IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anirudh Sharma
03/11/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using APIs
  - Data Collection with Web Scraping
  - Data Wrangling
  - EDA with SQL
  - EDA with Data Visualization
  - Visualizations with Folium
  - Predictions using Machine Learning (Classification Models)
- Summary of all results
  - EDA Results
  - Visualization Results
  - Predictive Analysis Results

Introduction

- Project background and context
  - Space X Falcon 9 rocket launches cost of 62 million dollars; others cost up to 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the rocket will land in the first stage, we can estimate the cost of a launch.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The relationship between various parameters that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collection performed using APIs and web scrapping. Data sources were SpaceX and Wikipedia
- Perform data wrangling
  - Categorical columns were identified and one-hot encoding was performed
- 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

Part 1:

1. Application of get requests to SpaceX API
2. Decode the response in a json object. Normalize the data and convert to pandas dataframe
3. Data cleansing – handling missing values to replace them with mean values

Part 2:

1. Web scrapping the Falcon 9 and Falcon Heavy Launches data from Wikipedia
2. Parsing the scrapped HTML tables to pandas dataframe for further analysis

Data Collection – SpaceX API

- Using get requests, performing data formatting cleaning and wrangling
- Link to the GitHub notebook- https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/jupyter_labs_spacex_data_collection_api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

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

```
# Decode the response as JSON
json_data = response.json()

# Convert the JSON data into a Pandas DataFrame
data = pd.json_normalize(json_data)
```

+ Code + Text

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

```
[ ]  # Filter data for Falcon 9 launches
     data_falcon9 = launch_df[launch_df['BoosterVersion'] != 'Falcon 1']
     print(data_falcon9)
```

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

```
[ ]  # Calculate the mean of PayloadMass
     payload_mass_mean = data_falcon9['PayloadMass'].mean()

     # Replace np.nan values with the calculated mean
     data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, payload_mass_mean)
```

Data Collection - Scraping

- Web scrapping Falcon 9 launch records with BeautifulSoup
- Parsing the table and converting it into a pandas dataframe
- GitHub Notebook for webscrapping - https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/jupyter_labs_webscraping.ipynb

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

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[ ] response = requests.get(static_url)
```

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

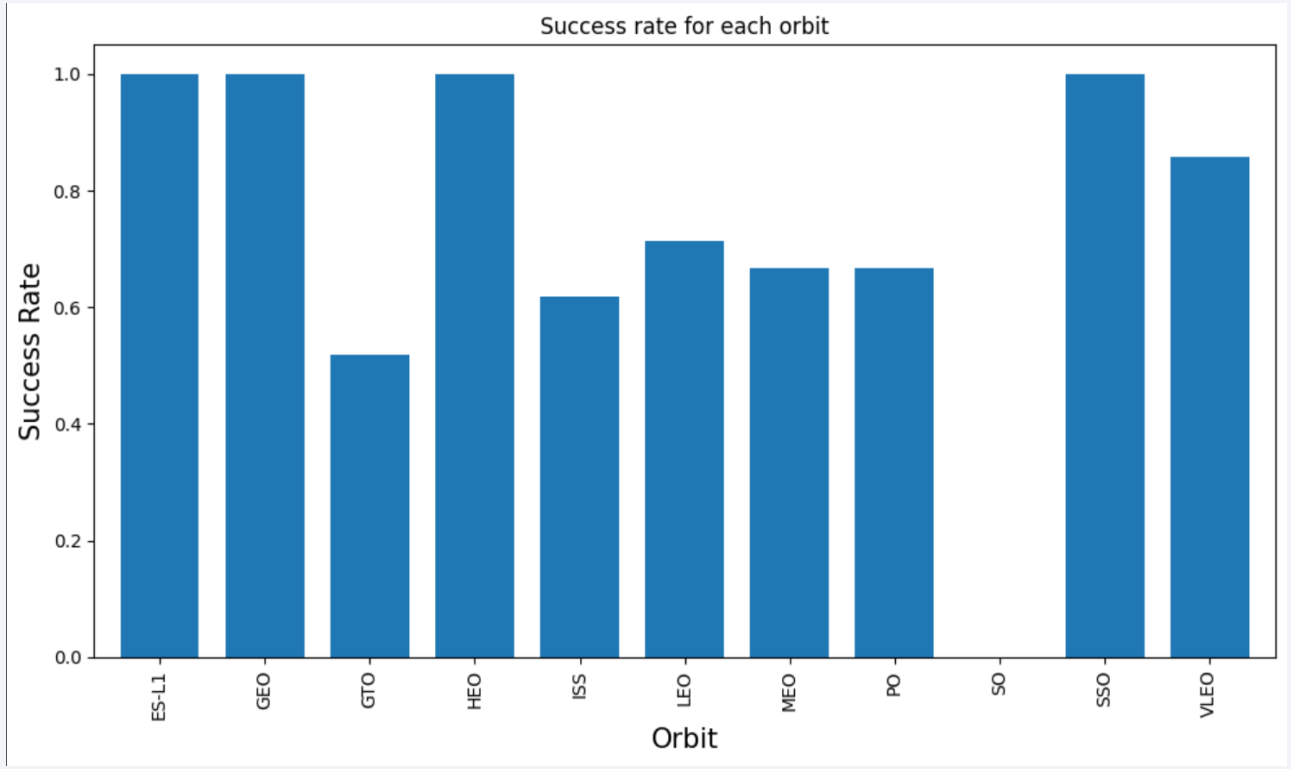
```
[ ] # Parse the HTML content
    soup = BeautifulSoup(response.content, "html.parser")
```

```
# Access the first table (assuming the desired table is the first one)
first_launch_table = html_tables[2]

# Find all <th> elements (table header cells) in the first row
table_header = first_launch_table.find_all('tr')[0].find_all('th')  # Get <th> elements from the first row

# Extract column names
column_names = []
for header_cell in table_header:
    name = extract_column_from_header(header_cell)
    if name is not None and len(name) > 0:
        column_names.append(name)

