SPACEX

The Use of Data Analytics in the Competitive Market of Space: the Case of SpaceX First Stage Rocket Landing Success Factors

Loso Judijanto

27 December 2022

Table of Contents

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 Results
  - ✓ Interactive Analytics in Screenshots
  - ✓ Predictive Analytics Results

Introduction

- ✓Project background and context
  - ✓ Space X claims that 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 Space X can reuse the first stage. Therefore, if it can be determined that if the first stage will land, the cost of a launch can be calculated. This information is very useful for a competing company which wants to bid against Space X for a rocket launch. This goal of the project is to create a machine learning to predict if the first stage will land successfully.
- ✓Business problems to solve:
  - ✓ What factors are needed to determine that the rocket will land successfully?
  - ✓ What is the interaction amongst various features that determine the success rate of a successful landing?
  - ✓ What operating conditions are needed to be in place to ensure a successful landing program?

Section 1: Methodology

Methodology

- ✓Executive Summary
- ✓Data collection methodology:
  - ✓Data was collected using SpaceX API and web scraping from Wikipedia.
- ✓Performing data wrangling
  - ✓One-hot encoding is applied to categorical features
- ✓Performing Exploratory Data Analysis (EDA) using visualization and SQL
- ✓Performing interactive visual analytics using Folium and Plotly Dash
- ✓Performing predictive analysis using classification models
  - ✓How to build, tune, and evaluate classification models

Data Collection

- ✓The data is collected using various methods as follows:
  - ✓ Data collection is done using get request to the SpaceX API.
  - ✓ Next, the response content is decoded as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - ✓ The data is cleaned, checked for missing values and filled in missing values where necessary.
  - ✓ Web scraping is performed from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - ✓ The main objective is to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for the further analysis.

Data Collection – SpaceX API

✓The get request to the SpaceX API is used to collect data, then to clean the requested data and to do some basic data wrangling and formatting.

✓The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scrapping

- The web scrapping technique is used to collect Falcon 9 launch records with BeautifulSoup from Wikipedia
- The table is parsed and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/jupyter-labs-webscraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

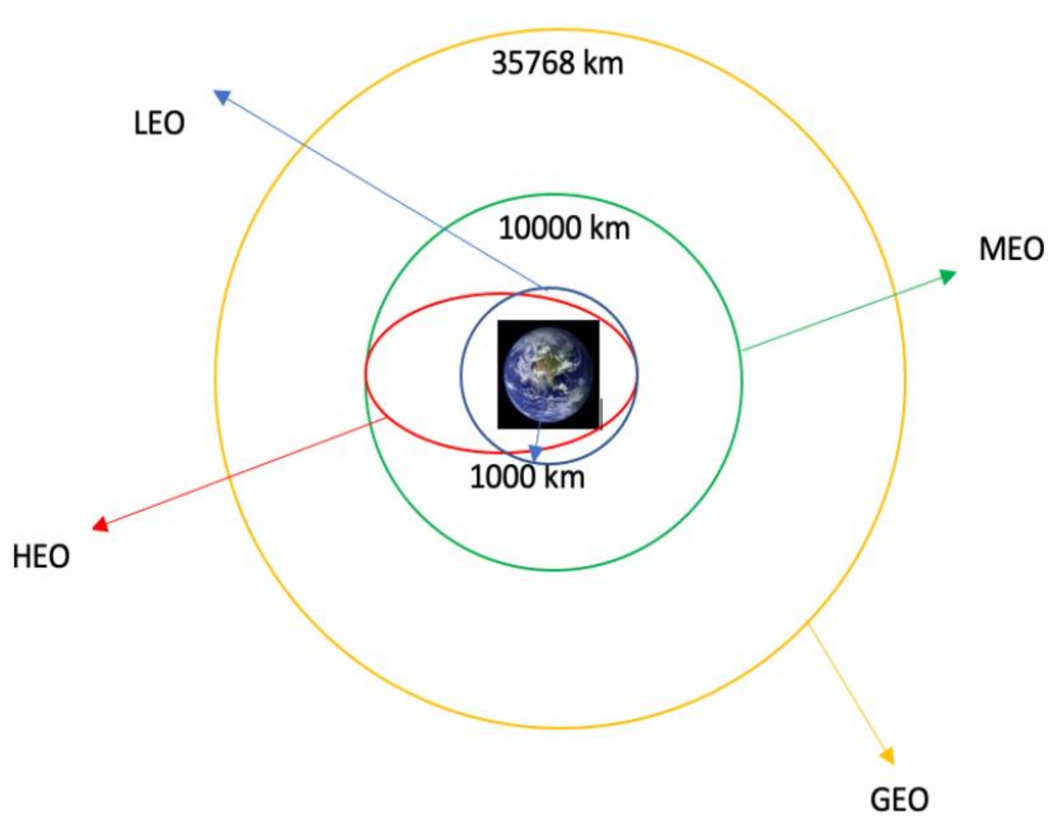
3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

