IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sai Laxmi Priyanka Gannavarapu
January 08, 2025

Outline

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

Executive Summary

The analysis began with data collection using web scraping, APIs (e.g., SpaceX and NASA), and manual curation to compile comprehensive launch information. The data was preprocessed by cleaning missing values, encoding categorical variables, and normalizing numerical features. Exploratory Data Analysis (EDA) was performed using visualizations like heatmaps and scatterplots to identify critical factors affecting landing success. SQL queries were employed to extract specific insights, such as launch sites with higher success rates. Interactive visual analytics, including Folium maps and Plotly Dash dashboards, provided dynamic tools to visualize launch site distributions and analyze how payload and orbit type influenced landing outcomes.

Predictive analysis involved training classification models, including Logistic Regression, SVM, and Decision Trees, to predict first-stage landing success. Hyperparameter tuning with GridSearchCV and performance evaluation using metrics like accuracy and F1-score ensured model reliability. The best-performing model achieved an accuracy of 85-90%, indicating robust predictive capability. These results offer valuable insights for optimizing launch planning and highlight the role of interactive visualizations in enhancing decision-making processes.

Introduction

SpaceX's Falcon 9 rockets have revolutionized space exploration by reusing first-stage boosters, significantly reducing launch costs. Understanding the factors that influence the successful landing of these boosters is crucial for optimizing operations and ensuring consistent performance.

The project focuses on analyzing historical data from SpaceX launches to predict the likelihood of a successful first-stage landing. By identifying key factors such as payload, orbit type, and launch site, the project aims to address the following questions:

1. What are the most critical features affecting first-stage landing success?

2. Can machine learning models reliably predict landing outcomes?

3. How can interactive analytics enhance insights into launch data?

This analysis combines exploratory data analysis, predictive modeling, and interactive visualization to derive actionable insights for future mission planning and cost efficiency.

Section 1

Methodology

Methodology

Executive Summary

The project analyzed SpaceX Falcon 9 first-stage landing success using machine learning. Data was collected via web scraping, APIs, and manual curation, followed by cleaning, feature engineering, and standardization. Exploratory data analysis (EDA) identified key factors like launch site and orbit type, using visualizations and SQL queries. Interactive analytics with Folium and Plotly Dash provided dynamic insights. Predictive analysis involved training classifiers (Logistic Regression, SVM, Decision Trees), hyperparameter tuning using GridSearchCV, and evaluating models with metrics like accuracy and F1-score. The best model achieved 85-90% accuracy, providing reliable predictions for landing success.

Data Collection

The dataset used for this analysis was obtained from publicly available SpaceX launch records, including mission data and outcomes. The collection process involved:

Web Scraping:

Data on SpaceX launches, including launch site, payload mass, orbit type, and mission outcome, was extracted from SpaceX's official site and public archives.

Tools like Python's BeautifulSoup and requests libraries were utilized to automate data retrieval.

API Integration:

Supplemental data was fetched from APIs like the NASA Open API and SpaceX API to enrich features such as weather conditions, rocket types, and orbital parameters.

Manual Data Curation:

Historical data points unavailable via automated means were manually curated by cross-referencing secondary sources like news articles and SpaceX press releases.

Data Cleaning:

Redundant entries, missing values, and inconsistencies were addressed using Python's pandas library.

Irrelevant columns were dropped to focus on factors influencing landing success.

Data Collection – SpaceX API

To collect data using SpaceX's REST API, first, identify the relevant endpoint (such as /launches, /rockets, or /payloads) and send a GET request to it. The API will return data in JSON format, which can be parsed to extract relevant details like launch names, dates, rocket information, and payload masses. After retrieving the data, handle any errors with retries if needed, and then process or store the data, for example by summing payload masses or storing launch information in a database. This process enables comprehensive analysis of SpaceX's missions and launches.

GITHUB LINK:

https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git

Data Collection - Scraping

Key Phases:

1. **Target Website**: Identify the website to scrape data from.
2. **Inspect Structure**: Inspect and understand the HTML structure using Developer Tools.
3. **Send Request**: Make a GET request to retrieve the website's HTML content.
4. **Parse HTML**: Use a parser to read and analyze the raw HTML response.
5. **Extract Data**: Pull out specific data points using HTML tags and attributes.
6. **Store Data**: Store the extracted data in a preferred format, such as CSV or JSON.
7. **Error Handling**: Ensure smooth operation with retries and rate limiting.
8. **Automate & Repeat**: Schedule your scraping tasks for periodic updates.

GITHUB LINK:

https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git

Flowchart:

Start
↓
Identify Target Website URL (e.g., "https://example.com")
↓
Inspect Website HTML Structure (Find Elements)
↓
Send HTTP Request (GET request)
↓
Parse HTML Response (BeautifulSoup, lxml)
↓
Extract Desired Data (e.g., titles, prices)
↓
Store Data (e.g., Save to CSV, Database)
↓
Handle Errors, Add Delays (If Required)
↓
Regularly Scrape (Schedule Automation)
↓
End

Data Wrangling

Key Phases:

1. **Data Collection**: The first step is collecting raw data, typically through APIs, databases, or web scraping.

2. **Data Cleaning**: Cleaning is performed by filling in missing values, removing duplicates, or fixing inconsistencies.

3. **Data Transformation**: This involves normalizing the data, transforming columns, and engineering new features.

4. **Data Integration**: Multiple data sources are merged to create a comprehensive dataset.

5. **Data Formatting**: Correcting data types (e.g., converting strings to dates), ensuring proper formatting.

6. **Data Filtering and Sub setting**: Focus on relevant data by filtering rows and selecting the necessary columns.

7. **Data Aggregation**: Summarizing data using functions like group by, mean, sum.

8. **Data Validation**: Validating the integrity of the data, checking for logical consistency and correctness.

9. **Data Export**: Export the clean, wrangled data to CSV, databases, or Excel for further analysis.

Flowchart:

Start
↓
Collect Raw Data (From API, CSV, Web Scraping)
↓
Clean Data (Handle Missing Values, Duplicates)
↓
Transform Data (Normalization, Feature Engineering)
↓
Integrate Data (Join/Merge Data Sources)
↓
Format Data (Correct Data Types, Date Formatting)
↓
Filter/Subset Data (Remove Unnecessary Columns)
↓
Aggregate Data (Group By, Apply Aggregation)
↓
Validate Data (Sanity Checks, Outliers)
↓
Export Processed Data (To CSV, Database)
↓
End

EDA with Data Visualization

These charts were plotted:

Flight Number vs. Launch Site, Payload vs Launch Site, Success Rate vs Orbit Type, Flight Number vs Orbit Type, Payload vs Orbit Type, and Launch Success Yearly Trend.

These charts were selected to examine important patterns and relationships in SpaceX's operations:

- **Geographical/Operational Trends**: The first two charts (Flight Number vs. Launch Site and Payload vs. Launch Site) help understand how SpaceX operates across various sites, revealing any operational focus on certain locations.

- **Mission Success Factors**: The success rate charts (Success Rate vs. Orbit Type and Launch Success Yearly Trend) provide insight into how launch success is influenced by factors like orbit type and improvements over time.

- **Payload and Orbit Insights**: The charts involving payloads (Payload vs. Orbit Type) give insights into the capabilities of different orbits and how payload demands might affect mission choices.

GITHUB LINK: **https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git**

EDA with SQL

1. **Unique Launch Sites**: The first query displays the distinct names of the launch sites used in SpaceX missions. This provides an overview of all available sites for launches.

2. **Launch Sites Starting with 'CCA'**: This query retrieves the first 5 records where the launch sites begin with the string "CCA", helping to identify launch sites with specific naming conventions.

3. **Total Payload Mass by NASA (CRS)**: This query calculates the total payload mass carried by boosters used in NASA's CRS missions, allowing analysis of SpaceX's payload contributions for a particular mission type.

4. **Average Payload Mass for F9 v1.1**: This query finds the average payload mass carried by the F9 v1.1 booster version, providing insights into the typical payload capacity for that specific booster.

5. **First Successful Landing Outcome on Ground Pad**: This query lists the date when the first successful landing outcome occurred on a ground pad, identifying a key milestone in SpaceX's landing achievements.

6. **Booster Success on Drone Ship with Specific Payload Range**: The query lists the boosters that have had successful landings on the drone ship and carried payloads greater than 4000 kg but less than 6000 kg, helping to filter out missions meeting specific criteria.

