IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jane Mburu
October 21, 2023

Outline

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

Executive Summary

- Summary of Methodologies:
  - Data Collection
  - Data wrangling and formatting
  - Exploratory Data Analysis (EDA) with Visualization
  - EDA with SQL
  - Building an Interactive map with Folium
  - Building a Dashboard with Plotly.
  - Predictive Analysis
- Summary of all Results
  - EDA Results
  - Interactive Analytics
  - Predictive Analysis (Classification)

Introduction

- The purpose of this project is to predict if Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. As such, if we can determine if the first stage will land, then we can determine the cost of a launch. This information is critical as it can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- The overarching question that this project is trying to answer is, for a given set of features about Falcon 9 launch including its payload mass, orbit type, launch site etc will the first stage of the rocket land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest Api
  - Web Scraping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and null values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, DT, SVM were built and evaluated for the best classifier.

Data Collection

- The following data sets were collected:
  - SpaceX API
  - WebScraping
- The link to SpaceX API:

Applied-Data-Science-Capstone-Project/jupyter-labs-spacex-data-collection-api.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

- The link to WebScraping:

Applied-Data-Science-Capstone-Project/jupyter-labs-webscraping.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- Link to the notebook: (Applied-Data-Science-Capstone-Project/jupyter-labs-spacex-data-collection-api.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

1.Get request for rocket launch data using API

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

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

2. Use Json normalize method to convert ison results to dataframe

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

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

3. Perform data cleaning and filling in missing values

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

data_falcon9.isnull().sum()
```

Data Collection - Scraping

- We applied webscraping to webscrap Falcon 9 launch records with Beautiful Soup.
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is:[Applied-Data-Science-Capstone-Project/jupyter-labs-webscraping.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)]

1.Apply HTTP GET method to request the Falcon9 Launch HTML 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
html_page = requests.get(static_url)
```

2. Create Beautiful Soup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_page.text, 'html.parser')
```

```
# Use soup.title attribute
soup.title
```

3. Extract column names from the HTML table header

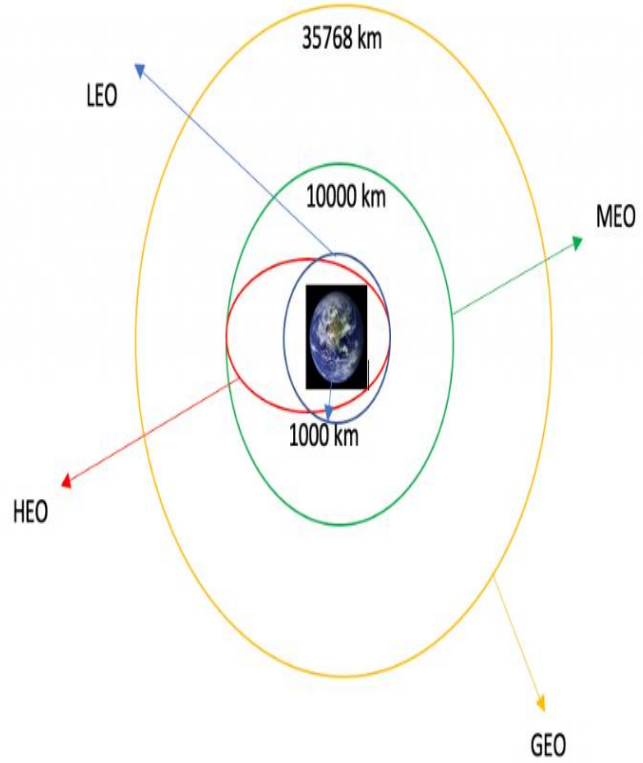
```
column_names = []

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

4. Create a dataframe by parsing the launch HTML tables.

5. Export data to CSV

Data Wrangling

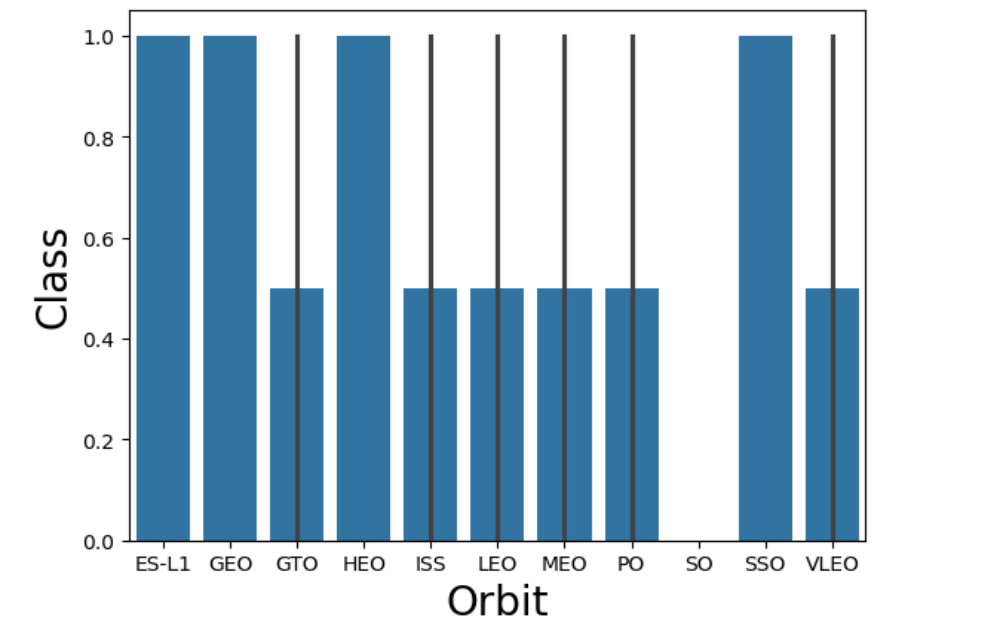


- We performed exploratory data analysis and determined the training lables.
- We then calculated the number of launches at each site, and the number of occurrence of each orbits.
- After, we created the landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is:[Applied-Data-Science-Capstone-Project/labs-jupyter-spacex-Data wrangling.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)]

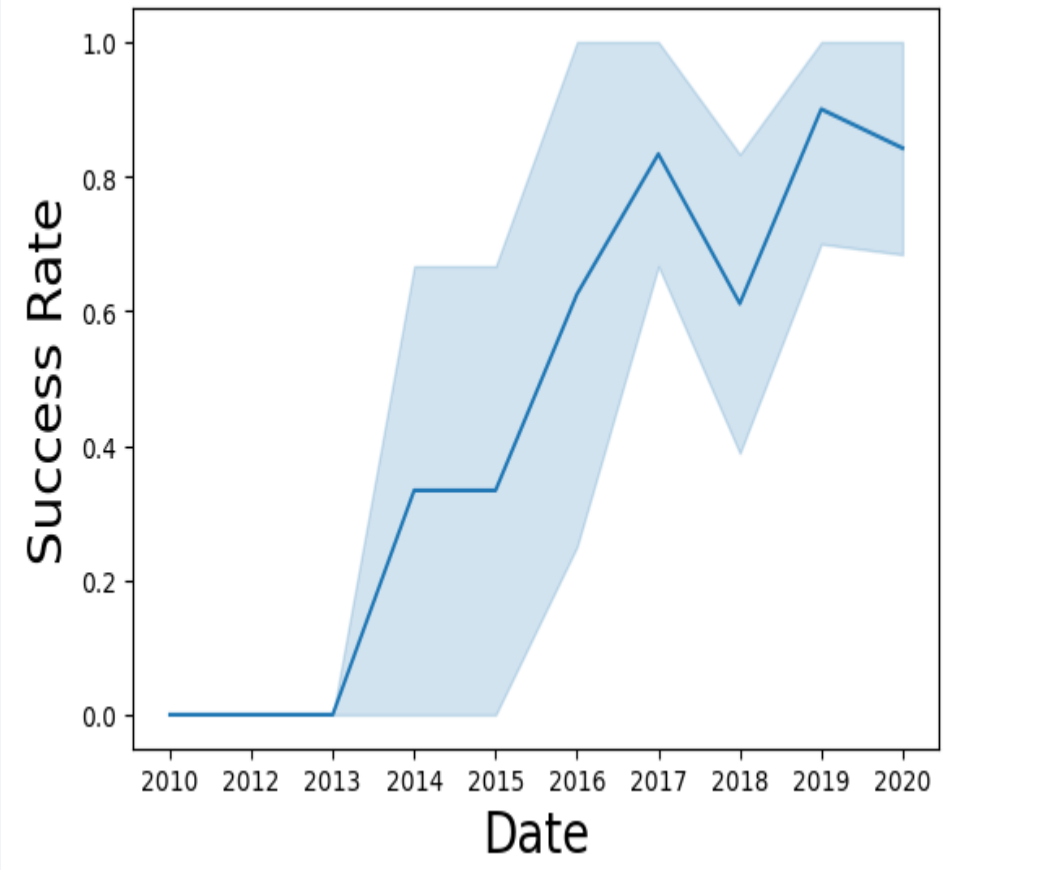
EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

