IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Andreas F.
16.09.2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Interactive Dashboards with Plotly Dash
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive Analytics in pictures
  - Predictive Analysis results

Introduction

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage on their next missions. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used for our competitor SpaceY to bid against SpaceX for a rocket launch.

Problems to examine:

- Identification of all factors influencing the landing outcome
- Finding relationships between all variables and how these affect the outcome
- Evaluating best conditions needed to increase the probability of a successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - Converting categorical data variables for machine learning purposes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building of Logistic Regression, Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbours models for evaluating the best classifier

Data Collection

Data collection involves gathering and measuring information on specific variables within a predefined system, allowing for the answering of key questions and assessing outcomes.
In this case, the dataset was obtained via a combination of REST API and web scraping from Wikipedia.

For the REST API, the process began with a GET request. The response was decoded into JSON format, which was then transformed into a Pandas DataFrame using json_normalize().
After that, the data was cleaned, missing values were identified, and appropriate actions were taken to fill them.

As for web scraping, BeautifulSoup was employed to retrieve launch records formatted as an HTML table. This table was parsed and converted into a Pandas DataFrame for further analysis.

Data Collection – SpaceX API

1. Getting the request for the rocket launch data using API
2. Using the json_normalize method to convert the json result to a dataframe
3. Performing data cleaning and filling the missing value

