Winning Space Race with Data Science

Joe Rutledge
1 Nov 2022

Outline

Executive Summary

Summary of methodologies

- Our research begins with the collection of our data, via webscraping or the SpaceX API. Next, we will process the data into a useable format for our visualization, which we can use to gain insight on our research target. Finally, our processed data can be run through our classification models, which are assessed for greatest accuracy. This will determine for us our greatest chance of accurately predicting the results of an attempted landing.

Summary of all results

- Determination of the probability of successfully landing and recovering the first phase of a SpaceX rocket can best be performed by a Support Vector Machine model. Using this model, we can predict the results of an attempted landing with as high as 83% accuracy.

Full GitHub Repository: https://github.com/jcrutle/Coursera-Capstone-Project

Introduction

Launching rockets into space is an enormously expensive venture. To reduce the cost of repeated launches, the commercial entity known as SpaceX has been developing technologies that will allow it to recover the first phase of the rocket, which can then be reused at a later launch. Successful recoveries would lower the cost of launches significantly, making further launches more feasible and accessible.

- Problems we must solve
- How can we gather the data we need for this research?
- How can we process the raw data into quantifiable and useable data?
- How can we best visualize the metrics?
- Finally, what will best determine the success or failure of a launch?

Section 1

Methodology

Methodology

Data collection methodology

- Data was first collected from the SpaceX API, which the company provides for private research. We were also able to gather data from webscraping articles from Wikipedia.

Perform data wrangling

- Data was prepared for analysis first by parsing, then cleaning. Outcomes were enumerated and combined into one dataset, finally followed by feature label encoding.

Perform exploratory data analysis (EDA) using visualization and SQL

- Plain, query-able information was displayed for quick data analysis, and visualization was performed using line, scatter, and bar plots.

Perform interactive visual analytics using Folium and Plotly Dash

- Points were plotted on a map to show geographic informatics, visualization was also performed with Plotly Dash's quick tools like dropdown menus and slide bars.

Perform predictive analysis using classification models

- We split our data into testing and training groups, then several models were compared side by side for accuracy using that information.

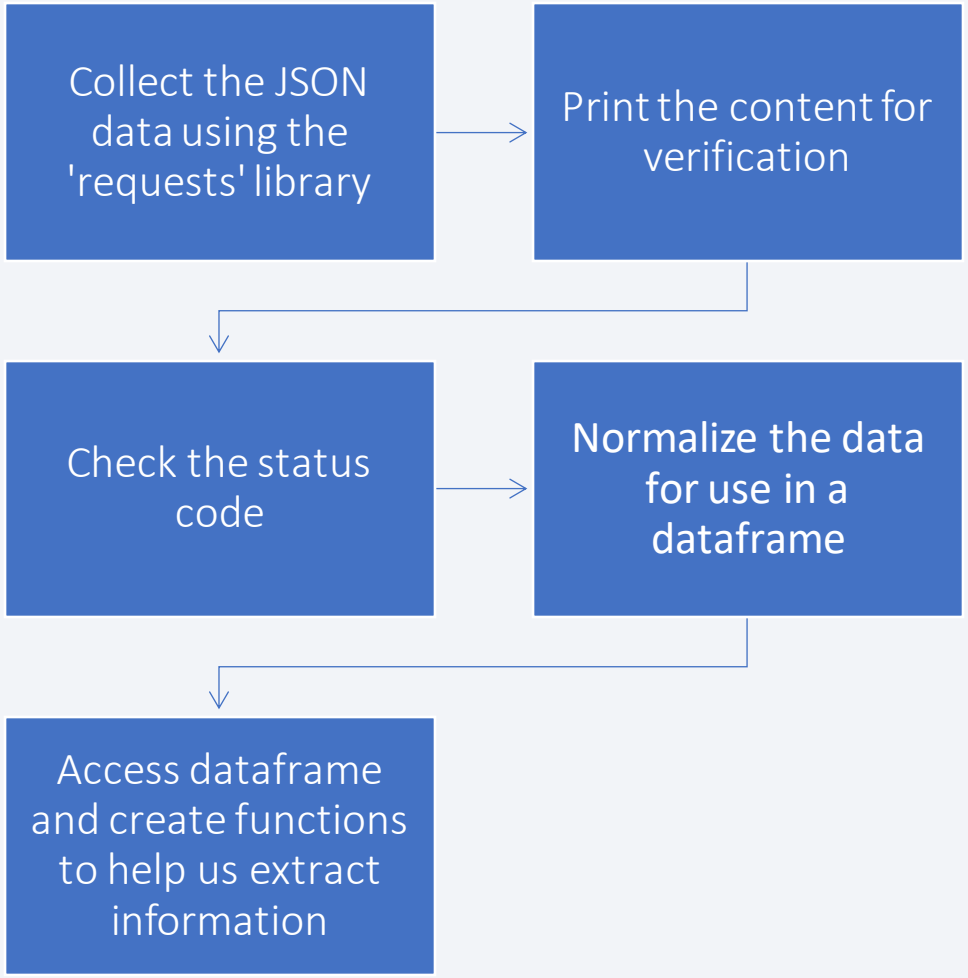
Data Collection

Data was collected in two ways: Webscraping and the SpaceX API.

- Webscraping allows us to conduct conduct research using publicly available data. This can give us quick access to our information, but also often leaves it in a very messy state.
- The API that SpaceX provided us with a very quick and easy way of obtaining our data, and as an added benefit, presents the data in a tidy way, reducing time needed for processing our data.

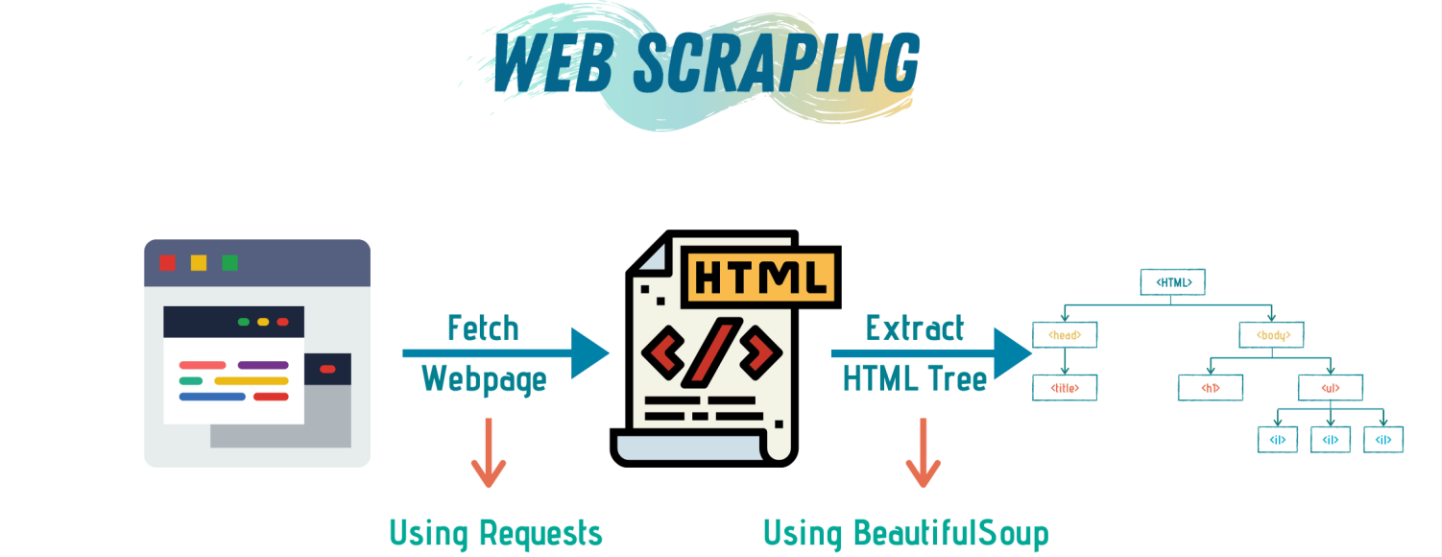
Data Collection – SpaceX API

- Given the SpaceX API, we made a REST call to collect our data, which we found in a JSON format. We were immediately able to normalize the JSON data using the Pandas 'json_normalize' method, allowing us to access a parsed dataframe right away.
- GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/SpaceX%20Data%20Collection%20API.ipynb



Data Collection - Scraping

- Scraping is performed first by fetching the webpage from the website in raw HTML using the 'requests' method. By employing the 'BeautifulSoup' library to parse the raw HTML, more useable information, such as tables can then be extracted.



GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb

https://towardsdatascience.com/a-step-by-step-guide-to-web-scraping-in-python-5c4d9cef76e8

Data Wrangling

- After looking through our data, we calculated the missing values in our dataset, as it is crucial that all our data can be processed to avoid negatively affecting our models' outcomes. We then identified column data types for our own awareness. Finally, we refined the landing outcomes (success/failure) into a column called "Class" (0/1), which now identifies the outcome in a quantifiable manner. When we knew the information we needed, we removed the rest to keep our processes efficient and data clean.
- Calculating the mean of Class will provide us with the rate of a successful landing, which we could in turn apply to features to gain insight on their impact on the success rate.

GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

- By comparing several independent variables, we can attempt to understand their effects on landing outcomes.
- By visualizing payload mass, flight numbers, and launch sites, we can see that greater payload mass seems to decrease our chance of success, higher flight numbers seems increase our chance of success, and that the KSC LC-39A and VAFB SLC-4E launch sites have the greatest rates of success. We can also see from our data that successful landings generally increase year to year.
- Testing some other features, such as orbit type, can inform our understanding of the data, even though they may not have a large impact on the outcome.
- GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/EDA%20with%20Visualization.ipynb

EDA with SQL

- Our SQL queries allowed us to identify the launch sites where the launches were performed, returning four unique sites in separate geographic locations. Given locations, we were able to isolate data according to their sites.
- We also queried for basic information such as: the date of the first successful landing, total number of landing attempts, and the ranking of every kind of landing outcome according to frequency of occurrence.
- Further SQL queries present opportunities to isolate data that fit our given criteria, giving the opportunity for further data exploration.
- GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- To help us with visualization, we created a Folium map with the launch sites marked with circles and labels with the sites' map coordinates.
- We also created cluster objects at each launch site which indicate each site's landing outcomes. Each outcome is represented by a colored icon (green for success, red for failure).
- Lastly, for good practice, we measured the distance of one of the launch sites to the nearest coastline and labeled the destination with its distance from the site.
- GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We created a dashboard that quickly renders visual data according to our selection. The first is a pie chart displaying the distribution of successful flights across the launch sites. We can further change the selection from a dropdown menu to display the success rates (as well as outcomes when the user mouses over the results) of each launch site.
- Lastly, we created a slide bar that can variably display the rocket boosters' landing outcomes according to payload mass (kg).
- Creating these visualization tools provides quick access to data by removing the need to manually search for it within the data or create separate diagrams for each datapoint.

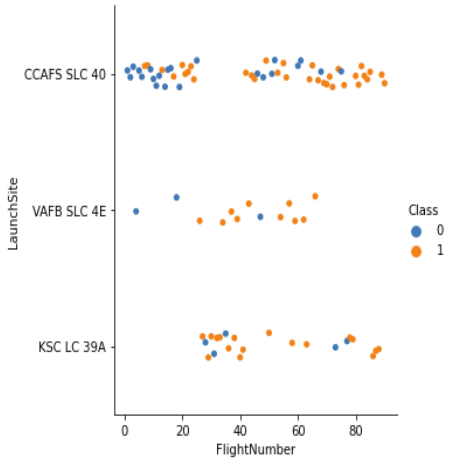
- GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/spacex_dash_app.py

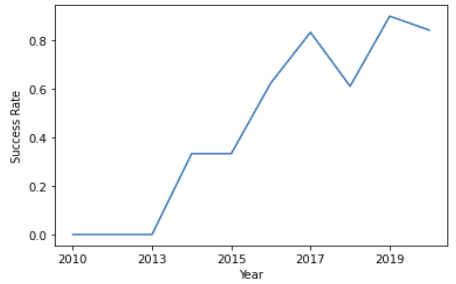
Predictive Analysis (Classification)

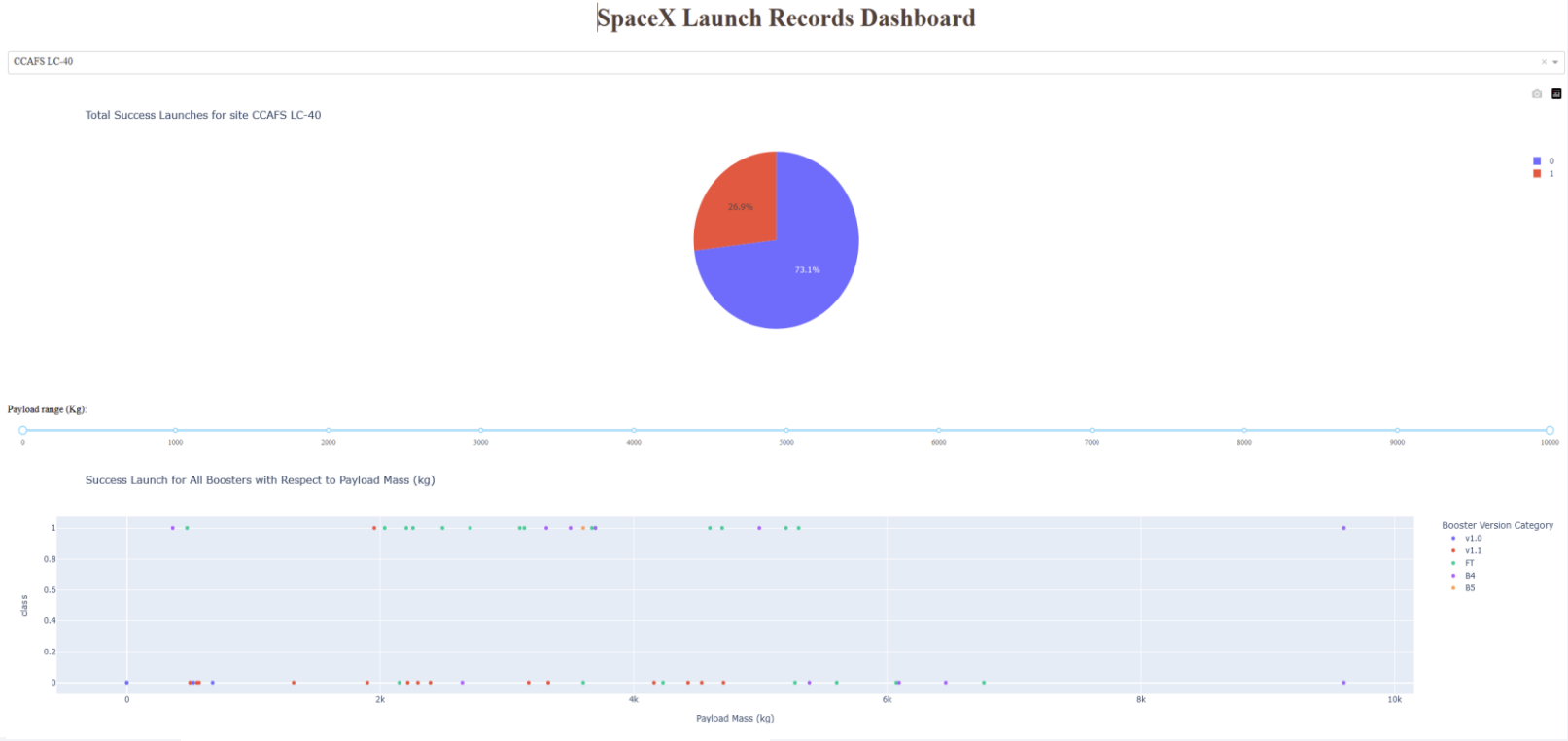
- To be able to best predict the outcome of a landing, we need to determine the model that is best able to predict the outcome based on training and testing data.
- In creating our models, we must first prepare our data for consumption. We standardize the data with the 'fit_transform' method, then split the independent variables (our features - X) and dependent variable (landing outcome - Y) into two groups for training and testing.
- We then select the models we would like to test. We chose logistic regression, support vector machine, decision tree, and K-nearest neighbor.
- Each of these algorithms is first fit to our data (X_train,Y_train), the results of which we will use to determine the model's best parameters. After determining the best parameters, we can determine the accuracy. We will know the most accurate model for our prediction when we obtain the accuracy of the models while they are using the best parameters.
- Using this flowchart, we can determine that all our models share the highest predictive accuracy at 83% accuracy.

