IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Crisenger Guerisma
3/9/2024

Outline

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

Executive Summary

Methodologies Summary

- The business problem this project is the company's inability to estimate launch costs in order to provide the best quote to its clients.
- We concluded that we will be able to address this problem using existing data. Spacex has an API available where can get launch data. We will also get additional launch and rocket data from Wikipedia.
- We used REST API and Web Scrapping to collect the data, and used various data wrangling techniques to address missing values, ensure data attributes are in correct data types, etc.
- We performed data visualization to gain insight in the data and to ensure it is sufficient to help us answer the business problem
- Finally, we build several machine learning models to help predict the outcome of a launch based on reusability of the Falcon 9 rocket.

Results Summary

- Most of the successful launches happened at KSC LC-39A
- Most of the successful launches happened for orbits ES-L1, GEO, HEO, SSO and VLEO

Introduction

Interest in space flight have never been higher. Many companies are competing to see who can deliver payloads and space tourists in orbit at the most affordable price. Several companies have been able to launch rockets into space, but SpaceX, by far, is the clear winner of the competition by producing a genius idea to re-use stage one of the falcon 9, which allow them the ability to provide the service at much lower price.

The financial cost of a trip depends entirely on the successful reuse of the stage 1 rocket, but currently, there is no way to predict whether it will successfully land or not. This project is to build a statistical model to help answer this question.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The first set of data for launch information was gathered using launch data API provided by SpaceX
  - The second set of data for the Falcon 9 and Falcon Heavy Launches Records were collected from Wikipedia using Web scrapping.
- Perform data wrangling
  1. We replaced null values in the PayloadMass column with the mean value that we calculated.
  2. Filtered the data to keep record only related to the Falcon 9 rockets.
  3. Not all NULL values were replaced because they are valid reasons that indicate an event did not happen, such as when a landing site is not used.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

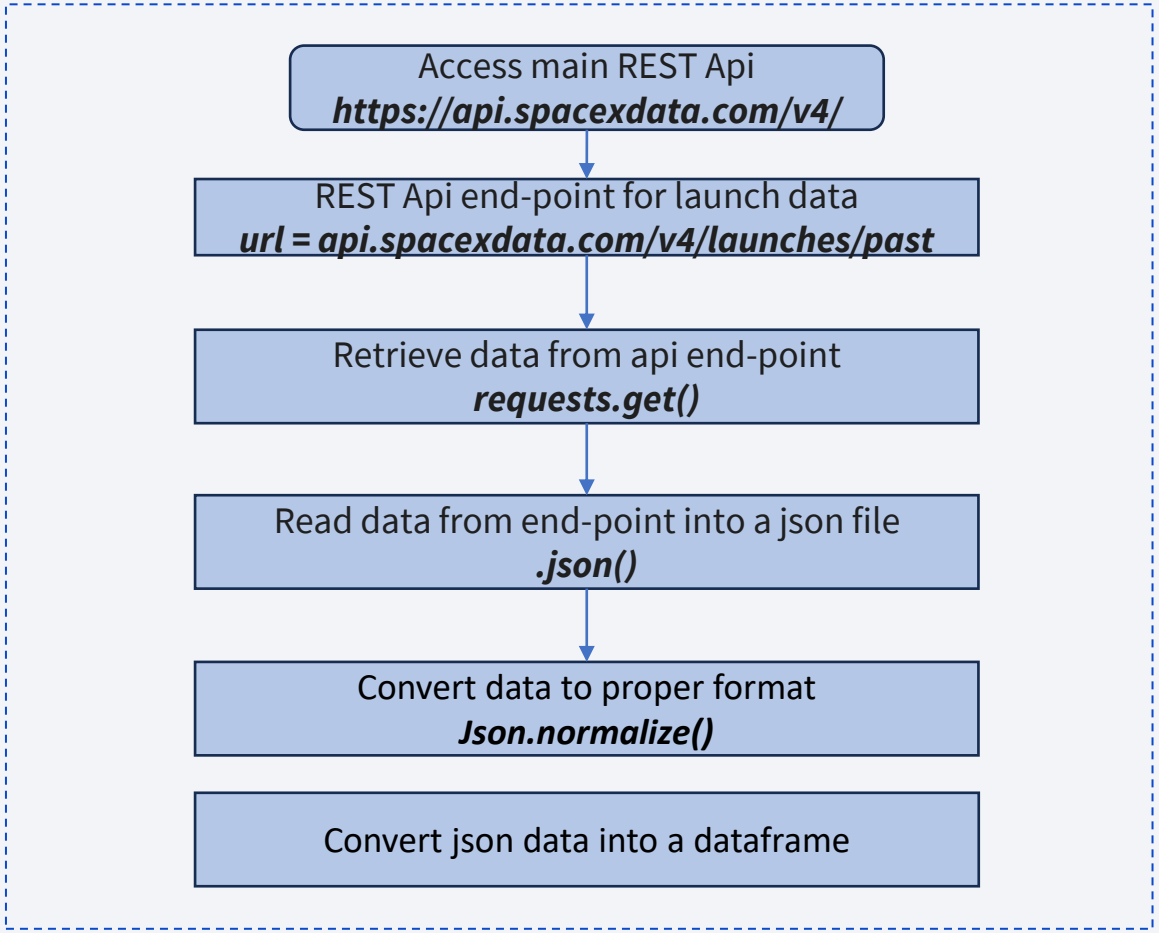
We used the SpaceX API to collect data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. We receive the data in .JSON() format and we converted it into data frames for easier data handling and manipulation.

We scrapped additional datapoint from the following website: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches " where we obtaining Falcon 9 Launch data. We used Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. We then parse the data from those tables and convert them into a Pandas data frame.

Since the data only contain reference ids or identification numbers, we used the API again targeting another endpoint to gather specific data for each ID number.

