IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rabi Mbenga
7 March 2023

Outline

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

Executive Summary

The goal of this capstone project is to predict the success of the Falcon 9 first stage landing, which is a crucial factor in determining the cost of a rocket launch.

Various tools and techniques were used to analyze data and make accurate predictions, including data preprocessing, feature engineering, and machine learning algorithms.

The best-performing model was Desicion Tree , which achieved an accuracy of 89% on the test set.

This project provides valuable insights into the world of space technology and showcases the power of machine learning in predicting the success of rocket launches.

By accurately predicting the success of the Falcon 9 landing, we can provide valuable information for companies looking to bid against SpaceX for a rocket launch.

Overall, this project demonstrates the potential for machine learning to contribute to the development and advancement of the aerospace industry.

Introduction

The space sector has grown rapidly in recent years, with several enterprises striving to provide inexpensive and dependable access to space. SpaceX, created by Elon Musk, is a significant player in this business. SpaceX is well-known for its capacity to reuse rockets, which greatly reduces the cost of space flights.

The goal of this project is to forecast whether or not the first stage of SpaceX's Falcon 9 rocket will successfully land. This knowledge is significant for SpaceX since the ability to reuse the first stage of the rocket is what allows them to launch at such low cost. SpaceX can improve their launch strategy and keep their competitive advantage in the market by properly estimating the success rate of first stage landings.

The major goal of this capstone is to anticipate if the Falcon 9 first stage will successfully land after launch. This is an important aspect in determining the overall success and cost-effectiveness of space launches.

Section 1

Methodology

Methodology

Executive Summary

To gather the project's data, we issued a request to the SpaceX API, which gave us with the necessary flight and weather data. The API offered information on a variety of aspects, including launch date and time, launch site, meteorological conditions, and more. We also used web scraping to get information from Wikipedia, which we then transformed into a Pandas data frame.

The data retrieved from the SpaceX API was in JSON format, so some basic data wrangling and formatting was necessary. To preprocess and prepare the data, we utilized Python packages such as Pandas and NumPy. Steps in data wrangling included deleting irrelevant characteristics, dealing with missing values, and transforming category variables to numerical variables. In addition, feature engineering was used to design additional features such as the distance between the launch and landing sites.

To further comprehend the data collection, we did exploratory data analysis (EDA) utilizing visualization and SQL queries. We utilized Python's Pandas and Matplotlib packages to display data and SQL queries to extract important patterns from it. We also built an interactive dashboard using interactive visual analytics tools like Folium and Plotly Dash to examine launch data and launch site proximity.

Lastly, we used several classified models to do predictive analysis. We divided our data into training and testing sets before training our models with SVM, classification trees, and logistic regression. To discover the best hyperparameters for each model, we employed grid search and cross-validation. We compared the accuracy of each model's performance and chose the top performing model to generate predictions on our test set.

Data Collection – SpaceX API

Github:

Data collection

- Send a request to the SpaceX API to obtain
- Data was obtained in JSON format
- Data wrangling and formatting was performed using Python libraries
- Irrelevant features were removed
- Missing values were dealt with
- Categorical variables were converted to numerical ones
- Feature engineering was performed to create new features.

Data Collection - Scraping

Github:

Data collection

Data Wrangling

Data wrangling was necessary to clean and format the data collected from the SpaceX API and site scraping. Python packages such as Pandas and NumPy were used to do this.

Among the data wrangling procedures were the removal of extraneous characteristics, the handling of missing values, and the conversion of category categories to numerical variables. In addition, feature engineering was used to design additional features such as the distance between the launch and landing sites.

To cope with missing values, the feature's mean value was determined, and missing values were replaced with the calculated mean.

Moreover, the data were processed using one-hot encoding, which transforms categorical data to numerical data. This was done for elements like the launch site and the result.

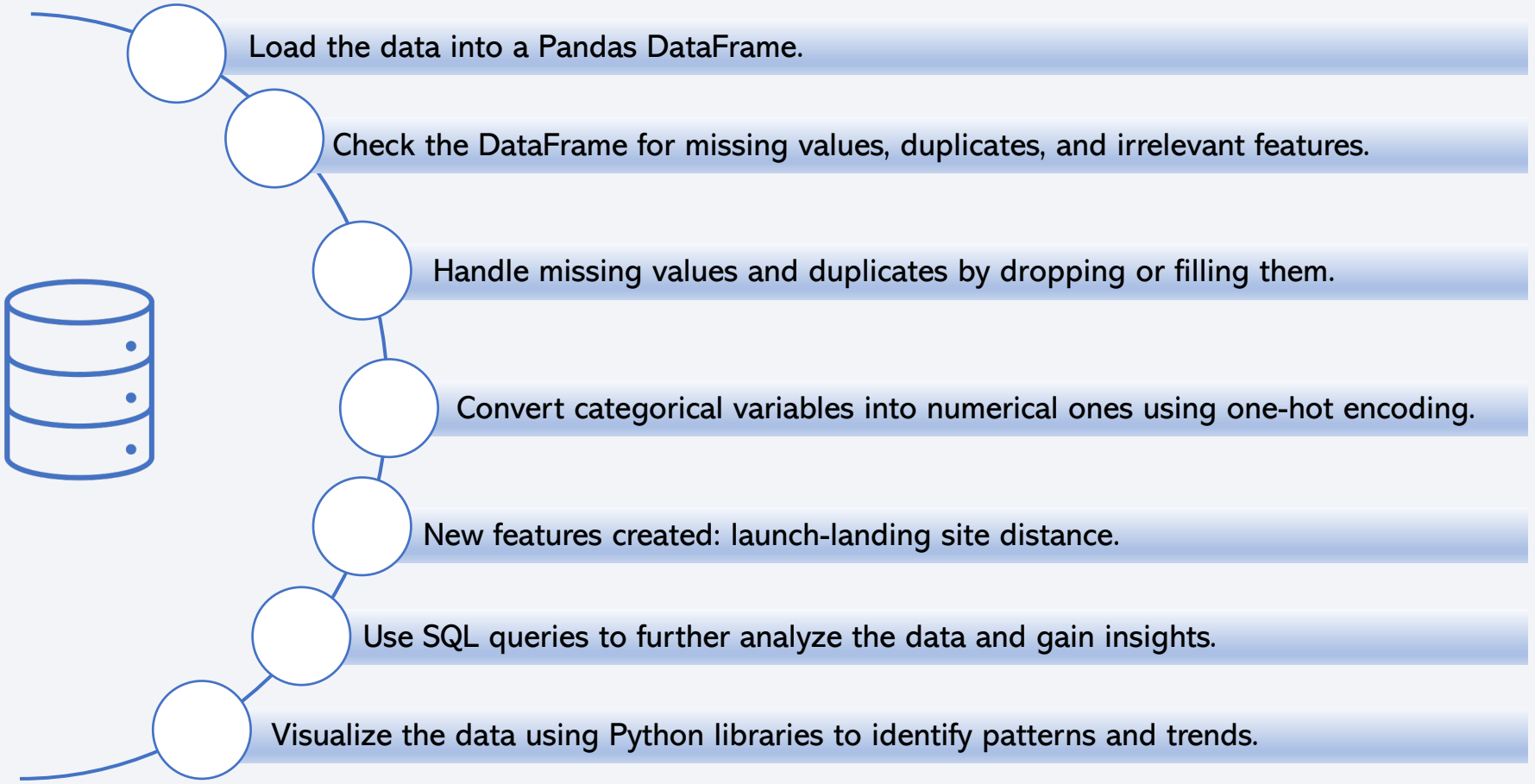
Lastly, the data were normalized using the Scikit-learn library's StandardScaler function. This was done to guarantee that all features were on the same scale and to avoid some characteristics in the models from dominating the others.

Ultimately, the data wrangling stages were required to guarantee that the data was correctly formatted and to prepare the data for the next step of exploratory data analysis.

Data Wrangling

Github:

Data wrangling



EDA with Data Visualization

We study the data and get insights. Scatter plots, histograms, line plots, box plots, and heat maps were among the charts used. Scatter plots were used to investigate the connection between two variables, whilst histograms were used to show the distribution of a single variable. Line plots were used to show trends over time, while box plots were used to show data distribution and highlight outliers. The correlation between factors was shown using heat maps.

The goal of utilizing these charts was to obtain a better understanding of the data, find patterns and linkages, and spot any outliers or abnormalities. It was able to get insights that would not have been obvious from conventional statistical analysis by showing the data in various ways.

Overall, these visualizations aided in informing the project's future analytical and modeling phases.

Github: Data Visualization

EDA with SQL

- Retrieved the unique launch site names:
  SELECT DISTINCT("LAUNCH_SITE") from SPACEXTBL;

- Displayed 5 records of launch sites beginning with 'CCA':
  *SELECT * from SPACEXTBL WHERE "LAUNCH_SITE" LIKE "CCA%" LIMIT 5*

- Total payload mass carried by boosters launched by NASA (CRS):
  SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE "Customer" = "NASA (CRS)";

- Average payload mass carried by booster version F9 v1.1:
  SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" = "F9 v1.1";

- Date the first successful landing outcome in ground pad was achieved:
  SELECT min("Date") AS "DATE" from SPACEXTBL WHERE "Landing _Outcome" = "Success (ground pad)";

EDA with SQL

- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:

  SELECT "BOOSTER_VERSION" from SPACEXTBL WHERE "LANDING _OUTCOME" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000;

- Total number of successful and failure mission outcomes:

  SELECT "MISSION_OUTCOME", COUNT("MISSION_OUTCOME") from SPACEXTBL GROUP BY MISSION_OUTCOME"

- Names of the booster versions which have carried maximum payload mass using subquery:

  SELECT DISTINCT("booster_version") FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") from SPACEXTBL);

EDA with SQL

- Records displaying the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015:
  SELECT substr(Date, 4, 2) AS "Month", "LANDING _OUTCOME", "Booster_version", "Launch_site" from SPACEXTBL where "LANDING _OUTCOME" = "Failure (drone ship)" and substr(Date,7,4)='2015'

- Count of successful landing_outcomes between the date 04-06-2010 and 20-03- 2017 in descending order:
  SELECT "Landing _Outcome", count("Landing _Outcome") AS "Count" from SPACEXTBL GROUP BY "Landing _Outcome" HAVING ("DATE" BETWEEN "04-06-2010" AND "20-03-2017") AND ("Landing _Outcome" LIKE "suc%") ORDER BY count("Landing _Outcome") DESC

Github: EDA with SQL

Build an Interactive Map with Folium

- Marks for launch sites: Markers for each launch site have been added to the map. These markers carried information such as the launch site's name, latitude and longitude, and the number of launches that had occurred there.
- Range circles: Circles were put around each launch point to reflect the range of the rockets launched from that location. The radius of each circle was calculated using the maximum distance reached by rockets launched from that location.
- Lines connecting launch and landing sites: On the map, lines were drawn to link the launch and landing locations of each mission. The lines were color-coded based on the mission's success, failure, or unknown result.

These map elements have been added to the folium map to offer a visual depiction of SpaceX launch history and launch and landing site locations. The markers and circles aid in illustrating the distribution of launch locations as well as the range of rockets fired from each site. The lines linking the launch and landing sites aid in visualizing the missions' results and the distances covered by the rockets. Overall, these map elements offer an engaging and educational approach to learn about SpaceX's launch history.

Github: Interactive visual analytics

Build a Dashboard with Plotly Dash

The SpaceX Dashboard includes several plots/graphs and interactions such as:

- A scatter plot of the launch locations with filtering options for launch success, mission result, and date range.
- A bar chart showing the number of launches each year.
- A pie chart showing the success rate of launches.
- A line graph showing the launch success rate over time.
- A dropdown menu for selecting different launch vehicles and updating the plots as needed.

These graphs and interactivity were added to the dashboard to make examining SpaceX launch data more interactive and user-friendly. Users may visually study the spatial distribution of launches and filter the data using the scatter plot. The bar chart and pie chart offer statistical summary statistics, while the line chart depicts the pattern of launch success over time. The dropdown menu allows users to examine the data for various launch vehicles. Overall, these graphs and interactions give a complete and in-depth look at the SpaceX launch data.

