IBM Developer
SKILLS NETWORK

SpaceX Falcon 9 First Stage Landing Prediction: Unveiling the Secrets of Rocket Launch Success

QAZI FABIA HOQ
31ST JANUARY 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project Background

Space X advertises Falcon 9 rocket launches at a cost of $62 million, a significant reduction compared to other providers, who charge upwards of $165 million per launch. A key factor in this cost savings is SpaceX's ability to reuse the first stage of the rocket. By predicting whether the first stage will successfully land, it becomes possible to estimate the overall cost of a launch. This information could be valuable for potential competitors aiming to bid against SpaceX for future rocket launches. The goal of this project is to build a machine learning pipeline to predict the success of the first stage landing.

- Business Problems

  - What are the key factors that influence the success of a rocket landing?
  - How do different features interact to affect the likelihood of a successful landing?
  - What operational conditions must be met to ensure the success of the landing program?

Section 1

Methodology

Methodology

- Data Collection Approach: Data was obtained through the SpaceX API and web scraping from Wikipedia.
- Data Processing: Cleaning and structuring of the dataset were conducted to ensure consistency.
- Feature Encoding: Categorical variables were transformed using one-hot encoding.
- Exploratory Data Analysis (EDA): Visualizations and SQL queries were used to analyze trends and patterns.
- Interactive Visualizations: Folium and Plotly Dash were utilized for dynamic data exploration.
- Predictive Modeling: Classification algorithms were applied to assess landing success.
- Model Optimization: Techniques for building, fine-tuning, and evaluating classification models were explored.

Data Collection

- The data was collected using various methods
  - Data collection was conducted using a GET request to the SpaceX API.
  - The response content was decoded into JSON format using the .json() function and converted into a Pandas DataFrame with .json_normalize().
  - The data was then cleaned, missing values were identified, and necessary imputations were performed.
  - Additionally, web scraping was carried out using BeautifulSoup to extract Falcon 9 launch records from Wikipedia.
  - The extracted launch records, originally in an HTML table format, were parsed and transformed into a Pandas DataFrame for further analysis.

Data Collection – SpaceX API

- A GET request to the SpaceX API was used to collect data, which was then cleaned, processed, and formatted through basic data wrangling.
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/data_collection_api.ipynb

Request and parse the SpaceX launch data using the GET request

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/
```

In [10]:
```
response=requests.get(static_json_url)
```

In [11]:
```
response.status_code
```

Out[11]: 200

The response content is decoded as JSON using the .json() method, and then it is converted into a Pandas dataframe using the .json_normalize() method.

In [12]:
```
# Use json_normalize meethod to convert the json result into a dataframe
# Decode the JSON response
data = response.json()

# Normalize the JSON response into a Pandas DataFrame
df = pd.json_normalize(data)

# Display the first few rows of the DataFrame
print(df.head())
```

```
      static_fire_date_utc  static_fire_date_unix    tbd    net  window  \
0  2006-03-17T00:00:00.000Z           1.142554e+09  False  False     0.0
1                      None                    NaN  False  False     0.0
2                      None                    NaN  False  False     0.0
3  2008-09-20T00:00:00.000Z           1.221869e+09  False  False     0.0
4                      None                    NaN  False  False     0.0
```

Data Collection – Web Scraping

- Web scraping was applied to extract Falcon 9 launch records using BeautifulSoup.
- The table was parsed and converted into a Pandas DataFrame.
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/web_scrapping.ipynb

Request the Falcon9 Launch Wiki page from its URL

The HTTP GET method will be performed to request the Falcon 9 Launch HTML page, retrieving it as an HTTP response.

In [5]:
```python
# Request the Falcon 9 Launch Wikipedia page using requests.get()
response = requests.get(static_url)

# Check the status code of the request to ensure it was successful
if response.status_code == 200:
    print("Request successful!")
else:
    print("Request failed with status code:", response.status_code)
```

```
Request successful!
```

Create a `BeautifulSoup` object from the HTML `response`

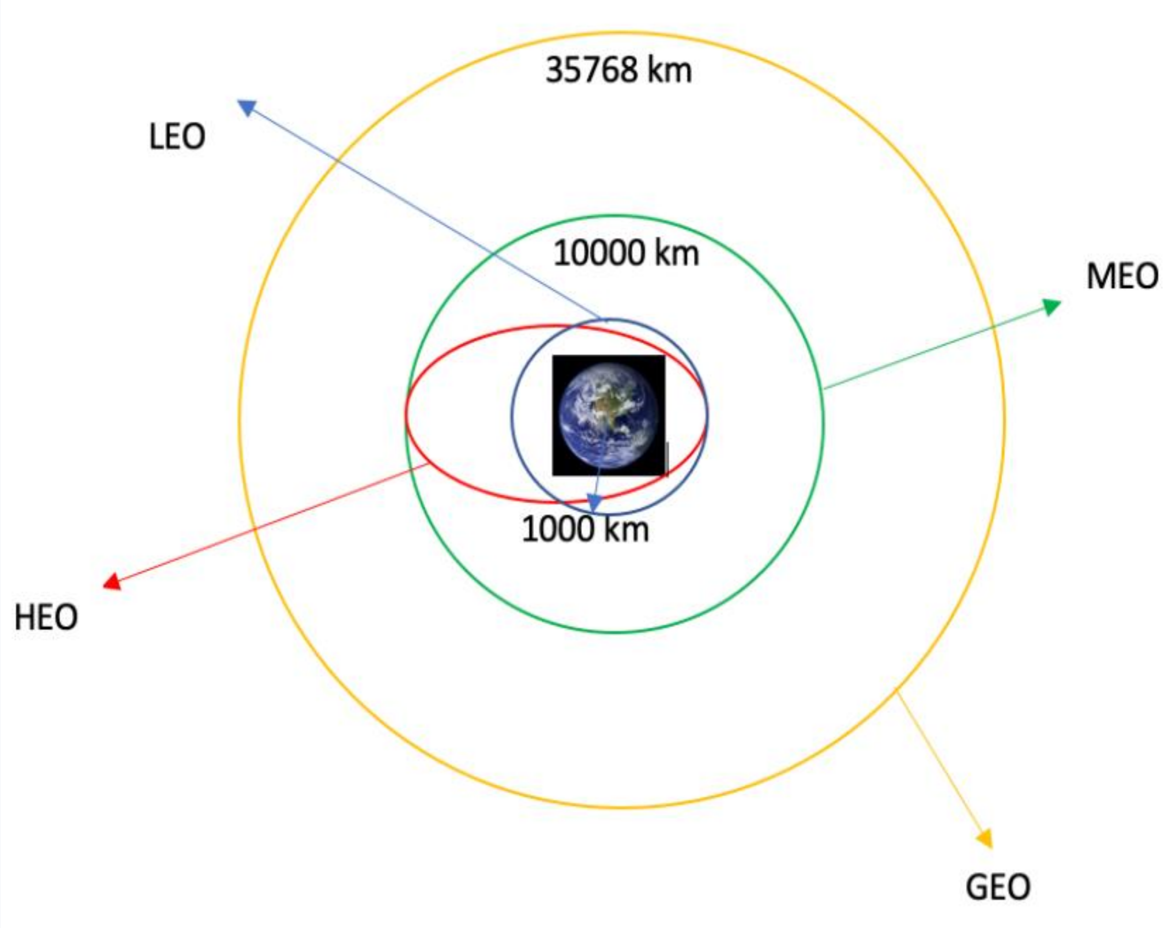
In [6]:
```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')