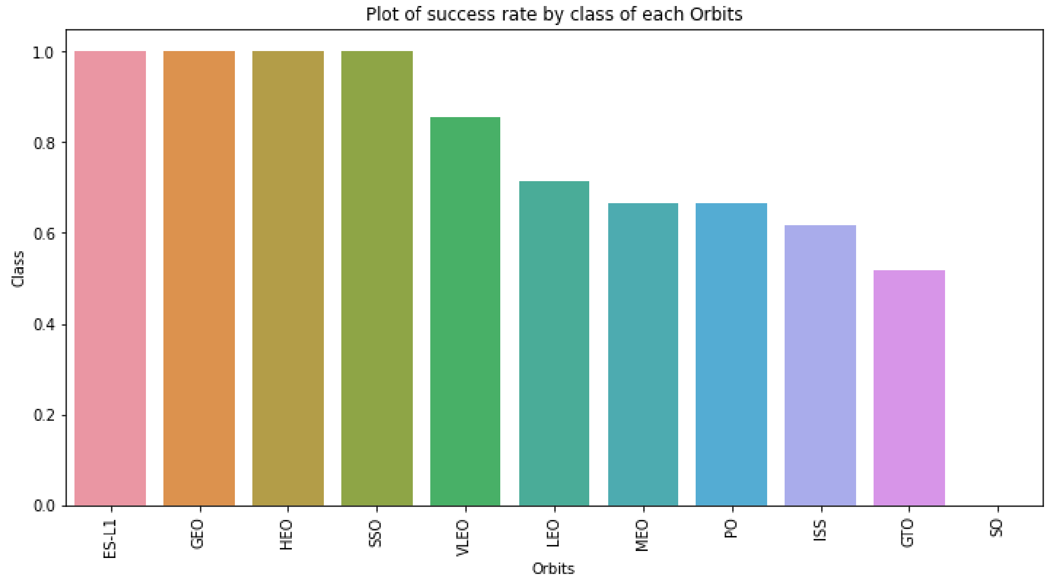
Data Wrangling

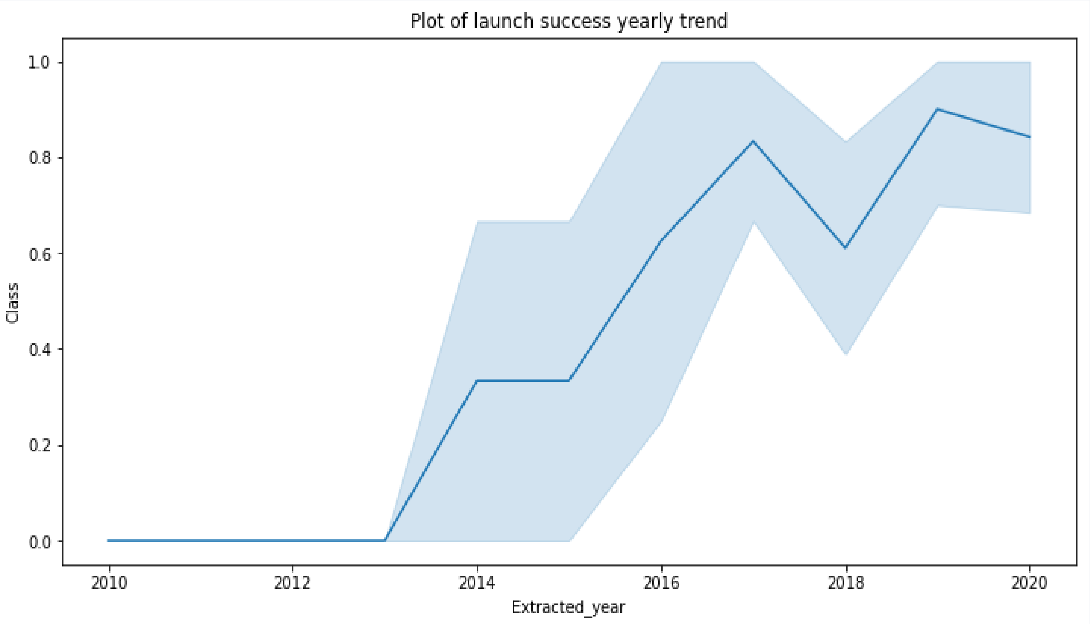
- ✓Exploratory data analysis is conducted and is used to determine the training labels.
- ✓The number of launches is calculated at each site, and the number and occurrence of each orbits
- ✓Landing outcome is created to label from outcome column and the results are exported to csv.
- ✓The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



EDA with Data Visualization

The data is explored by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- ✓The SpaceX dataset is loaded into a PostgreSQL database without leaving the jupyter notebook.
- ✓EDA with SQL is applied to get insights from the data. The queries are written to find out, among others:
  - ✓ 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.
- ✓The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Building an Interactive Map with Folium

- ✓ All launch sites are marked, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- ✓ The feature launch outcomes (failure or success) are assigned to class 0 and 1.i.e., 0 for failure, and 1 for success.
- ✓ The color-labeled marker clusters are used to identify which launch sites have relatively high success rate.
- ✓ The distances between a launch site to its proximities are calculated to answer some questions likes:
  - ✓ Are launch sites near railways, highways and coastlines?
  - ✓ Do launch sites keep certain distance away from cities?
- ✓ The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Building a Dashboard with Plotly Dash

- ✓ An interactive dashboard is built with Plotly Dash.
- ✓ Pie charts showing the total launches by a certain sites are created.
- ✓ Scatter graphs showing the relationship with Outcome and Payload Mass (Kg) for the different booster version are created.

Predictive Analysis (Classification)

- ✓The data is loaded using numpy and pandas, transformed, and then split into training and testing.
- ✓Different machine learning models are built and tuned using different hyperparameters using GridSearchCV.
- ✓Accuracy is used as the metric for the model, and then the model is improved using feature engineering and algorithm tuning.
- ✓The best performing classification model is selected using the criteria.
- ✓The link to the notebook is https://github.com/losojudijanto/IBM-Capstone-Project-SpaceX/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

