Winning Space Race with Data Science

Abinash Pun

Sep 23, 2022

Outline

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

Executive Summary

- Methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
    - Data Visualization
    - SQL
  - Interactive Visual Analysis
    - Folium
    - Dashboard
  - Predictive Analysis (Classification)
- Results
  - EDA results
  - Interactive analytics
  - Predictive Analysis

Introduction

- Project background and context
  - SpaceX claims to reduce the cost of Falcon 9 to $62m from usual cost of $165m. This reduction is mostly due to the reuse of first stage of the rocket.
- Problems you want to find answers
  - The project goal is to predict if the first stage of SpaceX Falcon9 will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - Removing null values and irreverent columns
  - One hot in coding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models were built and evaluated

Data Collection

- SpaceX REST API
  - Data about Launches, including information about the rocket used, payload delivered, launch specifications, landing specifications and landing outcome.
- Wikipedia web scrapping
  - Falcon 9 historical launch records are extracted using beautiful soup from Wikipedia page titled “List of Falcon9 an Falcon Heavy launches.”

Data Collection – SpaceX API

1. Make a GET request to the SpaceX REST API
2. Convert the response to .Jason file and normalize to Pandas Data frame
3. Clean Data using custom function
4. Assign list to dictionary and covert to data frame
5. Filter and export data frame to .csv file

- GitHub URL of the completed SpaceX API calls notebook.

1
```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

2
```
response.json()
data=pd.json_normalize(response.json())
```

3
```
# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4
```
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}

data = pd.DataFrame(launch_dict)
```

5
```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']

data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

1. Get Response from HTML
2. Create BeautifulSoup Object
3. Find Tables and Get Column names
4. Creation of dictionary
5. Append data to keys
6. Convert dictionary to dataframe
7. Saving dataframe to .csv

- GitHub URL of the completed web scraping notebook

1
```
r=requests.get(static_url)
```

2
```
soup = BeautifulSoup(r.text, "html.parser")
```

3
```
html_tables = soup.find_all("table")
column_names = []

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

4
```
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']=[]
```

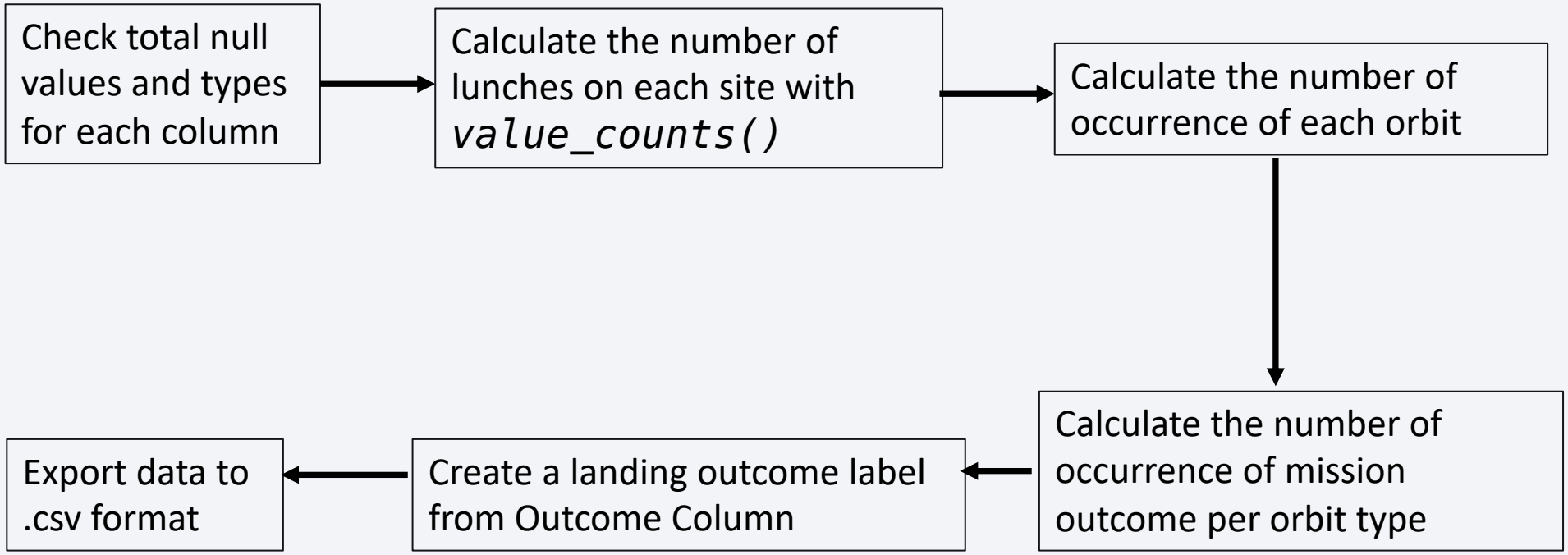
5
```
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
```

6
```
df=pd.DataFrame(launch_dict)
```

7
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

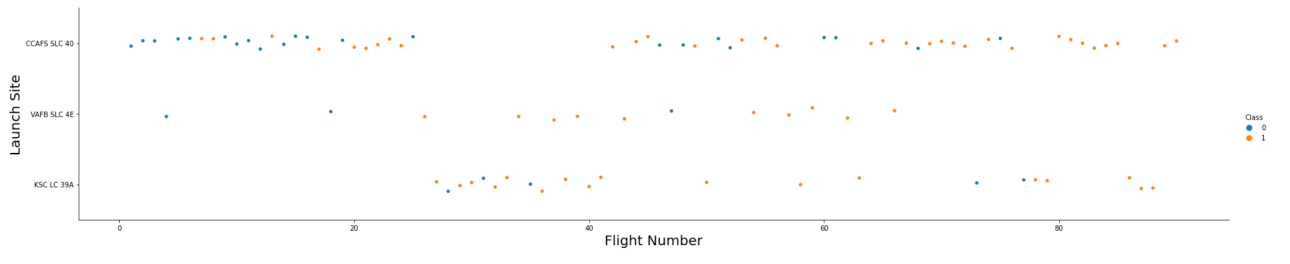
Data Wrangling

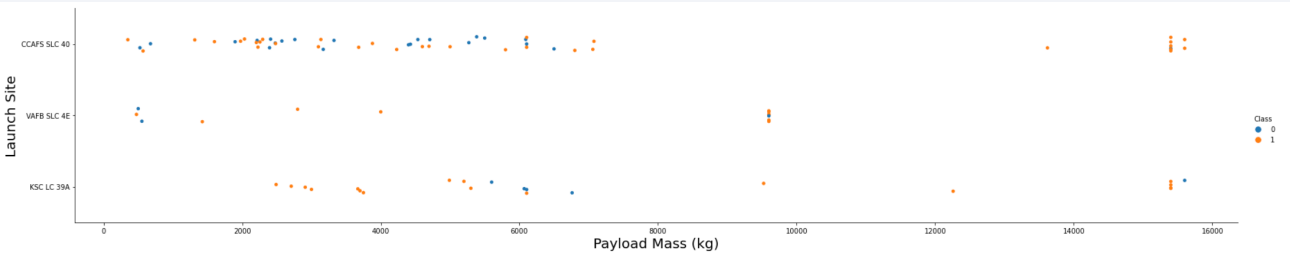
- GitHub URL of your completed data wrangling