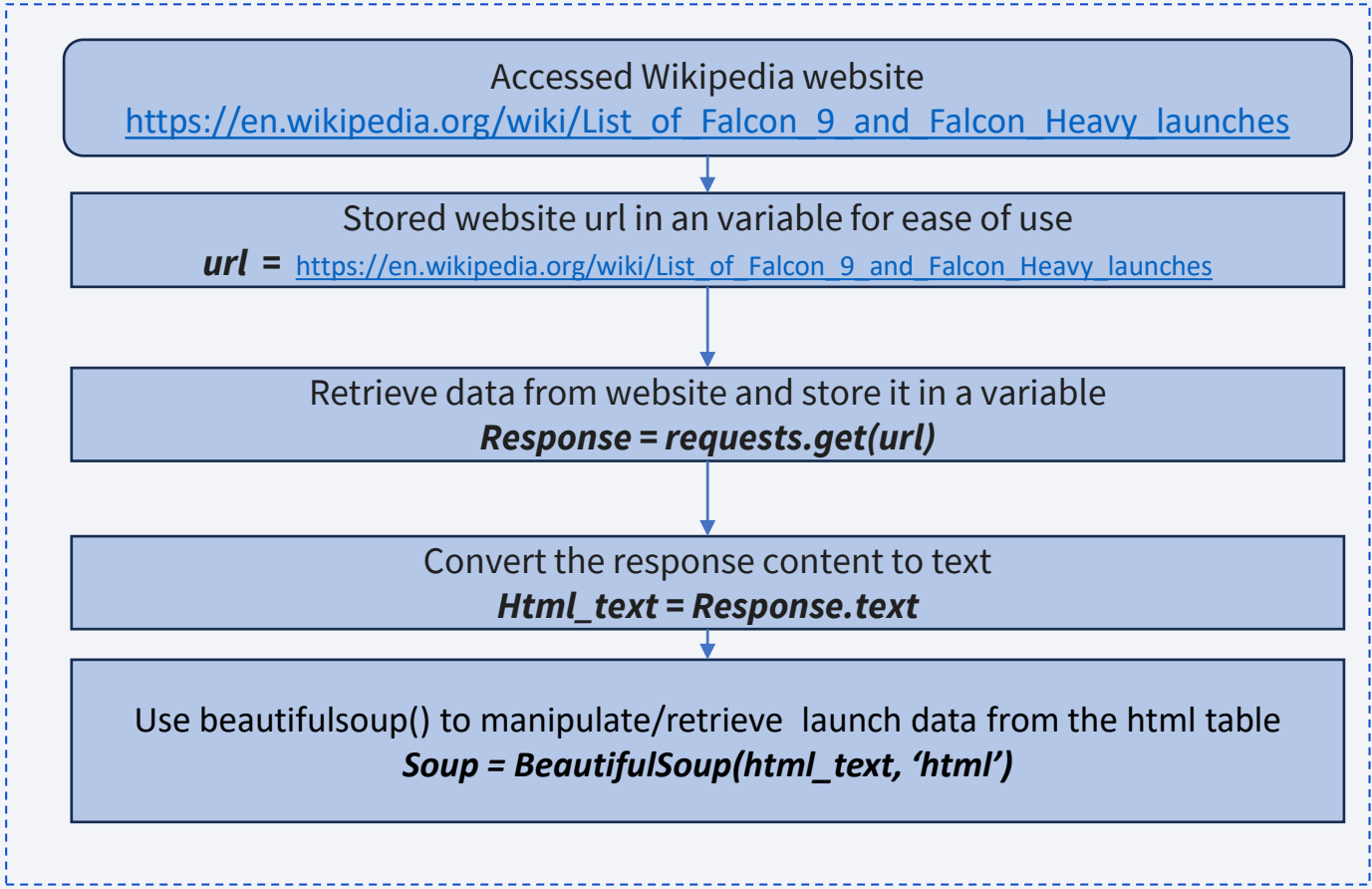
Data Collection – SpaceX API

https://github.com/gcris07/launchpad/blob/d8de0d848c9e9314390e81085b34ad6e4a2b5078/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

https://github.com/gcris07/launchpad/blob/aaaa2bf81048d426076f26d821f2cc220d448c88/jupyter-labs-webscraping.ipynb



Data Wrangling

After collecting the data using the Spacex API and scrapped the Wiki page for addition launch and payload details, we proceeded to start wrangling the data using the following processes:

1. Since some data from our initial collection were ID numbers, I used an additional APIs from to collection the necessary additional launchpad, Booster, payload, and core details.
2. The data was stored in list, which in turned to be converted into data frames from additional processing.
3. Below are some additional steps taken to clean the data:
    1. We replaced null values in the PayloadMass column with the mean value that we calculated.
    2. Filtered the data to keep record only related to the Falcon 9 rockets.
    3. Not all NULL values were replaced because they are valid reasons that indicate an event did not happen, such as when a landing site is not used.

EDA with Data Visualization

The chart listed below were used to help visualize the data in order quick find insights and patterns:

- The scatter plot chart was used to help visualize relationship between variables. (example, the relationship between Flight Number and Launch Site)
- The bar chart was used to help us compare multiple categories of datapoints. (example, it helps me to quick identify what orbit has the highest success rate)
- The line chart was used to help identify success rate trends over time

https://github.com/gcris07/launchpad/blob/d8de0d848c9e9314390e81085b34ad6e4a2b5078/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Select Distinct query was used to retrieve unique launch sites from the data
- Select with LIKE operator, and LIMIT clause was used to return for records with a specific string
- Select with aggregate functions were used to calculated Sum, Average, Minimum, Maximum, etc.
- Subqueries were used to help find the booster versions that have carried the max payload mass.
- Other functions like substr() were used to help extract specific character from strings

https://github.com/gcris07/launchpad/blob/d8de0d848c9e9314390e81085b34ad6e4a2b5078/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- I used *folium.Circle* in order to clearly circle around area of interest. In the scenario, it was used to circle around launch sites.
- I used *folium.markers* to add text, icon, etc. to mark the area of interest. In this scenario it was used to display the launch site's name.
- I used *MarkerCluster* to help simplify the current map since many of the markers have the same coordinate, therefore can all be group together in cluster.
- I used *mouseposition* to help me identify the coordinates (lat and long) of a specific point of interest on the map. This help calculate the distance between that point and a launch site.
- I used *folium.PolyLine* to display a line representing the distance between a point of interest and a launch site.

https://github.com/gcris07/launchpad/blob/d8de0d848c9e9314390e81085b34ad6e4a2b5078/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- I added a pie chart to be able to visualize the percentage of successful launch per site.
- I added a scatter chart to help me identify any correlation between payload and successful launch for all sites.
- I added a dropdown to help me filter the data for all or individual sites. I also added a slider to help identify correlation between specific payload mass and booster version category.

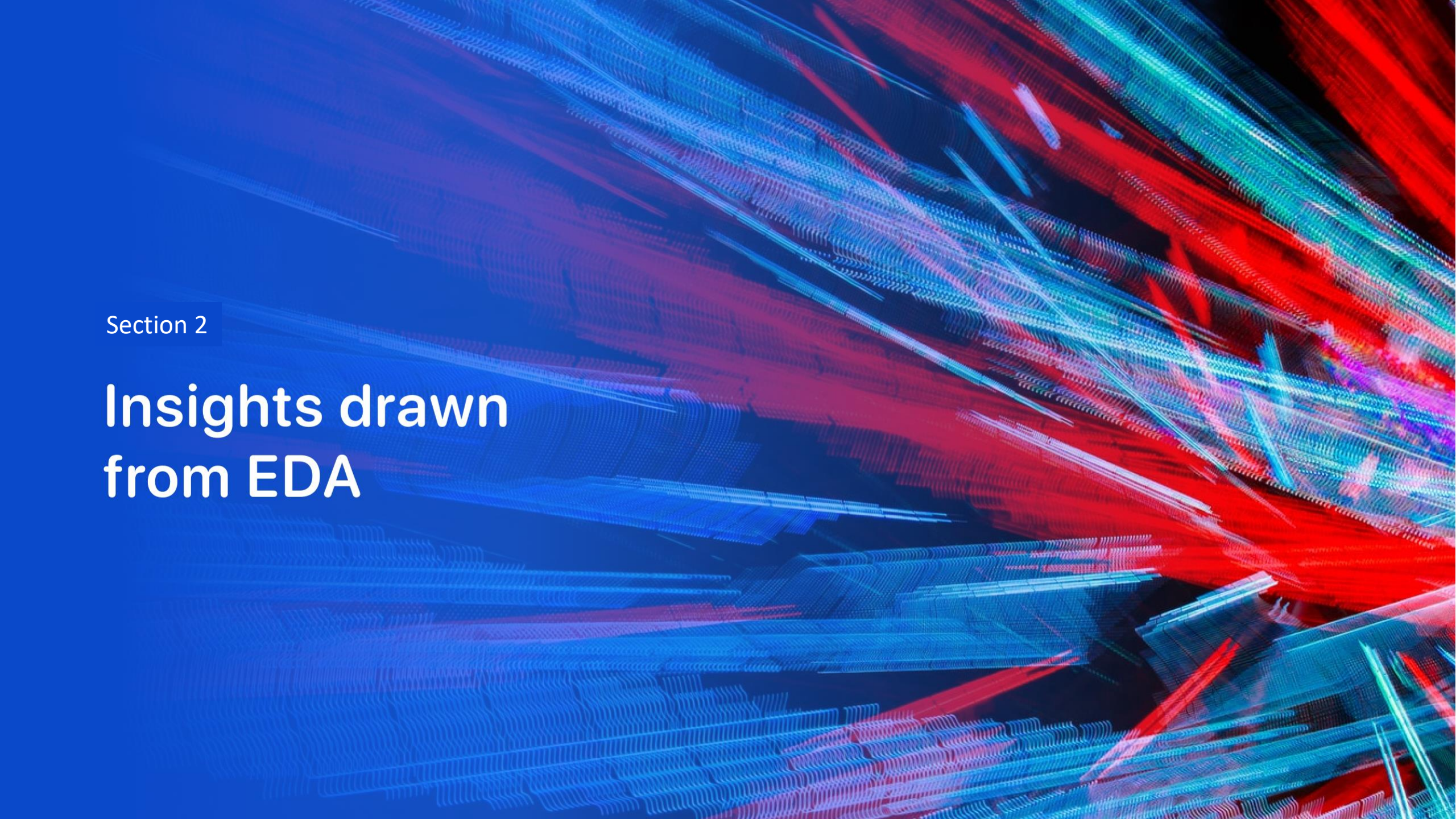
https://github.com/gcris07/launchpad/blob/58e3de873618f76cf8a7a567bca2f23219aea0b2/spacex_dash_app.ipynb

