IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Dr. Sean Hansen-Romu
November 14, 2023

# Outline

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

# Executive Summary

- The methodologies used were:
  - Data collection
  - Data wrangling
  - Exploratory data analysis (EDA) using visualization and SQL
  - Interactive visual analytics using Folium and Plotly Dash
  - Predictive analysis using classification models
- Summary of results
  - It was possible to collect publicly facing data and clean the data for use in a ML algorithm
  - Able to deliver a machine learning model that predicts the success of a mission using a tree decision classifier with accuracy of 83%.

# Introduction

- SpaceY a [fictional] new space startup funded by Allon Musk wants to step into the commercial space business
- Business Problems?
  - What is the price of each launch?
  - When does SpaceX reuse the first stage of the rocket?

Section 1

# Methodology

# Methodology

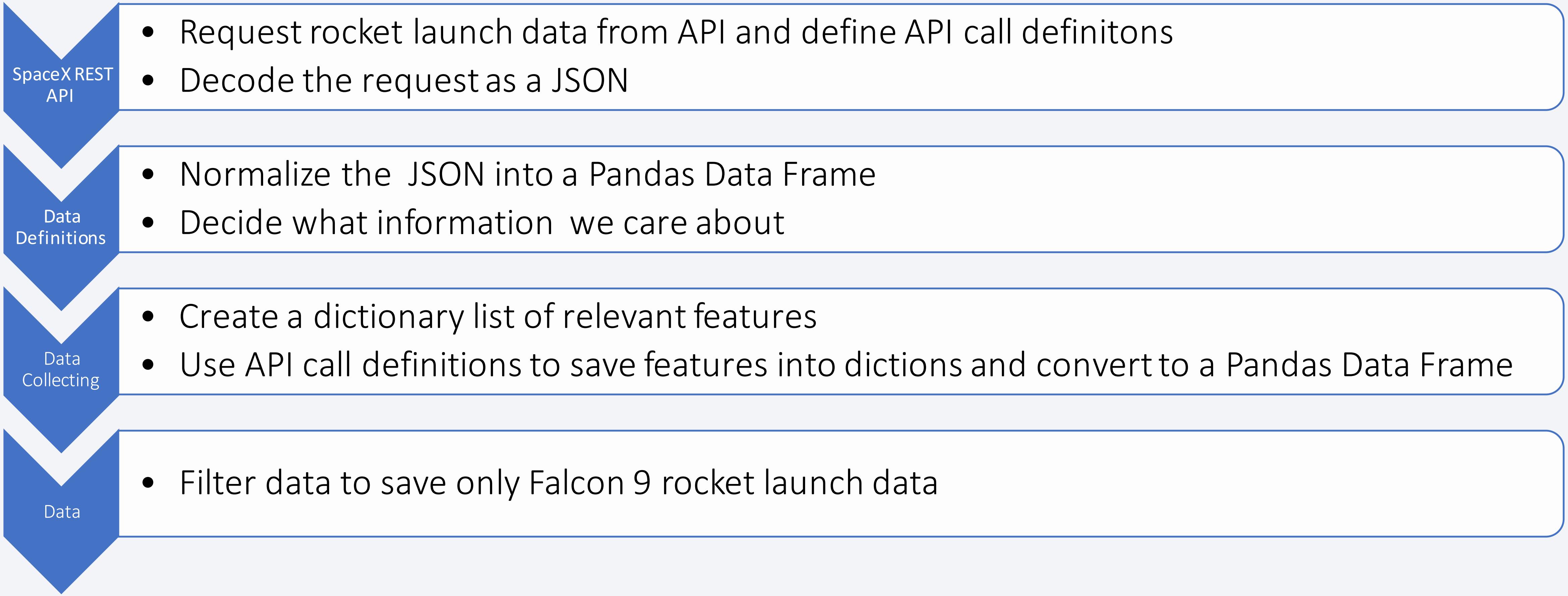
## Executive Summary

- Data collection methodology:
  - using SpaceX REST API and web scrapping
- Perform data wrangling
  - Average for missing data, One-hot-encoding for categorical data
- 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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts
- The Data was collected via API calls to SpaceX REST API and web scrapping from the internet.
  - This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

# Data Collection – SpaceX API



SpaceX REST API

- Request rocket launch data from API and define API call definitons
- Decode the request as a JSON

Data Definitions

- Normalize the JSON into a Pandas Data Frame
- Decide what information we care about

Data Collecting

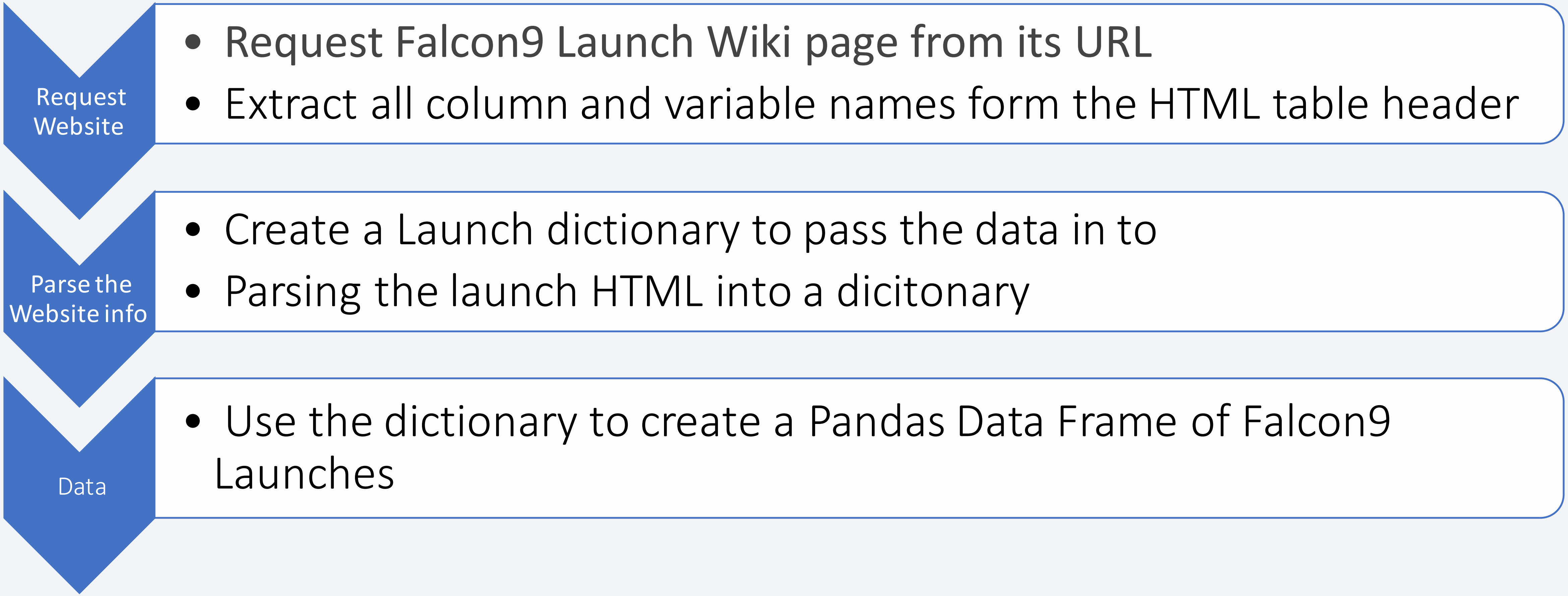
- Create a dictionary list of relevant features
- Use API call definitions to save features into dictions and convert to a Pandas Data Frame

Data

- Filter data to save only Falcon 9 rocket launch data

GitHub: Data Collection API

# Data Collection - Scraping



Request Website

- Request Falcon9 Launch Wiki page from its URL
- Extract all column and variable names form the HTML table header

Parse the Website info

- Create a Launch dictionary to pass the data in to
- Parsing the launch HTML into a dicitonary

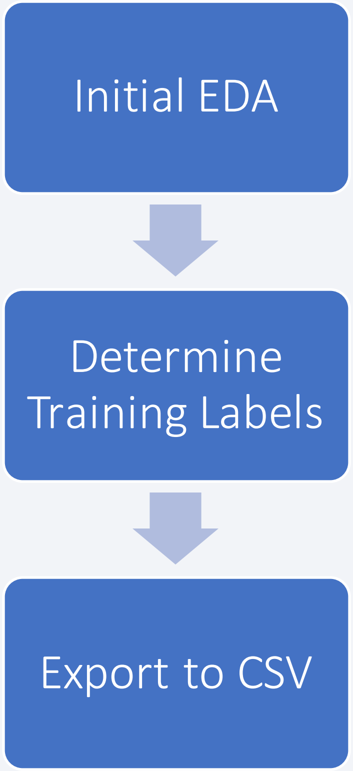
Data

- Use the dictionary to create a Pandas Data Frame of Falcon9 Launches

GitHub: Data Collection with Web Scrapping

# Data Wrangling

- The data was process by first performing initial exploratory data analysis (EDA)
  - Checking for NULL values and data types
  - Perform some EDA by calculating a few values like launcher per site, type of orbit aiming, and mission success rate.
- After performing some initial EDA we export the data for further analysis.



GitHub: Data Wrangling

# EDA with Data Visualization

- We plotted a number of charts:
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload Mass vs. Launch Site
  - Flight Number vs. Orbit
  - Payload Mass vs. Orbit
  - Average Success Rate vs. Year
- I obtained preliminary insights about how each important variable would effect the success rate

GitHub: Exploratory Data Analysis with Visualization

# EDA with SQL

- Summary of the SQL queries you performed
  - Unique launch site
  - Launch records
  - Total payload mass from NASA
  - Average Payload mass for booster F9 v1.1
  - Date of the first landing success
  - Booster version from landing on successful landing on drone ship
  - Failure rate
  - Booster name with max payload
  - Dates of various records
  - Landing outcomes ranked by particular dates

GitHub: Exploratory Data Analysis with SQL

# Build an Interactive Map with Folium

- Added launch site on the map
  - To visualize the locations on a map
- Added the success and failures for each launch site
  - To visualized the success and failures on the sites
- Added distances on the map to proximities

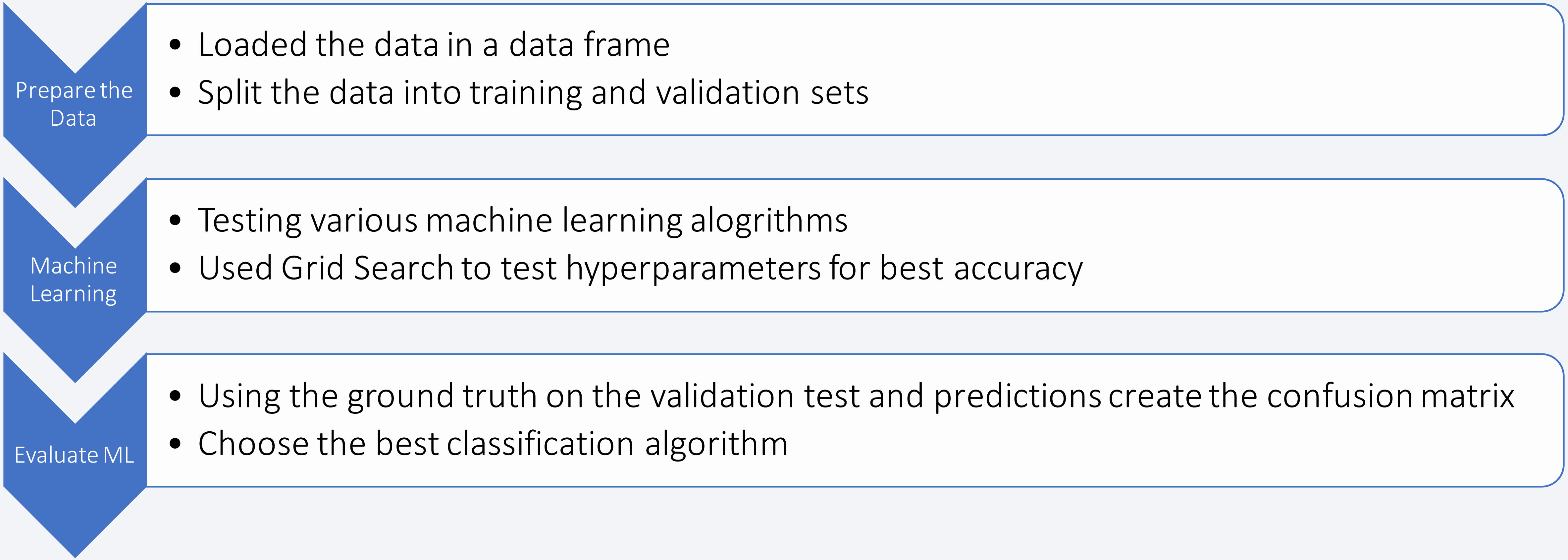
GitHub: Launch Site Locations Analysis with Folium

