IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mark Stephenson
7/18/2025

Outline

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

Executive Summary

- This project explores SpaceX launch data to predict the success of **first stage landings**, a critical factor in reducing the cost of space missions. SpaceX is a leader in commercial spaceflight, largely due to its ability to **reuse rocket boosters**. With launch costs nearly **60% lower** than competitors, predicting whether a first stage lands successfully can offer valuable insights into mission costs and operational efficiency.
- Using publicly available data and machine learning techniques, this project aims to:

Analyze how features such as **payload mass**, **launch site**, **orbit type**, and **flight history** influence landing success.

Assess how **landing success rates have evolved over time**.

Identify the **most effective machine learning model** for predicting binary outcomes (success vs. failure).

- The ultimate goal is to provide a predictive framework that could support future decision-making for clients, engineers, and competitors in the aerospace industry.

Introduction

- **SpaceX** is a trailblazer in the commercial space industry, known for drastically reducing launch costs by reusing the first stage of its Falcon 9 rockets.
- While traditional launch providers charge over $165 million, SpaceX offers launches for just $62 million — thanks to successful booster landings.
- Understanding what factors influence a successful landing can help predict launch costs and optimize mission planning.

What this Project Explores:

- How do features like payload mass, launch site, flight history, and orbit type affect first stage landing success?
- Has the landing success rate improved over time?
- What is the most accurate machine learning model to predict landing outcomes?
- Using public data and ML techniques, this project aims to uncover the patterns behind successful landings — and their cost-saving potential.

Section 1

Methodology

Methodology

- Data collection methodology:
  - Using SpaceX Rest API
- Perform data wrangling
  - Filtering the data
  - Dealing with missing values
  - Using One Hot Encoding to prepare the data to a binary classification
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluation of classification models to ensure the best results

Data Collection

- To gather a complete view of SpaceX launch history, we used a combination of **API access** and **web scraping**:
- **SpaceX REST API** provided structured data on key launch attributes.
- **Web scraping** from the official SpaceX Wikipedia page was used to supplement missing details and ensure comprehensive coverage.

Data collected via the SpaceX API includes:

- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite
- Outcome, Flights, GridFins, Reused, Legs, LandingPad
- Block, ReusedCount, Serial, Longitude, Latitude

Together, these methods enabled a richer, more detailed dataset for analysis

Data Collection – SpaceX API

- Launch data was retrieved directly from the **SpaceX REST API** using a **GET request**.
  The response, returned in **JSON format**, was then **parsed and decoded**, and the results were converted into a structured **Pandas DataFrame** for analysis.

- GitHub URL of the completed Data Collection API calls notebook :

https://github.com/mstephenson09/Data-Science-Capstone/blob/main/Spacex%20Data%20Collection%20API.ipynb

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API
```

We should see that the request was successfull with the 200 status response code

In [10]:
```
response=requests.get(static_json_url)
```

In [11]:
```
response.status_code
```

Out[11]: 200

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

In [26]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = response.json()
data_df = pd.json_normalize(data)
```

Using the dataframe `data` print the first 5 rows

In [27]:
```
# Get the head of the dataframe
data_df.head()
```

Data Collection - Scraping

- Used **BeautifulSoup** and **Requests** to scrape **Falcon 9 historical launch data** from Wikipedia.
- Extracted launch records from an HTML table on the Falcon 9 Wikipedia page.
- Parsed the table content and **converted it into a structured Pandas DataFrame** for analysis.
- This step ensured access to additional launch details not available through the API alone.
- GitHub URL of the completed webscraping notebook:

https://github.com/mstephenson09/Data-Science-Capstone/blob/main/Webscraping.ipynb

In [6]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
print(f"Response status code: {response.status_code}")
```

```
Response status code: 200
```

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

In [8]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

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

In [10]:
```
# Use soup.title attribute
print(soup.title)
```

Data Wrangling

In [11]:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key, value in df['Outcome'].items():
     if value in bad_outcomes:
        landing_class.append(0)
     else:
        landing_class.append(1)
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

In [12]:

```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[12]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

- After collecting and organizing the data into a **Pandas DataFrame**, we **filtered the dataset** using the `BoosterVersion` column to include only **Falcon 9 launches**.
- **Missing values** in the `PayloadMass` column were handled by replacing them with the **mean payload mass** for consistency and completeness.

- GitHub URL of completed data wrangling related notebooks: https://github.com/mstephenson09/Data-Science-Capstone/blob/main/Data%20wrangling.ipynb

EDA with Data Visualization

- Conducted **Exploratory Data Analysis (EDA)** and **feature engineering** using **Pandas** and **Matplotlib**.
- Key Visualizations:
- **Scatter Plots** to explore relationships between:
  - Flight Number vs. Launch Site
  - Payload Mass vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload Mass vs. Orbit Type
- **Bar Chart** to compare **success rates across orbit types**
- **Line Plot** to visualize **yearly trends in launch success**
- GitHub URL of completed EDA with data visualization notebook:

https://github.com/mstephenson09/Data-Science-Capstone/blob/main/EDA%20with%20Visualization.ipynb

EDA with SQL

- Performed SQL queries:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string ‘CCA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
- Ranking 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 of completed EDA with SQL notebook:

https://github.com/mstephenson09/Data-Science-Capstone/blob/main/EDA%20With%20SQL.ipynb

Build an Interactive Map with Folium

Launch Site Markers:

- Plotted all launch sites using their **latitude and longitude coordinates**.
- Each site includes a **circle marker, popup label**, and **text label** for clear identification.
- NASA Johnson Space Center was used as the **initial map center**.

Launch Outcome Markers:

- Used **colored markers** to represent launch outcomes:
  - **Green** for success
  - **Red** for failure
- Implemented **marker clustering** to visualize outcome density and identify sites with higher success rates.

Proximity Analysis:

- Added **lines and labels** to visualize distances from **KSC LC-39A** (as an example) to nearby:
  - Railways
  - Highways
  - Coastline
  - Closest city
  - These spatial insights help assess the **strategic location** of launch sites in relation to infrastructure and geography.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose: https://github.com/mstephenson09/Data-Science-Capstone/blob/main/EDA%20with%20Visualization.ipynb

Build a Dashboard with Plotly Dash

- Launch Site Dropdown
  - Implemented a **dropdown menu** to allow users to select a specific **launch site** or view data across **all sites**.
- Success Rate Pie Chart
  - Displays:
    - **Total successful launches** when "All Sites" is selected.
    - **Success vs. Failure distribution** for an individual site when selected.
- □ Payload Mass Slider
  - Added a **range slider** to filter data by **payload mass**, enabling targeted analysis.
- Scatter Plot: Payload vs. Success
  - Created a **scatter chart** showing the **relationship between payload mass and launch success**.
  - Colored by **booster version** to observe performance variations.
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

https://github.com/mstephenson09/Data-Science-Capstone/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

- To prepare for model training and evaluation, the following steps were taken:
- Loaded the cleaned dataset into a **Pandas DataFrame** and identified Class as the **target label**.
- Converted the Class column to a **NumPy array** (Y) to serve as the **outcome variable**.
- Applied **feature standardization** using StandardScaler() from **Scikit-learn** to ensure uniform scaling of all numeric features.
- **Split the dataset** into training and testing subsets using train_test_split, with:
  - **80% for training**
  - **20% for testing**
  - **Random state set to 2** for reproducibility
  - These steps ensured the data was ready for effective **model selection and performance evaluation**.