- GitHub link: https://github.com/jcrutle/Coursera-Capstone-Project/blob/master/Machine%20Learning%20Prediction.ipynb

Results







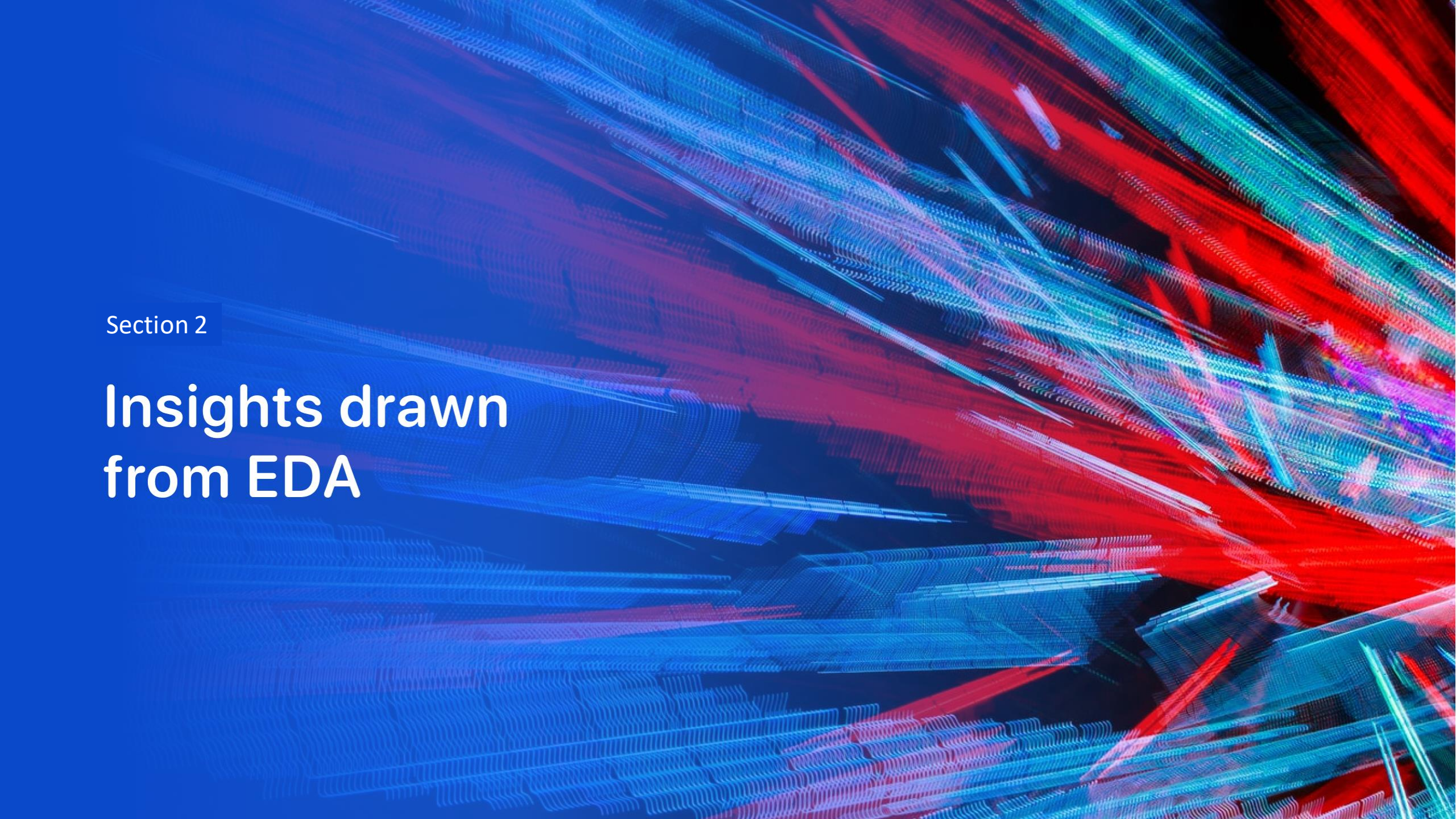

```
scores = [logreg_score,svm_score,tree_score,knn_score]

print(f"Logreg score: {logreg_score}")
print(f"SVM score: {svm_score}")
print(f"Tree score: {tree_score}")
print(f"KNN score: {knn_score}")
```

```
Logreg score: 0.875
SVM score: 0.9583333333333334
Tree score: 0.8472222222222222
KNN score: 0.8611111111111112
```