Predictive Analysis (Classification)

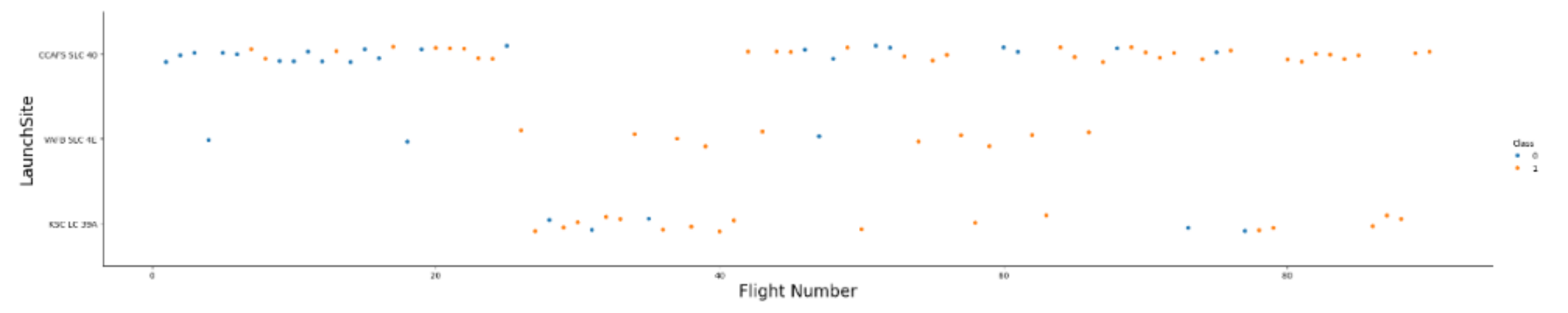
- After preparing the data for modeling. I elected to build a Logistic Regression, a Support Vector Machine, a Decision Tree and a K Nearest Neighbor model to predict the outcome of a launch.
- I used Train_Test_Split method to divide the data into training and test sets.
- I used GridSearch to help find the best parameters.
- I scored the model for accuracy and then I used the result to determine the best model
- https://github.com/gcris07/launchpad/blob/f3cfb4dce24f57f47cc5e264839288a6d098c60f/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb

Results

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

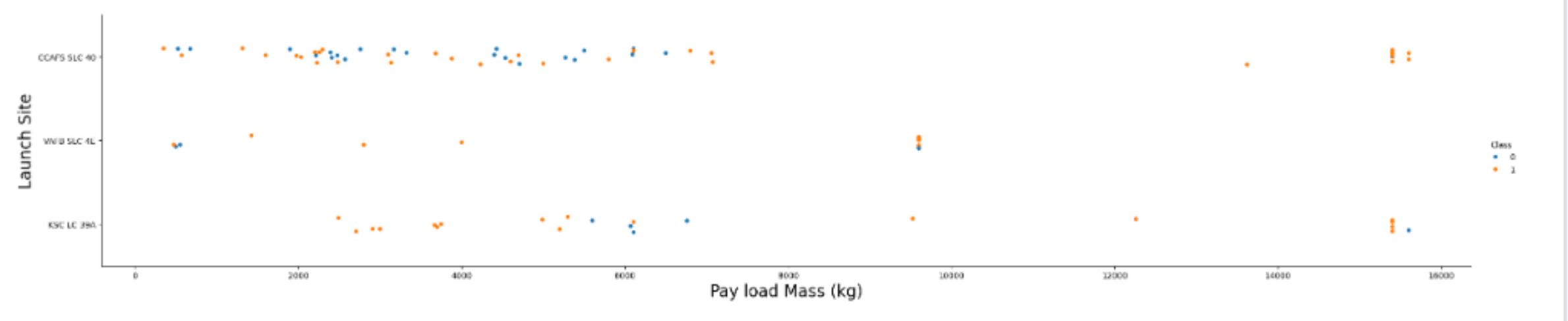

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



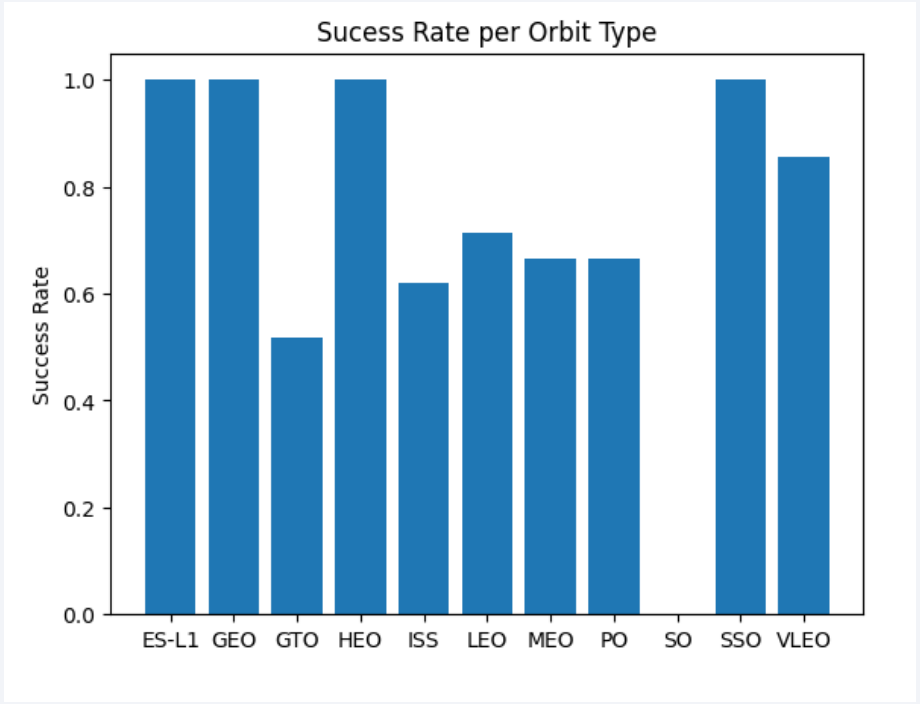
1. We that for sites CCAFS LC-40 and KSC LC 39 A have more successful launch when flight number is 80 or greater while VAFB SLC 4E has more success rate between 20 and 80

