IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Si-Jia Wu –University of Cape Town
18th April 2022

# Outline

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

# Executive Summary

The Objective of this project was to clean spaceX Falcon9 launch and landing data and use it to build a predictive model of whether the first stage of a rocket will land successfully and thus be reusable. The reusability of the first stage saves spaceX a lot of money. The following was done to achieve the final results.

- This project demonstrated two collection methods for SpaceX data: using the SpaceX API and using webscraping via the beautiful soup library.
- Performed EDA including
    - Using SQL to explore the data.
    - Using Pandas to compute summary statistics
    - Using Pandas and Matplotlib to create scatterplots, bar chart and line graphs to visualize various distributions and interactions
    - Built an interactive map with folium of with marked launch sites, successful and unsuccessful launches and proximity measures of a launch site to the nearest coast, city, rail and highway for insight into whether launch site location is intentional.
    - Built a plotly dash dashboard to see interactions between payload and boosterversions and success or failure of these landings.
- Lastly, model building was done by training and testing three methods on the spaceX data: logistic regression, SVM and classification trees. All four model methods had the same accuracy of prediction on the test set at 83%. All had the same confusion matrix and had a true positive rate of 100% and False positive rate of 50% implying that these models were good at predicting landings but not so good at predicting failed landings. Note: only 72 data points were used to train the models, there is a risk that the model is inaccurate due to insufficient training data.
- From EDA it seems that there is interaction with time, for all launch sites, landing success improved over time. And in general, success improved over time. Some launch sites have a significantly higher use than others. Payloads greater than 10000kg seem to correlate with successful landing. This interaction imply a lot variables are correlated which means that traditional logistic regression will require a variable selection technique such as lasso or ridge regression to improve performance. In general, plots show some non-linearity which means SVM, classification trees and KNN are reasonable considerations for models to handle this non-linearity in the data.
- Conclusion: **A model predisposed to predict successes if used in cost projection is likely to under account for costs. My recommendation is to train the models with more data points and perhaps tune it to reduce the false positive rate in training rather than maximize accuracy.**

# Introduction

- Project background and context

Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. SpaceX has an API regarding launch data and landing data, we can use this to answer our questions.

- Problems you want to find answers

How to predict if the first stage will be recovered? Which model is best?
Which launch site has the most successful landings?

Section 1

# Methodology

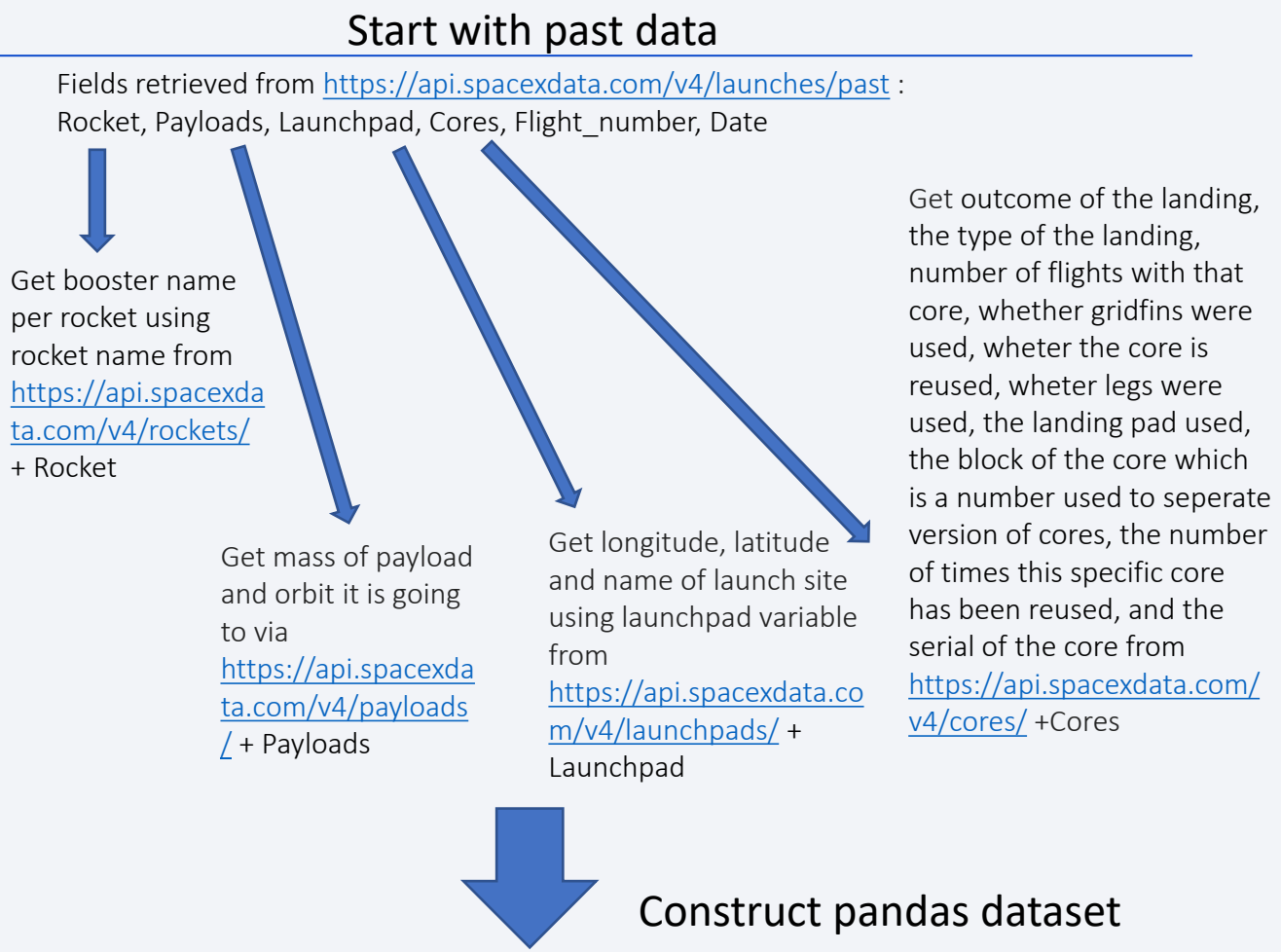
# Methodology

## Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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 – SpaceX API

- First, get the SpaceX past data. Then, use select variables of this data as input for the helper functions to get booster data, Launch site data, payload data and core data. All of which is used to construct the dataset. The data set is restricted to launches before 13/11/2020. Use the requests library to get all the data.
- https://github.com/Mashed-Pot80/TestRepo/blob/main/Data_collection_week1.ipynb



# Data Collection - Scraping

Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia using requests and beautiful soup libraries. Web scrape Falcon 9 launch records using requests library and then parse the data with BeautifulSoup to:

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

https://github.com/Mashed-Pot80/TestRepo/blob/main/Webscraping_week1.ipynb

Use requests library to get Raw falcon 9 and Falcon Launch tables from Wikipedia page: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Create Beautiful soup object to parse html records from requests response and extract all column/variable names from the HTML table header

Create dictionary object with column headers that will be turned into a pandas data frame and populate it by parsing the launch HTML tables using the find_all() function of the beautifulsoup object

Pandas dataframe with **flight no., launch site, payload, payload mass, orbit, customer, launch outcome, version booster, booster landing, date and time**

# Data Wrangling

Data wrangling and initial EDA process

Performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models

There are a number of cases that imply a failed launch attempt hence training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Functions from the pandas library and numpy were used to compute exploratory and summary techniques.

https://github.com/Mashed-Pot80/TestRepo/blob/main/Data_wrangling_week1.ipynb

Check for null values and data types (categorical or numerical) per column or variable. 40% of entries for landing pad are null.

Check number of launches from each launch site using value_counts() function shows that CCAFS SLC 40 has 55 launches, KSC LC 39A has 22 launches and VAFB SLC 4E has 13 launches.