Predictive Analysis (Classification)

- To identify the best predictive algorithm for classifying launch success, several supervised learning models were evaluated:
- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **K-Nearest Neighbors (KNN)**
- **Decision Tree Classifier**
- Each model was trained and tested on the **standardized dataset**, and their performance was compared using **accuracy metrics**.

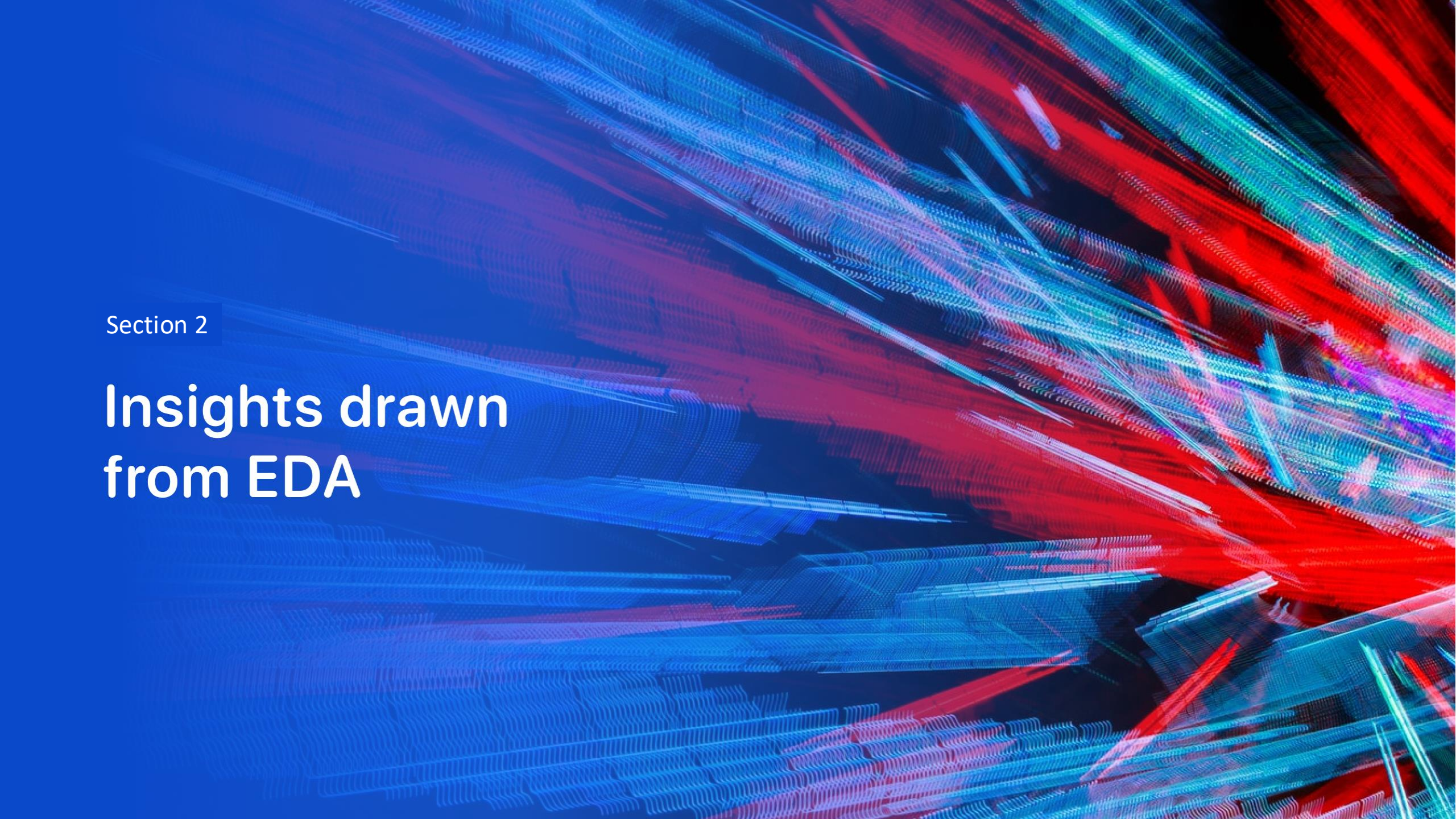
Model	Accuracy (%)
Logistic Regression	83%
SVM	84%
KNN (k=5)	80%
Decision Tree	78%

Support Vector Machine (SVM) demonstrated the **highest accuracy** and generalization performance.

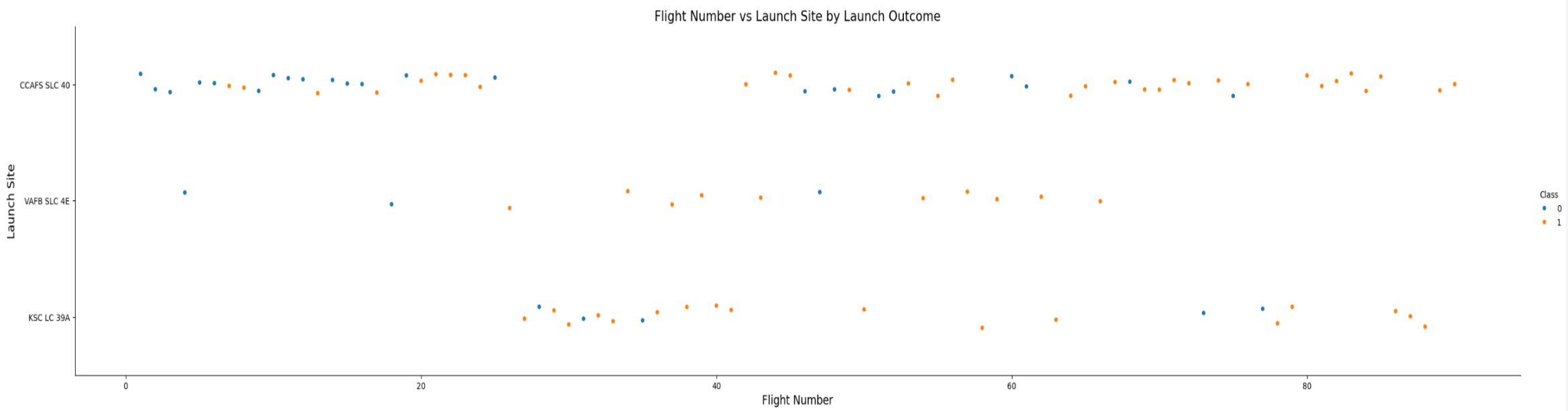
GitHub URL of completed predictive analysis lab, as an external reference and peer-review purpose:
https://github.com/mstephenson09/Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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


