IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Mazen Hamada
29 October 2024

# Outline

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

# Executive Summary

- Summary of methodologies
- Summary of all results

# Introduction

In this Project, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. In this presentation, we will provide an overview of the problem and the tools required for the task.

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

- Collecting data through two methods:

1. Request to the SpaceX API.

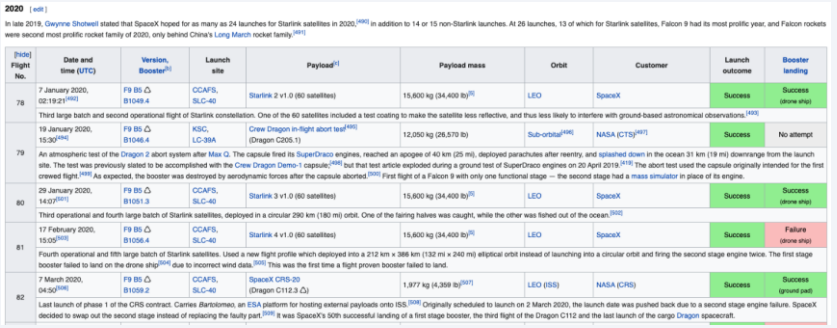
```
[6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

[7]: response = requests.get(spacex_url)
```

2. Web Scraping:

- Web scrap Falcon 9 launch records with **BeautifulSoup**:

  Extract a Falcon 9 launch records HTML table from Wikipedia

  Parse the table and convert it into a Pandas data frame



**2020** [edit]

In late 2019, Gwynne Shotwell stated that SpaceX hoped for as many as 24 launches for Starlink satellites in 2020,[490] in addition to 14 or 15 non-Starlink launches. At 26 launches, 13 of which for Starlink satellites, Falcon 9 had its most prolific year, and Falcon rockets were second most prolific rocket family of 2020, only behind China's Long March rocket family.[491]

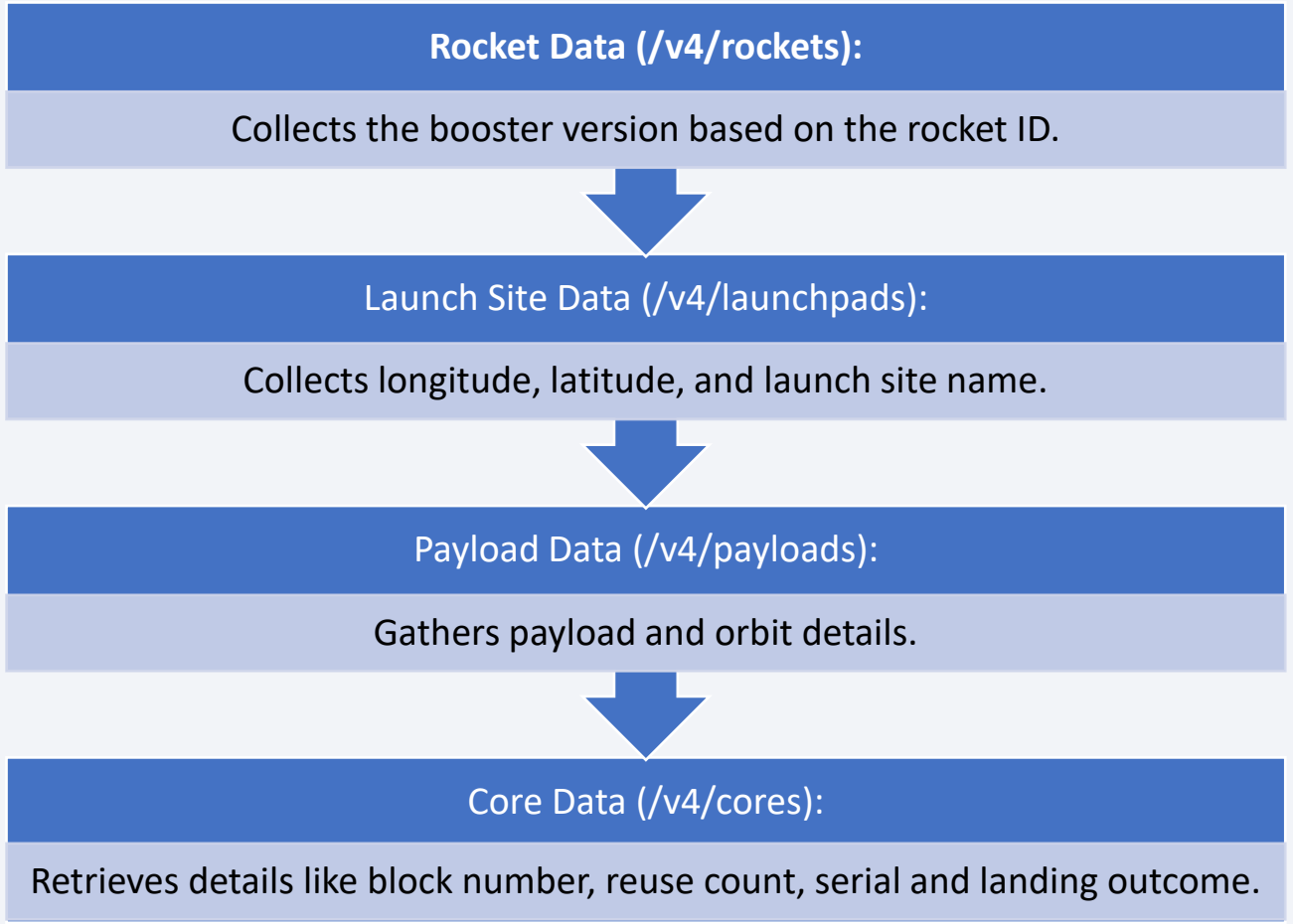
| [hide] Flight No. | Date and time (UTC) | Version, Booster[a] | Launch site | Payload[c] | Payload mass | Orbit | Customer | Launch outcome | Booster landing |
|---|---|---|---|---|---|---|---|---|---|
| 78 | 7 January 2020, 02:19:21[492] | F9 B5 △ B1049.4 | CCAFS, SLC-40 | Starlink 2 v1.0 (60 satellites) | 15,600 kg (34,400 lb)[D] | LEO | SpaceX | Success | Success (drone ship) |
| | Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493] | | | | | | | | |
| 79 | 19 January 2020, 15:30[494] | F9 B5 △ B1046.4 | KSC, LC-39A | Crew Dragon in-flight abort test[495] (Dragon C205.1) | 12,050 kg (26,570 lb) | Sub-orbital[496] | NASA (CTS)[497] | Success | No attempt |
| | An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule,[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[419] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage — the second stage had a mass simulator in place of its engine. | | | | | | | | |
| 80 | 29 January 2020, 14:07[501] | F9 B5 △ B1051.3 | CCAFS, SLC-40 | Starlink 3 v1.0 (60 satellites) | 15,600 kg (34,400 lb)[D] | LEO | SpaceX | Success | Success (drone ship) |
| | Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502] | | | | | | | | |
| 81 | 17 February 2020, 15:05[503] | F9 B5 △ B1056.4 | CCAFS, SLC-40 | Starlink 4 v1.0 (60 satellites) | 15,600 kg (34,400 lb)[D] | LEO | SpaceX | Success | Failure (drone ship) |
| | Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land. | | | | | | | | |
| 82 | 7 March 2020, 04:50[506] | F9 B5 △ B1059.2 | CCAFS, SLC-40 | SpaceX CRS-20 (Dragon C112.3 △) | 1,977 kg (4,359 lb)[507] | LEO (ISS) | NASA (CRS) | Success | Success (ground pad) |
| | Last launch of phase 1 of the CRS contract. Carries *Bartolomeo*, an ESA platform for hosting external payloads onto ISS.[508] Originally scheduled to launch on 2 March 2020, the launch date was pushed back due to a second stage engine failure. SpaceX decided to swap out the second stage instead of replacing the faulty part.[509] It was SpaceX's 50th successful landing of a first stage booster, the third flight of the Dragon C112 and the last launch of the cargo Dragon spacecraft. | | | | | | | | |

