IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Iael Perez
22/06/2022

# Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

# Executive Summary

- Summary of methodologies
  - Data collection using API and wrangling
  - Exploratory Data Analysis including SQL
  - Data Visualization (Graphs, Map with Folium and Dashboard with Plotly Dash)
  - Predictive Analysis

- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

# Introduction

- Project background and context
    - The objective of this project is to predict if the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX can reuse the first stage. Therefore, if it is determined that the first stage will land, it would be possible to determine the cost of a launch.
- Problems you want to find answers
    - On which variables does the success of the first stage landing depend?
    - What is the relationship between rockets variables and the success or failure of a landing?
    - Based on the information available, how good are the predictions of landing success?

Section 1

# Methodology

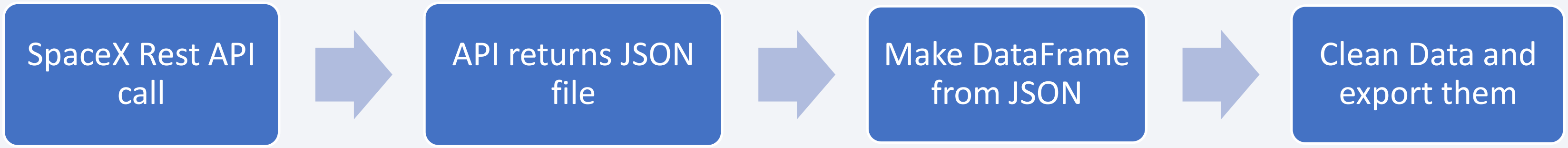
# Methodology

## Executive Summary

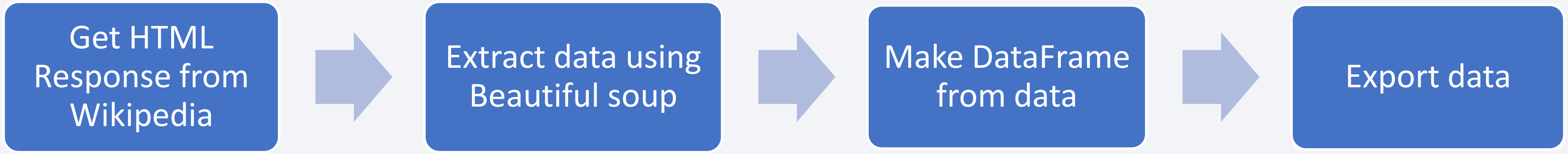
- Data collection methodology:
  - SpaceX REST API
  - Web scraping from Wiki pages
- Perform data wrangling
  - Clean the original data (null values, irrelevant data)
  - One Hot Encoding
- Perform exploratory data analysis (EDA) using different visualization tools and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, SVM, Tree Classifier, and KNN models have been built and evaluated (train and test accuracy, score, confusion matrix)

# Data Collection

- Rest SpaceX API
  - Data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome were obtained from the API (URL api.spacexdata.com/v4/)




- Web scrapping from Wikipedia
  - Data about launches, landings, and payloads were obtained from https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches.



# Data Collection – SpaceX API

Link

**1-Getting Response from API**

- spacex_spacex_url=https://api.spacexdata.com/v4/launches/past url=https://api.spacexdata.com/v4/launches/past
- response = requests.get(spacex_url)

**2- Convert Response to JSON into DataFrame**

data = pd.json_normalize(response.json())

**3-Cleaning and Transforming Data**

- getBoosterVersion(data)
- getLaunchSite(data)
- getPayloadData(data)
- getCoreData(data)

**4-Create dictionary with data and store in 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(launch_dict)`

**5-Export dataframe to file**

data_falcondata_falcon9.to_csv('dataset_part_1.csv', _1.csv', index=False)

# Data Collection - Scraping

Link

### 1-Getting Response from HTML

- static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- response =requests.get(static_url)

### 2-Creating a BeatifulSoup Object

```
soup = BeautifulSoup(response.text,'html5lib')
```

### 3-Finding tables

- html_tables =soup.findAll('table')

### 4- Getting the tables columns

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name)>0:
        column_names.append(name)
```

### 5-Creating dictionary to store data

```
launch_dict= dict.fromkeys(column_names)
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

### 6-Filling up the dictionary with records from the tables and converting it into a DataFrame

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

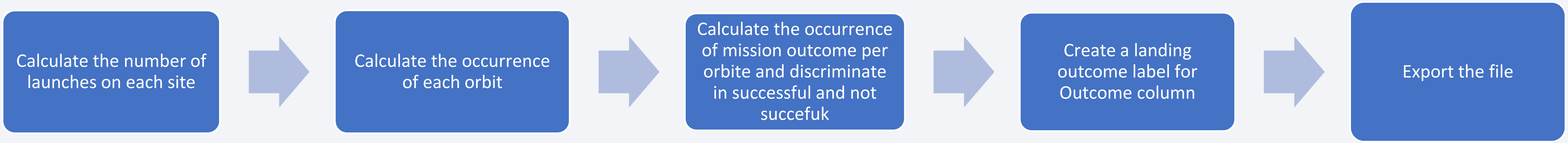
### 6-Exporting the DataFrame

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

# Data Wrangling

Link

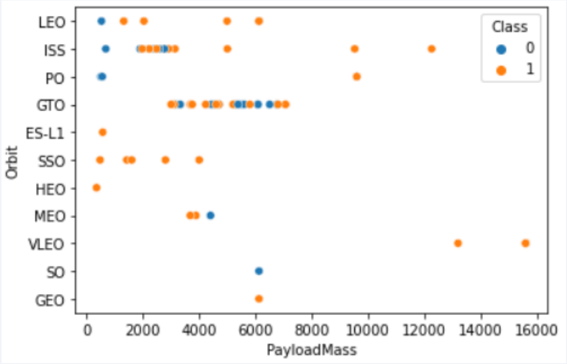
- Data presents different cases where the booster did not land successfully identify which different string labels. The idea was to simplify them into a categorical variable.
  - 1 successful
  - 0 failure