From:
https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/Data%20Collection%20API.ipynb

```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocke
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the l
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date le
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

1. Getting the response from HTML
2. Creating a BeautifulSoup object
3. Extracting all columns and variable names from the HTML Header

From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/Data%20Collection%20with%20Webscraping.ipynb

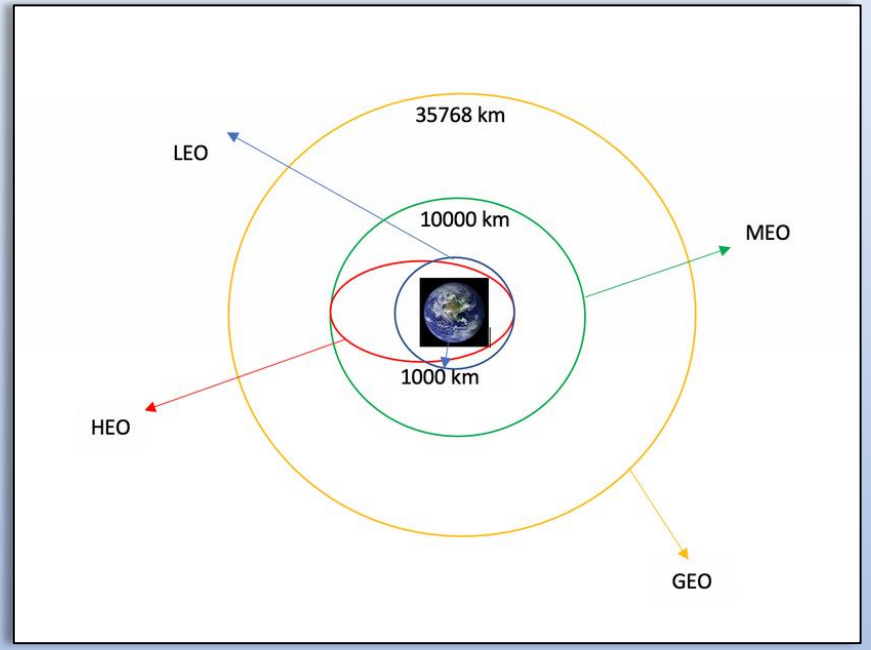
```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data,'html.parser')
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
```

Data Wrangling

- Data wrangling refers to the process of transforming and organizing disordered or intricate datasets to make them more accessible for analysis and Exploratory Data Analysis (EDA).
- It began by determining the number of launches at each location, followed by calculating the frequency and distribution of mission outcomes for each orbit type.
- Afterward, a landing outcome label has been generated based on the outcome column, simplifying future analysis, visualization, and machine learning tasks. Finally, the processed data has been exported as a CSV file.

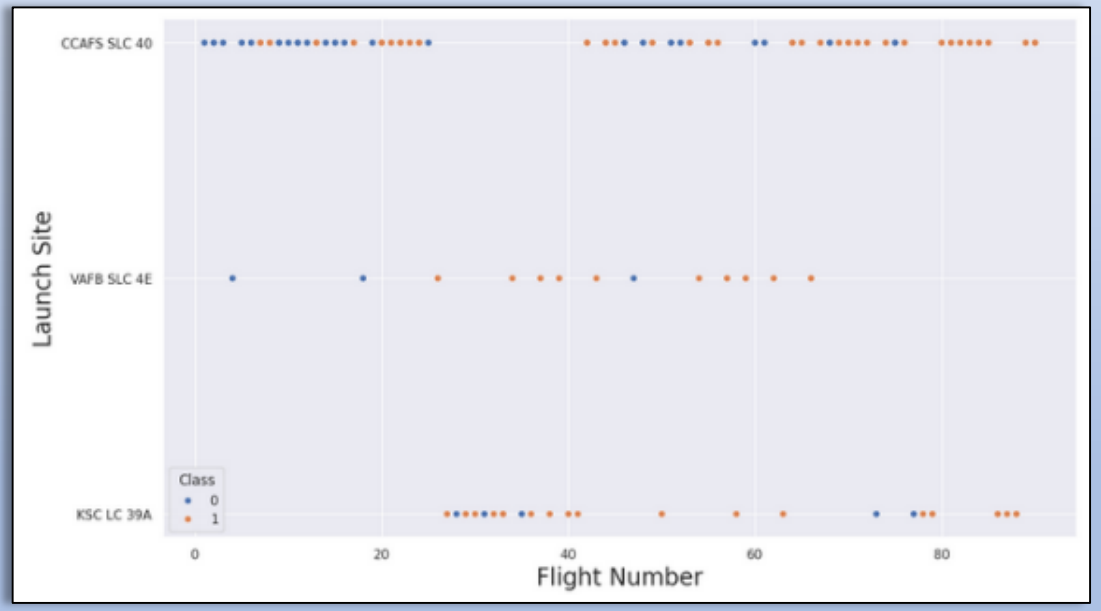


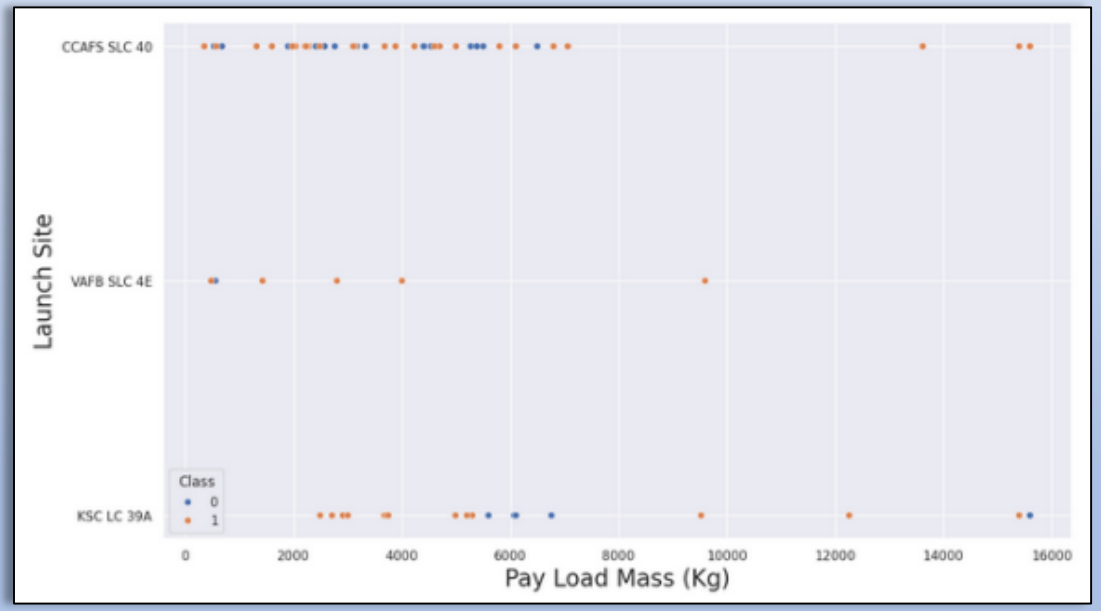
From:
https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

At the start scatter graphs were used to find relationships between the following attributes:

- Flight number and launch site
- Payload mass and launch site



