IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jose Real

February 11 2025

Analyzing SpaceX Launch Data: Uncovering Key Factors Influencing Landing Success

This presentation explores a data-driven analysis of SpaceX launch data, focusing on identifying key factors that influence successful landings. The analysis involves a systematic methodology encompassing data collection, wrangling, exploration, interactive visualization, and predictive modeling.

Outline

1 Executive Summary

A concise overview of the project, including its key objectives and methodologies.

2 Introduction

Background information on SpaceX, the importance of understanding landing success, and the project's overall goals.

3 Methodology

A detailed description of the steps involved in data collection, wrangling, analysis, visualization, and predictive modeling.

4 Data Collection

A breakdown of the data sources used, including SpaceX's API and web scraping techniques.

5 Data Wrangling

Explanation of data cleaning, transformation, and preprocessing methods to prepare the data for analysis.

6 Exploratory Data Analysis

Insights gained from examining the data using descriptive statistics, visualizations, and SQL queries.

7 Interactive Visualization

Demonstration of interactive visualizations using Folium and Plotly Dash to explore and communicate data patterns.

8 Predictive Analysis

Discussion of the predictive models used and their effectiveness in forecasting launch success.

9 Results

Presentation of the key findings and insights derived from the analysis.

10 Conclusions

Summary of the project's key takeaways, limitations, and potential future research directions.

Executive Summary

Objective

This project aims to analyze SpaceX launch data to identify factors influencing successful landings, using a comprehensive methodology encompassing data collection, wrangling, exploration, visualization, and predictive modeling.

Methods

The methodology involves data collection through the SpaceX API and web scraping, followed by data wrangling, exploratory data analysis using SQL and visualizations, interactive visual analytics with Folium and Plotly Dash, and predictive modeling using classification techniques to forecast launch outcomes.

Introduction

SpaceX's Significance

SpaceX has revolutionized the commercial spaceflight industry with its innovative reusable launch systems and ambitious goals. Understanding the factors influencing launch success is crucial for ensuring mission success, cost-effectiveness, and the advancement of space exploration.

Project Goals

This project aims to analyze historical launch data to: identify patterns in successful and failed missions, uncover trends influencing landing success, and build predictive models to estimate the likelihood of success for future launches. These insights will contribute to understanding the complexities of spaceflight and advancing the field of aerospace engineering.

Section 1

Methodology

Methodology

1. Data Collection

Data collection is conducted using SpaceX's API to retrieve launch data and web scraping to extract additional information from public sources.

2. Data Wrangling

Data is cleaned, transformed, and preprocessed to address missing values, inconsistencies, and ensure data quality for analysis.

3. Exploratory Data Analysis

Descriptive statistics, visualizations, and SQL queries are employed to understand the characteristics and relationships within the data.

4. Interactive Visualization

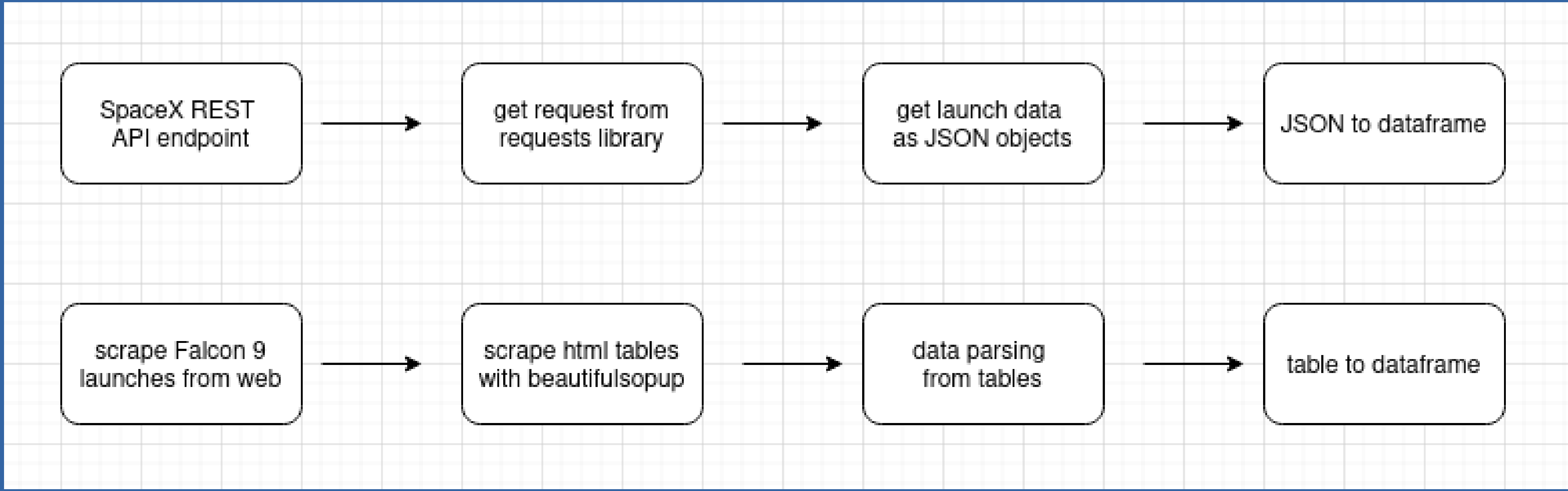
Folium and Plotly Dash are used to create interactive maps and dashboards, enabling dynamic exploration of the data and presentation of key insights.

5. Predictive Analysis

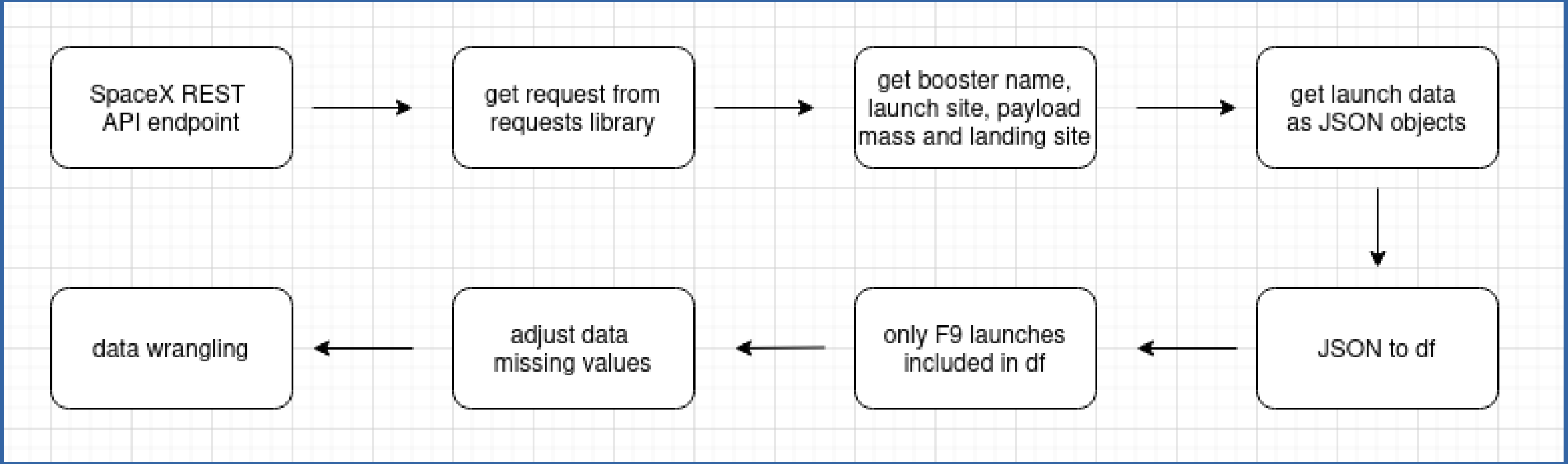
Classification models are trained on the prepared data to predict the likelihood of successful landings, based on identified factors.

