IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Lijun>
<2 Sep 2024>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

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. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

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

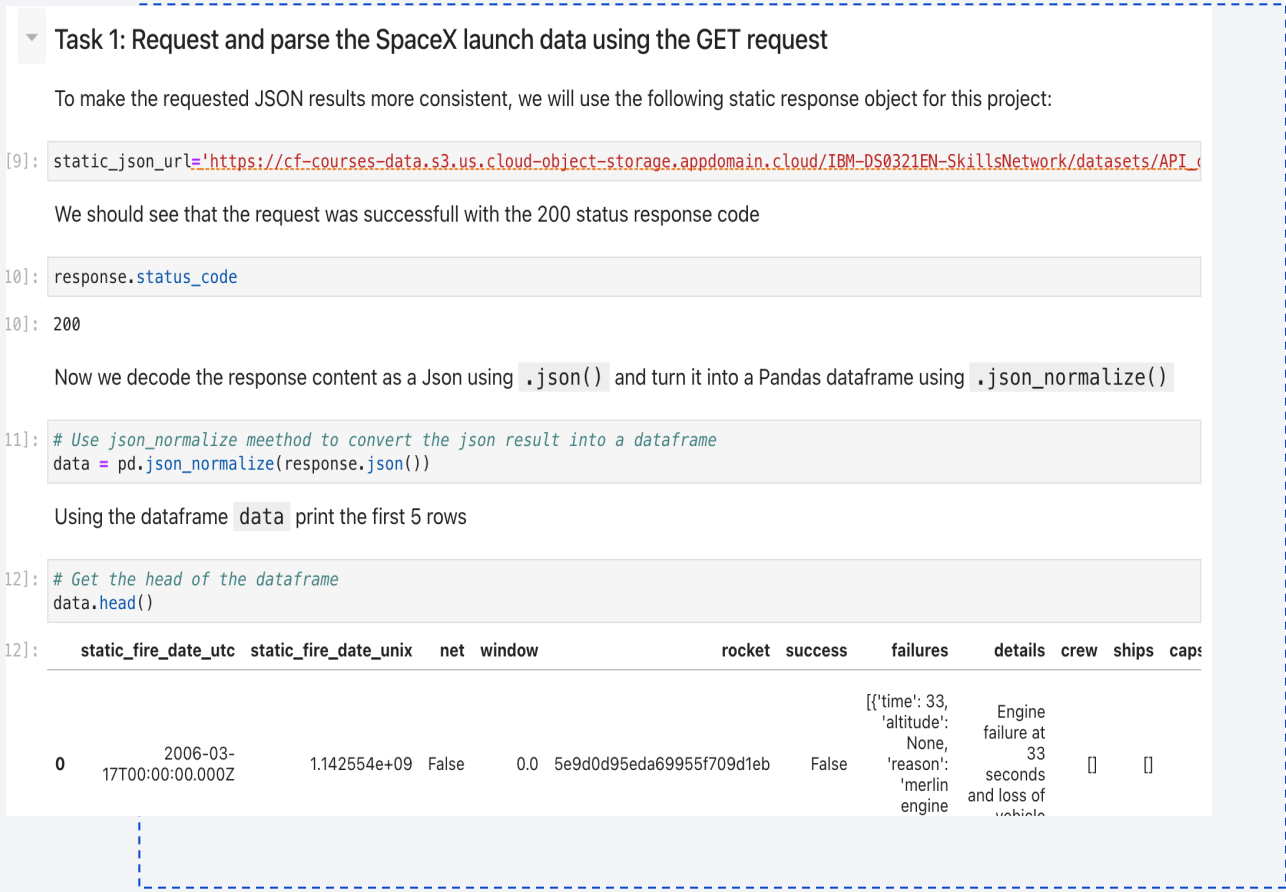
Data Collection

- Describe how data sets were collected.
  - Data collection was done using get request to the SpaceX API.
  - Decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - Then cleaned the data, checked for missing values and fill in missing values where necessary.
  - Performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Collect data using the SpaceX API. Make a get request to the SpaceX API to request and parse the SpaceX launch data using a GET request and decode the response content into a Json result and then convert it into a Pandas dataframe.

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/1.%20SpaceX%20Data%20Collection%20API.ipynb



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

We should see that the request was successfull with the 200 status response code

```
10]: response.status_code

10]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Using the dataframe `data` print the first 5 rows

```
12]: # Get the head of the dataframe
data.head()
```

12]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	caps
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine	Engine failure at 33 seconds and loss of	[]	[]	

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

[6]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

[7]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

[8]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches – Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

[9]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
```

- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into a pandas dataframe.

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/2.%20SpaceX%20Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.ipynb

Data Collection - Scraping

Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches at each site, and the number and occurrence of each orbits
- Created landing outcome label from outcome column and exported the results to csv.

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/3.%20SpaceX%20Data%20Wrangling%20spacex.ipynb

TASK 1: Calculate the number of launches on each site

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 VAFB SLC 4E , Vandenberg Air Force Base Space Launch Complex 4E (SLC-4E), Kennedy Space Center Launch Complex 39A KSC LC 39A .The location of each Launch Is placed in the column `LaunchSite`

Next, let's see the number of launches for each site.

Use the method `value_counts()` on the column `LaunchSite` to determine the number of launches on each site:

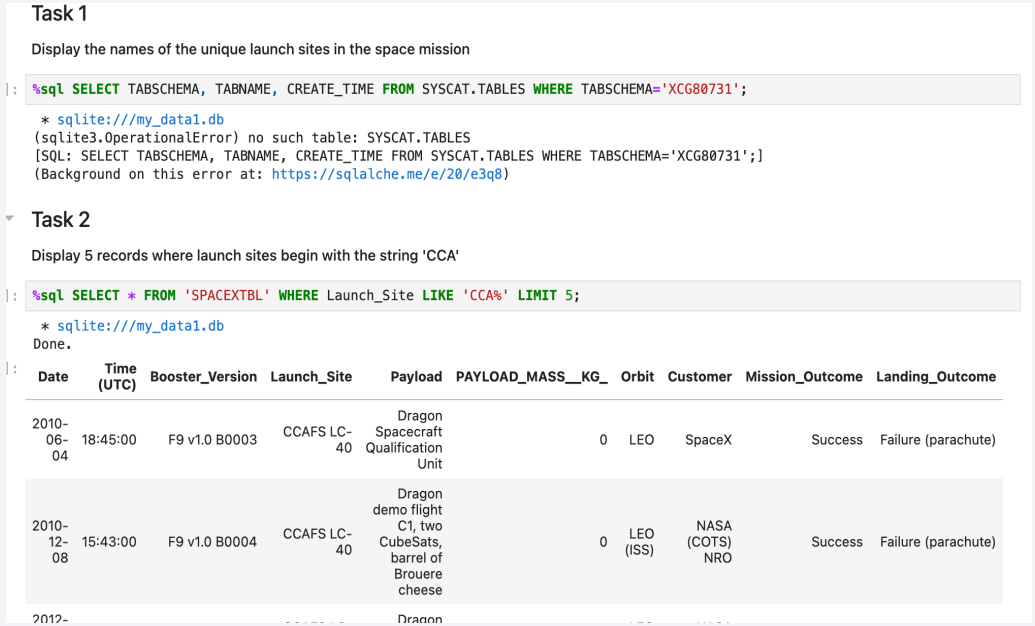
[5]:
```
# Apply value_counts() on column LaunchSite
df.LaunchSite.value_counts()
```

