IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Name: Siddhant Pathak
Date: 4th December, 2021

Table of Contents

Executive Summary

- **Summary of methodologies**

✓Data Collection

✓Data Wrangling

✓EDA with data Visualization

✓EDA with SQL

✓Building an interactive map with Folium

✓Building a Dashboard with Plotly Dash

✓Predictive analysis (Classification)

- **Summary of all results**

✓Exploratory data analysis results

✓Interactive analytics demo (using screenshots)

✓Predictive analysis results

Introduction

Project background and context

- Here, we predict whether SpaceX Falcon 9's first stage landing will be successful or not. This is crucial because SpaceX advertises Falcon 9's launch to cost 62 million dollars while other providers charge upwards of 165 million dollars. The lower cost that SpaceX can afford is only because it can make its first stage land, saving lots of money. Hence if we can predict whether the first stage will land successfully or not, we can determine the cost of the launch. This information can then be used while bidding against another company for a rocket launch.

Problems you want to find answers

- What factors influence in determining whether the launch will be successful or not?
- To determine what effect each variable /feature has to the possibility of landing successfully
- Under what parameters would SpaceX achieve optimal success in landing?

Section 1

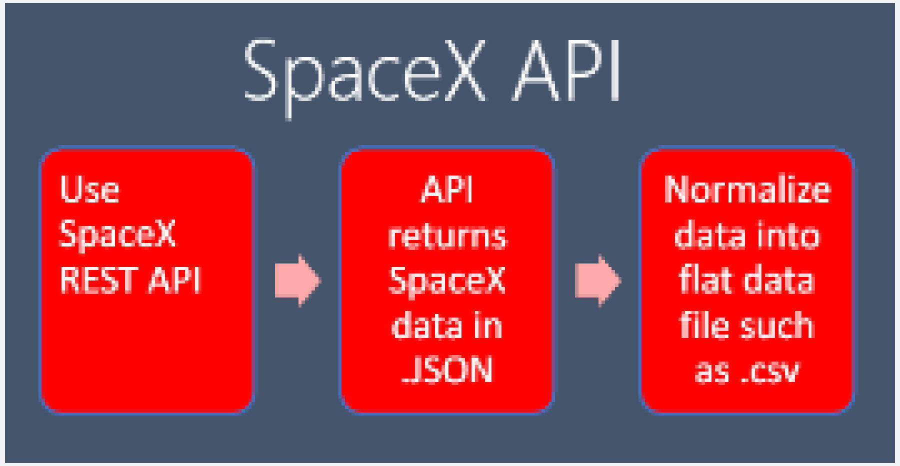
Methodology

Methodology

Executive Summary

- Data collection methodology:
    - The data was collected by sending get request to SpaceX API and by webscraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia.
- Perform data wrangling
    - Dealing with missing values, creating new columns, one hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - We test four different models of classification: Logistic Regression, Tree, SVM, KNN and chose the best performing model.

Data Collection – SpaceX API



- GitHub URL of the completed SpaceX API calls notebook

simplified flow chart

1 .Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2. Converting Response to a .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
getBoosterVersion(data)
```

4. Assign list to dictionary then dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

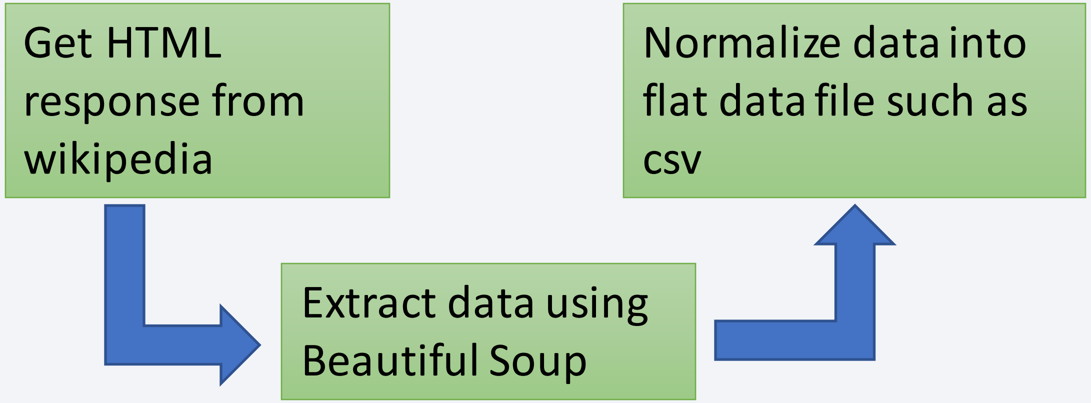
```
df = pd.DataFrame.from_dict(launch_dict)
```

5. Filter dataframe and export to flat file (.csv)

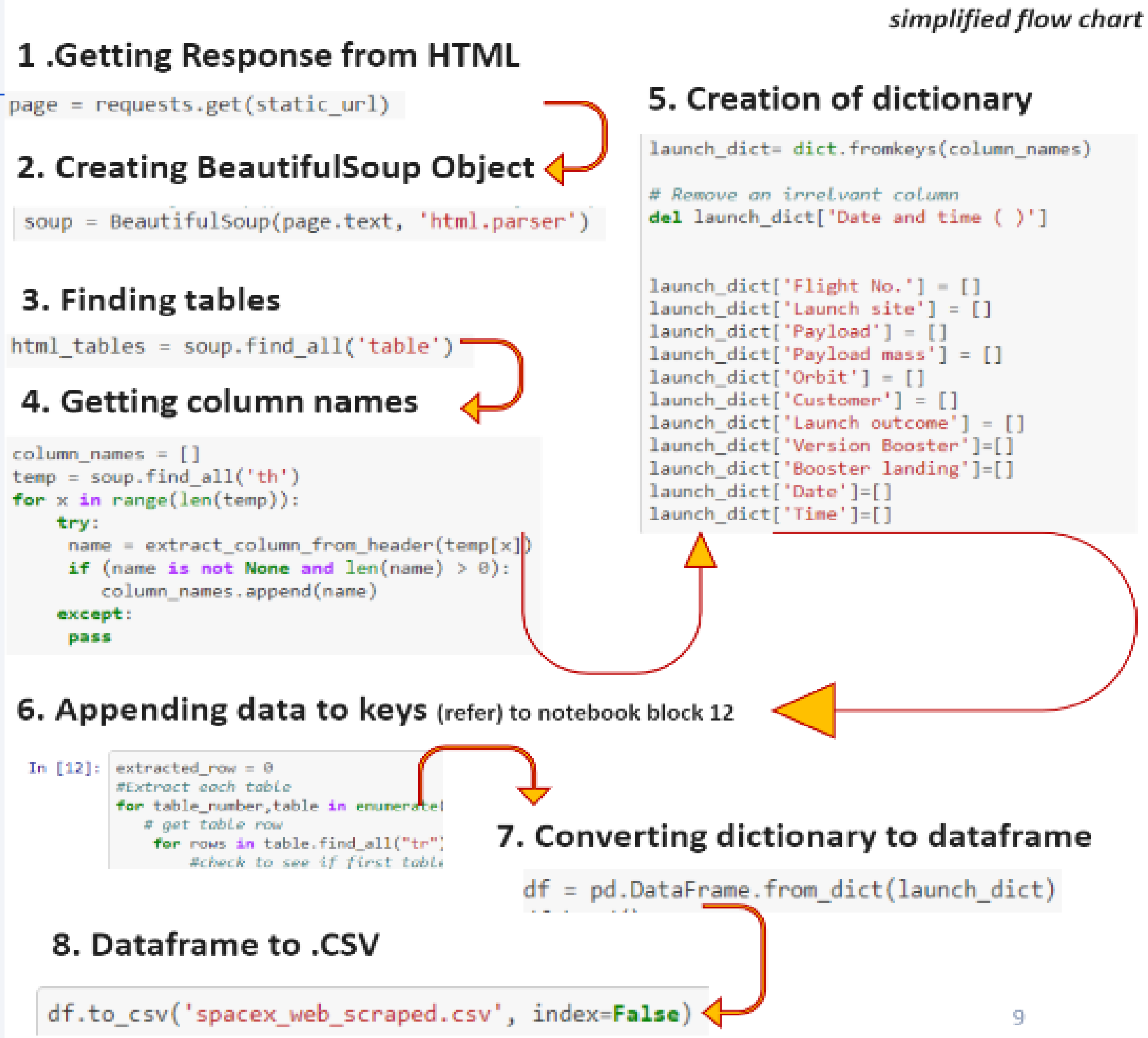
```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Web Scraping



