Winning Space Race with Data Science

Daniel Gomez
10/23/21

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection and data wrangling
  - EDA and interactive visual analytics
  - Predictive analysis
- Summary of all results
  - EDA with Visualization
  - EDA with SQL
  - Interactive map with Folium
  - Plotly Dash dashboard
  - Predictive analysis (classification)

Introduction

- **Project Background and Context**
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX 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 for SpaceY which is an alternate company that wants to compete with SpaceX.
- Problems to be solved
  - What determines whether a rocket lands successfully or not?
  - What variables determine a rockets success rate?
  - What conditions must be met to achieve the best results for a successful landing?

Section 1

Methodology

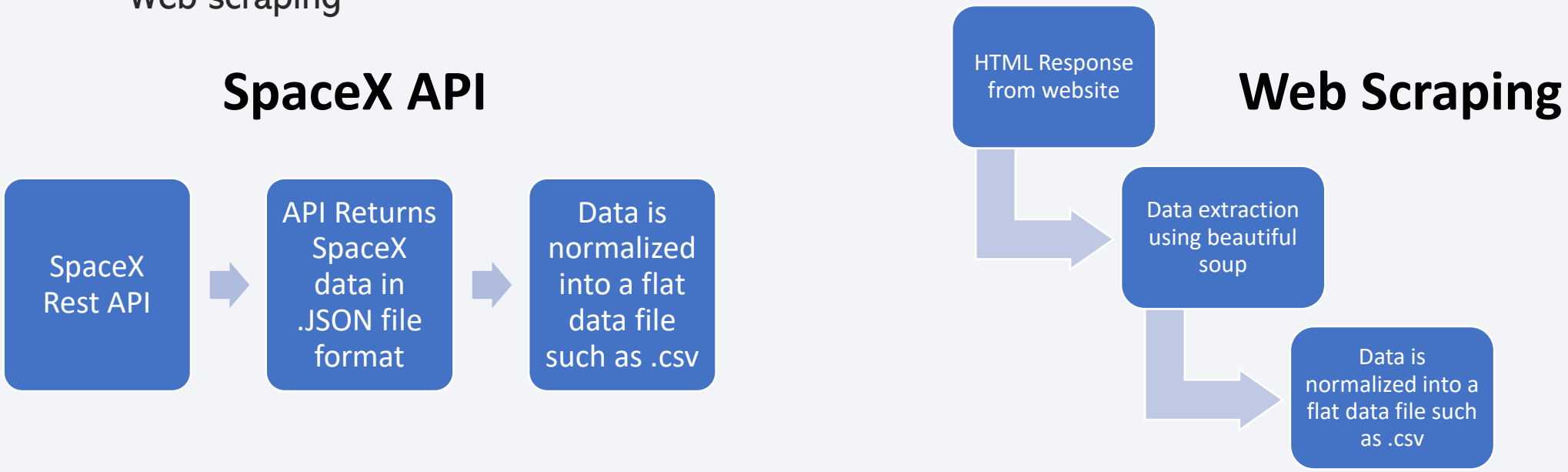
Methodology

Executive Summary

- Data collection methodology:
  - Web Scrapping
  - SpaceX Rest API
- Perform data wrangling
  - Dropping unnecessary data and hot encoding data fields 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

- Datasets were collected by:
  - Working with the provided SpaceX launch data from the SpaceX Rest API
    - Data includes information on the rocket used, payload delivered, launch specifications, and landing outcomes
    - The data allows for the successful prediction on whether SpaceX will attempt to land a rocket.
  - Web scraping



Data Collection – SpaceX API

1. Get Response from API

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

response = requests.get(spacex_url)
```

2. Convert Response to a .Json file

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

3. Clean Data

```
# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Assign List to Dictionary

```
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}
```

5. Assign Dictionary to Database

```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

6. Filter Dataframe

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

7. Export to .csv

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

[GitHub URL to Notebook]

Data Collection - Scraping

1. Get HTML Response

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

2. Create Beautiful Soup Object

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

3. Find Tables

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

4. Get Column Names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Create a Dictionary

```
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']=[]
```

6. Append 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"):
```

7. Convert Dictionary to Dataframe

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

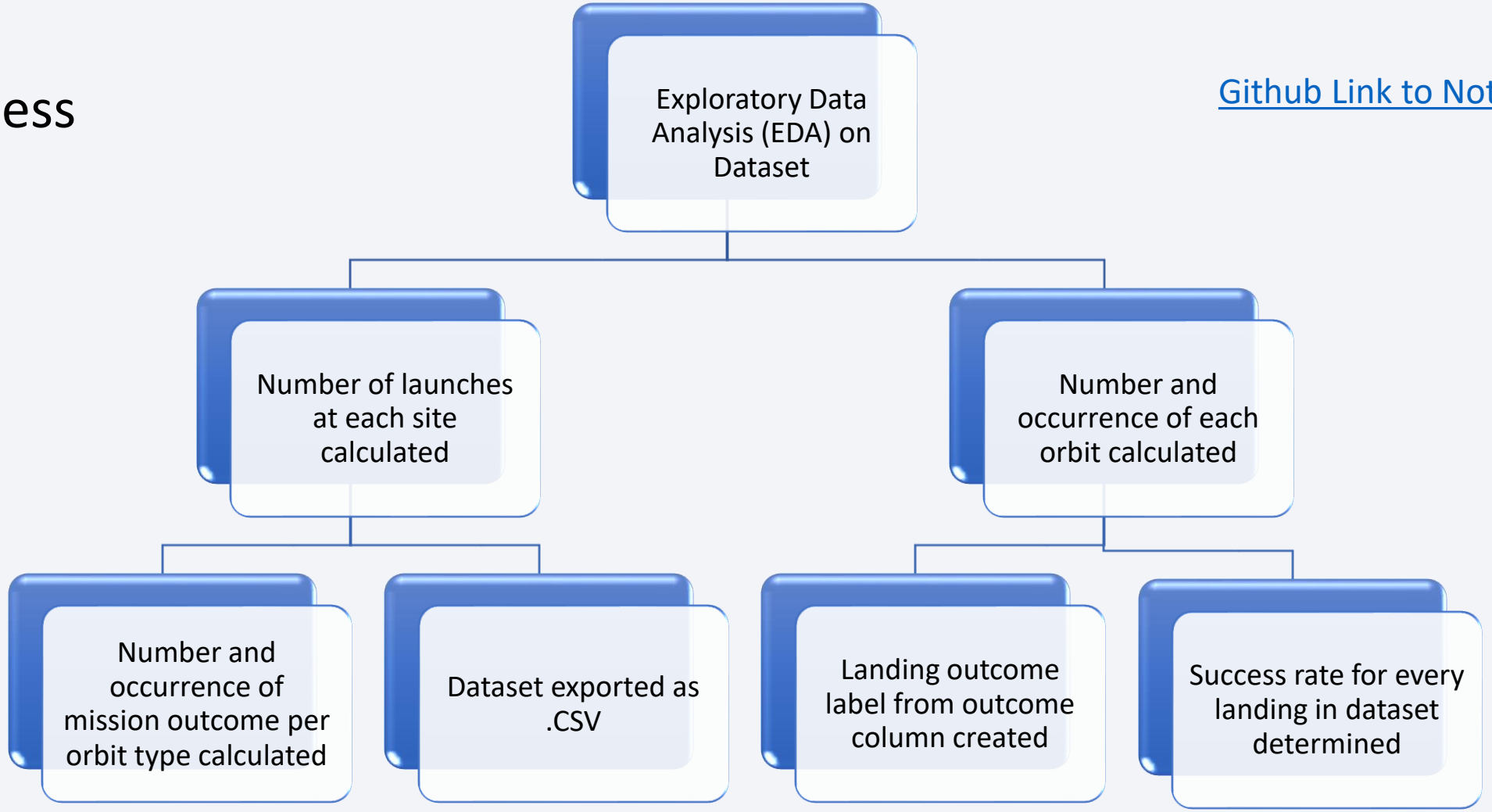
8. Convert to .csv

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

[Github URL to Notebook]

Data Wrangling

Process



Github Link to Notebook

EDA with Data Visualization

- Scatter graphs, a bar graph, and a Line graph were all used.
- Scatter plots usually contain large amounts of data and are used to show how much one variable is affected by another.
- The following scatter plots were used:
    - Flight Number vs. Payload Mass
    - Flight Number vs. Launch Site
    - Payload vs. Launch Site
    - Orbit vs. Flight Number
    - Payload vs. Orbit Type
    - Orbit vs. Payload Mass
- Line Graphs were used to show trends and to make predictions with the available data.
    - Mean vs. Orbit bar graph was used
- A bar Graph was used to compare different groups together and to show any big changes in data over time.
    - Success Rate vs. Year line graph was used

Github Link to Notebook

EDA with SQL

- SQL Queries Performed:
  - Displaying names of unique launch sites
  - Displaying 5 records with specific launch site strings such as ‘KSC’
  - Displaying total payload mass by specific boosters
  - Displaying average payload mass carried by booster version
  - Listing Dates of successful landing outcomes
  - Listing the names of boosters with success in ground pad and payload masses between 4000 and 6000
  - Listing the total number of successful or failure mission outcomes
  - Listing the names of the booster versions that have carried the maximum payload mass
  - Listing records that display month names, successful landing outcomes in ground pad, booster versions, and launch site for the months in the year 2017
  - Ranking the successful landing outcomes between 2010-06-04 and 2017-03-20 in descending order.

