IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mariia Snegireva
2025-03-15

Outline

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

Executive Summary

Summary of methodologies:

- We collected SpaceX launch data using **API requests** and **web scraping**
- Performed data **cleaning and preprocessing**, including handling missing values and formatting
- Conducted **Exploratory Data Analysis (EDA)** using SQL and visualized data with Folium and Dash
- Built **machine learning models** to predict the successful landing of the rocket's first stage

Summary of all results:

- Our analysis identified key **factors influencing successful landings**
- The developed model **achieved high accuracy** in predicting first-stage landings
- We created **interactive dashboards** to visualize data and predictions
- The **final result** is a tool that helps estimate the probability of first-stage reuse

Introduction

Project background and context:

- The commercial space industry is growing rapidly, with companies like **SpaceX, Blue Origin, and Rocket Lab** making space travel more affordable
- SpaceX stands out due to its **reusable rocket technology**, which significantly reduces launch costs
- The key to cost reduction is the **successful landing and reuse of the Falcon 9 first stage**
- This project aims to analyze **historical launch data** and use **machine learning** to predict first-stage landings

Problems you want to find answers:

- What factors **influence the success or failure** of a Falcon 9 first-stage landing?
- Can we develop a **machine learning model** to accurately predict landings?
- How can we use **interactive dashboards** to visualize launch outcomes and insights?

Section 1

Methodology

Methodology

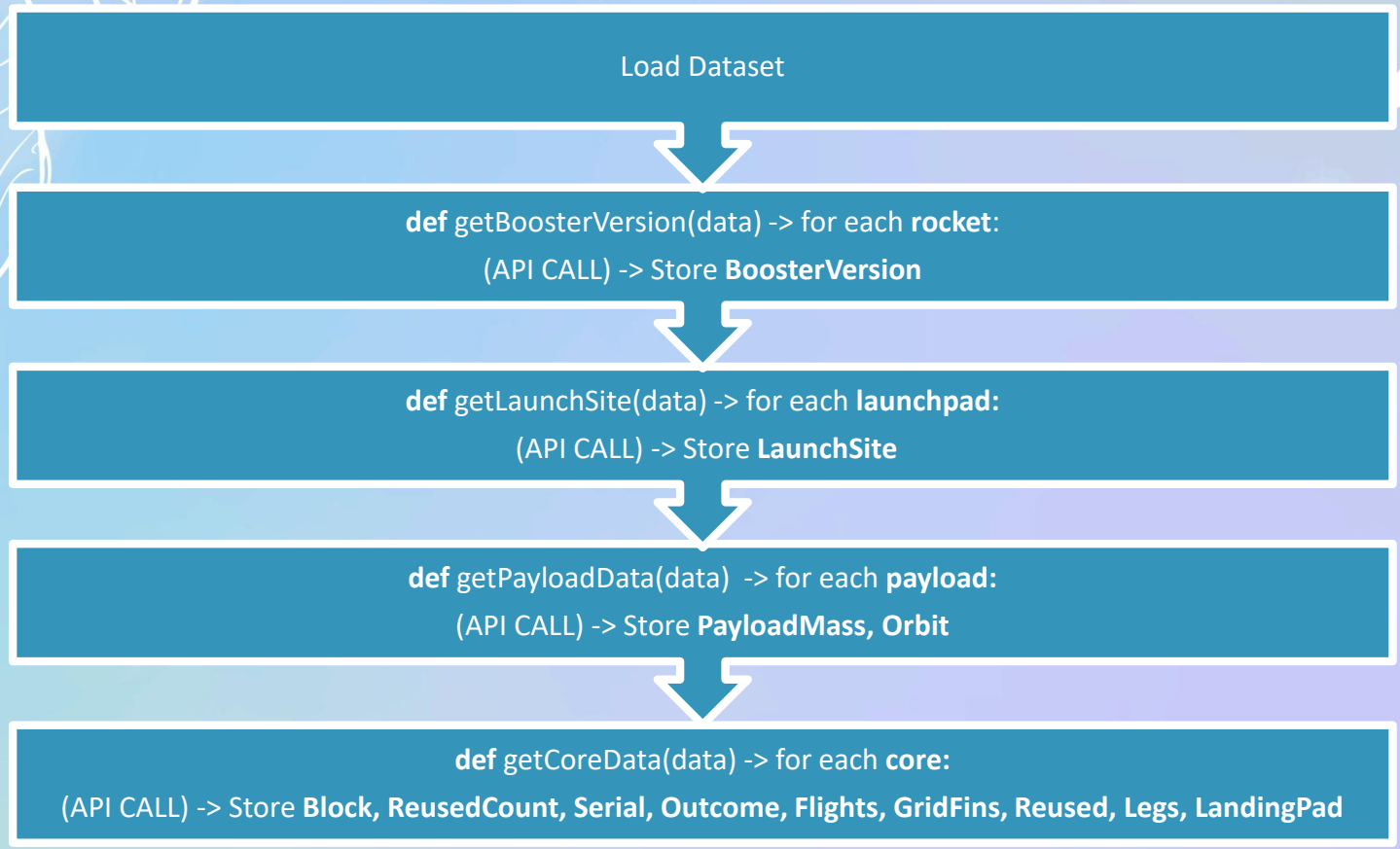
Executive Summary

- Data collection
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Step 1: API Data Retrieval
  - Used **SpaceX REST API** to collect launch data (date, payload, orbit, success/failure of landing).
  - **Python requests** library was used to fetch JSON data.
- Step 2: Web Scraping
  - Extracted **historical launch data** from **Wikipedia** using BeautifulSoup and pandas.
  - Cleaned and structured the extracted tables.
- Step 3: Data Preprocessing
  - Merged API and scraped data into a **single dataset**.
  - Removed missing values, standardized column formats.
- Step 4: Database Storage
  - Stored processed data in a SQL database for easy querying.

Data Collection – SpaceX API



- My Jupyter Notebook is available on this page

Data Collection - Scraping

Fetching Data:
- Sending HTTP request
- Parsing HTML using BeautifulSoup
- Identifying relevant tables

Data Extraction:
- Locating table rows & cells
- Extracting key information (date, booster version, launch outcome, etc.)

Data Processing:
- Using helper functions to clean & format data
- Removing irrelevant columns
- Handling missing values

Data Storage:
- Converting structured data into a Pandas DataFrame

➢ My Jupyter Notebook is available on this page

Data Wrangling

Loading Data	•Use pd.read_csv() to load data from a CSV file •Preview the first 10 rows with df.head(10)
Identifying Missing Values	•Calculate missing values with df.isnull().sum()/len(df)*100. •Identify problematic columns, e.g., LandingPad (28.89% missing)
Determining Data Types	•Use df.dtypes to distinguish numerical and categorical data •Key categorical variables: LaunchSite, Orbit, Outcome
Analyzing Launch Sites	•Apply df['LaunchSite'].value_counts() to count launches per site •Main sites: CCAFS SLC 40, KSC LC 39A, VAFB SLC 4E
Analyzing Orbits	•Use df['Orbit'].value_counts() to analyze orbit types •Common orbits: GTO, ISS, VLEO
Analyzing Mission Outcomes	•Use df['Outcome'].value_counts() to identify successful and failed landings •Categorize landing results (e.g., True ASDS, False Ocean)
Creating Landing Outcome Labels	•Define landing_class (0 = failure, 1 = success) •Add new column df['Class’]
Calculating Success Rate	•Compute df["Class"].mean() to determine the percentage of successful landings

- My Jupyter Notebook is available on this page

EDA with Data Visualization

- **Flight Number vs. Payload Mass** (Scatter Plot) – To analyze how flight experience and payload mass affect success
- **Flight Number vs. Launch Site** (Scatter Plot) – To explore if the launch site influences success
- **Payload Mass vs. Launch Site** (Scatter Plot) – To check for a correlation between payload mass and launch site
- **Orbit Type vs. Success Rate** (Bar Chart) – To compare success rates across different orbits
- **Flight Number vs. Orbit** (Scatter Plot) – To see if flight experience impacts success in various orbits
- **Payload Mass vs. Orbit** (Scatter Plot) – To examine how payload mass affects success in different orbits
- **Launch Success Over Time** (Line Chart) – To track SpaceX's success trends over the years
- My Jupyter Notebook is available on this page

