Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

Space X 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 Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. In this project, we want to find out this feasibility by creating a machine learning pipeline to predict if the first stage will land successfully.

- Problems we want to find answers

- What factors affect a rocket's successful landing?

- The interaction amongst various features that determine the success rate of a successful landing.

- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

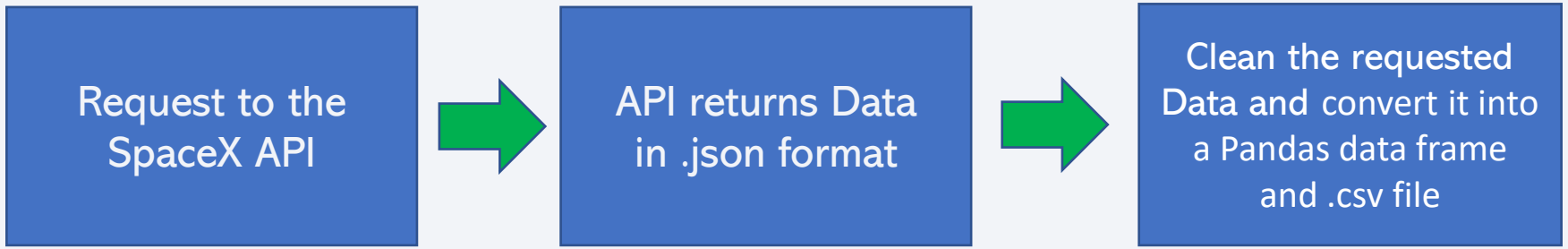
Executive Summary

- Data collection methodology:
  - The data collection was done from Wikipedia using Space X API and web scrapping by BeautifulSoup.
- Perform data wrangling
  - Determined Training Labels (landing outcomes), removed irrelevant columns and replaced missing values with NaN. One Hot Encoding was used for some columns whose values were non-numeric for Machine Learning.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

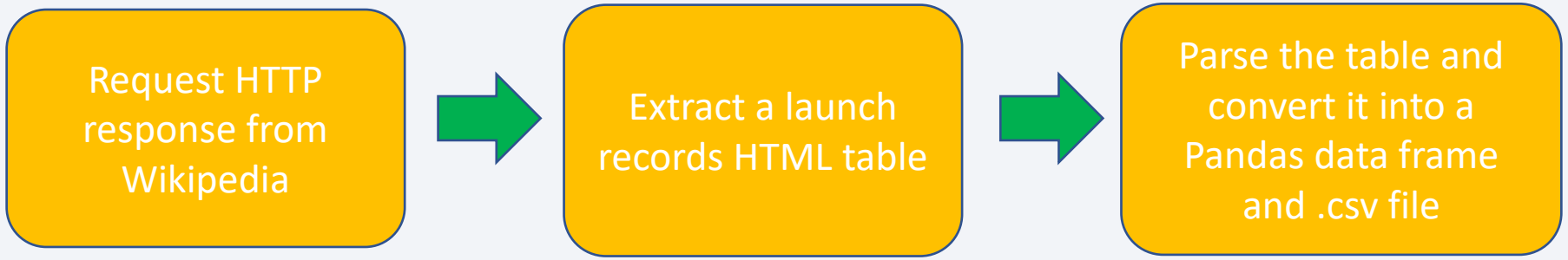
Data Collection

- Space X Launch Data was collected by the following two methods:

1. Using SpaceX REST API



2. Web scrapping using BeautifulSoup



Data Collection – SpaceX API

1. Request to the SpaceX API

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

response = requests.get(spacex_url)
```

2. API returns Data in .json format

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

3. Clean the requested Data and convert it into a Pandas data frame and .csv file

```
# Use helper functions to use the API to extract information using identification numbers in the launch data
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)

# Creating a Pandas Dataframe for the obtained data
data_falcon = pd.DataFrame(launch_dict)

# Filter the dataframe to only include Falcon 9 launches
data_falcon9 = data_falcon[data_falcon['BoosterVersion'] != 'Falcon 1']

# Dealing with missing values
mean_plm = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean_plm)

# Export the data into a CSV
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

The GitHub URL of the completed SpaceX API calls notebook: https://github.com/bistra3759/data-science-capstone/blob/2e4727a88bf3953816a7ebca14276146b99c375a/Applied%20Data%20Science%20Capstone_1.ipynb

Data Collection - Scraping

1. Request HTTP response from Wikipedia using BeautifulSoup

2. Extract a launch records HTML table

3. Parse the table and convert it into a Pandas data frame and .csv file

The GitHub URL of the completed SpaceX scrapping calls notebook: https://github.com/bistra3759/data-science-capstone/blob/master/Applied%20Data%20Science%20Capstone_2.ipynb

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(response.content, 'html.parser')
```

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

```
column_names = []

# Apply find_all() function with 'th' element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name ('if name is not None and len(name) > 0') into a list called column_names

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

```
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.'] = []
```

```
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:
```

```
df= pd.DataFrame({key:pd.Series(value) for key, value in launch_dict.items() })
```

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

Data Wrangling

Background

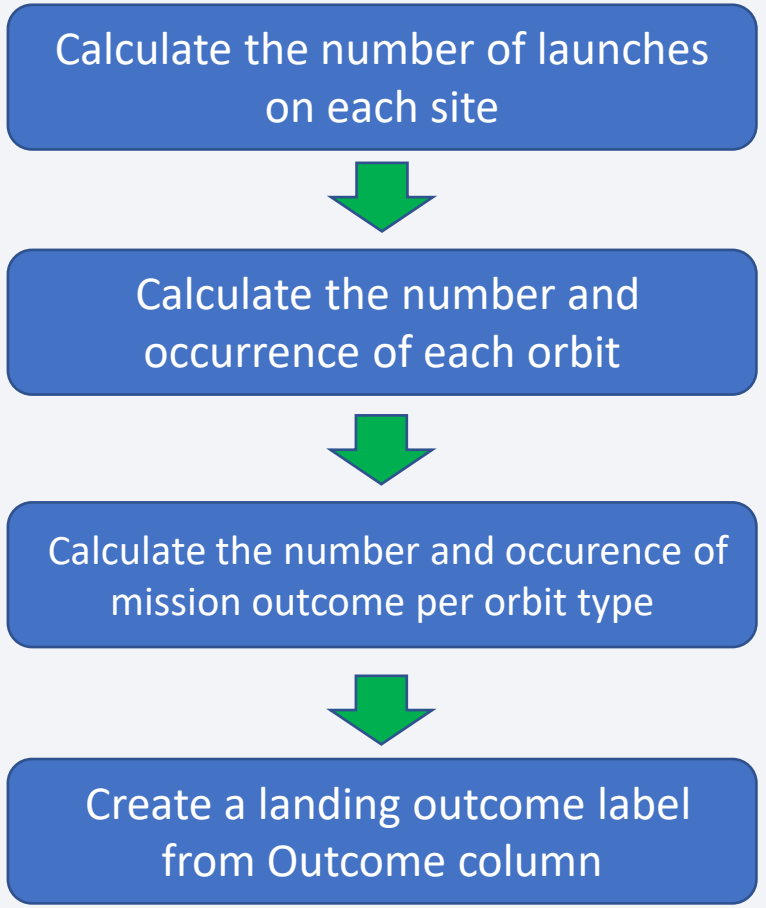
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident. For example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

We mainly convert those outcomes into Training Labels with 1 (booster successfully landed), and 0 (booster landing was unsuccessful).