Section 2
Insights drawn from EDA

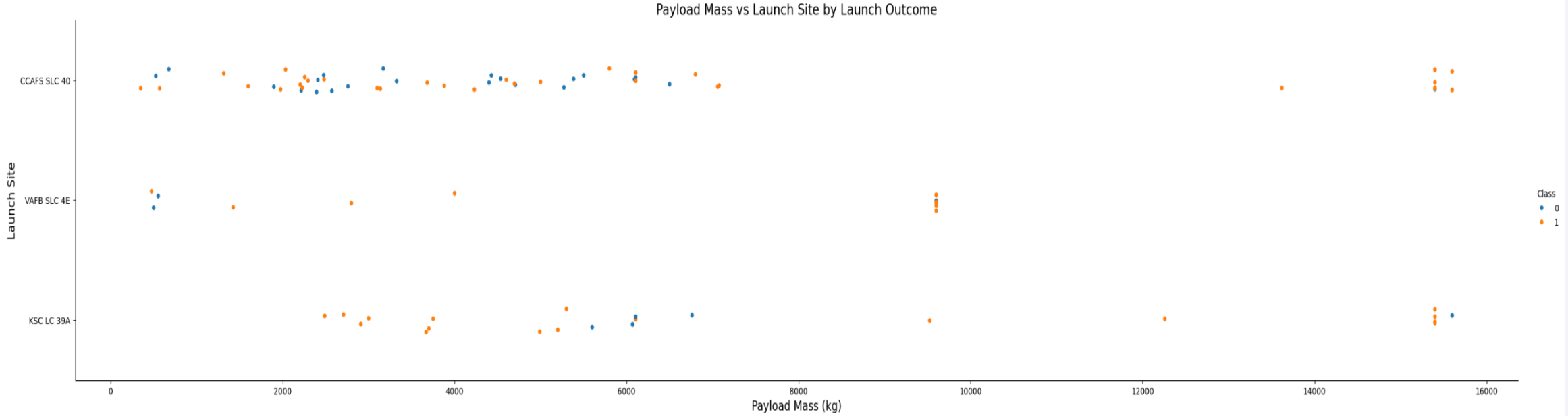
Flight Number vs. Launch Site



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

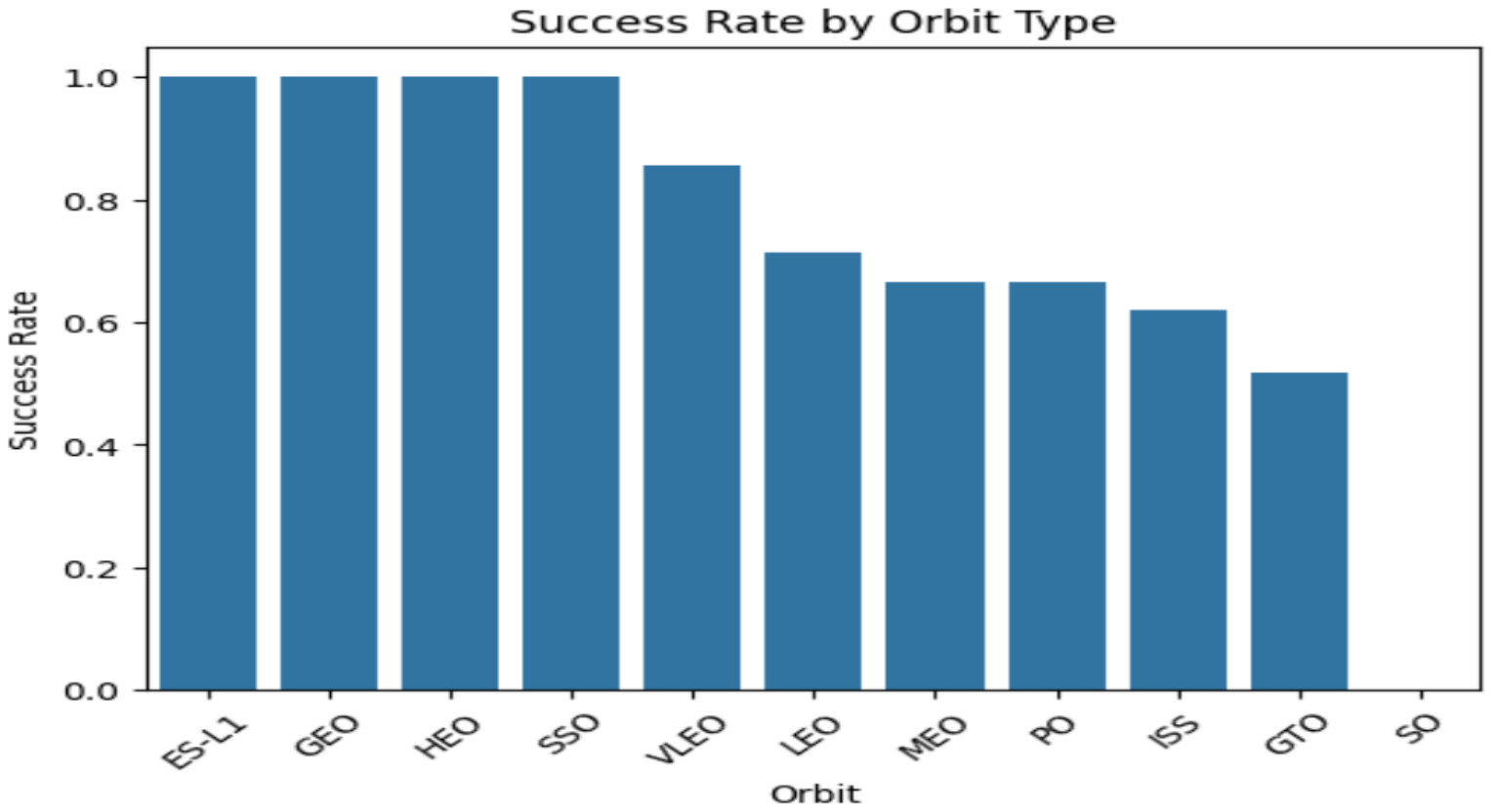
Launches at KSC LC-39A and CCAFS LC-40 show increasing success as FlightNumber increases. VAFB SLC-4E has fewer launches, and mostly lower Flight Numbers — success pattern less clear.