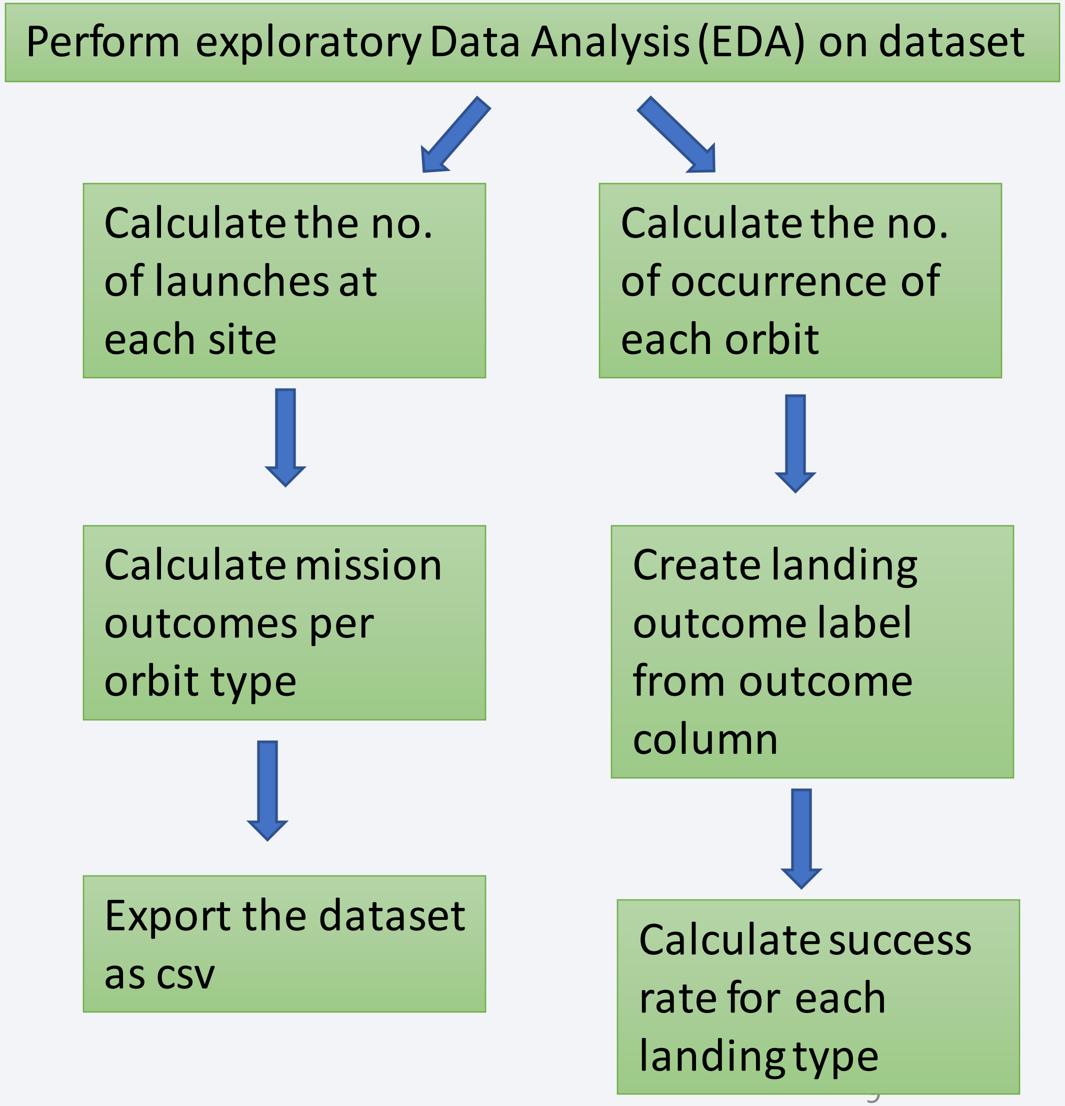
- GitHub URL of the completed web scraping notebook.



Data Wrangling

When a booster detaches itself during a rocket launch, it doesn't always land successfully and it fails in many different ways:

- False Ocean: Unsuccesfully landed to a specific region in the ocean
- True Ocean: Succesfully landed to a specific region in the ocean
- True RTLS: Successfully landed to the ground pad
- False RTLS: Unsuccessfully landed to the ground pad
- True ASDS: Successfully landed on a drone ship
- False ASDS: Unsuccessfully landed on a drone ship

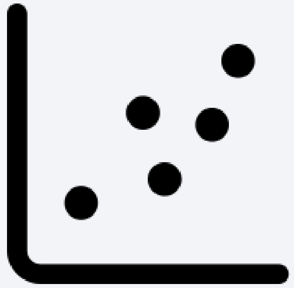


Github url to Notebook

EDA with Data Visualization

Scatter graphs used for exploration:

- Flight Number vs. Payload mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload mass



- Github url to Notebook

- Scatter plots help us visualize, how much one variable is affected by another, i.e. it shows the correlation between two variables.

- Apart from scatter plot we also used bar graph (Mean vs. Orbit) and Line graph (Success Rate vs. Year) to explore the trends of the given variables.

EDA with SQL

SQL queries that were performed to investigate/explore the dataset:

- ❑ Display the names of the unique launch sites in the space mission
- ❑ Display 5 records where launch sites begin with the string 'CCA'
- ❑ Display the total payload mass carried by boosters launched by NASA (CRS)
- ❑ Display average payload mass carried by booster version F9 v1.1
- ❑ List the date when the first successful landing outcome in ground pad was acheived.
- ❑ List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- ❑ List the total number of successful and failure mission outcomes
- ❑ List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- ❑ List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- ❑ 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

- GitHub URL of Notebook

Build an Interactive Map with Folium

- **To visualize the Launch Data into an interactive map.** We took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.
- **We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1** with Green and Red markers on the map in a MarkerCluster()
- **Using Haversine's formula we calculated the distance** from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks.

Github URL to Notebook

Build a Dashboard with Plotly Dash

- Plotly is a python wrapper on the javascript library 'leaflet'. It enables us to interact with our data visualizations and also host it as a website that stays up 24*/7, this is done using Flask and Dash web framework.
- What we have included:-

1. Pie Chart: That shows number of launches from each launch site as well as number of successful and failed launches from those sites.
2. Scatter Graph: Relationship between the success of a launch (Outcome) and Payload (in kg) for different versions of boosters.

Github URL to Notebook

Predictive Analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas and Transform Data

Split our data into training and test data sets

- Standardize/Normalize all the parameters

Github URL to Notebook

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms and plot Confusion Matrix for the same

IMPROVE MODEL

- Use feature engineering and model tuning for the same.

