Winning Space Race with Data Science

Soohyun Ahn
August 16th, 2022

Outline

- **Executive Summary**
- **Introduction**
- **Methodology**
- **Results**
- **Conclusion**

Executive Summary

Summary of methodologies

- Data collection via APIs and web scraping
- EDA with SQL
- EDA with data visualization
- Predictive analysis

Summary of all results

- EDA Analysis
- Interactive maps and dashboard
- Predictive results

Introduction

Project background and context

- The aim of this project is to predict whether the Falcon 9 first stage will land successfully.
- According to SpaceX, their Falcon 9 rocket launches have been financially efficient, only cost around 62 million dollars. This is remarkable given that other providers spend nearly 165 million dollars each.
- The cost differential is primarily explained by the fact that SpaceX can reuse the first stage under the right condition.
- By predicting the landing of the stage, the cost of a launch can be assessed accordingly. Accurate prediction is crucial in order for the competitors to make a compelling bid against SpaceX.

Problems you want to find answers

- What are the main characteristics of a successful or failed landing?
- What are the effects of each relationship of the rocket variables on the success or failure of a landing?
- What are the conditions which will allow SpaceX to achieve the best landing success rate?

Section 1

Methodology

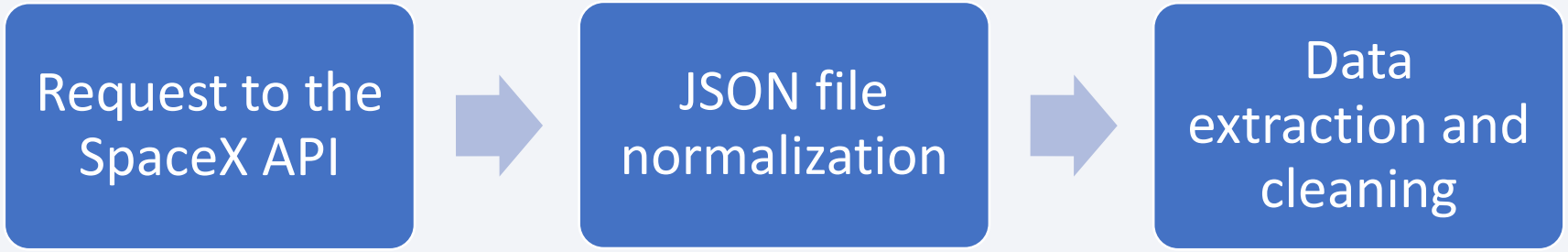
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web scraping from Wikipedia
- Perform data wrangling
  - Used `value_counts()` and `append` to process data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Compared each classification models using train/test sets

Data Collection

Data collection via API



Web scraping from Wikipedia

Data Collection – SpaceX API

1. Get response from API

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

2. Get JSON results and normalize it

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

3. Extract relevant data

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

4. Create dictionary

```
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}
```

5. Create dataframe

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

6. Filter dataframe

```
data_falcon9 = df[df['BoosterVersion']!= 'Falcon 1']
```

7. Data wrangling

```
data_falcon9.isnull().sum()
# Calculate the mean value of PayloadMass column
PayloadMassMean = data_falcon9['PayloadMass'].mean()
PayloadMassMean
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, PayloadMassMean, inplace=True)
data_falcon9.isnull().sum()
```

8. Export to CSV

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

For the complete code, click here.

Data Collection - Scraping

1. Request HTML page

```
data = requests.get(static_url).text
```

2. Create BeautifulSoup object

```
soup = BeautifulSoup(data, 'html.parser')
```

3. Extract tables

```
html_tables = soup.find_all('table')
```

4. Get column names

```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

5. Create dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
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. Fill table

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

7. Create dataframe

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

8. Extract to CSV

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

For the complete code, click here.

Data Wrangling

Data wrangling primarily focuses on converting various categories (e.g., False Ocean, True ASDS) into intuitive binary ones, “1” denoting a success, whereas “0” a failure.

1. Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()
```

2. Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculate the number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create a landing outcome label from Outcome column

```
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

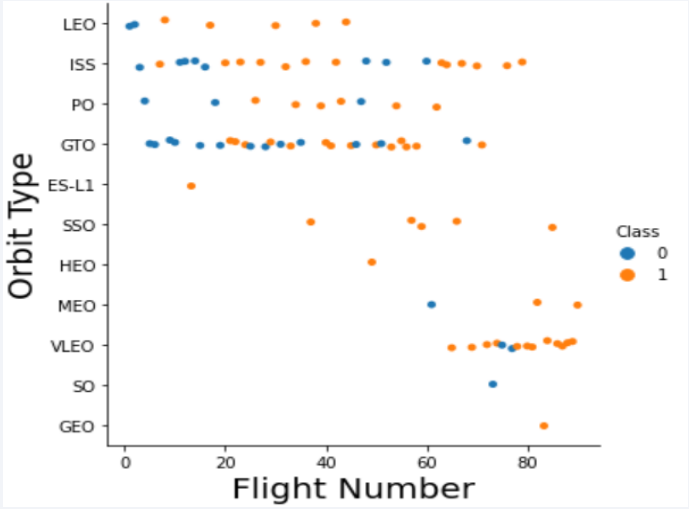
5. Export to CSV

```
df.to_csv("dataset_part\_2.csv", index=False)
```

For the complete code, click here.

EDA with Data Visualization

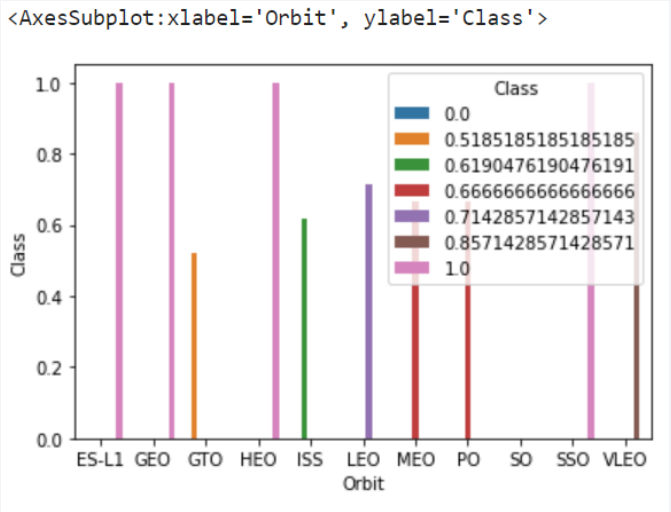
Type 1: Scatter point chart



Scatter charts were created to show correlations between variables.

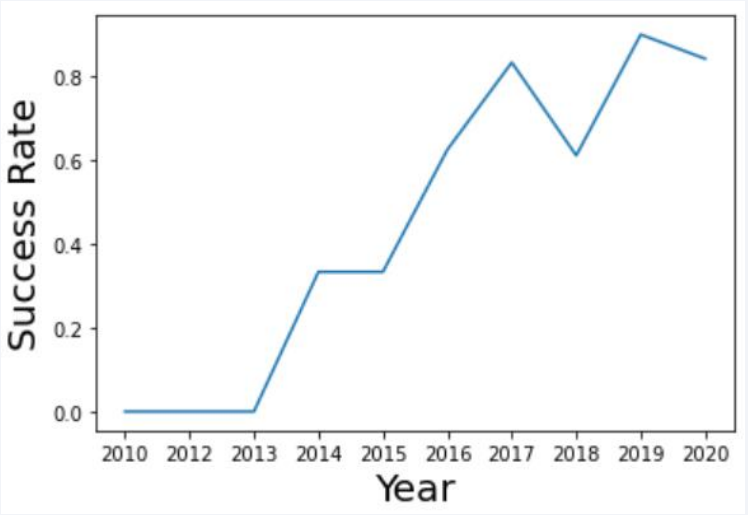
To see the complete code and charts, click here.

Type 2: Bar chart



Bar charts were created to quickly capture meaningful patterns.

Type 3: Line chart



Line charts were created to represent historical trend between variables.

EDA with SQL

SQL queries used to understand the dataset:

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

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL;
```