print(f"Extracted column names: {column_names}")
```

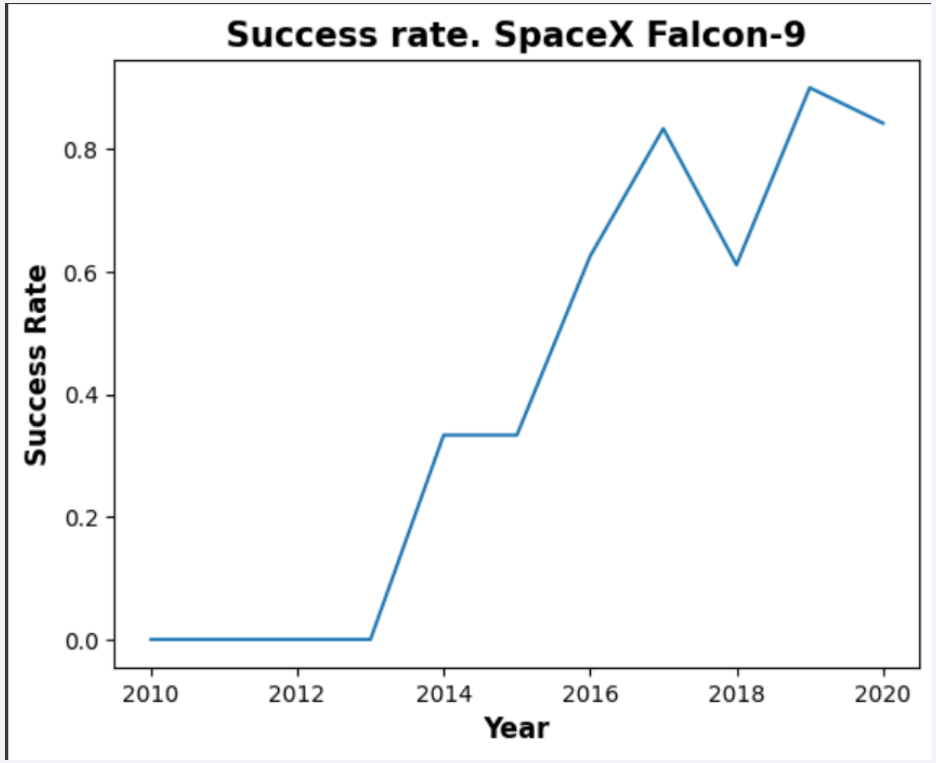
Data Wrangling

- Exploratory data analysis was performed and the training labels were determined.
- The number of launches at each site were calculated with the number and occurrence of each orbits
- Landing outcome label was created from outcome column and the results to csv were exported to a csv file
- GitHub URL: https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/labs_jupyter_spacex_Data_wrangling.ipynb

EDA with Data Visualization



GitHub link: https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/jupyter_labs_eda_dataviz.ipynb



The above figure shows the success rate of each orbit type. The figure on the right shows yearly success rate for Falcon 9 launcher. More graphs determining the relationship between flight number and launch Site, payload and launch site, etc., have been included in the GitHub notebook.

EDA with SQL

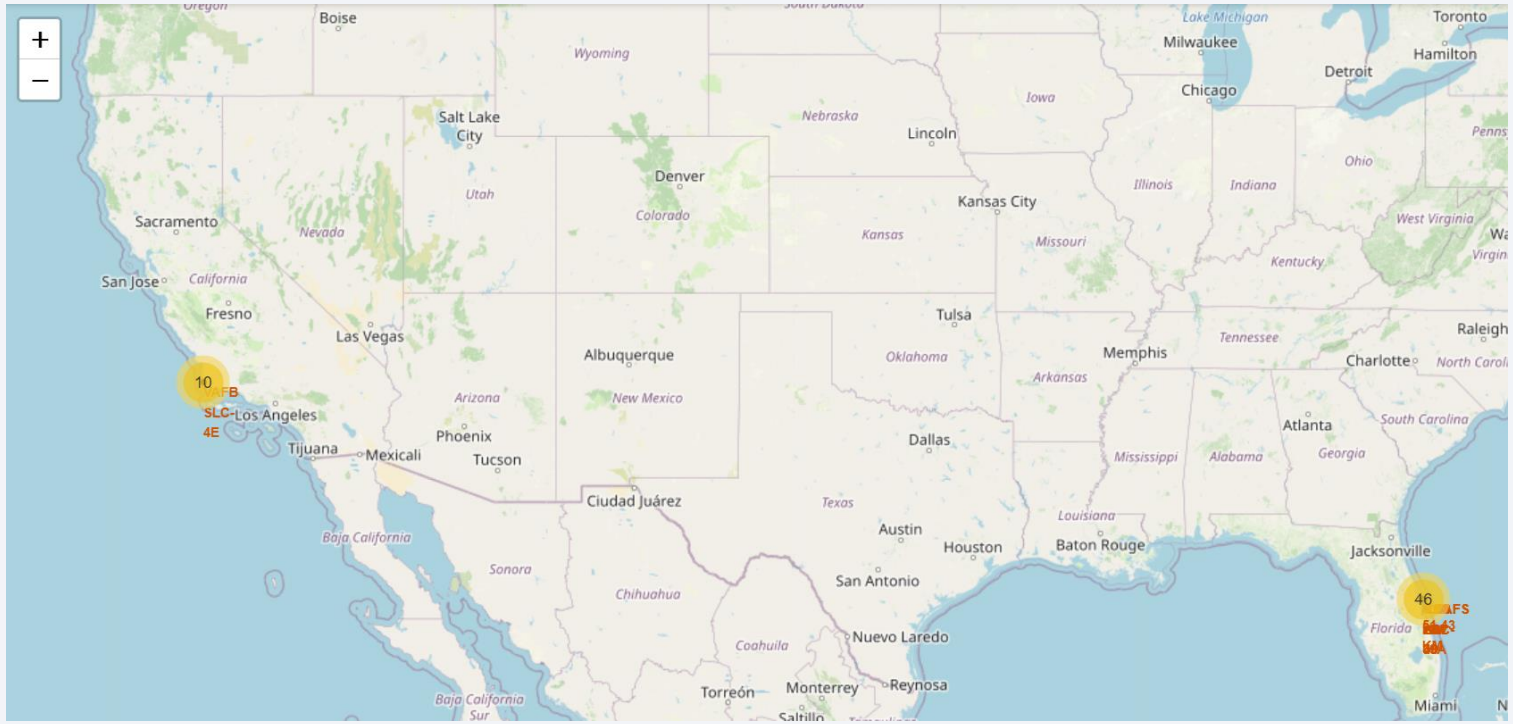
- Loaded the SpaceX dataset into a PostgreSQL database to be used in jupyter notebook
