Winning Space Race with Data Science

Austin Brazille
8/02/2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of methodologies

Data Collection

Data Wrangling

Exploratory Data Analysis

Data Visualization

Predictive Analysis

Summary of all results

Gained info on launch sites, mission outcomes and relationship between variables and success

Interactive visualization analytic screenshots

Our machine learning model was able to predict successful landings with 83.3% accuracy

Introduction

- This report is the final assignment for the IBM Data Science Certificate. As a part of the assignment, I'm taking on the role of a data scientist at a new rocket company, SpaceY. Using data acquired from SpaceX, we set out to determine if we can estimate the cost of launches. SpaceX advertises Falcon 9 launches with a cost of $62 million when the first stage of their rockets can be reused. This highlights the importance of reusing the first stage and the overall focus of the report. If we can predict when the first stage can be reused, we can more accurately determine cost.
- Some answers we hope to find
  - How successful is SpaceX at landing their first stage?
  - Do they always need to land the first stage?
  - What factors have the most impact on a mission being successful?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API request and Webscraping HTML tables on wikipedia
- Data wrangling
  - Performed some exploratory data analysis to determine the label for training models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data sets are collected from the SpaceX API via request and webscraping HTML tables on Wikipedia
- SpaceX API
  - https://api.spacexdata.com/v4/rockets/

Launch data is requested then parsed using a GET request → Data is then filtered to only show launches using Falcon9 boosters → Missing values for payload mass are then replaced with the mean

- Webscraping
  - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Falcon9 launch wiki is requested from its URL → We then extract all column & variable names from the HTML table header → Then a dataframe is created by parsing the html tables

Data Collection – SpaceX API

Import Libraries	Pandas, Numpy, Datetime etc.
Define	Define helper functions to better use API (ie. def getCoreData(data))
Request	Request Response from API (response = requests.get(spacex_url) print(response.content))
Decode	Decode response content using .json() and turn into Pandas dataframe (data = pd.json_normalize(response.json()) print(data))
Clean	Clean the data by keeping only the features and rows we need (data = data[])
Create Dictionary	Create dictionary using the data we obtained (launch_dict = {})
Create Dataframe	Create Pandas dataframe from dictionary (data = pd.DataFrame(launch_dict))
Filter	Filter dataframe to Falcon9 launches (data_falcon9 = data[data.BoosterVersion == 'Falcon 9'])
Replace	Replace missing values in data (data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mpm))

- capstone/jupyter-labs-spacex-data-collection-api.ipynb at main · abrazille/capstone (github.com)

Data Collection - Scraping

Install & Import	Define	Request	Create	Check	Apply	Create	Parse	Create
Install and import the necessary software and libraries	Define helper functions to process the HTML Table (def date_time(table_cells):)	Request Falcon9 Launch HTML page response (data = requests.get(static_url).text)	Create a BeautifulSoup object from the HTML response (soup = BeautifulSoup(data))	Check contents of target table (first_launch_table = html_tables[2] print(first_launch_table))	Apply find_all() function to extract column names (first_launch_table.find_all('th'))	Create an empty dictionary with keys from extracted columns (launch_dict= dict.fromkeys(column_names))	Parse extracted tables into the dictionary	Create dataframe from newly built dictionary (df=pd.DataFrame(launch_dict)

- capstone/jupyter-labs-webscraping.ipynb at main · abrazille/capstone (github.com)

Data Wrangling

- In this phase we perform exploratory data analysis to find patterns in the data

Calculated the number of launches from each site (df['LaunchSite'].value_counts())

Calculated the number and occurrence of each orbit (df['Orbit'].value_counts())

Calculated the number and occurrence for the mission outcome by orbit type (landing_outcomes = df['Outcome'].value_counts())

Created a landing outcome label from the outcome column (landing_class = [] for outcome in df['Outcome']:)

- capstone/data_wrangling_jupyterlite.jupyterlite.ipynb at main · abrazille/capstone (github.com)

EDA with Data Visualization

- Scatter Plots - 6
  - Scatter plots show the relationship between variables which is called correlation. This gives us insight on which variables matter most for our research.
- Bar Chart - 1
  - Bar charts show the correlation, if any, between numeric and categoric variables.
- Line Chart - 1
  - Line charts show trends in variables. Line charts can help show behavior on a larger scale and make predictions for unseen data

- capstone/EDA with Visualization lab.ipynb at main · abrazille/capstone (github.com)

EDA with SQL

- To gain understanding of the SpaceX dataset, we performed 10 SQL queries
  - Display the names of the unique launch sites in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass.
  - 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.
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- [capstone/Complete EDA with SQL.ipynb at main · abrazille/capstone (github.com)]

Build an Interactive Map with Folium

- TASK 1: Mark all launch sites on a map;
  - circle and marker added to make launch sites easily viewable on map
- TASK 2: Mark the success/failed launches for each site on the map;
  - marker clusters are added to each launch site so we can add markers for all launch records. We made an outcome column to denote a green or red marker in the map based on launch success
- TASK 3: Calculate the distances between a launch site to its proximities
  - Markers are added to different proximity objects and then we added a line between launch site and the proximity object to display how close the two are in a straight line.

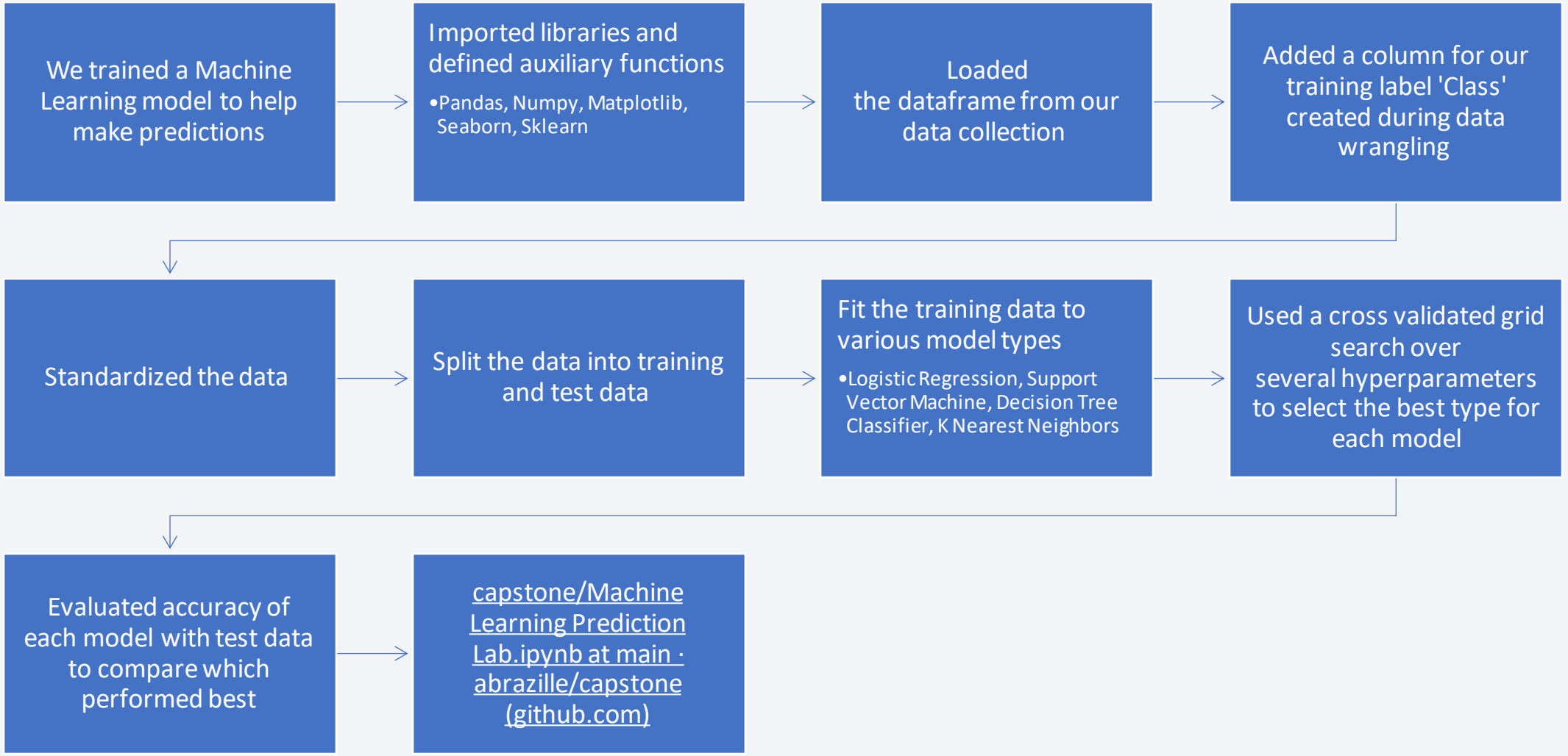
- capstone/Interactive Visual Analytics with Folium.ipynb at main · abrazille/capstone (github.com)

