IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shalini
20th 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 (This task is not included here as I struggled with it)
  - 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 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 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. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- 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:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Exploratory Data Analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash (I struggled here)
- Perform predictive analysis using classification models

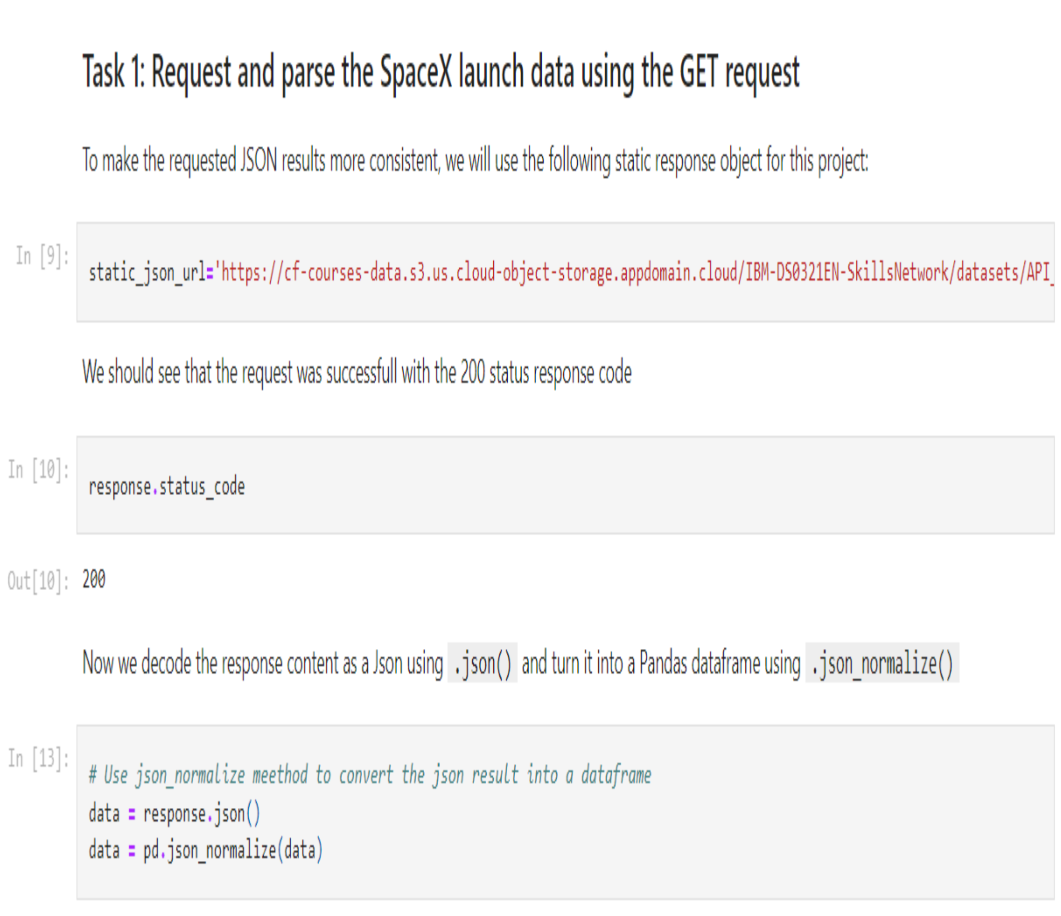
Data Collection

Following steps were followed while collecting the data:

- 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

- GET request to the SpaceX API was used to collect data, requested data was cleaned and some basic data wrangling and formatting was done.
- The link to the notebook is https://github.com/Shalini-Soni99/CapstoneProject/blob/6cedff5b91d14df24b3fcf1fb5e01b2a77ab7041/Data%20Collection.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:

In [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

In [10]:
```
response.status_code
```

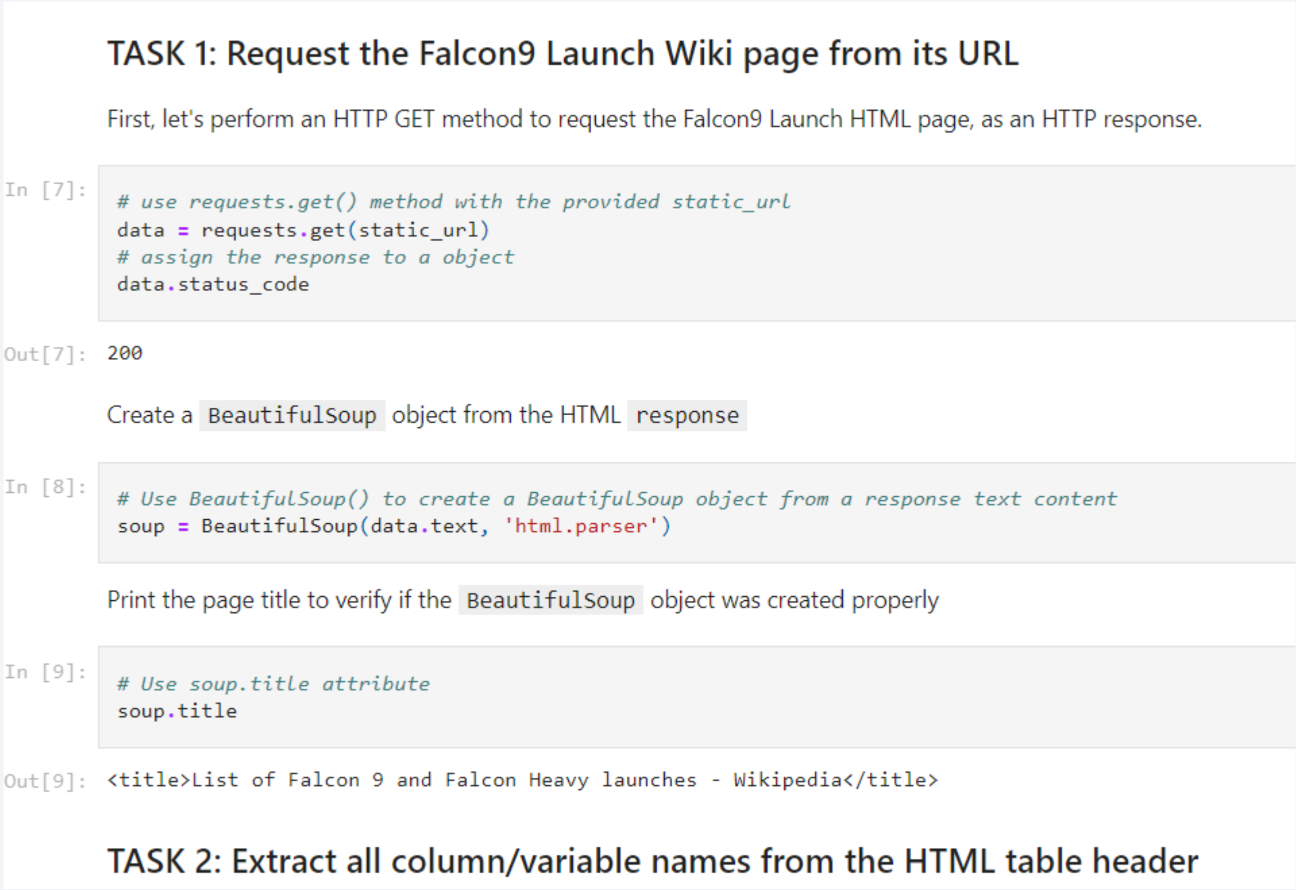
Out[10]:
```
200
```

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

In [13]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = response.json()
data = pd.json_normalize(data)
```

Data Collection - Scraping

- Web scrapping Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into a pandas dataframe.
- The link to the notebook is
- https://github.com/Shalini-Soni99/CapstoneProject/blob/829604c98d5d149e52c165724a1b10159feb2899/Data%20Collection_%20Web%20Scrapping.ipynb



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.

```
In [7]:  # use requests.get() method with the provided static_url
         data = requests.get(static_url)
         # assign the response to a object
         data.status_code

Out[7]:  200
```

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

```
In [8]:  # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
         soup = BeautifulSoup(data.text, 'html.parser')
```

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

