IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Leonardo Francelino
March 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wranglin
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using calls on SpaceX API
- 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

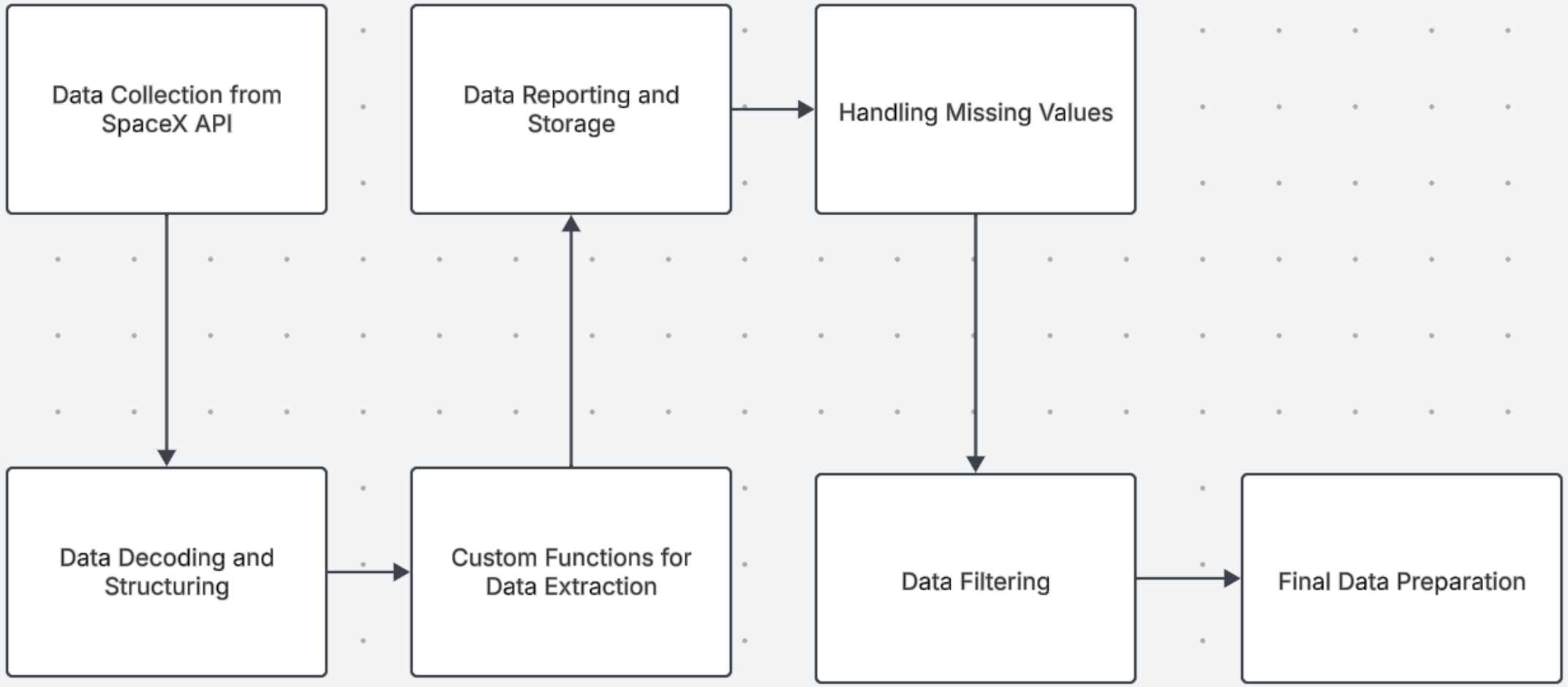
API call

- Began by **requesting rocket launch data from the SpaceX API**, which provided detailed information on launches. The API response, in JSON format, was decoded using Python's `.json()` method and transformed into a structured dataframe with `.json_normalize()`.
- To focus on key details like payload mass and launch site, we applied **custom functions** to extract and organize the data into a dictionary. This data was then saved to a **CSV file** for easy access and reproducibility.
- During cleaning, we addressed missing values in the **"Payload Mass" column** by replacing them with the column's mean. We also **filtered the dataframe** to include only **Falcon 9 launches**, ensuring the dataset was relevant and ready for analysis.
- Finally, we created a **clean and structured dataframe** from the dictionary, forming the foundation for our exploratory data analysis and machine learning modeling.

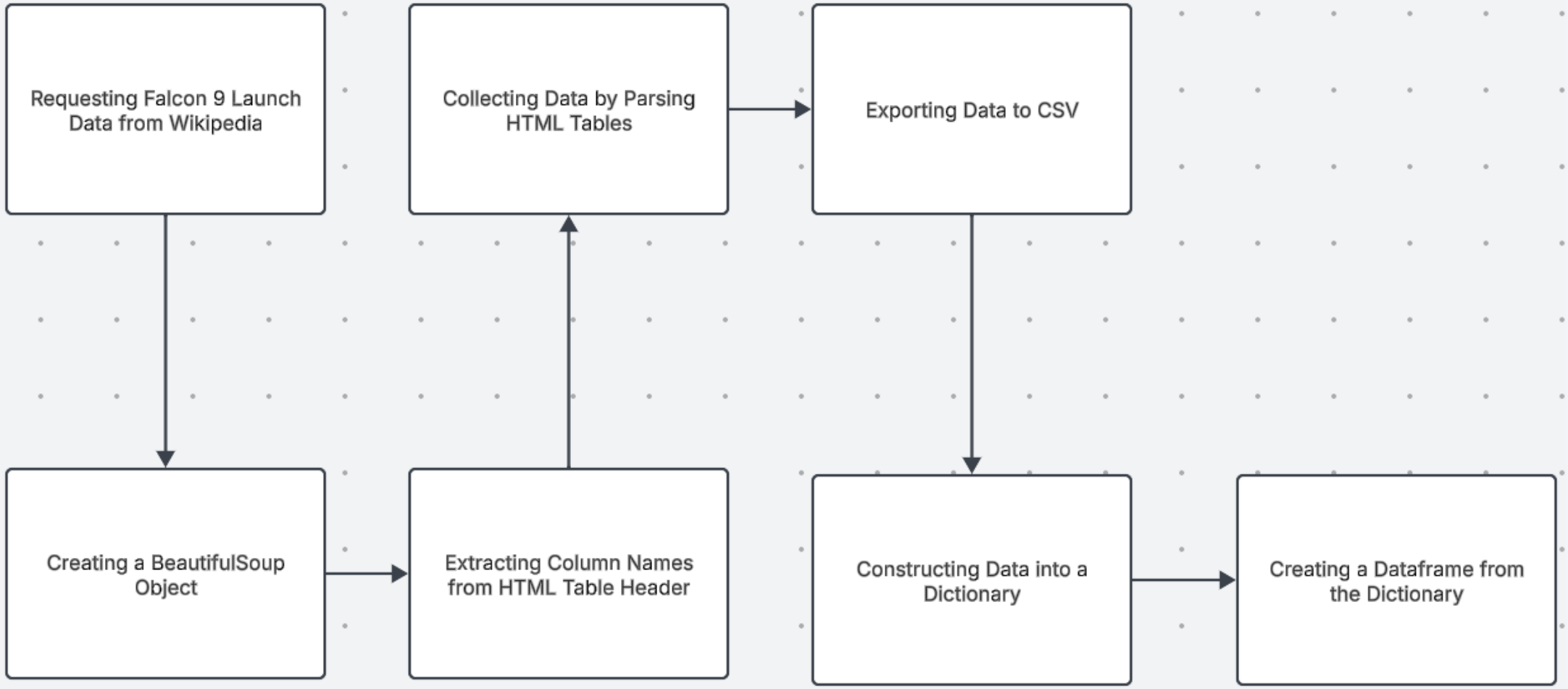
Web Scraping

- The data collection process also involved **web scraping Falcon 9 launch data from Wikipedia**. Using Python's `requests` library, we fetched the HTML content of the Wikipedia page containing the launch data. This HTML content was then parsed using **BeautifulSoup**, a Python library for extracting data from HTML and XML files.
- We began by **extracting all column names** from the HTML table header to understand the structure of the data. Next, we **collected the actual data** by parsing the HTML tables row by row. This step ensured that all relevant launch details, such as date, payload, and launch site, were captured.
- Once the data was collected, we **constructed it into a dictionary** for easier manipulation and storage. This dictionary was then used to **create a structured dataframe**, which allowed for efficient data analysis. Finally, to ensure the data could be reused and shared, we **exported it to a CSV file**.