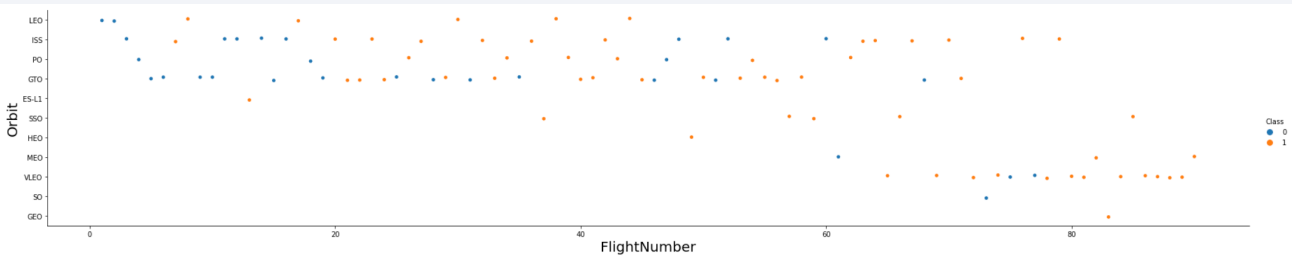


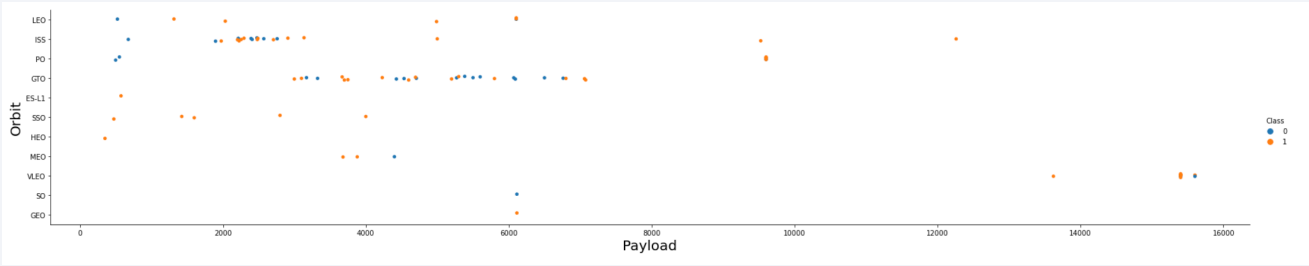
EDA with Data Visualization

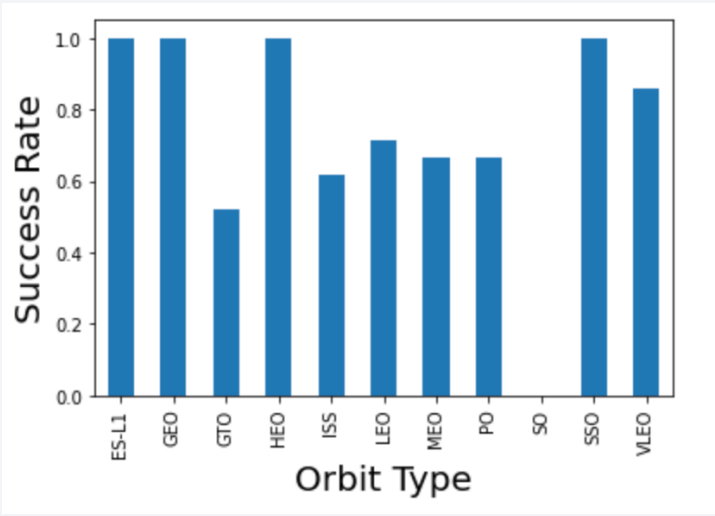
- Add the GitHub URL of your completed EDA with data visualization notebook

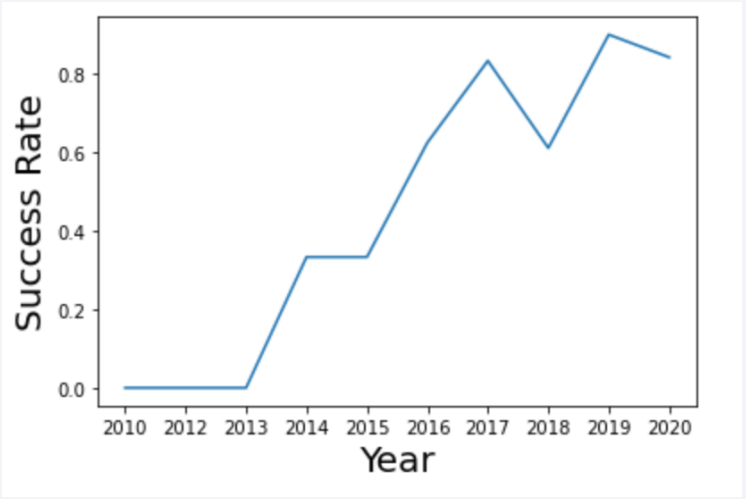








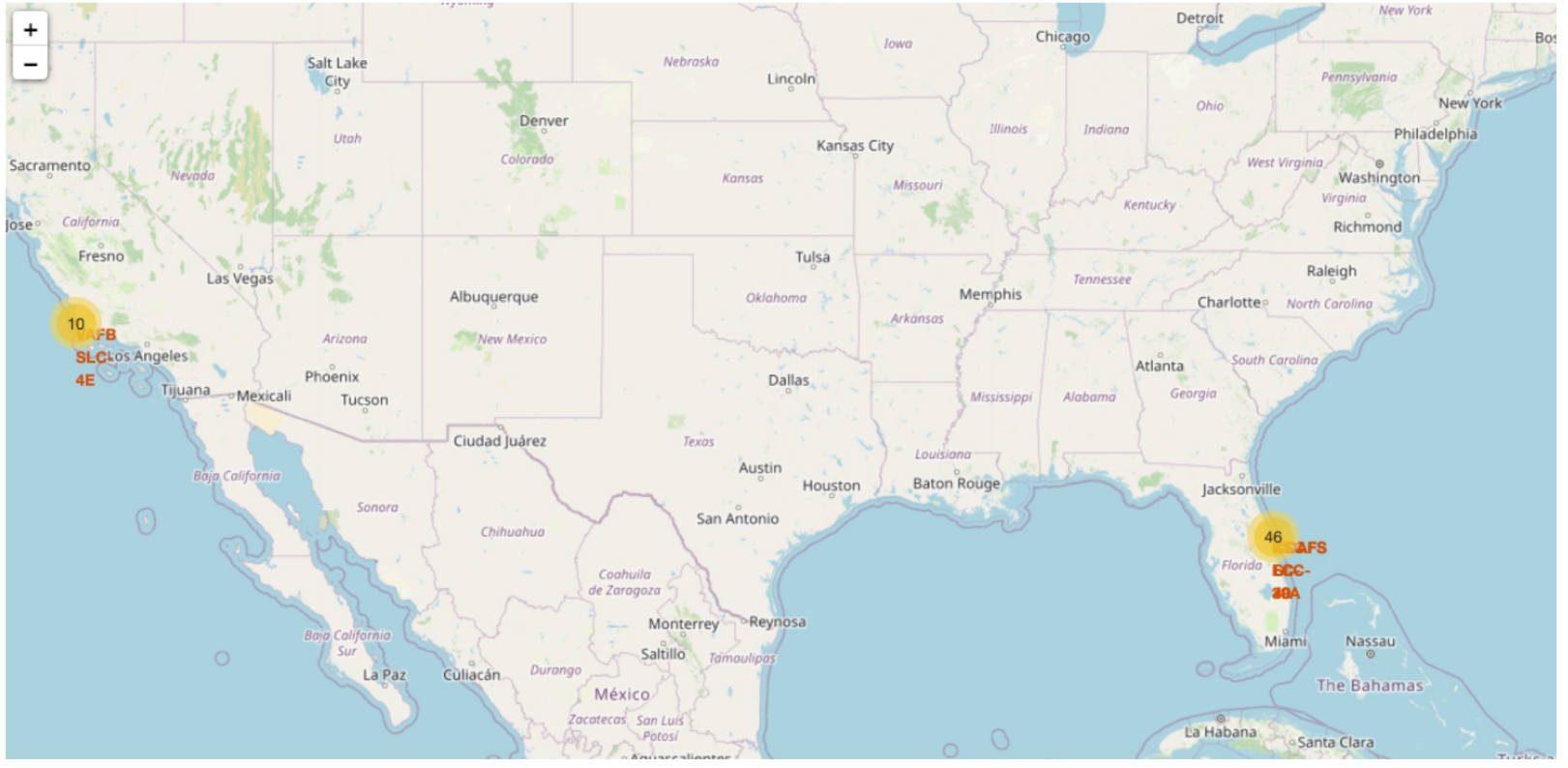




EDA with SQL

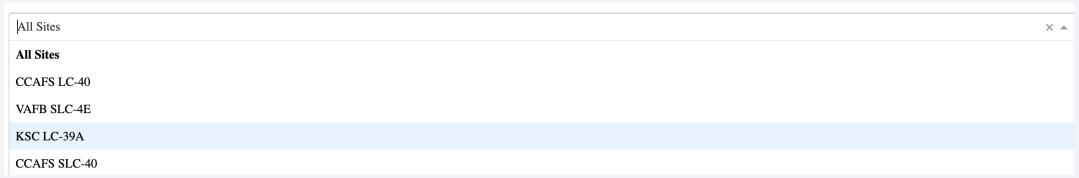
- Summary of the SQL queries
  - Display the names of the unique launch sites in the space mission
  - Display 5 records with launch sites beginning with string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date with successful landing outcome in the drone ship
  - List the names of booster with success in ground pad and 4000<payload mass<6000 kg
  - List the total number of failure and successful mission outcomes
  - List the names of booster version which carried maximum payload mass with subquery
  - List the records displaying month names, successful landing outcomes in ground pad, booster version, launch sites for months in year 2015
  - Rank the count of successful landing outcomes between 2016/06/04 to 2017/03/20 in descending order
- The GitHub URL of your completed EDA

