IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

<Bruce Jeffry Stark>
<April 29, 2024>

# Outline

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

# Executive Summary

Methods Summary:

- Launch data for analysis was collected from SpaceX using the API they provide and from a Wikipedia page about SpaceX launches.
- The data was cleaned and put into a format suitable for exploratory data analysis ('EDA') and machine learning.
- EDA was performed through visualization (seaborn, Folium and Plotly dashboard) and queries via SQL.
- Machine learning models were trained and evaluated using scikit-learn, specifically GridSearch.

Results Summary:

- SpaceX has improved landing success rate over the years.
- Heavier payloads have a lower success rate than lighter ones, with the best success rate being in the payload range of 3000 to 4000 kg.
- Launches into ES-L1, GEO, HEO and SSO have a higher success rate than other orbits.
- The best machine learning model produced was a Decision Tree with 87.5% accuracy.

# Introduction

- Project background and context

  The current leader in commercial space flight is SpaceX. It has this leadership position in great part due to its ability to cut costs, primarily by reusing the first stage of its rocket. This allows SpaceX to launch payloads for $62 million instead of the $165 million charged by its competitors.

- Project goal

  However, SpaceX can only reuse the first stage if it lands successfully, and that does not always happen. If a competitor could find the parameters that predict a successful landing with high enough accuracy it could gain a competitive advantage. The goal of this project is to analyze data on SpaceX launches to discover these parameters and to build a model based on them that can accurately predict whether a launch will land successfully.

Section 1

# Methodology

# Methodology

## Summary

1. Data collection
2. Data wrangling
3. Exploratory data analysis (EDA) using visualization and SQL
4. Interactive visual analytics using Folium and Plotly Dash
5. Predictive analysis using classification models

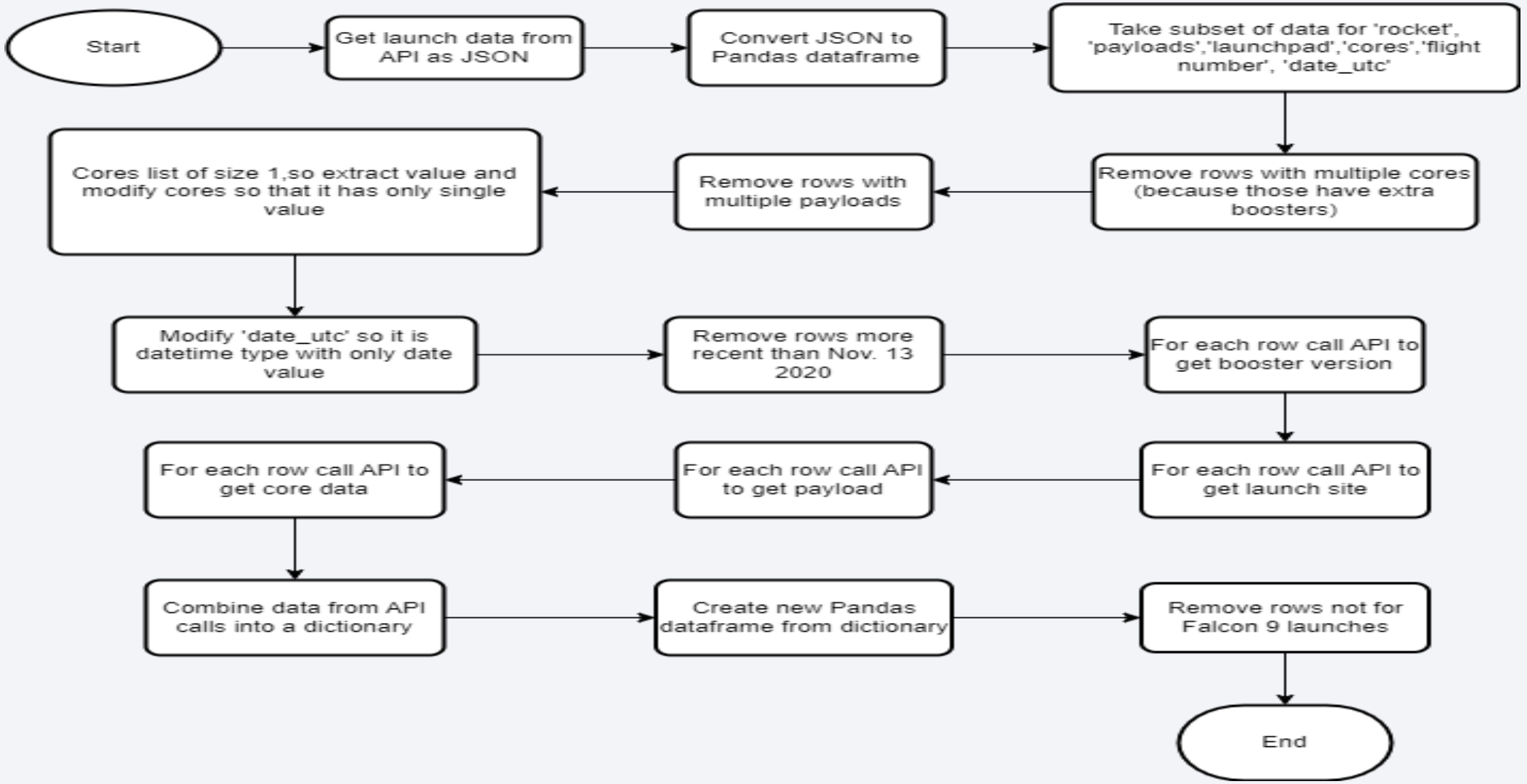
# Data Collection

- Data was collected from 2 sources:
  - SpaceX REST API
  - Web scrapping the Wikipedia page ‘List of Falcon 9 Heavy Launches’

# Data Collection – SpaceX API

- REST API calls:
  - Launch data - https://api.spacexdata.com/v4/launches/past
  - Booster version - https://api.spacexdata.com/v4/rockets/
  - Launch pad sites - https://api.spacexdata.com/v4/launchpads/
  - Payload data - https://api.spacexdata.com/v4/payloads/
  - Core data - https://api.spacexdata.com/v4/cores/
- Notebook URL: https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection – SpaceX API



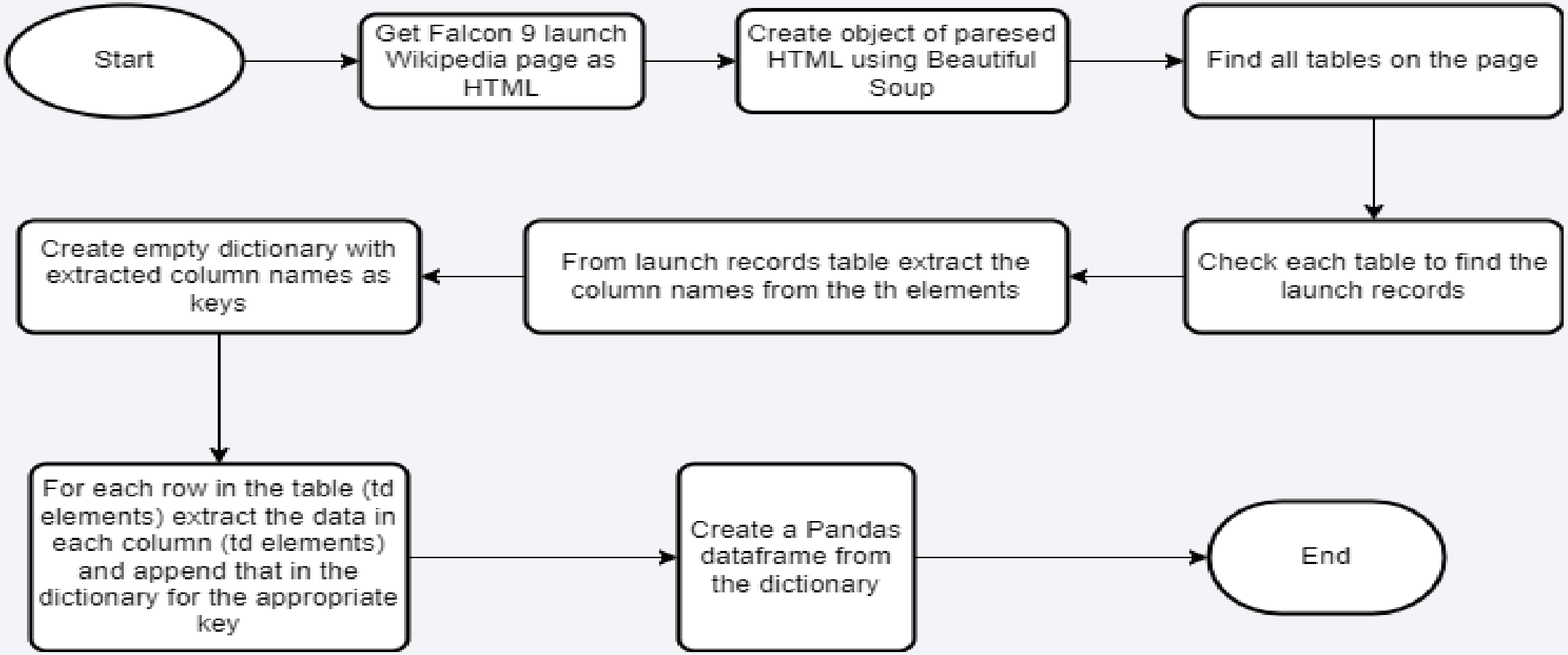
# Data Collection – Scraping: Procedure

Steps:

1. Retrieve the Wikipedia page using get function from requests module.
2. Parse the page using BeautifulSoup.
3. Collect all the tables on the page.
4. Looking at the tables on the page identify the one that has the data we are looking for, which in this case is the third table.
5. From the table extract the column names
6. Create a dictionary with the extracted column names as keys
7. For each row in the table get the data for each column and add it to the dictionary
8. Create a Pandas dataframe from the dictionary

Notebook URL: https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/jupyter-labs-webscraping.ipynb

# Data Collection – Scraping: Flowchart



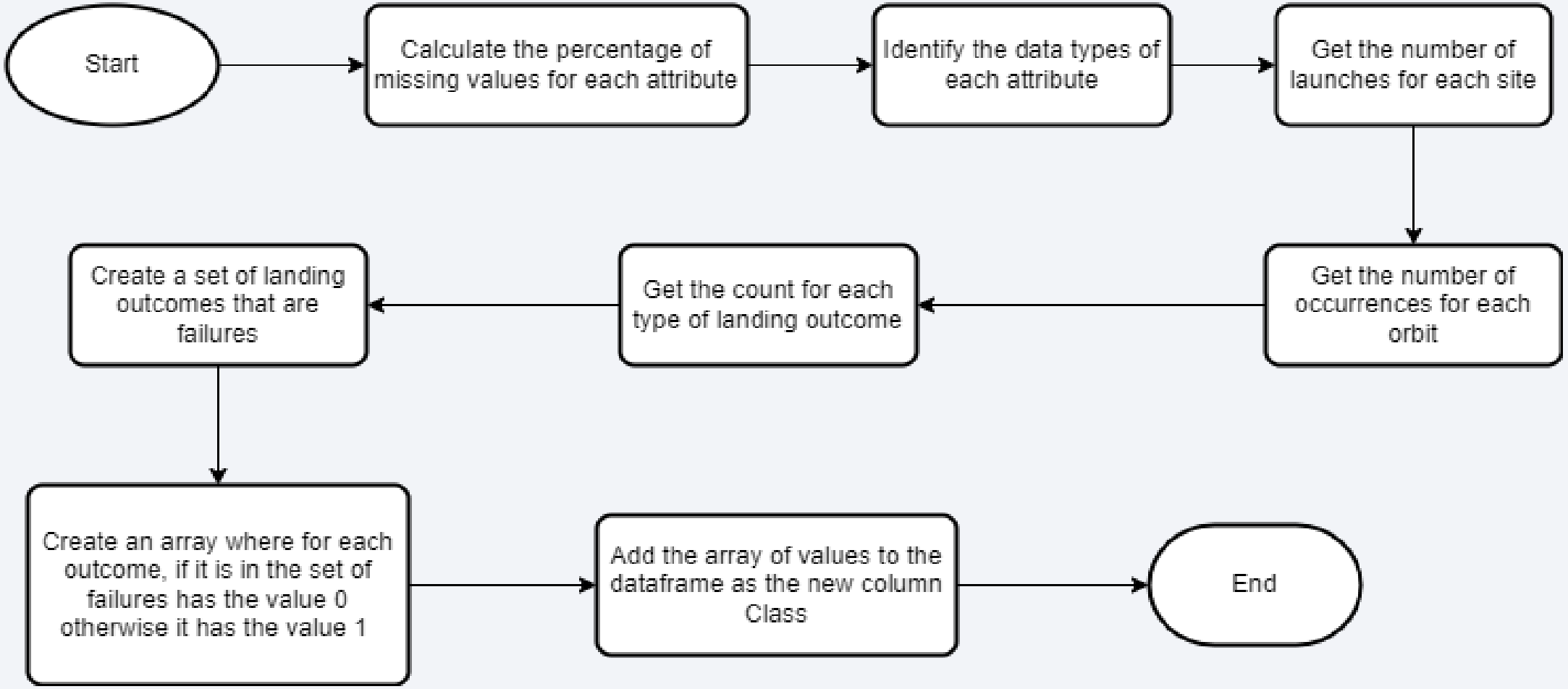
# Data Wrangling

Process:

1. The percentage of missing values was calculated for each column
2. The data types for each column were identified
3. The number of launches for each launch site were determined
4. The number of occurrences for each type of orbit was determined
5. The number for each type of outcome was determined
6. A new label called Class was created from the data in the Outcome column such that the value 0 represents a failed outcome and the value 1 represents a successful outcome
7. From the new Class column the average success rate was calculated as 66.6% or roughly 2/3

Notebook URL:
https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/labs-jupyter-spacex-Data%20wrangling.ipynb

# Data Wrangling



# EDA with Data Visualization

- Scatter plot of FlightNumber vs PayloadMass. Later flights more successful than earlier flights. More massive payloads less successful then lighter payloads.
- Scatter plot of FlightNumber vs LaunchSite. Earlier flights for all launch sites less successful than later flights.
- Scatter plot of PayloadMass vs LaunchSite. The VAFB-SLC launch site had no launches with a heavy payload (greater than 10000 kg).
- Bar chart of Orbit type and Class. The ES-L1, GEO, HEO and SSO orbits have the highest success rates.

# EDA with Data Visualization

- Scatter plot of FlightNumber vs Orbit. For launches into LEO orbits, earlier flights less successful than later flights. No apparent relationship for other orbits.
- Scatter plot of Payload vs Orbit. The LEO, ISS and Polar orbits have a higher success rates for heavy payloads than other orbit types.
- Line chart of average success rate per year. The success rate has steadily increased over the years since 2013.
- Notebook URL: https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/edadataviz.ipynb

# EDA with SQL

- List the unique names of the launch sites: select distinct "Launch_Site" from SPACEXTABLE;
- List 5 records where launch sites begin wth the string 'CCA': select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5;
- Display total payload mass carried by boosters launched by NASA: select "Customer", sum("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Customer" like "NASA%";
- Display average payload mass carried by booster version F9 v1.1: select avg("PAYLOAD_MASS__KG_") as average_mass from SPACEXTABLE where "Booster_Version" = "F9 v1.1";

# EDA with SQL

- List the date for the first successful landing outcome on the ground pad: select * from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)" order by Date asc limit 1;
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000: select distinct "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000;
- List the total number of successful and failure mission outcomes: select "Mission_Outcome", count(*) from SPACEXTABLE group by "Mission_Outcome";

# EDA with SQL

- List the names of the booster_versions which have carried the maximum payload mass (use a subquery): select "Booster_Version" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE);
- 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: select substr("Date",6,2), "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where "Landing_Outcome" = "Failure (drone ship)" and substr("Date",0,5) = '2015';

# EDA with SQL

- 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: select "Landing_Outcome", count(*) as Outcome_Count from SPACEXTABLE where "Date" between '2010-06-04' and '2017-03-20' group by "Landing_Outcome" order by "Outcome_Count" desc;
- Notebook URL: https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

- Created a Folium map object
- For each launch site added a circle and popup label so that users can easily identify them on the map.
- Also added markers for each launch site so that they can be identified on the map without having to click on the circle to get the popup label.
- For each site created a marker cluster for successful (green) and failed (red) launches. When clicking on the cluster the map displays all the markers making it easy for users to see them.
- Added a mouse position object to the map so that users could see the coordinates of the location their mouse was hovering over.
- For a selected site, CCAFS SLC40, added markers for the closest coastline, railway, highway and city. Using those markers added lines labeled with the distance between the launch site and the marked areas so that users could visually see how far the marked areas are from the launch site.
- Notebook URL (best viewed using nbviewer): https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/lab_jupyter_launch_site_location.ipynb

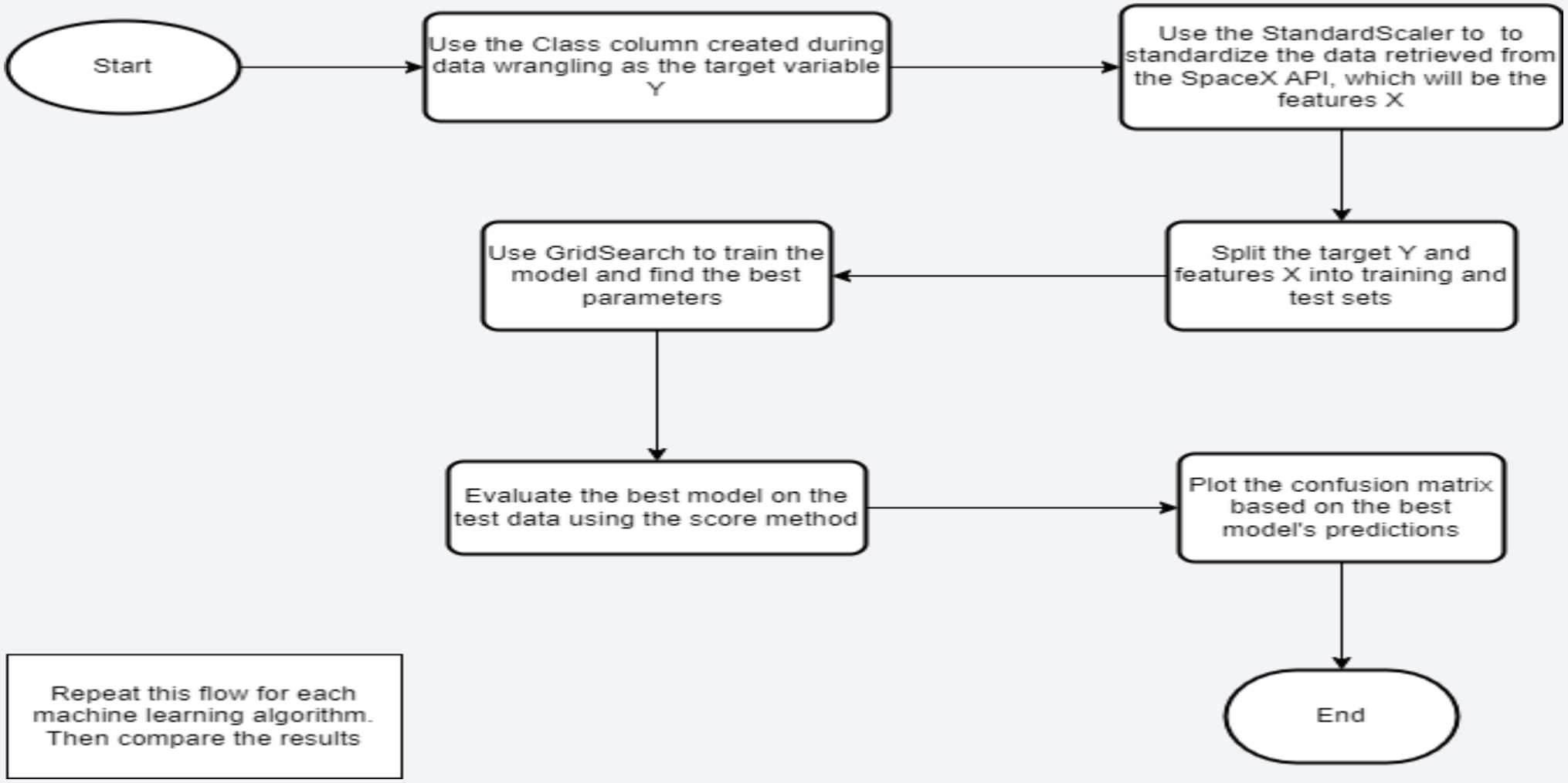
# Build a Dashboard with Plotly Dash

- Want to interactively display the launch success rate for different sites.
  - Added a drop down menu with options for each site and for all sites aggregated.
  - Added a pie chart display that shows the success rate based on the option selected.
- Want to interactively display the relationship between payload range and success.
  - Added a payload range slider
  - Added a scatter plot display where the x-axis is the payload range and the y-axis is mission outcome.
  - The scatter plot includes a color label for the version of the booster used in the mission, in order to display mission outcomes for different boosters.
- GitHub URL for dashboard application code: https://github.com/jstark997/IBMDataScienceCapstone/blob/main/spacex_dash_app.py

# Predictive Analysis (Classification)

- Created an array for the target variable Y from the Class column of the data.
- Standardized the data in the set of features X
- Split X into training and test data
- For each of the machine learning algorithms - Logistic Regression, SVM, Decision Tree and KNN:
  - Use Grid Search to train the model and find the best performing parameters
  - Evaluate the best model on the test data using the score method.
  - Plot the confusion matrix based on the best model's predictions to further evaluate the model
- Notebook URL: https://github.com/jstark997/IBMDataScienceCapstone/blob/main/Notebooks/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Predictive Analysis (Classification)



