IBM Developer
SKILLS NETWORK

Yashashvee Meena
27-04-2025

Outline

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

Executive Summary

Summary of Methodologies:

The analysis involved the following steps:

- **Data Collection:** Gathering the necessary raw data.
- **Data Wrangling:** Cleaning, transforming, and preparing the data for analysis.
- **Exploratory Data Analysis with Data Visualisation:** Investigating data patterns and insights using visual tools.
- **Exploratory Data Analysis with SQL:** Utilising SQL to query and analyse the data.
- **Building an Interactive Map with Folium:** Creating a dynamic geographical representation of the data.
- **Building a Dashboard with Plotly Dash:** Developing an interactive dashboard to present key findings.
- **Predictive Analysis (Classification):** Employing classification techniques to make predictions.

Summary of all results

The key outcomes of this analysis include:

- **Exploratory Data Analysis Results**: Insights and observations derived from the initial data exploration.
- **Interactive Analytics Demo in Screenshots:** Visual demonstrations of the interactive analysis capabilities.
- **Predictive Analysis Results:** The outcomes and evaluation of the classification models.

Introduction

Project Background and Context:

Spacex has emerged as a leading company in the commercial space industry by significantly reducing the cost of space travel. Their Falcon rocket launches are advertised at $62 million, a considerable saving compared to the $165+ million charged by other providers. This cost-effectiveness is largely attributed to SpaceX's ability to reuse the first stage of its rockets. Consequently, determining the conditions under which the first stage will successfully land is crucial for understanding and potentially optimizing launch costs. This project aims to predict whether SpaceX will reuse the first stage based on publicly available data and machine learning models.

Questions to be Answered:

This analysis seeks to address the following questions:

- How do factors like payload mass, launch site, number of previous flights, and orbital trajectory influence the success of the first stage landing?
- Has the rate of successful first-stage landings improved over time?
- Which binary classification algorithm is most effective for predicting first-stage landing success in this specific scenario?

Section 1

Methodology

Methodology

Data Collection:

- Data was gathered using the SpaceX REST API.
- Additional information was obtained through web scraping from Wikipedia.

Data Wrangling:

- The collected data was filtered to relevant information.
- Missing values within the dataset were addressed.
- One-Hot Encoding was applied to prepare the data for binary classification.

Exploratory Data Analysis (EDA):

- EDA was conducted using both data visualization techniques and SQL queries to understand data characteristics and patterns.

Interactive Visual Analytics:

- Interactive visualizations were created using Folium for maps and Plotly Dash for dashboards to explore and present the data dynamically.

Predictive Analysis:

- Predictive analysis was performed using various classification models.
- This involved building, tuning, and evaluating these models to achieve optimal prediction results.

Data Collection

The data collection process involved a two-pronged approach, combining API requests to the SpaceX REST API with web scraping of a table from SpaceX's Wikipedia page. This dual method was necessary to gather comprehensive information about the launches for a more detailed analysis.

Data Columns Obtained via SpaceX REST API:
The following data points were retrieved using the SpaceX REST API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.

Data Columns Obtained via Wikipedia Web Scraping:
Web scraping from Wikipedia yielded the following data columns: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.

Data Collection – SpaceX API

Requesting rocket launch data from SpaceX API. This initiates the retrieval of information.

Decoding the response content using .json() and turning it into a dataframe using .json_normalize(). The raw data is converted into a structured dataframe format.

Requesting needed information about the launches from SpaceX API by applying custom functions. Specific data points are extracted and processed using defined functions

Constructing data we have obtained into a dictionary. The processed information is organized into a Python dictionary.

Creating a dataframe from the dictionary. The dictionary is then transformed back into a dataframe for easier manipulation.

Filtering the dataframe to only include Falcon 9 launches. The dataset is narrowed down to focus on the relevant rocket type.

Replacing missing values of Payload Mass column with calculated .mean() for this column. Missing data in the 'Payload Mass' column is imputed using the mean value of that column.

Exporting the data to CSV. Finally, the cleaned and processed data is saved into a CSV file.

GitHub URL: jupyter-labs-spacex-data-collection-api-v2

Data Collection - Scraping

1. Falcon 9 launch data was requested from Wikipedia.
2. A BeautifulSoup object was created from the received HTML response.
3. All column names were extracted from the header of the HTML table.
4. Data was collected by parsing the HTML tables.
5. The obtained data was structured into a dictionary.
6. A dataframe was created from this dictionary.
7. Finally, the data was exported to a CSV file.

GitHub Link : [jupyter-labs-webscraping]

Data Wrangling

Data wrangling is the process of refining raw data by cleaning, transforming, and organizing it into a structured format that is ready for analysis.

Step 1: Data Cleaning

- Missing values in the dataset were identified and then either filled or removed.
- Suitable imputation methods were applied, or rows/columns with a significant amount of missing data were dropped.

Step 2: Data Transformation

- Data types were converted to their correct formats, such as date-time or numerical.
- Text data was standardized by applying transformations like converting to lowercase and removing extra whitespace.
- Numerical features were normalized or scaled to ensure consistency across different variables.

Step 3: Data Integration

- Datasets obtained from various sources, such as the API and web scraping, were combined into a unified dataset.
- Consistency in column names and data formats was ensured across all integrated datasets.

Step 4: Data Validation

- The dataset was checked for and cleansed of any duplicate records.
- The accuracy and consistency of the data entries were verified.

GitHub Link: jupyter-spacex-Data wrangling

EDA with Data Visualization

Charts Generated:

The following visualizations were created to explore the data:

- Relationships between Flight Number and Payload Mass, Flight Number and Launch Site, and Payload Mass and Launch Site were examined.
- The correlation between Orbit Type and Success Rate was visualized.
- The relationship between Flight Number and Orbit Type was plotted.
- The interplay between Payload Mass, Orbit Type, and Success Rate, as well as the yearly trend of these factors, were illustrated.

Chart Type Descriptions:

- **Scatter plots** were used to identify potential relationships between different variables, which could then inform the development of machine learning models.
- **Bar charts** were employed to compare distinct categories and highlight the relationship between these categories and a measured value.
- **Line charts** were utilized to display trends in the data over time, representing time series analysis.

GitHub Link: jupyter-labs-eda-dataviz

EDA with SQL

Aggregate Queries:

- Determined the total number of launches.
- Counted the number of successful and failed launches.
- Calculated the success rates categorized by launch site and rocket type.

Join Queries:

- Integrated launch records with supplementary data, such as rocket specifications, by joining tables.
- Combined different datasets to enable a more thorough analysis.

Filtering Queries:

- Selected specific subsets of data based on launch outcomes (success or failure).
- Applied criteria, such as launch date or rocket configuration, to extract relevant launches.

Sorting Queries:

- Arranged data to identify patterns or unusual values.
- Ordered launches by date or success rate to facilitate analysis.

Subqueries:

- Used nested queries to compute derived metrics, for example, the average payload mass per launch site.
- Employed subqueries to conduct detailed analyses within larger datasets.

GitHub Link: jupyter-labs-eda-sql-coursera_sqllite

Build an Interactive Map with Folium

Map Objects Created:

- **Markers:**
  - Individual markers were positioned on the map to denote specific launch sites.
  - Each marker corresponds to a precise geographical location where SpaceX has conducted launches.
- **Circles:**
  - Circles were added around launch sites to visually represent areas of proximity.
  - These circles help to visualize the surrounding areas of launch sites that might be relevant for operational considerations.
- **Lines:**
  - Lines were drawn to connect launch sites with nearby locations or other significant points.
  - These lines provide spatial context and illustrate connections between different points of interest related to launches.

Reasons for Adding Objects:

- **Markers:**
  - To precisely locate launch sites for spatial reference.
  - Helps users geographically identify where SpaceX launches have taken place.
- **Circles:**
  - To visually represent the potential impact zones around launch sites.
  - Provides a visual indication of safety perimeters or operational boundaries.
- **Lines:**
  - To show connections or relationships between launch sites and relevant features.
  - Enhances the understanding of spatial relationships and dependencies.

GitHub Link: lab-jupyter-launch-site-location

Build a Dashboard with Plotly Dash

•**Launch Sites Dropdown List:** A dropdown menu was created, allowing users to easily select and focus on specific launch sites. This provides the flexibility to analyze data for all sites combined or drill down to the performance of an individual site.

•**Pie Chart for Success Launches:** A pie chart was implemented to visualize the proportion of successful launches. This chart has a dual purpose:

•When no specific launch site is selected in the dropdown, it displays the total count of successful launches across *all* sites.