Data Collection

Data from SpaceX REST API and scraped from Wikipedia

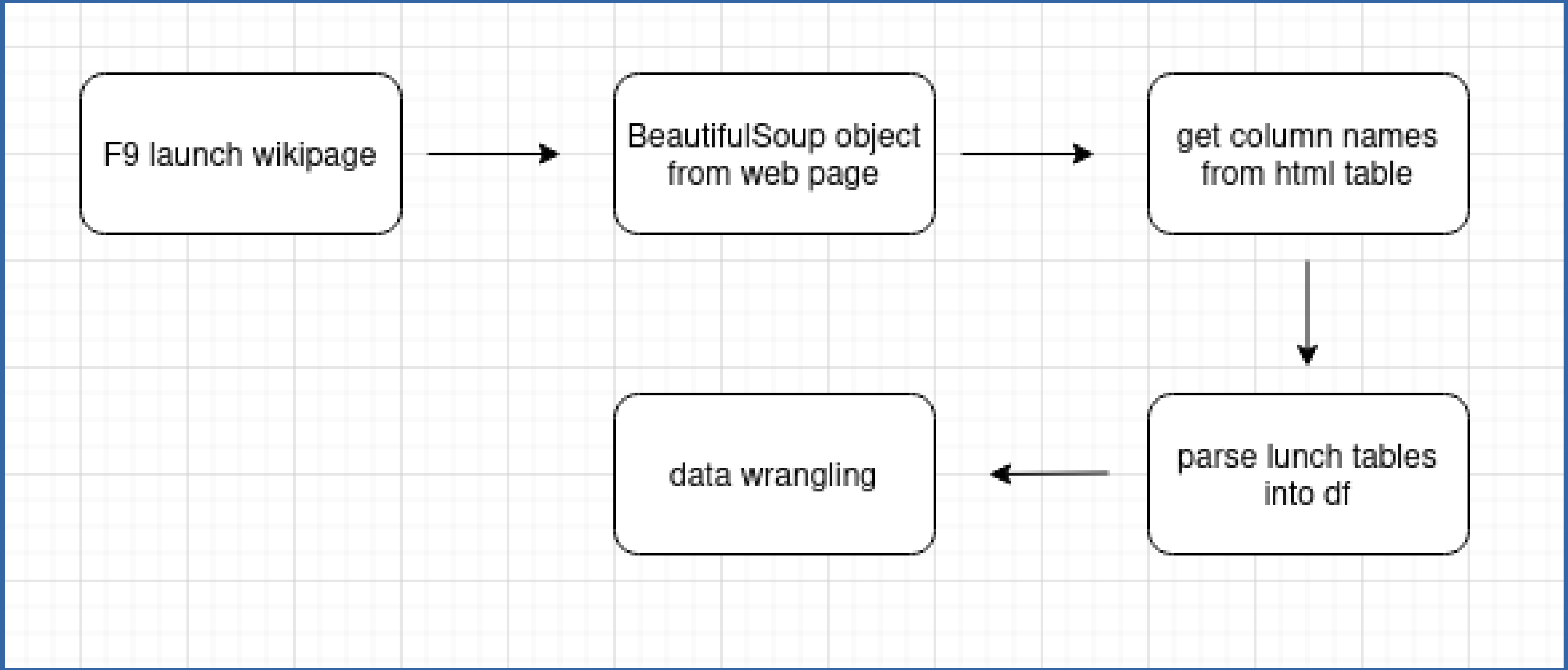


Data Collection – SpaceX API



http://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/1a-spacex-data-collection-api.ipynb

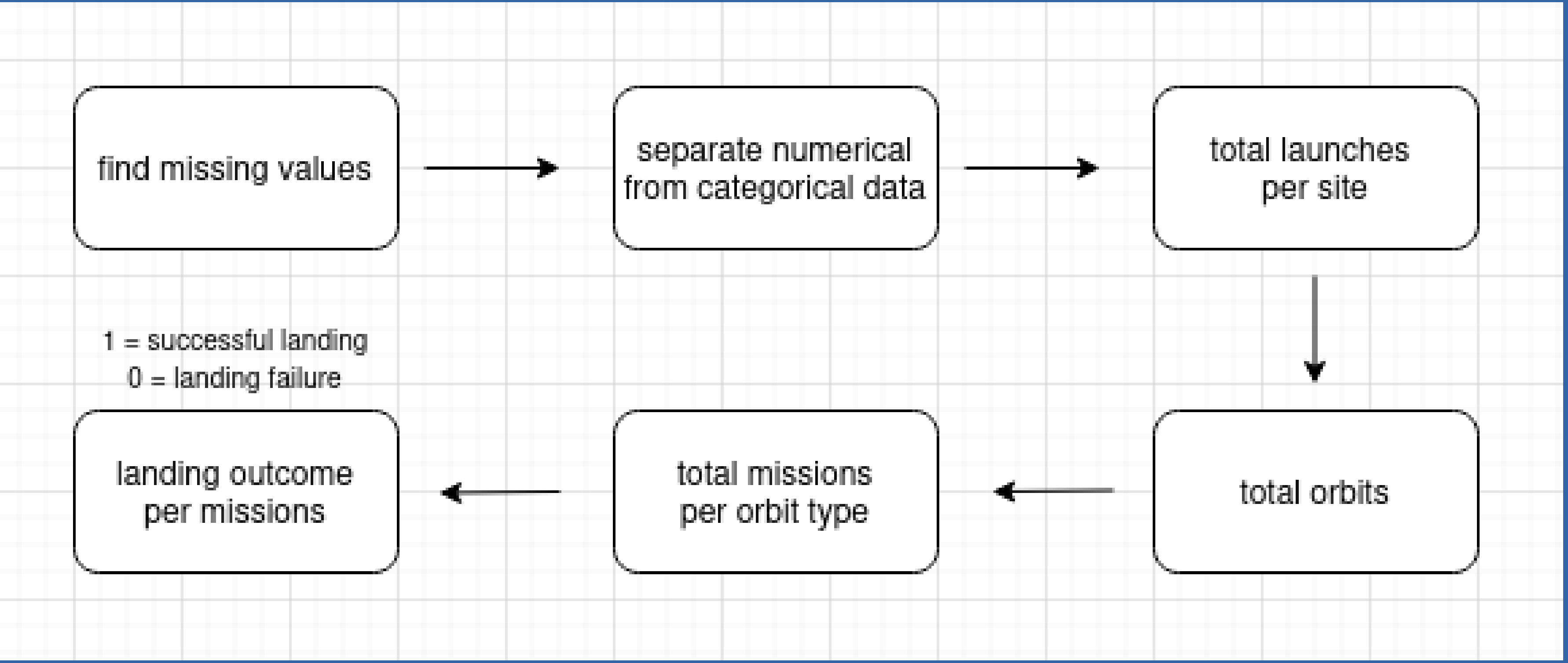
Data Collection - Scraping



https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/1b-spacex-data-collection-webscraping.ipynb

Data Wrangling

Exploratory Data Analysis (EDA) enables us to identify patterns in the data later to be used in supervised model training



https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/1c-spacex-data-wrangling.ipynb

EDA with Data Visualization

- flight number vs payload mass catplot
- flight number vs launch site catplot
- payload mass vs launch site catplot
- success rate by orbit type barplot
- number of lights vs orbit type scatterplot
- payload mass vs orbit type scatterplot
- launch success yearly trend lineplot

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/2b-spacex-eda-dataviz.ipynb

EDA with SQL (1 of 4)

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

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

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

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

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

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

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

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

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/2a-spacex-eda-sql.ipynb

EDA with SQL (2 of 4)

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

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

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

```
- List the total number of successful and failure mission outcomes
```

```
%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY
"Mission_Outcome";
```

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

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

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/2a-spacex-eda-sql.ipynb

EDA with SQL (3 of 4)

- 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.

```
%%sql
SELECT
    CASE
        WHEN substr("Date", 6, 2) = '01' THEN 'January'
        WHEN substr("Date", 6, 2) = '02' THEN 'February'
        WHEN substr("Date", 6, 2) = '03' THEN 'March'
        WHEN substr("Date", 6, 2) = '04' THEN 'April'
        WHEN substr("Date", 6, 2) = '05' THEN 'May'
        WHEN substr("Date", 6, 2) = '06' THEN 'June'
        WHEN substr("Date", 6, 2) = '07' THEN 'July'
        WHEN substr("Date", 6, 2) = '08' THEN 'August'
        WHEN substr("Date", 6, 2) = '09' THEN 'September'
        WHEN substr("Date", 6, 2) = '10' THEN 'October'
        WHEN substr("Date", 6, 2) = '11' THEN 'November'
        WHEN substr("Date", 6, 2) = '12' THEN 'December'
    END AS Month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Failure (drone ship)'
AND substr("Date", 0, 5) = '2015';
```

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/2a-spacex-eda-sql.ipynb

EDA with SQL (4 of 4)

- 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.

```
%%sql
SELECT
    "Landing_Outcome",
    COUNT(*) AS Outcome_Count
FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
```

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/2a-spacex-eda-sql.ipynb

Build an Interactive Map with Folium

- folium.Circle and folium.Marker add circle area with label on coordinates for launch sites on map

- MarkerCluster object simplifies visualization of markers on same site

- MousePosition highlights coordinates of mouse pointer onmap

- folium.PolyLine objects draws a line between launch site and nearby city, railway and highway

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/3a-spacex-interactive-viz-folium.ipynb

Build a Dashboard with Plotly Dash

Dashboard app has a drop down list and a range slider allowing for interactive input on a pie chart and a scatter point chart.

