Winning Space Race with Data Science

Wong Sze Tong
16 December 2021

Outline

Executive Summary

- Summary of methodologies
  - Data Collection with Web Scraping
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Create Interactive Visualization with Folium
  - Create Dashboard with Plotly Dash
- Summary of all results
  - Data Analysis on the results
  - Machine Learning prediction

Introduction

- Project background and context

  Commercial space age is here, companies are making space travel affordable for everyone. Perhaps the most successful companies is SpaceX. SpaceX's accomplishments include: spacecraft to the international Space Station. Starlink, a satellite internet constellation providing satellite internet access.

  The reason why SpaceX become one of the top in this industry is because SpaceX rocket launches are relatively inexpensive. SpaceX advertised Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollar each and much of the savings is because SpaceX can reuse the first stage of their rocket.

  In this project, we would like to find out the price for each rocket launch for company SpaceY, competitor of SpaceX. Since the rocket cost highly depends on the successful rate of reuse of the first stage of racket, we would like to study parameter of the First stage of rocket on successful launch.

- Project Objective - Problems you want to find answers

  As a Data Scientist for SpaceY, we would like to find out how the parameters in the first stage affect on the successful landing rate of Falcon 9. Can we find out the best parameters in the first stage to ensure the successful landing rate of our First Stage rocket so that we can able to determine the best cost for the Rocket Launch.

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - SpaceX REST API & Web Scaping from Falcon 9
  - Falcon Heavy Launches Records from Wikipedia using Beautiful Soap.
- Perform data wrangling
  - To convert those outcomes into Training Labels with 1 (Booster successfully landed) & 0 (Unsuccessful)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Obtain the best Hyperparameters for SVM, Classification Trees and Logistic Regression.

Data Collection

- The Data collection process includes a combination of API request from SpaceX REST API
- The API provide us information such as: Information of rocket, payload delivered. Launch specification, Landing specification, Landing outcome, location and etc.
- Using BeautifulSoup for web scaping on Wikipedia (Falcon 9 Launch data information)

GITHUB URL

Data Collection – SpaceX API

1. Getting Response from SpaceX Rest API

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

response = requests.get(spacex_url)
```

2. Converting Response to JSON file

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

3. Apply custom function to clean Data

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Apply list to Data Frame

```
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}
```

4. Apply list to Data Frame

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion']=='Falcon 9']
```

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

GITHUB URL

Data Collection – Web Scraping

GITHUB URL

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"
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data, 'html5lib')
```

2. Creating BeautifulSoup Object

```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url).text
```

3. Getting Columns 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)
```

4. Creating Dictionary

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

# Let's initial the launch_dict with
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']=[]
```

5. Appending Data

```
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()
```

6. Final DataFrame

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

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example,

- True Ocean = mission outcome was successfully landed to a specific region of the ocean
- False Ocean = mission outcome was unsuccessfully landed to a specific region of the ocean.
- True RTLS = mission outcome was successfully landed to a ground pad
- False RTLS = mission outcome was unsuccessfully landed to a ground pad.
- True ASDS = mission outcome was successfully landed on a drone ship
- False ASDS = mission outcome was unsuccessfully landed on a drone ship.

We will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

GITHUB URL

Perform Exploratory Data Analysis EDA on Dataset

Calculate the Number of Launch at each site

```
# Apply value_counts() on column LaunchSite
df.value_counts(df['LaunchSite'])
```

Calculate the Number of Occurrence of each orbit

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Calculate the Number of Occurrence of mission outcome per orbit type

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

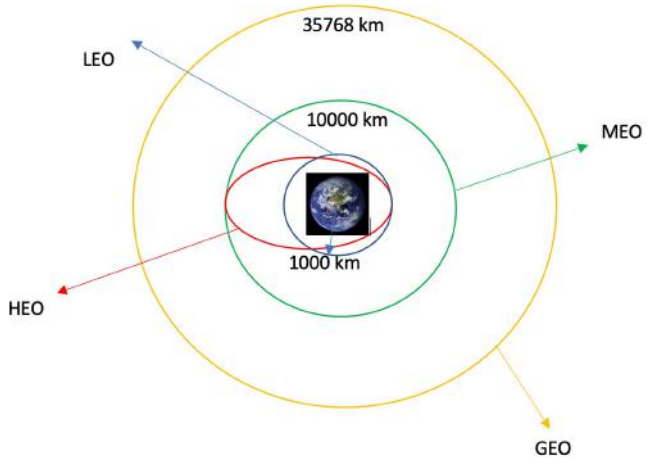
Create a landing outcome label from Outcome column

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

Work out success rate for every landing in dataset

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