Plot of Success Rate by Class



Plot of Launch success yearly trend



EDA with SQL

- We executed the following SQL queries to answer the project questions:
  - Names of unique launch sites in the space mission
  - Launch sites beginning with the string 'CCA'
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - Total number of successful and failure mission outcomes
  - Failed landing outcomes in drone ship, their booster version and launch sites n
- Link to the notebook is: Applied-Data-Science-Capstone-Project/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities to answer some project questions.
- Link to the notebook: Applied-Data-Science-Capstone-Project/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

Build a Dashboard with Plotly Dash

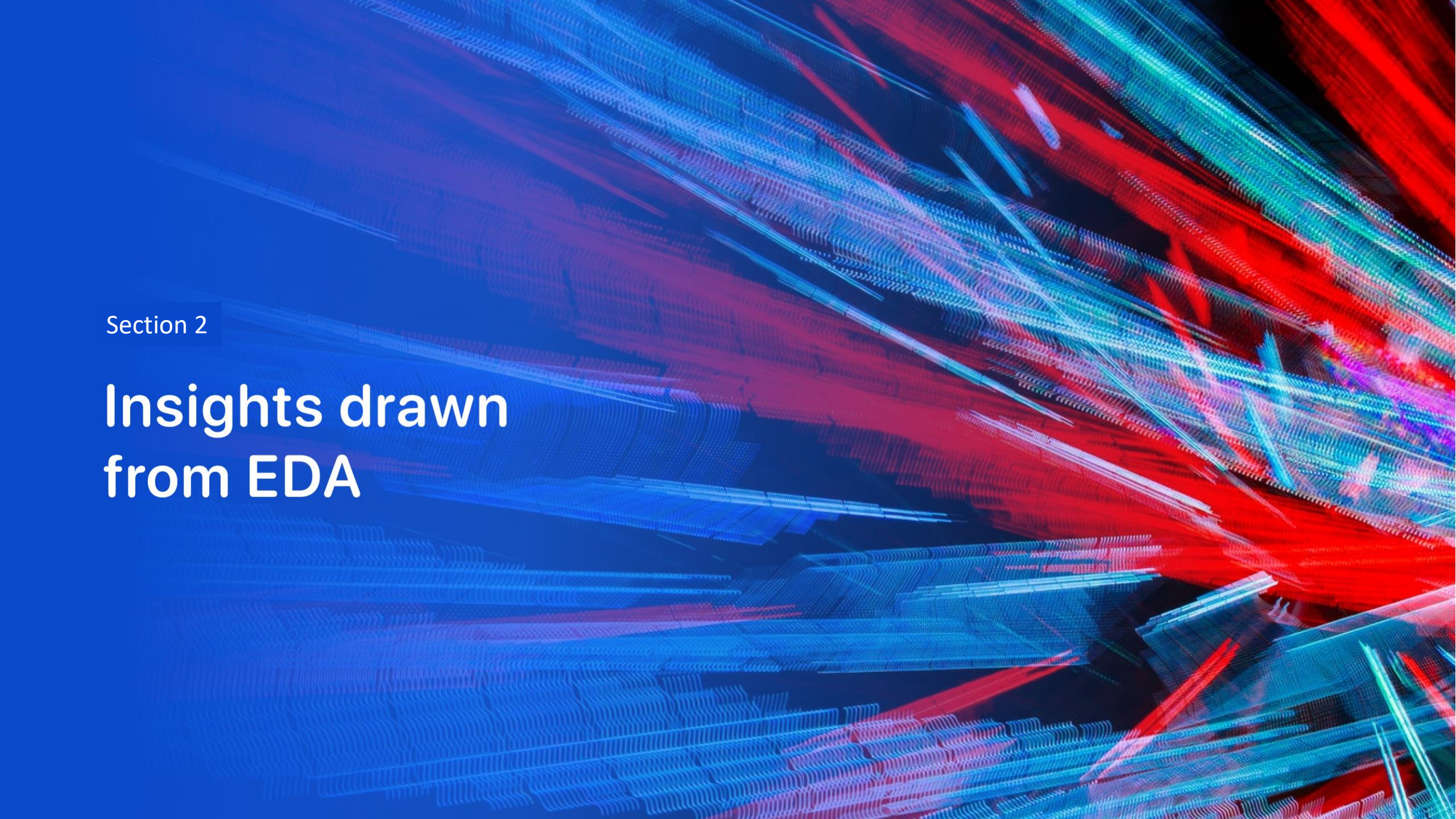
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version
- Link to the notebook: Applied-Data-Science-Capstone-Project/spacex_dash_app (1).py at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We created different machine learning models and tuned different hyperparameters using GridSearchCV.
- We calculated accuracy and used it as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- Link to the notebook: Applied-Data-Science-Capstone-Project/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · JMburu23/Applied-Data-Science-Capstone-Project (github.com)

