IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sadegh Jokar
17 August 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 result
- Interactive analytics in screenshots
- Predictive Analytics result

Introduction

Project background and context

Space X Falcon 9 costs 62 million dollars; other providers cost upward of 165 million dollars each, because Space X 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 space X for a rocket launch. The goal of the project is to create a machine learning model to predict if the first stage will land successfully.

Problems you want to find answers

- How payload mass, launch site, number of flights, and orbits affect first-stagelanding success
- Rate of successful landings over time
- Best predictive model for successful landing (binary classification)

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

- **Request data** from SpaceX API (rocket launch data)
- **Decode response** using .json() and convert to a dataframe using .json_normalize()
- **Request information** about the launches from SpaceX API using custom functions
- **Create dictionary** from the data
- **Create dataframe** from the dictionary
- **Filter dataframe** to contain only Falcon 9 launches
- **Replace missing values** of Payload Mass with calculated .mean()
- **Export data** to csv file

Data Collection – SpaceX API

Step1

- Request for rocket launch data using API

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

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

Step2

- Form the Dataframe and filter data

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

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=data[data['BoosterVersion']!='Falcon 1']
data_falcon9.head
```

Step3

- Data Cleaning and handling missing values

```
# Calculate the mean value of PayloadMass column
payloadmean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9.replace(np.nan,payloadmean)
```

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/1-jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Step1

- Request the Falcon9 Launch Wiki page from its URL

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

Step2

- Create a BeautifulSoup from the HTML response

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

Step3

- Extract all column/variable names from the HTML header