Payload vs. Launch Site



Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



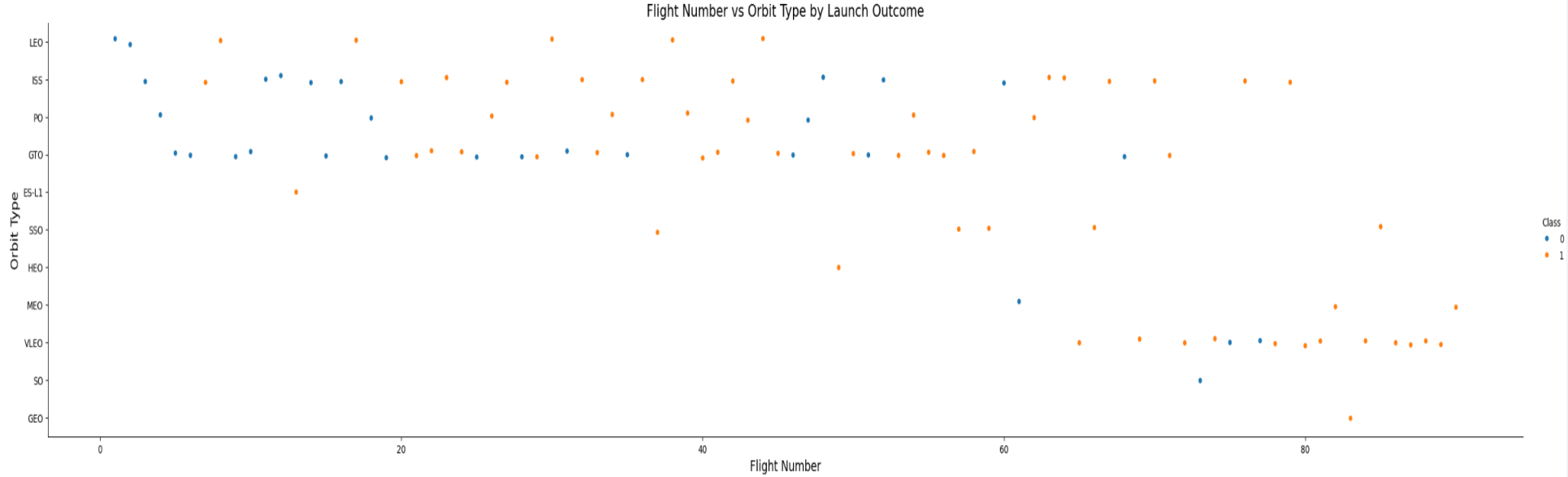
Analyze the plotted bar chart to identify which orbits have the highest success rates.

Orbits with 100% success rate are: ES-L1 GEO HEO SSO

Orbits with 0% success rate are: SO

Orbits with success rate between 50% and 85%: GTO ISS LEO MEO PO

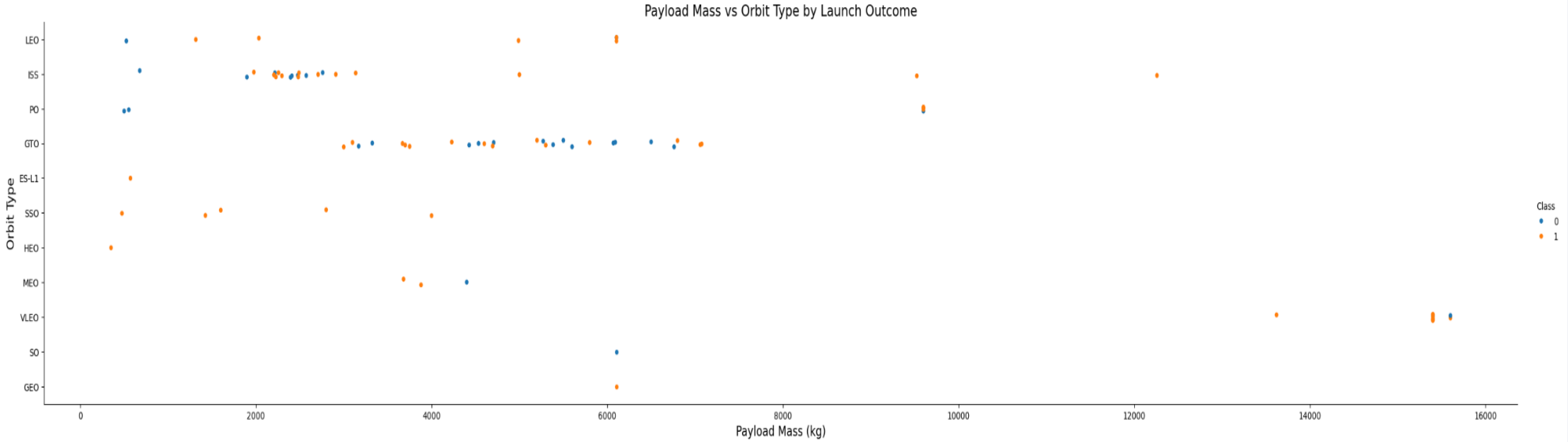
Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

File display

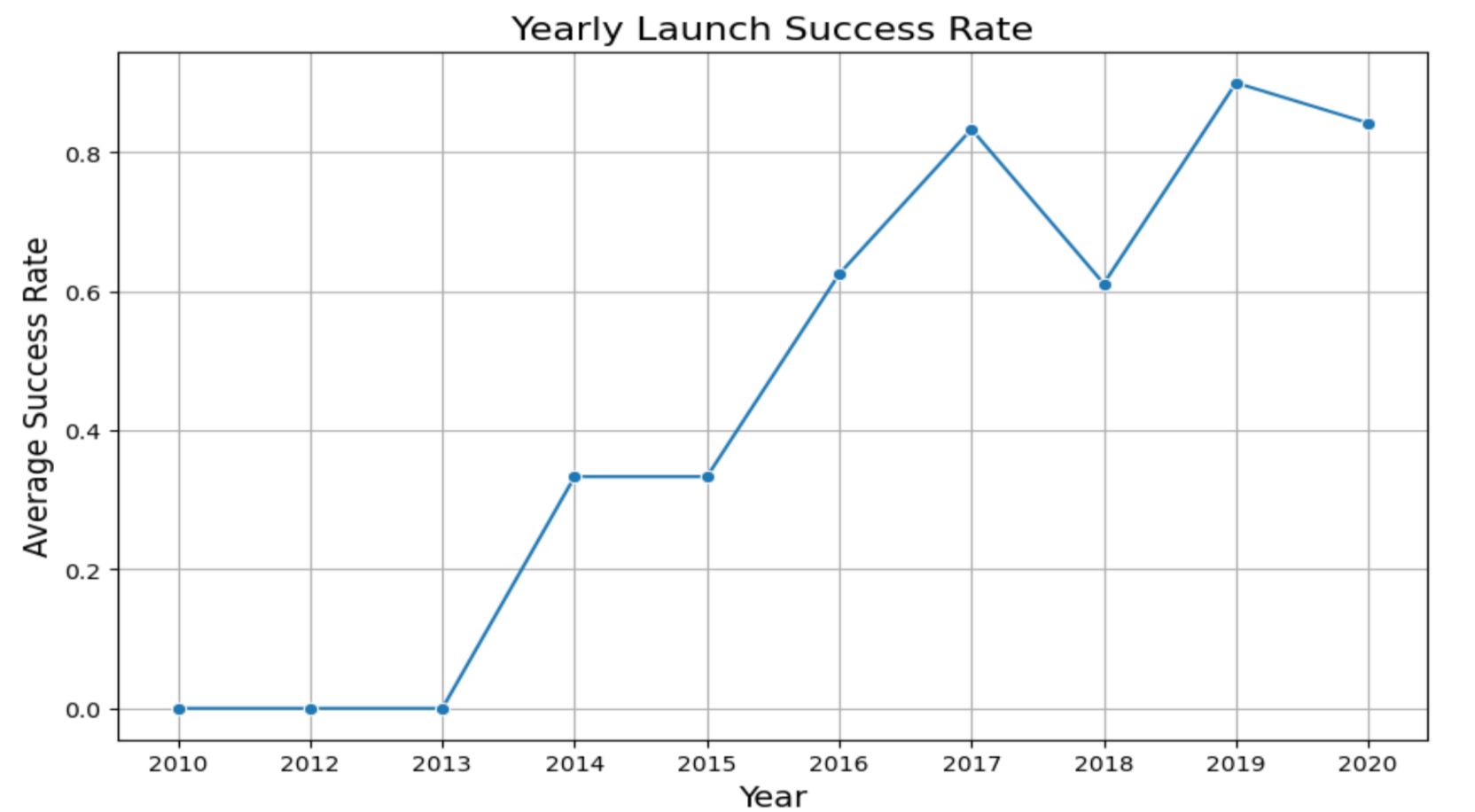
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- All launch sites are taking as a list called launch site. Using distinct we get all the unique different sites used in launching