# Print the first 500 characters of the parsed HTML to verify
print(soup.prettify()[:500])
```

```
<!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vector-feature-night-mode-enabled skin-theme-clientpref-day vect
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```python
# Print the page title to verify if the BeautifulSoup object was created properly
print(soup.title)
```

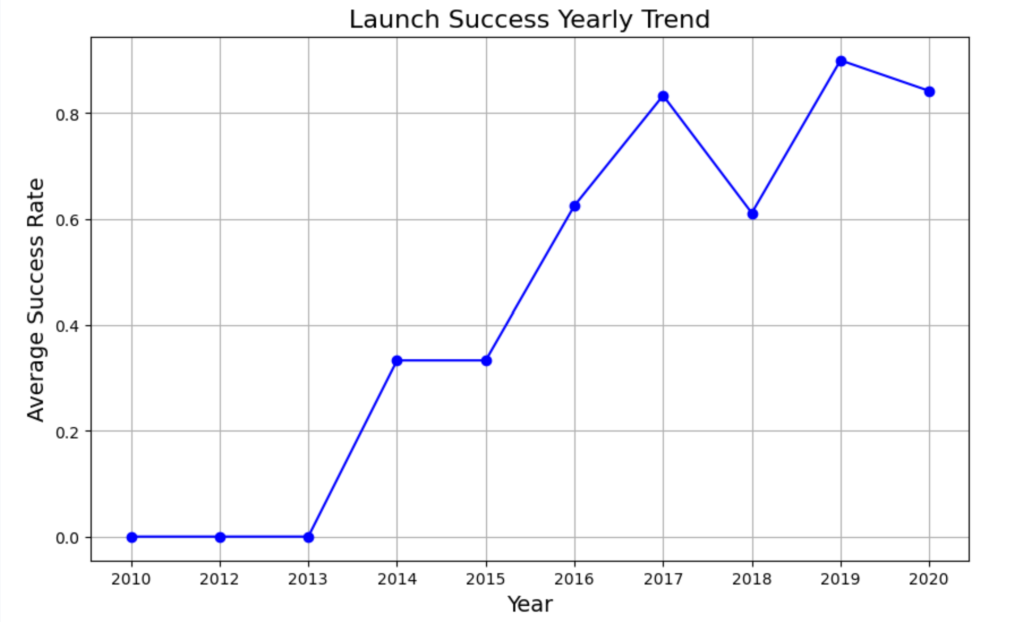
Data Wrangling

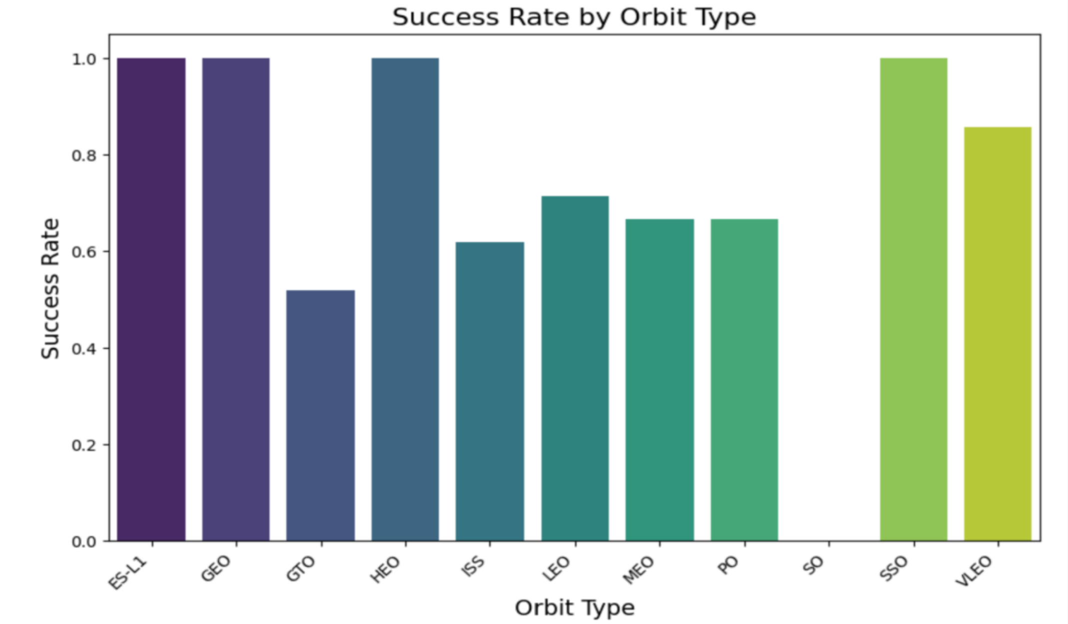


- Conducted exploratory data analysis and determined training labels.
- Calculated the number of launches at each site using the .value_counts() method on the LaunchSite column.
- Analyzed the number and occurrence of each orbit using .value_counts() on the Orbit column.
- Evaluated mission outcomes by applying .value_counts() to the Outcome column, assigning results to landing_outcomes.
- Created landing outcome labels from the Outcome column and exported the results to a CSV file.
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/wrangling.ipynb

EDA with Data Visualization

- Key relationships were analyzed through visualizations. The success rate of each orbit type was examined, and yearly launch success trends were plotted. Scatter plots explored the impact of payload mass on orbit type and launch site. Additionally, the relationship between flight number and launch site was visualized.





- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/data_analysis_visualizations.ipynb

EDA with SQL

- Loaded the SpaceX dataset into an SQL database using SQLAlchemy within Jupyter Notebook..
- Applied EDA with SQL to extract insights, including:
  - Unique launch site names.
  - Total payload mass carried by NASA (CRS) missions.
  - Average payload mass for booster version F9 v1.1.
  - Total count of successful and failed missions.
  - Failed landings on drone ships, including booster versions and launch sites.
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/EDA_SQL.ipynb

Geospatial Analysis with Folium

- Marked all launch sites and added map objects such as markers, circles, and lines to indicate launch success or failure.
- Assigned launch outcomes to binary classes: 0 for failure and 1 for success.
