IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abdul Muqadam
04/Jul/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
    - With API
    - With Web Scrapping
  - Data Wrangling
  - Exploratory Data Analysis (EDA)
    - With SQL
    - With Visualization
    - Folium
  - Machine Learning (ML) Models
- Summary of all results
  - EDA Results
  - Interactive Analyses
  - ML Predictions

Introduction

- SpaceX is a company that offers new and improved rocket launches at a fraction of the price.
- A typical rocket launch used to cost at least $165mn per launch depending upon the payload and other factors.
- SpaceX offers the same services at a base price of $62mn. This is achieved because SpaceX reuses first stage by making it land after a launch.
- Some launches are successful and some are not.
- This project aims to find
  - Factors influencing landing outcome
  - The relationship among variables affecting the outcome
  - How to increase the successful-landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected using SpaceX API and Web scrapping from Wikipedia
- Perform data wrangling
  - The missing payload values were replaced with mean payload.
  - A categorical variable named ‘class’ was added to show successful (1) and unsuccessful landing
  - All categorical variables were ‘one-hot-encoded’
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- The data was collected from two different sources
  - SpaceX REST API
    - API EndPoint used was https://api.spacexdata.com/v4/launches/past
    - Requests Object was transformed into JSON string using *.json()* method
    - The JSON string was converted into Pandas DataFrame using *json_normalize()* method
  - Wikipedia via Web scrapping
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [16]: # Use json_normalize meethod to convert the json result into a dataframe
temp = response.json()
#temp
data = pd.json_normalize(temp)
data.head()
```

Out[16]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	cap
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	

https://github.com/mhnaqvi/DataScienceCapstone/blob/main/jupyter-labs-spacex-data-collection-api_SOL.ipynb

Specify API Endpoint URL

Use Request Object to request API and assign it to response object

Convert Response to JSON string using .json() method

Convert JSON string to Pandas DataFrame using .json_normalize() method

Data Collection - Scraping

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [10]:
```
# use requests.get() method with the provided static_url
# assign the response to a object

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

Create a `BeautifulSoup` object from the HTML `response`

In [11]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

In [17]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

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

Specify Wikipedia URL

Use Request Object to get mentioned URL's text and assign it to response object

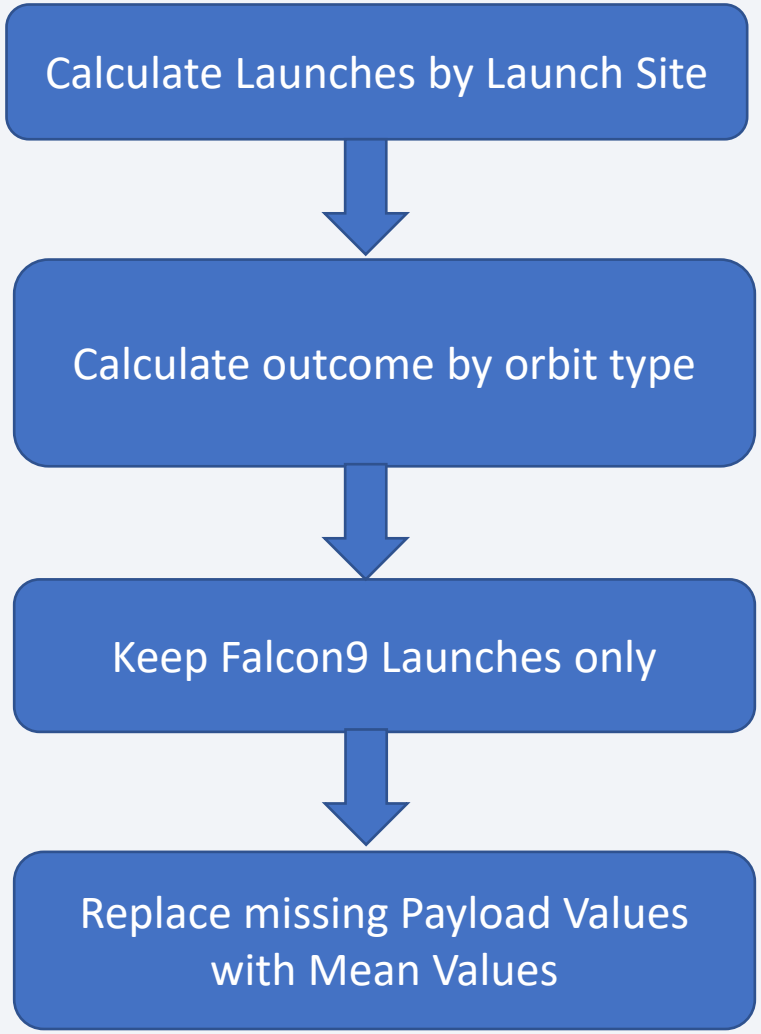
Use BeautifulSoup to parse the text

Find HTML Table and extract values

https://github.com/mhnaqvi/DataScienceCapstone/blob/main/jupyter-labs-webscraping_SOL.ipynb

Data Wrangling

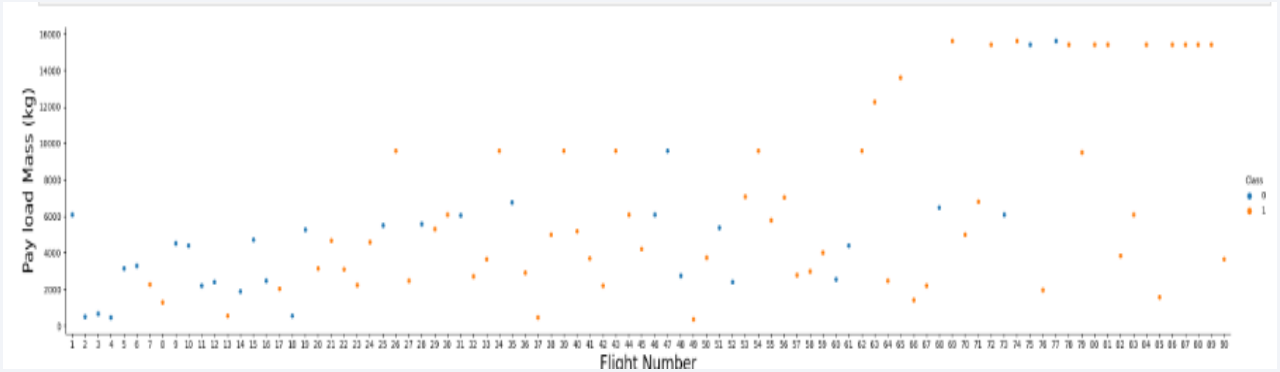
- Launches by launch site were calculated at first
- Then outcome by orbit type was calculated
- Falcon9 data is kept and the rest was discarded
- Missing PayloadMass values were replaced with mean/average values

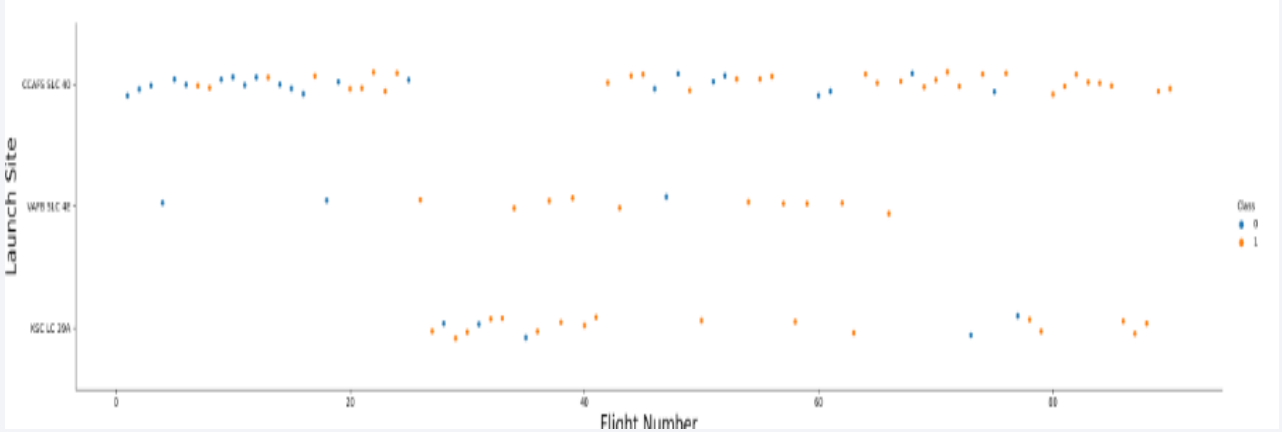


https://github.com/mhnaqvi/DataScienceCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling_SOL.ipynb

EDA with Data Visualization

- FlightNumber vs PayloadMass
- FlightNumber vs LaunchSite
- PayloadMass vs LaunchSite
- Orbit vs Outcome
- Orbit vs FlightNumber
- Orbit vs PayloadMass

