IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nishank
15 Jul 2025

Outline

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

Executive Summary

- Summary of methodologies

  The analysis focused on predicting the success of Falcon 9 rocket landings by leveraging comprehensive data collection and advanced analytics. Data was collected from two primary sources—the SpaceX API and a Wikipedia page via web scraping. After performing thorough data wrangling to address missing values and ensure data quality, we conducted exploratory data analysis using SQL and a variety of visualization tools. Interactive data visualizations were created with Folium for geographic insights, and dashboards were developed using Dash. The final phase involved training and evaluating multiple machine learning models to predict landing outcomes.

- Summary of all results

  Exploratory data analysis yielded key insights into historical Falcon 9 launches and successes, supported by interactive visualizations and dashboards. Predictive analytics were performed using four machine learning models: logistic regression, support vector machine, decision tree, and k-nearest neighbor. Among these, the decision tree model demonstrated the highest accuracy at 90.35%. Screenshots from the visual analytics further illustrate the findings and support the effectiveness of the data-driven approach.

Introduction

- **Project Background and Context**
  SpaceX has disrupted the commercial space industry by offering Falcon 9 rocket launches at $62 million, a fraction of the $165 million or more typically charged by other providers. This cost advantage largely stems from SpaceX's ability to reuse the rocket's first stage. Accurately predicting whether the first stage will land successfully is therefore crucial for understanding true launch costs. Such predictive insights are also valuable for potential competitors seeking to enter the launch market. The objective of this project is to develop a machine learning pipeline capable of forecasting the likelihood of a successful first-stage landing.
- **Research Questions**
  What key factors influence the successful landing of a rocket's first stage?
  How do various features interact to impact landing outcomes?
  What operational conditions must be met to achieve consistent, successful rocket landings?

Section 1

Methodology

Methodology

Executive Summary

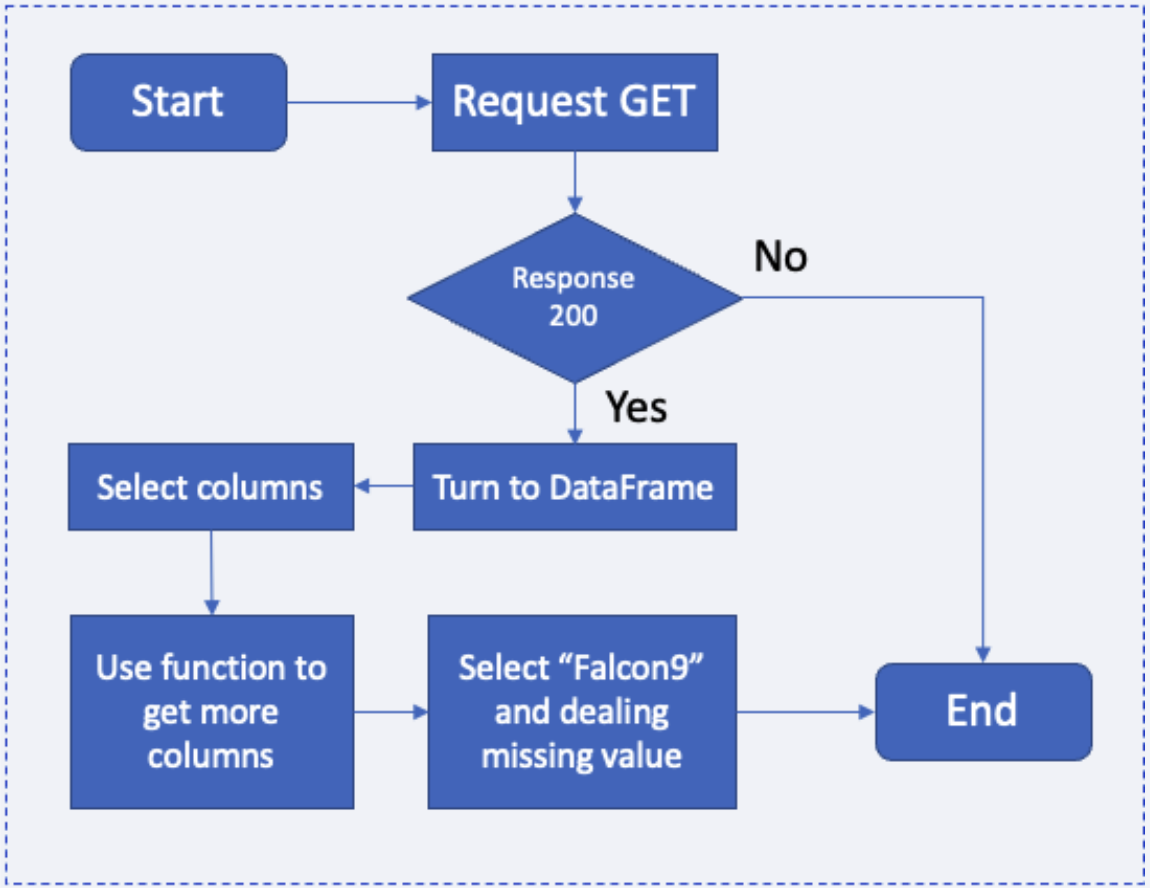
- Data collection methodology:
  - Gathered datasets from both the SpaceX API (ceXAPI(https://api.spacexdata.com/v4/rockets/) and by scraping relevant Wikipedia pages (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches) .
- Perform data wrangling
  - Addressed and handled missing data values by creating a landing outcome label based on outcome data after summarizing and analyzing 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

- SpaceX - API
  Data was collected from the SpaceX API, which included the following fields: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Lock, ReusedCount, Serial, Longitude, and Latitude.
- SpaceX Falcon 9 – Scraping
  Data was gathered from the Falcon 9 Launch Wikipedia page via its URL and transformed into a DataFrame, encompassing: Flight Number, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Booster Version, Booster Landing, Date, and Time.

Data Collection – SpaceX API

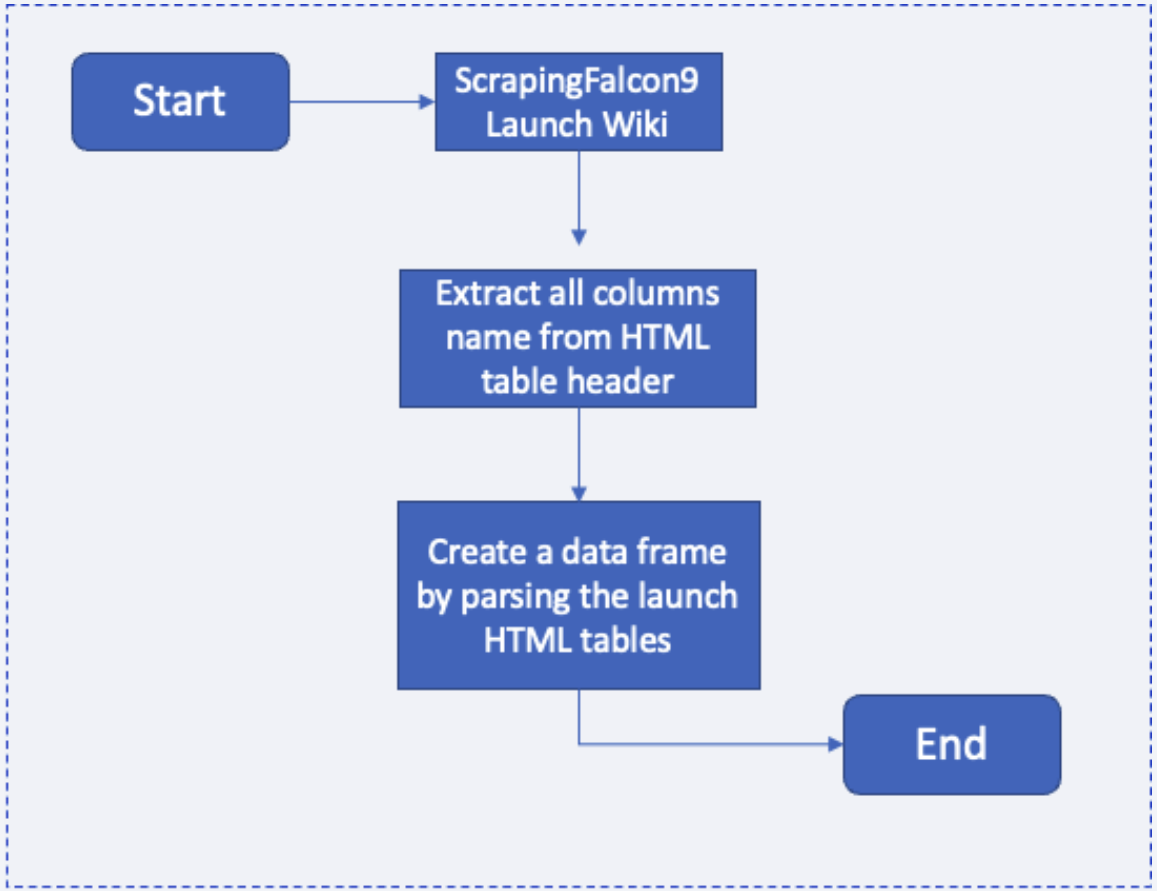
- Requested and parsed the SpaceX data using a GET request
  - Converted the JSON data to a DataFrame.
  - Utilized the following columns: rocket, payloads, launchpad, and cores.
  - Extracted BoosterVersion, LaunchSite, PayloadData, CoreData, and other relevant information.
- Filtered the DataFrame to include only entries where BoosterVersion is "Falcon 9."
- Replaced missing values with the mean.
- GitHub: [spacex-data-collection-api.ipynb]



