IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thasindu Amarasinghe
11/20/2025

Outline

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

Executive Summary

- In this capstone project, we will predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms.
- The main steps in this project include:
  - ❑ Data collection, wrangling, and formatting
  - ❑ Exploratory data analysis
  - ❑ Interactive data visualization
  - ❑ Machine learning prediction
- Our graphs show that some features of the rocket launches have a correlation with the outcome of the launches, i.e., success or failure.
- It is also concluded that decision tree may be the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

Introduction

- In this capstone project, our goal is to predict whether the Falcon 9 first stage will successfully land. SpaceX advertises Falcon 9 launches at around **$62 million**, while competitors may charge **over $165 million** per launch. A major reason for SpaceX's lower cost is its ability to **reuse the first-stage booster**. Therefore, accurately predicting whether the booster will land helps estimate launch costs and can be valuable for other companies aiming to compete with SpaceX.
- It's important to note that many unsuccessful landings are intentional—SpaceX sometimes performs controlled landings in the ocean.
- The central question we aim to answer is: **Given a set of launch features—such as payload mass, orbit type, launch site, and more—can we predict whether the Falcon 9 first stage will land successfully?**

Section 1

Methodology

Methodology

The overall methodology includes:

1. Data collection, wrangling, and formatting, using:
    - SpaceX API
    - Web scraping
2. Exploratory data analysis (EDA), using:
    - Pandas and NumPy
    - SQL
3. Data visualization, using:
    - Matplotlib and Seaborn
    - Folium
    - Dash
4. Machine learning prediction, using
    - Logistic regression
    - Support vector machine (SVM)
    - Decision tree
    - K-nearest neighbors (KNN)

Data Collection

- The dataset was assembled using several data collection techniques:
  - We retrieved launch data from the SpaceX API using GET requests.
  - The API response was decoded from JSON using the .json() method and then converted into a pandas DataFrame using json_normalize().
  - After loading the data, we performed cleaning operations, checked for missing values, and filled them where necessary.
  - Additionally, we used web scraping with BeautifulSoup to gather Falcon 9 launch records from Wikipedia.
  - Our goal was to extract the launch records from an HTML table, parse the data, and convert it into a pandas DataFrame for further analysis.

Data Collection – SpaceX API

- We sent a get request to the API and obtained the data and did some basic data wrangling

- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/data%20collection%20api.ipynb

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- We used beautifulsoup for webscraping and put that data into a pandas dataframe

- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/webscraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

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

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url)
        html_data.status_code
Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]: # Use soup.title attribute
        soup.title
Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