EDA with SQL

- **SELECT DISTINCT Launch_Site FROM spacex;** –> Retrieve unique launch sites.
- **SELECT * FROM spacex WHERE Launch_Site LIKE 'CCA%';** –> Find launch sites starting with 'CCA'.
- **SELECT SUM(PAYLOAD_MASS__KG_) FROM spacex WHERE Customer = 'NASA (CRS)';** – > Total payload mass for NASA (CRS).
- **SELECT AVG(PAYLOAD_MASS__KG_) FROM spacex WHERE Booster_Version = 'F9 v1.1';** –> Average payload mass for F9 v1.1.
- **SELECT MIN(Date) FROM spacex WHERE Landing_Outcome = 'Success (ground pad)';** –> First successful ground pad landing.
- **SELECT Booster_Version FROM spacex WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;** –> Successful drone ship landings (4000-6000 kg payload).
- **SELECT Landing_Outcome, COUNT(*) FROM spacex GROUP BY Landing_Outcome;** –> Count of successful and failed launches.
- **SELECT Booster_Version FROM spacex WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacex);** -> Boosters with the highest payload mass.
- My completed EDA with SQL notebook is available on this page

Build an Interactive Map with Folium

- **Launch Site Markers & Circles:** Each launch site is marked with a **circle** and labeled with a **marker** to visualize its location
- **Success/Failure Markers:** Individual launch records are marked using **green (success)** and **red (failure)** markers, grouped in a **MarkerCluster**
- **Proximity Analysis:**
  - **MousePosition** was used to find key landmarks like coastlines, railways, highways, and cities
  - **Lines (PolyLines)** were drawn to measure distances between launch sites and these proximities
  - **Markers** were added to highlight these points with distance labels
- Findings:
  - Launch sites are **close to the coastline** for safety and trajectory benefits
  - They are near railways & highways for easy transportation of rockets
  - Launch sites are away from cities to minimize risks in case of failures
- My Interactive Folium map is available on this page

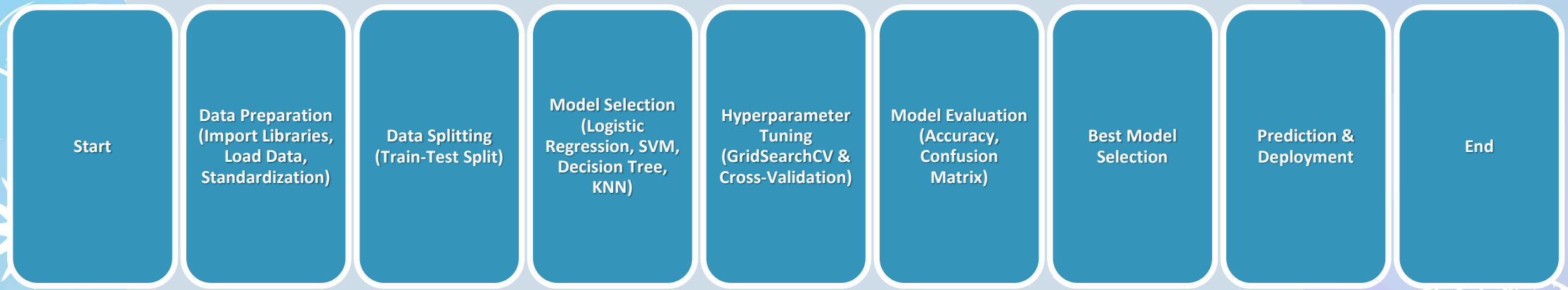
Build a Dashboard with Plotly Dash

- This dashboard visualizes SpaceX launch data with two key interactive graphs:

1. Success Pie Chart 🚀
   - Displays the success rate of launches
   - Users can filter by launch site using a dropdown
   - Helps identify which sites have the highest success rates
2. Payload vs. Success Scatter Plot 🚀
   - Shows the correlation between payload mass and launch success
   - Users can filter by site and adjust payload range with a slider
   - Highlights performance trends across different booster versions

- My completed Plotly Dash lab is available on this page

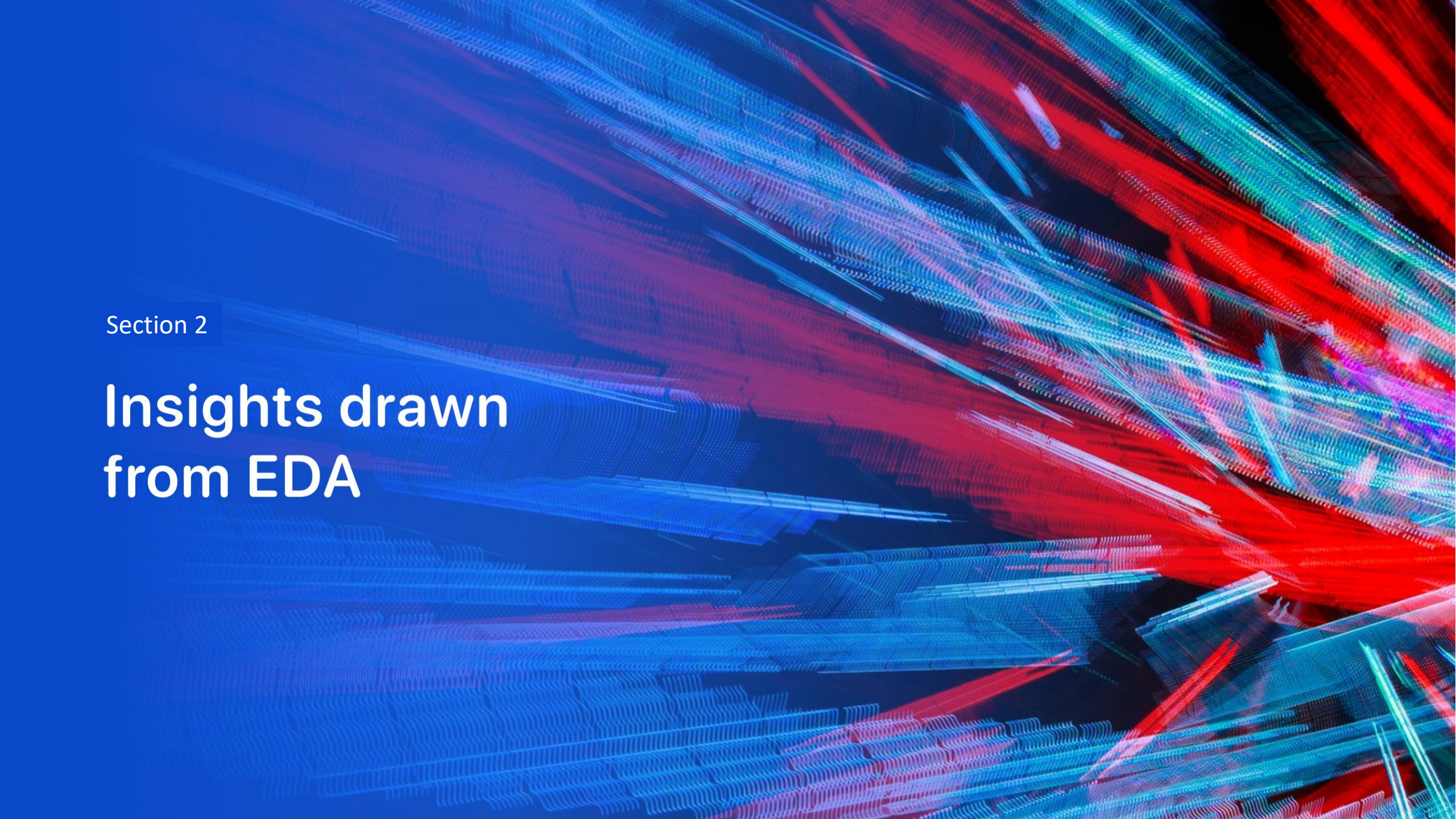
Predictive Analysis (Classification)



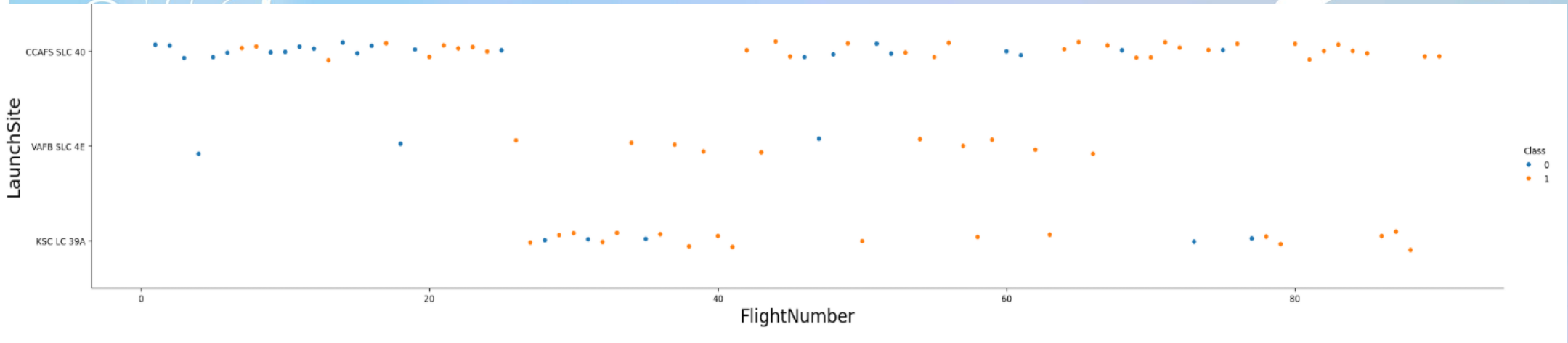
- My completed predictive analysis lab is available on this page