- Display 5 records where launch sites begin with the string 'CCA'

```
%sql select LAUNCH_SITE from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;
```

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

```
%sql select sum(PAYLOAD_MASS__KG_) as 'total payload mass' from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
```

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

```
%sql select avg(PAYLOAD_MASS__KG_) as 'average payload mass' from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1';
```

- List the date when the first successful landing outcome in ground pad was achieved

```
%sql select min(Date) as 'first successful landing outcome in ground pad' from SPACEXTBL where "Landing _Outcome" = 'Success (ground pad)';
```

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

```
%sql select "Booster_Version" from SPACEXTBL where "Landing _Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

- List the total number of successful and failure mission outcomes

```
%sql select "Mission_Outcome", count("Mission_Outcome") as outcomes from SPACEXTBL group by "Mission_Outcome";
```

For a complete list of performed SQL queries, click here.

Build an Interactive Map with Folium

Folium map object was created with an initial center location as NASA Johnson Space Center at Houston, TX.

- Used `folium.Circle` and `folium.map.Marker` to add a blue circle at NASA Johnson Space Center's coordinate with label showing its name
- Used `folium.Circle` and `folium.map.Marker` to add a blue circle for each launch site name
- Created and added `marker_cluster` to the map to easily identify launch sites with high success rates
- Used `folium.map.Marker` and `folium.PolyLine` to display the distance between launch sites and key locations, including railway and coastal lines.

For the complete code and interactive folium maps, click here.

Build a Dashboard with Plotly Dash

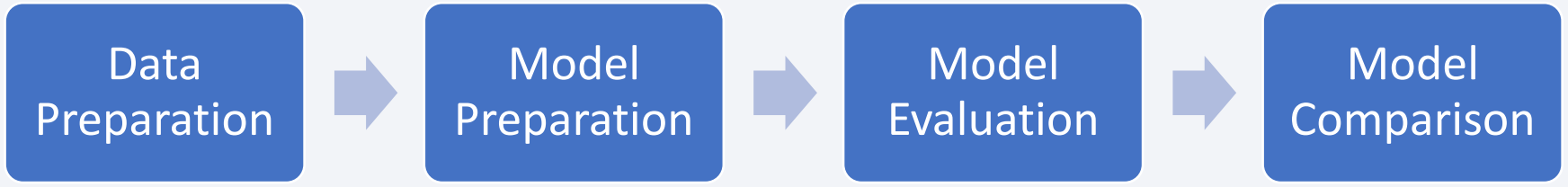
Created a Dashboard with dropdown list, pie chart, range slider and scatter plot

- Dropdown allows a user to choose a specific launch site or all launch sites
- Pie chart allows a used to compare the total success vs. the total failure for the launch site chosen with the dropdown component.
- Range slider allows a user to select a payload mass in a fixed range.
- Scatter plot allows a user to see the relationship between two variables, such as particular Success vs Payload Mass.

To see the complete code, click here.

Predictive Analysis (Classification)

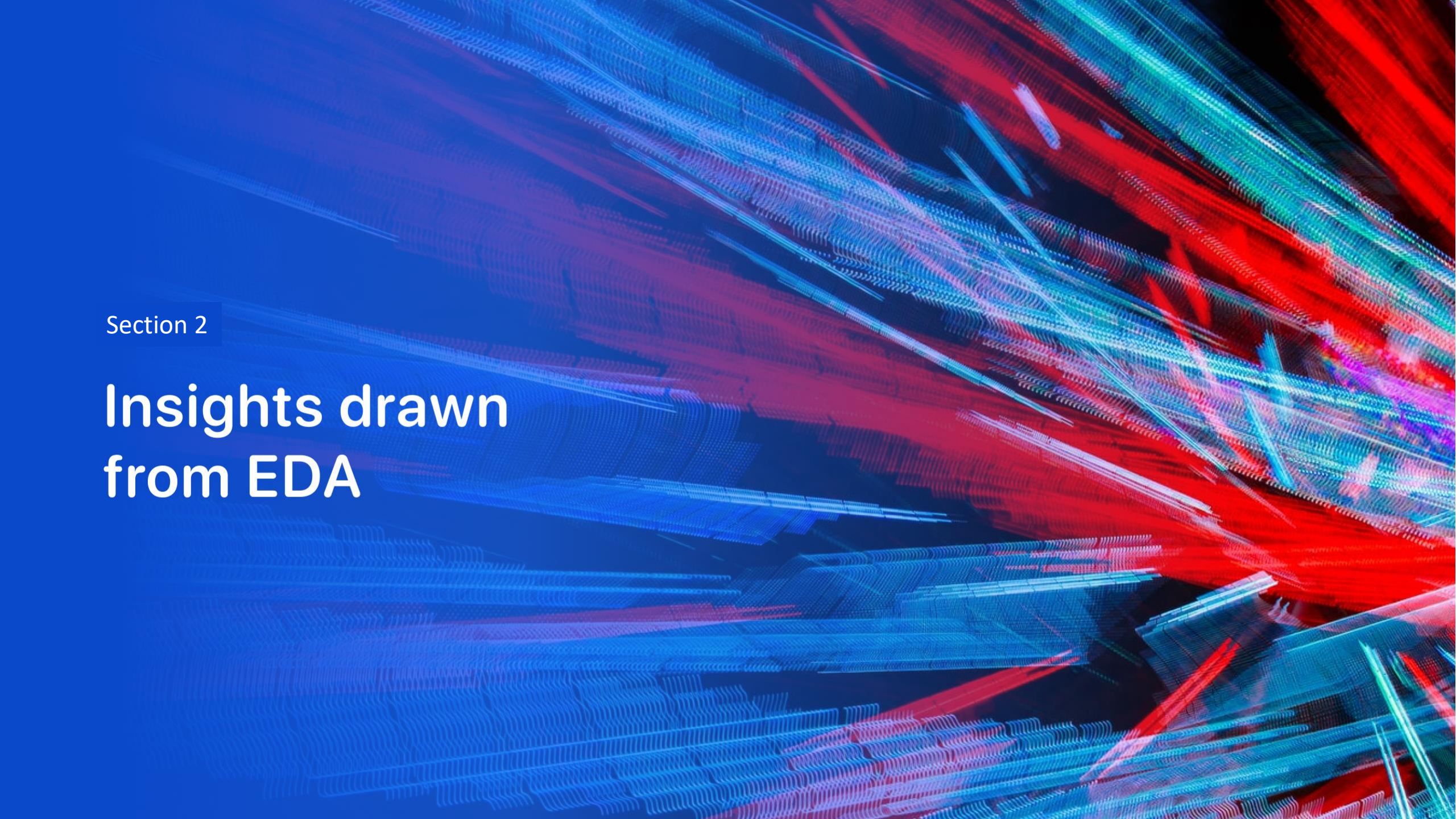