Build a Dashboard with Plotly Dash

- Added a dropdown list to enable Launch Site selection
- Added a pie chart to show the total successful launch count for all sites
- Added a callback function for `site-dropdown` as input, `success-pie-chart` as output
- Added a slider to select payload range
- Added a scatter chart to show the correlation between payload and launch success
- Added a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output

- capstone/spacex_dash_app.py at main · abrazille/capstone (github.com)

Predictive Analysis (Classification)

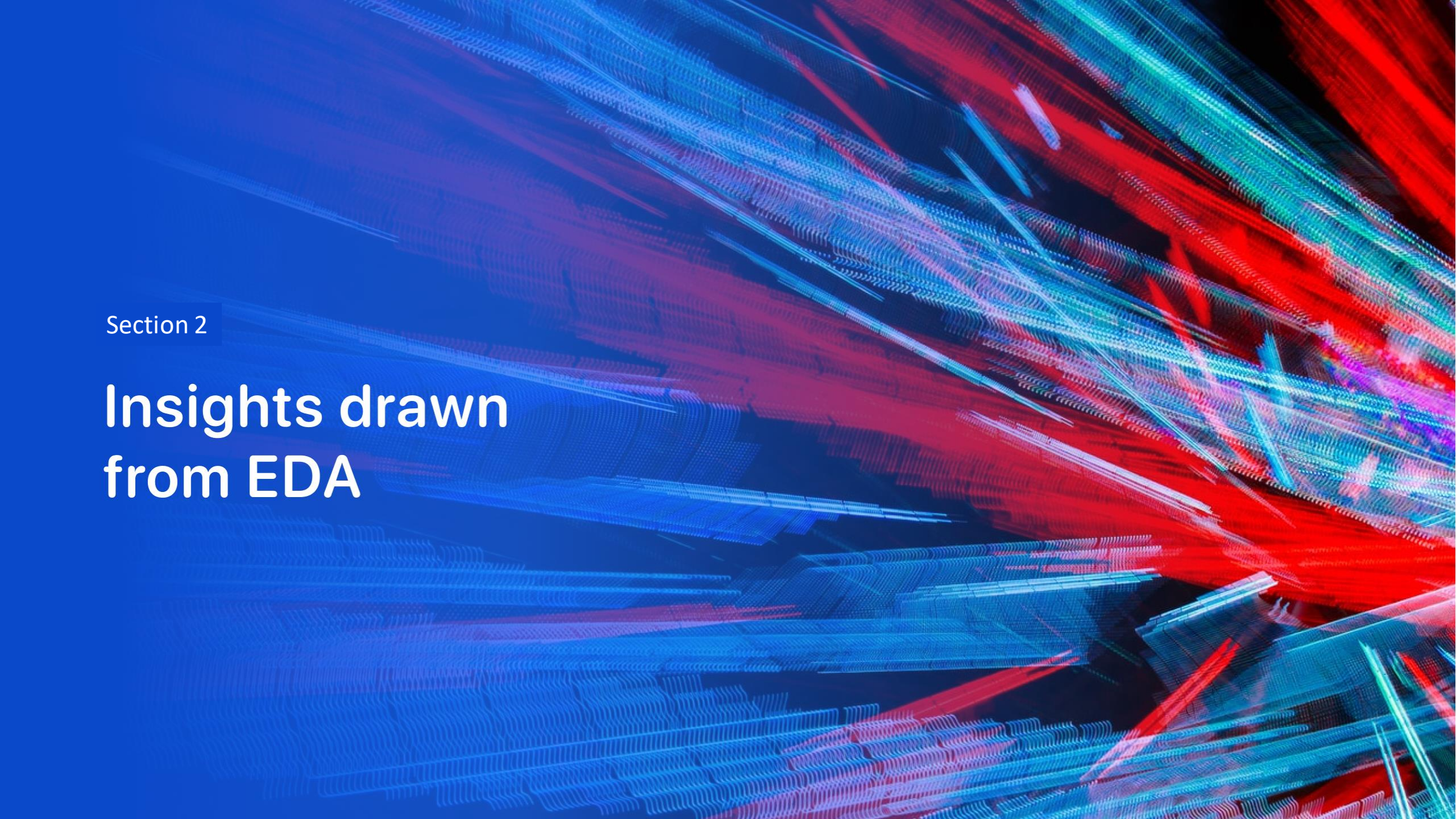


Results

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

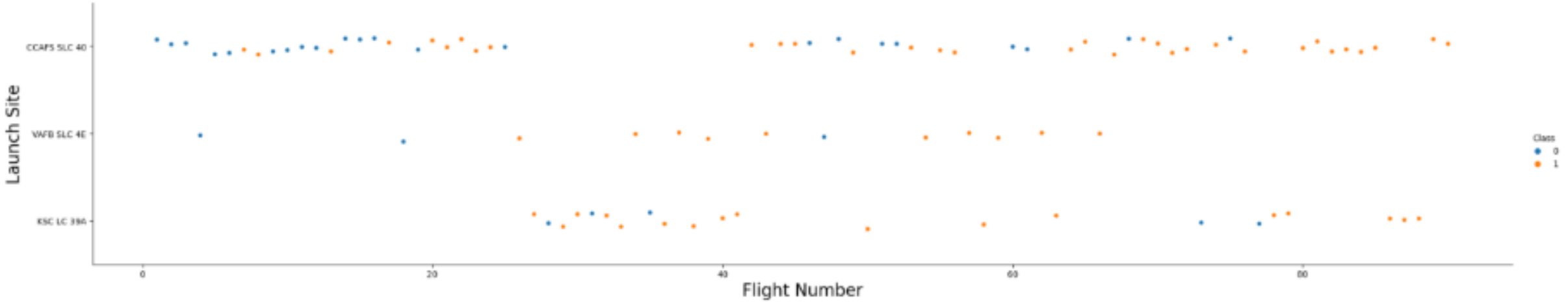
Section 2

Insights drawn from EDA



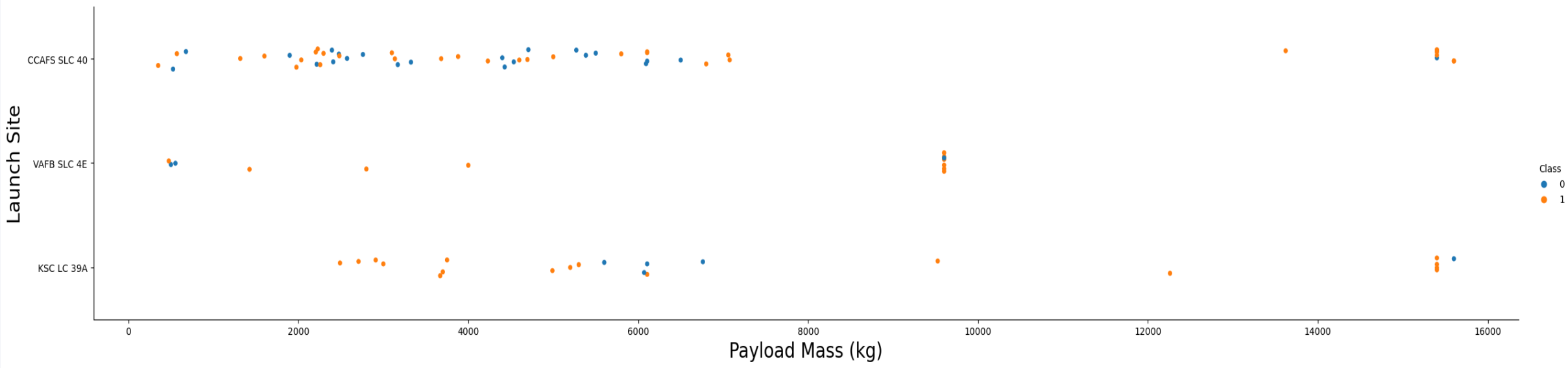
Flight Number vs. Launch Site

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class valu
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



