IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

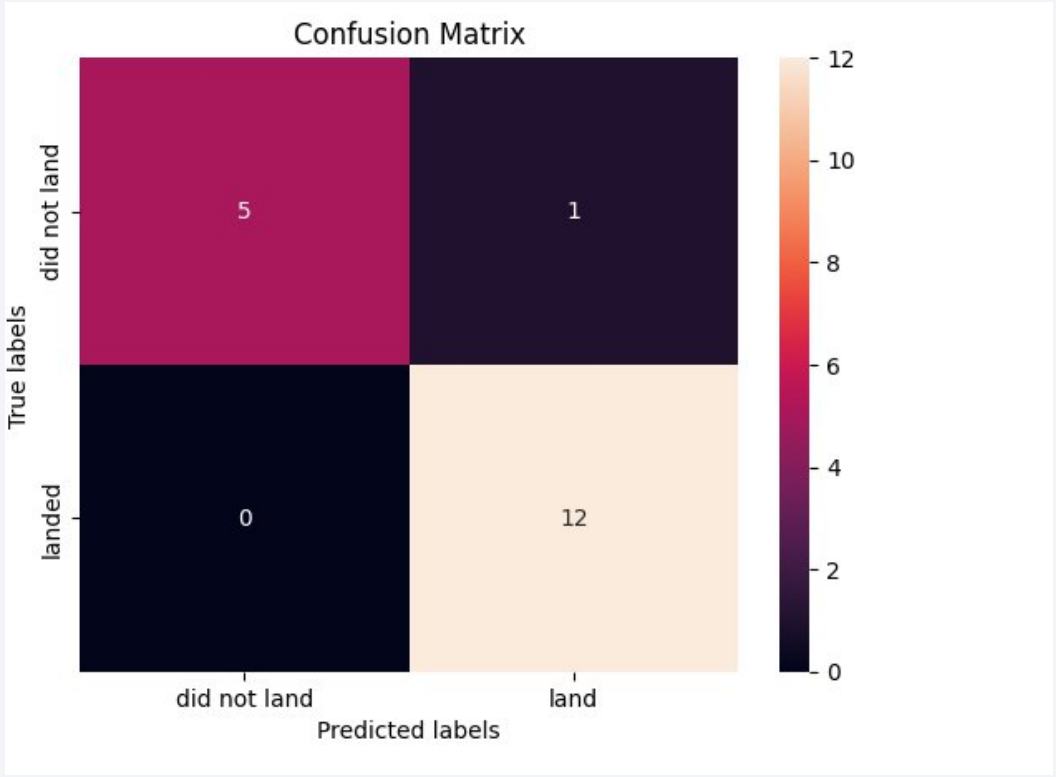
Abdul Basit Ahmad
25-05-2024

Outline

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

Executive Summary

- SpaceX Launch data is downloaded using wget and extracted using beautifulsoup into csv format.
- The data is loaded into a dataframe.
- EDA is performed to analyze the important features of the data.
- Using folium the data is plotted on a map to understand its spatial features.
- A dashboard summarizing launch data is prepared.
- Finally machine learning techniques are used to predict successful landings.



Introduction

- In this era of Commercial Space Age, SpaceX is the most successful space company.
- Part of their success is attributable to their utilization of reusable first stage, allowing it to launch a rocket for $62 million compared to $165 million for other competitors.
- The goal is to train a machine learning model on publicly available data to predict if a launch will be successful and the price of the launch.
- Using data from Wikipedia, the important parameters of a launch are obtained and analyzed using the Data Science packages of Python. This way an estimate for price of launch based on the parameters is obtained.

Section 1

Methodology

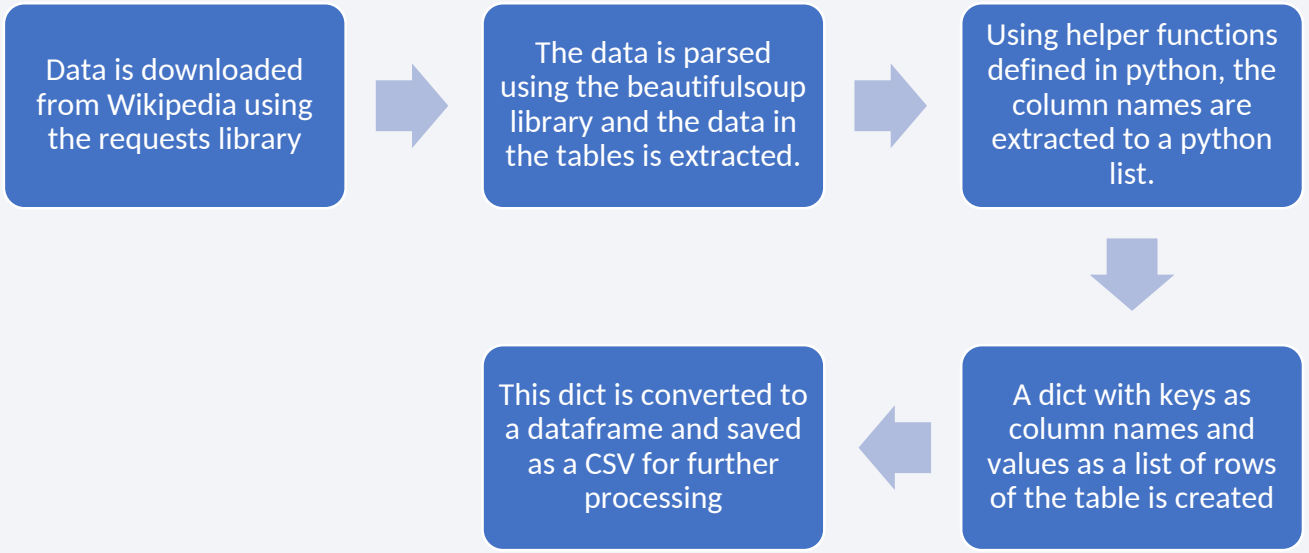
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from the publicly available data from Wikipedia
- Perform data wrangling
  - The data collected from Wikipedia was in the form of HTML and it was converted to CSV.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The failure and success of a launch is one-hot encoded as the Column ‘Class’ in a dataframe.
  - The whole dataset is split into a train and test set after scaling and standardization
  - The data is fit to various methods viz Decision Trees, SVM, Logistic Regression, kNN etc and the most accurate model is determined to be the kNN method and used to predict the price.

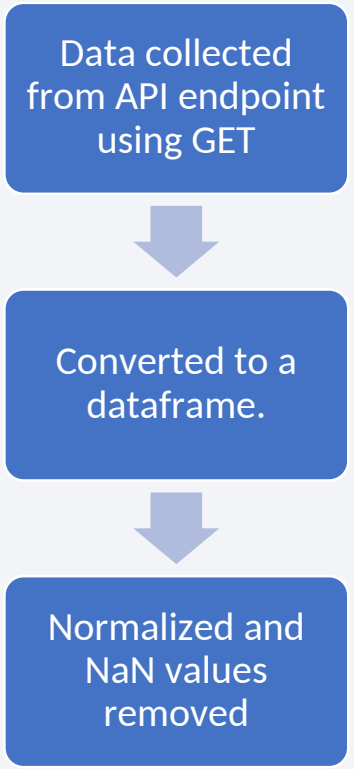
Data Collection

- The data set was collected from Wikipedia.



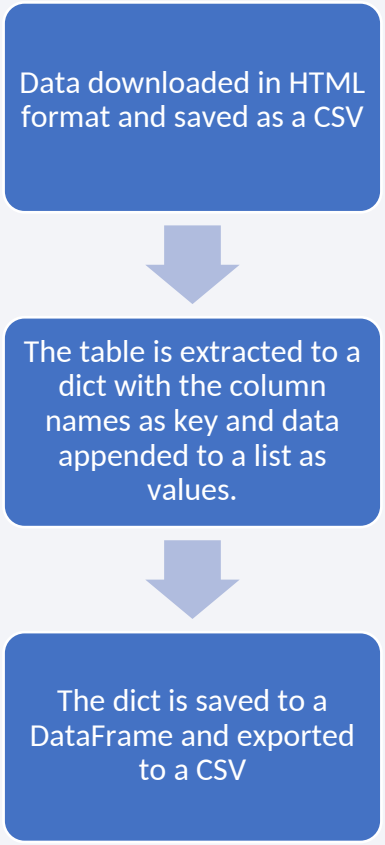
Data Collection – SpaceX API

- Data in JSON format is collected from the SpaceX REST API endpoint using GET
- It is converted to a DataFrame using the `pd.json_normalize` function.
- The data is standardized and normalized.
- The NaN values in Payload Mass column are replaced by averages.
- The data is exported to CSV.
- https://github.com/darthcoder/Capstone



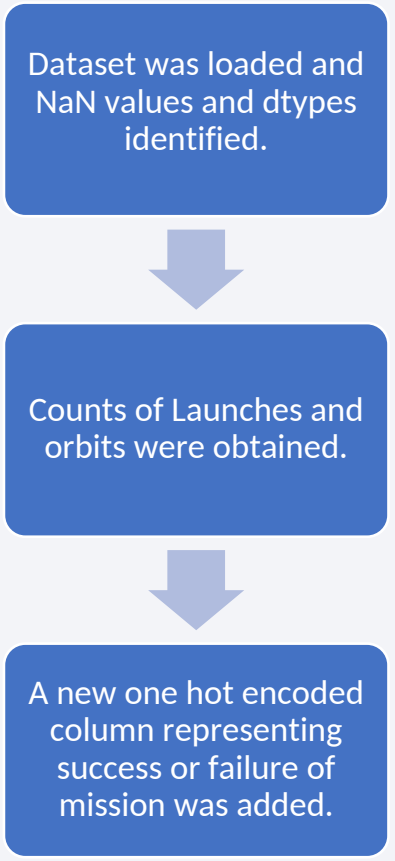
Data Collection - Scraping

- Using ‘requests’ data in HTML format is downloaded.
- This is saved to a BeautifulSoup object.
- The relevant table is extracted with the column names being saved as a key and the values being a list in a dict.
- The dict is saved to a DataFrame and exported to a CSV.
- https://github.com/darthcoder/Capstone/blob/master/week1/jupyter-labs-webscraping.ipynb



Data Wrangling

- The dataset was loaded, NaN values and column dtypes identified.
- Launches from each site counted as was value counts of each type of orbit.
- The outcomes were classified to a class variable.
- A new one-hot encoded column was added representing class of failure or success of launch.
- https://github.com/darthcoder/Capstone/blob/master/week1/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- A categorical plot of flight number vs payload mass colored by Class.
- A categorical plot of flight number vs launch site colored by Class.
- A scatter plot of payload vs launch site colored by Class.
- A bar plot of success rate vs launch site to visualize which sites are more successful at launch.
- A scatter plot of flight number vs Orbit type to understand the evolution of SpaceX’s space launch strategies.
- A plot of payload vs orbit type to understand how much payload is sent to which orbit.
- Yearly trend of launch success by a line plot.
- https://github.com/darthcoder/Capstone/blob/master/week2/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Distinct launch sites were identified.
- 5 records where launch sites begin with the string 'CCA'.
- total payload mass carried by boosters launched by NASA (CRS)
- average payload mass carried by booster version F9 v1.1
- the date when the first successful landing outcome in ground pad was achieved was found to be 2015-12-22
- the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- total number of successful and failure mission outcomes
- the names of the booster_versions which have carried the maximum payload mass.

