IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Sean Moch
10 June 2024

# Outline

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

# Executive Summary

- Summary of methodologies
- Summary of all results

# Introduction

- SpaceX advertises 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.
- In this project we will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

# Methodology

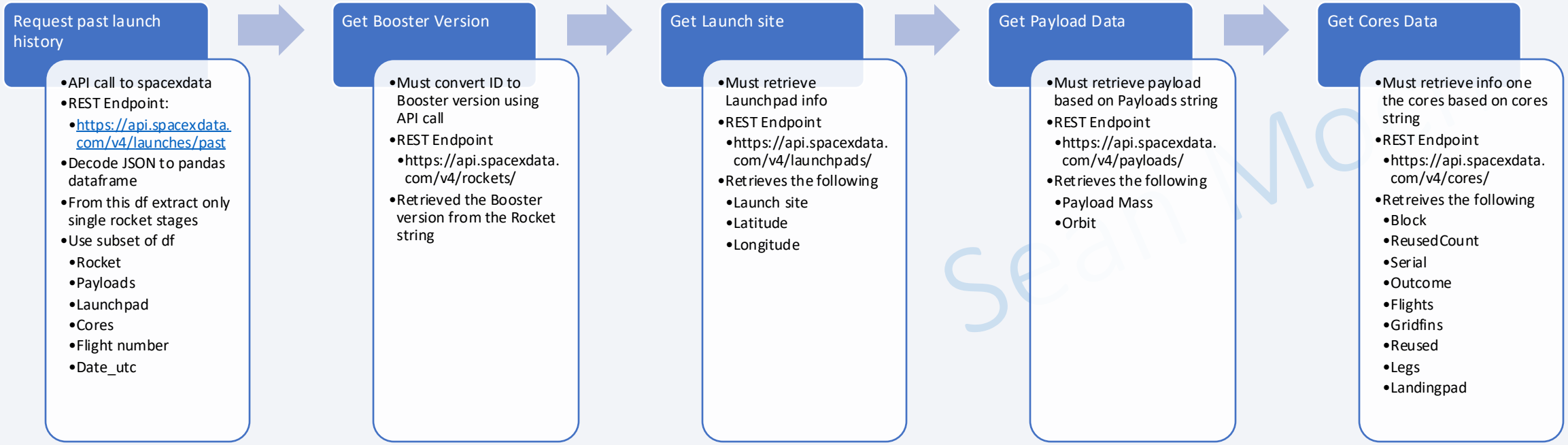
# Methodology

## Executive Summary

- Data collection methodology:
  - Data collected using API calls and Webscraping
- Perform data wrangling
  - Data processed using Pandas dataframe methods
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

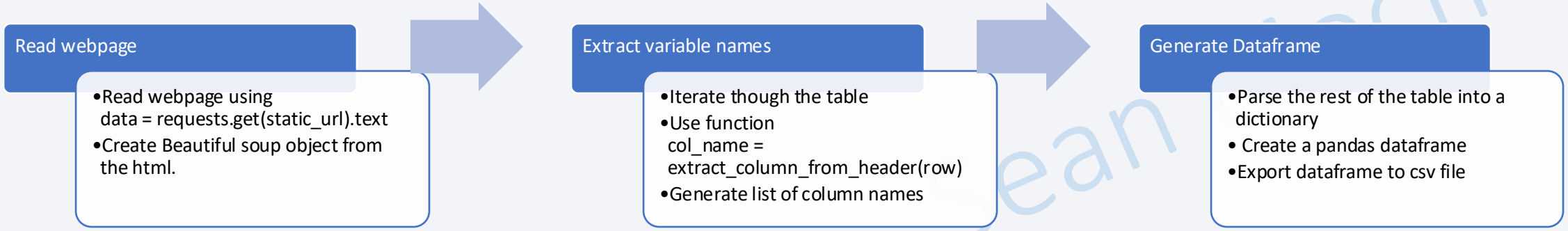
  Using scikit-learn to build, train and evaluate 4 different models.

# Data Collection – SpaceX API



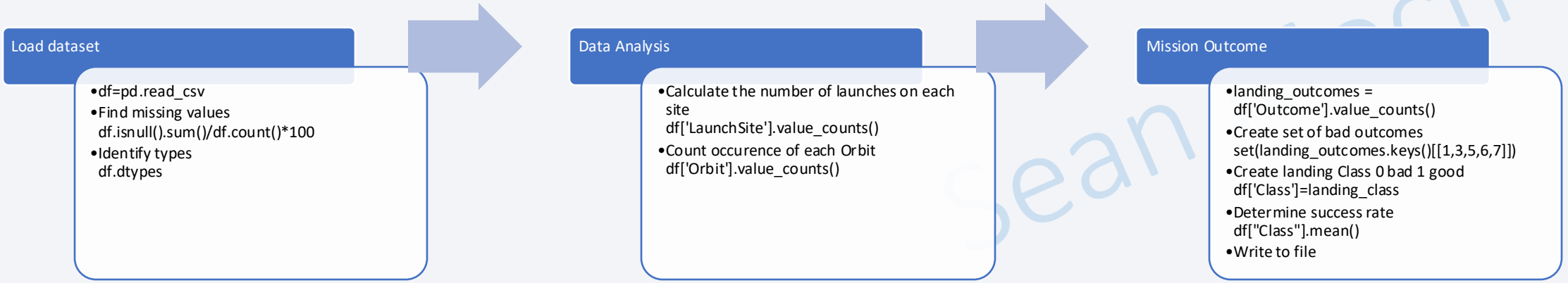

https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week1.1 jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection - Scraping




https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week1.2 jupyter-labs-webscraping.ipynb

# Data Wrangling




https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week1.3 labs-jupyter-spacex-Data wrangling.ipynb

# EDA with Data Visualization

- Categorical plots
  - Payload Mass v Flight Number – coloured by Class (success/fail)
  - Launch Site v Flight Number – coloured by Class (success/fail)
  - Launch Site v Payload Mass – coloured by Class (success/fail)
  - Orbit Type v Flight Number – coloured by Class (success/fail)
  - Orbit Type v Payload Mass – coloured by Class (success/fail)
- Bar Chart
  - Success Rate v Orbit Type
- Line Plot
  - Success Rate v Launch Year

https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week2.2 jupyter-labs-eda-dataviz.ipynb

# EDA with SQL

- Queries performed.
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week2.1_jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

- To visualise the launch locations
  - Added Circle and Marker text for each launch site
  - Can see if sites are close to coast, equator
- To visualise success/failure for each site
  - At each site added a cluster with a coded dot for each launch
  - Can see good/bad launches at each site
- Added a distance line from site to nearest coast
  - Line with calculated distance marker added
  - Can easily see the distance and direction to coast from site



https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week3.1lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

- Plotly Dash has been used to create a simple dashboard app
- The following Elements were added
  - Dropdown list to enable Launch Site selection. Default – All Sites
  - Pie chart displaying Total Successful launches for each site
    - If a specific site is selected then a success/failure Pie chart is displayed
  - Slider to enable payload weight range selection
  - Scatter Chart to show relationship between Success and Payload weight
    - Colour coded to Booster version
    - Filtered by the payload range slider
    - Filtered by Launch site dropdown

https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week3.2 spacex_dash_app.py

# Predictive Analysis (Classification)

**Data Preparation**

- Two dataframes loaded
- Y column extracted
- X data transformed using ***StandardScaler***
- ***Test_train_split*** used to create training and test datasets

**Classification Models**

- Models trained:-
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K Nearest Neighbors
- GridSearchCV used to train the models
- Several Hyperparamters used to find best score

**Assessment**

- Score of the training using the best Hyperparameters for the specific model
- Evaluation of the test data using the trained model
- Score of the predicted test data
- Confusion Matrix

- Classification Models built using Scikit-learn
- All models except Tree performed similarly

https://github.com/mocher72/datascience/blob/main/AppliedDataScienceCapstone/Week4.1SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