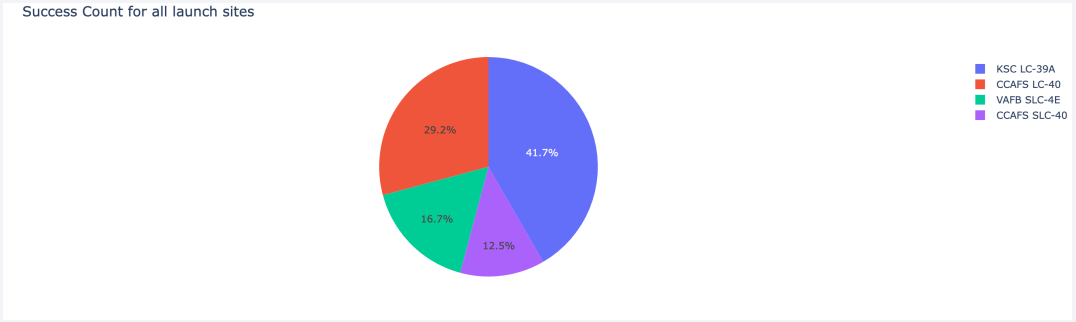
Build an Interactive Map with Folium

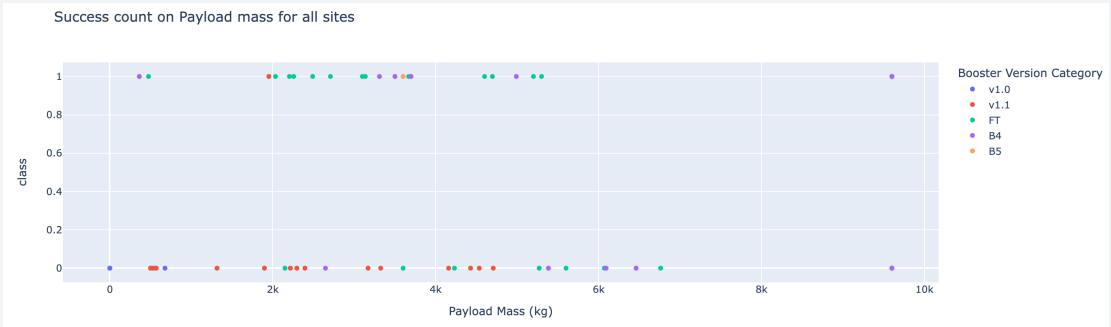


- Mark all launch sites (Circle)
- Mark success and failure of launches (Marker Color)
- Calculate distance between launch sites to its proximities (Lines)
- The GitHub URL of completed interactive map with Folium map

Build a Dashboard with Plotly Dash







- Pie Chart with drop down menu for launch sites:
  - Shows the successful launches for sites.
- Scatter Plot with range slider to choose the payload mass:
  - Correlation between payload and outcomes
- GitHub URL of your completed Plotly Dash lab

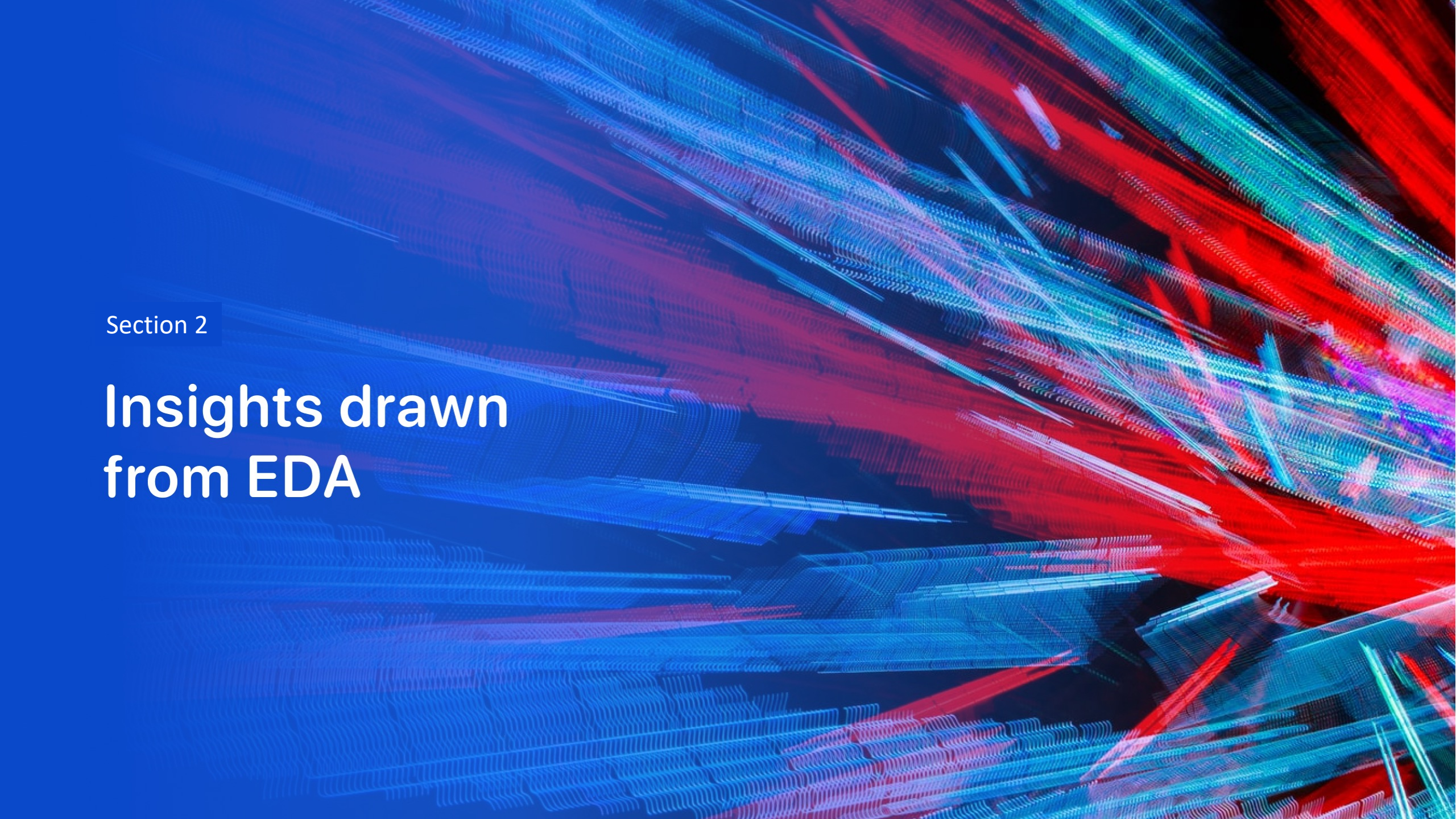
Predictive Analysis (Classification)

1. Model Development
   - Data Transformation
   - Splitting into training and testing sets
   - Choose ML algorithms (Logistic regression, SVM, Decision Tree and KNN) and use training data to train the model
2. Model Evaluation
   - Use test data set to evaluate model (score, confusion matrix)
3. Find the best model comparing the scores for different algorithms

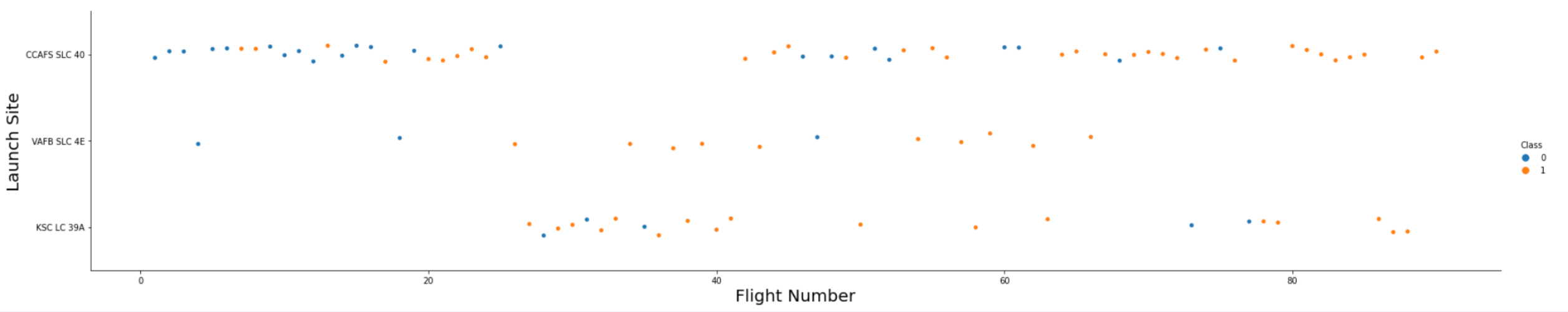
- GitHub URL of completed predictive analysis lab

