
# Winning Space Race with Data Science

Jiachen Zhuo

March 24, 2022

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Interactive Visual Analytics with Folium
  - Build a Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- 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, a 62% reduction as compared to other providers which cost upward of 165 million dollars each. Much of the savings is because Space X can reuse the first stage during launch.
- Problems we want to find answers
  - What factors will affect the successful landing of the rocket's first stage?
  - Can we build a machine learning model to predict the successful landing of the Falcon 9 rocket?

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - SpaceX API and Web scraping
- Perform data wrangling
  - One Hot Encoding for categorical features, replace of null values and grouping of launching results to binary outcome
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models compared based on prediction accuracy: Logistic Regression, Support Vector Machine, Decision Tree and K Nearest Neighbor

# Data Collection

- Data were collected from two sources:
  - SpaceX REST API from: https://api.spacexdata.com/v4/launches/past
    - From the main json file, more detailed information were obtained based on IDs provided:
      - rocket ID: -> BoosterVersion
      - payload ID: -> Launch Sites and corresponding Longitude and Latitude
      - launchpad ID: -> PayloadMass, Orbit
      - cores ID: -> Flights, GridFins, Reused, Legs, LandingPad, Launching Outcome
  - Web scraping from the SpaceX Wiki page: (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

# Data Collection – SpaceX API

- We used the get request from the SpaceX REST API.
- Requested other key information from specific ID-specific addresses.
- We then filtered data to get only ‘Falcon 9’ launches and filled in missing values.
- GitHub URL:

https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.1_Data_Collection_API.ipynb

**1. Request and parse the SpaceX launch data using the GET request**

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

```
response = requests.get(spacex_url)
```

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

**2. Request and fill in other key information**

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

**3. Filter data to get only ‘Falcon 9’ Launches**

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

**4. Fill in missing values in PayloadMass with mean value**

```
# Calculate the mean value of PayloadMass column
plm_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, plm_mean, inplace=True)
```

# Data Collection - Scraping

- We used Beautiful Soup to request the Falcon9 launch data from the Wiki page.
- After locating all the relevant tables, we extracted column information, and filled in the data frame with the rest of the HTML tables
- GitHub URL: [https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.1.2_SpaceX_web_scraping.ipynb](https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.1.2_SpaceX_web_scraping.ipynb)

**1. Use Beautiful Soup to request the Falcon9 Launch Wiki page**

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from
soup = BeautifulSoup(data, 'html.parser')
```

**2. Locate all the relevant tables and extract column information**

```
# 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')
len(html_tables)
```

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
```

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

**3. Create a data frame by parsing the launch HTML tables**

```
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']=[]
```

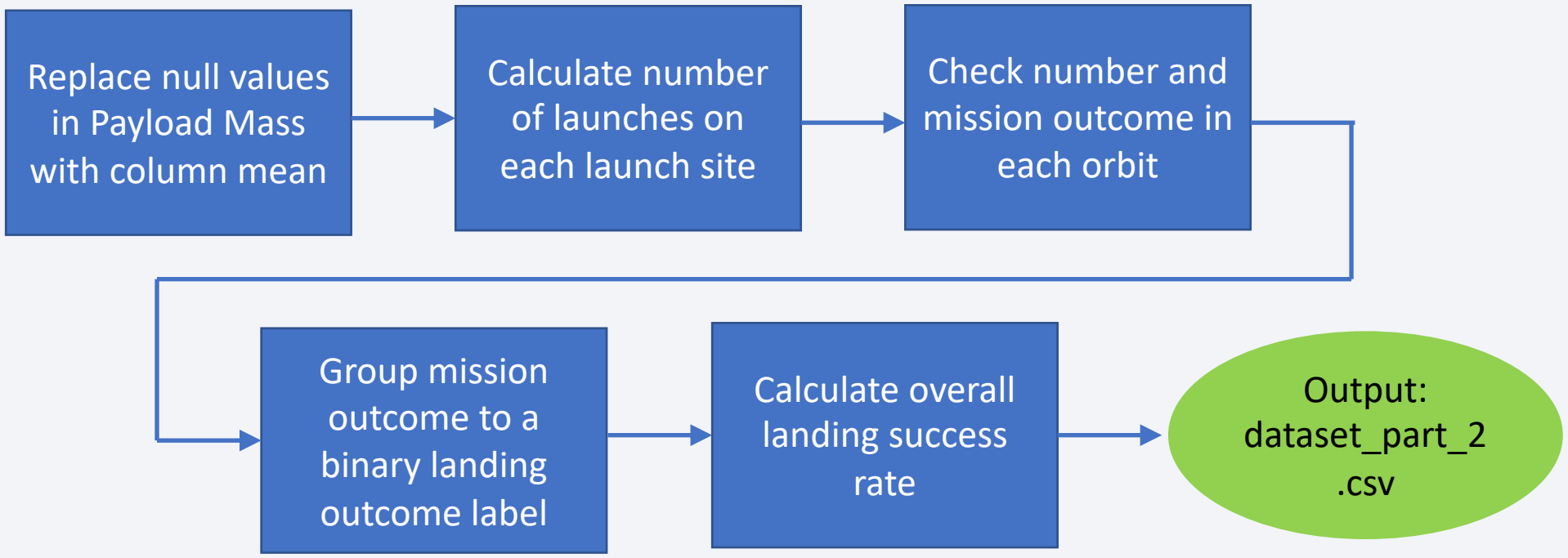
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)

