IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Le Hong Nhat Minh
18/10/2021

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

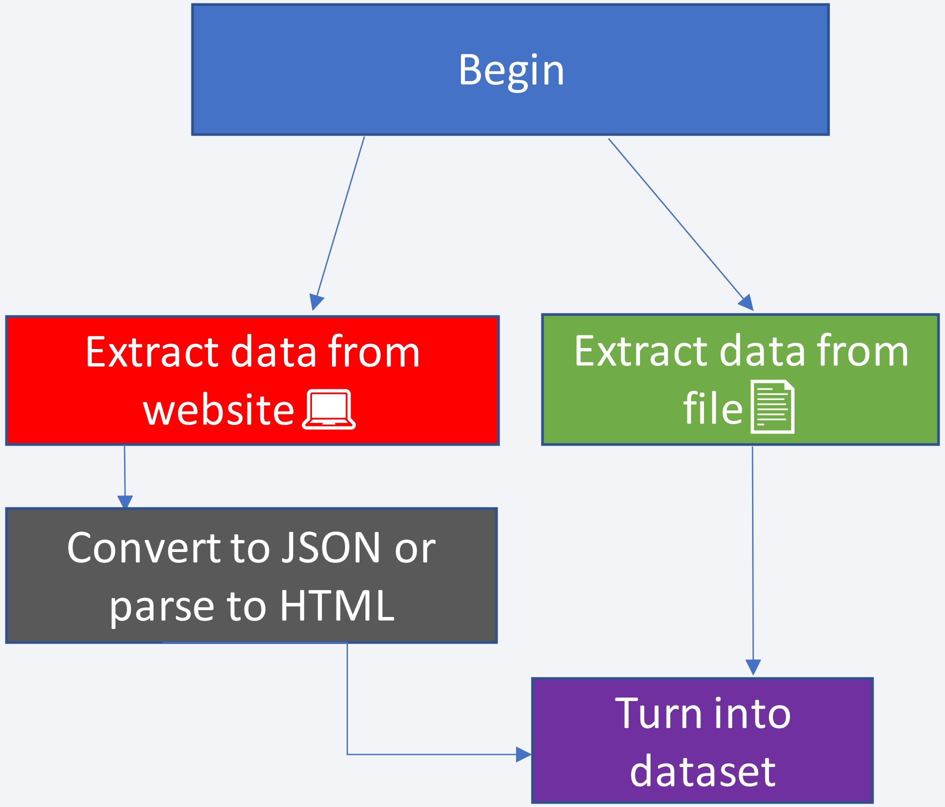
Data Collection

The data sets were collected using a process named **web scraping**, I.e, collected data from a website, using the python libraries **Requests** and **Beautiful Soup**, or by downloading a .csv file using **Pandas**.

To collect the data, we first need to extract the data from either a website or .csv file.

- When extracting from a website, we need to convert it to either decode the data to JSON or parse it to HTML, depending on the library used.
- When extracting from a .CSV file, we don't really need to do anything, Pandas will automatically sort it out for us.

After extracting, we'll need to stuff the extracted data into a **Pandas data set.**



Begin

Extract data from website

Extract data from file

Convert to JSON or parse to HTML

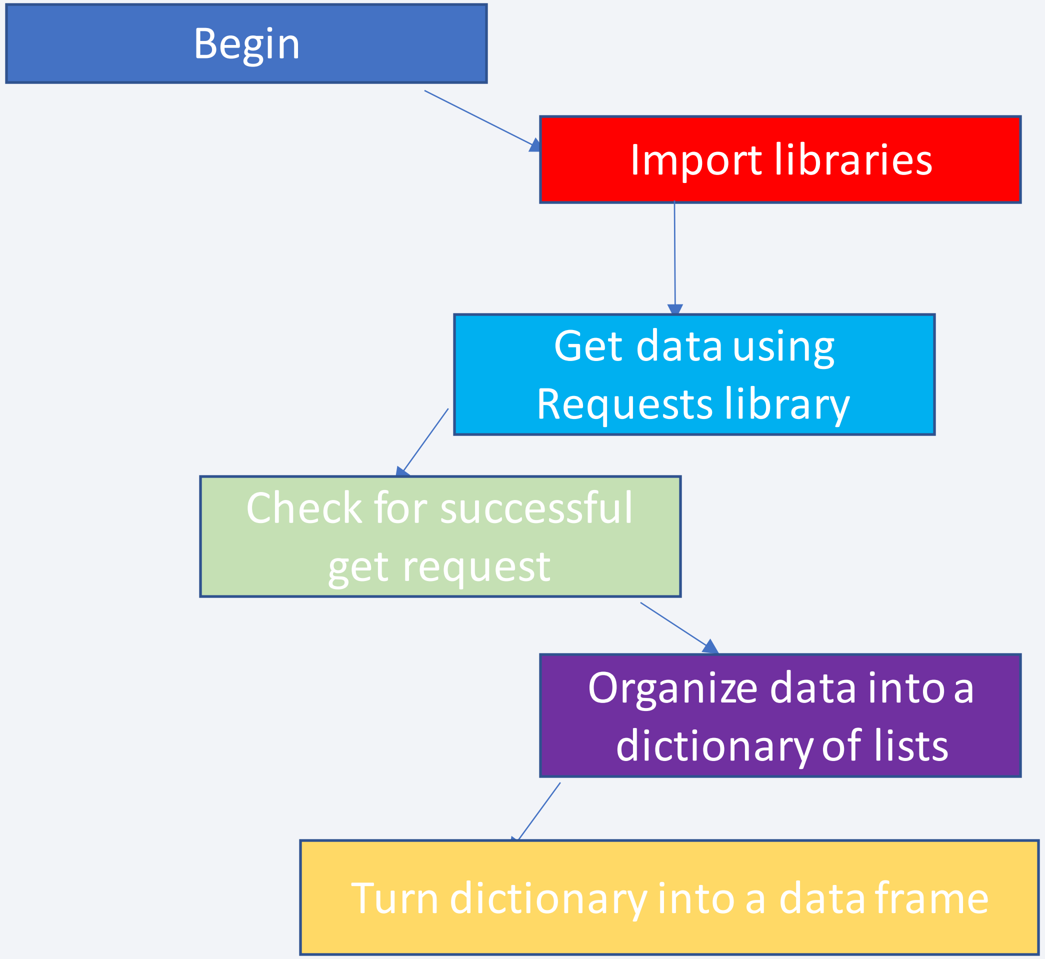
Turn into dataset

Data Collection – SpaceX API

To get a get request to the SpaceX API using Python,

- First we need to get the required libraries.
- After that, we need to get launch data using the **Requests** library's .get() function on the URL: https://api.spacexdata.com/v4/launches/past.
- We also check if the get request was successful by getting the response code, if it's 200, then it was successful.
- We then organize the data into several lists, and put them in a dictionary.
- Then turn the dictionary into a data frame of our own.

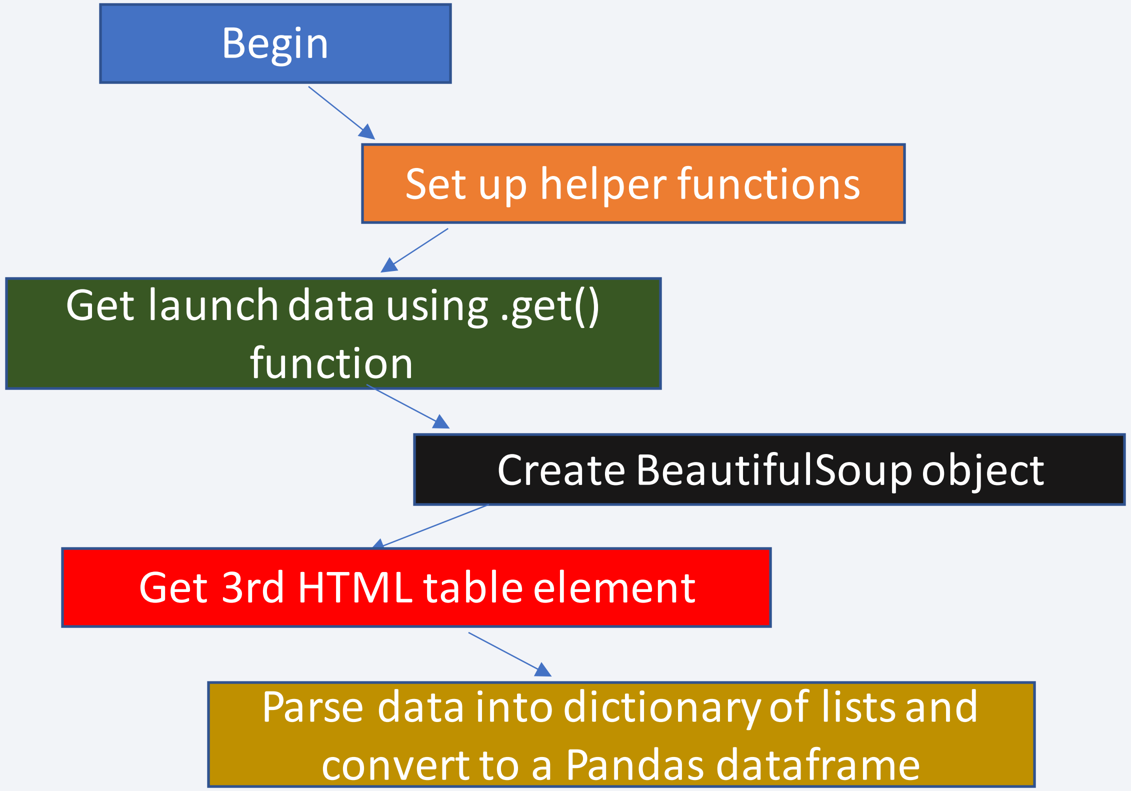
Link to Github: Capstone/jupyter-labs-spacex-data-collection-api.ipynb at main · Alexledev/Capstone (github.com)


Begin
Import libraries
Get data using Requests library
Check for successful get request
Organize data into a dictionary of lists
Turn dictionary into a data frame

Data Collection - Scraping

Collecting data by web-scraping isn't very hard, here are the steps to collect data by web-scraping:

- Just like with using the SpaceX API, we first import all the required Python libraries.
- We also set up some helper functions to process the data later.
- After that, we need to get launch data using the **Requests** library's .get() function on the URL: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- We also create a **BeautifulSoup** object from the HTML response.
- Then we get the 3rd HTML table element since that contains what we need.
- We then parse the data from the table to multiple lists in a dictionary and convert that dictionary to a Pandas data frame.


