IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

George Bocioroaga
09/08/2025

Outline

Executive Summary

Summary of Methodologies

- **Data Collection** – SpaceX API & Wikipedia web scraping for launch data.
- **Data Wrangling** – Cleaning, transforming, and merging datasets.
- **EDA** – SQL, Pandas, Matplotlib for trend and correlation analysis.
- **Geospatial Analysis** – Folium to map and analyze launch site locations.
- **Interactive Dashboard** – Plotly Dash for dynamic visualizations.
- **Machine Learning** – Logistic Regression, SVM, Decision Tree, KNN for landing success prediction.

Summary of Results

- Optimal payload range: **2000–6000 kg** → highest success rate.
- **KSC LC-39A** had the highest launch success rate.
- All launch sites are near coastlines and favorable latitudes.
- **Decision Tree** achieved ~89% accuracy in landing predictions.
- Interactive dashboard enables data exploration by site, payload, and booster type.

Introduction

Project Background & Context

- SpaceX's Falcon 9 first stage is designed to be reusable, reducing the cost of space launches.
- Successful landings of the first stage are critical for reusability and cost efficiency.
- Historical launch data contains valuable patterns that can help predict landing success.

Problems to Answer

- What factors influence the success of a Falcon 9 first stage landing?
- Which launch sites have the highest success rates?
- Is there an optimal payload mass range for maximizing landing success?
- Can we build a model to predict landing success for future launches?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

We gathered launch records from two main sources:

1. **SpaceX REST API** – provided detailed technical and outcome data for each launch, including date, payload mass, orbit type, and landing results.
2. **Wikipedia Falcon 9 & Falcon Heavy launch history** – scraped to fill in any missing details and cross-check the API data.

The raw data included:

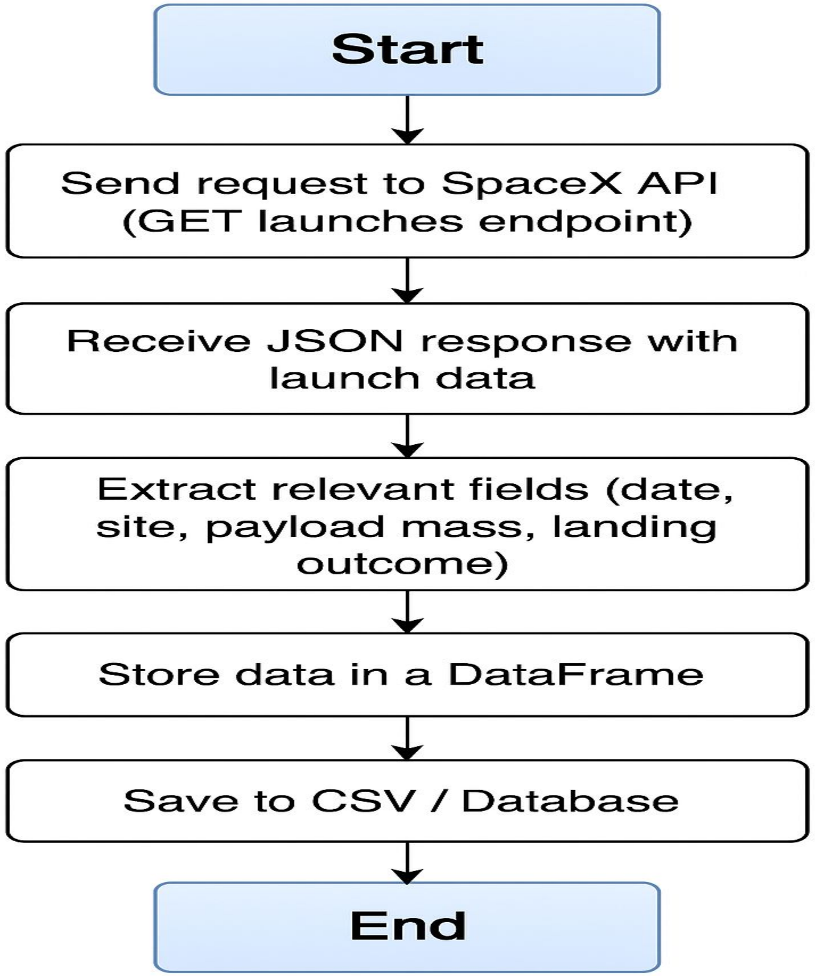
- Launch dates, sites, and rocket versions
- Payload mass and orbit information
- Landing types and success indicators

After collection, all datasets were combined into a single structured table, ready for cleaning and analysis.

Data Collection – SpaceX API

- Removed incomplete records and corrected inconsistent values (e.g., date formats, payload mass units).
- Standardized column names and merged API data with scraped Wikipedia data.
- Converted categorical fields (launch site, booster type, orbit) into machine-readable formats using one-hot encoding.
- Created a clear success indicator column for first stage landings.
- Saved the cleaned dataset in both CSV format and a database table for easy access during analysis.

GitHub:
https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/1_SpaceX_Data_Collection_API.ipynb



Data Collection - Scraping

Key Phrases:

- Target page: Wikipedia “Falcon 9/Heavy launch history”
- Fetch HTML; respect robots.txt & polite delays
- Parse tables/rows (BeautifulSoup / pandas.read_html)
- Extract fields: date, site, booster, payload mass, orbit, landing outcome
- Normalize types & units; handle missing/merged cells
- Iterate sections/years; follow anchors if needed
- De-duplicate (UTC date + rocket as key)
- Append to master DataFrame; save CSV/DB

GitHub: https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/2_SpaceX_Web_Scraping.ipynb

Data Wrangling

Key Phrases (add as bullets):

- Load raw datasets (SpaceX API JSON + Wikipedia tables)
- Unify schema: consistent column names & dtypes
- Clean missing values (drop/impute), normalize text, fix dates
- Standardize units (e.g., payload mass in kg)
- Remove duplicates (key = UTC date + rocket + flight)
- Outlier checks on payload and durations
- Feature engineering: one-hot for site/orbit/booster; derive `landing_success` flag
- Merge site geolocation (lat/lon) for mapping
- Quality checks (row counts, nulls, value ranges)
- Save cleaned dataset (CSV/DB) + data dictionary

GitHub:
https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/3_Space-X%20Data%20Wrangling%20spacex.ipynb

EDA with Data Visualization

Exploratory Data Analysis (EDA) — Charts & Purpose

- **Bar Charts** – Compared launch success counts across sites to identify the best-performing locations.
- **Pie Charts** – Showed proportion of successful vs. failed launches overall and per site.
- **Scatter Plots** – Visualized relationship between payload mass and landing success, colored by booster version.
- **Line Charts / Time Series** – Tracked launch success rates over time to detect trends.
- **Maps (Folium)** – Displayed geospatial distribution of launch sites and their proximity to coasts/equator.

Why these charts?

- Bar & pie charts: easy comparison of categorical success rates.
- Scatter plots: highlight correlations between payload and success probability.
- Time series: reveal patterns or improvements over years.
- Maps: provide geographical insight into strategic launch site placement.