- Instances of EDA queries using SQL:

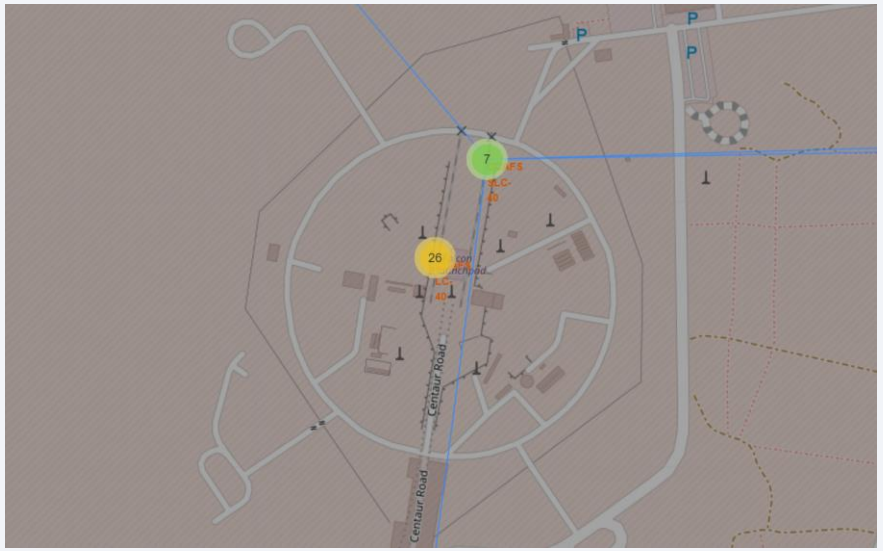
- Display the names of the unique launch sites in the space mission
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List 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.
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

GitHub link: https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/jupyter_labs_eda_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

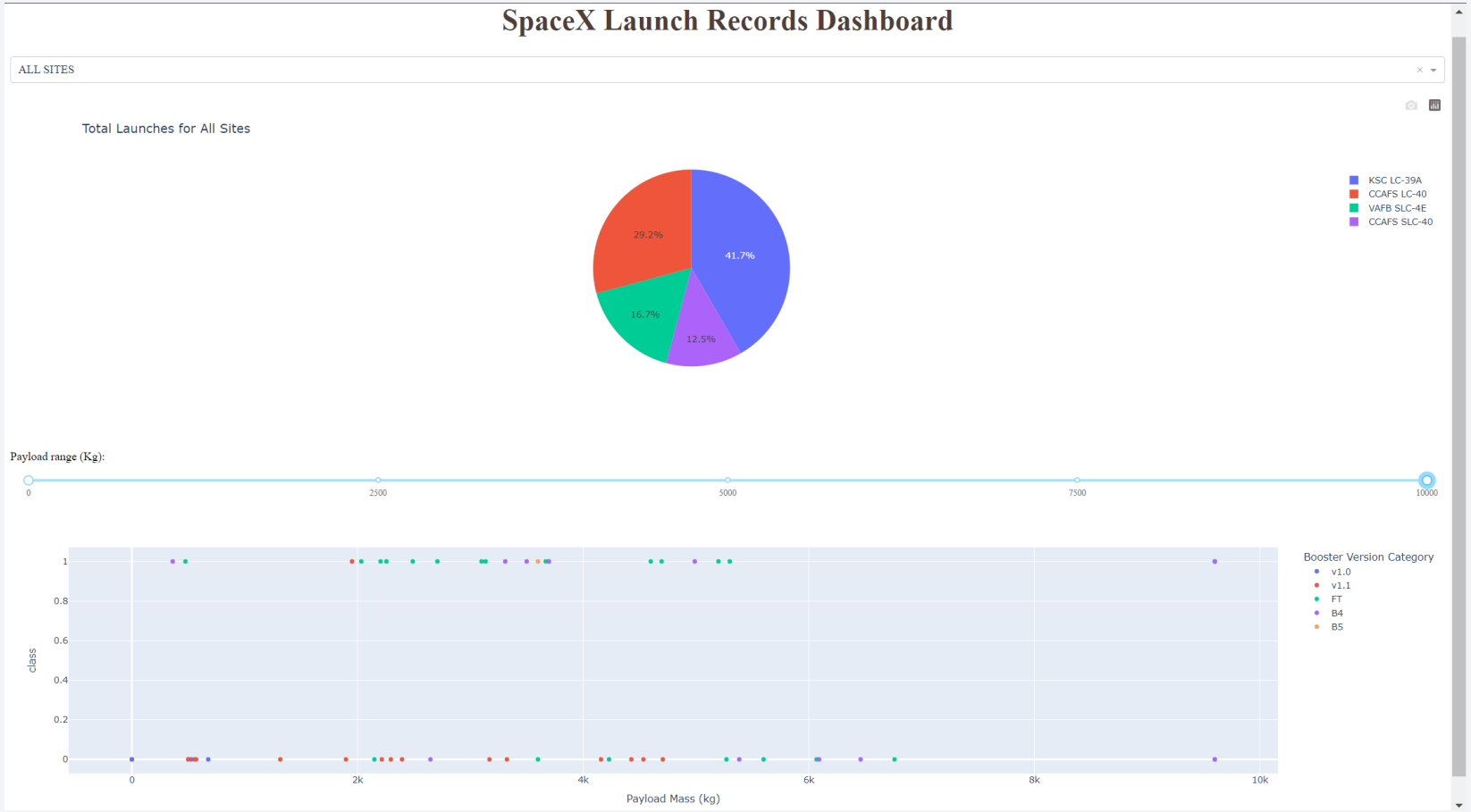
Markers added to the map showing optimal location for launch sites as shown:





GitHub URL: https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/lab_jupyter_launch_site_location_jupyterlite.ipynb

Build a Dashboard with Plotly Dash



Two graphs in the figure are for:

1. Total launches per site with site selection through a drop-down bar, results displayed via a pie-chart
2. The payload mass carried by different booster versions displayed via scatter plot and dynamic scroll bar for payload mass