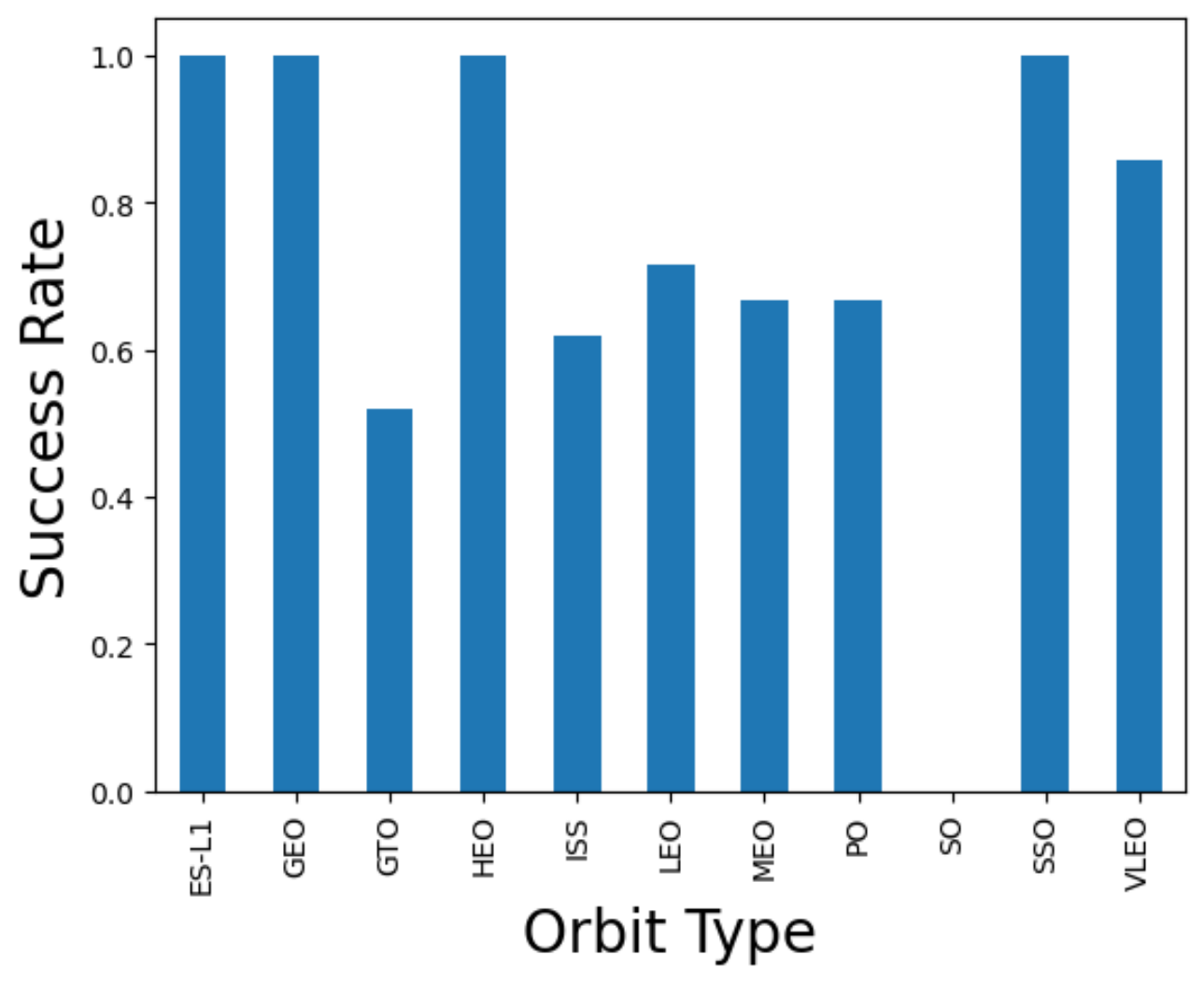
What this chart shows us is that the majority of launches came from CCAFS LC-40 and that they began to switch locations between launches once they became more successful.

Payload vs. Launch Site



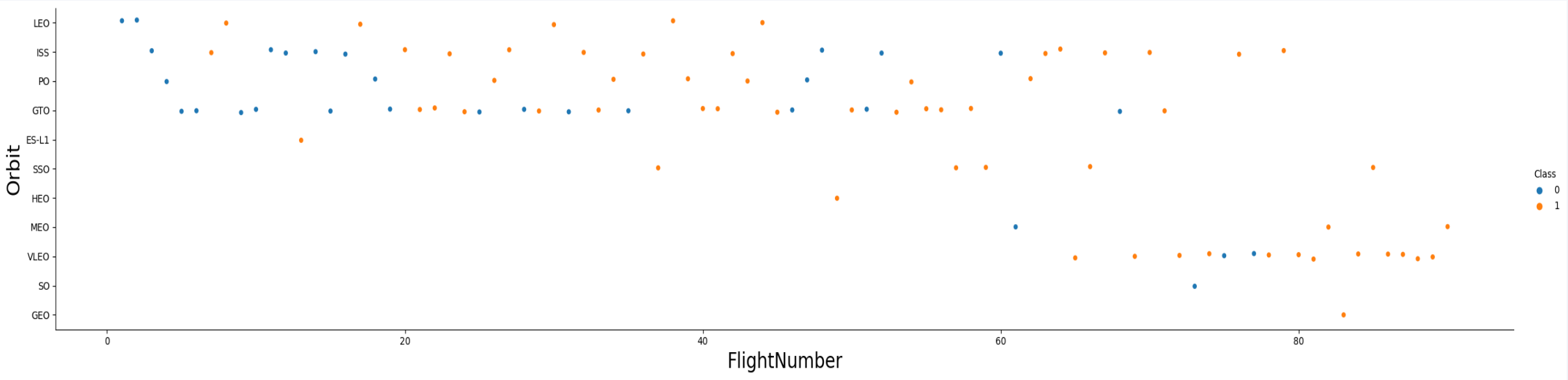
What we can observe in this chart is that launch site VAFB-SLC was never used for payloads above 10k kg and those heavy payload missions had a generally higher success rate than the lightweight launches.

Success Rate vs. Orbit Type



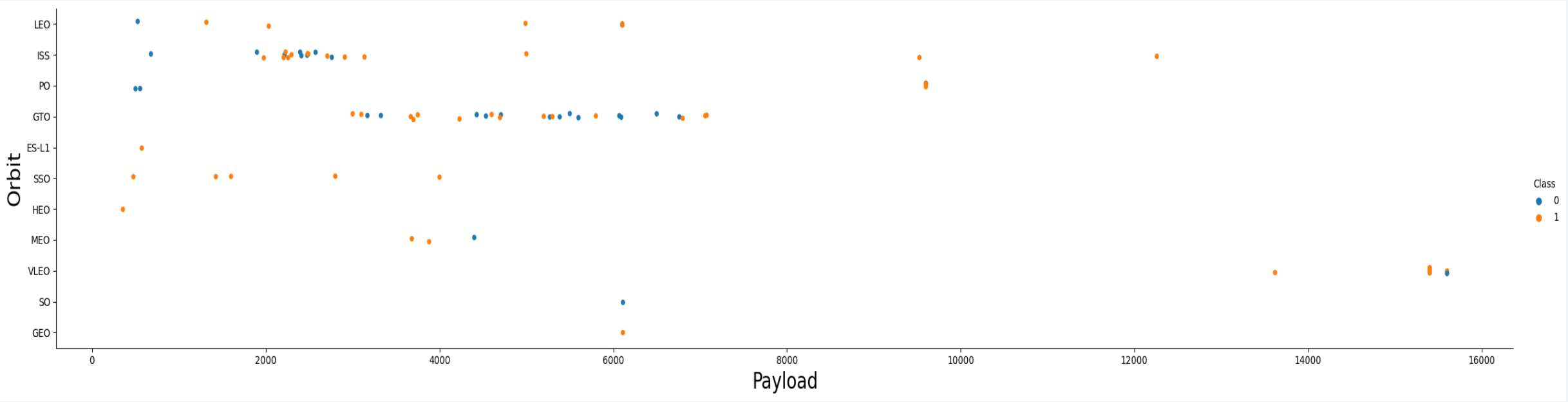
As you can see, our bar chart tells us that certain orbits have a 100% success rate while others are clearly less reliable.