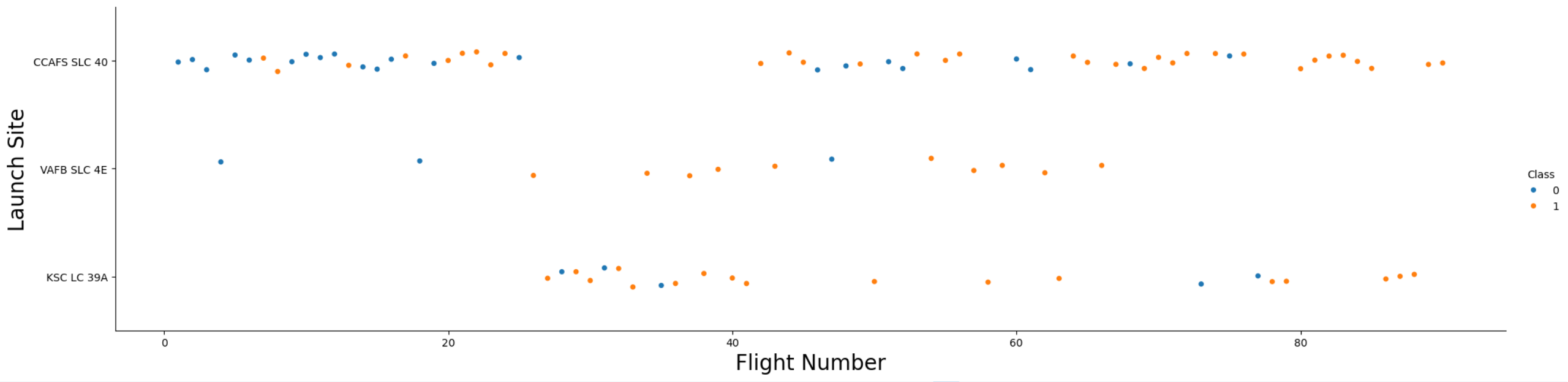
# Results

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

Section 2

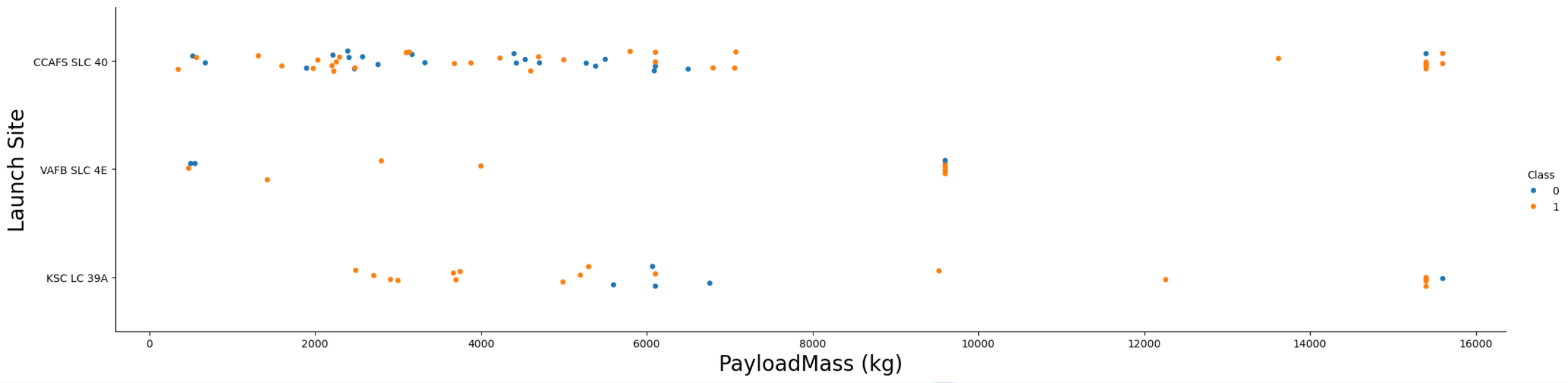
# Insights drawn from EDA

# Flight Number vs. Launch Site




- Three launch sites are used
- Flight success is encoded 0=Fail, 1=Success
- All flights after flight 78 are successful
- Most flights are from CCAFS SLC 40