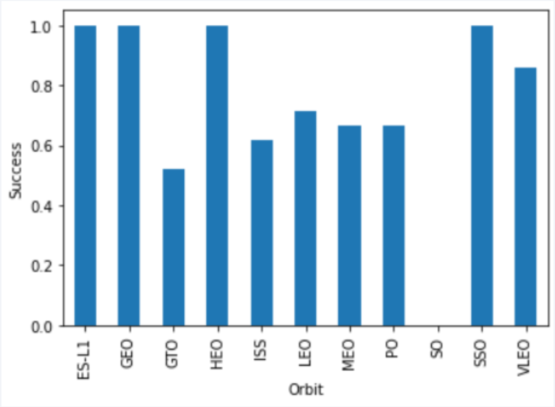
# EDA with Data Visualization

Link

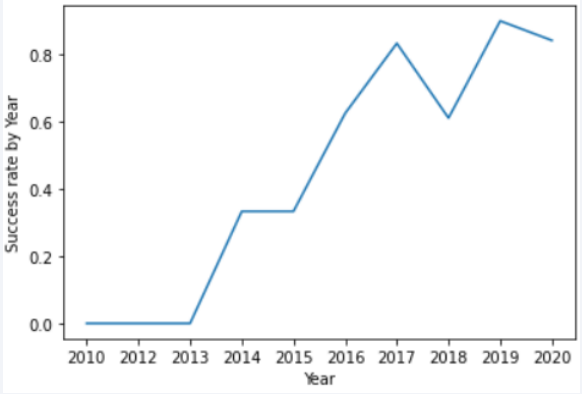
- ***Scatter plot** can suggest relationships (correlation) between variables.*
  - PayloadMass vs Flight Numbers
  - Flight Numbers vs Lunch Site
  - Payload vs Launch Site
  - Flight Number vs Orbit type
  - Payload vs Orbit type




- *Bar Graph is used to compare data among categories.*
  - Success vs Orbit type




- *Line plot is used to track changes over short and long periods of time.*
  - Launch success vs Years



# EDA with SQL

Link

## SQL queries performed:

- 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 achieved.
- List the names of the boosters which have success in drone ships and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failed mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in the year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

# Build an Interactive Map with Folium

Link

- Maps objects added to folium map centred on NASA, Texas
  - Marker all lunch sites with the corresponding name
  - Marker the success (green) and failed (red) lunches for each site
  - Line with the distances between a launch site to its proximities (coastline, railways, highways)
- These maps help to understand the problem and the data. It is an easy way to visualize the situation.

# Build a Dashboard with Plotly Dash

Link

- Dashboard has
  - Dropdown which allows selecting one specific launch site or all.
  - Pie plot that shows total success/failure for the selected launch site.
  - Rangeslider that allows choosing a mass payload range.
  - Scatter plot of success vs payload showing their relationship.

# Predictive Analysis (Classification)

Link

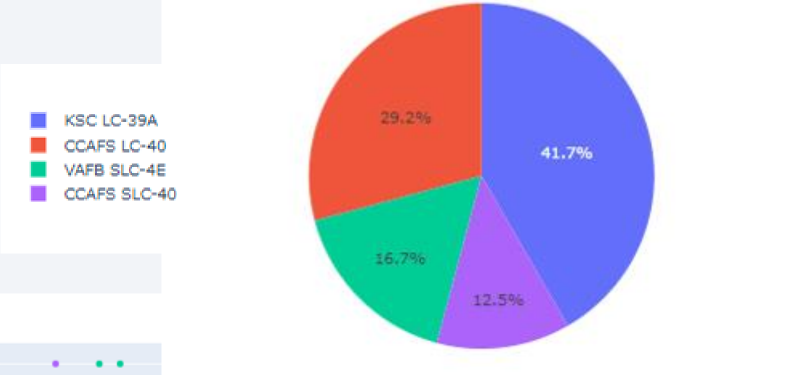
- Data Preparation
  - Load preprocessed data
  - Normalization data
  - Divide into train and test data
- Model Preparation
  - Selecting 4 models (Logistic Regression, SVM, KNN and Tree Classification)
  - Set parameters for each model, and use GridSearchCV
  - Training models
- Model Evaluation and comparison
  - Get the best parameters for each model
  - Calculate accuracy with test dataset and plot confusion Matrix
  - Comparison of  the accuracy of the models

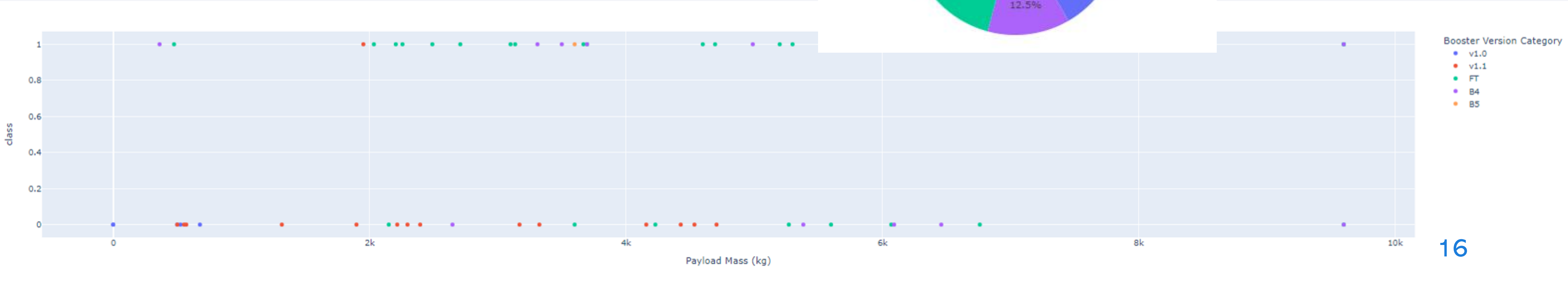
# Results

- All models have similar accuracy (80 %)
- The evolution of success rates show an increase over the years
- Different launch sites have different success rates