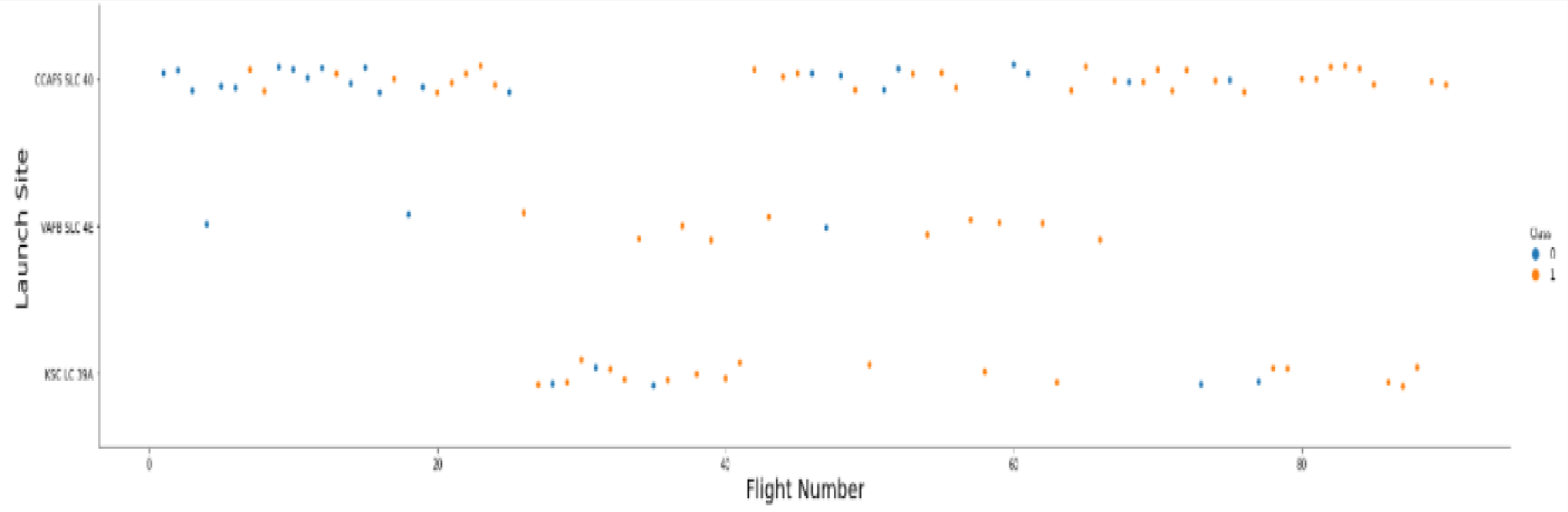
The Results

- ✓Exploratory Data Analysis results
- ✓Interactive Analytics demo in screenshots
- ✓Predictive Analysis results

Section 2:
Insights from Exploratory Data Analysis

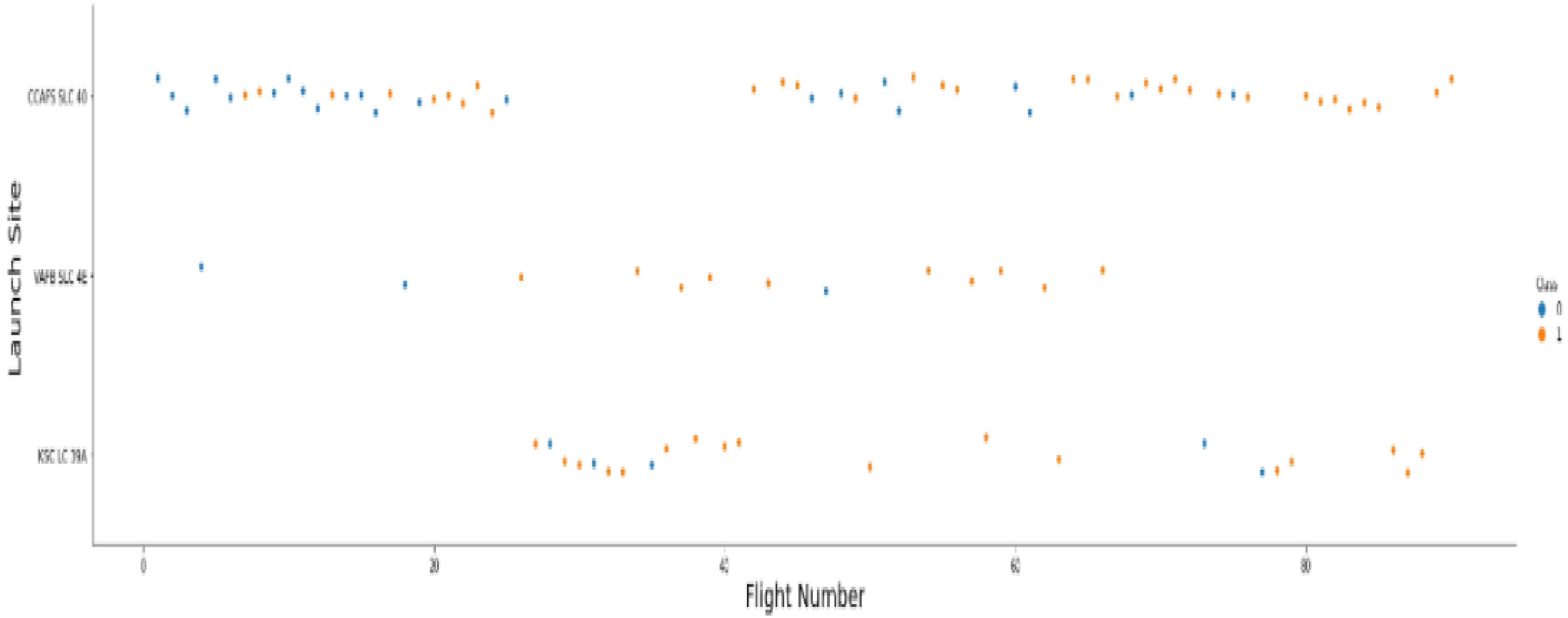
Flight Number and Launch Site

From the scatter plot below it can be recognized that the larger the flight amount at a launch site, the greater the success rate at a launch site.



Payload and Launch Site

From the scatter plot below it can be recognized that the greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket

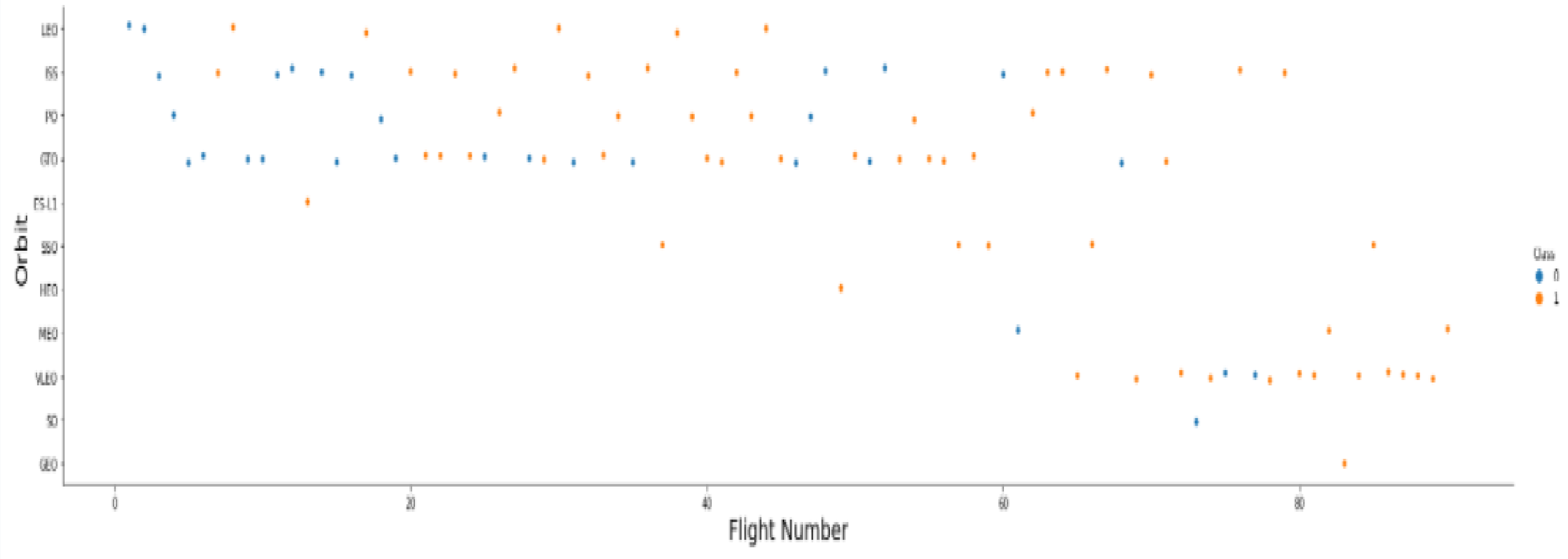


Success Rate and Orbit Type

From the plot below it can recognized that ES-L1, GEO, HEO, SSO, VLEO all have the most success rate

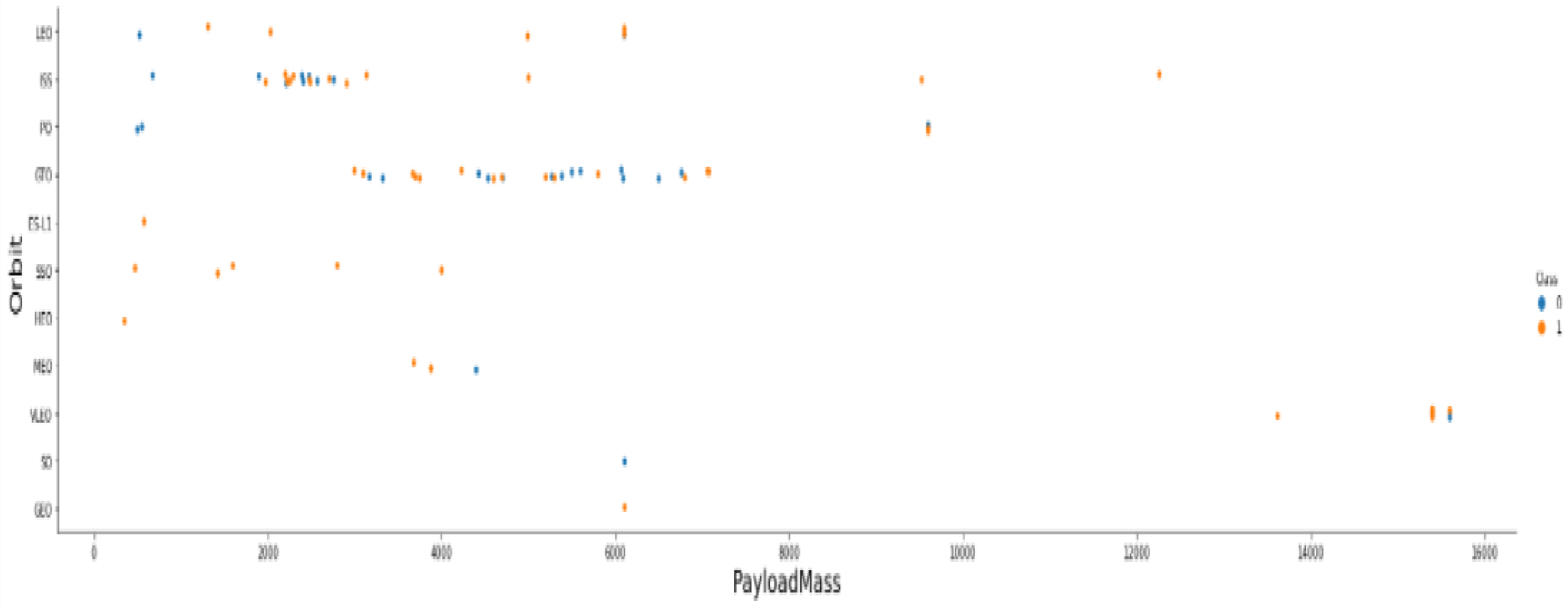
Flight Number and Orbit Type

The plot below shows the relationship between the Flight Number and Orbit type. It can be observed 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.