```
0.6666666666666666
```



EDA with Data Visualization

Scatter Graph

1.0 Flight Number VS Launch Site

2.0 Payload VS Launch Site

3.0 Flight Number VS Orbit Type

4.0 Payload VS Orbit Type

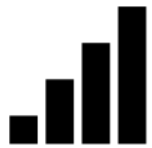
Scatter Plot show how much one variables is affected by another. Using Scatter plot, we can check their correlation between 2 variables.

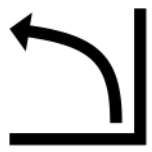
Bar Chart

1.0 Orbit Type VS Success Rate

Bar Chart make easy to compare dataset between multiple group at a glance

Bar Chart show big changes in data over time





Line Chart

1.0 Success Rate VS Year

Line Chart show data variables and trends very clearly and help to make prediction about results of data not yet recorded

GITHUB URL

EDA with SQL

GITHUB URL

Performed SQL queries to gather information about the dataset.

Displaying the names of the unique launch sites in the space mission
Displaying 5 records where launch sites begin with the string 'KSC’
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 where the successful landing outcome in drone ship was achieved.
Listing the names of the boosters which have success in ground pad 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 records which will display the month names, successful landing outcomes in ground pad ,booster
versions, launch site for the months in year 2017
Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

Object created and added to a folium map:

- Markers that show all launch sites on 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

By adding these objects, following geographical patterns about launch sites are found:

- Are launch sites in close proximity to railways? Yes
- Are launches sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

GITHUB URL

Build a Dashboard with Plotly Dash

The dashboard application contains a pie chart and a scatter point chart.

Pie Chart

- For showing total success launches by sites
- This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.

Scatter Chart

- For showing the relationship between Outcomes and Payload mass (kg) by different boosters.
- Has 2 inputs: All sites/ individual site & Payload mass on a slider between 0 and 10000 kg
- This chart helps determine how success depends on the launch point, payload mass, and booster version categories

GITHUB URL

Predictive Analysis (Classification)

GITHUB URL

Building Model

Load our dataset into Numpy and Pandas

Transform Data

Split our data into training and test data sets

Check how many test samples

Decide on which type of machine learning algorithms to apply

Evaluating Model

Check accuracy for each model

Get tuned hyperparameters for each type of algorithms

Improving Model

Feature Engineering

Algorithm Tuning

Finding The best Performing Classification Model

The model with the best accuracy score wins the best performing model