Results

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

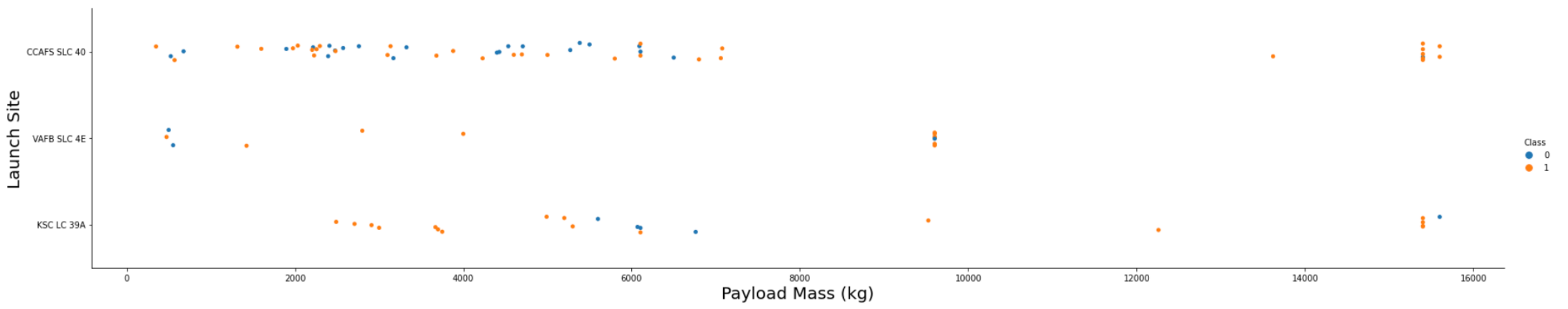

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- CCAFS SLC 40 has higher number of flights, VAFB SLC 4E has least number of flights and KSC LC 39A has no earlier flights
- For a launch site, the success rate of flight increases as the flight number increases

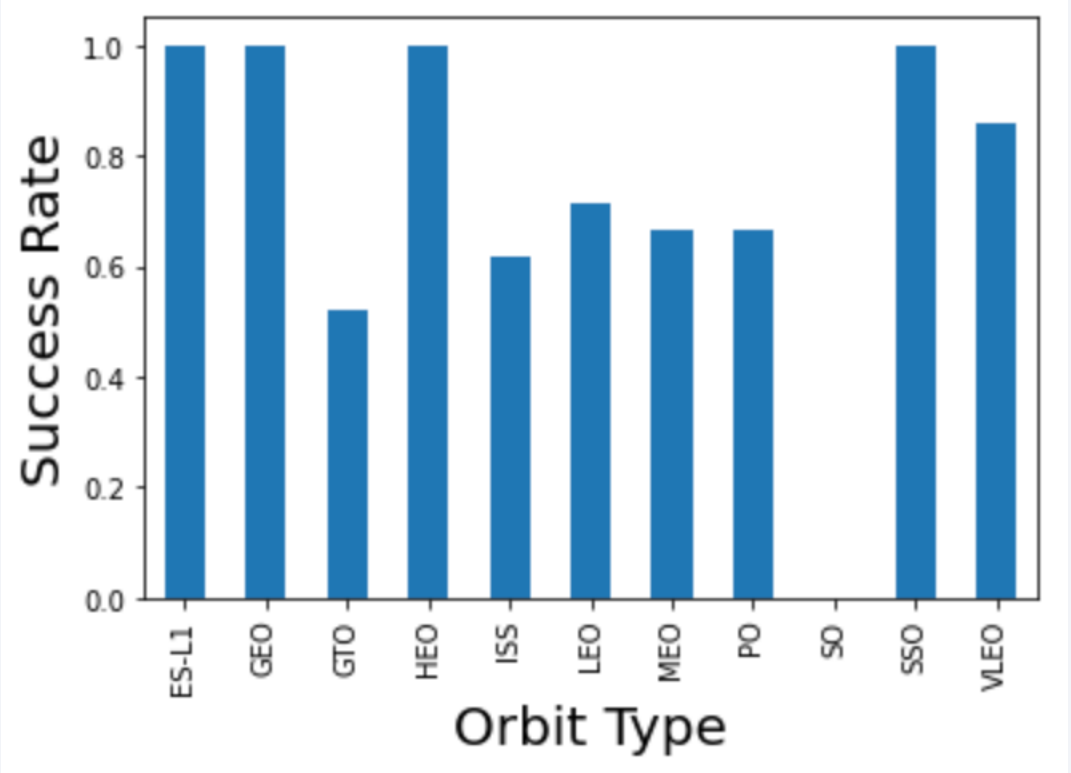
Payload vs. Launch Site



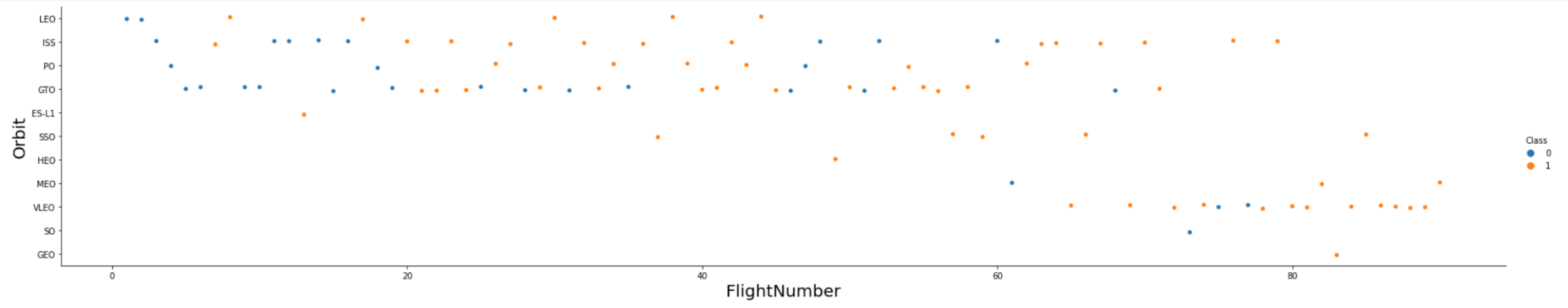
- Most of the flights are with pay load mass smaller than 8000 kg
- No clear correlation between payload mass and successful flights

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO has highest (100 %) success rate
- SO has lowest (0%) success rate.

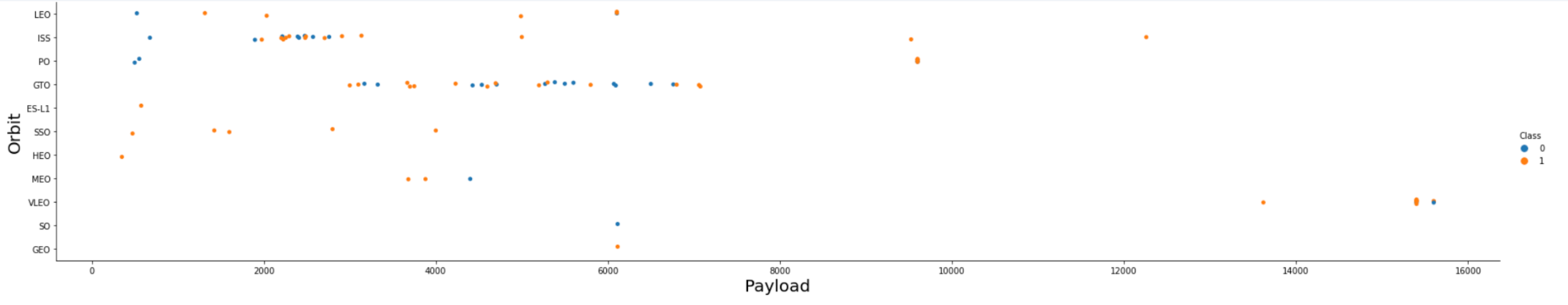


Flight Number vs. Orbit Type



- The LEO orbit type shows that the success of flight is related with flight number.
- The 100 % success rate of ES-L1 HEO and GEO can attributed to the fact that they had only one flight