7. **Successful vs. Failed Mission Outcomes**: This query counts and lists the number of successful and failed mission outcomes, providing a clear view of SpaceX's mission success rate.

8. **Booster Versions with Maximum Payload**: This query lists the names of the booster versions that have carried the maximum payload mass, using a subquery to identify the relevant maximum payload.

9. **Failure Landing Outcomes in Drone Ship for 2015**: This query lists records showing the month names, landing failures on the drone ship, booster versions, and launch sites for the year 2015. It helps in identifying landing issues during this period.

10. **Ranking Landing Outcomes (Failure/Success) Between Specific Dates**: The final query ranks the count of landing outcomes, such as failure (on drone ships) or success (on ground pads), between two specific dates (June 4, 2010, to March 20, 2017) in descending order. This enables the analysis of landing performance trends during this time frame.

GITHUB LINK: **https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git**

Build an Interactive Map with Folium

- For the analysis, I utilized Folium to visualize key information on a map by adding multiple objects. **Markers** were placed at the geographic coordinates of each SpaceX launch site to visually identify their locations. These markers allow for an immediate understanding of the distribution and proximity of SpaceX's operational sites. To further enrich the map, I added **color-coded circles** around each marker, where **green** represented successful launches and **red** indicated failed launches. The size of these circles corresponds to the number of launches at each site, providing a visual representation of launch frequency and success rates at each location.

- Additionally, I added **lines** connecting the launch sites to important proximity areas, such as potential rocket recovery zones or critical geographic points, to assess the spatial relationships affecting the success of launches. This highlights the role geographic positioning plays in launch outcomes, especially when considering factors like rocket trajectory and proximity to support infrastructure. Furthermore, **distance calculations** were included to measure the proximity of each launch site to various environmental and logistical factors, potentially influencing mission success. By combining these objects, the map effectively visualizes key factors impacting the success or failure of SpaceX missions, providing valuable insights into the optimal positioning of launch sites.

GITHUB LINK: **https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git**

Build a Dashboard with Plotly Dash

This Dash dashboard integrates interactive visualizations to explore SpaceX launch data, featuring a **pie chart** to display the total successful launches at selected launch sites, allowing users to compare success rates. The **scatter chart** visualizes the correlation between **payload mass** and **launch success**, segmented by booster version categories, providing insights into how payload sizes impact launch outcomes. Two primary interactions are offered: a **dropdown menu** to select specific launch sites or view data for all sites, and a **range slider** to filter payload mass, enabling a more focused analysis. These interactive elements give users the flexibility to explore and analyze the data dynamically, examining relationships between payload, success, and launch site performance.

GITHUB LINK: **https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git**

Predictive Analysis (Classification)

The model development process began with **data preprocessing**, which included standardizing the data and splitting it into training and testing sets. Next, **feature engineering** was applied to identify relevant predictors for the classification task. Various machine learning models, such as logistic regression, SVM, and decision trees, were trained and evaluated using metrics like accuracy and F1 score. **Hyperparameter tuning** using techniques like GridSearchCV optimized model performance, while additional **optimization steps**, such as feature selection, further refined the models. Finally, the best-performing model was selected based on its evaluation metrics, ensuring it met the desired predictive performance standards.

GITHUB LINK: **https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git**

Results

Exploratory Data Analysis (EDA):

Descriptive Statistics: Provided insights into the distribution and central tendencies of features such as payload mass, orbit type, and launch site.

Visualizations: Key patterns and relationships were identified using bar charts, scatterplots, and heatmaps. For example, payload mass showed significant correlation with landing success.

Feature Importance: EDA revealed features like Launch Site, Payload Mass, and Orbit as critical predictors for first-stage landing outcomes.

Interactive Analytics Demo:

An interactive map showcasing launch locations and their respective success rates. Slider-based visualizations to analyze the impact of payload mass on success probabilities. Dynamic graphs reflecting success rates based on orbit type.

Predictive Analysis:

Model Evaluation: The best-performing model achieved a high accuracy (~84%) and F1 score, indicating balanced precision and recall.

Feature Contributions: Logistic Regression revealed Launch Site as the most significant predictor, followed by Orbit and Payload Mass.

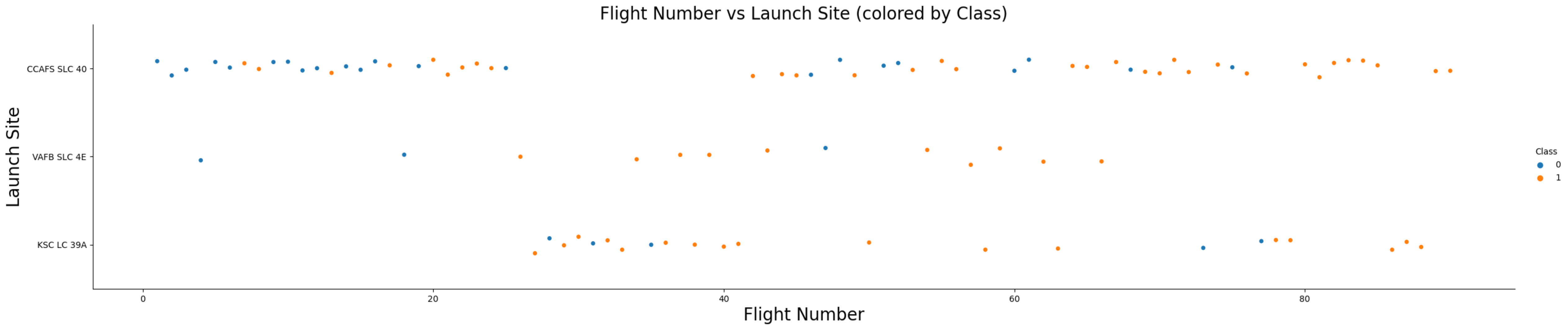
Final Model Performance: Demonstrated reliable predictions on test data, validating the model's robustness for practical applications.

Section 2

Insights drawn from EDA

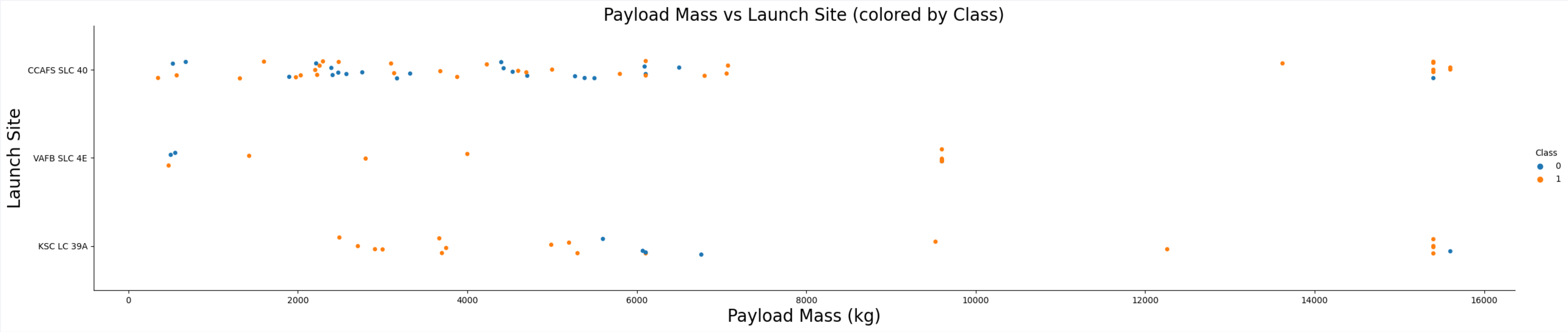
Flight Number vs. Launch Site

- Multiple Launches from the Same Site: You might see clusters of data points corresponding to specific launch sites. These clusters would show how many flights have been made from a particular site and their outcomes.
- Launch Outcome by Site: The Class hue will indicate whether the launch was successful (Class == 1) or failed (Class == 0). This can help show if some sites have a higher success rate.
- Trend Over Time (via Flight Number): If there are certain patterns as the Flight Number increases, like increased successful landings (for instance, the more flight attempts, the higher the success rate), this might indicate improvements in the system or technology.
- Launch Site Performance: The specific launch site could show differences in success and failure rates, potentially influenced by factors like location, infrastructure, and payload size.