Results

GITHUB URL

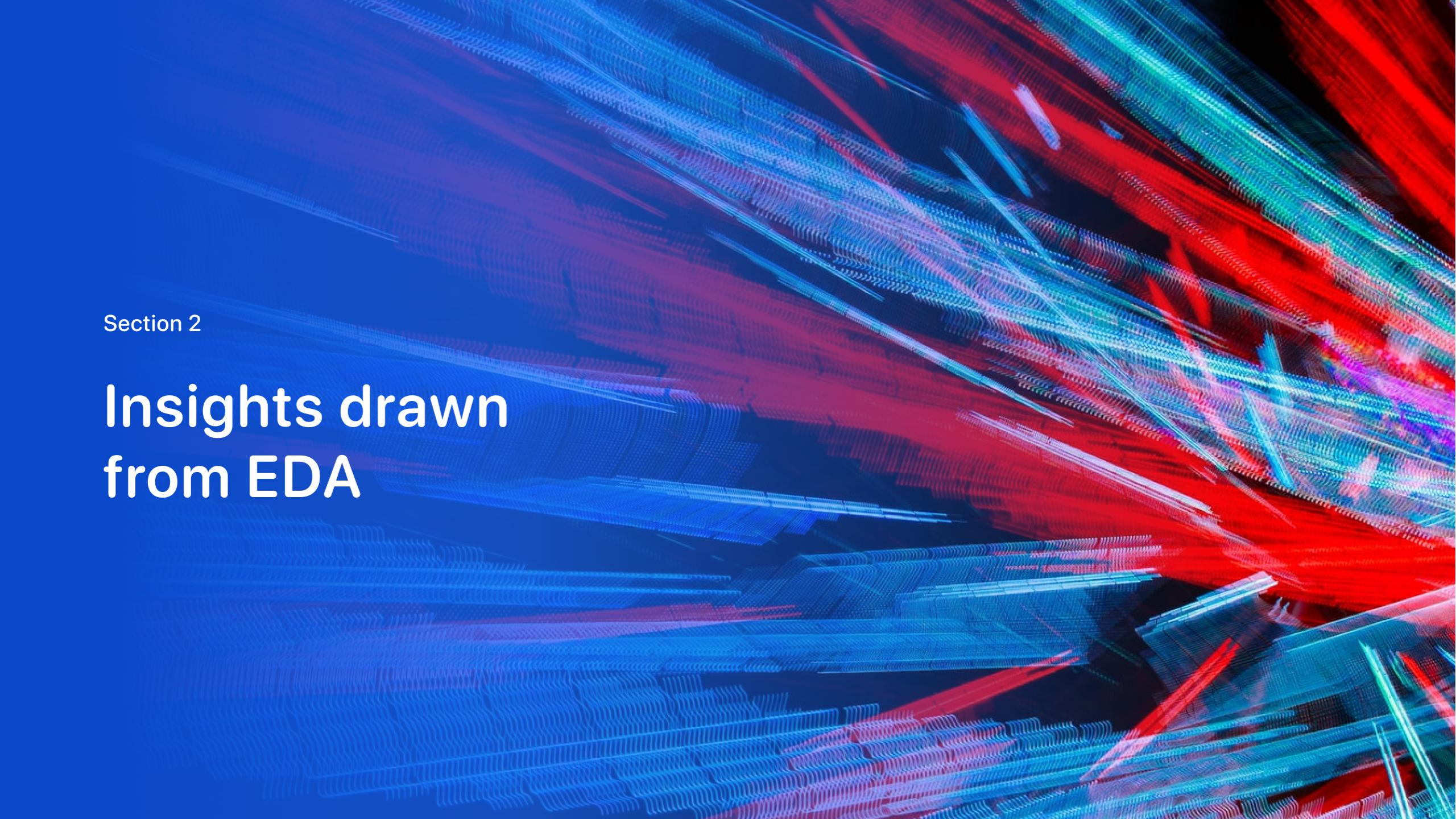
EXPLORATORY DATA ANALYSIS RESULTS

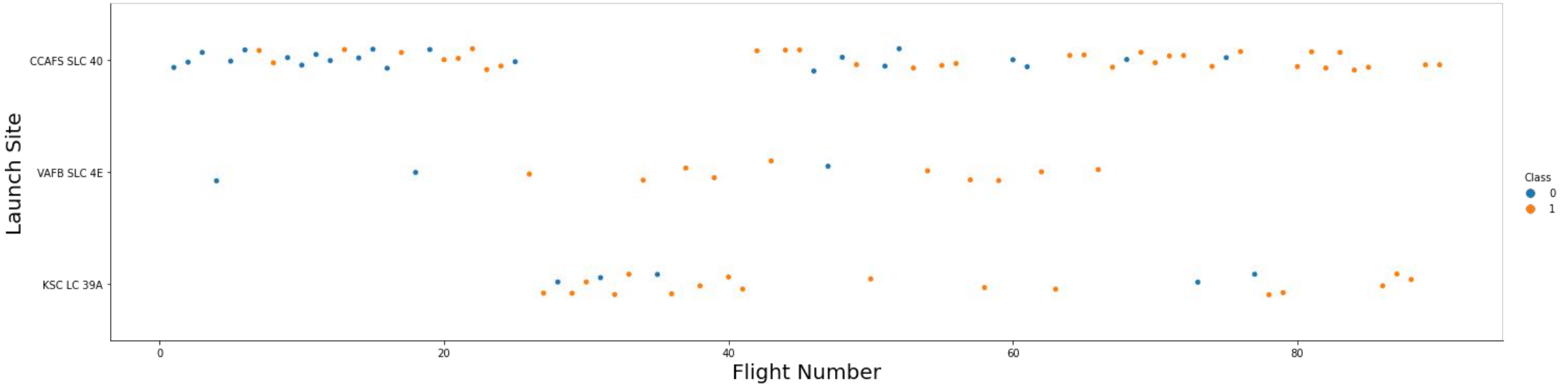
INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

Section 2

Insights drawn from EDA

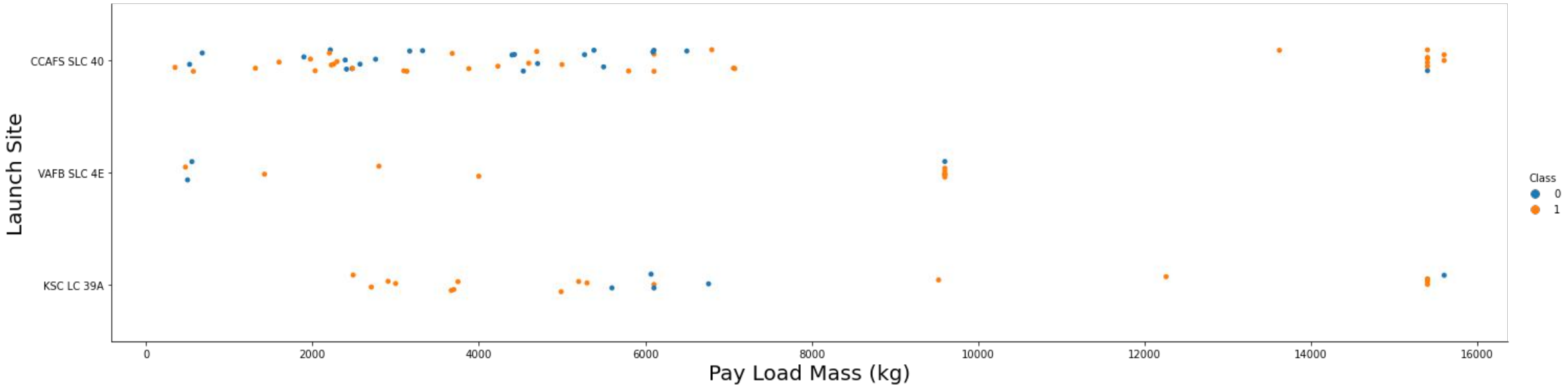




- Class 0 (Blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- This figure shows that the success rate increased as the number of flights increased
- As the success rate has increased considerably since the 20th flights. This point seems to be a big breakthrough

Flight Number vs. Launch Site

GITHUB URL



- Class 0 (Blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- At first glance, the larger pay load mass, the higher the rocket's success rate, but it seems difficult to make decisions based on this figure because no clear pattern can be found between successful launch Pay Load Mass

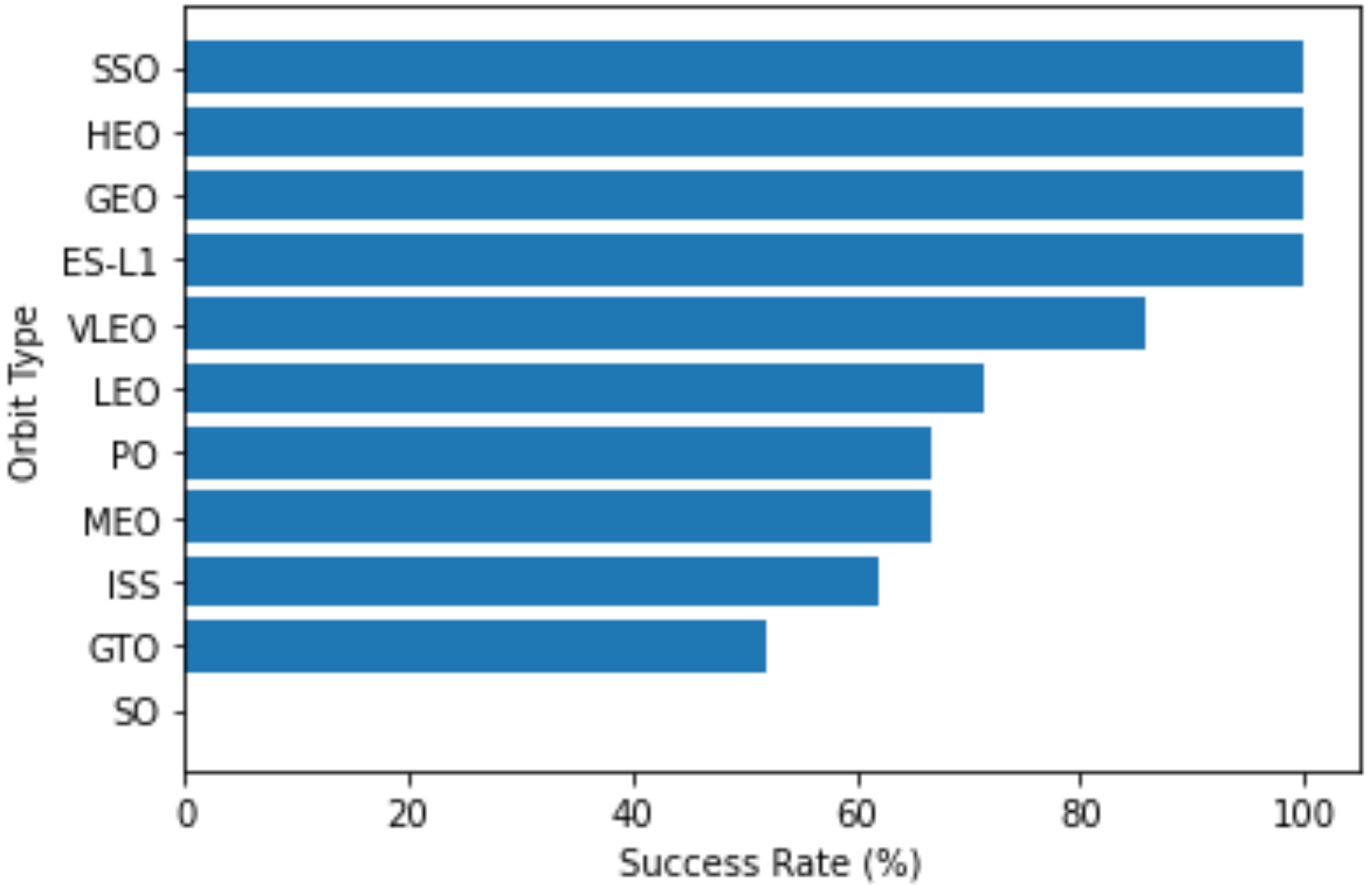
Payload vs. Launch Site

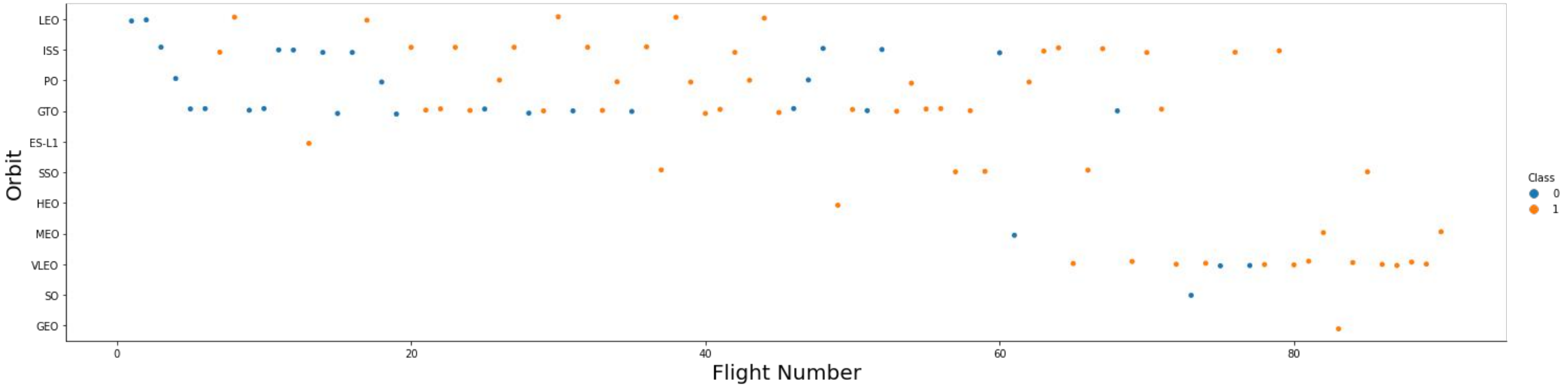
GITHUB URL

Success Rate vs. Orbit Type

- Orbit types SSO, HEO, GEO, and ES-L1 have the highest success rates (100%)
- On the other hand, the success rate of orbit type GTO is only 50%, and it is the lowest except for type SO, which recorded failure in a single attempt.

GITHUB URL

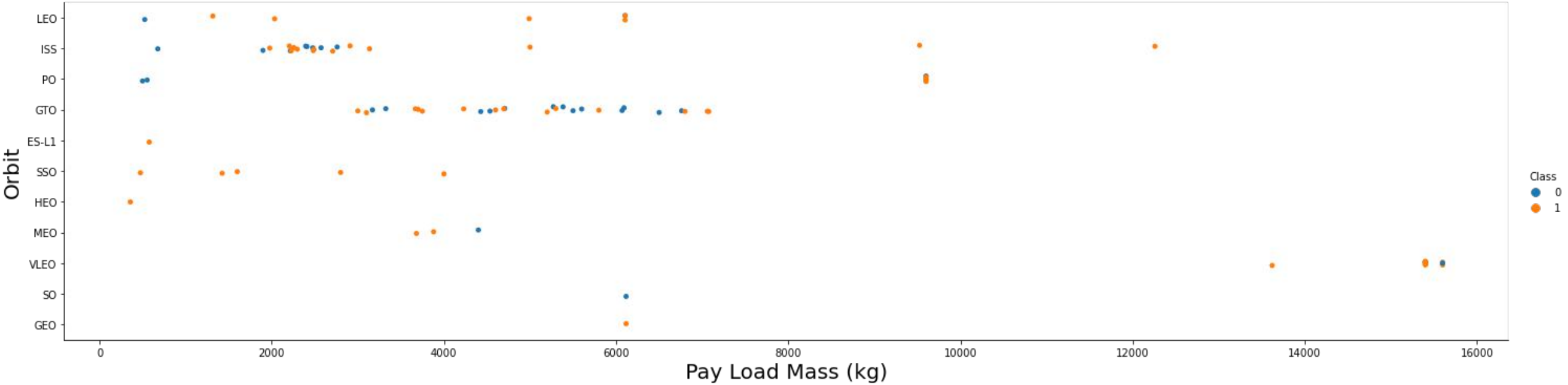




- Class 0 (Blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- In most cases, the launch outcome seems to be correlated with the flight number.
- On the other hand, in GTO orbit there seems to be no relationship between flight numbers and success rate.
- SpaceX 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

Flight Number VS Orbit Type

GITHUB URL



- Class 0 (Blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- With heavy payloads, the successful landing rate are higher for LEO and ISS.
- However, for GTO case, it is hard to distinguish between the positive landing rate and the negative landing because they are all gathered.

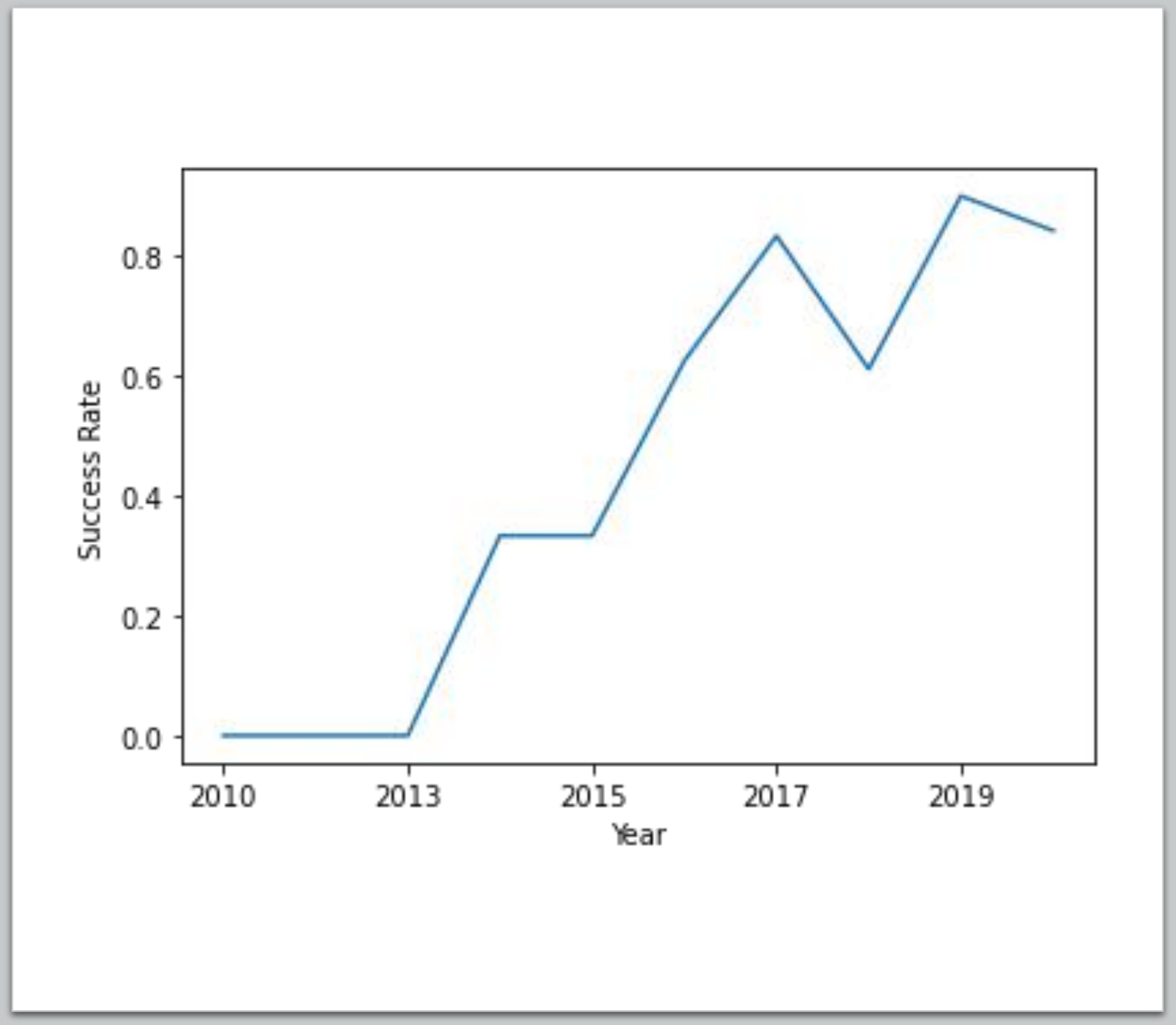
Payload VS Orbit Type

GITHUB URL

Launch Success Yearly Trend

- Since 2013, the success rate has continued to increase until 2017
- The rate decreased slightly in 2018
- Recently, it has shown a success rate of about 80%

GITHUB URL



All Launch Site Names

GITHUB URL

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

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

Query Explaination

Using the word DISTINCT in the query means that it will only show Unique values in the Launch_Site column from SpaceX Table

There are four unique launch sites: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- 5 records of the SpaceX table were displayed using LIMIT 5 clause in the query
- Using the LIKE operator and the percent sign (%) together, the LAUNCH_SITE name starting with CAA will be called

