IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

David Marquez
MM/DD/AAAA

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API and Web scraping.
  - Data Wrangling.
  - EDA with Data Analysis, Data Visualization and SQL.
  - Interactive map with Folium library.
  - Interactive dashboard with Plotly library.
  - Machine Learning predictive model.

- Summary of all results
  - EDA.
  - Interactive map and dashboard.
  - Predictive model.

Introduction

SpaceX's Falcon 9 rocket has revolutionized space travel by introducing reusable rockets. However, predicting whether a rocket will successfully land after launch remains a complex task. Multiple factors, such as payload mass, orbit, launch site and others, could affect the landing outcome. This capstone project aims to analyze and develop a predictive model to determine successful or failed landings.

We can determine the landing outcome by addressing the following questions:

- What are the main features that influence successful landings?
- Is there an evolution in successful landings or do failures continue?
- Do external factors affect the landing outcome?
- Can we determine the landing outcome using a predictive model?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - API SpaceX with multiple endpoints.
  - Web scraping from Wikipedia.
- Perform data wrangling:
  - Drop NaN Values.
  - Generate one-hot encoding for categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL:
  - Analyzing launch facilities and orbits.
  - Visualizing the relationship between variables such as Launch Site, Flight Number, Payload Mass and Orbit.
- Perform interactive visual analytics using Folium and Plotly Dash:
  - Total Success Launches by Site.
  - Correlation between Payload Mass and Success for all Sites.
- Perform predictive analysis using classification models:
  - Build, tune, evaluate classification models.

Data Collection

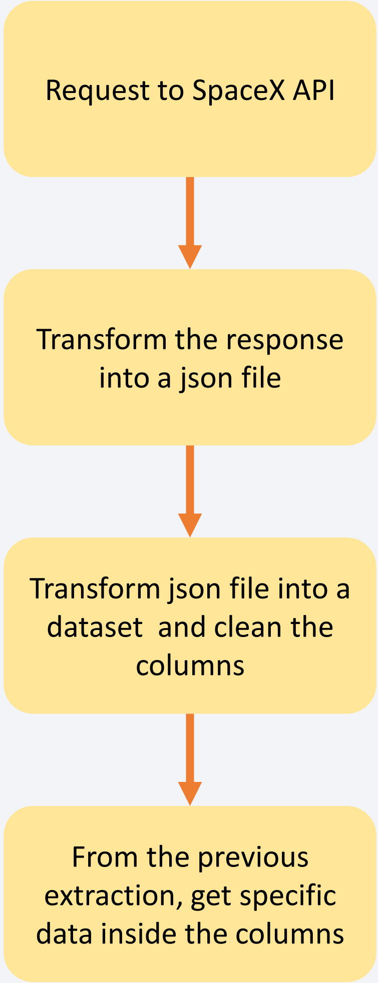
- The data was collected by API SpaceX: https://api.spacexdata.com/v4, specifically of:
  - FlightNumber
  - Date
  - BoosterVersion
  - PayloadMass
  - Orbit
  - LaunchSite
  - Outcome
  - Flights
  - GridFins
  - Reused
  - Legs
  - LandingPad
  - Block
  - ReusedCount
  - Serial
  - Longitude
  - Latitude

- Web scraping from Wikipedia URL (as of 9th June 2021): https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922 obtaining:
  - Flight No
  - Launch site
  - Payload mass
  - Orbit
  - Customer
  - Boosterlanding
  - Payload
  - Launch outcome
  - Version Booster
  - Date
  - Time

Data Collection – SpaceX API

- Collect the data from SpaceX API to obtain a response.
- Become the response object into json file.
- Transform json file in a dataset and clean the columns that doesn't add value.
- Use functions to extract from some features specific data inside them.

GitHub Notebook URL:

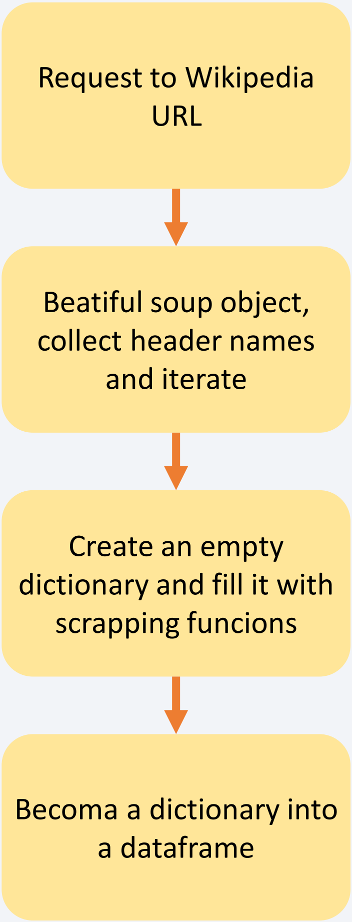


Request to SpaceX API
Transform the response into a json file
Transform json file into a dataset and clean the columns
From the previous extraction, get specific data inside the columns

Data Collection - Scraping

- Request to Wikipedia URL to obtain response code 200.
- Create a beautiful soup object from HTML response, collect relevant column names from header and iterate through the “th” elements to extract the columns.
- Create an empty dictionary with the columns obtained, delete one who don’t add value.
- Create empty lists and fill them with functions who scrapping the information, then add to dictionary. Then, Become a dictionary into a dataframe.

GitHub Notebook URL:



Request to Wikipedia URL

Beatiful soup object, collect header names and iterate

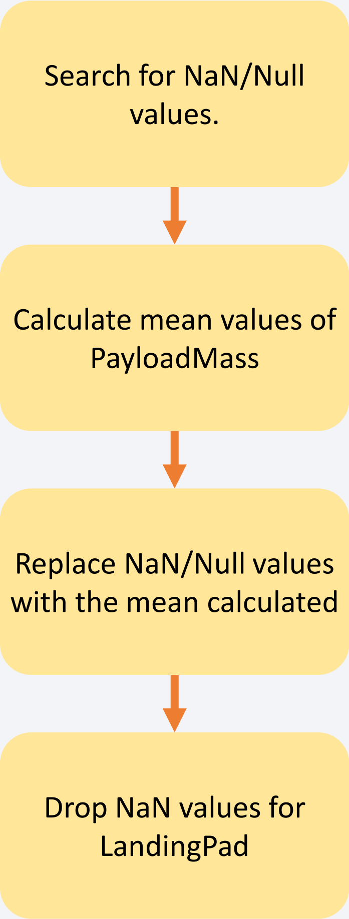
Create an empty dictionary and fill it with scrapping funcions

Becoma a dictionary into a dataframe

Data Wrangling

NaN/Null Values

- From the dataset obtained in Data Collection, search for NaN/Null values.
- For column Payload Mass calculate mean value.
- Replace NaN/Null values with the mean calculated previously.
- For Landing Pad drop the NaN values, because this columns has string values.



Search for NaN/Null values.

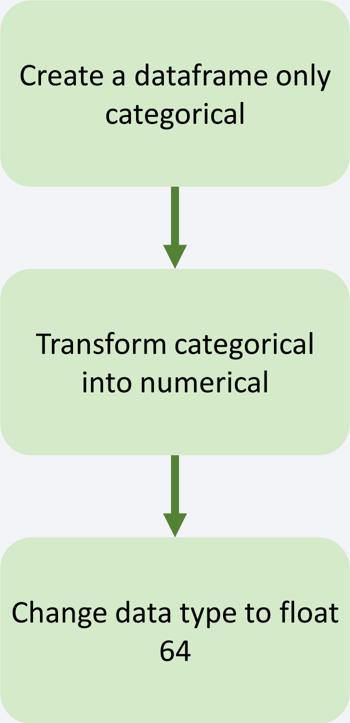
Calculate mean values of PayloadMass

Replace NaN/Null values with the mean calculated

Drop NaN values for LandingPad

One hot Encoding

- Create a dataframe with only categorical features.
- Transform categorical features to numerical using get dummies.
- Change datatype columns to float64.



Create a dataframe only categorical

Transform categorical into numerical

Change data type to float 64

GitHub Notebook URL:

EDA with Data Visualization

- Chart Summary:

  - Flight Number vs. Payload Mass (hue Class)
  - Flight Number vs Launch Site (hue Class)

  These graphics are to verify if Flight Number can affect successful landing considering the features Payload Mass and Launch Site.

  - Launch Site vs Payload Mass (hue Class) scatterplot and bar plot

  These graphics are to verify if features Payload Mass and Launch Site can affects over successful landing.

  - Orbit vs Success Count (hue Class)
  - Orbit vs Success Mean
  - Orbit vs Flight Number
  - Flight Number vs Orbit (hue Class)
  - Orbit vs Payload Mass
  - Payload Mass vs Orbit (hue Class)

  These ones is for see what orbit have the better success landing considering features Payload Mass, Launch Site and Flight Number.

  - Year vs Success Mean:

  We notice the success has increased over the years.

- GitHub Notebook URL:

EDA with SQL

- EDA SQL Summary:
  - Names of the unique launch sites in the space mission.
  - 5 records where launch sites begin with the string 'KSC’.
  - Total payload mass carried by boosters launched by NASA (CRS).
  - Average payload mass carried by booster version F9 v1.1.
  - Date where the successful landing outcome in drone ship was achieved.
  - Names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000.
  - Total number of successful and failure mission outcomes.
  - Names of the booster_versions which have carried the maximum payload mass.
  - Records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017.
  - Rank of 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

- GitHub Notebook URL:

Build an Interactive Map with Folium

- Map Objects Summary
  - Circle on NASA Johnson Space Center at Houston, Texas.
  - Circle on each launch site and show the names.
  - White marker with green or red center to identify for success or failed launches.
  - Circle on coastline, show the distance and a line from some launch site.
  - Line between a some launch site to its closest city, railway, and highway.

  These objects are added to identify geographical patterns and understand better the context problem.

- GitHub Notebook URL:

Build a Dashboard with Plotly Dash

- Plot Summary
  - Dropdown to select the launch site.
  - Pie Chart with success launch by launch site selected, include all.
  - Range slider to define the payload mass range for the next chart.
  - Scatter plot showing success or failed launch per boost version, by launch site selected and payload mass range defined.