Github Link to Notebook

Build an Interactive Map with Folium

- Lines were used to visualize the latitude and longitude coordinates of launch sites, highways, coastlines, and cities. Circles were used to represent the launch sites.
- Red and green markers were used to represent failure launch outcomes and successful launch outcomes respectively.
- No launch sites were found to be in close proximity of railways, highways, and cities but they were found to be within close proximity from cities.

Github Link to Notebook

Build a Dashboard with Plotly Dash

- A pie chart was added to the dashboard.
  - Pie chart showed the total launches for specific launch sites and for all launch sites.
  - Displayed multiple classes of data
  - A pie chart was chosen as a good visual representation of total quantity of a specific data set compared to others.
- A scatter plot was added to the dashboard
  - A scatter plot was used to show the relationship between outcome and payload mass for different booster versions.
  - Best method to show non linear patterns
  - Maximum and minimums can easily be determined
  - Observation are straightforward

Github Link to source code

Predictive Analysis (Classification)

- Model Building
  - Dataset was loaded into NumPy and Pandas
  - Data was transformed
  - Data was split into training and test data sets
  - Amount of sample tests was checked
  - Decision on what machine learning algorithms should be used
  - Parameters and algorithms were set to GridSearchCV
  - Datasets were set into GridSearchCV objects and dataset was trained
- Model Evaluation
  - Accuracy was checked for each model
  - Hyperparameter were tuned for each algorithm type
  - Confusion matrix were plotted
- Model Improving
  - Feature engineering and Algorithm tuning
- Best Performing classification model
  - Model with the best accuracy score won as the best performing model
  - A dictionary of algorithms with scores can be found in the notebook

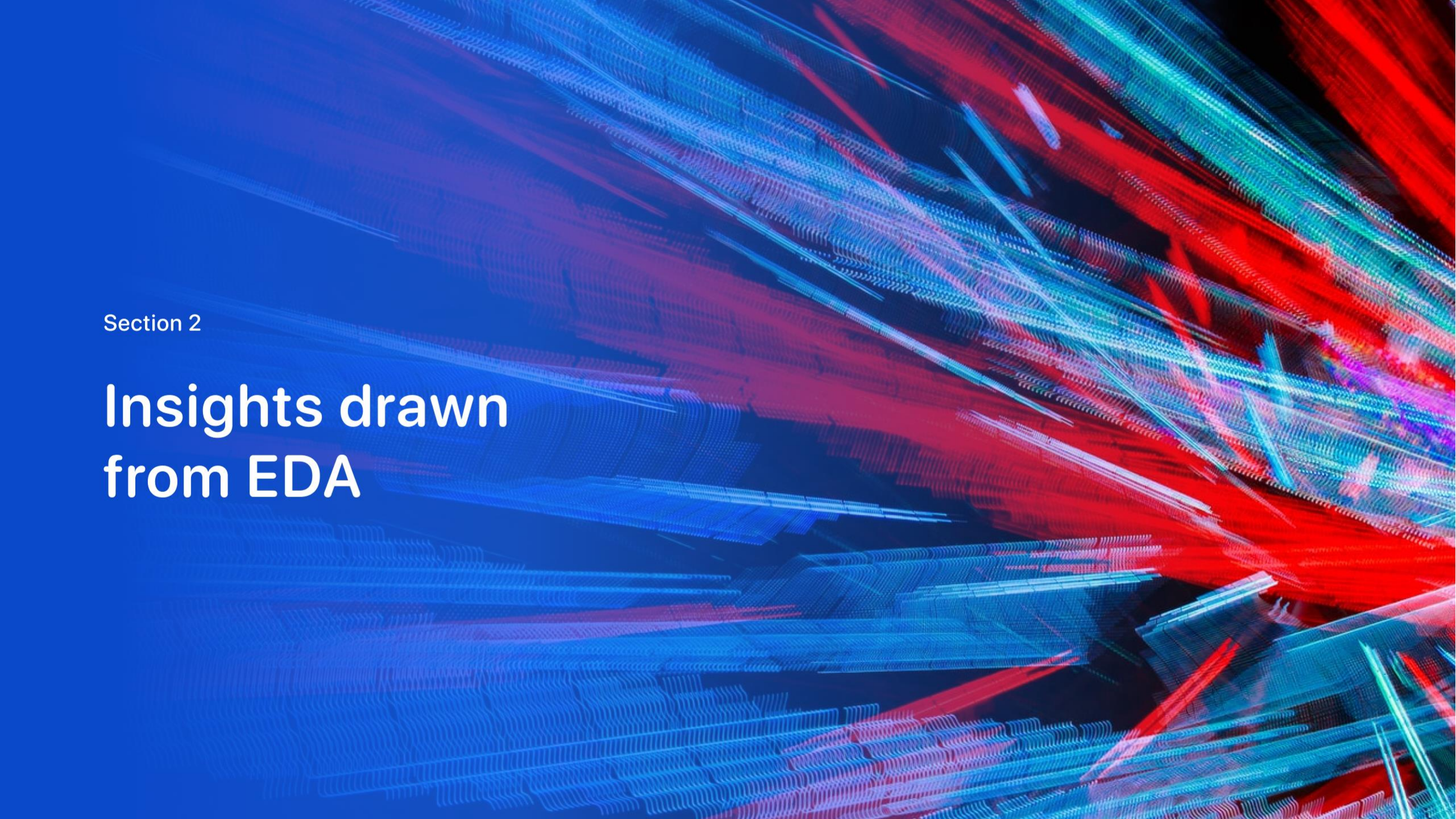
Github Link to Notebook

Results

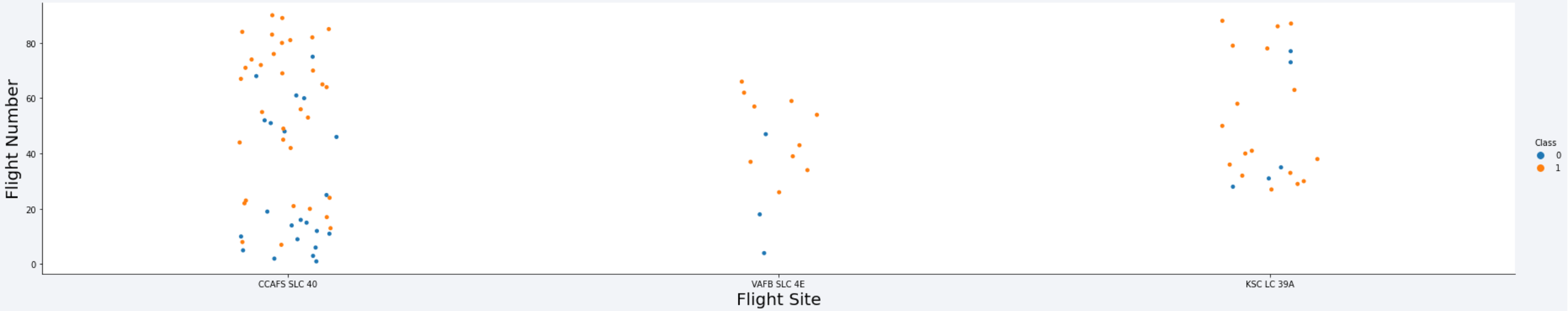
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