            # Time value
```

# Data Wrangling

- Data Processing Steps:




- GitHub URL: https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.1.3_Data_Wrangling.ipynb

# EDA with Data Visualization

- We used scatter plots, bar graphs and line plots to explore relationships between the launching outcome with flight number, launch site, payload mass, orbit type, as well as the trend of launch success rate over the years.
- GitHub URL: https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.2.1_jupyter-labs-eda-dataviz.ipynb

# EDA with SQL

- SQL queries performed:
  - Display the names of unique launch sites in the space mission
  - Display 5 records where launch sites begin with the 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 when the first successful landing outcome in ground pad was achieved
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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
- GitHub URL: https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.2.2_Exploratory_Data_Analysis_with_SQL.ipynb

# Build an Interactive Map with Folium

- Map objects created and added to a folium map:

| Map Object | Display | Purpose |
|---|---|---|
| Marker & Circle | Launch Sites name | To show where the launch sites are |
| MarkerCluster | Individual launch outcome | To show individual launch outcome for each launch site |
| MousePosition | Coordinates of mouse position | To explore proximities of launch sites |
| Line & Marker | Line between two coordinates with distance | To draw a line between a launch site to closest coastline and display the distance |

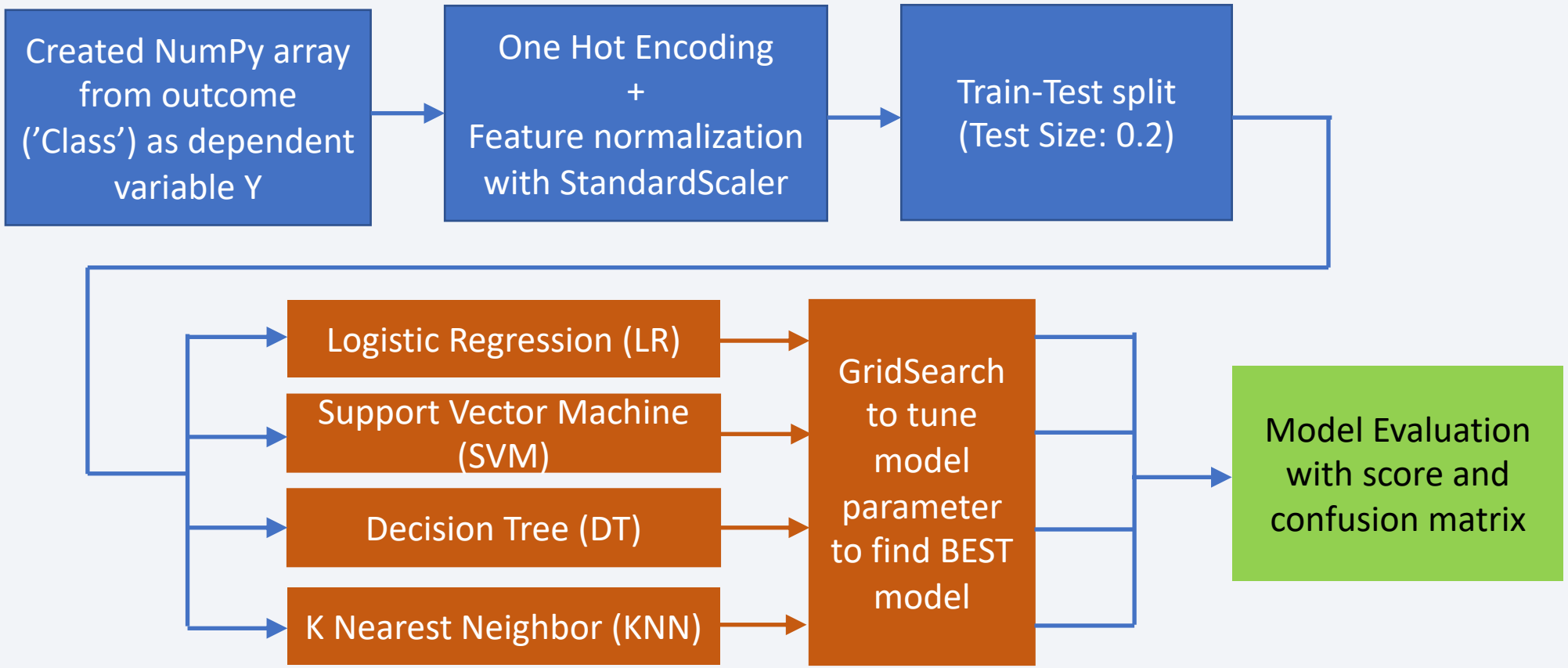
- GitHub URL: https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.3.1_lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

- Created pie chart to explore total success launches by site or for all sites with dropdown manual control.
- Created scatter plot to show correlation between Payload and Success for corresponding sites, where the payload range is controlled by a slider.
- GitHub URL: https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/spacex_launch_dash.py

