IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Belchenko Maryna
20.08.2024

Outline

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

Executive Summary

Summary of methodologies

I employed a comprehensive data science approach, starting with data collection via the SpaceX API. The data was then preprocessed to handle missing values and duplicates, followed by normalization and transformation into a suitable format. Exploratory data analysis was conducted using visualization tools like Matplotlib and Seaborn, along with SQL queries to uncover insights. Interactive visual analytics were performed using Folium for mapping and Plotly Dash for dashboard creation. Finally, classification models were developed, tuned, and evaluated to predict rocket landing success.

Summary of all results

The analysis revealed key insights into the factors influencing rocket landing success. Visualizations highlighted important trends in launch data, while SQL queries provided detailed breakdowns of performance metrics. The interactive maps and dashboards allowed for a deeper exploration of geographical data. The classification models achieved high accuracy, with the Decision Tree model performing best in predicting landing outcomes. Overall, the project successfully demonstrated the application of data science techniques to solve real-world aerospace challenges.

Introduction

Project background and context

The project focuses on analyzing and predicting the outcomes of SpaceX rocket landings, a critical aspect of modern space exploration. As SpaceX continues to innovate in the aerospace industry, understanding the factors that contribute to successful rocket landings is crucial. This project aims to provide data-driven insights into these factors, leveraging advanced data science methodologies.

Problems you want to find answers

The primary objectives of this project include predicting the success of SpaceX rocket landings and identifying the key factors that influence these outcomes. By analyzing historical launch data, the project seeks to answer questions such as: What are the critical determinants of a successful landing? How can predictive models improve our understanding and forecasting of landing success? The findings from this analysis are intended to contribute to SpaceX's ongoing efforts to enhance the reliability and efficiency of their missions.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

Data was collected using the SpaceX API, which provides information on rocket launches and landings. API requests allowed us to obtain data on launch dates, rocket types, landing outcomes, and other key metrics.

- Perform data wrangling

Data preprocessing was performed using the pandas and numpy libraries. During preprocessing, the data was cleaned of missing values and duplicates, and then transformed into the required format. The data was also normalized and structured for further analysis.

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

Classification models (Decision Tree, KNN, SVM) were developed and trained to predict rocket landing success. Model hyperparameters were tuned using GridSearchCV, and the models were evaluated based on accuracy on the test data.

Data Collection

Describe how data sets were collected:

The data collection process involved acquiring launch data from multiple sources. The primary source was the SpaceX API, which provided detailed records of launches, including rocket specifications, payloads, and landing outcomes. Additionally, web scraping techniques were employed to gather supplementary data from relevant aerospace websites. The collected data was then consolidated into a unified dataset for analysis.

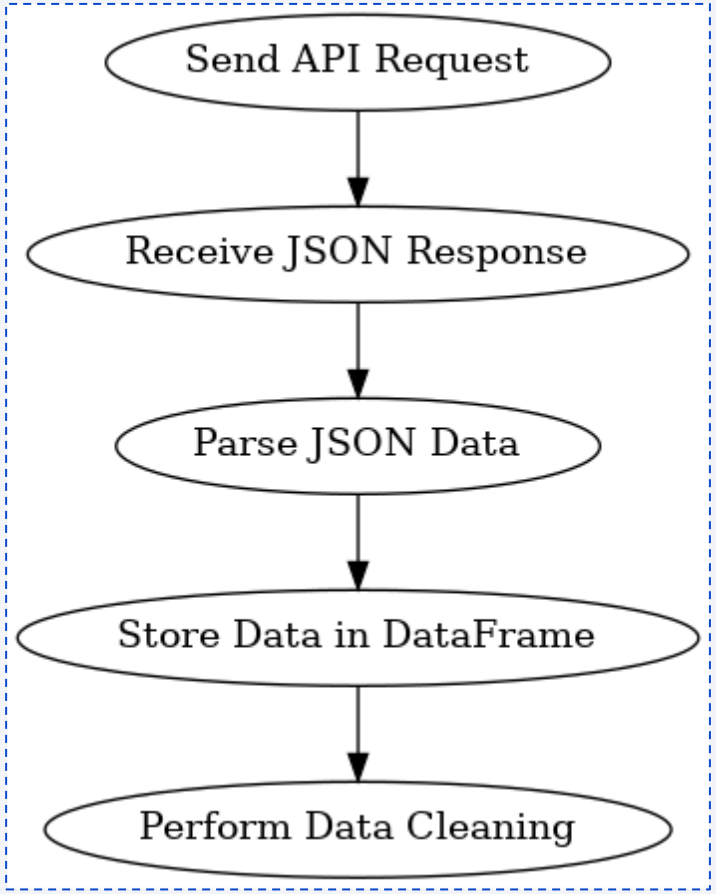
Data Collection – SpaceX API

SpaceX API was utilized to fetch real-time data on rocket launches.

Key data retrieved includes:

- rocket specifications
- launch details
- payload information

The data was collected using REST API calls and processed into a pandas DataFrame.



https://github.com/MaryBelch/SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb?short_path=5ada7c4

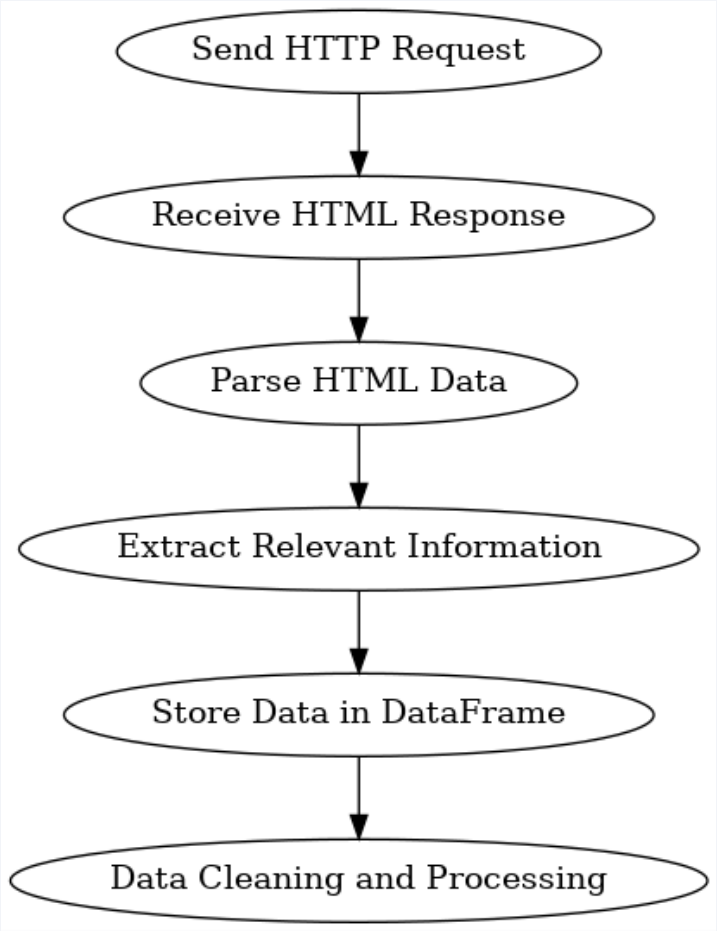
Data Collection - Scraping

Web scraping was employed to gather data not available via API.

Data extracted includes historical launch outcomes and additional launch site details.

Tools used: Python's BeautifulSoup and Requests libraries.

Data was cleaned and stored for further analysis.



https://github.com/MaryBelch/SpaceX/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

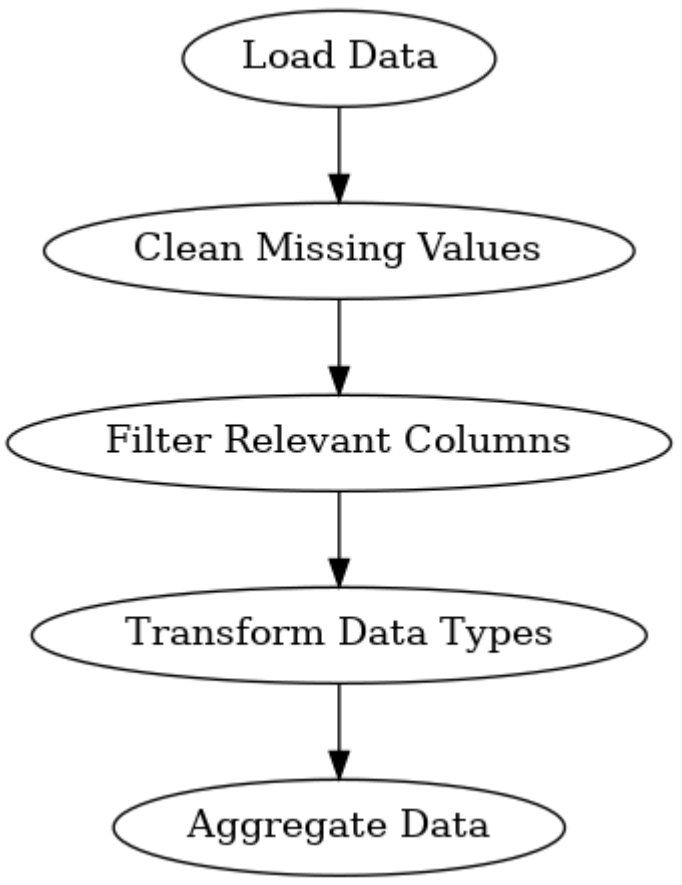
Data Wrangling

Data wrangling involved cleaning, transforming, and organizing the collected data.

Steps included:

- handling missing values
- converting data types
- filtering and merging datasets

Ensured data consistency for analysis and model training.



https://github.com/MaryBelch/SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

In my work, I used:

Chart 1: Distribution of Landing Outcomes by LaunchSite. To understand the success rate of landings at different launch sites.