```
%%sql
SELECT * FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

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

GITHUB URL

Total Payload Mass

- Using the SUM() function to calculate the sum of column PAYLOAD_MASS_KG
- The WHERE Clause filter the dataset to only perform calculations on CUSTOMER NASA (CRS)

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS_KG
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

total_payload_mass_kg
45596

GITHUB URL

Average Payload Mass by F9 v1.1

- Using the **AVG()** function to calculate the average value of column **PAYLOAD_MASS_KG**
- The WHERE clause filters the dataset to only perform calculation on **Booster_version = F9 v1.1**

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

avg_payload_mass_kg
2928

GITHUB URL

First Succesful Ground Landing Date

- Using the **MIN()** function to find out the earliest date in column **DATE**
- The WHERE clause filters the dataset to only perform filtration on **LANDING_OUTCOME**

```
SELECT MIN(DATE) AS FIST_SUCCESFUL_LANDING_DATE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

fist_succesful_landing_date
2015-12-22

GITHUB URL

Successful drone ship landing with payload between 4000 and 6000

- Selecting only BOOSTER_VERSION
- The WHERE clause filters the dataset to **LANDING_OUTCOME = Success (drone ship)**
- The **AND and BETWEEN clause specifies additional filter condition PAYLOAD_MASS_KG BETWEEN 4000 AND 6000**

GITHUB URL

```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)'
AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

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

Total Number of Successful and Failure Mission Outcomes

- Using the COUNT() function to filter the total number of columns
- Using GROUP BY function to group rows that have same values into summary rows to find the total number in each MISSION_OUTCOME
- SpaceX successfully completed nearly 99% of its mission based on the dataset

GITHUB URL

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

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

Boosters Carried Maximum Payload

- Using a subquery, find the maximum value of the payload using MAX() function, and then filter the dataset to perform search IF PAYLOAD_MASS_KG_ is the maximum value
- From the result, F9 B5 B10xx.x boosters carried the maximum payload