**Examples from dashboard**

*Success pie chart all sites*

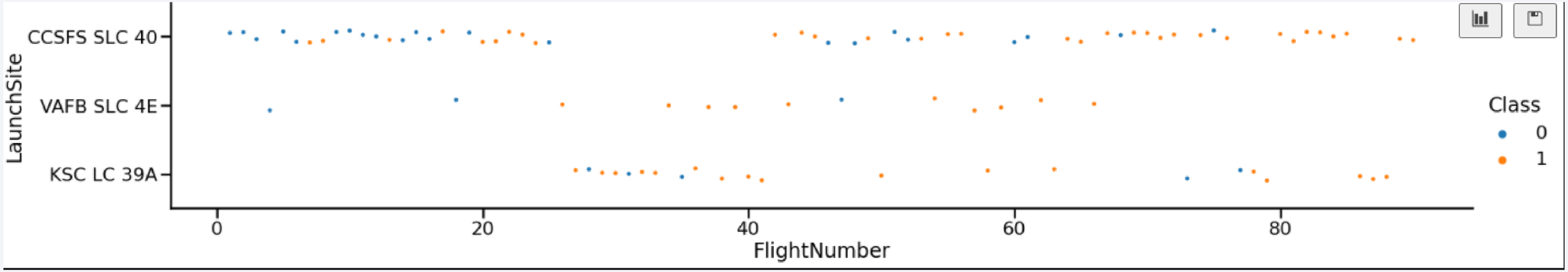


Section 2

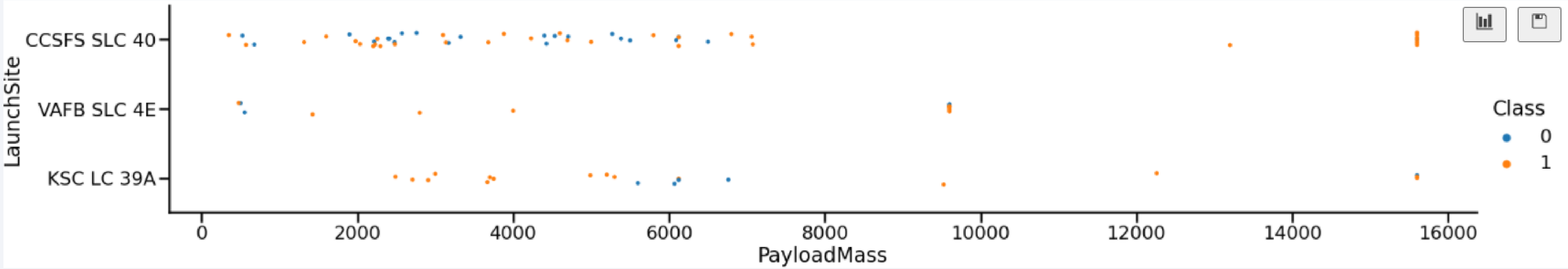
# Insights drawn from EDA

# Flight Number vs. Launch Site



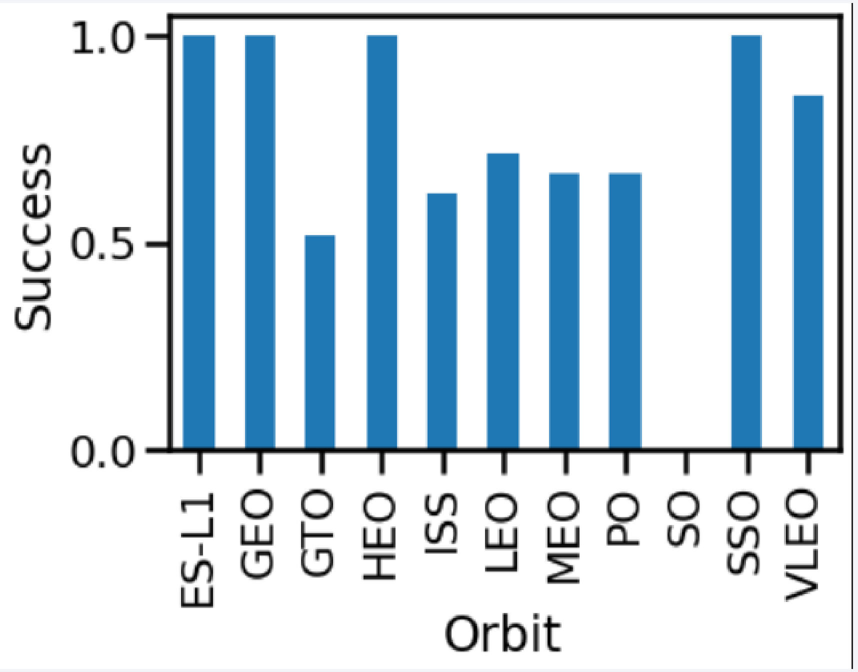

It is observed that the success rate increases, which is more clear in the launch site CCSFC SLC 40

# Payload vs. Launch Site



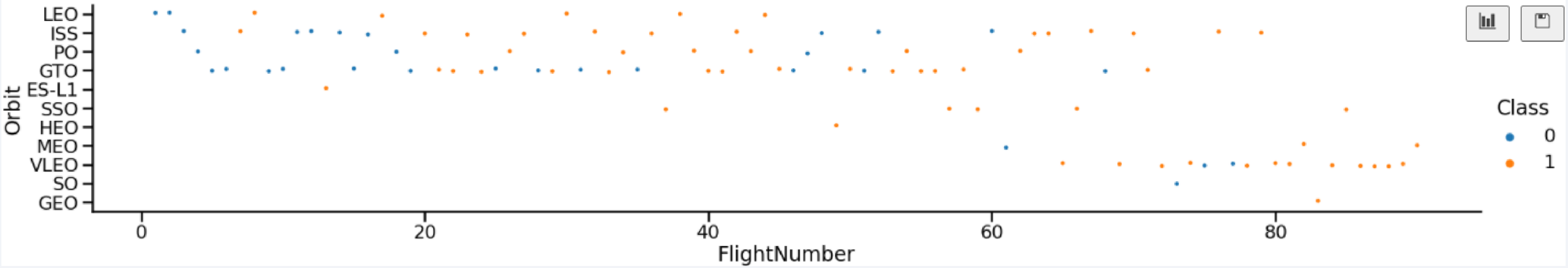

For some launch sites, the heavier payload, the larger the successful landing. However, if the payload is too large, there are some failed landings.