Begin

Set up helper functions

Get launch data using .get() function

Create BeautifulSoup object

Get 3rd HTML table element

Parse data into dictionary of lists and convert to a Pandas dataframe

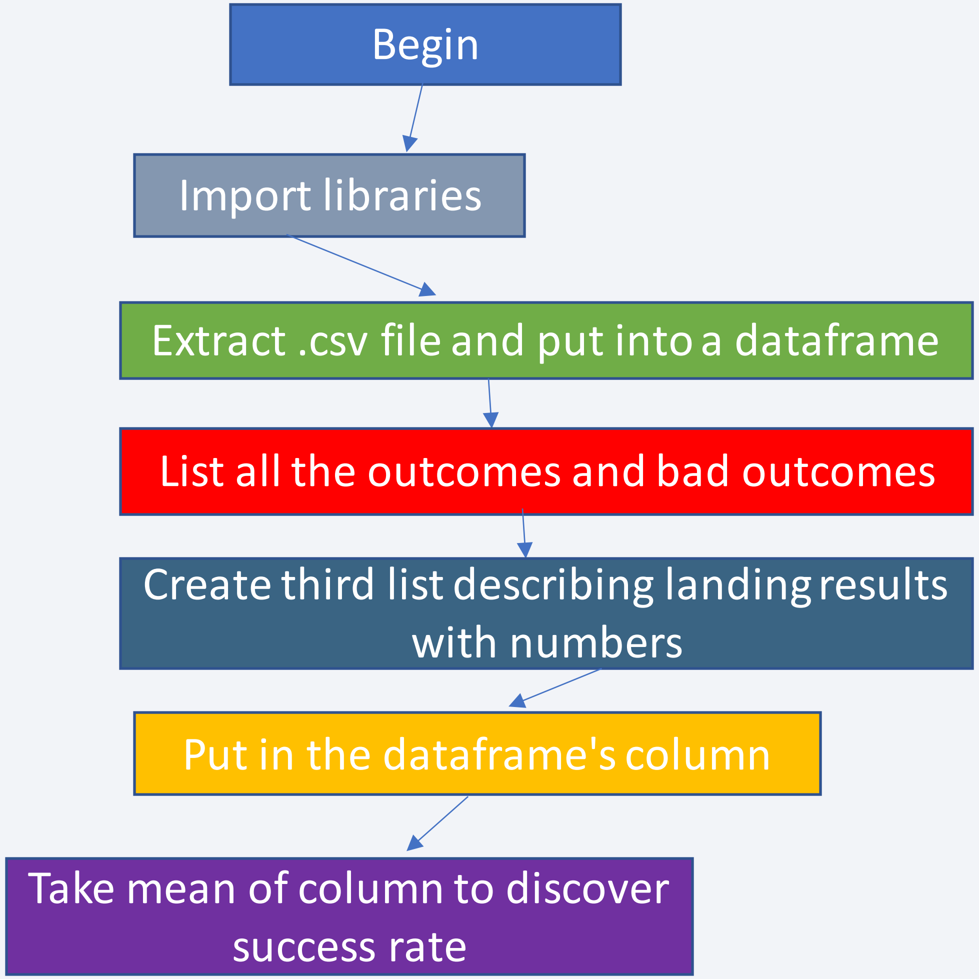
Github URL: Capstone/jupyter-labs-webscraping.ipynb at main · Alexledev/Capstone (github.com)

Data Wrangling

Here are the steps when Data Wrangling

- Before doing some data wrangling, we first need to import the python libraries **Pandas** and **NumPy.**
- We also extract the .csv file containing the required data and turned that into a dataframe.
- We then listed all the outcomes of each landing and put them in a list, we also listed the bad outcomes.
- Using the two lists, we made a third list containing the landing results described by numbers(1 = successful, 0 = unsuccessful)
- We then put the list in one of the dataframe's columns.
- Then we took the mean of that column and discovered the success rate of each landing.

Link to Github: [Capstone/labs-jupyter-spacex-Data wrangling.ipynb at main · Alexledev/Capstone (github.com)]()



Begin

Import libraries

Extract .csv file and put into a dataframe

List all the outcomes and bad outcomes

Create third list describing landing results with numbers

Put in the dataframe's column

Take mean of column to discover success rate

EDA with Data Visualization

When visualizing EDA data, we produced many charts:

1. The first chart displayed the payload mass and flight number, the pattern was that a larger payload mass would probably has a bigger flight number.
2. The second chart displayed the relationship between the flight number and the launch site. We saw that most of the flights numbers from 0-25 and 45-100 were from the launch site "CCAFS SLC 40".
3. The second chart visualized the relationship between the payload and the launch site. It seemed like the most of the launches had a payload mass of 0-8000 kg. Also, the VAFB-SLC launch site is the only one that hasn't had a flight with a payload mass of over 12000 kg
4. The third chart visualized the relationship between the success rate of each orbit type. We saw that launches to the orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rate(mostly due to the fact that those orbits had the least launches)
5. The fifth chart visualized the relationship between the flight number and orbit type. The pattern was that most of the launches to the orbits SSO, HEO, MEO, VLEO, SO, and GEO had bigger flight numbers
6. The sixth chart visualized the relationship between the payload and orbit type. Looking at the chart, it was clear that most of the launches to most of the orbits had a payload mass of about 8000 kg or lower.
7. The seventh and final chart visualized the launch success yearly trend. We saw that a lot of successful launches were in 2017-10-30, and also 3 pockets between 2018 and 2019. It also looks like we're going to have a lot more successful launches in 2020 onwards

Github link: [Capstone/jupyter-labs-eda-dataviz.ipynb at main · Alexledev/Capstone (github.com)]

EDA with SQL

Using SQL to do EDA.

1. Task 1: We displayed the names of the unique launch sites in the space mission using the SELECT function.

2. Task 2: Then used the LIMIT clause with the SELECT function to display 5 records where the launch site begins with the string 'CCA'.

3. Task 3: We used the SUM function to get the total payload mass carried by boosters launched by NASA.

4. Task 4: Using the AVG function, we displayed the average payload mass carried by booster version F9 v1.1.

5. Task 5: With the MIN function, we listed the date when the first successful landing outcome in ground pad was achieved.

6. Task 6: Then we listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 using the AND operator and the BETWEEN operator.

7. Task 7: We then used the COUNT function to list the total number of successful and failed mission outcomes.

8. Task 8: Then we listed the names of the booster_versions which have carried the maximum payload mass using some of the previously mentioned functions.

9. Task 9: We listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015 also using the previously mentioned functons.

10. Task 10: We then ranked the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order using the GROUP BY function

Github link: Capstone/jupyter-labs-eda-sql-coursera.ipynb at main · Alexledev/Capstone (github.com)

Build an Interactive Map with Folium

Using Folium, we placed markers, circles, lines, etc... on some maps:

- We first used a circle to mark the area of a location of the Johnson Space Center, Kennedy Space Center, and Vandenberg Space Launch Complex.
- We also used markers to mark the successful/failed launches for each site on the map.
- Then we drew a line from Johnson Space Center to the nearest coast.

Github Link: Capstone/lab_jupyter_launch_site_location.ipynb at main · Alexledev/Capstone (github.com)

Build a Dashboard with Plotly Dash

Building a Dashboard with Plotly Dash wasn't very challenging, to say the least:

1. The first item to our Dashboard was a drop down menu that let's us choose data from a particular launch site, or all of them if you wanted to.
2. We then added a pie chart that can do two things depending on the launch site selected from the drop down menu: It can display the percentage of successful rocket launches of all the sites at once, or show the percentage of both successful and unsuccessful launches of one site.
3. Another thing we added is a range slider that allowed us to select the payload weight range. It will come handy when we make the scatter plot.