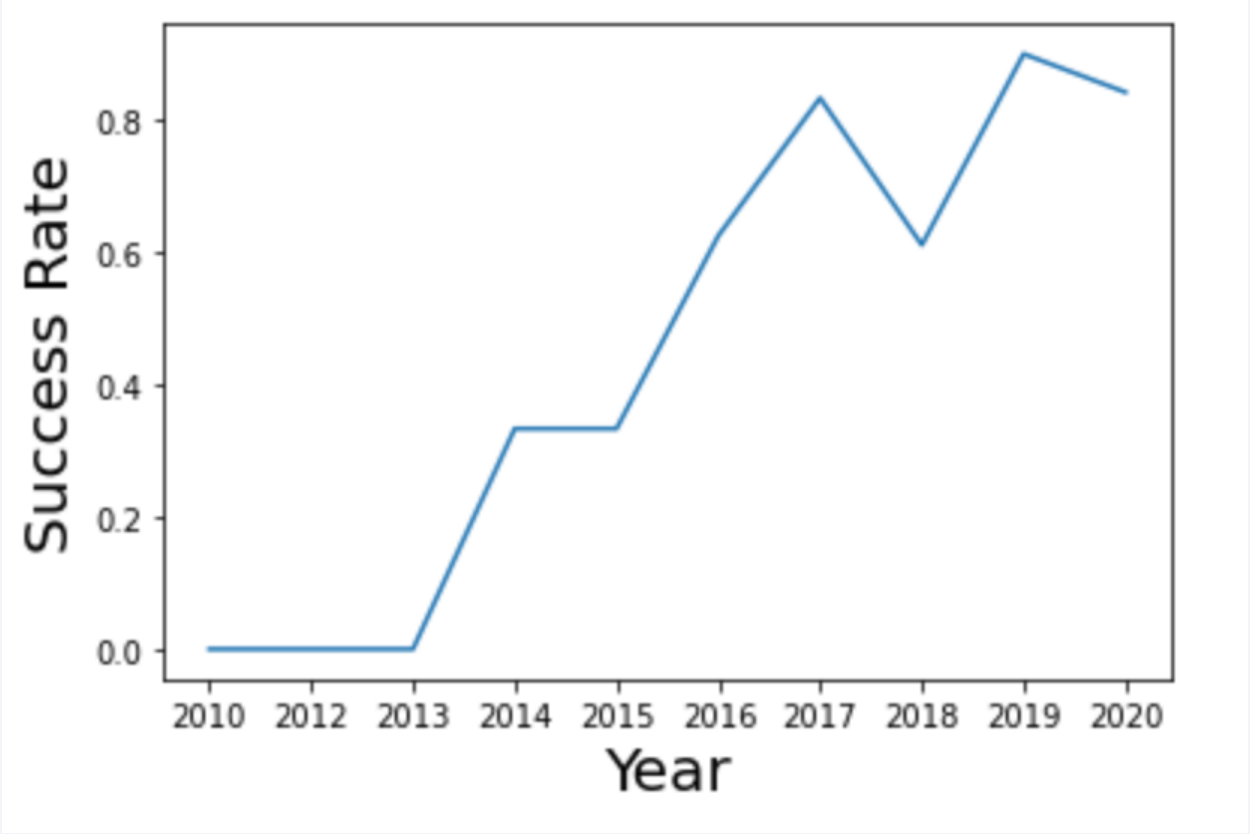
Payload vs. Orbit Type



- LEO and ISS seems to have higher success rate with larger payload mass.
- GTO orbit doesn't show any clear relation between payload mass and successful landing

Launch Success Yearly Trend

- No successful landings in 2010 - 2013
- After 2013, overall success rate is gradually increasing with year except
  - Plateau 2014-2015
  - 2018 dip
- The success rate is more than 50 % after year 2016



All Launch Site Names

%%sql SELECT DISTINCT LAUNCH_SITE **FROM** SPACEXTBL;

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

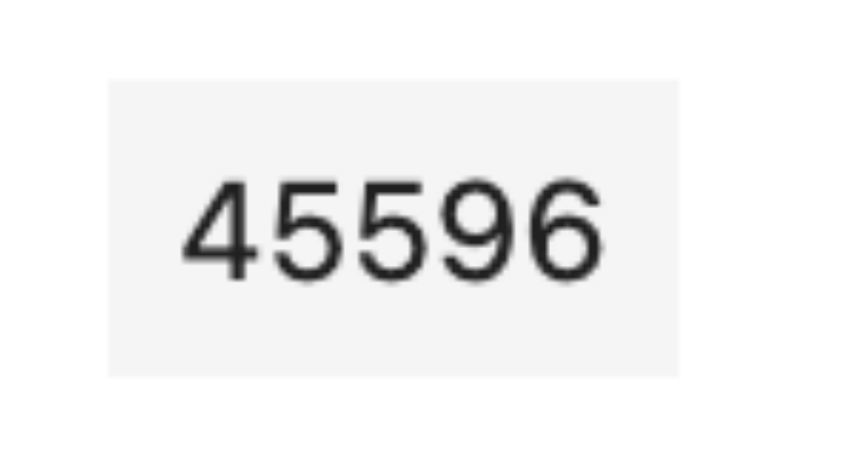
Launch Site Names Begin with 'CCA'

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

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

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



Average Payload Mass by F9 v1.1

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.0%';
```

First Successful Ground Landing Date

```
%%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing__Outcome = 'Success (ground pad)';
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

booster_version
F9 FT B1021.1
F9 FT B1023.1
F9 FT B1029.2
F9 FT B1038.1
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) 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

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

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

2015 Launch Records

%%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

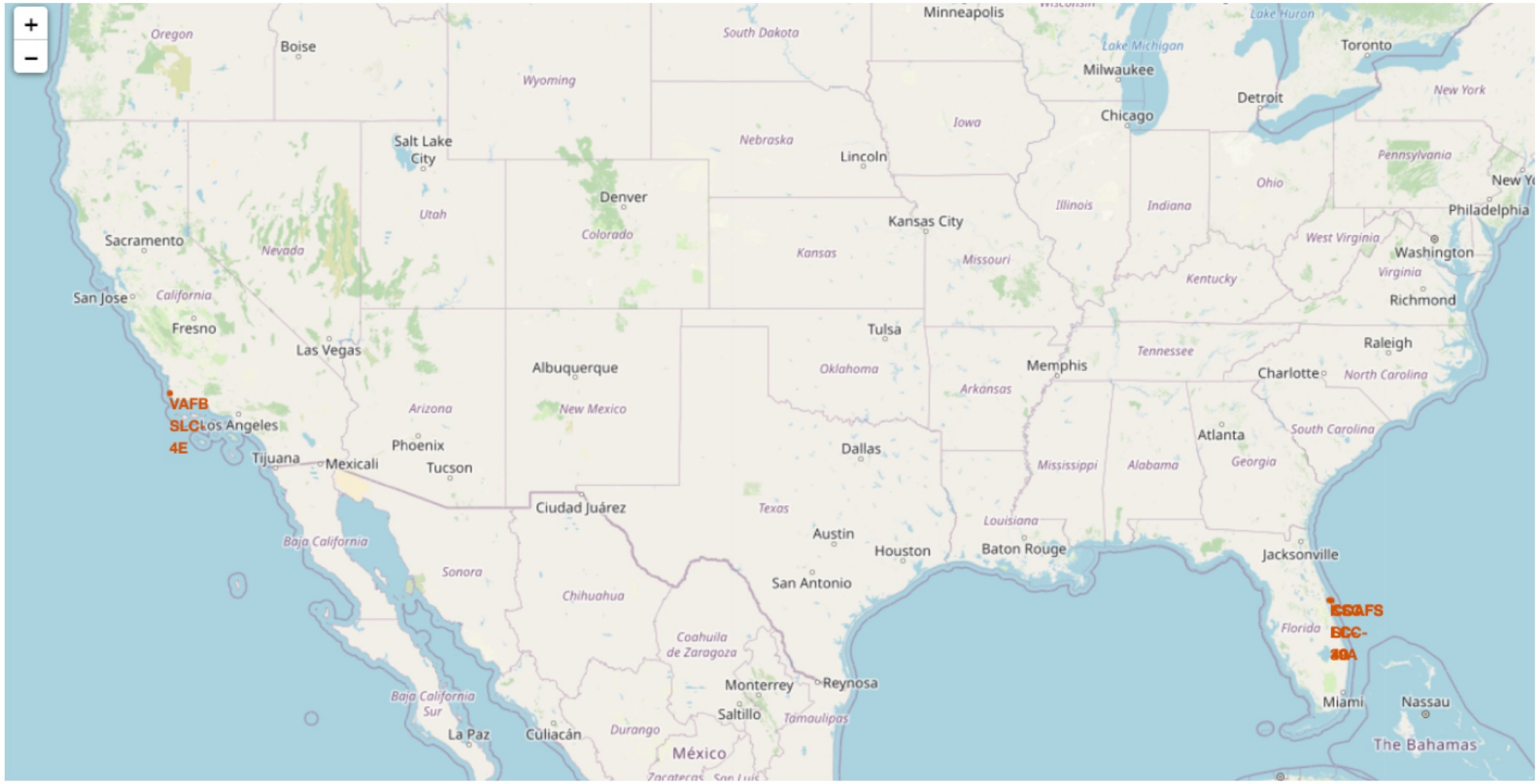
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

