
PROJECT SPACE MISSION
USING DATA SCIENCE
Siddhant Shetty
10th April 2024

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Result
- Conclusion

EXECUTIVE SUMMARY

- Data Collection using API
- Data Collection using web Scraping
- Data Wrangling
- Exploratory Data Analysis using SQL
- Exploratory Data Analysis using Visualization
- Interactive Visualizations
- Predictive Analytics and Results

Introduction

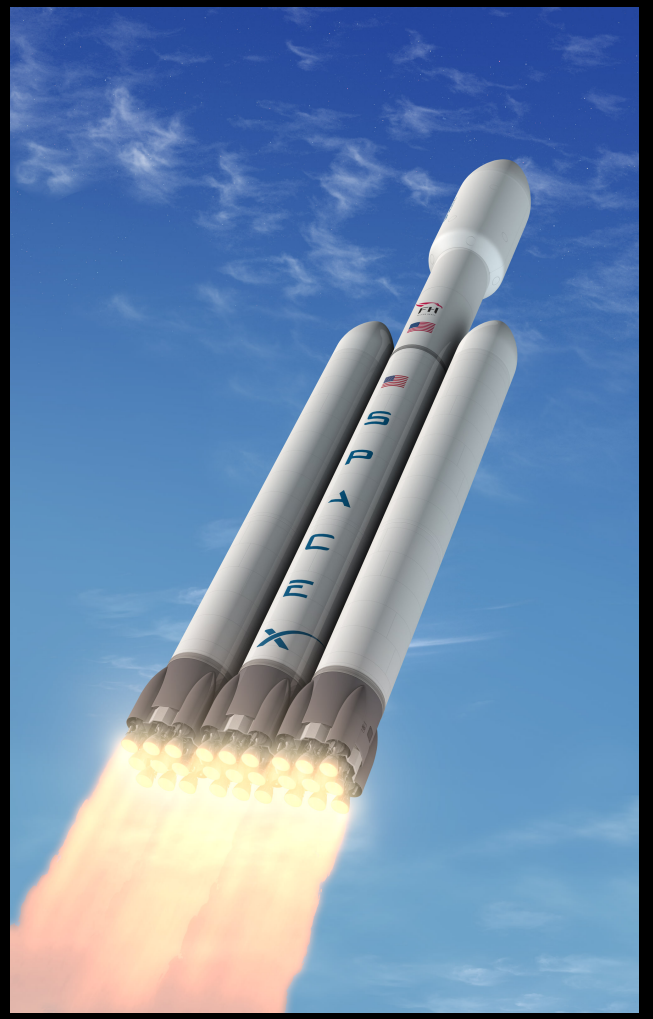
PROJECT BACKGROUND AND CONTEXT

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.

The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully.

This info. can be used by another company who wants top bid against SpaceX for a rocket launch.



METHODOLOGY

- Data Collection using SpaceX API and web scraping from Wikipedia
- Data Wrangling on the collected data
- EDA using SQL and Visualizations
- Interactive Analytics
- Performing Predictive Analytics using classification models

DATA COLLECTION

- The data is collected through Wikipedia as well as the SpaceX website.
- From the SpaceX website, data is collected using get request
- The response is coded as a json file that is converted into a pandas data frame using json_normalize
- Data is then checked for missing values and cleaned.
- The other data is web scraped from Wikipedia using beautiful soup
- The html table is parsed to extract launch records and convert into pandas dataframe.

DATA COLLECTION – SPACEX API

Get request is used on the SpaceX API to collect data, clean the data and do data wrangling.

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

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

Check the content of the response

```
print(response.content)
```

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[47]: 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

```
[48]: response.status_code
```

```
[48]: 200
```

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

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

Using the dataframe `data` print the first 5 rows

```
[50]: # Get the head of the dataframe
data.head(5)
```

DATA COLLECTION – WEB SCAPING

Web scraping is used to scrape launch records of falcon9 rocket with beautiful soup

Html table is parsed and converted. To a pandas data frame.

Further, wrangling is done on the data. This can be seen on the notebook uploaded on the github.