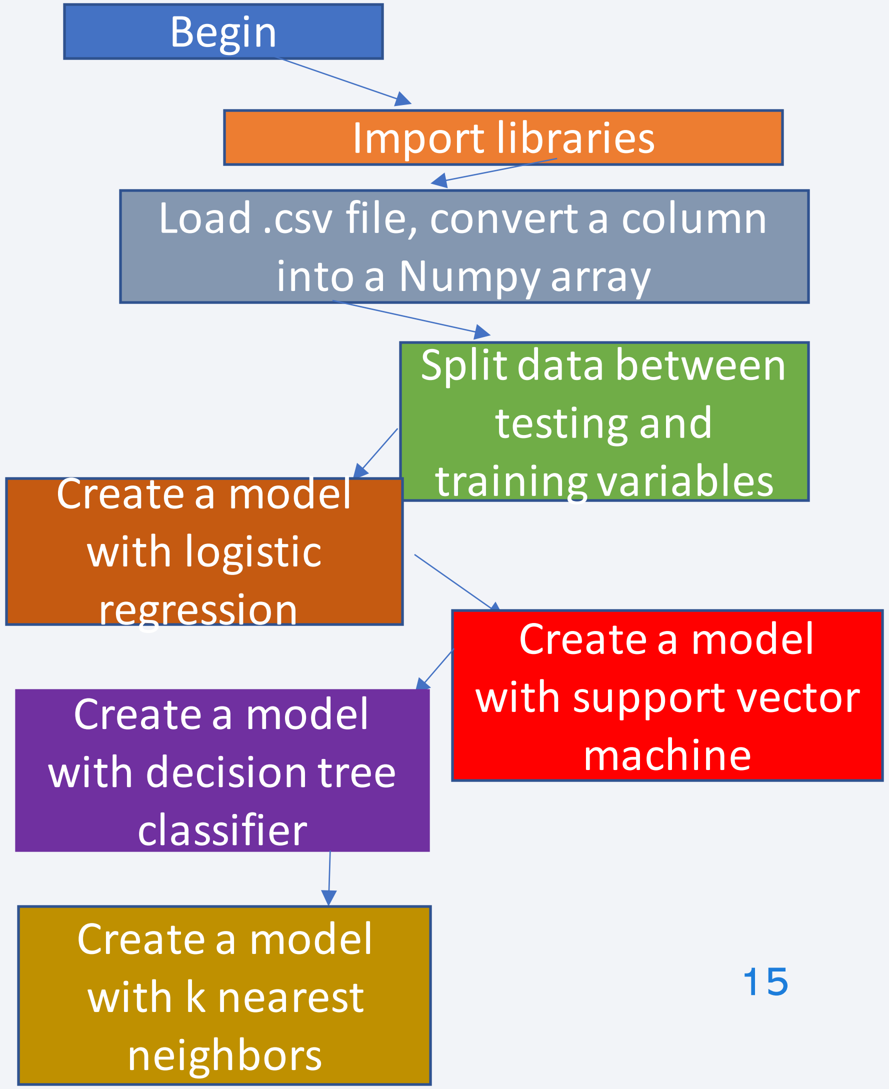
- The final item to be added was a scatter plot which showed the correlation between the payload and success rate for the sites. Each point is also color-coded to show to type of booster used.

Link: https://labs.cognitiveclass.ai/tools/theiadocker/?md_instructions_url=https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_3/lab_theia_plotly_dash.md&lti=true#

Predictive Analysis (Classification)

When doing Predictive Analysis, there are a lot of ways to model the data, some are better than others:

1. Firstly, we imported the required python libraries, as usual.
2. We also loaded the .csv file, and converted one of its columns into a Numpy array.
3. Next, we made some training and testing variables, and split the data between them.
4. We then created a logistic regression object and fit that into a GridSearchCV object. The accuracy of this model to the test data was about 0.833, to the training data it was 0.846
5. We also created a support vector machine object and fit that into a GridSearchCV object. The accuracy of the model to the test data was about 0.833 too, for the training data the score was 0.848
6. A descision tree classifier object was also created and fitted into a GridSearchCV object. The accuracy of the model to the test data was 0.778, for the training data it was 0.886
7. Finally, a "k nearest neighbors" object was made and fitted into a GridSearchCV object. The accuracy of the model to the test data was still 0.833, for the training data the score was 0.848



Begin

Import libraries

Load .csv file, convert a column into a Numpy array

Split data between testing and training variables

Create a model with logistic regression

Create a model with support vector machine

Create a model with decision tree classifier

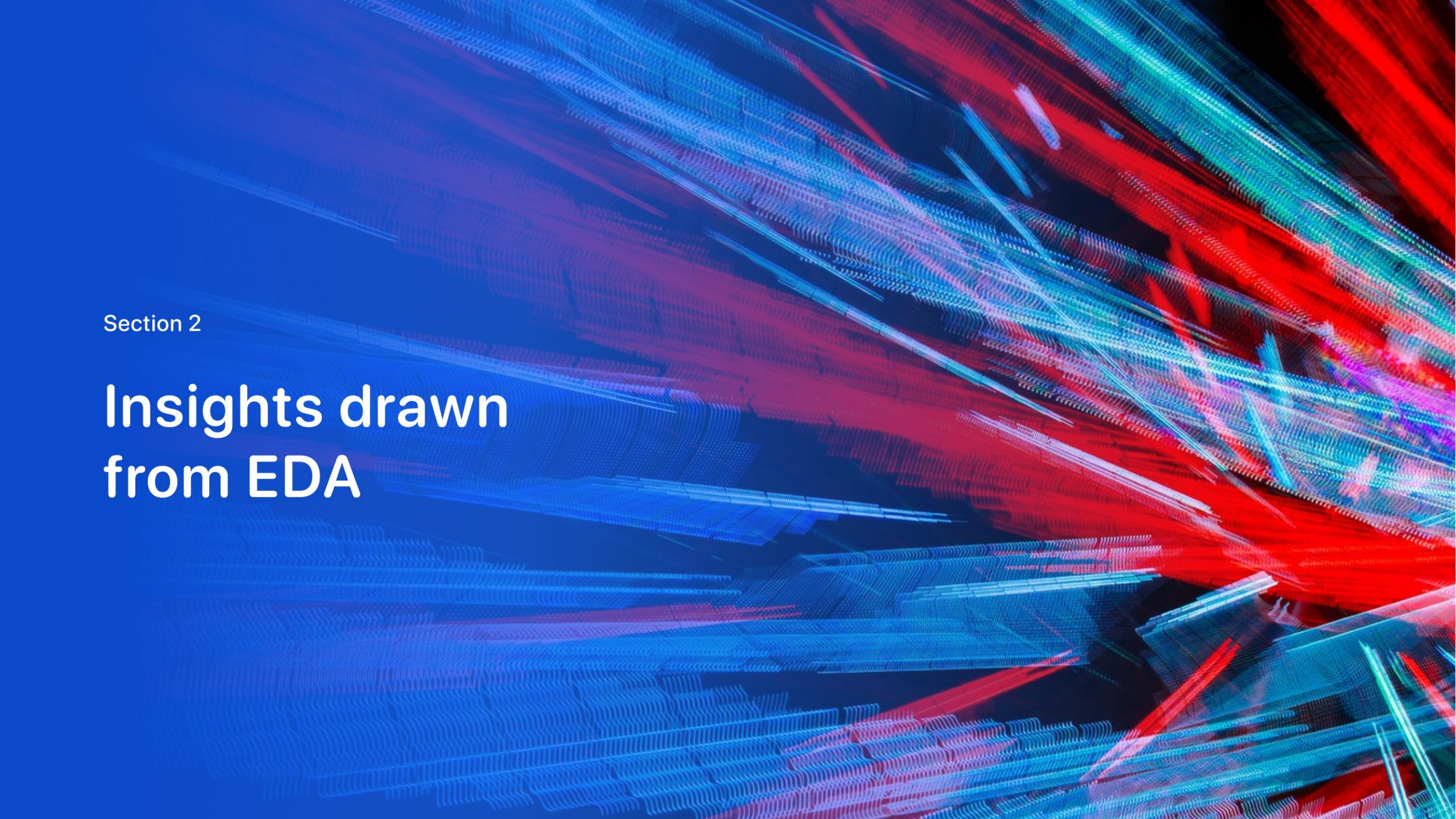
Create a model with k nearest neighbors

Results

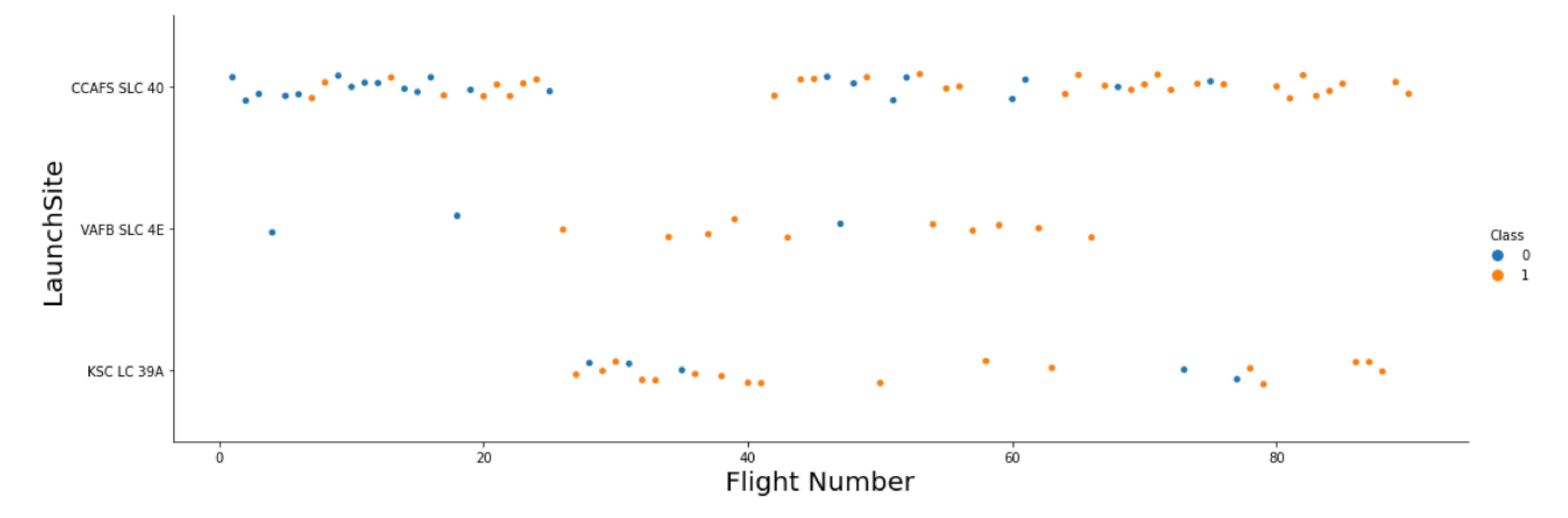
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