Data Collection - Scraping

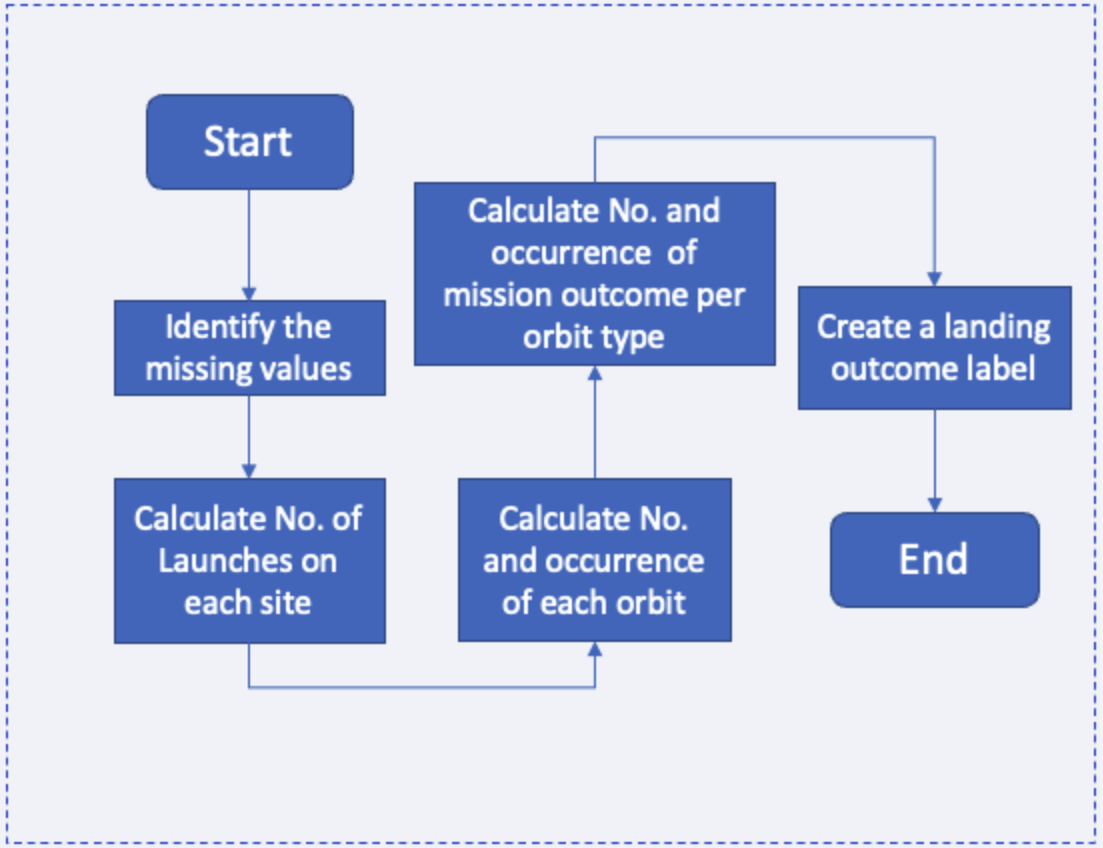
- Requested the Falcon 9 Launch Wikipedia page from the URL.
  - Used BeautifulSoup for web scraping.
  - Extracted all column names from the HTML table header.
  - Created a DataFrame by parsing the launch HTML tables.

- GitHub: [webscraping.ipynb]



Data Wrangling

- Analyzed the data.
  - Identified missing values.
  - Calculated the number of launches at each launch site.
  - Determined the number and frequency of each orbit.
  - Calculated the number and occurrences of mission outcomes per orbit type.
  - Created a landing outcome label (class).
- GitHub: [spacex-data_wrangling.ipynb]



EDA with Data Visualization

- Scatter and Line Chart
  Used numerical data to analyze the trend and relationship between two numerical variables
- Bar Chart
  Utilized categorical data to compare values across different categories.
- GitHub: eda-dataviz.ipynb

EDA with SQL

- SELECT DISTINCT: display the unique records
- WHERE Launch_Site LIKE 'CCA%': display records where Launch_Site with pattern 'CCA%'
- SUM(PAYLOAD_MASS_KG_): display total payload mass
- AVG (PAYLOAD_MASS_KG_): display average payload mass
- MIN(Date): display minimum date
- PAYLOAD_MASS_KG_ BETWEEN 4000 AND 6000: condition mass greater than 4000 but less than 6000
- GROUP BY Mission_Outcome: group the data by Mission_Outcome
- GitHub: eda-dataviz.ipynb