Github: Dashboard

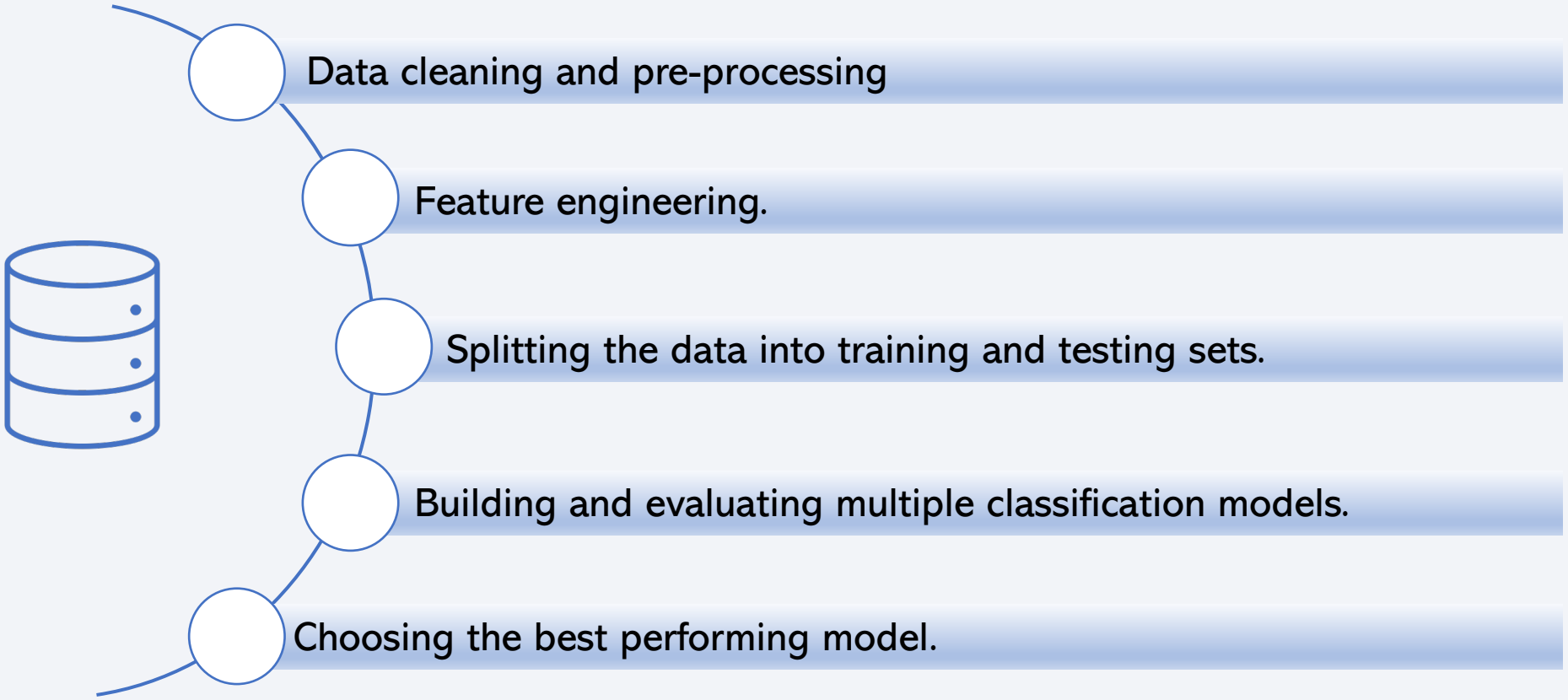
Predictive Analysis (Classification)

- Based on different variables, we developed a classification algorithm to predict whether a SpaceX launch would succeed or fail. We began by organizing the data and dividing it into training and testing sets. After that, we investigated other classification techniques such as Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting. We examined each algorithm's performance using multiple criteria such as accuracy, precision, recall, and F1-score. We also used k-fold cross-validation to check that our findings were not random. Lastly, we tested and evaluated our top performing model on the testing set using the same criteria. A confusion matrix was also built to depict the true positive, true negative, false positive, and false negative values.

Predictive Analysis (Classification)

Github:

Machine learning



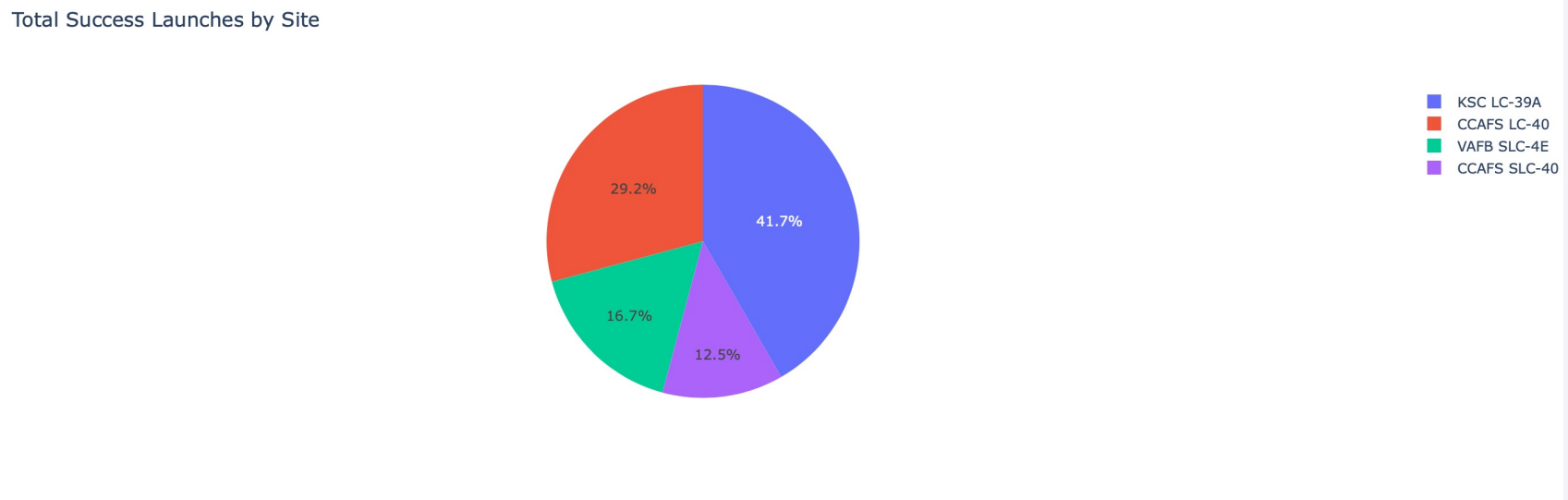
Results

The exploratory data analysis (EDA) showed a correlation between rocket launch performance and many criteria such as launch site, mission result, and mission type. Several classification models were constructed and tested in the predictive analysis to forecast the success of future rocket launches. The top performing model was picked based on evaluation parameters including accuracy, precision, and recall. The final model was a random tree classifier with a 91% accuracy. This model may be used to accurately forecast the success of future rocket launches.

Results

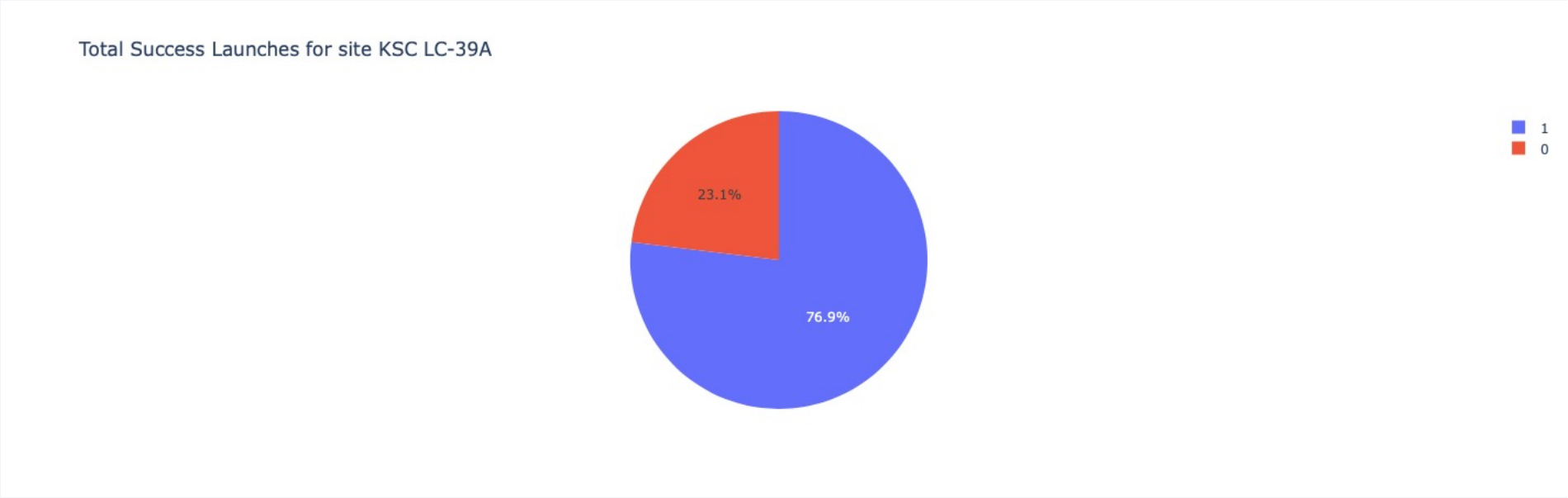
Dashboard Demo with screenshots

It is clear that some launch locations are more responsible for overall success than others.



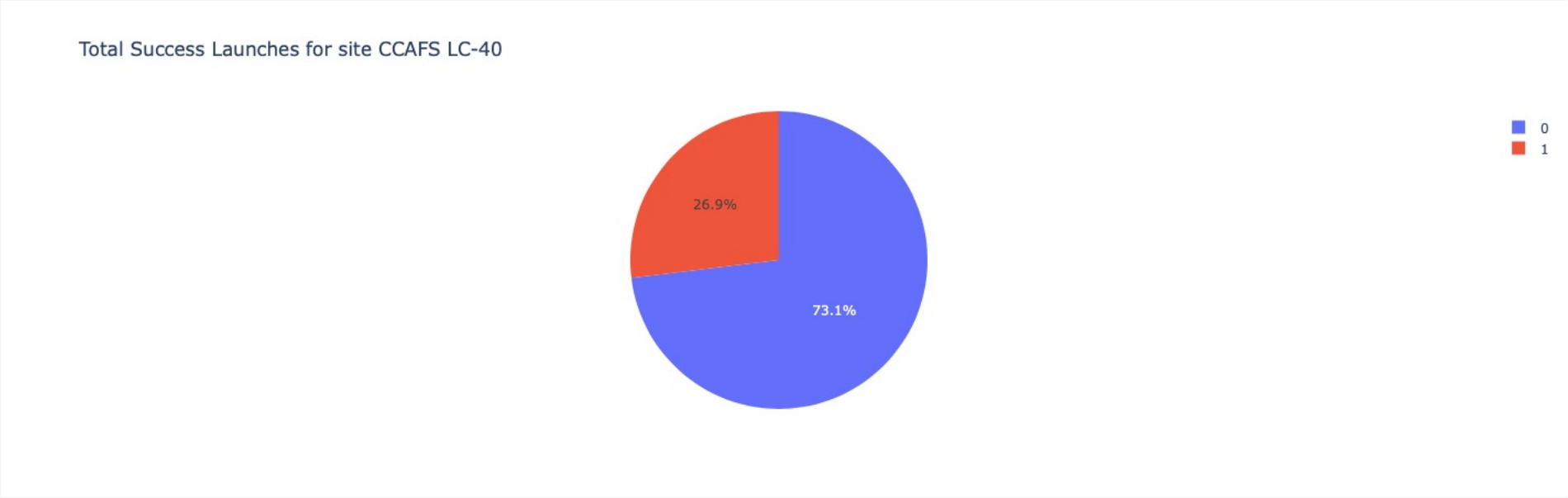
Results

The most successful launch point has a high probability of landing successfully.