```
In [9]:  # Use soup.title attribute
         soup.title

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

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

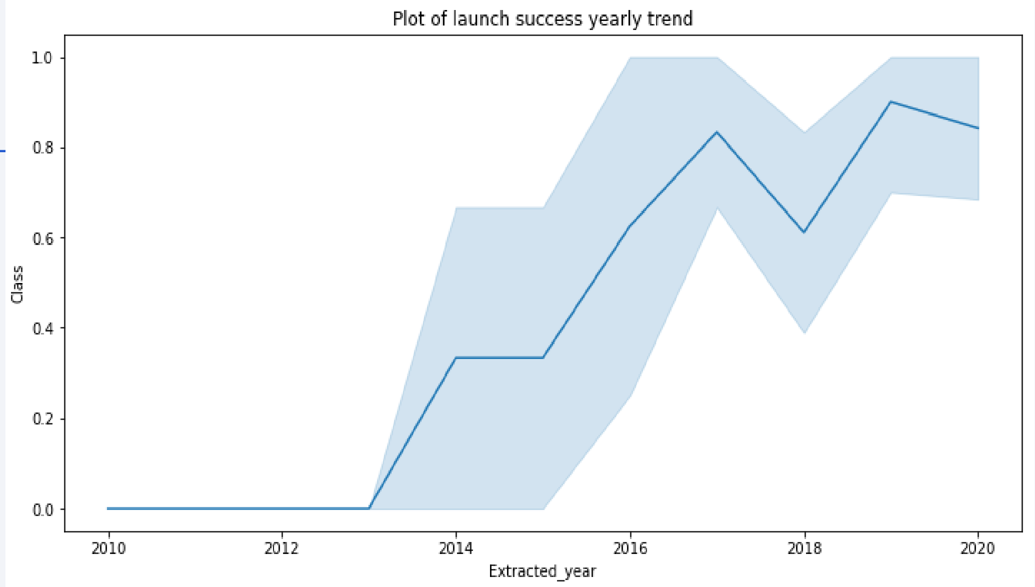
Data Wrangling

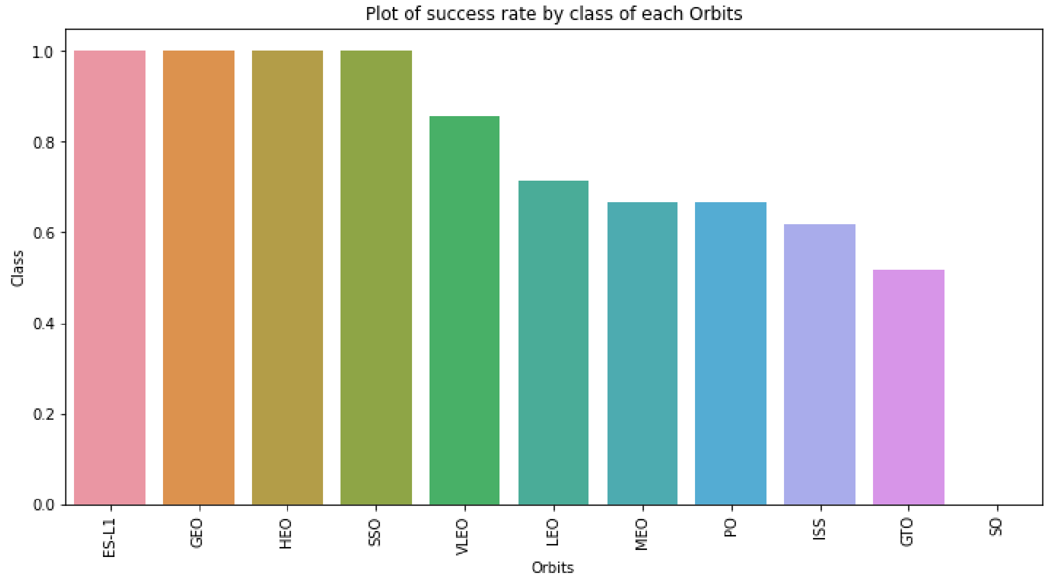
- Exploratory data analysis was performed and the training labels were determined.
- The number of launches at each site and the number and occurrence of each orbit were calculated.
- Landing outcome label from outcome column was created and the results were exported to csv.
- The link to the notebook is https://github.com/Shalini-Soni99/CapstoneProject/blob/64a1738015bdccf4d55d79d0f62b651a615fc72d/labs-jupyter-spacex-data_wrangling_jupyterlite.jup.ipynb

EDA with Data Visualization

- The data was 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/Shalini-Soni99/CapstoneProject/blob/6f2d9daabd047c26547c0cb339763bc988f54801/jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb





Build an Interactive Map with Folium

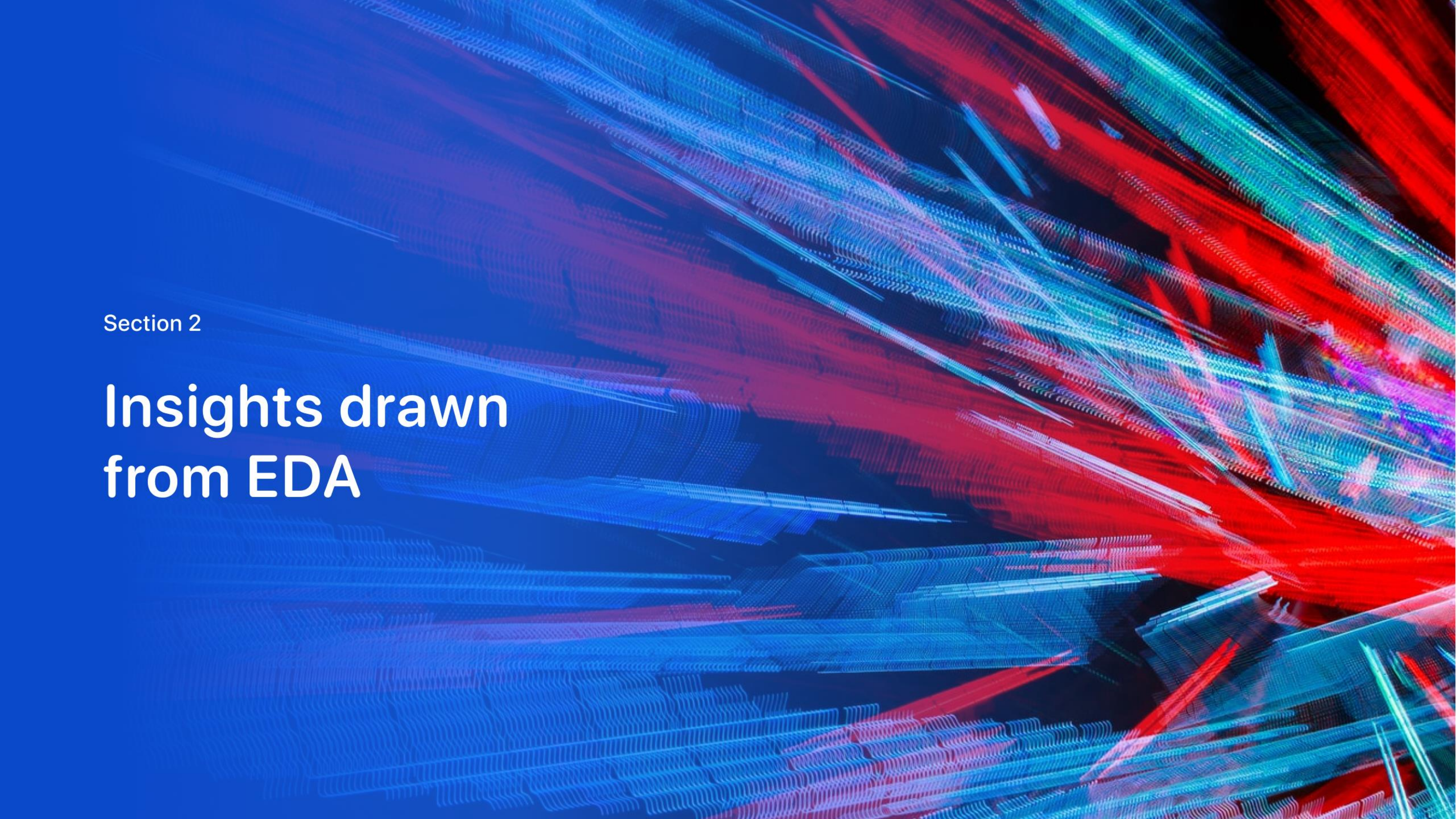
- All the launch sites were marked first, and then map objects such as markers, circles, lines were added to mark the success or failure of launches for each site on the folium map.
- The feature launch outcomes (failure or success) were assigned to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, it was identified that which launch sites have relatively high success rate.
- The distances between a launch site to its proximities were calculated.
- The link to the notebook is https://github.com/Shalini-Soni99/CapstoneProject/blob/eae3267a6ef82450b771c8a0e53604d14e6d2950/lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb

Predictive Analysis (Classification)

- The data was loaded using numpy and pandas, it was transformed and split into training and testing
- Different machine learning models were used.
- Accuracy was used as the metric for the model and it was improved using feature engineering and algorithm tuning.
- Then the best performing classification model was found based on the score.
- The link to the notebook is https://github.com/Shalini-Soni99/CapstoneProject/blob/4c9018a7b6a65eaddbb449ed9473c8ea889950cf/SpaceX_Machine_Learning_Prediction_Part_5.jupyterl.ipynb

Results

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

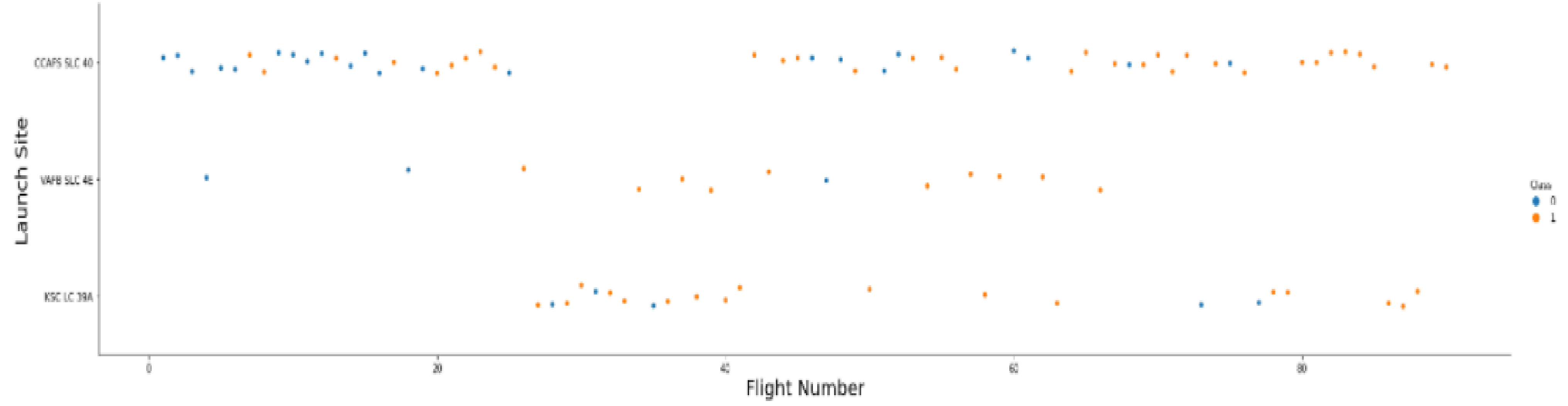


Section 2

Insights drawn from EDA

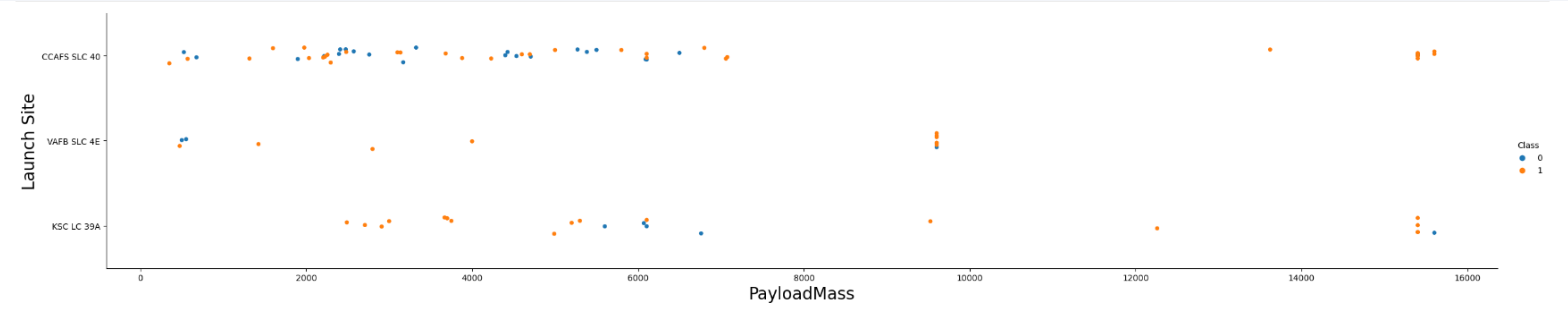
Flight Number vs. Launch Site

- From the plot, it was found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



