IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

John Liang
04-22-2025

Outline

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

Executive Summary

- **Objective**: Predict Falcon 9 first-stage landing success using data-driven methods.
- **Methodologies**:
  - Data gathered from SpaceX API & Wikipedia, cleaned & merged.
  - EDA performed with Pandas, Seaborn, SQL (e.g., site/orbit success, payload trends).
  - Folium used to map proximity to railways, highways, coastlines, and cities.
  - Interactive dashboard built with Plotly Dash (site & payload filters).
  - Classification models (LogReg, SVM, Tree, KNN) trained with GridSearchCV.
- **Key Results**:
  - CCAFS SLC-40 = most launches; KSC LC-39A = highest success consistency.
  - Success linked to payload range & orbit type (ISS, LEO).
  - Sites within 1–3 km of transport infrastructure, >15 km from cities.
  - Decision Tree achieved best CV accuracy (87.5%), with stable test accuracy (83.3%) across all models.

Introduction

- **Project Background & Context**:
- SpaceX is a pioneer in reusable rocket technology, with **Falcon 9** designed to reduce launch costs via successful first-stage landings.
- While some boosters land successfully, others fail due to mission complexity, payload weight, or trajectory.
- Predicting landing success is vital for **cost forecasting**, **mission planning**, and **risk mitigation**.
- **Problems to Explore**:
- Which **features** (e.g., orbit, payload, site) are most predictive of landing success?
- Are launch sites strategically located near infrastructure?
- Can a **machine learning model** accurately predict landing success?
- What launch patterns or insights can improve future decision-making?

Section 1

Methodology

Methodology

Executive Summary

- **Data Collection**:
- Launch data gathered from **SpaceX REST API** and **Wikipedia tables**.
- Supplemented with booster, payload, and launchpad metadata via API endpoints.
- **Data Wrangling**:
- Merged datasets using unique IDs.
- Cleaned nulls, filtered for **Falcon 9** missions, and imputed missing payloads.
- **EDA (Exploratory Data Analysis)**:
- Used **Pandas, Seaborn, and SQL** to explore:
    - Launch site performance
    - Orbit success patterns
    - Payload vs. success rate
    - **Interactive Visual Analytics**:

Methodology

Executive Summary (continued)

- Created **Folium map** to analyze proximity to coast, cities, highways, rail.
- Built **Plotly Dash dashboard** for live exploration of site, payload, and success.
- **Predictive Modeling**:
- Used **StandardScaler + train_test_split** to preprocess data.
- Trained and tuned **LogReg, SVM, Decision Tree, KNN** using GridSearchCV.
- **Evaluation**:
- Compared models using **accuracy, confusion matrix**, and **classification report**.
- Selected best model based on **cross-validation and test performance**.

Data Collection

- **Overview**
- Data for this project was collected from **multiple sources** to provide a comprehensive view of Falcon 9 launches and landing outcomes.
- **Key Sources:**
- **SpaceX REST API**
  - `launches/past`, rockets, `payloads`, `cores`, `launchpads`
  - Provided raw launch records and related metadata
- **Wikipedia Launch History Tables**
  - Web scraped Falcon 9 mission table using **BeautifulSoup**
- **Manual Feature Engineering**
  - Created additional features (e.g., `Outcome`, `GridFins`, `Reused`, `Legs`)
  - Derived labels for landing success (class = 1 or 0)

Data Collection – SpaceX API

- **Key Concepts & Phrases**
- **RESTful API** provided by SpaceX at https://api.spacexdata.com/v4
- Retrieved structured launch records and metadata using **GET requests**
- Used **Python requests library** to pull JSON data
- Converted JSON responses into **Pandas DataFrames**
- Used **.json_normalize()** to flatten nested fields
- https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/a3cf7421c1979bee856a73e7295a503744134f65/jupyter-labs-spacex-data-collection-api%20m1%20final.ipynb

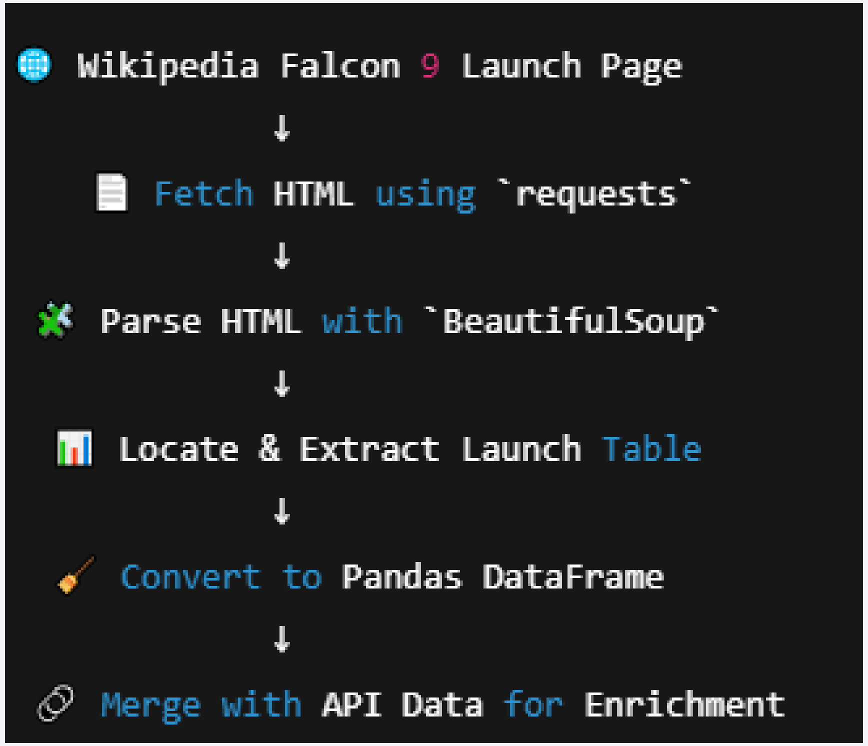
Flow chart

```
SpaceX API (v4)
      ↓
Endpoint: /launches/past
      ↓
JSON Response
      ↓
Data Normalization (json_normalize)
      ↓
Use IDs to call:
  ├ /rockets
  ├ /payloads
  ├ /cores
  └ /launchpads
      ↓
Merged DataFrames → Cleaned Dataset
```

Data Collection - Scraping

- **Key Phrases for Slide Bullets**
- Collected launch history table for Falcon 9 missions from **Wikipedia**
- Targeted the table titled **“Falcon 9 launch history”**
- Used **requests** to fetch the HTML content
- Used **BeautifulSoup** to parse HTML and locate launch table
- Extracted mission details:
  - Flight number, date, booster version, landing outcome,
  - etc.
- Converted HTML table to Pandas DataFrame for further cleaning
- and analysis
- Used as a **supplemental source** to fill gaps in the API data
- Github url: https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/a3cf7421c1979bee856a73e7295a503744134f65/jupyter-labs-webscraping%20m1%20final.ipynb

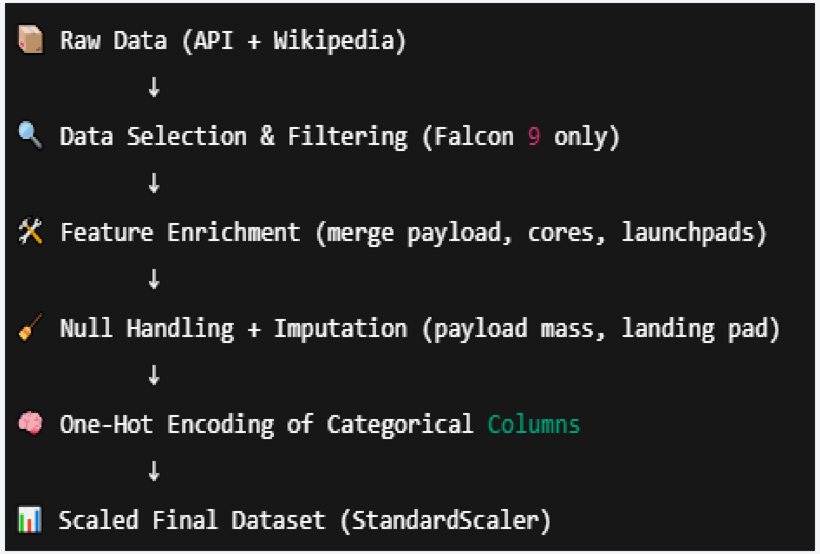
Flow chart



Data Wrangling

- Merged and processed raw data from **SpaceX API** and **Wikipedia web tables**
- Flattened nested JSON objects using json_normalize
- Mapped API identifiers (e.g., rocket ID, payload ID, core ID) to actual values
- Filtered for **Falcon 9** launches only (excluded Falcon 1 & Falcon Heavy)
- Replaced missing payload values with **mean imputation**
- Engineered new features:
  - Outcome → binary class (1 = landed, 0 = failed)
  - Binary indicators for Reused, GridFins, Legs
- One-hot encoded categorical columns: Orbit, LaunchSite, LandingPad, Serial
- Final clean dataset used for EDA, modeling, and dashboarding

Flow chart



https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/18228782a697d468bd275e8d589d3ad12ef8784a/labs-jupyter-spacex-Data%20wrangling%20m1%20final.ipynb

