IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ismael Fernández Portillo>
<12/02/2025>

Outline

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

Executive Summary

- Summary of methodologies
  - We will obtain the data through web scraping and the SpaceX API.
  - Unnecessary columns will be removed and null data will be processed.
  - The data will be transformed and visualized in graphs.
  - Positions, as well as successes and failures, will be displayed on the map.
  - Different models will be trained to predict whether launches will be successful or not.
- Summary of all results
  - All locations are near the sea and far from densely populated areas.
  - We have found that the location with the highest success rate is the CAAFS SLC-40.
  - The best booster version is the FT with a payload between 2000 and 6000 kg.

Introduction

Project Background & Context

- The ability of the SpaceX Falcon 9 first stage to land successfully has become a critical aspect of modern space missions. Its reusability model represents a major shift in the economics of spaceflight, as each successful landing allows SpaceX to reuse boosters and reduce the costs traditionally associated with manufacturing new rocket stages. Beyond the financial benefits, understanding landing performance is essential for improving mission reliability, ensuring payloads reach orbit safely, and strengthening overall launch success rates.
- Each landing attempt also generates valuable technical data, enabling engineers to analyze performance, enhance rocket design, and implement improvements in guidance, propulsion, and control systems. By achieving consistent and safe landings, SpaceX strengthens its competitive position within the commercial space industry, offering more affordable launch services and pushing innovation in aerospace engineering. This progress not only benefits space exploration but also drives technological advancements with broader applications in other industries.

Problems We Want to Find Answers To

- What factors most strongly influence whether the Falcon 9 first stage lands successfully?
- How do landing outcomes affect overall mission reliability and operational performance?
- In what ways does successful booster recovery contribute to cost efficiency and long-term sustainability of launch operations?
- What insights can be extracted from landing data to improve rocket design, guidance systems, and engineering decisions?
- How does landing performance impact SpaceX’s competitive advantage in the commercial space industry?
- What technological innovations emerge from solving the challenges associated with rocket landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collect the data through the SpaceX REST API and Web Scrapping.
- Perform data wrangling
  - All Falcon 1 Booster Version were removed.
  - All Payload Mass missing values were filled with the mean.
- 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 Overview**

    1. The SpaceX REST API provides data about launches, including rocket details, payloads, and launch specifications.

    2. The main endpoint used for this project is api.spacexdata.com/v4/launches/past, which retrieves past launch data.You need to present your data collection process use key phrases and flowcharts

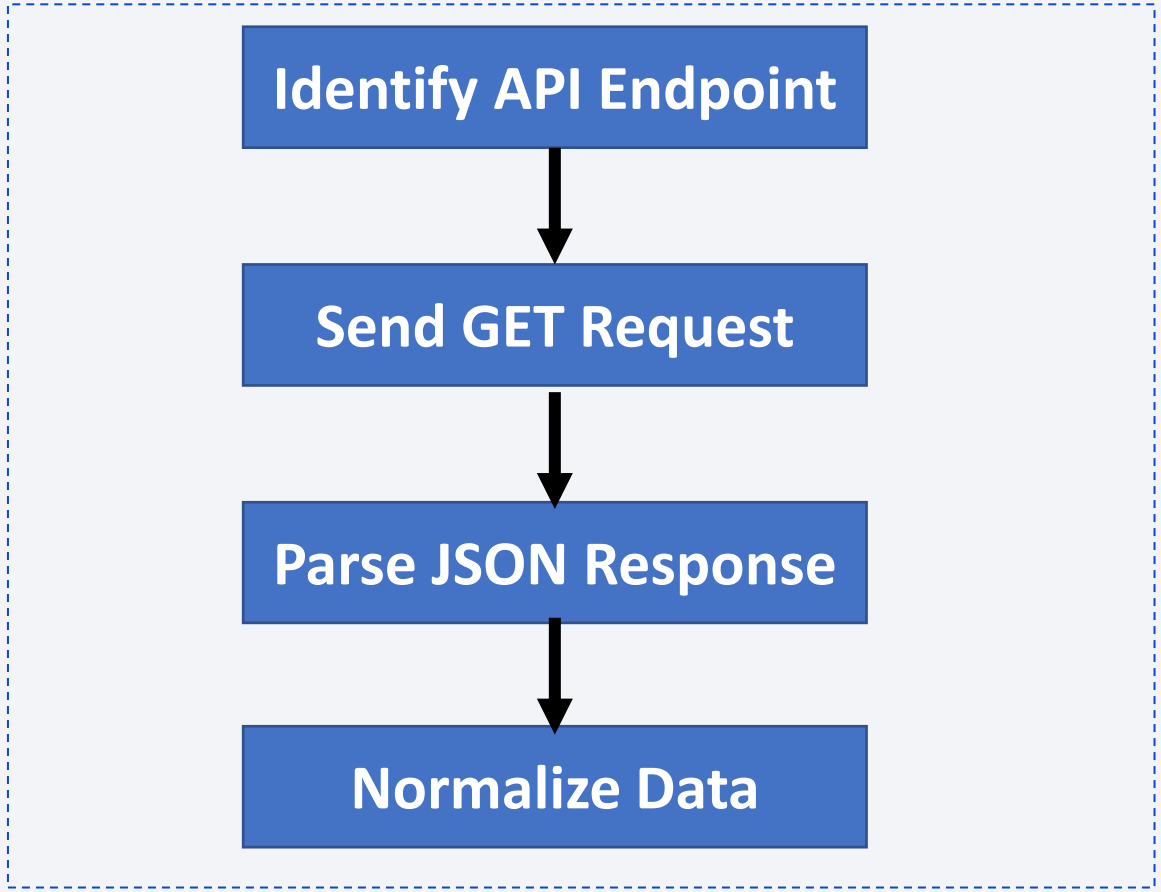
- **Data Retrieval Process**

    1. A GET request is performed using the requests library to obtain the launch data, which is returned in JSON format.

    2. The JSON response consists of a list of objects, each representing a launch, which can be converted into a dataframe using the json_normalize function.

Data Collection – SpaceX API

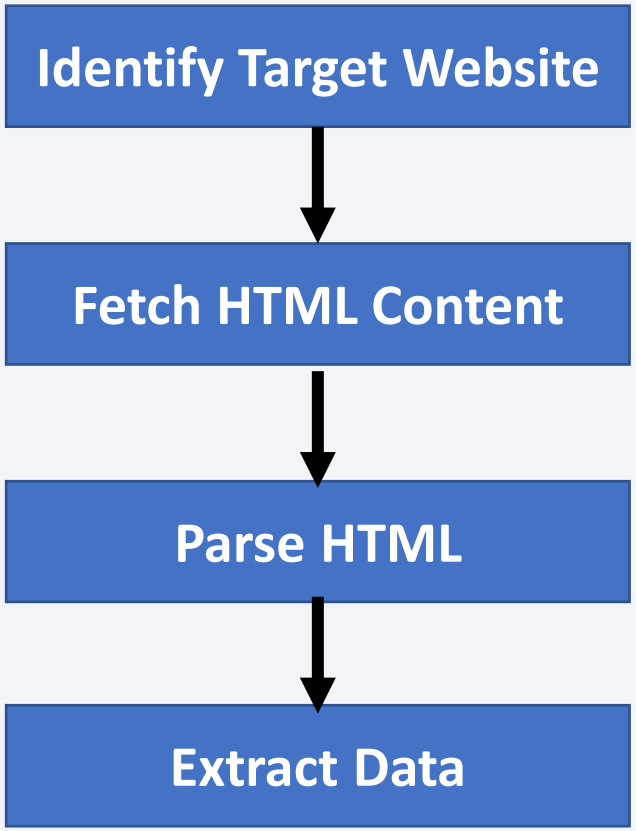
- Determine the specific API endpoint to access the desired data.
- Use a library like *request* in python to send a GET request o the API.
- Convert the response to JSON format
- Use *json_normalize* to convert the JSON data into a flat table (DataFrame)