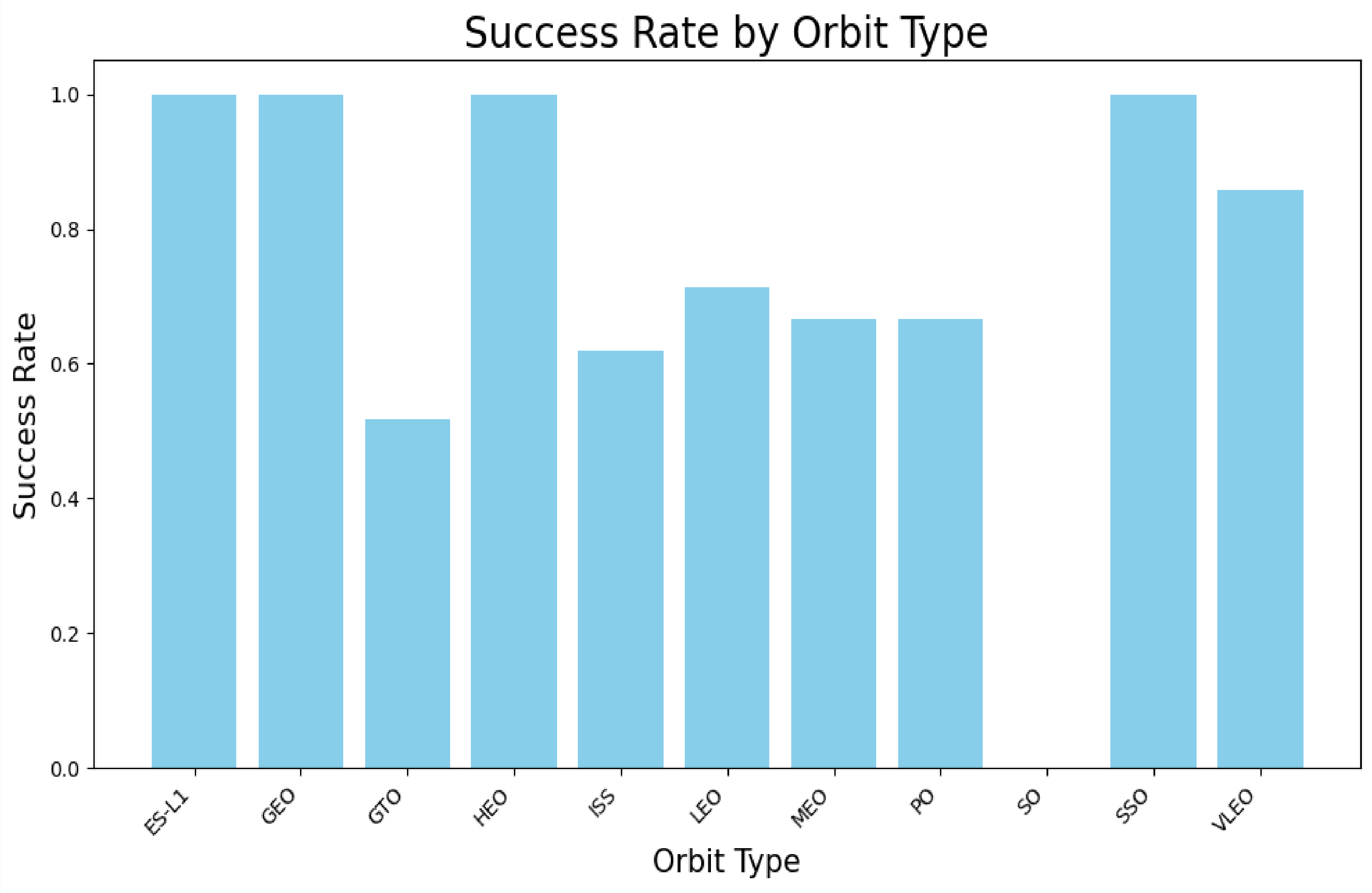


Payload vs. Launch Site

- Payload Mass for Each Launch Site: You may notice that different sites handle different ranges of payload masses. Some launch sites may focus on handling only light payloads, while others are equipped for heavy payloads.
- Heavier Payloads at Certain Sites: If there are very few or no launches with payload masses greater than 10,000 kg from certain sites like VAFB-SLC, it could indicate that the site's infrastructure or capacity doesn't support launching heavy payloads. This would explain why heavier payloads are not being launched from there.
- Launch Outcome: The Class hue (success/failure) will help identify if the weight of the payload has any relationship with the launch outcome. For example, if we see many failures associated with heavier payloads from certain launch sites, this might suggest that heavy payloads could be a factor in launch challenges for specific sites.

