IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chan Weng Howe
26-Mar-2022

Outline

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

Executive Summary

This project uses SpaceX launch data to predict if the Falcon 9 will land successfully. The data were collected through SpaceX API and scraped from Wikipedia. Then, data wrangling is done to make sure the missing values are resolved so that the data can be analyzed further in exploratory data analysis. EDA was performed in two ways, one with SQL and one with data visualization, both ways allow us to understand more about the data. Next, interactive map were built using Folium in Python for proximity analysis and a Plotly Dash dashboard was developed that consists of interactive elements for analyzing the data. Lastly, predictive analysis or more specifically classification was done using four algorithms.

At the last part of this presentation, the results from EDA are explained, the interactive analytics are demonstrated in screenshots and the predictive analysis results are shown and discussed.

Introduction

Project Background

Advancement of the commercial space age making space travel affordable to everyone. Being one of the most successful company, SpaceX able to make rocket launches relative in expensive by reusing the first stage, spending 62 million dollars instead of 165 million dollars by others. Thus, the ability to determine if the first stage will land, will determine the cost of a launch as well, where this information will be important for potential competitors in competing against SpaceX.

Problem:

- How to predict if the Falcon 9 first stage will land successfully?
- What's the factor that would influence the landing outcome of the rocket?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - using SpaceX API and Web Scraping from Wikipedia
- Perform data wrangling
  - Resolving the missing values, transform outcome to binary label (0,1), one-hot encoding
- 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

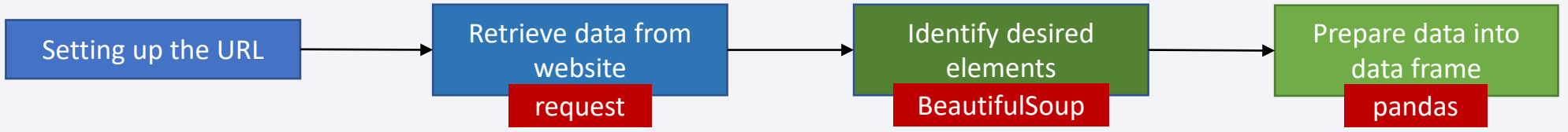
- In this project, the data collection is done using two methods:

METHOD 1: Using SpaceX API

- This method uses `requests` library in Python to retrieve data from the SpaceX API.
- In general, data was retrieved (in JSON) from the API through the prefix: api.spacexdata.com/v4/ and then processed and prepared into data frame.

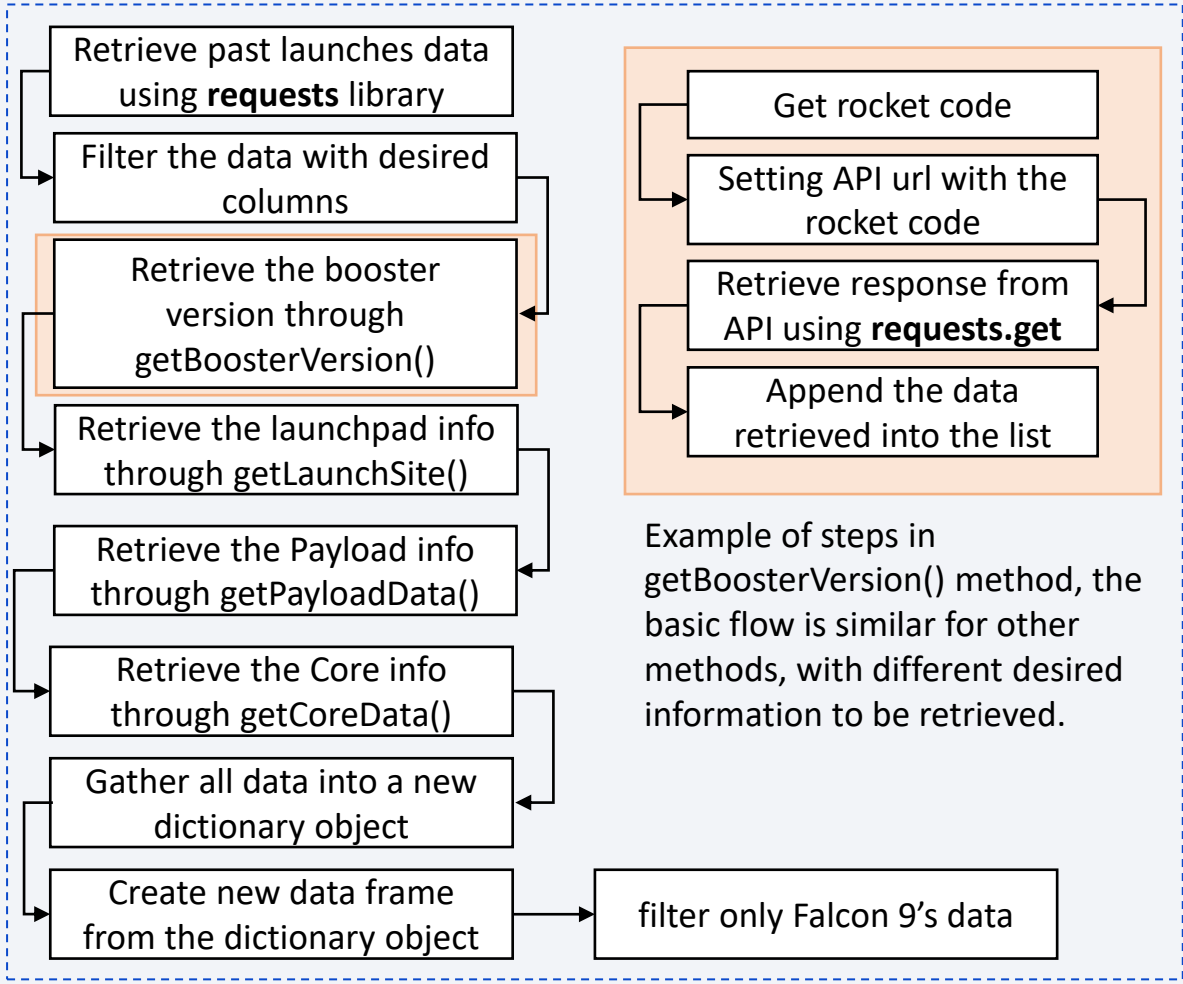
METHOD 2: Using Web Scraping

- This method uses `requests` library in Python to retrieve the data from web pages and processed using `BeautifulSoup` library into more structured forms, and lastly the desired data are transformed into data frame.



Data Collection – SpaceX API

- Information of the rocket launches (specifically for Falcon 9) was collected using SpaceX API
- The initial history data of launches are gathered; however, these data contains only the IDs, thus, based on the retrieved data, further details are extracted through following API URLs:
  - https://api.spacexdata.com/v4/cores/
  - https://api.spacexdata.com/v4/payloads/
  - https://api.spacexdata.com/v4/rockets/
  - https://api.spacexdata.com/v4/launchpads/
- The retrieved data are processed into data frame for further analysis



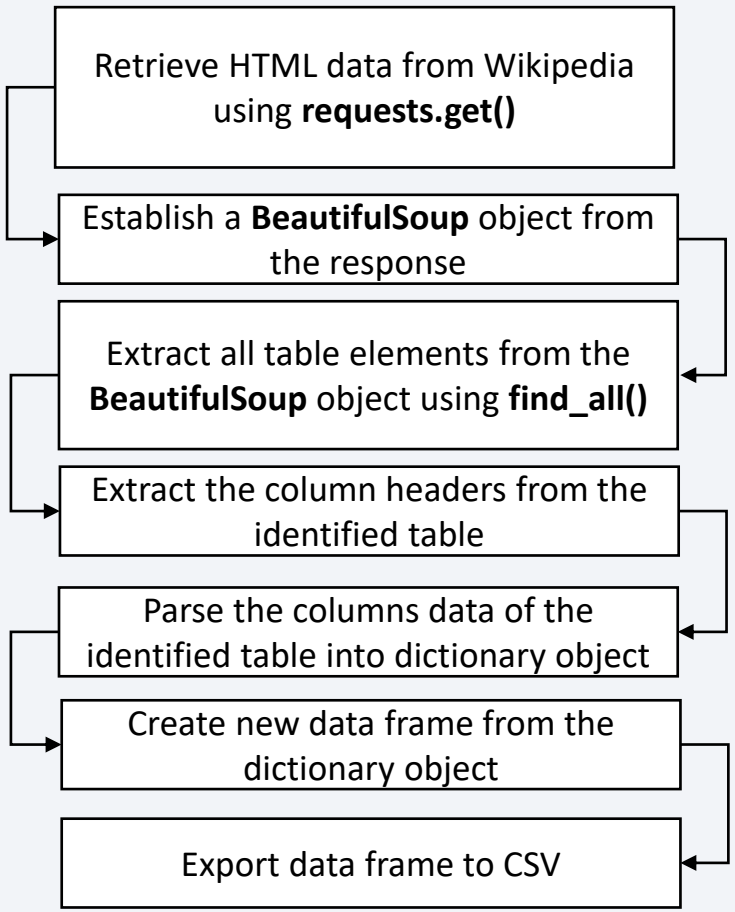
Details step please refer to the notebook in the provided GitHub link

[GitHub link to notebook]

[Watson Studio link to notebook]