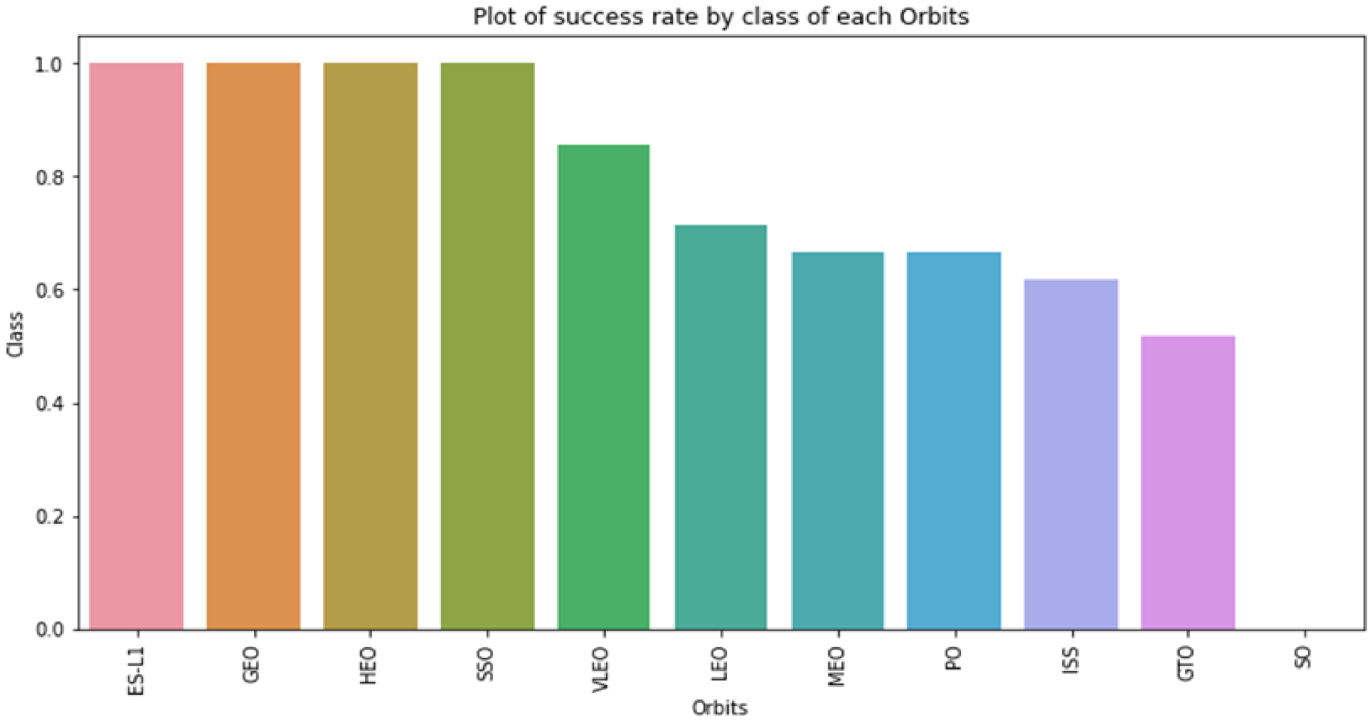
Pay Load vs. Launch Site

- From the plot, it was found that the greater the payload mass for launch site CCAFS SLC 40, the higher the success rate for the rocket.



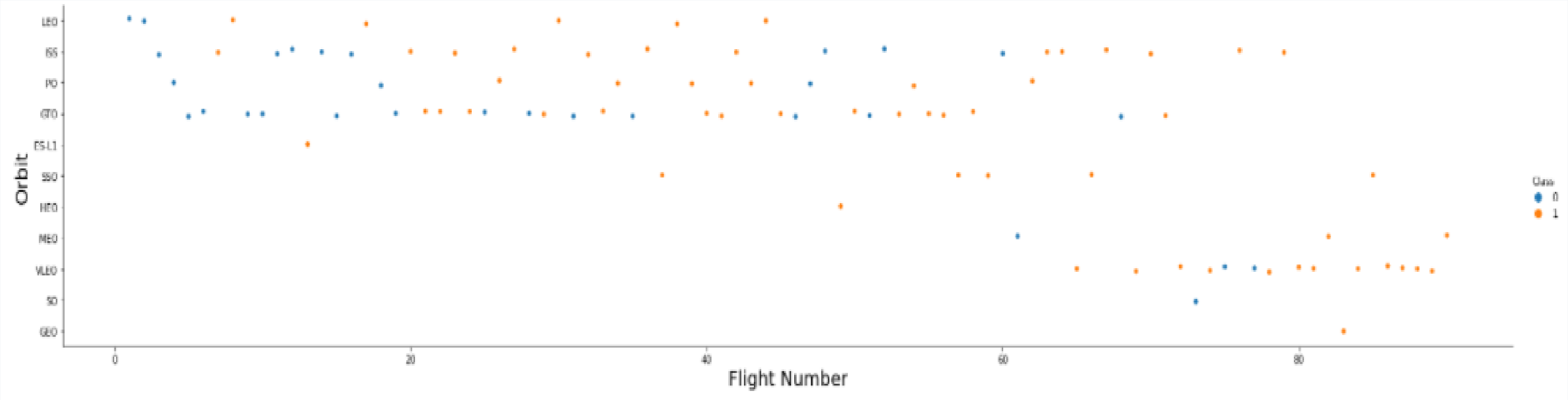
Success Rate vs. Orbit Type

- From the plot, it can be observed that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