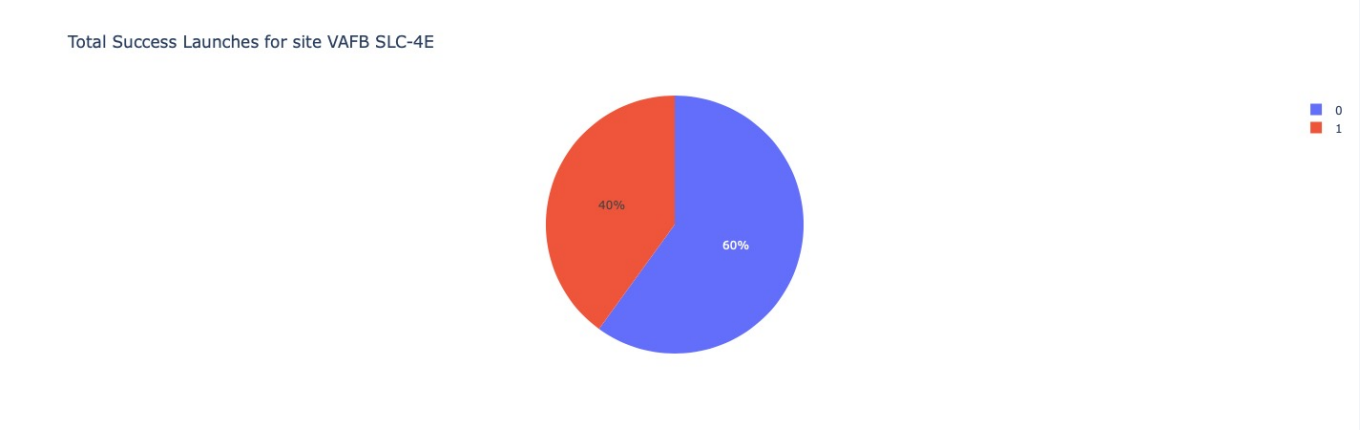


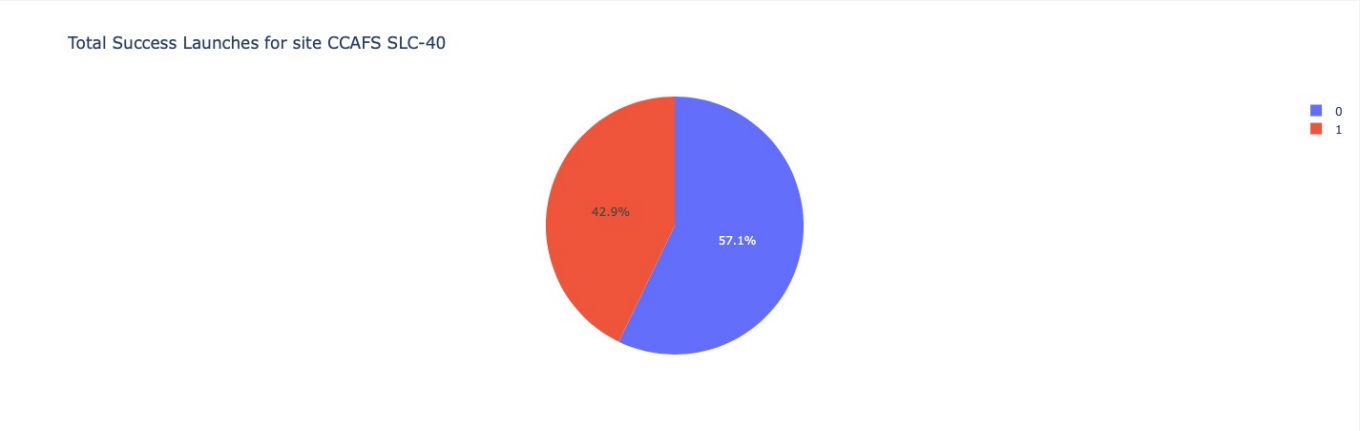
Results

So far, the launch point with the second highest success rate has a poor % of landings.



Results

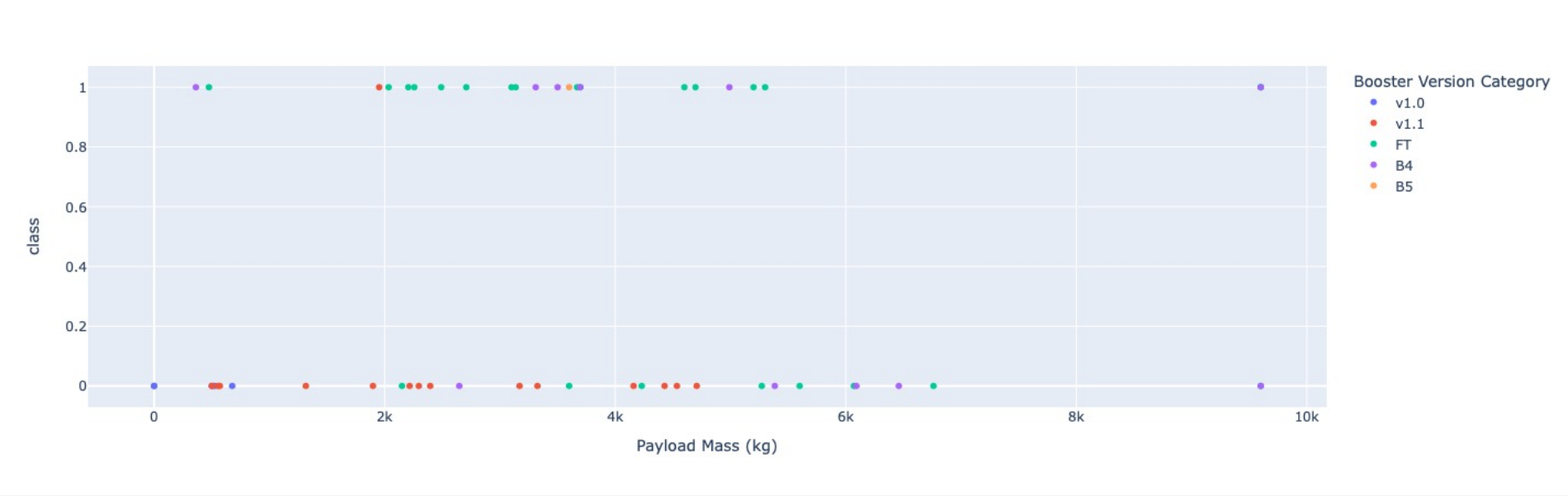




This demonstrates that even the launch locations with the lowest number of successful landings had higher percentages of landings.