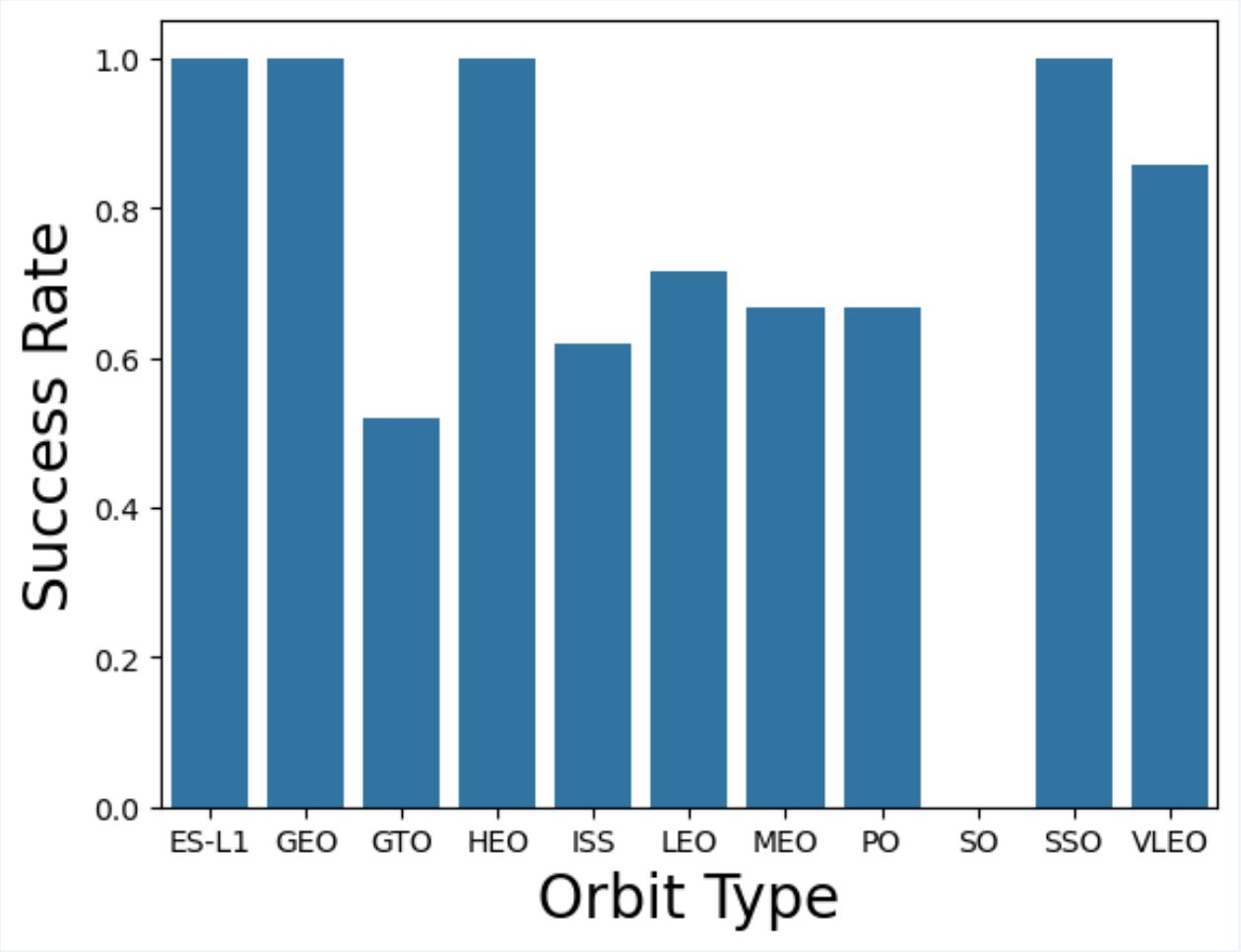
# Payload vs. Launch Site



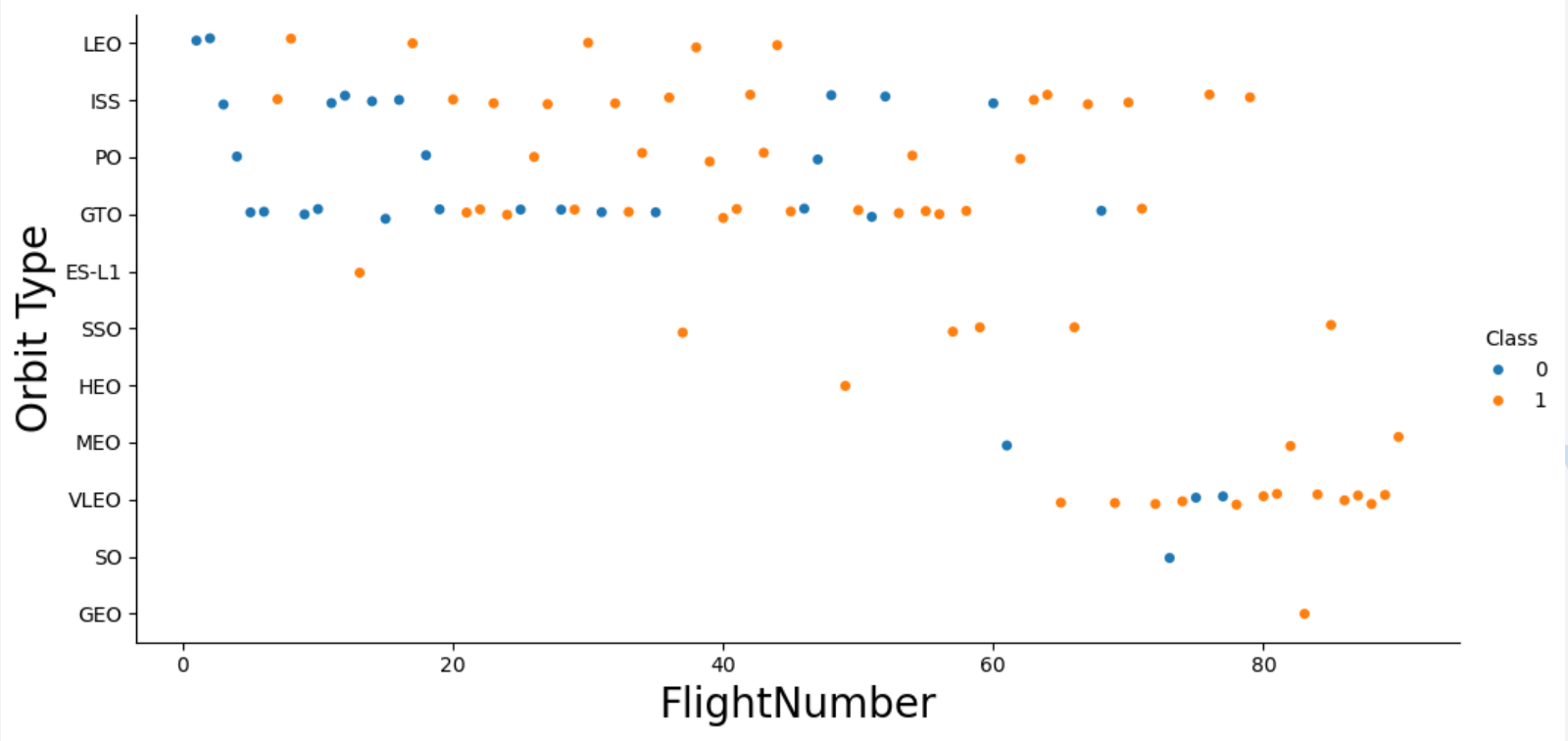

- Large variation of payloads below 7000kg
- No payloads above 10000kg on VAFB SLC 4E
- Above 10000kg mostly maximum payloads are launched

# Success Rate vs. Orbit Type

- Orbit types ES-L1, GEO, HEO & SSO all have 100% success
- SO orbit type has zero success

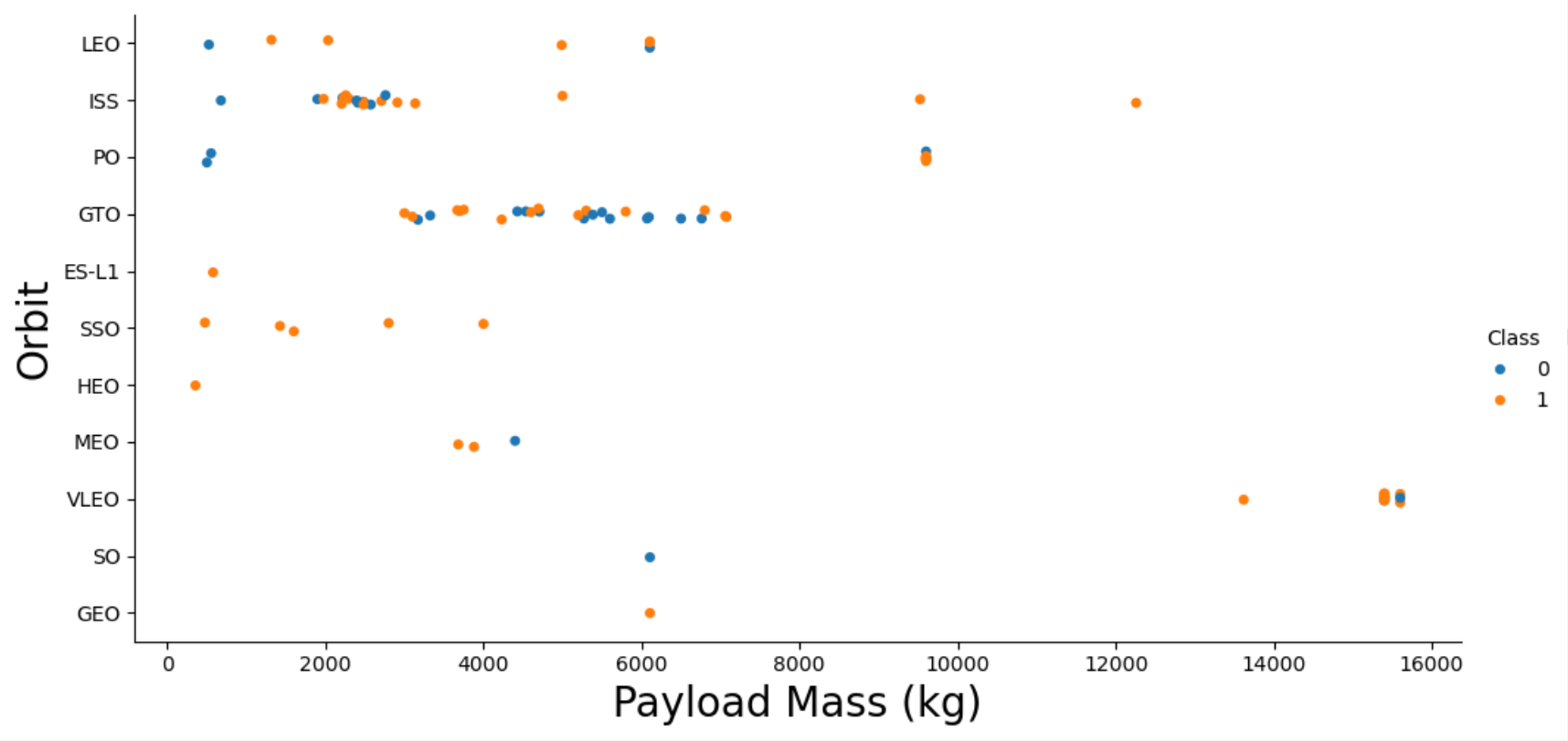


# Flight Number vs. Orbit Type




- Most later flights have been to VLEO orbit with high success rate
- ISS had some failures in the mid 50s flight number range.