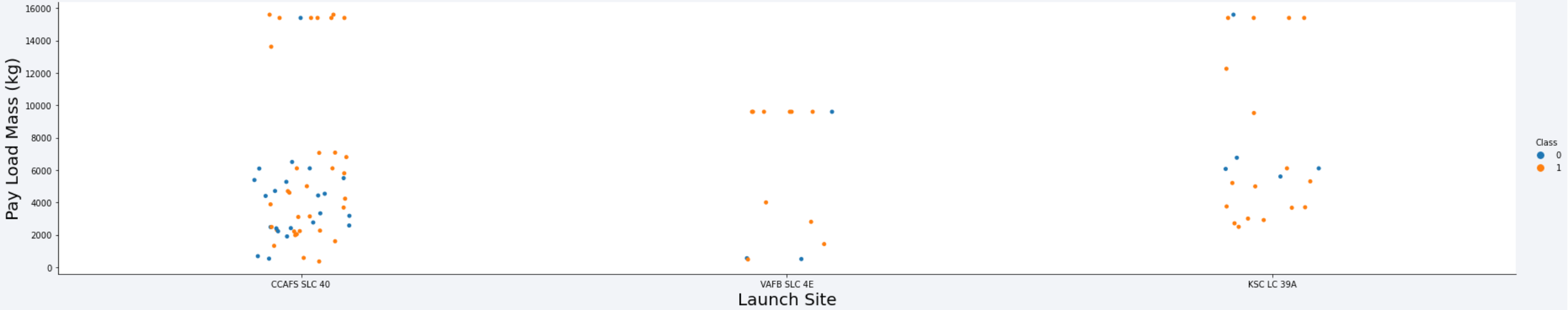


Flight Number vs. Launch Site



- The greater the amount of flights at a launch site, the greater the success rate at a launch site.

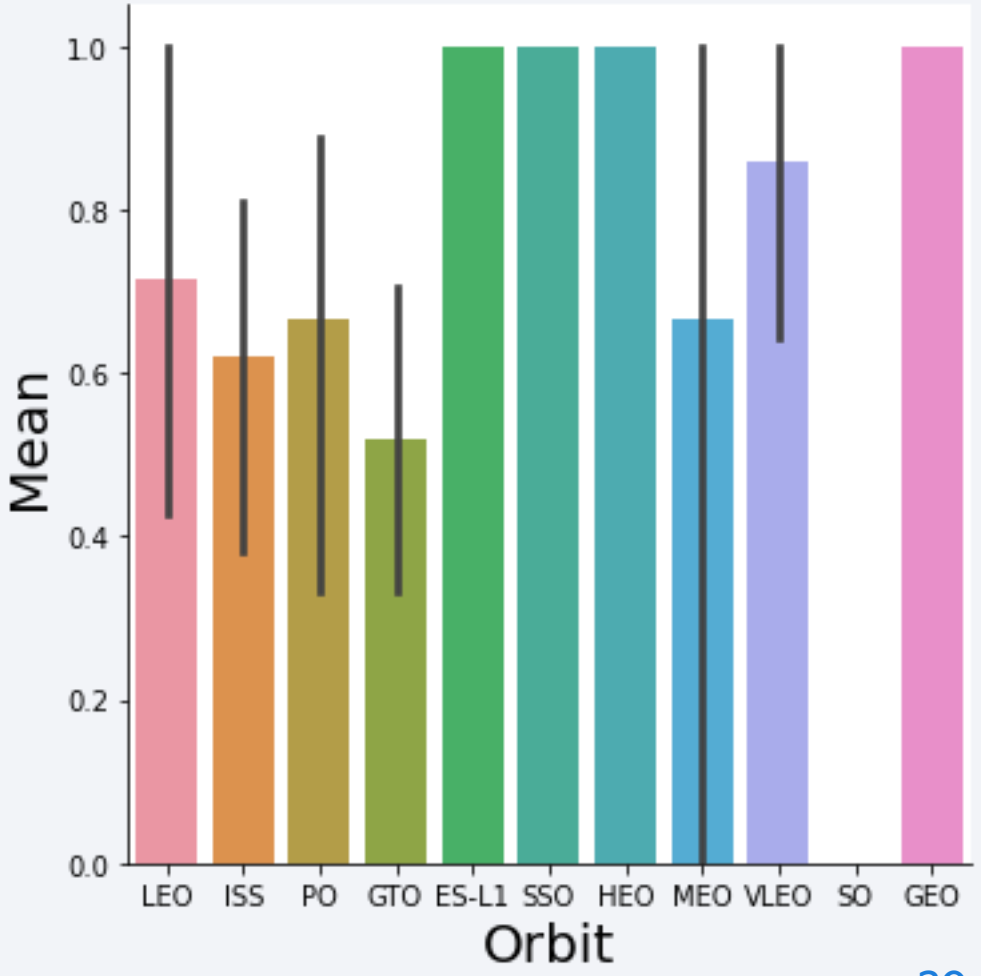
Payload vs. Launch Site



- The higher the payload mass, the more successful the launches are. There are no heavy payload launches at the VAFB SLC4E Launchsite.

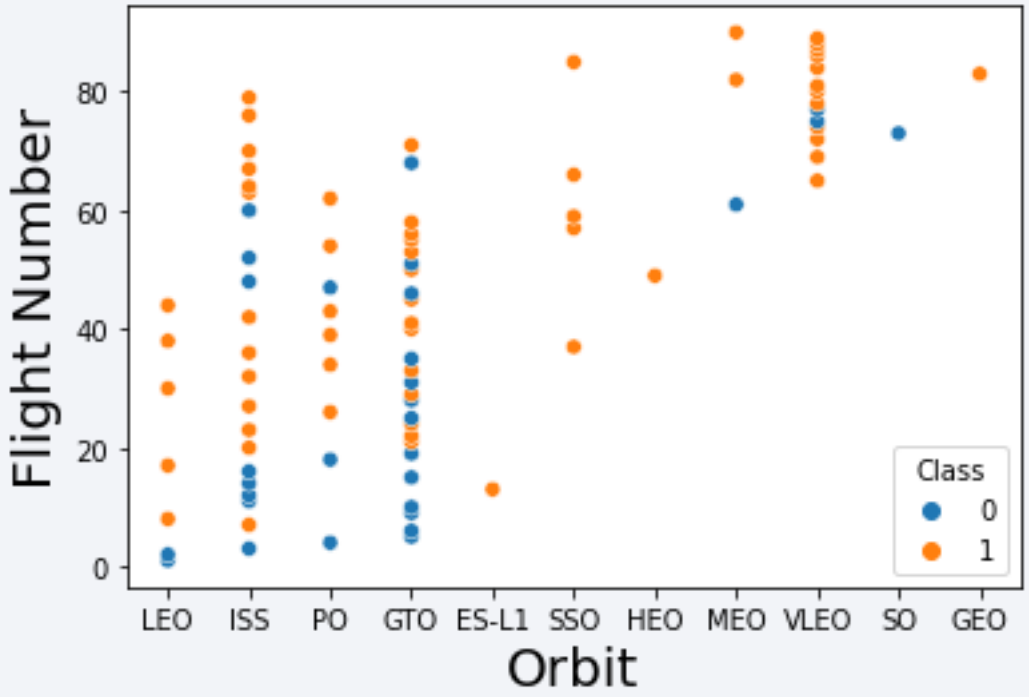
Success Rate vs. Orbit Type

- Orbits GEO, HEO, SSO, and ES-LI have the highest success rates.