# Success Rate vs. Orbit Type



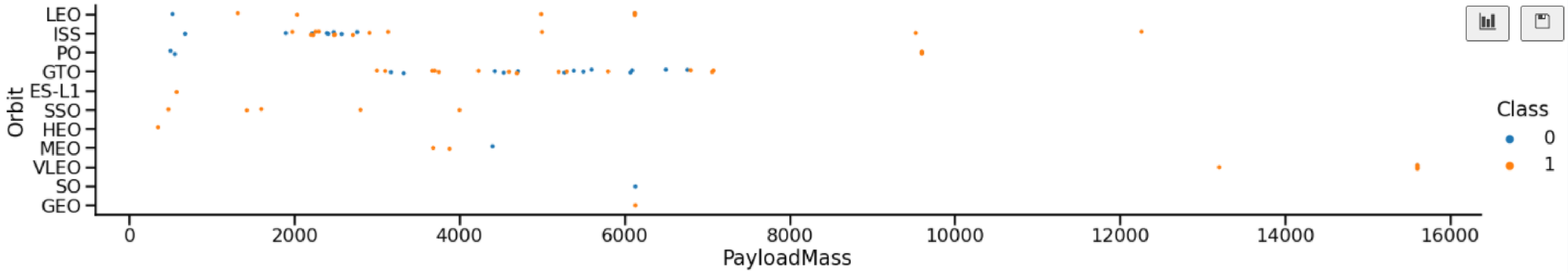

- This graph shows the success rates for different orbit types.
- Some of them (ES-L1, GEO, HEO, and SSO) have the highest values.

# Flight Number vs. Orbit Type



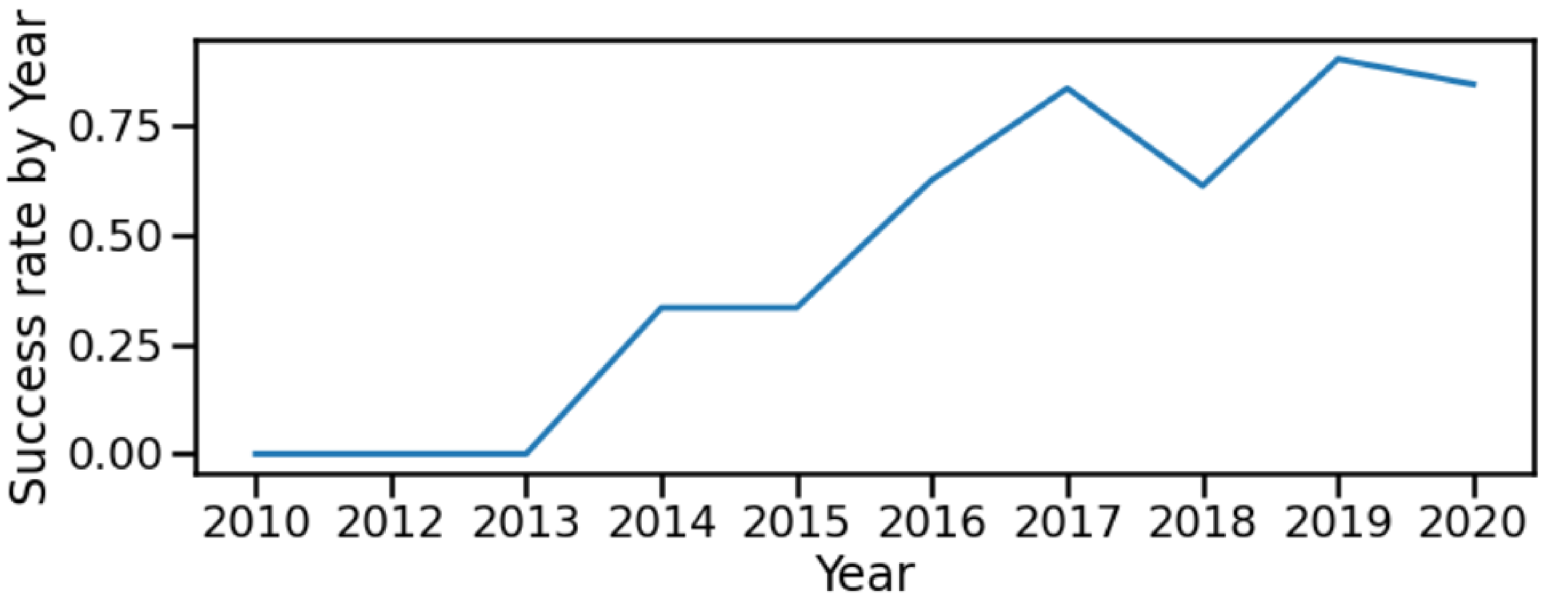

It is observed that the success appears related to the number of flights in the case of LEO orbit, and on the other hand, there seems to be no relationship between both variables in the GTO orbit.

# Payload vs. Orbit Type




The payload has an impact on the success at least for some orbits. Heavy payloads are more successful for Polar, Leo, and ISS orbits

# Launch Success Yearly Trend




This graph shows the evolution of the success, and from 2013 it increases over the years.

# All Launch Site Names

- SQL query

```
%sql select DISTINCT "Launch_Site" from SPACEXTBL
```

- Results

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

This query lists all distinct variables in the column “Lauch site” from the table SPACEXTBL.

# Launch Site Names Begin with 'CCA'

- SQL query

```
%sql select * from SPACEXTBL where "Launch_Site" like 'CCA%' limit 5
```

- Results

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 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) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

- This query finds firstly all row where column “Launch_site” starts with CCA and “limit 5” indicates to list only the first 5 lines.

# Total Payload Mass

- SQL query

```
%sql select sum ("PAYLOAD_MASS__KG_") from SPACEXTBL where "Customer" =='NASA (CRS)'
```

- Results

| sum ("PAYLOAD_MASS__KG_") |
|---|
| 45596 |

- This query returns the sum of payloads masses filtering where the costumer is NASA (CRS).