GITHUB URL

```
%%sql
SELECT DISTINCT BOOSTER_VERSION, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

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

- In the WHERE clause, filter the dataset to perform a search if Landing_Outcome is Failure (drone ship)
- Use AND operator to display a record if YEAR is 2015
- There were two landing failures on drone ships in 2015

GITHUB URL

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```

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

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

- In the WHERE clause, filter the dataset to perform search for DATE between 2010-06-04 and 2017-03-20
- Using ORDER clause to sort the records by total number of landing and DESC clause to sort the records in descending order.

GITHUB URL

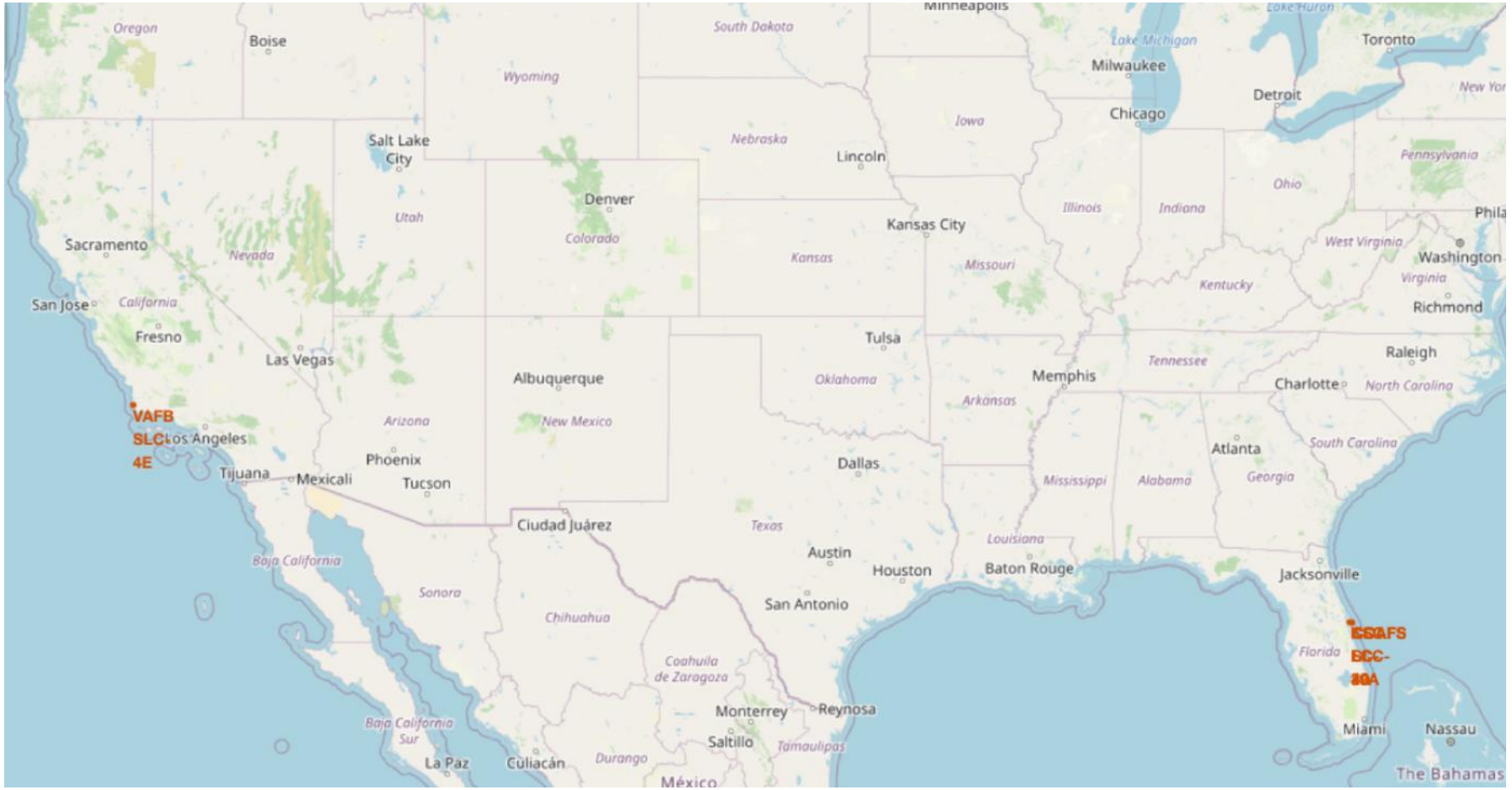
