IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rivira Lisbel Adrian
April 5th 2023

Outline

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

Executive Summary

- Summary of methodologies

1. Collecting Data
2. Data Wrangling
3. EDA(Exploratory Data Analysis)
4. Visual Analytics
5. Predictive Analysis

- Conclusion, Falcon 9 rockets have a very good probabilities of success first stage landing

Introduction

- Space X is the most successful company in the commercial space age and one of the reason for that is because the rocket launches are really inexpensive. A normal rocket launch cost around 165 millions dollars, instead Falcon 9 rocket launch cost about 62 million dollars.
- The main advantage of Space X is that they can reuse the first stage, so, if we can determine the success of first stage landing, we can measure Space X's advantage over other companies.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
  - Clean the requested data
- 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

1. Start **requesting** rocket launch data from SpaceX API with the URL: https://api.spacexdata.com/v4/launches/past

2. Decode the response content as a Json using **.json()** and turn it into a Pandas dataframe using **.json_normalize()**

3. take a subset of our data frame keeping only the features we want

4. apply **get** function method to BoosterVersion, LaunchSite, PayloadData and CoreData

5. combine the columns into a dictionary

6. create a Pandas data frame from the dictionary

7. Filter the dataframe to only include Falcon 9 launches and reset the FlgihtNumber column

8. **Use .**isnull()**.**sum() for see the rows with missing values in our dataset.

9. Calculate the mean for the PayloadMass using the .mean(). Then use the mean and the .replace() function to replace np.nan values in the data with the mean you calculated.

- Data Collection Git Hub Link Lab1

Data Collection - Scraping

1. Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response
2. Create a BeautifulSoup object from the HTML response
3. Print the page title to verify if the BeautifulSoup object was created properly
4. *Use the find_all function in the BeautifulSoup object, with element type `table`.Assign the result to a list called `html_tables`*
5. *Apply find_all() function with `th` element on first_launch_table*
6. *Iterate each th element and apply the provided extract_column_from_header() to get a column name*
7. *Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names*
8. Create an empty dictionary with keys from the extracted column names
9. Fill up the launch_dict with launch records extracted from table rows
10. Creat a Data Frame form launch_dic

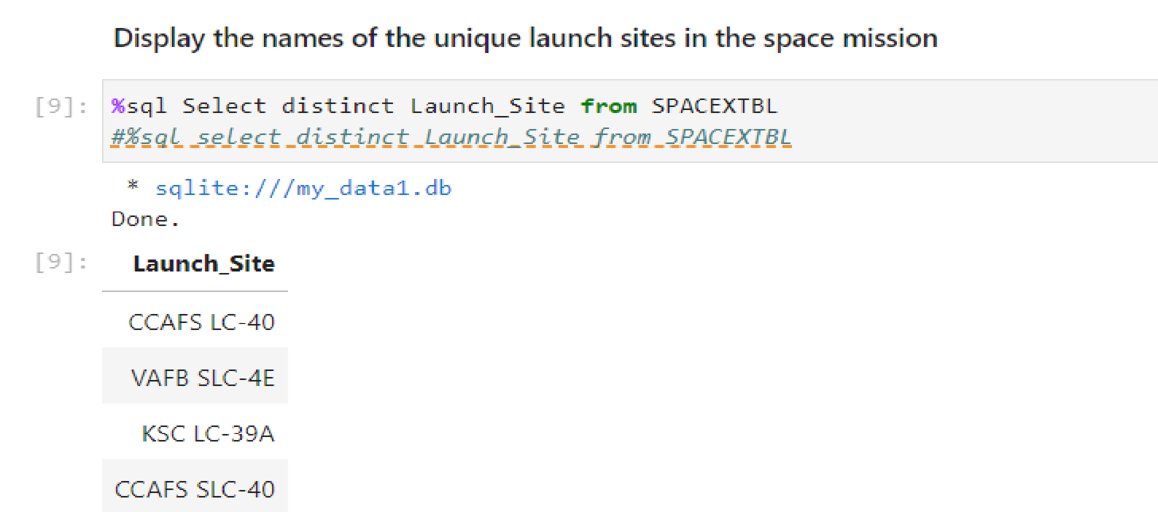
- Data Collectiomn Git Hub Link Lab 2

Data Wrangling

1. The data contains several Space X launch facilities , so we use the method **value_counts()** on the column LaunchSite to determine the number of launches on each site
2. Create a set of outcomes where the second stage did not land successfully
3. create a set of outcomes where the second stage did not land successfully
4. This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully
5. use the method **.mean()** to determine **the success rate, which is 0.66**

- Data Wranglin GitHub link Lab 1

EA with SQL



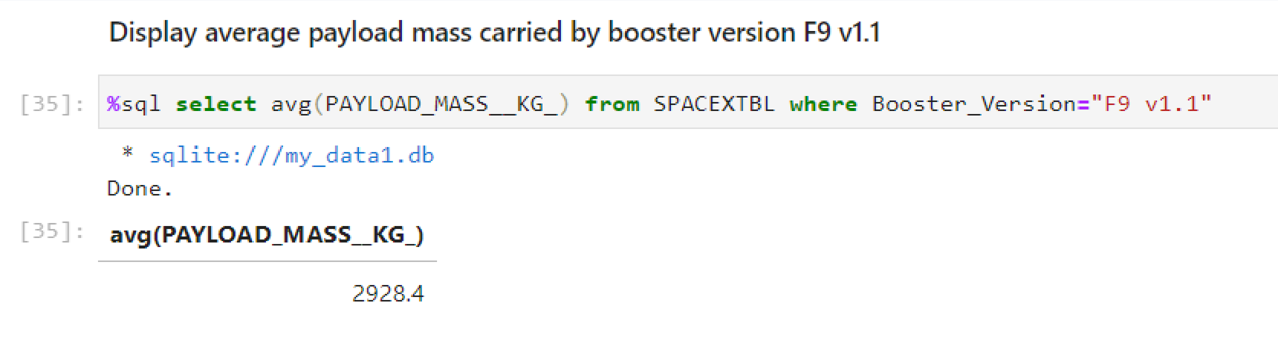
Display the names of the unique launch sites in the space mission

```
[9]: %sql Select distinct Launch_Site from SPACEXTBL
     #%sql select distinct Launch_Site from SPACEXTBL
```

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

[9]:

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



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

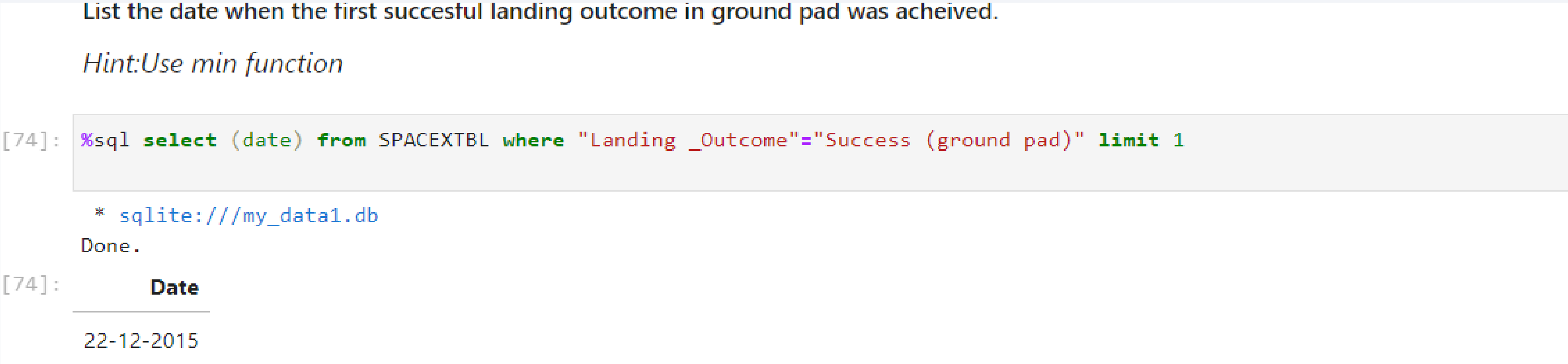
```
[35]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version="F9 v1.1"
```