FIND THE BEST PERFORMING CLASSIFICATION MODEL

- Find the model with best accuracy score and also the best performing parameters for that model

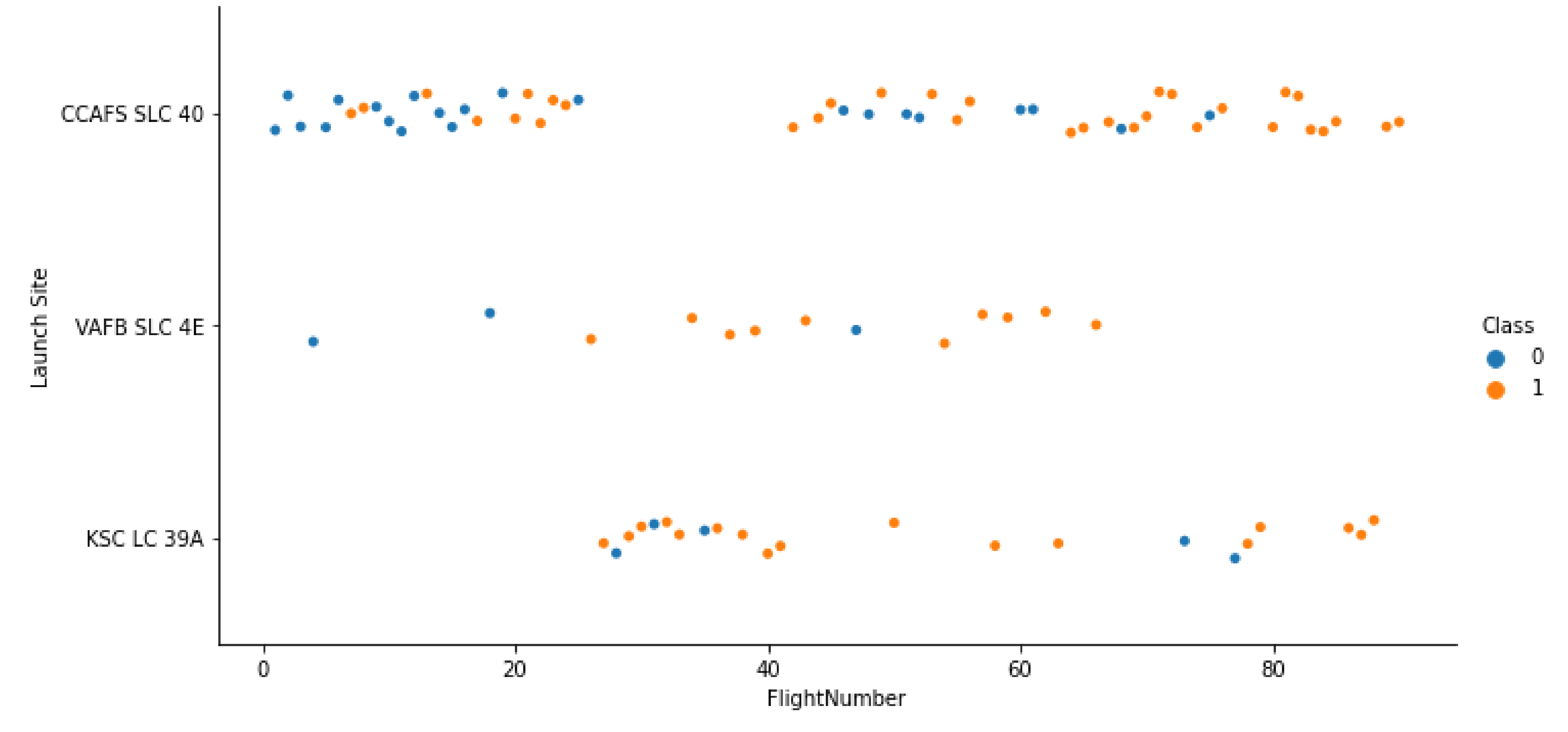
Results

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

Section 2

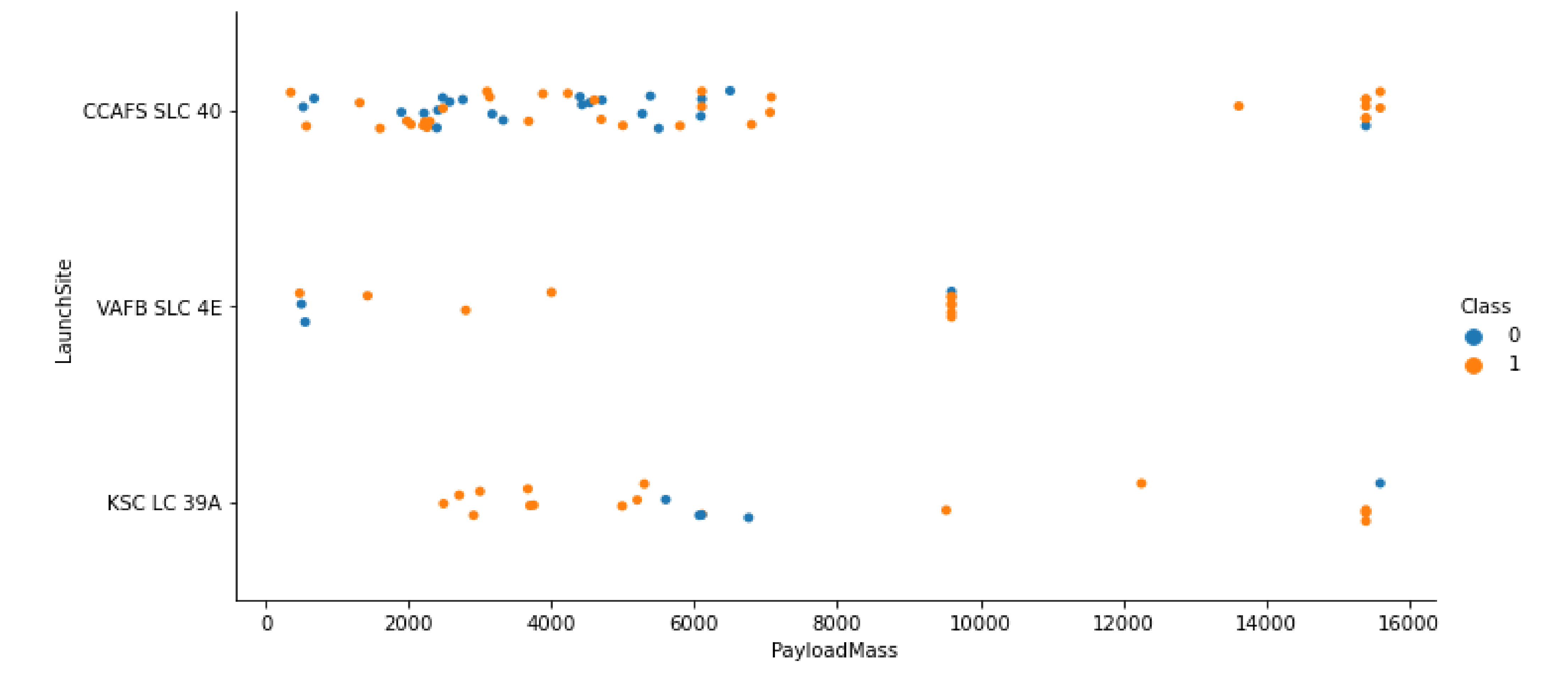
Insights drawn from EDA

Flight Number vs. Launch Site



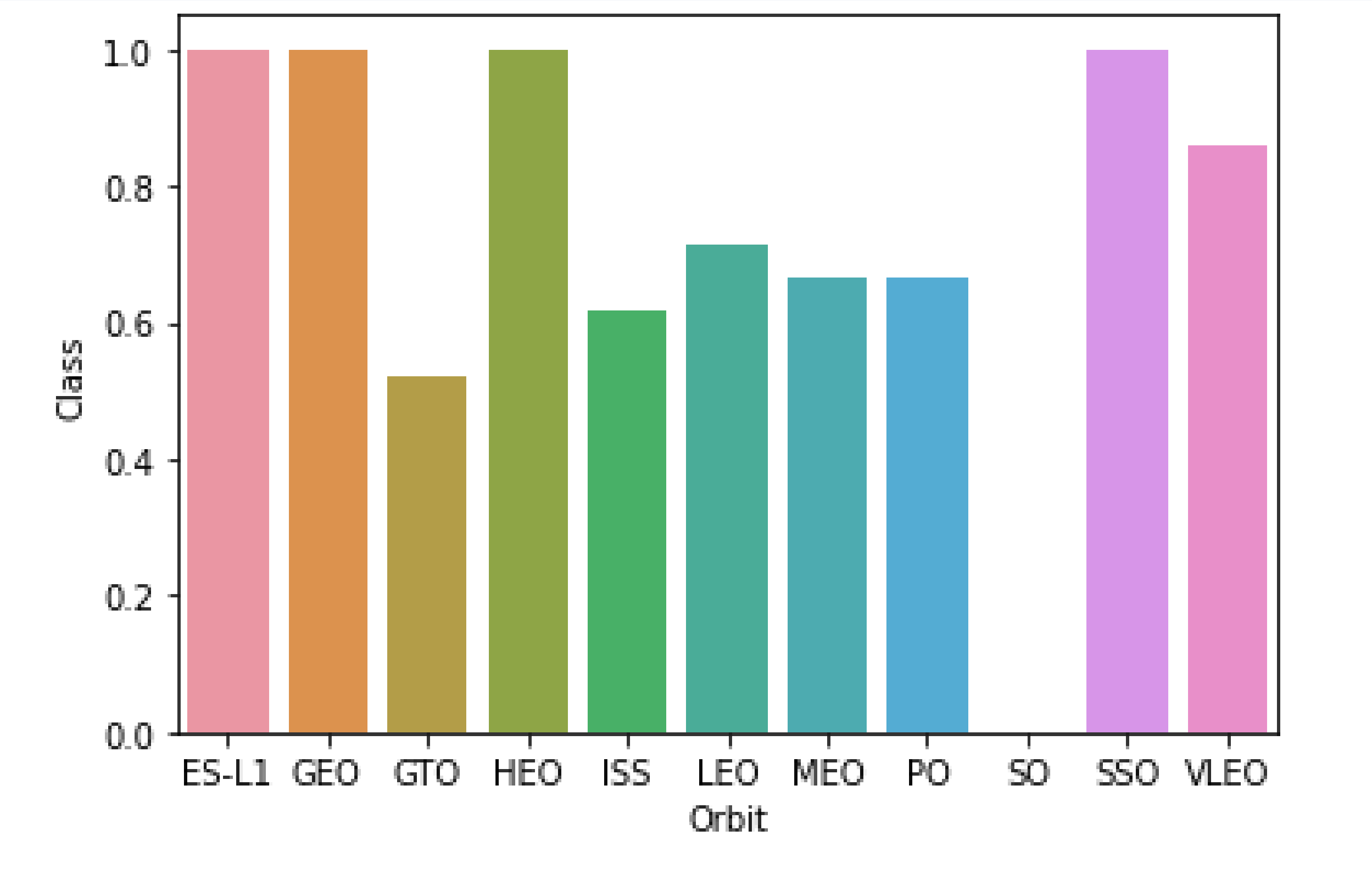