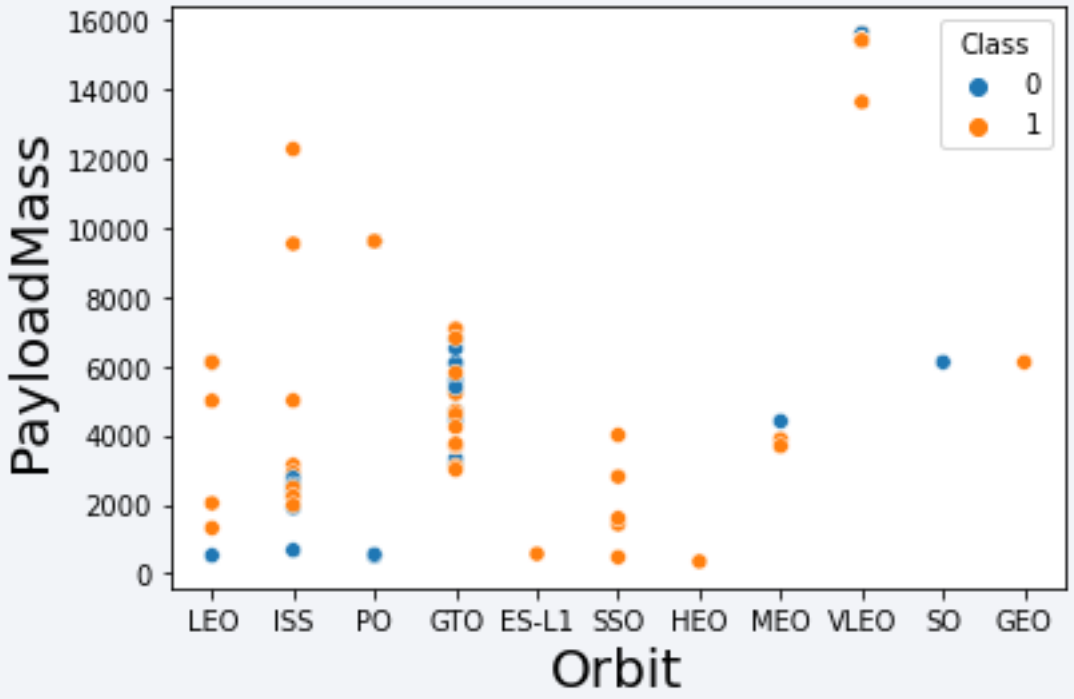
Flight Number vs. Orbit Type

- In the LEO Orbit, the higher flight numbers are successful.
- The GTO and ISS orbits have a lower success rate.
- ES-LI and HEO only had one flight each. Both were successful.
- The VLEO orbit had the highest success rate.



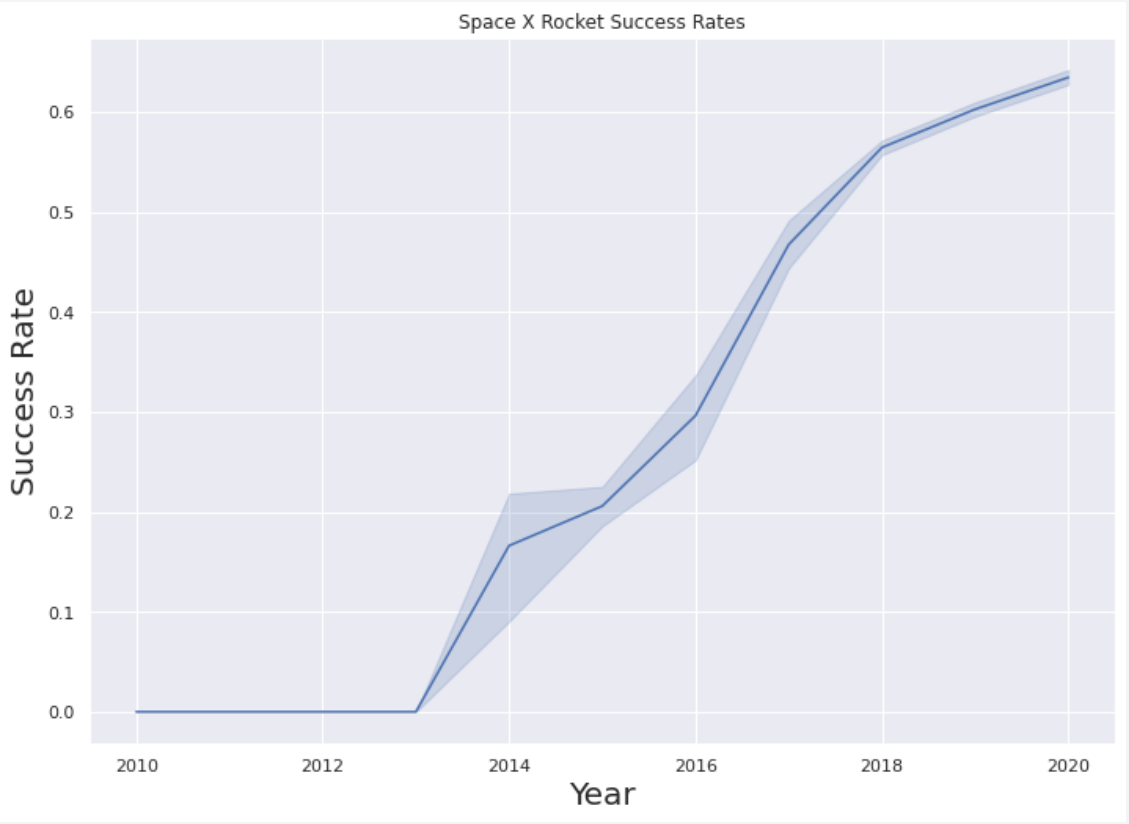
Payload vs. Orbit Type

- Heavy payloads are less common in each orbit.
- The GTO orbit maintains its payloads between 2000 and 8000.
- The ISS and PO and VLEO orbits all have seen payloads greater than 8000 and all have a high success rate.



Launch Success Yearly Trend

- The line graph shows an upward trend in the success rate starting from 2013.



All Launch Site Names

- SQL Query
  - Select DISTINCT Launch_Site from SPACEXTBL

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

- Explanation of Query
  - Using DISTINCT guarantees that only unique values in the launch_site column within SPACEXTBL are chosen.

Launch Site Names Begin with 'CCA'

- SQL Query
  - 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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Explanation of Query
  - Using WHERE will select the Launch_Site column and LIKE will select only names that begin with “CCA”. LIMIT 5 will select the top 5 results.

Total Payload Mass

- SQL Query
  - Select SUM(PAYLOAD_MASS__KG_) TotalPayloadMass from SPACEXTBL where customer = 'Nasa (CRS)'

totalpayloadmass
45596

- Explanation of Query
  - SUM adds all payload mass kg values from the SPACEXTBL table were the customer is 'NASA (CRS)'

Average Payload Mass by F9 v1.1

- SQL Query
  - Select AVG(PAYLOAD_MASS__KG_) AveragePayloadMass from SPACEXTBL where Booster_Version = 'F9 v1.1'

averagepayloadmass
2928

- Explanation of Query
  - AVG takes the average of the Payload mass kg column from the SPACEXTBL table where the booster version is 'F9 v1.1'

First Successful Ground Landing Date

- SQL Query
  - Select MIN(Date) SuccessfulLandingOutcome from SPACEXTBL where Landing__Outcome = ‘Success (drone ship)’

successfullandingoutcome
2016-05-27

- MIN selects the minimum date value from SPACEXTBL table where the Landing Outcome is a success for a drone ship.

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL Query
  - Select Booster_Version from SPACEXTBL where Landing__Outcome = 'Success (ground pad)' AND Payload_MASS__KG_ > 4000 AND Payload_MASS__KG_ < 6000

booster_version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

- Explanation of Query
  - The booster version column is viewed in the SPACEXTBL table where the Landing outcomes are success (ground pad). AND includes only payload mass kg that falls between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

- SQL Query
  - Select COUNT(Mission_Outcome) as Mission_Outcome from SPACEXTBL where Mission_Outcome Like 'Failure%' UNION select COUNT(Mission_Outcome) from SPACEXTBL where Mission_Outcome Like 'Success%'

mission_outcome
1
100

- Explanation of Query
  - COUNT counts the mission outcome from the SPACEXTBL table where there are both failures and success. UNION combines both queries. 1 failure and 100 successful missions.