To find the best performing method, the following processes were executed.



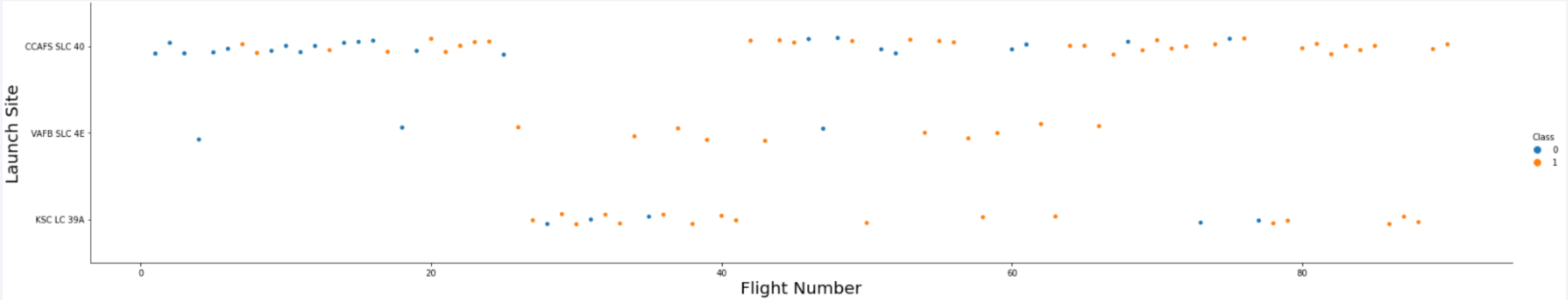
For the complete analysis, click here.

Results

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


Section 2
Insights drawn from EDA

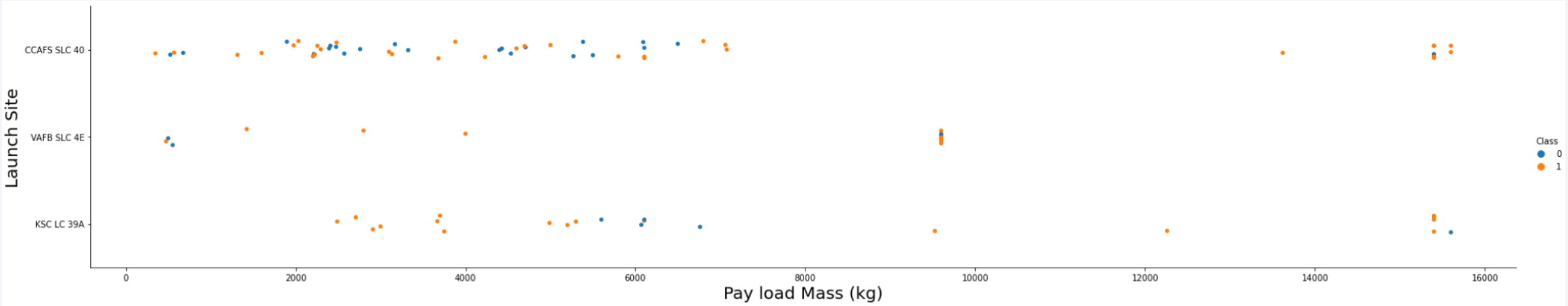
Flight Number vs. Launch Site



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

We observe that as flight number increases, the success rate for each site is also increasing.

Payload vs. Launch Site

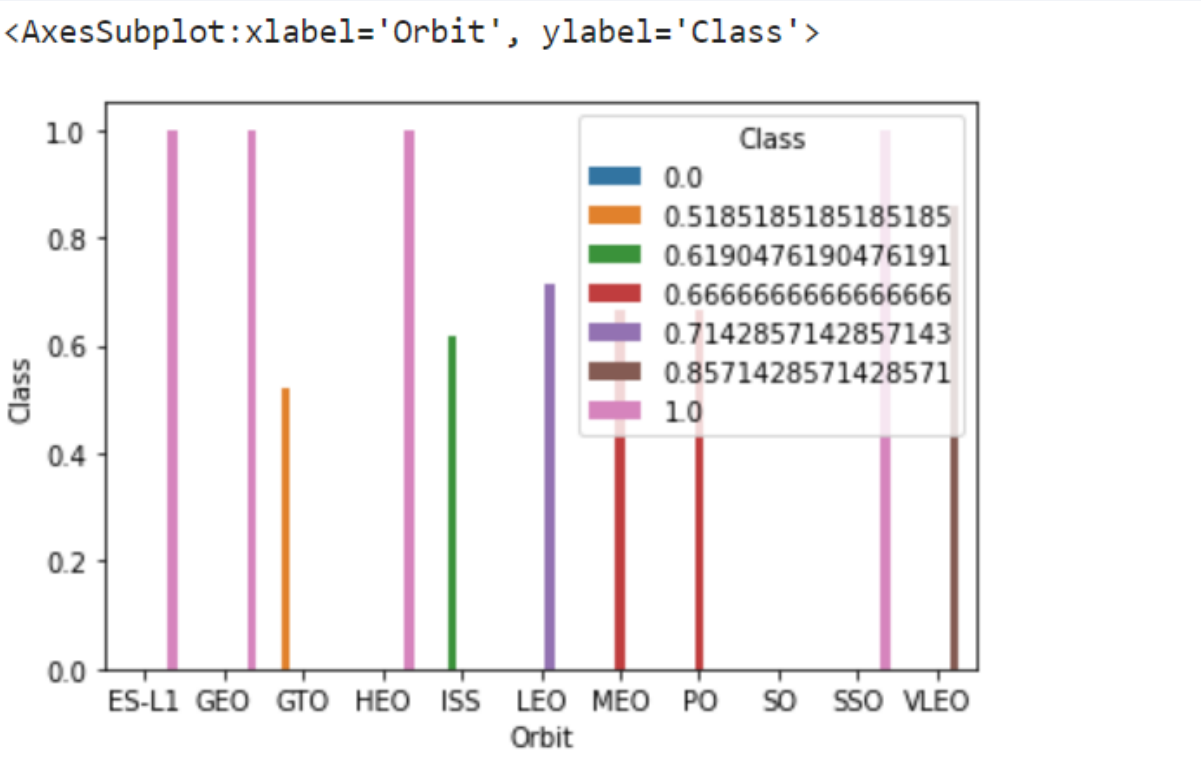


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Depending on the launch site, a heavier payload can be a condition for success.

However, given data suggests that for some launch sites(e.g., KSC LC 39A) a too heavy payload can result in failure.

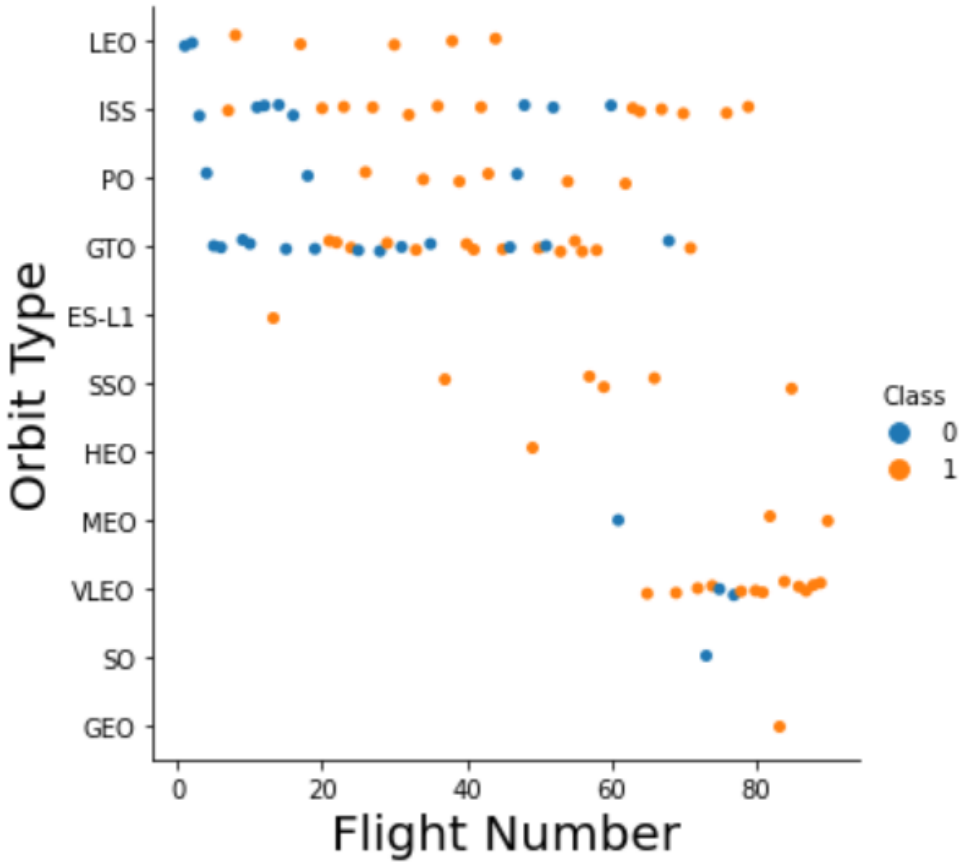
Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

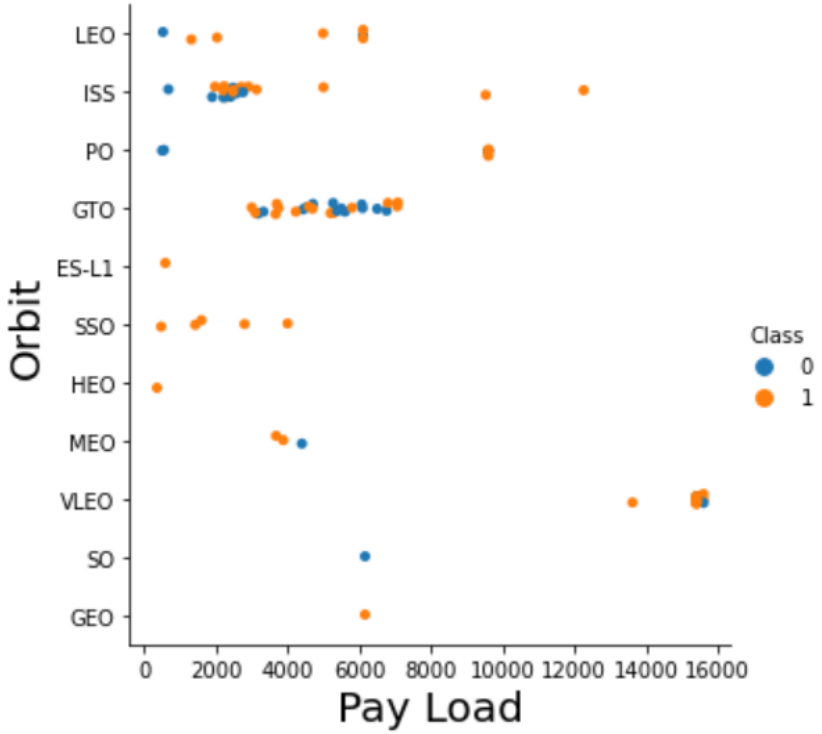
We note that ES-L1, GEO, HEO, SSO have the highest success rate.

Flight Number vs. Orbit Type



You should see 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.

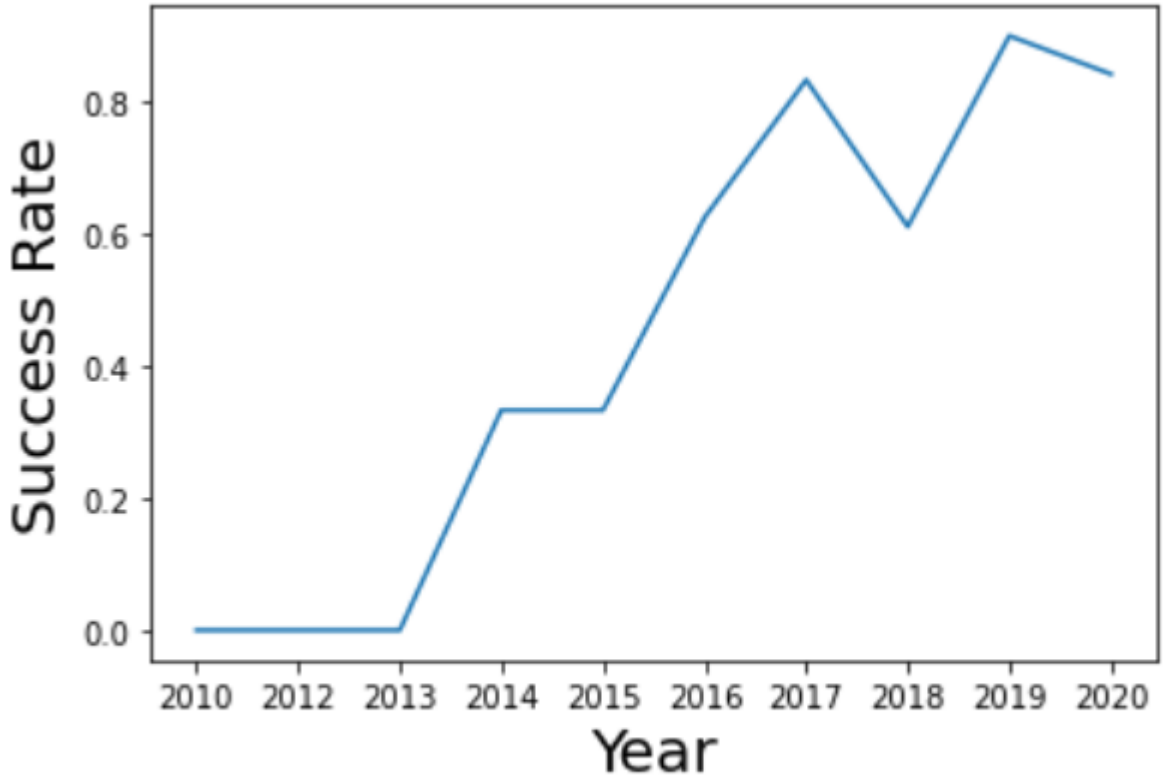
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

SQL query of finding the names of the unique launch sites:

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL;
```