GitHub: https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/1_SpaceX_Data_Collection_API.ipynb

EDA with SQL

ChatGPT said:

Here's a bullet-point summary of the SQL queries from the **SpaceX EDA Using SQL** notebook:

- Selected specific columns (e.g., launch site, payload mass, landing outcome) from the dataset for focused analysis.
- Counted total successful and failed launches across all sites.
- Filtered launches by specific sites (e.g., `KSC LC-39A`) to compare performance.
- Queried launches with payload mass in certain ranges (e.g., between 2000–4000 kg) to check effect on success rate.
- Grouped launches by orbit type to see which orbits had higher success rates.
- Found the average payload mass per launch site.
- Counted the number of launches per year to identify trends over time.
- Retrieved the top booster versions by success rate.
- Identified launches where landing outcome was successful and payload mass was in optimal range.
- Ordered query results by payload mass or date for trend inspection.

GitHub: https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/4_SpaceX_jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

Map Objects Created in Folium

- **Markers** – Placed at each SpaceX launch site to indicate exact locations.
- **Pop-up Labels** – Added to markers to display site name and basic launch info when clicked.
- **Circle Markers** – Used to highlight launch sites with a radius representing proximity or importance.
- **Lines** – Drew lines from each launch site to the equator to visualize latitude differences.
- **Marker Clusters** – Grouped close-by markers to keep the map uncluttered and easier to navigate.

Why These Objects Were Added

- **Markers & Pop-ups** – Give clear, interactive identification of each site.
- **Circles** – Provide a visual emphasis on launch site areas.
- **Lines** – Help illustrate how latitude might affect launches.
- **Marker Clusters** – Improve usability when zooming out and viewing all sites together.

GitHub:
https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/6_Space-X%20Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

Plots / Graphs Added

- **Pie Chart** – Shows proportion of successful launches overall or for a selected site.
- **Scatter Plot** – Displays correlation between payload mass and landing success, color-coded by booster version.

Interactions Added

- **Dropdown Menu** – Allows selecting a specific launch site or viewing all sites.
- **Payload Range Slider** – Filters the scatter plot by payload mass range.

Why These Were Added

- **Pie Chart** – Provides a quick, visual summary of success rates and allows site-level comparison.
- **Scatter Plot** – Helps identify trends between payload size, booster version, and success probability.
- **Dropdown & Slider** – Enable interactive exploration of the data without writing queries, making the dashboard more engaging and user-friendly.

GitHub:https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/7_SpaceX_Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.py

Predictive Analysis (Classification)

Key Phrases

- Loaded cleaned dataset with features and target (`landing_success`).
- Split data into training and test sets.
- Trained baseline models: Logistic Regression, SVM, Decision Tree, KNN.
- Evaluated using accuracy score and confusion matrix.
- Tuned hyperparameters via GridSearchCV for each model.
- Re-trained models with optimal parameters.
- Compared performance metrics to select the best model.
- **Decision Tree** achieved the highest accuracy (~89%).
- Saved the best model for future predictions.

GitHub:https://github.com/GeorgeBocioroaga/Applied-Data-Science-Capstone/blob/main/8_SpaceX%20Machine%20Learning%20Prediction.ipynb

Results

Exploratory Data Analysis — Results

- **KSC LC-39A** had the highest launch success rate.
- Optimal payload range: **2000–6000 kg** → highest landing success probability.
- All launch sites located near coasts and favorable latitudes.
- Booster version type influences landing success.

Interactive Analytics — Dashboard (Screenshots)

- **Pie Chart**: success distribution per site.
- **Scatter Plot**: payload vs. success, colored by booster type.
- **Filters**: site selection (dropdown) & payload mass range (slider) for dynamic exploration.

Predictive Analysis — Results

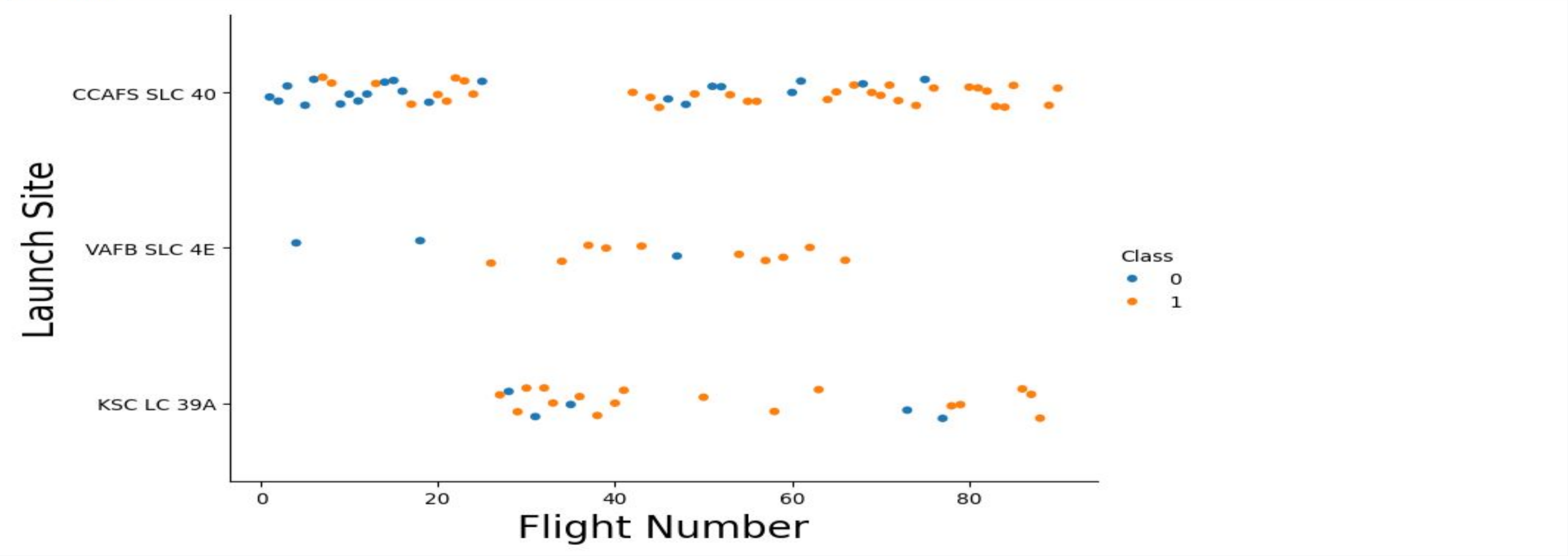
- Tested Logistic Regression, SVM, Decision Tree, KNN.
- Tuned models with GridSearchCV for optimal performance.
- **Decision Tree** achieved the best accuracy (~89%).
- Model predicts probability of first stage landing success for new launches.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 1.5 ,height=5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