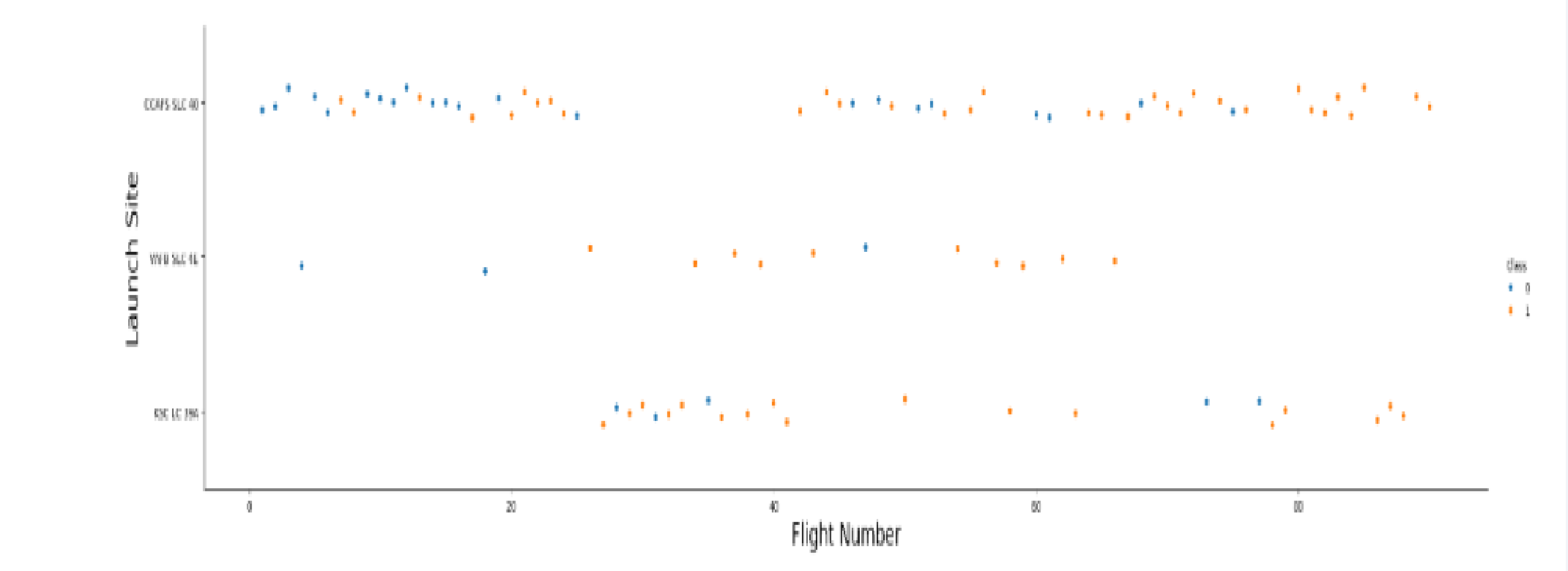
Results

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


Section 2
Insights drawn from EDA

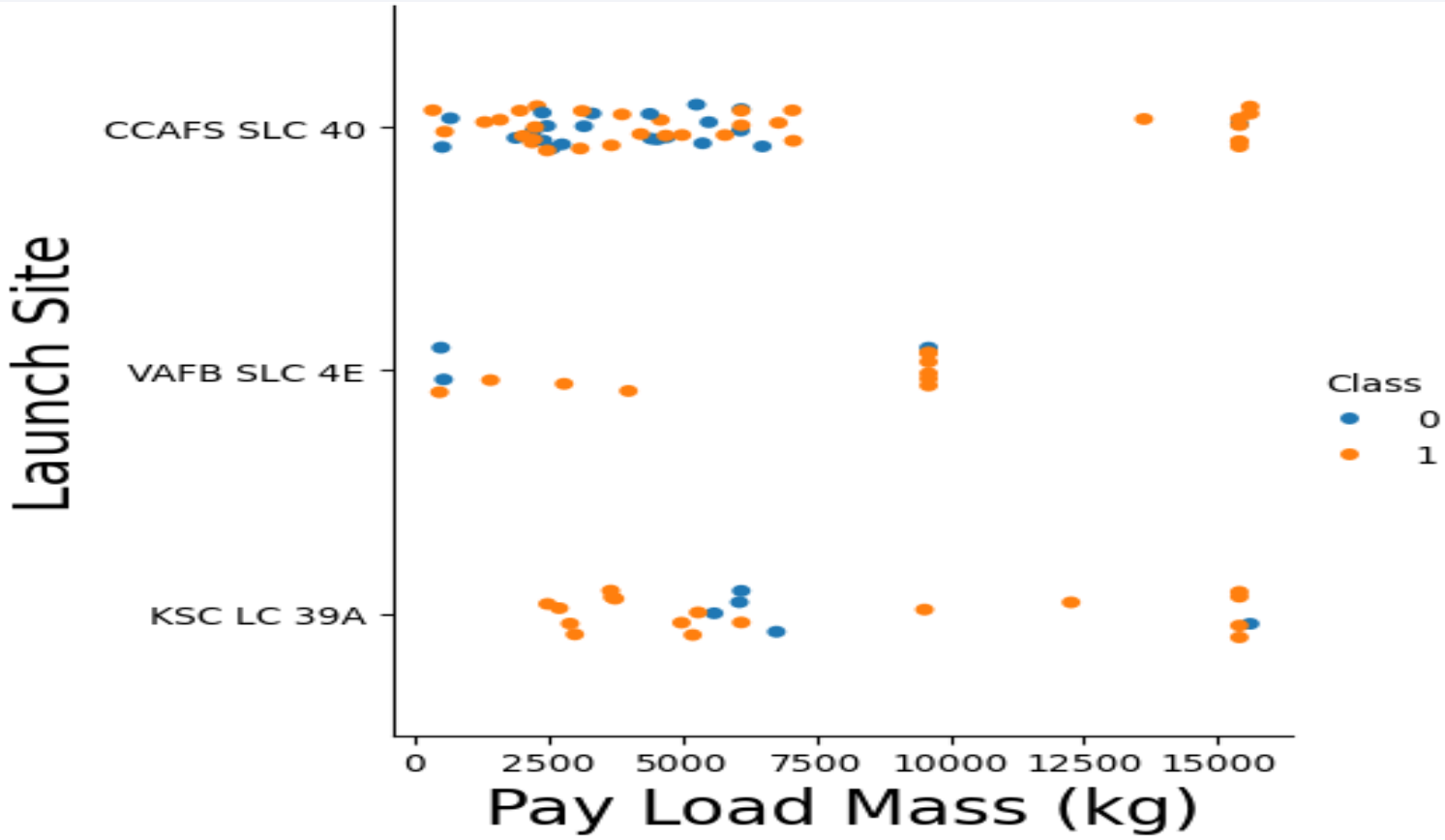
Flight Number vs. Launch Site

- From the plot, we found that the success rate at a launch site increases with the increase in the flight number.



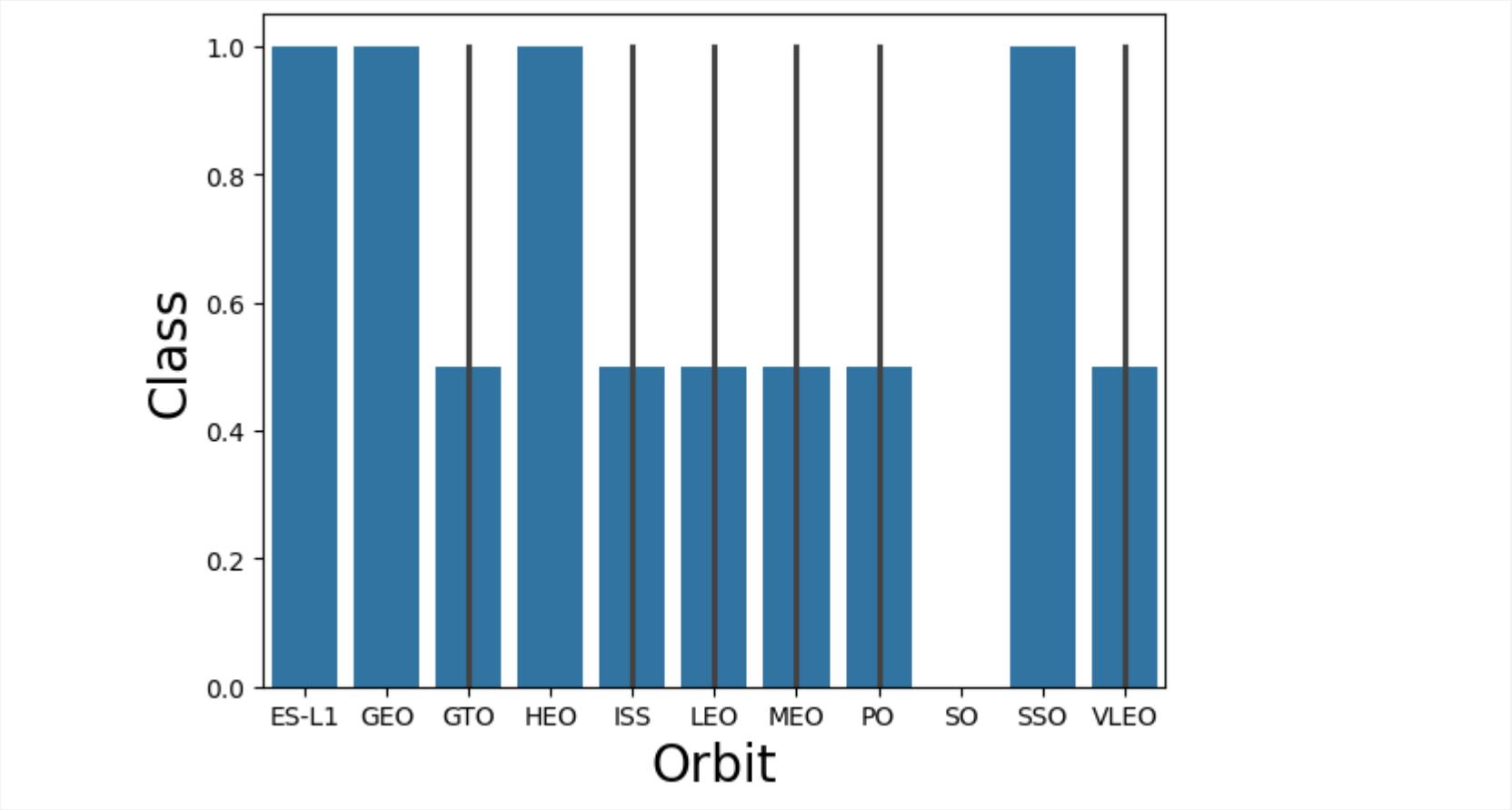
Payload vs. Launch Site

From the scatter plot, it is evident that increase in Pay Load Mass (kg) increases the launch site success rate of CCAFS SLC 40