Flight Number vs. Orbit Type



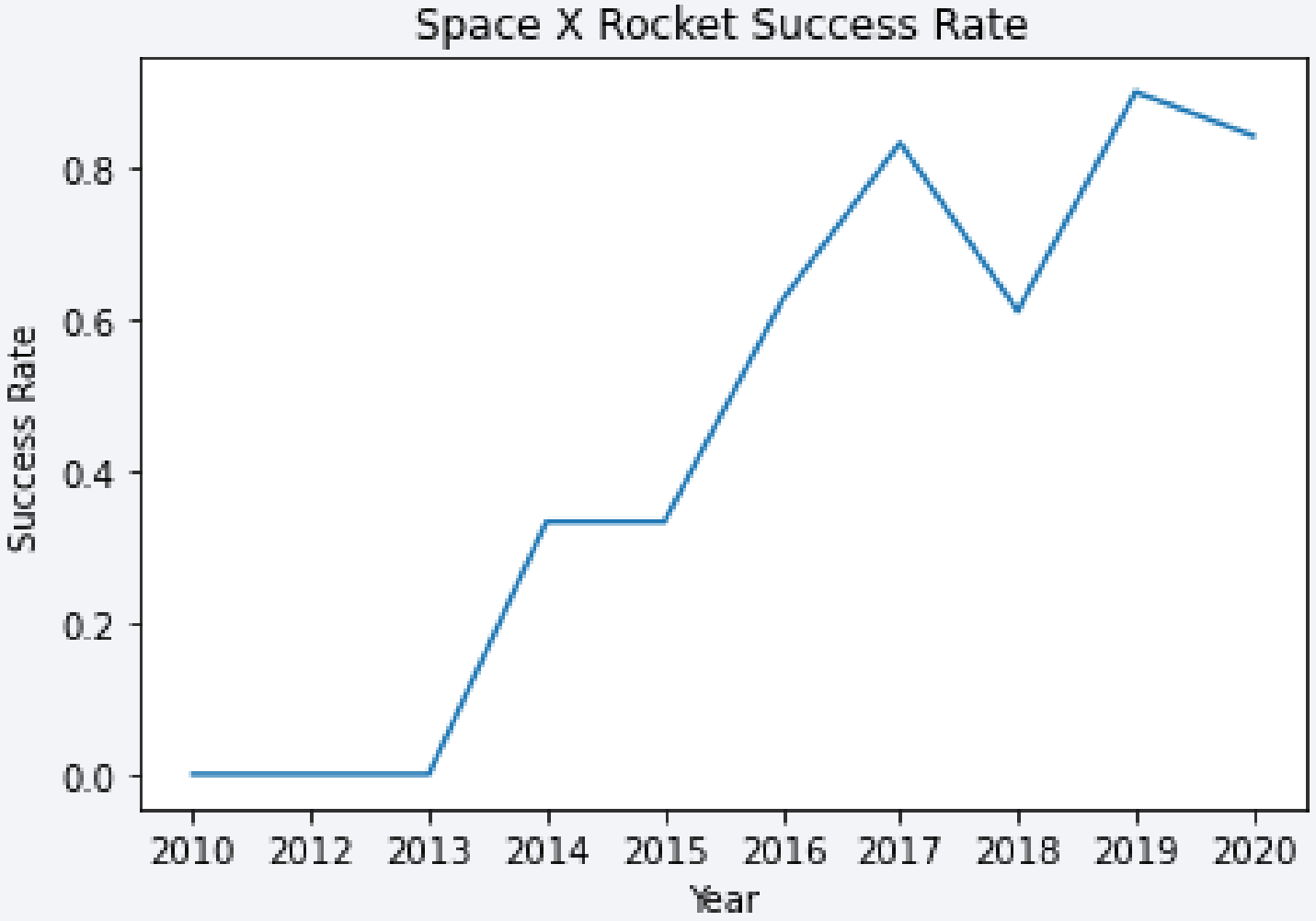
First thing we can note in this chart is that certain orbits weren't even attempted until 40, 50 or 60 flights had been taken. We can see some orbits became more successful with time, some were only attempted a few times or even just once and some orbits seemingly have no relationship between the amount of flights and successful outcomes.

Payload vs. Orbit Type



For PO, LEO and ISS have a higher successful landing rate with heavy payloads over lighter ones. Some orbits only get lighter payloads, like SSO, while others like VLEO only receive heavy payloads both of which are majority successful. However for GTO we cannot draw an immediate conclusion about the relationship because the results are so mixed.

Launch Success Yearly Trend



This chart is pretty straight forward in showing that the success rate of launches rose every year since 2013 except in 2018. So we can safely assume that they are improving with time.

All Launch Site Names

Using the DISTINCT function to access our table SPACEXTBL, we are able to list the names of each launch site without showing duplicates.

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

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL;
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Beginning with 'CCA'

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

```
%%sql
SELECT LAUNCH_SITE
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

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

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

Using the WHERE and LIKE functions, we filter the data to provide launch sites that start with 'CCA'. Our LIMIT function tells the database how many results to deliver, this case being 5.

Total Payload Mass

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596.0

We used the SUM function to add up payloads for all launches by NASA using WHERE to filter the data down to just NASA results

Average Payload Mass by F9 v1.1

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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version = 'F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

This query uses the AVG function to return the average payload mass only for boosters labeled F9 v1.1 due to our WHERE function

First Successful Ground Landing Date

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

Hint:Use min function

```
%%sql
SELECT MIN(Date)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (ground pad)';
```

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

MIN(Date)
2015-12-22

For this query we use the MIN function to return the earliest date

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)'
    AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

In this query we used the AND function to further define our filter function WHERE and narrow our results down to the selected payload mass range

Total Number of Successful and Failure Mission Outcomes

We utilized the COUNT function to add up our mission outcomes which came out to 100 successful missions with 1 failure

List the total number of successful and failure mission outcomes

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

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

Mission_Outcome	TOTAL_NUMBER
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

```
* 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

For this query we had to perform a subquery under our WHERE function to filter out boosters that didn't carry the MAX payload mass.

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql
SELECT LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, substr(Date, 4, 2)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
    AND substr(Date,7,4)='2015';
```

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

Landing_Outcome	Booster_Version	Launch_Site	substr(Date, 4, 2)
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	04

This query posed a special challenge since SQLLite doesn't support month names. Instead we had to use the substr() operative to return the results for month

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

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