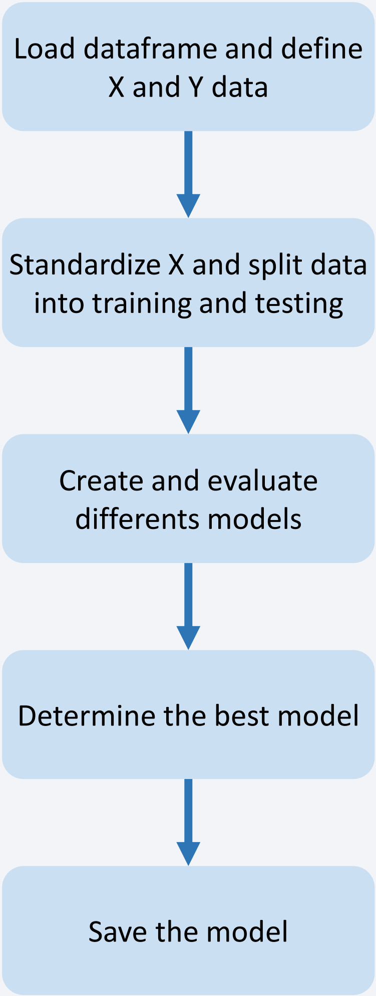
  These interactive items give us more information about how different features simultaneously could affects successful landing.

- GitHub Code URL:

Predictive Analysis (Classification)

- We load the dataframe from URL https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv .
- Define X varible and Y (goal) variable into numpy array.
- Standardize X and reassign it to the variable X.
- Split X data in training (80%) and testing (20%)
- Create models: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest.
- Evaluate them with their accuracy and confusion matrix-
- With the data obtained previously, determine the best model.
- Save (best) model object.

- GitHub Notebook URL:
- GitHub Model URL:



Load dataframe and define X and Y data

Standardize X and split data into training and testing

Create and evaluate differents models

Determine the best model

Save the model

Results

- With EDA we found that:
  - KSC LC-39A and VAFB SLC 4E has the best success rate
  - The best orbits for successful landing are SSO, VLEO, MEO, and GEO
  - The successful landing has increased over the years.
  - Payload Mass doesn’t affect the success rate

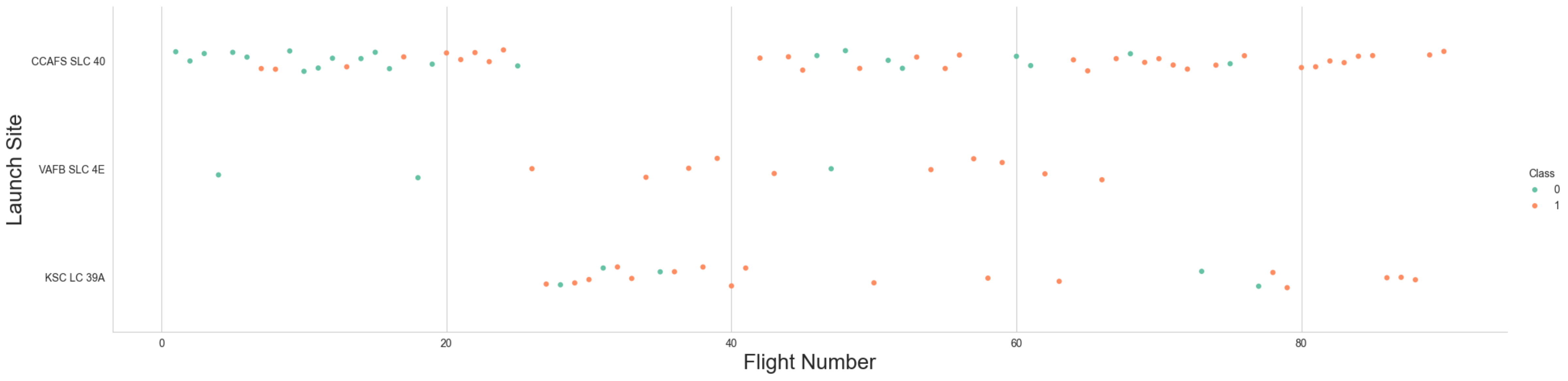
- From Interactive Dashboard we notice:
  - KSC LC-39A is the best site.
  - FT is the best boost version for lower values of Payload Mass

- We found that Logistic Regression is the best predictive model.

Section 2

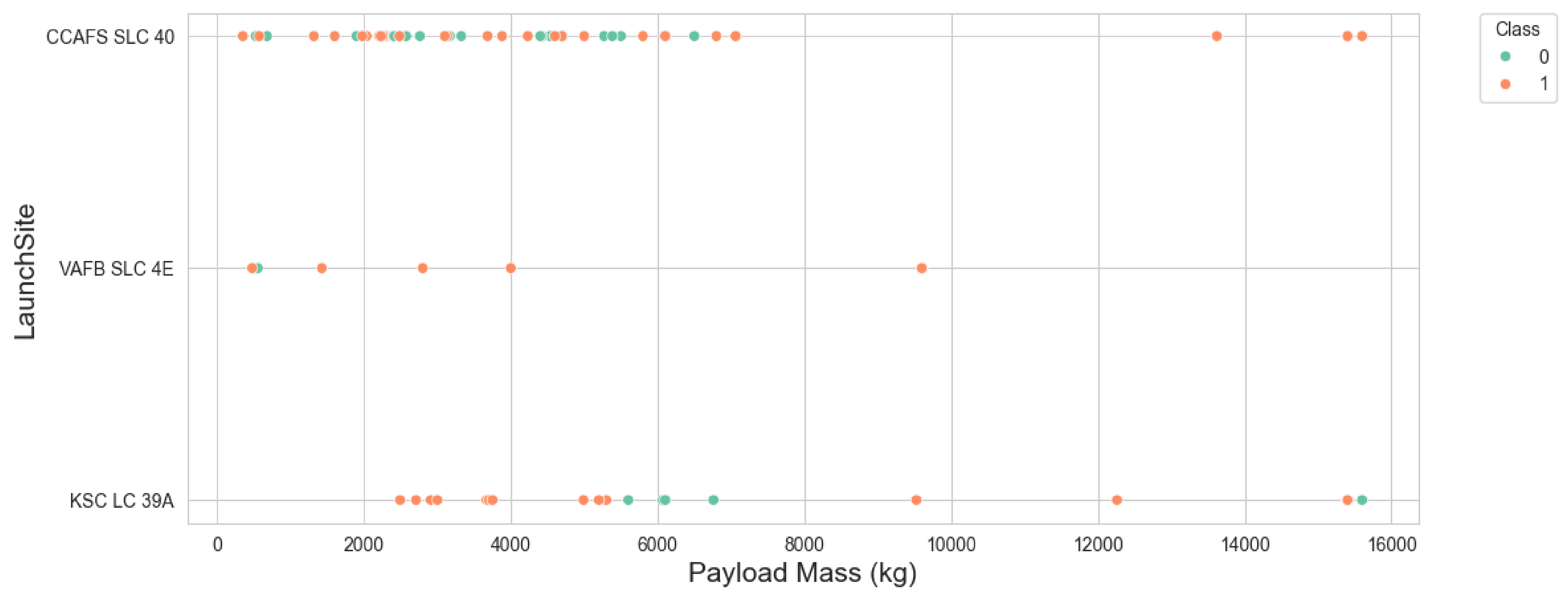
Insights drawn from EDA

Flight Number vs. Launch Site



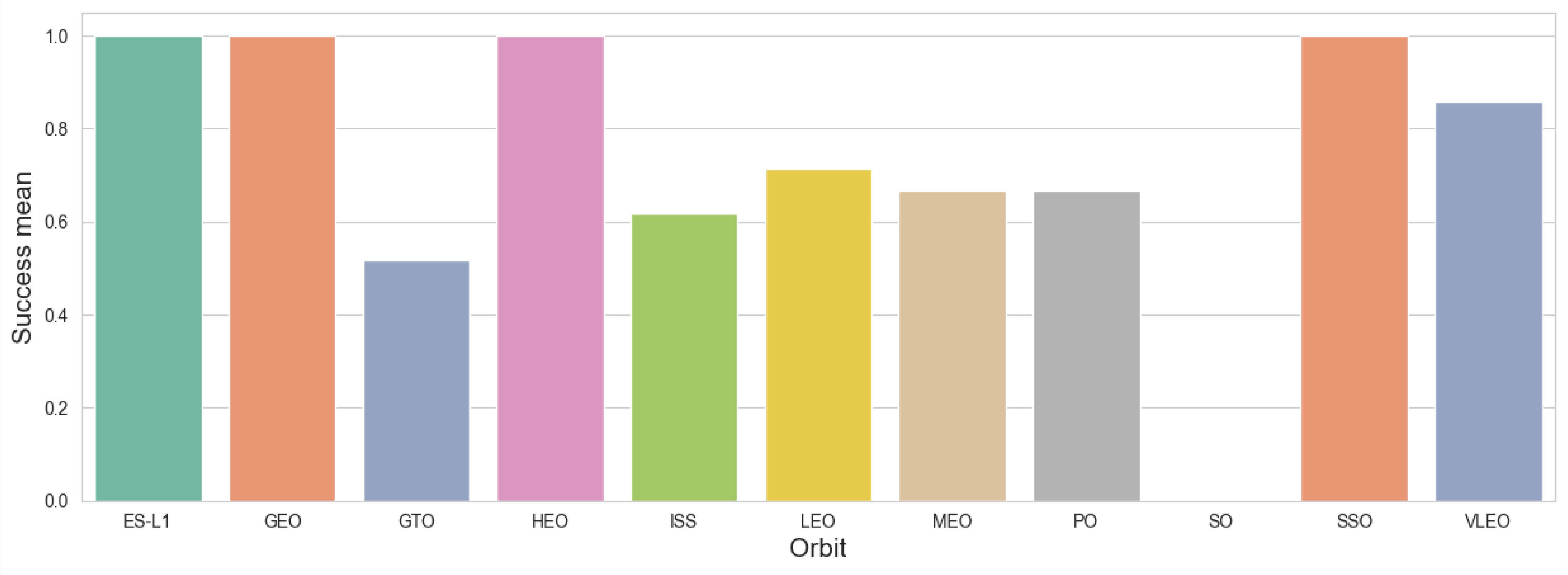
We see that at all sites, as the flight number increases, the successful landings increase. Another insight is that CCAFS SLC-40 was the test site; we can see at the other sites that there are fewer failed landings.