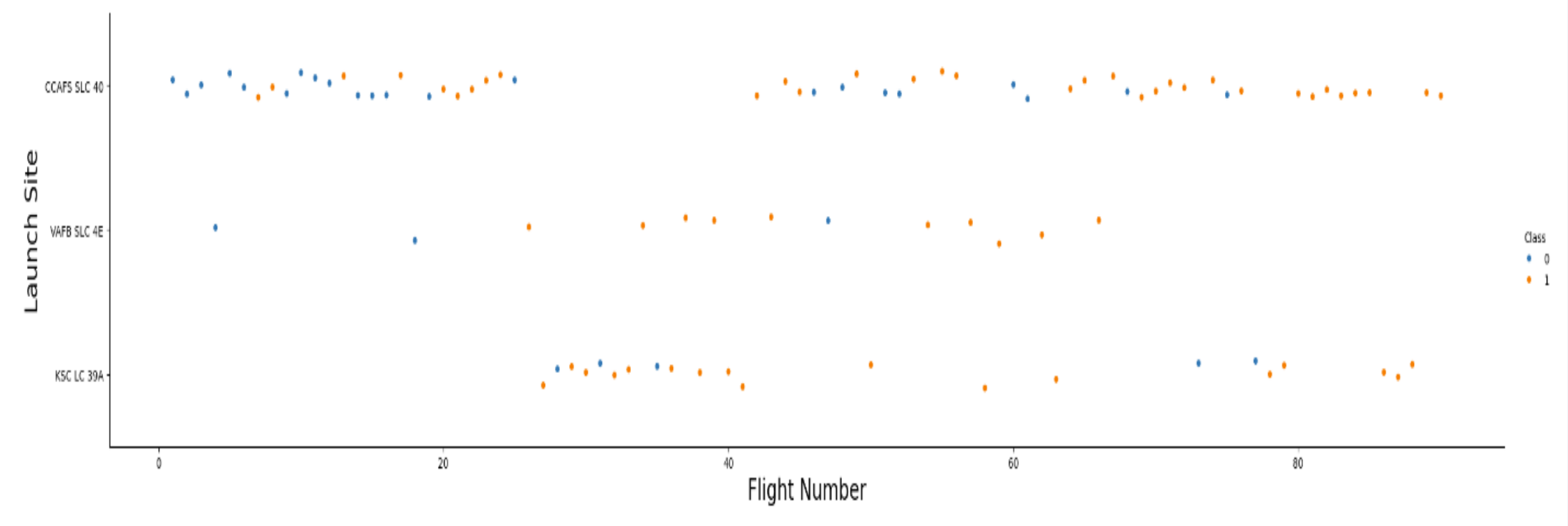
# Results

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

Section 2

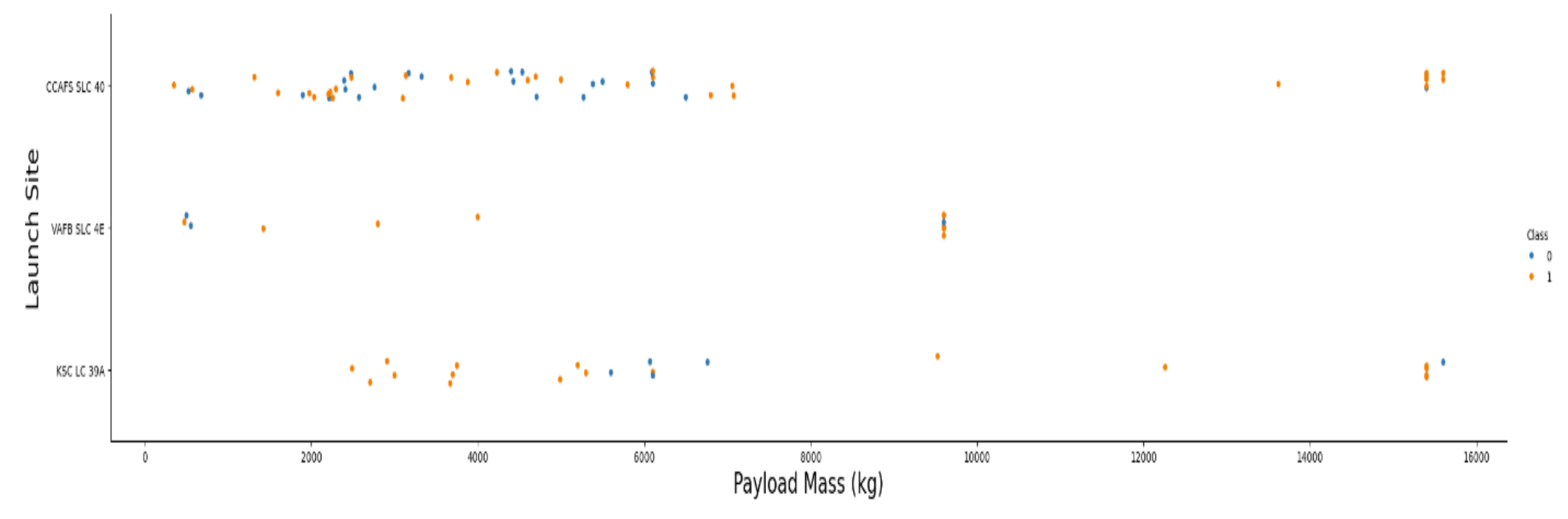
# Insights drawn from EDA

# Flight Number vs. Launch Site



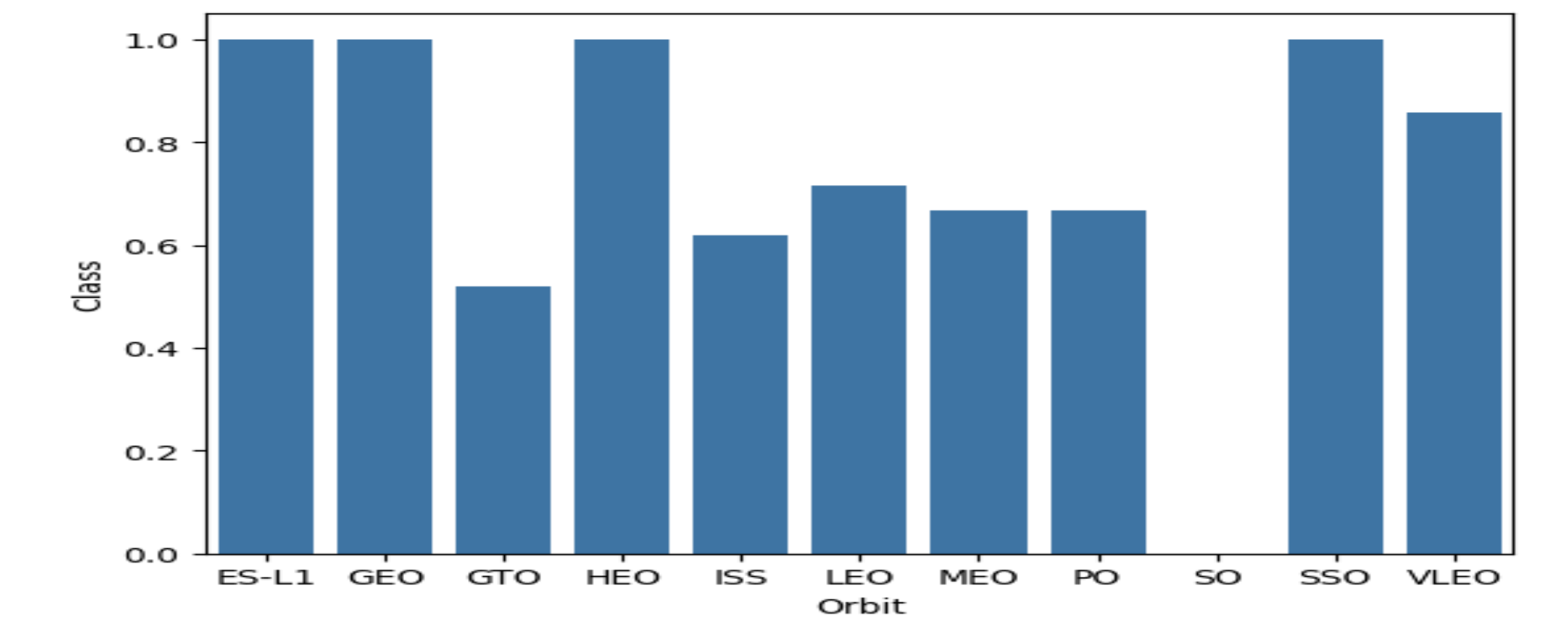

All the launch sites had more successful outcomes with later flights than with earlier. The VAFB SLC 4E and KSC LC 39A sites appear to have proportionally more successes than the CCAFS SLC 40 site.

# Payload vs. Launch Site



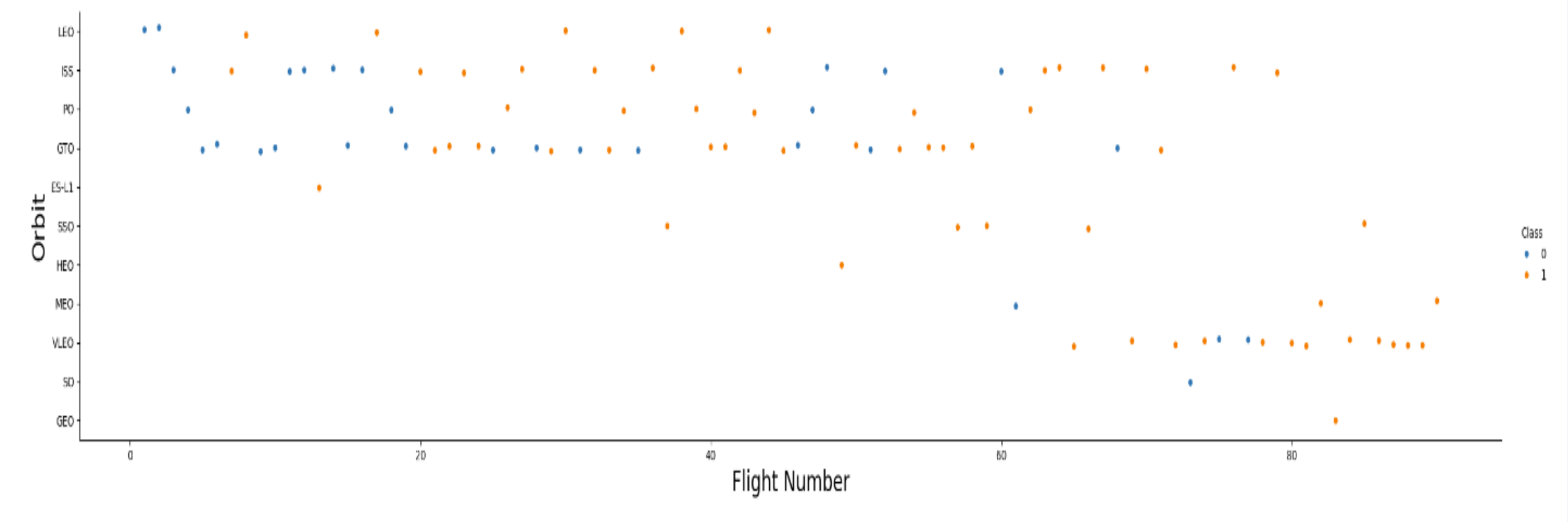

For all the launch sites the majority of launches had payloads less then 10000 kg and the VAFB SLC 4E site had no launches with a payload heavier than 10000 kg.

# Success Rate vs. Orbit Type



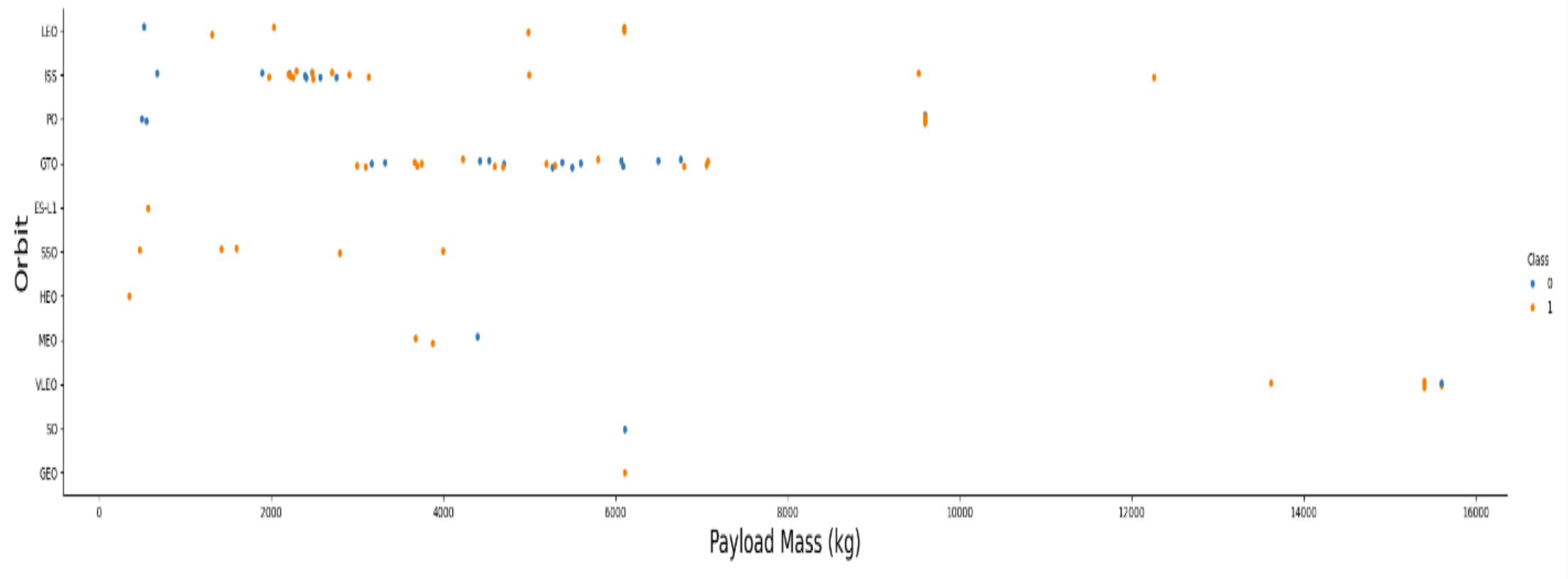

Launches into the ES-L1, GEO, HEO and SSO orbits have the highest success rates.