%%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

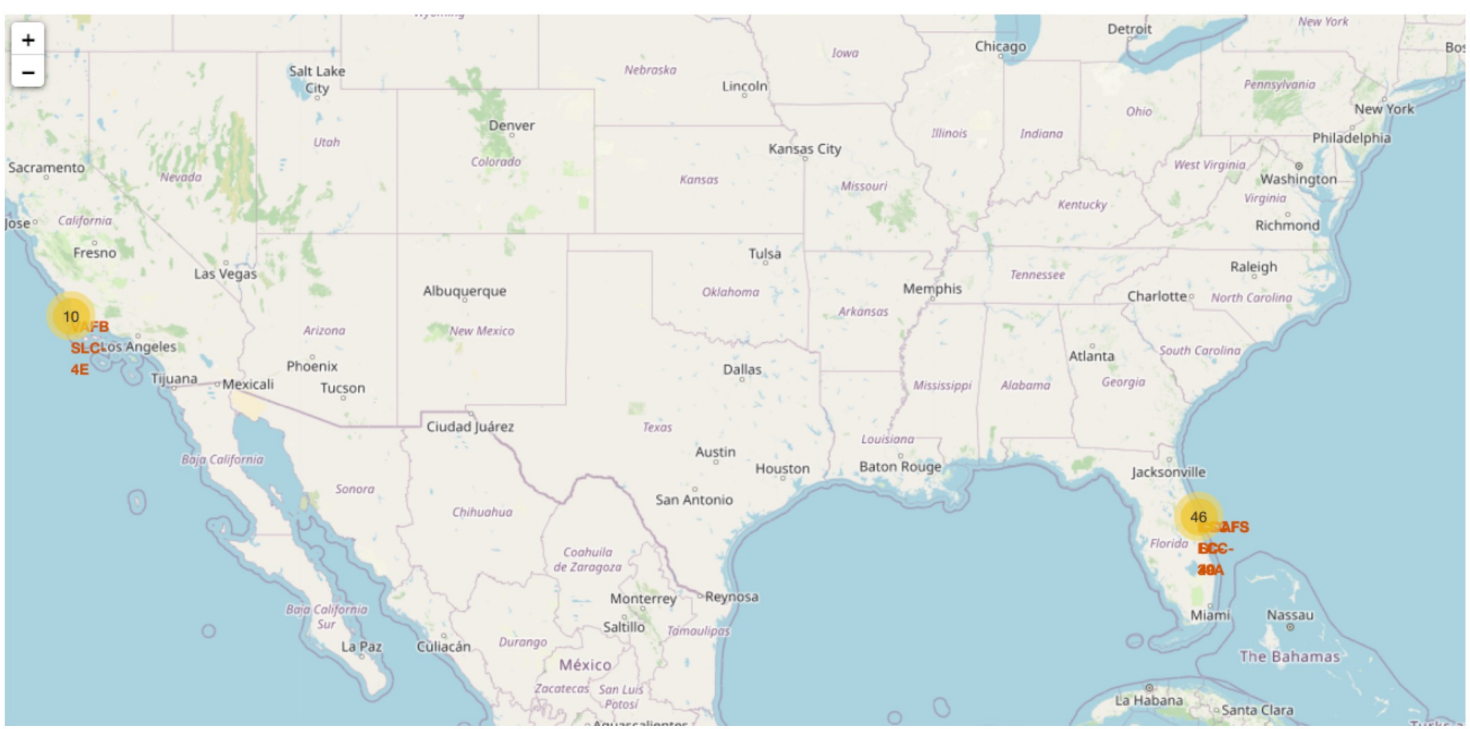

Section 3
Launch Sites
Proximities Analysis

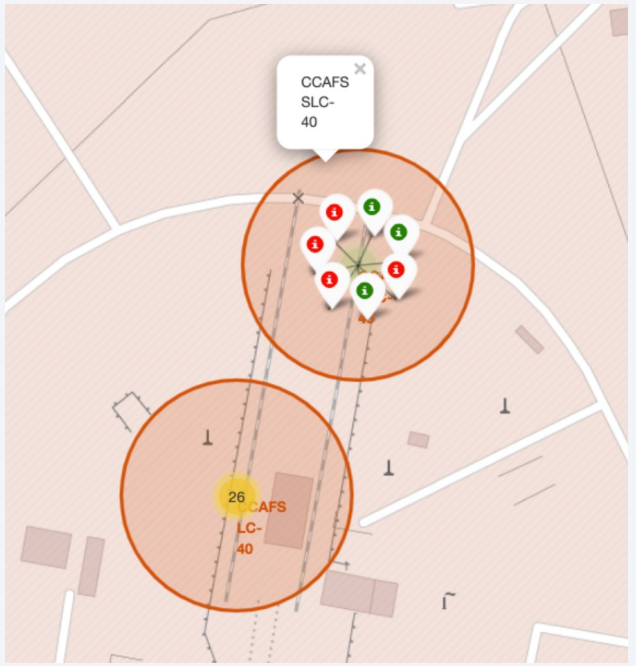
All Launch Sites



All launch sites are in costal regions (California and Florida) of United States.