Payload vs. Launch Site



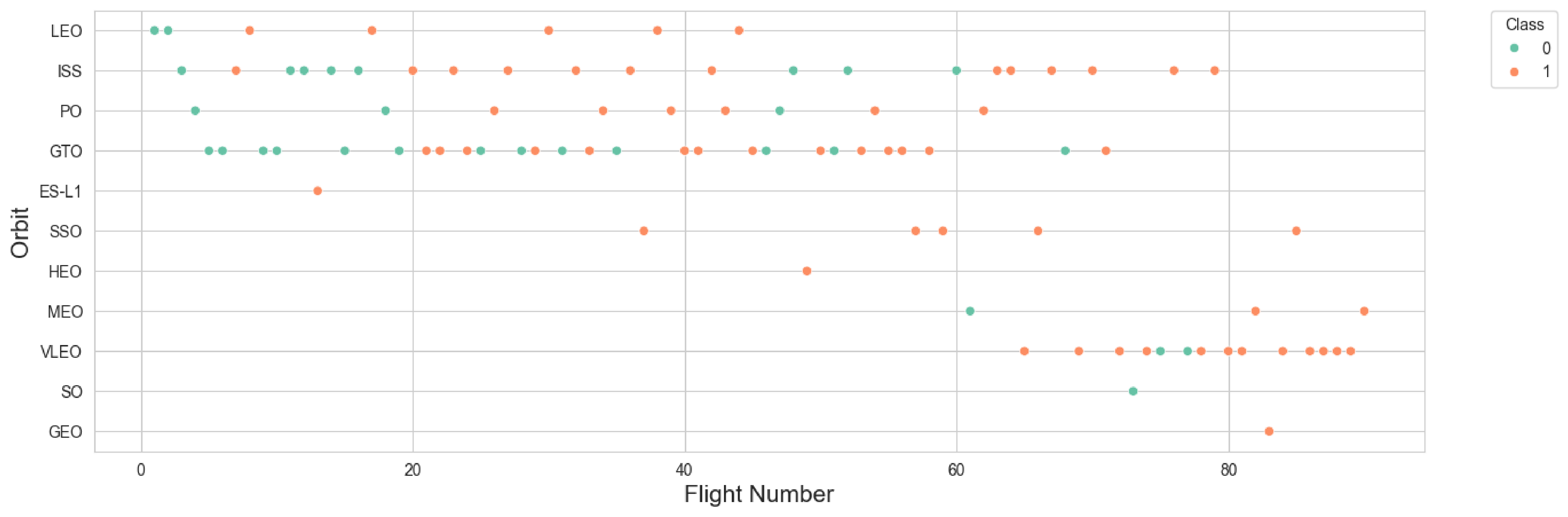
We see that the payload mass doesn't have a relationship with the launch site in terms of increasing the quantity of successful landings.

Success Rate vs. Orbit Type



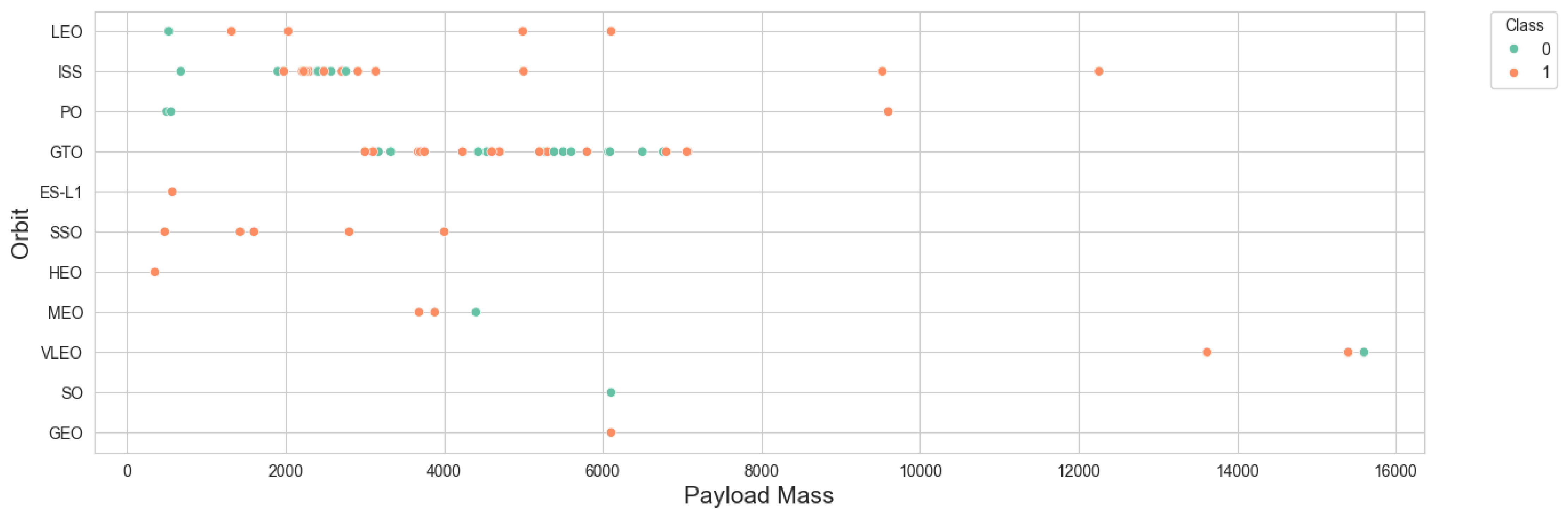
There are a high number of success landing un VLEO and SSO orbits. HEO and ES-L1 could be a good orbits but we need more information.

Flight Number vs. Orbit Type



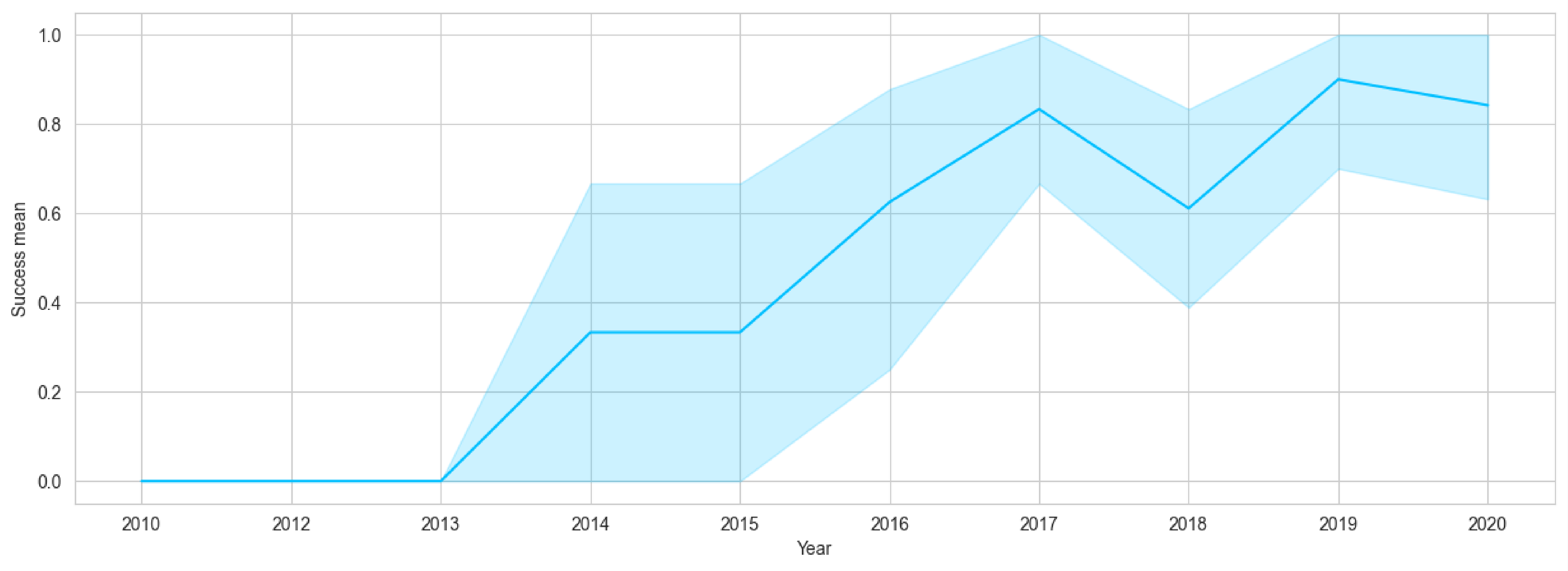
VLEO have better success landing in fewer Flight Number. LEO have the better in major. SSO don't has failed landing. HEO only has one flight number and it's successful. The other orbits have a mix of results.

Payload vs. Orbit Type



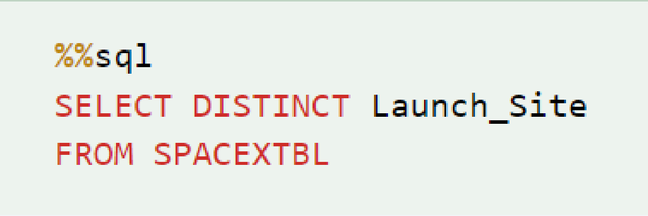
For LEO and ISS when the payload mass has incremented, the successful landing too. For SSO has successful landing in lower payload mass, but we don’t know what happen if it be increased.

Launch Success Yearly Trend

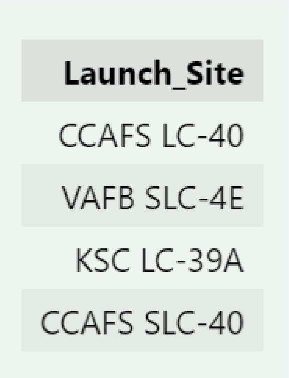


We can observe that the success rate since 2013 kept increasing till 2020.