[11]:
```
# 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
```

[11]:
```
200
```

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

[14]:
```
# 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

[15]:
```
# Use soup.title attribute
soup.title
```

[15]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

[16]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

[17]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
<th scope="col">Flight No.
</th>
```

DATA WRANGLING

```
[7]: as is a software library written for the Python programming language for data manipulation and analysis.
     pandas as pd
     is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection
     numpy as np
```

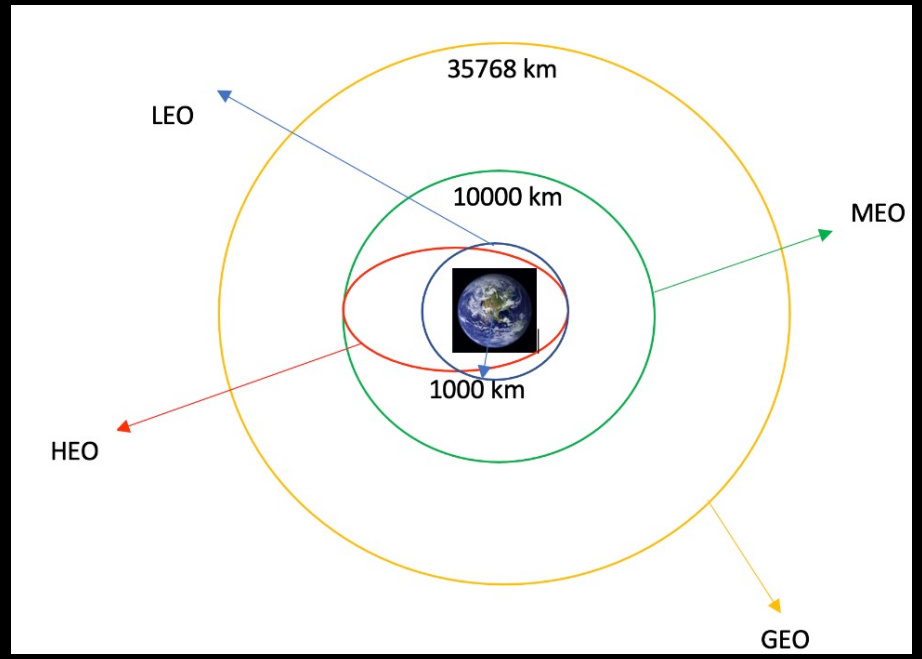
Data Analysis

Load Space X dataset, from last section.

```
[8]: df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
     df.head(10)
```

[8]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	L
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-8
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-8



Exploratory data analysis is performed on the data.

Number of Launches on each site and number/occurrence of each orbits are calculated.

Landing outcome label is created and results are exported to csv.

EDA WITH SQL

- The SpaceX dataset is loaded to IbmDB2 database. A connection is made between the jupyter notebook and SQL database.
- SQL queries are further written to find names of unique launches, total payload mass, no. of successful and failure mission outcomes.

```
%sql select distinct launch_site from SPACEXDATASET;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518
198/bludb
Done.
```

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

Distinct keyword used to show only unique launch sites.

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, count(*) as count_outcomes from SPACEXDATASET
      where date between '2010-06-04' and '2017-03-20'
      group by landing__outcome
      order by count_outcomes desc;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cl
198/bludb
Done.
```

landing__outcome	count_outcomes
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3

Landing Outcomes during a particular period.

MORE SQL QUERIES

Task 3

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

In [6]:
```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[6]:

total_payload_mass
45596

Task 4

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

In [7]:
```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXDATASET where booster_version like '%F9 v1.
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[7]:

average_payload_mass
2534

Task 5

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

Hint:Use min function

In [8]:
```
%sql select min(date) as first_successful_landing from SPACEXDATASET where landing__outcome = 'Success (ground pa
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[8]:

first_successful_landing
2015-12-22

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [9]:
```
%sql select booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[9]:

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

Task 7

List the total number of successful and failure mission outcomes

In [10]:
```
%sql select mission_outcome, count(*) as total_number from SPACEXDATASET group by mission_outcome;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[10]:

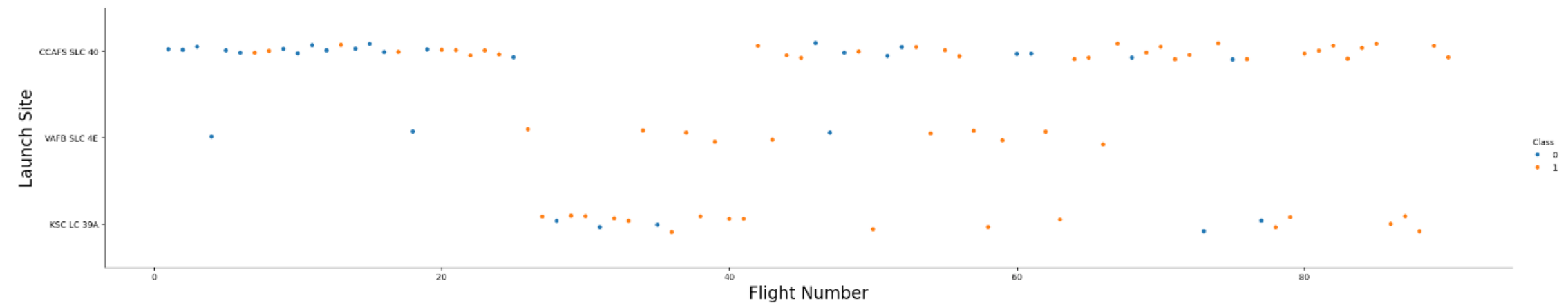
mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

EDA WITH VISUALIZATION

- Data is explored by visualizing the relationship between flight number and site launch, payload and launch site, flight number and orbit, success rate of each orbit type.

- The following visualization results are shared in the upcoming slides.