•When a particular launch site *is* selected, the pie chart dynamically updates to show the breakdown of success versus failure counts specifically for that chosen site.

•**Slider for Payload Mass Range:** A slider was added to enable users to filter the data based on a specific range of payload masses. This interactive element allows for the exploration of how launch outcomes might vary depending on the weight of the payload.

•**Scatter Chart of Payload Mass vs. Success Rate for Different Booster Versions:** A scatter plot was generated to visualize the relationship between payload mass and launch success. Importantly, this chart differentiates between different booster versions, allowing for the identification of potential correlations or performance variations among them across various payload weights.

GitHub URL: spacex-dash-app.py

Predictive Analysis (Classification)

We start by taking the column labeled "Class" from our dataset and turning it into a NumPy array. This array will hold the values we want our model to predict.

→ Used StandardScaler. This involves fitting the scaler to our data (calculating the mean and standard deviation) and then transforming the data to have zero mean and unit variance.

→ Used the train_test_split function to divide our dataset into two parts: a training set to teach the model and a testing set to see how well it generalises to unseen data.

→ Created a GridSearchCV object. Tried out different combinations of parameters (hyperparameters) for our models.

→ Applied GridSearchCV process to several popular algorithms: Logistic Regression (LogReg), Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN).

→ Used the .score() method on our testing set to calculate the accuracy for each model. This gives us a straightforward measure of how many predictions the model made.

→ To get a more nuanced understanding of our models' performance, we'll look at the confusion matrix for each. This matrix shows just the overall accuracy.

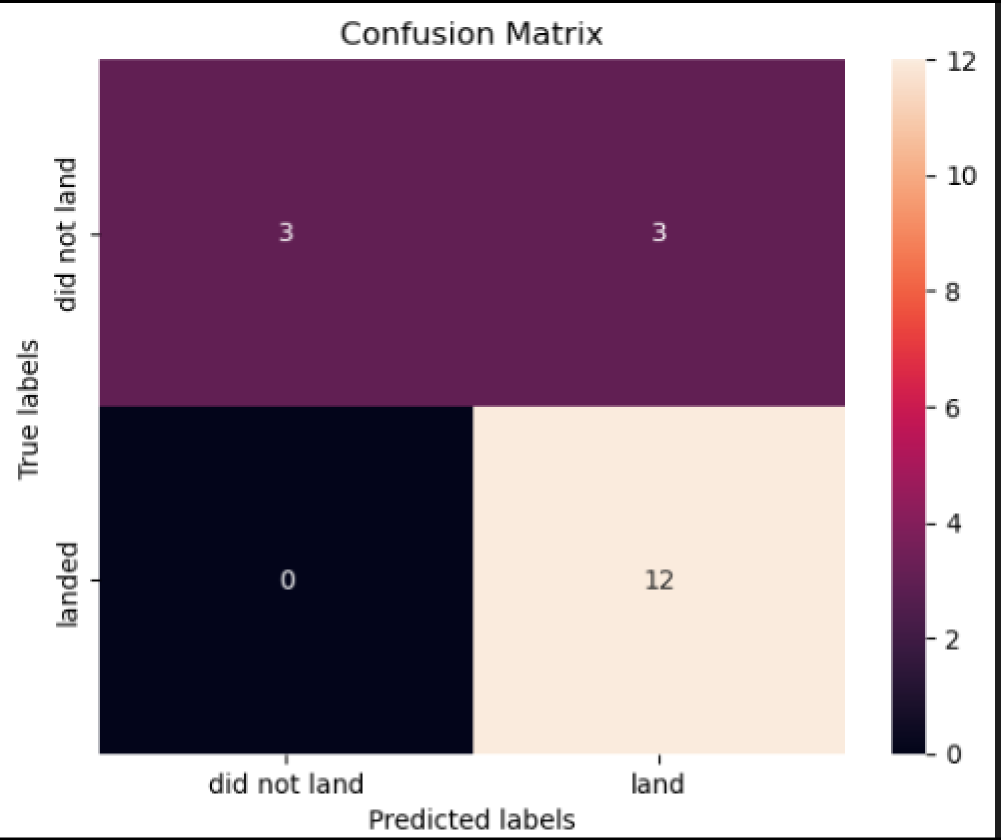
→ Finally, to decide which model is the best, we'll examine the Jaccard score and F1-score metrics.

GitHub URL: [SpaceX-Machine-Learning-Prediction]()

Results

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

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.783784	0.819444
F1_Score	0.909091	0.916031	0.878788	0.900763
Accuracy	0.866667	0.877778	0.822222	0.855556