Each launch aims to an dedicated orbit, and these are the counts of launches per orbit calculated using value_counts() function. The orbit **GTO** is targeted by the most launches at **27 launches**

| | |
|---|---|
| GTO | 27 |
| ISS | 21 |
| VLEO | 14 |
| PO | 9 |
| LEO | 7 |
| SSO | 5 |
| MEO | 3 |
| ES-L1 | 1 |
| SO | 1 |
| GEO | 1 |
| HEO | 1 |

| | |
|---|---|
| True ASDS | 41 |
| None None | 19 |
| True RTLS | 14 |
| False ASDS | 6 |
| True Ocean | 5 |
| None ASDS | 2 |
| False Ocean | 2 |
| False RTLS | 1 |

Calculate the number of launches per occurence of mission outcome using value_counts(). Any outcome prefixed by True implies a successful landing and that which contains False or None is an unsuccessful landing. This is encoded to be 1 for successful landing and 0 for unsuccessful. There **are 60 successful landings** and **30 unsuccessful landings** in this dataset

| | |
|---|---|
| 1 | 60 |
| 0 | 30 |

# EDA with Data Visualization

- Plot scatterplot of FlightNumber vs. PayloadMass and overlayed with launch outcome to see how the FlightNumber (indicating the continuous launch attempts.) and Payload variables would affect the launch outcome. Plot scatterplot of FlightNumber vs. launch site overlayed with launch outcome to see how the FlightNumber and launch site would affect the launch outcome. VAFB SLC 4E has the highest proportion of unsuccessful launches.
- Plot scatterplot of launch sites and their payload mass overlayed with launch outcome to check how payload mass and launch site interact to affect launch outcome. For example: VAFB-SLC launchsite has no rockets launched for heavy payload mass>10000kg
- Plot Bar chart of success rate per orbit type to check if orbit type influences the success rate of the launch outcome.
- Plot scatterplot of FlightNumber and Orbit type overlayed with outcome to check if success rate is due to the number of flights.
- Scatterplot of payload vs orbit overlayed with launch outcome to check if payloads affect the launch outcome at different orbits.
- Line chart of launch success rate per year to check how launch outcome has changed over time for SpaceX. The graph show a general trend of improvement in successful launches from 2013 to 2020.

https://github.com/Mashed-Pot80/TestRepo/blob/main/Eda-dataviz_week1.ipynb

# EDA with SQL

In order of the tasks in the notebook, the following sql statements were executed.

Note: I was unable to connect to IBM database via the notebook, which is why screenshots of the statements run in IBM Db2 on Cloud are provided in the github notebook.

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. 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

https://github.com/Mashed-Pot80/TestRepo/blob/main/Eda-SQL_week2.ipynb

# Build an Interactive Map with Folium

1. Added circle and marker to all launch sites on a map (there are 4 launch sites) to mark see where the launch sites are
2. Marked the success/failed launches for each site on the map using red markers for failed launches and green markers for successful launches. Marker clusters were used to simplify the visualisation.
3. Marked proximity to certain locations by distance marker and line to location from launch site. Specific proximity locations are nearest railway, nearest coastline, nearest highway and nearest city.

This was to see why the launch sites are where they are. After adding these elements, it seems that launch sites are generally near the coast and close to road transportation such as roads and railway but far from highly populated areas like cities.

Please see last loaded folium map in notebook: https://nbviewer.org/github/Mashed-Pot80/TestRepo/blob/main/Launch_site_location_week3.ipynb

Note that the link is to nbviewer which renders the map, unfortunately github cannot render folium maps.

# Build a Dashboard with Plotly Dash

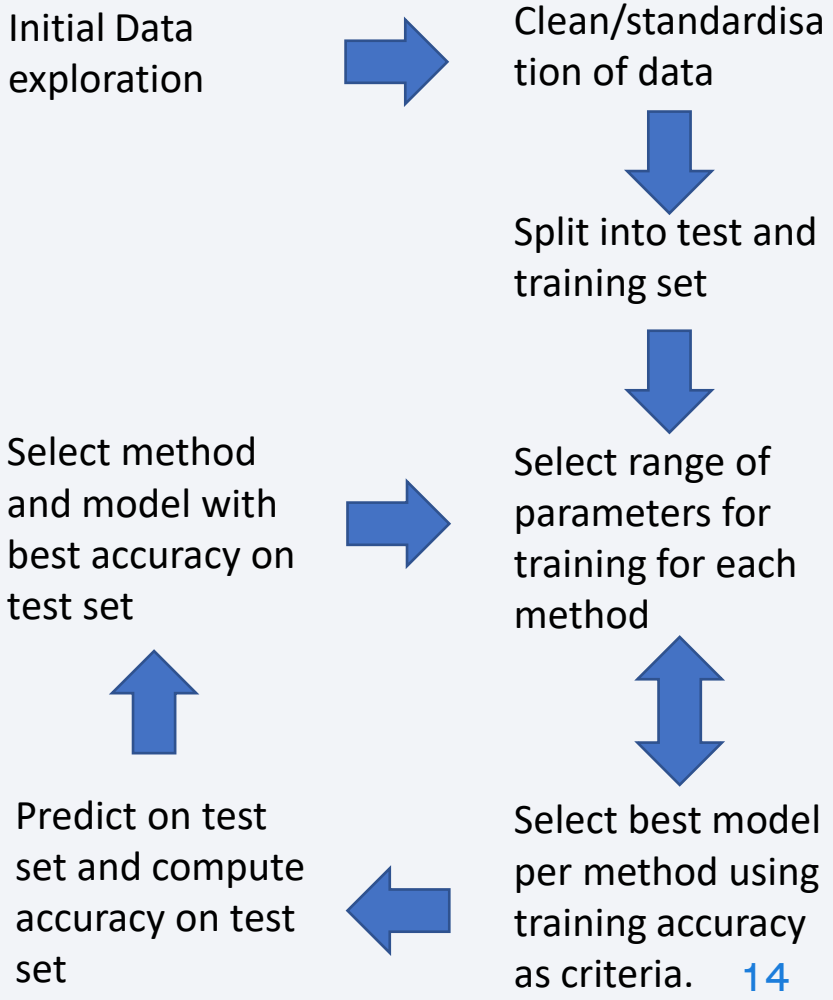
- Pie chart of % success of all sites to give an overview of which launch sites had more successful launches overall.
- On selection of specific site: Pie chart of success and failure of specific Launch site to check how successful site is.
- Slider to select payload mass. Restrict payload mass to specific range to see for certain payload mass ranges if some Launch sites perform better than others. Note. Only the scatterplot reacts to changes in the slider.
- Scatter plot with the x axis as payload and the y axis as launch outcome (i.e., class column). As such, we can visually observe how payload may be correlated with mission outcomes for selected site(s). Scatterplot is color coded by booster version on each scatter point so that we may observe mission outcomes with different boosters.

Please download and run python script to render dashboard: https://github.com/Mashed-Pot80/TestRepo/blob/main/dash_app.py

# Predictive Analysis (Classification)

Model development. Note: method implies logistic regression, SVM and or Classification

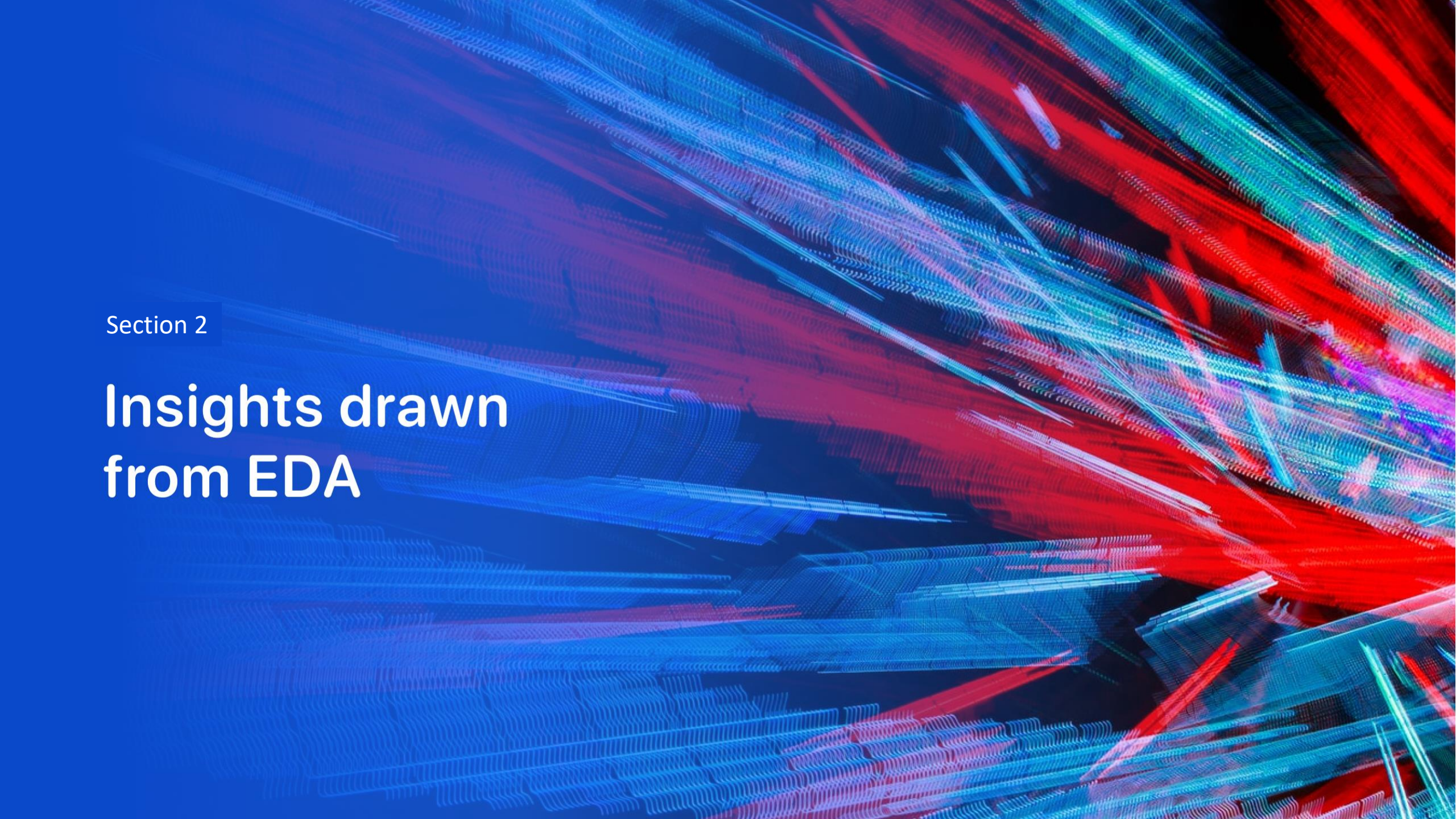
- Perform exploratory Data Analysis and determine Training Labels
  - create a column for the class so land or not land is 1 or 0
  - Standardize the data so that large differences in the ranges of the explanatory variables don't affect training
  - Split dataset into training data and test data with 80-20 split resulting in 72 training data and 18 test data
- Train and find best Hyperparameter for SVM, Classification Trees and Logistic Regression. For each method, models were built on training sets and then the best model was selected based on the best accuracy on the training set.
  - Logistic regression: tried a range of hyperparameters varying the 'C' (0.1, 0.01 or 0.001) and lasso (l1) or ridge (l2) methods with 10-fold cross validation. The best parameters turned out to be{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
  - SVM: tried a range of hyperparameters for 'C', 'G' and kernel using 10-fold cross validation using the training set. The best parameters were {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
  - Classification Trees: Tried a range of hyperparameters resulting in many tree models on the data, the selected best parameters were {'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
  - K-Nearest neighbours: built from 1 to 10 nearest neighbours and found that 9 nearest neighbours yielded best training accuracy.
- The best models from logistic regression, SVM, KNN and classification trees were used to predict on the test set to evaluate the three methods against each other. I use the accuracy of prediction on test set to evaluate the three methods.
- If there were inconsistencies, I went back and repeated the process of training and selecting models to fine tune them. Model building is iterative.
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- https://github.com/Mashed-Pot80/TestRepo/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