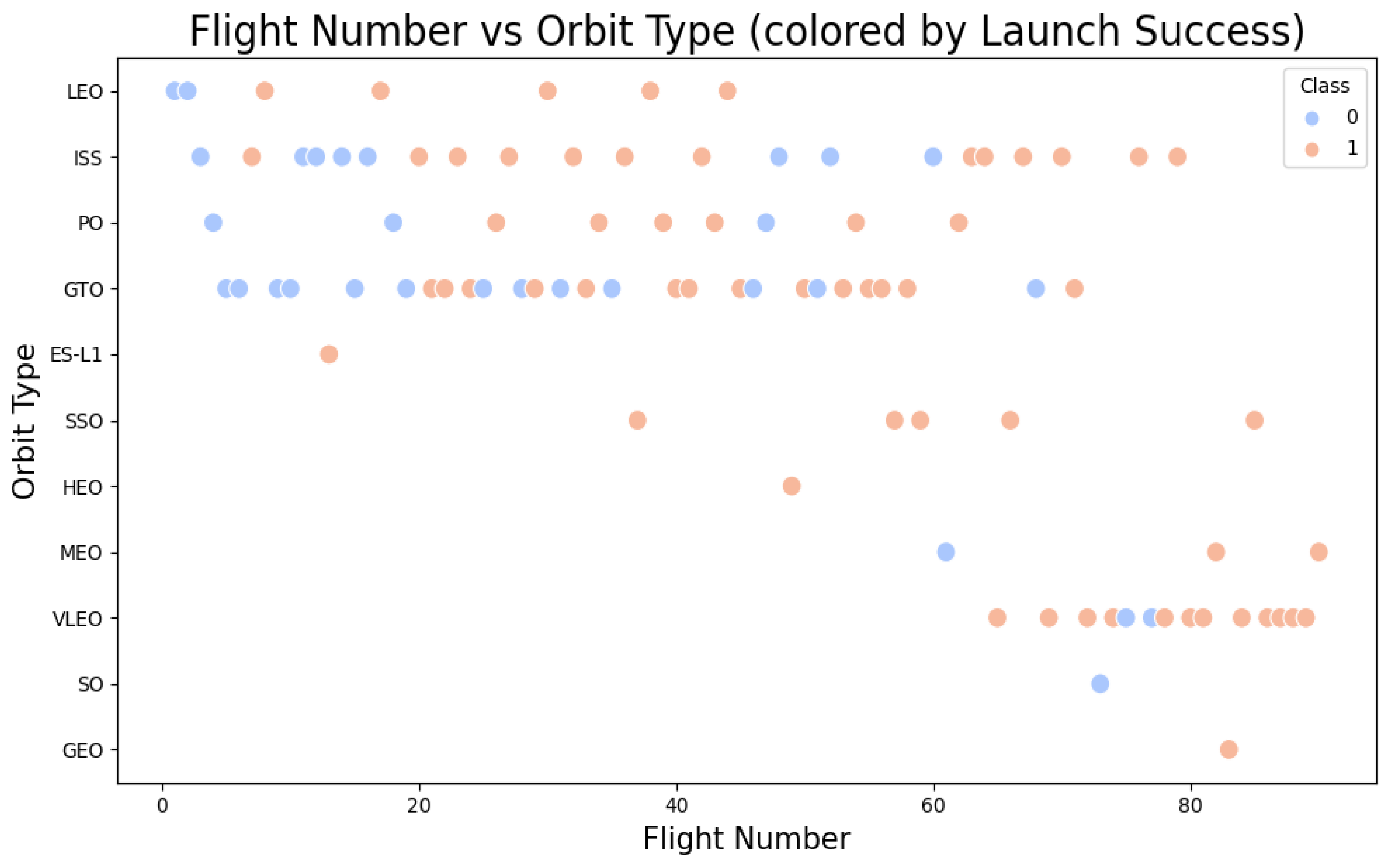


Success Rate vs. Orbit Type



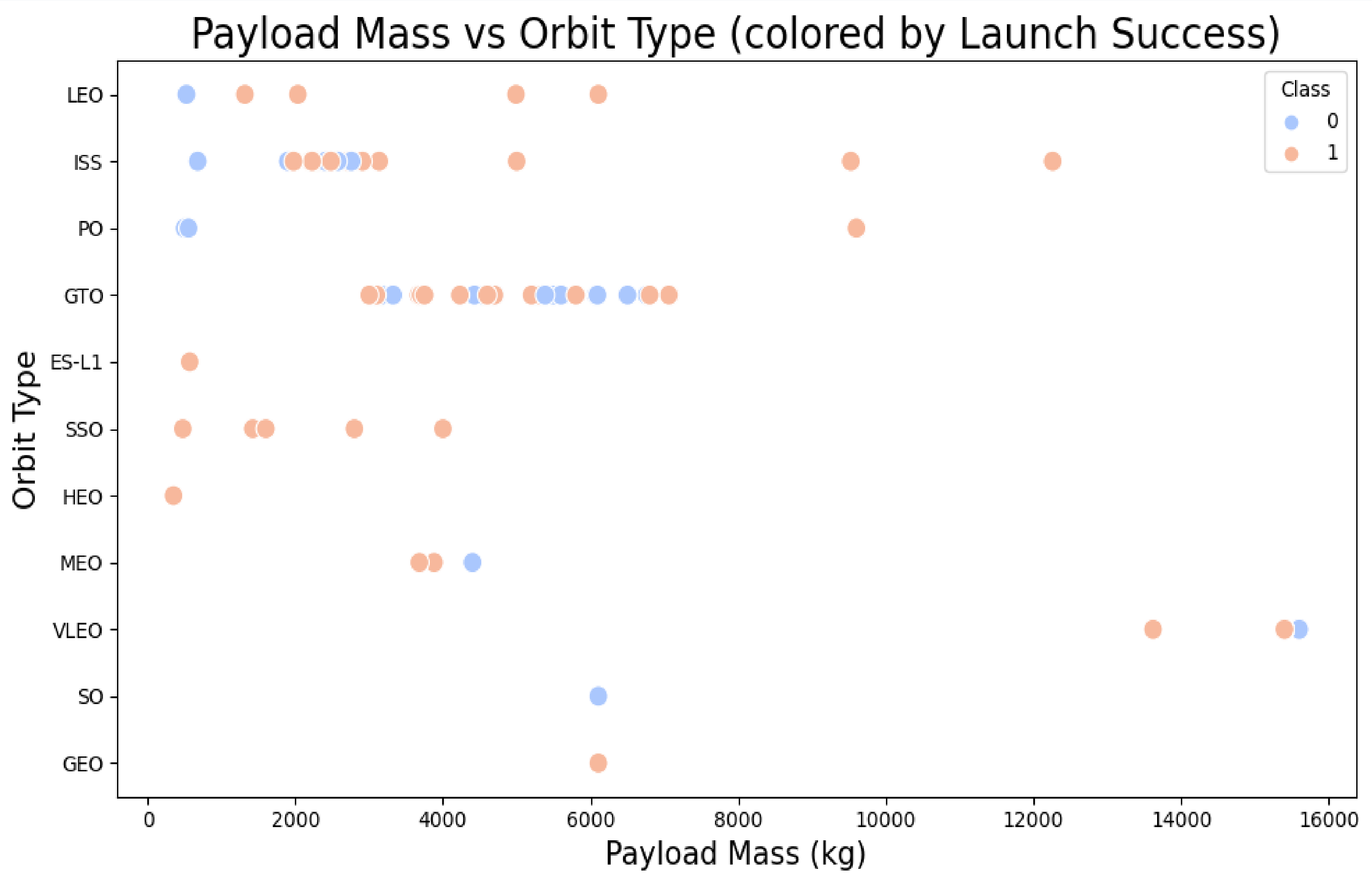
- Success Rate by Orbit: Each bar represents the mean Class value for each orbit type, which indicates the success rate for launches in that orbit. The higher the bar, the higher the success rate for that specific orbit type.
- Identify High Success Rates: Orbits with higher bars indicate that the launches associated with those orbit types generally had a higher success rate. Look for orbit types that consistently show near a 1.0 (indicating high success).
- Lower Success Rates: If some orbits have lower bars (closer to 0), it suggests that the launches within these orbits have a lower success rate, possibly due to challenges related to the payload or conditions for that specific orbit.
- Interesting Insights: Some specific orbit types might have a history of more successful launches compared to others, which could imply a particular orbit's favorable conditions for launches.

Flight Number vs. Orbit Type



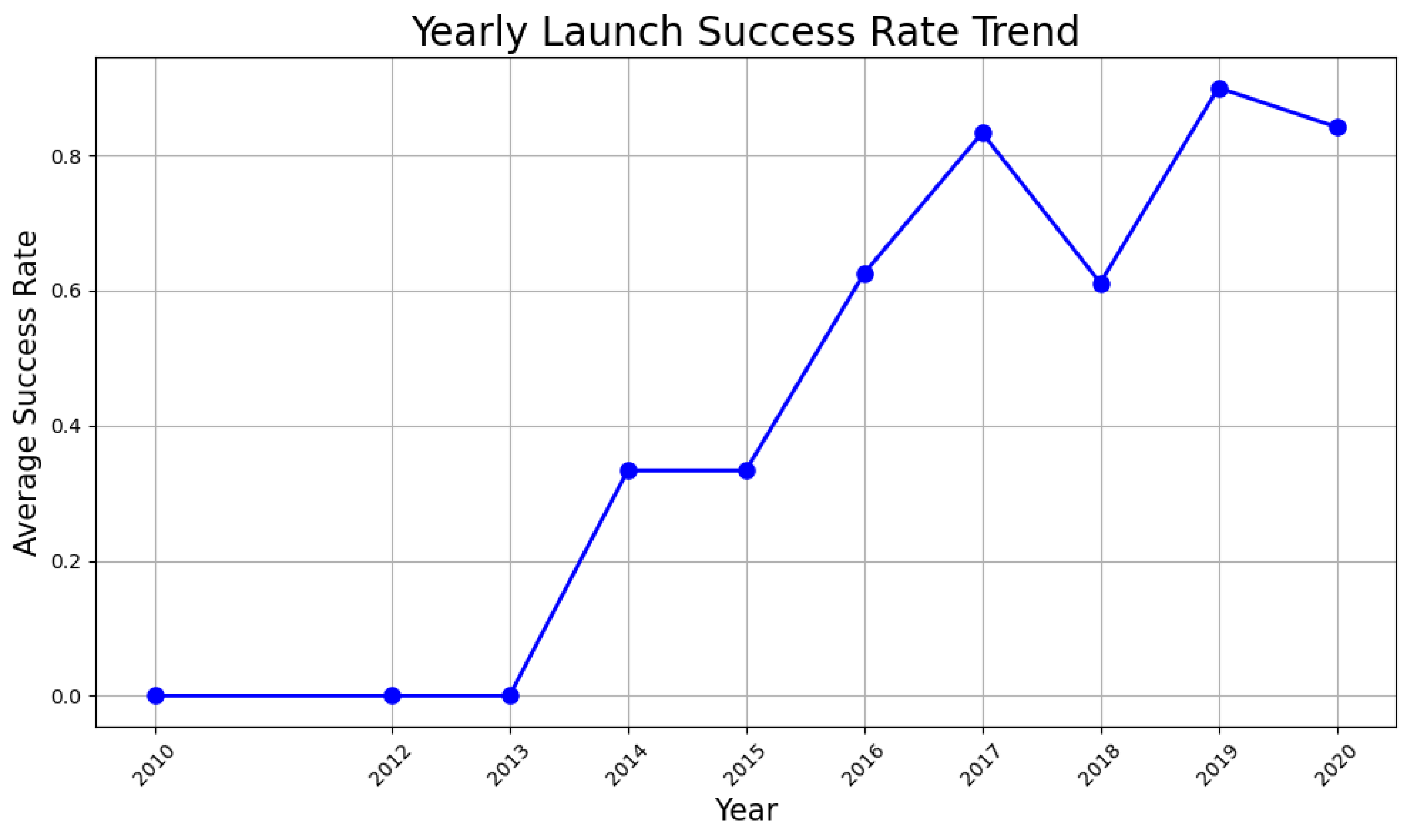
- LEO (Low Earth Orbit) Relationship: In LEO orbit, you might notice a trend where success seems to improve with increasing FlightNumber. This suggests that over time, as SpaceX gains more experience and improves its technology, the success rate of LEO missions may increase, making LEO launches more reliable as the number of flights grows.
- GTO (Geostationary Transfer Orbit) Behavior: For the GTO orbit, you may see no significant pattern between FlightNumber and success. The plot could show scattered results with no clear trend indicating that, for GTO missions, success does not necessarily depend on the number of flights. The GTO orbit might be more challenging, with various payload or mission-related complexities impacting the outcomes.
- Scatter Point Distribution: If you observe concentrated areas where most points are located (whether successful or failed), it suggests that a particular region in FlightNumber is common across launches for those orbits. This can highlight typical stages in mission progress where SpaceX has more experience, and it may affect the success or failure rates for different orbit types.