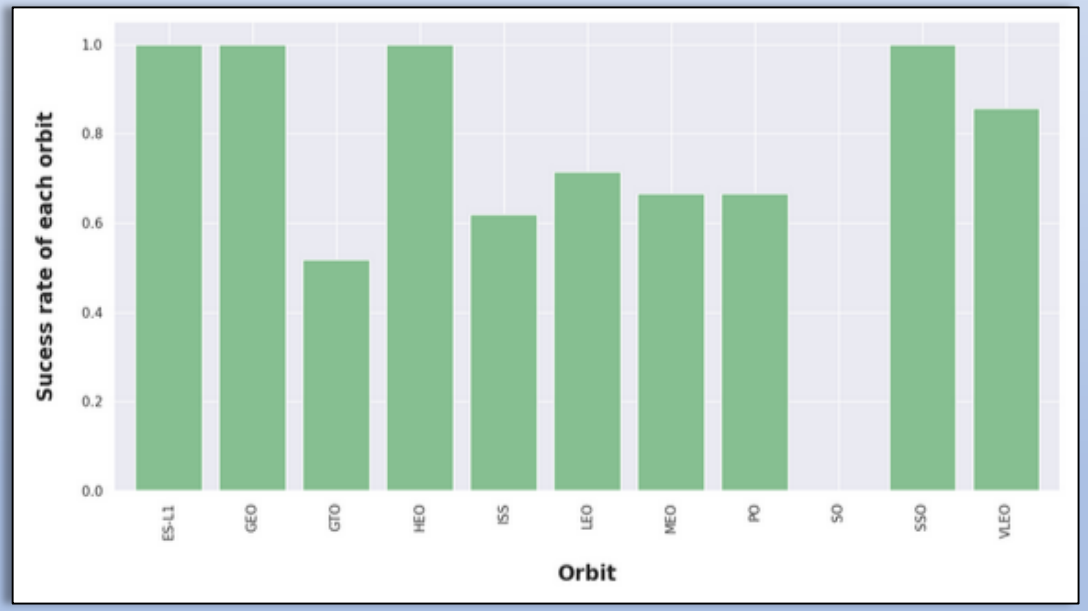


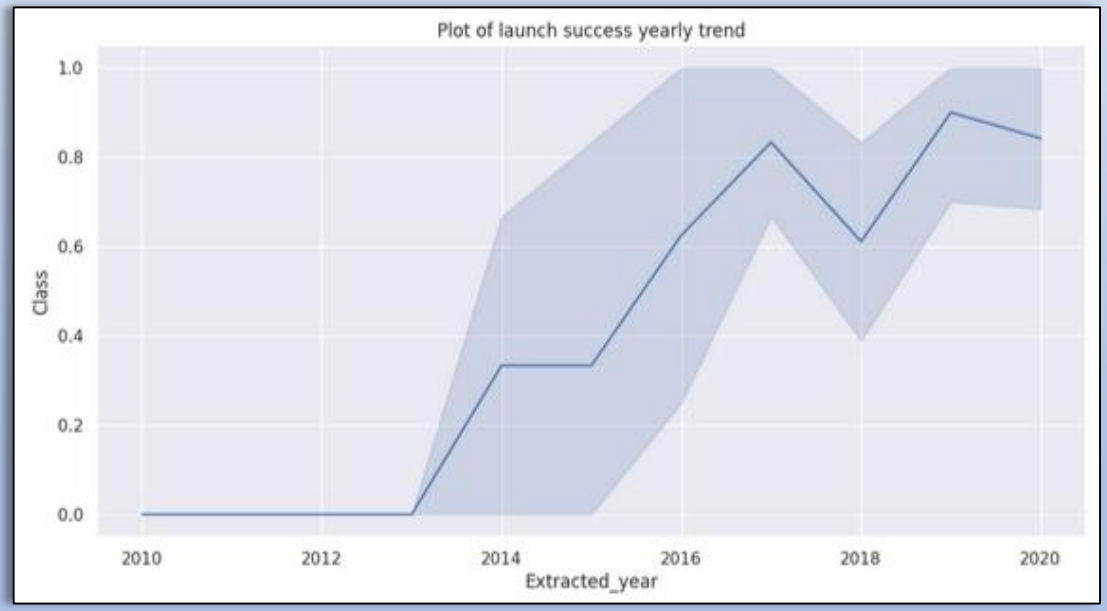
From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20Data%20Visualization.ipynb

EDA with Data Visualization

Afterwards bar and line charts were used to interpret the relationship between the attributes.

- The following bar chart determined which orbits have the highest probability success
- The line chart was used to show trends or patterns of attributes over time, in this case the launch success yearly trend





From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

Using SQL, various queries were executed to gain deeper insights into the dataset, such as:

- Retrieving the names of the launch sites.
- Fetching 5 records where the launch site names start with 'CCA'.
- Calculating the total payload mass carried by boosters launched by NASA (CRS).
- Finding the average payload mass for the booster version F9 v1.1.
- Identifying the date of the first successful landing outcome on a ground pad.
- Listing the boosters that had successful landings on drone ships and carried a payload mass between 4000 and 6000.
- Counting the total number of successful and failed mission outcomes.
- Listing the booster versions that transported the maximum payload mass.
- Retrieving failed landing outcomes on drone ships, along with booster versions and launch site names, for the year 2015.
- Ranking landing outcomes or successes between 2010-06-04 and 2017-03-20 in descending order.

From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