- drop down input: allows to select from one of 4 launch sites or all sites

- range slider: adjust payload range from 0 to 10000 kg

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/3b-spacex-interactive-dashboard-ploty.ipynb

Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels
- create a column for the class
- Standardize the data
- Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

https://github.com/ppreal/IBM-DS-Cert-Capstone/blob/main/4-spacex-machine-learning-prediction.ipynb

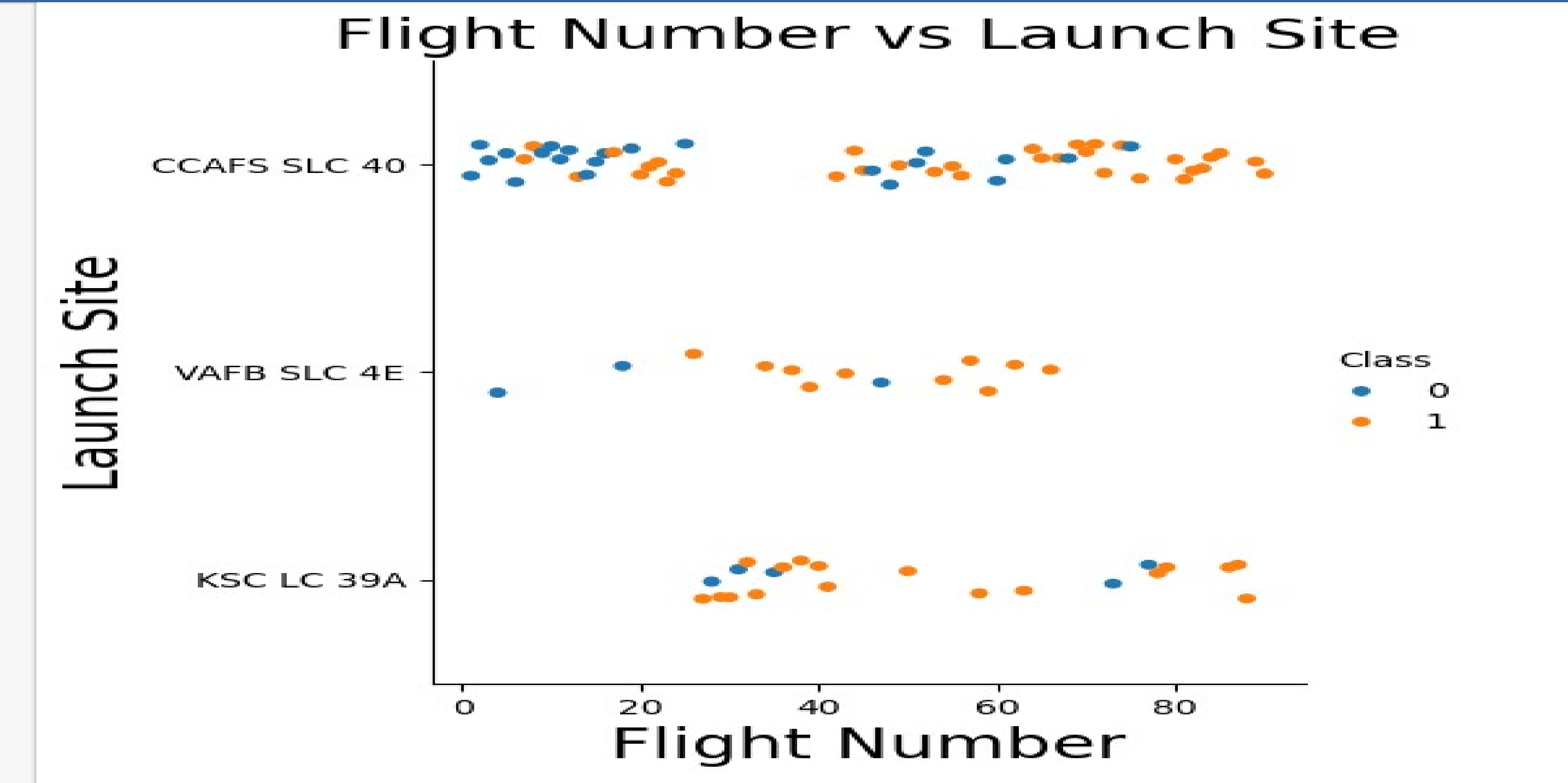
Results

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

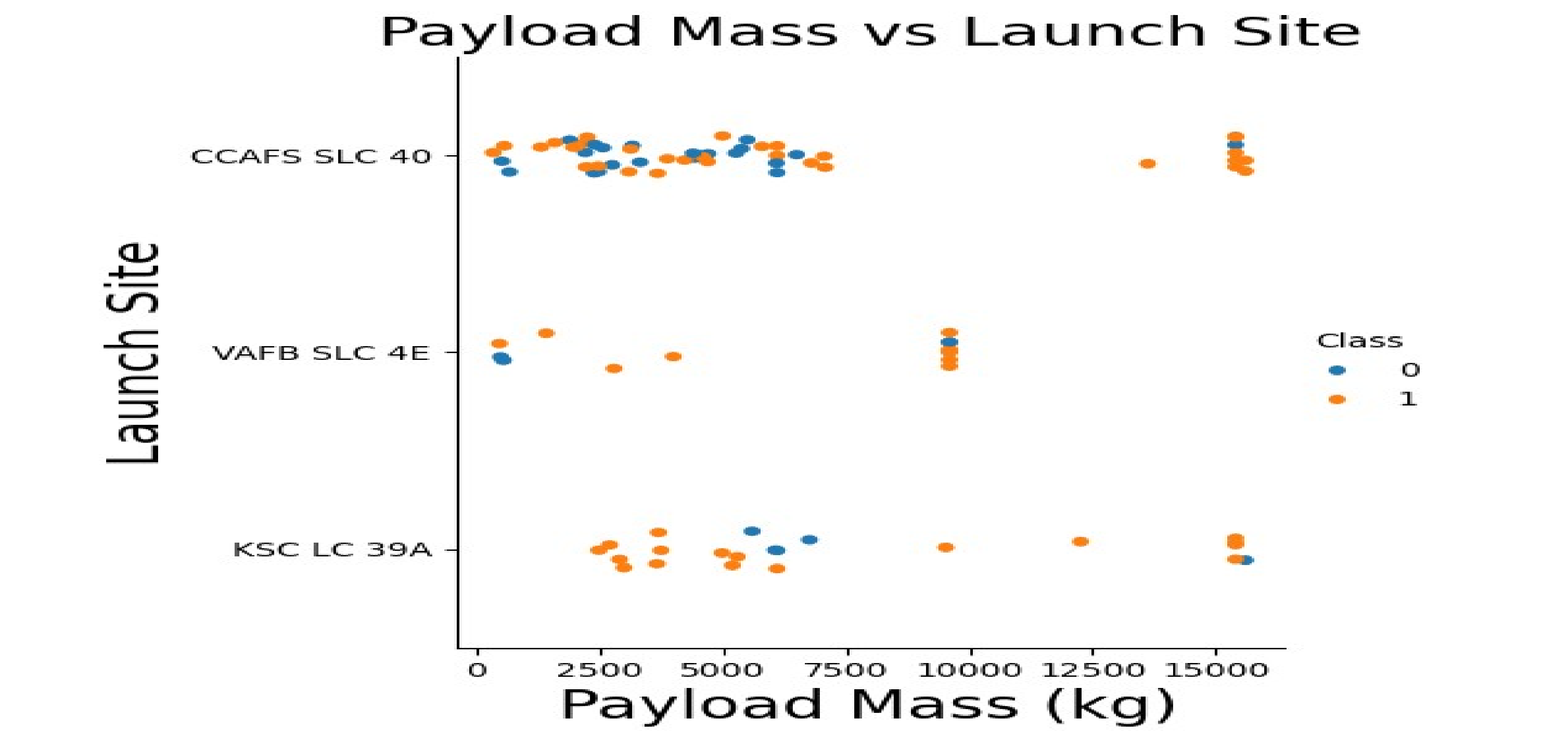
Section 2

Insights drawn from EDA

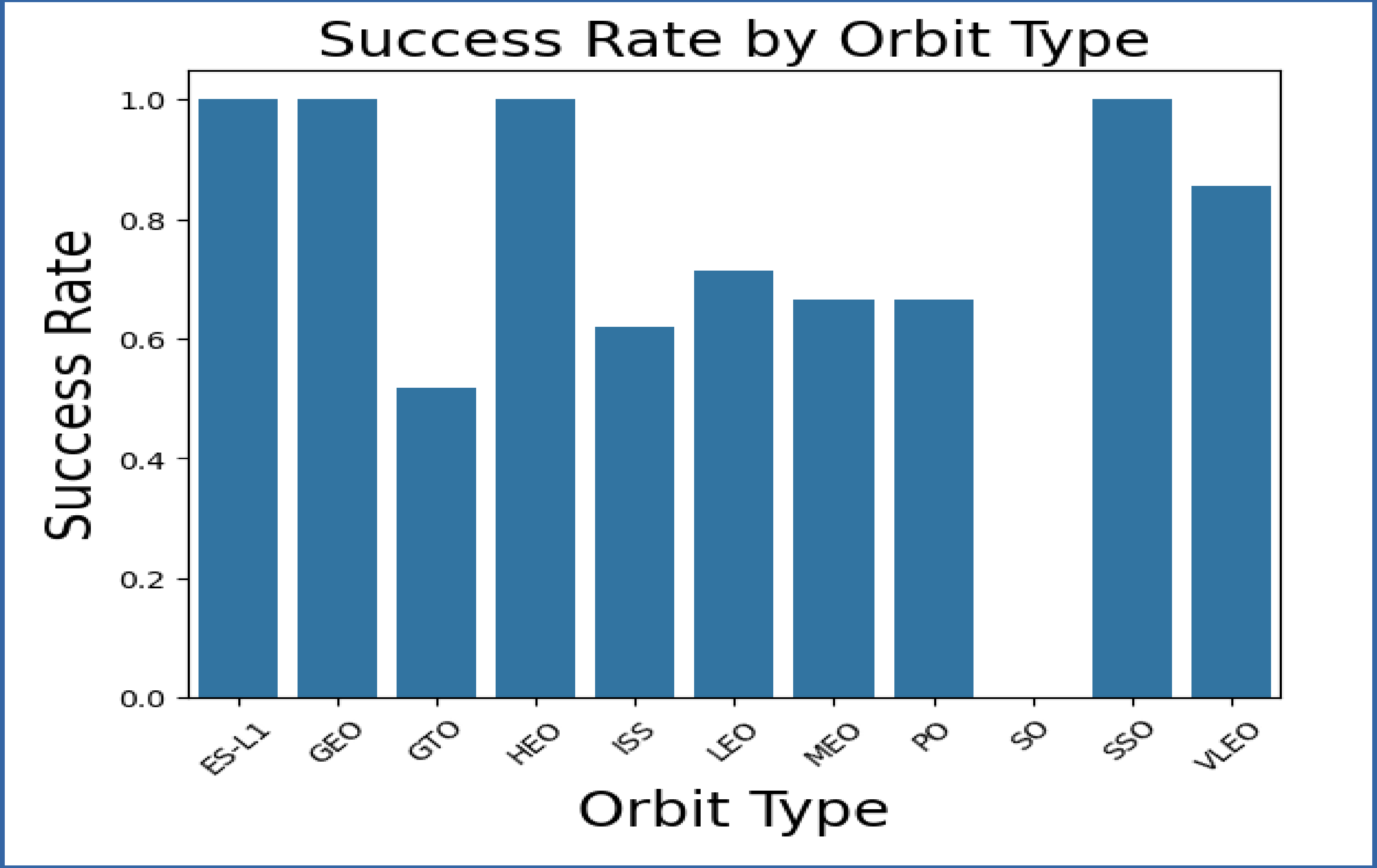
Flight Number vs. Launch Site



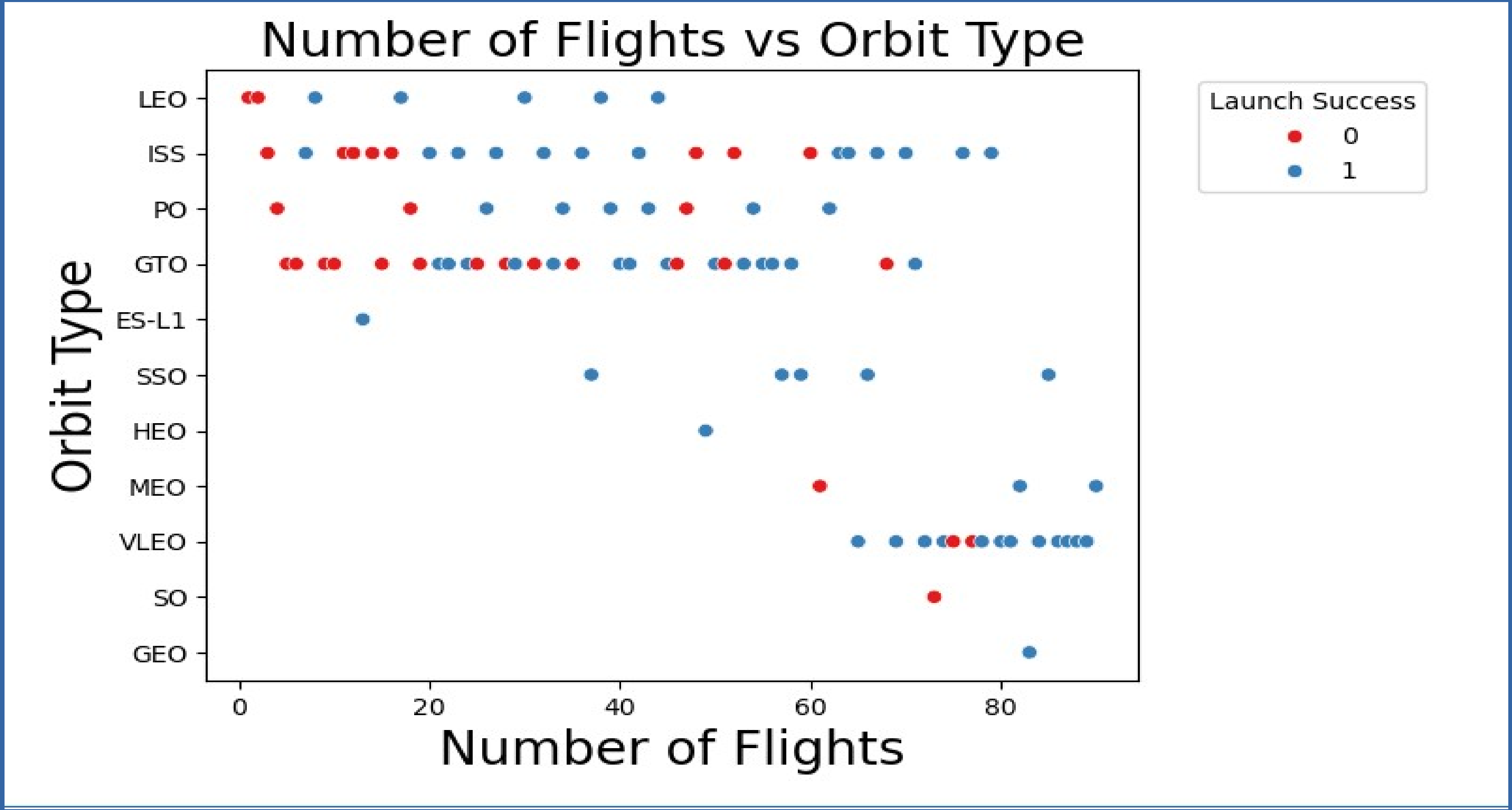
Payload vs. Launch Site



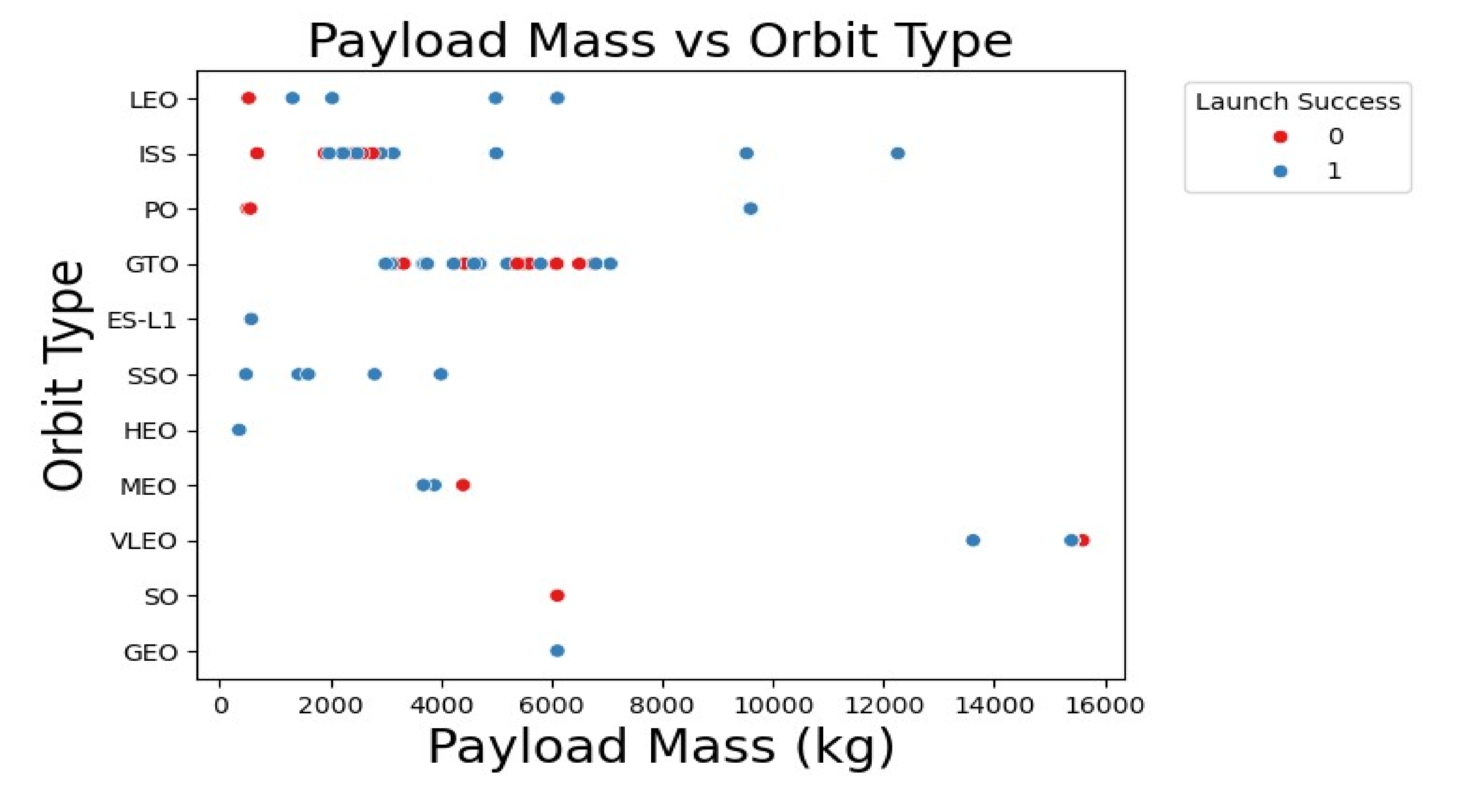
Success Rate vs. Orbit Type



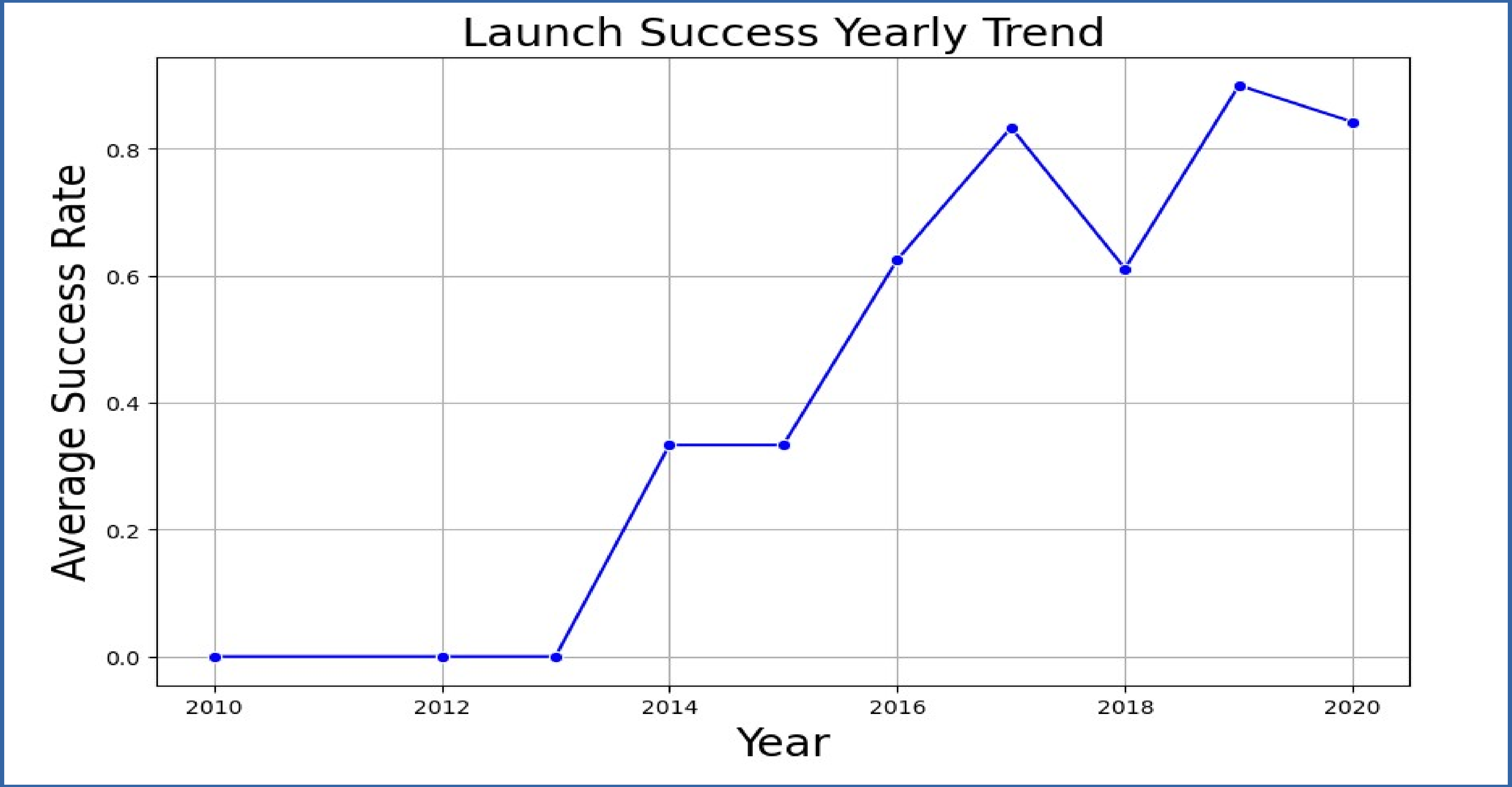
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

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

```
%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'

Display 5 records where launch sites begin with the string '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

Total Payload Mass

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

```
%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

Display average payload mass carried by booster version 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

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

Hint:Use min function

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

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

First_Successful_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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 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

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

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

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

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

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 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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%%sql
SELECT
    CASE
        WHEN substr("Date", 6, 2) = '01' THEN 'January'
        WHEN substr("Date", 6, 2) = '02' THEN 'February'
        WHEN substr("Date", 6, 2) = '03' THEN 'March'
        WHEN substr("Date", 6, 2) = '04' THEN 'April'
        WHEN substr("Date", 6, 2) = '05' THEN 'May'
        WHEN substr("Date", 6, 2) = '06' THEN 'June'
        WHEN substr("Date", 6, 2) = '07' THEN 'July'
        WHEN substr("Date", 6, 2) = '08' THEN 'August'
        WHEN substr("Date", 6, 2) = '09' THEN 'September'
        WHEN substr("Date", 6, 2) = '10' THEN 'October'
        WHEN substr("Date", 6, 2) = '11' THEN 'November'
        WHEN substr("Date", 6, 2) = '12' THEN 'December'
    END AS Month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Failure (drone ship)'
AND substr("Date", 0, 5) = '2015';
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	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.

```
%%sql
SELECT
    "Landing_Outcome",
    COUNT(*) AS Outcome_Count
FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
```