- Used color-labeled marker clusters to identify launch sites with relatively high success rates.
- Calculated distances between launch sites and nearby features, addressing questions such as:
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites maintain a certain distance from cities?
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/follium_map_launchsite.ipynb

Interactive Dashboard with Plotly Dash

- Built an interactive dashboard using Plotly Dash.
- Plotted pie charts to show the total number of launches by each site.
- Created scatter plots to visualize the relationship between outcome and payload mass (kg) for different booster versions.
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/spacex_dash_app.py

Predictive Analysis

- Loaded the data using NumPy and Pandas, transformed it, and split it into training and testing sets.
- Built various machine learning models and tuned hyperparameters using GridSearchCV.
- Used accuracy as the evaluation metric, improving the model through feature engineering and algorithm tuning.
- Identified the best-performing classification model.
- The link to the notebook is https://github.com/qazifabiahoq/Space-X_Landing-Prediction_CAPSTONE_PROJECT_IBM/blob/main/notebooks_dashapp/Machine%20Learning%20Prediction.ipynb

Results

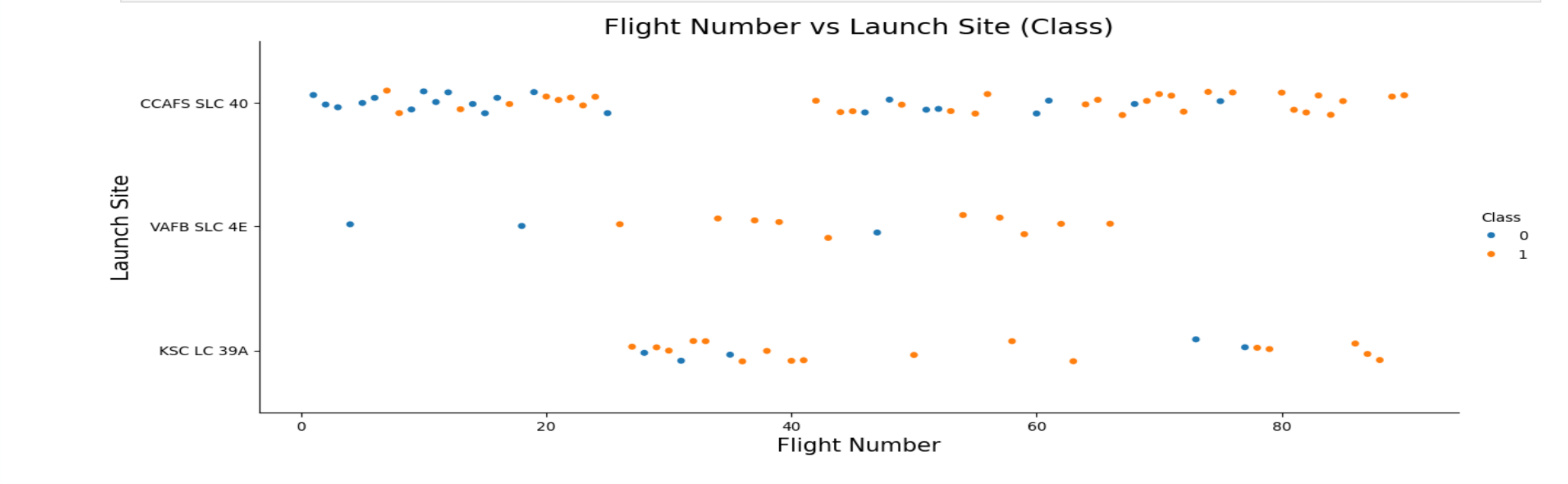
- Presented exploratory data analysis (EDA) results.
- Included interactive analytics demo with screenshots.
- Showcased predictive analysis results.

Section 2

Insights drawn from EDA

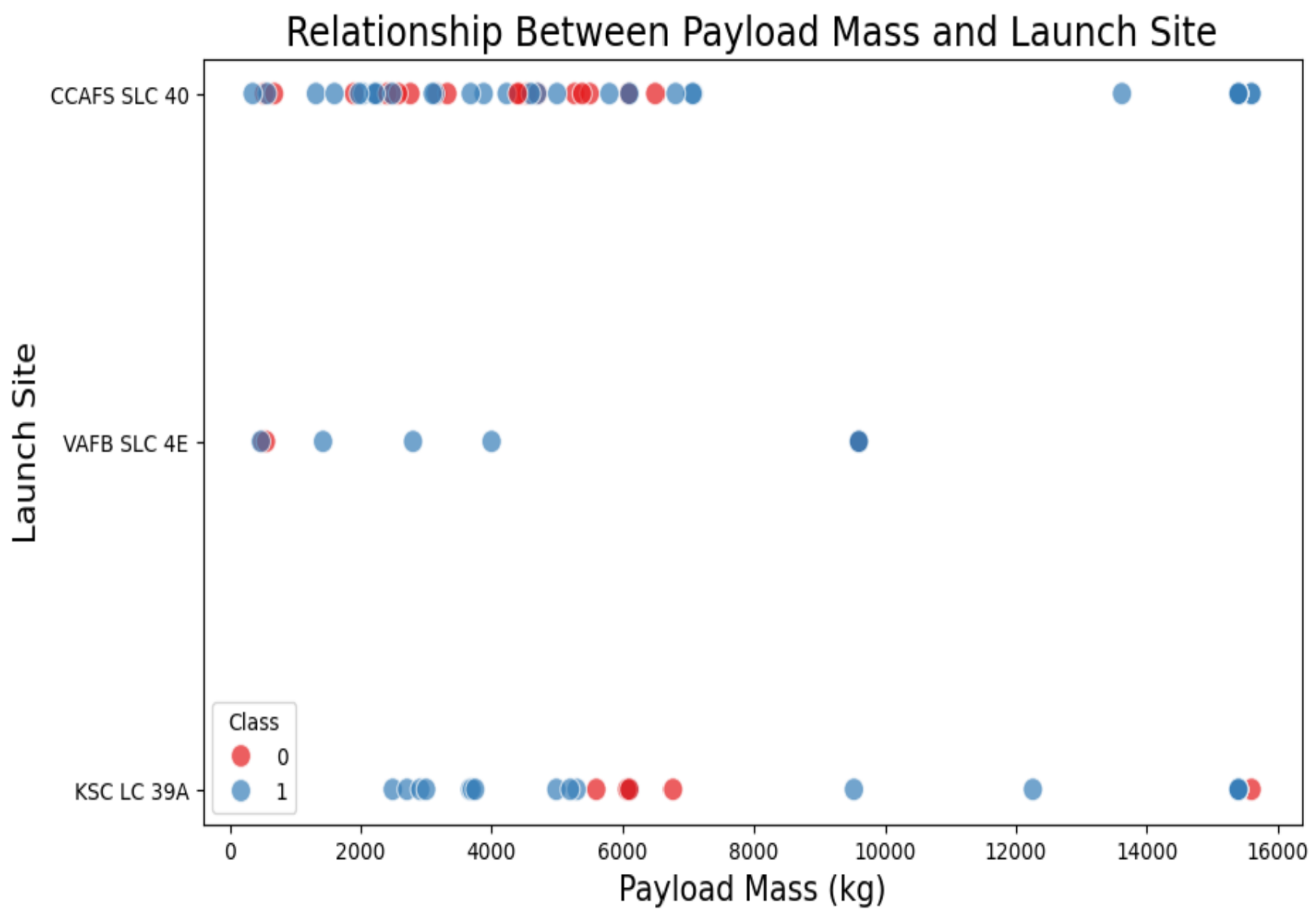
Flight Number vs. Launch Site

- CCAFS SLC 40 is the most active site, with a high density of launches and a mix of successful and unsuccessful landings. VAFB SLC 4E and KSC LC 39A have fewer launches, with KSC LC 39A showing a lower frequency beyond flight number 40.
- The analysis shows that as the number of launches increases at a site, the success rate also improves, with more Class 1 (successful) landings observed as flight numbers rise.



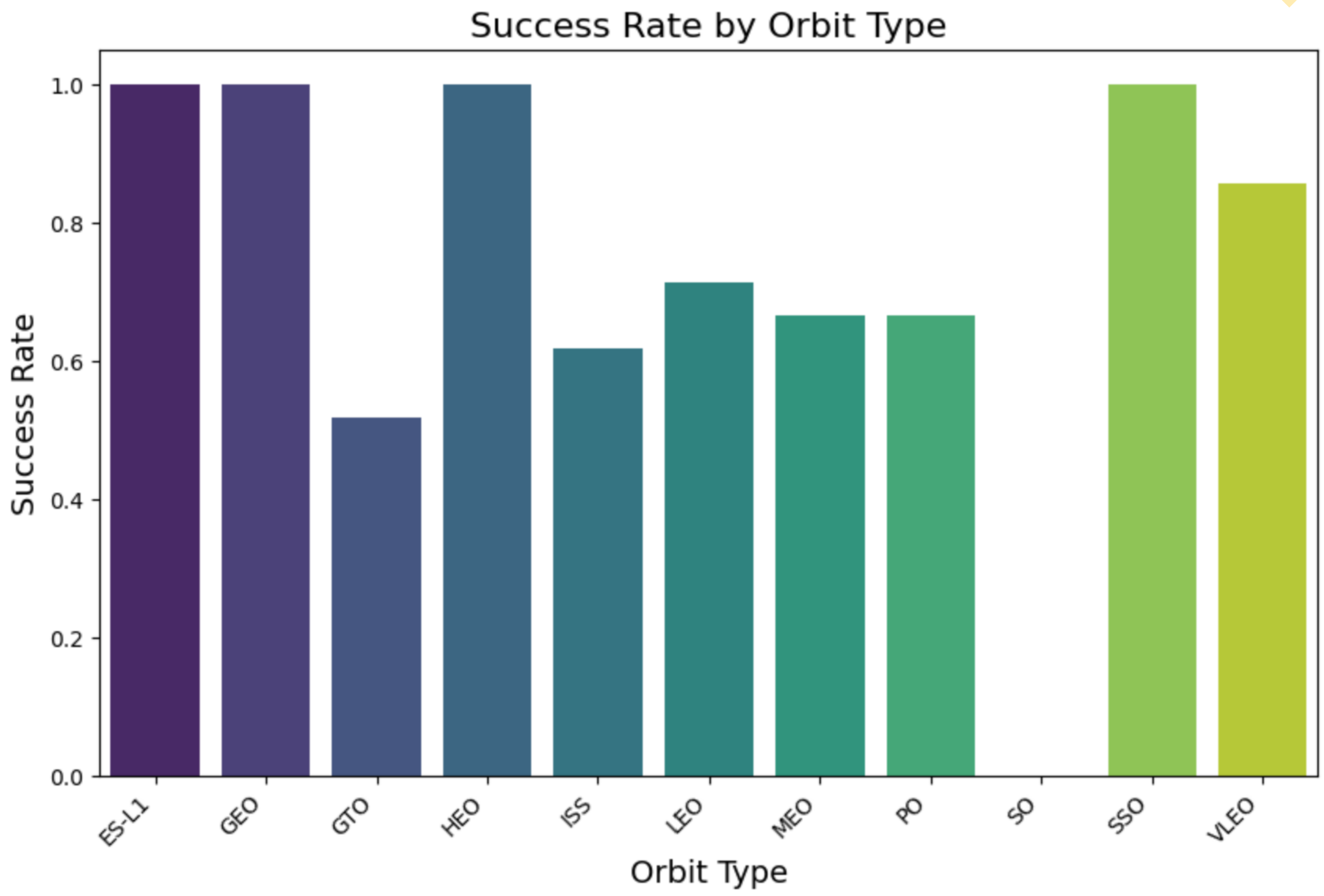
Payload vs. Launch Site