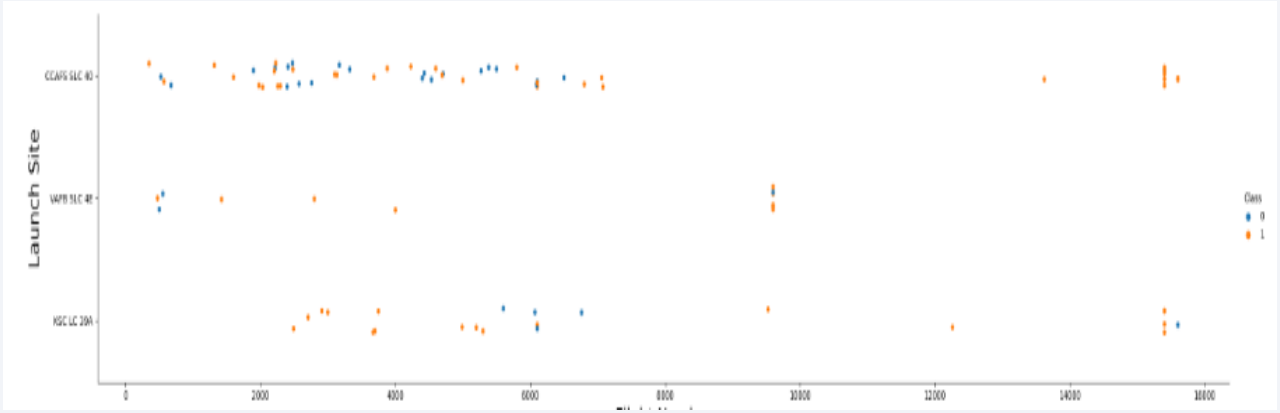


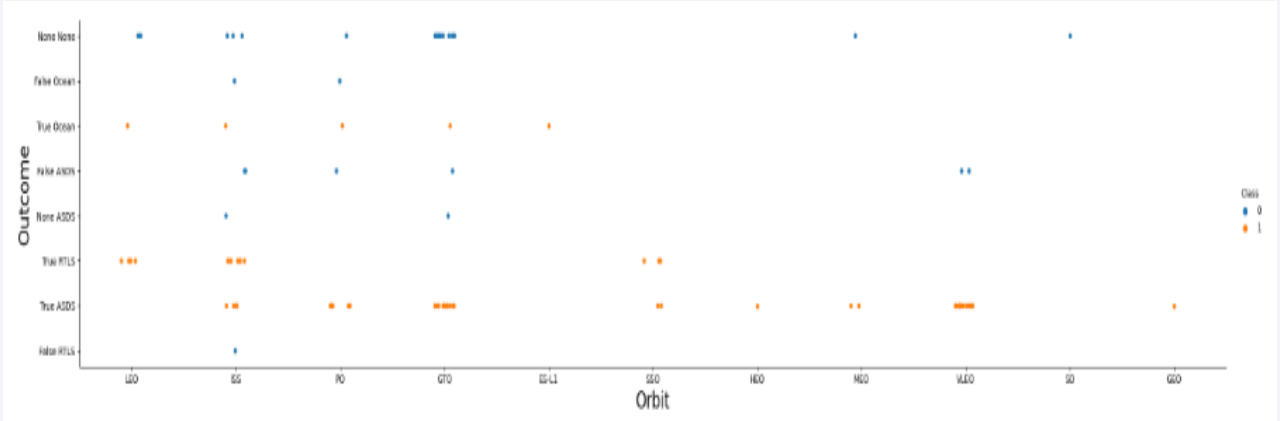


https://github.com/mhnaqvi/DataScienceCapstone/blob/main/jupyter-labs-eda-dataviz_SOL.ipynb

EDA with Data Visualization

- FlightNumber vs PayloadMass
- FlightNumber vs LaunchSite
- PayloadMass vs LaunchSite
- Orbit vs Outcome
- Orbit vs FlightNumber
- Orbit vs PayloadMass

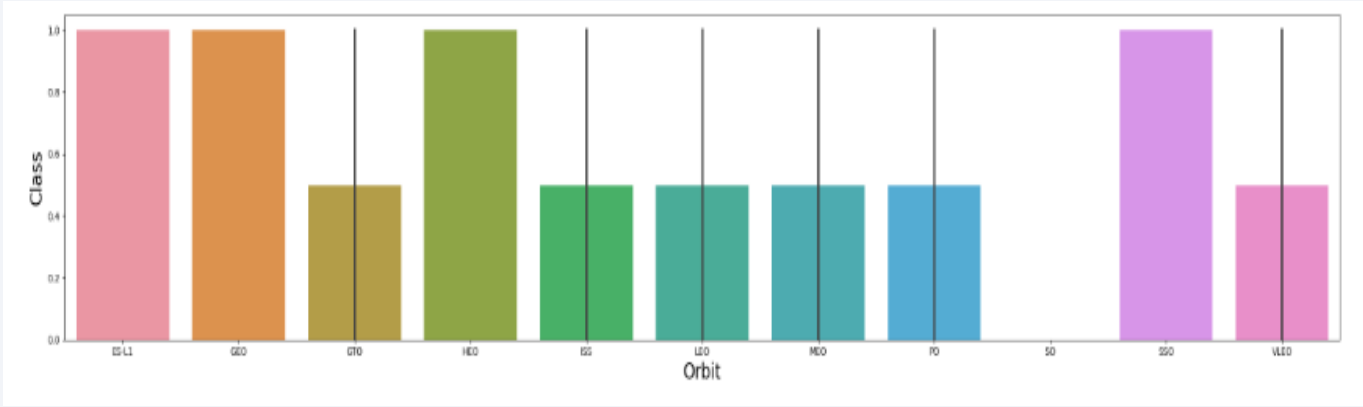


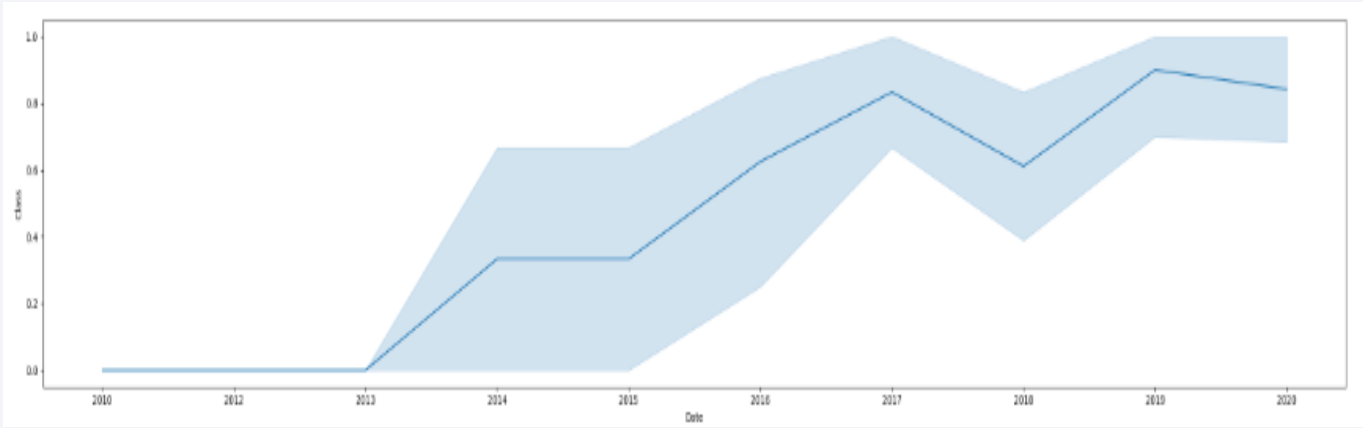


https://github.com/mhnaqvi/DataScienceCapstone/blob/main/jupyter-labs-eda-dataviz_SOL.ipynb

EDA with Data Visualization

- FlightNumber vs PayloadMass
- FlightNumber vs LaunchSite
- PayloadMass vs LaunchSite
- Orbit vs Outcome
- Orbit vs FlightNumber
- Orbit vs PayloadMass





https://github.com/mhnaqvi/DataScienceCapstone/blob/main/jupyter-labs-eda-dataviz_SOL.ipynb

EDA with SQL

- Displayed
  - the names of the unique launch sites in the space mission
  - 5 records where launch sites begin with the string 'CCA'
  - the total payload mass carried by boosters launched by NASA (CRS)
  - average payload mass carried by booster version F9 v1.1
  - the list and the date when the first successful landing outcome in ground pad was achieved.
  - the list and the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - the list and the total number of successful and failure mission outcomes
  - the list and the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - the list and the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
  - the 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

https://github.com/mhnaqvi/DataScienceCapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite_SOL.ipynb

Build an Interactive Map with Folium

- Latitude and longitudes of launch locations were marked with circular markers around each site.
- Green marker signifies successful outcome whereas the red marker signifies unsuccessful outcome
- The distance of railways, highways, and coastline from launch site was calculated

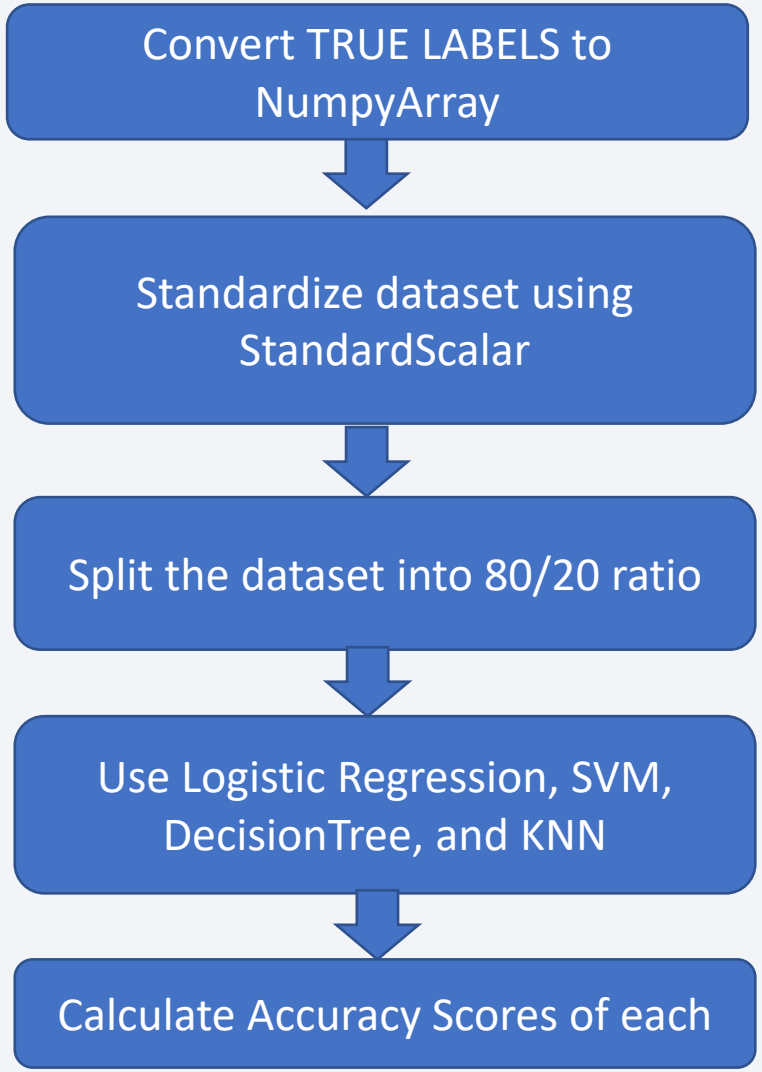
https://github.com/mhnaqvi/DataScienceCapstone/blob/main/lab_jupyter_launch_site_location_SOL.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard was created
- User can extract charts based on
  - Launch site (Pie Charts)
  - Payload (Scatter Charts based on slider)