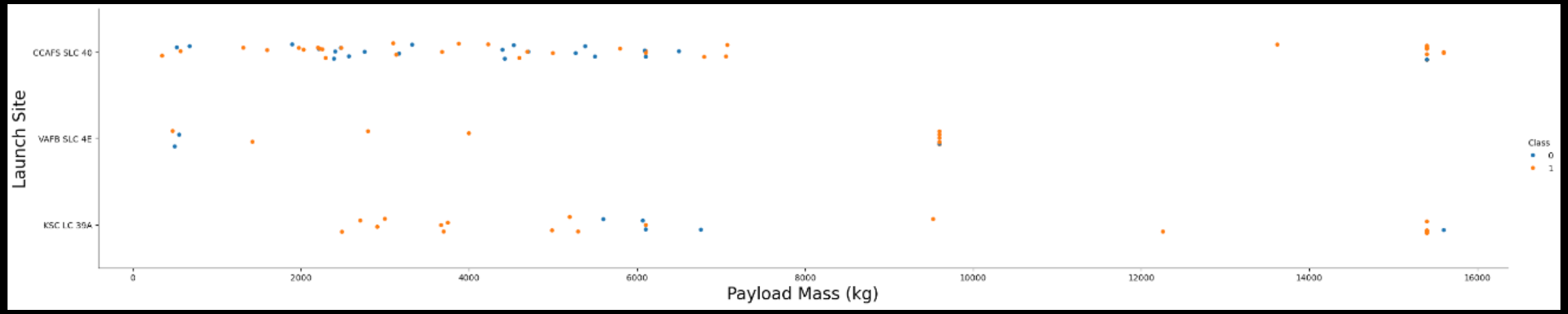
FLIGHT NUMBER VS LAUNCH SITE



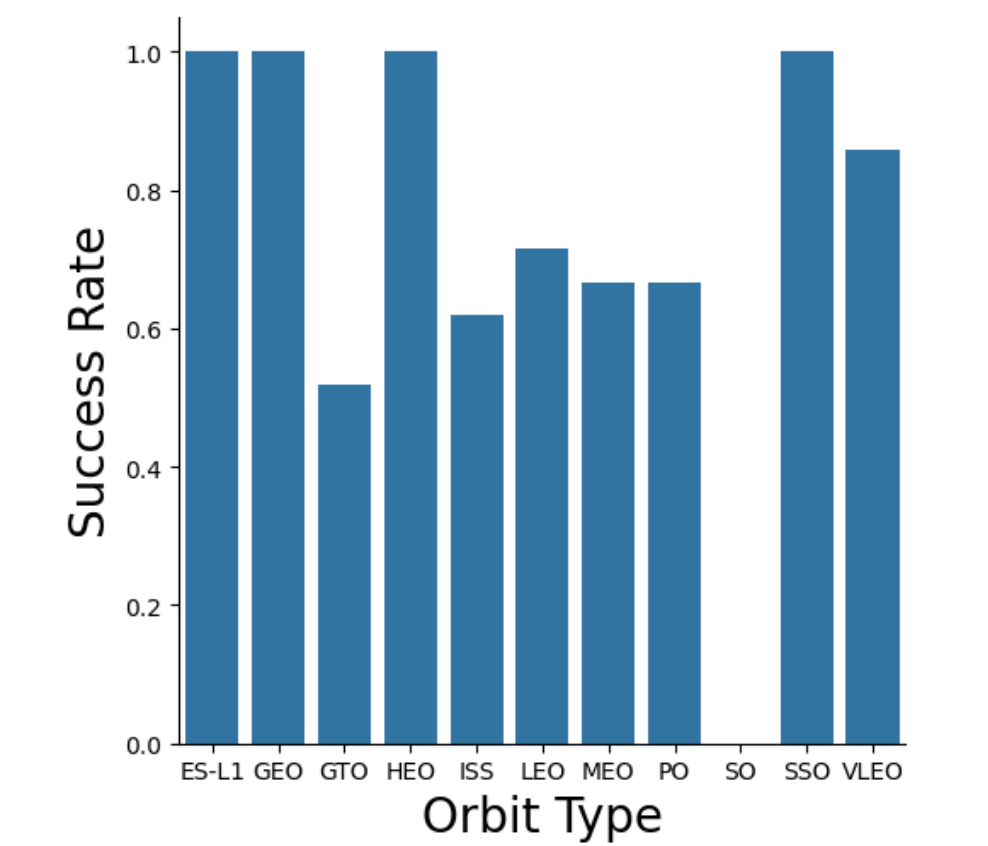
From the plot it is clearly evident. Especially for CCAFS SLC 40, more the flight number , more the launch site success rate. It stands true for the other two launch sites as. Well.

PAYLOAD VS LAUNCH SITE

The greater the mass for launch site CCAFS SLC 40, the higher the success rate of the rocket.