SQL query result:

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

SQL query explained: The “select distinct” statement removes duplicates and returns unique names.

Launch Site Names Begin with 'CCA'

SQL query of finding 5 records where launch sites begin with the string 'CCA':

```
%sql select LAUNCH_SITE from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;
```

SQL query result:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

SQL query explained: The “like” operator looks for a specific pattern, and the “%” sign represents multiple characters. The “limit 5” clause returns only 5 results.

Total Payload Mass

SQL query of calculating the total payload carried by boosters from NASA:

```
%sql select sum(PAYLOAD_MASS__KG_) as 'total payload mass' from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
```

SQL query result:

total payload mass
45596

SQL query explained: The query returns the sum of payload mass where customer is NASA (CRS).

Average Payload Mass by F9 v1.1

SQL query of calculating the average payload mass carried by booster version F9 v1.1:

```
%sql select avg(PAYLOAD_MASS__KG_) as 'average payload mass' from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1';
```

SQL query result:

average payload mass
2928.4

SQL query explained: The AVG () function returns the average of payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SQL query of finding 5 records where launch sites begin with the string 'CCA':

```
%sql select min(Date) as 'first successful landing outcome in ground pad' from SPACEXTBL where "Landing _Outcome" = 'Success (ground pad)';
```

SQL query result:

first successful landing outcome in ground pad
01-05-2017

SQL query explained: The “min(date)” function returns the oldest time record.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL query of finding names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:

```
%sql select "Booster_Version" from SPACEXTBL where "Landing _Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

SQL query result:

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

SQL query explained: The “where” and “between…and…” clauses filters the target outcomes.

Total Number of Successful and Failure Mission Outcomes

SQL query of finding the total number of successful and failure mission outcomes:

```
%sql select "Mission_Outcome", count("Mission_Outcome") as outcomes from SPACEXTBL group by "Mission_Outcome";
```

SQL query result:

Mission_Outcome	outcomes
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

SQL query explained: The “group by” statement sort the dataset according to “Mission_Outcome” and then count the number in each category.

Boosters Carried Maximum Payload

SQL query of finding the names of the booster_versions which have carried the maximum payload mass:

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

SQL query result:

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

SQL query explained: The subquery filters data by returning only the heaviest payload mass with “max” function. The main query uses subquery results and returns booster version with the heaviest payload mass.

2015 Launch Records

SQL query of displaying the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015:

```
%sql select substr("Date",4,2) as month, "Booster_Version", "Launch_Site" from SPACEXTBL \
WHERE "Landing _Outcome"='Failure (drone ship)'and substr("Date",7,4) = 2015;
```

SQL query result:

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

SQL query explained: The “substr” function returns date in order so that extracting month and year is possible.

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

SQL query of ranking 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", "Date", count(*) as count_launches \
from SPACEXTBL where "Landing _Outcome" like 'Success%' and "Date" between '04-06-2010' AND '20-03-2017'\
group by "Landing _Outcome" order by count_launches desc;
```

SQL query result:

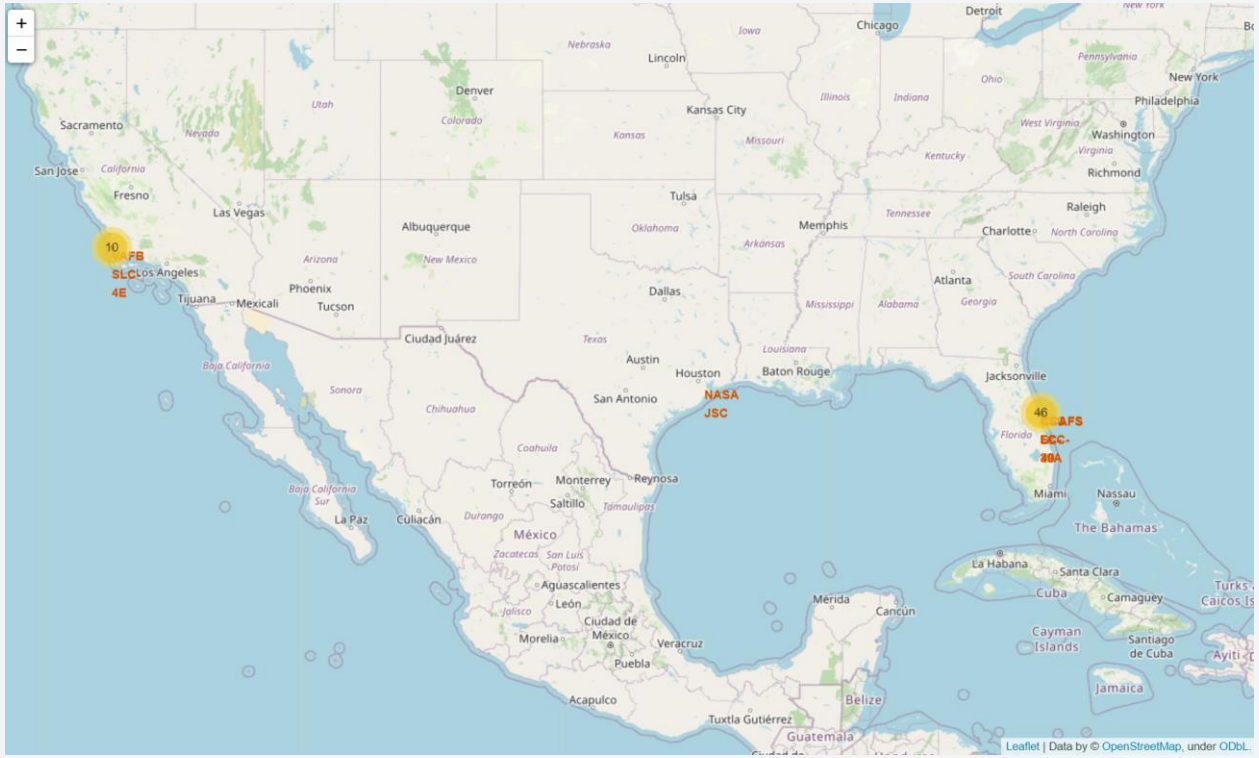
Landing _Outcome	Date	count_launches
Success	07-08-2018	20
Success (drone ship)	08-04-2016	8
Success (ground pad)	18-07-2016	6

SQL query explained: The “Success%” clause returns landing outcomes that are successful, and the “date…between…and…” clause limits results that belongs to a particular period.

Section 3

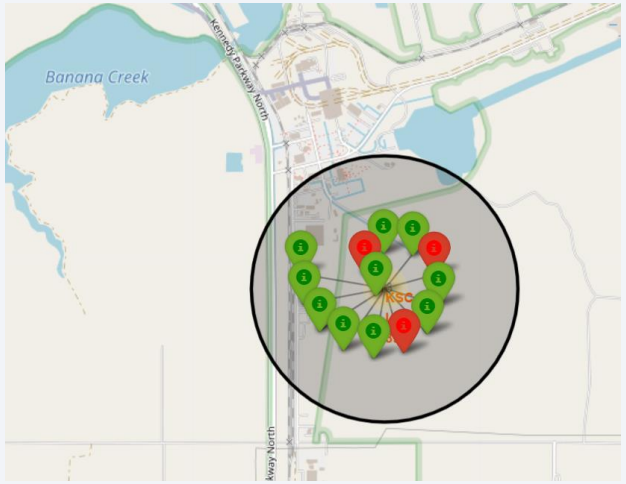
Launch Sites Proximities Analysis

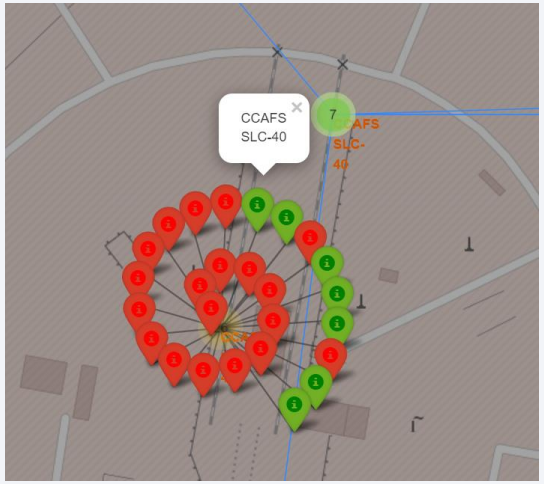
<Folium Map Ground Stations>



The generated map shows SpaceX launch sites located on coastal lines of the U.S.