[5]:
```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Each launch aims to an dedicated orbit, and here are some common orbit types:

- LEO: Low Earth orbit (LEO)is an Earth-centred orbit with an altitude of 2,000 km (1,200 mi) or less (approximately one-third of the radius of Earth),[1] or with at least 11.25 periods per day (an orbital period of 128 minutes or less) and an eccentricity less than 0.25.[2] Most of the manmade objects in outer space are in LEO [1].
- VLEO: Very Low Earth Orbits (VLEO) can be defined as the orbits with a mean altitude below 450 km. Operating in these orbits can provide a number of benefits to Earth observation spacecraft as the spacecraft operates closer to the observation[2].
- GTO A geosynchronous orbit is a high Earth orbit that allows satellites to match Earth's rotation. Located at 22,236 miles (35,786 kilometers) above Earth's equator, this position is a valuable spot for monitoring weather, communications and

EDA with Data Visualization



Task 1

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

```
%sql SELECT TABSCHEMA, TABNAME, CREATE_TIME FROM SYSCAT.TABLES WHERE TABSCHEMA='XCG80731';
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such table: SYSCAT.TABLES
[SQL: SELECT TABSCHEMA, TABNAME, CREATE_TIME FROM SYSCAT.TABLES WHERE TABSCHEMA='XCG80731';]
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