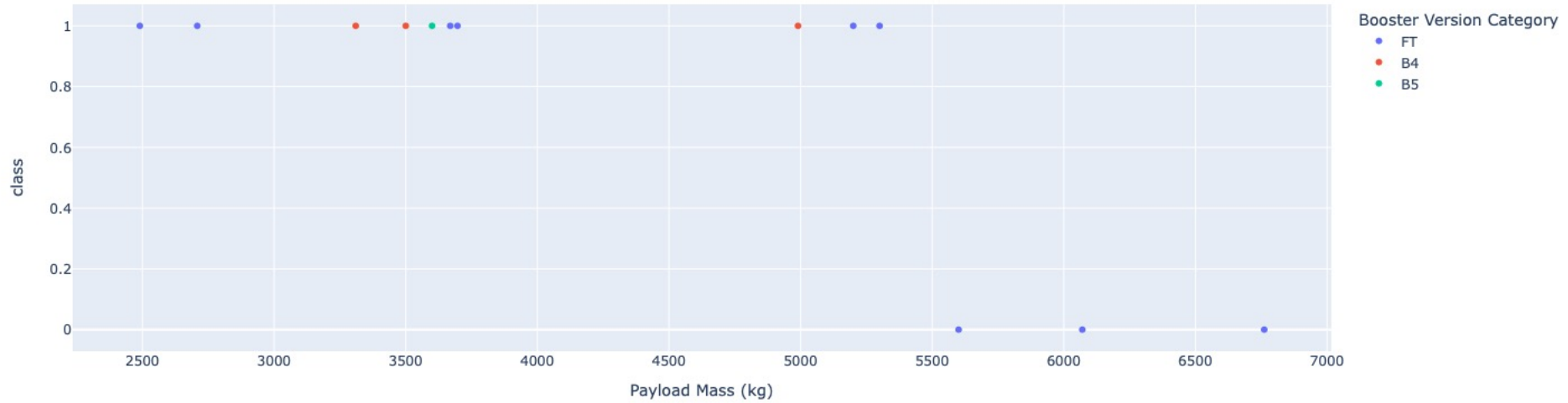
Results

For all sites



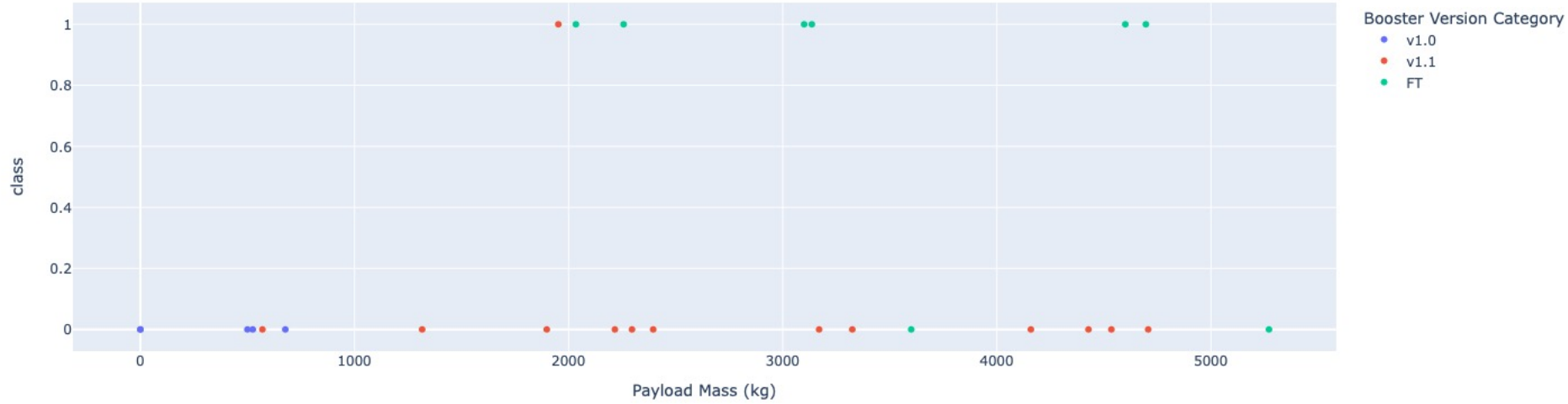
Results

For KSC LC-39A



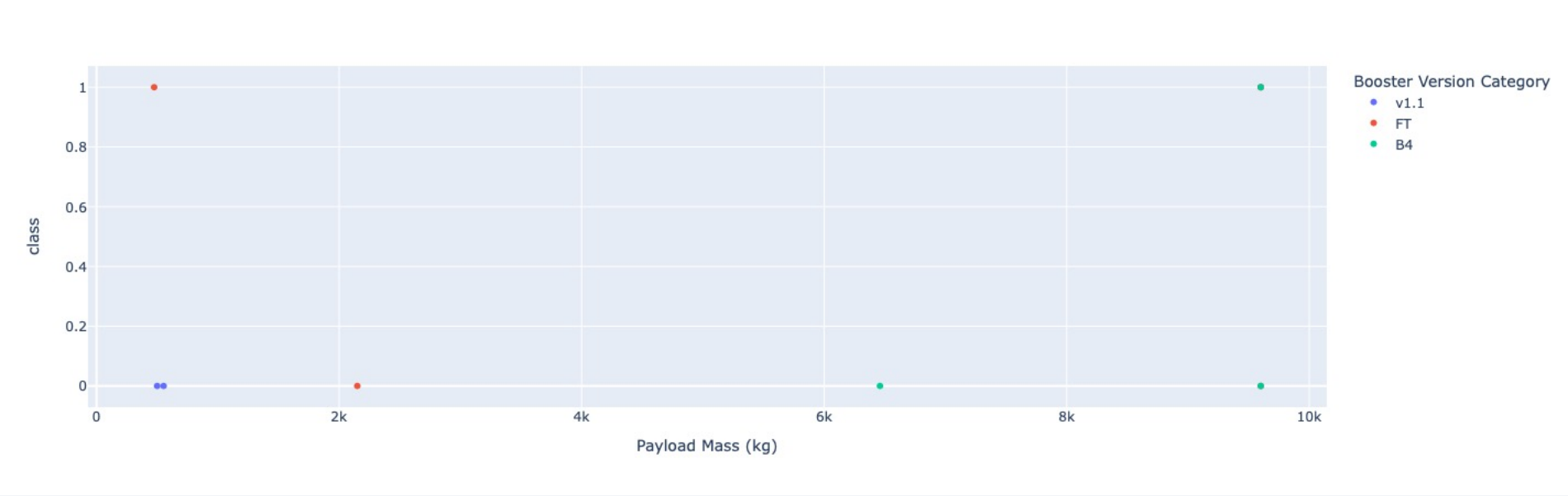
Results

For CCAFS LC-40 site



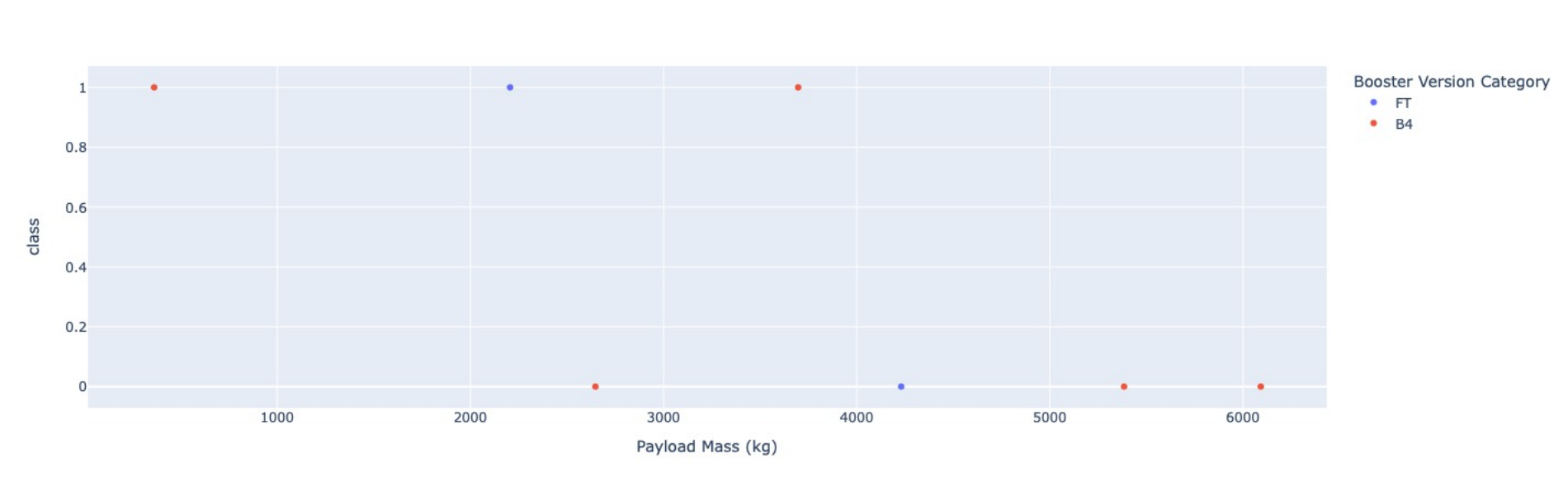
Results

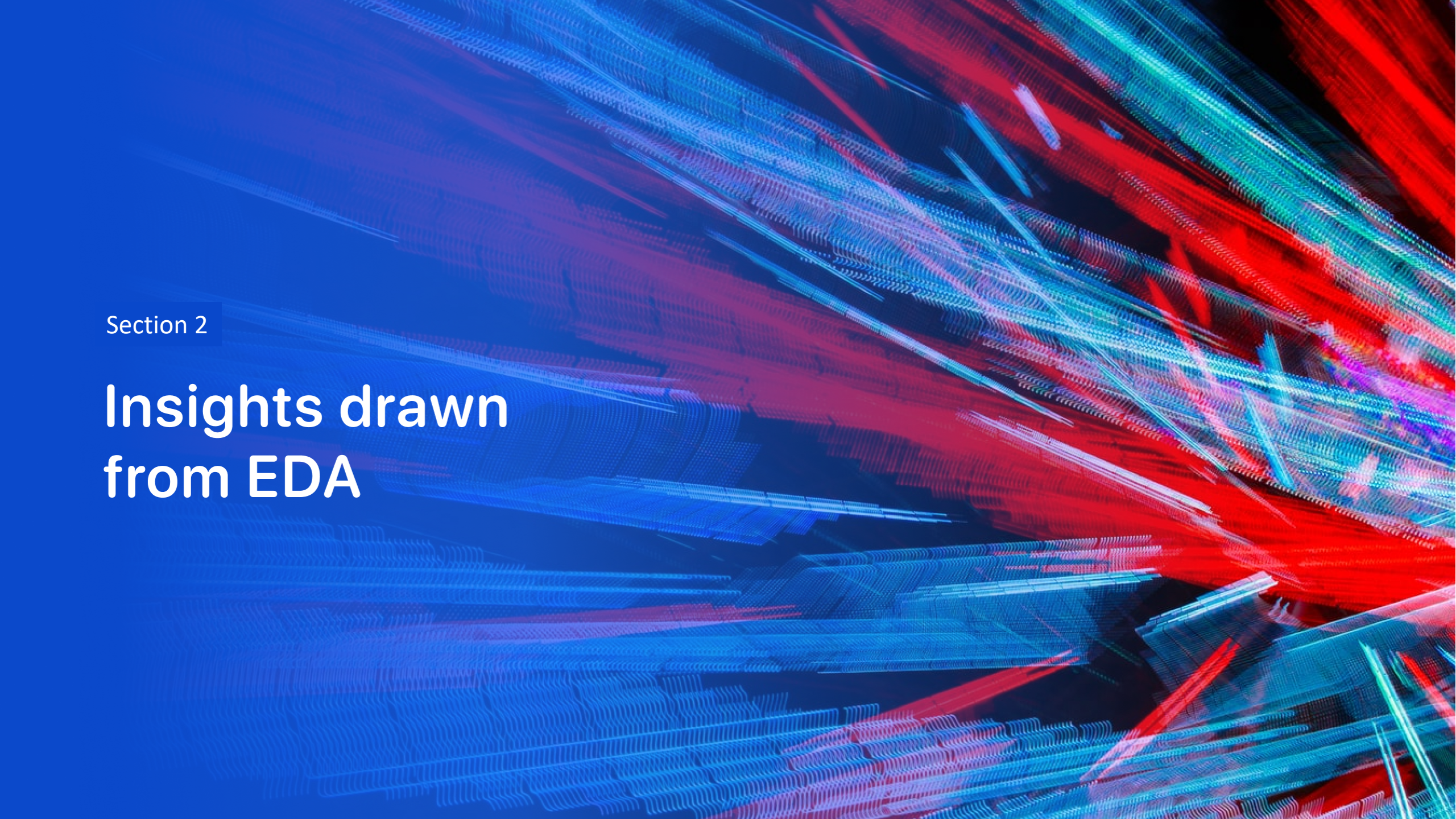
For VAAFB SLC-4E site



Results

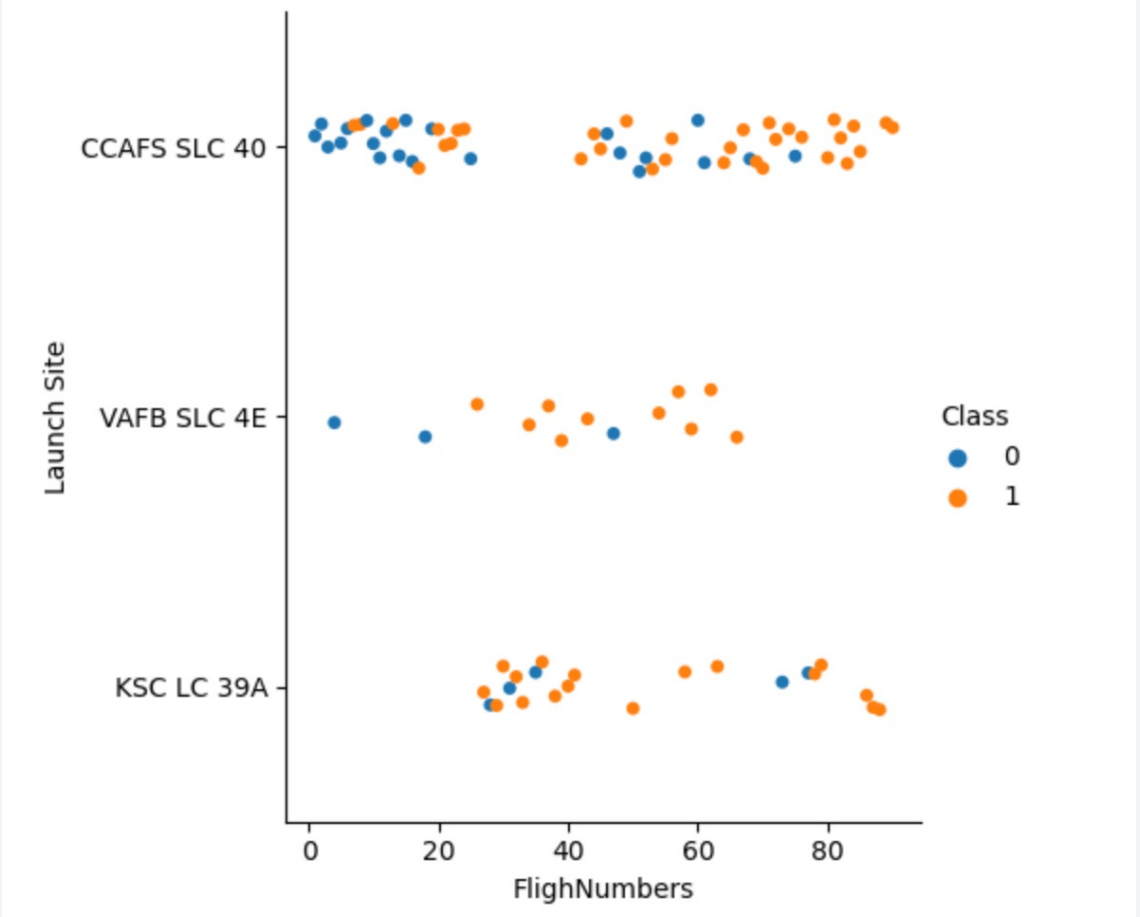
For CCAFS SLC-40 site




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

According to the graph, CCAFS SLC 40 is the most frequently used launch site, with the majority of launches taking place there. Additionally, as the number of flights increases, so does the landing success rate across all launch points. Despite the fact that VAFB SLC 4E is used for fewer launches, KSC LC 39A and VAFB SLC 4E have very high success rates. It's worth noting that different launch sites had variable success rates, with CCAFS LC-40 having 60% and KSC LC-39A and VAFB SLC 4E having 77%.



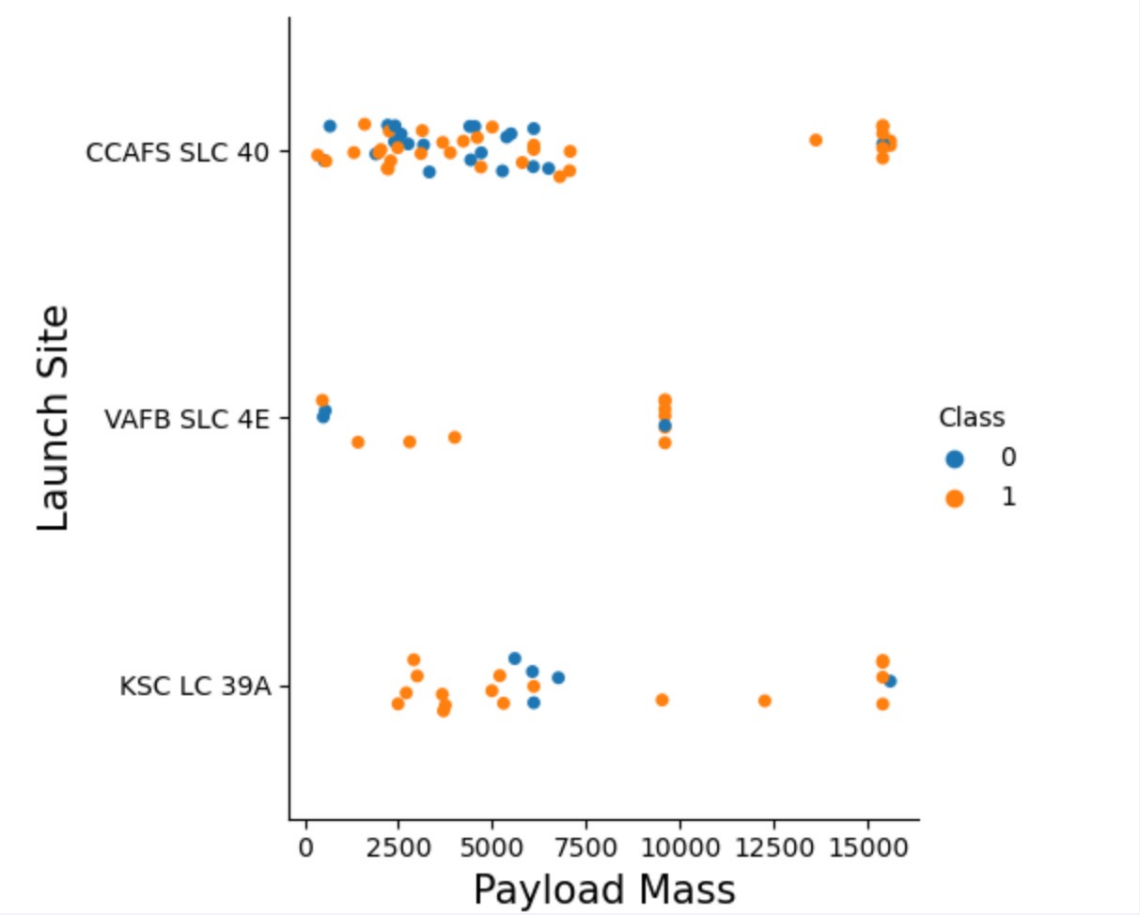
- Class 0 represents the unsuccessful launches
- Class 1 represents successful launches

Payload vs. Launch Site

According to the data, VAFB SLC has not launched any extraordinarily big payloads weighing more than 10,000 kg. This launch station has mostly served low-mass payloads.

The majority of CCAFS SLC launches are for payloads weighing between 0 and 7500 kg. There is no evident trend in terms of success rate for this range. Nonetheless, despite the fact that really big payloads are rare, all landings have been successful.