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

[35]:

avg(PAYLOAD_MASS__KG_)
2928.4

EA with SQL



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

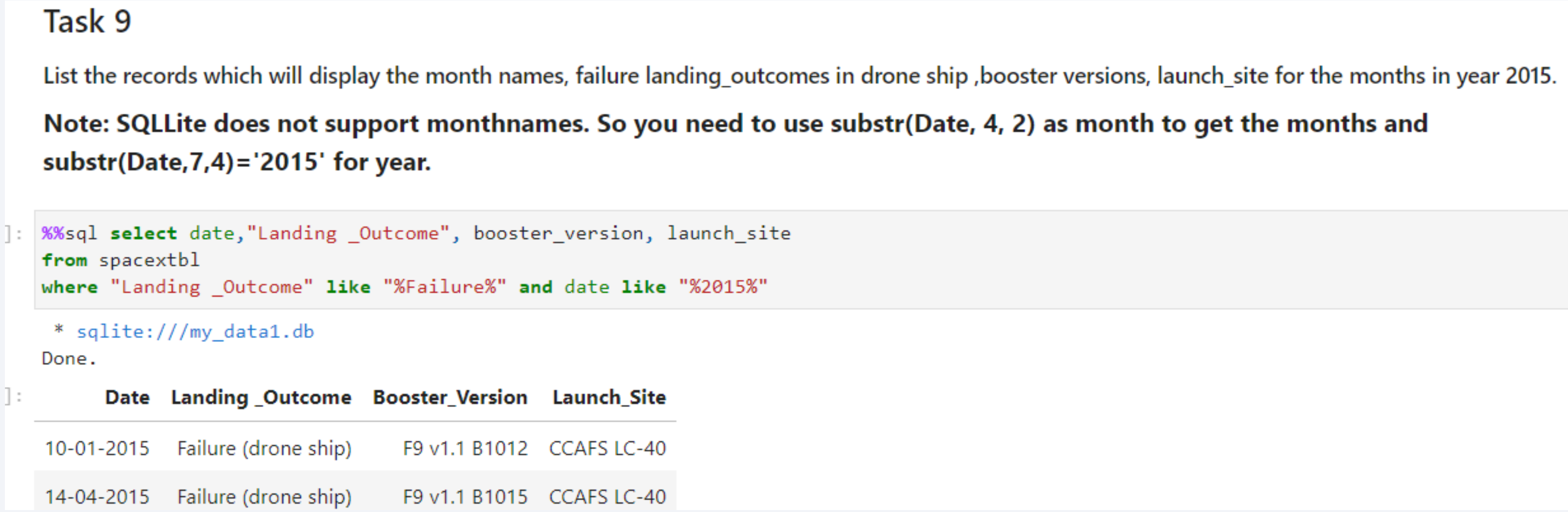
Hint:Use min function

```
[74]: %sql select (date) from SPACEXTBL where "Landing _Outcome"="Success (ground pad)" limit 1
```

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

[74]:

Date
22-12-2015



Task 9

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.

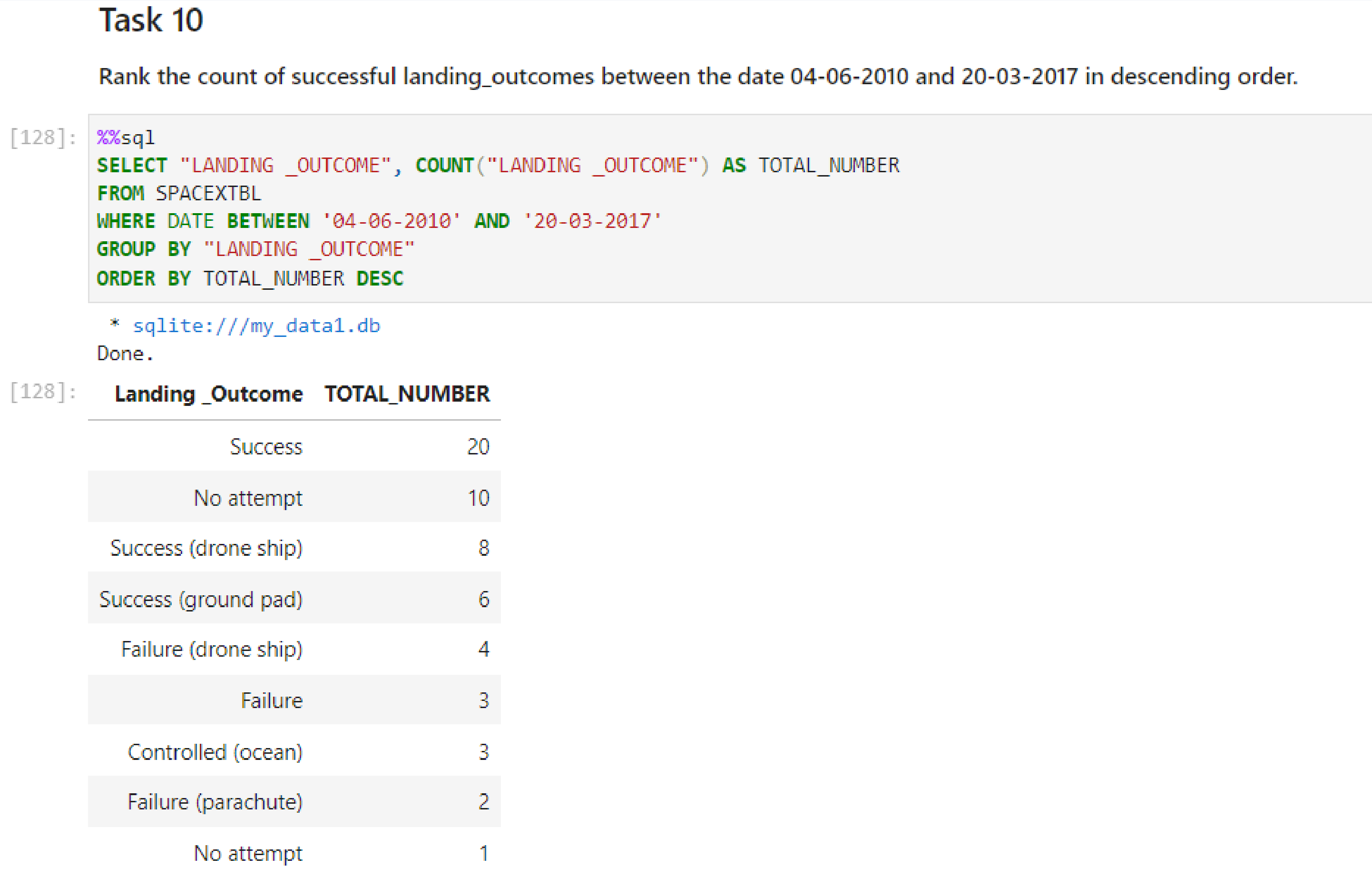
```
]: %%sql select date,"Landing _Outcome", booster_version, launch_site
   from spacextbl
   where "Landing _Outcome" like "%Failure%" and date like "%2015%"
```