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

In [25]:
```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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

Out[25]:

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

Launch Site Names Begin with 'CCA'

- This is the list of launch site specifically starts with CCA with the of LIKE query and using LIMIT query to get only top 5 results.

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

In [28]:
```
%sql SELECT*FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

Out[28]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

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

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

In [29]:
```
%sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

Out[29]:

Total_Payload
48213

Average Payload Mass by F9 v1.1

- The average payloadmass can be find by using AVG on the column of payload_mass_kg

```
In [30]: %sql SELECT AVG("Payload_Mass__kg_") AS Avg_Payload FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

Out[30]:

Avg_Payload
2928.4

First Successful Ground Landing Date

- Listing the date when the first successful landing outcome in ground pad was achieved.

```
In [31]: %sql SELECT MIN(Date) AS First_Ground_Success FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
 * sqlite:///my_data1.db
Done.
Out[31]: First_Ground_Success
         2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [33]:
```
%sql SELECT "Booster_Version", "Payload_Mass__kg_" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Pa
```

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

Out[33]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

In [34]:
```
%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

Out[34]:

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

File display

Boosters Carried Maximum Payload

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

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

In [35]:
```
%sql SELECT "Booster_Version", "Payload_Mass__kg_" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__k
```

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

Out[35]:

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

2015 Launch Records

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

In [36]:
```
%sql SELECT substr(Date, 6, 2) AS Month,"Landing_Outcome","Booster_Version","Launch_Site" FROM SPACEXTABLE WHERE substr(Date
```

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

Out[36]:

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

- 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

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 [37]:
```
%sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date >= '2010-06-04' AND Date <= '2017-03-20'
```

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

Out[37]:

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

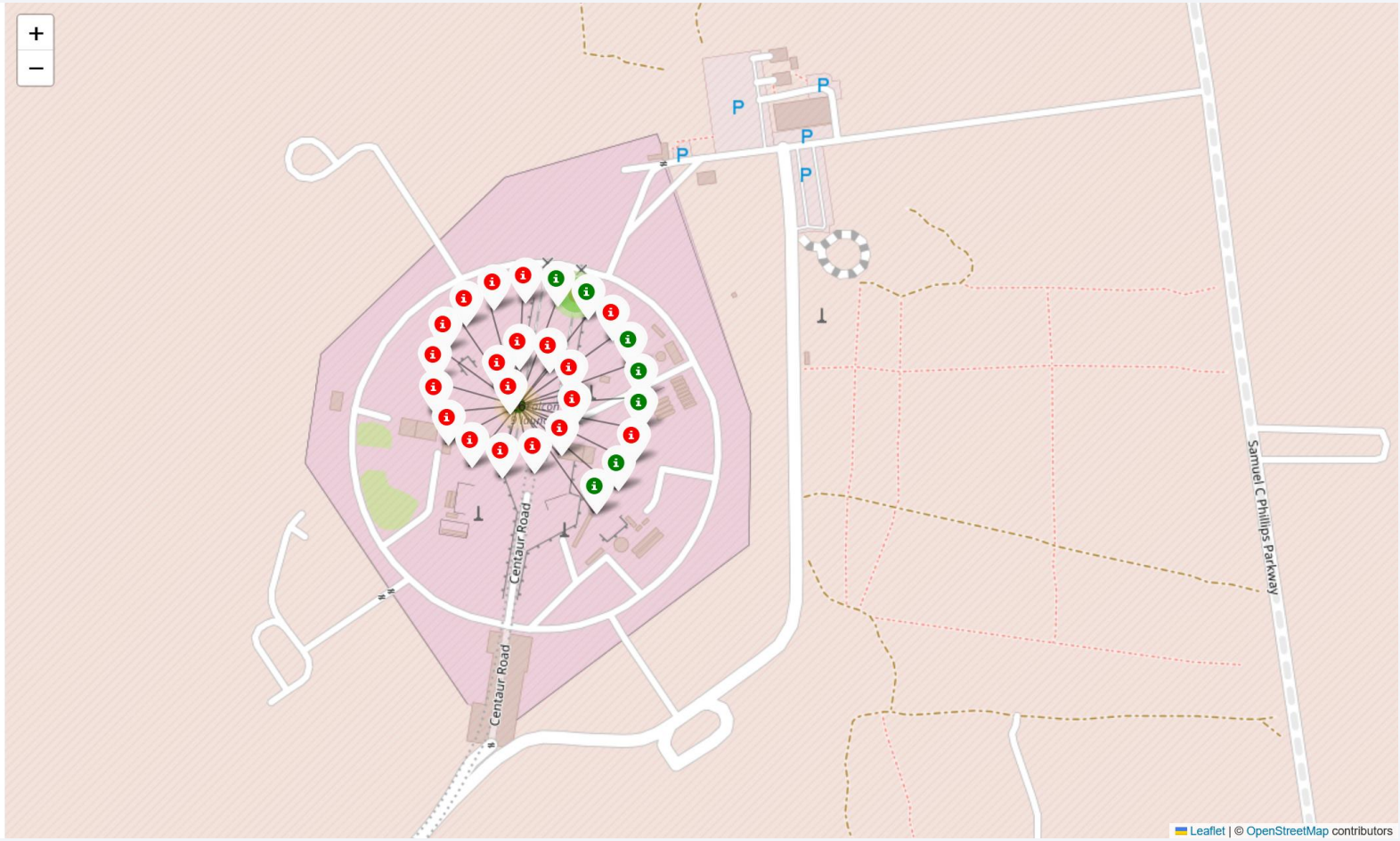

Section 3
Launch Sites
Proximities Analysis

Folium Map

- There are total 4 launch sites of SpaceX:
  - VAFB SLC-4E: Vandenberg Space Launch Complex 4 (CA)
  - KSC-LC29A: Kennedy Space Center - Merritt Island (FL)
  - CCAFS-LC40: Cape Canaveral Launch Complex 40 (FL)
  - CCAF-SLC40: Cape Canaveral Space Launch Complex 40(FL)