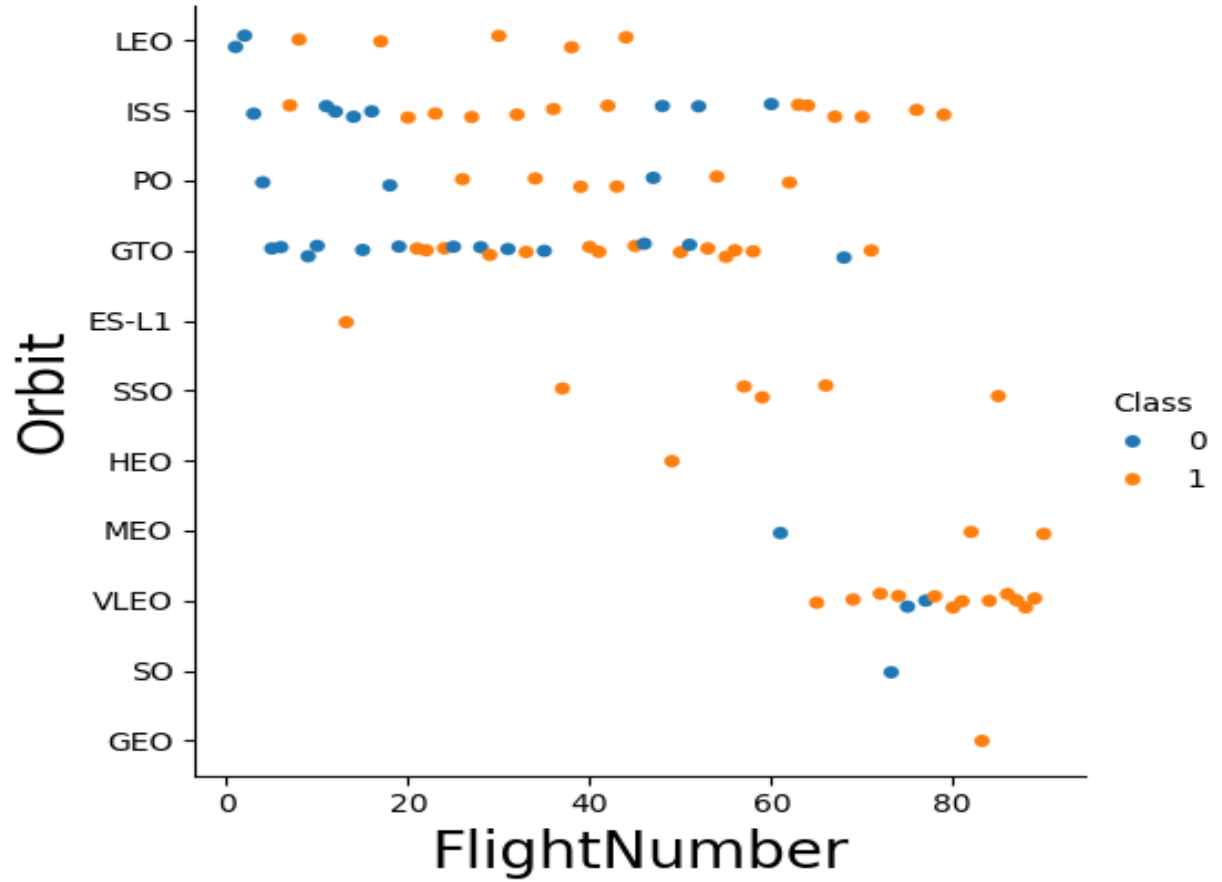
Success Rate vs. Orbit Type

From the bar chart, it is evident that orbit type ES-L1, GEO, HEO, SSO had the most success rate



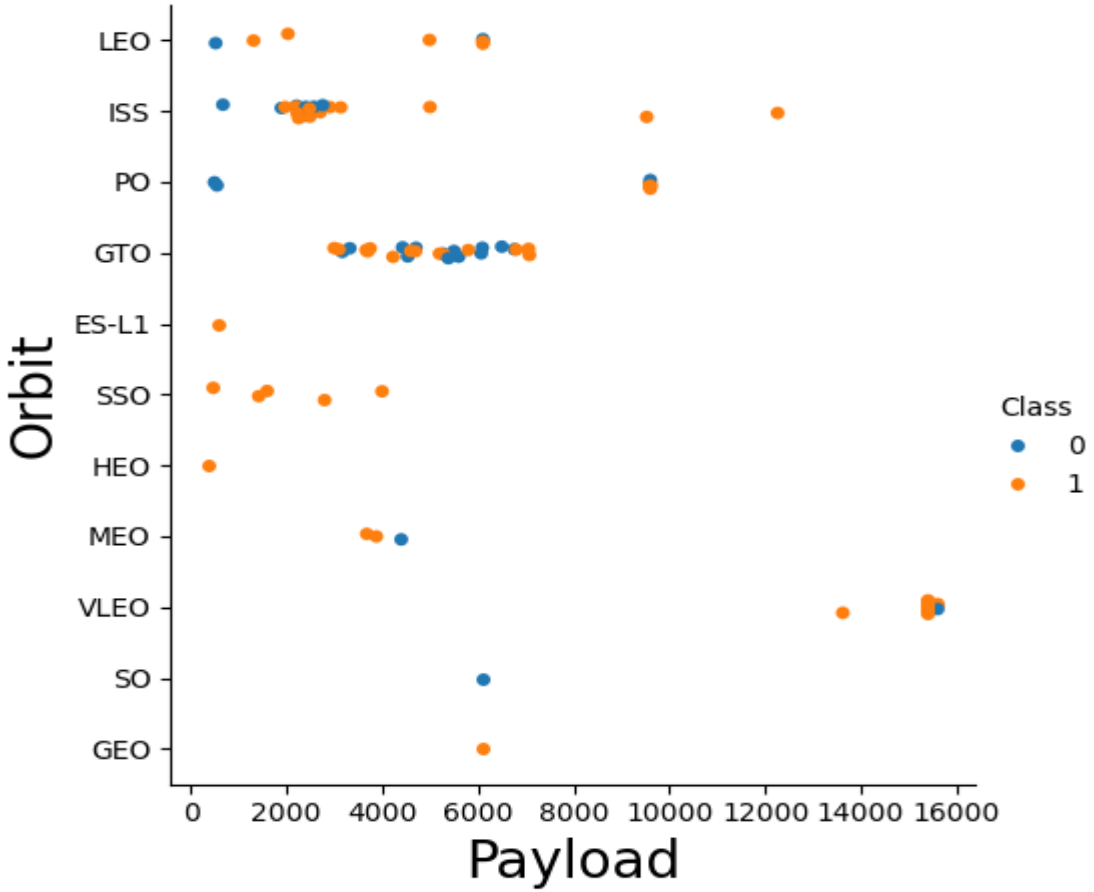
Flight Number vs. Orbit Type

In this scatter plot, we observe that the success rate of LEO orbit is related to the number of flights whereas there appears to be no relationship between flight number and GTO orbit.