# Data Collection – SpaceX API

- Data collection with SpaceX REST calls flowcharts

- GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

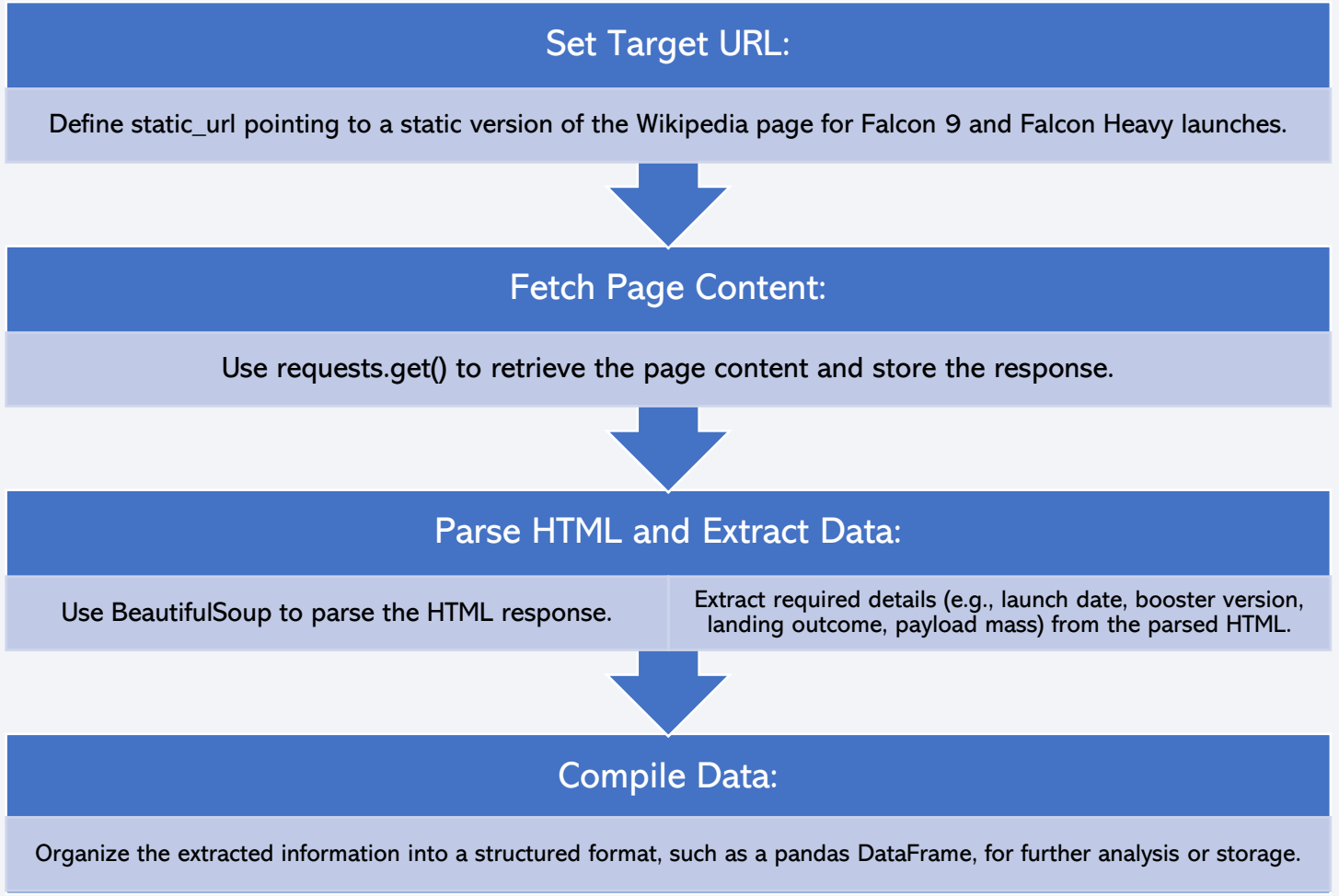


# Data Collection - Scraping

- Web scraping process flowcharts

- GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

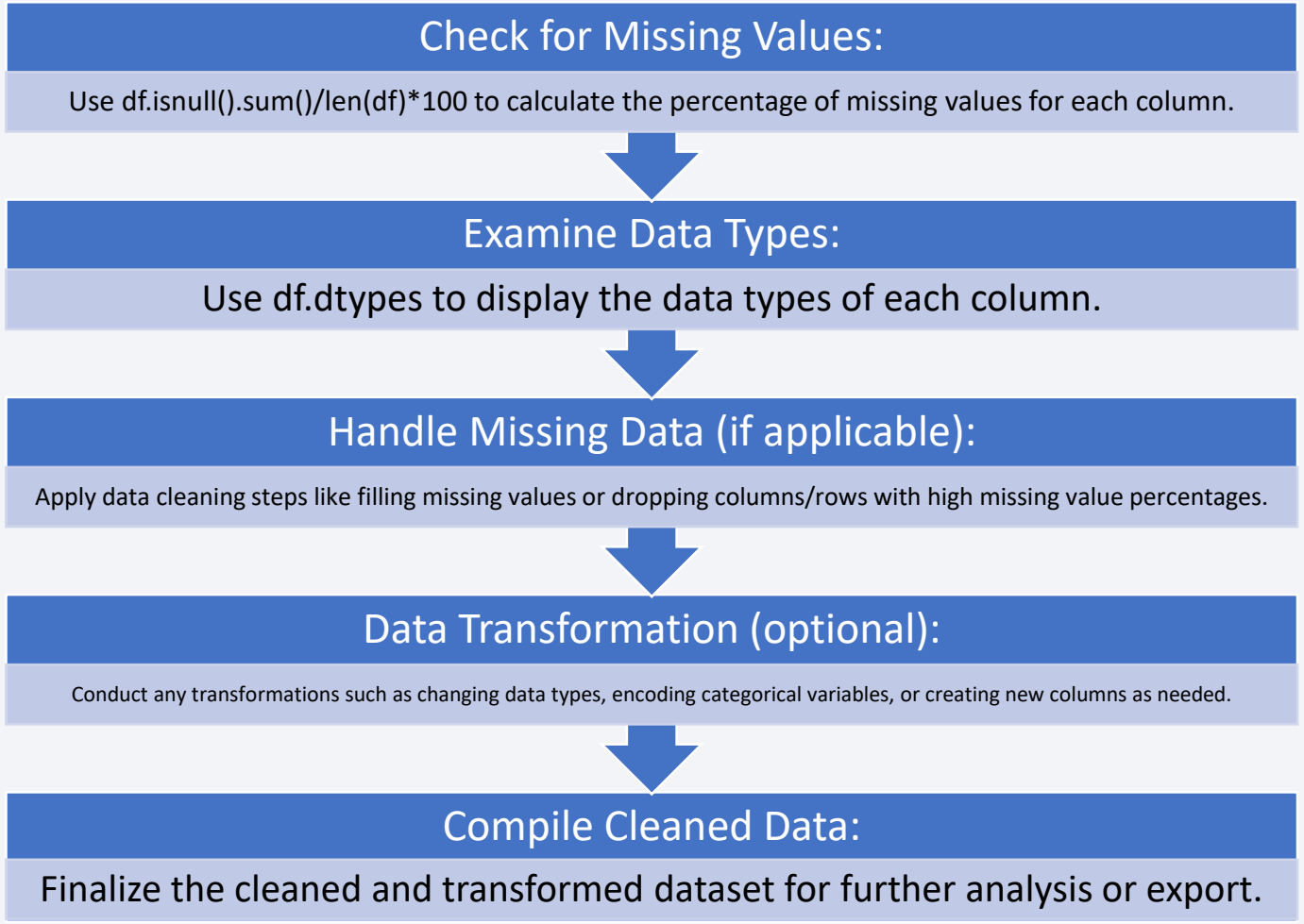


# Data Wrangling

- Data wrangling process flowcharts.

- Add the GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



# EDA with Data Visualization

**Plots/Charts used:**

1. A **Scatter plot** showing the relationship between **Flight Number** and **Launch Site.**
2. A **Scatter plot** showing the relationship between **Payload Mass** and **Launch Site.**
3. A **Bar Chart** showing the relationship between **Success Rate** of **Orbit Type.**
4. A **Scatter Plot** showing the relationship between **Flight Number** and **Orbit Type.**
5. A **scatter plot** showing the relationship between **Payload Mass** and **Orbit Type.**
6. A **Line Chart** showing the **Launch Success Yearly Trend.**

GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb

# EDA with SQL

**Summary of the SQL queries performed:**

- Created a new table SPACEXTABLE with data from SPACEXTBL where the date is not null.
- Selected distinct launch sites from SPACEXTBL, ordering results.
- Filtered launches from SPACEXTBL with launch sites starting with 'CCA', limiting the results to 5.
- Calculated the total payload for records with payload description containing 'CRS'.
- Found the average payload for records where the booster version is 'F9 v1.1'.
- Retrieved the earliest successful landing on a ground pad.
- Selected distinct booster versions for payloads between 4000 and 6000 kg with successful drone ship landings.
- Counted and grouped launches by mission outcomes.
- Retrieved distinct booster versions for the maximum payload mass.
- Extracted records of failures on a drone ship in 2015, adding a month name column based on the date.

## GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

1. **Markers**: Placed at specific coordinates to represent points of interest, such as launch sites. Each marker typically has a popup or tooltip to display additional information when clicked. Adding markers allows viewers to easily identify and obtain information on exact locations.

2. **Circles**: Used to draw areas around certain points to represent zones of influence or safety distances around the launch sites. Circles visually emphasize the proximity of other points to these central locations, making it easier to understand spatial relationships.

3. **Polylines**: Created between locations to illustrate paths, distances, or routes. This can help indicate directionality or show how close other sites are to each other.

- These map objects enhance the map's clarity by visually distinguishing between locations, distances, and areas of influence, making the data easier to interpret spatially.Explain why you added those objects

GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

1. A Dropdown (input) component:
   - A dropdown list to enable Launch Site selection, The default select value is for ALL sites.
2. Pie chart (output) component:
   - A pie chart to show the total successful launches count for all sites, If a specific launch site was selected, show the Success vs. Failed counts for the site.
3. Range slider (input) component:
   - A slider to select payload range.
4. Scatter chart (output) component:
   - A scatter chart to show the correlation between payload and launch success.

- GitHub Notebook URL:

https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

# Predictive Analysis (Classification)

## Model Development Process:

### 1- Data Preprocessing:

- Data Cleaning.
- Feature Engineering.
- Splitting Data into Training and Testing Sets.

### 2- Model Selection and Building:

- Selected models: Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and K-Nearest Neighbors (KNN).
- Initial Training on Base Models.

### 3- Model Evaluation:

- Metrics used: Accuracy.
- Confusion Matrix Analysis.

### 4- Model Improvement:

- Hyperparameter Tuning (using GridSearchCV or RandomizedSearchCV).
- Feature Selection/Engineering.
- Cross-Validation.

### 5- Best Model Selection:

- Comparing Metrics across Models.
- Final Model Selection based on Highest Performance.

- GitHub Notebook URL:

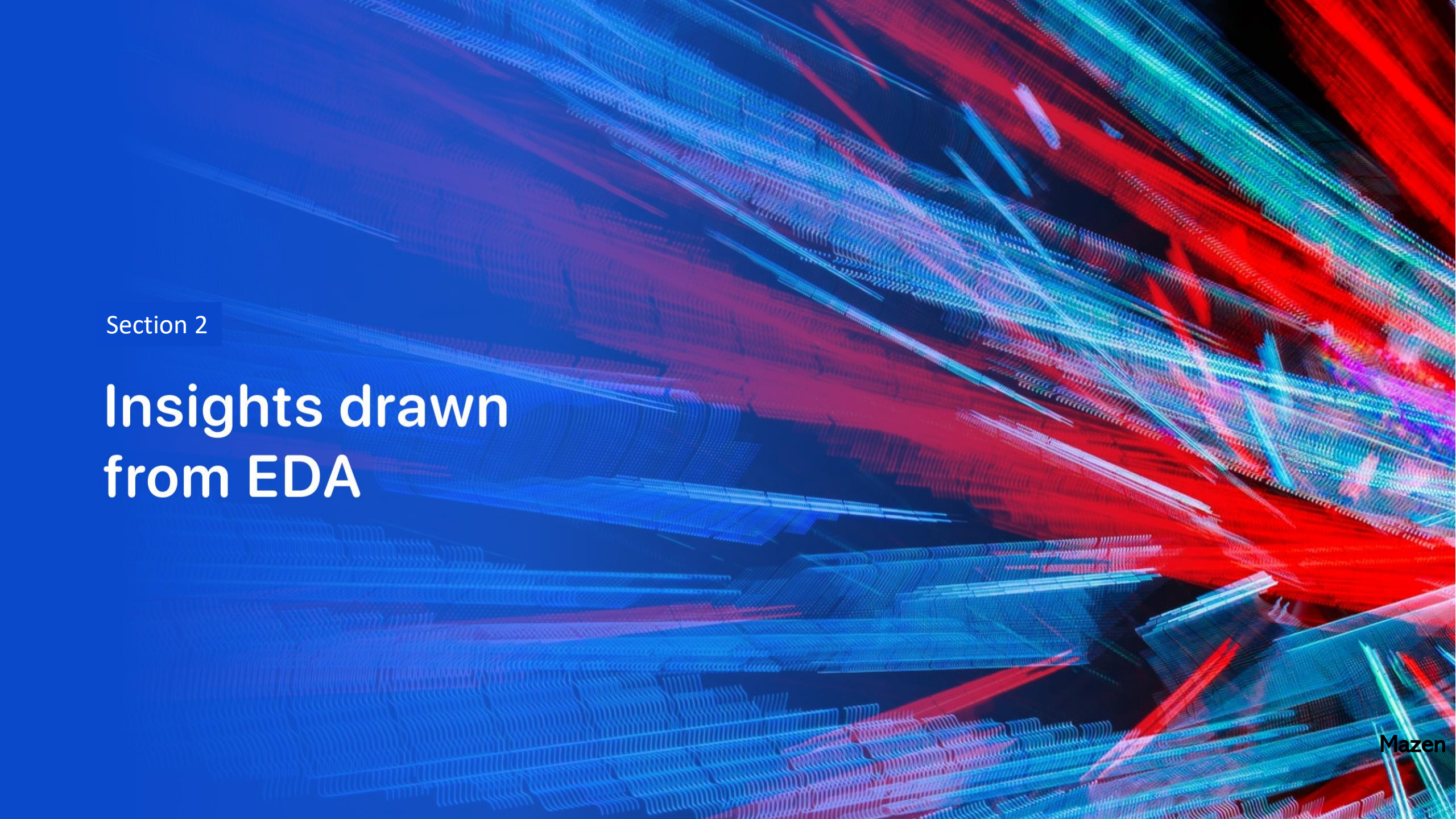
https://github.com/MazenHamada/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results