```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS total_number
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY total_number DESC
```

landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

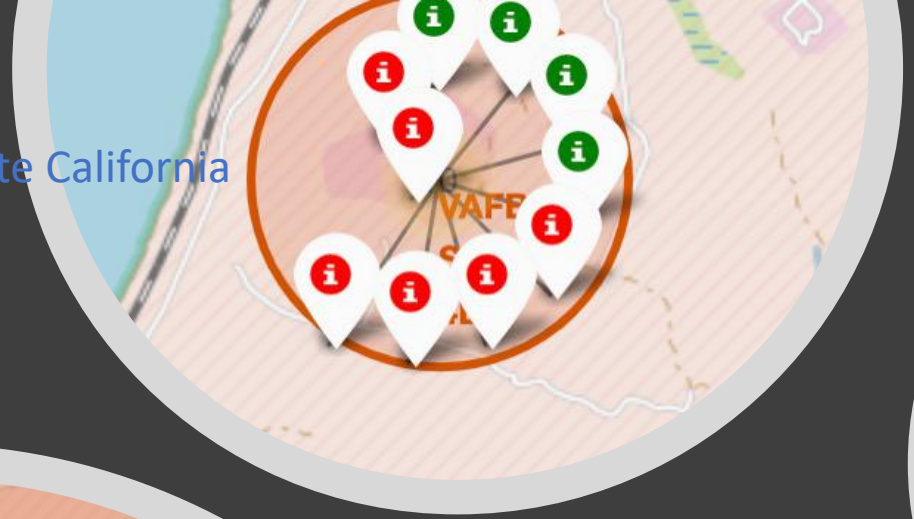
All Launch Site Location

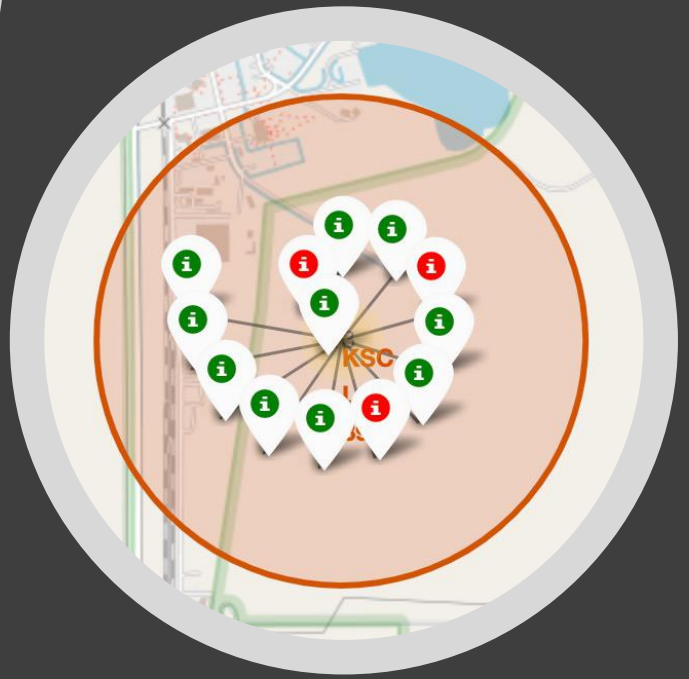


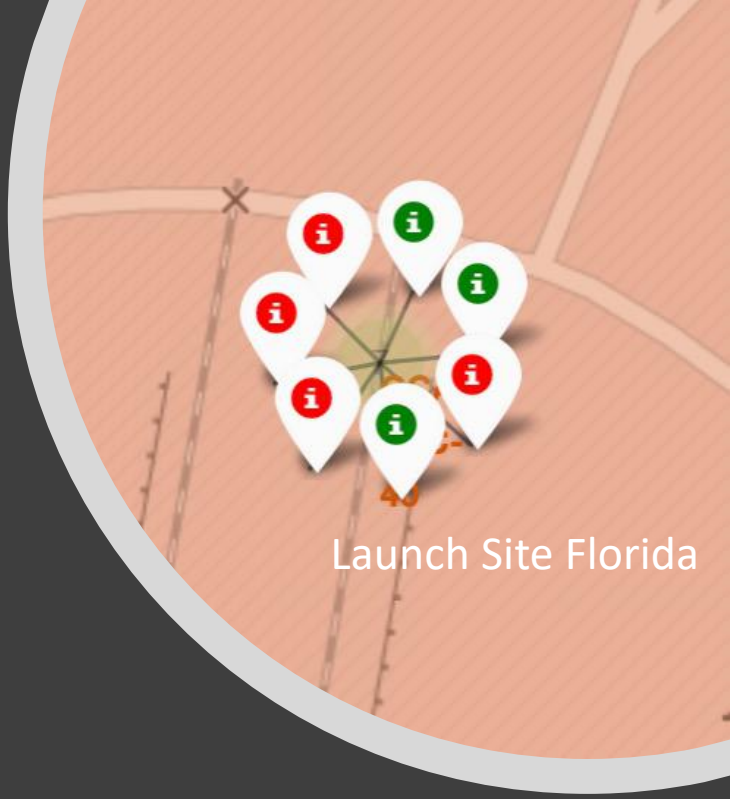
Most of the SpaceX launch site are in US coast area

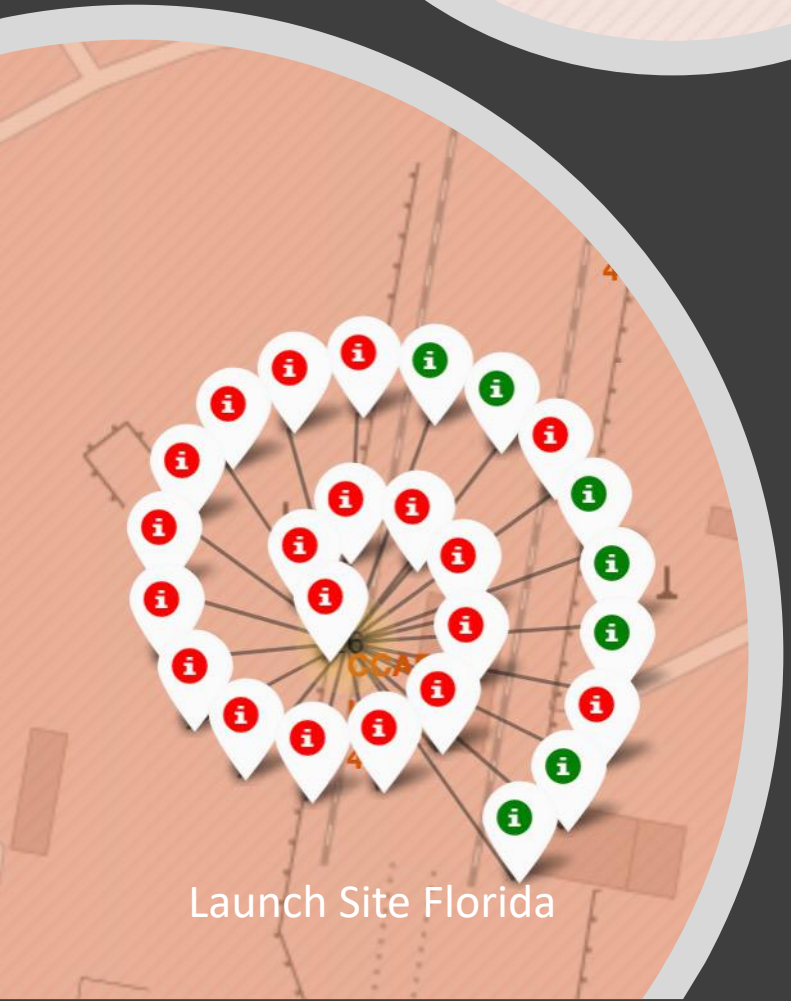
Florida and California

GITHUB URL





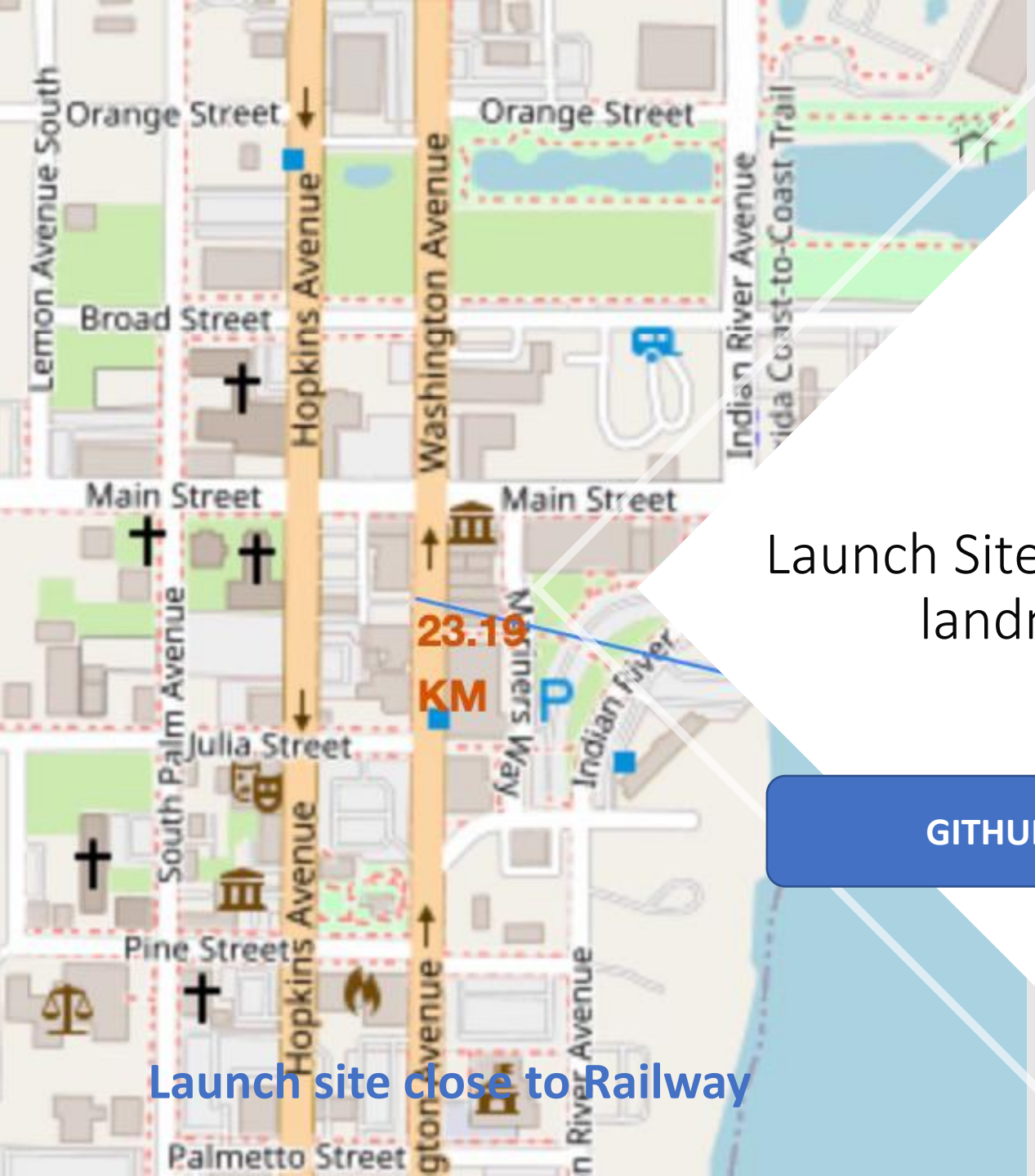




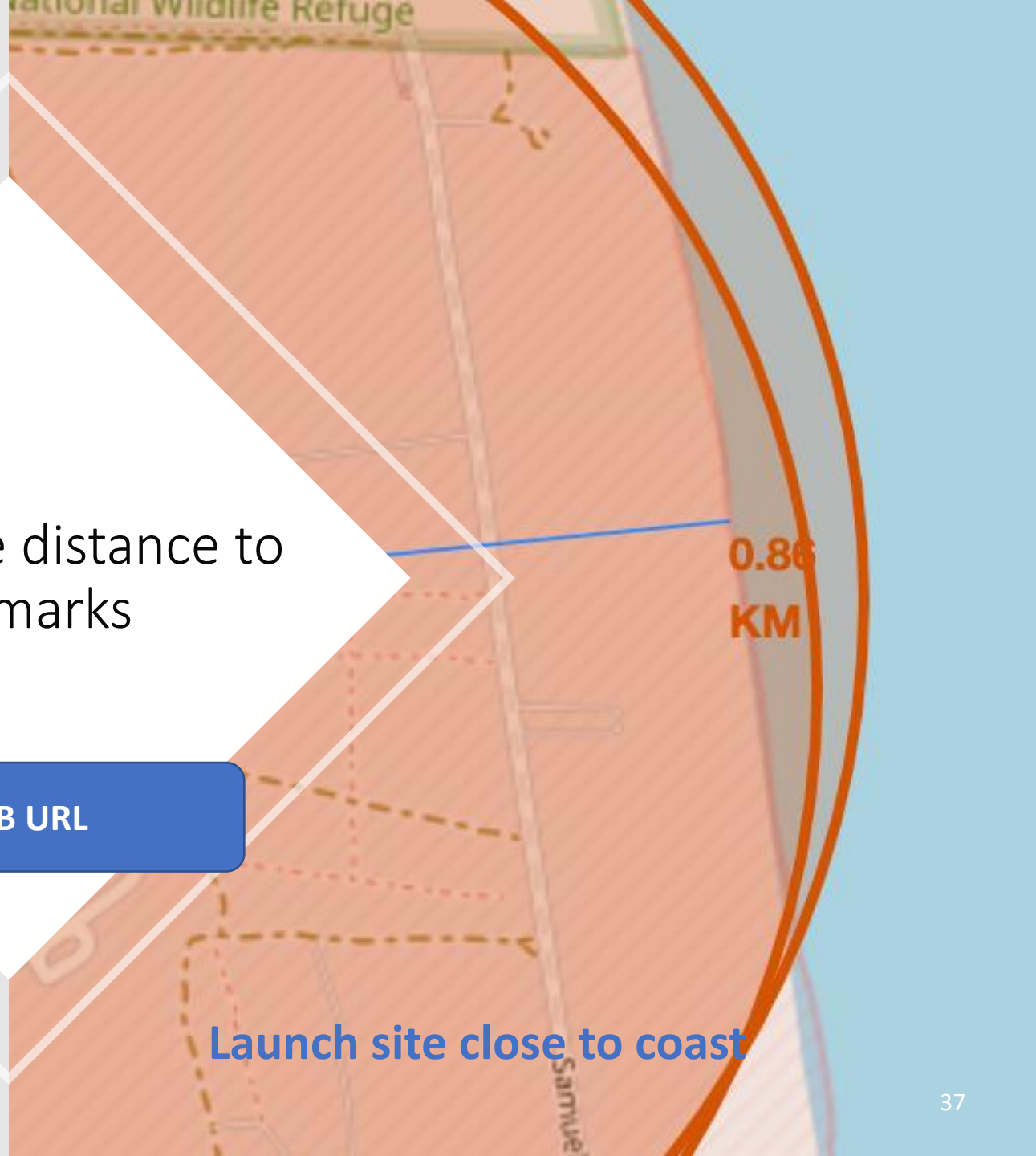
Colour Labelled Markers

GITHUB URL

Launch Site distance to landmarks

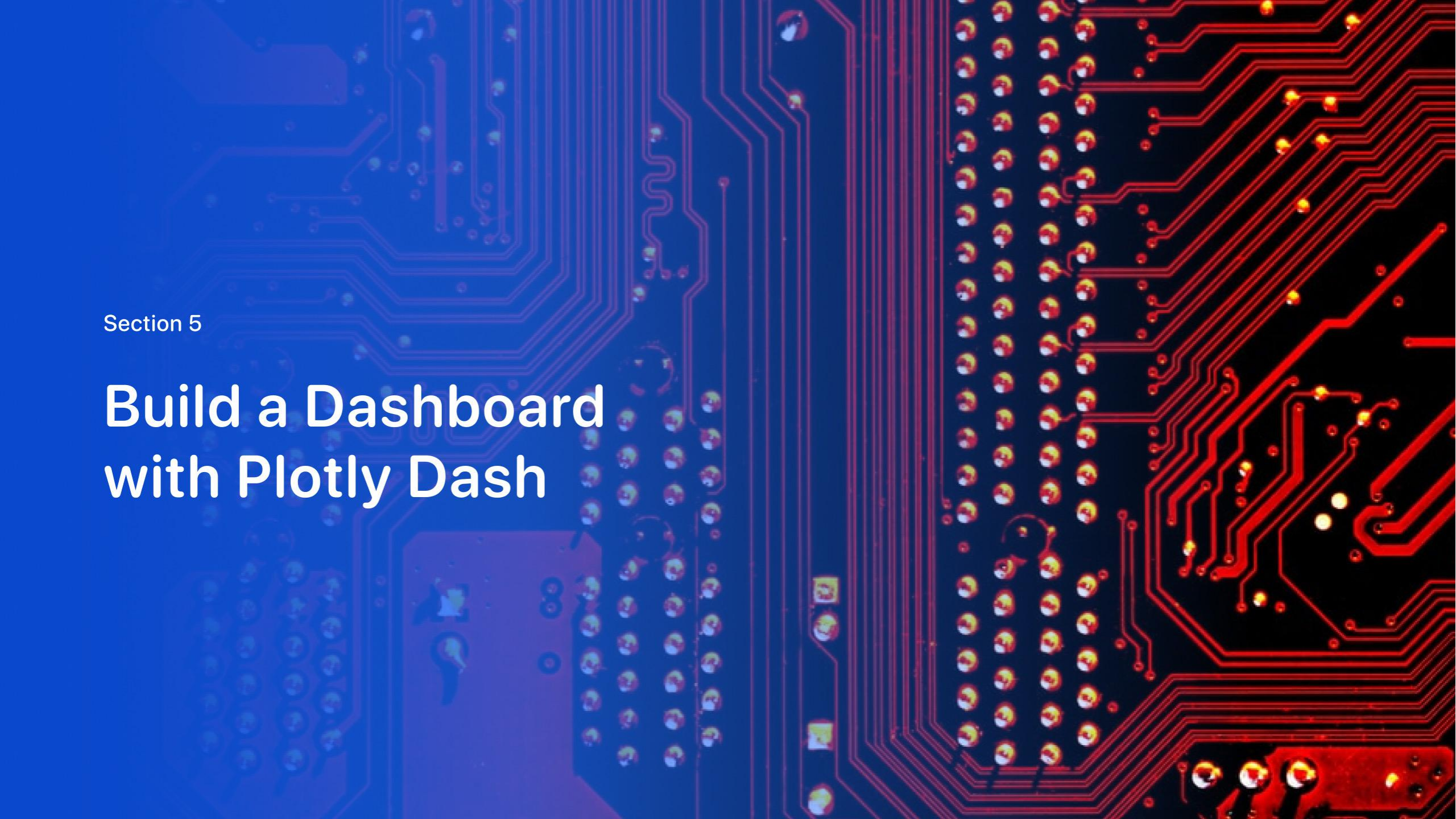


Launch site close to Railway



Launch site close to coast

GITHUB URL

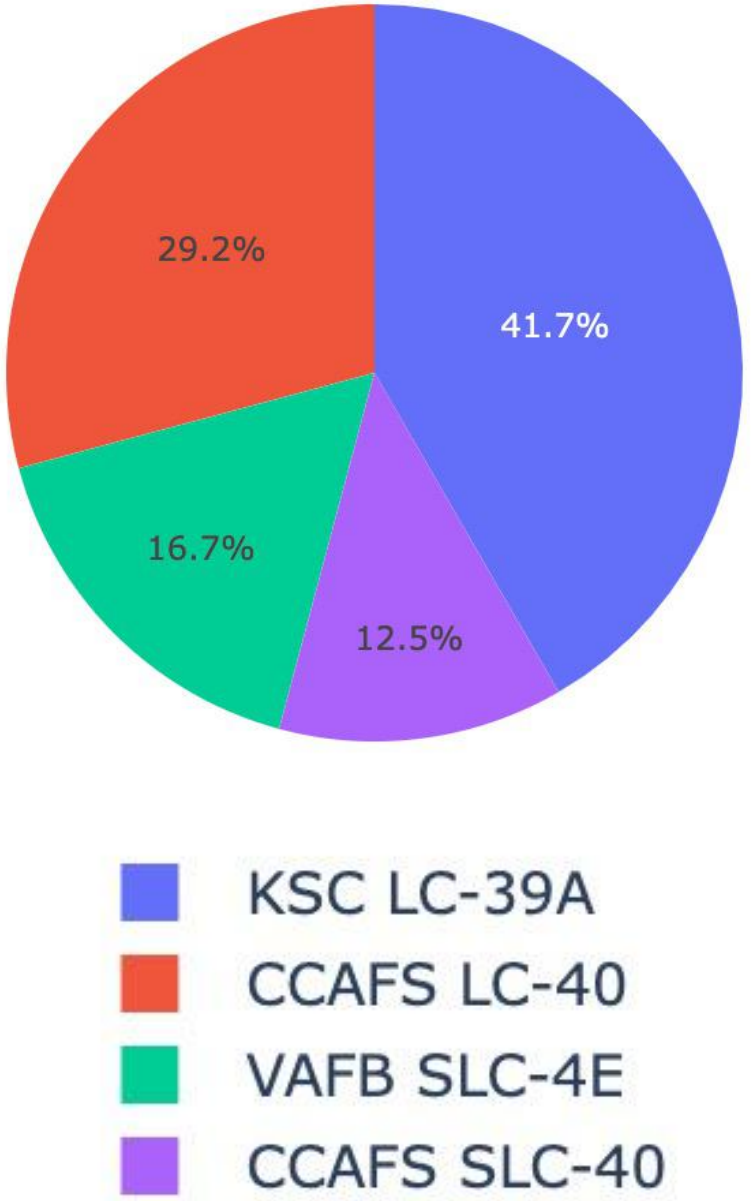


Section 5

Build a Dashboard with Plotly Dash

