IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amrutha C V
20 November 2023

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
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background
  SpaceX is the most successful company which is making space travel affordable for everyone. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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. Spaces X's Falcon 9 launch like regular rockets. This stage is quite large and expensive. Unlike other rocket providers, SpaceX's Falcon 9 Can recover the first stage.As a data scientist working for a new rocket company, goal of this project is to determine the price of each launch.

- Problems you want to find answers
  - Identifying all factors that influence the landing outcome.
  - The relationship between each variables and how it is affecting the outcome.
  - The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scraping from Wikipedia
- Perform data wrangling
  - Data was processed using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data collection is the process of gathering data from various sources, including demographic, clinical, coverage, and pharmaceutical information.As mentioned, the dataset was collected by RESTAPI and Web Scraping from Wikipedia.

- For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe.

- Data cleaning is performed and missing values are replaced

Data Collection – SpaceX API

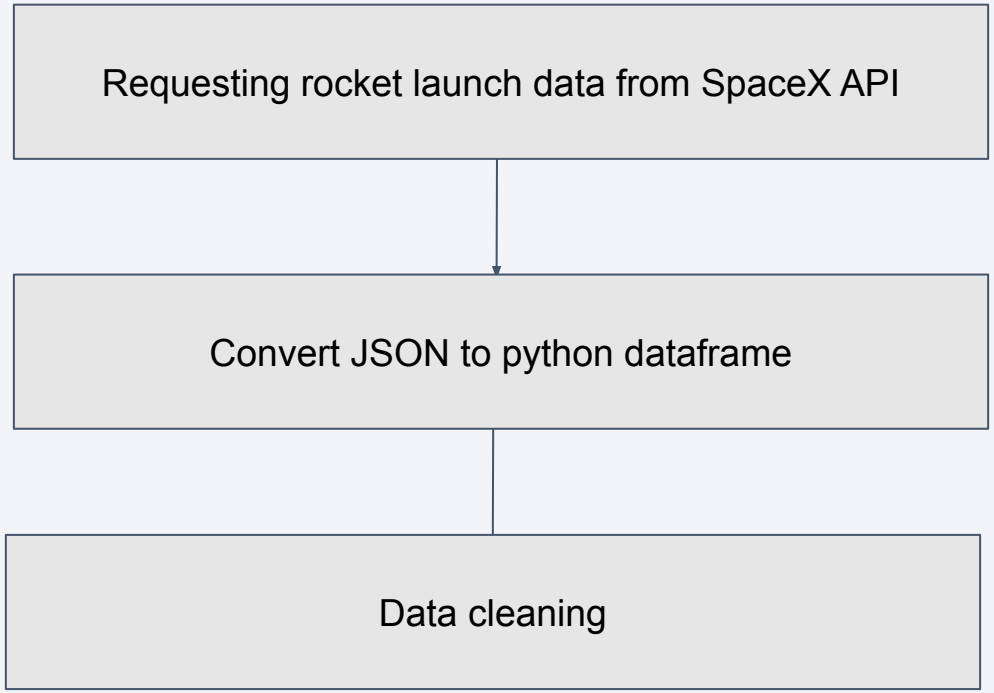
Link:
https://github.com/amruthacv/Applied-datascience-capstone/blob/main/1.1jupyter-labs-spacex-data-collection-api-answers.ipynb