Task 2

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

```
%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-				Dragon					

- Performed data Analysis and Feature Engineering using Pandas.
- Used scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, Payload and Orbit type.
- Used Bar chart to Visualize the relationship between success rate of each orbit type
- Line plot to Visualize the launch success yearly trend.

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/5.%20SpaceX%20EDA%20DataViz%20Using%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/4.%20SpaceX%20EDA%20Using%20SQL.ipynb

Build an Interactive Map with Folium

- To find geographical patterns in the data the following items were marked on a map of launch sites:
  - All Launch Sites
  - Successful and Failed Launches
  - Distances between a launch site and proximate landmarks

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/6.%20SpaceX%20Launch%20Sites%20Locations%20Analysis%20with%20Folium-Interactive%20Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard application with Plotly dash by:
  - Adding a Launch Site Drop-down Input Component
  - Adding a callback function to render success-pie-chart based on selected site dropdown
  - Adding a Range Slider to Select Payload
  - Adding a callback function to render the success-payload-scatter-chart scatter plot

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/7.%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash%20-%20spacex_dash_app.py

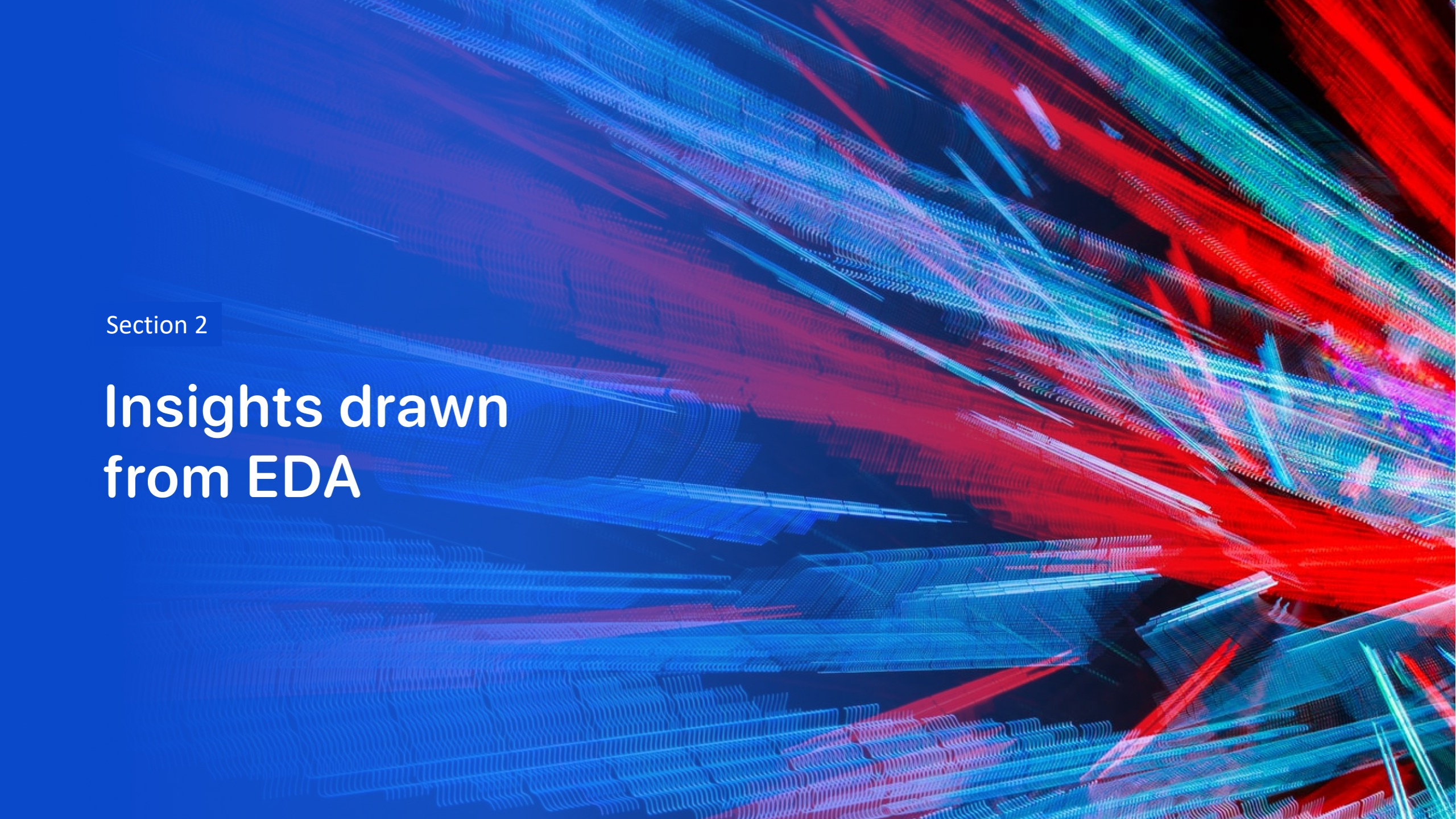
Predictive Analysis (Classification)

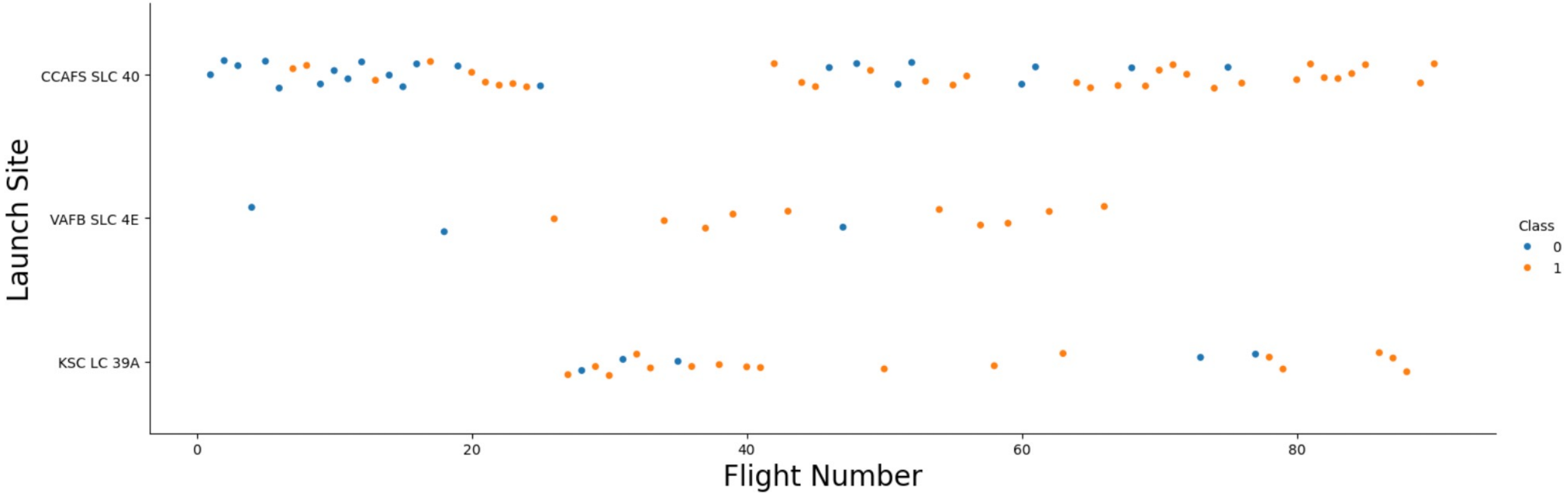
- After loading the data as a Pandas Data frame, I set out to perform exploratory Data Analysis and determine Training Labels by.
- In order to find the best ML model/ method that would performs best using the test data between SVM, Classification Trees, k nearest neighbours and Logistic Regression.
- The table below shows the test data accuracy score for each of the methods comparing them to show which performed best using the test data between SVM, Classification Trees, k nearest neighbours and Logistic Regression.

- GitHub URL: https://github.com/Lizzie-29/Final-Project-SpaceX-/blob/main/8.%20SpaceX%20Machine%20Learning%20Prediction.ipynb

Results

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


Section 2
Insights drawn from EDA

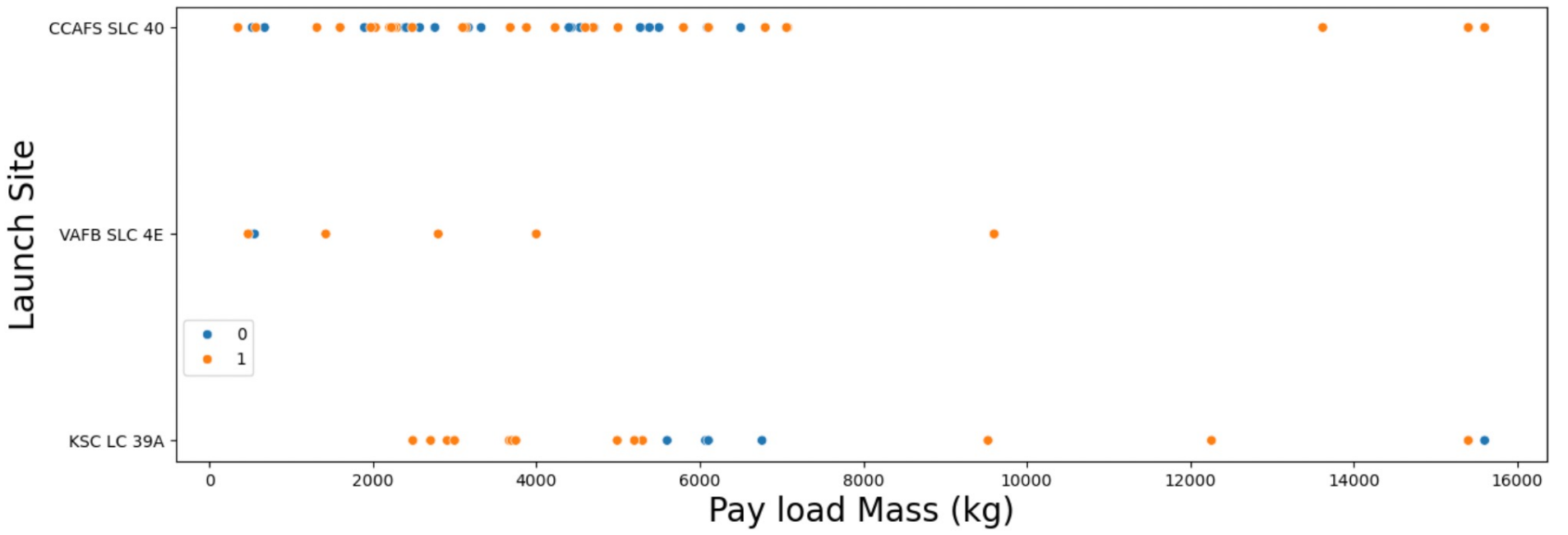


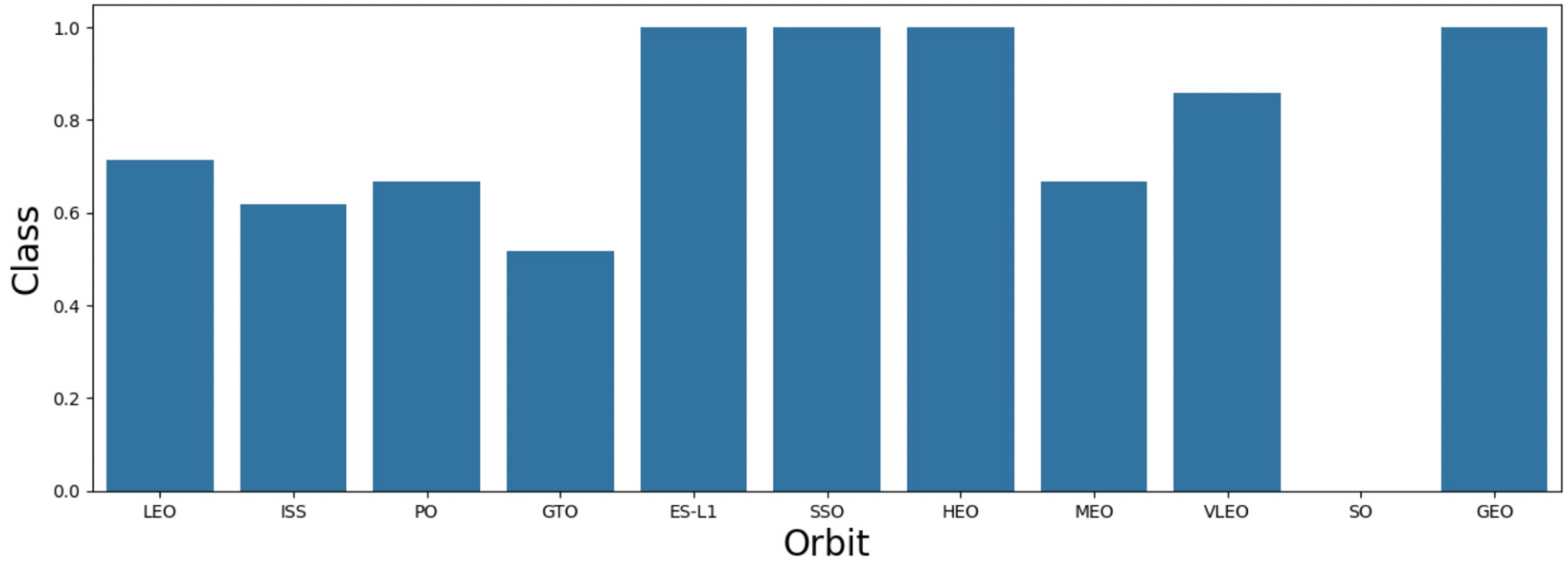
Flight Number vs. Launch Site

- All sites show a mix of first stage landing successes and failures, with successes increasing overtime.
- Early flights predominantly resulted in failures, indicating improvements to technology or process.
