IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Steven Morgan
5/23/2024

# Outline

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

# Executive Summary

- We began our analysis by collecting data on SpaceX. There are two sources of data that we need. SpaceX own API, and Wikipedia. Both sources have most of the same data, but Wikipedia has some extra values that are useful. We then process and clean the data, getting it in a state that is allows us to use it. We then perform our exploratory data analysis(EDA) to find any features that we might use for our predictions. Then we run it through some predictive models to find if we can predict the landings.
- With our tree_cv model we were able to get a predictions success of 88% and some possible relationship between payload and orbit that could influence success.

# Introduction

- Spaceflight is expensive, With the improvements of landing and reusing the first stage of a rocket we can make spaceflight cheaper and safer. Cheaper flights will increase our ability to colonize other planets, Mine asteroids and make discoveries about our universe.
- The problem with reuse is that there is a rate of failure when landing that we need to predict and reduce. This requires us to have a firm and thorough understanding of how our rockets fail. We will address this by analyzing SpaceX launches and build a model to predict success and failure of the first stage landing.

Section 1

# Methodology

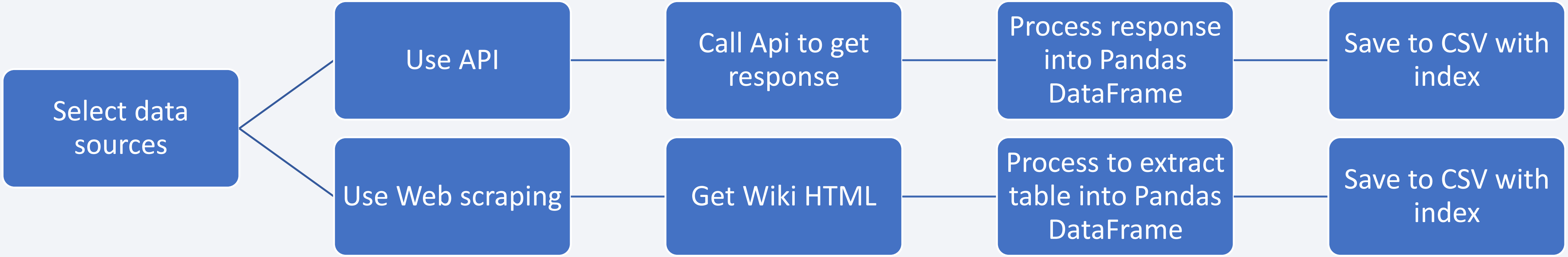
# Methodology

## Executive Summary

- Data collection methodology:
  - Data collection using SpaceX API
  - Data collection using Wikipedia to web scrape the table of launches
- Perform data wrangling
  - Performed data cleaning and simplifying the data so that the values are better processed during predictive analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Performed a series of predictive analysis to see what models performed the best. The results were compared to show which performed the best.

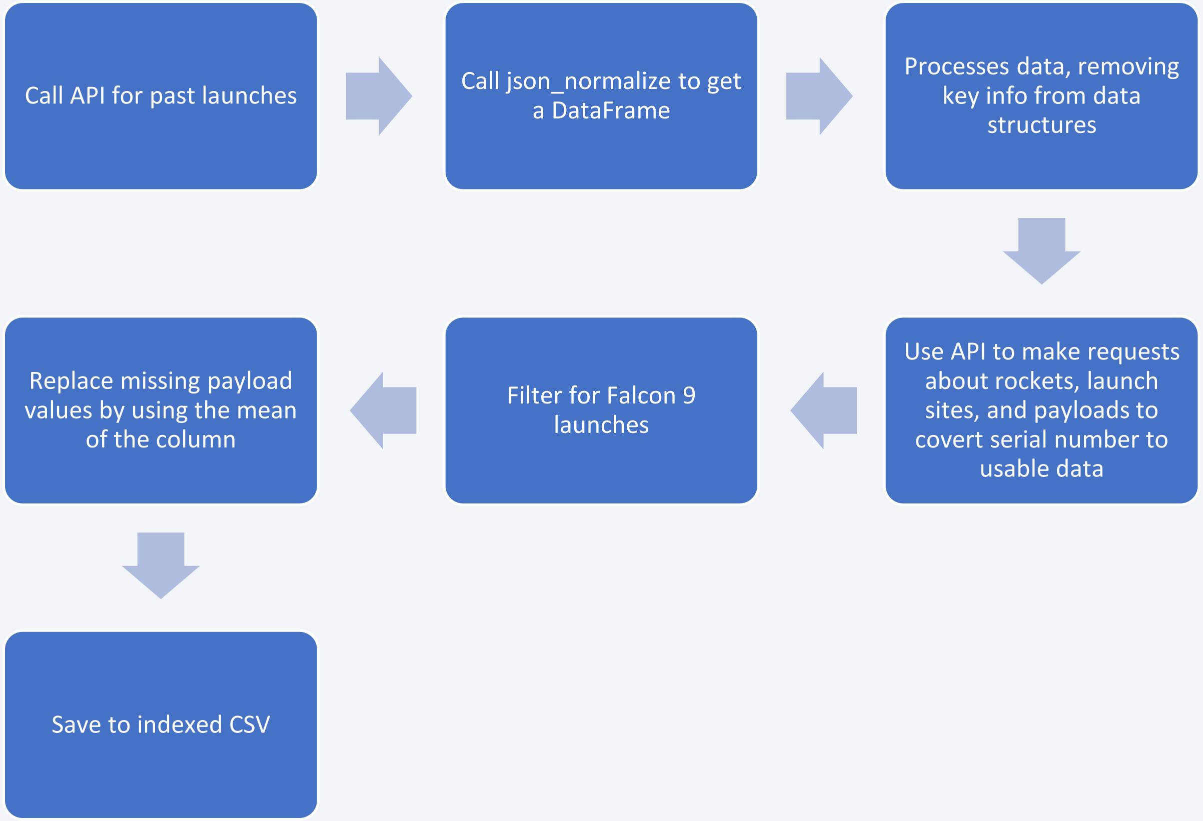
# Data Collection

- The data collection was done in two ways. First the SpaceX API was used to download the launch data. This gives us data from SpaceX themselves. Second is the data scraped from the wiki page on SpaceX, Using the BeautifulSoup library to process the html we can extract the table we need with launch data.



# Data Collection – SpaceX API

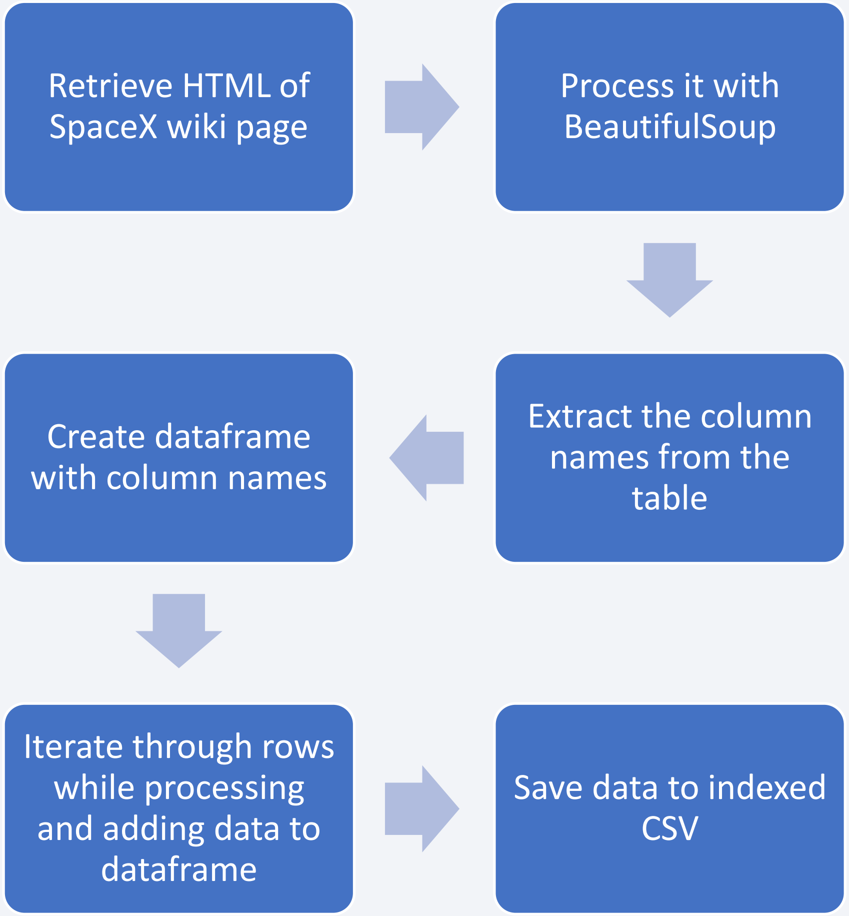
- Our collection from the SpaceX API is quite simple. They API allows us to request past launches. These are delivered in a JSON object which is easy to convert into a DataFrame.
- With that DataFrame we can process the data, cleaning things to make it easy to processes for our models later.
- Rockets payloads, and launch sites are listed by serial numbers. To get human friendly names, we call the API and request details about specific details. Cross referencing this we update the values in our main data set.
- We then filter out the non-Falcon 9 rockets since we are only concerned about their performance.
- We then replace the missing values in the Payload column by using the mean value of the column. This is to keep these entries while not skewing the data.
- We then save our results to an indexed CSV.