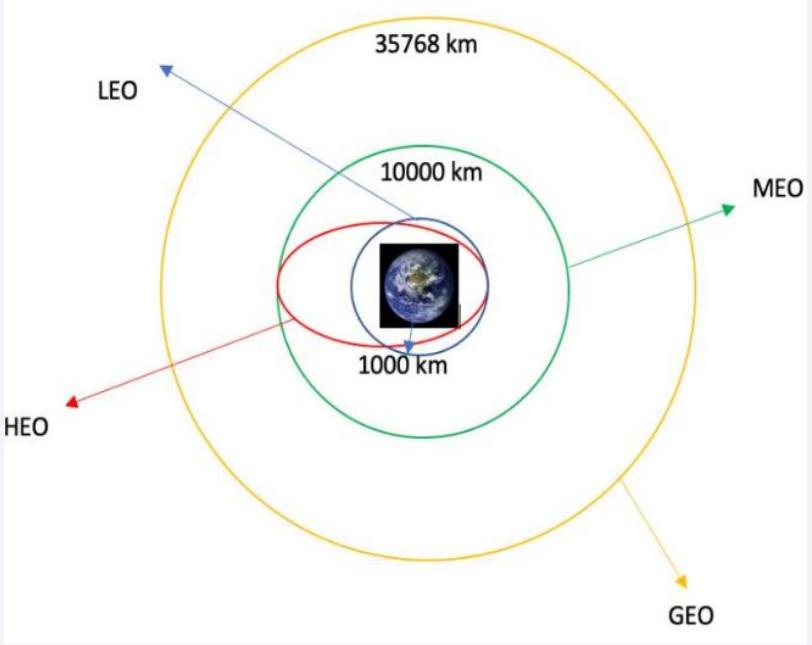
3. Extract all column names from the HTML table header

```
In [10]: 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

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

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

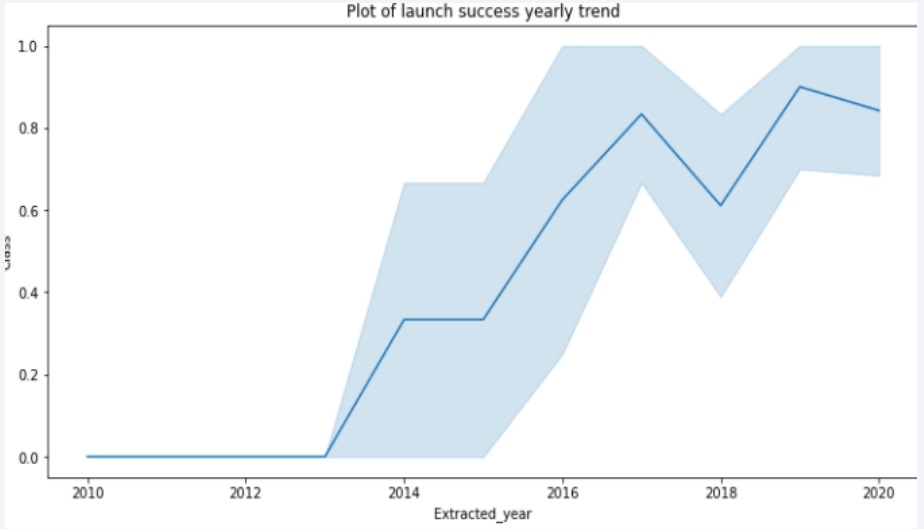
Data Wrangling

- We used EDA to analyze the data to determine the training factors.
- We calculated the number of launches from each site and the orbits
- We created an outcome table from the table
- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/data%20wrangling.ipynb



EDA with Data Visualization

- We explored the data by seeing which landing site,orbit,payload had the best success rate and the yearly success trend.
- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/data%20vizualization.ipynb



EDA with SQL

- We used a postgresql database to upload our data.
- We applied EDA to that database to get,
  - Names of launch sites
  - Total payload mass
  - Averge payload mass
  - Total success and failure outcomes
- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/sql.ipynb

Build an Interactive Map with Folium

- We marked the launch sites and added objects like lines rounds markers etc to show the data better.
