IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kevin Vilca
04/03/2024

Outline

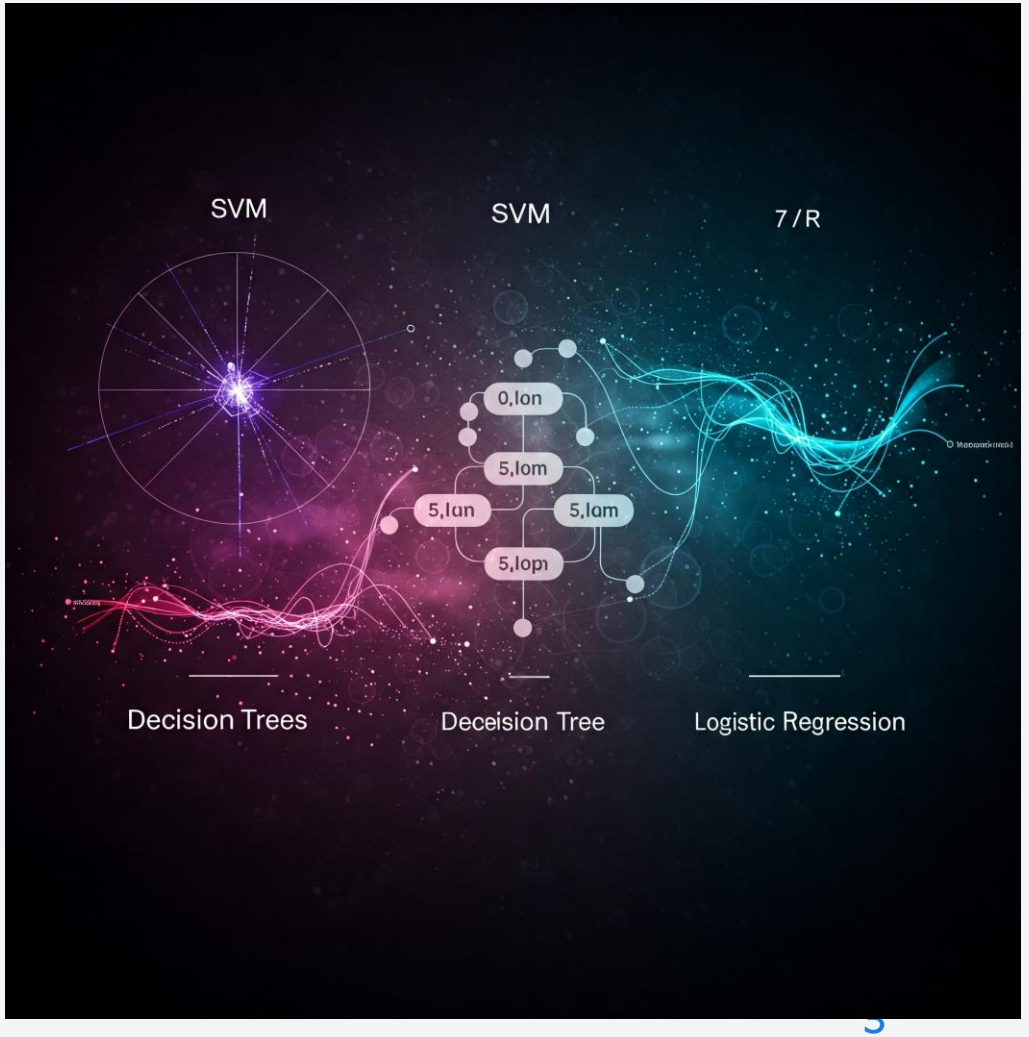
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

The goal of this project is to predict whether the Falcon 9 first stage will land successfully, which is critical to optimize SpaceX's launch costs.

Through data collection using a RESTful API and web scraping, exploratory analysis was performed and an interactive dashboard was built to visualize the data. Subsequently, machine learning techniques were applied to predict landing success.

Classification models such as SVM, Decision Trees and Logistic Regression were trained, with an observed optimal performance of 83.33%



Introduction

SpaceX has been able to reduce launch costs by reusing the first stage of its Falcon 9 rockets. This project aims to predict whether the first stage will land successfully, which directly impacts the cost and feasibility of future missions. Through the collection of data from previous landings and analysis of related variables, it will seek to develop a predictive model that will allow SpaceX to make informed decisions.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

The dataset was gathered through multiple methods to ensure comprehensive coverage of Falcon 9 launch records:

- We retrieved data using GET requests from the SpaceX API, obtaining detailed information about past launches.
- The API response, formatted as JSON, was decoded using the .json() function and then transformed into a Pandas DataFrame using .json_normalize() for easier analysis.
- The dataset was then cleaned by identifying and handling missing values, ensuring data consistency.
- Additionally, web scraping was performed using BeautifulSoup to extract Falcon 9 launch records from Wikipedia.
- The extracted data, originally structured as an HTML table, was parsed and converted into a Pandas DataFrame for further analysis.

This multi-source data collection approach allowed for a more complete and structured dataset, which was essential for subsequent analysis and modeling.

Data Collection – SpaceX API

- We collected launch data from the SpaceX API using a structured approach involving RESTful calls. Below is an overview of the key steps taken
- Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

```
response = requests.get(spacex_url)
```

Check the content of the response

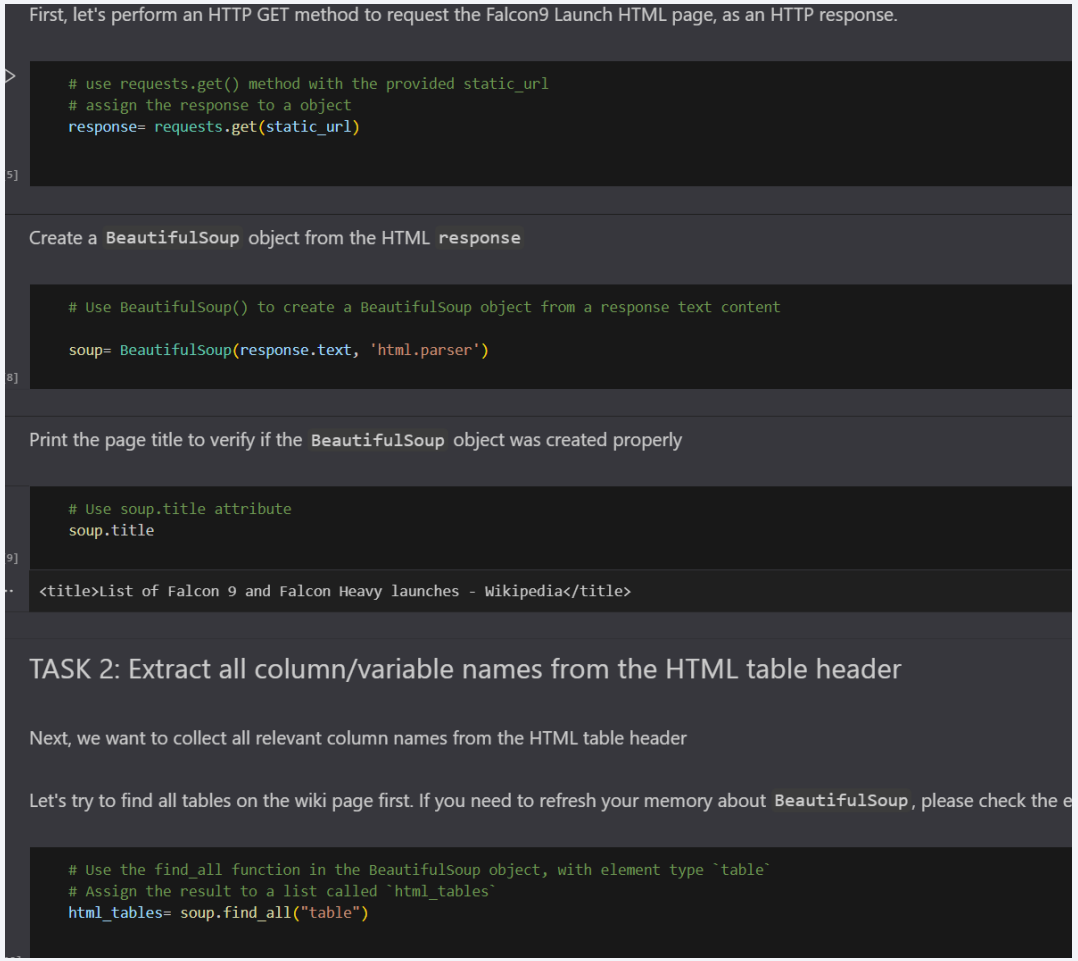
```
print(response.content)
```

Data Collection - Scraping

Key Phrases

- Make a GET request
- Create a beautiful soup
- Find all the tables
- loop through the third table
- In the table, find all of the rows
- check the table heading
- Save all cells in a dictionary
- create a data frame from the dictionary

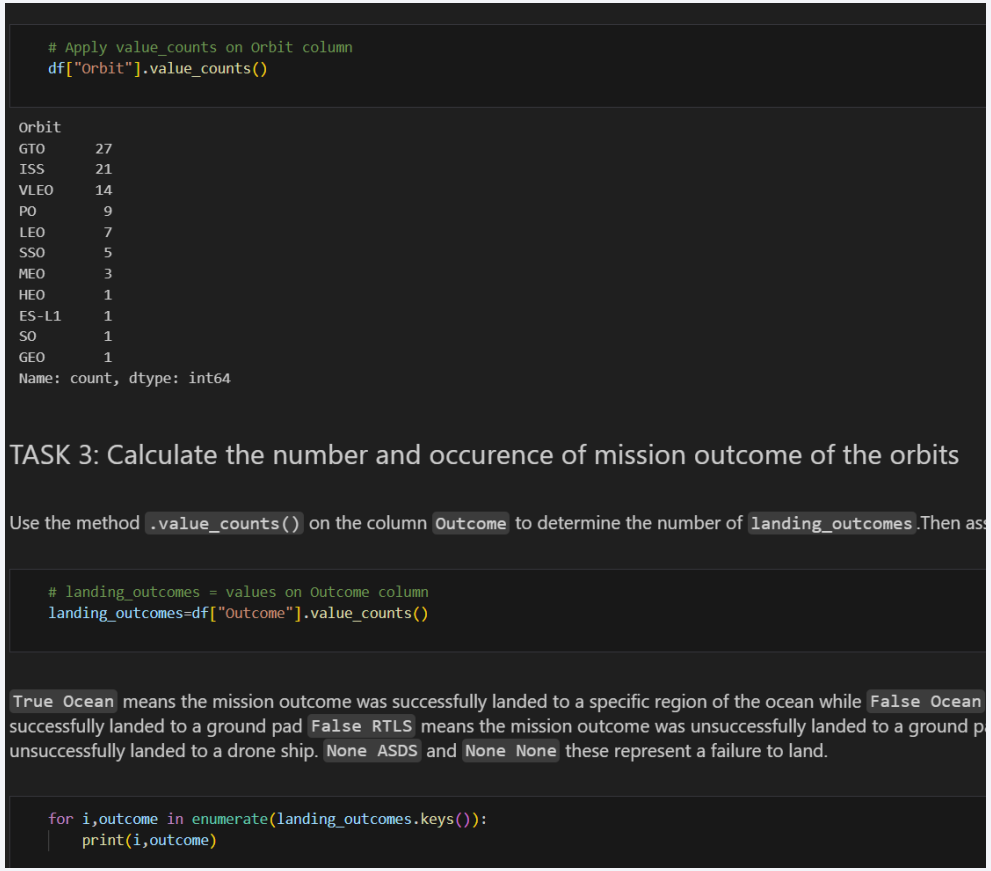
- Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