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

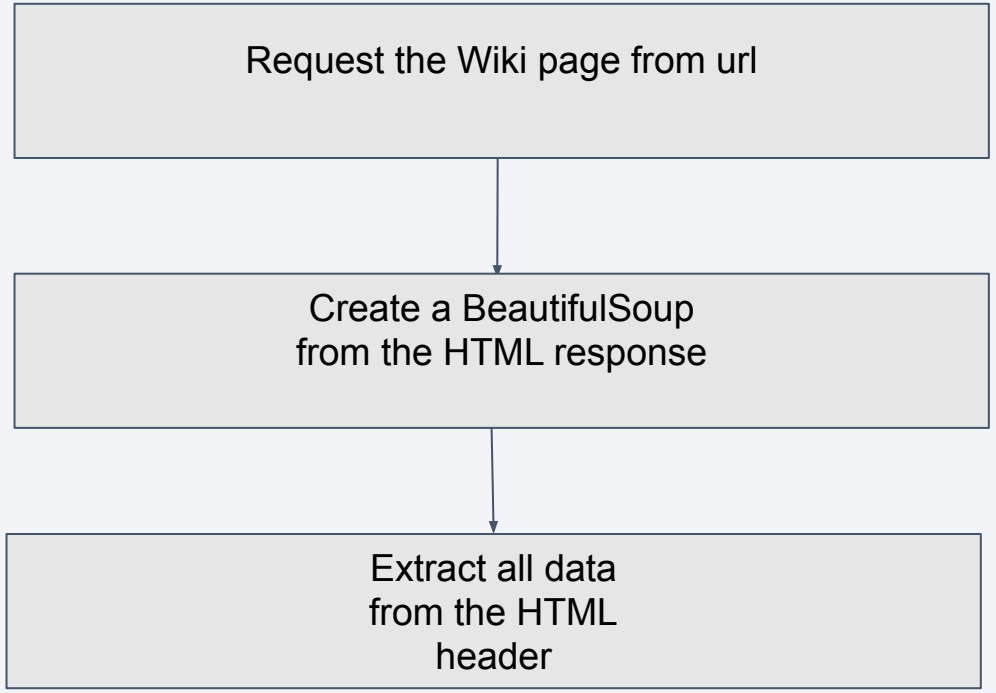
```
response = requests.get(spacex_url)
```

```
# 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, and date_utc.
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 rocket boosters and rows
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 list and replace the
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 leaving the time
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

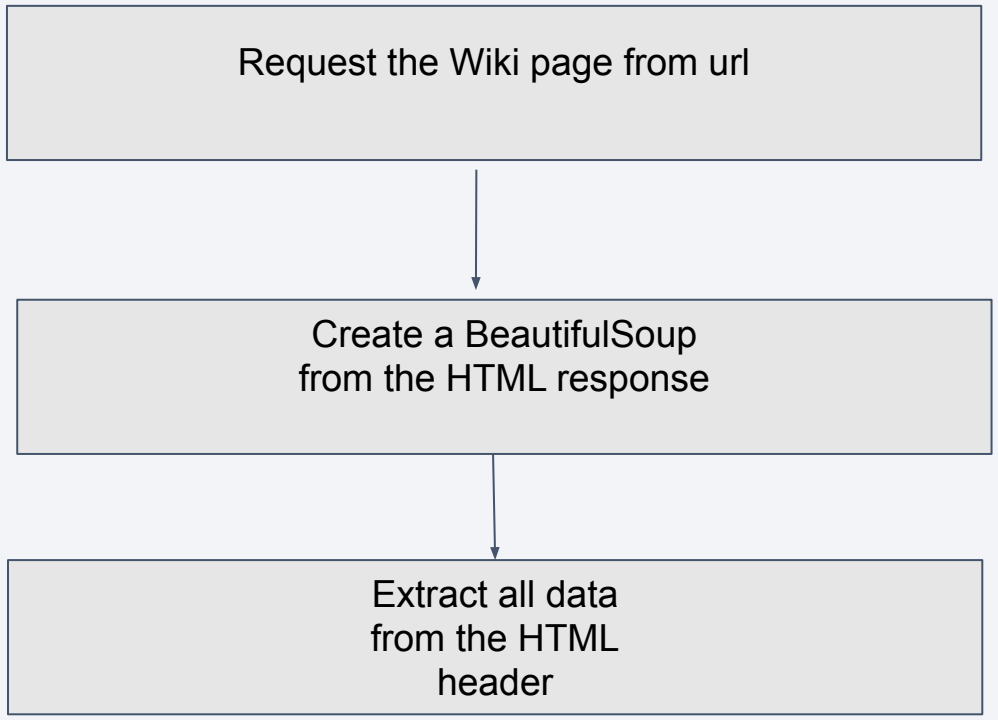


Link: https://github.com/amruthacv/Applied-datascience-capstone/blob/main/1.2jupyter-labs-webscraping-answers.ipynb

Data Wrangling

• Initially some Exploratory Data Analysis (EDA) was performed on the dataset.
• Then the summary launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.
• Finally, the landing outcome label was created from Outcome column.

Link:
https://github.com/amruthacv/Applied-datascience-capstone/blob/main/1.3labs-jupyter-spacex-Data%20wrangling-answers.ipynb



EDA with Data Visualization

