Winning Space Race with Data Science

James Lonergan
2/21/2024

Outline

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

Executive Summary

Summary of methodologies

- Booster test data was gathered from SpaceX via both web scraping and the SpaceX API
- The data was parsed formatted and cleaned
- Intial statistical and graphical analysis, was performed using SQL, Pandas, Numpy, Seaborn, Plotly, and Folium
- Some results are displayed in a Dash App
- Predictive analysis was performed using a variety of models and grid search optimization

Summary of all results:

The Falcon 9 rocket performances can be seen to be the product of a few factors: Launch Site, Payload Mass, Orbit, and Flight Date, with more recent flights having higher success rates. Large payloads, and certain Orbits tended to have lower success rates. Similarly, at Launch Sites with lower success rates, more of the early flights occured. Launch Success predictions were made with an accuracy of 83%, and little variability across the tested algorithms. A larger test set size might improve the predictive capacity of the models.

Introduction

Project background and context

- This capstone project leverages a wide range of data science skills to analyze SpaceX rocket flight data from the perspective of a SpaceX competitor.
- Successful recovery of the rocket booster reduces the launch cost by 62%. A competitor who can predict the success of a planned launch is in an advantageous position.
- By investigating the data, factors that contribute to flight success can be determined, and a predictive model developed that informs project budgets and future flight decisions

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from the SpaceX website using two methods:
    - SpaceX provides the data in an API in JSON format
    - The data can be scraped directly from the webpage
- Perform data wrangling
  - The data was cleaned and processed using SQL and Numpy, null values were removed, and a Class column was created that represents launch success as 1 and failure as 0.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Trends in the data can be seen using a variety of statistical and graphical methods.
  - Relationships can be seen using scatterplots, bar charts and time series plots

Methodology

Executive Summary Cont.

- Perform interactive visual analytics using Folium and Plotly Dash
  - Spatial data can be evaluated using Folium maps, and interactive dashboards make visual analysis of variables, trends and relationships easy and fast.
- Perform predictive analysis using classification models
  - Models were developed using a comparitive approach, and a grid search optimization algorithm.
  - 4 model types were used: linear, SVN, Decision Trees, and KNN.

Data Collection

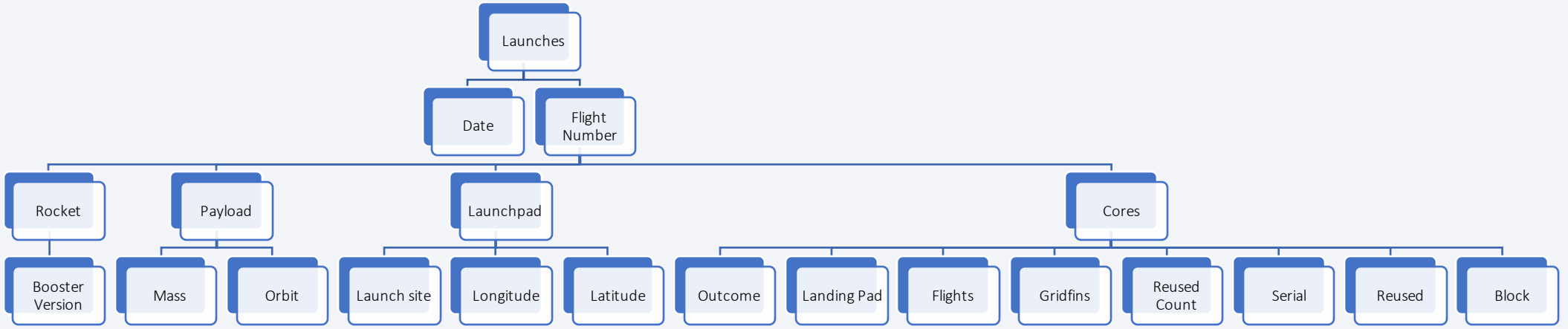
The data was gathered from the SpaceX website API, and also via web scraping. It was necessary to download some reference data, and map that data to additional tables, using a few additional API requests, and keys found in the reference dataset called launches. The following slides present detailed process diagrams for the different data collection methods, as well as the notebooks they were performed in.

Data Collection – SpaceX API

API Requests Flow Chart

GitHub URL:

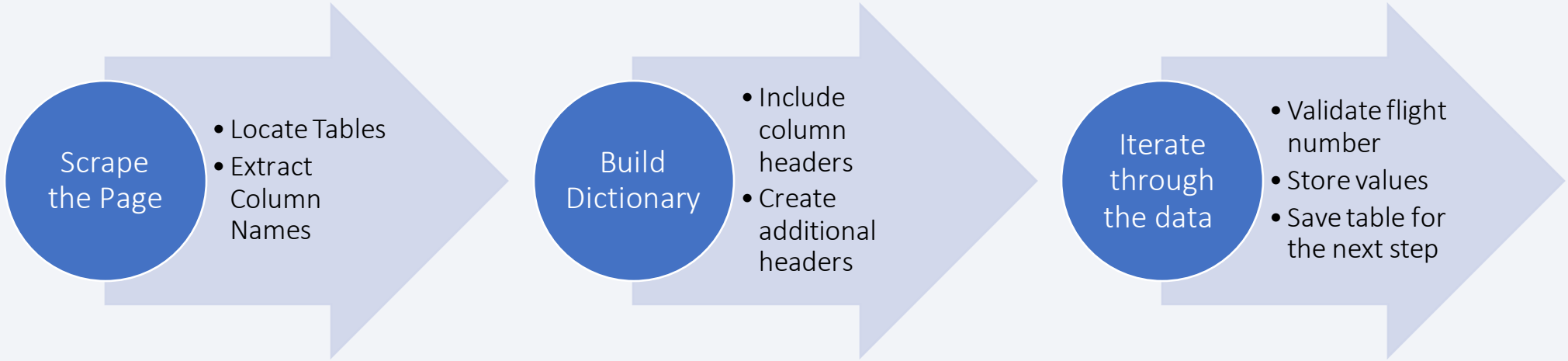
- https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

WebScraping notebook:

- https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/jupyter-labs-webscraping%20(1).ipynb



Data Wrangling

- Once Gathered, the data was processed, cleaned and formatted for analysis
- The data was filtered to include only a certain date range, and only Falcon 9 Rockets
- Null values were replaced with mean values for the column Payload Mass
- The data was analyzed to determine which landing outcomes constitute a success and which a failure
- A new column was created to indicate success or failure with 1 or 0

- URL: https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Data Visualization Charts:
  - A scatter plot of flight number vs payload mass (grouped by success class)
  - A scatter plot of flight number vs launch site (grouped by success class)
  - A scatter plot of payload mass vs launch site (grouped by success class)
  - A bar chart of success rate by orbit
  - A scatter plot of flight number vs orbit (grouped by success class)
  - A scatter plot of payload mass vs orbit
  - A time series line plot of average annual success rate
  - URL: https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

EDA with SQL

- SQL queries
  - sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
  - sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
  - Sql SELECT "Launch_Site" FROM SPACEXTABLE WHERE "Launch_Site" LIKE "%CCA%" LIMIT 5
  - Sql SELECT sum(Payload_Mass_KG_) FROM SPACEXTABLE
  - Sql SELECT avg(Payload_Mass_KG_) FROM SPACEXTABLE WHERE "Booster_Version" LIKE "F9 V1.1"
  - Sql SELECT min(Date) FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (ground pad)
  - Sql SELECT "Landing_Outcome" FROM SPACEXTABLE WHERE "Booster_Version" = "Success (drone ship)" AND "Payload_Mass_KG_" > 4000 AND "Payload_Mass_KG_" < 6000
  - Sql SELECT COUNT(Mission_Outcome), "Mission_Outcome" FROM SPACEXTABLE GROUP BY "Mission_Outcome"
  - Sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass_KG_" = SELECT MAX(Payload_Mass_KG_) FROM SPACEXTABLE
  - %sql 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' "
  - %sql SELECT count(Landing_Outcome),"Landing_Outcome" FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' and '2017-03-20' group by "Landing_Outcome" ORDER BY count(Landing_Outcome) DESC
  - URL: https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/SQL_lab_coursera.pdf (PDF, the lab would not save any of my work, so I took screens)

Build an Interactive Map with Folium

- In order to make inferences on the geospatial data, map markers were added to the folium map.
- Circle markers were added to indicate each of the launch sites, with hover over text.
- Marker clusters were added to indicate launch success or failure at each launch site and color coded
- A mouse position marker was added to show geospatial coordinates at mouse locations. This marker was used to measure the distance between one of the launch sites and the coast.
- A line was added to mark the distance measured between on of the launch sites and the nearest coastline.