# Build a Dashboard with Plotly Dash

- Added an interactive success pie chart which displayed the success rate for each launch location and relatively
- Added an interactive map with a payload mass that displayed Payload Mass vs. Launch Success rate with launch location as a color hue

GitHub: Interactive Visual Analytics and Dashboards

# Predictive Analysis (Classification)



Prepare the Data

- Loaded the data in a data frame
- Split the data into training and validation sets

Machine Learning

- Testing various machine learning alogrithms
- Used Grid Search to test hyperparameters for best accuracy

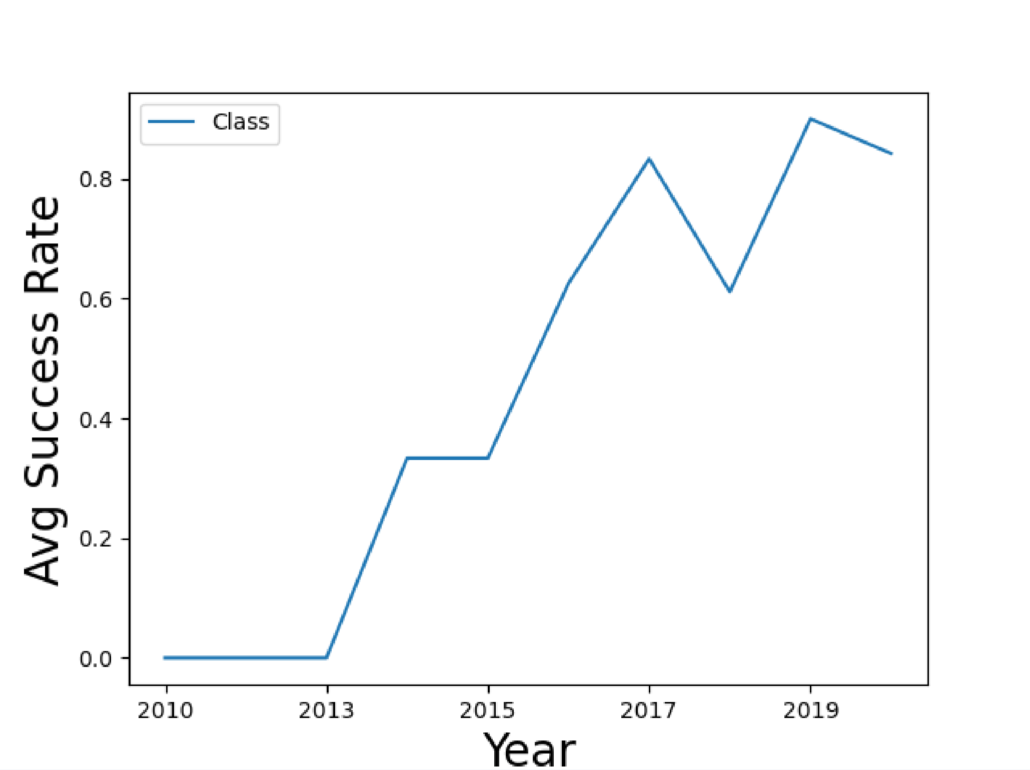
Evaluate ML

- Using the ground truth on the validation test and predictions create the confusion matrix
- Choose the best classification algorithm

GitHub: Machine Learning Perdiction

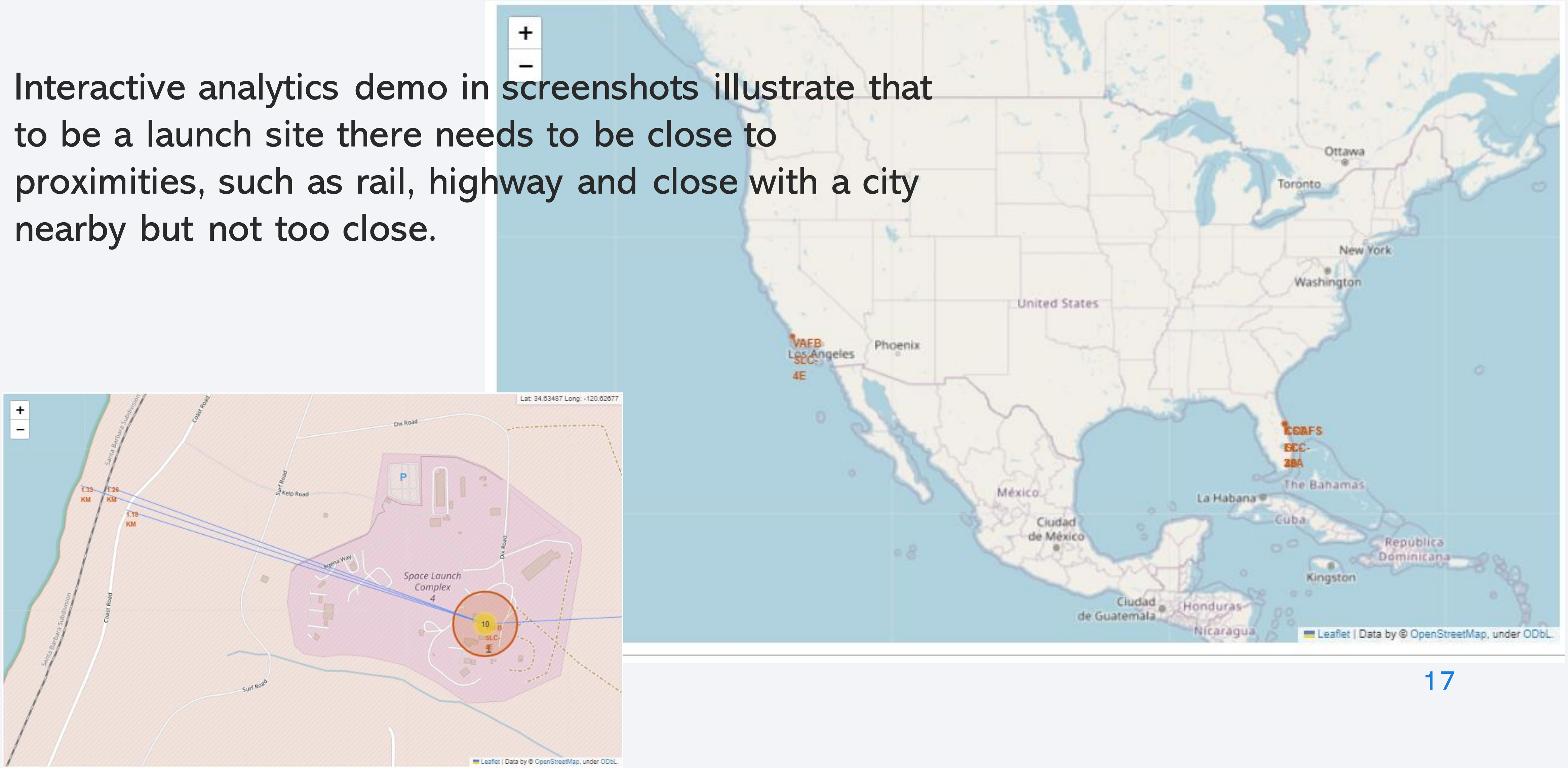
# Results

- If we are to emulate SpaceX in terms of success we need to have a realistic time frame and plans in place
- Perhaps using similar booster than the one used in later times, but understanding that lead times might be comparable



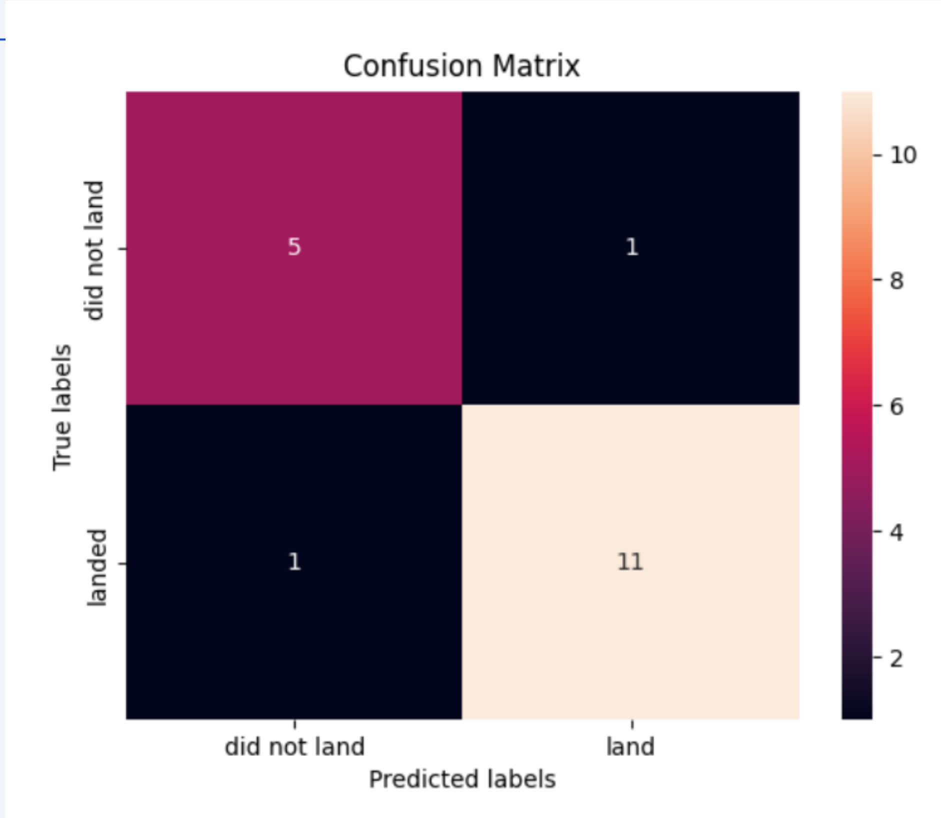
# Results

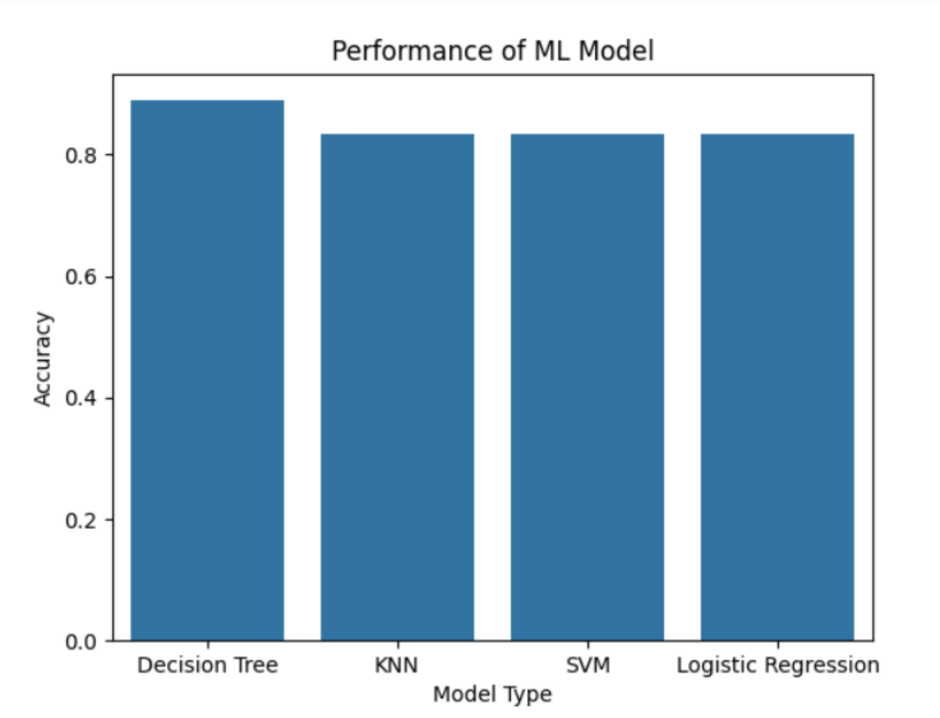
- Interactive analytics demo in screenshots illustrate that to be a launch site there needs to be close to proximities, such as rail, highway and close with a city nearby but not too close.