# Average Payload Mass by F9 v1.1

- SQL query

```
%sql select avg ("PAYLOAD_MASS__KG_") from SPACEXTBL where "Booster_Version" =='F9 v1.1'
```

- Results

| avg ("PAYLOAD_MASS__KG_") |
|---|
| 2928.4 |

- This query returns the average of payloads masses filtering where  the booster version is F9 v1.1

# First Successful Ground Landing Date

- SQL query

```
%sql select min(Date) from SPACEXTBL where "Landing _Outcome" == 'Success (ground pad)'
```

- Results

| min(Date) |
|---|
| 01-05-2017 |

- This query returns the oldest successful landing considering only ground landing.

# Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL query

```
%sql select "Booster_Version" from SPACEXTBL where ("Landing _Outcome" == 'Success (drone ship)') AND ("PAYLOAD_MASS__KG_">4000 AND "PAYLOAD_MASS__KG_"<6000)
```

- Results

| sum ("PAYLOAD_MASS__KG_") |
|---|
| 45596 |

- This query returns the Bosster Veersion filtering by two columns. Only returns Booster version where Landing Outcomes is Success drone ship, and where Payload masses in between 4000 and 6000 kg.

# Total Number of Successful and Failure Mission Outcomes

- SQL query

```
%sql SELECT(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') as SUCCESS,\
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') as Failure
```

- Results

| SUCCESS | Failure |
|---|---|
| 100 | 1 |

- In this case subqueries were used. The first one counts the success, the second one the failure, using the filter "where".

# Boosters Carried Maximum Payload

- SQL query

```
%sql select "Booster_Version","PAYLOAD_MASS__KG_" from SPACEXTBL where "PAYLOAD_MASS__KG_" ==(select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
```

- Results

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

In this case, two subqueries were used again. The main is for listing the Boster_version and Payload masses, and the second one is for filtering the maximum payload.

# 2015 Launch Records

- SQL query

```
%sql select substr(Date, 4, 2) as Month ,"Landing _Outcome","Booster_Version","Launch_Site" from SPACEXTBL where substr(Date,7,4)='2015' and "Landing _Outcome" = 'Failure (drone sh
```

- Results

| Month | Landing _Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

- This query returns the month, landing outcome, booster version, and launch site where the landing was unsuccessful and took place in 2015.

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

- SQL query

```
%sql select "Landing _Outcome", count("Landing _Outcome") from SPACEXTBL where ("Date" Between '04-06-2010' and '20-03-2017') and "Landing _Outcome" like '%Success%'\
group by "Landing _Outcome" \
order by count("Landing _Outcome") desc
```

- Results

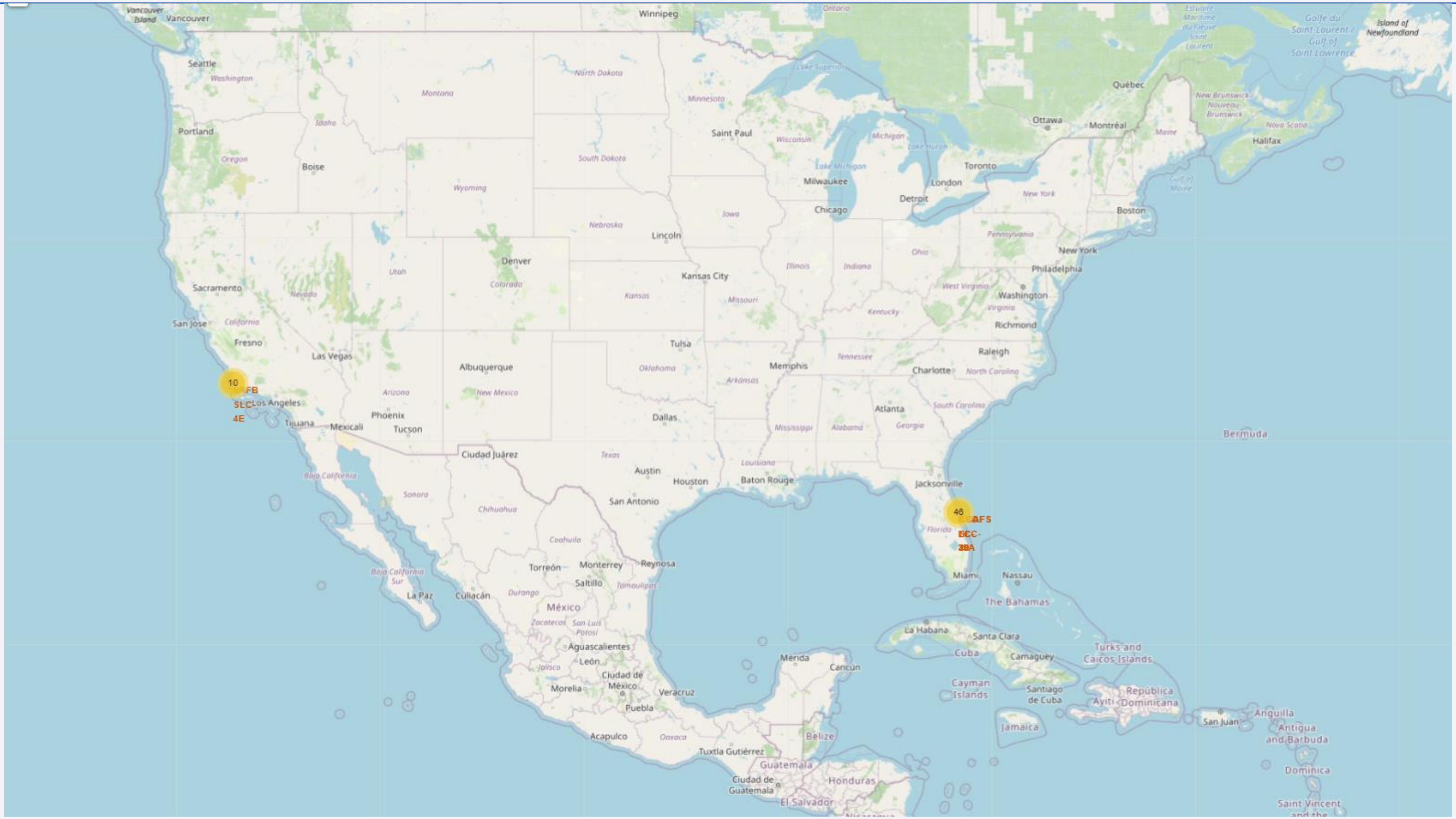
| Landing _Outcome | count("Landing _Outcome") |
|---|---|
| Success | 20 |
| Success (drone ship) | 8 |
| Success (ground pad) | 6 |

- This query returns the Landing outcomes and their counts for dates between 04-06-2010 and 20-03-2017, also it groups them by the landing outcome, and finally shows them in decreasing order.