All Launch Site Names



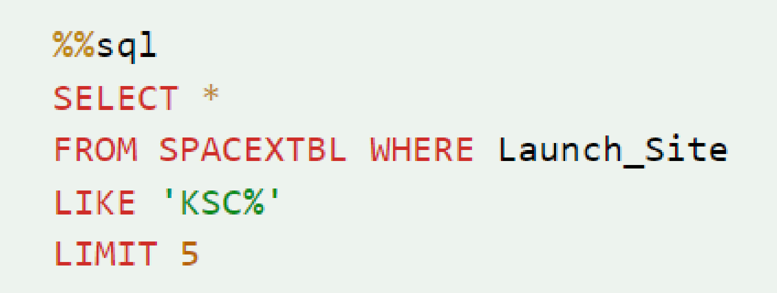
```
%%sql
SELECT DISTINCT Launch_Site
FROM SPACEXTBL
```



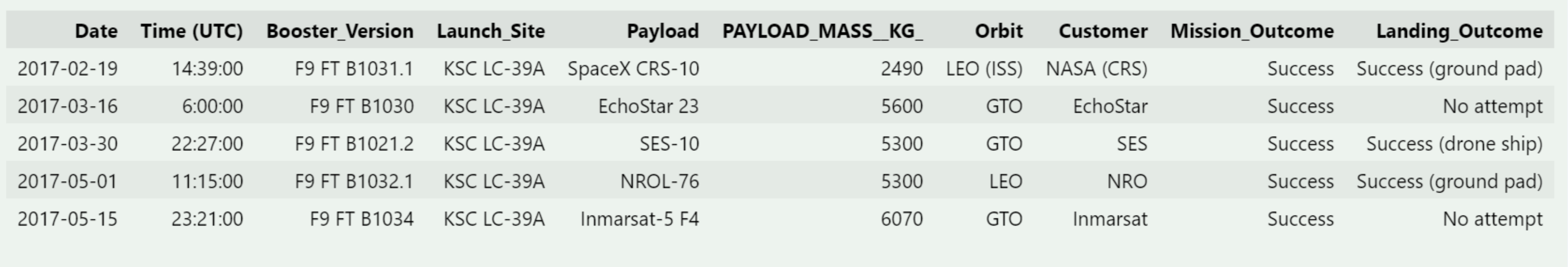
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

We use DISTINCT to find the Site Names.

Launch Site Names Begin with 'KSC'



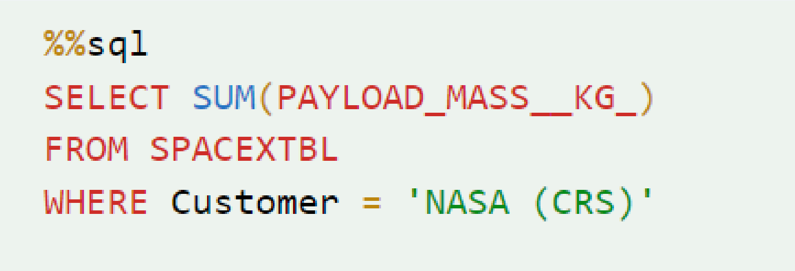
```
%%sql
SELECT *
FROM SPACEXTBL WHERE Launch_Site
LIKE 'KSC%'
LIMIT 5
```



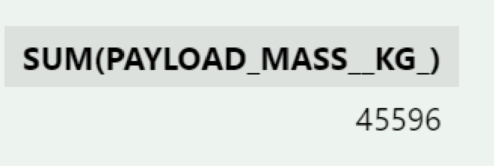
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

We use LIKE to only found KSC records and LIMIT 5 to find only five launch sites.

Total Payload Mass



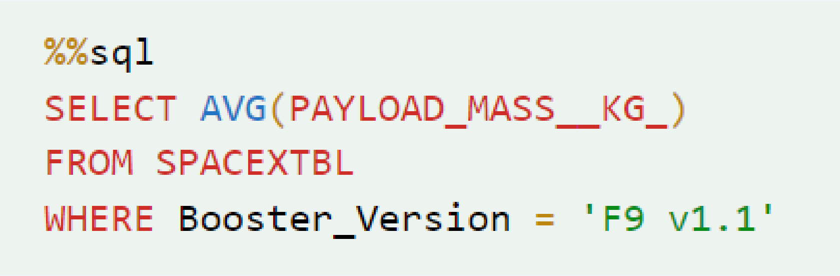
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)'
```



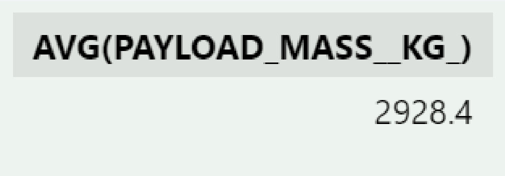
SUM(PAYLOAD_MASS__KG_)
45596

To find the total payload mass we use SUM function and use WHERE to define only for costumer NASA (CRS).

Average Payload Mass by F9 v1.1



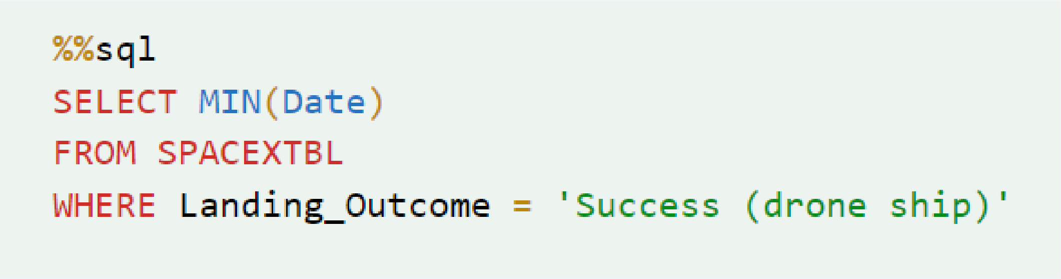
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version = 'F9 v1.1'
```



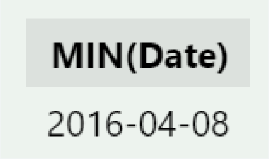
AVG(PAYLOAD_MASS__KG_)
2928.4

To find the average payload mass we use AVG function and use WHERE to define only for Booster version ‘F9 v1.1.

First Successful Ground Landing Date



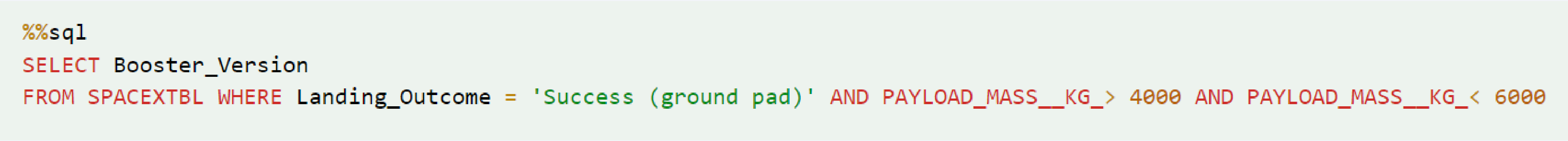
```
%%sql
SELECT MIN(Date)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (drone ship)'
```



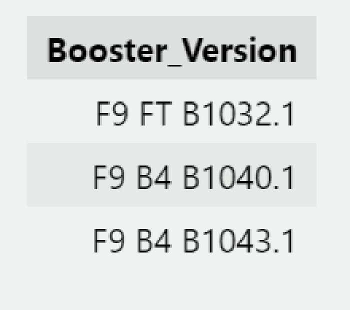
MIN(Date)
2016-04-08

To find the first date that have successful lading we use MIN function over Date column and use WHERE to define only for landing outcome ‘Success (drone ship)’

Successful Drone Ship Landing with Payload between 4000 and 6000



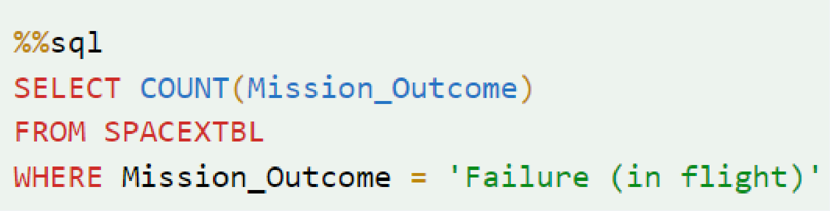
```
%%sql
SELECT Booster_Version
FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)' AND PAYLOAD_MASS__KG_> 4000 AND PAYLOAD_MASS__KG_< 6000
```



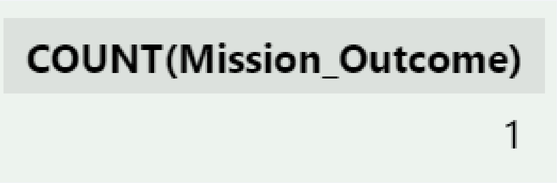
Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

We only SELECT Booster version, and use WHERE command on Landing Outcome equal to 'Success (ground pad)' for only successful landing. Then, use AND command to define range between 4000 and 6000.

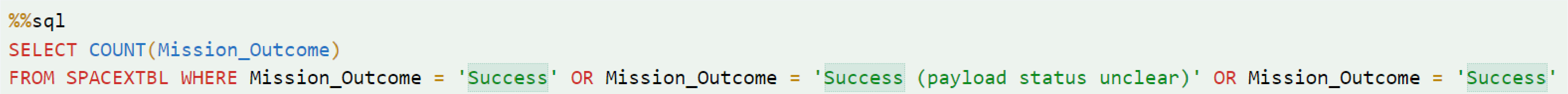
Total Number of Successful and Failure Mission Outcomes



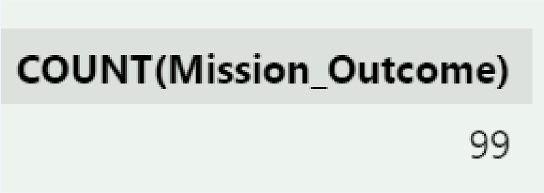
```
%%sql
SELECT COUNT(Mission_Outcome)
FROM SPACEXTBL
WHERE Mission_Outcome = 'Failure (in flight)'
```



