IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Eleftherios Nikolopoulos
2024-06-07

Outline

Executive Summary

Methodology Summary

- Collecting data via Space X REST API.
- Collecting data from Wikipedia using the method of web scraping
- Data wrangling in order to transform raw data into a more processed shape.
- Analyzing data (Exploratory Data Analysis) by using both SQL and Pandas, Matplotlib for visualization
- Using Folium and Plotly in order to create Interactive Visual Analytics and Dashboard so that users can find visual patterns faster and more effectively.
- The employment of Predictive Analysis in order to build a machine learning pipeline and predict if the first stage of the Falcon 9 lands successfully.

Executive Summary

Summary of Results

- Exploratory Data Analysis by using SQL.

The total number of successful and failure mission outcomes are 99 and 1 respectively.

- Exploratory Data Analysis by using Pandas and Matplotlib

Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

- Predictive Analysis

All the methods (Logistic Regression, SVM, Decision Tree, KNN) perform the same accuracy of 0.833333 on the test Data

Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars. Other providers cost upwards of 165 million dollars each.

- The competitive advantage of SpaceX is the fact that it can reuse the first stage.

- A machine learning model will predict whether the first stage will successfully land, in order to determine what attributes are correlated with successful Falcon 9 rocket landings and therefore determine the cost of a launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collection with Space X REST calls
  - Data collection with Web Scraping
- Perform data wrangling
  - Cleaning the data by removing or correcting inaccuracies, inconsistencies and duplicates
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash

Methodology

- Perform predictive analysis using classification models
  - We split data into training and testing data
  - Training data is divided into validation data
  - The models are trained and hyper parameters are selected
  - We display the best parameters and the accuracy on the validation data
  - We calculate the accuracy on the test data

Data Collection

Space X REST API

- We use the SpaceX REST API with the endpoint "api.spacexdata.com/v4/launches/past"
- We perform a get request using the requests library to obtain the past launch data from the API.
- We view the results by calling the .json() method and the response is in the form of a JSON.
- We convert the form of a JSON to a dataframe by using the json_normalize function.

Web Scraping

- We use the Python Beautiful Soup package to web scrape some HTML tables from Wiki pages.
- We parse the data from HTML tables and convert them into a Pandas data frame for further visualization and analysis.

Data Collection – SpaceX API

- Rocket launch data are collected by using SpaceX REST API and making a get request to the API. The response content with data is parsed and decoded as a form of JSON using .json() and turn it into a Pandas dataframe using json_normalize()

- Add the GitHub URL of the completed SpaceX API calls notebook https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/1.SpaceX%20Falcon9%20first%20stage%20landing%20prediction.ipynb

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

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

```
response.status_code
```

```
200
```

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

```
# Use json_normalize meethod to convert the json result into a dataframe
respjson = response.json()
data = pd.json_normalize(respjson)
```

Data Collection – Scraping

- In order to collect Falcon 9 past launch data from Wiki, beautiful soup package is used for parsing HTML files and finally extracting HTML data which are converted into pandas Data Frame structure.

- Add the GitHub URL of the completed web scraping notebook: https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/2.Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia%20.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.

In [5]:
```
# 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`

In [6]:
```
# 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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

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

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

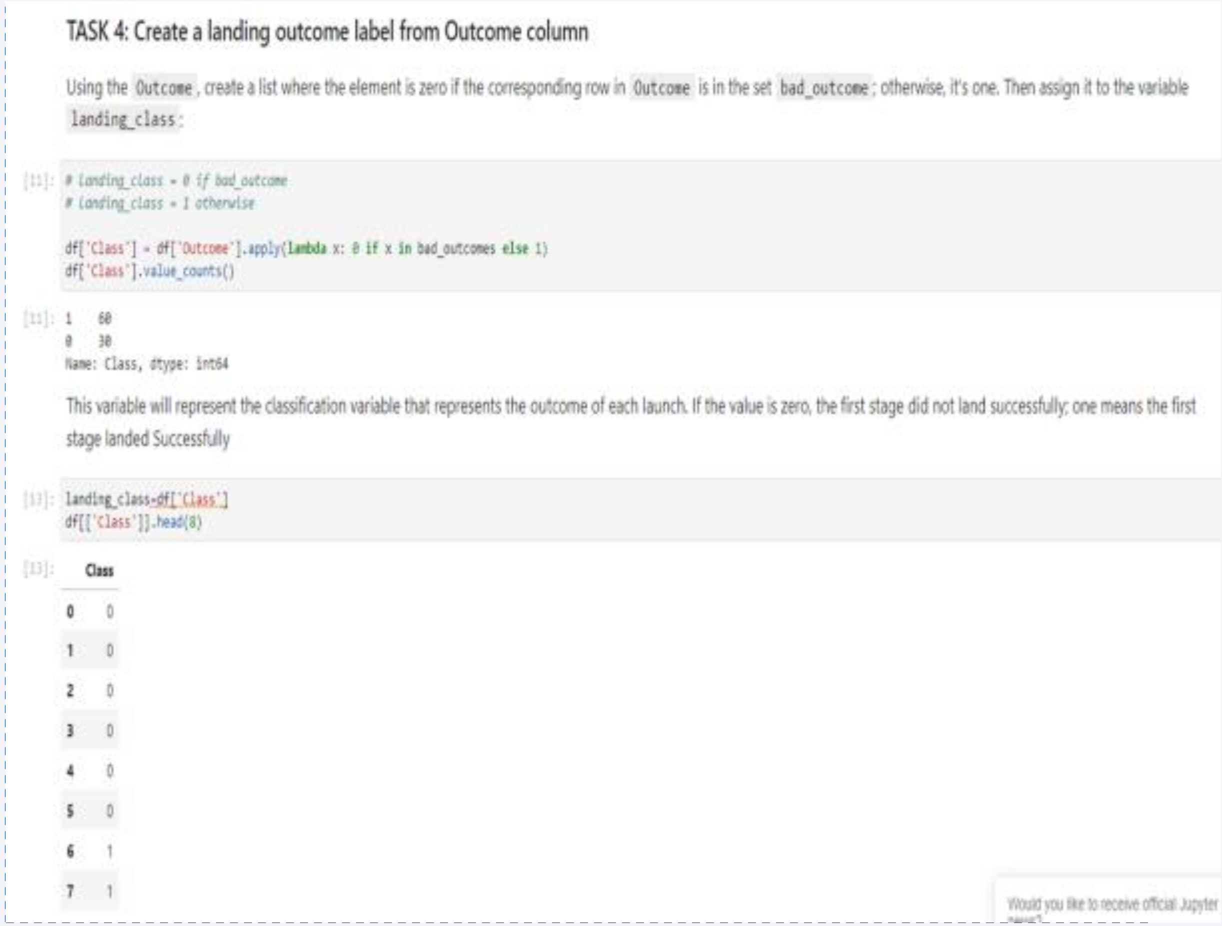
In [10]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

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

Starting from the third table is our target table contains the actual launch records.

In [11]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Data Wrangling

- The data of Booster Version column are filtered to keep falcon 9 launches.
- The missing values of Landing Pad and Pay load Mass columns are dealt with.
- Some Exploratory Data Analysis (EDA) is performed in order to determine the label for training supervised models. The labels are converted into training labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- Add the GitHub URL of your completed data wrangling related notebooks: https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/3.%20Data%20Wrangling.ipynb



EDA with Data Visualization

- **Scatter plot** is used to visualize the relationship between variables such as:
  - flight number and launch site
  - Payload and launch site
  - flight number and orbit type
  - Payload and orbit type
- **Bar chart** is used to visualize the relationship between success rate of each orbit type
- **Line plot** is used to visualize the launch success yearly trend

Add the GitHub URL of your completed EDA with data visualization notebook:
https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/5.%20EDA%20with%20Visualization%20.ipynb

EDA with SQL

SQL queries are performed in order to answer the following questions of the research:

- The names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string “CCA”
- The total payload mass carried by boosters launched by NASA
- The average payload mass carried by booster version F9 v1.1
- The date when the first successful landing outcome in ground pad was achieved
- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- The total number of successful and failure mission outcomes
- The names of the booster versions which have carried the maximum payload mass
- The month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015
- The count of successful landing outcomes between the date 04-06-2010 and 20-03-2017

Add the GitHub URL of your completed EDA with SQL notebook: https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/4.%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- An interactive folium map show all the **launch sites** using **marker cluster** object to display a large number of markers on a map like **color-labeled markers** to identify which launch sites have relatively high success rates.
- Specifically, we create **markers** for all launch records, if a **launch** was **successful** (class=1), then a ***green marker*** is used and if a **launch** was **failed**, a ***red marker*** (class=0) is used
- Additionally, an interactive folium map uses **lines** to explore and analyze the proximities of launch sites such as highways, coastline and cities.
- Building an interactive map aims to discover some preliminary **correlations** between the **launch site** and **success rates** by performing more interactive visual analytics using Folium.

Add the GitHub URL of your completed interactive map with Folium map:
https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/6.%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

In order to build the Dash application, several tasks are performed, such as:

- The addition of a Launch Site ***Drop-down Input Component***
- The addition of a callback function to render ***success-pie-chart*** based on selected site dropdown
- The addition of a ***Range Slider*** to Select Payload
- The addition of a callback function to render the ***success-payload-scatter-chart scatter plot***

Add the GitHub URL of your completed Plotly Dash lab: https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/7.%20An%20Interactive%20Dashboard%20with%20Plotly%20Dash.py

Predictive Analysis (Classification)

❑ Exploratory Data Analysis is performed and Training Labels are determined in order to:

- **create a column for the class**
  - ✓ Class values: 1 and 0 for successful and unsuccessful launch respectively
  - ✓ A NumPy array is created from the column Class in data, by applying the method to_numpy() then assign it to the variable Y

- **standardize the data**
  - ✓ The data are standardized in X then reassign it to the variable X

- **split into training data and test data**
  - ✓ The function train_test_split is used to split the data X and Y into training and test data by setting the parameter test_size to 0.2 and random_state to 2.
  - ✓ The training data is divided into validation data, a second set used for training data

Predictive Analysis (Classification)

- We find the best Hyperparameter for SVM, Classification Trees and Logistic Regression and KNN by creating the following objects:

- a **logistic regression** object and then a **GridSearchCV** object (*logreg_cv*) with cv = 10.
- a **support vector machine** object and then a **GridSearchCV** object (*svm_cv*) with cv =10
- a **decision tree classifier** object and then a **GridSearchCV** object (*tree_cv*) with cv = 10
- a **k nearest neighbors** object and then a **GridSearchCV** object (*knn_cv*) with cv = 10.

- We calculate the accuracy on the test data, using the method "score" for each method (logistic regression, classification tree, SVM, KNN), in order to find the method that performs best

Add the GitHub URL of your completed predictive analysis lab: https://github.com/LeftyN/Nikolopoulos-Lefteris-/blob/5edbb72cda5348ba69096371461a758c46c83206/8.%20Space%20X%20Machine%20Learning%20Prediction.ipynb