Section 3

# Launch Sites Proximities Analysis

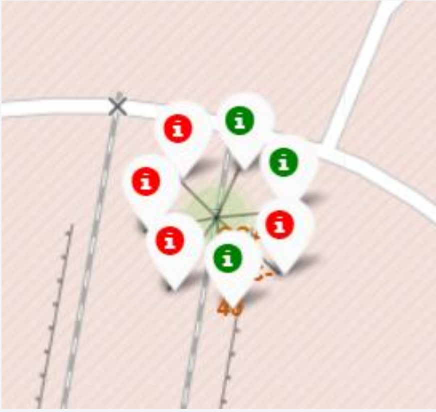
# Folium Map- Stations



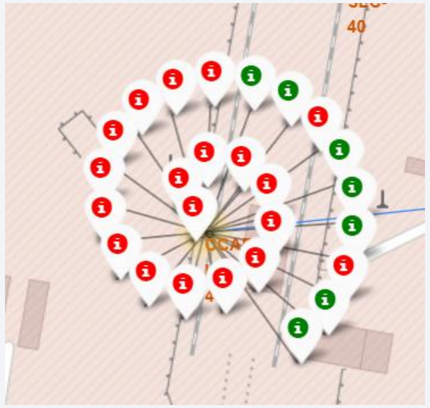

It can be observed the Space X launch sites on both coast.

# Folium Map- Succeed and Failure in color markers

VAFB SLC-4E



CCAFS SLC-40

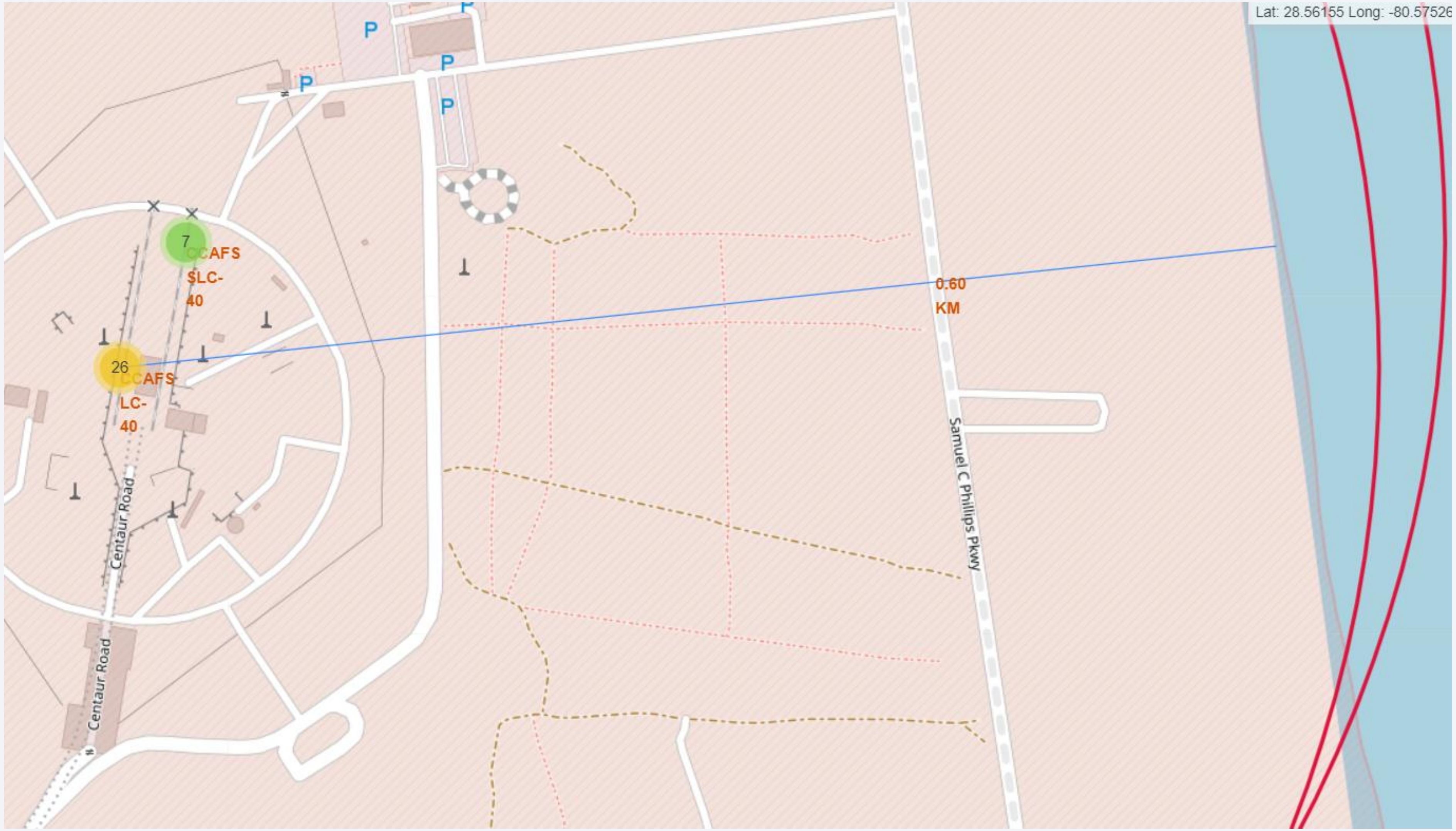


CCAFSLC-40

KSC LC-39A

Examples for each launch site were the succeed is indicated in green and failure in red