COUNT(Mission_Outcome)
1



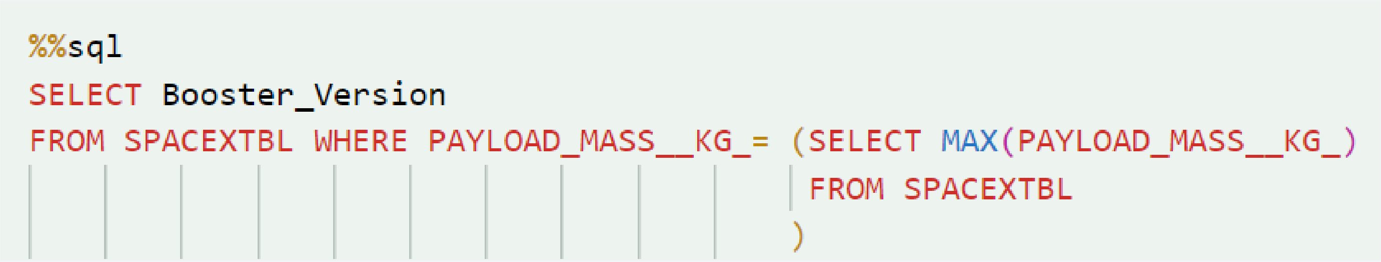
```
%%sql
SELECT COUNT(Mission_Outcome)
FROM SPACEXTBL WHERE Mission_Outcome = 'Success' OR Mission_Outcome = 'Success (payload status unclear)' OR Mission_Outcome = 'Success'
```



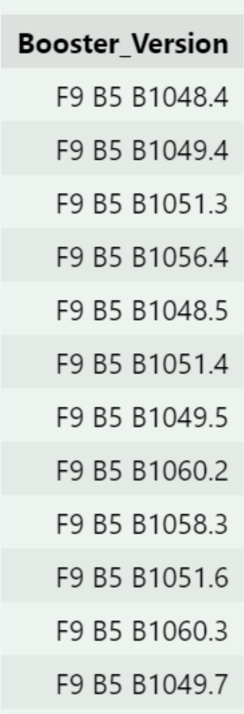
COUNT(Mission_Outcome)
99

We make two queries, one counting Mission Outcome equal to ‘Failure (in flight)’ that it means failed landing, other to count values ‘Success’ and ‘Success (payload status unclear)’, that they mean successful landing

Boosters Carried Maximum Payload



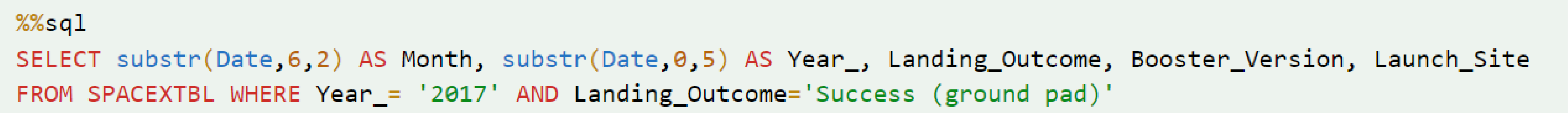
```
%%sql
SELECT Booster_Version
FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_)
                                          FROM SPACEXTBL
                                         )
```



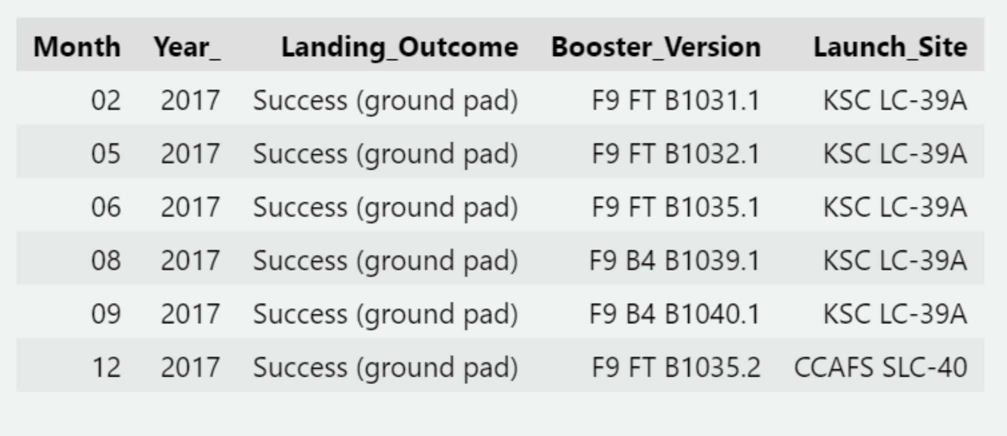
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

We make subquery to find MAX payload mass, and then make a query to find that value for every Booster version

2017 Launch Records



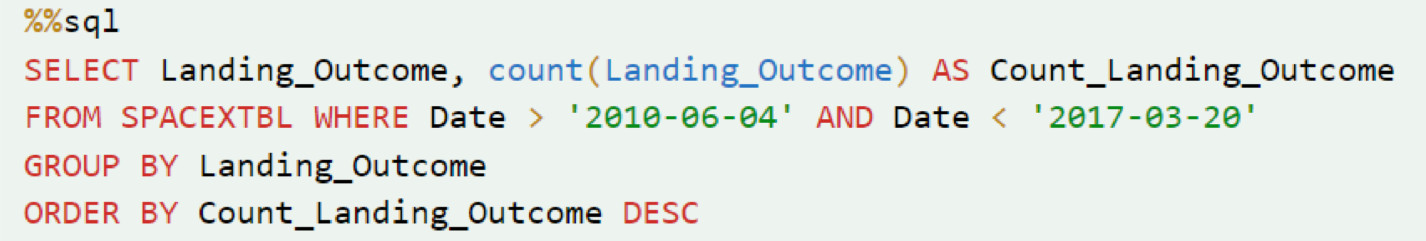
```
%%sql
SELECT substr(Date,6,2) AS Month, substr(Date,0,5) AS Year_, Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTBL WHERE Year_= '2017' AND Landing_Outcome='Success (ground pad)'
```



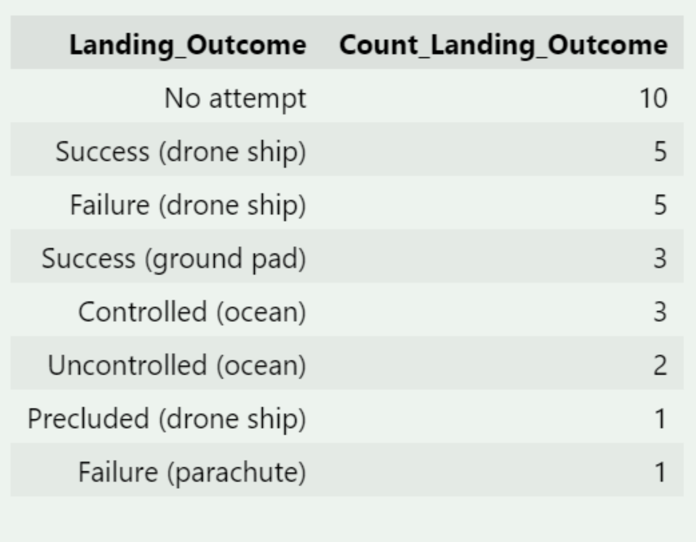
Month	Year_	Landing_Outcome	Booster_Version	Launch_Site
02	2017	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	2017	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	2017	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	2017	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	2017	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	2017	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

For obtain the month we extract from column ‘Date’. With the command WHERE define the year 2017 and landing outcome = ‘Success (ground pad)’.

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



```
%%sql
SELECT Landing_Outcome, count(Landing_Outcome) AS Count_Landing_Outcome
FROM SPACEXTBL WHERE Date > '2010-06-04' AND Date < '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Count_Landing_Outcome DESC
```



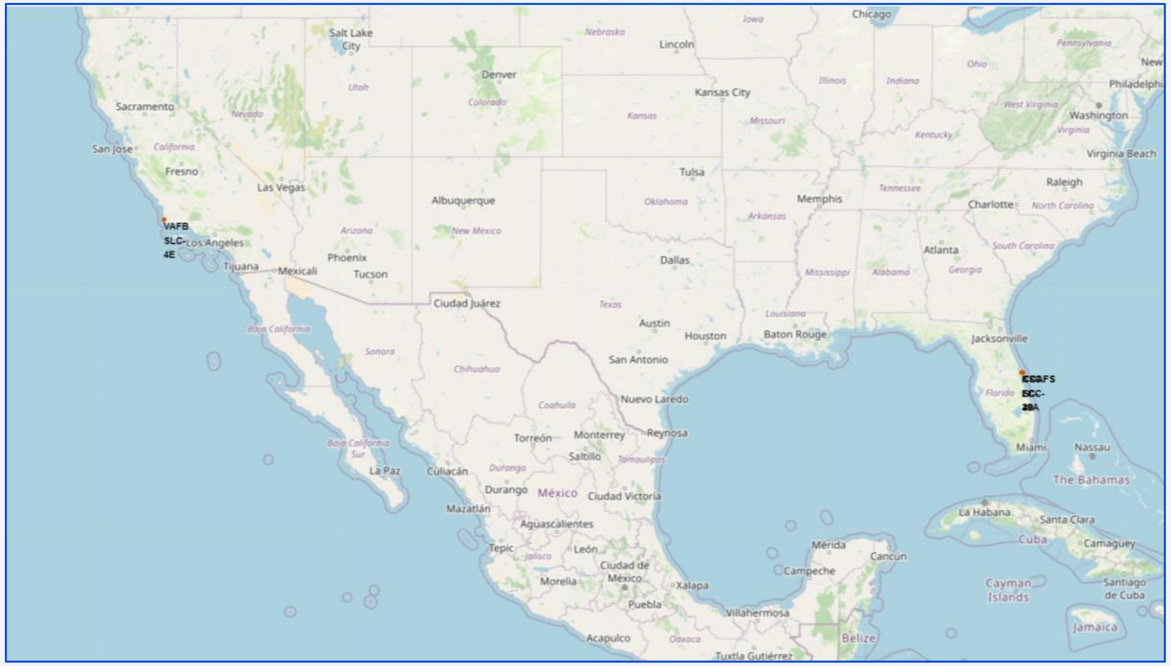
Landing_Outcome	Count_Landing_Outcome
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