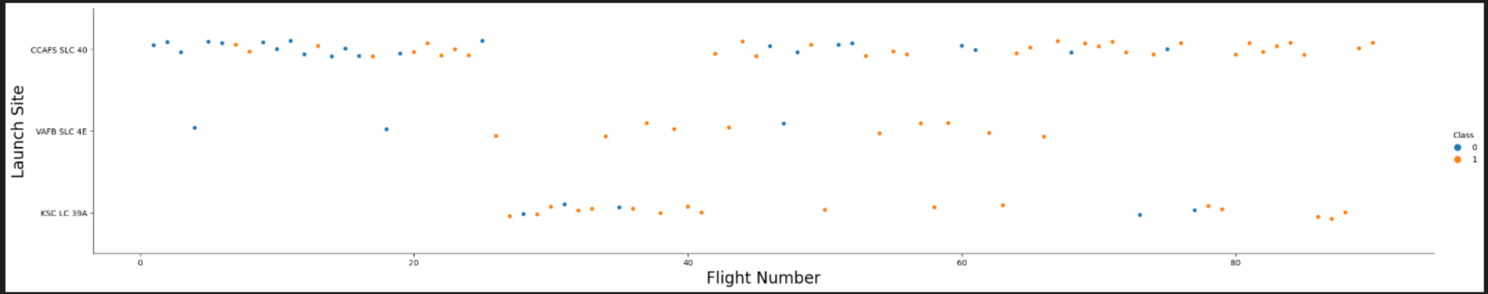


TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

✓ 0.3s



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

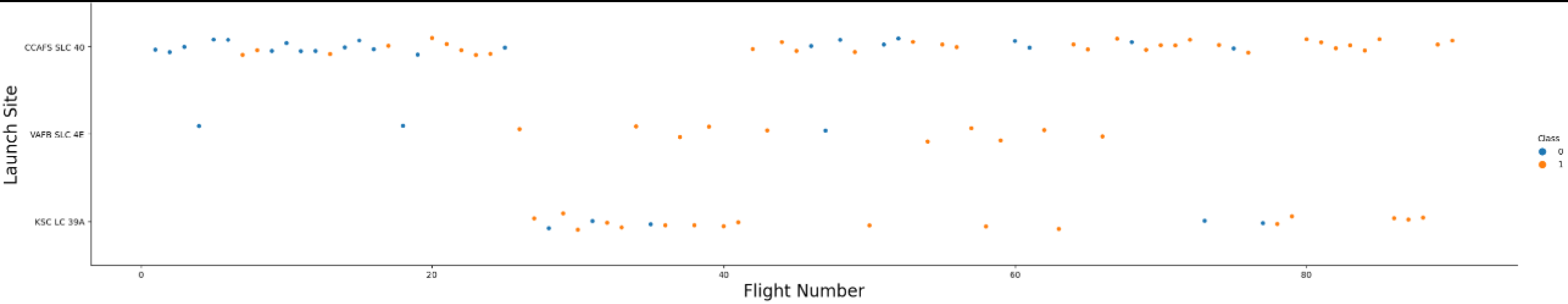
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect = 5)
```

Section 2

Insights drawn from EDA

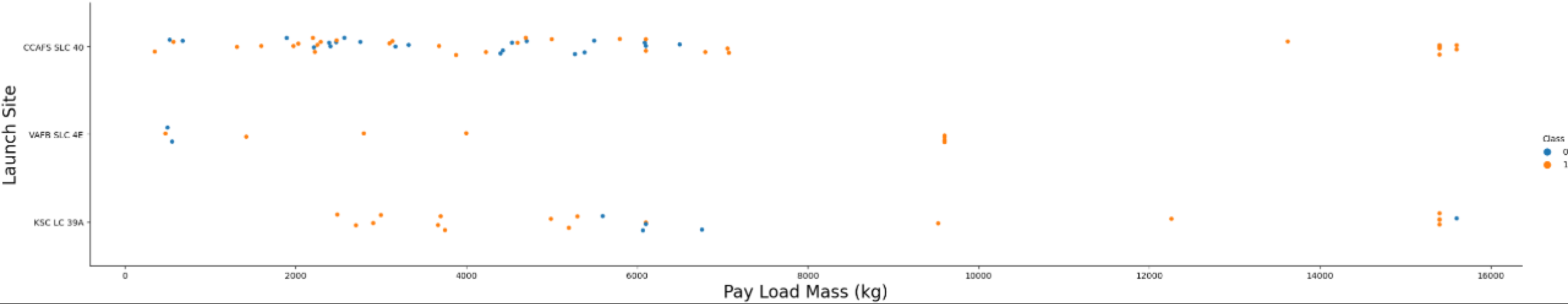
Flight Number vs. Launch Site

- Certain launch sites, such as CCAFS SLC-40 and KSC LC-39A, show a higher number of successful landings (Class = 1) as the flight number increases.
- The VAFB SLC-4E site has fewer launches overall, and the success rate is less consistent
- As the flight number increases, the success rate improves across all launch sites, indicating a learning curve and technological advancements over time.



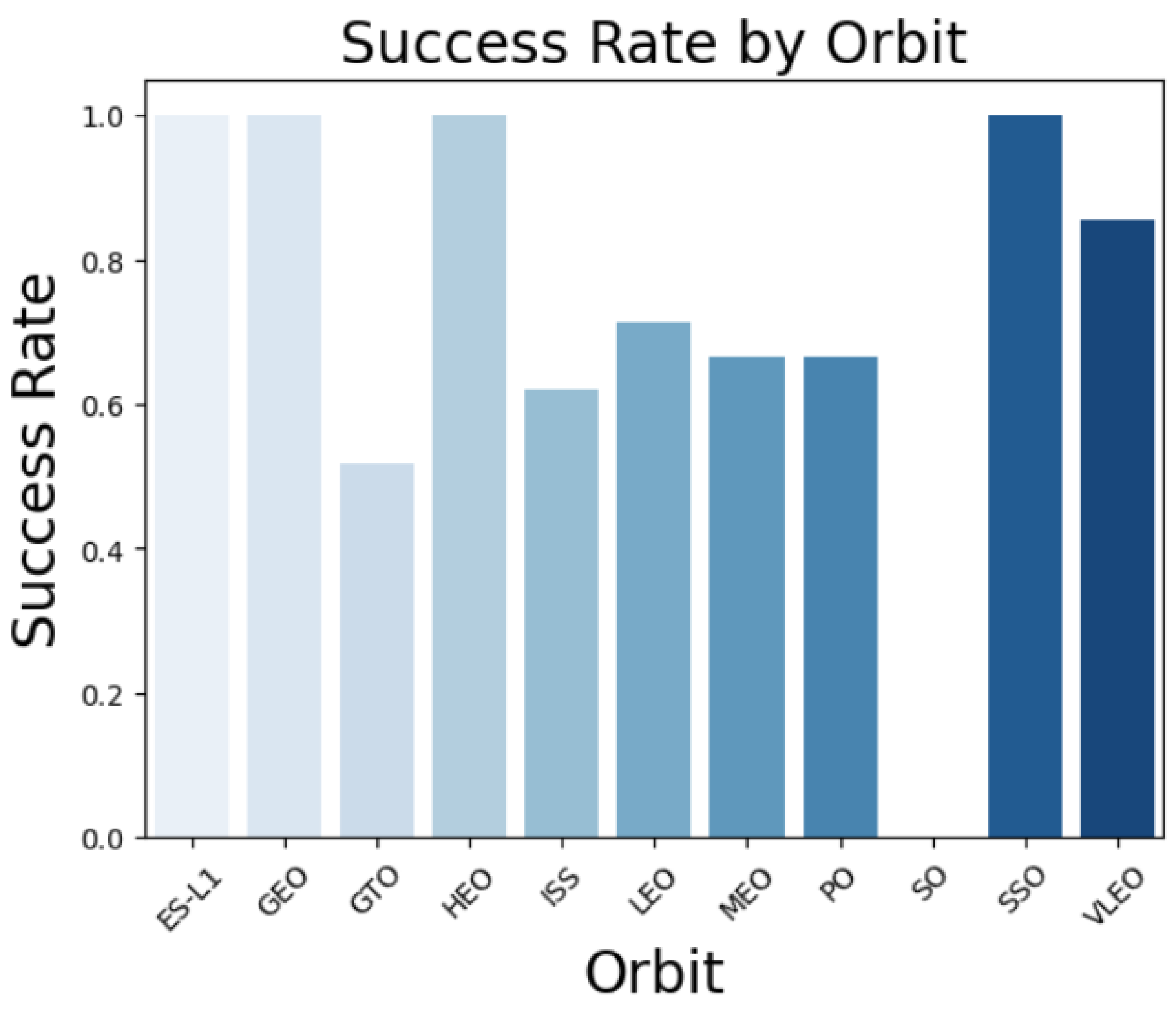
Payload vs. Launch Site

- The VAFB-SLC launch site does not handle heavy payloads (greater than 10,000 kg).
- Heavier payloads (greater than 10,000 kg) are more likely to succeed at CCAFS SLC-40 and KSC LC-39A.
- The graph suggests that certain launch sites specialize in specific payload ranges, which may be due to infrastructure, mission type, or other operational factors.



Success Rate vs. Orbit Type

- Orbits like LEO (Low Earth Orbit) and ISS (International Space Station) have the highest success rates.
- Polar and SSO (Sun-Synchronous Orbit) have moderate success rates, indicating some challenges, but are still relatively reliable.
- GTO (Geostationary Transfer Orbit) has a lower success rate than other orbits. This could be due to the higher complexity and energy requirements for these missions.



Flight Number vs. Orbit Type

1. LEO (Low Earth Orbit):

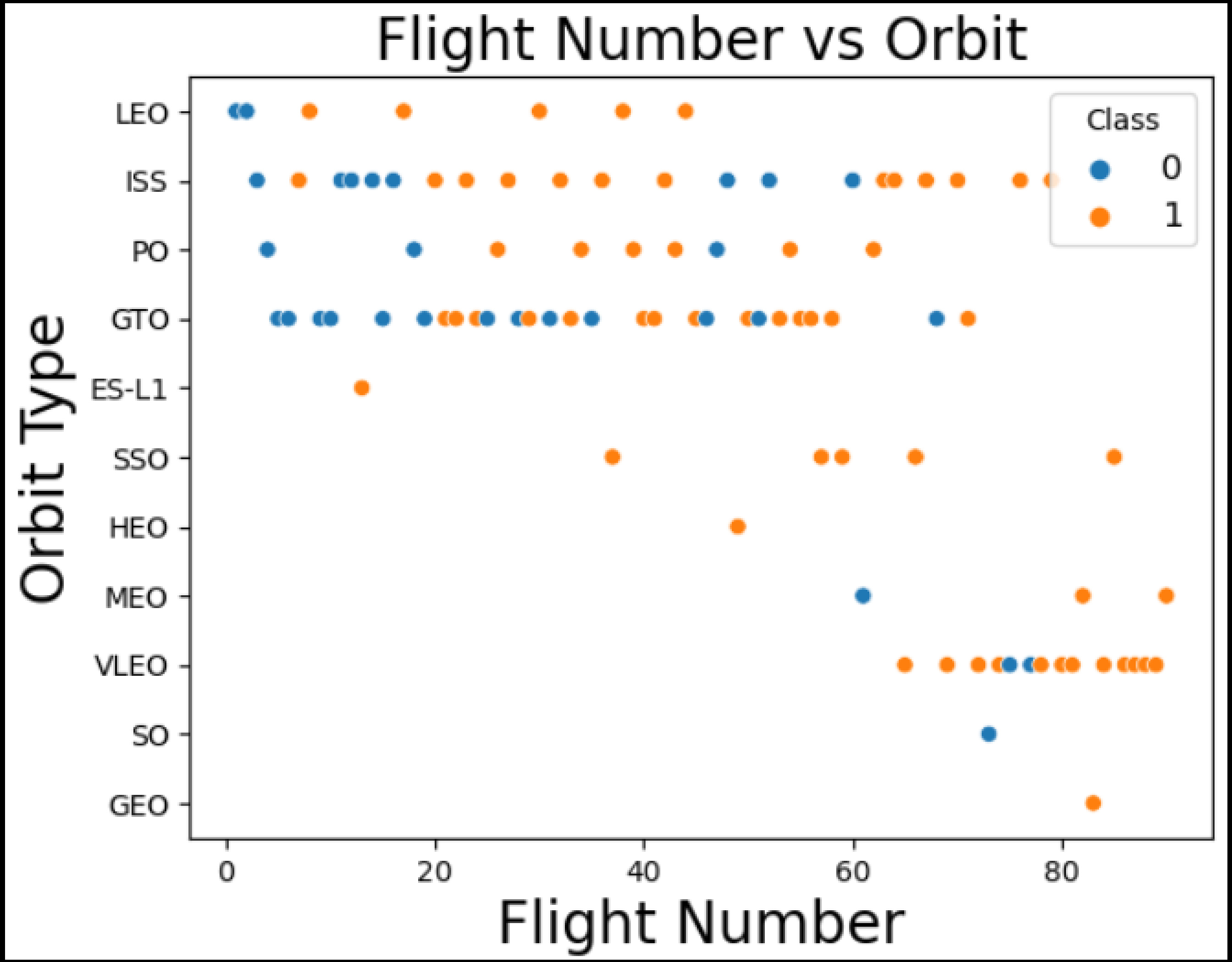
1. Success rates improve as the flight number increases, indicating a learning curve and better optimisation over time.
2. Most missions in LEO are successful, especially in later flights.