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

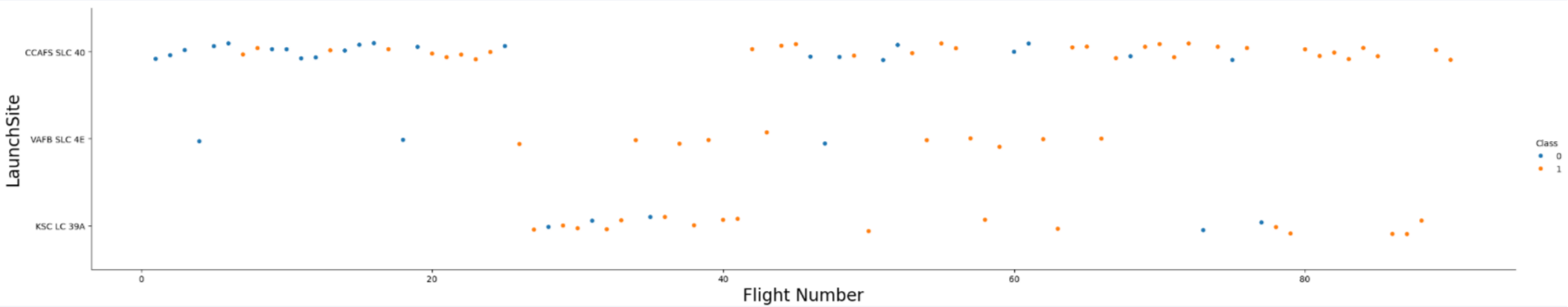
Section 2

# Insights drawn from EDA



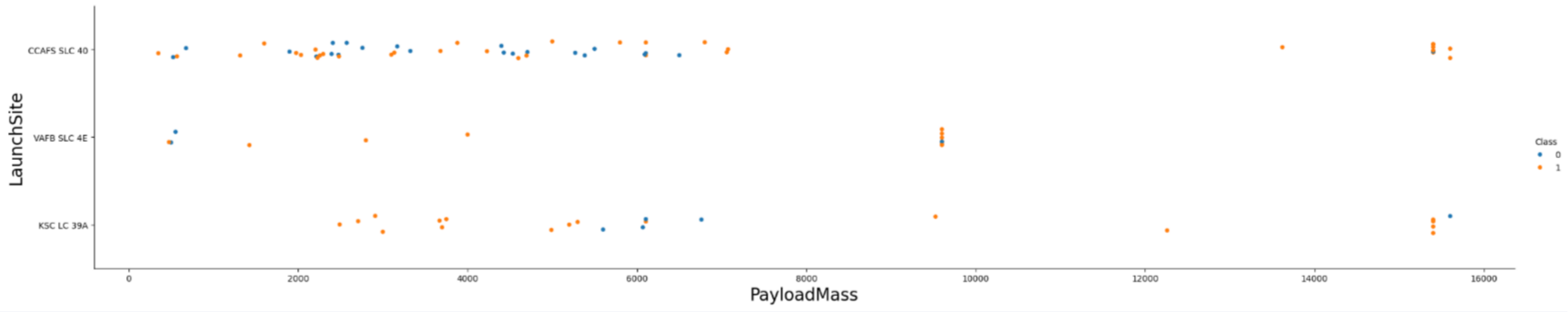
Mazen

# Flight Number vs. Launch Site



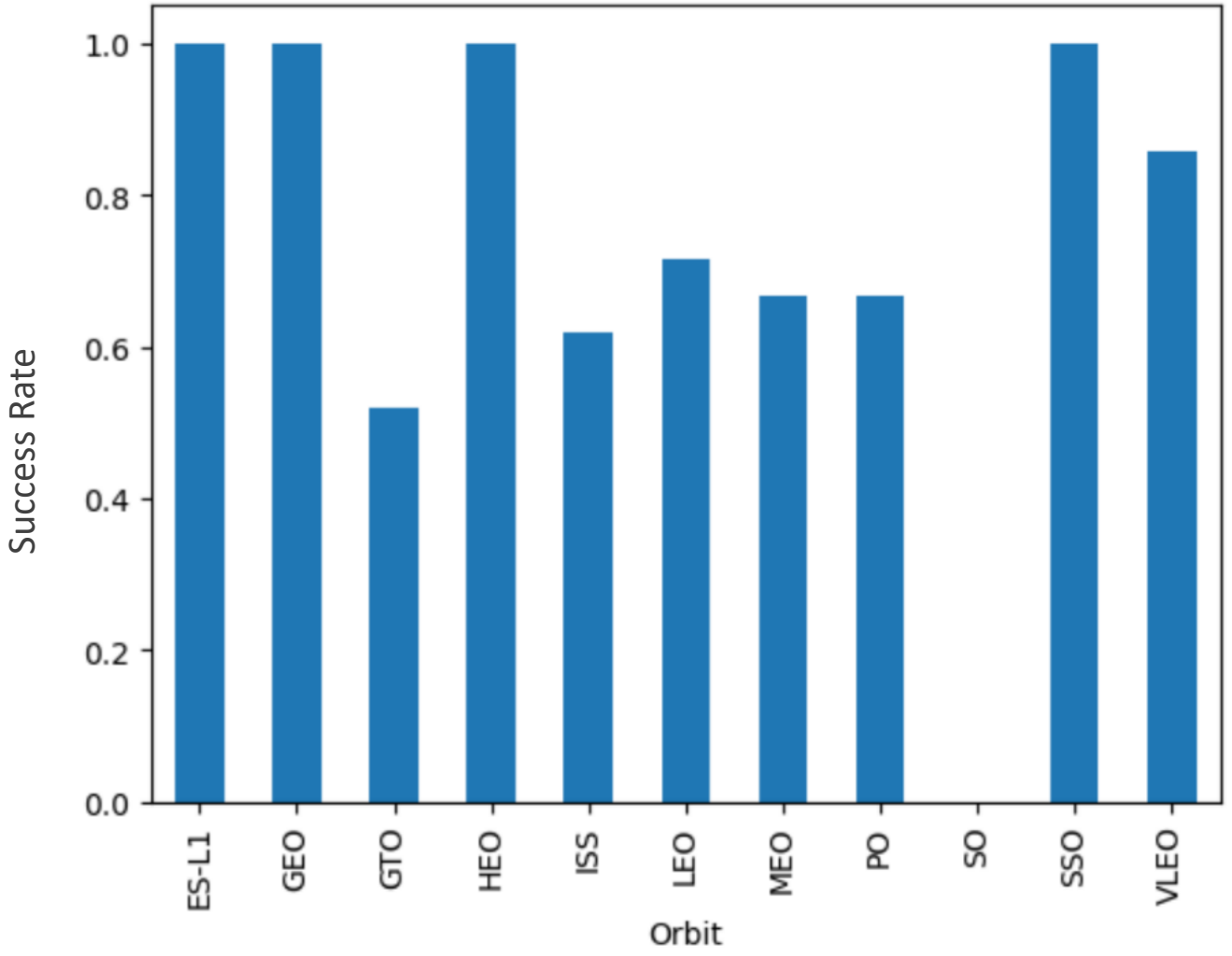

The bigger the flight number the more successful landings.

# Payload vs. Launch Site



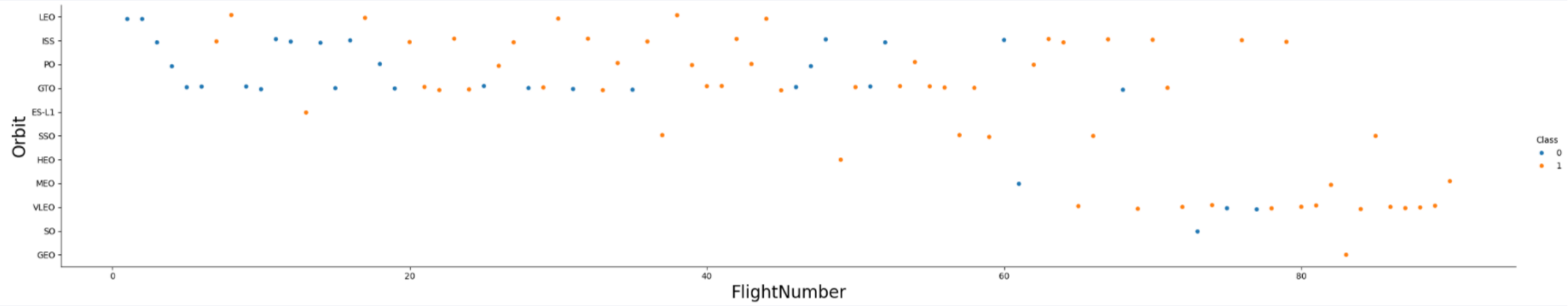

For the VAFB-SLC launchsite there are no rockets launched for heavy pay load mass(greater than 10000).

# Success Rate vs. Orbit Type



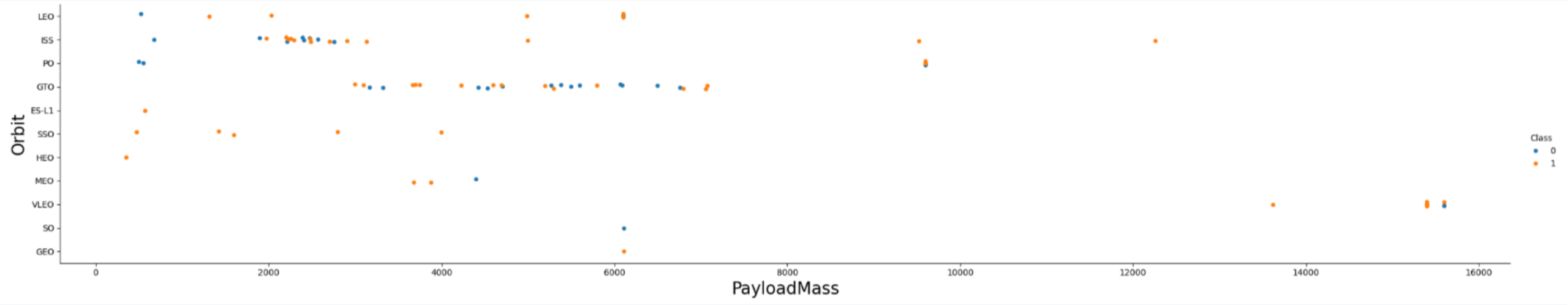

(ES-L1), (GEO), (HEO) and (SSO) have high success rate while (SO) has low success rate.