https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

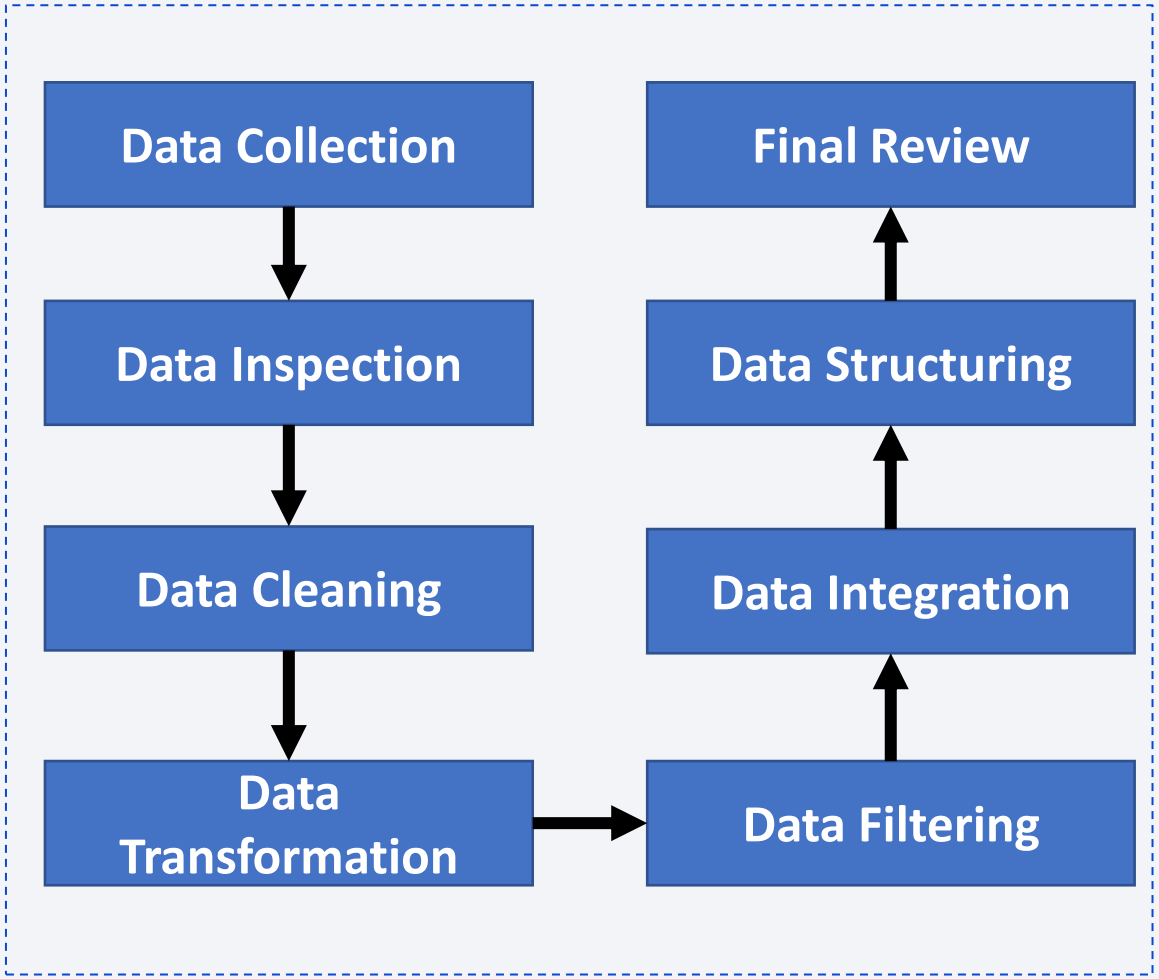
- Choose the website containing the relevant data
- Use *resquest* to get the HTML content of the page
- Use BeautifulSoup to parse the html content and locate the desired tables
- Loop through the tables to extract the relevant data and store it in a DataFrame.



https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Gather Data from various sources
- Review the structure and content of dataset. Identify data types and check for missing values
- Handle missing values and correct inconsistencies
- Normalize or standardize data formats, and convert categorical variables into numerical.
- Remove irrelevant or unnecessary data.
- Combine data into a single dataset.
- Reshape data into suitable format for analysis
- Conduct a final inspection.



https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Several catplots have been carried out to verify the relationship between the payload used, the number of flights, and the launch location on the success of the landings.
- A bar chart has been created to observe the success rate depending on the type of orbital launch performed
- Once it was observed that some orbital launches have a higher success rate than others, catplots were created to see the relationship between the success of a launch to a specific orbit with its payload and the number of flights.
- Finally, a line plot was created to observe how the success rate of pitches had changed over the years.

https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb

EDA with SQL

- %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
- %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
- %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;
- %sql SELECT DISTINCT(Landing_Outcome) FROM SPACEXTABLE;
- %sql SELECT * FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%' ORDER BY Date ASC LIMIT 1;
- %sql SELECT COUNT(*) FROM SPACEXTABLE;
- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000 AND Landing_Outcome LIKE 'Success (drone ship)';
- %%sql SELECT Landing_Outcome, COUNT(*) AS TOTAL from SPACEXTABLE GROUP BY Landing_Outcome;

https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

EDA with SQL

- %sql SELECT CASE WHEN Landing_Outcome LIKE 'Success%' THEN 'Success’ WHEN Landing_Outcome LIKE 'Failure%' THEN 'Failure’ WHEN Landing_Outcome LIKE 'No attempt' THEN 'Failure’ END AS Outcome_Type, COUNT(*) AS total FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%'OR Landing_Outcome LIKE 'Failure%’ OR Landing_Outcome LIKE 'No attempt%’ GROUP BY Outcome_Type;
- %sql SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;
- %sql SELECTsubstr(Date, 6, 2) AS month,Landing_Outcome,Booster_Version,Launch_SiteFROM SPACEXTABLEWHERE substr(Date, 1, 4) = '2015'AND Landing_Outcome LIKE 'Failure (drone ship)%'ORDER BY month;
- %sql SELECT Landing_Outcome, COUNT(*) AS total FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20’ GROUP BY Landing_Outcome ORDER BY total DESC;

https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- The three launch and landing sites have been marked with a circle.
- A pop-up window has been added to display the launch name.
- Rocket-style icons have been added to the launch site to indicate whether the landing was successful or unsuccessful.
- All these elements have been added to easily and visually locate all the test sites and to track the number of successes and failures.

https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

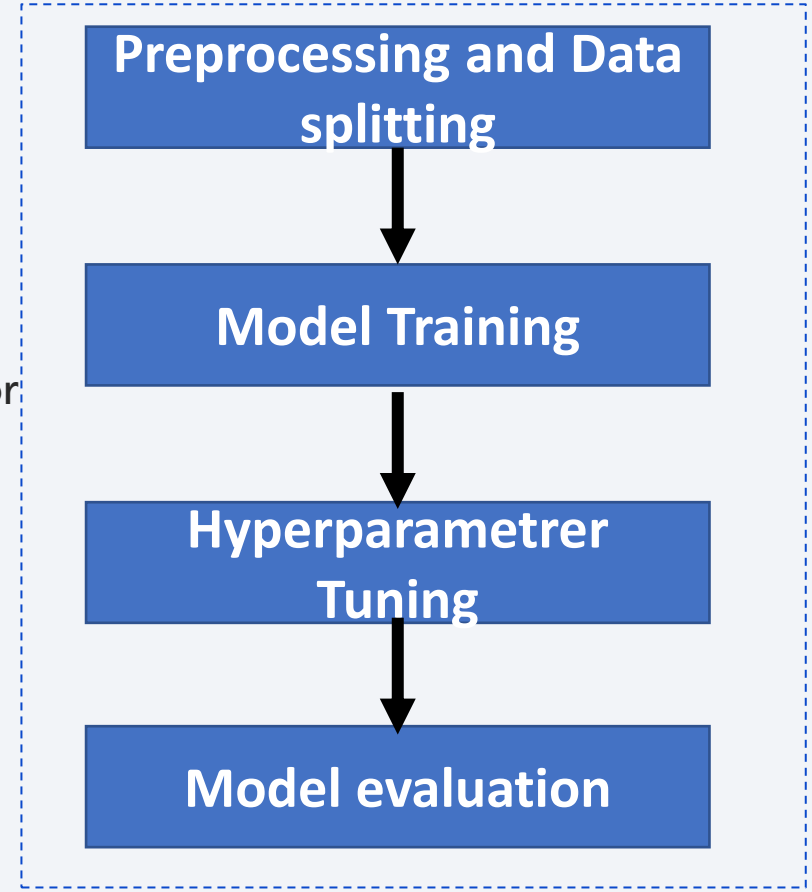
Build a Dashboard with Plotly Dash

- In this dashboard, we've introduced the ability to select the launch location and view the success and failure rates via a pie chart. We've also added a scatter chart where you can see the launch success and subsequent landing rates based on the booster version and payload mass. You can also choose the payload range you want to see on the scatter plot.
- All these interactions have been added based on the previously conducted data exploration. It has revealed the significant impact that the launch site, booster version, and playload have on the success of a proper launch and landing.

https://github.com/lfp93/Applied-Data-Science-Capstone/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

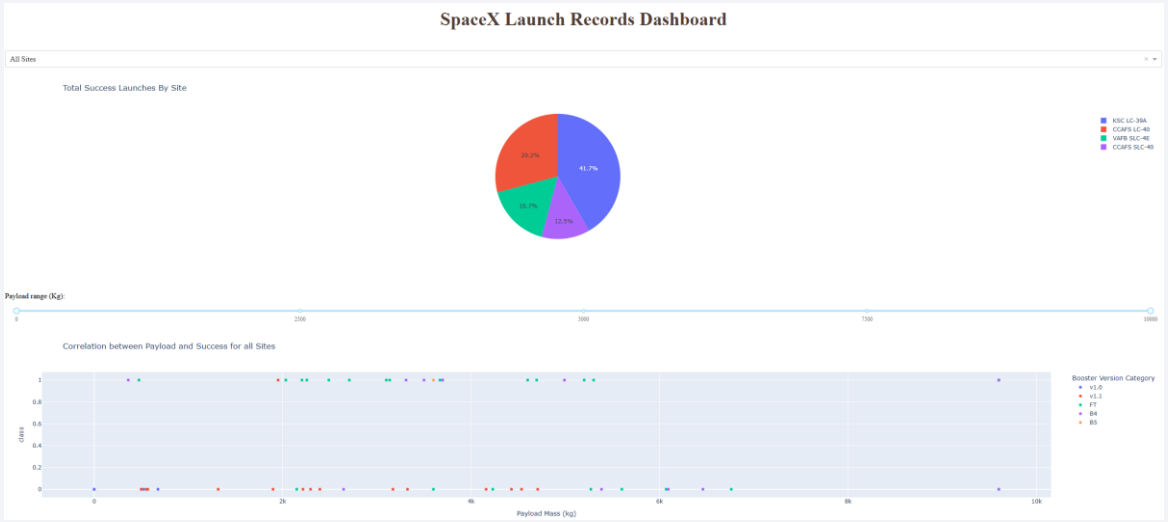
- Preprocessing and Data Splitting
  - Standardize data through preprocessing.
  - Split data into training and testing sets using Train_test_split.
- Model Training and Hyperparameter Tuning
  - Train various models including Logistic Regression and Support Vector Machines.
  - Perform Grid Search to identify optimal hyperparameters for model performance.
- Model Evaluation
  - Determine the model with the best accuracy using training data.
  - Output the confusion matrix to assess model performance.