Boosters Carried Maximum Payload

- SQL Query
  - Select DISTINCT Booster_Version, MAX(PAYLOAD_MASS__KG_) AS MaximumPayloadMass FROM SPACEXTBL GROUP BY Booster_Version ORDER BY MaximumPayloadMass DESC

booster_version	maximumpayloadmass
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
F9 B5 B1049.6	15440
F9 B5 B1059.3	15410
F9 B5 B1051.5	14932
F9 B5 B1049.3	13620
F9 B5B1058.1	12530
F9 B5B1061.1	12500
F9 B5B1051.1	12055
F9 B5 B1046.4	12050
F9 B4 B1041.2	9600
F9 B4 B1041.1	9600
F9 B5 B1049.2	9600
F9 B5B1048.1	9600
F9 FT B1036.2	9600
F9 FT B1029.1	9600
F9 FT B1036.1	9600
F9 B5B1047.1	7075
F9 B5B1049.1	7060
F9 B5 B1056.3	6956
F9 FT B1037	6761
F9 B5 B1047.3	6500
F9 B4 B1043.2	6460
F9 B4 B1044	6092
F9 FT B1034	6070
F9 B5 B1046.2	5800
F9 FT B1030	5600
F9 B5 B1058.2	5500
F9 B4 B1040.2	5384
F9 B5 B1047.2	5300
F9 FT B1021.2	5300
F9 FT B1032.1	5300
F9 FT B1020	5271
F9 FT B1031.2	5200
F9 B4 B1043.1	5000
F9 B4 B1040.1	4990
F9 B5 B1048.3	4850
F9 v1.1 B1016	4707
F9 FT B1022	4696
F9 FT B1026	4600
F9 v1.1	4535
F9 v1.1 B1011	4428
F9 B5B1054	4400
F9 B5B1060.1	4311
F9 B5B1062.1	4311
F9 FT B1032.2	4230
F9 B5 B1051.2	4200
F9 v1.1 B1014	4159
F9 B5 B1046.3	4000
F9 FT B1029.2	3669
F9 B5 B1046.1	3600
F9 FT B1024	3600
F9 B4 B1042.1	3500
F9 B4 B1039.1	3310
F9 FT B1021.1	3136
F9 B5 B1059.4	3130
F9 FT B1023.1	3100
F9 B5 B1048.2	3000
F9 B5 B1058.4	2972
F9 FT B1035.1	2708
F9 B4 B1045.2	2697
F9 B4 B1039.2	2647
F9 B5B1059.1	2617
F9 B5B1050	2500
F9 B5B1056.1	2495
F9 FT B1031.1	2490
F9 v1.1 B1012	2395
F9 B5 B1056.2	2268
F9 FT B1025.1	2257
F9 v1.1 B1010	2216
F9 FT B1035.2	2205
F9 FT B1038.2	2150
F9 FT B1019	2034
F9 B5 B1059.2	1977
F9 v1.1 B1018	1952
F9 v1.1 B1015	1898
F9 B5B1063.1	1192
F9 v1.0 B0007	677
F9 v1.1 B1013	570
F9 v1.1 B1017	553
F9 v1.0 B0005	525
F9 v1.0 B0006	500
F9 v1.1 B1003	500
F9 FT B1038.1	475
F9 B4 B1045.1	362
F9 v1.0 B0003	0
F9 v1.0 B0004	0

- Explanation of Query
  - DISTINCT selects only unique values. MAX selects the maximum payload mass from SPACEXTBL. GROUP BY groups the data by booster version and DESC orders the data in descending order.

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

- SQL Query
  - Select COUNT(Landing__Outcome) AS Successful_Landing_Outcomes_Between_Dates FROM SPACEXTBL WHERE (Landing__Outcome LIKE 'Success%') AND (Date>'2010-06-04') AND (Date<'2017-03-20')

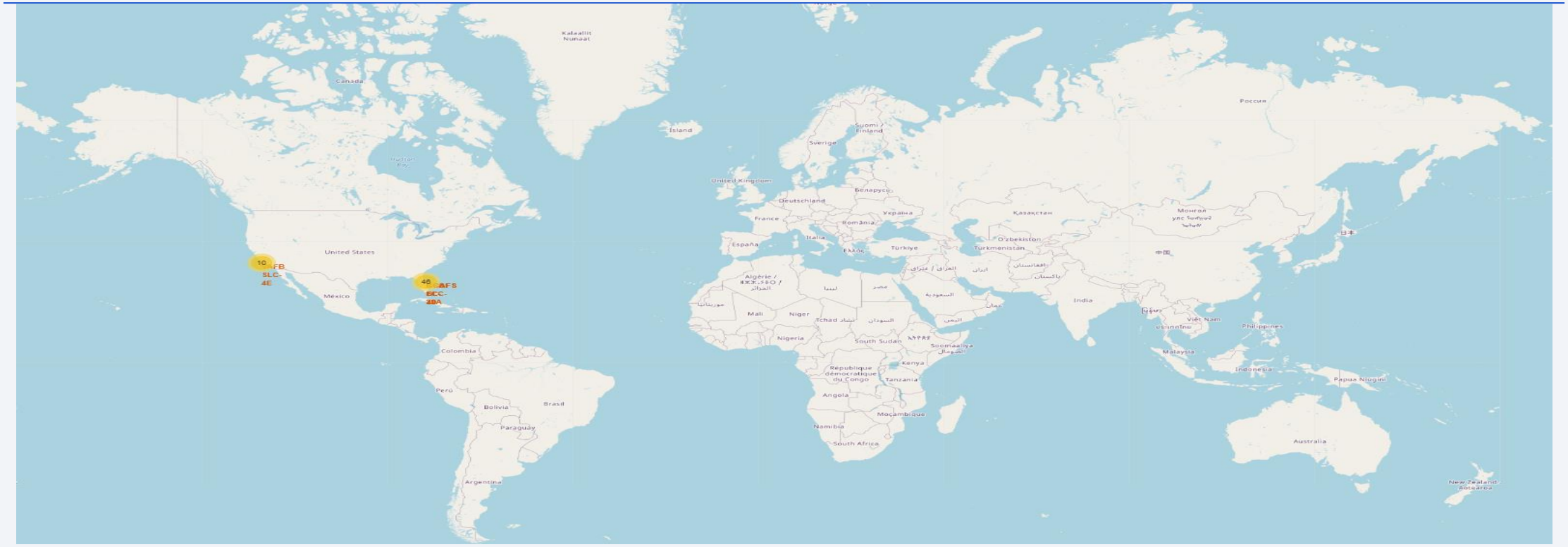
successful_landing_outcomes_between_dates
10

- Explanation of Query
  - COUNT counts the landing outcomes. WHERE further filters results to only include success outcomes. AND further filters the data to only include data between the dates '2010-06-04 and '2017-03-20'

