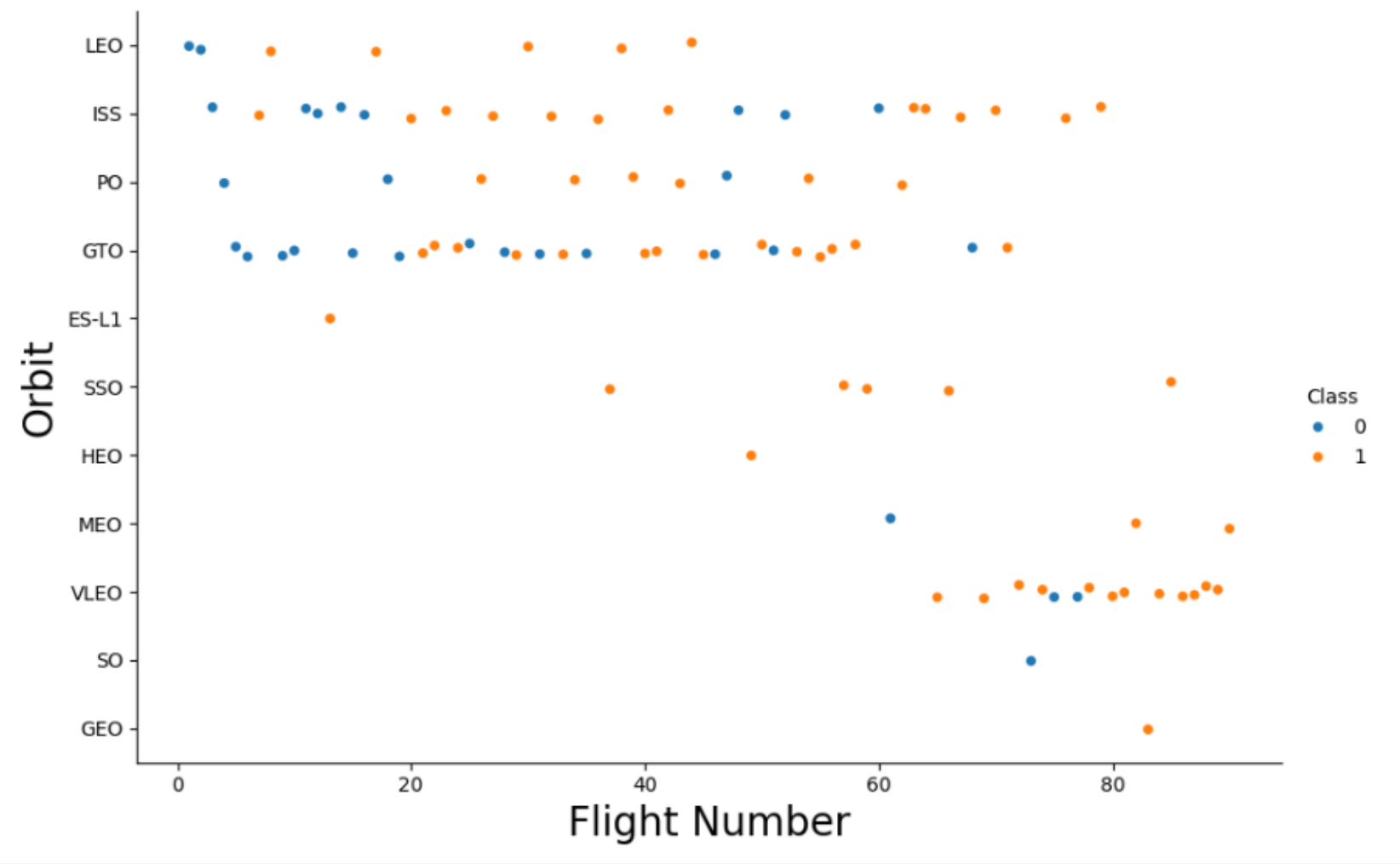
For the **CCAFS SLC 40** launch site, a **higher payload mass** is associated with an **higher rocket success rate.**