- The more amount of launches at a site, higher the success rate of that launch site (CCAFS SLC40)

Payload vs. Launch Site



- There is no clear pattern to be found using this visualization to come to a conclusion if the Launch Site is dependant on Pay Load Mass for a successful launch or not.

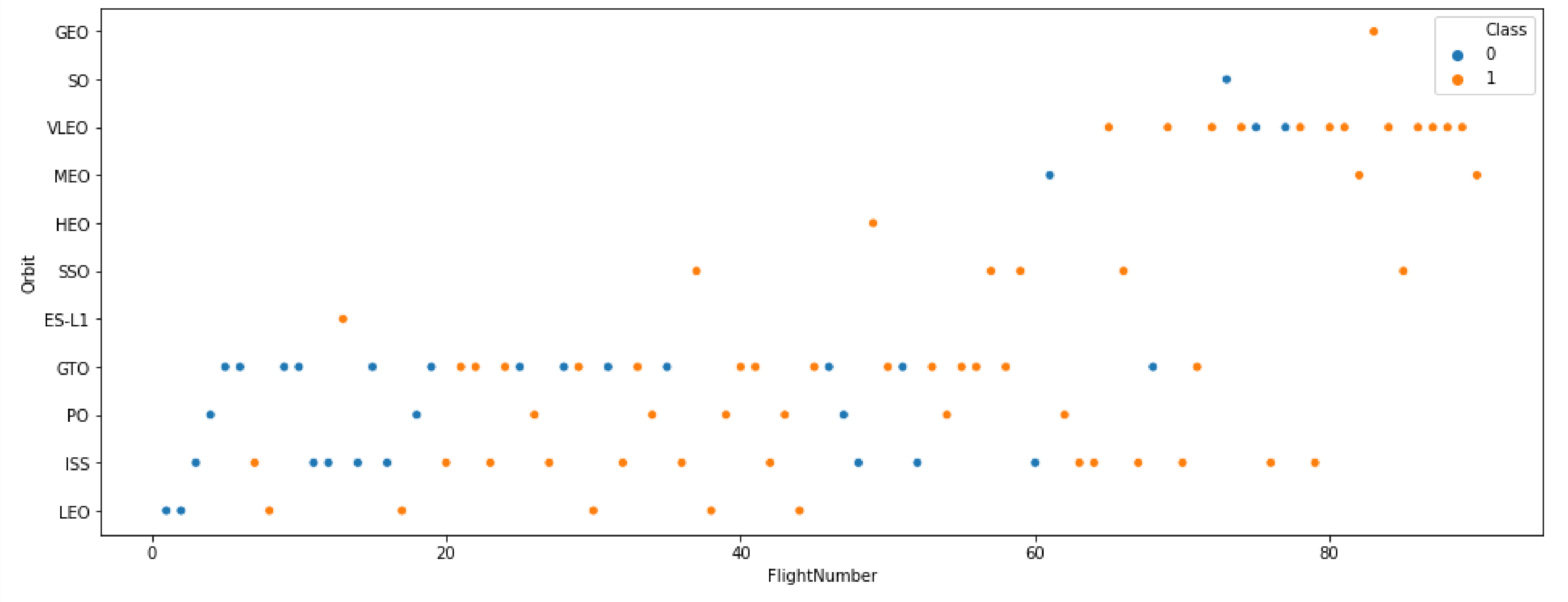
Success Rate vs. Orbit Type



As seen from the bar graph:

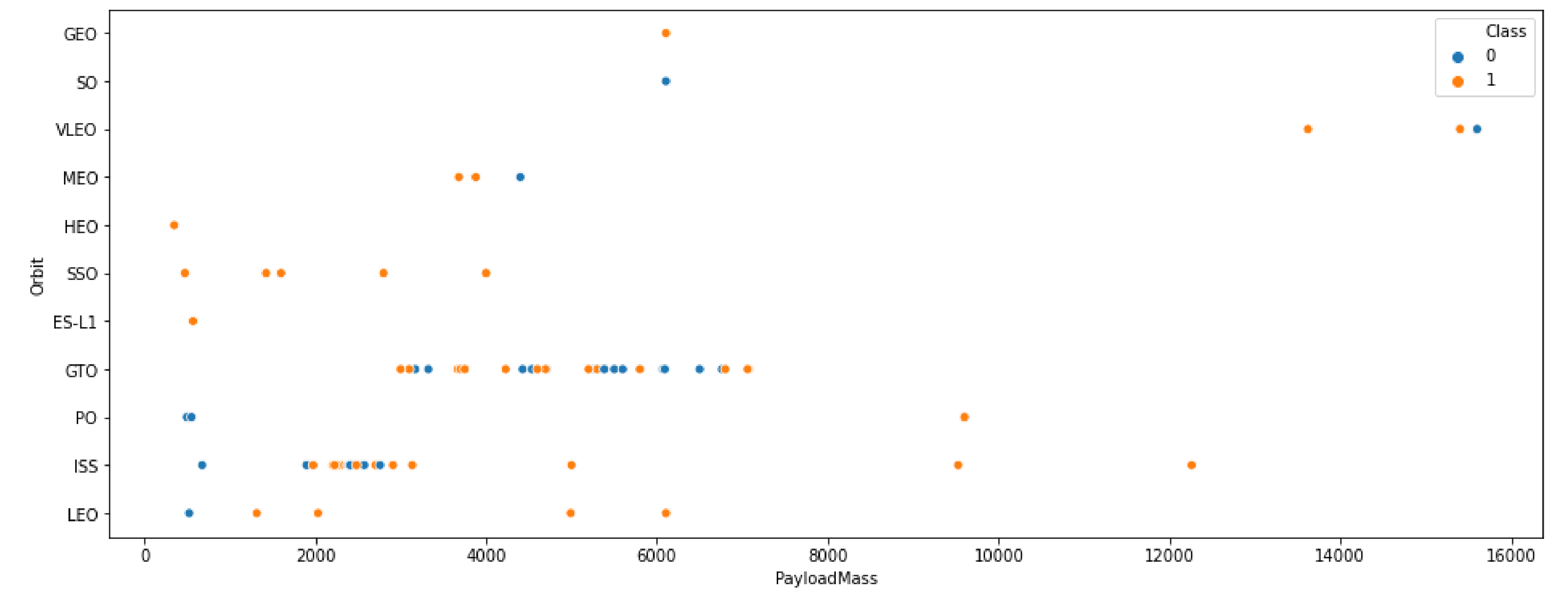
Orbits GEO,HEO,SSO,ES-L1 have the best Success Rates.