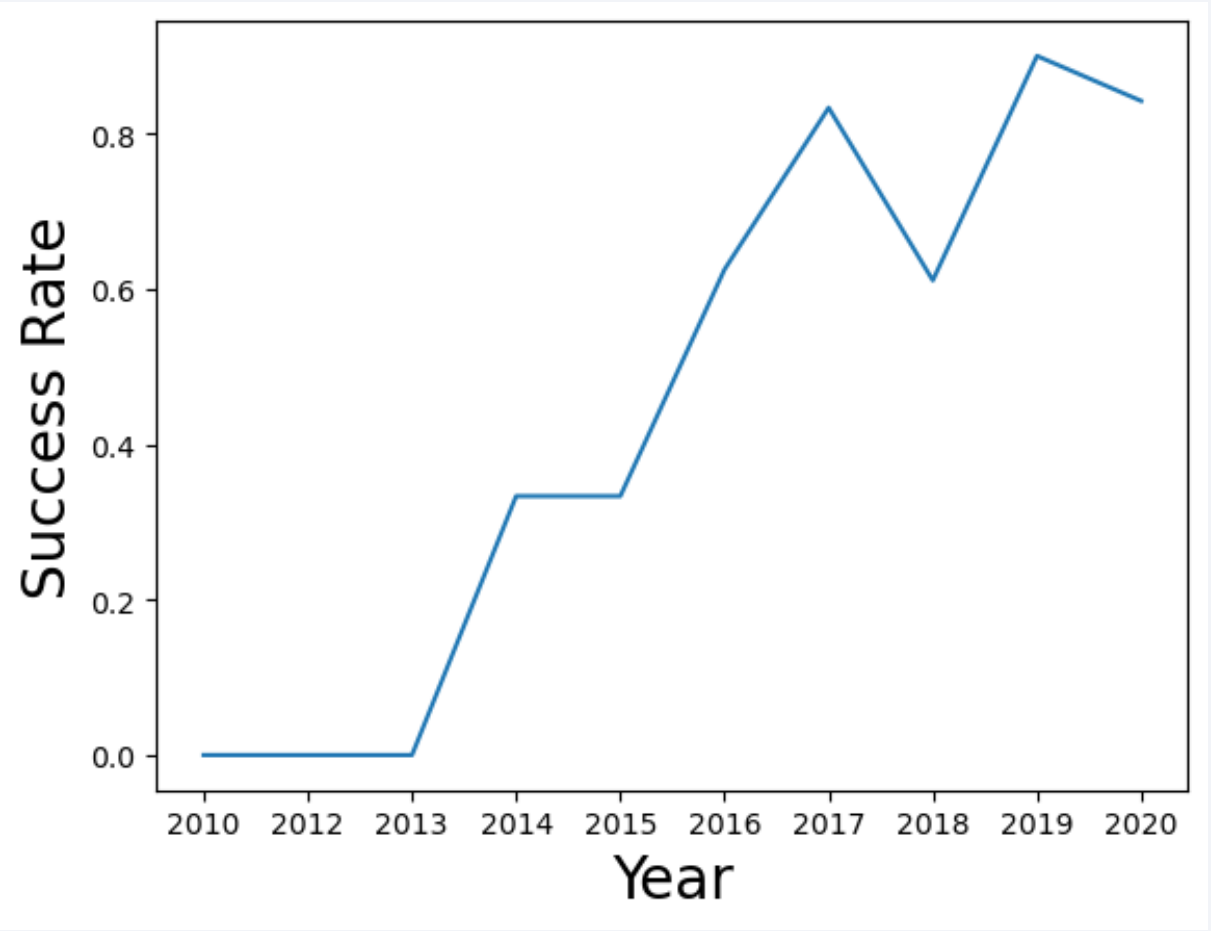
# Payload vs. Orbit Type




- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

# Launch Success Yearly Trend

- You can observe that the success rate since 2013 kept increasing till 2020 with a little dip in 2018



# All Launch Site Names

- Unique Launch Sites:-
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40
- Query string:-
  - cur.execute("SELECT DISTINCT launch_site FROM SPACEXTBL")
  - SELECT selects the ‘launch_site’ entries and filters using the DISTINCT keyword

# Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA` results are shown to the right
- Query:-
  - cur.execute("SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5")
  - Selecting all entries where the launch_site begins ‘CCA” retuning a maximum of 5 records

[('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')]

# Total Payload Mass

- Total payload carried by boosters from NASA is 45596kg
- Query:-
  - cur.execute("SELECT SUM(PAYLOAD_MASS__KG_) as sum_payload_mass FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)'")
  - This query selects the payload data and sums it returning this as ‘sum_payload_mass’ filrering the entries where the customer is ‘NASA (CRS)’

# Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 is 2928.4kg
- Query:-
  - cur.execute("SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEXTBL WHERE booster_version = 'F9 v1.1'")
  - Payload mass values are averaged and returned as 'average_payload_mass' with entries filtered using booster_version –atching 'F9 v 1.1'

# First Successful Ground Landing Date

- First successful landing outcome on ground pad
  - 2015-12-22
- Query:-
  - cur.execute("SELECT MIN(date) AS first_successful_landing_date FROM SPACEXTBL WHERE landing_outcome LIKE '%success%' AND landing_outcome LIKE '%ground pad%'")
  - Selects the MIN date from records filtered for landing outcome having ‘success’ AND ‘ground pad’ within text.

# Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - 'F9 FT B1022’
  - 'F9 FT B1026’
  - 'F9 FT B1021.2’
  - 'F9 FT B1031.2’
- Query:-
  - cur.execute("SELECT Booster_Version FROM SPACEXTBL WHERE landing_outcome LIKE '%Success%' AND landing_outcome LIKE '%Drone Ship%' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000")
  - Selects the booster versions landing on Drone_Ship successfully and payload 4000-6000kg

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
  - Successful missions: 100
  - Failed missions: 1
- Query:-
  - cur.execute("SELECT COUNT(*) as success_count FROM SPACEXTBL WHERE mission_outcome LIKE '%Success%'")
  - cur.execute("SELECT COUNT(*) as failure_count FROM SPACEXTBL WHERE mission_outcome LIKE '%Failure%'")
  - First query counts the mission successes
  - Second query counts the failures

# Boosters Carried Maximum Payload

- Boosters which have carried the maximum payload mass
  - ['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 ']
- Query:-
  - cur.execute("SELECT booster_version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)")
  - Return the booster_version where the payload matches the maximum payload found

# 2015 Launch Records

- Failed landing_outcomes in drone ship, launch month, booster versions, and launch site names for in year 2015
  - ('01', 'F9 v1.1 B1012', 'CCAFS LC-40', 'Failure (drone ship)')
  - ('04', 'F9 v1.1 B1015', 'CCAFS LC-40', 'Failure (drone ship)')
- Query:-

```
SELECT substr("Date", 6, 2) AS month, "Booster_Version", "Launch_Site",
"Landing_Outcome"

FROM SPACEXTBL

WHERE substr("Date", 1, 4) = '2015' AND "Landing_Outcome" = 'Failure
(drone ship)';
```

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

- Landing outcomes:-

1. No attempt 10
2. Success (drone ship) 5
3. Failure (drone ship) 5
4. Success (ground pad) 3
5. Controlled (ocean) 3
6. Uncontrolled (ocean) 2
7. Failure (parachute) 2
8. Precluded (drone ship) 1

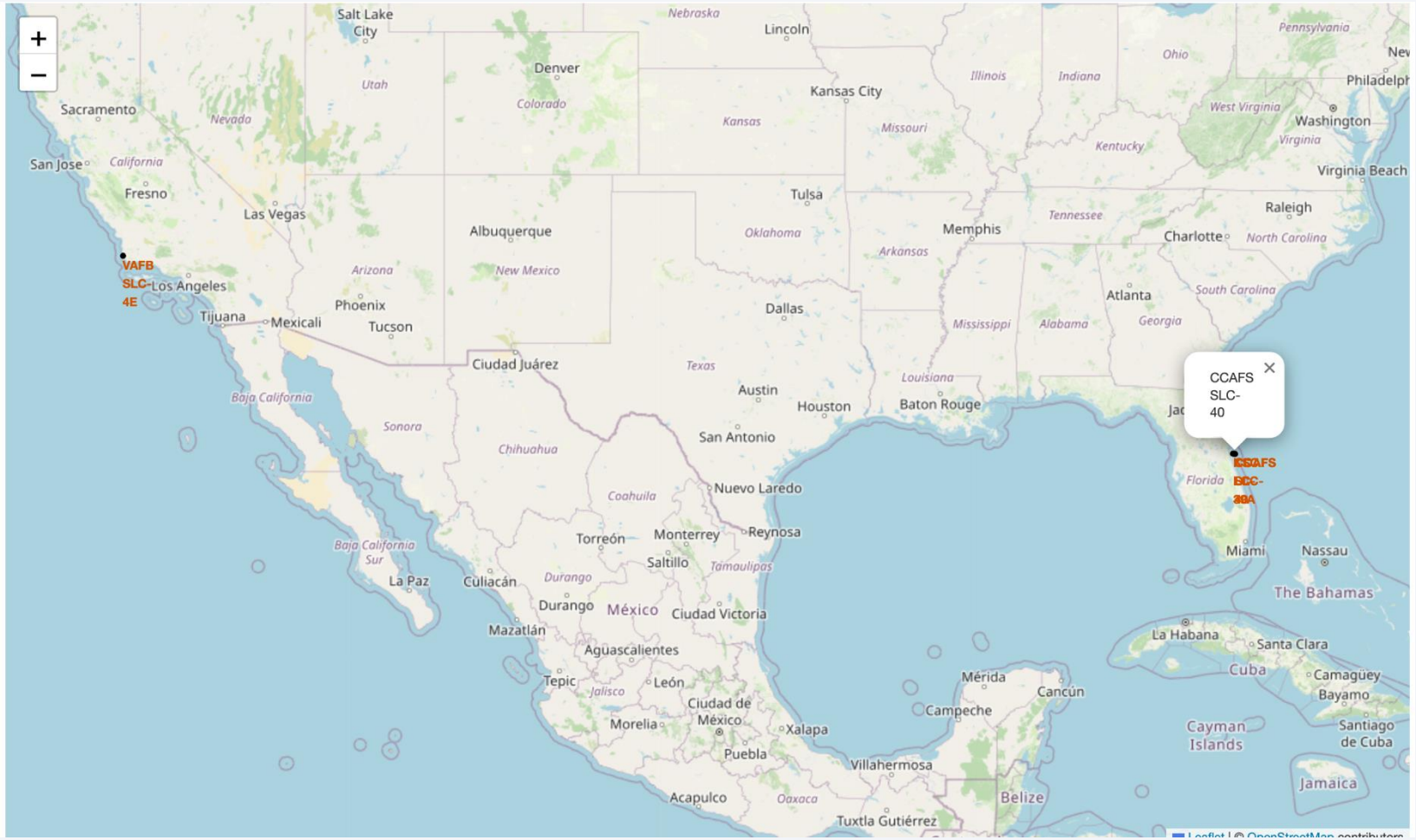
- Query:-
  - query = """SELECT "Landing_Outcome", COUNT(*) FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT(*) DESC;"""