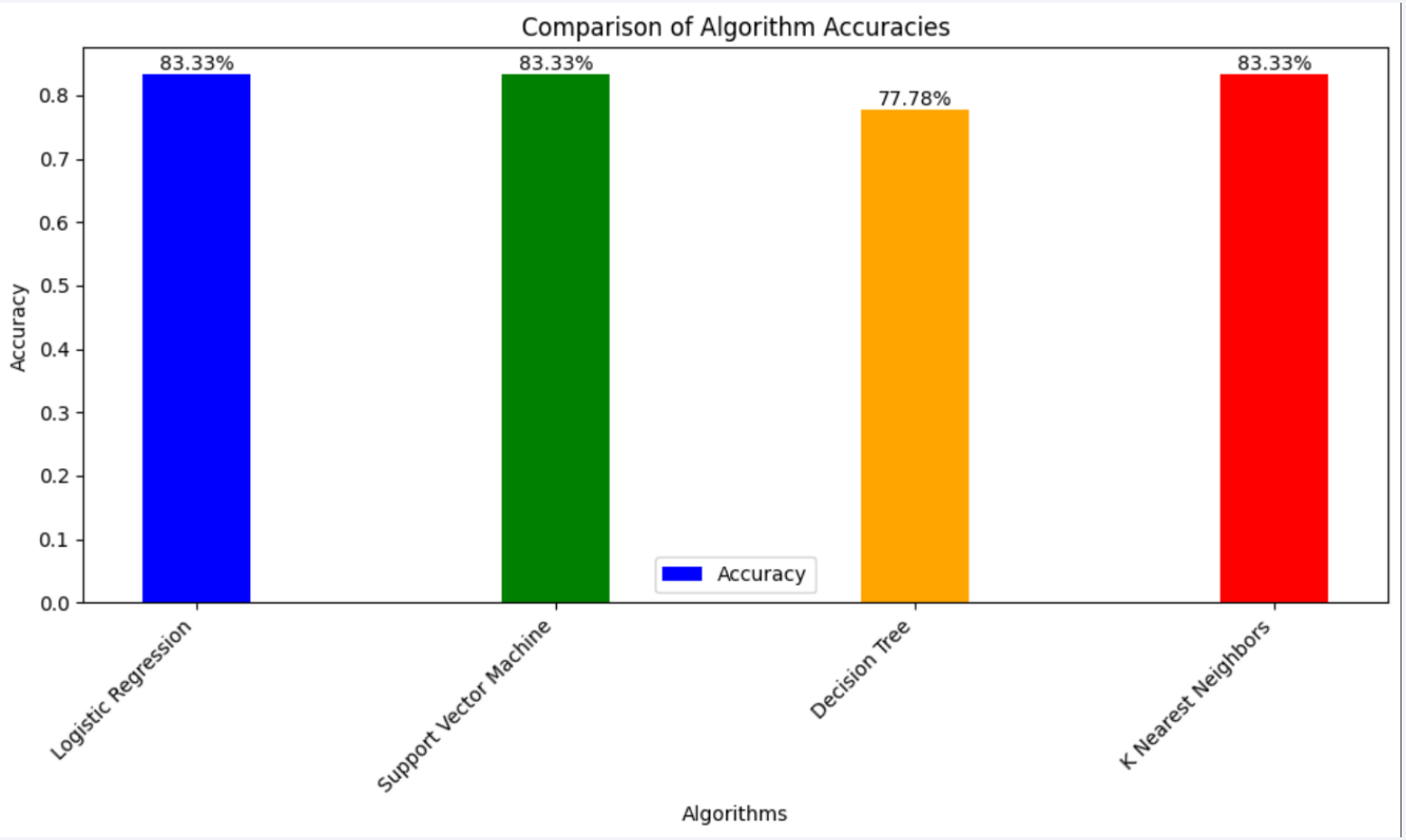
GitHub URL: https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/spacex_dash_app.py

Predictive Analysis (Classification)

Logistic Regression, SVM and k-NN algorithms perform best for the data with a predictive accuracy of 83.33%.

Data was loaded using numpy and pandas, then we transformed the data, and split our data into training and testing.

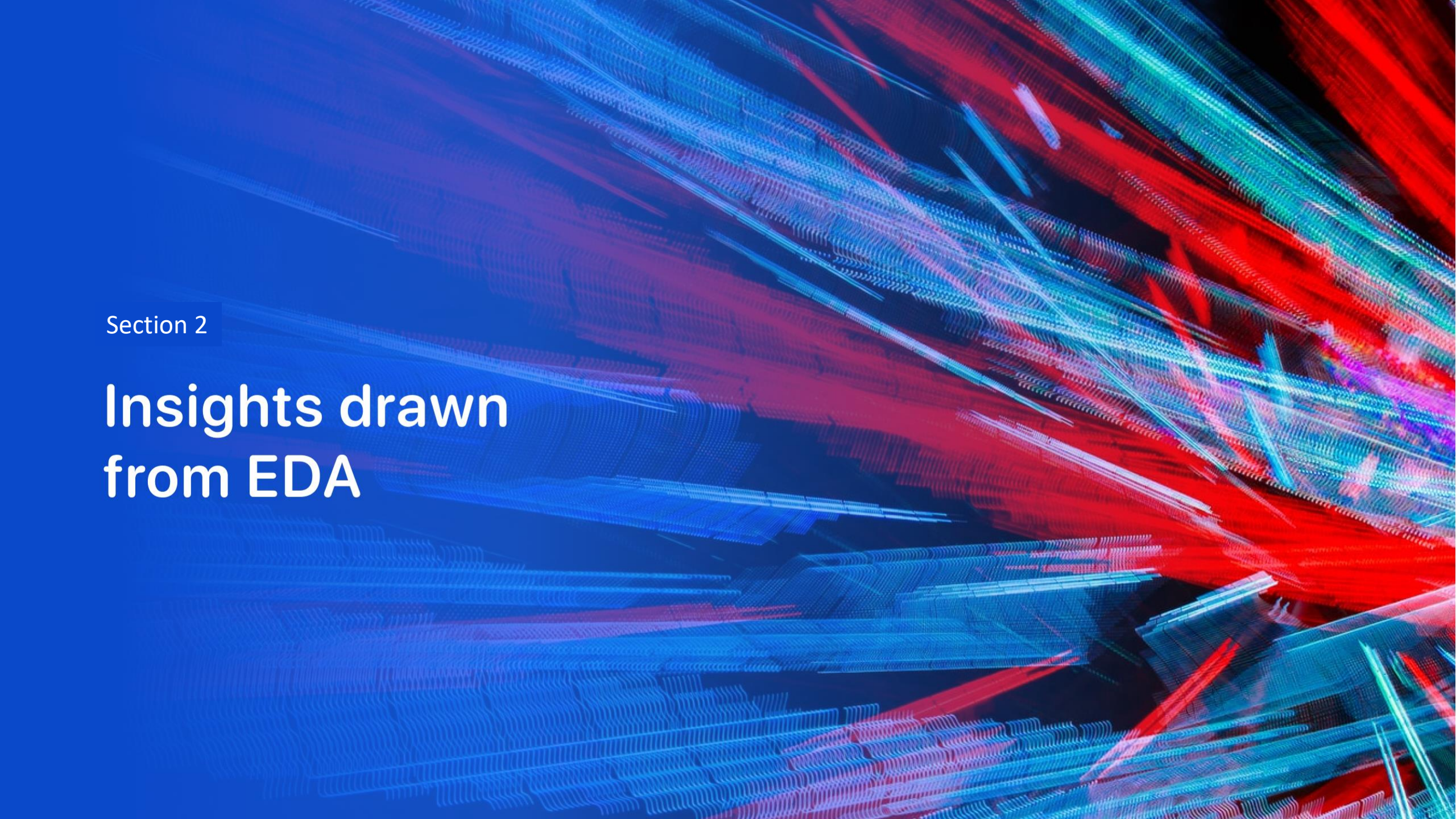
We built different machine learning models and tune different hyperparameters using GridSearchCV.



GitHub URL: https://github.com/anirudh-data-science/DataScience_Projects/blob/Capstone_Project/SpaceX_Machine_Learning_Prediction_Part_5_jupyterlite.ipynb

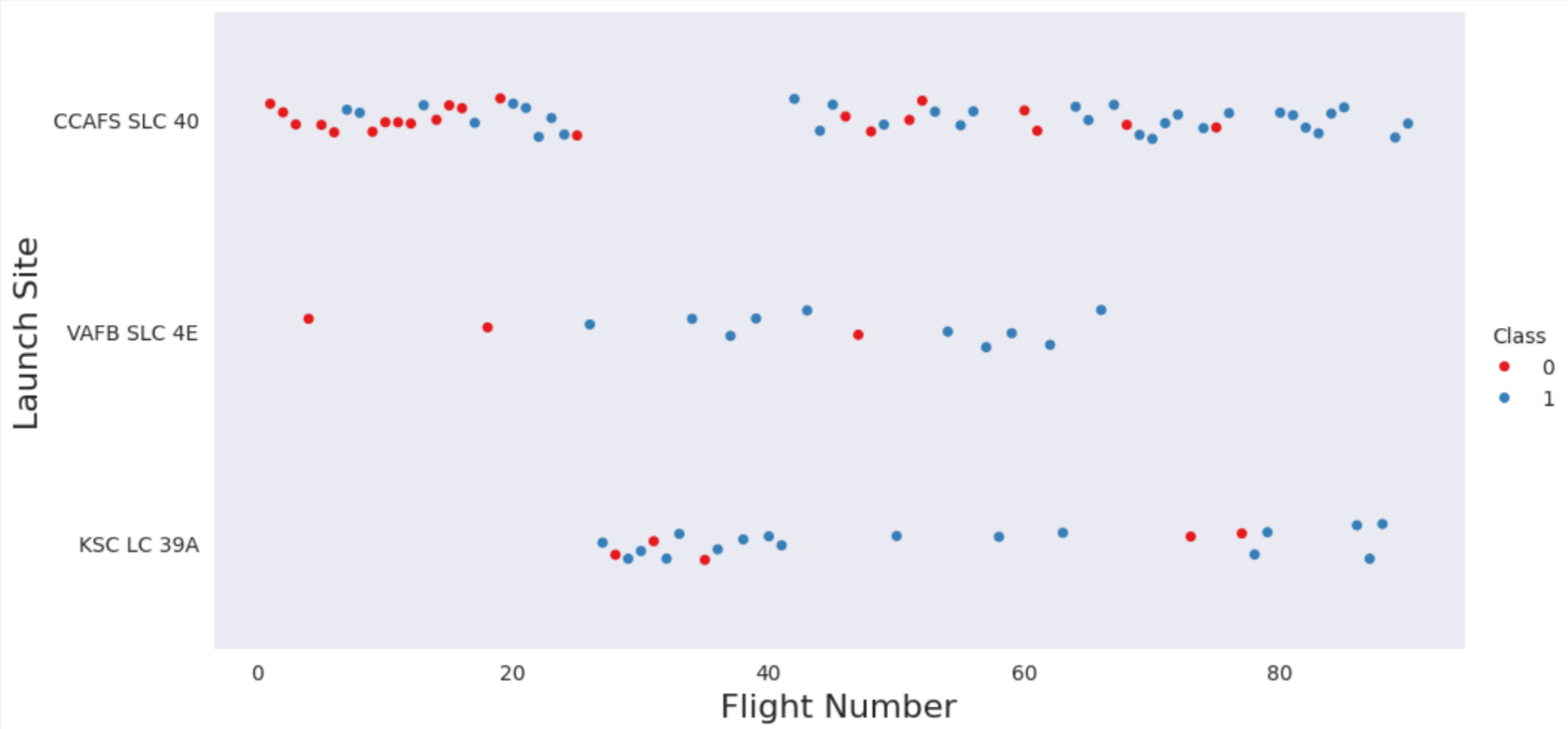
Results

- K-NN, SVM and LR algorithms perform the best in terms of predictive accuracy
- Launchers with lower payloads have higher success rates over higher payloads
- KSC LC 39A has the highest success rate amongst all the launch sites
- Orbits GEO, SSO, HEO, ES L1 have the best success rates


Section 2
Insights drawn from EDA