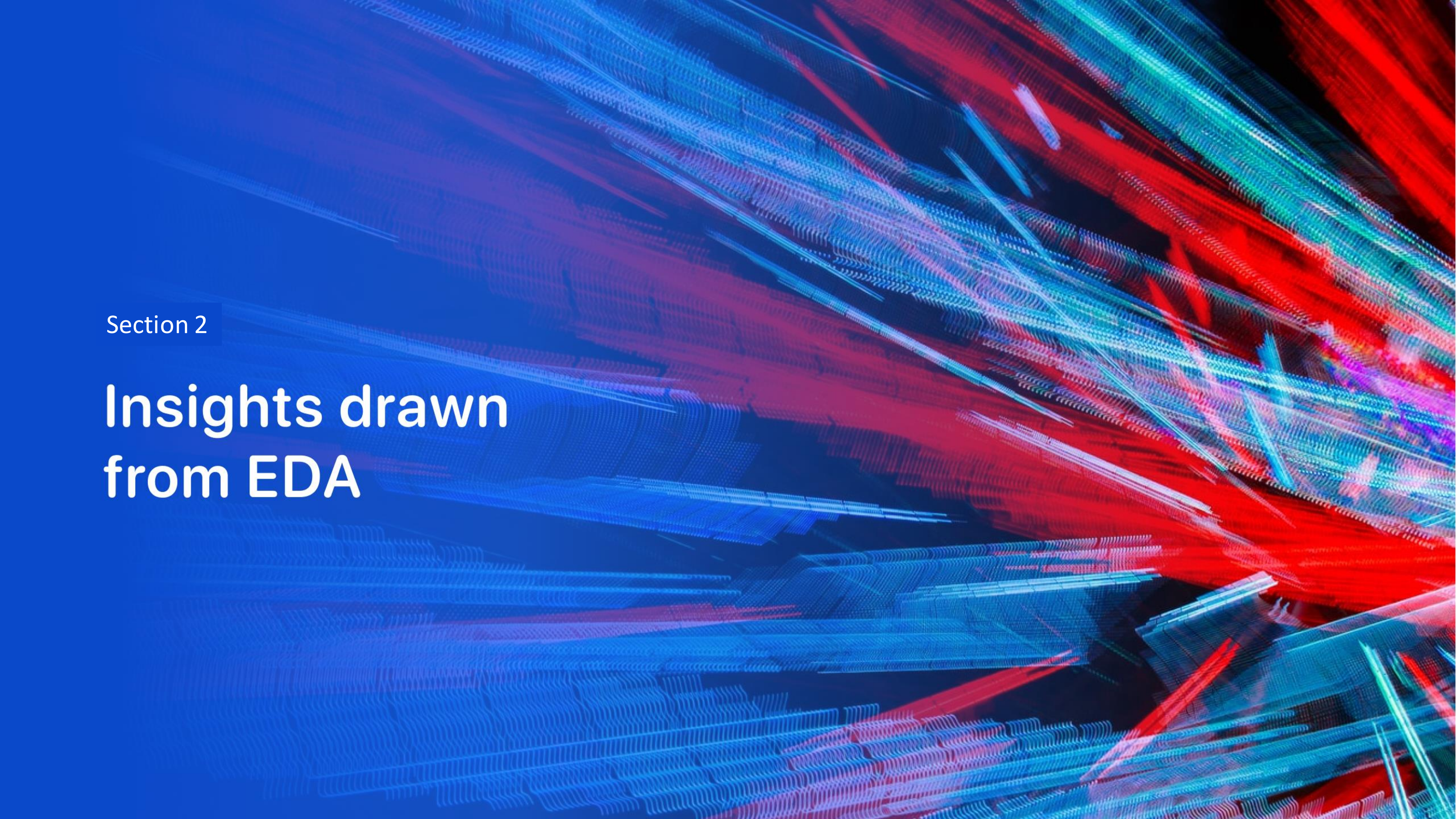


# Results

- The best model for predictive classification for launch success based on our feature set is a decision tree classifier



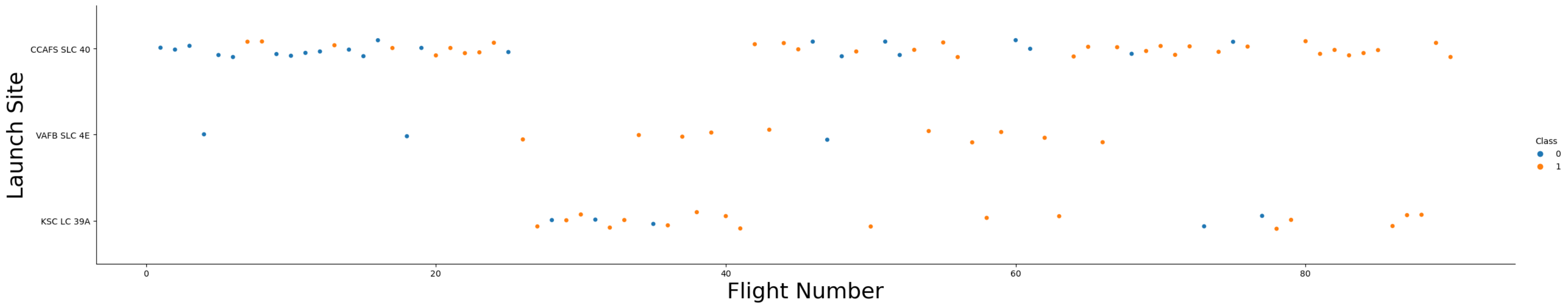




Section 2

# Insights drawn from EDA

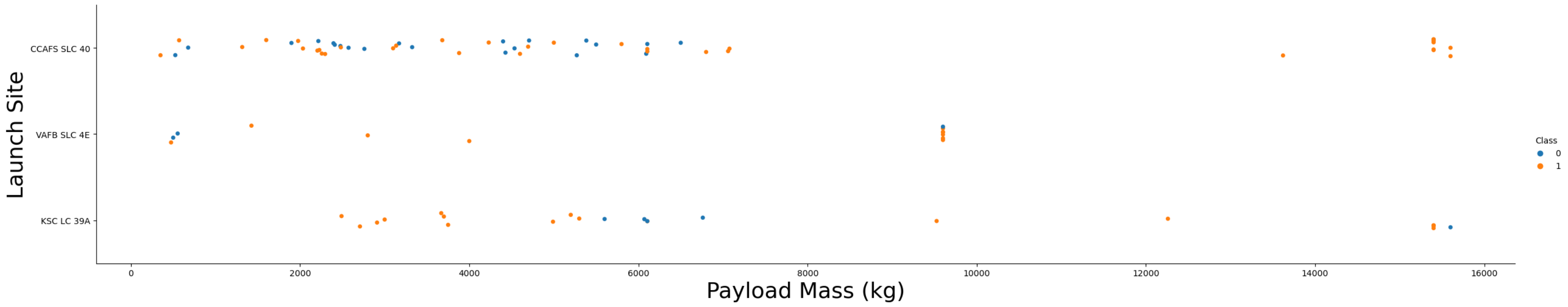
# Flight Number vs. Launch Site



A general trend of more success over time is shown

It is shown that the best site for launching from is CCAF5 during the most recent flights