https://github.com/mhnaqvi/DataScienceCapstone/blob/main/spacex_dash_app_SOL.py

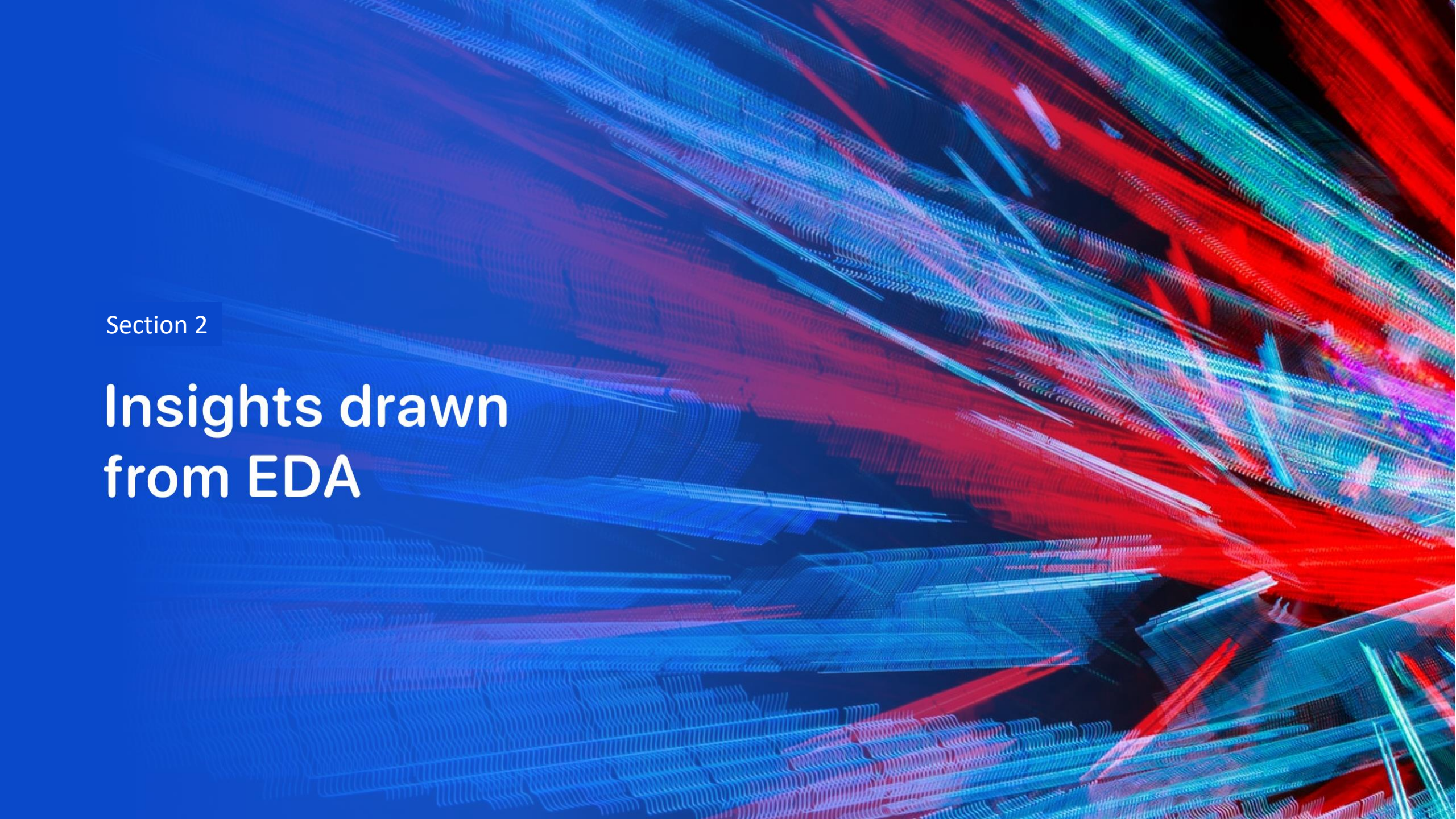
Predictive Analysis (Classification)



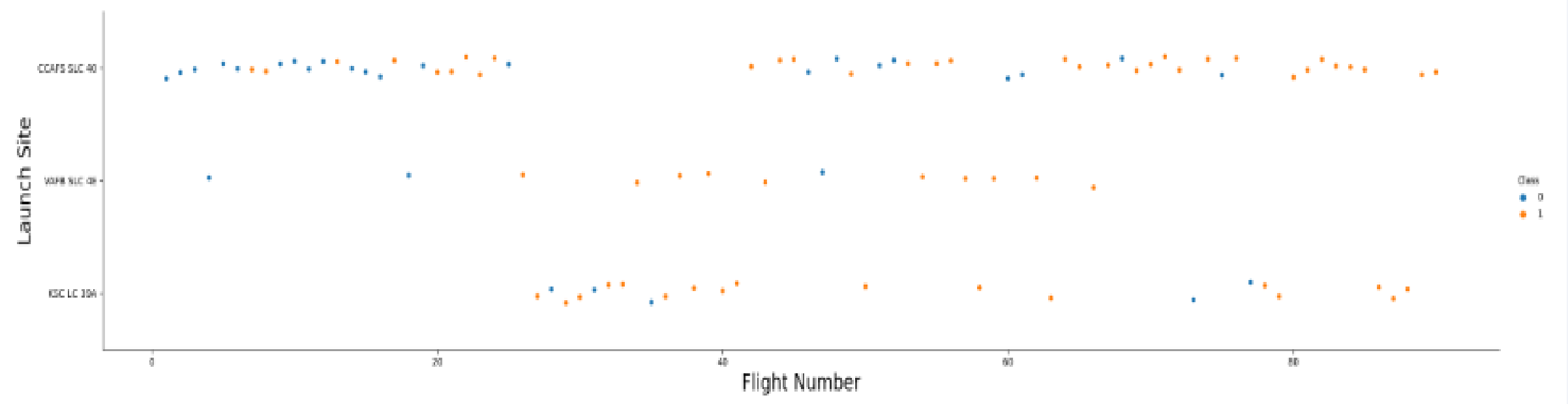
https://github.com/mhnaqvi/DataScienceCapstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_SOL.ipynb

Results

- The results are categorized in the following categories
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

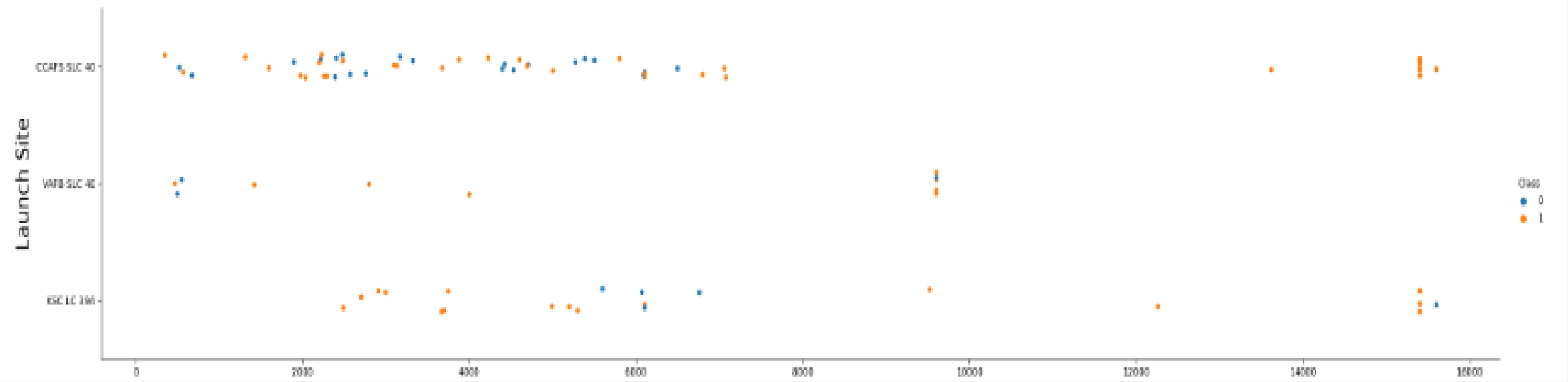

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



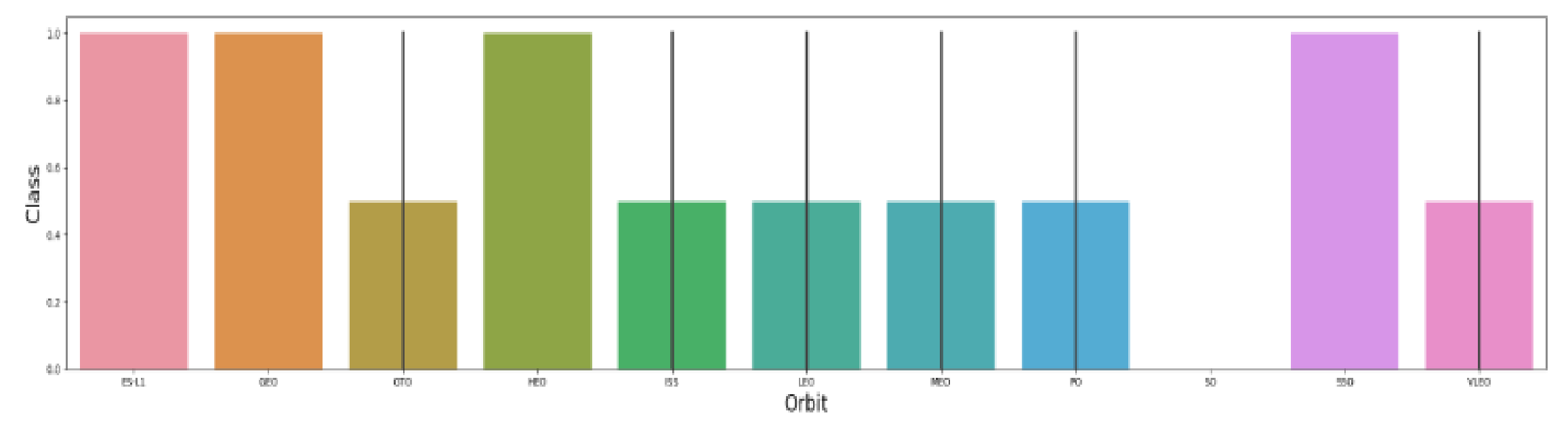
- This plot shows that the number of flights is directly proportional to launch sites.
- But, the CCAFS SLC 40 has a better record in terms of successful outcomes