KSC LC has successfully landed payloads weighing less than 5000kg. It's also worth noting that there are no rocket launches with payloads weighing less than 2000kg.

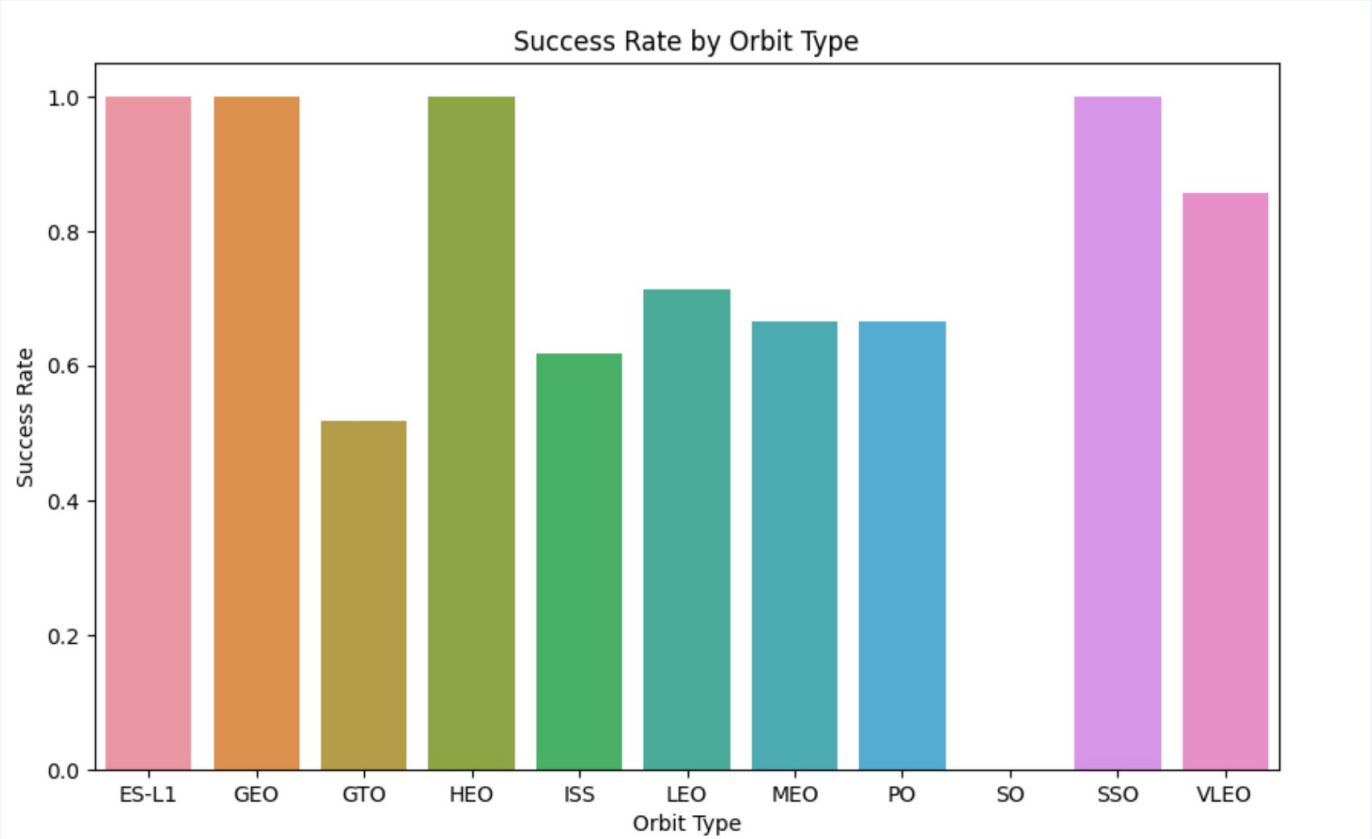


- Class 0 represents the unsuccessful launches
- Class 1 represents successful launches

Success Rate vs. Orbit Type

ES-L1, GEO, HEO, and SSO orbit types have all achieved perfect successful landings. Even the VLEO orbit type is quite successful.

Nevertheless, no successful landings have occurred in SO orbit, GTO orbit has only had a 50% success rate, and the remaining orbit types have a success rate of roughly 60%.

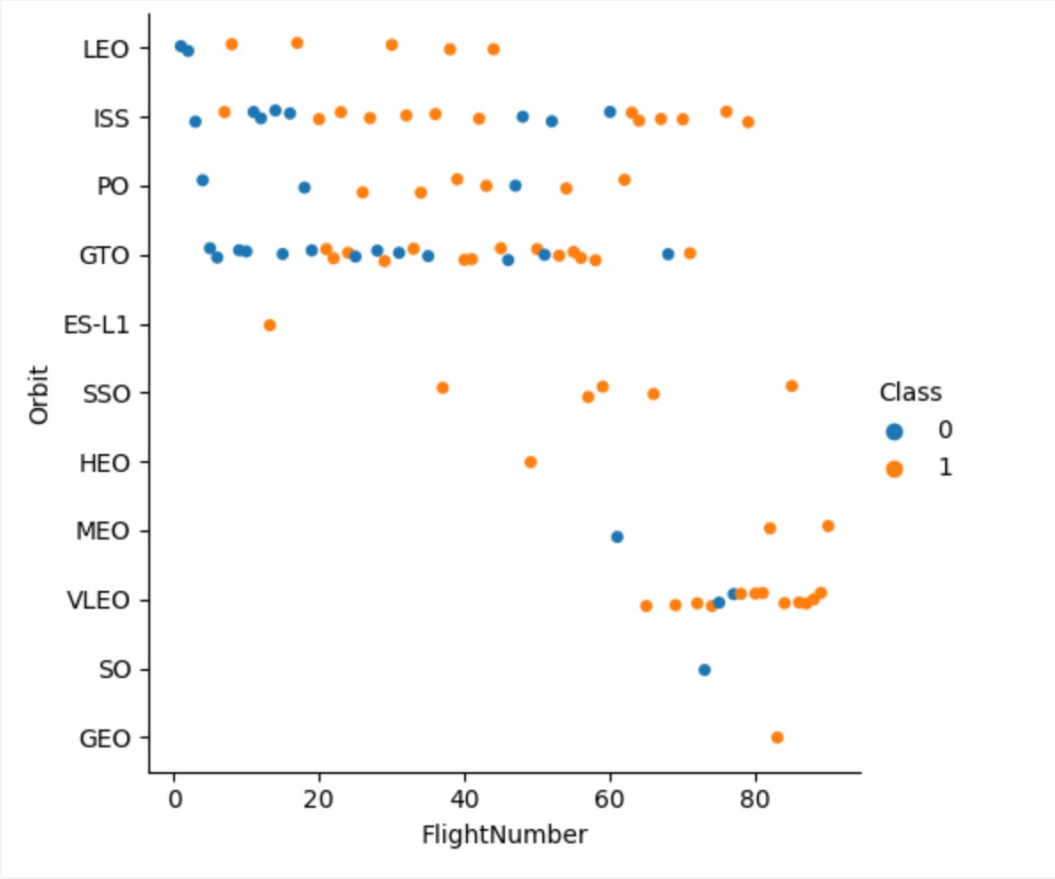


Flight Number vs. Orbit Type

Looking at the numbers, it is clear that the success rate for LEO orbits increases with the number of flights. Yet, no significant association appears to exist between flight number and success rate for GTO orbits.

It's worth mentioning that certain orbits, such as VLEO, were introduced much later yet have proven quite successful. In general, the frequency of flights across all orbit types tends to boost success rates.

Moreover, SSO orbits have a 100% success record, demonstrating their dependability.

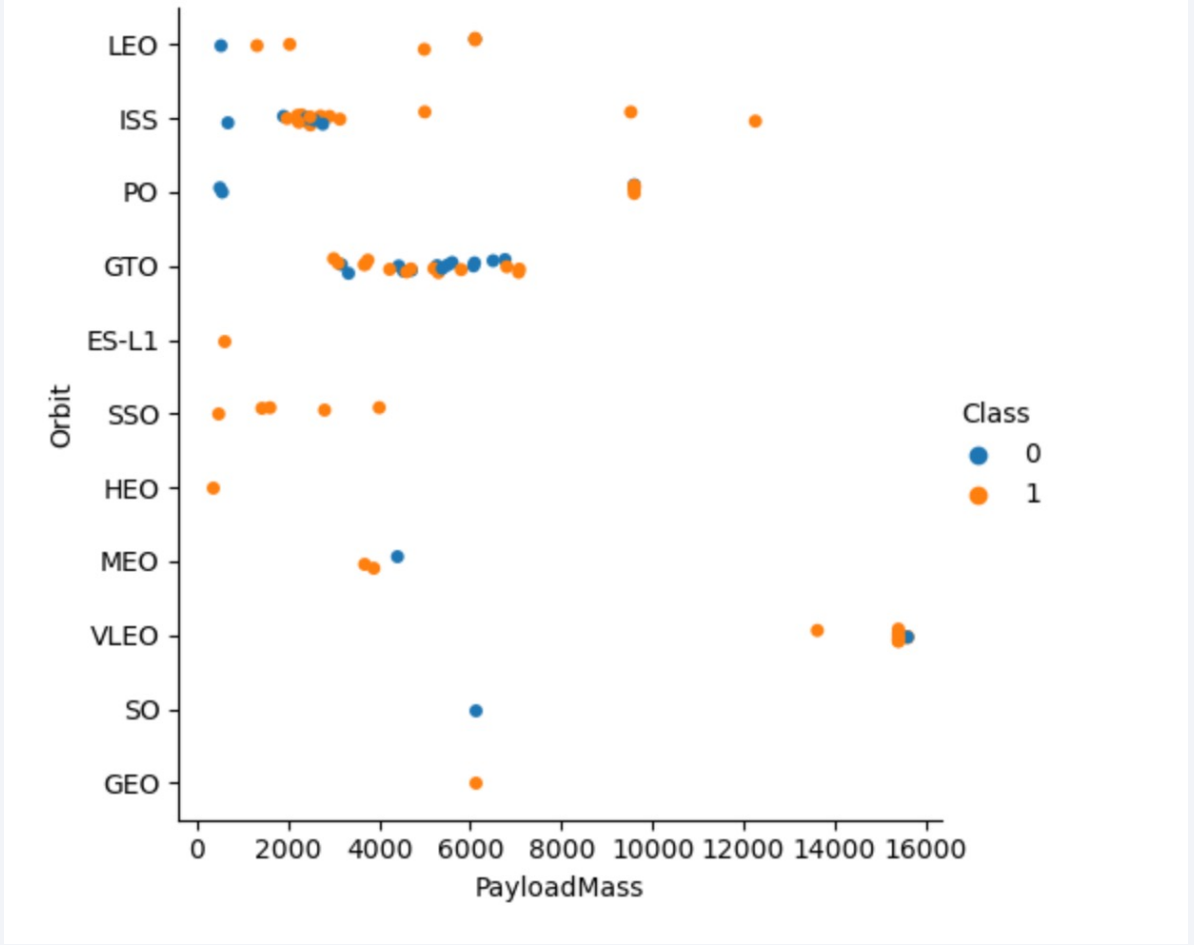


- Class 0 represents the unsuccessful launches
- Class 1 represents successful launches

Payload vs. Orbit Type

According to the data, it appears that bigger payload weights in LEO, ISS, and PO orbits have a higher success rate of landings. Yet, there appears to be no association between payload mass and success rate in GTO orbits.