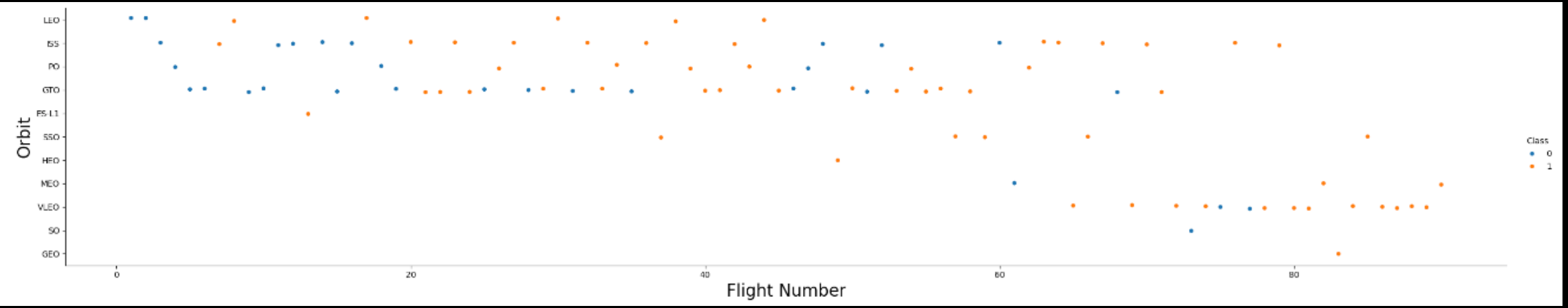
SUCCESS RATE VS ORBIT TYPE



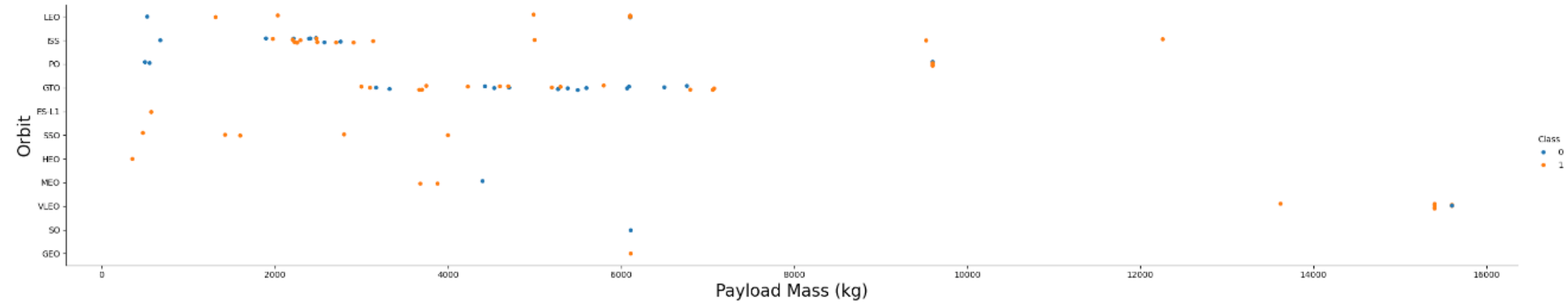
From the plot, we can see, SSO, ES-L1, GEO and HEO have. The most success rate.

FLIGHT NUMBER. VS ORBIT TYPE

The success is related to the number of flights in the LEO orbit. There is no relation for GTO orbit.