- https://github.com/darthcoder/Capstone/blob/master/week2/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch sites are marked on a map using Icons and grouped into Circles.
- All launches are indicated with a marker with the color of marker indicating whether the launch was successful or not.
- Red color was used to denote failure and green for success.
- This way one can get a quick overview of whether or not a given launch was successful and where it was on the map.
- Interesting points of interest were marked on the map near the launch sites.
- https://github.com/darthcoder/Capstone/blob/master/week3/lab_jupyter_launch_site_location.ipynb

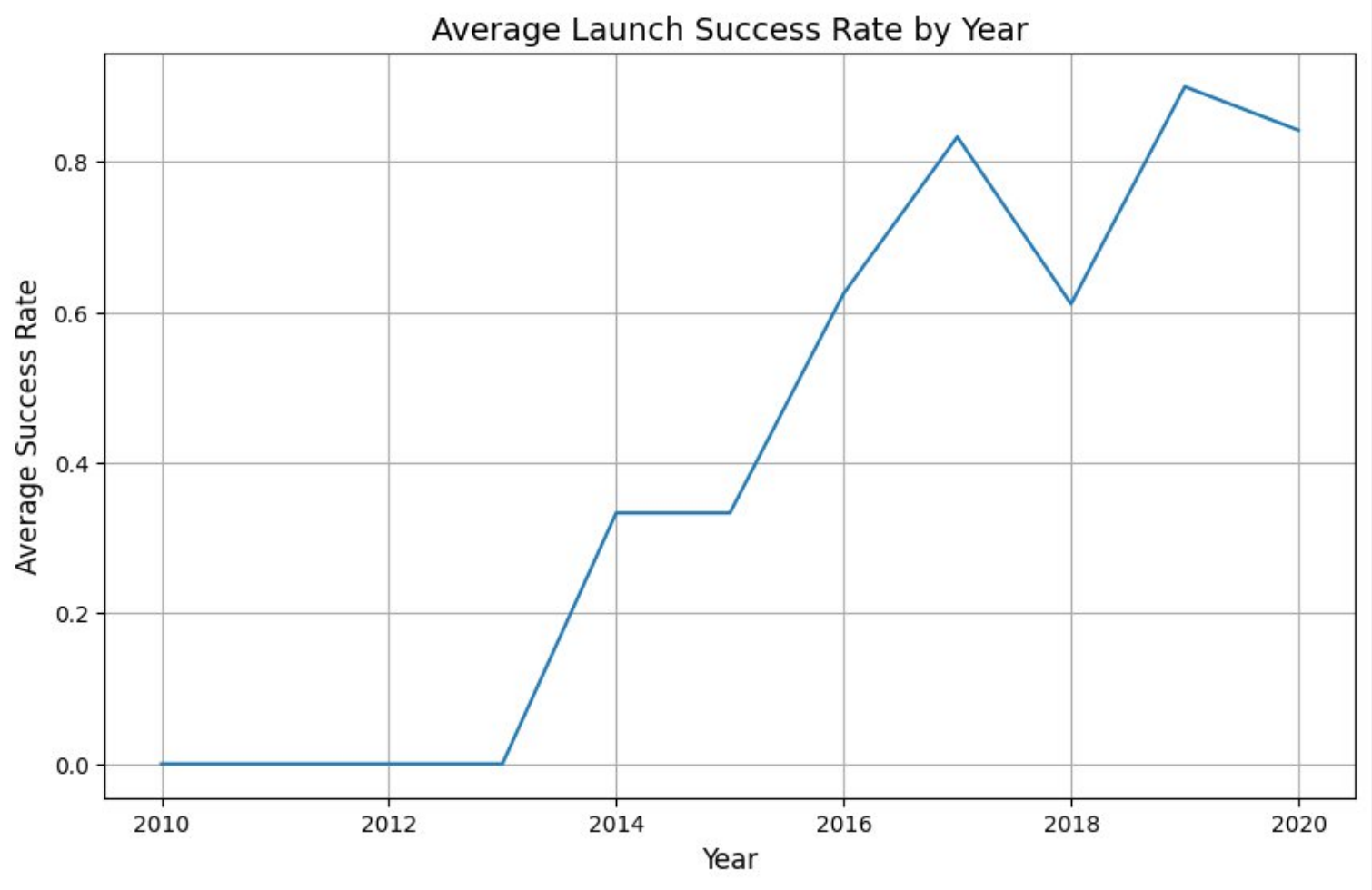
Build a Dashboard with Plotly Dash

- A dropdown for selecting the launch site, a slider for selecting the payload mass were made.
- On selecting a launch site, a pie chart showing the percentage of successful and failed launches from that site are displayed. There is the option of displaying an aggregated pie chart of all the sites.
- A range slider was made to select the payload mass.
- A scatter plot of payload mass vs success was displayed, colored by the booster type to easily spot trends in payload mass and success and its dependence on the type of booster used.
- https://github.com/darthcoder/Capstone/blob/master/week3/spacex_dash_app.py

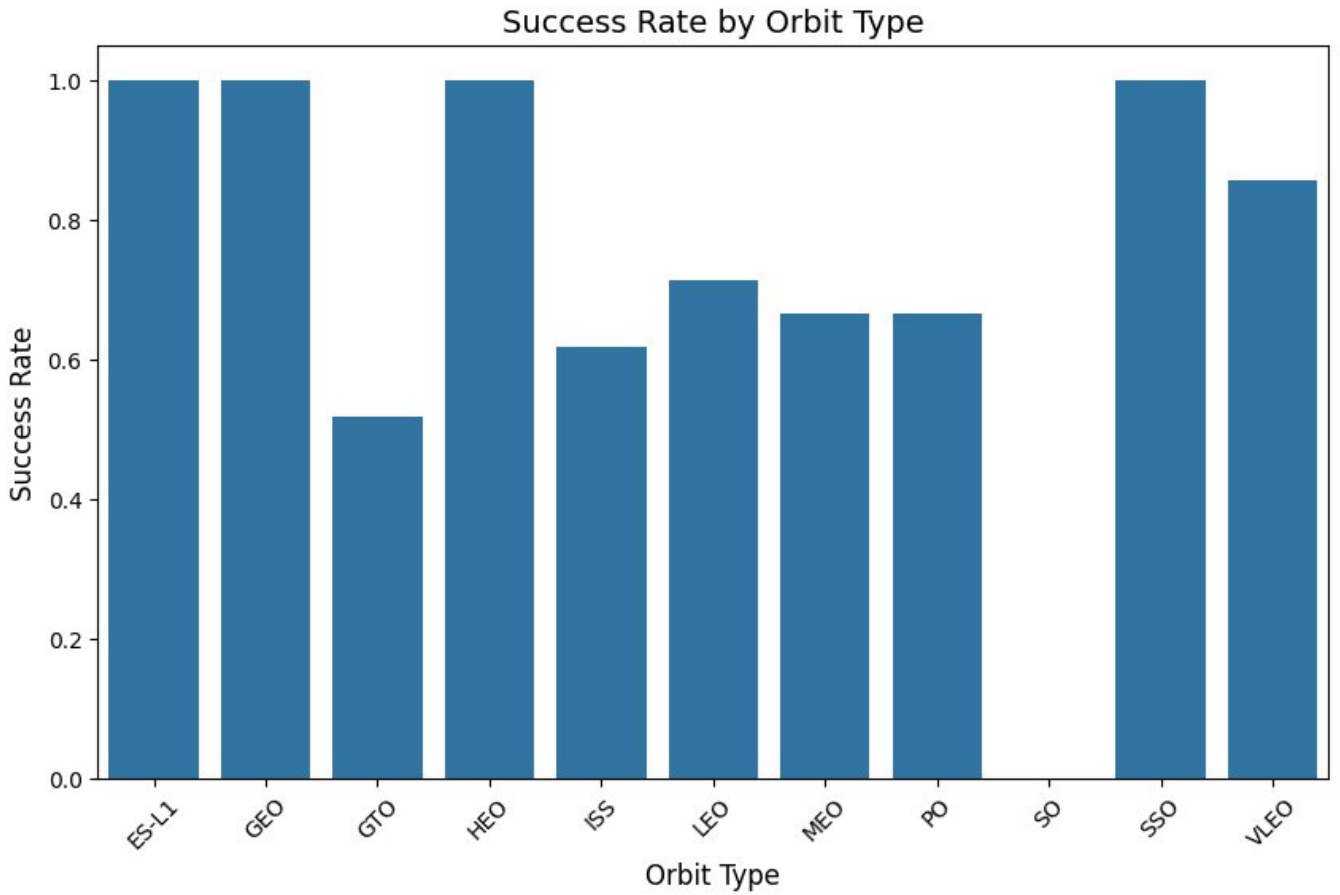
Predictive Analysis (Classification)

- The data is scaled and transformed using the fit and transform methods of the StandardScaler class.
- The ‘Class’ one hot encoded column that stored the success or failure of a launch is converted to a Numpy array named Y to allow it to be
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