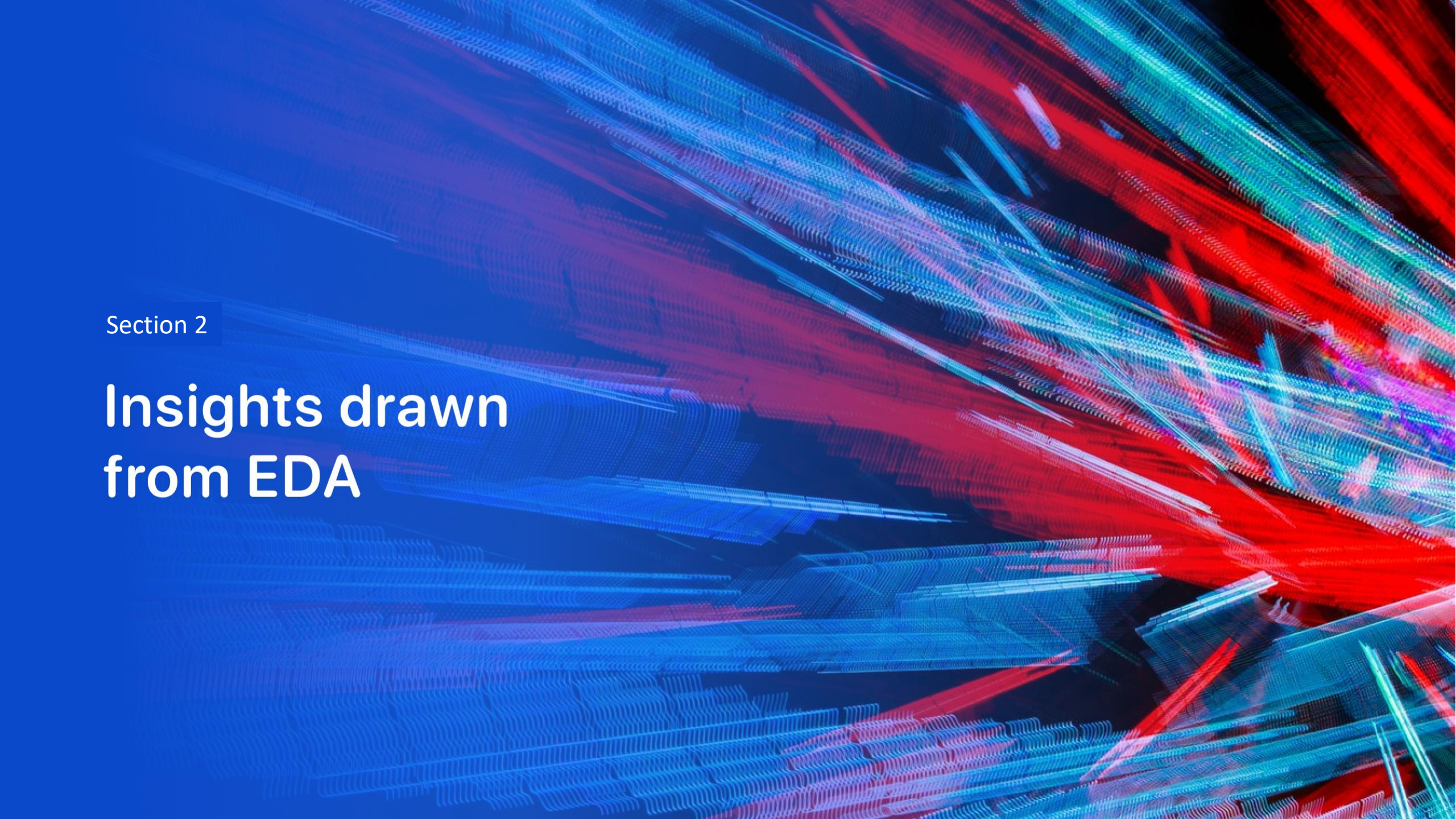
https://github.com/Ifp93/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- It has been observed that the type of orbit, the number of flights, the payload, and the launch site all have a significant impact on landing success.
- Interactive analytics demo in screenshots

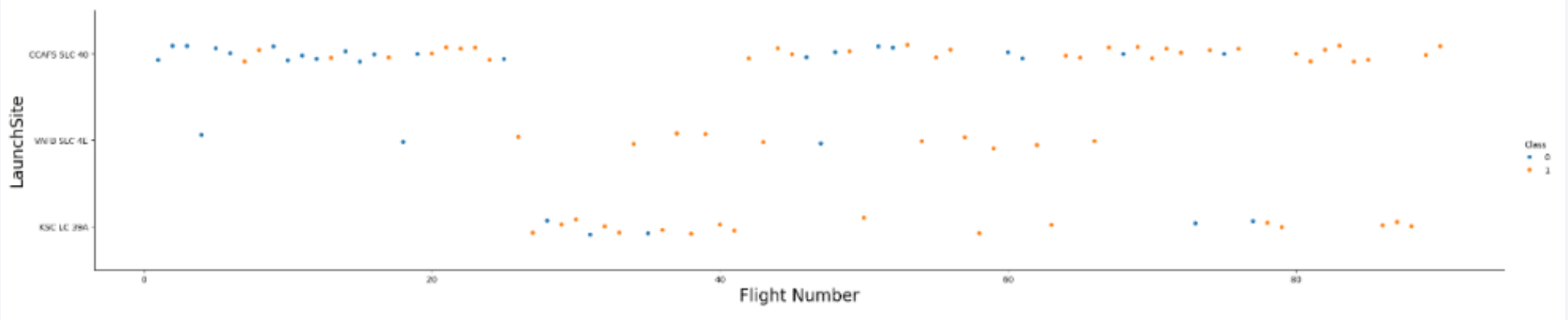


- Except for the decision tree classifier, all other methods have the same accuracy with both training and test data; only the decision tree classifier has greater accuracy with training data, showing that this model overfits.

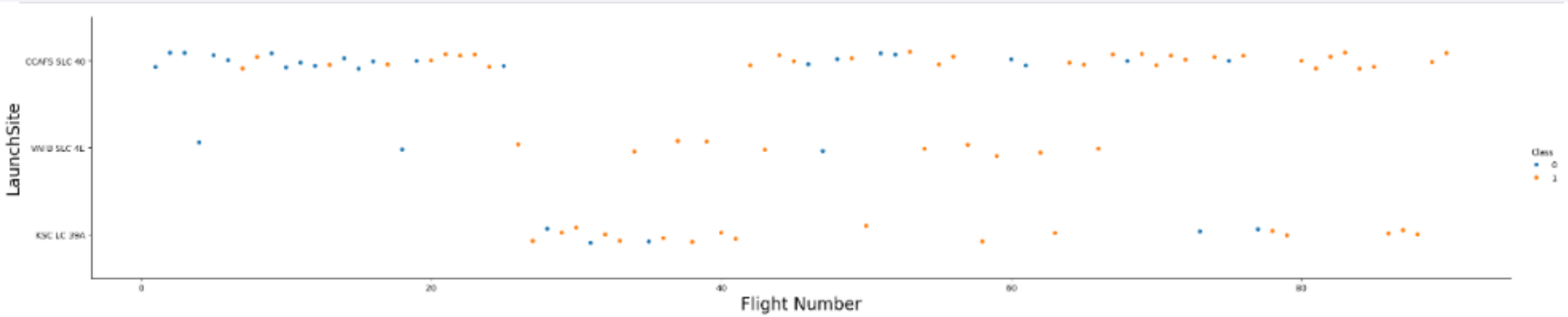

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site



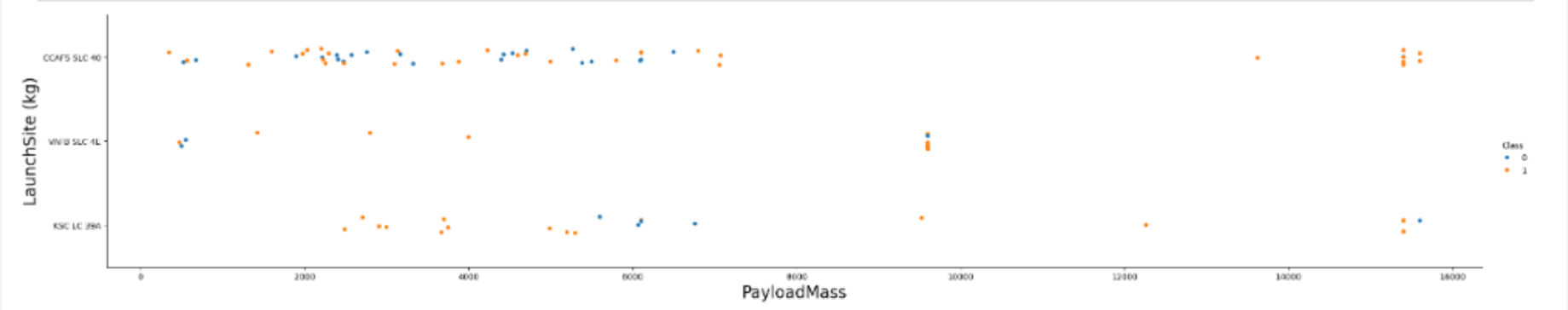
- Show the screenshot of the scatter plot with explanations