Build an Interactive Map with Folium

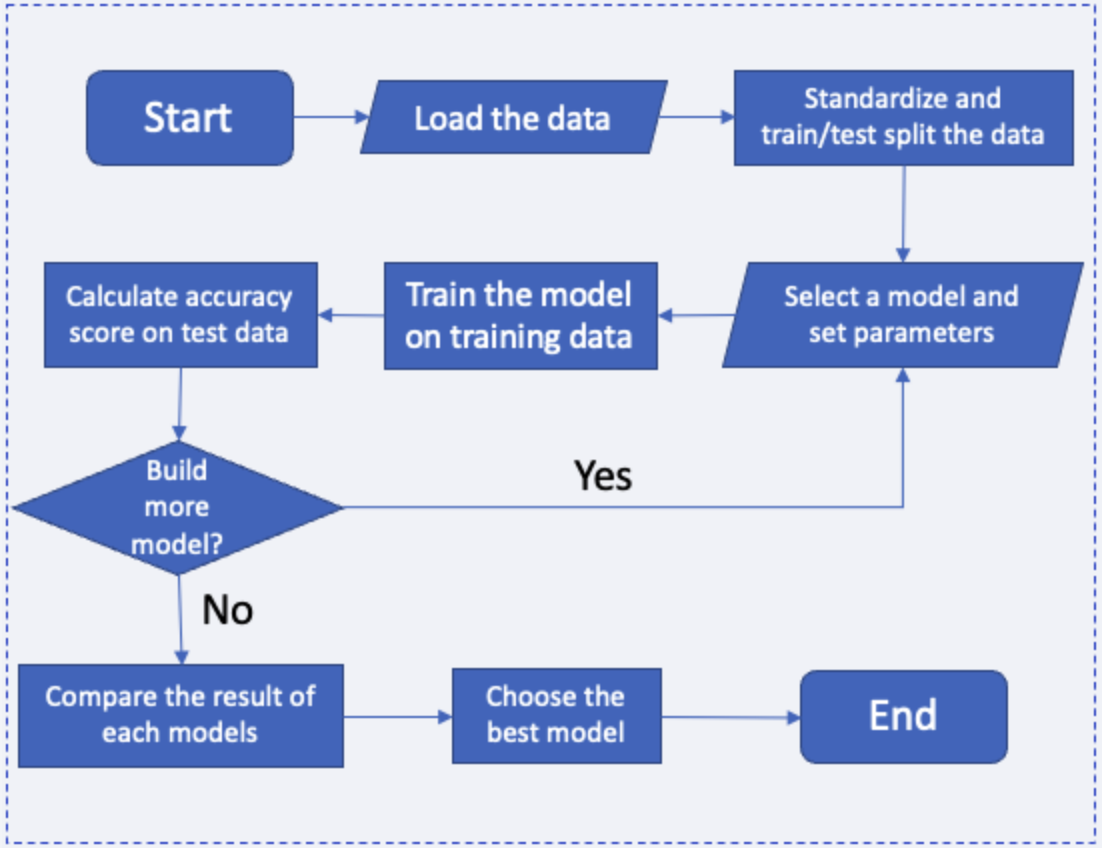
- **Map:** Initialized the map with a specified center location.
- **Marker:** Placed a marker to represent a specific location on the map.
- **Circle:** Drew a circle around the desired location.
- **Lines:** Connected multiple points on the map using lines.
- **MarkerCluster:** Grouped multiple markers at the same coordinates using a marker cluster.
- **GitHub:** launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Dropdown:** Used to capture the selected launch site type as input. Allowed users to choose from predefined options to display different types of graphs based on the selection.
- **Pie Chart:** Displayed the proportion of successful launches across all sites when “All Sites” was selected, and showed the success vs. failure ratio for individual launch sites.
- **Scatter Chart:** Plotted the relationship between Payload Mass (kg) and launch outcome (class).
- **GitHub:** dashboard

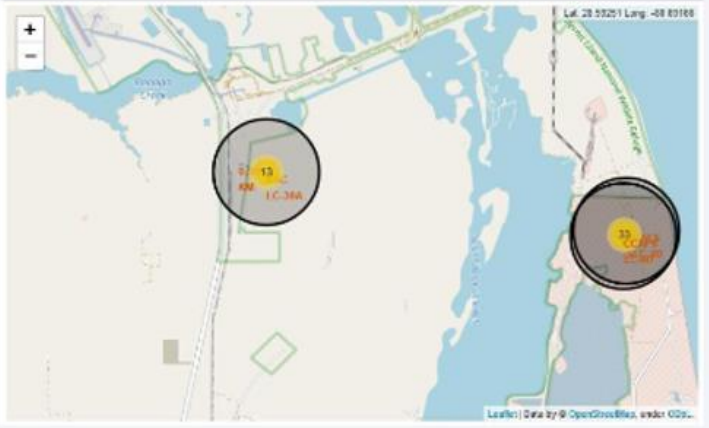
Predictive Analysis (Classification)

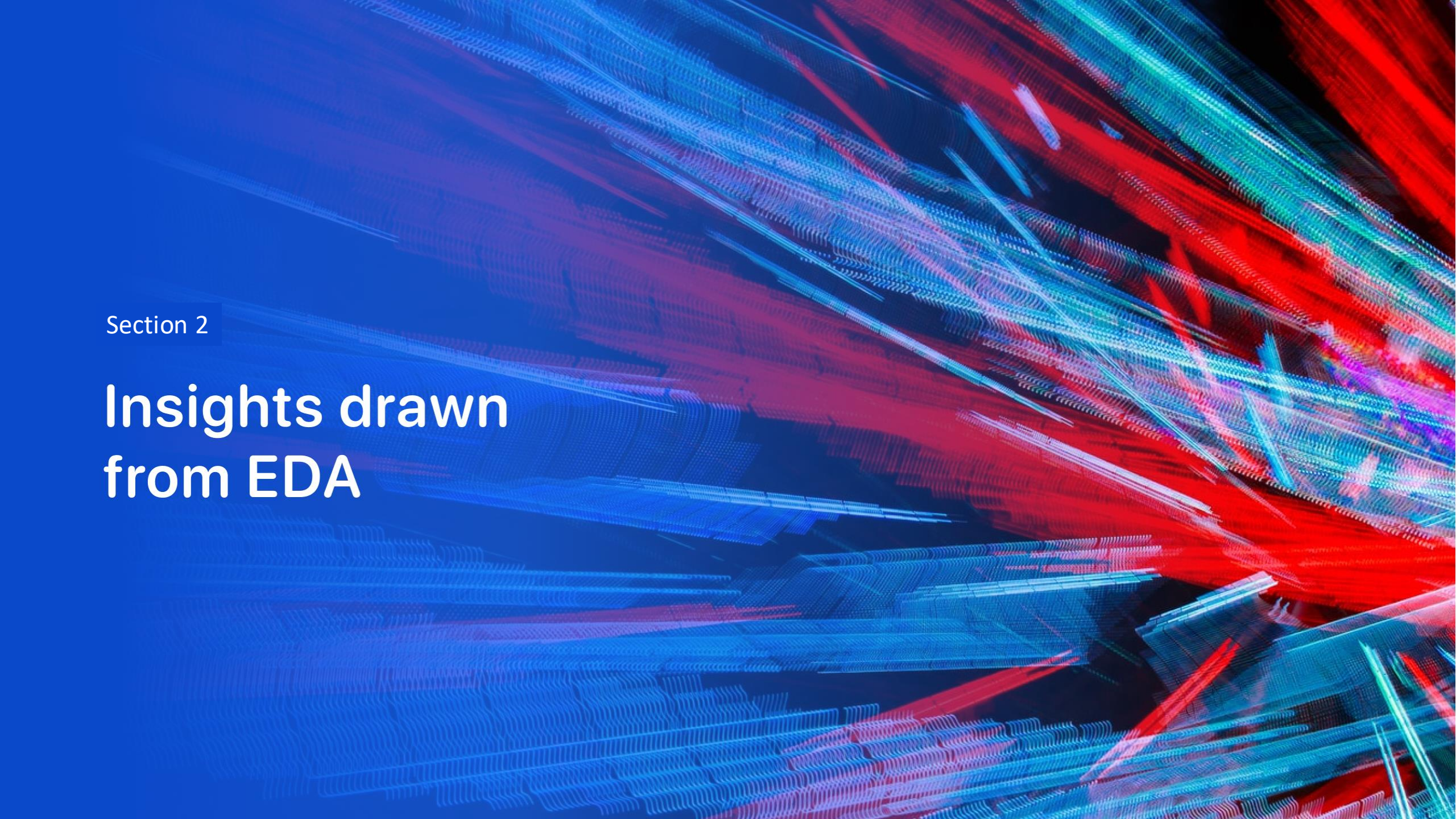
- **Data Preparation:** Standardized the dataset and split it into training and testing subsets.
- **Model Selection:** Chose a classification model and defined its parameters.
- **Model Training:** Trained the selected model on the training data.
- **Evaluation:** Calculated the accuracy score using the test dataset.
- **Model Iteration:** Assessed whether additional models needed to be trained.
- **Model Comparison:** Compared the performance of all models and selected the best-performing one.
- **GitHub:** SpaceX_Machine_Learning_Prediction.ipynb