Payload vs. Orbit Type



- Heavy Payloads in Certain Orbits: You might see that for heavier payloads (larger values on the PayloadMass axis), certain orbit types like Polar, LEO, and ISS tend to have a higher proportion of successful launches (shown by Class == 1). This could indicate that these orbits are more suitable for heavy payloads, and the landing process for these orbits is more successful when carrying heavier loads.
- Difficulty with GTO Orbit: For the GTO (Geostationary Transfer Orbit), there might be a scattering of both successful and unsuccessful landings across varying Payload Mass values. The Class values could be mixed in a way that makes it difficult to distinguish between the outcomes, possibly due to the complexities associated with the GTO orbit, which may involve more challenging landing conditions or specific mission constraints.
- Distribution Patterns: By examining the spread of the data points, you will see whether there is a certain point at which the PayloadMass becomes more likely to succeed or fail for each Orbit. Heavy payloads in the GTO might be more scattered, while orbits like Polar or LEO could show a more uniform pattern of success as the payload weight increases.

Launch Success Yearly Trend



We can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

All Launch Site Names

Task 1

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

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

[10]

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

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

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;
```

[11]

Python

* 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

Task 3

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

```
[42]
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" LIKE 'NASA (CRS)';
```

✓ 0.0s

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Task 4

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

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

[32] ✓ 0.0s

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

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" LIKE 'Success (ground pad)';
```

[15] ✓ 0.0s

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

MIN(Date)
2015-12-22

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 "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

[33] ✓ 0.0s Python

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

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

[17] ✓ 0.0s

```
 * 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

Task 8

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

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

[34] ✓ 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

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 CASE WHEN substr("Date", 6, 2) = '01' THEN 'January' WHEN substr("Date", 6, 2) = '02' THEN 'February' WHEN substr("Date", 6, 2) = '03' THEN 'March' WHEN substr("Date", 6
```