To create an interactive map for visualizing the launch data, the latitude and longitude coordinates for each launch site have been obtained and a circle marker has been placed at each location, labeled with the launch site name.

Next, the launch_outcomes dataframe (failure, success) has been categorized into classes 0 and 1, using red and green markers, respectively, within a MarkerCluster() on the map.

Then the Haversine formula has been applied to calculate the distances between launch sites and various landmarks to address questions such as:

- How close are the launch sites to railways, highways, and coastlines?
- How close are the launch sites to nearby cities?

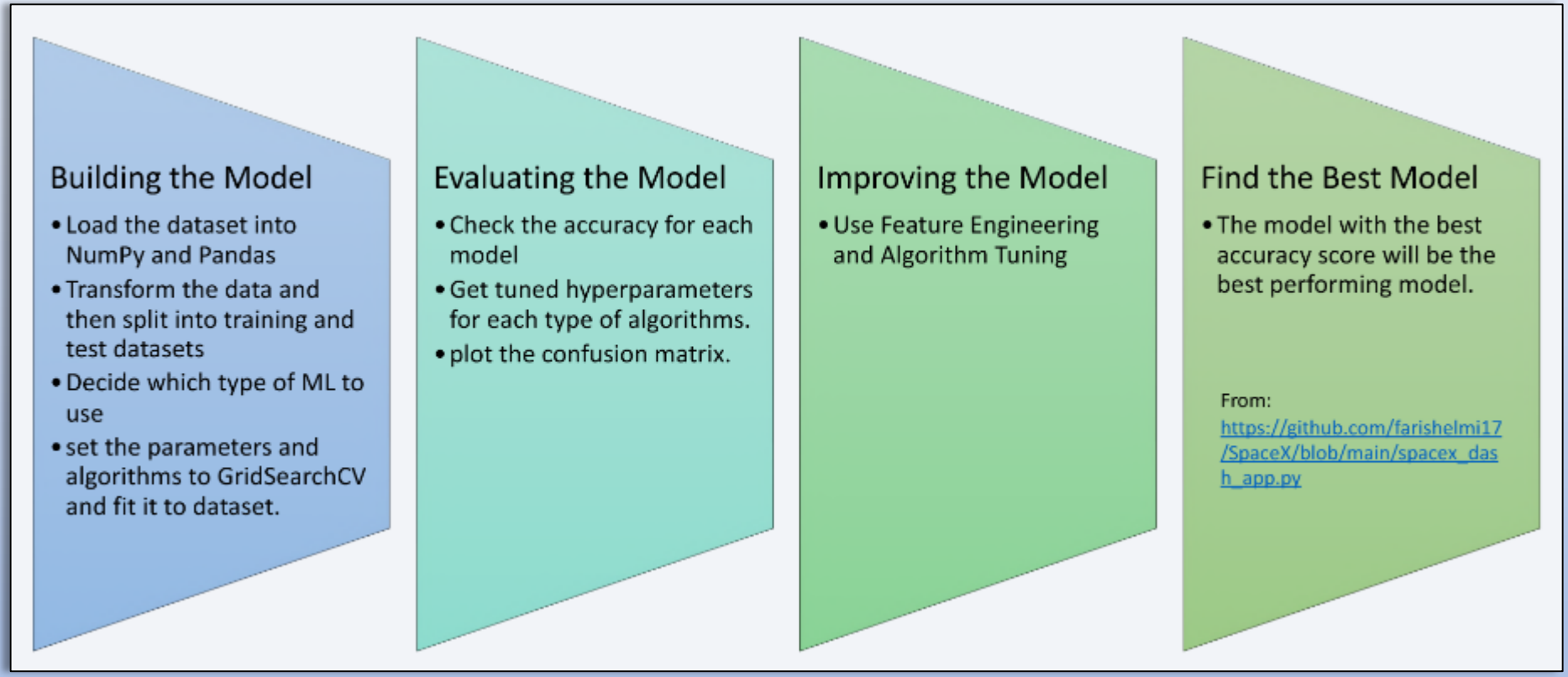
From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard has been build with Plotly Dash
- For showing the total launches by certain sites plotted pie chars were used
- To see the relationship between Outcome and Payload mass (kg) for the different booster version, plotted scatter graphs were included

From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

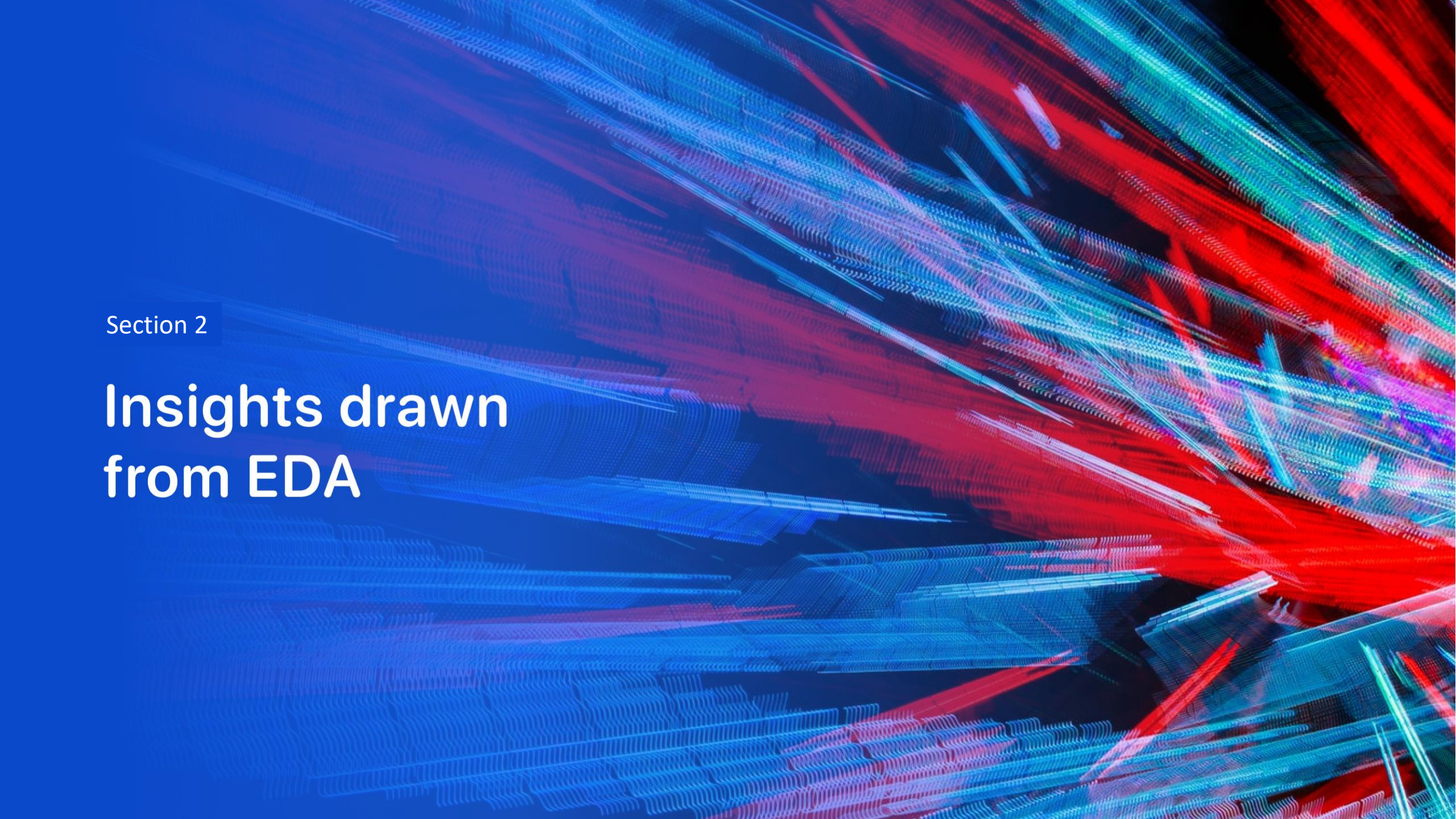
Predictive Analysis (Classification)



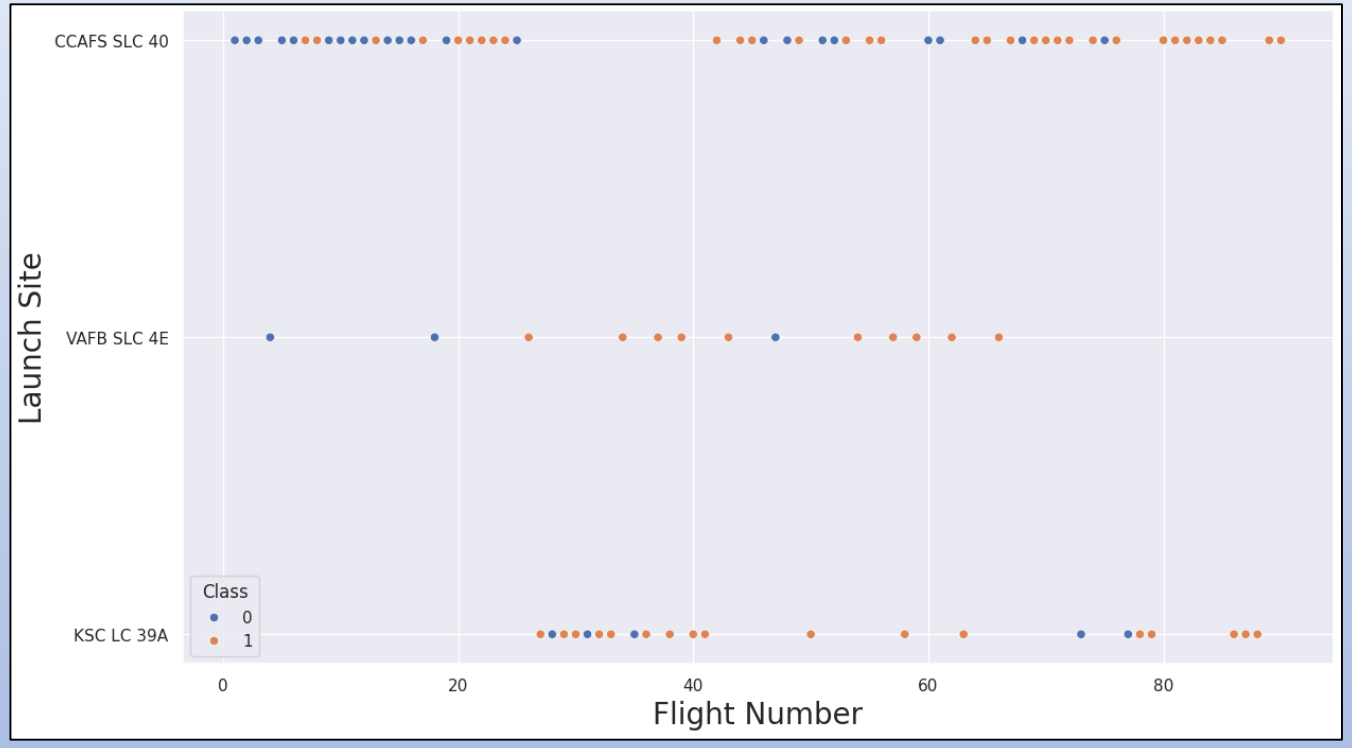
From: https://github.com/Faqqin/Applied-Data-Science-Capstone-SpaceX/blob/main/Predictive%20Analysis%20Machine%20Learning.ipynb

Results

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


Section 2
Insights drawn from EDA

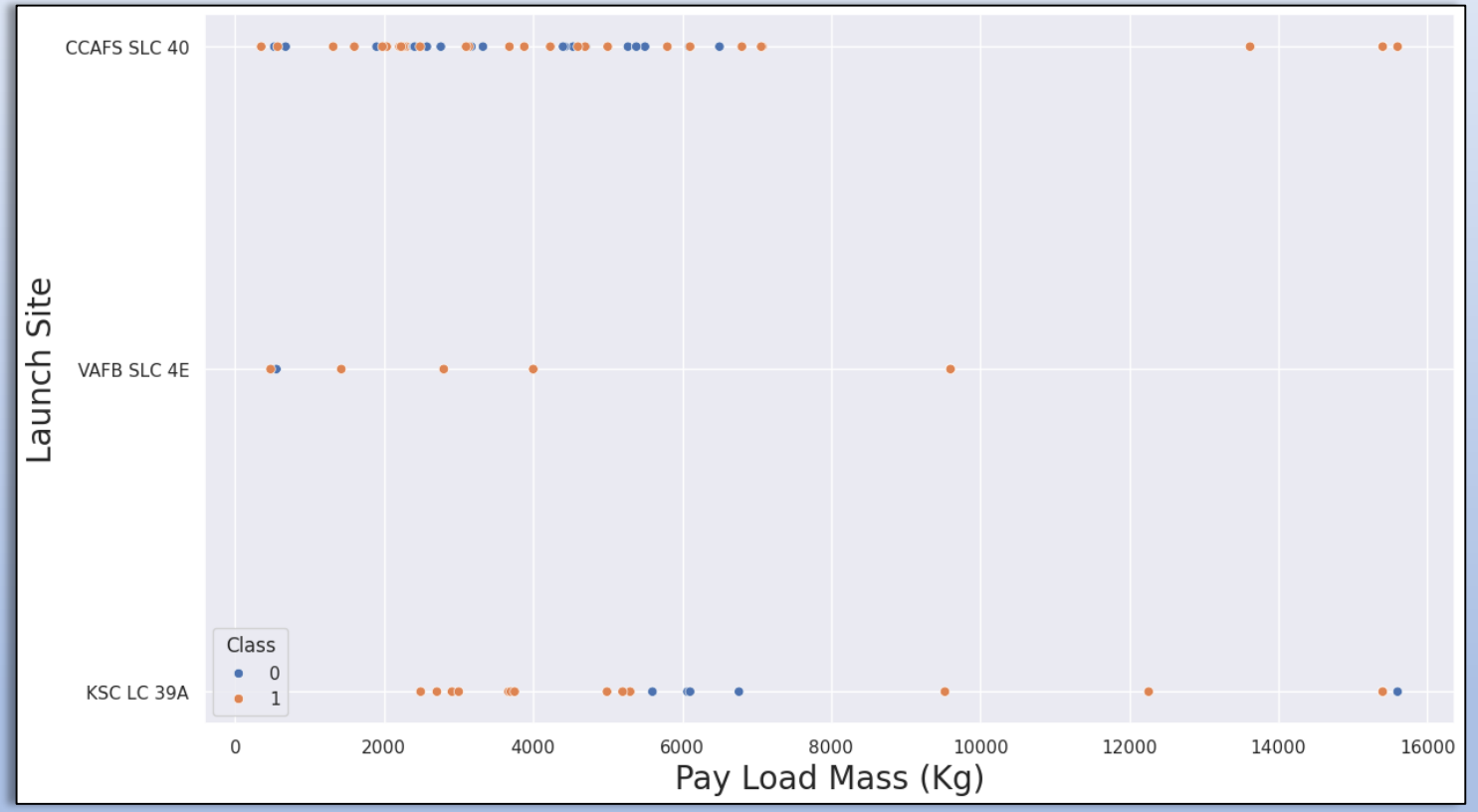
Flight Number vs. Launch Site



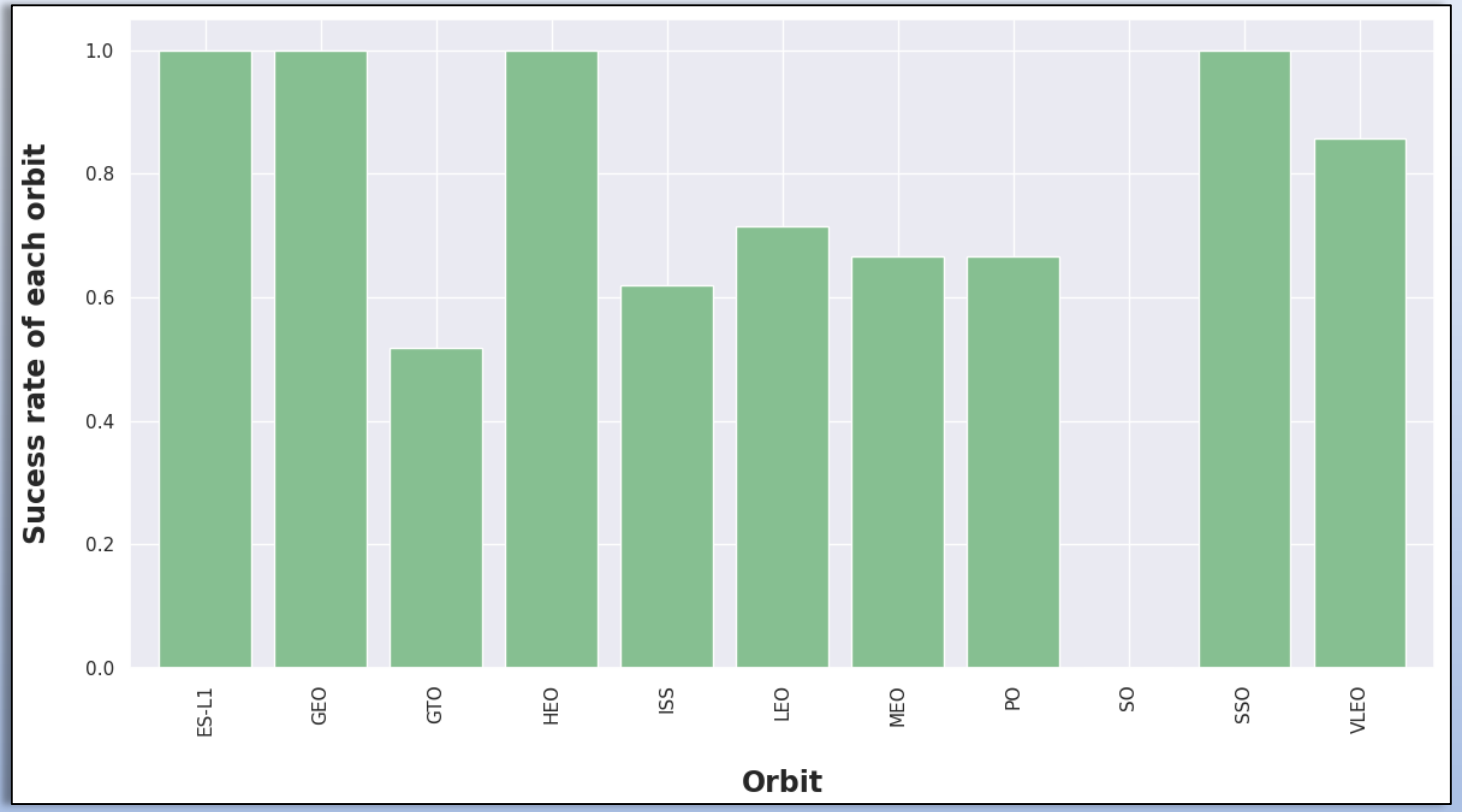
The scatter plot shows that the larger the amount of flights at a launch site the greater is the success rate at that launch site

Payload vs. Launch Site