Payload vs. Launch Site



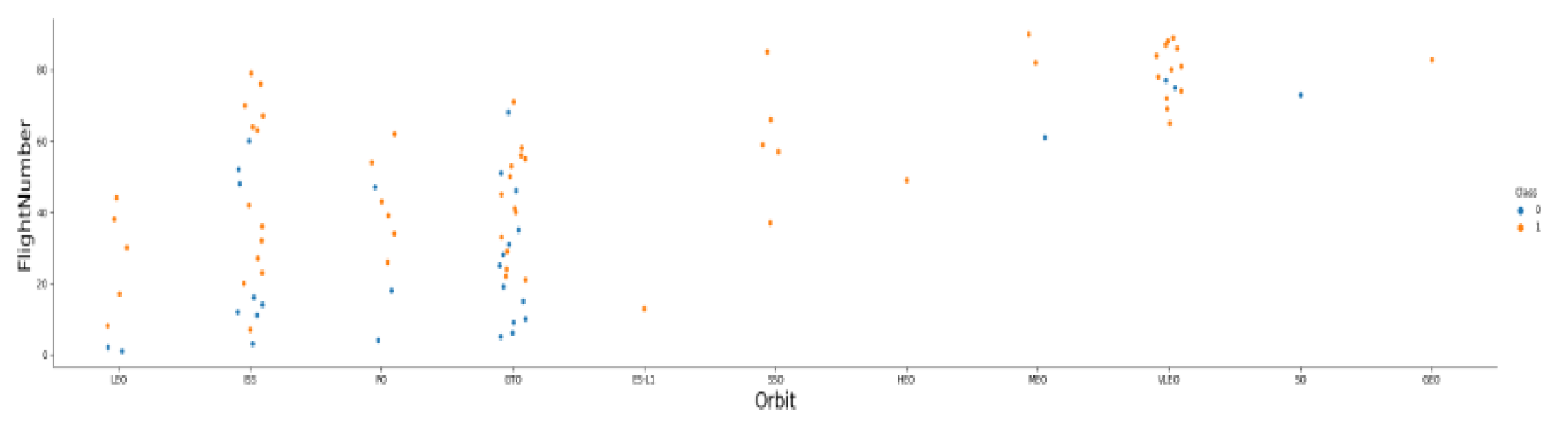
- This plot shows that as the payload mass increases the successful outcome increases.
- Launch site has no effect on successful outcome.

Success Rate vs. Orbit Type



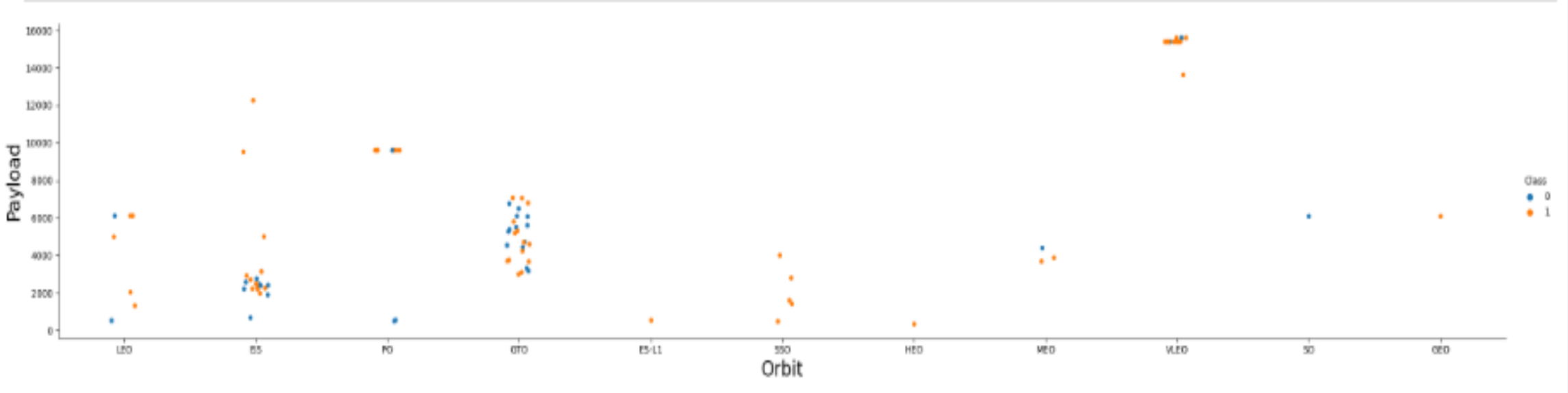
- No effective conclusion is drawn

Flight Number vs. Orbit Type



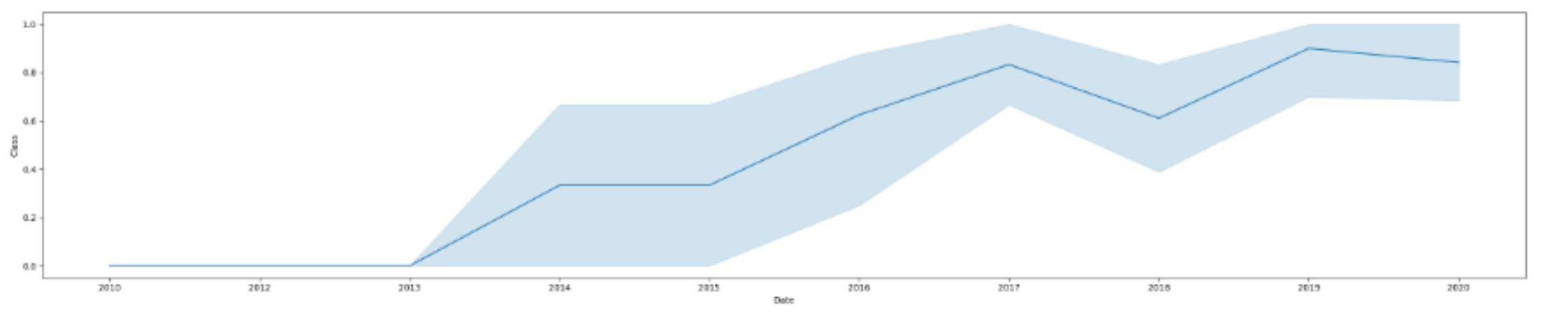
- Higher number of launches for each orbit leads to a higher number of successful outcome

Payload vs. Orbit Type



- Higher payload leads to successful outcome for LEO, ISS, and PO orbits. Whereas it has a negative impact on MEO and VLEO orbits
- GTO has no effect
- SO, GEO, and HEO are unconclusive

Launch Success Yearly Trend



- Since 2013, the success is increasing until the available date i.e. 2020

All Launch Site Names

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

In [9]:
```
%sql select distinct(Launch_Site) from SPACEXTBL
```

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

Out[9]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

- Distinct is used to display the site name exactly once

Launch Site Names Begin with 'CCA'

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

In [11]:
```
%sql select * from SPACEXTBL where Launch_site like 'CCA%' limit 5
```

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

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

- Five records starting where the launch site name starts from are displayed

Total Payload Mass

Task 3

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

In [16]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)'
```

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

Out[16]:

sum(PAYLOAD_MASS__KG_)
45596.0

- Total payload mass where the client is NASA (CRS) is calculated

Average Payload Mass by F9 v1.1

Task 4

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

In [17]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_version='F9 v1.1'
```

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

Out[17]:

avg(PAYLOAD_MASS__KG_)
2928.4

- The average payload mass carried by booster version F9 v1.1 is calculated

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [28]:
```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome='Success (ground pad)'
```

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

Out[28]:

min(DATE)
01/08/2018

- The date of the first successful landing outcome on ground pad is calculated

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

In [30]:
```
%sql select Booster_version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and
```

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

Out[30]:

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

- Lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [34]:
```
%sql select * from SPACEXTBL limit 5
%sql select count(mission_outcome) from SPACEXTBL where mission_outcome = 'Success' or mission_outcome='Failure'
```

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

Out[34]:

count(mission_outcome)
98

- This query displays the total number of successful and total number of failures

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [35]:
```
%sql select * from SPACEXTBL limit 5
%sql select distinct(booster_version) from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL
```

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

Out[35]:

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

- The names of the booster which have carried the maximum payload mass are listed

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [50]:
```
%sql select * from SPACEXTBL
%sql select substr(Date, 4,2) as Month, substr(Date, 7,4) as Year, Landing_outcome, booster_version, launch_site from SPACEX
```

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

Out[50]:

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
01	2016	Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
04	2016	Failure (drone ship)	F9 FT B1020	CCAFS LC-40
06	2016	Failure (drone ship)	F9 FT B1024	CCAFS LC-40

- The query displays the list of the failed landing_outcomes in drone ship, their booster versions, and launch site names for since 2015

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

Task 10

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.

In [58]:
```
%sql select count(landing_outcome) from SPACEXTBL where Date between '04/06/2010' and '20/03/2017' order by date desc
```