- We assigned outcomes to the locations
- We used color based markers to show the outcomes
- We calculated the distance between a lunch site and other stuff near them
- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/folium.ipynb

Build a Dashboard with Plotly Dash

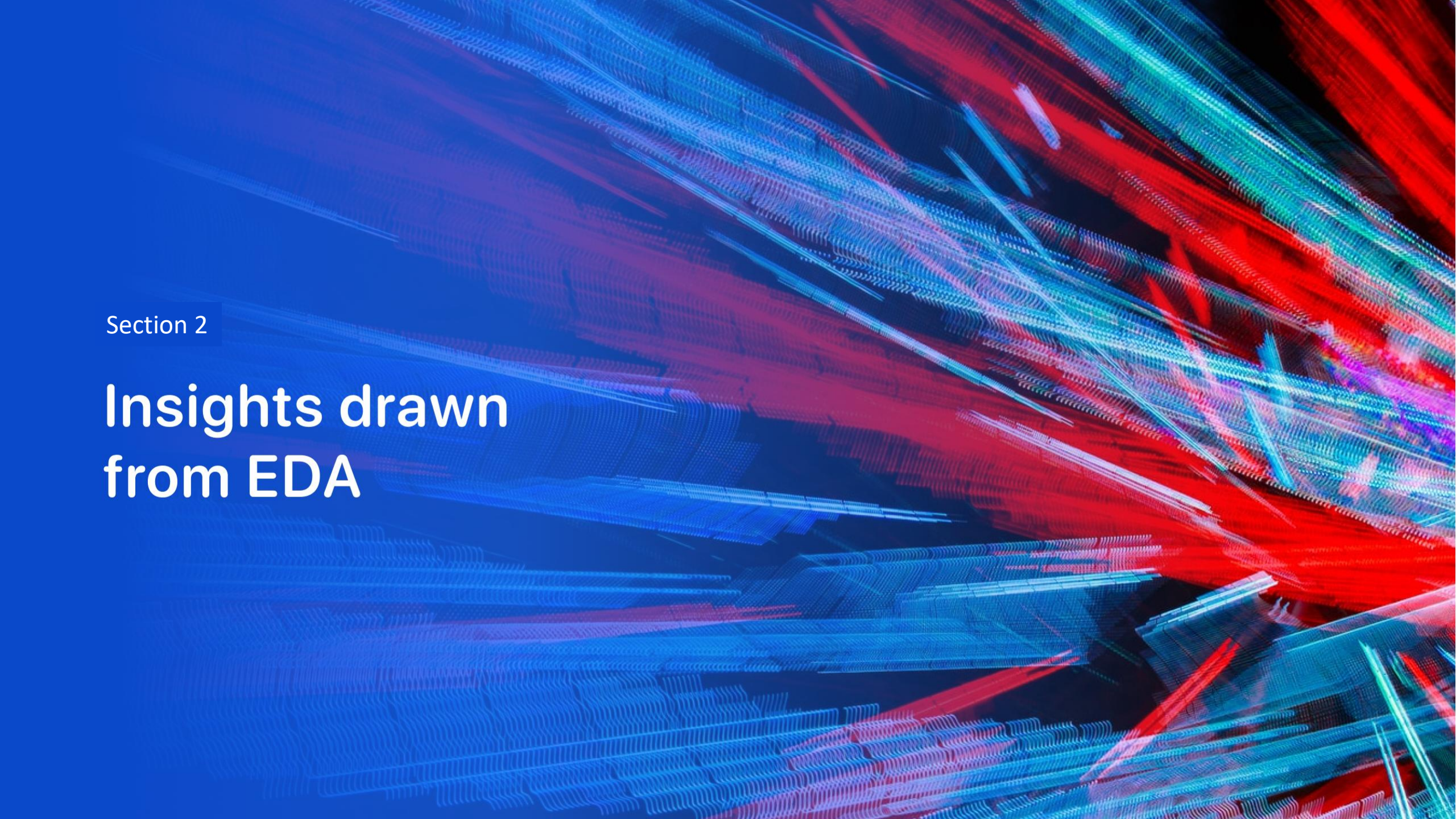
- We used dash to build an interactive dashboard
- We used it to present our charts showing total launches by a site
- We used it to show the relationship between payload and outcome
- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/dash.py

Predictive Analysis (Classification)

- We had the data in a pandas dataframe and transformed it using scikit learn and split into training and testing sets
- Then we used GridSearchCV to tune the different hyperparameters
- We used the accuracy as our main factor for this model and we further tuned it to improve accuracy
- We found the best model by doing so
- https://github.com/Piecodezz/IBM-data-science-capstone/blob/main/model.ipynb

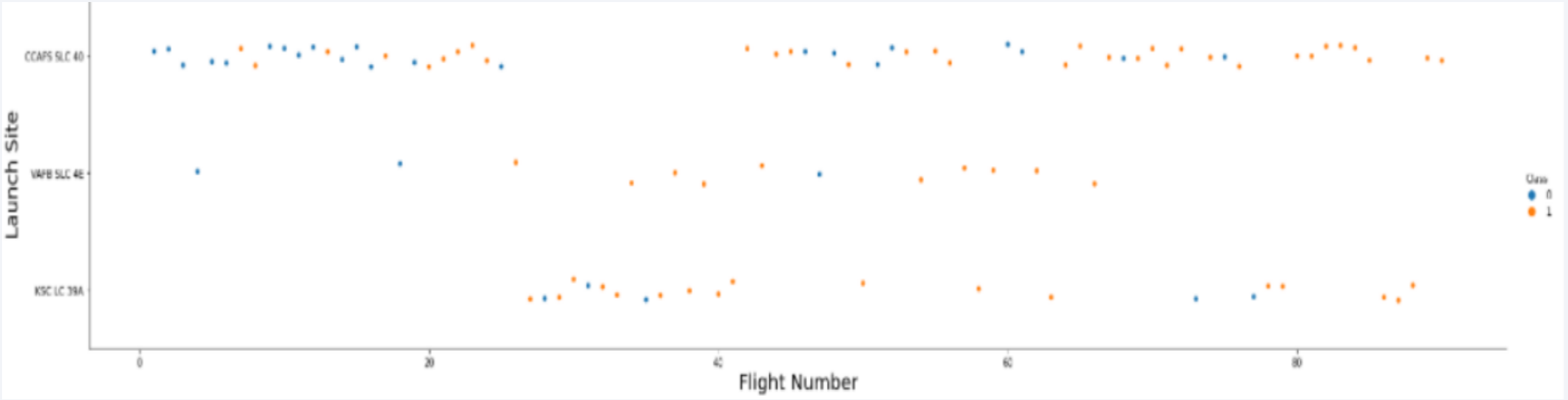
Results

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


Section 2
Insights drawn from EDA

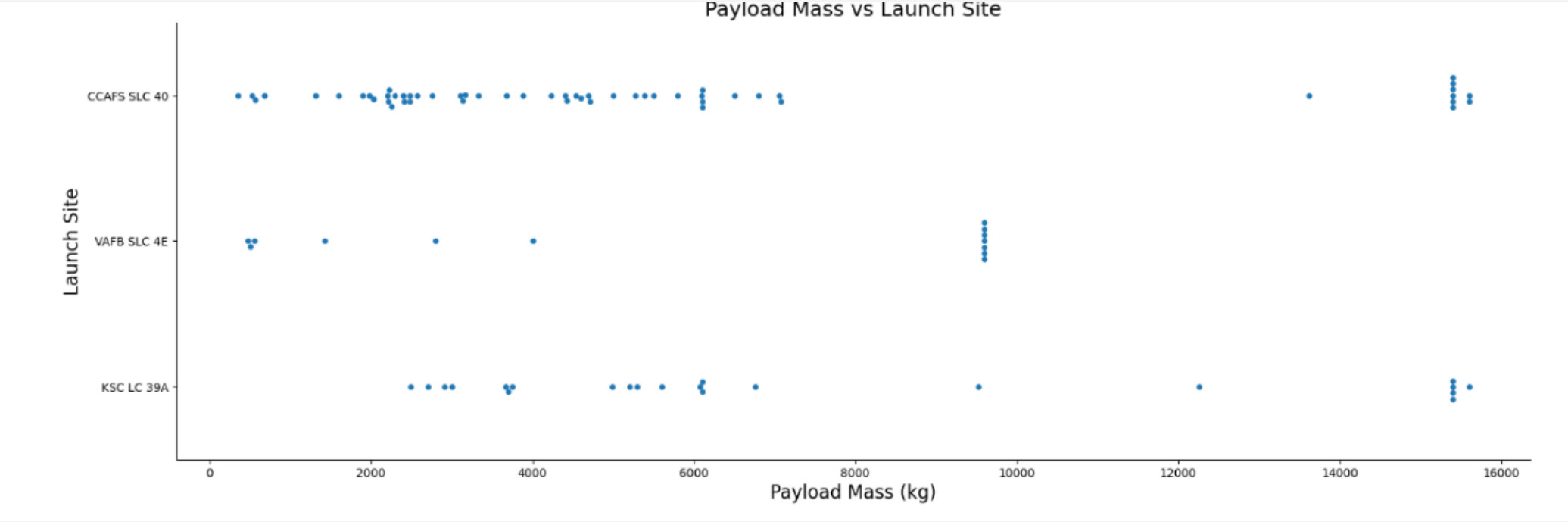