Section 3

# Launch Sites Proximities Analysis

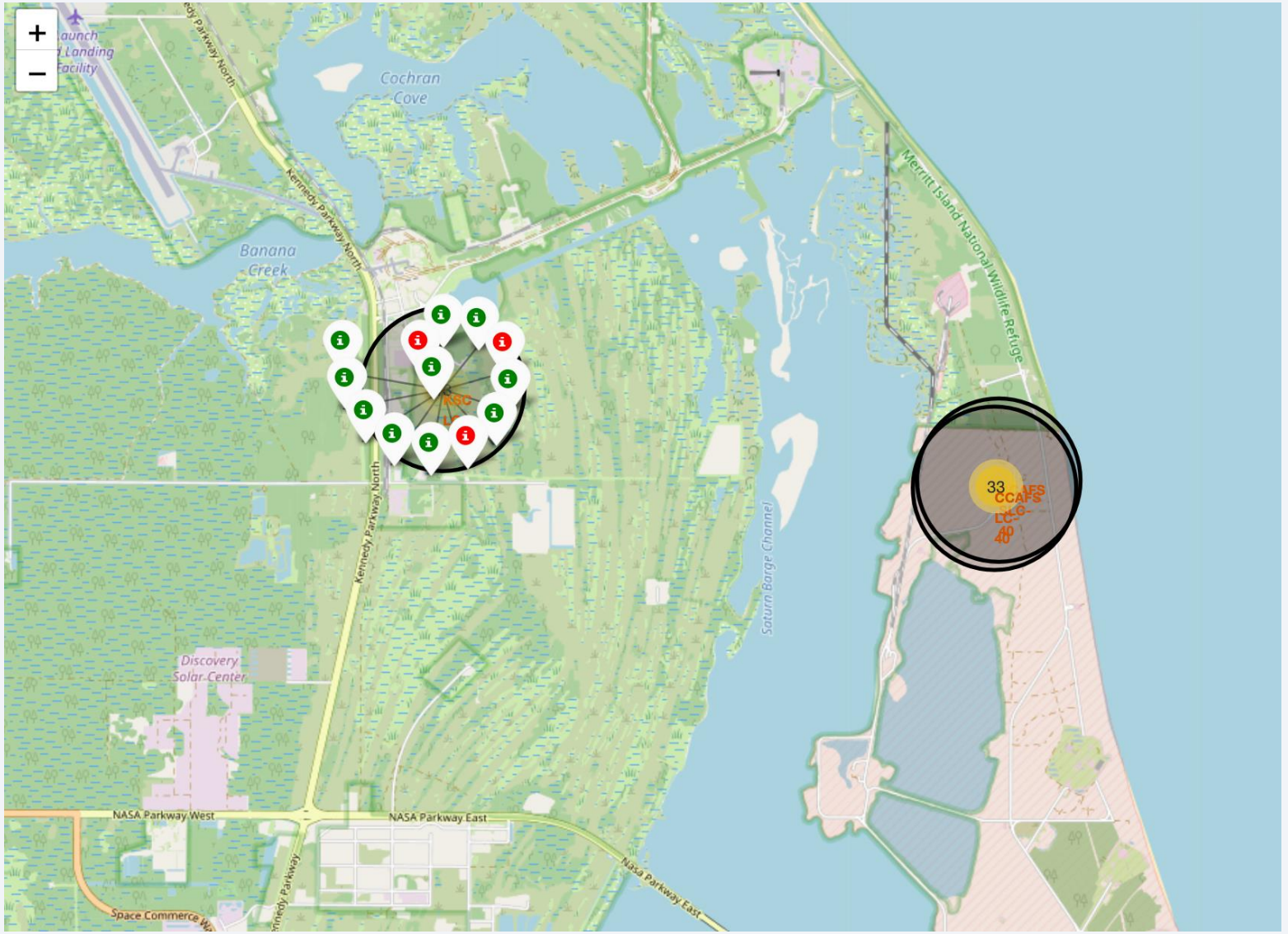
# Overview of Launch Sites

- All launch sites are in close proximity to the coast.
- Launch sites in Florida a in close proximity to each other



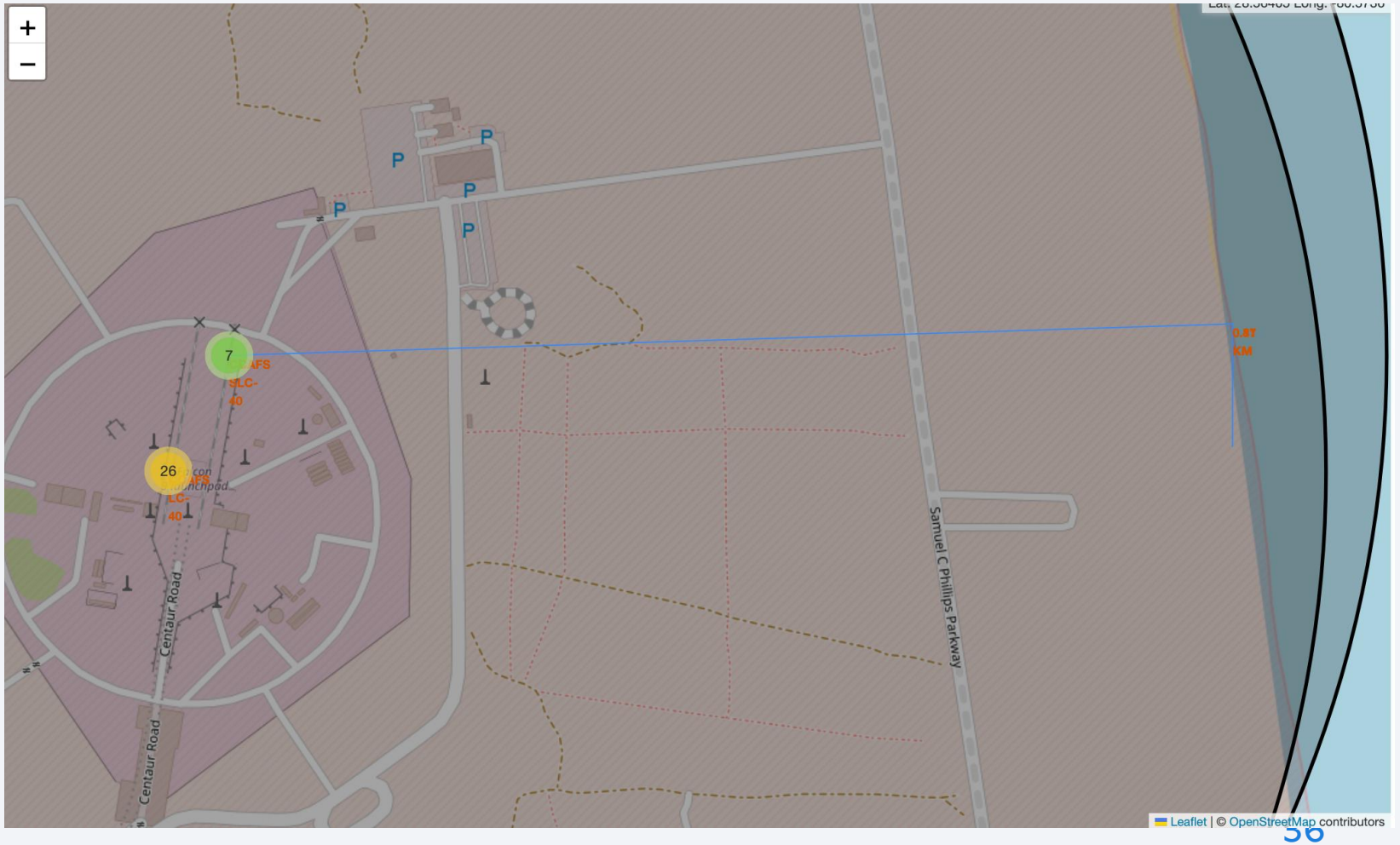
# Colour coded Launch Outcomes

- For each site, once you have zoomed in, you can expand the site to show the launches with colour coding to show if it was a success or failure.