Success Rate vs. Orbit Type



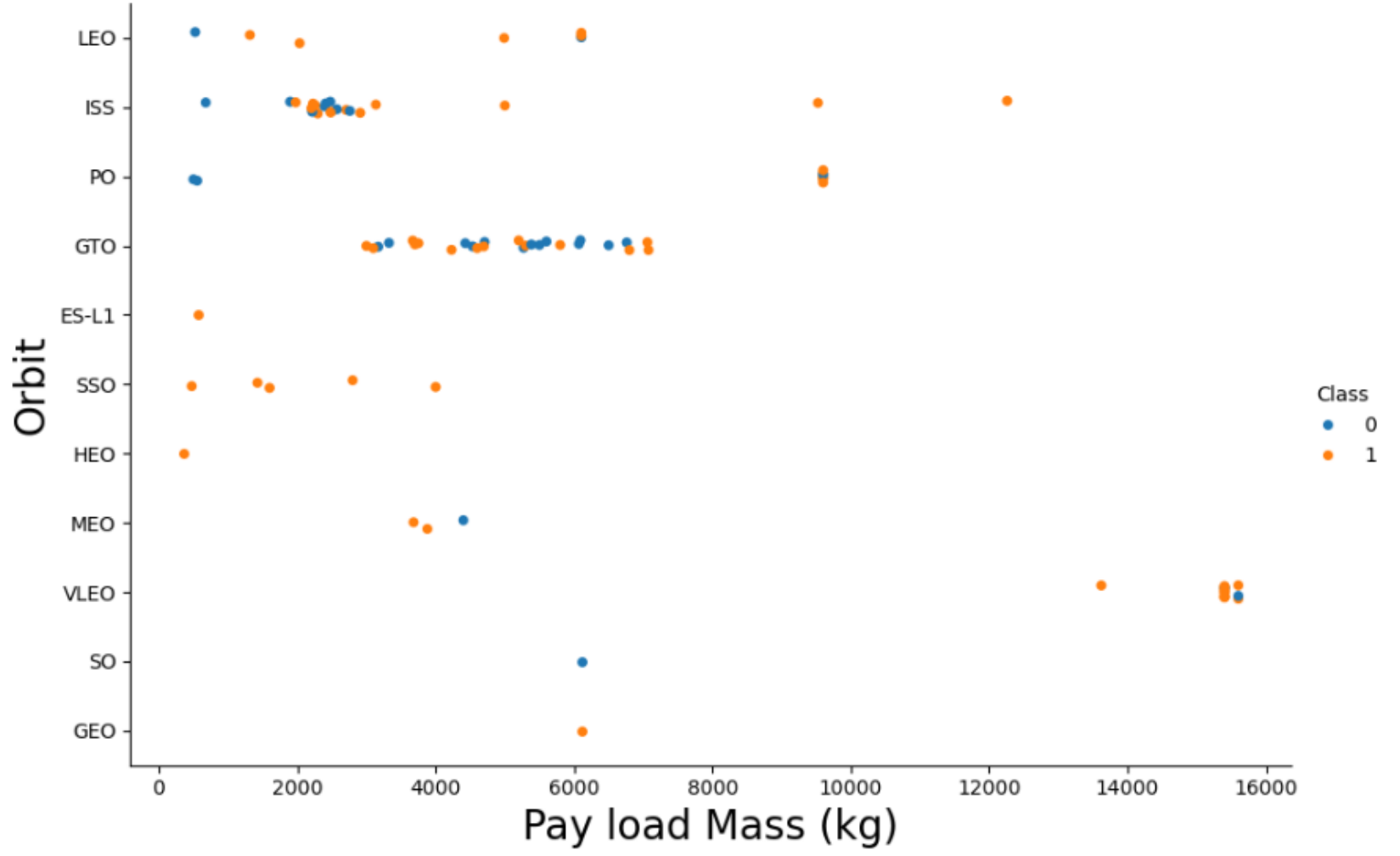
From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type



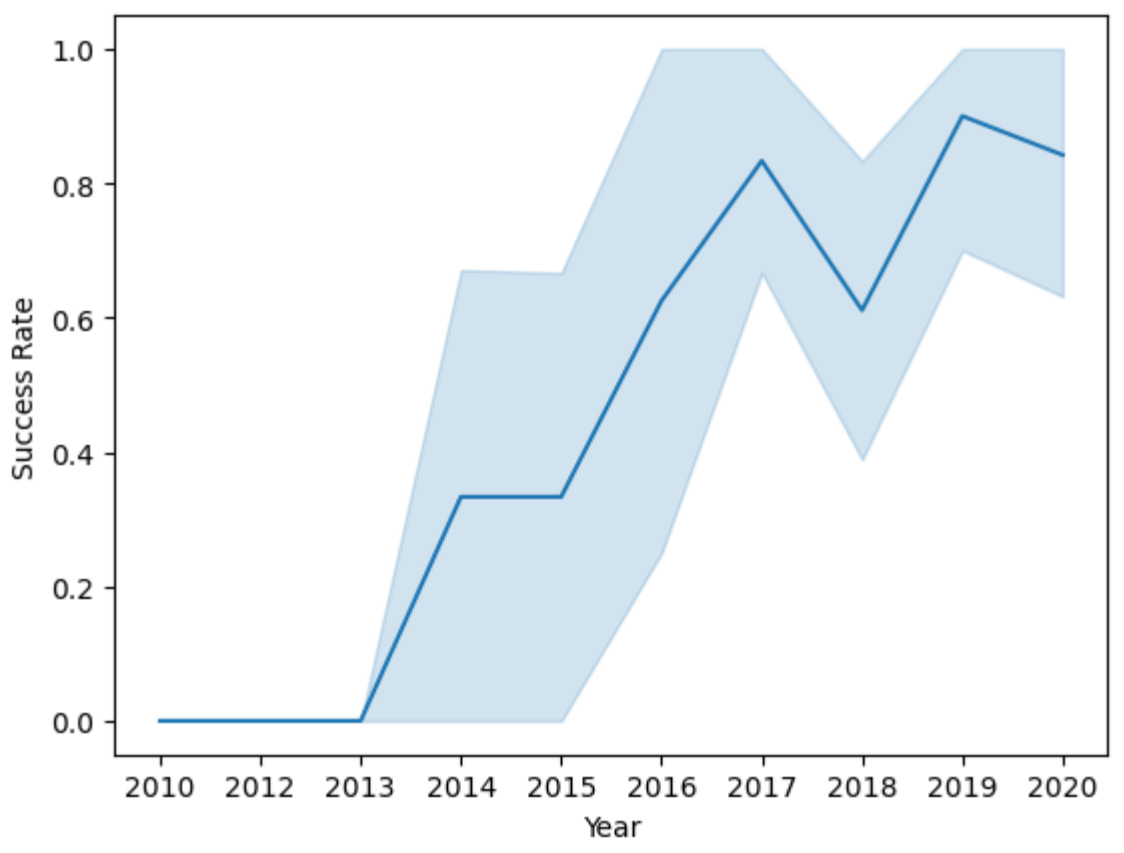
- The plot illustrates the relationship between Flight Number and Orbit Type.
- It shows that, for the LEO orbit, success is influenced by the number of flights, while no such correlation exists between flight number and success in the GEO orbit.