```
%%sql
SELECT "LANDING_OUTCOME", COUNT("LANDING_OUTCOME") AS TOTAL
FROM SPACEXTBL
WHERE DATE >= '2010-06-04' and DATE <= '2017-03-20'
GROUP BY "LANDING_OUTCOME"
ORDER BY COUNT("LANDING_OUTCOME") DESC ;
```

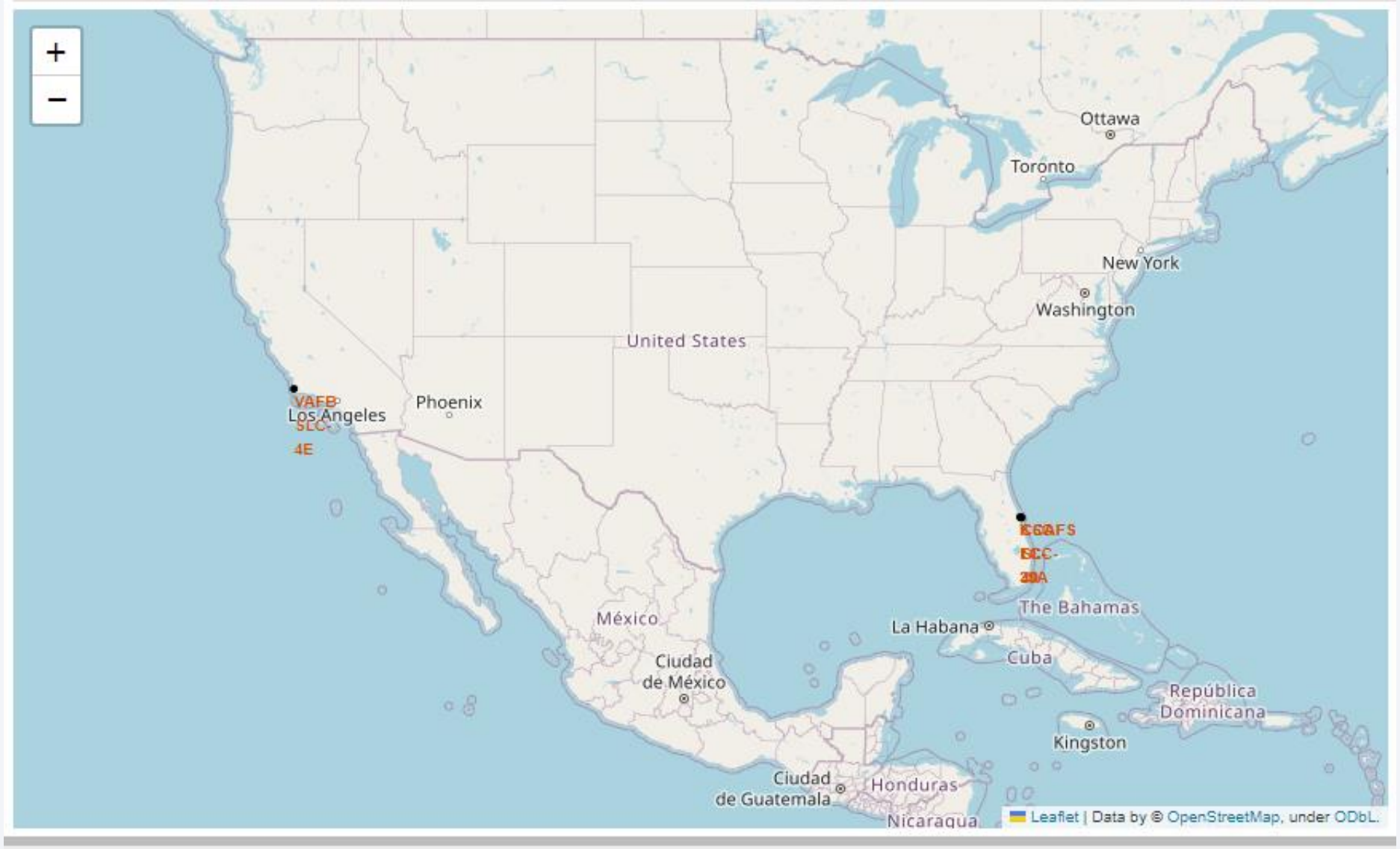
```
 * sqlite:///my_data1.db
Done.
```

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

In our final task, we used GROUP BY, ORDER BY and DESC to clean up and organize our results to make them more easily understood


Section 3
Launch Sites
Proximities Analysis

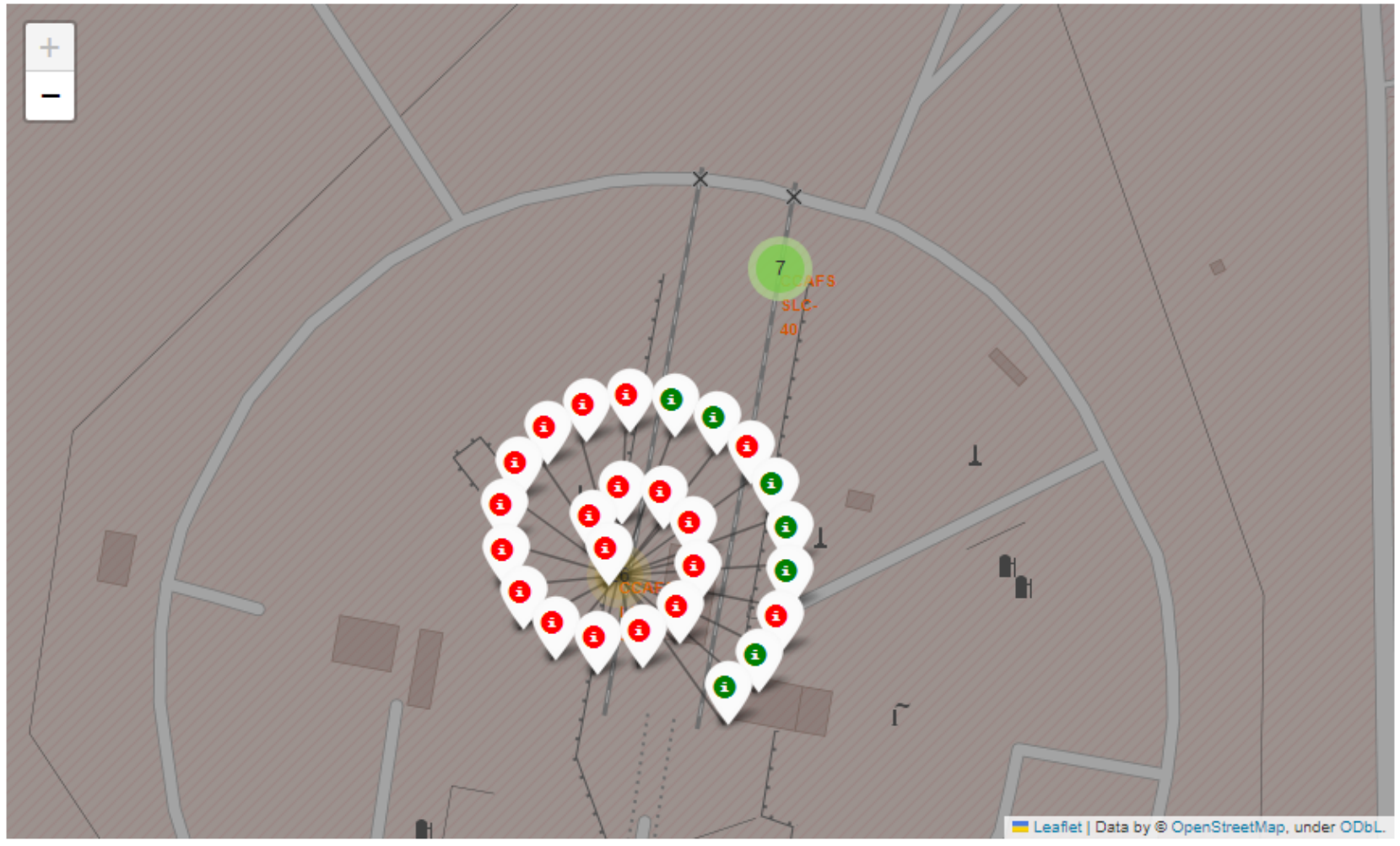
Folium Map Launch Sites



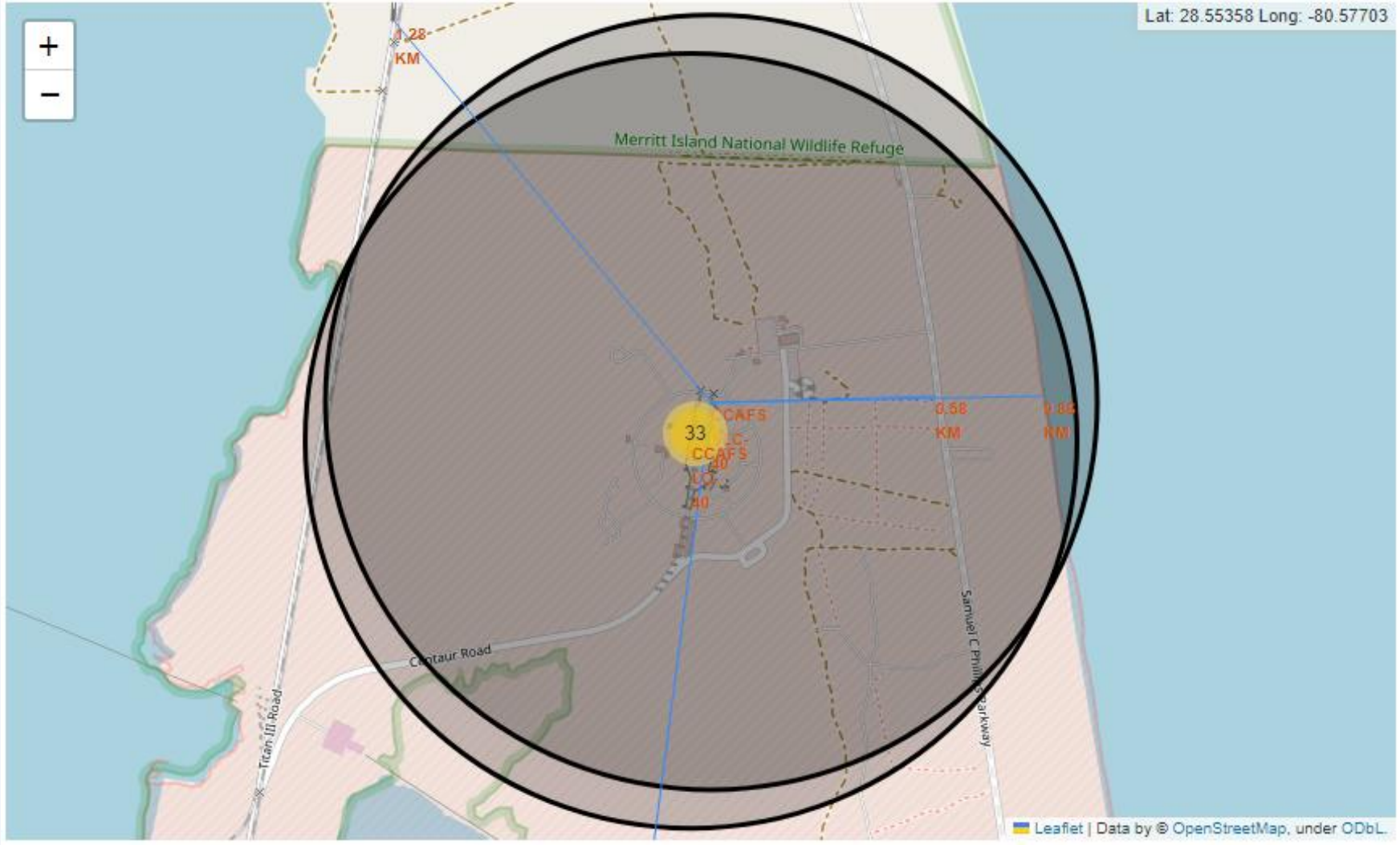
In this first map we generated, we created markers to label each SpaceX launch site. As you can see, they are located on the East & West coasts of the U.S.

Folium Map Outcome Markers

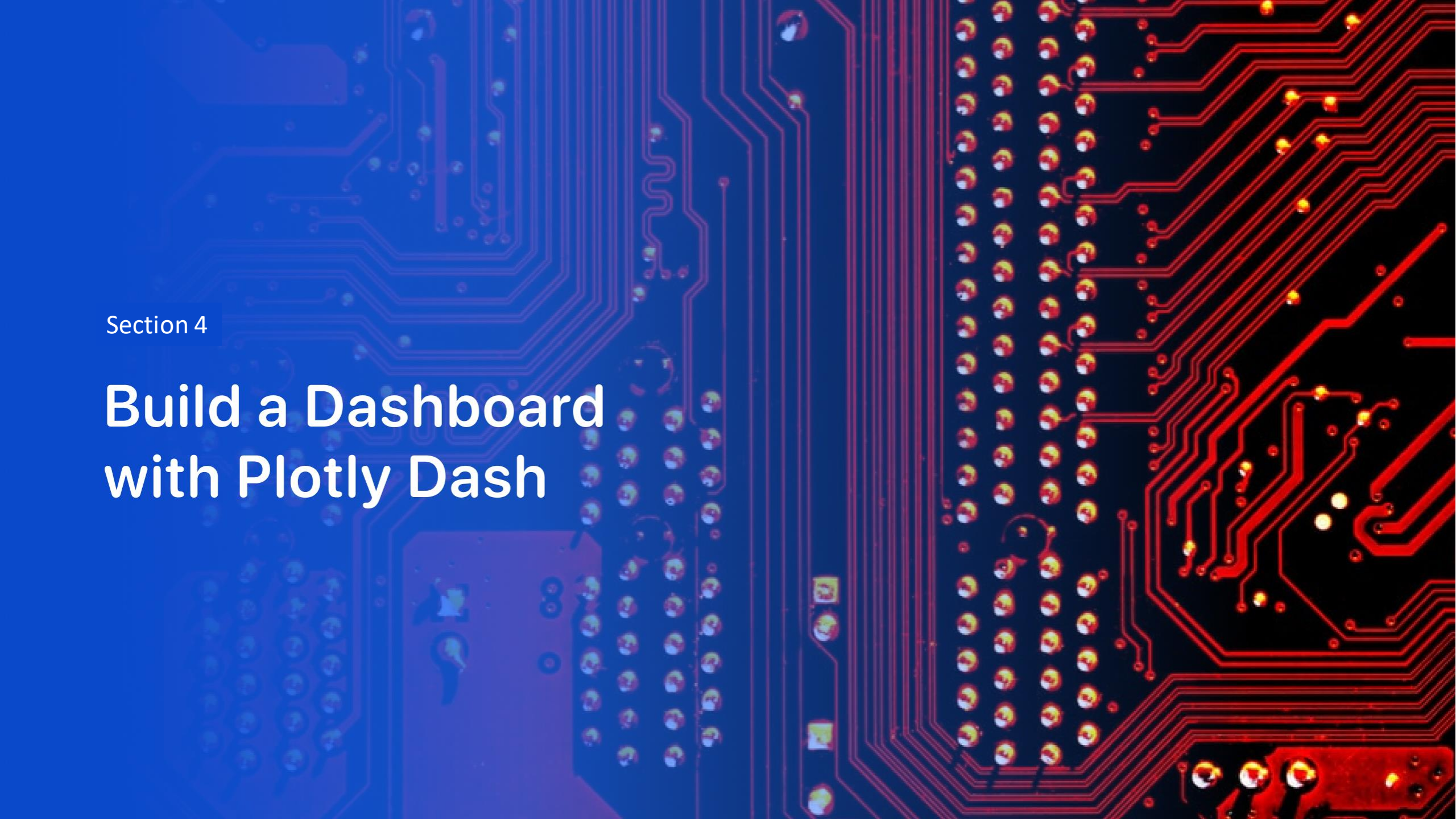
Next we added marker clusters to the map so we could display the landing outcome for each launch at each launch site. In this screenshot we have two launch sites in close proximity. As you can see, it shows the number of launches for the location before you click on it, and the marker indicating outcome for each launch after opening. Green represents a successful mission while the Red markers indicate failure.