2. GTO (Geostationary Transfer Orbit):

1. There is no clear relationship between flight number and success rate.
2. Both successful and unsuccessful missions are observed across all flight numbers.

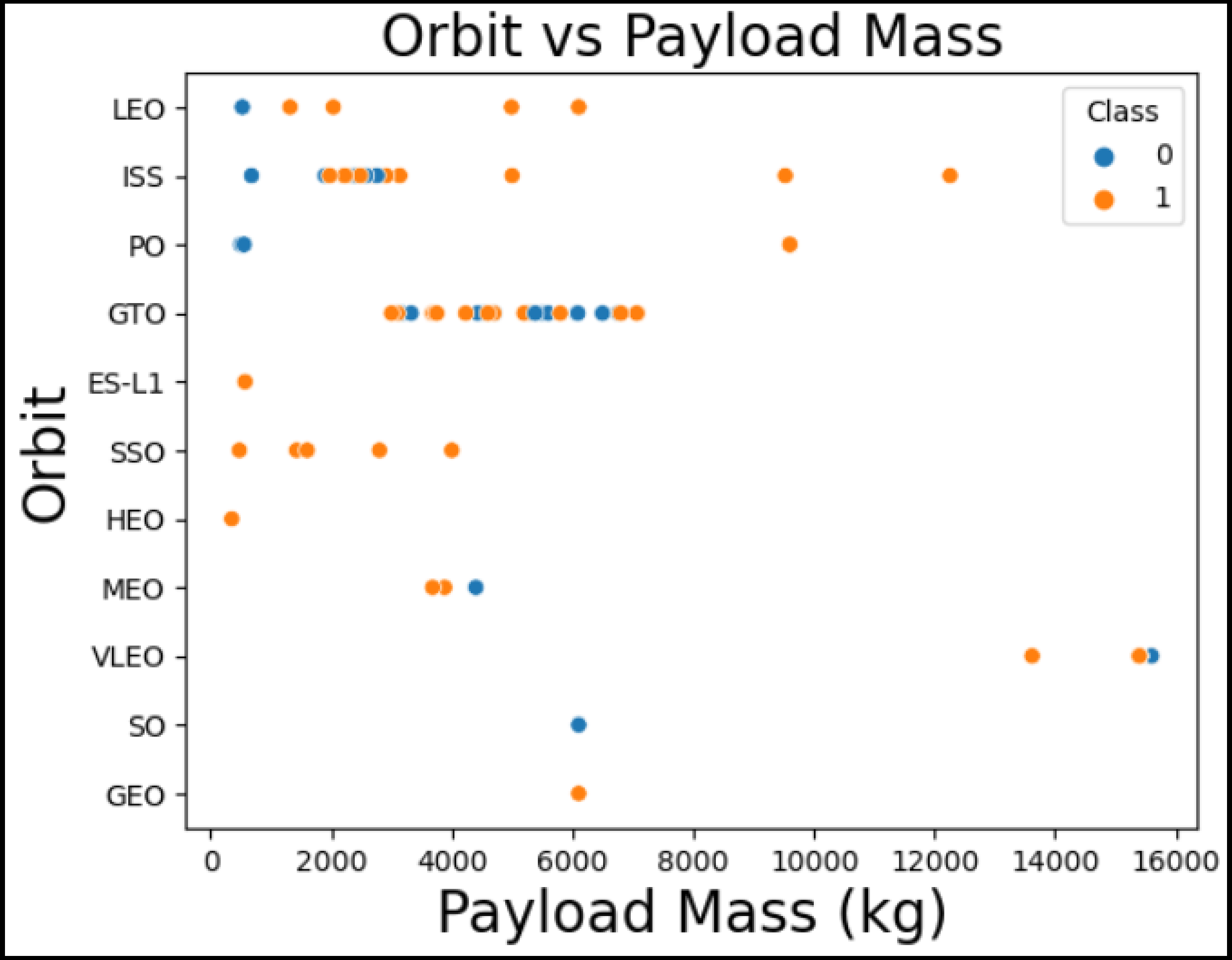
3. Other Orbits (e.g., Polar, ISS):

1. Success rates are generally higher, but the number of missions is lower compared to LEO and GTO.
2. These orbits show consistent success in later flights.



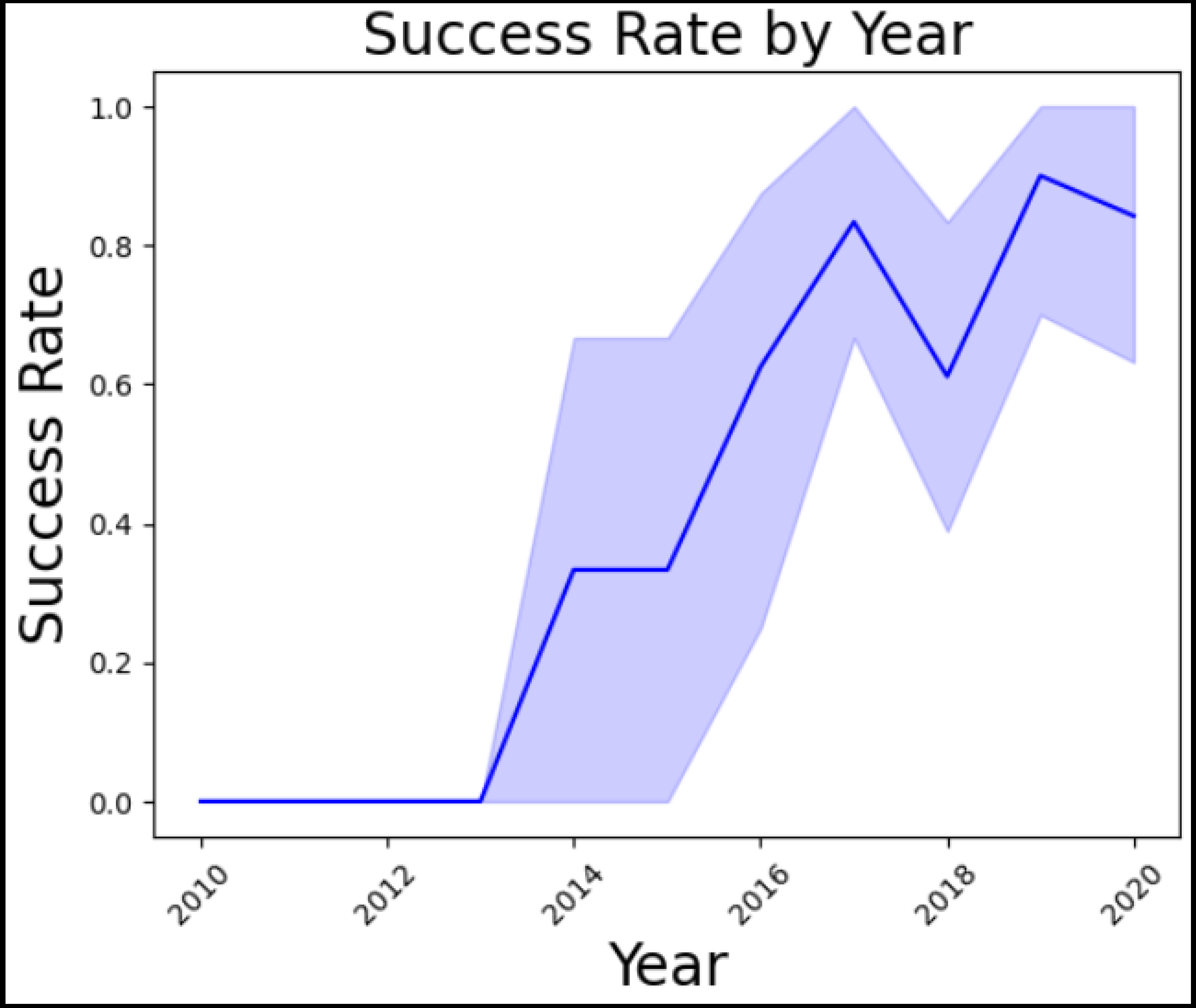
Payload vs. Orbit Type

1. **Heavy Payloads**:
   1. Orbits like **Polar**, **LEO (Low Earth Orbit)**, and **ISS (International Space Station)** show higher success rates (Class = 1) for heavy payloads.
   2. **GTO (Geostationary Transfer Orbit)** has mixed results for heavy payloads, with both successes and failures.
2. **Light Payloads**:
   1. Light payloads generally have higher success rates across all orbit types, indicating that lighter payloads are easier to manage.
3. **GTO Challenges**:
   1. GTO missions show a mix of successes and failures regardless of payload mass, suggesting that this orbit type is more challenging due to its higher energy requirements



Launch Success Yearly Trend

1. **Increasing Success Rate**:
    1. The success rate has steadily increased over the years, particularly after 2015. This indicates improvements in technology, processes, and experience.
2. **Stability in Recent Years**:
    1. After 2017, the success rate appears to have stabilized at a high level, reflecting consistent performance and reliability in launches.
3. **Early Challenges**:
    1. In the earlier years (before 2015), the success rate was lower, likely due to the learning curve and initial challenges in developing reusable rocket technology.



Retrieves a list of unique launch sites from the SPACEXTABLE

All Launch Site Names

Display the names of the unique launch sites in the space mission