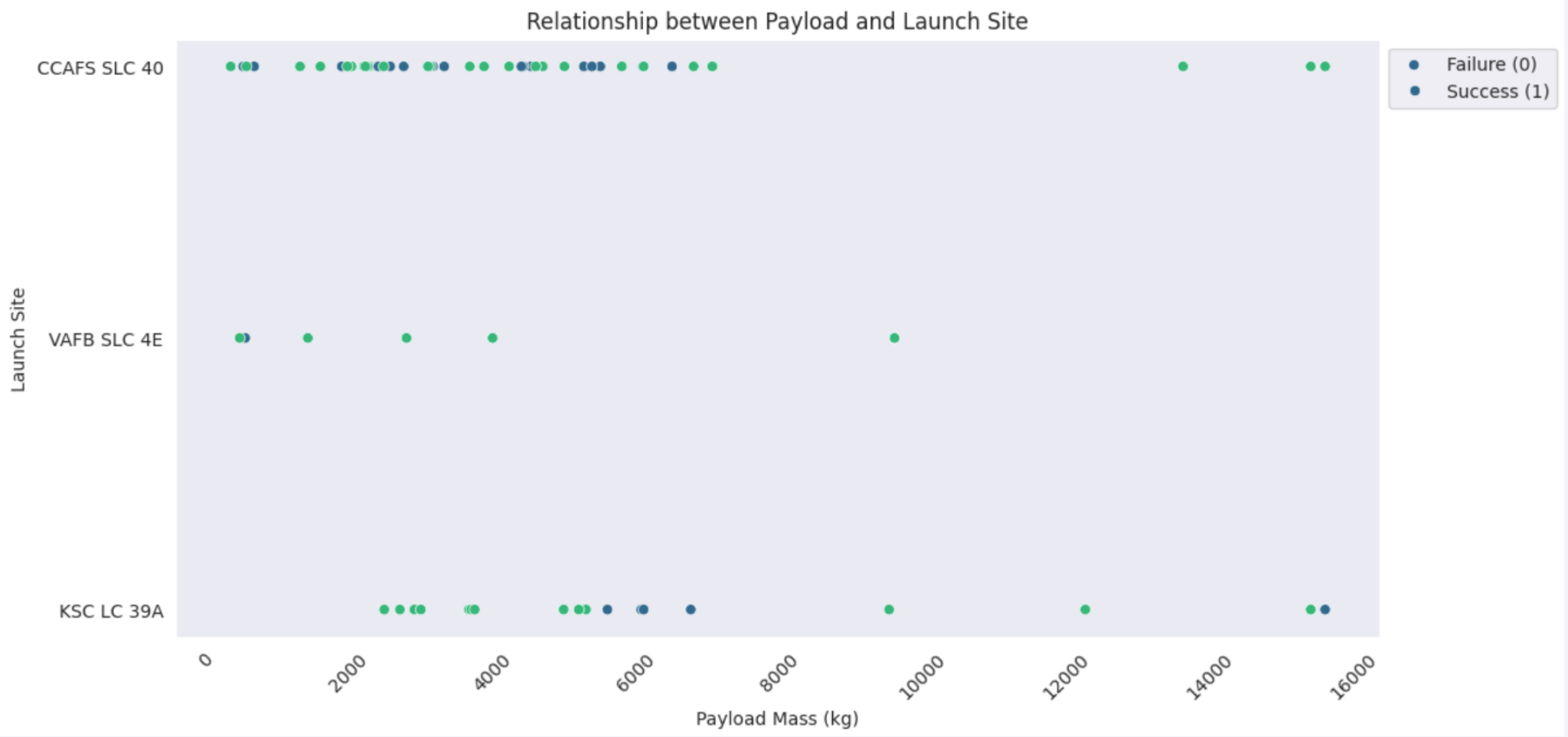
Flight Number vs. Launch Site

CCAFS SLC 40 has maximum number of



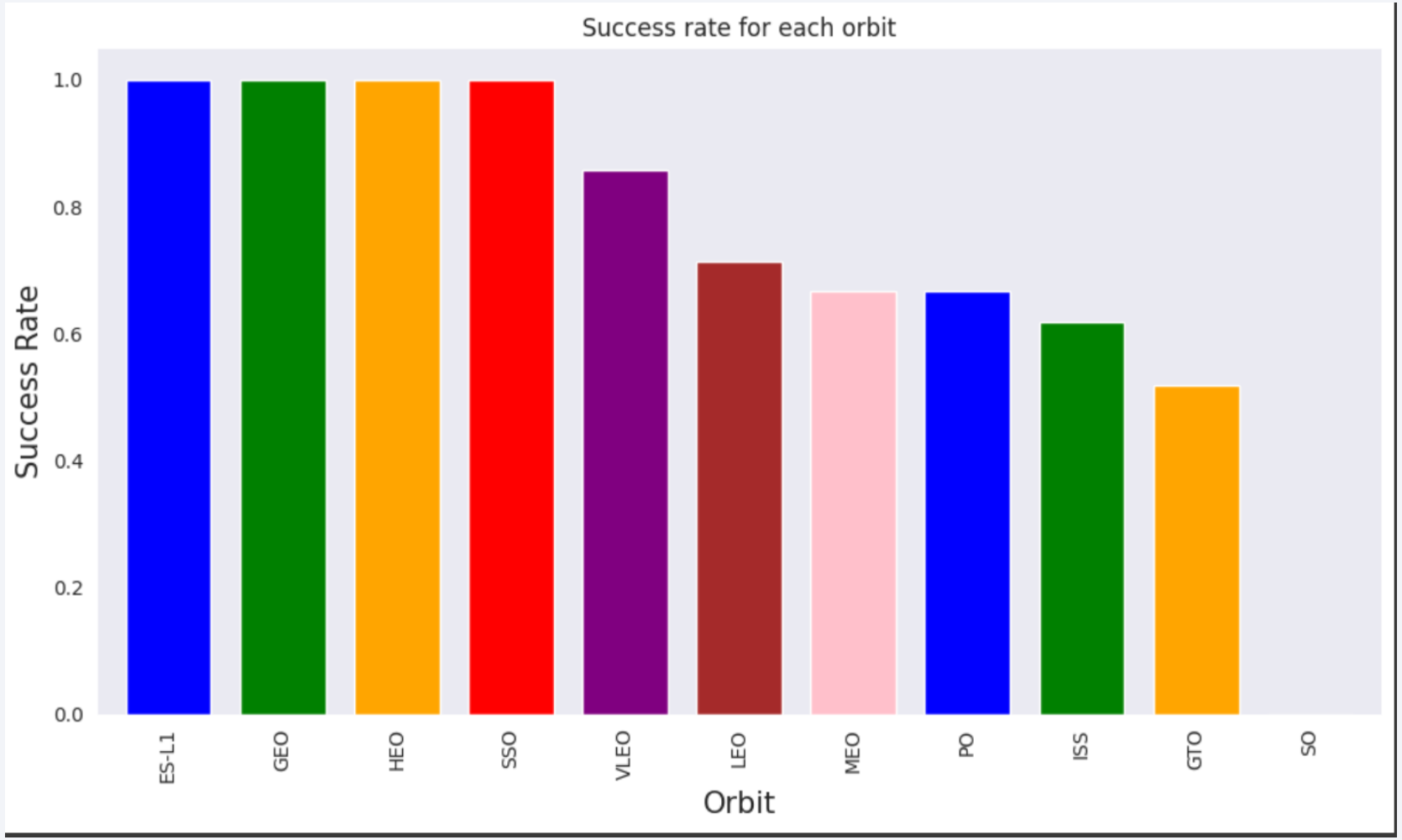
Payload vs. Launch Site

- Most of the payloads with lesser mass have been launched from CCAFSSLC 40

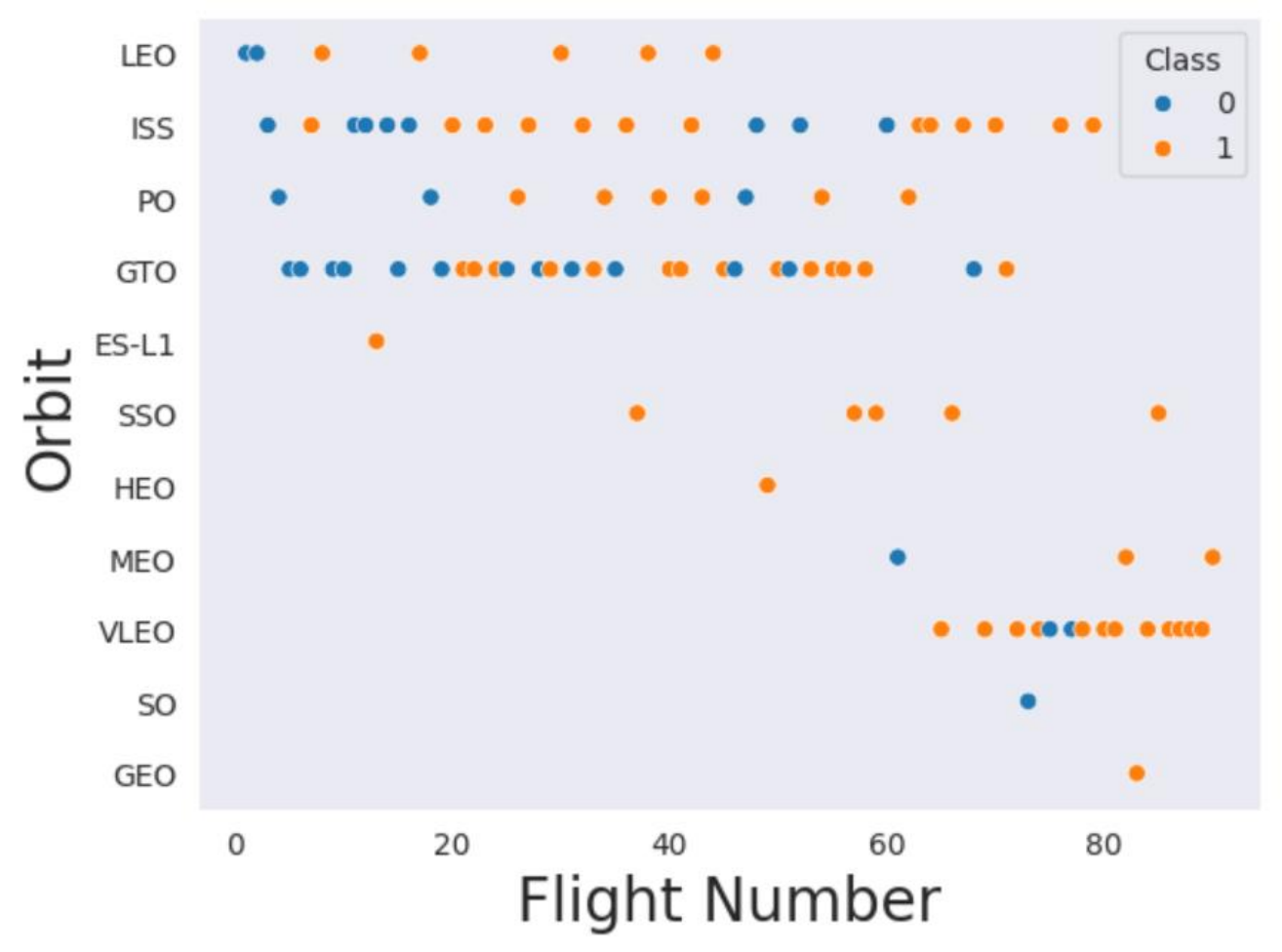


Success Rate vs. Orbit Type

Orbits ES-L1, GEO, HEO and SSO have the highest success rates

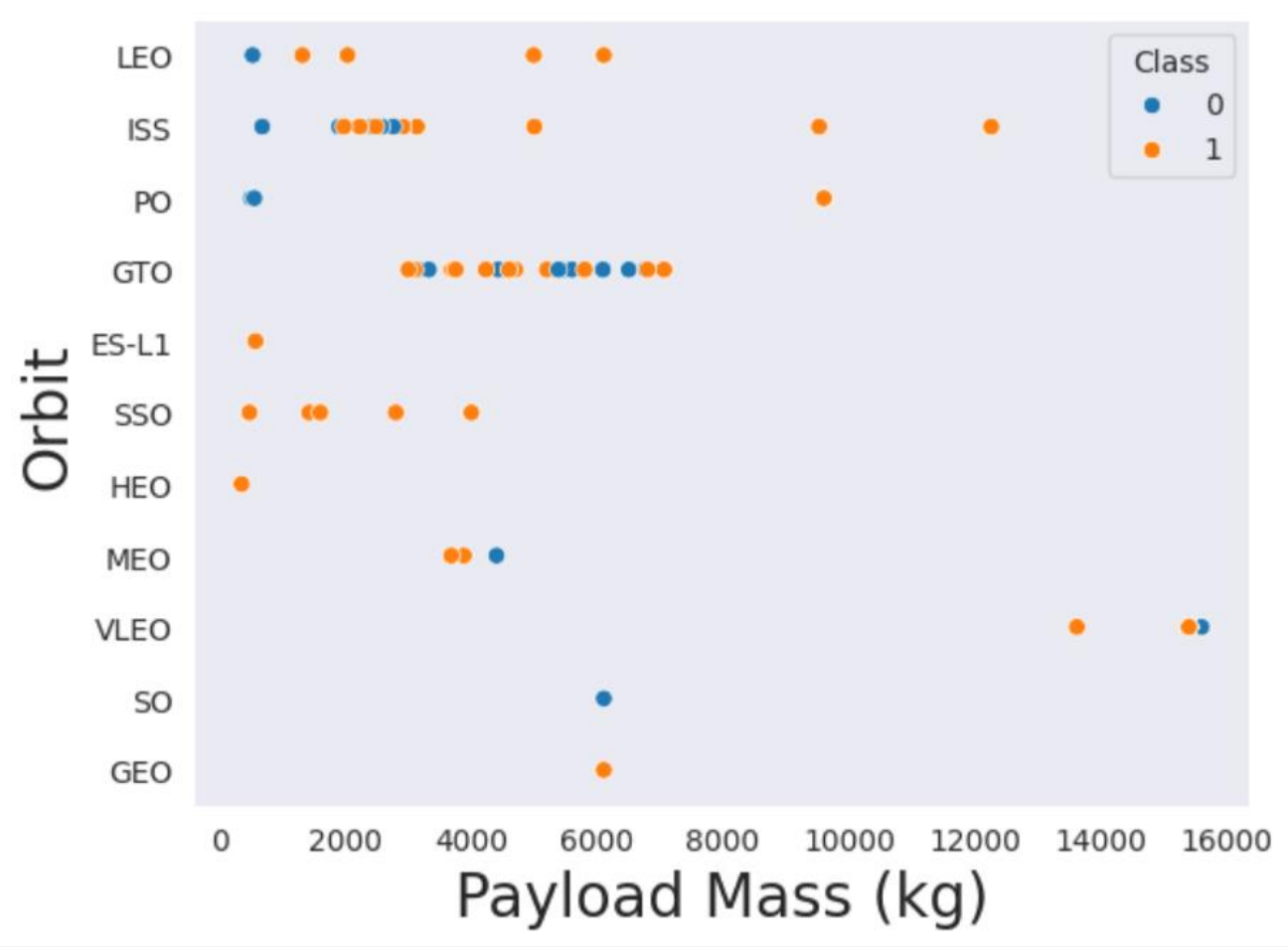


Flight Number vs. Orbit Type



Number of successful flights have increased from the VLEO location over recent years

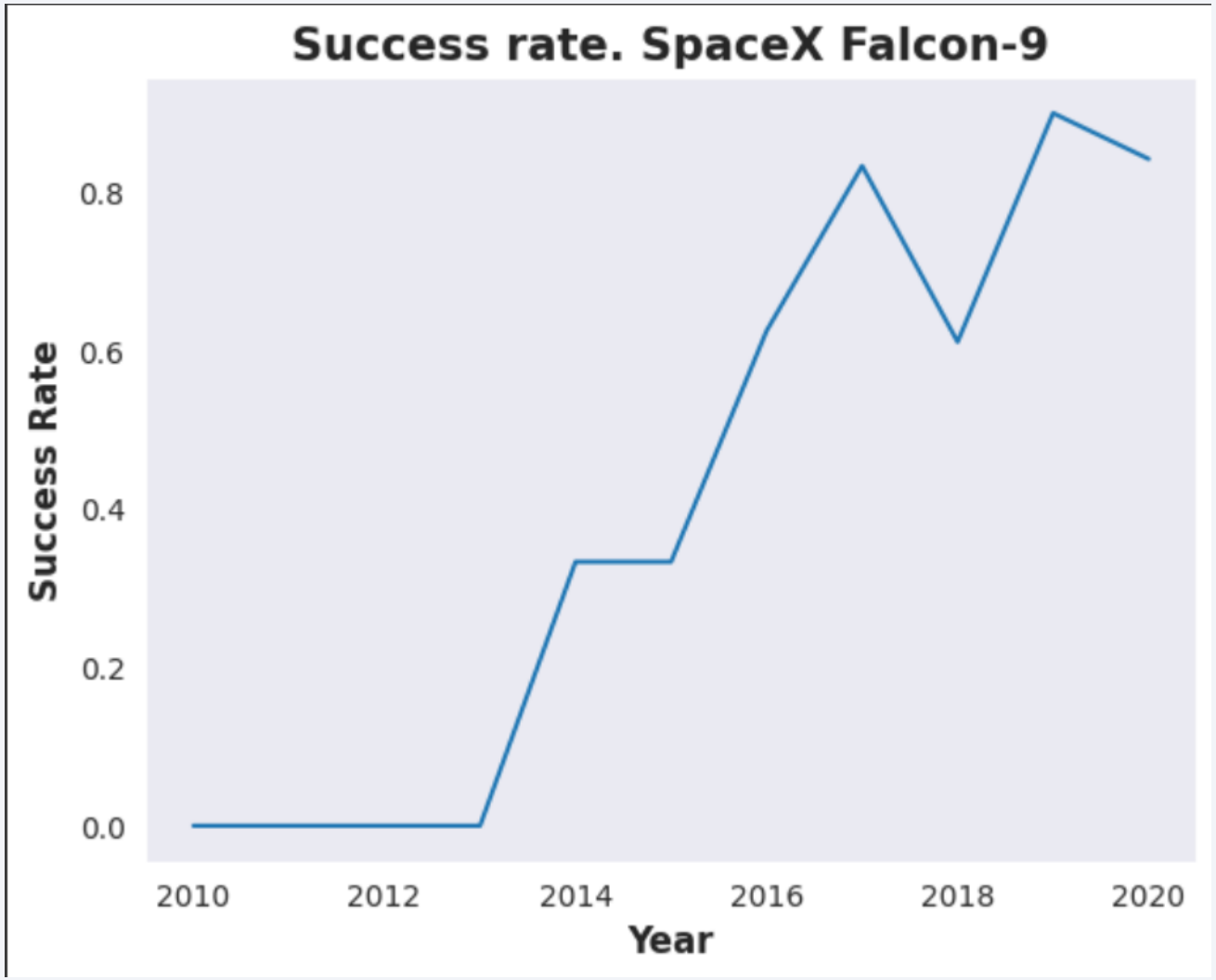
Payload vs. Orbit Type



For ISS, there is a strong relation in the number of launches and payload mass between 2000-4000 kgs

Similarly, for GTO the payload mass varies between 4000-7000 kgs unlike other orbits where payload mass distribution does not show a clear pattern

Launch Success Yearly Trend



The success rates have increased over the years considerably and stabilized since 2019 probably because of the lessons learnt over the years

All Launch Site Names