Data Wrangling

Key phrases:

- Identify and calculate the percentage of missing values in each attribute.
- Identify which columns are numeric and categorical.
- Calculate the number of launches at each site.
- Calculate the number and occurrence of each orbit.
- Calculate the number and occurrence of orbits mission outcome.
- Create a landing result label from the Result column.
- Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



```
# Apply value_counts on Orbit column
df["Orbit"].value_counts()
```

```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome of the orbits

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes`.Then ass

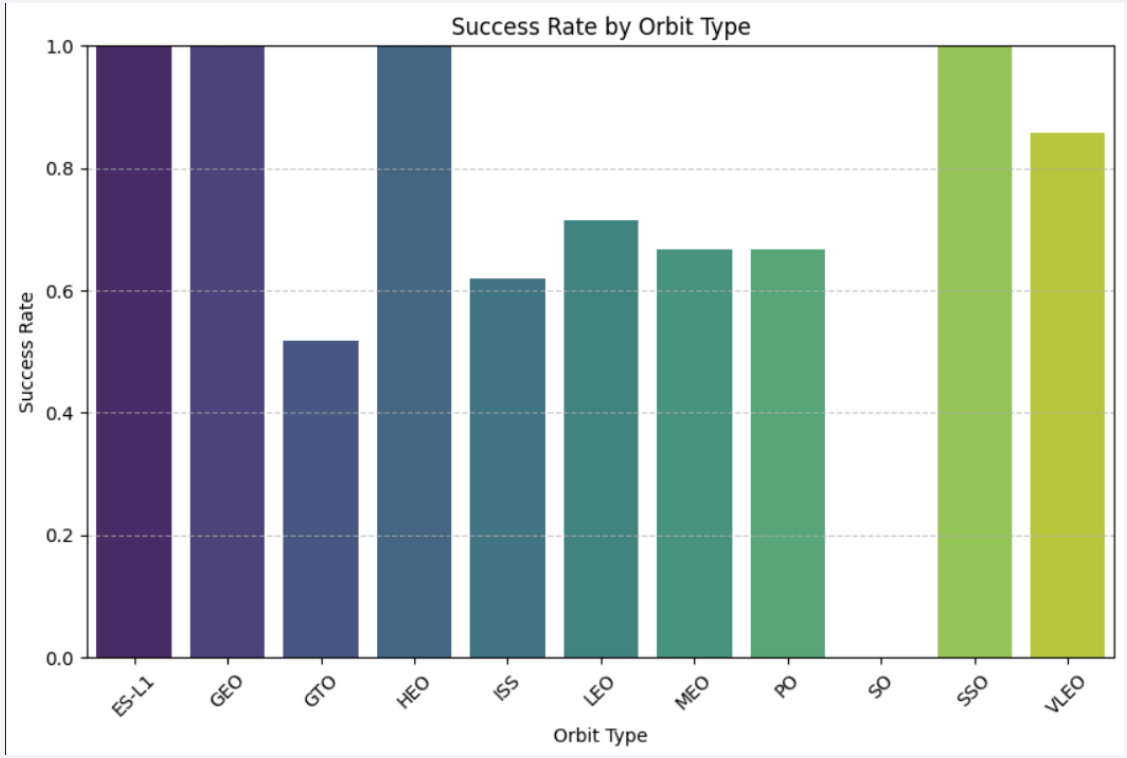
```
# landing_outcomes = values on Outcome column
landing_outcomes=df["Outcome"].value_counts()
```

`True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground p unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

EDA with Data Visualization

- **Correlation Heatmap:**
  - Used to identify relationships between variables, helping to detect potential multicollinearity.
- **Scatter Plots (Flight Number vs. Payload Mass/Launch Site, Payload Mass vs. Launch Site):**
  - Visualized how these variables relate to launch outcomes.
- **Bar Charts (Orbit Success Rate, Launch Site Success Rate):**
  - Showed which orbits and launch sites had the highest success rates.
- **Scatter Plots (Average Flight Number/Payload Mass vs. Orbit/Launch Site Success Rate):**
  - Examined the relationship between these factors and success rates.
- **Line Chart (Success Rate Over Time):**
  - Tracked the change in SpaceX launch success rates over time.
- Essentially, the charts aimed to:
- Understand variable relationships, Identify factors influencing launch success and Visualize trends and patterns in the data



Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/edadataviz.ipynb

EDA with SQL

SQL Queries Performed:

- **Unique Launch Sites:** SELECT DISTINCT LaunchSite FROM SPACEXTABLE;
- **Launch Sites Starting with 'CCA':** SELECT * FROM SPACEXTABLE WHERE LaunchSite LIKE 'CCA%';
- **Total Payload Mass Launched by NASA (CRS):** SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
- **Average Payload Mass Launched by F9 v1.1:** SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BoosterVersion = 'F9 v1.1';
- **First Successful Landing Date on Ground Pad:** SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing _Outcome = 'Success (ground pad)';