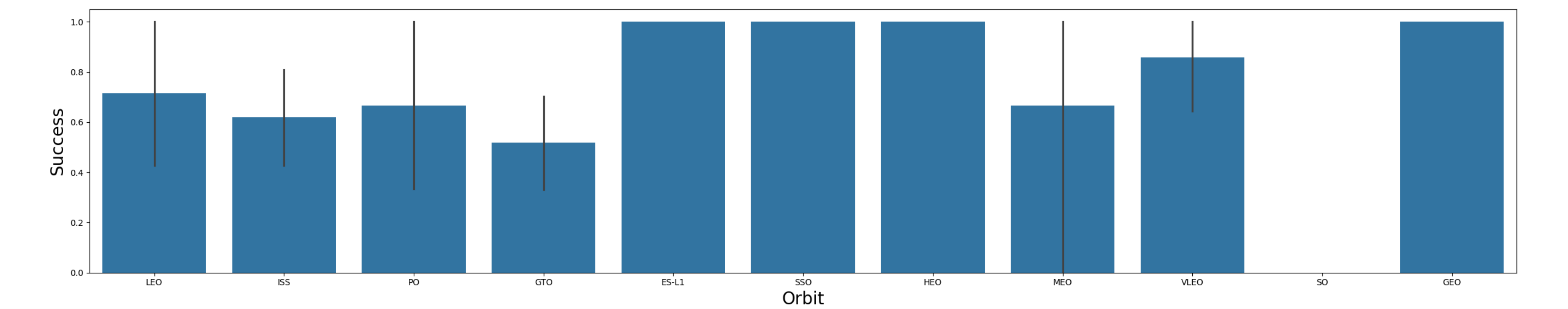
# Payload vs. Launch Site



Extremely heavy payloads have good chance of success

VAFB SLC 4E seems to have a weight limit of under 10 000 kg

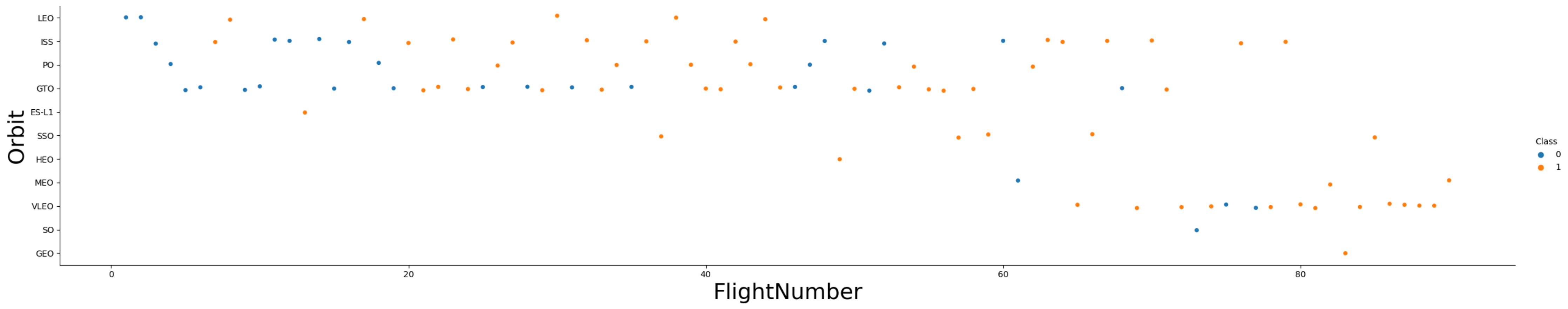
# Success Rate vs. Orbit Type



Orbits ES-L1, SSO, HEO and GEO have success rate of 100%

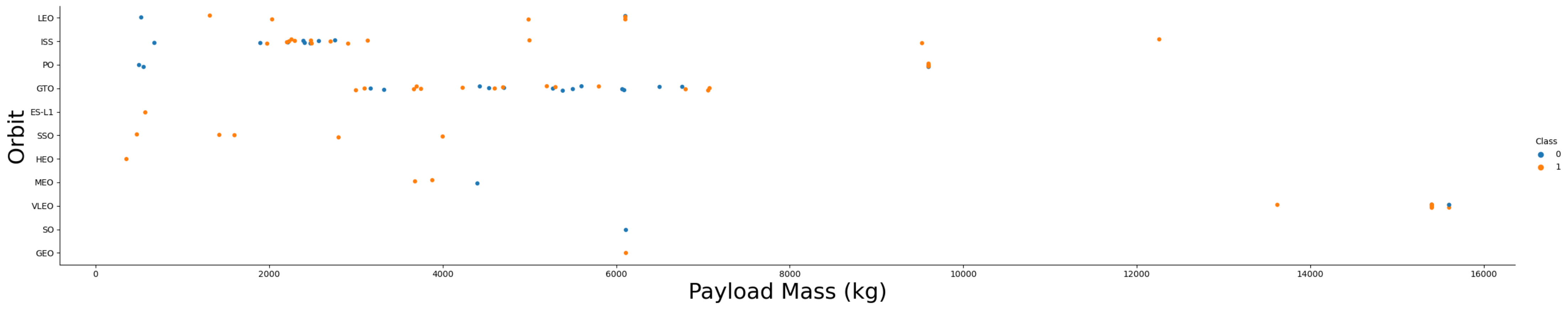
SO has a success rate of 0%

# Flight Number vs. Orbit Type



Success rate of all orbits increased over time

# Payload vs. Orbit Type

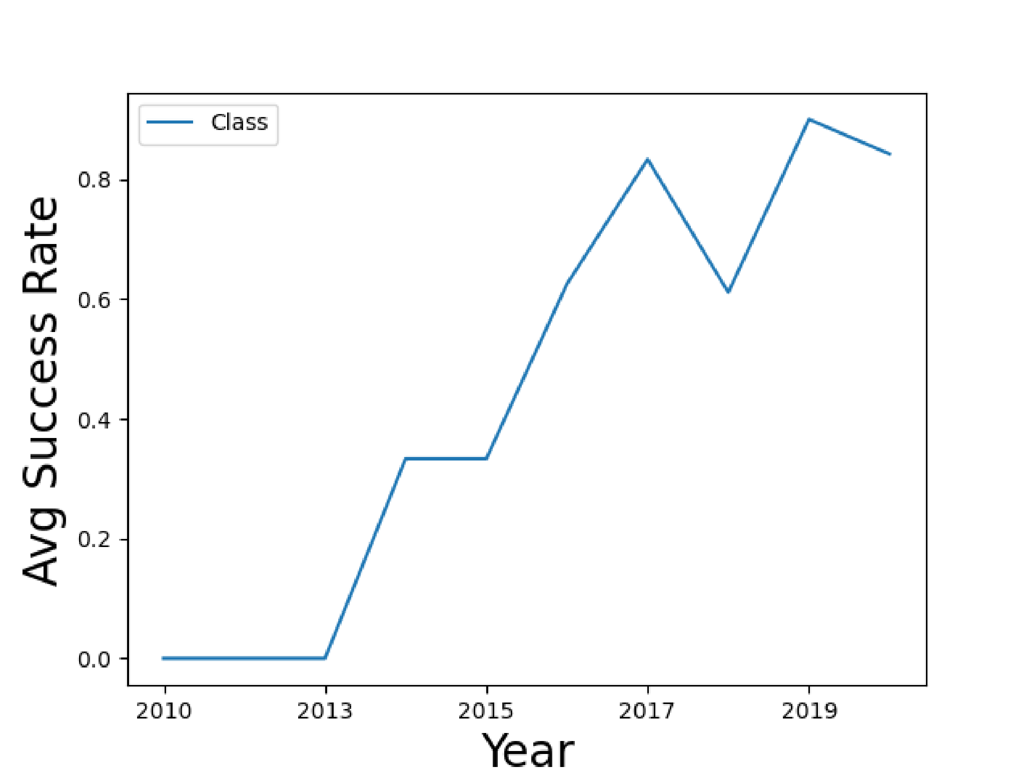


Payload mass and orbit varys wildly for GTO and ISS

# Launch Success Yearly Trend

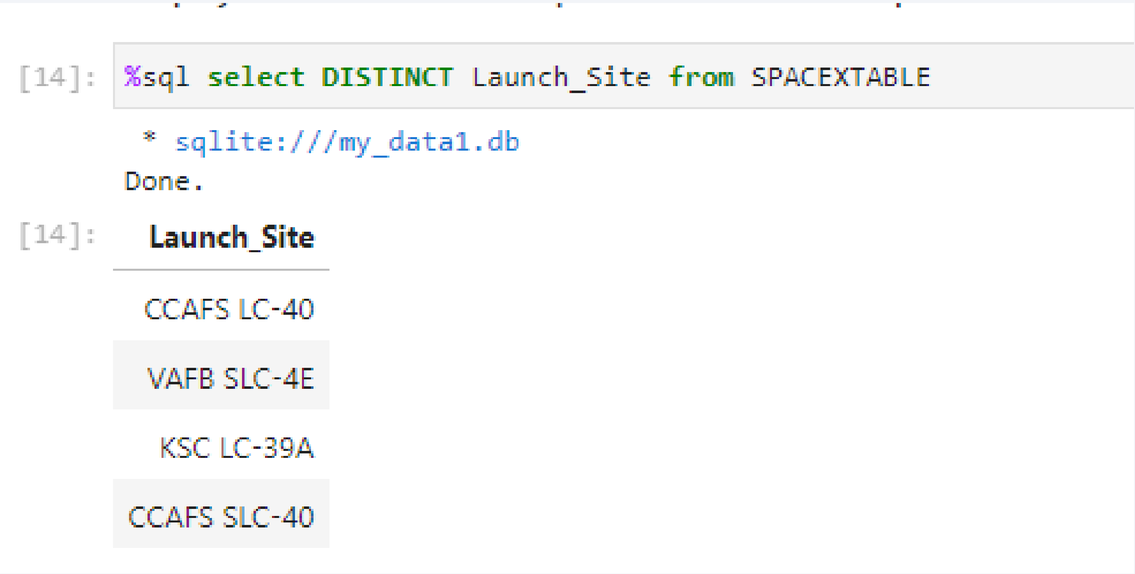
First three years had no successful launches, so to get in the business we must be persistent and not easily dissuaded

After 2013 success rate have been increasing



# All Launch Site Names

- Find the names of the unique launch sites
- There are 4 unique launch sites, although there are 2 similar launch sites that start with CCAFS



```
[14]: %sql select DISTINCT Launch_Site from SPACEXTABLE
```

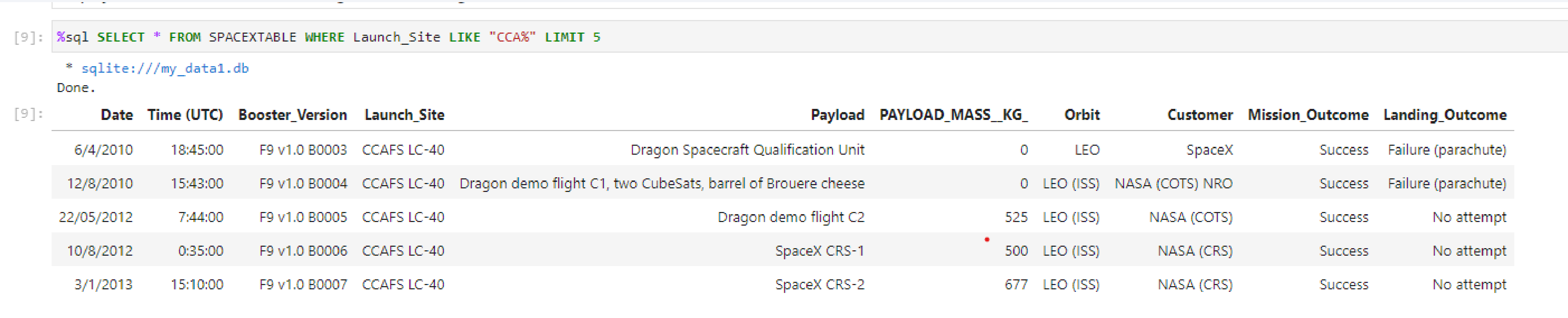
```
 * sqlite:///my_data1.db
Done.
```

[14]:

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

# Launch Site Names Begin with 'CCA'

Here is 5 sample launch site that begin with CCA



```
[9]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5
```

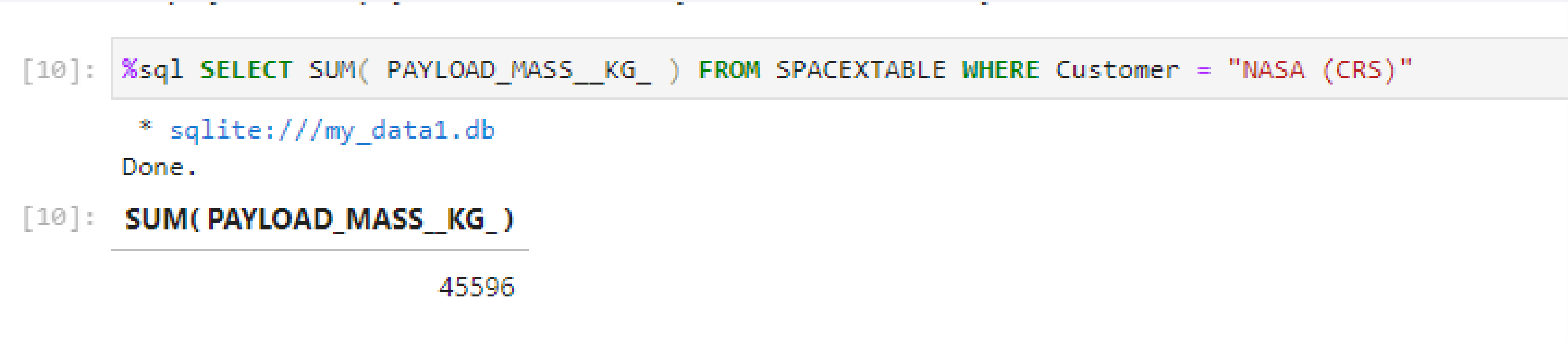
```
 * sqlite:///my_data1.db
Done.
```

[9]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 6/4/2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 12/8/2010 | 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) |
| 22/05/2012 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 10/8/2012 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 3/1/2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

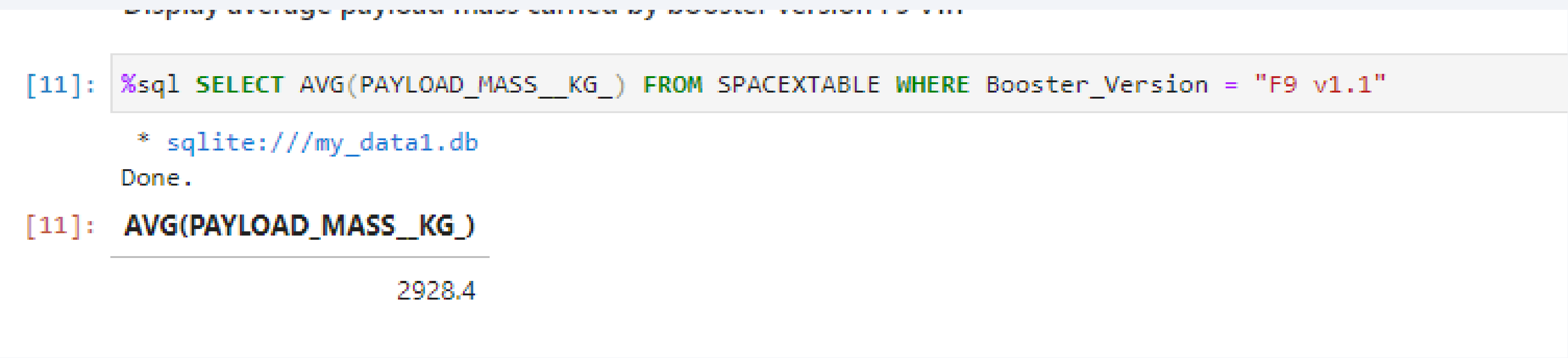
The total payload carried by boosters from NASA



```
[10]: %sql SELECT SUM( PAYLOAD_MASS__KG_ ) FROM SPACEXTABLE WHERE Customer = "NASA (CRS)"

 * sqlite:///my_data1.db
Done.

[10]: SUM( PAYLOAD_MASS__KG_ )
                      45596
```

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1



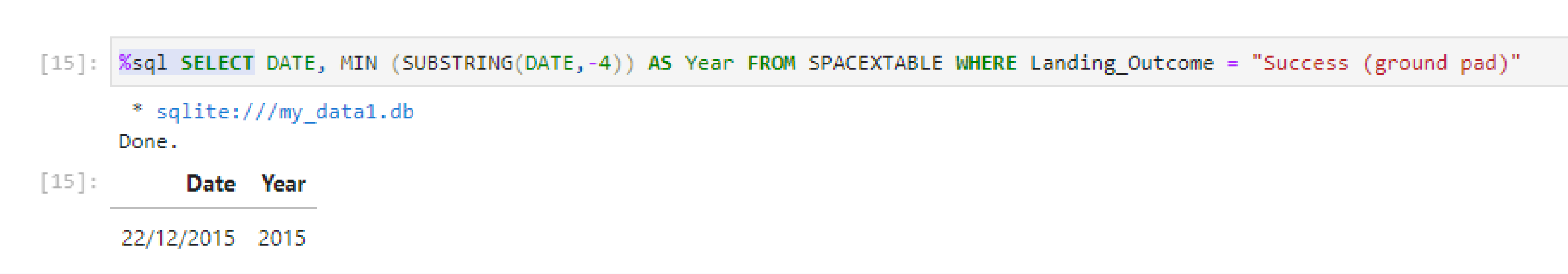
Display average payload mass carried by booster version F9 v1.1

```
[11]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = "F9 v1.1"

       * sqlite:///my_data1.db
      Done.

[11]:  AVG(PAYLOAD_MASS__KG_)
                       2928.4
```

Using this rocket the average mass and the toatl mass for NASA would estimate a total of 15 contracts, though more is possible with lighter weight

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



```
[15]: %sql SELECT DATE, MIN (SUBSTRING(DATE,-4)) AS Year FROM SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)"
```