Flight Number vs. Orbit Type



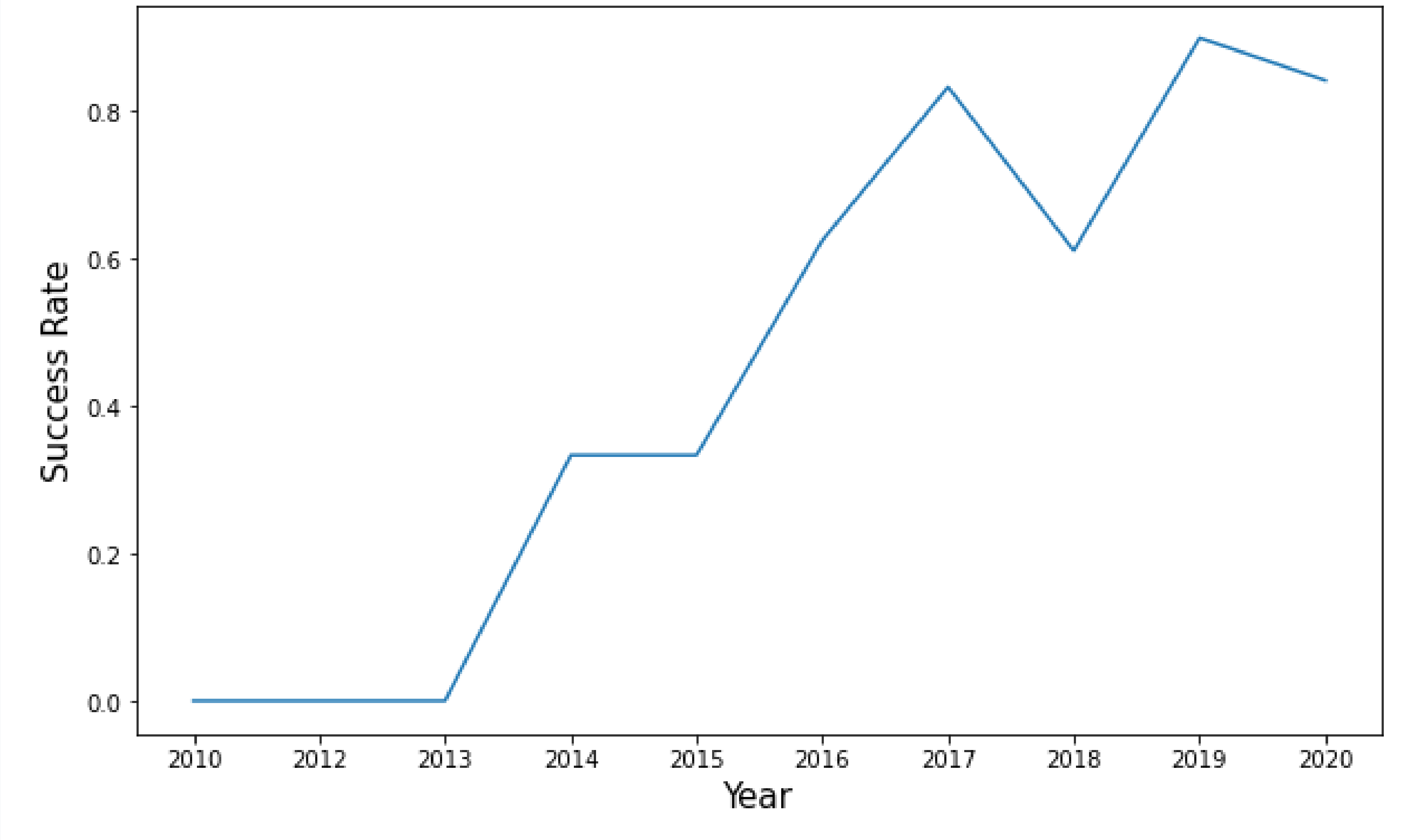
- It seems from the graph that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



- The success rate of landings have steadily increased from 2013 to 2020

All Launch Site Names

- **Query:**
  `%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;`

- **Result:**

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

- **Explanation:** Using the word DISTINCT in the column Launch_Site means it will show only unique values for the sites.

Launch Site Names Begin with 'CCA'

- **Query:** `%%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;`
- **Result:**

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- **Explanation:** Using 'LIMIT 5' means it will only show 5 records from SPACEXTBL and LIKE 'CCA%' means the launch sites it chooses will start from 'CCA'

Total Payload Mass

- **Query:** %%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

- **Result:**

1
45596

- **Explanation:** The function 'SUM' does summation over all the values in the column 'PAYLOAD_MASS_KG'.
  'WHERE' clause filters the customers to only have 'NASA (CRS)'.

Average Payload Mass by F9 v1.1

- **Query:** %%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';

- **Result:**

1
2534

- **Explanation:** The function 'AVG' returns the average value of the entire column 'PAYLOAD_MASS__KG_'
  'WHERE' clause filters dataset to contain only rows with boosters 'F9 v1.1'

First Successful Ground Landing Date

- **Query:**
  %%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';

- **Result:**

1
2015-12-22

- **Explanation:** The function 'MIN' returns the minimum/earliest date.
  'WHERE' clause filters the rows to contain only successful landings on ground.

Successful Drone Ship Landing with Payload between 4000 and 6000

- **Query**: %%sql SELECT DISTINCT(BOOSTER_VERSION), LANDING__OUTCOME, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
- **Result**:

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600

- **Explanation**: 'WHERE' clause filters data to contain only successful launches on the ship. 'AND' clause provides additional filter to contain boosters that has weight between 4000 – 6000 kg

Total Number of Successful and Failure Mission Outcomes

- **Query**: 1) %%sql SELECT COUNT(LANDING__OUTCOME) AS SUCCESSFUL_MISSIONS FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success%';
  2) %%sql SELECT COUNT(LANDING__OUTCOME) AS FAILURE_MISSIONS FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Failure%';

- **Result**:

successful_missions
61

failure_missions
10

- **Explanation**: The 'COUNT' function counts the total number of rows returned by the query.
  'WHERE' clause finds the landing outcomes that were either a success or a failure.

Boosters Carried Maximum Payload

- **Query**: %%sql SELECT DISTINCT(BOOSTER_VERSION), PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

- **Result**:

- **Explanation**: The function 'DISTINCT' returns only unique boosters used for the launch. The 'WHERE' clause here returns the result from another query (subquery) which is the maximum payload that was carried.