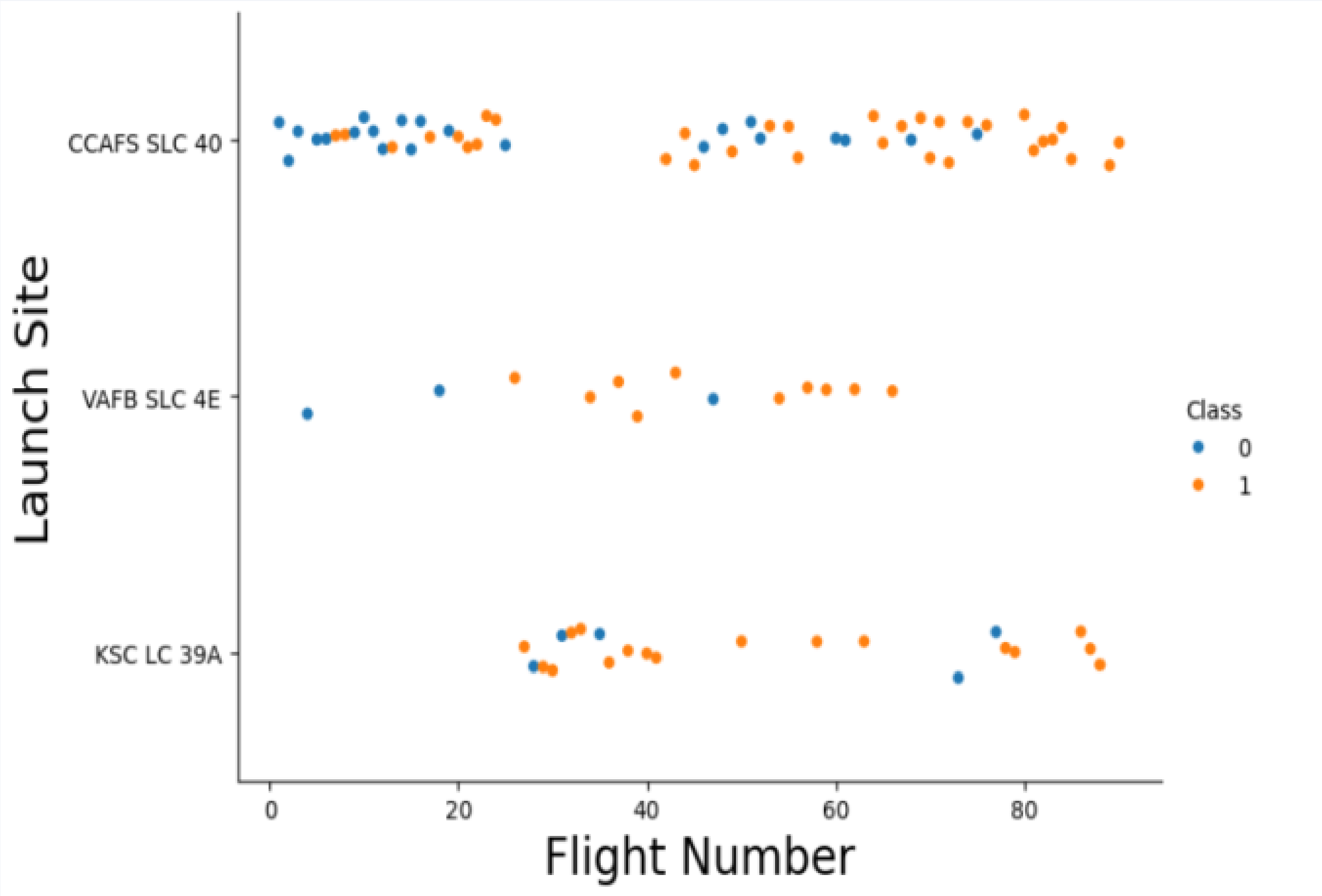
Results

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

Section 2

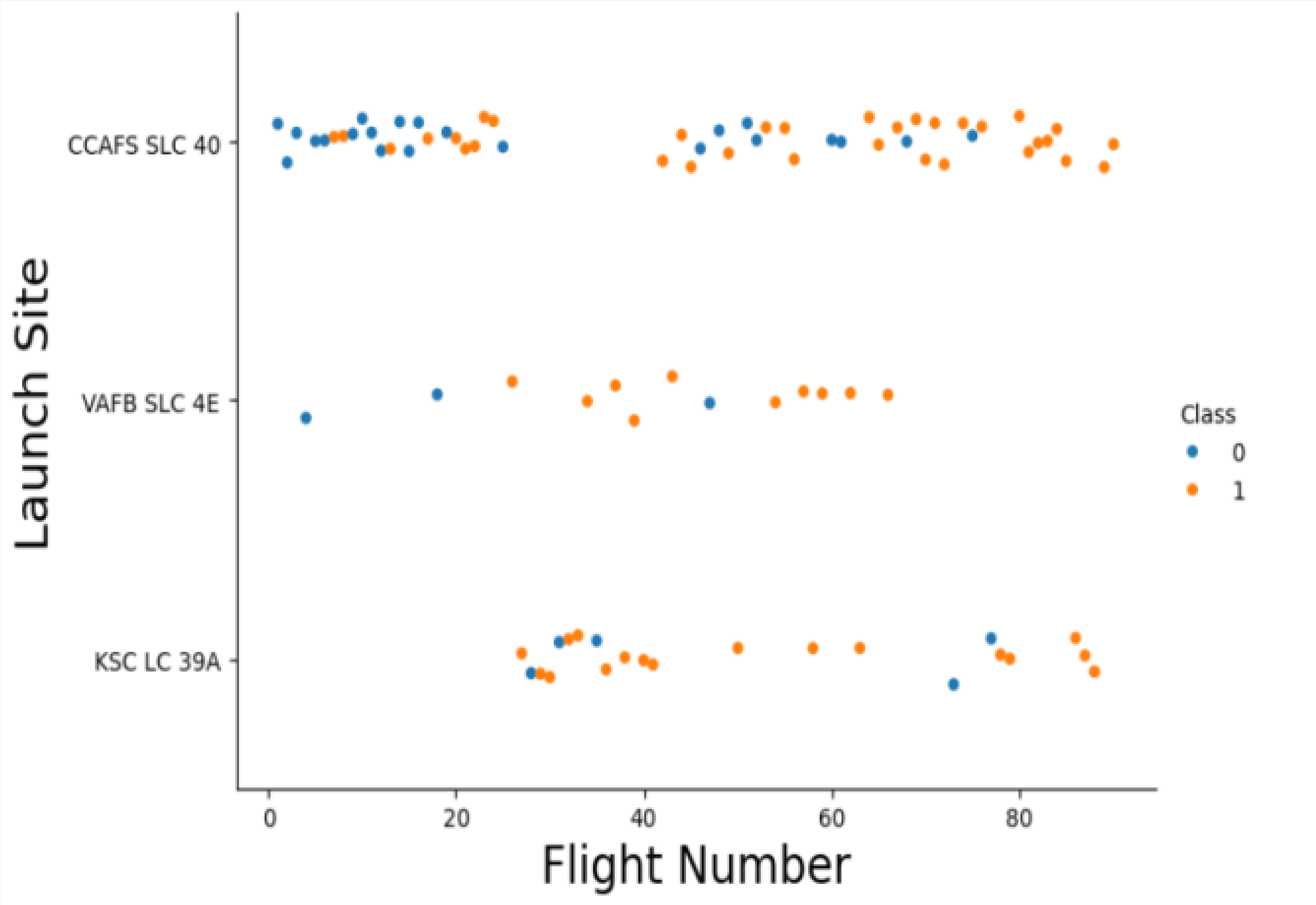
Insights drawn from EDA

Flight Number vs. Launch Site



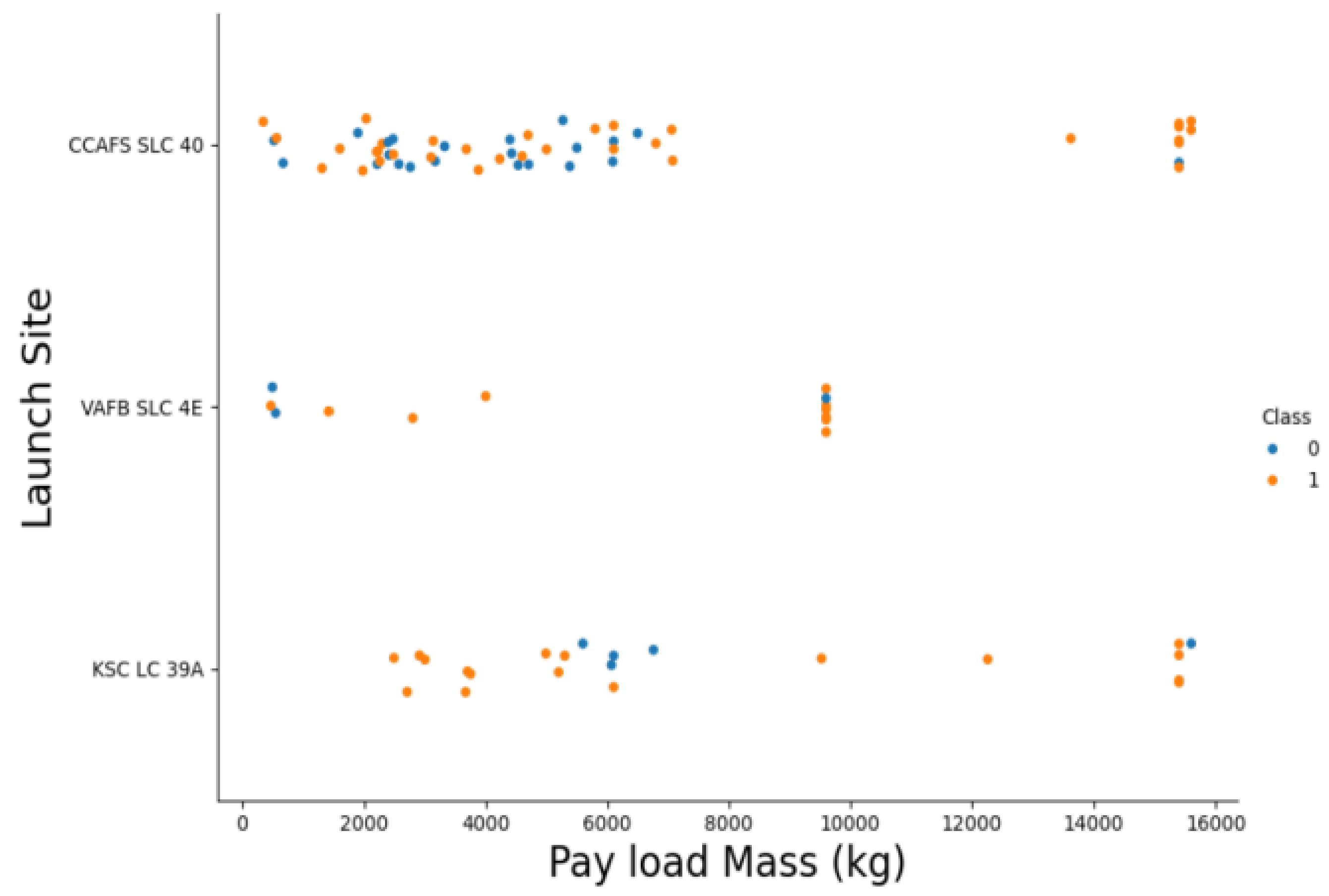
The success (class 1) or the failure (class 0) of a launch outcome in correlation with the launch sites and the flight number

Flight Number vs. Launch Site



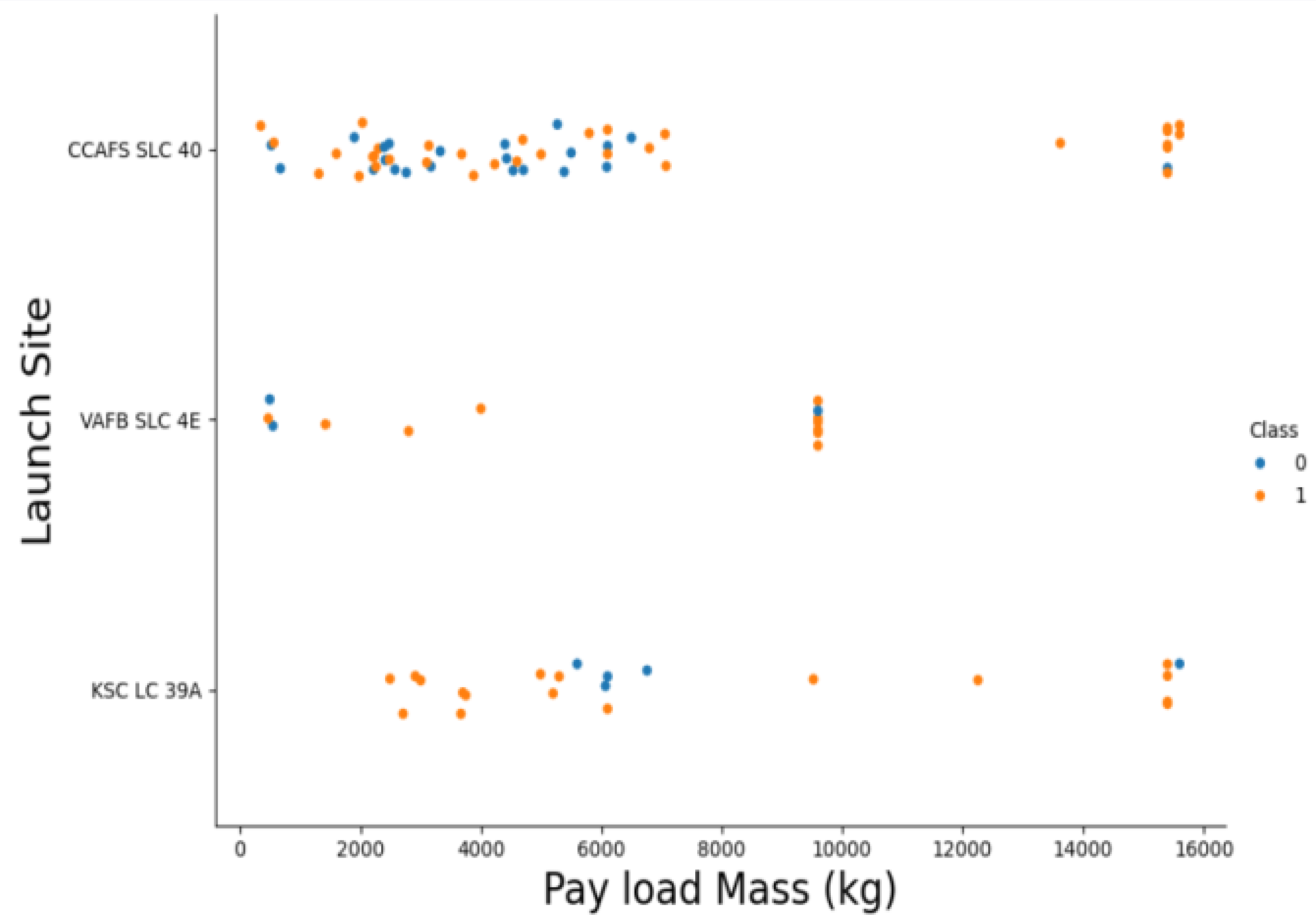
It can be noticed that the launch success rate (class 1) of all launch sites gets higher as the flight number of rockets increases. The launch site with the higher success rate is KSC LC 39A