- VAFB SLC 4E has relatively fewer launches compared to other sites and does not support payloads greater than 10,000 kg, suggesting it may be specialized for lighter payloads.
- CCAFS SLC 40 and KSC LC 39A handle a broader range of payloads, including those above 10,000 kg.
- The analysis shows that as payload mass increases, the success rate for launches at these sites also tends to be higher.



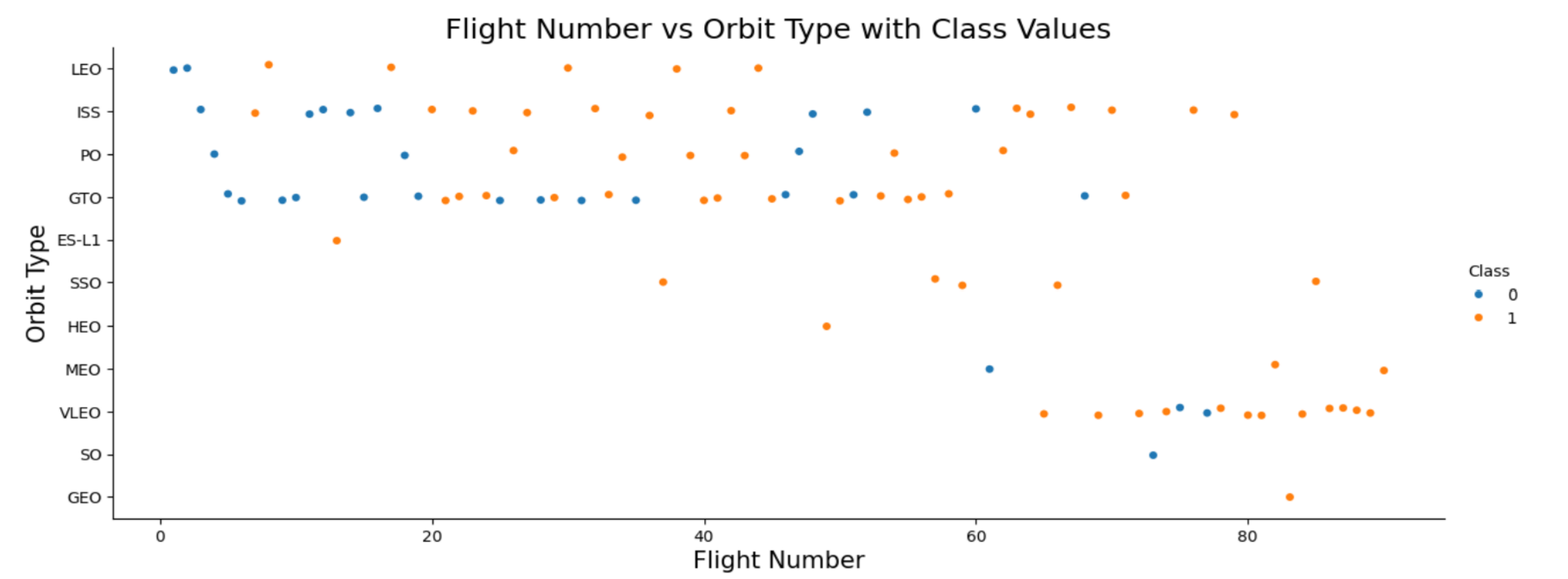
Success Rate vs. Orbit Type

- Highest Success Rate (1.0): ES-L1, GEO, and HEO orbits show perfect success with a 100% success rate.
- Near-Perfect Success Rate (~1.0): SSO orbit has nearly perfect success, followed by VLEO with a high success rate around 0.9.
- Moderate Success Rate: LEO, MEO, and PO orbits have moderate success rates (0.7 to 0.6).
- Lowest Success Rate (0.5): GTO orbit has the lowest success rate at 0.5, indicating a higher failure rate.
- No Success: SO orbit has a success rate of 0, with no successful missions.



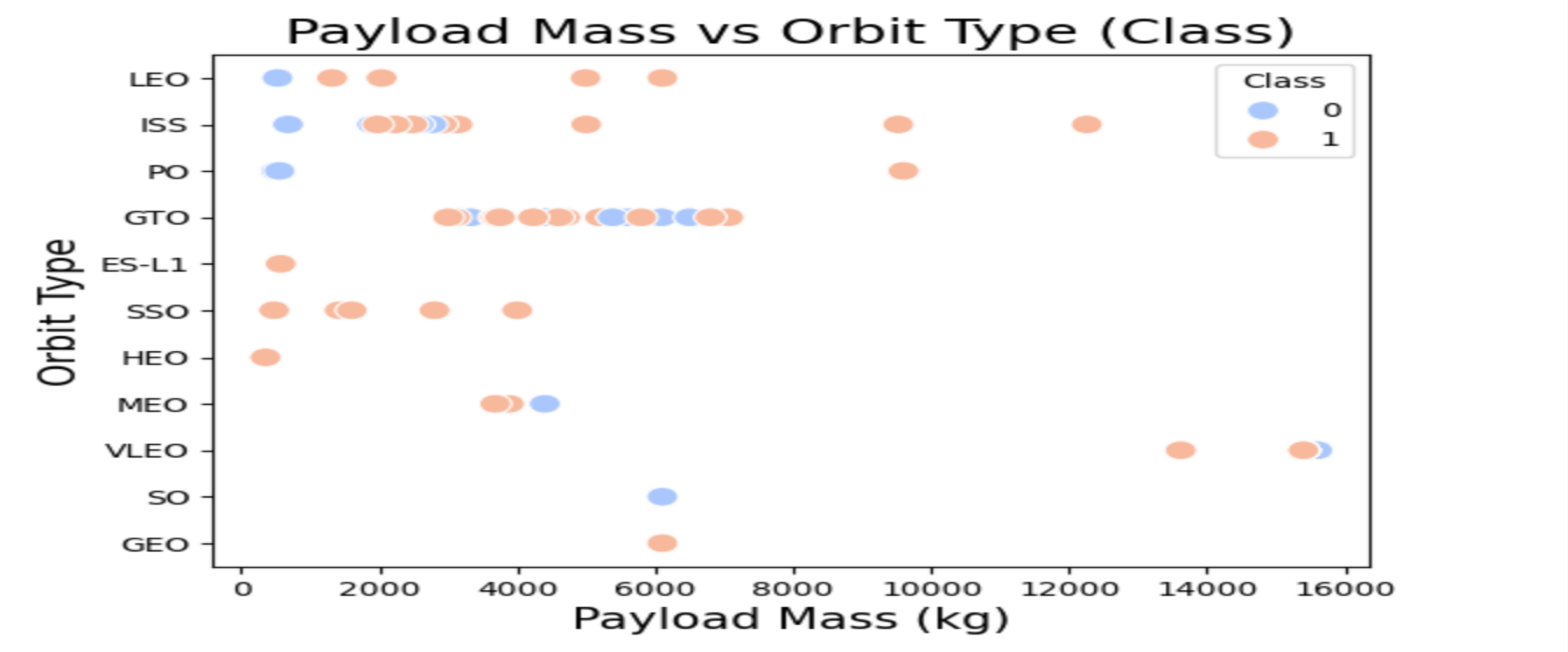
Flight Number vs. Orbit Type

- The plot below illustrates the relationship between flight number and orbit type. It reveals that in the LEO orbit, success tends to increase with the number of flights, while in the GTO orbit, no such relationship is observed between flight number and success rate.



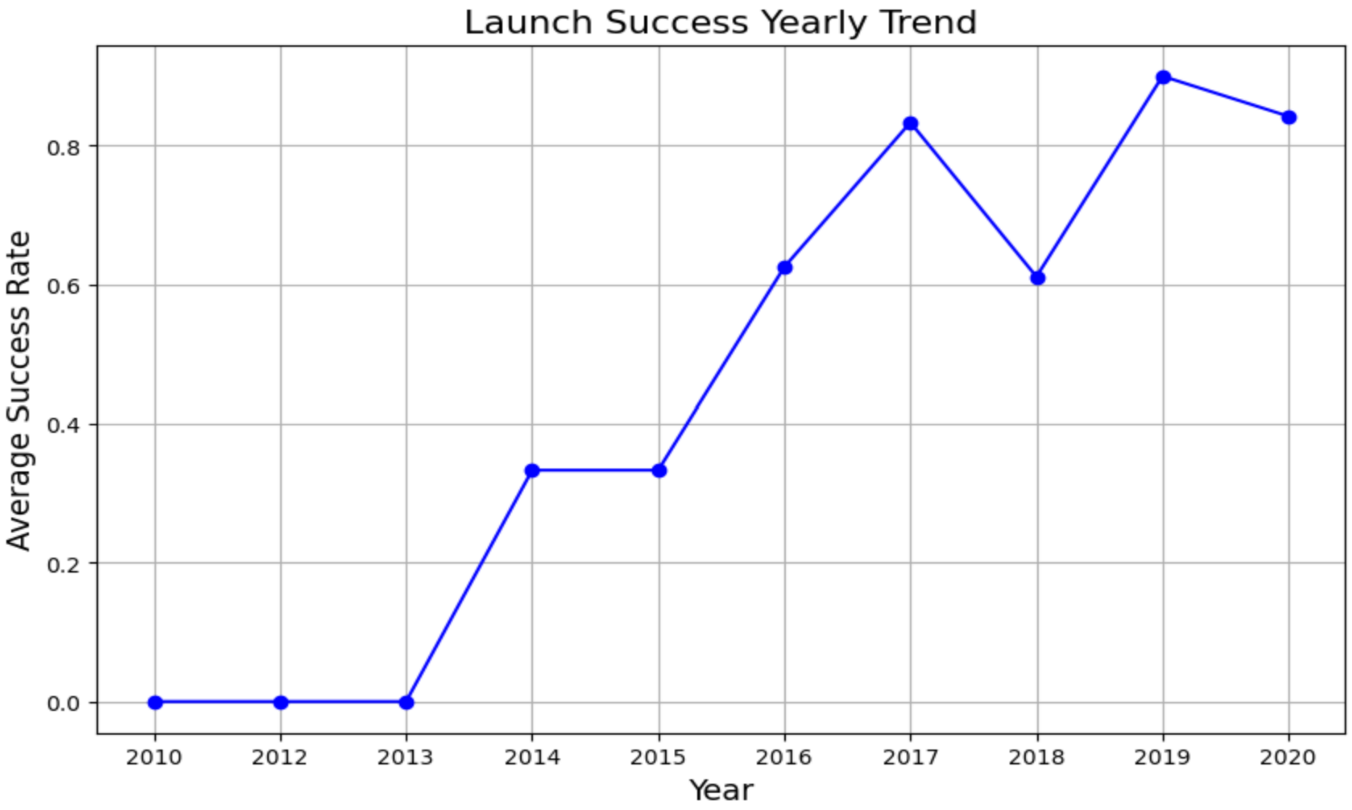
Payload vs. Orbit Type

- For Polar, LEO, and ISS orbits, heavier payloads tend to have a higher success rate in landings.
- In GTO, there is no clear relationship between payload mass and landing success, as both successful and unsuccessful landings are observed across a wide range of payloads.



Launch Success Yearly Trend

- The success rate of SpaceX launches has steadily increased since 2013, reflecting continuous improvements in technology and operations, especially between 2013 and 2020.



All Launch Site Names

- The DISTINCT SQL clause was used to display only the unique launch sites from the SpaceX data.

The unique launch sites in the space mission will be displayed.

In [11]:

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

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

Five records where the launch sites begin with the string 'CCA' will be displayed.

