IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amogh Bajpai
24.09.2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

Summary of Methodologies:

1. Data collection
2. Data wrangling
3. EDA using visualisation
4. EDA with SQL
5. Building an interactive map with Folium
6. Building a dashboard with Plotly Dash
7. Predictive analysis with classification

Summary of Results:

1. EDA results
2. Visualisation screenshots
3. Classification prediction

Introduction

Project background and context

Since its beginnings in the Cold War, the Space race has now become commercial, with many corporations getting involved in the tussle to make space travel accessible to the general public. The most successful corporation among them is, arguably, SpaceX.

Here we try to predict whether the SpaceX Falcon9 rocket will land successfully among its historical test sites. This particular vessel has been chosen because it is relatively far less expensive than its market competitors. We can, as a corollary, also determine the cost of a launch on the basis of our landing prediction.

Problems for which you need to find answers

Correlation between rocket variables and successful landing rate

Conditions to get the best results and ensure the best successful landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and Webscraping from Wikipedia: Falcon9 and Falcon Heavy Launches
- Perform data wrangling
  - Convert outcomes into Training Labels with the booster successfully/unsuccessful landed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

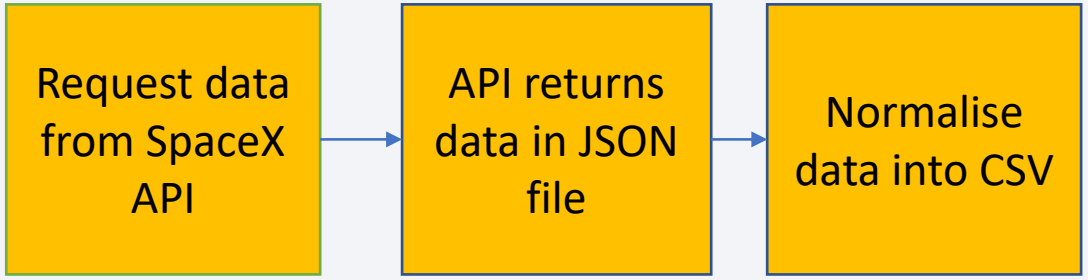
Data Collection

The data collection process includes a combination of API requests from the SpaceX API and web scraping data from a table in the Wikipedia page of SpaceX, Falcon 9 and Falcon Heavy Launches Records.

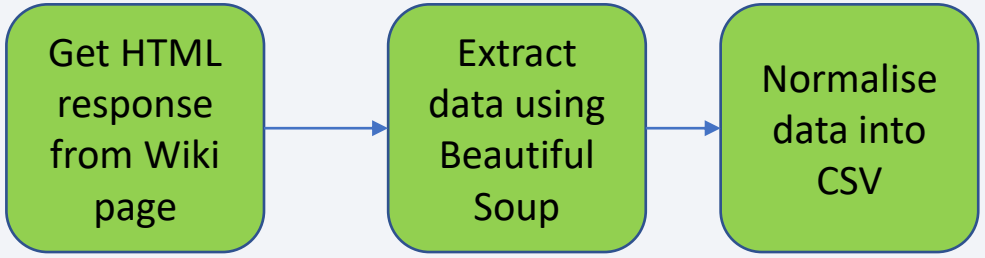
From API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

From Wikipedia: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

API collection flow chart



Webscraping process flow chart



Data Collection – SpaceX API

1. Requesting data from API

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

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

2. Normalising

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

3. Cleaning

```
BoosterVersion[0:5]
```

```
['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4. Creating 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}
```

```
df1= pd.DataFrame.from_dict(launch_dict)
```

Notebook URL

Data Collection - Scraping

1. Getting response from HTML

```
page= requests.get(static_url)
```

2. Parsing into Beautiful Soup

```
soup= BeautifulSoup(page.text, 'html.parser')
```

3. Finding tables

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

4. Getting column names

```
column_names = []


header= first_launch_table.find_all('th')

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

5. Creating dataframe

```
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']=[]
```

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

Notebook URL

Data Wrangling

- There were several cases in which the booster failed to successfully land on the dataset, and sometimes it attempted to land but failed because of accident:
  True Ocean: the mission result has successfully landed in a specific area of the ocean
  False Ocean: the mission result has not successfully landed in a specific area of the ocean
  True RTLS: the mission result successfully landed on the ground pad
  False RTLS: the mission result has not successfully landed on the ground pad
  True ASDS: the mission result has successfully landed on the drone ship
  False ASDS: the mission result has not landed on the drone ship
- Converting these results into training labels:
  0- Failure
  1- Success

Notebook URL

Data Wrangling

1. Number of launches at each site

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

2. Number of occurences of each orbit

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

3. Landing outcome per orbit type

```
landing_outcomes= df.Outcome.value_counts()
landing_outcomes
```

4. Labelling of outcomes

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

5. Calculating success rate