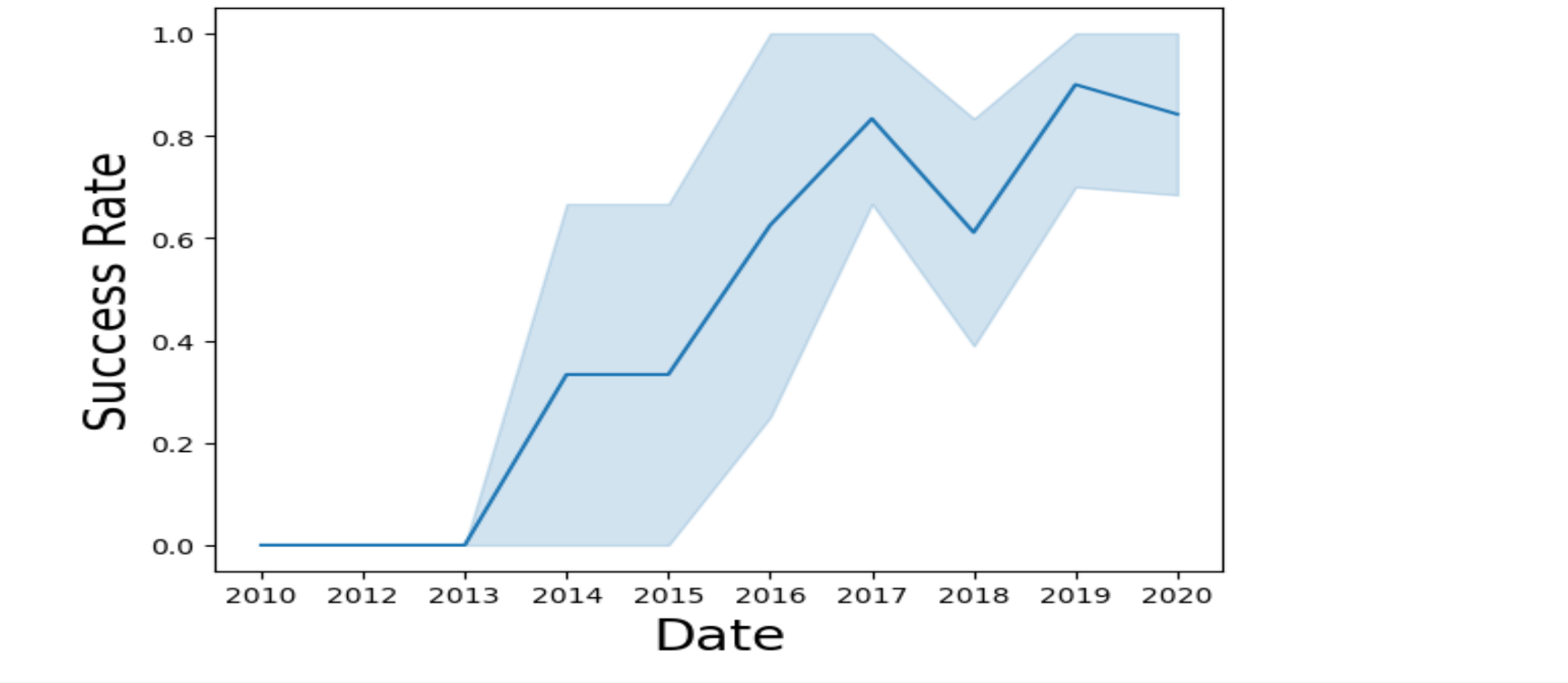
Payload vs. Orbit Type

From this scatter plot, we can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits, however, for GTO we cannot distinguish this.



Launch Success Yearly Trend

From the plot, we can observe that success rate since 2013 kept on increasing till 2020



All Launch Site Names

We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

In [43]:
```
%sql select distinct launch_site from SPACEXTABLE;
```

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

Out[43]:

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

Launch Site Names Begin with 'CCA'

We used the below query to display 5 records where launch sites begin with the string ‘CCA’

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

In [44]:
```
%sql select * from SPACEXTABLE where launch_site like 'CCA%' limit 5;
```

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

Out[44]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	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

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

In [45]:
```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTABLE where customer = 'NASA (CRS)';
```

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

Out[45]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- We used the query below to calculate the average payload mass carried by booster version F9 v1.1 as 2534

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

In [47]:
```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTABLE where booster_version like '%F9 v1.1%';
```

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

Out[47]:

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

- We used the code below to calculate the dates of the first successful landing outcome on ground pad as December 22, 2015

```
In [48]: %sql select min(date) as first_successful_landing from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)';
 * sqlite:///my_data1.db
Done.
Out[48]: first_successful_landing
---
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [49]:
```
%sql select booster_version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and payload_mass__kg_ between 4(
```

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

Out[49]:

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

Total Number of Successful and Failure Mission Outcomes

We used the query below to calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

In [50]:
```
%sql select mission_outcome, count(*) as total_number from SPACEXTABLE group by mission_outcome;
```

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

Out[50]:

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We used the query below to determine the names of the booster which have carried the maximum payload mass

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

```
In [51]: %sql select booster_version from SPACEXTABLE where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

Out[51]:

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

We used the query below to determine the failed landing outcomes in drone ship, their booster versions, and launch site names in year 2015

In [64]:
```
%sql SELECT "Landing_Outcome", substr(Date,0,5) as "year", SUBSTR(Date,6,2) AS "Month", Booster_Version, Launch_site FROM SP
```

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

Out[64]:

Landing_Outcome	year	Month	Booster_Version	Launch_Site
Failure (drone ship)	2015	10	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	2015	04	F9 v1.1 B1015	CCAFS LC-40

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

We selected the landing outcomes and used WHERE clause to filter the landing outcomes BETWEEN2010-06-04 and 2017-03-20. We the applied the GROUP BY clause to group the landing outcomes and ORDER BY clause to order the grouped landing outcome in descending order.

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.

In [59]:
```
%%sql select Landing_Outcome, count(*) as count_outcomes from SPACEXTABLE
      where date between '2010-06-04' and '2017-03-20'
      group by Landing_Outcome
      order by count_outcomes desc;
```

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

Out[59]:

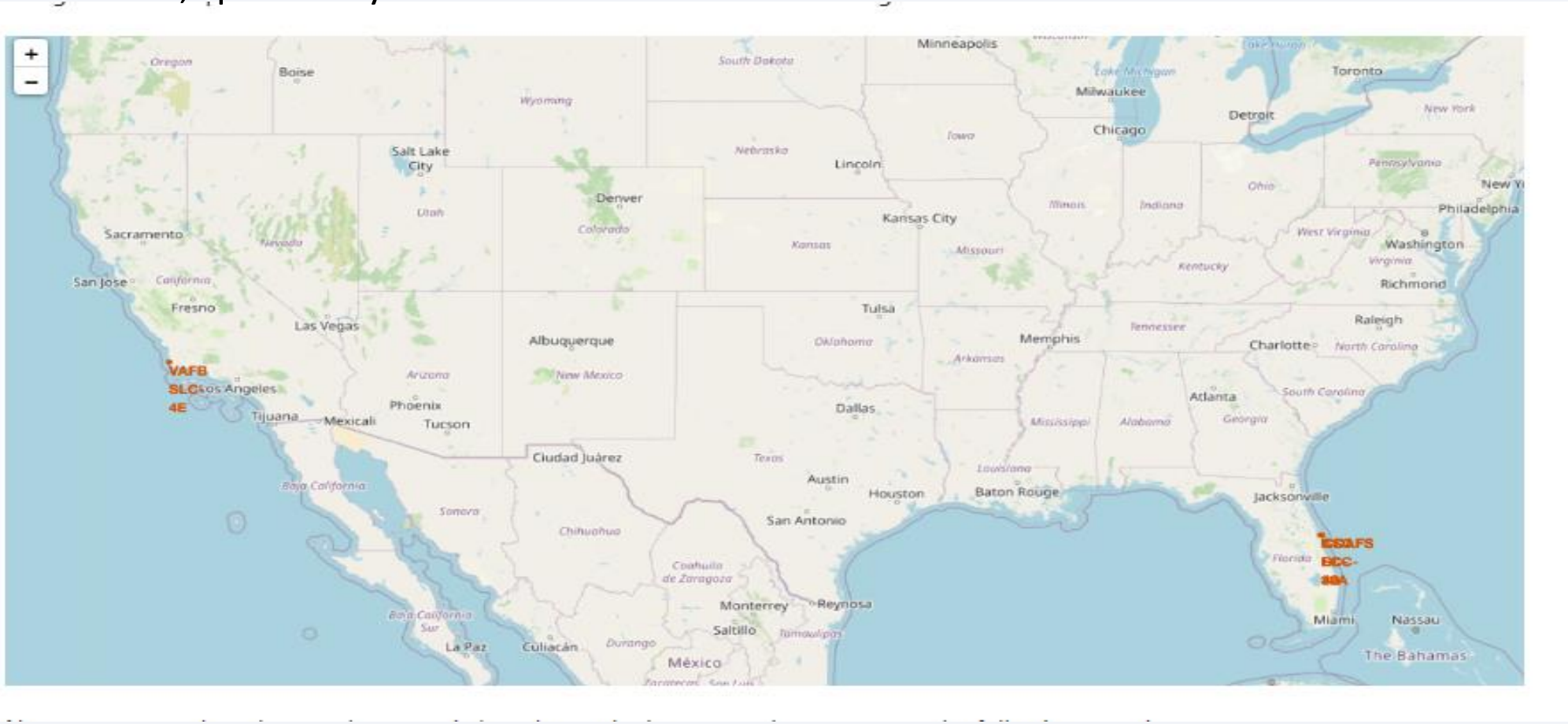
Landing_Outcome	count_outcomes
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