```
%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_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
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 (parach
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

- The query above was used to display 5 records where launch sites begin with CCA.

Total Payload Mass

- The total payload mass carried by boosters launched by NASA (CRS), totaling 48,213 kg, was calculated using the query below.

The total payload mass carried by boosters launched by NASA (CRS) will be displayed.

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass \
FROM SPACEXTABLE \
WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

Total_Payload_Mass
48213

Average Payload Mass by F9 v1.1

- The average payload mass carried by the booster version F9 v1.1, calculated to be 2,928.4 kg, was determined.

The average payload mass carried by the booster version F9 v1.1 will be displayed.

```
# Execute SQL query to calculate the average payload mass for booster version F9 v1.1
result = %sql SELECT AVG("PAYLOAD_MASS__KG_") AS "Average_Payload_Mass" FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v
result
```

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

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

- The first successful landing on a ground pad was on 22nd July 2018.

The date when the first successful landing outcome on a ground pad was achieved will be listed.

```
# Execute SQL query to find the earliest date for a successful landing outcome on a ground pad
result = %sql SELECT MIN("Date") AS "First_Successful_Landing_Date" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success'
result
```

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

First_Successful_Landing_Date
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The names of the boosters that have successfully landed on a drone ship and have a payload mass greater than 4000 but less than 6000 will be listed.

```
result = %sql SELECT "Booster_Version" \
FROM SPACEXTABLE \
WHERE "Landing_Outcome" = 'Success (drone ship)' \
AND "Payload_Mass__KG_" > 4000 \
AND "Payload_Mass__KG_" < 6000;

# Display the result
result
```

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

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

- The WHERE clause was used to filter for boosters that successfully landed on a drone ship, with an additional AND condition to select successful landings where the payload mass was greater than 4,000 kg but less than 6,000 kg.

Total Number of Successful and Failed Mission Outcomes

The total number of successful and failed mission outcomes will be listed.

```
result = %sql SELECT "Landing_Outcome", COUNT(*) AS "Total_Missions" \
FROM SPACEXTABLE \
GROUP BY "Landing_Outcome";
result
```

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

Landing_Outcome	Total_Missions
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

- The total number of successful and failed mission outcomes was calculated by using the wildcard '%' to filter for different Landing_Outcome categories, such as controlled ocean landings, drone ship landings, and ground pad landings. This allowed for grouping and counting the outcomes based on success or failure.

Boosters Carrying The Maximum Payload

- The booster that carried the maximum payload was determined using a subquery in the WHERE clause along with the MAX() function.

The names of the booster versions that have carried the maximum payload mass will be listed using a subquery.

```
result = %sql SELECT "Booster_Version" \
FROM SPACEXTABLE \
WHERE "Payload_Mass__KG_" = (SELECT MAX("Payload_Mass__KG_") FROM SPACEXTABLE);
result
```

```
 * 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

- A combination of the WHERE clause, LIKE, AND, and BETWEEN conditions was used to filter for failed landing outcomes on a drone ship, along with their booster versions and launch site names for the year 2015.

The records will display the month names, failure landing outcomes on the drone ship, booster versions, and launch sites for the months in the year 2015.

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

# Display the result
result
```

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

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

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

The count of landing outcomes (such as Failure on the drone ship or Success on the ground pad) between the dates 2010-06-04 and 2017-03-20 will be ranked in descending order.

```
result = %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;

# Display the result
result
```