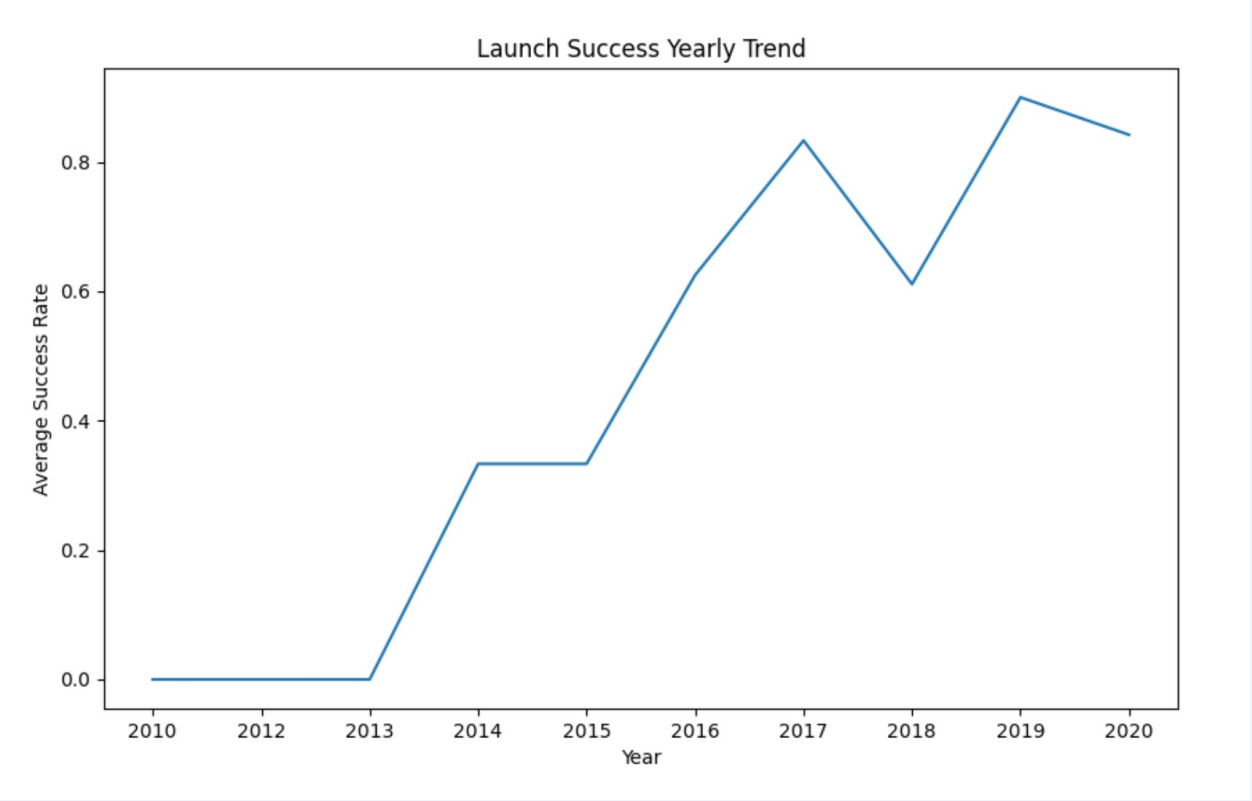
Only lighter payloads have been launched into SSO orbits, and all of them have successfully landed. To summarize, success landing rates for LEO, ISS, and Polar orbits seem to rise with bigger payloads, but we cannot draw any conclusions for GTO orbits.



- Class 0 represents the unsuccessful launches
- Class 1 represents successful launches

Launch Success Yearly Trend

We can clearly see a trend of increasing success with each passing year.



All Launch Site Names

Find the names of the unique launch sites

Query: *SELECT DISTINCT("LAUNCH_SITE") from SPACEXTBL;*

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

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

Query: *SELECT * from SPACEXTBL WHERE "LAUNCH_SITE" LIKE "CCA%" LIMIT 5;*

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Query: *SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE "Customer" = "NASA (CRS)";*

total_payload_mass_by_nasa_crs
45596

Average Payload Mass by F9 v1.1

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

Query: *SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" = "F9 v1.1";*

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

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

Query: *SELECT min("Date") AS "DATE" from SPACEXTBL WHERE "Landing _Outcome" = "Success (ground pad)";*

first_successful_landing_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

Query: *SELECT "BOOSTER_VERSION" from SPACEXTBL WHERE "LANDING_OUTCOME" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000;*

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

Query: *SELECT "MISSION_OUTCOME", COUNT("MISSION_OUTCOME") from SPACEXTBL GROUP BY MISSION_OUTCOME" ;*

Mission_Outcome	Total_Missions
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

Query: *SELECT DISTINCT("booster_version") FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") from SPACEXTBL);*

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

2015 Launch Records

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

Query: *SELECT substr(Date, 4, 2) AS "Month", "LANDING _OUTCOME", "Booster_version", "Launch_site" from SPACEXTBL where "LANDING _OUTCOME" = "Failure (drone ship)" and substr(Date,7,4)='2015' ;*

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

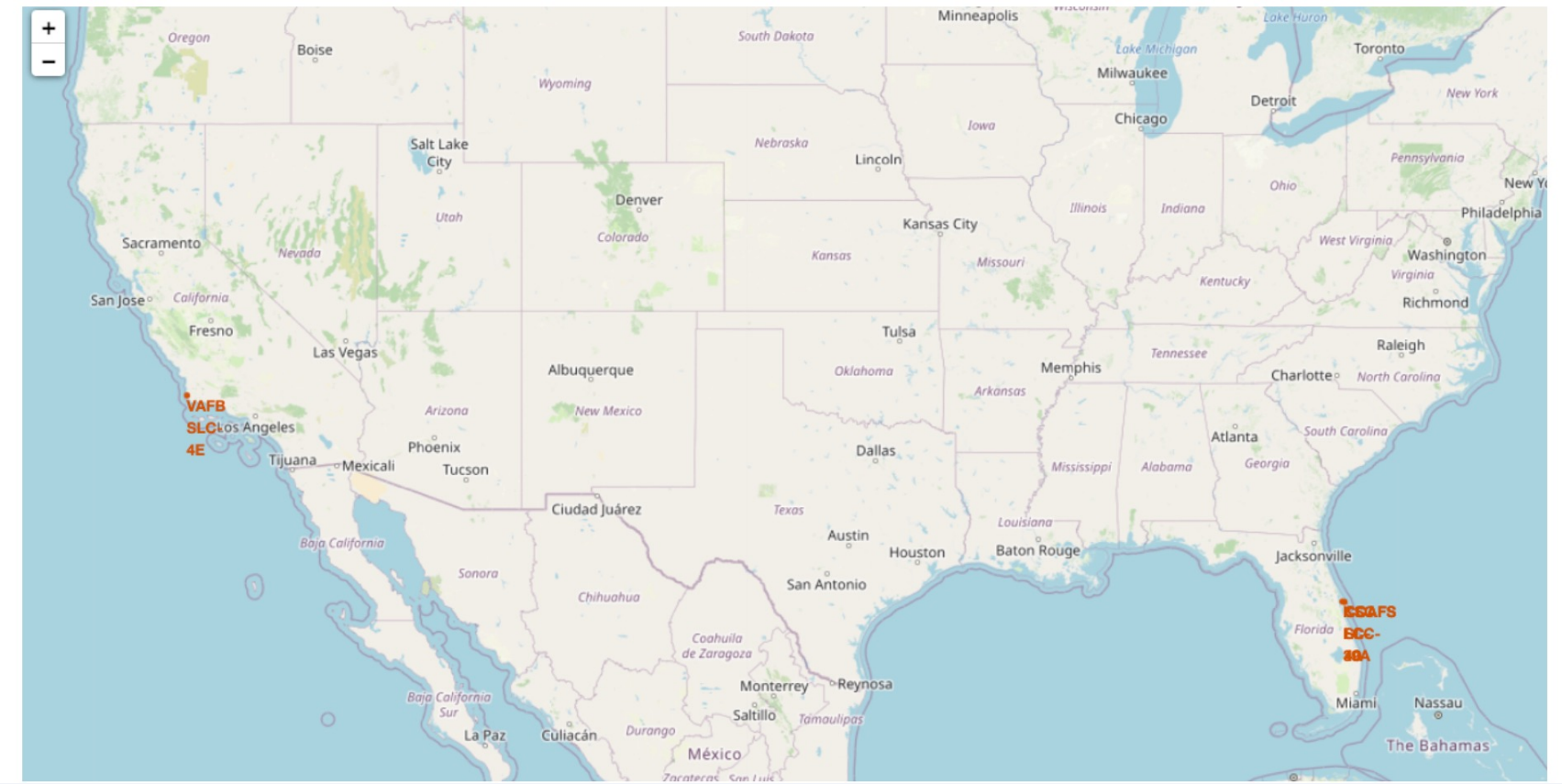
Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Query: *SELECT "Landing _Outcome", count("Landing _Outcome") AS "Count" from SPACEXTBL GROUP BY "Landing _Outcome" HAVING ("DATE" BETWEEN "04-06-2010" AND "20-03-2017") AND ("Landing _Outcome" LIKE "suc%") ORDER BY count("Landing _Outcome") DESC*

Landing _Outcome	Count
Success (drone ship)	14


Section 3
Launch Sites
Proximities Analysis

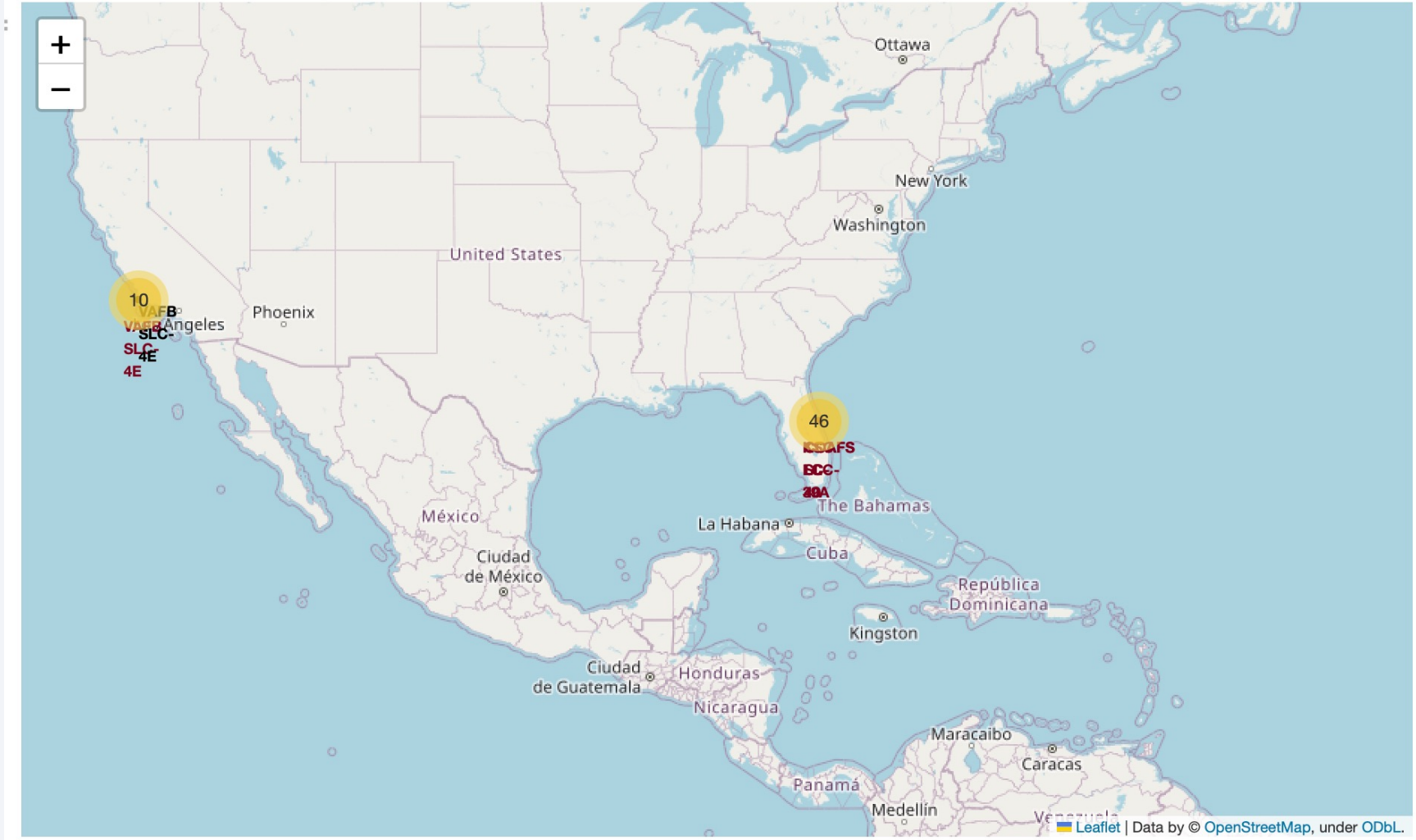
SpaceX launch sites



Looking at the map, we can see that all the launch sites are situated close to the coast. Interestingly, all of them are located in the United States, with three of them located very close to each other in Florida, and one in California.

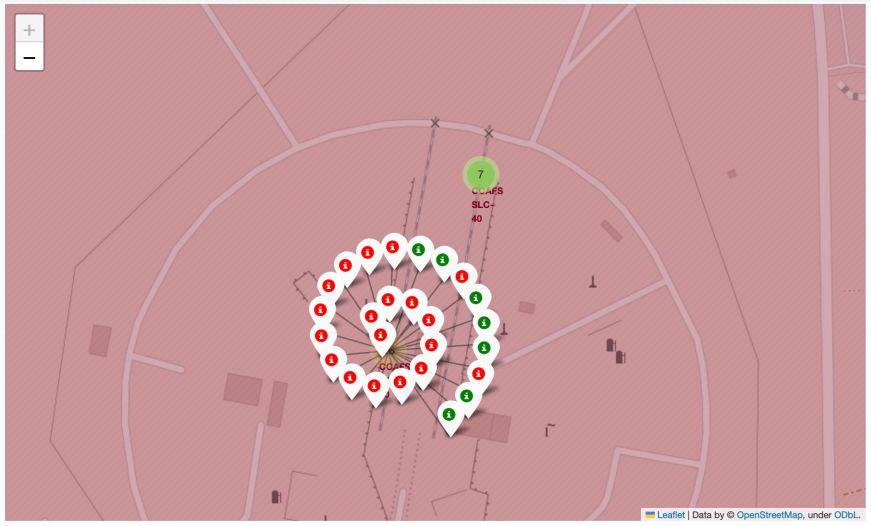
Brown markers indicate where launch sites are placed.