Payload vs. Orbit Type



- It is seen that heavier payloads tend to result in more successful landings for PO, LEO, and ISS orbits.

Launch Success Yearly Trend



We can see that success rate since 2013 kept on increasing till 2020.

All Launch Site Names

Task 1

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

In [10]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[10]:

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

- We used the SELECT DISTINCT to show only unique launch sites from the SpaceX data

Launch Site Names Begin with 'CCA'

In [11]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[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

We used the query with a LIMIT to display 5 records where launch sites begin with `CCA`

Total Payload Mass

```
In [13]: %sql SELECT Customer, SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass (Kg)" FROM SPACEXTBL WHERE Customer ='NASA (CRS)';
 * sqlite:///my_data1.db
Done.
```

Out[13]:

Customer	Total Payload Mass (Kg)
NASA (CRS)	45596

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

Average Payload Mass by F9 v1.1

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

In [14]:
```
%sql SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass (Kg)" FROM SPACEXTBL WHERE Booster_Versic LIKE'F9 v1.1%';
```

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

Out[14]:

Booster_Version	Average Payload Mass (Kg)
F9 v1.1 B1003	2534.6666666666665

- We calculated the average payload mass carried by booster version F9 v1.1 as 2534.67

First Successful Ground Landing Date

In [25]:
```
%sql SELECT Landing_Outcome, Min(DATE) FROM 'SPACEXTBL' WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

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

Out[25]:

Landing_Outcome	Min(DATE)
Success (ground pad)	2015-12-22

- We found that the date of the first successful landing outcome on ground pad was on **December 22, 2015**

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [31]: %sql SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (drone ship)  AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
 * sqlite:///my_data1.db
Done.
```

Out[31]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

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

Total Number of Successful and Failure Mission Outcomes

In [34]:
```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) AS Total_Number FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

Out[34]:

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

Total number of Mission outcome was a success or a failure

Boosters Carried Maximum Payload

In [36]:
```
%sql SELECT "Booster_Version","PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Out[36]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

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

2015 Launch Records

In [44]:
```
%sql SELECT substr(Date,0,5), substr(Date, 6, 2),Booster_Version, Launch_Site, Payload, PAYLOAD_MASS__KG_, Mission_Outcome,
Landing_Outcome FROM SPACEXTBL WHERE substr(Date,0,5) like '2015' AND Landing_Outcome LIKE 'Failure (drone ship)';
```

Out[44]:

substr(Date,0,5)	substr(Date, 6, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing_Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

- We used a combinations of the WHERE, LIKE, AND, to filter and get launch site names for year 2015

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

In [46]:
```
%sql SELECT * FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success%' AND (Date BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY Date DESC;
```

Out[46]:

Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

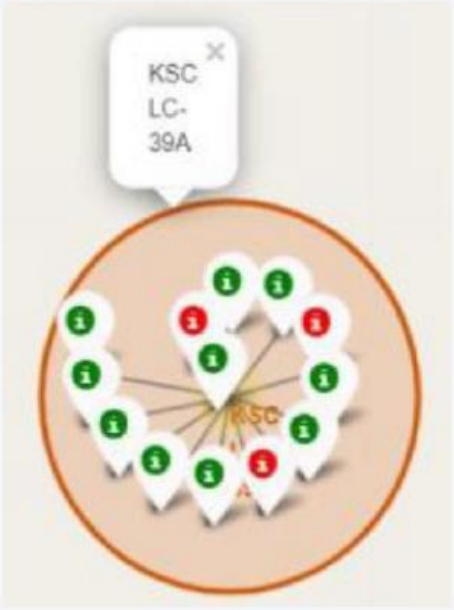
Launch Sites Proximities Analysis

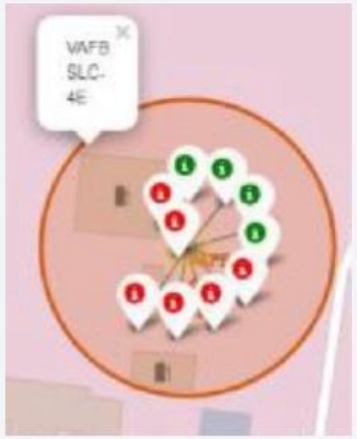
Launch sites global map

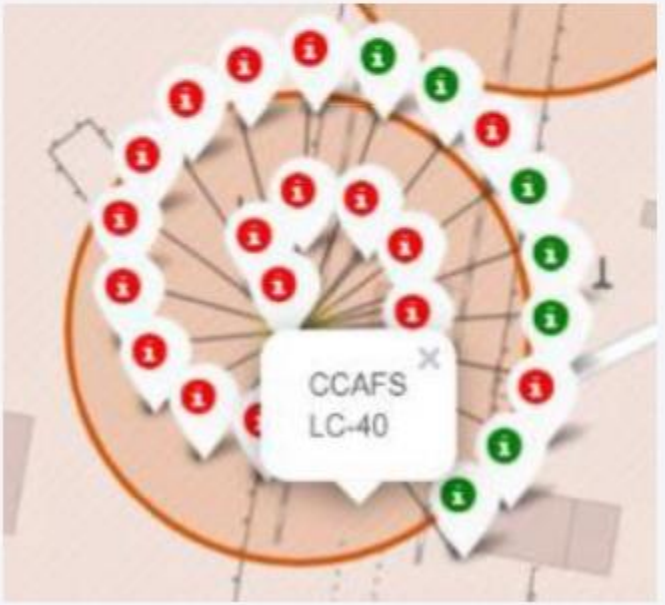
- We can see that the Space launch sites are in the United States of America in Florida and California

Launch sites with color labels

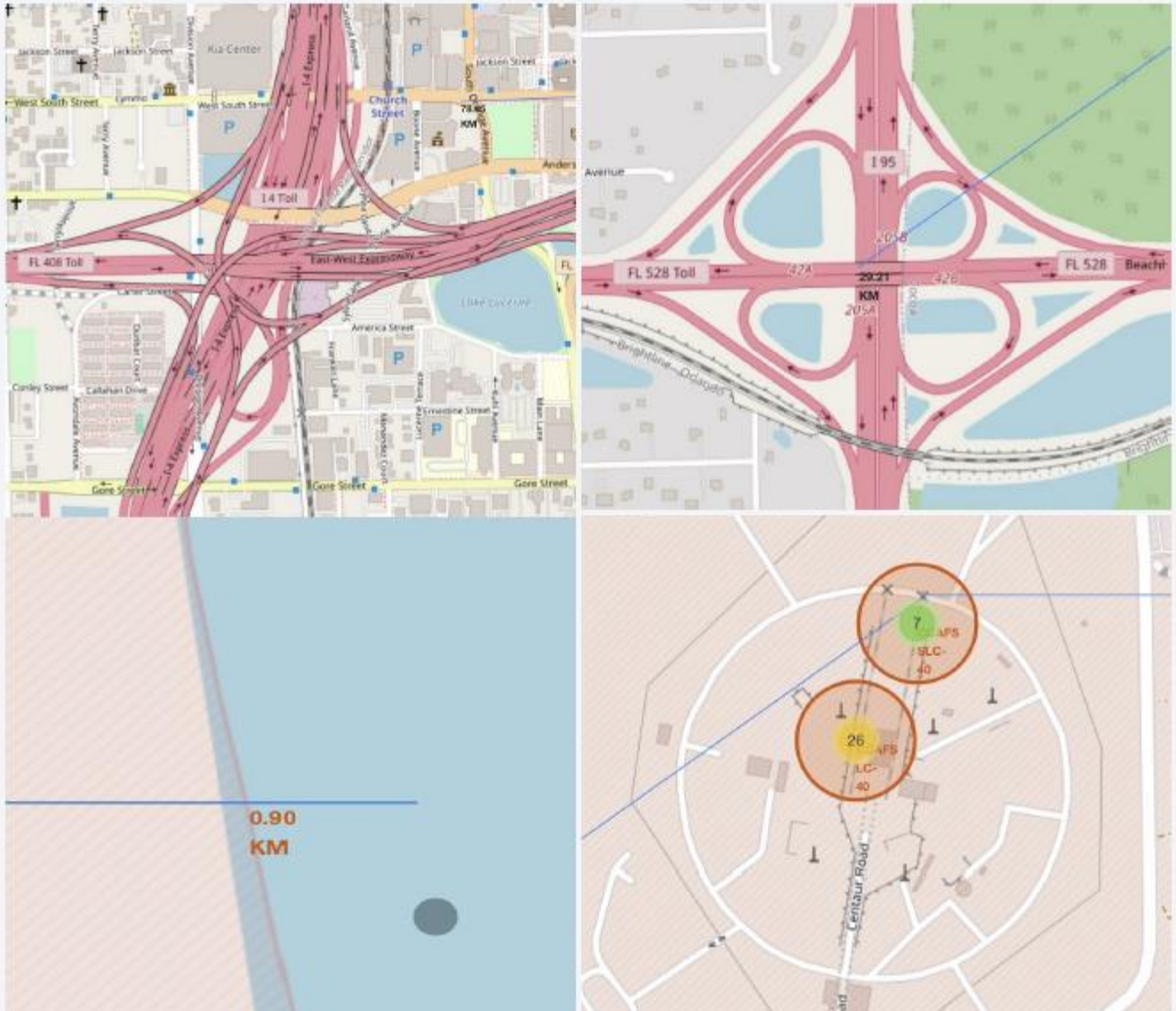
- GREEN MARKER shows **Successful** Launches
- RED MARKER shows **Unsuccessful** Launches







Launch Site distance to landmarks



- Are lunch sites close proximity to railways?