Total Success Launches by all Sites

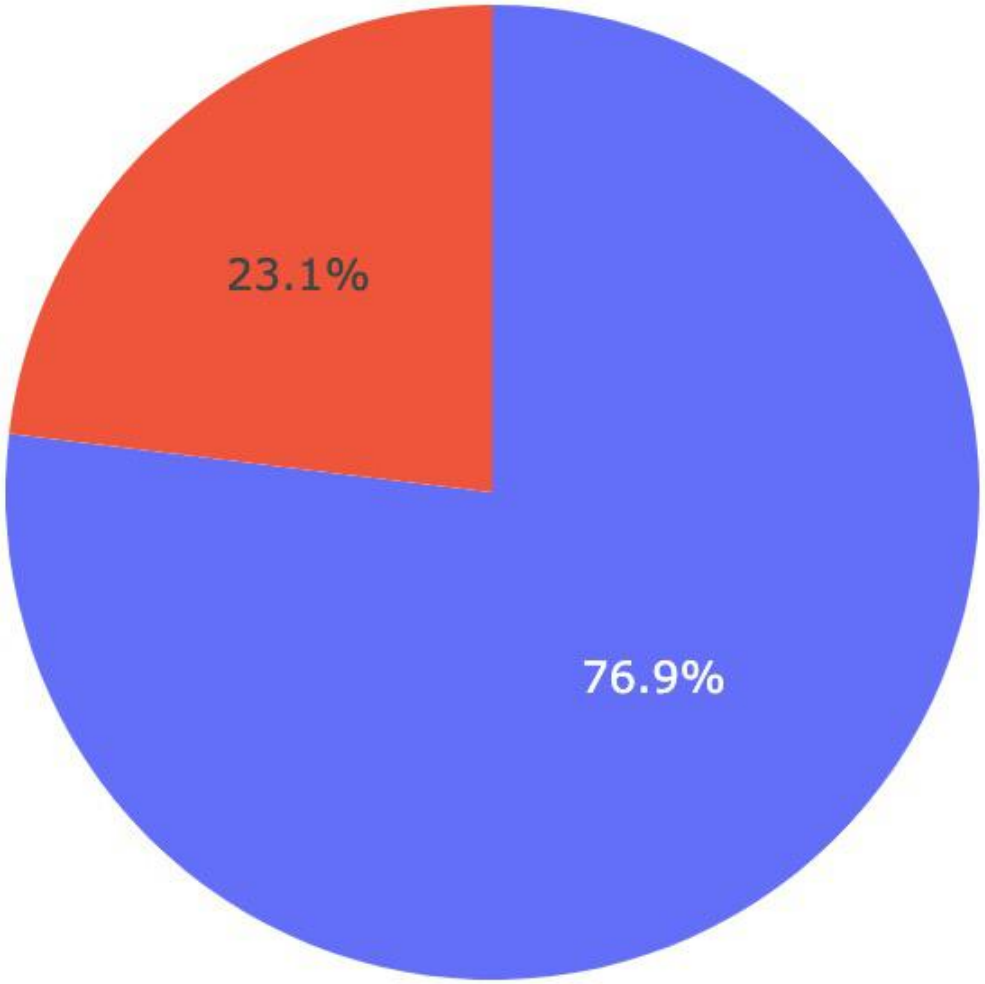
- KSLC – 39A records the most launch success among all sites.
- VAFB SLC-4E has the lowest success launch



GITHUB URL

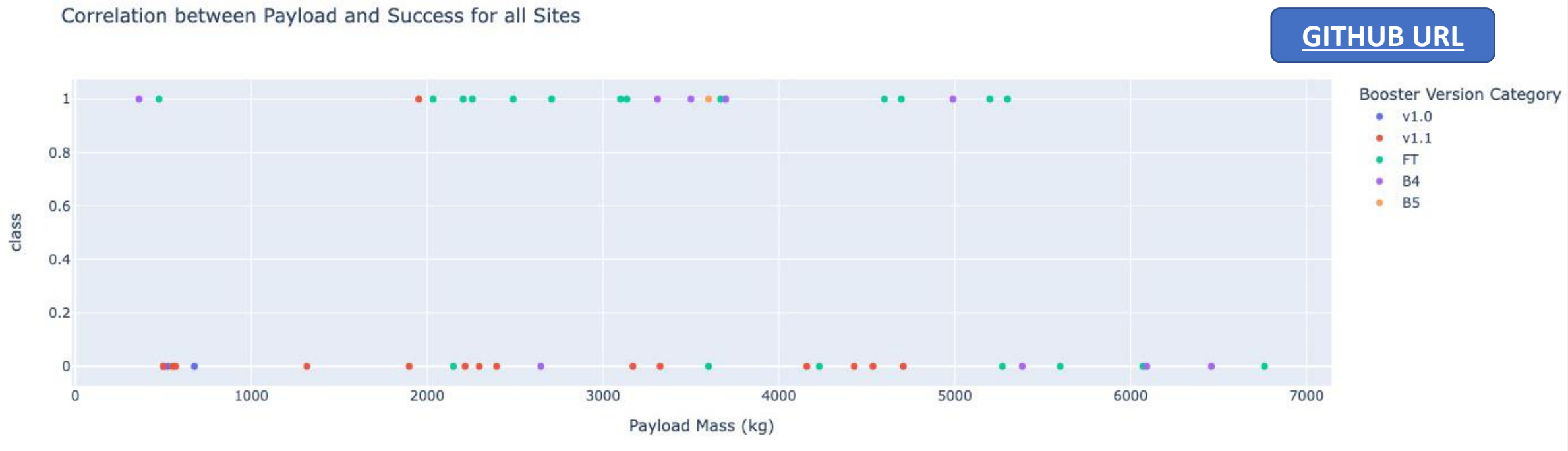
Launch Site with Highest launch Success Ratio

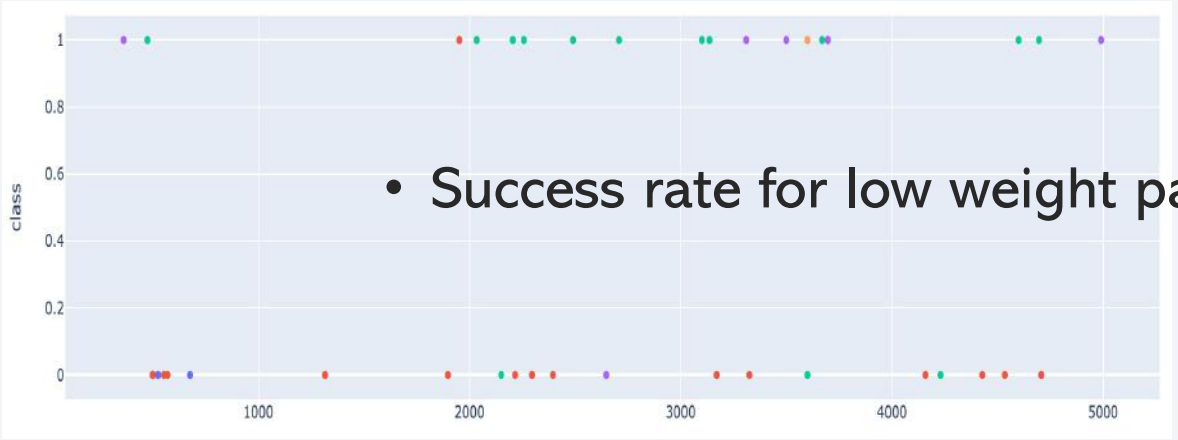
- KSC-LC-39A achieved a 76.9% success rate with total of 13 landing

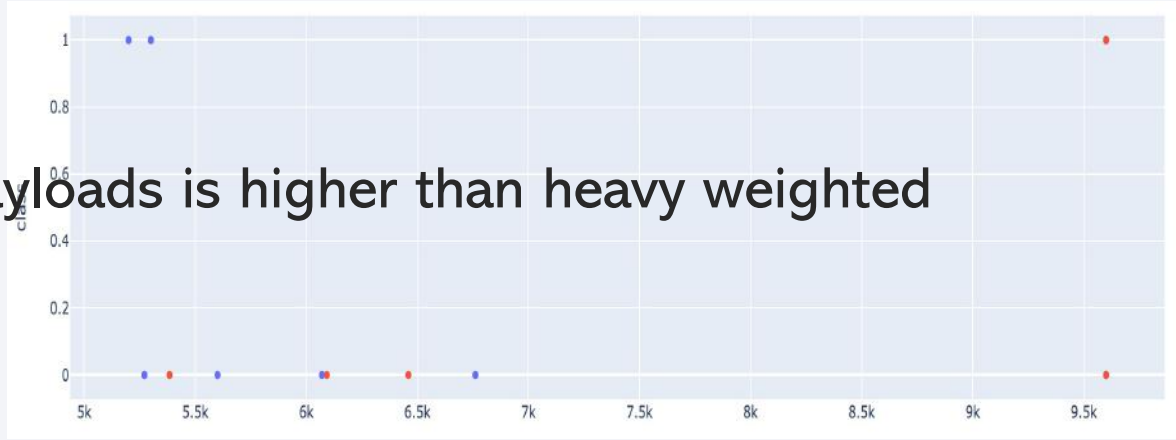


GITHUB URL

Payload VS Launch Outcome Scatter Plot for all Sites







- Success rate for low weight payloads is higher than heavy weighted

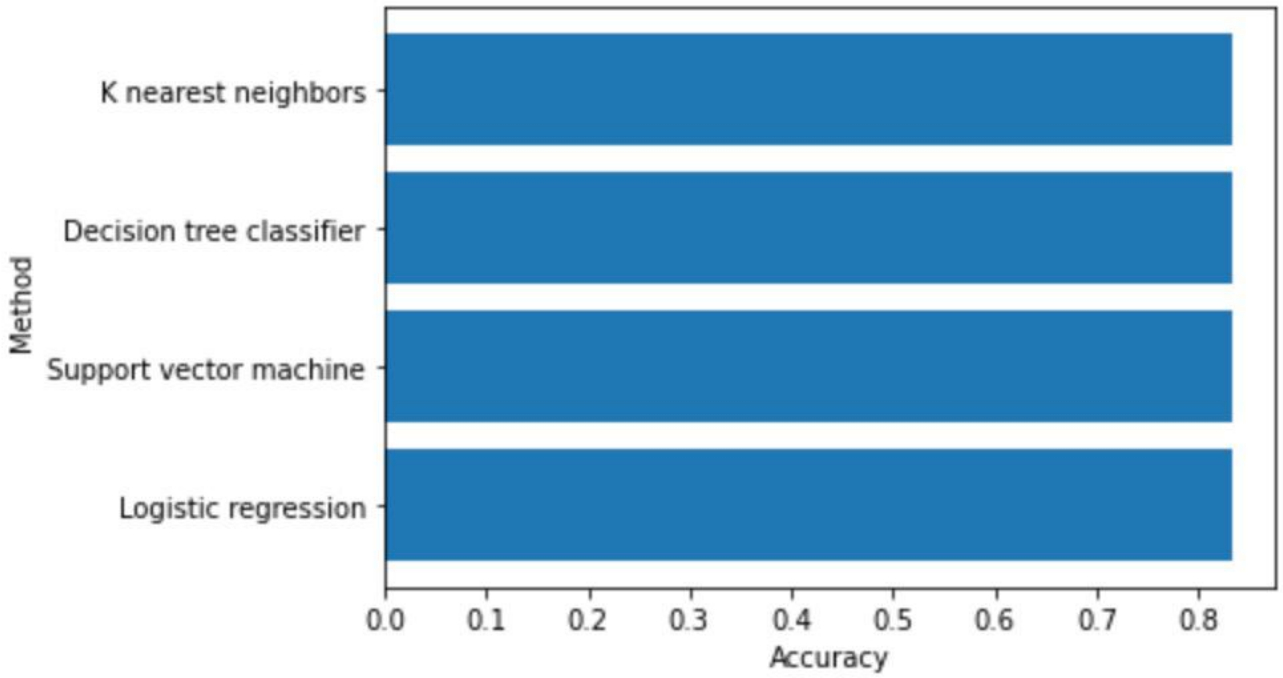
Section 6

Predictive Analysis (Classification)

Classification Accuracy

- In the test set, the accuracy of all models was virtually the same at 83.33%
- More data is needed to improve the model