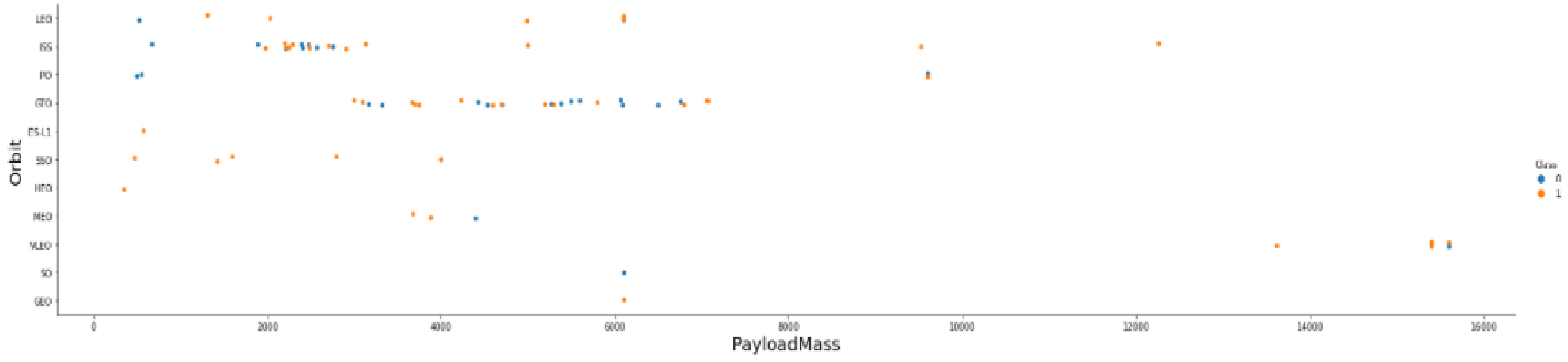
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. 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.