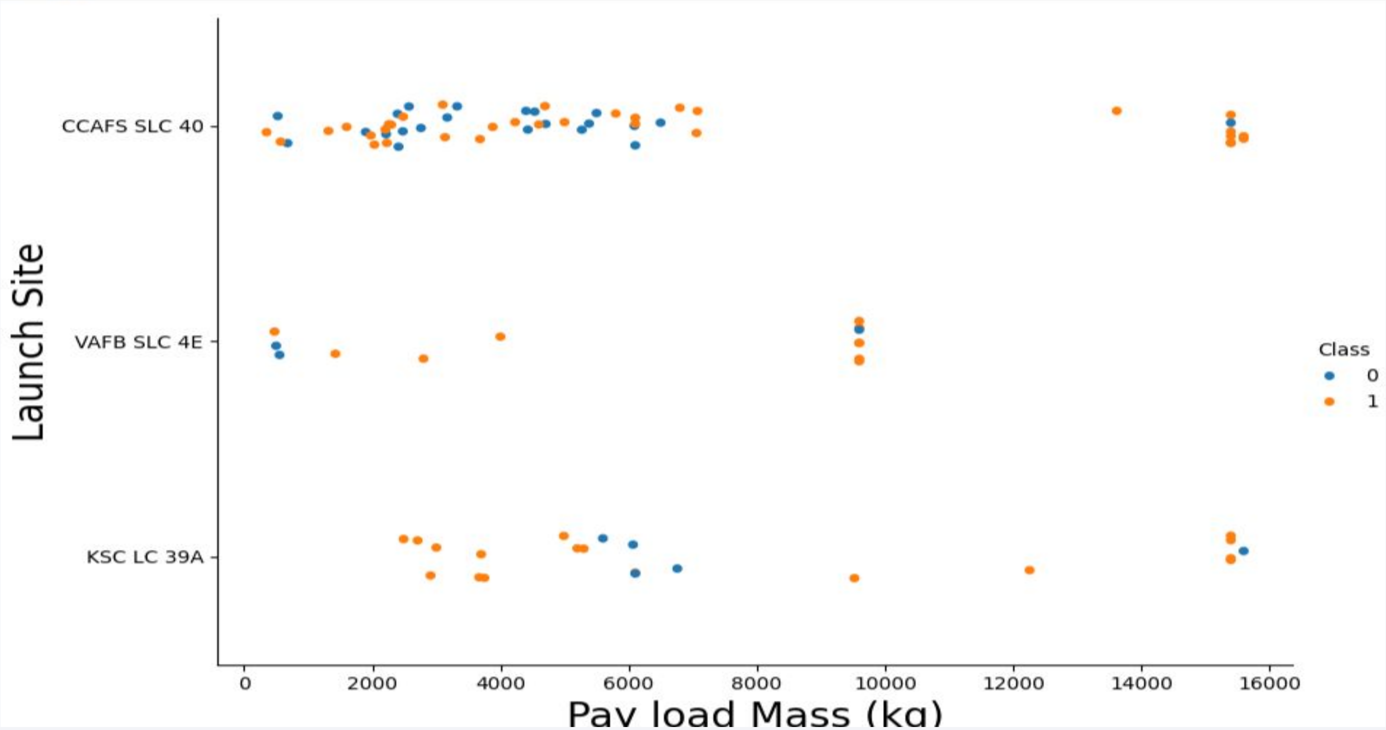
As flight number increases, success rates improve across all three major launch sites.

VAFB SLC-4E: 100% success rate after the 50th flight.

KSC LC-39A and CCAFS SLC-40: both reach 100% success after the 80th flight.