Results

- Exploratory Data Analysis (EDA)
  - Distribution of successful vs. failed landings.
  - Key factors affecting landing success (e.g., payload, launch site)
  - Correlation heatmap of important variables
- Interactive Analytics Demo
  - Dashboard for exploring launch data
  - Filters for launch sites, payloads, and success rates
- Predictive Analysis
  - Best Model: KNN
  - Accuracy: 83%
  - Key Features: Flight Number, Orbit Type, Year, Payload Mass

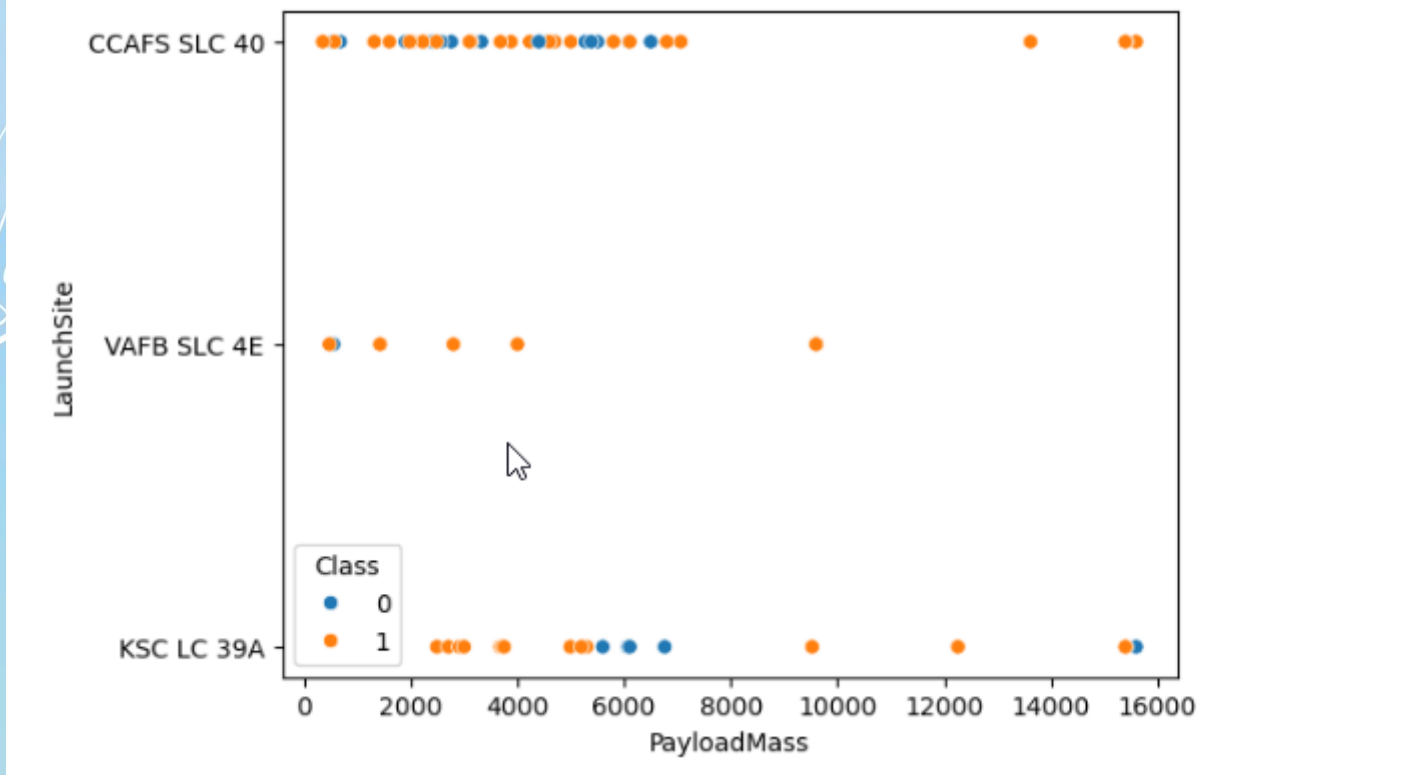

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



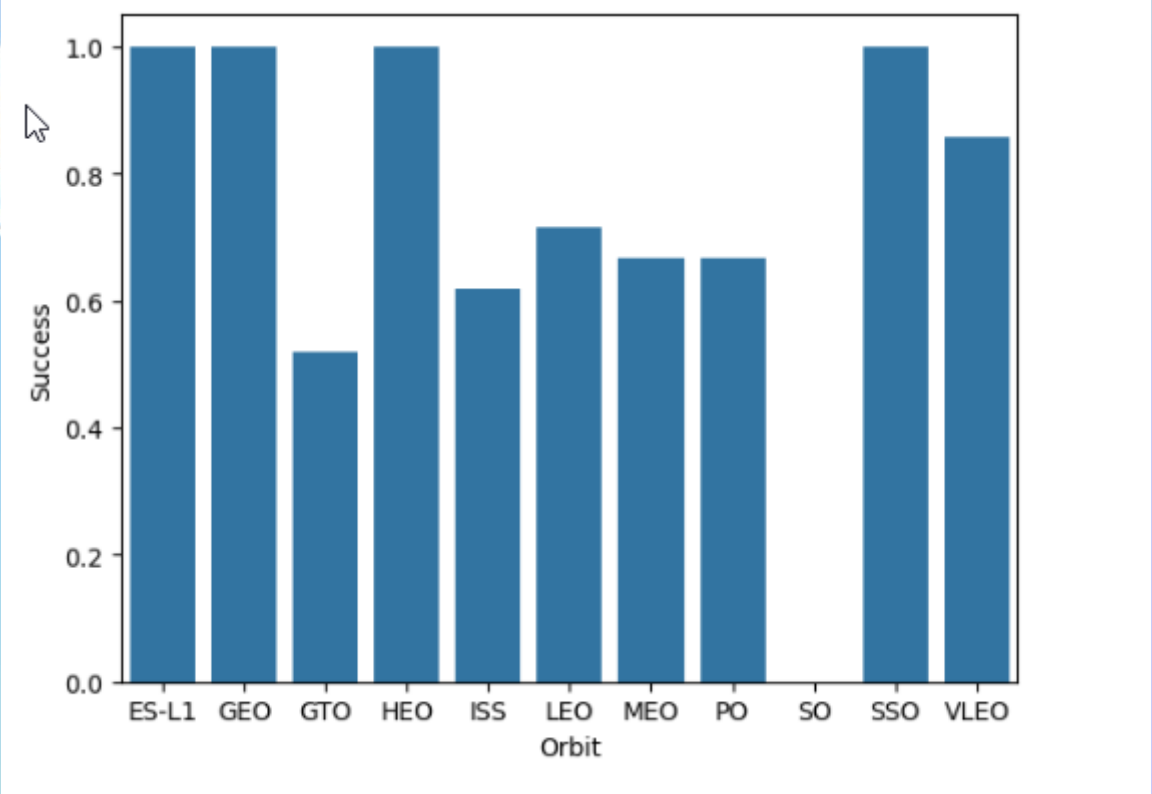
- In the beginning (**low** Flight Numbers), there were more **unsuccessful** launches (blue dots).
- As the **Flight Number increased** (indicating more experience for SpaceX), the number of **successful** launches (orange dots) also increased.
- Different **launch sites** show **varying success rates**:
  - KSC LC 39A (bottom) has a high percentage of successful launches
  - CCAFS SLC 40 and VAFB SLC 4E show a more mixed success pattern