```
[13]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE;
```

 * sqlite:///my_data1.db
Done.

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

Retrieves a list of unique launch sites from the SPACEXTABLE

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5; ...
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Retrieves the first 5 records from SPACEXTABLE where the LAUNCH_SITE starts with "CCA"

Total Payload Mass

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

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

[13] ✓ 0.0s

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

Total_Payload_Mass
45596

Calculated the total payload mass carried by boosters launched for NASA (CRS).

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTABLE WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

Average_Payload_Mass
2928.4

Calculated the average payload mass carried by boosters of version F9 v1.1.

First Successful Ground Landing Date

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

Hint:Use min function

```
%sql SELECT MIN(DATE) AS First_Launch_Date FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

First_Launch_Date
2015-12-22

Found the earliest date when a successful landing on a ground pad occurred.

Successful Drone Ship Landing with Payload between 4000 and 6000

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 (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000) AND LANDING_OUTCOME = 'Success (drone ship)';
```

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

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

Listed the booster versions that successfully landed on a drone ship and carried a payload mass between 4000 and 6000 kg

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS TOTAL_NUMBER FROM SPACEXTABLE GROUP BY MISSION_OUTCOME;
```

✓ 0.0s

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

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

Counted the total number of successful and failed missions.

Boosters Carried Maximum Payload

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTABLE WHERE (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) = PAYLOAD_MASS__KG_;
```

✓ 0.0s