# Flight Number vs. Orbit Type



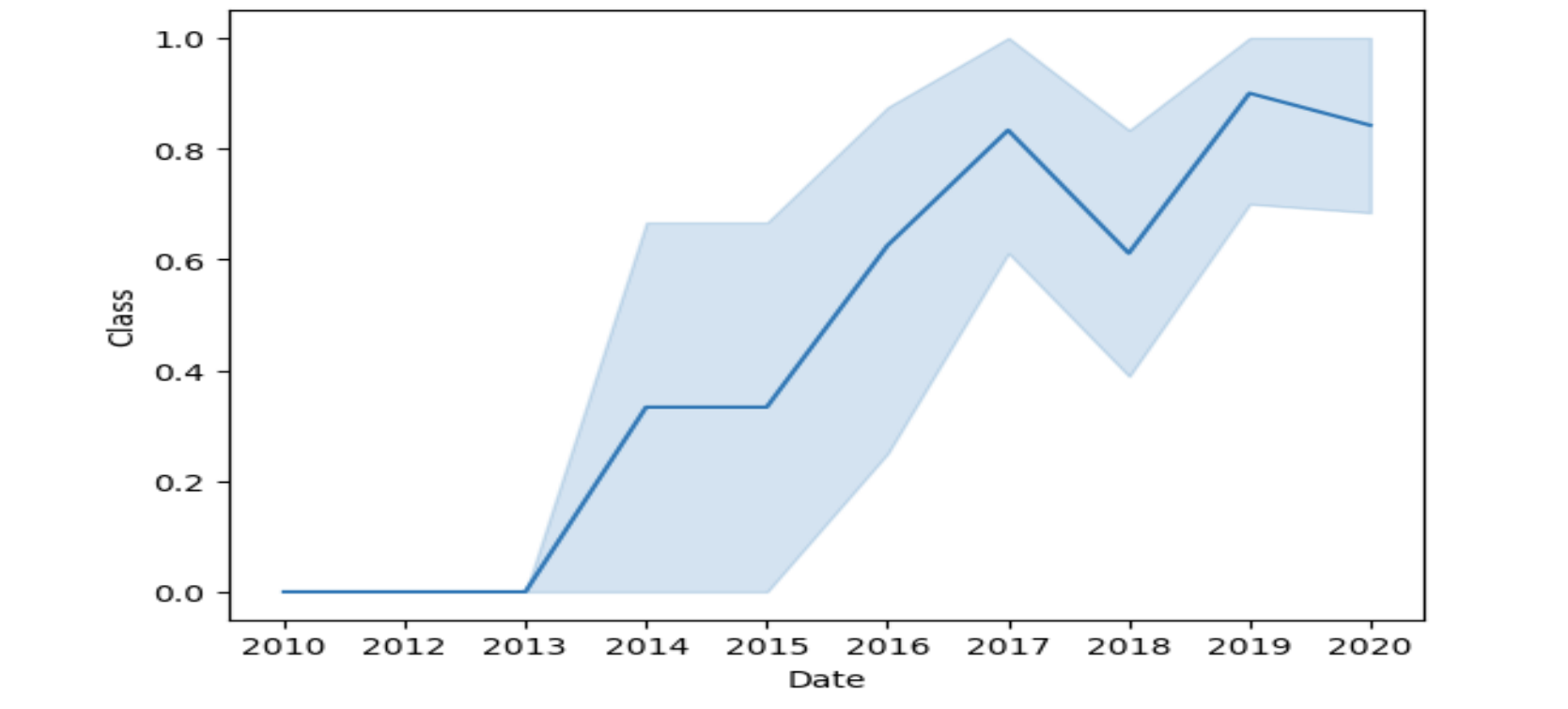

The success of launches into the LEO orbit the appears to improve with later flights. Launches into other orbits don't appear to show a clear pattern based on flight number.

# Payload vs. Orbit Type




For launches into LEO, ISS or Polar (PO) orbits success appears to increase with heavier payloads. For launches into GTO orbit the success rate is mixed across different payloads.

# Launch Success Yearly Trend




The yearly success rate, overall, increased from 2013 to 2020, with a dip in 2018.

# All Launch Site Names

Display the names of the unique launch sites in the space mission

```
%sql select distinct "Launch_Site" from SPACEXTABLE;
```

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

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

There are 4 unique launch sites. The distinct operator ensures that only unique values are returned in the result set.

# Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5;
```

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

| 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 |

The like operator in the where clause matches all records where the value in the Launch_Site column begins with ‘CCA’. The limit clause is given the value 5 which means only 5 of the records that match will be returned.

# Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select "Customer", sum("PAYLOAD_MASS__KG_") as "Total Payload Mass" from SPACEXTABLE where "Customer" like "NASA%";
```

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

| Customer | Total Payload Mass |
|---|---|
| NASA (COTS) NRO | 99980 |

The sum function adds the values in the result set in the PAYLOAD_MASS_KG_ column. The where clause conditions that the records returned will have Customer values that begin with 'NASA'.

# Average Payload Mass by F9 v1.1

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

```
%sql select avg("PAYLOAD_MASS__KG_") as average_mass from SPACEXTABLE where "Booster_Version" = "F9 v1.1";
```

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

| average_mass |
| --- |
| 2928.4 |

The avg function averages the values in the PAYLOAD_MASS__KG_ column. The where clause conditions the records returned to have Booster_Version values equal to ‘F9 v1.1’.

# First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

*Hint:Use min function*

```
%sql select * from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)" order by Date asc limit 1;
```

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

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2015-12-22 | 1:29:00 | F9 FT B1019 | CCAFS LC-40 | OG2 Mission 2 11 Orbcomm-OG2 satellites | 2034 | LEO | Orbcomm | Success | Success (ground pad) |

First successful ground pad landing was on December 22, 2015. The where clause conditions the records returned to have Landing_Outcome equal to ‘Success (ground pad). The result is ordered by Date in ascending order and limited to 1 record which would be the record with the earliest date.

# Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%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.
```

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

Four different booster versions have successfully landed on the drone ship with payloads between 4000 and 6000 kg. The where clause conditions the records returned to have Landing_Outcome value of ‘Success (drone ship)’ and PAYLOAD_MASS__KG_ value between 4000 and 6000. The distinct operator returns only the unique Booster_Version values in the result set.

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql select "Mission_Outcome", count(*) as "Total" from SPACEXTABLE group by "Mission_Outcome";
```

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

| Mission_Outcome | Total |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

Appears to be a total of 100 successful mission outcomes. The group by clause groups the result set by the value of Mission_Outcome. The count(*) function counts the number of records in each group.

# Boosters Carried Maximum Payload

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

```
%sql select "Booster_Version" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

| 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 |

There are 12 booster versions that have carried the maximum payload mass. The subquery returns the maximum value in the PAYLOAD_MASS__KG_ column. The where clause conditions the records to have a PAYLOAD_MASS_KG_ column value equal to the maximum value.

# 2015 Launch Records

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.

**Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.**

```
%sql select substr("Date",6,2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where "Landing_Outcome" = "Failure (drone ship)" and substr("Date",0,5) = '2015';
```

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

| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

In 2015 in the months of January and April there were failed drone ship landings. The where clause conditions the records to have Landing_Outcome value equal to 'Failure (drone ship)' and a Date value with the year equal to 2015.

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

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.