Started by using scatter graph to find the relationship

between the attributes such as between:

- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

Link:
https://github.com/amruthacv/Applied-datascience-capstone/blob/main/2.1jupyter-labs-eda-sql-coursera_sqllite-answers.ipynb

EDA with SQL

The following SQL queries were performed:

- Names of the unique launch sites in the space mission;
- Top 5 launch sites whose name begins with the string 'CCA';
- Total pay load mass carried by boosters launched by NASA (CRS);
- Average payload mass carried by booster version F9 v1.1;
- Date when the first successful landing outcome in ground pad was achieved;
- Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000

kg;

- Total number of successful and failure mission outcomes;
- Names of the booster versions which have carried the maximum payload mass;
- Failed landing out comes in droneship, their booster versions, and launch site names for in year 2015; and
- Rank of the count of landing outcomes (such as Failure (droneship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.

Link:
https://github.com/amruthacv/Applied-datascience-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs2.2jupyter-labs-eda-dataviz-answers.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines and marker clusters were used with Folium Maps
- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.

Link:
https://github.com/amruthacv/Applied-datascience-capstone/blob/main/3.1lab_jupyter_launch_site_location.jupyterlite-answers.ipynb

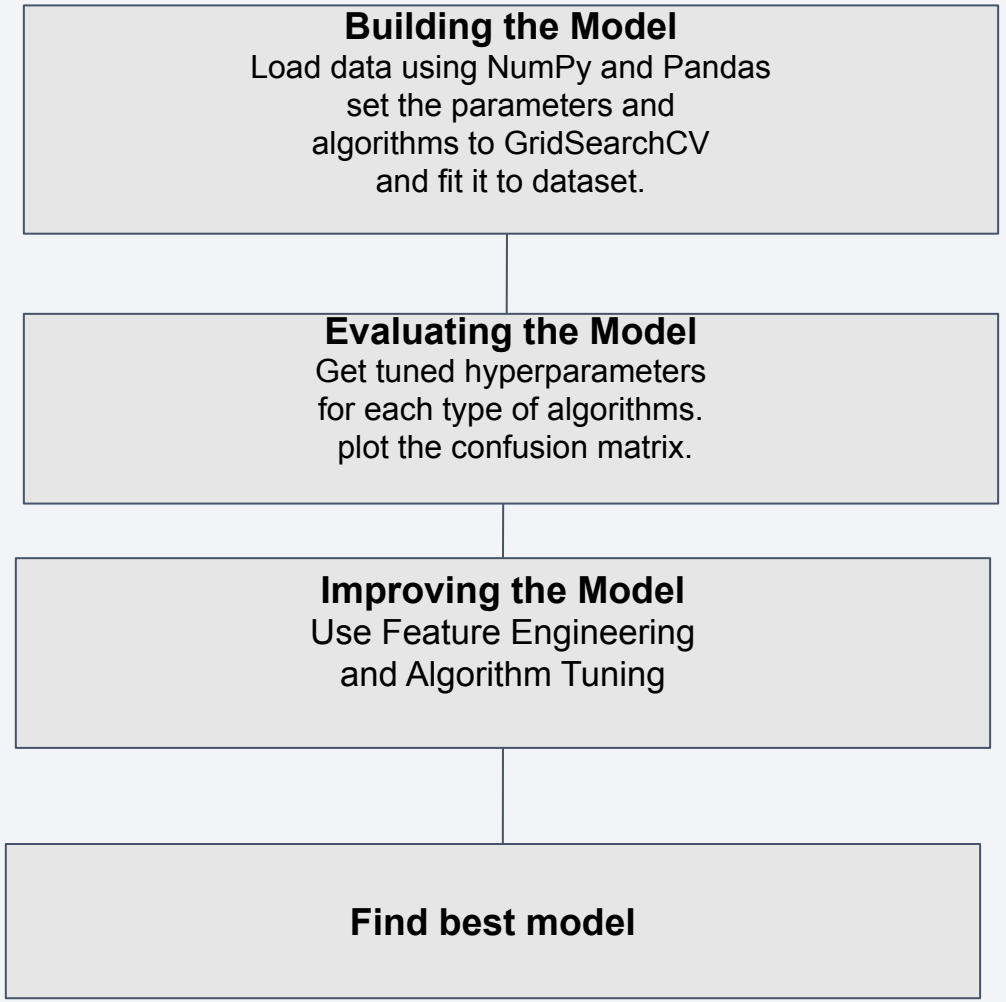
Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- Plotted pie charts showing the total launches by a certain sites.
- Plotted scatter graph showing the relationship with Outcome and Payload

Mass (Kg) for the different booster version.

Link:
https://github.com/amruthacv/Applied-datascience-capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)



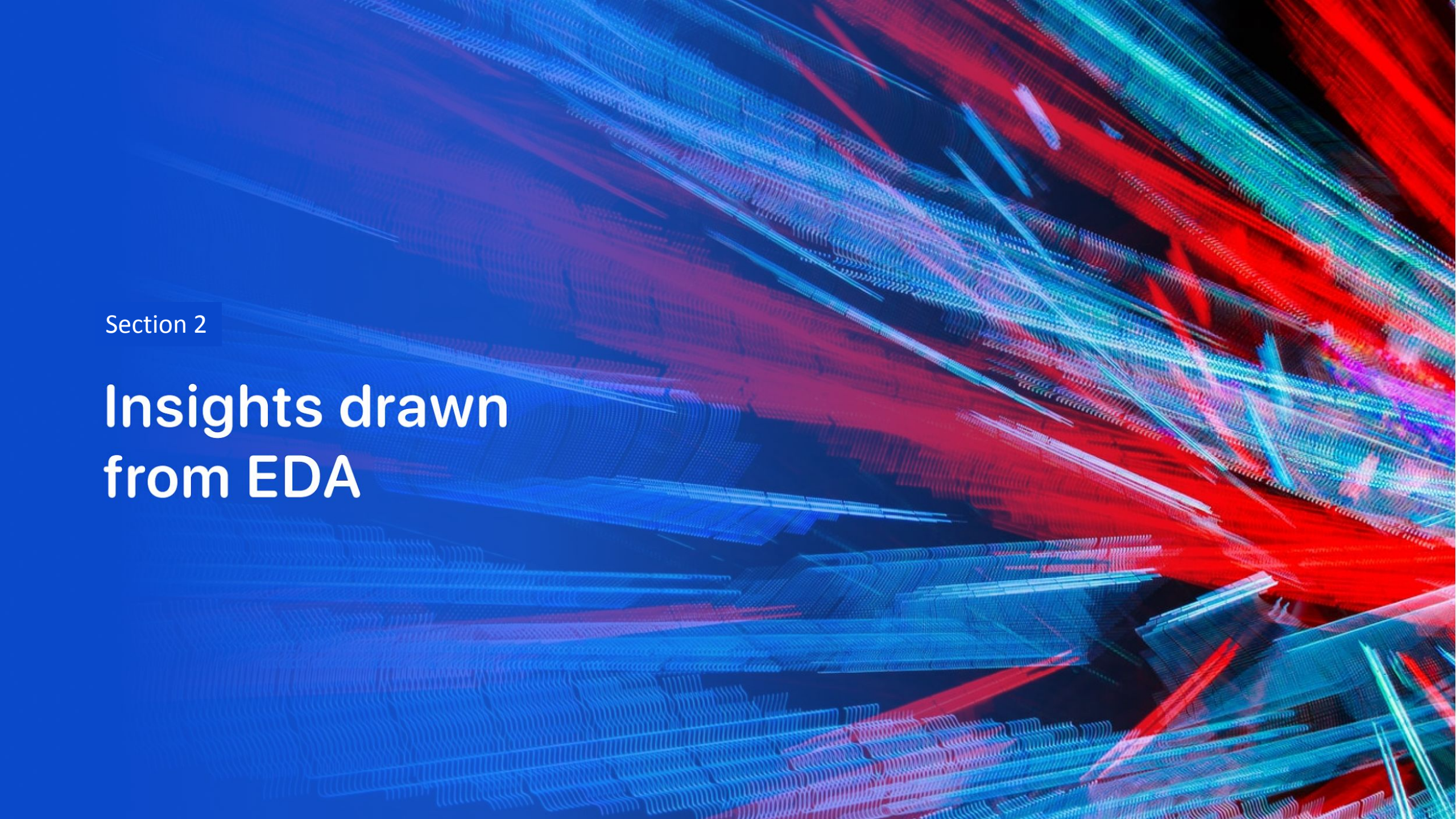
Link:https://github.com/amruthacv/Applied-datascience-capstone/blob/main/4.1SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite-answers.ipynb

Results