Flight Number vs. Launch Site

- We found that more flight numbers mean a greater success rate.



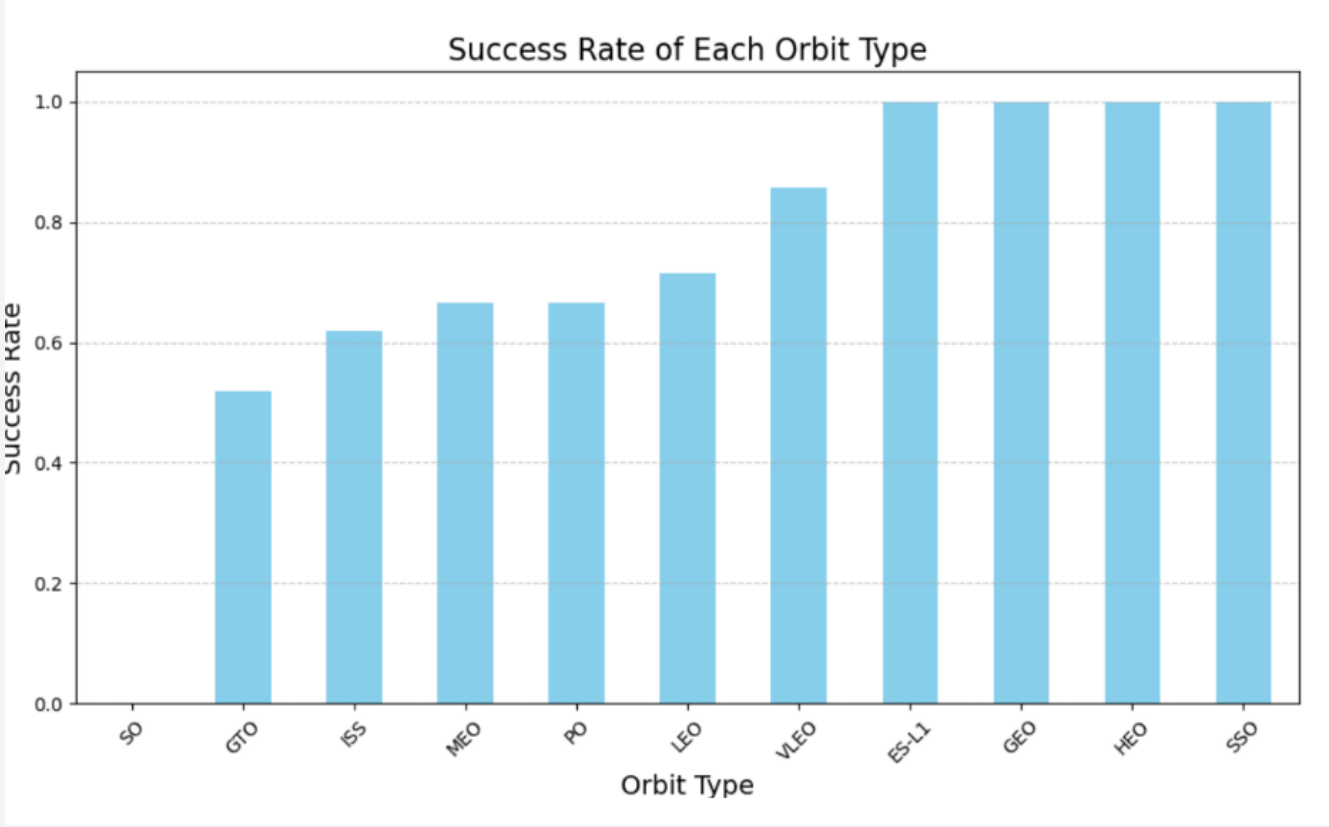
Payload vs. Launch Site

- Different launch sites have different capacities for example the CCAFS SLC 40 handle payloads below 10 000kg while others have a greater range of masses



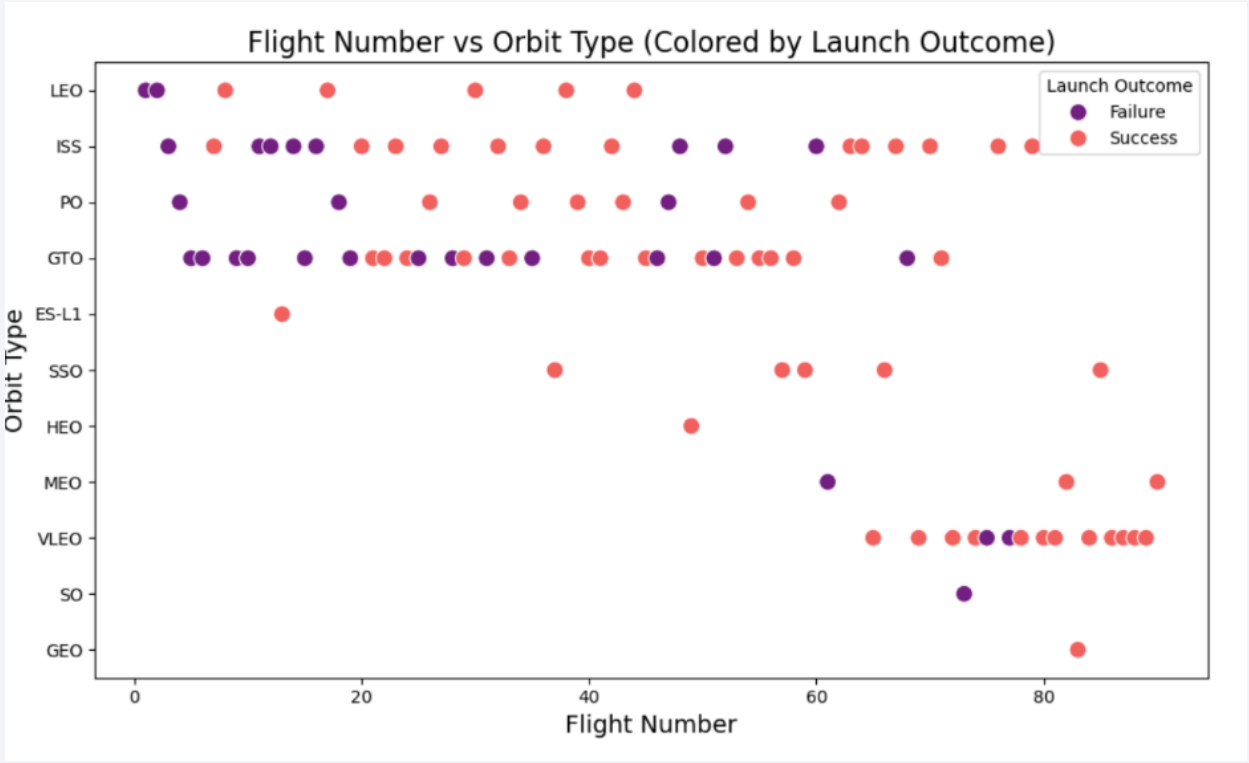
Success Rate vs. Orbit Type

- We can see that VLEO ES-L1, GEO, HEO, and SSO have the highest success rate
- And GTO has the lowest meaning its lower than the rest



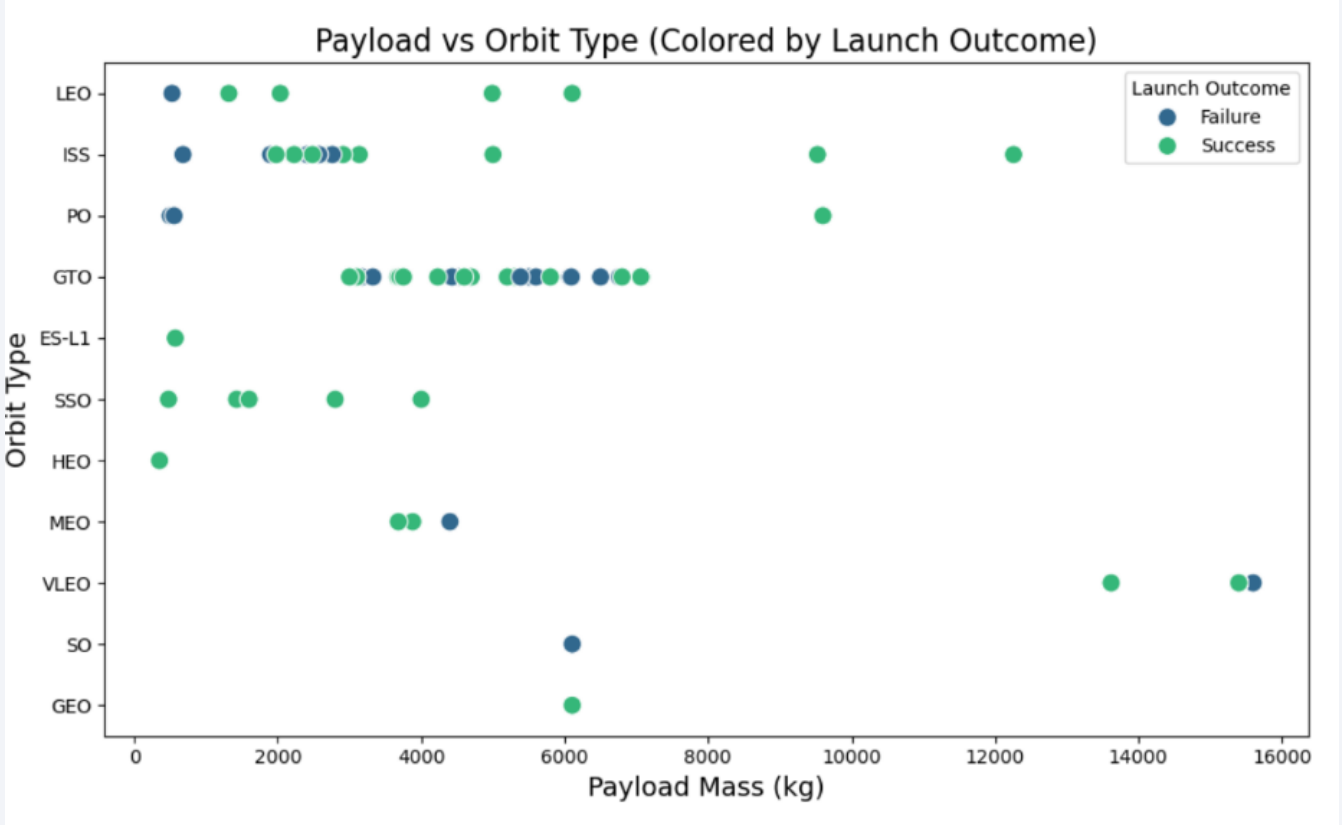
Flight Number vs. Orbit Type

- We can see as the flight number increases the success rate also increases
- And GTO and ISS orbits see better performance recently showing its improved



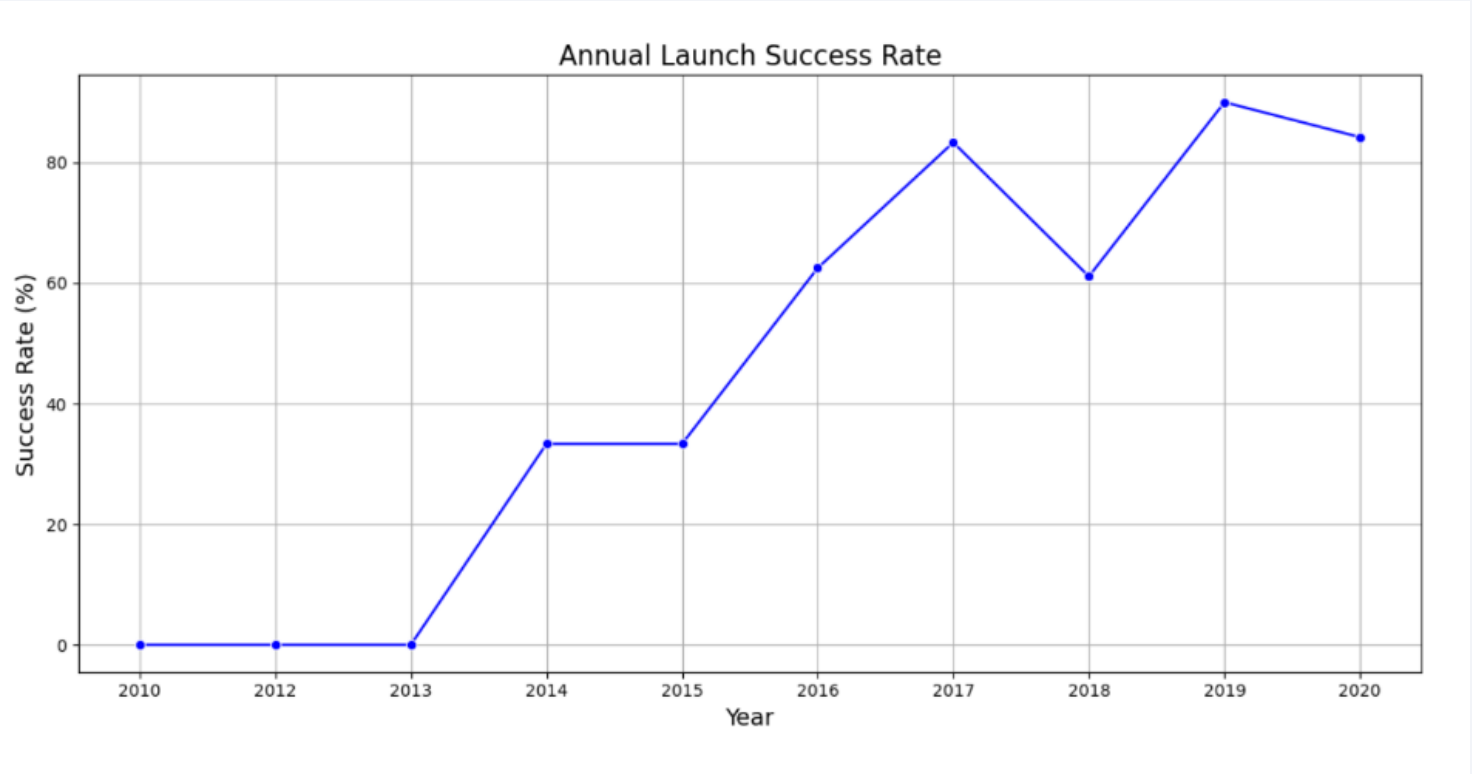
Payload vs. Orbit Type

- When the payload is less than 6000kg the success rate is higher
- When the payload is high the success rate drops



Launch Success Yearly Trend

- We can see a clear improvement over the years reaching over 80% by 2020
- This indicates the reliability of spaceX



All Launch Site Names

- We found the unique names of all the launch sites by SQL by using the keyword DISTINCT

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

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

Launch Site Names Begin with 'CCA'

- Then by using WHERE we found the launch sites starting with CCA

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

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We calculated the total mass by booster from NASA

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

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- We found the avg payload mass by the F9 v1.1 using SQL

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

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- We found the first successful landing date

In [14]:

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- By the aid of the where clause we found the successful drone ship landings and we used AND to determind the payload

In [15]:
```
task_6 = '''
          SELECT BoosterVersion
          FROM SpaceX
          WHERE LandingOutcome = 'Success (drone ship)'
              AND PayloadMassKG > 4000
              AND PayloadMassKG < 6000
          '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We found the count of the total successful and failed missions