- While CCAFS SLC 40 has the most total flights, VAFB SLC 4E appears to have a relatively higher proportion of successful landing outcomes.

Payload vs. Launch Site

- All sites show a wide range of payload weights, from light to heavy payloads.
- Early flights trend toward lighter payloads, representing the bulk of the landing failures.
- This suggests technological or operational improvements lead to a greater rate of success with heavier payloads.

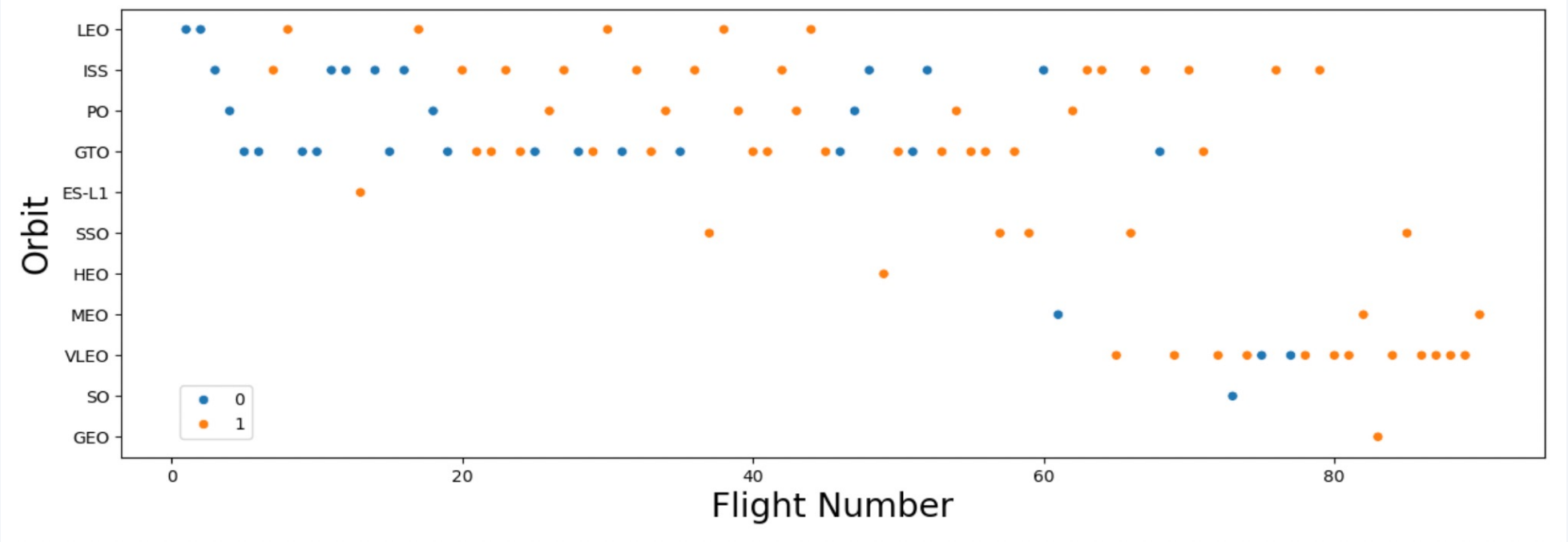




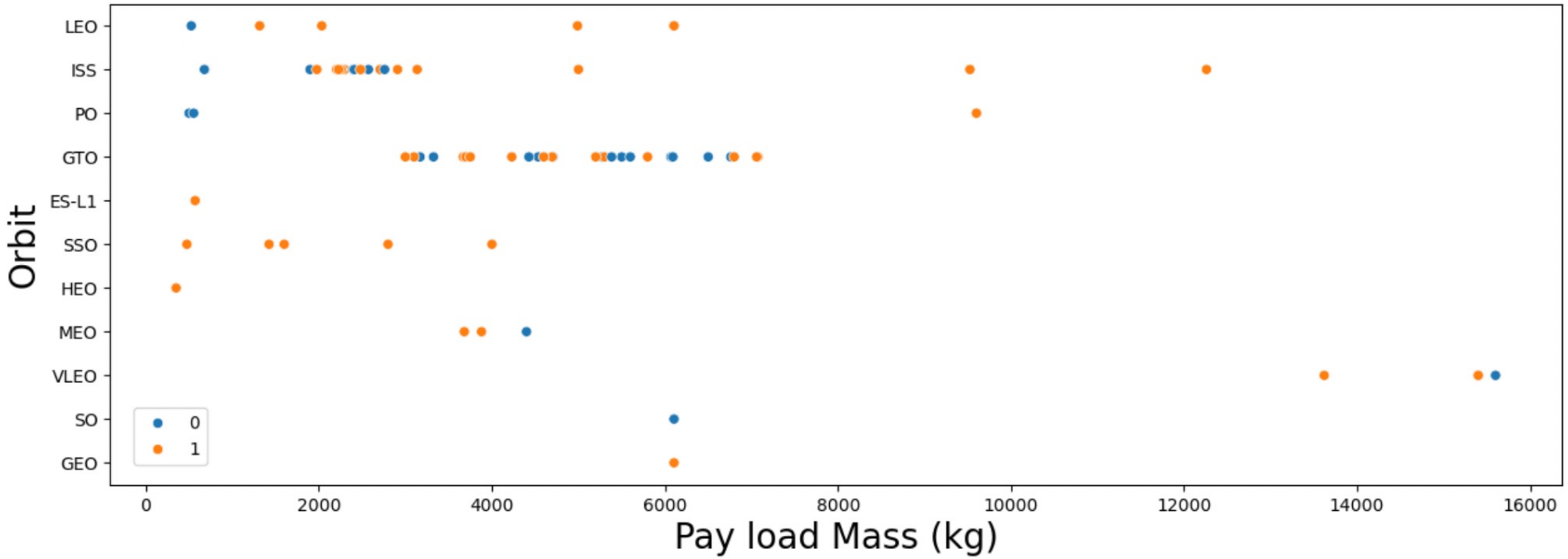
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- Others such as GTO show more mixed outcomes, suggesting some orbit types may introduce operational or technological challenges.

Flight Number vs. Orbit Type



- The plot shows the Flight Number vs Orbit type. In the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



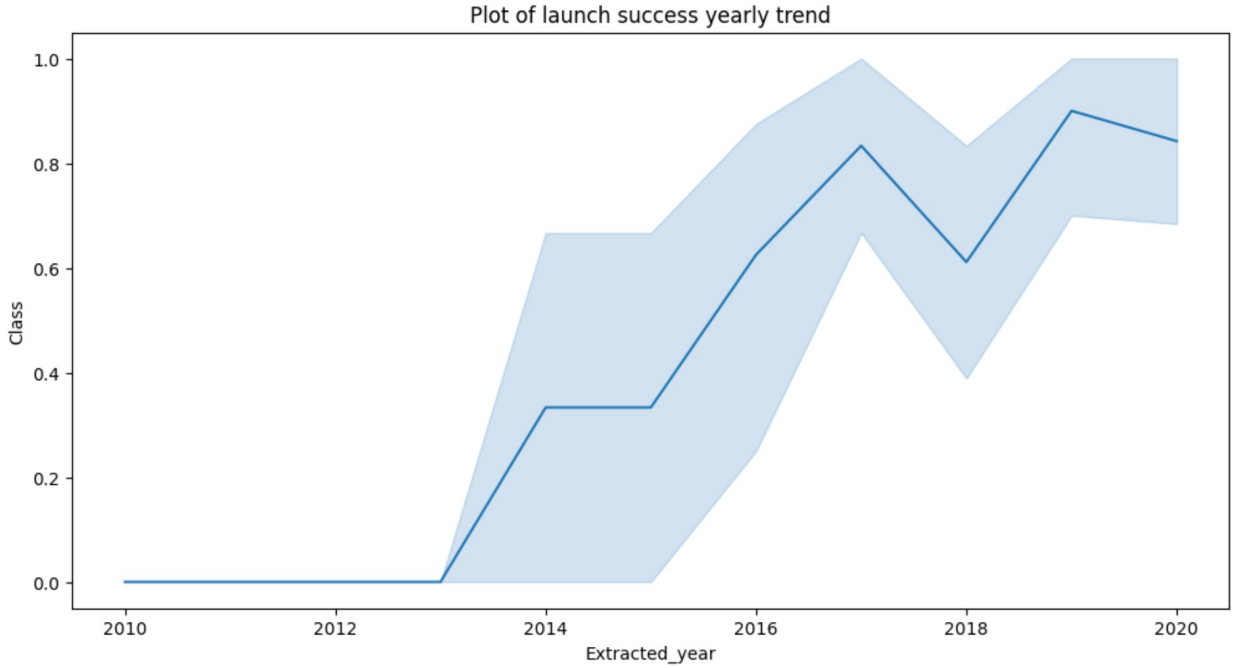
- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.
- For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) both have near equal chances.

Payload vs. Orbit Type

Launch Success Yearly Trend

- From 2016 onward, SpaceX experienced year over year improvement in success rate with a minor setback in 2018.

- Since 2013, the success rate kept going up till 2020.



Task 1

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