```
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:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
```

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/2-jupyter-labs-webscraping.ipynb

Data Wrangling

Step1

- **Calculate the number of launches on each site**

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

Step2

- **Calculate the number and occurrence of each orbit**

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Step3

- **Calculate the number and occurence of mission outcome of the orbits**

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

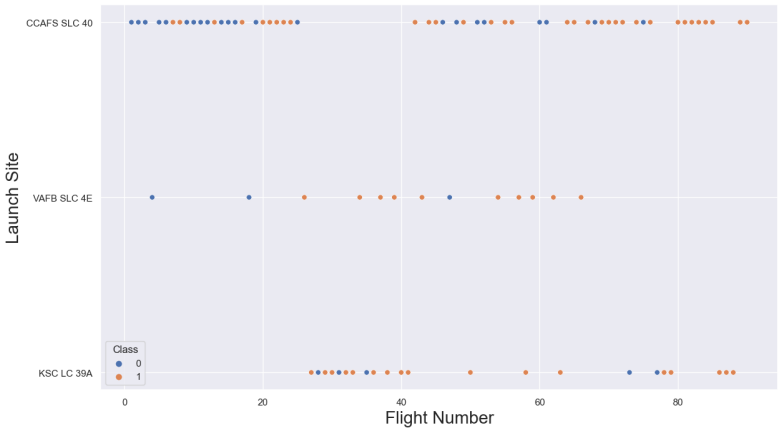
Step4

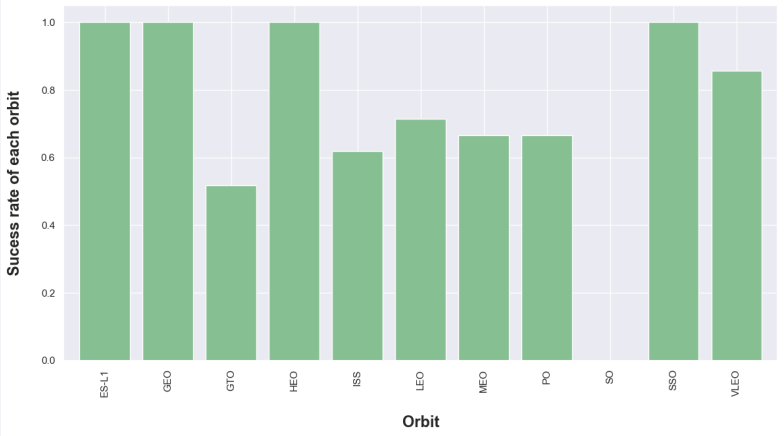
Create a landing outcome label from Outcome column

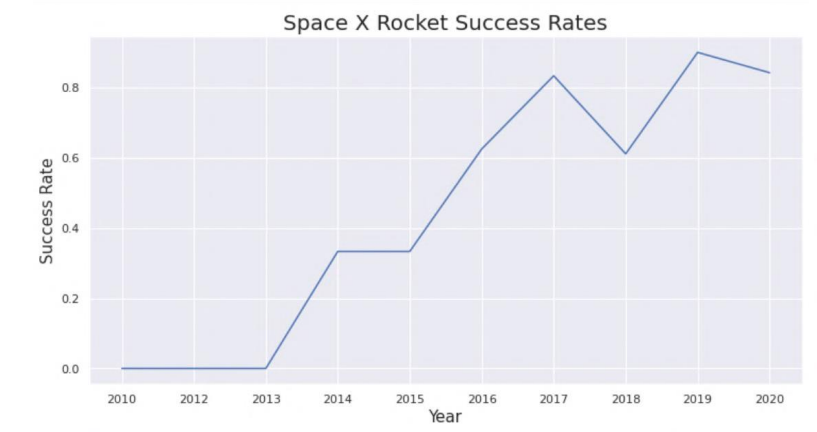
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key,value in df['Outcome'].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/3-labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization







Observe the relationships using scatter plot and use other visualization for further analysis.

Use Bar to interpret the relationship between the attributes and determine which orbits have the highest probability of success.

show a trends or pattern of the attribute over time using line graph and see the launch success yearly trend.

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

Queries that were performed to understand dataset better :

- Names of unique launch sites
- 5 records where launch site begins with ‘CCA’
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1.
- Date of first successful landing on ground pad
- Names of boosters with success landing on drone ship and payload mass between 4,000 and 6,000
- Total number of successful and failed missions
- Names of booster versions which have carried the max payload
- Failed landing outcomes on drone ship, their booster version and launch site for the months in 2015
- Count of landing outcomes between 2010-06-04 and 2017-03-20 (desc)

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