Data Collection - Scraping

- In this project, the data are also collected by scraping from the web, in this case from the Wikipedia.
- Basically, the steps involved are:
  1. Retrieving from the defined URL using `requests.get`
  2. Then extract the desired HTML element using `BeautifulSoup`
  3. Then, parsing the extracted HTML element into data frame



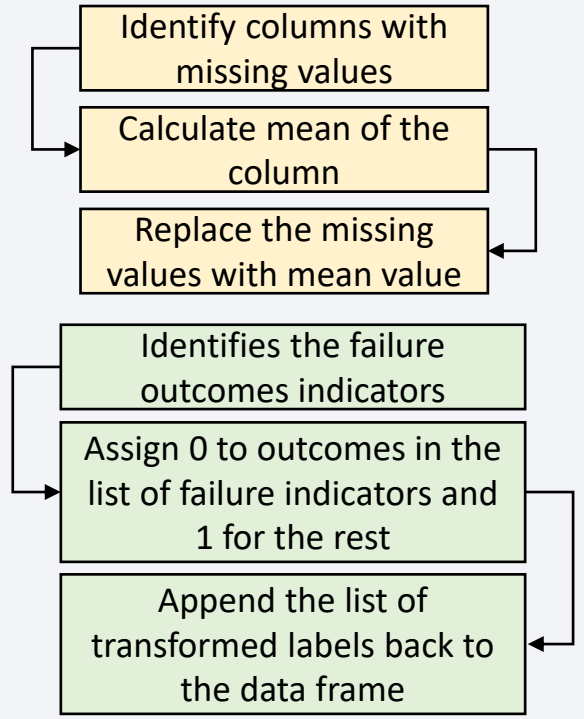
Details step please refer to the notebook in the provided GitHub link

[GitHub link to notebook]

[Watson studio link to notebook]

Data Wrangling

- Several processing were done on the data:
  - Resolve the missing values
    - Existence of missing values were checked using the `isnull()` method on the data frame.
      i.e.: the missing payload mass information were replaced using the mean value of all the payload mass values available.
  - Determining the training labels from the mission outcome.
    - There are several terms used on the mission outcome even though all of those are between success or failure. Thus, for ease of machine learning analysis and in Visualization, these outcome has been transformed to values of 0 and 1 representing failure and success, respectively.
  - One-hot encoding
    - This is an important process to prepare for machine learning analysis. Categorical values will be transformed into numerical columns using `get_dummies()` method in pandas.



GitHub link to notebook

Watson studio link to notebook

EDA with Data Visualization

GitHub link to notebook

Watson studio link to notebook

Scatter plot

To visualize the relationship between two variables, indicating how one variables affects the another. In this project, additional dimension is added by using colors according to the class (failure/success). Following graphs have been plotted:

- Flight number vs launch site
- Payload mass vs launch site
- Flight number vs orbit type
- Payload mass vs orbit type

Bar plot

To ease the comparison between success rate on different orbit type, a bar plot was used to generate the insight.

Line plot

Line plot is suitable for showing trends, in this project, the trends of success rate for year 2010-2019 was plotted using a line plot.

EDA with SQL

GitHub link to notebook

Several SQL queries were performed to gather different insights from the dataset:

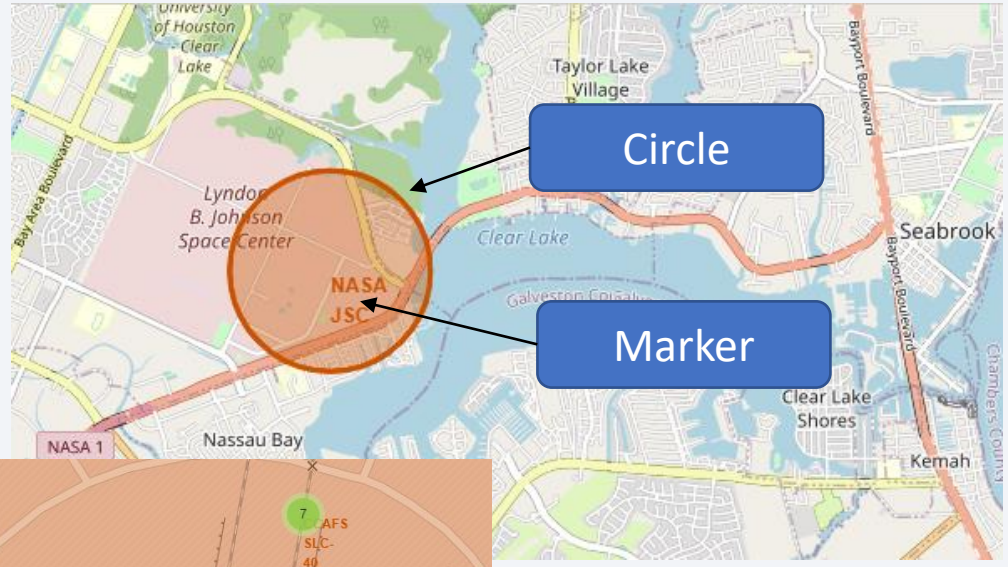
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcome
- List the names of the booster_versions which have carried the maximum payload mass.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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.

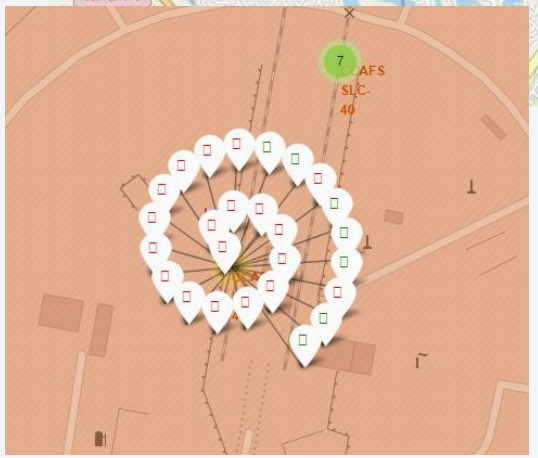
Build an Interactive Map with Folium

GitHub link to notebook

In this project, Folium is used to generate interactive map to visualize the launch sites and the success and failure launches with the use of several map objects:

Circle: to outline the area of interest
Marker: to include icons or text
MarkerCluster: this is used to group multiple markers in the same group, for visualizing the success and failure launches for each site.
PolyLine: this is used to show the distance from the launch site to
MousePosition: to get the coordinator of the mouse position as the mouse is hovering over the map.





Build a Dashboard with Plotly Dash

GitHub link to source code

- The dashboard consists of several components:
  - For interaction, two kinds of interaction components are added:
    - Dropdown menu, which allow users to interact with and choose the desired launch sites or all sites.
    - Slider bar which allow user to select specific range of payload mass.
  - For plots, two kinds of plots are added:
    - A pie chart is used to display the success launches across all sites if user select “All sites” in the dropdown menu. While for other specific sites are chosen, the pie chart will show the success vs failure launches in that site.
    - A scatter plot is used to display the success/failure launches against the payload mass and the data points also coloured according to the booster version, this allow us to see the relationship of the payload mass on the outcome and relationship on different booster version.

Predictive Analysis (Classification)

GitHub link to notebook
Watson studio link to notebook

For predictive analysis, `sklearn` library is used. The steps involved can be summarized as follows:

Building the model
- Loading dataset
- Feature engineering
- Data splitting (training and testing data)
- Determine the ML algorithms
- Define the initial parameters for each algorithm
- Train the model with defined parameters in GridSearchCV

Model evaluation
- Checking best training accuracy (from GridSearchCV)
- Checking the accuracy on testing data
- Interpreting the confusion matrix