<Folium Map Color-labeled Markers>

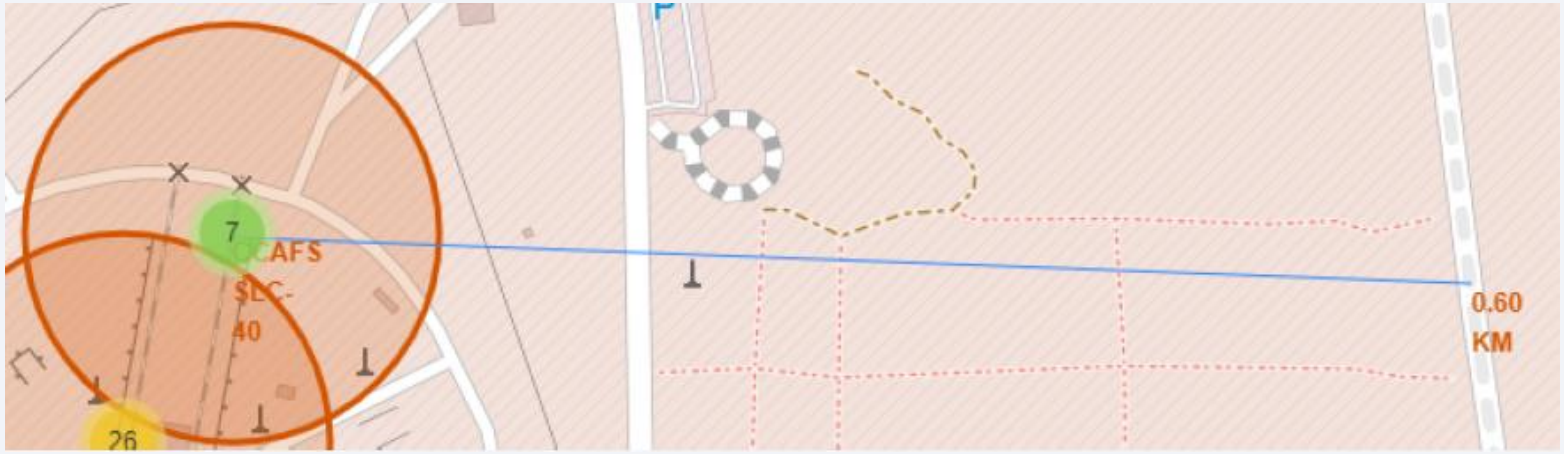




Green markers represent successful launch outcomes whereas Red markers denote failures.

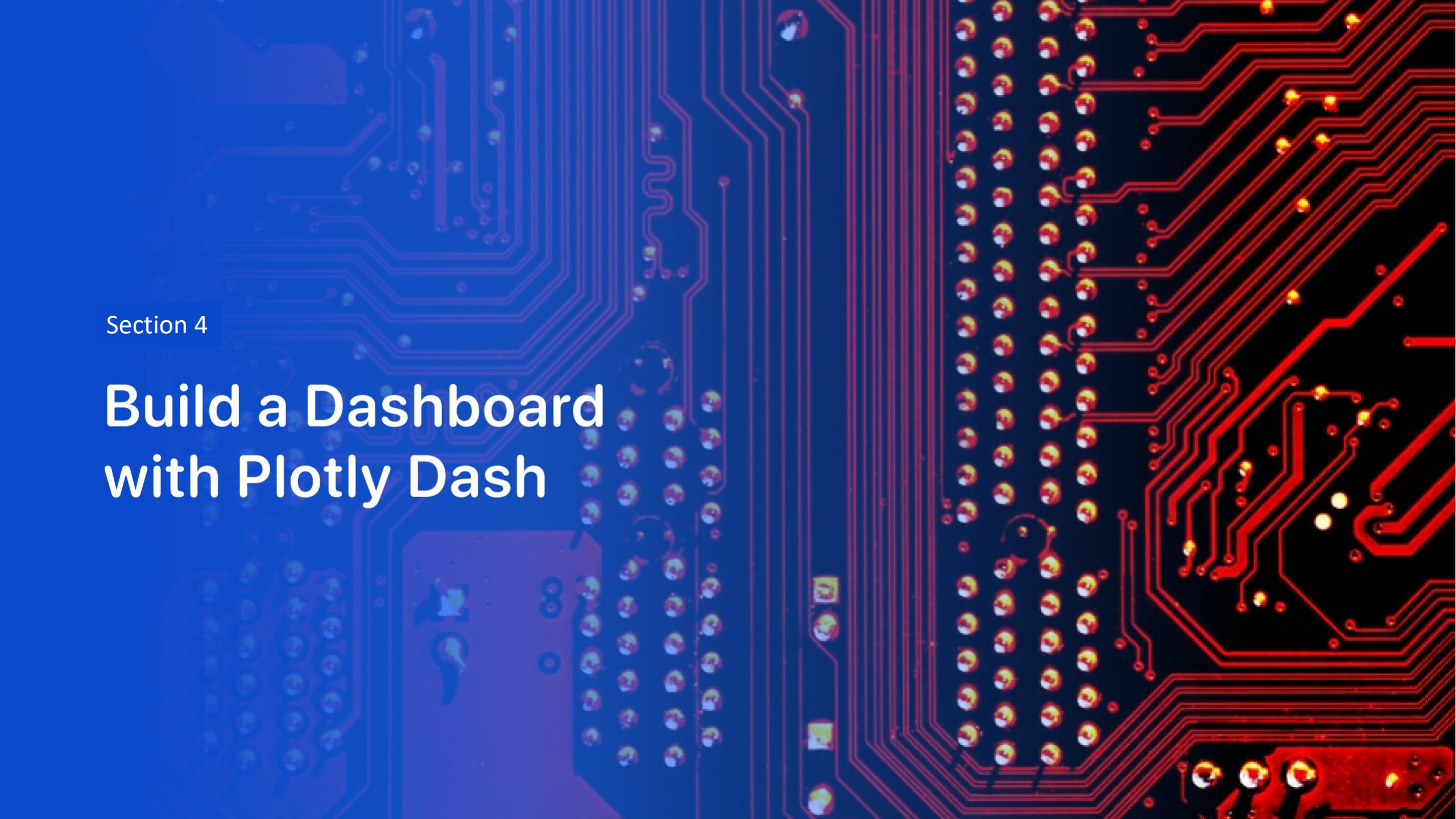
By clicking and checking the number of success/failures, we can know which site (e.g., KSC LC 39A) has a highest launch success rate.