```
df["Class"].mean()
```

```
0.6666666666666666
```

EDA with Data Visualization

- Scatter chart:

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

- Bar chart:

  Orbit Type vs. Success Rate

- Line chart:

  Year vs. Success Rate

Notebook URL

EDA with SQL

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

Notebook URL

Build an Interactive Map with Folium

- Objects created and added to folium map:
  - Markers that show all launch sites on a map
  - Markers that show the success/failed launches for each site on the map
  - Lines that show the distances between a launch site to its proximities
- The markers led to the following conclusions:
  - Launch sites are near railways
  - launch sites are near highways
  - Launch sites are near the coastline
  - Launch sites are far from cities

Notebook URL

Build a Dashboard with Plotly Dash

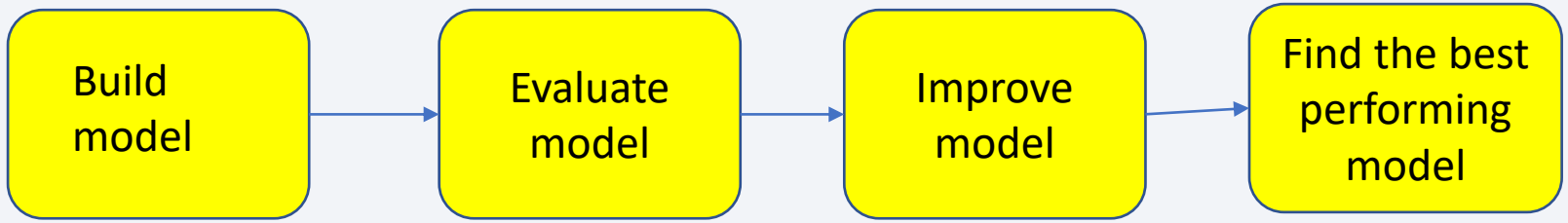
- The dashboard contains a pie chart and scatter chart.
- Pie chart-
  To show the successful launches by site
  To indicate landing distribution across all sites
- Scatter chart-
  To show the relationship outcomes between payload mass and outcomes from different boosters
  Input of payload was based on slider, and the output showed the landing outcomes of all boosters carrying the specified input range