Objectives

Exploratory Data Analysis

Determine Training Labels

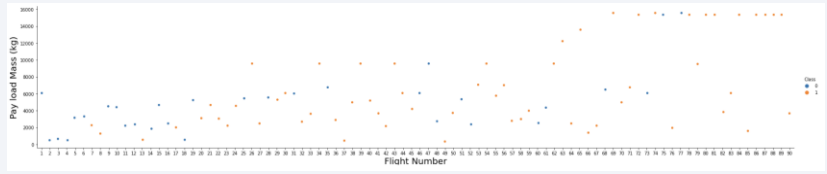


- The GitHub URL of the completed data wrangling related notebook: https://github.com/bistra3759/data-science-capstone/blob/master/Applied%20Data%20Science%20Capstone_3.ipynb

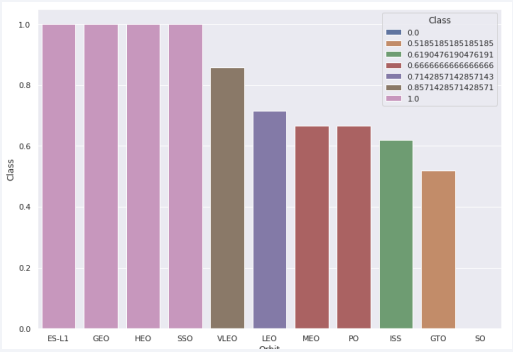
EDA with Data Visualization

Scatter Charts plotted to visualize the relationship between:

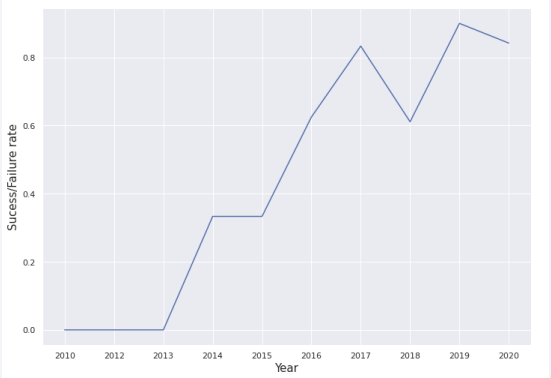
1. **Flight Number** & **Launch Site**
2. **Payload** & **Launch Site**
3. **Flight Number** & **Orbit type**
4. **Payload** & **Orbit type**



Bar Graph plotted for Success rate of each orbit type



Line Graph plotted to visualize the launch success yearly trend



- The GitHub URL of your completed EDA with data visualization notebook: https://github.com/bistra3759/data-science-capstone/blob/master/Applied%20Data%20Science%20Capstone_5.ipynb

EDA with SQL

We performed SQL queries on SpaceX Dataset to get a useful information such as:

- The names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Date where the first successful landing outcome in ground pad was achieved.•
- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- The total number of successful and failure mission outcomes
- The names of the booster versions which have carried the maximum payload mass.
- 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

The GitHub URL of the completed EDA with SQL notebook: https://github.com/bistra3759/data-science-capstone/blob/master/Applied%20Data%20Science%20Capstone_4.ipynb

Build an Interactive Map with Folium

- Launch success rate may depend on the location and proximities of a launch site.
- We used Folium Map to visualize the location of each launch site with markers, circles to display the name of the site and launch success/failure, and lines for displaying the distances to nearest coastline, railway, highway, and cities.
- Calculations of distances using coordinates are done by Haversine Fomula:
- We can answer the following questions from these findings:
  - Are all launch sites in proximity to the Equator line?
  - Are all launch sites in very close proximity to the coast?
  - Are launch sites in close proximity to railways?
  - Are launch sites in close proximity to highways?
  - Are launch sites in close proximity to coastline?
  - Do launch sites keep certain distance away from cities?

The GitHub URL of the completed interactive map with Folium map: https://github.com/bistra3759/data-science-capstone/blob/master/Applied%20Data%20Science%20Capstone_6.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash for a user to perform interactive visual analytics on SpaceX launch data in real-time

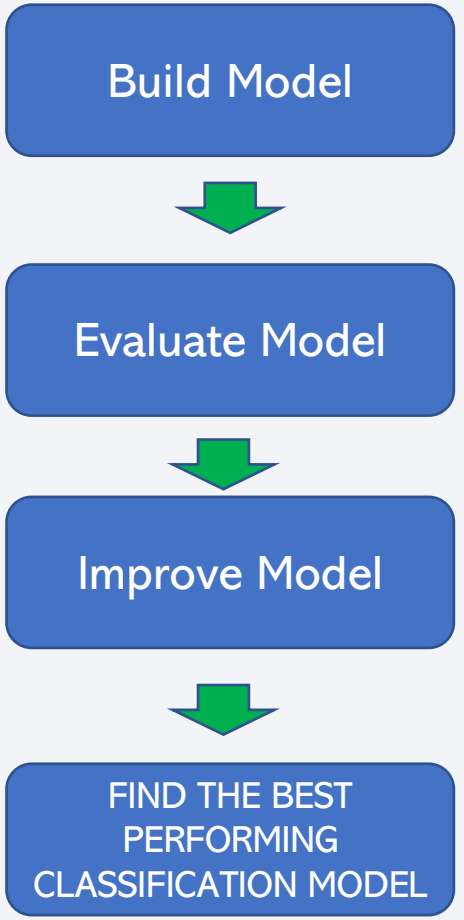
- Pie chart is added to the dashboard for visualizing launch success counts for a user selected launch site

- Scatter Graph is added to the dashboard for visualizing launch success counts depending on various Payload Mass for different Booster Version Categories

The GitHub URL of the completed Plotly Dash lab: https://github.com/bistra3759/data-science-capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- Build Model
  - Load the SpaceX dataset into Numpy and Pandas DataFrame
  - Standardize the data and split it into training and testing data
  - Choose from Machine Learning Algorithms and set the parameters using GridSearchCV
  - Fit the dataset into GridSearchCV objects to find the best parameters
- Evaluate Model
  - Calculate the accuracy on the test data
  - Plot and Examine the confusion matrix
- Improve Model
  - Perform Feature Engineering
  - Perform Algorithm tuning
- Find the best performing Classification Model
  - Compare the accuracy for each algorithm



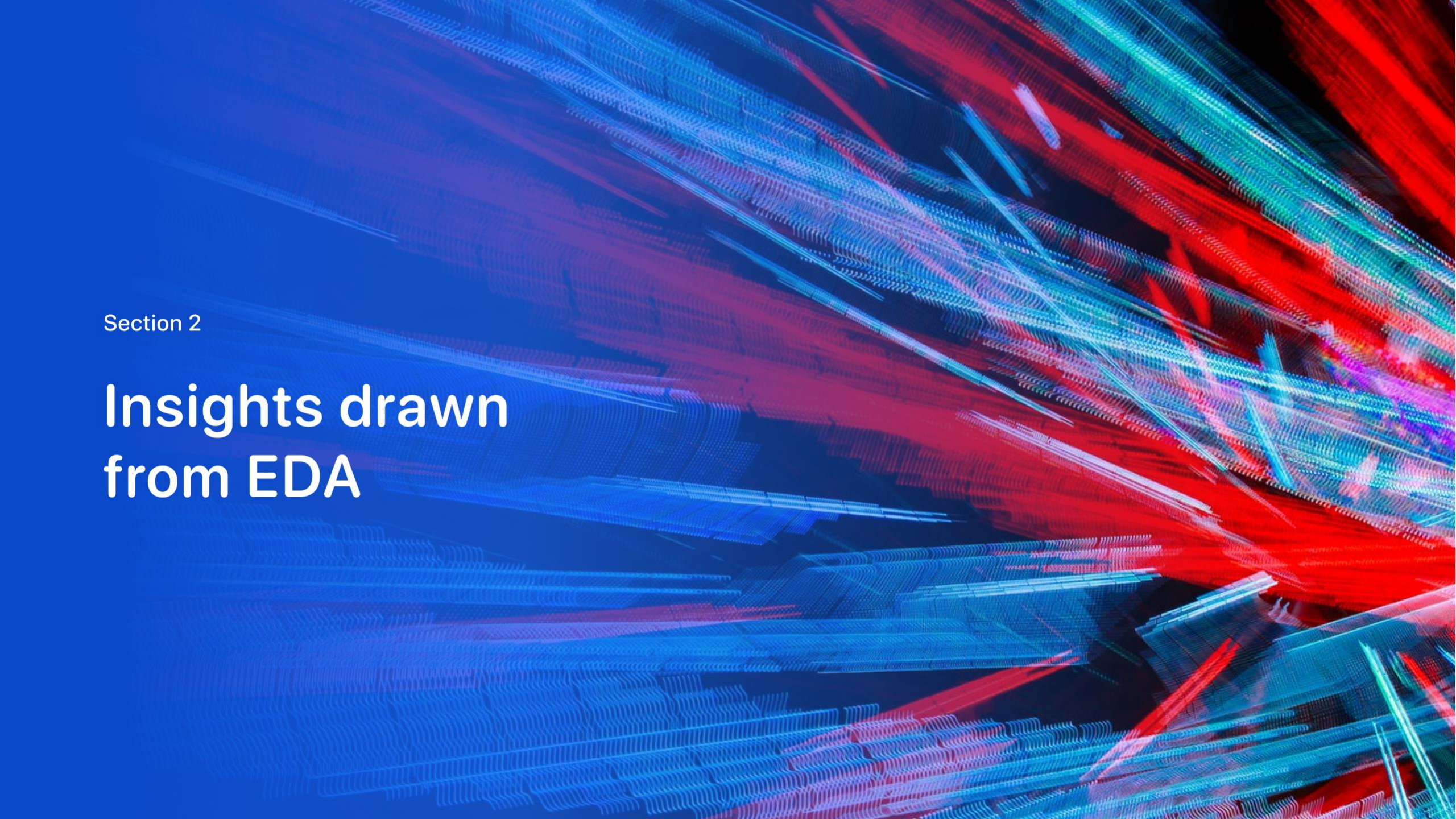
- The GitHub URL of the completed Predictive Analysis: https://github.com/bistra3759/data-science-capstone/blob/master/Applied%20Data%20Science%20Capstone_7.ipynb