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

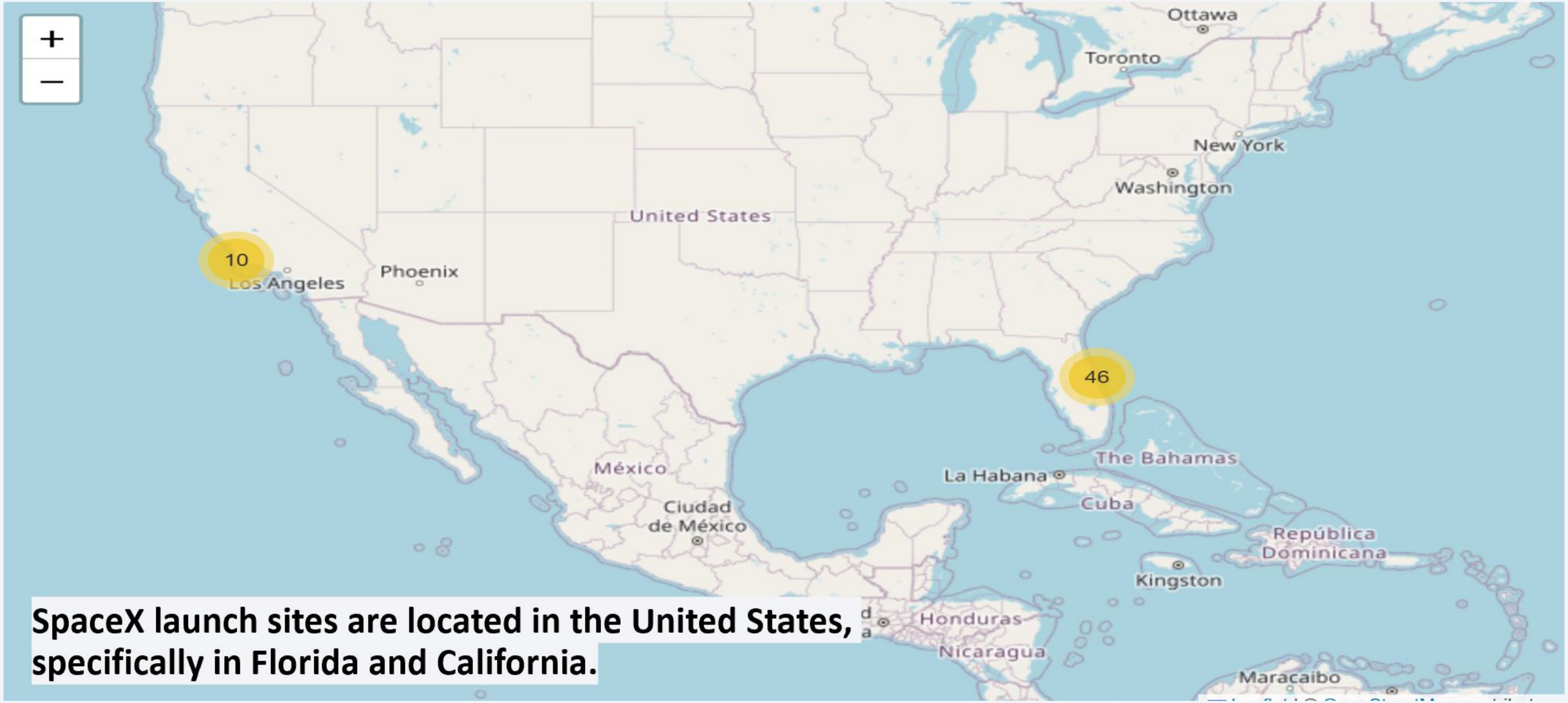
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- We selected the COUNT of landing outcomes (such as Failure on the drone ship or Success on the ground pad) from the data, using the WHERE clause to filter for landing outcomes between the dates 2010-06-04 and 2017-03-20.
- The GROUP BY clause was applied to group the landing outcomes, and the ORDER BY clause was used to rank the grouped outcomes in descending order.

Section 4

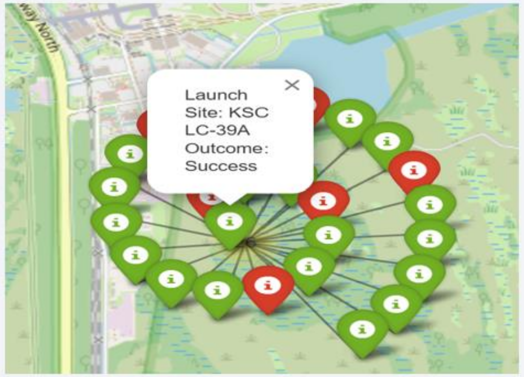
Launch Sites Proximities Analysis

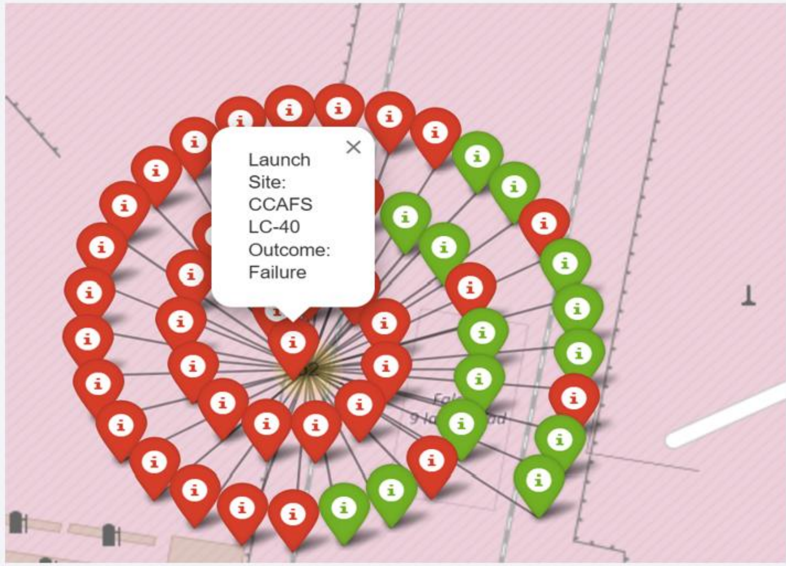
All launch sites



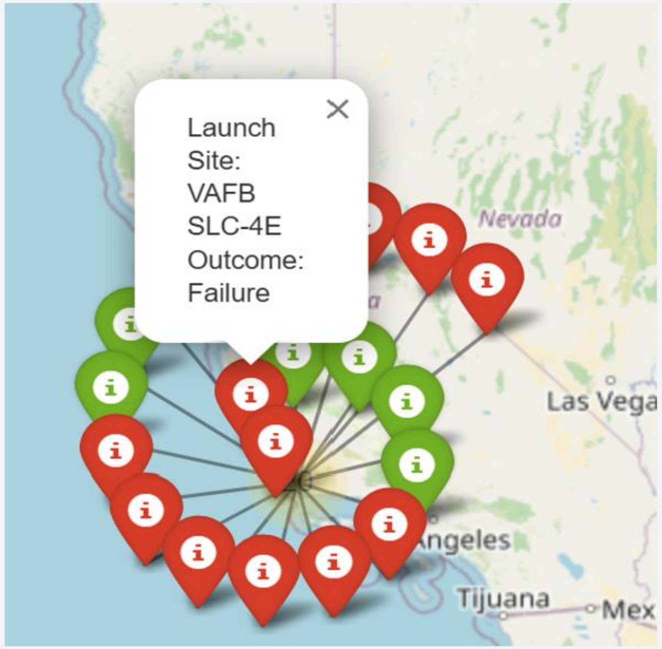
SpaceX launch sites are located in the United States, specifically in Florida and California.

Markers showing launch sites with color labels





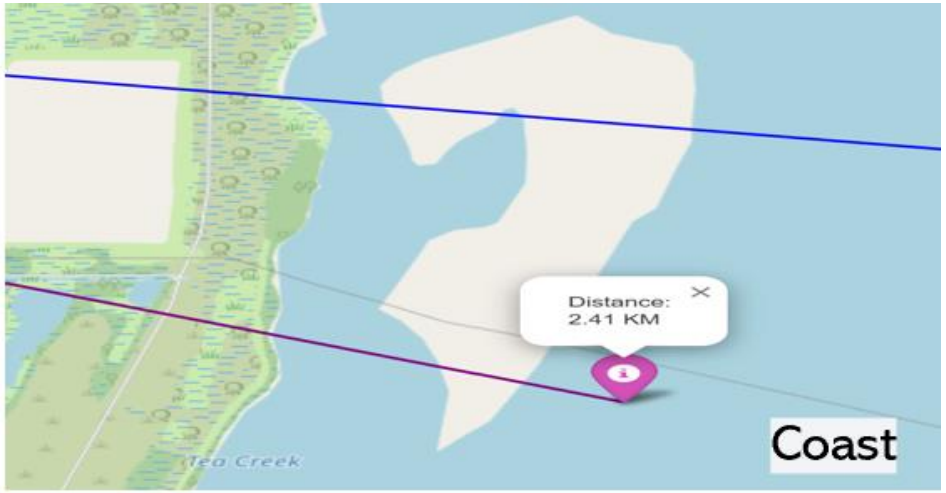
California Launch Site

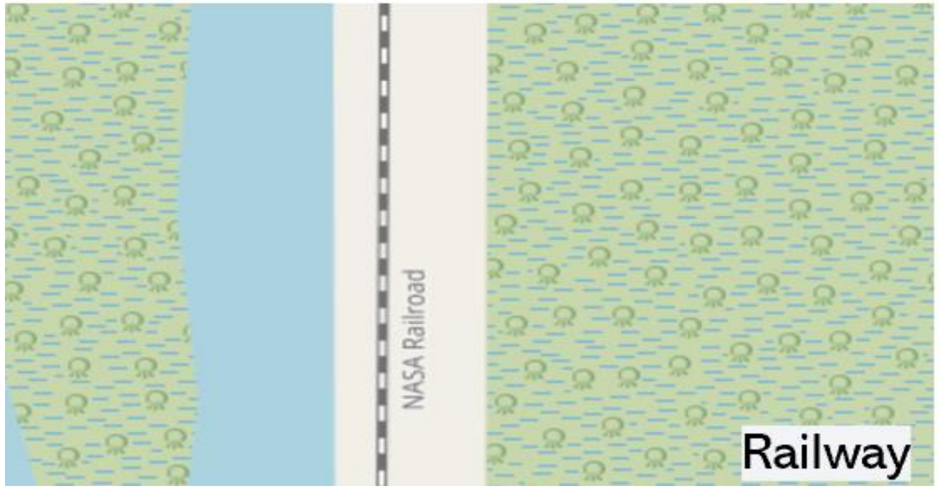


Florida Launch Site

The green marker indicates a successful landing, while the red marker represents a failed landing.