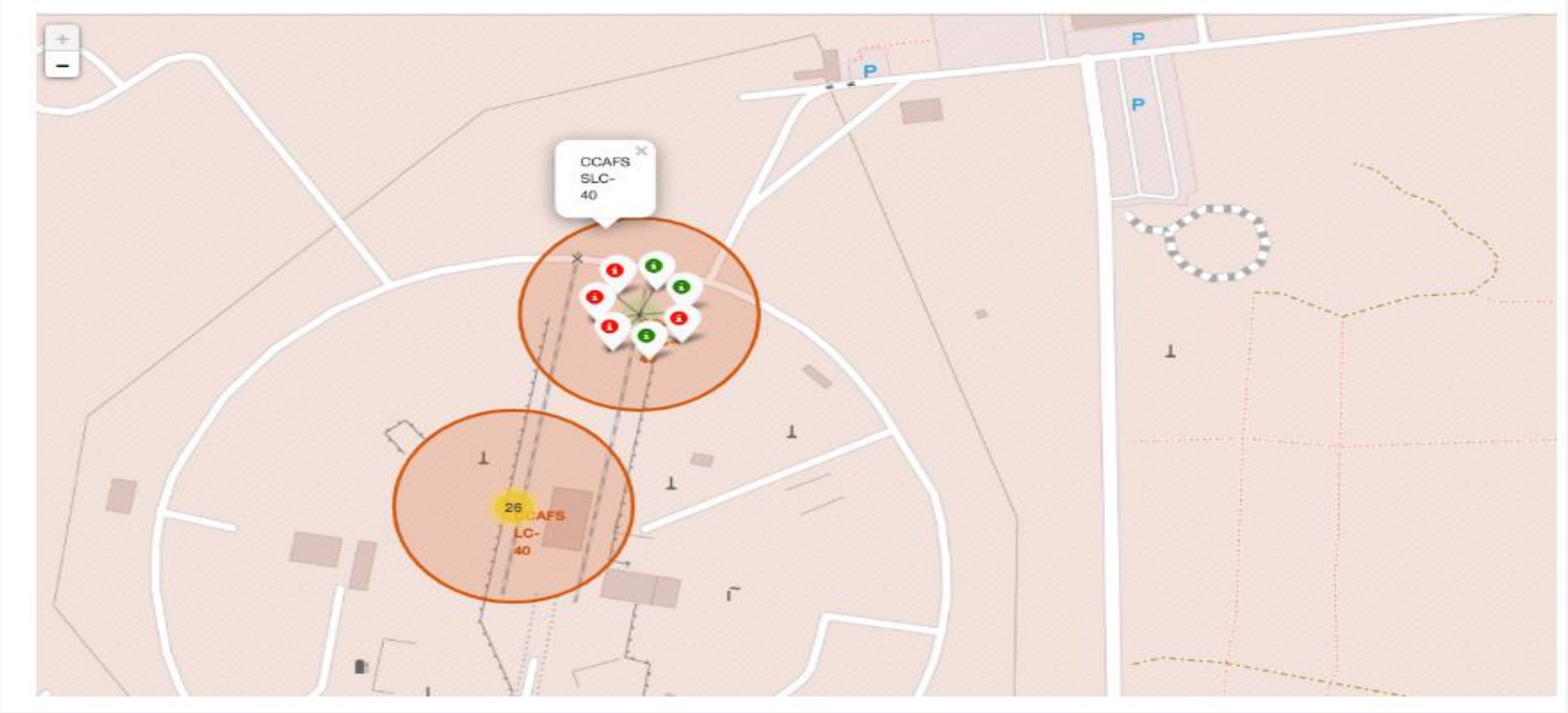
Launch Sites Global Map Markers

From the map below, it is evident that all launch sites are located within United States of American coasts, specifically California and Florida.



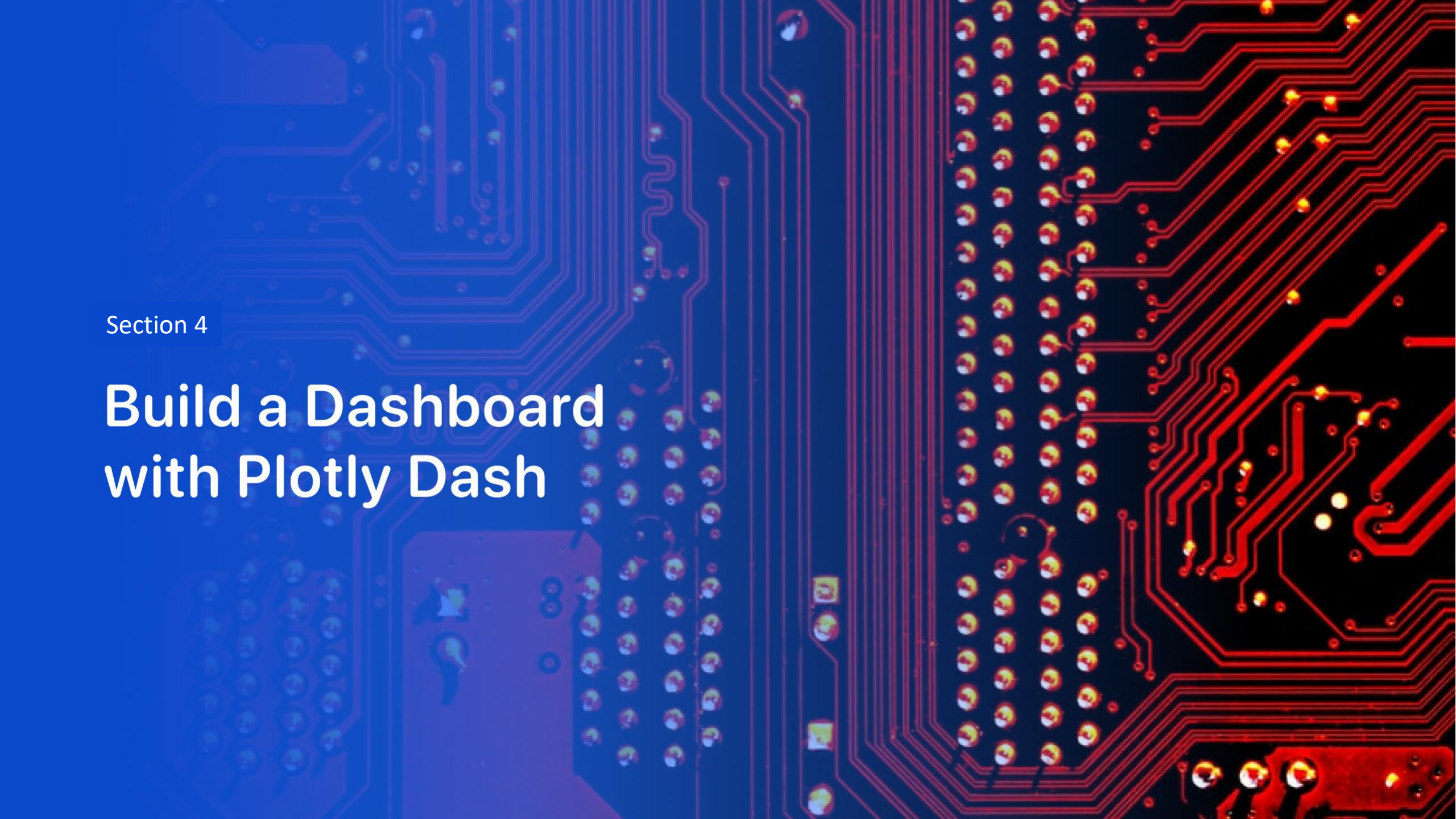
Markers Showing Launch sites with color labels