EDA with Data Visualization

- Charts were plotted
- **Flight Number vs. Landing Success (Scatter Plot)**
    - Visualized trends over time; showed increasing success rates as flight numbers grew
- **Payload Mass vs. Landing Success (Scatter Plot)**
    - Assessed how payload weight impacted landing outcome; identified performance range thresholds
- **Launch Site vs. Landing Success (Categorical Plot)**
    - Compared success rates across different sites; revealed KSC LC-39A as most consistent
- **Orbit vs. Landing Success (Categorical Plot)**
    - Showed which orbit types (e.g., LEO, ISS, GTO) had higher success likelihood
- **Payload Mass vs. Orbit (Scatter Plot)**
    - Explored the distribution of payloads per orbit type; helped understand orbit-related complexity
- **Yearly Trends in Success Rate (Line Plot)**
    - Extracted year from launch dates to track overall progress in landing reliability

https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/638c72c52b954c390732214ac547ebd3caa59e3e/edadataviz%20m2%20final2.ipynb

EDA with SQL

- **SQL Queries Summary**
- Selected unique launch sites using `SELECT DISTINCT`
- Filtered launches from specific sites using `WHERE Launch_Site LIKE 'CCA%'`
- Calculated total payload mass for NASA (CRS) missions
- Found average payload mass for specific booster versions (e.g., F9 v1.1)
- Identified date of first successful ground landing using `MIN()` with `WHERE`
- Listed booster versions with successful drone ship landings and payloads between 4000–6000 kg
- Counted total number of successful and failed missions using `GROUP BY Mission_Outcome`
- Found booster version(s) that carried the maximum payload using a subquery
- Filtered and displayed failures on drone ships for the year 2015 using `SUBSTR()` to extract year and month
- Ranked mission outcomes between specific dates using `ORDER BY` with `COUNT()`
- https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/bb79437684b4d718d703e449074ae6613c4b6a75/jupyter-labs-eda-sql-coursera_sqllite%20m2%20final2%20.ipynb

Build an Interactive Map with Folium

Map Objects Added

- **NASA JSC Marker & Circle**
  Marked base location with radius for visibility
- **Launch Site Markers**
  Added site names using DivIcon for clarity
- **Marker Cluster for Launch Outcomes**
- Green marker: Successful landing
- Red marker: Failed landing
- Used MarkerCluster() to avoid overlapping icons

- **Mouse Position Tool**
  Displays live coordinates on hover
- **Lines to Infrastructure (PolyLines)**
  Drew lines from each launch site to:
- Closest coastline
- Nearest highway, railway, and city
- **Distance Labels**
  Floating text showing distance in km (e.g., 0.52 km)

Purpose

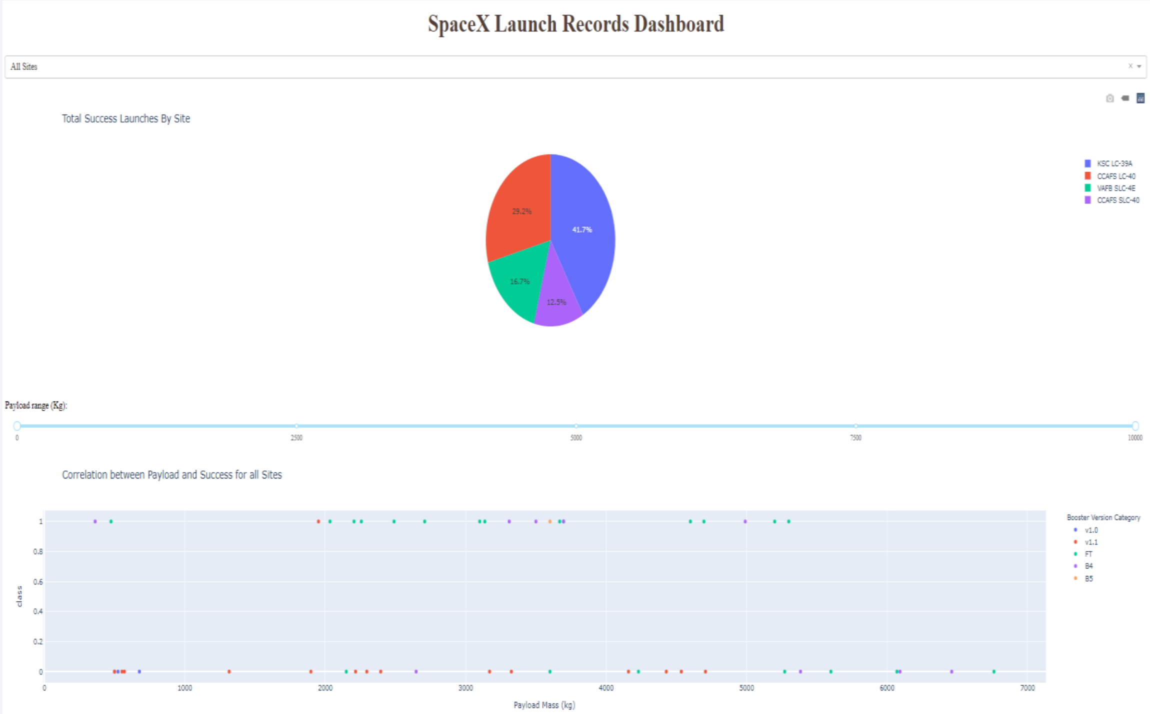
- Visualize **proximity** to coastlines, railways, highways, and cities
- Confirm **strategic placement** of launch sites
- Support analysis of **logistics**, **safety**, and **accessibility**
- Make geographic patterns and relationships **easier to interpret**

https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/18228782a697d468bd275e8d589d3ad12ef8784a/lab_jupyter_launch_site_location%20m3%20final.ipynb

Build a Dashboard with Plotly Dash

- **Dashboard Components**
- **Launch Site Dropdown Menu**
  - Allows user to filter data by specific launch site or select “All Sites”
  - Enables focused analysis by location
- **Payload Range Slider**
  - Lets users select a payload mass range (0–10,000 kg)
  - Used to explore how payload weight affects landing outcomes

- Pie Chart – Launch Success Count
- Shows:
  - Total success by launch site (when “All Sites” selected)
  - Success vs. failure for individual site (when one site is selected)
- **Purpose:** Quickly compare performance across or within sites
- Scatter Plot – Payload vs. Success
- Interacts with **site dropdown** and **payload slider**
- **Purpose**: Explore relationship between payload and success
- Visualize impact of booster version and site on outcome



https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/638c72c52b954c390732214ac547ebd3caa59e3e/spacex-dash-app%20%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash%20m3%20final.py

Predictive Analysis (Classification)

- Selected **target variable**: `Class` (1 = successful landing, 0 = failure)
- Prepared feature matrix using processed dataset (X) and converted target to NumPy array (Y)
- Applied **`StandardScaler`** to normalize feature values
- Split data into training and test sets using `train_test_split(test_size=0.2, random_state=2)`
- Trained and evaluated **four classification models**:
- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree
- K-Nearest Neighbors (KNN)
- Used **`GridSearchCV`** for each model with 10-fold cross-validation to:

https://github.com/jliang1112/Applied-Data-Science-Capstone/blob/638c72c52b954c390732214ac547ebd3caa59e3e/SpaceX_Machine%20Learning%20Prediction_Part_5%20m5%20final.ipynb

Results

Exploratory Data Analysis – Results

- **Launch Success by Site**:
  - CCAFS SLC-40 had the most launches
  - KSC LC-39A showed the **highest success rate**
- **Orbit Type Insights**:
  - Missions to **LEO** and **ISS** had higher success
  - **GTO** and **polar orbits** showed slightly lower performance
- **Payload Mass vs. Success**:
  - Landings were most successful when payloads were **< 8,000 kg**
  - Very high payloads (near 10,000 kg) had lower success
- **Time Trend**:
  - Launch success rate improved steadily over time (by flight number/year)
- **Launch Site Location**:
  - All sites were within **3 km of a coastline**
  - Sites were placed away from cities but close to infrastructure (roads, rail)