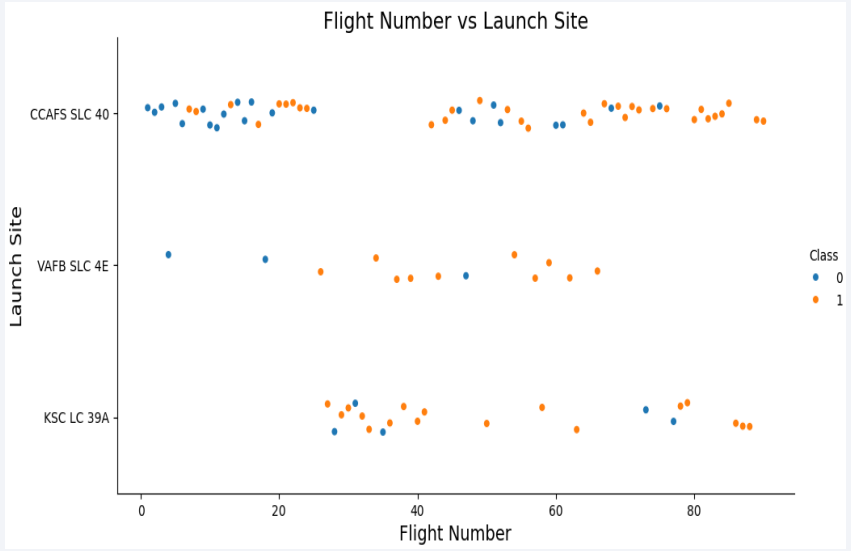


Chart 2: Payload Mass vs. Outcome. To explore the relationship between payload mass and landing success.

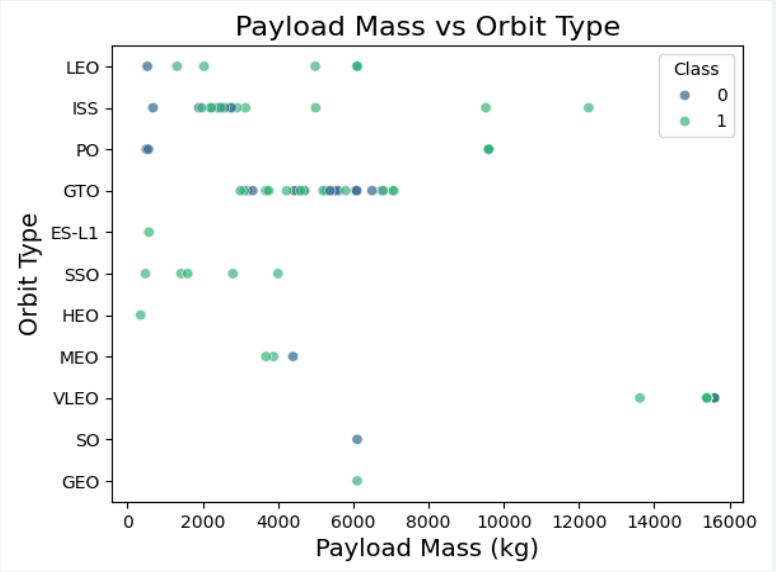
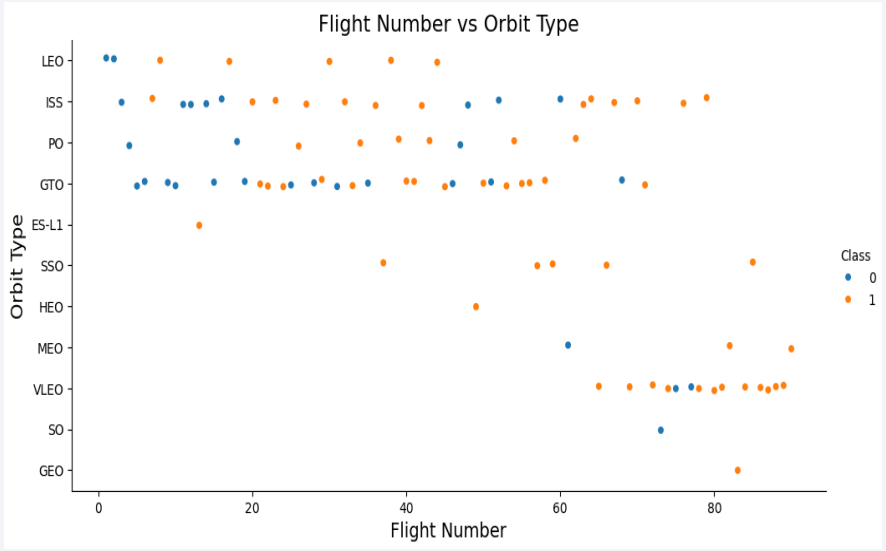


Chart 3: Number of Flights per Booster Version. To analyze the reusability of different booster versions.



https://github.com/MaryBelch/SpaceX/blob/main/edadataviz.ipynb

EDA with SQL

Executive Summary

Launch Sites: Retrieved unique names of launch sites used in SpaceX missions.

Payload by Launch Site: Displayed records of payloads launched from sites beginning with 'CCA'.

Total Payload Mass: Calculated total payload mass carried by NASA (CRS) boosters.

Average Payload Mass: Determined average payload mass for boosters of version F9 v1.1.

First Successful Landing: Identified the date of the first successful landing on a ground pad.

Boosters by Payload Mass: Listed boosters that successfully landed on drone ships with payloads between 4000 and 6000 kg.

Mission Outcomes: Tallied the number of successful and failed mission outcomes.

Max Payload Boosters: Identified boosters that carried the maximum payload mass.

2015 Failures: Listed failures in drone ship landings in 2015 with corresponding months and launch sites.

Landing Outcome Ranking: Ranked landing outcomes by frequency between 2010-06-04 and 2017-03-20.

https://github.com/MaryBelch/SpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Summary of Map Objects Created and Their Purpose:

Markers. Plotted markers for each launch site. To visually identify the location of each launch site on the map.

Circles. Added circles around each launch site. To highlight and provide a visual context for the area around each launch site.

Colored Markers (Green/Red). Used colored markers to indicate the success (green) or failure (red) of each launch. To easily visualize and differentiate between successful and failed launches on the map.

Marker Clusters. Implemented marker clusters for launch records with the same coordinates. To simplify the map and reduce clutter by grouping markers at the same location.

PolyLines. Drew lines between launch sites and points of interest (e.g., coastline, cities, railways). To analyze the proximity of launch sites to these points of interest and understand their spatial relationships.

https://github.com/MaryBelch/SpaceX/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

Summary of Plots/Graphs and Interactions Added to the Dashboard:

Pie Chart (Success Rate). Added a pie chart to display the success rate of launches for selected sites. To visualize the proportion of successful vs. failed launches at each site or across all sites.

Range Slider (Payload Mass). Implemented a range slider to select payload mass ranges.
To filter the scatter plot based on payload mass and analyze how payload size correlates with launch outcomes.

Scatter Plot (Payload Mass vs. Outcome). Added a scatter plot to show the relationship between payload mass and launch outcomes.
To investigate how different payload sizes affect the success rate of launches and to differentiate by booster version.

Dropdown Menu (Launch Site Selection). Included a dropdown menu for selecting different launch sites.
Why: To enable users to filter and view data specific to individual launch sites or aggregate data for all sites.

Interactive Elements. Added interactive components such as dropdowns and sliders.
To allow users to dynamically interact with the data, customize their view, and perform real-time visual analytics.

https://github.com/MaryBelch/SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

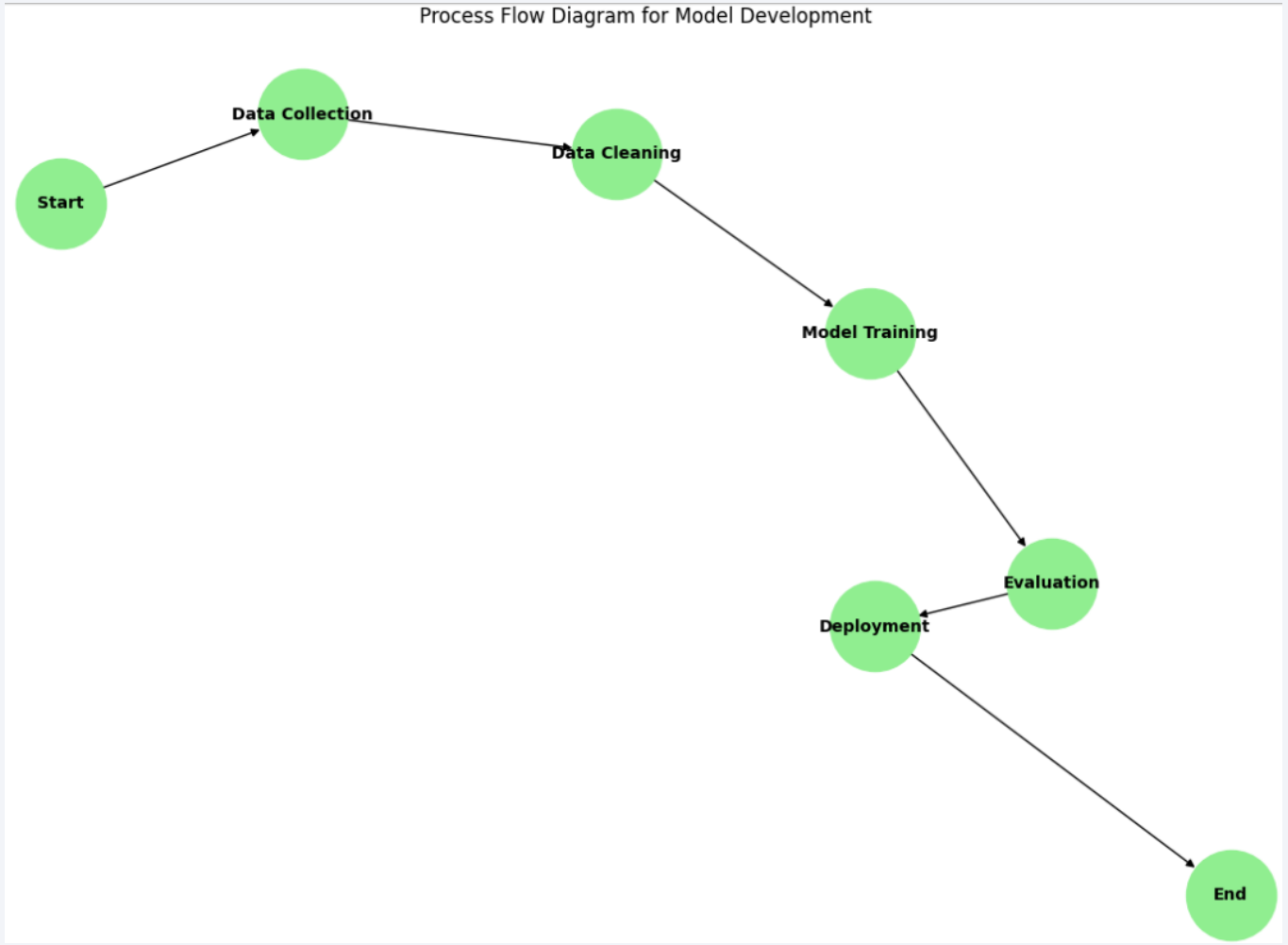
Data Preparation includes class creation, standardization, and data splitting.