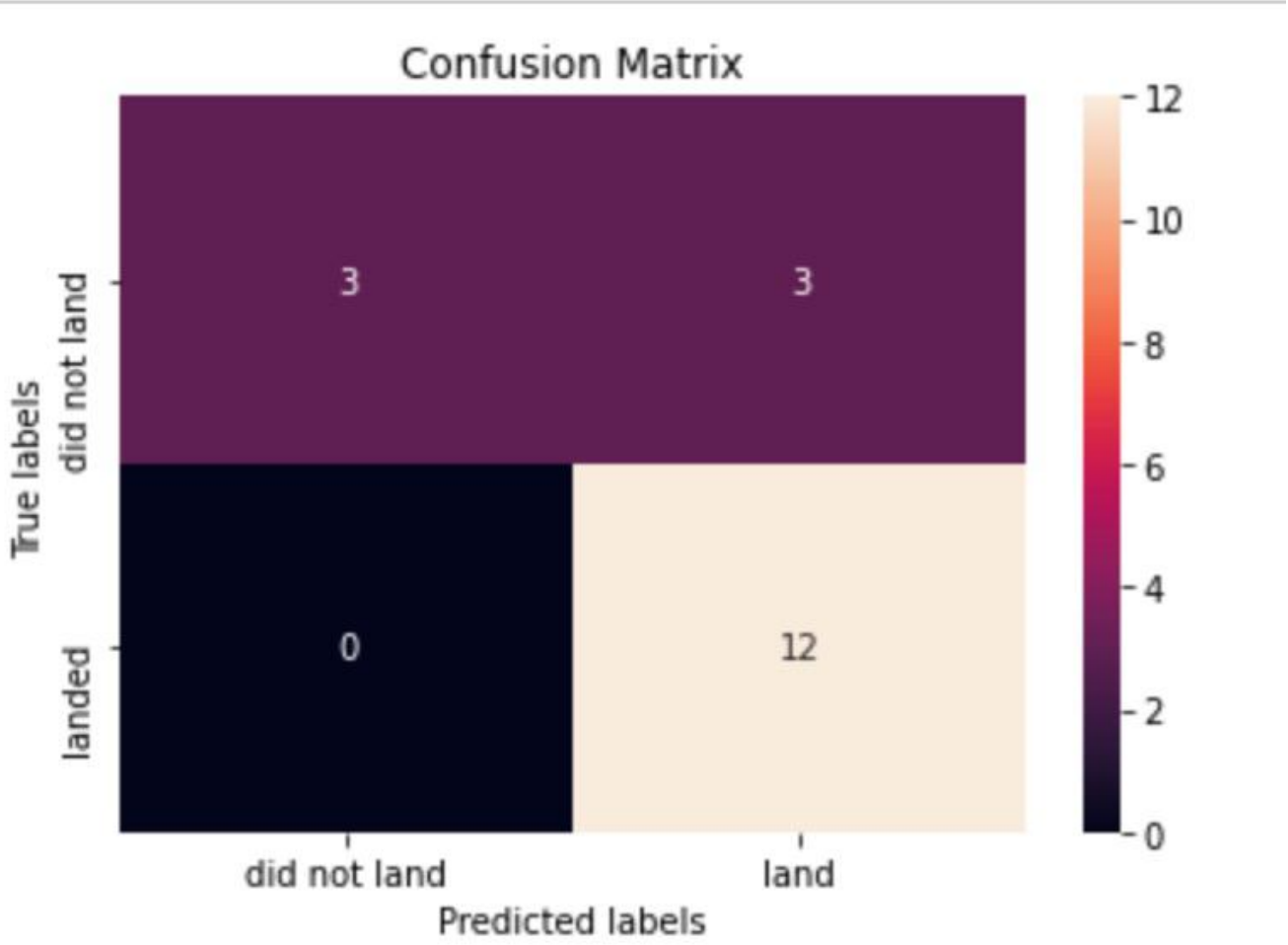
GITHUB URL



Confusion Matrix

- The confusion matrix is the same for all models
- The models predicted 12 successful landing when the true label was successful.
- The model predict 3 failed landing while the actual label was not successful.
- Overall, the model predict quite good at successful landings.

GITHUB URL



Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%).

KSLC-39A has the highest number of launch successes and the highest success rate among all sites.

Low weighted payloads perform better than the heavier payloads

In this dataset, all models have the same accuracy (83.33%), but it seems that more data is needed to determine the optimal model due to the small data size.

Conclusions

GITHUB URL

Appendix

- GitHub URL
- Coursera Applied Data Science Capstone URL

Thank you!