We took the latitude and longitude coordinates at each launch site To visualize the launch data into an interactive map and added a circle marker around each launch site with a label of the name of the launch site.

Then we assigned the dataframe launch_outcomes(failure,success) to classes 0 and 1 with Red and Green markers on the map in MarkerCluster().

Then we used the Haversine's formula to calculated the distance of the launch sites to various landmark to find answer to the questions of:

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

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/6-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/7_SpaceX_Interactive_Visual_Analytics_Plotly.py

Predictive Analysis (Classification)

Building the Model

- Load the dataset into NumPy and Pandas,Transform the data and then split into training and test datasets
- Decide which type of ML to use
- set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- plot the confusion matrix.

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

- The model with the best accuracy score will be the best performing model.

LINK:
https://github.com/Sjokar/IBM-DS-Capstone/blob/main/8-SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Exploratory Data Analysis

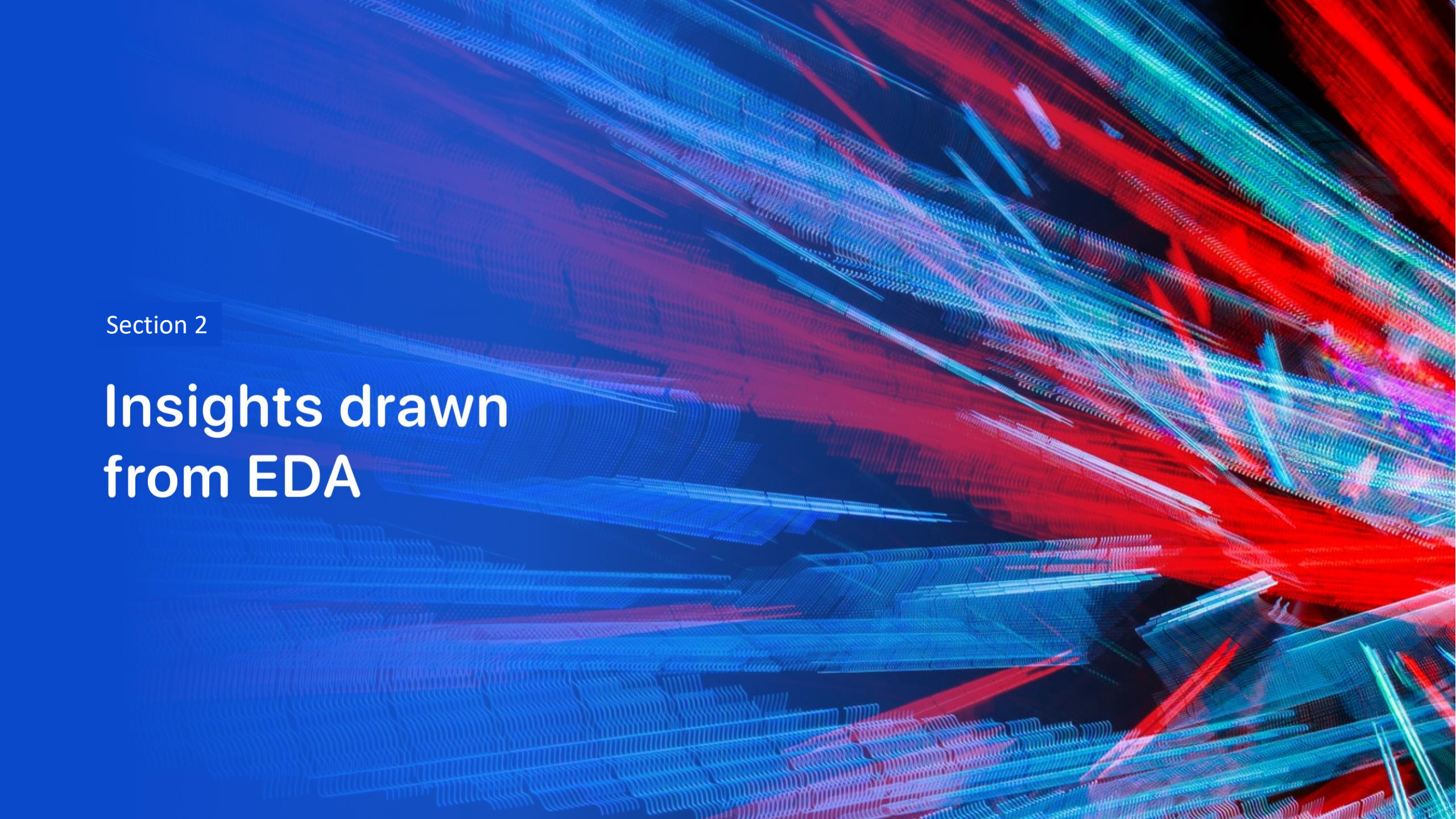
- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate

Visual Analytics

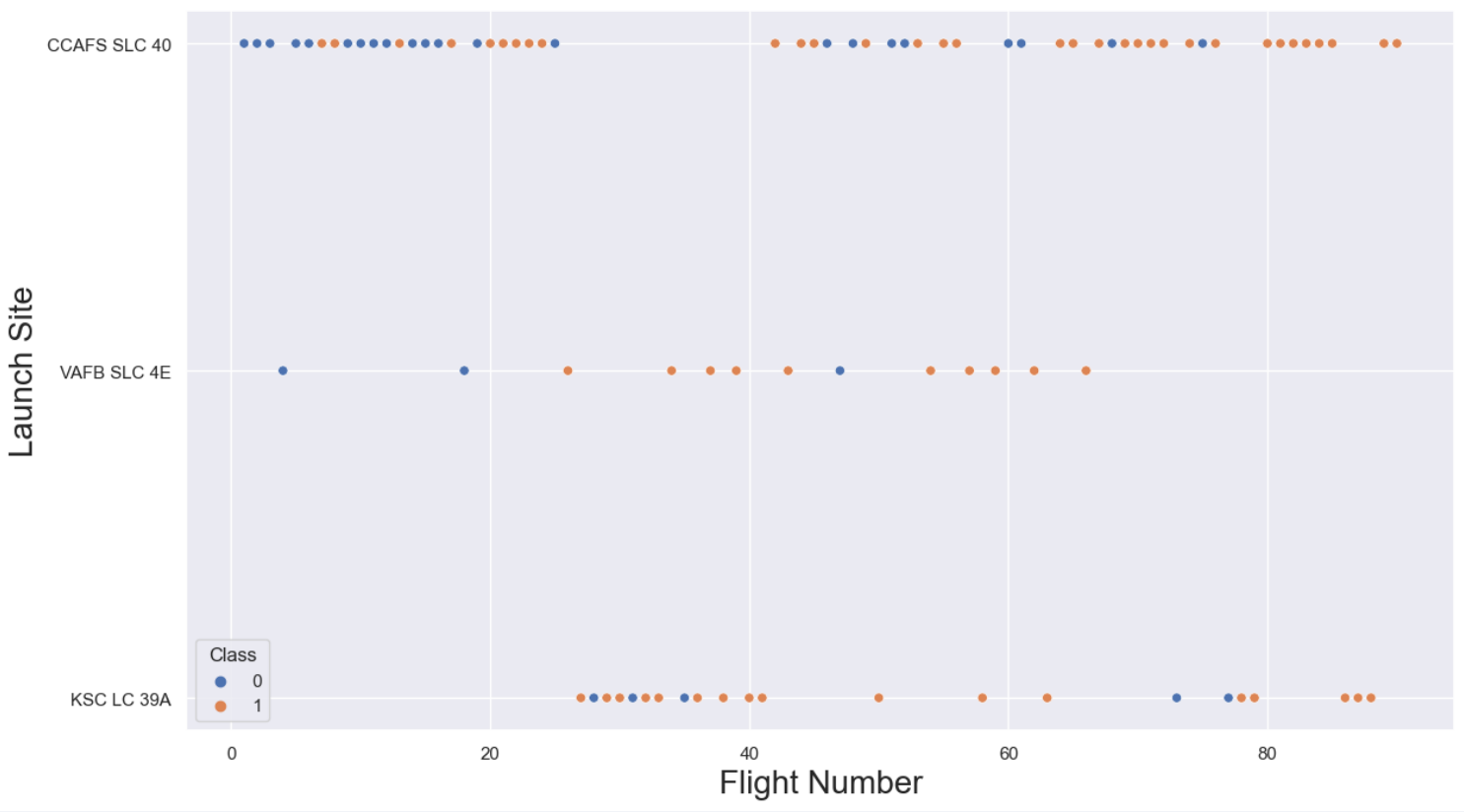
- Most launch sites are near the equator, and all are close to the coast
- Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities

Predictive Analytics

- Decision Tree model is the best predictive model for the dataset

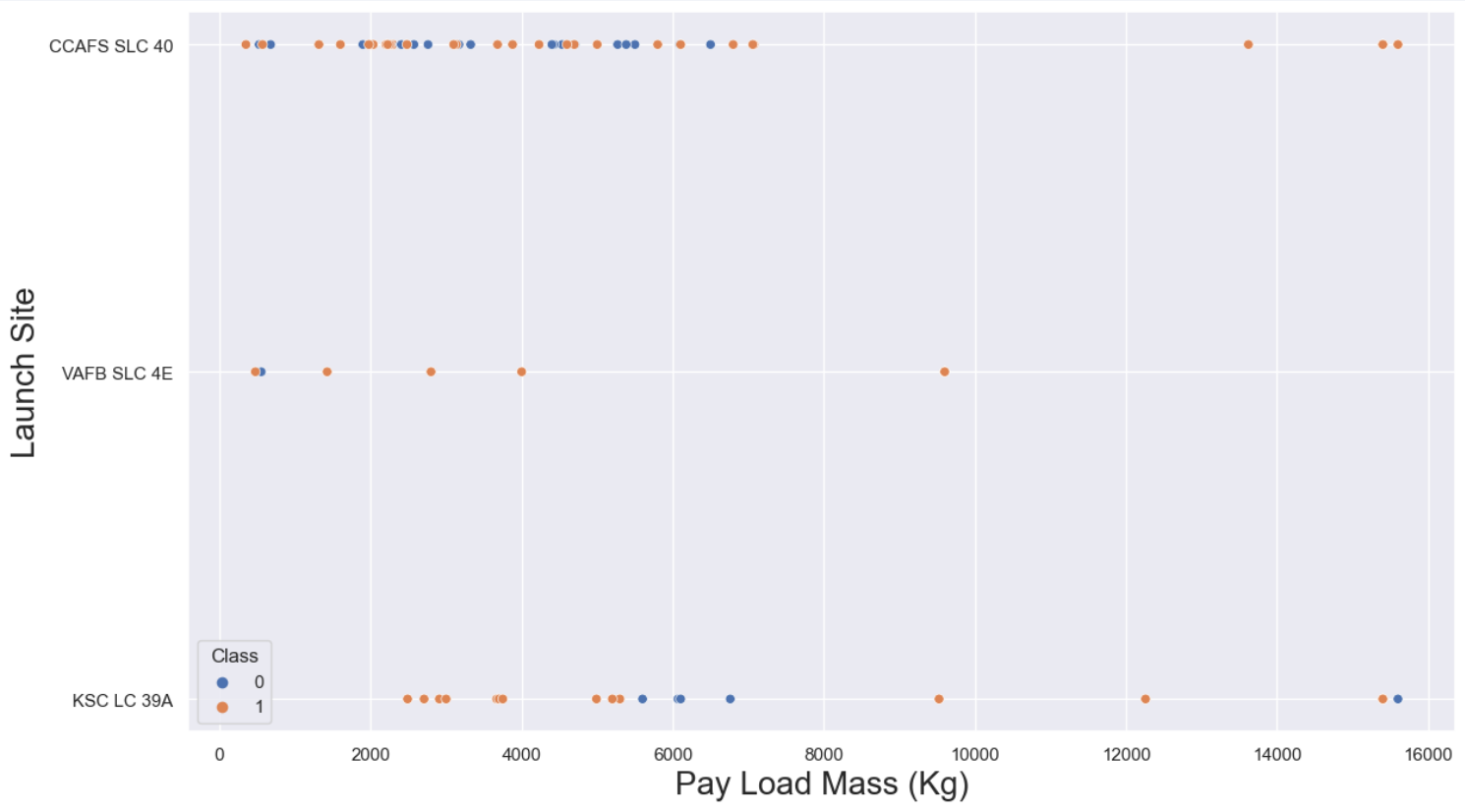

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



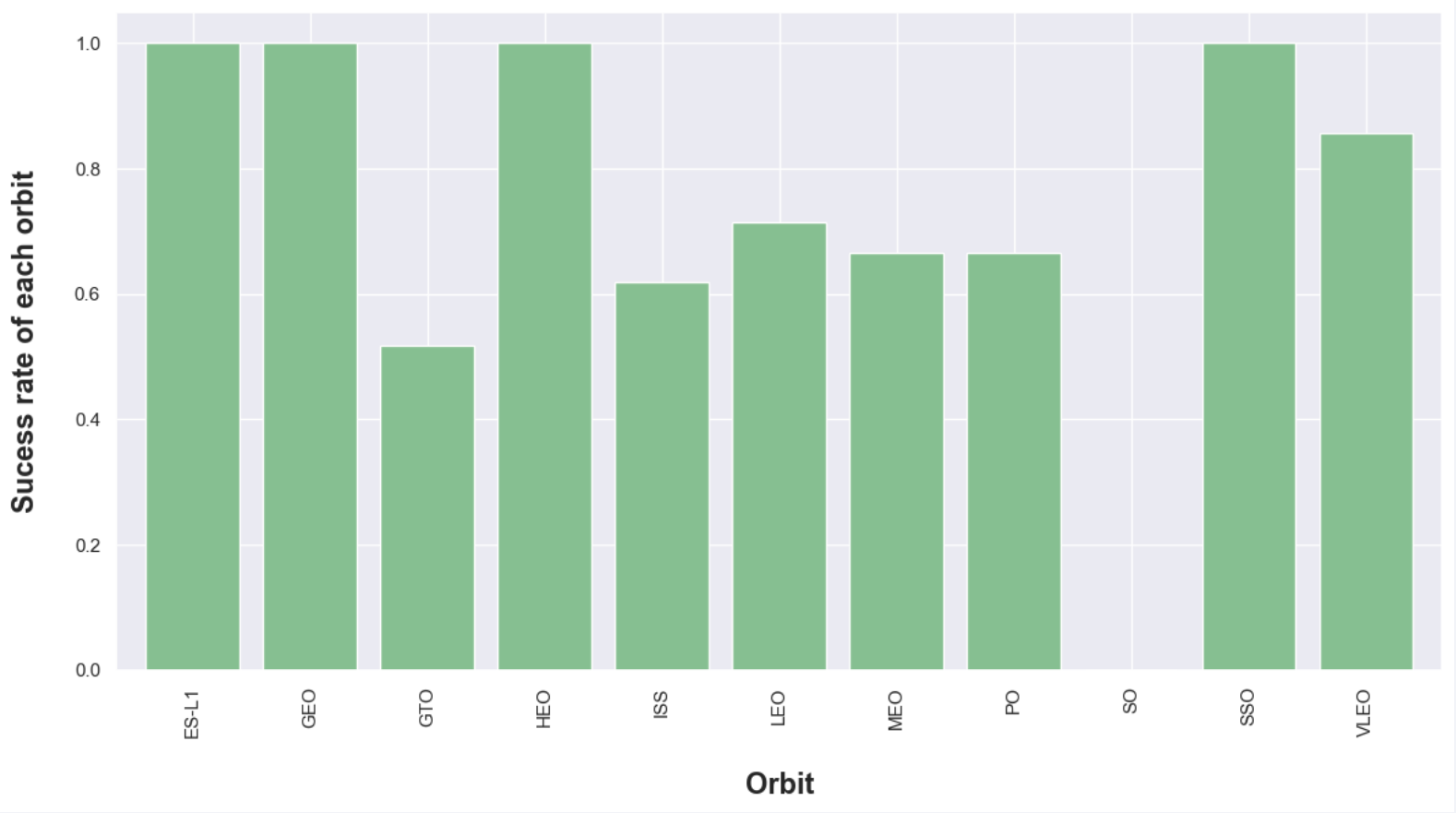
- It can be concluded that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site



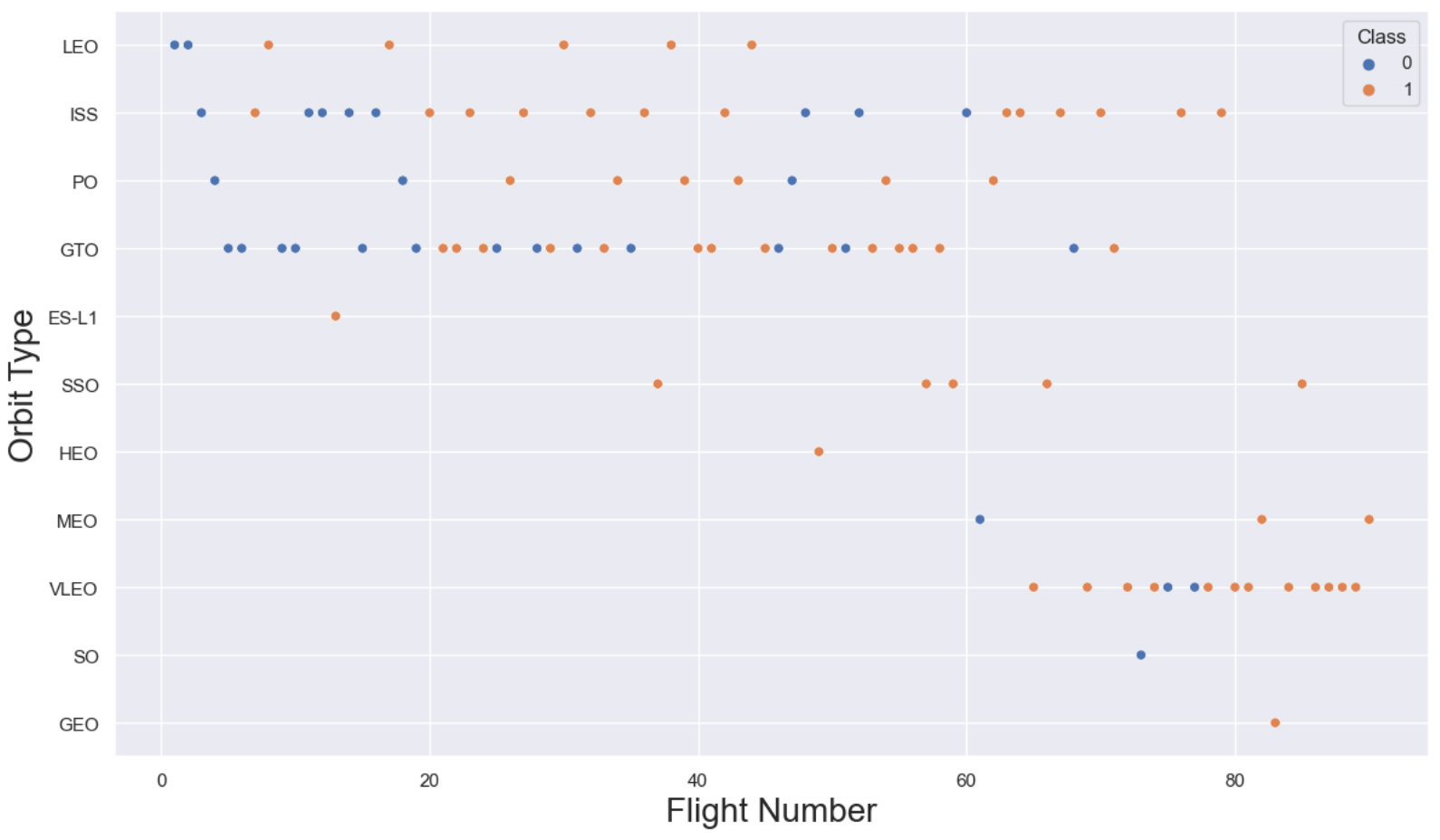
- The plot shows that once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased

Success Rate vs. Orbit Type



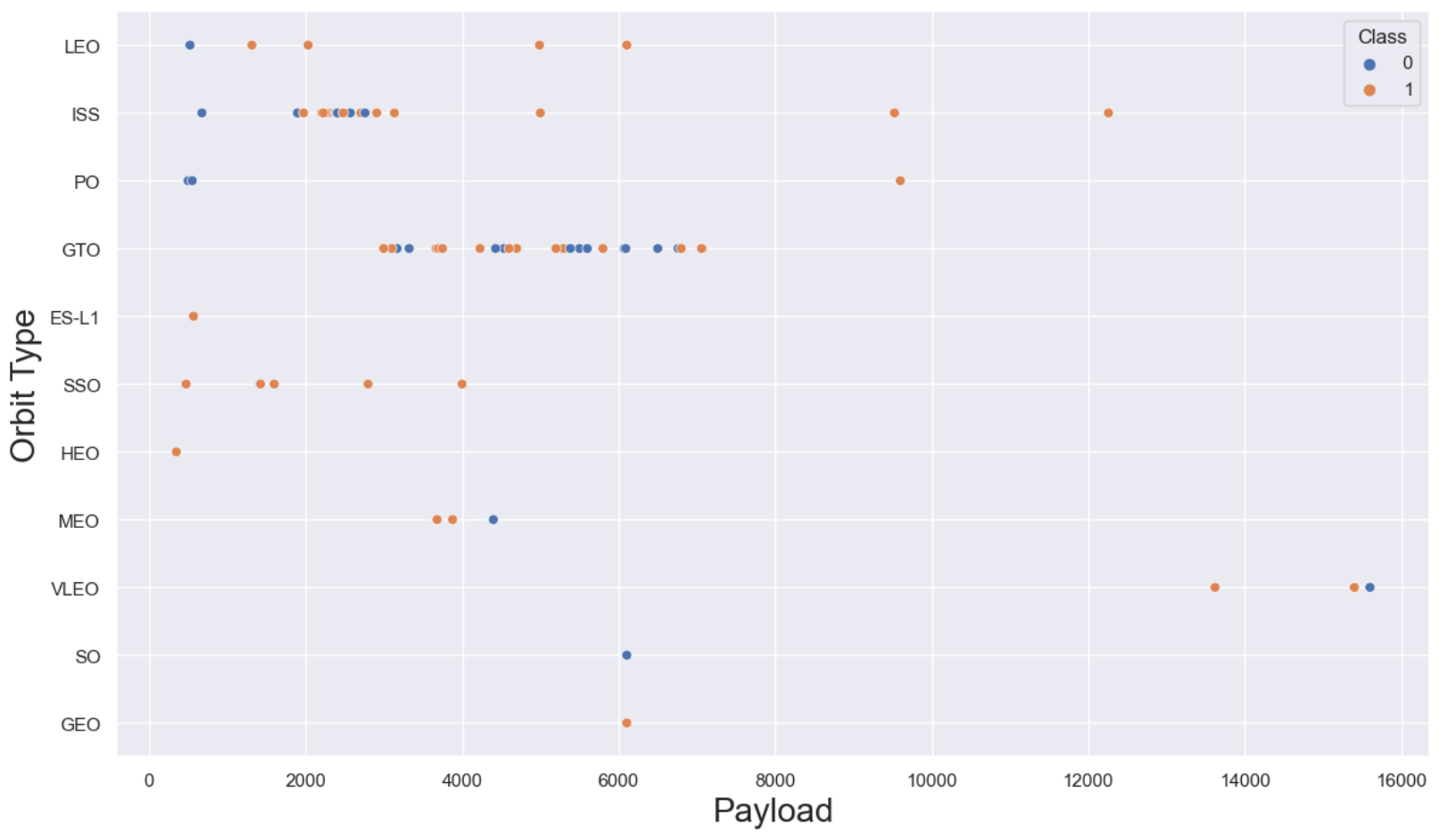
- It can be concluded that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type



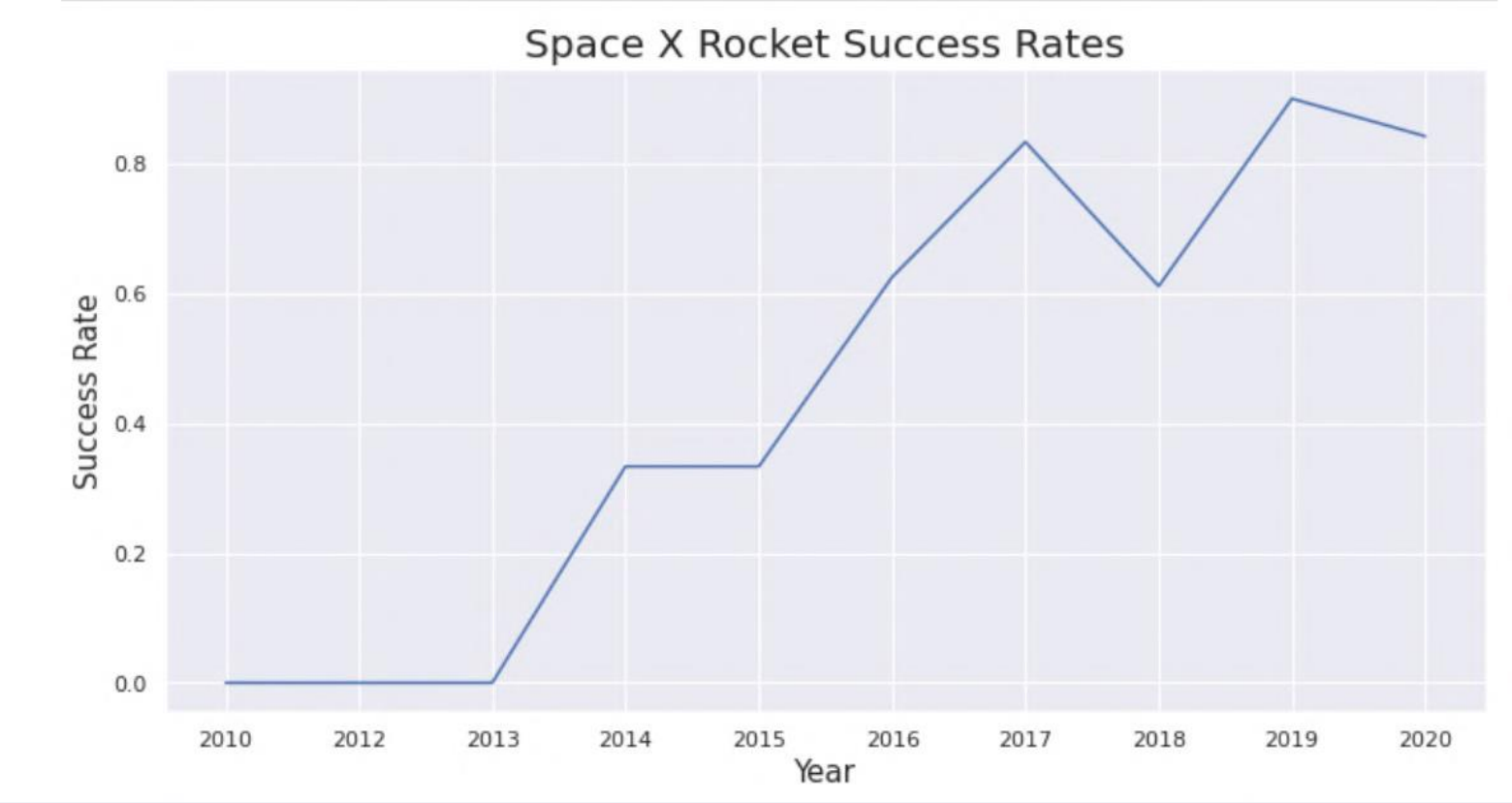
- The success rate typically increases with the number of flights for each orbit
- This relationship is highly apparent for the LEO orbit
- The GTO orbit, however, does not follow this trend

Payload vs. Orbit Type



- Heavy payloads are better with LEO, ISS and PO orbits
- The GTO orbit has mixed success with heavier payloads

Launch Success Yearly Trend



- From the plot, we can observe that success rate since 2013 kept on increasing till 2020

All Launch Site Names

Task 1

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

In [11]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[11]:

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

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

Launch Site Names Begin with 'CCA'

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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

This query is used to display 5 records where launch sites begin with `CCA`

Total Payload Mass

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

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

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

SUM(PAYLOAD_MASS__KG_)
45596

The total payload carried by boosters from NASA is calculated as 45596 using the above query.

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version 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

2,928 kg (average) carried by booster version F9 v1.1

First Successful Ground Landing Date

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

Hint:Use min function

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

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

First Successful Landing
2015-12-22

The dates of the first successful landing outcome on ground pad was 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

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 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

WHERE clauseis used to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

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

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

succesful mission
100