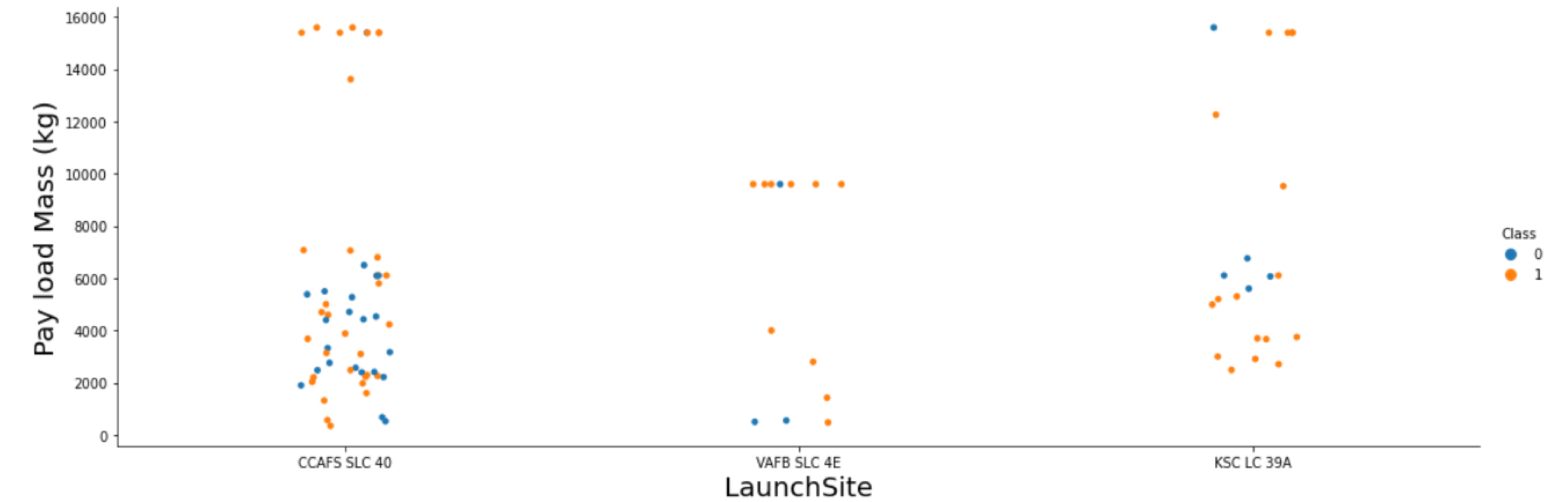


Flight Number vs. Launch Site



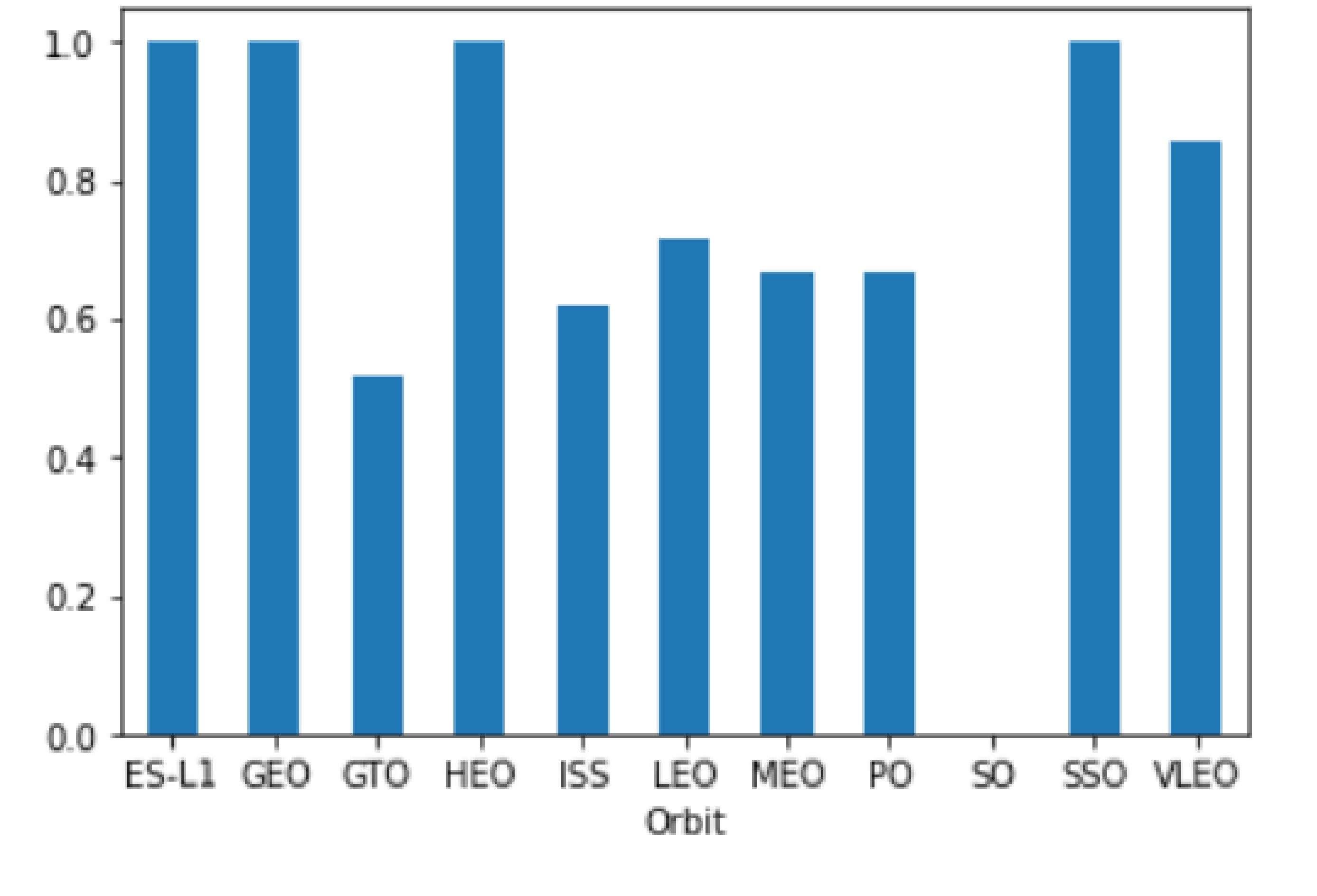
Judging from the chart, It looks like most flight numbers are from the launch site CCAFS SLC 40, the launch site's flight numbers are centered between 0-30 and 40-100. The KSC LC 39A launch site fills the 30-40 gap of the CCAFS SLC 40 launch site. While the VAFB SLC 4E launch site has nothing particularly interesting.

Payload vs. Launch Site



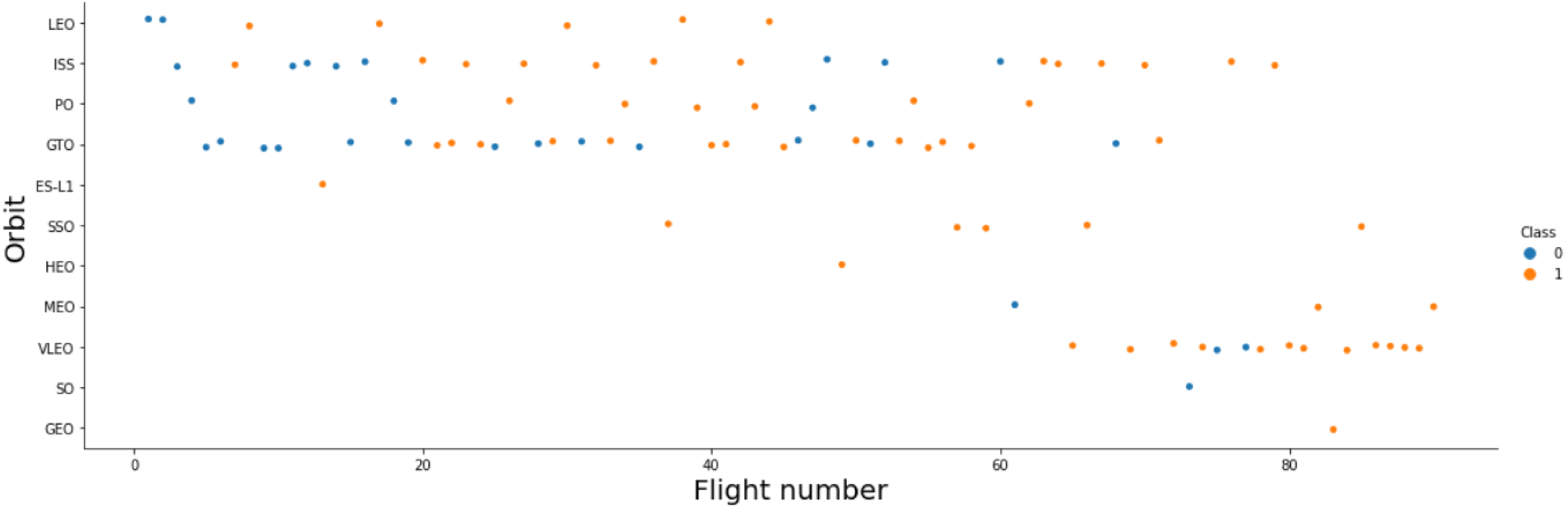
Looking at the chart, we can see that the launch sites CCAFS SLC 40 and KSC LC 39A both had launches with a payload of up to 16000 kilograms. On the other hand, the launch site VAFB SLC 4E only had launches with payloads up to only 10000 kilograms. Also, most launches with a payload mass of over 10000 kilograms haven't failed.