Section 4

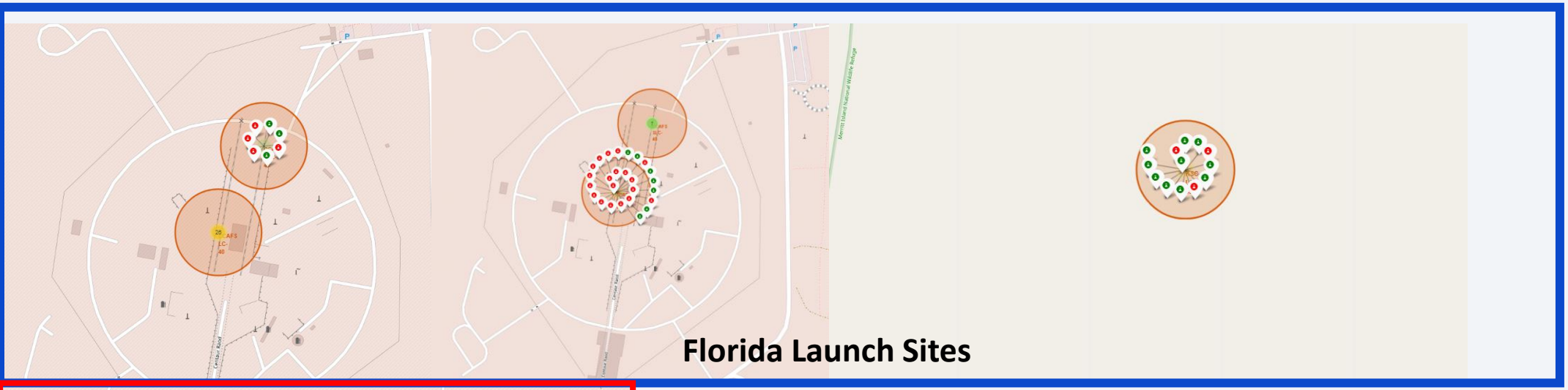
Launch Sites Proximities Analysis

All SpaceX Launch Sites Global Map



- All SpaceX Launch sites are in the U.S.A

Color Labeled Launch Sites

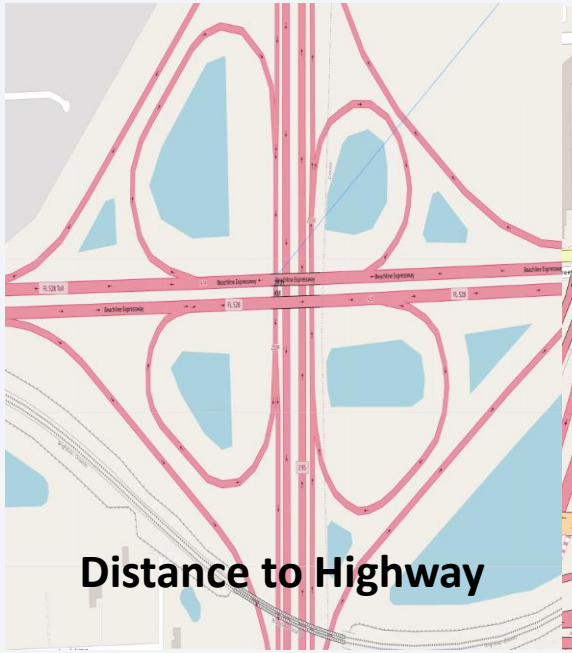


Florida Launch Sites

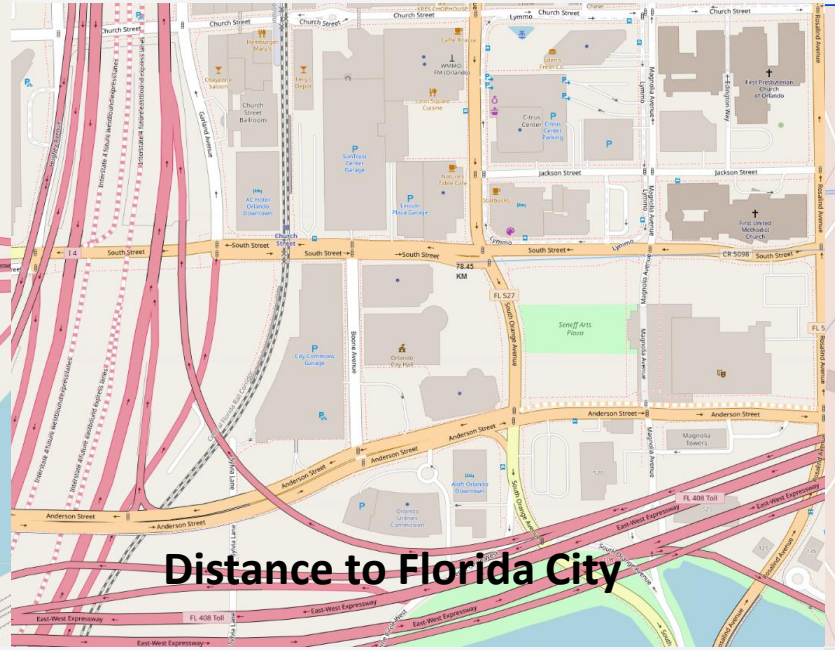
California Launch Site

- The green markers show successful launches.
- The red markers show unsuccessful launches.
- There are 3 launch sites in Florida and 1 in California.

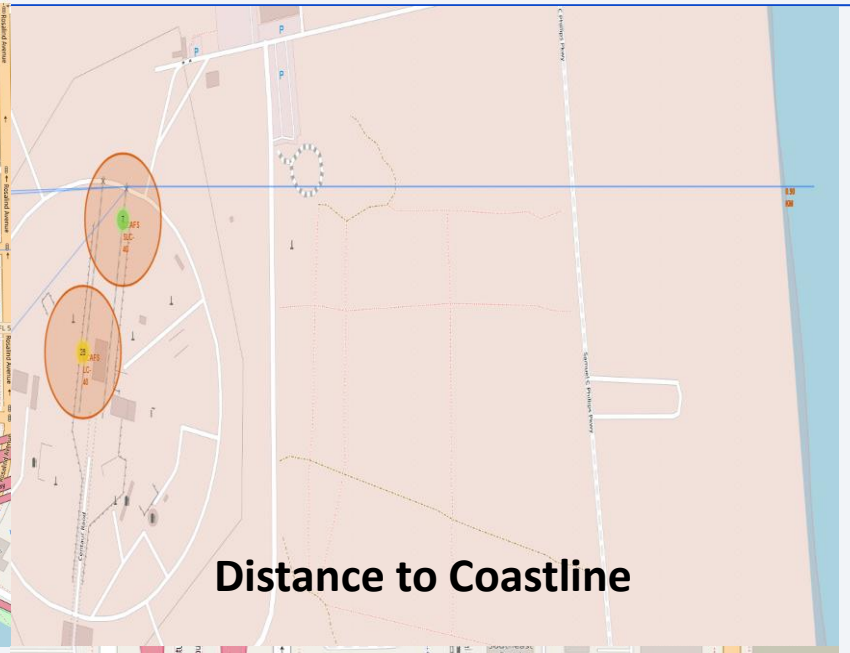
CCAFS SLC-40 Launch Site Distance from Landmarks



Distance to Highway

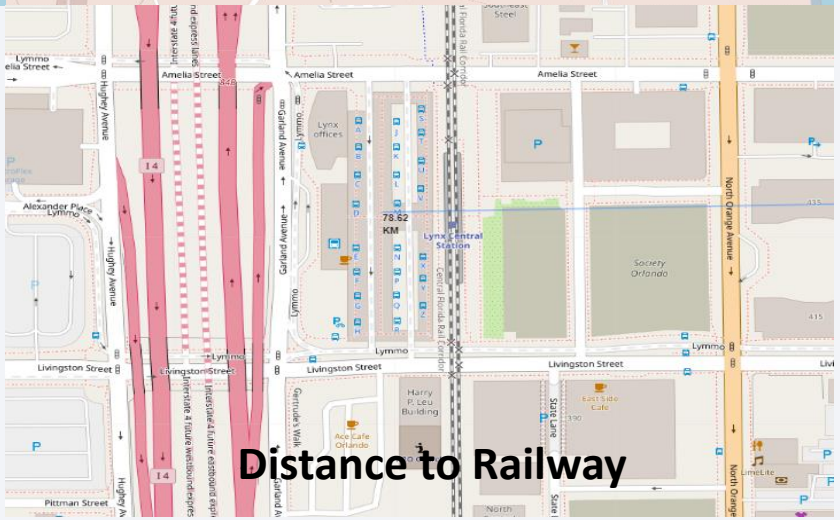


Distance to Florida City



Distance to Coastline

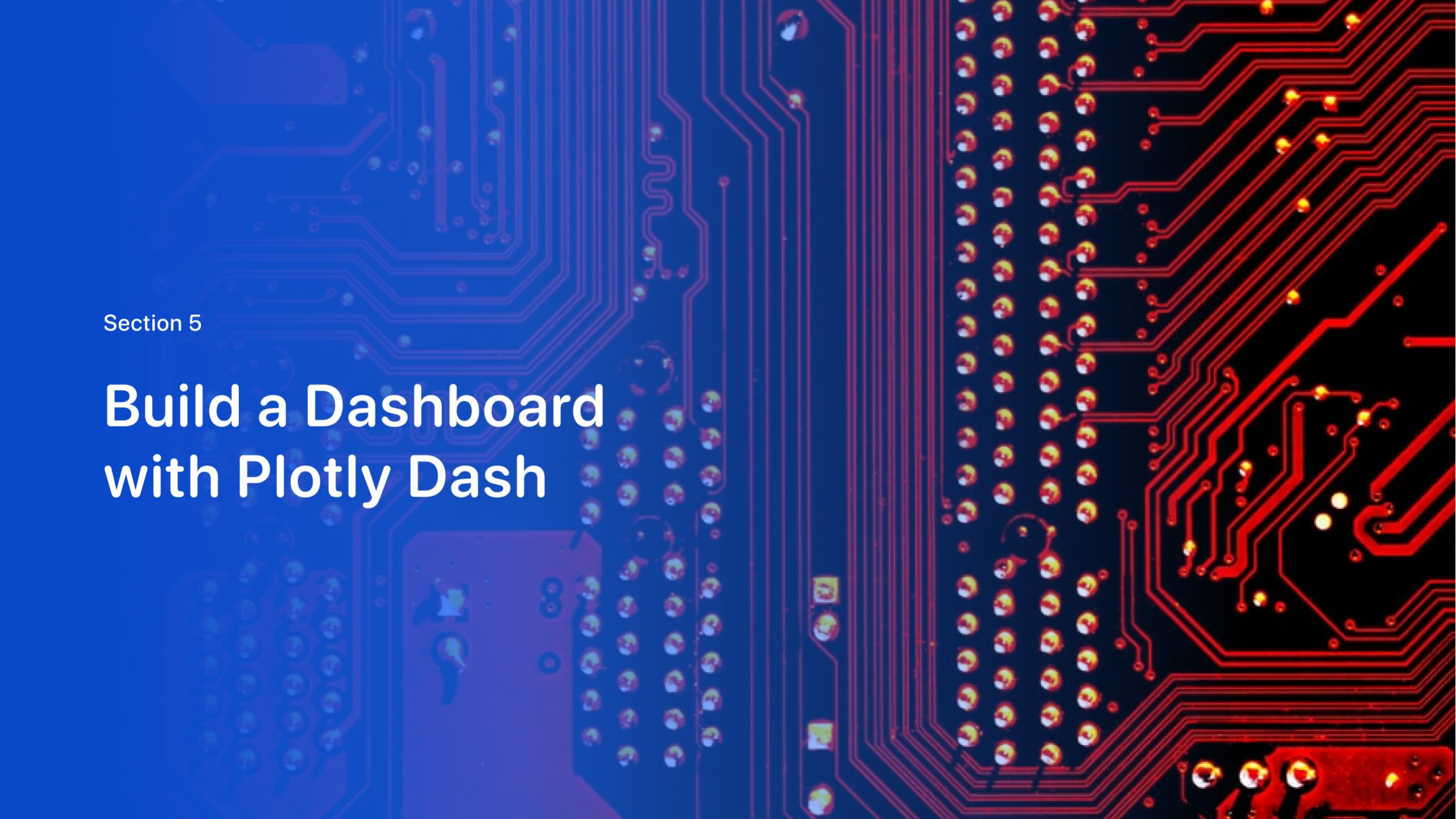
- The launch site is not at a close distance to highways, cities, or railways but is close to the coastline.