```
%sql SELECT COUNT (MISSION_OUTCOME) AS 'failured mission' FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%'
```

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

failured mission
1

Like ‘%’ is used to filter for WHERE MissionOutcome was a success or a failure.

Boosters Carried Maximum Payload

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

```
%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

The booster that have carried the maximum payload is found using a subquery in the WHERE clause and the MAX() function

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
LANDING_OUTCOME = 'Failure (drone ship)';
```

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

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

A combinations of the WHERE clause, LIKE, AND, and BETWEEN is used to filter failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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.

```
%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 (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

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

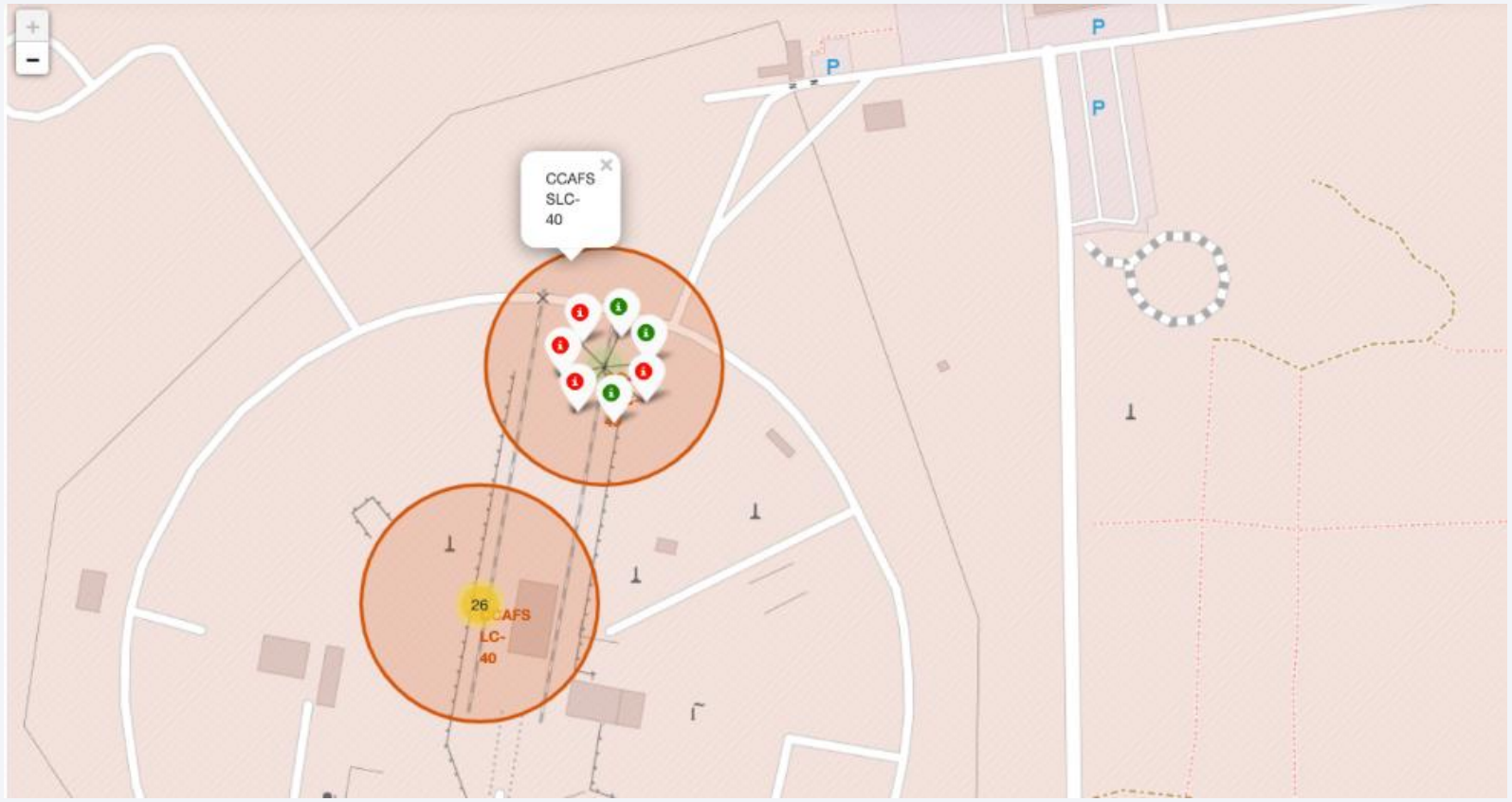
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

