IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Louis Baker
02/10/24

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection through API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with SQL, Pandas, and Matplotlib
  - Interactive Visual Analytics with Folium and Plotly Dash
  - Predictive Analysis
- **Summary of all results**
  - Results of Exploratory Data Analysis, Interactive Analytics, and Predictive Analysis (Logistic Regression, SVM, Decision Tree, KNN)

Introduction

- **Project background and context**

  This project uses rocket launch data from the aerospace company SpaceX. SpaceX uses a multi-stage rocket design; the first stage can be recovered and reused under the proper circumstances, enabling SpaceX to operate at much lower costs than competitors. At SpaceY, we aim to harness the power of data science to gather insight from SpaceX launch data to better predict if the first stage can be recovered.

- **Problems you want to find answers**
  - Can we use data science to predict a successful landing?
  - Which factors determine a successful landing?

Section 1

Methodology

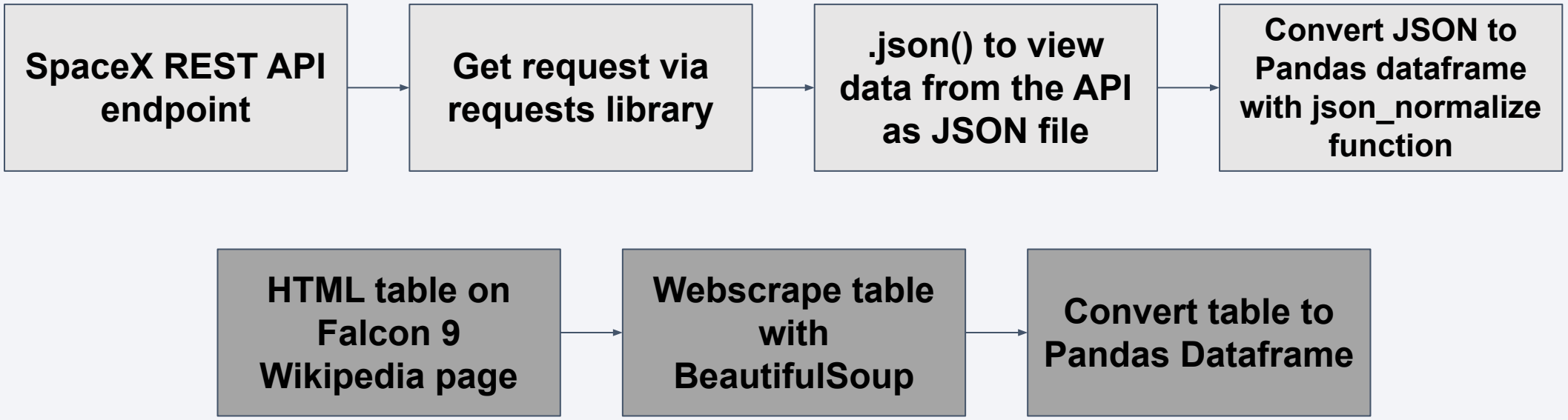
Methodology

Executive Summary

- Data collection methodology:
  - Data was gathered with the SpaceX REST API and Wikipedia
- Perform data wrangling
  - Data was converted to a Pandas dataframe
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine learning prediction models (Logistic Regression, SVM, Decision Tree, KNN)

Data Collection

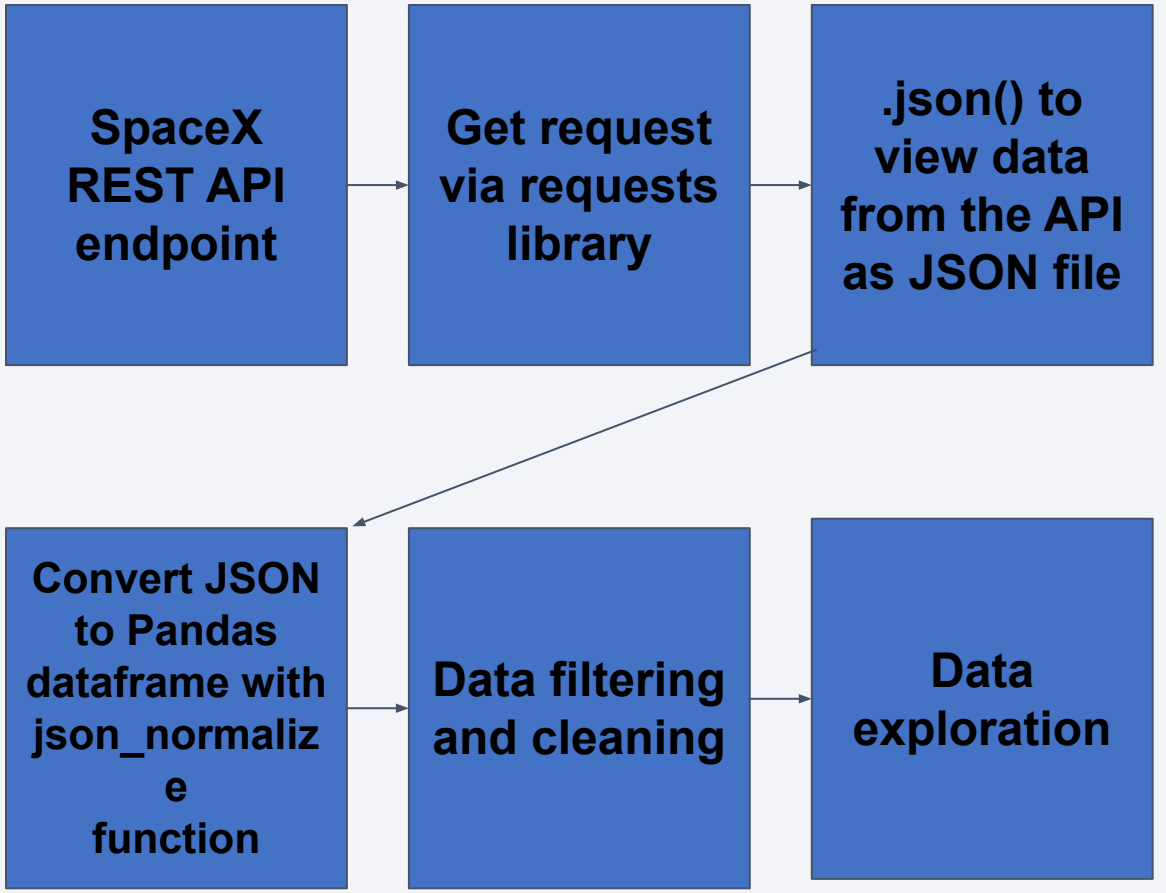
- Data was collected using the SpaceX REST API. BeautifulSoup was also used to webscrape Falcon 9 Launch Data from HTML tables.



Data Collection – SpaceX API

- Data is loaded into a Pandas dataframe, enabling us to explore the data.

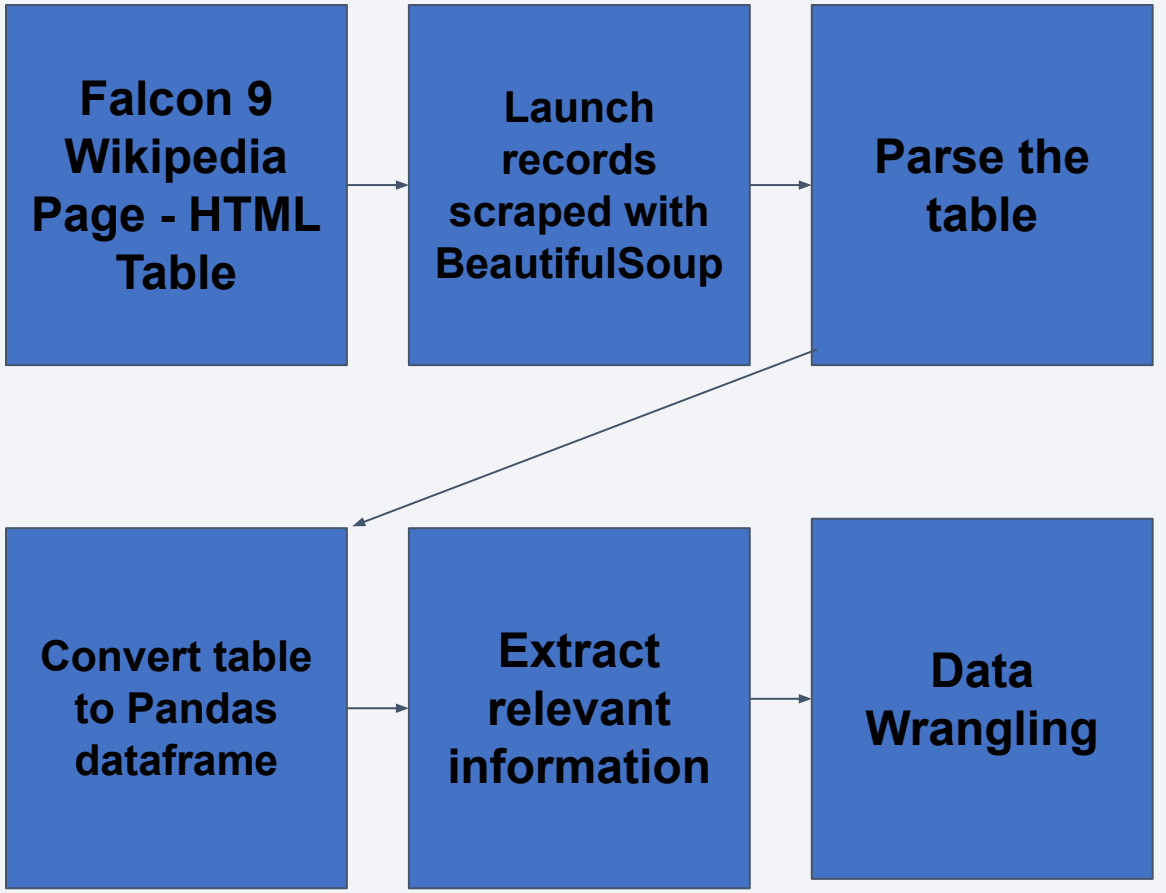
- The notebook can be viewed here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb



Data Collection - Scraping

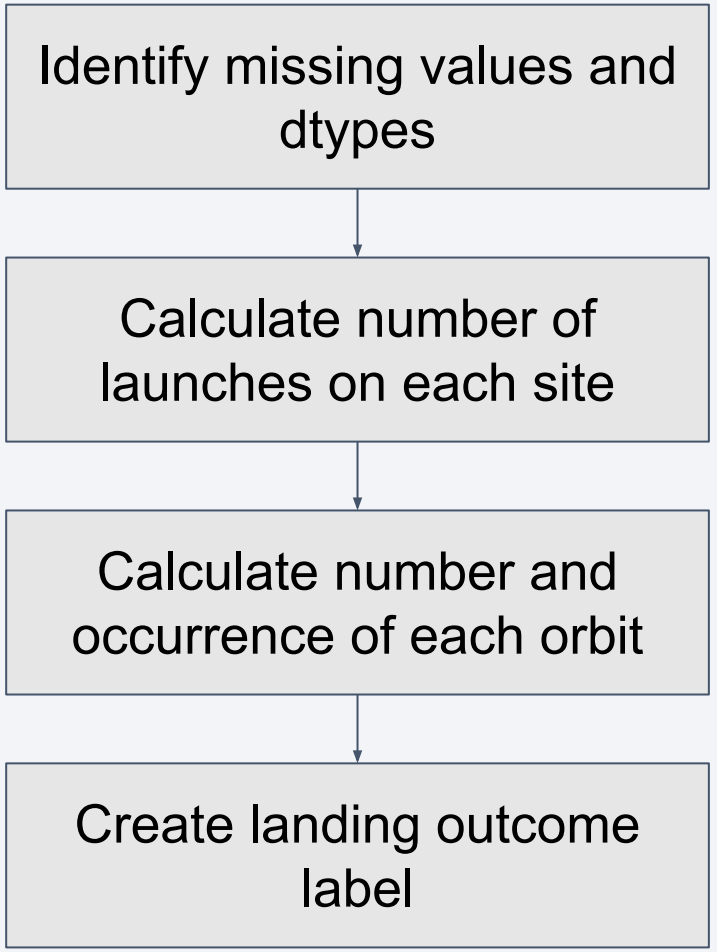
- Data is scraped from the Wikipedia page, parsed, and converted to a Pandas dataframe.

- Notebook can be viewed here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/jupyter-labs-webscraping.ipynb



Data Wrangling

- The Falcon 9 launch dataset, now in Pandas dataframe format, was utilized to calculate the launch count from each site and the frequency of each orbit type (Low Earth Orbit, Geosynchronous Orbit, etc). A new column named ‘Class’ was added to categorize launch outcomes as either good or bad based on the values in the ‘Outcome’ column.
- You need to present your data wrangling process using key phrases and flowcharts
- View the notebook here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- **Catplots** of Flight Number vs Payload Mass, Flight Number vs Launch Site, Flight Number vs Orbit, Payload Mass vs Orbit and Payload Mass vs Launch Site to determine relationships between variables (Hue set to Class)
- **Bar Chart** to determine success rate of each Orbit type
- **Line Chart** to determine average success rate by year

- Notebook here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- **SQL queries were performed on the data:**
  - Display names of the unique launch sites
  - Display 5 records where launch sites begin with ‘CCA’
  - Display the total payload mass carried by boosters launched by NASA
  - Display average payload mass carried by booster version F9 v1.1
  - List the data of the first successful landing outcome
  - List the names of boosters which have success in drone ship and have payload mass between 4000 and 6000
  - Total number of successful and failure mission outcomes
  - Names of the booster versions that carried the max payload mass
  - Records which display month names, failure landing outcomes (drone ship), booster versions, and launch site in 2015
  - Count of landing outcomes between 2010-06-04 and 2017-03-20

- https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL (cont.)

- **SQL queries were performed on the data:**
  - select distinct "Launch_Site" from SPACEXTABLE
  - select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
  - select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" like "NASA%"
  - select round(avg(PAYLOAD_MASS__KG_), 2) from SPACEXTABLE where "Booster_Version" like "F9 v1.1%"
  - select min("Date") from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"
  - select distinct "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and PAYLOAD_MASS__KG_ between 4000 and 6000
  - select "Mission_Outcome", count(*) from SPACEXTABLE where "Mission_Outcome" like "Success%" or "Mission_Outcome" like "Failure%" group by "Mission_Outcome"
  - select "Booster_Version" from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
  - select (substr(Date, 6,2)), "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where (substr(Date,0,5)='2015') and ("Landing_Outcome" = "Failure (drone ship)")
  - select "Landing_Outcome", count("Landing_Outcome") from SPACEXTABLE where ("Date" between "2010-06-04" and "2017-03-20") group by "Landing_Outcome" order by (count("Landing_Outcome")) DESC

Build an Interactive Map with Folium

- **folium.Circle** to add a circle at each launch site location
- **folium.Marker** for each launch result and closest coastlines to launch sites
- **MarkerCluster** object to cluster launch results together at the correct launch site
- **MousePosition** to display coordinates at mouse location on map
- **PolyLine** from launch site to closest coastline/railroad/highway
- Link here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/lab_jupyter_launch_site_location.jupyterlite.ipynb

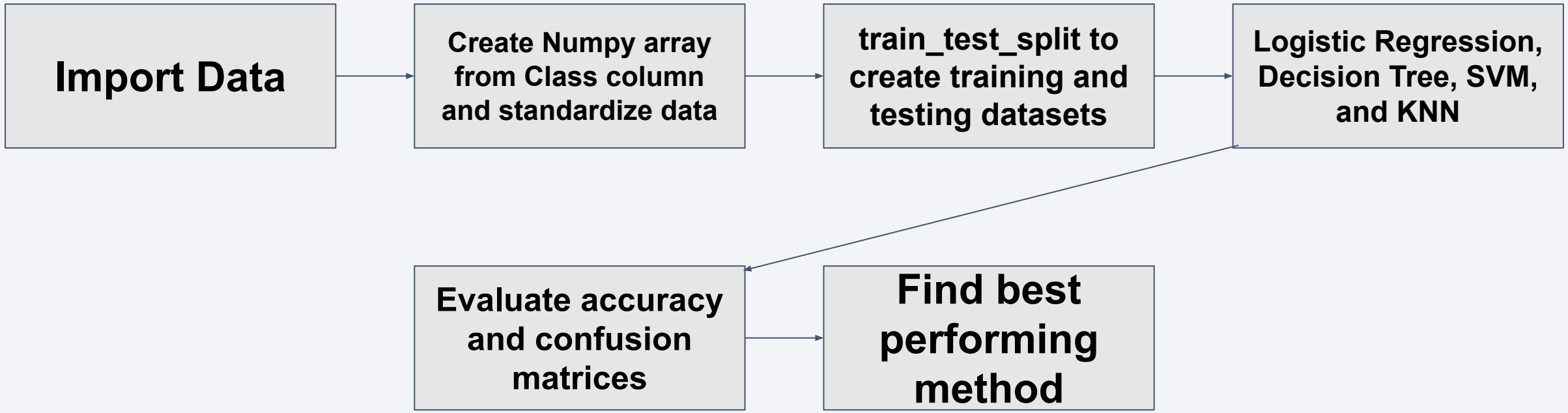
Build a Dashboard with Plotly Dash

- **Pie chart** showing the total successful launch count for all sites, and success vs failed counts for specific launch sites
- **Scatter chart** to show the correlation between payload and launch success
- Link here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- A NumPy array was created from the Class column from our dataframe. Data was then standardized and split into training and testing data.
- Hyperparameters were evaluated via Logistic Regression, Decision Tree, SVM, and KNN methods. Accuracy of each method was calculated with the score method, and confusion matrices were plotted for each method.
- Notebook here: https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Predictive Analysis (Classification)



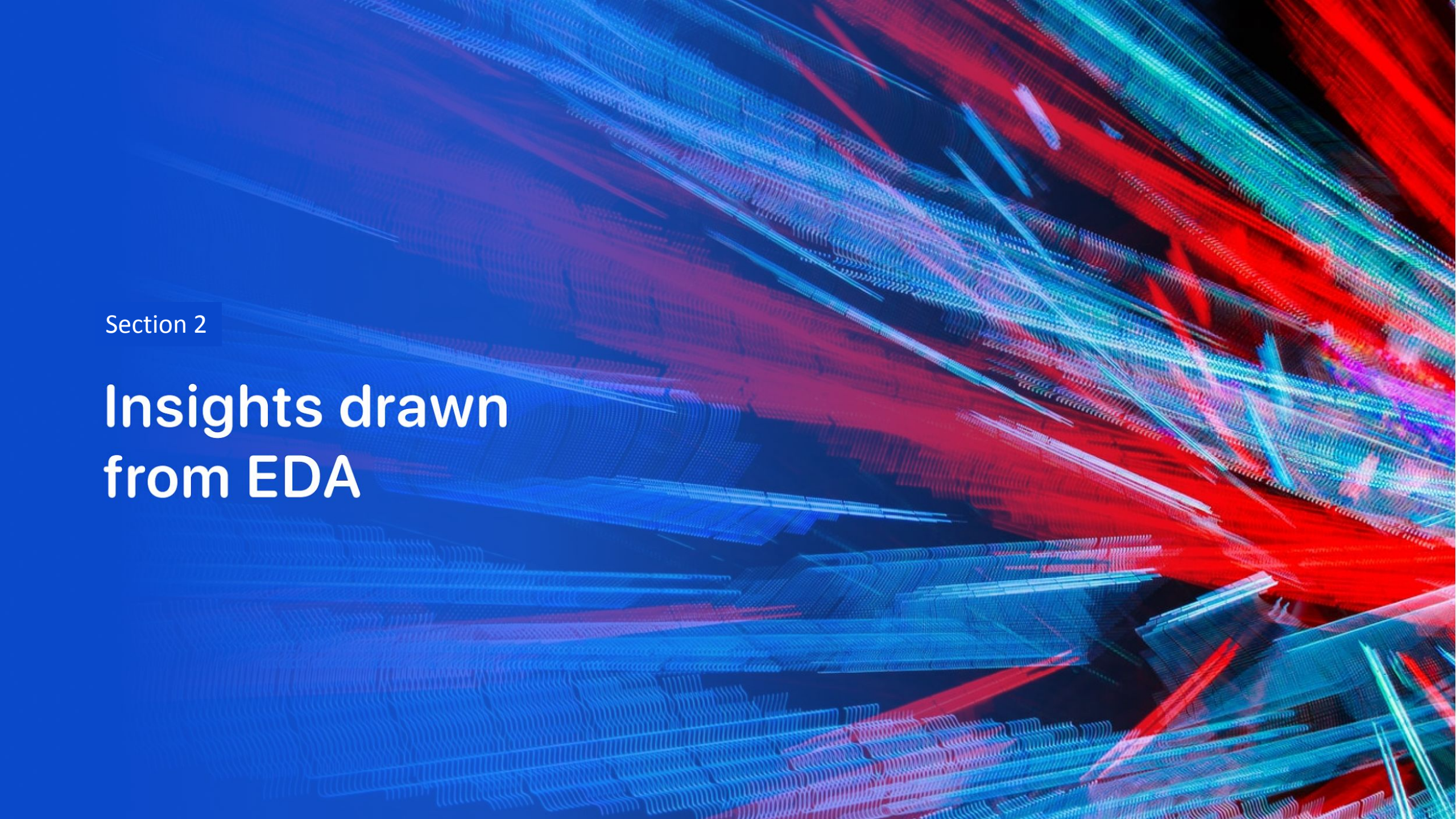
Notebook here:
https://github.com/louisdbaker/IBM-Data-Science-Capstone/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