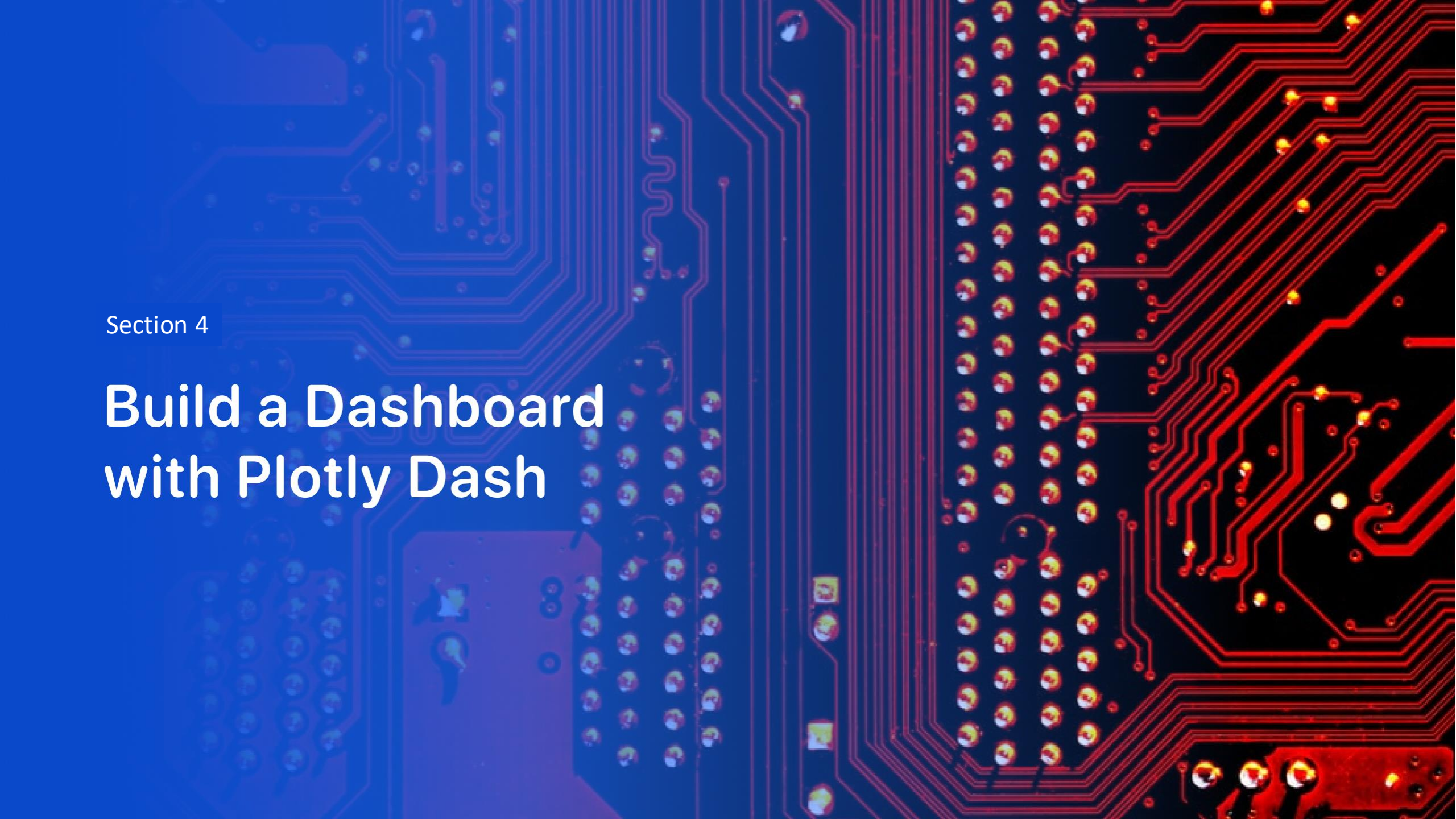
Folium Map Cont.

- Launch site CCAFS SLC-40 have two launch site close to each other.
- The first site have 26 launches out of which only 7 were successful while the other left representation shows total 7 launches out of which only 3 were successful.
- In general it is clear that the launches from this area are mostly unsuccessful.



Folium Map Cont. 3

- Launch site we use to check the distance with highways, train ways and city is Cape Canaveral (FL) CCAFS-SLC40
- This map shows the line that defines the closest distance of launch site with the highway.
- Distance with the highway is 0.58 km
- Melbourne is the closest big city from the launch site which is around 51.43 km
- The distance with railway line is almost 3.28 km.