# Predictive Analysis (Classification)

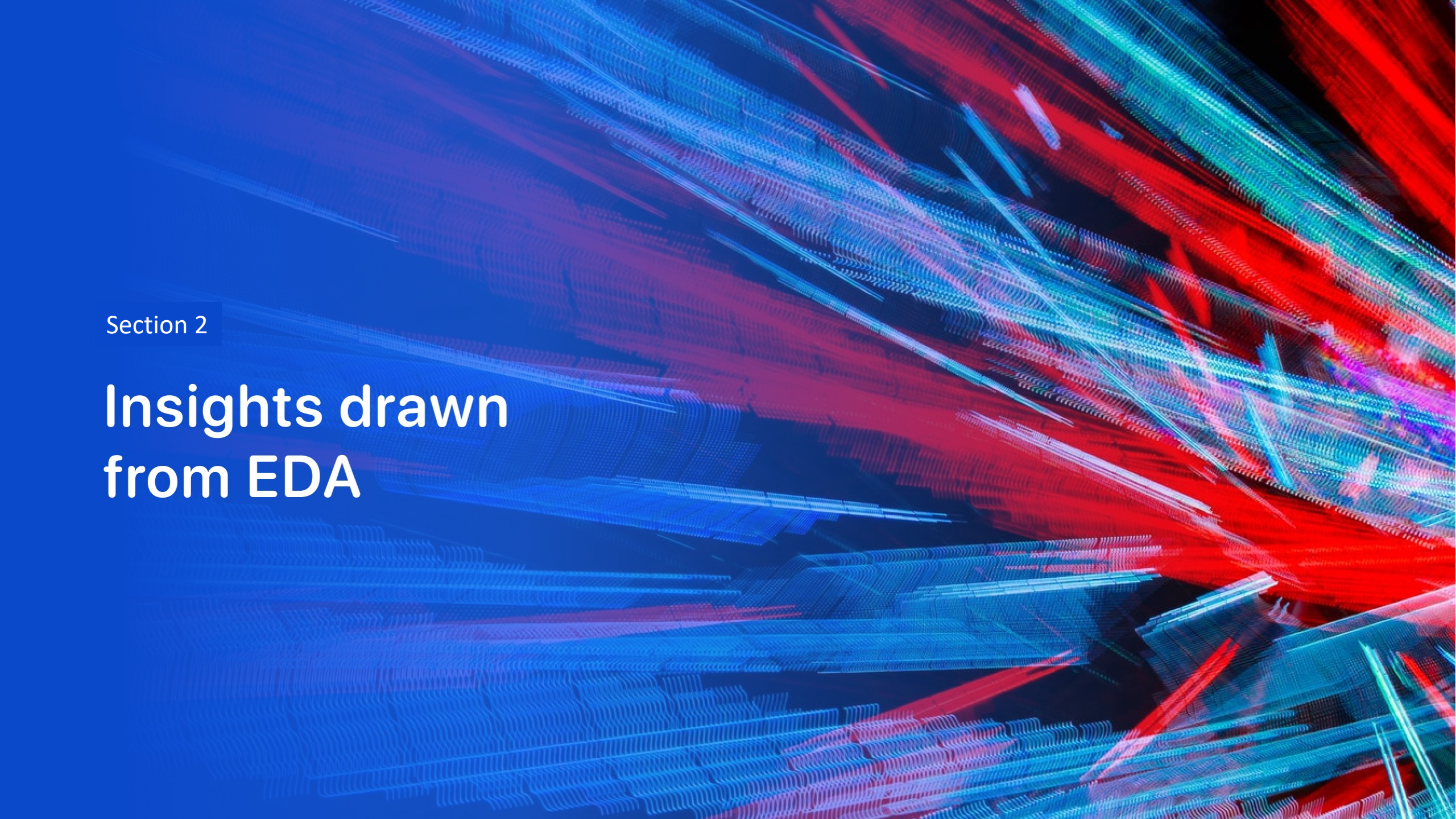
- Procedure:




- GitHub URL: https://github.com/jZhuoMRI/IBM_Data_Science_Capstone/blob/main/10.4.1_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

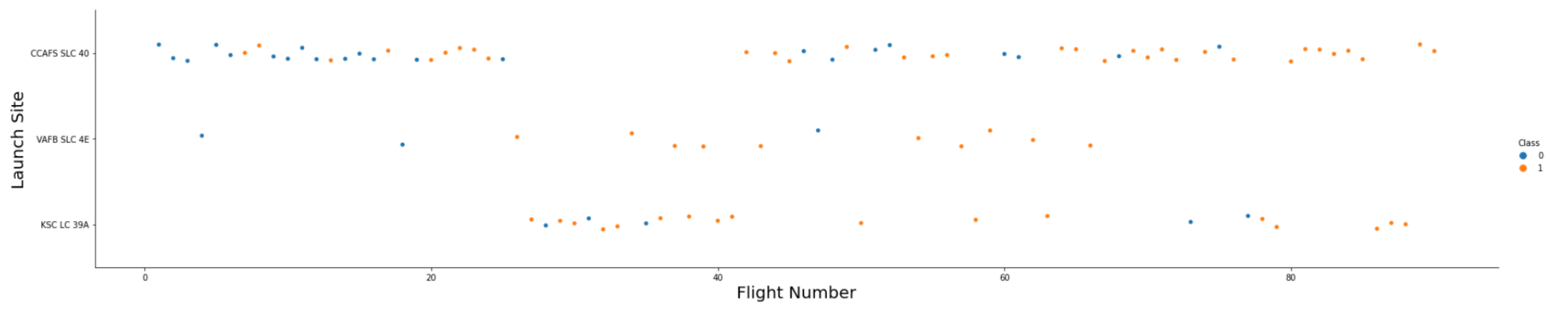
# Results

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


Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site

Scatter plot of Flight Number vs. Launch Site (Class: 0 – Failure, 1 – Success)

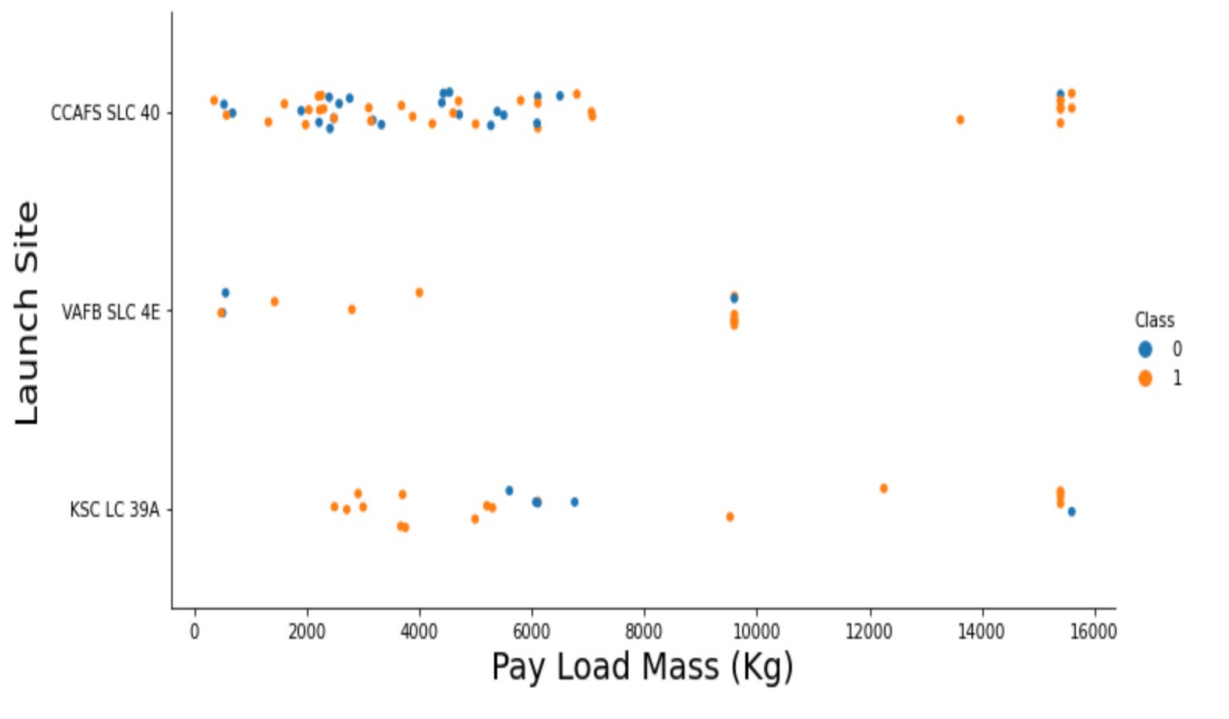



- Early launches are mostly at CAFS SLC40 with low success
- Later launches are mostly at CAFS SLC40 and KSC LC-39A
- Success rate increases as flight number increases at all launching sites

# Payload vs. Launch Site

Scatter plot of Payload vs. Launch Site

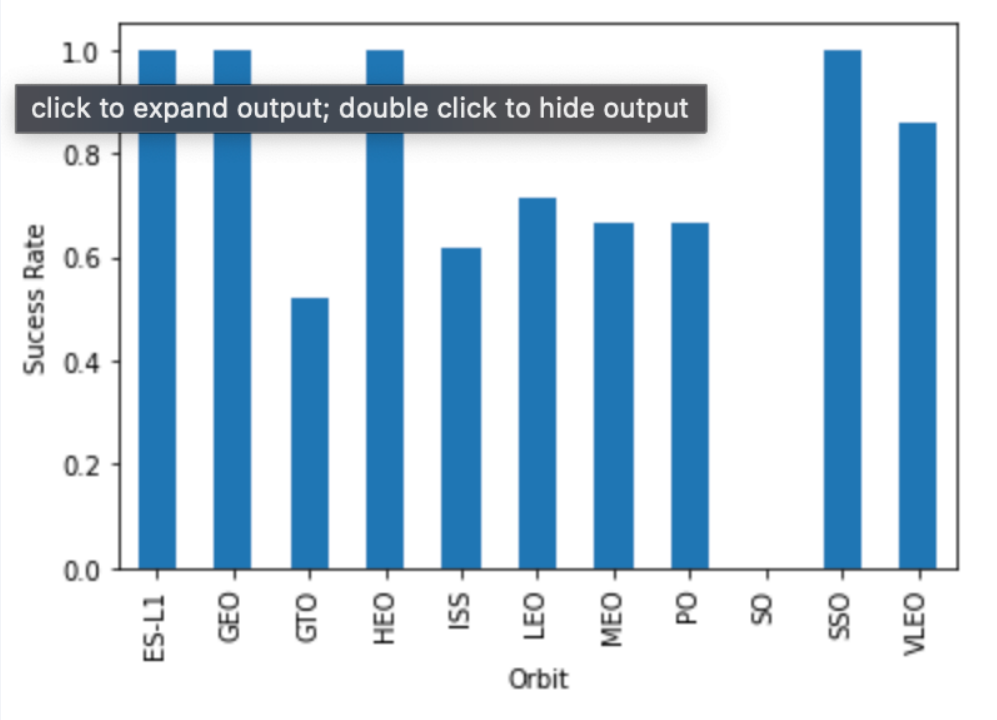
(Class: 0 – Failure, 1 – Success)

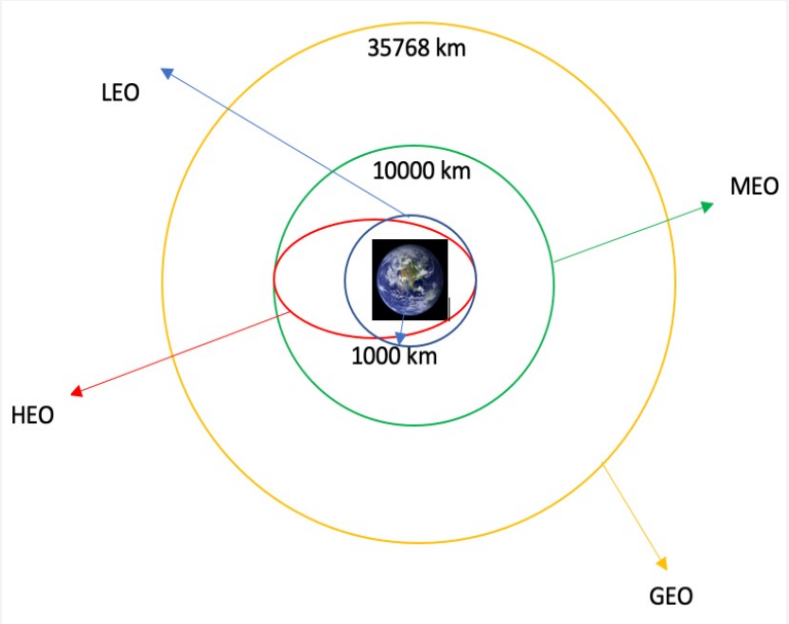



- WAFB SLC 4E does not do any high Payload launches
- KSC LC 39A and VAFB SLC 4E have higher success rates in low Payload launches than CCAFS SLC 40

# Success Rate vs. Orbit Type

Bar chart for the success rate of each orbit type

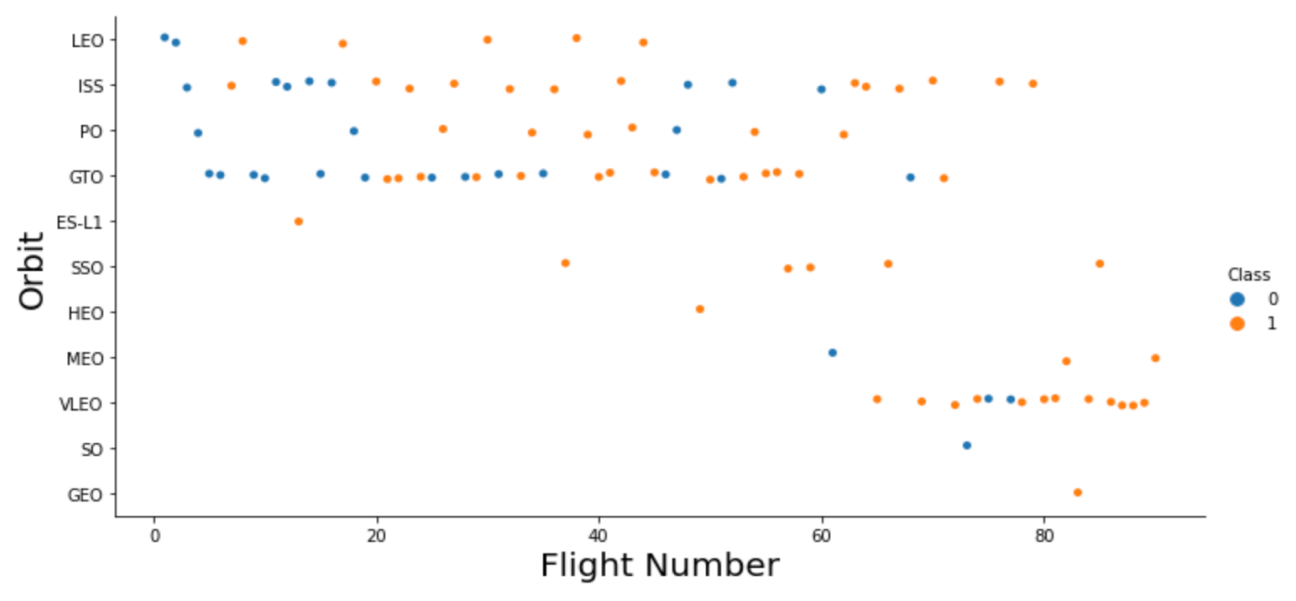



- ES-L1, GEO, HEO and SSO has 100% success rate
- GTO has the lowest success rate

# Flight Number vs. Orbit Type

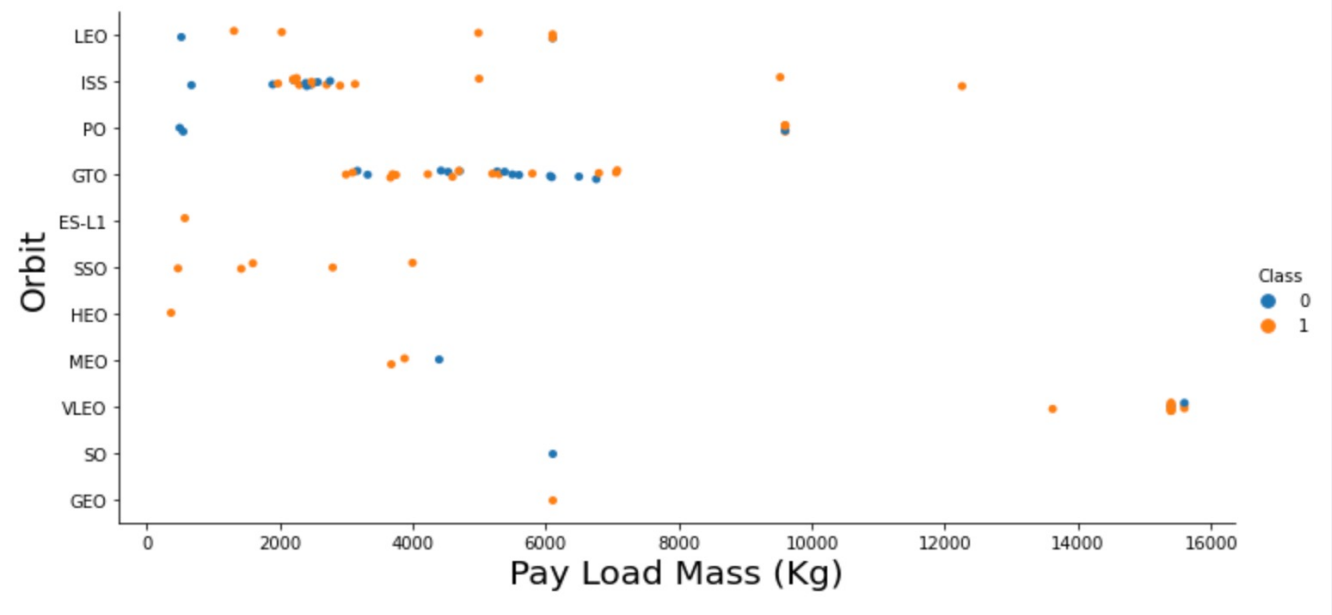
Scatter plot of Flight number vs. Orbit type




- The initial launches were mostly focused on LEO, ISS, PO and GTO. Later other orbit types were being used.
- SSO had 5 launches and were all successful

# Payload vs. Orbit Type

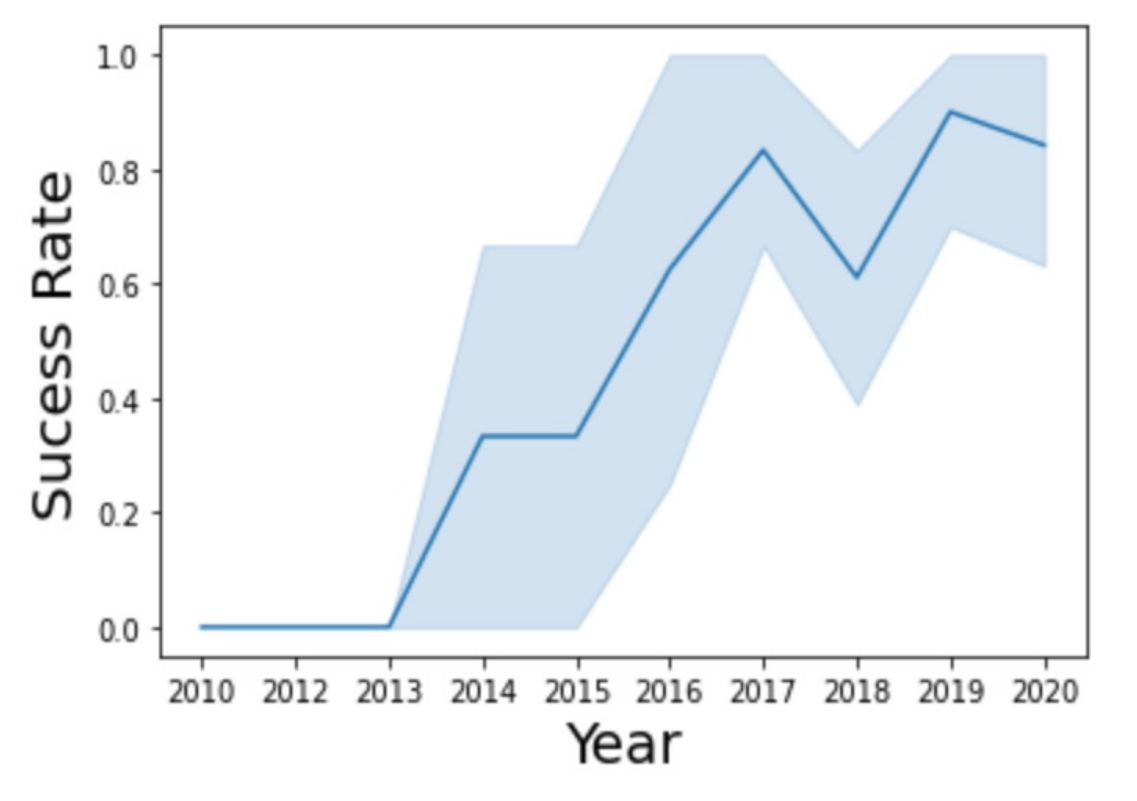
Scatter plot of payload vs. orbit type




- Payload are distributed differently for differ orbit types.
- Some orbits are only for low payload launches (e.g. SSO, LEO, GTO)
- VLEO is strictly used for high payload launches

# Launch Success Yearly Trend

Line chart of yearly average success rate




- Average success rate has been increasing over the years
- Success rate of 2019 and 2020 has averaged above 80%

# All Launch Site Names

- Find the names of the unique launch sites