```
%sql select "Landing_Outcome", count(*) as Outcome_Count from SPACEXTABLE where "Date" between '2010-06-04' and '2017-03-20' group by "Landing_Outcome" order by "Outcome_Count" desc;
```

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

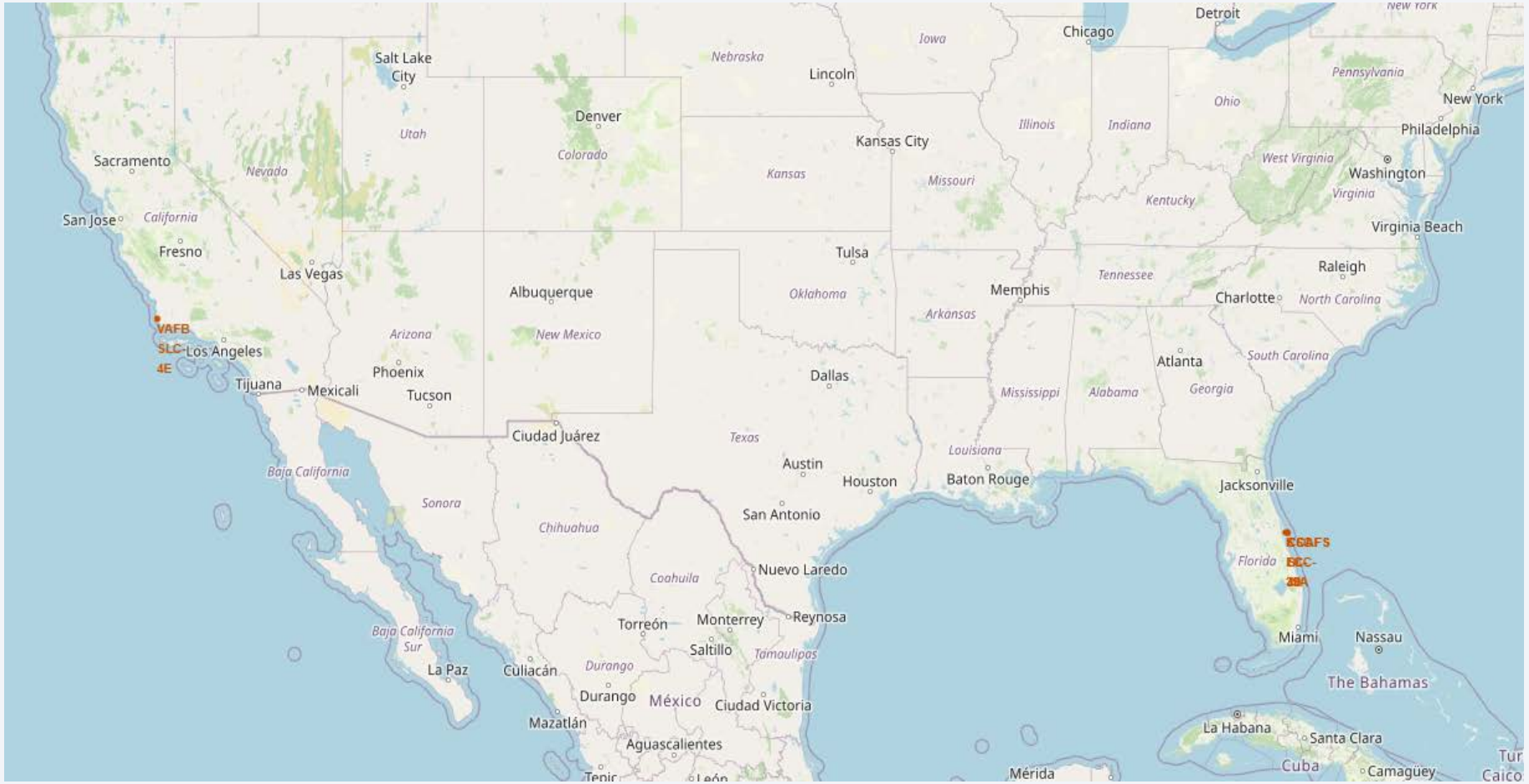
| Landing_Outcome | Outcome_Count |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

Between June 4, 2010 and March 20, 2017 there were 8 different landing outcomes, with ‘No attempt’ being the most numerous at 10. The where clause conditions the records to have Date value between ‘2010-06-04’ and ‘2017-03-20’. The group by clause groups the records by Landing_Outcome. The order by clause lists the grouped records in descending order by Outcome_Count. The count(*) function counts the records in each group.

Section 3

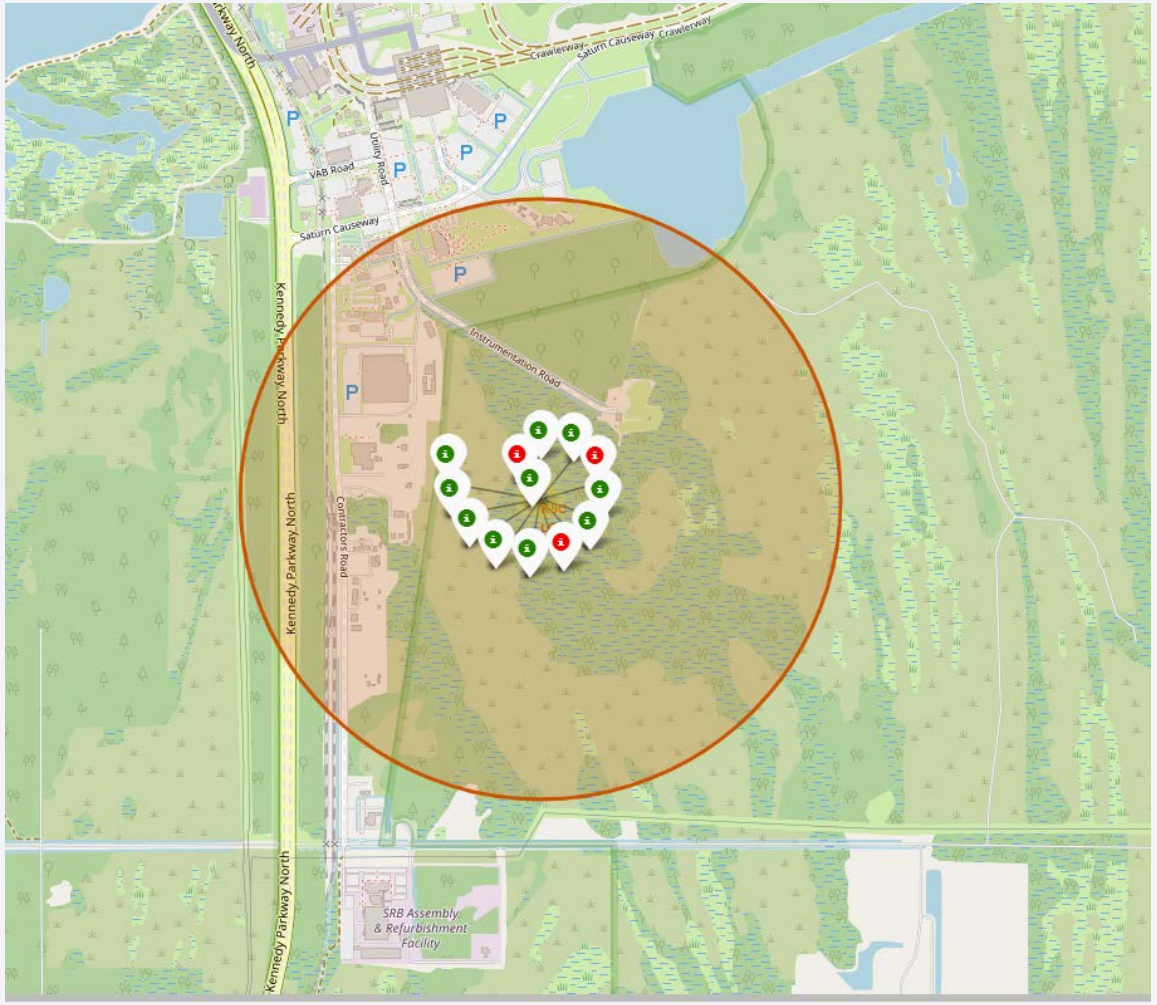
# Launch Sites Proximities Analysis

# Launch Site Locations




This map shows the locations of the 4 launch sites. Three (CCAFS SLC-40, CCAFS LC-40 and KSC LC-39A) in Florida at Cape Canaveral. One (VAFB SLC-4E) in California at Vandenberg Space Force Base.

# Launch Outcomes – Example KSC LC-39A



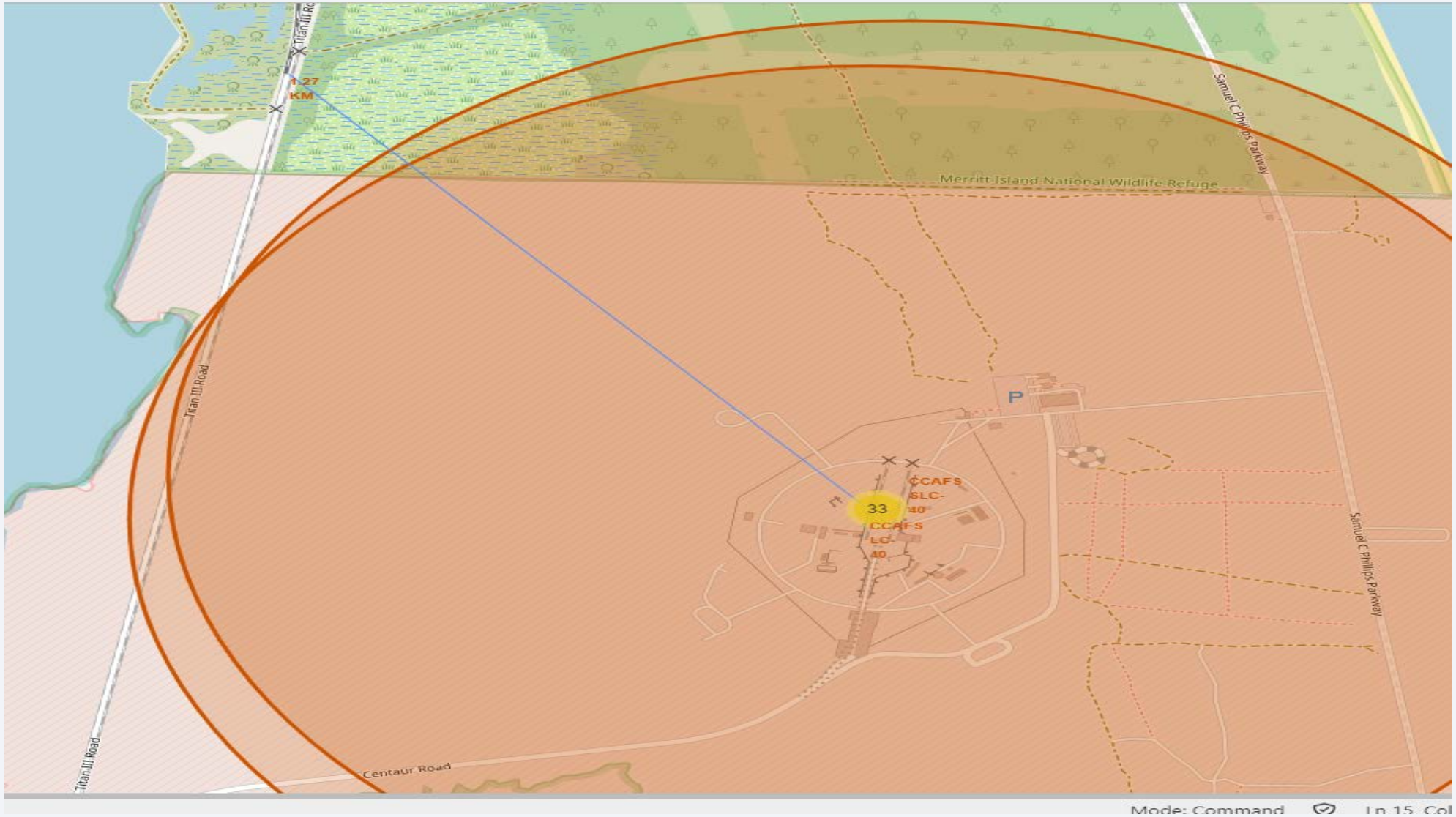

This map shows the launch outcomes at the KSC LC-39A site. The green markers indicate a successful outcome, while the red markers indicate a failure. The map shows there were 10 successful outcomes and 3 failures.