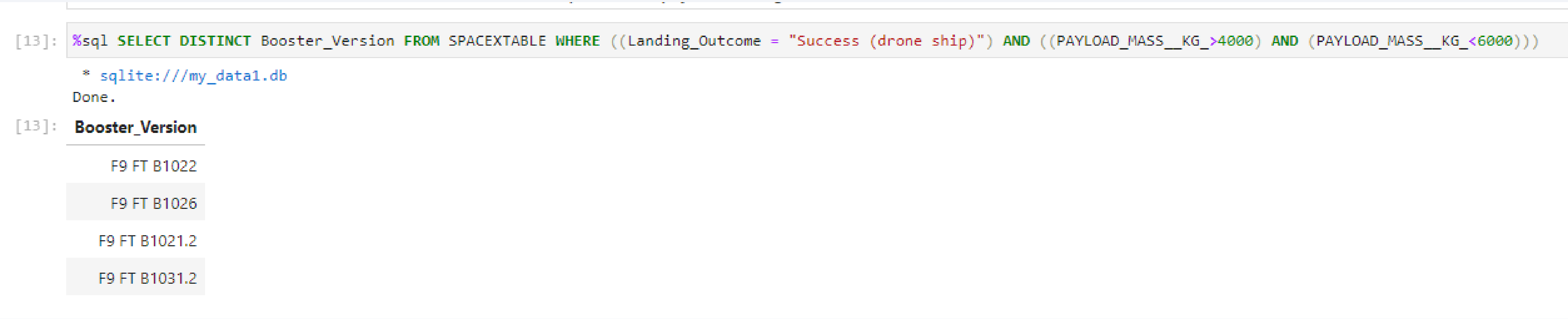
```
 * sqlite:///my_data1.db
Done.
```

[15]:

| Date | Year |
|---|---|
| 22/12/2015 | 2015 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



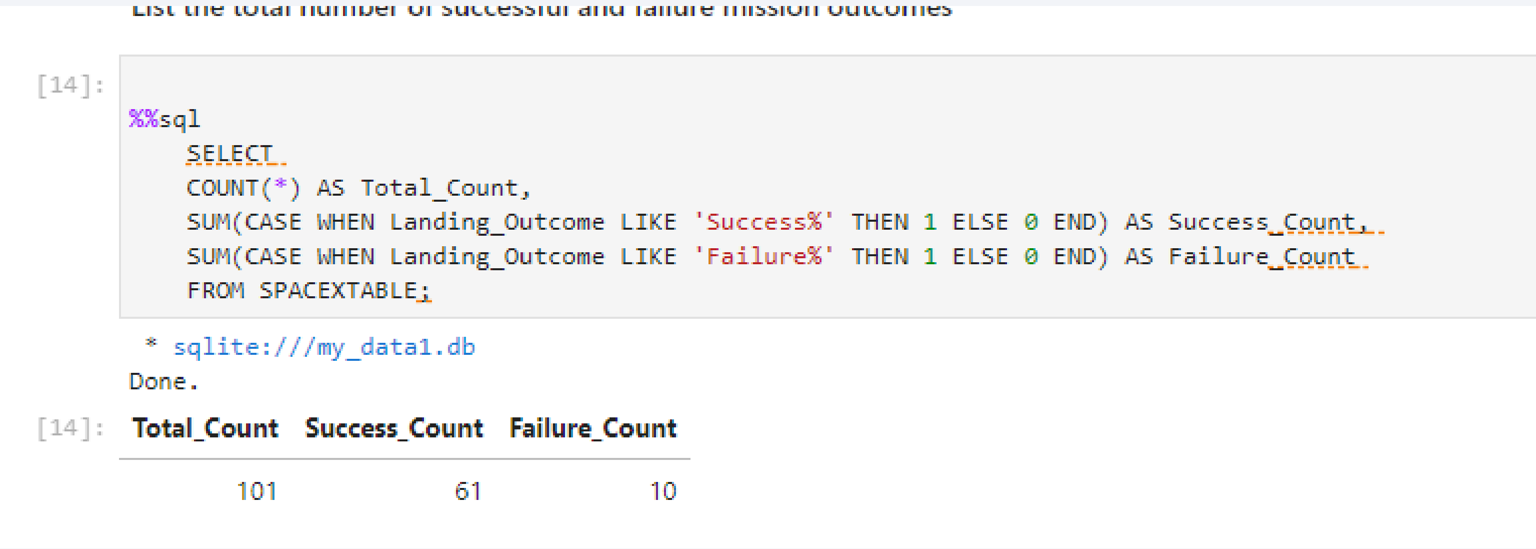
```
[13]: %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE ((Landing_Outcome = "Success (drone ship)") AND ((PAYLOAD_MASS__KG_>4000) AND (PAYLOAD_MASS__KG_<6000)))

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

[13]:

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

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- 60% success, 10% failure with 30% in the other category



List the total number of successful and failure mission outcomes

[14]:

```
%%sql
    SELECT
    COUNT(*) AS Total_Count,
    SUM(CASE WHEN Landing_Outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS Success_Count,
    SUM(CASE WHEN Landing_Outcome LIKE 'Failure%' THEN 1 ELSE 0 END) AS Failure_Count
    FROM SPACEXTABLE;
```

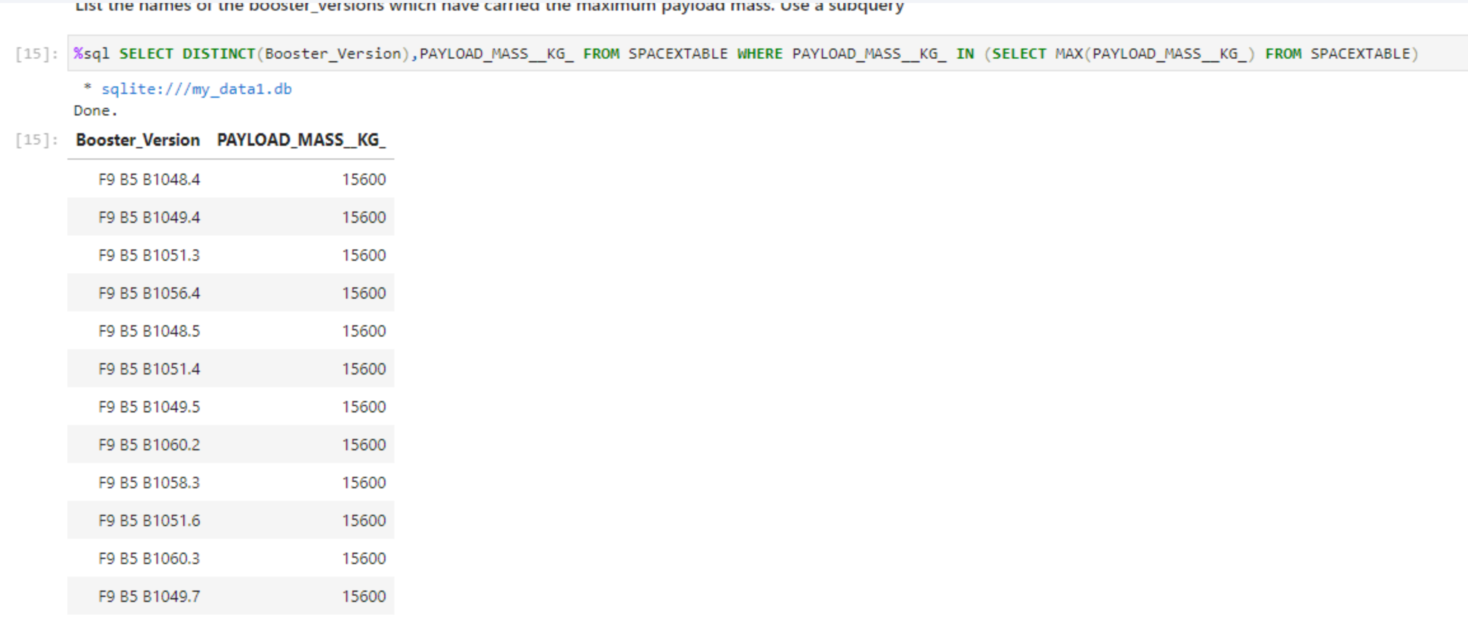
```
 * sqlite:///my_data1.db
Done.
```

[14]:

| Total_Count | Success_Count | Failure_Count |
|---|---|---|
| 101 | 61 | 10 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- To emulate SpaceX we would want to emulate the best performing rockets



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

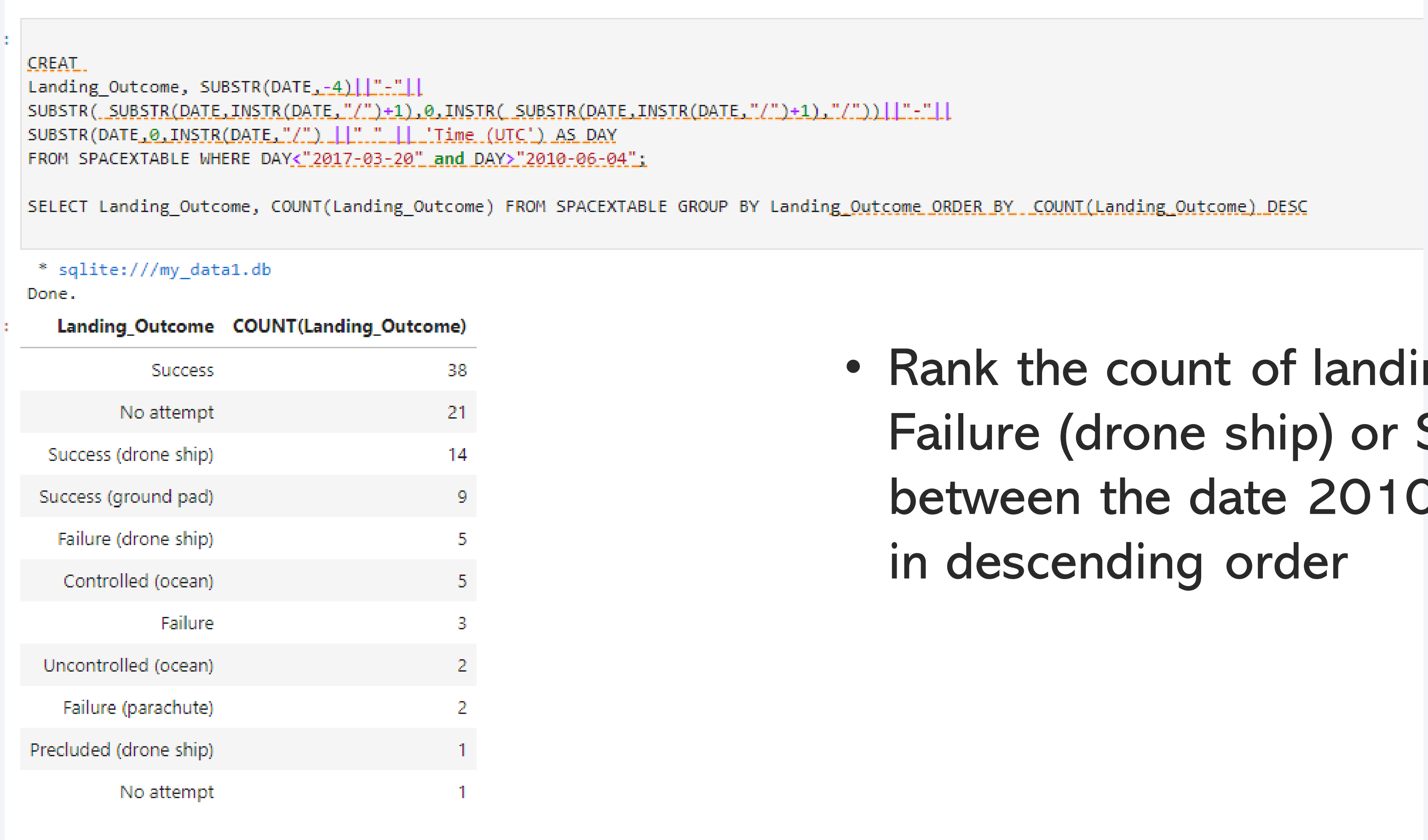
```
[15]: %sql SELECT DISTINCT(Booster_Version),PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

[15]:

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

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



```
CREAT
Landing_Outcome, SUBSTR(DATE,-4)||"-"||
SUBSTR( SUBSTR(DATE,INSTR(DATE,"/")+1),0,INSTR( SUBSTR(DATE,INSTR(DATE,"/")+1),"/"))||"-"||
SUBSTR(DATE,0,INSTR(DATE,"/") ||" " || 'Time (UTC') AS DAY
FROM SPACEXTABLE WHERE DAY<"2017-03-20" and DAY>"2010-06-04";

SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY  COUNT(Landing_Outcome) DESC
```