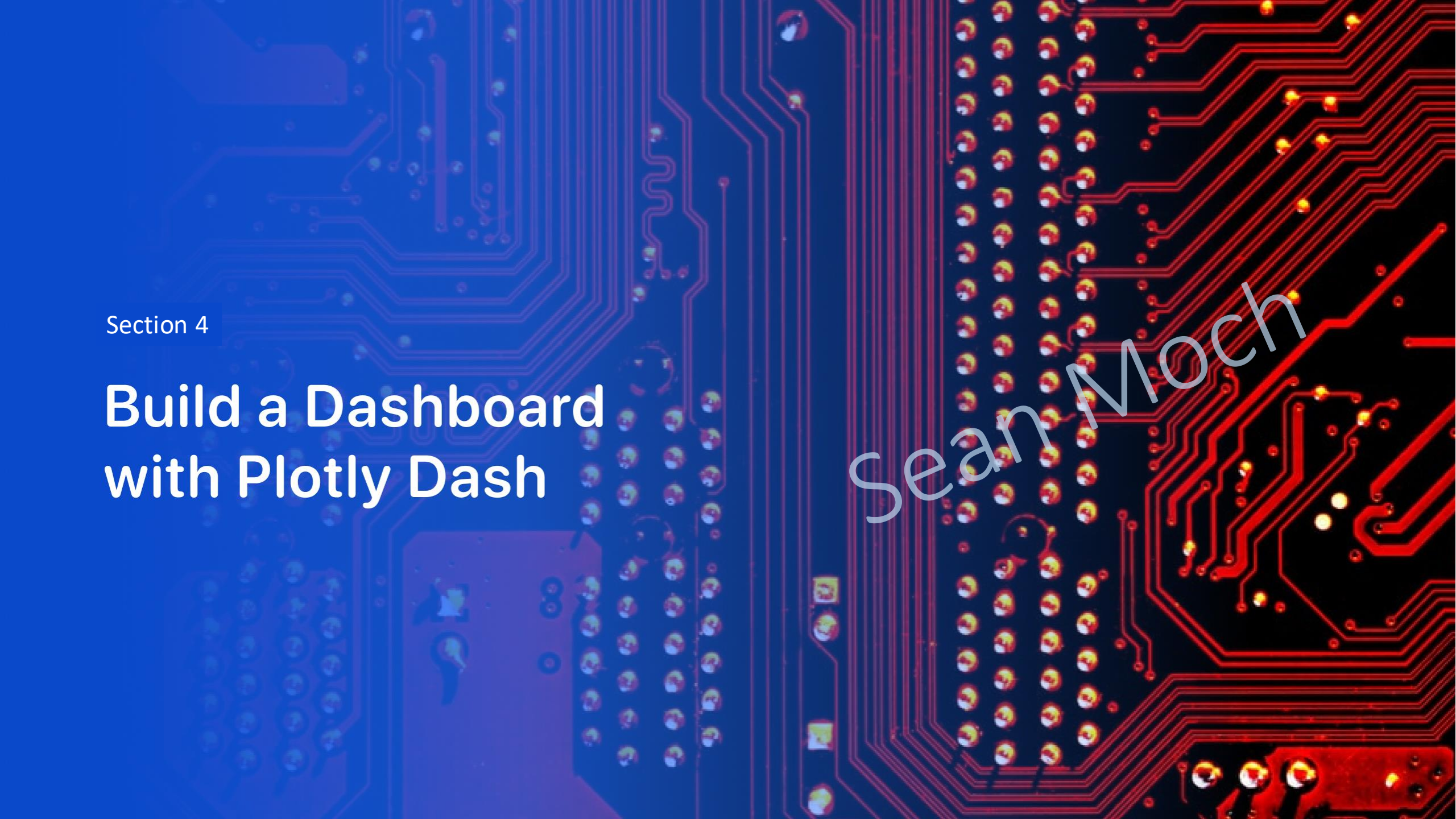
# Coast Proximity to CCAF SLC-40

- Zoomed area showing launch pad locations
- Distance line and marker to coastline shown in blue
- Calculated distance is 0.87km