Results

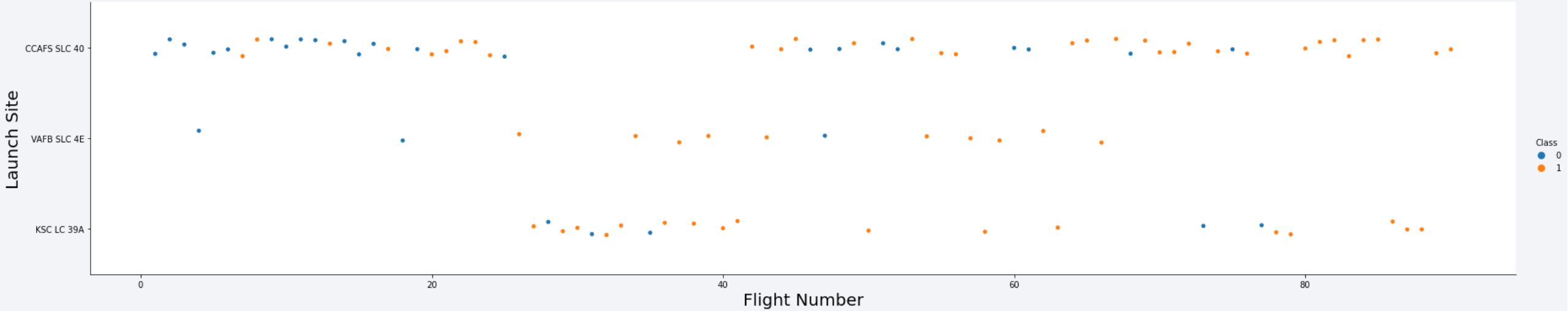
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

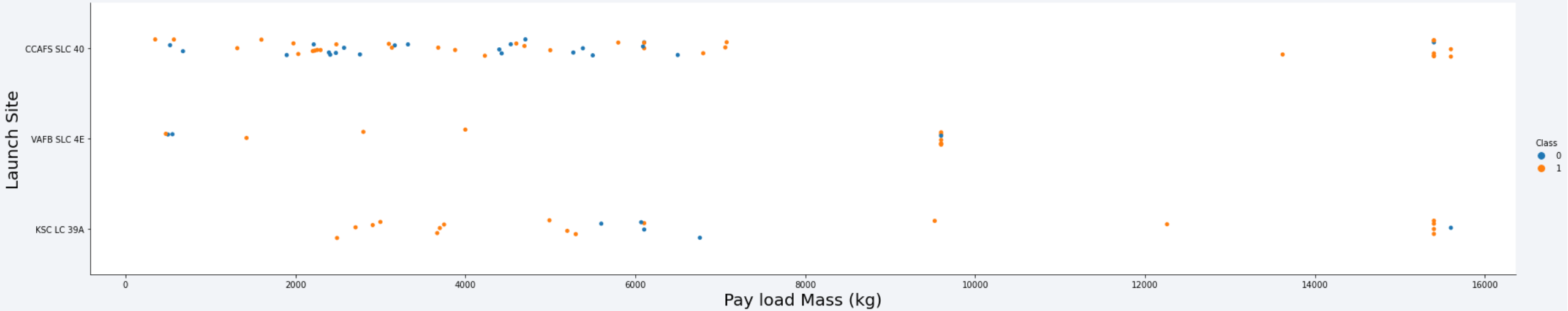


Flight Number vs. Launch Site



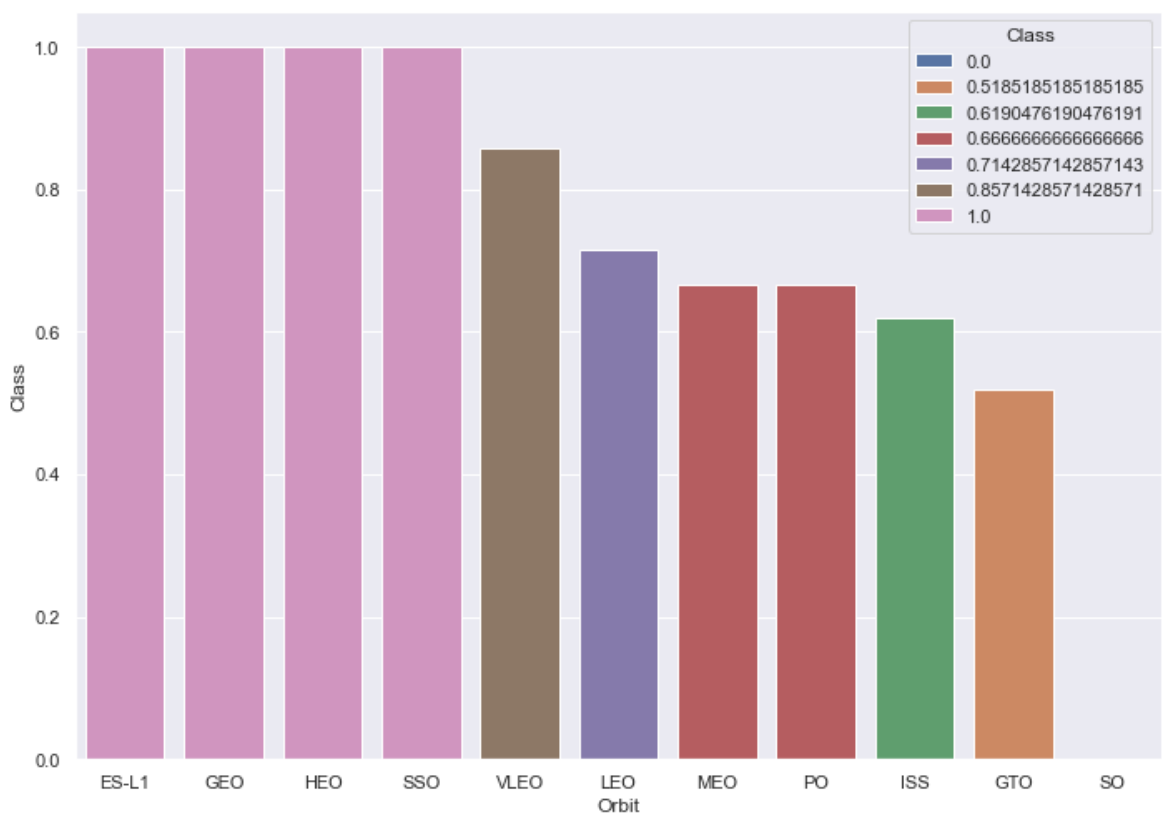
We can see the larger the flight numbers, the more successful launch rate at a launch site with the number of orange dots (Class 1: success) increase in proportion to Flight Number at each Launch Site.