Payload vs. Launch Site



The success (class 1) or the failure (class 0) of a launch outcome in correlation with the launch sites and the payload mass

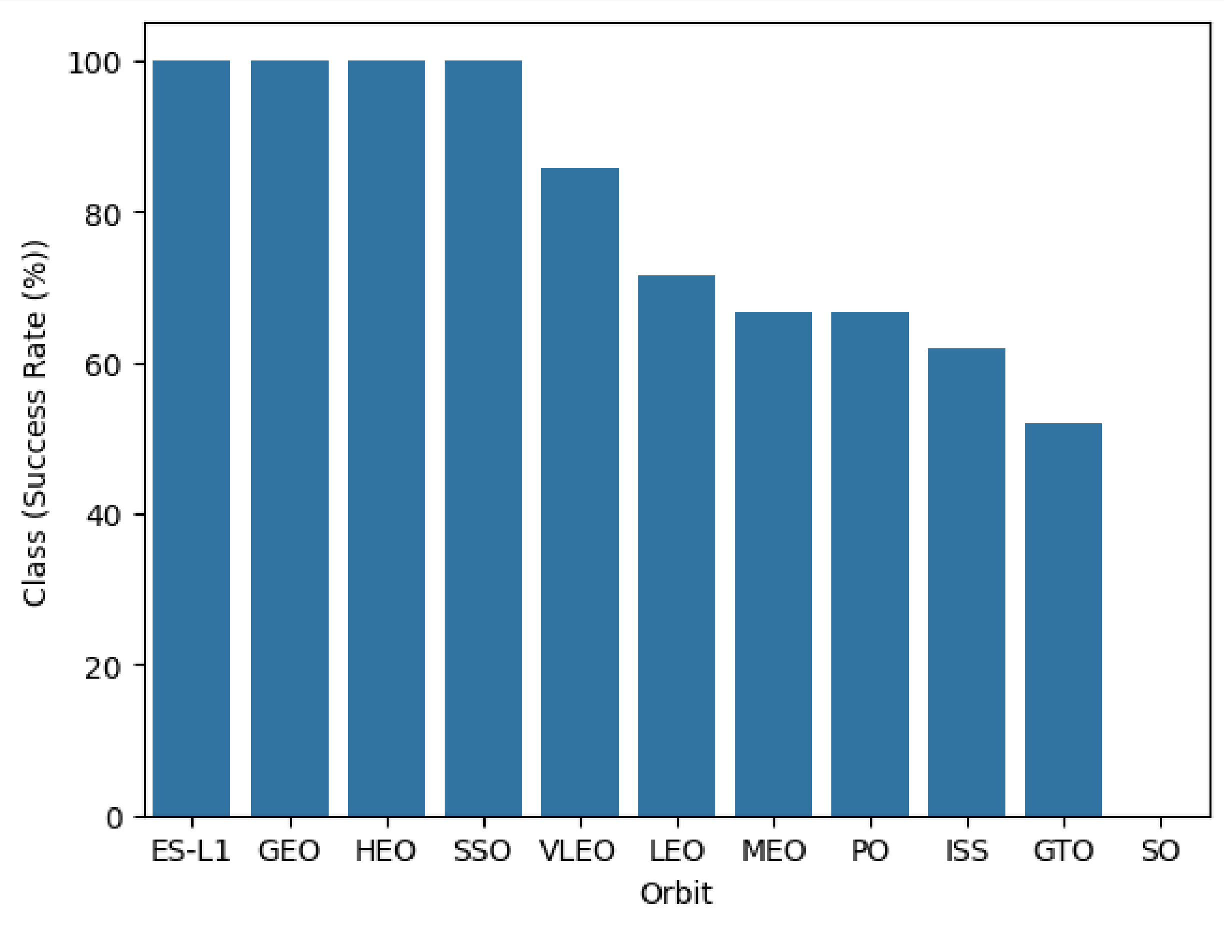
Payload vs. Launch Site



It can be noticed that rockets, which have been launched from KSC LC 39A launch site, perform the highest success rate at 100%, provided the pay load mass is equal or less than 5,000 kg.

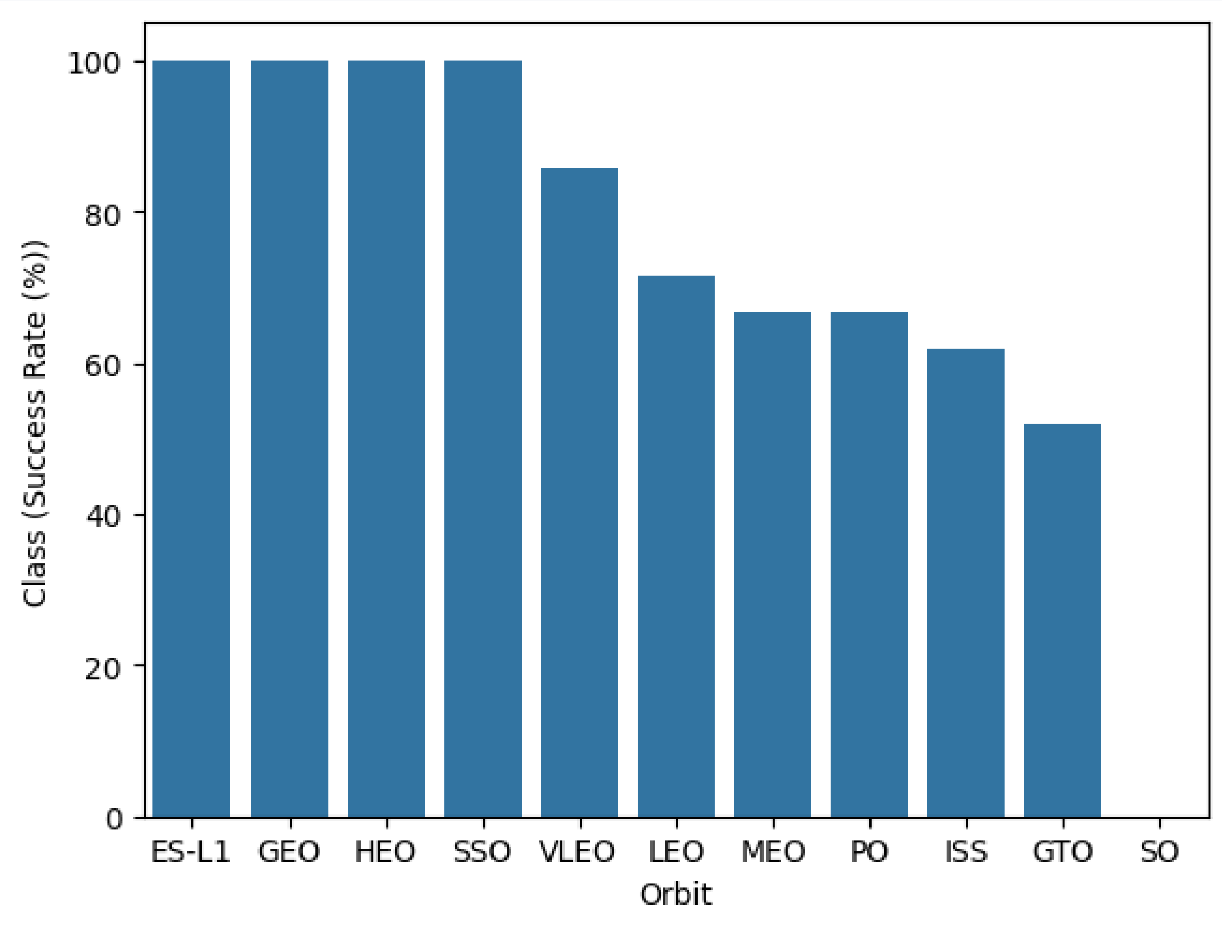
In general, payload mass seems that is not a determining factor for the launch outcome in correlation with the launch sites.

Success Rate vs. Orbit Type



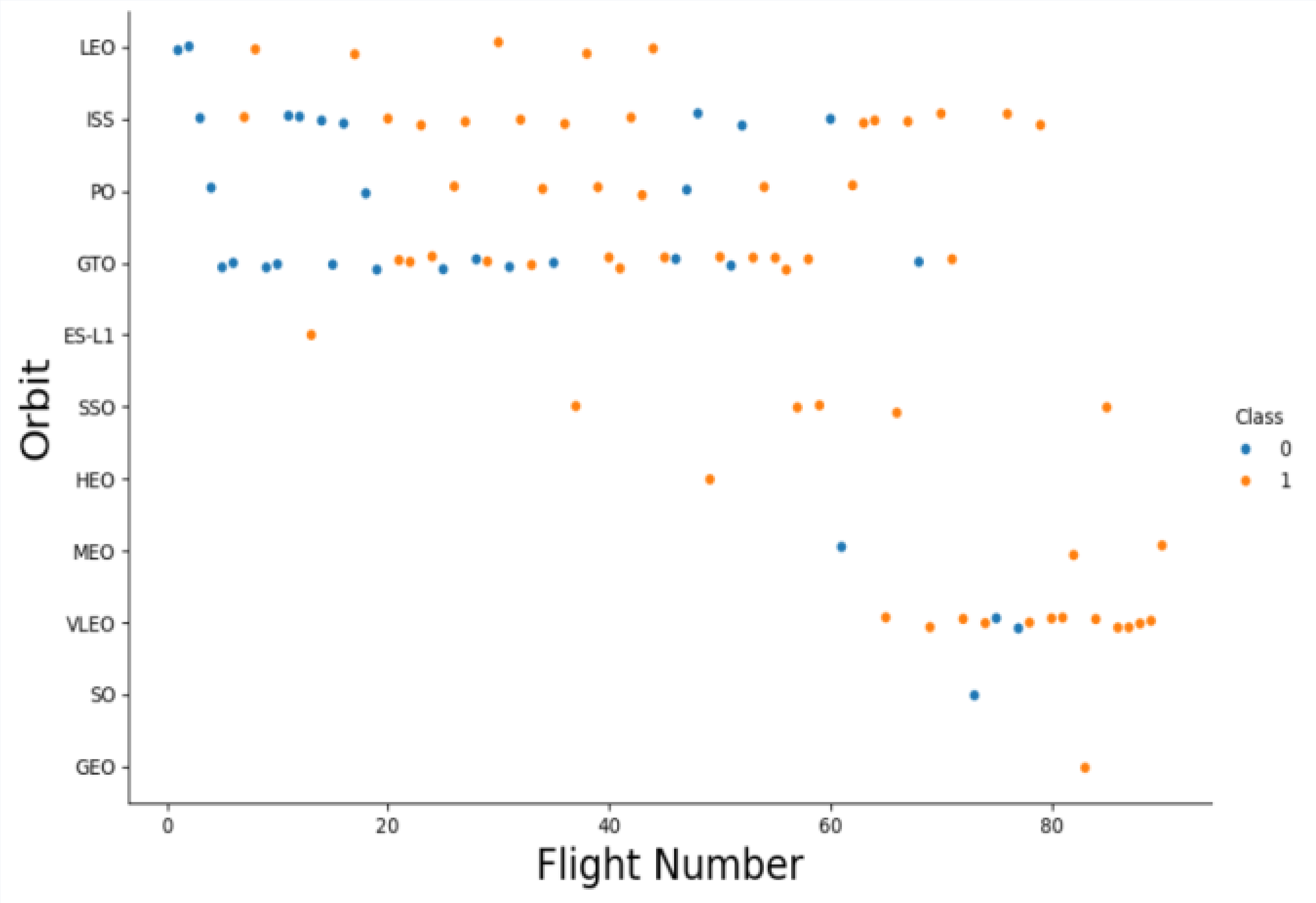
The success rate of launch outcome in correlation with the type of orbits that rockets have carried out.