```
%%sql

SELECT DISTINCT Launch_Site from SPACEXTABLE;
```

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

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

All Launch Site Names

There are four unique Launch Sites

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

Task 2

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

```
%%sql

SELECT * FROM SPACEXTABLE 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

- The total payload carried by boosters from NASA (CRS) is 45,596kg.

Task 3

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

```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

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

TOTAL_PAYLOAD
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2,534.67kg.

Task 4

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

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

AVG_PAYLOAD_MASS
2534.6666666666665

First Successful Ground Landing Date

- The first successful landing outcome on ground pad occurred on December 22nd, 2015.

Task 5

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

Hint:Use min function

```
%%sql

SELECT MIN(Date) as LaunchDate FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

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

LaunchDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

Task 6

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, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%%sql

SELECT CASE
    WHEN Mission_Outcome LIKE 'Success%' THEN 'Success'
    WHEN Mission_Outcome LIKE 'Failure%' THEN 'Failure'
END as Mission_Status,
COUNT(*)
FROM SPACEXTABLE
GROUP BY Mission_Status;
```

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

Mission_Status	COUNT(*)
Failure	1
Success	100

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

```
%%sql

SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_
    FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
    ORDER BY Booster_Version;
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

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

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%%sql

SELECT
    CASE strftime('%m', Date)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END as Month,
    Landing_Outcome, Booster_Version, Launch_Site, Date
FROM SPACEXTABLE
WHERE strftime('%Y', Date) = '2015' AND Landing_Outcome = 'Failure (drone ship)';
```

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

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

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

Task 10

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(*) as Count
    FROM SPACEXTABLE
        WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY Landing_Outcome
        ORDER BY Count DESC;
```

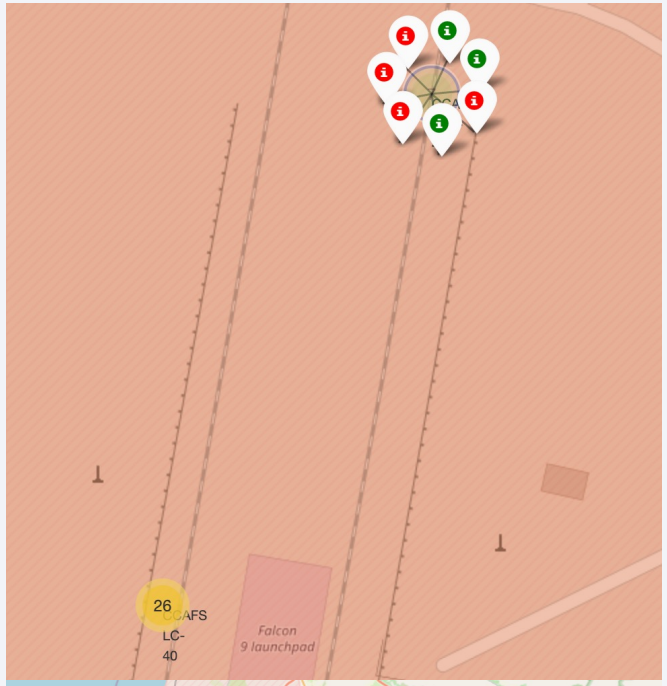
```
 * sqlite:///my_data1.db
Done.
```

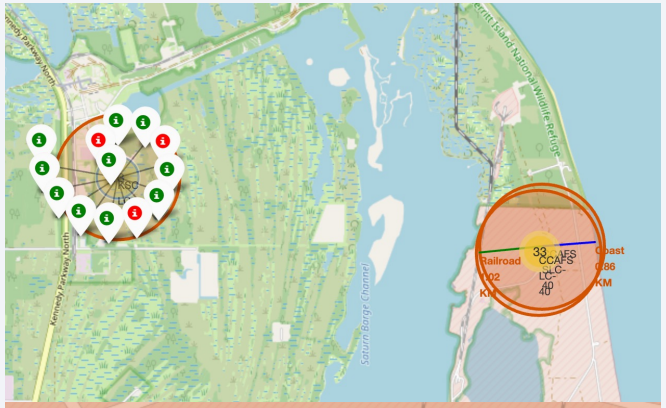
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

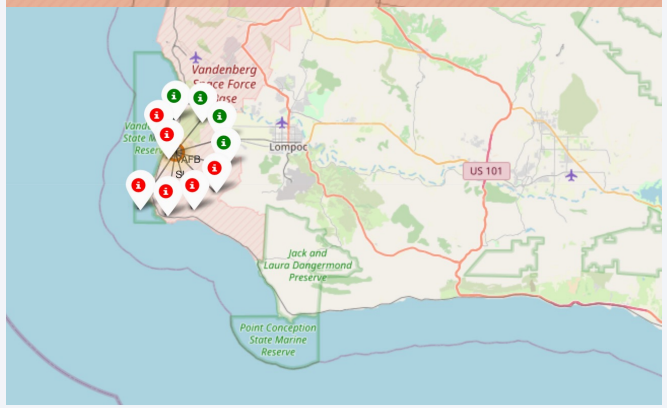

Section 3
Launch Sites
Proximities Analysis

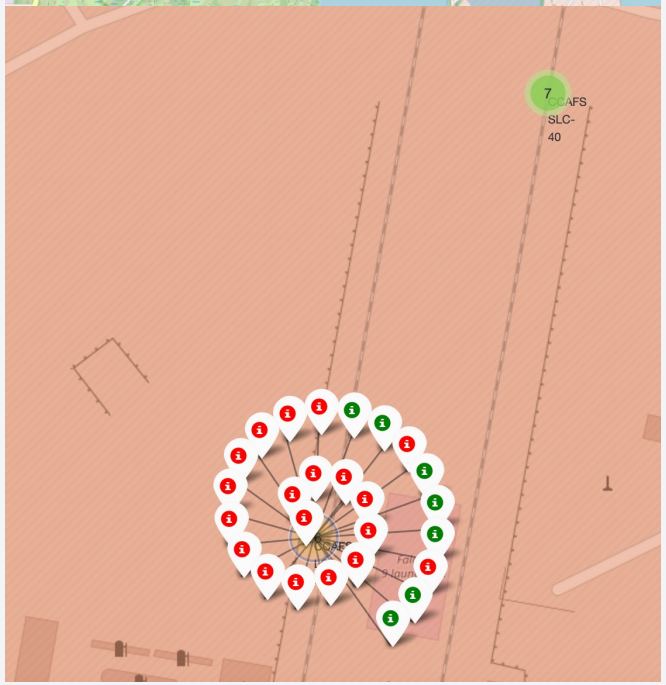
Launch site locations

- Launch sites are located near coastal regions in Florida and California to reduce risk of catastrophic failures affecting human activities.