- URL: https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

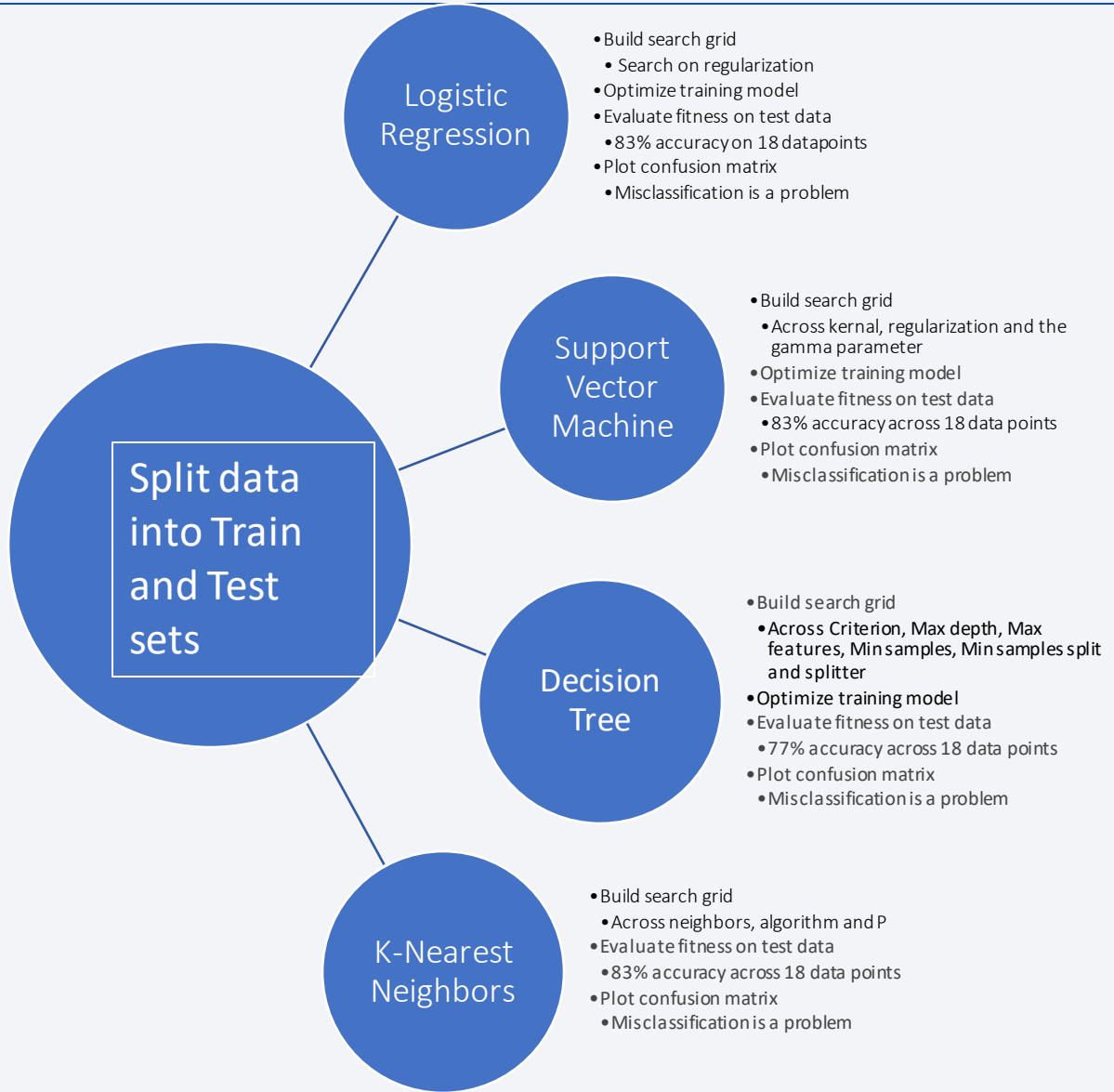
Build a Dashboard with Plotly Dash

- A Dash web application was built to render interactive figures for data analysis.
- The figures were developed using the Plotly package, which is a high-level interactive plotting library with many interactive features built in.
- Two figures were created:
- A pie chart that incorporated a user definable dropdown menu, with options to select All Launch Sites or an individual launch site. The pie chart rendered the percentage of successful launches from each site if all sites are selected, or if 1 site is selected the pie chart shows the percentage of succesful and not succesful launches.
- A scatter plot was created with a slider range based on payload mass, comparing payload mass to success rate with the color variable set to booster type
- These plots highlight some of the likely dependencies found in the data, and allow the data scientist to interact with the data to get a better sense of the impact of variables on each other, and to change the resolution of the data.
- URL of app code and screenshots:

https://github.com/jlonergan35/python_capstone/blob/e8108a58c6b7e853a01db4fd24354fe4d43b5e10/dash_interactive.py

https://github.com/jlonergan35/python_capstone/blob/a06c81f1d6965ab7aed5d7940882728600e1f28b/dash%20app%20screens.pdf

Predictive Analysis (Classification)

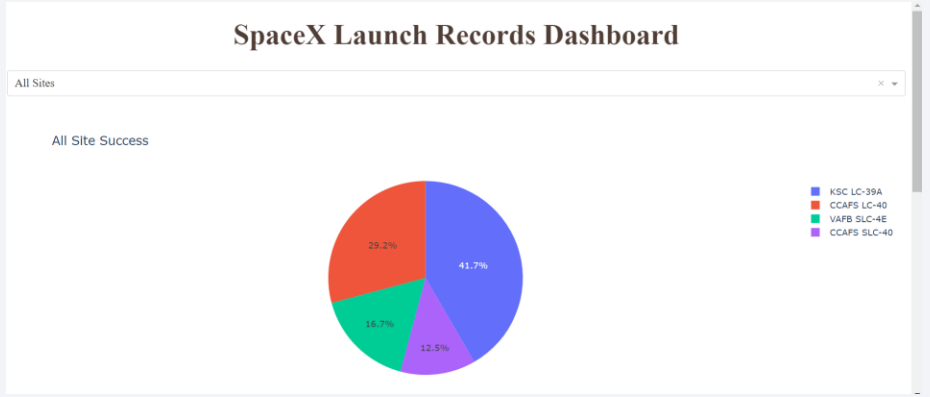


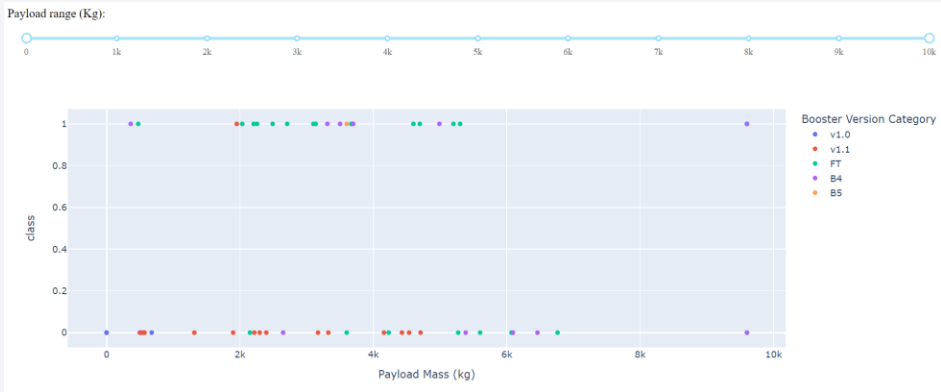
Results

- Exploratory data analysis results
  - Correlations were observed between success rate and payload mass, launch site and Orbit
  - The strongest corrolation was found between success rate and flight number
  - The success rate for any configuration of rocket and launch site will improve linearly based on the number of previous launch attempts

- Predictive analysis results
  - LR, SVM and KNN all performed equally well, with a test accuracy of 83.3%,
  - Decision Trees had an accuracy of 77%, however fit the training data better than the other three models.
    - The small size of the test data set may have impacted the fitness of the decision tree model, a larger data set and a larger test set might change the results