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- **Query**: %%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, YEAR(DATE) AS DATE_YEAR FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'

- **Result**:

landing__outcome	booster_version	launch_site	date_year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

- **Explanation**:
'YEAR(DATE)' returns only years from the date column and renames ir 'date_year'. The 'WHERE' clause returns the result of a subquery that returns failed landing on drones for the year 2015.

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

- **Query**: %%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS COUNT FROM SPACEXTBL WHERE DATE BETWEEN '2016-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY COUNT DESC
- **Result**:

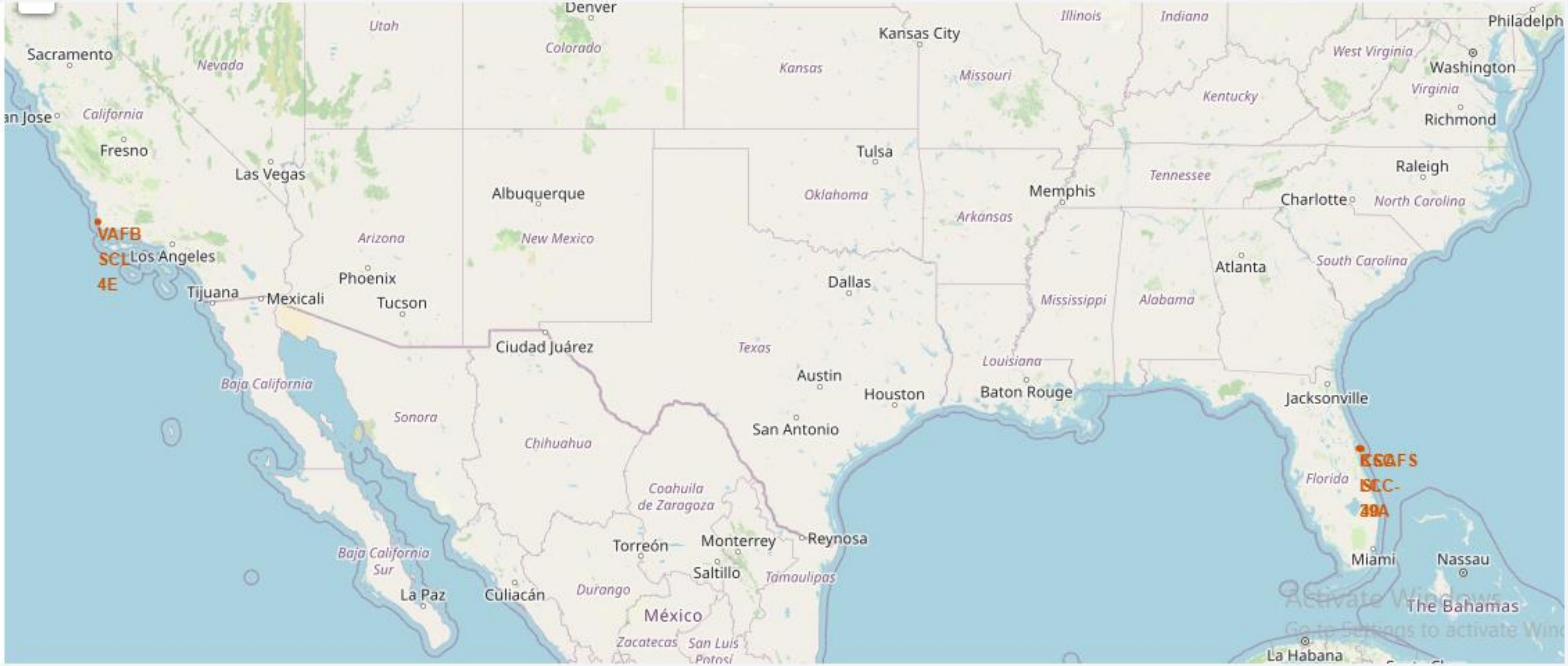
landing__outcome	COUNT
Success (drone ship)	2
Success (ground pad)	2
Failure (drone ship)	1
No attempt	1

- **Explanation**: The 'BETWEEN' function in 'WHERE' clause returns rows that has dates between '2016-06-04' and '2017-03-20'.
'GROUP BY' function combines all the rows based on the function (COUNT)
'ORDER BY' function orders them in descending order using function 'DESC'.