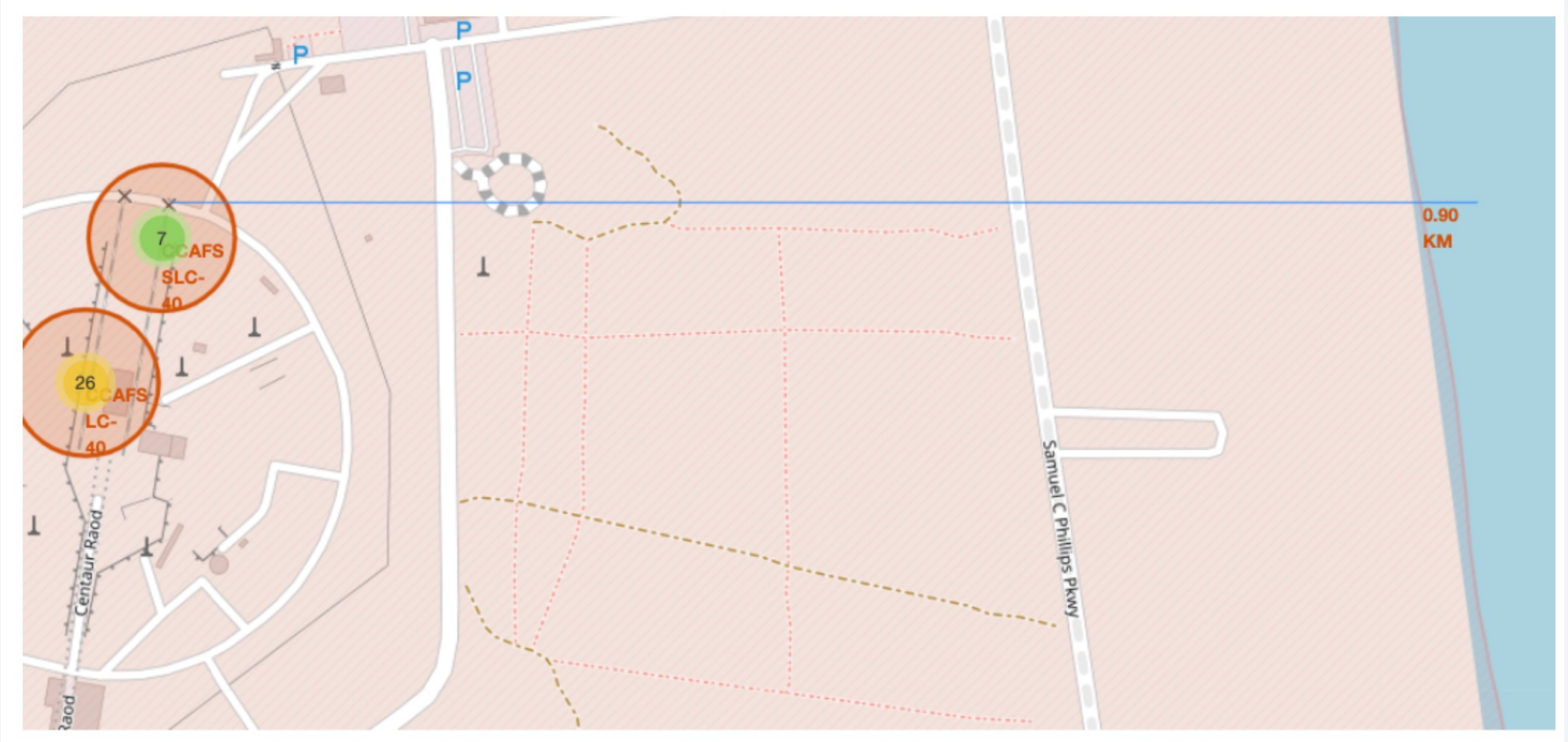
Success and Failed Launches



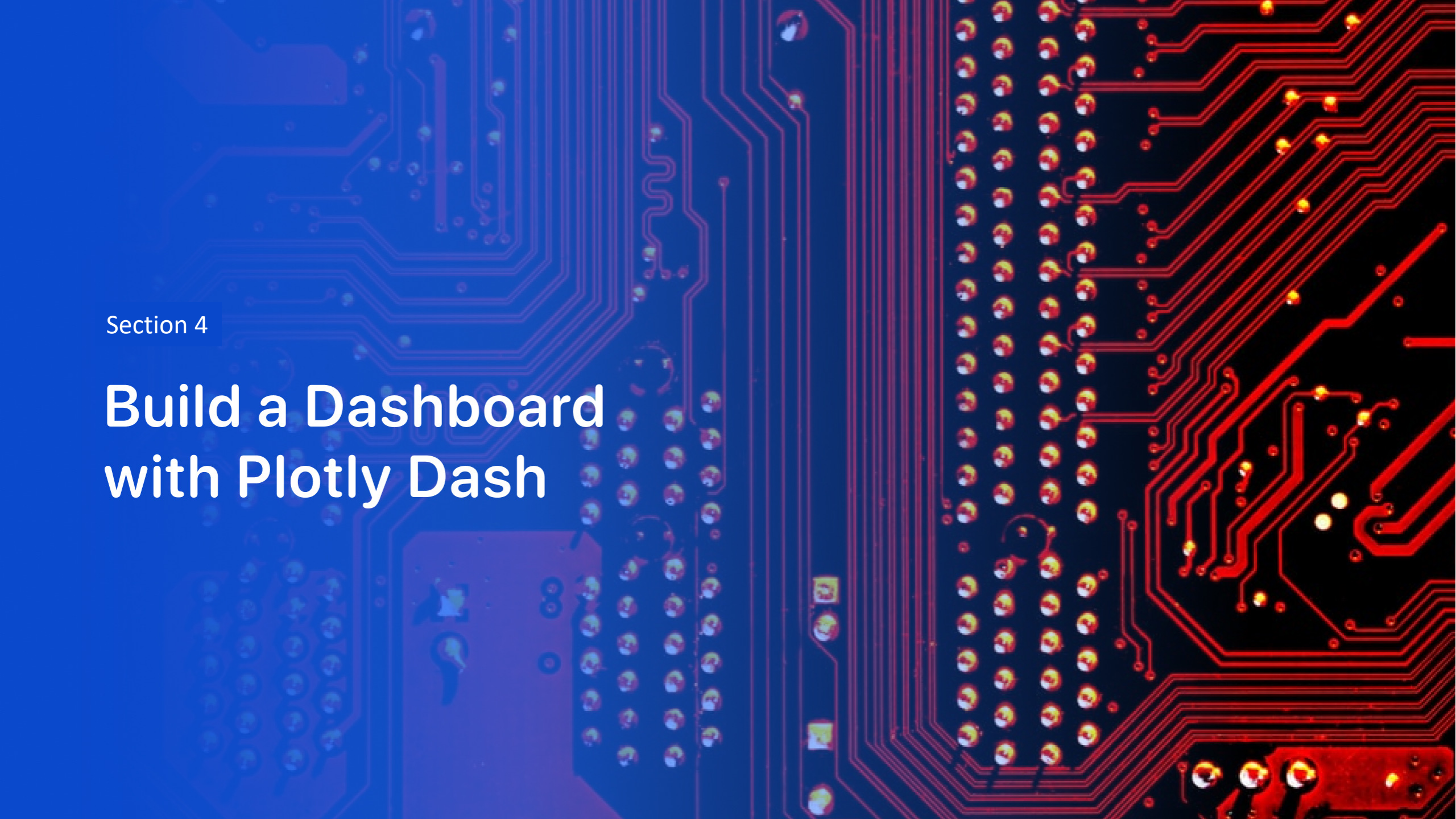


Launches are grouped into clusters; green color showing success and red color showing failed flights.

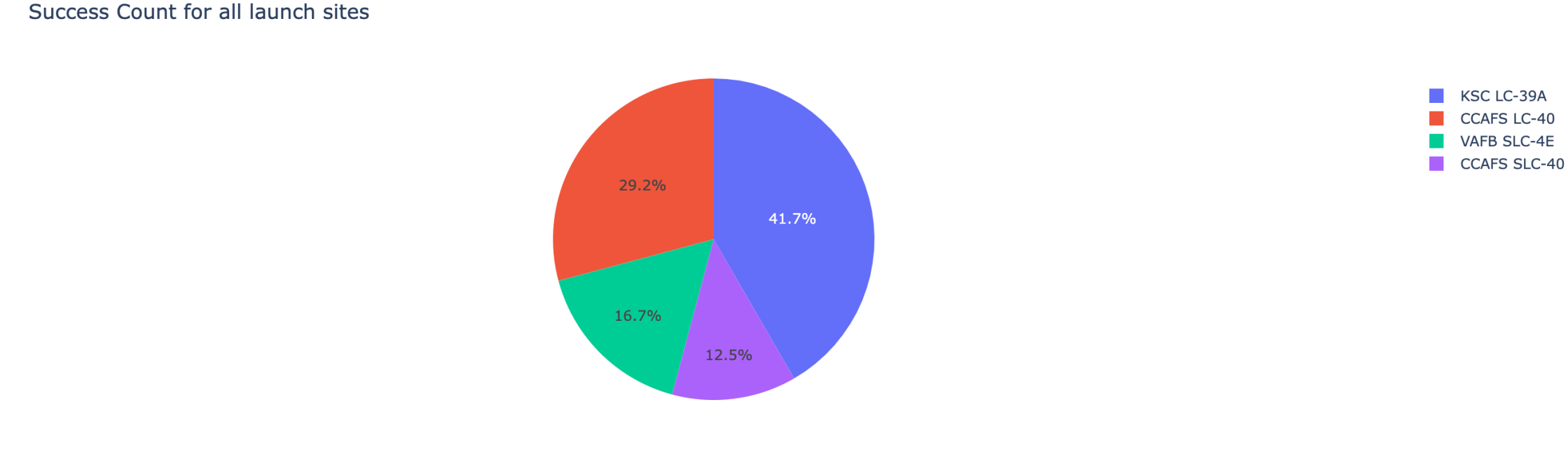
Near Launch Sites



- CCAFS SLC-40 and its nearby places are shown as example
  - 0.9 km from coastline

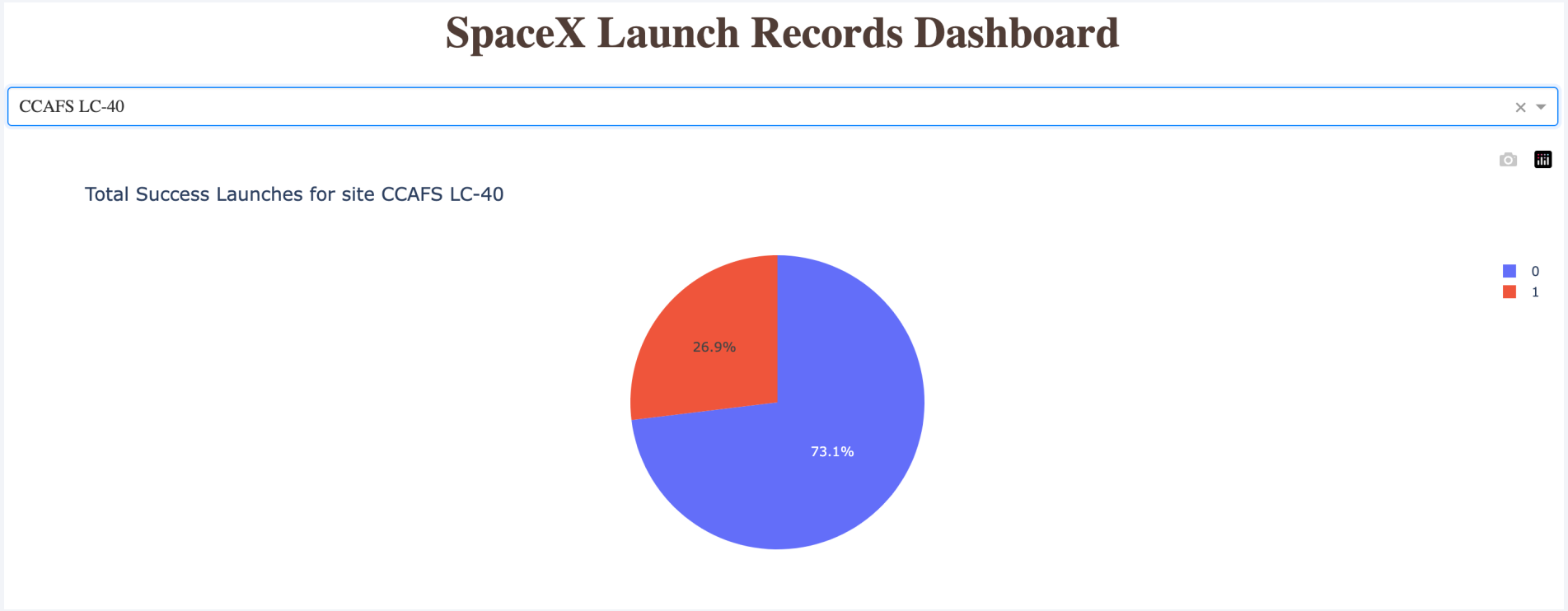

Section 4
Build a Dashboard with Plotly Dash

Success Count for all Launch Sites



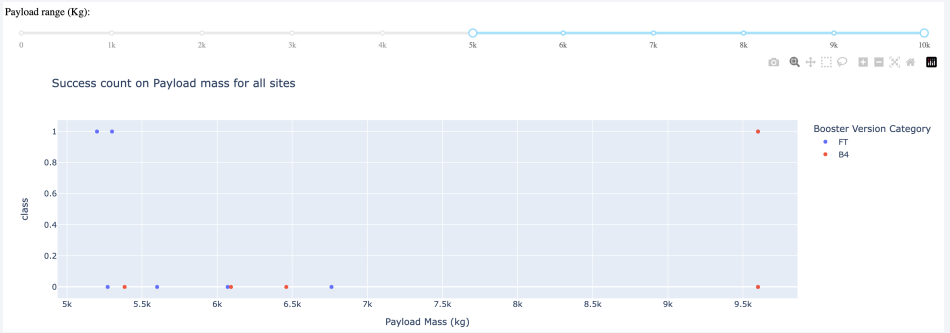
KSC LC-39A has the highest successful launch with 41. 7 %