Success Rate vs. Orbit Type



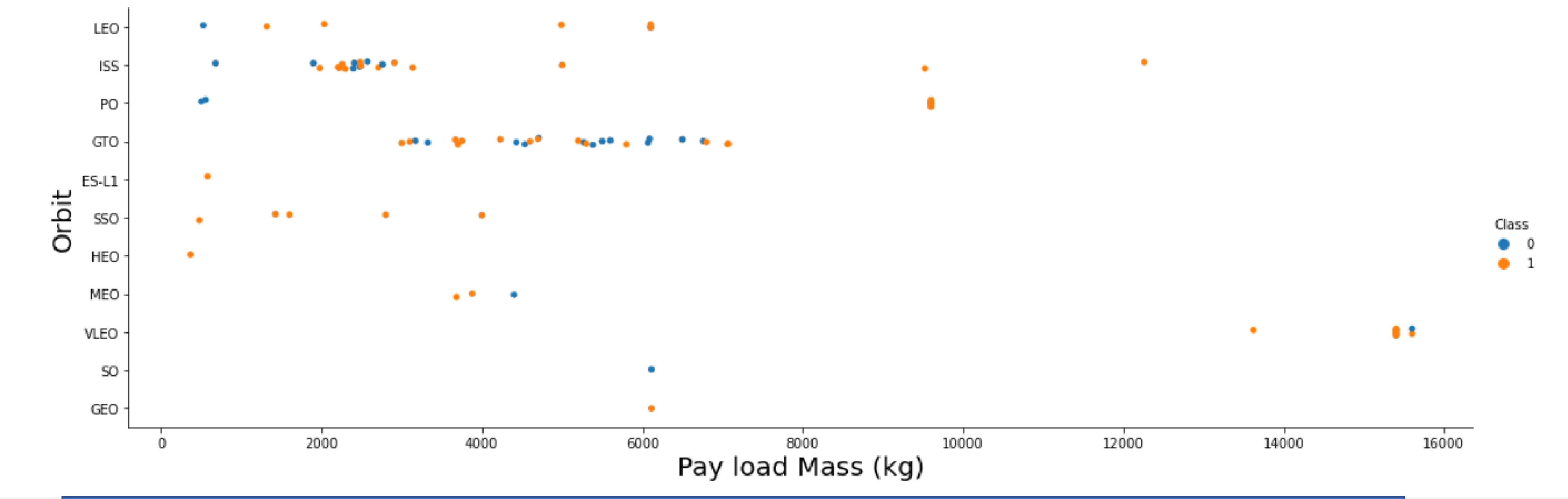
The graph clearly shows that all flights to the orbits ES-L1, GEO, HEO, and SSO so far haven't failed. Also, no flight to the SO orbit has succeeded.

Flight Number vs. Orbit Type



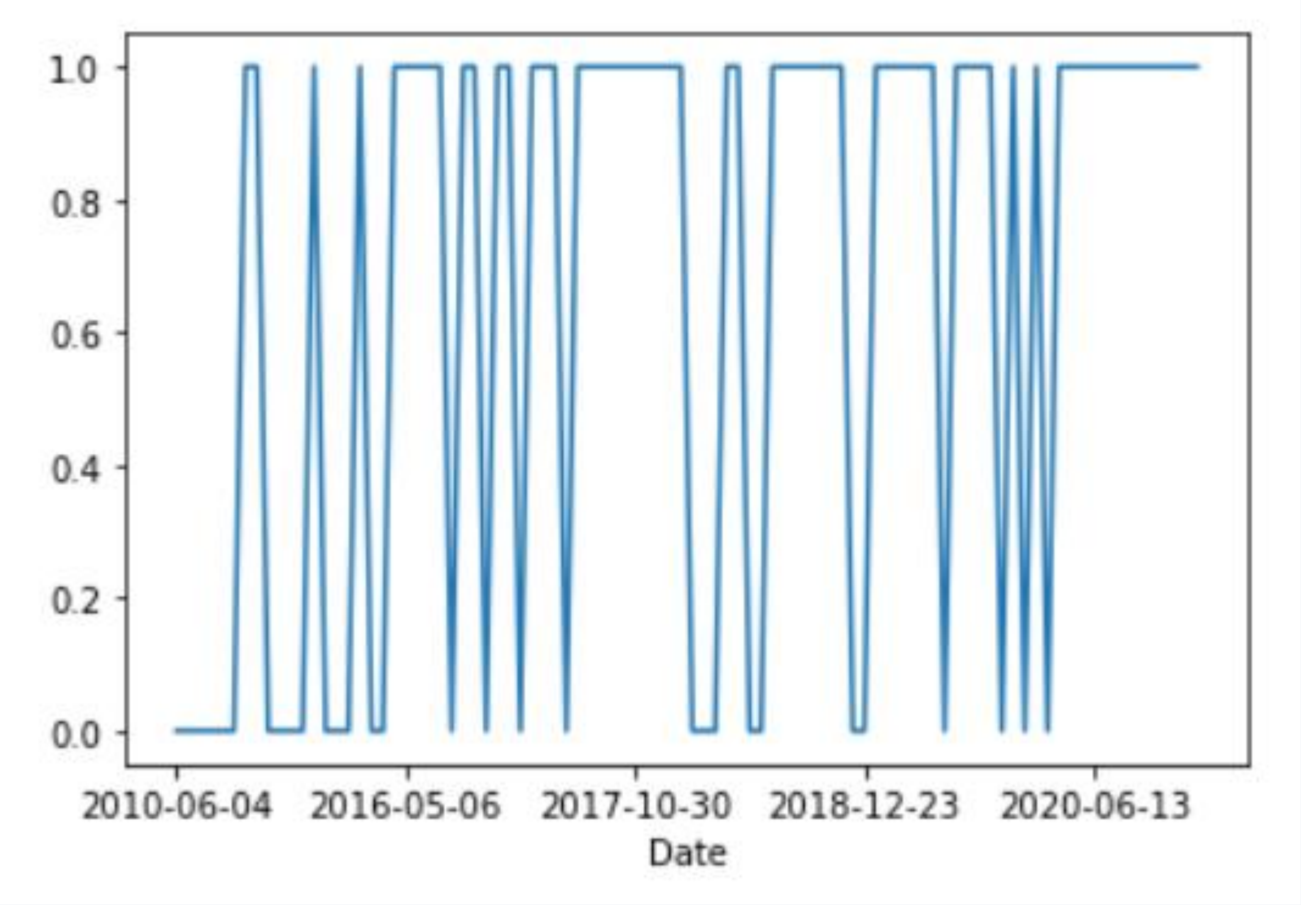
From the graph, we can see that most flights with a flight number from 0 to about 58 are mostly to the orbits LEO, ISS, PO, and GTO. However, the trend quickly disintegrates from flight number 60 onwards.

Payload vs. Orbit Type



Looking at the graph, we can see that a lot of flights with a payload mass of about 3000 kilograms to 7500 kilograms were going to the orbit GTO, flights with a payload mass of about 2000 kilograms to 2500 kilograms were to the ISS. The flights with the most payload mass were to the orbits VLEO, PTO, and ISS

Launch Success Yearly Trend



Judging by the graph, we can see that most successful flights were in 2016, 2017, and some places between the years 2018 and 2019. Looking at the trend from 2020 onwards, it looks like there is going to be a long string of successful launches.

All Launch Site Names

Names of all the unique launch sites:

1. CCAFS LC-40
2. VAFB SLC-4E
3. KSC LC-39A
4. CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Records:

1. '2010-06-04', '18:45:00', 'F9 v1.0 B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'LEO', 'SpaceX', 'Success', 'Failure (parachute)'

2. '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)'

3. '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'

4. '2012-10-08', '00:35:00', 'F9 v1.0 B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt'

5. '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

- Total payload carried by boosters that were from NASA: 48213 kilograms

Average Payload Mass by F9 v1.1

Average payload mass carried by the booster version F9 v1.1: 2534.67 kilograms

First Successful Ground Landing Date

Date of the first successful landing outcome on a ground pad: 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Names of the boosters which have successfully landed on a drone ship and had a payload mass that was greater than 4000 kg but less than 6000 kg:

1. "F9 FT B1022"
2. "F9 FT B1026"
3. "F9 FT B1021.2"
4. "F9 FT B1031.2"

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Total number of successful and failed mission outcomes:

- `Failure : 1`
- `Success : 99`
- `Success (payload status unclear) : 1`

Boosters Carried Maximum Payload

Names of the boosters that have carried the maximum payload mass:

- 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

Failed drone ship landing outcomes from 2015, displaying booster versions, and launch site names :