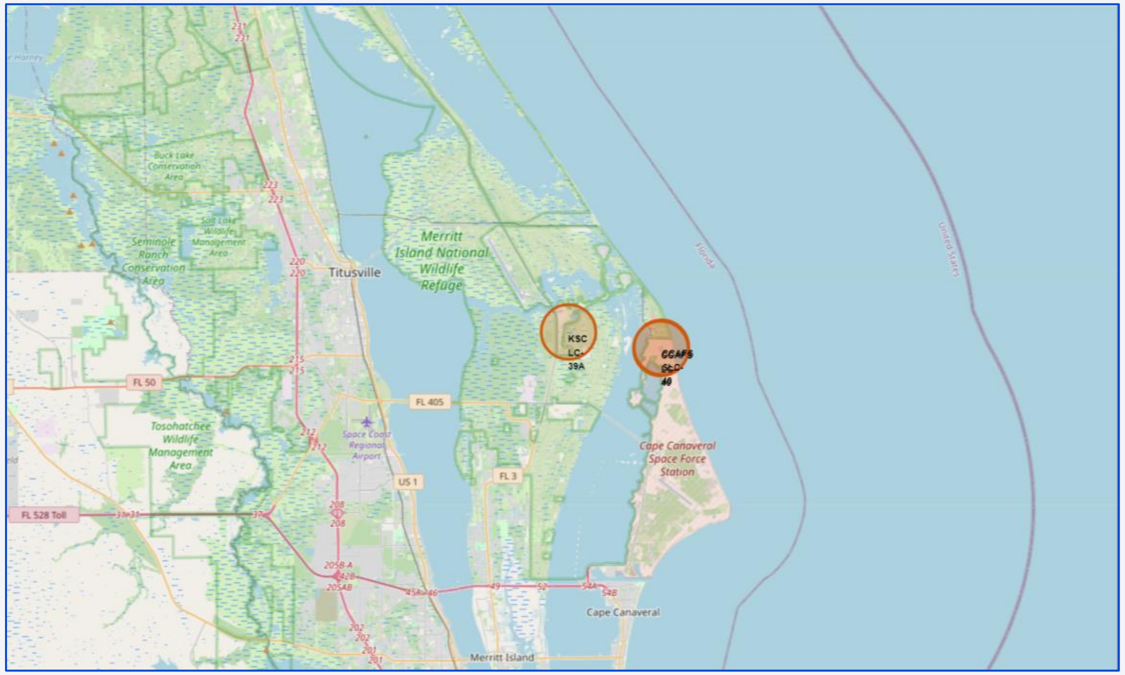
First, we count Landing outcome on the dates specified, then use the GROUP BY command, and fin finally use the ORDER BY command to see the records in descendant order.

Section 3

Launch Sites Proximities Analysis

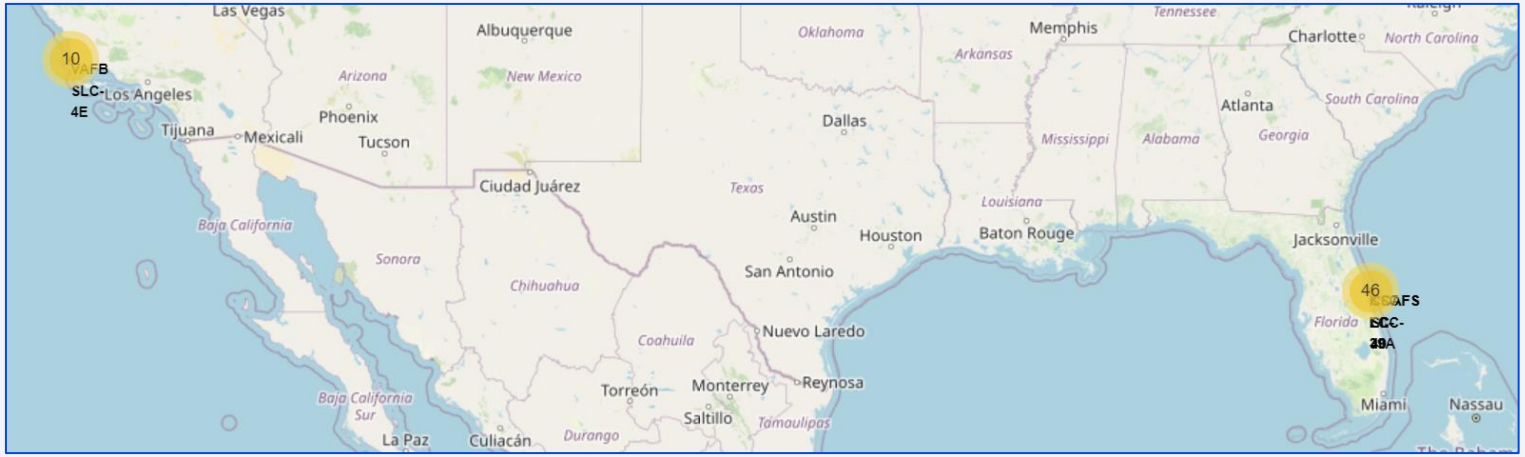
Ubicating Launch Sites on the map



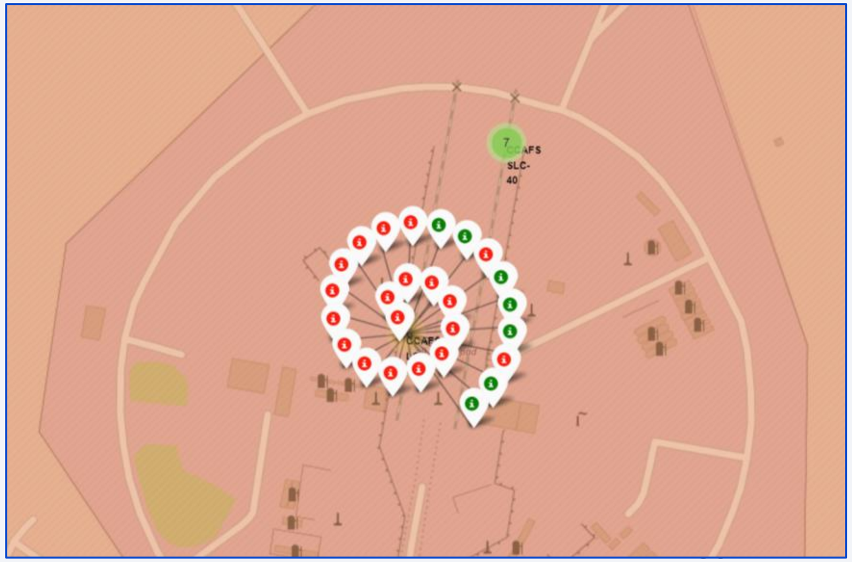

KSC
LC-
39A
CCAFS
SLC-
40
Titusville
Merritt Island National Wildlife Refuge
Seminole Ranch Conservation Area
Buck Lake Conservation Area
Salt Lake Wildlife Management Area
Tosohatchee Wildlife Management Area
Space Coast Regional Airport
Cape Canaveral Space Force Station
Cape Canaveral
Merritt Island
FL 50
FL 405
FL 3
US 1
FL 528 Toll

First we ubicate the launch sites on the map using circles form and identify them by their name