```
 * 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

Retrieved the booster versions that carried the maximum payload mass

2015 Launch Records

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 LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, substr(Date, 6, 2) as month FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND LANDING_OUTCOME = 'Failure (drone ship)';
```

✓ 0.0s Python

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

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

Lists records from 2015 where the landing outcome was a failure on a drone ship, including the landing outcome, booster version, launch site, and month.

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.

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

✓ 0.0s Python

* sqlite:///my_data1.db
Done.

Landing_Outcome	COUNT_OUTCOMES
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

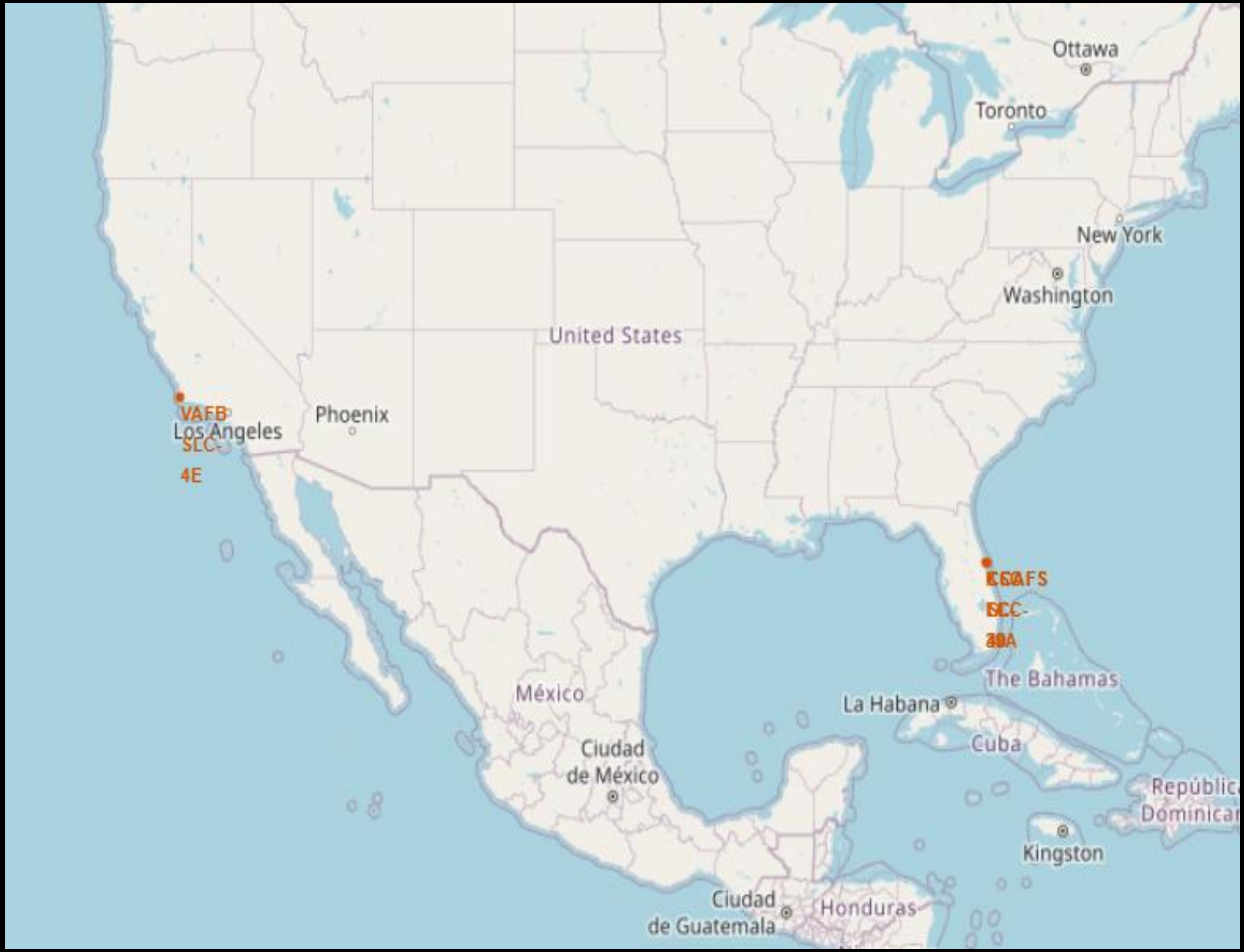
Counted the number of failures on drone ships and successes on ground pads between the specified dates, sorted in descending order.

Section 3

Launch Sites Proximities Analysis

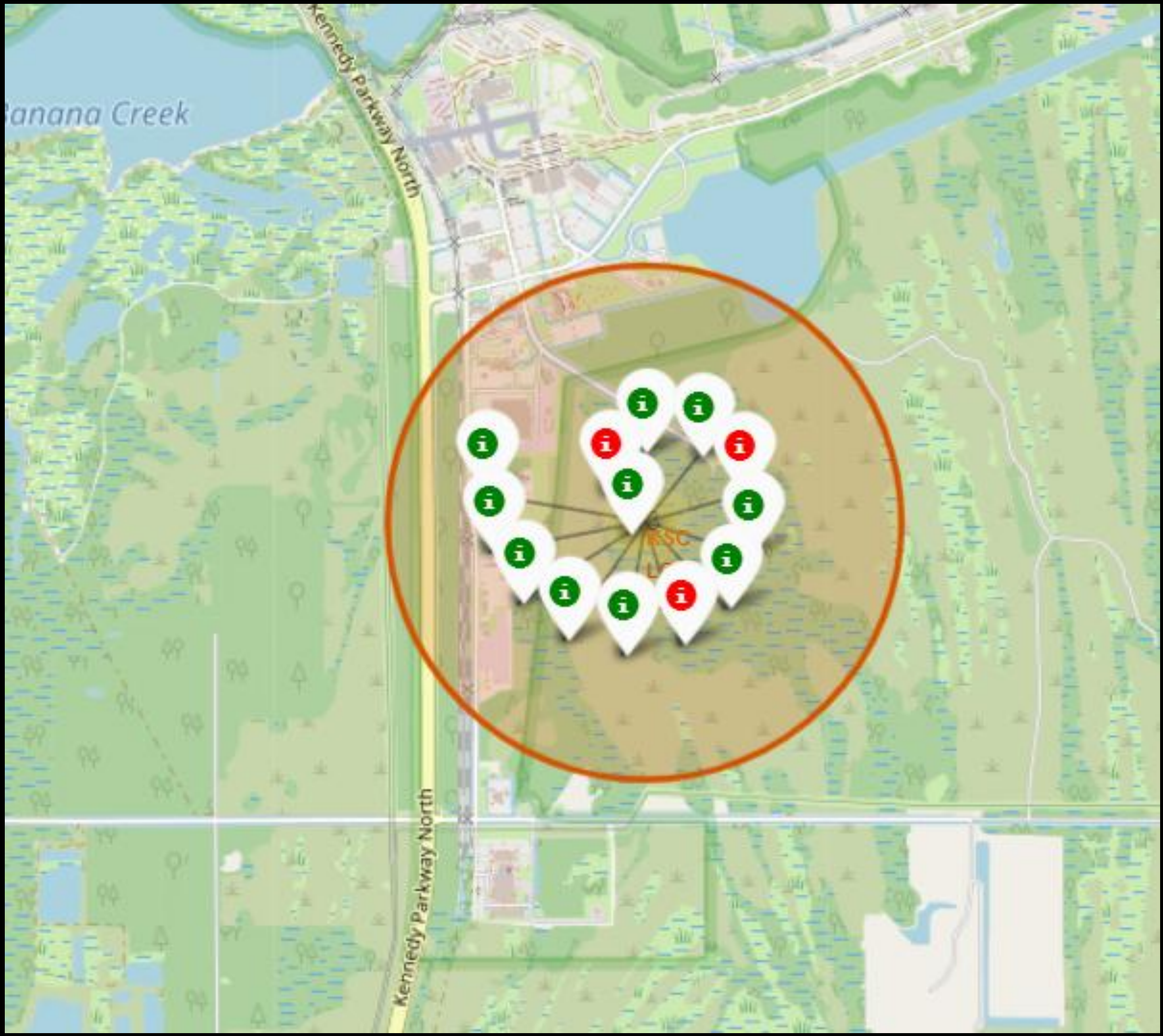
Launch Sites Marked on the Map

- **Graph:** A map with circles and markers for each launch site.
- **Outcome:**
    - The map shows the geographical locations of all SpaceX launch sites.
    - The launch sites are marked with circles and labeled with their names.
    - **Findings:**
        - All launch sites are located near the coastline, which is essential for safety and trajectory optimization.
        - The sites are relatively close to the equator, which helps maximize the rocket's velocity due to Earth's rotation.



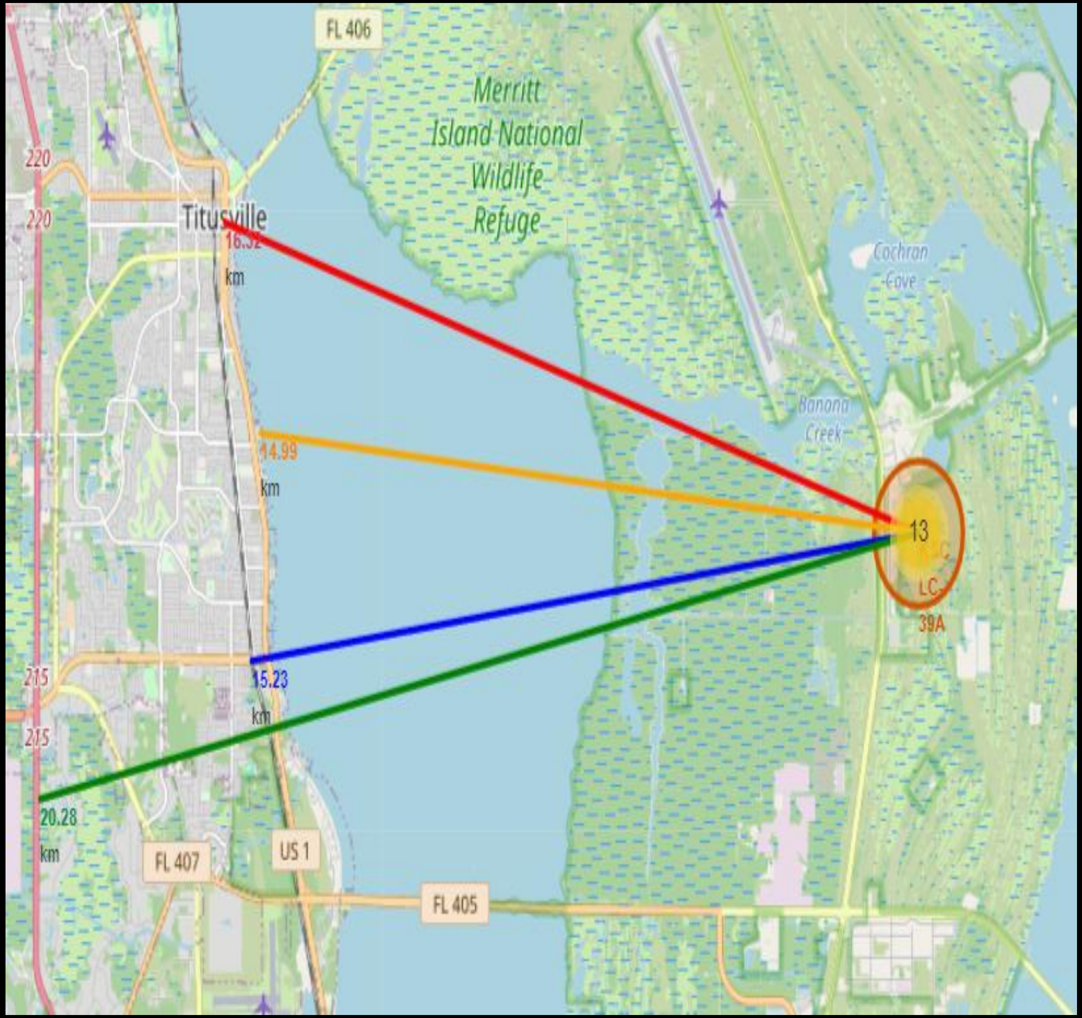
Success/Failure Launches Marked on the Map

- **Graph:** A map with green and red markers for successful and failed launches, respectively.
- **Outcome:**
  - Each launch is represented by a marker, with green indicating success (class=1) and red indicating failure (class=0).
  - **Findings:**
    - Some launch sites have a higher concentration of green markers, indicating better success rates.
    - The visualization helps identify which sites are more reliable for successful launches



Distance to Other Proximities (Railways, Highways, Cities)

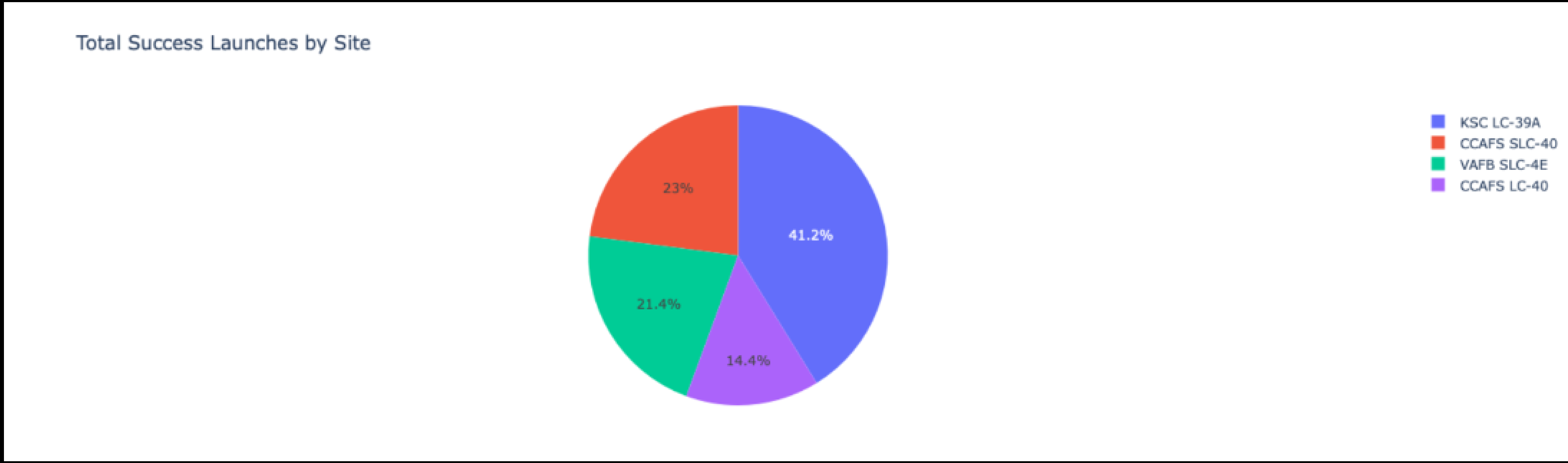
- **Graph:** Lines connecting launch sites to nearby railways, highways, and cities.
- **Outcome:**
  - The distances to key infrastructure like railways, highways, and cities are visualized.
  - **Findings:**
    - Launch sites are generally close to railways and highways for easy transportation of rockets and equipment.
    - Launch sites are located at a safe distance from cities to minimize risks to populated areas.



Section 4

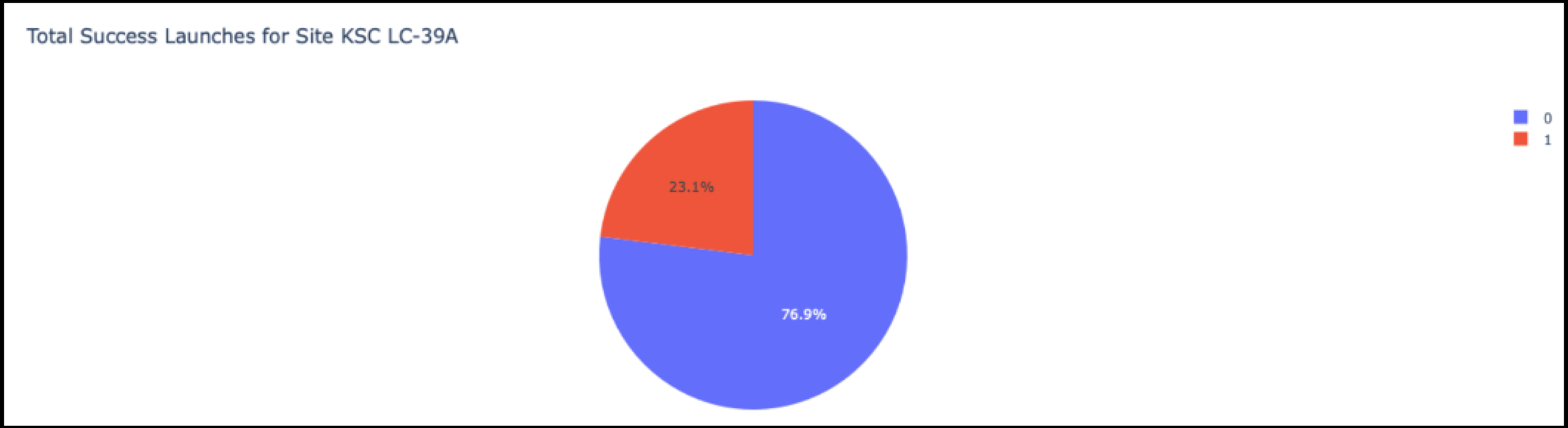
Build a Dashboard with Plotly Dash

Total Success Launches by Site



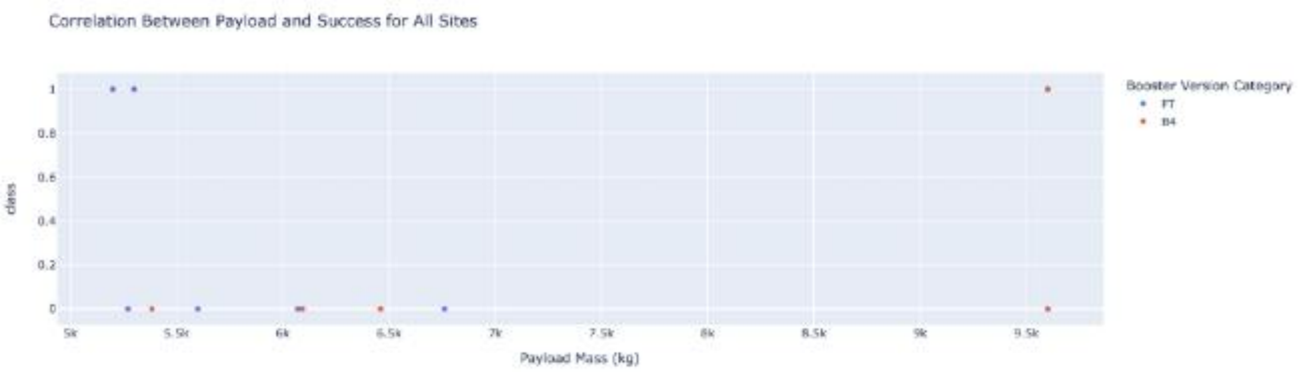