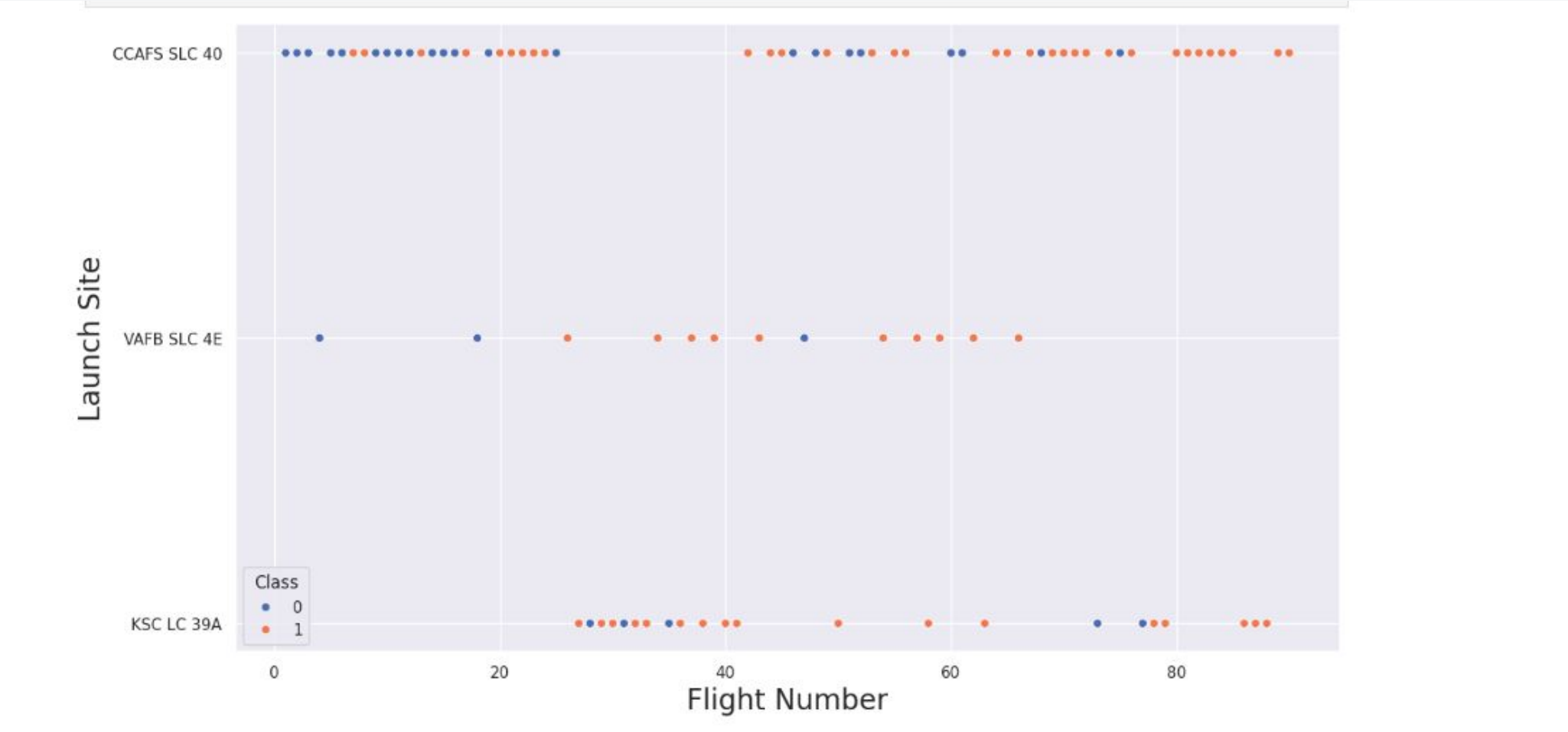
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

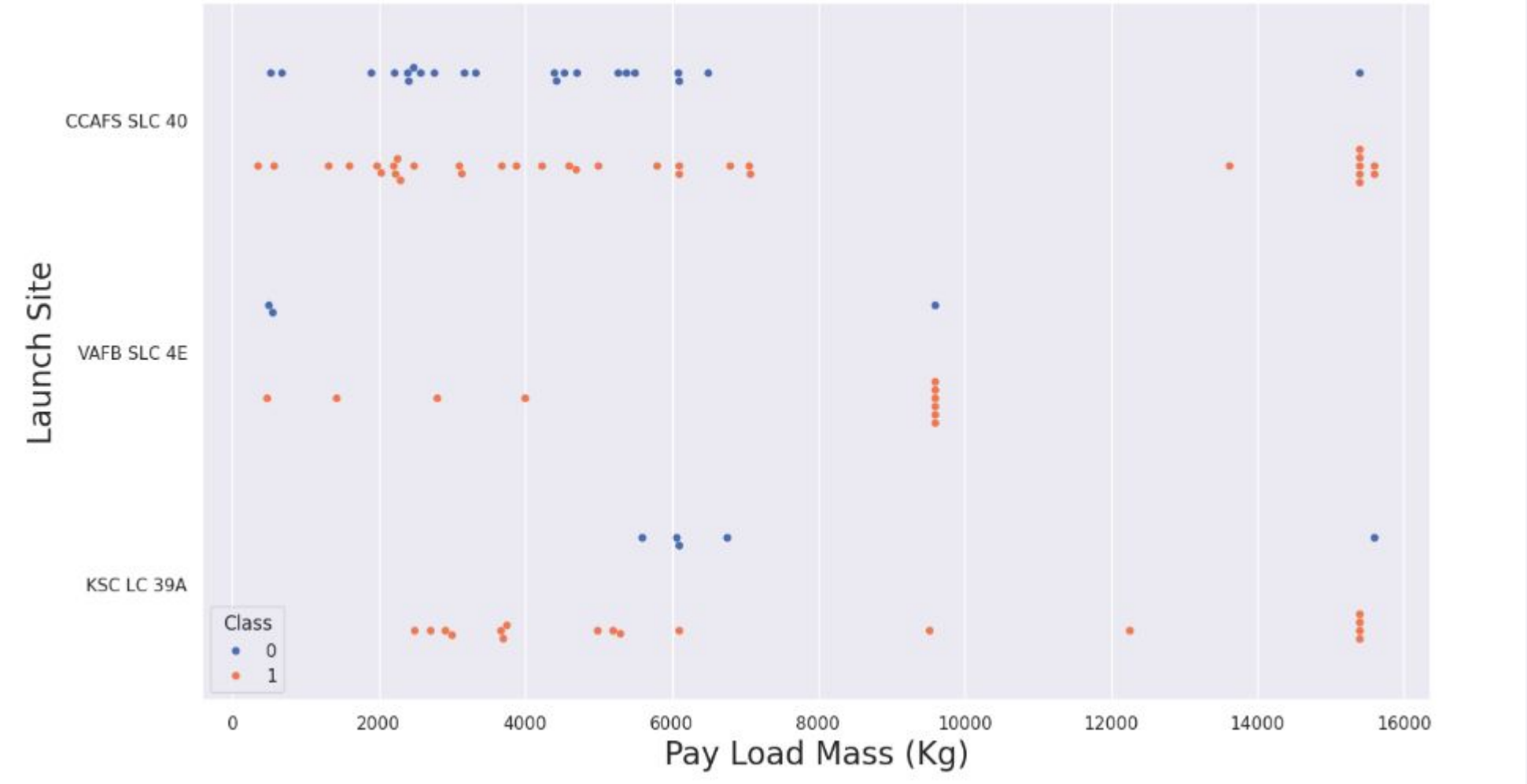


Flight Number vs. Launch Site



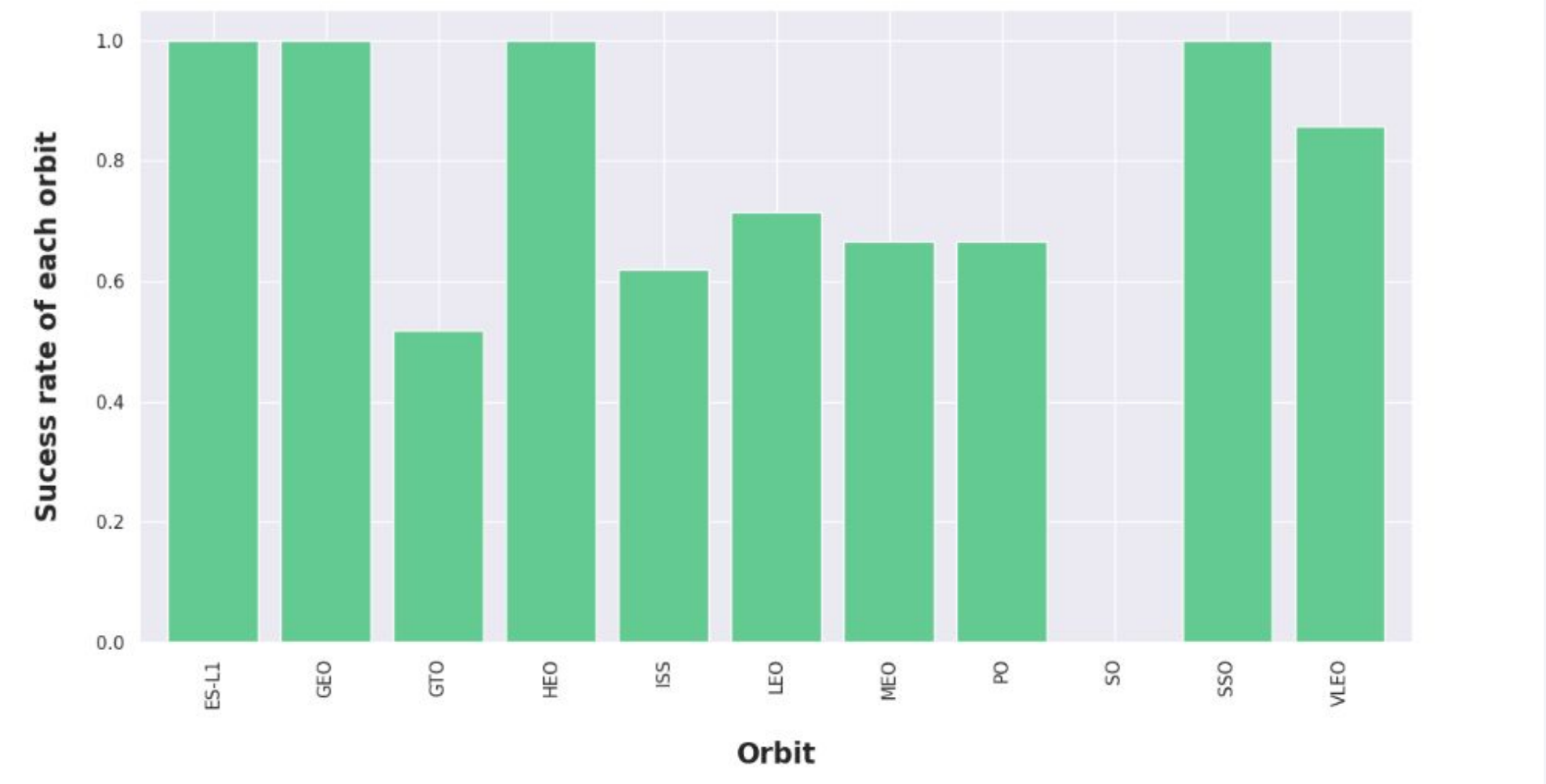
larger the flights amount of the launch site, the greater the the success rate will be.

Payload vs. Launch Site



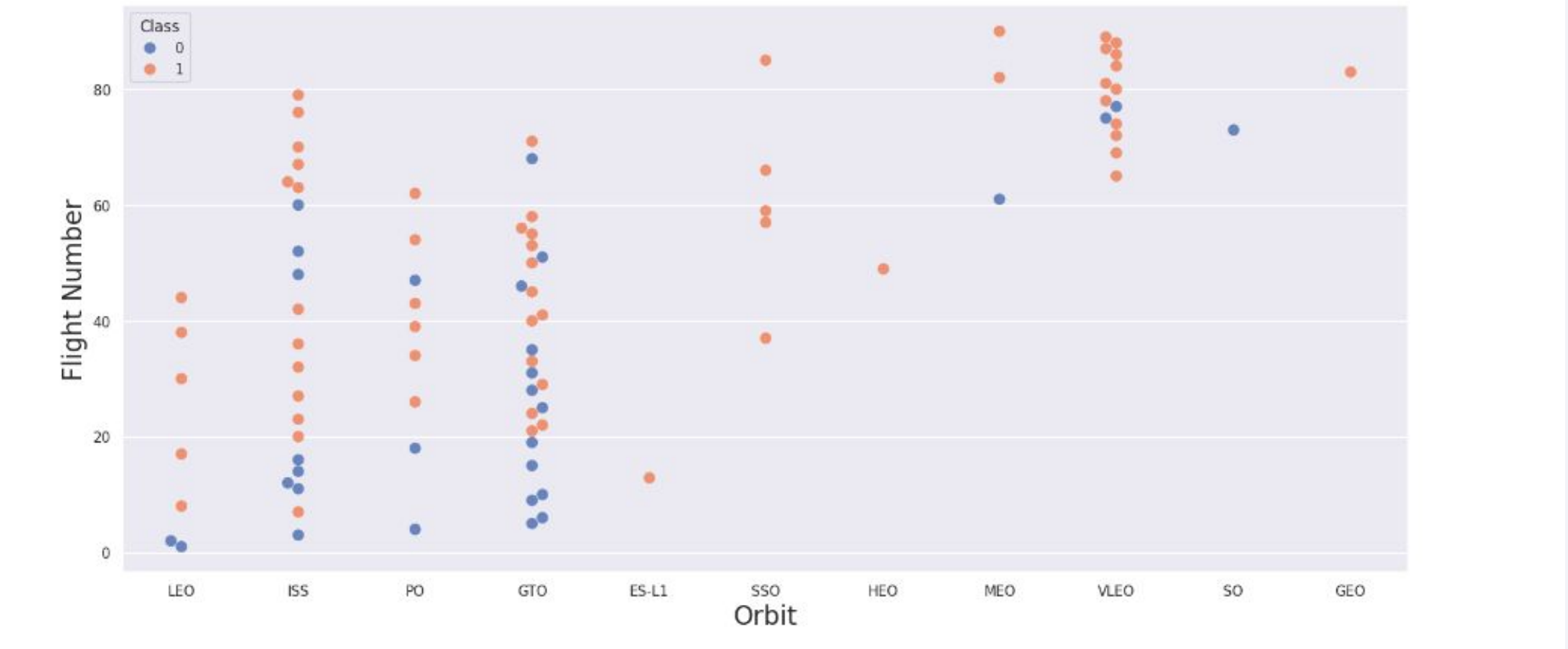
once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased.

Success Rate vs. Orbit Type



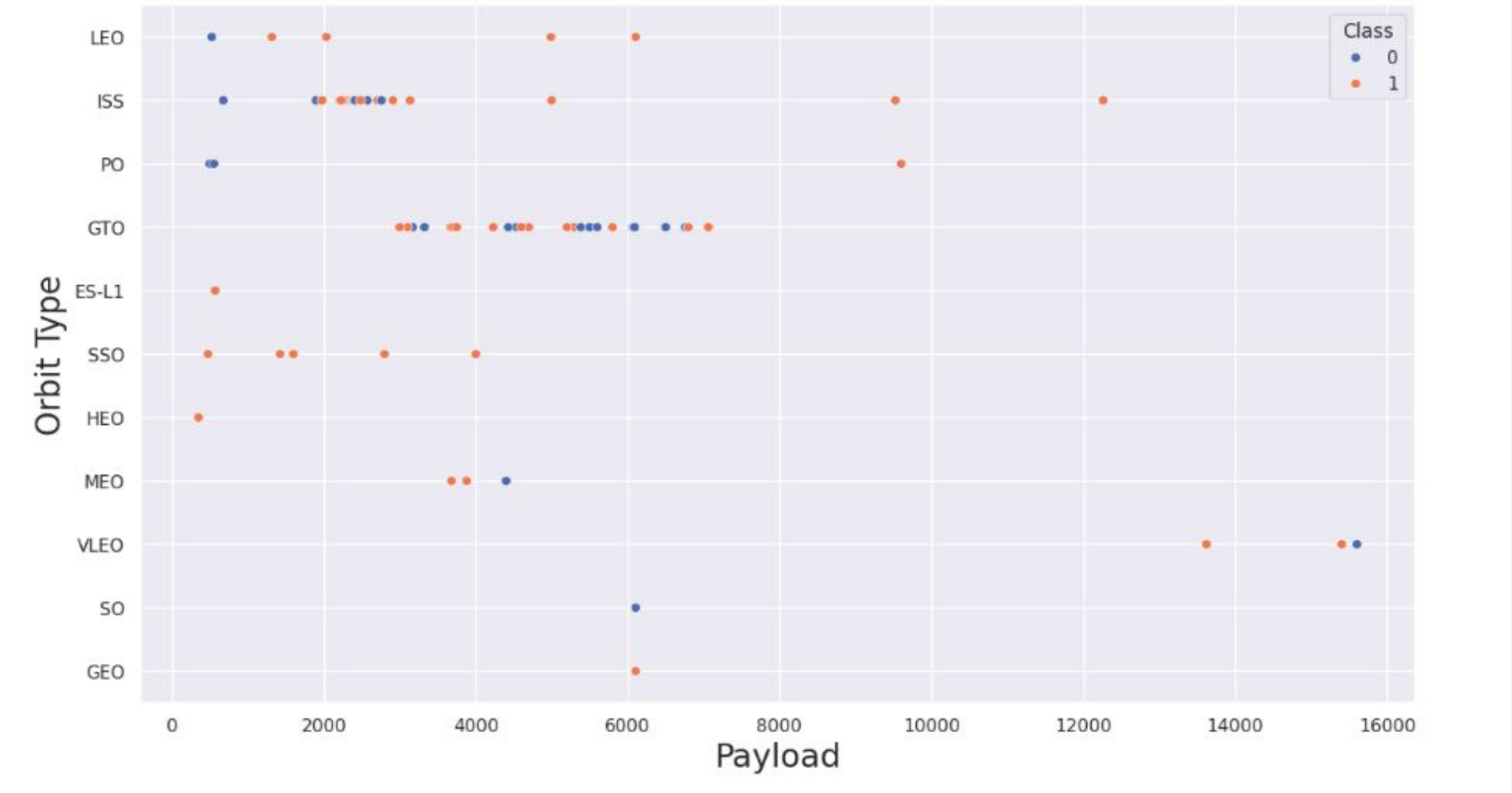
From the plot, we can see that ES-L1, GEO, HEO, SSO,VLEO had the most success rate

Flight Number vs. Orbit Type



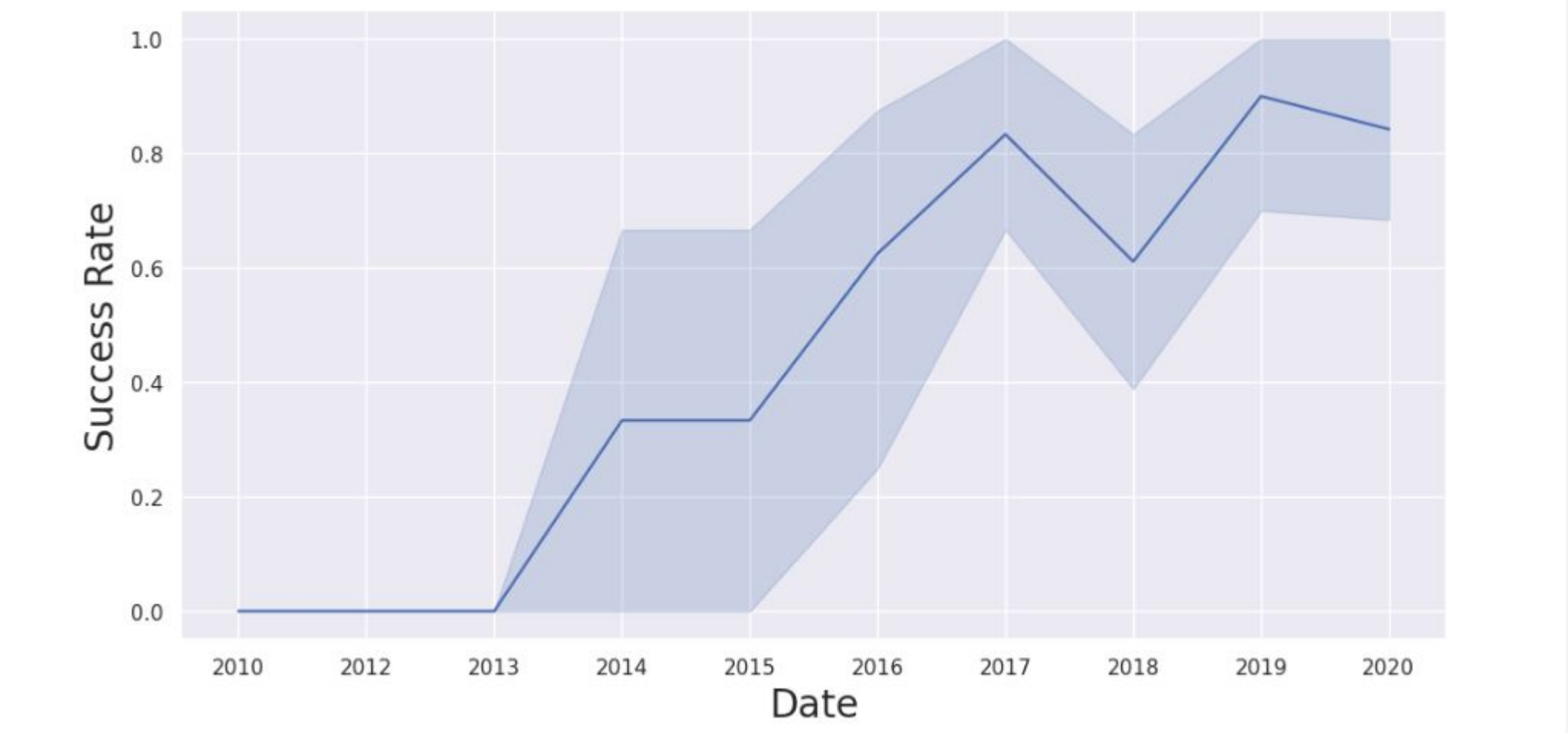
We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type



with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

Launch Success Yearly Trend



success rate since 2013 kept on increasing till 2020.

All Launch Site Names

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%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'

In [10]:
```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