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

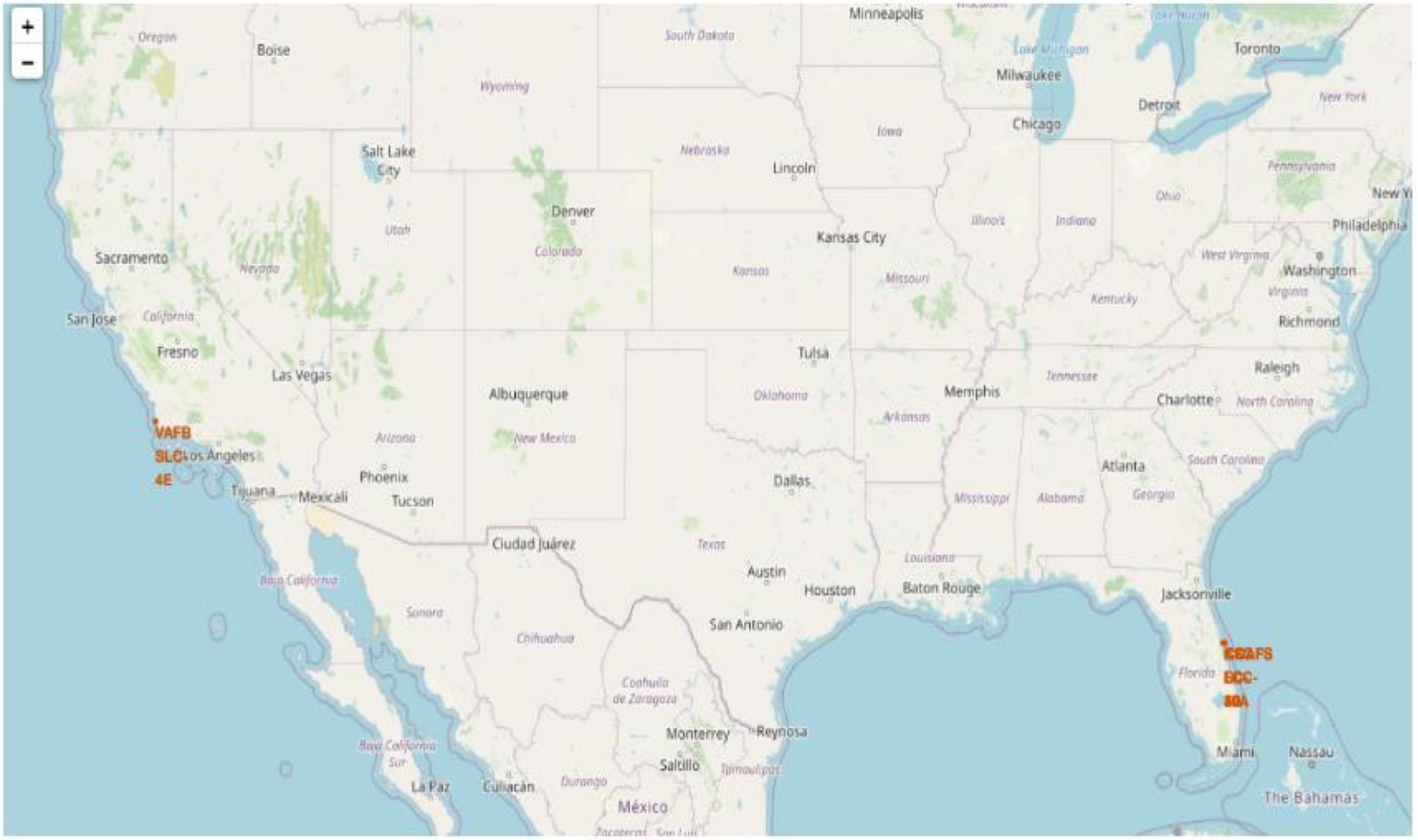
Out[58]:

count(landing_outcome)
55

- 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

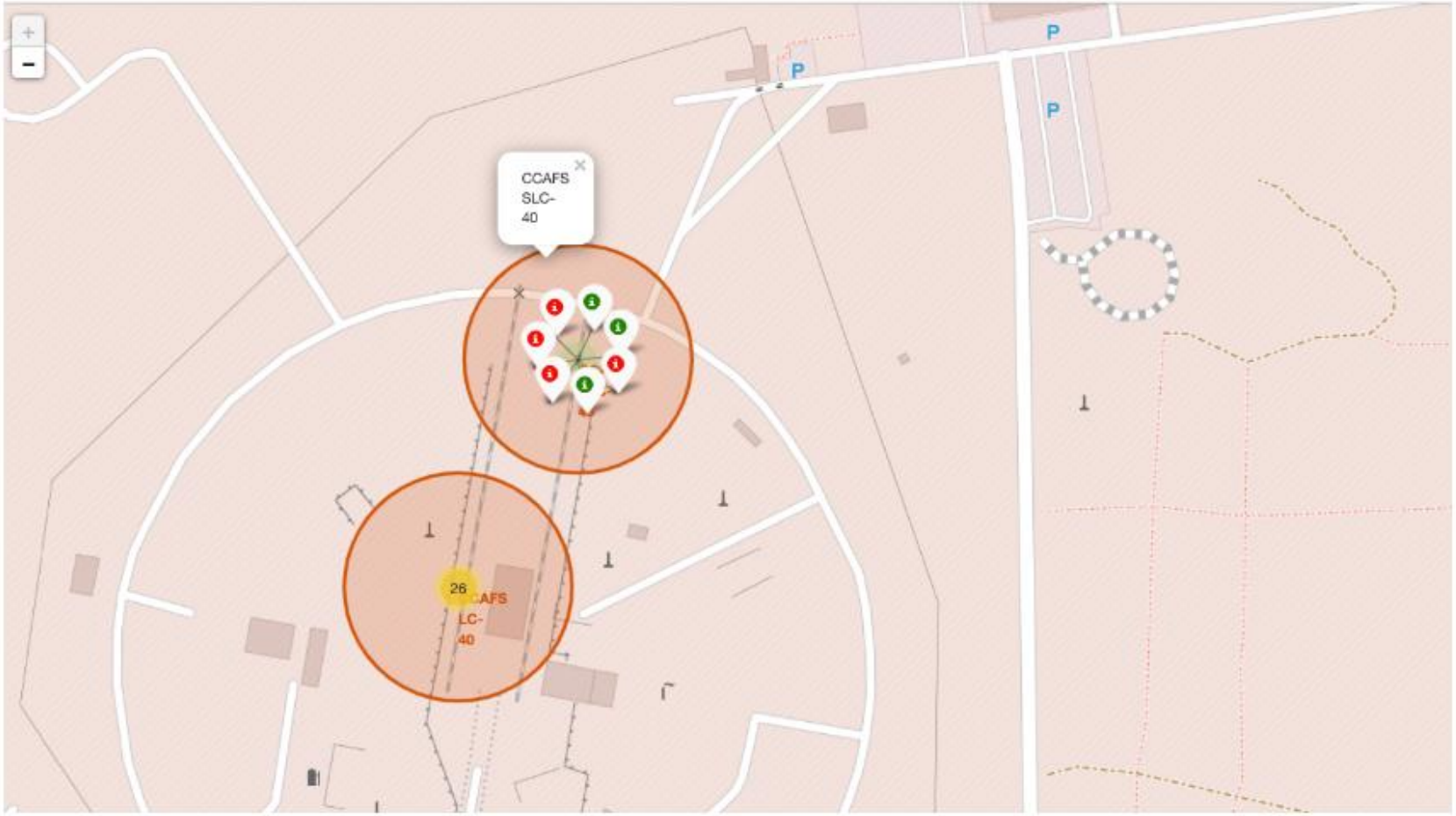

Section 3
Launch Sites
Proximities Analysis

Launch Sites Locations



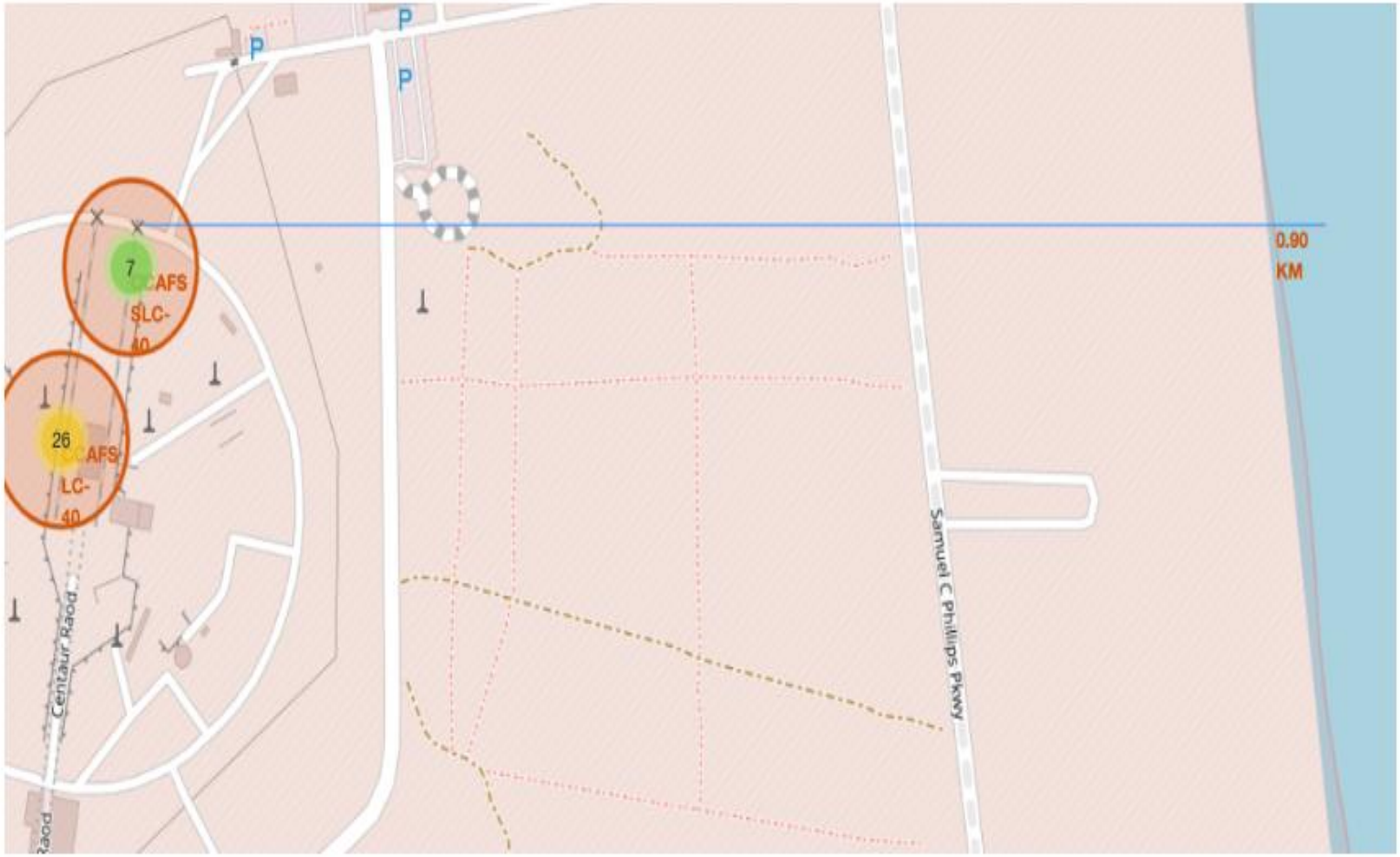
- This folium chart shows the SpaceX's launch sites in United States