Call API for past launches
Call json_normalize to get a DataFrame
Processes data, removing key info from data structures
Use API to make requests about rockets, launch sites, and payloads to covert serial number to usable data
Filter for Falcon 9 launches
Replace missing payload values by using the mean of the column
Save to indexed CSV

https://github.com/bobthebaka/DS-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection - Scraping

- Our process of collecting data from the Wiki page is made easy by the library ‘BeautifulSoup’. After doing a simple request of the page we have the HTML. BeautifulSoup will parse it and give us an object we can query.
- To get the table info we want, We query for the 3rd table and check the contents.
- We extract the table rows into a dictionary of lists, by looping through all rows. That dictionary is then used to create a DataFrame
- We then save the DataFrame to an indexed CSV


Retrieve HTML of SpaceX wiki page

Process it with BeautifulSoup

Extract the column names from the table

Create dataframe with column names

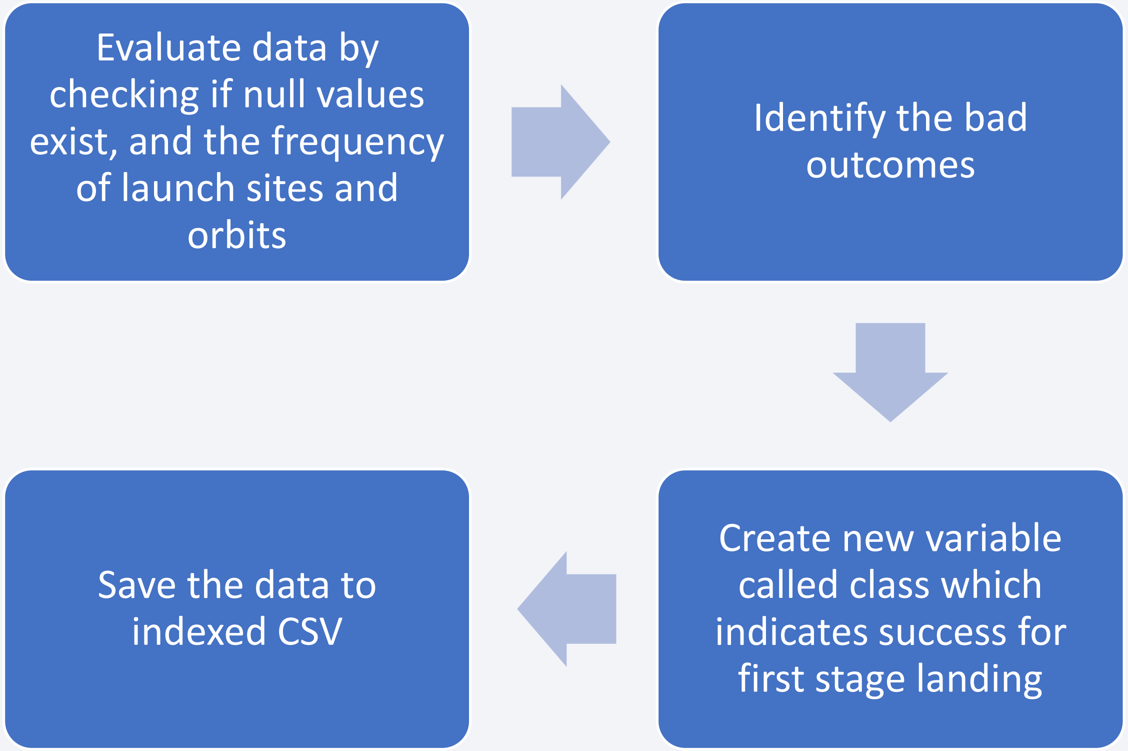
Iterate through rows while processing and adding data to dataframe

Save data to indexed CSV

https://github.com/bobthebaka/DS-Capstone/blob/main/jupyter-labs-webscraping.ipynb

# Data Wrangling

- We first check the data to get an idea of the distribution. Then we look for null values.
- One thing that we notices is that the landing outcome has 8 different outcomes, 5 of which can be considered failure. This won't be helpful for predictive analysis which is much better with a binary value.
- We then take the bad outcomes and create a new column named class. It is set to 1 if the first stage lands and 0 for everything else.
- The new DataFrame is then saved to an indexed CSV



Evaluate data by checking if null values exist, and the frequency of launch sites and orbits

Identify the bad outcomes

Create new variable called class which indicates success for first stage landing

Save the data to indexed CSV

https://github.com/bobthebaka/DS-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with Data Visualization

- The charts that were used are Scatter plot, Bar Chart, and Line graph. These plots were used to illustrate the various aspects of the data.
- The scatter plots were helpful because they made visualizing the success of the rocket launches with the comparison of two variables. This made groups of failures much easier to see as there would be groups of different color dots to compare.
- The bar chart made it easy to see the success rate of different orbit types. Allowing us to have an easy and precise comparison between discreet values.
- The line chart was useful for showing the change of success by year. We can see that SpaceX has been increasing their rocket's success every year with few exceptions.

https://github.com/bobthebaka/DS-Capstone/blob/main/edadataviz.ipynb

# EDA with SQL

- Using SQL queries to identify the distinct launch sites used by SpaceX
- We query for launches that happened at any site that started with ‘CCA’. Allowing us to find all launches in an area
- We query for the total payload mass carried for a customer. Allowing us to see who is sending the most to space with SpaceX
- We query for the average payload by booster type. This might give insight into what SpaceX is more comfortable with the rocket carrying.
- We query for the earliest successful landing of a booster.
- We queried for the boosters that successfully landed on a drone ship with a payload in the range or 4000 to 6000 kg. This would be helpful if we needed to see what is succeeding when another version might be failing in a similar task.
- We queried for the count of successes and failures. Allowing us to have a ratio of success
- We queried for the boosters that carried the max payloads. Allowing us to compare the booster versions.
- We queried for the count of landings outcomes in a date range. Allowing us insight to a timeframe that might show the result of different manufacturing processes.

https://github.com/bobthebaka/DS-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

- We added a circle markers to the map to each launch site. This helps differentiate markers on the map. The circle helps show the launch site and area around it.
- We used markers to indicate the launches. The markers color was set to reflect the success or failure. The markers are at the same location, causing them to be stacked and represented by a number.
- Lines with markers are used to measured the distance from the launch site. The line and marker color is different for each feature that is being measured against. The marker has a popup that will show the distance from the launch site.

https://github.com/bobthebaka/DS-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

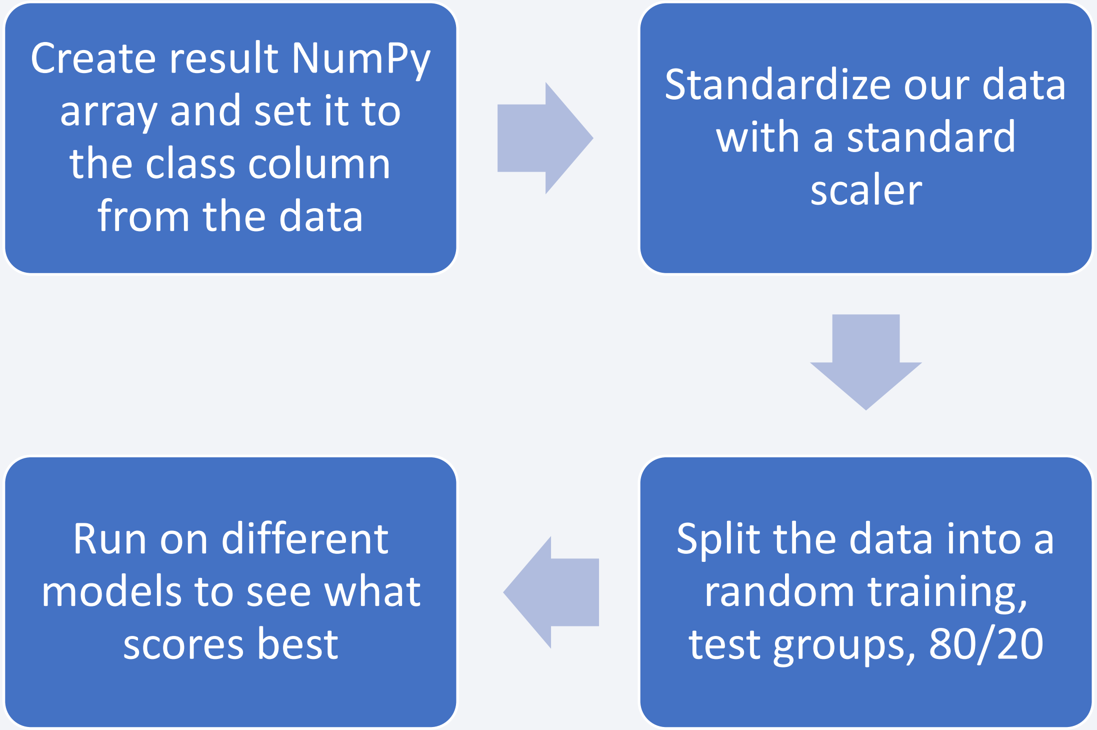
# Build a Dashboard with Plotly Dash

- The dashboard consist of two graphs, the first is a pie chart, second is a scatter plot. There are two controls, one allow you to select a launch site and the second a payload range.
- The pie chart works with the launch site selector. If you select all it will show a breakdown of the different pads with their success percentage in relation to all successes. If you select a pad, it shows the success and failure percentages for just that pad.
- The Scatter plot works with both the scatter plot and range. It will show the different pads with different color dots. The X axis is the payload, and the y is the success. If the range is set only those launches that had a payload in that range are shown.

https://github.com/bobthebaka/DS-Capstone/blob/main/spacex_dash_app.py

# Predictive Analysis (Classification)

- To perform predictive analysis, we start with preparing our value we are predicting, in this case its class. We create a NumPy array that will tell if the rocket landed successfully.
- Second, we need to prep the data by setting it to a value in a standard size. The standard scaler will do that by setting it from -1 to 1.
- Third, we split our data into test and training sets.