Payload vs. Launch Site



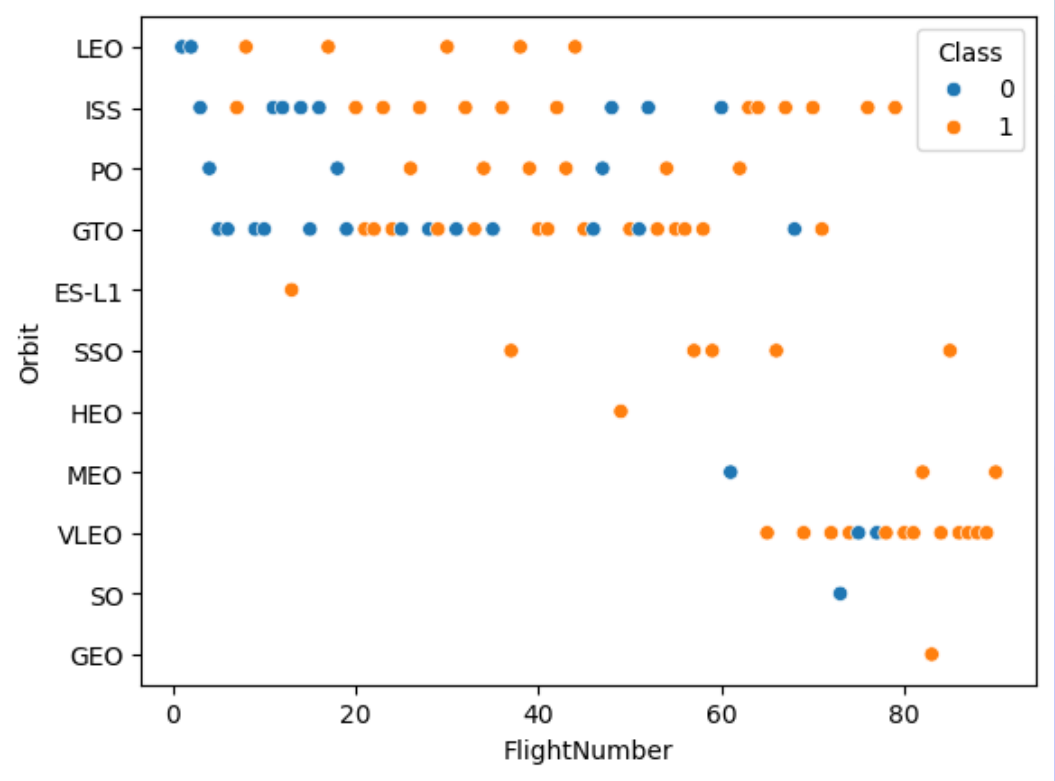
- **CCAFS SLC** 40 has a high success rate across various payload masses
- Successful launches (orange) **outnumber** failures (blue)
- **VAFB-SLC** launchsite hasn't rockets launched for heavy payload mass(greater than 10000)

Success Rate vs. Orbit Type



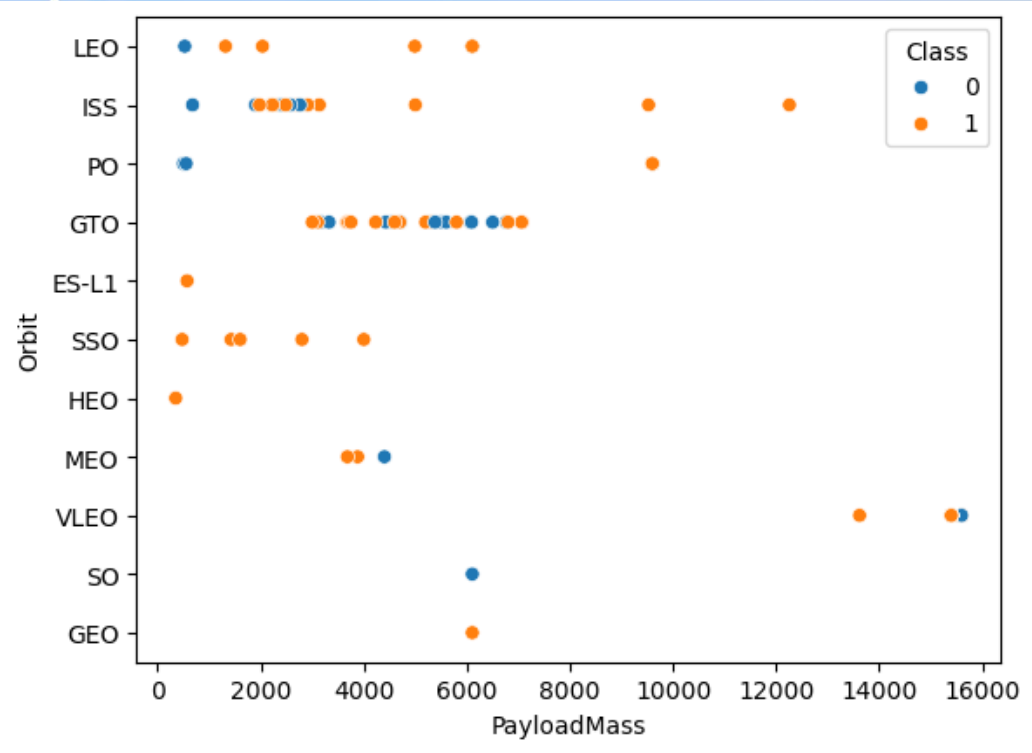
- Missions to **ES-L1, GEO, HEO, SSO** show a **100% success rate**.
- **ISS, LEO, MEO, PO, and VLEO** have success rates between **60-85%**
- **GTO** has the lowest success rate among all orbits

Flight Number vs. Orbit Type



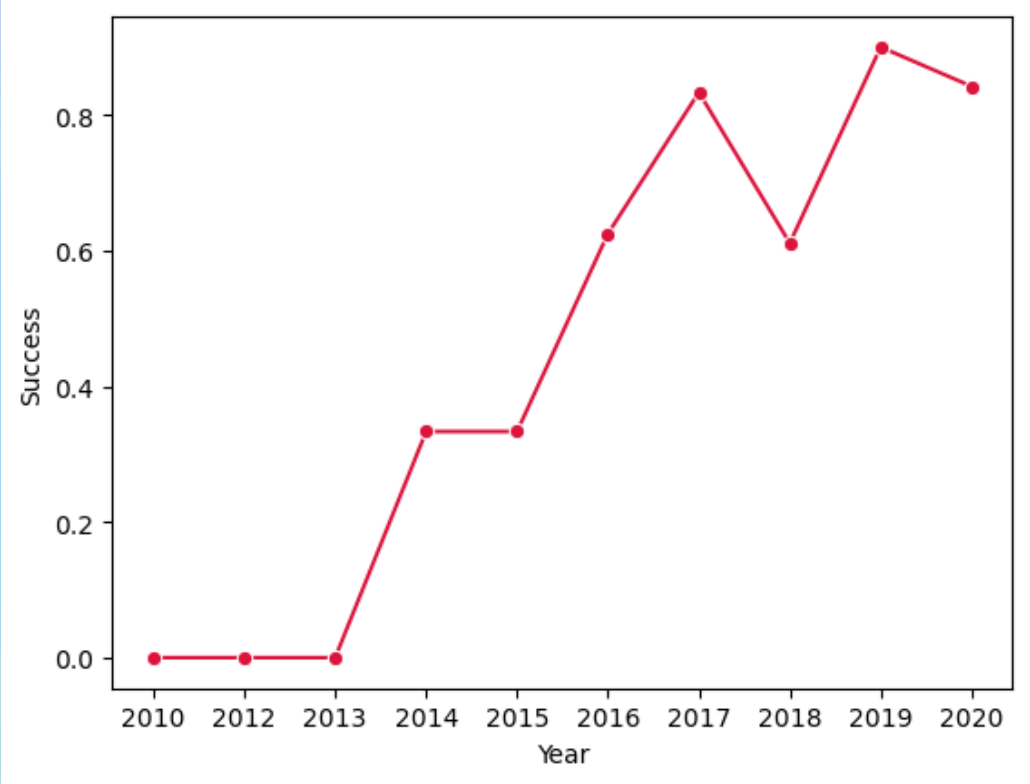
- As the **flight number increases**, the proportion of successful missions (**orange dots**) grows
- **LEO, ISS, and SSO** show a high number of successful launches
- **GTO and some other orbits** have more failures, suggesting they are more challenging
- Most recent missions (higher flight numbers) show fewer failures, indicating technological and operational advancements

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for PO, VLEO and ISS
- Failures (**blue dots**) are more common at lower payloads
- **GTO** orbit shows both successes and failures, indicating higher launch difficulty

Launch Success Yearly Trend



- Success rate remained at **0% until 2013** but started increasing from **2014 onward**
- Significant growth in **2016-2017**, reaching over **80% success rate**
- The success rate dropped in **2018**, indicating potential challenges or failures
- Success rate peaked in **2019 (~90%)** and remained high in **2020 (~85%)**, showing consistent performance

All Launch Site Names

In [11]:
```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

```
In [15]: %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

Out[15]:

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 [29]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" like '%NASA (CRS)%'
```

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

Out[29]:

sum(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

In [27]:
```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
      WHERE "Booster_Version" = "F9 v1.1"
```

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

Out[27]:

AVG(PAYLOAD_MASS__KG_)
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

In [31]:
```
%%sql
SELECT MIN("Date") FROM SPACEXTABLE
WHERE "Mission_Outcome" = "Success"
```

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

Out[31]:

MIN("Date")
2010-06-04

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

In [33]:
```
%%sql
SELECT "Booster_version" FROM SPACEXTABLE
WHERE "Landing_Outcome" = "Success (drone ship)"
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

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

Out[33]:

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 [36]:
```
%%sql
SELECT "Mission_Outcome", COUNT(*) AS total_missions
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

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

Out[36]:

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

Boosters Carried Maximum Payload

In [40]:
```
%%sql
SELECT "Booster_Version" FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
```

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

Out[40]:

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

In [45]:

```
%%sql
SELECT substr(Date, 6,2), "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.
```

Out[45]:

substr(Date, 6,2)	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

In [48]:

```
%%sql
SELECT "Landing_Outcome", COUNT(*) AS total
FROM SPACEXTABLE
WHERE "Date" BETWEEN "2010-06-04" and "2017-03-20"
GROUP BY "Landing_Outcome"
ORDER BY total desc
```

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

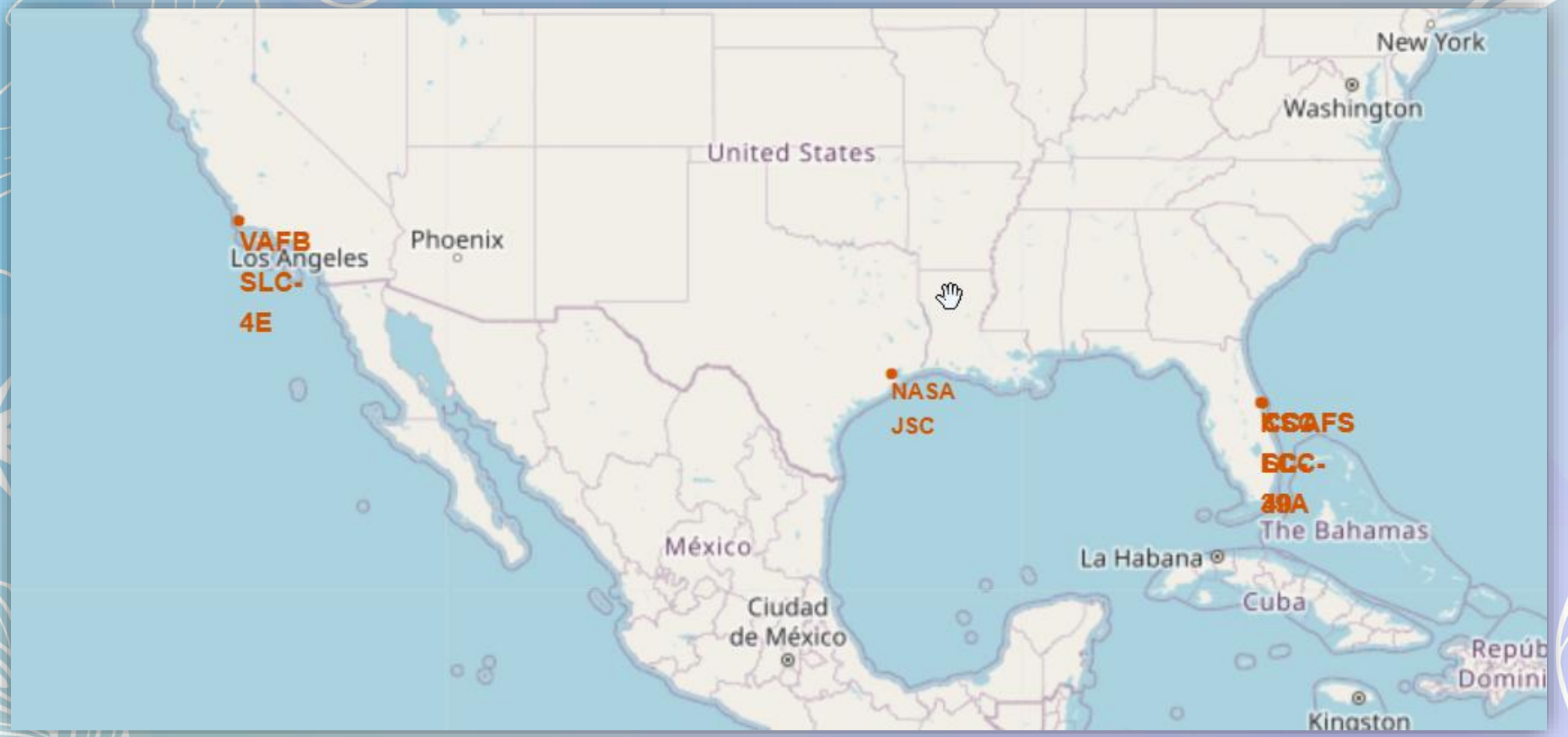
Out[48]:

Landing_Outcome	total
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

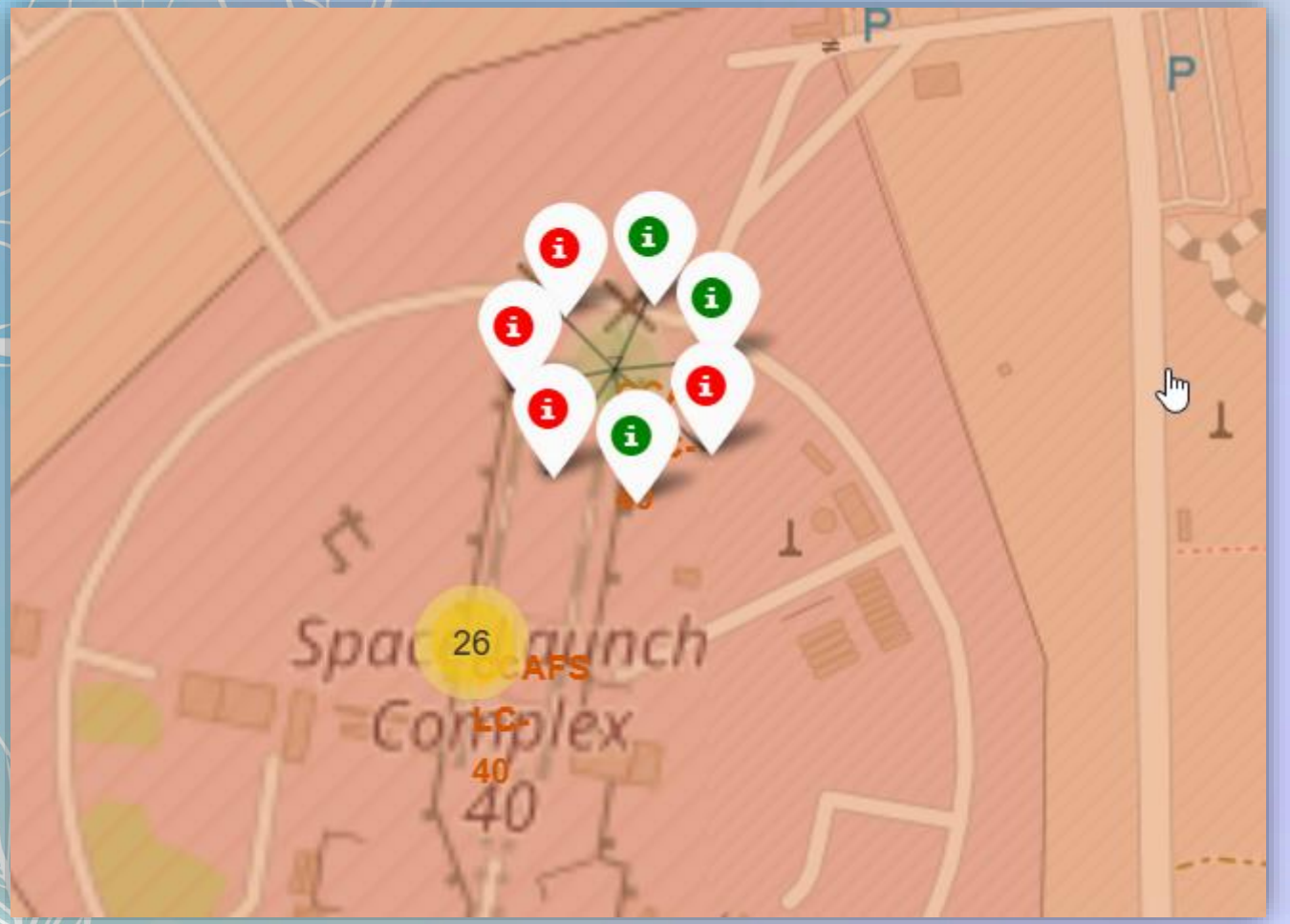
Section 3

Launch Sites Proximities Analysis

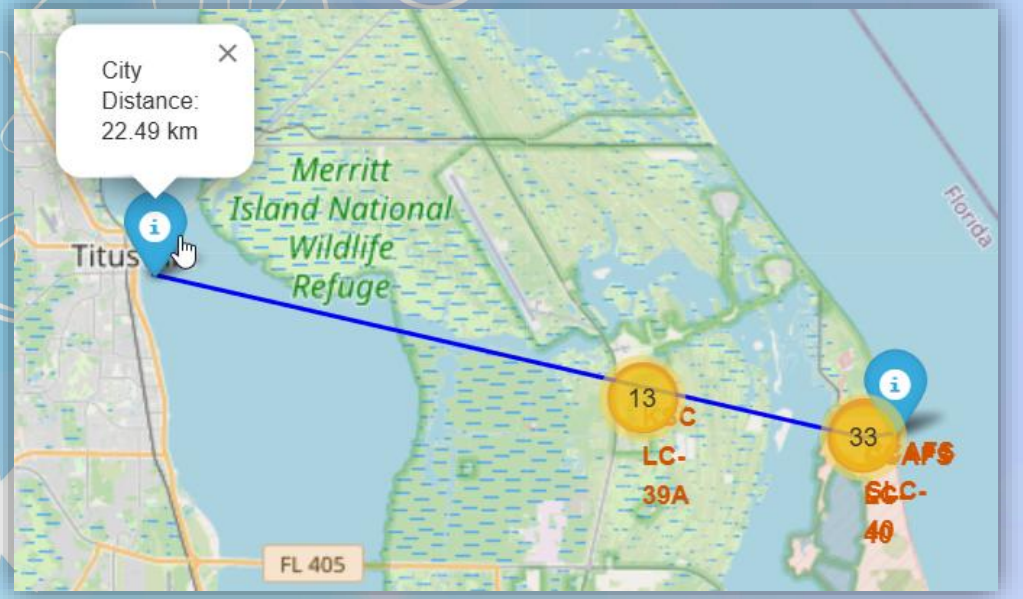
Folium Map: Site's Location

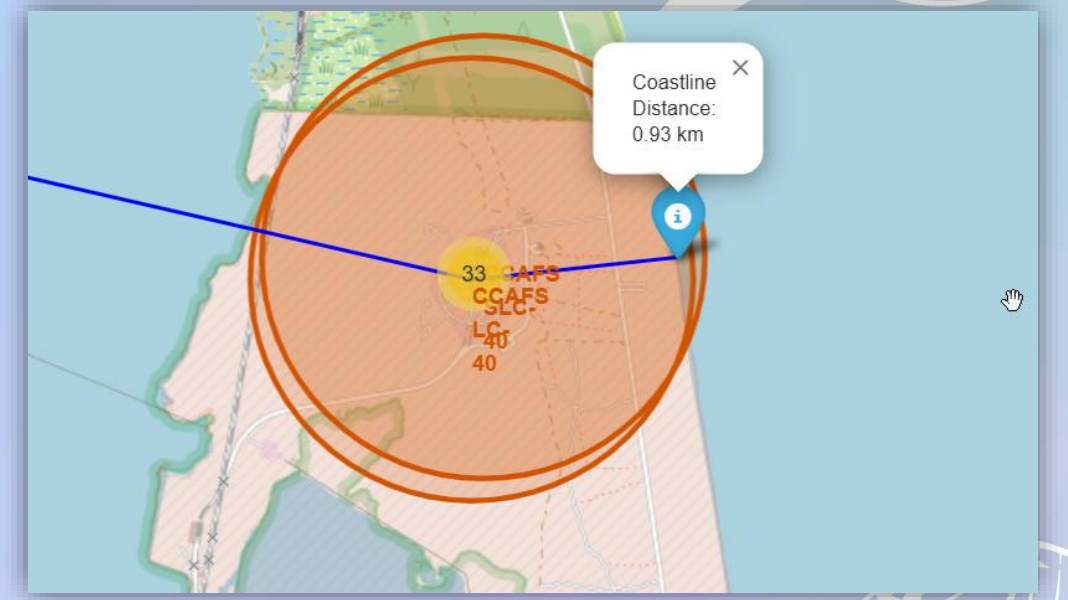


Folium Map: Markers for launch records

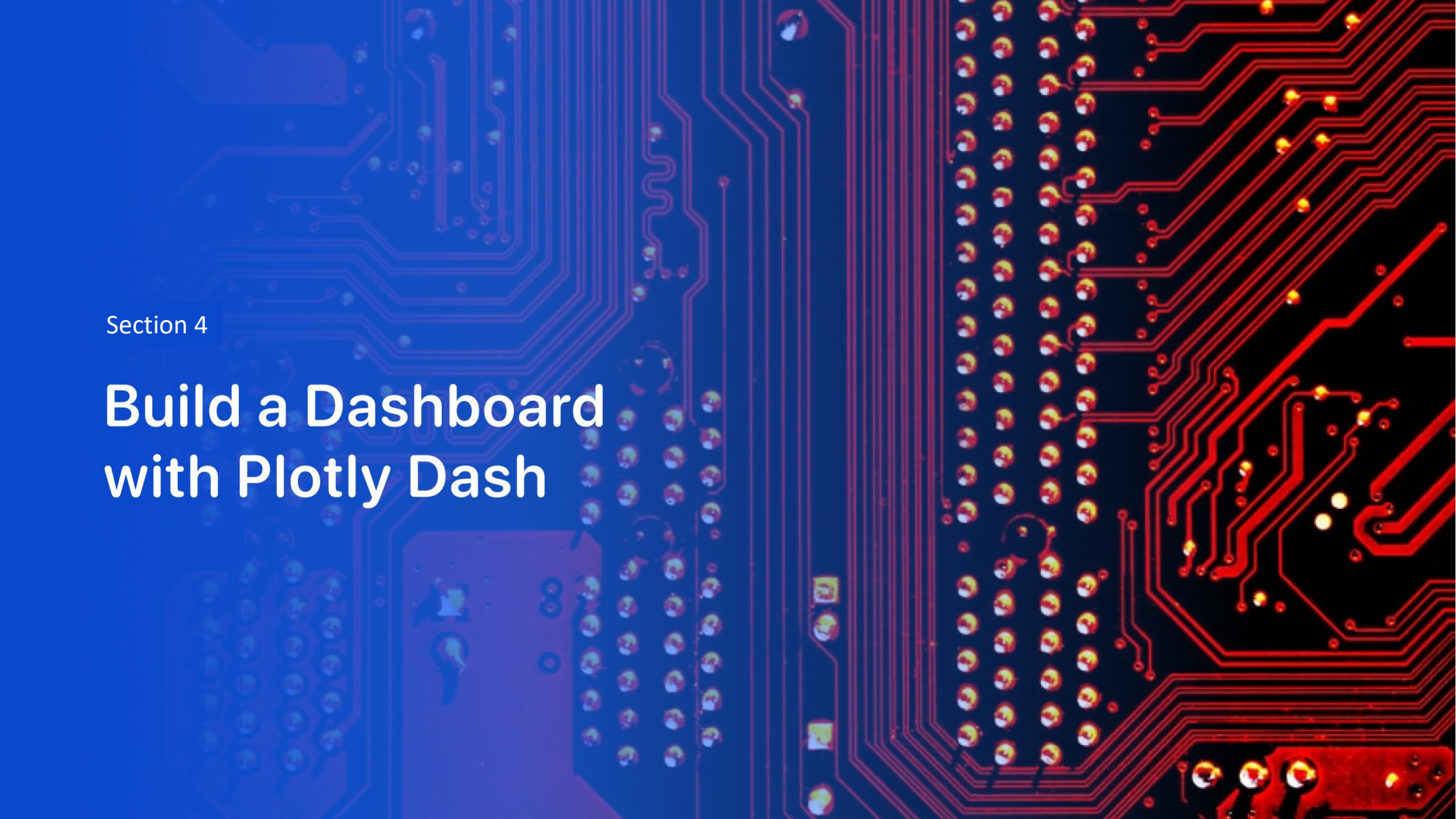


Folium Map: Proximities

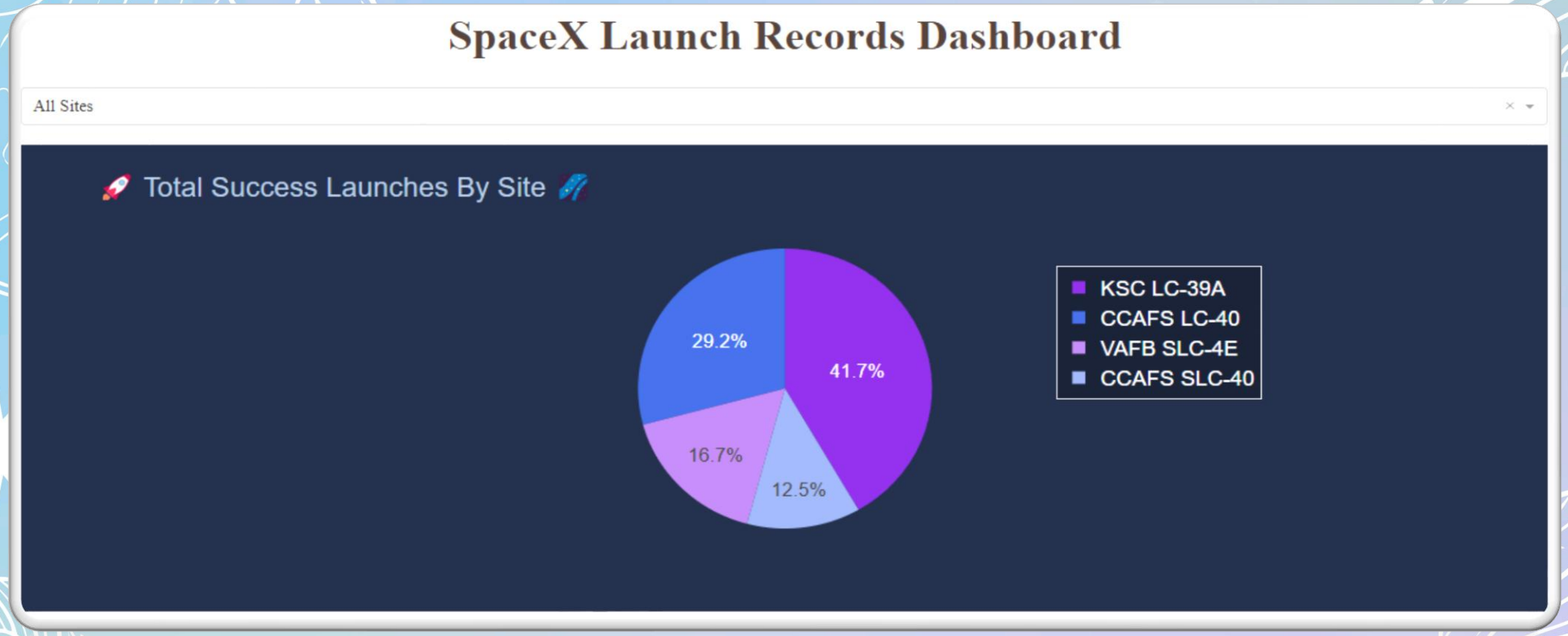




- Launch sites are located near the coastline to reduce risks in case of failure, close to railways and highways for efficient transport of rocket components, and far from cities to ensure safety