Model Training covers training logistic regression, SVM, and decision tree models.

Model Evaluation involves evaluating models, including constructing confusion matrices.

Model Improvement describes improving models if necessary.

Final Model Selection determines the best model for final selection.



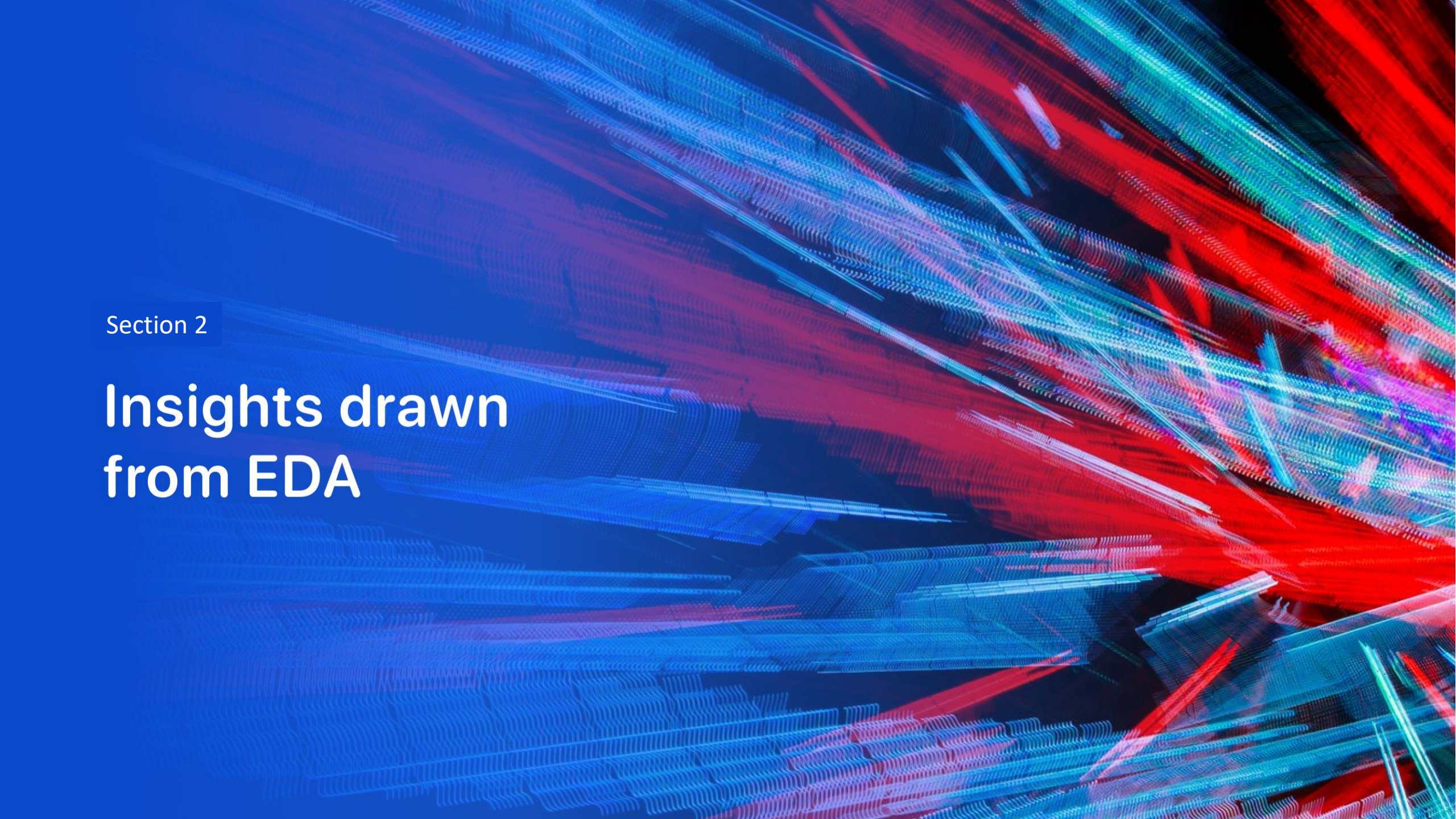
https://github.com/MaryBelch/SpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

In this capstone project, I tackled the challenge of predicting the successful landing of SpaceX's Falcon 9 first stage. My goal was to determine the likelihood of a successful landing, which is crucial for assessing the cost-efficiency of SpaceX's reusable rocket technology. By predicting landing success, I can help gauge the competitiveness of SpaceX's launches against other providers.

I developed Python code to manipulate and analyze data using Pandas, transforming a JSON file into a usable Pandas DataFrame. This involved loading and cleaning the dataset to extract meaningful insights. Additionally, I created and shared a Jupyter notebook on GitHub, demonstrating my ability to effectively communicate my findings.

Through this project, I applied data science methodologies to frame and address a real-world business problem. My analysis not only explored key trends and patterns but also improved the model's accuracy through iterative refinement. This hands-on experience enhanced my skills in data preparation, model training, and evaluation, showcasing my ability to tackle complex data-driven challenges.