Mark the successful and failed launches

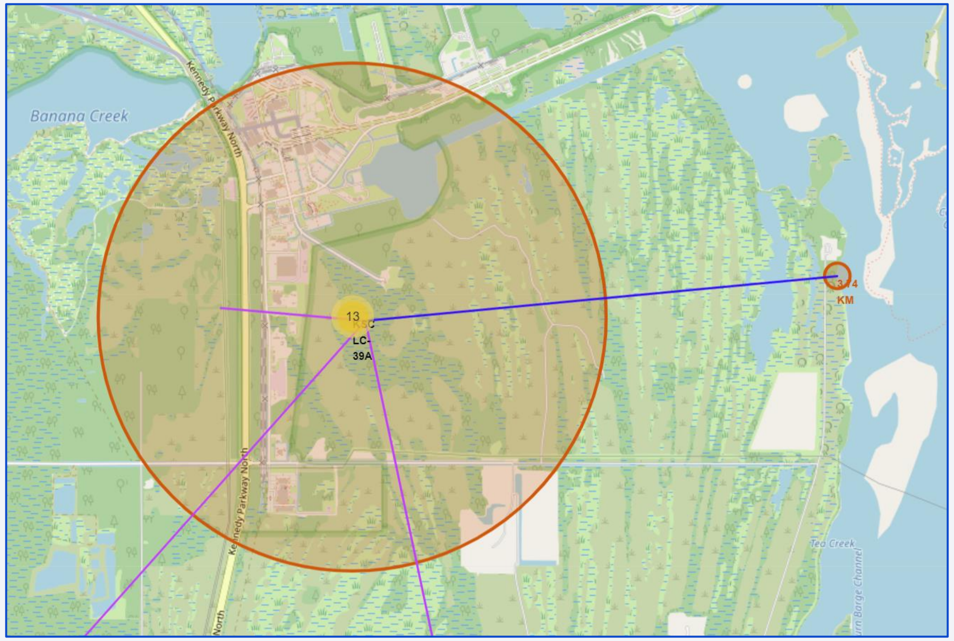


13
LC-39A
CCAFS

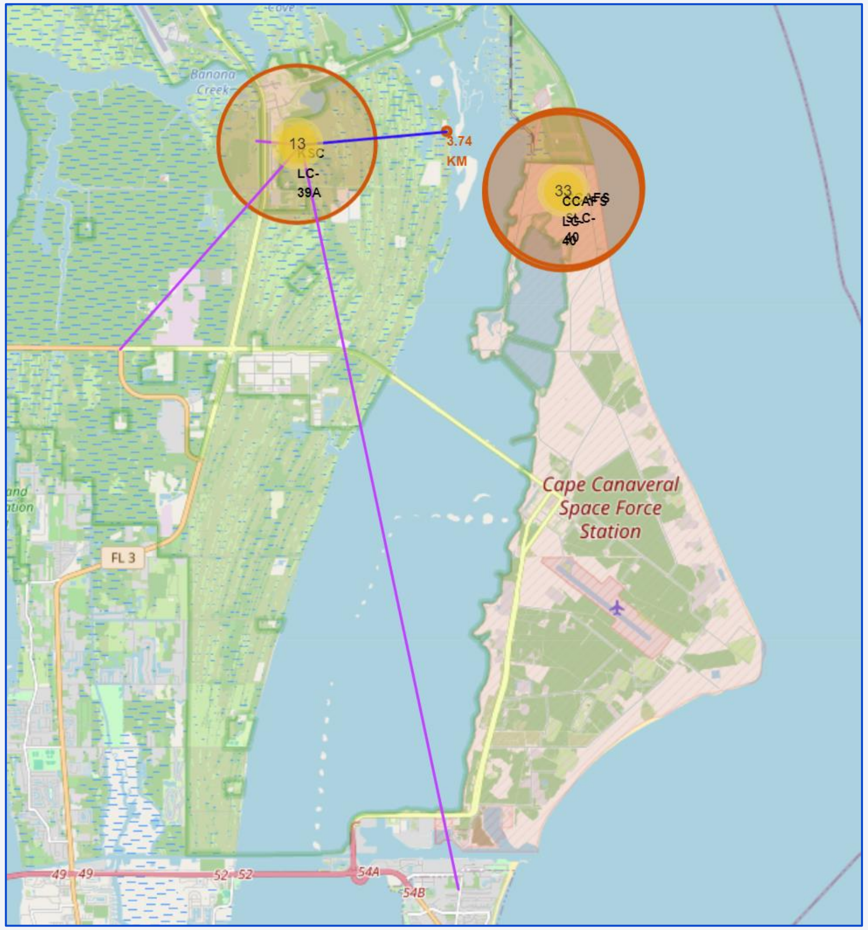


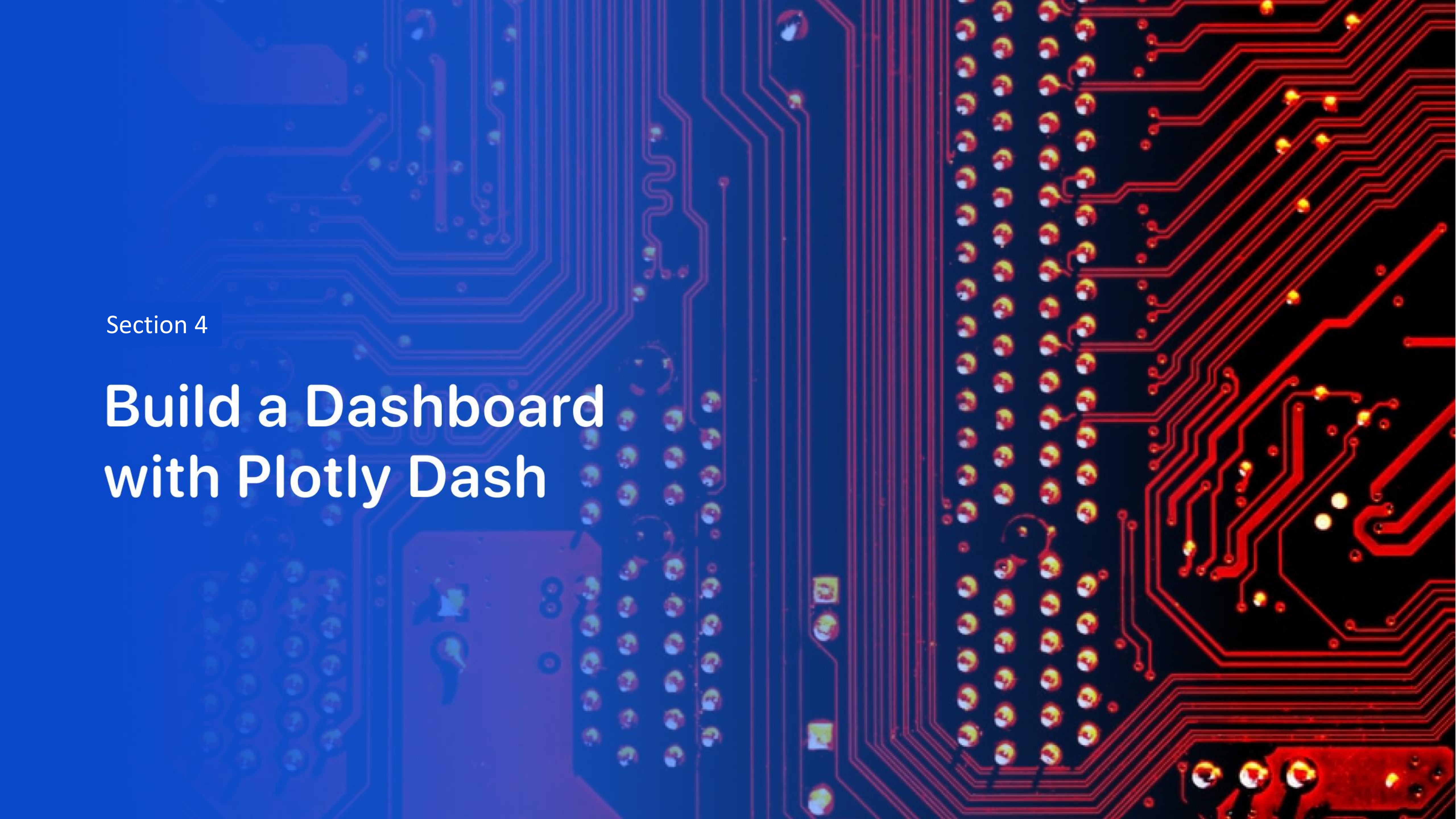
Then, we mark the sites with the successful launches with green marker, and failed launches with the red color.

Launch proximities



Finally, we ubicate the launch proximities like the coast, railway, highway and the coastline, and draw a line between an any launch site to them.