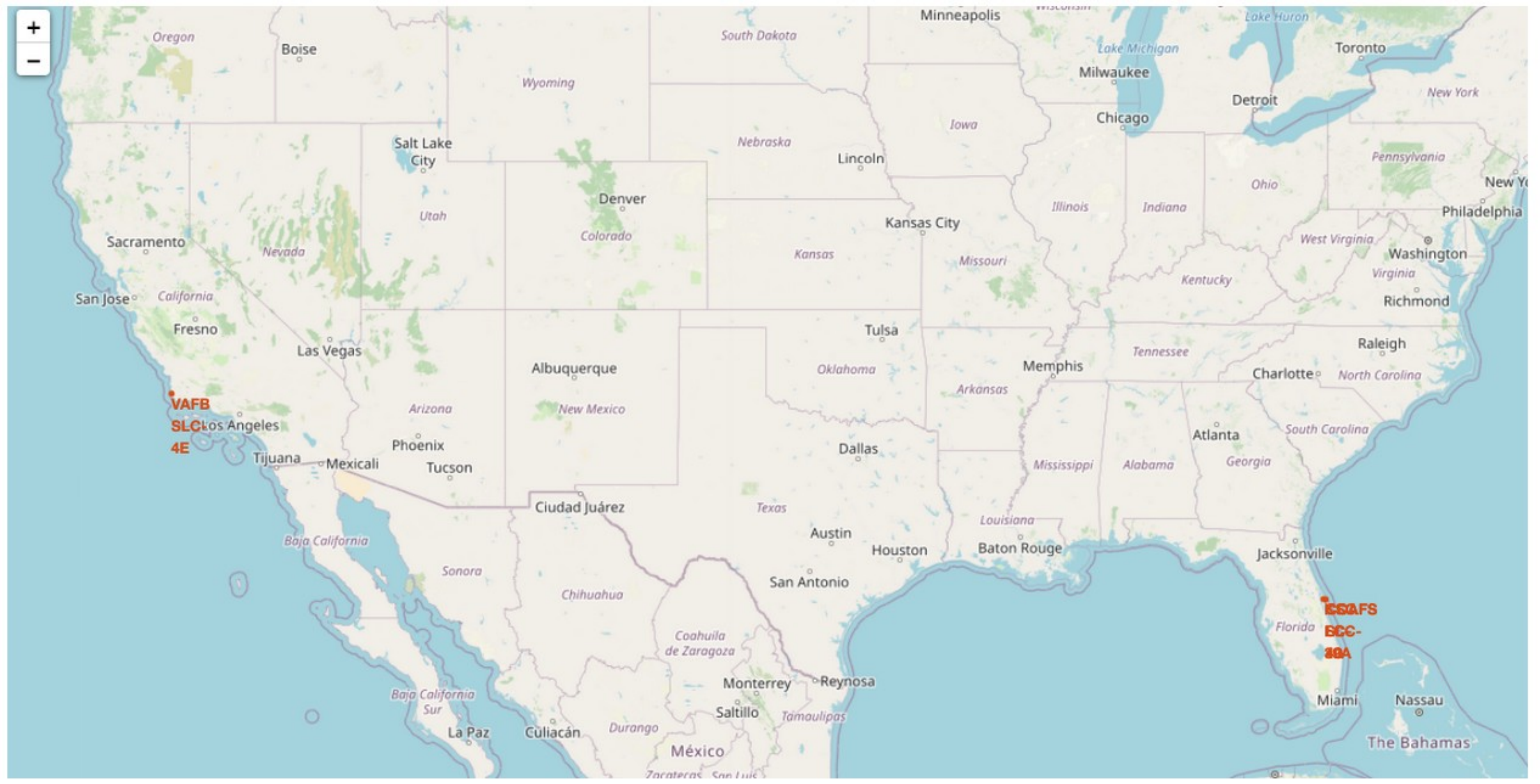
```
 * sqlite:///my_data1.db
Done.
```

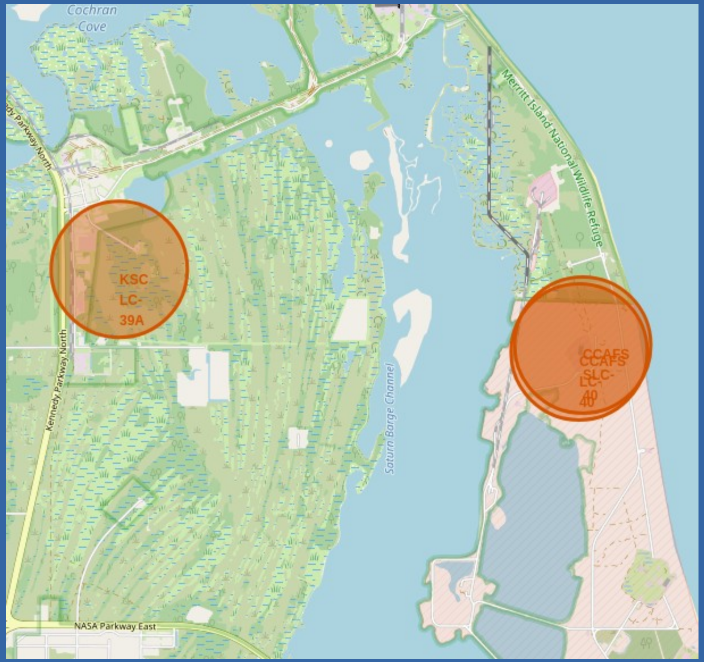
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