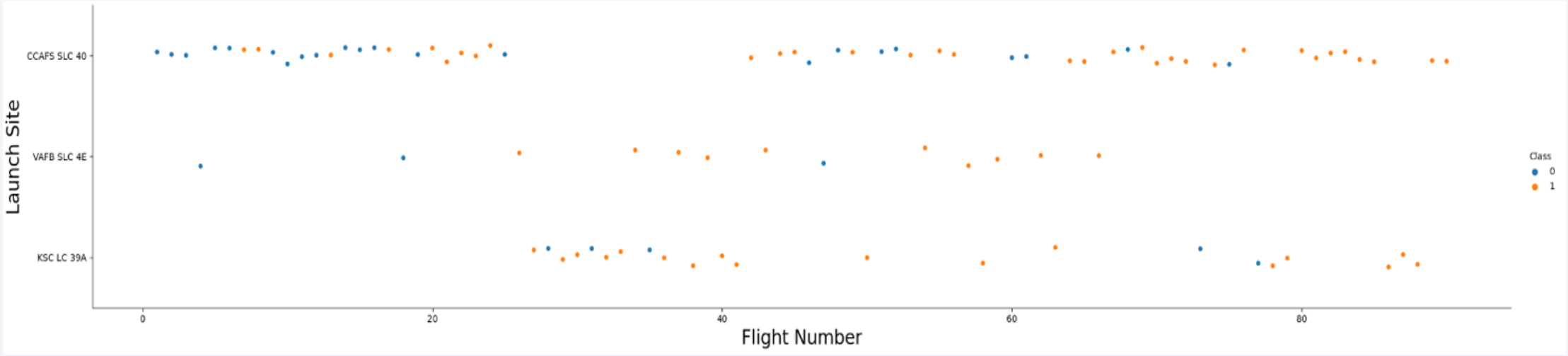
Results

- Insight from Interactive Analytics:
  Interactive visual analytics revealed that launch sites are typically located in safe areas—often near the sea—and are supported by strong logistical infrastructure.
  Additionally, most launches were concentrated along the East Coast launch sites.



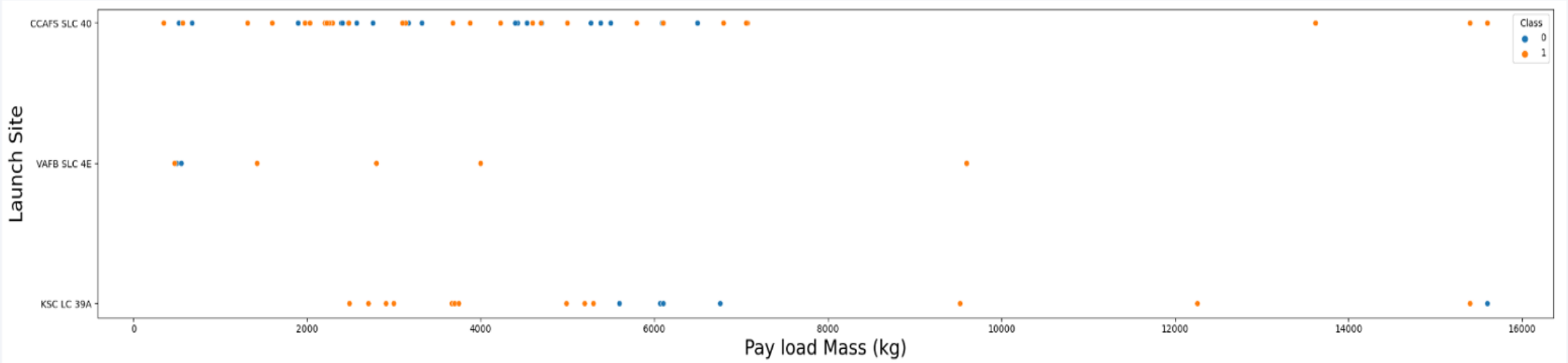

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



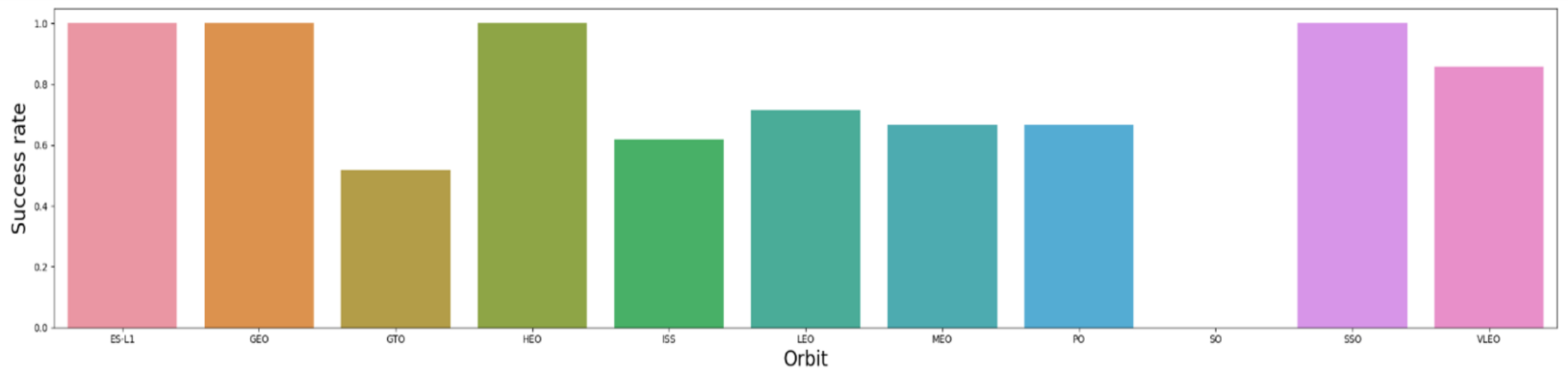
The CCAFS SLC 40 has no Flight number from around 25 to 40.
The KSC-LC 39A has no Flight number from 0 to around 20.

Payload vs. Launch Site



The VAFB-SLC is no rockets launched for heavy payload mass(greater than 10000).

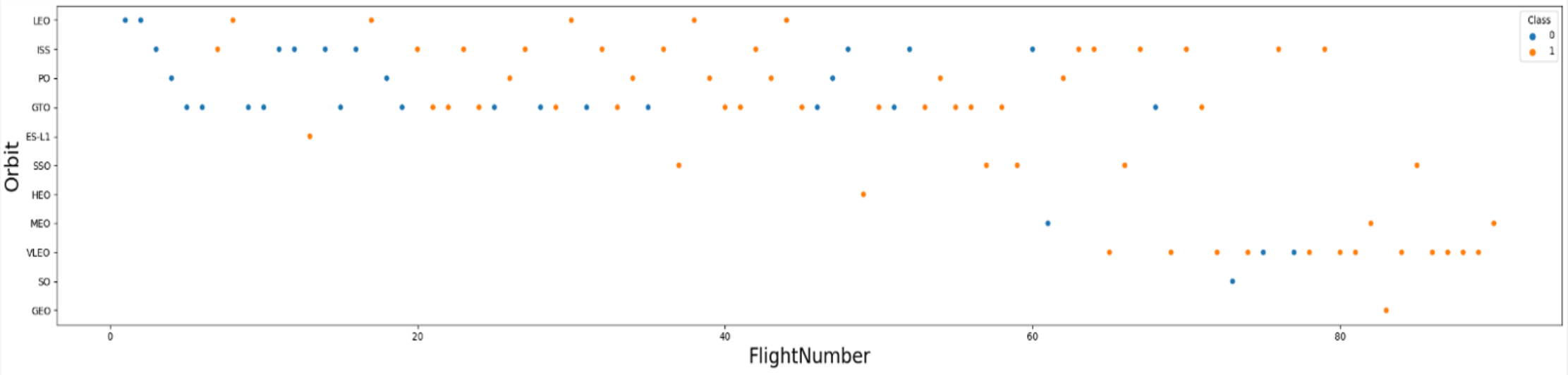
Success Rate vs. Orbit Type



The SO Orbit type has 0 success rate.