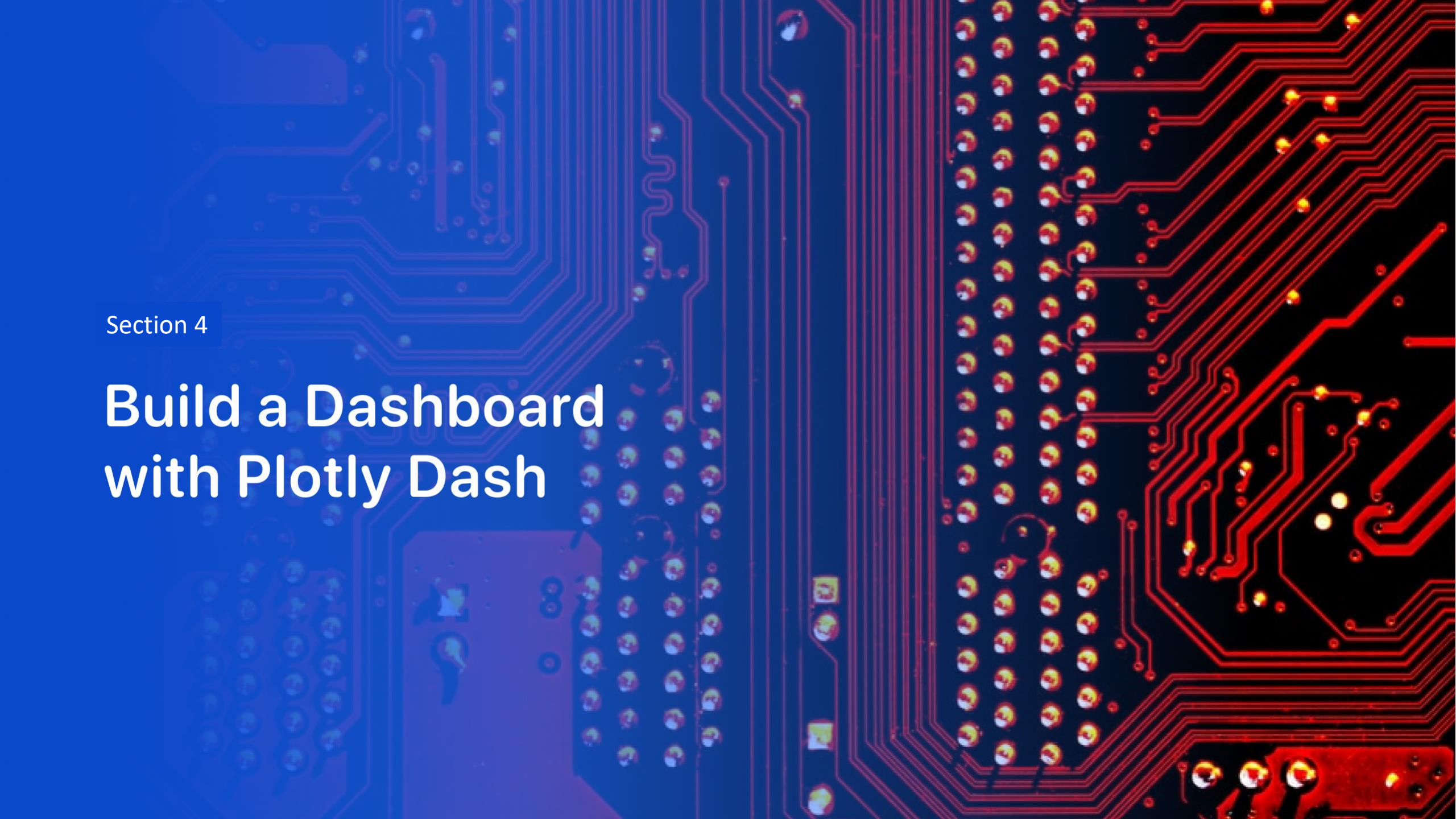
No

- Are lunch sites is close proximity to highway? **No**

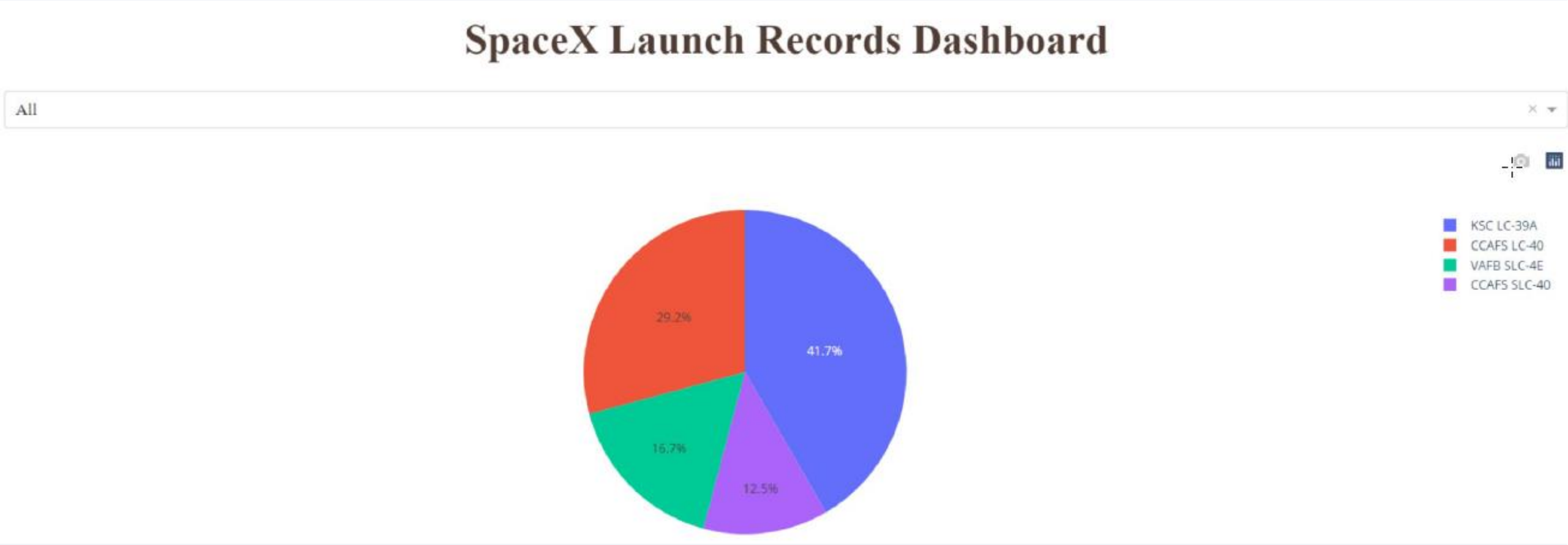
- Are lunch sites close proximity to coastline?