Data Collection – SpaceX API



Data Collection - Scraping



Data Wrangling

The data wrangling process began with **understanding the landing outcomes** in the dataset. These outcomes included:

- **True Ocean** (successful ocean landing) and **False Ocean** (unsuccessful ocean landing).
- **True RTLS** (successful ground pad landing) and **False RTLS** (unsuccessful ground pad landing).
- **True ASDS** (successful drone ship landing) and **False ASDS** (unsuccessful drone ship landing).

To prepare the data for machine learning, we **converted these outcomes into training labels**. A label of **"1"** was assigned for successful landings, while **"0"** was used for unsuccessful ones. This binary classification simplified the dataset for model training.Next, we performed **exploratory data analysis (EDA)** to gain insights into the data. This included:

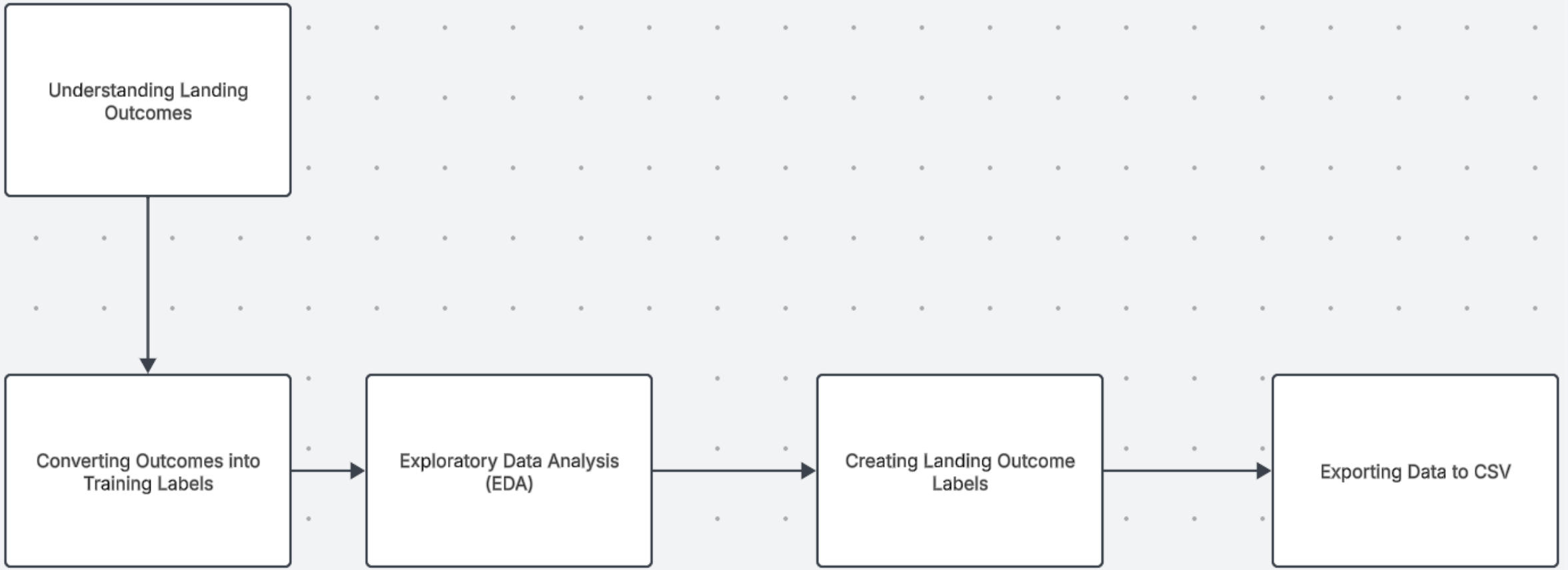
Calculating the **number of launches per site** to understand launch site activity.

Analyzing the **number and occurrence of each orbit type** to identify trends.

Examining the **number and occurrence of mission outcomes per orbit type** to uncover patterns.

From the "Outcome" column, we **created landing outcome labels** to serve as the target variable for our machine learning model. Finally, the wrangled data was **exported to a CSV file** for further analysis and modeling.

Data Wrangling



EDA with Data Visualization

- During the **Exploratory Data Analysis (EDA)**, we used **data visualization** to uncover patterns and relationships in the dataset. Several charts were plotted to analyze different aspects of the data:
- **Scatter Plots**:
  - **Flight Number vs. Payload Mass**: To explore the relationship between the sequence of launches and the mass of the payload.
  - **Flight Number vs. Launch Site**: To analyze how launch activity varied across different sites over time.
  - **Payload Mass vs. Launch Site**: To understand how payload mass correlates with specific launch sites.
  - **Payload Mass vs. Orbit Type**: To examine the relationship between payload mass and the type of orbit.
- **Bar Charts**:
  - **Orbit Type vs. Success Rate**: To compare the success rates of different orbit types.
  - These charts helped visualize comparisons among discrete categories and their measured values.
- **Line Charts**:
  - **Success Rate Yearly Trend**: To track the trend of mission success rates over time.
  - The **scatter plots** were particularly useful for identifying potential relationships between variables, which could be leveraged in machine learning models. **Bar charts** provided clear comparisons among categories, while **line charts** highlighted trends over time, such as improvements in success rates.

EDA with SQL

- **Unique Launch Sites**
  - Displaying the names of unique launch sites.
- **Filtering Launch Sites**
  - Displaying 5 records where launch sites begin with 'CCA'.
- **Payload Mass Analysis**
  - Total payload mass carried by NASA (CRS) boosters.
  - Average payload mass carried by booster version F9 v1.1.
- **Successful Landing Outcomes**
  - Date of the first successful ground pad landing.
- **Filtering Boosters**
  - Names of boosters with successful drone ship landings and payload mass between 4000 and 6000.
- **Mission Outcomes**
  - Total number of successful and failed missions.
- **Maximum Payload Mass**
  - Names of booster versions that carried the maximum payload mass.
- **Failed Drone Ship Landings**
  - Failed landing outcomes in drone ships for 2015, including booster versions and launch site names.
- **Ranking Landing Outcomes**
  - Ranking the count of landing outcomes (e.g., Failure (drone ship) or Success (ground pad)) between 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

We built an **interactive map** using the **Folium library** to visualize the geographical locations of launch sites and analyze their proximity to key landmarks. Here's an overview of the process:

- **Markers of All Launch Sites**:
  - We added **markers with circles, popup labels, and text labels** for each launch site using their latitude and longitude coordinates. This allowed us to visualize the geographical distribution of launch sites and their proximity to the equator and coastlines.
  - The map started with **NASA Johnson Space Center** as the initial location.
- **Colored Markers for Launch Outcomes**:
  - To analyze launch success rates, we added **colored markers** to indicate successful (green) and failed (red) launches for each site.
  - We used **Marker Cluster** to group markers, making it easier to identify launch sites with relatively high success rates.
- **Distances Between Launch Sites and Proximities**:
  - We added **colored lines** to show the distances between a specific launch site (e.g., **KSC LC-39A**) and its proximities, such as railways, highways, coastlines, and the closest city. This helped us understand the logistical advantages of each launch site.