  Helps determining how the success depends on launch point, payload mass and booster variation used.

Notebook URL

Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data

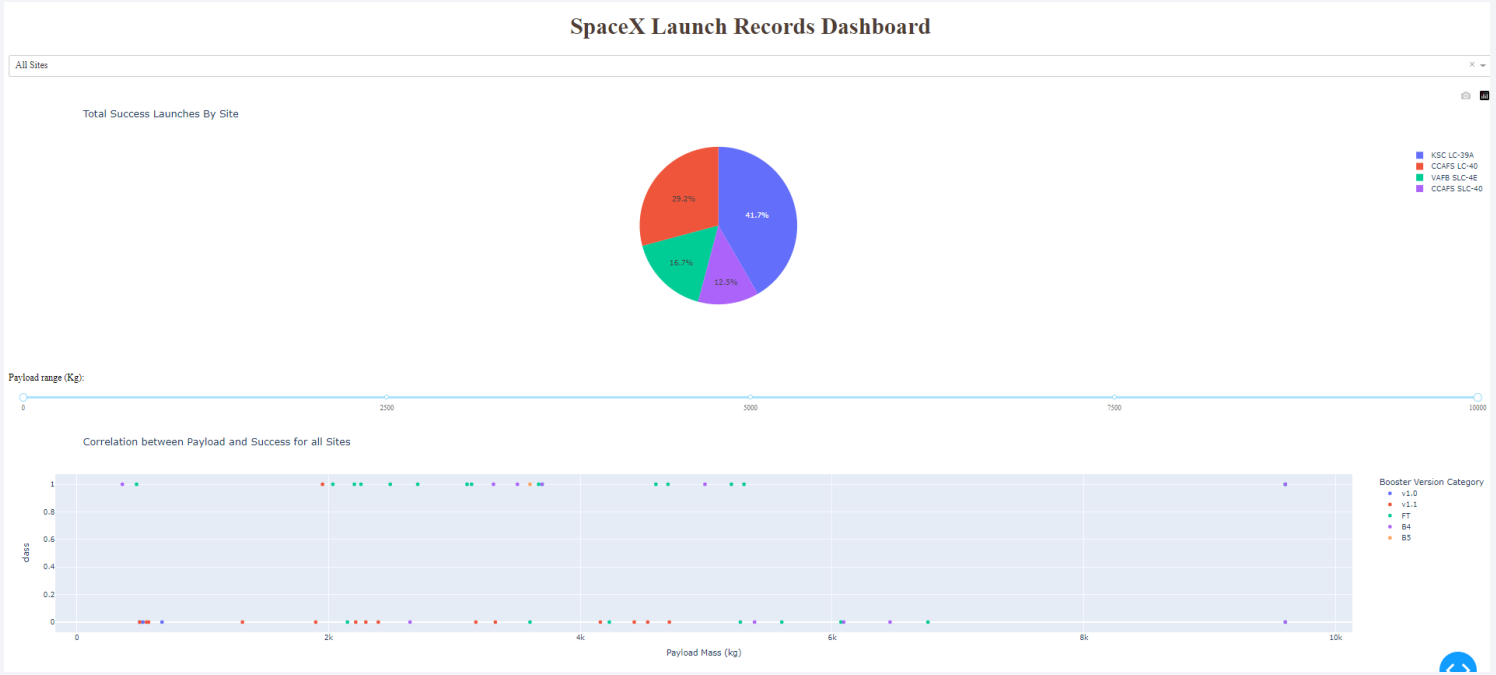
Notebook URL

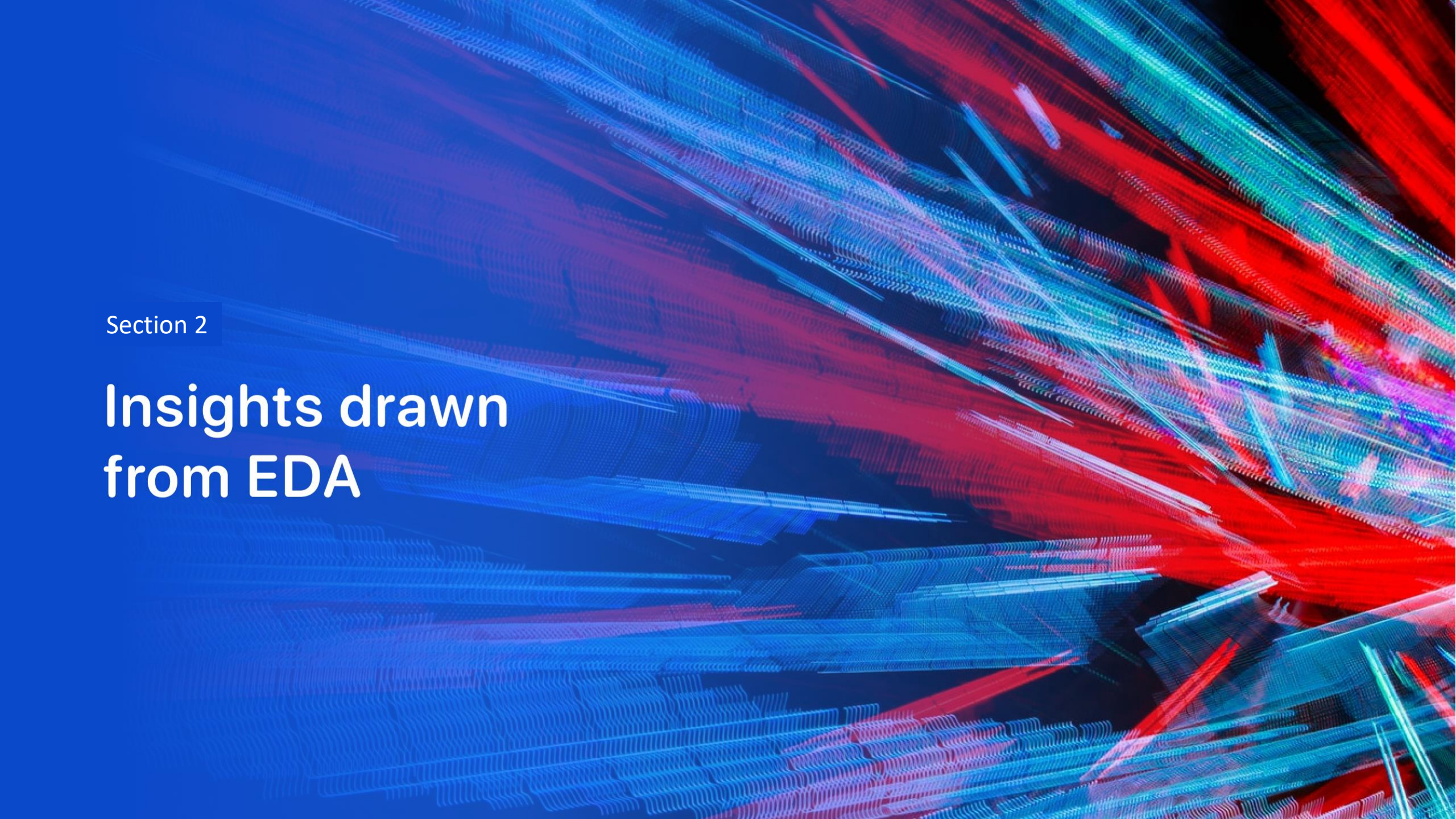


Results

- The results of EDA will be shown in the next slides
- The predictive analysis using Decision tree classifier gave maximum accuracy.

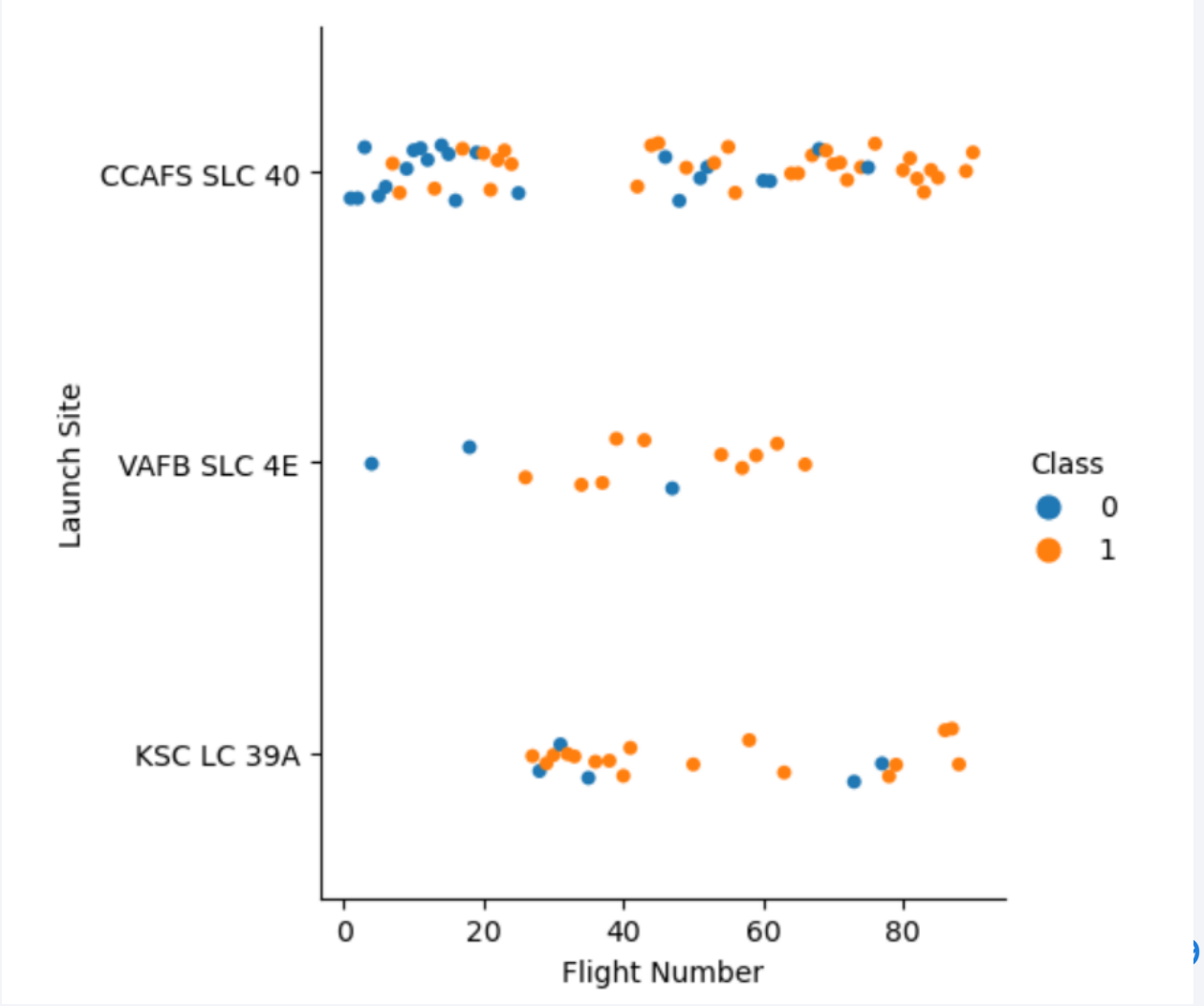
Screenshot of dashboard




Section 2
Insights drawn from EDA

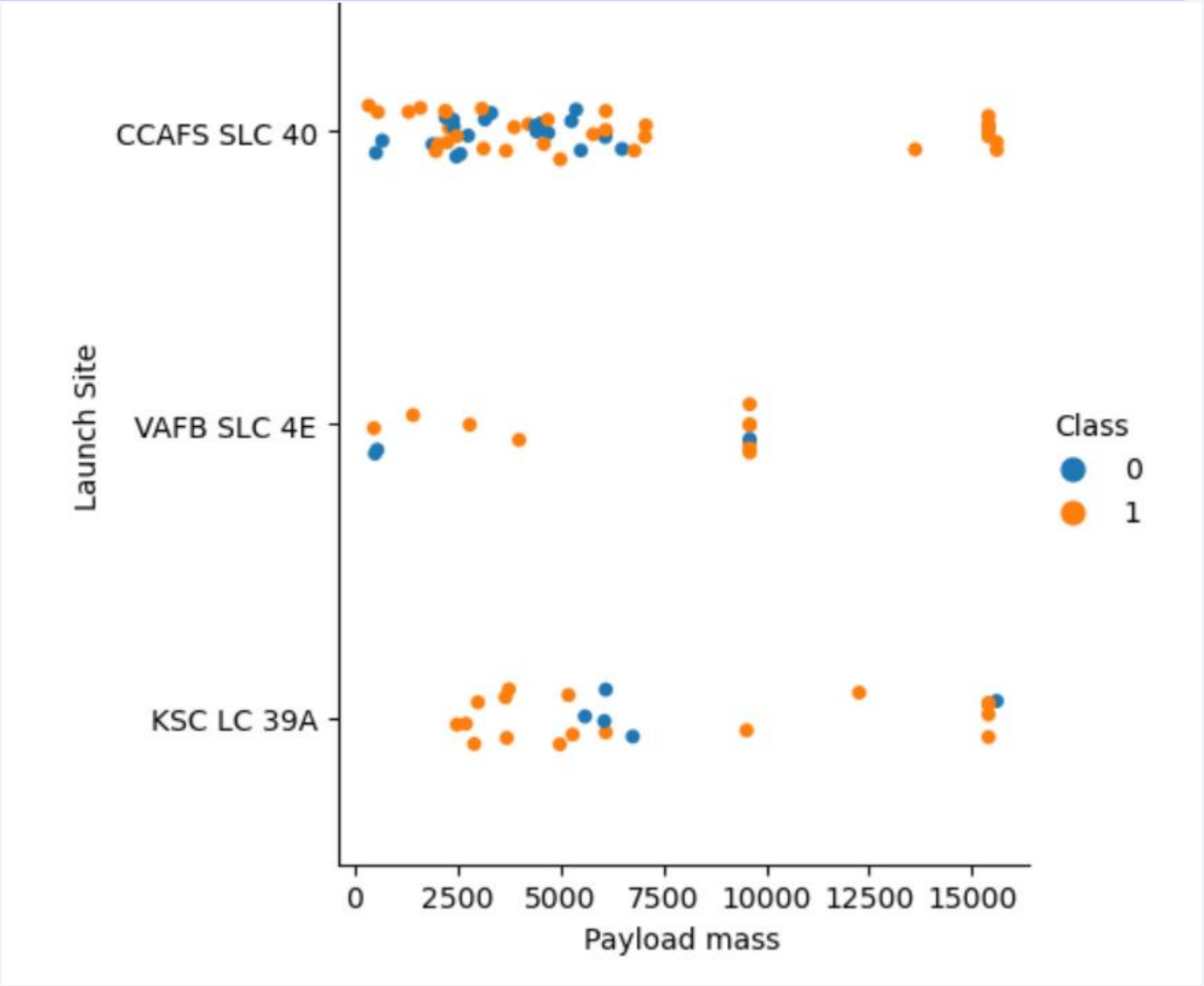
Flight Number vs. Launch Site

- Class 0 gives the failures while Class 1 gives the successful outcomes
- With more number of flights the overall chances of success increase
- After 20 flights or so there is a rapid increase in the successful landings



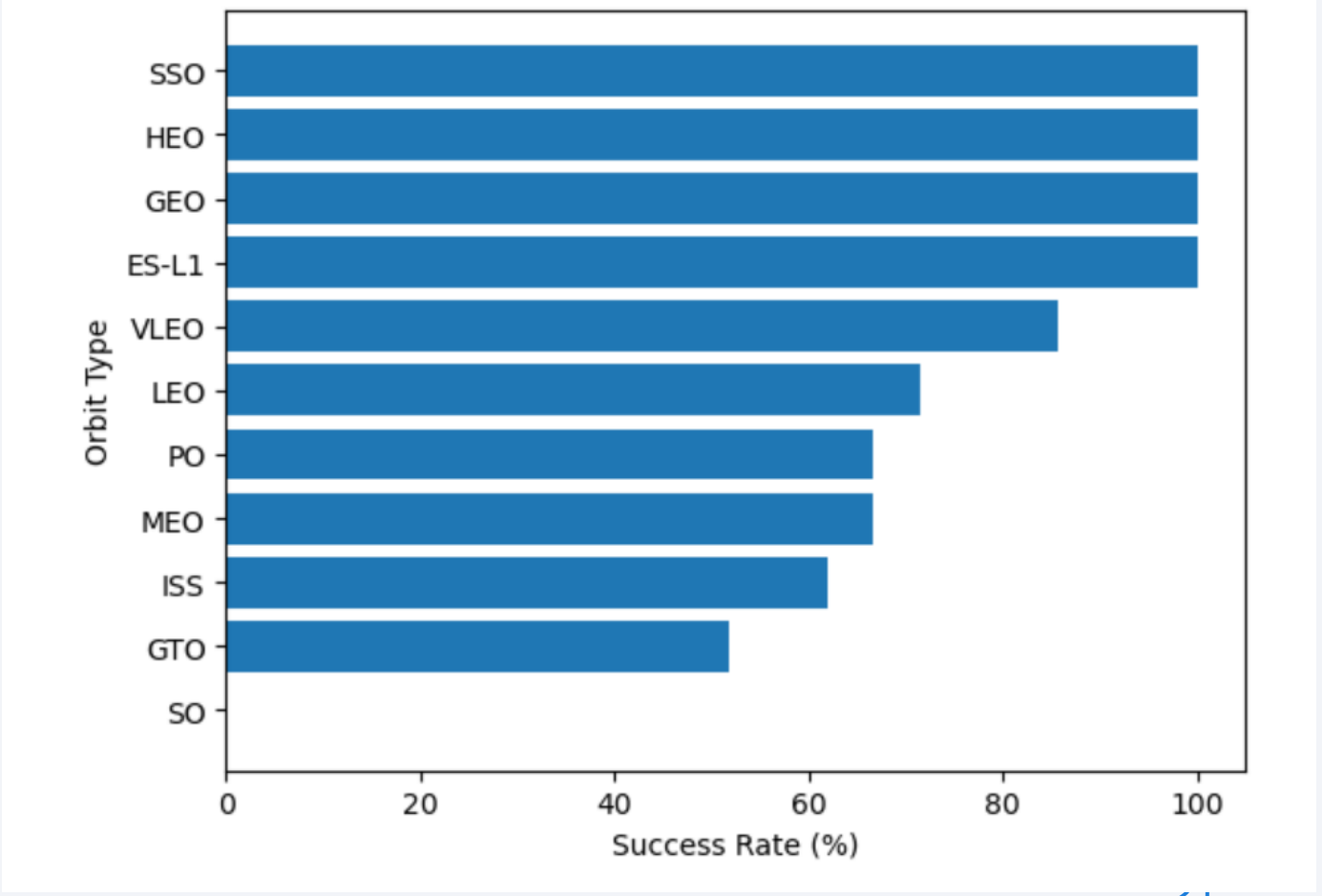
Payload vs. Launch Site

- Class 0 shows failures, while Class 1 shows successes
- Heavier payloads have recorded a higher success percentage but it is not wise to comment on it because of very few such attempts.
- Any patter is unclear at this level of analysis.



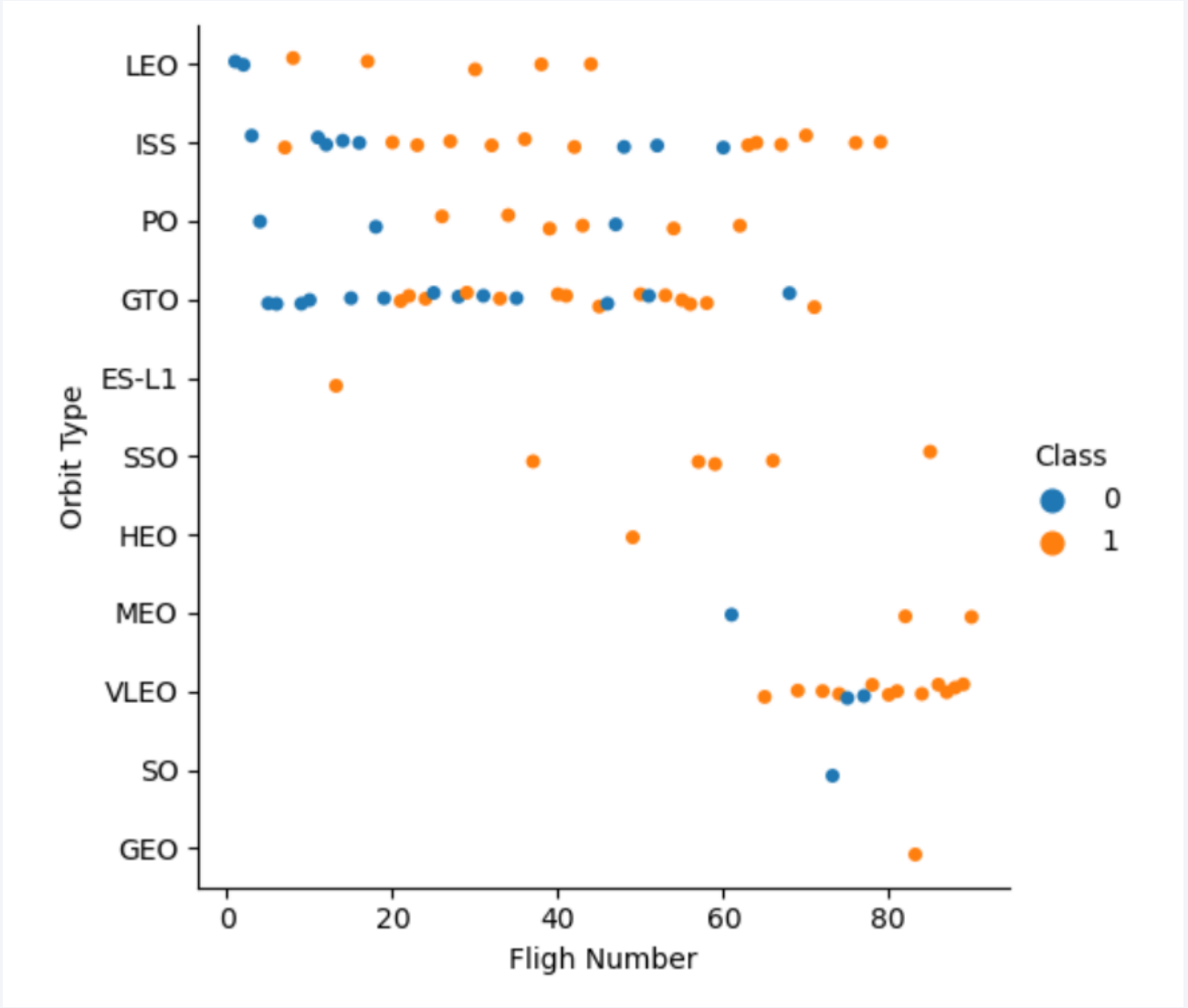
Success Rate vs. Orbit Type