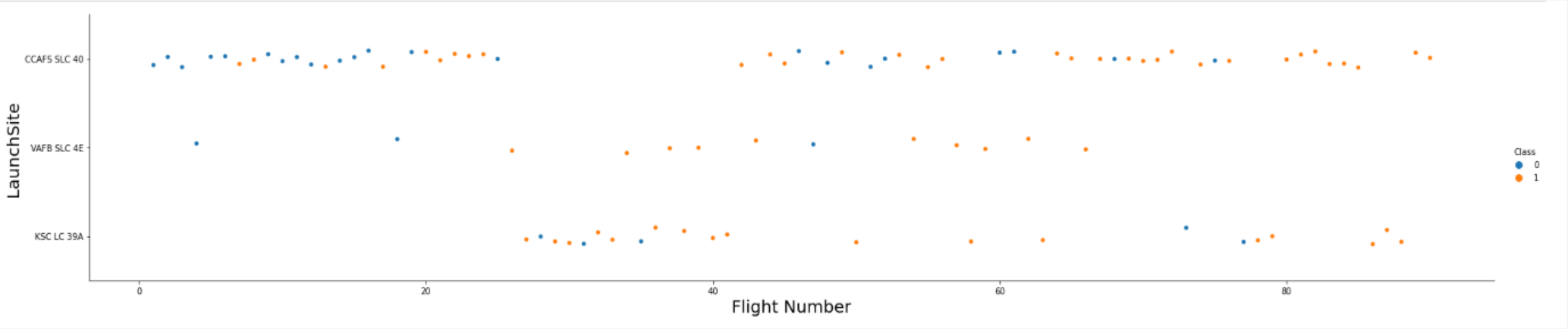
# Results

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

See following sections for insights on results

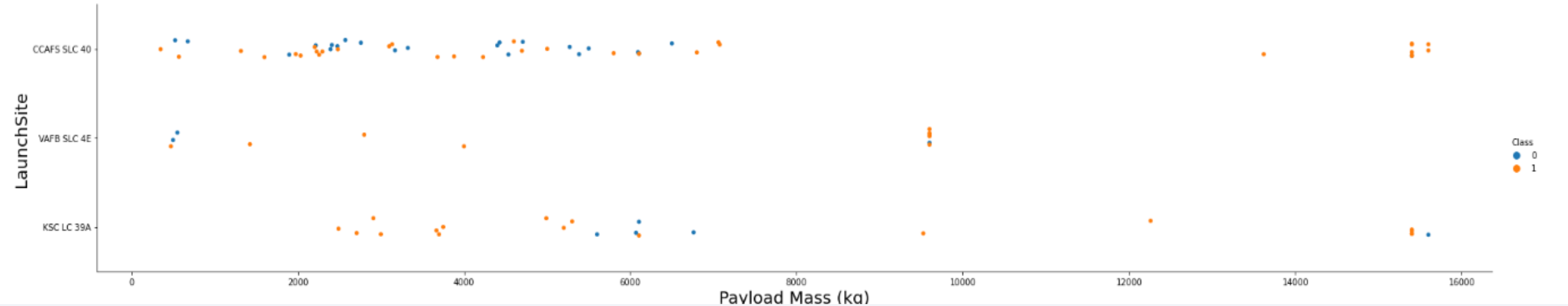

Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site




- Flight number indicates continuous launch attempts. Later flights are more successful than failures.
- We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- VAFB SLC 4E  has significantly less launches than the two others.
- CCAFS LC-40 has the lowest success rate due to being the launch site most used within the first 20 launches, the majority of which were failures.
- It seems success is likely less to do with launch site and more to do with improvement in launch, expertise and technology with successive launches