Launch Site distance to landmarks

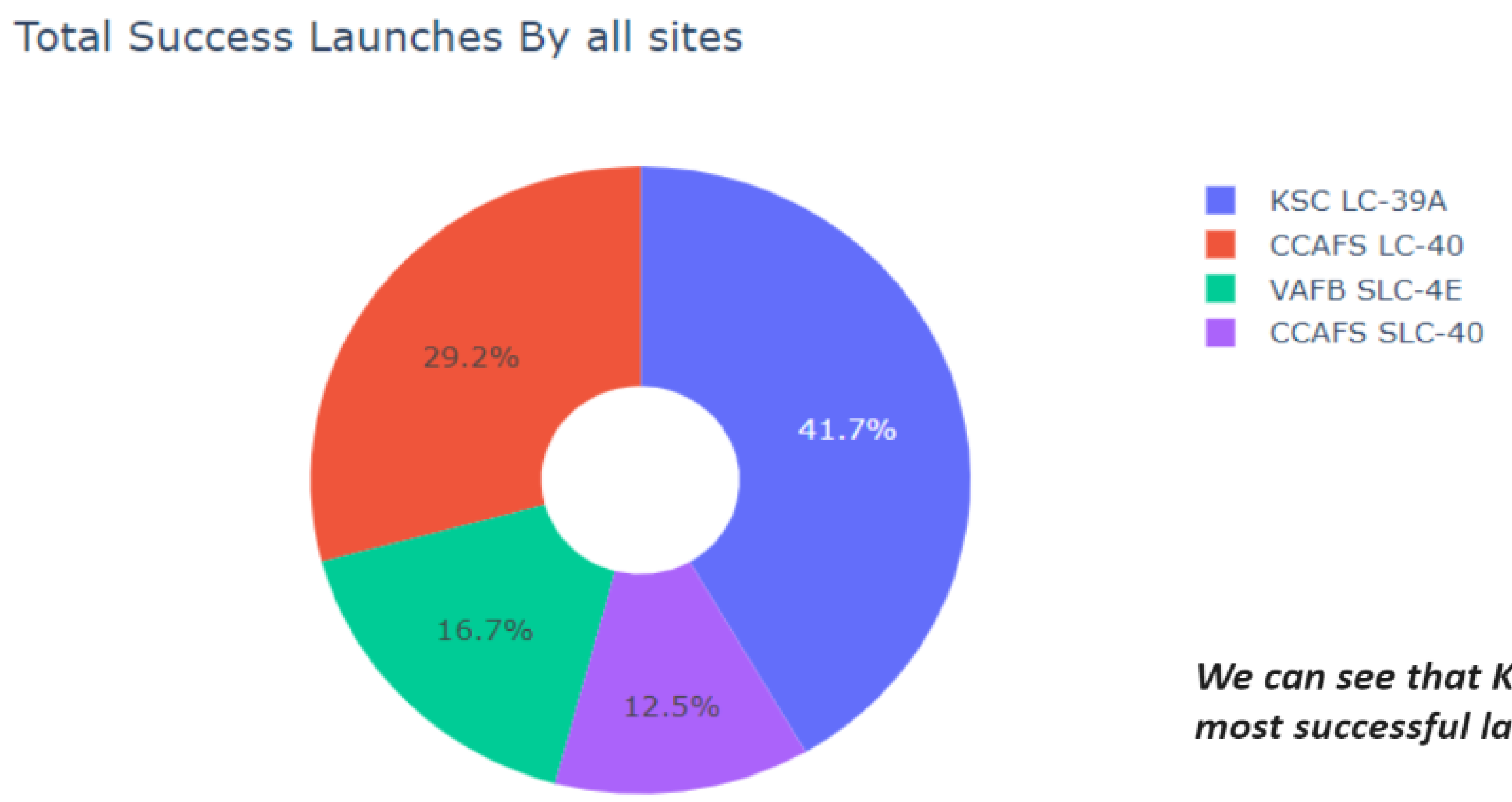




Section 5

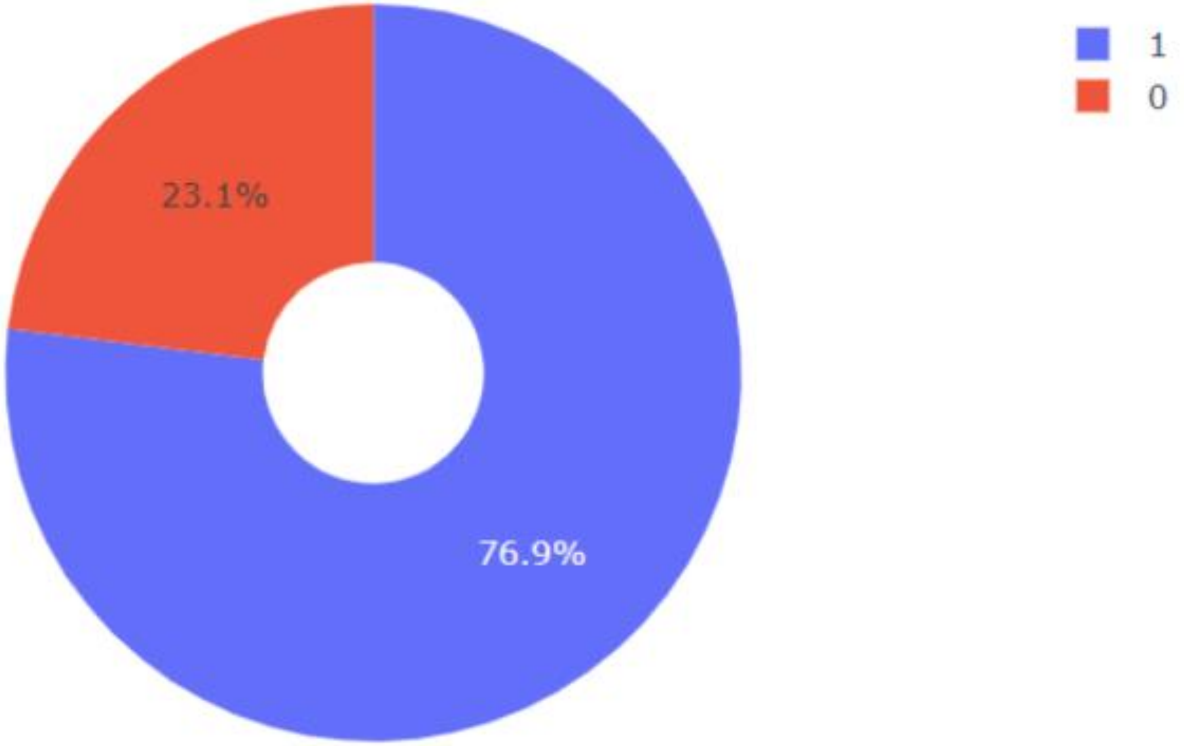
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