```
[ ]  %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;

 * sqlite:///my_data1.db
Done.
  Launch_Site
 CCAFS LC-40
 VAFB SLC-4E
 KSC LC-39A
 CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Using the LIMIT clause to restrict the output with desired results

Total Payload Mass

```
[ ]  %sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" LIKE 'NASA (CRS)%';
```

```
 * sqlite:///my_data1.db
Done.
 Total_Payload_Mass
 48213
```

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

First_Successful_Landing_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Number_of_Outcomes FROM SPACEXTABLE GROUP BY "Landing_Outcome" ORDER BY "Landing_Outcome" ASC;
```

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

Landing_Outcome	Number_of_Outcomes
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_")  FROM SPACEXTABLE);
```

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

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

2015 Launch Records

```
%sql SELECT substr(Date, 6, 2) AS Month,"Landing_Outcome","Booster_Version","Launch_Site" FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (dro
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
[21] %sql WITH RankedOutcomes AS (SELECT "Landing_Outcome",COUNT(*) AS Outcome_Count,RANK() OVER (ORDER BY COUNT(*) DESC) AS Outcome_Rank FROM SPACEXTABLE WHERE substr(Date, 1, 4) ||
```

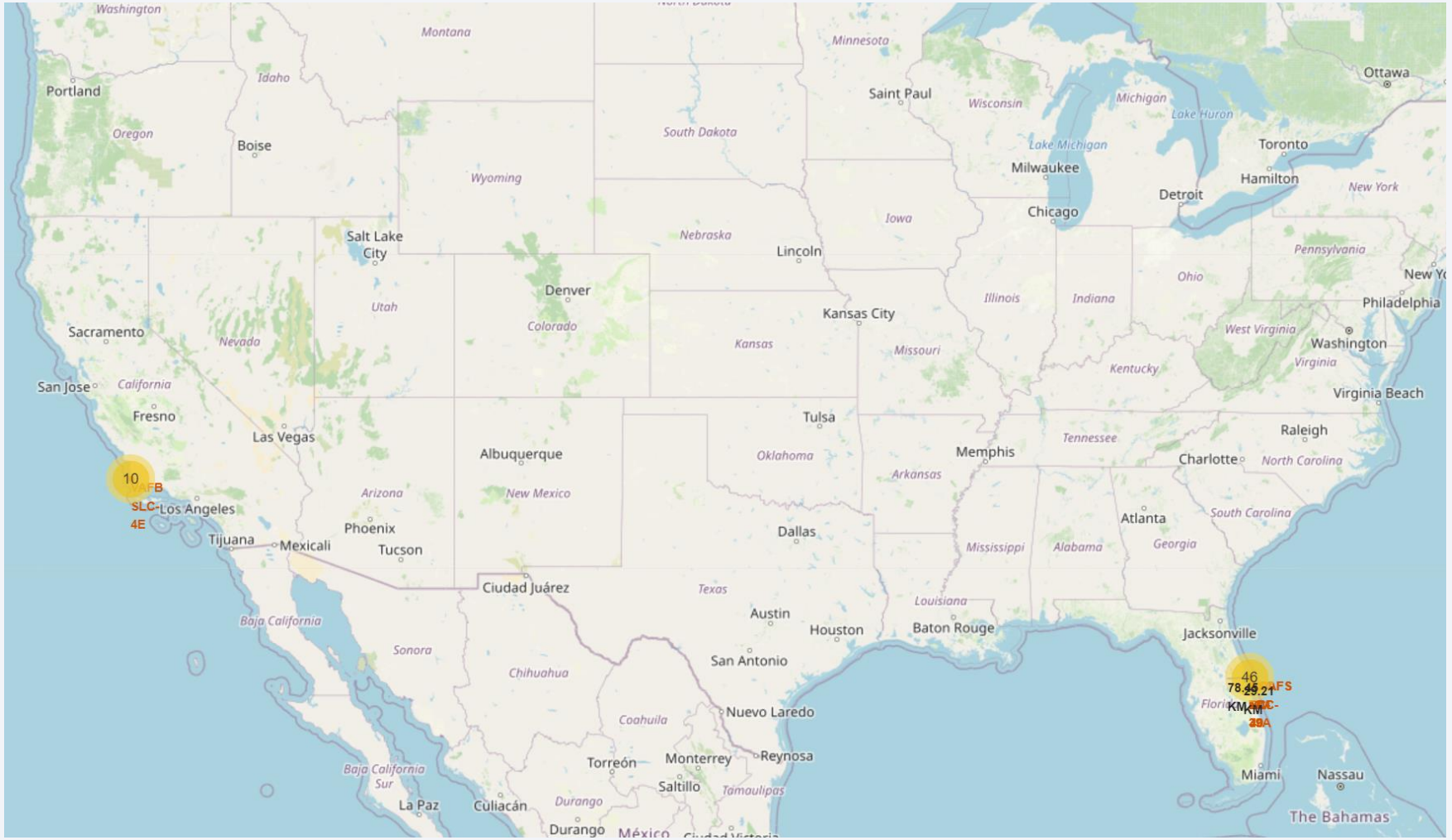
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Outcome_Count	Outcome_Rank
No attempt	10	1
Success (drone ship)	5	2
Failure (drone ship)	5	2
Success (ground pad)	3	4
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Precluded (drone ship)	1	8

Section 3

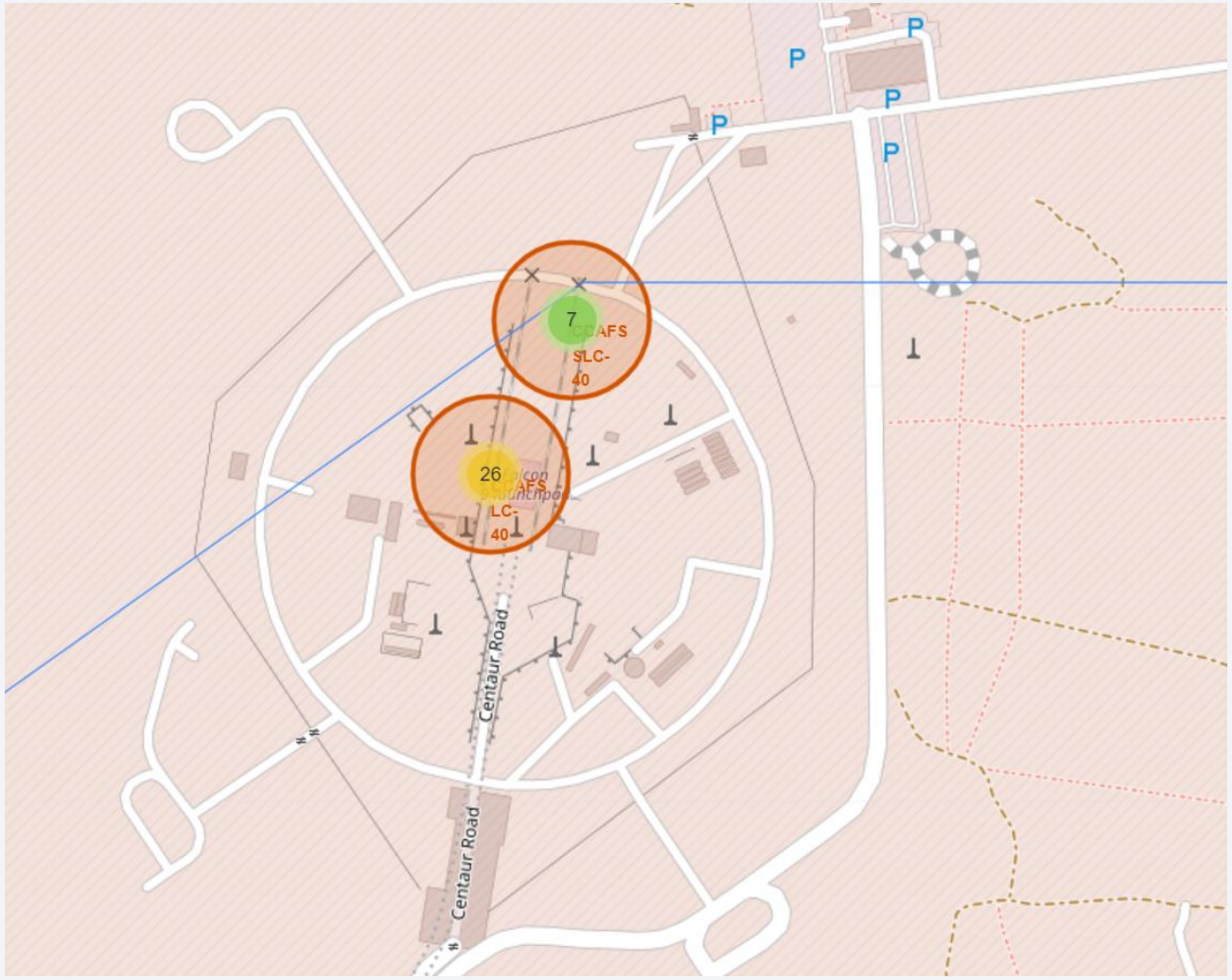
Launch Sites Proximities Analysis

All Launch Sites on a Map



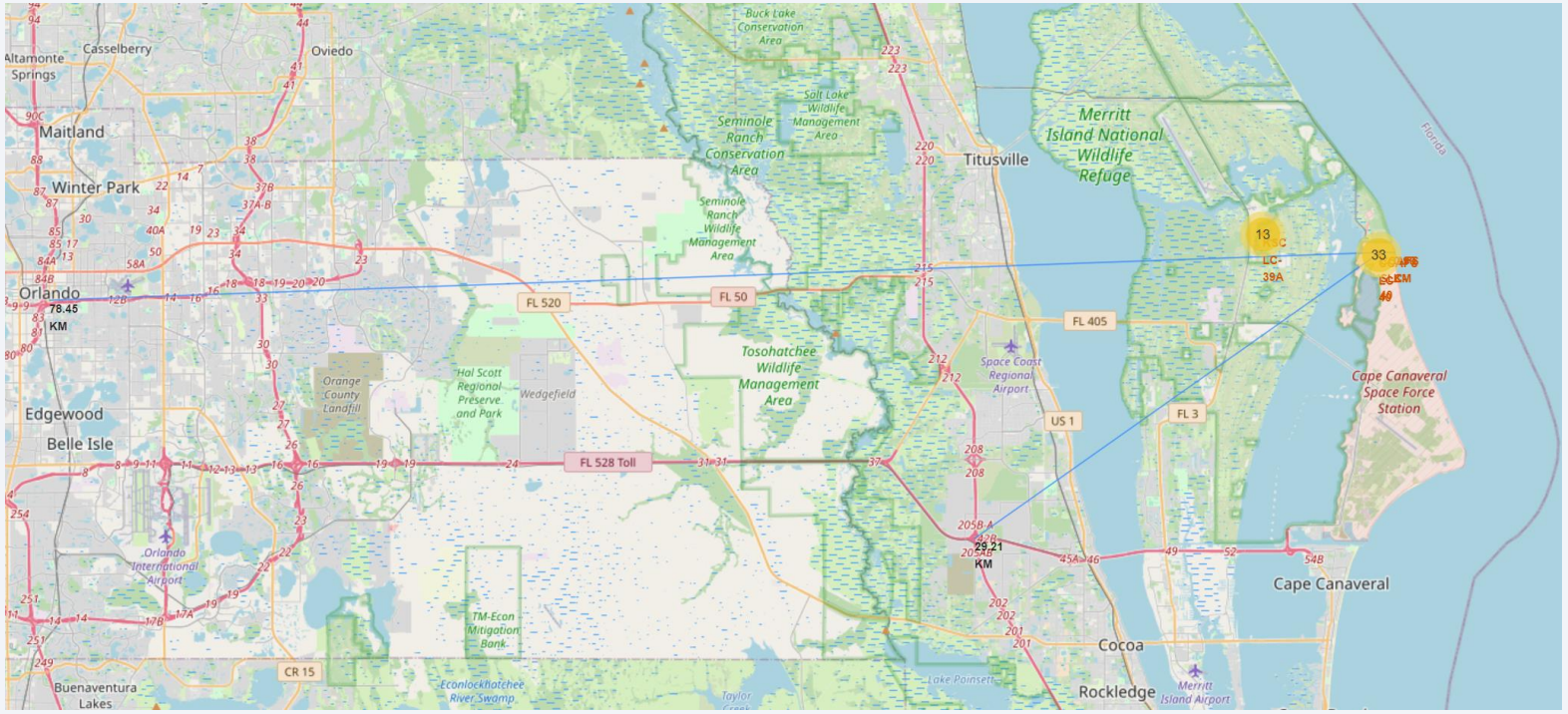
Two launch sites in Florida and California