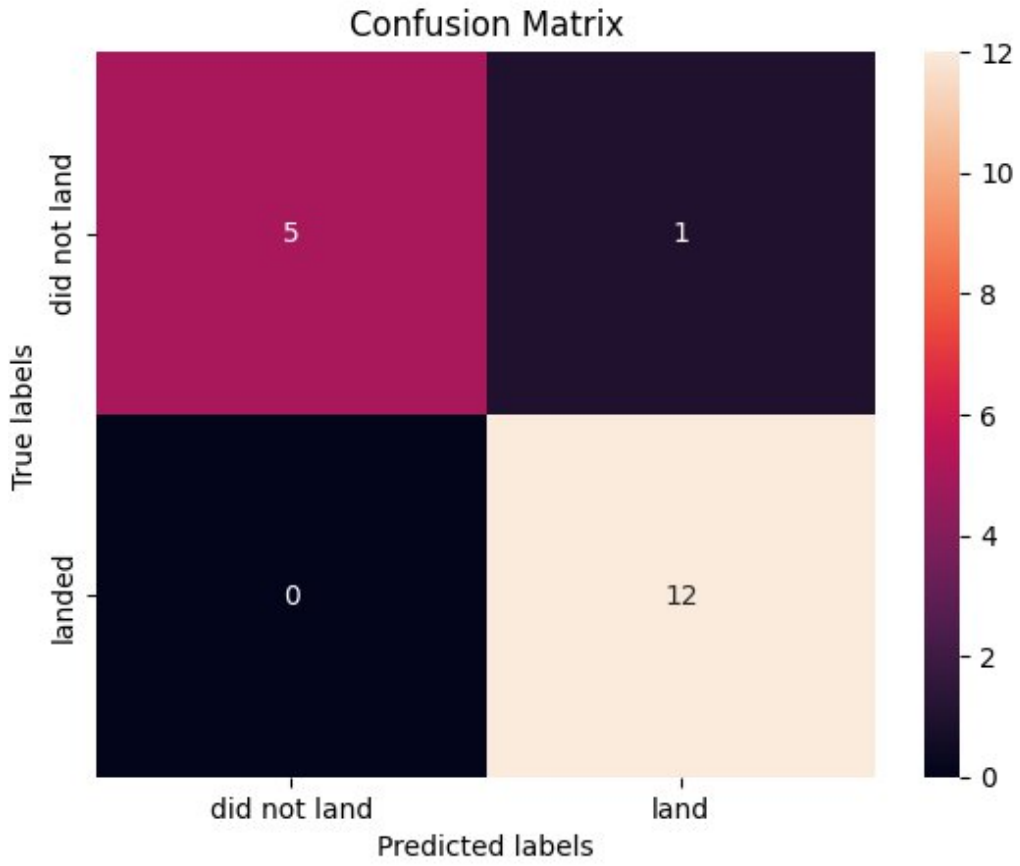
Results

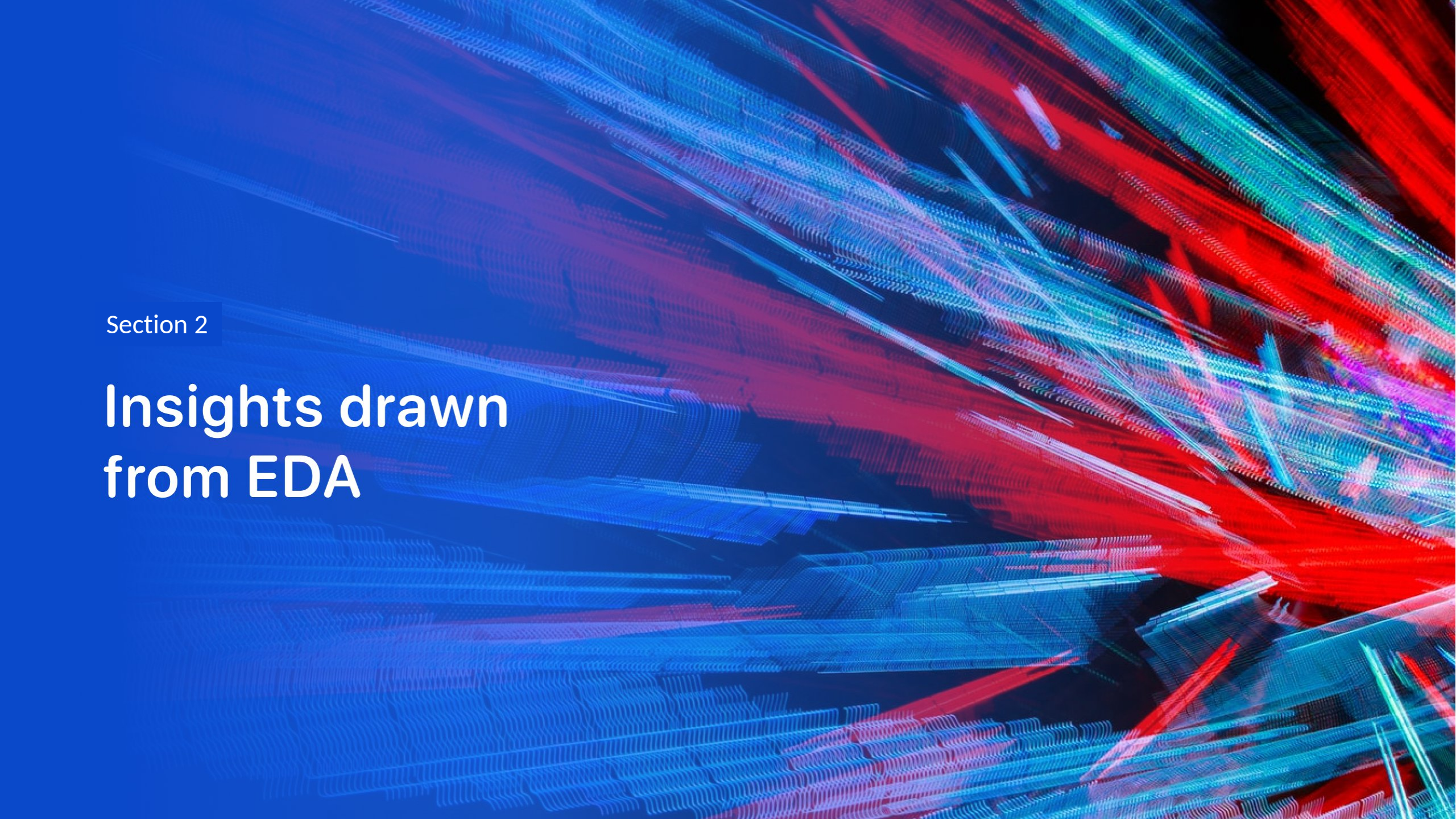

Average Launch Success Rate by Year
Average Success Rate
0.0
0.2
0.4
0.6
0.8
2010
2012
2014
2016
2018
2020
Year

Results

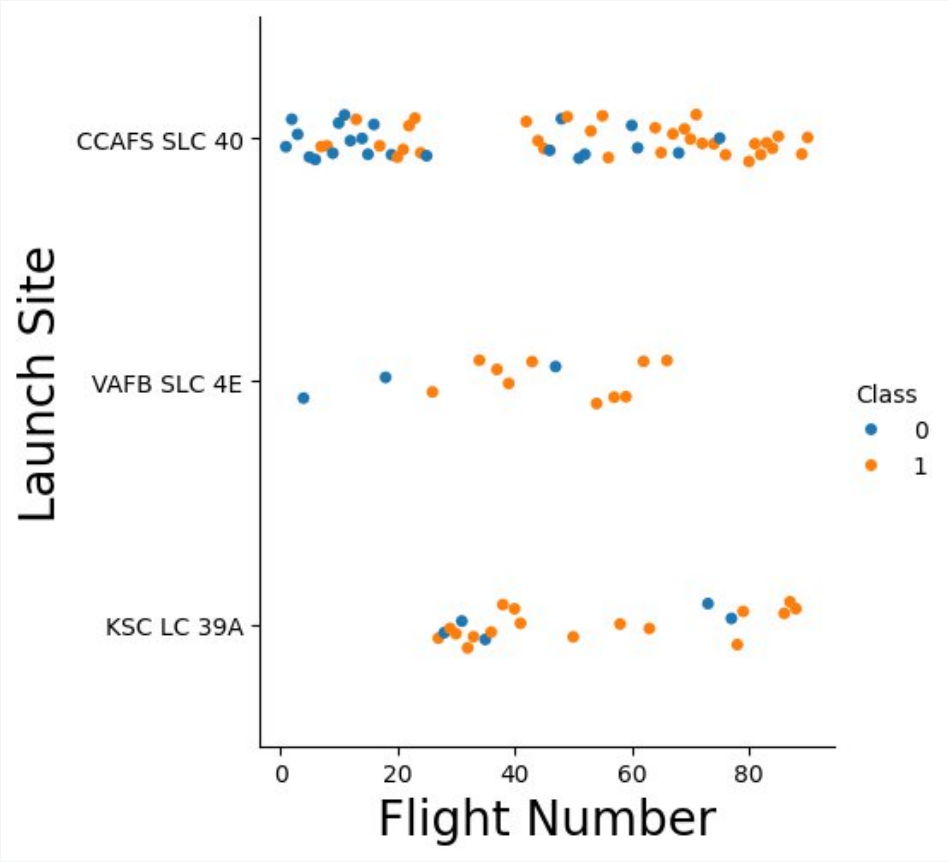

Success Rate by Orbit Type
Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
ES-L1
GEO
GTO
HEO
ISS
LEO
MEO
PO
SO
SSO
VLEO
Orbit Type