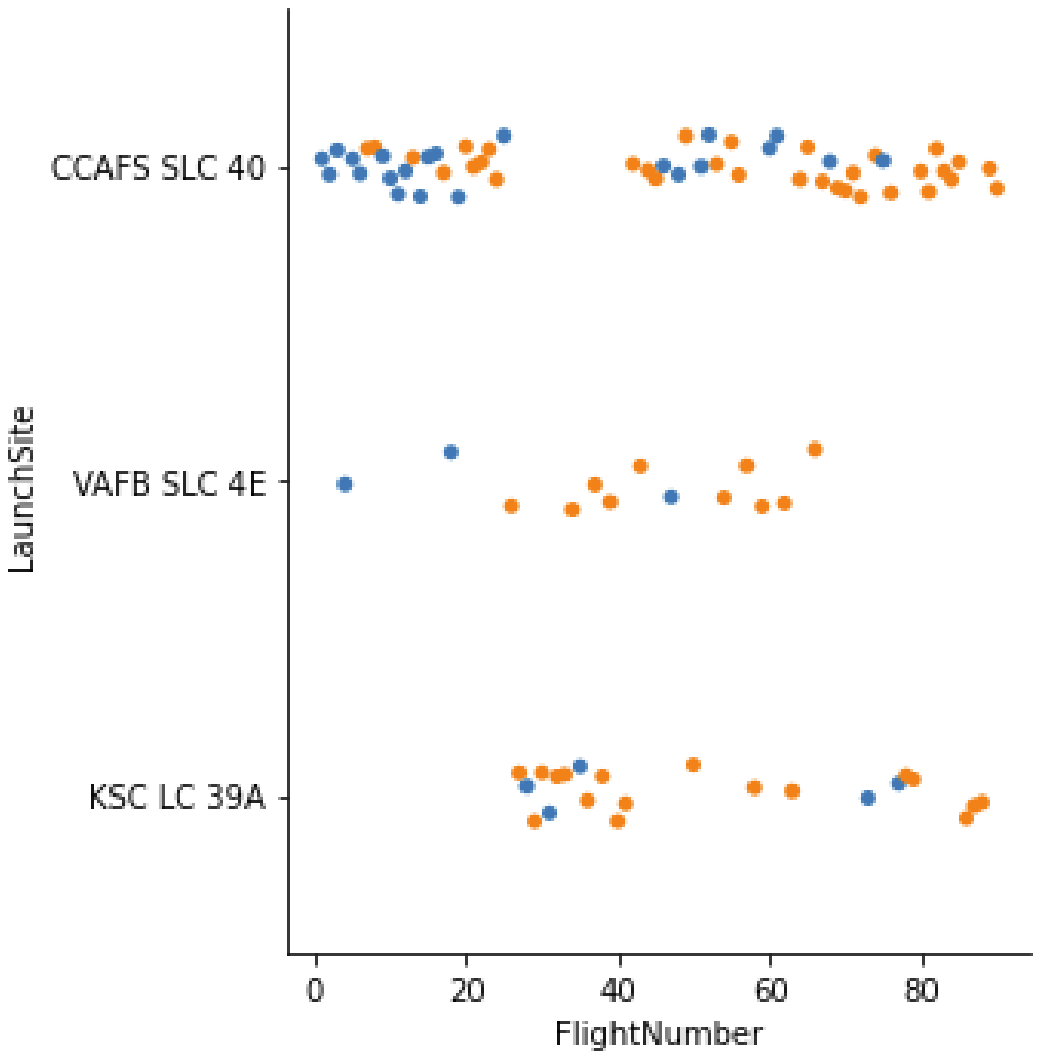
Section 2

Insights drawn from EDA

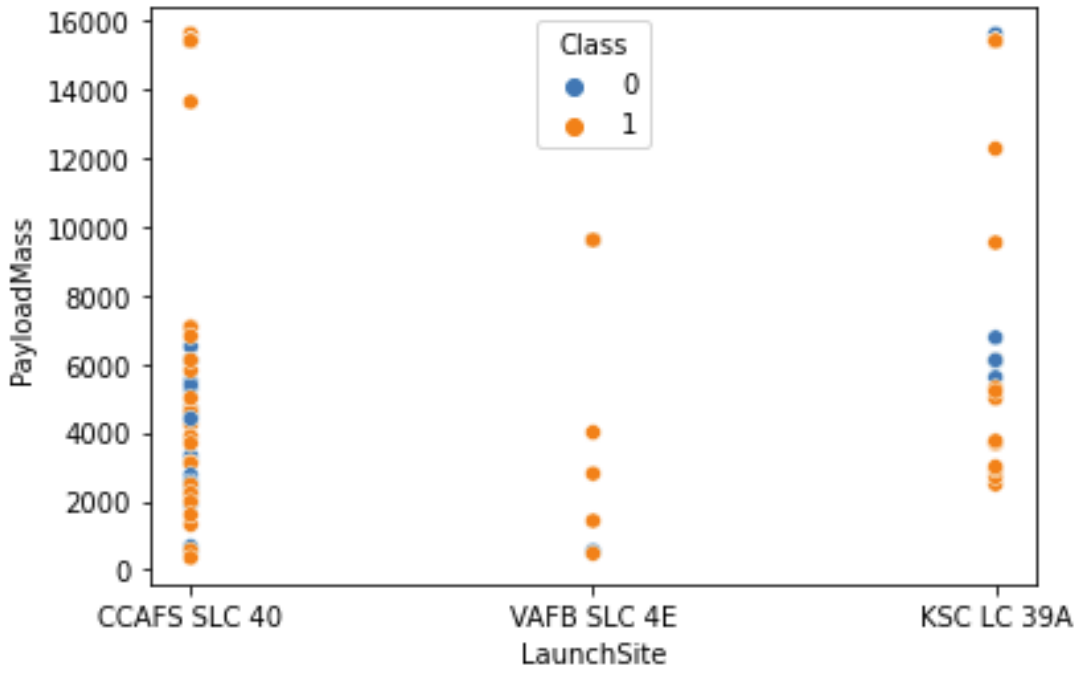


Flight Number vs. Launch Site

Here we see a visual distribution of the launches per site. We can see the most launches take place in Cape Canaveral, while the fewest take place at Vandenberg Air Force Base.



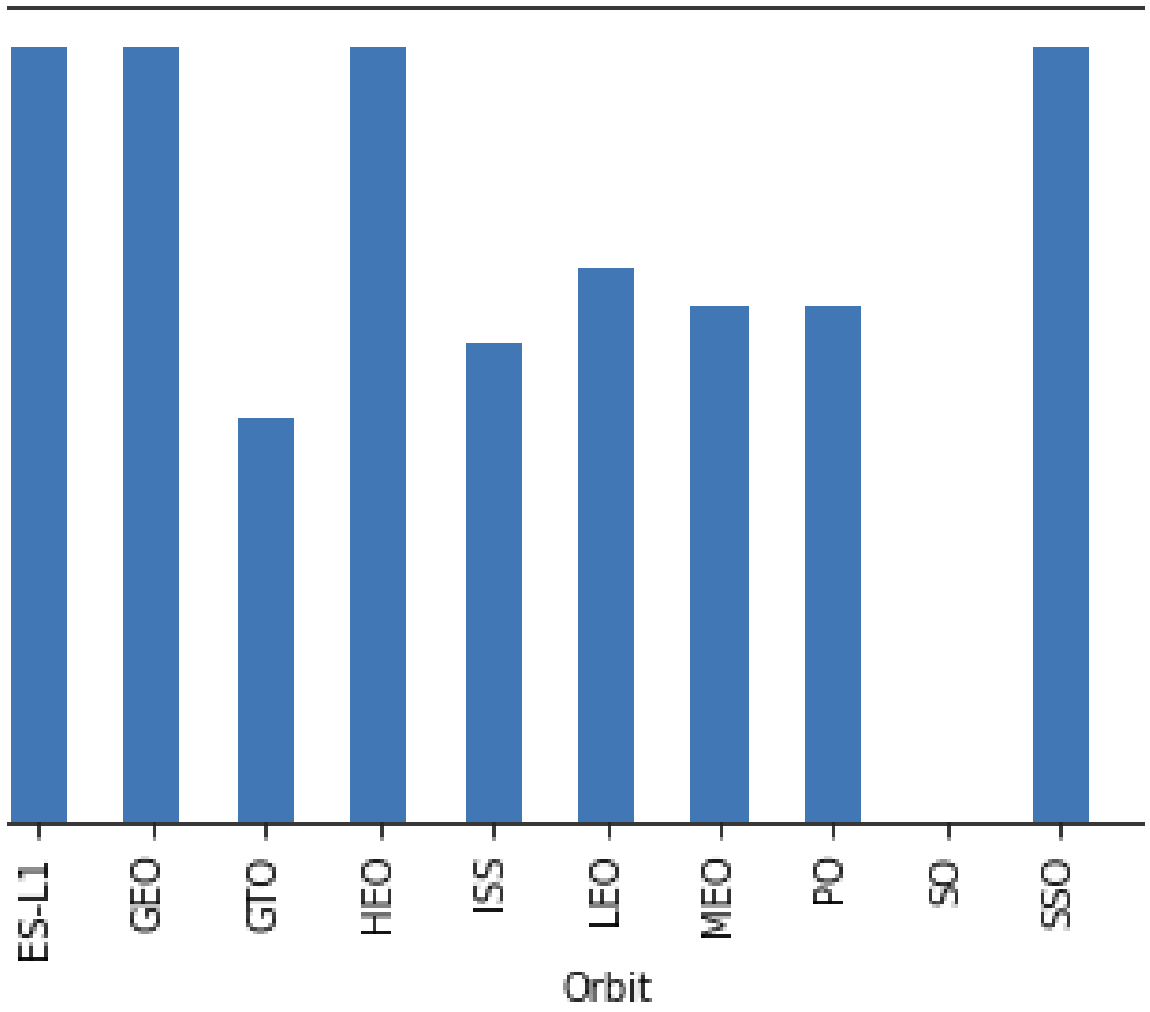
Payload vs. Launch Site



- There is a fairly even distribution between the launch sites in respect to the payload mass. The launches at Vandenberg tend to have lighter payloads.

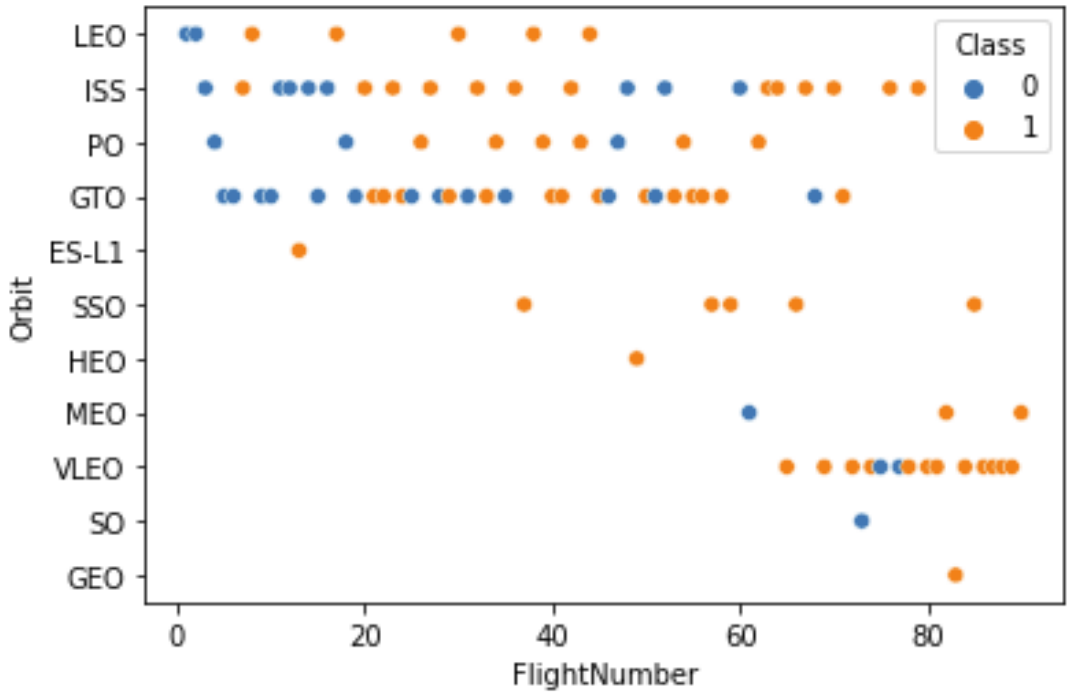
Success Rate vs. Orbit Type

There are several orbit types with 100% success rates. Given the number of flights in each orbit type, we could determine whether the data is trustworthy or possibly an outlier.