Yes

Do launch sites keep certain distance away from cities? **Yes**

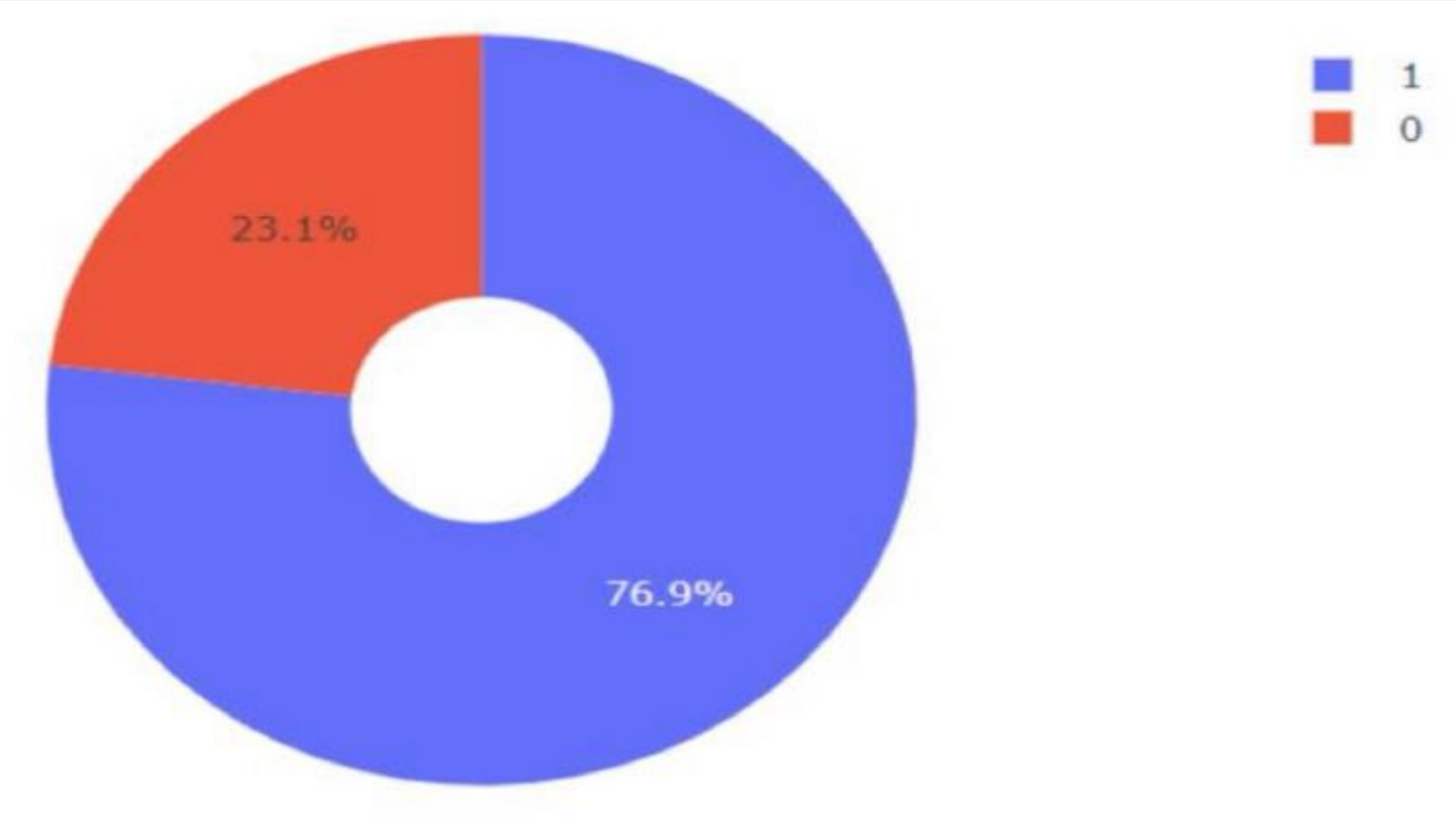

Section 4
Build a Dashboard with Plotly Dash

Success achieved by each launch site



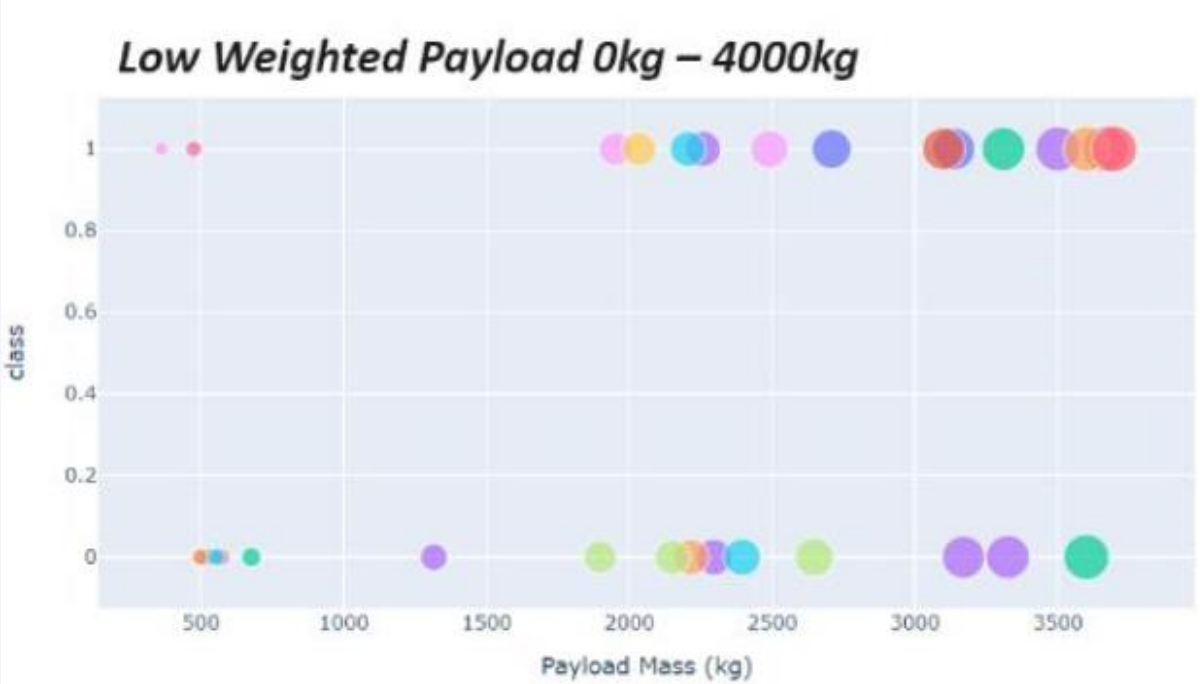