1. Booster version: F9 v1.1 B1012, launch site: CCAFS LC-40
2. Booster version: F9 v1.1 B1015, launch site: CCAFS LC-40a

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

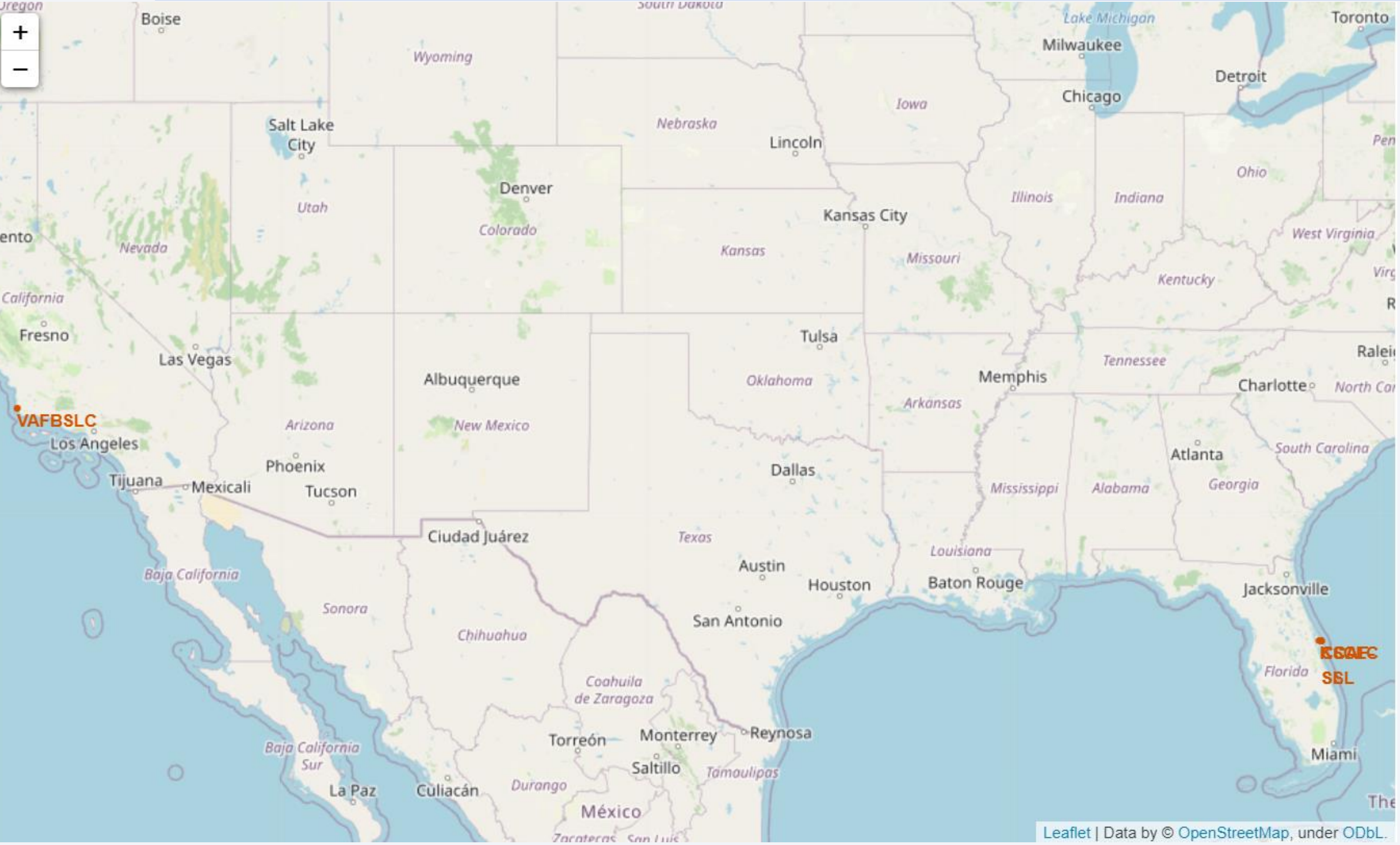
Count of landing outcomes between the date 2010-06-04 and 2017-03-20 ranked in descending order:

```
1. ('No attempt', 10)
2. ('Failure (drone ship)', 5)
3. ('Success (drone ship)', 5)
4. ('Controlled (ocean)', 3)
5. ('Success (ground pad)', 3)
6. ('Uncontrolled (ocean)', 2)
7. ('Failure (parachute)', 1)
8. ('Precluded (drone ship)', 1)
```

Section 4

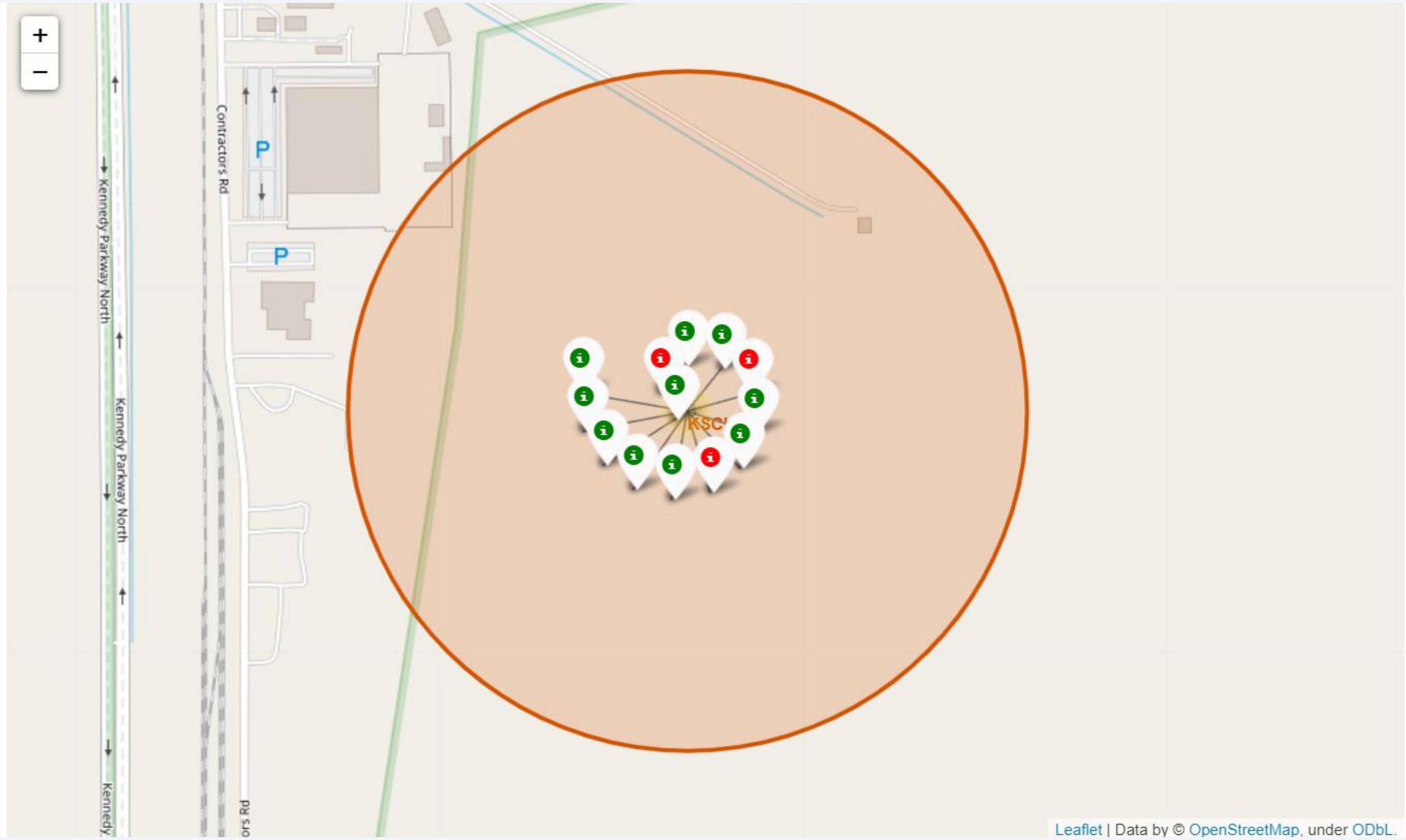
Launch Sites Proximities Analysis

Locations of All launch sites on map



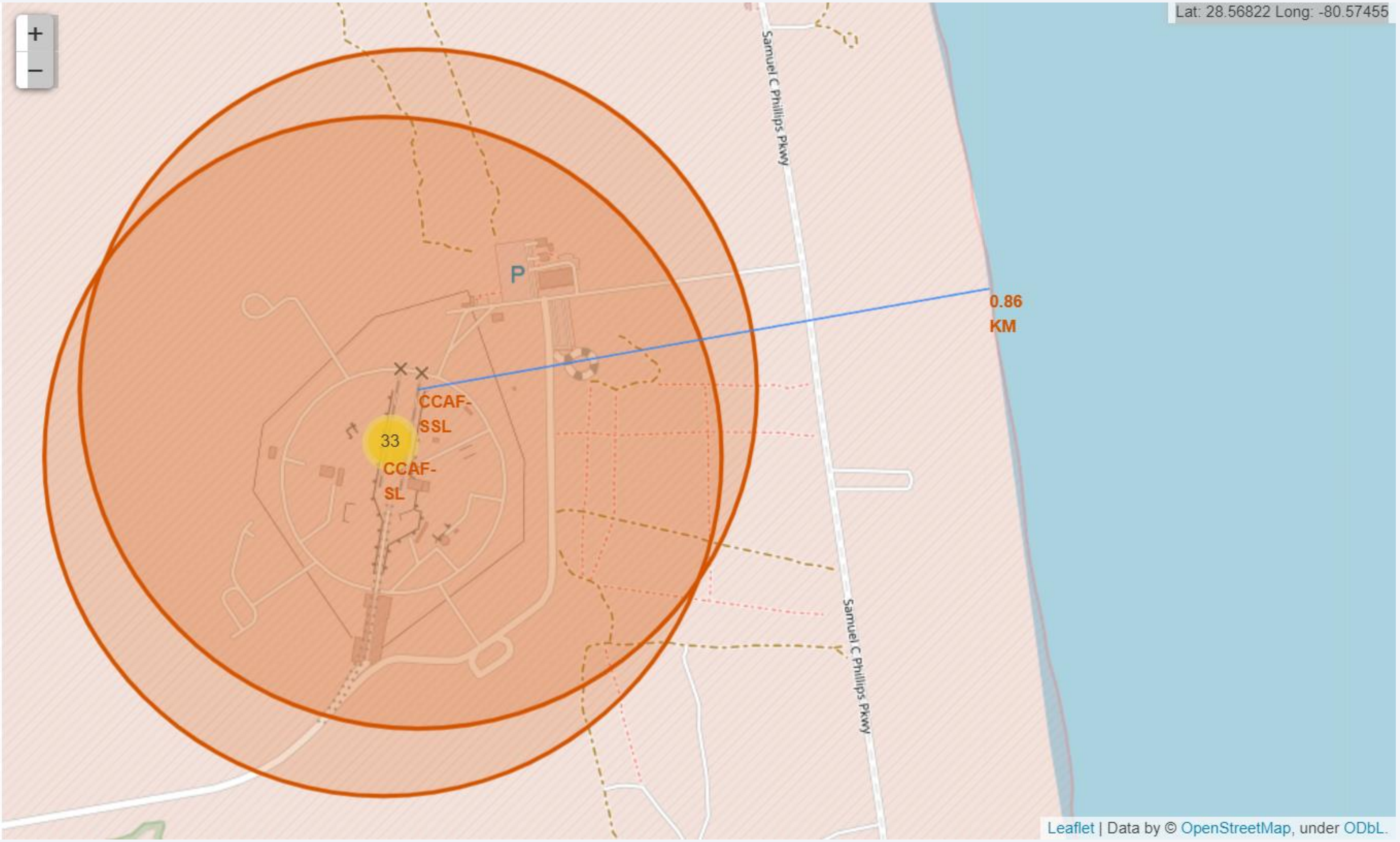
From the map, we can see the locations of all the launch sites that have launched SpaceX rockets. Most of the launch sites are clumped up on the coast of Florida. The VAFB SLC-4E seems to be located on the coast of California.