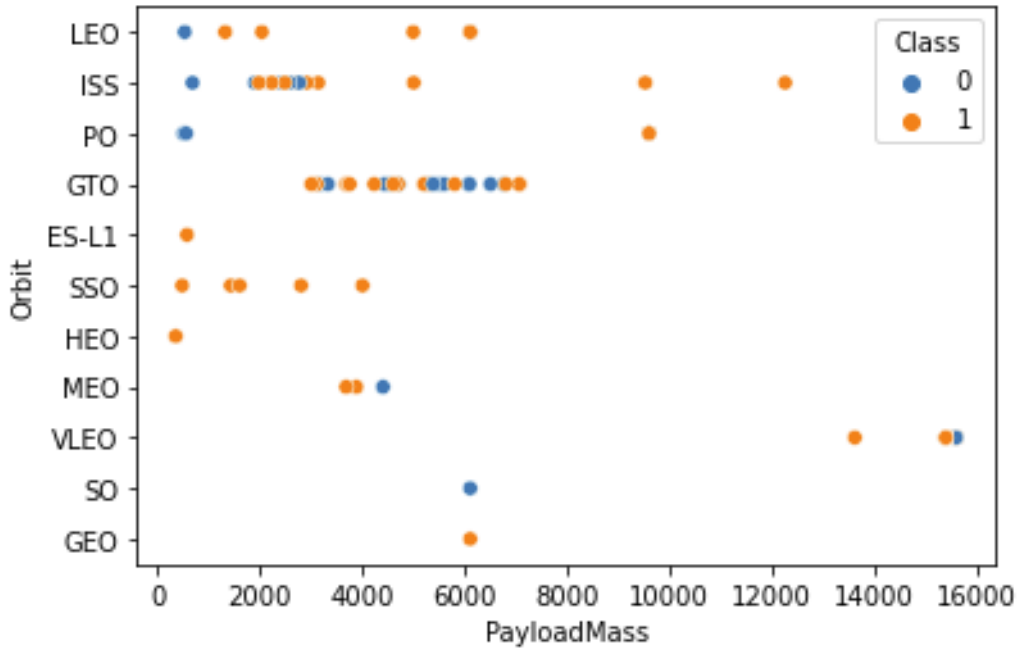


Flight Number vs. Orbit Type

- Flight number should not be confused with the number of flights, it is the order of the flights. Therefore, there is exactly one point for each flight number along the x-axis. The data shows that in the later flights, the rockets tend to orbit in the "VLEO" orbit type.



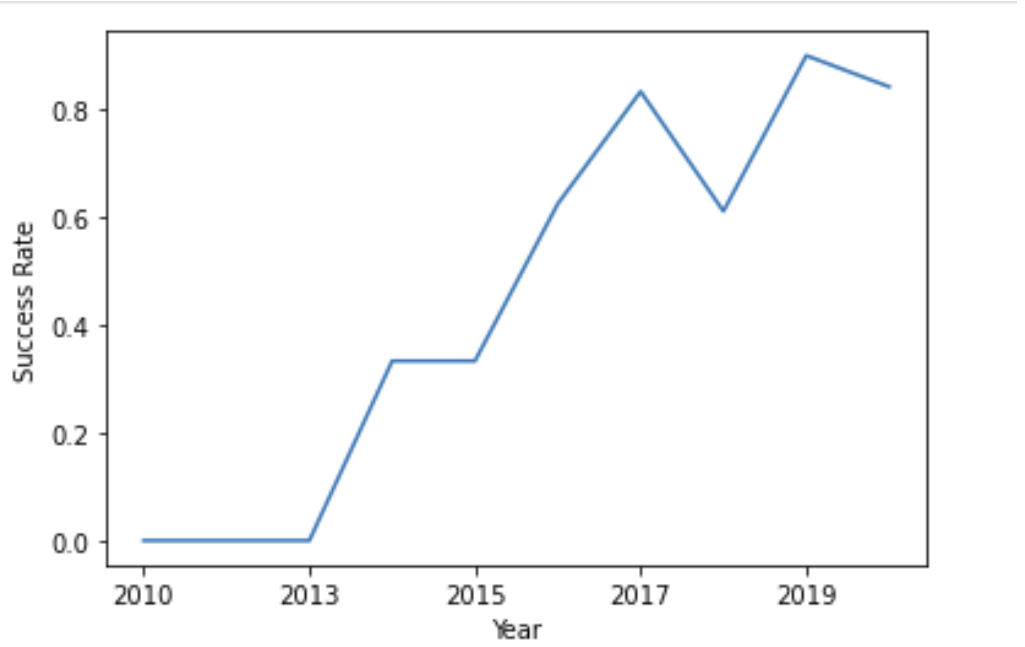
Payload vs. Orbit Type



We see that rockets with a typical payload mass largely have a "GTO" orbit type. Those with maximum or near-maximum payloads stay inside the "VLEO" orbit type.

Launch Success Yearly Trend

The rate of success landing outcomes typically increases year to year, with a couple exceptions in 2018 and 2020.



All Launch Site Names

Collecting all unique site names using SQL.

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

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEX;
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

Displaying all data from all Cape Canaveral sites.

```
%sql select * from SPACEX where LAUNCH_SITE like 'CCA%'
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
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	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

Calculating the total payload mass for NASA's launches.

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEX where CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

1
45596

Average Payload Mass by F9 v1.1

Calculating the average payload mass (kg) carried when using the F9 v1.1 boosters.

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEX where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

1
2928

First Successful Ground Landing Date

Querying for the date of the first successful landing.

```
%sql select min(DATE) from SPACEX where LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql select BOOSTER_VERSION from SPACEX where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEX)
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
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

Total Number of Successful and Failure Mission Outcomes

Calculating the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEX where MISSION_OUTCOME like 'Success%' or MISSION_OUTCOME like 'Failure%'
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

1
101

Boosters Carried Maximum Payload

Listing the names of the booster which have carried the maximum payload mass.

```
%sql select BOOSTER_VERSION from SPACEX where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEX)
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
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

2015 Launch Records

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

```
%sql select BOOSTER_VERSION,LAUNCH_SITE from SPACEX where LANDING__OUTCOME like 'Failure%' and year(DATE) = 2015
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