Payload vs. Launch Site



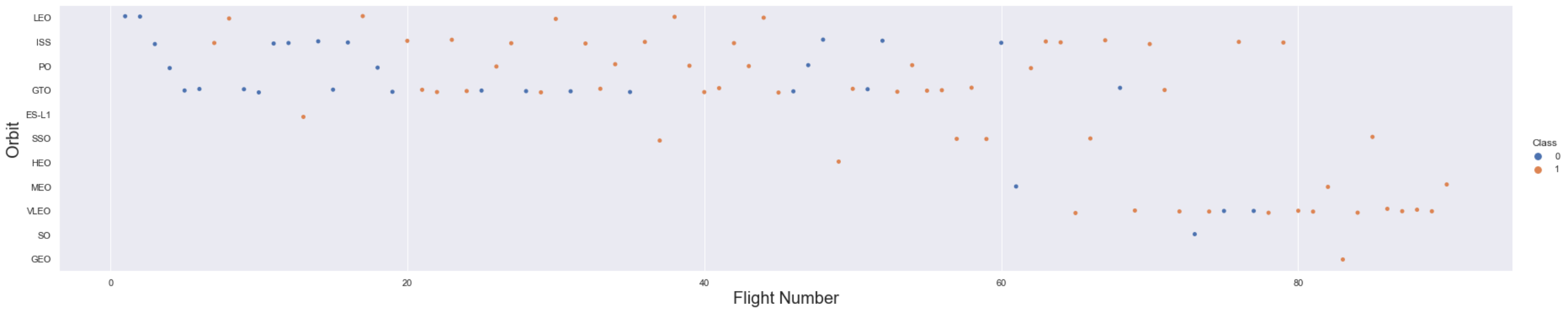
The greater the payload mass, the higher the launch success rate for CCAFS SLC 40, but no clear relationship can be found for the other two. Overall, we can't find any meaningful pattern to make a decision based on this payload vs Launch Site visualization.

Success Rate vs. Orbit Type



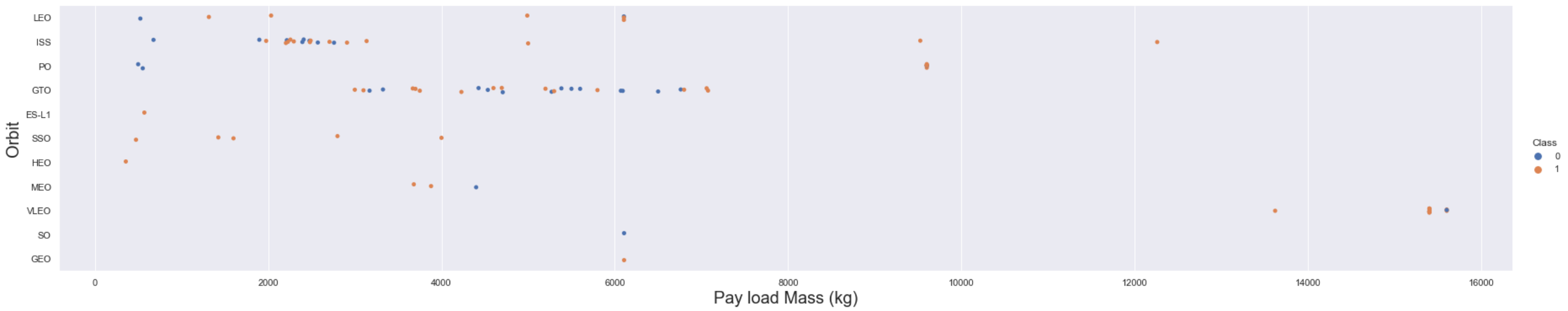
We can see Orbit ES-L1, GEO, HEO, SSO have the best Success Rate of 100%.

Flight Number vs. Orbit Type



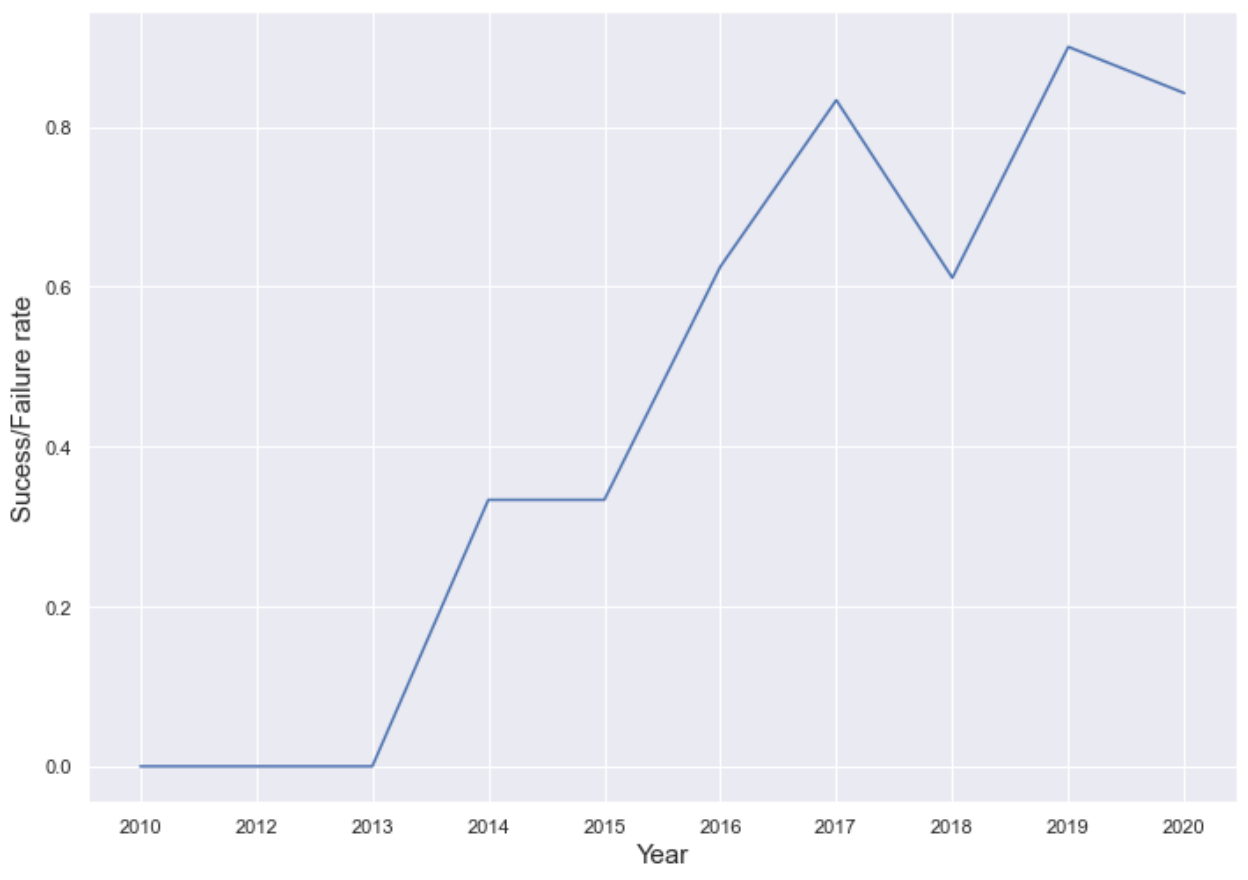
LEO orbit's success appears to relate to the number of flights, but there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



For Polar, LEO and ISS, heavier payloads seem to relate to more successful landing rate. But for GTO, we can't distinguish this well because both successful landing rate and unsuccessful landing exist there.

Launch Success Yearly Trend



We can observe that the success rate since 2013 kept increasing till 2020.

All Launch Site Names

We used SQL query keyword **DISTINCT** to find the following four unique Launch Site names:

1. CCAFS LC-40
2. CCAFS SLC-40
3. KSC LC-39A
4. VAFB SLC-4E

```
In [5]: %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

Out [5]:

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

Launch Site Names Begin with 'CCA'

```
In [6]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

Out [6]:

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

We used SQL query using the keywords **LIKE** to find the Launch Site name begins with ‘CCA’ and **LIMIT** to show only 5 records.

Total Payload Mass

In [74]:
```
%%sql
SELECT SUM(payload_mass__kg_) AS Total_Payload_Mass
FROM SPACEXTBL
WHERE customer LIKE 'NASA (CRS)'
```

Out[74]:

total_payload_mass
45596

We used SQL query keyword SUM and LIKE to produce Total Payload Mass (**45596 kg**) carried by boosters launched by NASA (CRS).