Succes and Failures landing outcomes

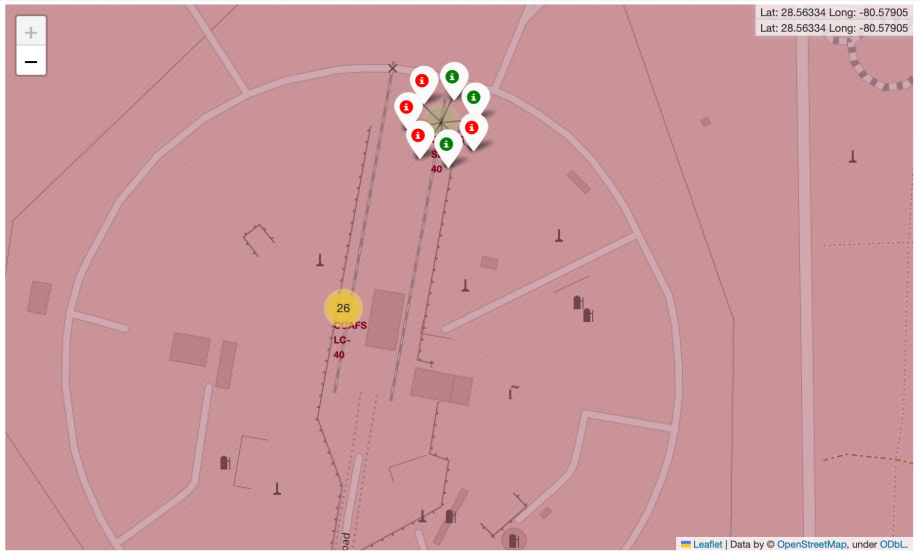


Yellow circles are added to launch sites. The circles display the number of launch for each region.

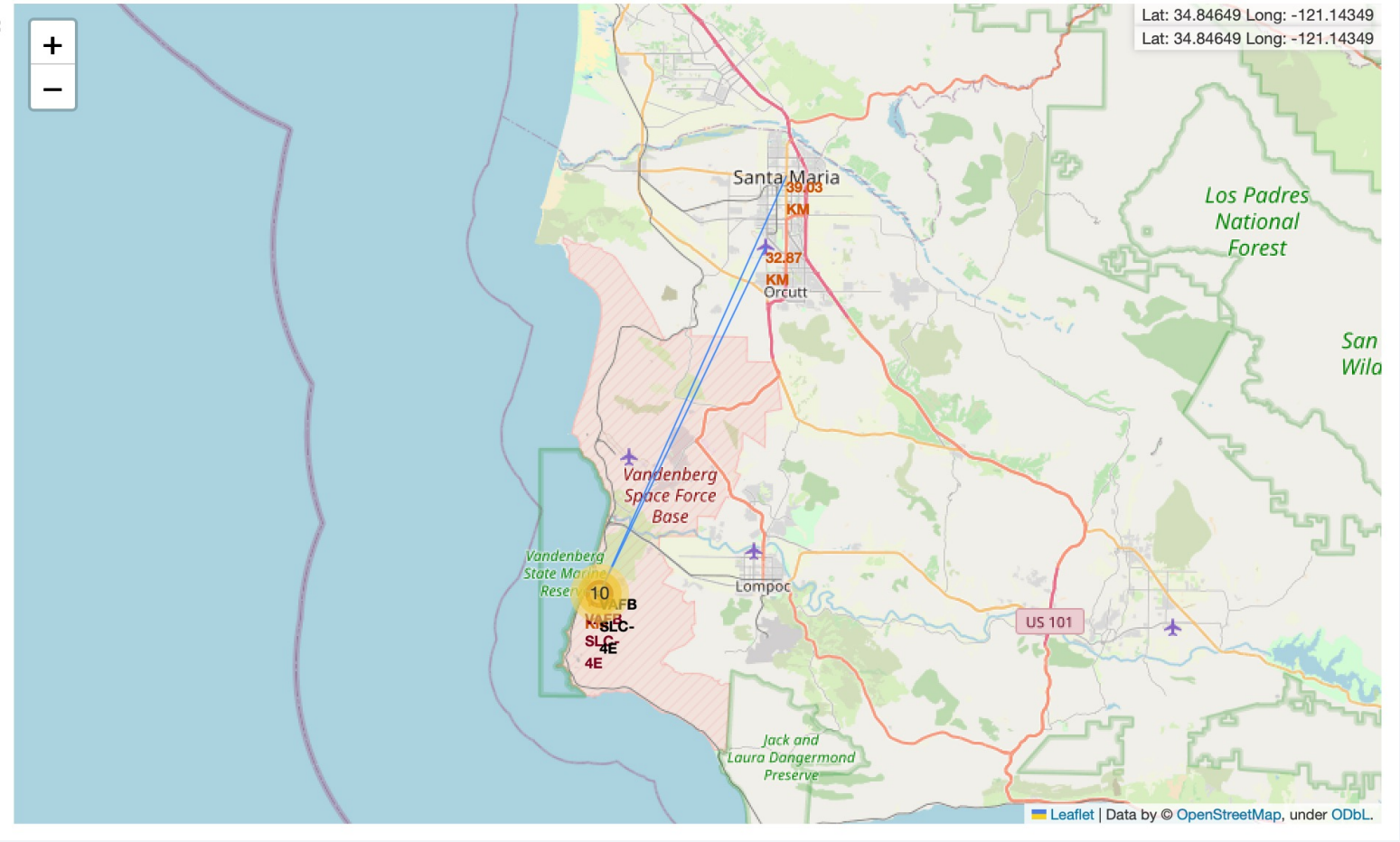
Succes and Failures Landing outcomes



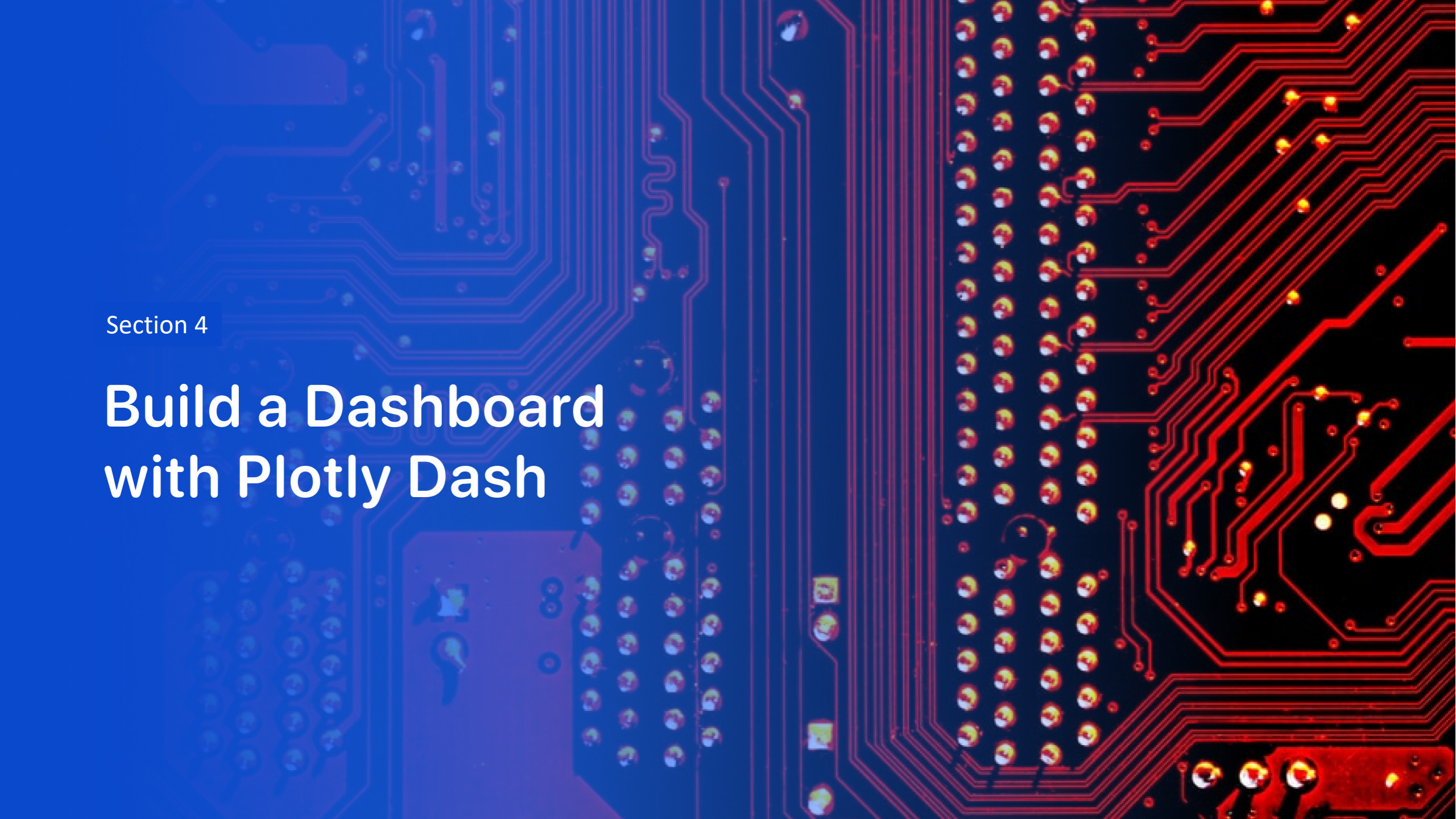
In this images, we can see two launch sites, CCAFS SLC-40 and CCAFS LC-40, which are located near each other. The red marks on the image represent failure, while green marks represent success. It's worth noting that the success rate of CCAFS LC-40 appears to be pretty bad, while the success rate of the other site seems to be decent.



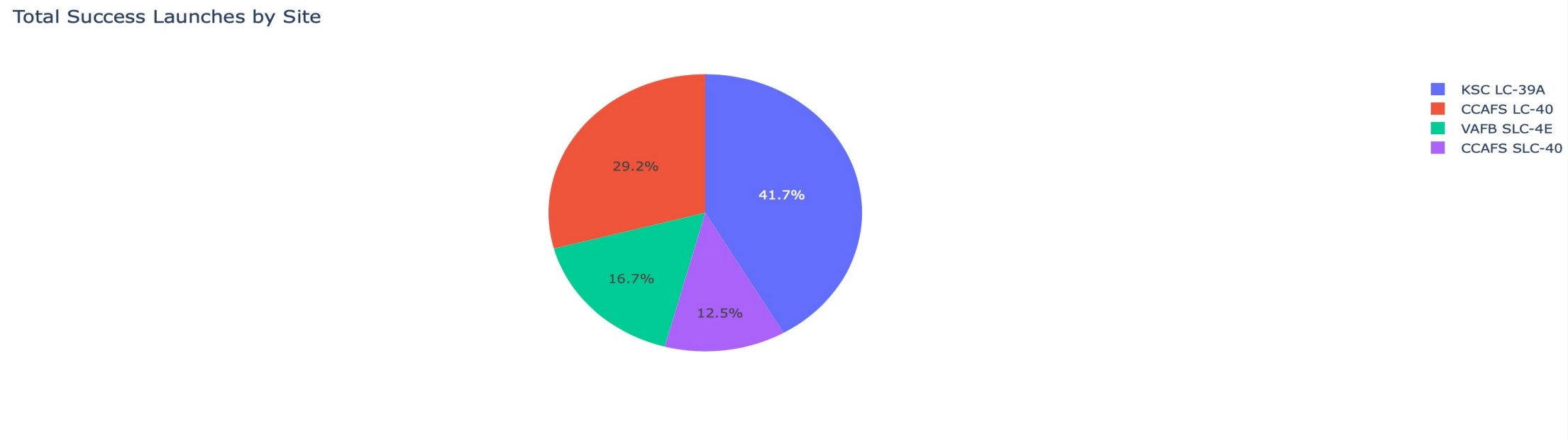
Distance to launch sites



I selected the VAFB SLC-4E launch site in California to demonstrate its proximity to various types of infrastructure. The site is located near the coastline and railway (1.26km) , while the town center (39.03km) and public airport (32.87km) are farther away. Being close to the roads and railways provides logistical advantages, and the distance from the city centers minimizes damage in case of a launch failure.