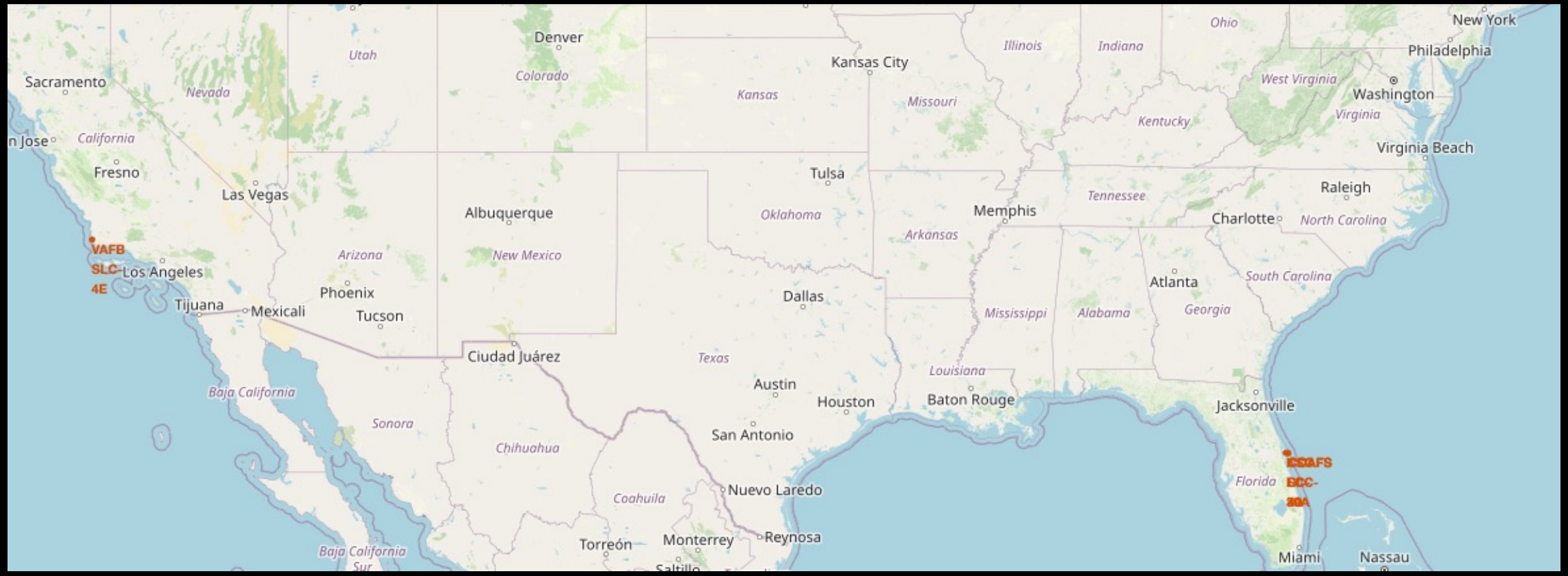


PAYLOAD VS ORBIT



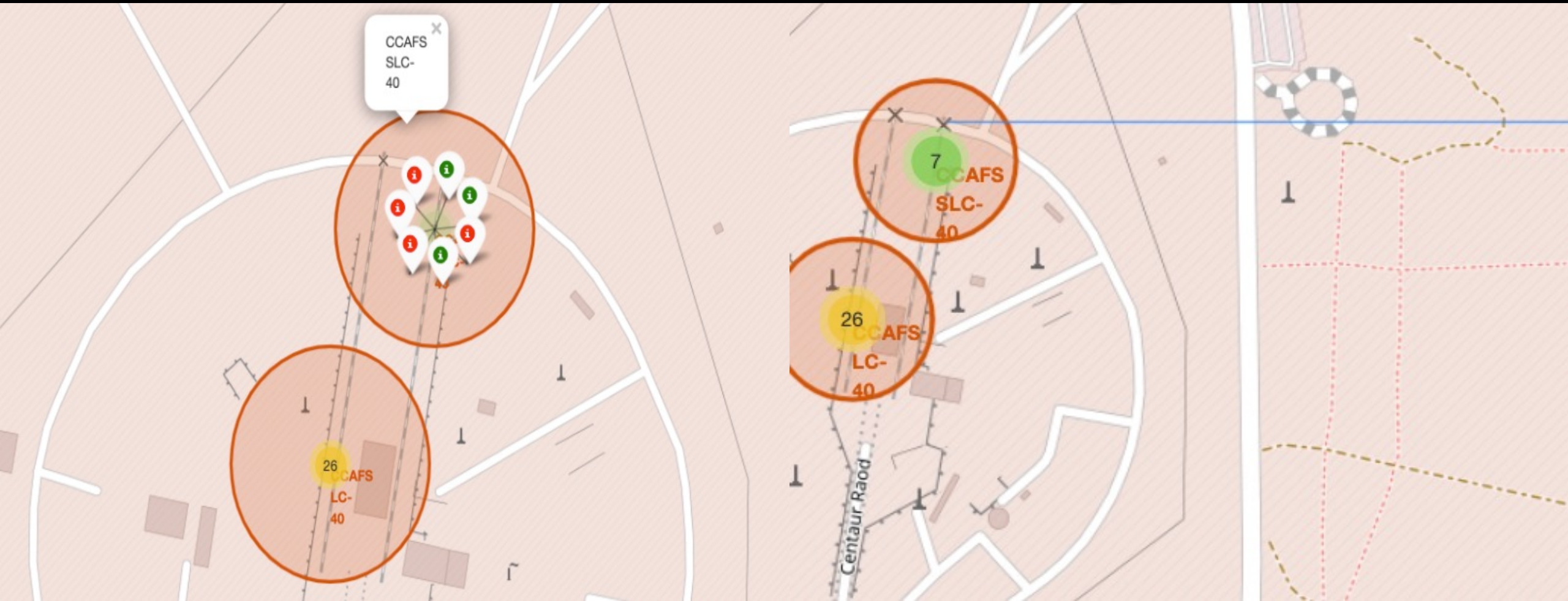
The successful landings. Are for PO,LEO, and ISS orbits.