# Flight Number vs. Orbit Type



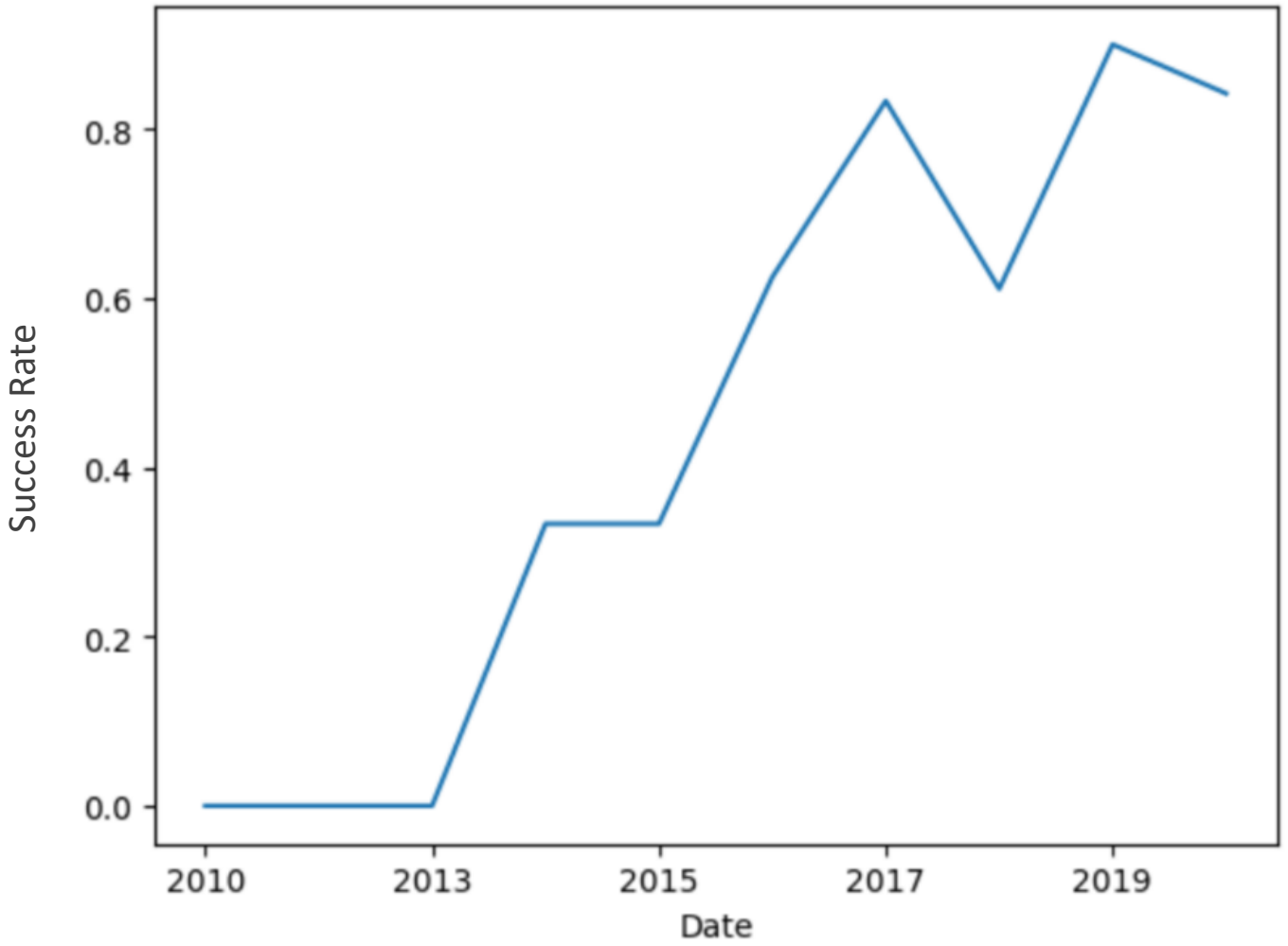

You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

# 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 success rate since 2013 kept increasing till 2020.

# All Launch Site Names

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

In [23]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

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

Out[23]:

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

# Launch Site Names Begin with 'CCA'

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

In [24]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[24]:

| 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 [25]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

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

Out[25]:

| TOTAL_PAYLOAD |
|---|
| 111268 |

# Average Payload Mass by F9 v1.1

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

In [26]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

Out[26]:

| AVG_PAYLOAD |
|---|
| 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 [28]:
```
%sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

Out[28]:

| FIRST_SUCCESS_GP |
|---|
| 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

[29]:
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING_OUTCOME = 'Success (drone ship)';
```

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

[29]:

| 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

[30]:
```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

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

[30]:

| Mission_Outcome | QTY |
|---|---|
| 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

[31]:
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY BOOSTER_VERSION;
```

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

[31]:

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

# 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.**

[34]:
```
%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'
```

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

[34]:

| Month | Booster_Version | Launch_Site | Landing_Outcome |
|---|---|---|---|
| January | F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) |
| April | F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) |

# 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.

[37]:
```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY QTY DESC;
```

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

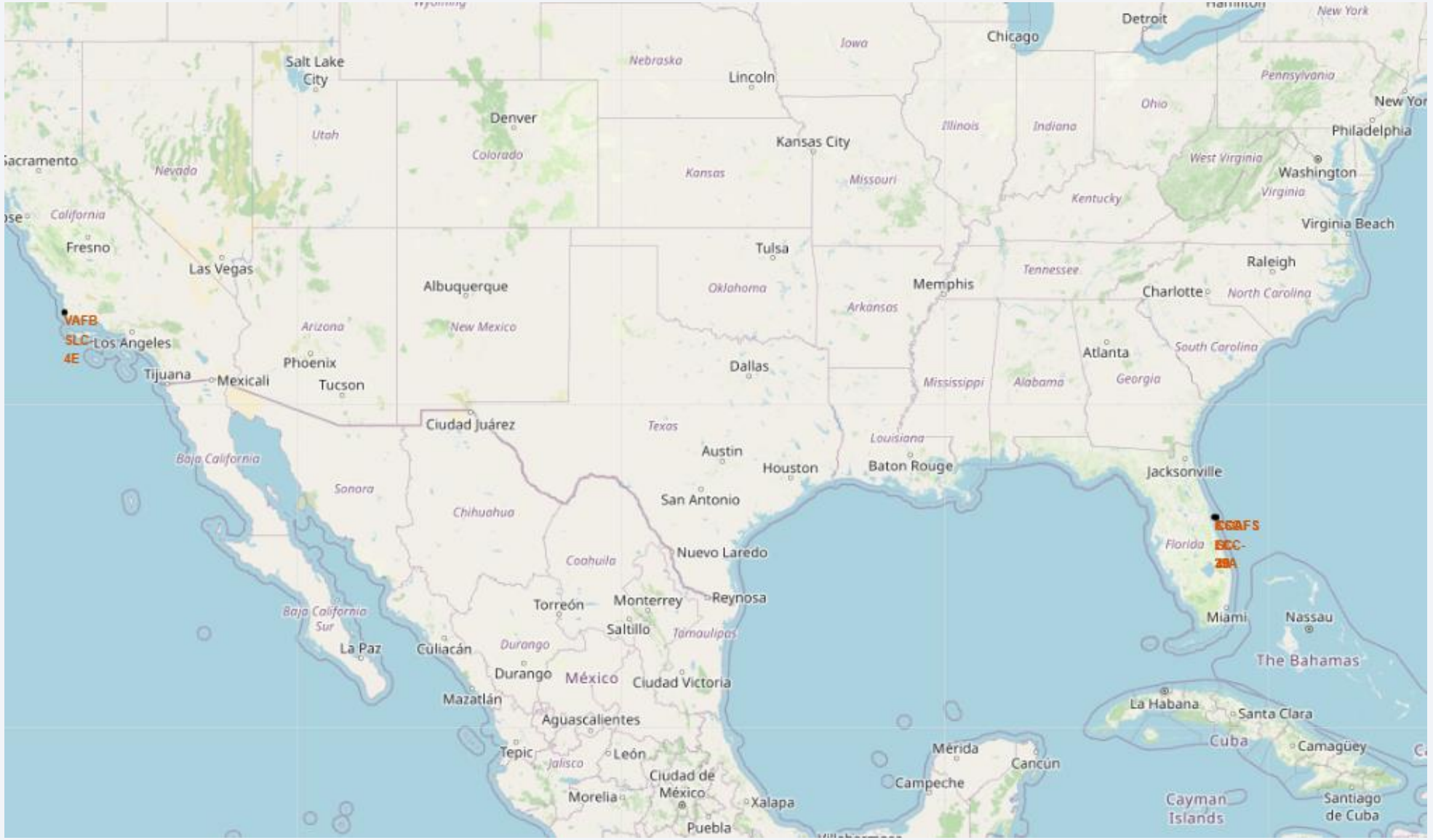
[37]:

| Landing_Outcome | QTY |
|---|---|
| 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 |