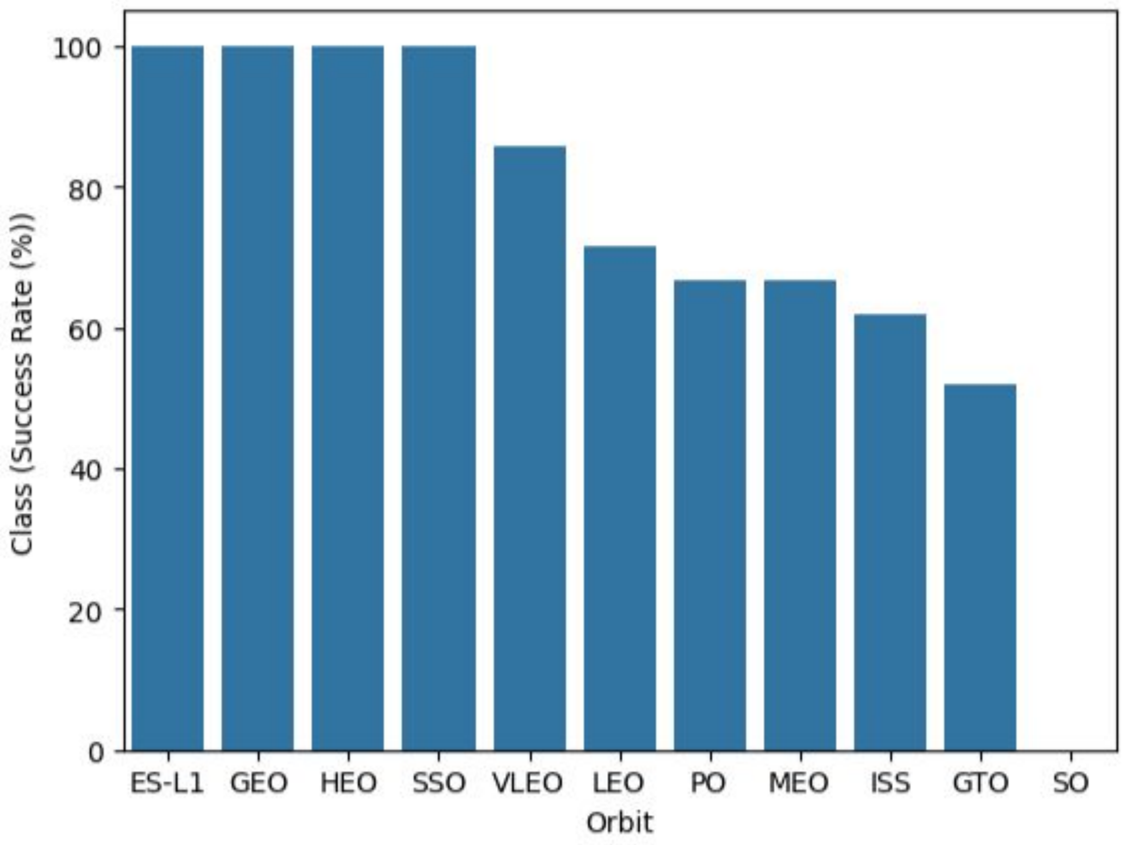
Payload vs. Launch Site

```
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 1.5 ,height=6)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



In the **Payload vs. Launch Site** scatter plot, the **VAFB SLC-4E** site shows **no launches with heavy payloads** (greater than 10,000 kg).

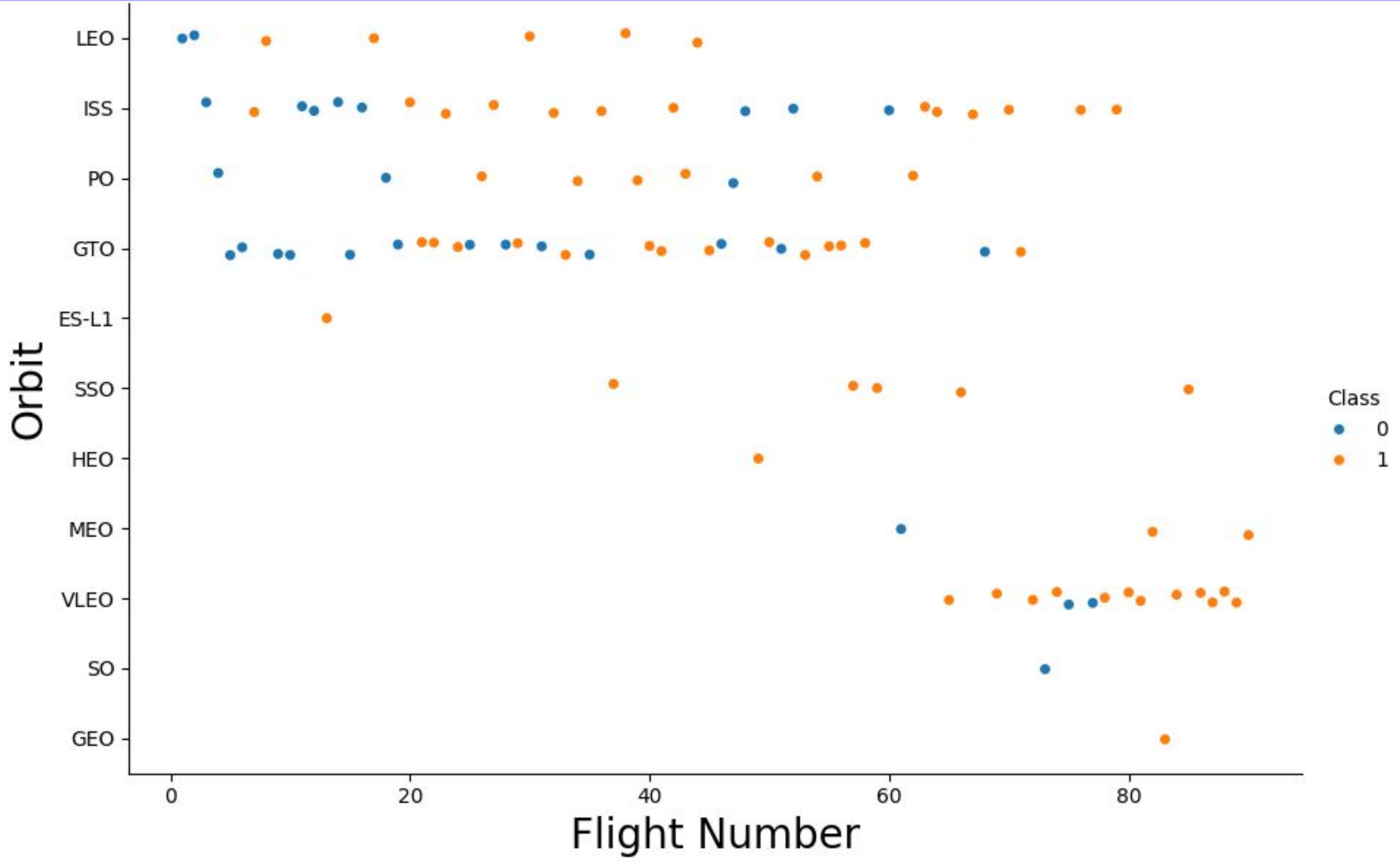
Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

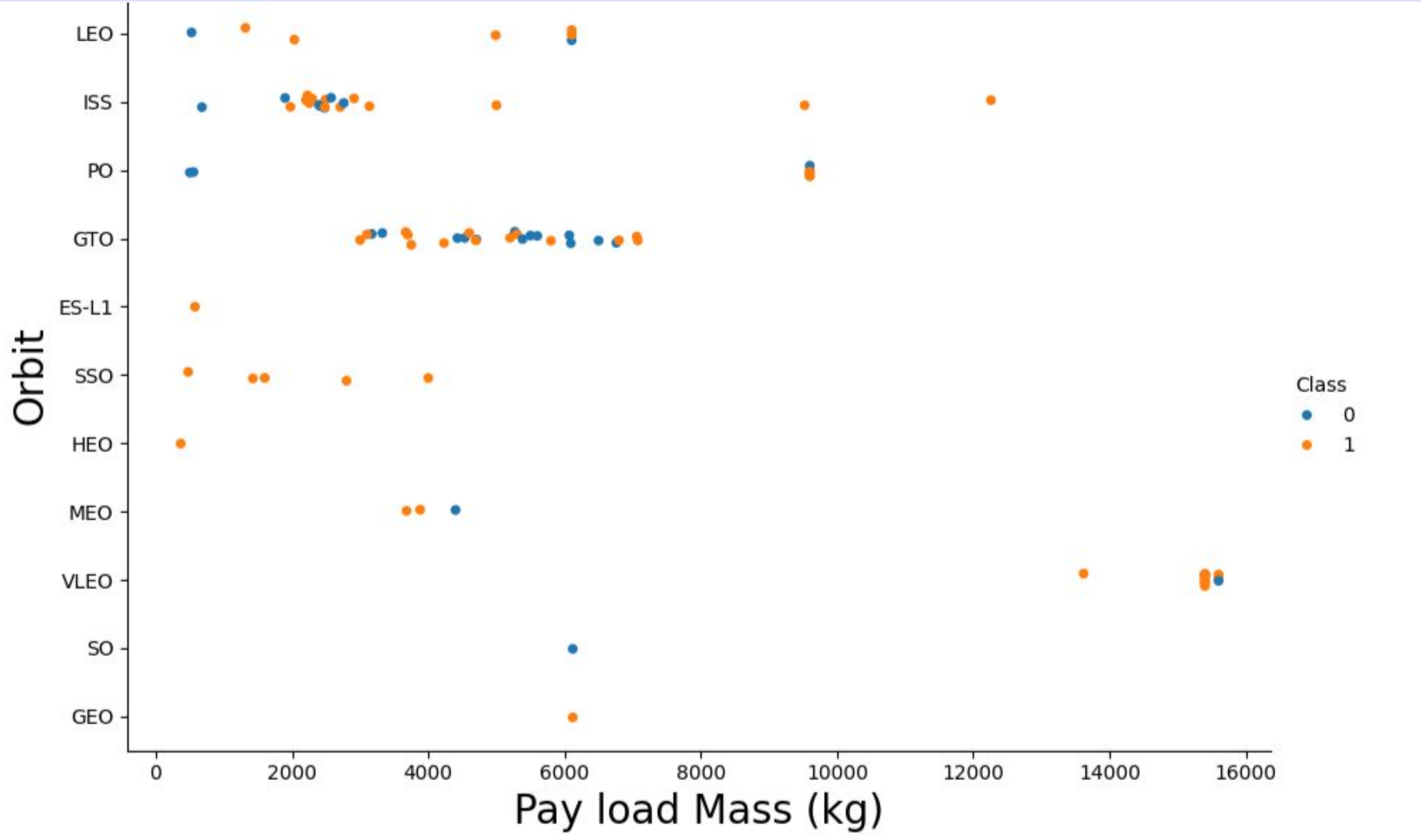
- ES-L1, GEO, HEO, and SSO orbits achieved a 100% success rate.
- GTO orbit recorded the lowest success rate (~50%) overall.
- SO orbit had 0% success.

Flight Number vs. Orbit Type