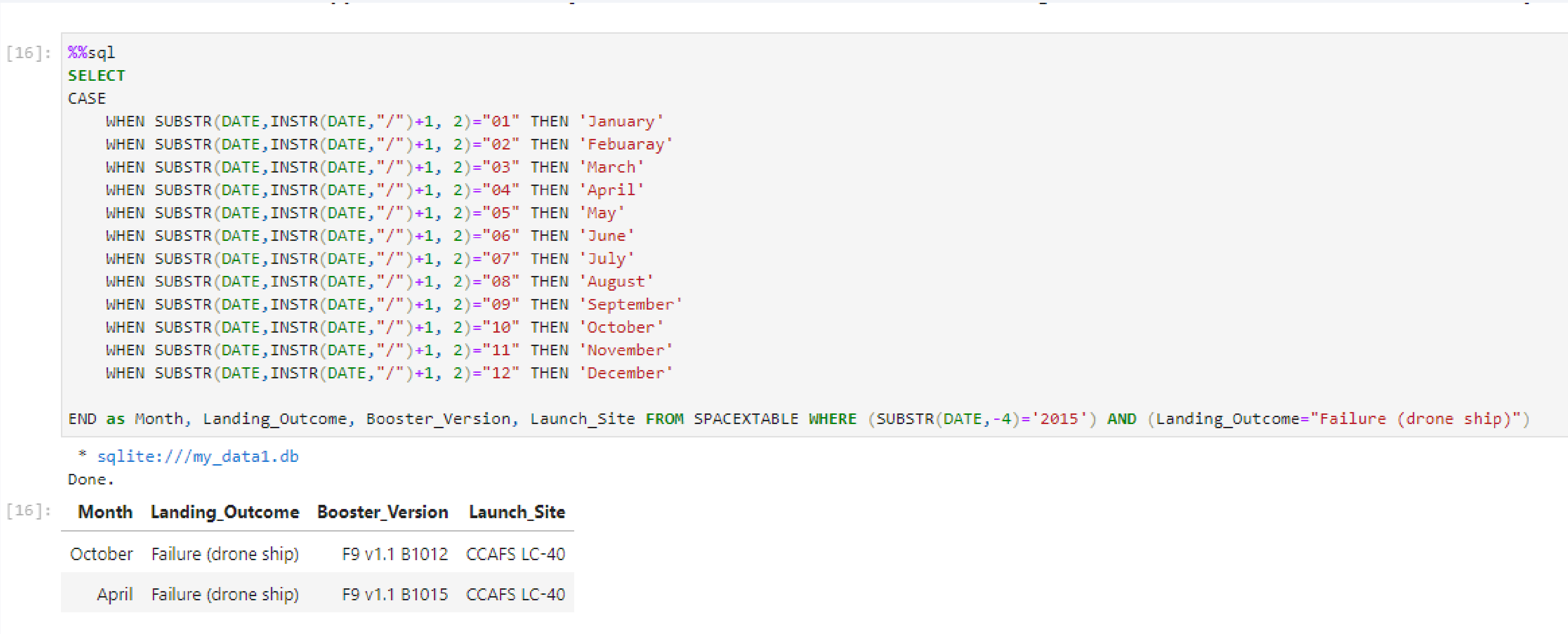
```
 * sqlite:///my_data1.db
Done.
```

| Landing_Outcome | COUNT(Landing_Outcome) |
|---:|---:|
| Success | 38 |
| No attempt | 21 |
| Success (drone ship) | 14 |
| Success (ground pad) | 9 |
| Failure (drone ship) | 5 |
| Controlled (ocean) | 5 |
| Failure | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |
| No attempt | 1 |

- 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

# 2015 Launch Records

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015



[16]:

```
%%sql
SELECT
CASE
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="01" THEN 'January'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="02" THEN 'Febuaray'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="03" THEN 'March'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="04" THEN 'April'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="05" THEN 'May'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="06" THEN 'June'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="07" THEN 'July'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="08" THEN 'August'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="09" THEN 'September'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="10" THEN 'October'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="11" THEN 'November'
    WHEN SUBSTR(DATE,INSTR(DATE,"/")+1, 2)="12" THEN 'December'

END as Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE (SUBSTR(DATE,-4)='2015') AND (Landing_Outcome="Failure (drone ship)")
```

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

[16]:

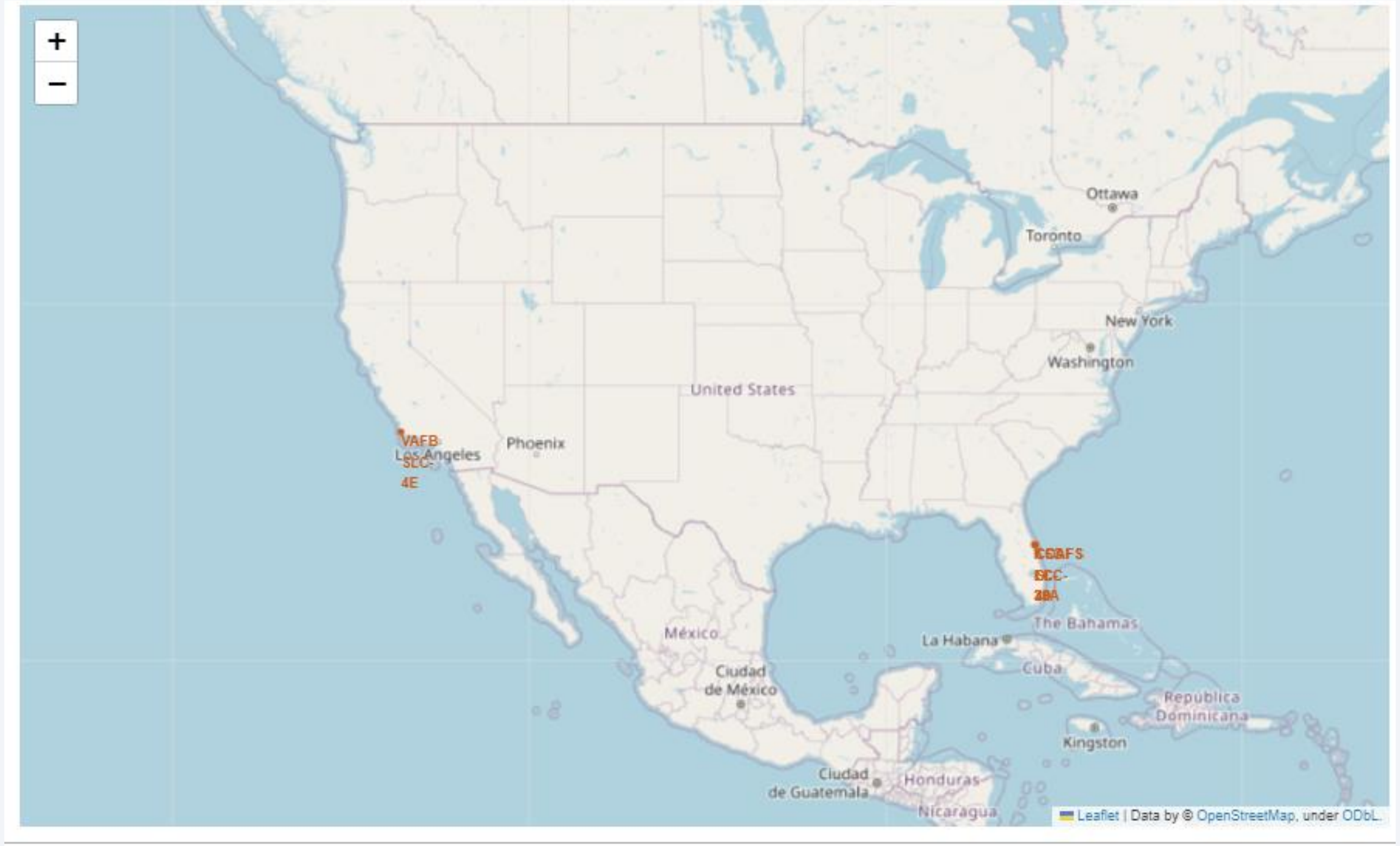
| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| October | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