Section 4

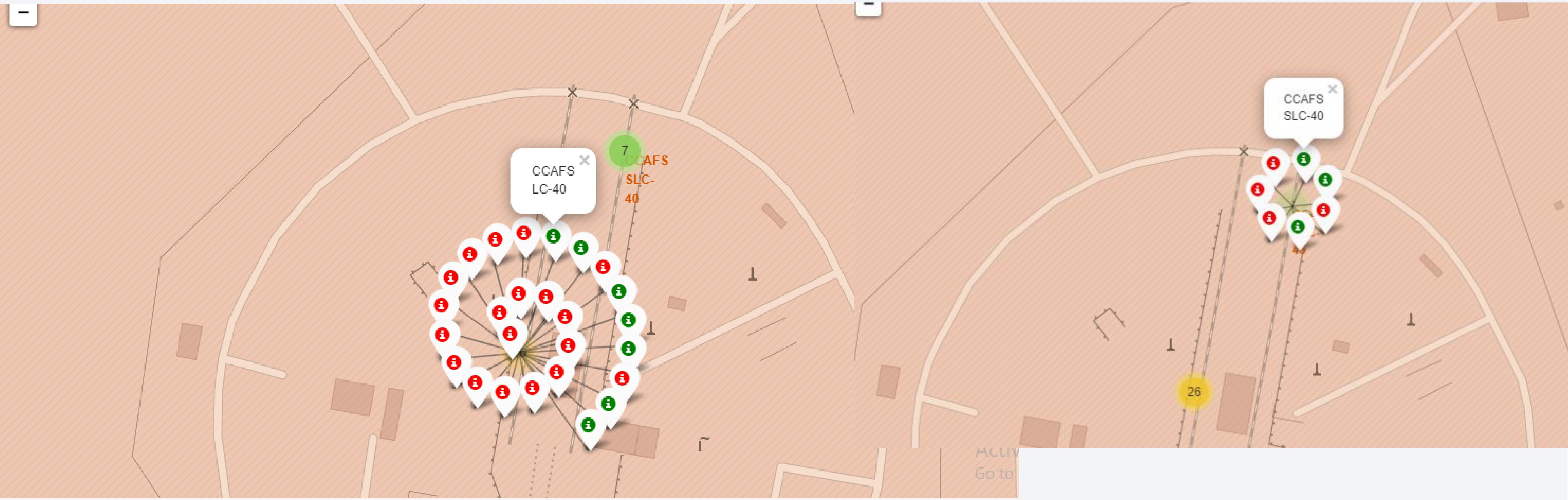
Launch Sites Proximities Analysis

Folium Map for global sites



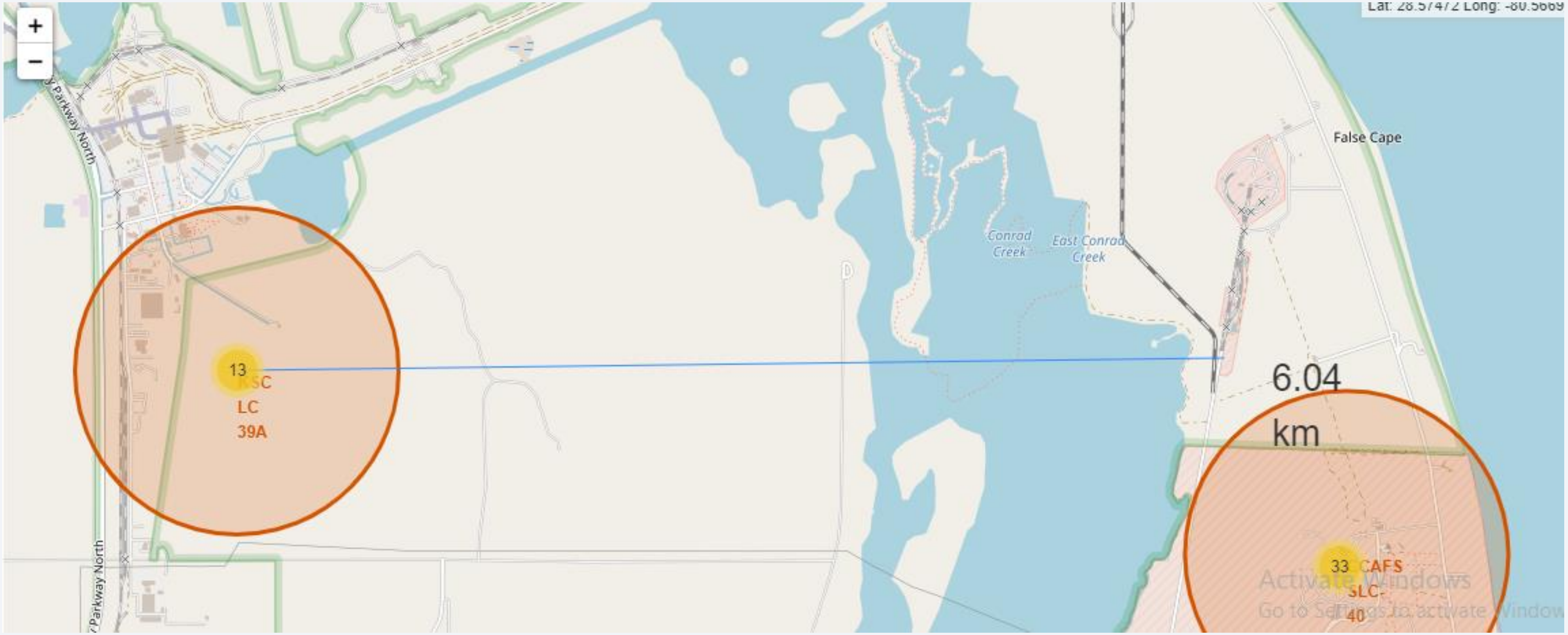
- As shown in the above figure, the SpaceX launch sites are in the USA coasts of Florida and California.

Folium Map: Color labeled Markers



- Green Markers = Successful launches
- Red Markers = Failed Launches

Folium Map: Finding distance

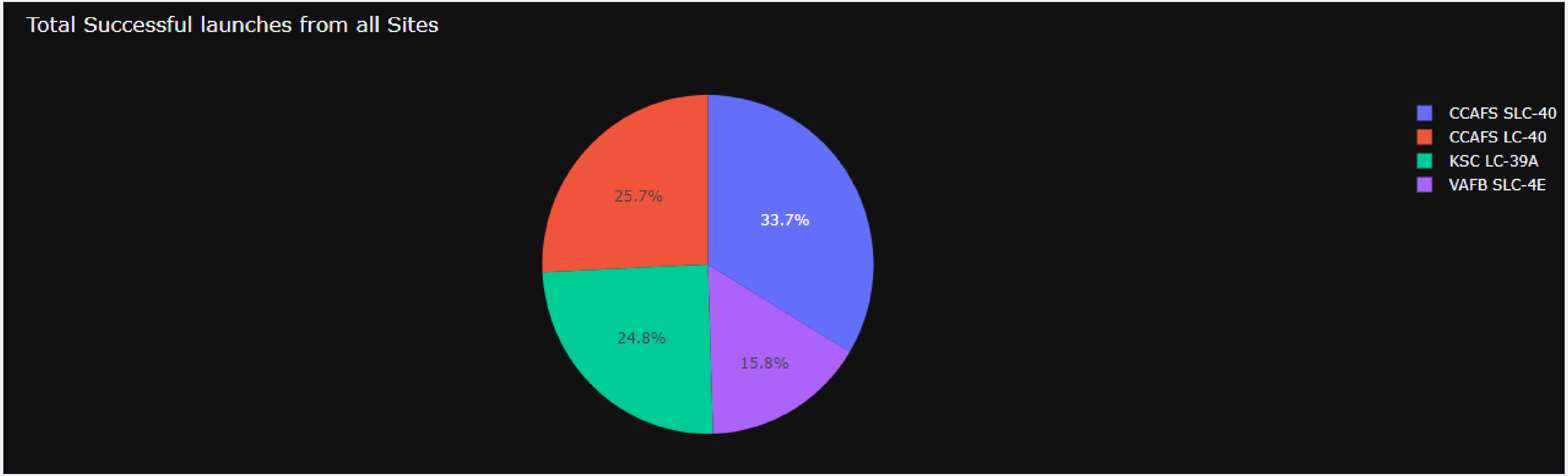


- Here we find the distance from Launch Site (KSC LC 39A) to 'nearest railway line' using Haversine formula.

Section 5

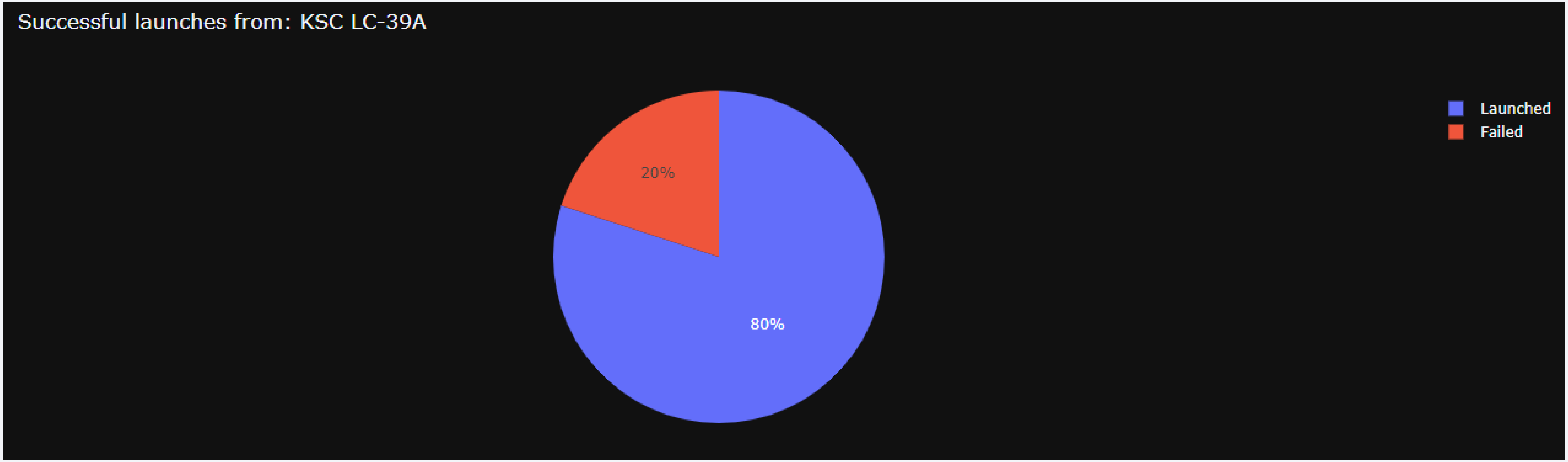
Build a Dashboard with Plotly Dash

Success counts for all the launch sites



- We can see that CCAFS SLC-40 had the most number of successful launches.

Launch site with highest success ratio



- KSC LC-39A is the most successful launch site, it had 80% of its launches successful

Payload vs. Launch outcome scatter plot for heavy payloads (5000-10000 kg)

- It seems the successfull launch ratio for heavy payload is less

Payload vs. Launch outcome scatter plot for light payloads (0-5000 kg)

- The successfull launch ratio for light payload is more than that of heavy payload.