# Folium Map- Distance

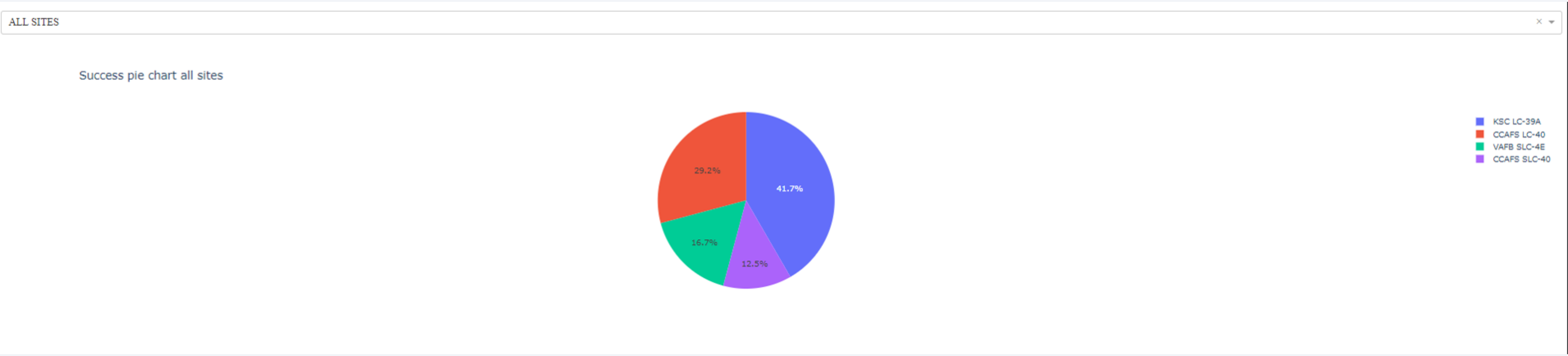



Examples of the distance between CCAFS LC-40 launch site and the coast (0.6 km).

Section 4

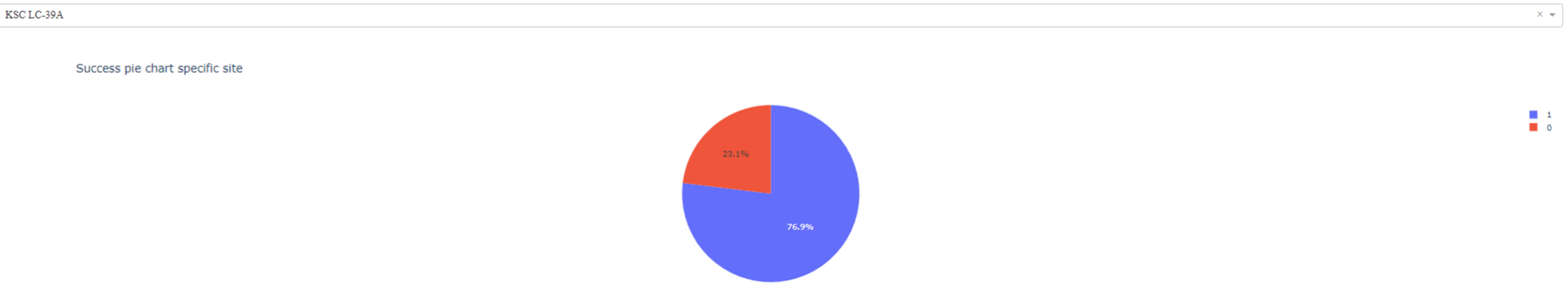
# Build a Dashboard with Plotly Dash

# Dashboard Total success by site



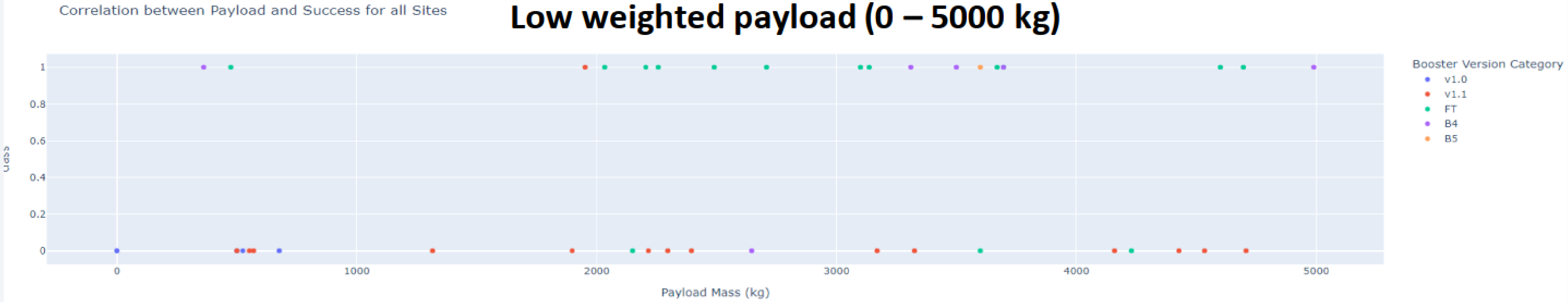

It can be seen that the KSC LC 39A has the best performance.

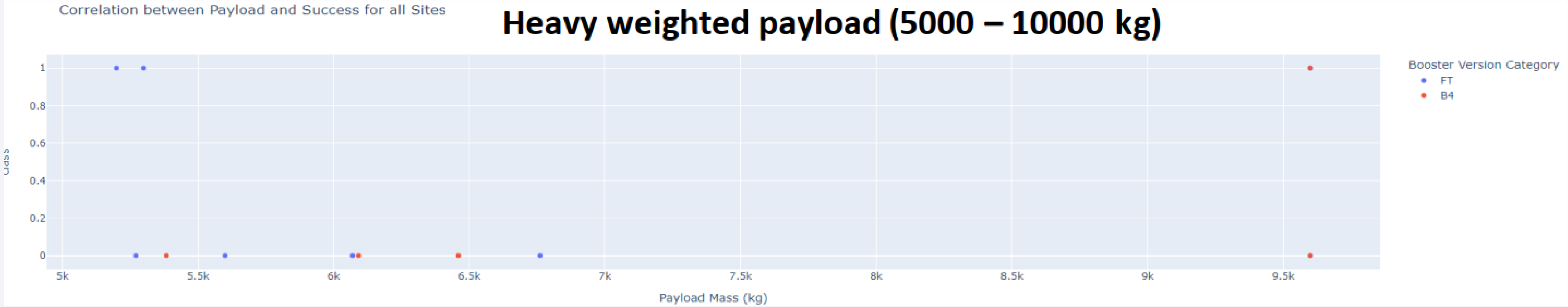
# Dashboard Total Success for site KSC LC 39A




On KSC LC 39A, the 76.9 % of launches were successful and 23.1 % unsuccessful.