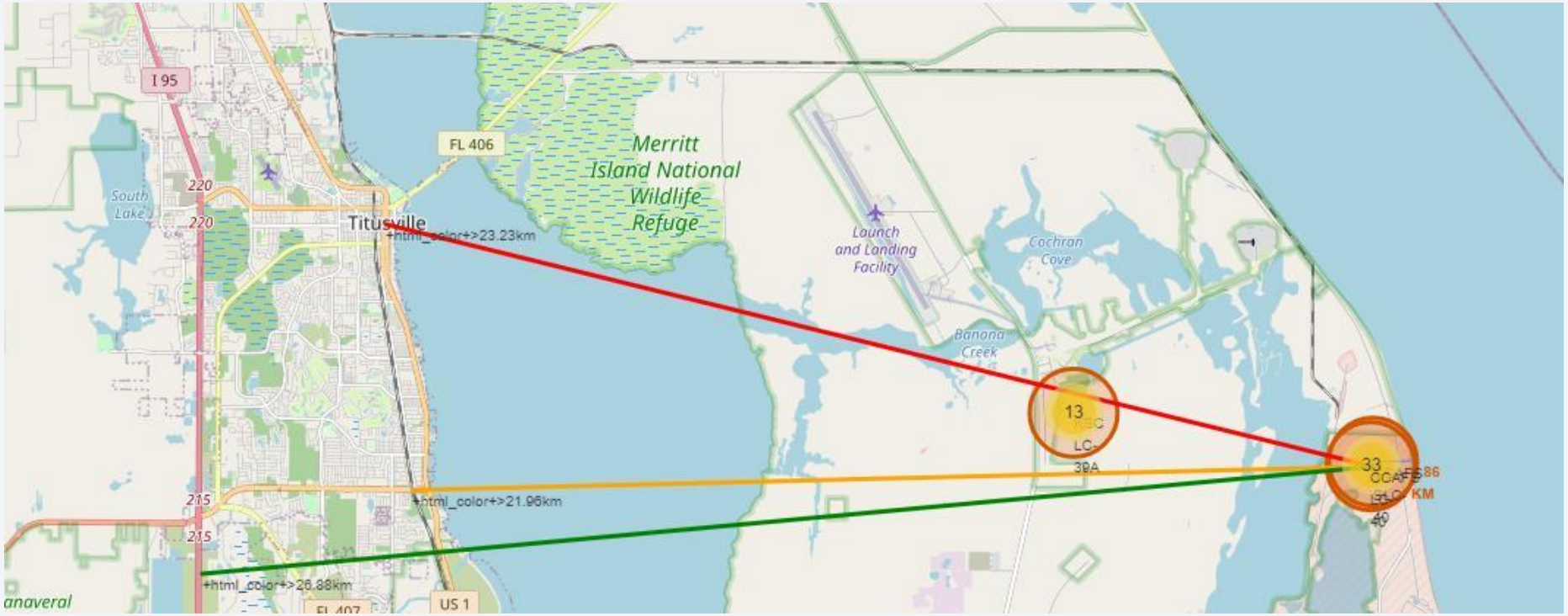
We observe that the closer the launch site to the equator, the easier it is to launch to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit.

<Folium Map Screenshot 2>



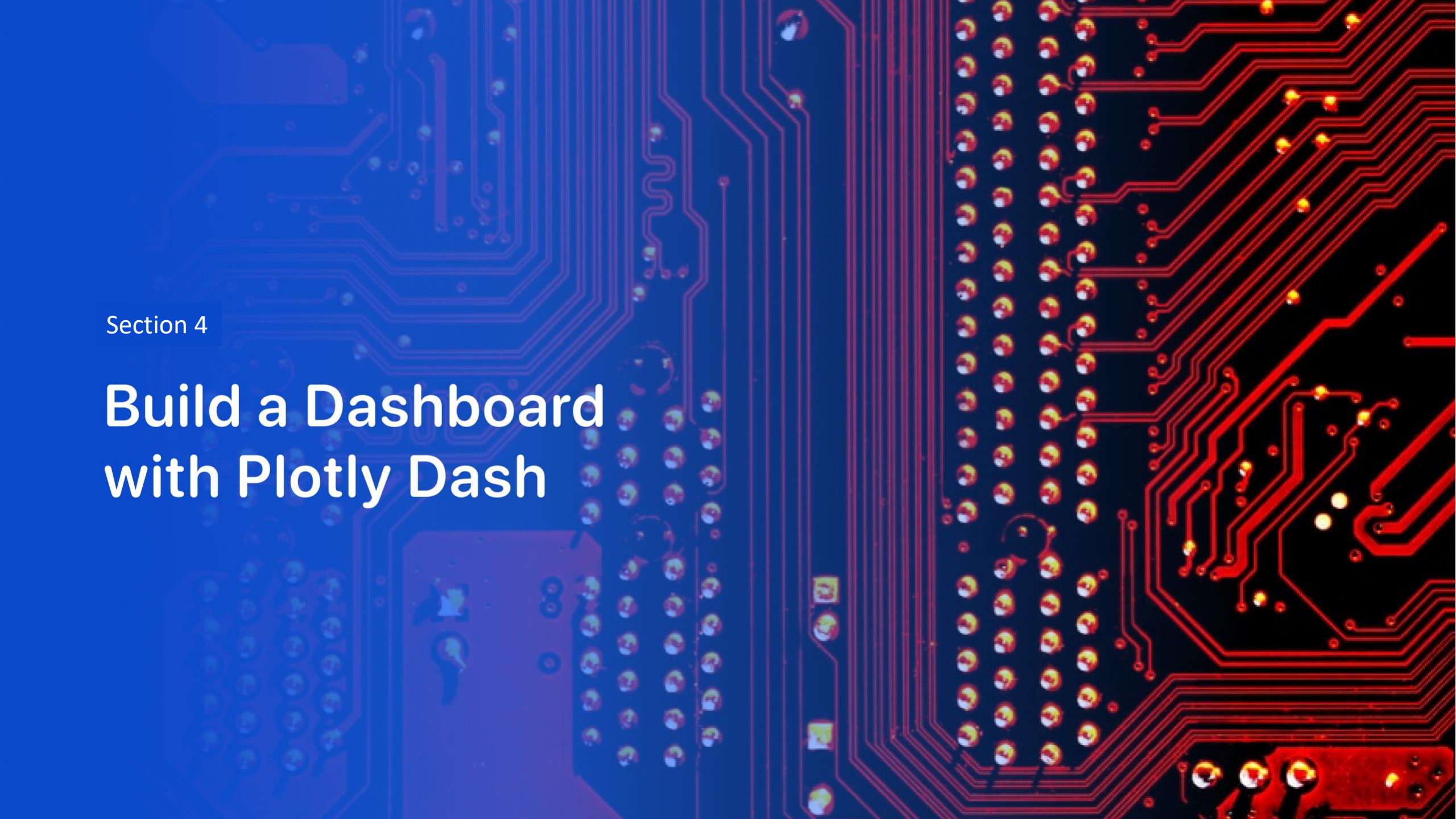
- Green markers for successful launches
- Red markers for unsuccessful launches
- Launch site CCAFS SLC-40 has a 3/7 success rate (42.9%)