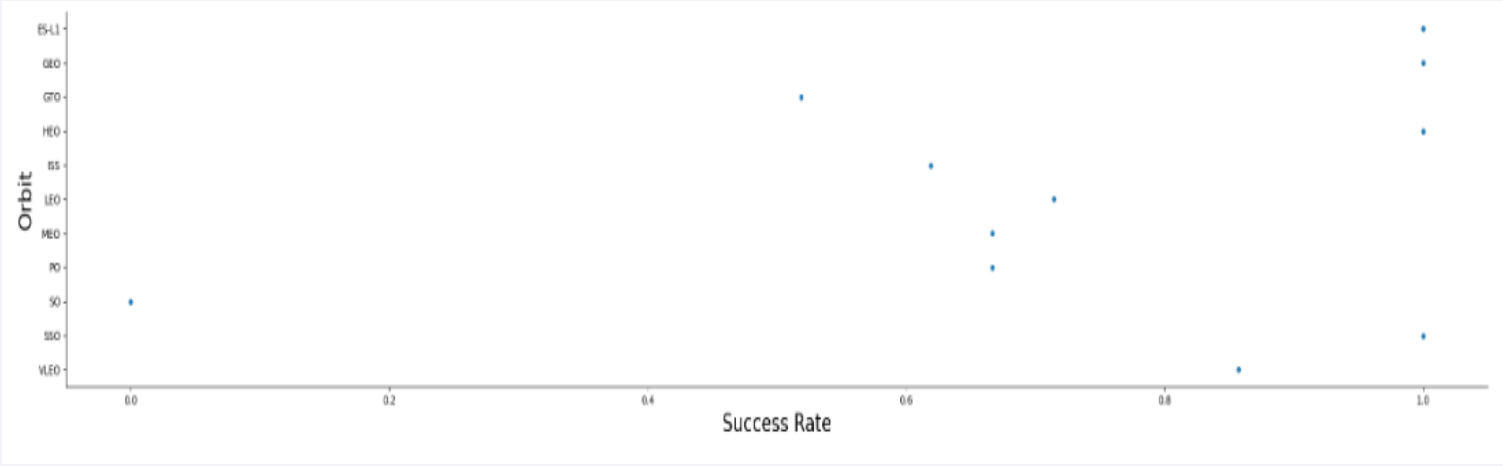
Payload vs. Launch Site



Payload Vs. Launch Site scatter point chart shows that for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

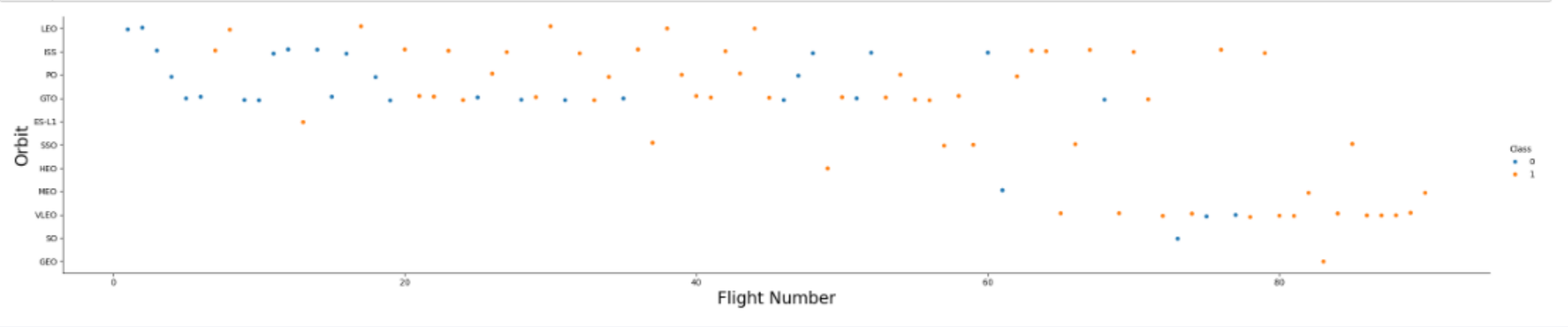
Success Rate vs. Orbit Type





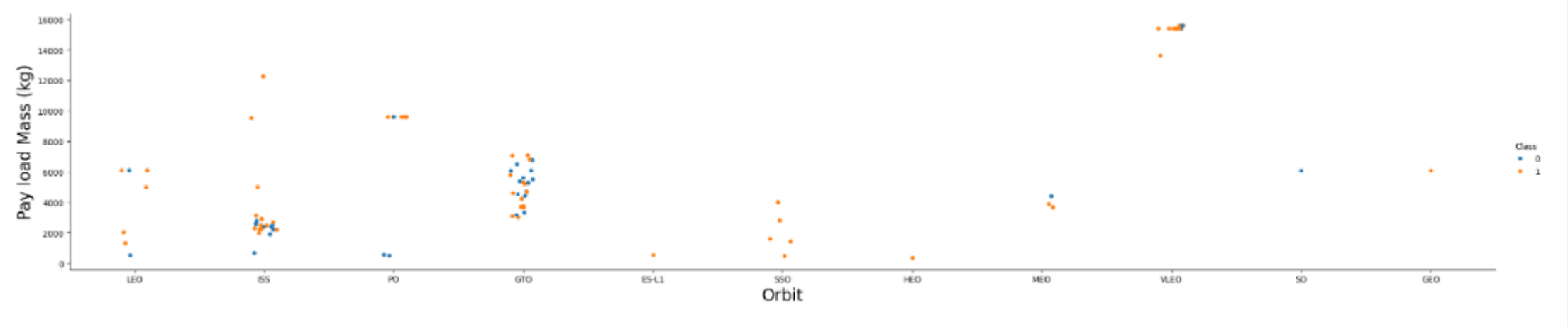
Both graphs show that orbits ES-L1, GEO, HEO, SSO and VLEO are have the highest success rate.

Flight Number vs. Orbit Type



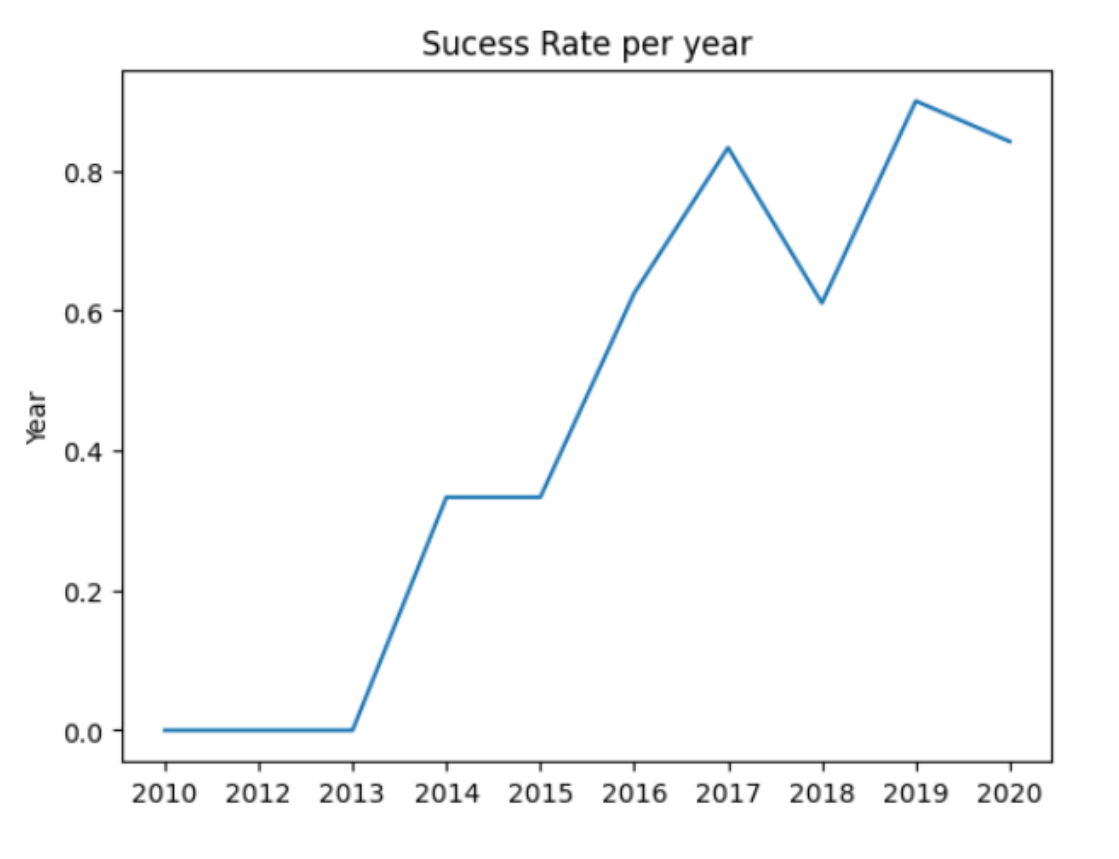
The data show that LEO orbit's success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



The successful landing rate are more for Polar, LEO and ISS when payloads are heavy. However, for GTO payloads do not seem to make a difference.