Successful and Unsuccessful launches from a launch site



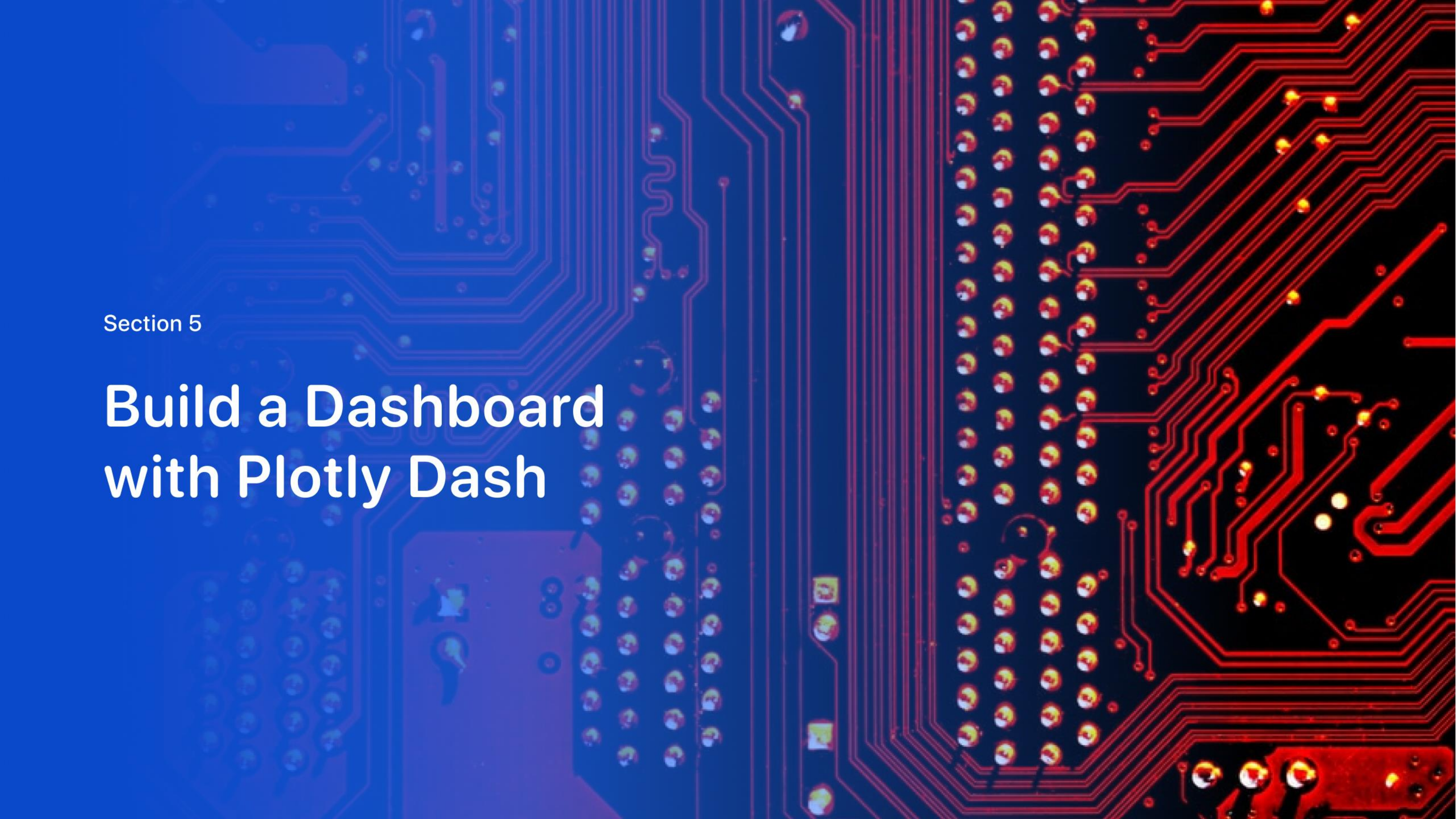
This zoomed-in map of the KSC LC-39A launch site reveals all the successful and unsuccessful launches from that launch site. The green labels show a launch that was successful and red shows an unsuccessful launch

Launch site to coast



+
−
Lat: 28.56822 Long: -80.57455
Samuel C Phillips Pkwy
P
0.86 KM
CCAF-SSL
33
CCAF-SL
Samuel C Phillips Pkwy
Leaflet | Data by © OpenStreetMap, under ODbL.

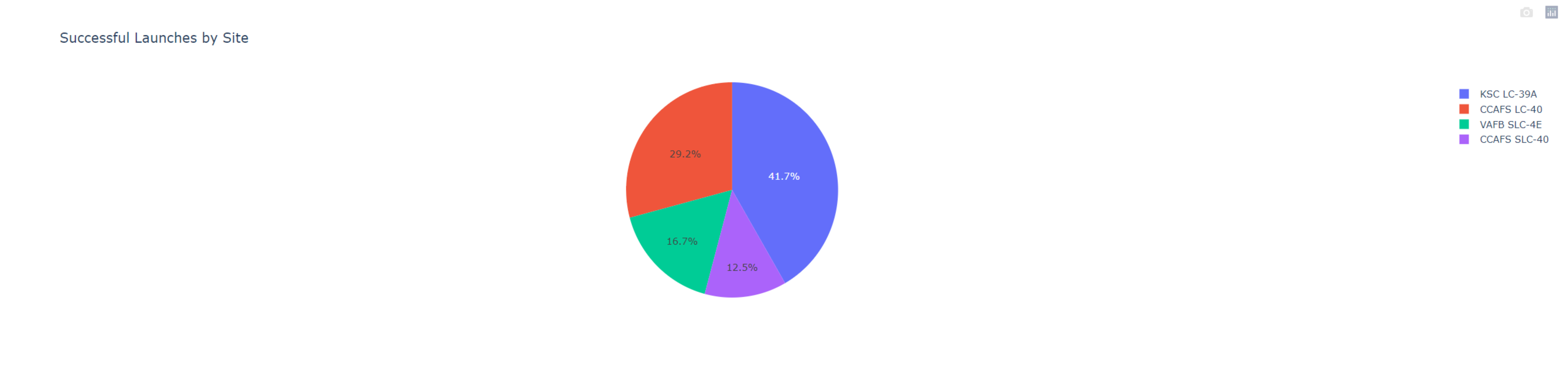
The zoomed-In map shows us a line between the CCAF-SSL 40 launch site and the nearest coast. We can see that the distance between them is about 0.86 kilometers.