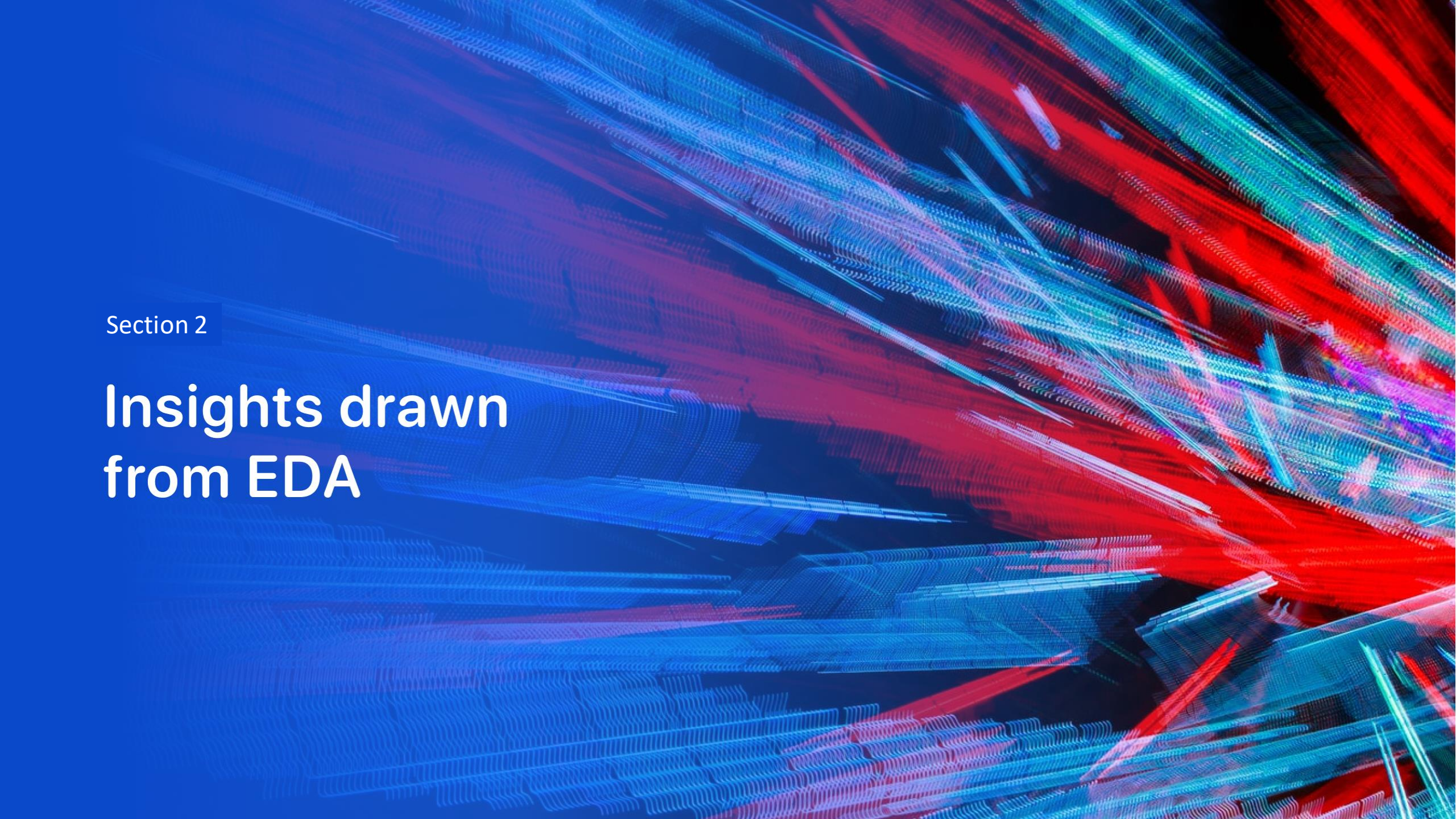
Interactive analytics demo in screenshots



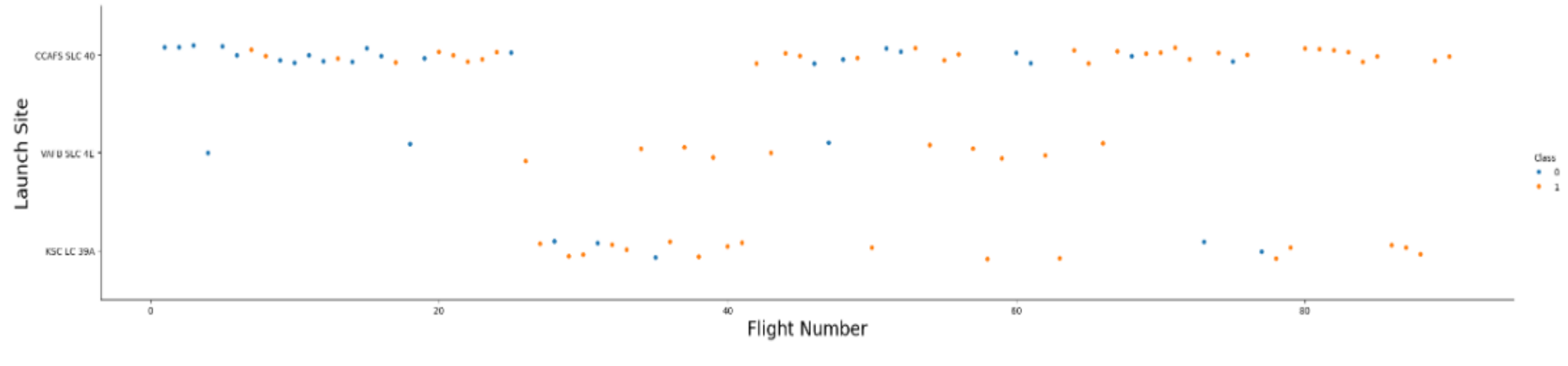


Section 2

Insights drawn from EDA



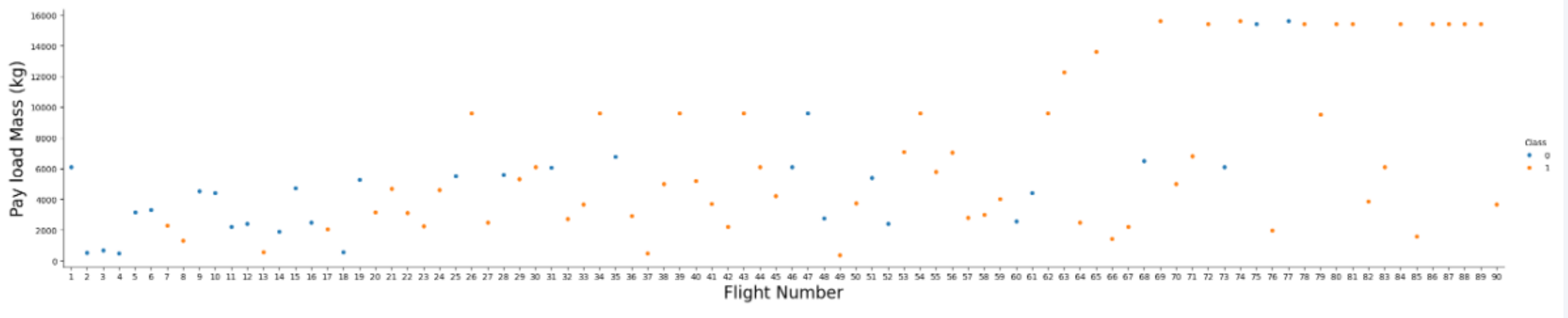
Flight Number vs. Launch Site



It can be observed that CCAFS SLC 40 has the highest rate of failure, but that those failures are mostly early launch numbers

the other two sites had fewer total launches and KSC LC 39A does not show much improvement over time compared to the other two sites