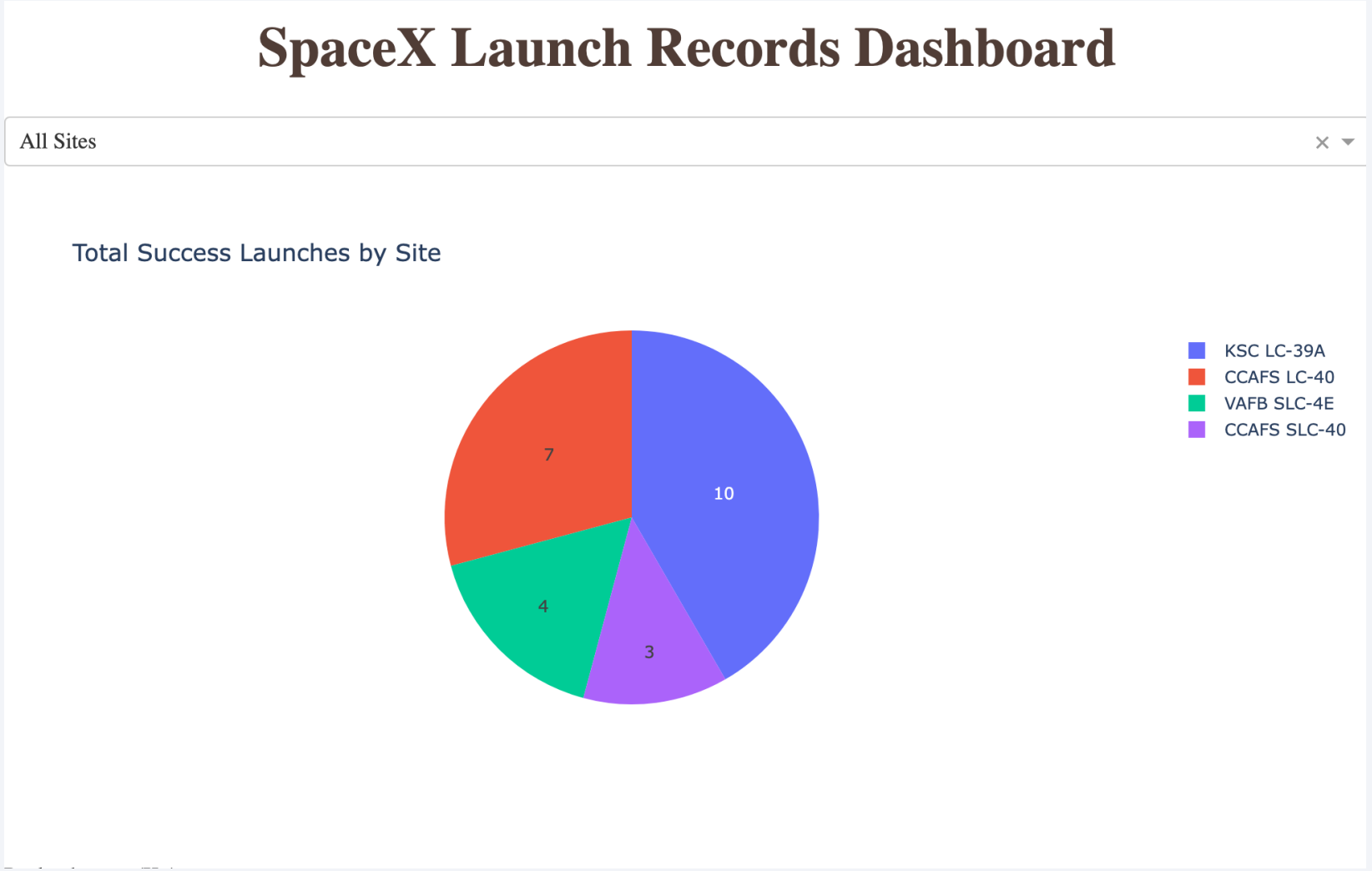
Section 4

# Build a Dashboard with Plotly Dash



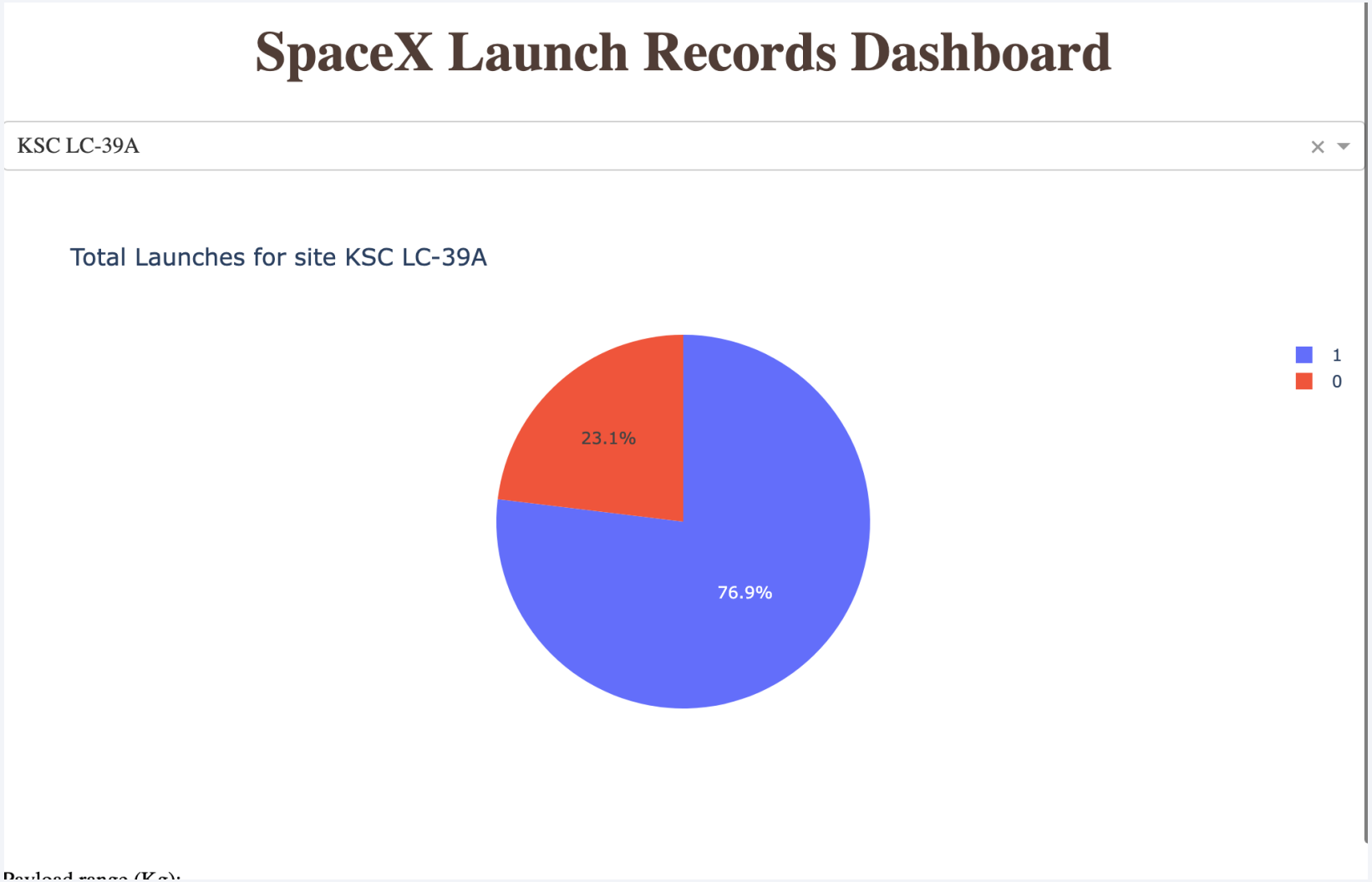
# Success Count All Sites

- Pie chart is colour coded by launch site
- Total successful launches per site is shown in the figure
- Site KSC LC-39A has the most successful launches



# Site with Highest Success Ratio

- By looking at each site we can determine the success ratio at each site.
- When individual site is selected the success/failure ratios are shown.
- KCS LC-39A has the highest success ratio at 76.9%



# Success v Payload

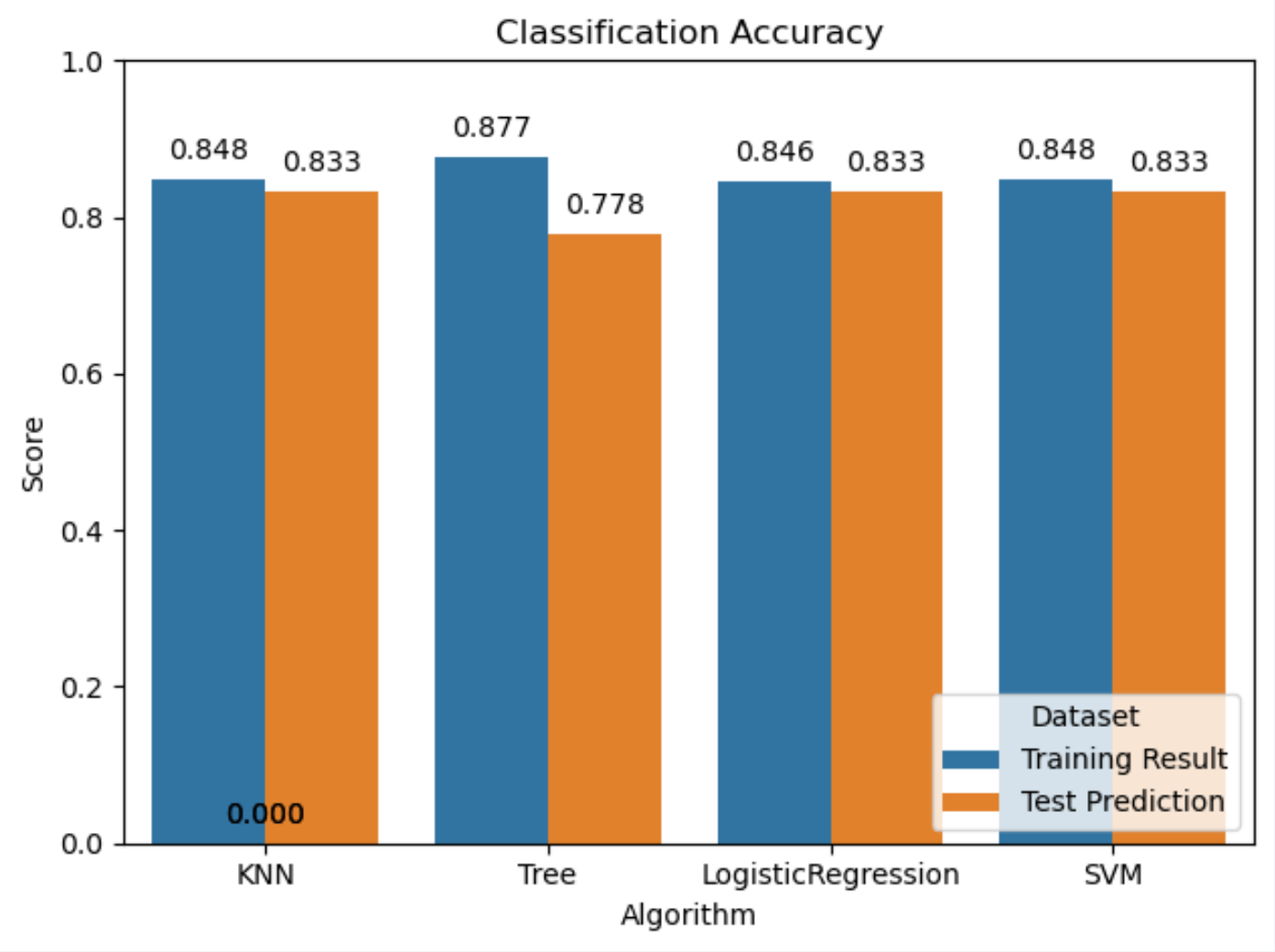
- Success v Payload with sites selected from the dropdown are displayed.
- Class shows the success or failure.
- Launches are colour coded according to the booster technology.