Create result NumPy array and set it to the class column from the data

Standardize our data with a standard scaler

Split the data into a random training, test groups, 80/20

Run on different models to see what scores best

https://github.com/bobthebaka/DS-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

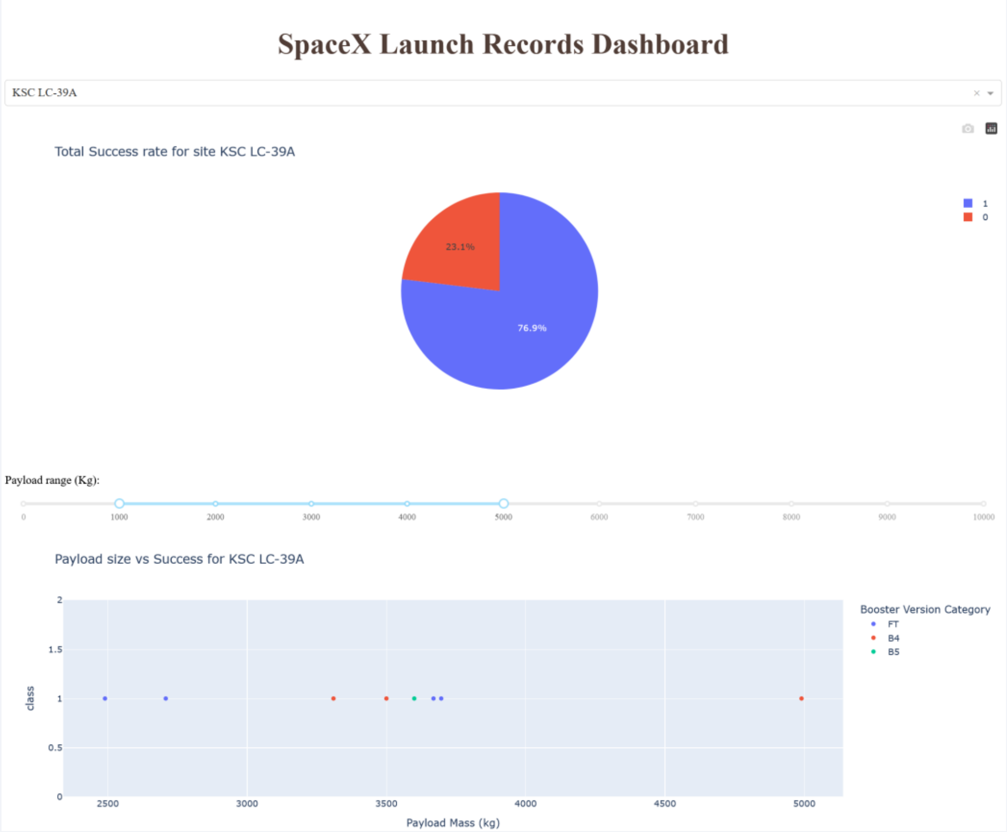
# Results

- During EDA we were able to see trends in the data regarding payload and orbit type. There looks to be a higher failure rate for payloads around the 4-6k kg range. This could be skewed by the failures for early landings.

- For the plot of Orbit type vs Payload, we see that GTO orbits have a higher rate of landing failure. This could be explained by the energy need to achieve the orbit. GTO is a high orbit around halfway to the moon. Getting there requires more from the rocket, leaving it with little to no fuel to land

- We also see that early rocket designs failed quite often which is expected for a new product. There were 22 launches before the first successful landing on a drone ship. These failures should be considered in their own category as SpaceX was still learning how the land the first stage.

- With the predictive analysis we were able to get an 88% accuracy rate of predicting landing success.



SpaceX Launch Records Dashboard

KSC LC-39A

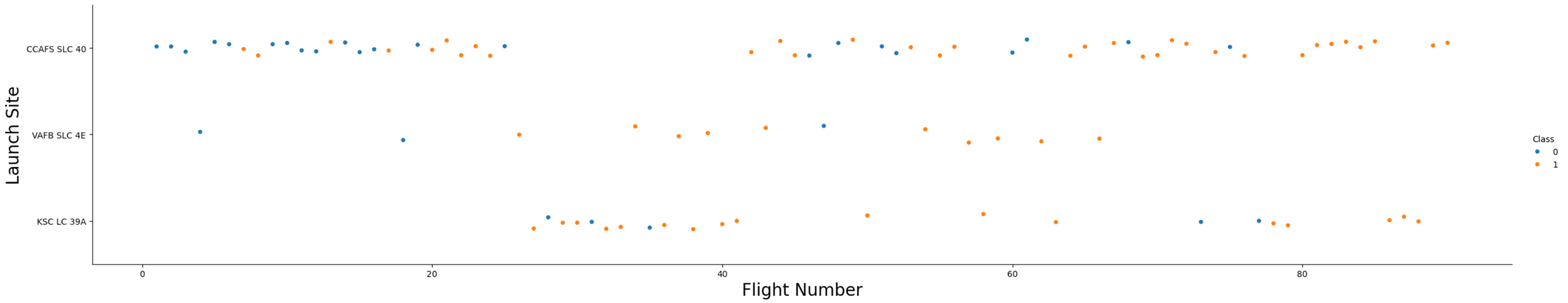
Total Success rate for site KSC LC-39A

Payload range (Kg):

Payload size vs Success for KSC LC-39A

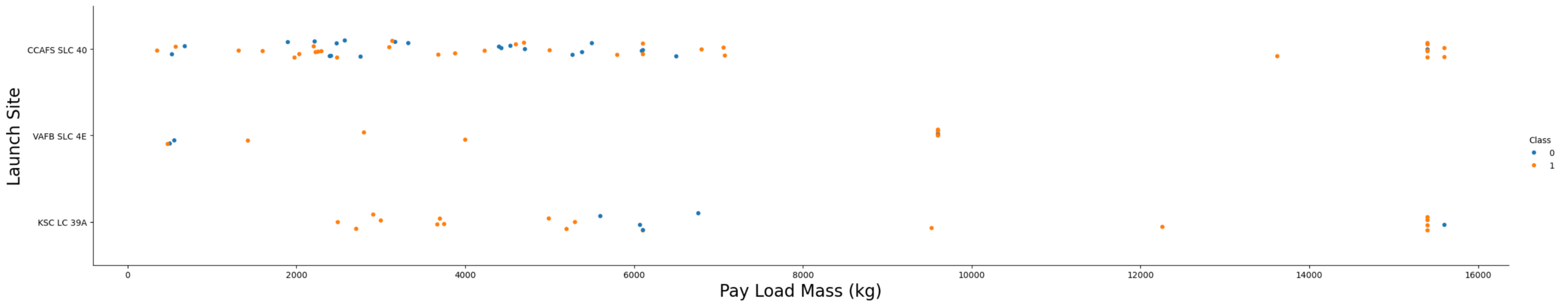
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

- This scatter plot shows us more about where the rockets that failed were in place and time. The flight number is in order of launch so we can see the earliest launches are on the left. Those are also the launches with the highest failure rate.
- We can also see that the most failures came from the most used site.
- This shows that even if CCAFS SLC-40 has the highest failure rate, its probably not the site, but the nature of more launches and the rockets had a higher failure rate early on.