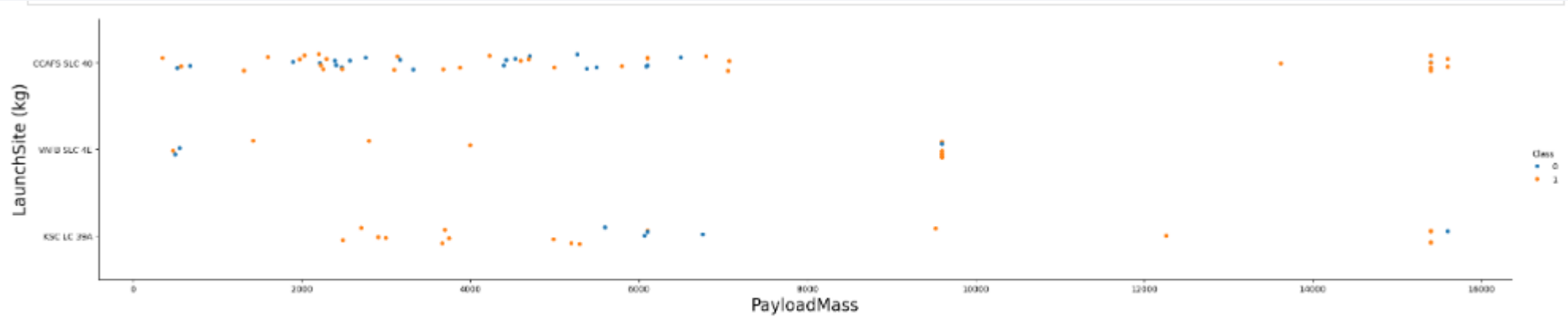
We can see how in CCAFS, as the number of flights increases, so does the probability of success. In VAFB, we see how, as with CAAFS, success increases drastically with the number of flights. However, in KSC, we can see how even with a large number of flights, there is still a high probability of failure.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



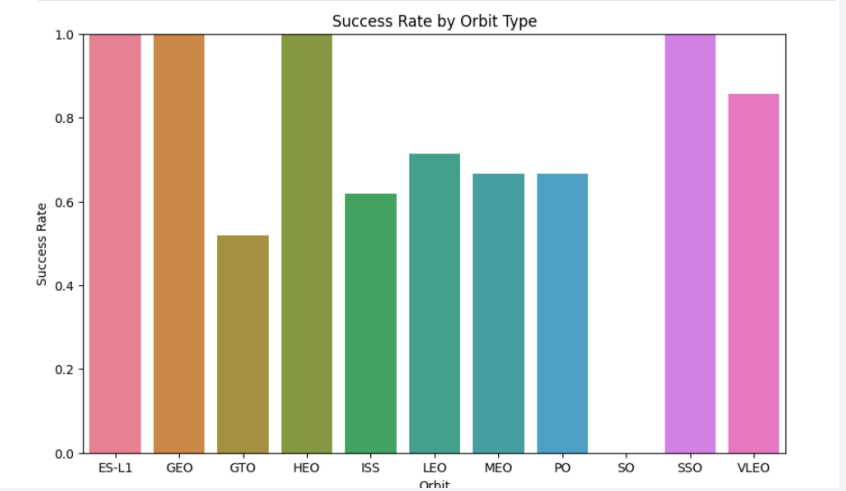
- Show the screenshot of the scatter plot with explanations



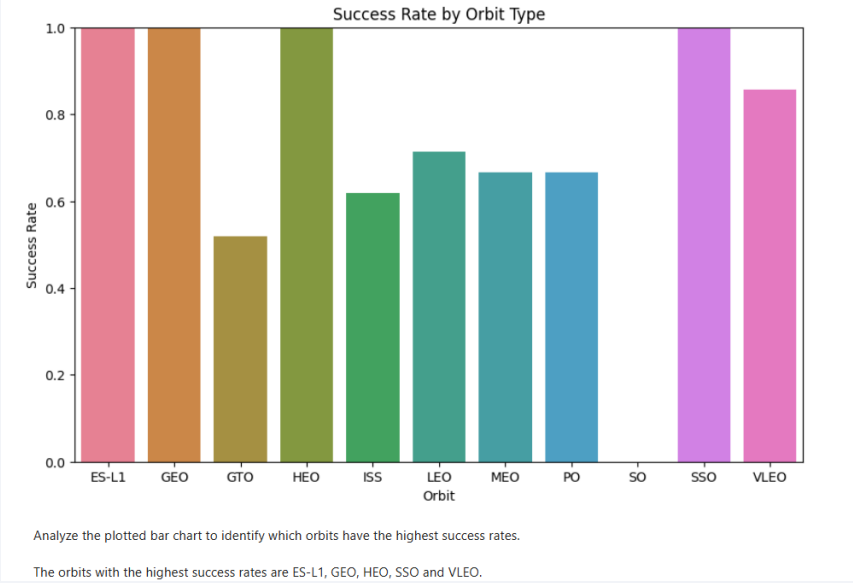
Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