Results

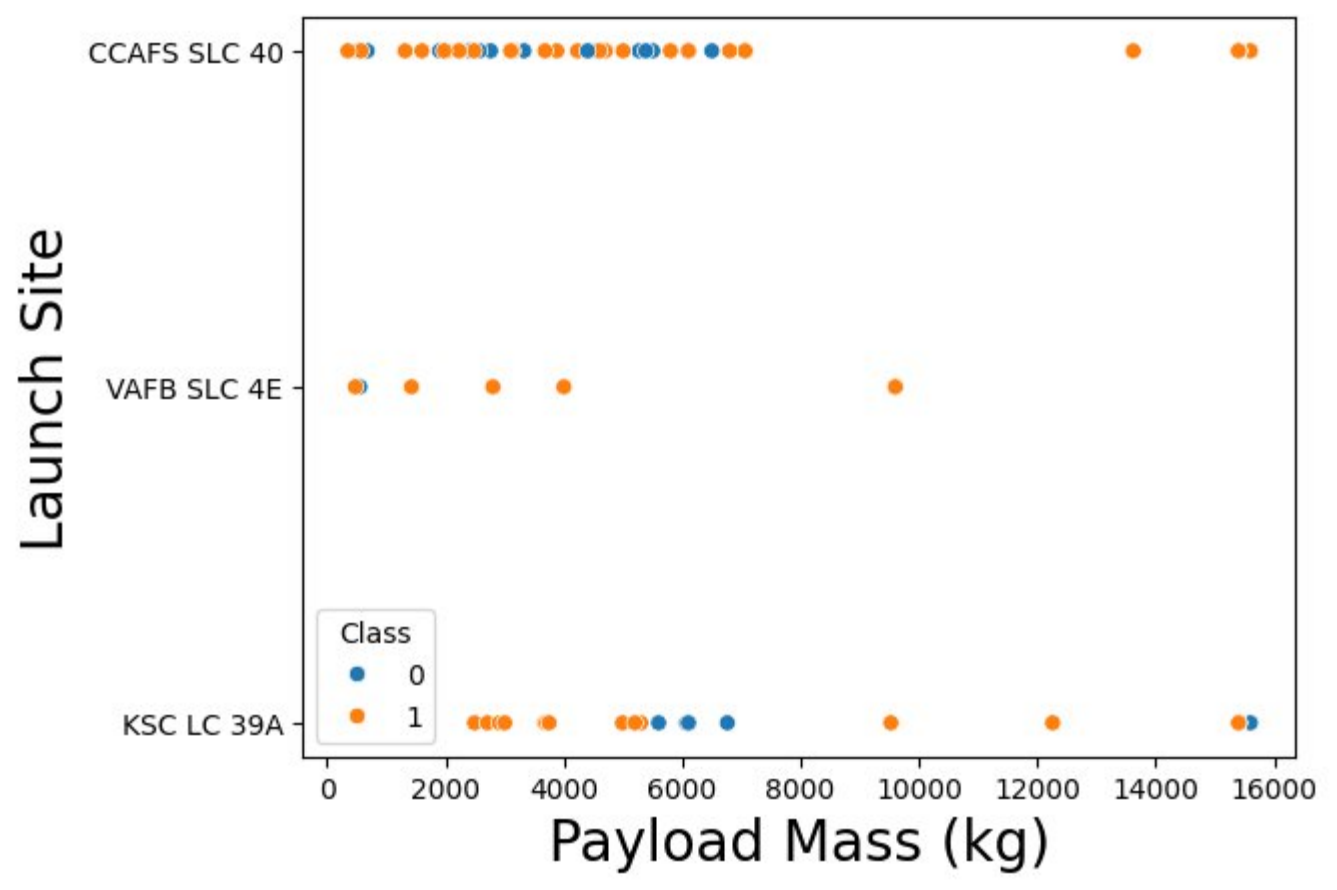

Confusion Matrix
5
1
0
12
did not land
landed
True labels
did not land
land
Predicted labels
12
10
8
6
4
2
0