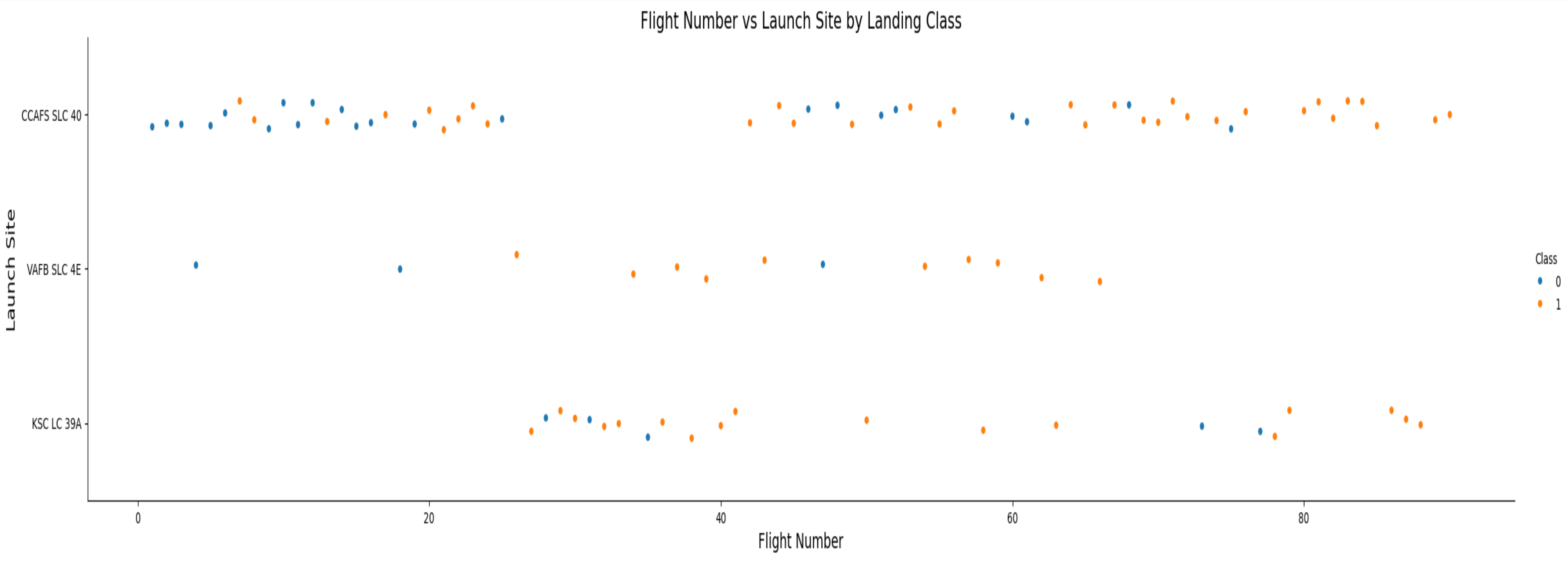
Predictive Analysis – Results

- **Features Used**:
- Payload mass, orbit, launch site, grid fins, reused booster, legs, landing pad
- **Models Evaluated**:
- Logistic Regression
- SVM
- Decision Tree
- K-Nearest Neighbors (KNN)
- **Best Model**:
- **Decision Tree Classifier**
- **87.5%** cross-validation accuracy
- **83.3%** test accuracy
- **Evaluation Metrics Used**:
- Accuracy
- Confusion Matrix
- Precision, Recall, F1-Score
- All models performed consistently, but Decision Tree offered the best **balance of accuracy and interpretability**

Section 2

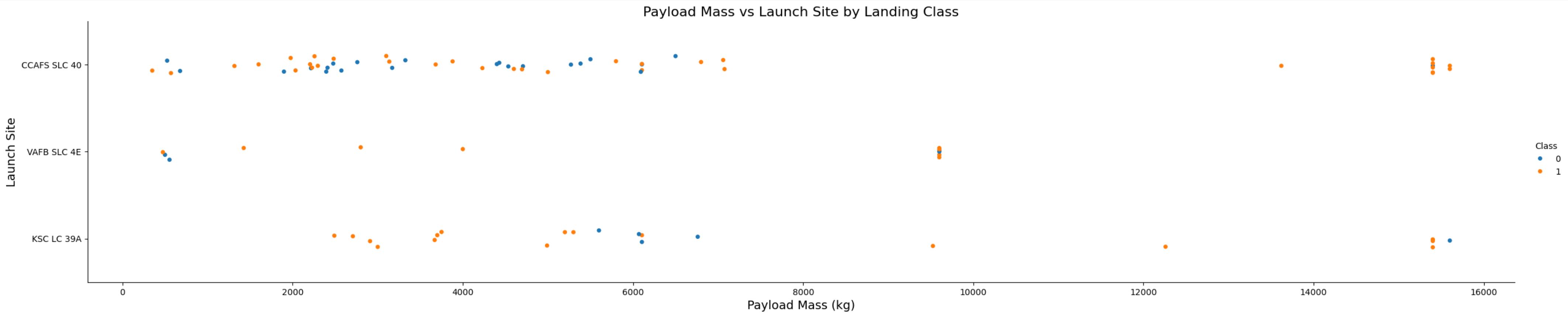
Insights drawn from EDA

Flight Number vs. Launch Site



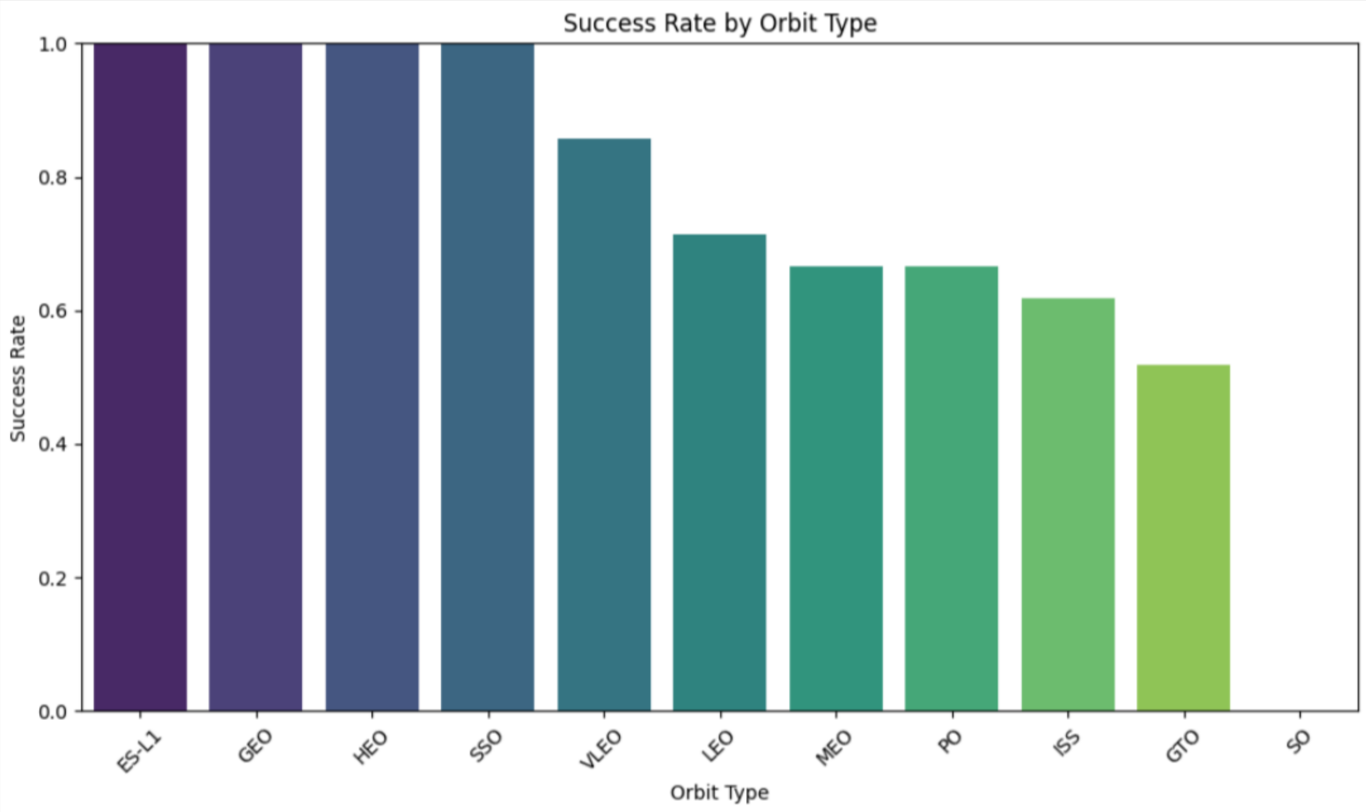
- Launches from **KSC LC-39A** had a **higher success rate** over time.
- **CCAFS SLC-40** had the **highest number of launches**.

Payload vs. Launch Site



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

Success Rate vs. Orbit Type



ES-L1 (Earth–Sun Lagrange Point 1)

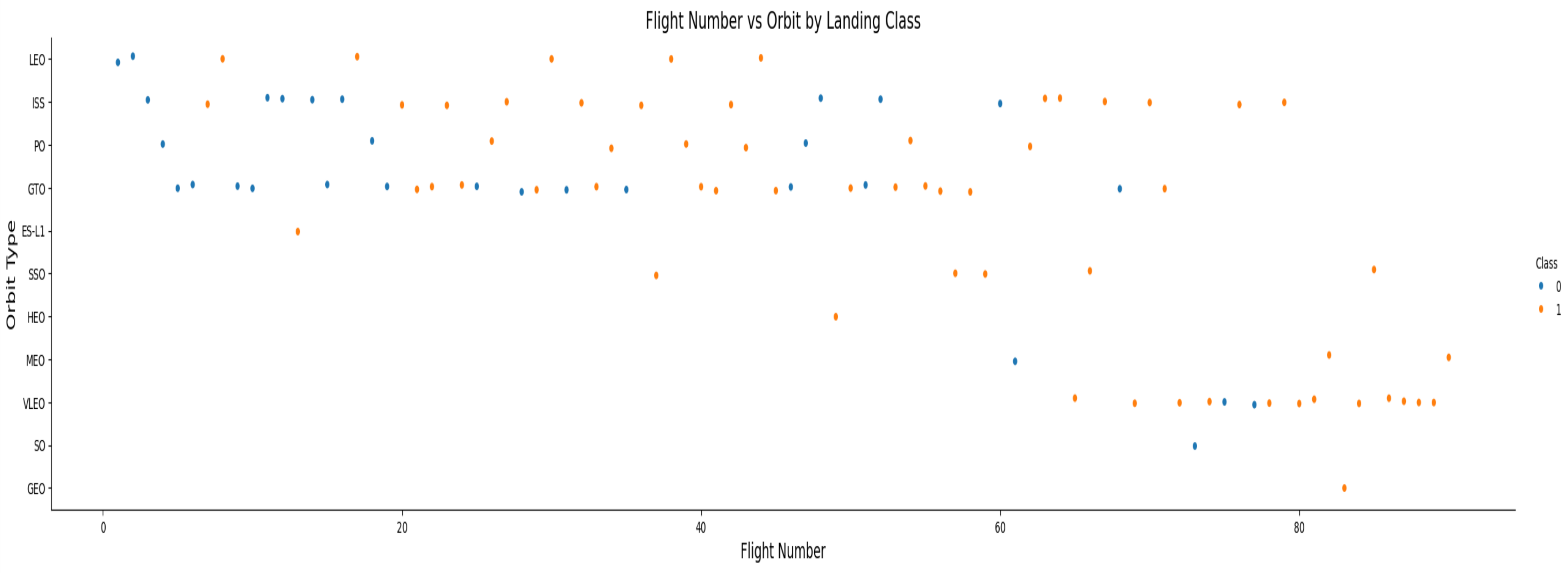
- Only **1 recorded launch** in the dataset
- Resulted in a **successful landing**