- Orbit types SSO, GEO, HEO and ES-LI have shown the highest success rates with 100% each.
- GTO has shown 50% success rate while SO has not recorded a single success yet.



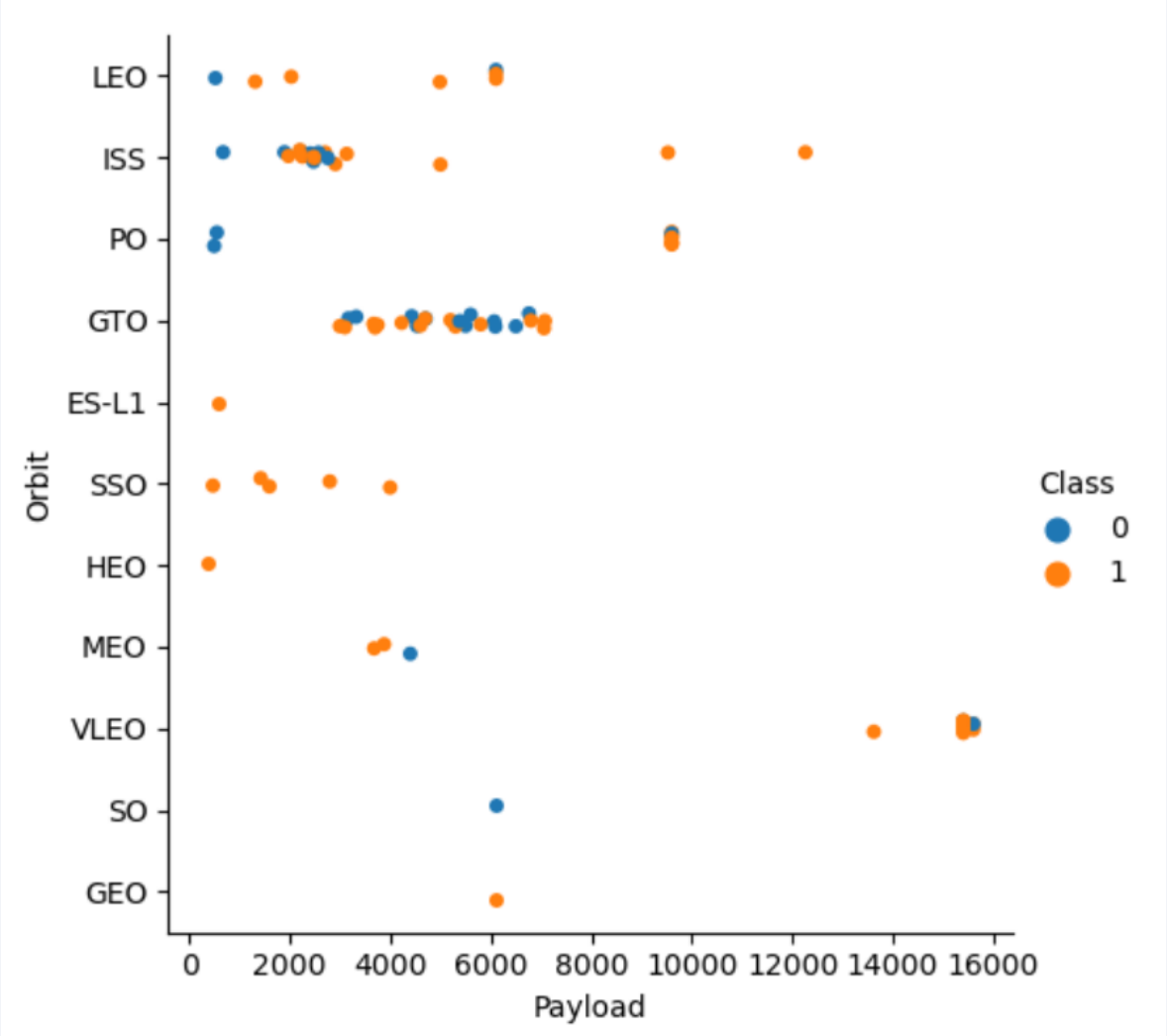
Flight Number vs. Orbit Type

- Class 0 shows unsuccessful attempts while Class 1 shows successful ones.
- In most cases there appears to be a correlation with the flight number and orbit type, but in case of GTO nothing of this kind can be said concretely
- SpaceX starts with LEO with a moderate success rate, and it seems that VLEO, which has a high success rate, is used the most in recent launches



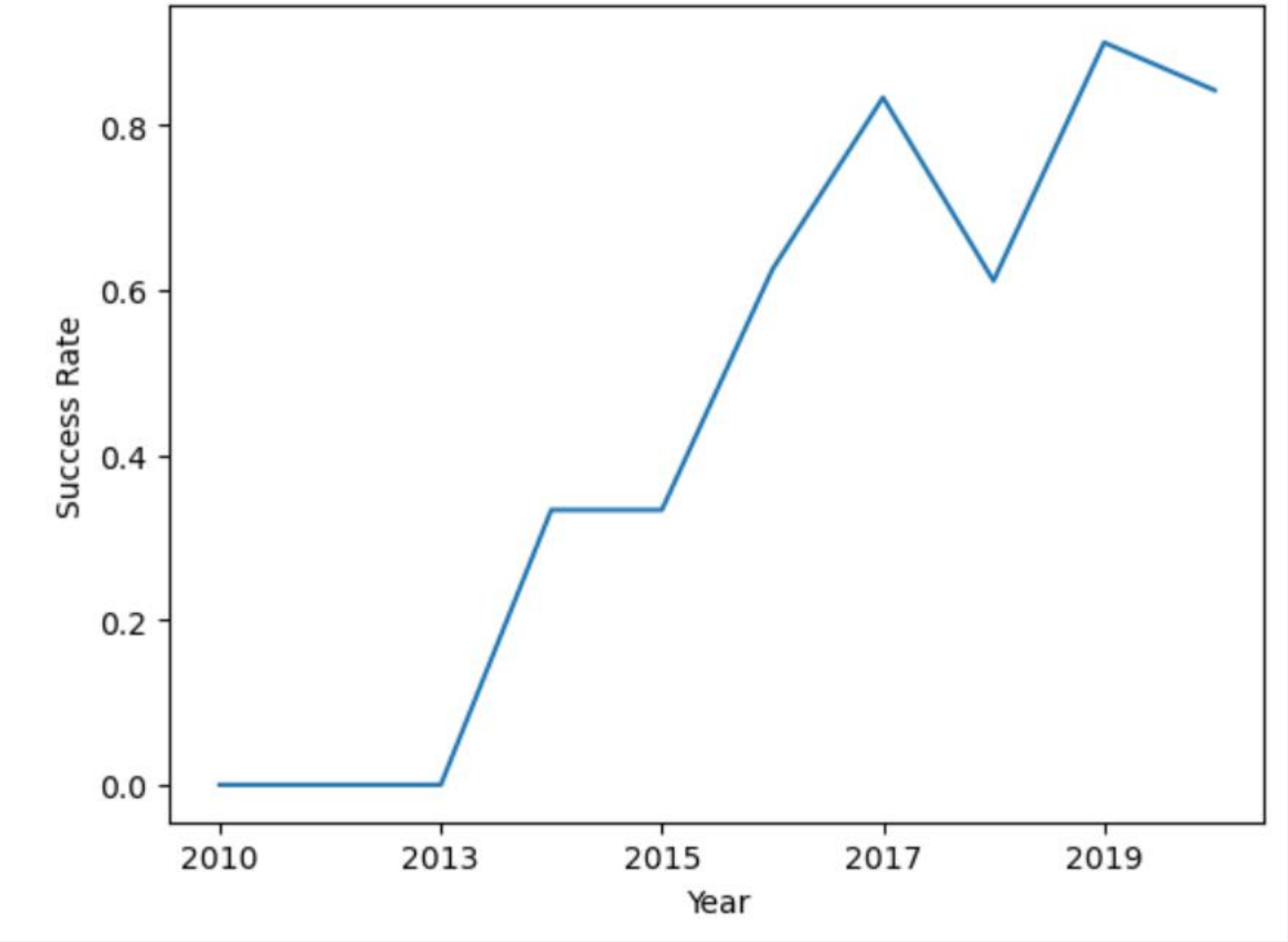
Payload vs. Orbit Type

- Class 0 shows unsuccessful, Class 1 shows successful launches
- LEO and ISS orbit types show higher success rates with heavier payloads
- In the case of GTO again, it is hard to find a correlation as such.



Launch Success Yearly Trend

- Since 2013 the success rate increased till 2017.
- It took a dip in 2018, while regained traction in 2019 when it reached its peak at about 90%
- Currently it stands near 80%



All Launch Site Names

There are four distinct launch sites here- CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, and CCAFS SLC-40