Build a Dashboard with Plotly Dash

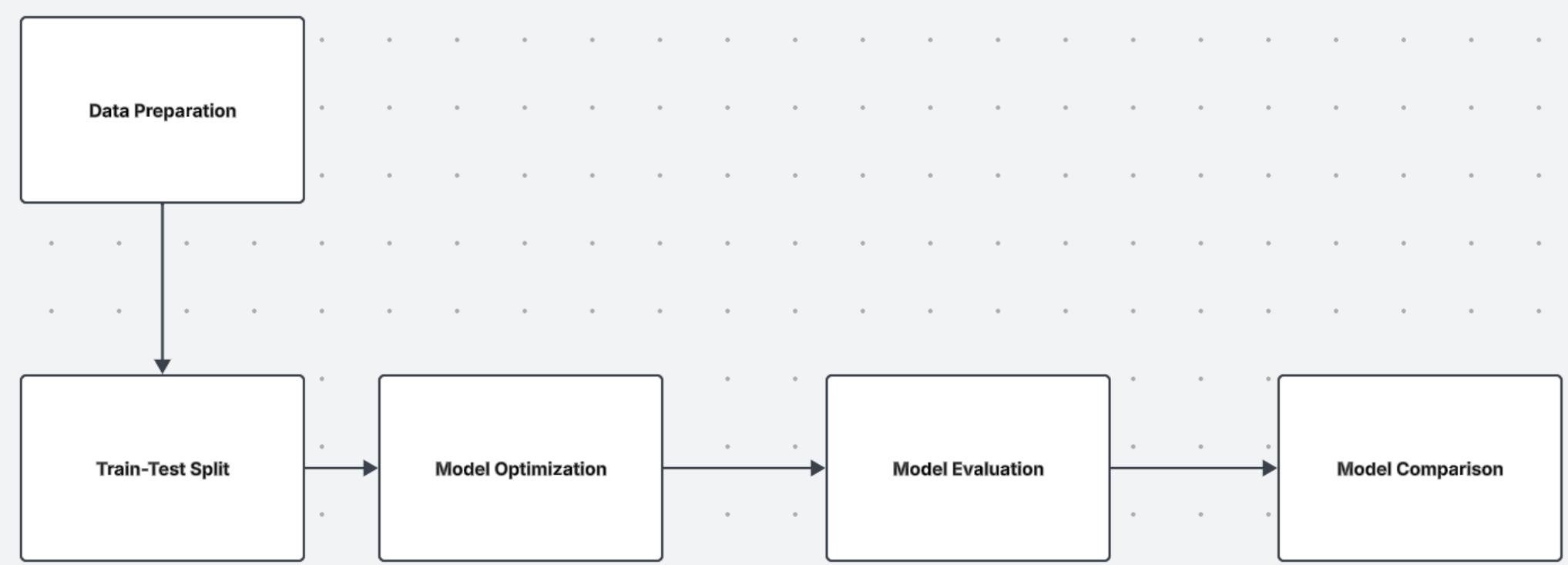
- **Summarized Text for Dashboard with Plotly Dash**
- We built an **interactive dashboard** using **Plotly Dash** to analyze and visualize SpaceX launch data. Here's an overview of the key components:
- **Launch Sites Dropdown List**:
  - We added a **dropdown list** to allow users to select specific launch sites. This feature enabled dynamic filtering of data based on the selected site.
- **Pie Chart for Success Launches**:
  - A **pie chart** was included to display the total number of successful launches across all sites.
  - When a specific launch site was selected, the pie chart updated to show the **success vs. failed launch counts** for that site.
- **Payload Mass Range Slider**:
  - We implemented a **slider** to filter the data based on payload mass range. This allowed users to focus on launches within a specific payload mass interval.
- **Scatter Chart for Payload Mass vs. Success Rate**:
  - A **scatter chart** was added to visualize the relationship between payload mass and launch success for different booster versions. This helped identify trends and correlations between payload mass and mission outcomes.

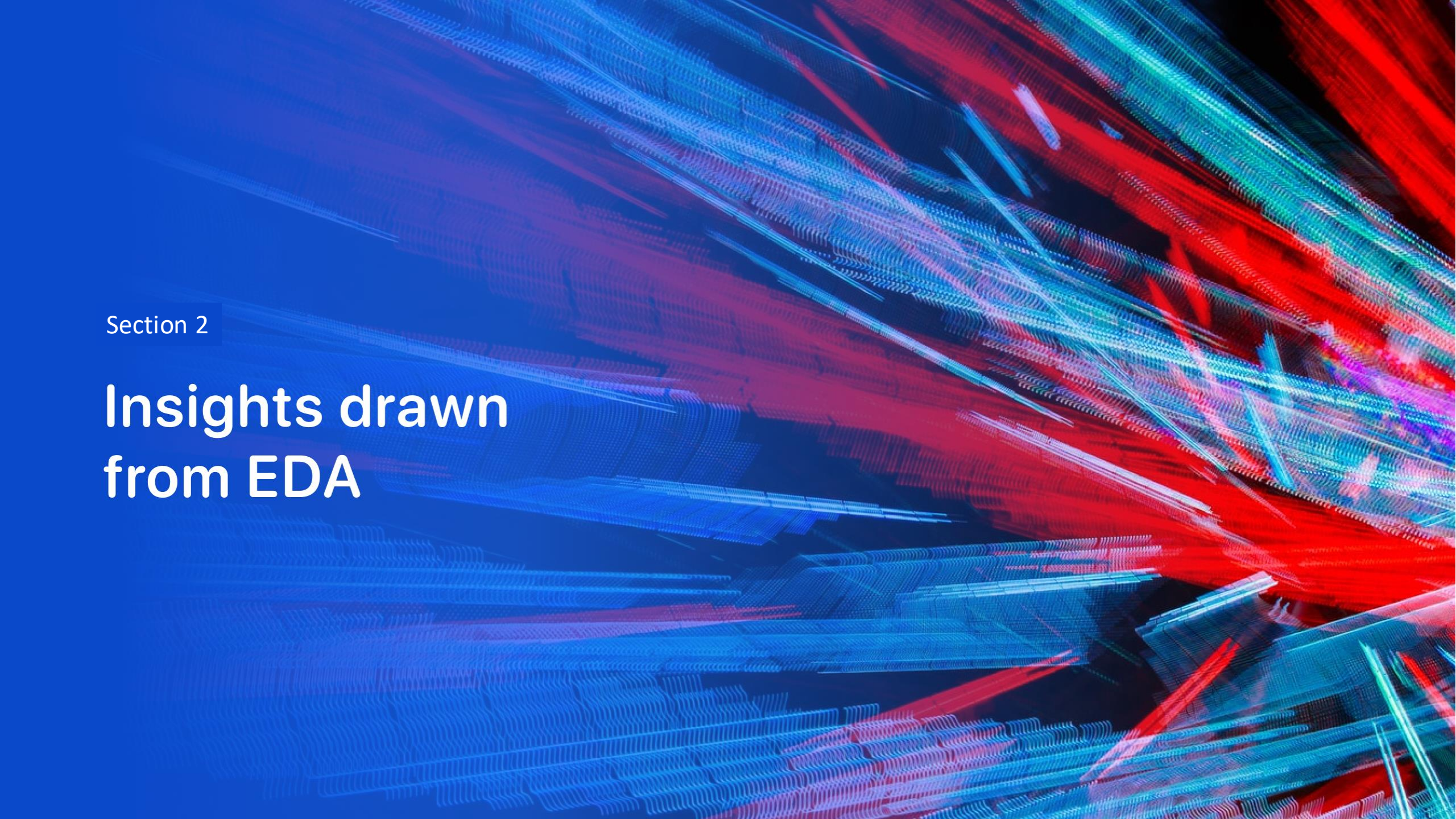
Predictive Analysis (Classification)

We performed **predictive analysis** using classification techniques to predict mission outcomes. Here's an overview of the process:

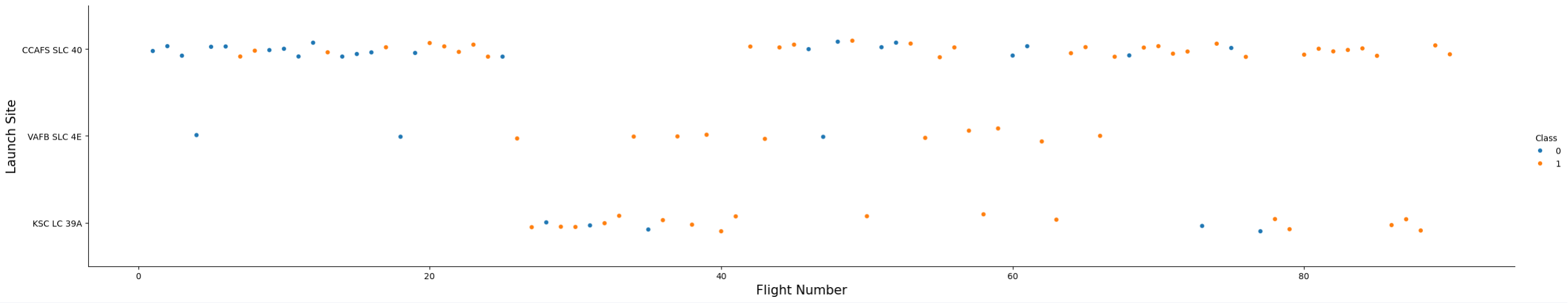
- **Data Preparation**:
  - We created a **NumPy array** from the "Class" column, which contained the target variable (e.g., success or failure).
  - The data was **standardized** using StandardScaler to ensure all features were on the same scale, improving model performance.
- **Train-Test Split**:
  - The dataset was split into **training and testing sets** using the train_test_split function. This allowed us to train the models on one subset and evaluate them on another.
- **Model Optimization**:
  - We used **GridSearchCV** with 10-fold cross-validation to find the **best hyperparameters** for each model. This ensured that the models were optimized for performance.
- **Model Evaluation**:
  - The models were evaluated using metrics such as **Jaccard score**, **F1 score**, and **accuracy** to assess their performance.
  - We also examined the **confusion matrix** for each model to understand the distribution of true positives, true negatives, false positives, and false negatives.
- **Model Comparison**:
  - We applied GridSearchCV to several models, including **Logistic Regression (LogReg)**, **Support Vector Machine (SVM)**, **Decision Tree**, and **K-Nearest Neighbors (KNN)**.
  - Based on the evaluation metrics, we identified the **best-performing model** for predicting mission outcomes.

Predictive Analysis (Classification)



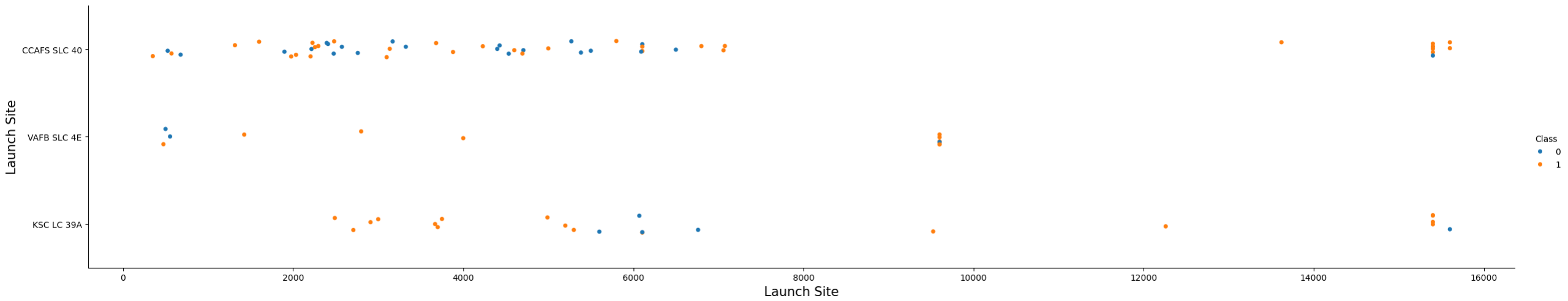

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- CCAFS SLC 40 has the most number of flights.
- Only 20% of the flights before Flight Number 20 landed successfully while 7% failed after that

Payload vs. Launch Site

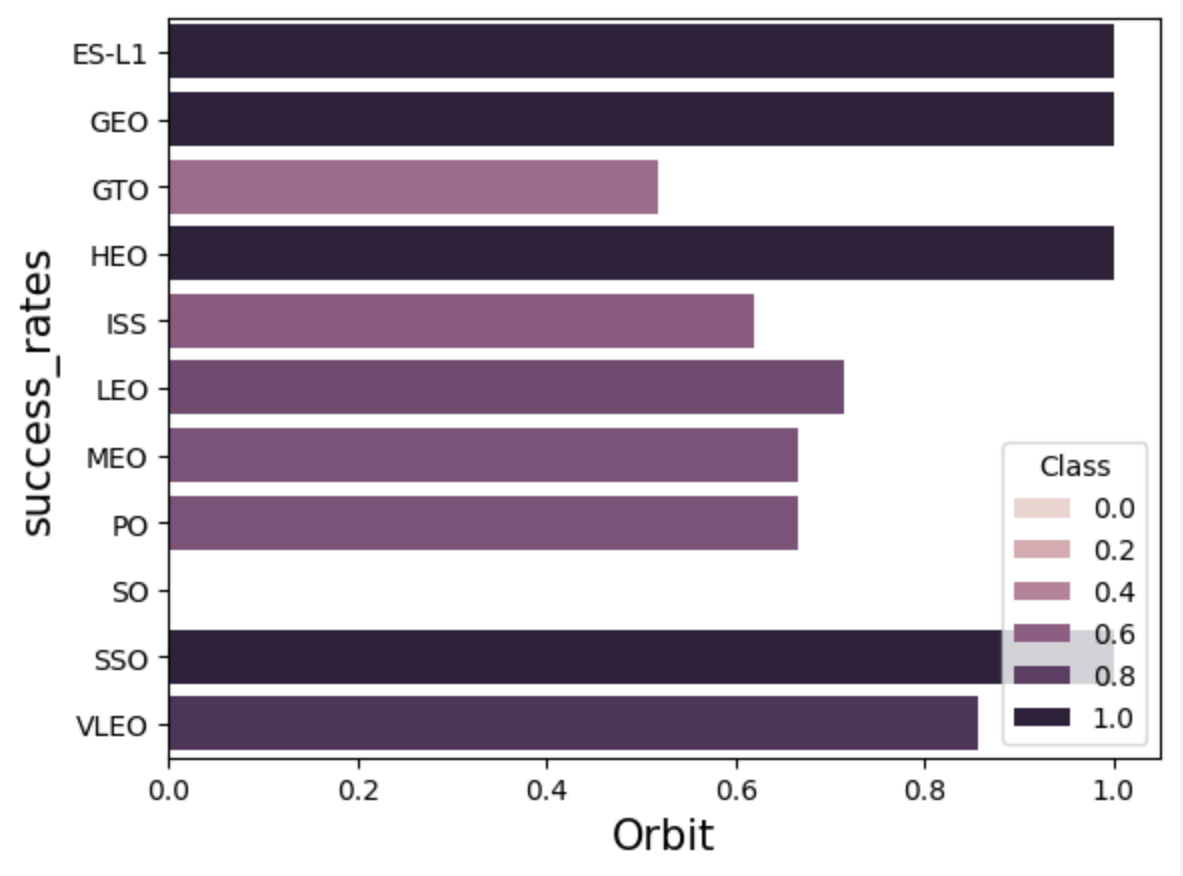


- There are no rockets launched for heavy payload mass(greater than 10000) for VAFB-SLC launchsite
- KSC LC 39A has a 100% successful rate when payload mass under 5500kg

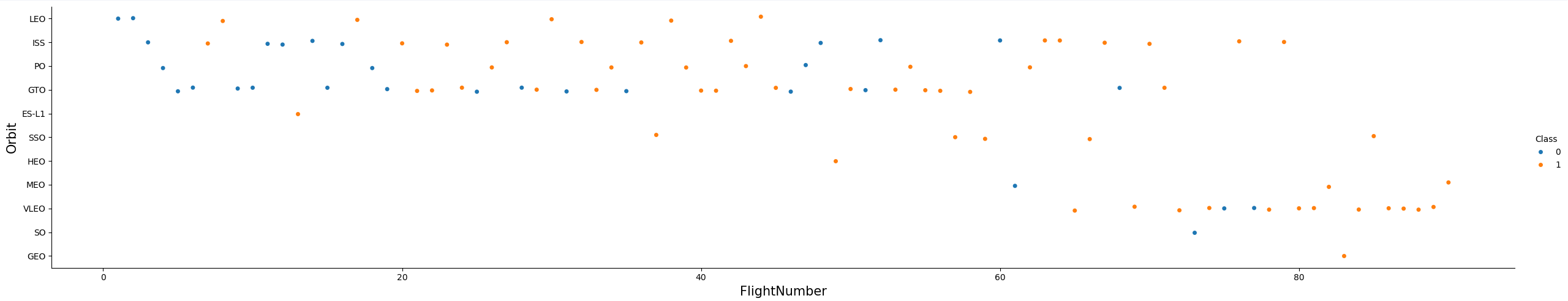
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO have the most successful landing