List the total number of successful and failure mission outcomes

```
task_7a = '''
          SELECT COUNT(MissionOutcome) AS SuccessOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Success%'
          '''

task_7b = '''
          SELECT COUNT(MissionOutcome) AS FailureOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Failure%'
          '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

Boosters Carried Maximum Payload

- We found the boosters caried maximum payload using the MAX and WHERE keywords

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

In [17]:
```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- We used a combination of keywords like WHERE AND LIKE and BETWEEN to determind this data

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

In [18]:
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We used GROUP BY AND ORDER BY to determine these

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

In [17]:
```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

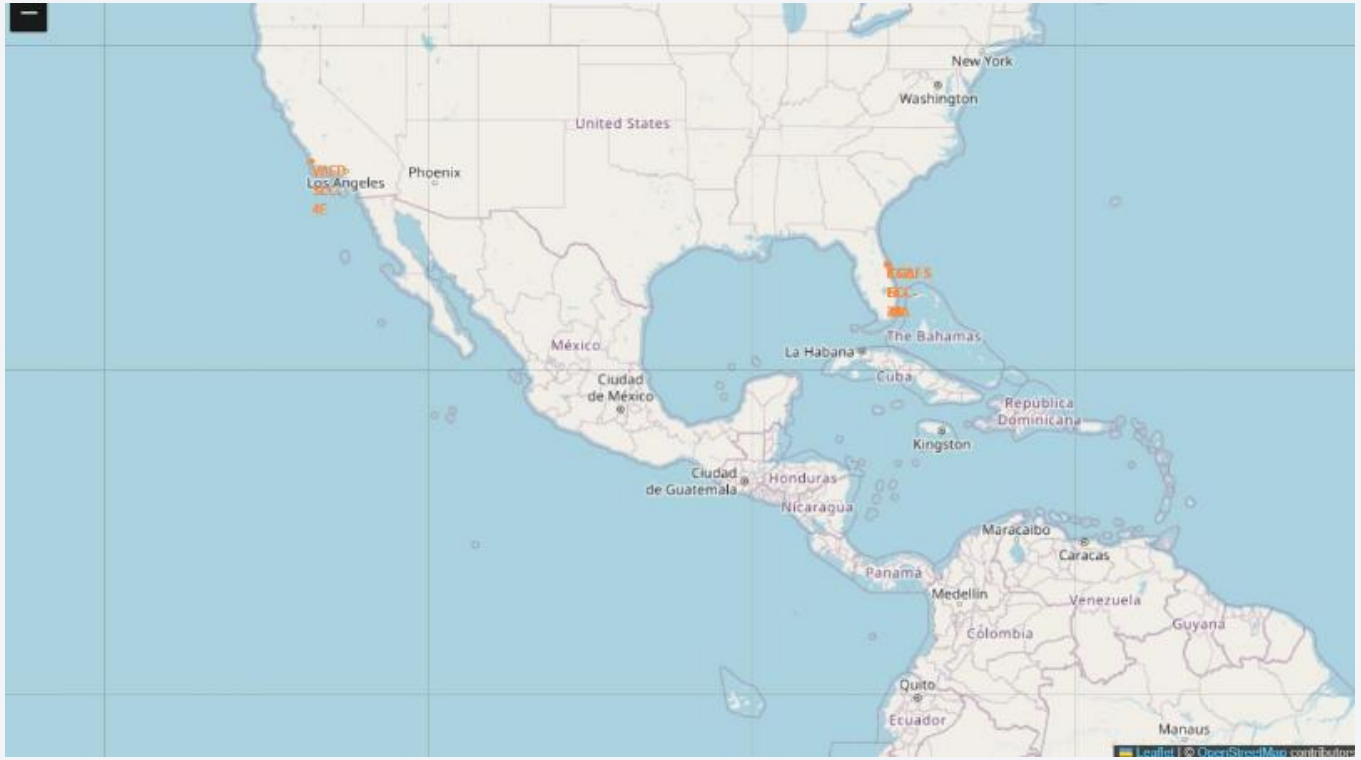
	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

Section 3

Launch Sites Proximities Analysis

All the launch sites on a map

- We can see all the launch sites are near to the coast including CCAFS LC-40 and CCAFS SLC-40 and the KSC LC-39A sites in florida and VAFB SLC-4E is also near the coast