Section 3

# Launch Sites Proximities Analysis

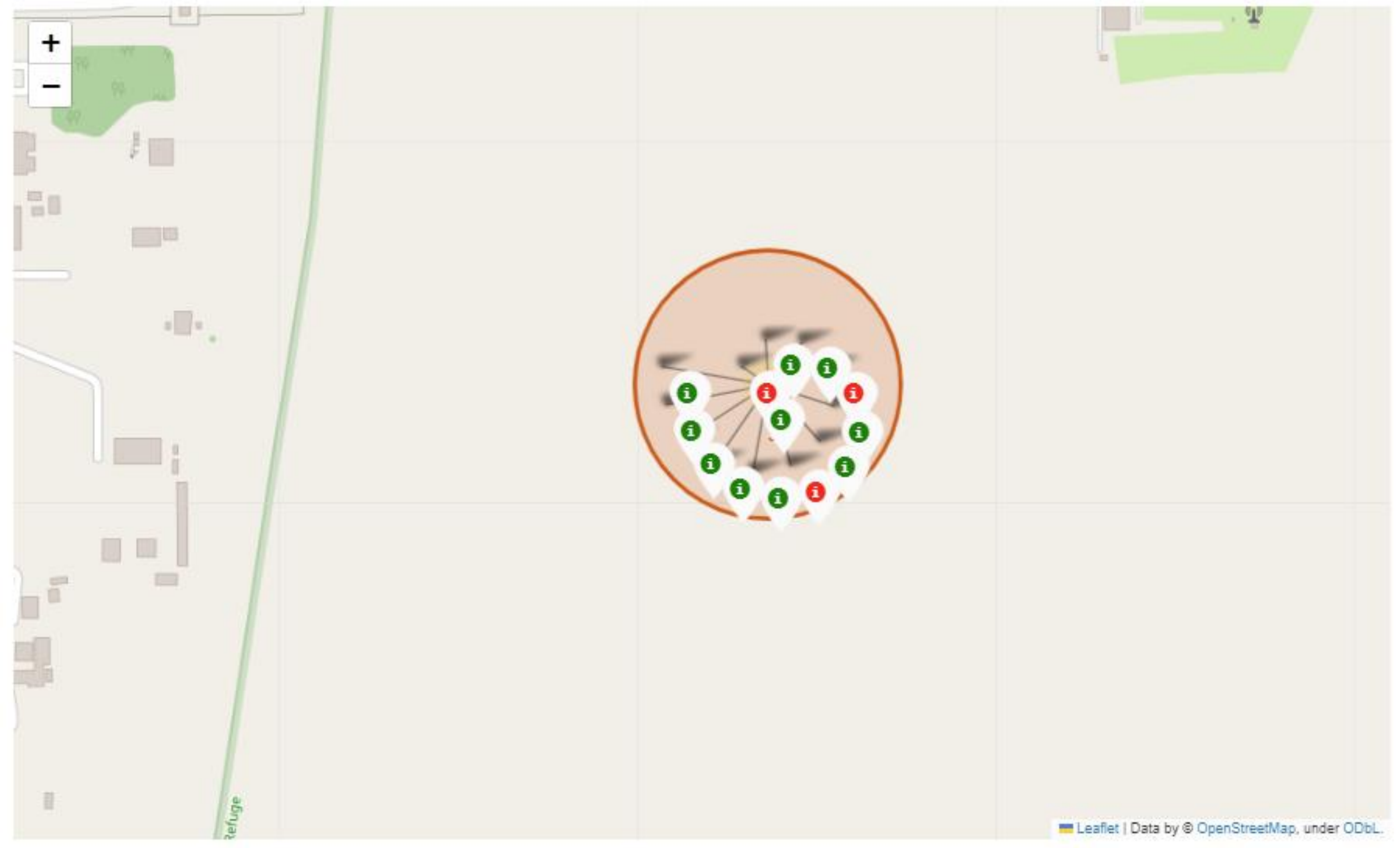
# Folium Map of Launch Site locations

- Launch Site Locations



# Folium Map of Success Markers

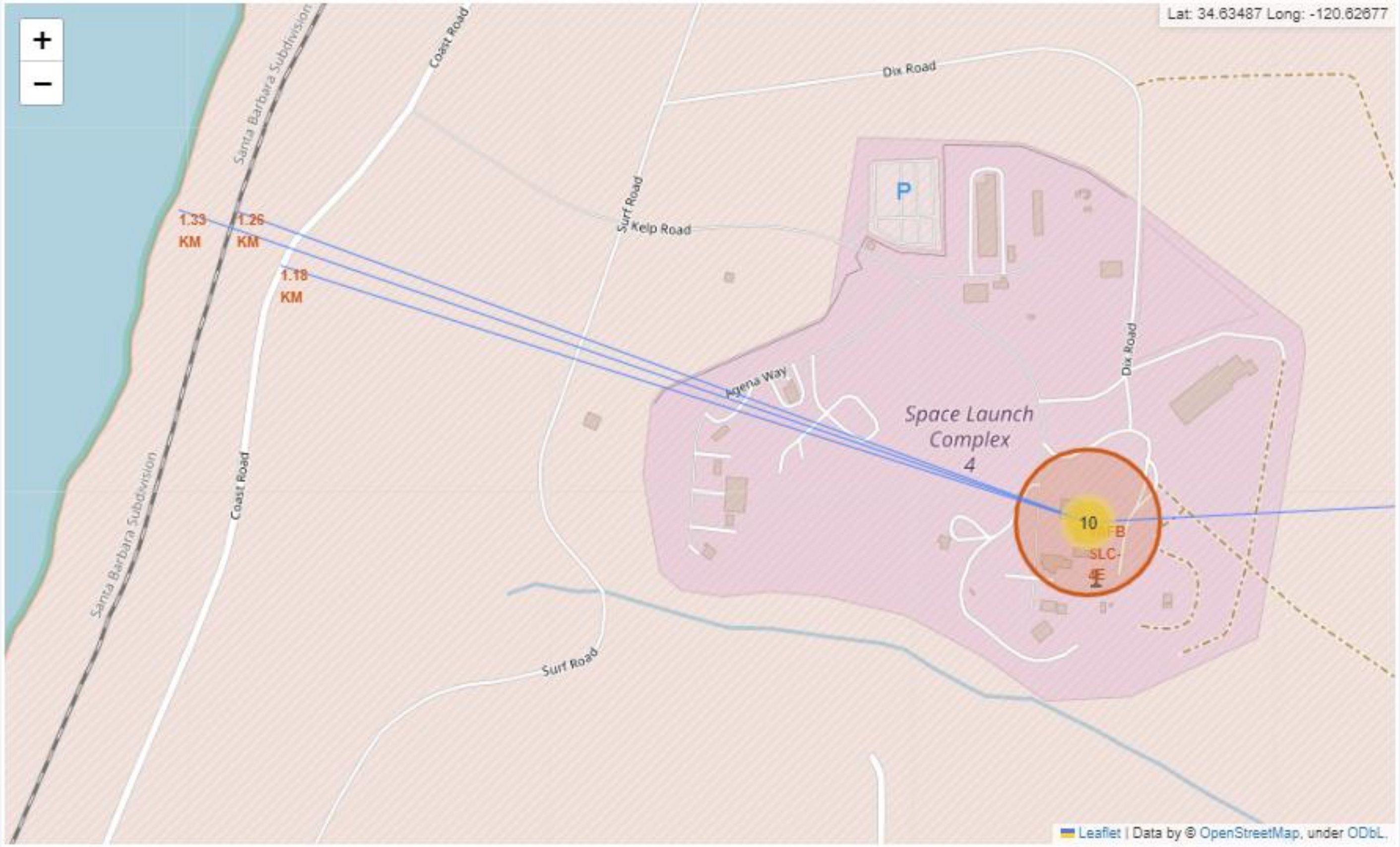
- Launch Site Location Success Markers

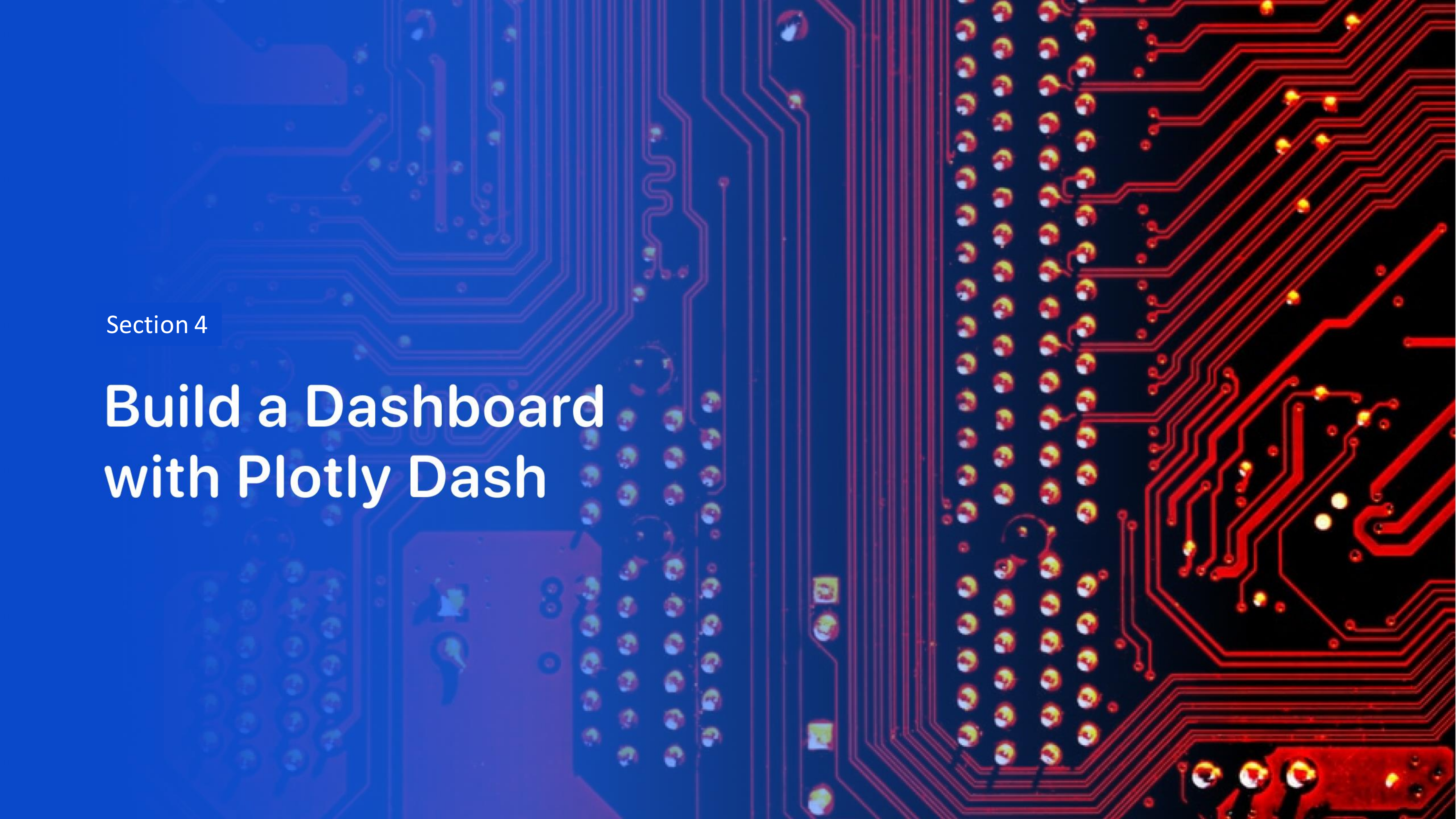


# Folium Map Proximities

- Generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- These are important factors when considering where to build a launch site



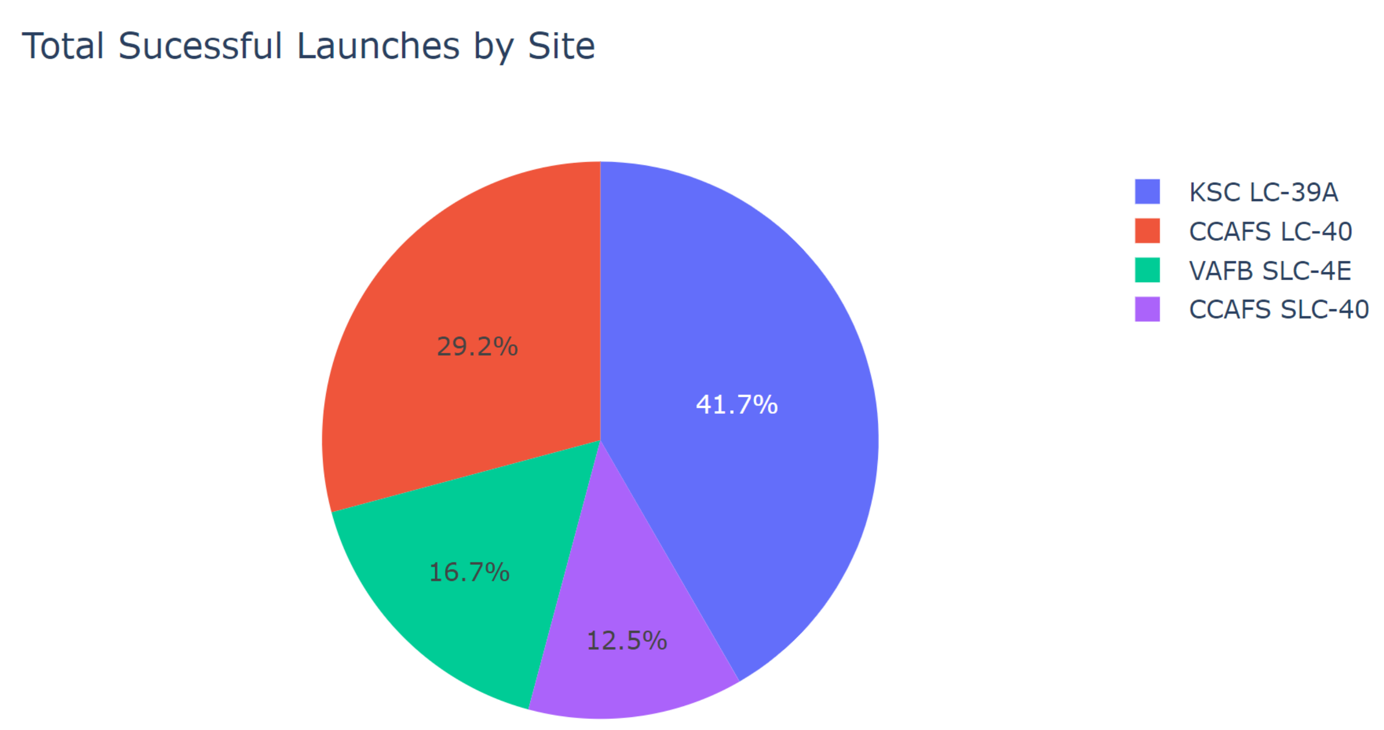


Section 4

# Build a Dashboard with Plotly Dash

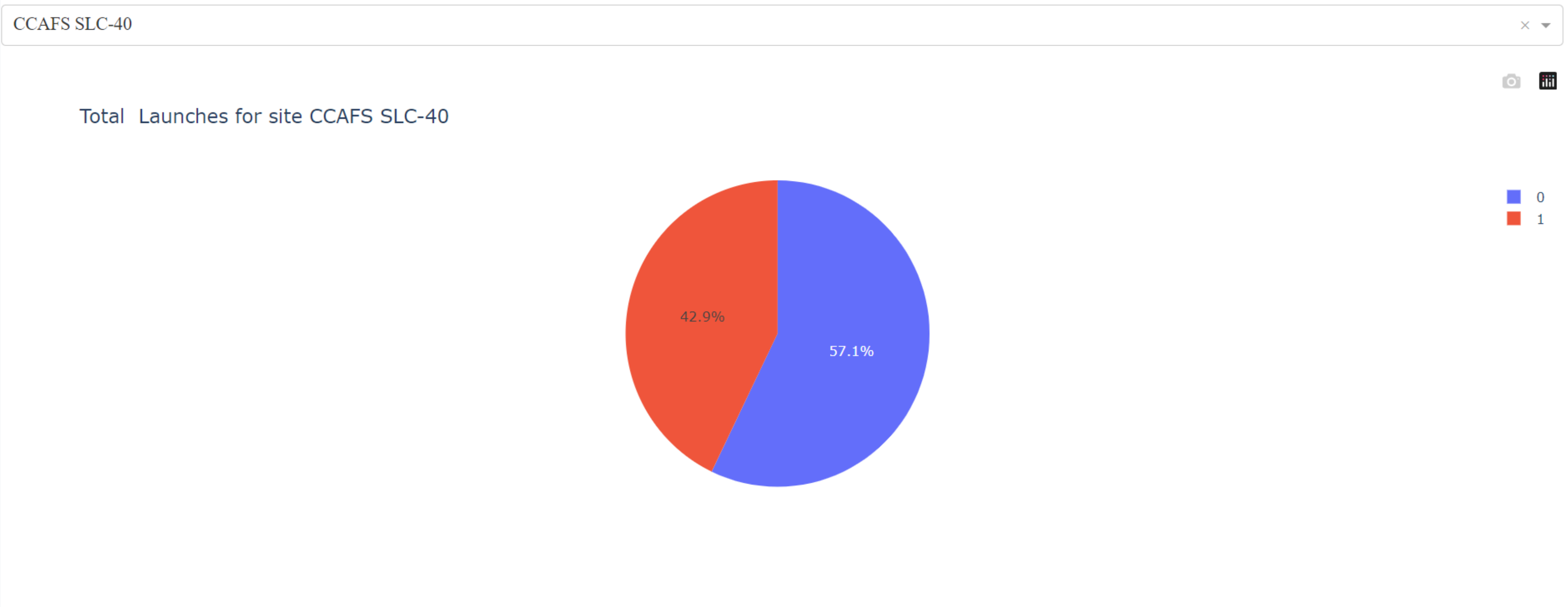
# Dash Launch Success for all sites

- Launch success count for all sites, in a piechart
- KSC LC-39A has the most sucesses