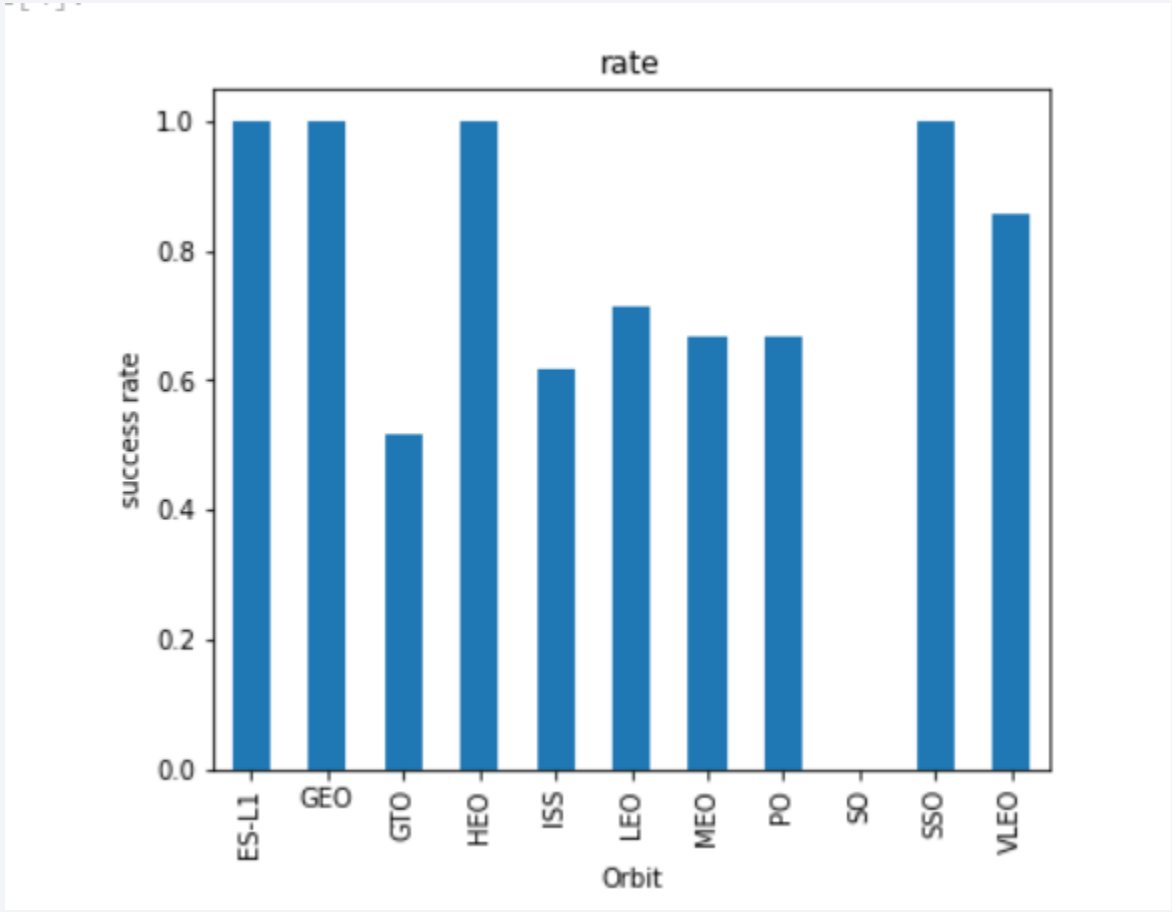
# Payload vs. Launch Site



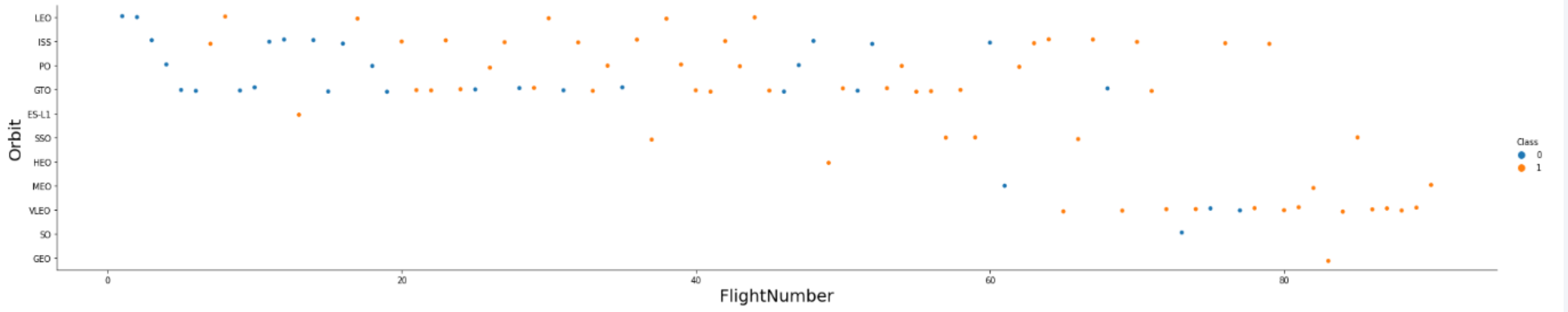

- For the VAFB-SLC launchsite there are no rockets launched for heavypayload mass (greater than 10000)
- Heavy payloads greater than 8000 seem more likely to succeed
- There is a possible non-linear pattern between the two variables

# Success Rate vs. Orbit Type

- Orbit SO has a 0% success rate.
- ES-L1, GEO, HEO and SSO have 100% success rate.
- GTO has the lowest success rate

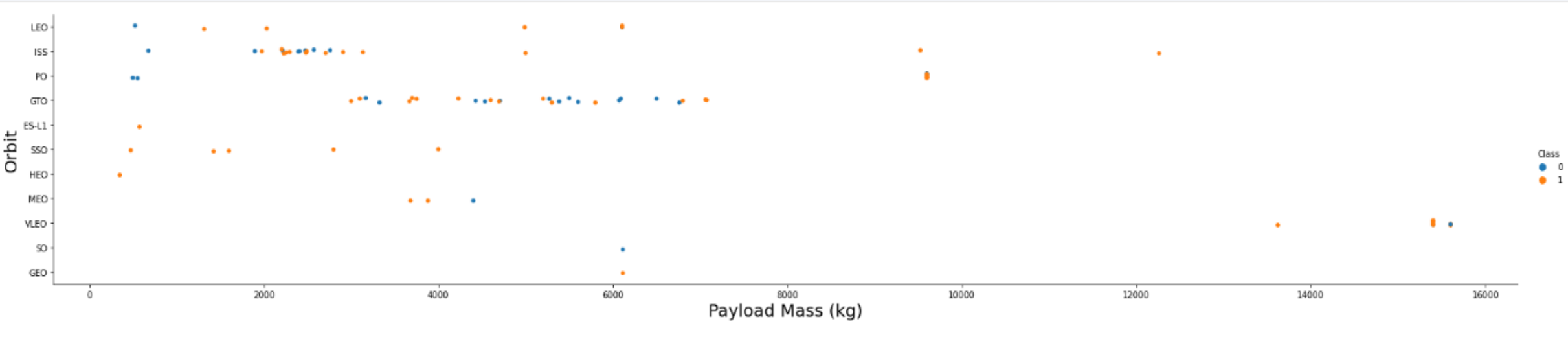


# Flight Number vs. Orbit Type




- Some orbits were only launched into later in time judging by flight number whereas GTO which has the lowest success rate has been used consistently.
- Further evidence that improvement was just due to time and not necessarily orbit.

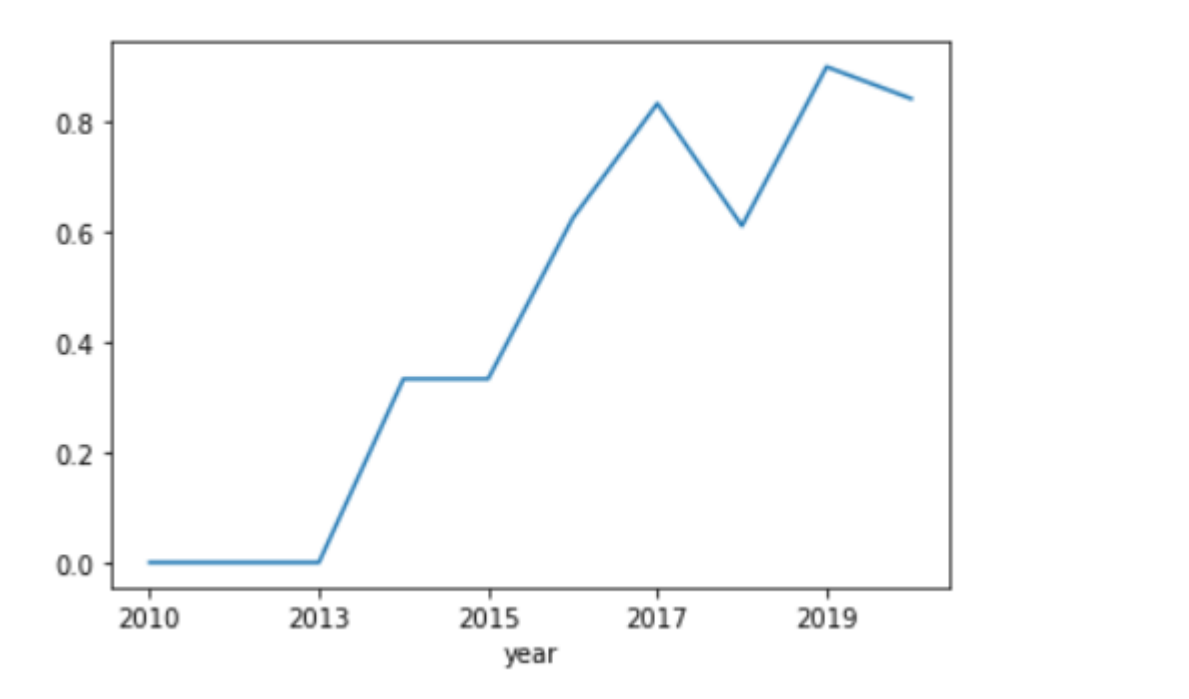
# Payload vs. Orbit Type




- With heavy payloads the successful landing or positive landing rate are higher for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.
- Payloads greater than 8000kg are only launched into ISS, IPO and VLEO orbits.
- SSO successes all have payloads of less than 4000kg.
- Possible interaction between payload and orbit.

# Launch Success Yearly Trend

- Success increases annually regardless of launch site and other factors
- Indication that early failures likely due to technology and expertise, rather than factors like payload and launch site. After all, only later launches had higher payloads.



# All Launch Site Names

Query: select unique(LAUNCH_SITE) from QYX67349.SXTBL

Result: use the unique function to get a unique set of launch sites of which there are 4.

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

# Launch Site Names Begin with 'CCA'

Query: select * from QYX67349.SXTBL where LAUNCH_SITE like 'CCA%';

Result: Notice the % sign in the where part of the statement. This is a wildcard character that implies that so long as the launch site name starts with CCA it will be selected. Limited to the first 5 records. This query is to get a general idea of how the data is and all the variables.

| DATE | TIME__UTC_ | BOOSTER_VERSION | LAUNCH_SITE |
|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 |
| 2010-12-08 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 |
| 2012-05-22 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 |
| 2012-10-08 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 |

# Total Payload Mass

Query: select sum(PAYLOAD_MASS__KG_) from QYX67349.SXTBL where CUSTOMER='NASA (CRS)';

Result: This query is used to calculate the total payload carried by boosters from NASA (CRS). Sum function is used to calculate the total. The total is 91192 kg

| |
|---|
| 91192 |

# Average Payload Mass by F9 v1.1

Query: select avg(PAYLOAD_MASS__KG_) from QYX67349.SXTBL where BOOSTER_VERSION= 'F9 v1.1’

Result: The average payload mass carried by booster version F9 v1.1 is 2928kg

# First Successful Ground Landing Date

Query: select MIN(DATE) from QYX67349.SXTBL where LANDING__OUTCOME='Success (ground pad)';

Result: Found by using the min() function on the date variable and restricting the dataset to only successful landings, 2015-12-22 is the earliest successful landing date.