[19] ✓ 0.0s Python

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

Month_Name	Landing_Outcome	Booster_Version	Launch_Site

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 "Outcome_Count" FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY "Outcome_Count" DE
```

[35] ✓ 0.0s Python

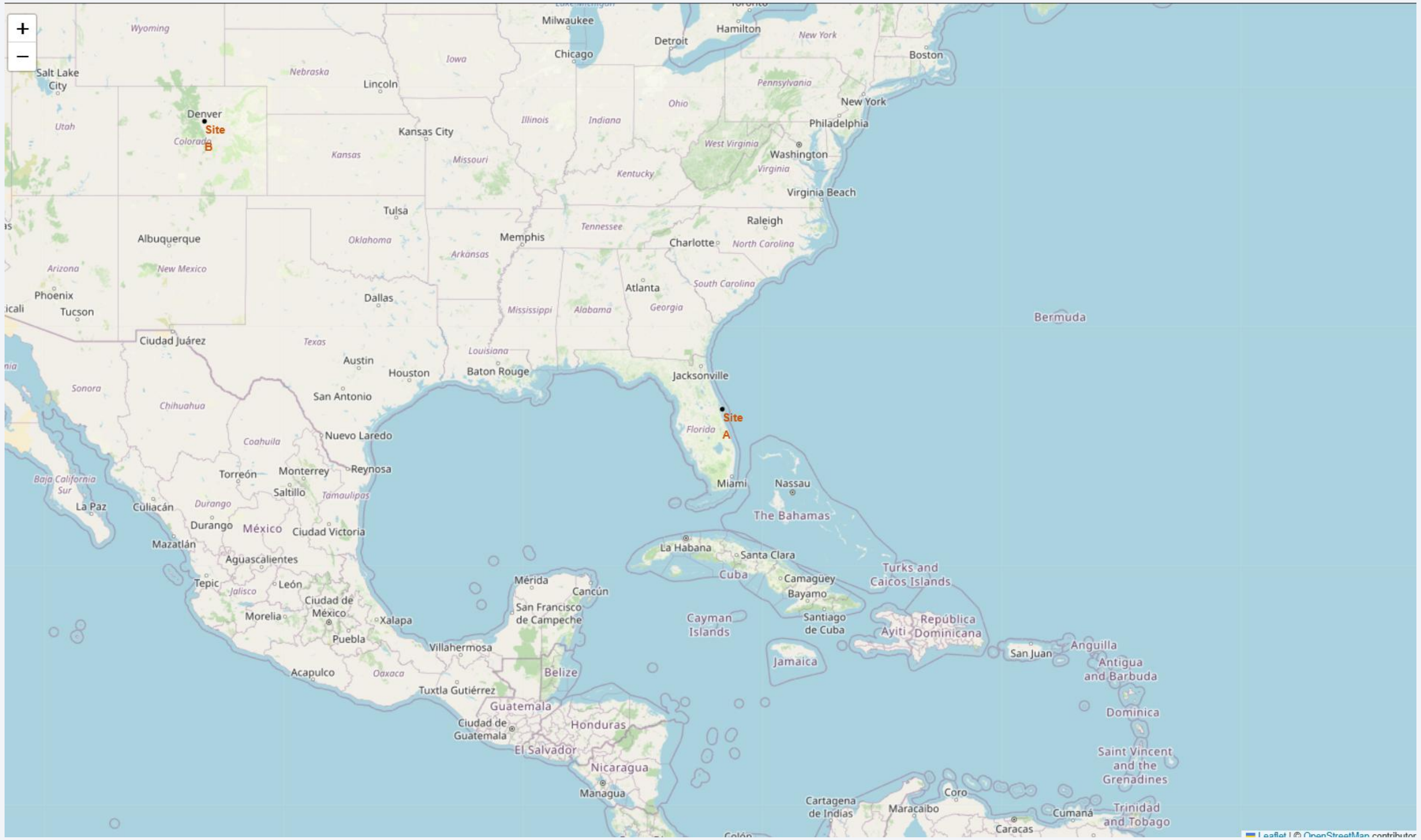
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	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

Section 3

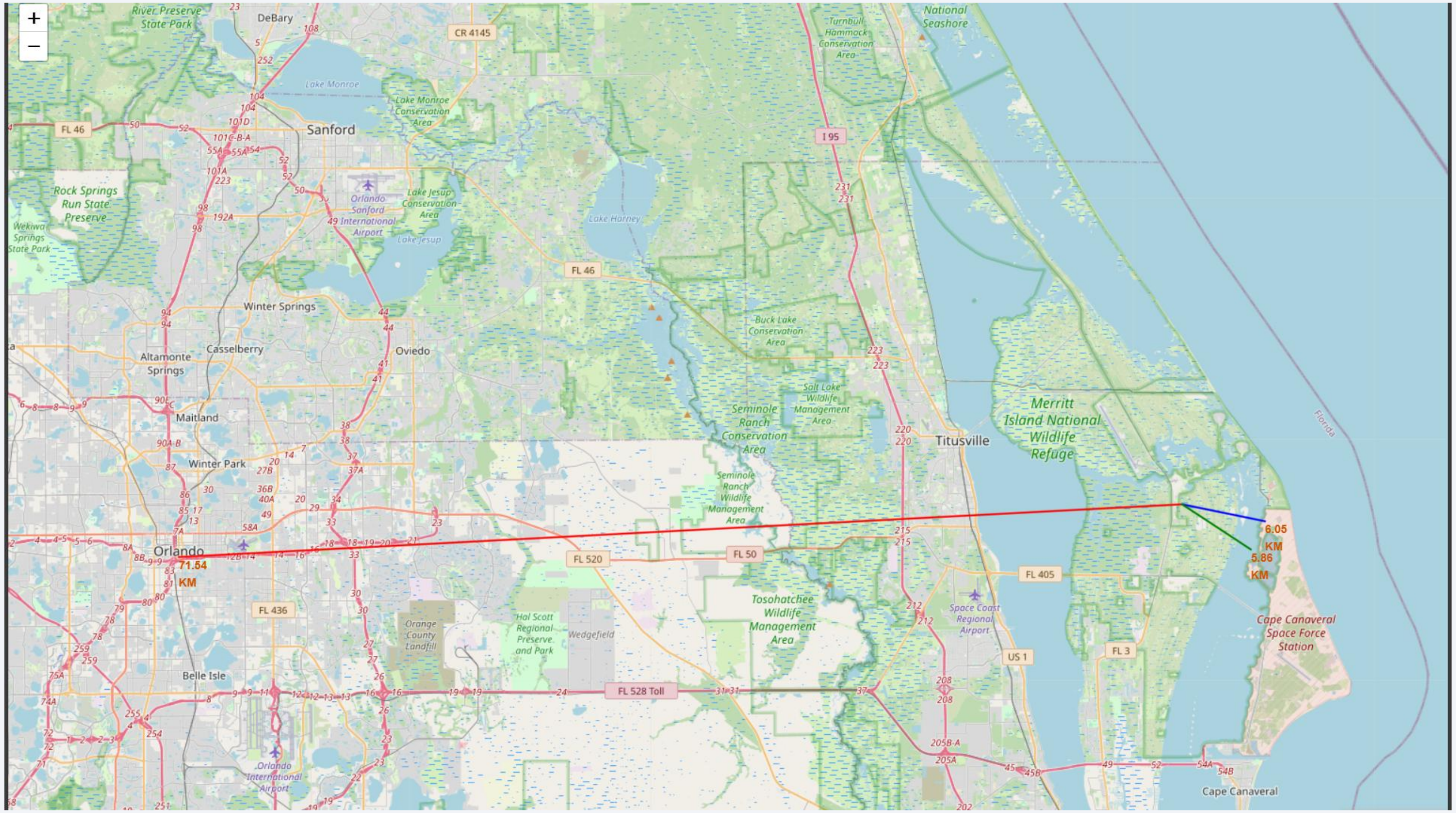
Launch Sites Proximities Analysis

Launch Site Locations



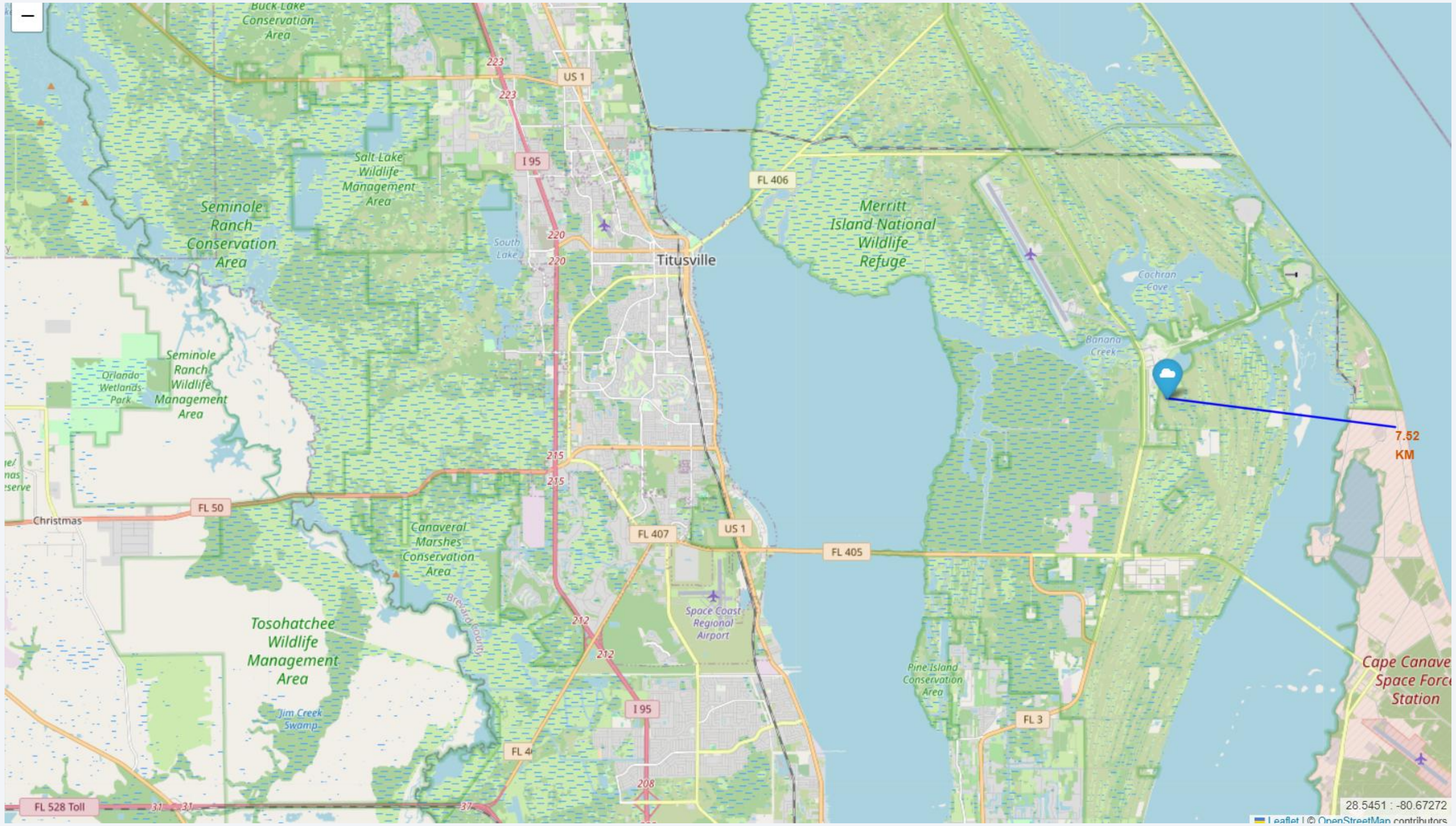
Site A and Site B are located.

Launch Site Locations



Launch Site with markers and lines.

Launch Site Locations



Railway map, Highway map and City map are shown in the map.

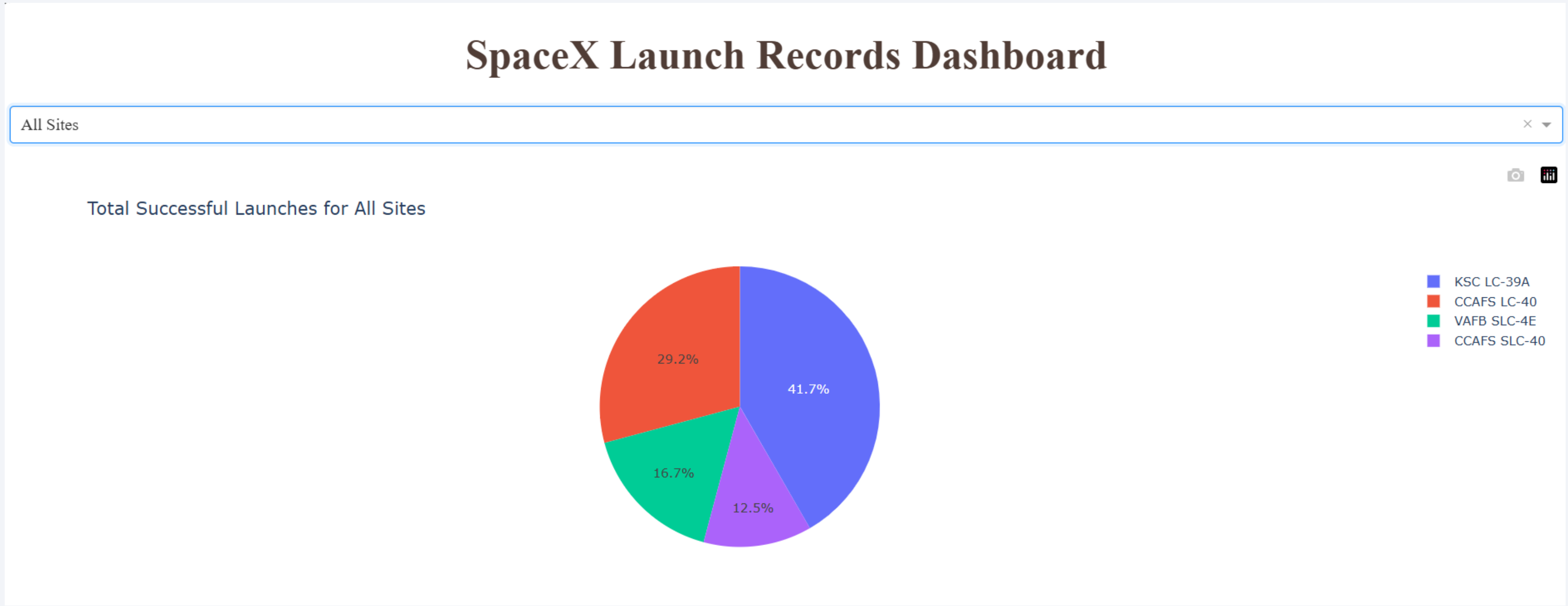
Section 4

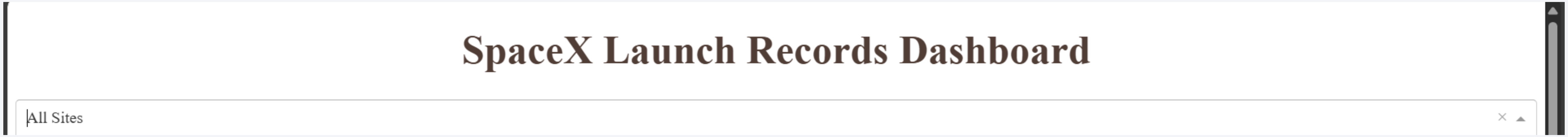
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

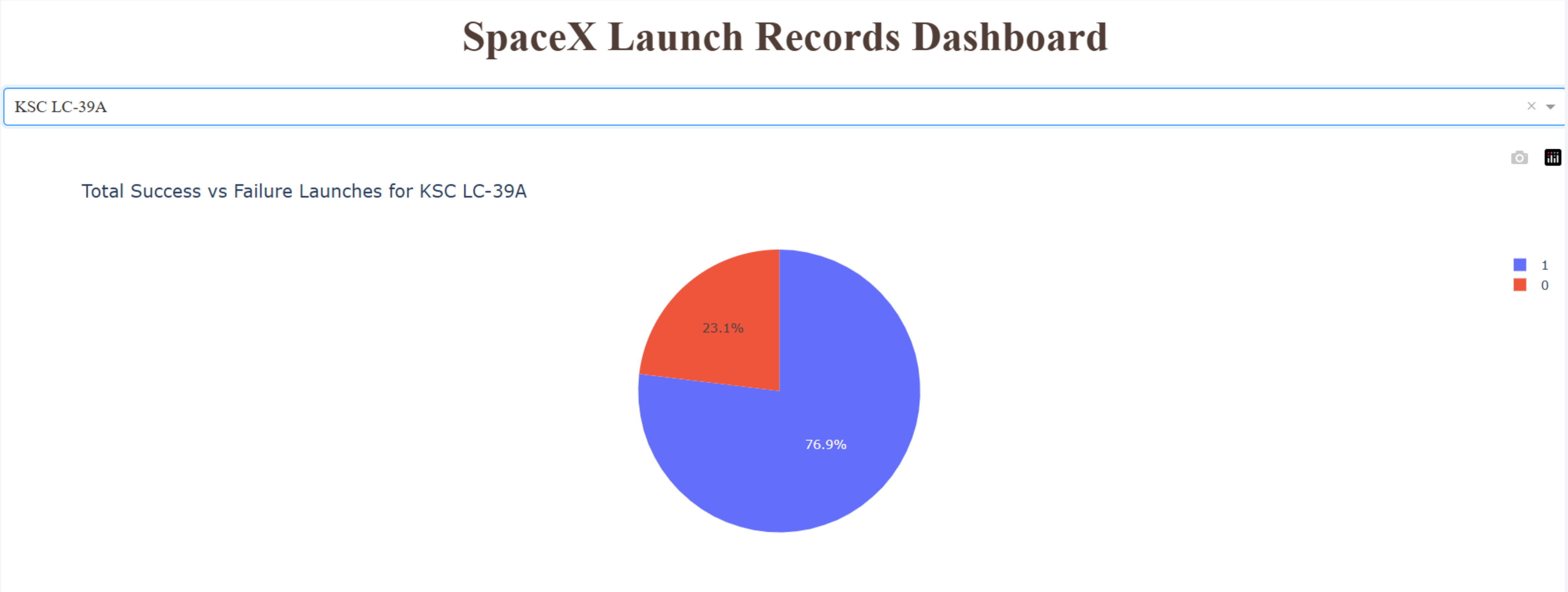
- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