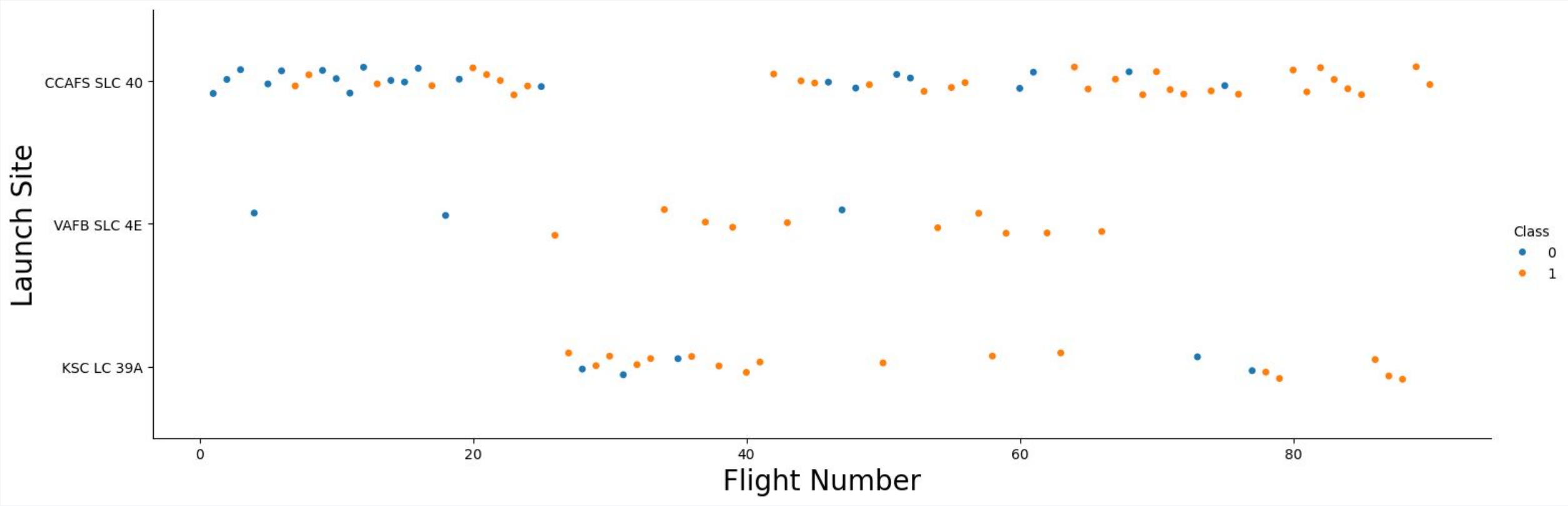
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

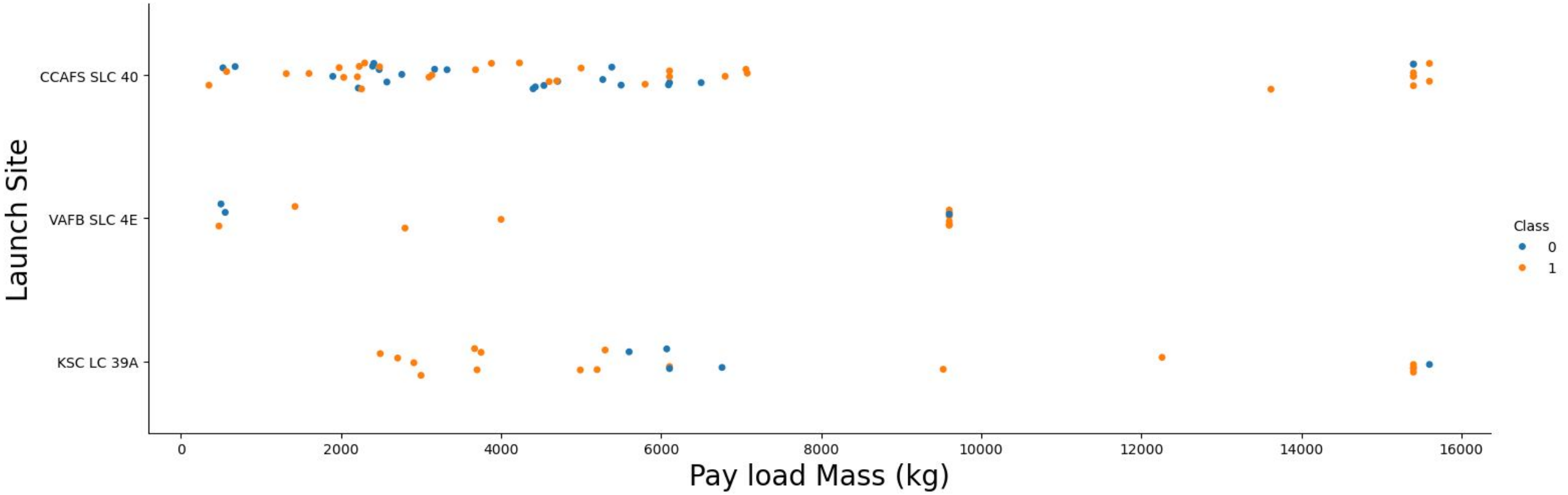


Flight Number vs. Launch Site



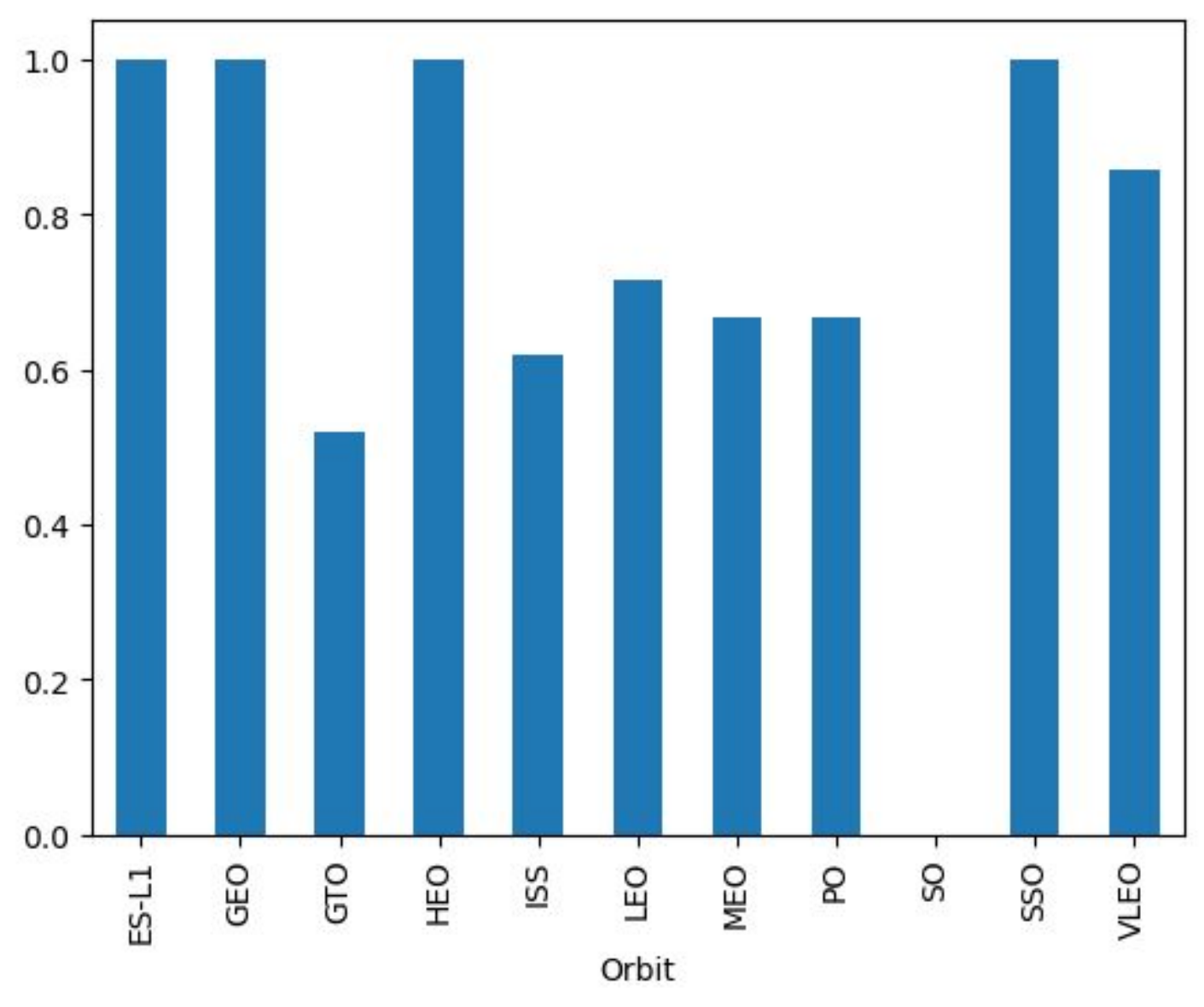
Here we can see the launch outcomes of each launch and their distribution across launch sites. Note how later launches from CCAFS SLC 40 are more likely to be successful than the earlier launches.

Payload vs. Launch Site



Note that there were no rockets launched from VAFB SLC 4E with a payload greater than 10,000 kg. Likewise, there were no light payloads (<2,000 kg) from KSC LC-39A, and a gap in payload size for CCAFS SLC 40.