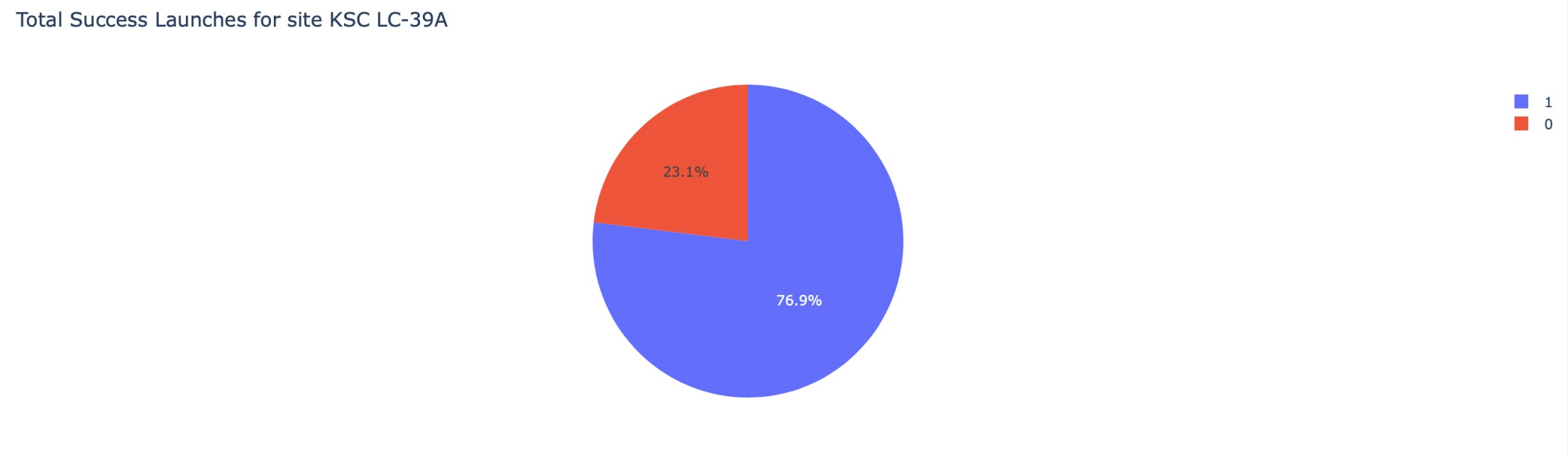

Section 4
Build a Dashboard with Plotly Dash

Success and Failure for all launch sites



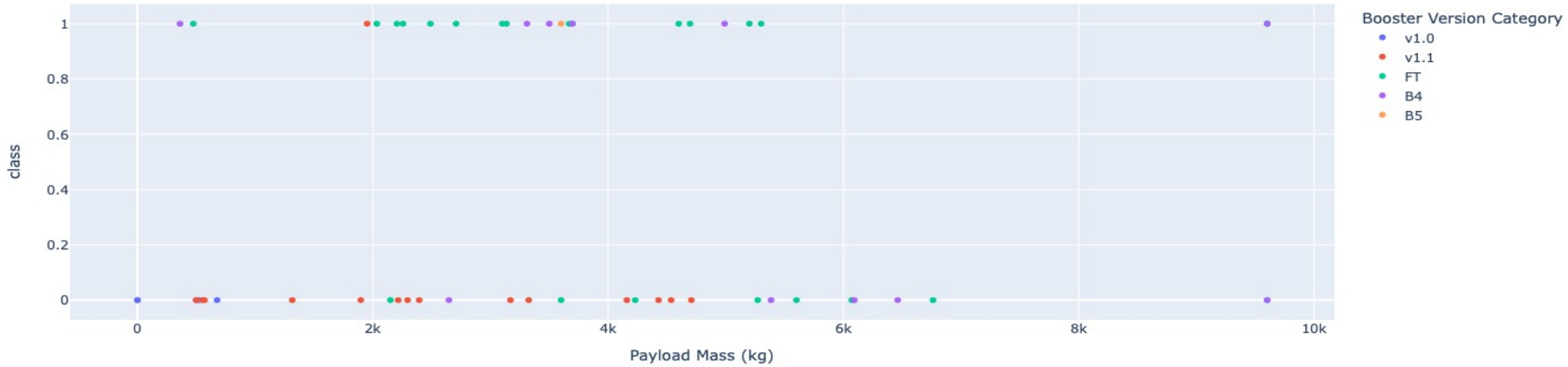
The blue and red pie charts show higher success rates compared to the green and purple sites. KSC LC-39A has the highest success rate, accounting for 41.7% of successful launches, while CCAFS SLC-40 has the lowest success rate, accounting for only 12.5%.

Success and Failure for KSC LC-39A



This launch site, KSC LC-39A, had the highest number of successful launches on our previous chart, and we can see on this chart that it has an amazing success rate of 76.9%.

Payload vs Outcome



We can observe how payload mass and booster version affect the launch outcome, color-coded for all sites.

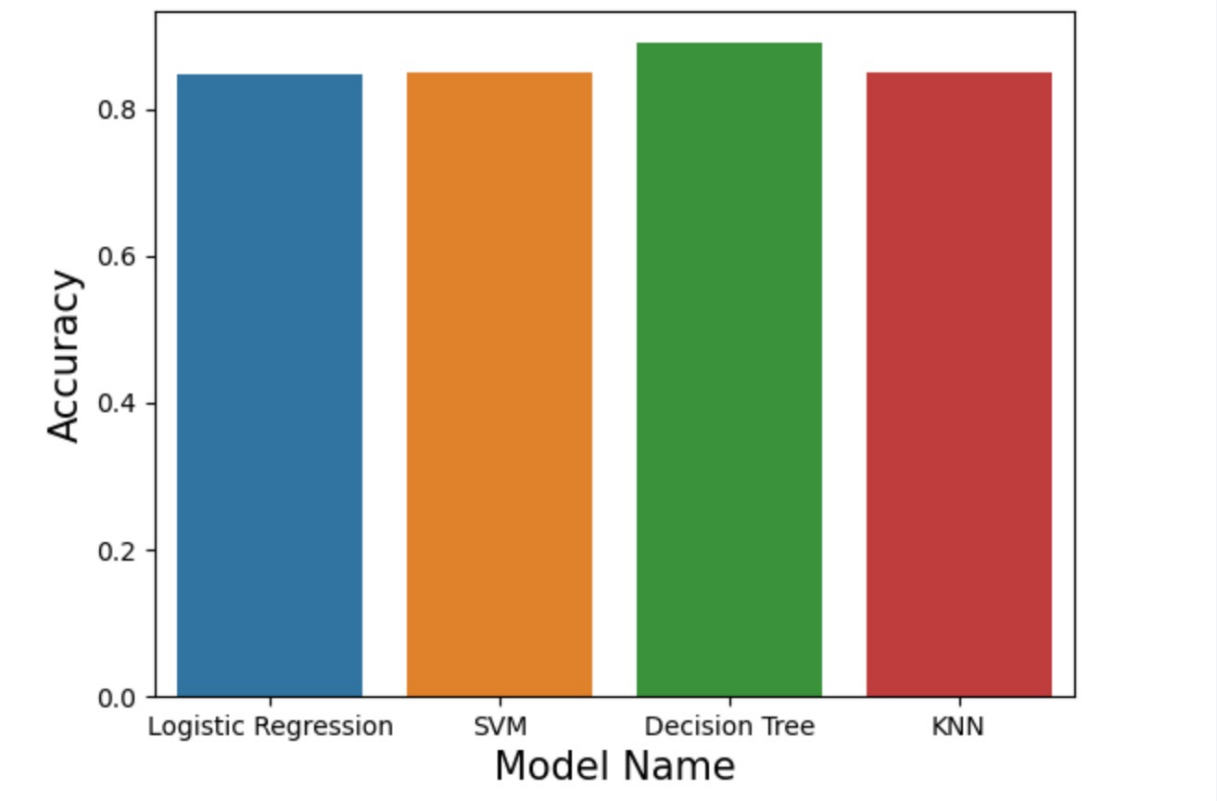
Most successful launches within the payload mass of 0-6000kg use the FT booster category (green), while most of the failures occur with the v1.1 booster category (red). Light payloads below 2000kg have a low success rate. The B4 booster category also has a decent success rate.

Section 5

Predictive Analysis (Classification)

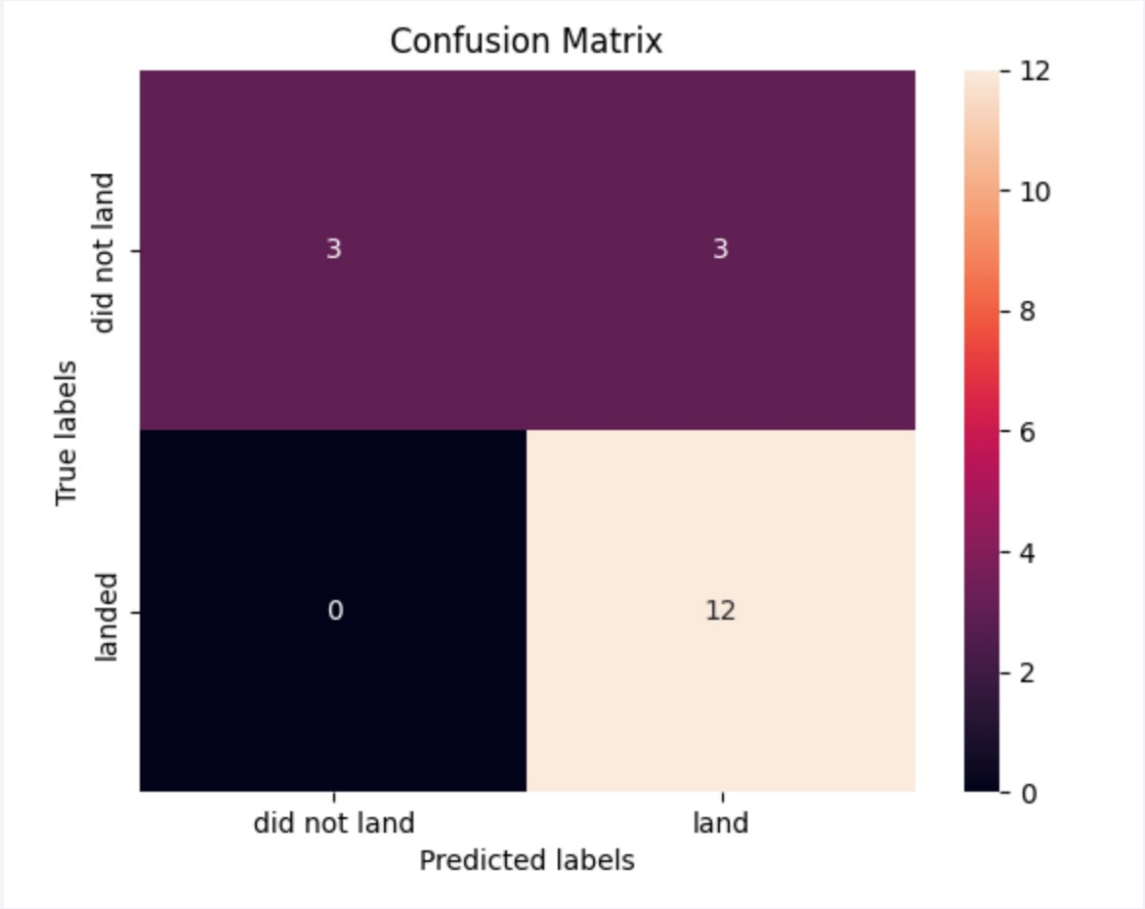
Classification Accuracy

The accuracy of four classification methods were compared and Decision Tree had the highest accuracy (0.89), followed closely by the other methods.



Confusion Matrix

The best performing model was the Decision Tree. It correctly predicted all the labels that landed and only had one false positive. Out of the six labels indicating a failed landing, 3 were predicted correctly. The confusion matrix for the model shows that it correctly predicted all 12 successful landings (True Positive), and correctly predicted only 3 out of 6 unsuccessful landings (True Negative), but incorrectly predicted 3 unsuccessful landings as successful (False Positive).



Conclusions

- Launches to ISS, SSO, and LEO with lighter payloads (up to 6000kg) have the best success rates. VLEO has also been successful in recent flights.
- KSC LC-39A is the launch site with the highest success rate, while VAFB SLC-4E performs well with lighter payloads and CCAFS SLC-40 performs well with heavier payloads.
- The booster version category FT has a great success rate for payloads below 6000kg, while v1.1 has a poor success rate for the same mass range.
- Success rates have increased over the years and are correlated with the number of flights. ES-L1, GEO, HEO, and SSO are the most successful orbits.
- Launch sites are placed near transportation infrastructure and away from cities for safety.
- The best predictive method is the Decision Tree Classifier with an accuracy of 89%

Appendix

- Github: Capstone SpaceX

Thank you!