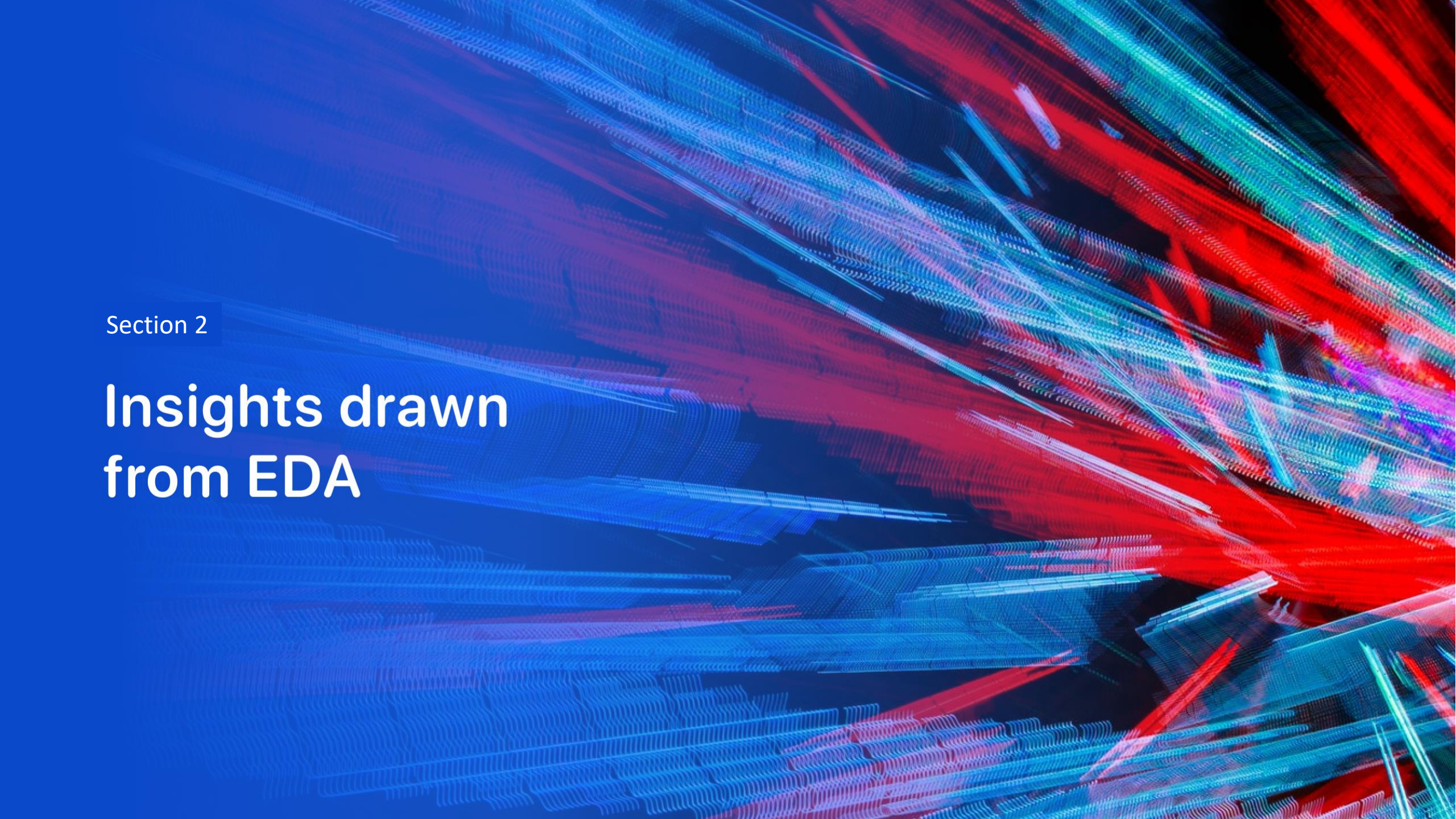
Model refinement
- Through GridSearchCV that identifies best parameters for the model

Finding best performing classification model

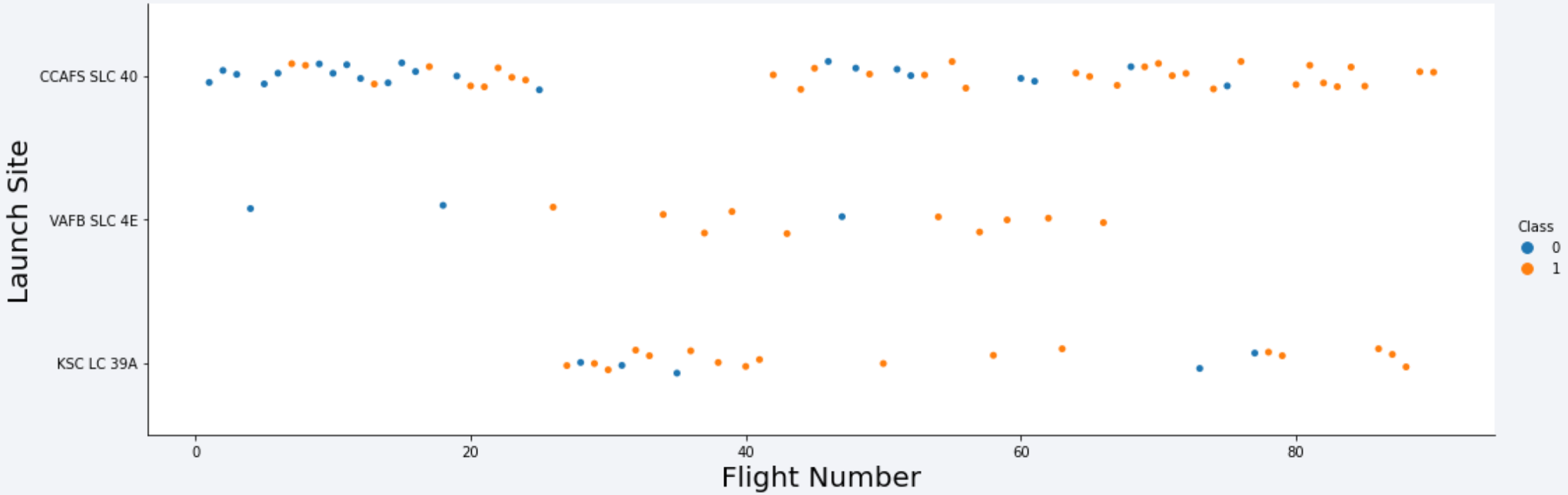
- Compare the performance for each model in a table.

Results

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

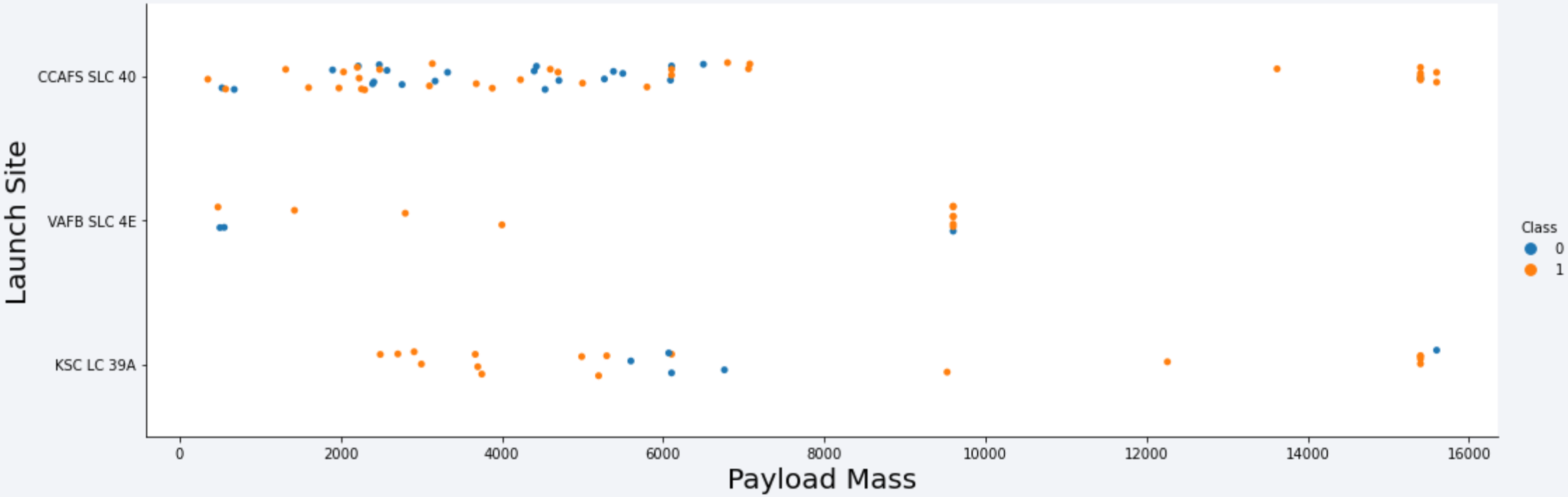

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



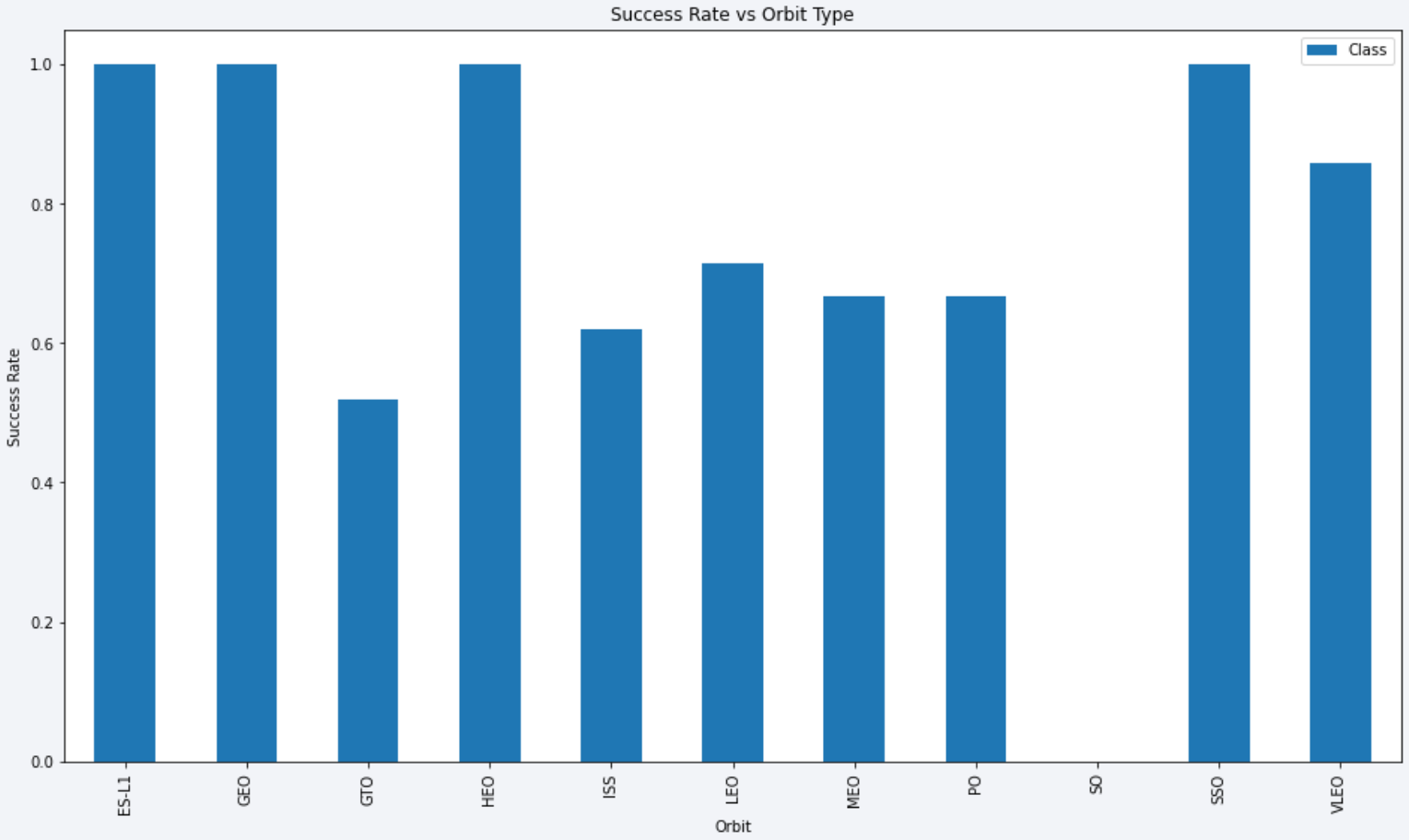
Based on the plot, all sites shows increment of successful launches as the flight number increases