# Successful Drone Ship Landing with Payload between 4000 and 6000

Query: select UNIQUE(BOOSTER_VERSION) from QYX67349.SXTBL where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

Result: List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000. Unique() function is used to get a unique list. Multiple Boolean statements are used in the where statement to find the 4 booster versions that successfully landed on a drone ship with payload masses between 4000 and 6000 kgs.

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

# Total Number of Successful and Failure Mission Outcomes

Query: select MISSION_OUTCOME, count(MISSION_OUTCOME) AS TOTAL from QYX67349.SXTBL GROUP BY MISSION_OUTCOME;

Result: The total number of successful and failure mission outcomes by using a group by statement and count() function. The result in 2 failures and 198 successes and 2 successes where the payload is unclear.

| MISSION_OUTCOME | TOTAL |
|---|---|
| Failure (in flight) | 2 |
| Success | 198 |
| Success (payload status unclear) | 2 |

# Boosters Carried Maximum Payload

Query: select unique(BOOSTER_VERSION) from QYX67349.SXTBL where PAYLOAD_MASS__KG_= (select  max(PAYLOAD_MASS__KG_) from QYX67349.SXTBL);

Result: List the names of the booster which have carried the maximum payload mass by using a sub-query. Unique function is used to get a unique list of booster versions. 12 booster versions carried the maximum payload.

| BOOSTER_VERSION |
| --- |
| F9 B5 B1048.4 |
| F9 B5 B1048.5 |
| F9 B5 B1049.4 |
| F9 B5 B1049.5 |
| F9 B5 B1049.7 |
| F9 B5 B1051.3 |
| F9 B5 B1051.4 |
| F9 B5 B1051.6 |
| F9 B5 B1056.4 |
| F9 B5 B1058.3 |
| F9 B5 B1060.2 |
| F9 B5 B1060.3 |

# 2015 Launch Records

Query: select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from QYX67349.SXTBL where LANDING__OUTCOME='Failure (drone ship)' and DATE between '2015-01-01' and '2015-12-31';

Result: The failed landing_outcomes in drone ship, their booster versions, and launch site names in year 2015, there are 4 failures and their entries are shown below.

| 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 |
| Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

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

Query: select LANDING__OUTCOME, count(LANDING__OUTCOME) as COUNT from QYX67349.SXTBL where DATE between '2010-06-04' and '2017-03-20' group by LANDING__OUTCOME order by COUNT desc;

Result: Rank of 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. The highest is no attempt landing outcome of which there were 20.

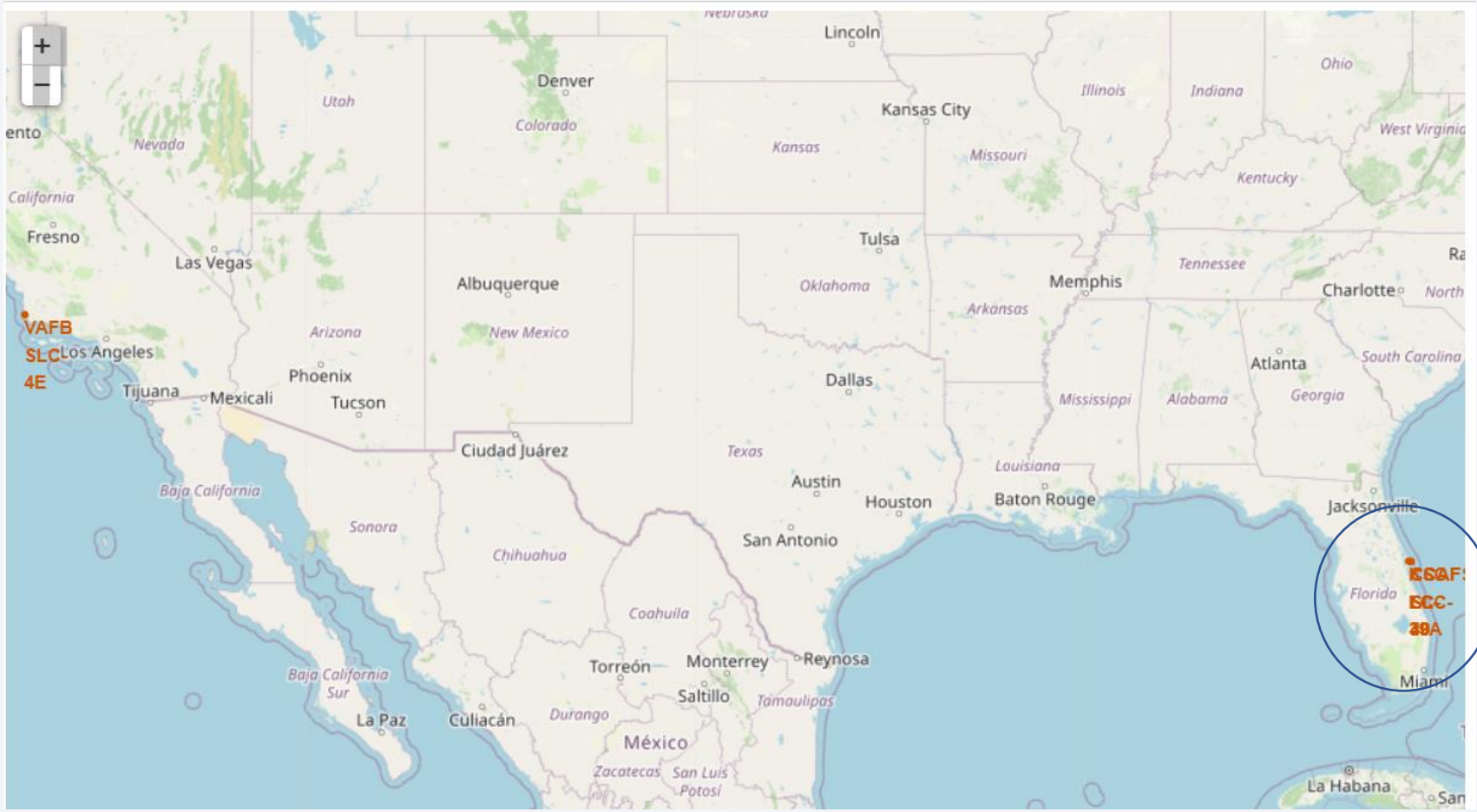
| LANDING__OUTCOME | COUNT |
|---|---|
| No attempt | 20 |
| Failure (drone ship) | 10 |
| Success (drone ship) | 10 |
| Controlled (ocean) | 6 |
| Success (ground pad) | 6 |
| Failure (parachute) | 4 |
| Uncontrolled (ocean) | 4 |
| Precluded (drone ship) | 2 |

Section 3

# Launch Sites Proximities Analysis

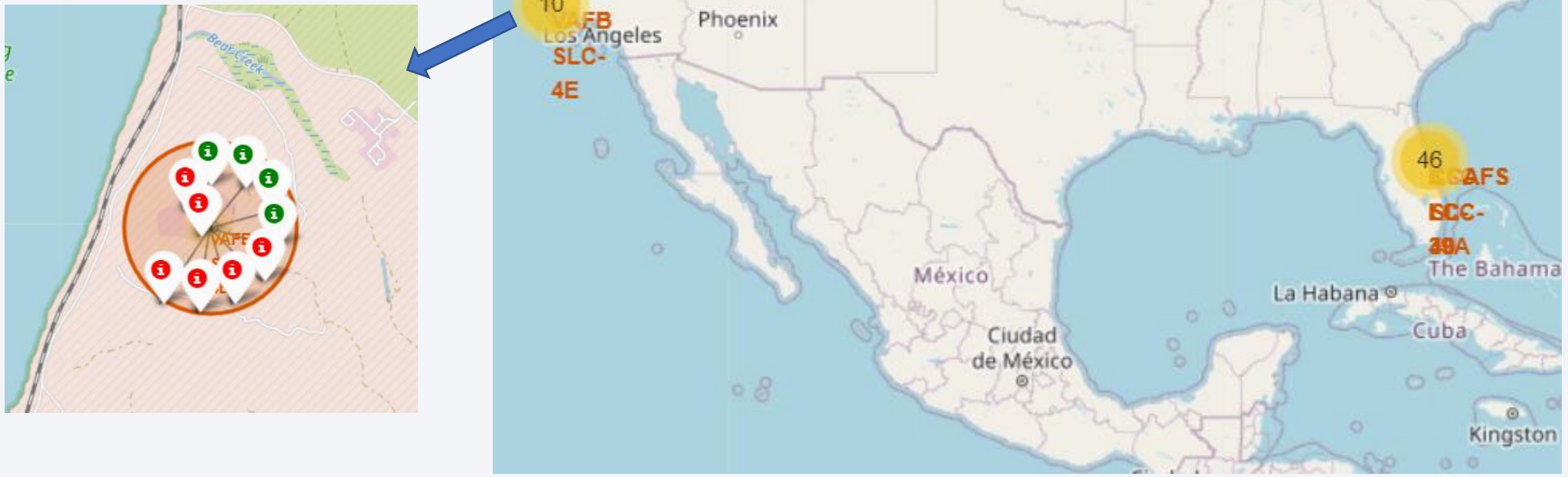
# Launch Site Locations

There are 3 launch sites on the east coast in Florida state (CCAFS LC-40, CCAFS SLC-40 and KSC LC-39A) 1 launch site in California state on the west coast (VAFB SLC 4E). The zoomed screenshot shows the 3 clustered launch sites on the east coast.