Average Payload Mass by F9 v1.1

In [76]:
```
%%sql
SELECT AVG(payload_mass__kg_)
FROM SPACEXTBL WHERE booster_version = 'F9 v1.1'
```

Out[76]:

1
2928

We used SQL query keyword AVG to find the average payload mass (**2928 kg**) carried by booster version F9 v1.1.

First Successful Ground Landing Date

In [77]:
```
%%sql
SELECT MIN(DATE) FROM SPACEXTBL
WHERE (mission_outcome = 'Success') & (landing__outcome LIKE '%ground pad%')
```

Out[77]:

1
2015-12-22

We used SQL query keyword MIN fo find the date of the first successful landing outcome on ground pad (**Dec 22nd, 2015**).

Successful Drone Ship Landing with Payload between 4000 and 6000

In [82]:
```
%%sql
SELECT booster_version FROM SPACEXTBL
WHERE (landing__outcome = 'Success (drone ship)') & (payload_mass__kg_ BETWEEN 4000 AND 6000)
```

Out [82]:

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

We used filer conditions with the keyword BETWEEN to find the names of boosters which have successfully landed on drone ship and had payload between 4000 kg and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

In [13]:
```
%%sql
SELECT mission_outcome, count(*) AS COUNT
FROM SPACEXTBL GROUP BY mission_outcome
```

Out [13]:

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

We used COUNT and GROUP BY functions to calculate the total number of successful and failure mission outcomes, along with one case of unclear success (payload status unclear).

Boosters Carried Maximum Payload

In [83]:
```
%%sql
SELECT booster_version, payload_mass__kg_ AS Maximum_Payload_Mass
FROM SPACEXTBL
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
```

Out [83]:

booster_version	maximum_payload_mass
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

We used a subquery using another SELECT statement to list the names of the booster_versions which have carried the maximum payload mass.

2015 Launch Records

In [36]:
```
%%sql
SELECT landing__outcome, booster_version, launch_site
FROM SPACEXTBL
WHERE EXTRACT(YEAR FROM DATE) = 2015 AND landing__outcome LIKE '%Failure (drone ship)%'
```

Out [36]:

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

We used Extract function and LIKE keyword to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.

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

In [84]:
```
%%sql
SELECT landing__outcome, COUNT(landing__outcome) AS COUNT FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY COUNT(landing__outcome) DESC
```

Out [84]:

landing__outcome	COUNT
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

We used GROUP BY and ORDER BY keyword to rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

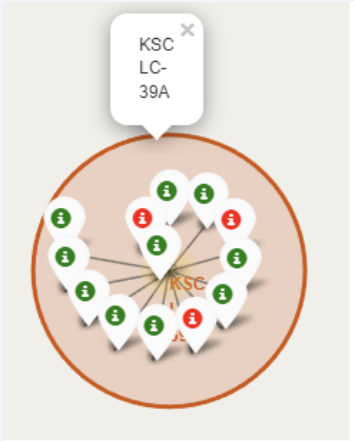
Section 4

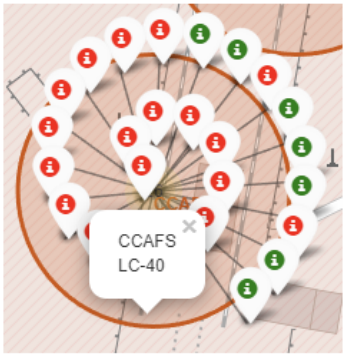
Launch Sites Proximities Analysis

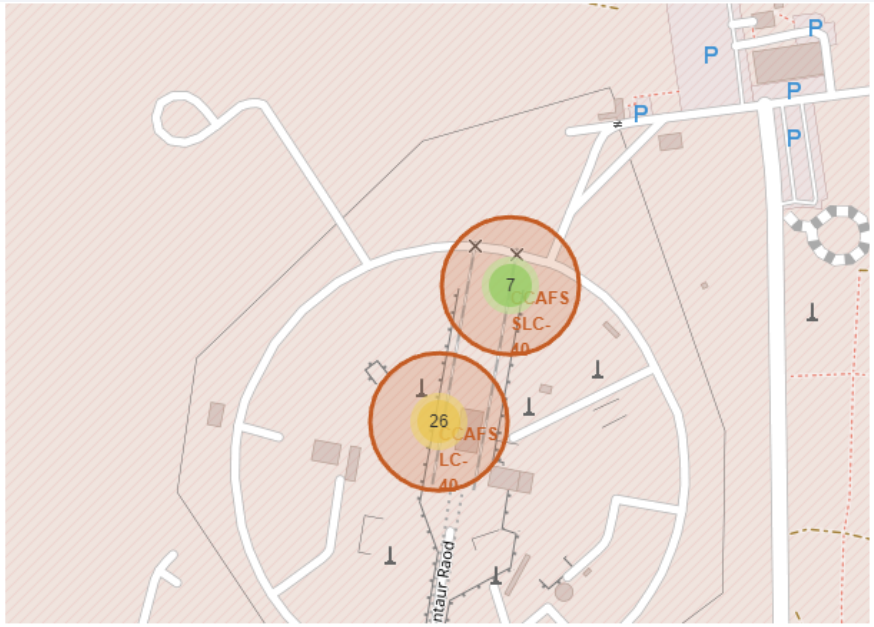
All Launch Sites' Global Map Markers

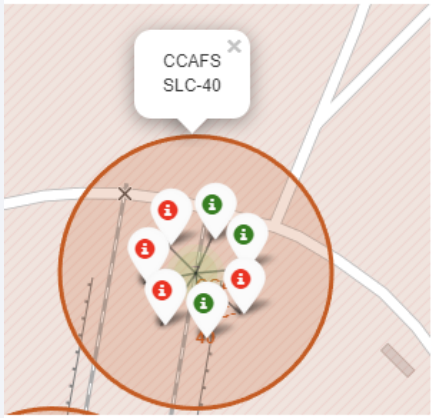
We can see the SpaceX Launch Sites are located in the two regions (California and Florida) of the United States.