Payload vs. Launch Site

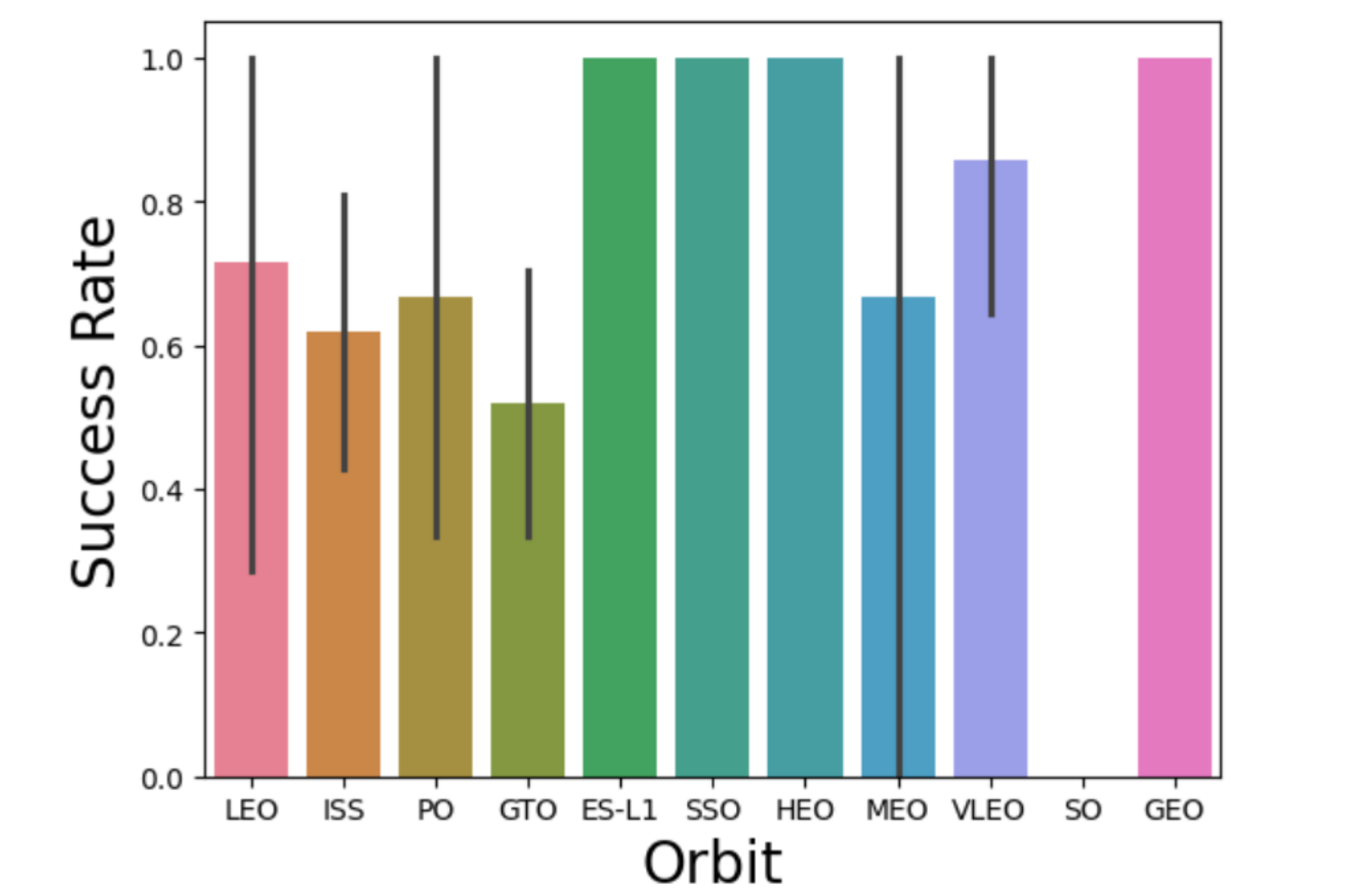


Payload mass increases with launch number, and success rate improves with launch number.

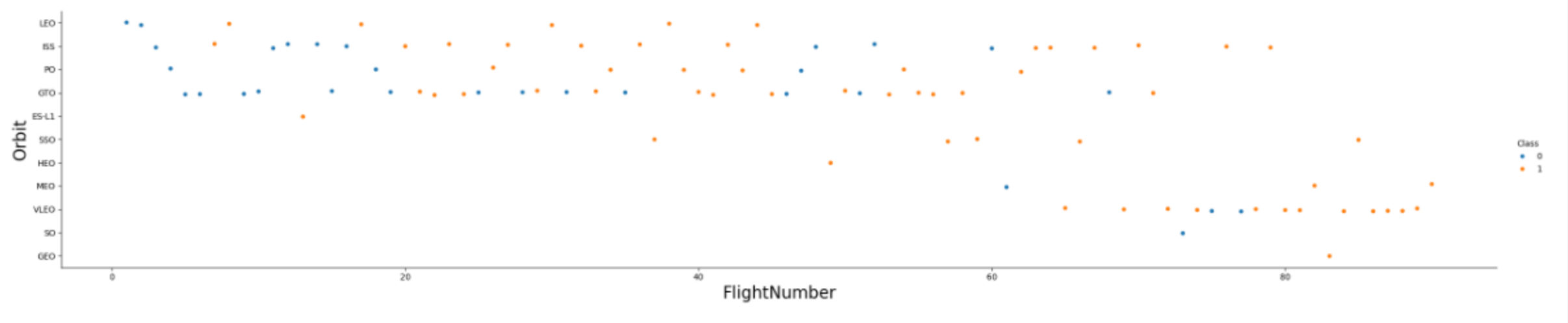
Smaller payloads showed more improvement than larger payloads

Success Rate vs. Orbit Type

- The success rate for some of the orbits is 100%
- GTO has the lowest success
- GTO and ISS are statistically unlikely to have 100%
- success, the remaining orbits vary between 75% and 90%

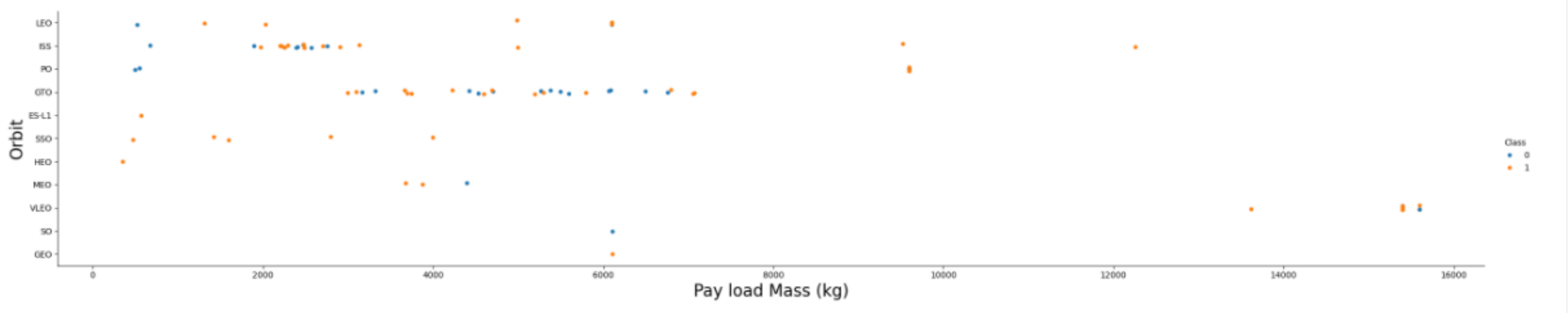


Flight Number vs. Orbit Type



- The orbits with the lowest success rate also have the earliest flight numbers
- The GEO orbit, which has 100% success, only had 1 flight, similarly SO only had 1 flight and it failed.
- Success rate seems to improve for all orbits with more than 1 flight as flight number goes up.