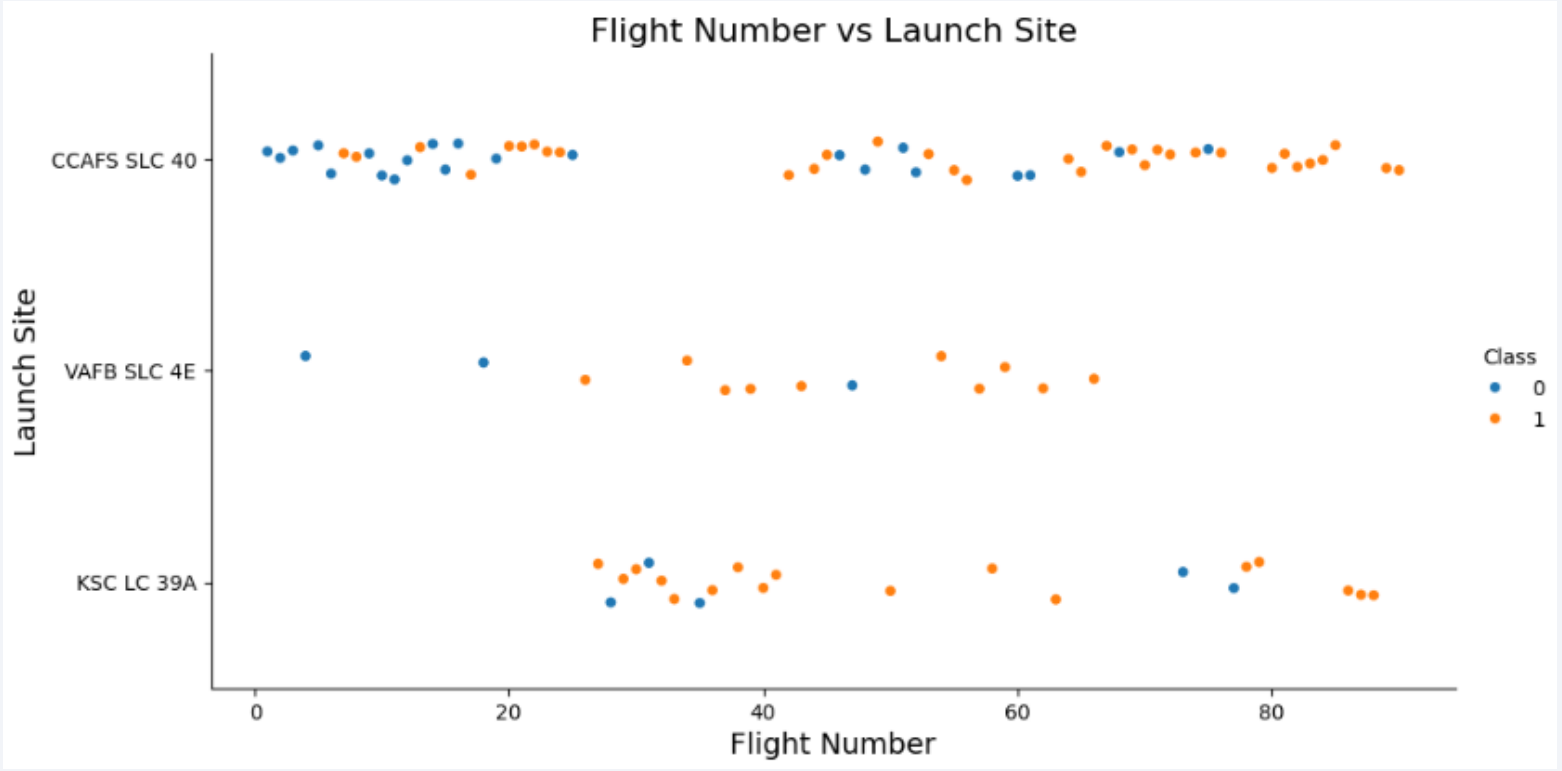

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

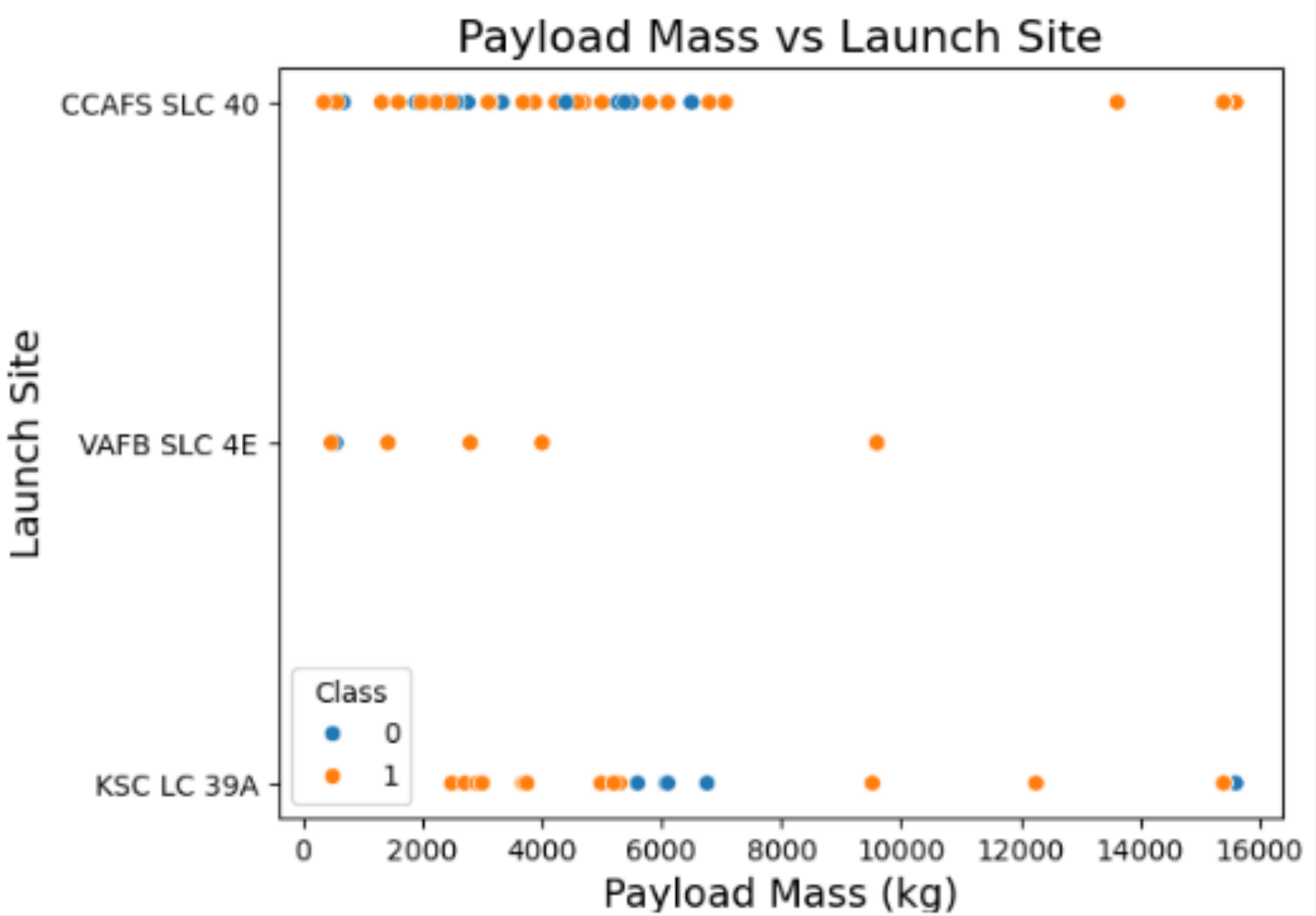
According to the plot above, it's possible to verify that the best launch site nowadays is CCAF5 SLC 40, where most of recent launches were successful;

In second place VAFB SLC 4E and third place KSC LC 39A;

It's also possible to see that the general success rate improved over time.



Payload vs. Launch Site



Payloads over 9,000kg (about the weight of a school bus) have excellent success rate;

Payloads over 12,000kg seems to be possible only on CCAFS SLC 40 and KSC LC 39A launch sites.

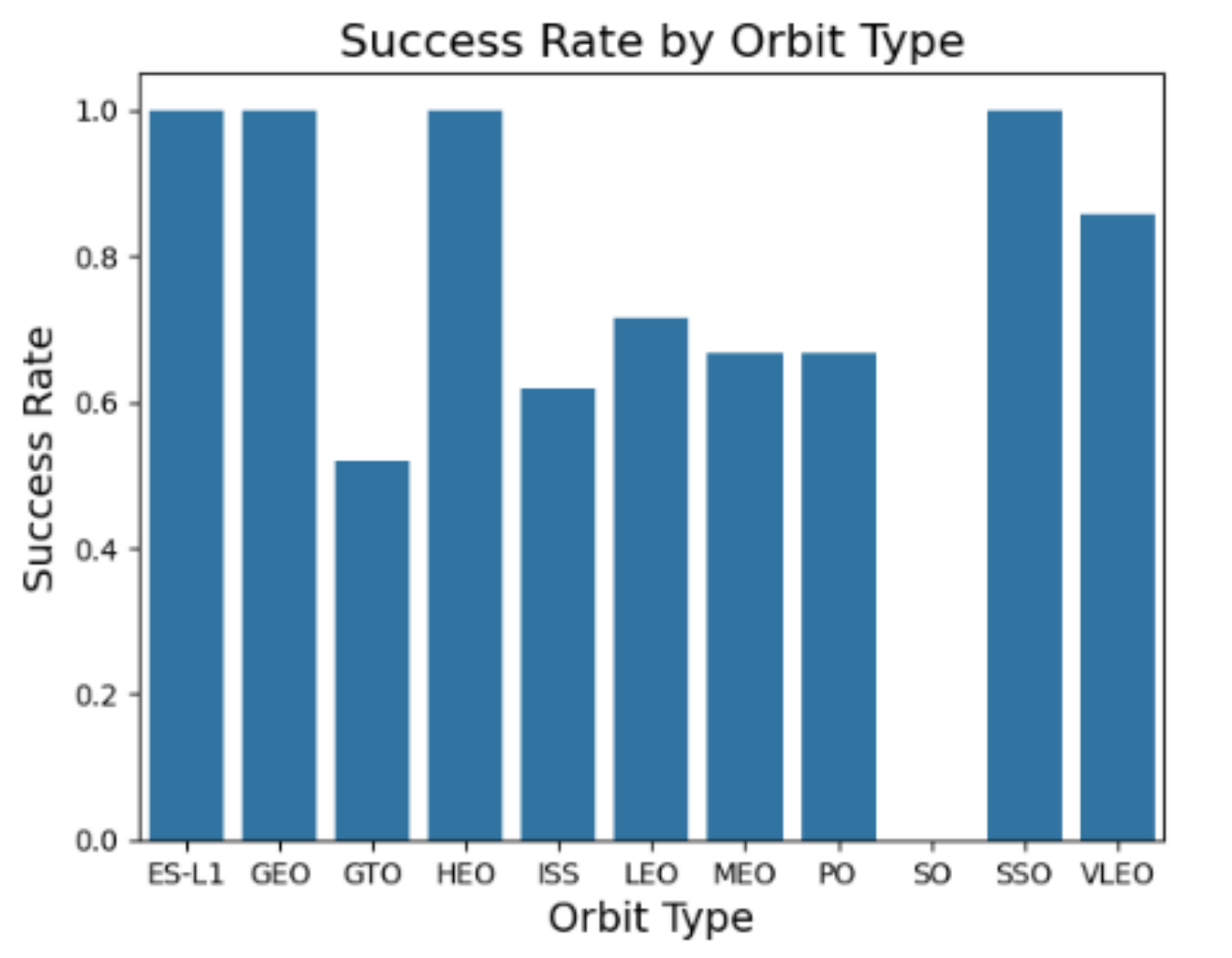
Success Rate vs. Orbit Type

The biggest success rates happens to orbits:

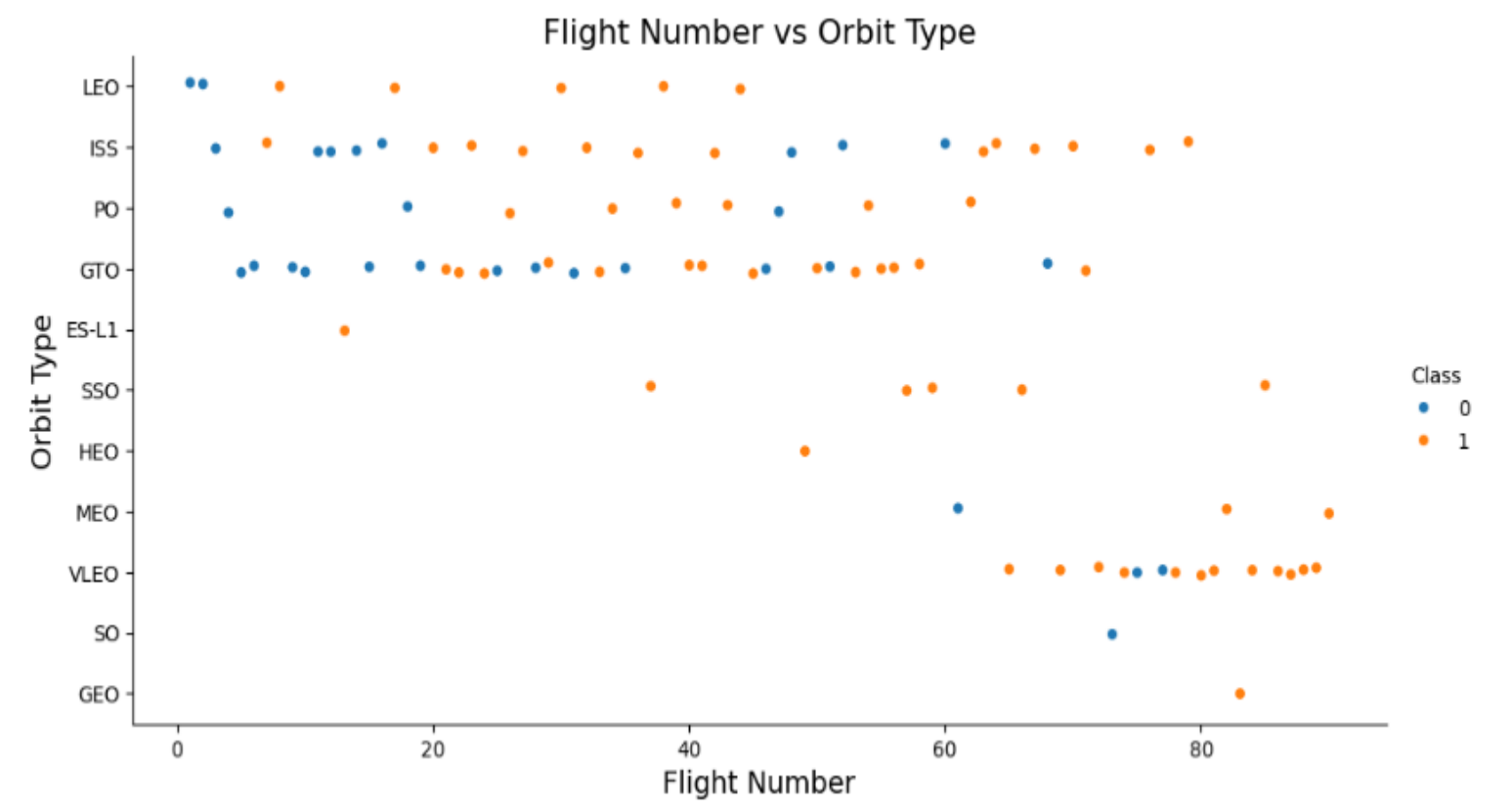
- ES-L1;
- GEO;
- HEO; and
- SSO.