Success Rate vs. Orbit Type



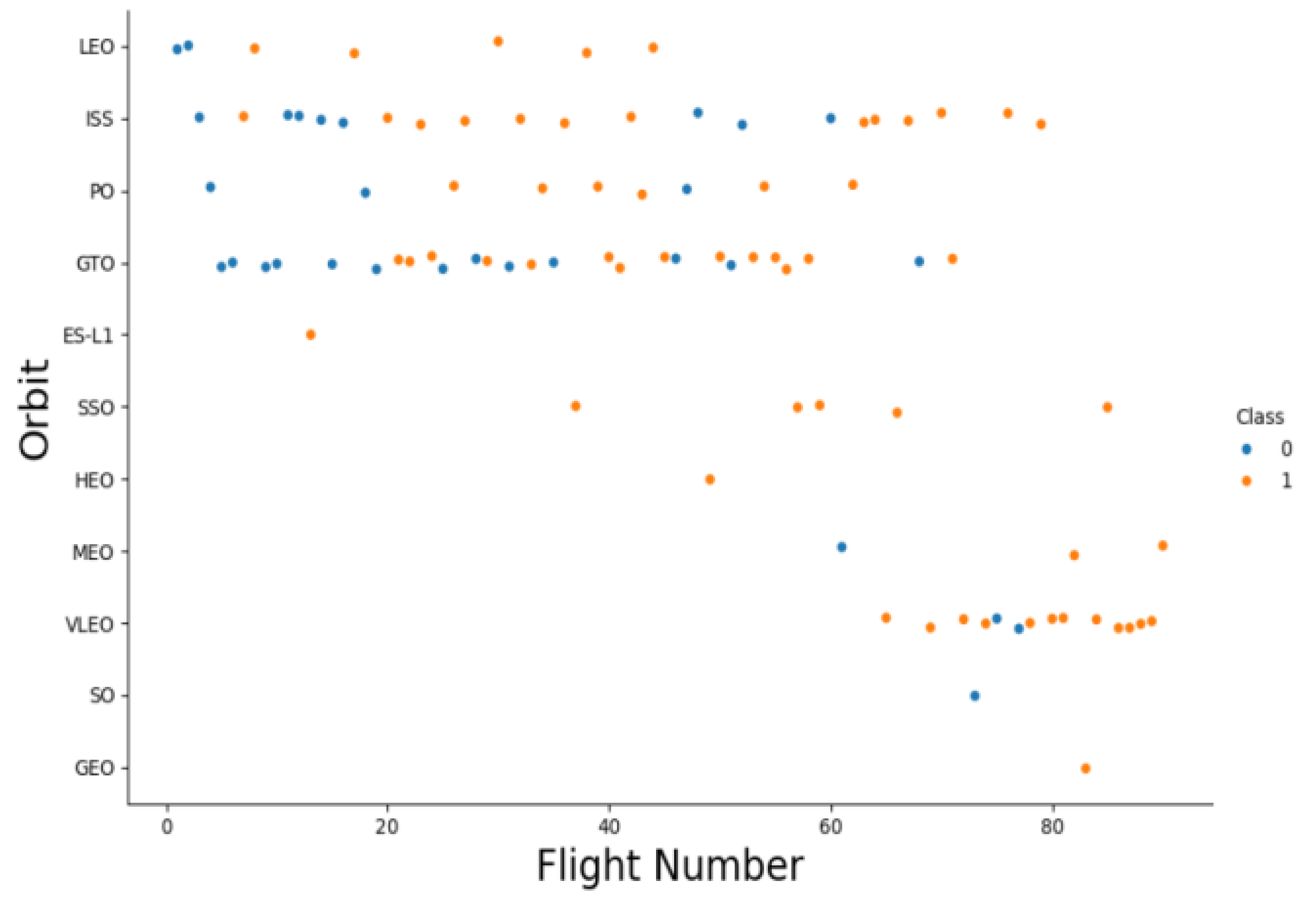
It can be noticed that rockets that have created certain type of orbits like ES-L1, GEO, HEO, SSO, the rate of successful launch outcome is 100%. To the contrary, rockets that have created SO orbit, the success rate is 0%.

Flight Number vs. Orbit Type



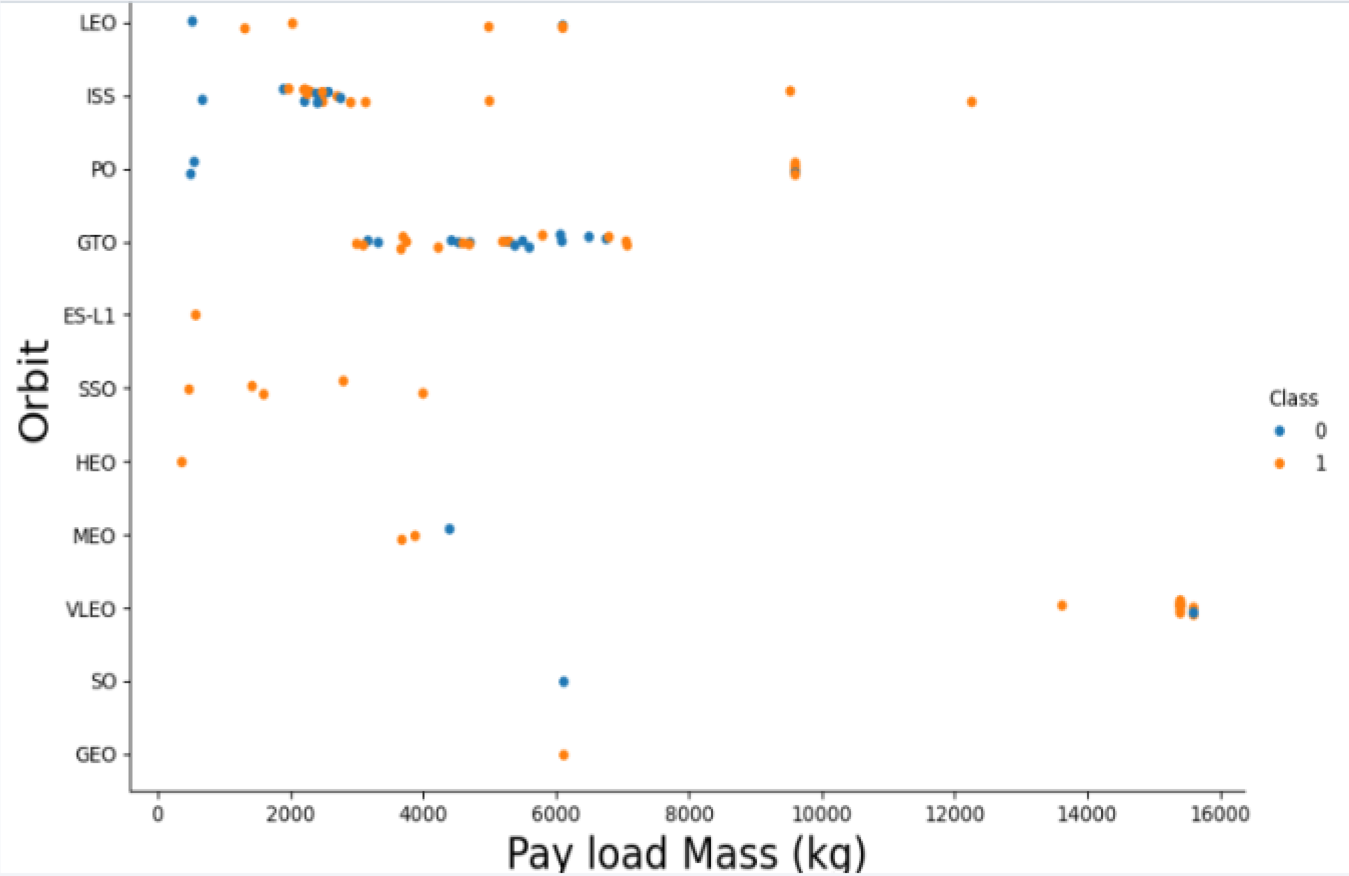
The success (class 1) or the failure (class 0) of a launch outcome in correlation with the orbit type that a rocket have made and the flight number

Flight Number vs. Orbit Type



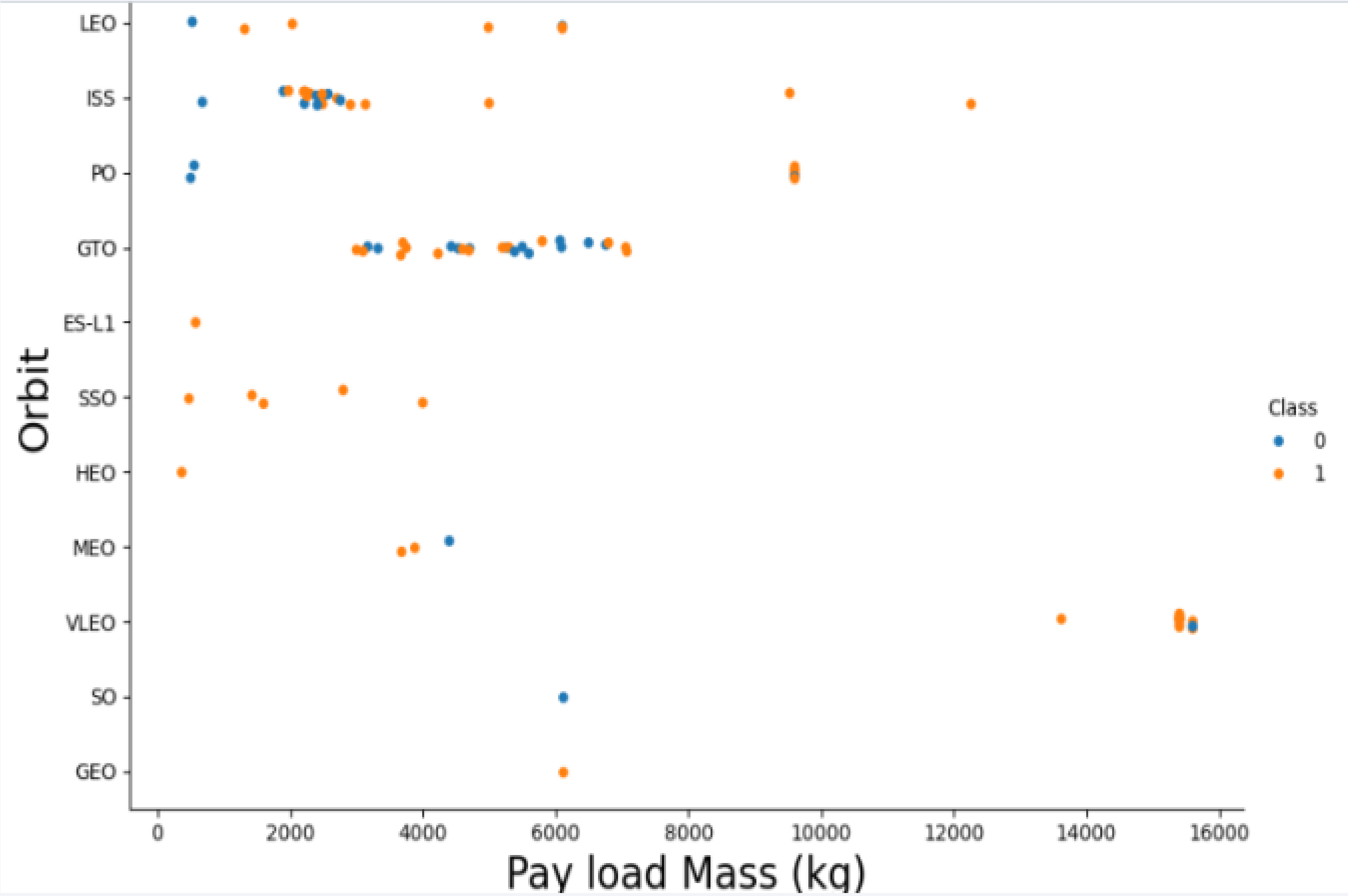
It can be noticed that the launch success rate (class=1) increases for all types of orbit that rockets have created as the flight number increases, and especially, that happens for rockets with VLEO and ISS orbits.