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

]:

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

EA with SQL



Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
[128]: %%sql
       SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") AS TOTAL_NUMBER
       FROM SPACEXTBL
       WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
       GROUP BY "LANDING _OUTCOME"
       ORDER BY TOTAL_NUMBER DESC
```

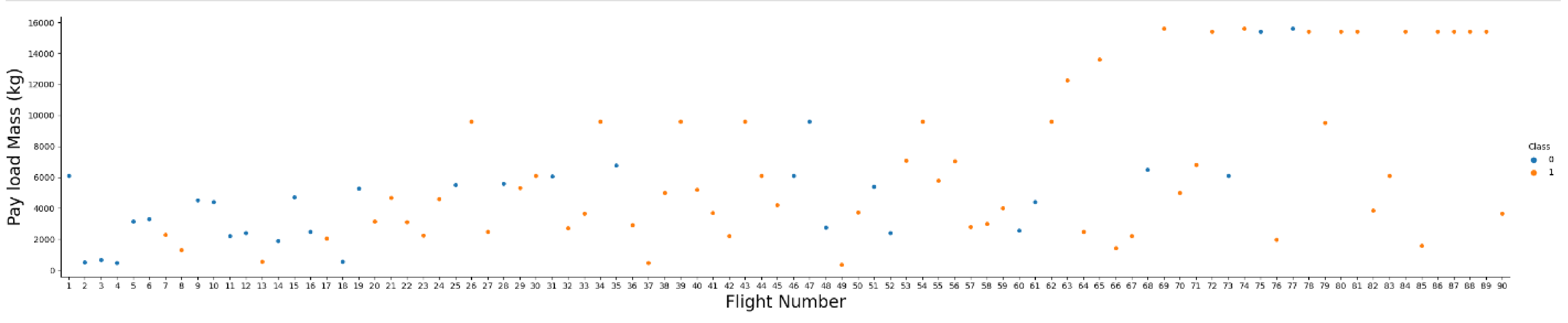
```
 * sqlite:///my_data1.db
Done.
```

[128]:

Landing _Outcome	TOTAL_NUMBER
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

[EDA with SQL Github Link]

EA with Data Visualization

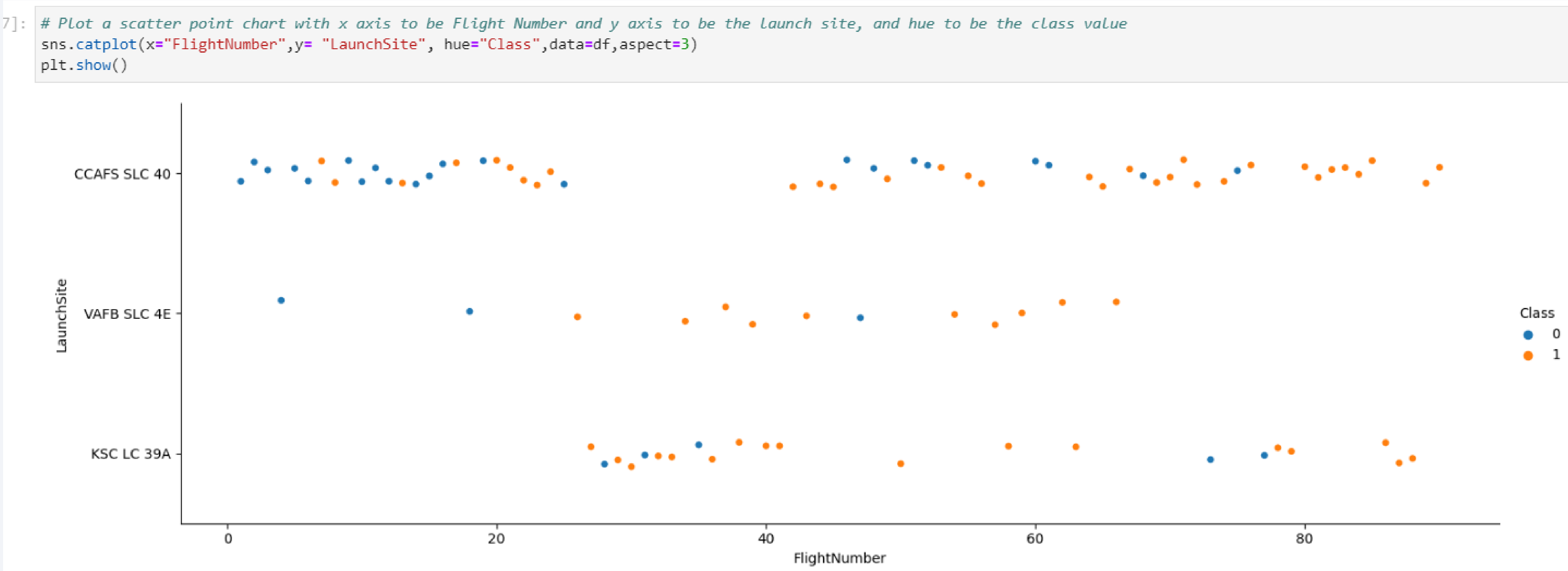


1. as the flight number increases, the first stage is more likely to land successfully

2. more massive the payload, the less likely the first stage will return

3. EA(Pandas and Matplot) GitHub Link

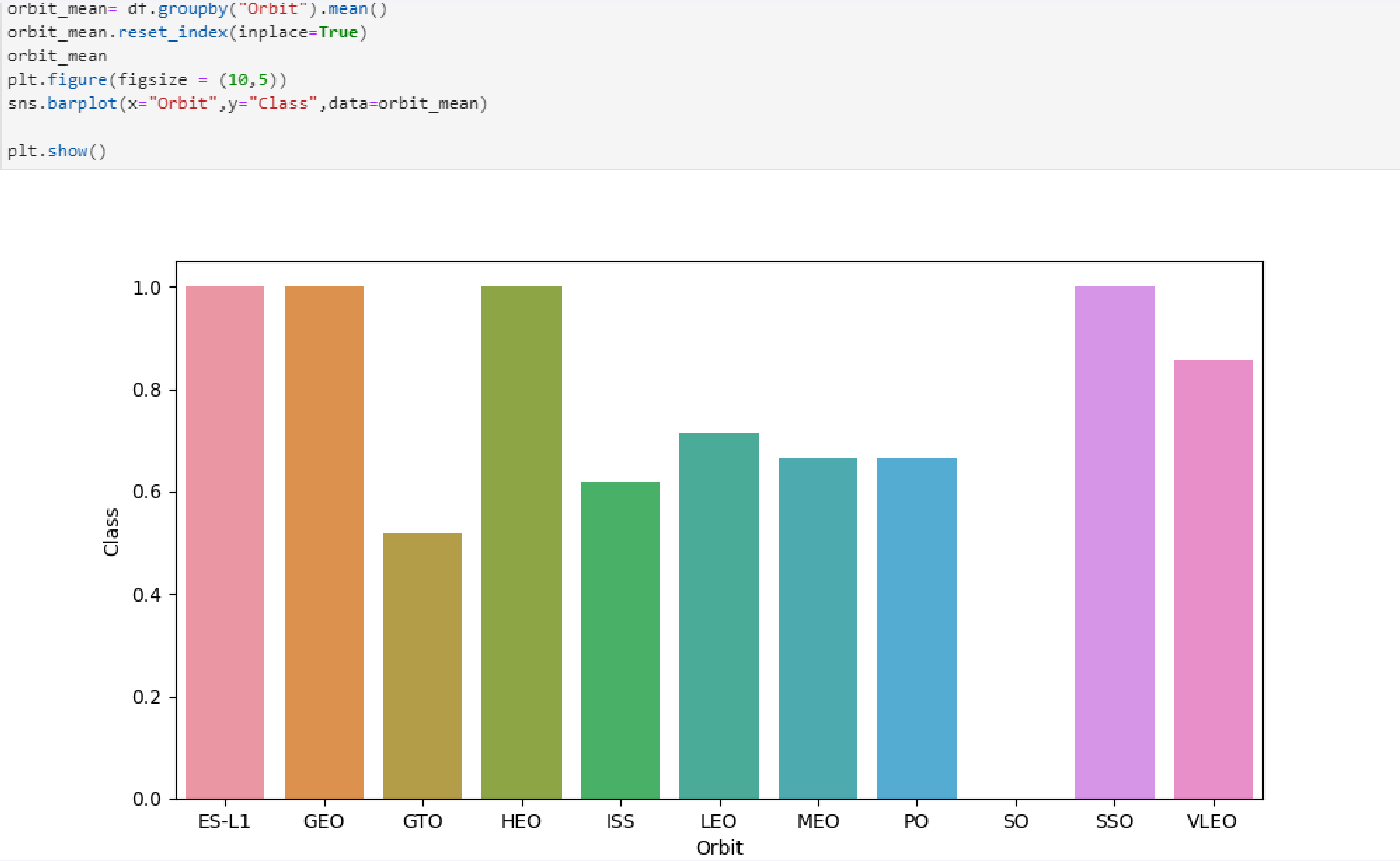
EA with Data Visualization