The ES-L1, GEO, HEO, and SSO have highest success rate.

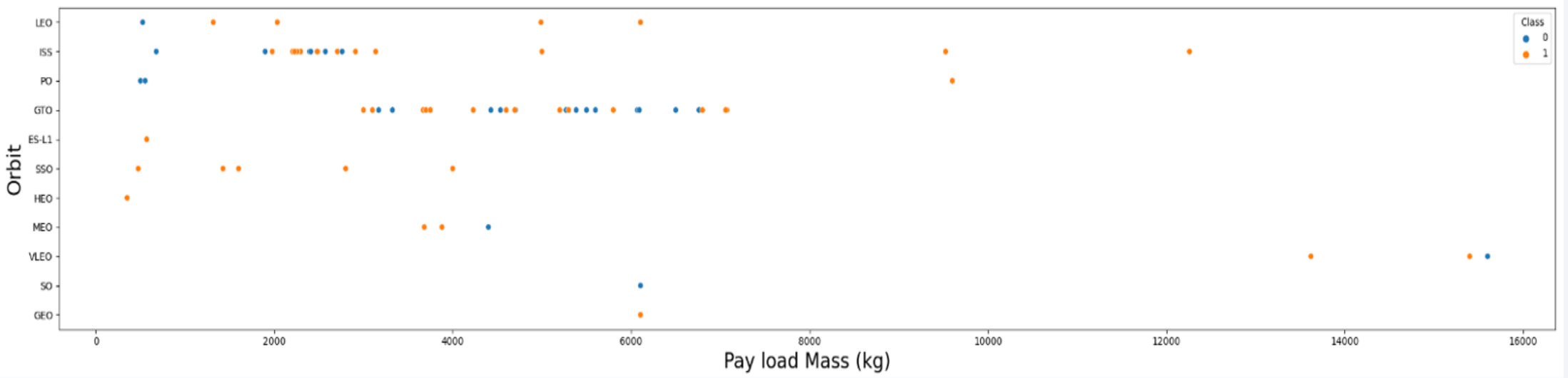
Flight Number vs. Orbit Type



The LEO orbit the Success appears related to the number of flights.

There seems to be no relationship between flight number when in GTO orbit.

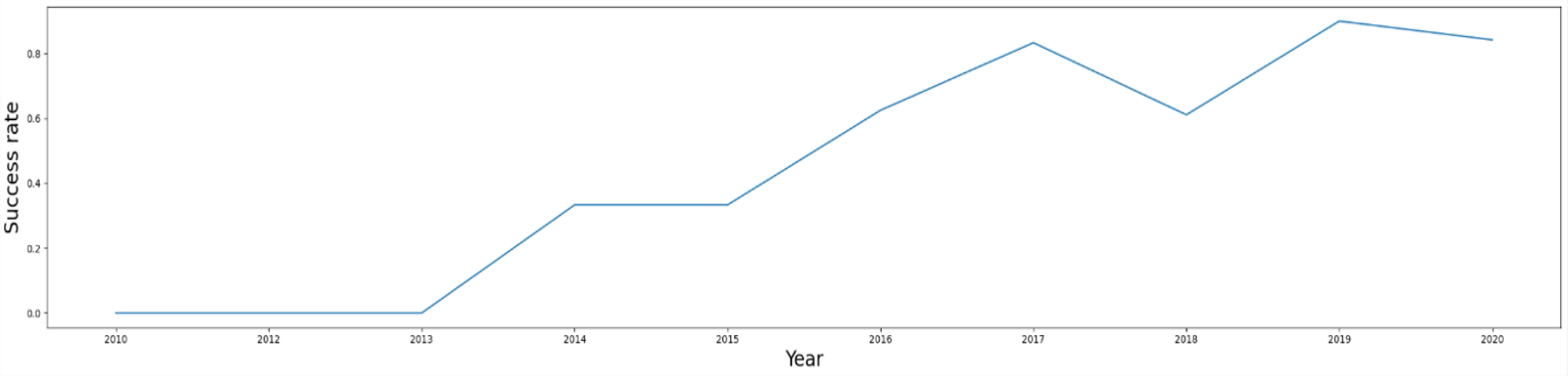
Payload vs. Orbit Type



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

GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission).

Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020

All Launch Site Names

- Launch Site Overview:
  Identified four distinct launch site names:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

Task 2

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

We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

Task 3

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

```
[29]: %sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

```
[29]: ,,,,,,,,,,,
```

SUM("PAYLOAD_MASS__KG_")
45596

The total payload carried by boosters from NASA stood at 45596 using the query above

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 was calculated at 2928.4

Task 4

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

```
[13]: %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

```
[13]: ,,,,,,,,,,,
```

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

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

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

MIN("Date")
2015-12-22

It was observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 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

We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total 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

We used wildcard like '%' to filter for **WHERE** Mission_Outcome was a success or a failure.

Boosters Carried Maximum Payload

Task 8

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
%sql SELECT DISTINCT "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

the names of the booster which have carried the maximum payload mass can be seen here.

the alternatively, a query can be added to show the payload mass in kg. When the query was run, all these Boosters seemed to carry 15600kg payload

2015 Launch Records

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

```
 * 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

We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

Rank Landing Outcomes Between 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 OutcomeCount \
FROM SPACEXTABLE \
WHERE "Date" >= '2010-06-04' AND "Date" <= '2017-03-20' \
GROUP BY "Landing_Outcome" \
ORDER BY OutcomeCount DESC;
```

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

Landing_Outcome	OutcomeCount
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

We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

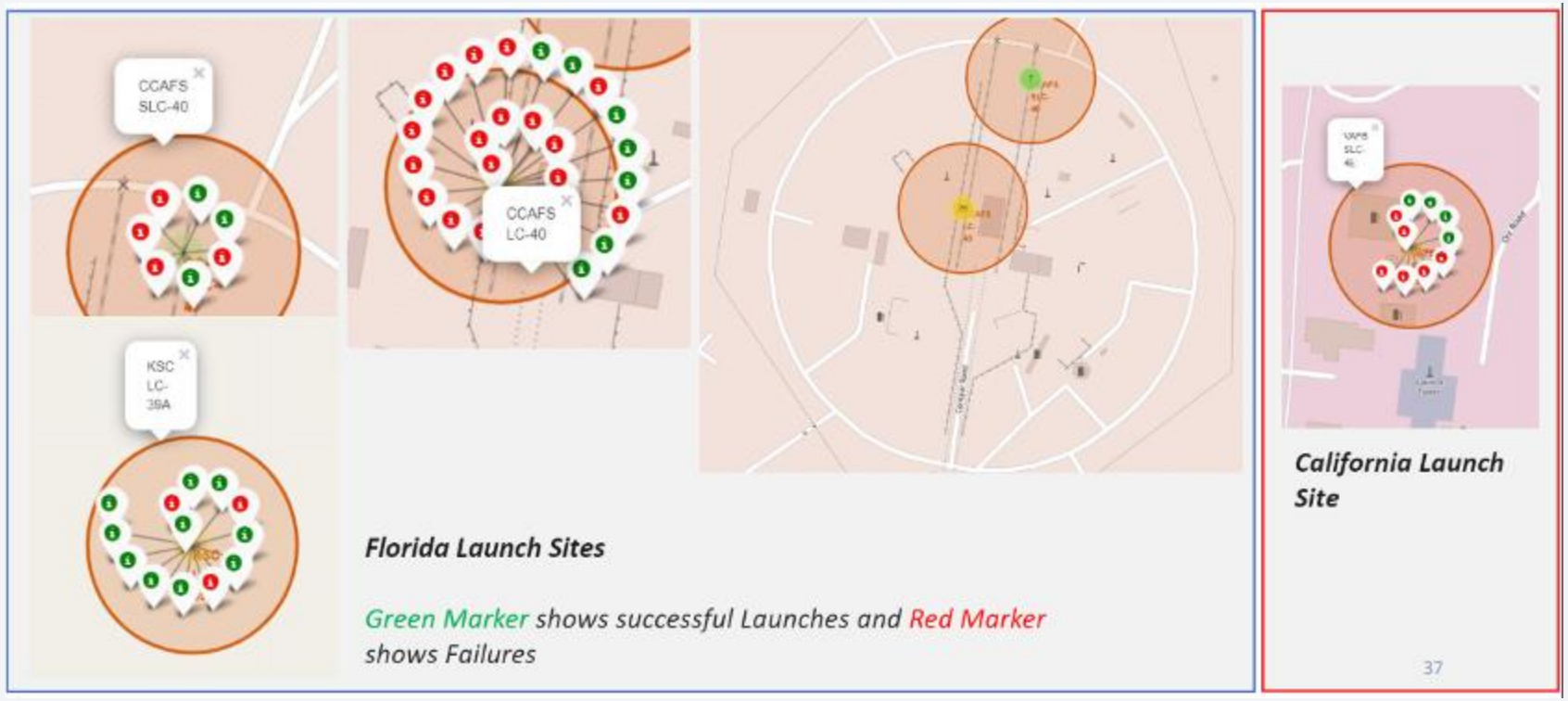
We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 3