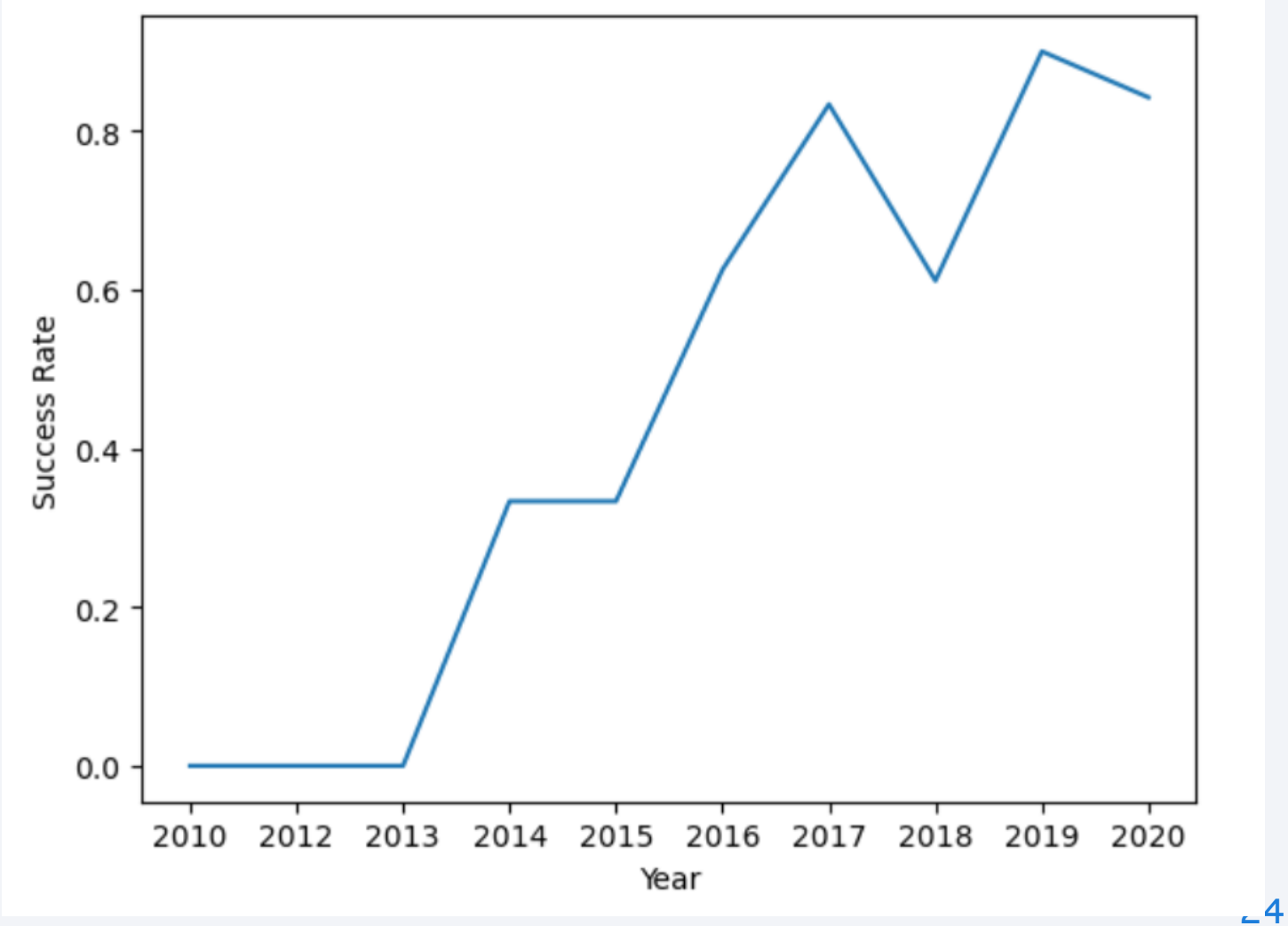
Payload vs. Orbit Type



- As payload mass increases success rate for Polar, LEO and ISS

Launch Success Yearly Trend

- Success rate increases linearly over time, with the exception of 2018
- This trend is the strongest indicator of success in the data set
- Equivalent launches will increase in success the more attempts are made



All Launch Site Names

- This query shows the unique launch site names

Task 1

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

```
[9]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

[9]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

```
1]: %sql select "Launch_Site" from SPACEXTABLE where "Launch_Site" like "%CCA%" limit 5
 * sqlite:///my_data1.db
Done.
```

1]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- The query looks for the string CCA and limits the records to the first 5

Total Payload Mass

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" = "NASA (CRS)"
```

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

sum(PAYLOAD_MASS__KG_)
45596

This query calculates the total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
]: %sql select avg(PAYLOAD_MASS__KG_) as payloadmass_avg from SPACEXTABLE where "Booster_Version" like "F9 v1.1"
```

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

]:

payloadmass_avg
2928.4

This query calculates the average payload for booster F9 v1.1

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
19]: %sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"
 * sqlite:///my_data1.db
Done.
```

19]:

min(Date)
2015-12-22

- This query selects the minimum date that meets the landing outcome condition of Success (Ground Pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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 "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_
```

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

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

- This query selects the boosters that had the desired landing outcome and payload range

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select count(Mission_Outcome),"Mission_Outcome" from SPACEXTABLE group by "Mission_Outcome"
```

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

count(Mission_Outcome)	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

This query shows the different mission outcome totals, 100 successful missions and 1 failure

Boosters Carried Maximum Payload

Task 8

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

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

boosterversion
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3

- This query filters the booster version to select all the versions that carried the max payload mass

2015 Launch Records

Task 9

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.

```
]: select substr(Date, 6,2), "Landing_Outcome","Booster_Version","Launch_Site" from SPACEXTABLE where "Landing_Outcome" = "Failure (drone ship)" AND subst
```

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

]:

substr(Date, 6,2)	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

- This query lists the drone ship failures in 2015 by month, as well as the launch site and the booster version

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

Task 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.

```
[49]: %sql SELECT count(Landing_Outcome),"Landing_Outcome" FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' and '2017-03-20' group by "Landing_Outcome" ORD
```

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

[49]:

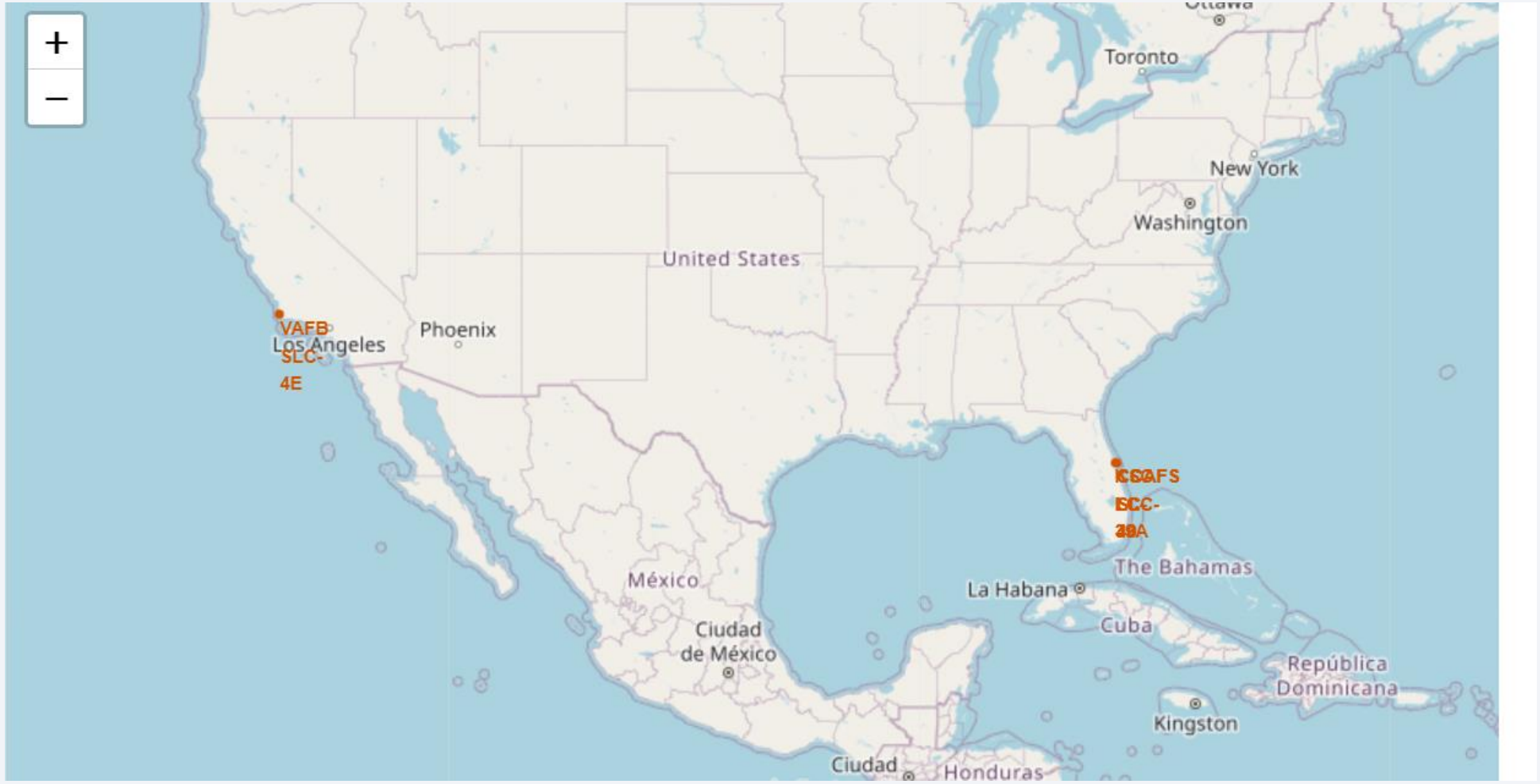
count(Landing_Outcome)	Landing_Outcome
10	No attempt
5	Success (drone ship)
5	Failure (drone ship)
3	Success (ground pad)
3	Controlled (ocean)
2	Uncontrolled (ocean)
2	Failure (parachute)
1	Precluded (drone ship)

- This query ranks the count of landing outcomes between the specified dates in descending order