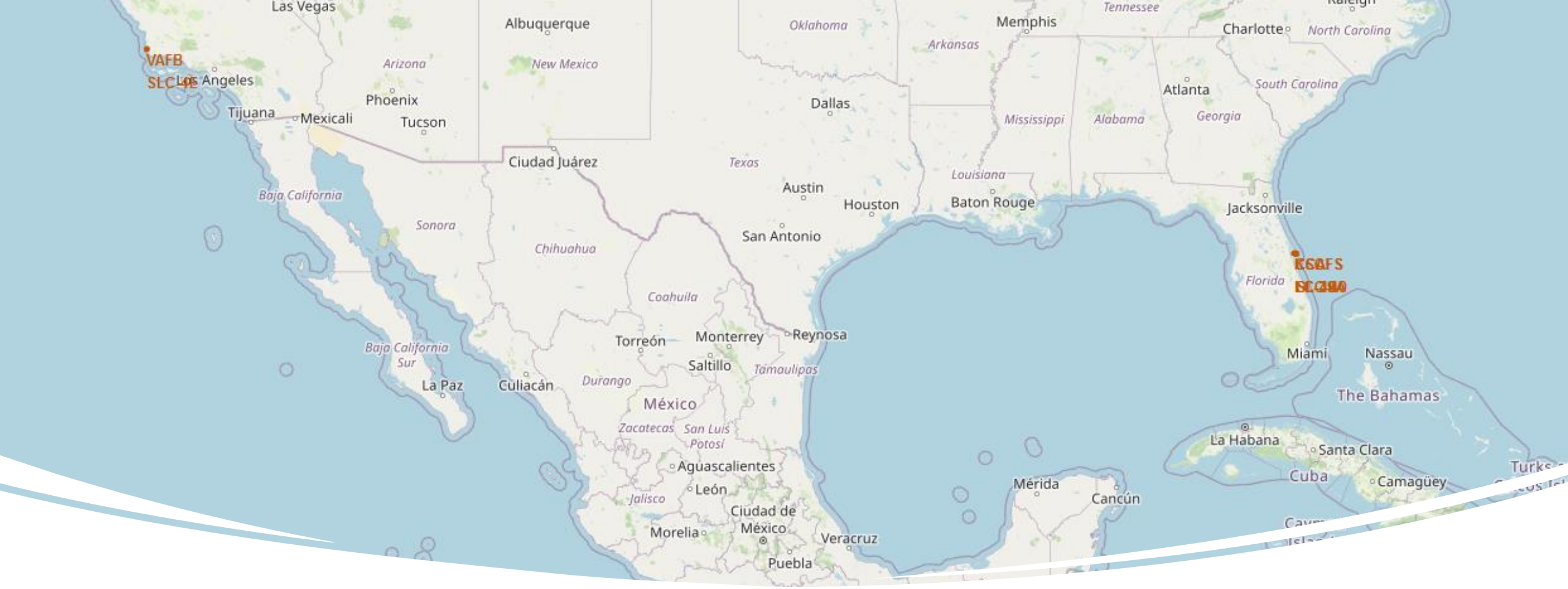
Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
%sql select LANDING__OUTCOME,count(LANDING__OUTCOME) from SPACEX where DATE between '2010-06-04' and '2017-03-20' group by LANDING__OUTCOME order by coun
```