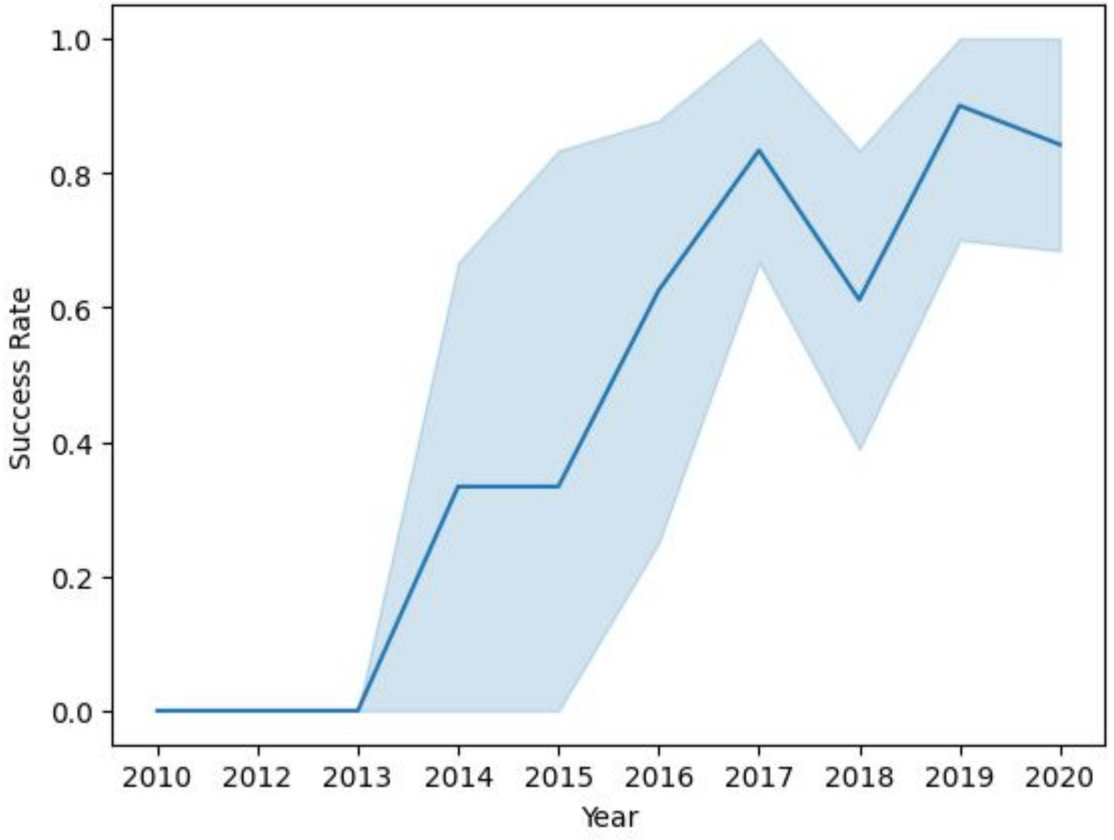
- In LEO orbit, success rate appears to improve as the number of flights increases.
- In GTO orbit, there is no clear relationship between flight number and success rate.

Payload vs. Orbit Type



- For heavy payloads, Polar, LEO, and ISS orbits show higher positive landing rates.
- In GTO orbit, the distribution of successes and failures is mixed, making it difficult to distinguish a clear trend.

Launch Success Yearly Trend



From 2013 to 2020, the overall launch success rate showed a steady upward trend, reaching its peak in 2020.

All Launch Site Names

Task 1

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