Followed by:

- VLEO (above 80%); and
- LFO (above 70%).



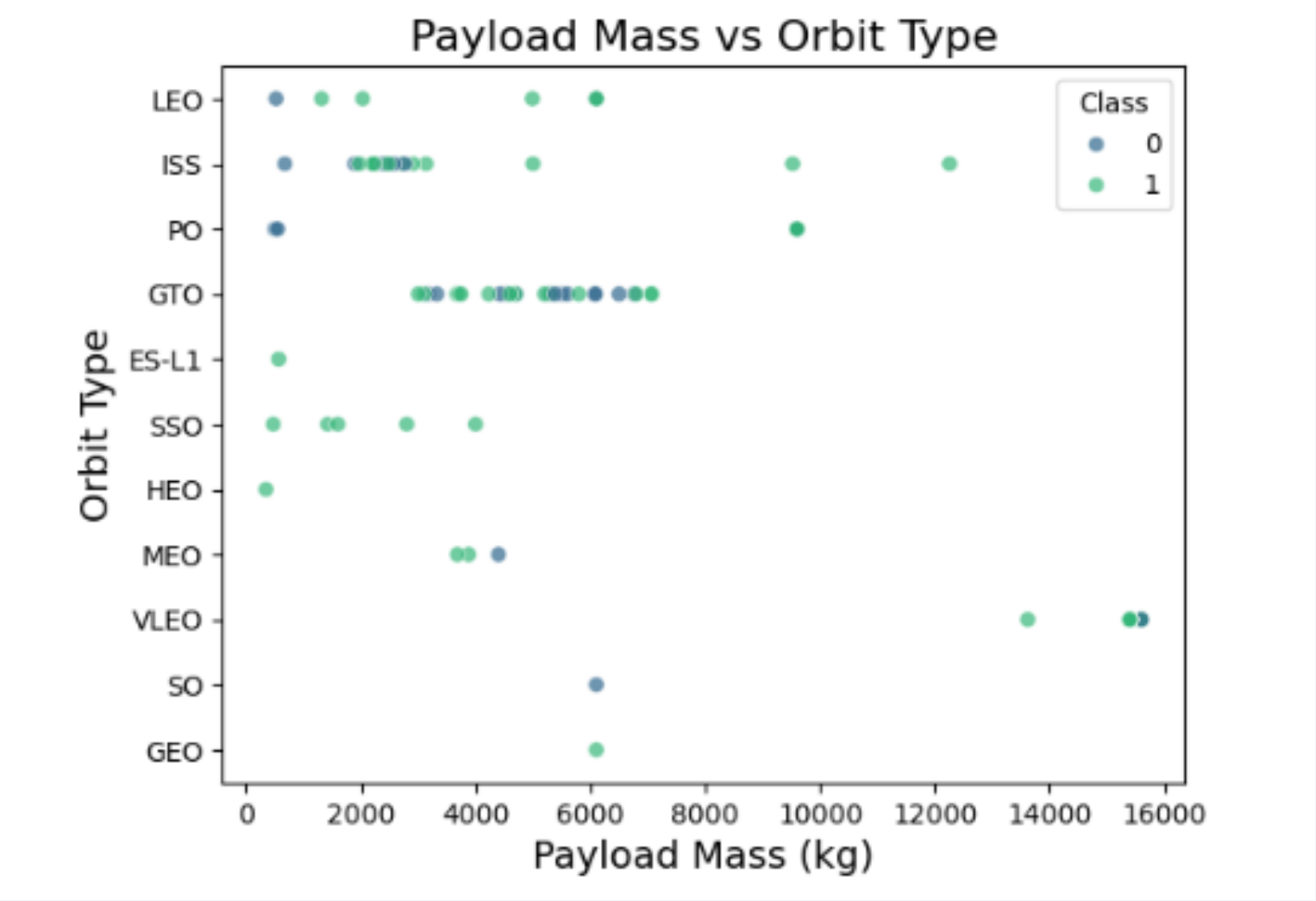
Flight Number vs. Orbit Type



Apparently, success rate improved over time to all orbits;

VLEO orbit seems a new business opportunity, due to recent increase of its frequency.

Payload vs. Orbit Type

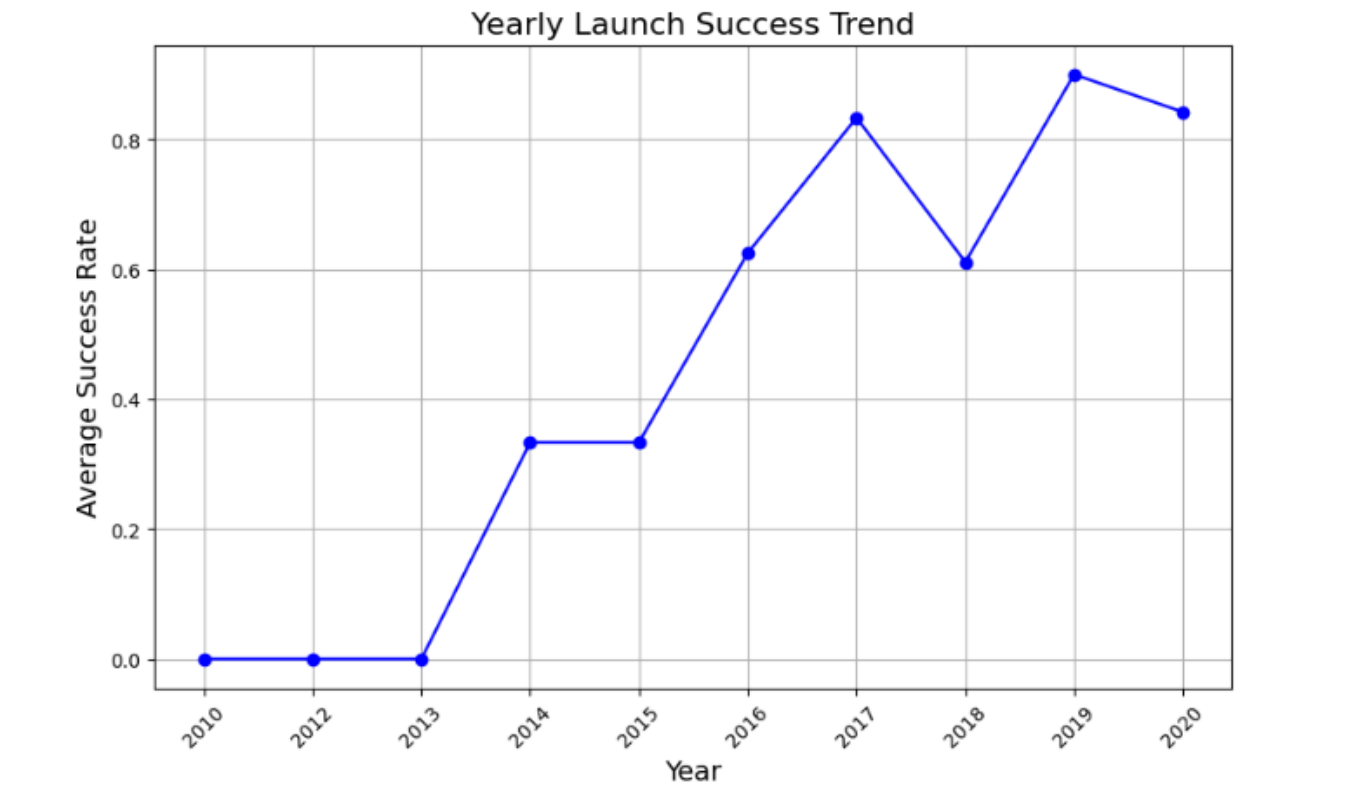


Apparently, there is no relation between payload and success rate to orbit GTO;

ISS orbit has the widest range of payload and a good rate of success;

There are few launches to the orbits SO and GEO.

Launch Success Yearly Trend



Success rate started increasing in 2013 and kept until 2020;

It seems that the first three years were a period of adjusts and improvement of technology.

All Launch Site Names

```
%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

According to data, there are four launch sites:

They are obtained by selecting unique occurrences of “launch_site” values from the dataset.

Launch Site Names Begin with 'CCA'

5 records where launch sites 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

Here we can see five samples of Cape Canaveral launches.

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%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

Total payload calculated above, by summing all payloads whose codes contain 'CRS', which corresponds to NASA.

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version 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

Filtering data by the booster version above and calculating the average payload mass we obtained the value of 2,928 kg.

First Successful Ground Landing Date

First successful landing outcome on ground pad:

Task 5

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

Hint:Use min function

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

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

First_Successful_Landing_Date
2015-12-22

By filtering data by successful landing outcome on ground pad and getting the minimum value for date it's possible to identify the first occurrence, that happened on 12/22/2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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 DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

Selecting distinct booster versions according to the filters above, these 4 are the result.

Total Number of Successful and Failure Mission Outcomes

Number of successful and failure mission outcomes:

Task 7

List the total number of successful and failure mission outcomes

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

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

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

Grouping mission outcomes and counting records for each group led us to the summary above.

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT "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

Boosters which have carried the maximum payload mass

These are the boosters which have carried the maximum payload mass registered in the dataset.

2015 Launch Records

Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

Task 9

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     substr("Date", 1, 4) AS Year,    CASE substr("Date", 6, 2)         WHEN '01' THEN 'January'         WHEN '02' THEN 'February'WHEN '03' THEN 'March'
```

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

Year	Month	Landing_Outcome	Booster_Version	Launch_Site
2015	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

The list above has the only two occurrences.