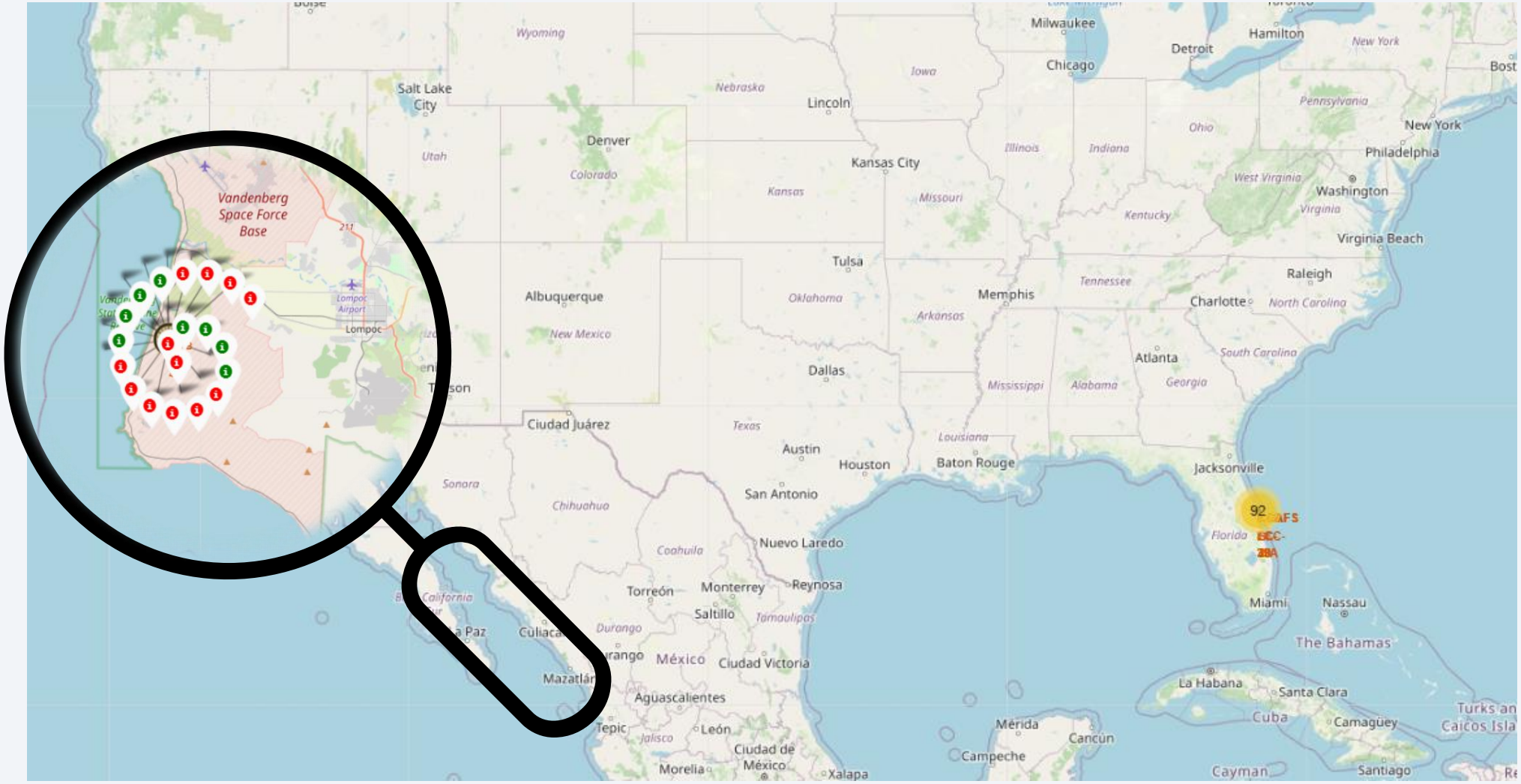
Section 3

# Launch Sites Proximities Analysis

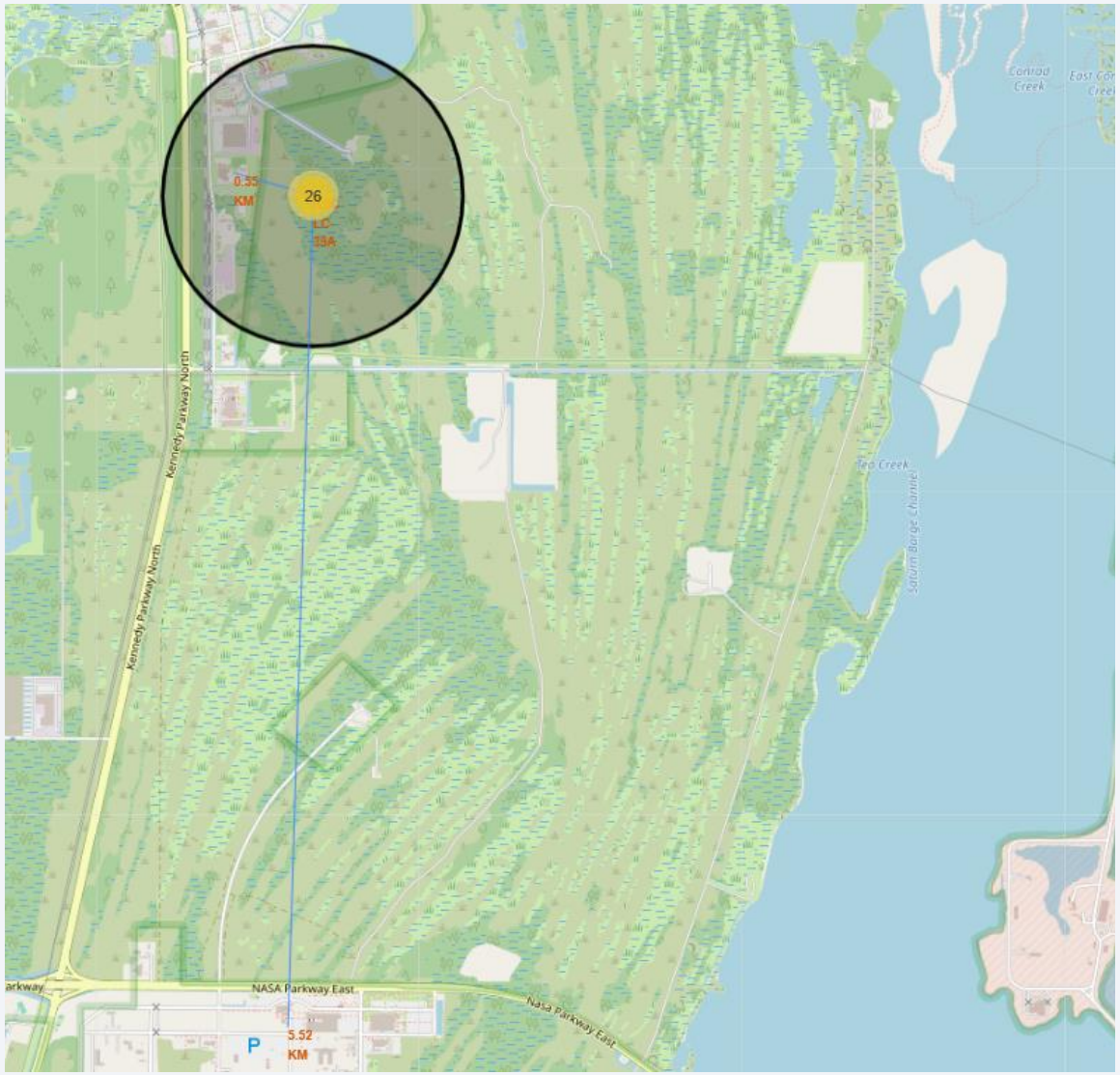
# Launch Sites Location



# Launch Results for each Launch Site

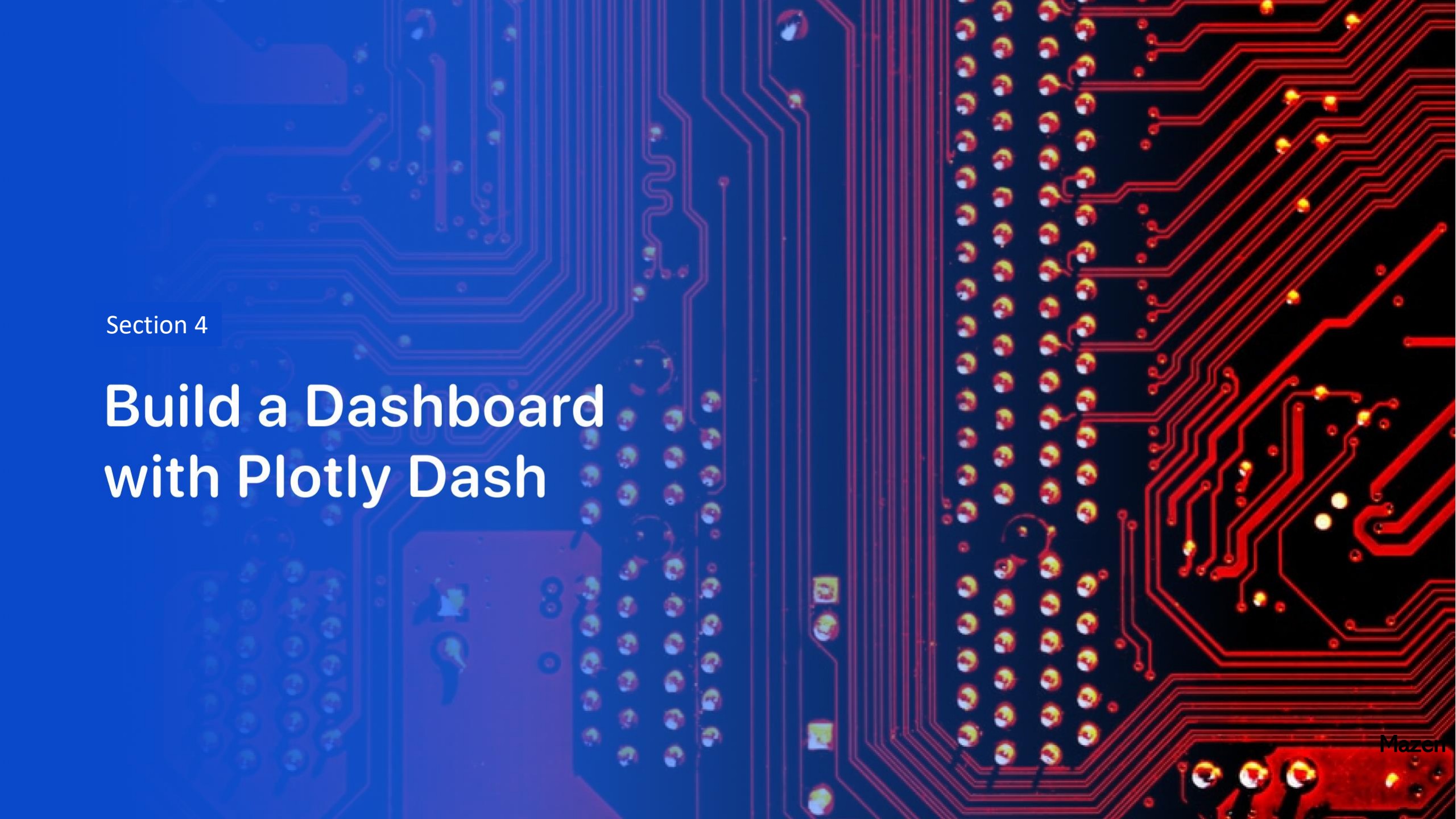


# Launch Sites to its Proximities, with Distance Calculated and Displayed



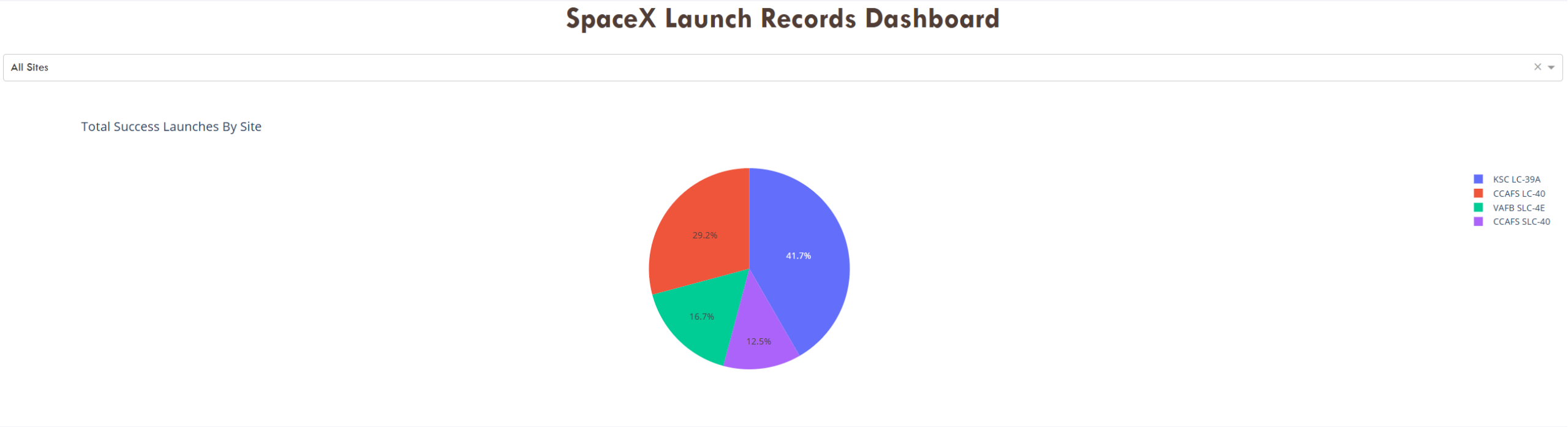
Section 4

# Build a Dashboard with Plotly Dash

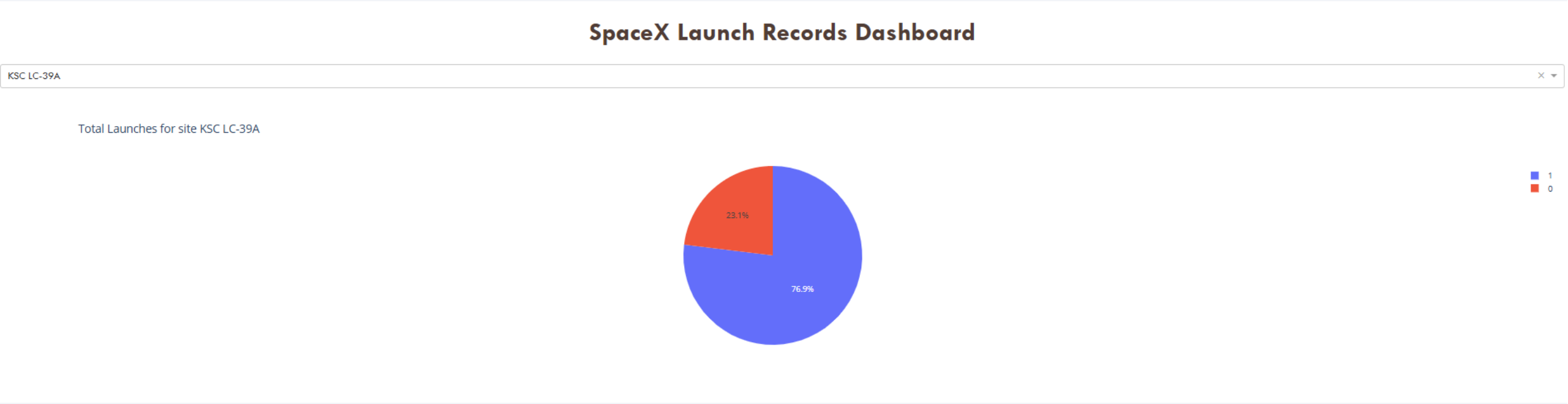