```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x="FlightNumber",y= "LaunchSite", hue="Class",data=df,aspect=3)
plt.show()
```

1. Most Launches are Launched from CCAFS-SLC-40
2. The earliest flights all failed while the latest flights all succeeded.
3. When the Payload Mass is under 6000kg VAFB SLC 4E and KSCLC 39 A are the better options
4. There is only a few launched for heavy payload

EA with Data Visualization



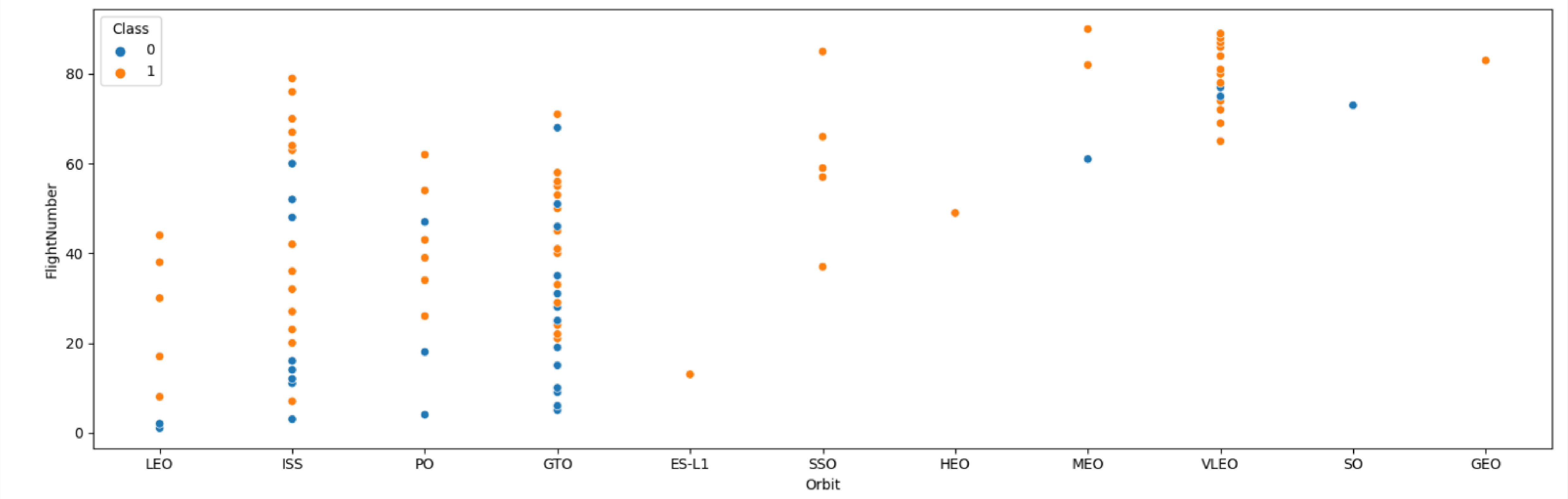
```
orbit_mean= df.groupby("Orbit").mean()
orbit_mean.reset_index(inplace=True)
orbit_mean
plt.figure(figsize = (10,5))
sns.barplot(x="Orbit",y="Class",data=orbit_mean)