```
%%sql
select distinct(Launch_Site) from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

- 5 records were displayed here for launch sites beginning with CCA, since the limit had been set to 5.

```
%%sql

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

```
 * sqlite:///my_data1.db
Done.
```

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

Total Payload Mass

- The aggregate of all payload mass carried in the missions where the customer was NASA (CRS) was 45596kg

```
%%sql

select sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTBL
    where Customer= "NASA (CRS)"
```

```
 * sqlite:///my_data1.db
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 was 2928.4 kg per mission.

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

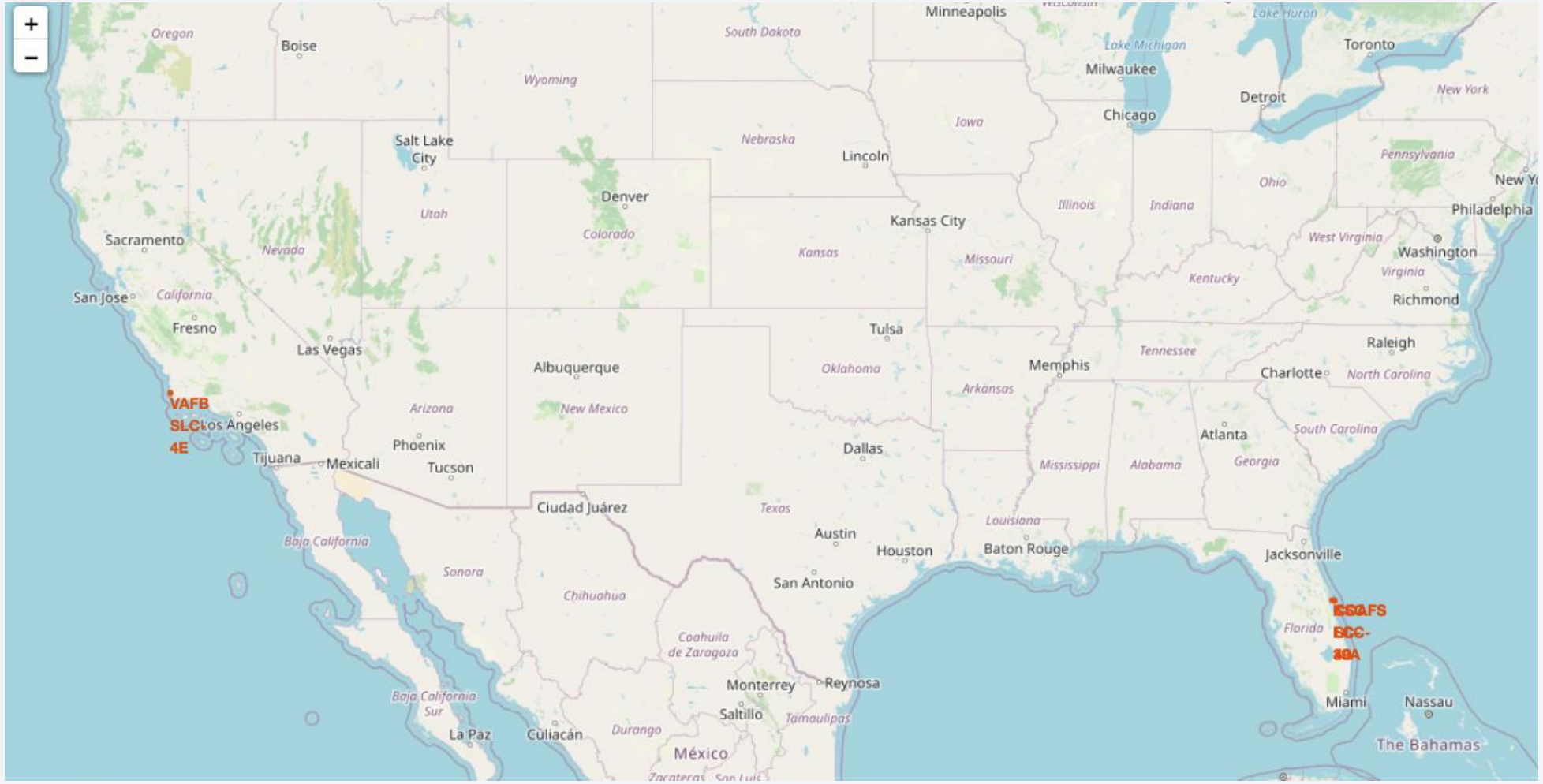
```
 * sqlite:///my_data1.db
Done.
```

avg_payload_mass
2928.4

Section 3

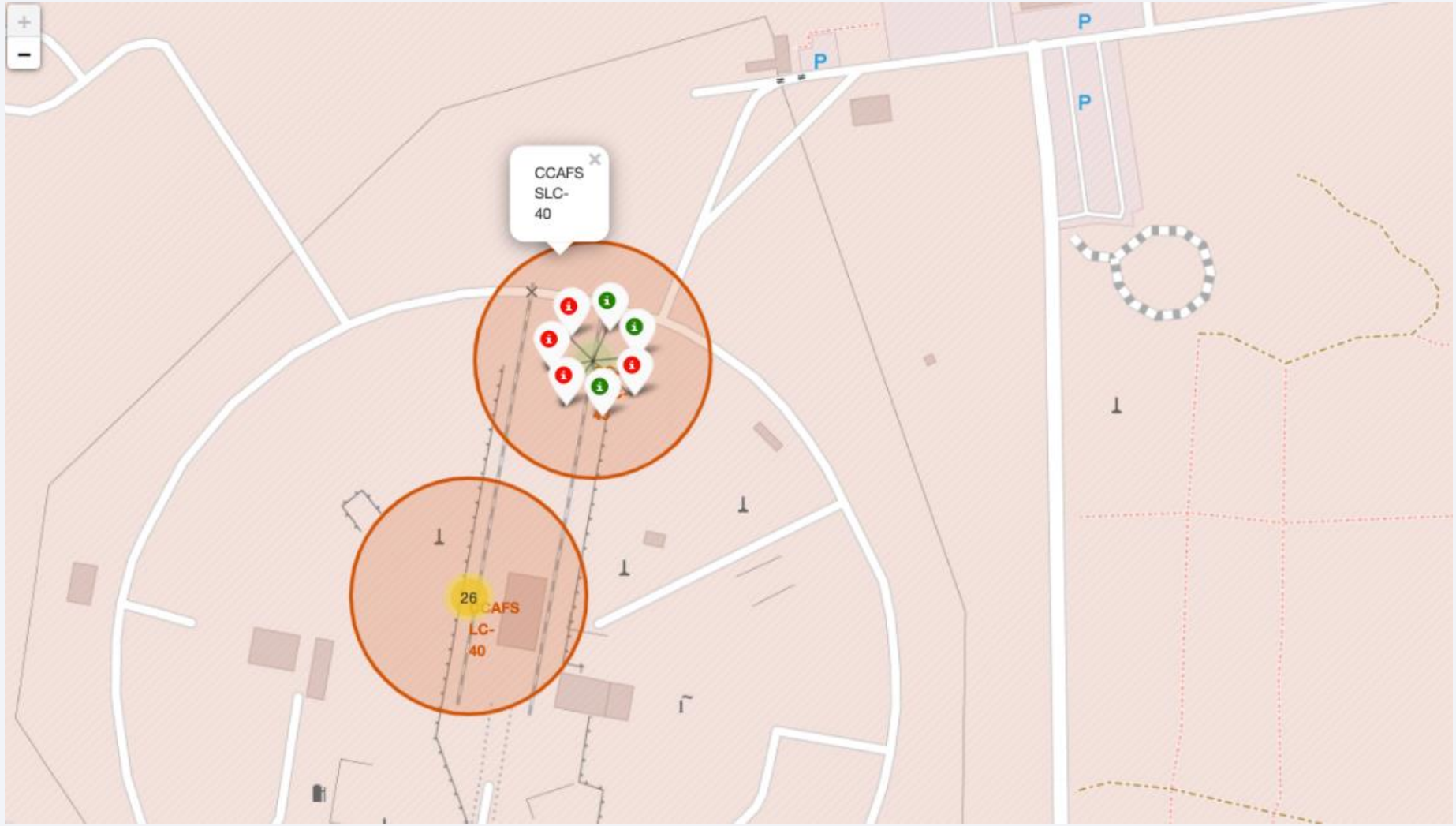
Launch Sites Proximities Analysis

All launch sites



- Most launch sites were on the US East coast off Florida

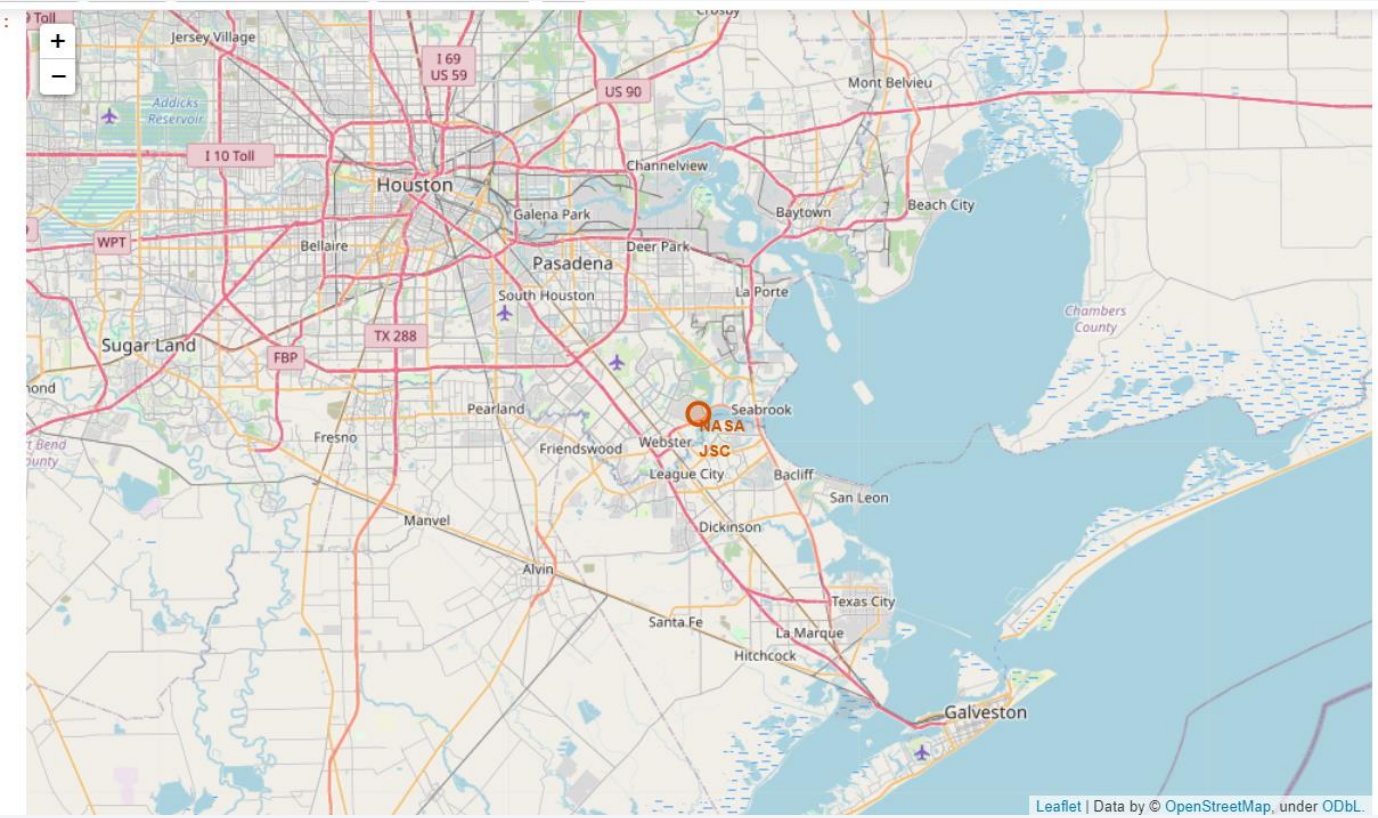
Landing markers

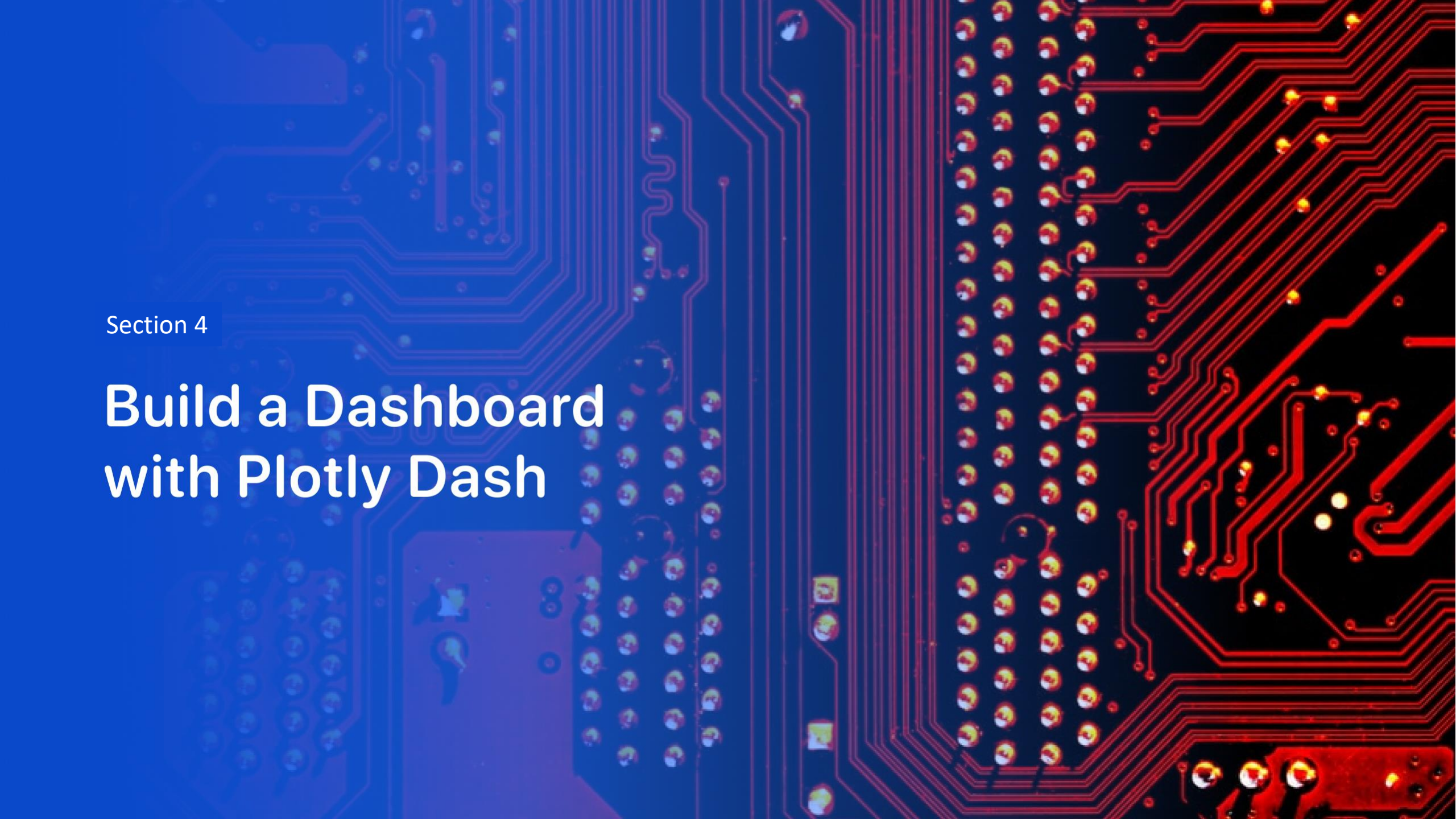


- The green markers imply a successful landing while the reds imply unsuccessful attempts

Launch site proximity

- Launch site is far from the city center, and closer to the sea.