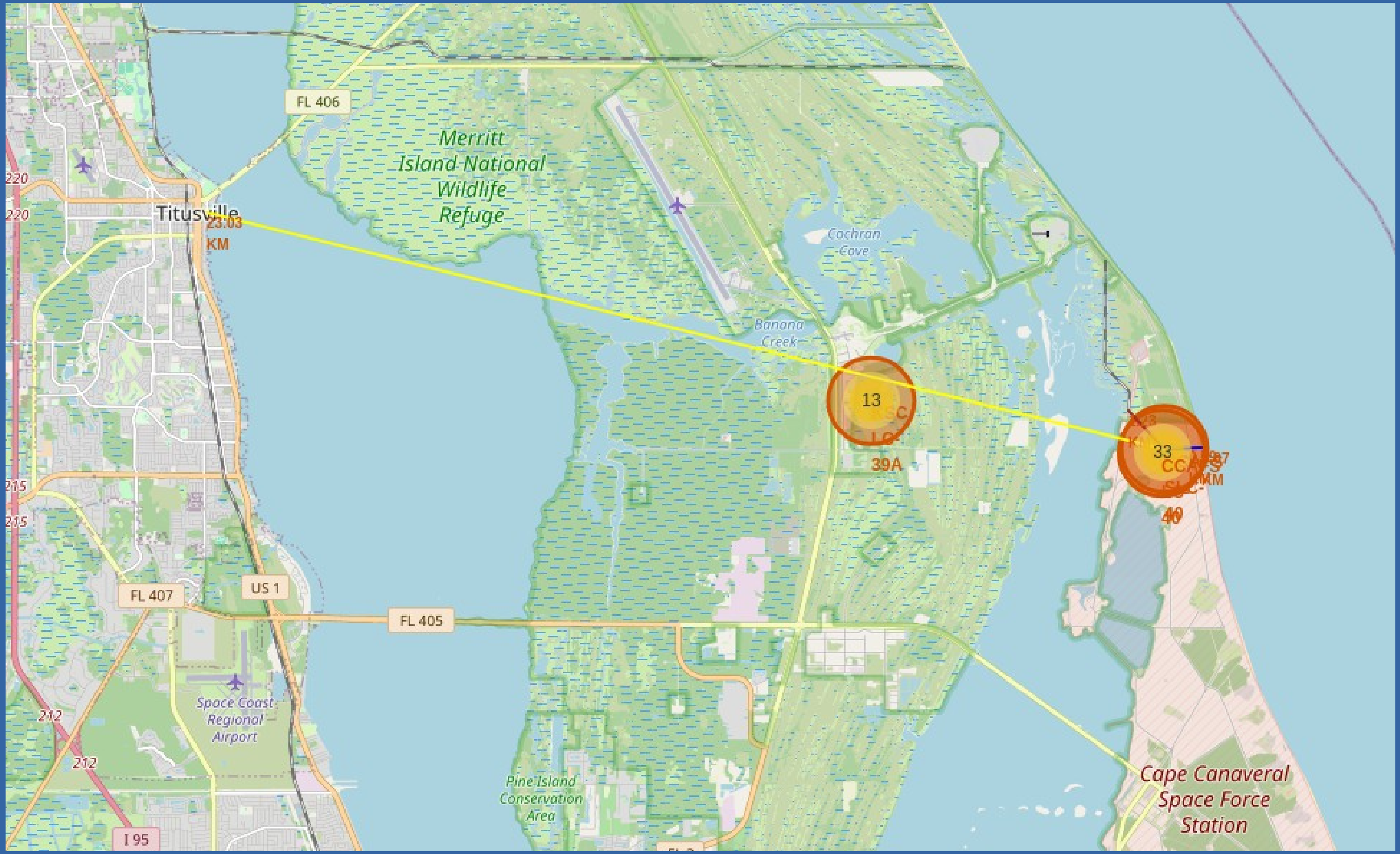
All launch sites

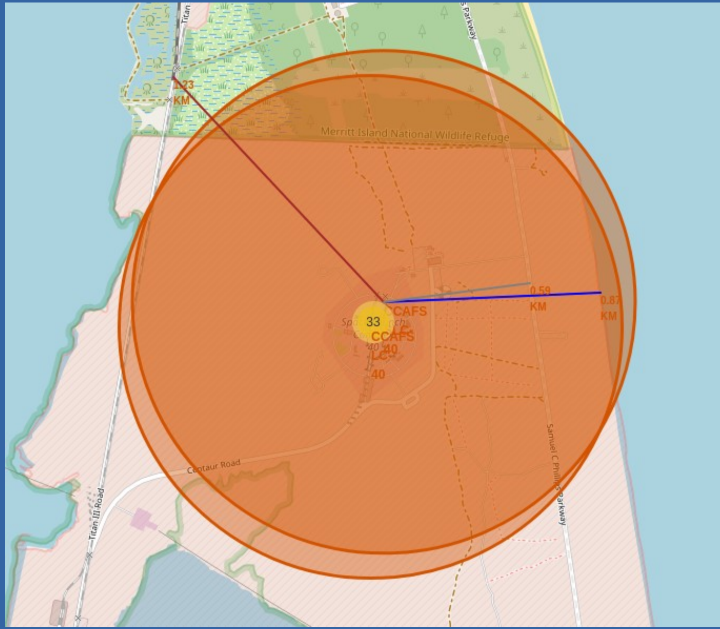




Launch outcomes

Launch site proximities

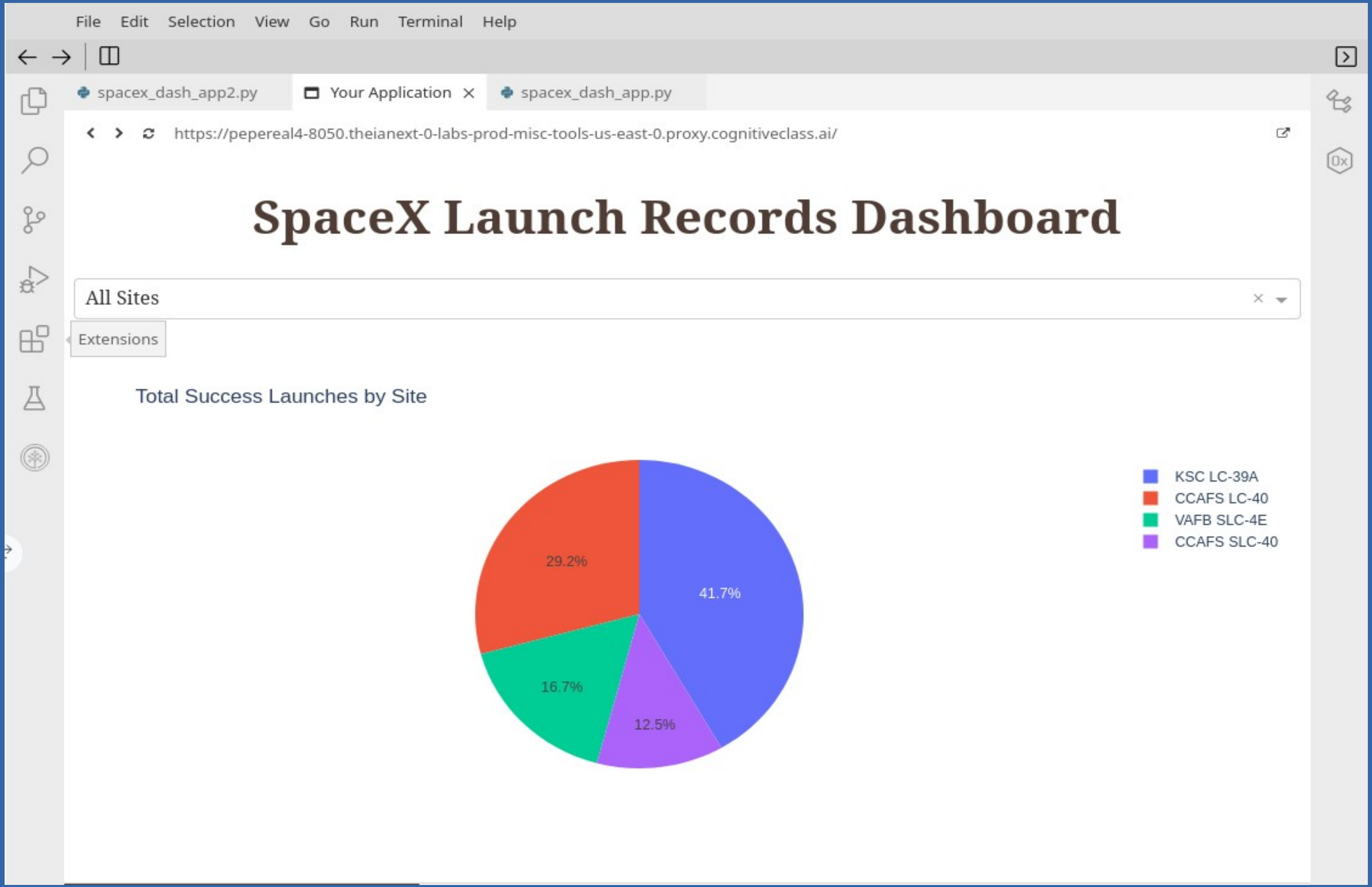




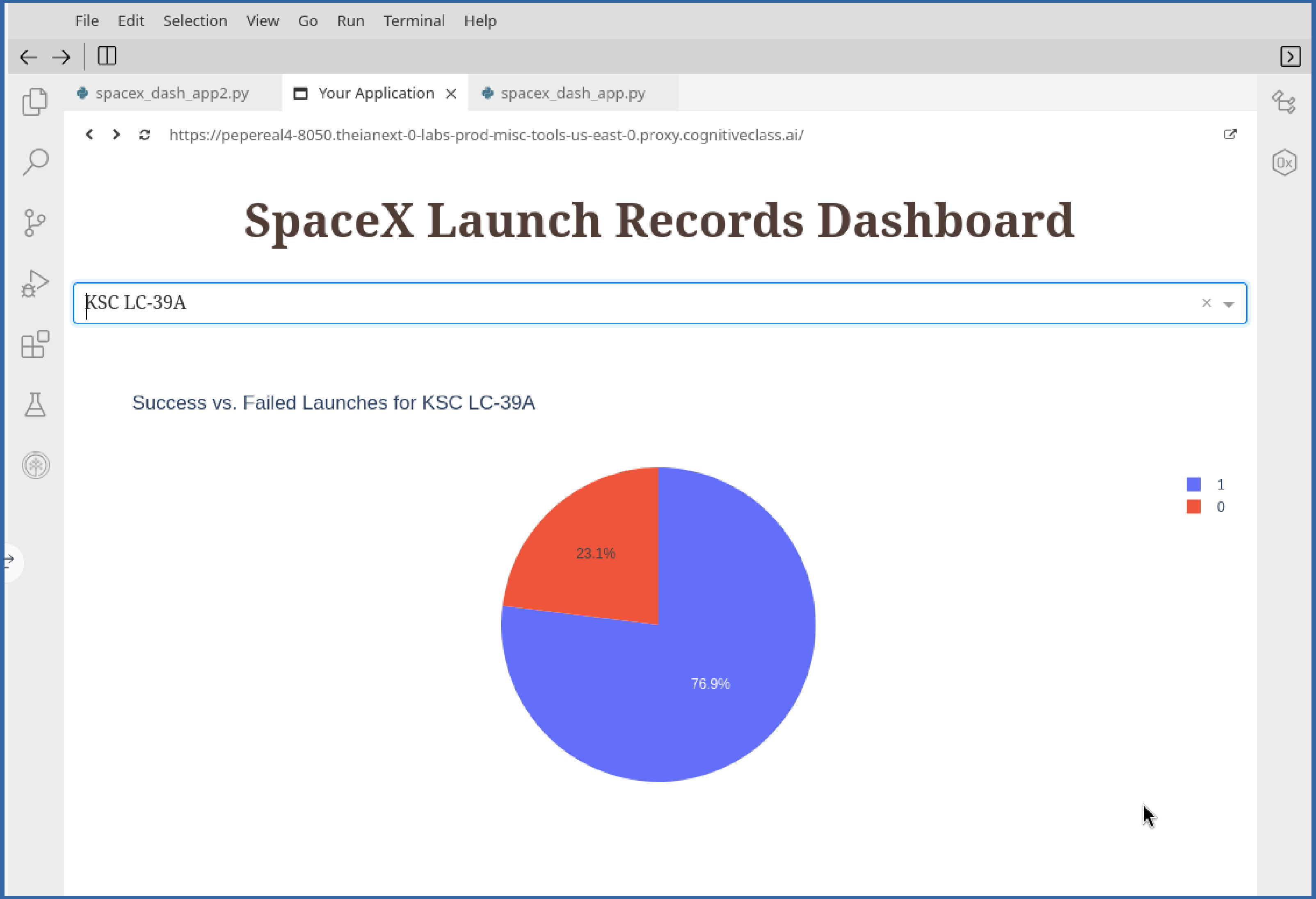
Section 4

Build a Dashboard with Plotly Dash

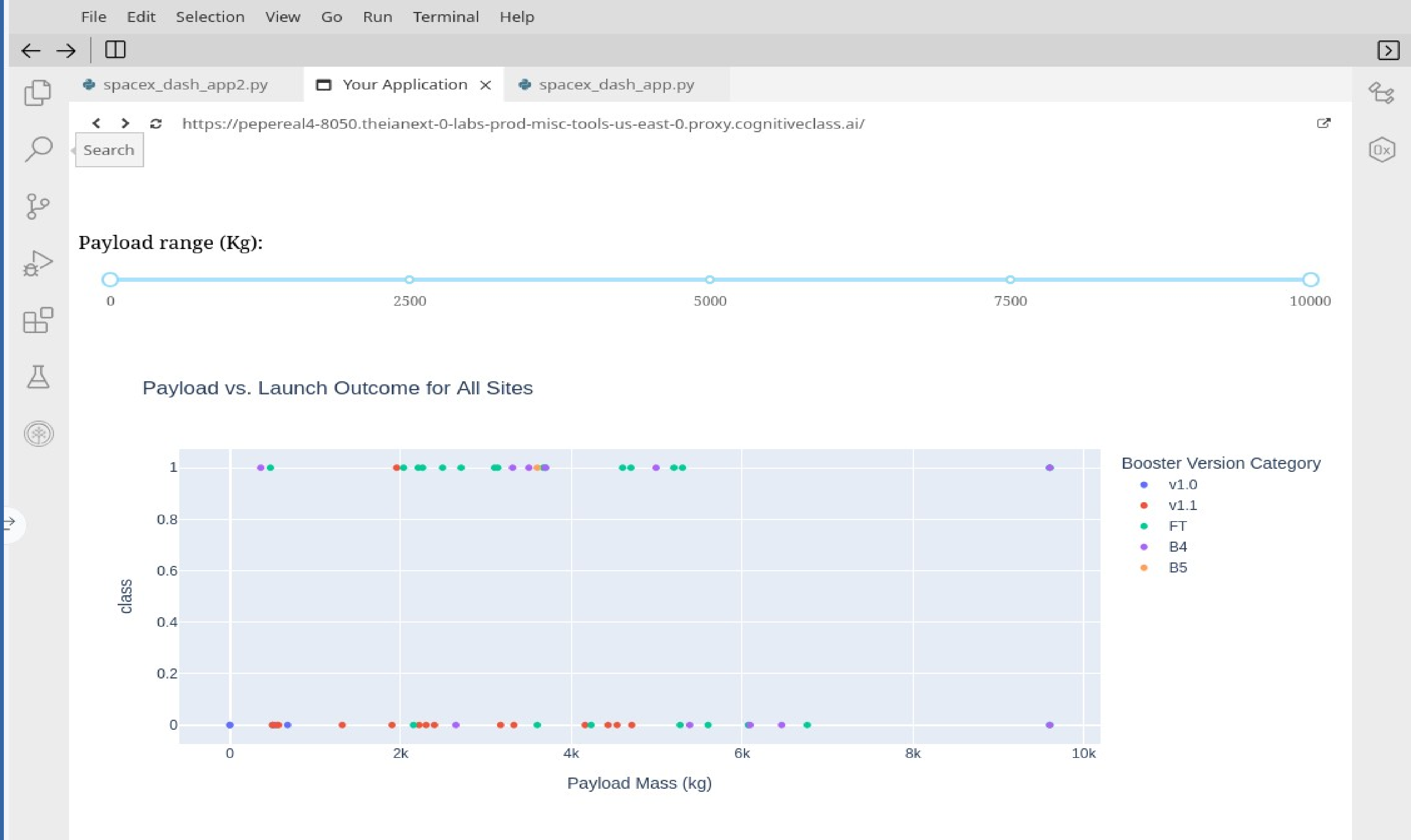
Total success launches by site



KSC LC-39A


File Edit Selection View Go Run Terminal Help
spacex_dash_app2.py
Your Application
spacex_dash_app.py
https://pepereal4-8050.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/
SpaceX Launch Records Dashboard
KSC LC-39A
Success vs. Failed Launches for KSC LC-39A
23.1%
76.9%
1
0