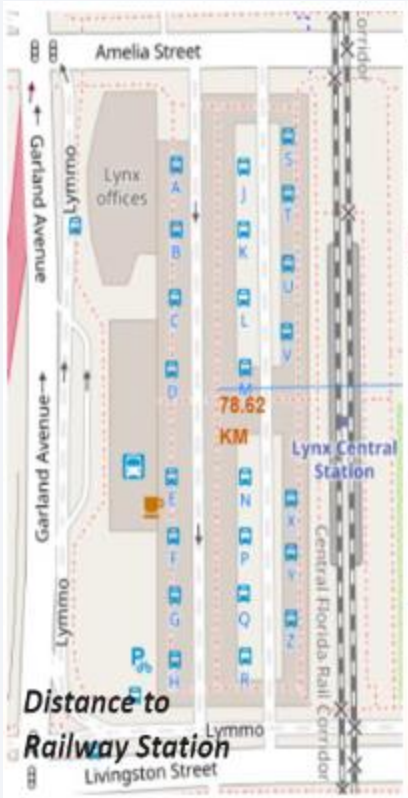
Launch Sites Proximities Analysis

Launch sites global map markers

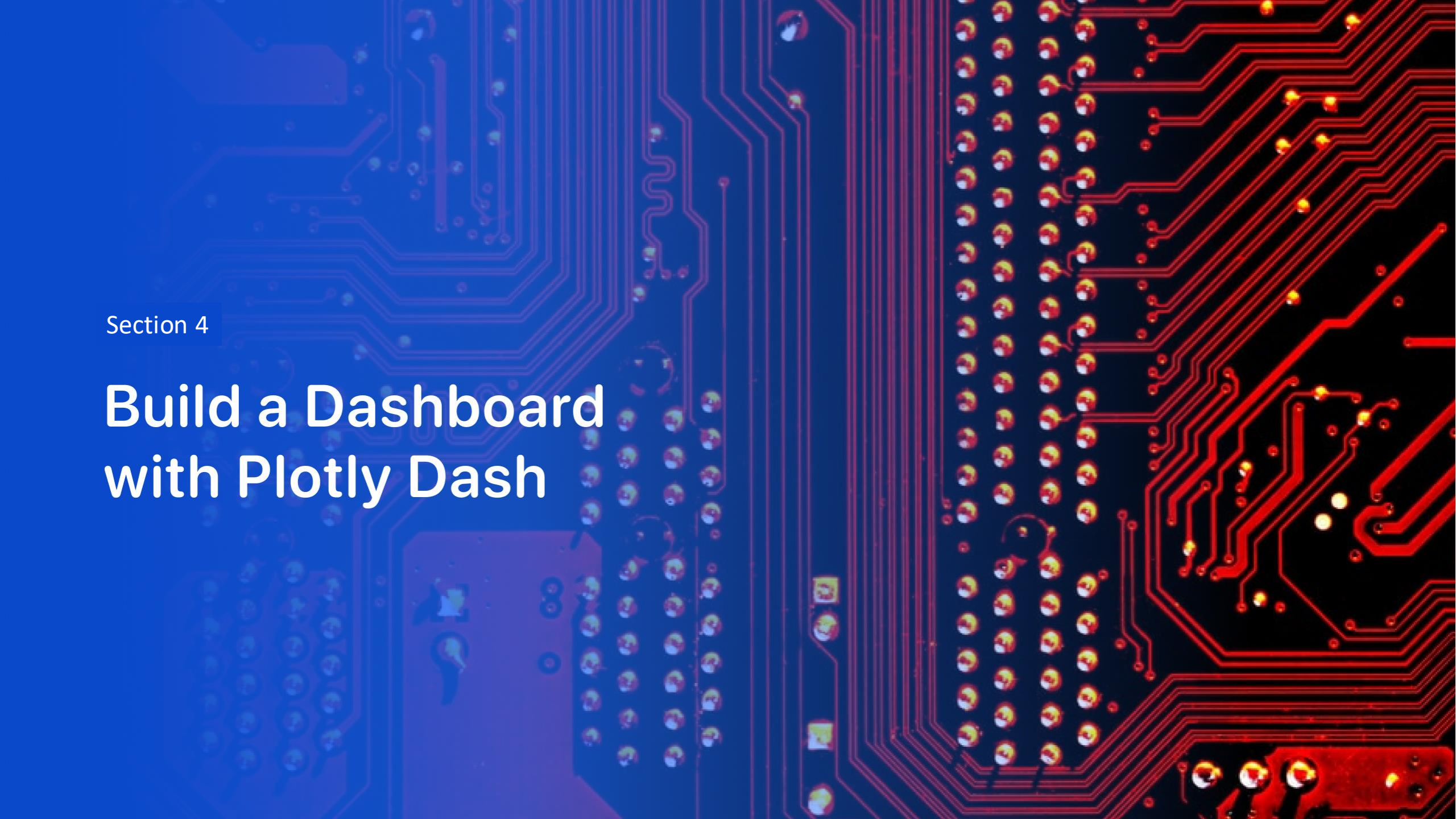
Couloured Markers showing Launch Sites



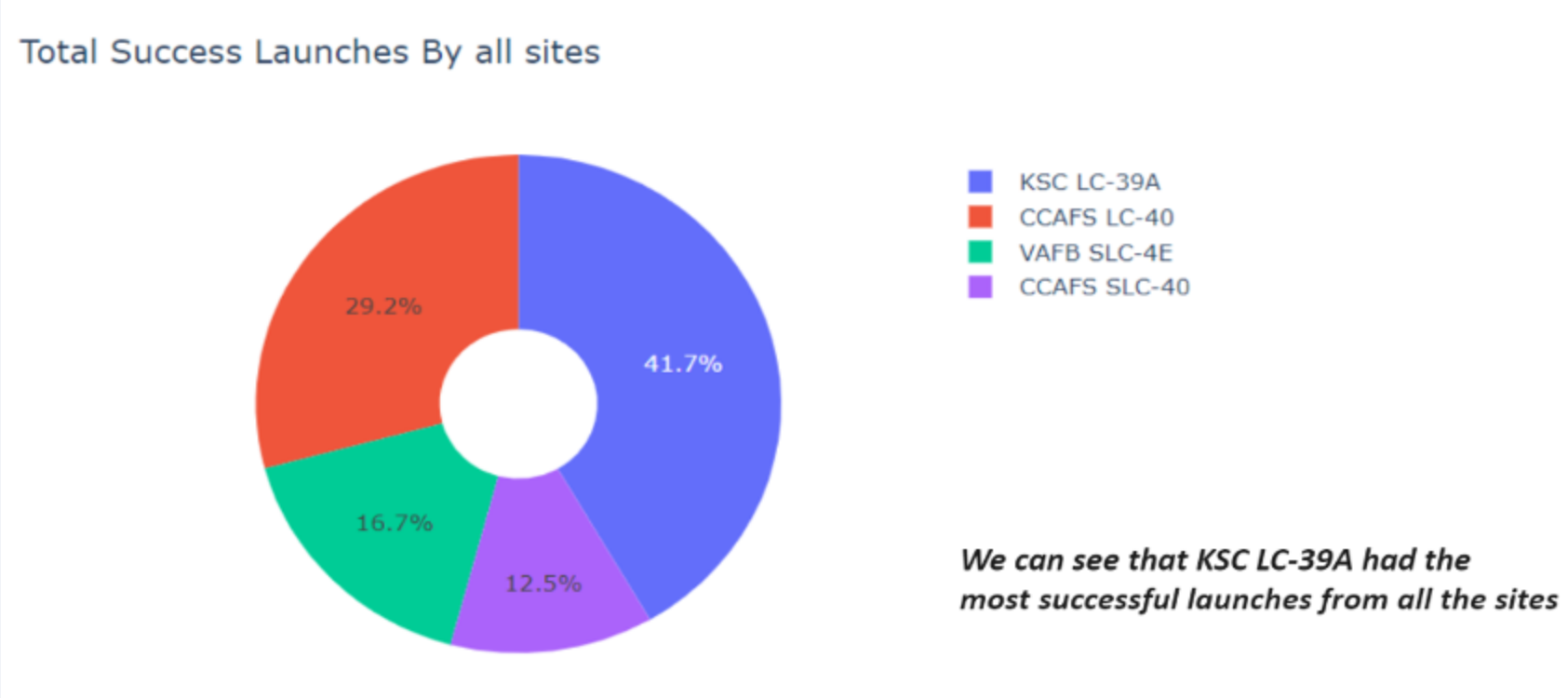
Launch Site distance to landmarks



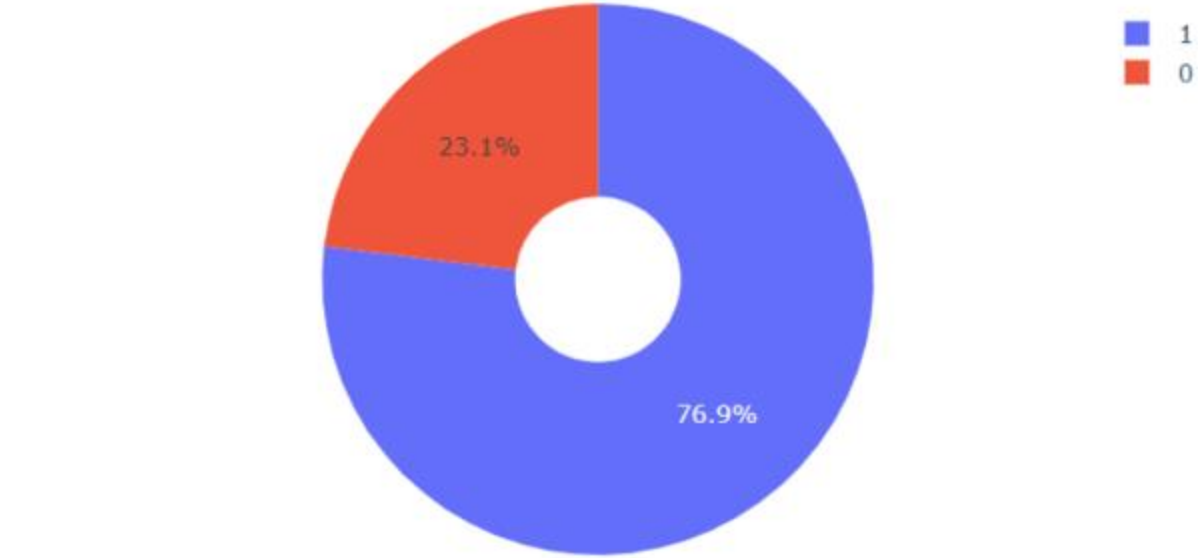
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