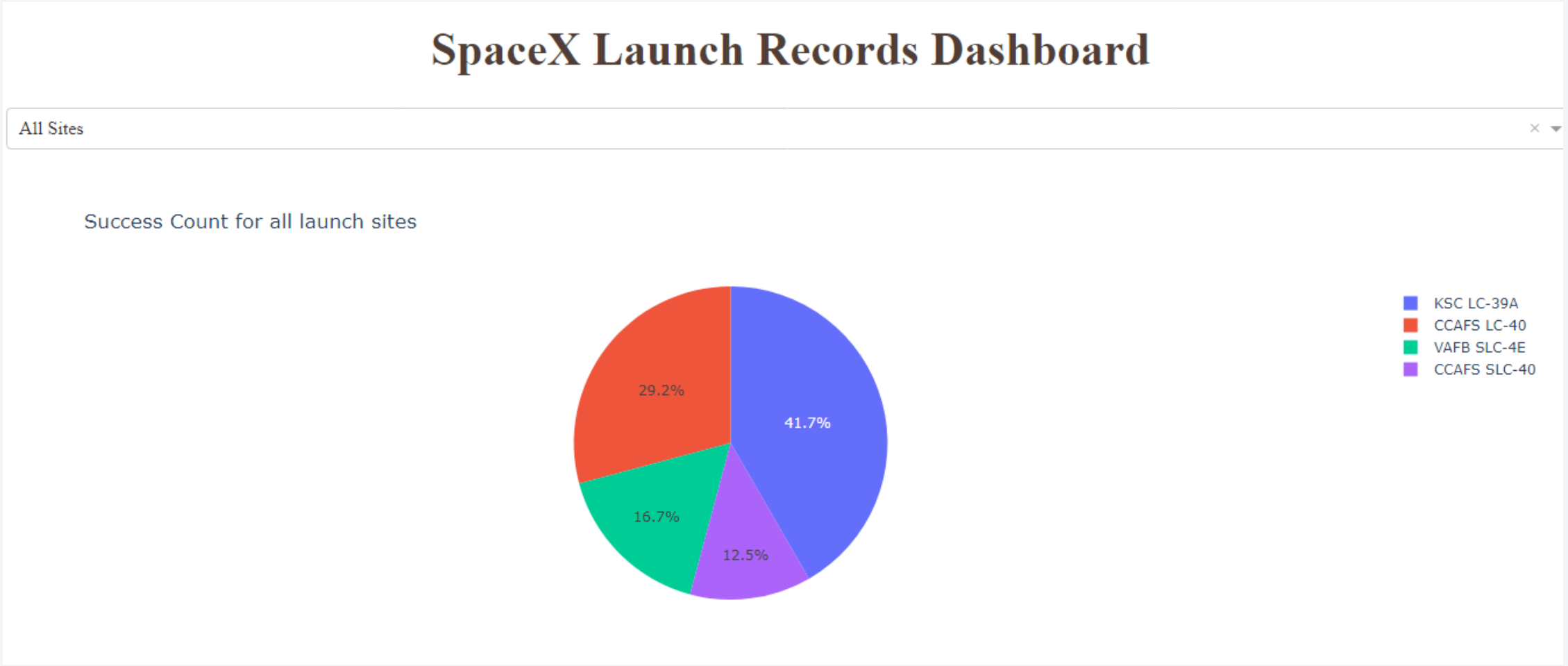
Folium Map Proximities



Our final task using Folium was to calculate the distance between a launch site and its proximities. We picked a launch site and then added lines to show the distance to specific proximity points. In the screenshot you can see blue lines headed from the launch site to the nearest highway, coastline and railway. You can also the calculated distance to these proximity points. We can conclude that launch sites are very close to these three things but not cities.

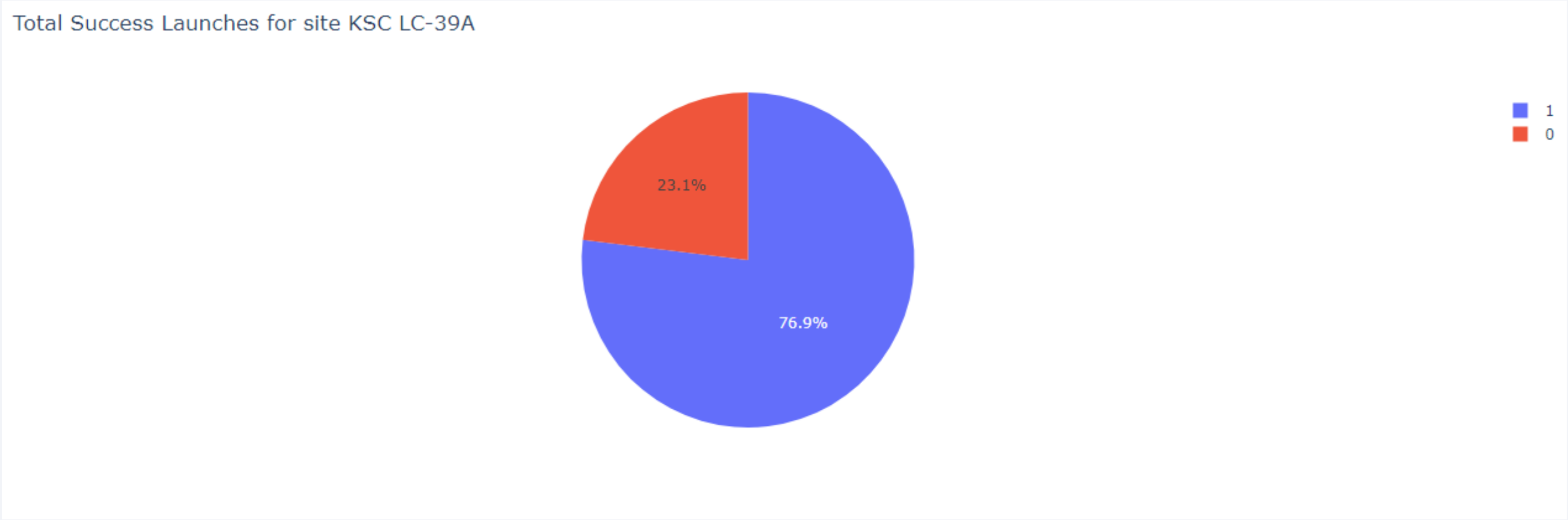

Section 4
Build a Dashboard with Plotly Dash

Interactive Dashboard – Launch Success Pie Chart



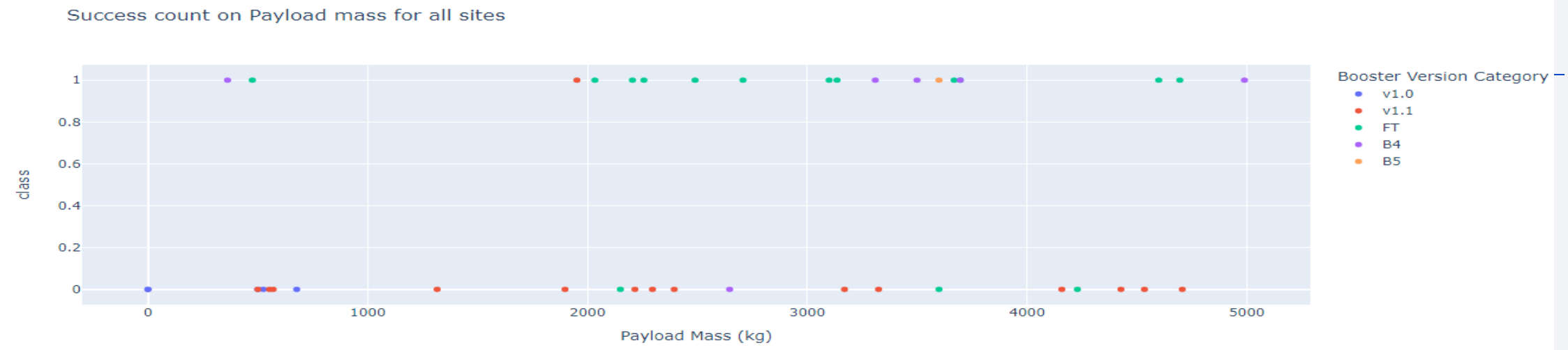
This pie chart shows what percentage each launch site contributes to successful outcomes. For example, KSC LC-39A has the biggest portion of success which is most likely due to its volume of launches compared to other sites

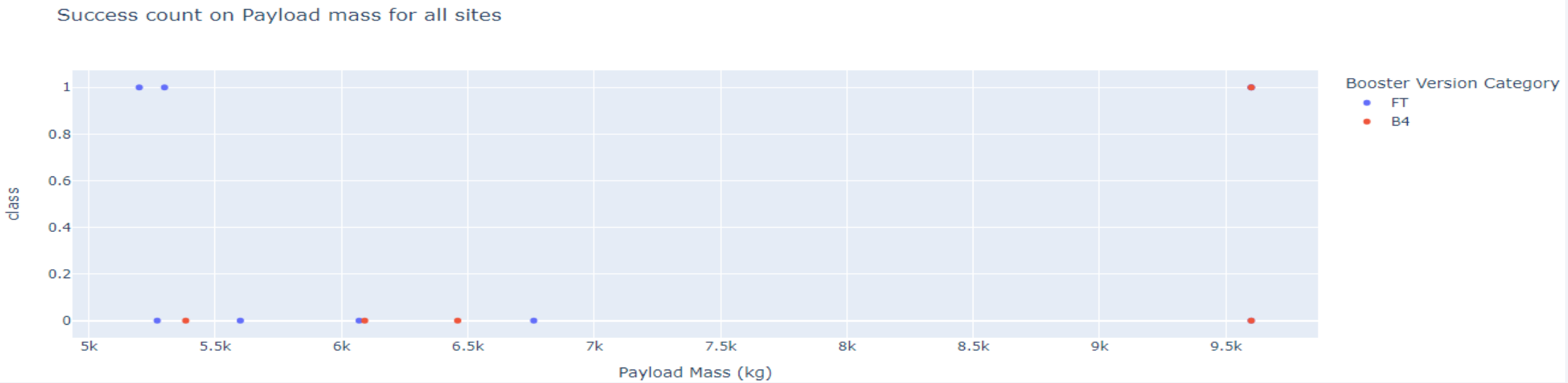
Interactive Dashboard – Highest Success Ratio