- **Booster Versions with the Highest Payload Mass:** SELECT BoosterVersion FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
- **Total Payload Mass Launched from CCAFS LC-40:** SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE LaunchSite = 'CCAFS LC-40';
- **Count of Successful and Failed Launches:** SELECT SUM(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS "Successful Launches", SUM(CASE WHEN Mission_Outcome LIKE 'Failure%' THEN 1 ELSE 0 END) AS "Failed Launches" FROM SPACEXTABLE;
- **Booster Versions with Successful Landing and Dates:** SELECT BoosterVersion, Date FROM SPACEXTABLE WHERE Landing _Outcome = 'Success (ground pad)';
- **Latitude and Longitude of Launch Sites:** SELECT LaunchSite, Latitude, Longitude FROM SPACEXTABLE

Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch sites were identified and plotted on an interactive Folium map, adding markers, circles and lines to visualize the success or failure of launches at each location.
- Launch results were classified into two categories: 0 for failed and 1 for successful.
- Through color-coded markers, the sites with the highest success rate were analyzed.
- Distances between each launch site and its environment were measured, answering questions such as:
  - Are they located near key infrastructure such as railroads, highways and coastlines?
  - Are they kept at a safe distance from urban areas?
- Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

This dashboard visualizes SpaceX launch records, including data on launch sites, payload, and mission outcomes. It includes:

- A dropdown list to select different Launch Sites.
- A pie chart to display overall success vs. failure launches.
- A slider to select payload ranges.
- A scatter chart to show the correlation between payload and launch success.
- Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Key Phrases

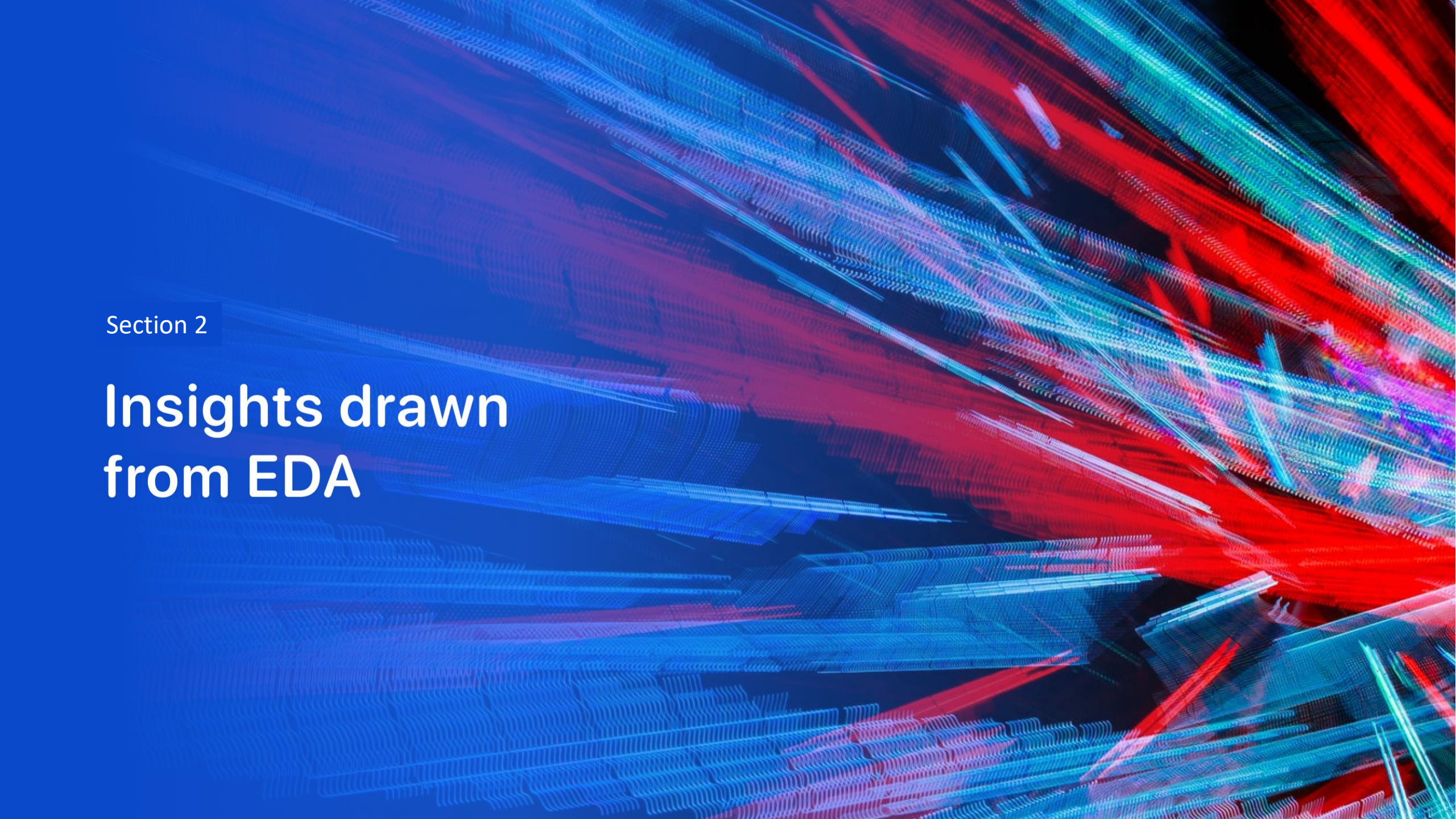
- Throughout this process, several key phrases are used:
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Model Selection
- Hyperparameter Tuning
- Model Evaluation
- Model Improvement
- Best Model Selection

In summary, building a high-performing classification model is an iterative process that involves understanding the data, selecting and tuning appropriate models, and continuously evaluating and improving the model's performance.

Link to the notebook is https://github.com/kevin444528/Final_Project_DS/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