Markers on Launch Sites

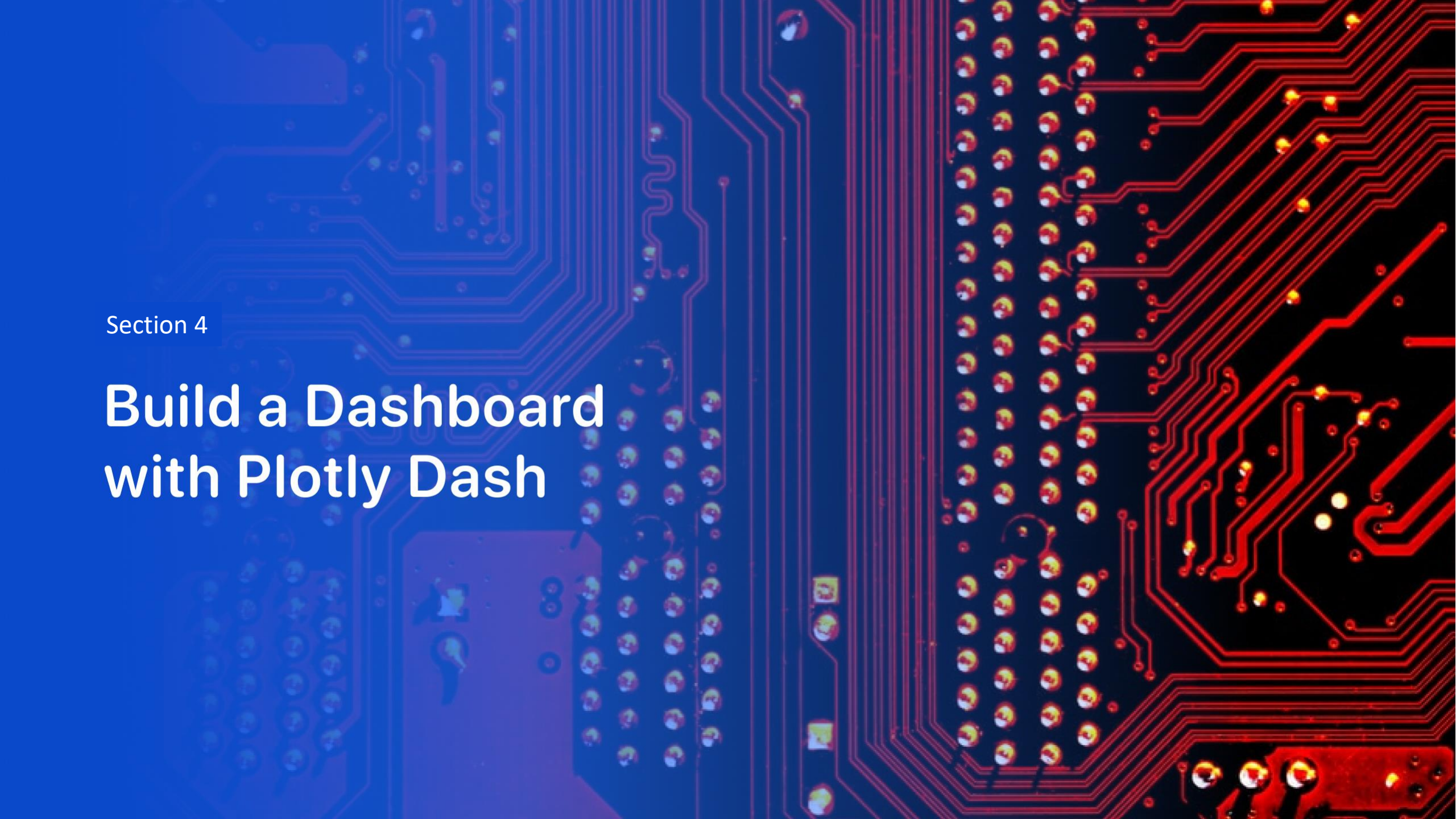


The green marker with count shows number of successful launches from the launch pad location on the map

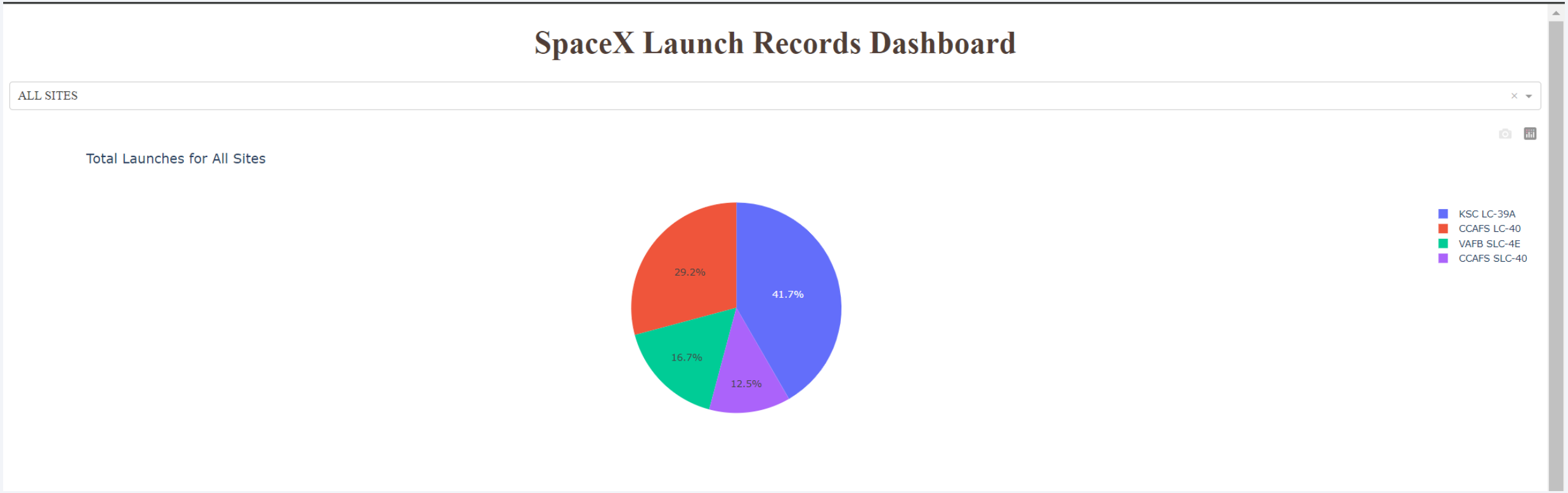
Lauch site distance from landmarks



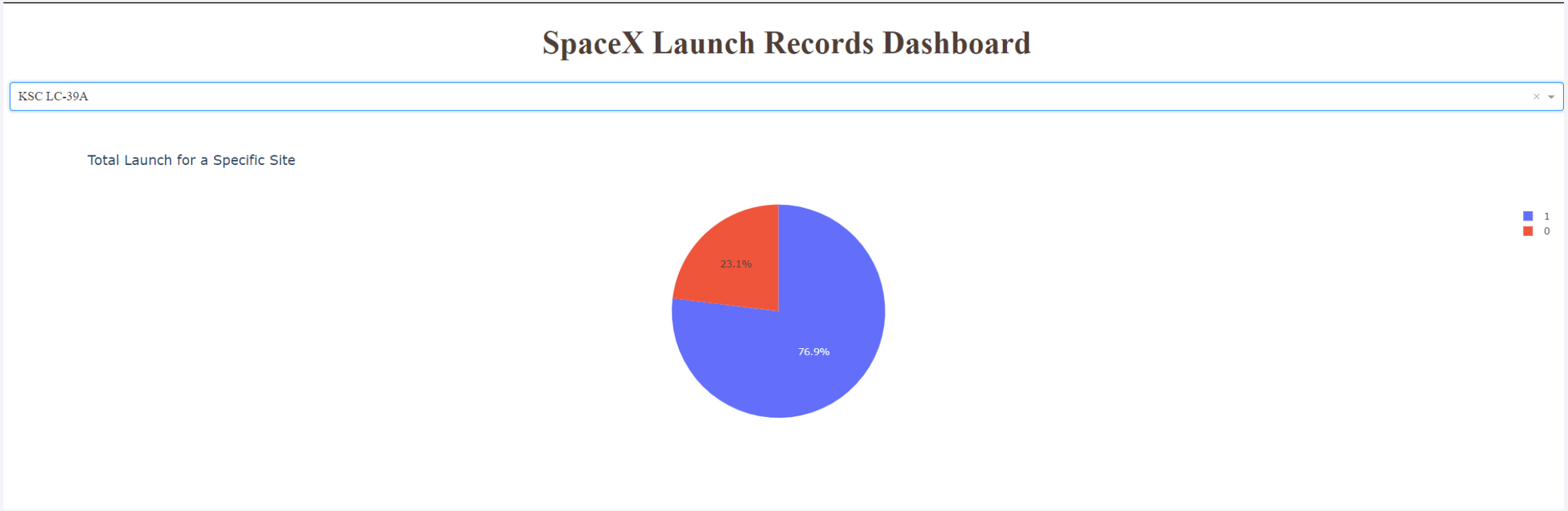

```
distance_highway = 0.5834695366934144  km
distance_railroad = 1.2845344718142522  km
distance_city = 51.434169995172326  km
```


Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches by Each Site



Lauch Site with highest success ratio



KSC LC-39A has 76.9% success ration, highest of all the sites

Different payloads vs outcomes using scatter plots

Figure 1: Low weight payload(0-4000kgs), ,more in number

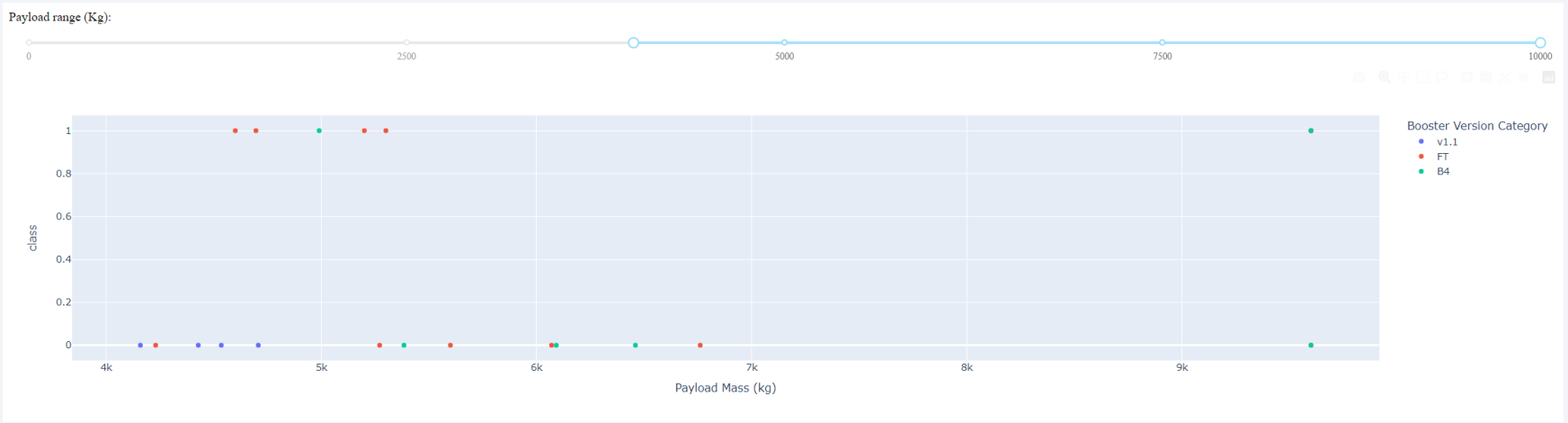
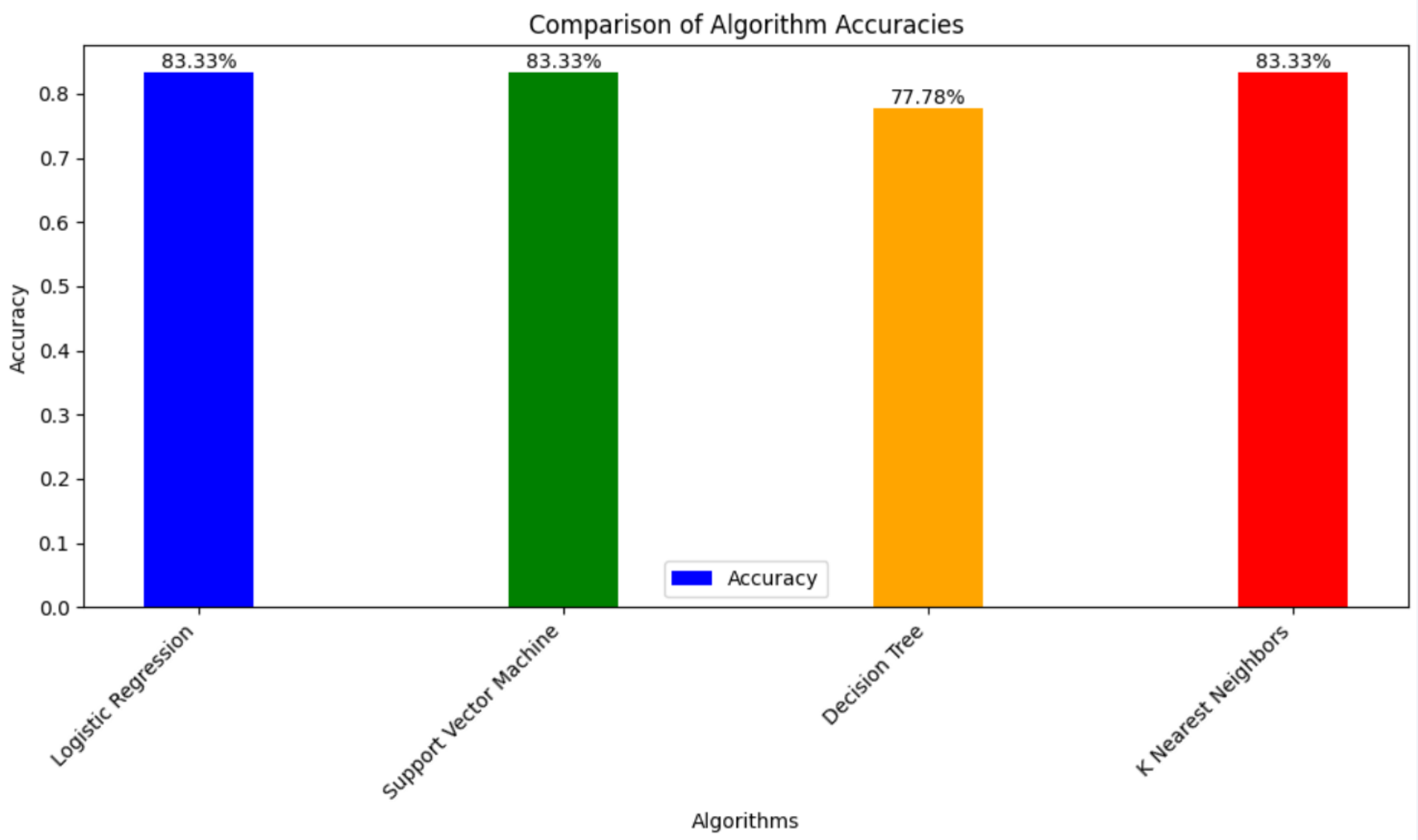


Figure 2: High weight payload(>4000kgs), less in number

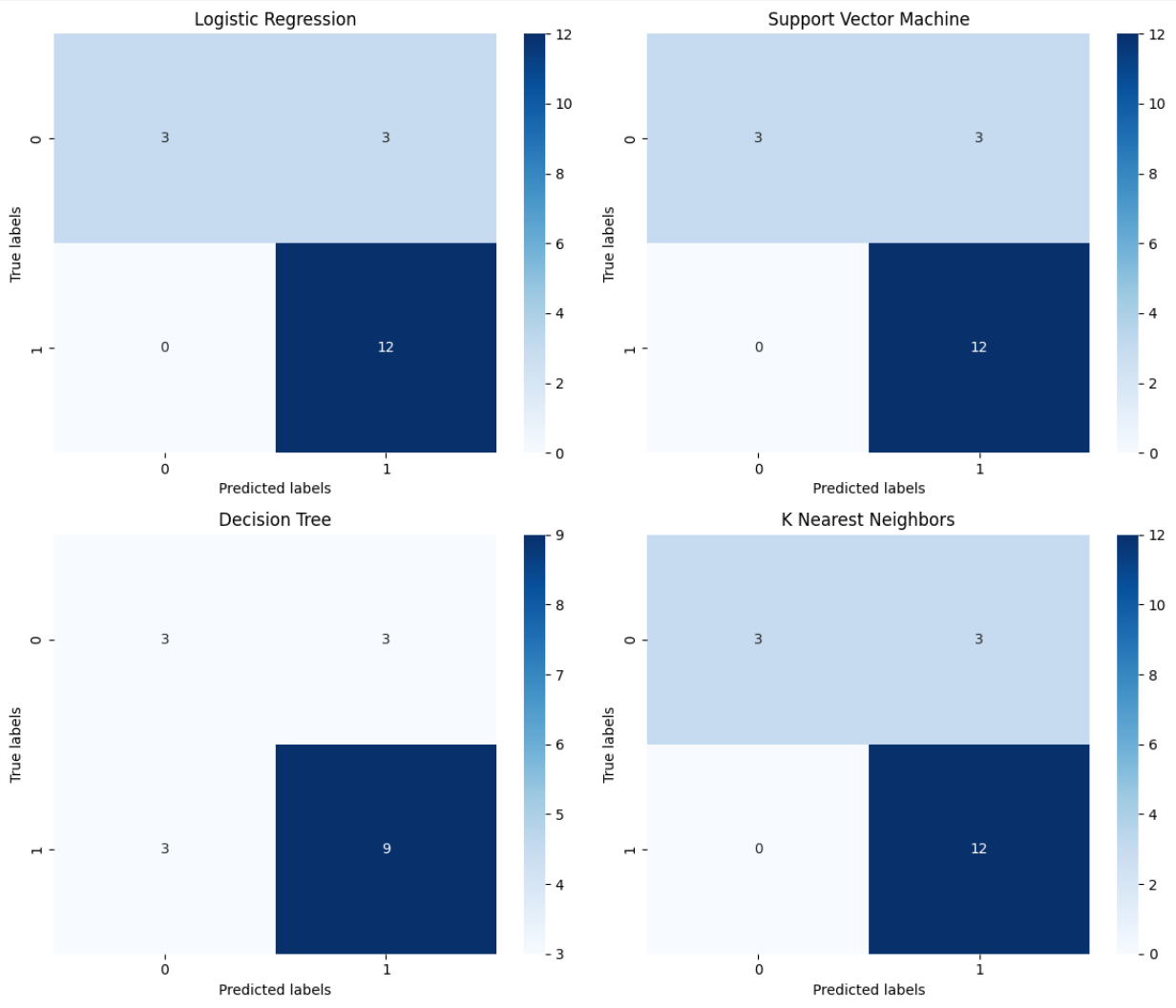
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



The confusion matrix for the KNN, LR and SVM show that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing

Conclusions

- SVM, KNN and LR are the best models in terms of prediction accuracy
- Low weight payloads perform better over heavier payloads
- With time and betterment in space technology, the success rates have improved
- KS LC 39A is the most desirable launch site for successful launches
- Orbits ES-L1, GEO, HEO, SSO, VLEO have the most success rates over the years

Thank you!