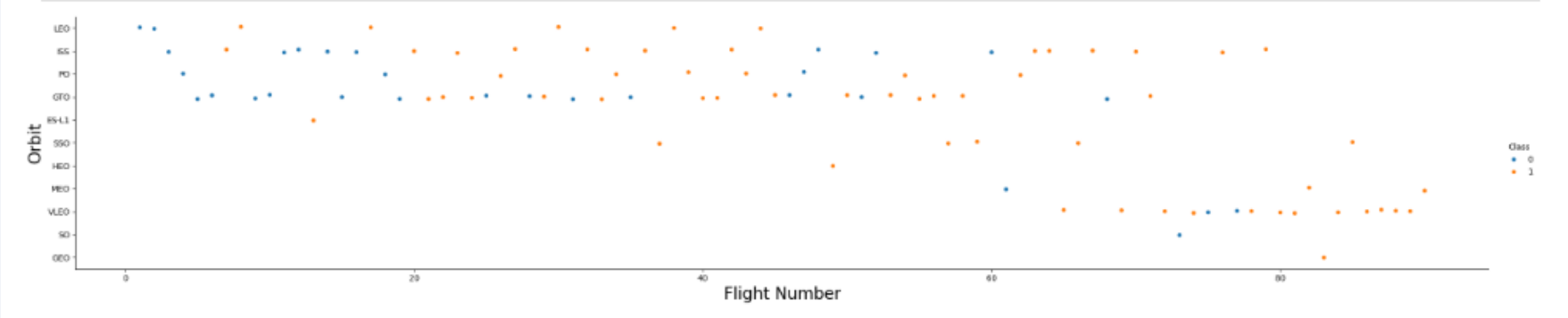


- Show the screenshot of the scatter plot with explanations

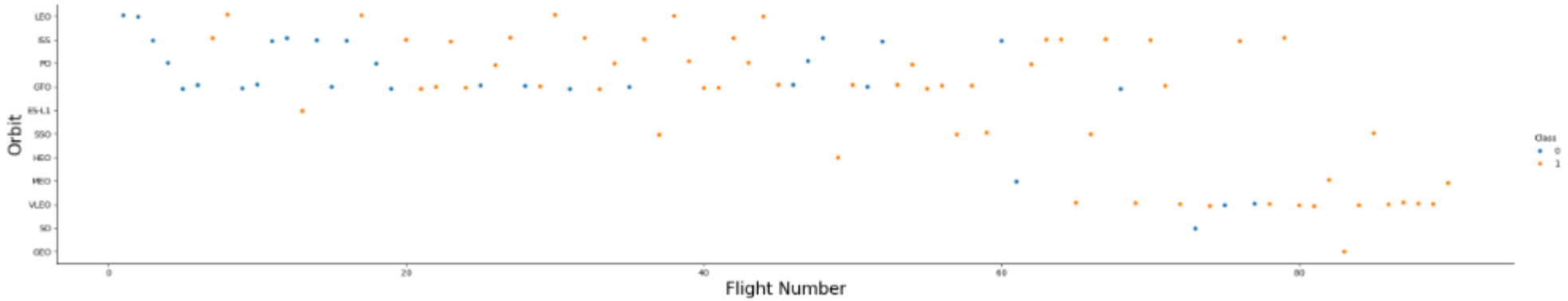


Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type



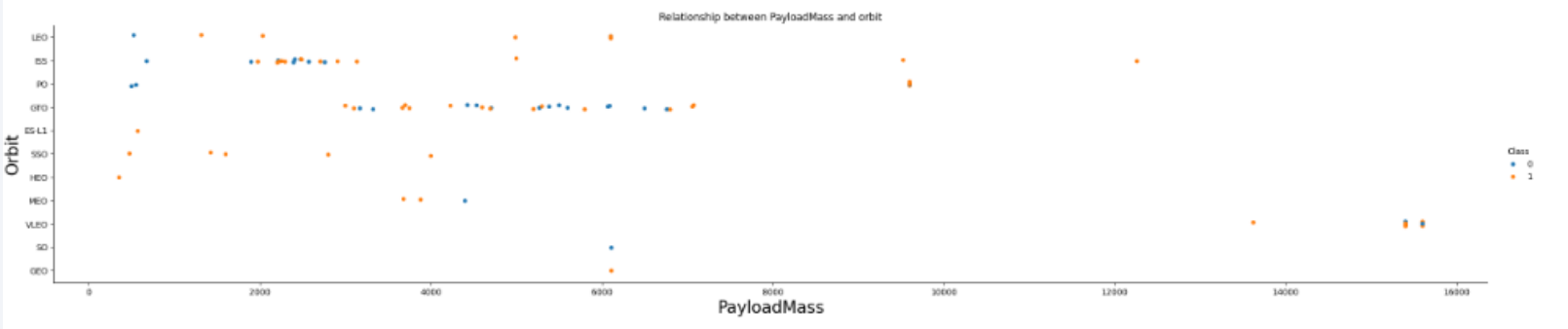
- Show the screenshot of the scatter plot with explanations



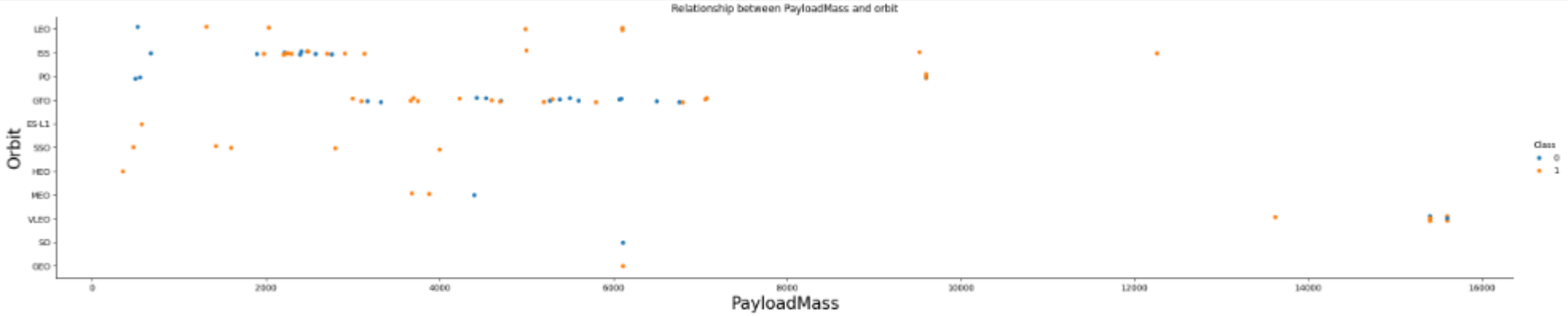
You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



- Show the screenshot of the scatter plot with explanations

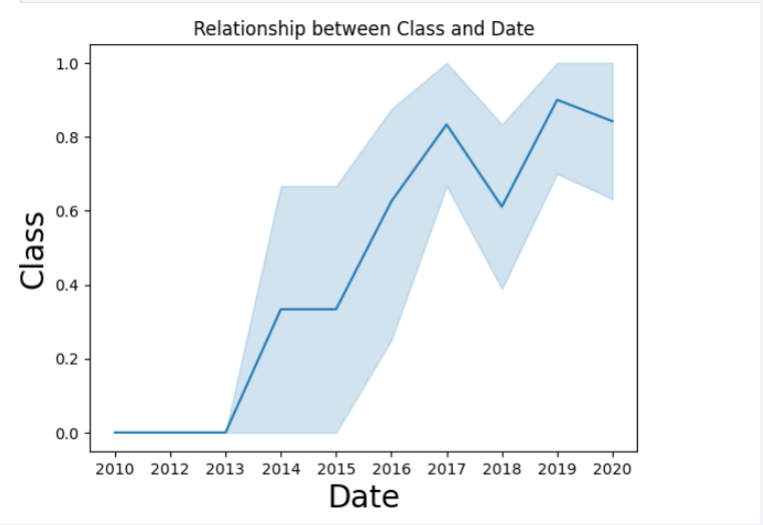


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

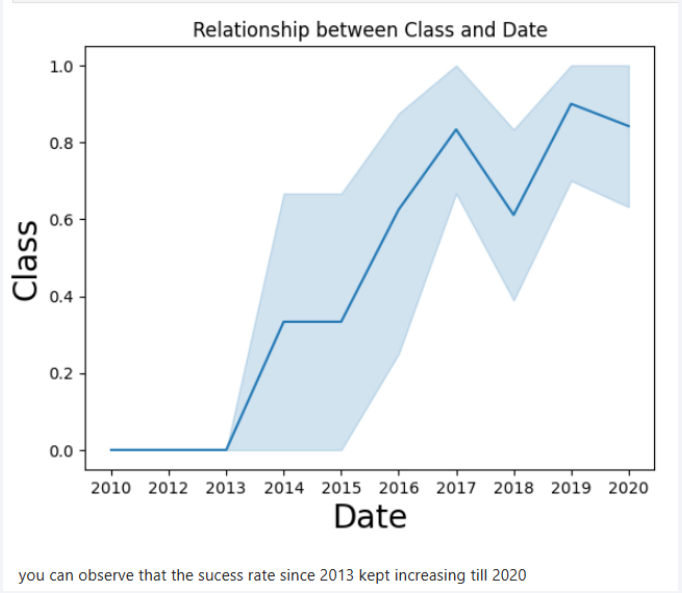
However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- Find the names of the unique launch sites

- Present your query result with a short explanation here

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

Using an SQL query, we asked it to tell us the different launch locations available.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

Using an SQL query, we requested the first 5 entries where the launch location begins 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

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Using an SQL query, we requested that it sum all the Pay Loads to find out the total.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;
```

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

SUM(PAYLOAD_MASS__KG_)
619967

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

Using an SQL query, we requested the average of Pay Loads for the F9 V1.1 version.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Using an SQL query, we asked it to sort the date in ascending order, from oldest to most recent, and to return the first one it found where the mission had been a success.

```
%sql SELECT * FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%' ORDER BY Date ASC LIMIT 1;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here

An SQL query was used to determine which booster names had successfully landed on a drone ship with a payload between 4000 and 6000.

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000 AND Landing_Outcome LIKE 'Succes
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Using an SQL query, all Landing Outcomes were selected, counted, and grouped by landing location to provide the sum of all of them according to each Landing Outcome.

```
%%sql SELECT Landing_Outcome, COUNT(*) AS TOTAL from SPACEXTABLE GROUP BY Landing_Outcome;
```

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

Landing_Outcome	TOTAL
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

An SQL query was used to ask which versions carried the maximum payload

```
%%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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here

Using an SQL query, the date 2015 and the drone ship-related failures were filtered out.

```
%%sql SELECT
        substr(Date, 6, 2) AS month,
        Landing_Outcome,
        Booster_Version,
        Launch_Site
    FROM SPACEXTABLE
    WHERE substr(Date, 1, 4) = '2015'
      AND Landing_Outcome LIKE 'Failure (drone ship)%'
    ORDER BY month;
```

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

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

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

- 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
- Present your query result with a short explanation here