Payload vs. Launch Site



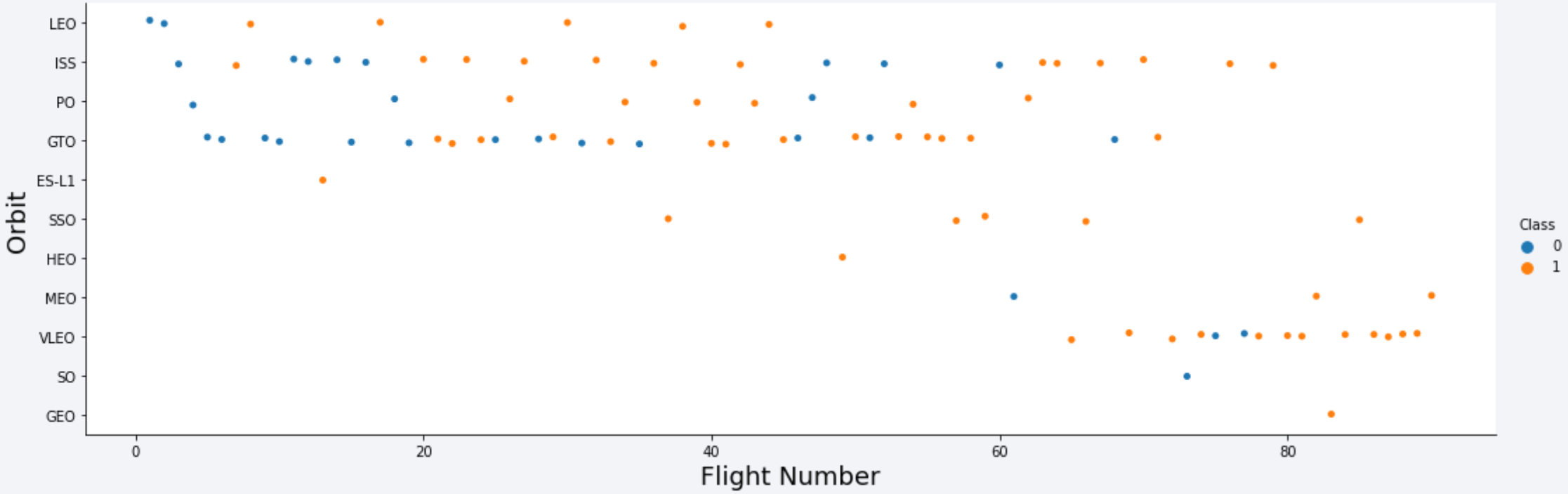
Based on the plot, greater success rate is observed at all sites when the payload mass is larger than 8000kg. Moreover, CCAFS SLC 40 site shows 100% success launches with payload mass larger than 12000kg.

Success Rate vs. Orbit Type



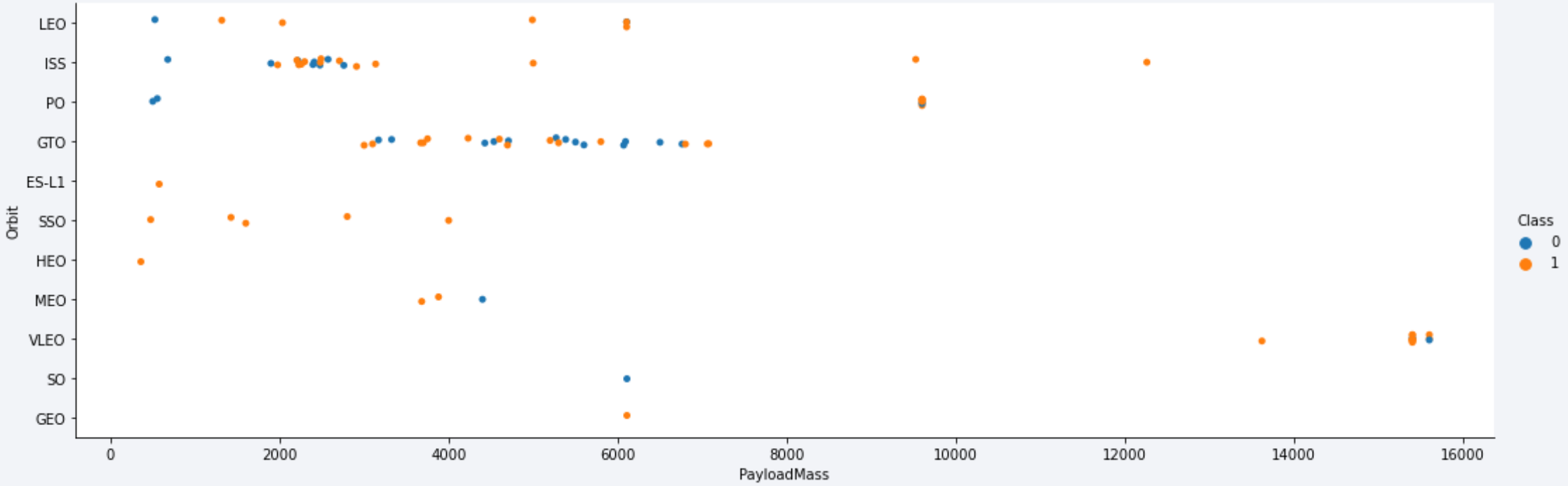
The launches that aimed to orbits of ES-L1, GEO, HEO, SSO have the highest success rate.

Flight Number vs. Orbit Type



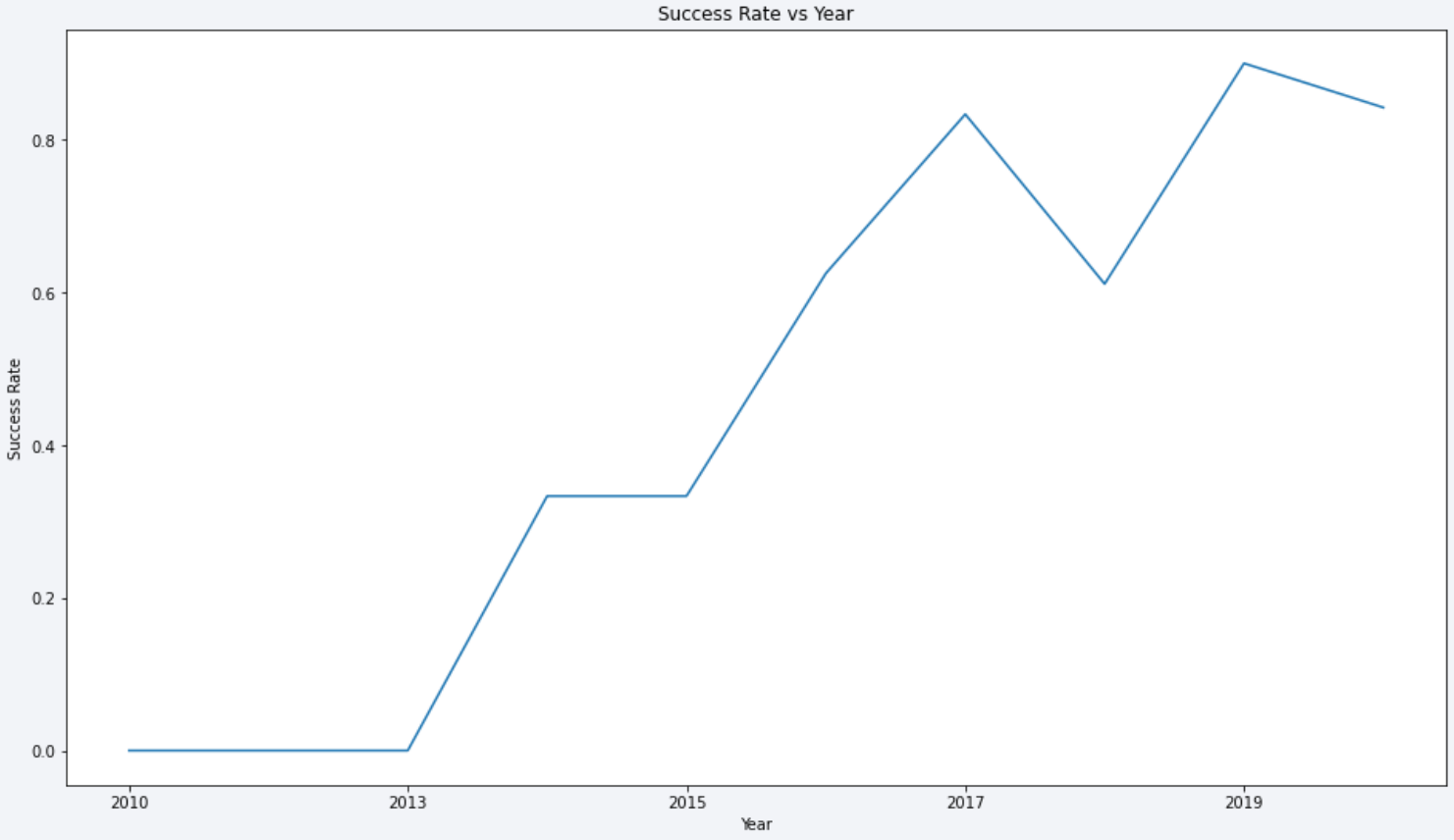
Based on the plot, the number of successful launch as the flight number increases. However, there are still number of failure launches as the flight number increases especially for the launches that target to the orbits ISS, PO, GTO.