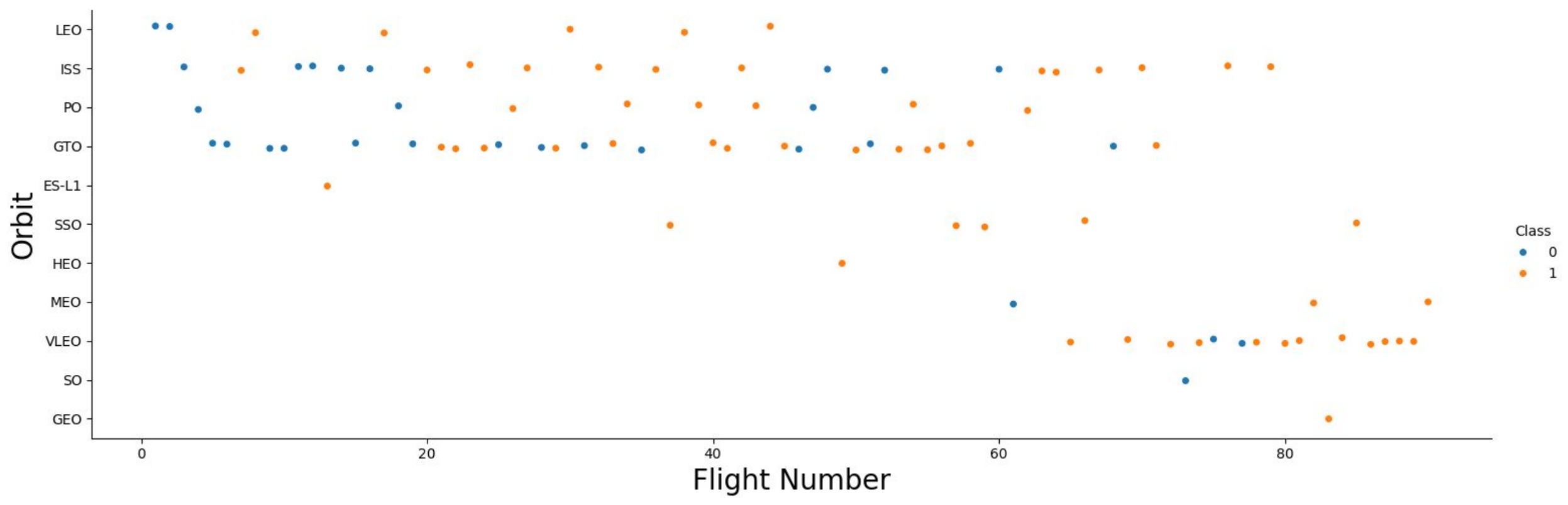
Success Rate vs. Orbit Type



ES-L1, GEO, HEO, and SSO orbit types had a perfect launch success rate.

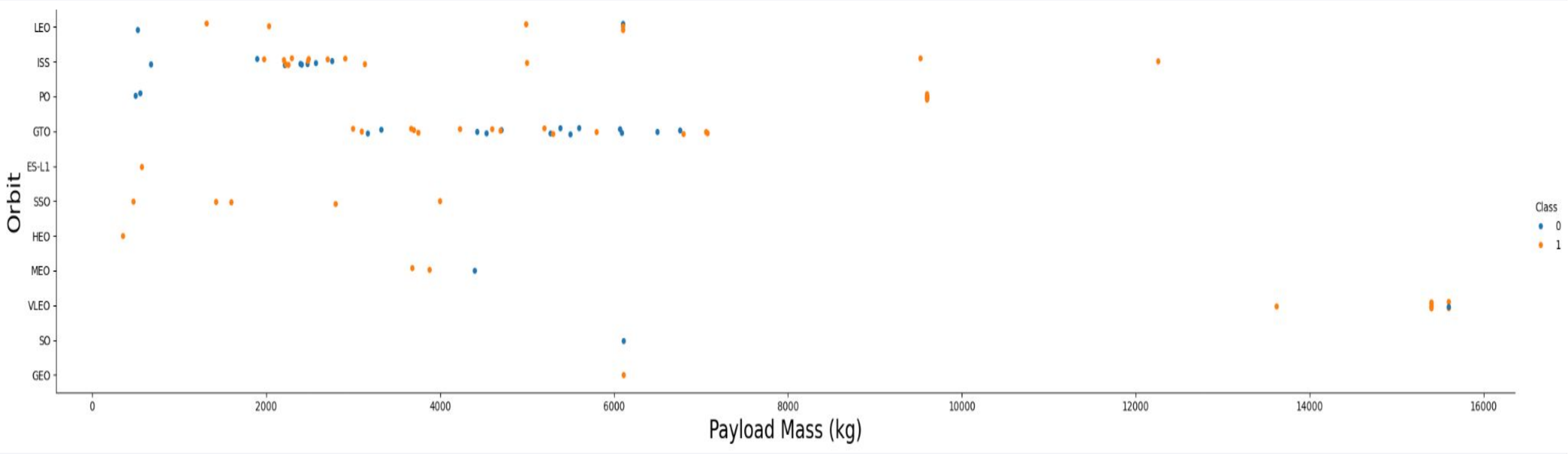
There were no successful launches for the SO orbit type.

Flight Number vs. Orbit Type



This plot further contextualizes the previous slide. Note that ES-L1, GEO, HEO, and SO had only one launch each. When determining success rate, it is important to also consider the number of datapoints available for each orbit type. Note that VLEO, an orbit type with a more comprehensive set of datapoints, has a very high success rate (>0.8).

Payload vs. Orbit Type

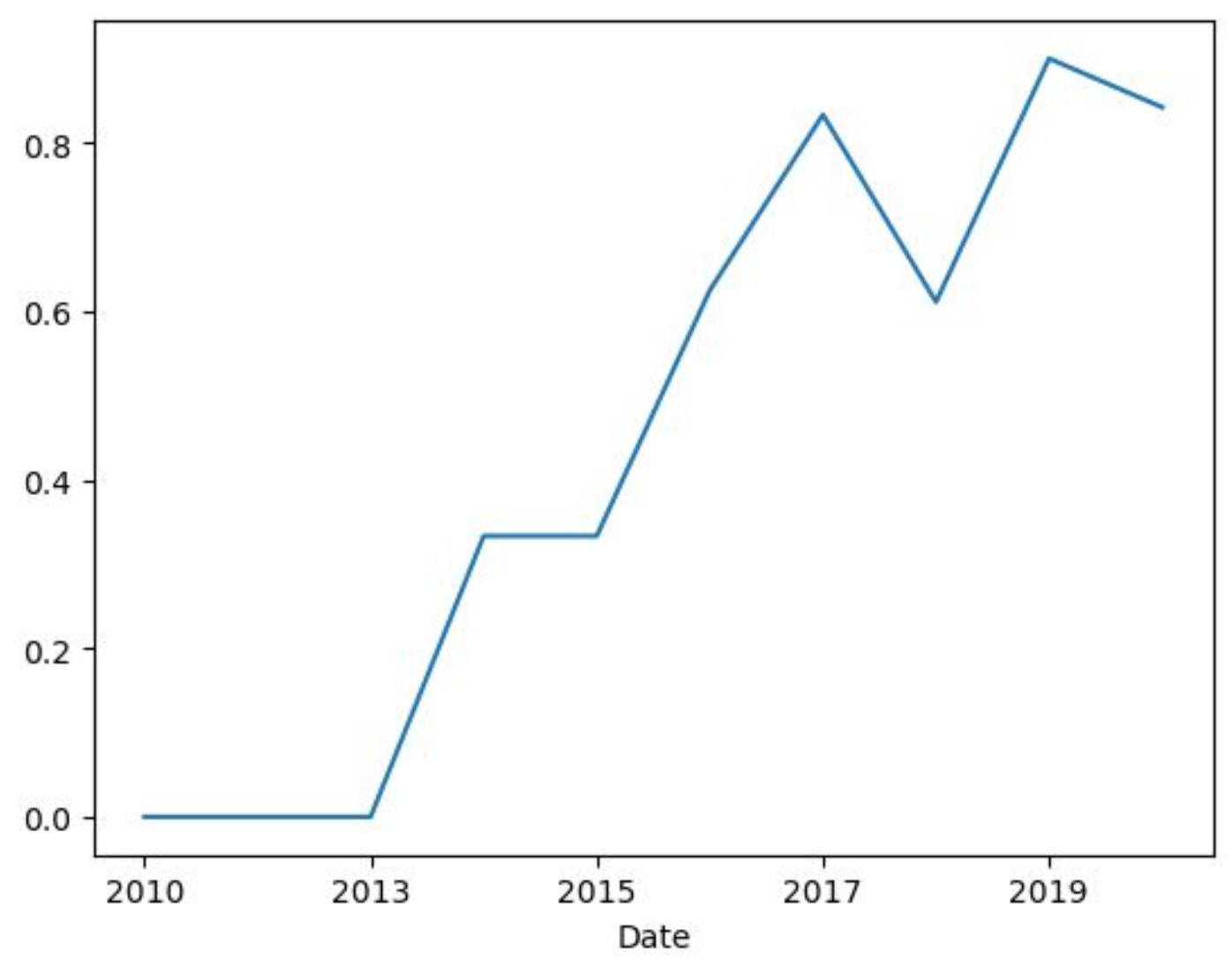


Note that with heavy payloads, we see Polar, LEO and ISS have consistently successful landing rates. VLEO is mostly successful.

ISS and GTO have varying success rates at low to mid payload mass.

Launch Success Yearly Trend

Here we see that as years pass, successful launches are achieved more consistently.



All Launch Site Names

```
%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

This query returns all launch site names in the table. “Distinct” is used to avoid duplicates.

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

- Find 5 records where launch sites begin with `CCA`
- LIKE ‘CCA%’ filters for strings that begin with CCA in the Launch_Site column. LIMIT 5 returns only five records.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- This query returns the total SUM of PAYLOAD_MASS__KG_ where the value in the Customer column includes “NASA”

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

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

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

sum(PAYLOAD_MASS__KG_)
99980

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- This query uses AVG to find the average payload mass for Booster_Version F9 v1.1.

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

```
%sql select round(avg(PAYLOAD_MASS__KG_), 2) from SPACEXTABLE where "Booster_Version" like "F9 v1.1%"
```

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

round(avg(PAYLOAD_MASS__KG_), 2)
2534.67

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- This query uses MIN to find the earliest date that meets the criteria (Successful landing outcome on ground pad)

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

Hint:Use min function

```
[2]: %sql select min("Date") from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"
```

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

[2]:

min("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- DISTINCT is used to avoid duplicates. AND and BETWEEN are used to specify the desired payload mass.

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_ between 4000 and 6000
```

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

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
- COUNT returns the number of each specific outcome. GROUP BY is used to group by mission outcome.