- SO has zero successful landings

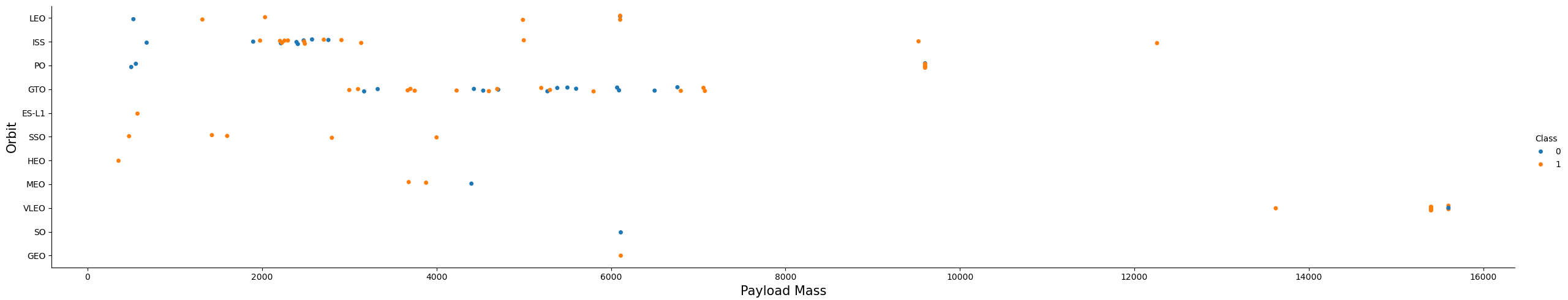


Flight Number vs. Orbit Type



- Orbits with 100% success has a low number of flights (<5)
- There is an increasing success rate as the number of flights rise independent from the Orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing are higher for Polar, LEO and ISS.

Launch Success Yearly Trend

- Until 2014, unsuccessful landings were more common
- The successful landings kept rising until 2018, when dropped 0.2 points
- Overall, the landings got more successful over the years

All Launch Site Names

- Query to get all the launch sites

```
%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'

- Query results presenting 5 launch sites beginning with "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

- Query results with the total payload carried by boosters from NASA

```
%sql select sum("PAYLOAD_MASS__KG_") as total_payload_mass from spacextable where customer = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
total_payload_mass
45596
```

Average Payload Mass by F9 v1.1

- Query results with the average payload mass carried by booster version F9 v1.1

```
%sql select avg("PAYLOAD_MASS__KG_") as avg_payload_mass from spacextable where booster_version like 'F9 v1.1%'
 * sqlite:///my_data1.db
Done.
 avg_payload_mass
2534.6666666666665
```

First Successful Ground Landing Date

- Query result with the date of the first successful landing outcome on ground pad

```
%sql select min(date) from spacextable where landing_outcome like '%ground%'
 * sqlite:///my_data1.db
Done.
 min(date)
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query result with the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%sql select distinct booster_version from spacextable where landing_outcome like '%drone%' and payload_mass__kg_ between 4000 and 6000
```

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

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Query results with the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(*) from spacextable group by mission_outcome
```

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

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

Boosters Carried Maximum Payload

- Query result with the names of the booster which have carried the maximum payload mass

```
%sql select Booster_Version, payload_mass__kg_ from (select max("payload_mass__kg_") as payload_mass__kg_,Booster_Version from spacextable group by booster_version order by payload_mass__kg_ desc l
 * sqlite:///my_data1.db
Done.
```

Booster_Version	payload_mass__kg_
F9 B5 B1060.3	15600

2015 Launch Records

- Query results with the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
[32]: %sql select  substr(Date, 6,2) as month, substr(Date, 0,5) as year, landing_outcome, booster_version, launch_site from spacextable where  substr(Date, 0,5) = '2015' and lower(landing_outcome) like '%f
 * sqlite:///my_data1.db
Done.
```

[32]:

month	year	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Query results ranking 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(*) from spacextable where date between '2010-06-04' and '2017-03-20' group by landing_outcome order by count(*) desc
```