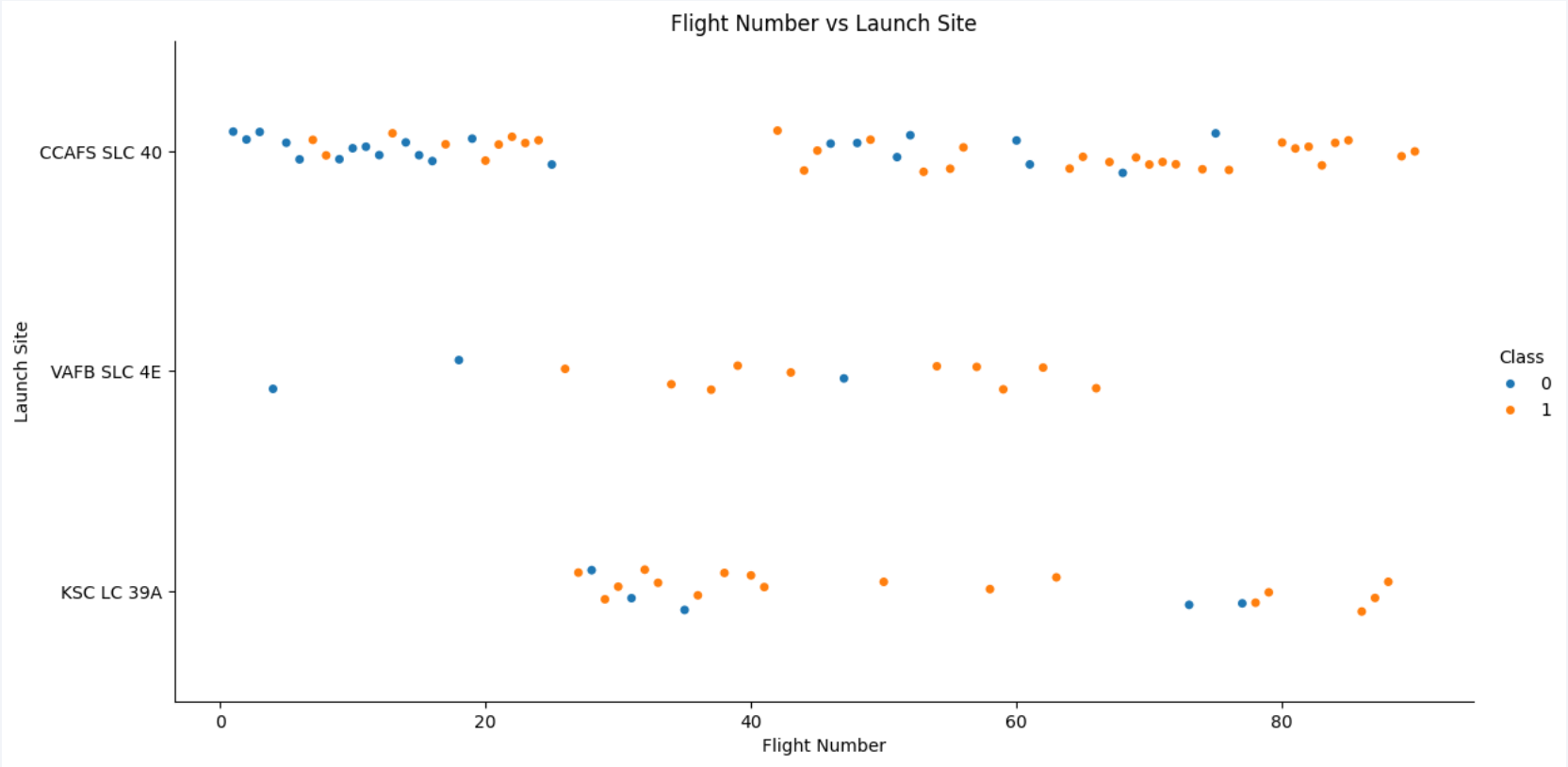
Results

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


Section 2
Insights drawn from EDA

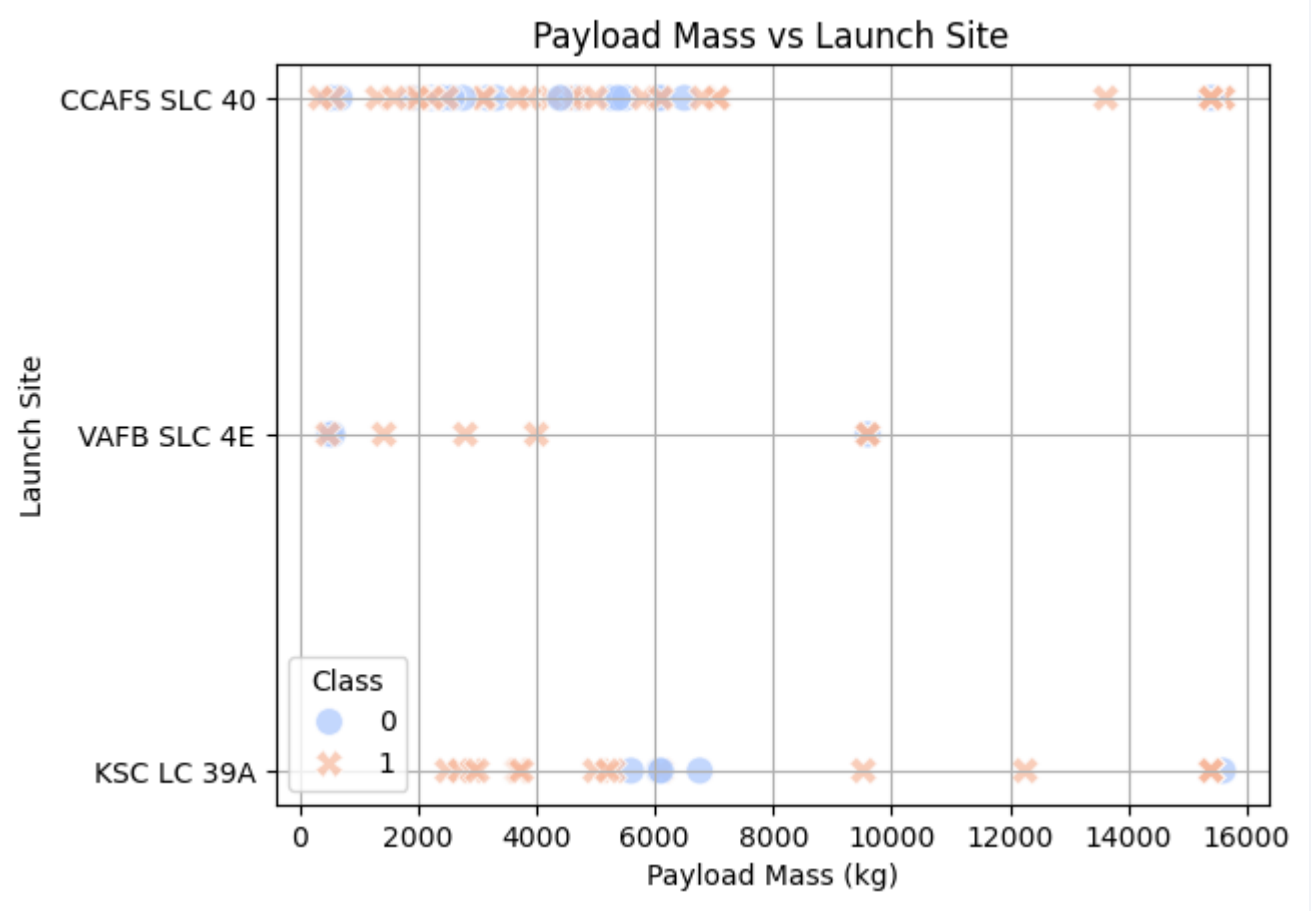
Flight Number vs. Launch Site

- **Overall Success:** The majority of launches have successful first stage landings, as indicated by the prevalence of blue dots.
- **Launch Site Impact:**
  - **CCAFS SLC 40:** This launch site has the highest number of successful landings, suggesting it might have favorable conditions or be better suited for rocket launches.
  - **VAFB SLC 4E:** This launch site also has a high success rate, but with fewer total launches compared to CCAFS SLC 40.
  - **KSC LC 39A:** This launch site has the lowest success rate among the three, with more failed landings than successful ones.



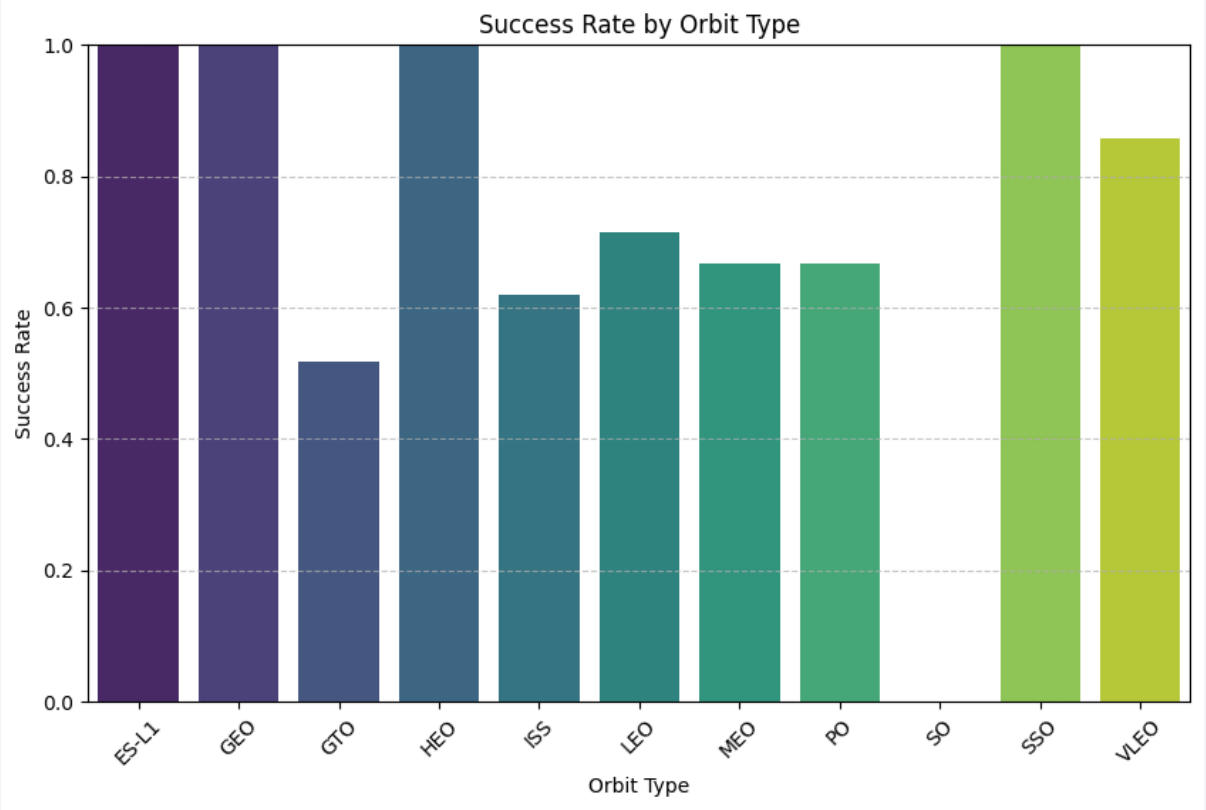
Payload vs. Launch Site

- The plot reveals that CCAFS SLC 40 handles a wide range of payloads and has a mix of successful and unsuccessful launches.
- VAFB SLC 4E shows launches with moderate to high payload masses and predominantly successful outcomes.
- KSC LC 39A features a cluster of launches with lower payload masses and a mix of successes and failures, while also having a few launches with very high payload masses



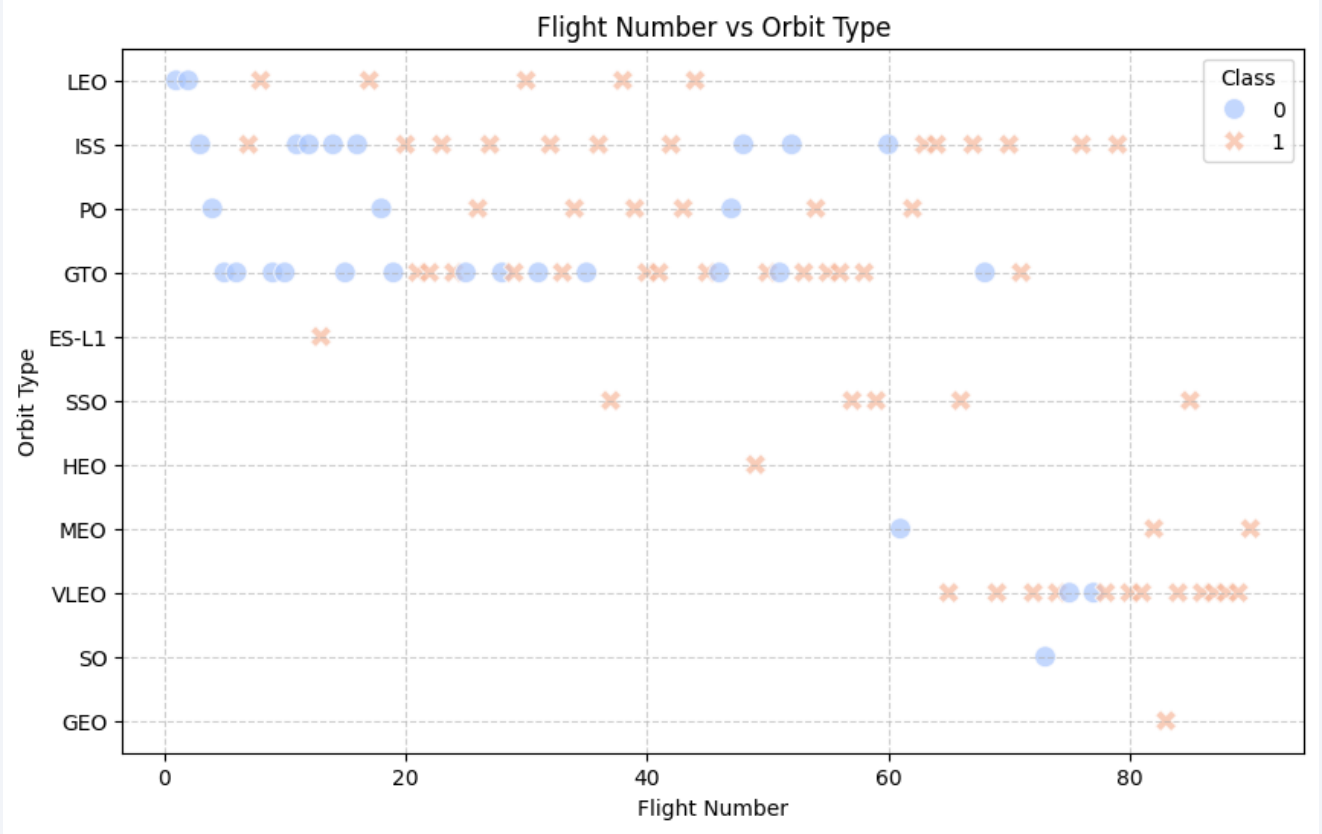
Success Rate vs. Orbit Type

This bar chart displays the success rate of SpaceX launches across various orbit types. The Y-axis represents the success rate, ranging from 0 to 1, while the X-axis categorizes each orbit type. The chart reveals significant variations in success rates depending on the orbit.



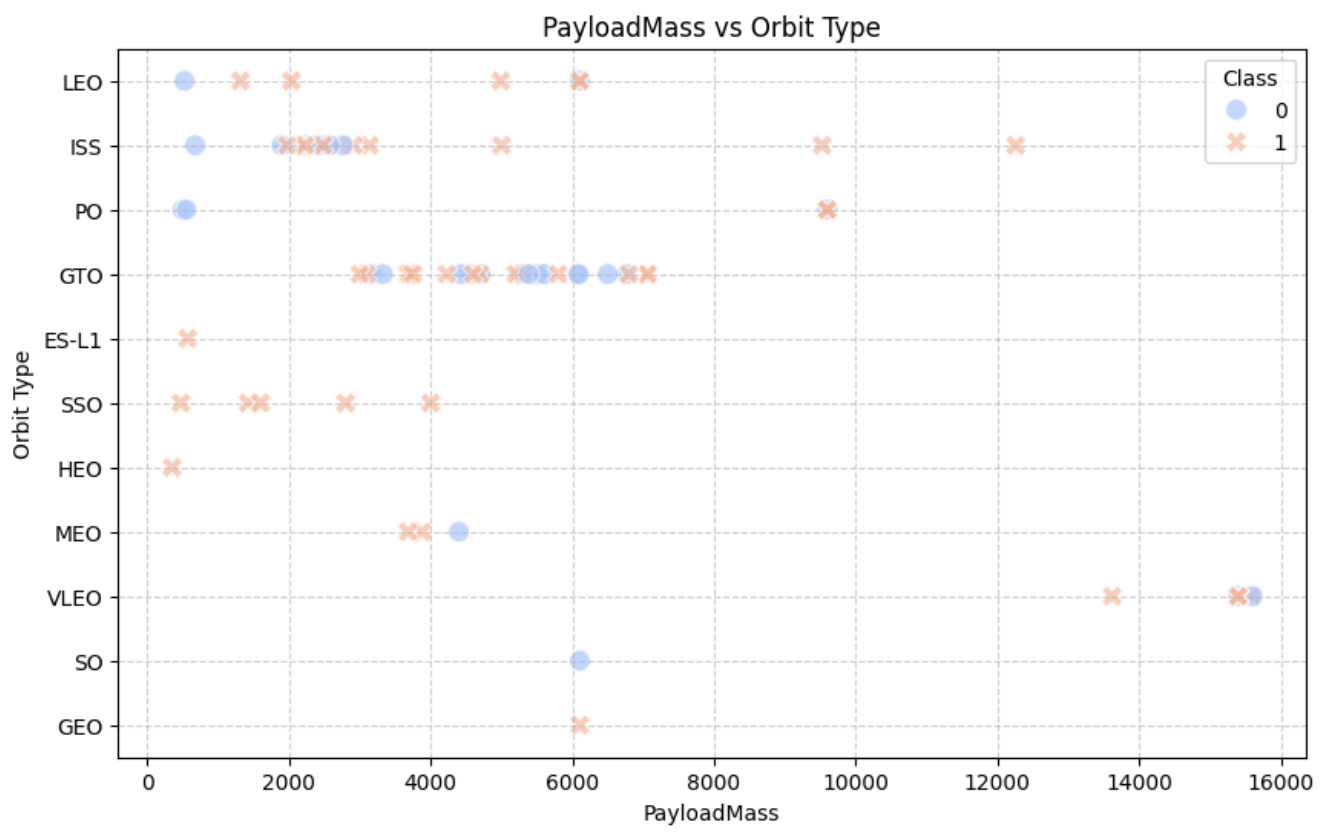
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



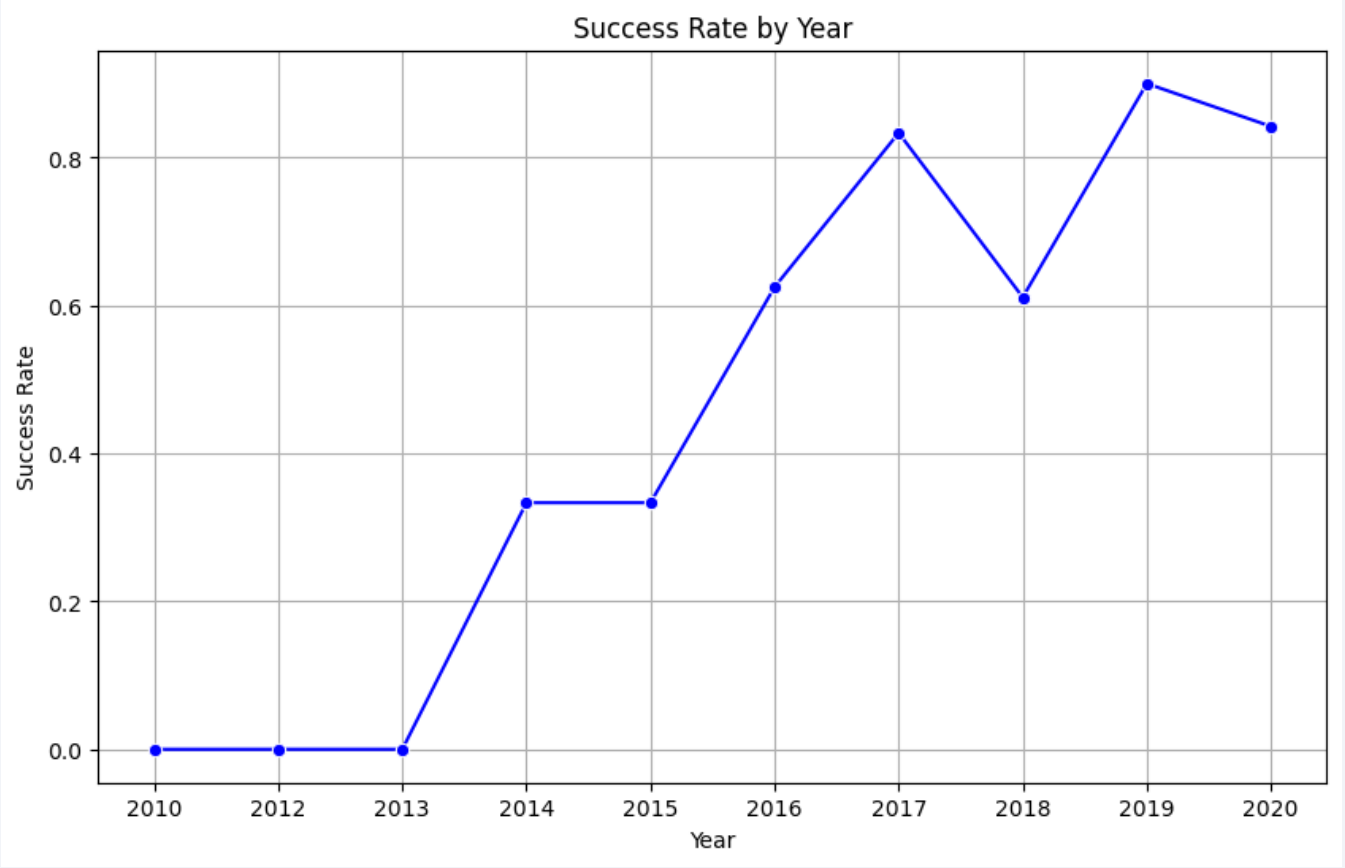
Payload vs. Orbit Type

The plot reveals that certain orbits, like GTO and LEO, have a wider distribution of payload masses, while others, like ES-L1 and SO, have launches with specific payload ranges.



Launch Success Yearly Trend

The line chart titled "Success Rate by Year" displays the SpaceX launch success rate over time. The X-axis represents the year, while the Y-axis represents the success rate, ranging from 0 to 1. The chart reveals a gradual increase in success rate from 2010 to 2016, followed by a decline in 2017.



All Launch Site Names

We used the function *group by* to show only unique launch site from Space X data.