Payload vs. Orbit Type



Based on the plot, increases of the payload mass seems to negatively affect the outcome of launches target to orbits ISS and GTO.

Launch Success Yearly Trend



The trends overall shows increment from 2013 till 2020. However, the success rate shows decrement in year 2018 and 2020 .

All Launch Site Names

Find the names of the unique launch sites

```
SELECT DISTINCT(launch_site) FROM SPACEXTBL
```

Using the DISTINCT function to retrieve only unique values from the query from SPACEXTBL table, on the launch_site column

Out[4]:

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

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

```
SELECT *
FROM SPACEXTBL
WHERE launch_site LIKE 'CCA%' LIMIT 5
```

Out[6]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

This query returns all info from SPACEXTBL table with specific condition that must be met for launch_site column. Using the LIKE command to search for specific patterns of string ('CCA%') where % defines the wildcards that possibly come after CCA, and the results were limited using the LIMIT command

Total Payload Mass

Calculate the total payload carried by boosters from NASA

```
SELECT SUM(payload_mass__kg_)
FROM SPACEXTBL
WHERE customer = 'NASA (CRS)'
```

Out[7]:

1
45596

In this query, the SUM command is used to calculate the sum of payload mass returned from the SPACEXTBL with the condition where the customer is equal to ‘NASA (CRS’) as stated in the lab.

Average Payload Mass by F9 v1.1

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

```
SELECT AVG(payload_mass__kg_) FROM SPACEXTBL
WHERE booster_version LIKE 'F9 v1.1'
```

Out[12]:

1
2928

In this query, AVG command is used to retrieve the average value (in this case for the column of payload mass) from the SPACEXTBL table and the specific condition is set at the WHERE clause that only get the data with the booster version of F9 v1.1

First Successful Ground Landing Date

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

```
SELECT min(DATE) FROM SPACEXTBL WHERE
landing__outcome='Success (ground pad)'
```

Out[17]:

1
2015-12-22

In this query, the data of date is queried from SPACEXTBL table with specific condition in WHERE clause that only gets the data with landing outcome of ‘Success (ground pad)’, then the data we interest is the minimum date (a.k.a the earliest date) with the use of MIN command.

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
SELECT booster_version
FROM SPACEXTBL
WHERE landing__outcome='Success (drone ship)'
AND payload_mass__kg_ BETWEEN 4000 AND 6000
```

Out[24]:

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

In this query, the names of boosters are queried from SPACEXTBL table with specific condition in WHERE clause that only gets the data with landing outcome of ‘Success (drone ship)’ and the payload mass is between 4000 and 6000 using the BETWEEN command.

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

```
SELECT mission_outcome,
COUNT(mission_outcome) AS TOTAL
FROM SPACEXTBL
GROUP BY mission_outcome
```

Out[34]:

mission_outcome	total
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

In this query, mission outcome and the count of entry is retrieved from SPACEXTBL, the outcome is group by the mission outcome.

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload mass

```
SELECT booster_version,
FROM SPACEXTBL
WHERE payload_mass__kg_=
(SELECT MAX(payload_mass__kg_)
FROM SPACEXTBL)
```

In this query, the names of boosters are queried from SPACEXTBL table with specific condition in WHERE clause that only gets the data with payload mass that equal to the maximum payload mass in the table (the maximum is obtained in a sub query).

Out[41]:

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

```
SELECT landing__outcome, booster_version, launch_site
FROM SPACEXTBL
WHERE landing__outcome='Failure (drone ship)'
AND YEAR(DATE)=2015
```

Out[49]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

In this query, landing outcomes, the booster versions and launch site data are queried from SPACEXTBL table with specific condition in WHERE clause that only gets the data with failure in drone ship and in year 2015 (the year is extracted from the date data using the YEAR command)

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