<Folium Map Screenshot 3>

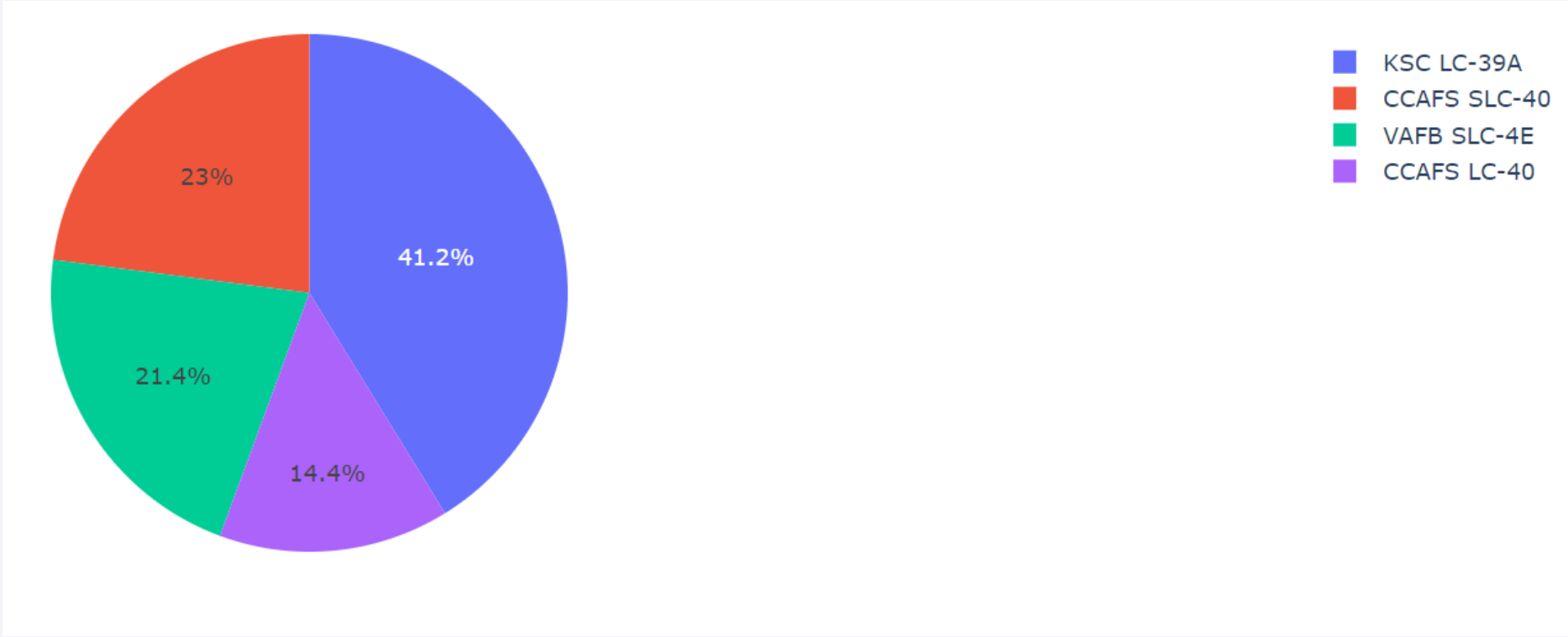


CCAFS SLC-40

- **.86 km** from nearest coastline
- **21.96 km** from nearest railway
- **23.23 km** from nearest city
- **26.88 km** from nearest highway