```
%sql select Launch_Site from SPACEXTABLE group by Launch_Site
```

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

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

Launch Site Names Begin with 'CCA'

- We use *limit* to show just 5 rows and *like* for filter by patterns

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

Total Payload Mass

- The SUM() function in SQL is used to add up all the values in a specified column. In this case, it's adding up all the values in the PAYLOAD_MASS__KG_ column, which likely represents the payload mass for each launch. The result is the total payload mass of all launches in the dataset.

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

```
%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

- The AVG() function calculates the average of the values in the specified column (PAYLOAD_MASS__KG_).
- The WHERE clause filters the results to include only those rows where the Booster_Version starts with "F9 v1.1", indicated by the LIKE "F9 v1.1%" condition.
- The output shows the specific booster version (F9 v1.1 B1003) and its corresponding average payload mass

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

```
%sql select Booster_Version, AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version LIKE "F9 v1.1%"
```

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

Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1 B1003	2534.6666666666665

First Successful Ground Landing Date

- The SQL query SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome = "Success" retrieves the earliest date when a successful landing occurred. The result, 2018-07-22, indicates that the first successful landing in the dataset happened on July 22, 2018.

```
%sql select MIN(DATE) from SPACEXTABLE where Landing_Outcome  = "Success"
```

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

MIN(DATE)
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL lists the booster versions that meet the following criteria:
  - Landing: The Landing_Outcome is "Success".
  - Payload Mass Range: The PAYLOAD_MASS__KG_ is greater than 4000 kg and less than 6000 kg.

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 Booster_Version from SPACEXTABLE where Landing_Outcome  = "Success" AND PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_<6000
```

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

Booster_Version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- The SQL query counts the number of occurrences for each unique value in the Mission_Outcome column. This indicates that there have been 98 successful missions, one failure (in flight), and two other missions with successful outcomes (one with an unclear payload status).

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

- The SQL query aims to find the booster versions that carried the maximum payload mass. It does this in two steps:
- Find the Maximum Payloadexplanation
- Find the Booster Versions:

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

```
maximo = %sql SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;
resultado= maximo[0][0]
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = :resultado;
```

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

- The SQL query retrieves the month, date, launch site, and booster version for each launch in 2015 that resulted in a "Failure (drone ship)" landing outcome.

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, 6,2),Date, Launch_Site,Booster_Version  FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)'  AND substr(Date,0,5)='2015
```

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

substr(Date, 6,2)	Date	Launch_Site	Booster_Version
01	2015-01-10	CCAFS LC-40	F9 v1.1 B1012
04	2015-04-14	CCAFS LC-40	F9 v1.1 B1015

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

The SQL query *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;* counts the occurrences of each landing outcome within the specified date range and ranks them in descending order based on their counts.

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

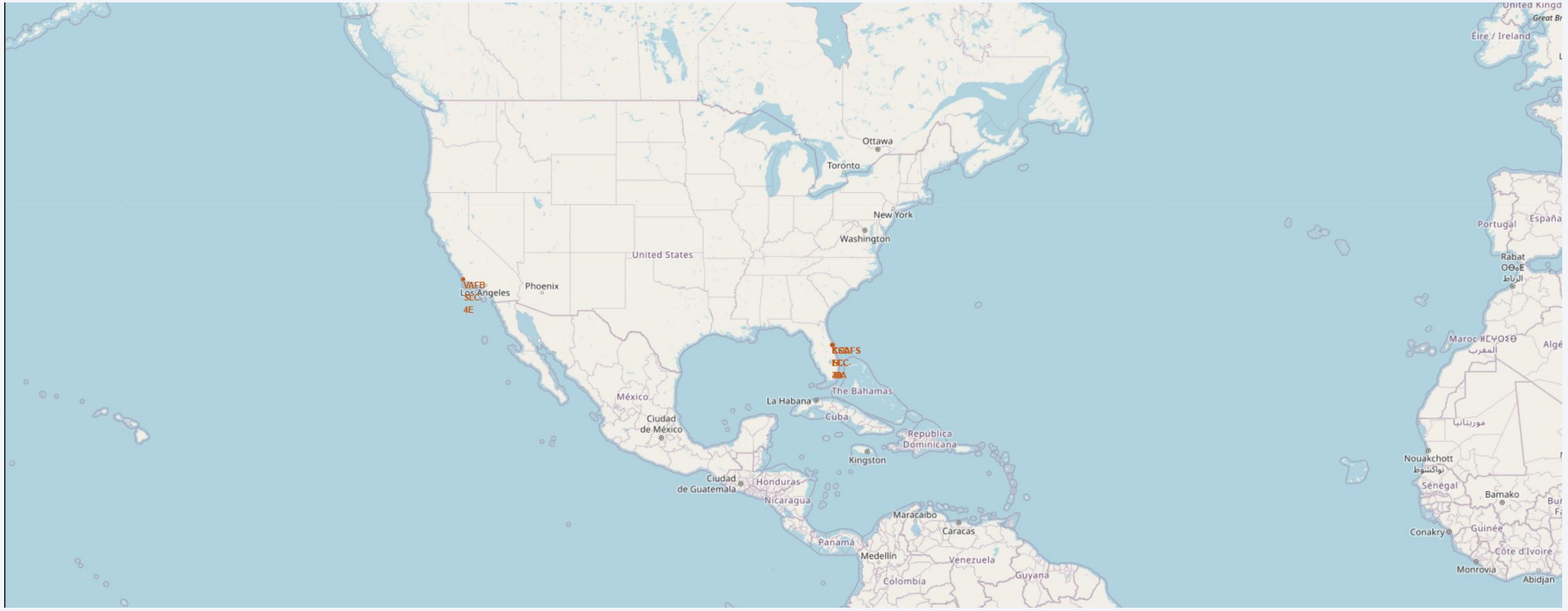
```
%sql SELECT  Landing_Outcome, count() AS TOTAL FROM SPACEXTABLE WHERE DATE between "2010-06-04" AND "2017-03-20" Group by Landing_Outcome ORDER BY TOTAL DESC ;
```