Marked Folium Map

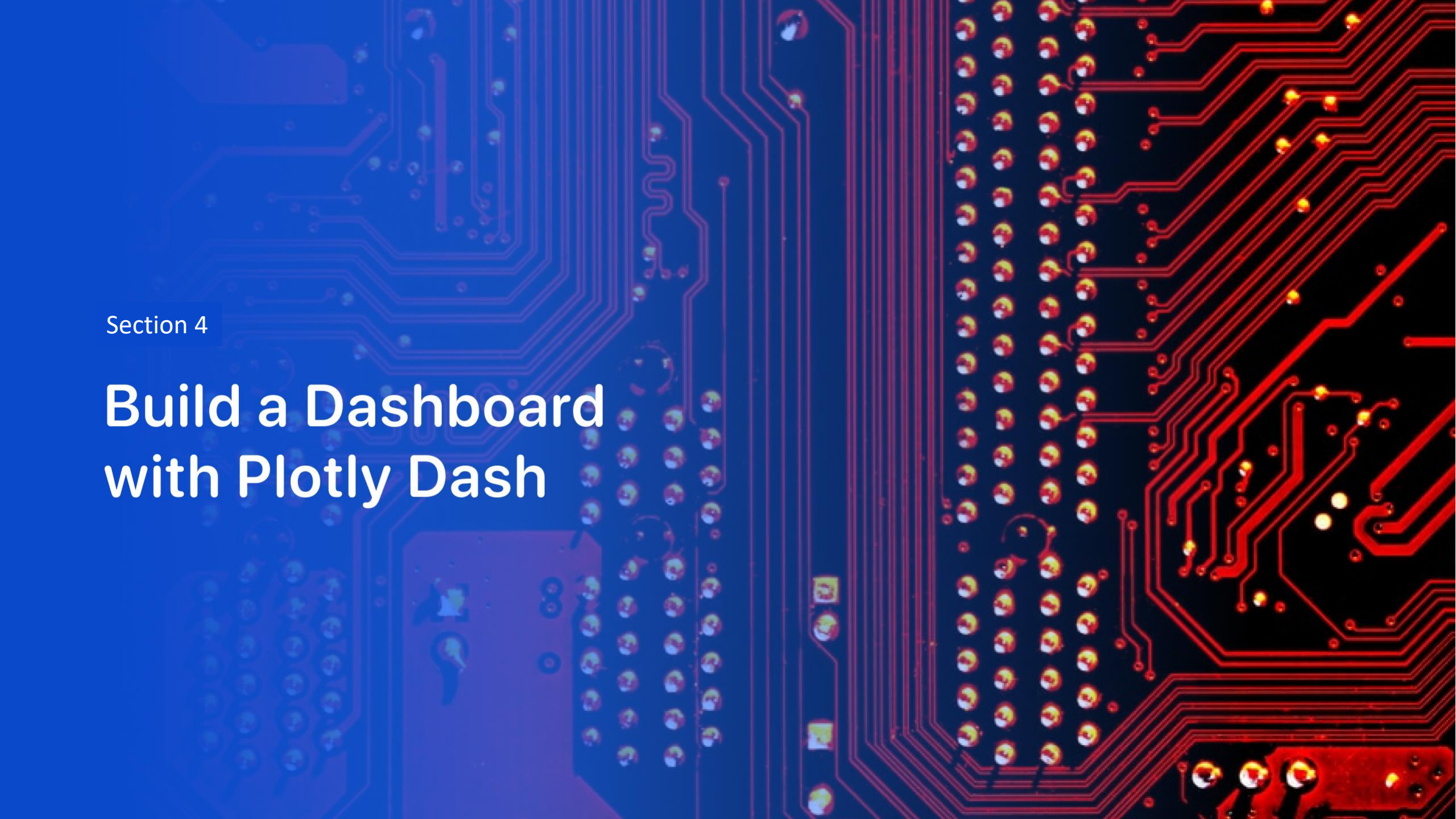


- This map shows the marked sites. Green for successful landings and red for unsuccessful landings