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

Out[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parac
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 (parac
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No att
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No att
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No att

We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
In [11]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
Out[11]: sum(PAYLOAD_MASS__KG_)
---------------------------
                       45596
```

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version

F9 v1.1 as 2928.4

```
In [12]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
Out[12]: avg(PAYLOAD_MASS__KG_)
         2928.4
```

First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was 22 nd December 2015

```
In [14]: %sql SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)
 * sqlite:///my_data1.db
Done.
```

Out[14]:

First Successful Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
In [26]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ >
 * sqlite:///my_data1.db
Done.
```

Out[26]:

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

Total Number of Successful and Failure Mission Outcomes

We used wildcard like '%' to filter for WHERE MissionOutcome was a success or a failure.

In [28]:
```
%sql SELECT COUNT MISSION_OUTCOME AS "succesful mission "FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%'
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "AS": syntax error
[SQL: SELECT COUNT MISSION_OUTCOME AS "succesful mission "FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%']
(Background on this error at: http://sqlalche.me/e/e3q8)
```

In [29]:
```
%sql SELECT COUNT MISSION_OUTCOME AS "failure mission " FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Fail%'
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "AS": syntax error
[SQL: SELECT COUNT MISSION_OUTCOME AS "failure mission " FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Fail%']
(Background on this error at: http://sqlalche.me/e/e3q8)
```

In [30]:
```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEXTBL;
```

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

Out[30]:

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in thenWHERE clause and the MAX() function.

In [31]:
```
%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.
```

Out[31]:

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

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

In [36]:

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
LANDING_OUTCOME = '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 SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

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

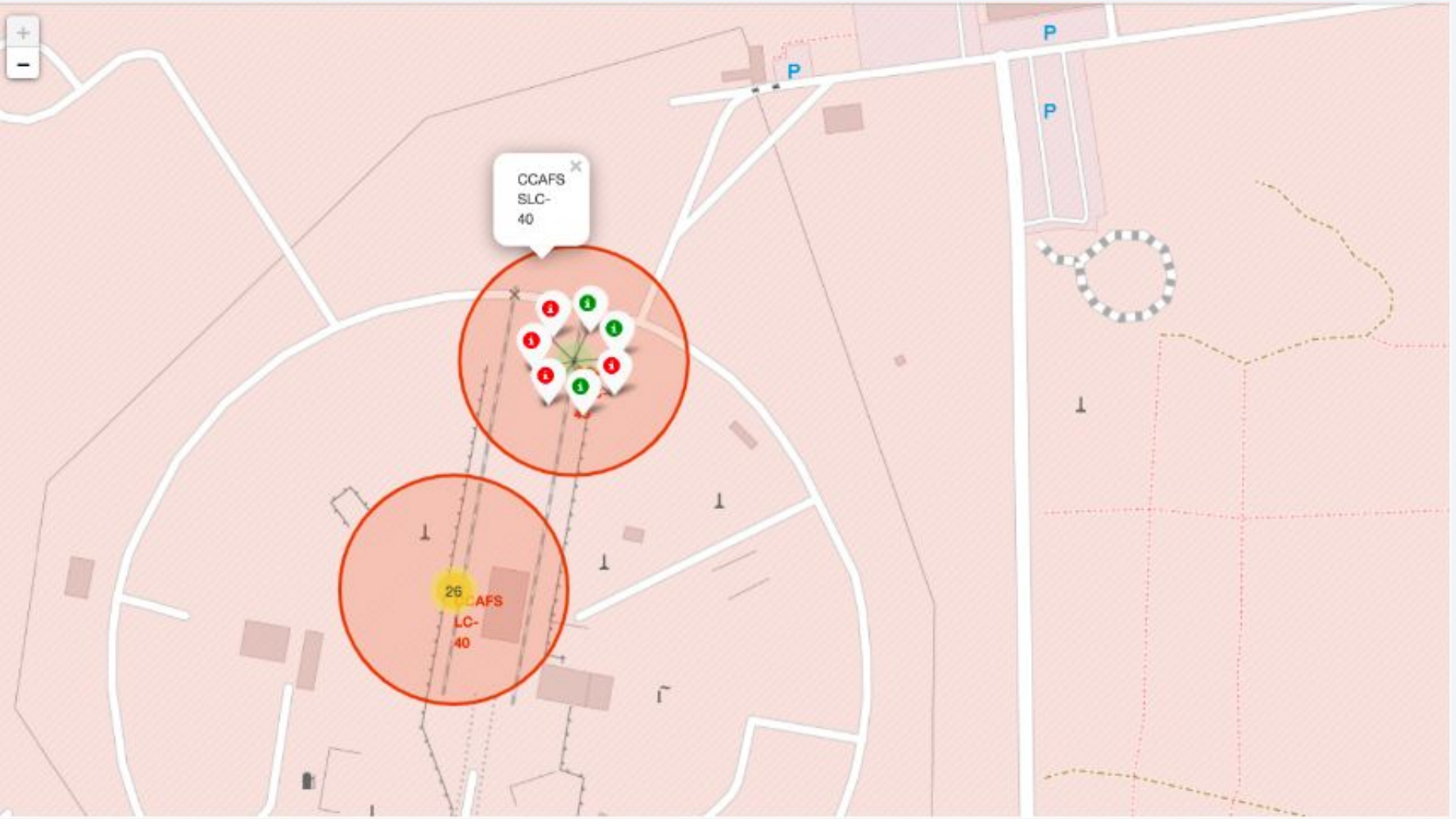
- Selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- Applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.