Launch Success Yearly Trend



Success rate has been trending upward since 2013 up to 2020

All Launch Site Names

Results:

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

I used distinct in to return only unique site names, since a site can have multiple records in the database

Launch Site Names Begin with 'CCA'

Results:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Using the “like” operator return any record with the string CCA in it.

Total Payload Mass

Result:

Total Payload
619967

Used the sum function to add all the payloads amount

Average Payload Mass by F9 v1.1

Result:

Total Payload
2534.6666666666665

I used the avg() function to calculate the average pay for all F9 v1.1 rockets

First Successful Ground Landing Date

Result:

First successful ground pad landing date
2015-12-22

I use the min() function to find the earliest date for the successful ground pad landing outcome

Successful Drone Ship Landing with Payload between 4000 and 6000

Result:

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

I used AND and Between for the booster version between 4000 and 6000 kgs

Total Number of Successful and Failure Mission Outcomes

Result:

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

I used “LIKE” and “Group by” to find the total number of successful and failure outcomes.

Boosters Carried Maximum Payload

Result:

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

I used a subquery to find the maximum payload for booster version

2015 Launch Records

Result:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

I used substr() to find the failed landing from the date for 2015

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

Result:

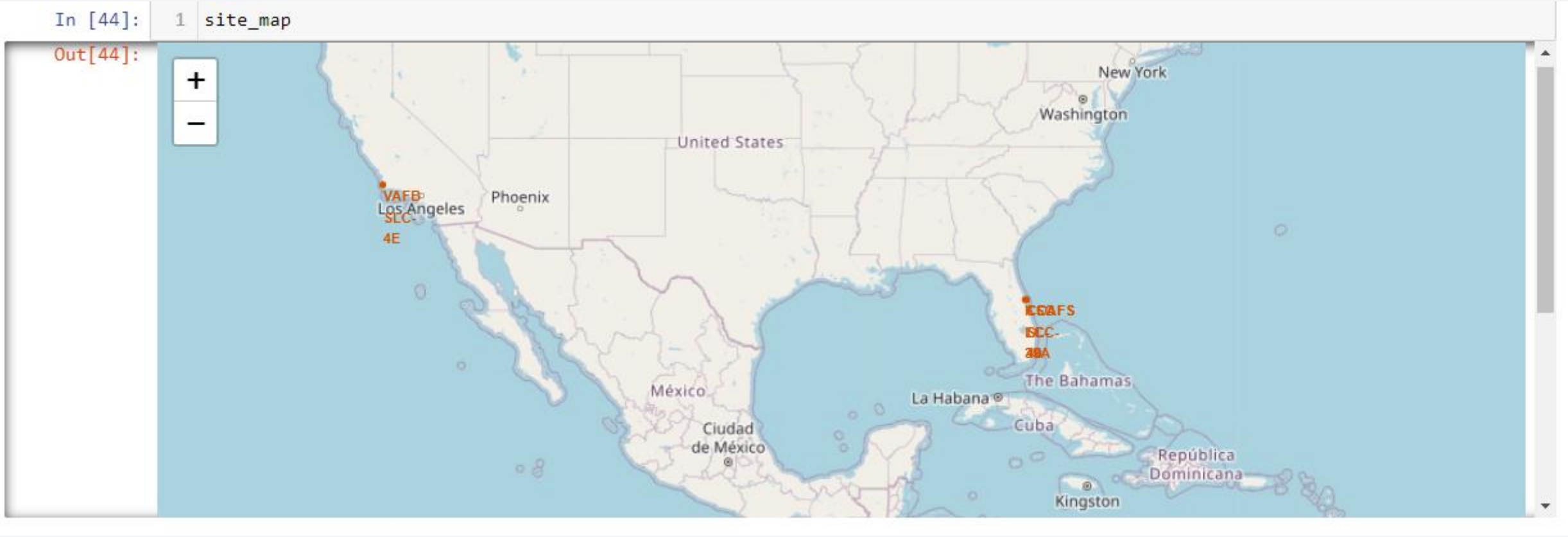
Total	Landing_Outcome
3	Success (ground pad)
5	Failure (drone ship)