```
#df['Launch_Site'].unique()
```

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

Task 2

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

```
#df5 = df[df['Launch_Site'].str.startswith('CCA', na=False)]
#df5.head()
```

```
%sql SELECT * FROM 'SPACEXTBL' 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

Total Payload Mass

Task 3

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

```
tm = df.loc[df['Customer'] == 'NASA (CRS)', 'PAYLOAD_MASS__KG_'].sum()
tm
```

```
np.int64(45596)
```

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
avg = df.loc[df['Booster_Version'].str.startswith('F9 v1.1'), 'PAYLOAD_MASS__KG_'].mean()
avg
```

```
np.float64(2534.6666666666665)
```

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
datee = df.loc[df['Landing_Outcome'] == 'Success (ground pad)', 'Date'].min()
datee
```

'2015-12-22'

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

* sqlite:///my_data1.db

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

```
# %sql SELECT * FROM 'SPACEXTBL'
```

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_M
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
tn = df.loc[df['Mission_Outcome']
```

```
df['Mission_Outcome'].unique()
```

```
array(['Success', 'Failure (in flight)',
       'Success (payload status unclear)', 'Success '], dtype=object)
```

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

Task 8

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

In [21]:
```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_
```

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

Out[21]:

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Ou
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDEI
```

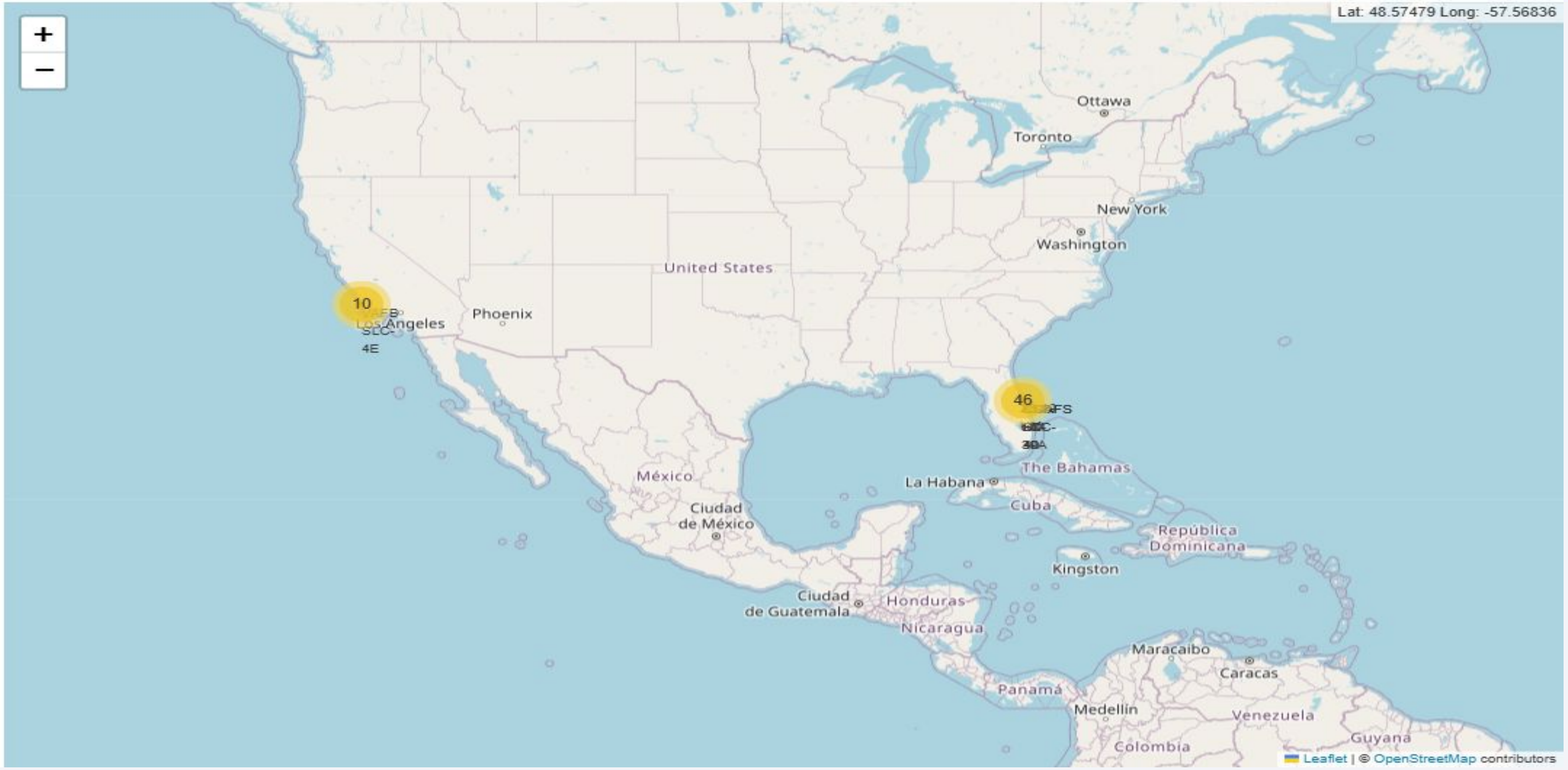
```
* sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-04-2018	22:51:00	F9 B4 B1045.1	CCAFS SLC-40	Transiting Exoplanet Survey Satellite (TESS)	362	HEO	NASA (LSP)	Success	Success (drone ship)
17-12-2019	00:10:00	F9 B5 B1056.3	CCAFS SLC-40	JCSat-18 / Kacific 1, Starlink 2 v1.0	6956	GTO	Sky Perfect JSAT, Kacific 1	Success	Success

Section 3

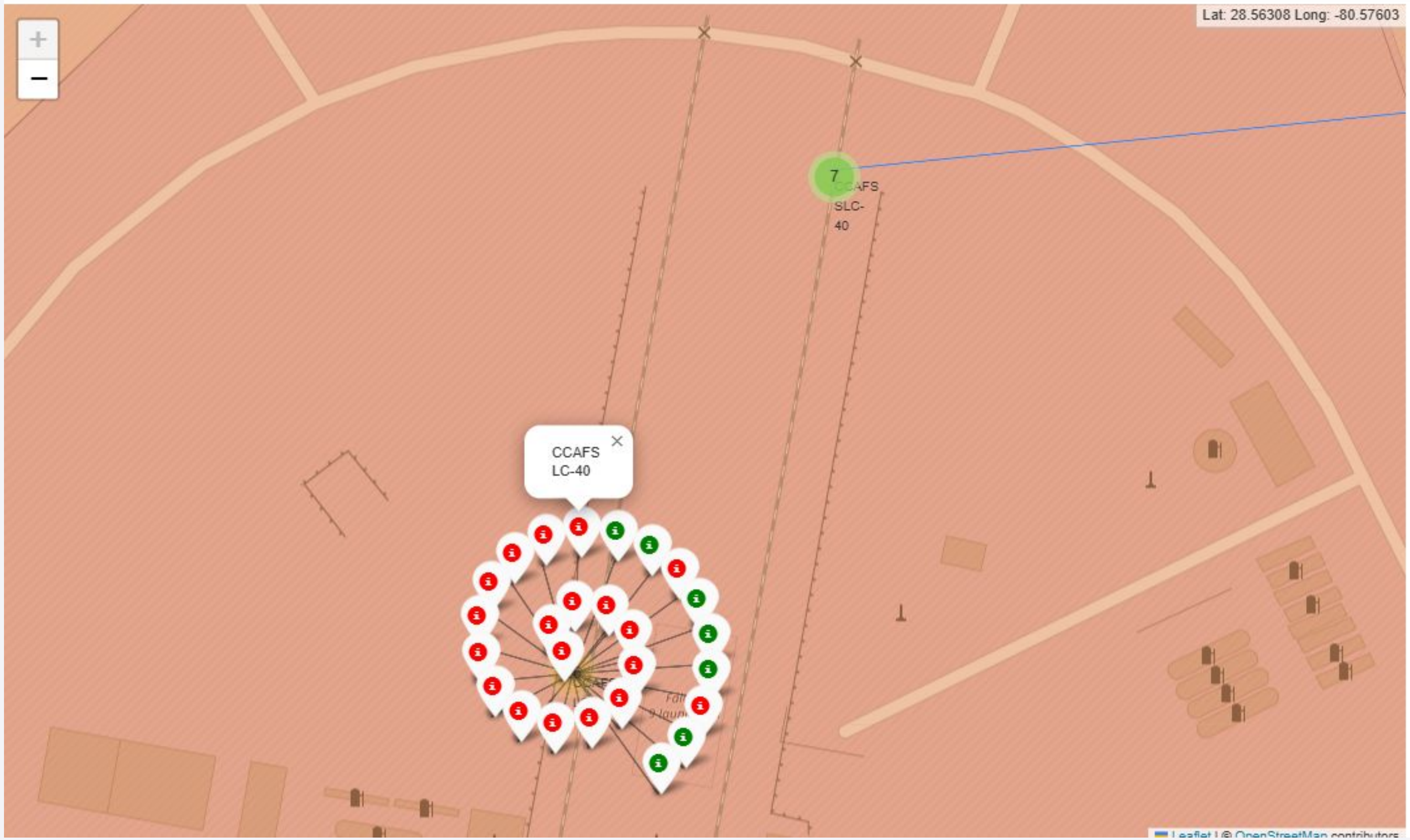
Launch Sites Proximities Analysis

Mark all launch sites on a map



Success/failed launches for each site on the map

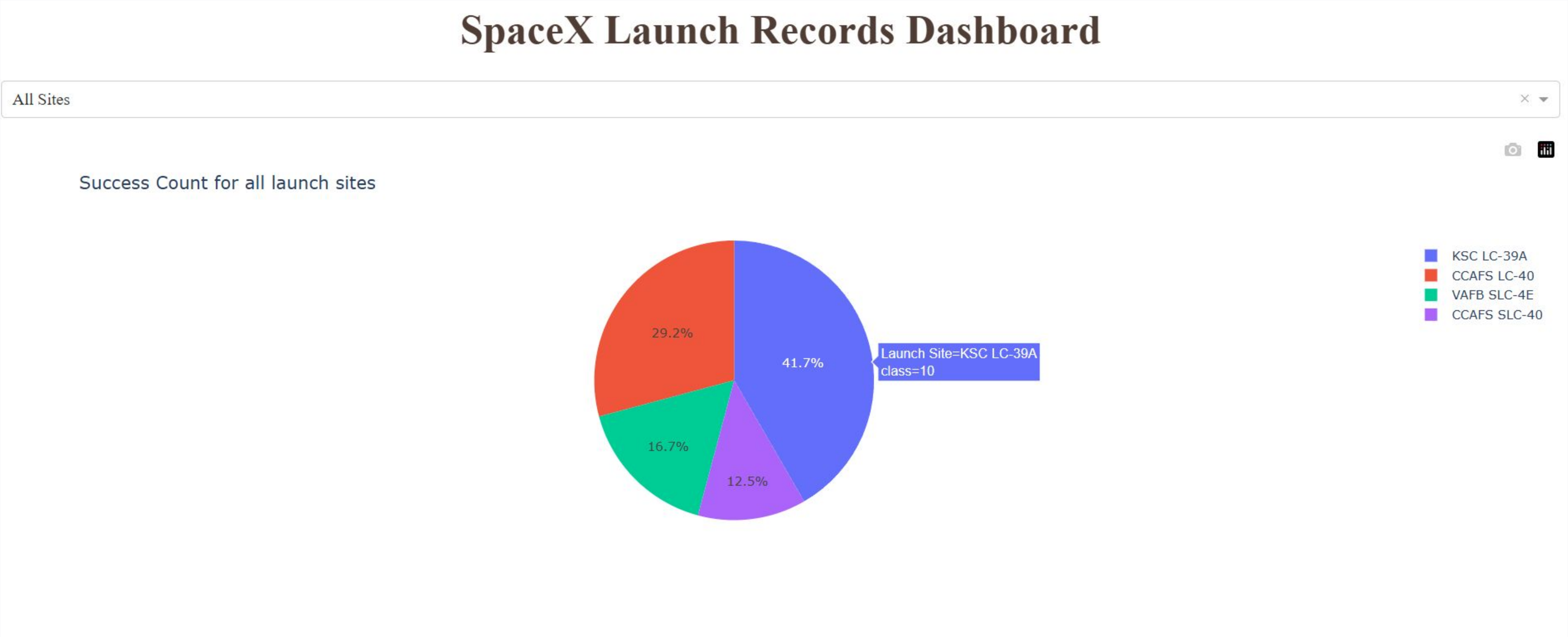
```
launch_site_coordinates = [launch_site_lat, launch_site_lon]
lines=folium.PolyLine(locations=[coast_coordinates, launch_site_coordinates], weight=1)
site_map.add_child(lines)
```



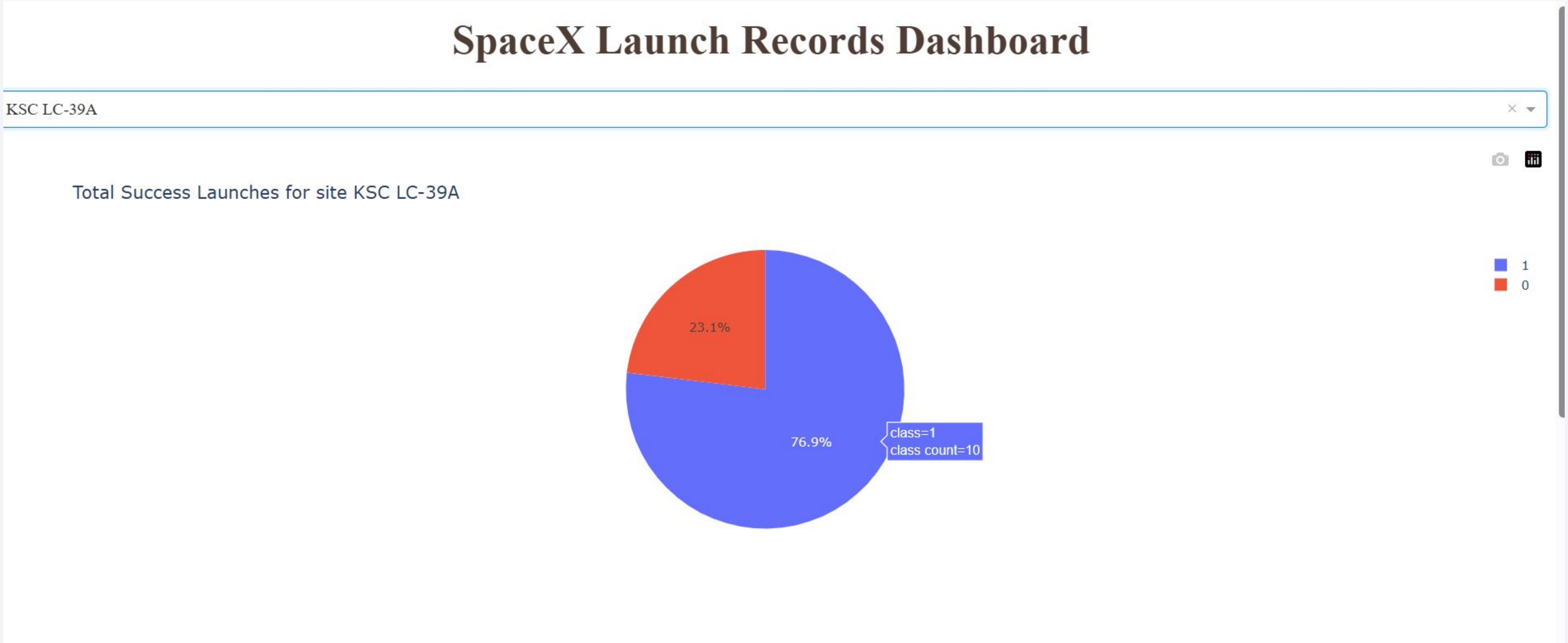
Section 4

Build a Dashboard with Plotly Dash

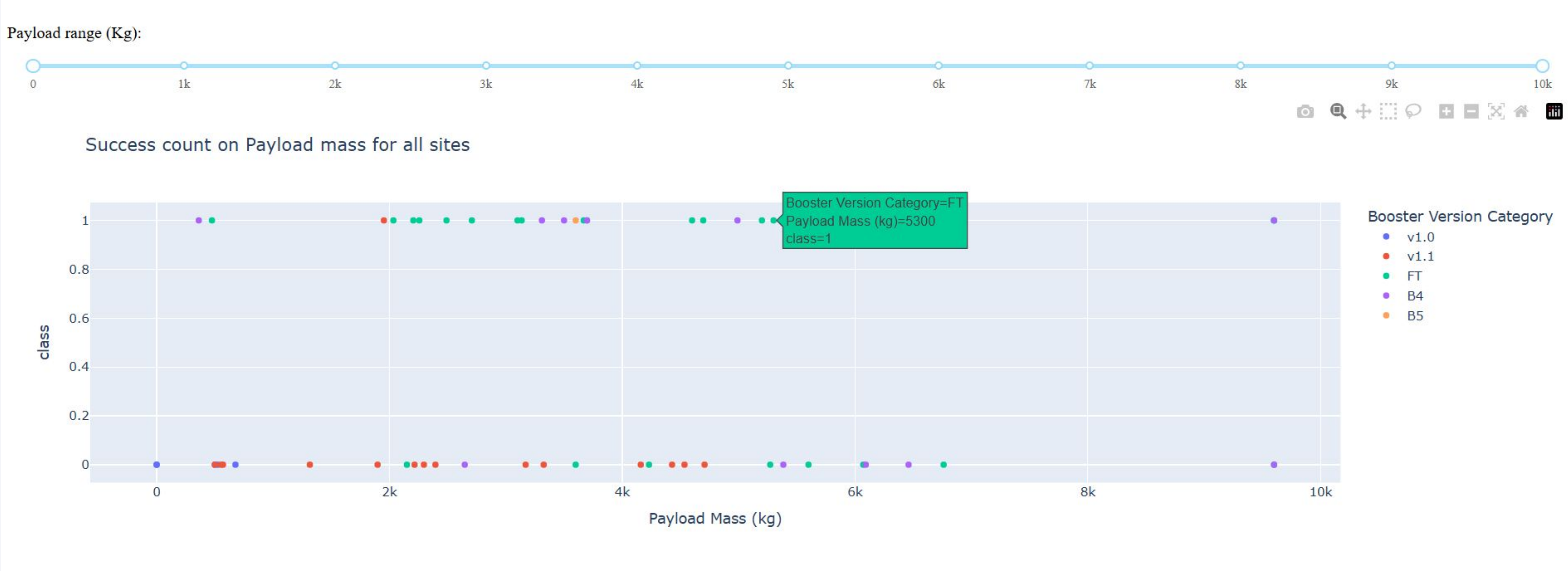
Success Count for all launch sites



Total Success launches for site KSC LC-39A

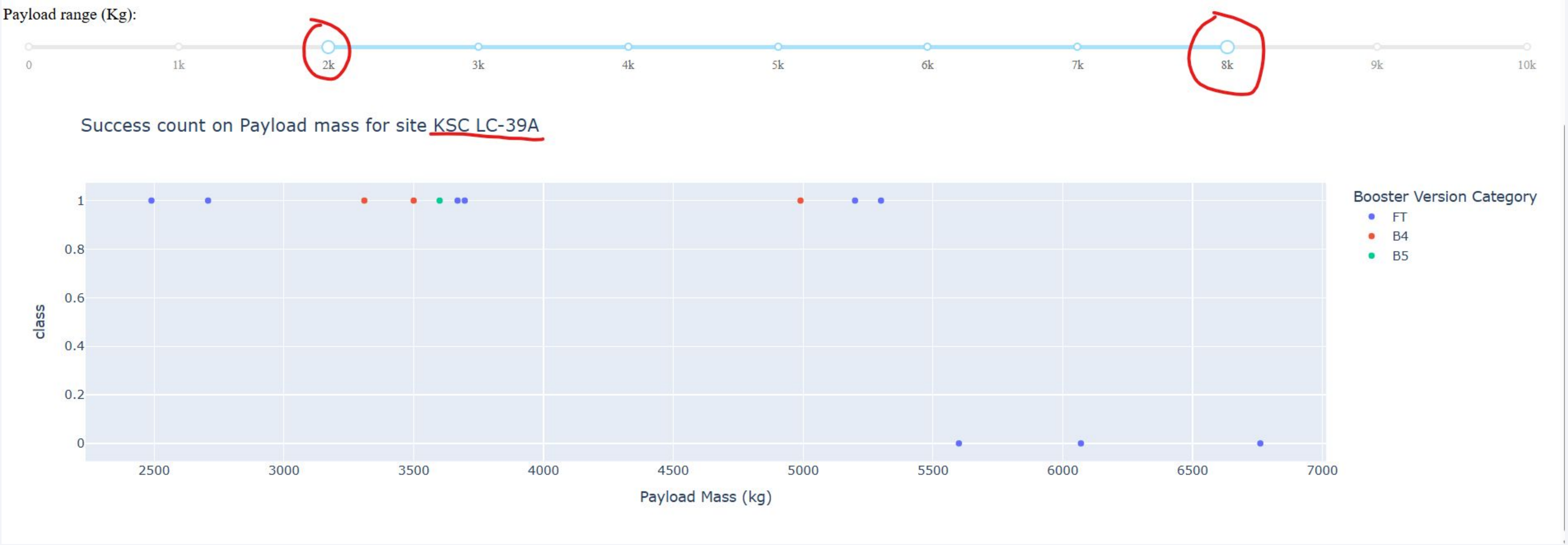


Scatter plot with PayloadMass vs Class



I can change the site from the top list and change the weight range.