Section 3
Launch Sites
Proximities Analysis

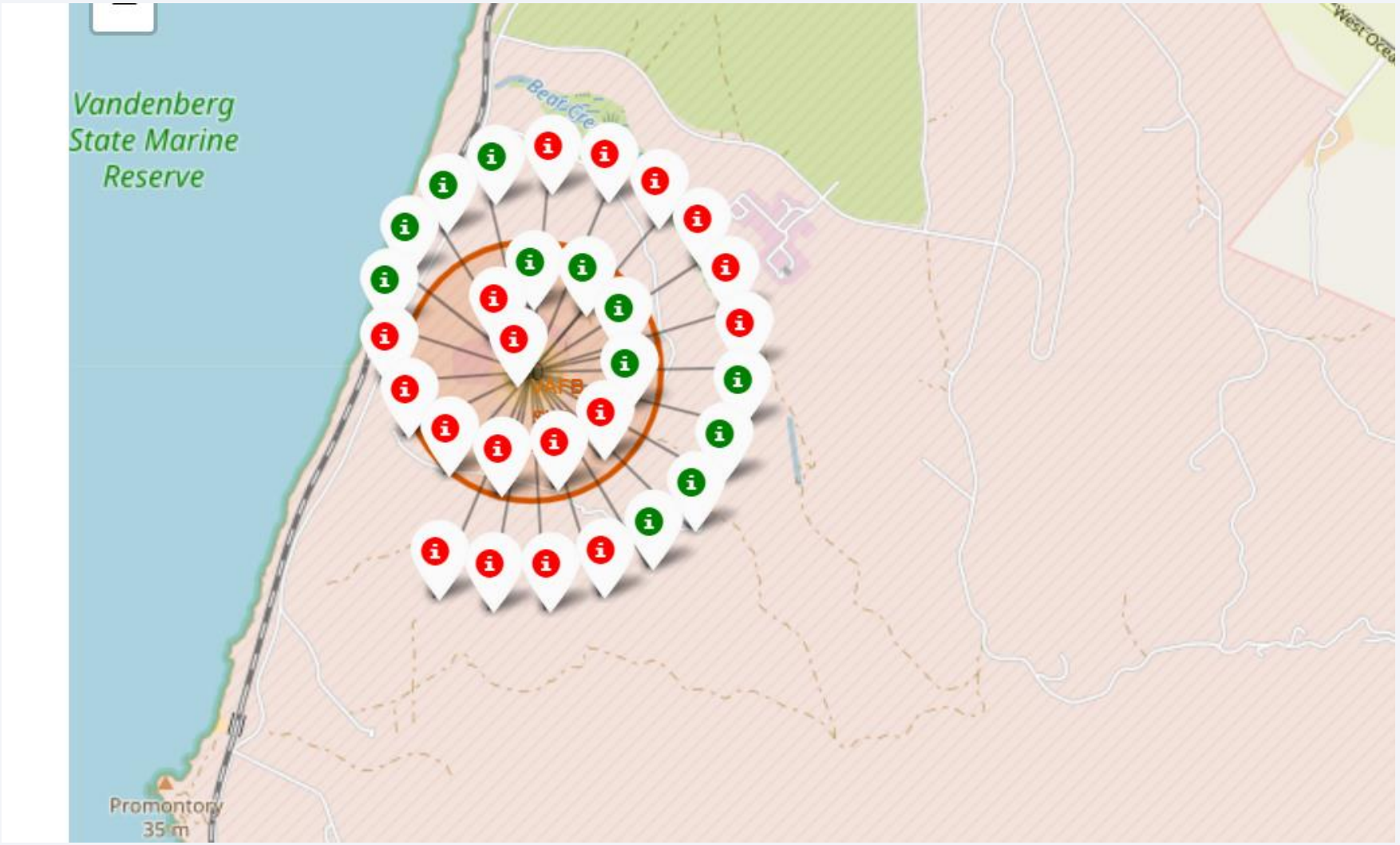
Folium Map Showing all Launch Sites

- This map indicates the four launch sites



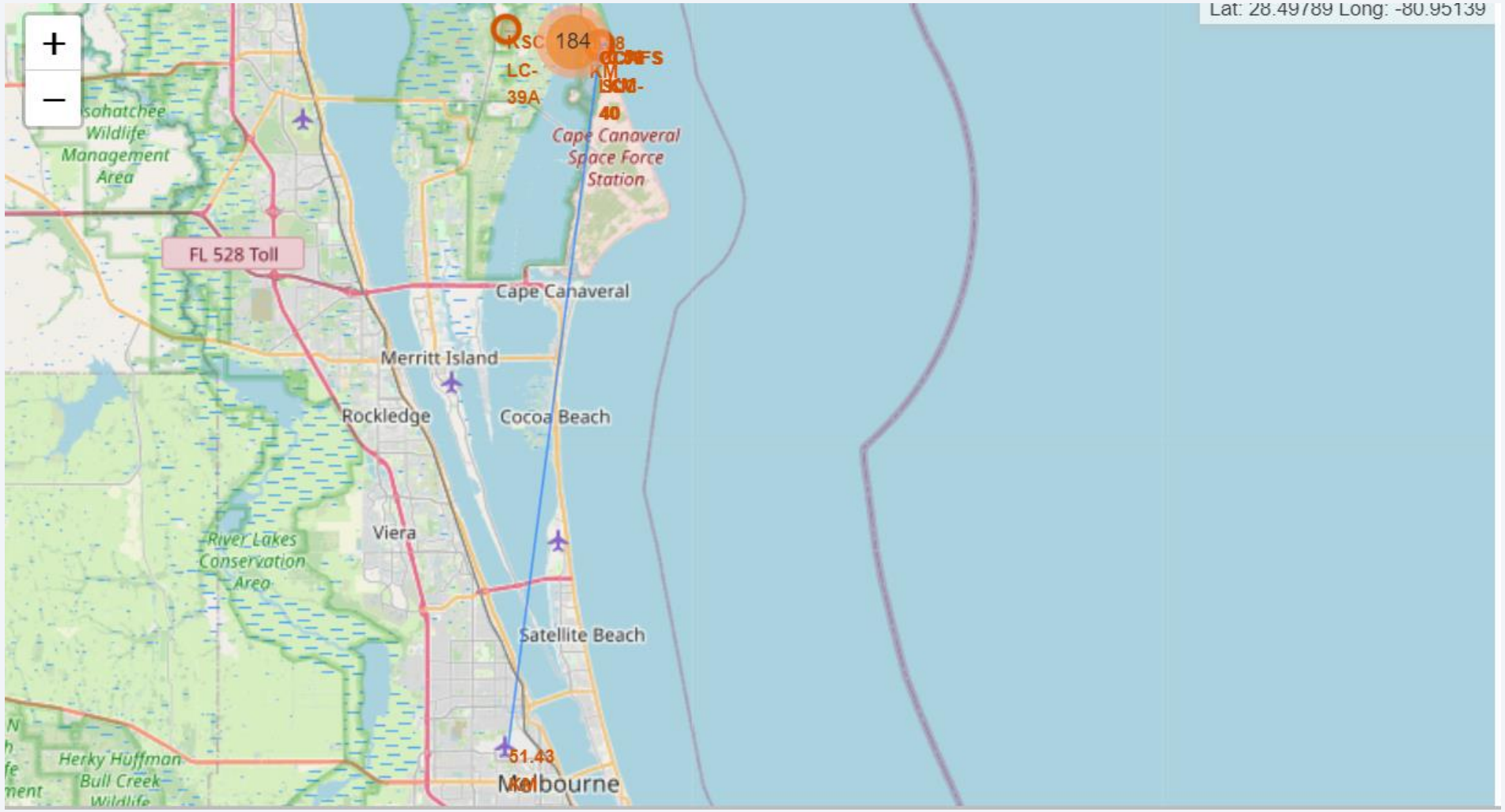
Folium Map of Launch Outcomes at Vandenberg