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

Ranking of all landing outcomes between the date 2010-06-04 and 2017-03-20:

Task 10

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 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.
```

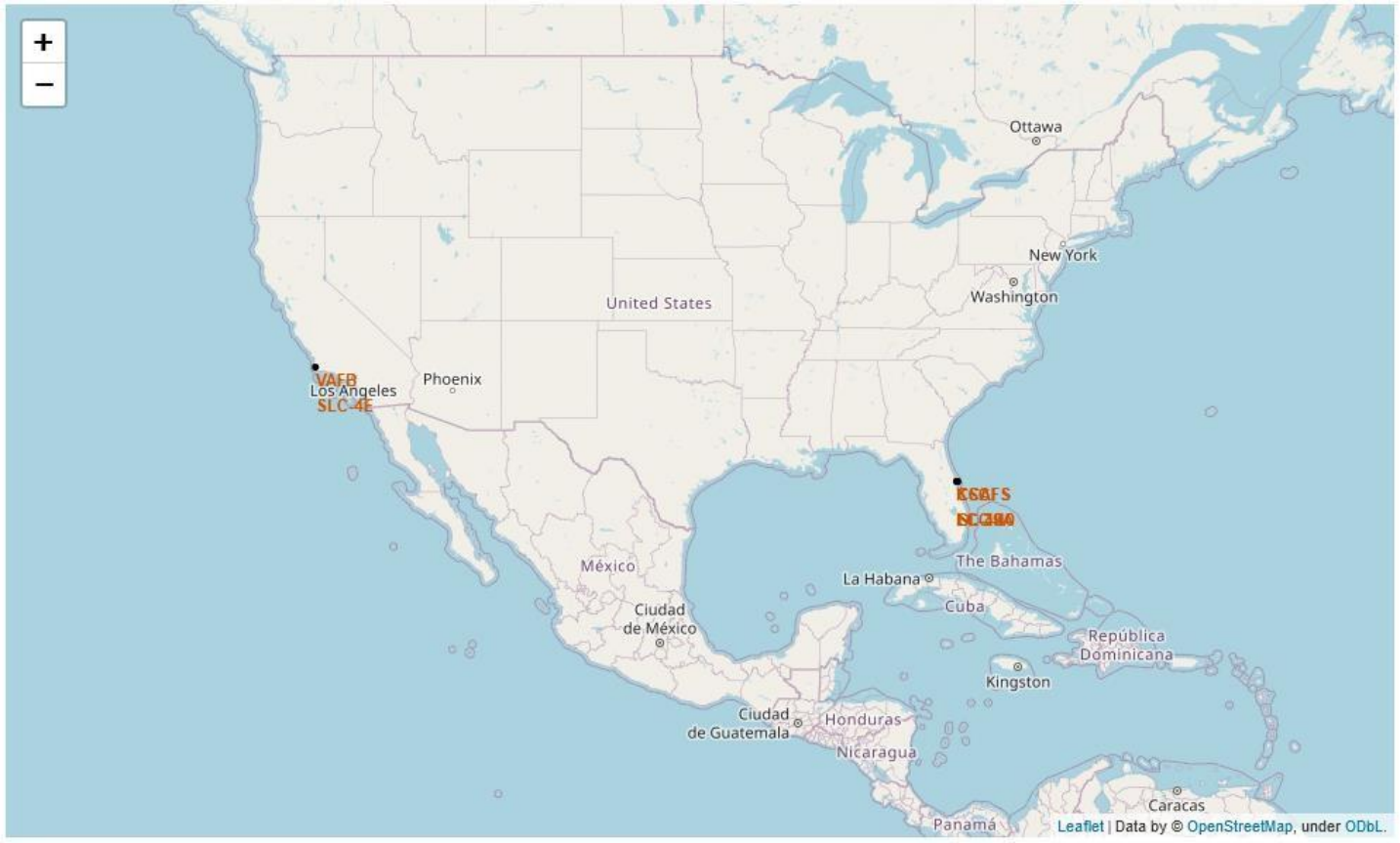
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

This view of data alerts us that “No attempt” must be taken in account.

Section 3

Launch Sites Proximities Analysis

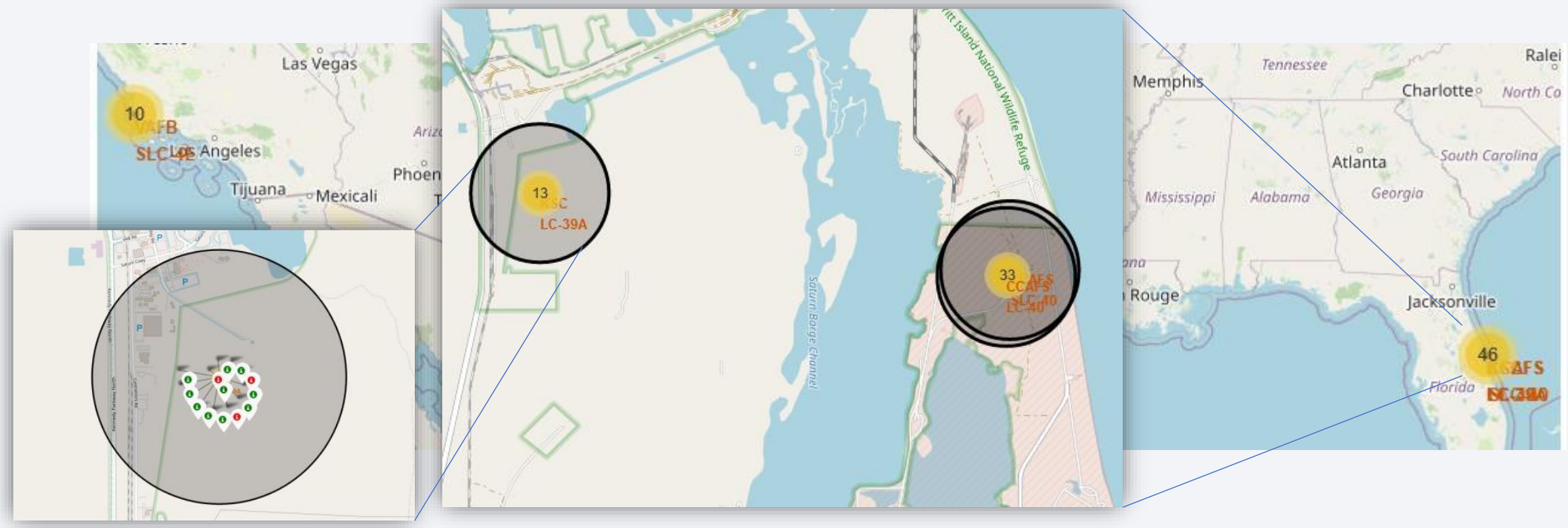
All launch sites



Launch sites are near sea, probably by safety, but not too far from roads and railroads.

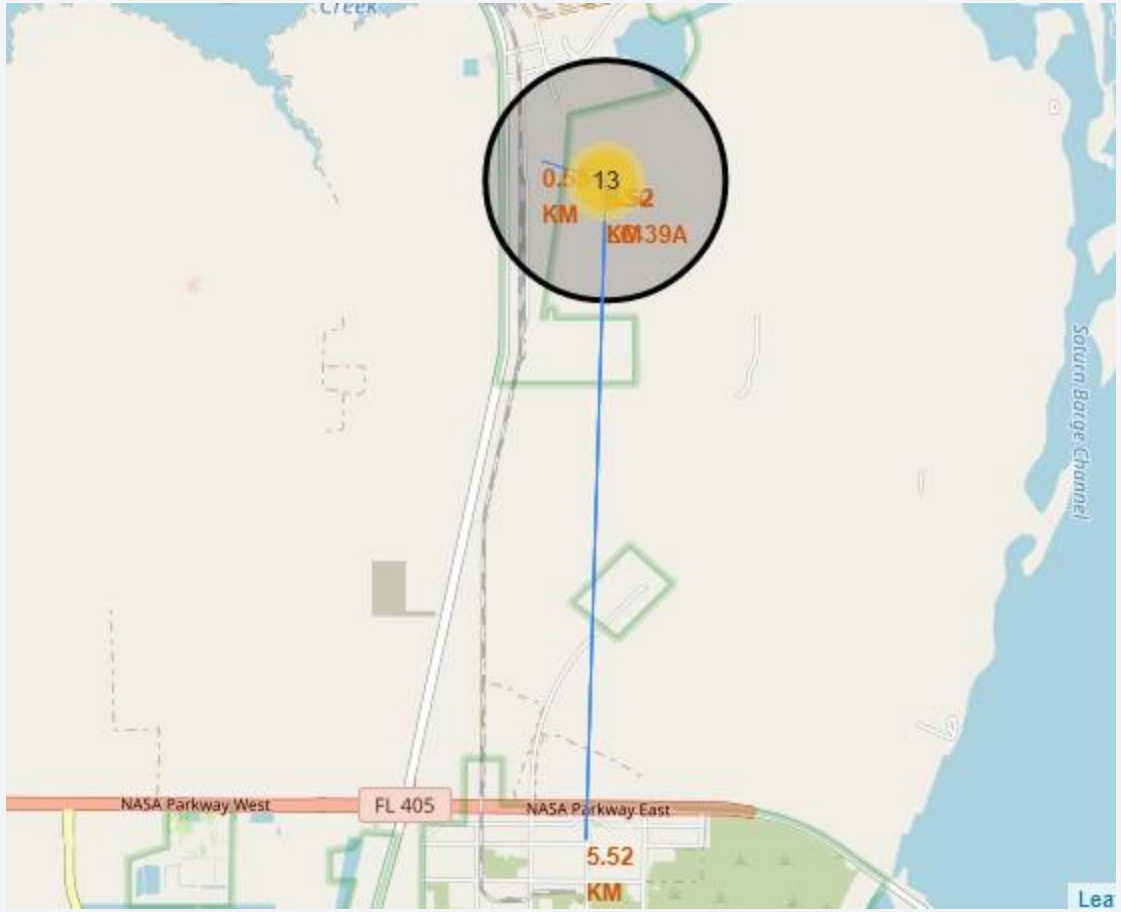
Launch Outcomes by Site

Example of KSC LC-39A launch site launch outcomes

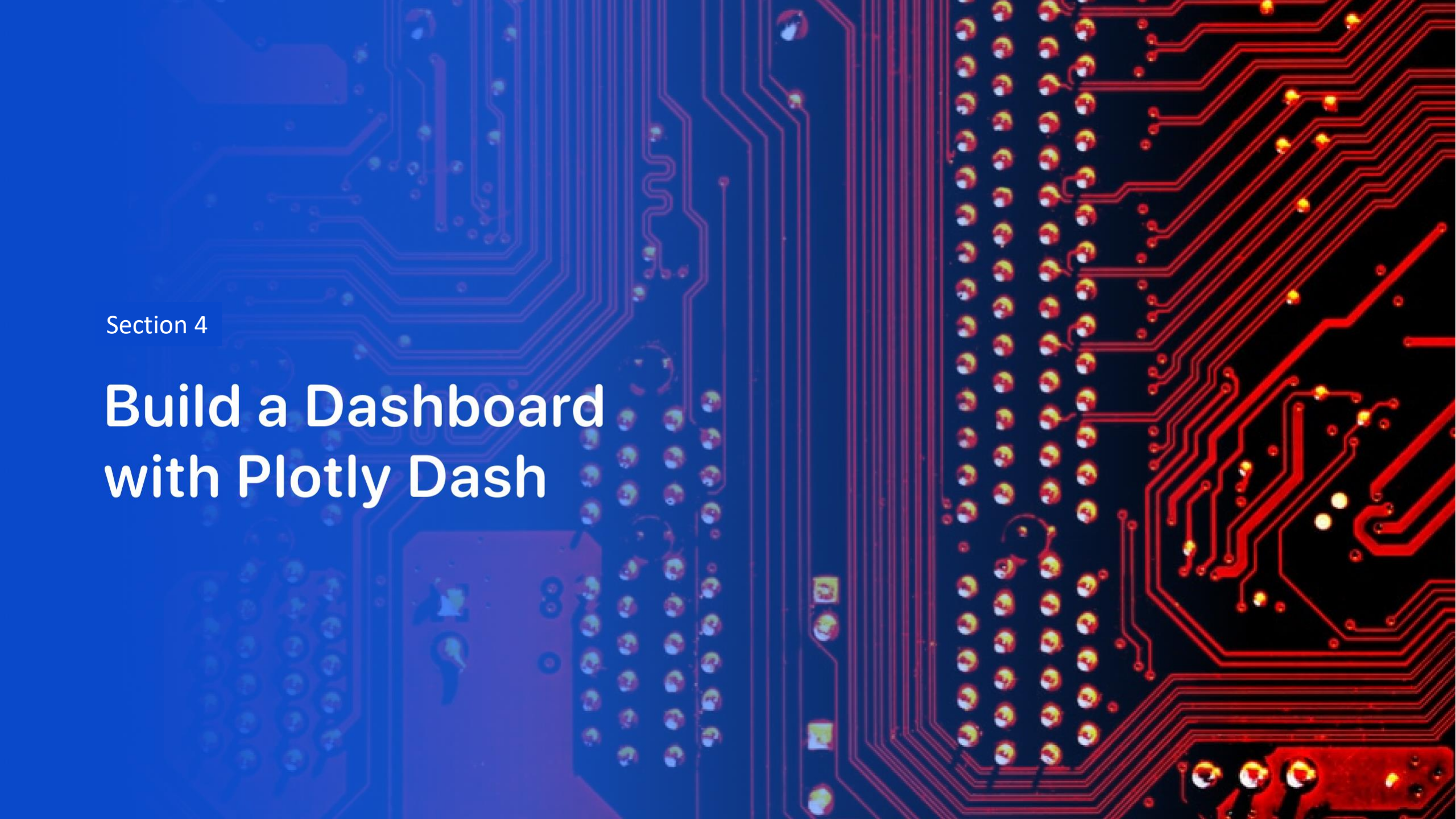


Green markers indicate successful and red ones indicate failure.

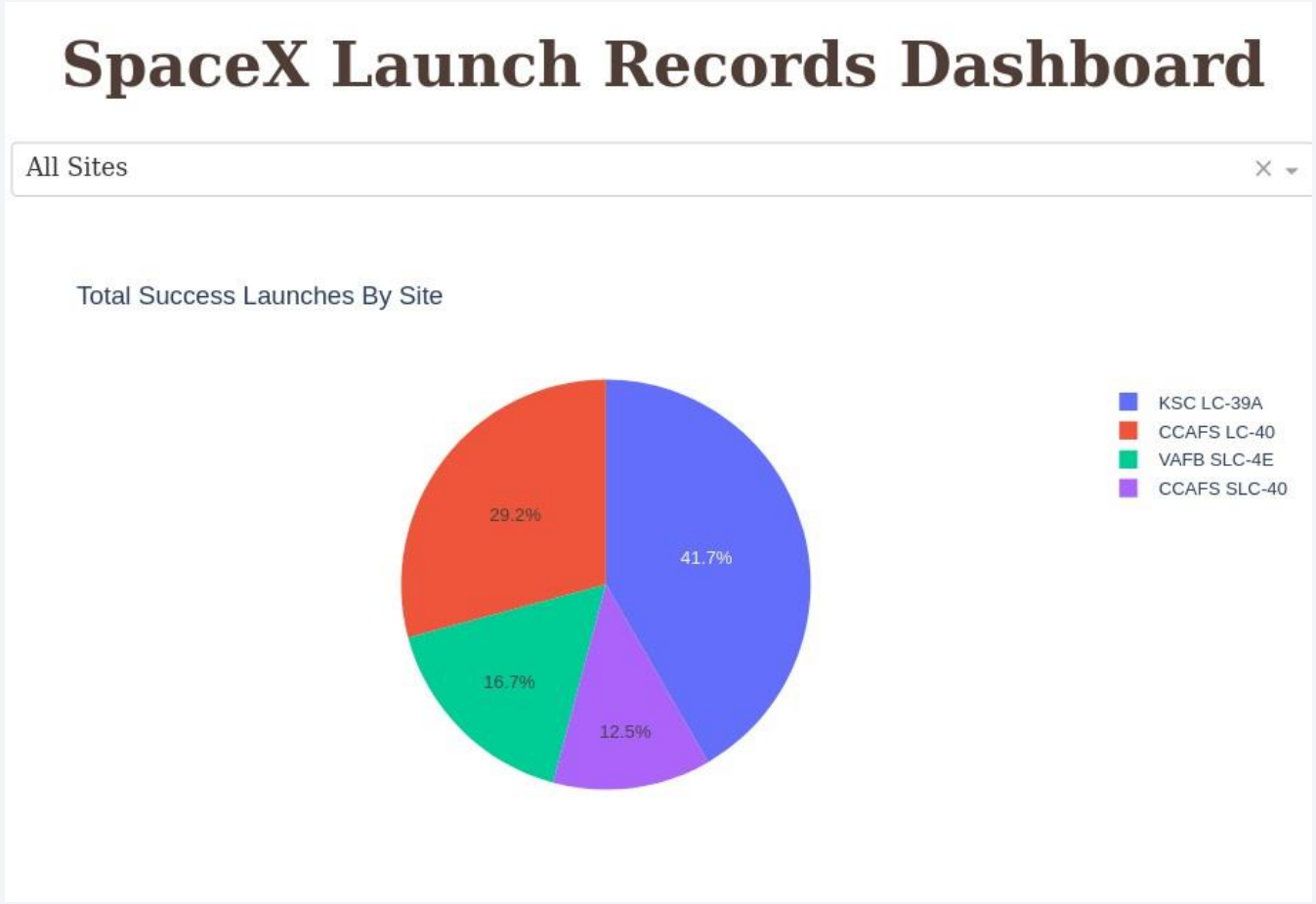