Section 3

Launch Sites Proximities Analysis

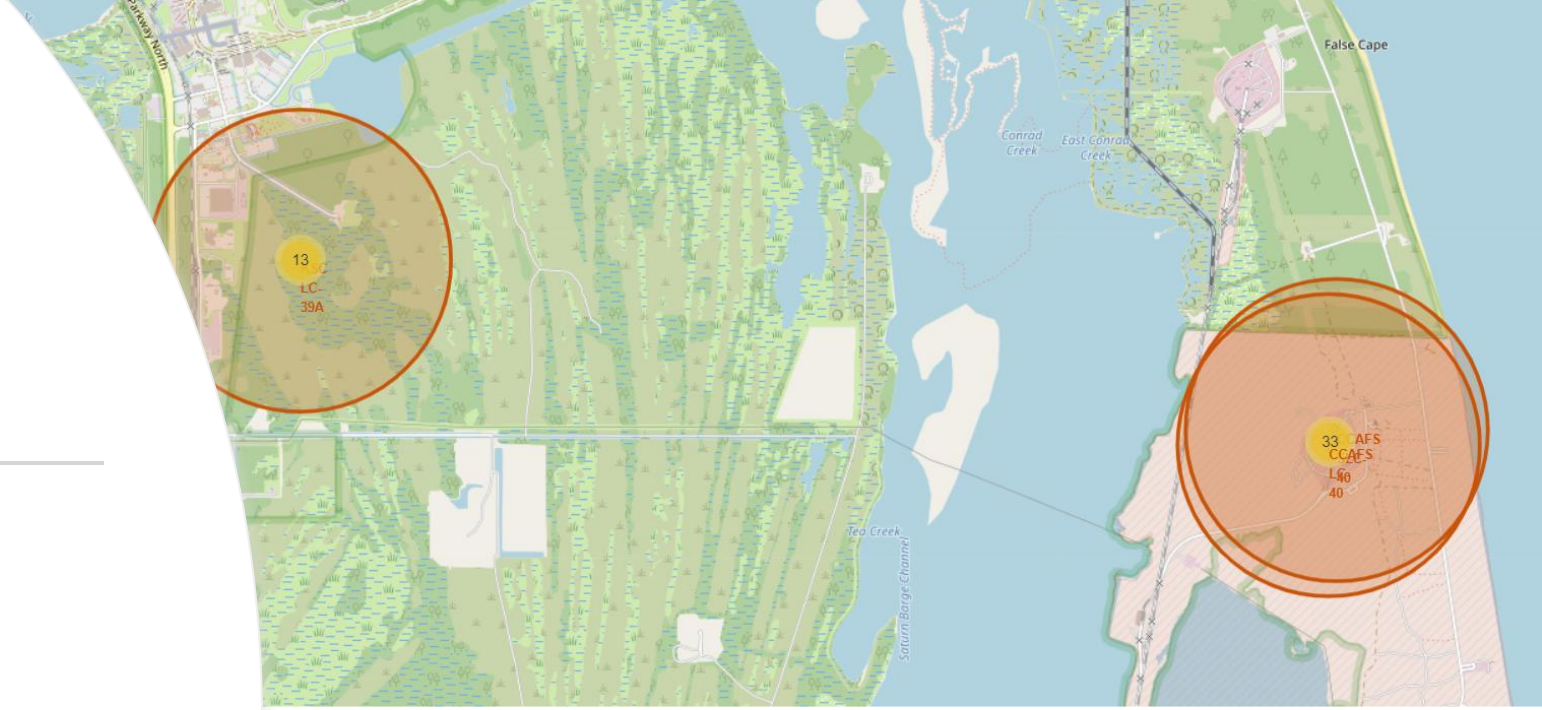
All launching sites global map markers

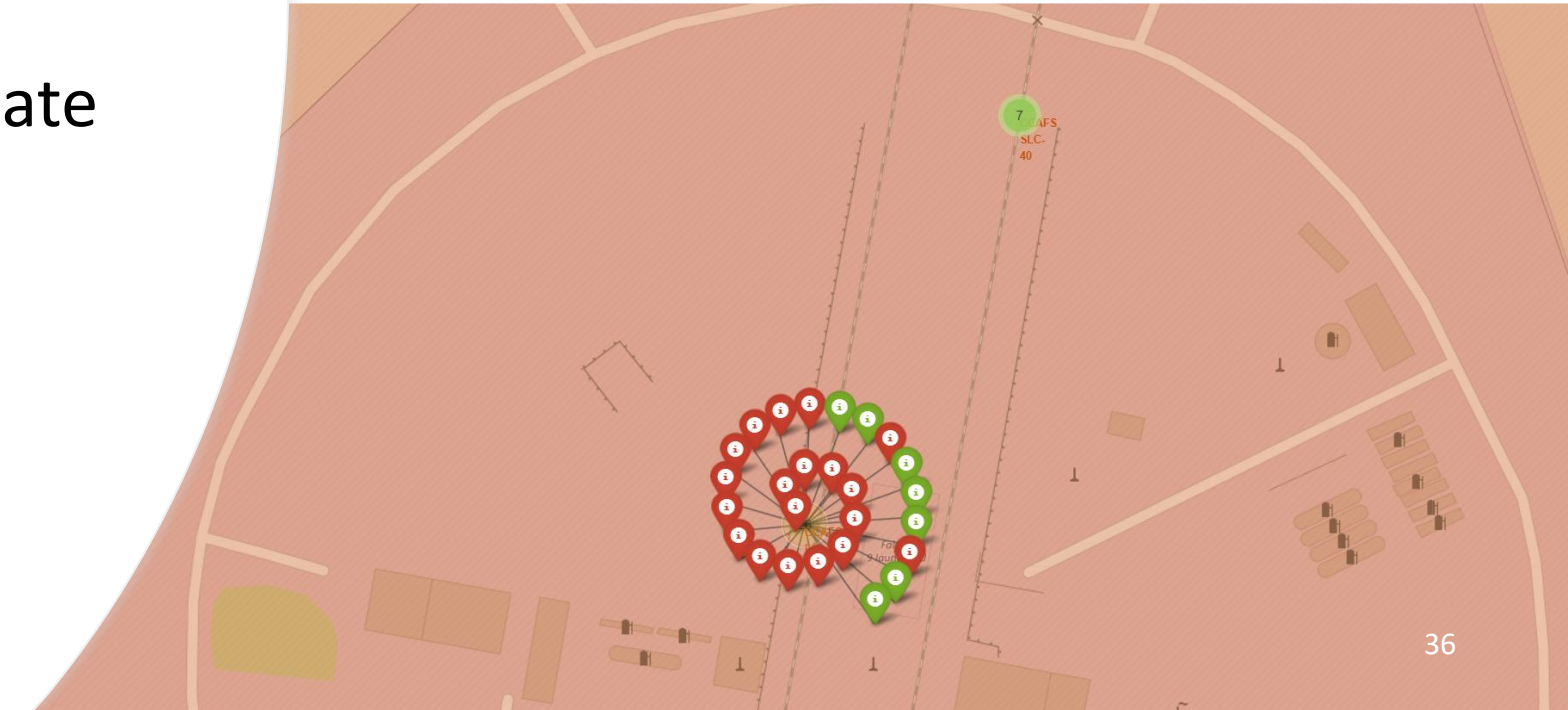
- As can be seen, all launch sites are located in the USA.

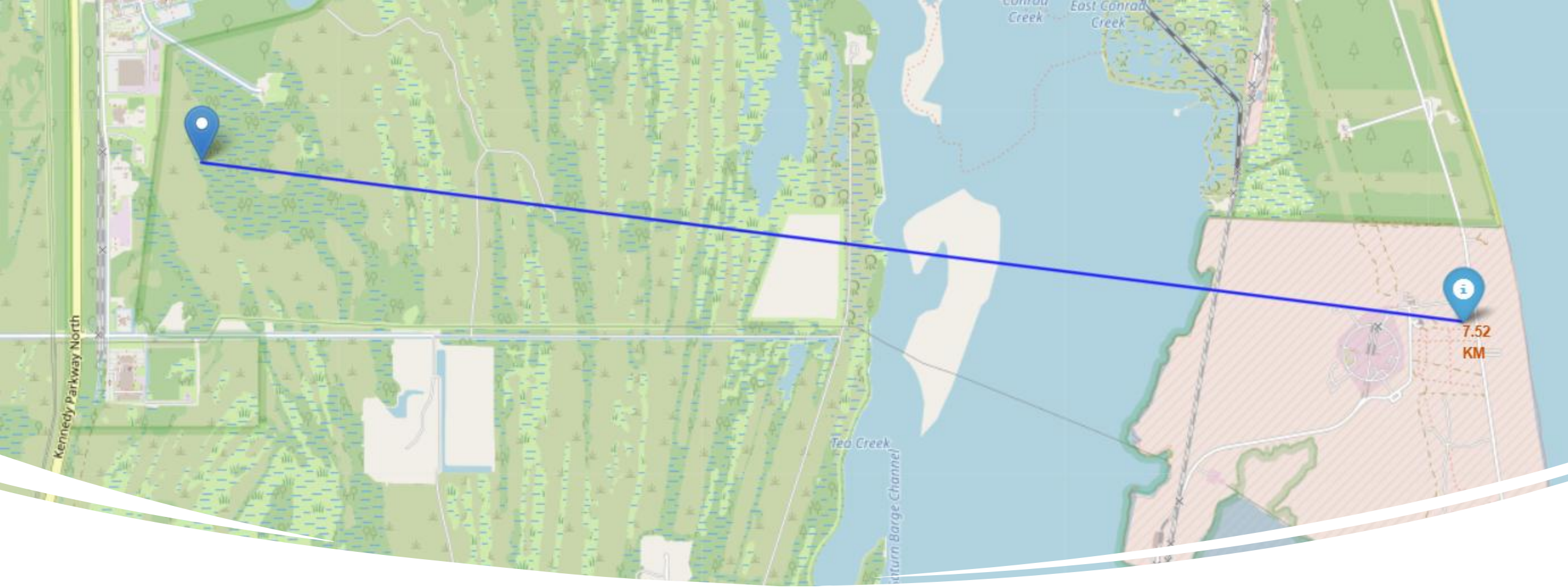


Launch Outcomes by Site

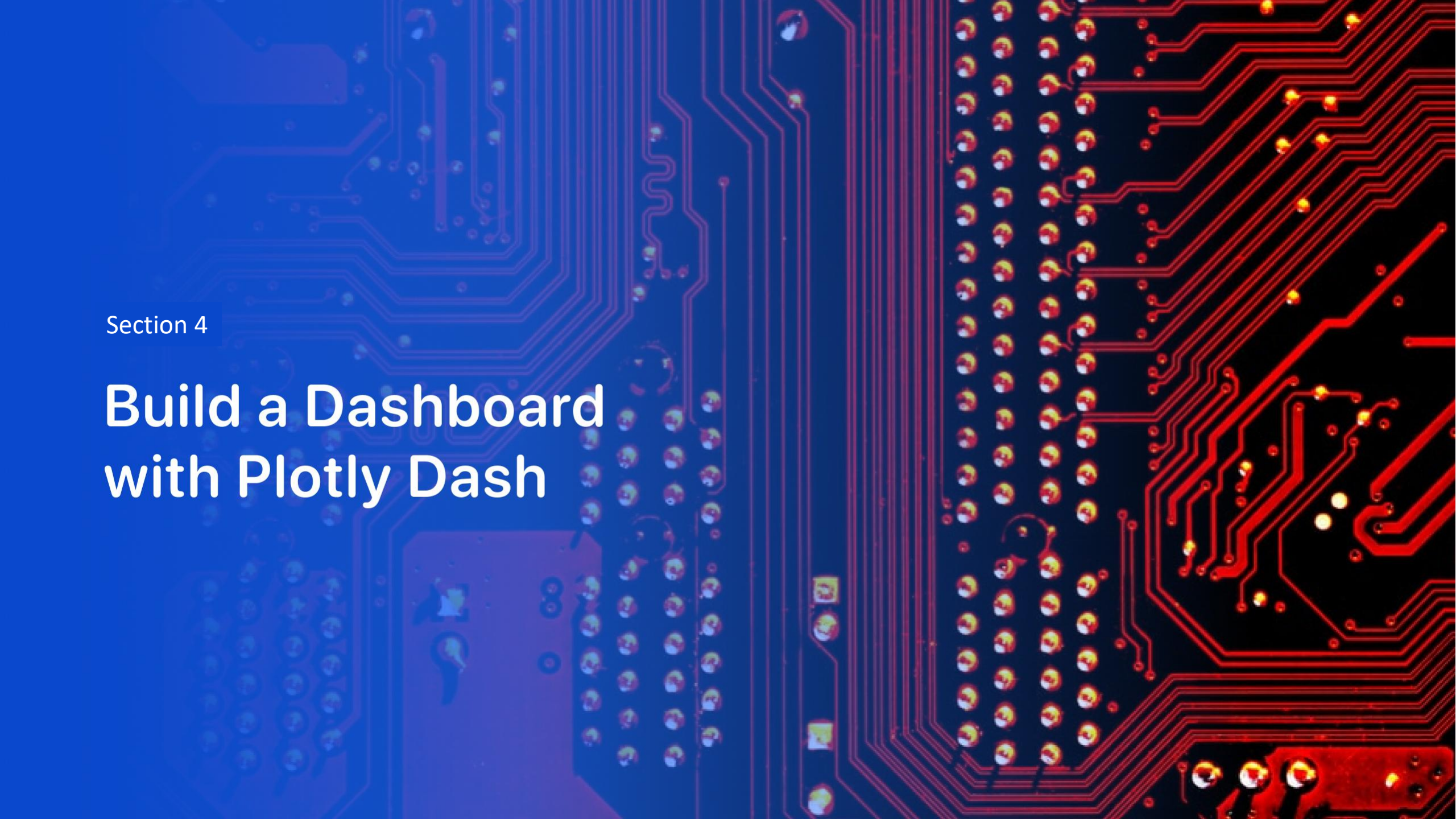
Green markers indicate successful and red ones indicate failure