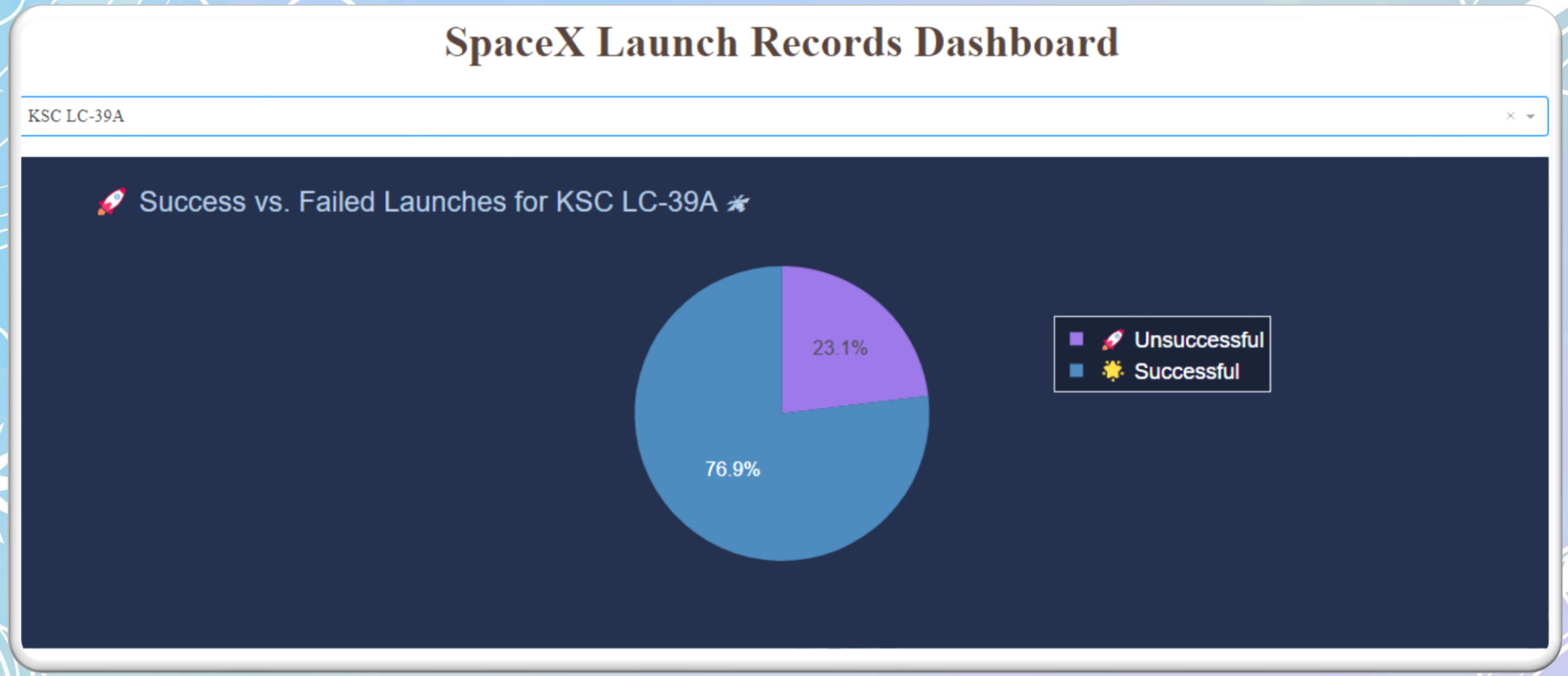

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by Site



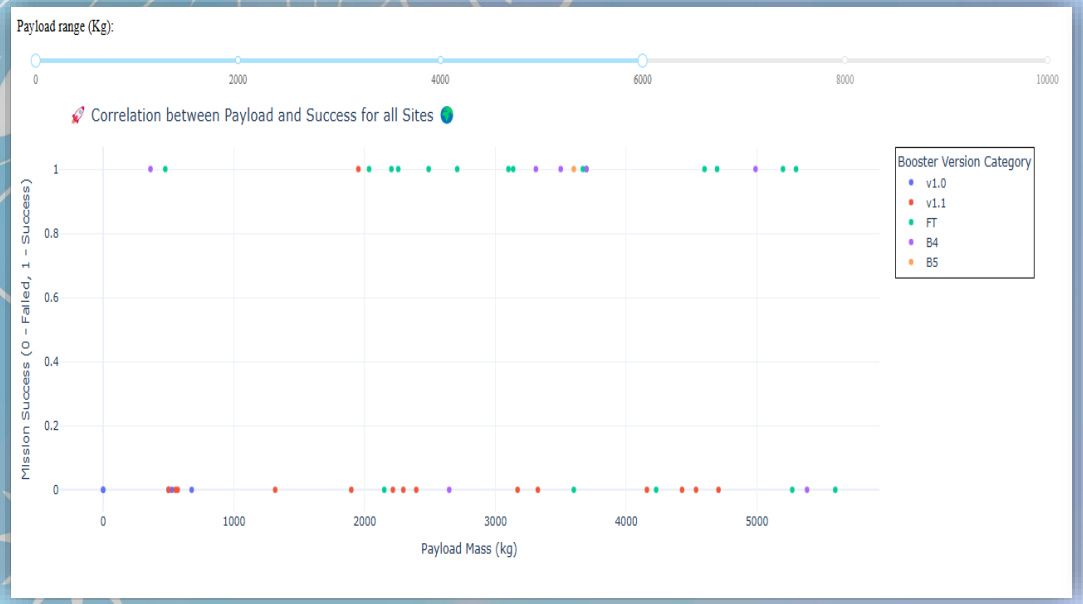
- KSC LC-39A has the highest success rate (41.7%), followed by CCAFS LC-40 (29.2%). VAFB SLC-4E and CCAFS SLC-40 contribute less but still play a role in SpaceX's launch success

Launch Success Rate at KSC LC-39A



- The KSC LC-39A launch site has a **76.9% success rate**, meaning most launches from this site are successful. However, **23.1% of launches have failed**, indicating room for improvement in reliability

Launch Success vs. Payload range

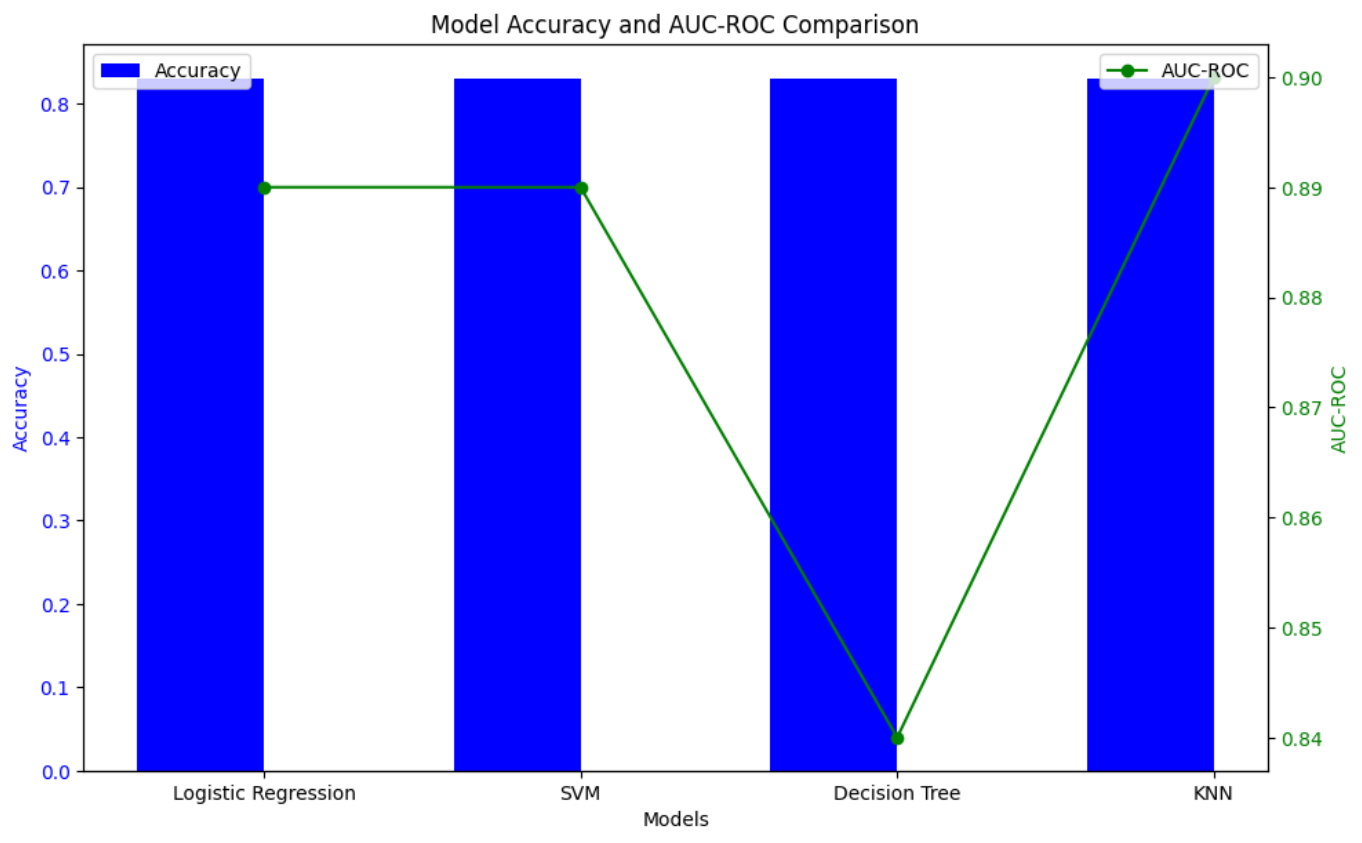


- No strong correlation between payload mass and success
- Second graph shows lower success rates
- Possible reasons: different booster versions or other

Section 5

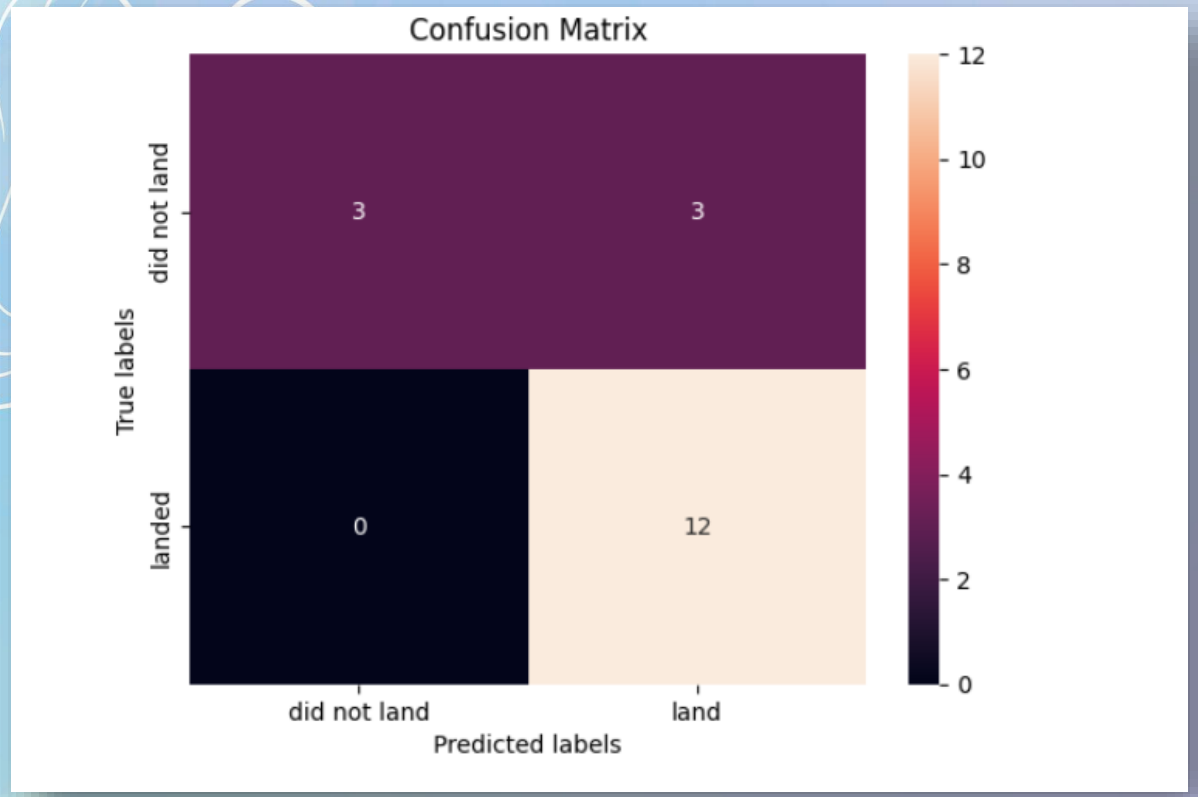
Predictive Analysis (Classification)

Classification Accuracy



- All four models (Logistic Regression, SVM, Decision Tree, and KNN) showed identical accuracy (0.83), indicating they all performed similarly in terms of classification accuracy
- KNN stood out with the highest AUC-ROC (0.8958), which suggests it has the best ability to distinguish between the classes

Confusion Matrix



- The confusion matrices for all four models (Logistic Regression, SVM, Decision Tree, and KNN) are identical, indicating that all models are making the same classification errors and correct predictions for the given test data. Specifically, the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are the same across all models.

Conclusions

- Key Metrics
  - Sensitivity for class 0 was relatively low across all models, particularly in Logistic Regression and SVM, where it was 0.50
  - F1-scores for both classes were good for all models, especially for class 1
- Model Selection
  - While accuracy was consistent across models, KNN's higher AUC-ROC gives it a slight edge in distinguishing classes
  - If class separation and distinguishing ability are critical, KNN might be the best choice
- Insights
  - Despite similar accuracy, differences in AUC-ROC and other metrics highlight that models may behave differently under various circumstances, especially when handling imbalanced classes or distinguishing subtle patterns in the data
  - The results suggest that model selection should consider not only accuracy but also how well models handle class separation and overall performance on unseen data

Thank you!