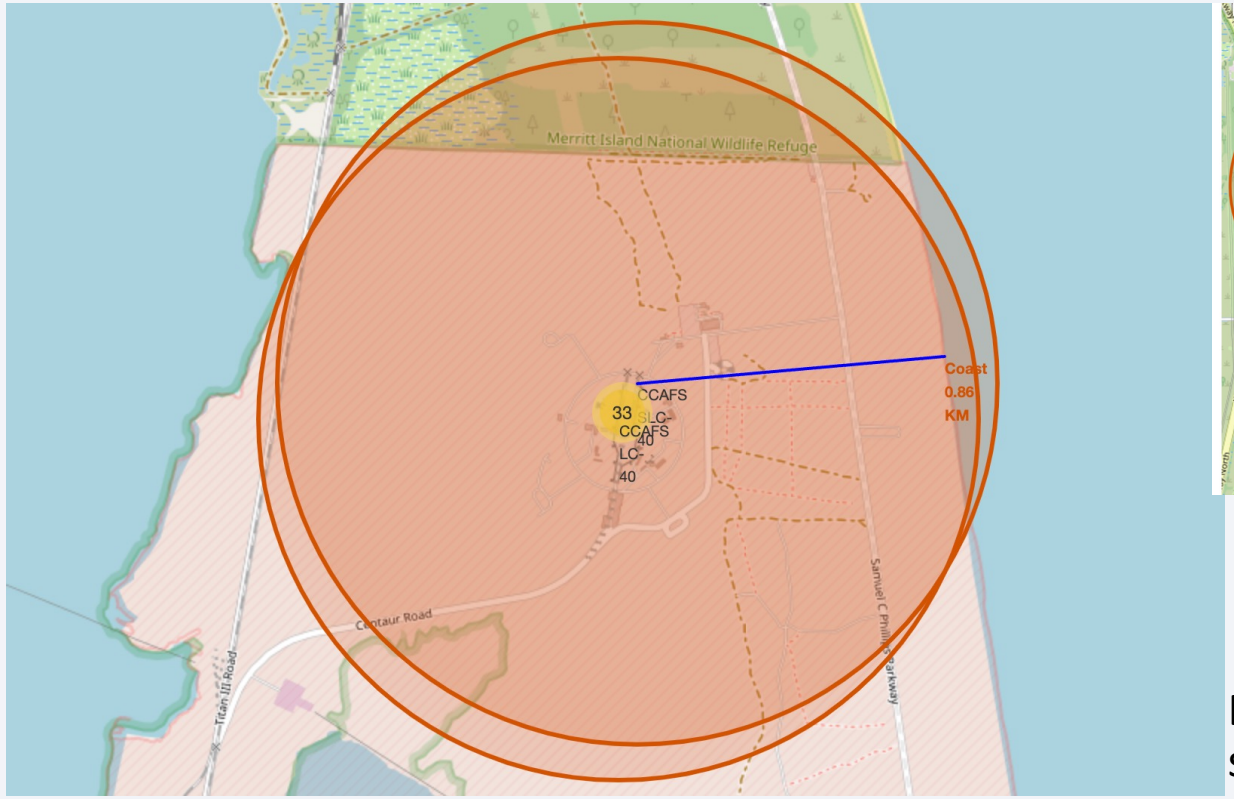




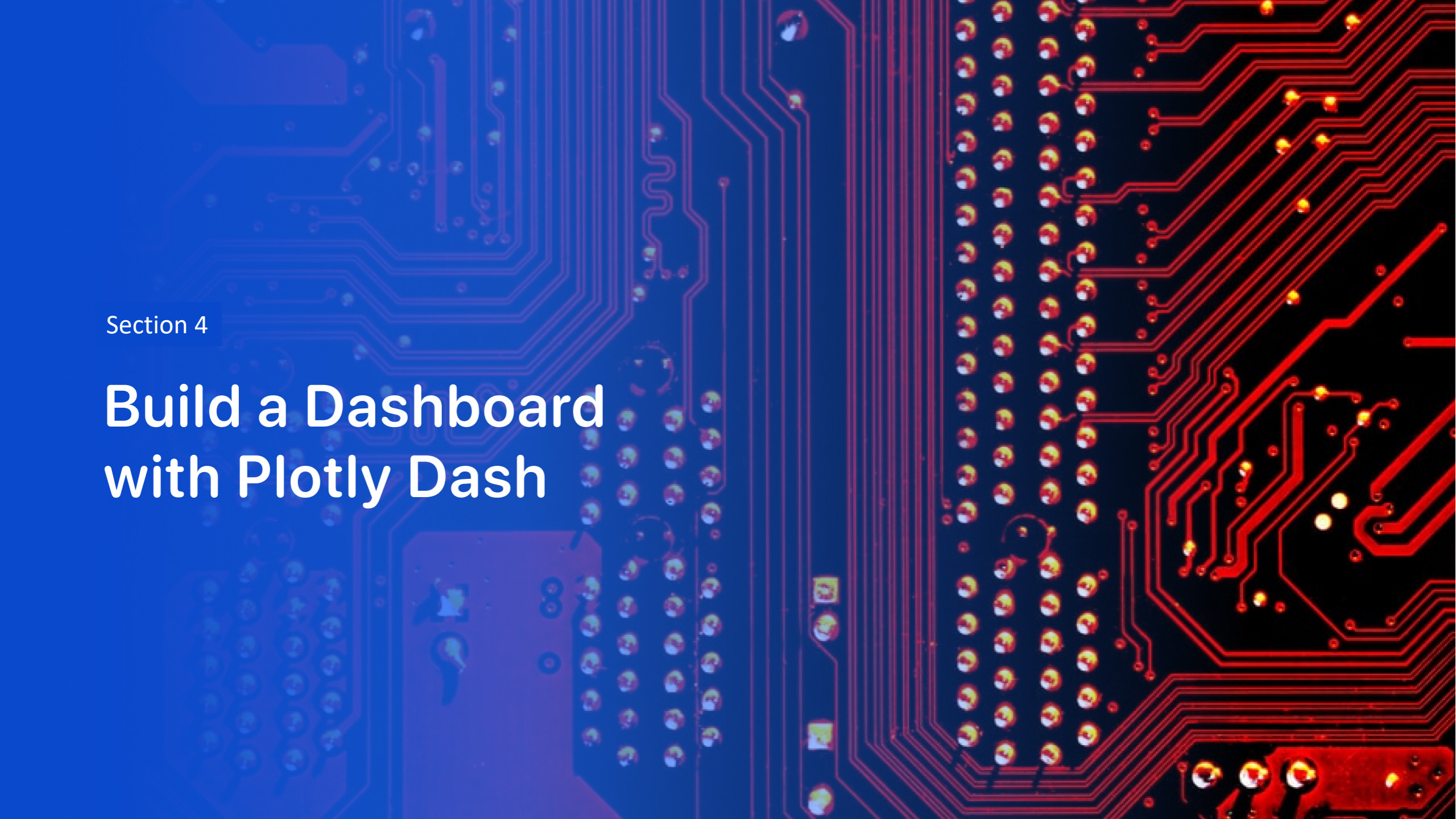


Launch Site distance to landmarks

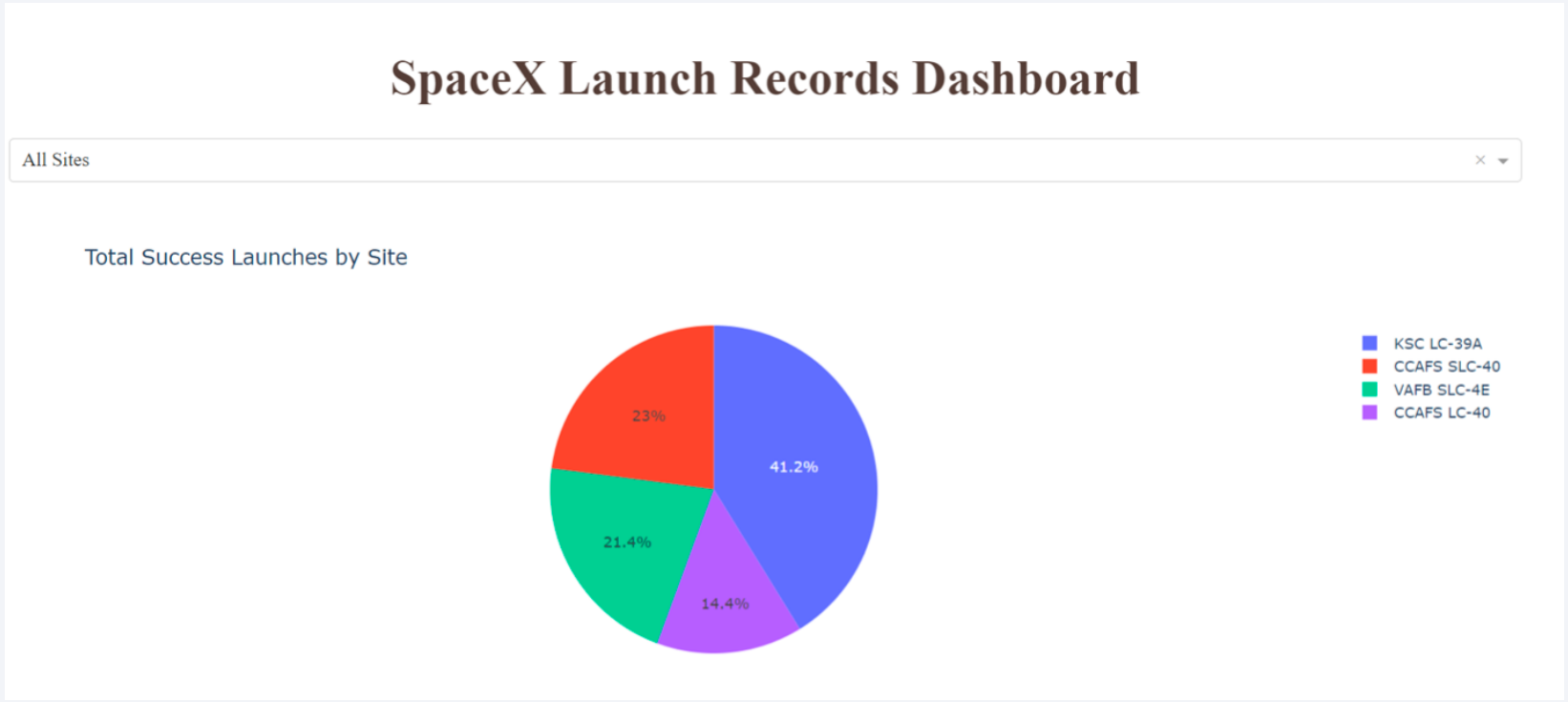
Distances between a launch site to its proximities



Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

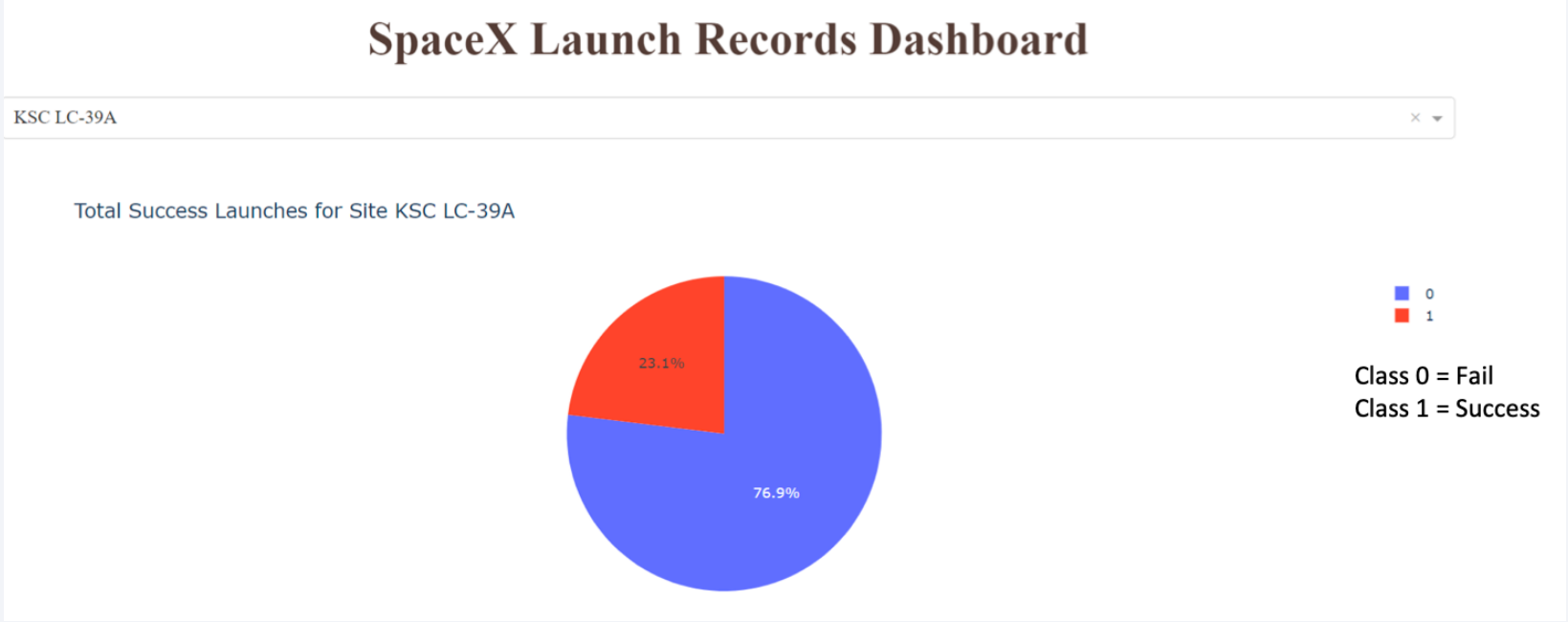

Section 4
Build a Dashboard with Plotly Dash

Launch success by site



- KSC LC-39A had the most successful launches from all the sites.