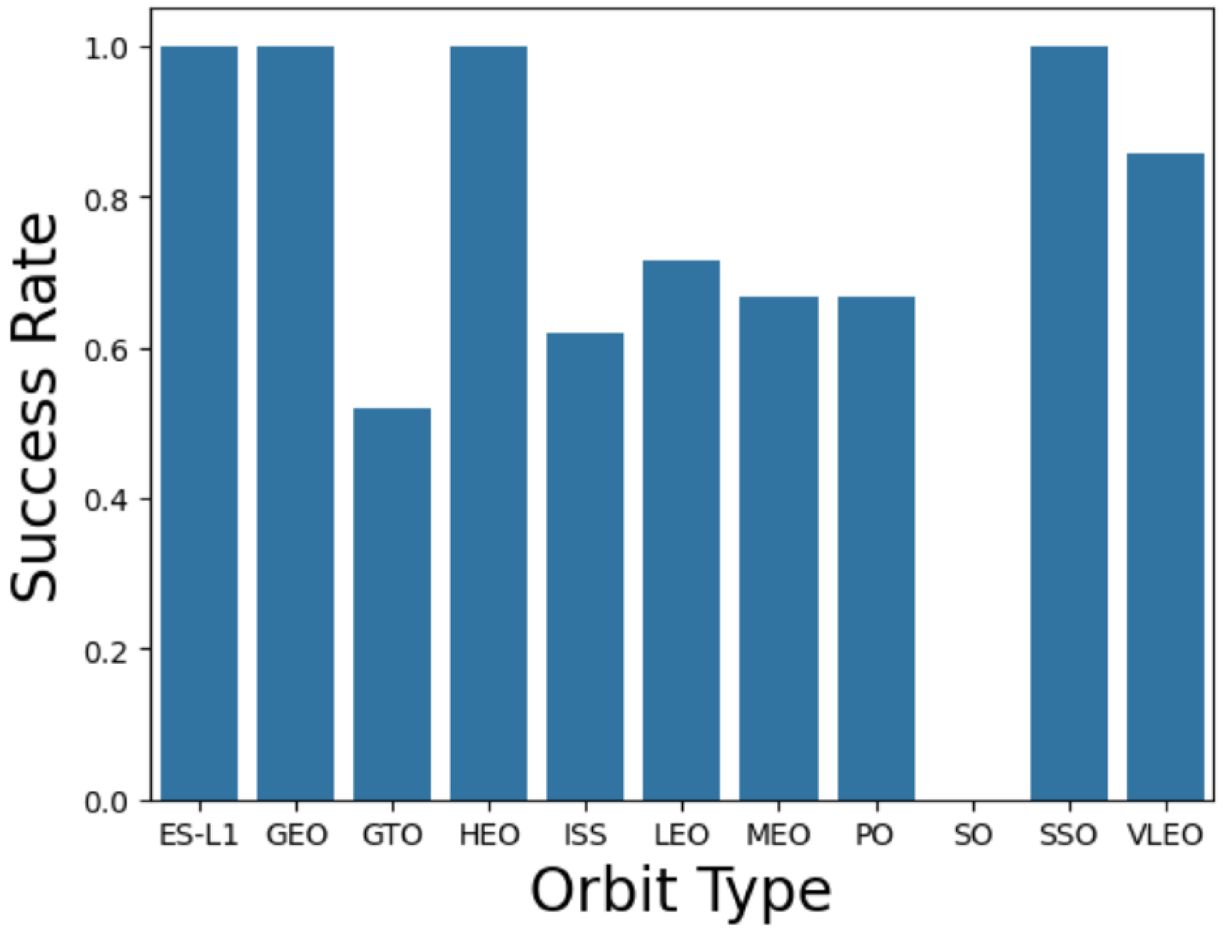


# Payload vs. Launch Site

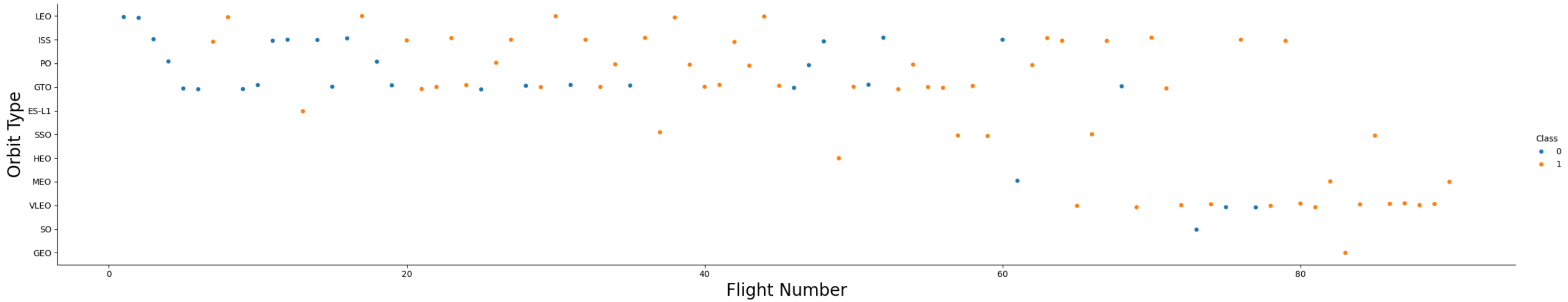
- This scatter plot shows the relationship with the payload mass and the sites. We can see the failure of launches seems to group around the 4-6k kg range. This could be an important factor as it seems to happen regardless of launch site.

# Success Rate vs. Orbit Type

- This bar chart show the types of orbits and their success rate.
- Different orbits will require different performance from the rocket, that might lead to failure from strain.
- We can see that orbits “ES-L1”, “GEO”, “HEO”, and “SSO” have a 100% success rate.

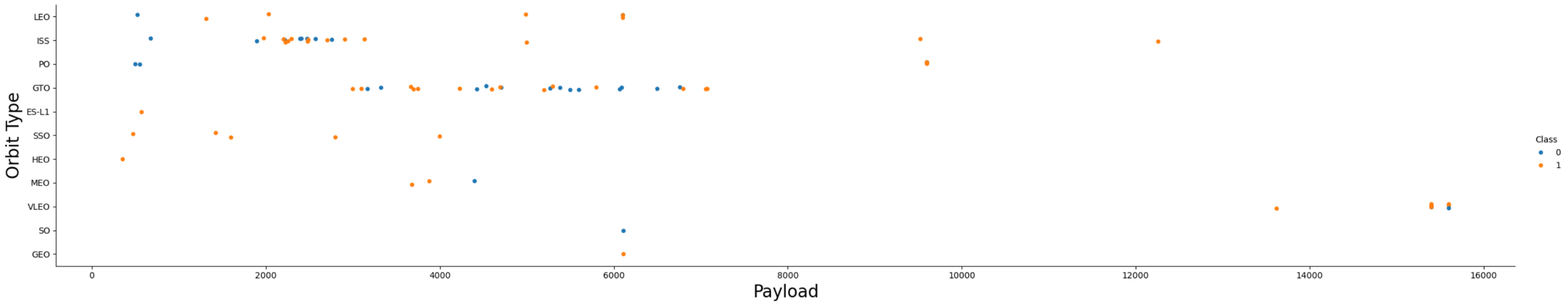


Low Earth Orbit (LEO), Very Low Earth Orbits (VLEO), Geosynchronous Orbit (GTO), Sun-synchronous Orbit (SSO / SO), Lagrange Points (ES-L1), Highly Elliptical Orbit (HEO), International Space Station Orbit (ISS), Geocentric Orbits 2,000 km to 35,786 km (MEO), Circular Geosynchronous Orbit 35,786 km (GEO), Polar Orbit (PO)



# Flight Number vs. Orbit Type

- This graph shows the success of the flight while comparing Orbit type to the flight number.
- We see that the early launches are matched with mainly four types of orbits. These orbits are what we saw with a lower success rate in the previous slide. That can help account for the low success rate seen previously.

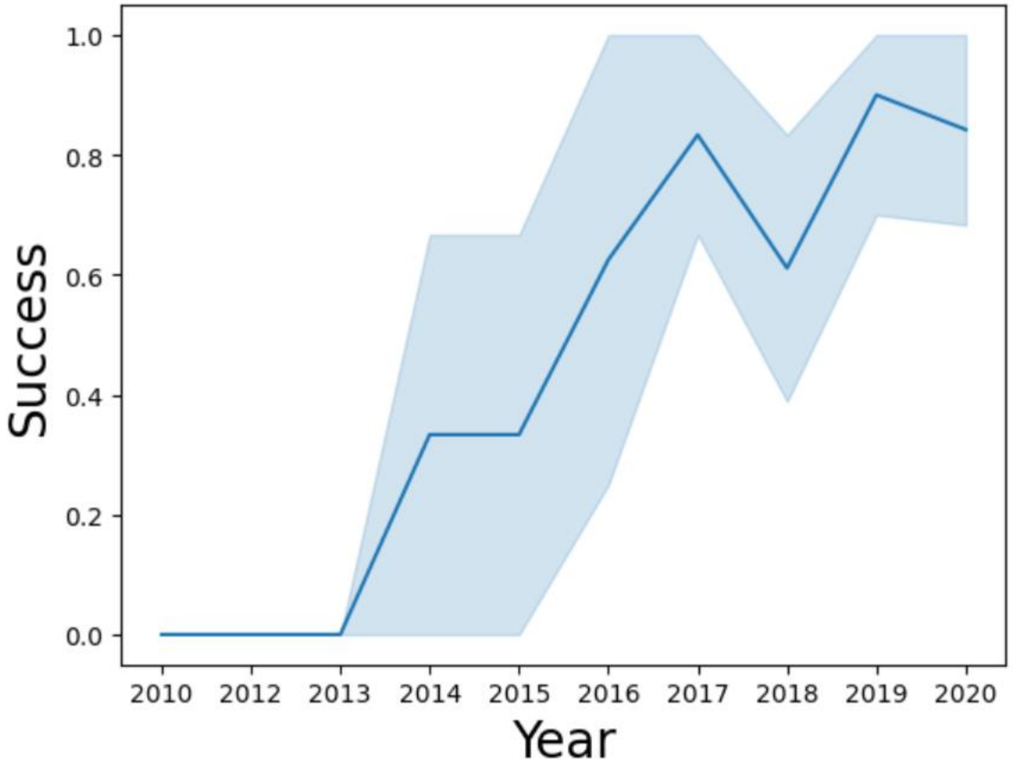


# Payload vs. Orbit Type

- This graph shows the success while comparing Payload vs Orbit type.
- We can see what looks like a cluster of failures around the 6000 kg point. This might indicate that GTO orbits with that payload could be too much for the rockets to take.

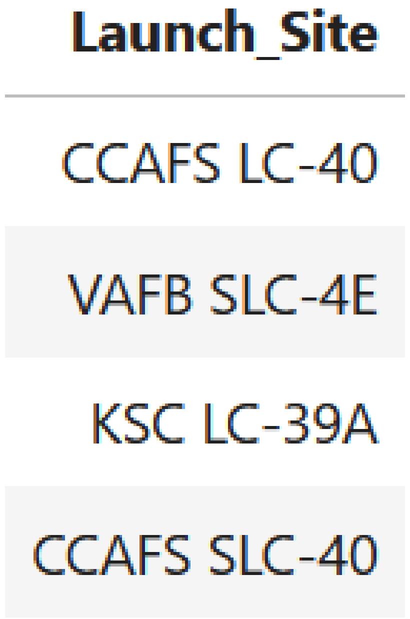
# Launch Success Yearly Trend

- This shows the trend line of the success for each year.
- We see that around 2013 the success rate starts climbing. Except for 2018, and 2020 the success rate is going up.

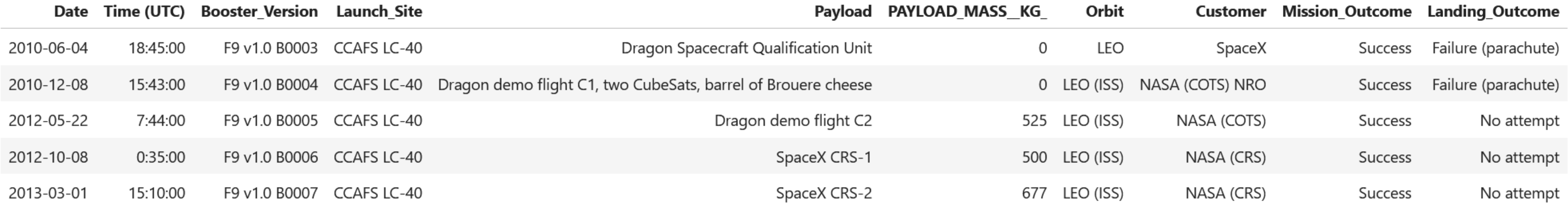


# All Launch Site Names

- We can query the names of the launch sites. By making use of the name of the launch site it allow us to group data to a physical location to give us the best comparison.



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



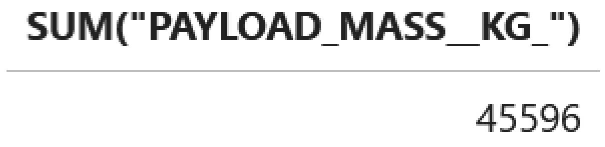
| | Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | 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 |

# Launch Site Names Begin with 'CCA'

- We can look at sites that start with the same code, that allows the comparison of launch pads that are relatively close and should have similar conditions.