- **KSC LC-39A** (represented by the blue slice, accounting for **41.2%**) has the **highest** number of successful launches among all the sites. This is clearly the largest slice in the pie chart.
- The other launch sites, **CCAFS SLC-40** (red, **23%**), **VAFB SLC-4E** (green, **21.4%**), and **CCAFS LC-40** (purple, **14.4%** - note that "CCAFS LC-40" appears twice in the legend, which might be a typo and should likely represent a different site), have a smaller share of the total successful launches compared to KSC LC-39A.

Total Success Launches for Site KSC LC-39A



- **KSC LC-39A** has the **highest launch success rate** at **76.9%**. This is represented by the larger blue slice in the pie chart.
- This success rate is calculated based on **10 successful landings** (represented by the blue color in the legend, likely denoted as '1') and only **3 failed landings** (represented by the red color in the legend, likely denoted as '0').

Payload Mass vs. Launch Outcome for all sites



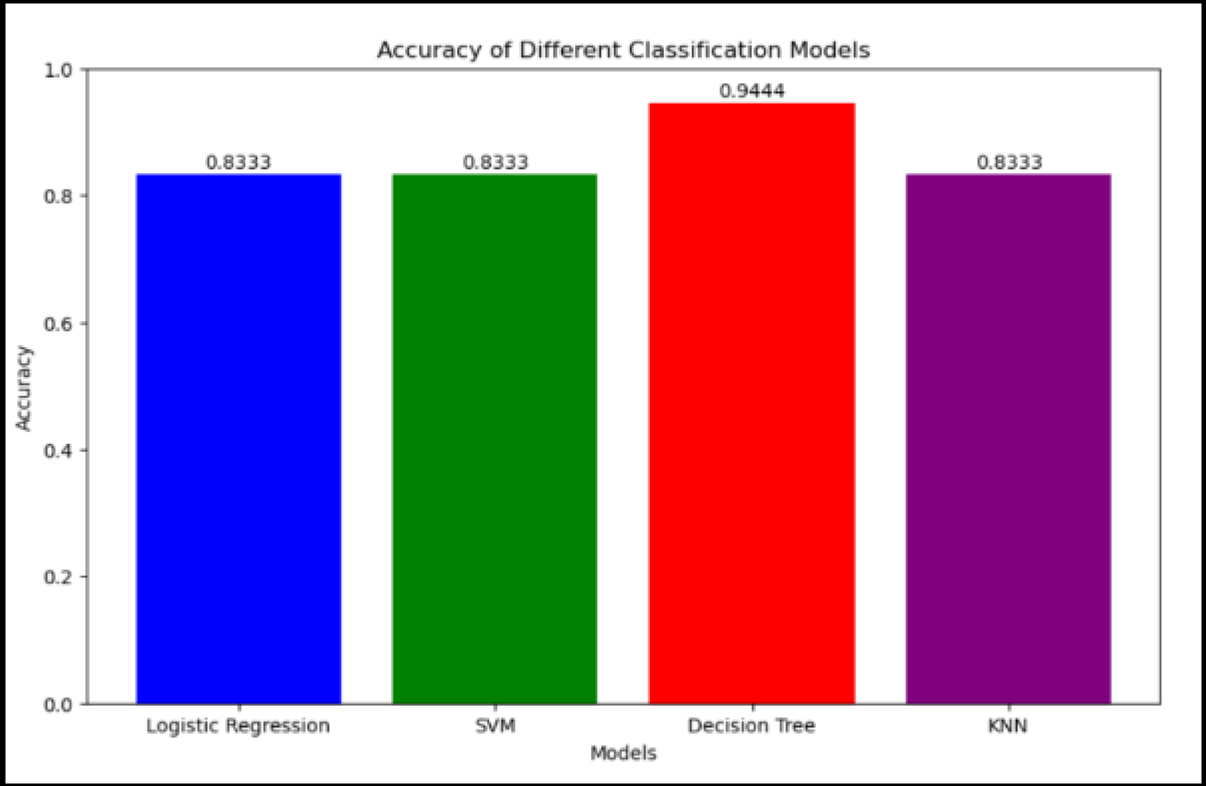
•**Payload Mass Range for Highest Success:** The key takeaway from the explanation is that **payloads with a mass between 2000 and 5500 kg** appear to have the **highest success rate**.
•**Visual Representation:** The scatter plots themselves show individual data points, where the x-axis represents the Payload Mass (in kg) and the y-axis likely represents the Booster Version Category (as indicated by the legend on the right). The color or shape of each point probably indicates the launch outcome (e.g., success or failure), although this isn't explicitly stated in the explanation or clearly visible without a closer look at the legend and data points.
•**Booster Version Differentiation:** The legend on the right ("Booster Version Category") indicates that different booster versions (e.g., B2, F9 v1.0, FT, etc.) are represented by different colors or markers. This allows us to see if certain booster versions perform better or worse across different payload mass ranges.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

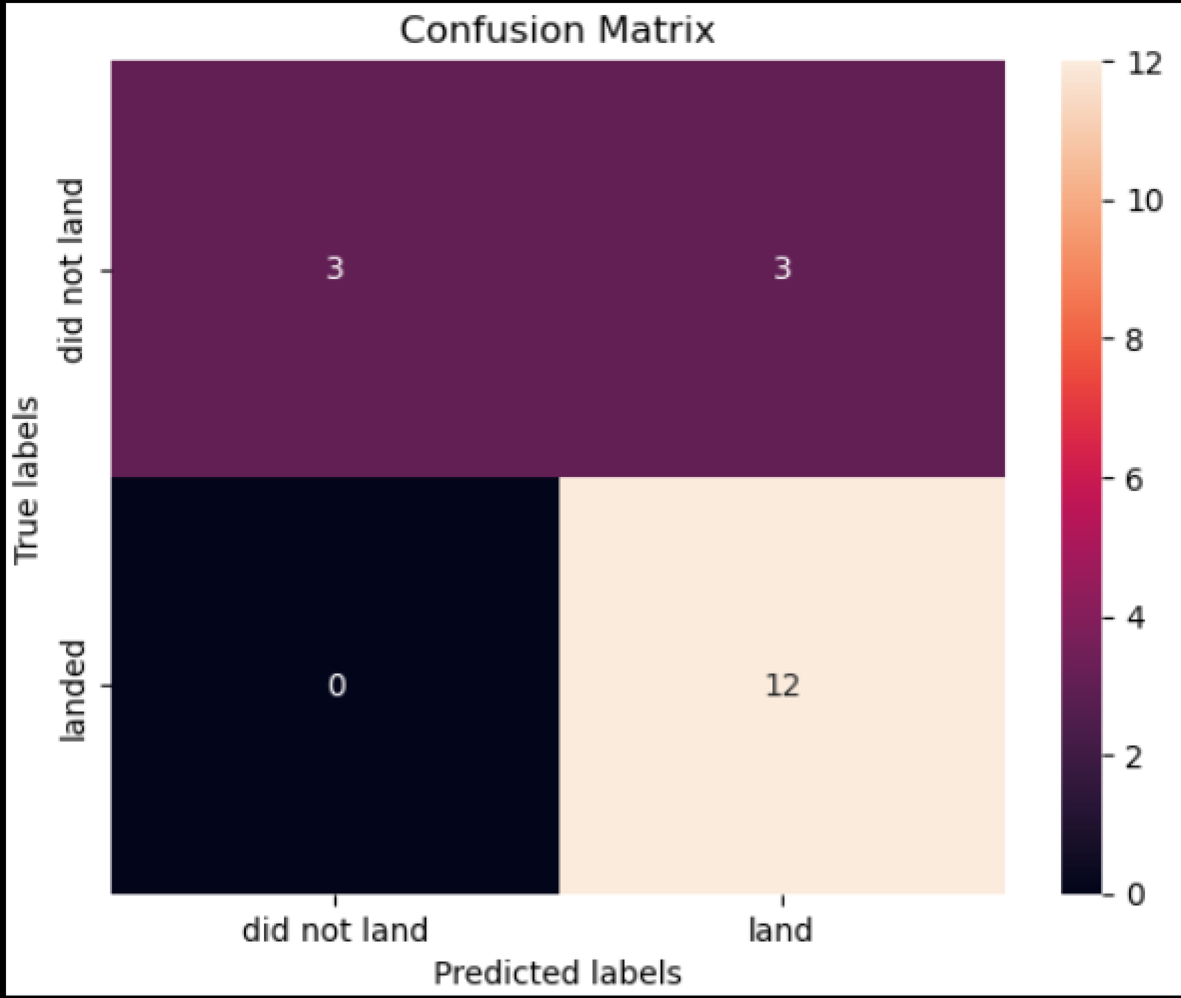
- The Decision Tree model came out on top in terms of classification accuracy on the test data. It achieved an accuracy of 0.9444.
- In comparison, the Logistic Regression, Support Vector Machine, and K-Nearest Neighbors models all achieved the same accuracy of 0.8333.
- The conclusion drawn here is that, for this particular dataset, the Decision Tree model appears to be the most suitable as it demonstrated a higher ability to correctly classify the instances in the test set compared to the other models.



Confusion Matrix

Key Insights from the Explanation:

- **High Accuracy (94.44%):** As we saw before, the model achieved a high accuracy. This is supported by the significant number of true positives (correctly predicting landings) and true negatives (correctly predicting non-landings).
- **No False Negatives:** This is a crucial observation, especially in aerospace operations. A false negative would mean the model predicted a landing failure when it was actually successful. The absence of false negatives indicates the model is very reliable in identifying successful landings, which is vital for safety and mission assurance.