plt.show()
```

1. Orbits:
ES-L1, GEO, HEO and SSO have success rate of 100%

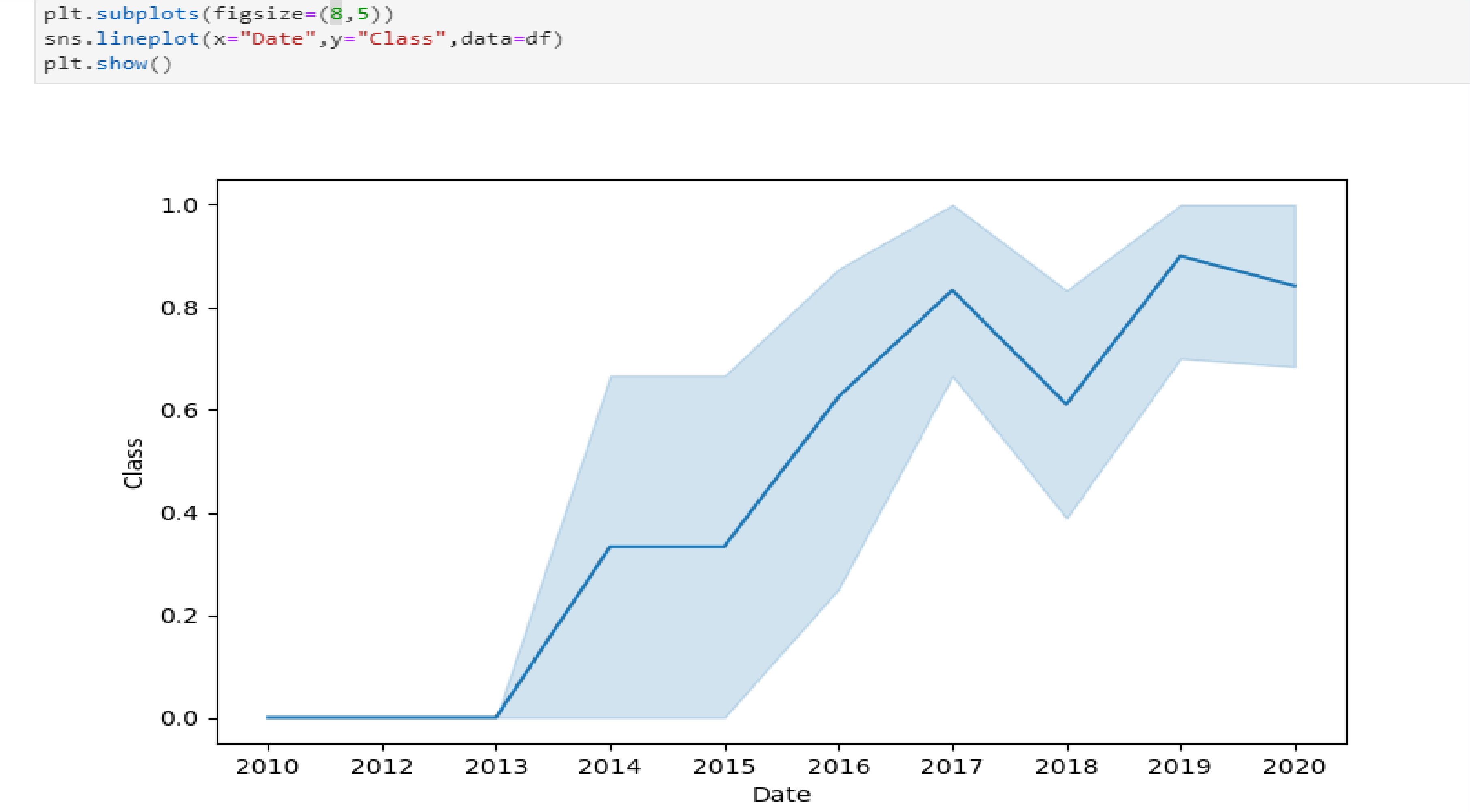
EA(Pandas and Matplot) GitHub Link

EA with Data Visualization



1. Only the orbits GTO and SO have not improve launch success at the last launches
2. EA(Pandas and Matplot) GitHub Link

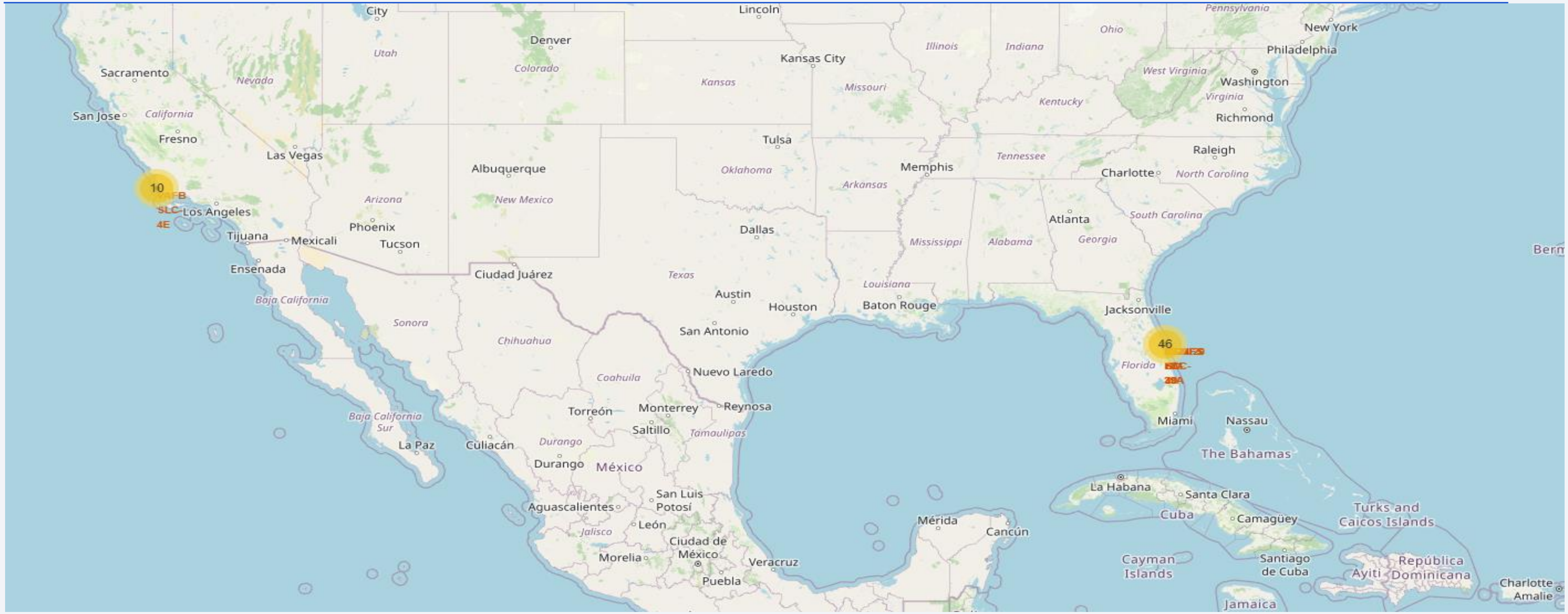
EA with Data Visualization