GEO (Geostationary Earth Orbit)

- Also had **very few launches**
- **Mixed outcomes**: not consistently successful

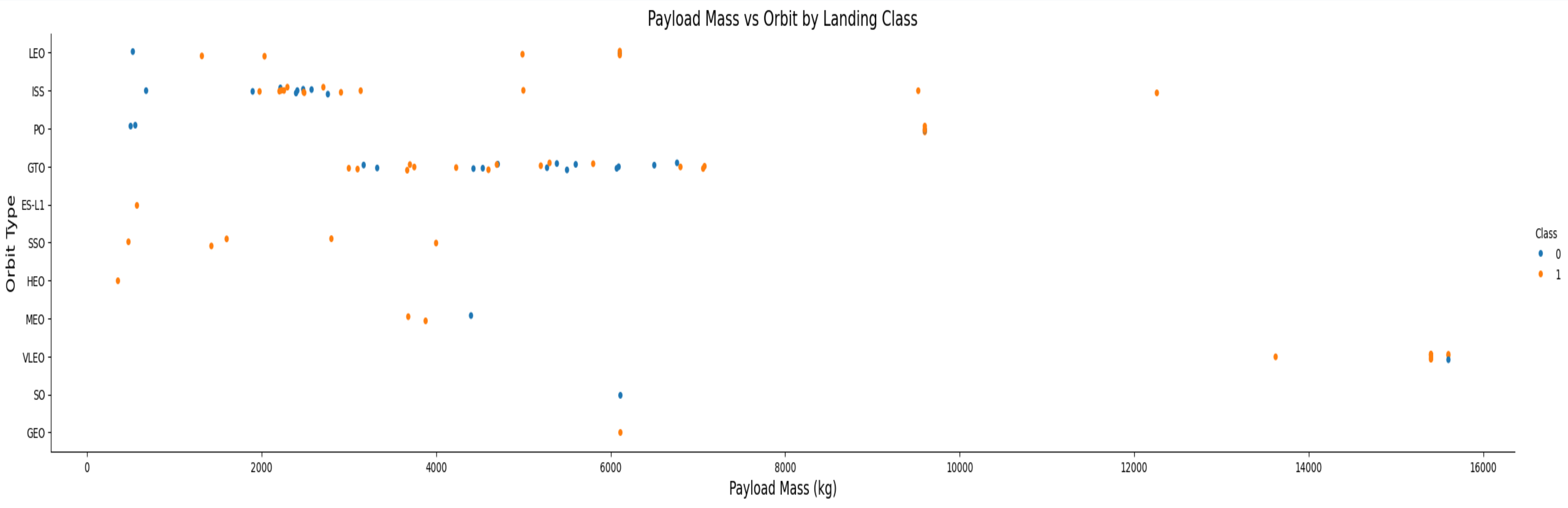
LEO (Low Earth Orbit) and **ISS (International Space Station)** missions had the **highest success rates**

Flight Number vs. Orbit Type



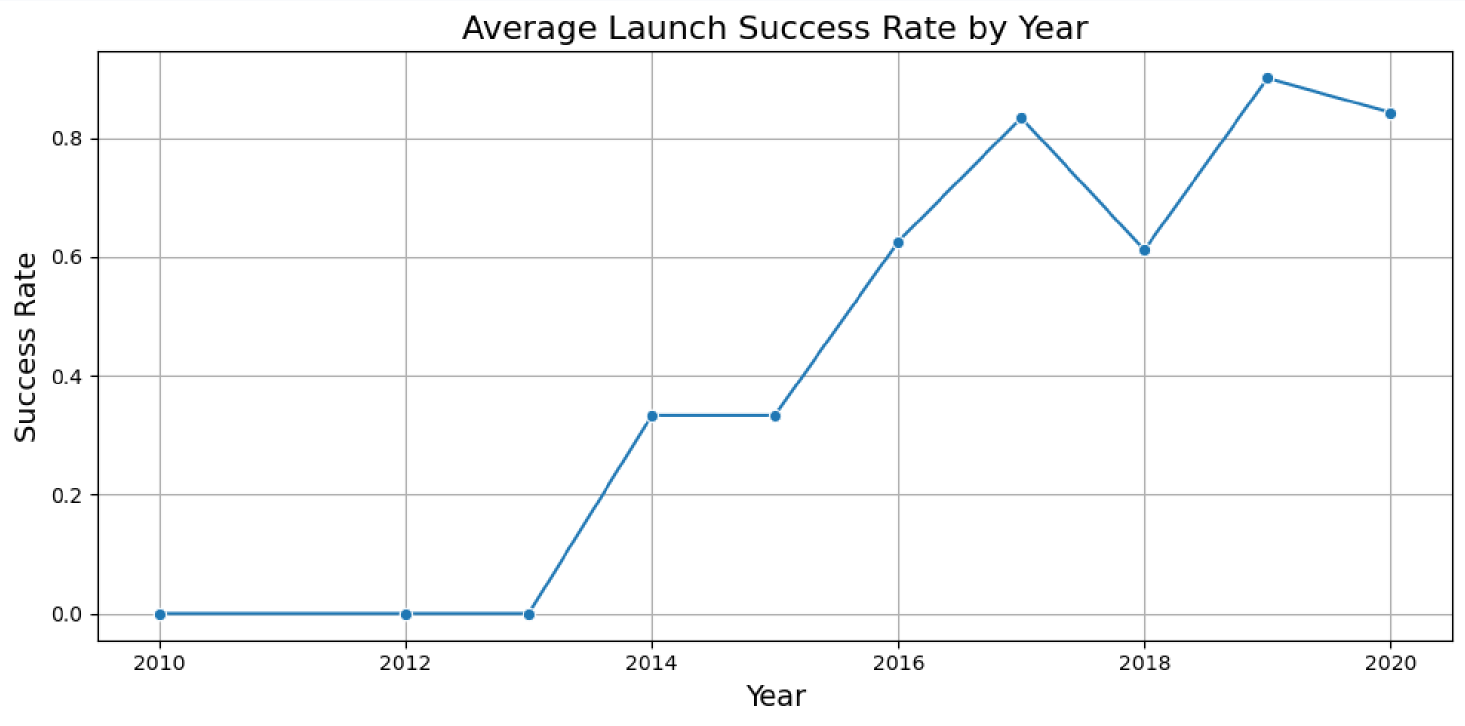
You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

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

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

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

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)

In [14]:

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

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

Out[14]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

Task 4

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

Task 5

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

Hint:Use min function

17]:
```
%load_ext sql

%sql sqlite:///my_data1.db

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

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
 * sqlite:///my_data1.db
Done.
```

17]:

First_Successful_Ground_Landing
2015-12-22

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 Booster_Version FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass__kg_" > 400(
```

```
 * 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

In [19]:

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

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

Out[19]:

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

Boosters Carried Maximum Payload

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

```
]:  %%sql
    SELECT DISTINCT "Booster_Version"
    FROM SPACEXTABLE
    WHERE "Payload_Mass__kg_" = (
        SELECT MAX("Payload_Mass__kg_")
        FROM SPACEXTABLE
    );
```

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

]:

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

2015 Launch Records

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 Booster_Version, Launch_Site, Landing_Outcome, substr(Date, 6, 2) AS Month FROM SPACEXTABLE WHERE substr(Date, 
```

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

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

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.
```