Logistics and Safety



Launch site KSC LC-39A has good logistics aspects, being near railroad and road and relatively far from inhabited areas.

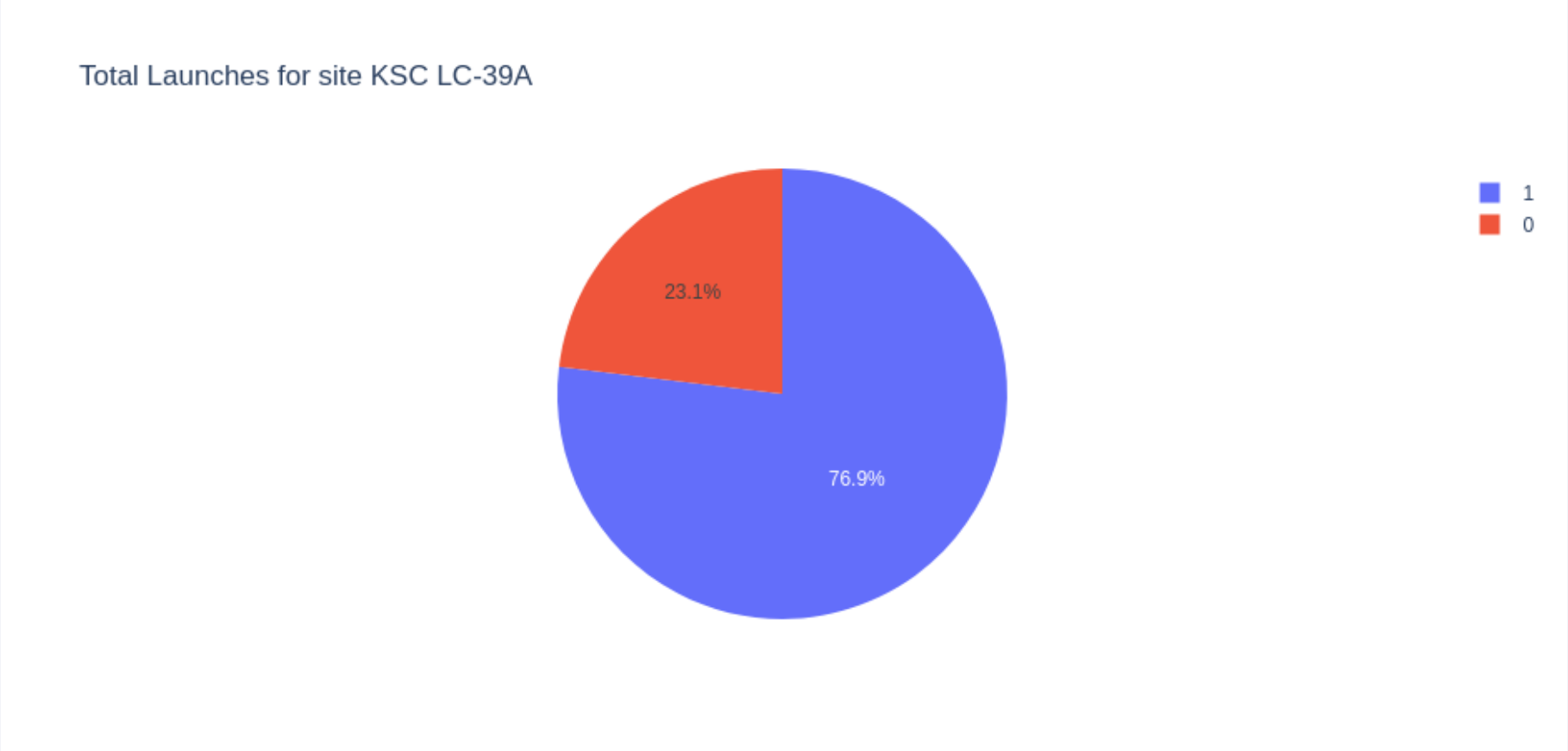

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by Site



The place from where launches are done seems to be a very important factor of success of missions.

Launch Success Ratio for KSC LC-39A

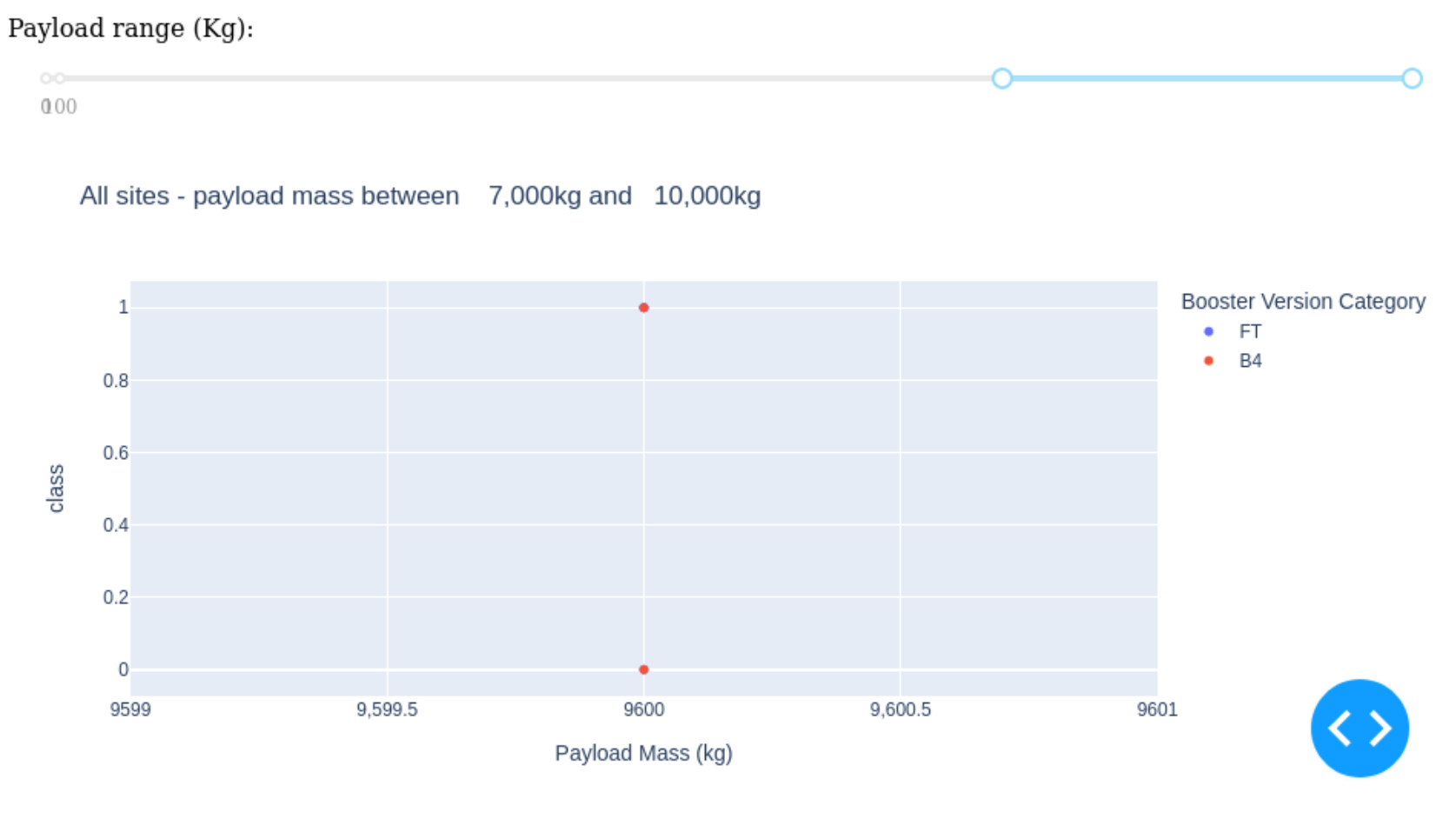


76.9% of launches are successful in this site.

Payload vs. Launch Outcome

Payloads under 6,000kg and FT boosters are the most successful combination.

Payload vs. Launch Outcome



There's not enough data to estimate risk of launches over 7,000kg

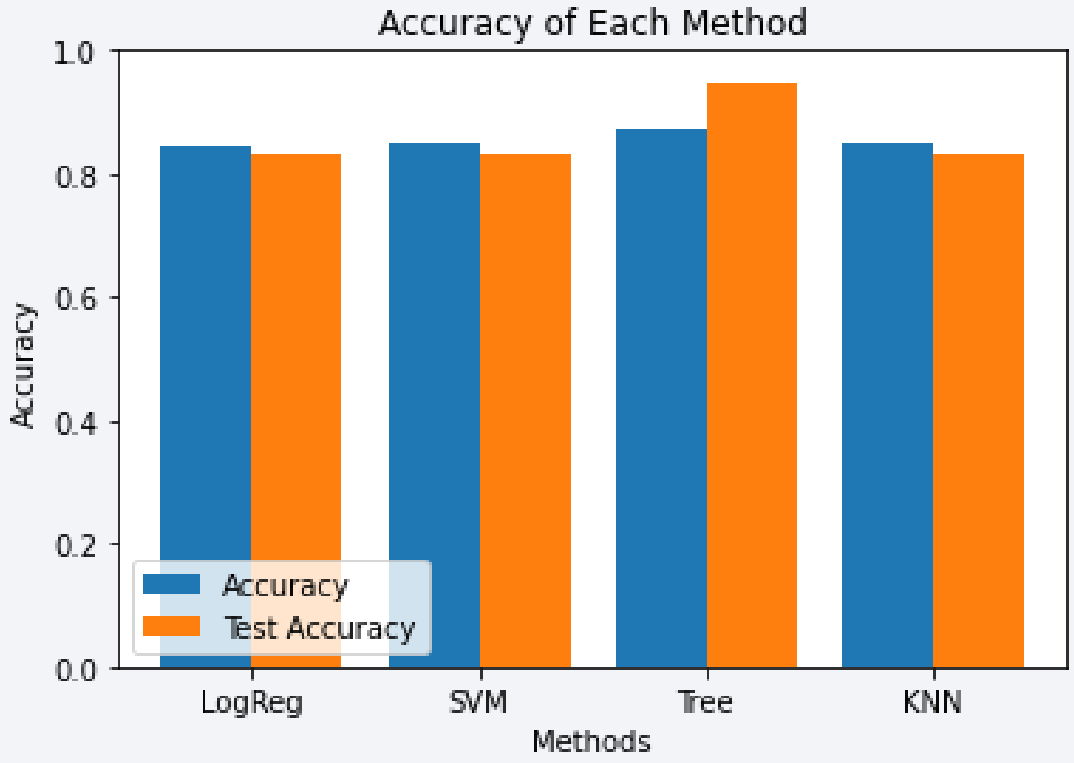
Section 5

Predictive Analysis (Classification)

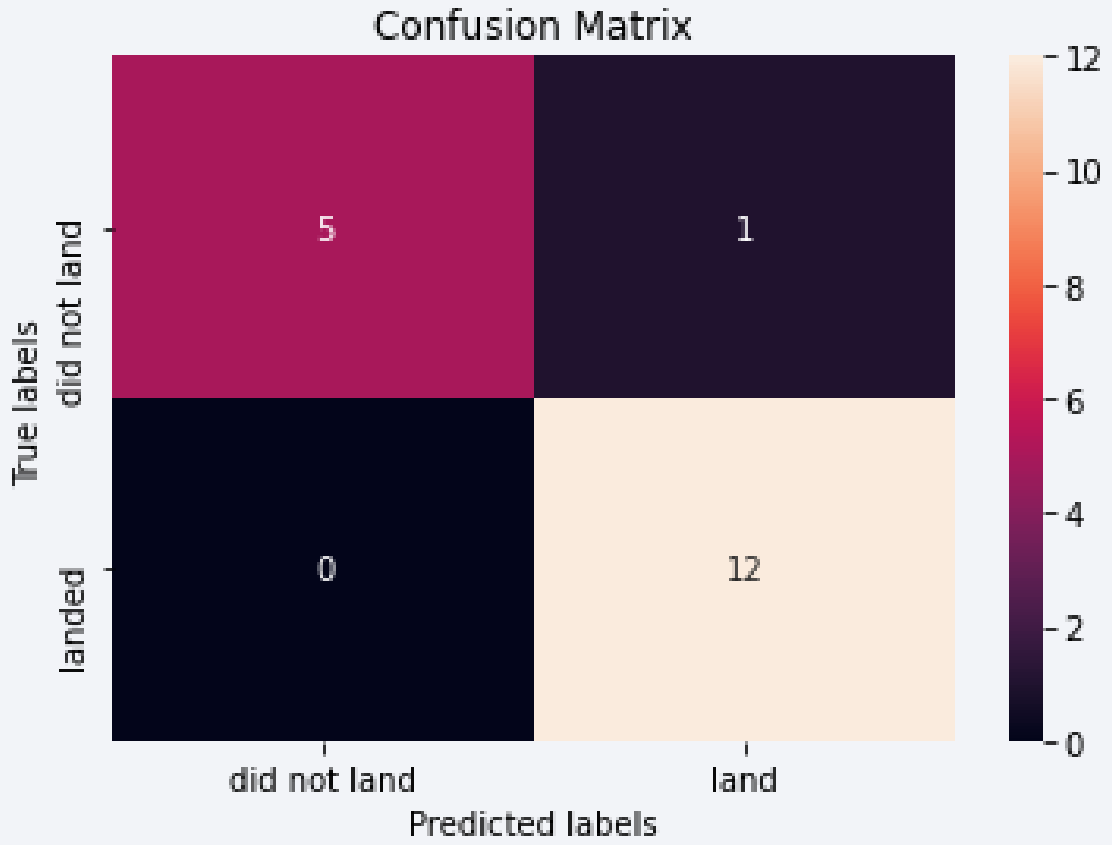
Classification Accuracy

Four classification models were tested, and their accuracies are plotted beside;

The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%.



Confusion Matrix



Confusion matrix of Decision Tree Classifier proves its accuracy by showing the big numbers of true positive and true negative compared to the false ones.

Conclusions

1. Different data sources were analyzed, refining conclusions along the process;
2. The best launch site is KSC LC-39A;
3. Launches above 7,000kg are less risky;
4. Although most of mission outcomes are successful, successful landing outcomes seem to improve over time, according the evolution of processes and rockets;
5. Decision Tree Classifier can be used to predict successful landings and increase profits.

Appendix

- As an improvement for model tests, it's important to set a value to np.random.seed variable;
- Folium didn't show maps on Github, so I took screenshots.

Thank you!