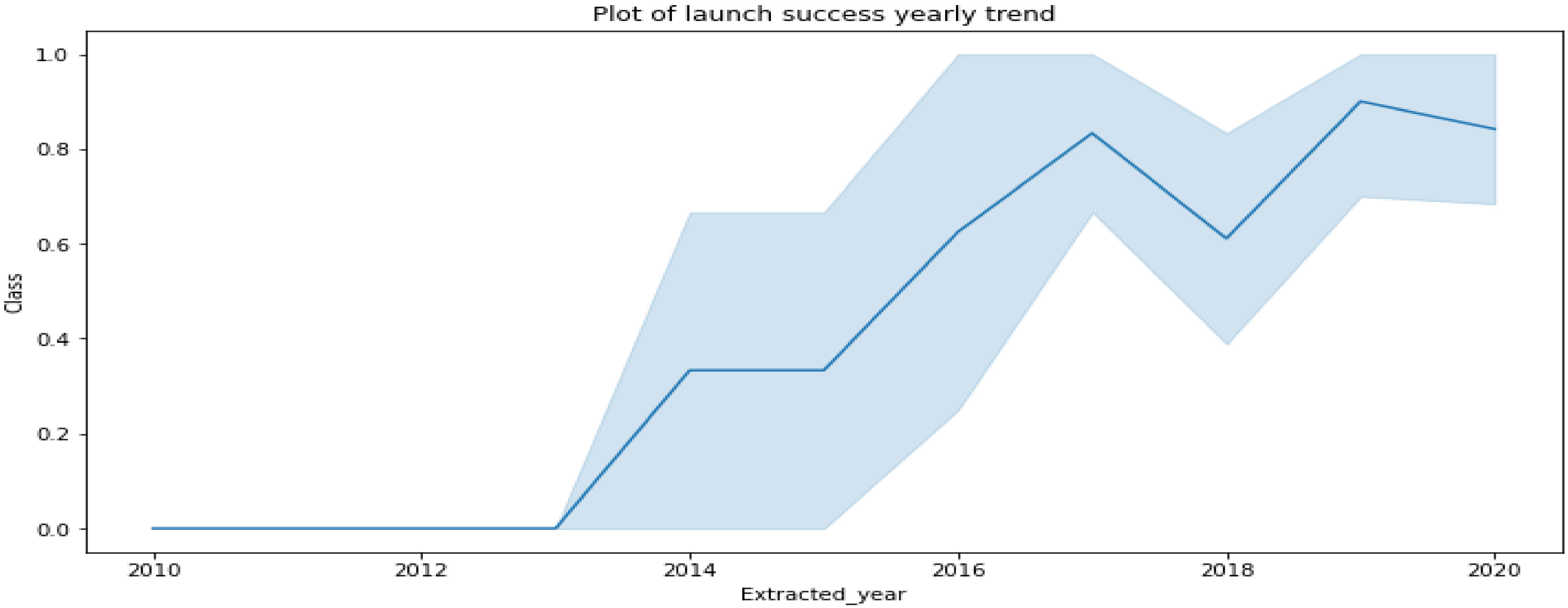
Payload and Orbit Type

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



Launch Success Yearly Trend

From the plot below it can recognized that the success rate since 2013 tends to keep on increasing until 2020.



All Launch Site Names

The key word **DISTINCT** is applied to filter out only unique launch sites from the SpaceX data.

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

In [10]:

```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

The query below is applied to display 5 records where launch sites begin with `CCA`

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

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	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 as 45596 is calculated using the query below.

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

In [12]:

```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass of Rocket Falcon 9 v1.1

The average payload mass carried by booster version F9 v1.1 is calculated and the result is 2928.4 kgs.

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

In [13]:

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

It can be observed that the dates of the first successful landing outcome on ground pad is 22nd December 2015.