I used COUNT() and IN() to pull the records from the data

Section 3

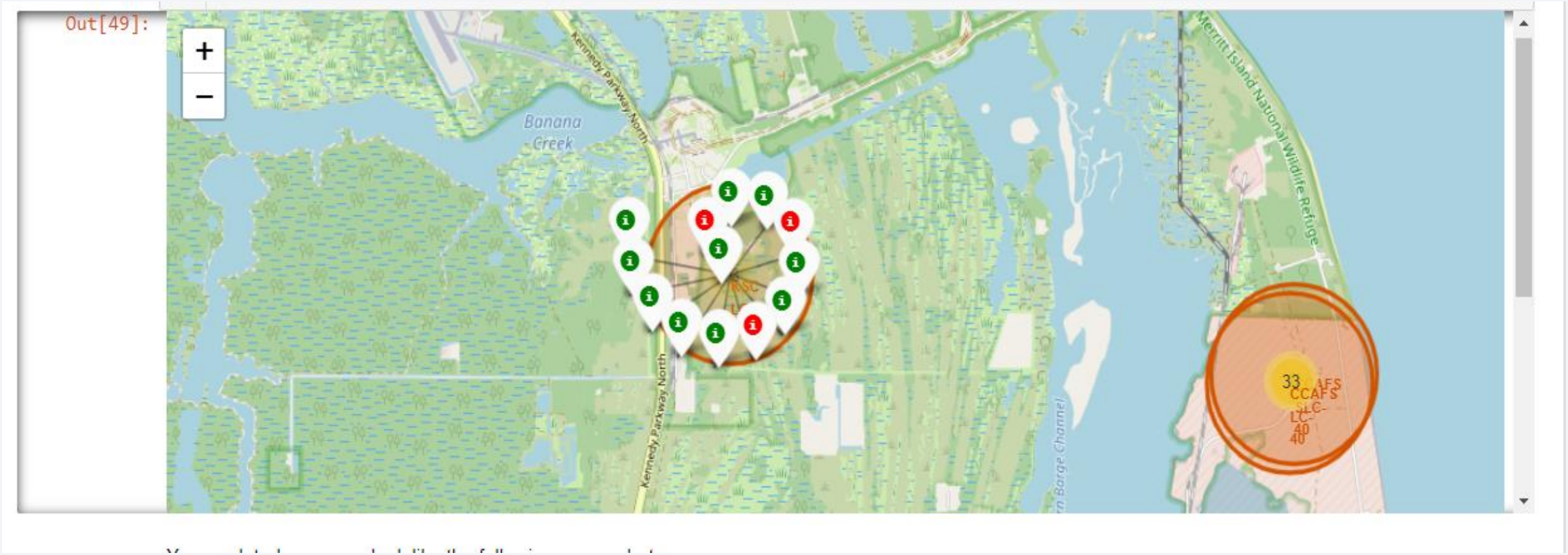
Launch Sites Proximities Analysis

Launch site with the continental US



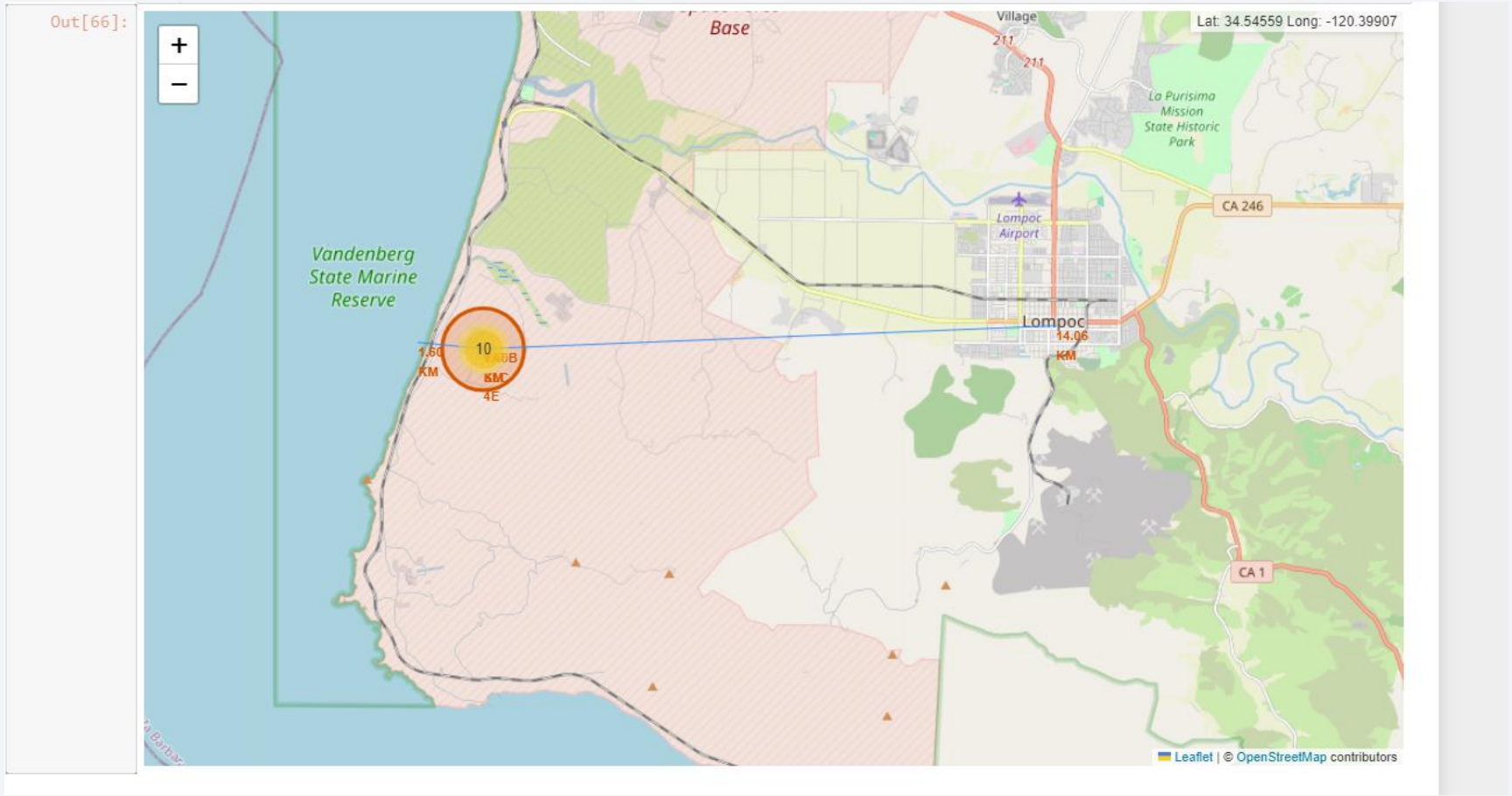
1. We see that the launch site are location on west and east coast of the US
2. Each launch site name is displayed