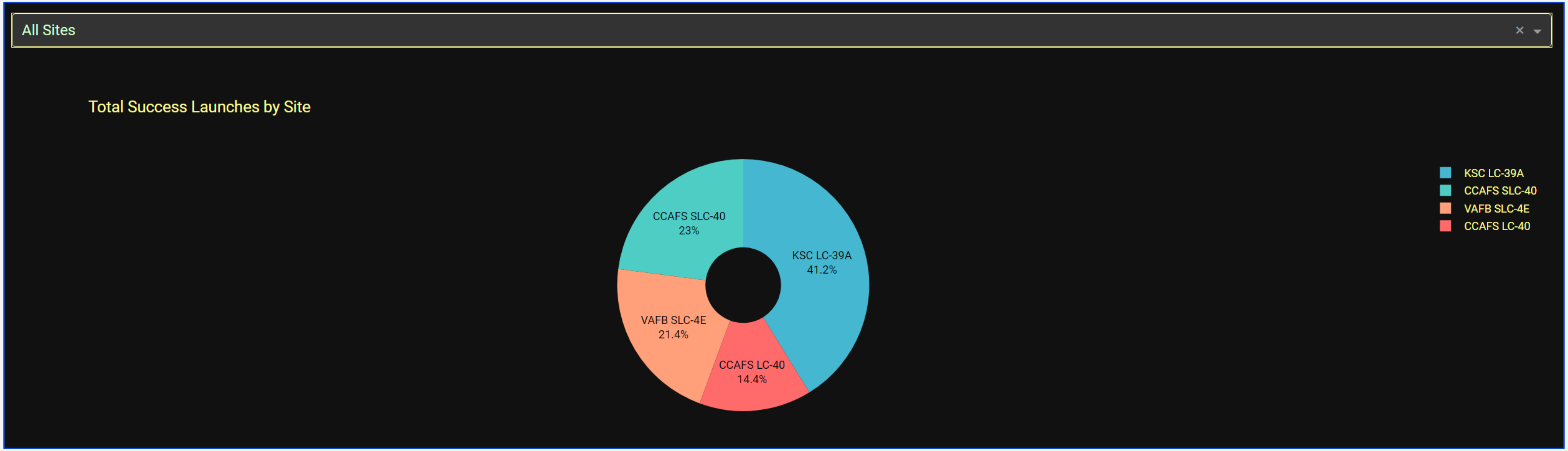




Section 4

Build a Dashboard with Plotly Dash

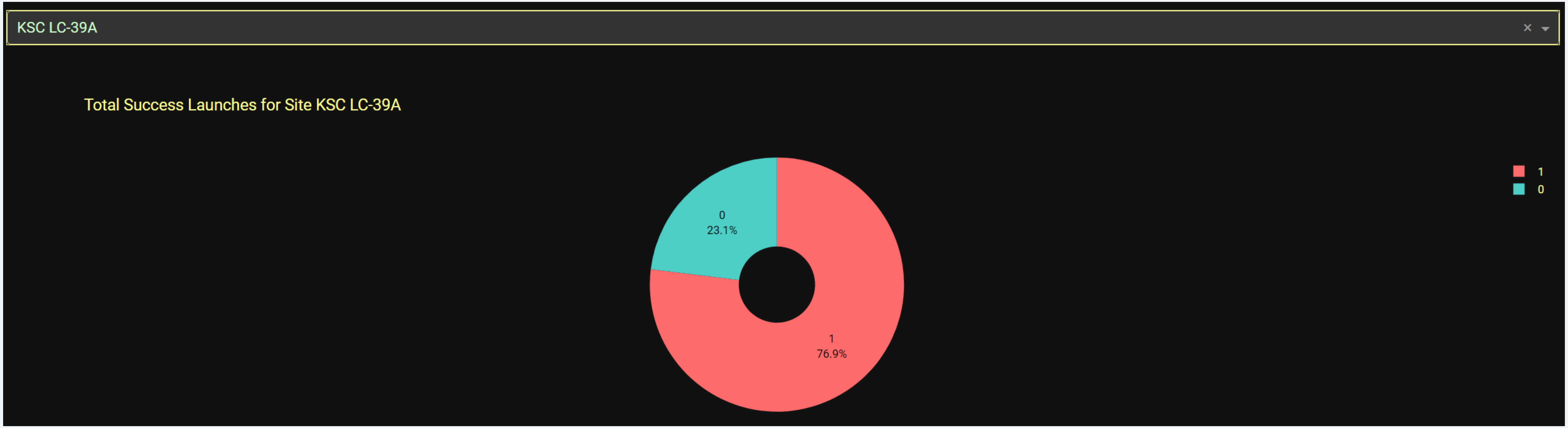
Launch Sites Pie Chart



All Sites

KSC LC-39A is the best site with the 412% of successful landing

Pie Chart of KSC LC-39A

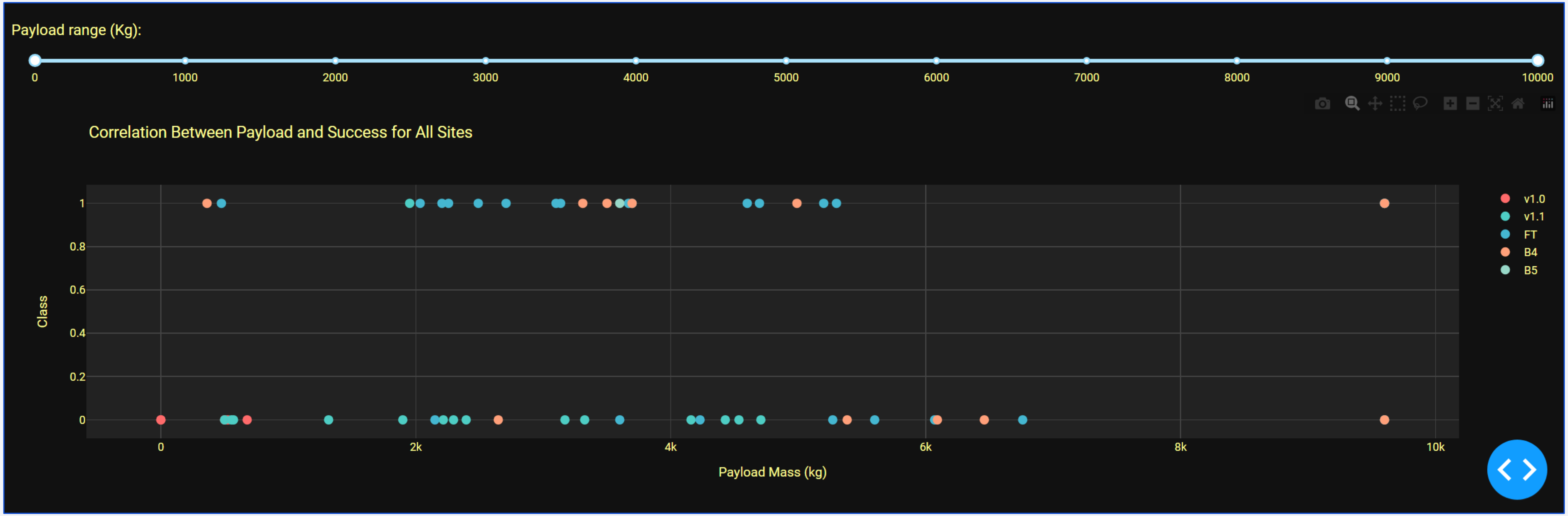


KSC LC-39A

Total Success Launches for Site KSC LC-39A

For KSC LC-39A has the 76.9% of successful landing

Payload Mass vs. Launch Outcome

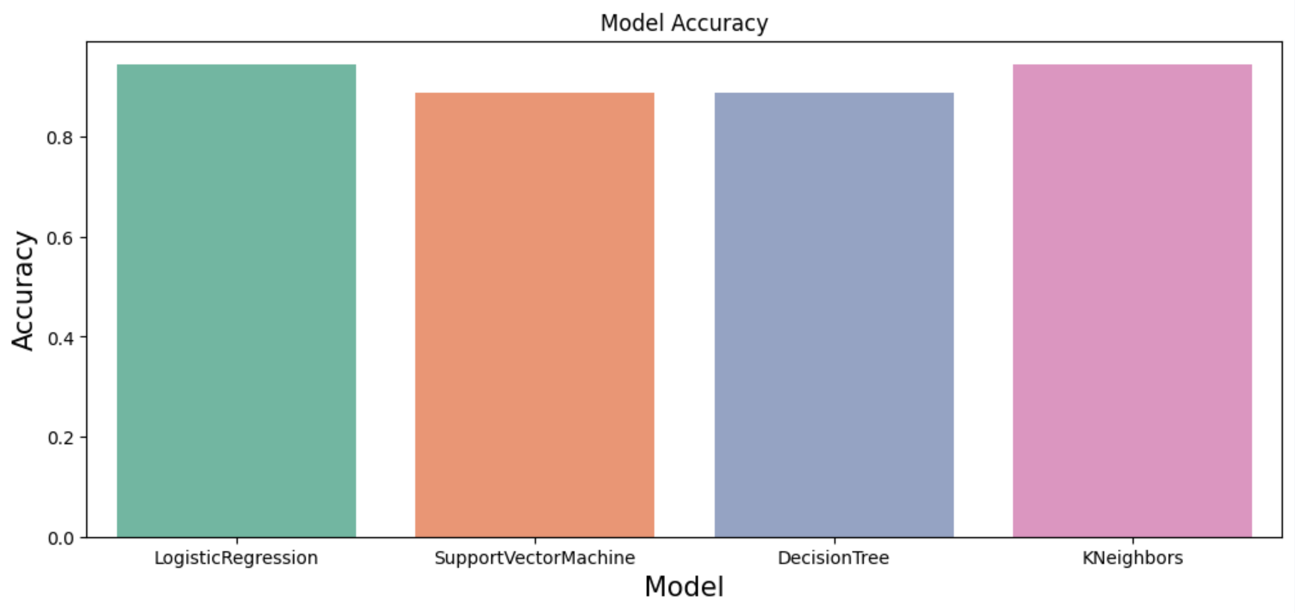


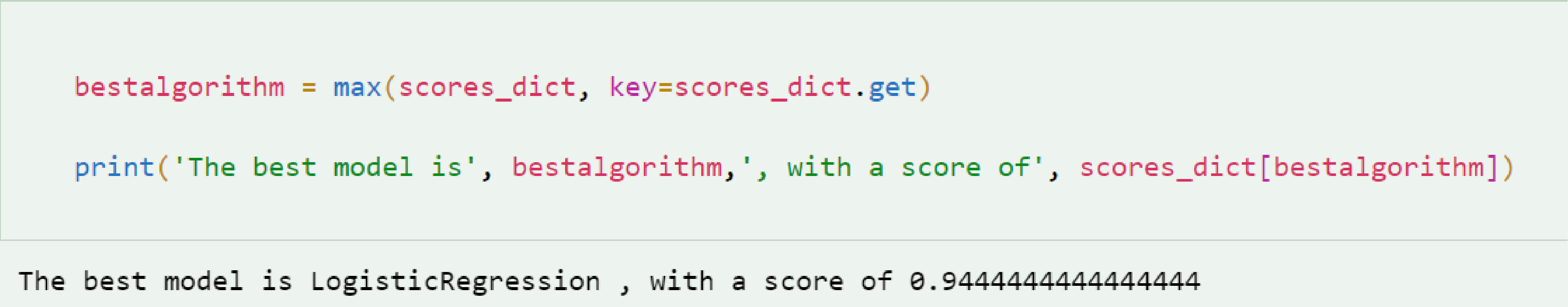
FT is the best boost version for lower values of Payload Mass for all range, and V1.1 has the worst

Section 5

Predictive Analysis (Classification)

Classification Accuracy



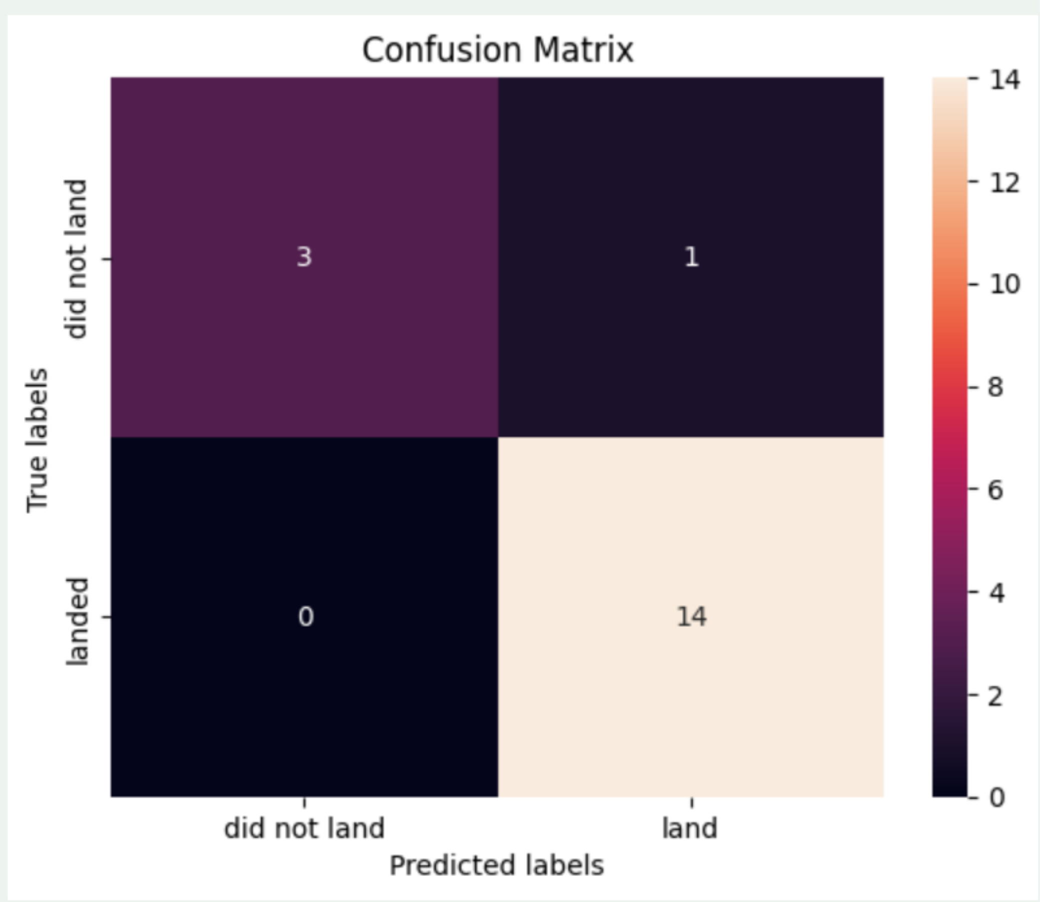


```
bestalgorithm = max(scores_dict, key=scores_dict.get)

print('The best model is', bestalgorithm,', with a score of', scores_dict[bestalgorithm])
```

```
The best model is LogisticRegression , with a score of 0.9444444444444444
```

Confusion Matrix



Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

Conclusions

- The main features that influence successful landing are Launch site, Payload mass, booster version, and Orbit. This one is the external factor that can affect the landing outcome. The success rate since 2013 kept increasing till 2020.
- The orbits with the high number of success landing are VLEO and SSO.
- Launch site KSC LC-39A has the best successful landing.
- The best model for prediction is Logistical Regression and we can determine the landing outcome with 94.44% of accuracy.

Appendix

- Data folder:
- Notebooks folder:
- Saved model folder:

Thank you!