```
%%sql SELECT
        Landing_Outcome,
        COUNT(*) AS total
    FROM SPACEXTABLE
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY total DESC;
```

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

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

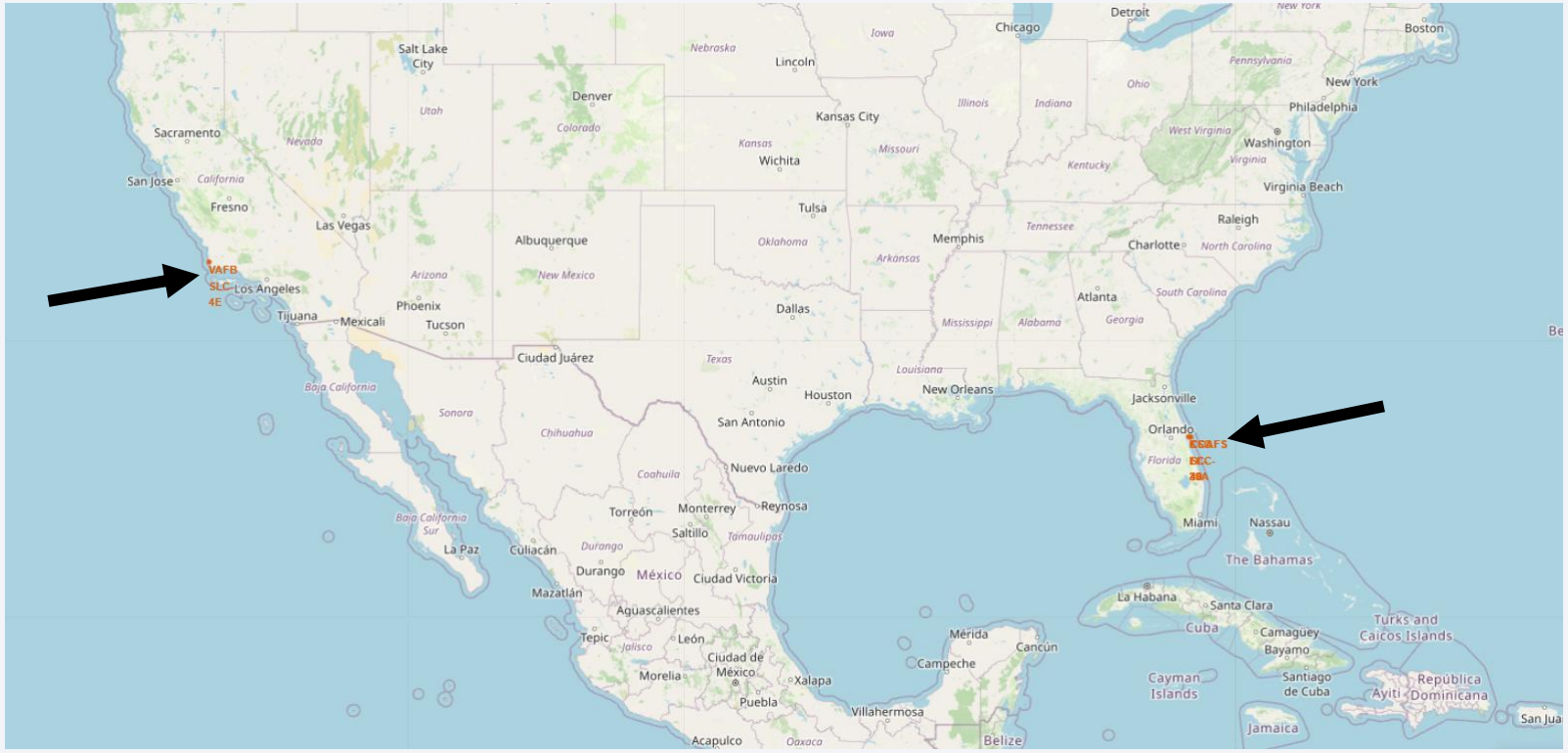
Using an SQL query, the requested dates were filtered, ordered by Landing Outcome, and sorted in descending order.

Section 3

Launch Sites Proximities Analysis

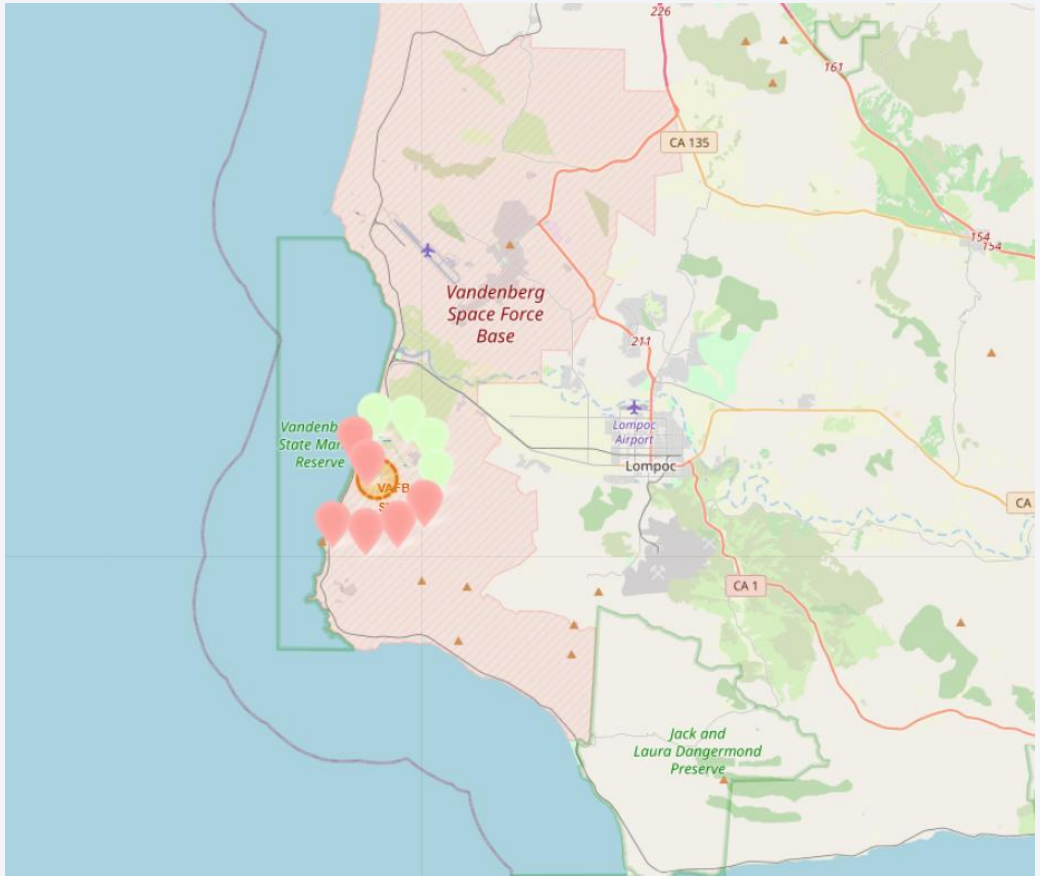
All launch sites in global map

Seeing the location of launch sites on a map is vitally important, because it provides us with relevant information; we can directly see how they are close to water and far from cities.



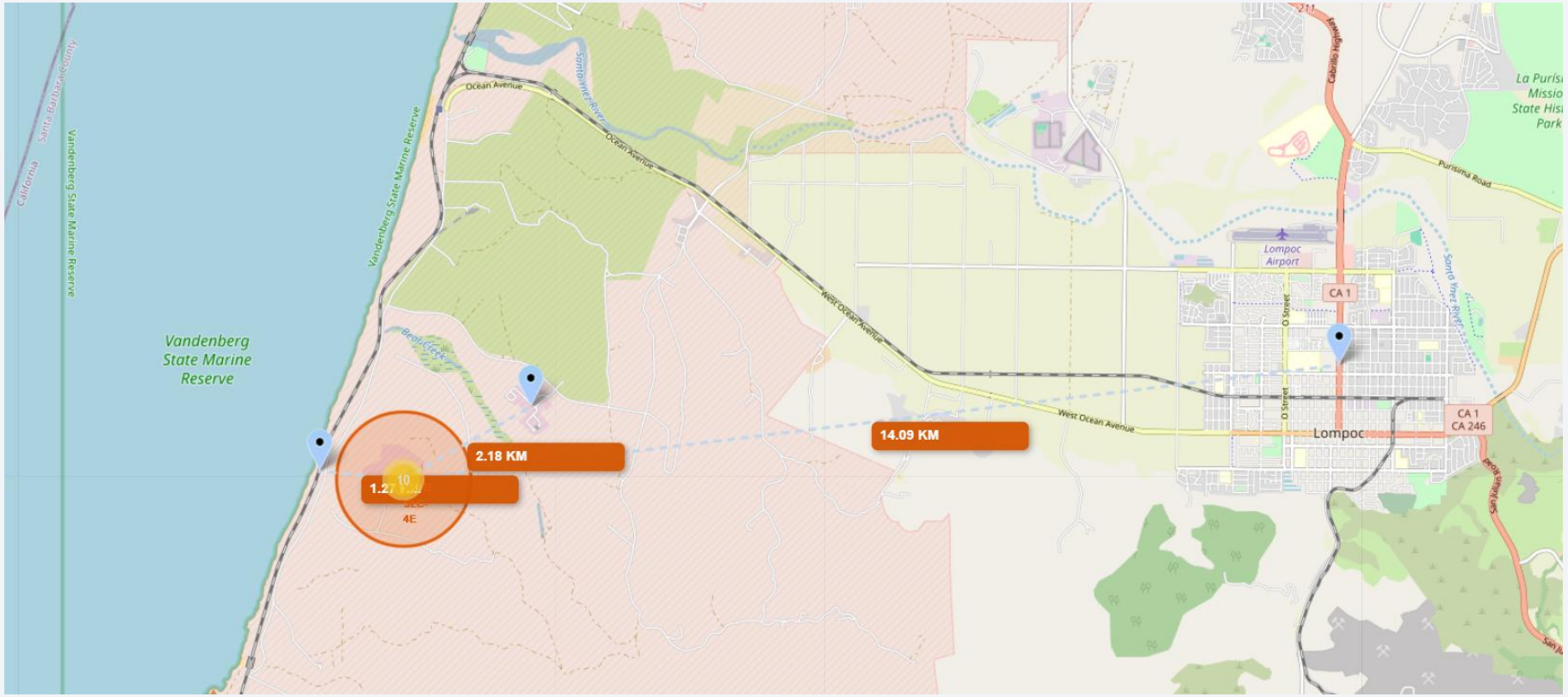
Launch Outcomes

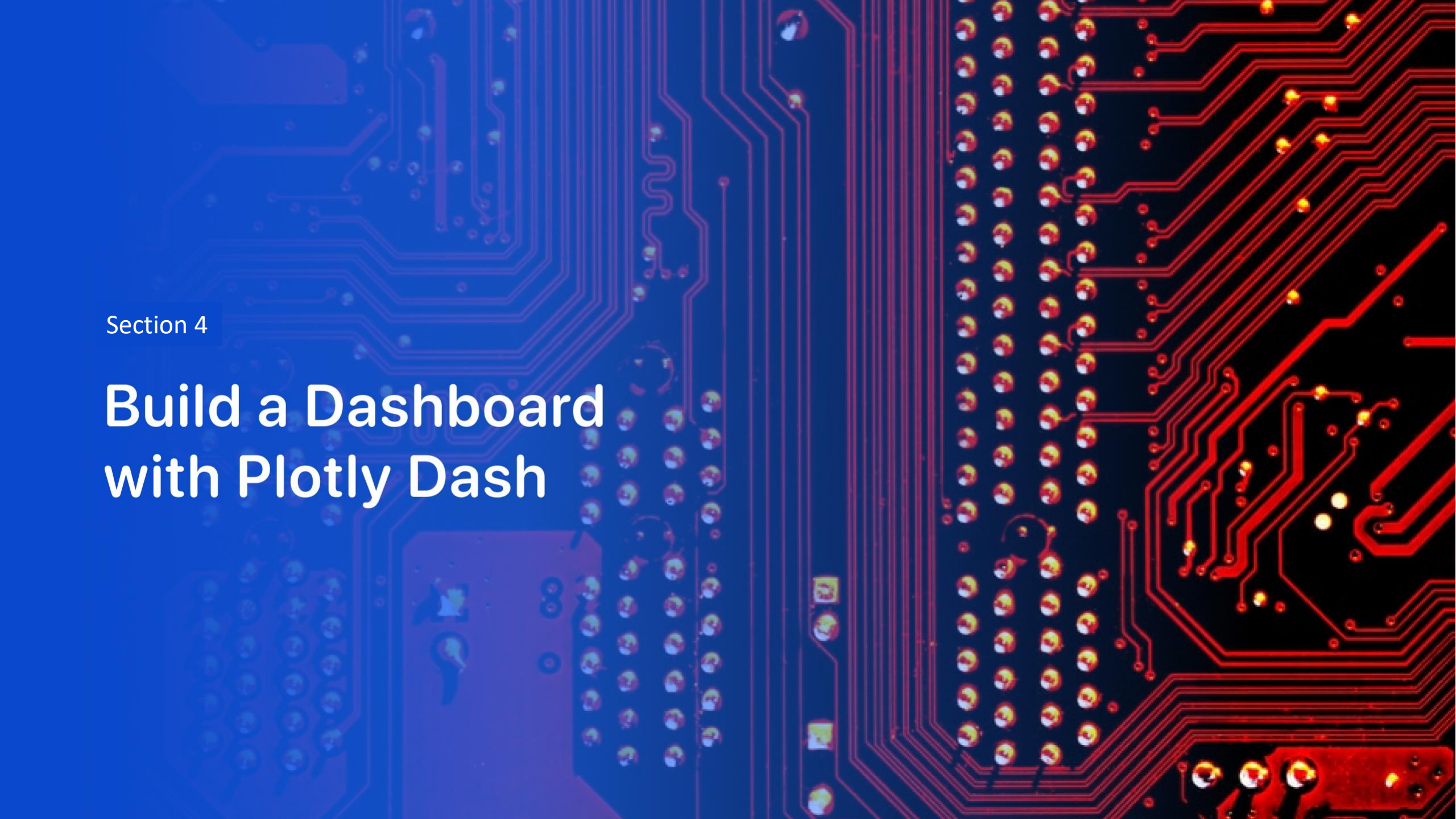
When we add the successes and failures of the throws to the zones, it quickly gives us a visual representation of their success rate and allows us to form a mental image of how profitable the throws are in these zones.



Proximities

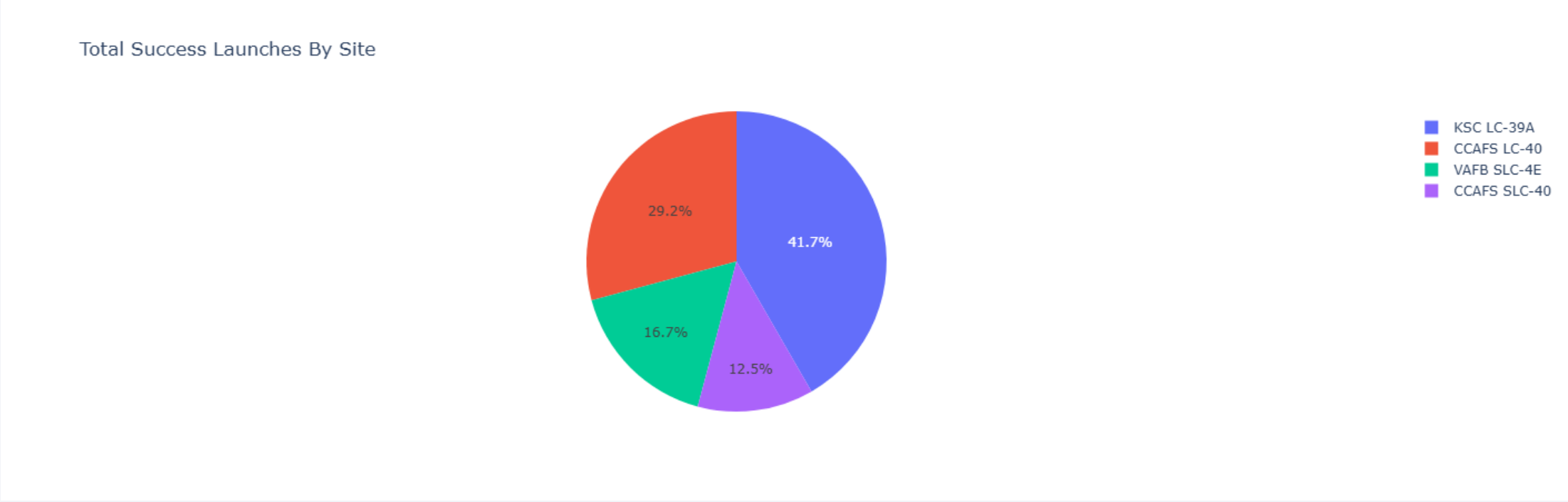