```
SELECT landing__outcome, COUNT(*) as count
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY count DESC
```

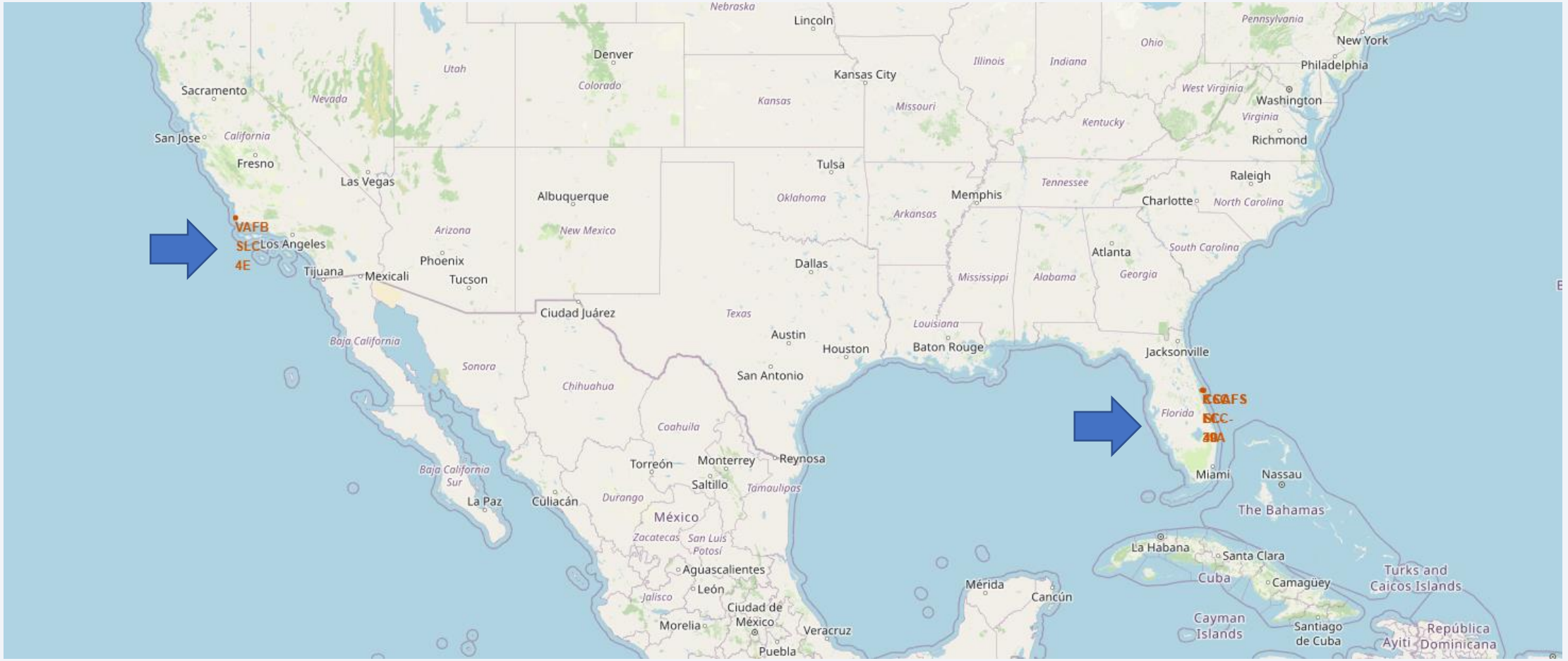
Out[58]:

landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

In this query, the landing outcomes and the count of data is queried from the SPACEXTBL with specific condition in the WHERE clause that only gets data with date between 2010-0604 and 2017-03-02, then group the data based on the landing outcome using GROUP BY command and lastly sort the results using ORDER BY based on the count in descending order using the DESC keyword.

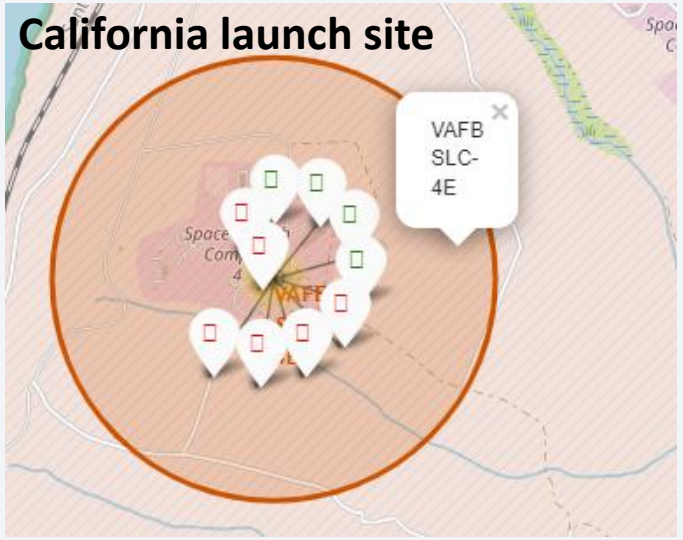

Section 3
Launch Sites
Proximities Analysis

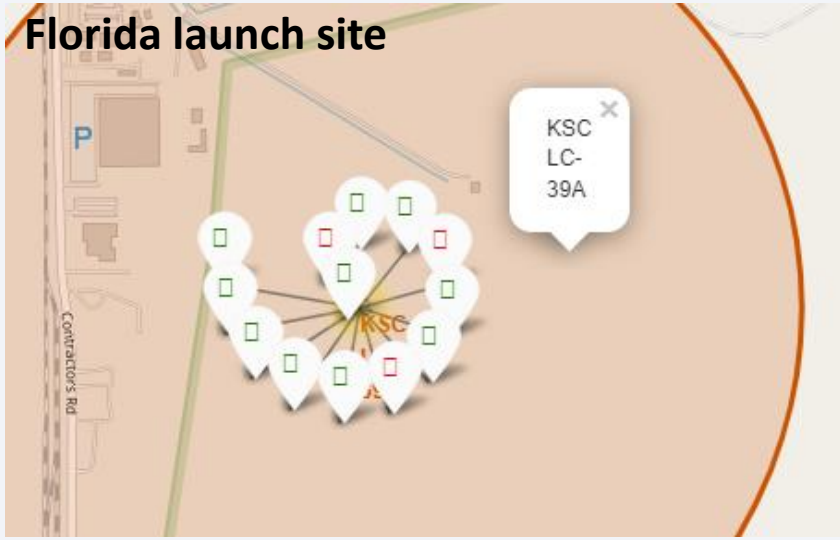
Map of all launch sites

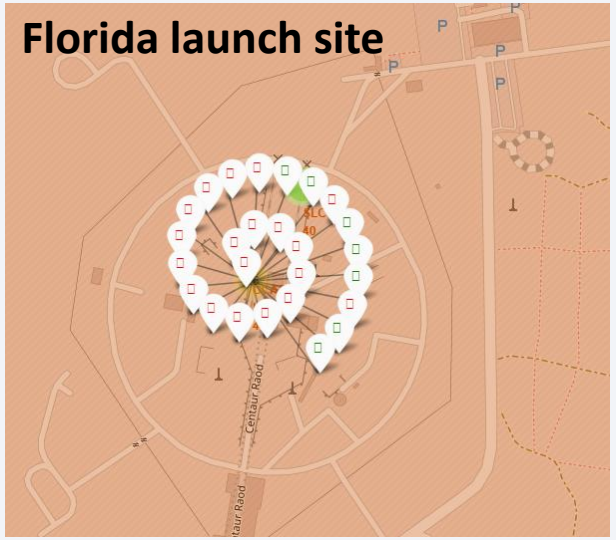


All sites are located at coasts of Florida and California

Launches details for each site





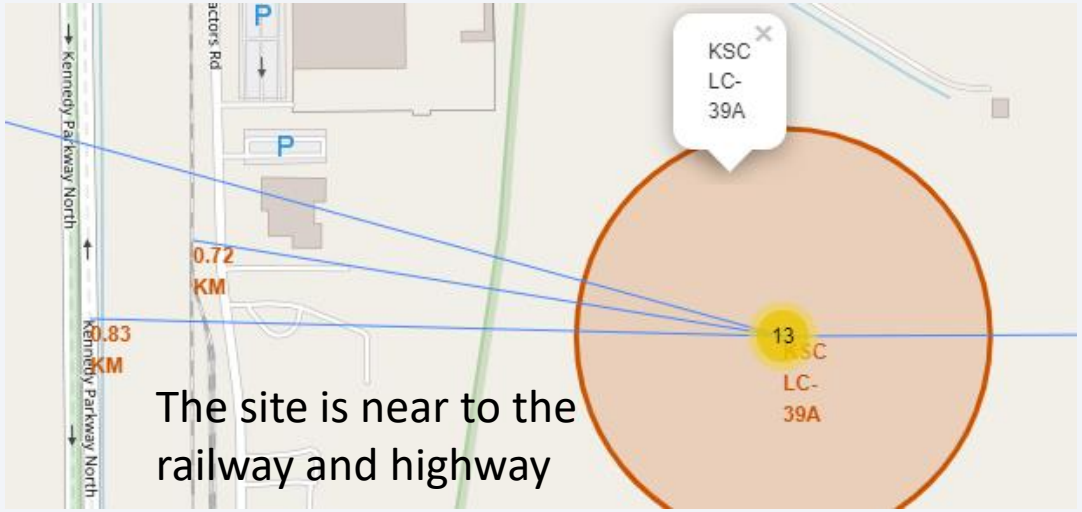


Florida launch site

The success and failure launches for all sites are visualized using **MarkerCluster** that contains multiple markers correspond to each success or failure launch. Each of these is represented in color, Green and Red for success and failure respectively.