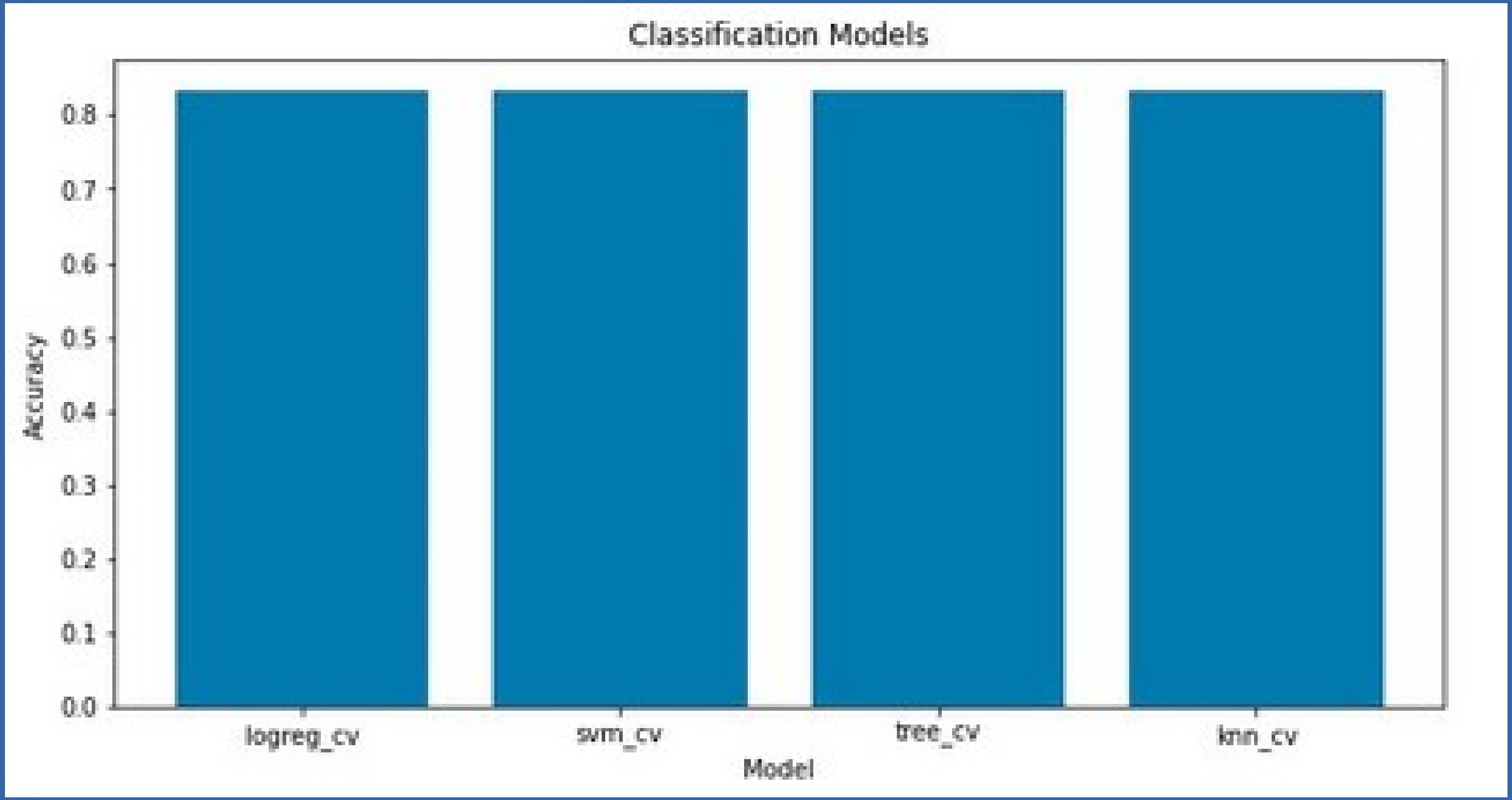
Payload vs launch outcome



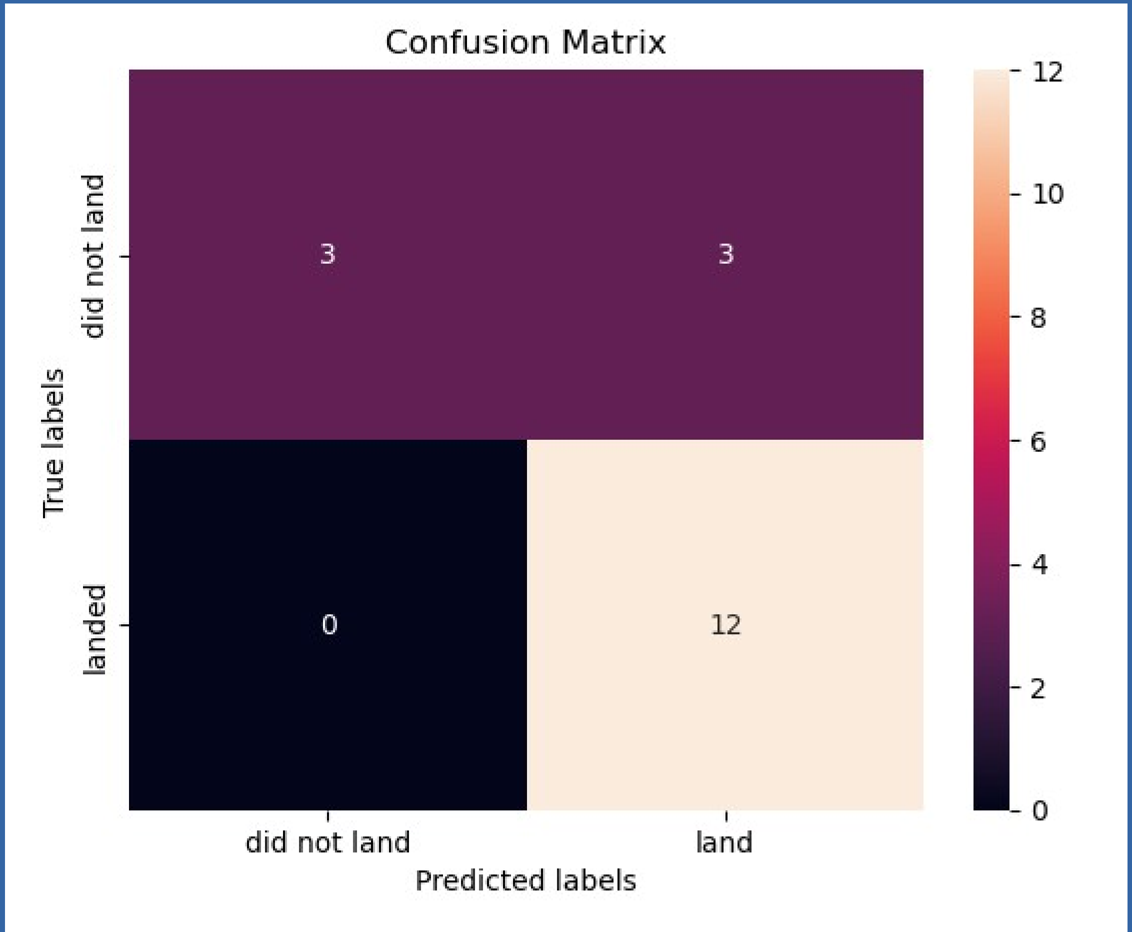
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Same result for all models

Conclusions

Model Performance: Logistic Regression, SVM, and KNN achieved similar cross-validation (~84.6%) and test accuracies (83.33%), while Decision Tree overfitted with the highest cross-validation accuracy (87.68%) but lower test accuracy (77.78%).

Best Model: Logistic Regression, SVM, and KNN were the most consistent, generalizing better than the Decision Tree.

Confusion Matrix Insights: All models struggled with false positives, indicating challenges in distinguishing unsuccessful landings.

Final Thoughts: The lab demonstrated the full machine learning pipeline, showing ML's potential in predicting first-stage landing outcomes for cost estimation.

Thank you!