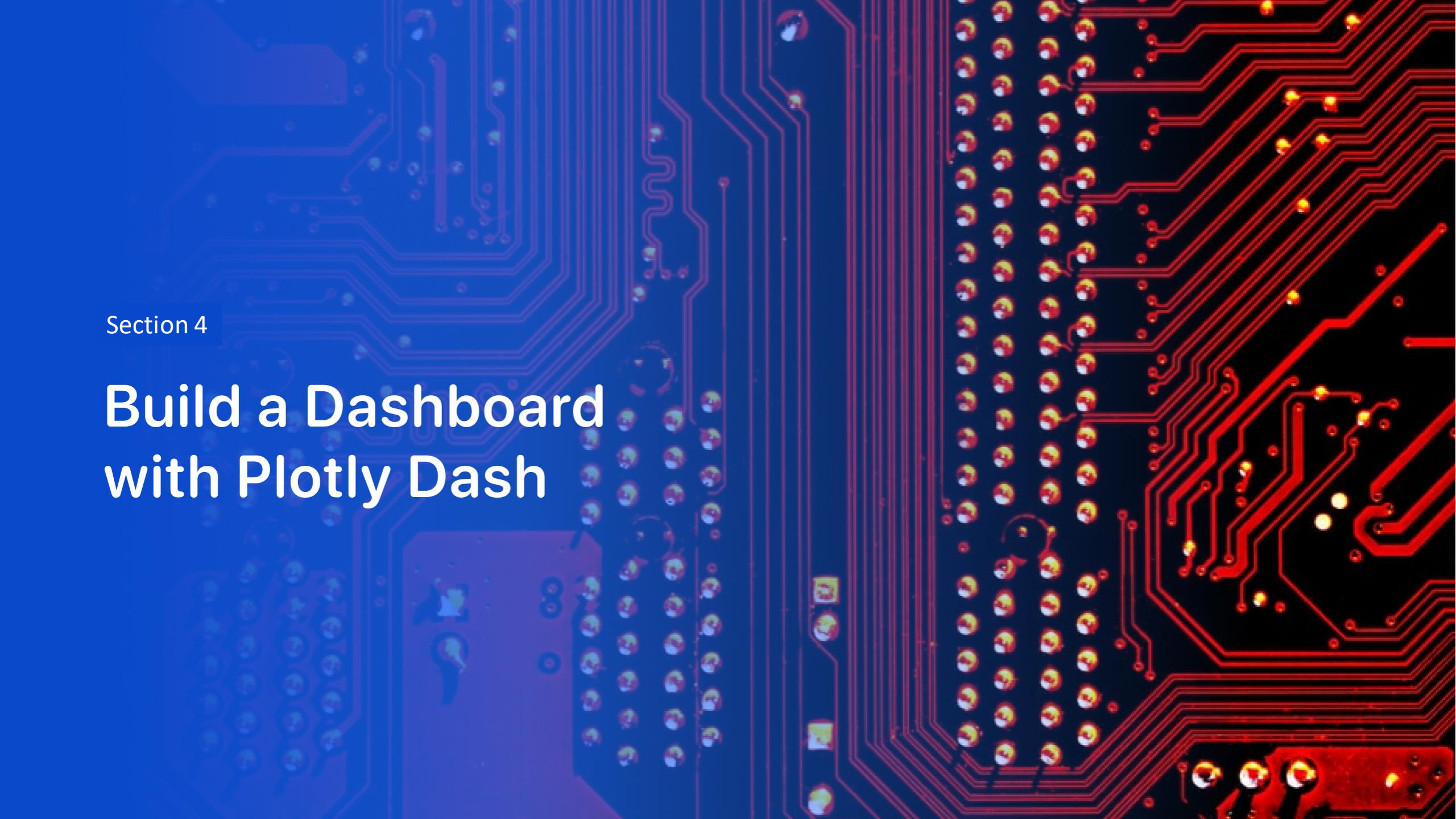
- This map shows the color labelled launch outcomes for Vandenberg.



Folium Map 3: Distance lines

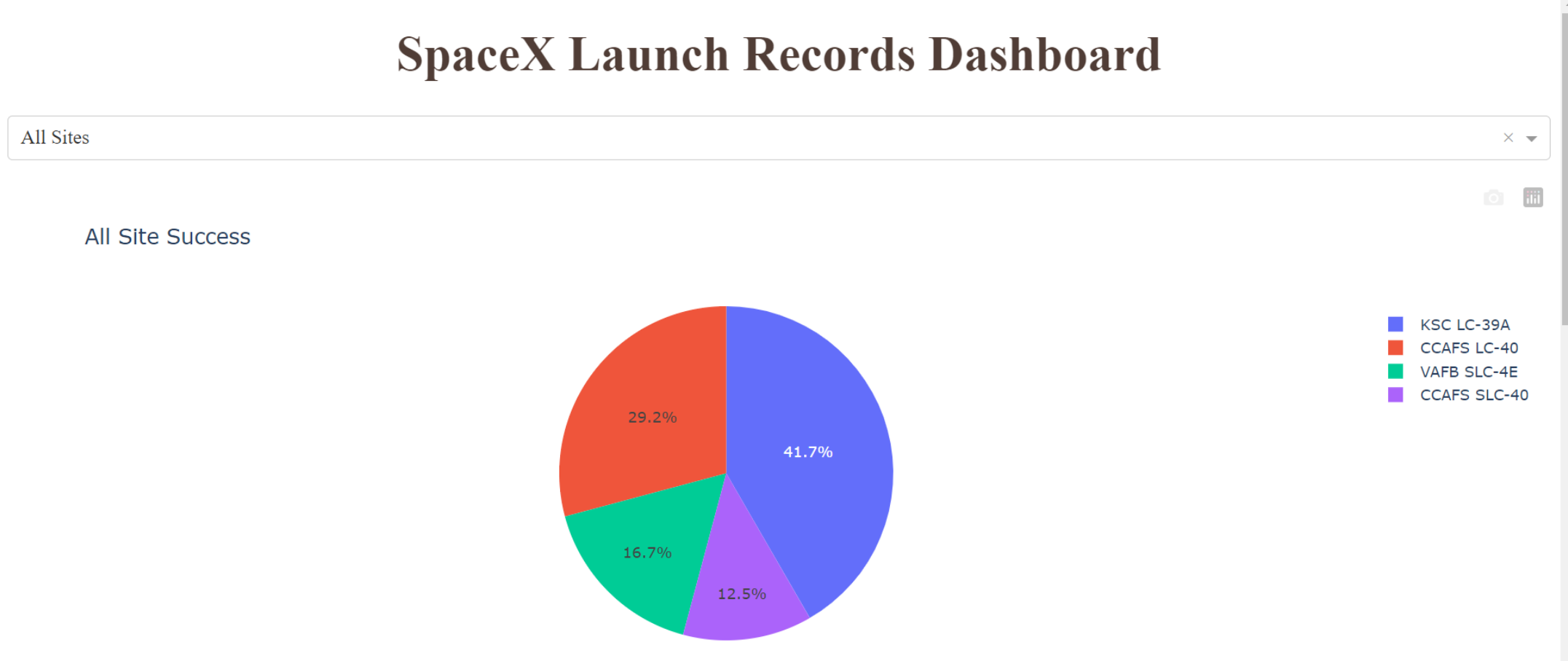
This map shows the nearest city, railroad, and road to the CCAFS SLC-40 launch site



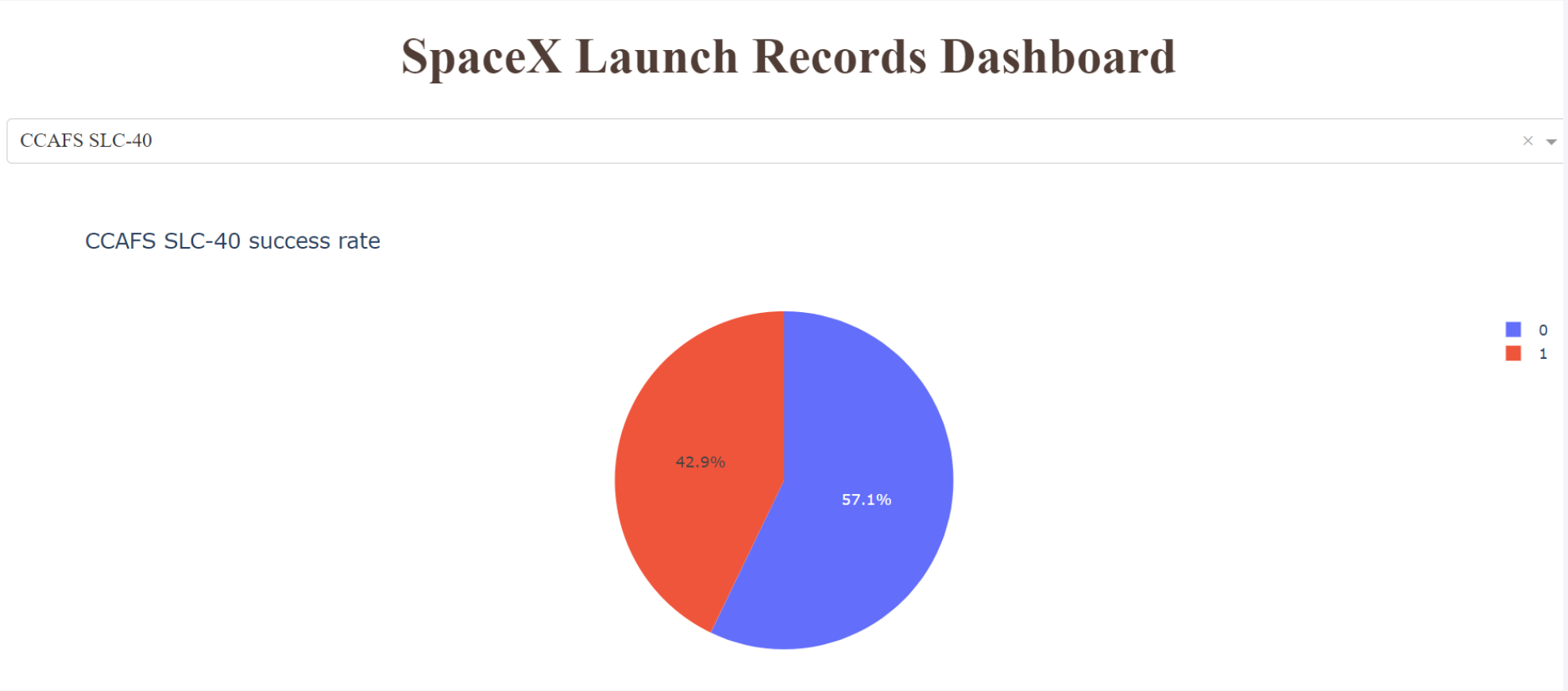

Section 4
Build a Dashboard with Plotly Dash

Pie Chart of All Launch Successes

- This figure shows the percent of successful launches out of each site. KSC LC-39A has the most successful launches