This plot shows that if the pay load mass is higher as around 7.000kg the success rate the success rate gets very high, while at a lower rate it is more unstable.



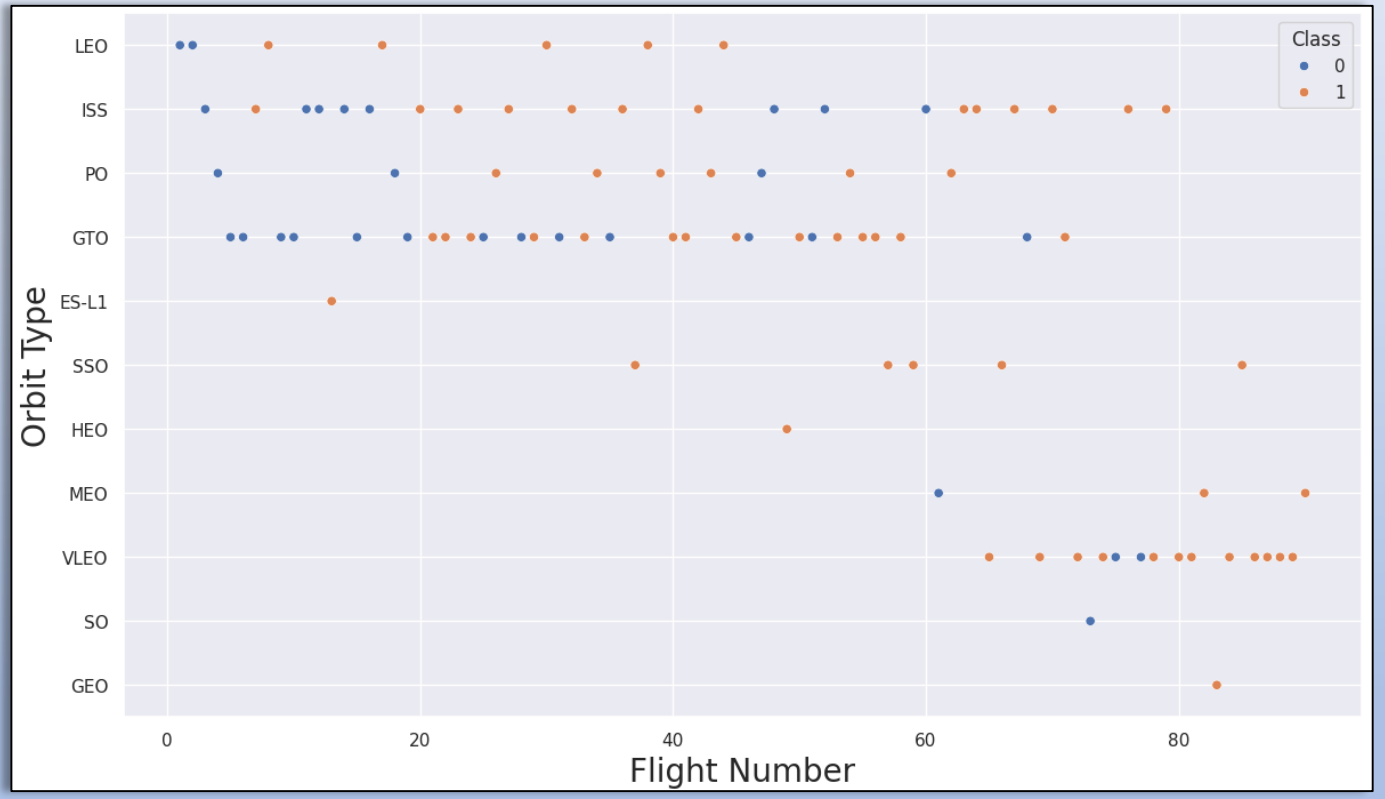
Success Rate vs. Orbit Type



This bar chart describes the success rate of each orbit. While ES-L1, GEO, HEO and SSO have a success rate of 100%, SO, for example, has a success rate of 0%.

Obviously this also heavily depends on the amount of datasets included in the analysis to find more patterns or trends.

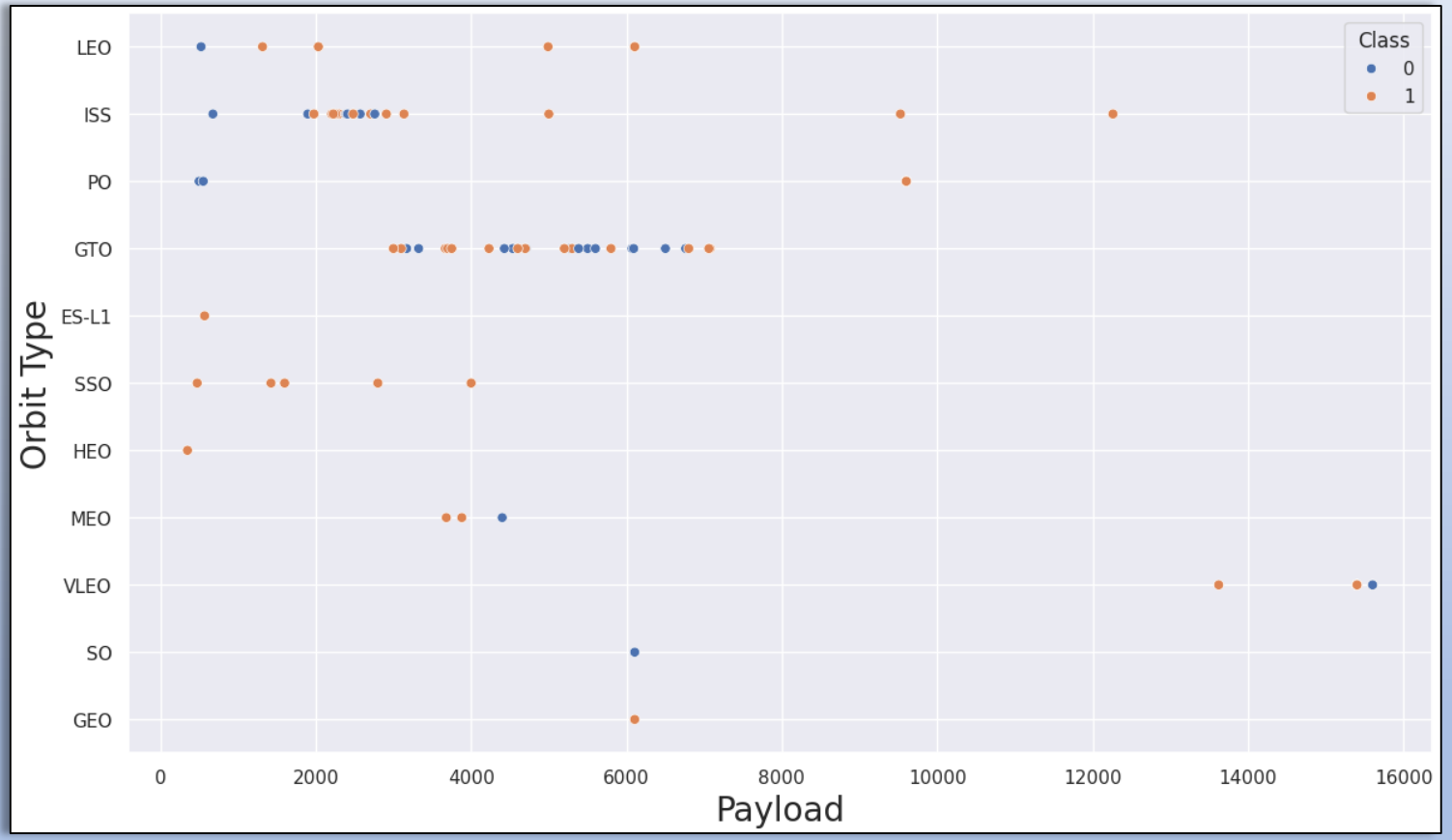
Flight Number vs. Orbit Type



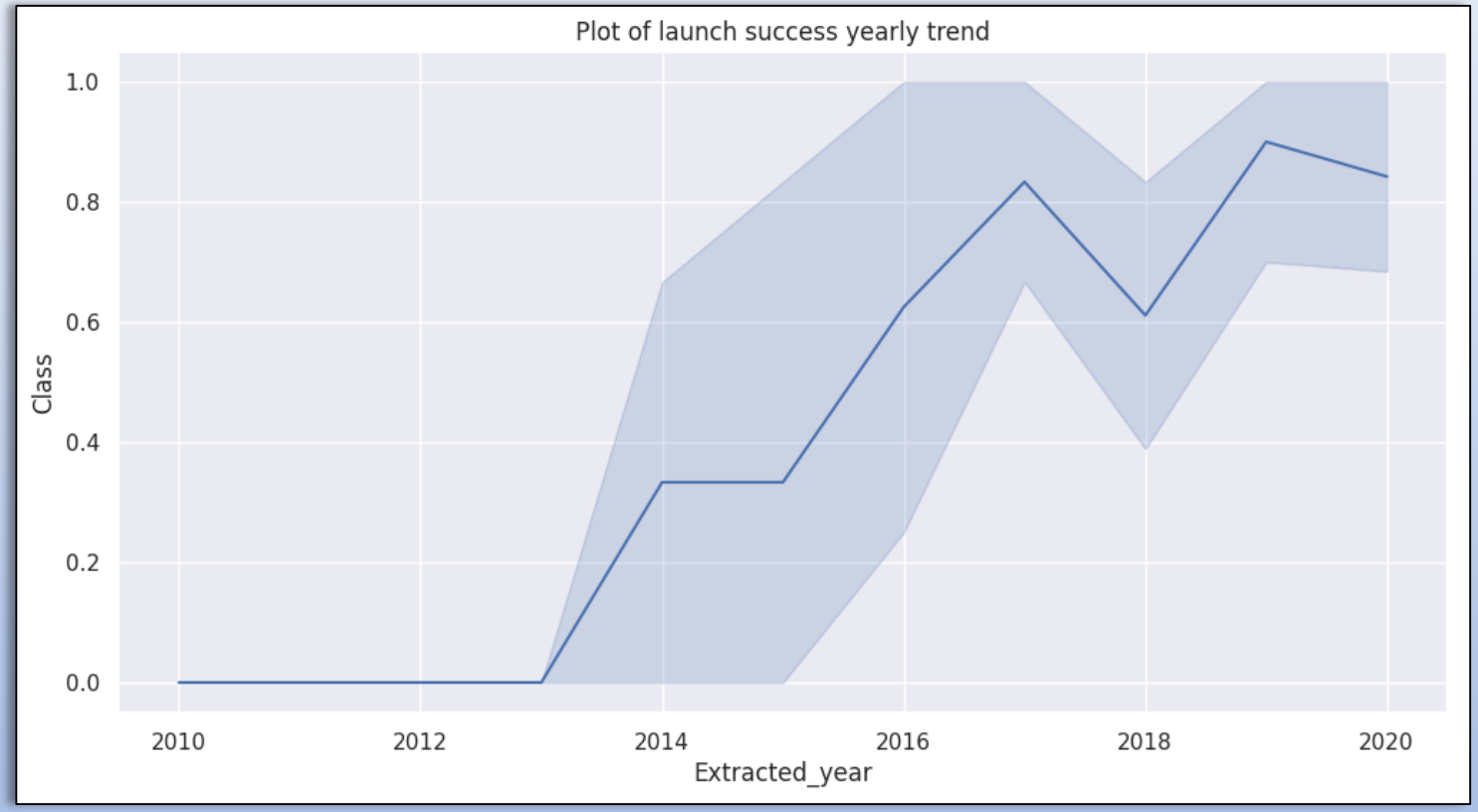
While the GTO orbit shows no relations between the flight number and the flight numbers of the orbit, this plot generally shows a higher success rate when the numbers of flights increases.

Payload vs. Orbit Type

In this plot we can see that heavier payload has a positive impact on the LEO, ISS and PO orbits. While there again seems to be no relation for the GTO orbit between these attributes, the rest the orbits need more data to get any conclusion.



Launch Success Yearly Trend



The success rate in this plot the increasing success rate from year to year since 2010 to 2020.

All Launch Site Names

To find the unique launch site names we used the key word **DISTINCT**:

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

This query shows 5 records where the launch sites starts with CCA using '%' as wildcard character:

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

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

Total Payload Mass

We choose ‘NASA (CRS)’ as customer and sum the payload_mass_kg to get 45.596kg:

```
%sql SELECT SUM (PAYLOAD_MASS__kg_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)' ;
```

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

SUM (PAYLOAD_MASS__kg_)
45596

Average Payload Mass by F9 v1.1

In this example the function AVERAGE (AVG) is used. This one returns the average value of the payload mass by 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

First Successful Ground Landing Date

```
%sql SELECT MIN (DATE) AS "First Successful Ground Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

First Successful Ground Landing
2015-12-22

The first successful ground landing date was on the 22. December in 2015.
To get this return we used MIN-Function and filtered the specific landing outcome.

Successful Drone Ship Landing with Payload between 4000 and 6000

The following list shows every successful drone ship landing while having a payload between 4000 and 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

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "successful mission" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%';
```

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

successful mission
100

```
%sql SELECT COUNT (MISSION_OUTCOME) AS "failure mission " FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Fail%'
```

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

failure mission
1

In total we have 100 successful missions and 1 failure mission as outcomes.

Boosters Carried Maximum Payload

The query on the right returned every unique booster version which carried the maximum payload mass.

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Booster Versions which 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

With the function substr we got the month and also the year for the filtering. By also filtering after LANDING_OUTCOME we got two results for failed launch records in 2015.

```
%sql SELECT substr(Date,4,2) AS month,DATE,BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure (drone ship)' AND substr(Date,0,5) = '2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
5-	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
5-	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

```
%sql SELECT LANDING_OUTCOME as "Landing Outcome", COUNT(LANDING_OUTCOME) AS "Total Count" FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY  LANDING_OUTCOME \
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

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

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

Between 04.06.2020 and 20.03.2017 we had 10 counts of zero landing outcomes, while we had 5 successful and 5 unsuccessful outcomes of drone ship landings.

Section 3

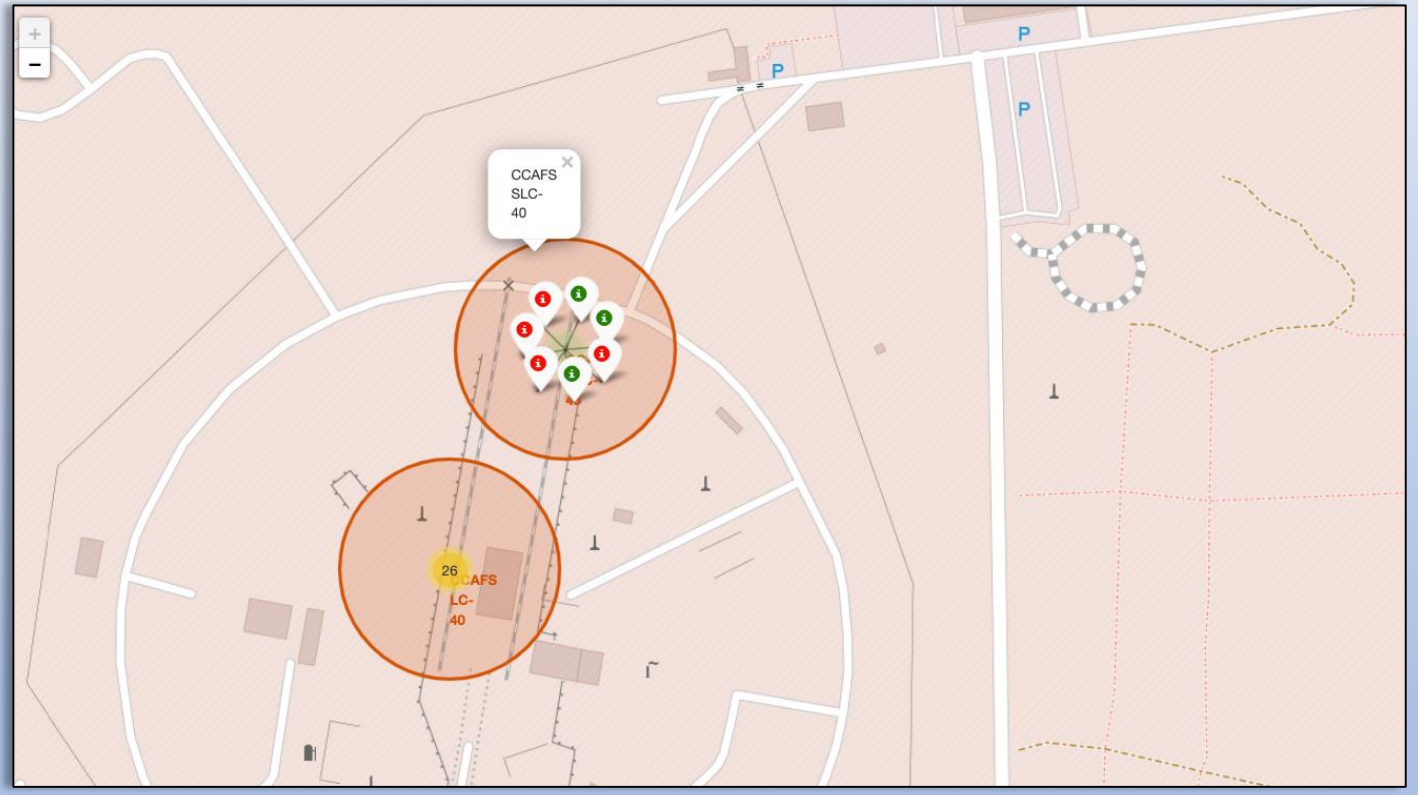
Launch Sites Proximities Analysis

All launch sites markers on a global map

The SpaceX launch sites are in the United States of America, in Florida and California.

Both are located at coasts.

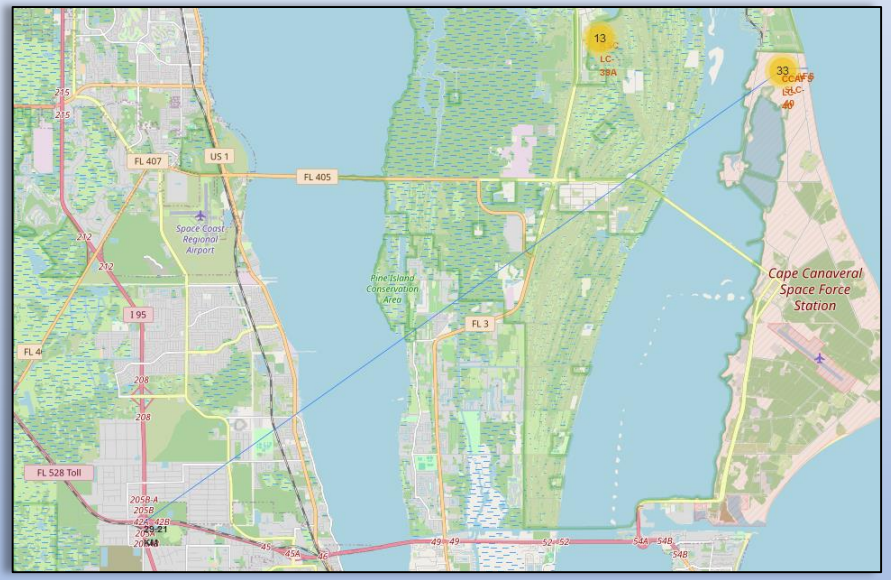
Successors vs failures in color

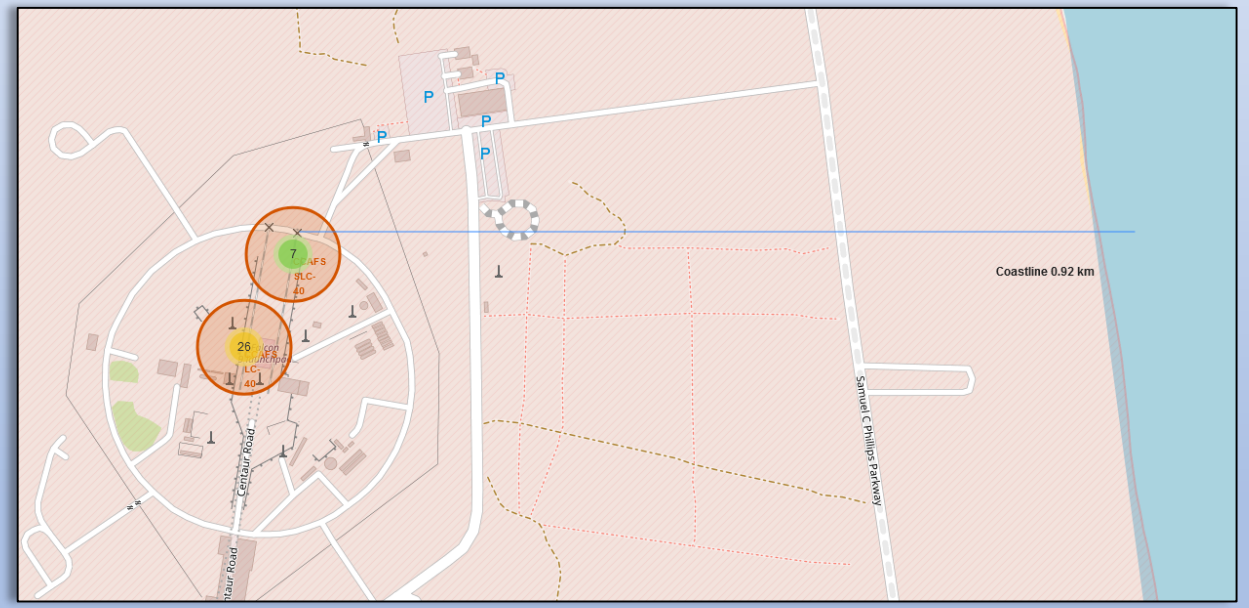


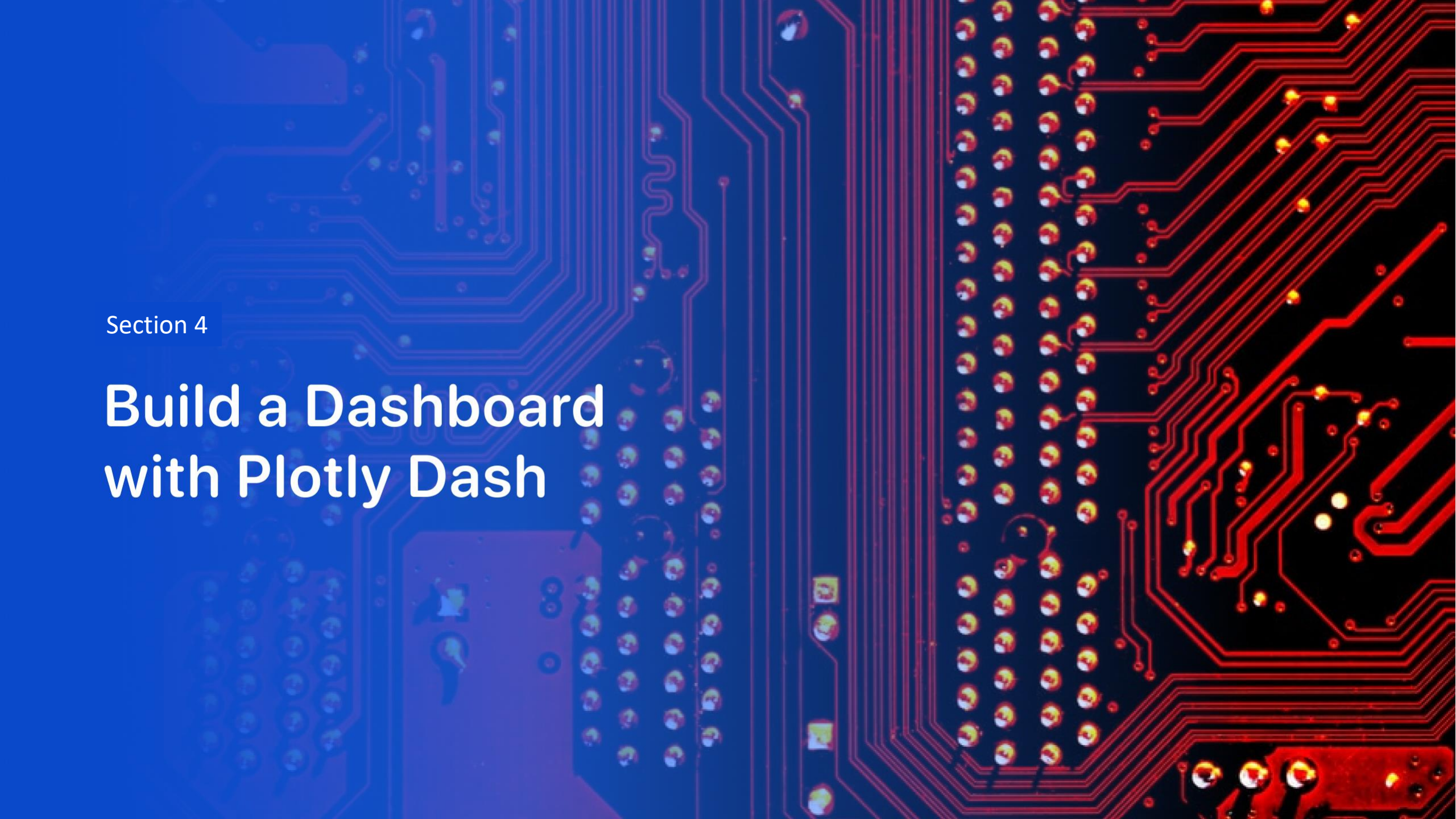
Green markers show the successful launches, while the red ones show every failure.

Calculating distances to its proximities

Launch sites, like in the following examples of the CCAFS-SLC-40 shown, are not in proximity of cities but for sure in proximity of coastlines, highways and railways.