```
%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'

- Find 5 records where launch sites begin with `CCA`

```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5;
```

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 2010-12-08 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (parachute) |
| 2012-05-22 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(payload_mass__kg_) from SPACEXTBL where customer = 'NASA (CRS)';
```

| 1 |
| --- |
| 45596 |

- The total payload carried by boosters from NASA is 45596 Kg.

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
%sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version like 'F9 v1.1%';
```

| 1 |
| --- |
| 2534 |

- Average payload mass by booster F9 v1.1 is 2534 Kg.

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%sql select min(DATE) from SPACEXTBL where landing__outcome = 'Success (ground pad)'
```

| 1 |
| --- |
| 2015-12-22 |

- The first successful landing outcome on ground pad is Dec 22, 2015.

# Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%%sql
select booster_version, landing__outcome, payload_mass__kg_ from SPACEXTBL
    where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000
```

| booster_version | landing__outcome | payload_mass__kg_ |
|---|---|---|
| F9 FT B1022 | Success (drone ship) | 4696 |
| F9 FT B1026 | Success (drone ship) | 4600 |
| F9 FT B1021.2 | Success (drone ship) | 5300 |
| F9 FT B1031.2 | Success (drone ship) | 5200 |

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%sql select landing__outcome, count(*) from SPACEXTBL group by landing__outcome
```

| landing__outcome | 2 |
|---|---|
| Controlled (ocean) | 5 |
| Failure | 3 |
| Failure (drone ship) | 5 |
| Failure (parachute) | 2 |
| No attempt | 22 |
| Precluded (drone ship) | 1 |
| Success | 38 |
| Success (drone ship) | 14 |
| Success (ground pad) | 9 |
| Uncontrolled (ocean) | 2 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
%%sql
select booster_version,payload_mass__kg_ from SPACEXTBL
    where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL);
```

| booster_version | payload_mass__kg_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 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 |

- There are many booster versions that has a maximum payload mass of 15600 Kg.