In [14]:
```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000 kgs

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

In [15]:

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

The wildcard such as '%' is used to filter for **WHERE** Mission Outcome is a success or a failure.

List the total number of successful and failure mission outcomes

In [16]:

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Boosters Carried Maximum Payload

The booster that have carried the maximum payload can be determined using a subquery in the **WHERE** clause and the **MAX()** function.

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

In [17]:

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

The combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions are applied to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015.

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

In [18]:

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- ✓The landing outcomes and the **COUNT** of landing outcomes are selected from the data and the **WHERE** clause is applied to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- ✓The **GROUP BY** clause is applied to group the landing outcomes and the **ORDER BY** clause is used to order the grouped landing outcome in descending order.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

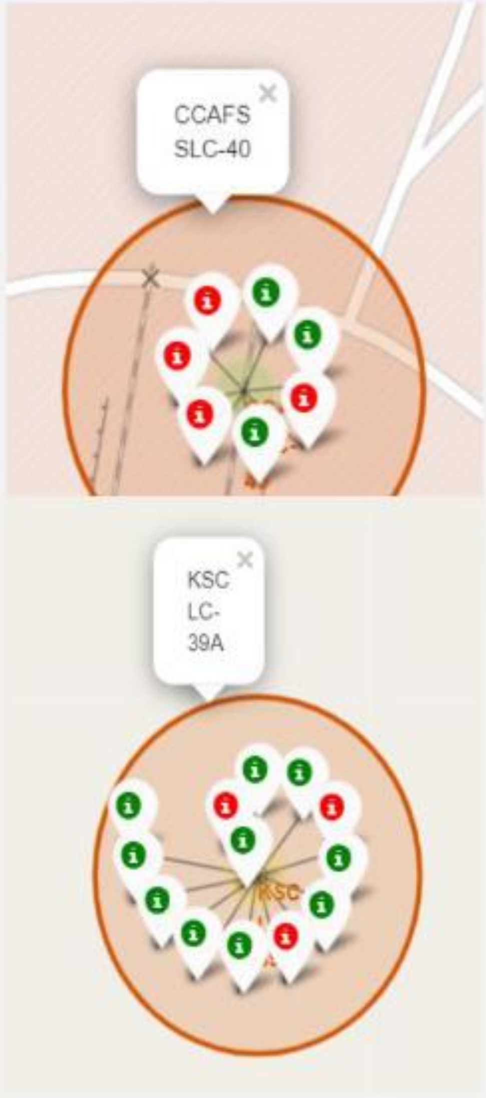
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

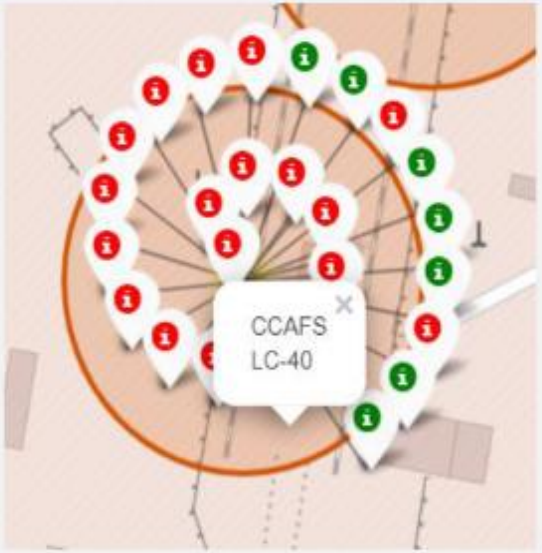
Section 3: Launch Sites Proximities Analysis

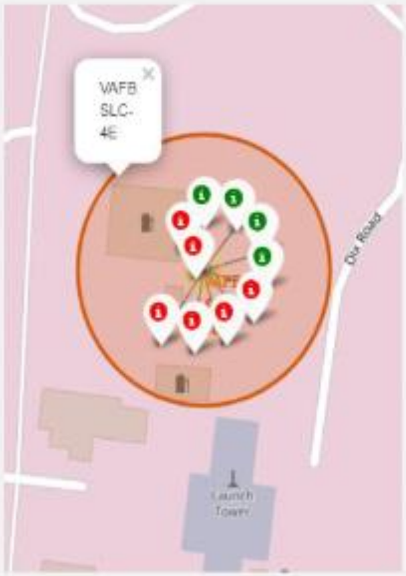
All Launch Sites Global Map Markers

It can be observed that the SpaceX launch sites are in the United States of America coasts: Florida and California.

Markers Showing Launch Sites with Color Labels







California Launch Site

Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

Launch Site Distance to Certain Landmarks

•Are launch sites in close proximity to railways? No

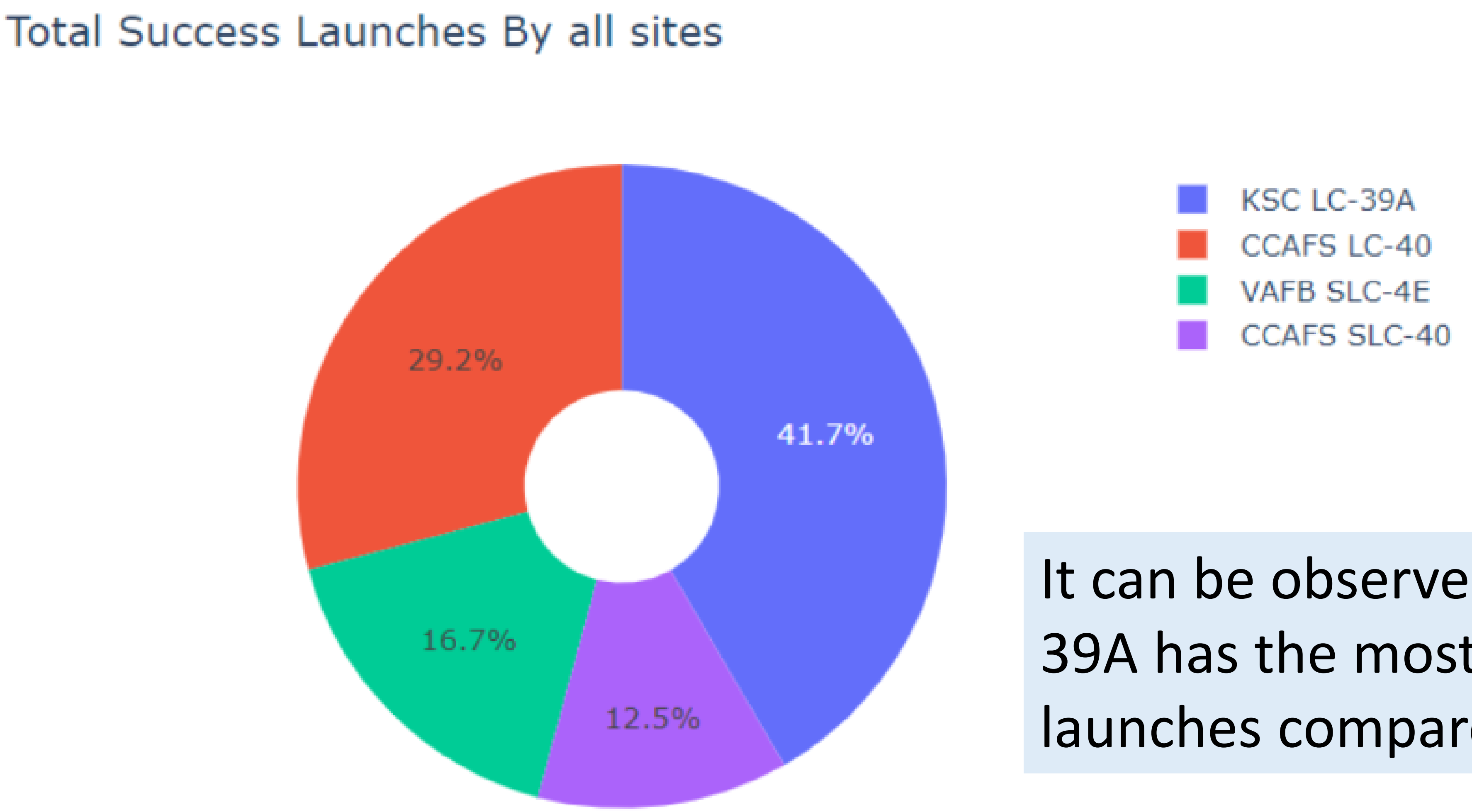
•Are launch sites in close proximity to highways? No

•Are launch sites in close proximity to coastline? Yes

•Do launch sites keep certain distance away from cities? Yes

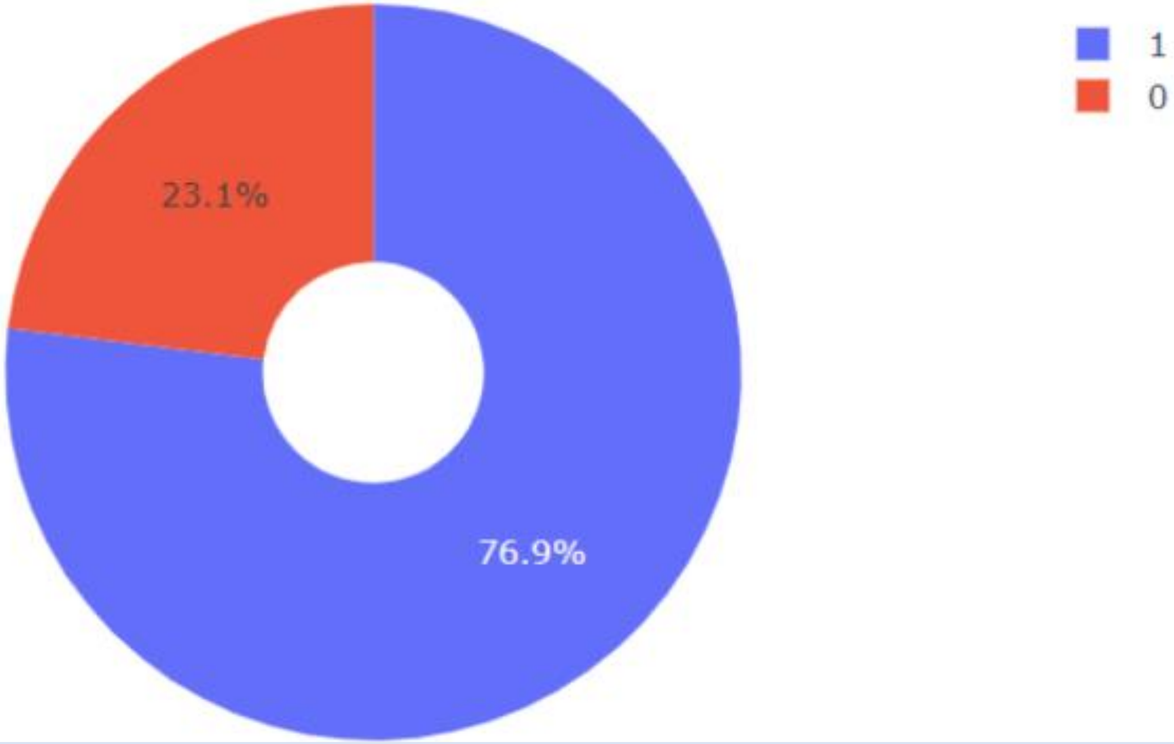
Section 4: Building a Dashboard with Plotly Dash

Pie Chart: The Success Percentage by Each Launch Site



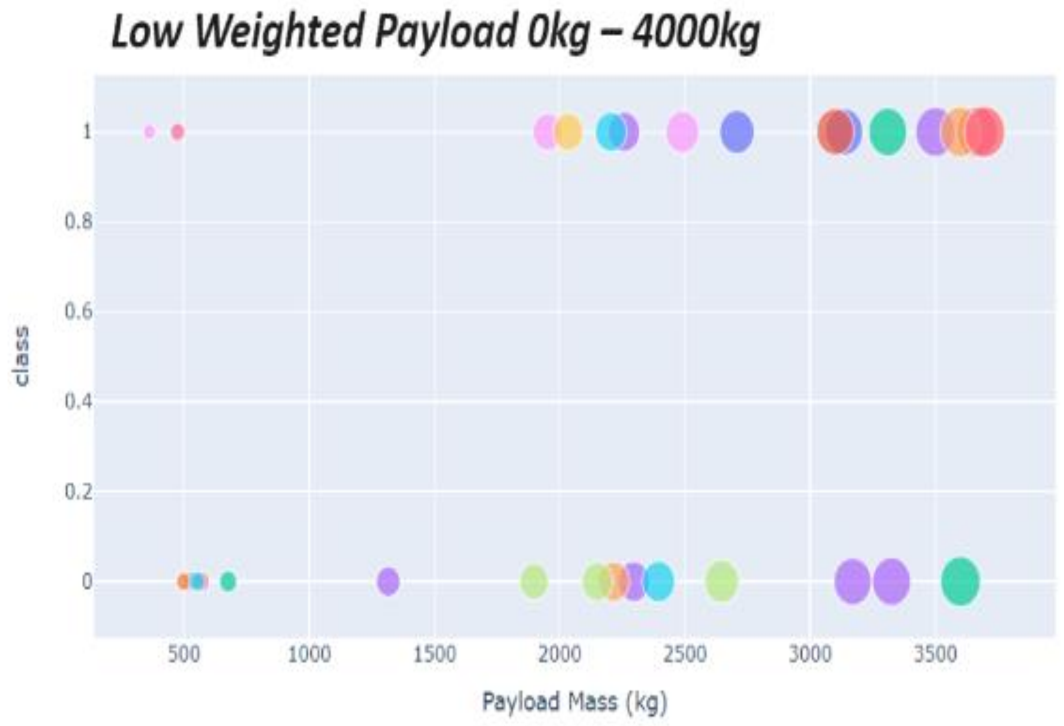
It can be observed that KSC LC-39A has the most successful launches compared to all the sites.