List the total number of successful and failure mission outcomes

```
%sql select "Mission_Outcome", count(*) from SPACEXTABLE where "Mission_Outcome" like "Success%" or "Mission_Outcome"
like "Failure%" group by "Mission_Outcome"
```

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

- This query uses a subquery to find the maximum payload mass. This is used to find the booster versions that have carried that maximum mass.

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- substr is used to get the desired month and year. The SELECT statement gets the desired columns for that month and year.

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)), "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where (substr(Date,0,5)='2015')
and ("Landing_Outcome" = "Failure (drone ship)")
```

```
 * 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

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
- Returns the COUNT of landing outcomes in the desired date range, then uses GROUP BY, ORDER BY, and DESC to properly organize and order the list.

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.

```
%sql select "Landing_Outcome", count("Landing_Outcome") from SPACEXTABLE where ("Date" between "2010-06-04" and "2017-03-20")
group by "Landing_Outcome" order by (count("Landing_Outcome")) DESC
```

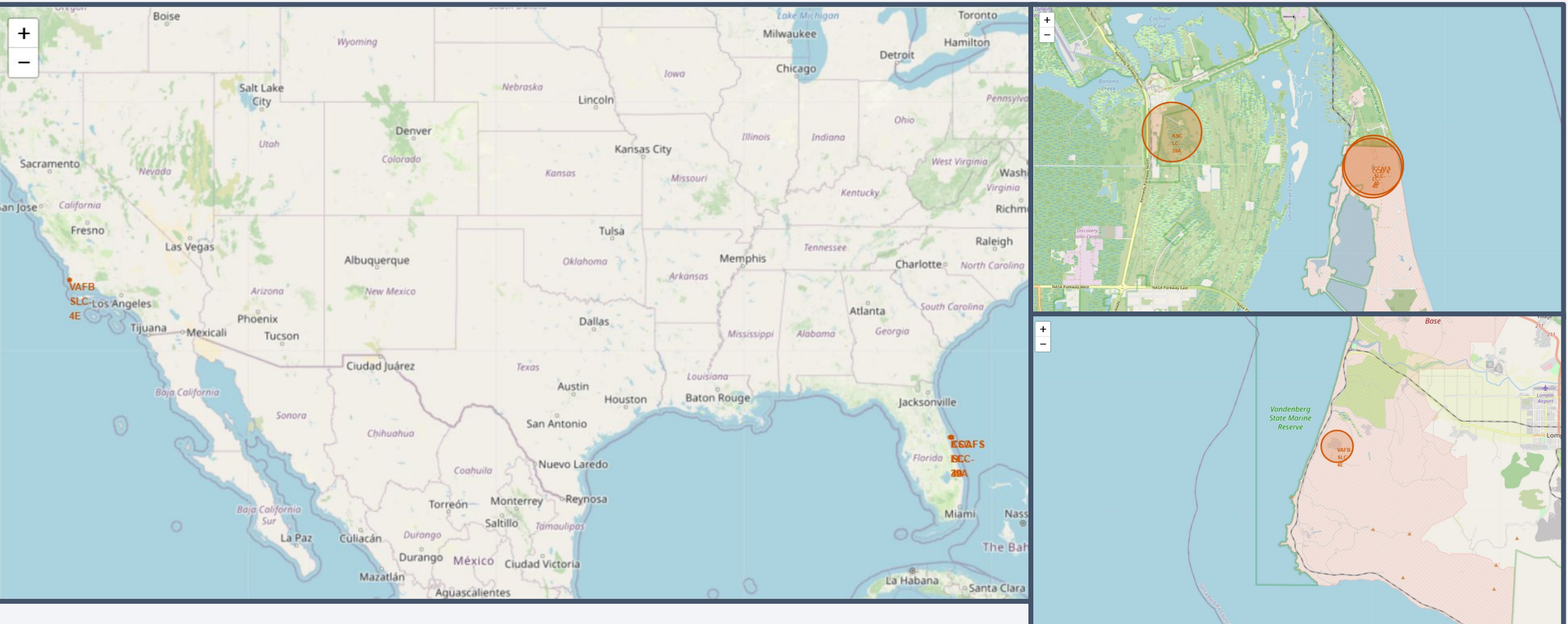
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count("Landing_Outcome")
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

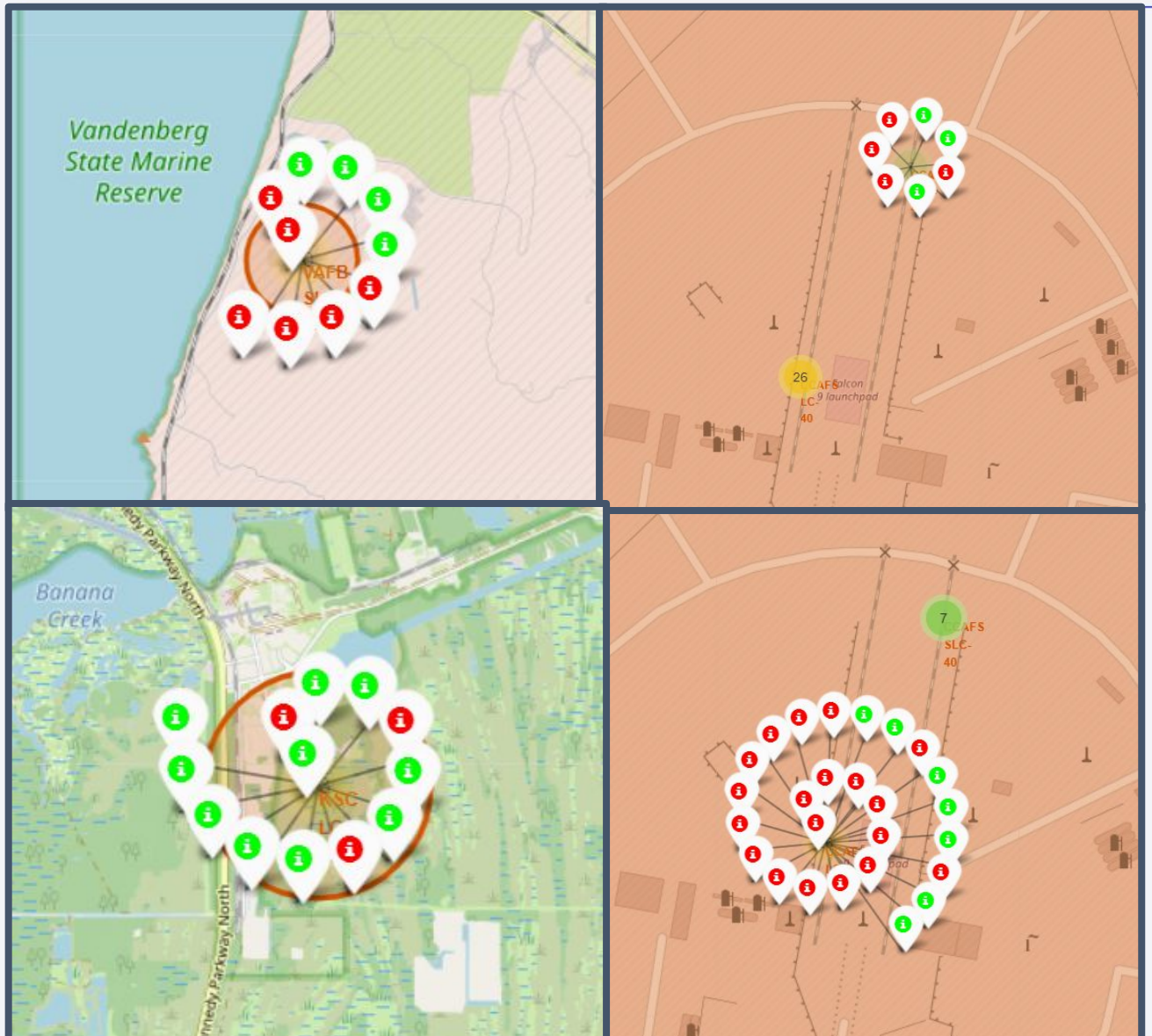

Section 3
Launch Sites
Proximities Analysis

Mark All Launch Sites on Map

Here are the locations of the four launch sites in the dataframe (three on the Florida coast, one on the California coast.