Section 5

# Predictive Analysis (Classification)

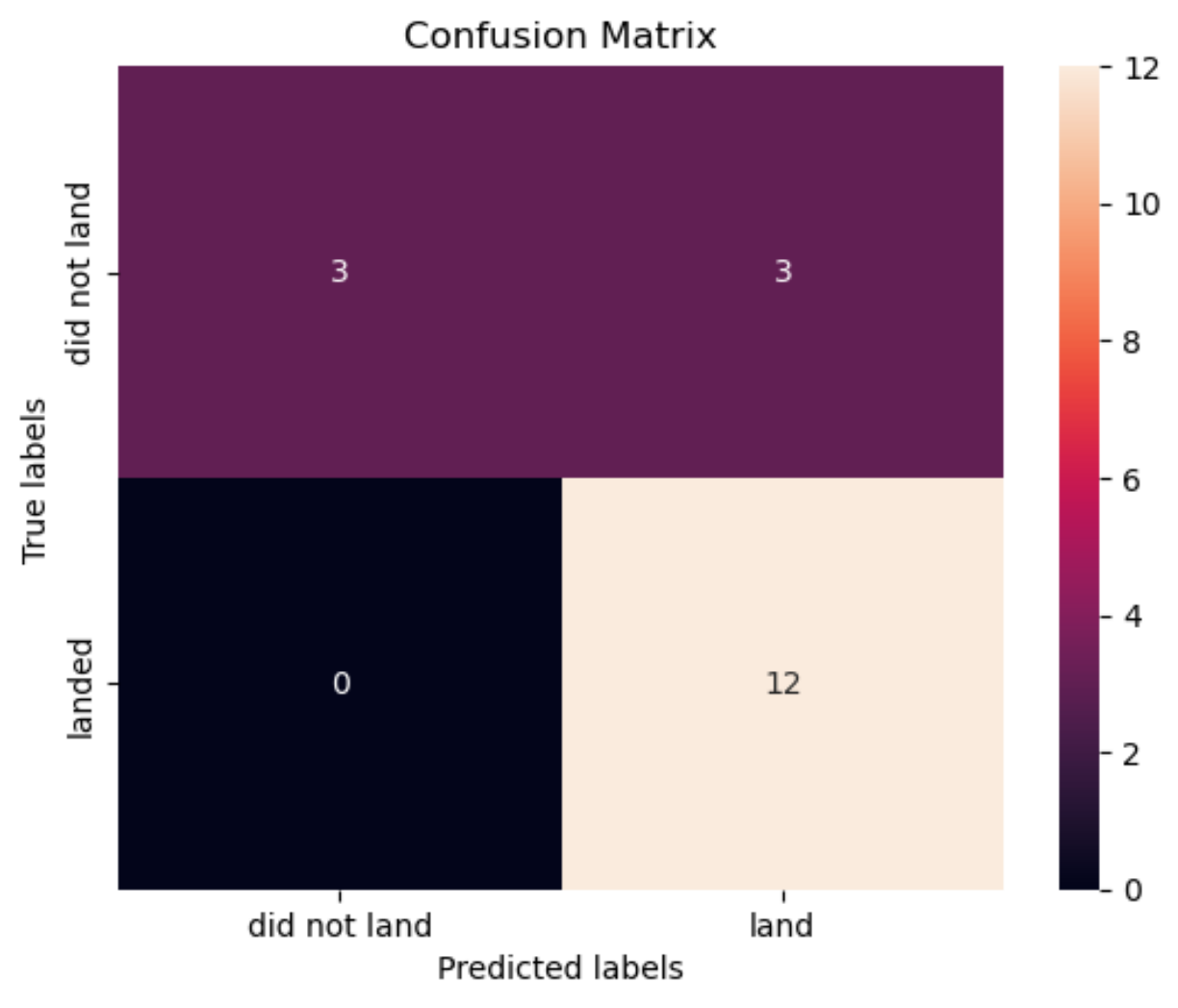
# Classification Accuracy

- Here is shown the classification accuracy for the four different prediction models used.
- The training accuracy and the test dataset prediction accuracy are shown.
- Although the Tree classifier has the highest training accuracy it has the lowest prediction score so is overfitting.
- All other algorithms have the same prediction accuracy.



# Confusion Matrix

- Three methods tied for the best performance on the test dataset with KNN, Logistic Regression and SVM all yielding the same confusion matrix.
- Predictions of the landed stages was 100% correct.
- For the failures the predictions were poor with 3 correct and 3 false positives.

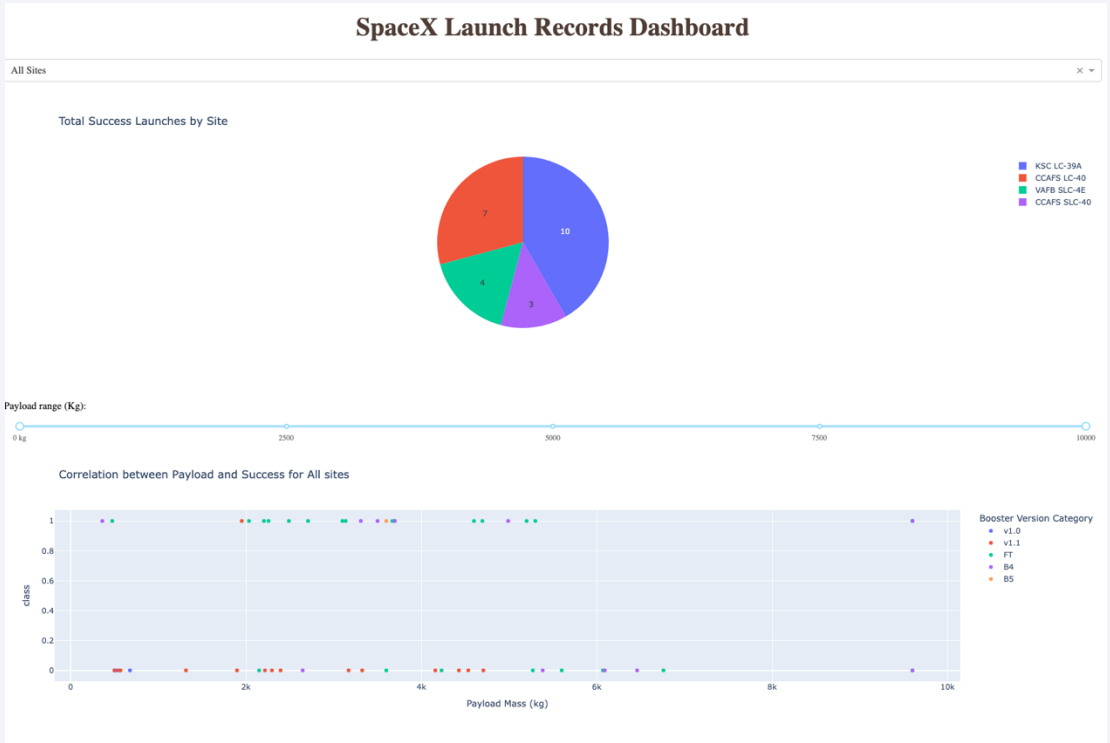


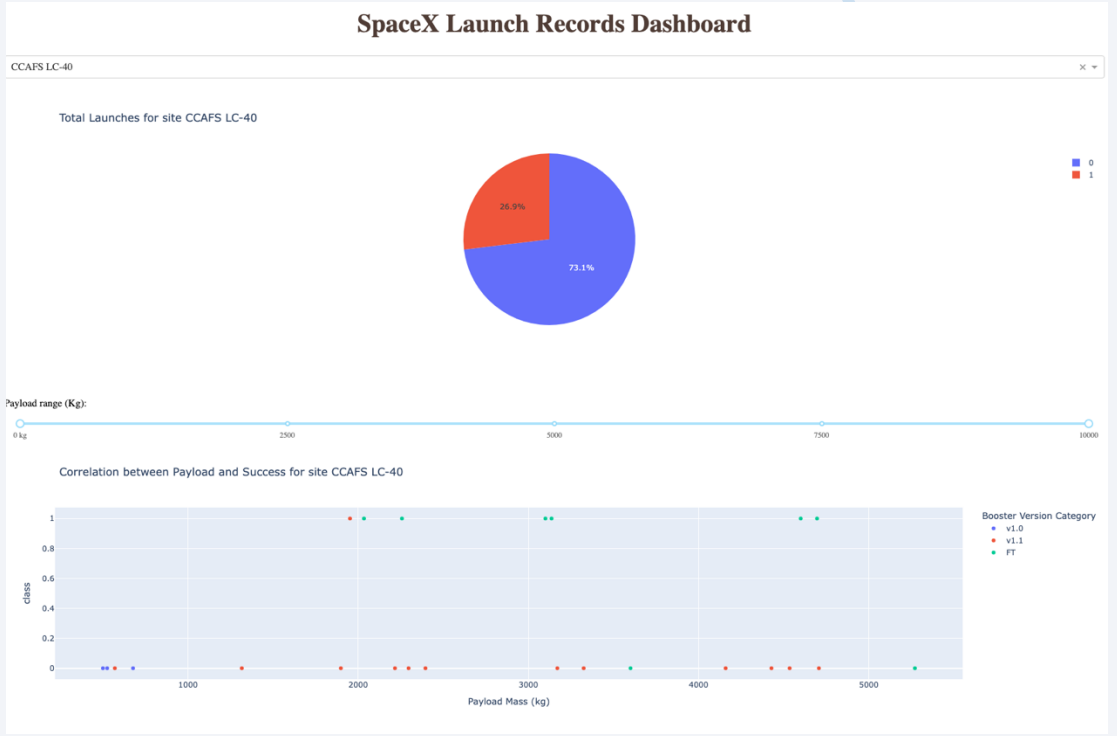
# Conclusions

- All four methods used for modelling yield a model which attains reasonable predictions.
- KNN, Logistic Regression and SVM all yield similar performance for training and prediction.
- Tree method overfits to the data so has better training score and worse test score.
- To improve the false positives for the did not land outcomes further data is required to improve the model.

# Appendix

- Dash App screenshot All sites v Single Site



# Thank you!