Section 6

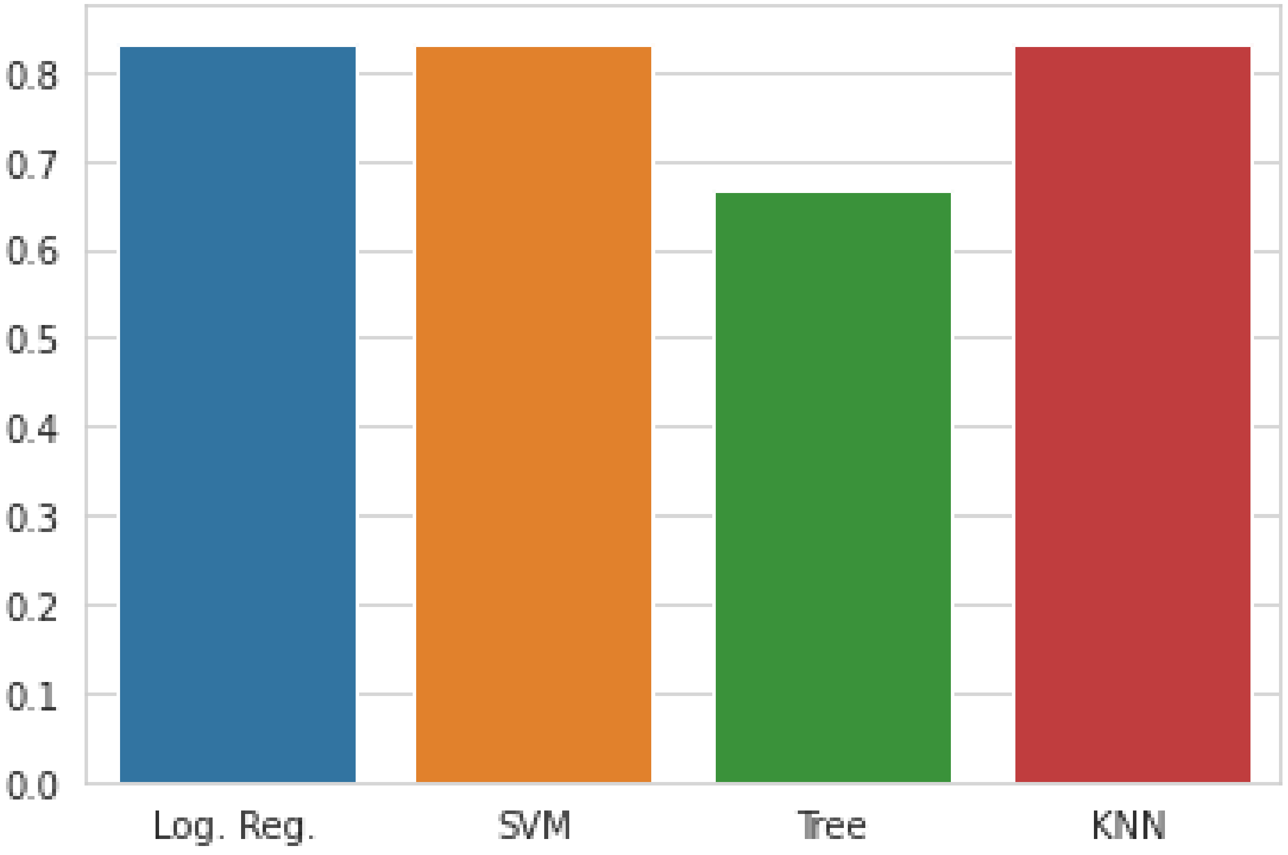
Predictive Analysis (Classification)

Classification Accuracy

- As we can see, the classification accuracy of all but Tree algorithm is extremely close at 83.33%

- The best algorithm is SVM with the following parameters:

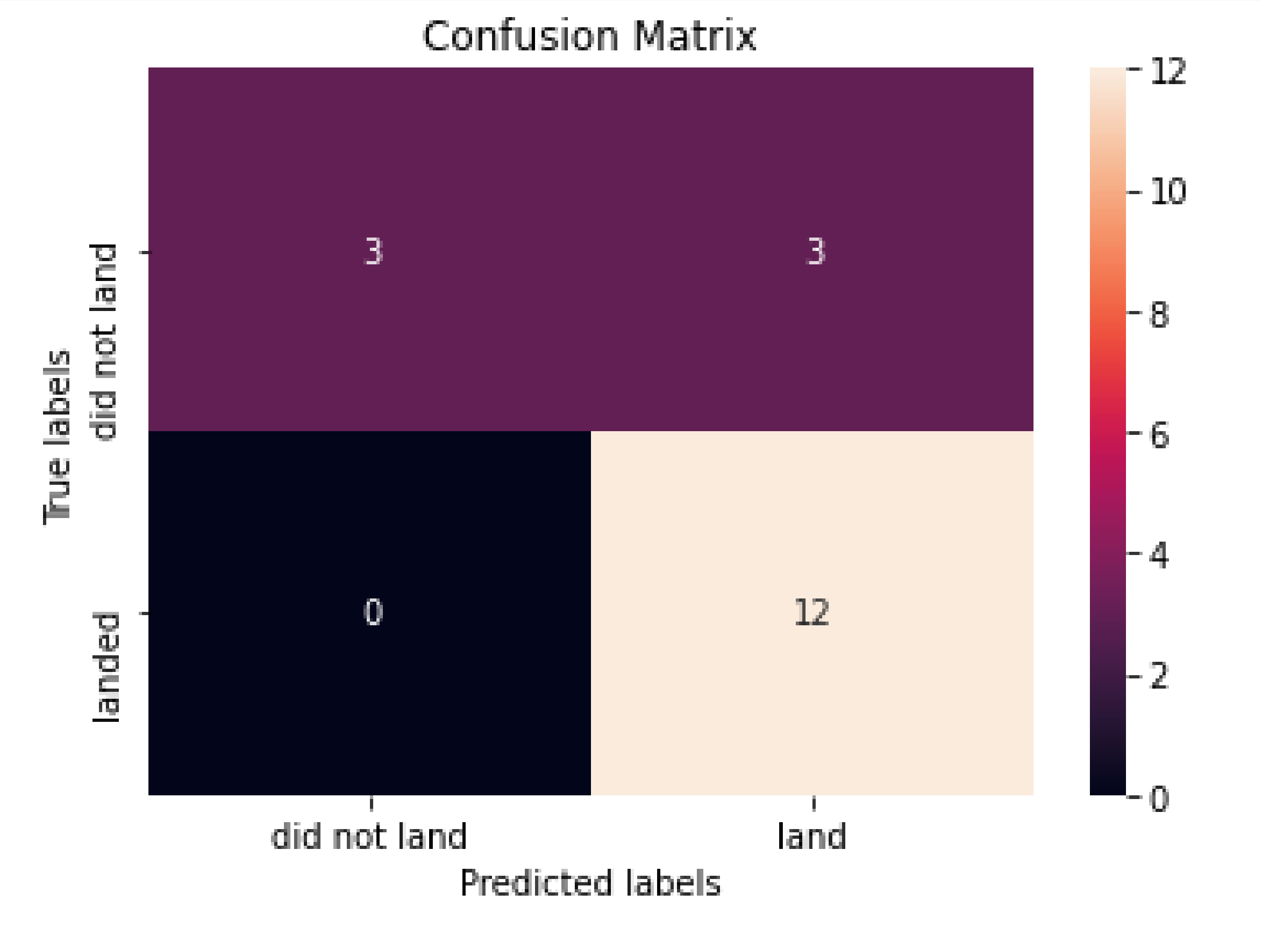
- Kernel = sigmoid, C = 1.0 and gamma = 0.0316



Confusion Matrix

- On the right is the confusion matrix for SVM classification algorithm.

- As we see the False Negative = 0, hence the problem is that of False positives.



Conclusions

- The SVM Classifier Algorithm is the best for Machine Learning for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rate of launches has steadily increased for SpaceX over the years (2013-2020) hence it seems that they will eventually perfect the launches.
- We can see that KSC LC-39A had the most successful launches of all the launch sites.
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate.

Thank you!