```
plt.subplots(figsize=(8,5))
sns.lineplot(x="Date",y="Class",data=df)
plt.show()
```

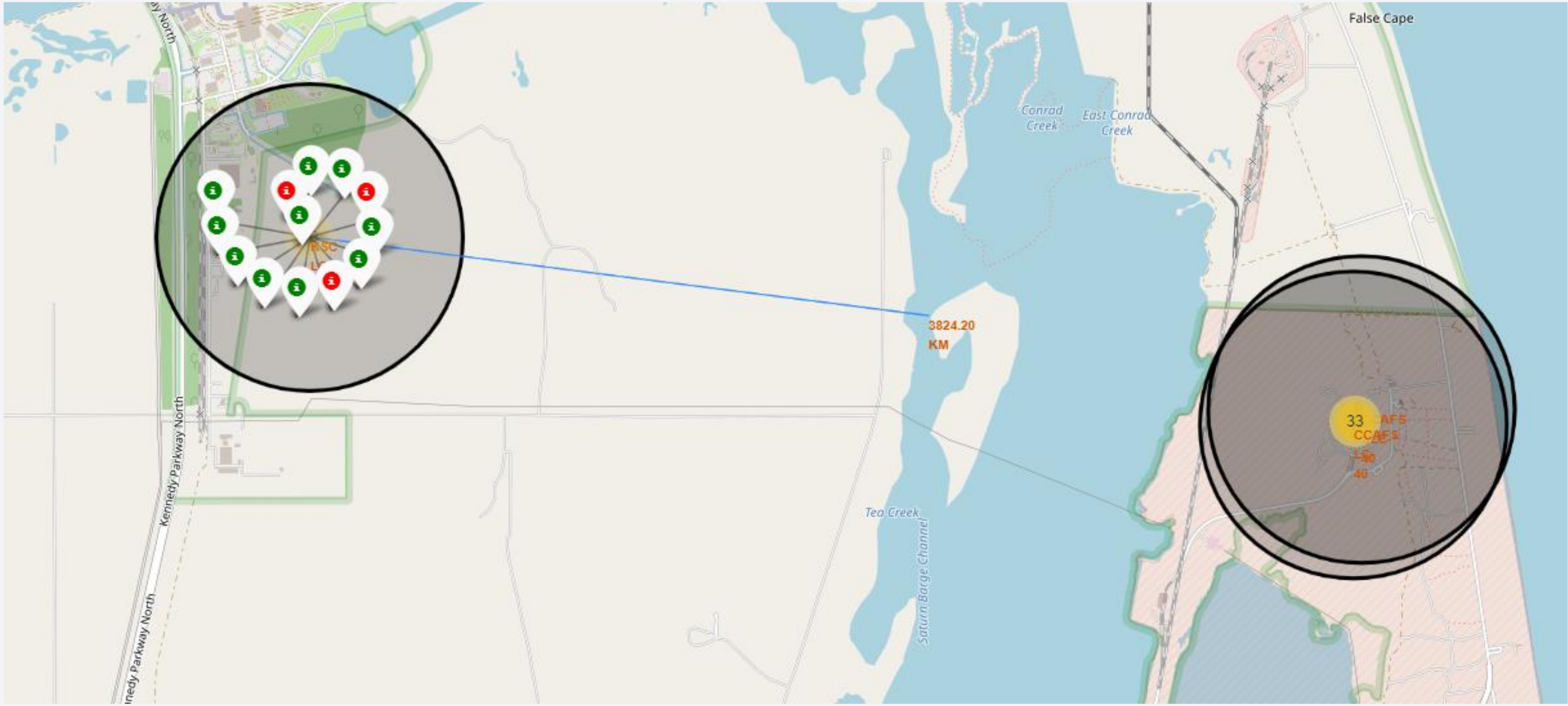
1. The Success rate increase with the years
2. EA(Pandas and Matplot) GitHub Link

Build an Interactive Map with Folium



We localize the launch sites on a map. We can observe that the all the launch sites are proximity to the equator line, and near to the ocean for success probabilities and safety reasons.
Most of the launches where from east coast
Map with Folium GitHub Link

Build an Interactive Map with Folium

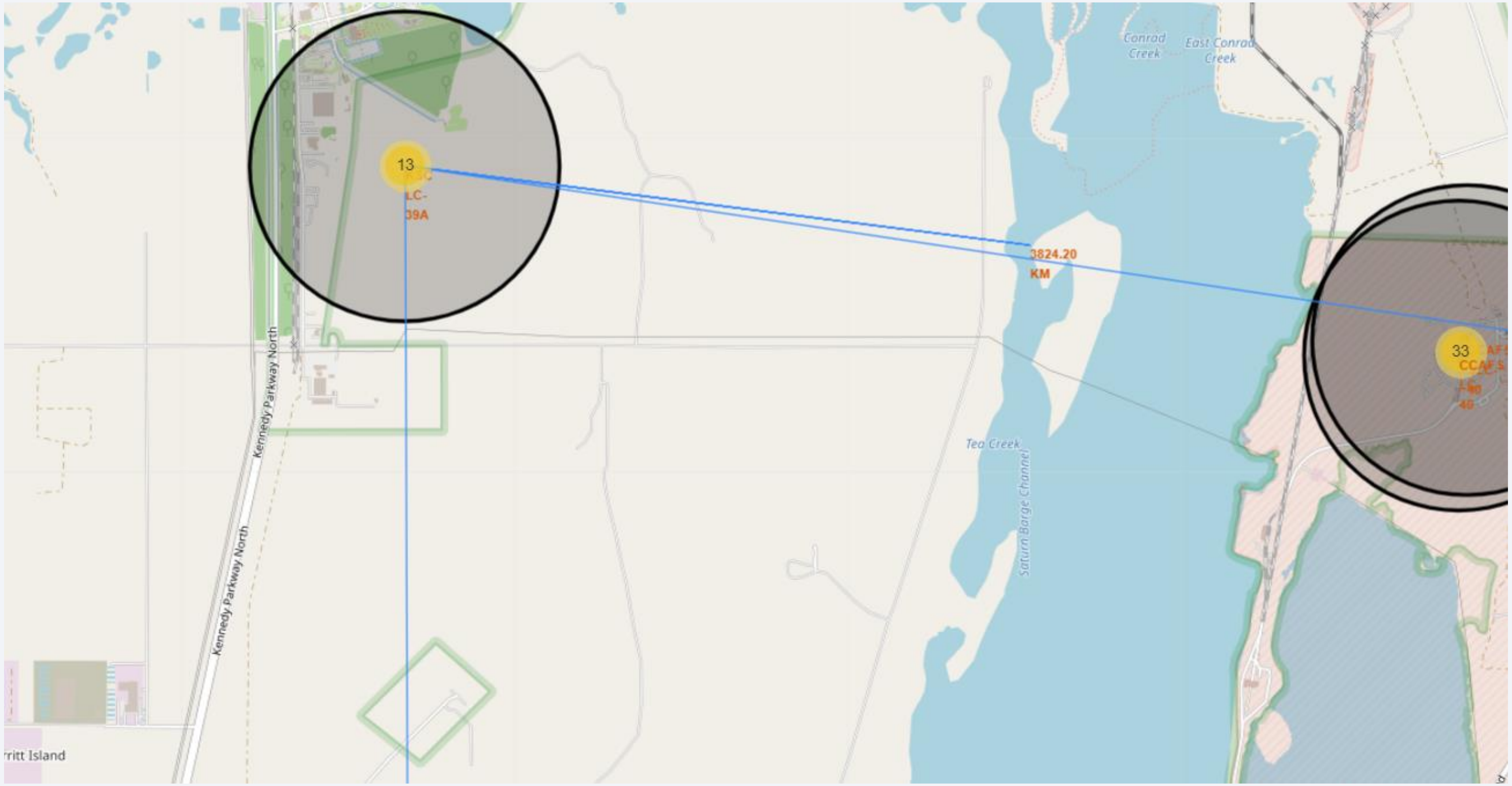


We create a Marker object and customize the Marker's icon property to indicate if this launch was successful or failed.

Launch Site KSC LC-39A has a very high Success Rate.

Map with Folium GitHub Link

Build an Interactive Map with Folium

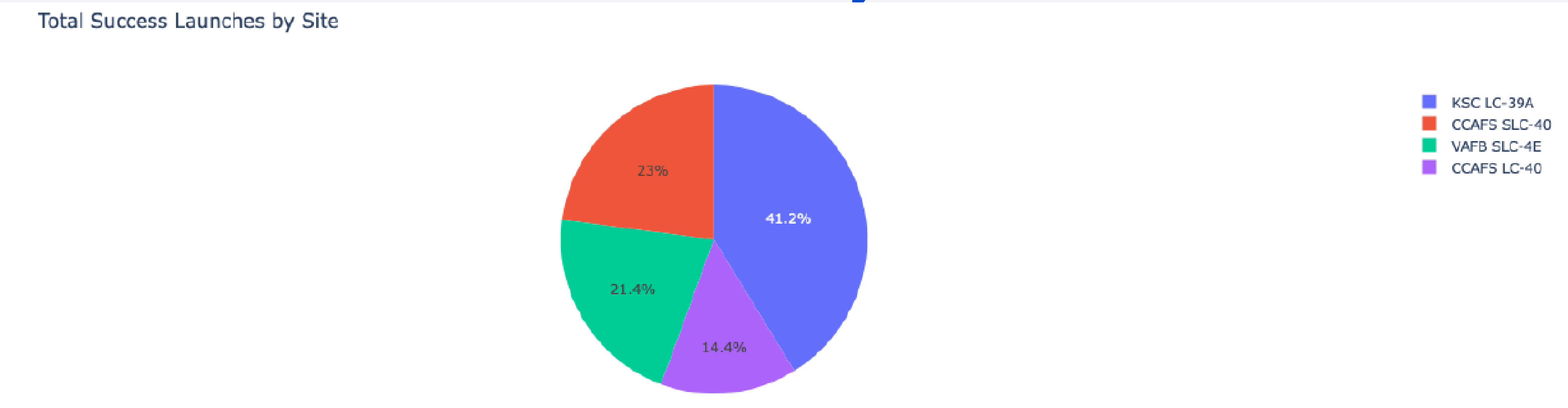


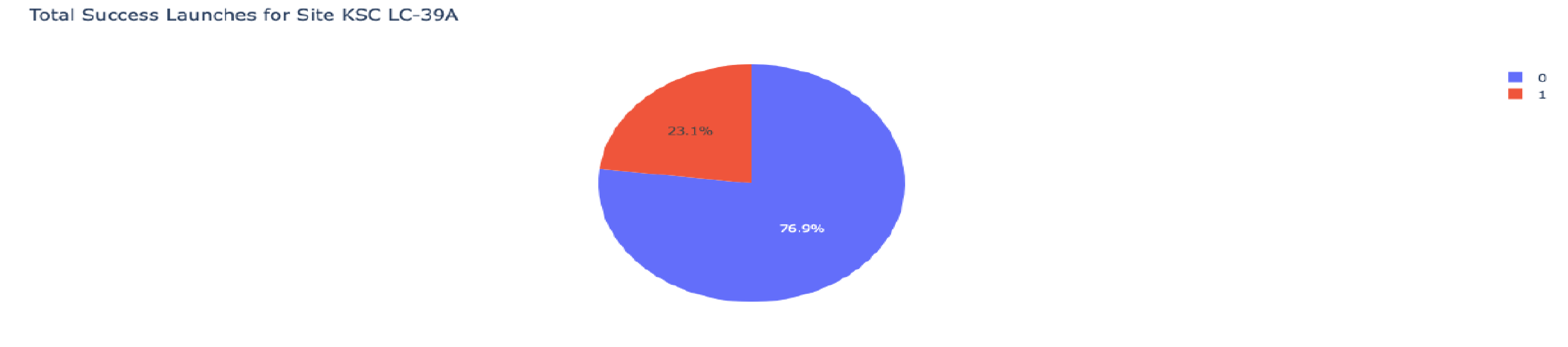
distance_highway = 7.51 km
distance_railroad 6.02 km
distance_city = 52.11 km

Launch sites need infrastructure relative close but for safety city must not be so close