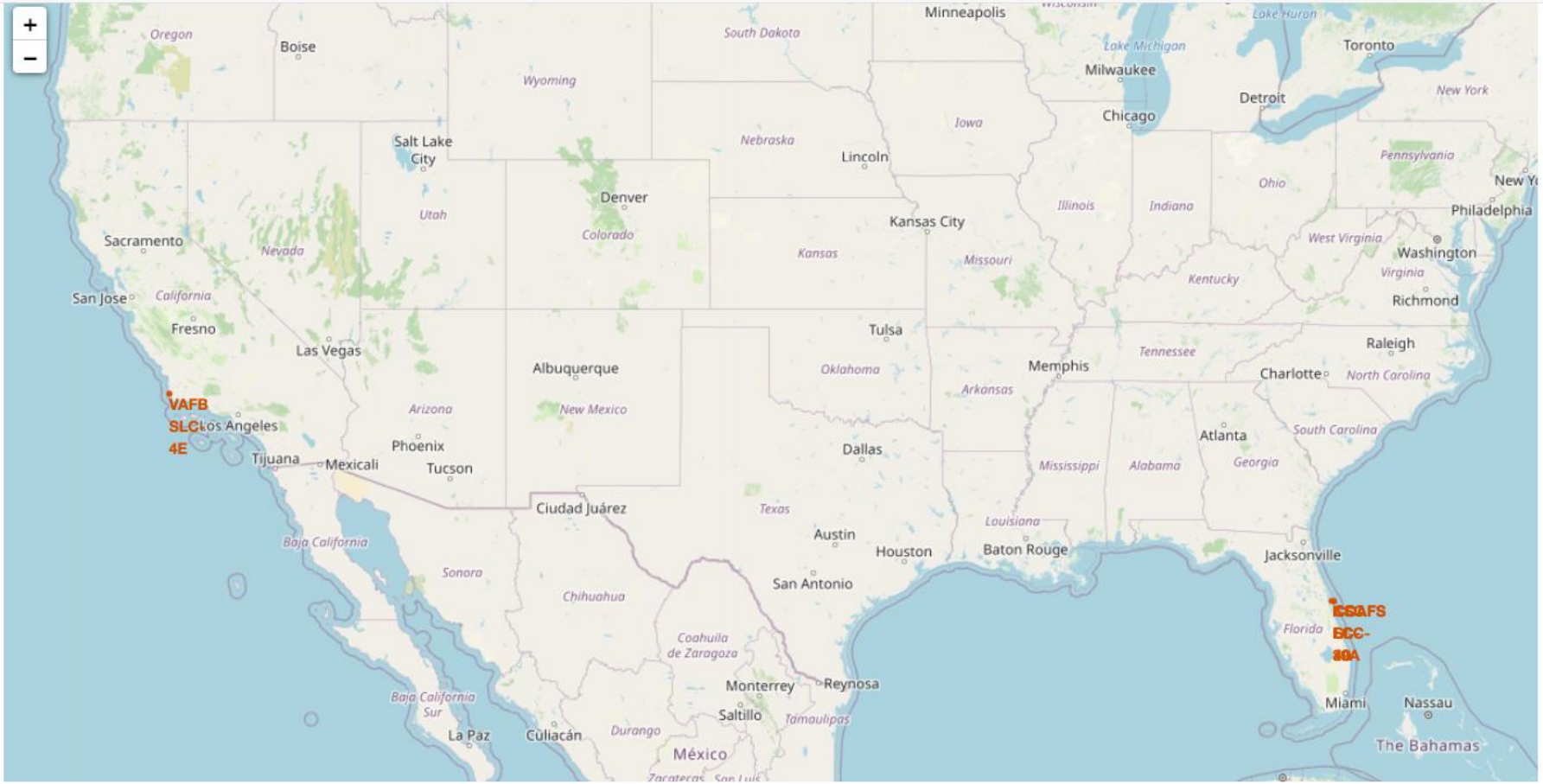
```
 * sqlite:///my_data1.db
Done.
```

Landing_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

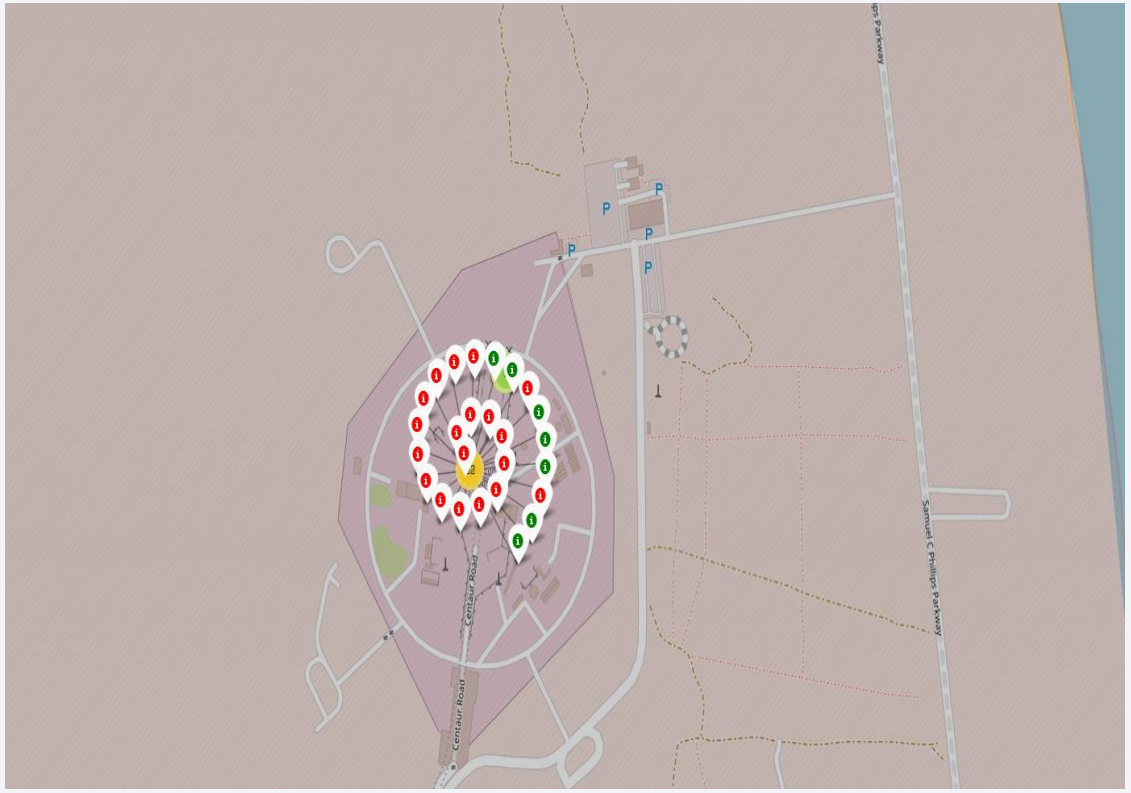
Map with Launch Site Locations



Launches Outcomes

- SLC – 40 has the most number os launches, even though it's not the most successful one

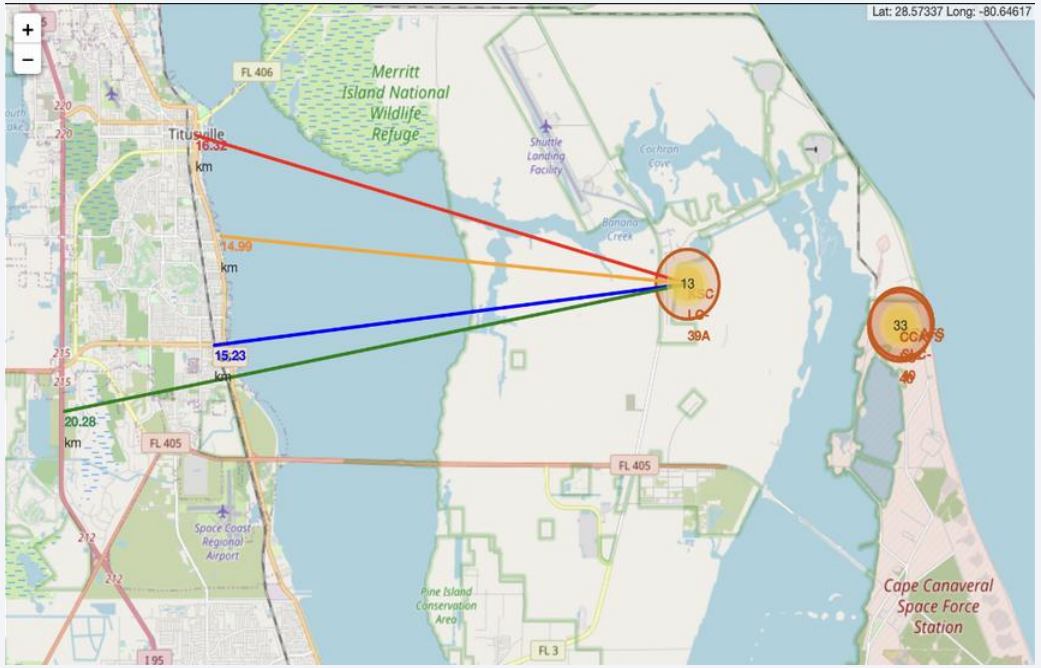
- Green markers shows the successful launches while red, the failed ones.

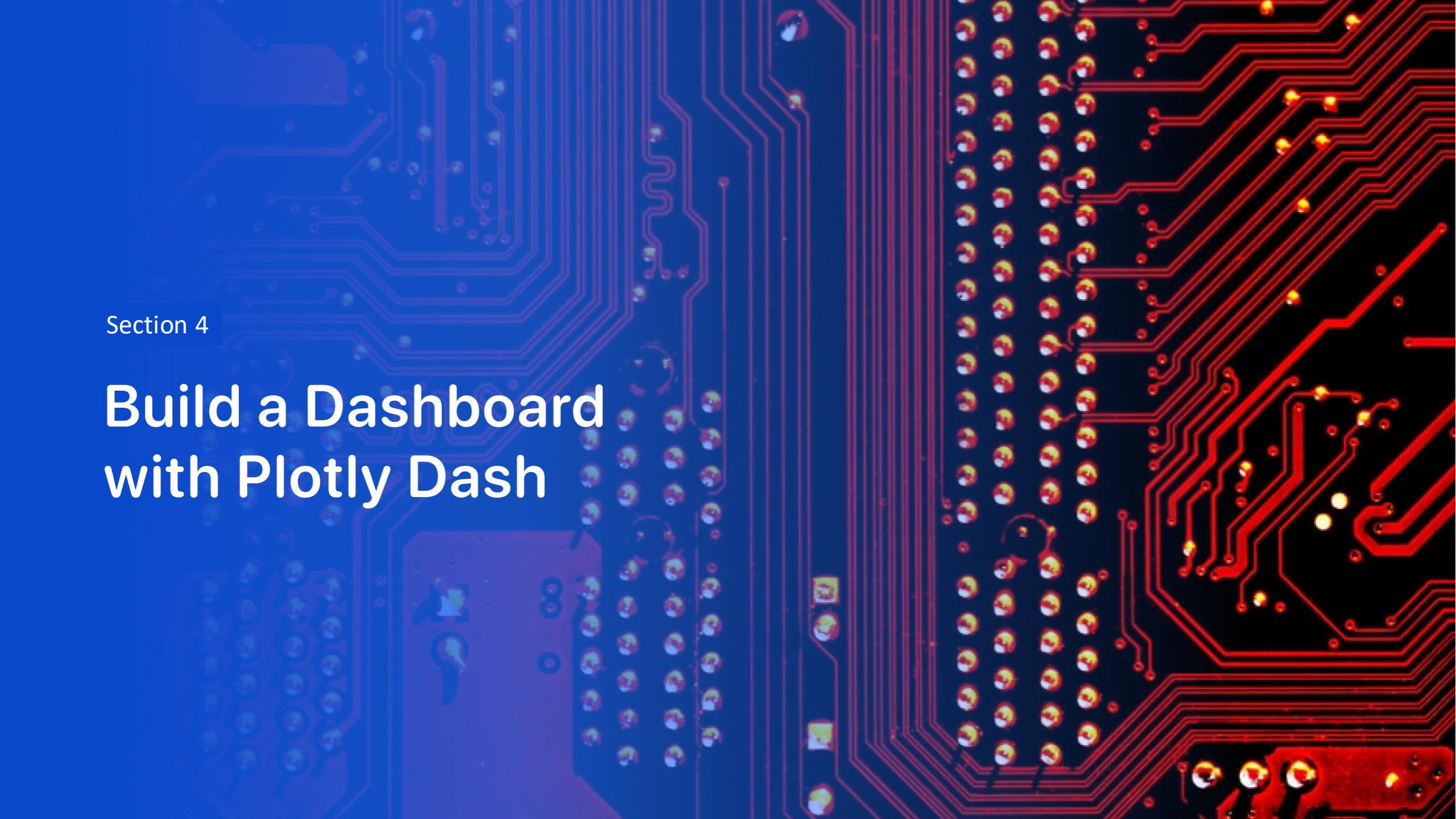


Distance between Launch Site and City

- Showing the distance between launch site and city infrastructure.