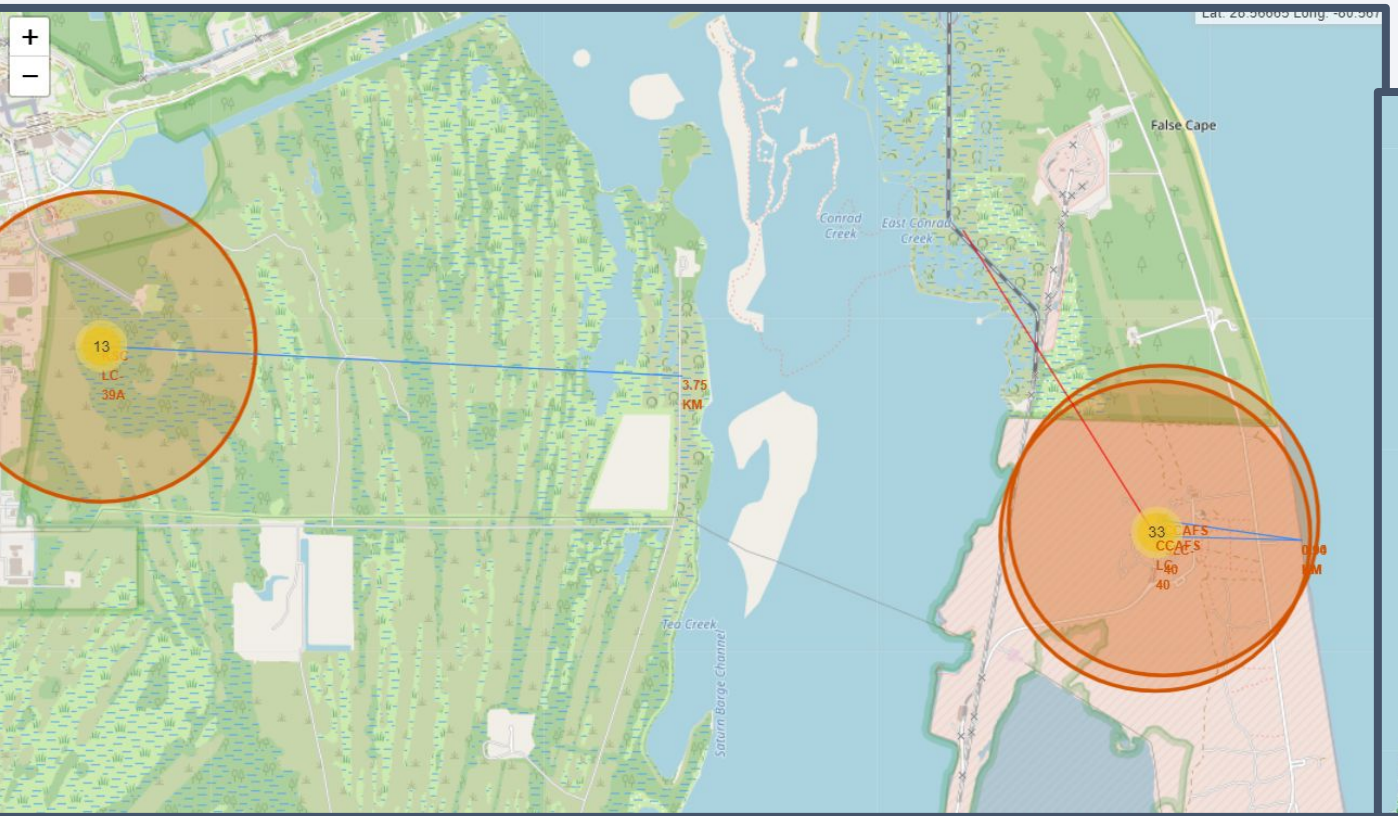
Success/Failed Launches for Each Site

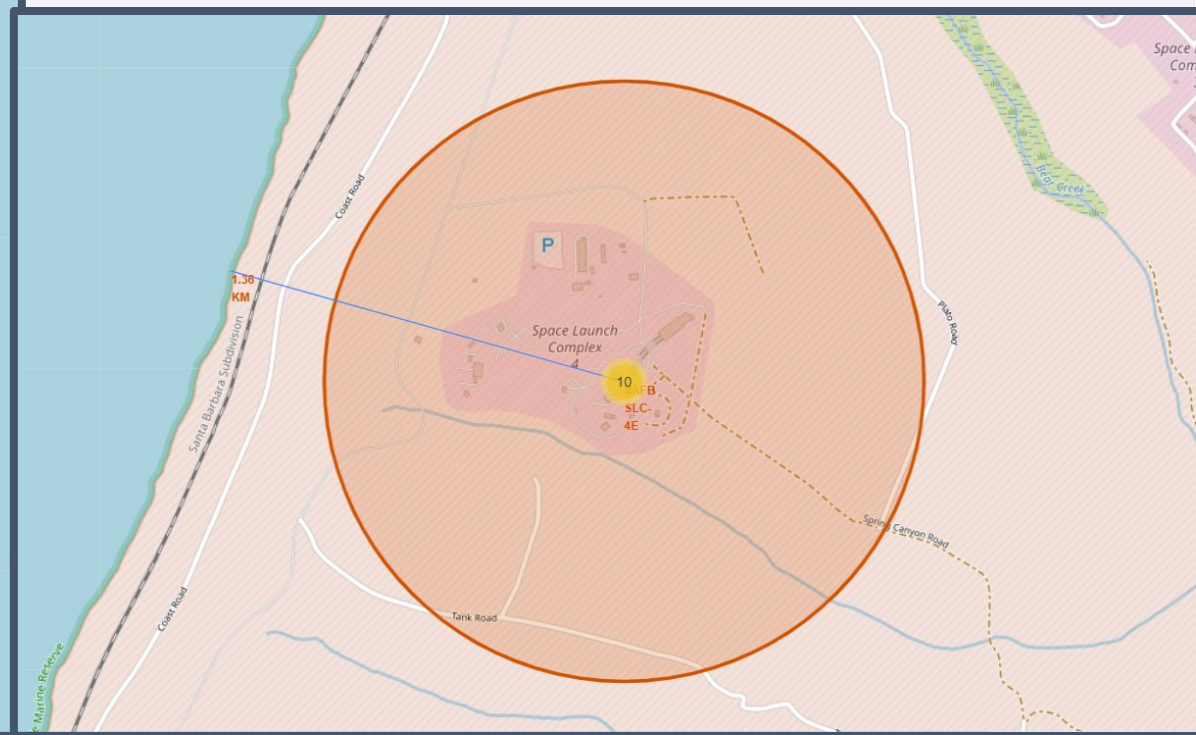


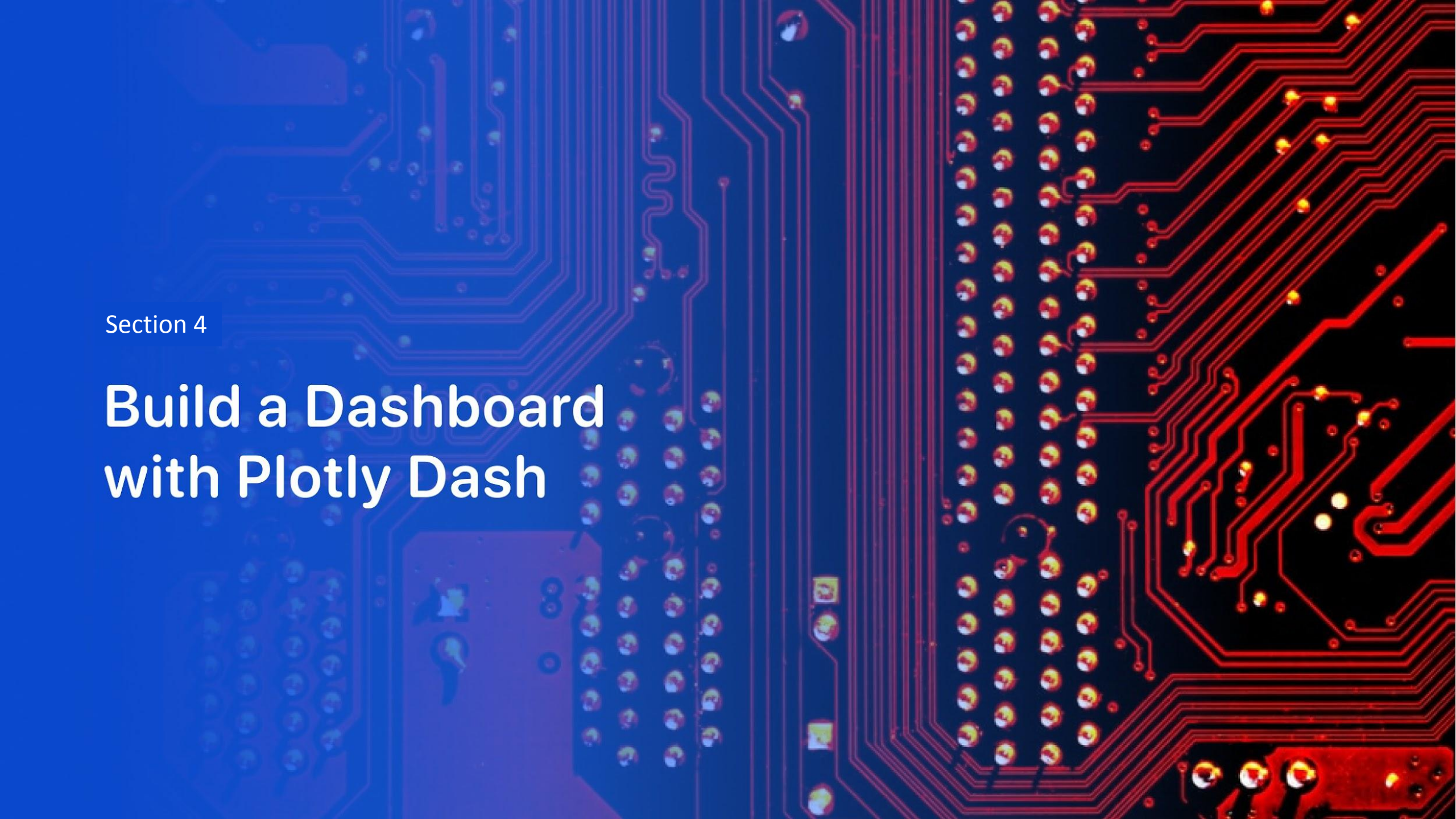
For each launch site, a green marker represents a successful launch while a red marker represents an unsuccessful launch. In this format, we can see the launch outcomes for each individual launch site.