Color coded landing outcome per site

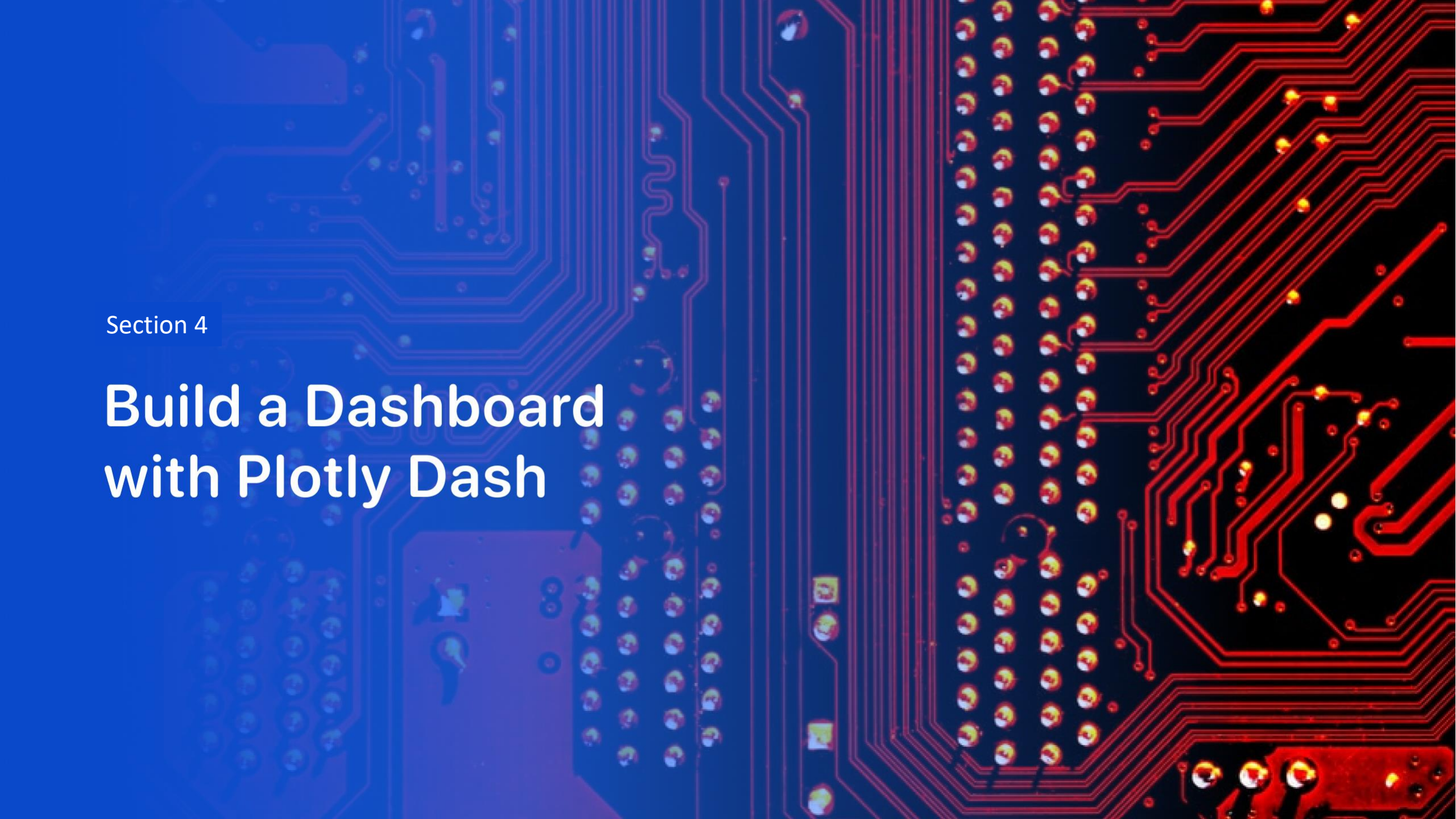


1. We that landing outcomes are marked in color for a launch site

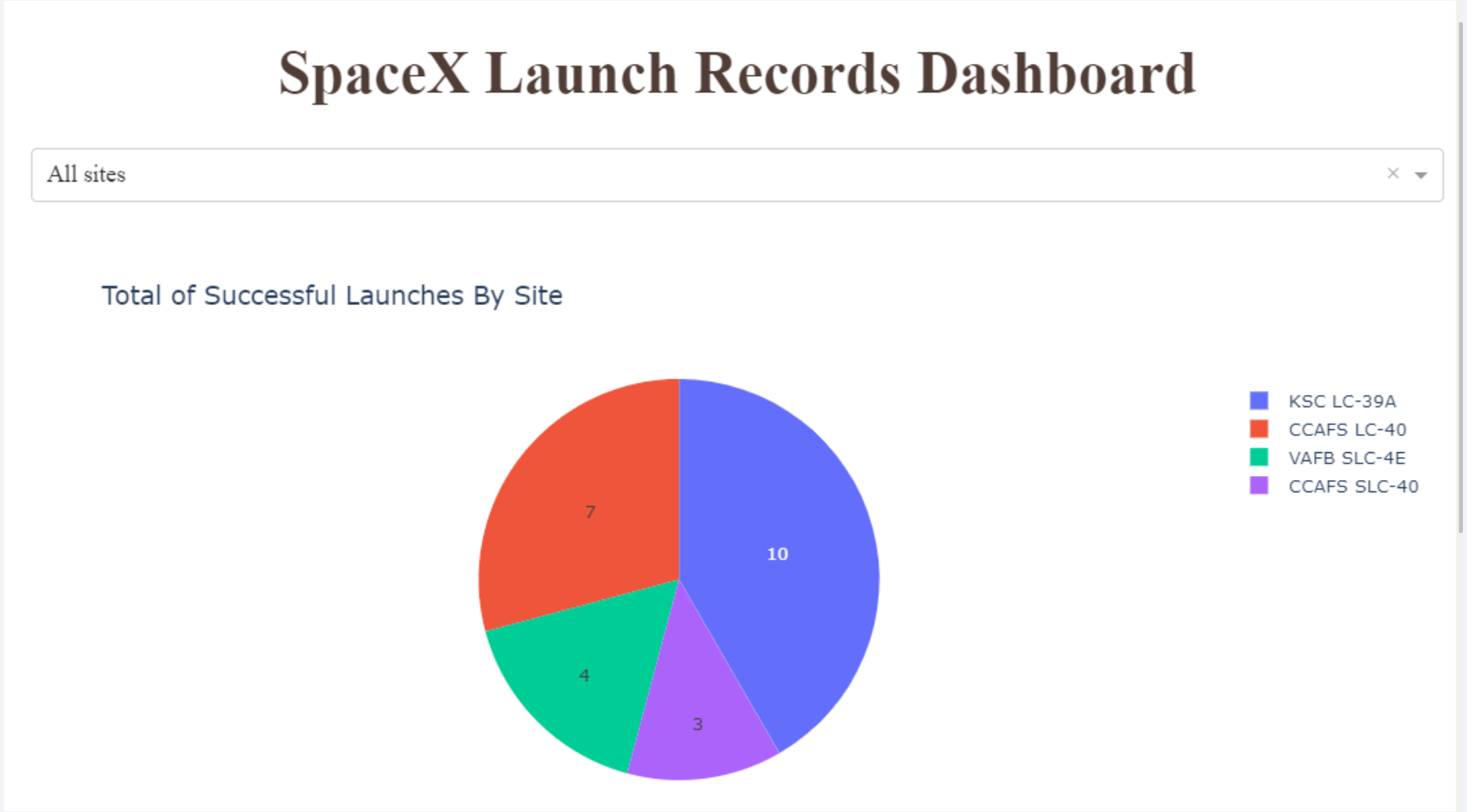
Distance between Lompoc and the Launch Site