INTERACTIVE ANALYTICS USING FOLIUM

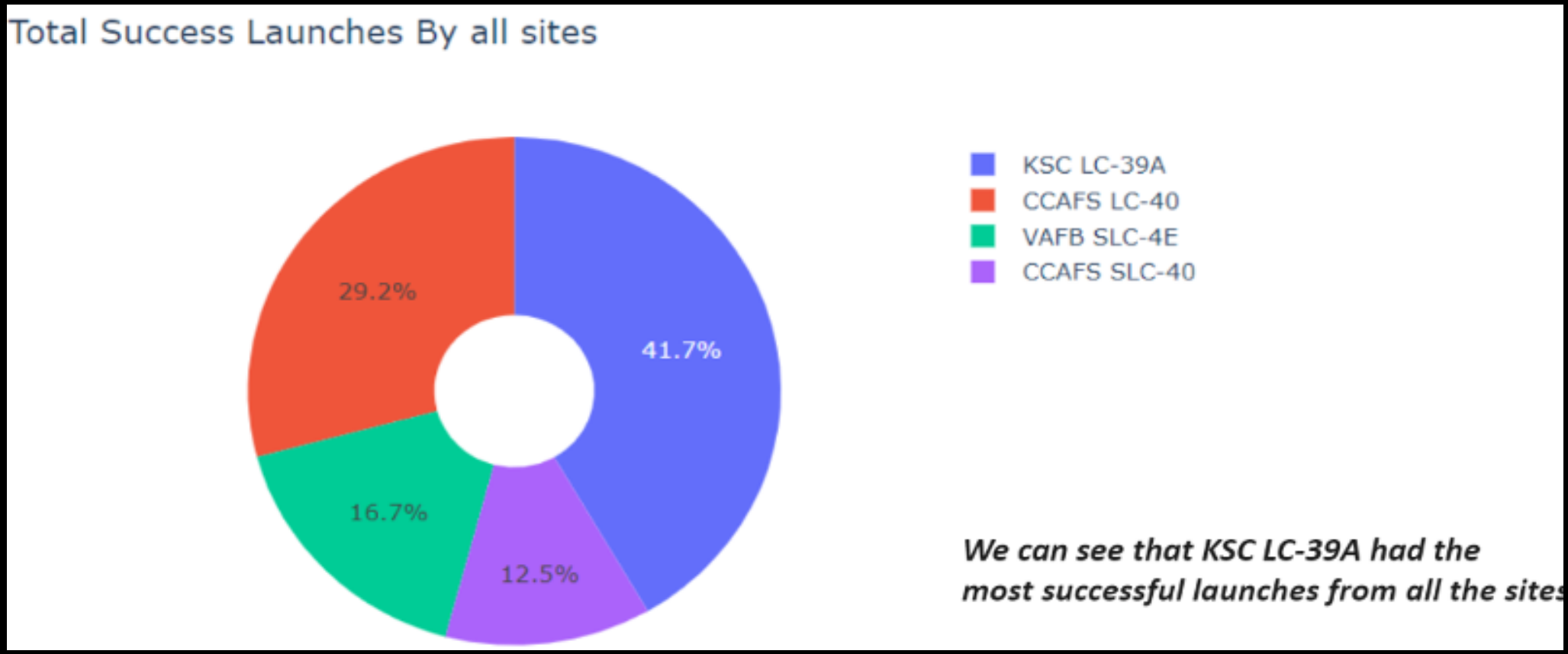


The only two launch sites of SpaceX in the world.