It is vitally important to observe our surroundings. By positioning the launch site on the map and calculating the distance between risk areas where people are present, we can get an idea of the safety perimeter that the tests will have.




Section 4
Build a Dashboard with Plotly Dash

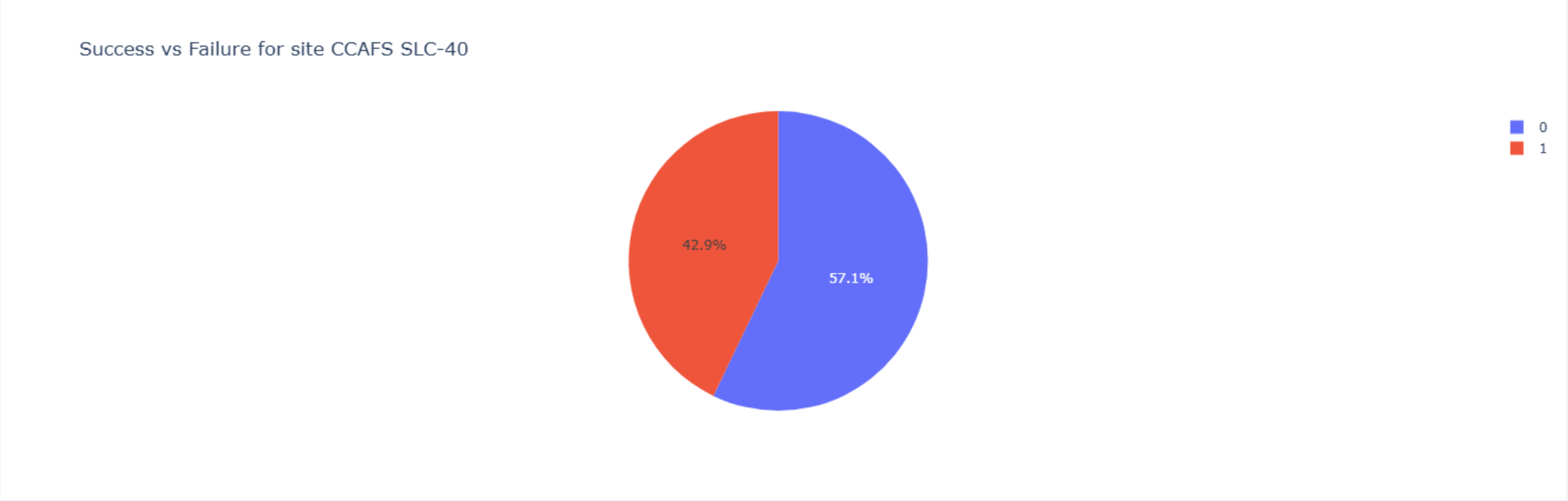
Total Success Launches By Site

This first image shows us how global successes are distributed among all launch positions, allowing us to see where there are a greater number of successes, although we must be cautious with this, since their success rate may be low and it may simply be that many launches are made in that place



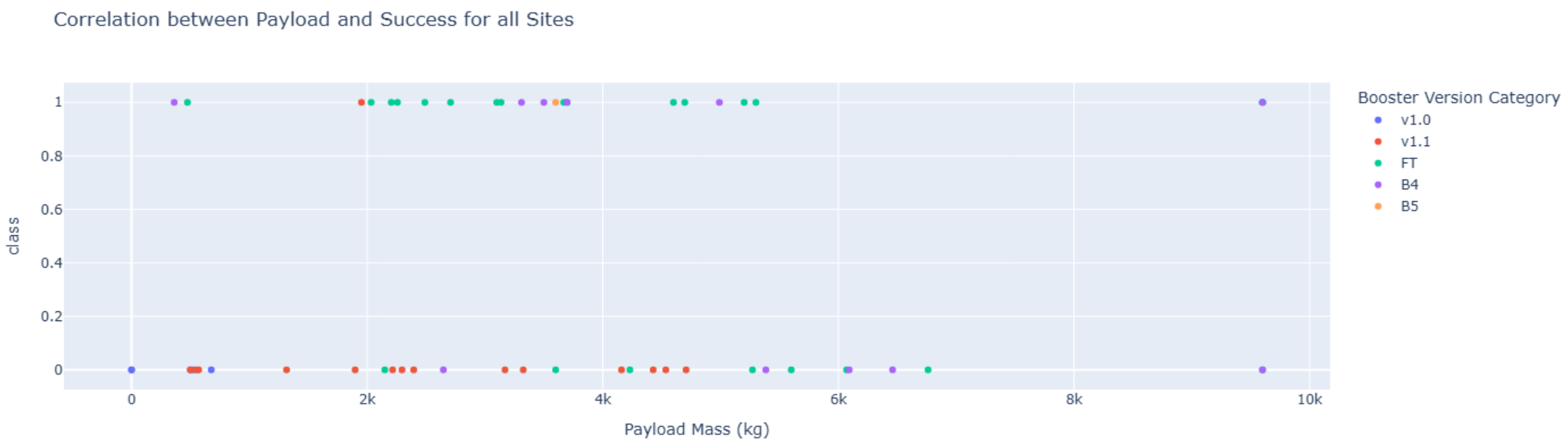
Highest launch success ratio

- Thanks to this image, we can quickly see which location has the highest landing success rate. With this information, we can begin to ask ourselves why this location has the best rate.



Payload vs Launch Outcome

This chart is very instructive, as we can see the successes of the different rocket versions, and it is ordered according to their payload. We can observe that between 2,000 and 6,000 kg, the FT is a clear winner.

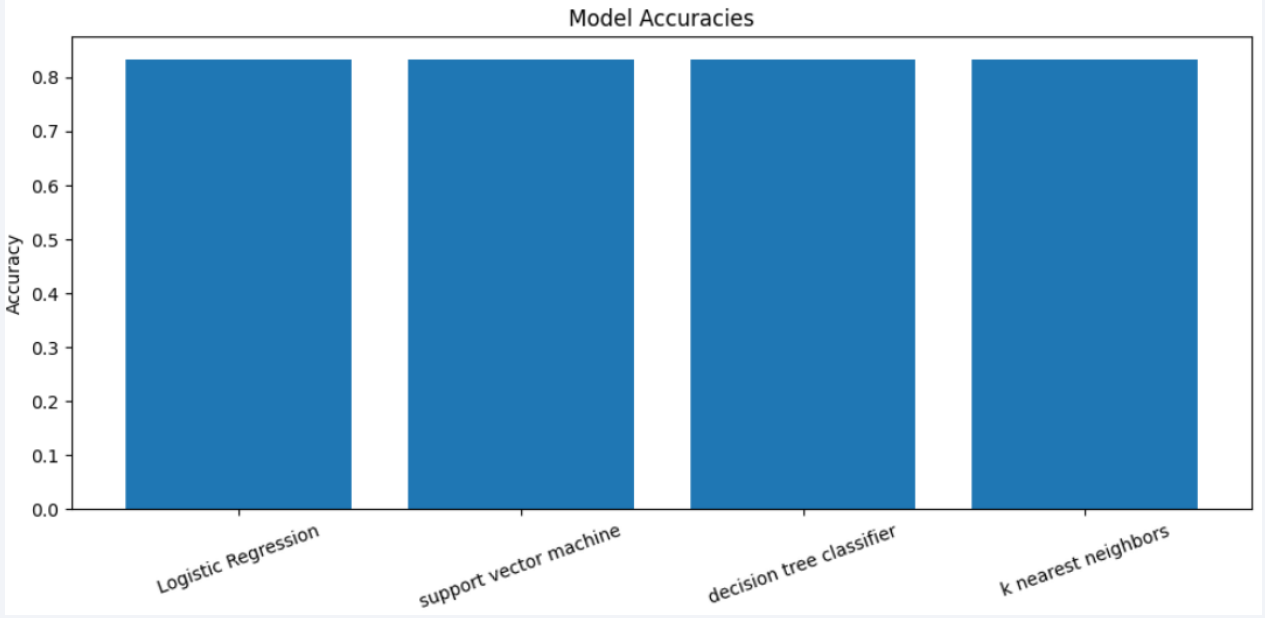


Section 5

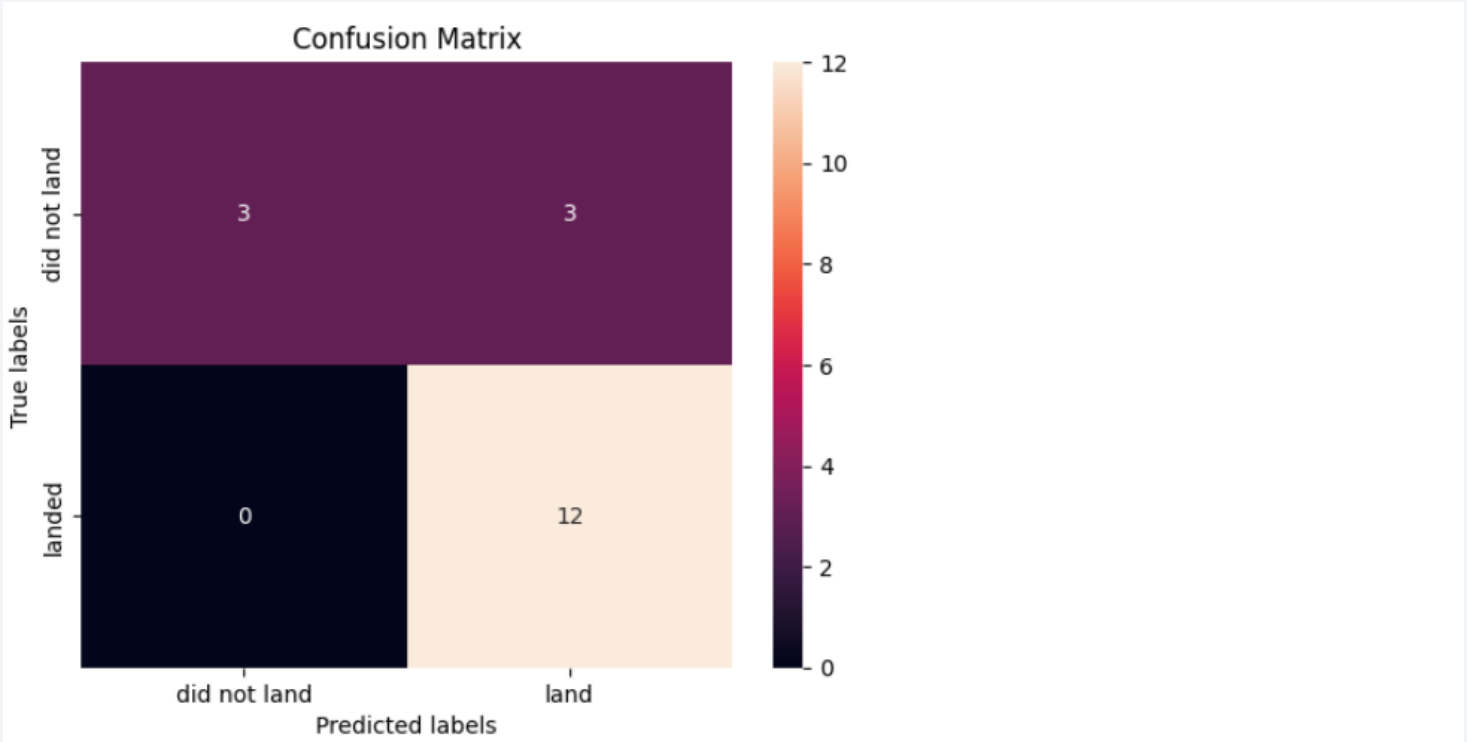
Predictive Analysis (Classification)

Classification Accuracy

Except for the decision tree classifier, all other methods have the same accuracy with both training and test data; only the decision tree classifier has greater accuracy with training data, showing that this model overfits.



Confusion Matrix



Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the problem is false positives.

Overview:

True Postive - 12 (True label is landed, Predicted label is also landed)

False Postive - 3 (True label is not landed, Predicted label is landed)

Conclusions

- It is vitally important that launch sites are located away from any place that could pose a risk to people or infrastructure.
- The best model is the FT with a payload between 2000 and 6000kg; it has the best success rate.
- The best launch site, with the best ratio, is the CCAFS SLC-40
- All the prediction models that have been trained have performed similarly
- It can be seen how the technology has been refined over the years, making it possible for the success rate to rise to 80%, which positions the company far ahead of its competitors in saving money thanks to the reuse of the first phase of the launch

Thank you!