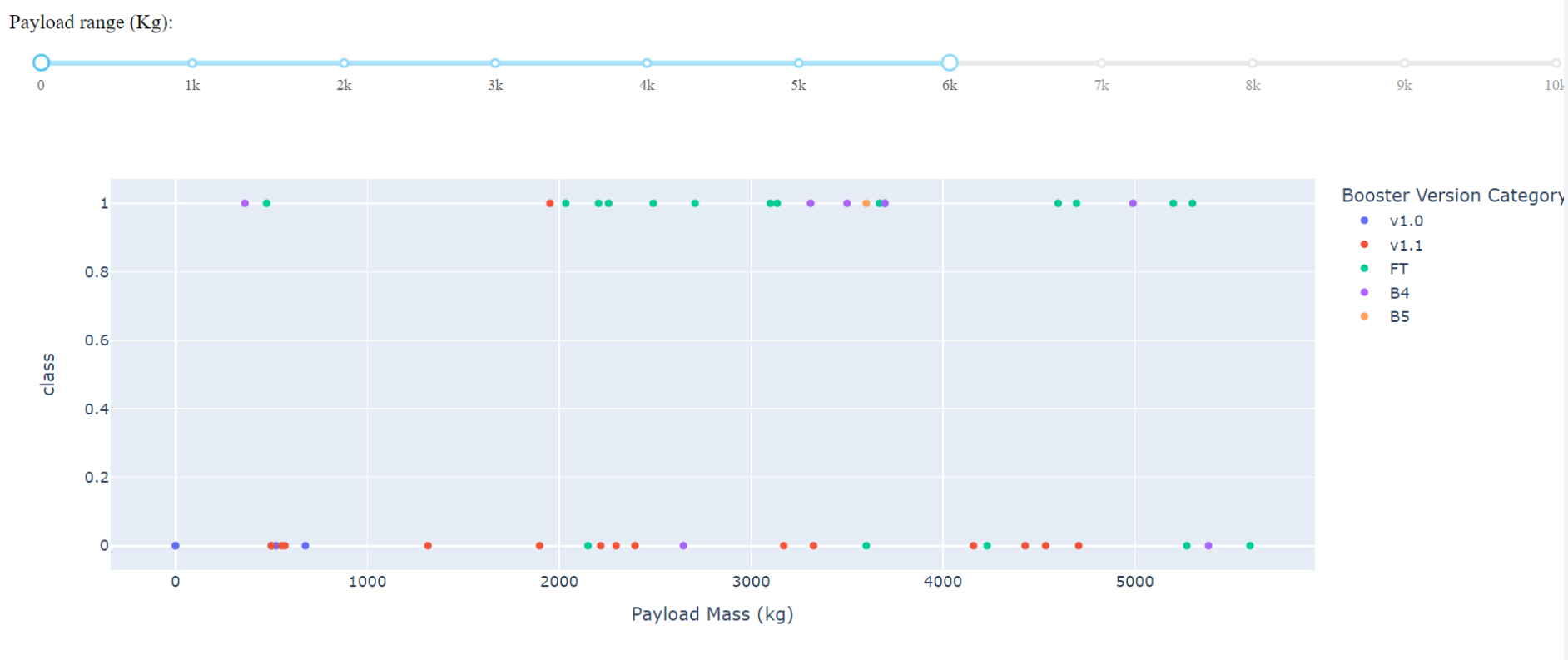


Highest Launch Success Ratio Site



This figure shows the site with the highest success ratio, with 42% of launches being successful

<Dashboard Screenshot 3>



- This figure shows the relationship between payload mass, booster version and success rate. Booster V1.1 has a low success rate

Section 5

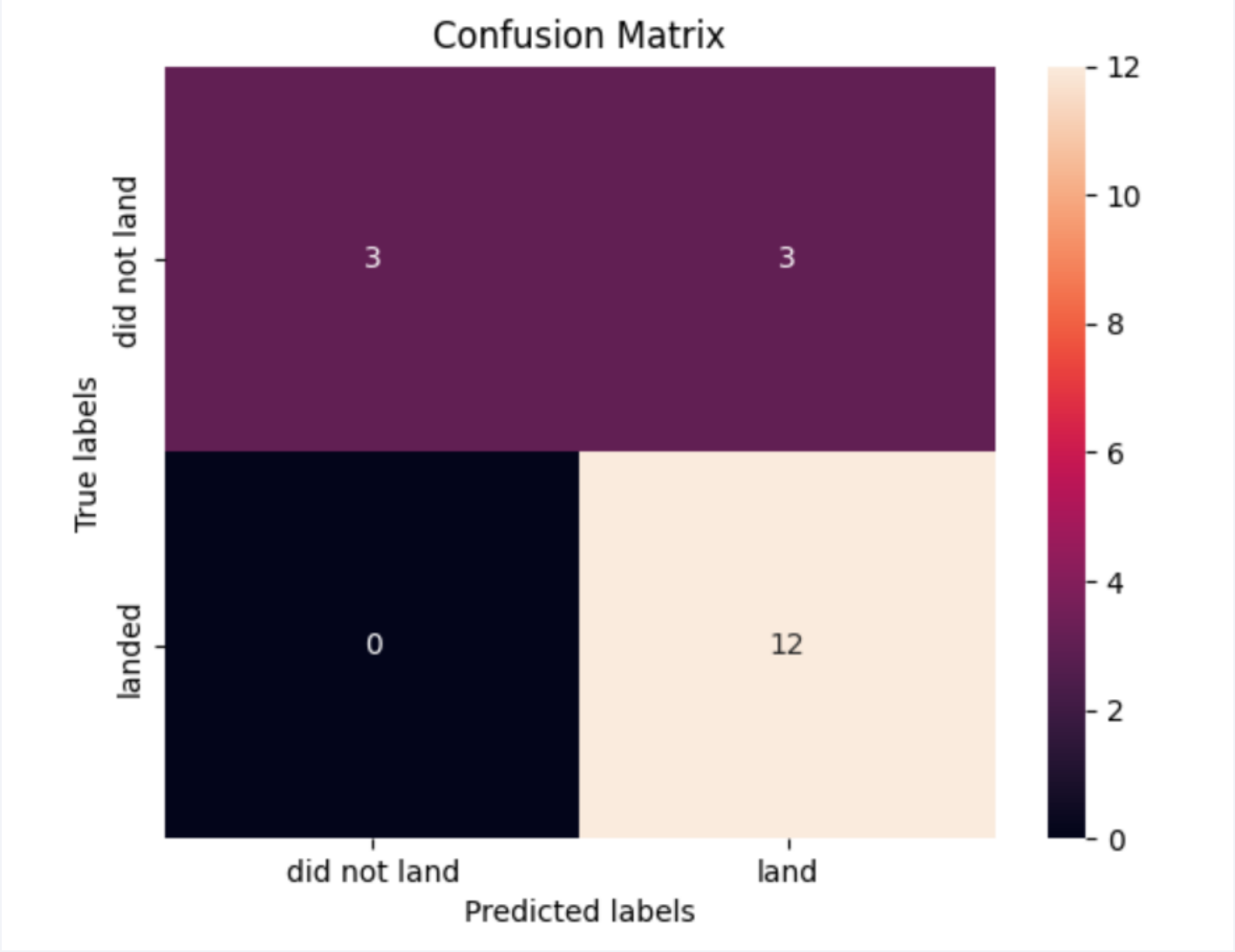
Predictive Analysis (Classification)

Classification Accuracy

- LR, SVM and KNN models all had equally high-test accuracy, at 83.3%
- Decision Trees over fit the training data, and tested at 77% accuracy

Confusion Matrix

The confusion matrix shows that of the 18 test samples, the model correctly classified 15, and missclassified 3, with 83% accuracy



Conclusions

- Launch success increases with practice
- Small payloads
- For orbits like GEO and SSO, there is insufficient data to predict success rates
- As payload mass increases success rates go up for orbits POLAR, LEO, and ISS
- All the early flights occurred with low payload mass, at the CCAFS SLC-40 launch site which falsely indicates low payload has lower success rate, and weights the launch site success rate un-realistically
- Orbits with few flights are likely to have a lower success rate
- While the accuracy of the predictive model is not too bad, I would not bet my company on it yet

Appendix

- URL of the github repo with all my project files:

- https://github.com/jlonergan35/python_capstone

Thank you!