# Dashboard Payload vs Launch outcome for all sites



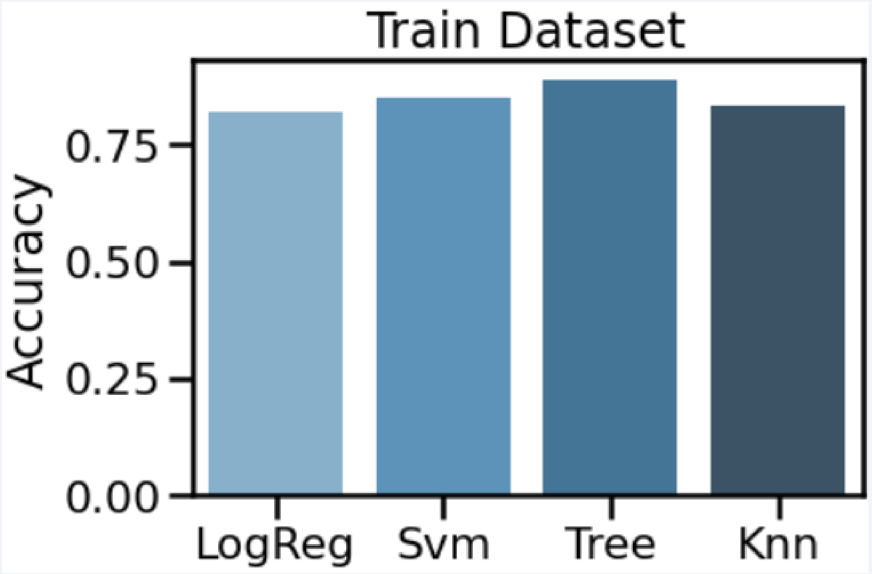



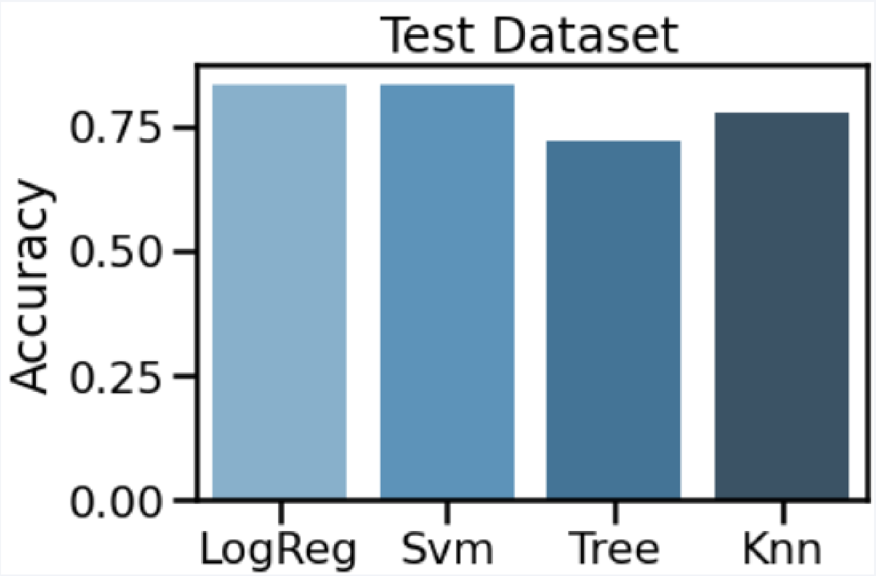

The success rate is better for low weighted payload.

Section 5

# Predictive Analysis (Classification)

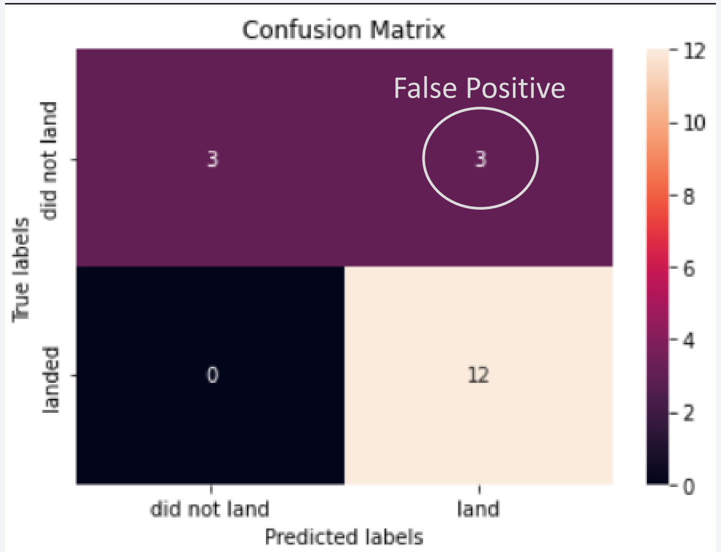
# Classification Accuracy




- Based on this data the models have a similar accuracy (train Dataset)
- LogReg and Svm have slightly higher accuracy than Tree and Knn

# Confusion Matrix




- This is the confusion Matrix for the SVM model.
- The problem with this model is the false positives.

# Conclusions

- From this analysis it can be concluded that the success of landing depends on many factors such as the launch site, the orbit, and the number of previous launches.
- ES-L1, GEO, HEO, and SSO are the orbits with the highest values of success.
- In general terms, the success rate is better for a low-weighted payload.
- An increase in the success rate since 2013 was observed.
- 4 models were used to predict future missions (Logistic Regression, SVM, KNN, and Tree Classifier), and LR and SVM have a slight best accuracy (on test datasets)

# Thank you!