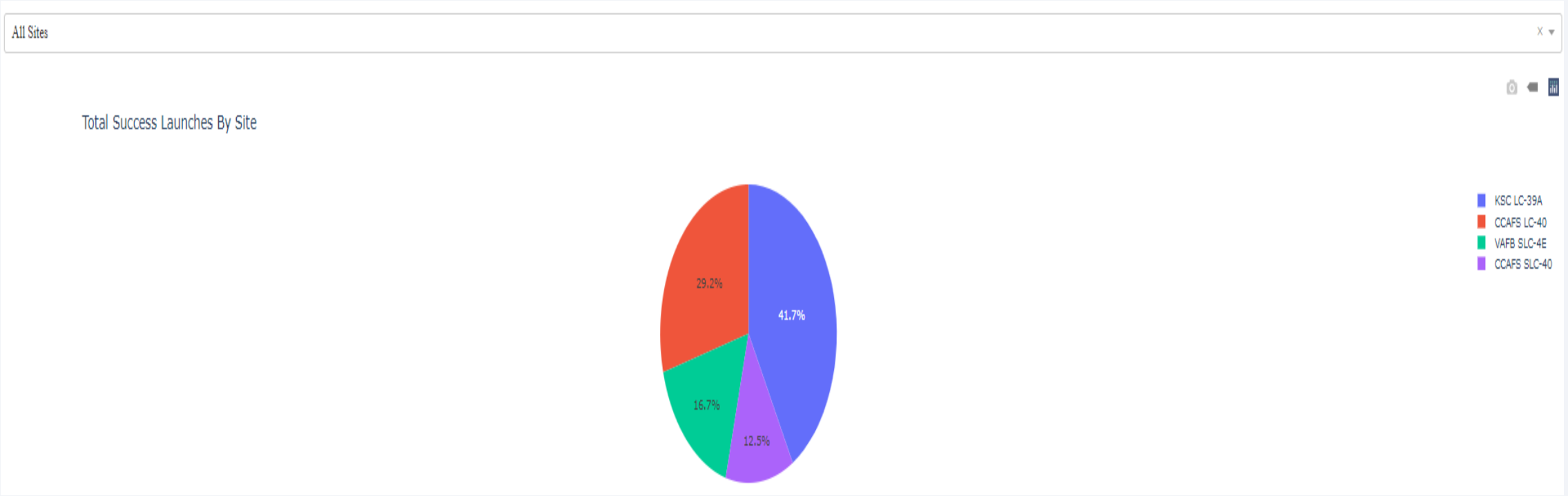
- While good to receive material or being atractive for hiring, might be dangerous if something goes wrong with the launch




Section 4
Build a Dashboard with Plotly Dash

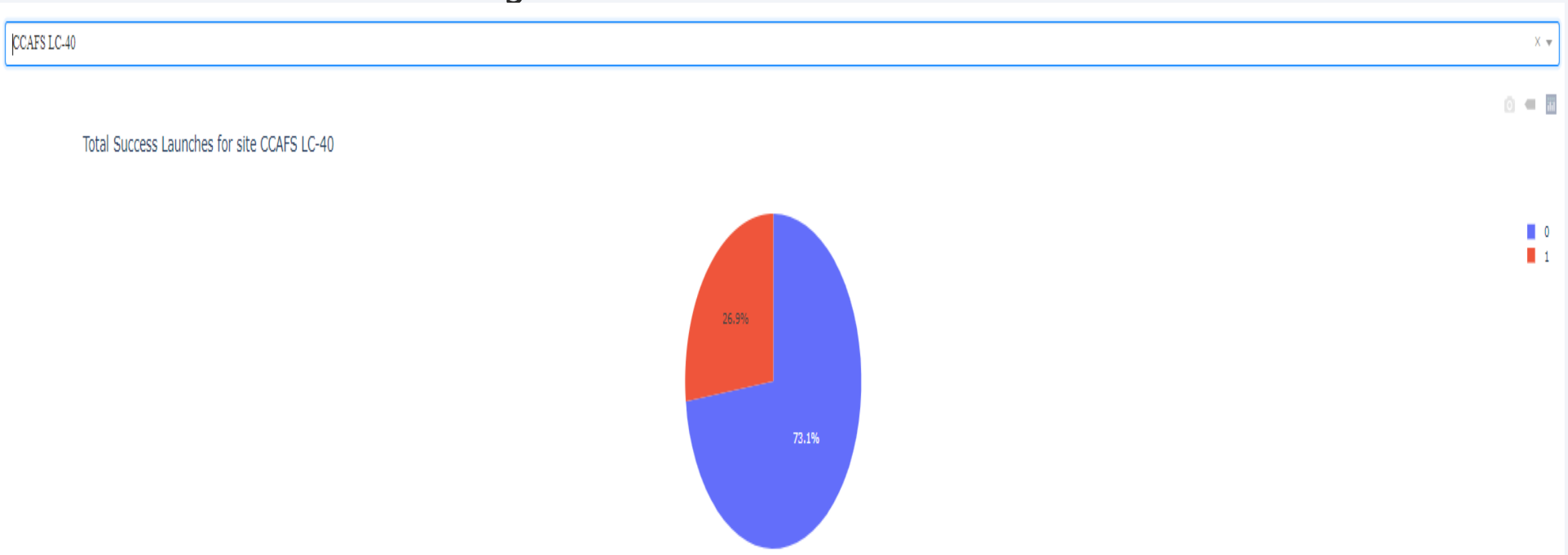
Success Rate by Launch Site

- KSC LC-39A is the most successful launch site.