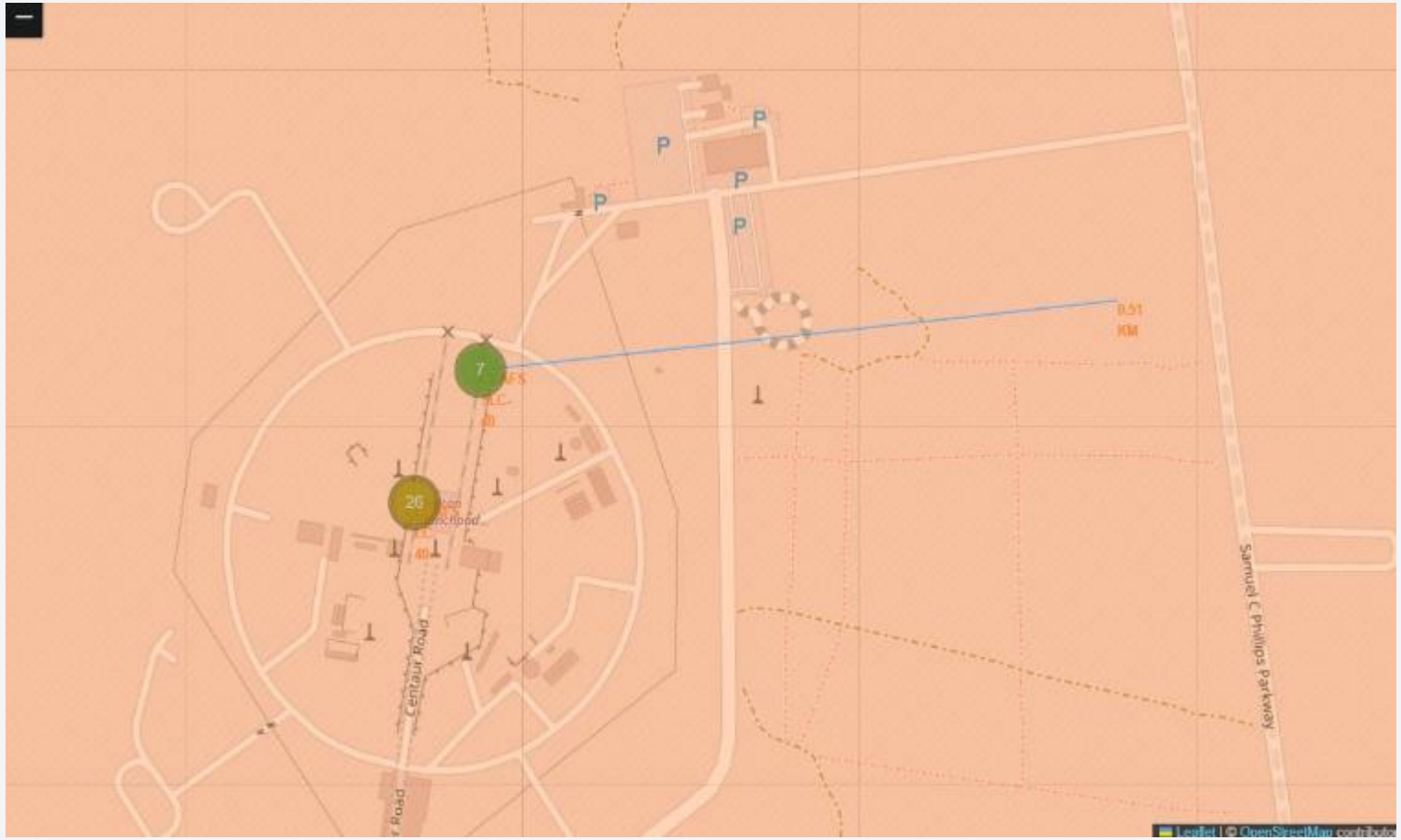


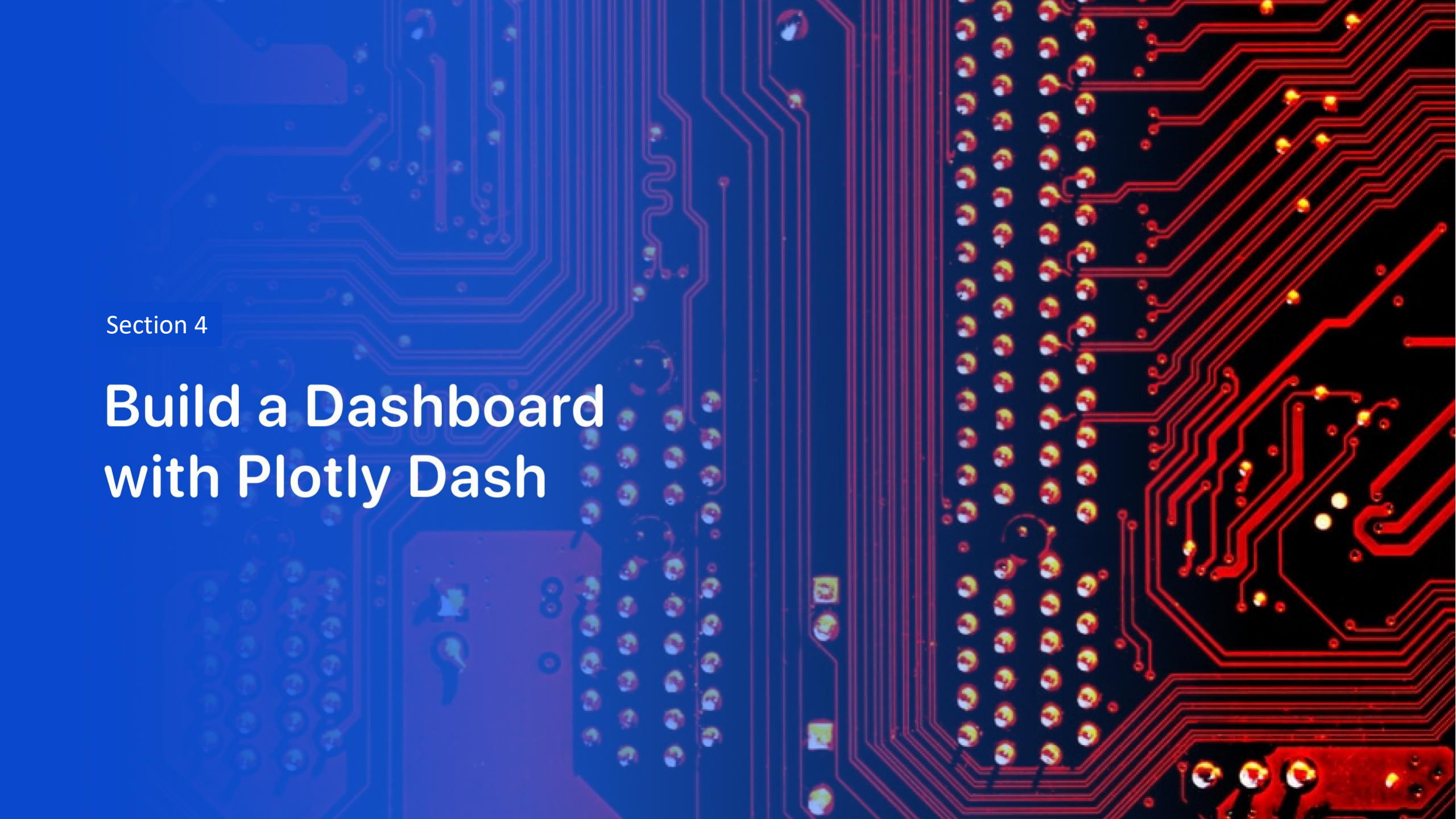
Successful and failed launches by site

- We used markers with green to indicate a success and a red one to indicate a failure clearly

Launch site and near proximities

- We used folium to display the diustance to the coastline




Section 4
Build a Dashboard with Plotly Dash

Launch success for all the sites

- We used a pie chart to display the success percentage for each site

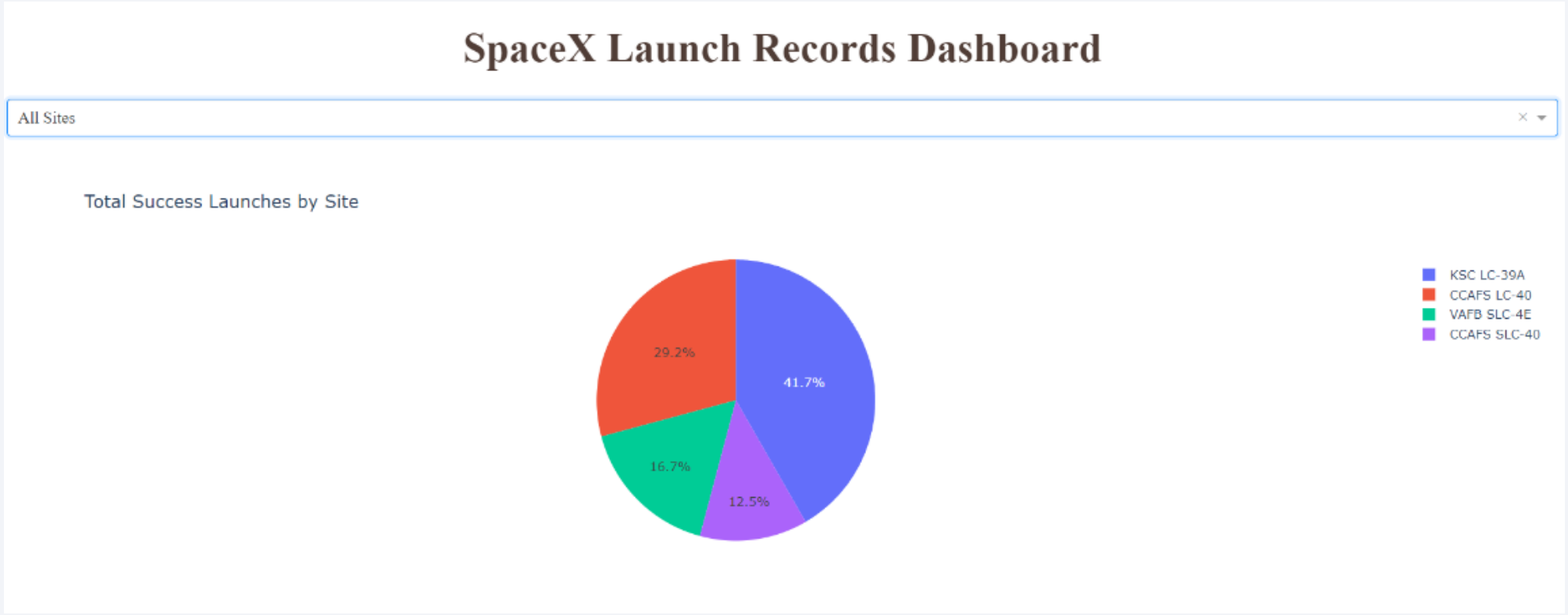
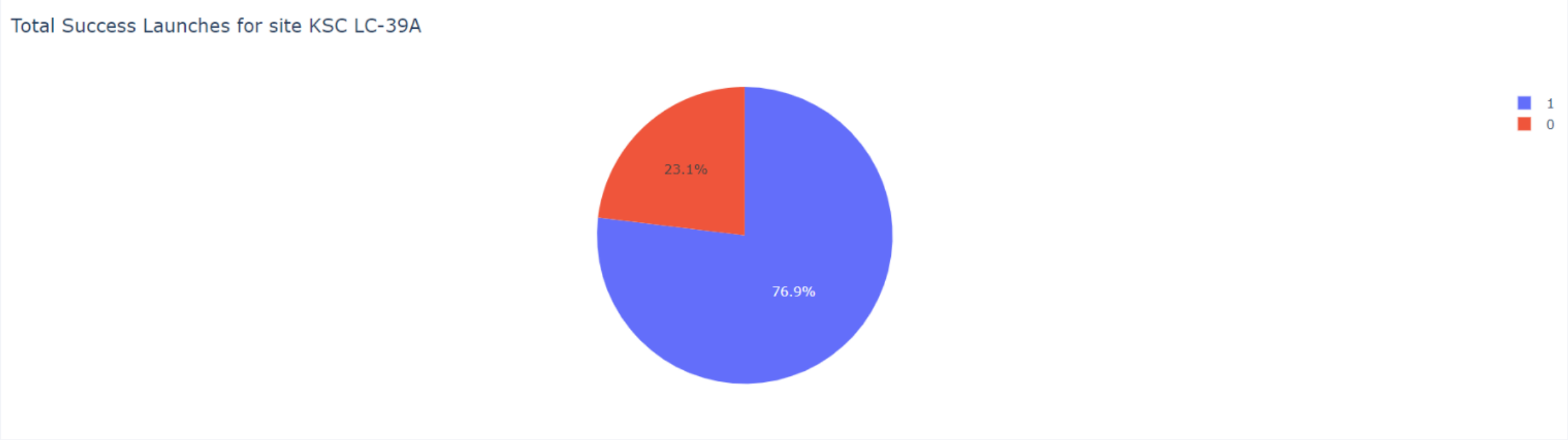


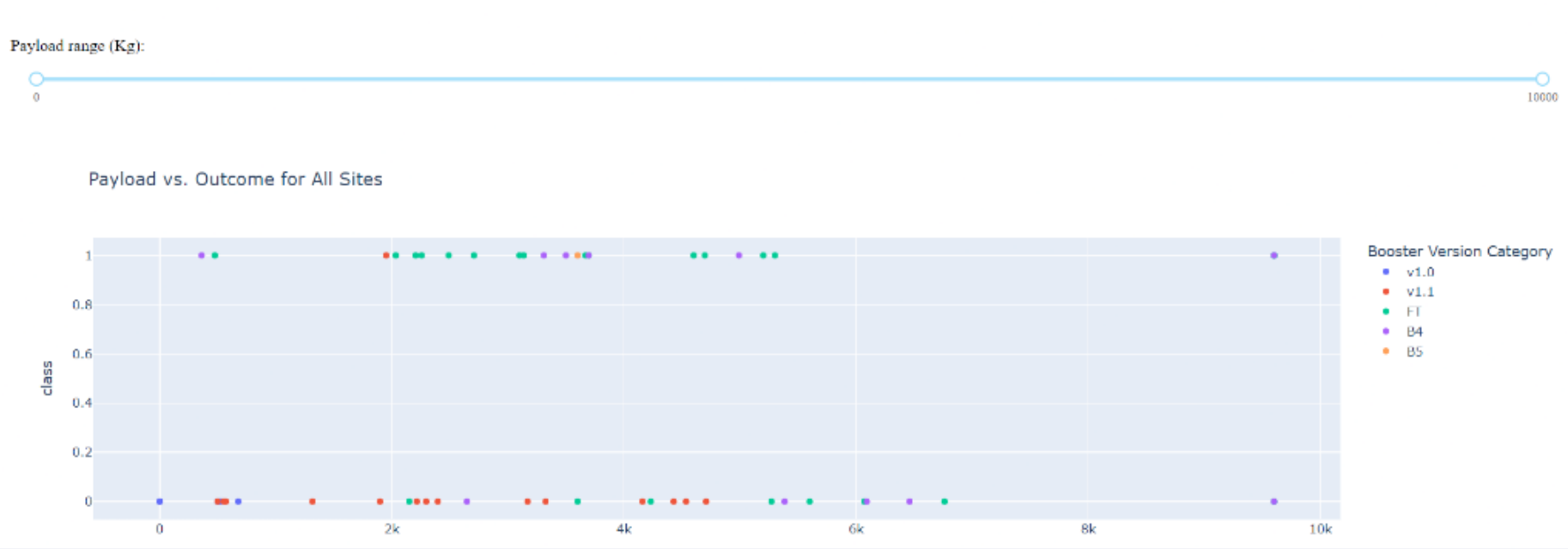
Chart to display the highest success ratio

- We used a pie chart to display this information too



Payload vs outcome charts on dash

- We used a scatter plot to plot this information and through all this we found that CCAFS LC-40 has the most successful launch rate and others have low rates
- But we were unable to find a drastic correlation between payload and outcome

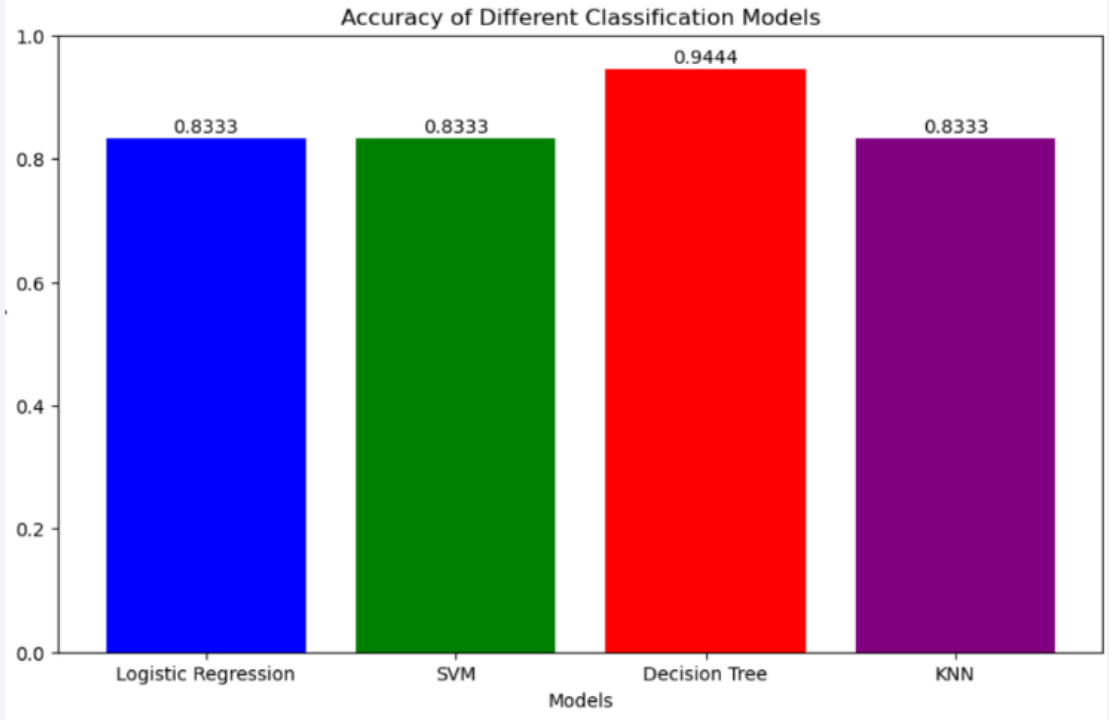


Section 5

Predictive Analysis (Classification)

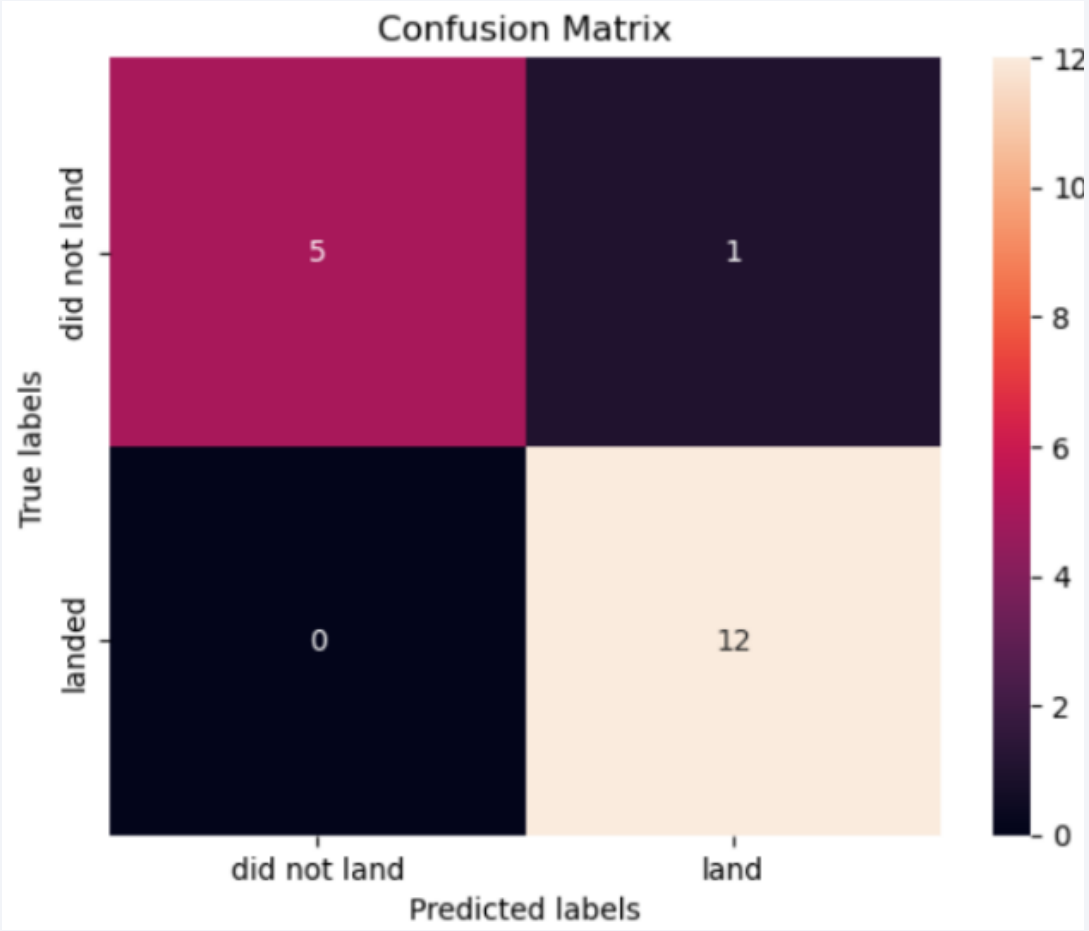
Classification Accuracy

- We found that Decision tree model performed the best while all the others like KNN and SVM all showed the same accuracy showing that Decision tree is the clear winner



Confusion Matrix

- We found that the model showed a 94% accuracy score and with no false negatives and only one false positive it shows a great performance.
- While there is one false positive it is very manageable and overall the model shows a very balances performance but it does have a small bias towards predicting successful launches but that aligns with our needs so this model is overall a good model



Conclusions

- **Point 1:**
  Our analysis shows that the **CCAFS LC-40** launch site has the **highest success rate**, contributing to **43.7% of all successful launches**. This suggests that the site may offer favorable conditions or operational advantages that consistently support successful landings.
- **Point 2:**
  The scatter plot analysis indicates that the **FT booster version** maintains a high success rate across a wide range of payload masses. This highlights its reliability and robustness compared to other booster versions and suggests that future missions could benefit from using this variant to improve success outcomes.
- **Point 3:**
  We found **no strong correlation** between higher payload mass and lower landing success rates. This implies that variables such as launch site conditions and booster version type have a more substantial impact on mission success than payload mass alone.

Thank you!