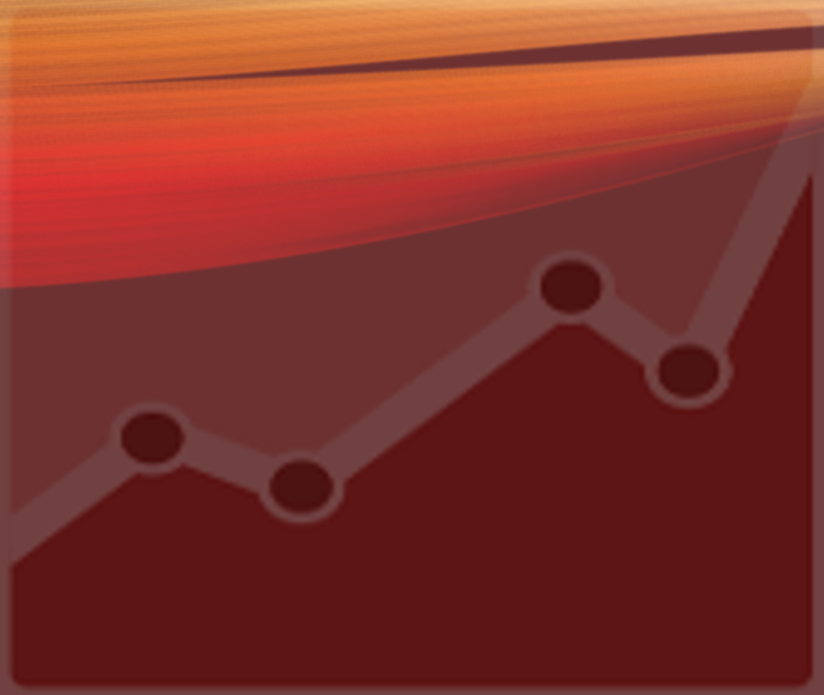
LAUNCH SITES WITH COLOR LABELS



SUCCESS ACHIEVED BY EACH LAUNCH SITE



PREDICTIVE ANALYSIS





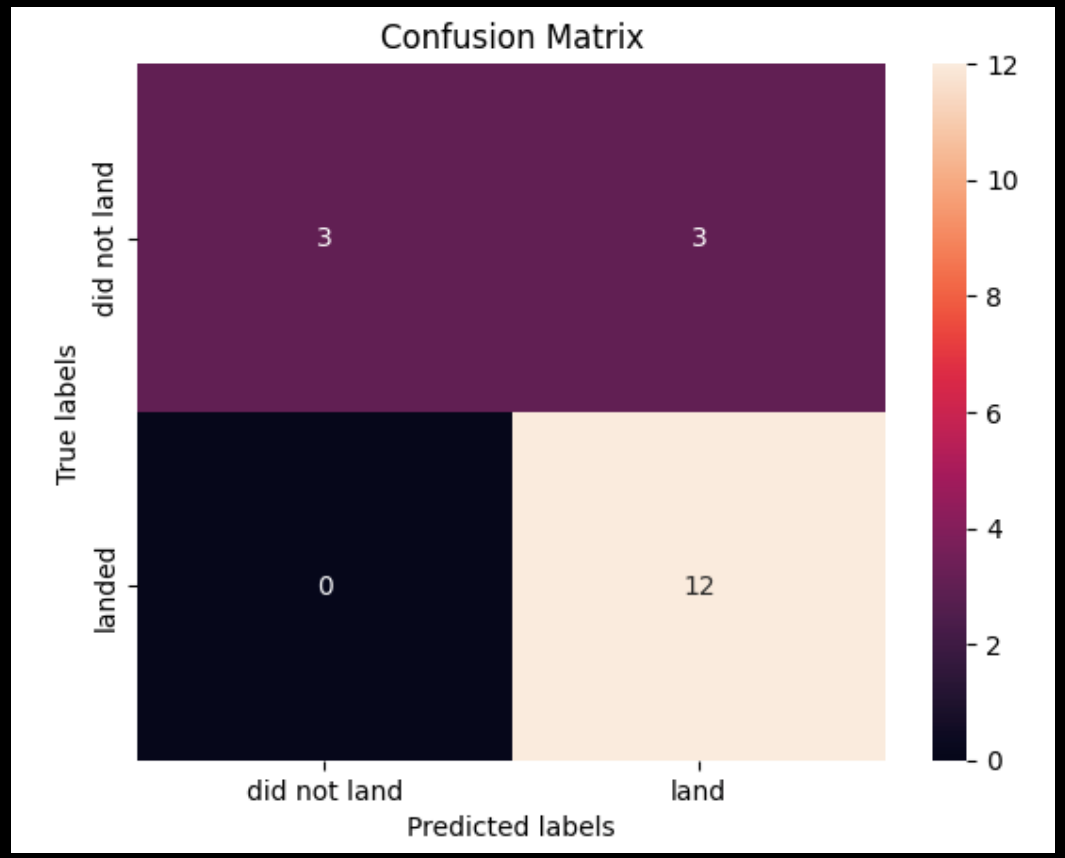

CLASSIFICATION ACCURACY

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("tree_accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'ma
'random'}
tree_accuracy : 0.8642857142857142
```

The decision tree classifier is the model with highest classification accuracy

CONFUSION MATRIX



- The matrix for the decision tree can distinguish between different classes.
- Major problem is the false positives.

SOME CONCLUSIONS

- Larger the flight amount at a launch site, greater the success rate at a launch site.
- Launch success rate started to increase from 2013-2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO have the most success rates.
- KSC LC-39A has the most successful launches of any sites.
- Decision Tree classifier is the best machine learning algorithm in this case.