Launch Sites with Markers





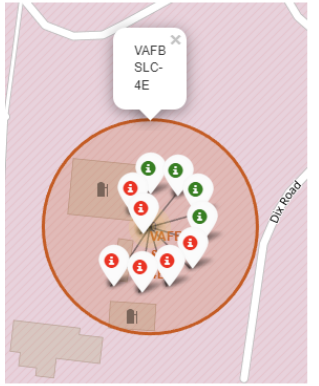




Launch Sites in Florida

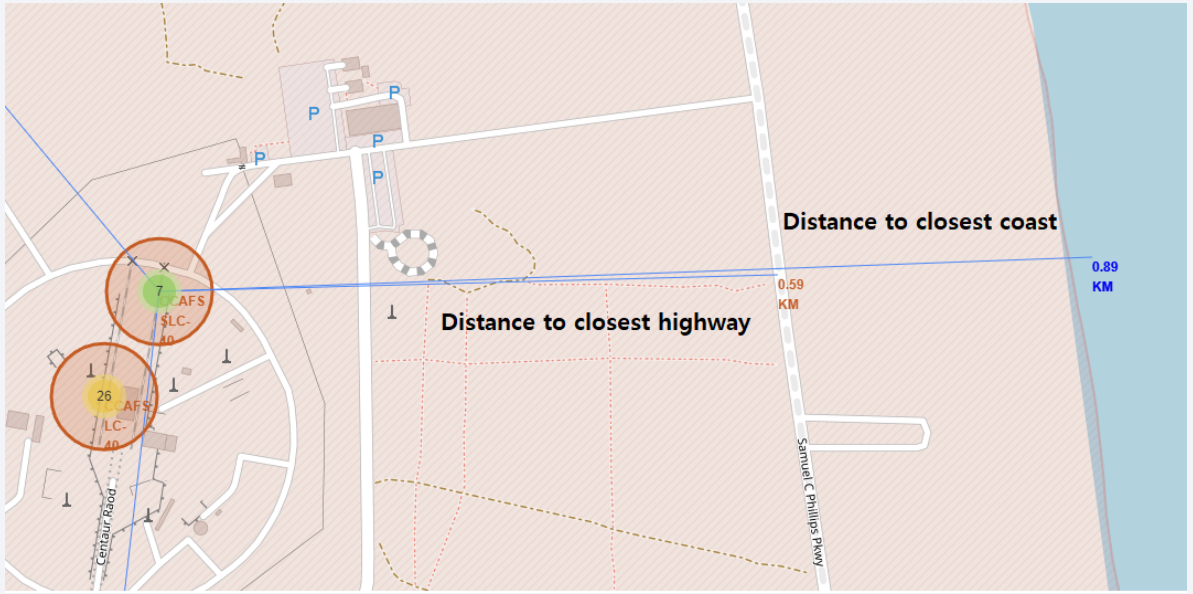
Green Markers: Successful Launches

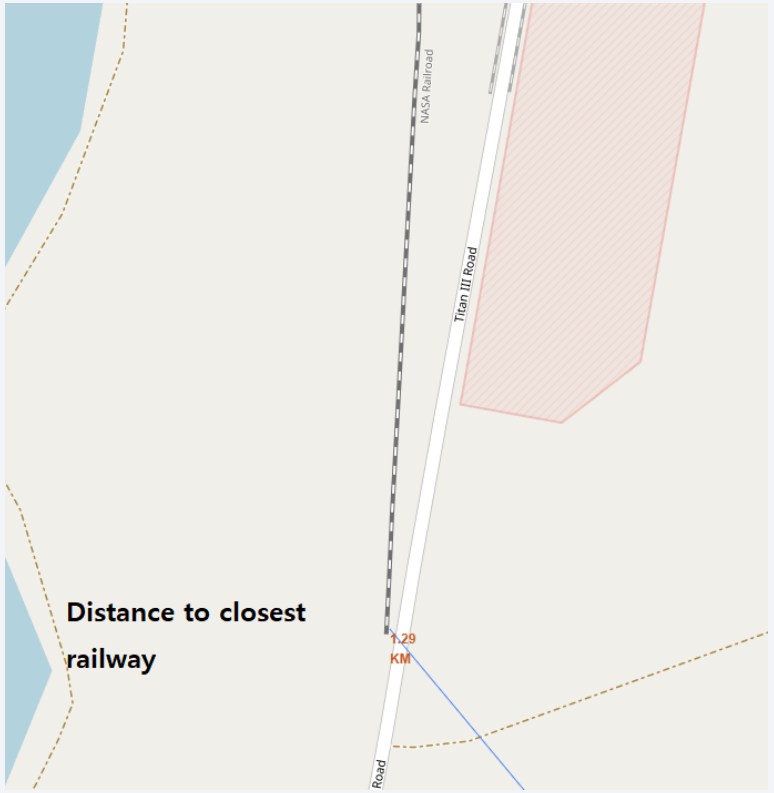
Red Markers: Unsuccessful Launches



The Launch Site in California

Distances Between a Launch Site to its Proximities

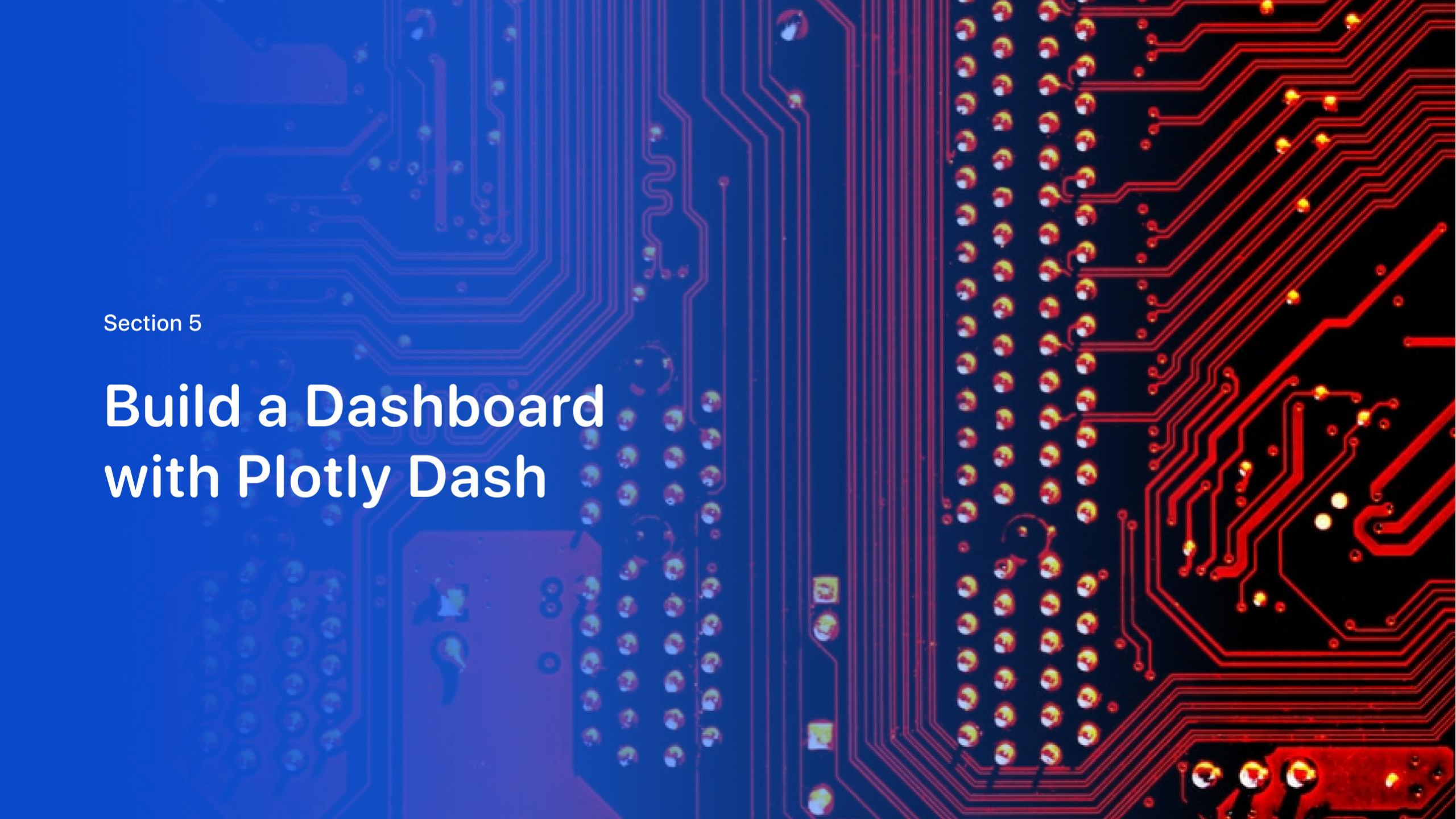




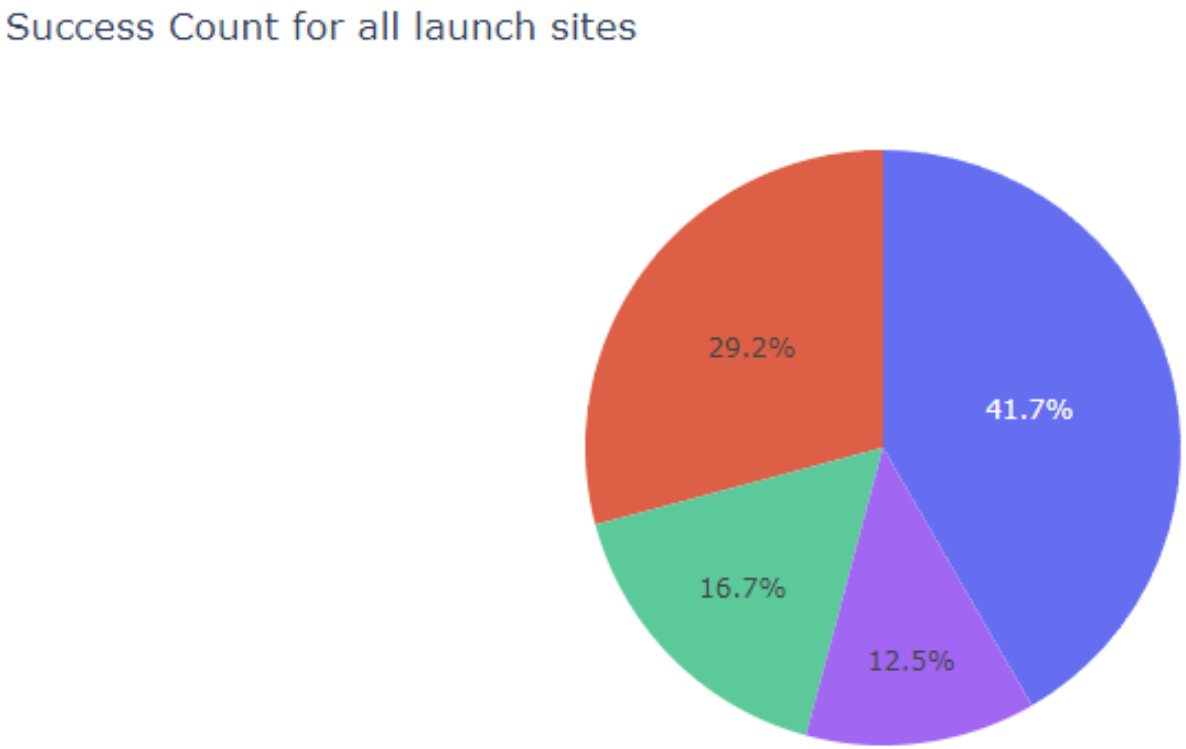
Are launch sites in close proximity to railways? Yes
Are launch sites in close proximity to highways? Yes
Are launch sites in close proximity to coastline? Yes