Map with Folium GitHub Link

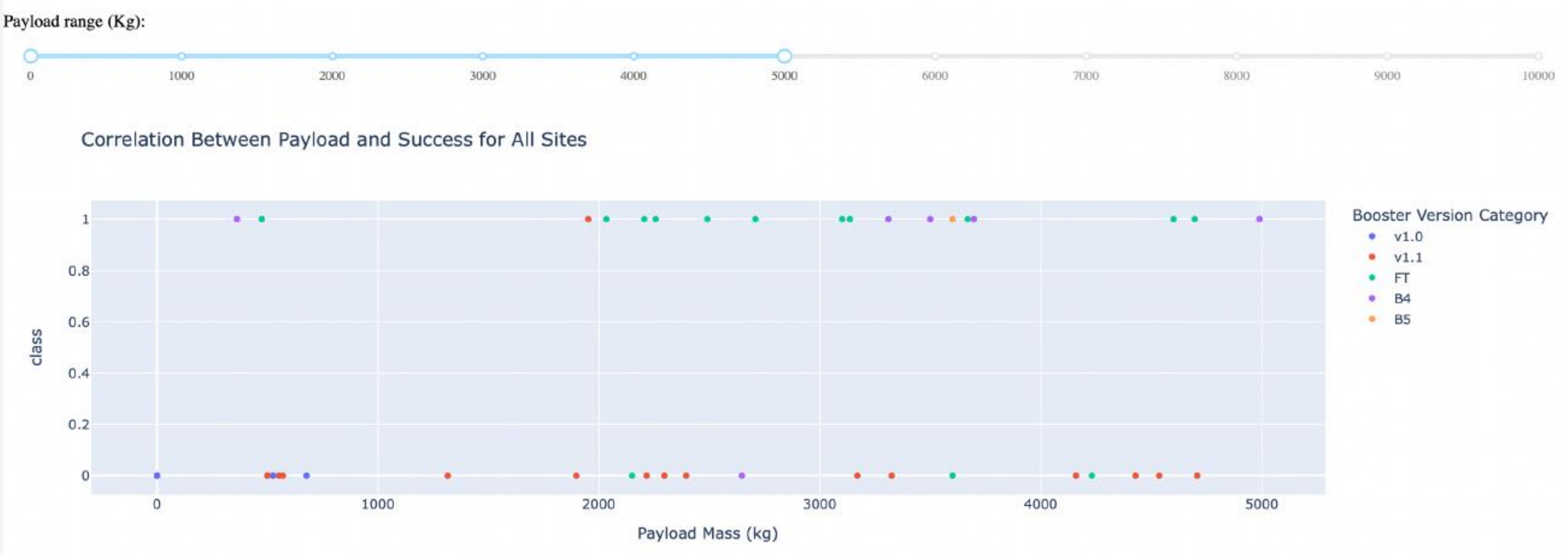
Build a Dashboard with Plotly Dash





KSC LC-39A has the highest success rate,77%

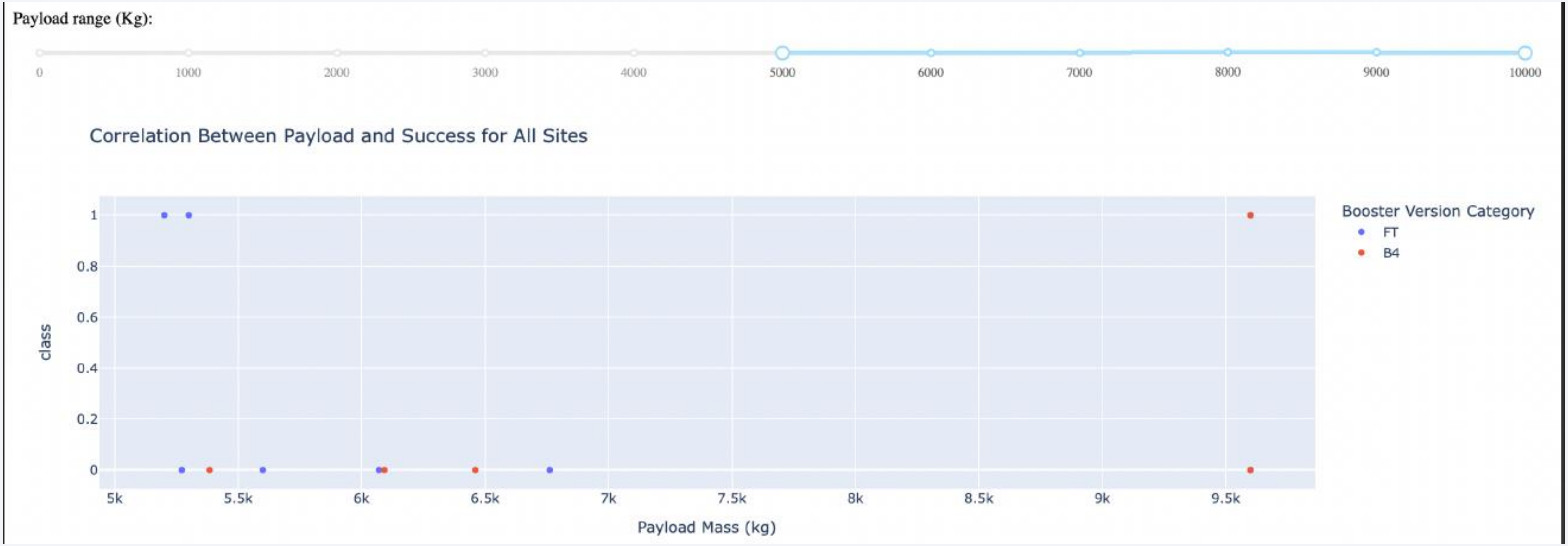
Build a Dashboard with Plotly Dash



Payload range (Kg):

Correlation Between Payload and Success for All Sites

Build a Dashboard with Plotly Dash



Payload range (Kg):

The success rate decrease over Payload Mass of 5000kg

Predictive Analysis (Classification)

1. Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y

2. Standardize the data in X then reassign it to the variable X using the transform provided below.

3. We split the data into training and testing data using the function train_test_split

4. Hyperparameters are selected using the function GridSearchCV

5. We output the **GridSearchCV** object for **logistic regression, svm, KNN,** and **DTC.**

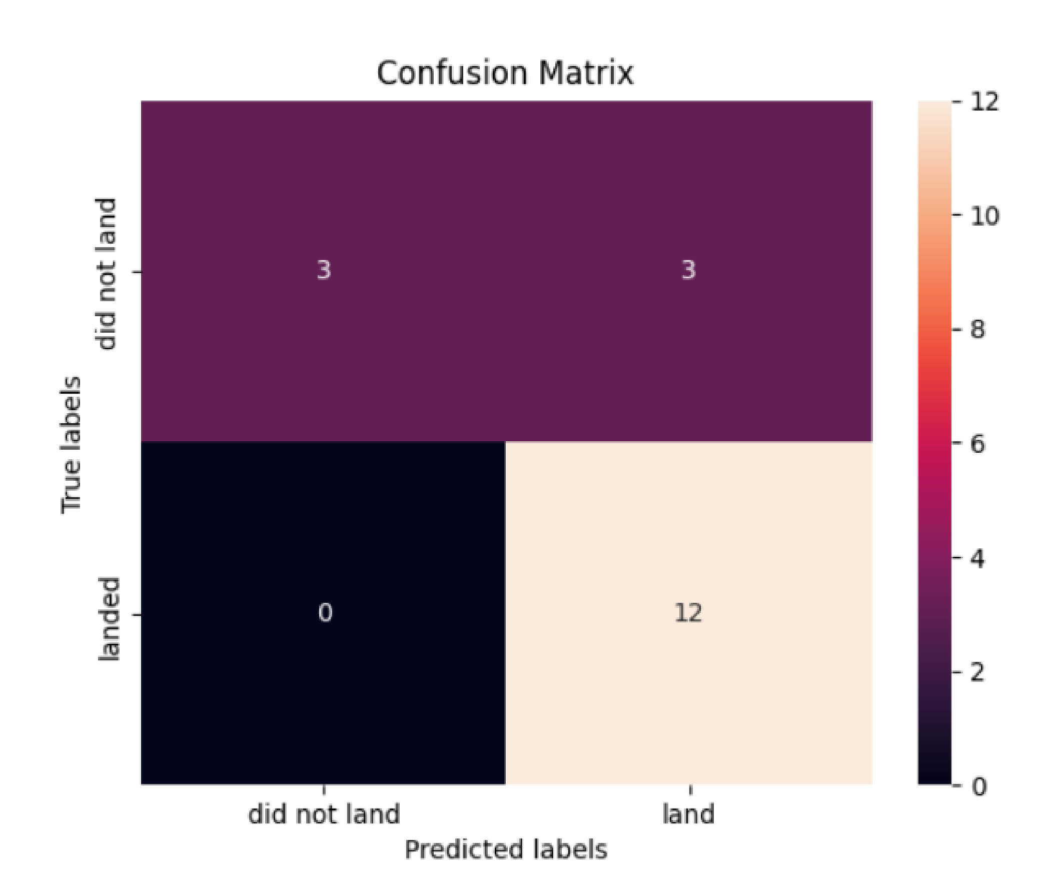
6. We display the best parameters using the data attribute **best_params**_ and the accuracy on the validation data using the data attribute **best_score_.**