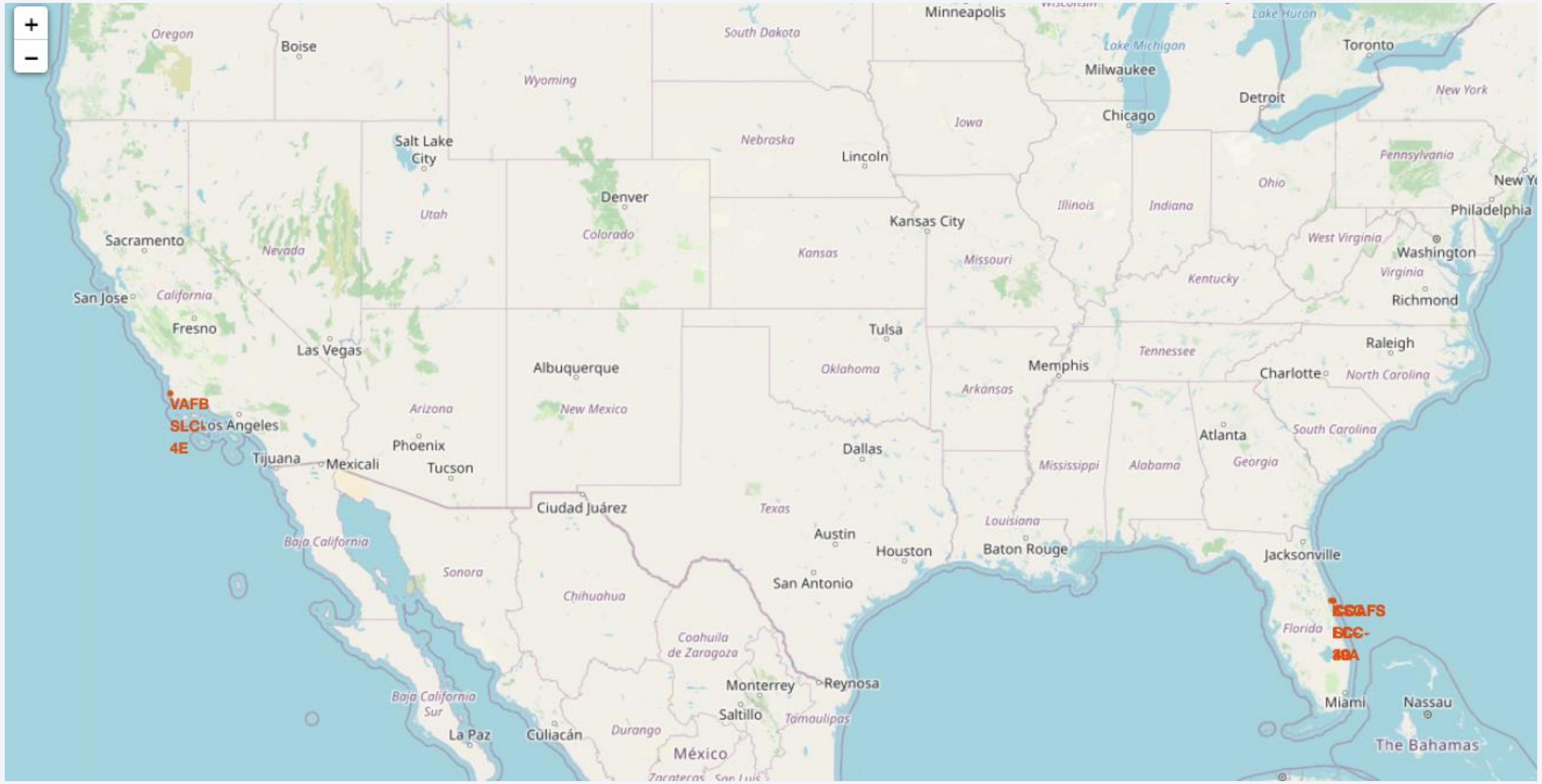
]:

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

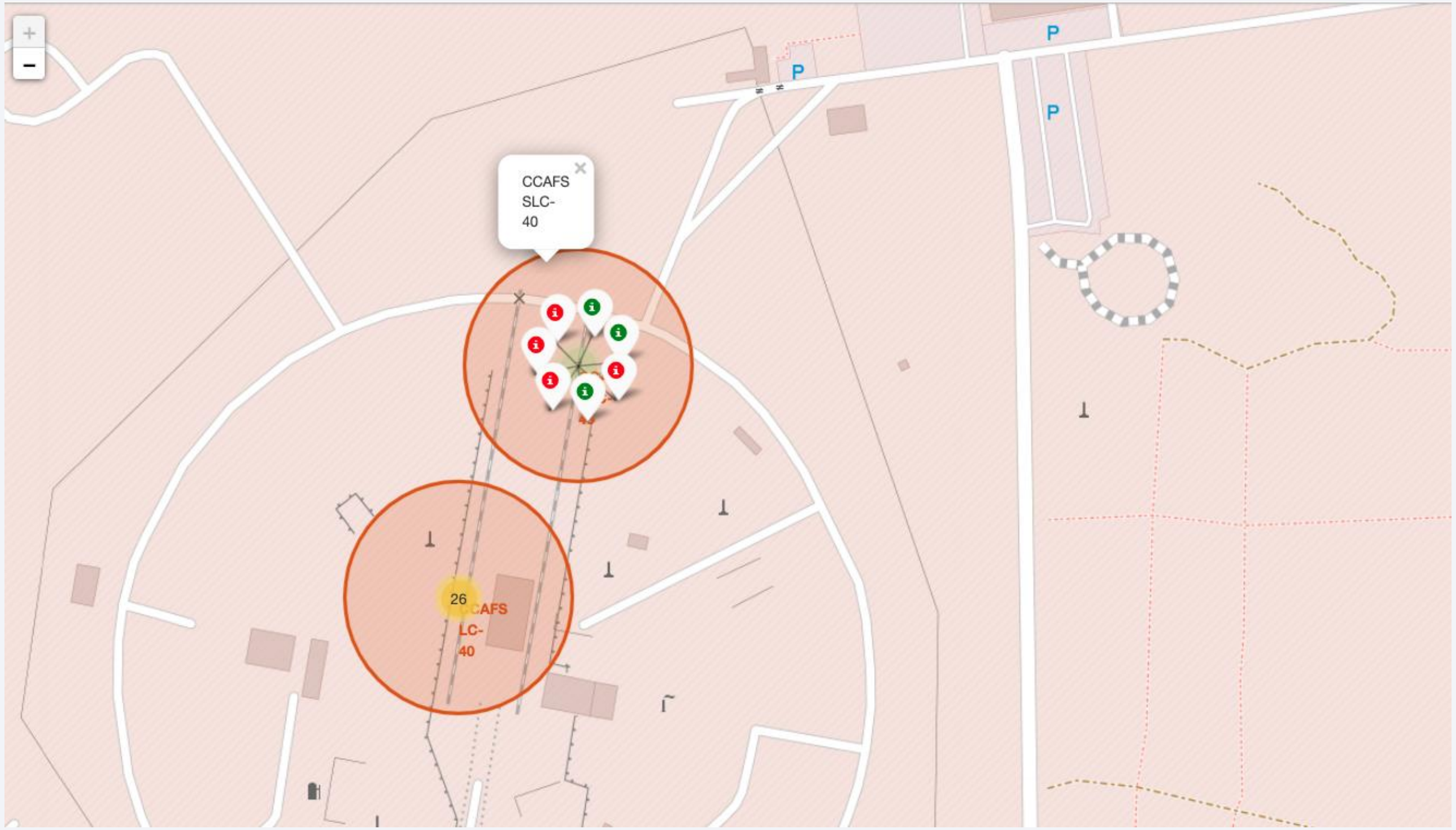
Launch Sites Proximities Analysis

The generated map with marked launch sites



- All launch sites in proximity to the Equator line
- All launch sites in very close proximity to the coast

Color-labeled launch outcomes map



Green = Success
Red = Failed

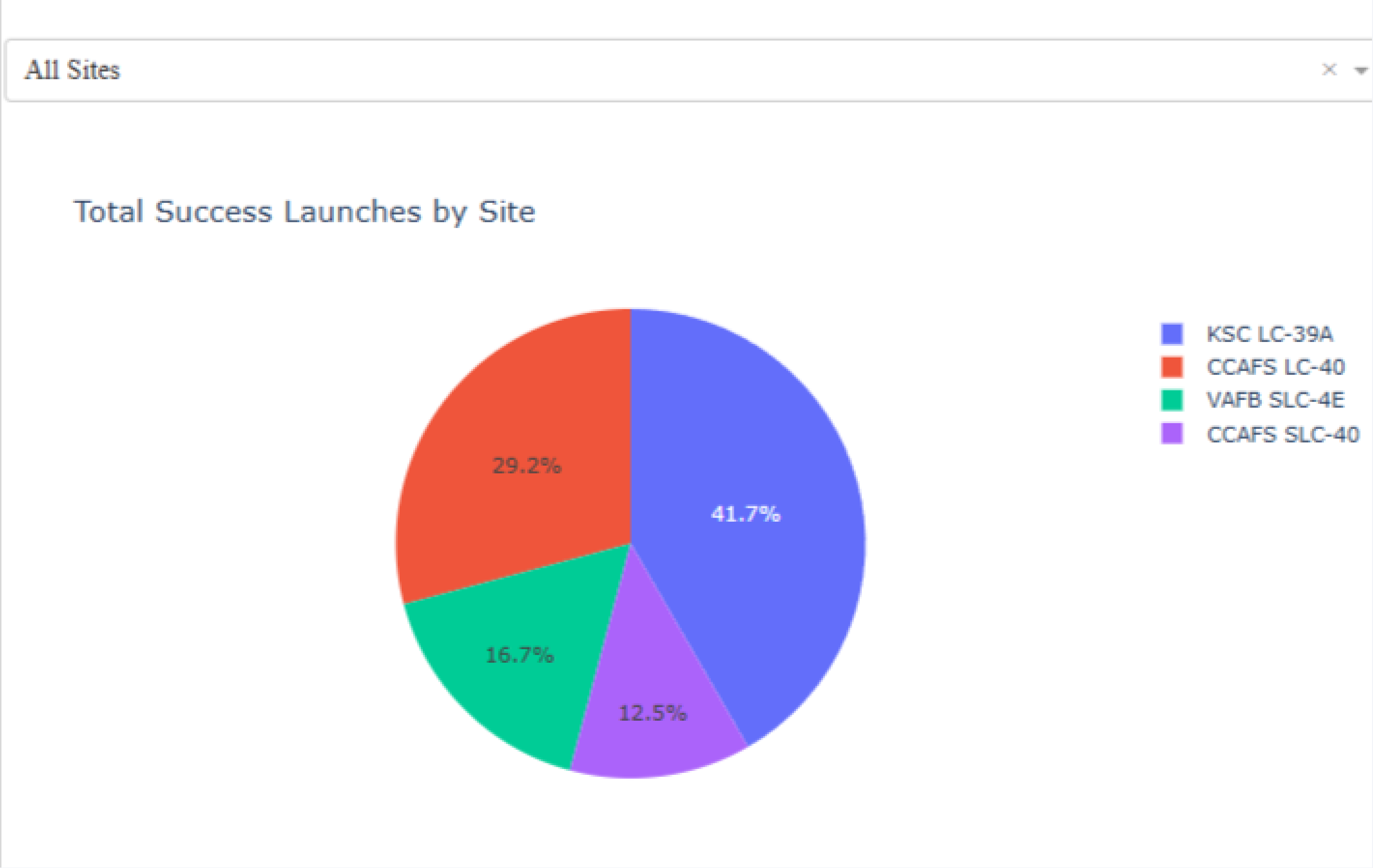
```
spacex_df['marker_color'] = spacex_df['class'].apply(lambda x: 'green' if x == 1 else 'red')
```

Launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

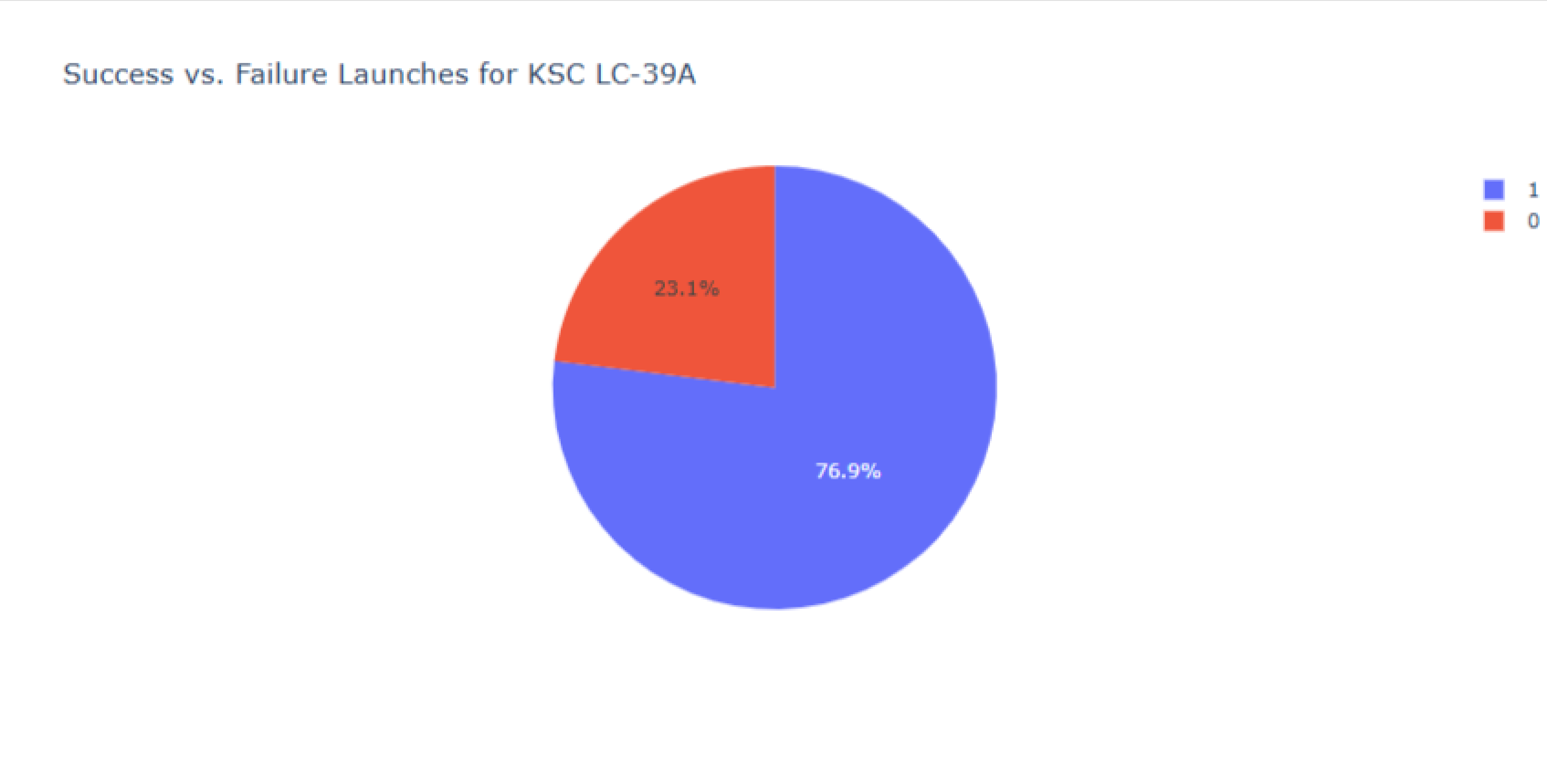
Section 4

Build a Dashboard with Plotly Dash

Total Success Launches by Site

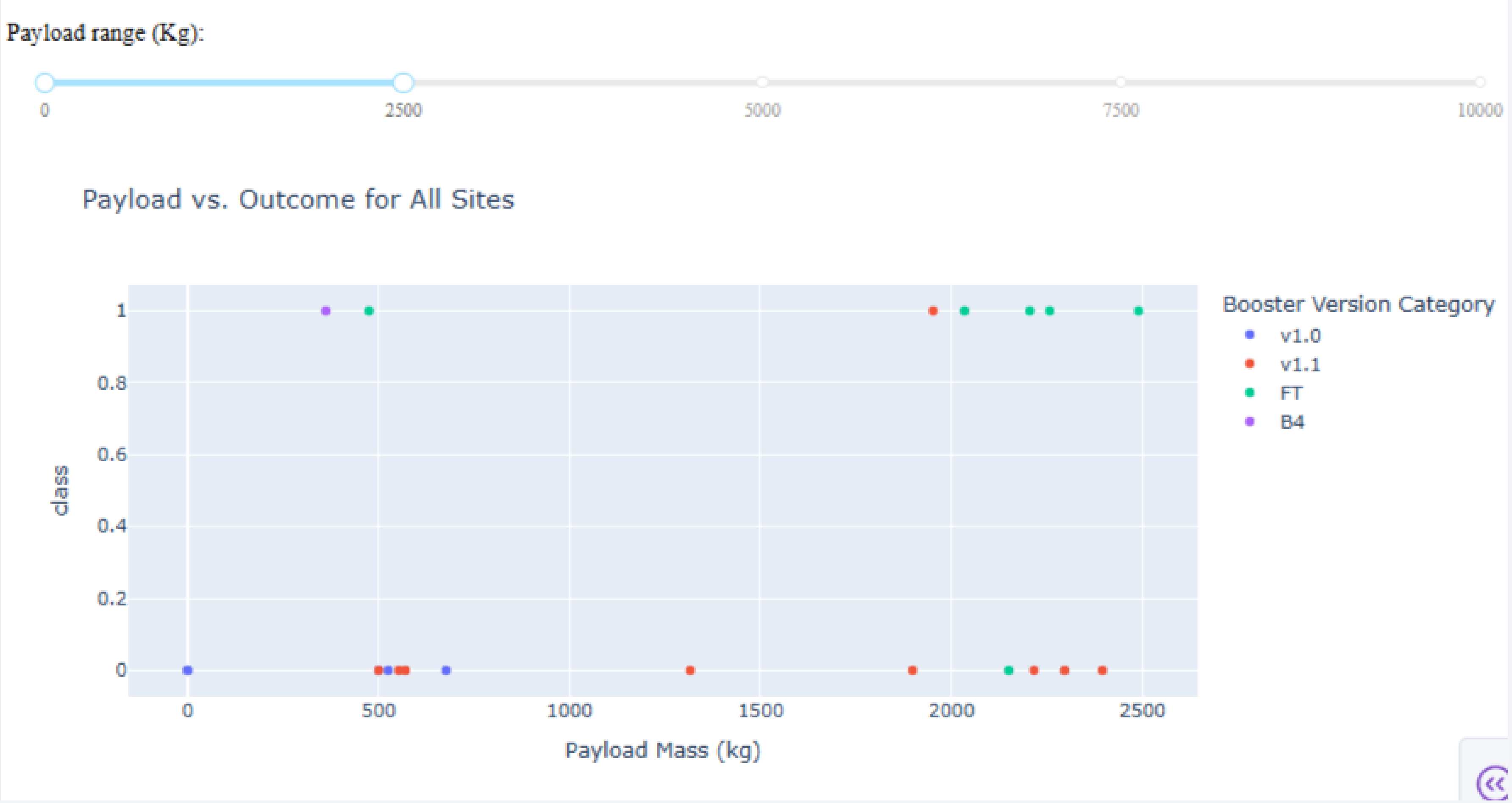


launch site with highest launch success ratio

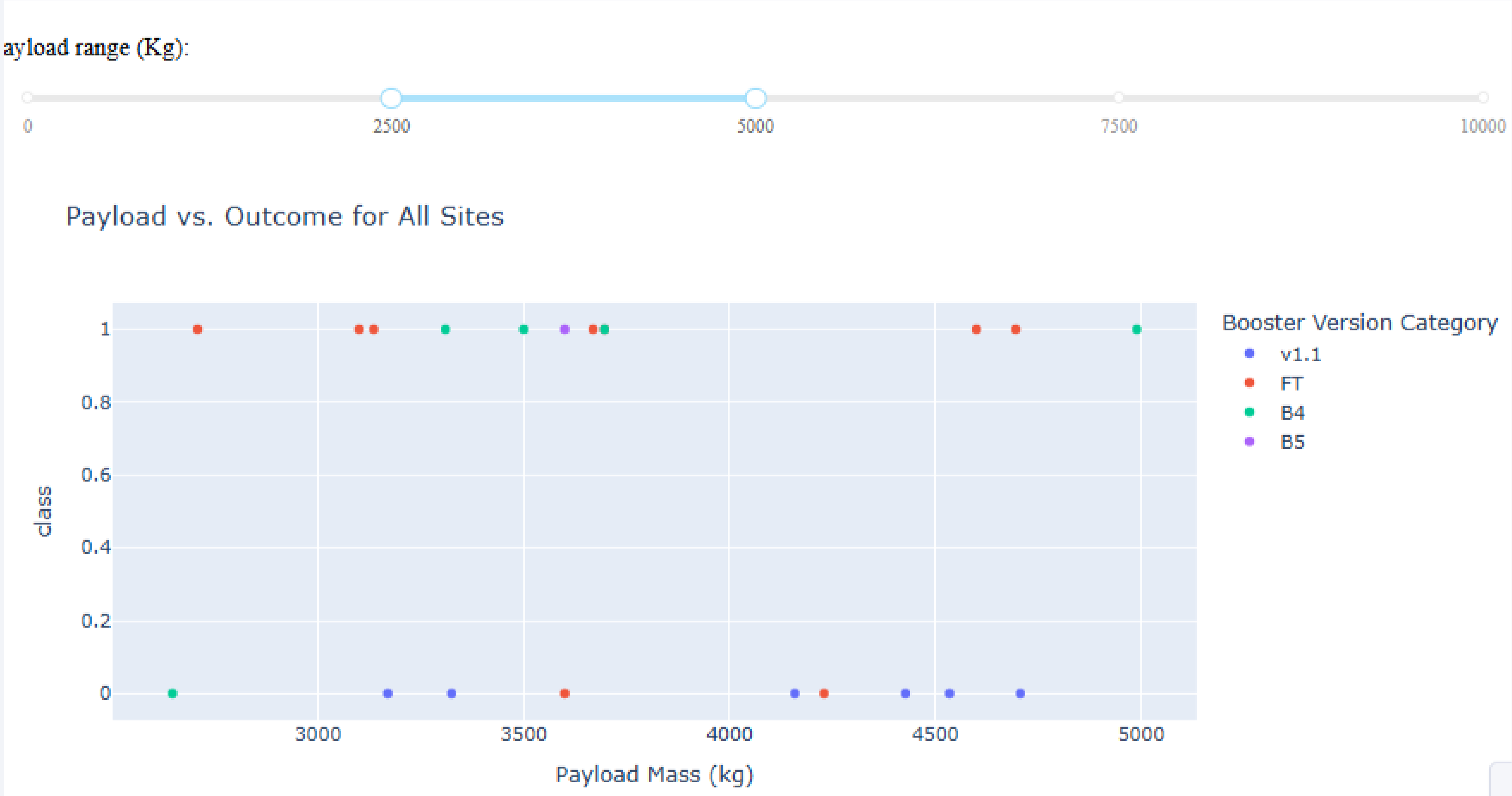


KSC LC-39A has the highest success launch ratio

Payload from 0-2500 kg vs. Launch Outcome scatter plot

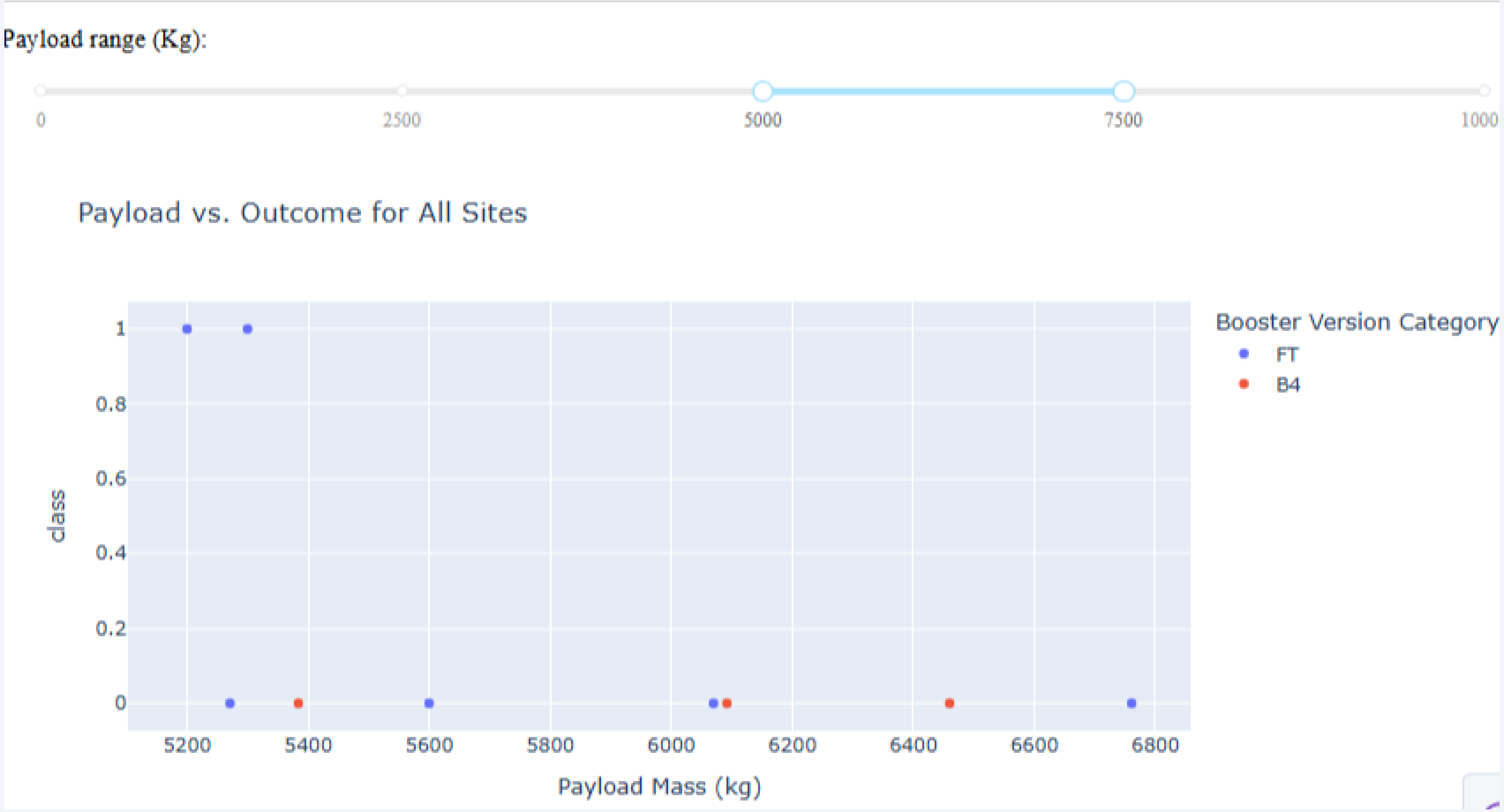


Payload from 2500-5000 kg vs. Launch Outcome scatter plot

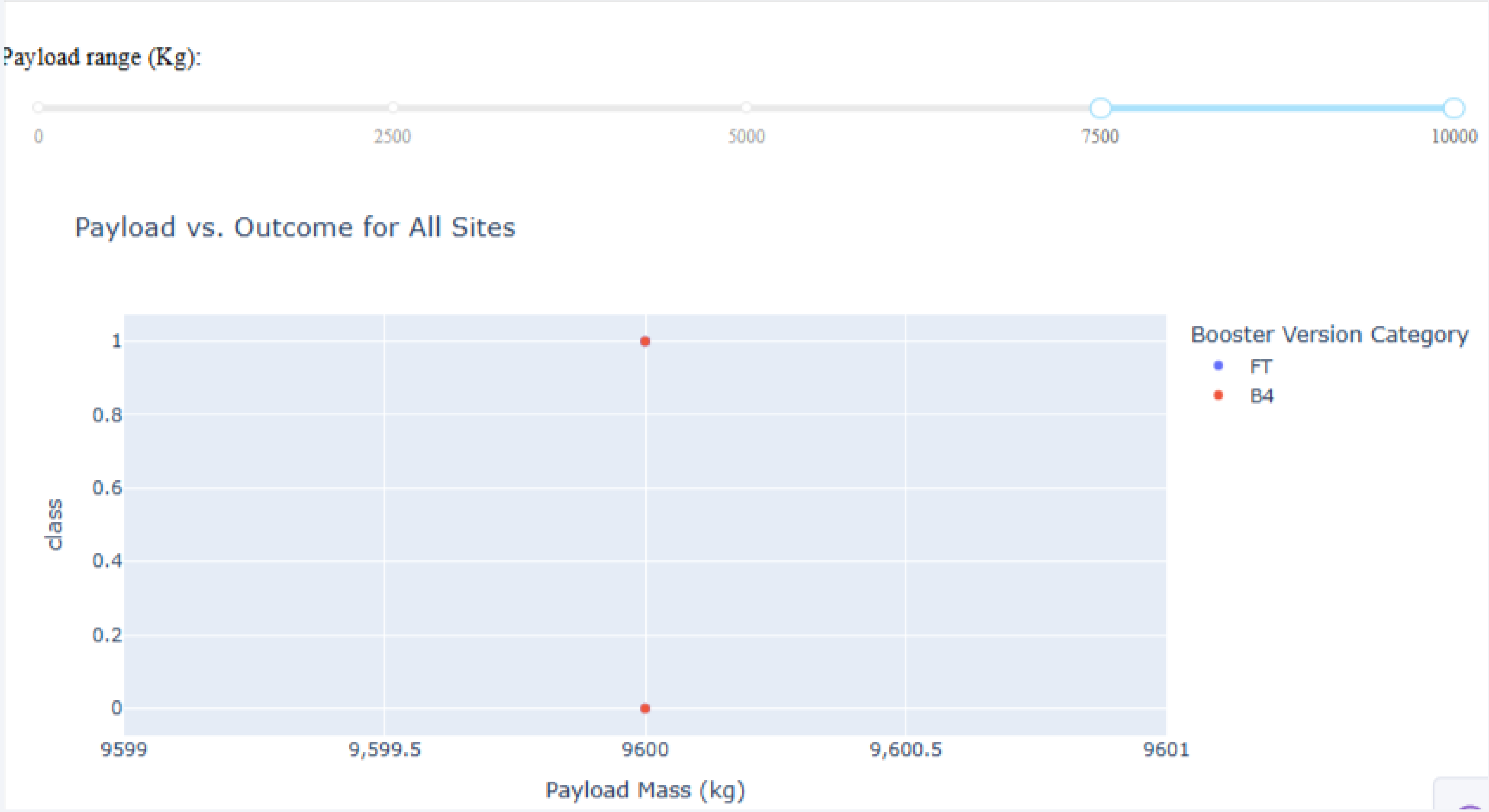


Booster FT have the largest success rate

Payload from 5000-7500 kg vs. Launch Outcome scatter plot