The green markers show successful launches while the red markers show failures

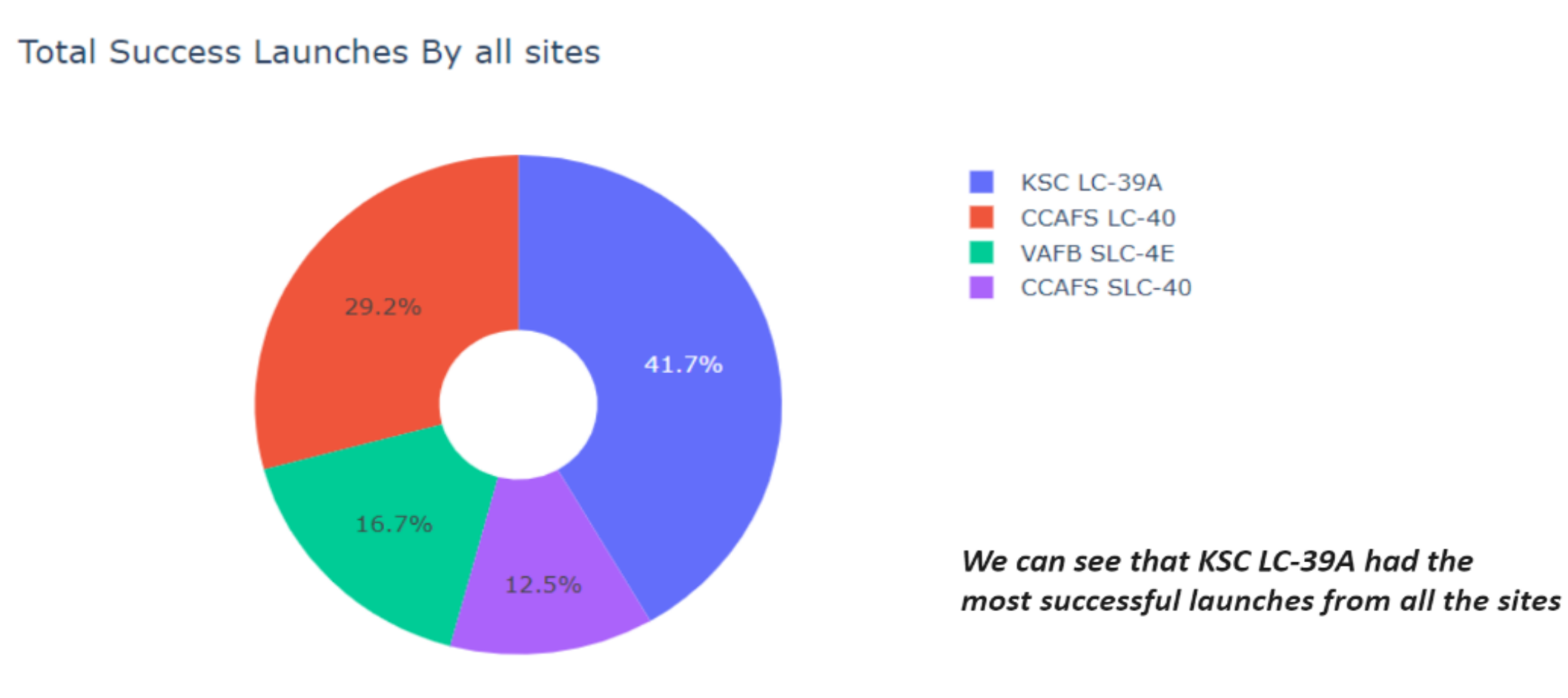


Launch Site Distance to Landmarks

From this map, it is evident that launch sites are in close proximity to coasts than highways and railways. This picture shows that the distance between launch site CCAFS SLC-40 to coastline is 0.90km.