We can see that KSC LC-39A had the most successful launches from all the sites.

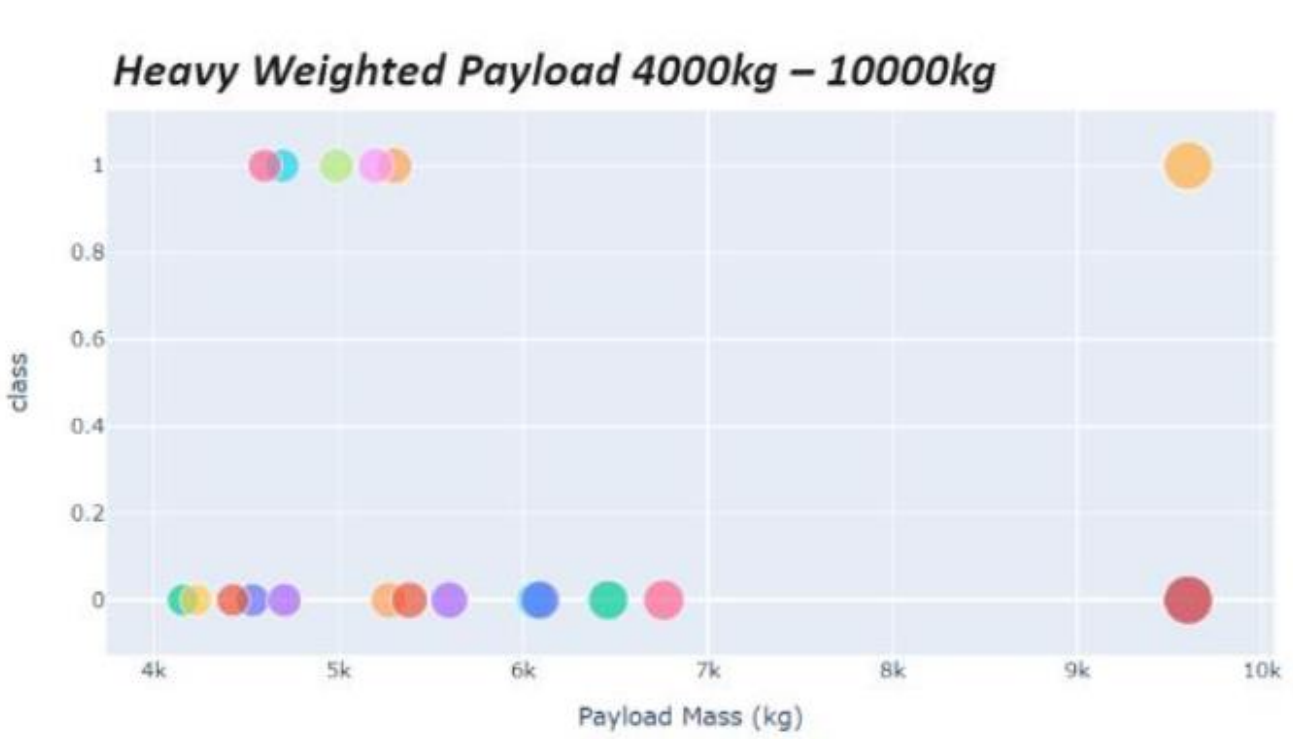
Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate the highest from all sites