```
 * ibm_db_sa://ycl47962:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/bludb
Done.
```

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


Section 3
Launch Sites
Proximities Analysis

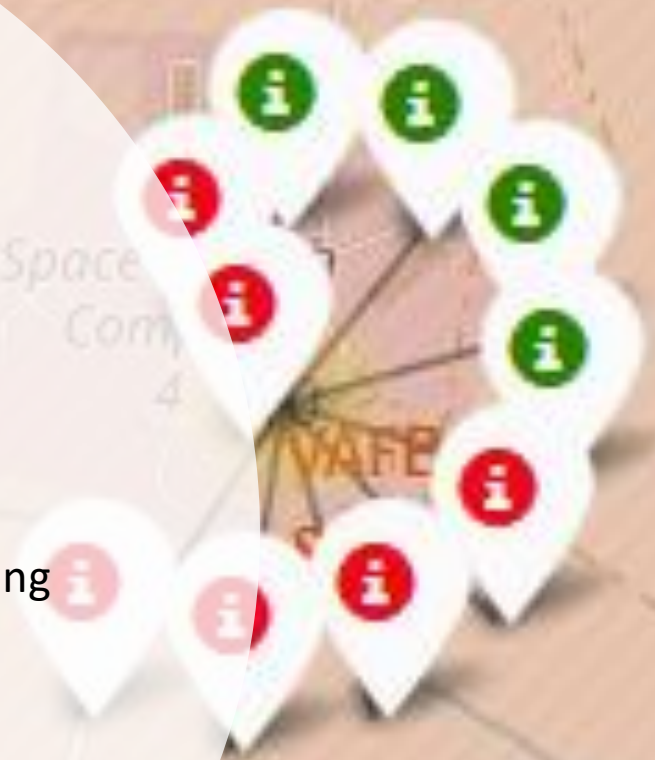


Launch Sites in Folium

- The launch sites are found in this screenshot taken of a map made and marked in Folium. One site is located in California, the other three are located in Florida.

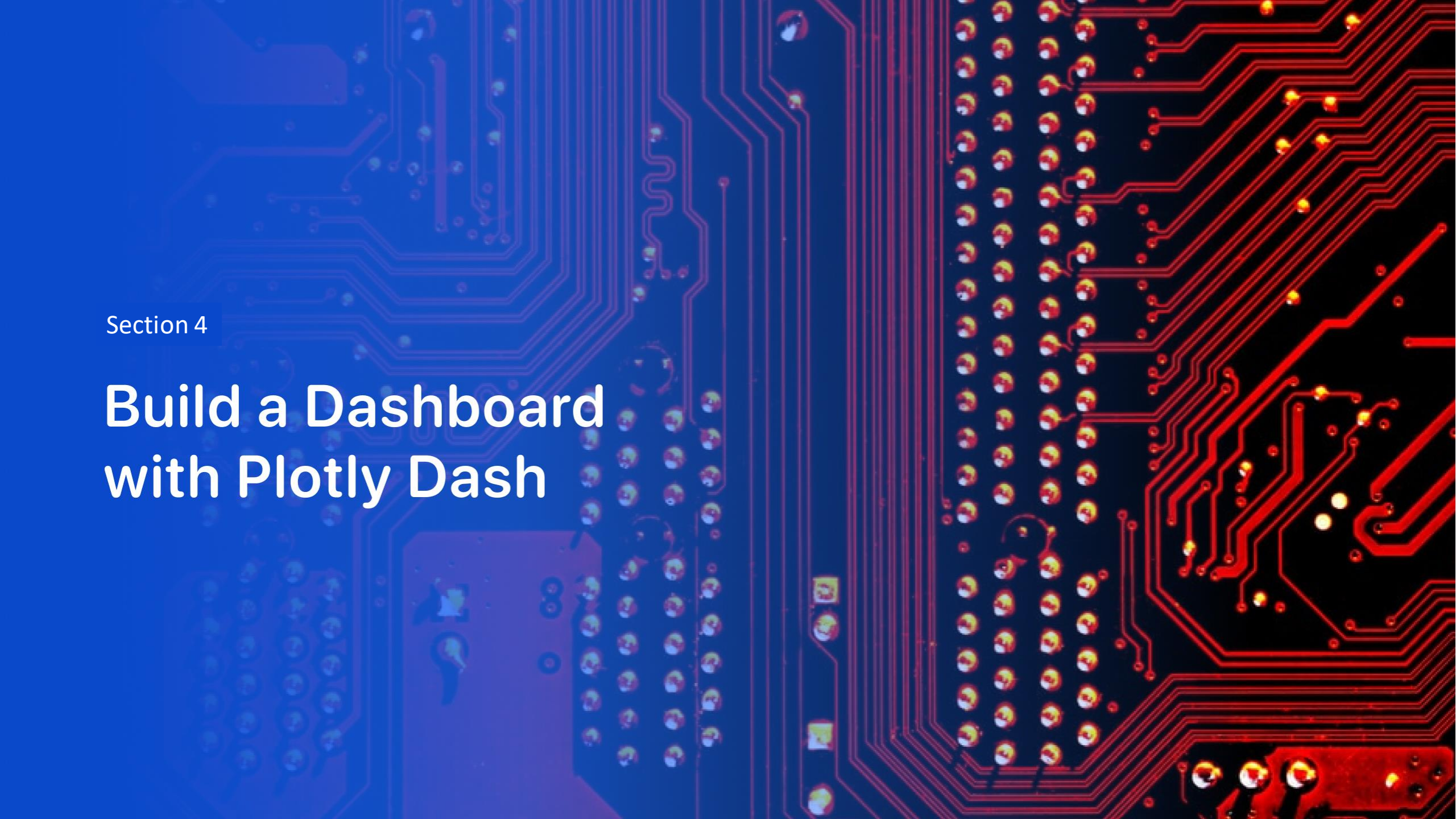
Color-labeled Launch Outcomes

- Icons are created and color-coded to indicate the number of flights and their landing outcome. Successful outcomes are colored green, while failures are colored red.

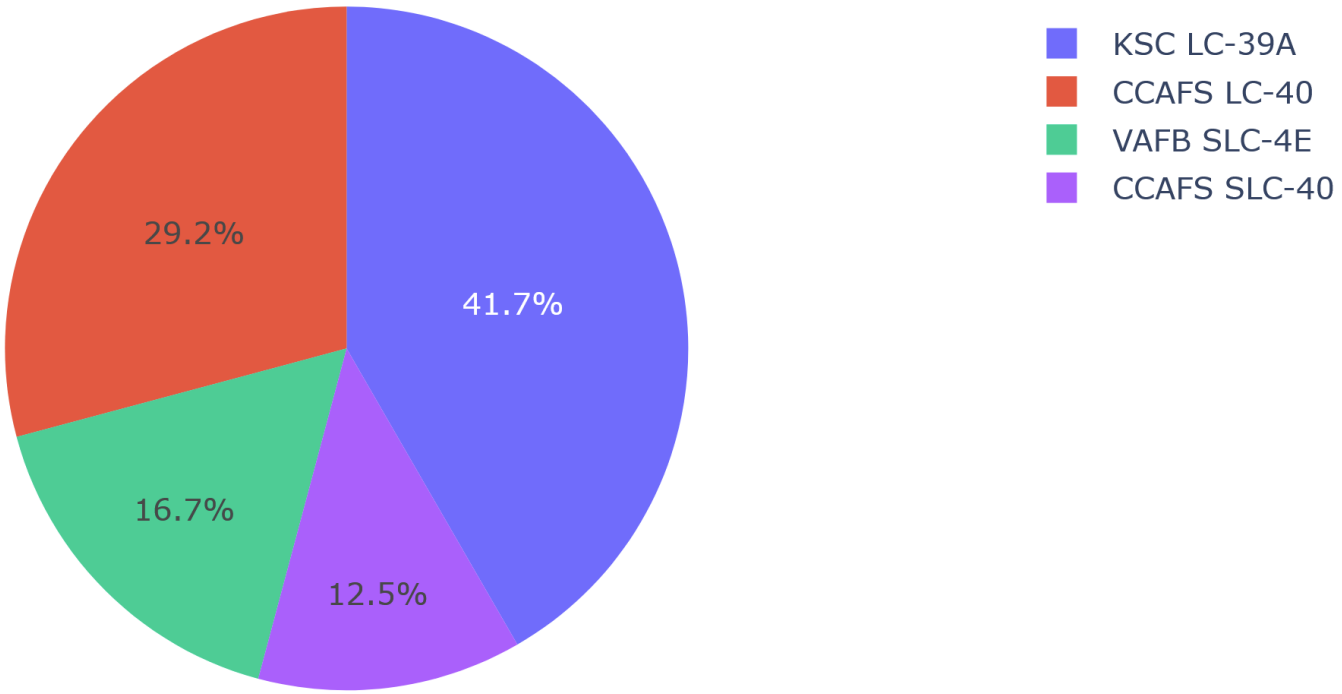


Launch Site to Proximity Distance

The distance measures 1.34km from the nearest coastline. Proximal locations such as coastlines and railways always show a strict minimum distance between the location and the launch site.

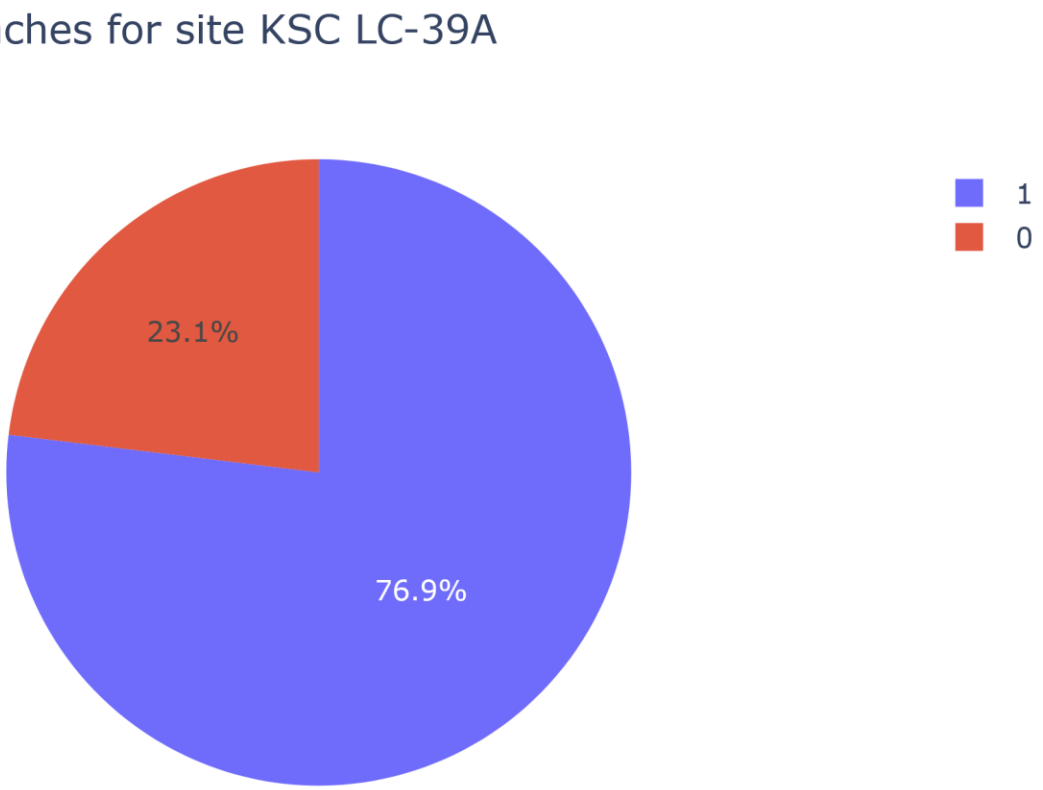

Section 4
Build a Dashboard with Plotly Dash

Plotly Dash Interactive Visualization



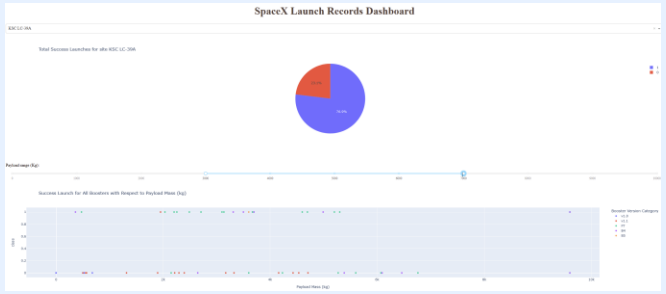
In the pie chart we see the distribution of successful landing outcomes across all the launch sites.

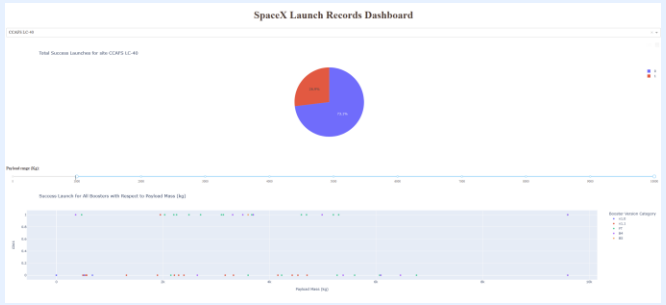
Launch Site with the Greatest Success Rate

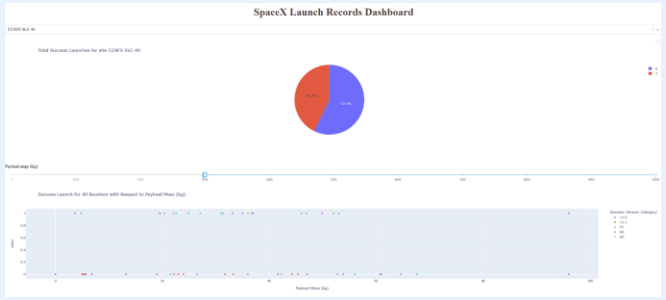