# Total Payload Mass

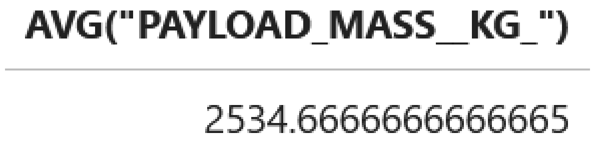
- We can sum up the payloads of launches from a specific customer
- We can see here that NASA has launched 45,596 Kg with SpaceX



| SUM("PAYLOAD_MASS__KG_") |
|---:|
| 45596 |

# Average Payload Mass by F9 v1.1

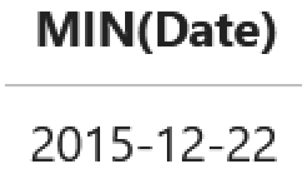
- We can query the average payload by the booster type F9 v1.1 which is 2534.6 kg.
- This can be helpful in determining if this rocket type is carrying different loads.



| AVG("PAYLOAD_MASS__KG_") |
|---|
| 2534.6666666666665 |

# First Successful Ground Landing Date

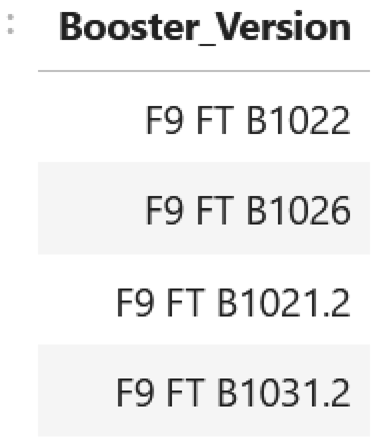
- We can query when the first landing data was. Allowing us to set landmark dates for better analysis when we see changes in success.



| MIN(Date) |
|---|
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

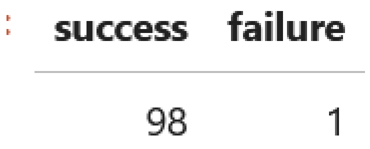
- We queried what boosters carried 4-6 k kg payloads and landed on the drone ship. This can give us info on what rockets might have changes that improved landing success.



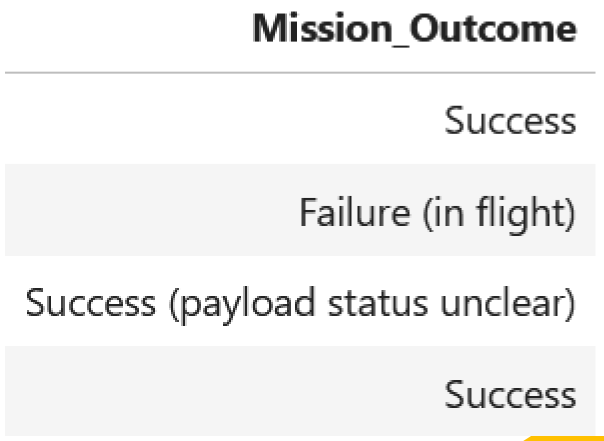
| Booster_Version |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

# Total Number of Successful and Failure Mission Outcomes

- The mission's outcomes are different then landing outcomes. If the payload reaches its intended orbit, then we have success regardless if the booster doesn’t land. We can query the success counting the different outcomes that indicate success.



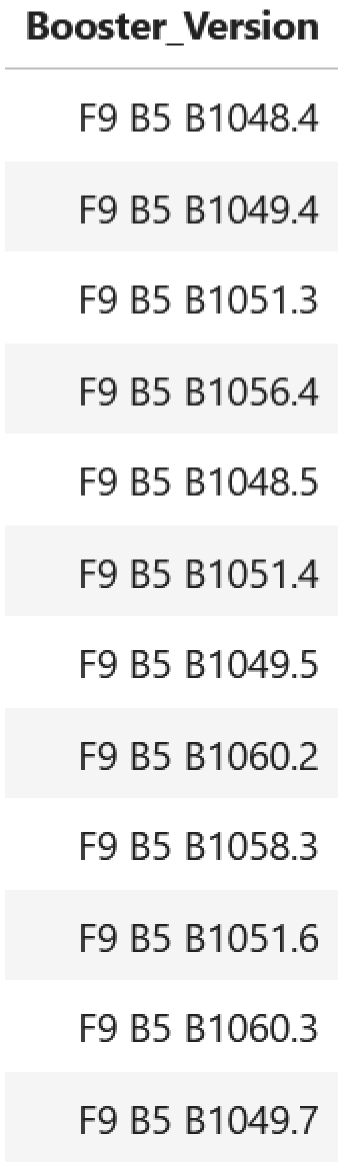
| success | failure |
|---|---|
| 98 | 1 |



| Mission_Outcome |
|---|
| Success |
| Failure (in flight) |
| Success (payload status unclear) |
| Success |

# Boosters Carried Maximum Payload

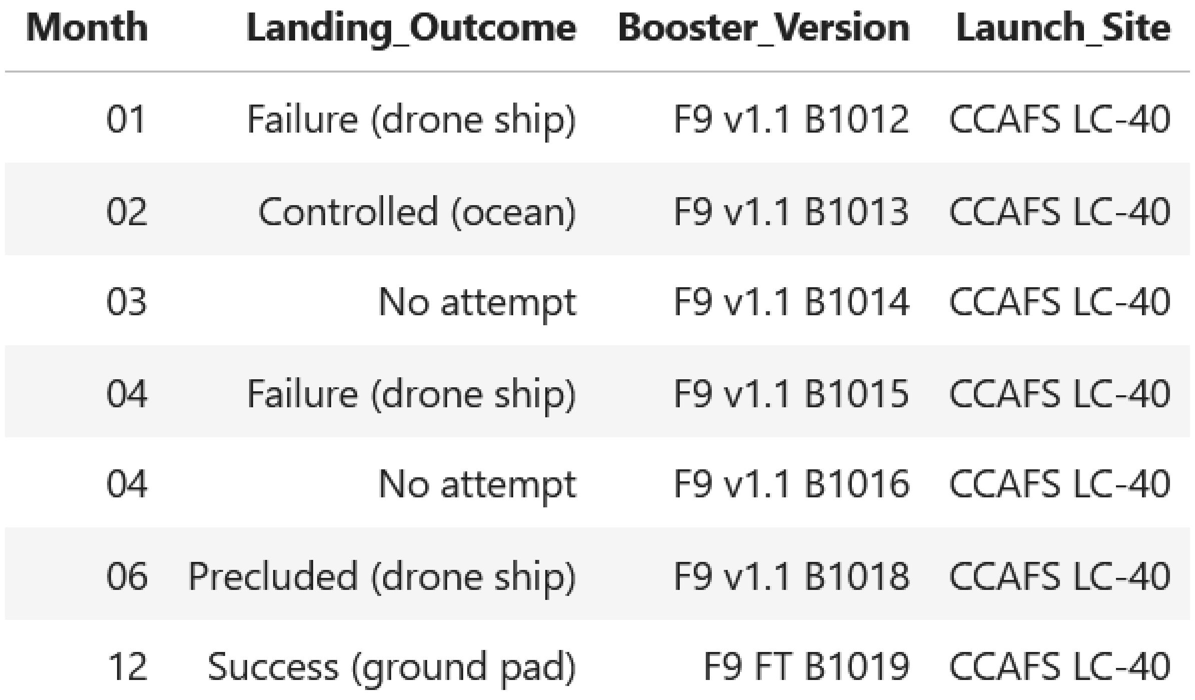
- Along with having info on boosters that carried payloads in a range, we can see what booster were maxed out. This is useful to see if newer versions are doing better or worse.



| 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

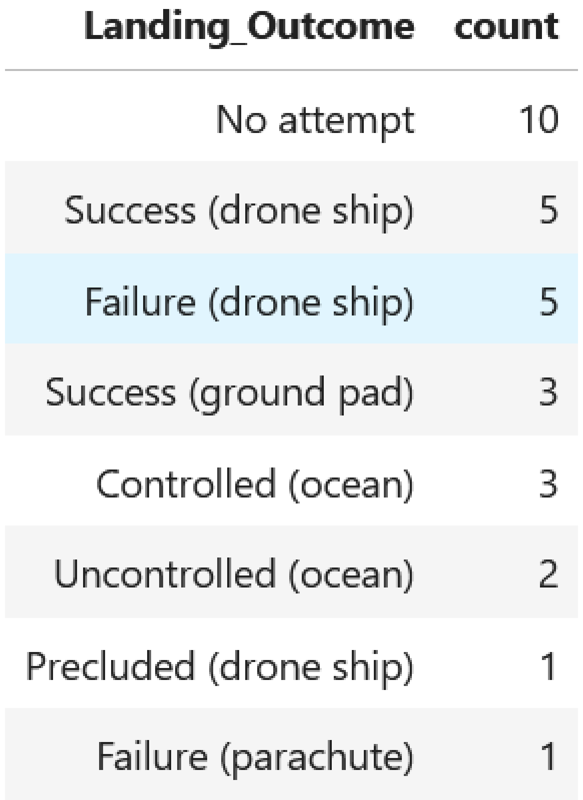
- This query allows us to focus on just one year. We want to look at the launches by month and see if there was a change through the year.



| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 02 | Controlled (ocean) | F9 v1.1 B1013 | CCAFS LC-40 |
| 03 | No attempt | F9 v1.1 B1014 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |
| 04 | No attempt | F9 v1.1 B1016 | CCAFS LC-40 |
| 06 | Precluded (drone ship) | F9 v1.1 B1018 | CCAFS LC-40 |
| 12 | Success (ground pad) | F9 FT B1019 | CCAFS LC-40 |