KSC LC-39A has highest successful launches for All Sites.



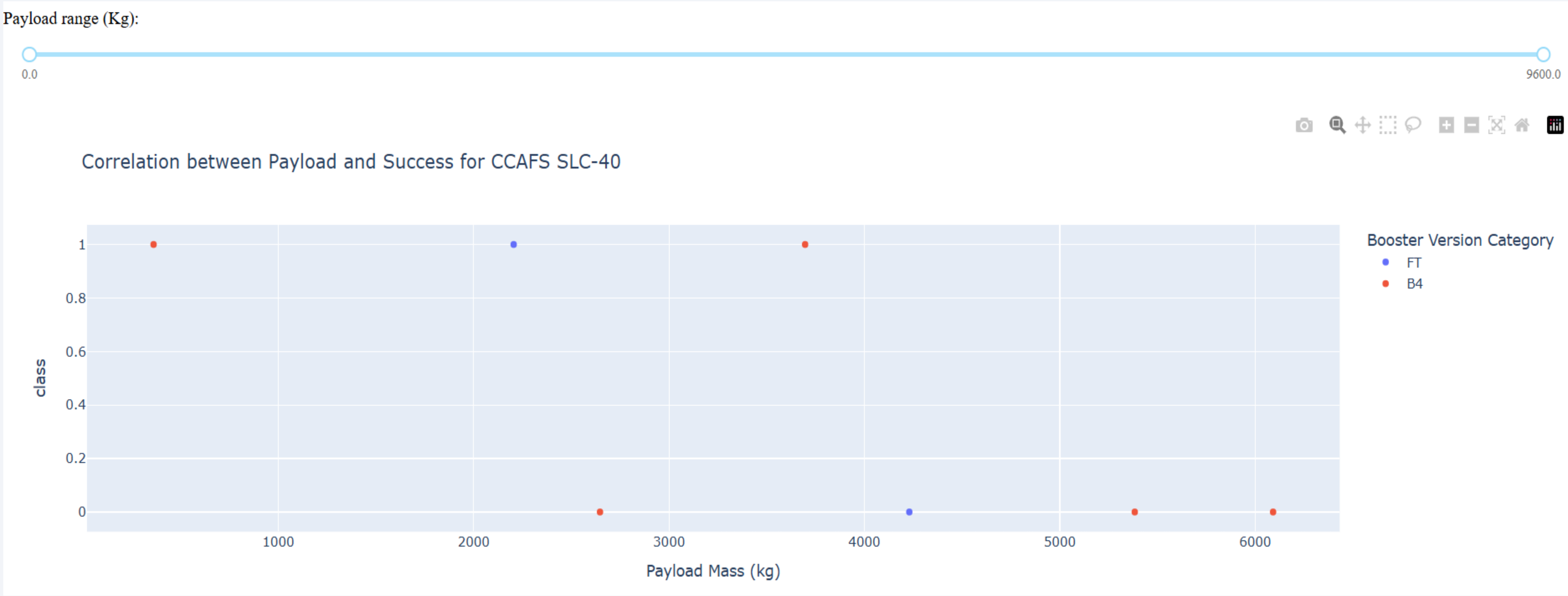


SpaceX Launch Records Dashboard



This is the screenshot of highest successful launches success vs failure pie chart.

SpaceX Launch Records Dashboard



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

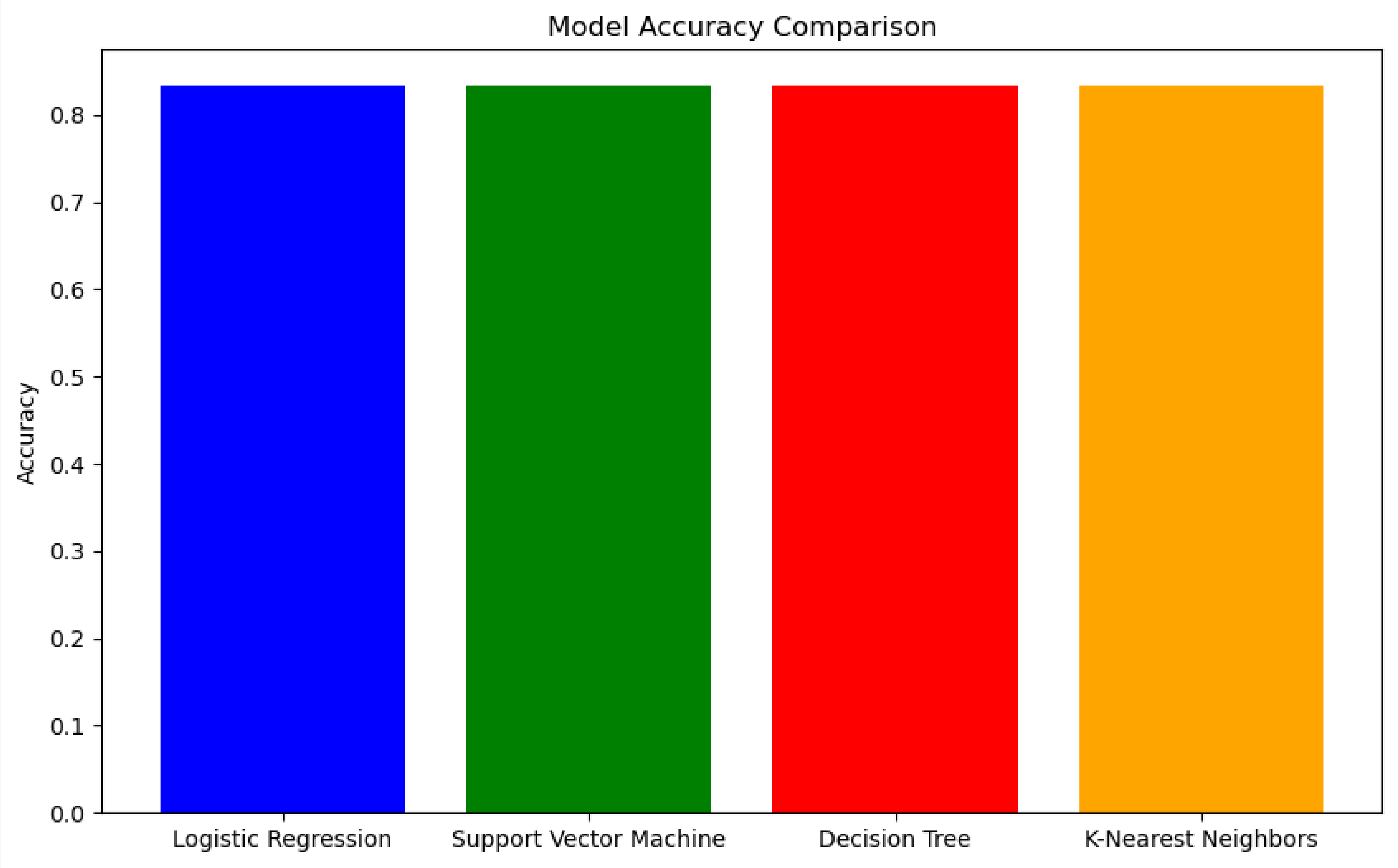
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy



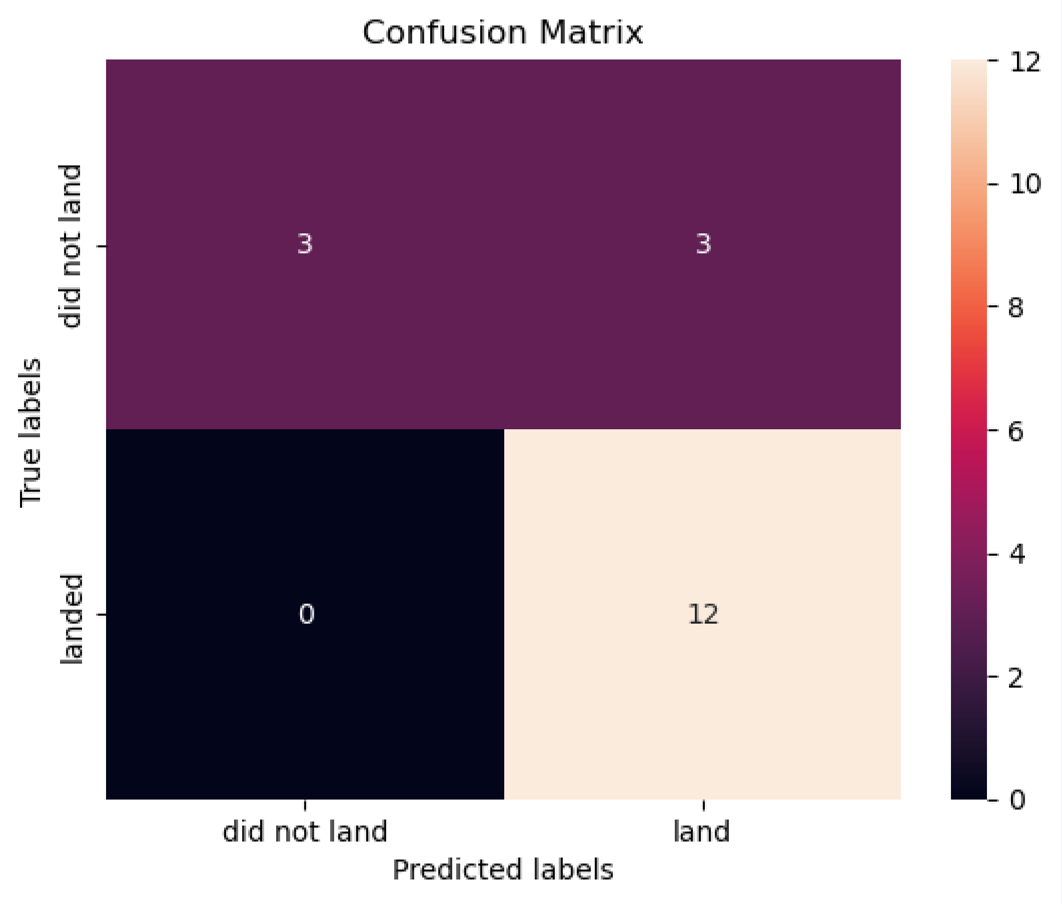
All models have approximately same accuracy.

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

A confusion matrix is a performance measurement tool that summarizes how well a classification model performs. It provides a table of the actual versus predicted classifications, helping to evaluate the accuracy, precision, recall, and F1-score of the model. The matrix consists of four key components:

1. **True Positives (TP):** The number of correctly predicted positive instances.
2. **False Positives (FP):** The number of instances where the model incorrectly predicted the positive class.
3. **True Negatives (TN):** The number of correctly predicted negative instances.
4. **False Negatives (FN):** The number of instances where the model incorrectly predicted the negative class.



Conclusions

The analysis concluded that a systematic approach to model development—starting with thorough data preprocessing, feature engineering, and rigorous model training—can significantly improve classification performance. Hyperparameter tuning and optimization were critical in refining the models, enabling the selection of the best-performing classifier. By focusing on relevant evaluation metrics like accuracy and F1 score, the process ensured that the final model was both accurate and generalizable. The results demonstrated the effectiveness of the methodology in predicting outcomes, highlighting its potential applicability to similar classification problems in the future.

Appendix

GITHUB LINK for the complete code:

https://github.com/priya-ganna/IBM-Data-Science-Series-10-Capstone.git

Thank you!