Section 2
Insights drawn from EDA

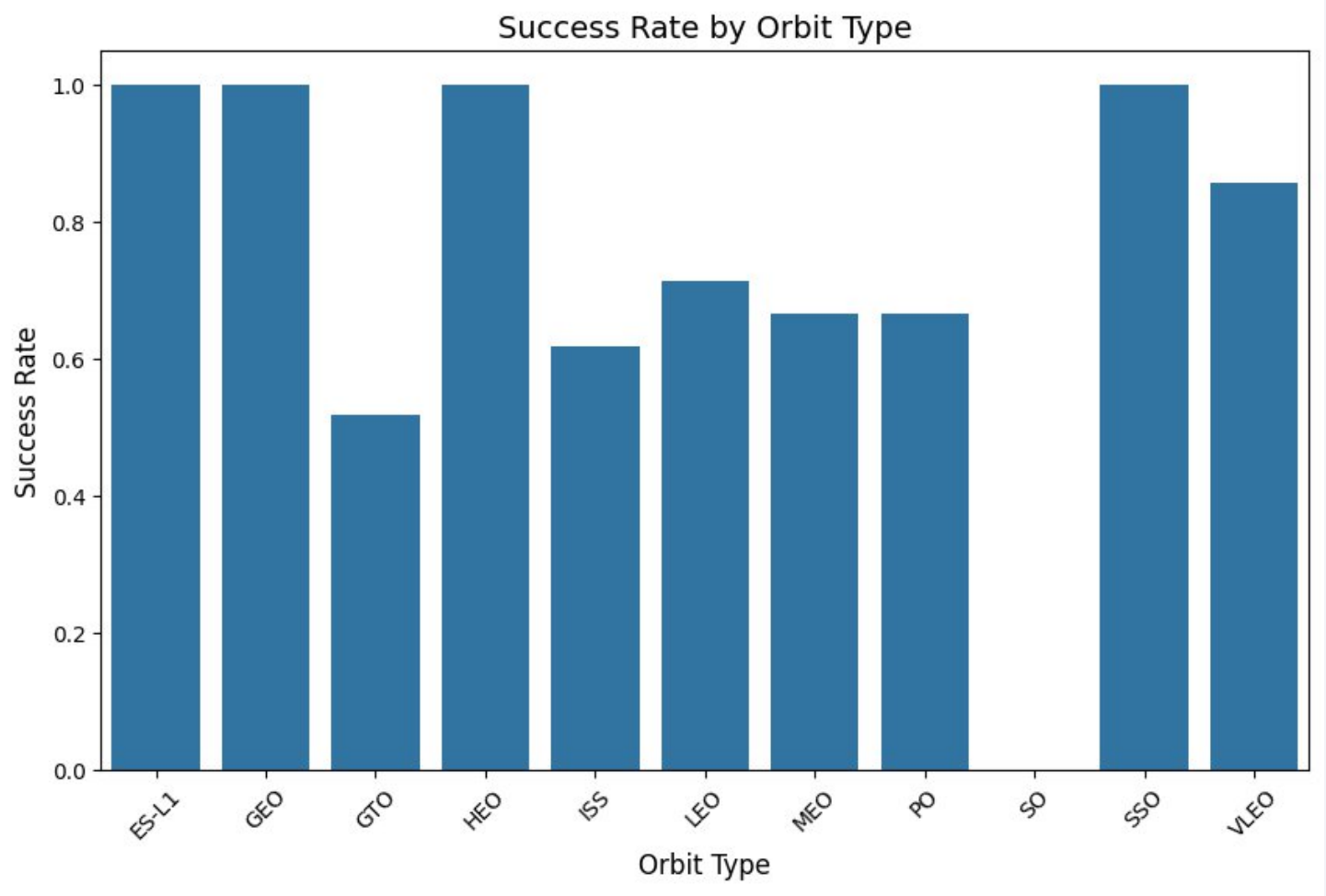
Flight Number vs. Launch Site



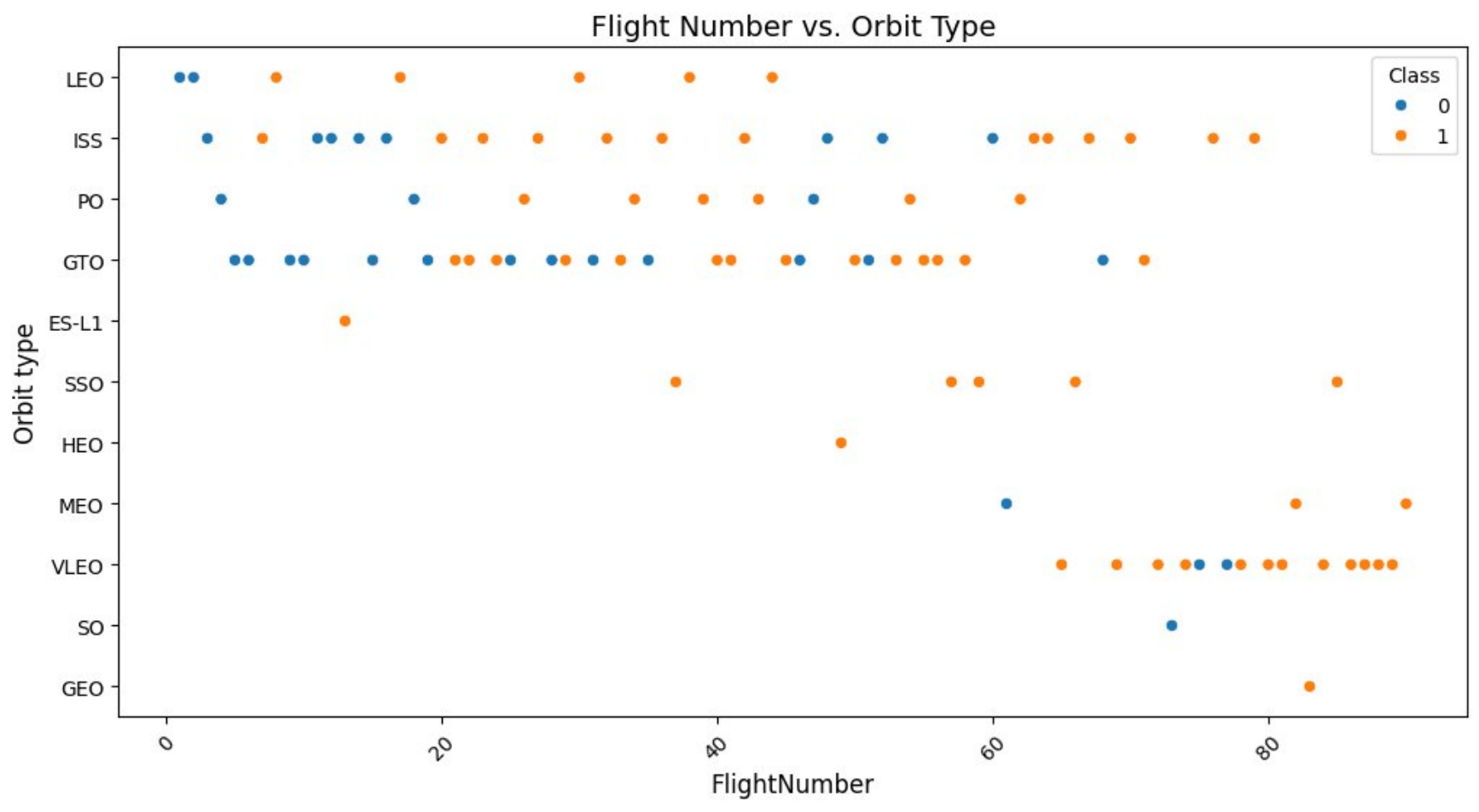
Payload vs. Launch Site



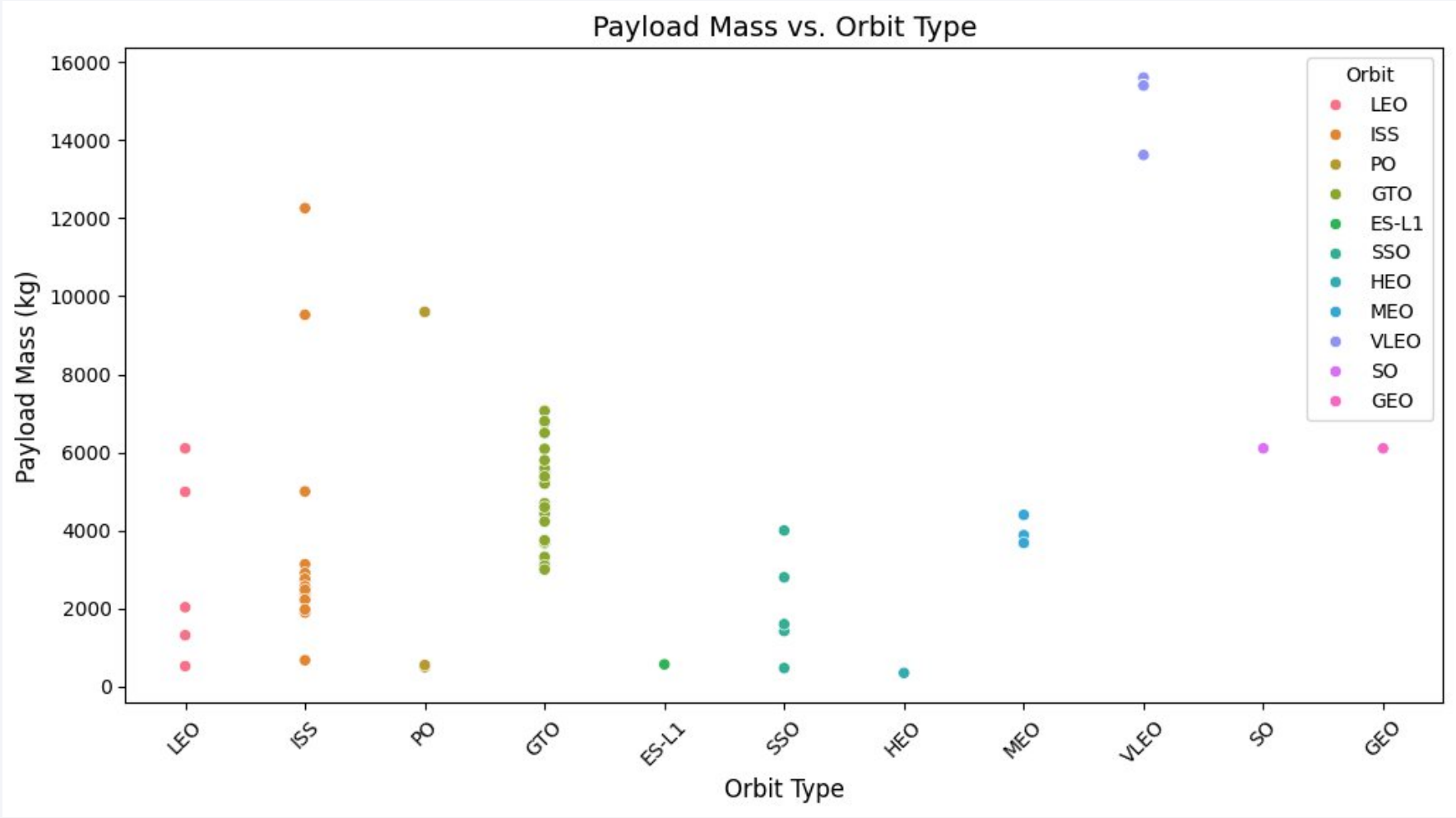
Success Rate vs. Orbit Type



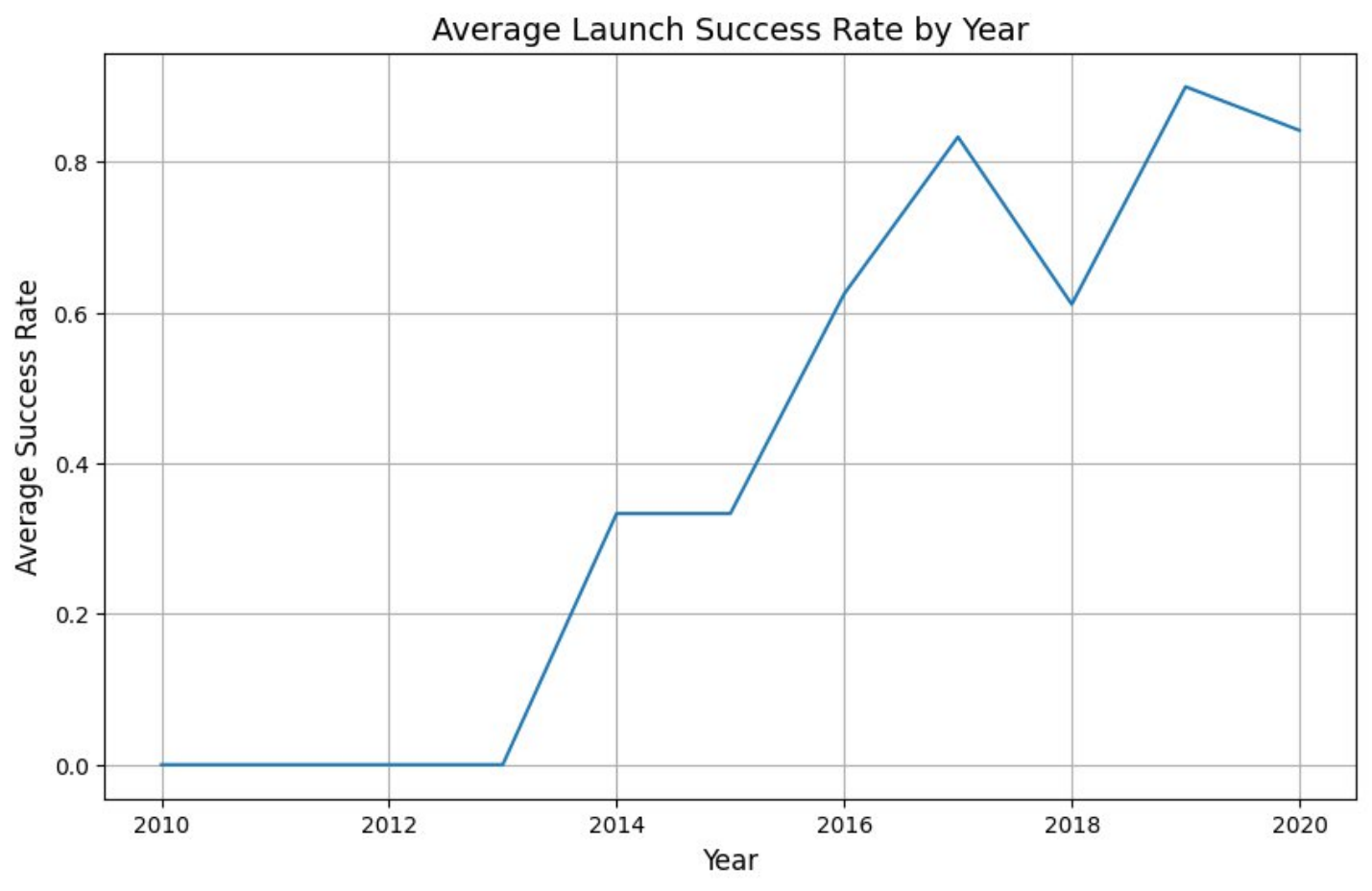
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

- The distinct clause is used to select the distinct launch sites.

```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

- The distinct launch sites are as follows:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

```
%sql select * from spacextable where "Launch_Site" like "CCA%" limit 5
```

- The results are as follows

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- The total payload carried by boosters from NASA

SUM(payload_mass__kg_)
45596

- The query is shown below

```
%sql SELECT SUM(payload_mass__kg_) FROM spacextable WHERE customer = 'NASA (CRS)';
```

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1

avg(payload_mass__kg_)
2534.6666666666665

- Query is as follows

```
%sql select avg(payload_mass__kg_) from spacextable where booster_version like "F9 v1.1%"
```

First Successful Ground Landing Date

- First successful landing outcome on ground pad

min(date)
2015-12-22

- Query is as follows

```
%sql select min(date) from spacextable where Landing_outcome like "Success%"
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

- The query result with a short explanation is shown here

```
%sql select Booster_Version from spacextable where Landing_outcome is "Success (drone ship)" and payload_mass__kg_ between 4000 and 6000
```

- The booster version column is chosen from the table where the condition is expressed by the joint Boolean condition Landing_outcome is Success (drone ship) and the payload_mass__kg_ column is between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Success_Count	Failure_Count
98	1

- The query result with a short explanation here

```
%%sql
SELECT
    COUNT(CASE WHEN "Mission_Outcome" = 'Success' THEN 1 END) AS Success_Count,
    COUNT(CASE WHEN "Mission_Outcome" like 'Failure%' THEN 1 END) AS Failure_Count
FROM spacextable;
```

- The cases of Mission outcome are summed using the Count function and displayed.