- **Manageable False Positives:** There is 1 false positive. This means the model predicted a landing when it didn't happen. The explanation suggests that over-preparation due to a false positive is more manageable than under-preparation caused by a false negative in this context. For practical applications in aerospace, it's often better to err on the side of caution.
- **Balanced Performance with a Slight Bias:** The model shows a balanced performance but with a slight bias towards predicting successful landings. This aligns well with the practical needs in the aerospace industry, where ensuring successful landings is paramount for cost estimation and planning.



Conclusions

•**Launch Site Advantage:** The "CCAFS LC-40" launch site stands out with the highest success rate among all sites analyzed. It accounted for a significant 43.7% of all successful launches. This suggests that this particular launch site might have favorable conditions or well-optimized processes that contribute to a higher likelihood of a successful launch.

•**Booster Reliability:** The "FT" booster version demonstrates a high success rate across a range of different payload masses. This indicates that the "FT" booster is a reliable and robust option compared to other booster versions considered in the analysis. The implication here is that future missions could potentially benefit from using the "FT" booster to increase their chances of success.

•**Payload Mass Independence:** The analysis did not reveal any clear link between higher payload masses and lower success rates. This suggests that the payload mass, within the range considered, might not be a primary factor influencing the outcome of a launch. Instead, other factors such as the launch site conditions and the booster version used appear to play a more significant role in determining whether a launch is successful.

- **Power of Interactive Visualizations:** The use of interactive data visualization tools like Folium and Plotly Dash proved to be highly beneficial. These tools offered valuable insights into the geographical and operational patterns of SpaceX launches. By enabling stakeholders to explore the data visually and interactively, they facilitated a deeper understanding and supported more informed decision-making based on comprehensive visual analytics.

Overall Conclusion:

- The predictive analysis, combined with the interactive visualizations, has successfully illuminated key factors that influence SpaceX's launch success. Furthermore, this work has established a robust framework that can be used for future assessments and decision-making within the aerospace industry. The insights gained from this analysis have the potential to contribute to the improvement of launch strategies and further advance the ongoing success of reusable rocket technology.

THANK YOU