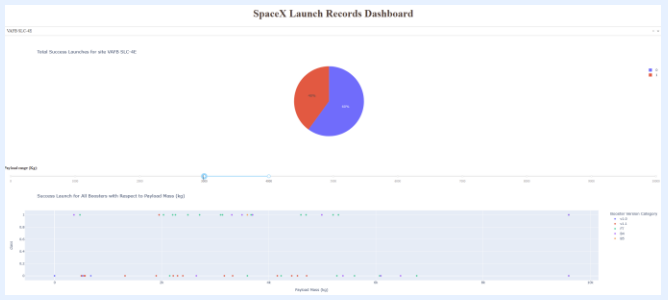
We can see that KSC LC-39A had the greatest success rate at 76.9%.

Variable Interactive Data









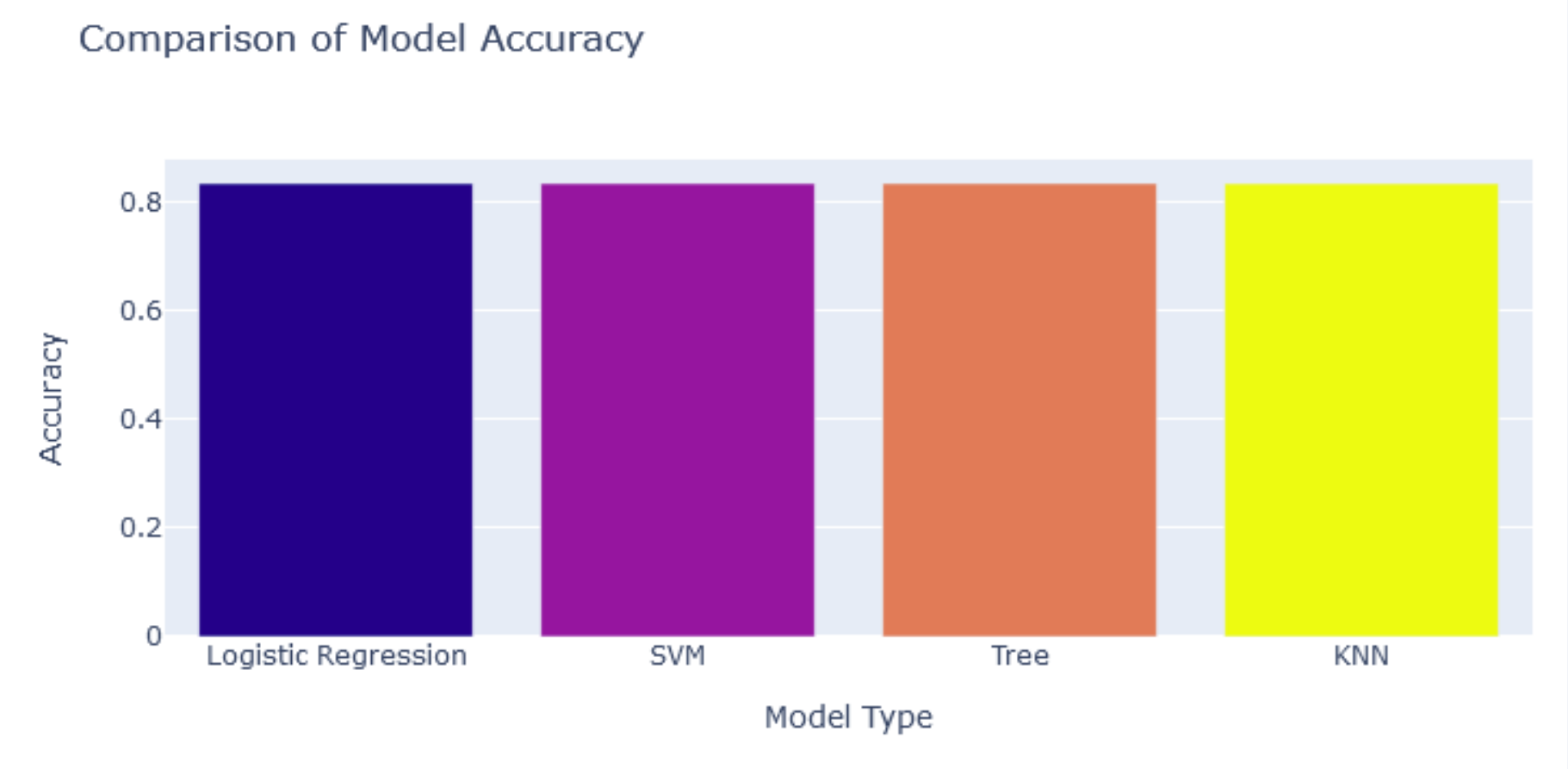
The data visualization can be molded according to the needs of the analyst. In this case, using a slide bar and dropdown menu.

Section 5

Predictive Analysis (Classification)

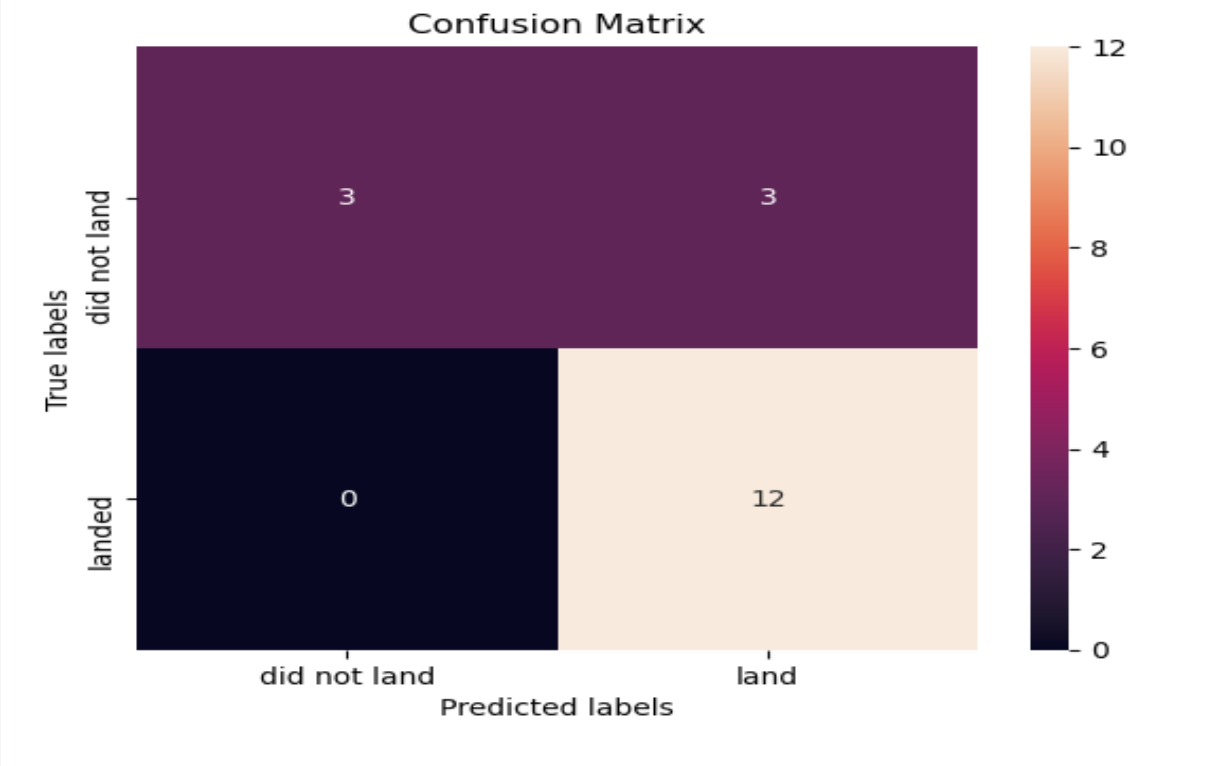
Classification Accuracy

We see that all our classification models have the same accuracy, all reaching 83.3%



Confusion Matrix

A confusion matrix showing the SVM model's predictions. It correctly predicted 15/18 outcomes. All models achieved the same results.



Conclusions

- All models have the same accuracy of predicting landing outcomes.
- The KSC launch site has the greatest chance of a successful outcome.
- The orbit types with the greatest chance of success are ES-L1, GEO, HEO, and SSO.
- Launches with less payload mass are typically more successful.
- Successful landing outcomes will generally improve year to year.
- Full GitHub Repository: https://github.com/jcrutle/Coursera-Capstone-Project

Appendix

The Jupyter notebooks must be updated to reflect accuracy changes in the GitHub files.

- Must fix the contents in slide 29
- Machine learning notebook – must change .score(Y_test,yhat) to .score(X_test,Y_test)

Thank you!