# Launch outcomes at different locations

The addition of successful (green) and failed (red) launch outcomes been added such that when unzoomed CCAFS LC-40, CCAFS SLC-40, KSC LC-39A show a combined 46 launches and on zoom show specific launch outcome. Zooming into VAFB SLC 4E which had 10 launches show that 6 launches had failure outcomes and 4 had successful outcomes judging by the markers.

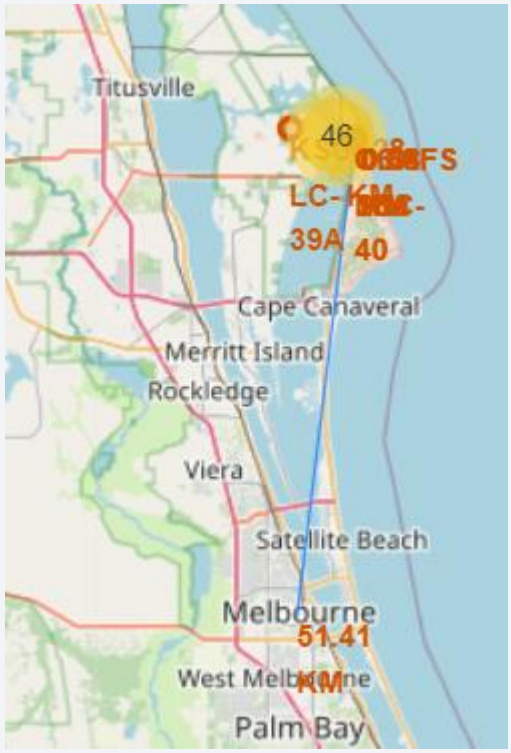


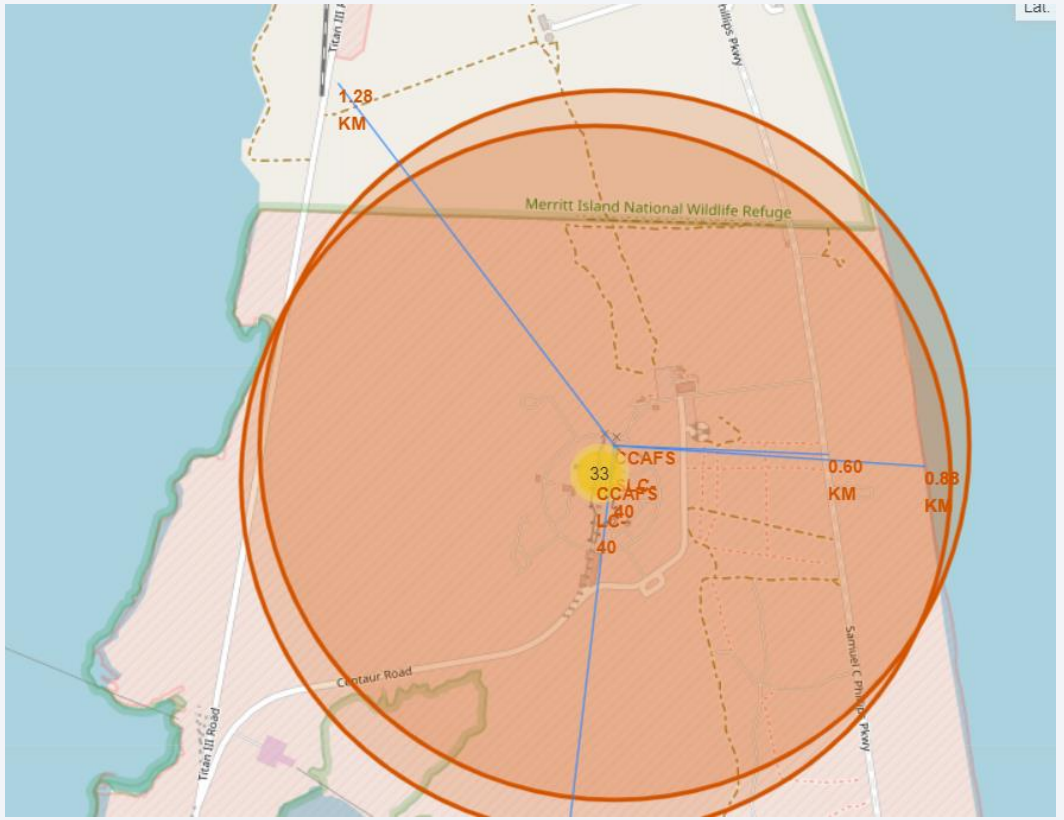
# Distance to proximities of launch site

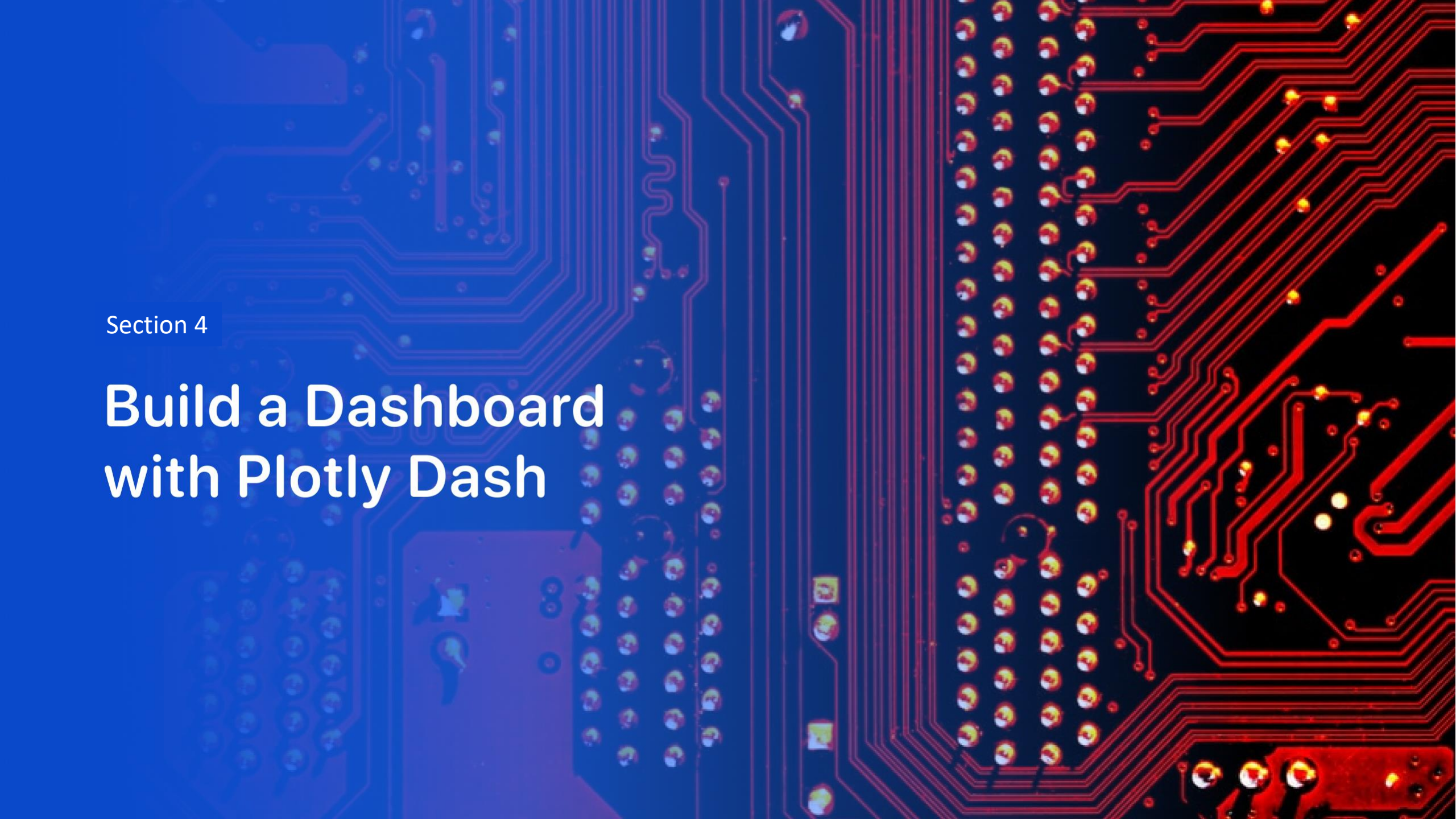
Exploring launch site CCAFS SLC-40 further shows that it is:

- 51.34 km from the nearest City, Melbourne.
- 1.28km from the nearest railroad.
- 0.6km from the nearest highway
- 0.88km from the nearest coastline

This shows that launch site is situated close to transport structures for easy access. As well as close to coastlines for ease of retrieving sea landings. However, it is far from highly populated areas like Cities, to minimize civilian access, noise complaints at launches and casualties in case of a catastrophic accident at the site.