Scatter plot with PayloadMass vs Class



I selected the KSC LC-39A site and the 2K-8K weight range

Section 5

Predictive Analysis (Classification)

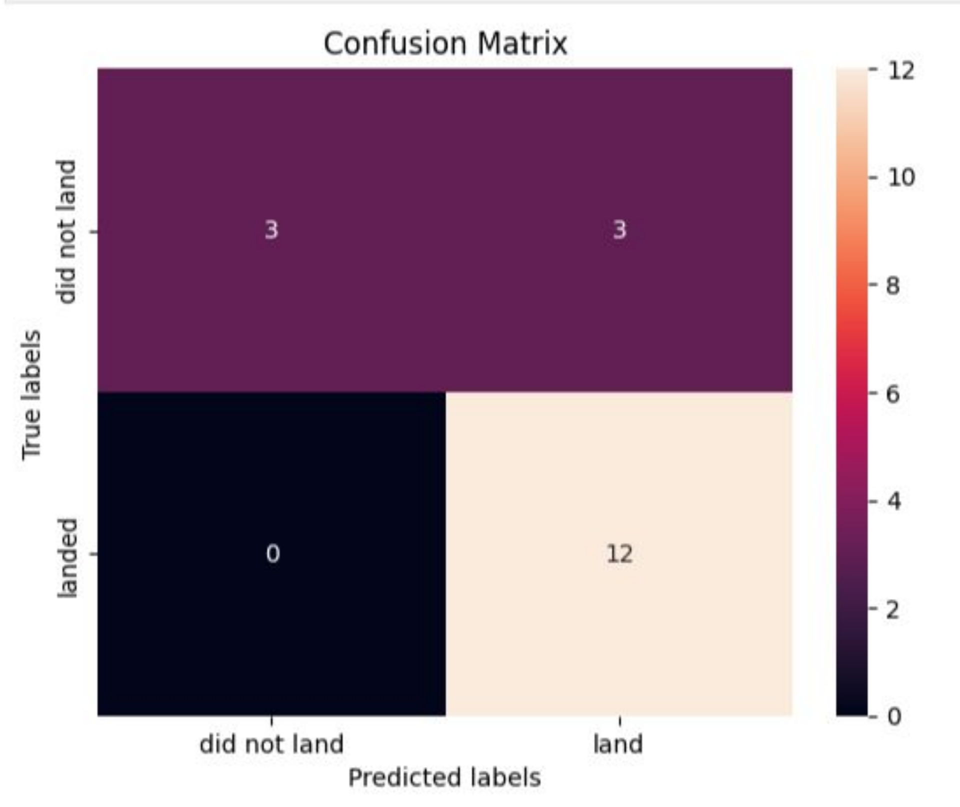
Classification Accuracy

[128]:

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

Three of the four methods perform equally well on the test data, each achieving an accuracy of 0.833333

Confusion Matrix



All four classification models produced identical confusion matrices and showed equal ability to distinguish between the classes. The main issue across all models was a high number of false positives.

Conclusions

- **Launch site performance** — KSC LC-39A achieved the highest overall success rate; some sites reached 100% after a certain number of flights.
- **Payload impact** — Payloads between 2,000–6,000 kg showed the highest probability of a successful landing; heavy payloads performed better in Polar, LEO, and ISS orbits.
- **Orbit influence** — Certain orbits (ES-L1, GEO, HEO, SSO) achieved 100% success, while others like SO and GTO had significantly lower rates.
- **Temporal trends** — Success rates steadily improved from 2013 to 2020, reflecting technological advancements and operational experience.
- **Predictive capability** — Decision Tree model achieved ~89% accuracy in predicting first stage landing success, enabling informed planning for future launches.

Thank you!