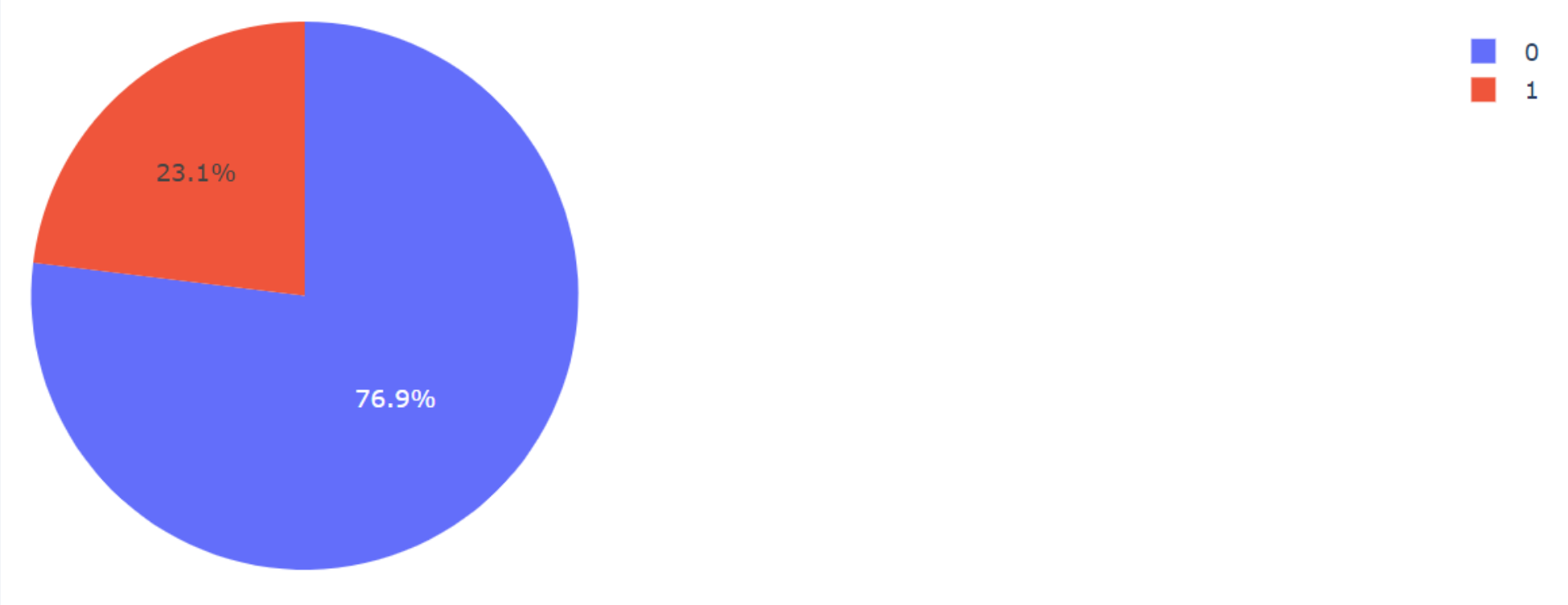

Section 4
Build a Dashboard with Plotly Dash

The Success Rate in each site



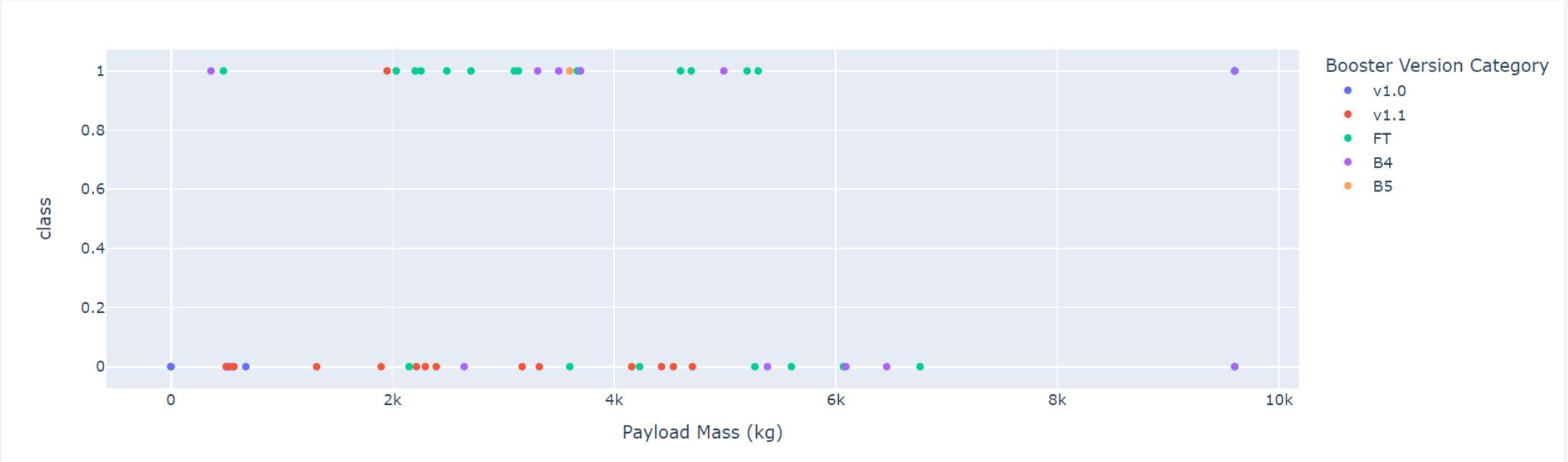
KSC LC-39A has the most successful launches amongst launch sites (41.2%)

The highest launch success ratio



KSC LC-39A has the highest success rate amongst launch sites (76.9%)

Playload mass and success

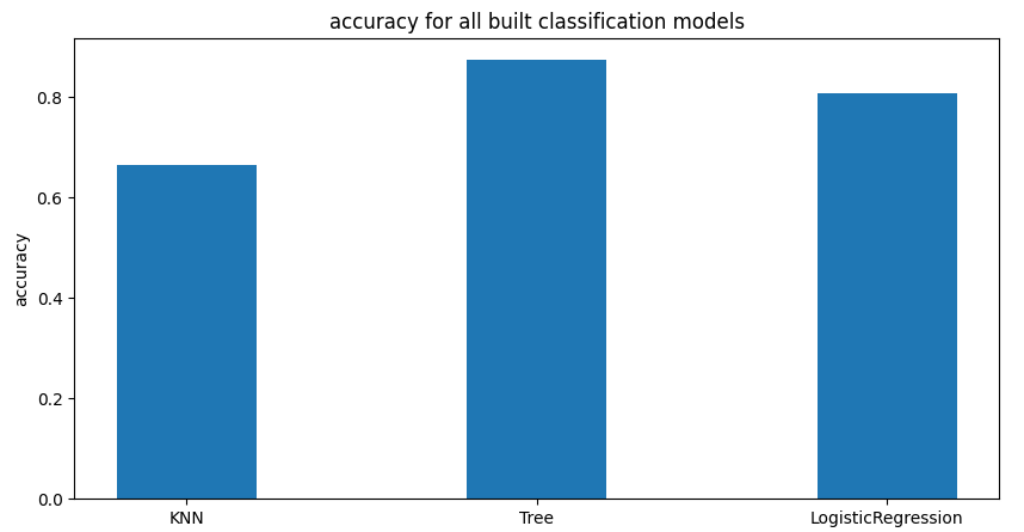


Payloads between 2,000 kg and 5,000 kg have the highest success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy



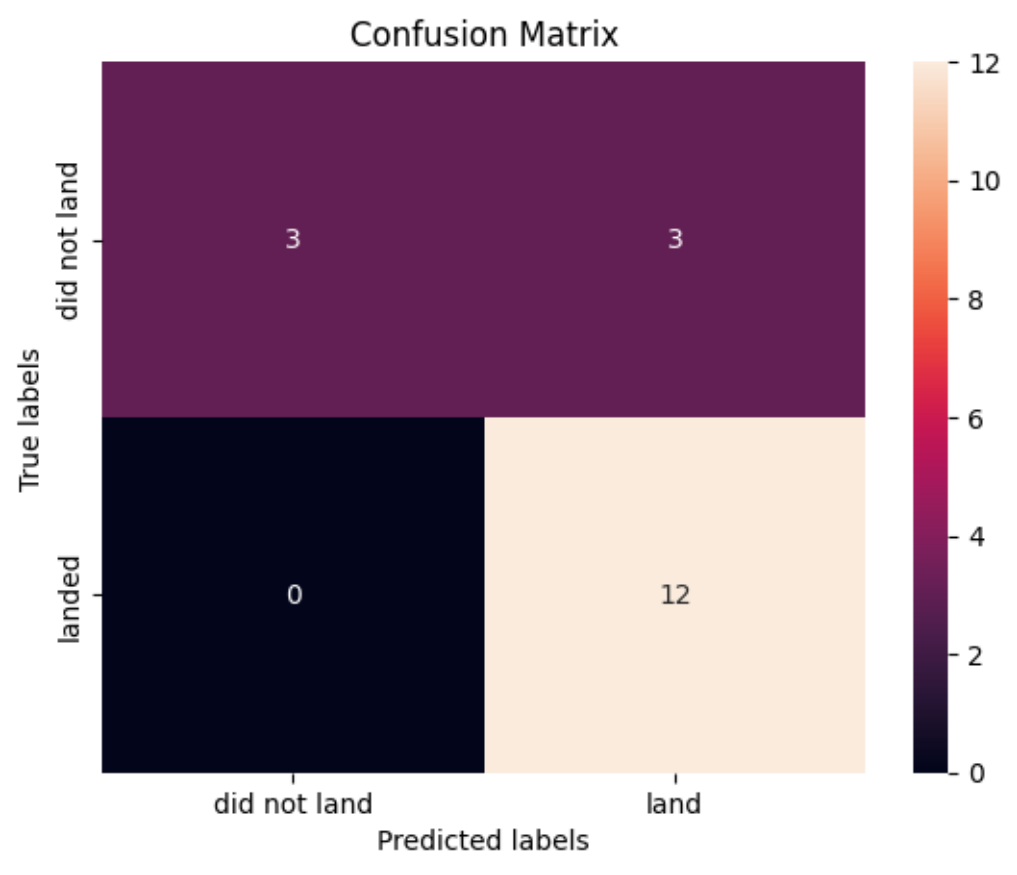
The decision tree classifier is the model with the highest classification accuracy

Find the method performs best:

```
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 0.8732142857142856
Best Params is : {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split':
10, 'splitter': 'best'}
```

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

The following conclusions can be drawn :

- The Decision tree classifier is the best machine learning algorithm for this project
- KSC LC-39A: Has the highest success rate among launch sites.
- Across all launch sites, the higher the payload mass (kg), increase the success rate
- Launch success rate started to increase in 2013 till 2020
- SSO orbit have the most success rate; 100% and more than 1 occurrence

Thank you!