# 2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select landing__outcome, booster_version, launch_site, DATE from SPACEXTBL
    where landing__outcome = 'Failure (drone ship)' and DATE like '2015%';
```

| landing__outcome | booster_version | launch_site | DATE |
|---|---|---|---|
| Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 | 2015-01-10 |
| Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 | 2015-04-14 |

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

- 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

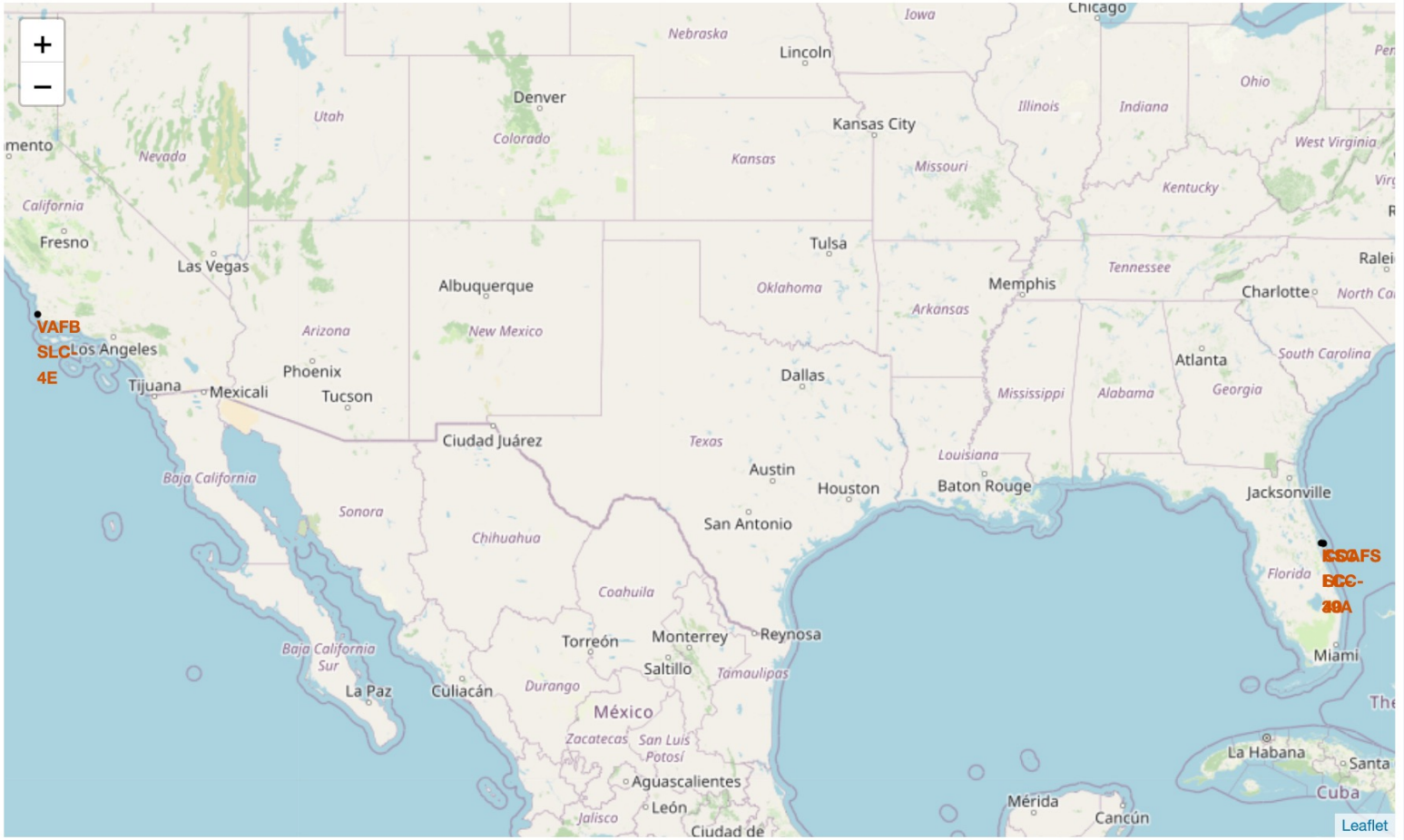
```
%%sql
select landing__outcome, count(*) from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20'
    group by landing__outcome order by count(*) desc;
```

| landing__outcome | 2 |
|---|---|
| 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

# Rocket Launching Sites

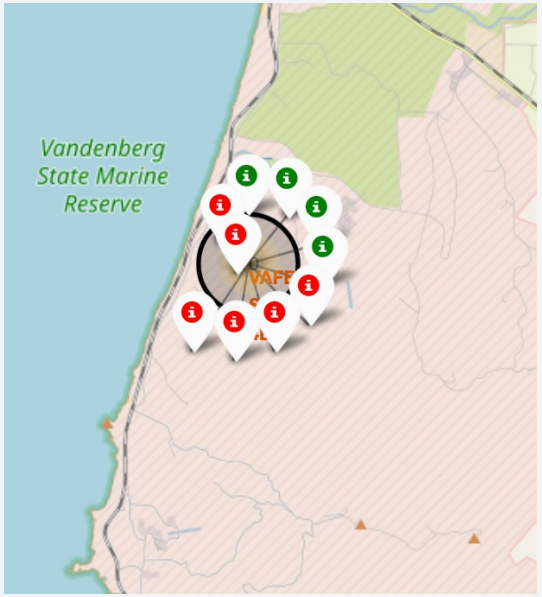
| | Launch Site | Lat | Long |
|---|---|---|---|
| 0 | CCAFS LC-40 | 28.562302 | -80.577356 |