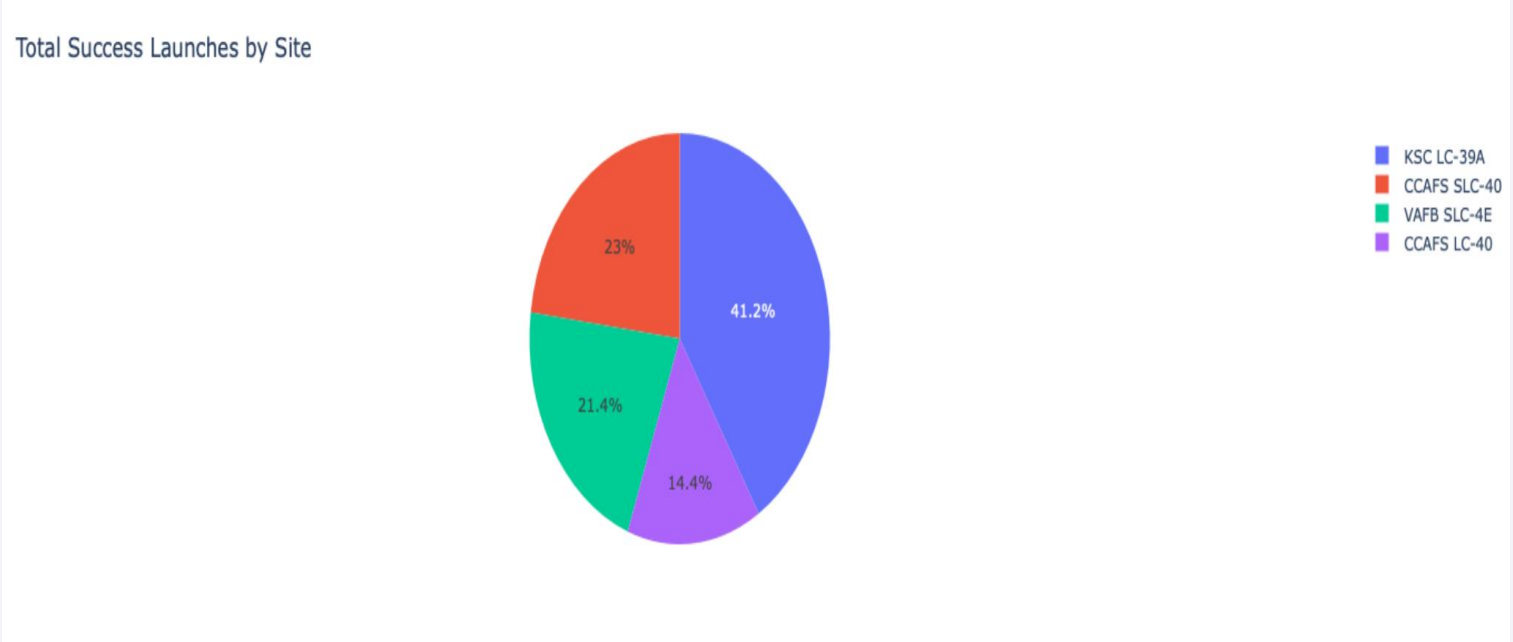

Section 4
Build a Dashboard with Plotly Dash

Plotly Dashboard – Launch Successes

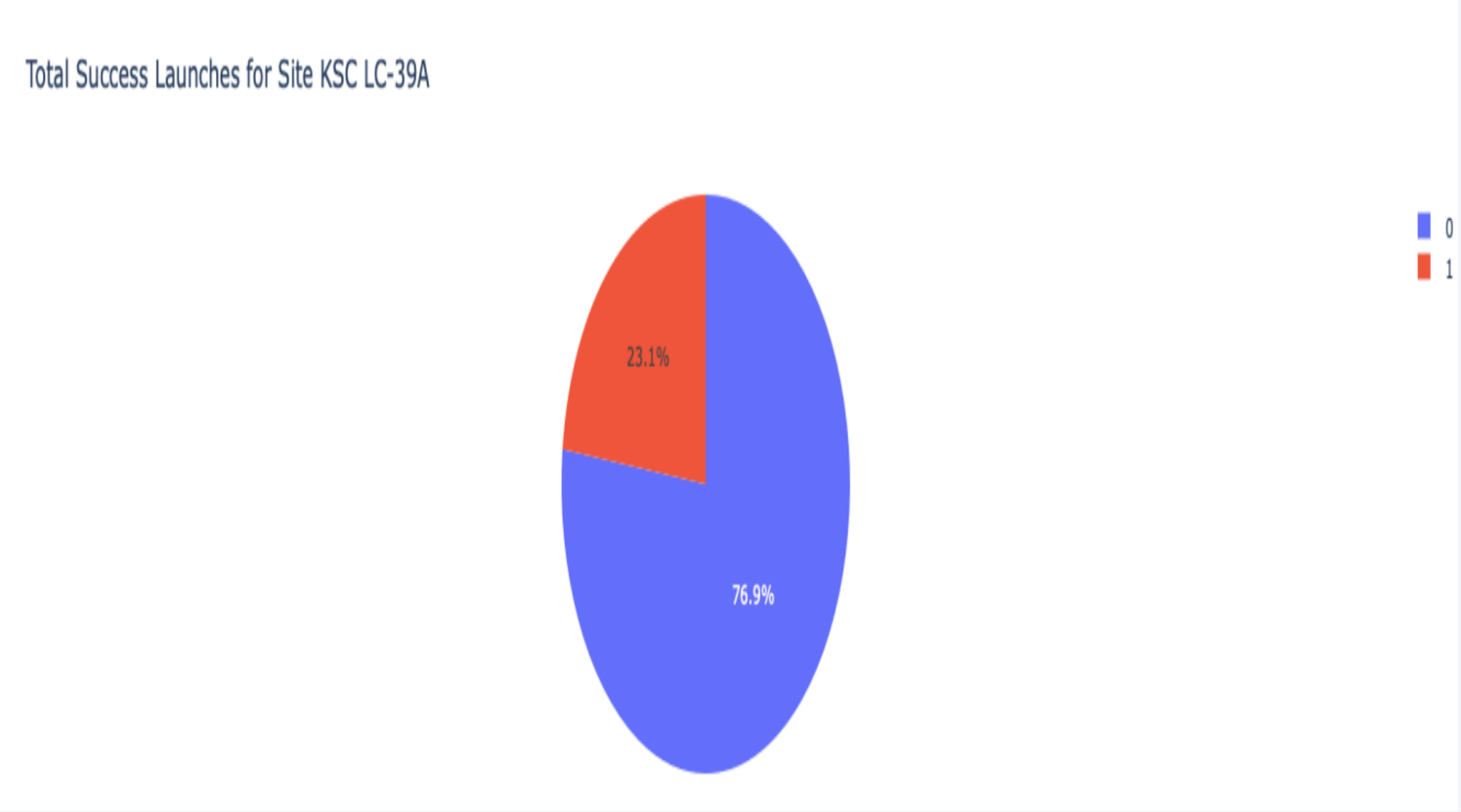
We developed an interactive dashboard using **Plotly Dash**, featuring:

- **Dropdown Menu** to select and filter data by **launch site**
- **Pie Chart** to display **launch success rates** overall or per site
- **Scatter Plot** to visualize the relationship between **payload mass**, **launch site**, and **outcomes**
- **Range Slider** to filter results by **payload mass (kg)** for targeted analysis

This dashboard allows users to explore SpaceX launch data dynamically and gain insights in real time.