<Folium Map: Distances and Proximities>

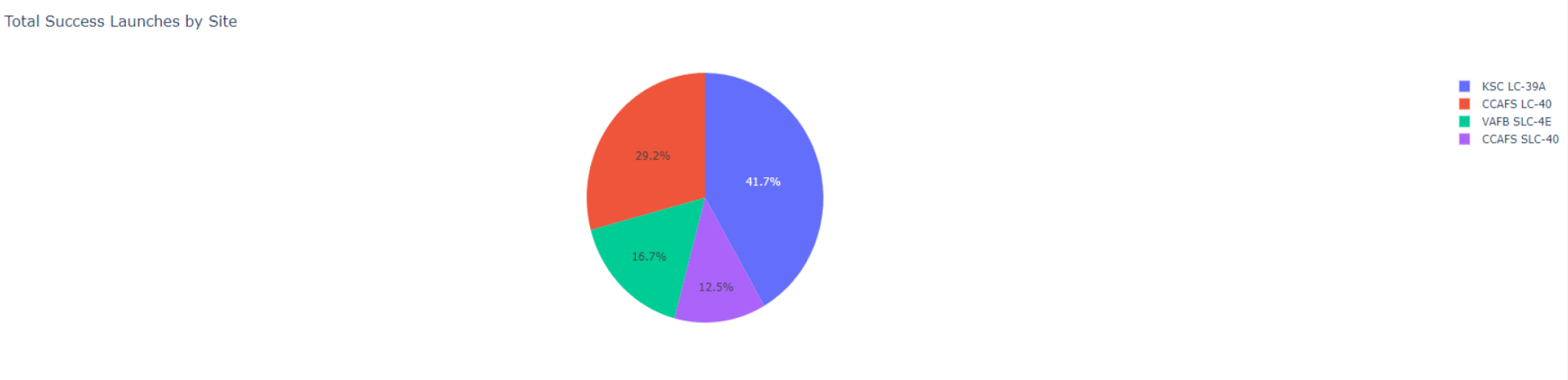


As the above map shows, CCAFS SLC-40 is 0.60 km away from the highway (Samuel C Phillips Pkway).

Folium maps allow measuring distances and determining proximities.

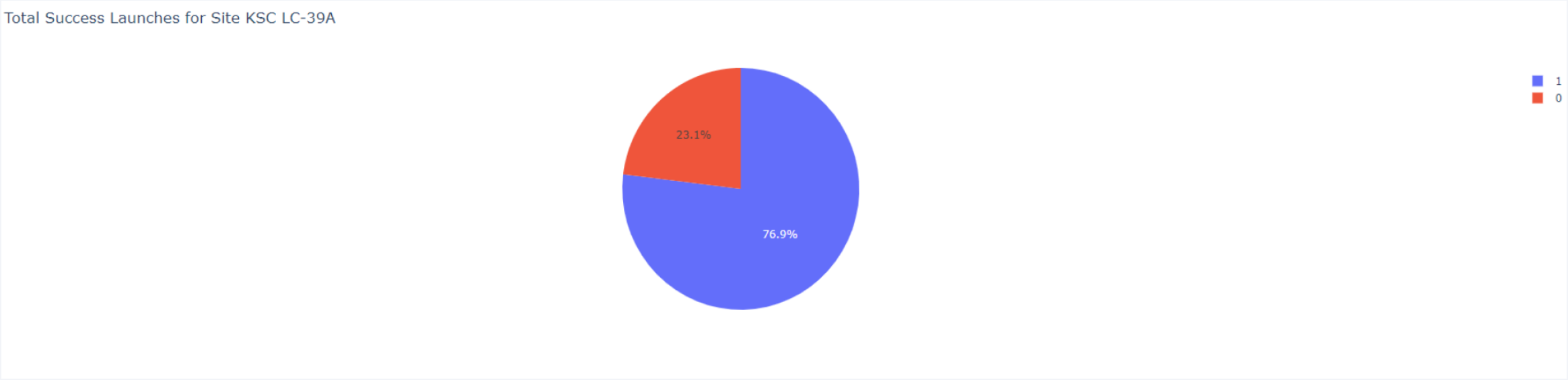

Section 4
Build a Dashboard with Plotly Dash

<Dashboard: Total Success Launches by Sites>



The pie chart indicates that KSC LC-39A has the highest success rate for launches.

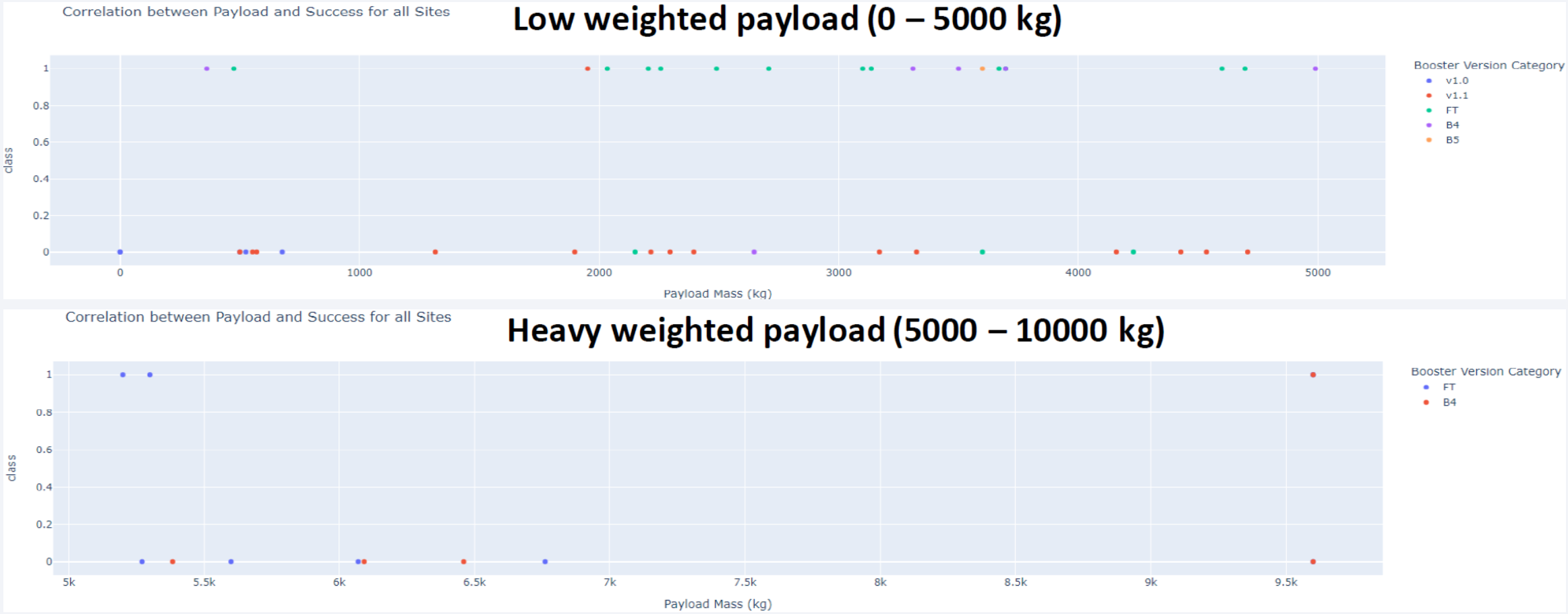
<Dashboard: Total Success Launches for KSC LC-39A>



KSC LC-39A has a 76.9% of success rate whereas failure rate amounts to 23.1%.