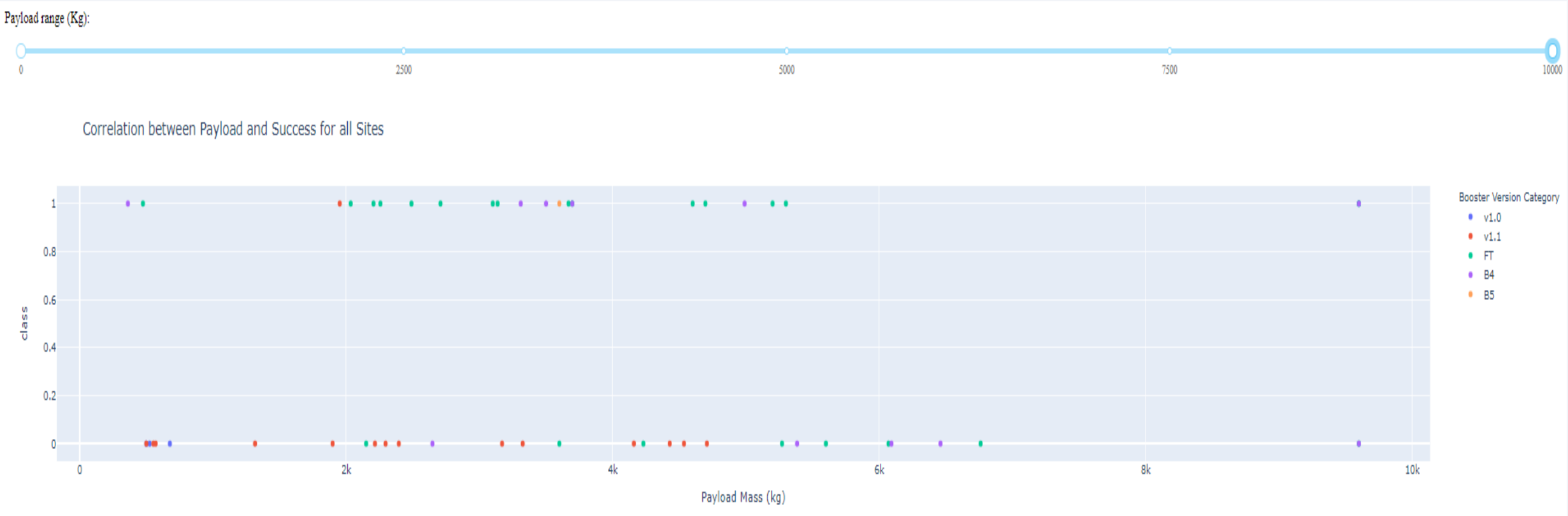
Highest Launch Success Ratio

- CCAFS LC-40 has the highest launch success ratio



Payload vs. Launch Outcome

- We can see that payload is not determinant for the Launch Outcome

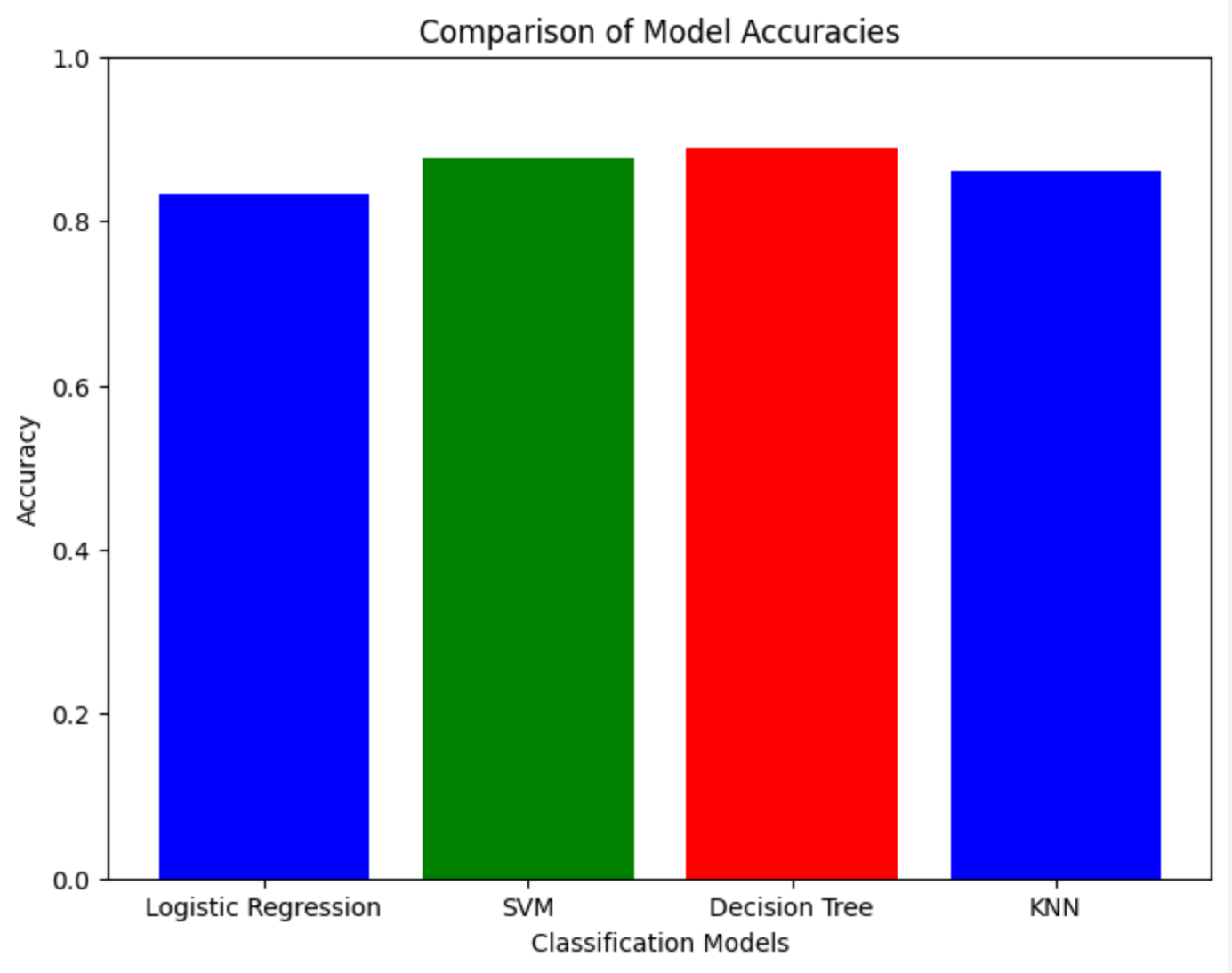


Section 5

Predictive Analysis (Classification)

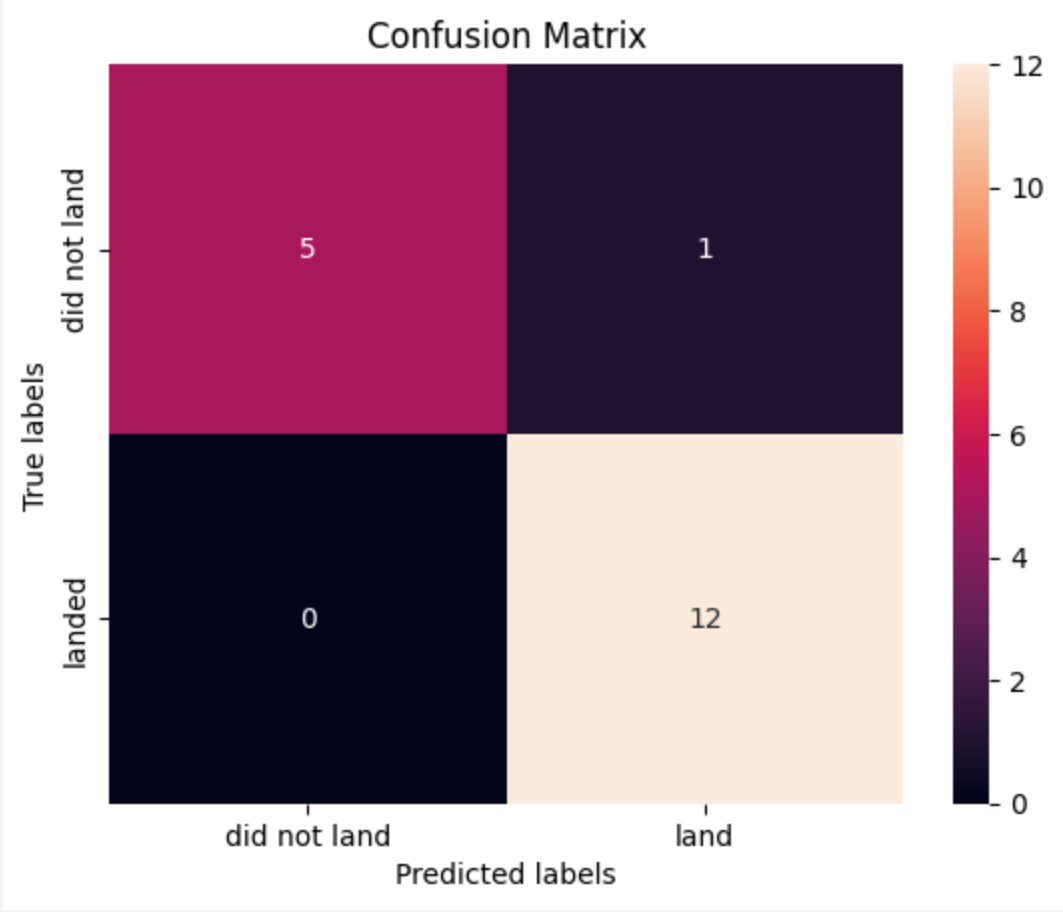
Classification Accuracy

- Decision Tree has the highest accuracy among all the methods, so it fits better our dataset.



Confusion Matrix

- The decision tree has predicted correctly 5 of the failed landings and 12 of the true landings, with only one False Positive.



Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

Thank you!