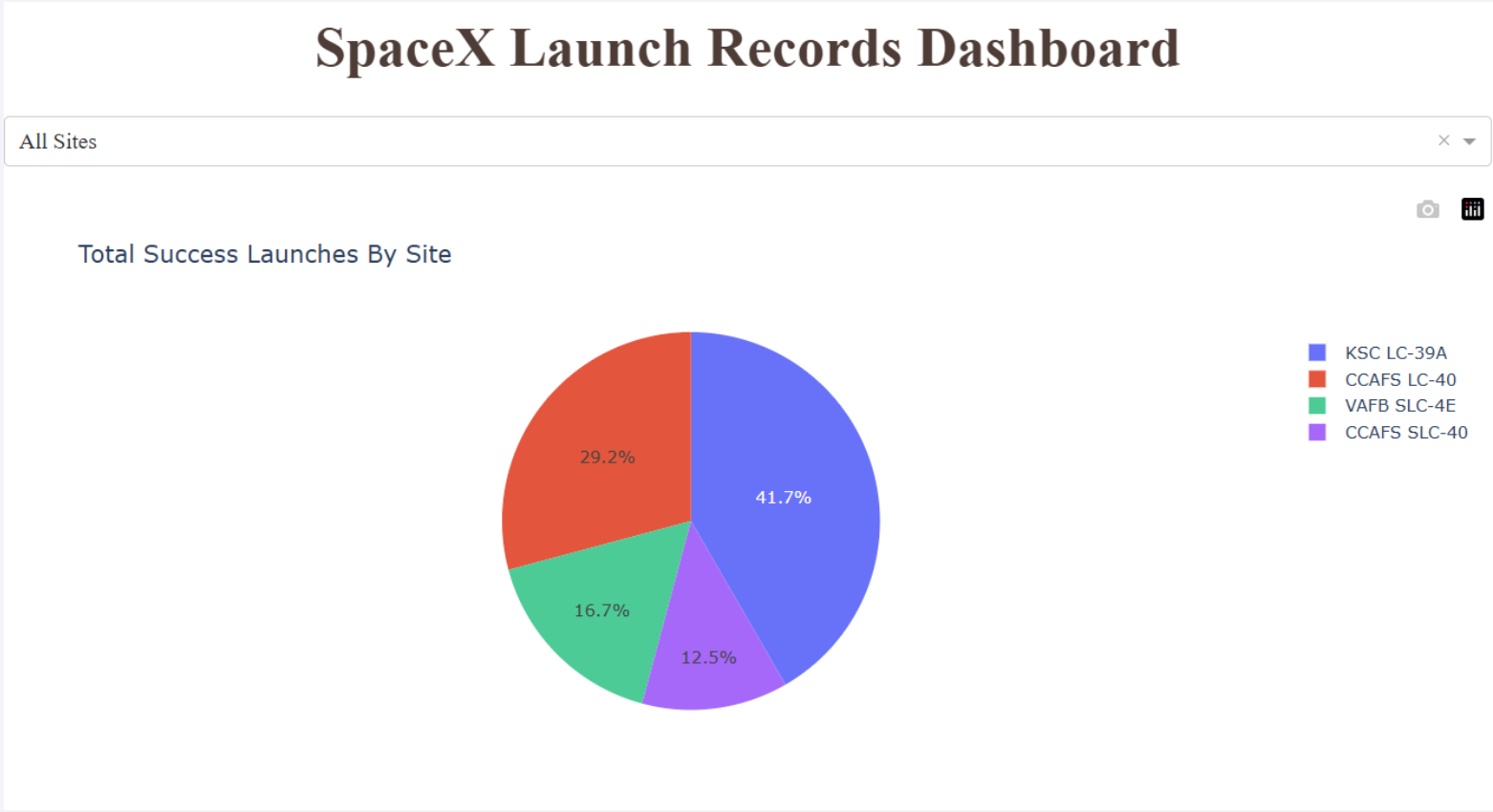


Launch Site distance to landmarks


Section 4
Build a Dashboard with Plotly Dash

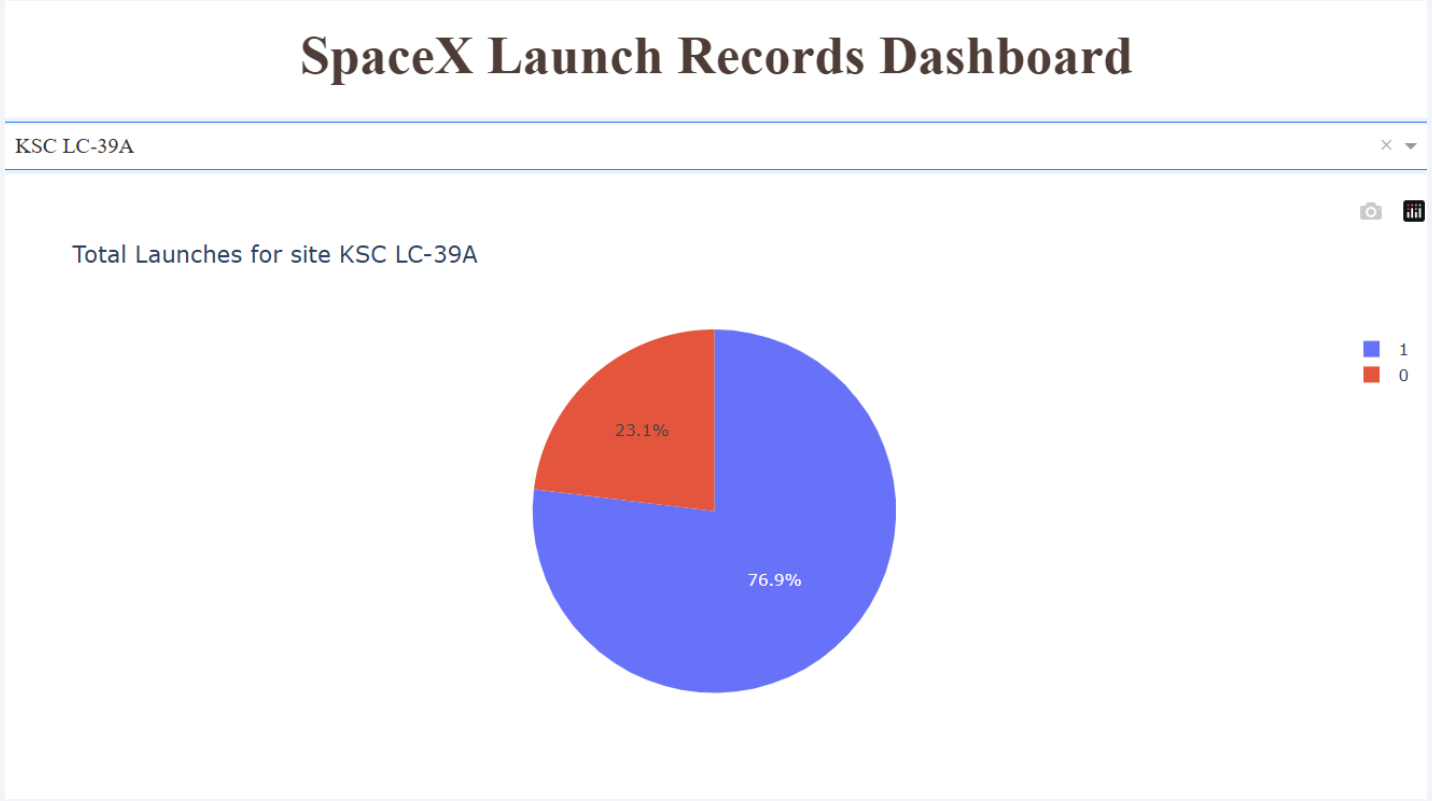
Success by launch sites

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



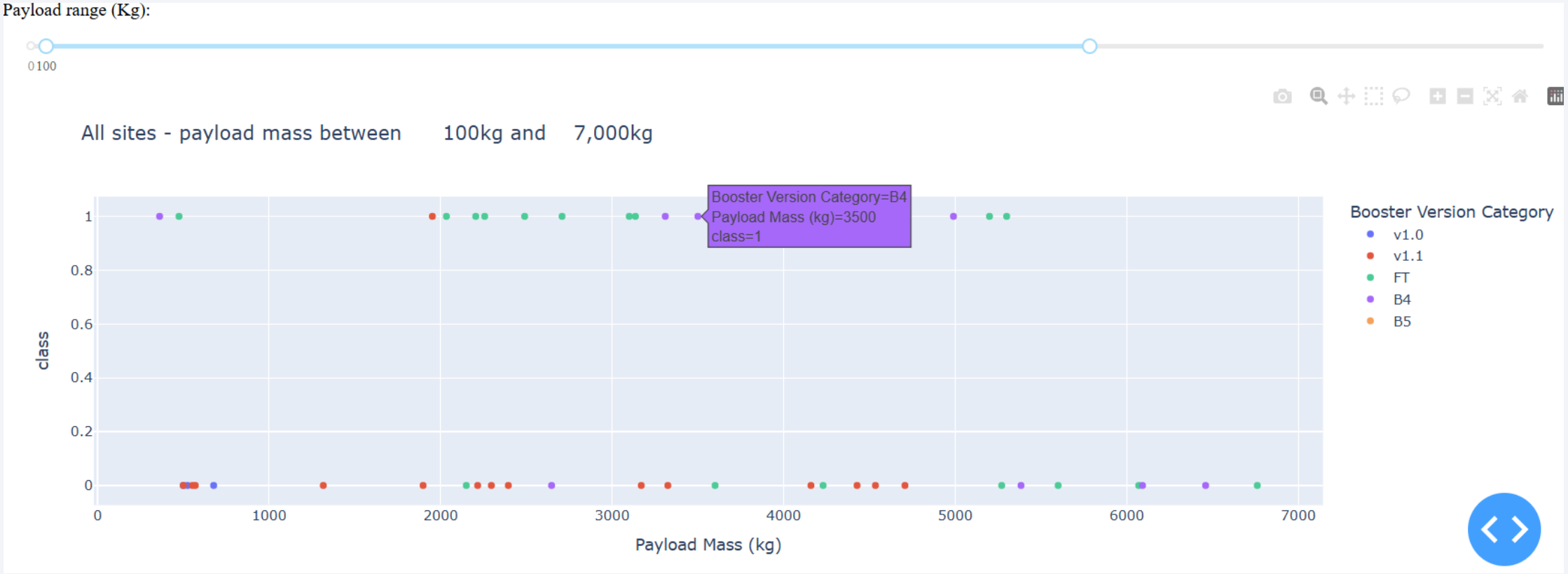
Successful Ratio for KSC LC-39A

- 76,9 percent of launches are successful here



Payload Mass vs Launch Outcomes

Medium weight loads are quite successful

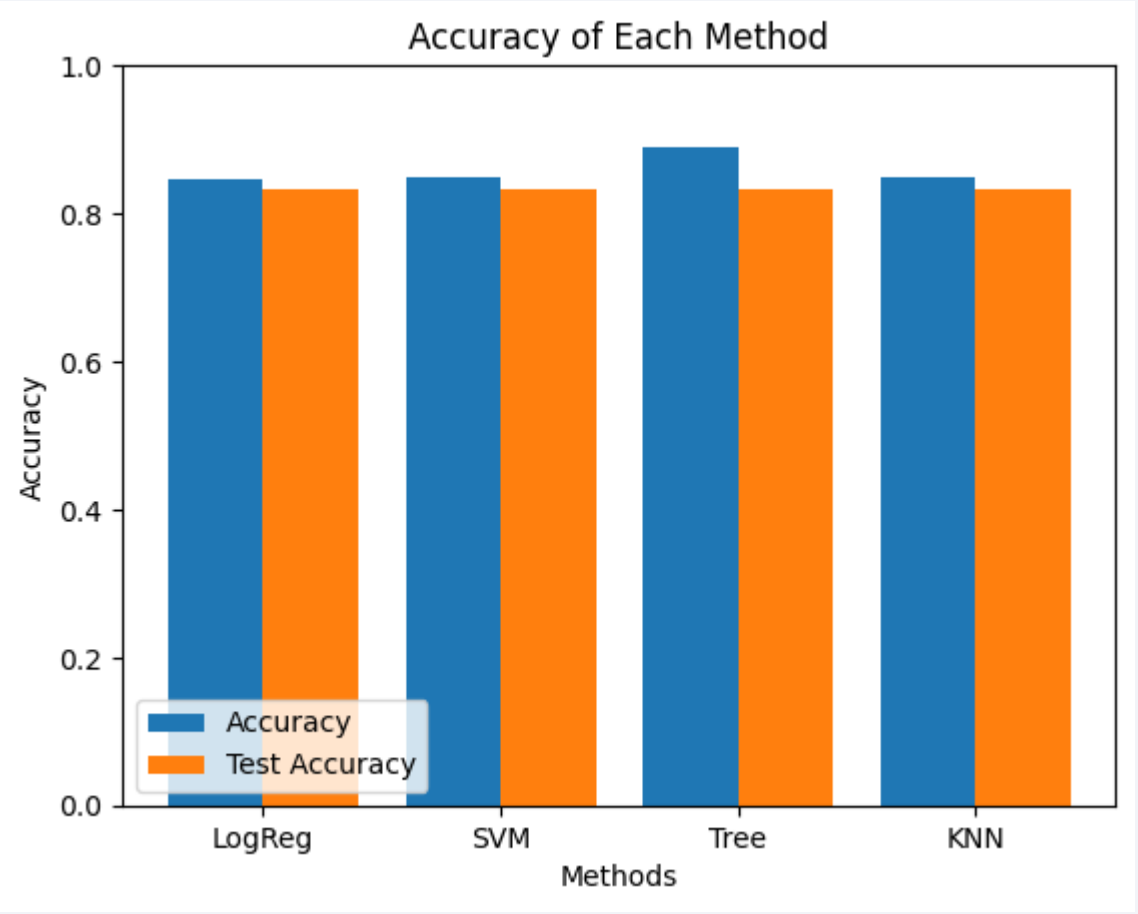


Section 5

Predictive Analysis (Classification)

Classification Accuracy

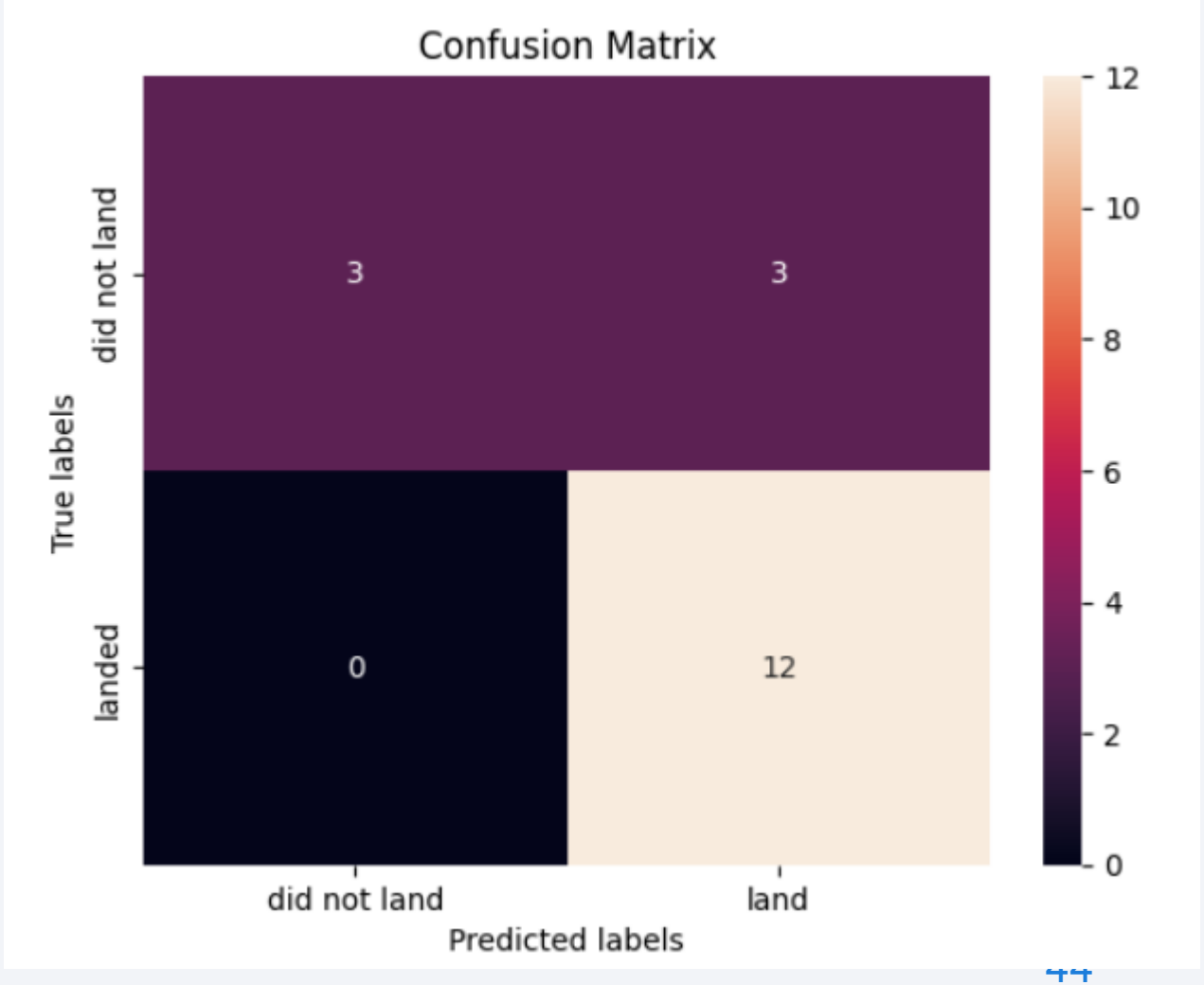
- The chart suggests that the Decision Tree model is the best performing model among the four, as it has the highest accuracy on the training data.
- However, the consistent test accuracy across all models indicates a potential issue with overfitting or the test dataset. Further analysis, such as cross-validation or increasing the size of the test dataset, might be necessary to get a more reliable assessment of the models' performance



Model	Accuracy	TestAccuracy
LogReg	0.84643	0.8333333333333334
SVM	0.84821	0.8333333333333334
Tree	0.88929	0.8333333333333334
KNN	0.84821	0.8333333333333334

Confusion Matrix

This confusion matrix evaluates the performance of a Decision Tree model, which is a popular and effective classification algorithm. The matrix reveals that the model correctly predicted 12 instances of successful landings ("landed"), but incorrectly predicted 3 instances where landings failed ("did not land") as successful. Conversely, it accurately identified 3 failures but misclassified no successful landings as failures. This suggests the Decision Tree model has a strong ability to predict successful landings



Conclusions

Throughout the project, a variety of data sources were analyzed, refining the findings as the process progressed. Several key trends were identified:

- The most prominent launch site is KSCLC-39A, which showed the highest number of successful launches.
- Rocket launches with a payload greater than 7,000 kg have a lower risk, suggesting a direct relationship between payload and probability of success.
- Although most launches have successful outcomes, successful landings appear to improve over time, reflecting the evolution in rocket processes and technology.
- The use of Decision Tree Classifier proved to be effective in predicting successful landings, which could optimize processes and increase profits.
- It was also observed that the volume of launches at a site increases the launch success rate, indicating that accumulated experience plays a crucial role in performance.
- The launch success rate began to improve significantly starting in 2013, with a positive trend continuing through 2020.
- The ES-L1, GEO, HEO, SSO and VLEO orbits had the highest success rates.
- The Decision Tree Classifier proved to be the best machine learning algorithm for this task, highlighting its predictive capability in this context.

Thank you!