(“1” indicates success/ “0” for failure)

<Dashboard: Payload vs Success>

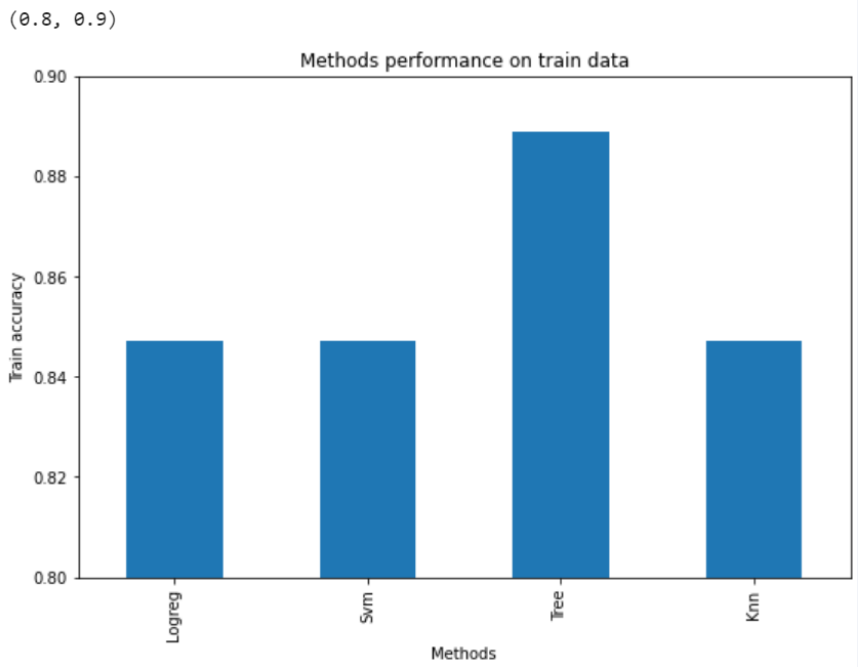


Low weighted payloads tend to be more successful than heavy weighted ones.

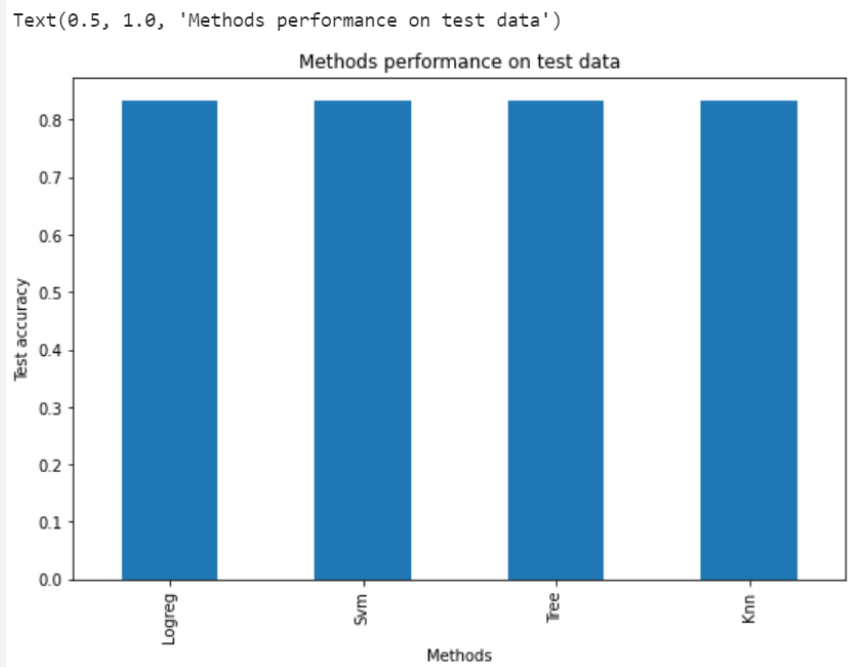
Section 5

Predictive Analysis (Classification)

Classification Accuracy



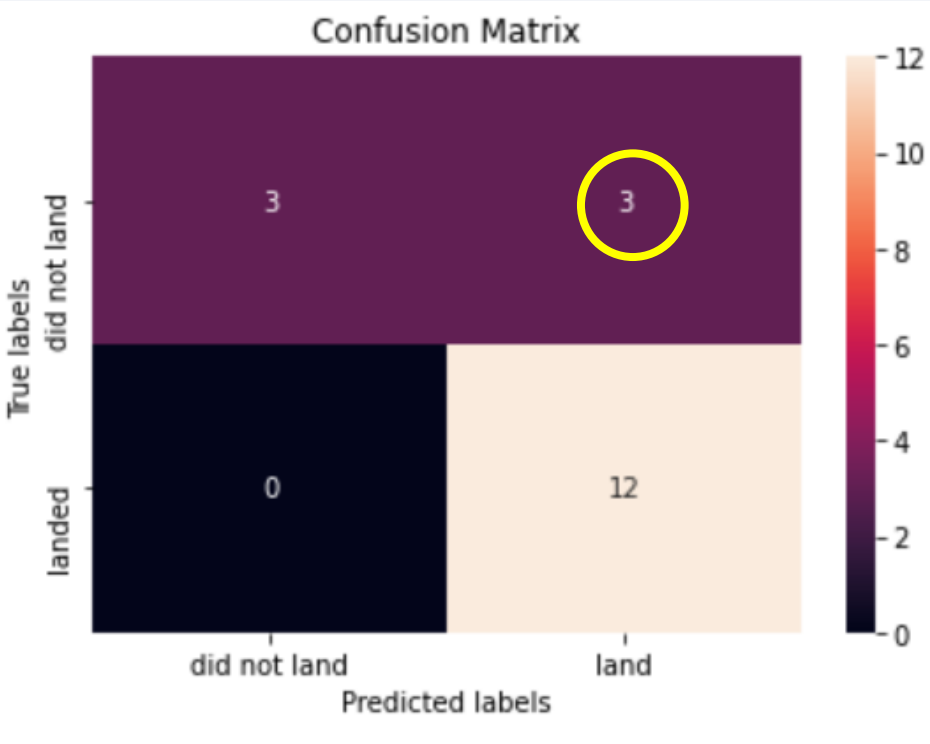
	Accuracy Train	Accuracy Test
Logreg	0.847222	0.833333
Svm	0.847222	0.833333
Tree	0.888889	0.833333
Knn	0.847222	0.833333



Accuracy for each method in training and testing data is visualized in bar charts.

Performance on test data is nearly the same for each method. However, since the tree method shows relatively higher accuracy in training set, we recommend using the tree method.

Confusion Matrix



The confusion matrix for each method is the same.

According to the matrix, false positives (shown in a yellow circle) are problematic.

Conclusions

- In this project, our aim is to distinguish relevant variables to predict the success of a mission.
- Several factors, including launch sites and the orbit type, are relevant to the success or failure of the launch.
- The orbits with the highest success rates are GEO, HEO, SSO, and ES L1.
- For some orbit types, the payload mass plays a crucial role in determining the success. However, generally speaking, low weighted payloads tend to perform better than the heavy weighted ones.
- Although we can stipulate the best launch site, namely KSC LC 39A, based on the given data, it is still vague why that site performs better than other sites. Further analysis is required to answer this question.
- The best classification method is the Decision Tree Algorithm given its accuracy result on training set (the results on test set are identical across all methods).

Thank you!