| 1 | CCAFS SLC-40 | 28.563197 | -80.576820 |
| 2 | KSC LC-39A | 28.573255 | -80.646895 |
| 3 | VAFB SLC-4E | 34.632834 | -120.610746 |



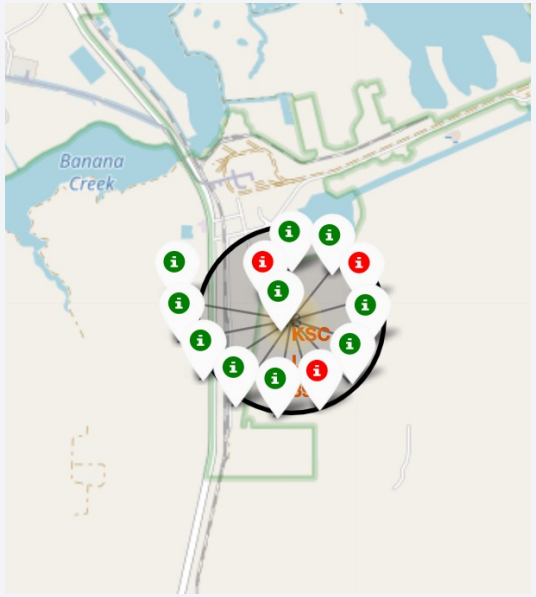

- Three of the launching sites are in Florida, one in California.

# Success/failed launches for each site

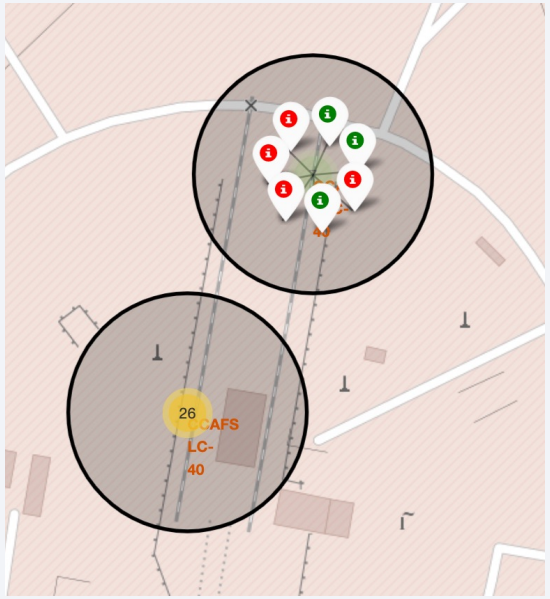
**VAFB SLC-4E**



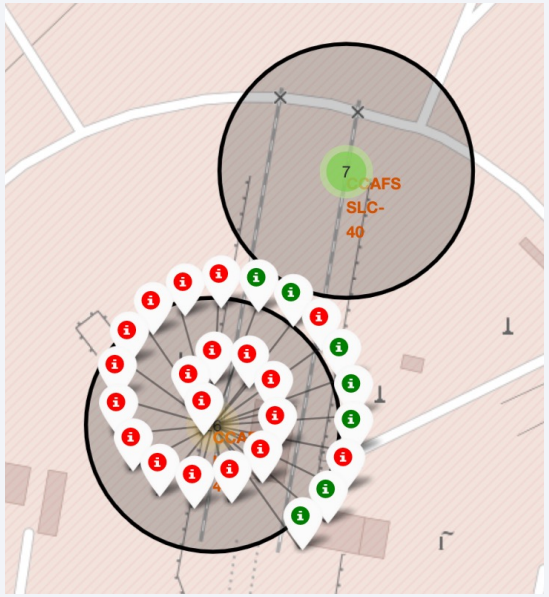

**KSC LC-39A**




**CCAFS LC-40**



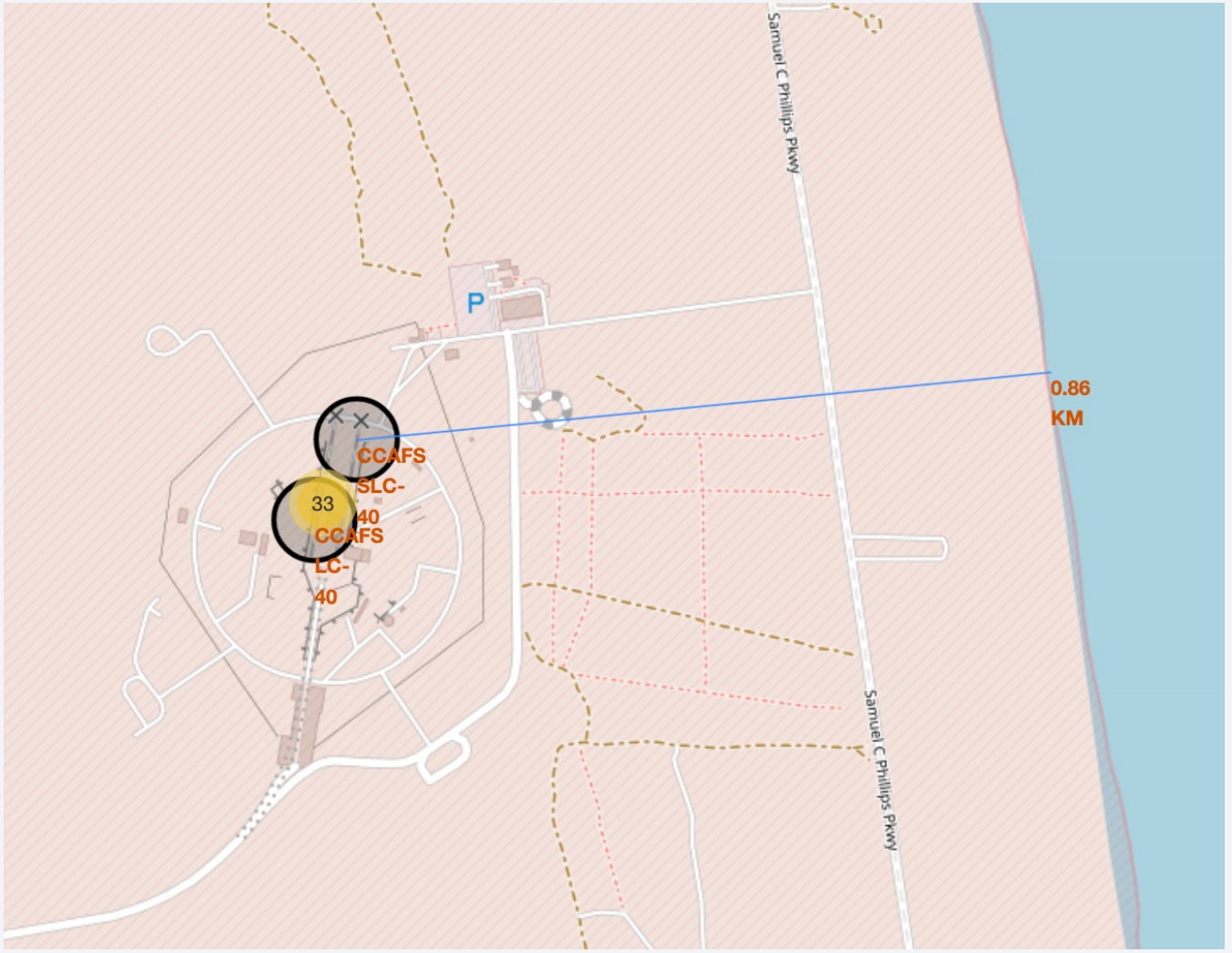

**CCAFS SLC-40**

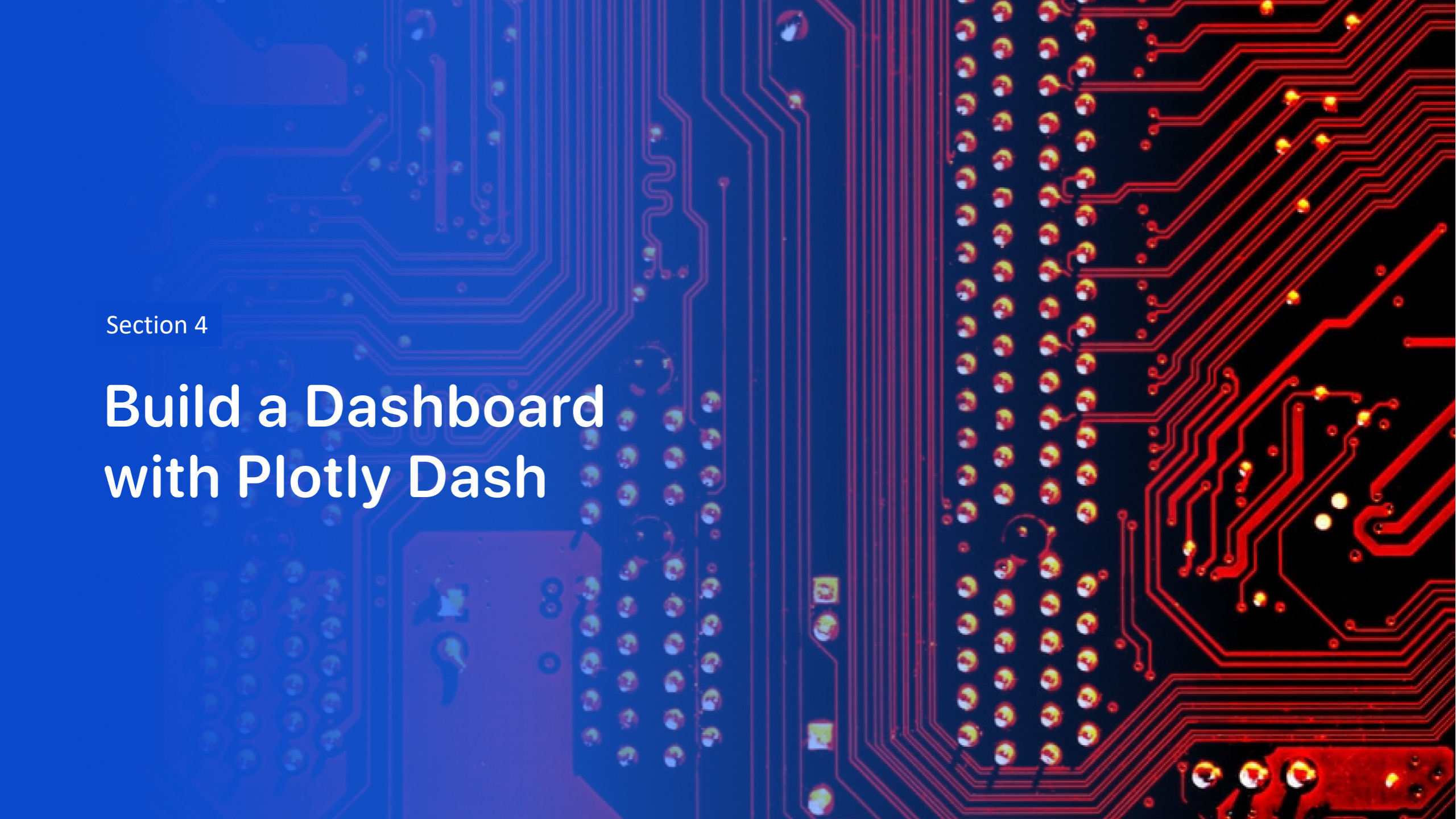



- CCASF SLC-40 had the highest number of launches. KSC LC-39A has the highest success rate.

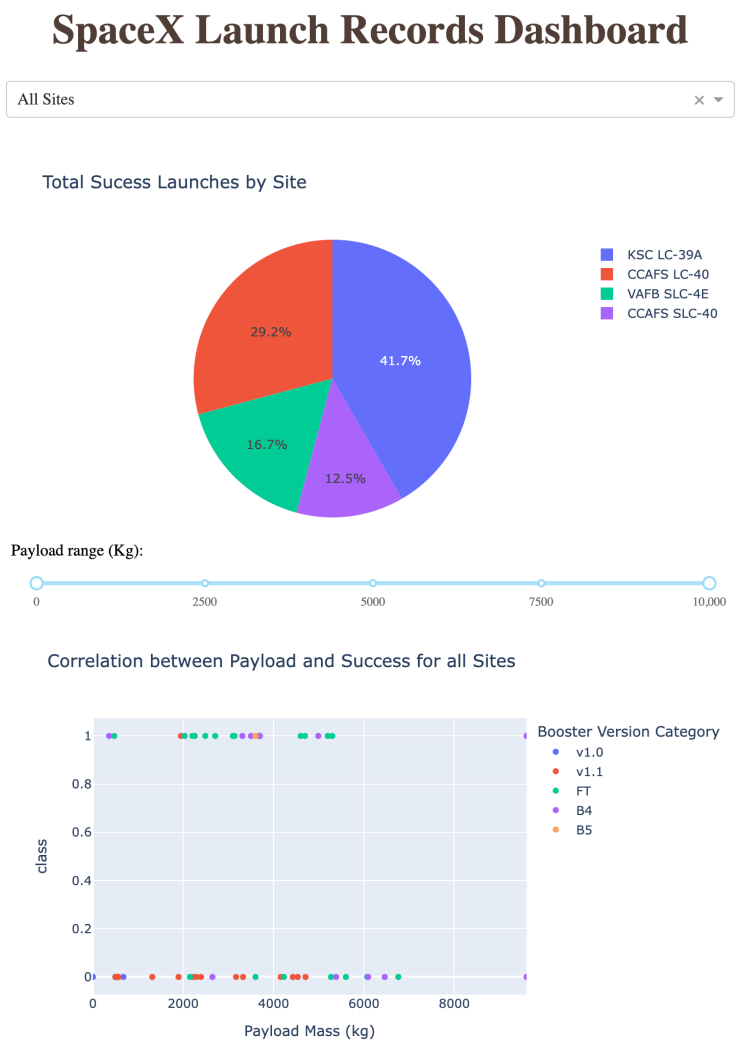
# Distance from CCAFS SLC-40 to the closest coastline

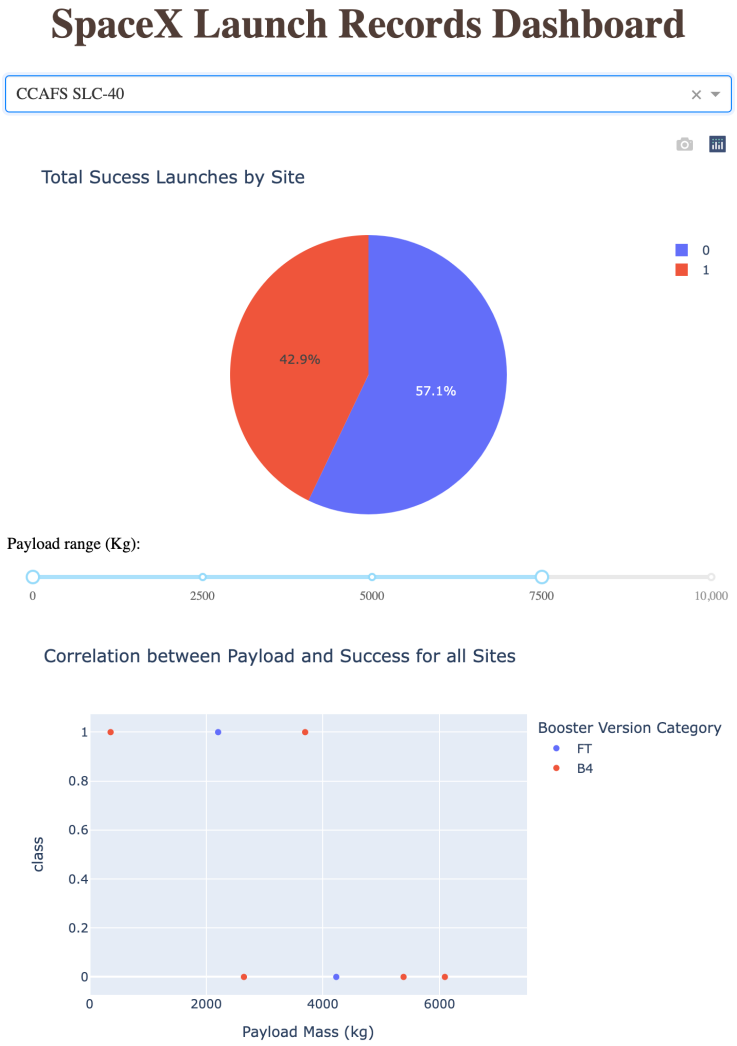
- The distance from CCAFS SLC-40 to the closest coastline is 0.86 KM.




Section 4
Build a Dashboard with Plotly Dash

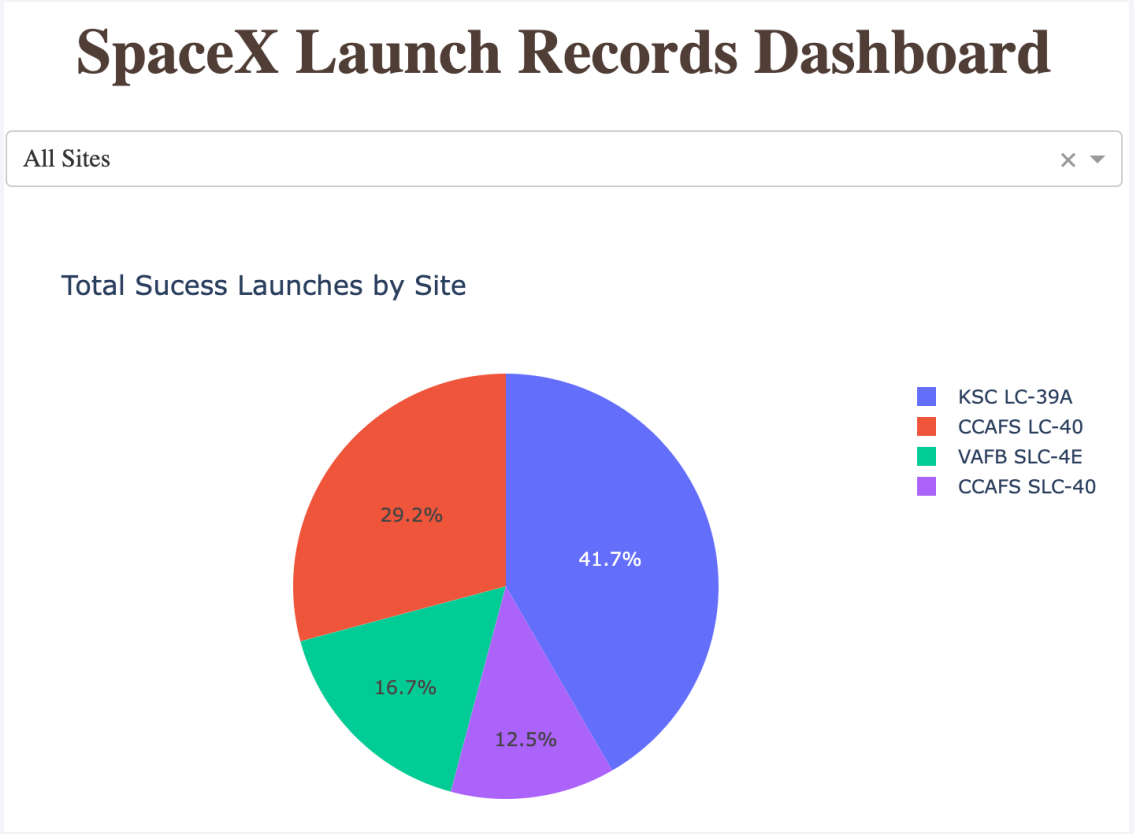
# Build a Dashboard with Plotly Dash



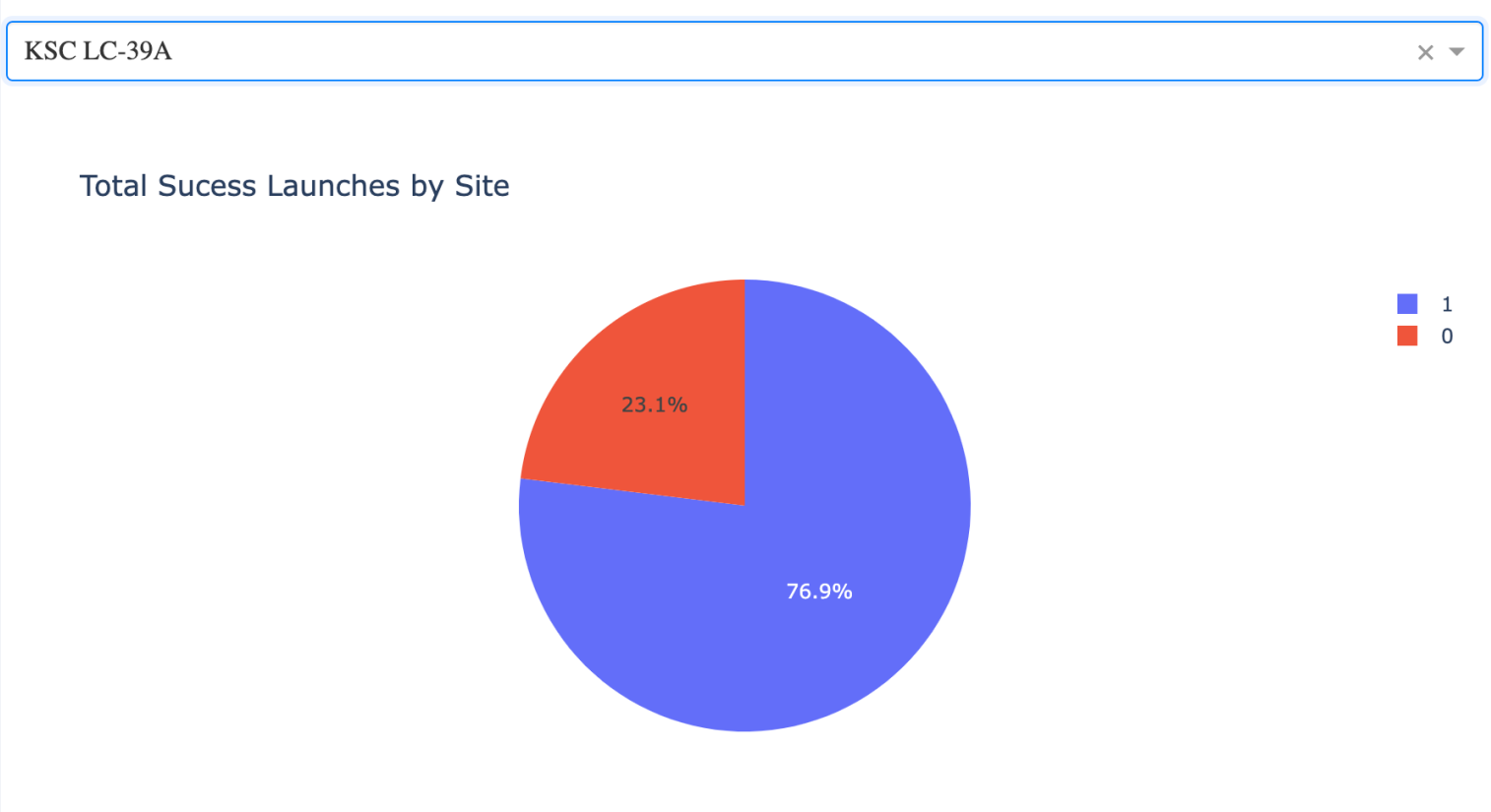