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

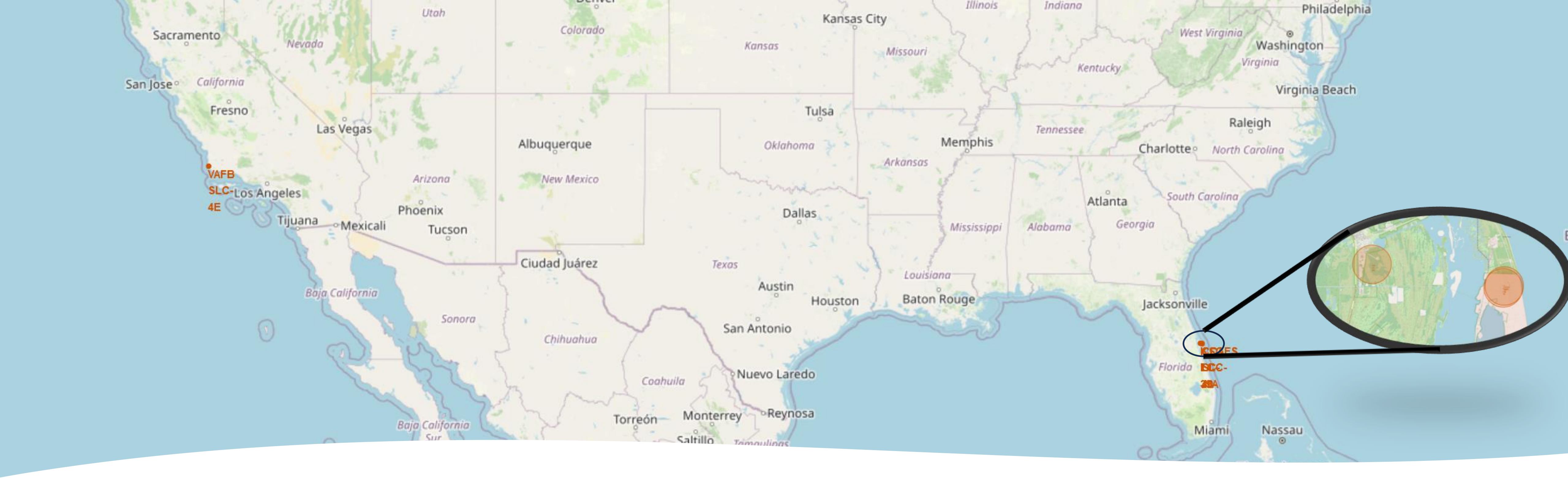
- We also will want to look at a range of launches and see how the landing rank.
- We can see in these ~7 years that there were several no attempts, but a decent number of successes.



| Landing_Outcome | count |
|---|---|
| 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 |

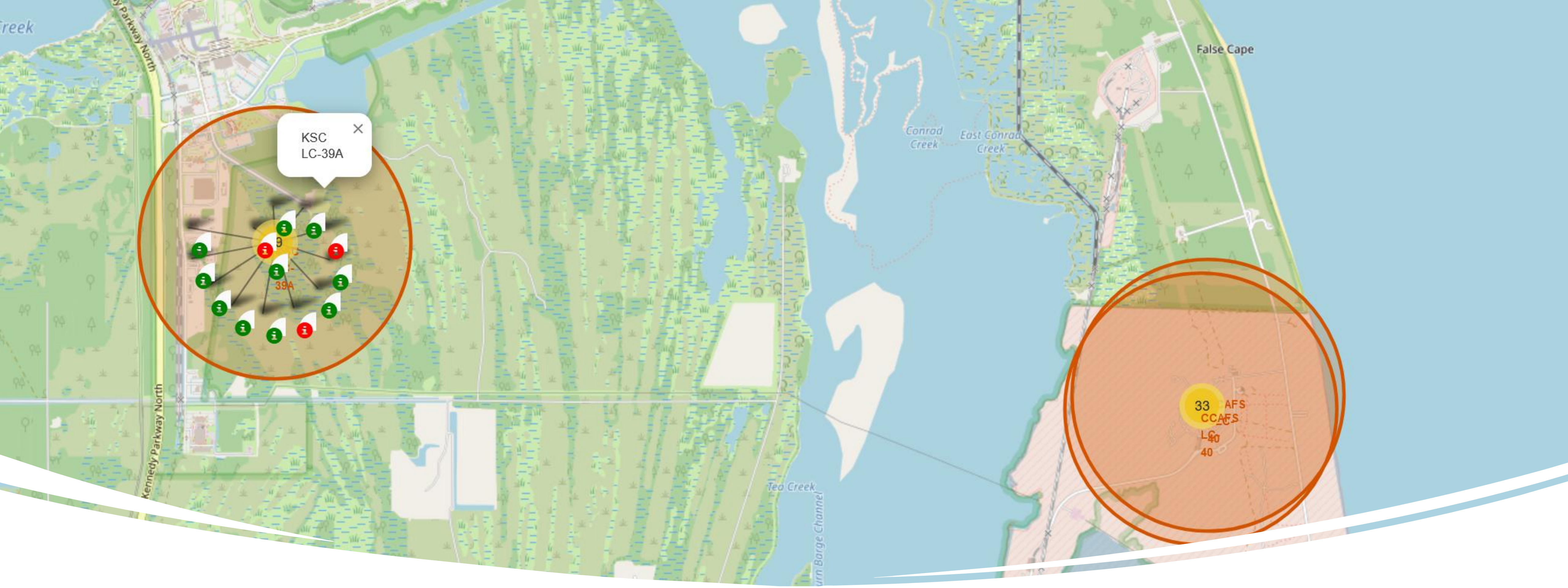
Section 3

# Launch Sites Proximities Analysis



# Locations of SpaceX Launch Sites

- ¾ of the launch sites are located within a few miles of each other on the east coast. This allows for analysis to be simplified as they have similar Weather and Environmental factors.
- All the east coast launch sites are non-military sites. This might affect launch requirements.
- The launch site at Vandenburg is the only site that allows for polar orbits. Polar orbits are different as they do not go with spin of the atmosphere and could experience different forces on the rocket.

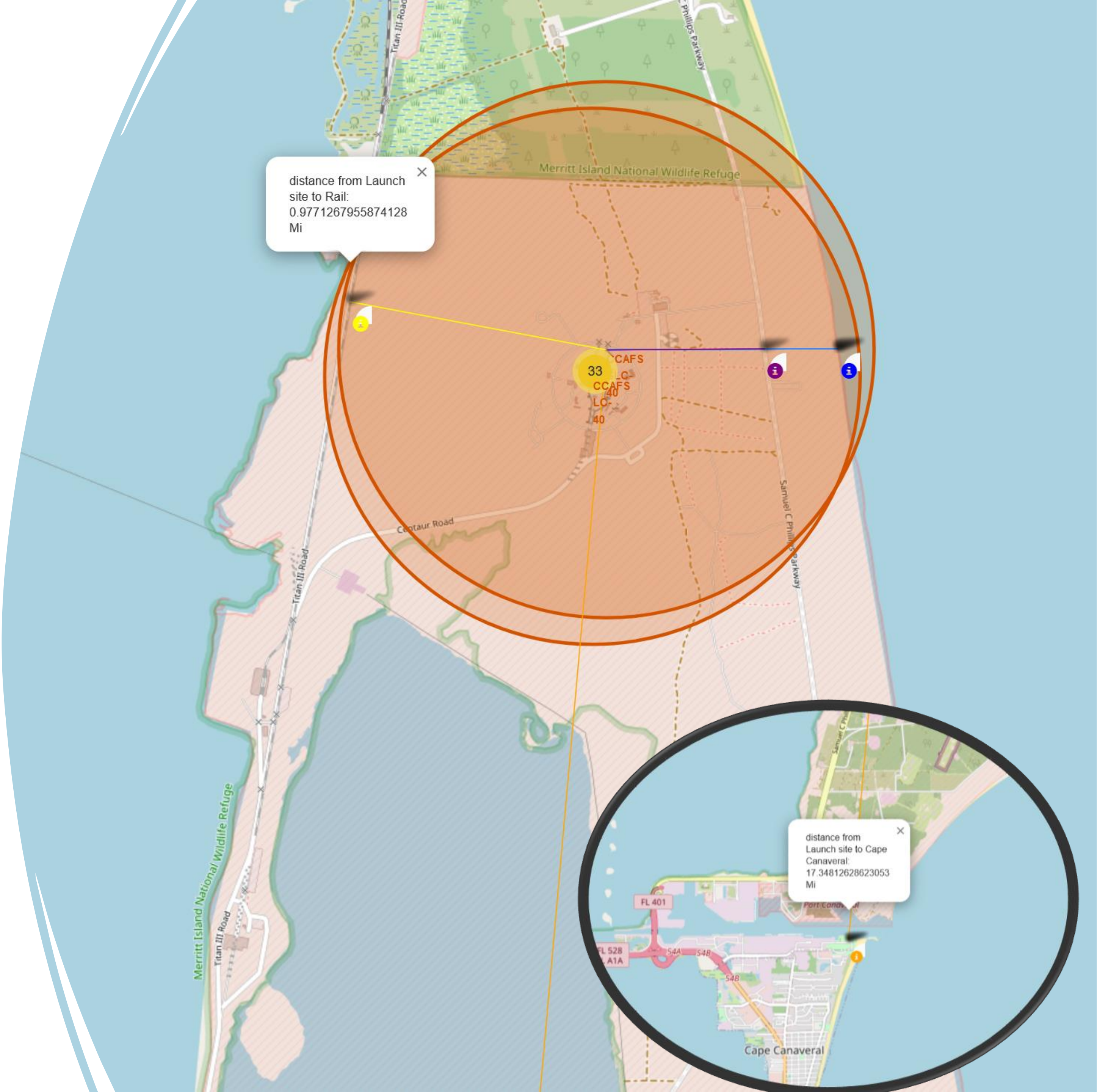


# Launch Clusters with Success indicators

- The Map is updated to show the location of launches and the success of the launch. Green indicating success and Red indicating a failure of some kind.
- To have a clean map the launched are grouped into yellow circles showing the number of launches. Once clicked it reveals all the launches in a spiral pattern.
- This allows for easier analysis at a glance to see where physical the most failures occurred.