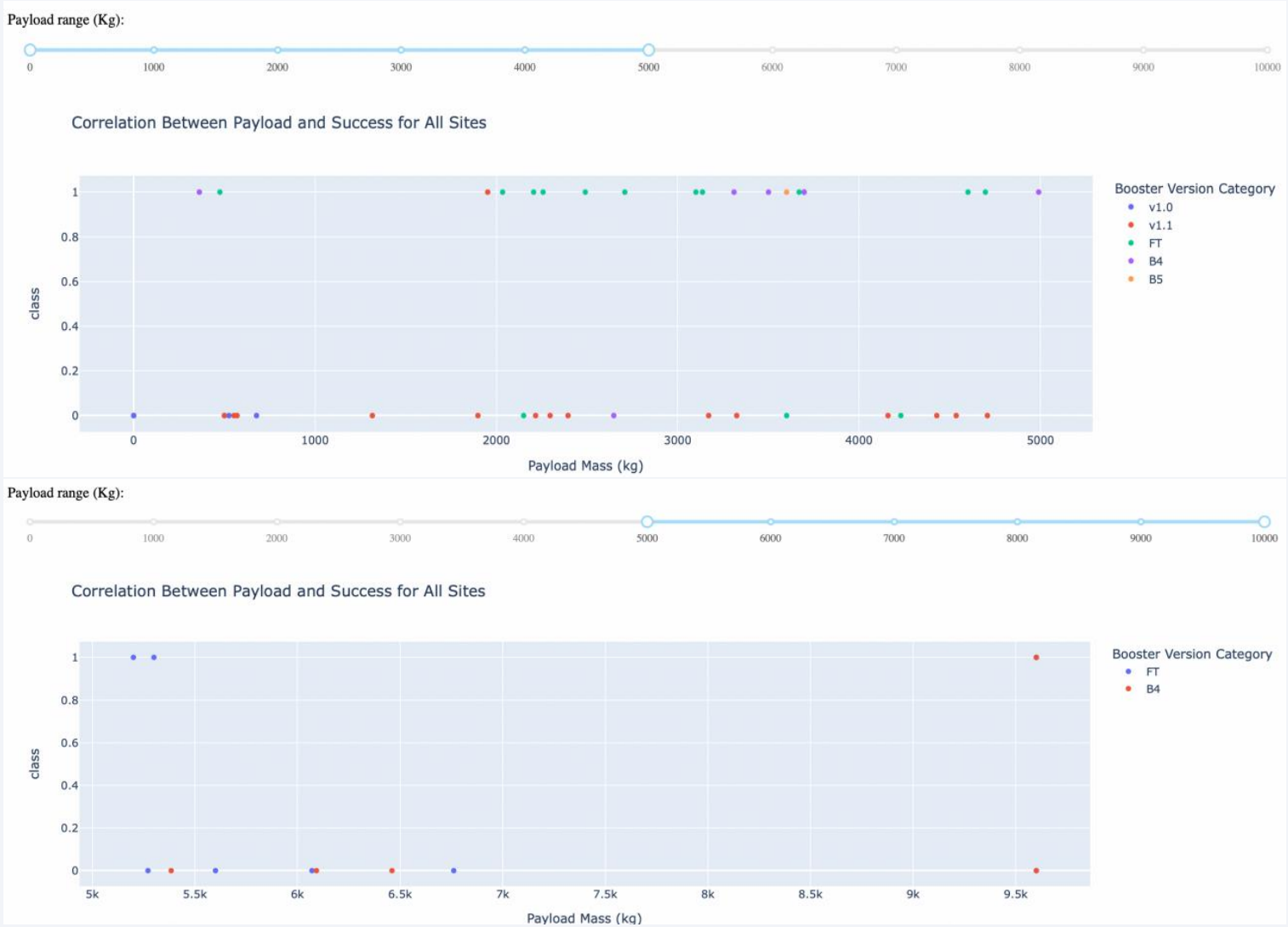


Plotly Dashboard – Total Success



- KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

Plotly Dashboard - Payload vs.Launch Outcome



Explanation:

- The charts show that payloads between 2000 and 5500 kg have the highest success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

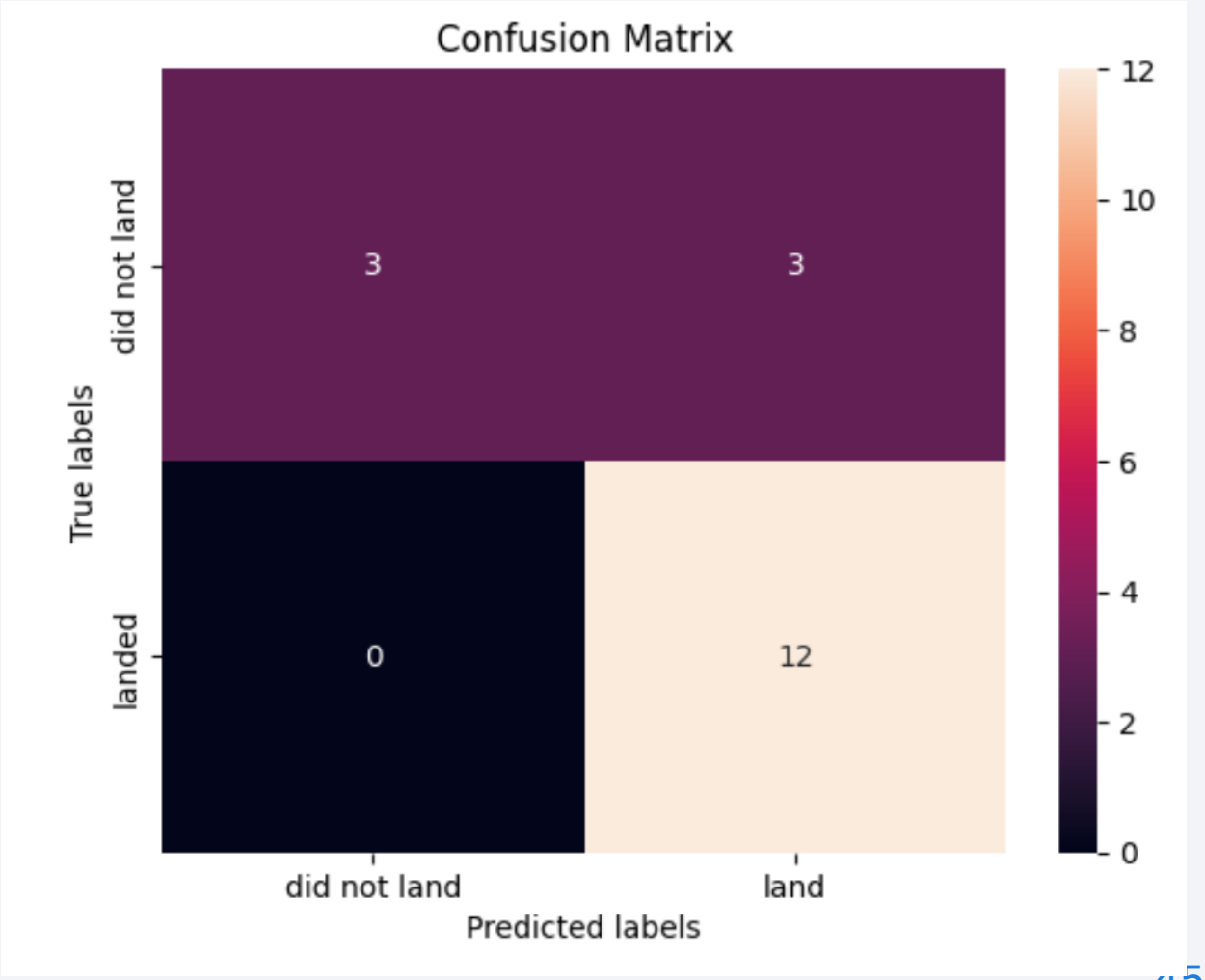
- The results are presented in two columns:
- **Prediction Method**
- **Test Accuracy Score**
- We evaluated **four predictive models**:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K-Nearest Neighbors (KNN)
- **Decision Tree** achieved the **highest accuracy of 0.88**, outperforming the others, which reached up to **0.83**.

[52]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.777778
KNN	0.833333

Confusion Matrix

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.



Conclusions

- **Decision Tree** proved to be the most effective algorithm for predicting launch success on this dataset.
- **Lower payload masses** are generally associated with **higher success rates**.
- Most **launch sites are located near the Equator** and are all in **close proximity to the coast**, likely for orbital efficiency and safety.
- **Launch success rates have improved steadily over the years**, indicating operational advancements.
- **KSC LC-39A** recorded the **highest launch success rate** among all launch sites.
- Orbits such as **ES-L1, GEO, HEO, and SSO** achieved a **100% success rate** in the dataset.

Thank you!