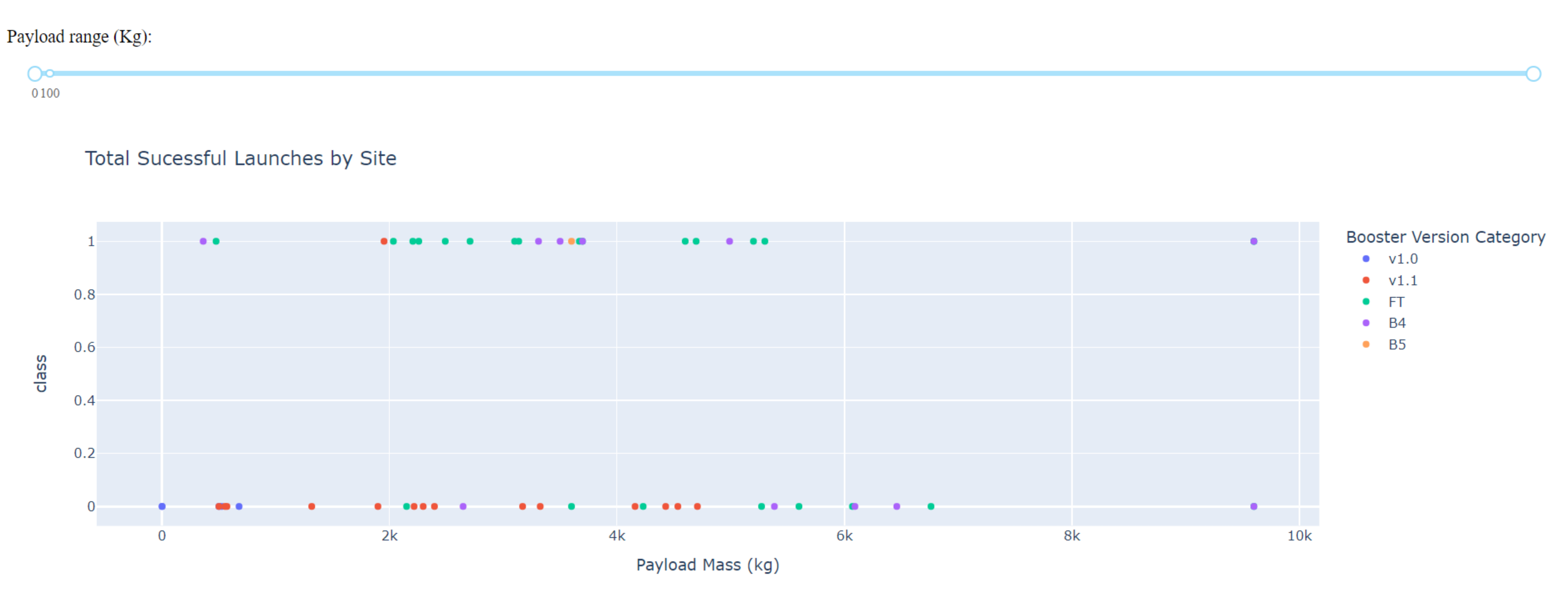
# Dashboard of Success Rate by Site

- Pie chart for the launch site with highest launch success ratio
- CCAFS SLC-40 has the highest success rate with 57.1%

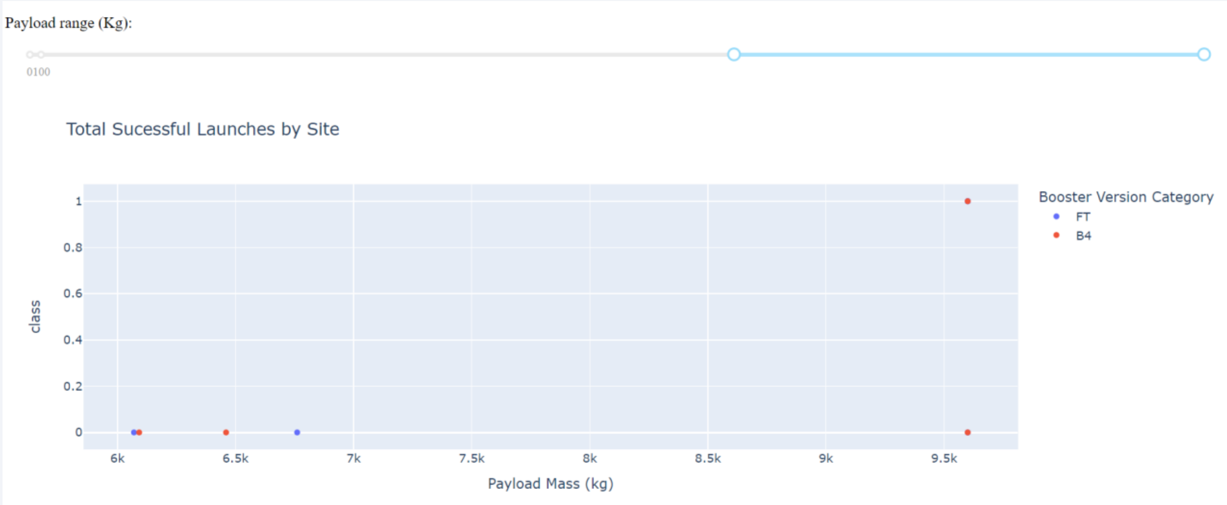


CCAFS SLC-40

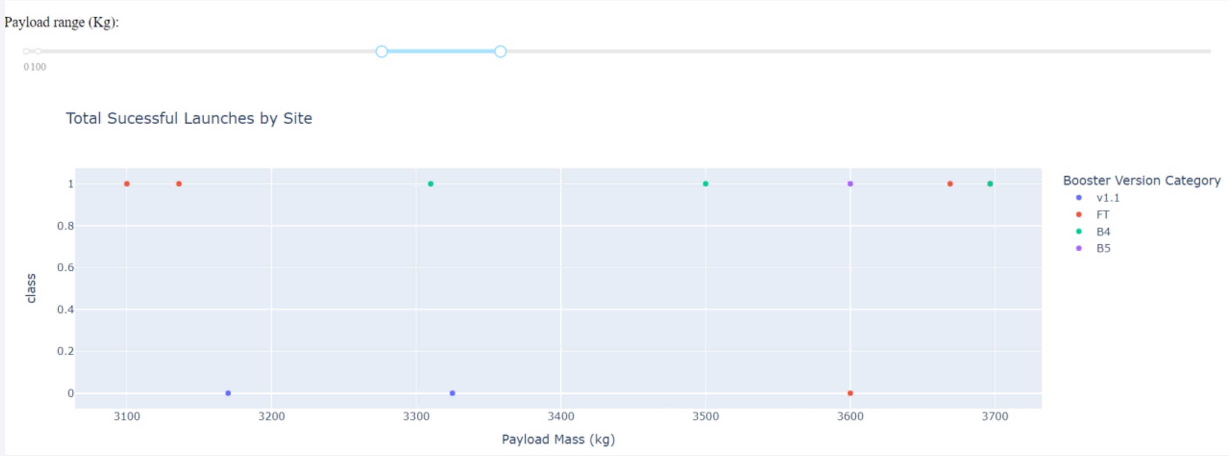
# Dashboard of Payload Mass for success rate



# <Dashboard Screenshot 3>



The high end rannge of mass above 6k  seems to have to most failure, though stacking of points might come into effect



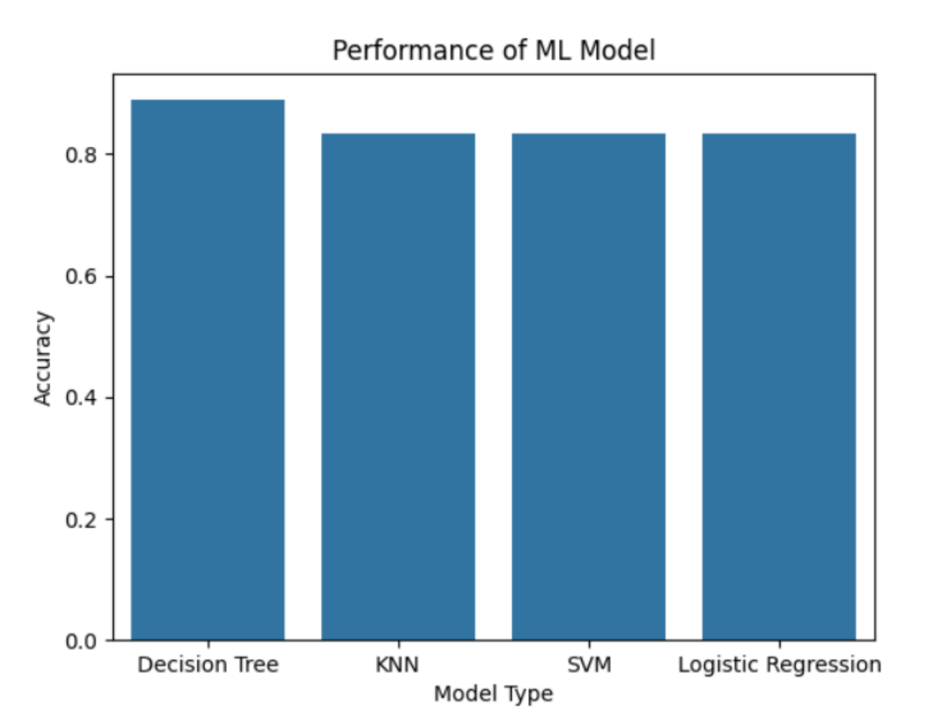
The mass range of 3000, to 4000 kg has a high success rate

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

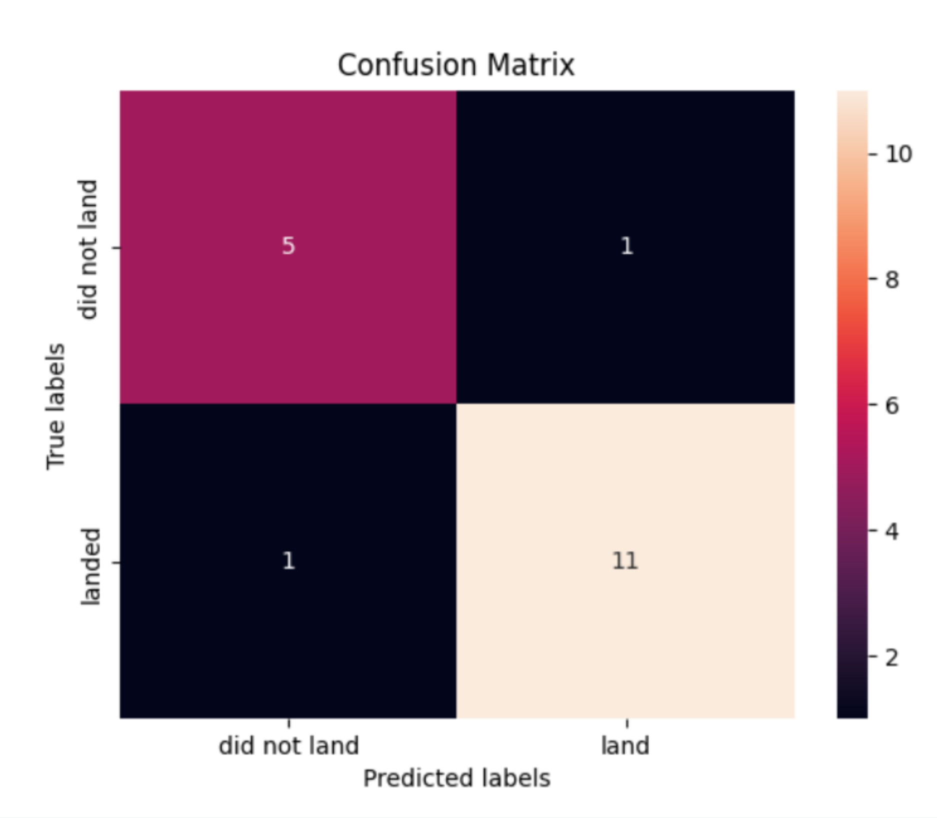
- The highest classification accuracy is the Decision Tree with 88.87% accurarcy



# Confusion Matrix

- The Decision Tree's confusion matrix

- It correctly classifies 16 out of 18
    - With 2 errors.



# Conclusions

- SpaceX and Wikipedia were mined for data
- Launches near proxmities are best
- It takes time to develop a successful space laucnh program
- Decision tree classifier can be used to successfully predict landings

# Appendix

GitHub Repo of all notebooks and code snipets

# Thank you!