Section 3
Launch Sites
Proximities Analysis

Location of all the Launch Sites

We can see that all the SpaceX launch sites are located inside the United States

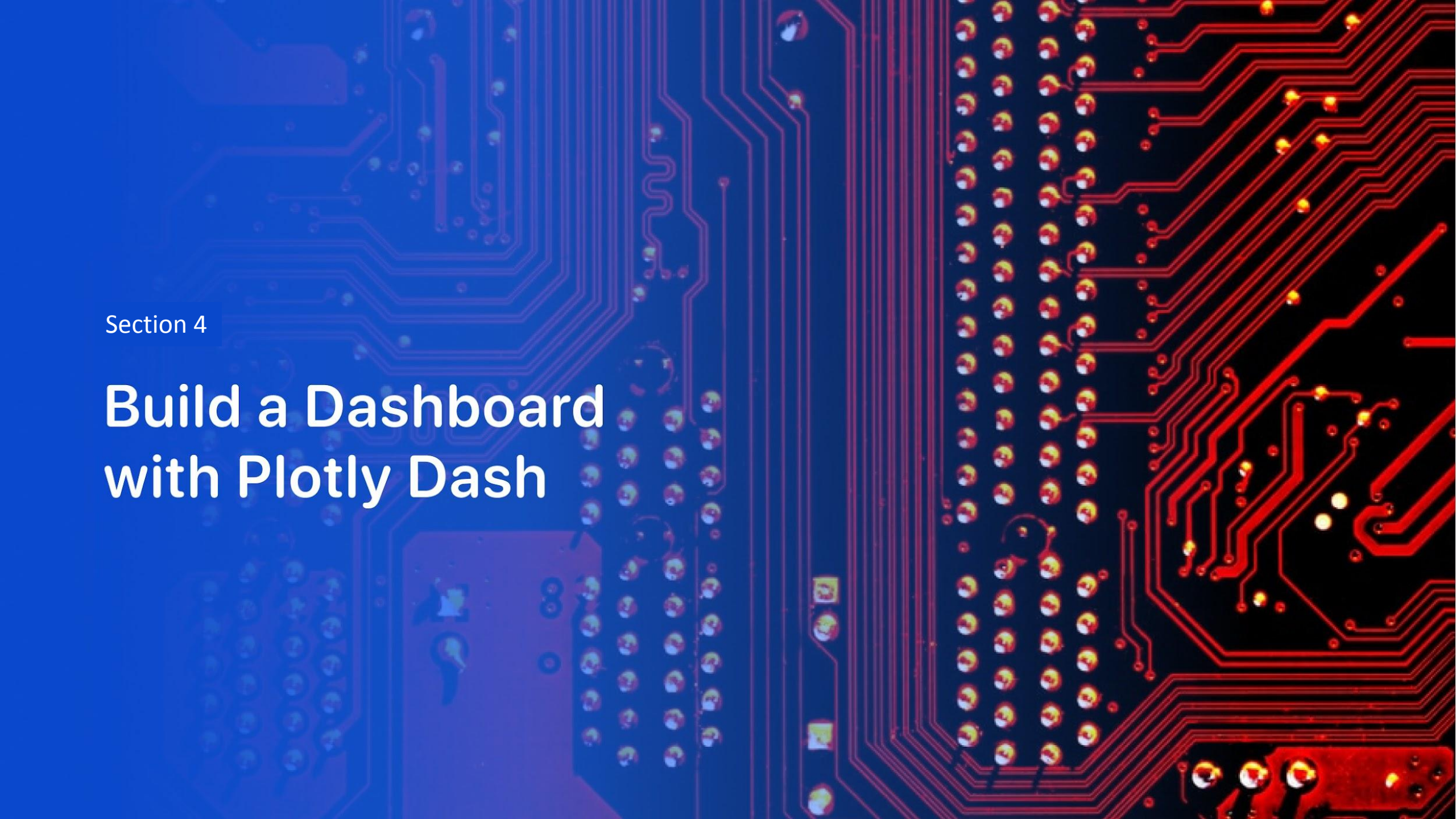
Markers showing launch sites with color labels



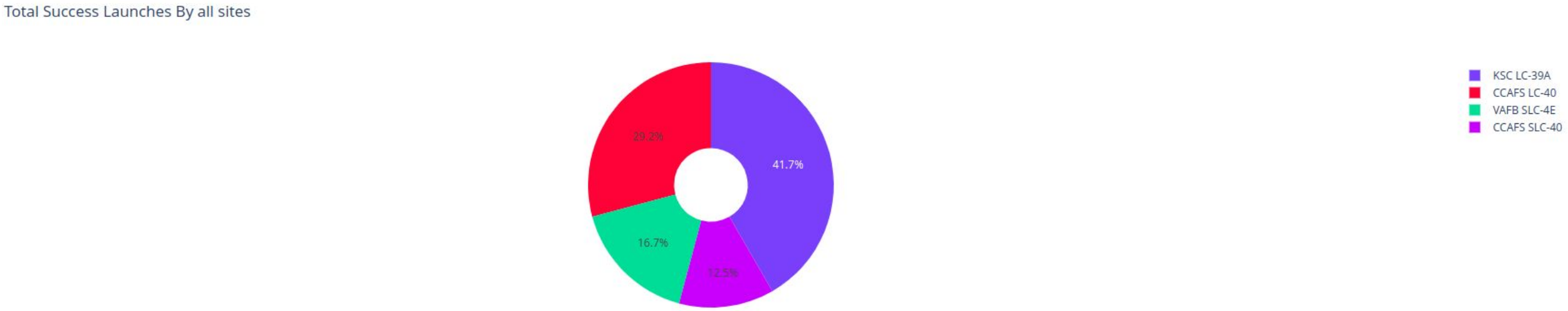
Red -> Successful launches
Green-> Failures

Launch Sites Distance to Landmarks

- Are launch sites in close proximity to railways? - No
- Are launch sites in close proximity to highways? - No
- Are launch sites in close proximity to coastline? - Yes
- Do launch sites keep certain distance away from cities? - Yes