Distances Between Launch Sites and Coasts/Railways

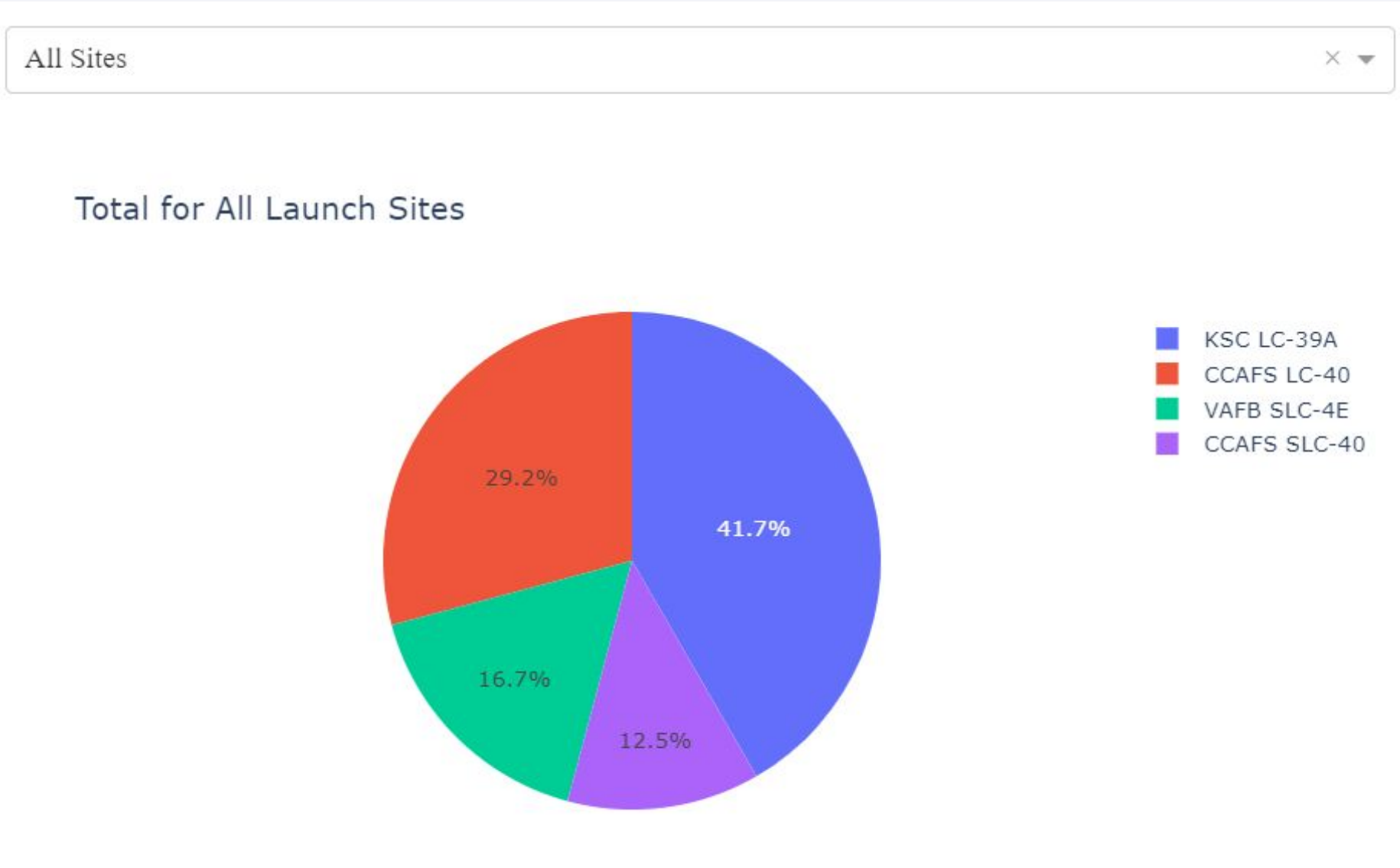
All launch sites are near coastlines (path from site to coast marked with blue PolyLine) and far from major cities. Railways are also visible in both images (path from CCAFS LC-40 to railway marked with red polyline)





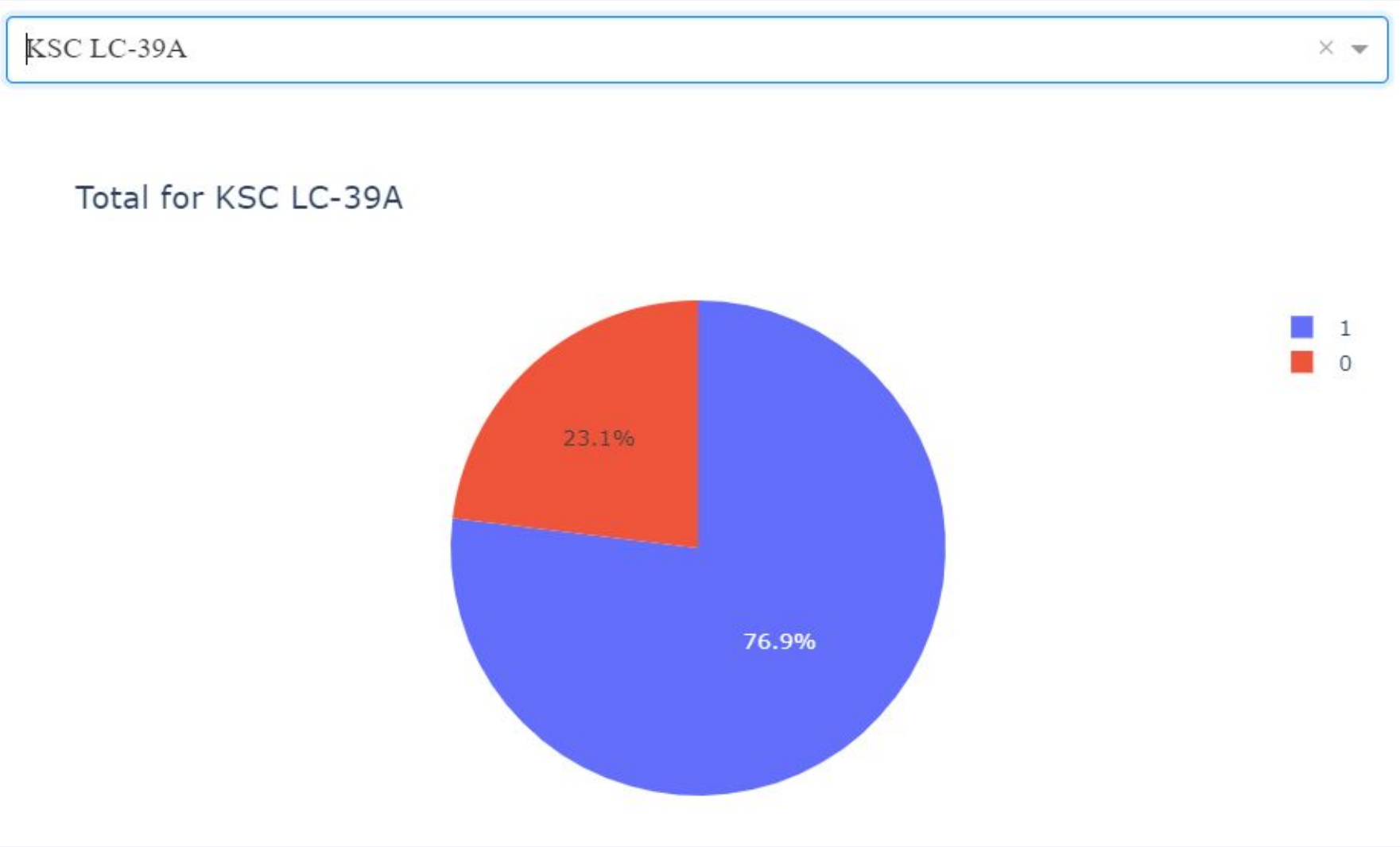

Section 4
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites



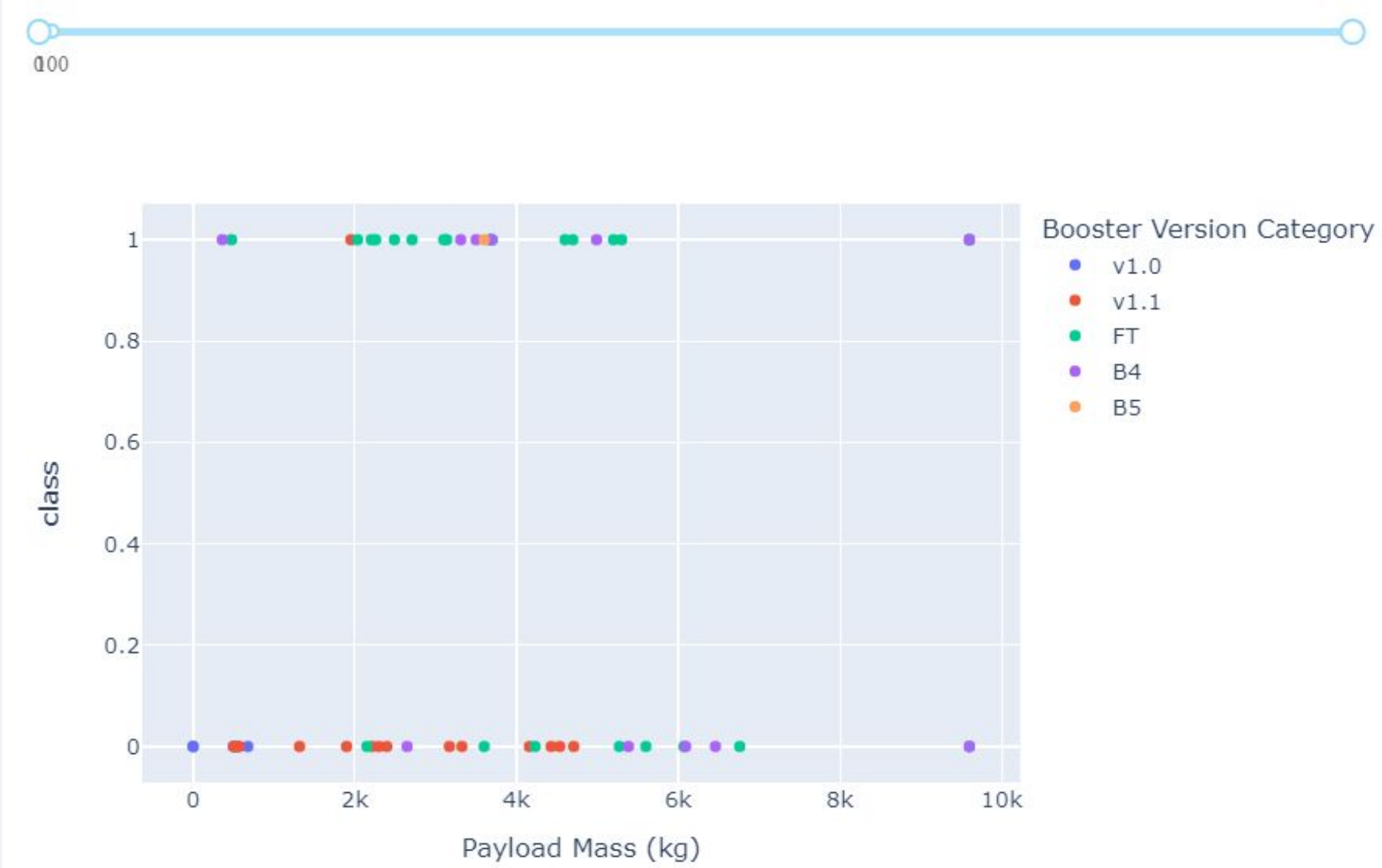
The KSC LC-39A launch site had the most successful launches, followed by CCAFS LC-40, followed by VAFB SLC-4E, followed by CCAFS SLC-40.