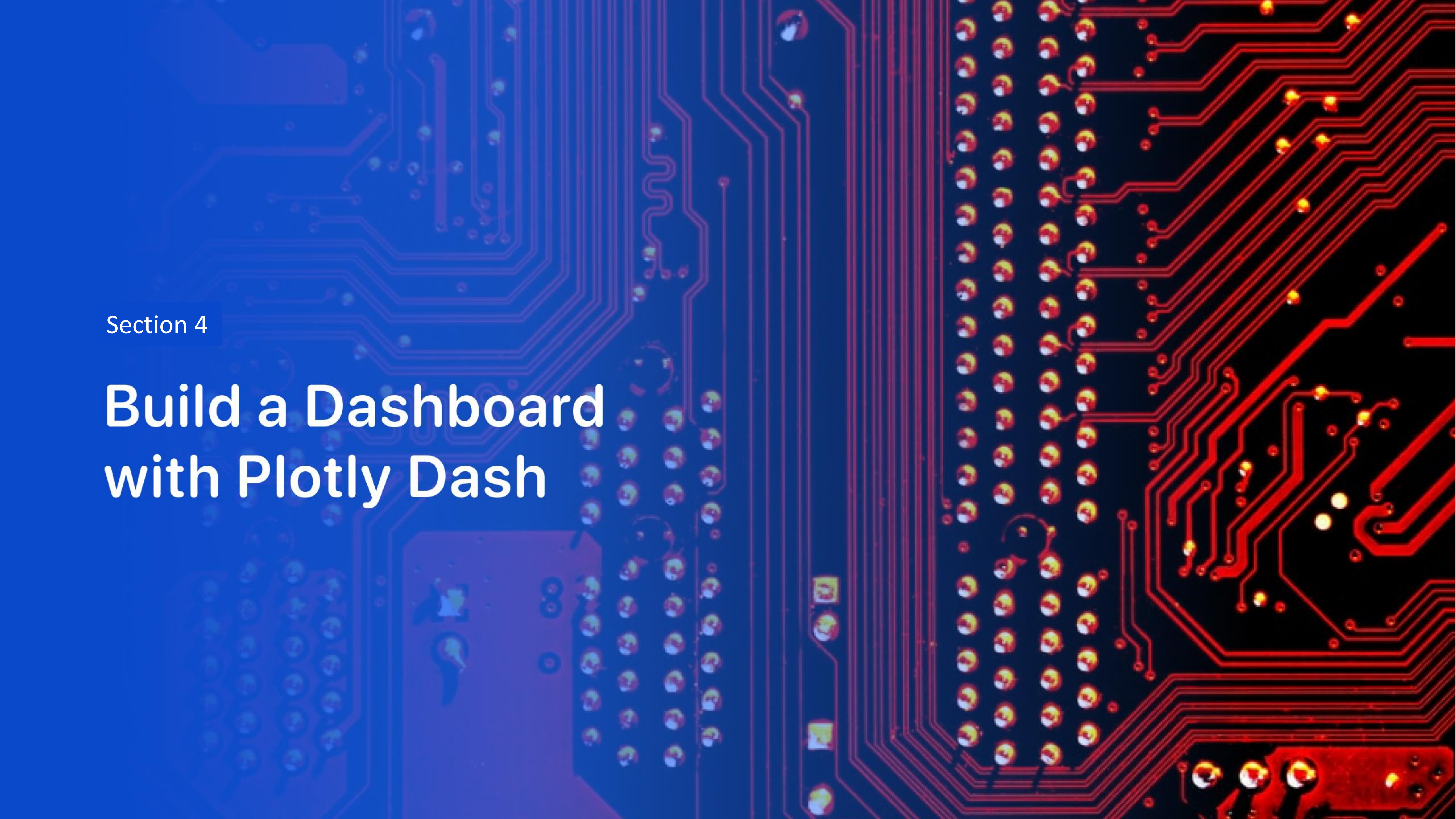
# Distances to near by infrastructure

- The location of near by buildings and infrastructure might play a role in the success of rockets. Roads or track could be damaged by rockets launches if too close. Passing traffic could endanger the rockets, that is why they are over half a mile to the closest highway or track
- The nearest city is cape Canaveral which is over 17 miles away. The safety of the people is much greater as rockets when launching produce large pressure waves and sound, And much more so when they fail!



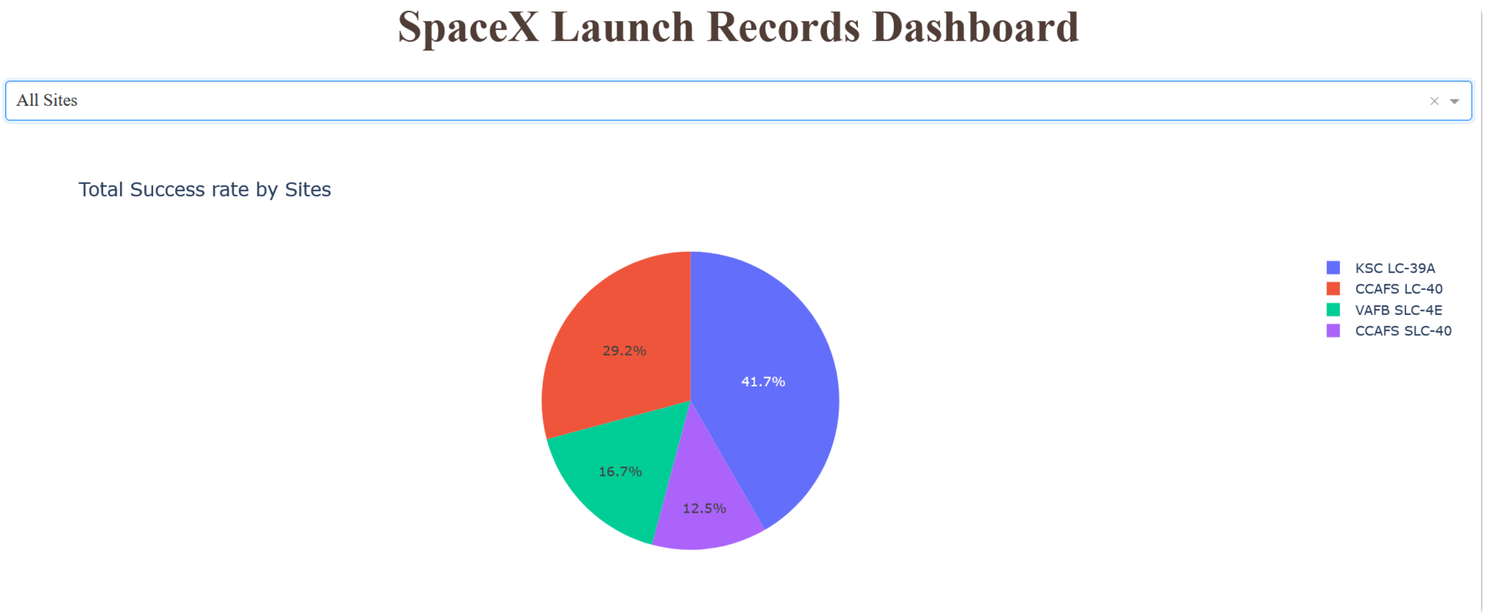
distance from Launch site to Rail: 0.9771267955874128 Mi

distance from Launch site to Cape Canaveral: 17.34812628623053 Mi



Section 4

# Build a Dashboard with Plotly Dash

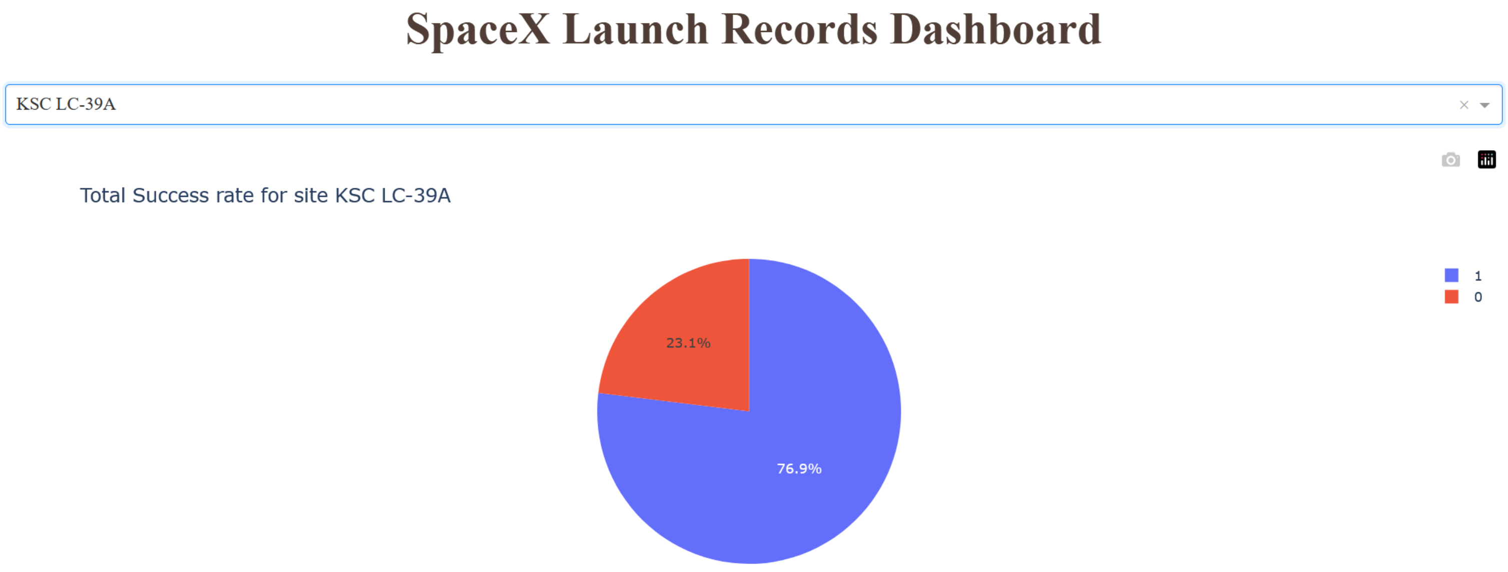


SpaceX Launch Records Dashboard

All Sites

# Dashboard – All sites pie chart

- With the dashboard we can see the distributions of successful landings by launch sites.
- This allows us to see the total number of successful landings from each site.
- This data does not show how successful the site is as it doesn’t reflect the total number of launches out of each site. Just the successes of each site. For example, CCAFS LC-40 has the most launches of any site with 26 launches but only 7 Successful landings.