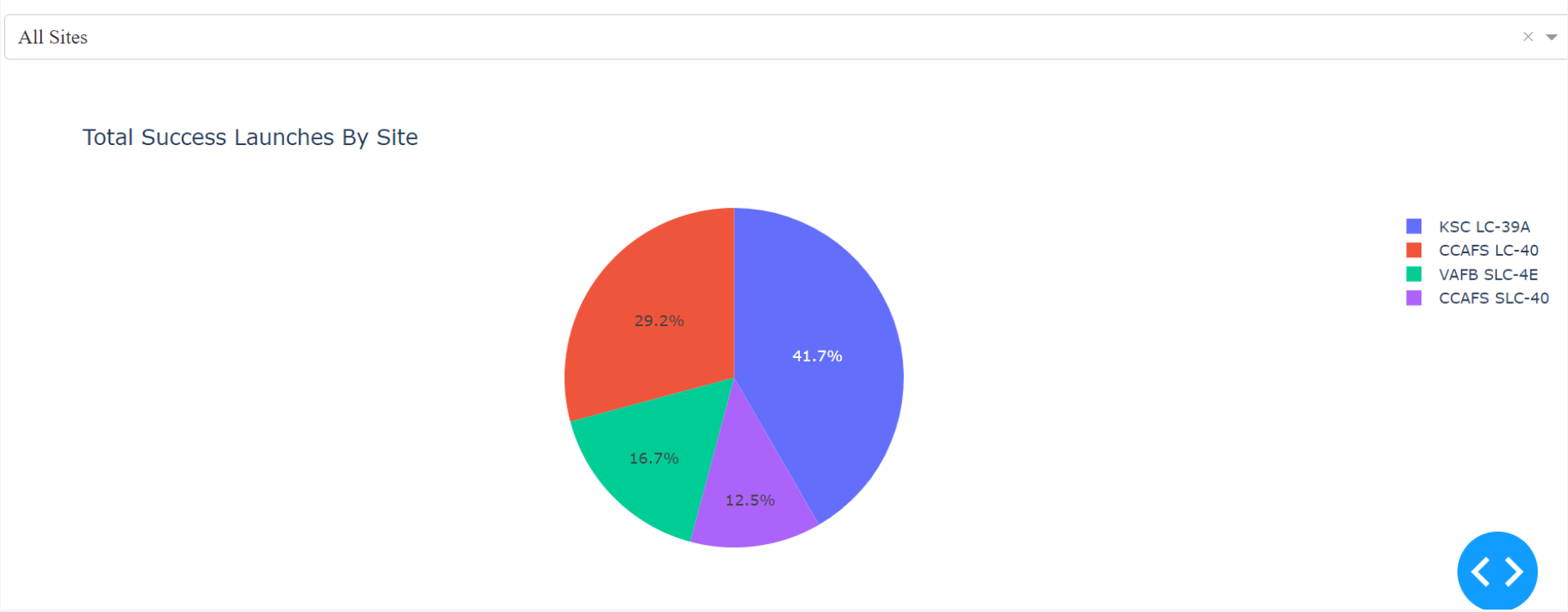



Section 4
Build a Dashboard with Plotly Dash

Total success launches by site

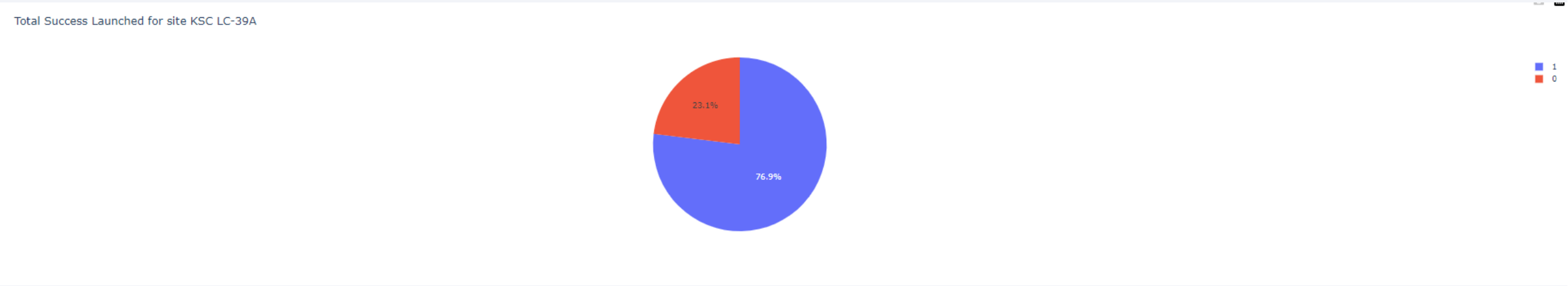
KSLC-39A records the most launch success among all sites.

The VAFB SLC-4E has the fewest launch success, possibly because a)the data sample is small, or b) because it is the only site located in California, so the launch difficulty on the west coast may be higher than on the east coast.