Launch Site with the Highest Launch Success Ratio



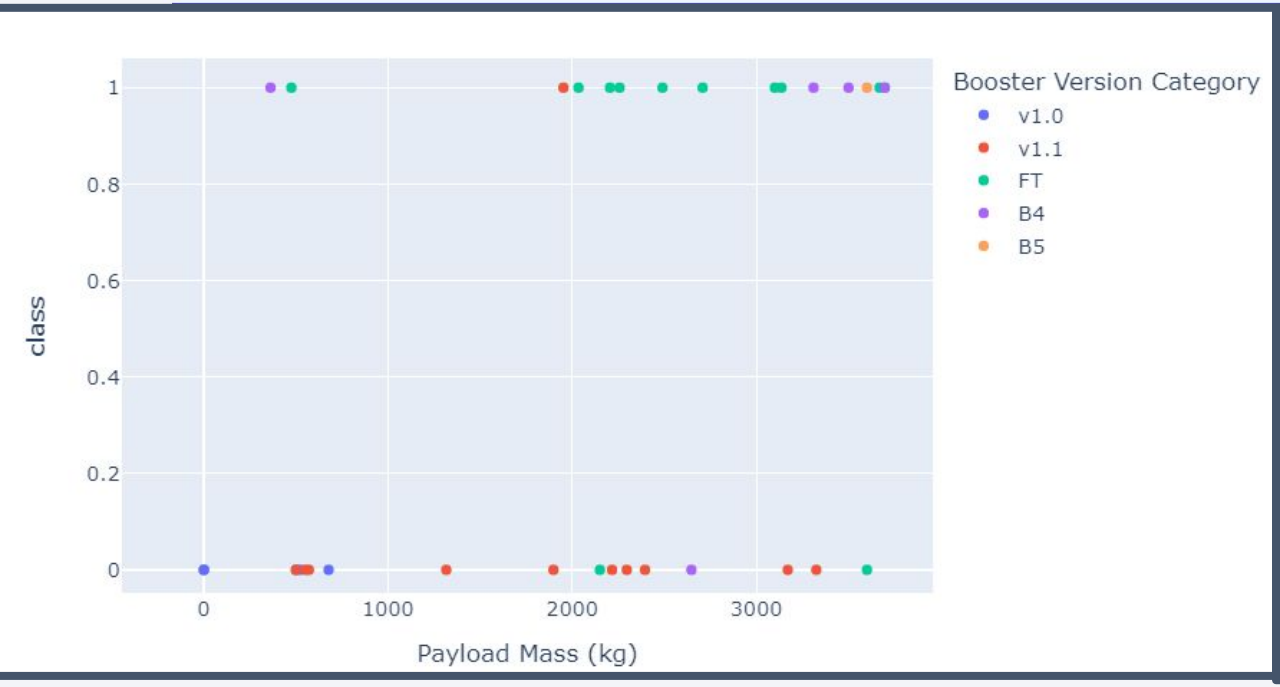
Of each launch site, KSC LC-39A had the highest rate of successful launches (76.9 percent of launches).

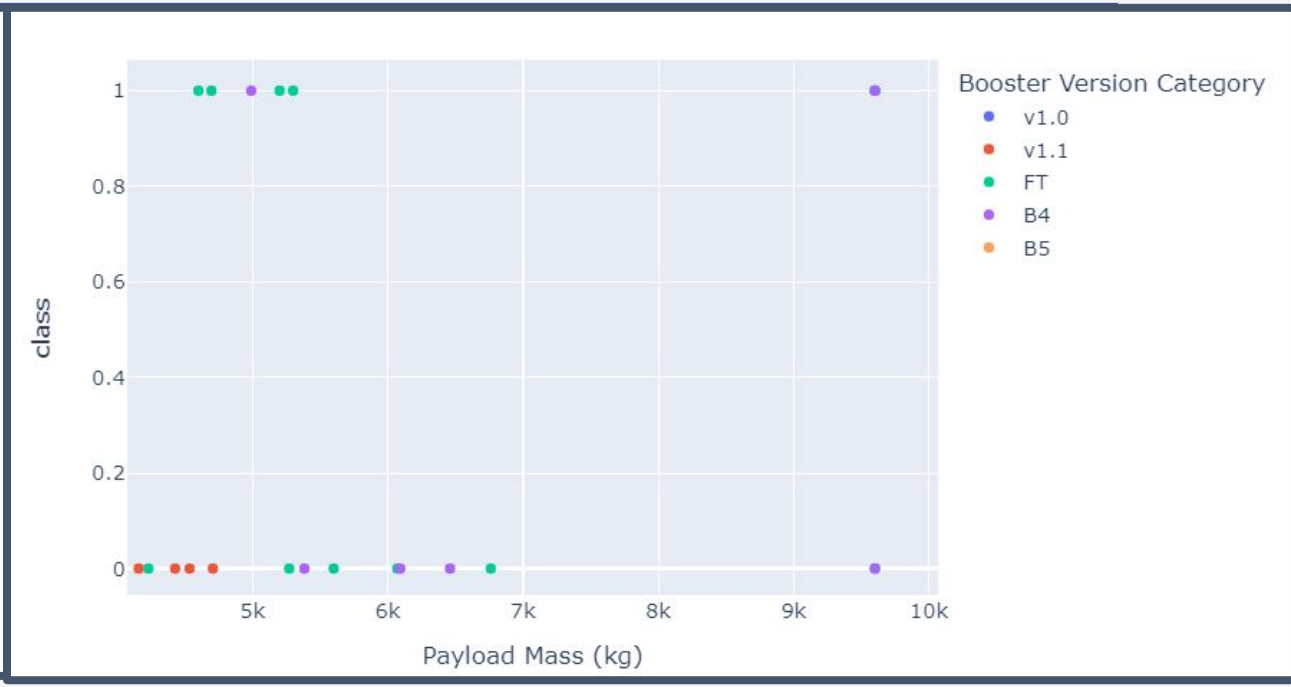
Payload vs. Launch Outcome Scatterplot



We can see that the FT booster category (in green) seems well represented among successful launches. Conversely, the v1.1 booster category (in red) has only one successful launch, with all others being unsuccessful.

Payload vs. Launch Outcome Scatterplot (cont.)





On the left are launches with a low payload mass, on the right are launches with a greater payload mass. Note that there is only one successful launch outcome with a payload greater than 5300 kg. Successful launch outcomes are most reliably achieved from about 2000 to 4000 kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Upon calculating the accuracy of the four machine learning methods used in this project, the Decision Tree method had a slight edge over the other three.

Find the method performs best:

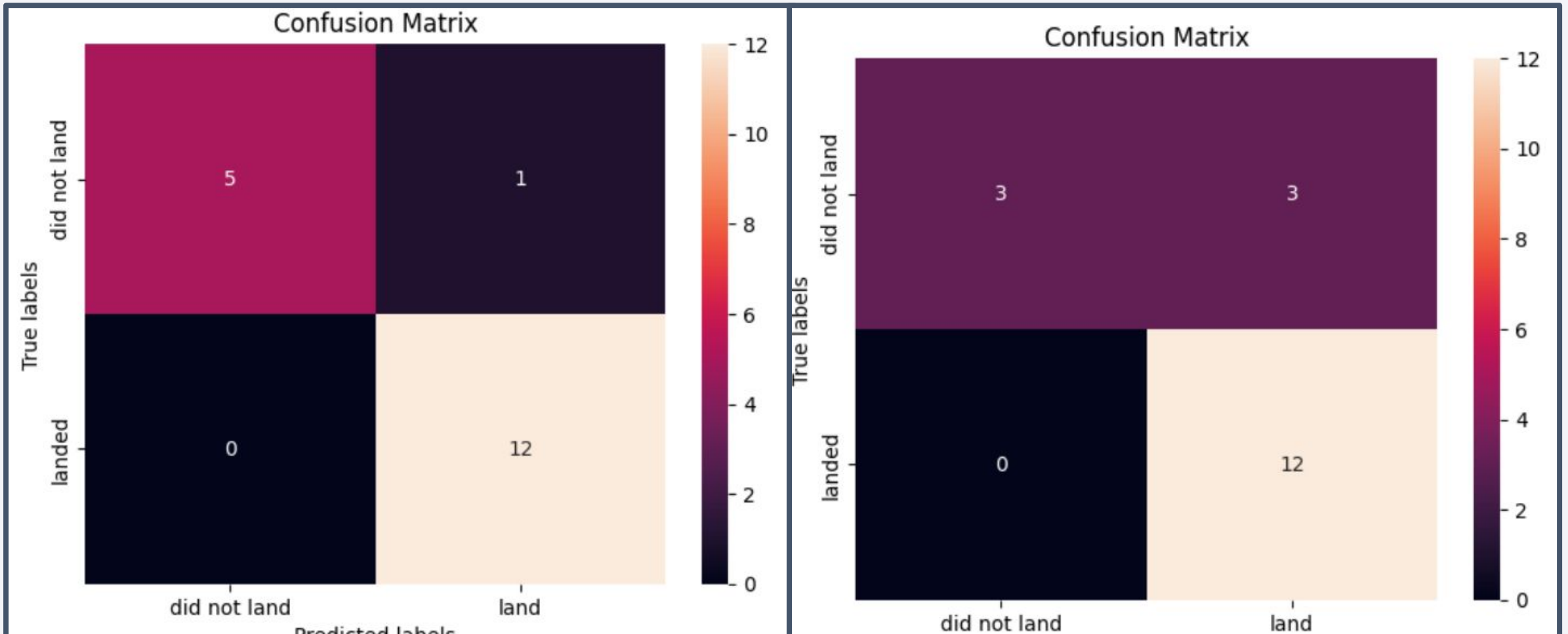
```
print("Logistic Regression accuracy = ", logreg_cv.score(X_test, Y_test))
print("Tree accuracy = ", tree_cv.score(X_test, Y_test))
print("SVM accuracy = ", svm_cv.score(X_test, Y_test))
print("KNN accuracy = ", knn_cv.score(X_test, Y_test))
```

```
Logistic Regression accuracy =  0.8333333333333334
Tree accuracy =  0.9444444444444444
SVM accuracy =  0.8333333333333334
KNN accuracy =  0.8333333333333334
```

The above scores indicate that the DecisionTreeClassifier gave the most accurate results.

Confusion Matrix

On the left is the Confusion Matrix for the Decision Tree method. On the right is the Confusion Matrix for the other three methods (Logistic Regression, SVM, KNN)



Conclusions

- The Decision Tree method most accurately predicted launch results.
- The KSC LC-39A launch site had both the highest number of successful launches, as well as the highest launch success ratio.
- Through SQL queries, we have made launch data more easily navigable for stakeholders.
- As years pass, successful launches are more consistently accomplished.

Appendix

- The full Github Repository can be accessed here:
  https://github.com/louisdbaker/IBM-Data-Science-Capstone/tree/master

Thank you!