Launch success(KSC LC-39A)



- KSC LC-39A has the highest success rate amongst launch sites (76.9%)
- 10 successful launches and 3 failed launches

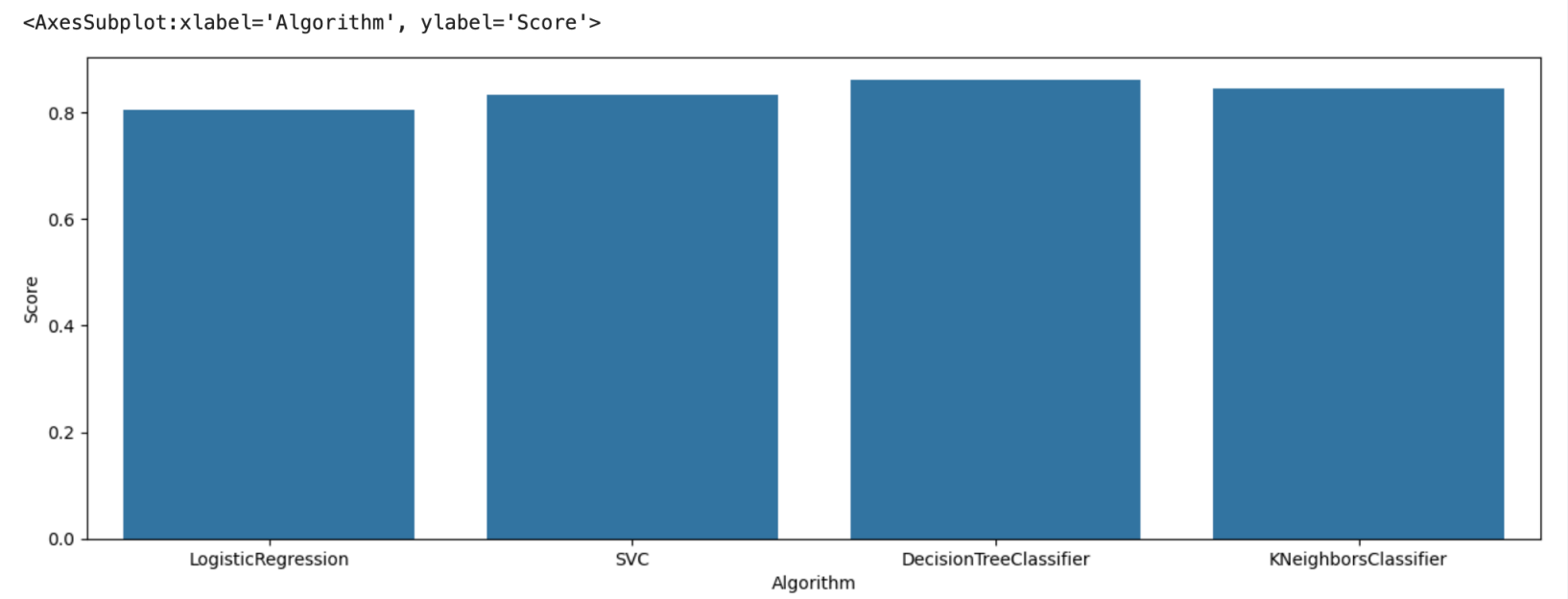
Payload vs. Launch Outcome scatter plot for all sites

- Payloads between 2,000 kg and 5,000 kg have the highest success rate

Section 5

Predictive Analysis (Classification)

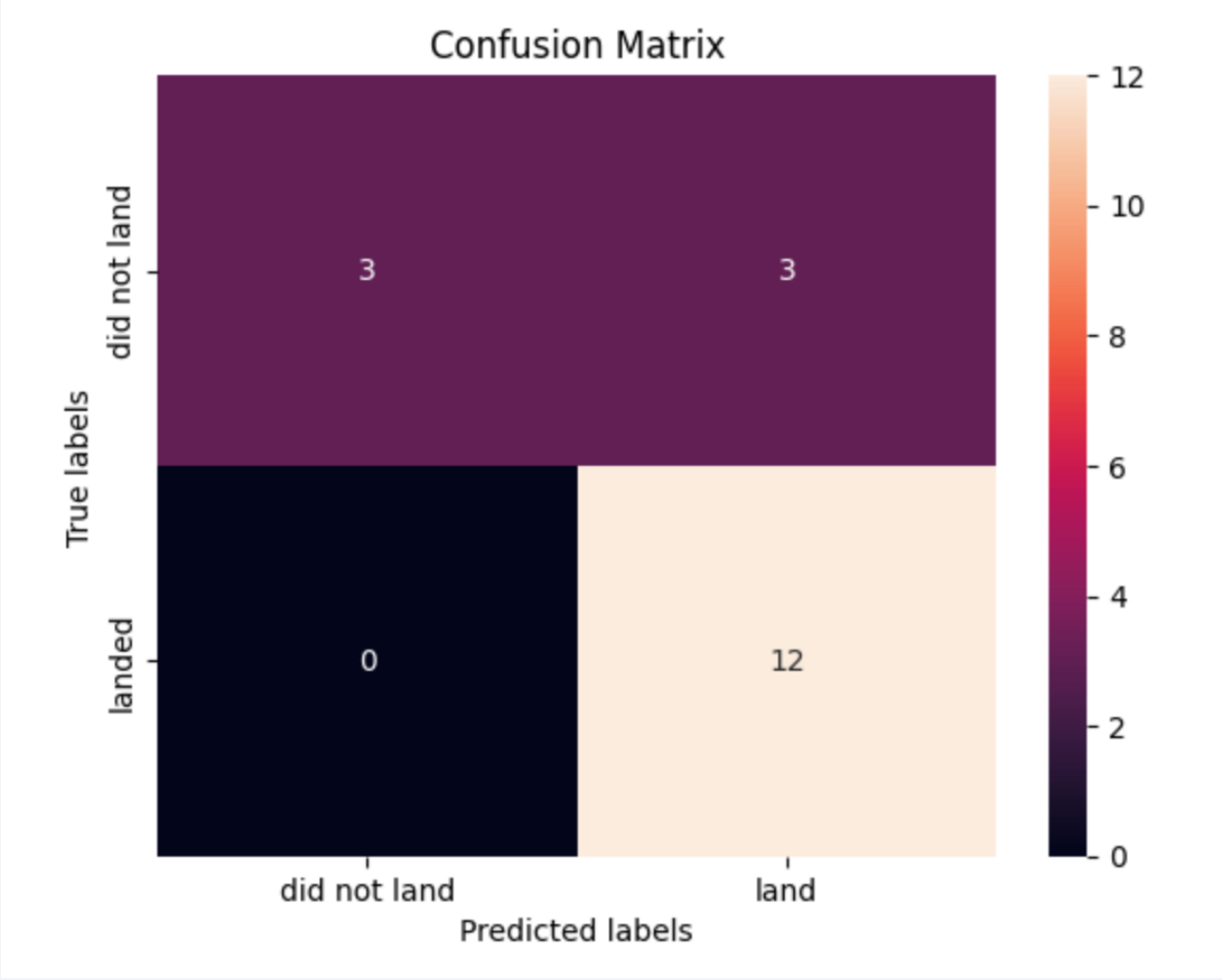
Classification Accuracy



- Of the algorithms tested, the DecisionTreeClassifier was the most accurate.

Confusion Matrix

- All the classification model had the same confusion matrixes and were able equally distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- Success rates increase over time, across all factors, which indicates continuous and incremental operational improvements and technological advancements.
- Different orbits have varying success rates, with ES-L1, SSO, HEO, and GEO showing consistently successful outcomes.
- Launch site was a highly predictive factor, with KSC LC-39A being a top performer, closely followed by CCAFS LC-40.
- Many of the predictive models evaluated were able to predict landing outcome with an acceptable level of accuracy. In the testing performed, DecisionTreeClassifier produced best results with high accuracy, precision, and recall.

Thank you!