When we filter the graph to show only this launch site, we can see the actual percentage of successful vs failed launches. KSC LC-39A has a near 77% success rate, the highest of all launch sites.

Interactive Dashboard – Scatter Plot





In the top graph, the payload range is 0-5k kg and it appears to be an almost equal mix in landing outcomes. In the lower graph, the payload range is 5k-10k kg and you can see a stark increase in failures for heavier loads but only 2 booster versions attempted loads above 5k. The best booster in both graphs is FT and the worst is actually v1.0 having no successful attempts.

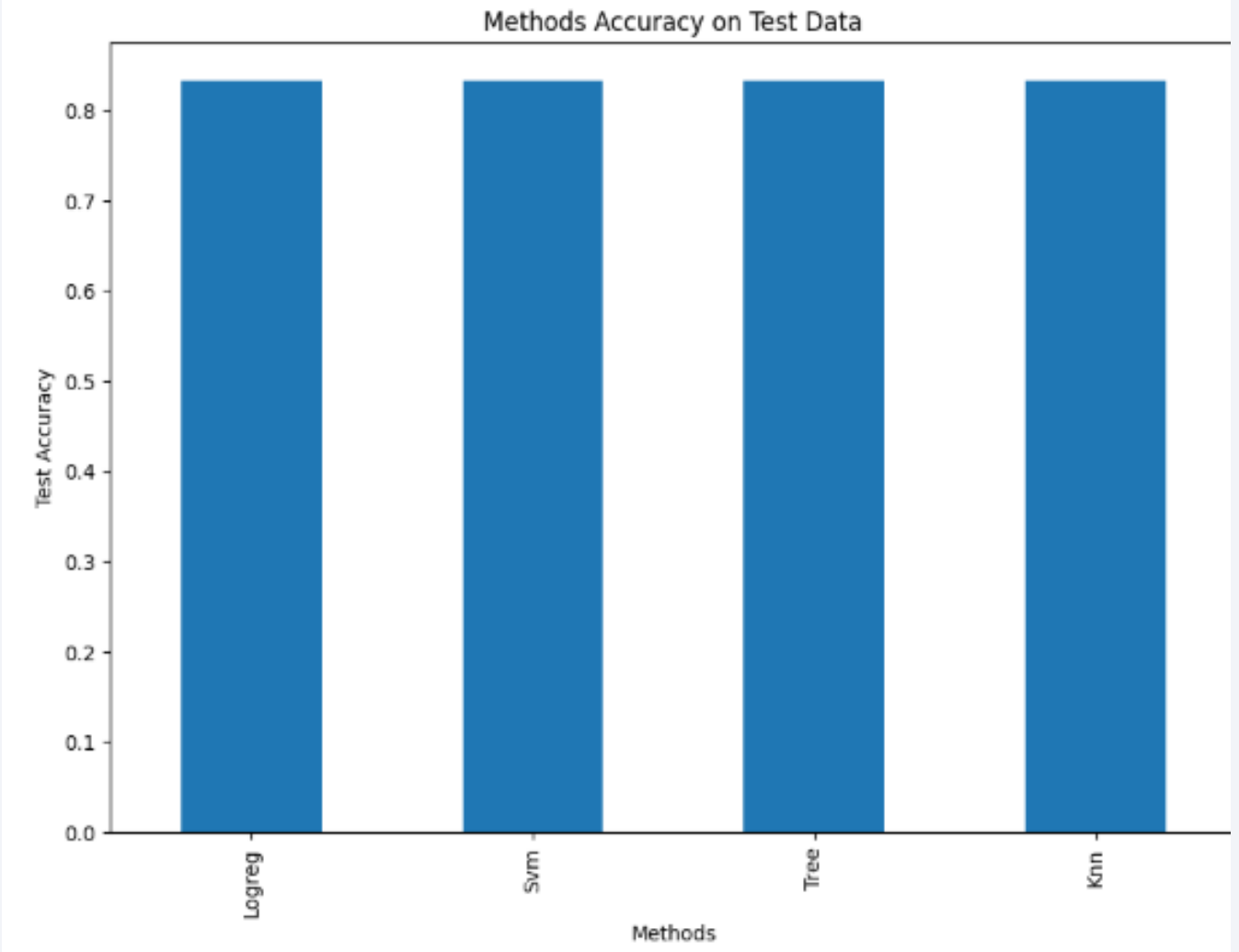
Section 5

Predictive Analysis (Classification)

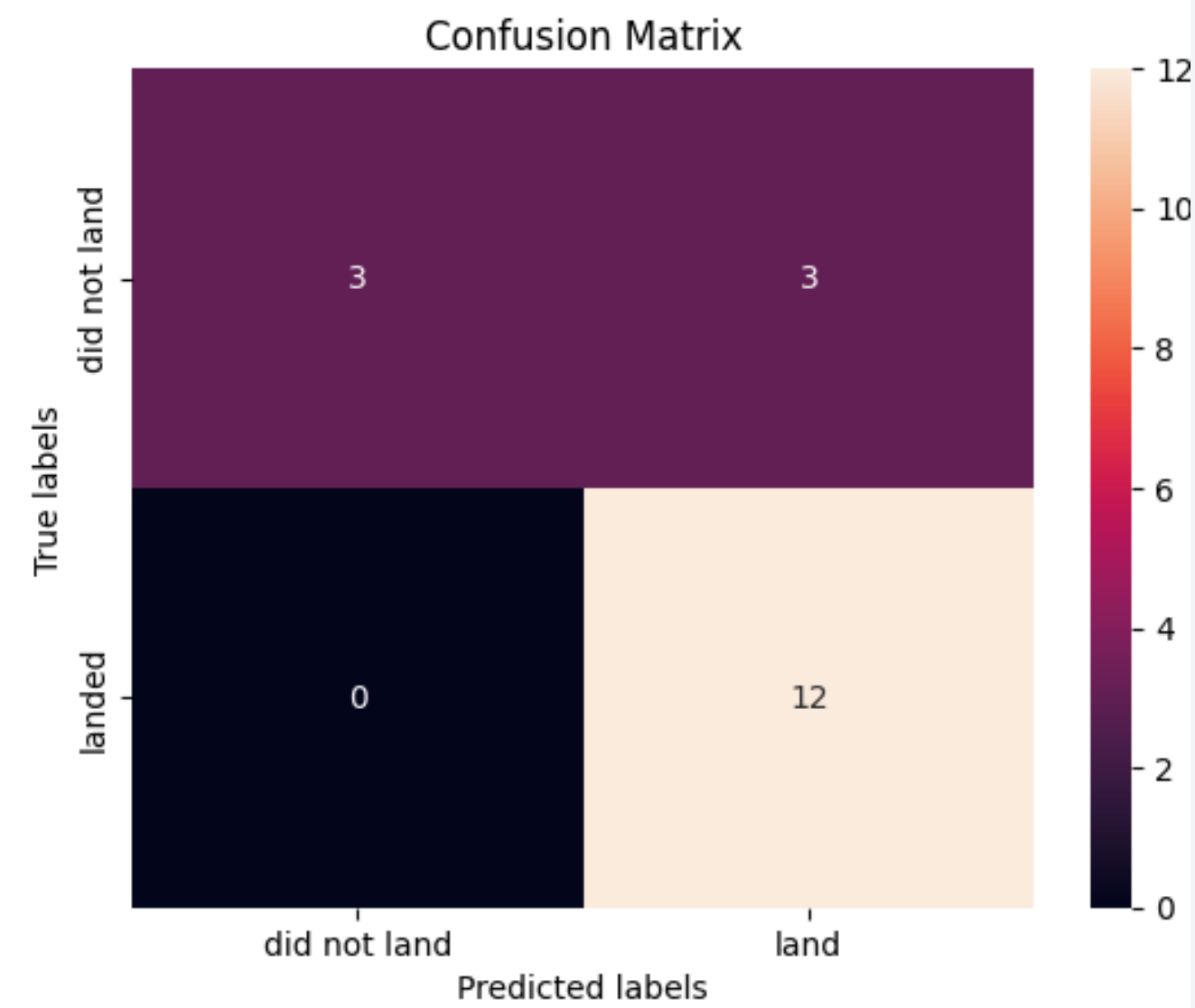
Classification Accuracy

When it came to classification, every method performed the same on the test data. The training data also gave similar results but the decision tree had a slight bump in accuracy as seen below.

	Accuracy Train	Accuracy Test
Logreg	0.846429	0.833333
Svm	0.848214	0.833333
Tree	0.862500	0.833333
Knn	0.848214	0.833333



Confusion Matrix



Since each method came back with the same accuracy, every method produced the same exact Confusion Matrix as shown to the left. The biggest problem with these is false positives.

Conclusions

- Based on our EDA, we can conclude that few variables actually impact mission success. It is my belief that payload mass and booster type play the most significant roles in determining mission success while orbit type or launch site have a lesser impact. Orbit type, for example, is less reliable because some orbit types have only been attempted once so real data is lacking.
- We focused on booster v1.1 heavily but the most successful booster, across the payload scale, was FT. Booster v1.1 failed more often and was only used for lighter payloads, making FT the more widely reliable booster.
- Launch sites aren't a reliable factor because they aren't used equally. The data shows us that the original site was the only active site until launches became more successful. Also some launch sites only went to specific orbits or had more heavy payload flights which were more likely to fail.
- The success rate of missions can be predicted to an accuracy of 83.3%. We used several different methods to test this and while they all performed the same on the test data, the decision tree classifier showed slightly more promise with higher accuracy on the training data.

Thank you!