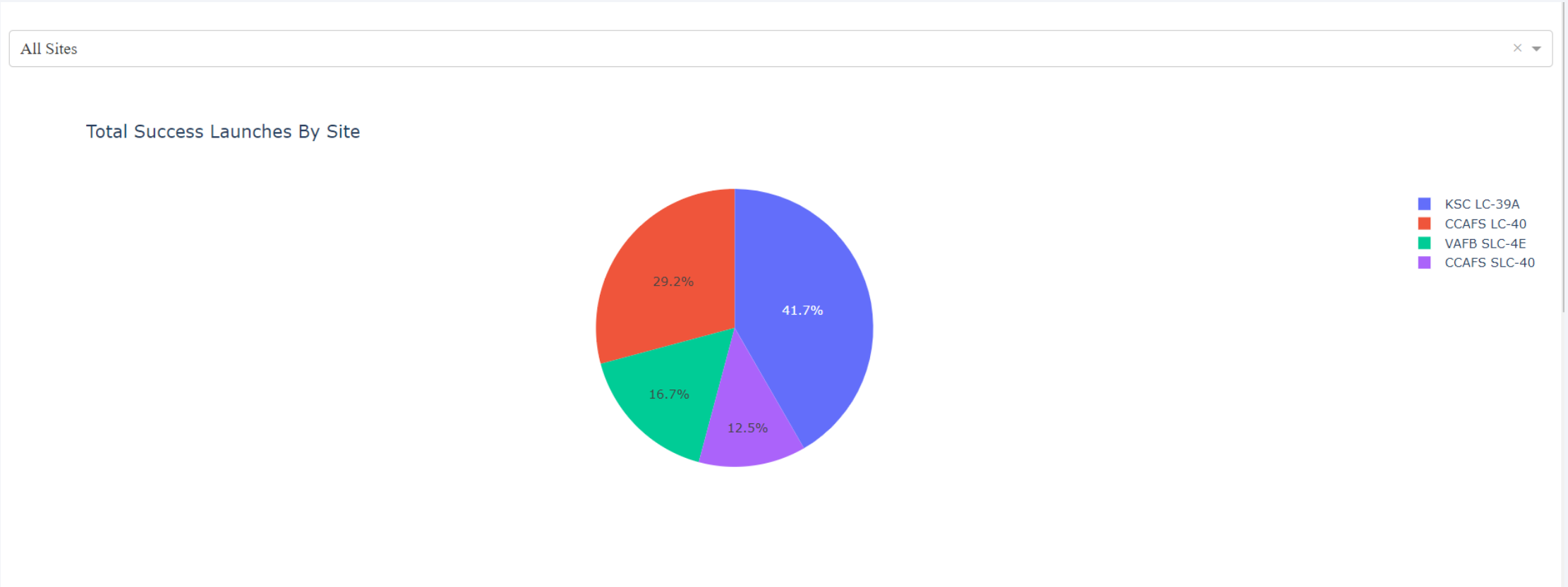



Section 4
Build a Dashboard with Plotly Dash

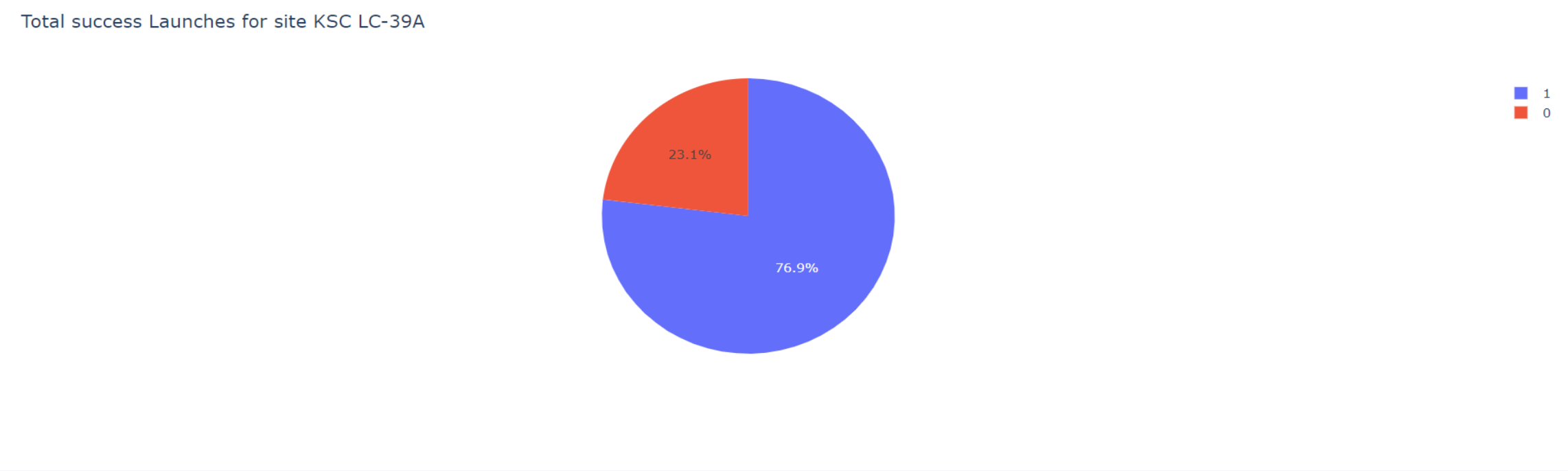
# Total launch success per site

KSC LC-39A had the highest launch success rate out of the four launch sites accounting for 41.7% of all successful launches in the dataset. CCAFS LC-40 has the smallest proportion of overall successful launches accounting for only 12.5% of successes.



# Launches for site with most success

This shows that of all 13 launches at site KSC LC-39A, 76.9% (10 launches) were successful while 23.1% (3 launches) of launches at this site were unsuccessful.

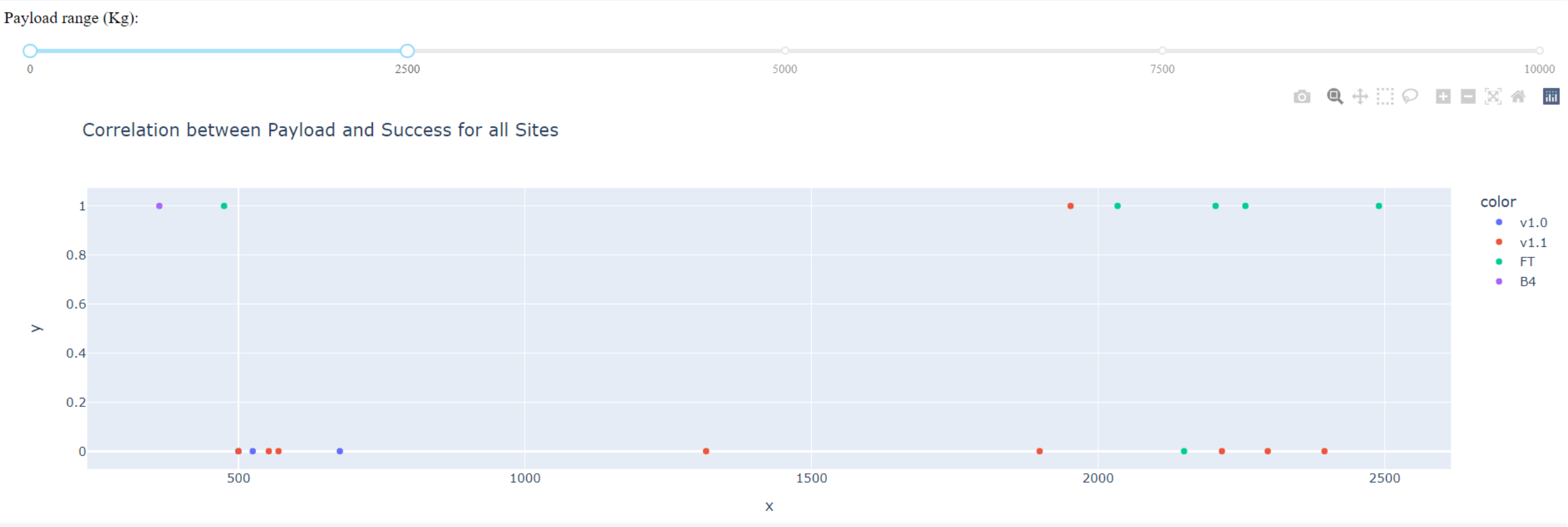


# Interaction between payload, launch outcome and booster version (1)

It seems FT booster has the most successes however it only carries payloads less than 7000kg. Similarly, v1.1 had the most failures even though it carried loads less than 5000kg. B4 has carried the highest payloads and is the only booster to carry payloads higher than 7000kg however it is uncertain whether it's performance at higher payloads is random as it has one failure and one success at the highest payload.