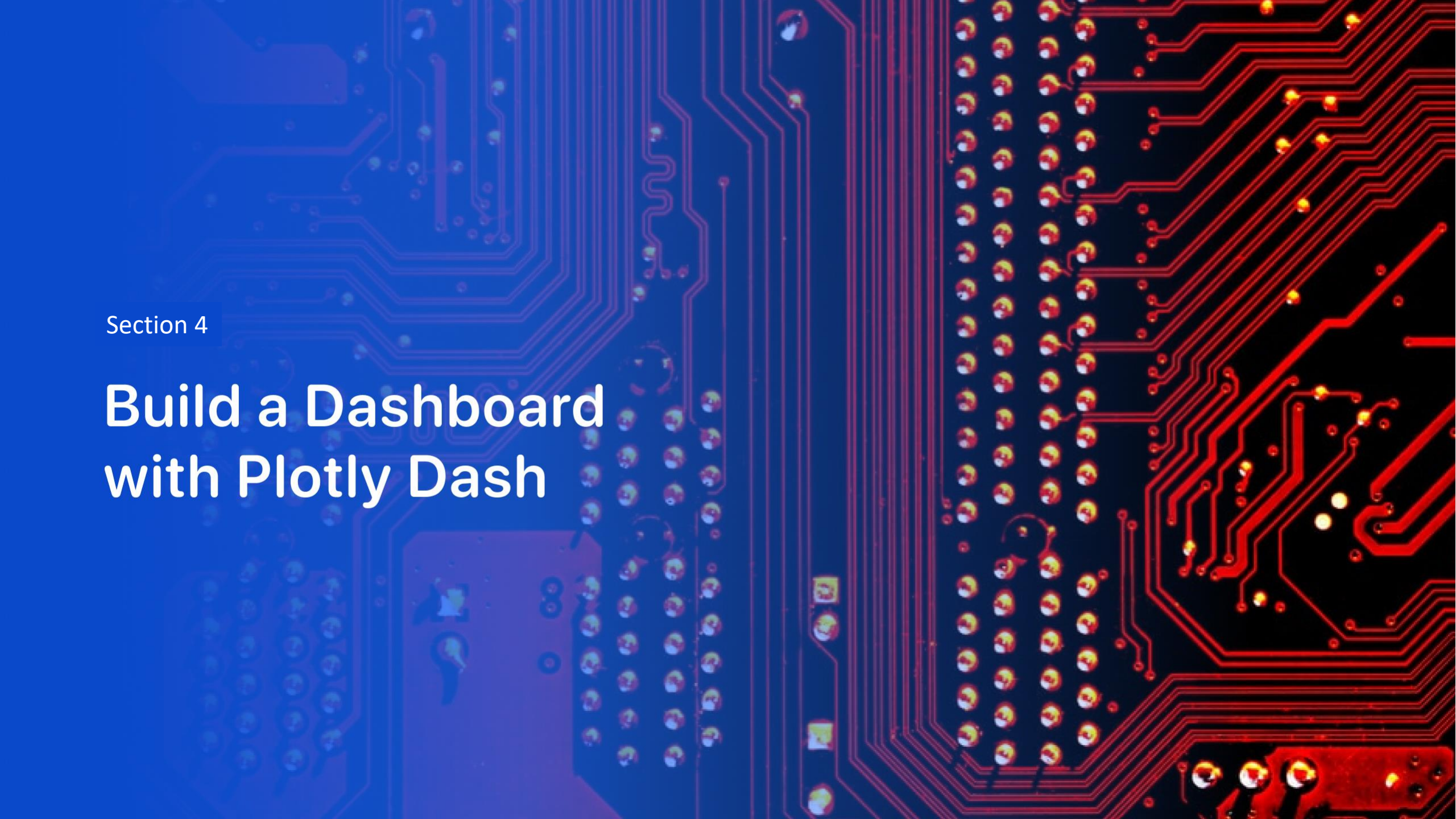
Proximity analysis (using KSC LC-39A as case)

The site is relatively close to coastline

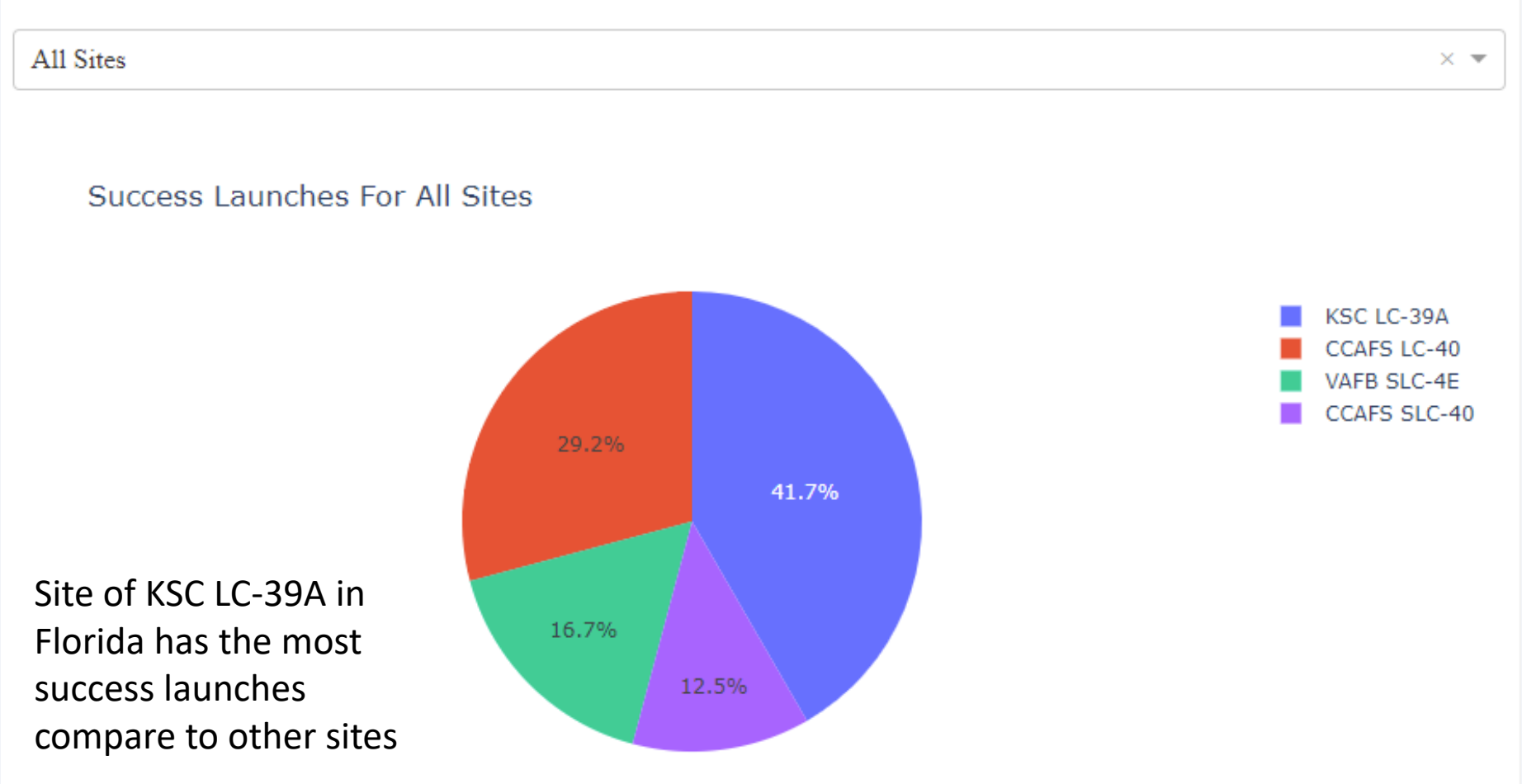


The site is near to the railway and highway

The site is not too far from a nearby city


Section 4
Build a Dashboard with Plotly Dash

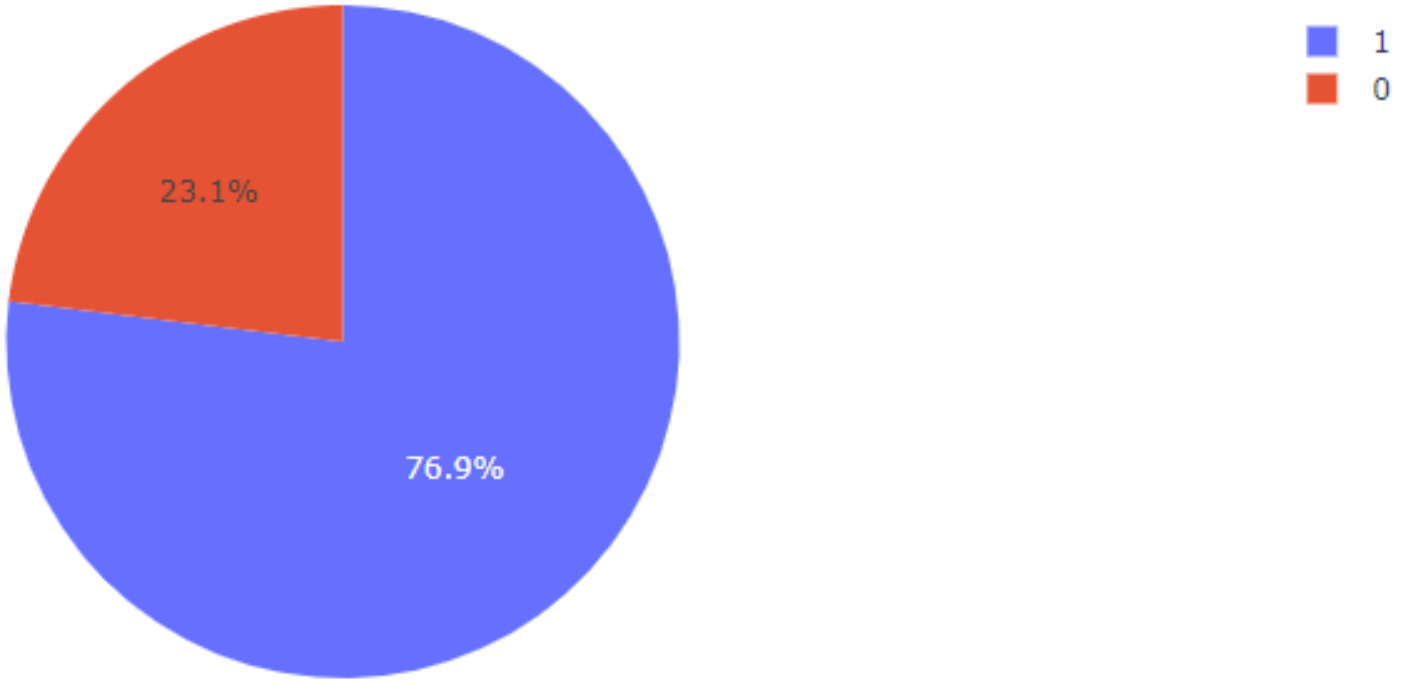
DASHBOARD: Launch success rate for all sites



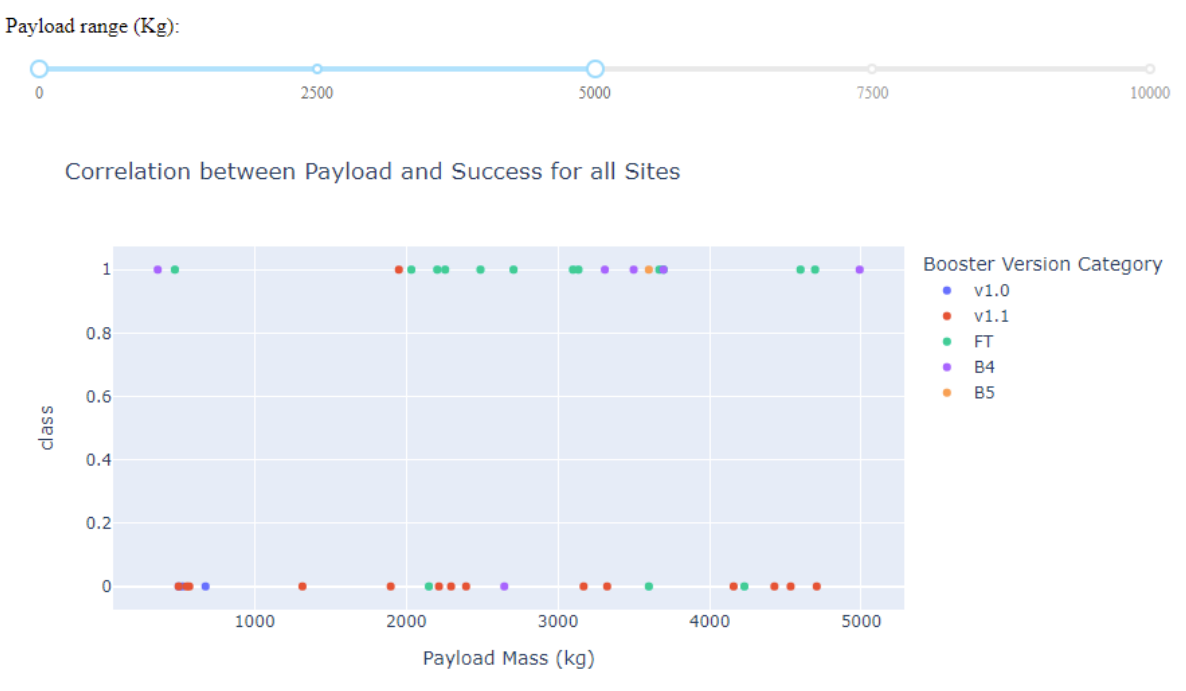
Site of KSC LC-39A in Florida has the most success launches compare to other sites

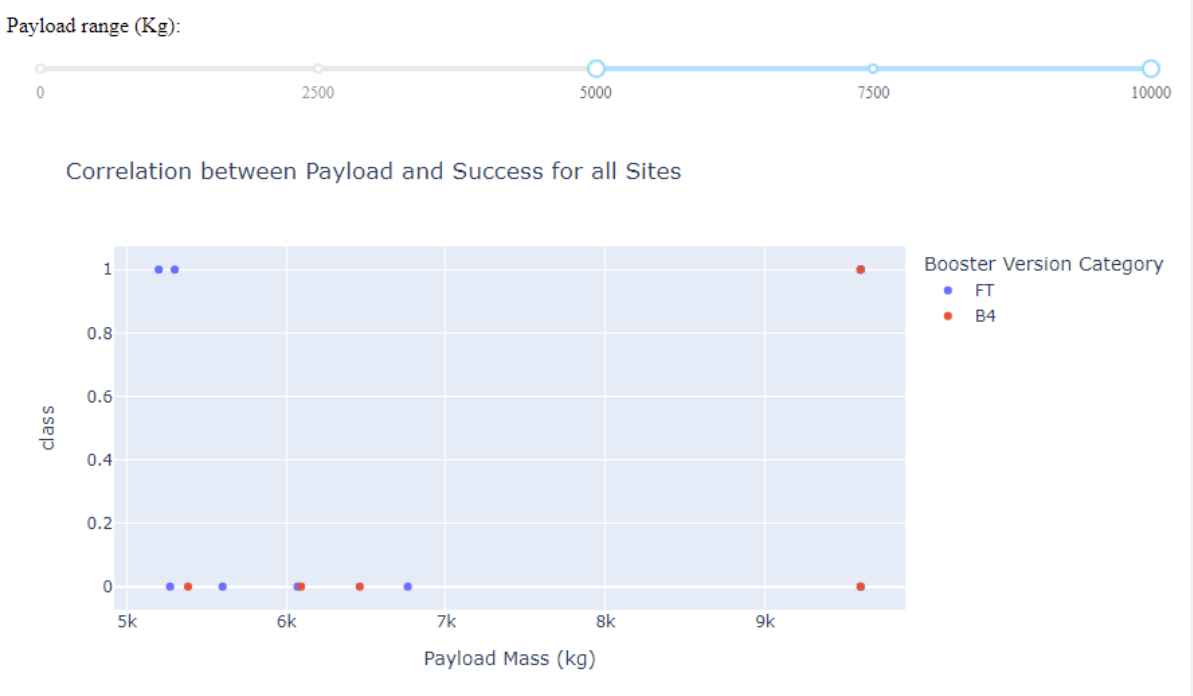
DASHBOARD: Pie chart for the launch site with highest launch success ratio

Among all launches at the site KSC LC-39A, there are 23.1% failure launches.



DASHBOARD: Payload vs Launch Outcome



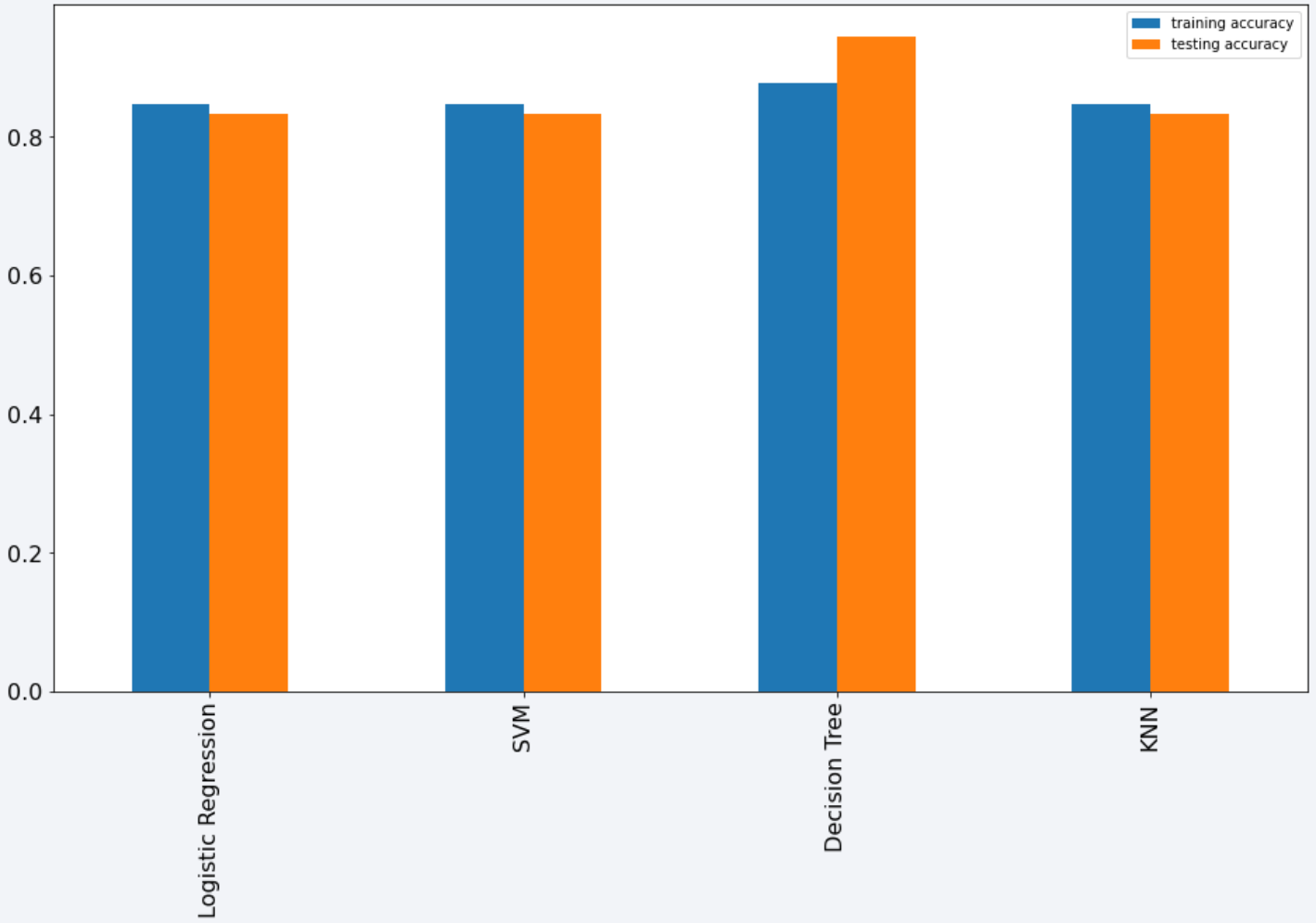


1. For the payload mass that above 5000 kg, only few booster version involved and with very small success rate.
2. For the payload mass that below 5000 kg, success rate is obviously higher (especially for booster version “FT”)

Section 5

Predictive Analysis (Classification)

Classification Accuracy



Out[197]:

	training_accuracy	testing_accuracy
Logistic Regression	0.846429	0.833333
SVM	0.848214	0.833333
Decision Tree	0.876786	0.944444
KNN	0.848214	0.833333

Figure shows the comparison between training accuracy and testing accuracy for all models

The best algorithm here is **Decision Trees**.

GridSearchCV is run for each model with CV=10 to identify the best model. For Decision Tree is as follow:

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1,
'min_samples_split': 2, 'splitter': 'best'}
accuracy : 0.8767857142857143
```

Confusion Matrix

As shown in the confusion matrix of the decision tree model, this model able to perform very well and able to identify almost all positive and negative classes accurately. Only one false positive in this case.

Conclusions

- The site of KSC LC 39A has the greatest number of success launches.
- The outcome of the launches is negatively affected by the payload mass.
- The launches that aimed for the orbits of ES-L1, GEO, HEO, SSO have the highest success rate.
- For predictive analytics, Decision Tree is the best performing algorithm for the dataset in this project.

Thank you!