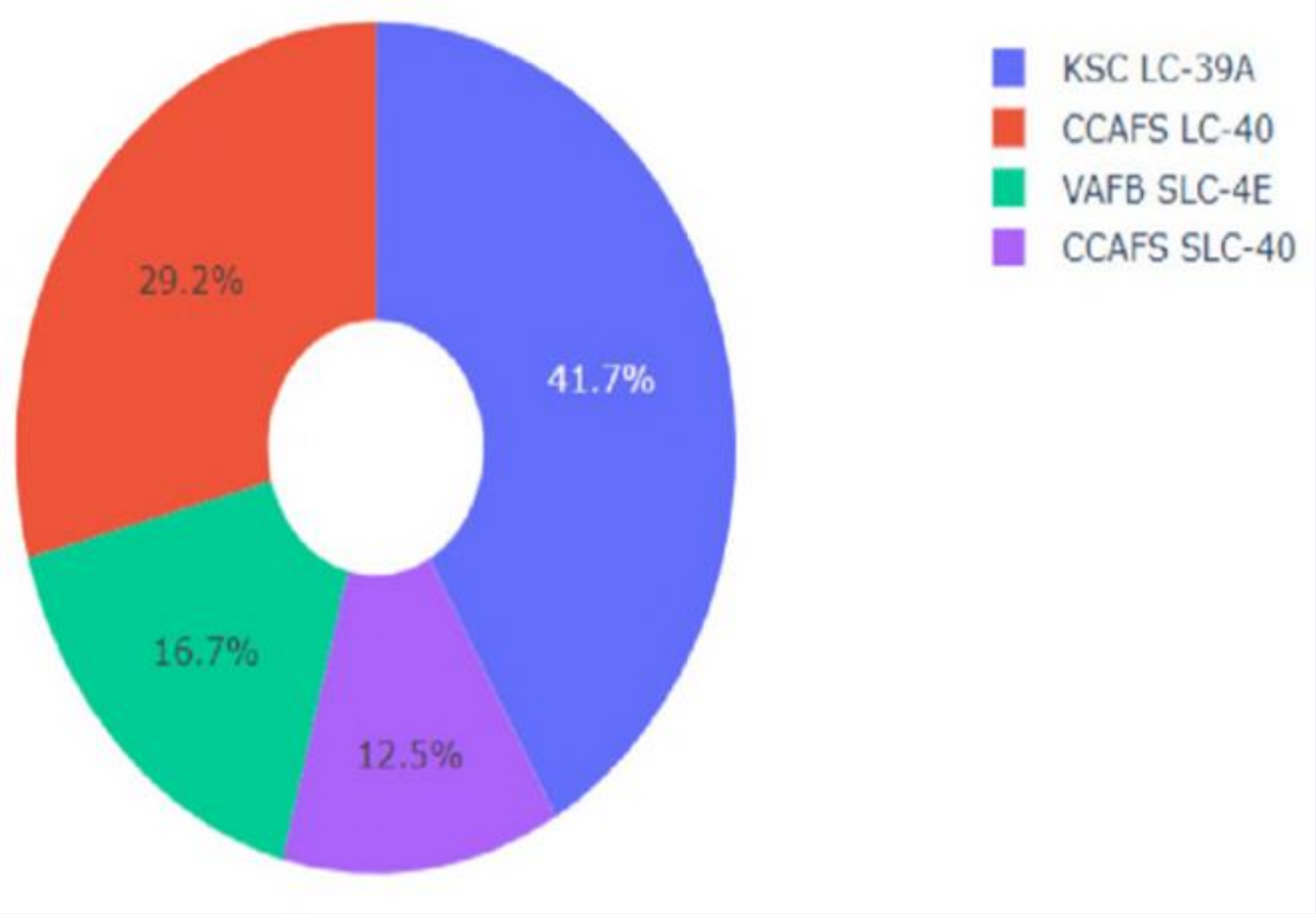
Distance Folium Map



- Distance to coastline

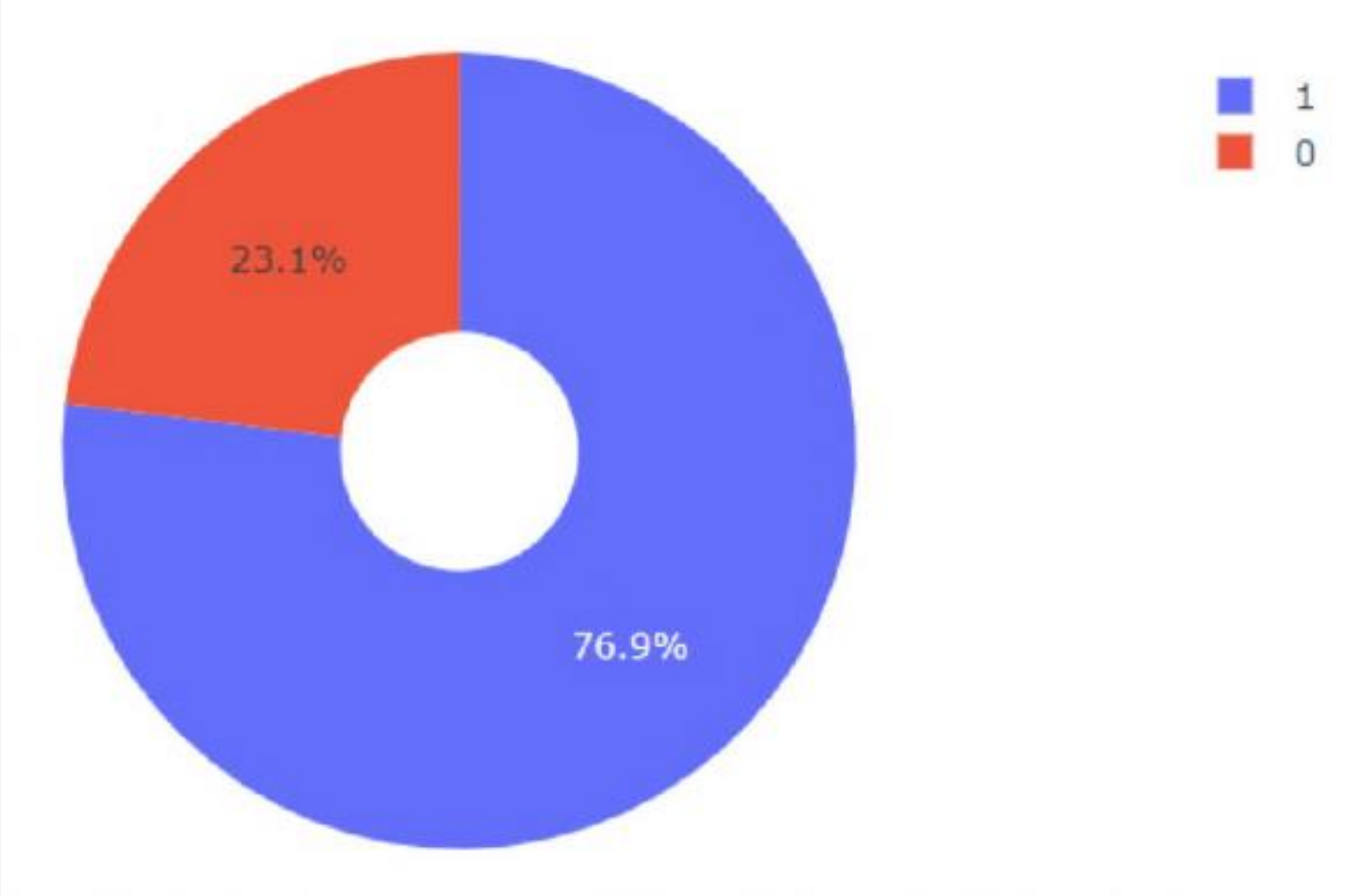

Section 4
Build a Dashboard with Plotly Dash

Site-wise Success Rate



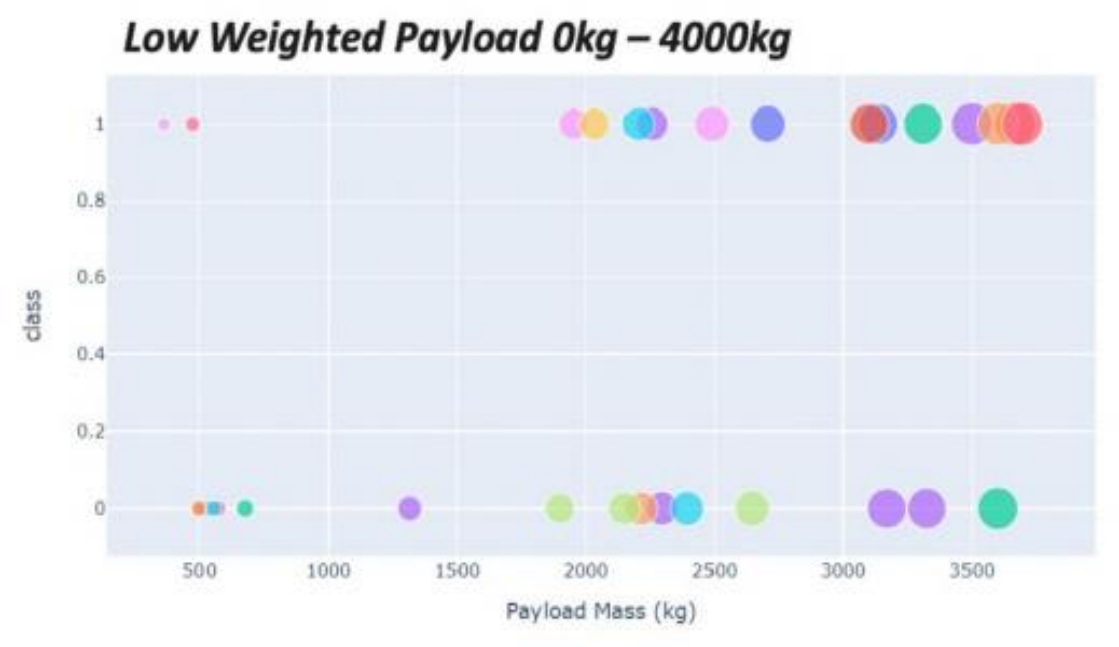
- 41.70% of successful landings are achieved by KSC LC-39A
- Followed by CCAFS LC-40 at 29.2%, VAFB SLC-4E at 16.7%, and CCAFS SLC-40 at the last

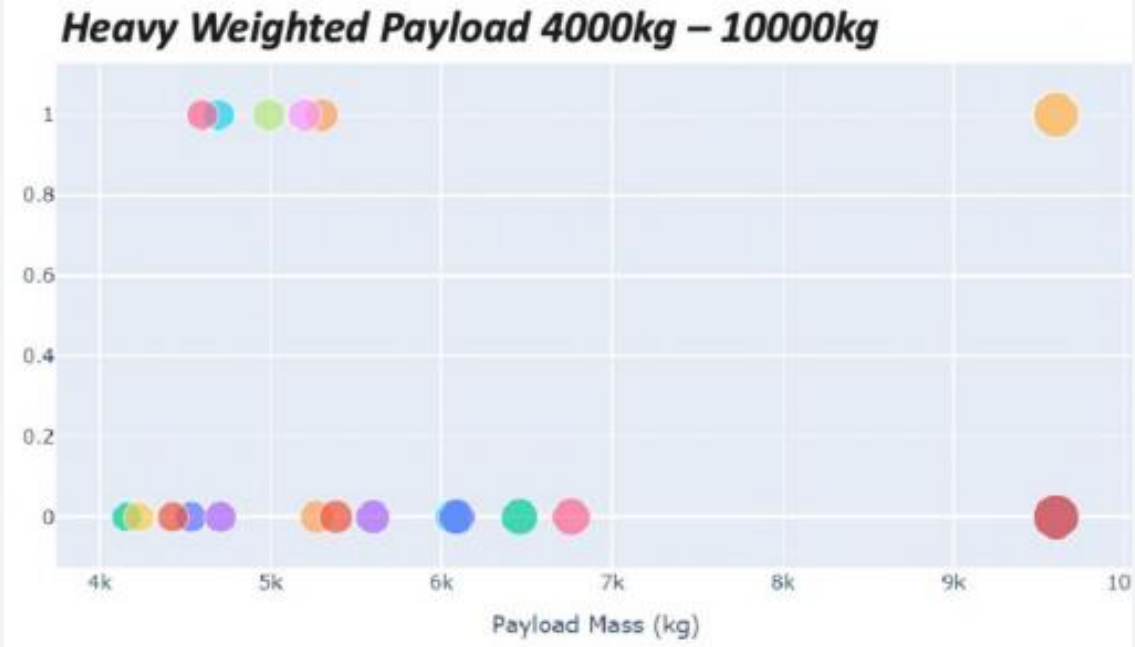
KSC LC-39A Launch Ratio



- Launch success ratio of KSC LC-39A
- It achieved 76.90% success

PayloadMass vs Outcome Plot





- Low payload has a higher success rate

Section 5

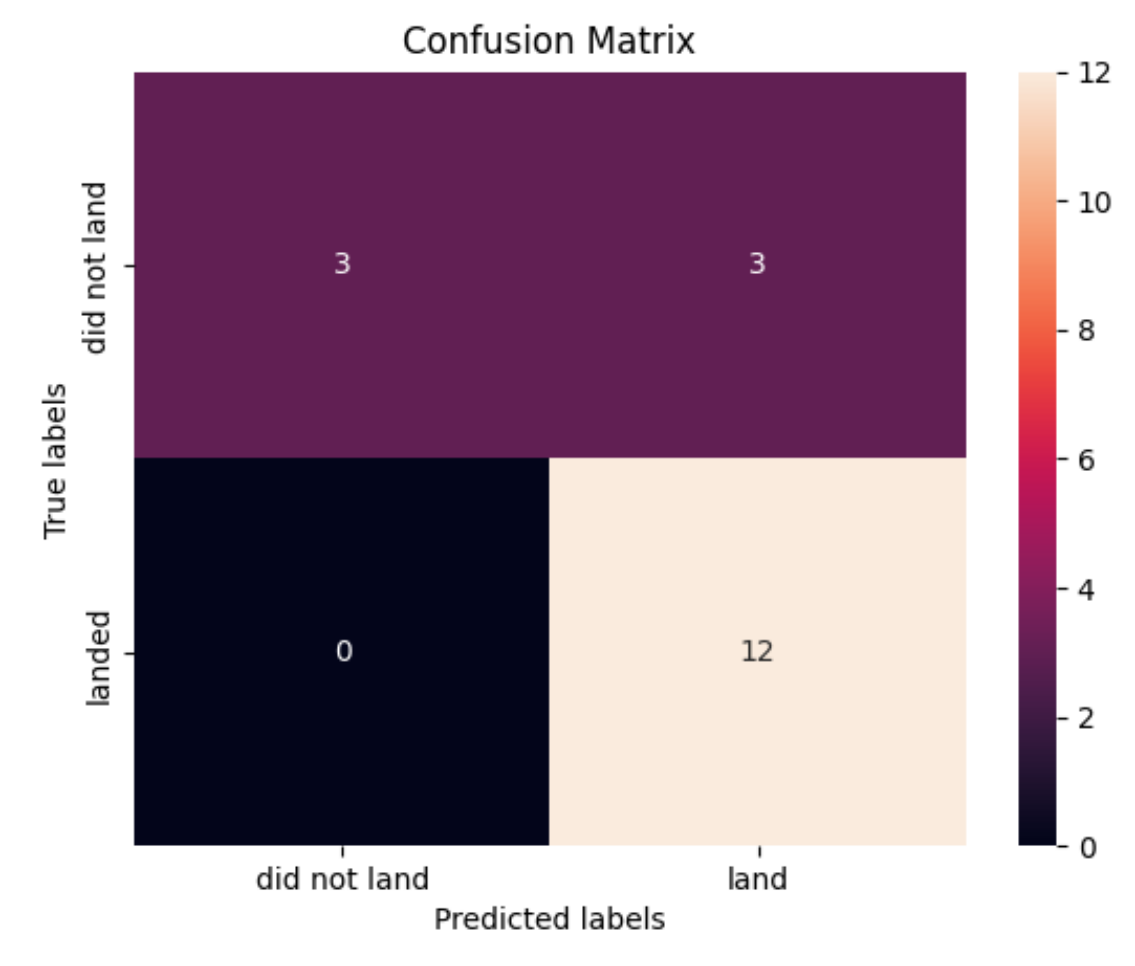
Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