Success Percentage For Each Launch Site

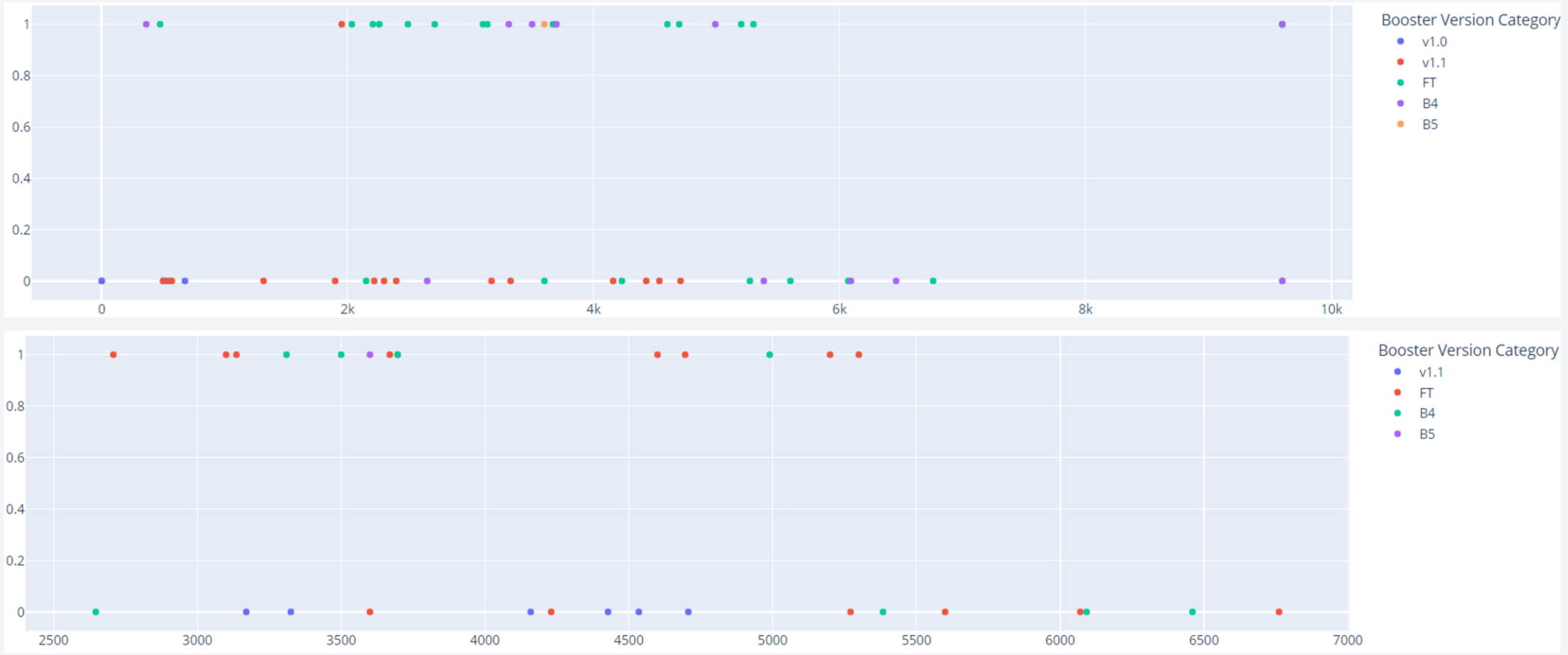


Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs. Launch Outcome



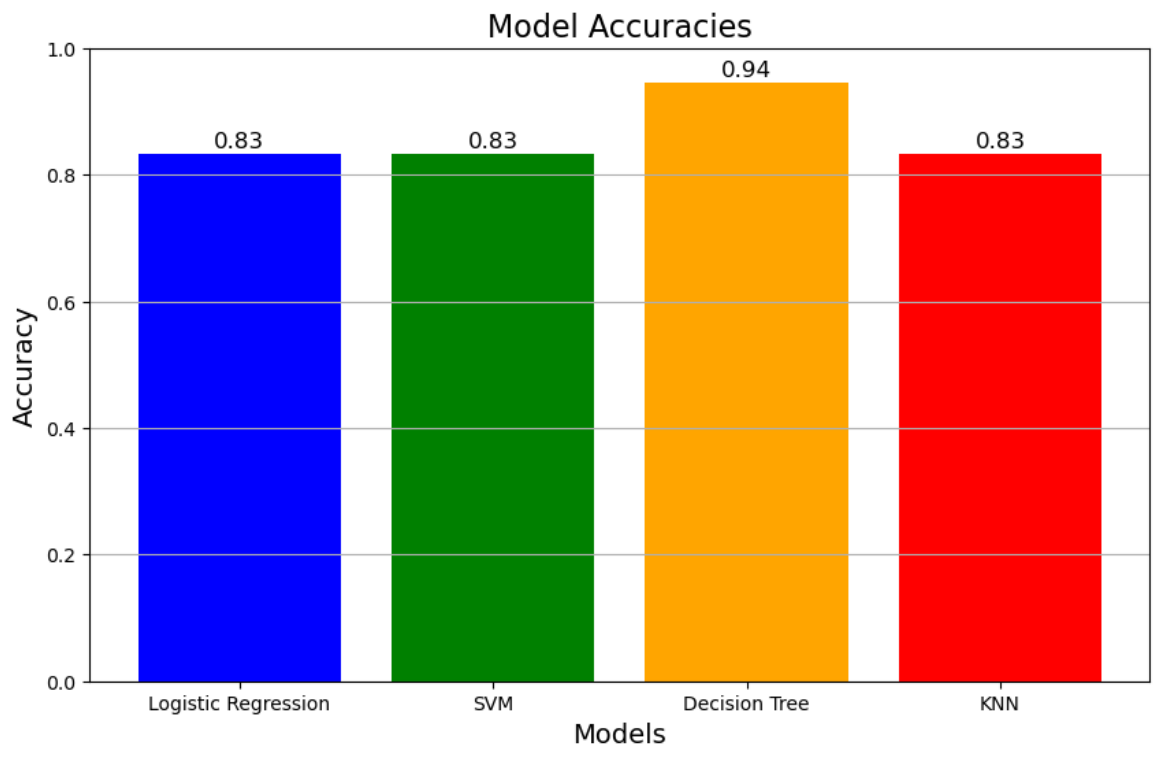
The B4 has success launches more than fail.

The v.1.0 is the only Booter Version has Payload more than around 7K.

Section 5

Predictive Analysis (Classification)

Classification Accuracy




```python
# Store accuracies in a dictionary for easy comparison
model_accuracies = {
    'Logistic Regression': test_accuracy,
    'SVM': test_accuracy_svm,
    'Decision Tree': test_accuracy_tree,
    'KNN': test_accuracy_knn
}
# Print the accuracies of all models
for model, accuracy in model_accuracies.items():
    print(f"{model} Accuracy: {accuracy:.4f}")
# Find the best performing model
best_model = max(model_accuracies, key=model_accuracies.get)
best_accuracy = model_accuracies[best_model]
print(f"\nBest Performing Model: {best_model} with Accuracy: {best_accuracy:.4f}")
```

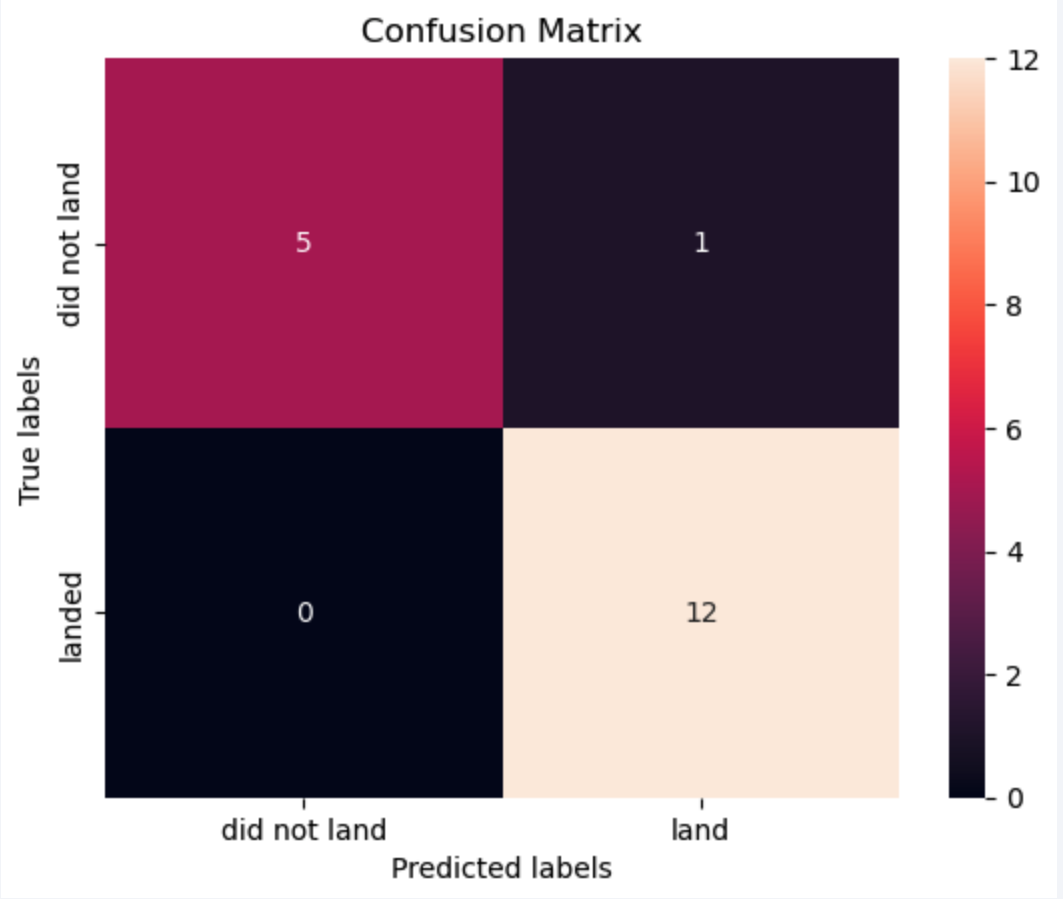
```
Logistic Regression Accuracy: 0.8333
SVM Accuracy: 0.8333
Decision Tree Accuracy: 0.9444
KNN Accuracy: 0.8333

Best Performing Model: Decision Tree with Accuracy: 0.9444
```

- From the bar chart and the python code, it can be seen that Decision Tree has the highest accuracy at 0.9444

Confusion Matrix

- The accuracy of the model 'Decision Tree' can be seen in the Confusion matrix. Only 1 false positive was detected.



Conclusions

It can be concluded that:

- A higher volume of flights at a launch site corresponds to an increased success rate.
- The launch success rate experienced a rise from 2013 to 2020.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO demonstrated the highest success rates.
- KSC LC-39A recorded the most successful launches compared to other sites.
- The Decision Tree classifier was identified as the most effective machine learning algorithm for this task.

Appendix

- As the necessary code and charts have already been mentioned here, no appendix has been shared.

Thank you!