Section 5

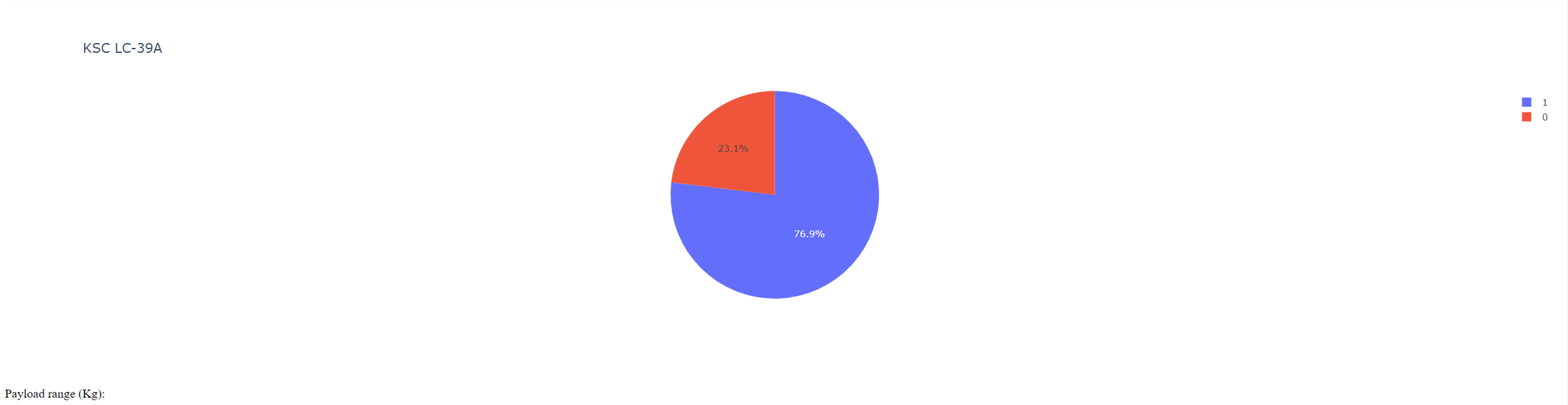
Build a Dashboard with Plotly Dash

Successful launches Pie-chart



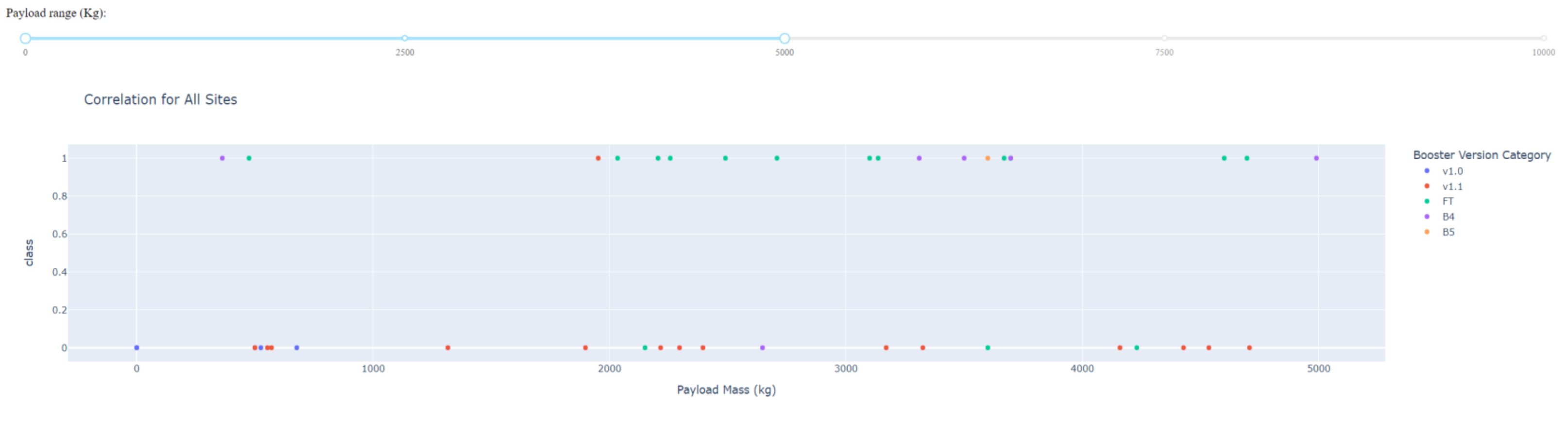
This pie chart displays the successful launches of all sites. Looking at the chart we can see that the launch site KSC LC-39A has the most successful launches with CCAFS LC-40 following behind.

Successful and Unsuccessful launches of a site



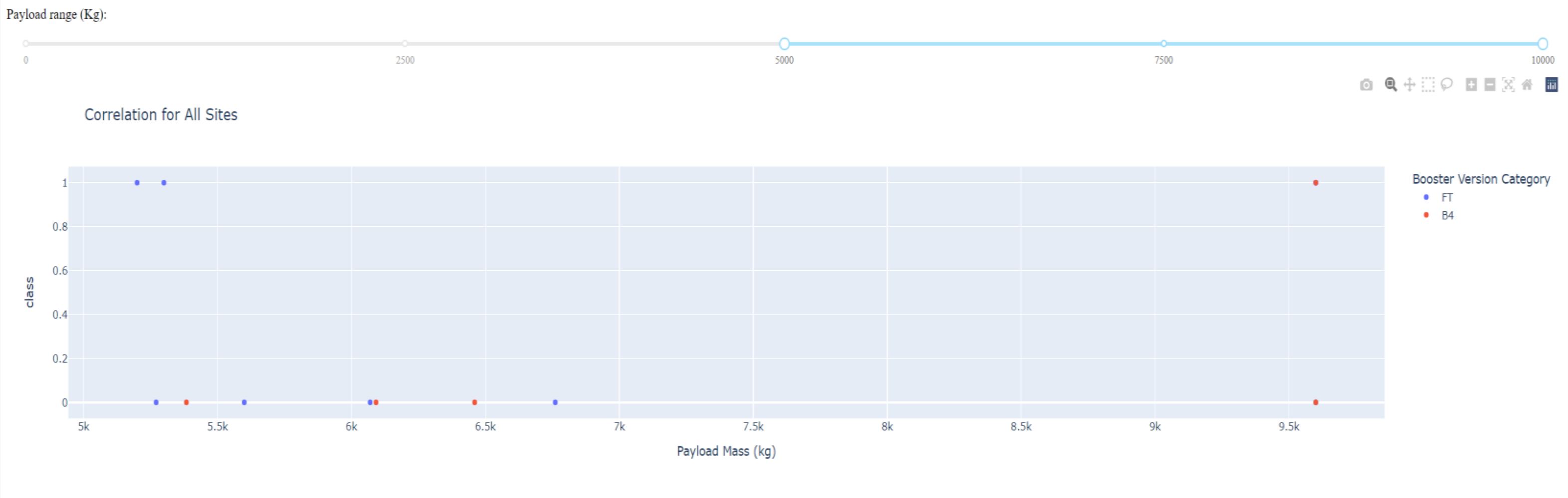
This pie-chart displays the successful and unsuccessful launches of the KSC LC-39A launch site, which had the most successful launches. We can see that a launch has only about 23% chance of failing.

Boosters and Payloads



Here is a scatter plot with a payload range from 0-5000 kilograms. We can see that most flights carried payloads within these ranges. The FT boosters seem to have the most failiures.

Here is a scatter plot with a payload range from 5000-0 kilograms. There are few launches in this range. We can see that the only boosters carrying the payloads here are the FT and B4 boosters



Section 6

Predictive Analysis (Classification)

Classification Accuracy

Accuracy of all methods (test data and training data)

-Logistic Regression: test score: 0.8333, training score: 0.8464

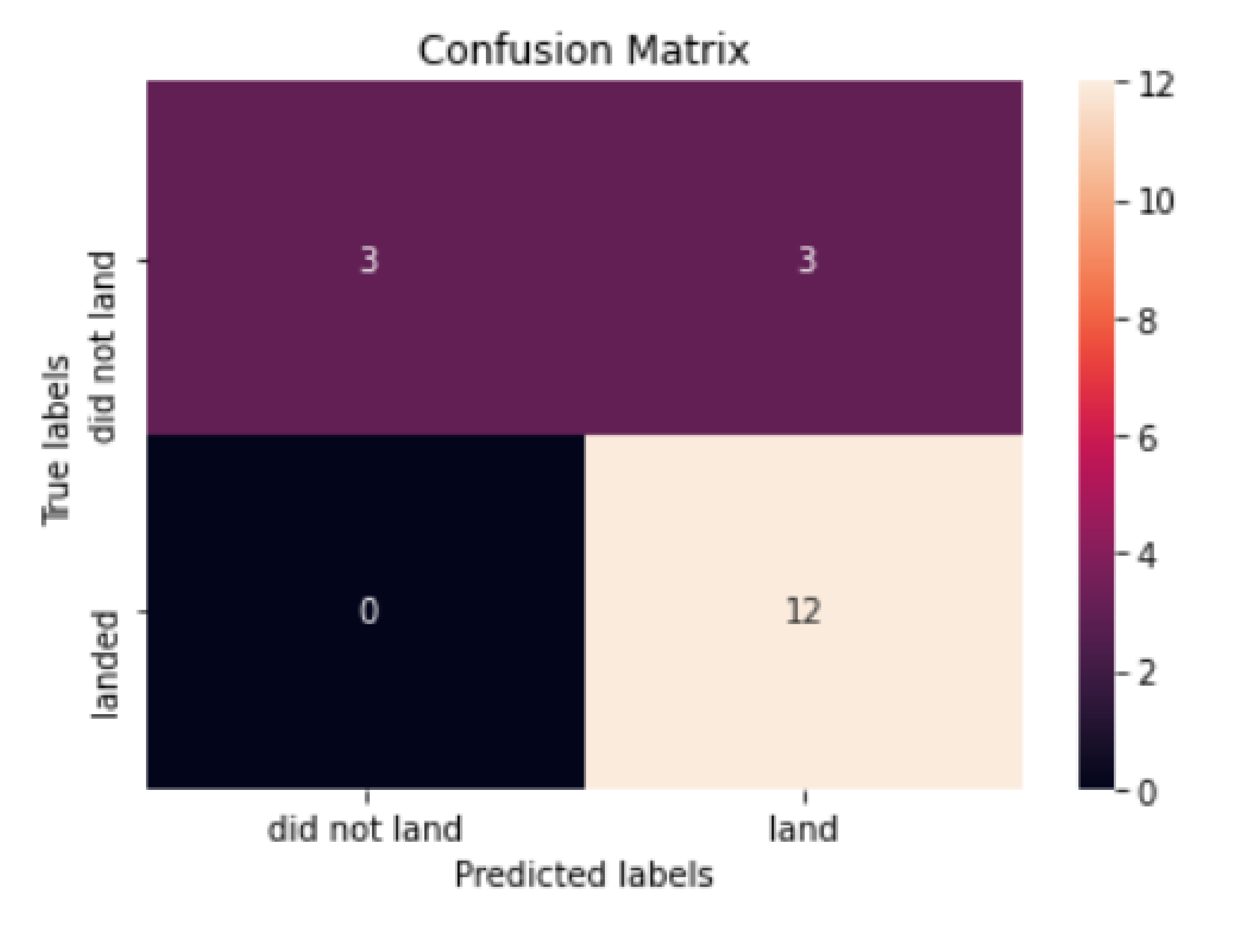
-Support Vector Machine: test score: 0.8333, training score: 0.8482

-Decision Tree Classifier: test score: 0.7778, training score: 0.8857

-K Nearest Neighbors: test score: 0.8333, training score: 0.8482

Taking the sum of all the scores (training score + test score), we can see that the Support Vector Machine method and the K Nearest Neighbors method are the most accurate (at 1.6635 score)

Confusion Matrix



Because both Support Vector Machine method and the K Nearest Neighbors method have the same accuracy, they have the same Confusion Matrix. Looking at the confusion matrix, we can see that both models are particulary accurate. However, there are 3 wrong values in the top right corner of the matrix.

Conclusions

- We first learned that there were multiple ways to collect data, either through web-scraping or loading a file from a cloud or computer.
- We then visualised data using Pandas, Seaborn, and more.
- We discovered that we can use Folium to make an interactive map.
- We also found out that using Plotly Dash, we can create interactive dashboards too.
- We also used SQL to fetch data from a database.
- Lastly, we used multiple methods to classify data, and found out that some methods are better than others.

Appendix

Datasets:

- Capstone/spacex_launch_geo (1).csv at main · Alexledev/Capstone (github.com)

Thank you!