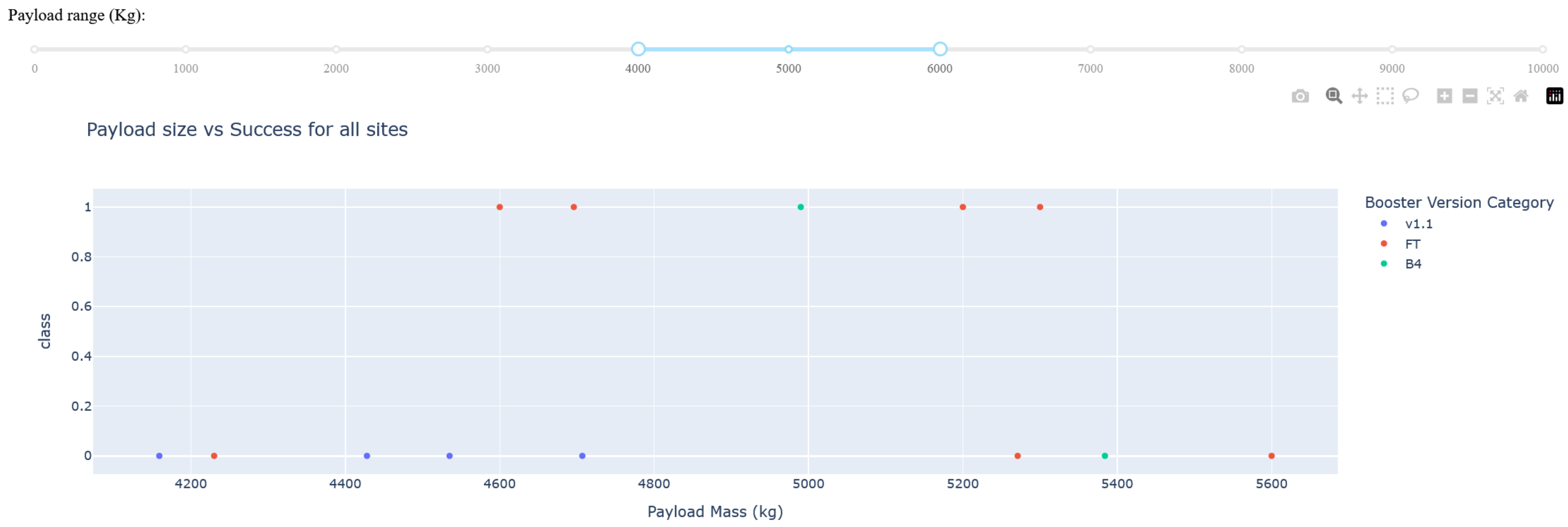
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success rate for site KSC LC-39A

# Dashboard – KSC LC-39A

- By selecting a site from the dropdown, we can see the details about how successful all launches have been. For KSC LC-39A we see that the success rate is 76.9%
- This part of the dashboard is designed to allow the inspection of each launch site. With its visualizations we can easily compare the different sites.



# Dashboard – Payload Analysis

- The Dashboard also includes a section to look at the launches by comparing the payload mass to the success while showing what booster version was being launched (show by dot color)

- With the slider above the scatter plot we can set the payload limits. By setting the limits it will only show the payloads inside that range. This graph will also show the launches out of which pad is selected at the top of the page.

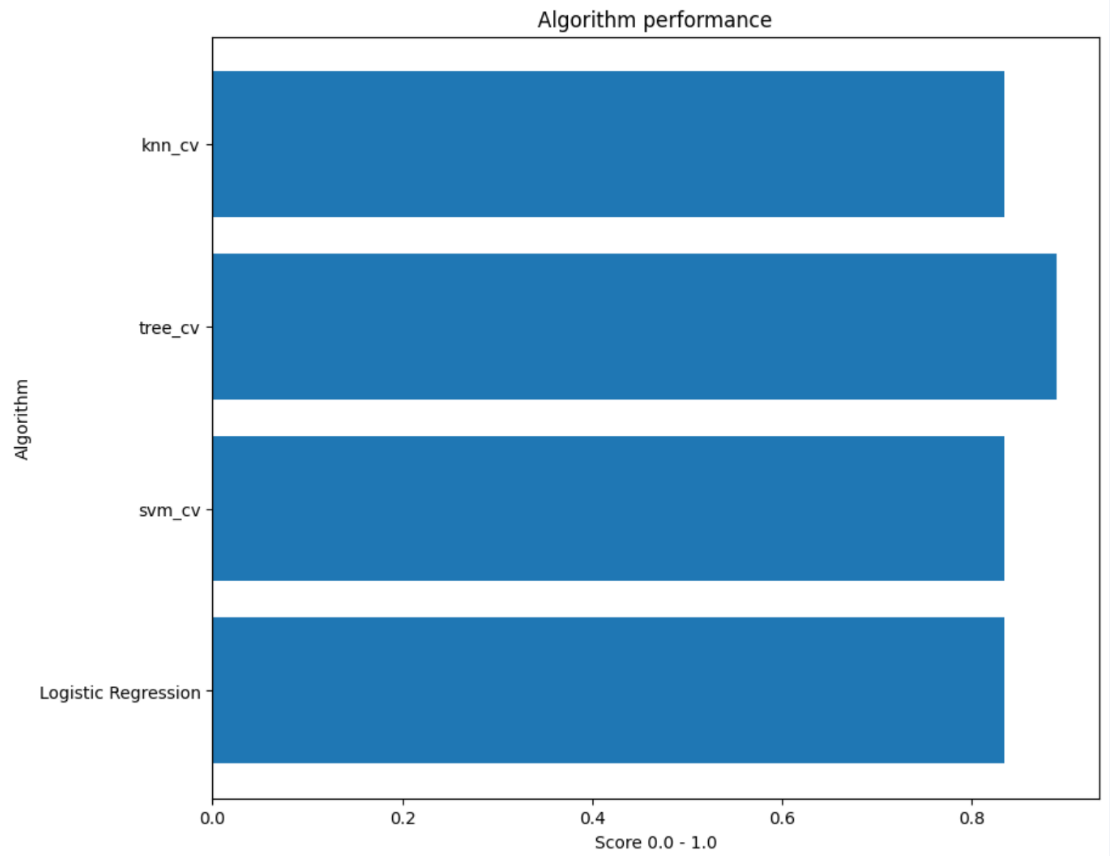
- This will provide another useful way to compare launches by providing information about payload, booster version and success in an easy to digest manner.

Section 5

# Predictive Analysis (Classification)

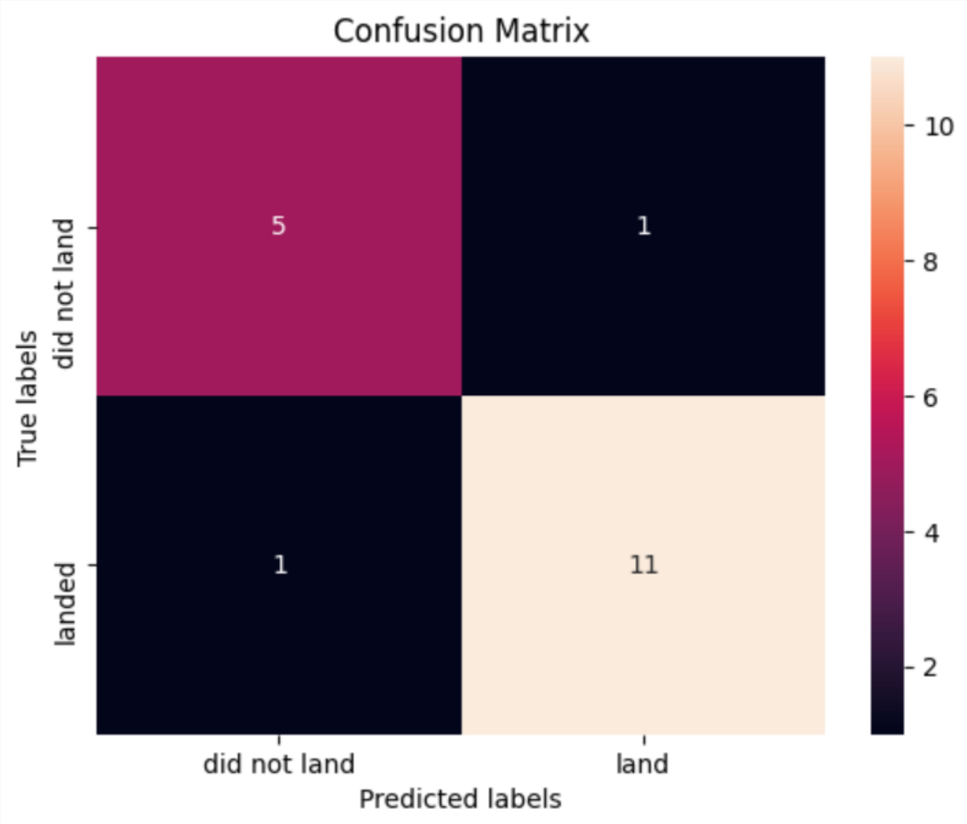
# Classification Accuracy

- The four algorithms we tried ended up being quite close in the outcomes. We can see that most have an 83.3% with Tree_cv having 88.9% success.



# Confusion Matrix

- Here we have the confusion matrix of the predicted results. There are 18 landings in this dataset, 6 failures and 12 successes. The prediction made good predictions of 5 didn't land and 11 did land. It also made bad predictions of 1 false positive and 1 false negative.



# Conclusions

- SpaceX landings are getting more reliable. With landings success sitting around 80-90%
- Orbit types influence the success rate of the landing.
- Payloads appear to influence the rate of failure in landings but appears to have non-linear relationship.
- The relationship with rocket launch parameters and the success of the landing are strong enough to get decent predictions.

# Appendix

## SQL queries

### Finding total payload from a customer

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)’
```

### Finding payload range

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND
"PAYLOAD_MASS__KG_" < 6000
```

### Finding date range

```
%sql SELECT substr(Date, 6,2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE substr(Date,0,5)='2015' ORDER
BY substr(Date, 6,2) ASC
```

Thank you!