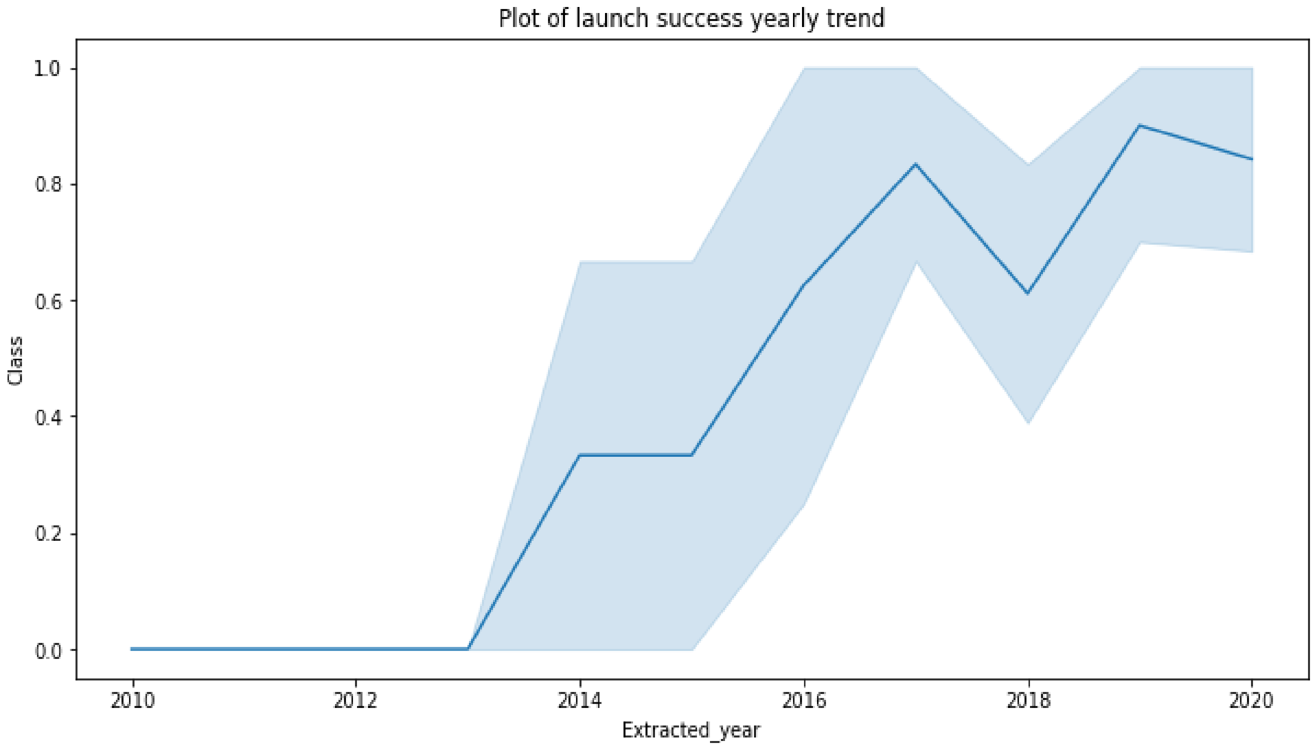
Payload vs. Orbit Type

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



Launch Success Yearly Trend

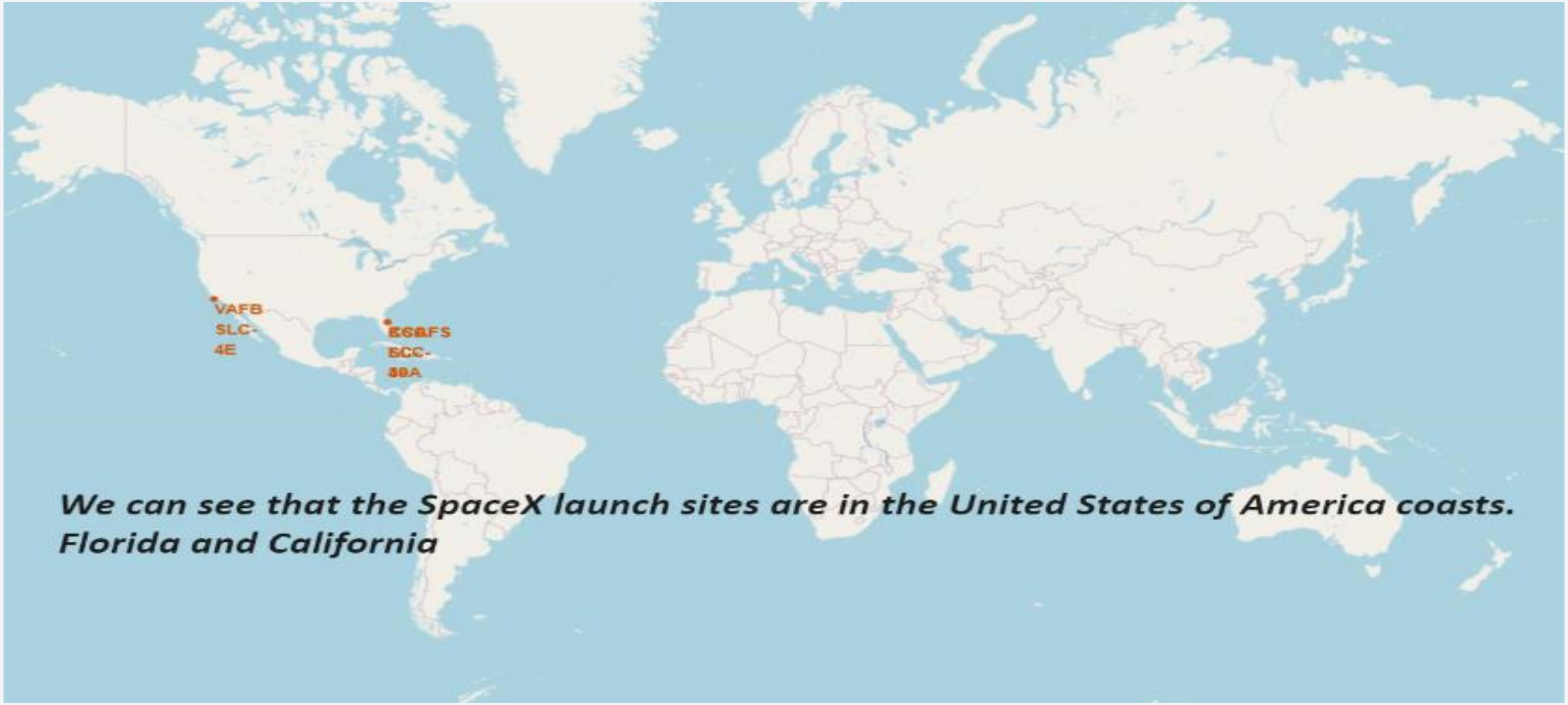
- From the plot, it can be observed that success rate since 2013 kept on increasing till 2020.