Do launch sites keep certain distance away from cities? No

Section 5

Build a Dashboard with Plotly Dash

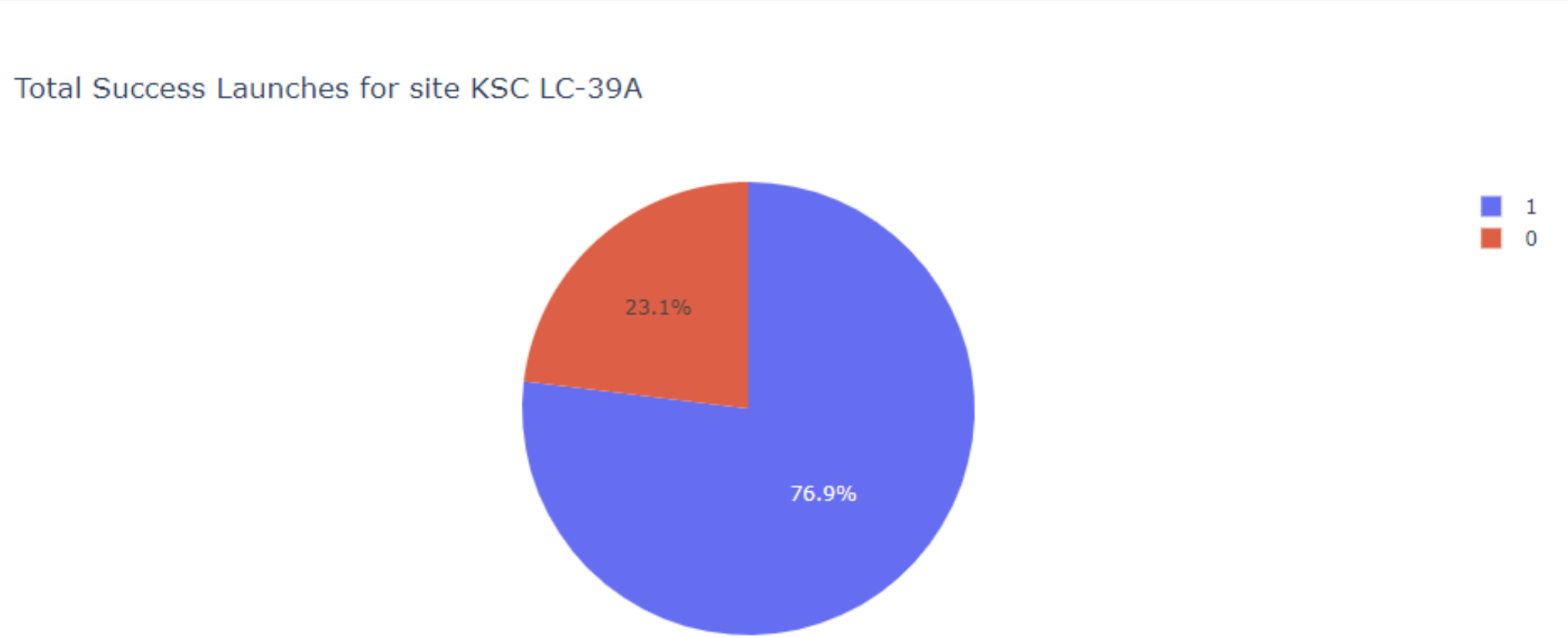


Dashboard's Pie Chart displaying the Success Launch Rate achieved by each Launch Site



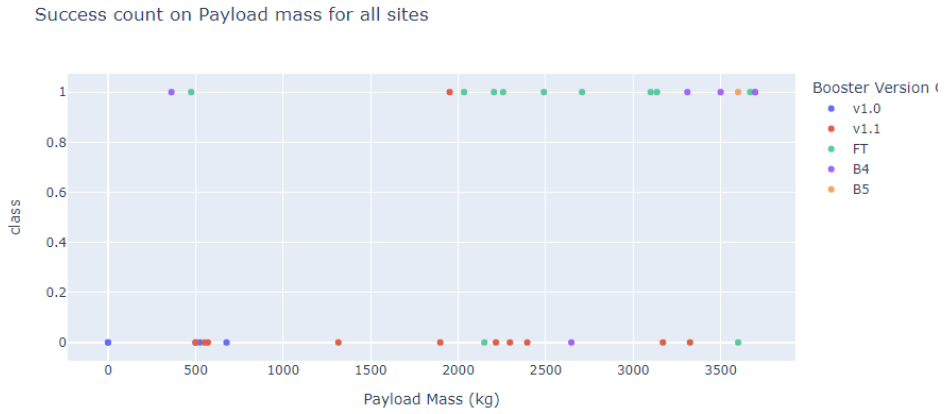
From this pie chart, we can see KSC LC-39A has the highest launch success rate among all launch sites

Dashboard's Pie Chart displaying the Launch Site with the Highest Success Launch Rate

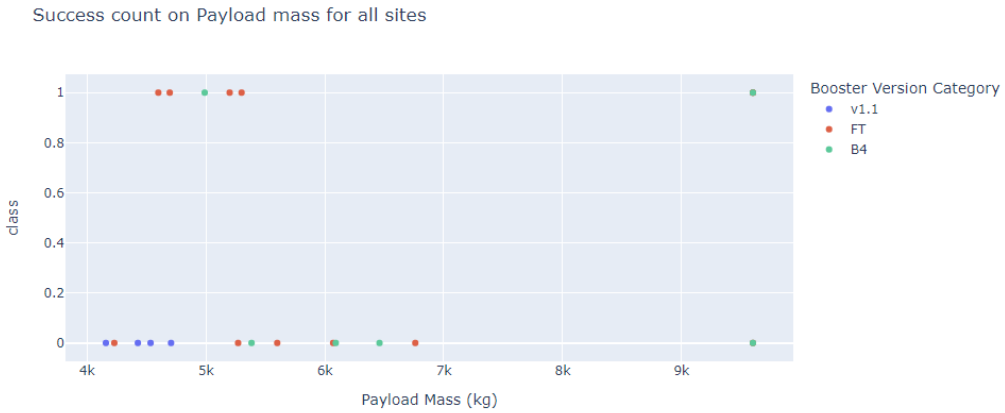


KSC LC-39A achieved 76.9% of successful launch rate and 23.1% of unsuccessful launch rate.