# Interaction between payload, launch outcome and booster version (2)

Considering payloads less than 2500kg: we find that v1.1 still has the most failures in this range. While FT has the highest number of successes in this range. Interestingly, B5 does not carry any payloads below 2500kg.

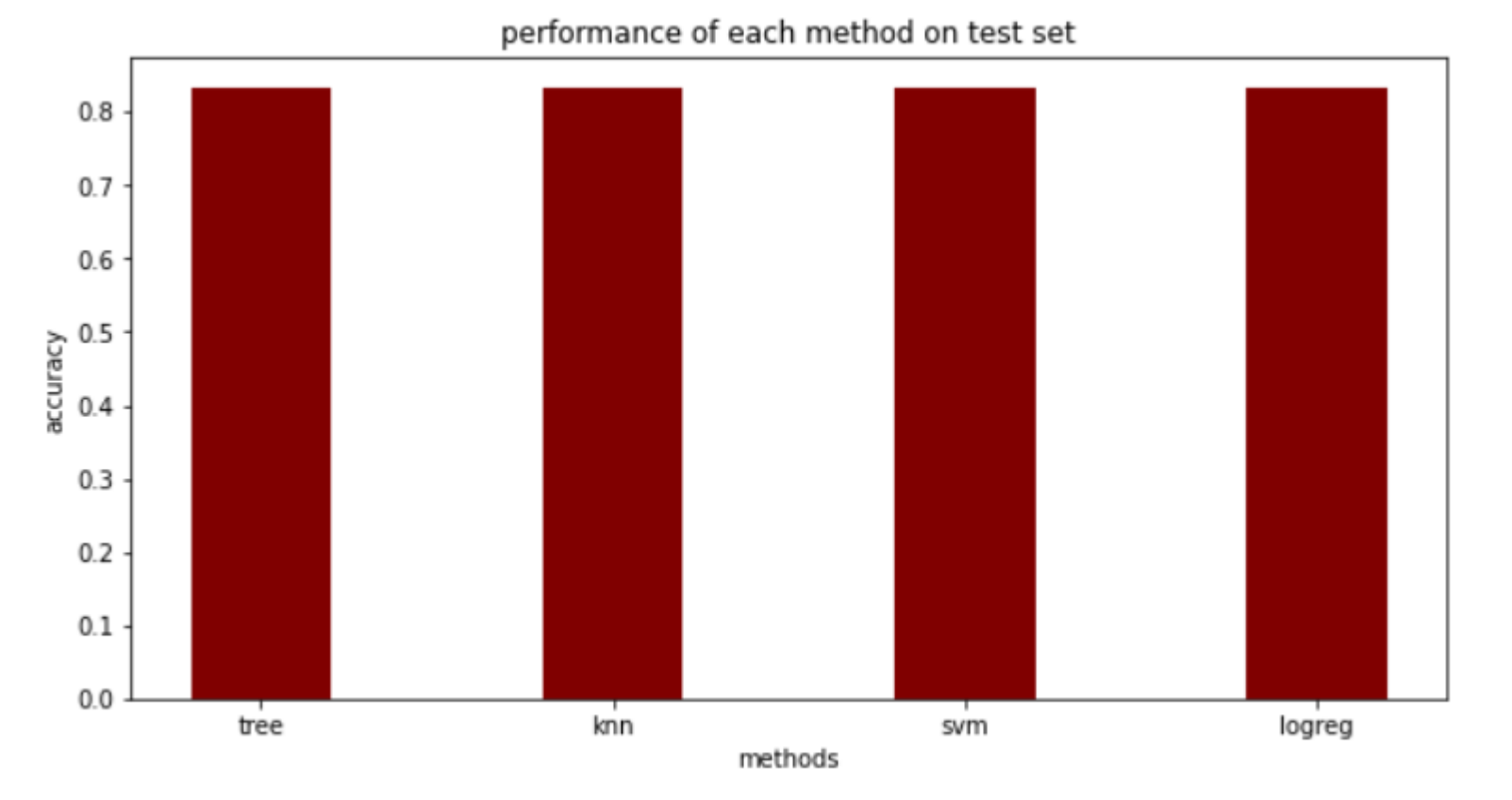


Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

All 4 methods performed the same on the test set with 83.33% accuracy on prediction. No method was better than the other, this is possibly because there is only 18 test data points which may be too small to predict on for any method to differentiate itself in terms of performance.



# Confusion Matrix

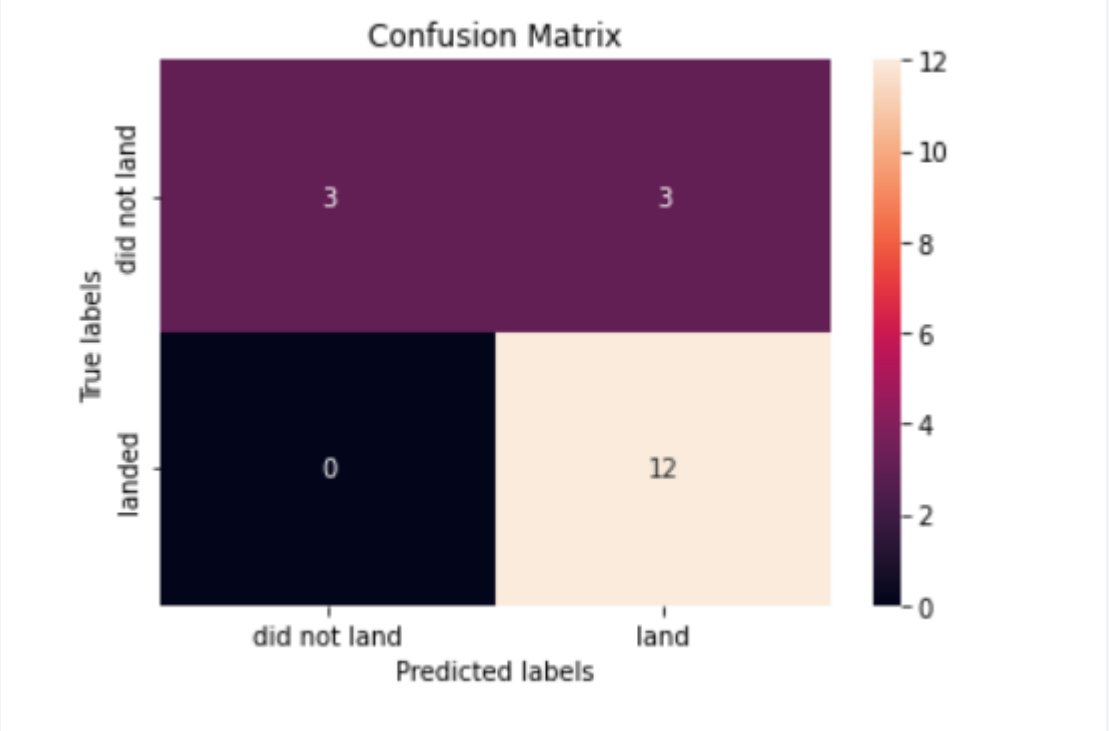
## All 4 methods yielded the same confusion matrix.

A false positive result would incorrectly classify a did not land as a landing while a false negative the model would classify a landing as non-landing.

Of the 18 test data points, there was 0% false negative rate as no landed outcomes were classified as landed. This also means a 100% true positive rate.

There were however 3 false positives meaning a false positive rate of 50%.

The model is better at classifying landings than non-landings.



# Conclusions

- Launch success improved over time which indicated that it was rather a matter of improvement in technology and expertise rather than launch site that was a good predictor in landing.
- The reason some launch sites had better landing success rates than others was because some had been in operation much longer and had more experience of early launches which had mostly failed landings.
- In general, launch sites are located near coasts and have easy access to transportation while being far from population dense areas like cities.
- All 4 modelling methods (KNN, Logistic regression, SVM and classification trees) performed the same on the test set with an accuracy of 83%. However, all had a high false positive rate at 50%. This means it is predisposed to predict successes.
- A model predisposed to predict successes if used in cost projection is likely to under account for costs. My recommendation is to train the models with more data points and perhaps tune it to reduce the false positive rate in training rather than maximize accuracy.

# Thank you!