Payload vs. Orbit Type



The success (class 1) or the failure (class 0) of a launch outcome in correlation with the orbit type that a rocket have made and the pay load mass.

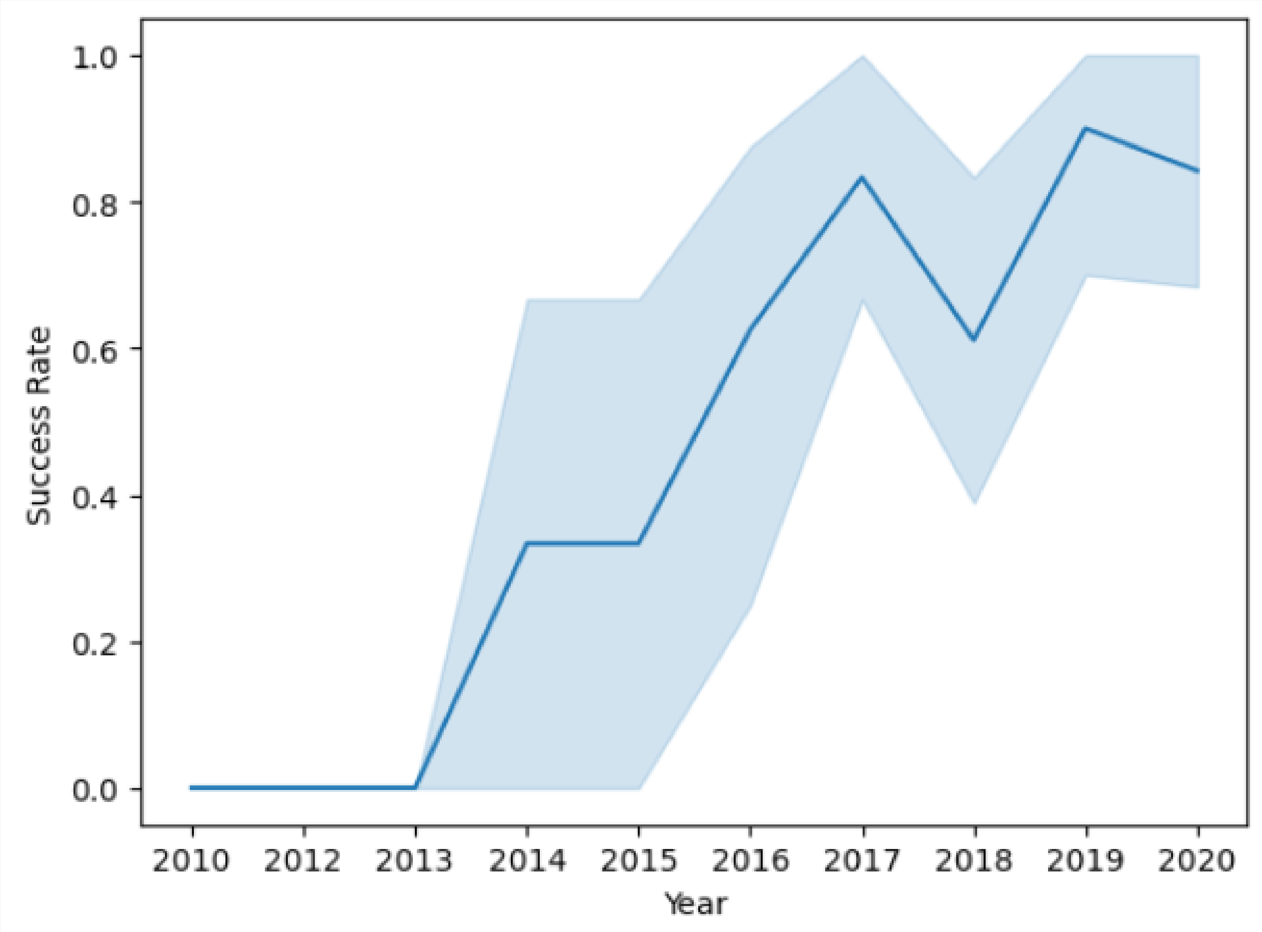
Payload vs. Orbit Type



It can be noticed that the launch success rate (class=1) is 100% when rockets have created SSO orbit and the pay load mass is equal or less than 4,000 kg.

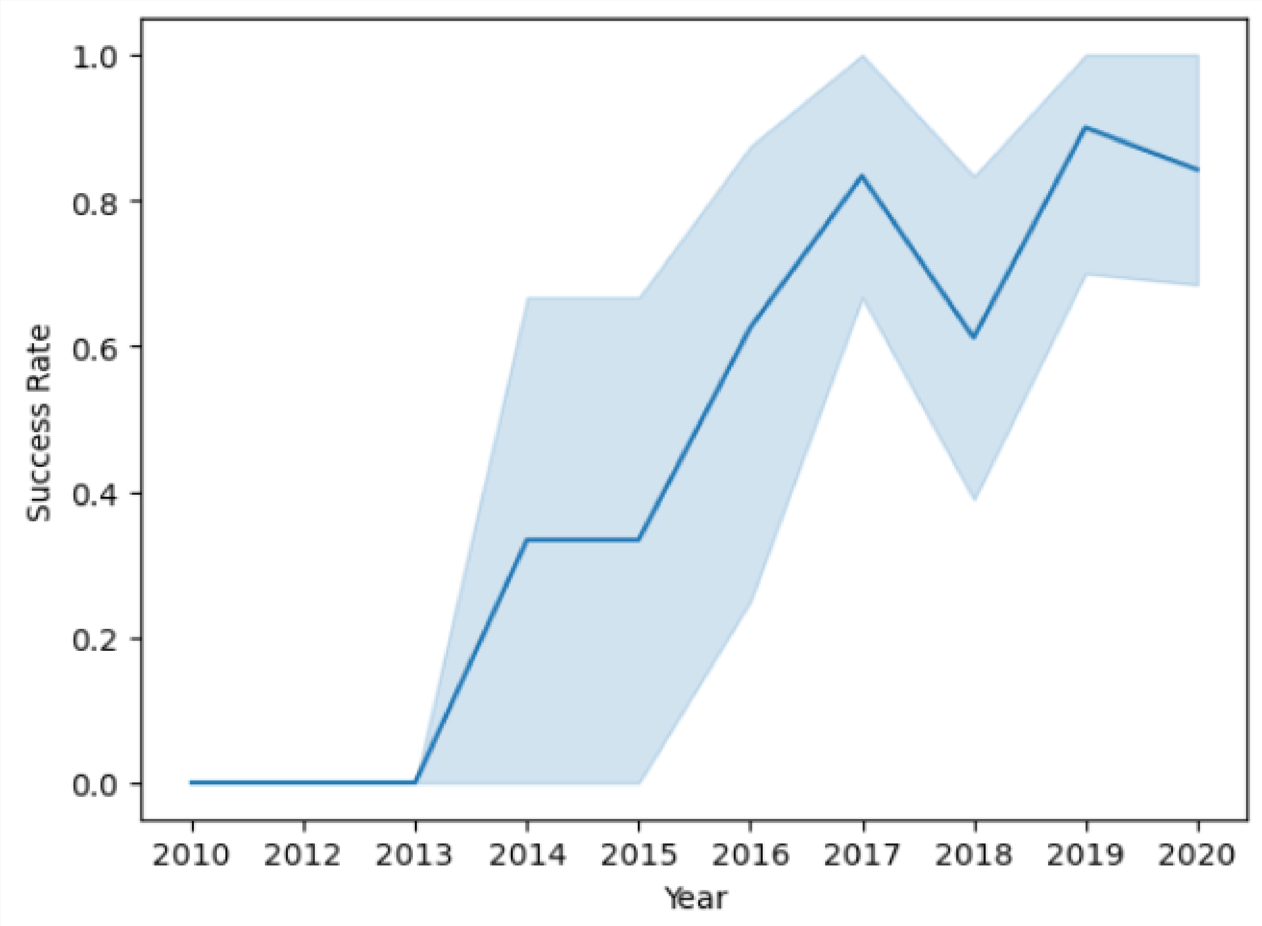
However, pay load mass seems to be a non-determining factor for the launch outcome in correlation with the type of orbit.

Launch Success Yearly Trend



The yearly average launch success rate of space X rockets

Launch Success Yearly Trend



According to the plot, the average launch success rate increases at 80% in 2017, over the period from 2013 to 2017. In 2018, the average success rate falls at 60%. Lastly, the average success rate peaks at 90% in 2019.

All Launch Site Names

- The names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

- There are four unique launch sites

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

Launch Site Names Begin with 'CCA'

- **5 records where launch sites begin with the string “CCA”**

```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

- **The records include the launch site CCAFS LC-40**

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
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 (parach
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass

- **The total payload mass carried by boosters and launched by NASA**

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

- **The total payload mass used by NASA is 45,596 Kg**

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

Average Payload Mass by F9 v1.1

- **The average payload mass carried by booster version F9 v1.1**

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

- **The average payload mass carried by Falcon 9 booster is 2534.66 Kg**

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

- **The date when the first successful landing outcome in ground pad was achieved**

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' where "Landing_Outcome" = "Success (ground pad)";
```

- **The date when the first successful landing in ground pad took place is on 12 December 2015.**

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql select distinct "Booster_Version", "Payload" from "SPACEXTBL" where "Landing_Outcome" = "Success (drone ship)" and PAYLOAD_MASS__KG_> 4000 and PAYLOAD_MASS__KG_<6000 ;
```

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

- **There are four distinct booster names with successful landing in drone ship and payload mass greater than 4000 but less than 6000**

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- **The total number of successful and failure mission outcomes**

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

- **The total number of successful and unsuccessful space X mission outcome is 100 and 1 respectively**

]:

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

Boosters Carried Maximum Payload

- **The names of the booster versions which have carried the maximum payload mass using sub query**

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Boosters Carried Maximum Payload

- **There are twelve boosters have carried payload mass equals 15,600 kg which is equivalent to the maximum payload mass**

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

- Launch records that include month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015. We use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql select substr(Date,0,5) as Year, substr(Date,6,2) as Month,"Booster_Version", "Launch_Site", "Landing_Outcome" from "SPACEXTBL" where substr(Date,0,5)="2015" and "Landing_Out
```

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

- There are two 2015 launch records, in January and April when the Falcon 9 boosters failed to land in drone ship at CCAFS LC-40 launch site

Year	Month	Booster_Version	Launch_Site	Landing_Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- **The count of landing outcomes between the date 04-06-2010 and 20-03-2017**

```
%sql select "Landing_Outcome", count(*) from SPACEXTBL where ("Date" between "2010-06-04" and "2017-03-20") group by "Landing_Outcome" order by count(*) desc ;
```

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

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

- Mostly counted landing outcome is the fact that rockets could not enter into the landing process (No attempt). Other types of landing outcome that took place between 2010-06-04 and 2017-03-20 are successful landings both in drone ship (Success drone ship) and ground pad (Success ground pad), the failure landings in drone ship (Failure drone ship) or by using parachute (Failure parachute), controlled (Controlled ocean) and uncontrolled (Uncontrolled ocean) landings in ocean, and finally, precluded landings (Precluded drone ship) where rocket was not allowed to land in drone ship.