Section 4

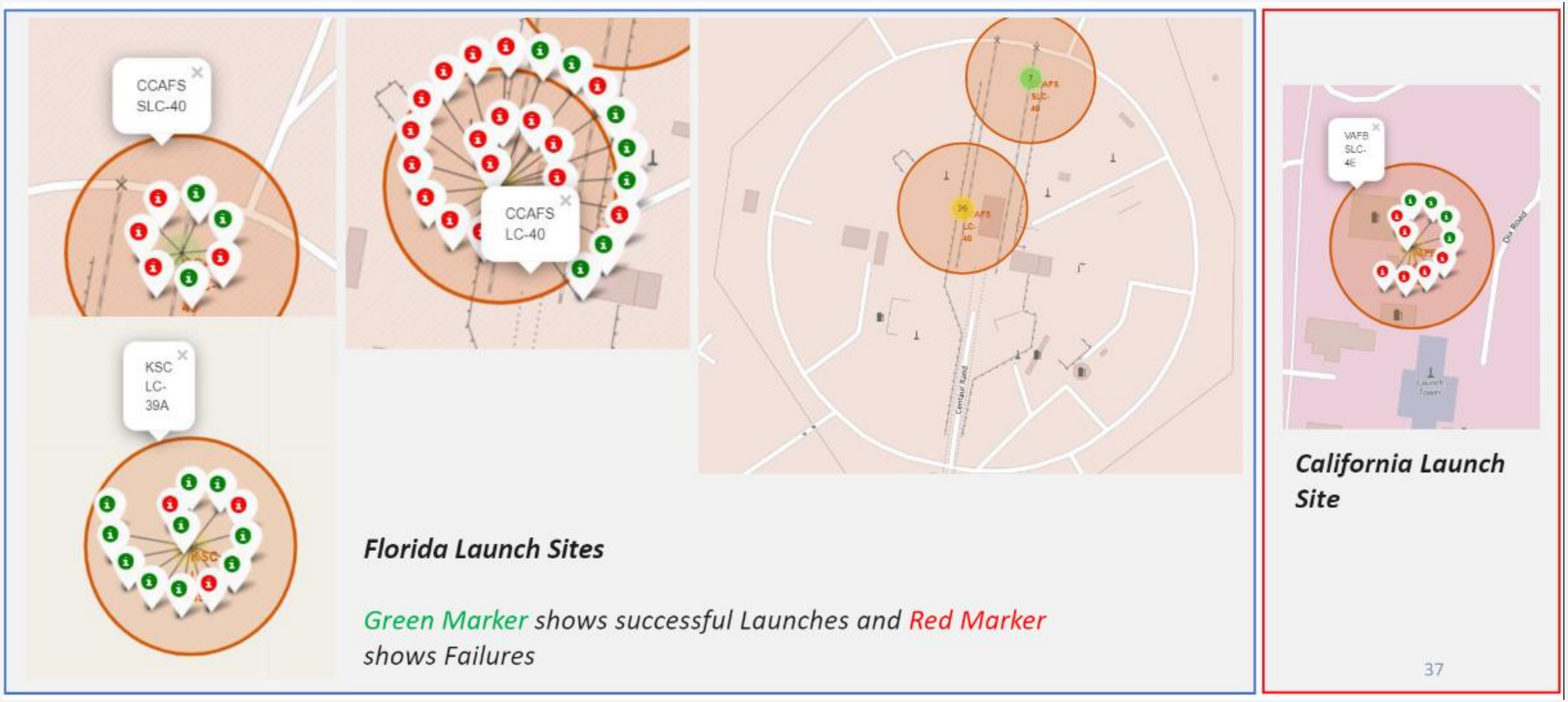
Launch Sites Proximities Analysis

All launch sites global map markers



We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels



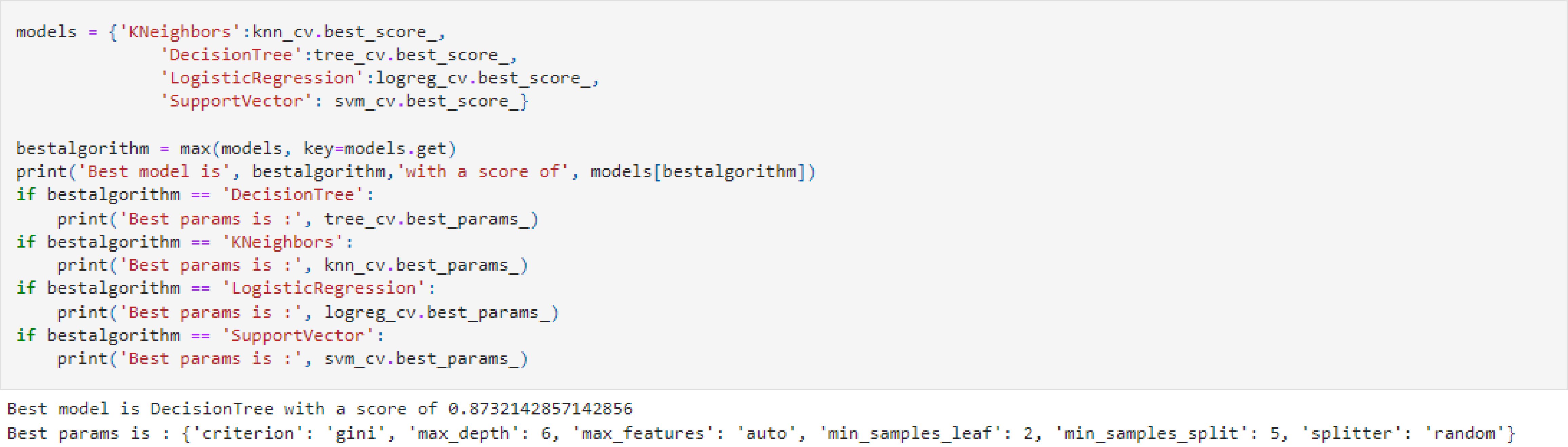
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

Classification Accuracy

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



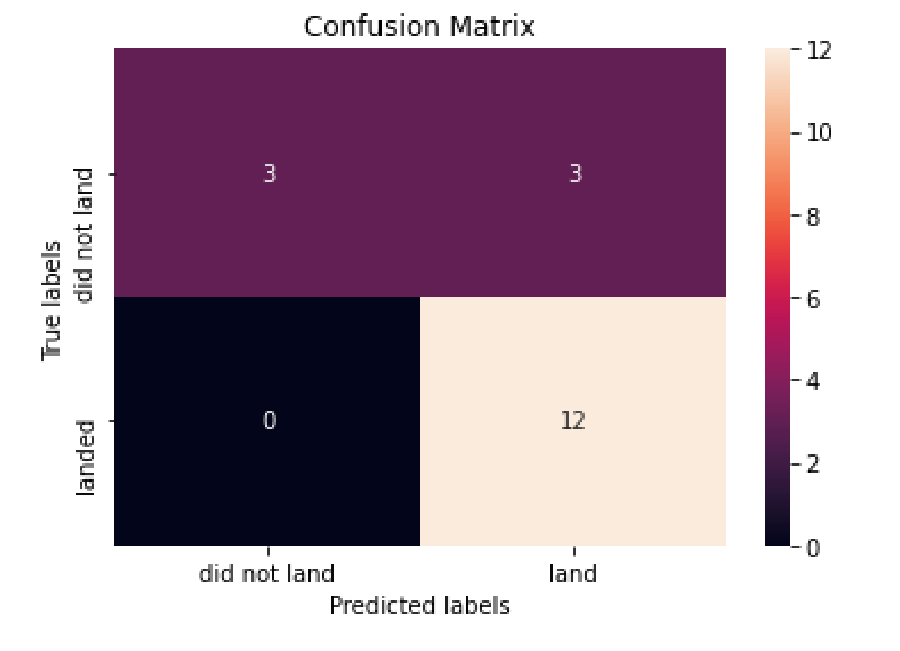
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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

- A confusion matrix summarizes the performance of a classification algorithm
- All the confusion matrices were identical
- The fact that there are false positives (Type 1 error) is not good



Conclusions

We can conclude 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 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!