Boosters Carried Maximum Payload

- The query is:

```
%sql select distinct booster_version from spacextable where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextable)
```

- List of the names of the booster which have carried the maximum payload mass

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 of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Landing_Outcome	Outcome_Count
No attempt	2
Failure (drone ship)	2
Success (ground pad)	1
Precluded (drone ship)	1
Controlled (ocean)	1

- The query is as shown

```
%%sql

SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS Outcome_Count
FROM spacextable
WHERE Date BETWEEN '2015-01-01' AND '2015-12-31'
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
```

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

- Rank of 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

- Query

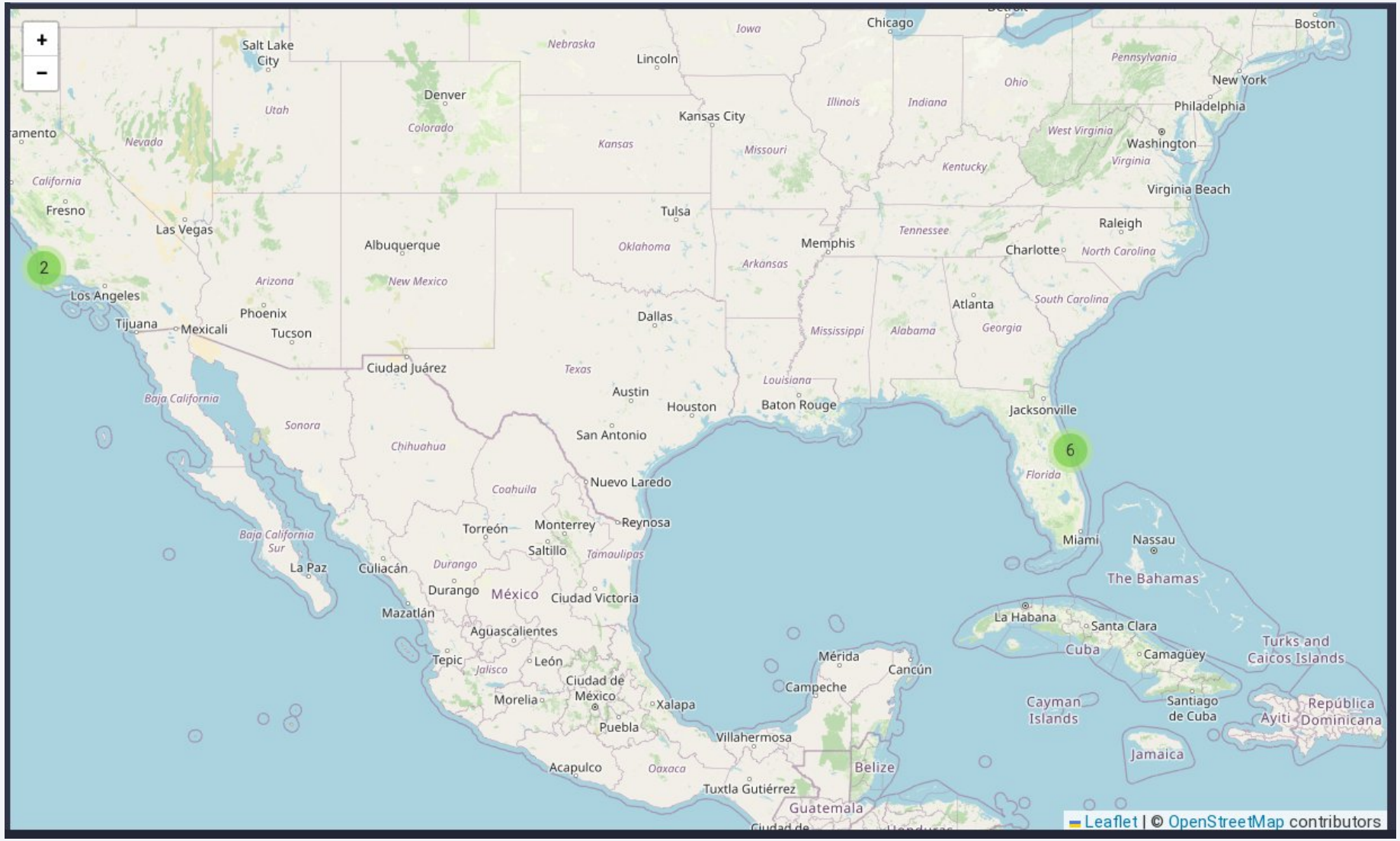
```
%%sql

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

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

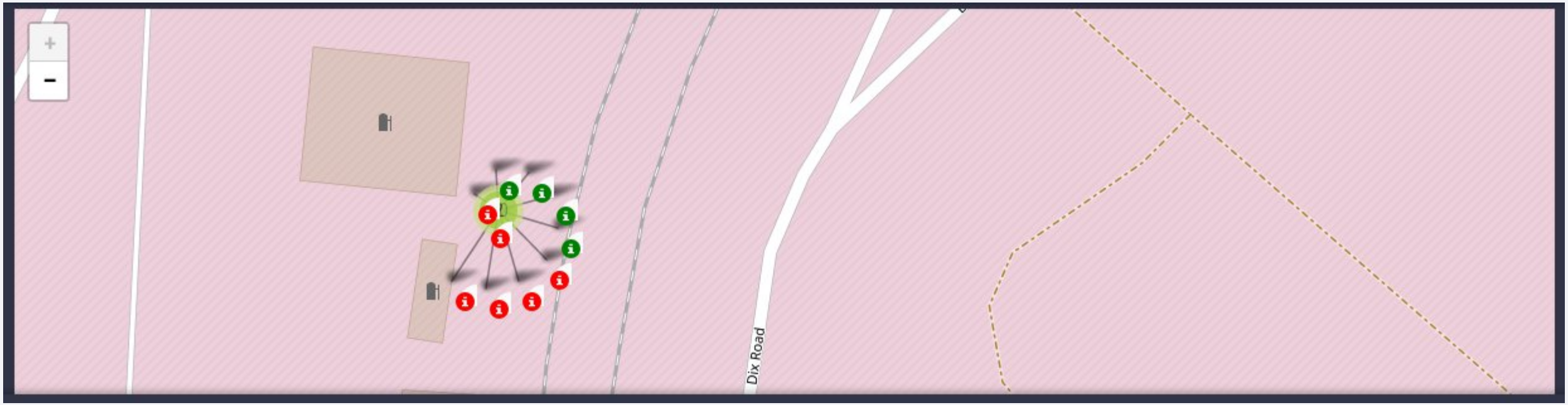
Section 3

Launch Sites Proximities Analysis

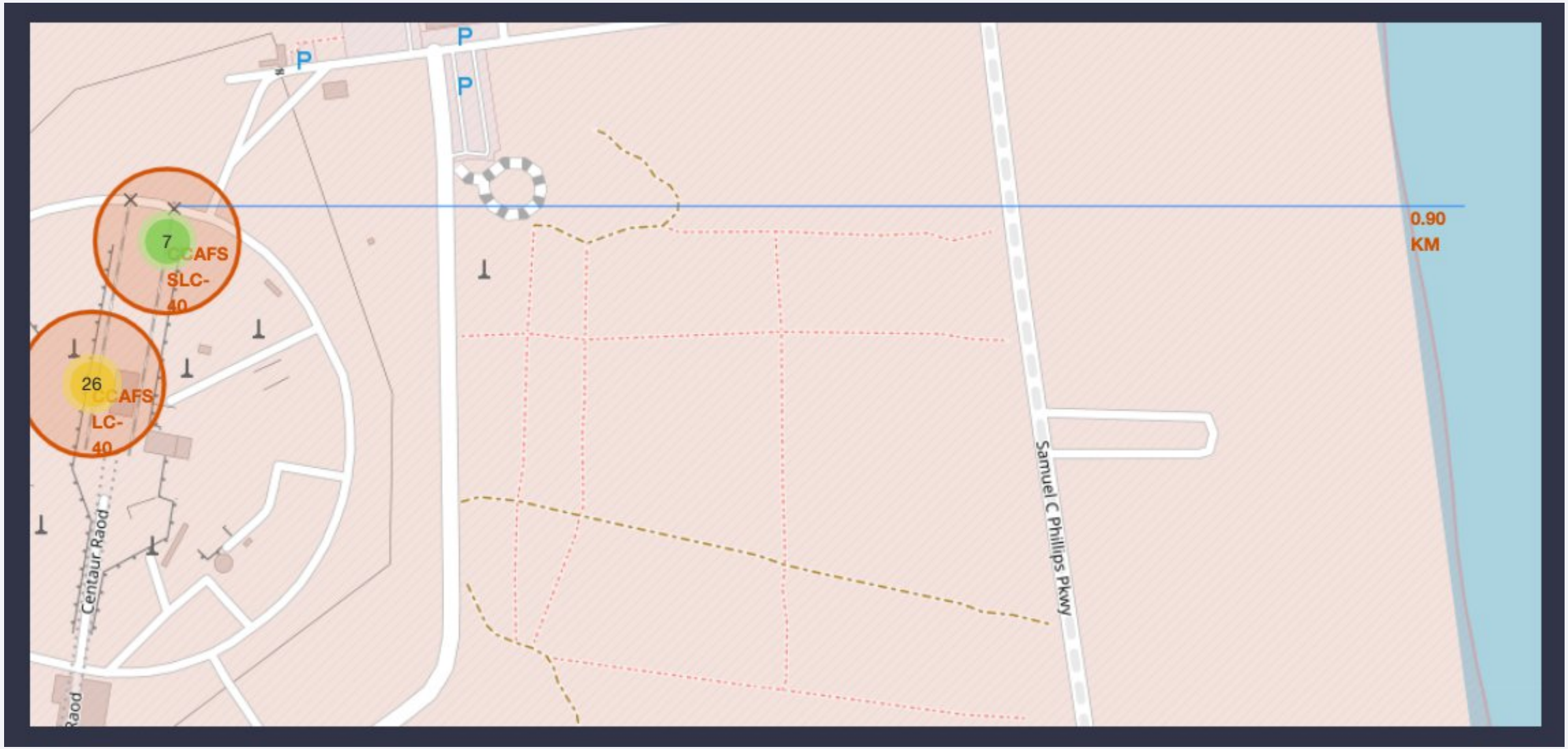
Map of Launch Sites

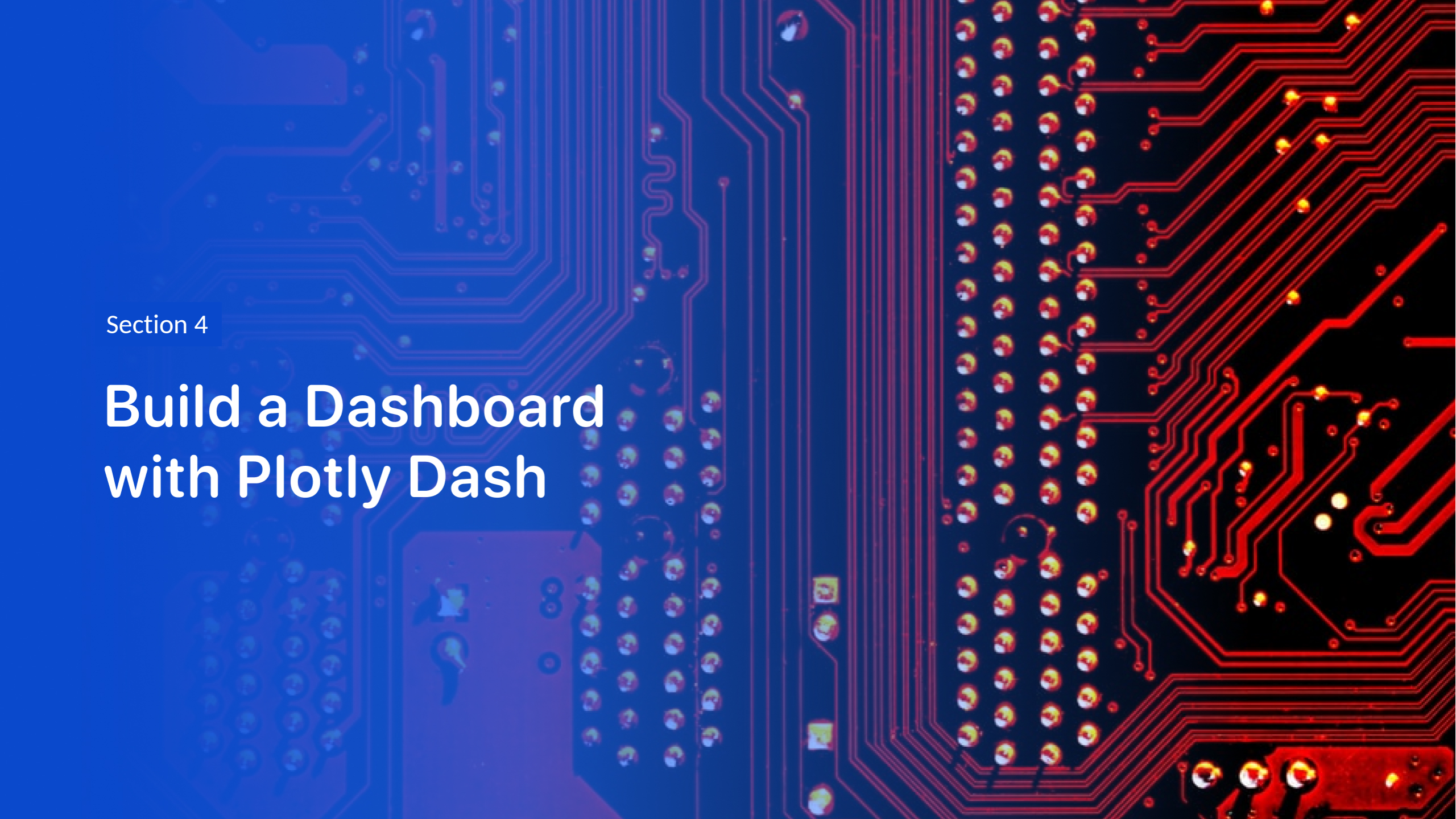


Color Labelled Launch Outcomes

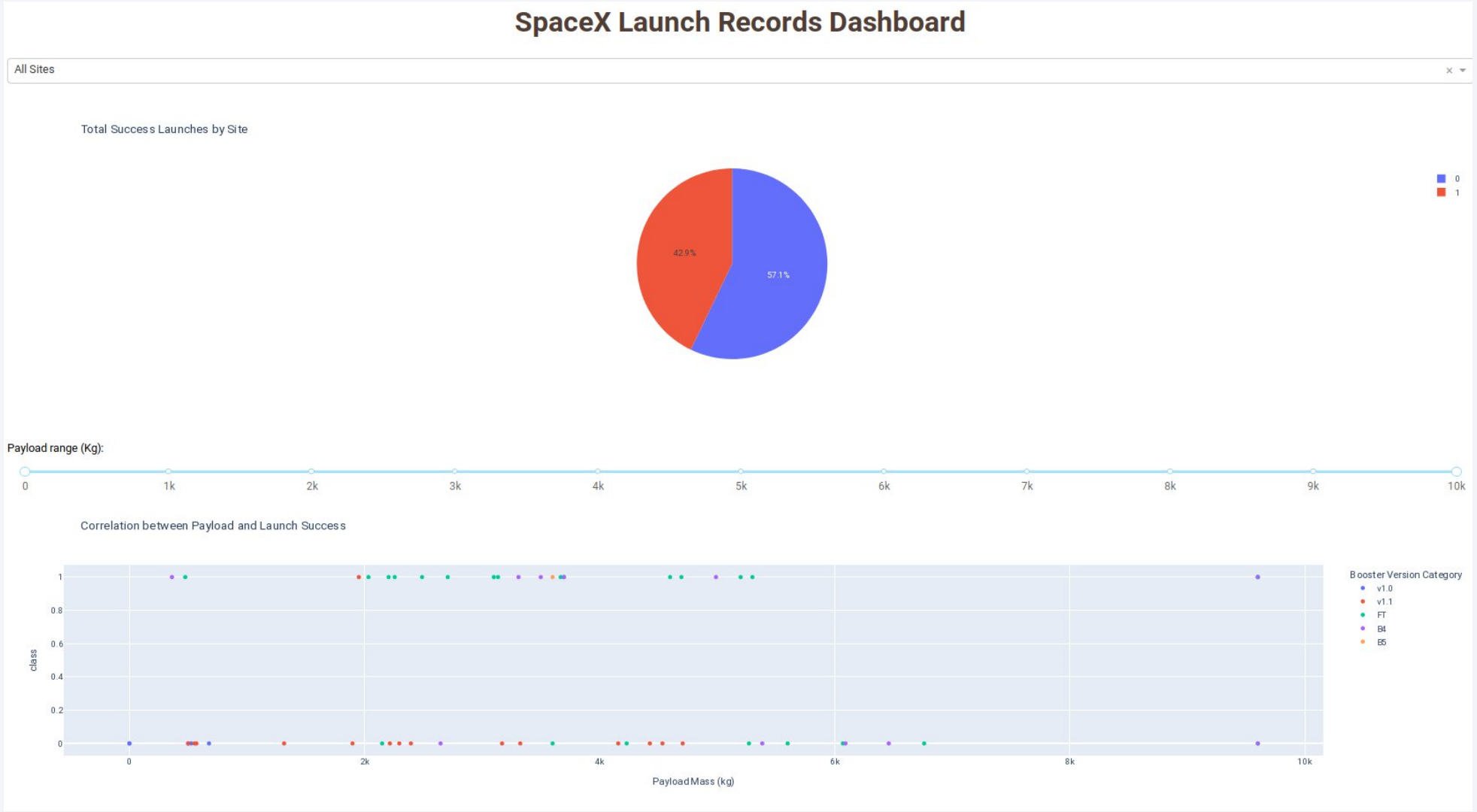