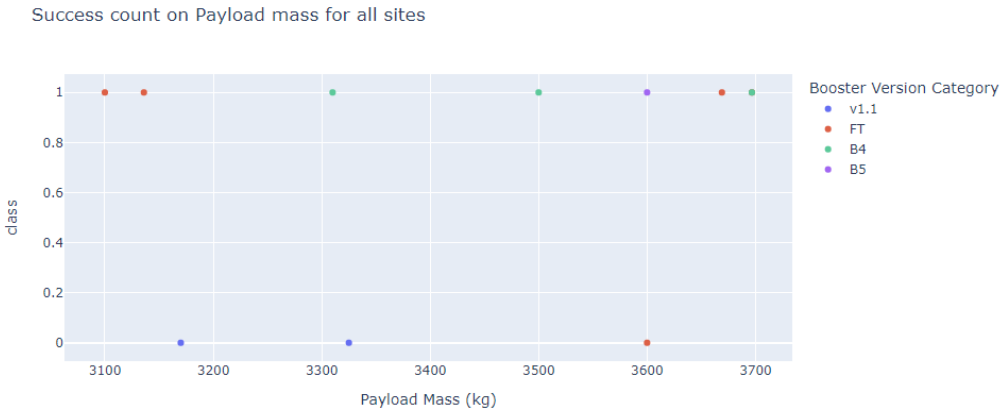
Dashboard's Scatter plot of Payload vs Launch Outcome for all Launch Sites with different payload selected in the range slider



Low weighted payload (0 - 4000 kg)



High weighted payload (4000 - 10000 kg)



The payload range (3000 - 4000 kg) with the highest launch success rate

Which site has the largest successful launches? KSC LC-39A with 10 success launches

Which site has the highest launch success rate? KSC LC-39A with 41.7% success rate

Which payload range(s) has the highest launch success rate? 3000 kg - 4000 kg with 70% success rate (7 successes and 3 failures)

Which payload range(s) has the lowest launch success rate? 6000 kg -7000 kg with 0% success rate (0 success and 4 failures)

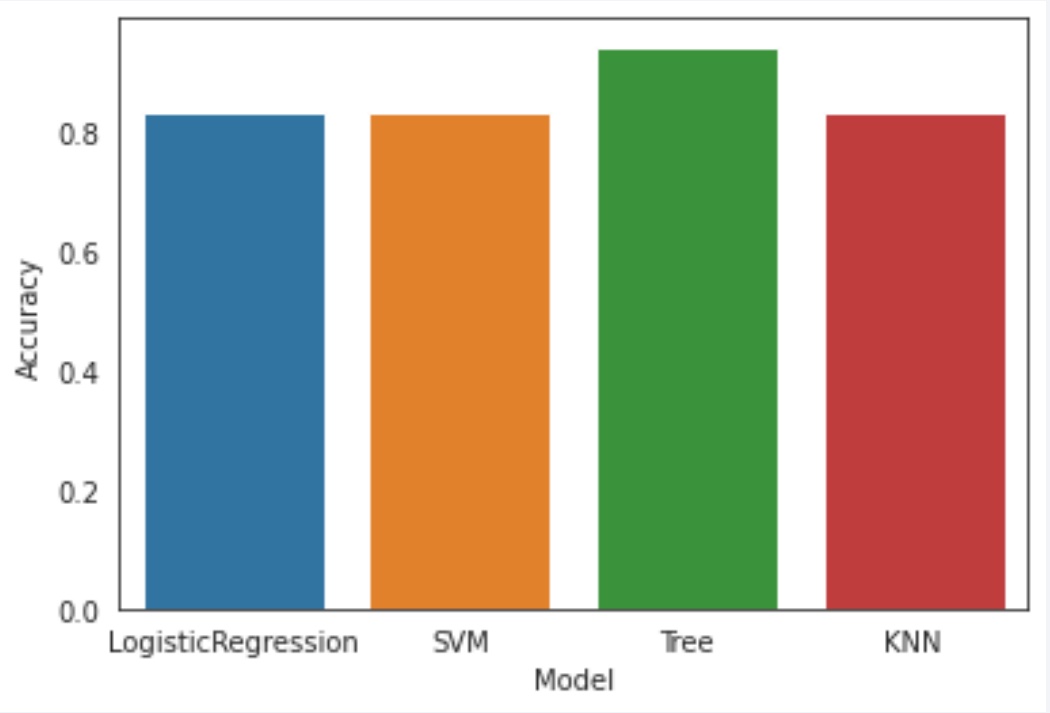
Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? FT has the highest launch success rate of 67% (16 success and 8 failures)

Section 6

Predictive Analysis (Classification)

Classification Accuracy

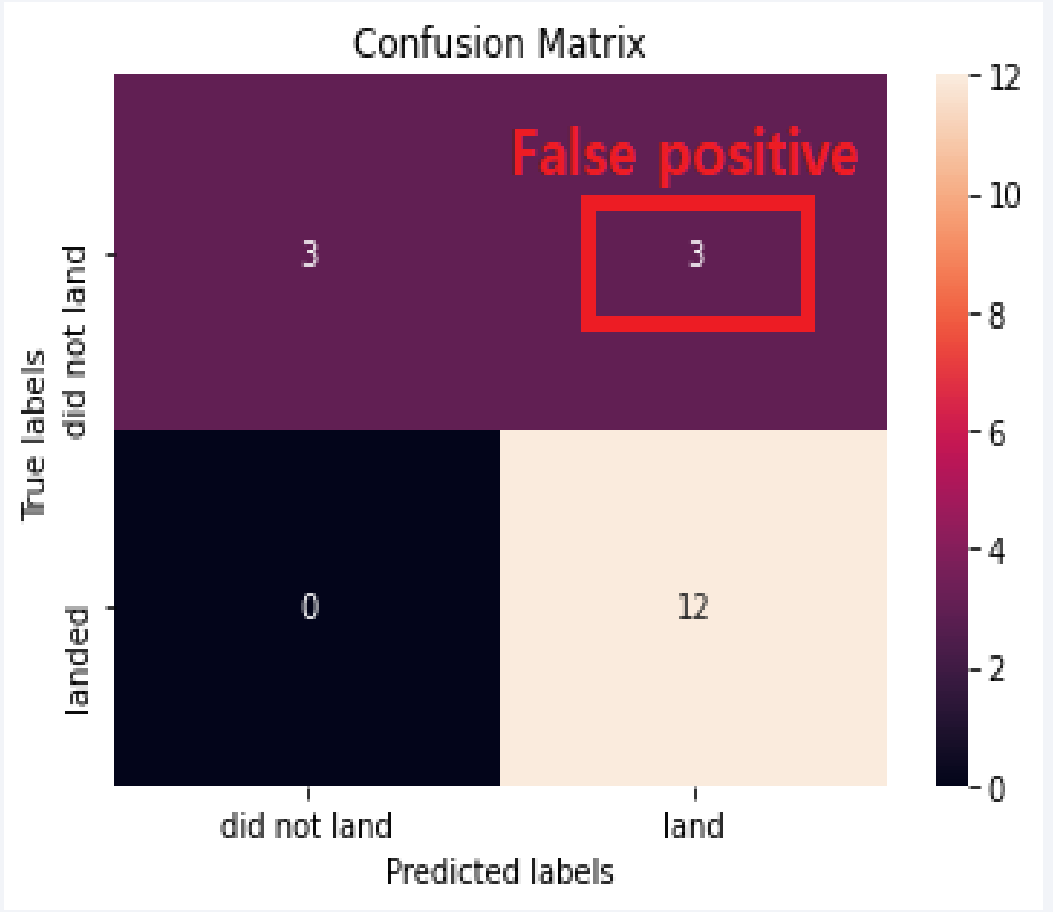
	Model	Accuracy
0	LogisticRegression	0.833333
1	SVM	0.833333
2	Tree	0.944444
3	KNN	0.833333



Model accuracy is almost identical for LogisticRegression, SVM, and KNN, and Tree model has the highest accuracy of 0.944444.

Confusion Matrix for the Decision Tree Classifier

Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives (The model falsely predicted successful landing for 3 occurrences.



Conclusions

From this SpaceX lab, we can conclude:

1. Launch Success rate generally improved over the years from 2013 to 2020
2. Low weighted payloads did better than heavier loads with the best range 3000 kg – 4000 kg
3. Orbit ES-L1, GEO, HEO, SSO have the best Success Rate of 100%
4. KSC LC 39A had the highest success rate of launches among all launch sites
5. The Tree Classifier is the best for Machine Learning model for this dataset

Thank you!