# Site CCAFS LC-40 Distance to Coastline



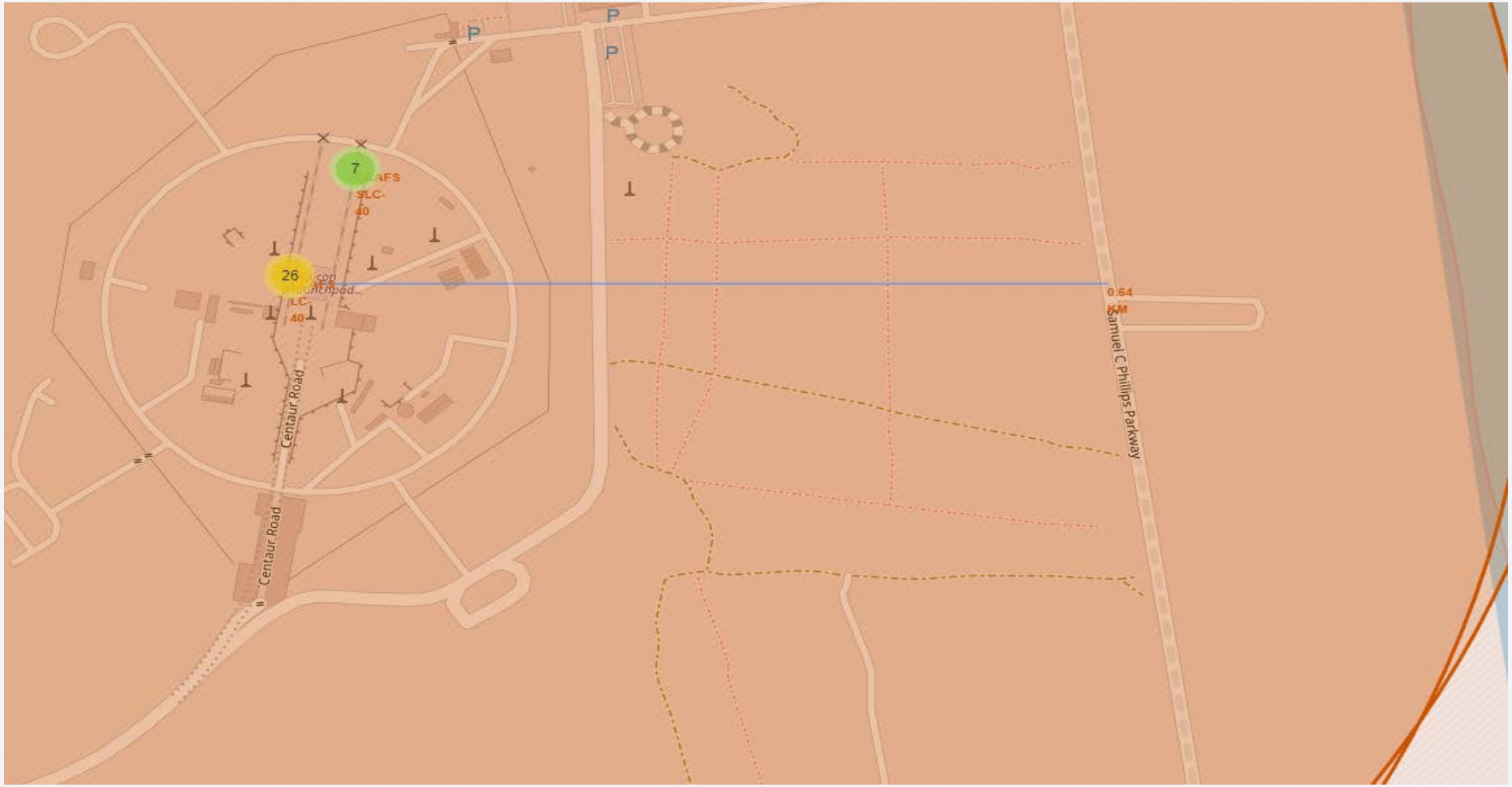

This map shows the distance of the CCAFS SLC-40 site to the nearest coastline, which is about 0.92 km.

# Site CCAFS LC-40 Distance to Railway



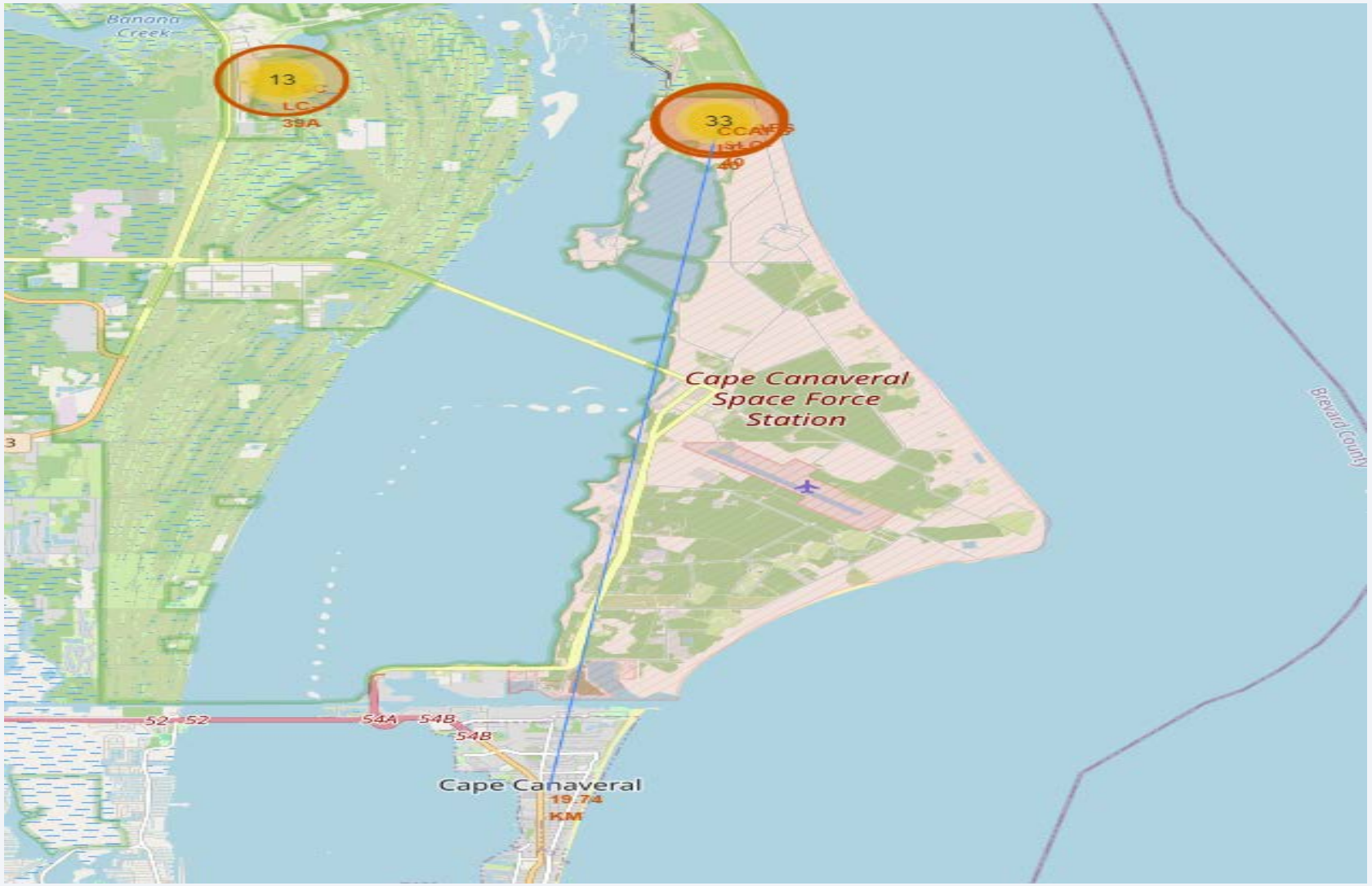

This map shows the distance from the CCAFS SLC-40 site to the nearest railway, which is about 1.27 km.

# Site CCAFS LC-40 Distance to Highway

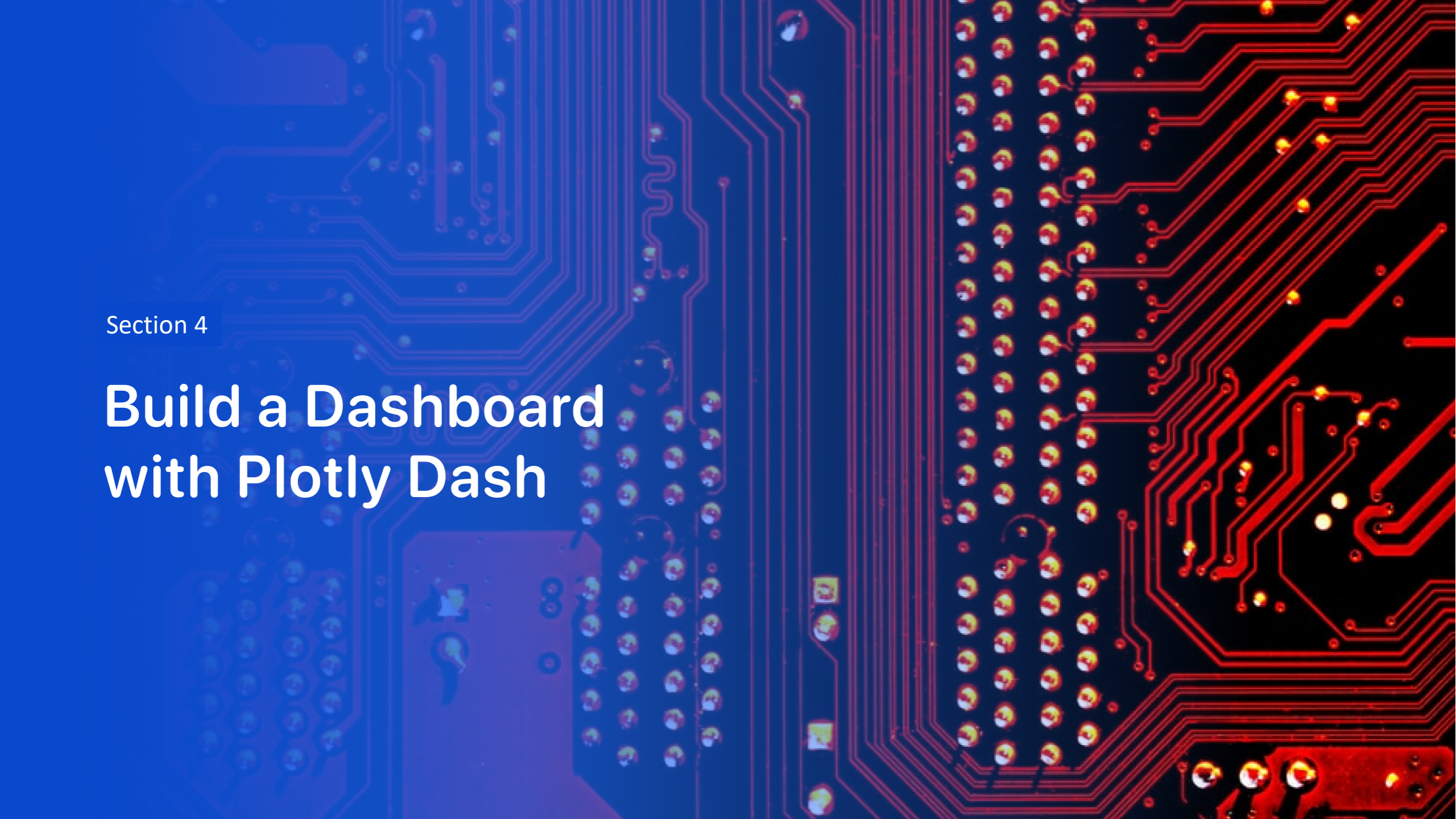



This map show the distance from the CCAFS SLC-40 site to the nearest highway, which is about 0.64 km.

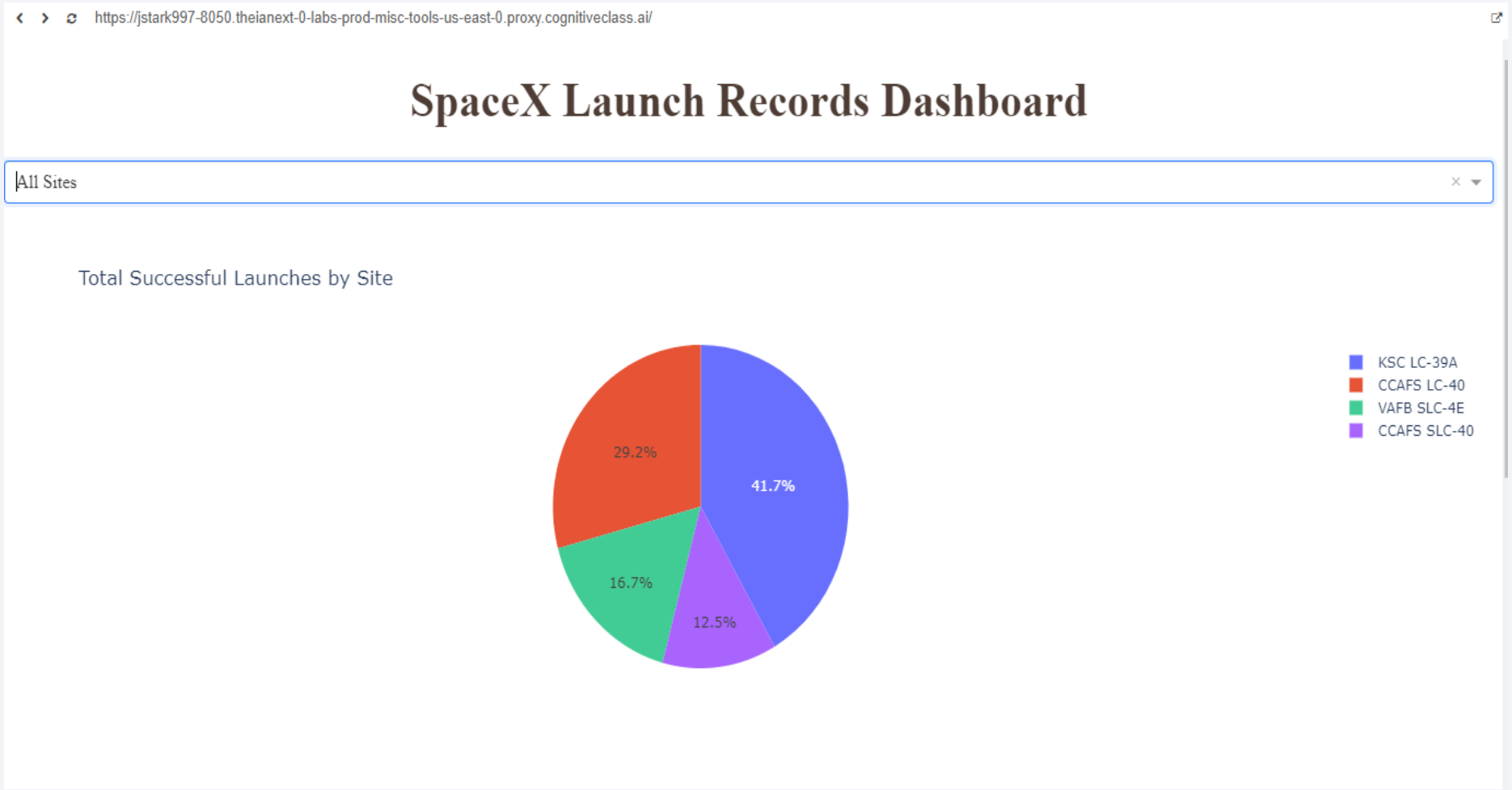
# Site CCAFS LC-40 Distance to Town




This map shows the distance from the CCAFS SLC-40 site to the nearest town, Cape Canaveral, which is about 19.74 km.

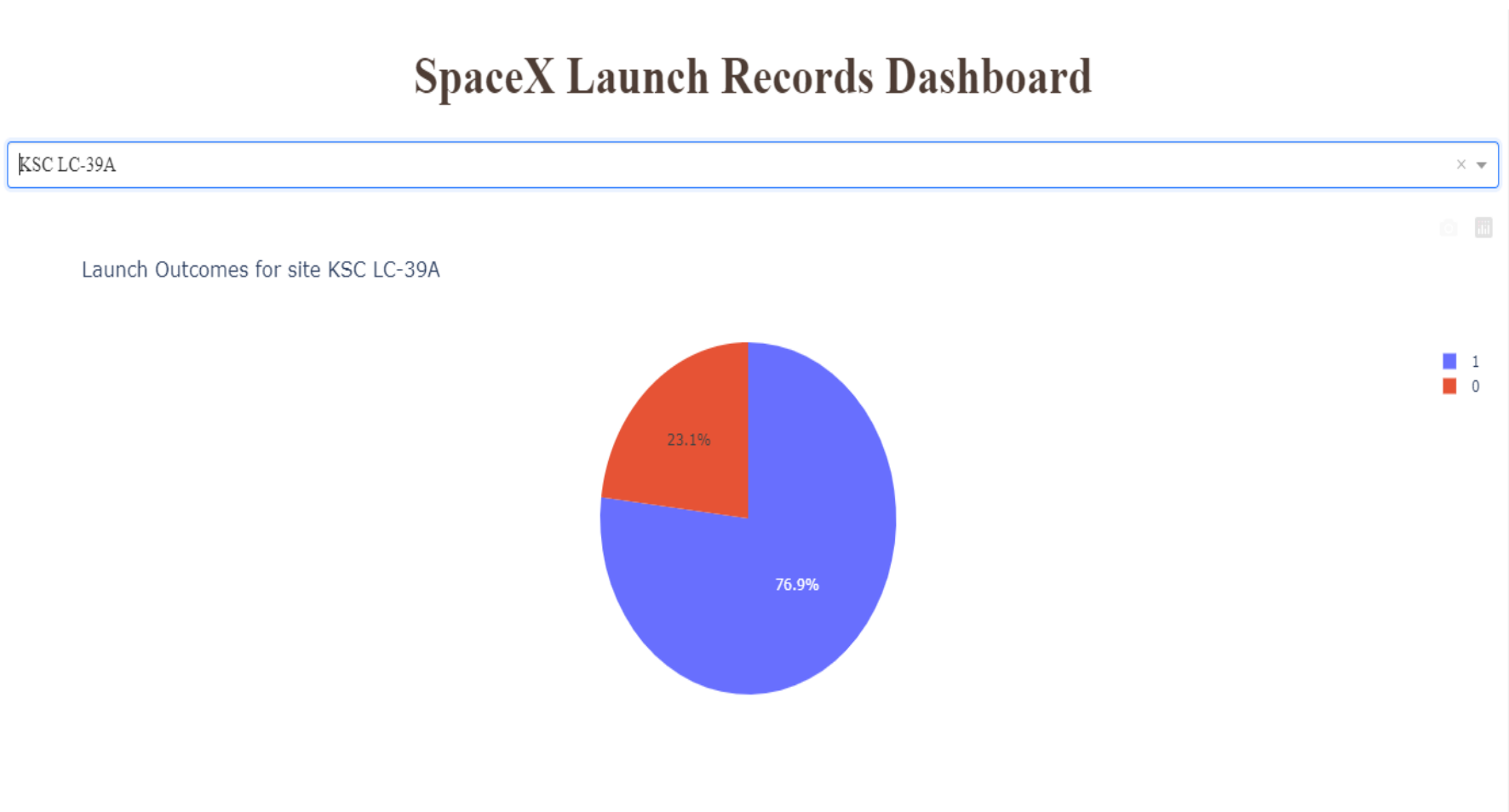

Section 4
Build a Dashboard with Plotly Dash

# Total Successful Launches - All Sites



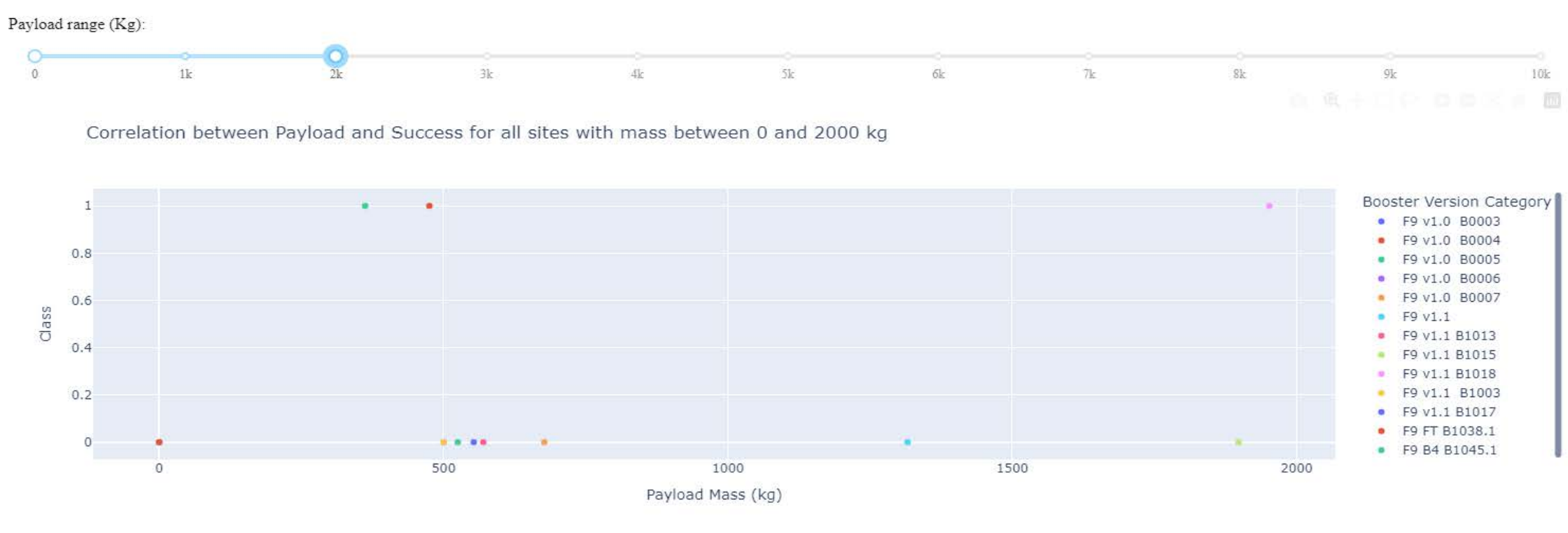

The pie chart shows the percentage of successful launches for all launch sites with the site KSC LC-39A having the largest percentage of successful outcomes at 41.7%.

# Site with Highest Launch Success Ratio



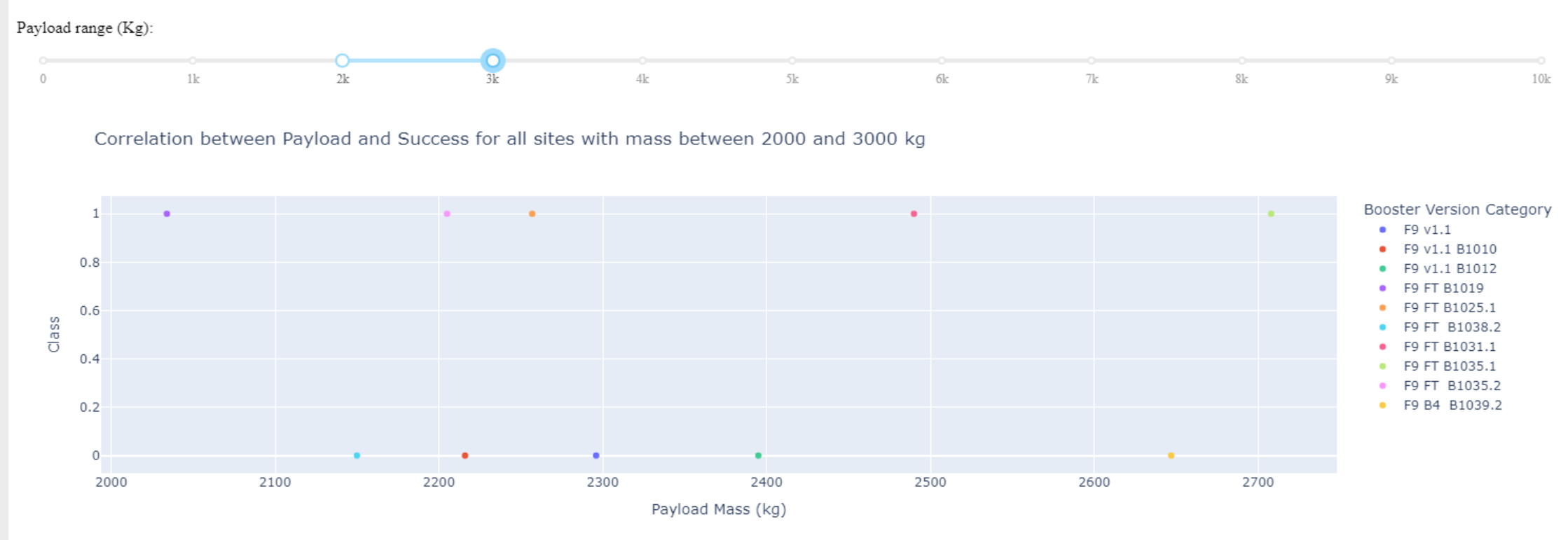

The pie chart show the ratio of success versus failure at the KSC LC-39A site. This site has the best success rate at 76.9%

# All Sites Outcomes Payload Range 0 to 2000 kg



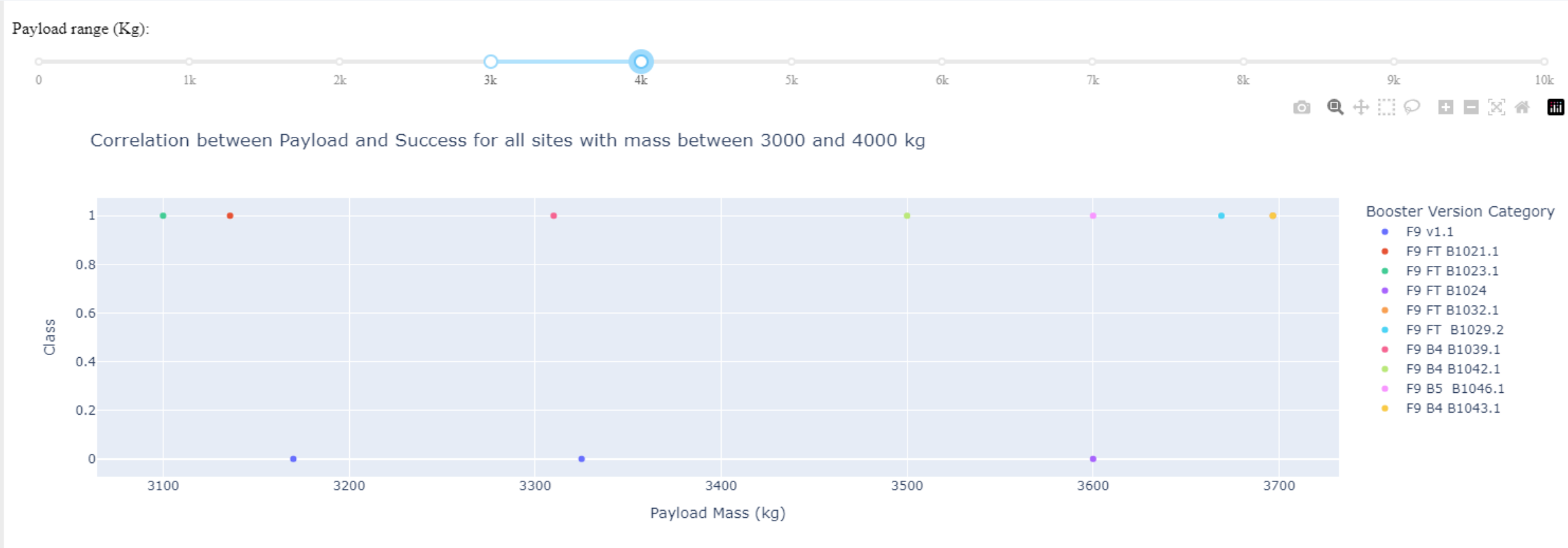