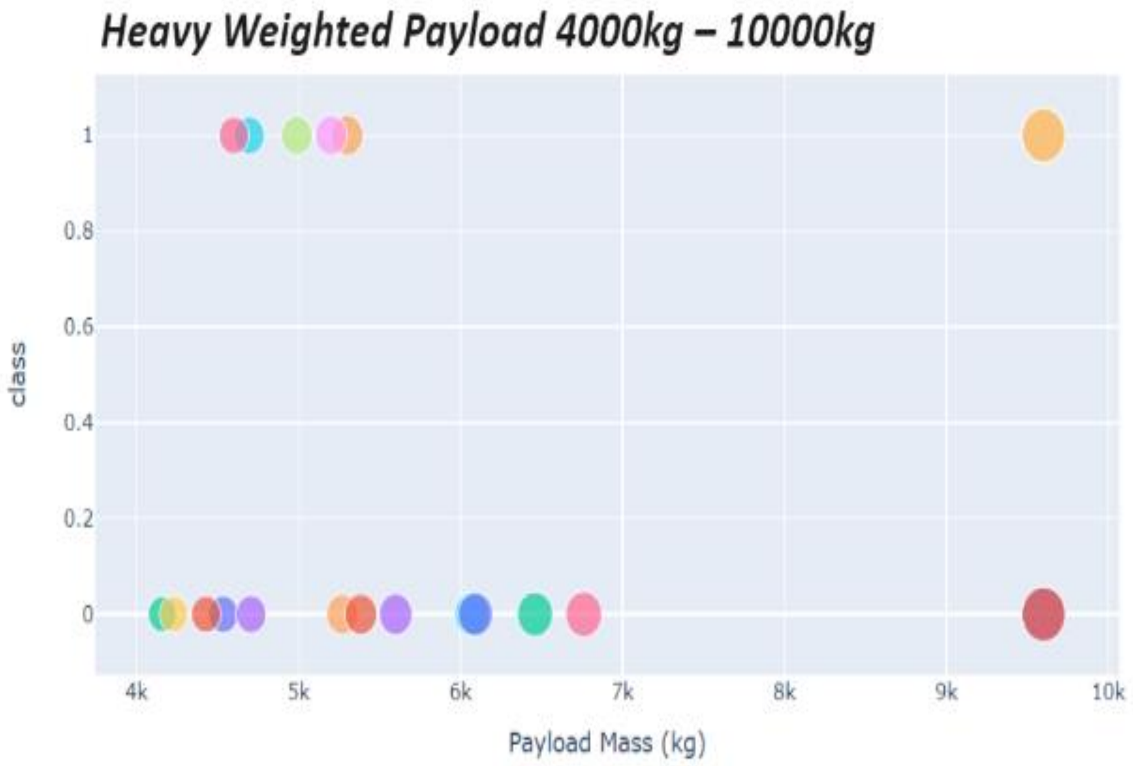
Pie Chart: The Launch Site Success Ratio



KSC LC 39A has achieved a record of 76.9% success rate and of 23.1% failure rate

Scatter Plot of Payload and Launch Outcome for All Sites, with Different Payload in the Range Slider





It can be observed that the success rates for the low-weighted payloads are relatively higher than that of heavy-weighted payloads.

Section 5: Predictive Analysis (Classification)

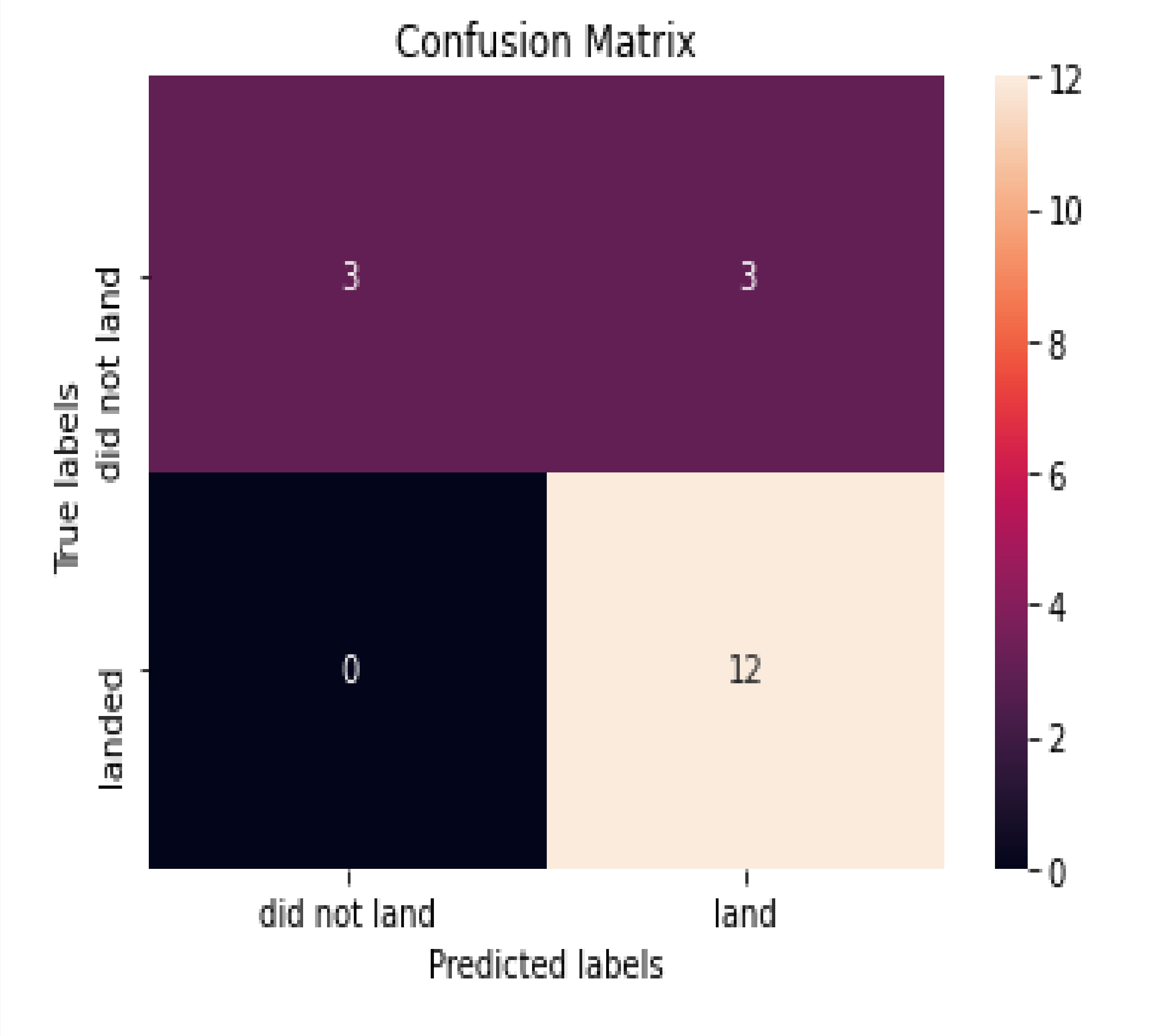
Classification Accuracy

The decision tree classifier results the model with the highest classification accuracy with a score of 0.87.

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes properly.

The problem is only the false positives cases in which unsuccessful landing marked as successful landing by the classifier.

Conclusions

Based on the previous results and analysis, it can be concluded that:

- 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 and tend to increase to 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO have the most success rate.
- KSC LC-39A has the most successful launches of any sites.
- The Decision Tree classifier is the best machine learning algorithm for this classification task in this case.

Thank You!