Mazen

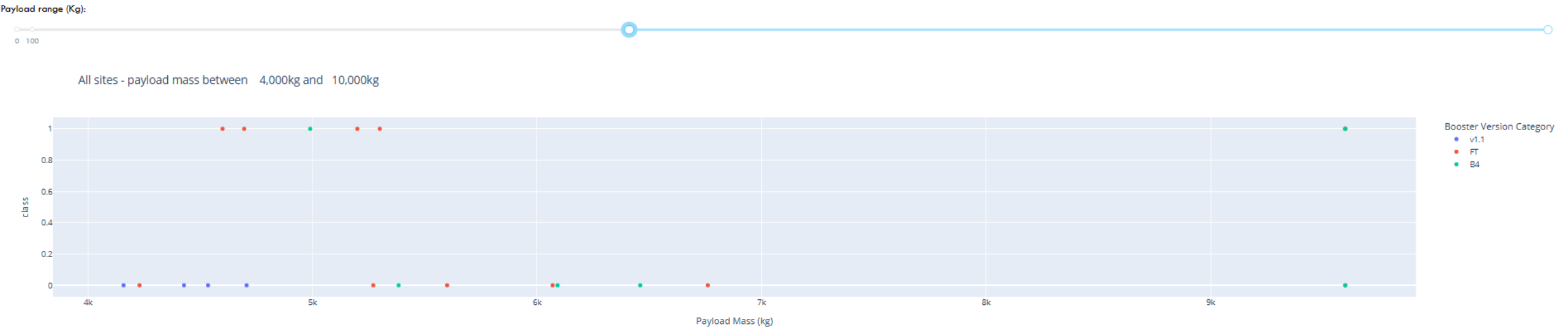
# Launch Success Count for All Sites



# Launch Site with Highest Launch Success Ratio



# Payload vs. Launch Outcome Scatter Plot for all Sites, with Different Payload

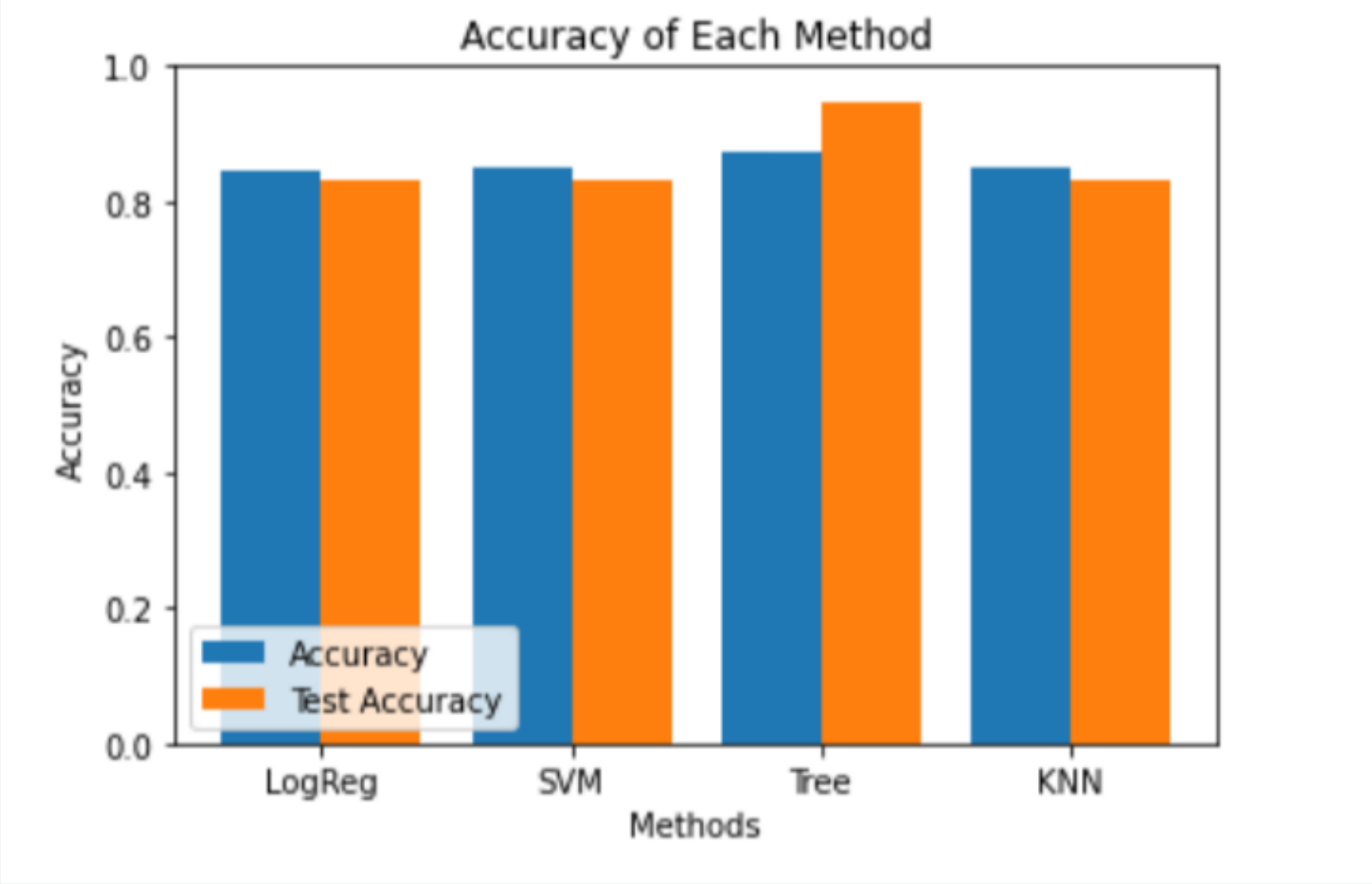



Section 5

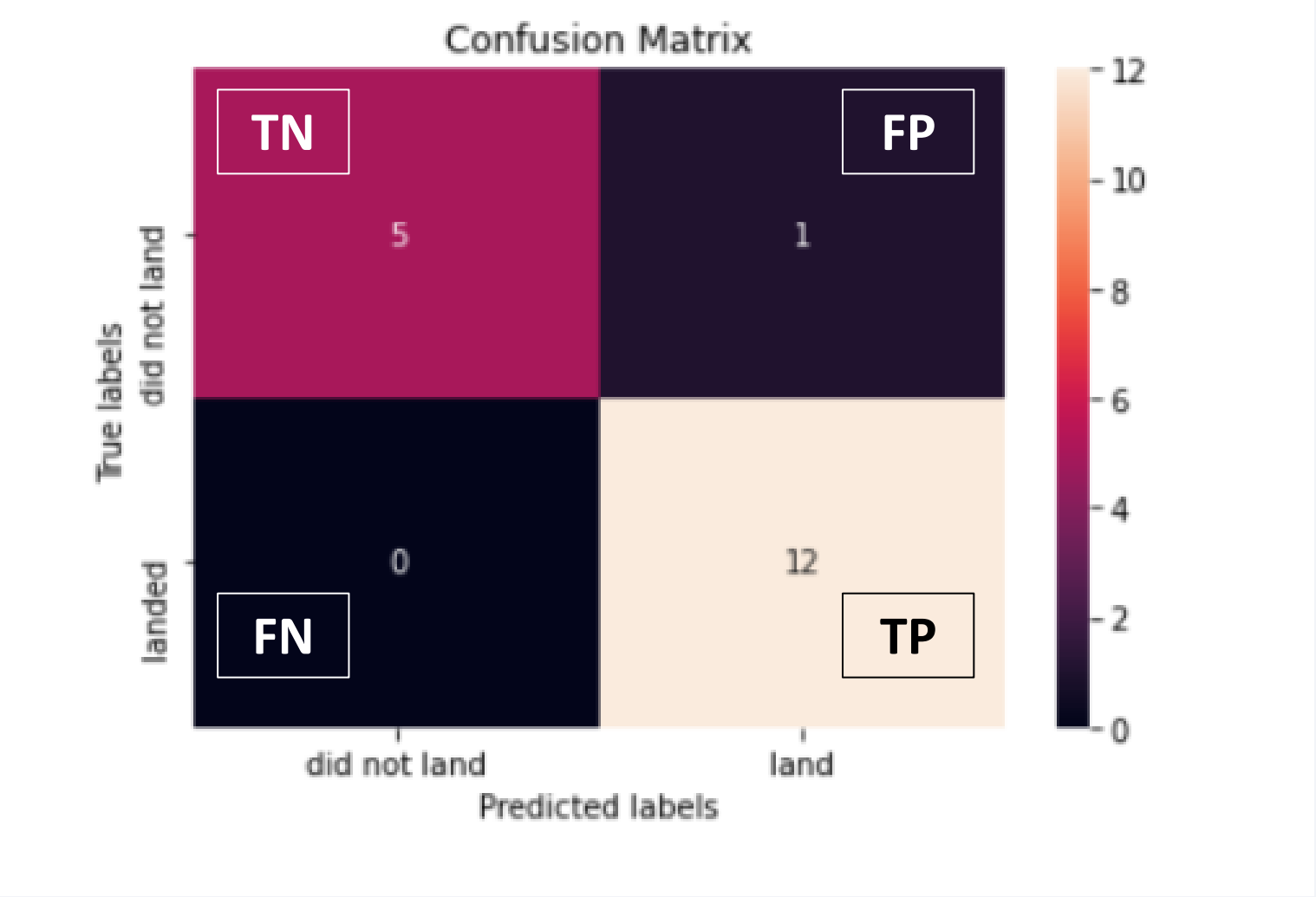
# Predictive Analysis (Classification)

Mazen

# Classification Accuracy



# Confusion Matrix



# Conclusions

- Described different data collection methodologies.
- Performed data wrangling.
- Performed exploratory data analysis (EDA) using visualization and SQL.
- Performed interactive visual analytics using Folium and Plotly Dash.
- Performed predictive analysis using classification models.

# 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!

Mazen