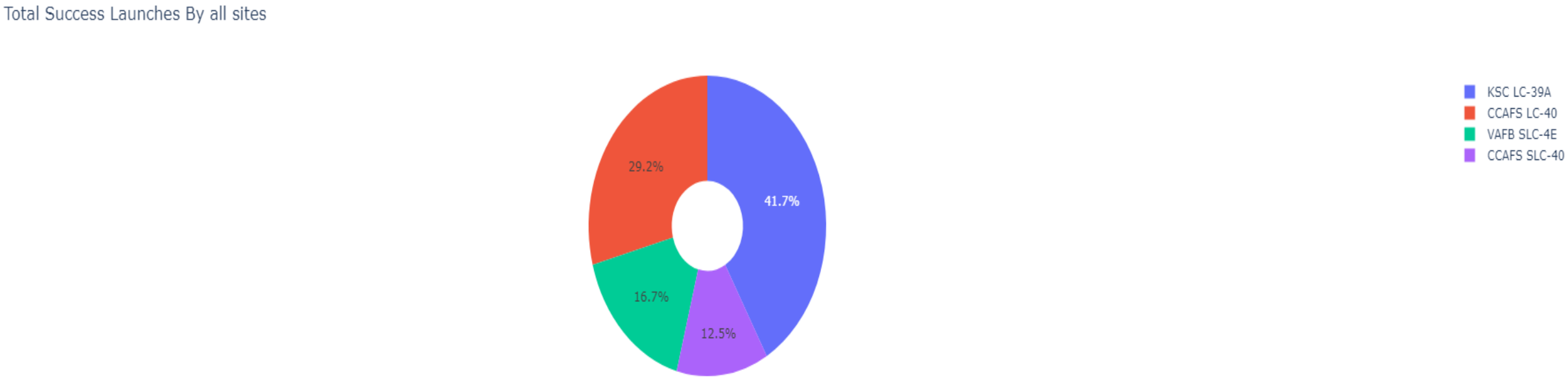
Distance to Railway

Section 5

Build a Dashboard with Plotly Dash

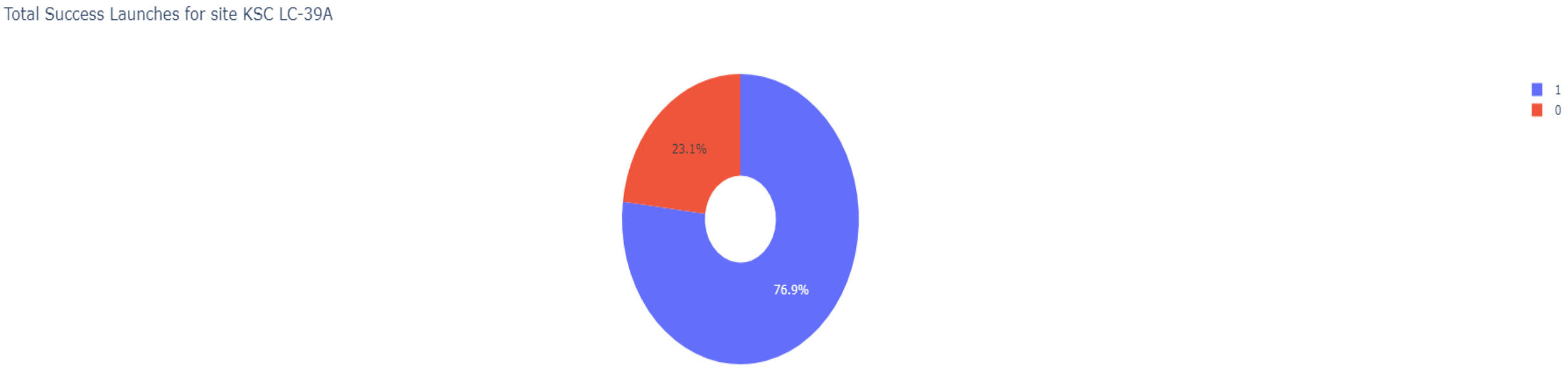


SpaceX Success Count for All Sites



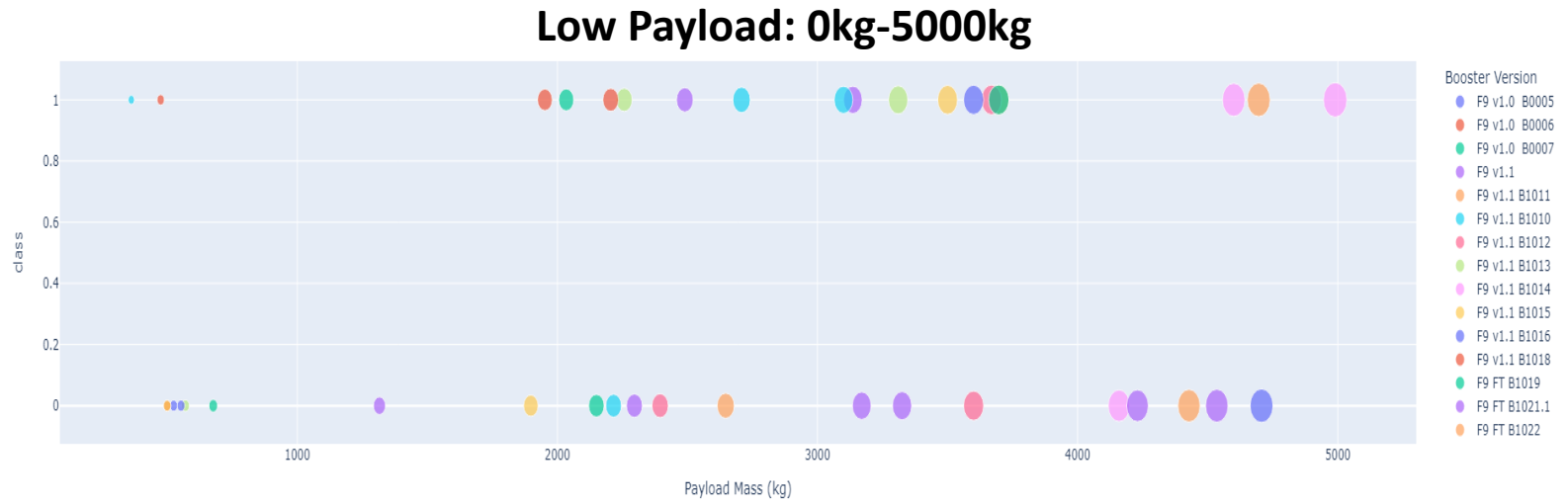
- The highest success rate belongs to launch site KSC LC-39A