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

Section 3

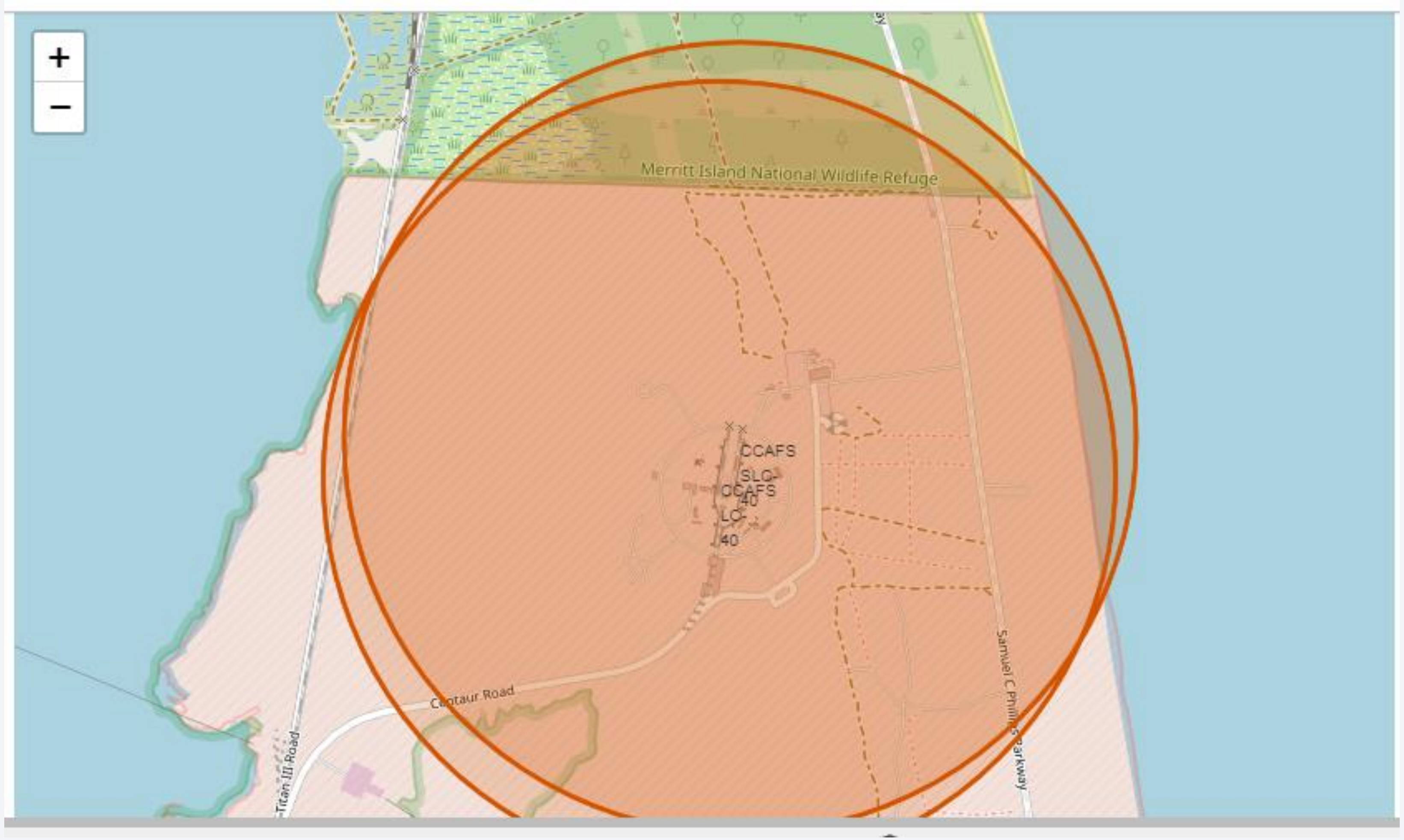
Launch Sites Proximities Analysis

Marking launch sites with circle objects on the map

KSC LC 39A launch site

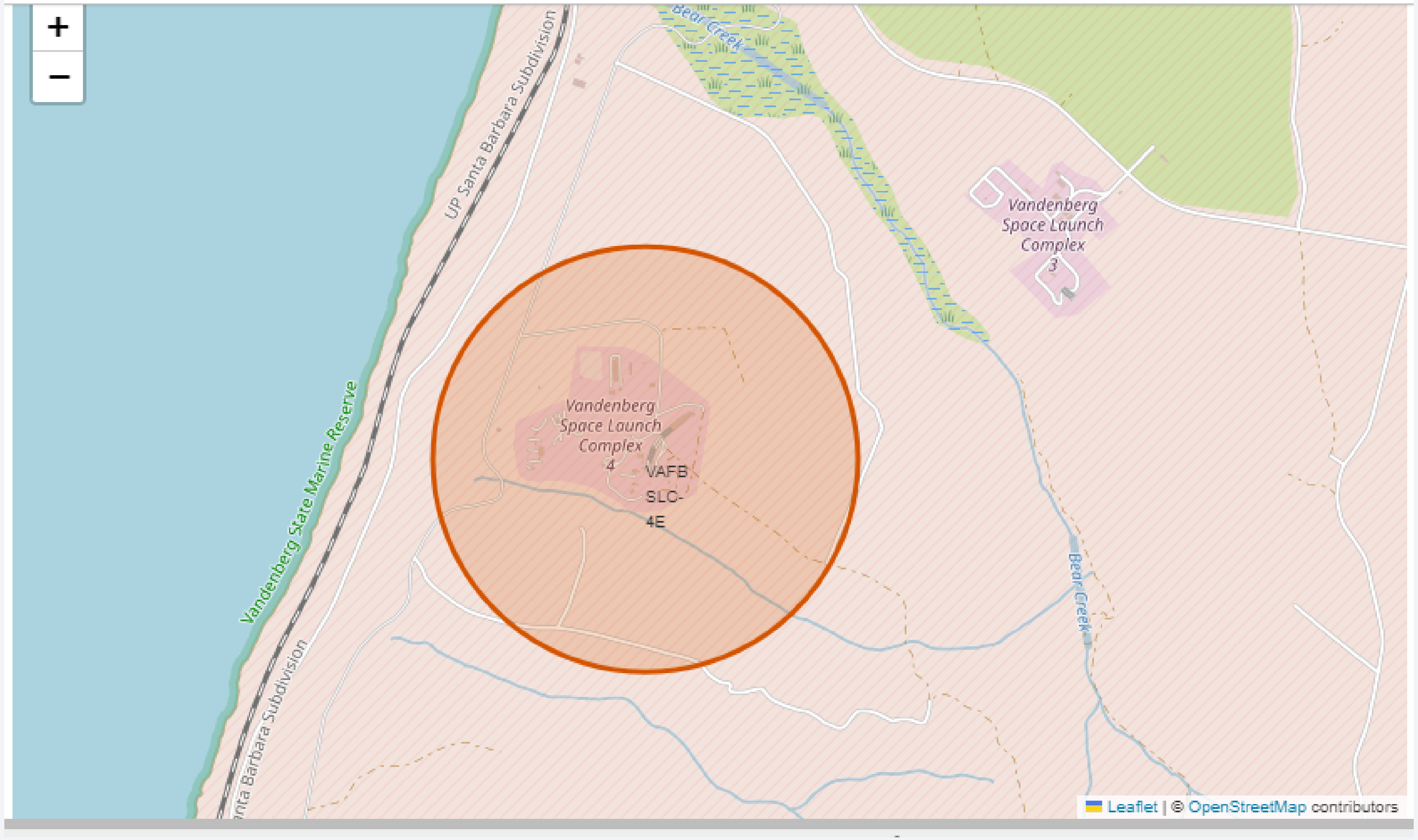
in Florida.

Marking launch sites with circle objects on the map



CCAFS SLC-40 and CCAFS LC-40 launch sites in Florida

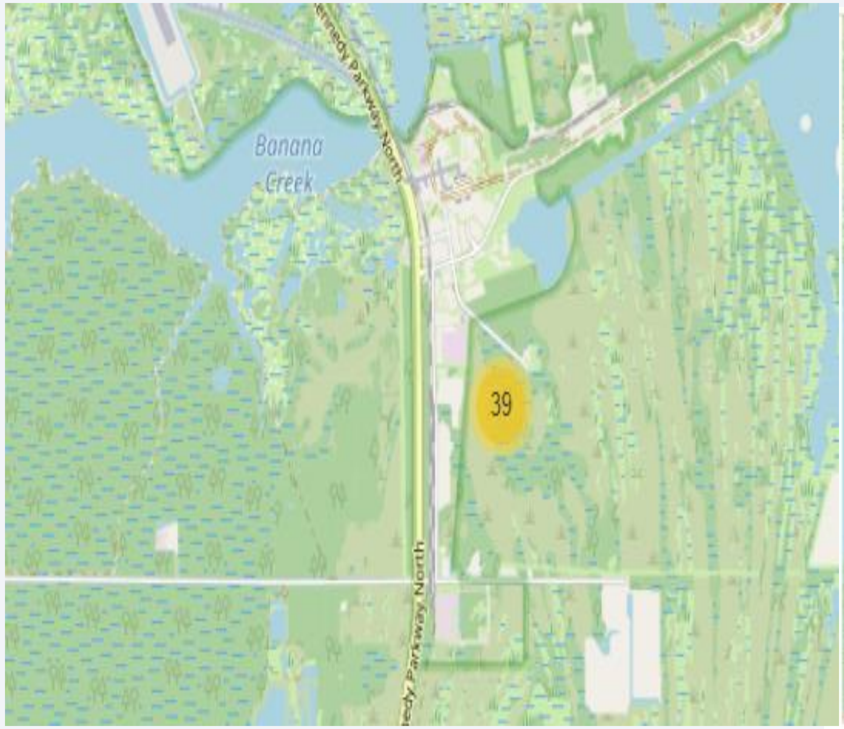
Marking launch sites with circle objects on the map

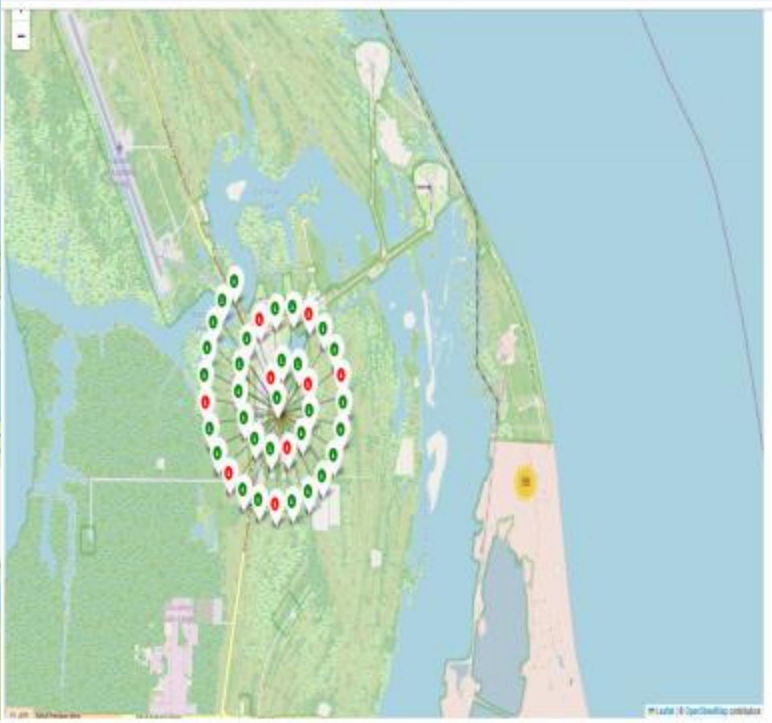


VAFB SLC-4E launch site

in California

Using color markers for launch outcomes

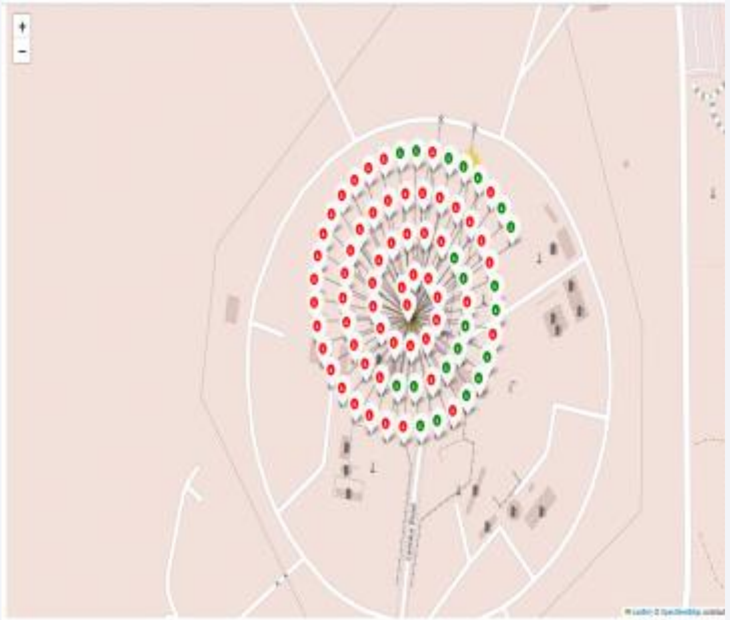


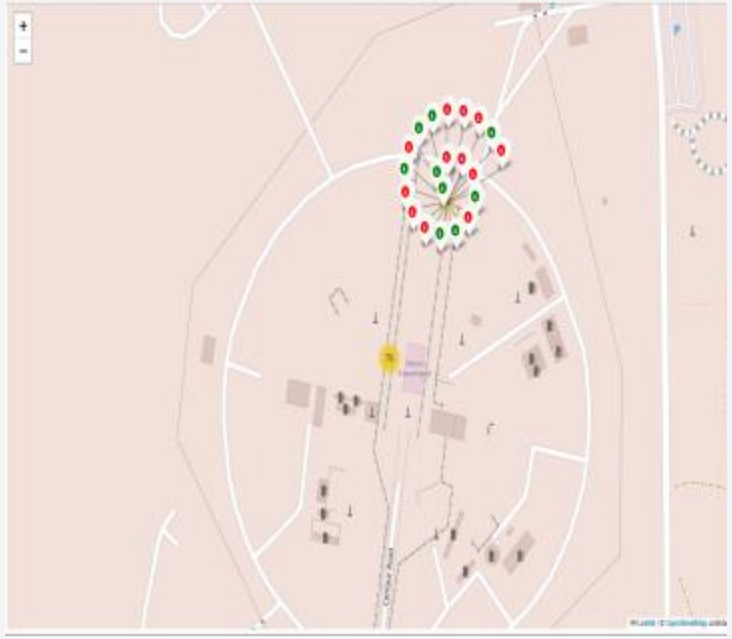


KSC LC 39A launch site in Florida

There are 39 color markers. Green color markers (class=1) that significantly outnumber red color markers (class=0). That means successful rocket launches are much more than unsuccessful ones.