In the payload range of 0 to 2000 kg there were 3 successes and 8 failures. The boosters for the successes were B1045.1, B1038.1, B1018.

# All Sites Outcomes Payload Range 2000 to 3000 kg



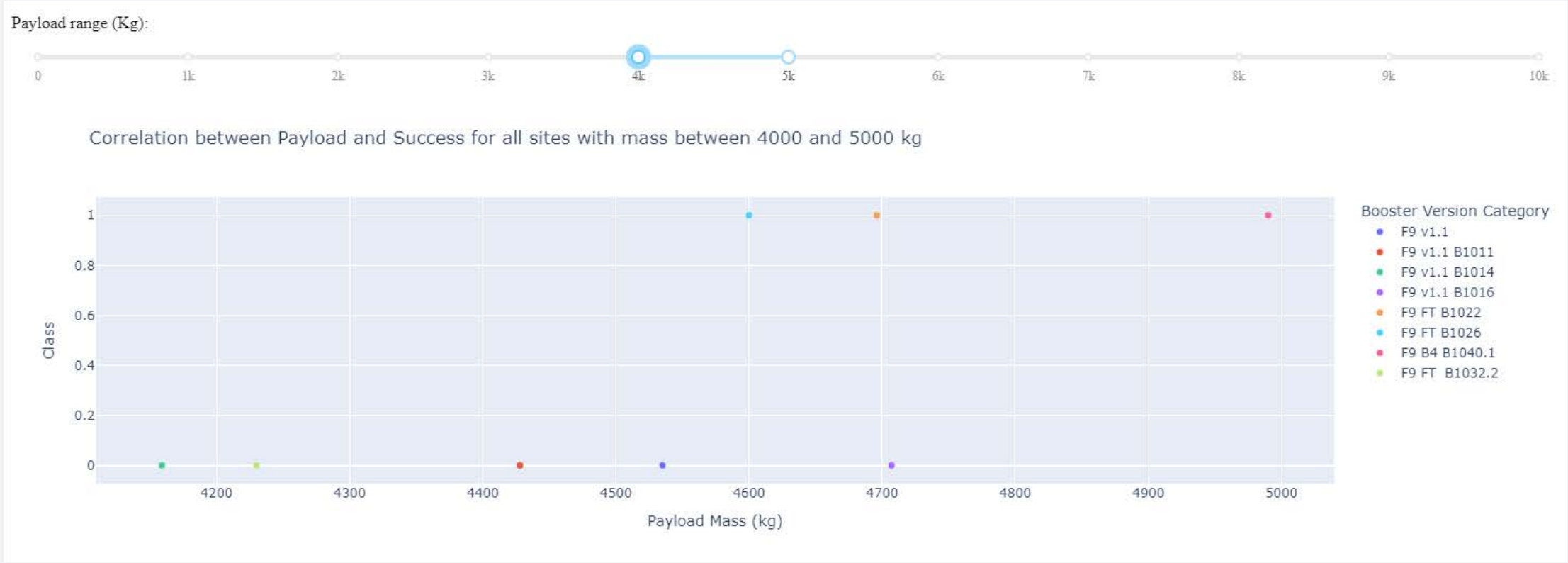

In the payload range of 2000 to 3000 kg there were 5 successes and 5 failures. The boosters for the successes were B1019, B1035.2, B1025.1, B1031.1 and B1035.1.

# All Sites Outcomes Payload Range 3000 to 4000 kg



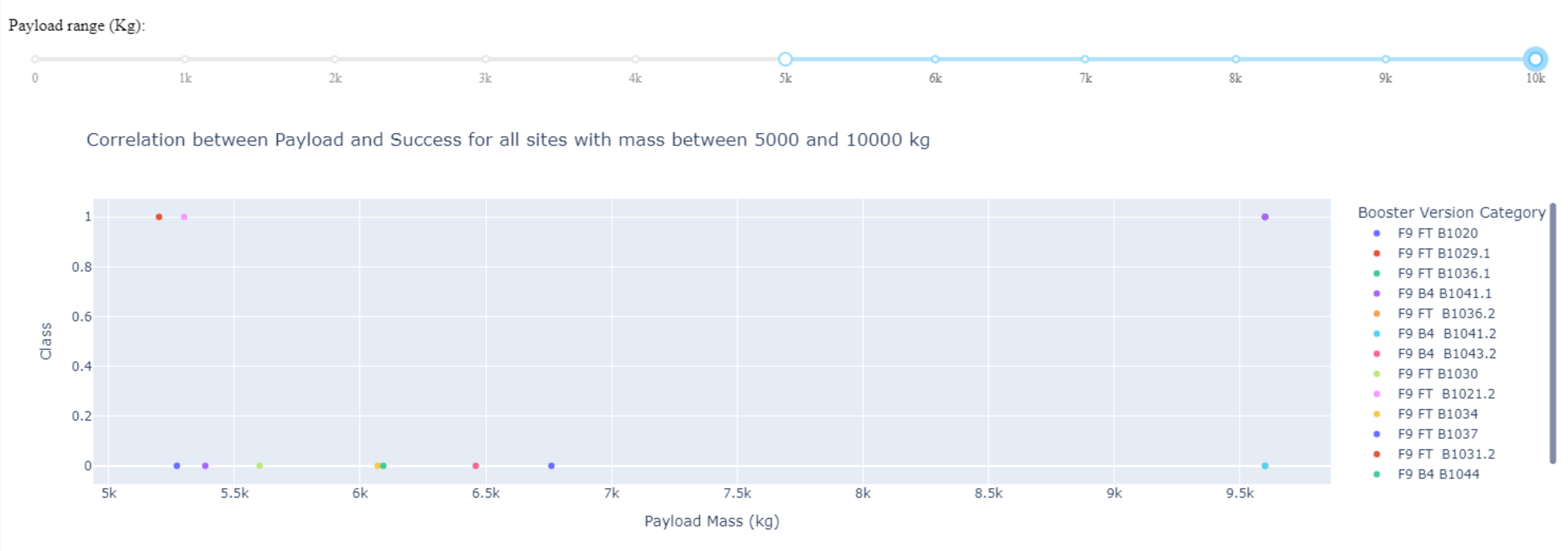

In the payload range of 3000 to 4000 kg there were 7 successes and 3 failures (which is the highest success rate for any payload range). The boosters for the successes were B1023.1, B1021.1, B1039.1, B1042.1, B1046.1, B1029.2, B1043.1.

# All Sites Outcomes Payload Range 4000 to 5000 kg




In the payload range of 4000 to 5000 kg there were 3 successes and 5 failures.
The boosters for the successes were B1026, B1022, B1040.1.

# All Sites Outcomes Payload Range 5000 to 10000 kg

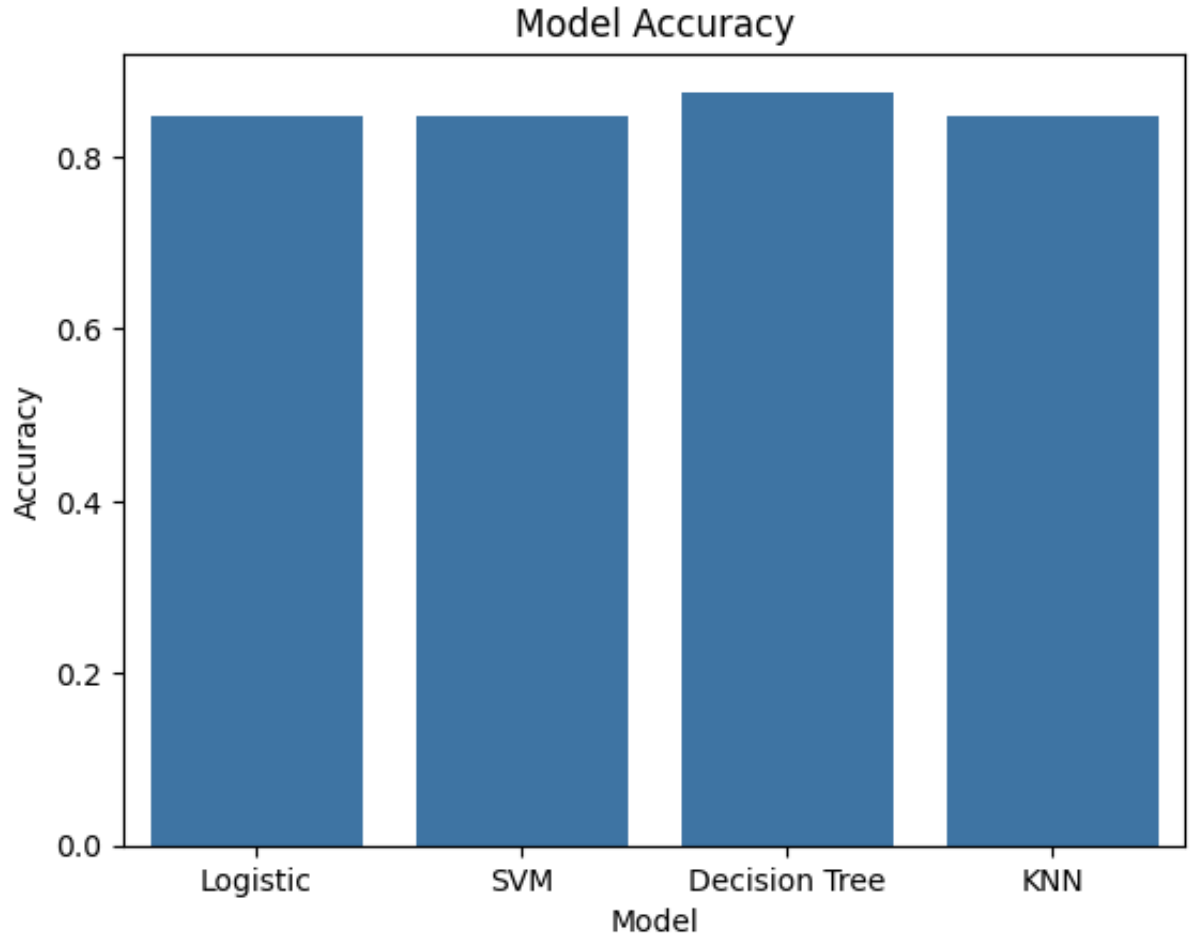



In the payload range of 5000 to 10000 kg there were 3 successes and 8 failures.
The boosters for the successes were B1031.2, B1021.2, B1041.1.

Section 5

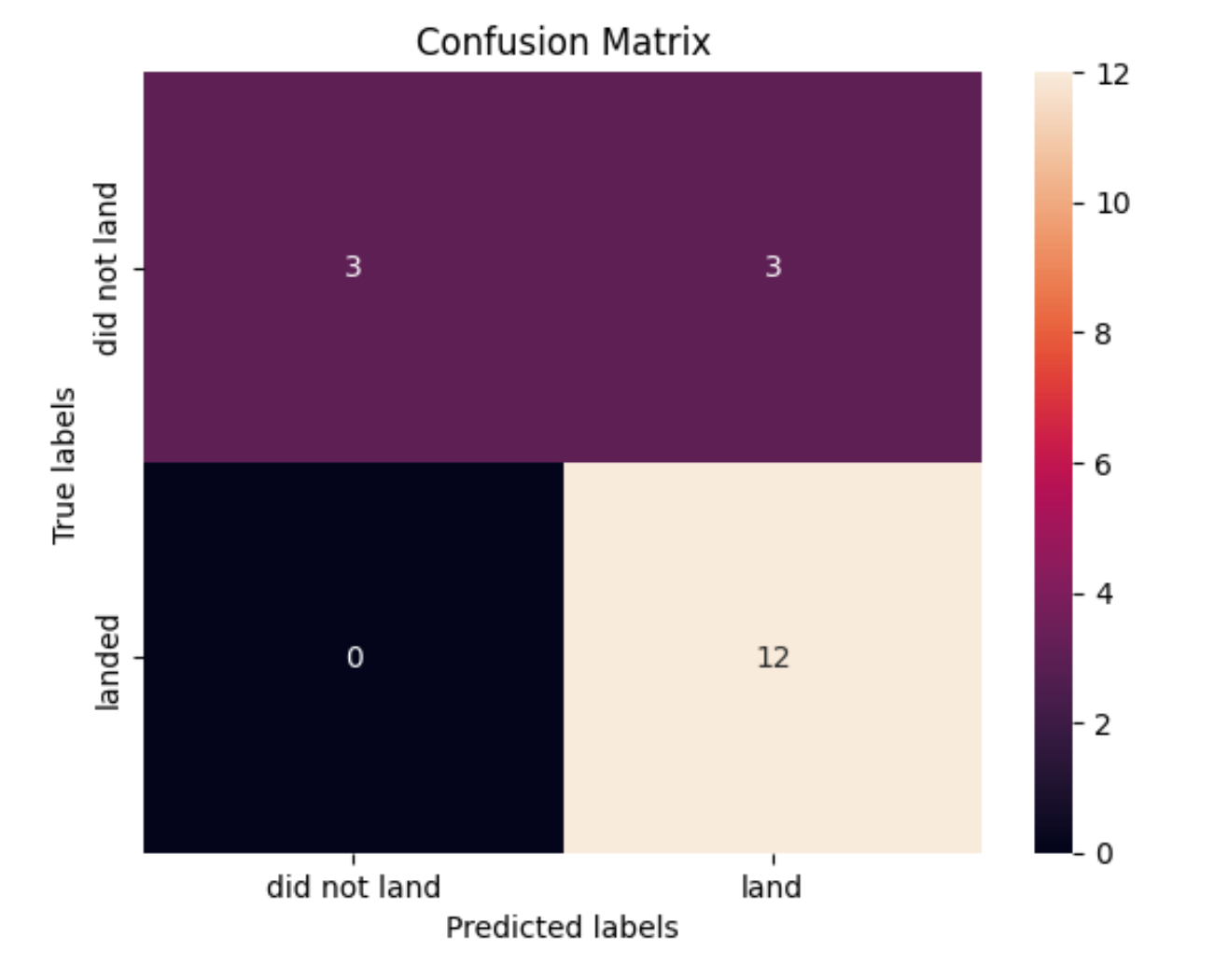
# Predictive Analysis (Classification)

# Classification Accuracy




The Decision Tree model has the highest accuracy at 0.875.

# Confusion Matrix – Decision Tree




The Decision Tree model correctly classified 3 instances as landing failures and 12 instances as landing successes.

However it miss classified 3 instances as landing successes when in fact they were failures.

Therefore this Decision Tree model is sometimes prone to false positives.

# Conclusions

- SpaceX has improved it's landing success rate over the years as later flights have landed successfully more often the earlier flights.
- All launch sites had successful landings, but the KSC LC-39A site near Cape Canaveral, Florida had the highest success rate at 76.9%.
- Perhaps unsurprisingly launches with heavier payloads landed successfully less often than with lighter payloads.
- The payload range with the highest success rate was 3000 to 4000 kg.
- Launches into the ES-L1, GEO, HEO and SSO orbits landed successfully more often than launches into other orbits.
- For heavier payloads, launches into LEO, ISS and Polar orbit had a higher success rate than launches into other orbits.
- A Decision Tree is the best model to predict if a launch will be successful with an accuracy of 87.5%. However about 1/6 of the time it produces a false positive.
- Based on exploratory analysis of the SpaceX data, the highest chance of a successful landing is a launch from the KSC LC-39A site with a payload between 3000 and 4000 kg into a ES-L1, GEO, HEO or SSO orbit.

# Appendix

- If you have trouble viewing a notebook from the provided link please use nbviewer: https://nbviewer.org/ and paste the link there.

# Thank you!