Payload vs Launch Outcome





Success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

In [36]:
```
algo_score = {'Logistic regresssion': [logreg_cv.best_score_], 'SVM': [svm_cv.best_score_], 'Decision tree':[tree_cv.best_score_], 'KNN': [knn_cv.best_score_]}
df = pd.DataFrame.from_dict(algo_score, orient='index', columns=['Best scores'])
df
```

Out[36]:

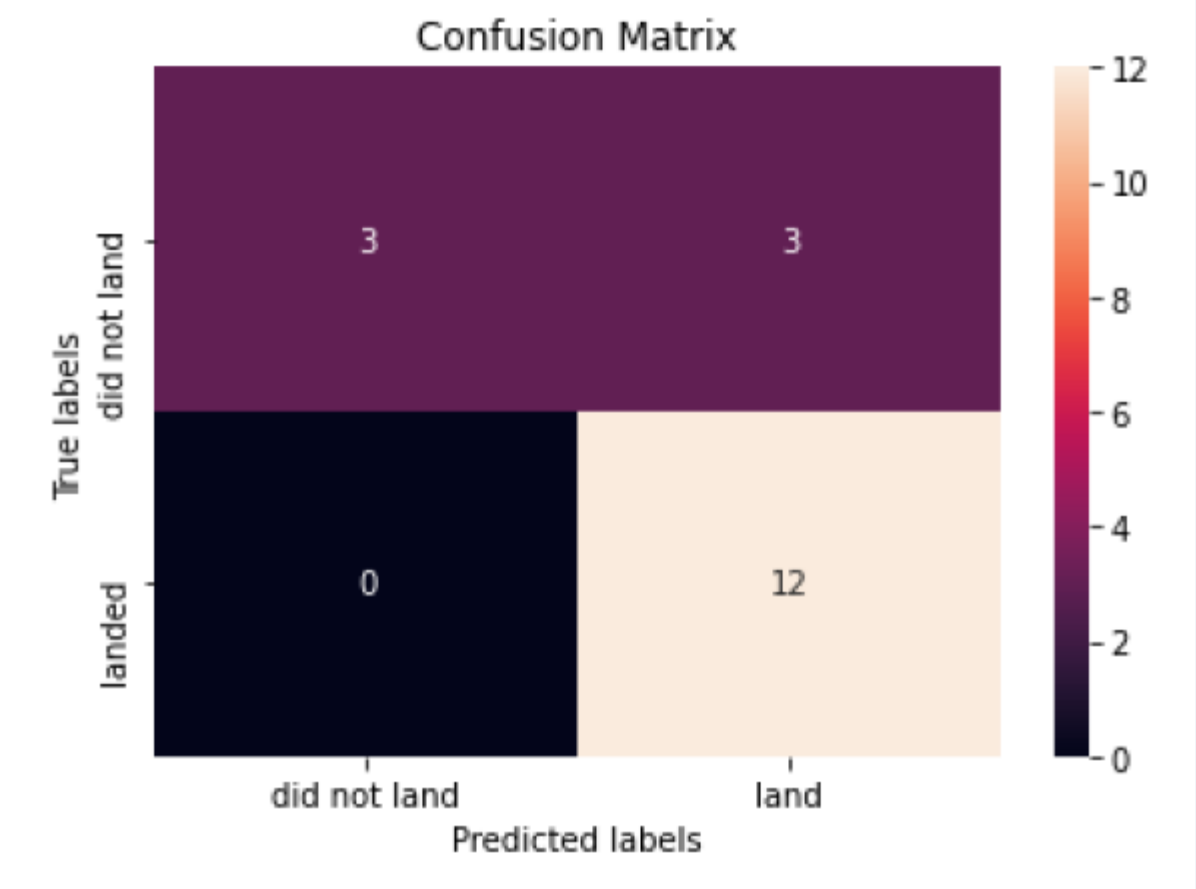
	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.876786
KNN	0.848214

Putting the results of all 4 models side by side, we can see that they all share the same accuracy score and confusion matrix when tested on the test set.

Based on the best scores, the models are ranked in the following order with the first being the best and the last one being the worst:

1. Decision tree
2. K nearest neighbors, KNN
3. Support vector machine, SVM
4. Logistic regression

Confusion Matrix



The confusion matrix for the decision tree classifier indicates that it is capable of differentiating between the various classes.

However, the primary issue lies in the false positives, where unsuccessful landings are incorrectly classified as successful.

Conclusions

- A higher number of flights at a launch site correlates with an increased success rate at that site.
- The launch success rate began to rise steadily from 2013 to 2020.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO experienced the highest success rates.
- KSC LC-39A had the most successful launches among all sites.
- The Decision Tree classifier proved to be the most effective machine learning algorithm for this project

Appendix

- You can find all SpaceX project notebooks and files on GitHub

https://github.com/amgfigueiredo

Thank you!