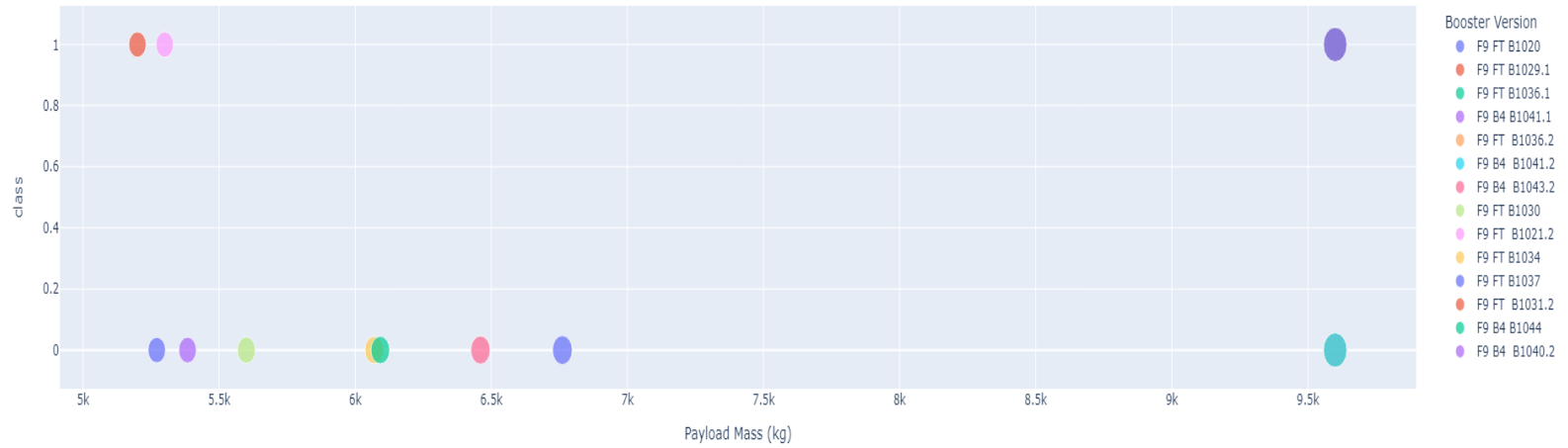
KSC LC-39 Launch Site



- Site KSC LC-39A has the highest success ratio with only a 23.1% failure rate.

Scatter Plot Payload vs. Launch Outcome for all sites





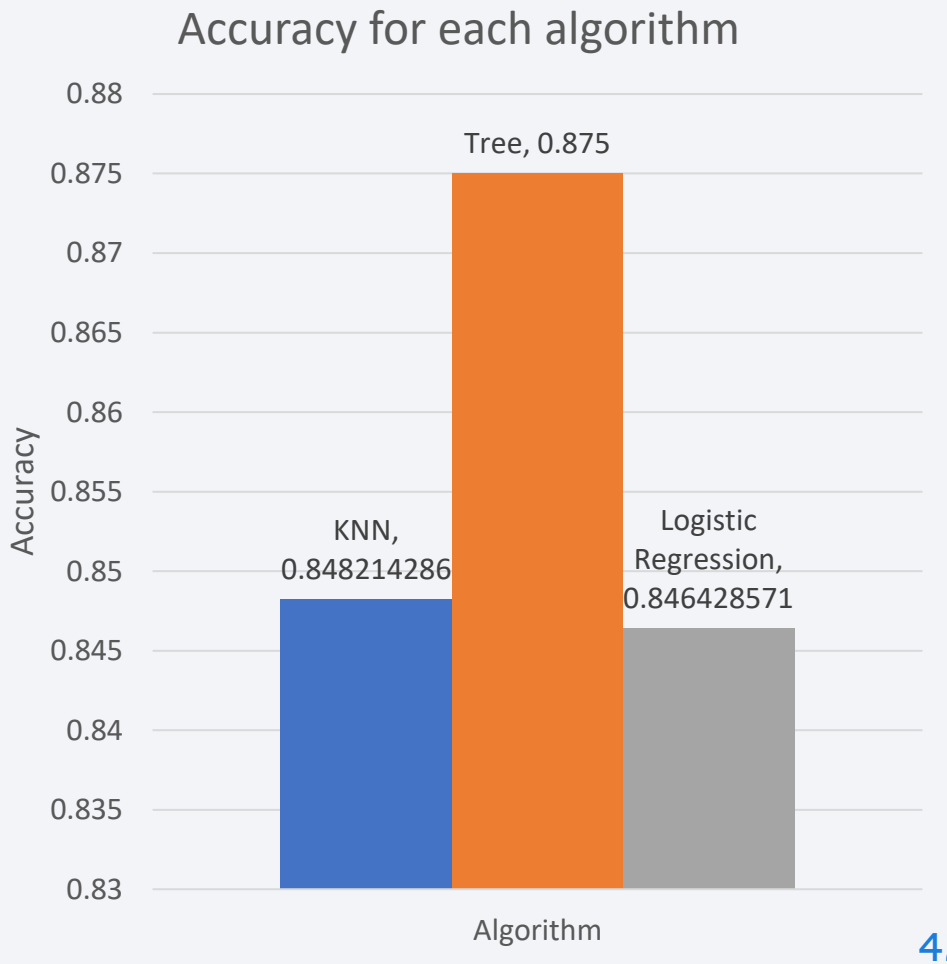
- The amount of launches in the low payload range are higher than the high payload range.
- Higher payload has lower successful launches
- F9 FT B1019 and F9 FT B1021.1 are the only booster versions that go above a payload of 8000kg

Section 6

Predictive Analysis (Classification)

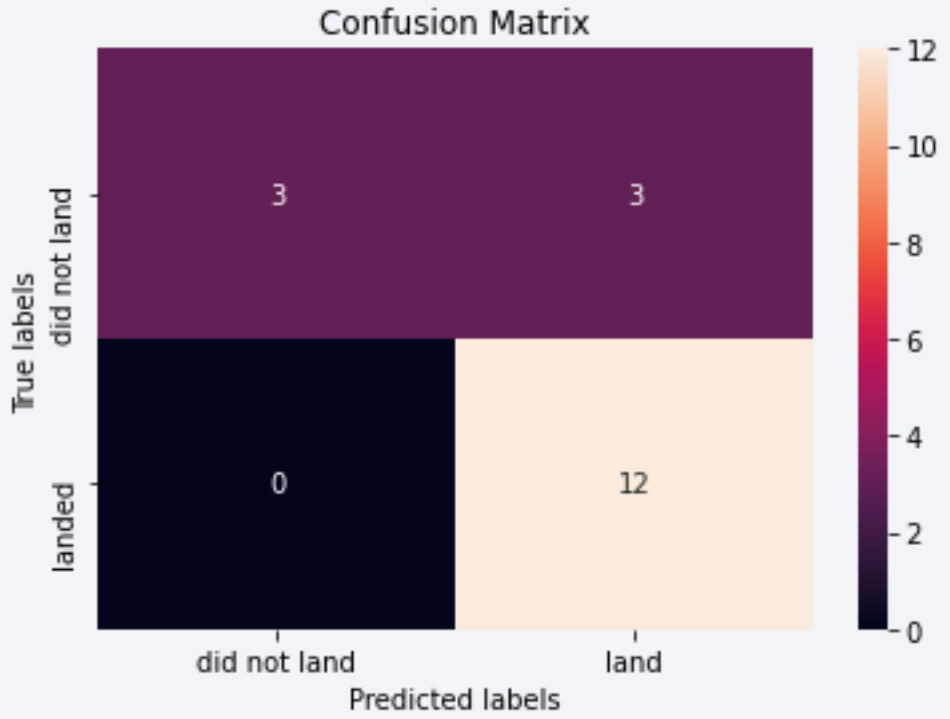
Classification Accuracy

- Both the KNN and Logistic Regression accuracies are close. Within 0.0014 of each other.
- The Tree algorithm has the best accuracy.



Confusion Matrix for the Tree Algorithm

- The confusion matrix shows 12 true positives, 3 false negatives and 3 true negatives.



Conclusions

- The Tree Algorithm performed best for this dataset for machine learning.
- Low weighted payloads are less common but have a higher success rate.
- The success rates for SpaceX launches increase throughout the years.
- The GEO,HEO,SSO, and ES-L1 have the highest success rates.
- KSC LC-39A is the site with the most successful launches.

Thank you!