Using color markers for launch outcomes

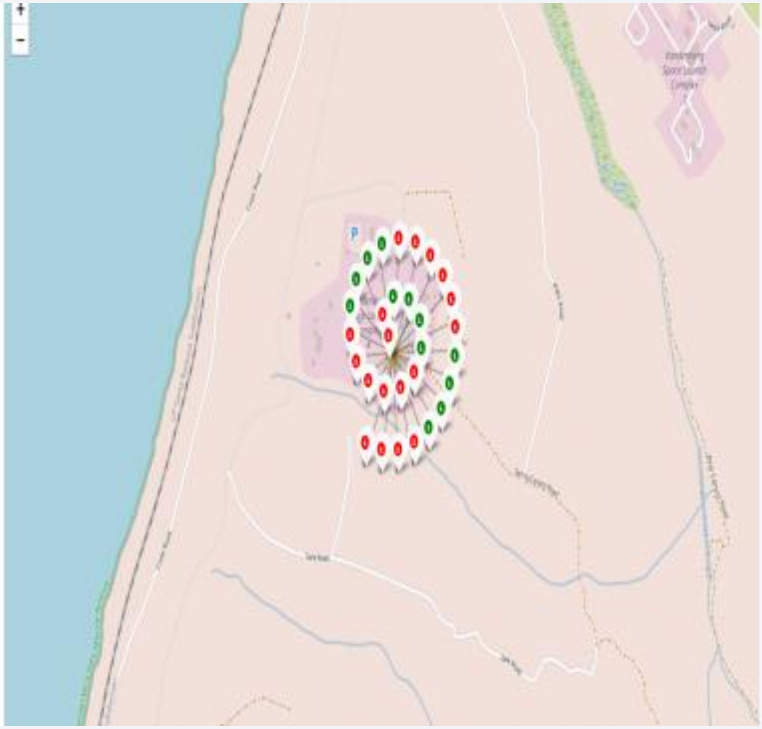




There are 21 color markers on the CCAFS SLC-40 launch site and 78 color markers on the CCAFS LC-40 launch sites in Florida

The predominant color of markers is red (class=0) on both CCAFS LC-40 and CCAFS SLC-40 launch sites. That means unsuccessful rocket launches prevail over successful ones.

Using color markers for launch outcomes



VAFB SLC-4E launch site in California

There are 30 color markers. Green color markers (class=1) are slightly fewer than red color markers (class=0). It is not clear whether the launch site in California affects the launch outcome.

Calculating the distances between a launch site and its proximities

CCAFS SLC-40 launch site in Florida

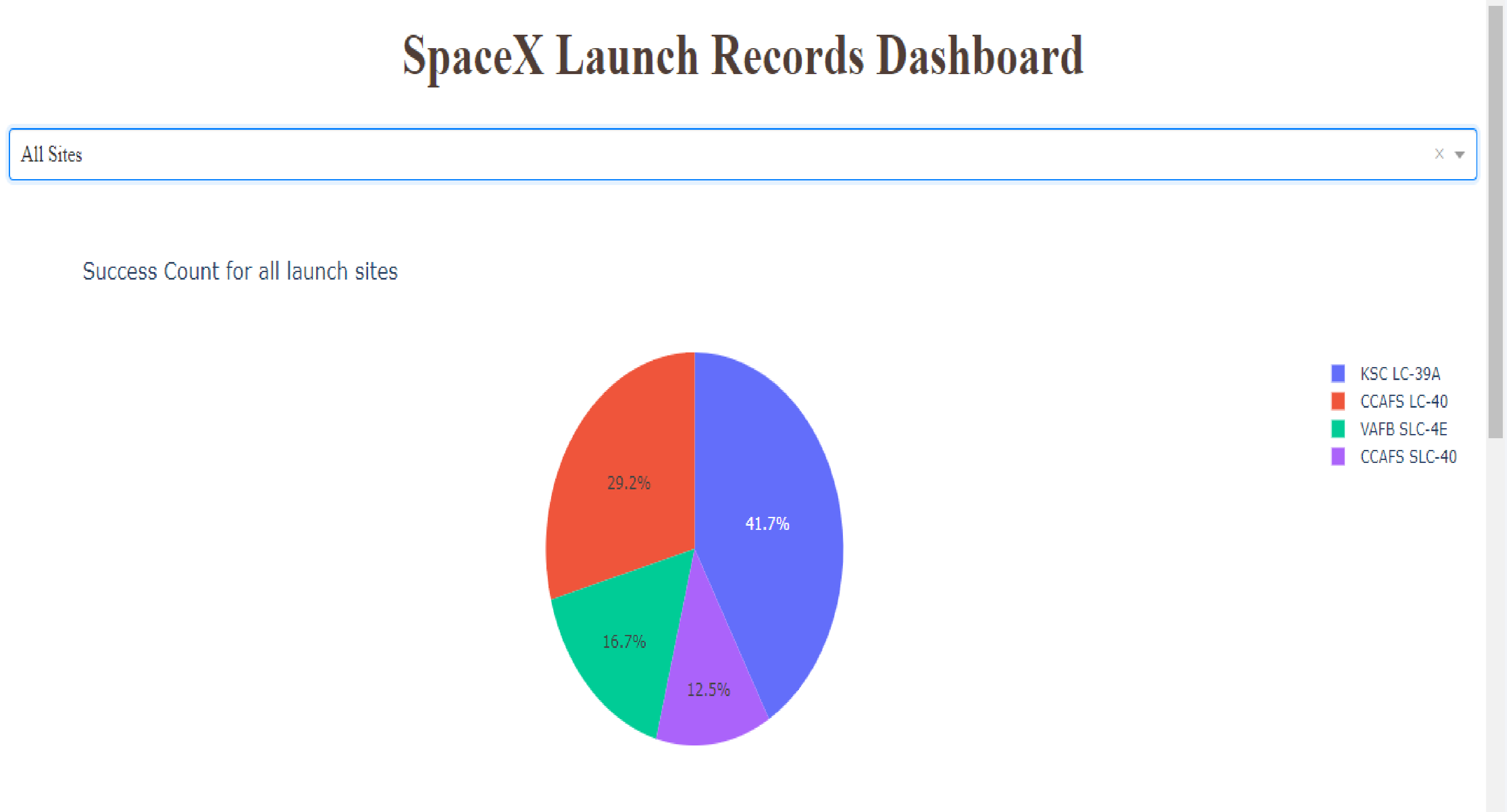
- The distance from Titusville city (latitude = 28.61208, longitude = -80.80764) is 23.19 Km
- The distance from the coastline is 0.86 Km

Section 4

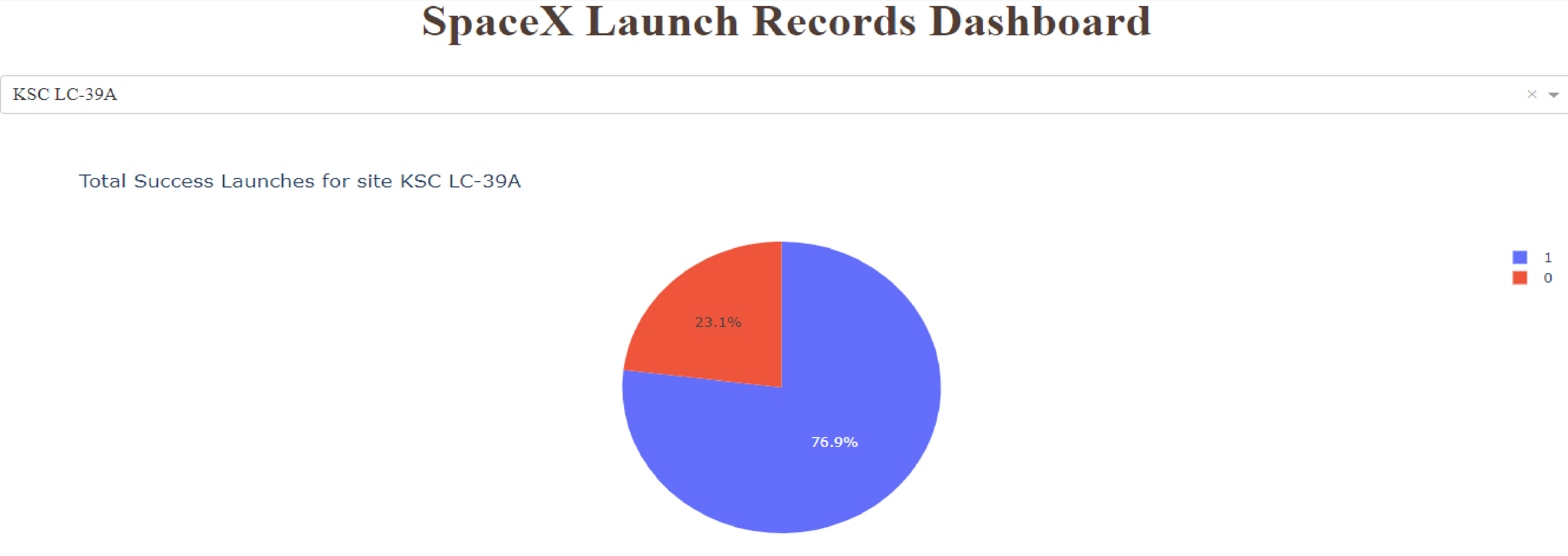
Build a Dashboard with Plotly Dash

Launch success count for all launch sites

- KSC LC-39A has the highest success rate at 41.7%
- The success rate of CCAFS LC-40 launch site is at 29.2%
- The success rate of VAFB SLC-4E launch site is at 16.7%
- CCAFS SLC-40 launch site has the lowest success rate at 12.5%