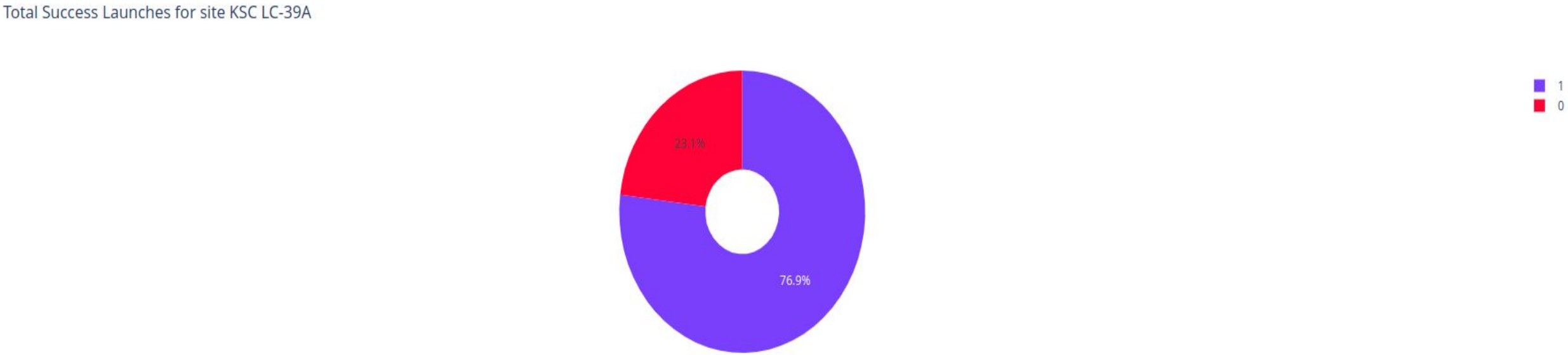

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



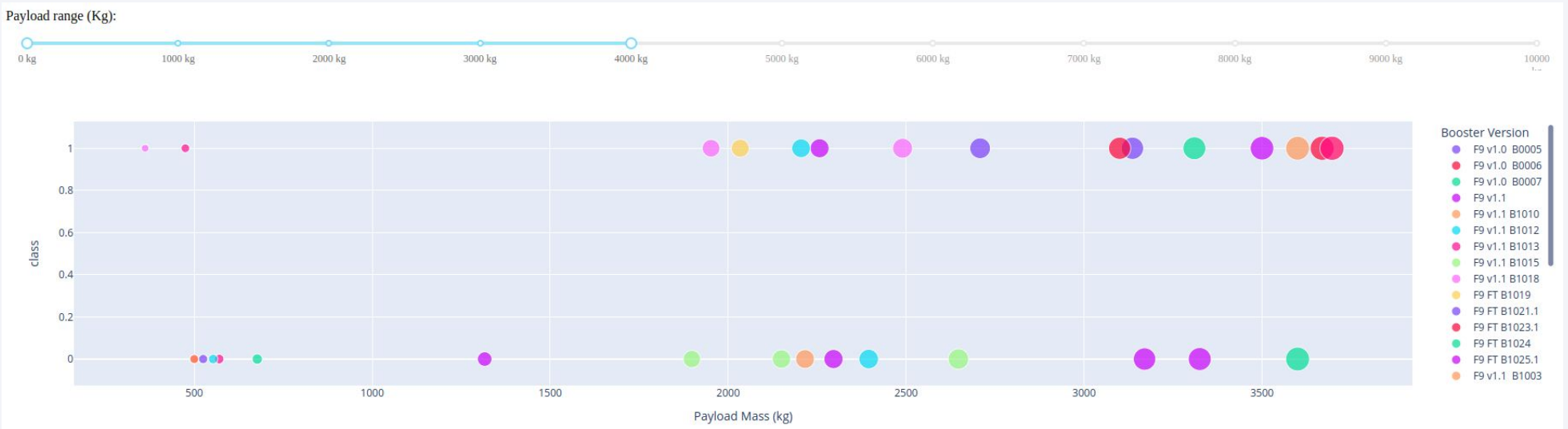
KSC LC-39A had most successful launches from all the sites

The highest launch-success ratio: KSC LC-39A



KSC LC-39A has 76.9% success rate and 23.1% failure rate.

Payload vs Launch Outcome Scatter Plot



Low weighted payload is higher than heavy weighted payload

Section 5

Predictive Analysis (Classification)

Classification Accuracy

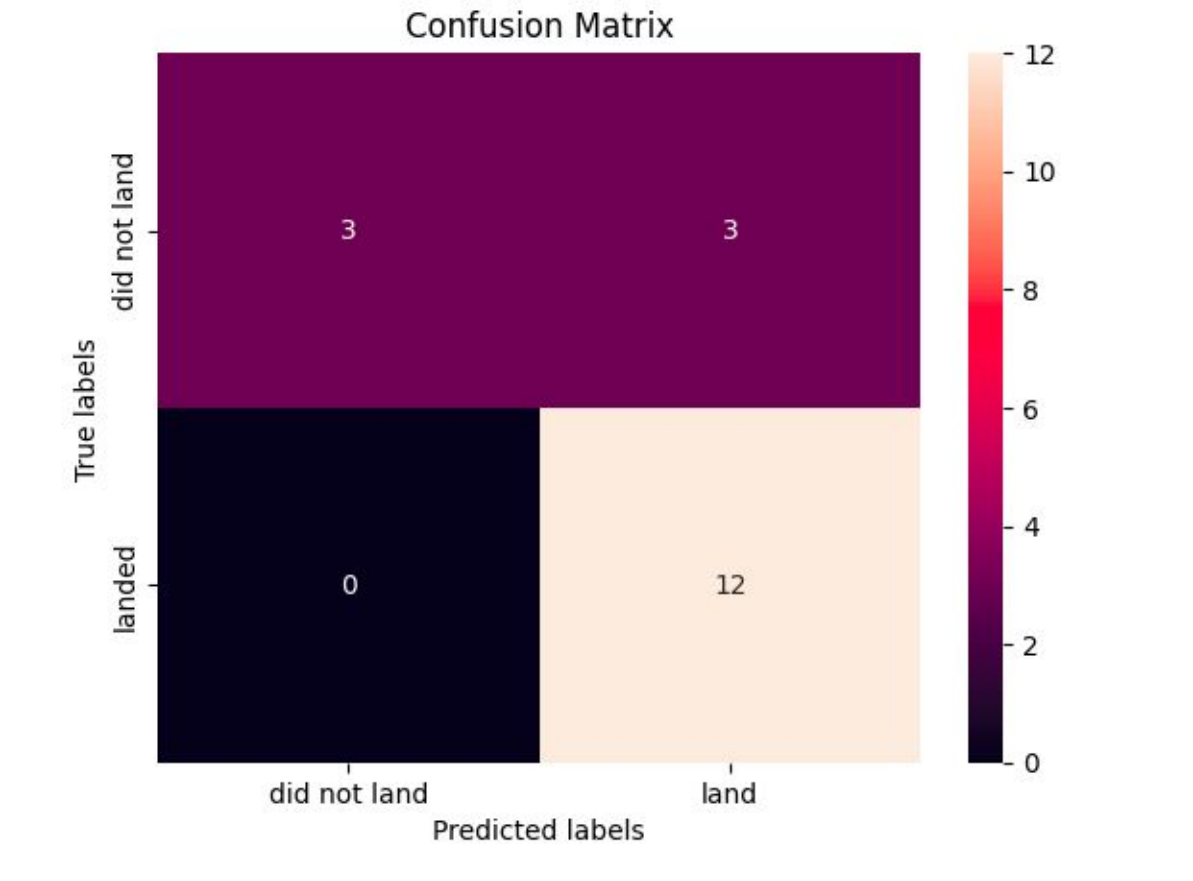
In [57]:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8892857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samp
les_split': 10, 'splitter': 'random'}
```

The best algorithm to be the Tree Algorithm which have the highest classification accuracy.

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

Conclusions are:

- The Decision tree classifier is the best machine learning algorithm for this task.
- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.

Thank you!