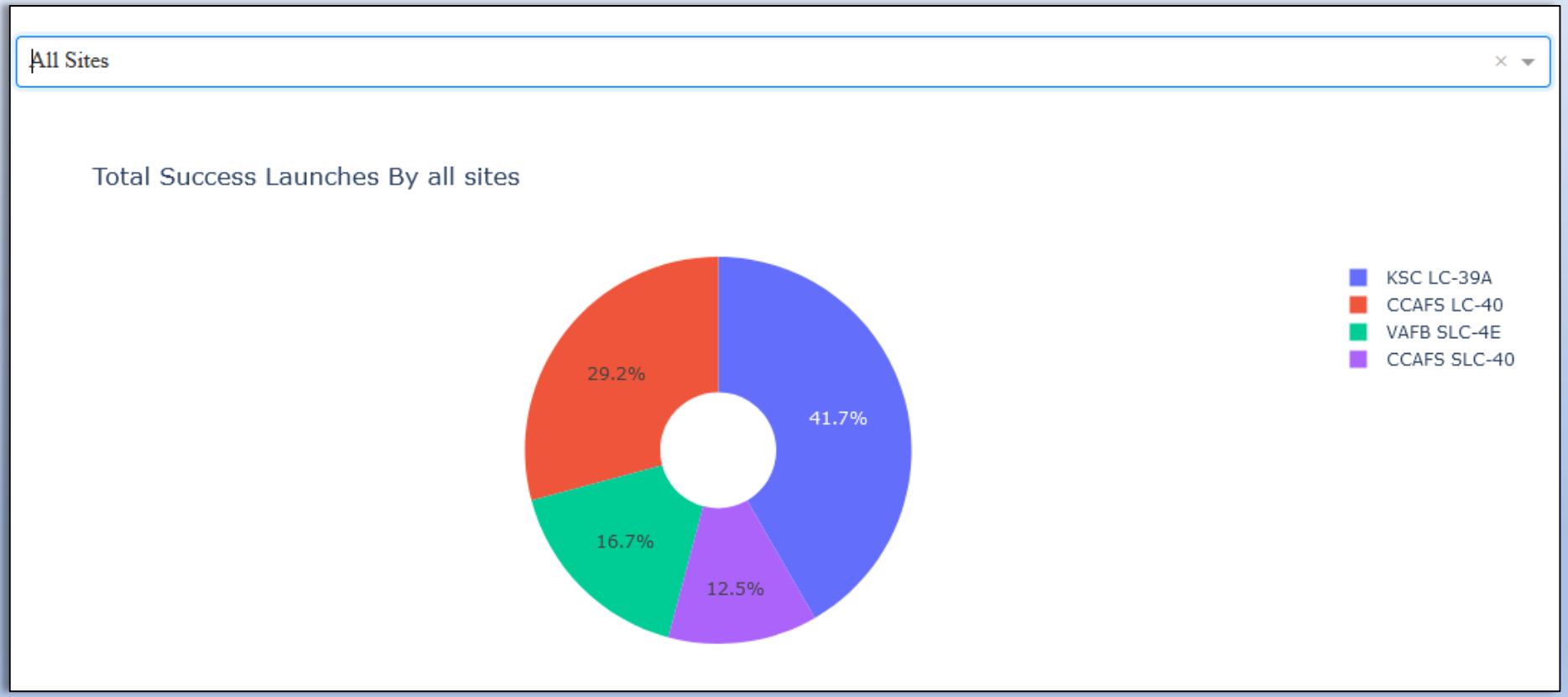





Section 4
Build a Dashboard with Plotly Dash

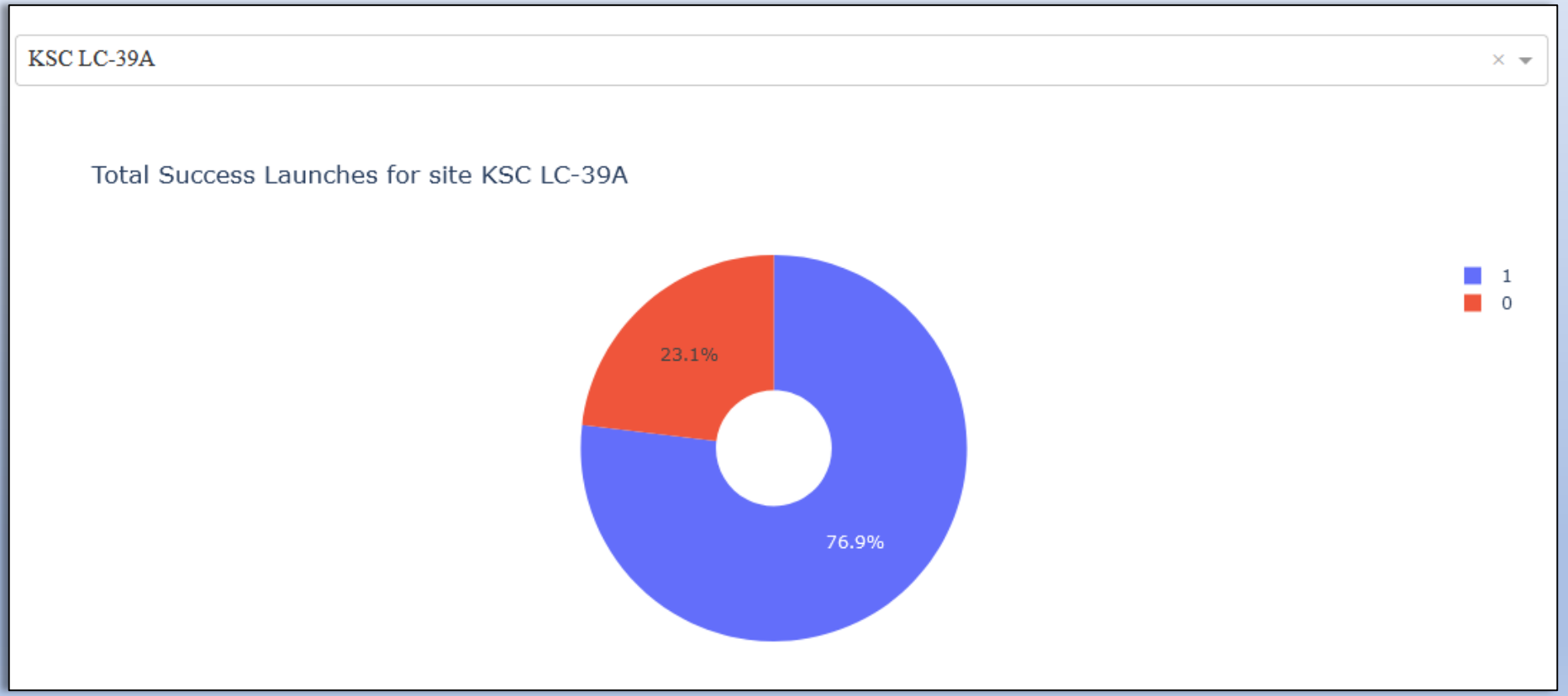
Comparing the total success launches by all sites

The following pie chart shows that the KSC LC-39A launch site had the most successful launches from all the sites.

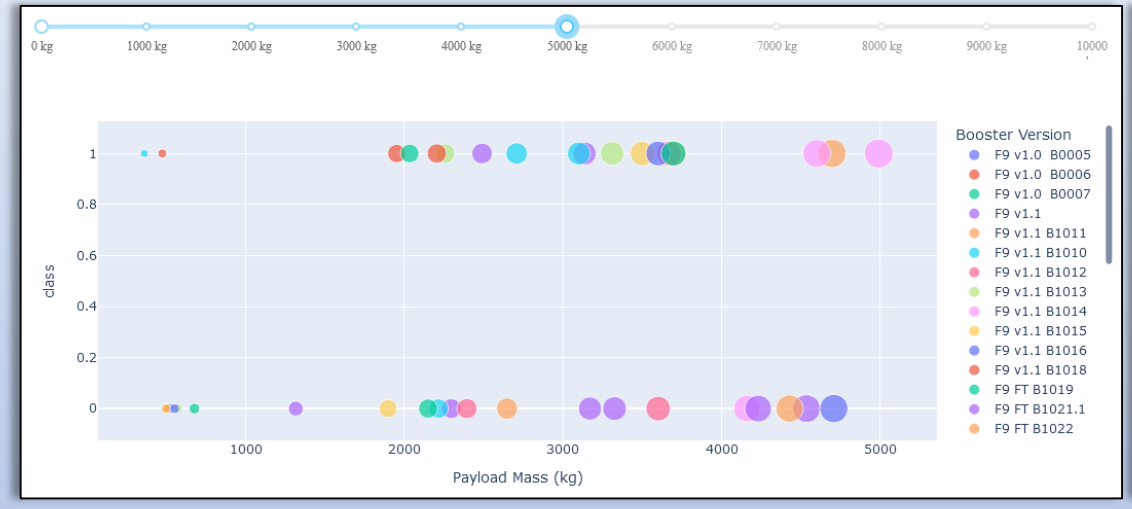


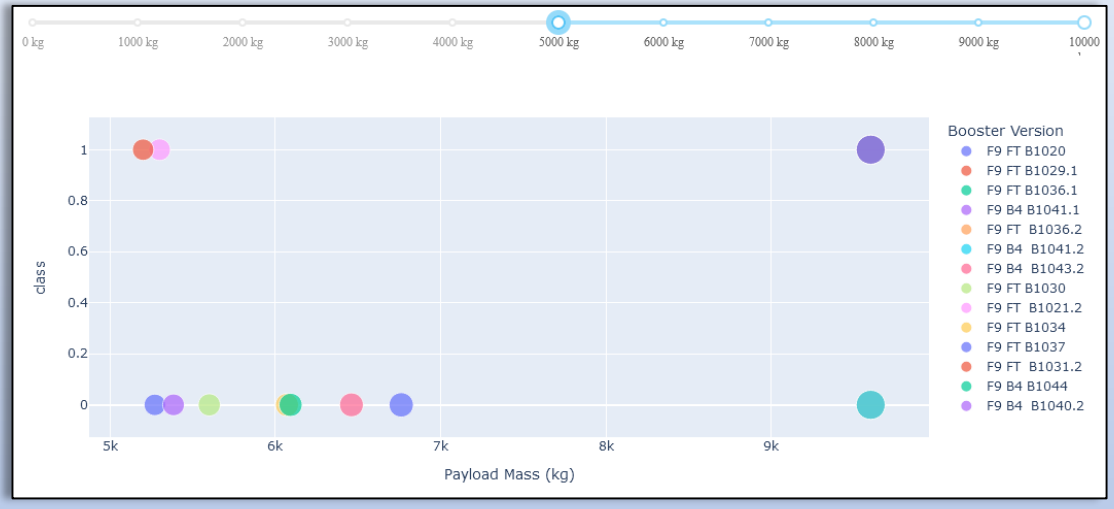
Highest launch success launch site

The KSC LC-39A launch site achieved a success rate of 76.9% while getting a 23.1 failure rate.



Payload Mass VS Launch Outcome



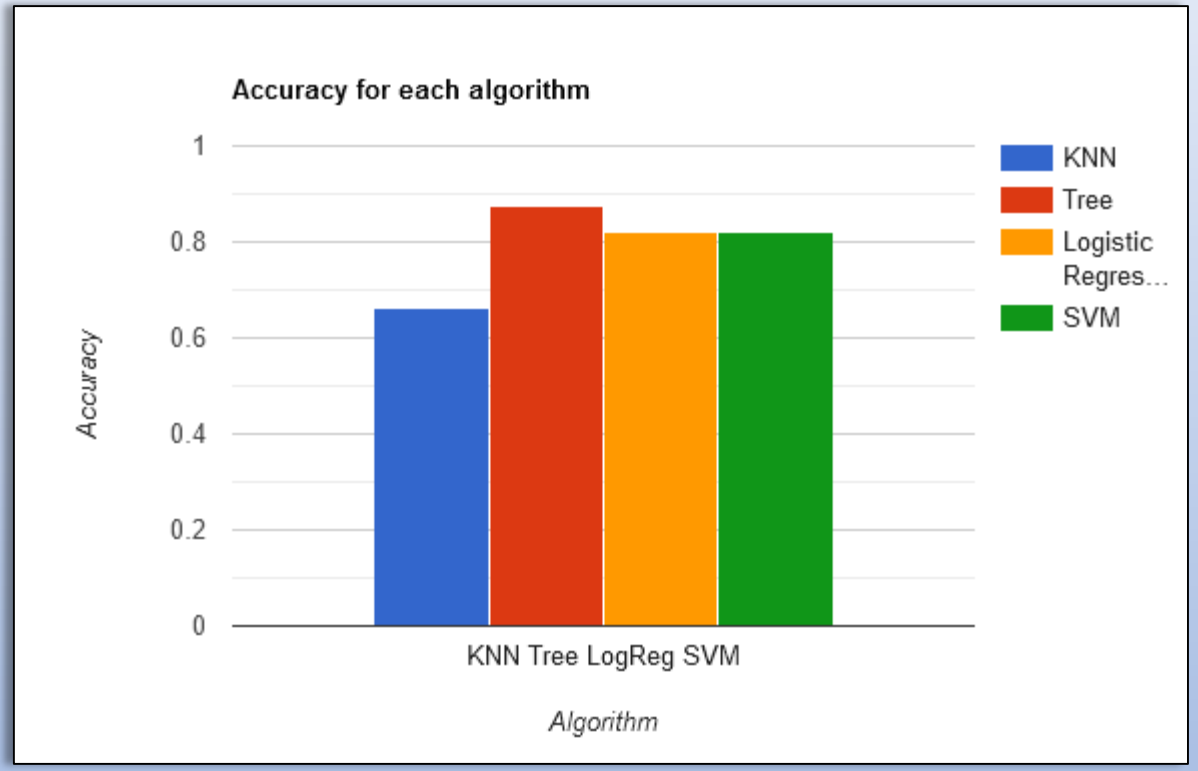


The two scatter plots show low weighted payload from 0kg to 5.000kg on the left and heavy weighted payload from 5.000kg to 10.000kg on the right.
As seen in these plots, the success rates for low weighted payloads is quite higher than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



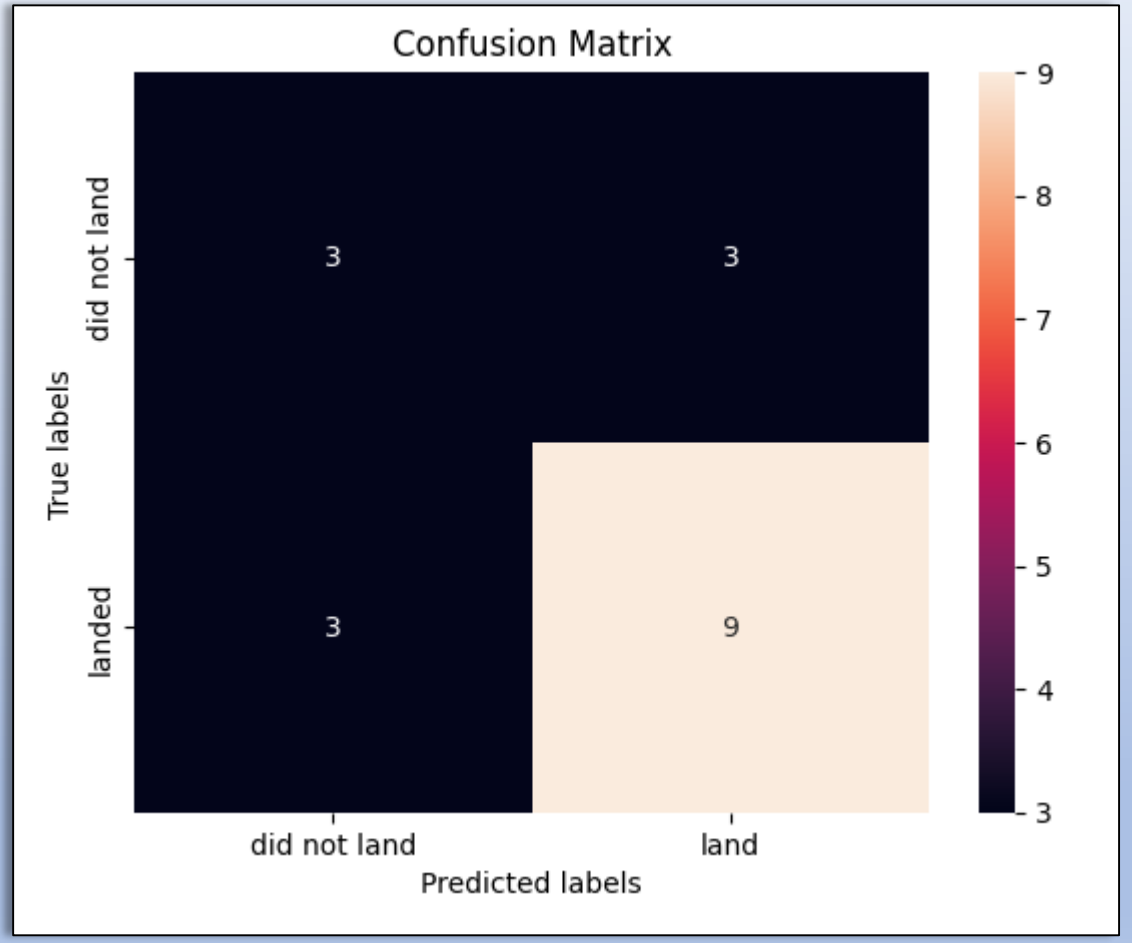
The bar chart on the left shows the accuracies for all four algorithms.

The tree one has the highest classification accuracy with 87,68%.

Confusion Matrix

The matrix shows predicted and true labels regarding labels.

The main problem are the false positives, meaning unsuccessful landings predicted as successful landings by the classifier.



Conclusions

In conclusion, we can observe the following:

- A higher number of flights at a launch site correlates with a higher success rate.
- The launch success rate saw a steady increase from 2013 to 2020.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.
- KSC LC-39A recorded the most successful launches among all sites.
- The Decision Tree classifier proved to be the most effective machine learning algorithm for this analysis.

Thank you!