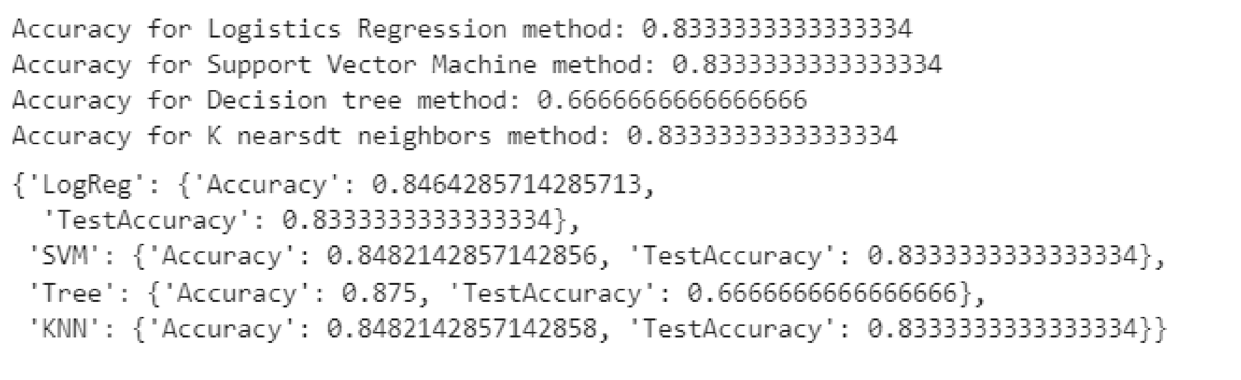
7. Calculate the accuracy on the test data using the method score

First Stage Landing Prediction Git Hub Link

Predictive Analysis (Classification)

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that **the major problem is false positives**.





```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.6666666666666666
Accuracy for K nearsdt neighbors method: 0.8333333333333334
{'LogReg': {'Accuracy': 0.8464285714285713,
  'TestAccuracy': 0.8333333333333334},
 'SVM': {'Accuracy': 0.8482142857142856, 'TestAccuracy': 0.8333333333333334},
 'Tree': {'Accuracy': 0.875, 'TestAccuracy': 0.6666666666666666},
 'KNN': {'Accuracy': 0.8482142857142858, 'TestAccuracy': 0.8333333333333334}}
```

Decision Tree Model is the best algorithm for this dataset.

First Stage Landing Prediction Git Hub Link

Results

Increase Success rate with:

1. Payload Mass under 6000kg
2. Orbits: ES-L1, GEO, HEO and SSO
3. Predictive analysis results
4. Launch sites are in proximity to the Equator line