Launch Site with highest Success Rate



CCAFS LC-40 has highest success rate of 73.1 %

Payload vs. Launch Outcome

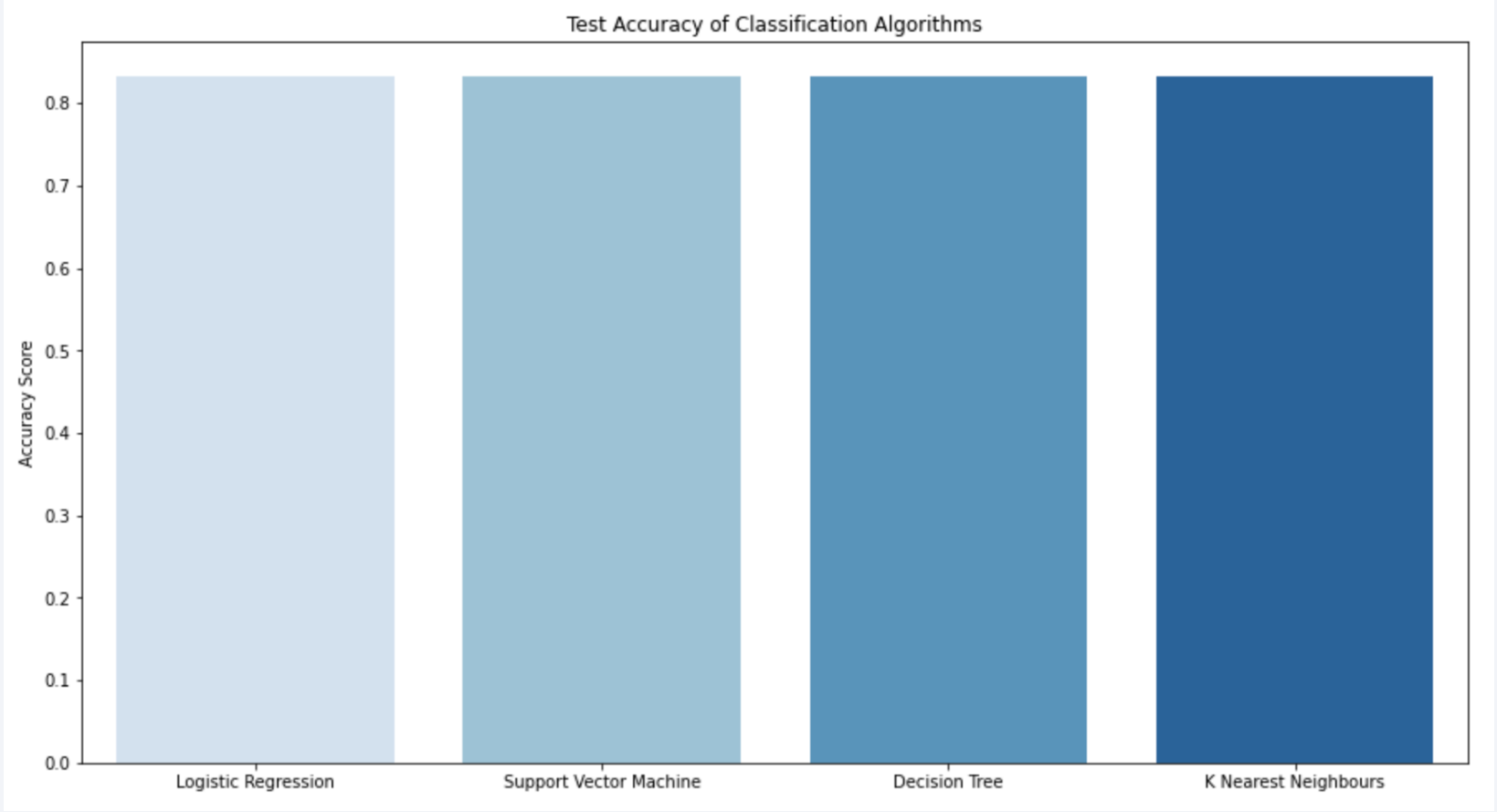


- Higher mass range: few success rate
- Lower mass range: roughly similar success and failure rate

Section 5

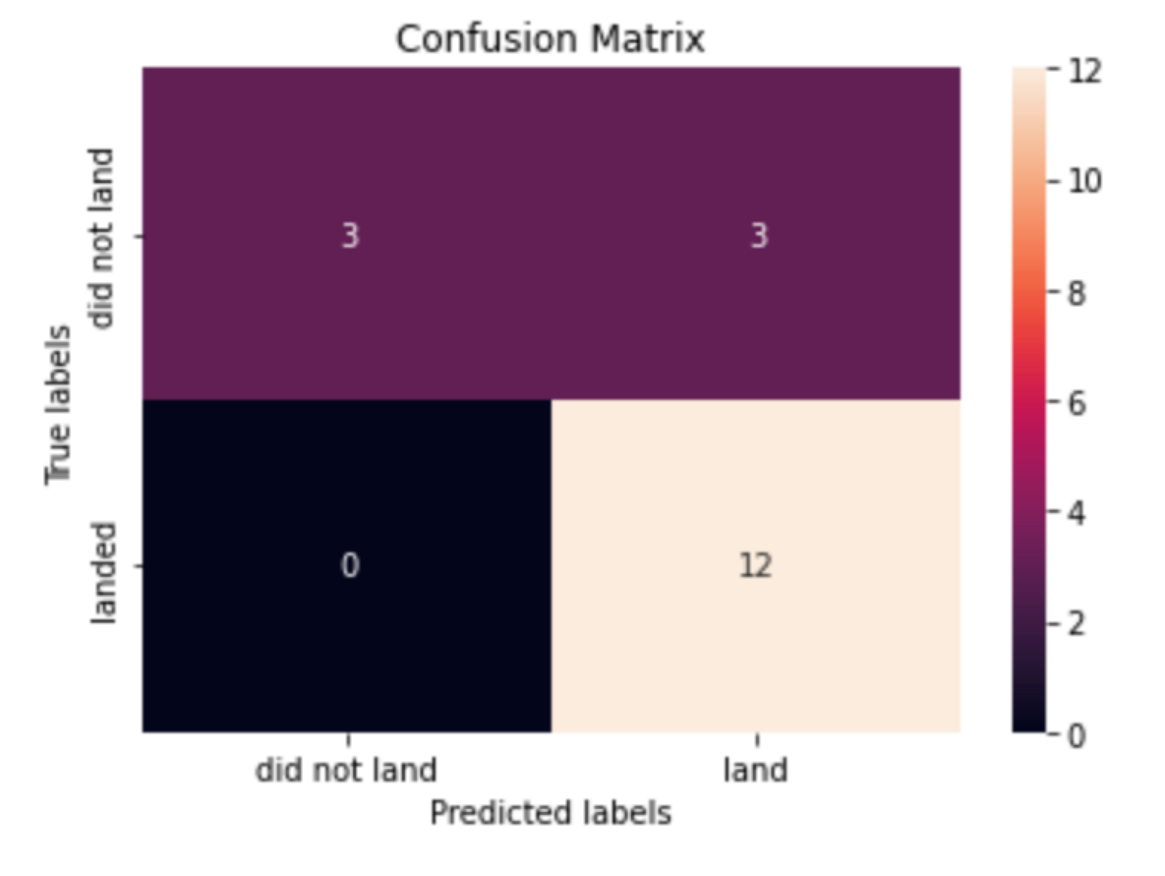
Predictive Analysis (Classification)

Classification Accuracy



All model has identical accuracy

Confusion Matrix



Although the training set accuracies are slightly different, the test accuracies are same for all the models

Conclusions

- For a launch site, the success rate of flight increases as the flight number increases
- ES-L1, GEO, HEO and SSO orbits have highest (100 %) success rate
- The success rate of flight has been increasing gradually with years
- CCAFS LC-40 launch site has highest success rate of 73.1 %
- The Logistic regression, SVM, KNN and Decision Tree all have same prediction accuracy for the test set

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!