- Created pie chart to explore total success launches by site or for all sites with dropdown manual control.
- Created scatter plot to show correlation between Payload and Success for corresponding sites, where the payload range is controlled by a slider.
- GitHub URL:

# Total Success Launches by Sites



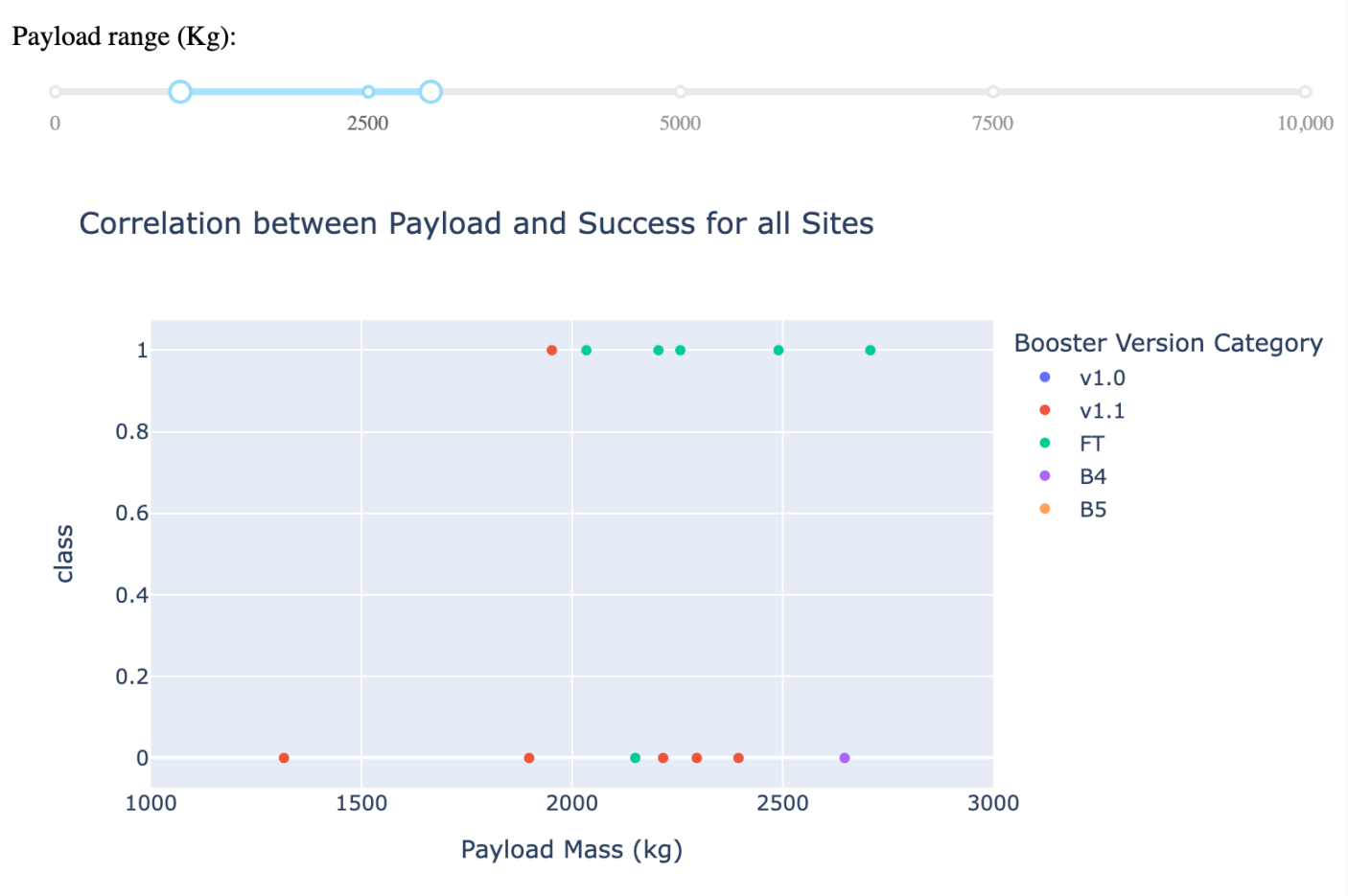

- KSC LC-39A has the highest percentage of success launches

# Success Launches at Site: KSC LC-39A




- KSC LC-39A HAS A 76.9% launch success rate.

# Correlation between Payload and Success for all Sites



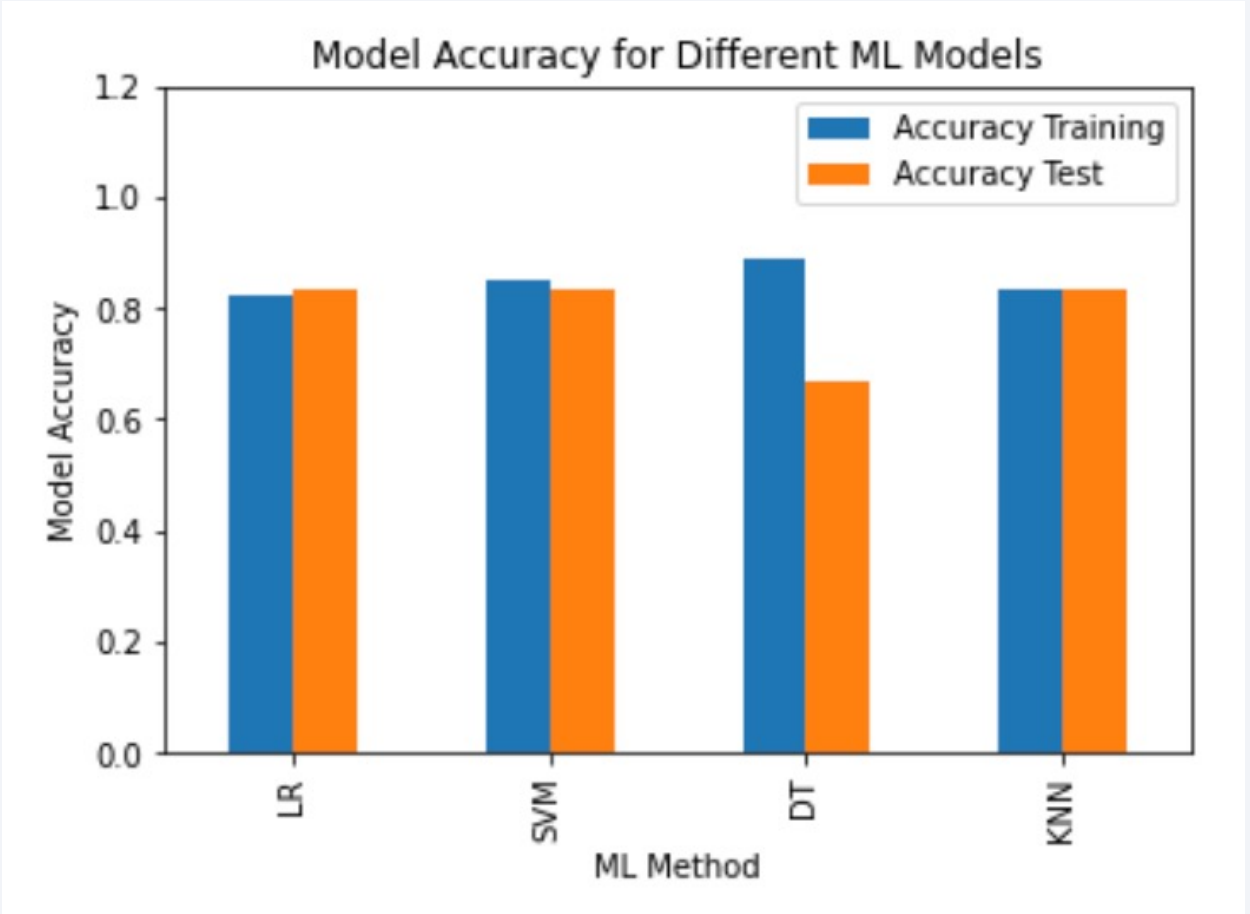

- Within a Payload range of 1000 to 3000 kg, Booster Version FT has the largest success rate.

Section 5

# Predictive Analysis (Classification)

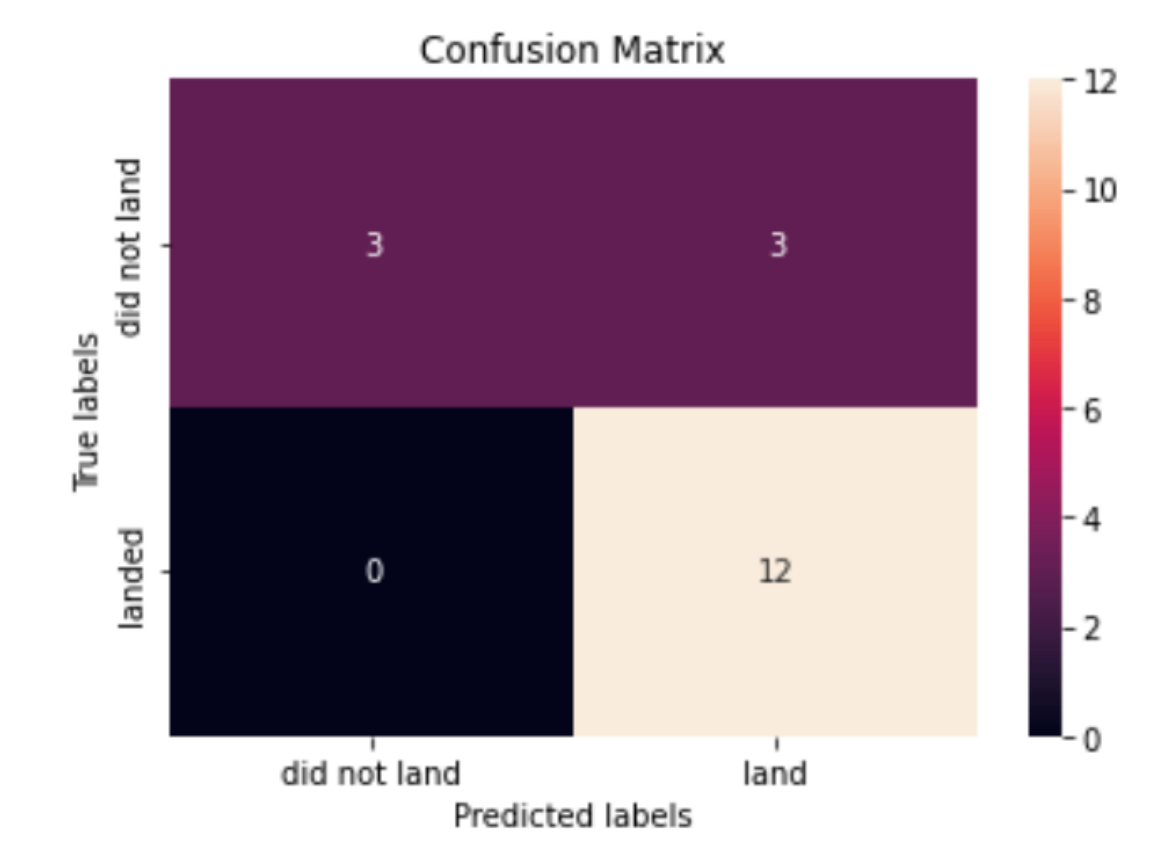
# Classification Accuracy

- While DT model had a high training accuracy, its performance in the test data was the worst (67%).
- The rest of three models (LR, SVM and KNN) all had the same model accuracy (83%).



# Confusion Matrix

**LR, SVM, KNN**




- Except for DT, all other models performed the same on the test data
- All models predicted successful landing well
- Error in classification is mostly false positives.

# Conclusions

- The launching success rate increase over time as more launching are being conducted. This is also reflected by higher success rate with larger flight number.
- Orbits ES-L1, GEO, HEO, SSO and VLEO had the highest success rate
- KSC LC-39A is the launching site with the highest success rate
- Machine learning models can predict successful launch reasonably well (83% accuracy). The model tends to have high false positive rates though.
- The Decision Tree model had the highest model accuracy with training data but performed poorly in the test data. It is hard to judge which model performed the best due to the limited sample size. More future data will be helpful to validate our model.

# Thank you!