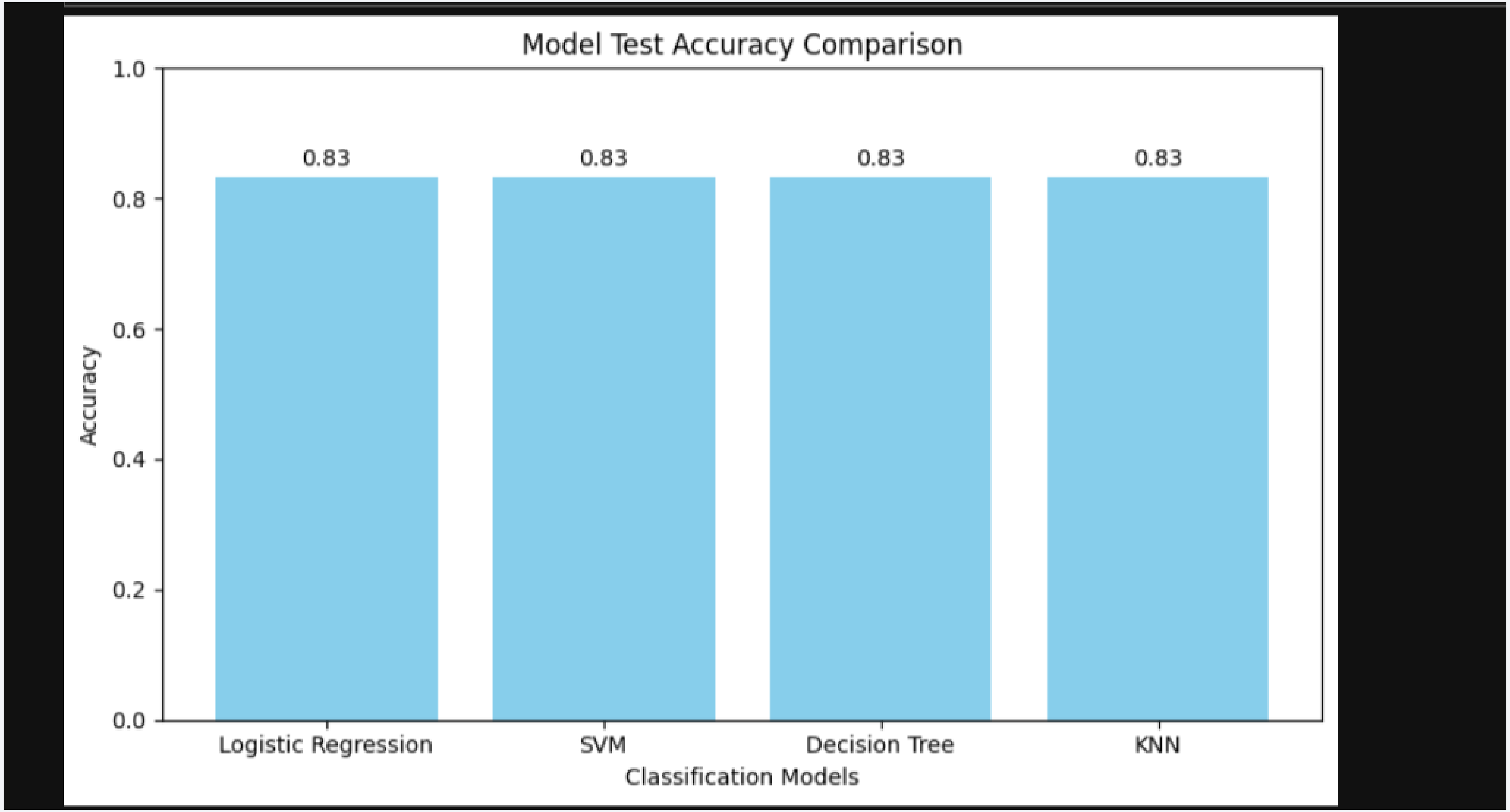
Payload from 7500-10000 kg vs. Launch Outcome scatter plot



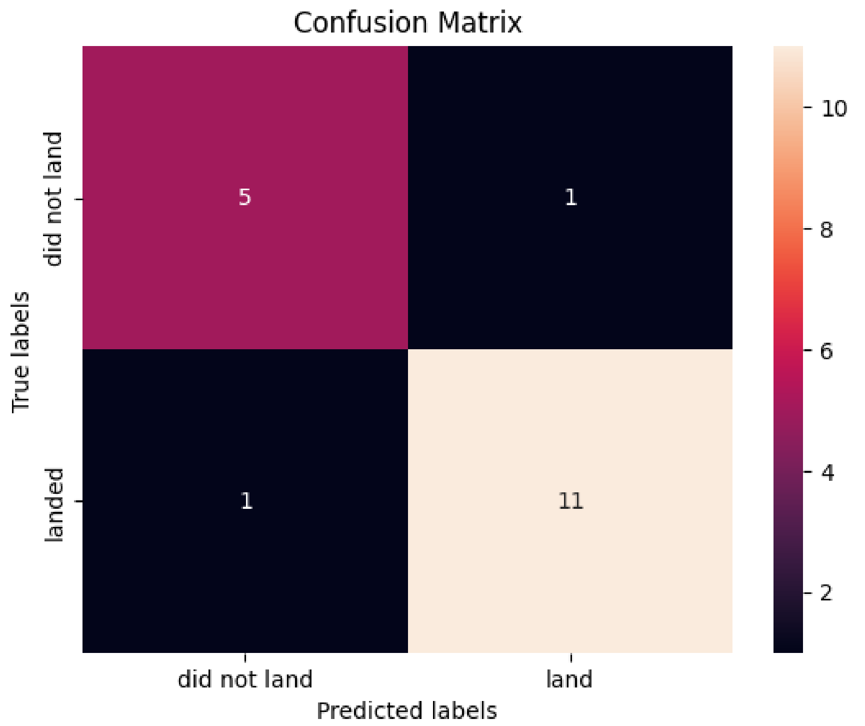
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



	precision	recall	f1-score	support
0	0.83	0.83	0.83	6
1	0.92	0.92	0.92	12
accuracy			0.89	18
macro avg	0.88	0.88	0.88	18
weighted avg	0.89	0.89	0.89	18

- Decision Tree Classifier has the
- **Highest cross-validation accuracy: 87.5%**
- Matched test accuracy with other models: **83.3%**

Conclusion & Key Insights

- **Exploratory Data Analysis (EDA)** revealed:
- **KSC LC-39A** had the **highest landing success rate**
- **CCAFS SLC-40** had the **most launches**
- **LEO and ISS orbits** showed **higher success** than GTO or GEO
- **Payloads between 4000–6000 kg** had the highest landing success
- **Folium maps** showed that launch sites are **strategically located near coastlines and infrastructure**, and away from dense cities.
- **Decision Tree Classifier** had the **highest accuracy (87.5% CV)** and **stable performance (83.3% test accuracy)**.
- Predictive modeling demonstrated that **landing success is learnable and predictable** from mission attributes.
- This approach can support **future mission planning** and **cost optimization**.

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!