1. We see that the distance between the city of Lompoc is displayed, and a line is drawn between them.

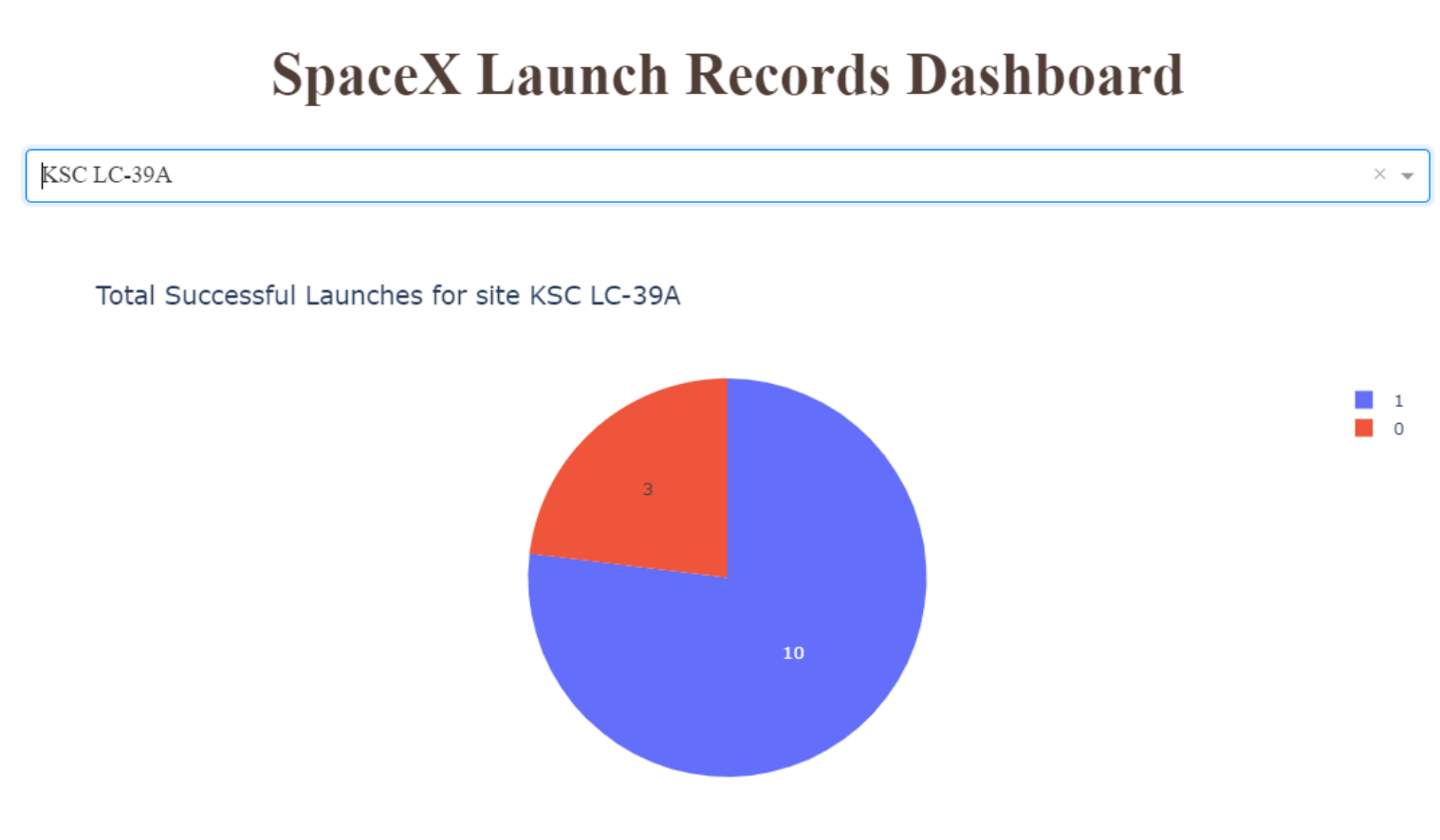

Section 4
Build a Dashboard with Plotly Dash

Total of Successful Launch Per Site



The KSC LC-39A has the most Successful launch

Total Successful Launch for Site KSC LC-39A



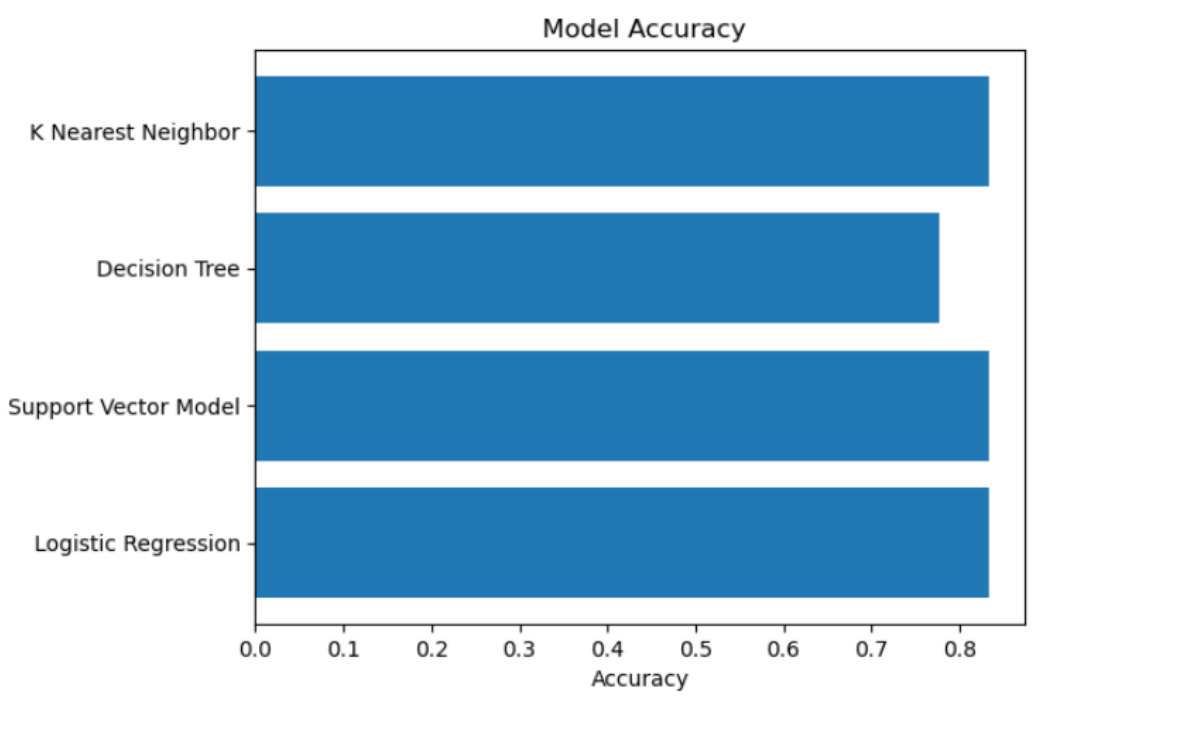
Correlation between payload and success for all site

Only v1.1, FT, and B4 have success between 4500 and 5500 kgs

Section 5

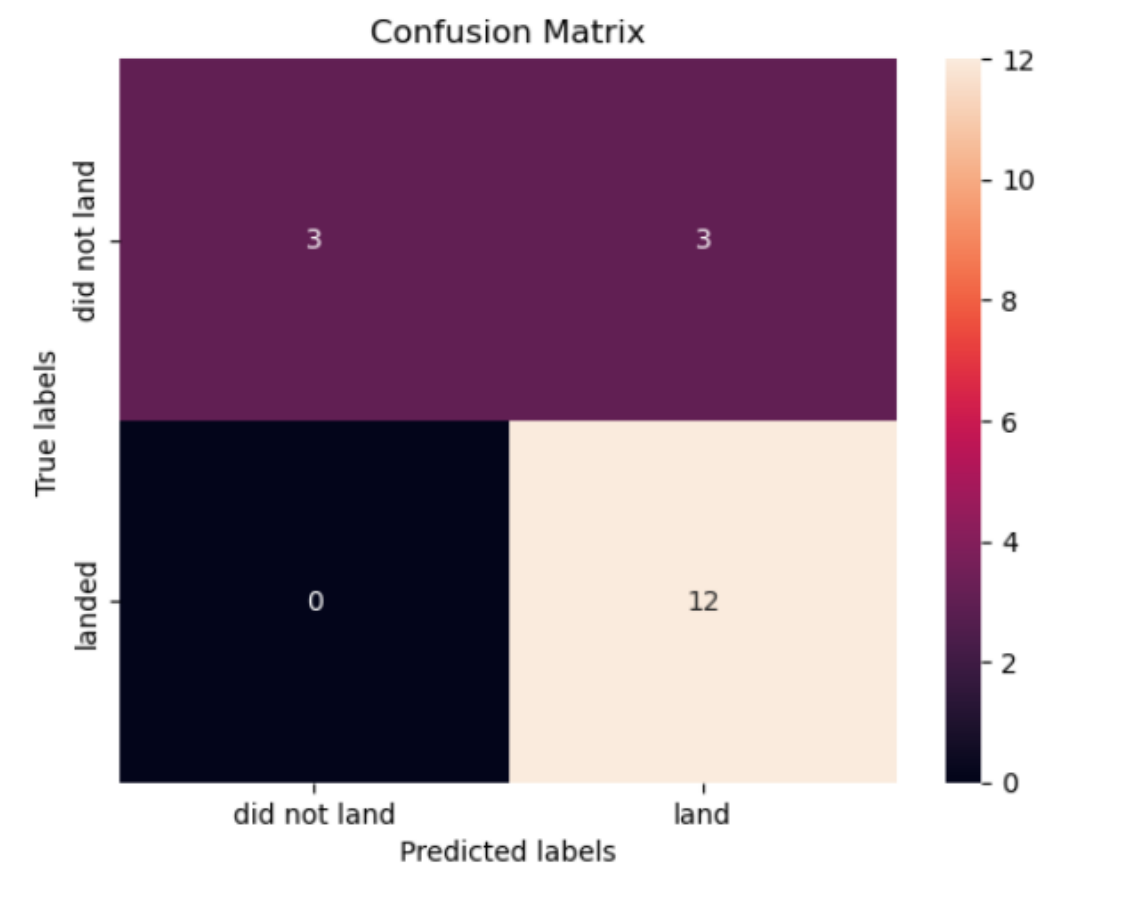
Predictive Analysis (Classification)

Classification Accuracy



3 of the model share the same accuracy which is the highest

Confusion Matrix



Three of the models have the same confusion matrix, which shows that the models has a type 1 error (False Positive) where is classifying launches that we not successful as successful.

different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

Conclusions

- Success rates differ per site CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- Most of the successful launches happened for orbits ES-L1, GEO, HEO, SSO and VLEO
- There are no rockets launched for heavypayload mass (greater than 10000) at the VAFB-SLC launch site
- LEO Orbit tends to have more success as there is clear correlation between number of flights.
- Polar, LEO and ISS tend to have better successful or positive landing rate with heavy payloads.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!