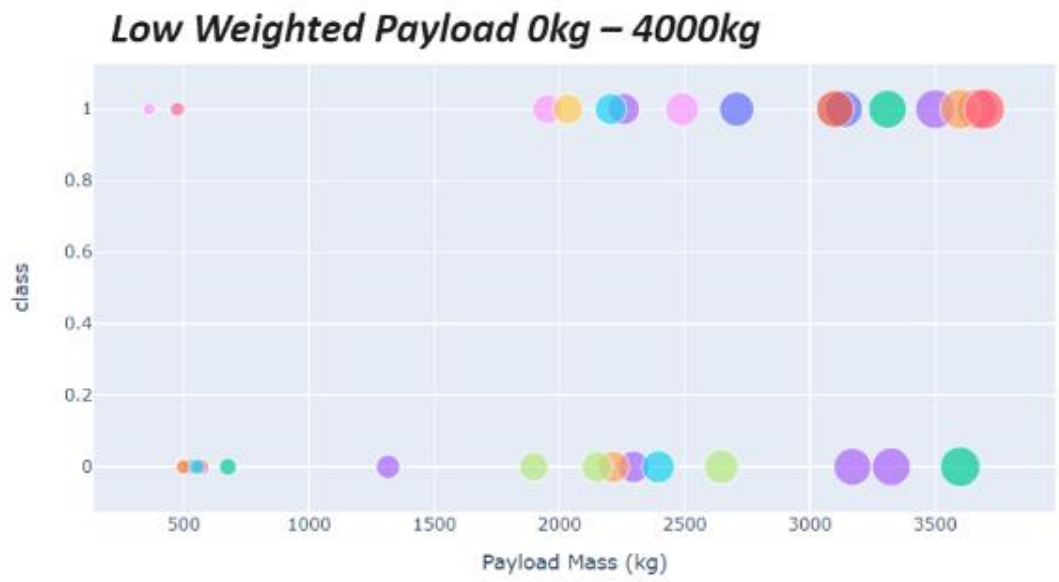
We can see that KSC LC-39A had the most successful launches from all the sites

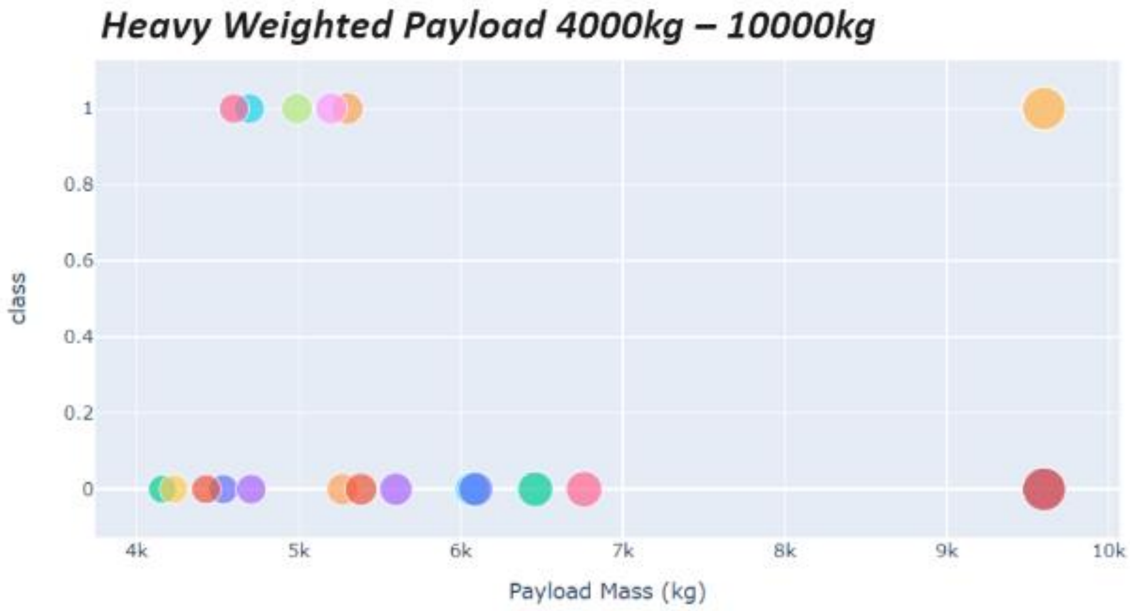
Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

Calculate the accuracy of tree_cv on the test data using the score method.

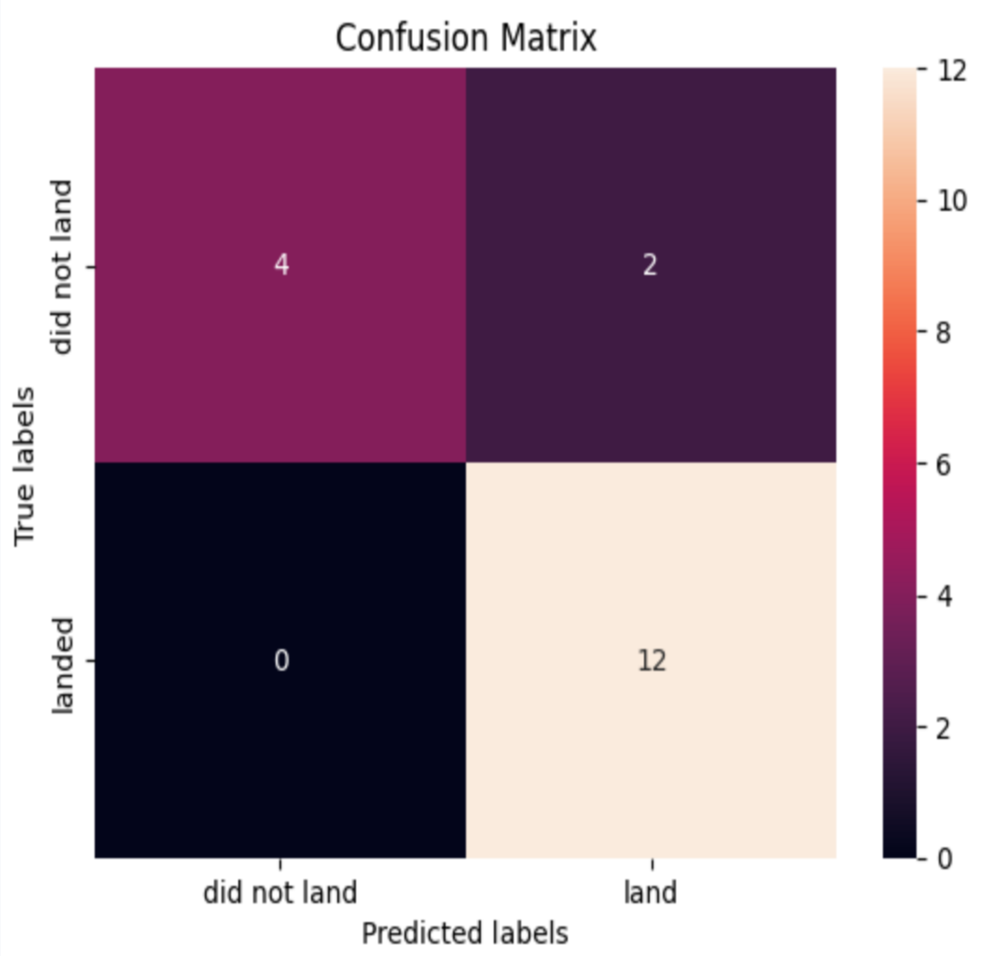
```
# Calculate the accuracy of tree_cv on the test data using the method score
test_accuracy = tree_cv.score(X_test, Y_test)
print("Test Accuracy for Decision Tree Classifier: ", test_accuracy)
```

```
Test Accuracy for Decision Tree Classifier:  0.8888888888888888
```

Confusion matrix:

Confusion Matrix

- The Decision Tree model is the best performing model, with an excellent confusion matrix. It correctly predicted 12 successful landings (True Positives), 4 failed landings (True Negatives), and had only 2 false positives. Notably, there were no false negatives, indicating the model's high accuracy in predicting successful landings.



Conclusions

Key Conclusions:

- CCAFS SLC 40 is the most active site, VAFB SLC 4E handles lighter payloads, and KSC LC 39A supports a broader range of payloads.
- The success rate of SpaceX launches steadily increased from 2013 to 2020, reflecting technological improvements.
- ES-L1, GEO, and HEO orbits have 100% success rates, while GTO orbits show the lowest success at 50%.
- Heavier payloads generally correlate with higher landing success, especially in Polar, LEO, and ISS orbits.
- The larger the flight amount at a launch site, the greater the success rate at that site.
- The first successful landing on a ground pad occurred on 22nd July 2018.
- The total payload mass for NASA (CRS) missions was 48,213 kg, and the average payload mass for F9 v1.1 was 2,928.4 kg.
- KSC LC-39A had the most successful launches of any site.
- The Decision Tree model achieved 88% accuracy, with a strong confusion matrix showing 12 True Positives and no False Negatives.

Thank you!