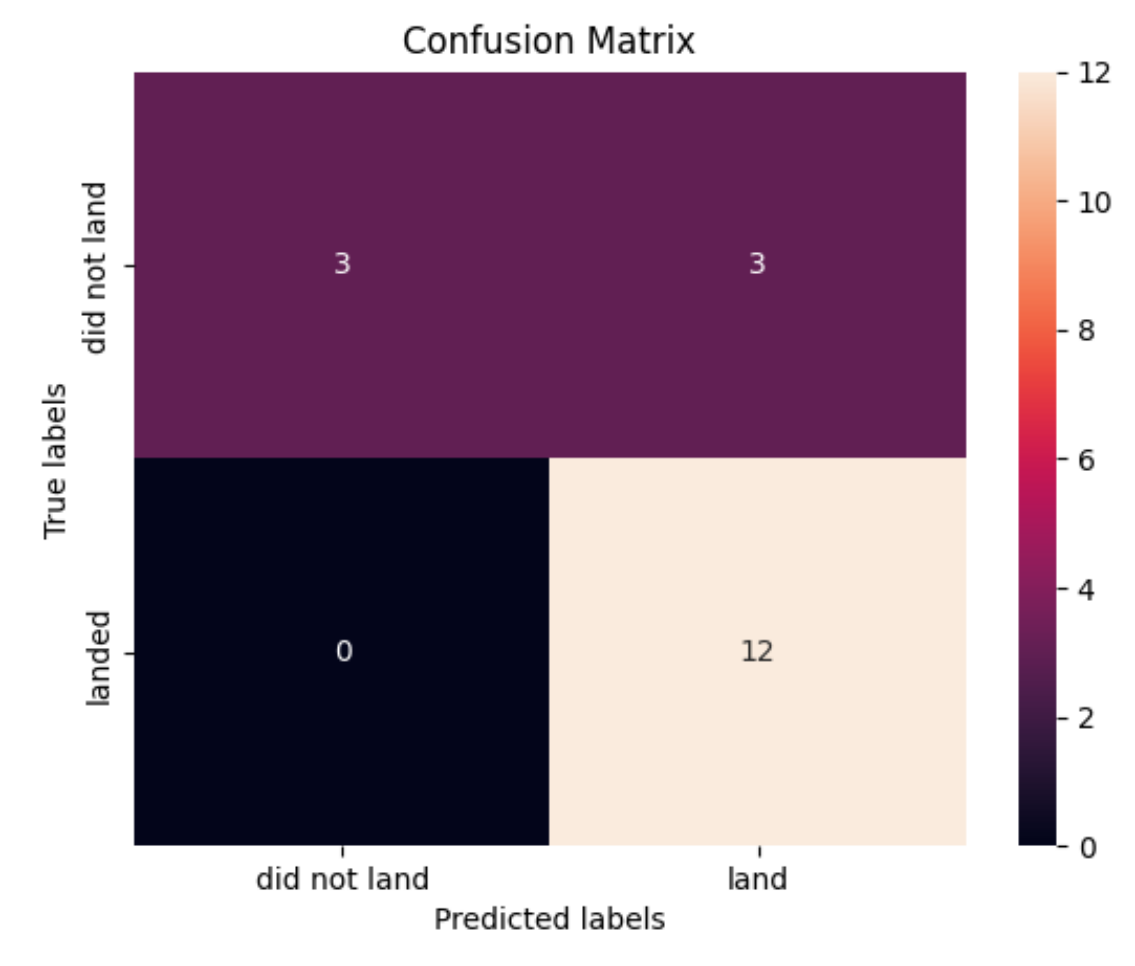
- Find which model has the highest classification accuracy

Confusion Matrix – Logistic Regression



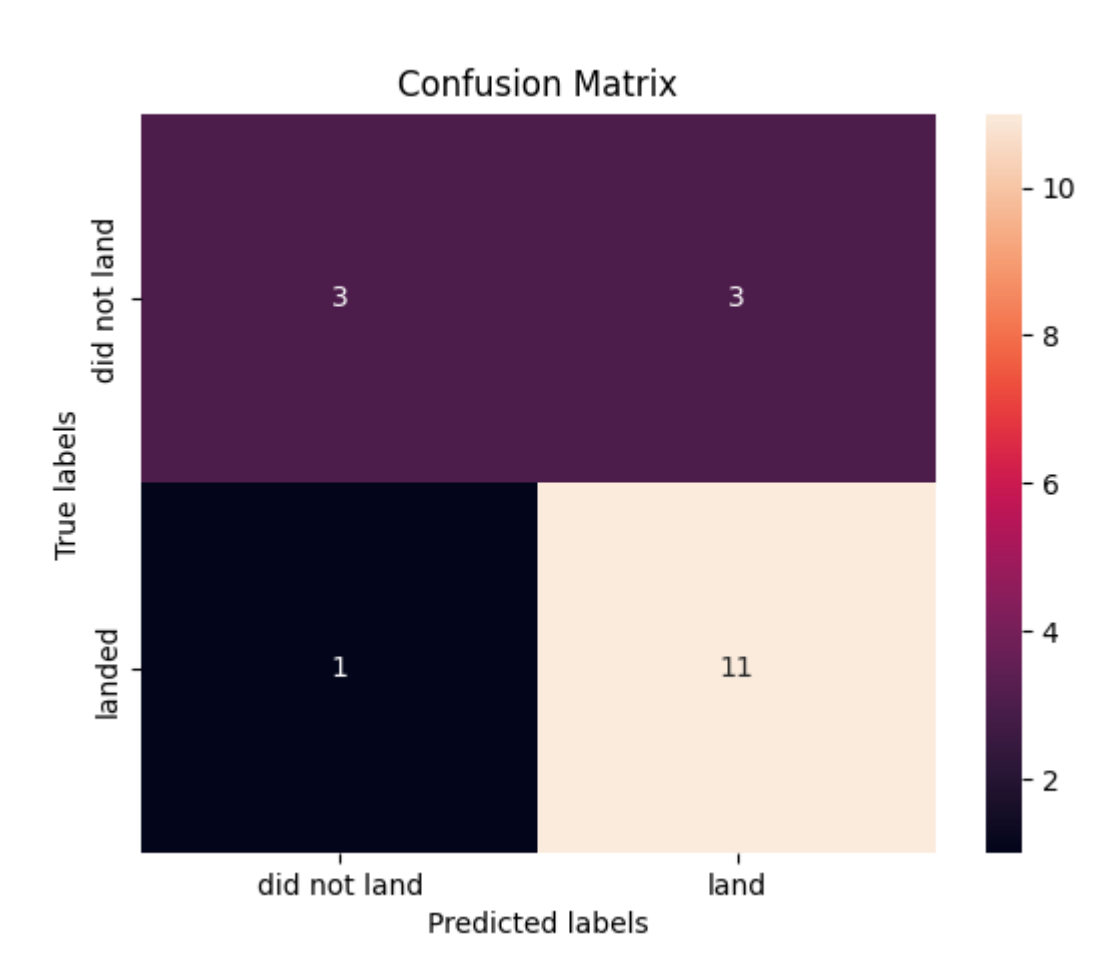
- Show the confusion matrix of the best performing model with an explanation

Confusion Matrix – SVM



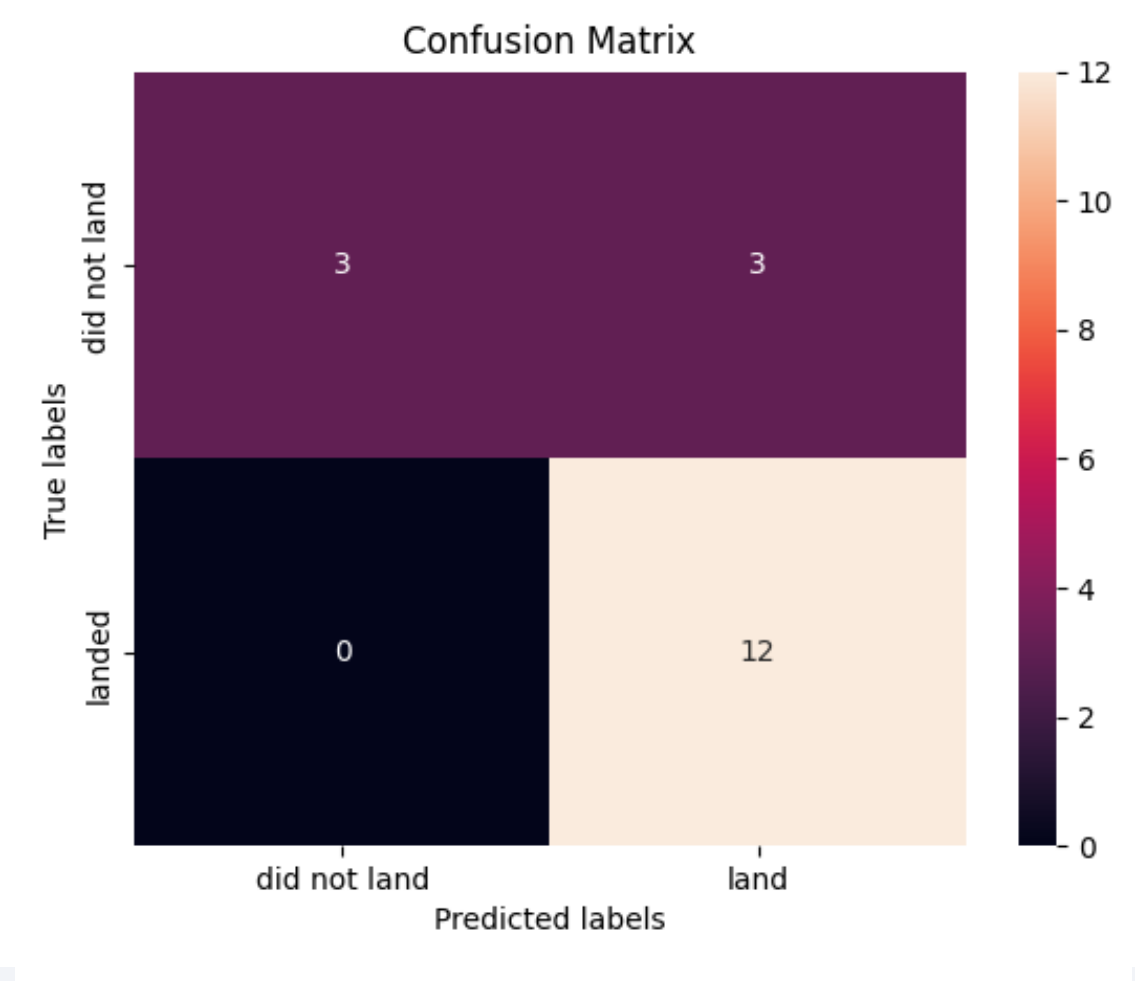
- Show the confusion matrix of the best performing model with an explanation

Confusion Matrix – Decision Tree



- Show the confusion matrix of the best performing model with an explanation

Confusion Matrix – KNN



- Show the confusion matrix of the best performing model with an explanation

Conclusions

- KSC LC39-A is the best performing site
- Light payloads performed better
- Since 2013 the launches have improved in terms of success
- Decision Tree was the worst performing algorithm

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!