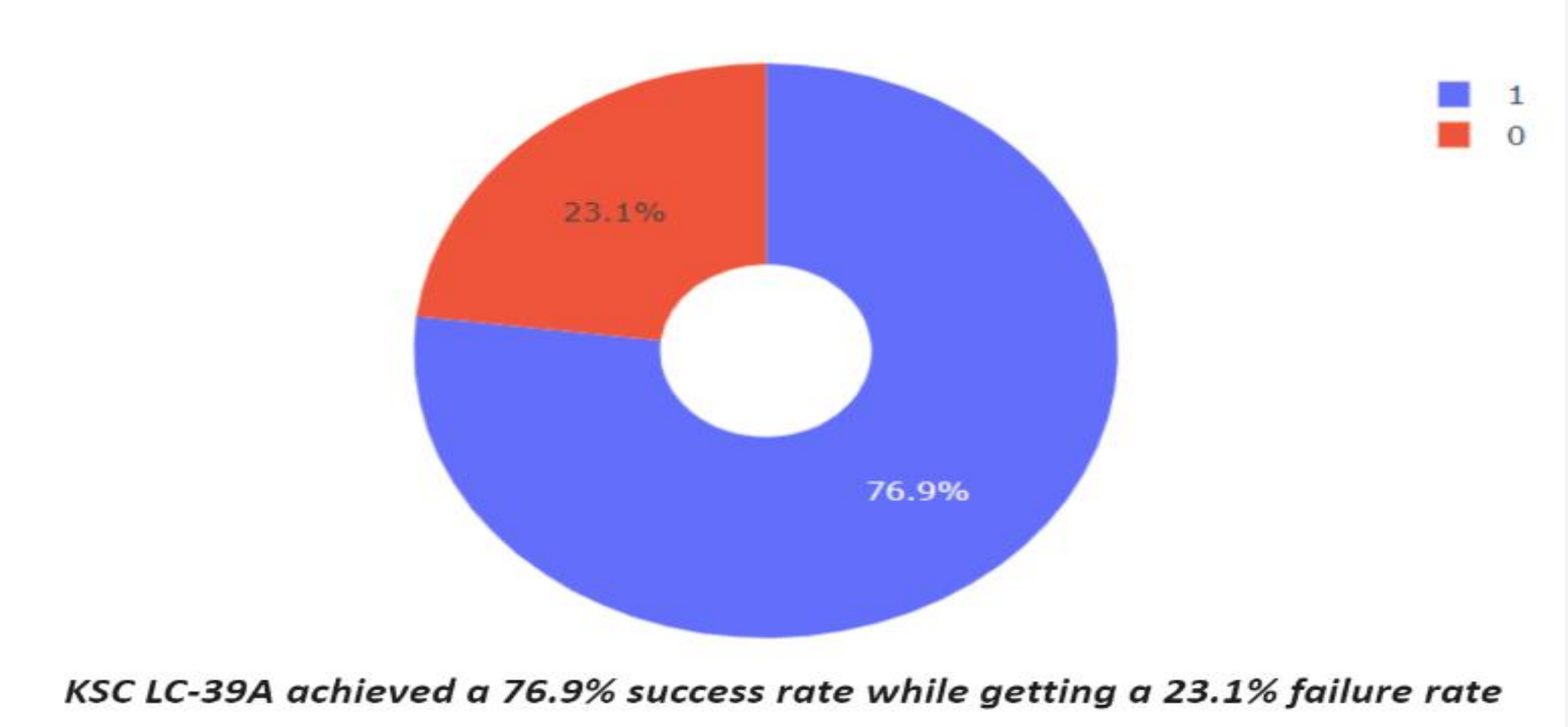

Section 4
Build a Dashboard with Plotly Dash

Pie Chart Showing Total Success Launches by Suites



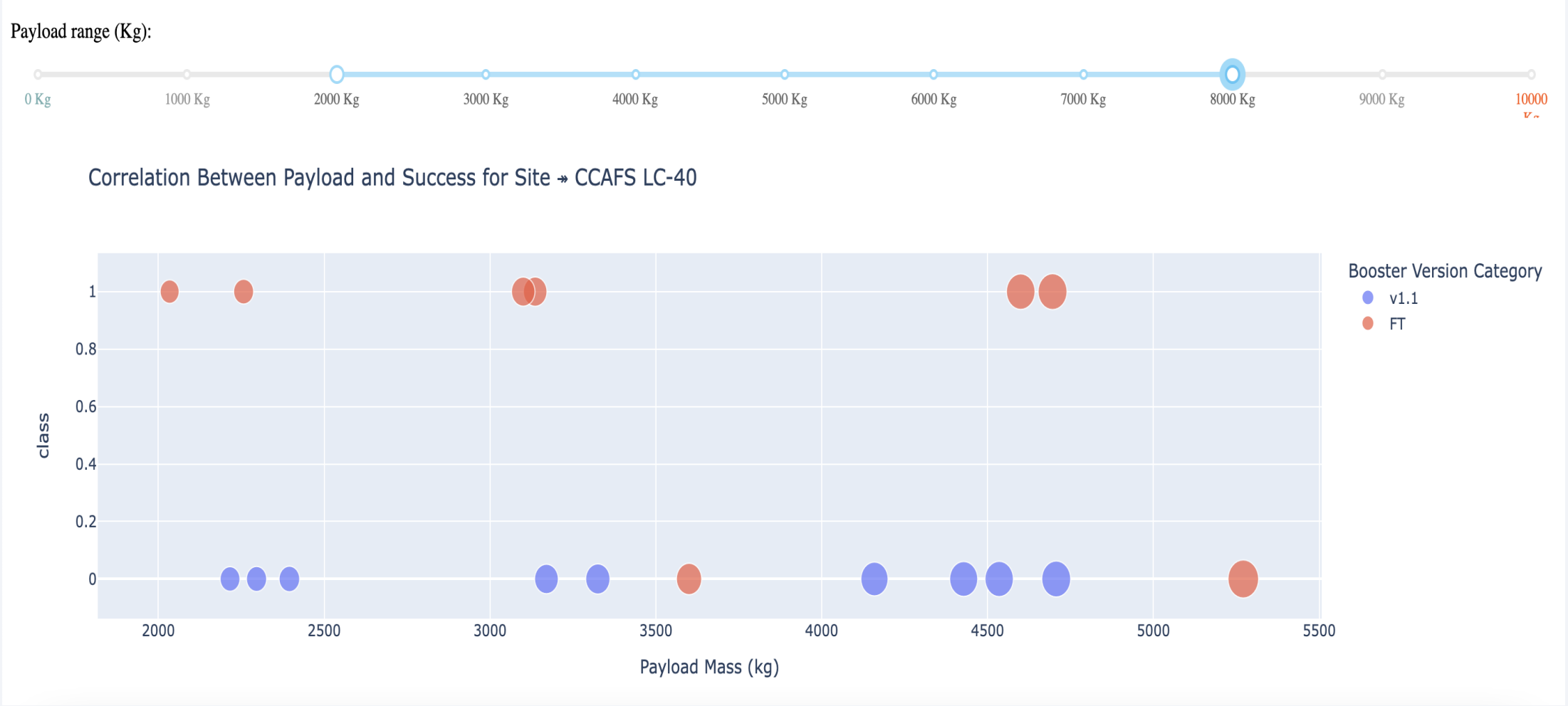
We can see that KSC LC-39A had the most successful launches from all the sites

Launch Site with the Highest Success Rate



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Correlation Between Pay Load and Success of All Sites



Section 5

Predictive Analysis (Classification)

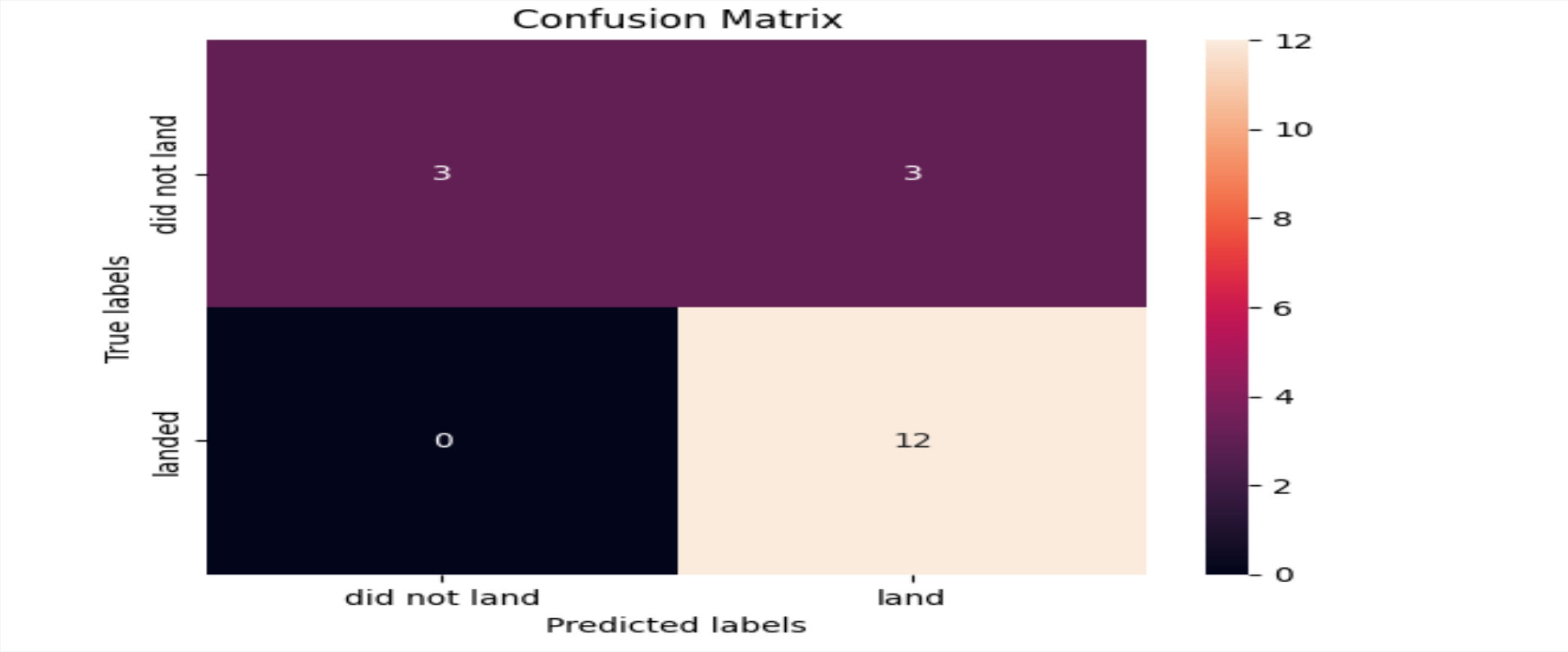
Classification Accuracy

- Model Accuracy: Decision Tree = 0.944
- Model Accuracy: K-Neighbors = 0.833
- Model Accuracy: Logistic Regression: 0.833
- Model Accuracy: Support Vector: 0.833

- The decision tree classifier is the model with the highest classification accuracy

Confusion Matrix

The logistic regression has the best confusion matrix because it can distinguish between the different classes. The major problem is false positives.



Conclusions

Based on the results of this project, we can conclude that:

- The greater the number of flights at a launch site, the greater the success rate.
- Launch success rate was seen to increase from 2013 to 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success launch rates.
- Of all the fours sites, KSC LC-39A seems to have had the most successful launches.
- The Decision tree classifier emerged as the best machine learning algorithm for this project.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!