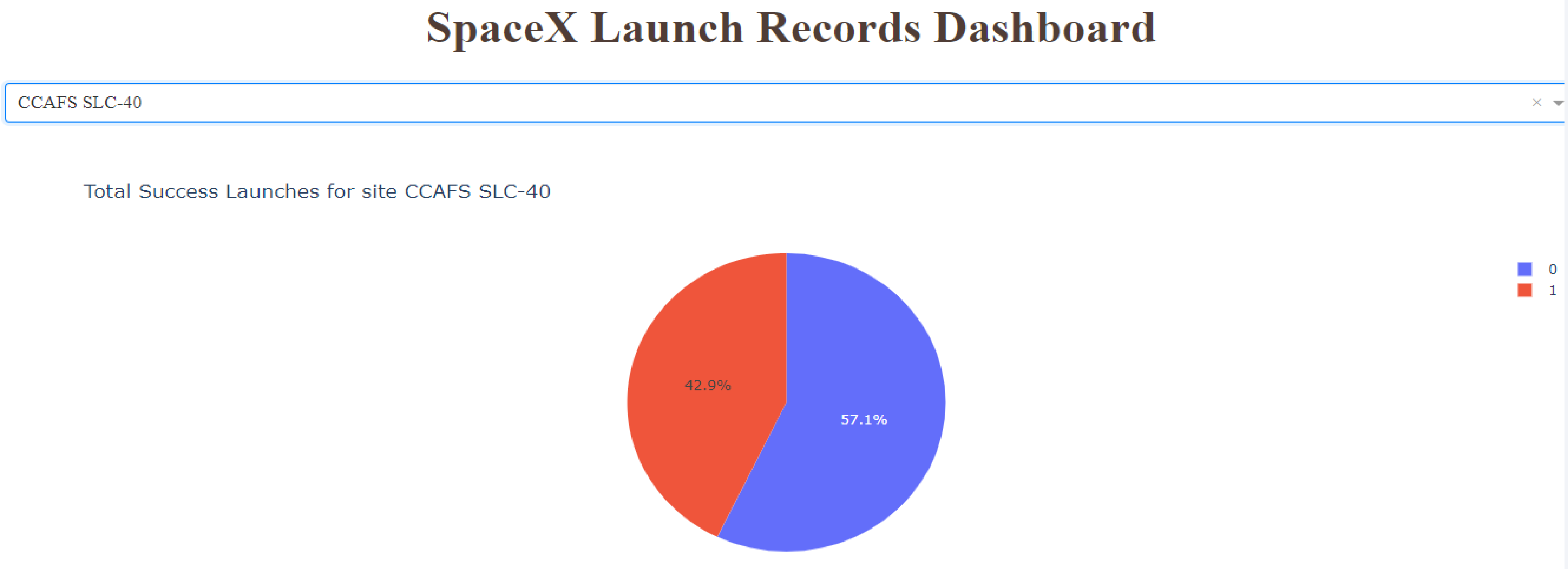


The launch success ratio of a launch site with the highest launch success rate



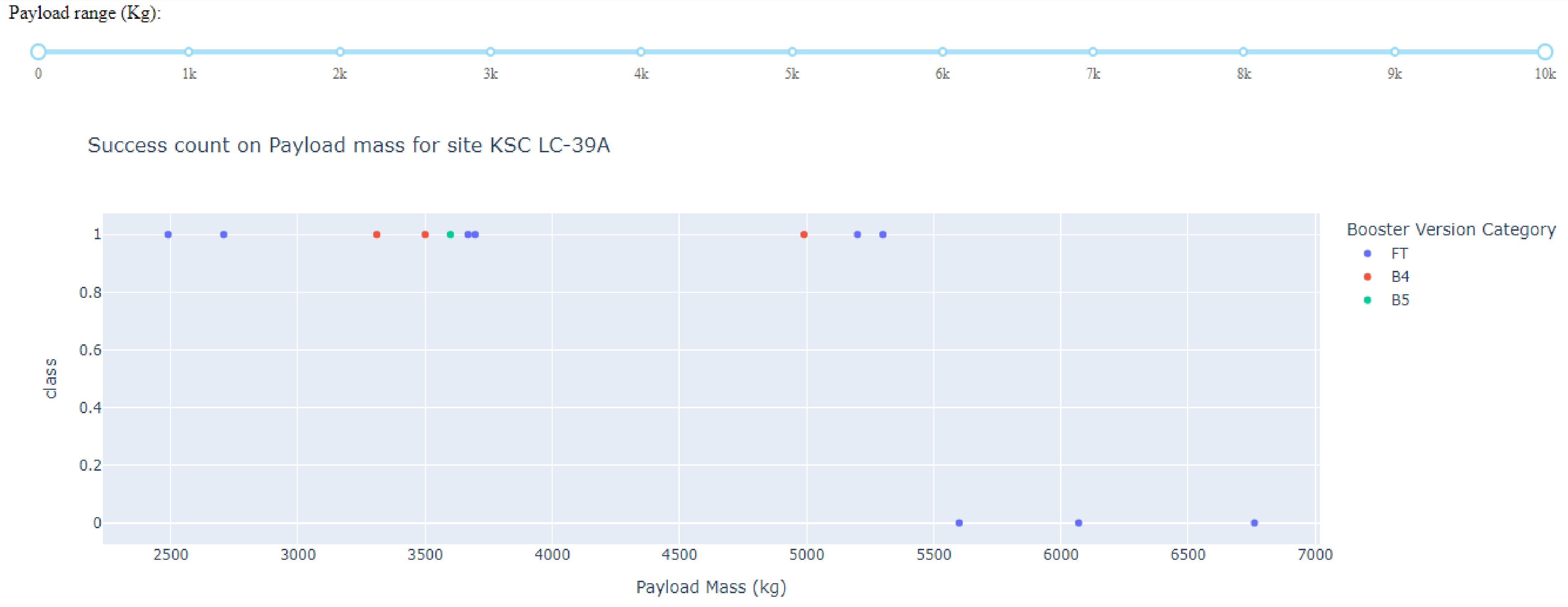
- The launch success rate of the KSC LC-39A launch site is at 76.9%
- The failed launch rate of the KSC LC-39A launch site is at 23.1%

The launch success ratio of a launch site with the lowest launch success rate



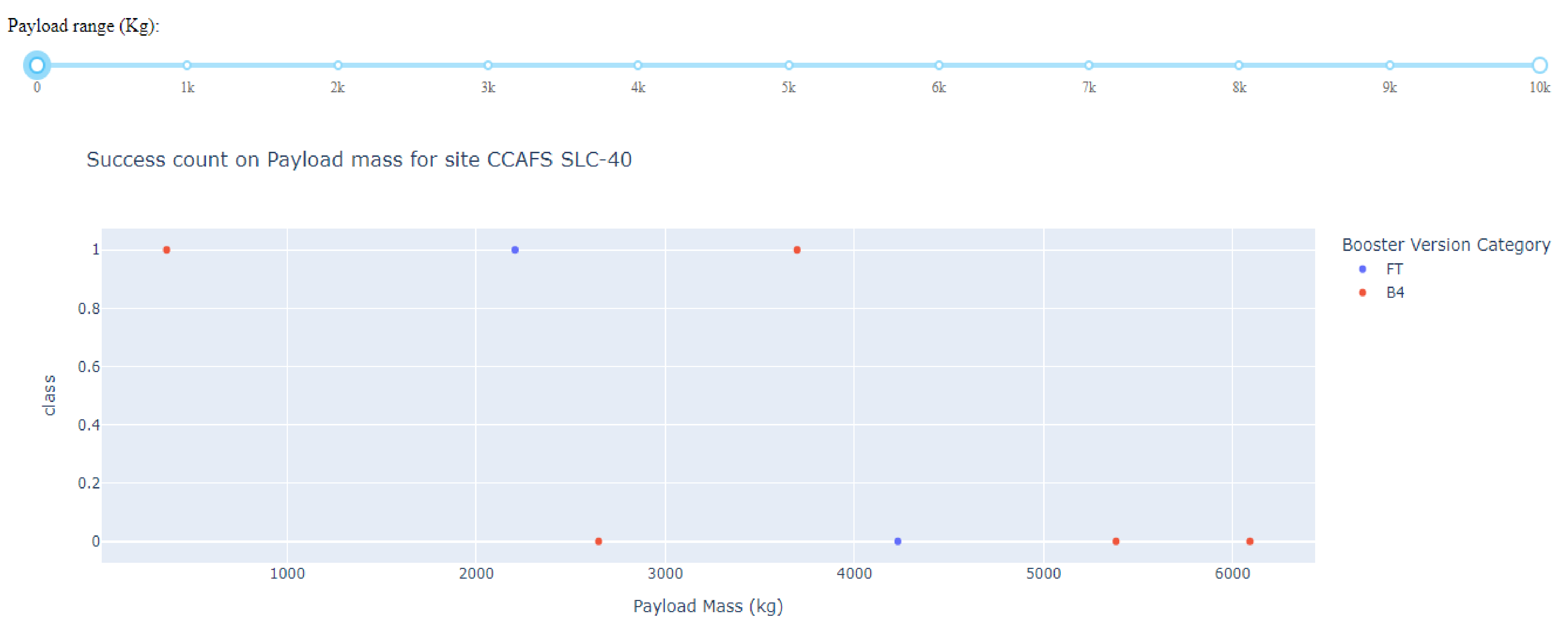
- The launch success rate of the CCAFS SLC-40 launch site is at 57.1%
- The failed launch rate of the CCAFS SLC-40 launch site is at 42.9%

Launch outcome in correlation with payload mass for KSC LC-39A



The KSC LC-39A launch site has unsuccessful launches (class=0) using the FT booster version and the payload mass is higher than 5,500 Kg.

Launch outcome in correlation with payload mass for CCAFS SLC-40



The CCAFS SLC-40 launch site has mainly unsuccessful launches (class=0) using the B4 booster version and the payload mass is higher than 2,500 Kg.

Section 5

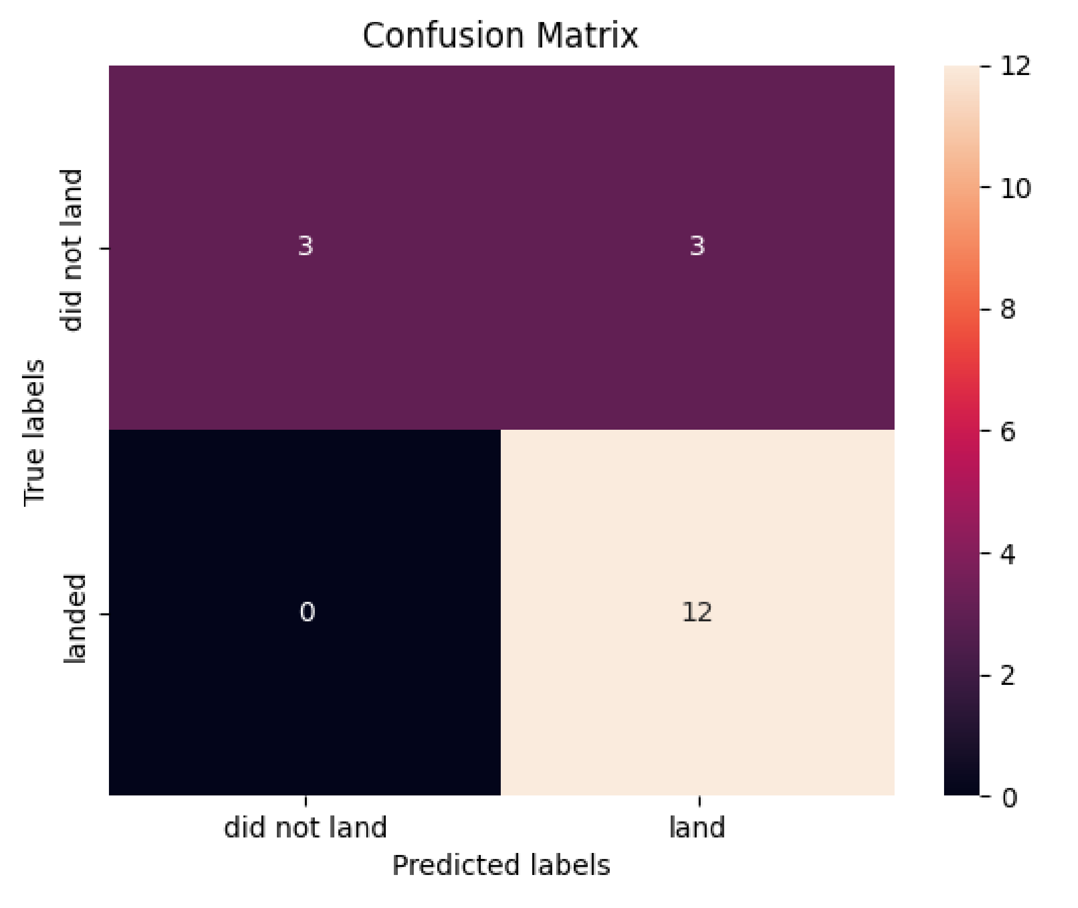
Predictive Analysis (Classification)

Classification Accuracy

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

The accuracy rate for each method (Logistic regression, Decision Tree, SVM, KNN) is the same

Confusion Matrix



- According to the confusion matrix of all methods, we can deduce that all algorithms produce the same results
- All the algorithms have failed to predict correctly for 3 times out of 18. Specifically, models have incorrectly predicted that rockets would land successfully but actually they did not manage to do so.
- All the algorithms have correctly predicted for 15 times out of 18. Specifically, models have correctly predicted for 12 times that rockets would land successfully and for 3 times that rockets would not land.

Conclusions

- VAFB SLC 4E, KSC LC 39A and CCAFS SLC 40 launch sites show that launch success rate is getting higher as flight number is increasing.
- KSC LC 39A launch site shows the highest launch success rate, at 100%, as long as the payload mass is equal or less than 5,000 kg.
- Rockets, which create types of orbit such as ES-L1, GEO, HEO & SSO, have the highest success rates at 100%.
- Rockets that have made SSO orbit and the pay load mass is equal or less than 4,000 kg, the launch success rate is 100%.
- KSC LC 39A launch site has the highest launch success rate at almost 42% and CCAFS SLC-40 has the lowest launch success rate at 12.5%.
- The **class** is **zero** for the **KSC LC-39A** launch site when the FT booster version is used and the payload mass is higher than 5,500 Kg, while the **class** is **zero** for the **CCAFS SLC-40** launch site when the B4 booster version is used and the payload mass is higher than 2,500 Kg.
- All the classification methods (Logistic Regression, Decision Tree, SVM, KNN) have the same accuracy rate at 0.833.

Thank you!