Points of Interest

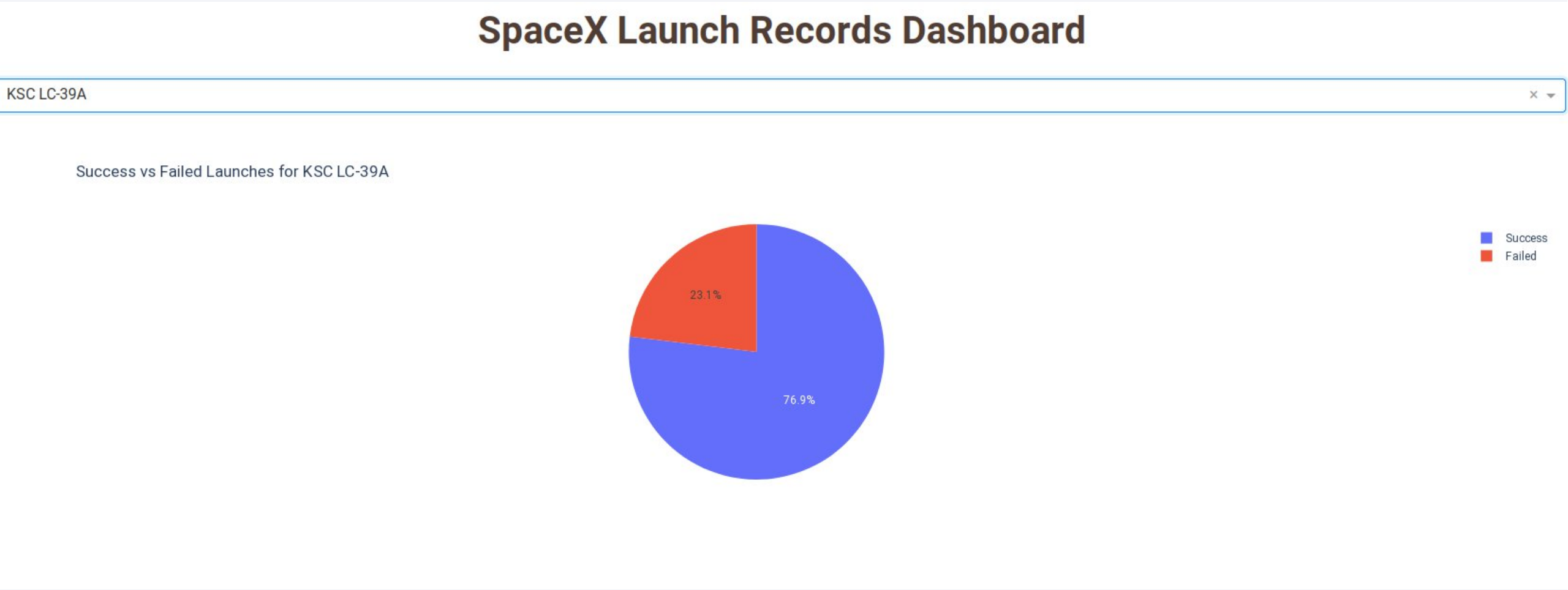



Section 4
Build a Dashboard with Plotly Dash

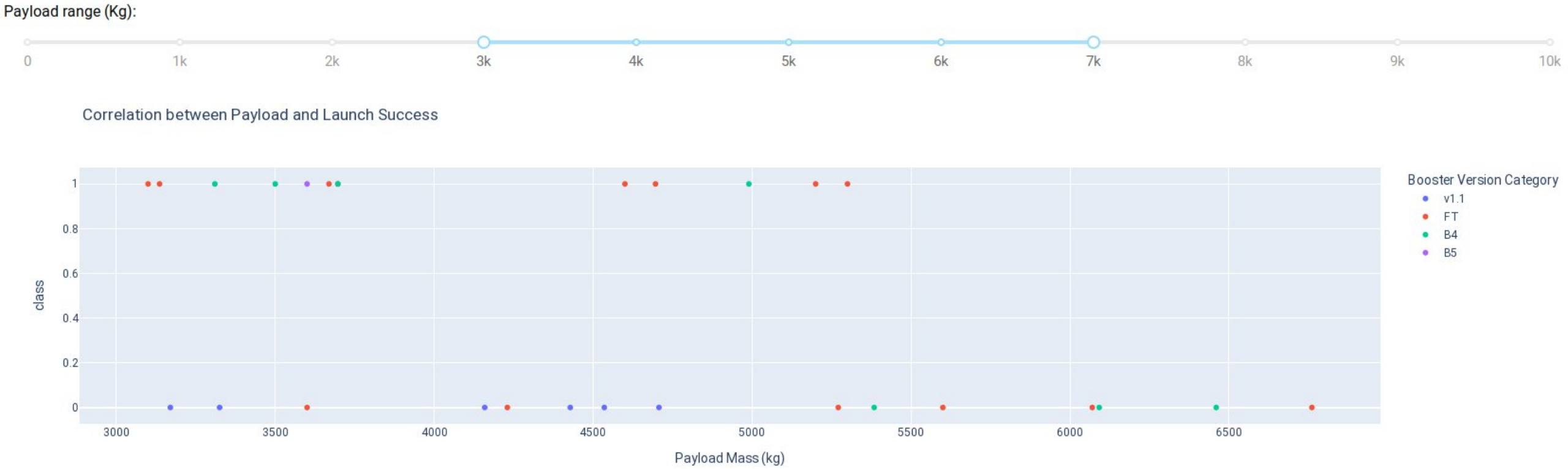
Total Dashboard



Pie-chart with Highest Success Ratio



Payload vs. Launch Outcome

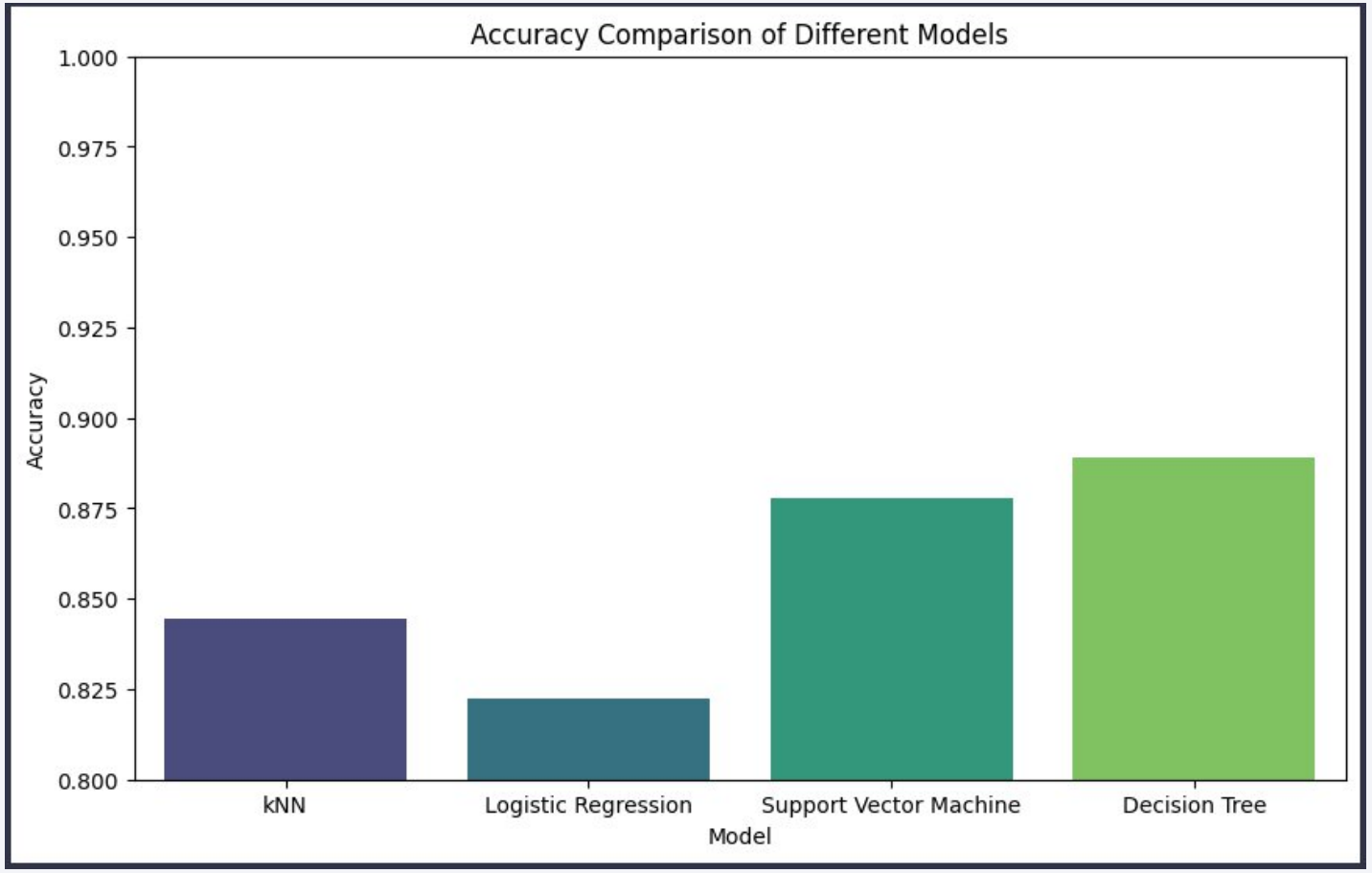


Section 5

Predictive Analysis (Classification)

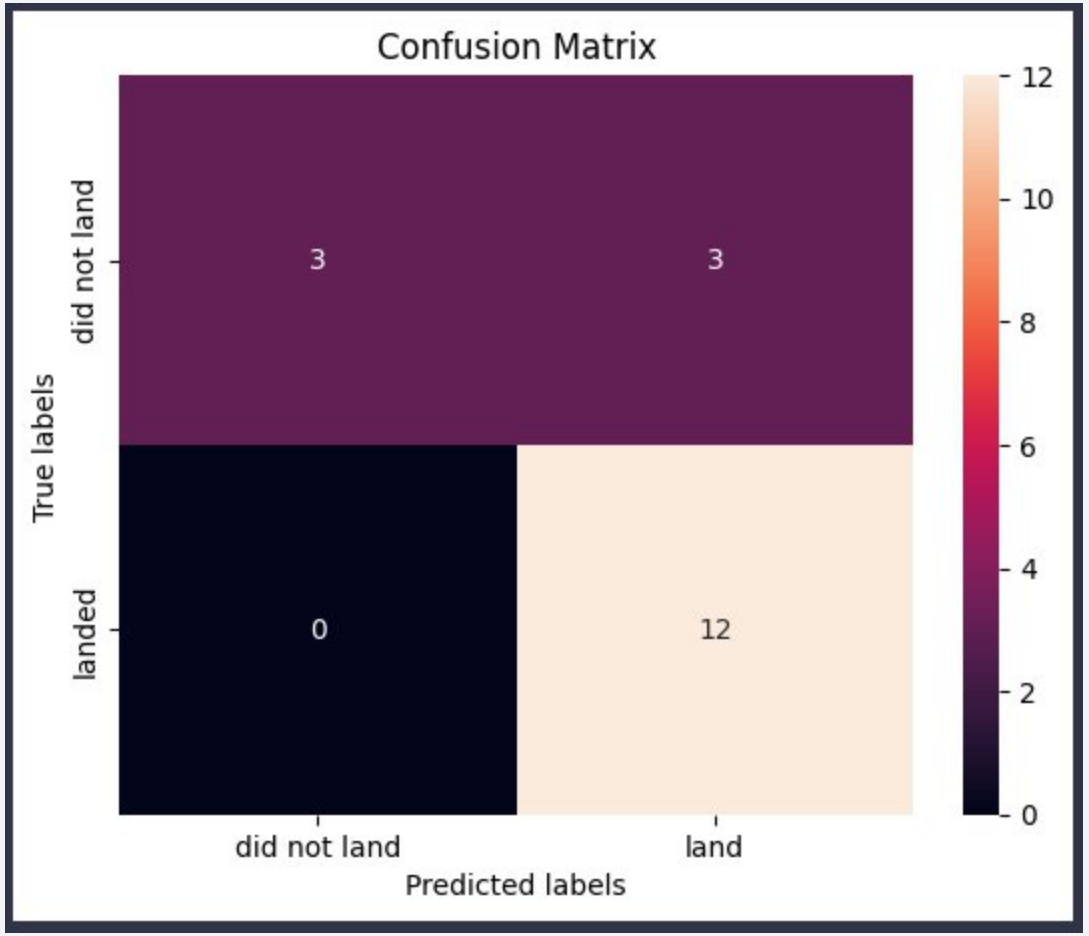
Classification Accuracy

- Decision tree has the highest accuracy



Confusion Matrix

- Confusion matrix for Decision tree.



Conclusions

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

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!