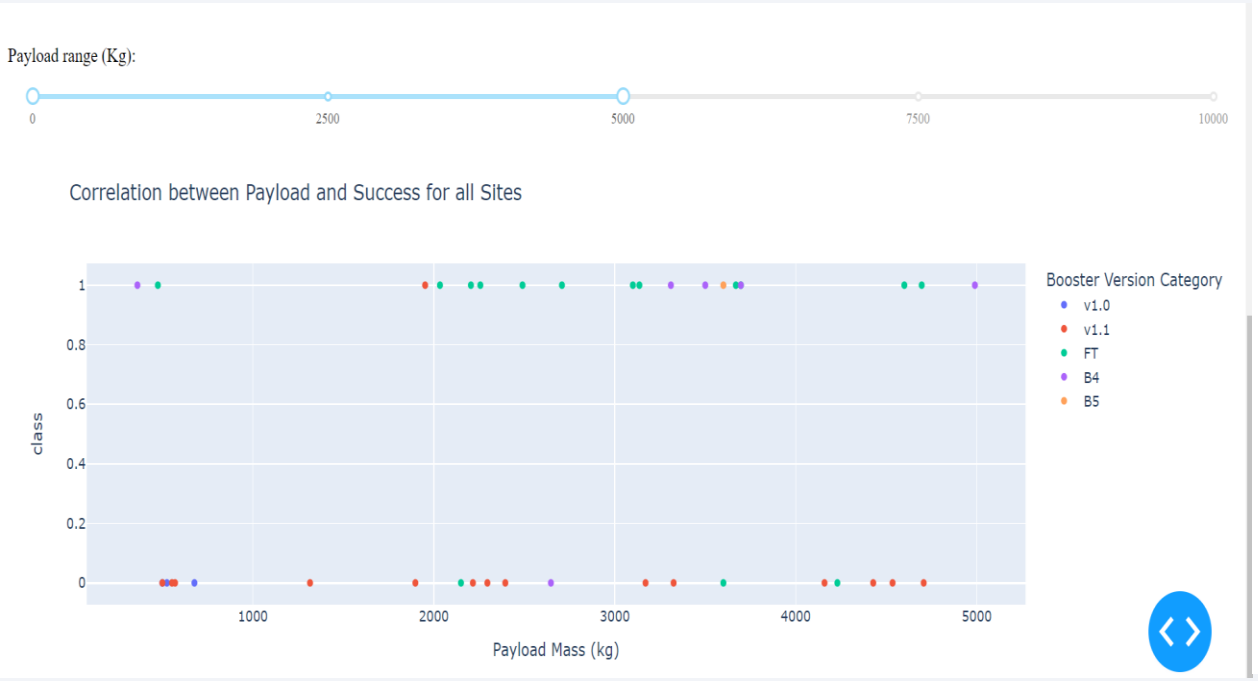
Site with the highest success rate

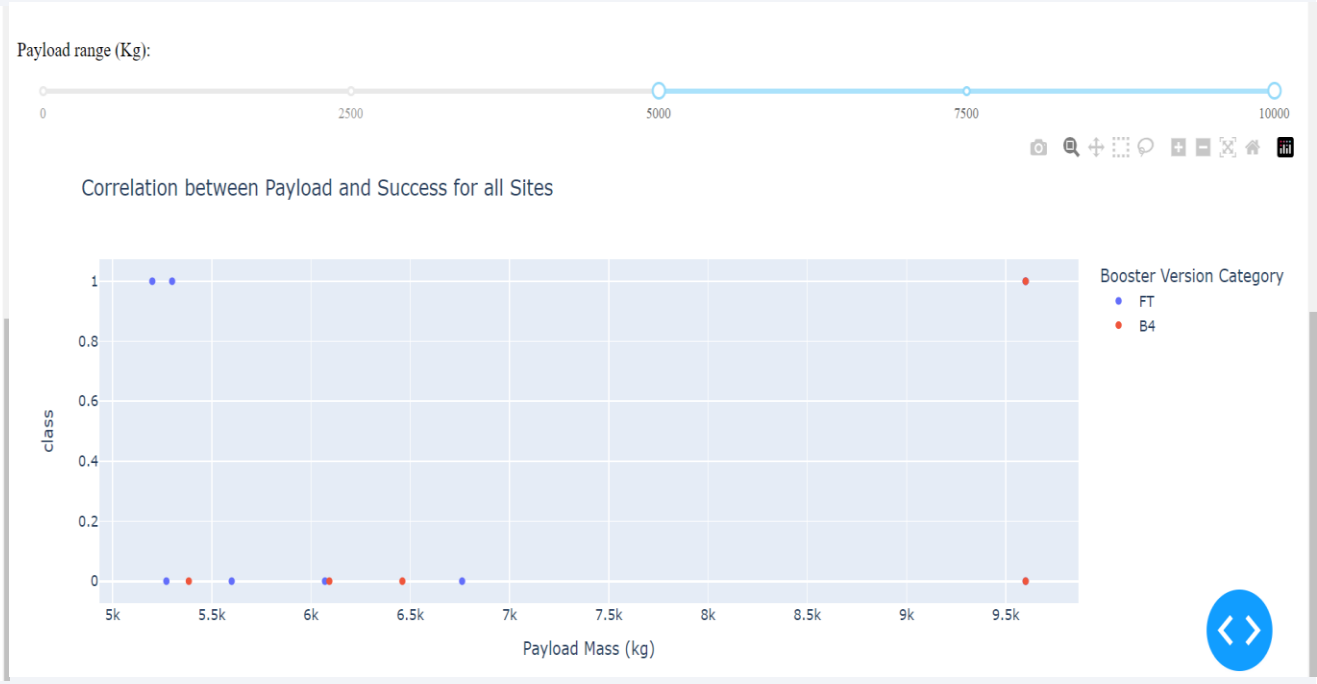
KSC LC-39A shows the highest success rate at 76%



Payload vs Launch outcome for all sites

Success rates (class 1) for low payloads is seemingly higher than those with high payloads



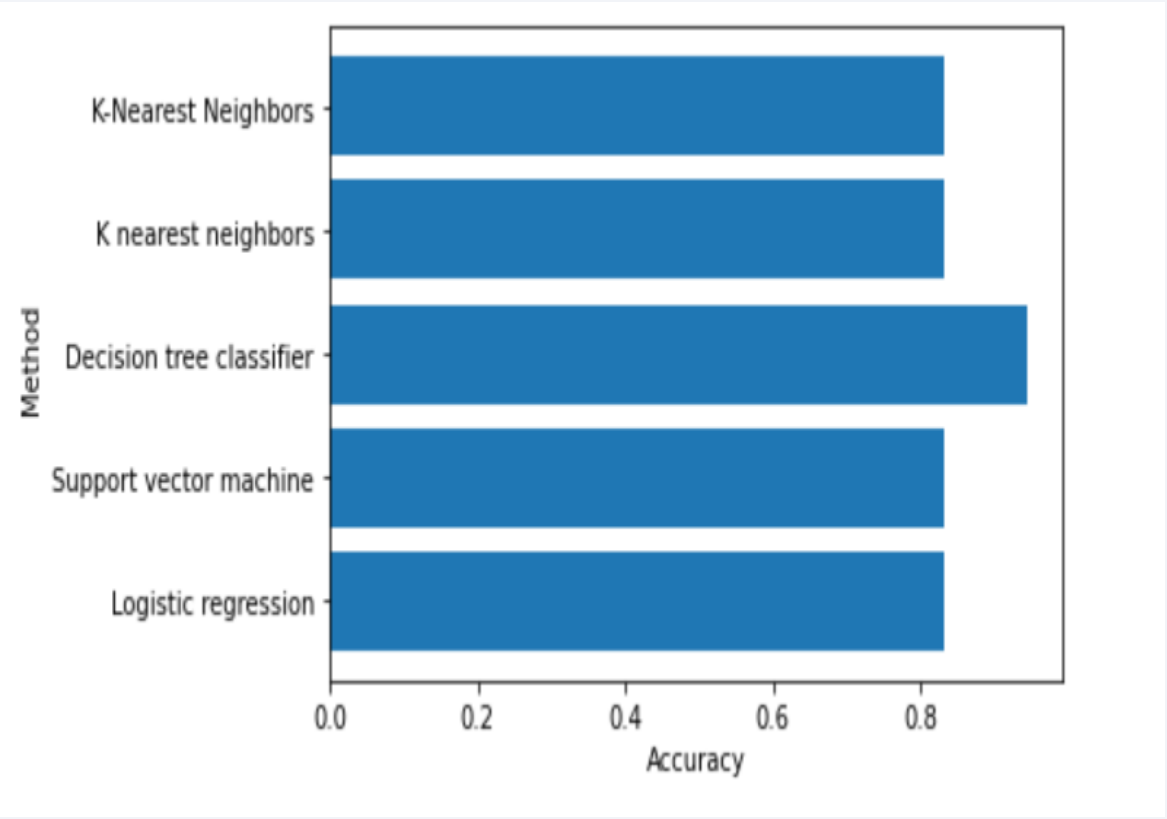


Section 5

Predictive Analysis (Classification)

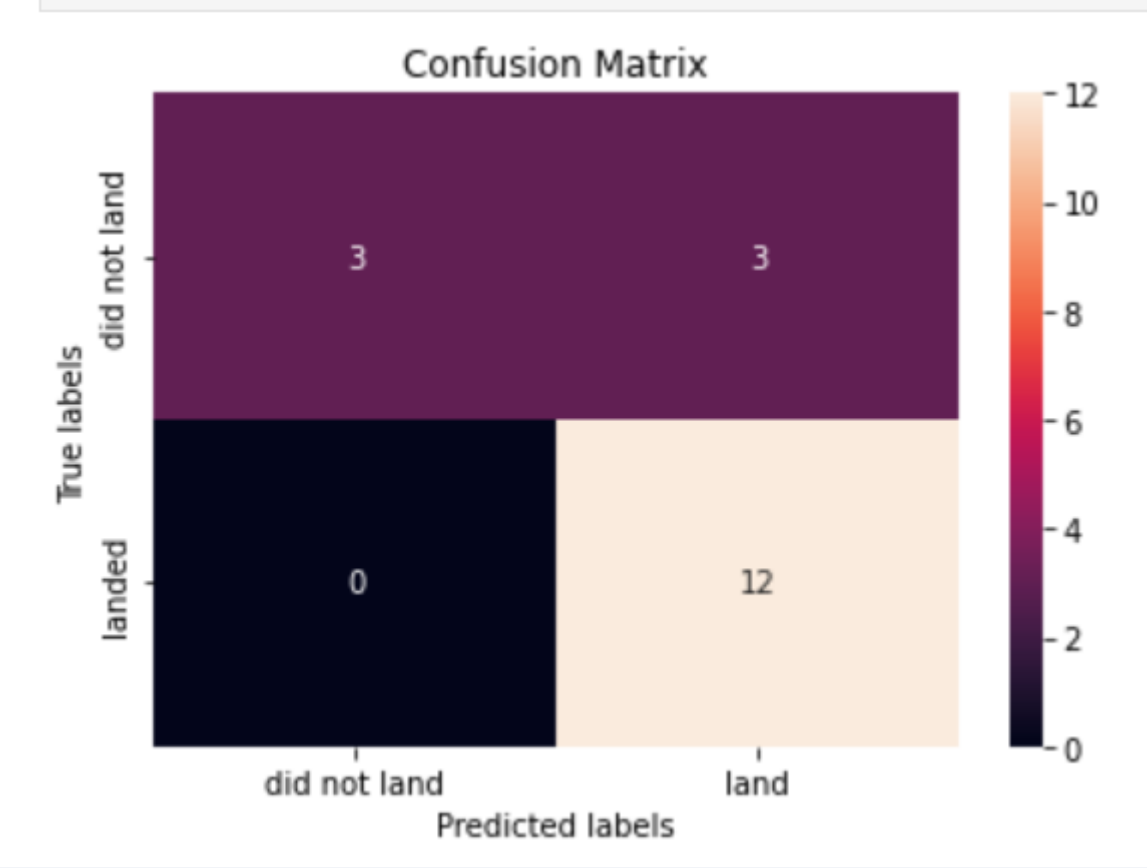
Classification Accuracy

- The decision tree classifier gave the best accuracy at close to 95%
- The test size was small, at 18.



Confusion Matrix

- The confusion matrix was the same for all models



Conclusions

- The success rates have increased with an increase in number of flights
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%).
- The launch site is close to railways, highways, and coastline, but far from cities.
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites.
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads.
- The decision tree classifier was the most accurate with a CV score of 95% and this is good given a small dataset but with a larger dataset we may find a more optimal classifier.

Appendix

Github URL

Coursera data science capstone course URL

Thank you!