IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chaw Su Aung

30 May 2025

Outline

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

Executive Summary

This is predictive analytics project on the SpaceX Falcon 9 launch dataset predicting the mission outcome (success or failure).

Real-world data was collected using API, Web-scraping and performed basic exploratory data analysis. Data visualization was created with Folium maps and Plotly Dash apps.

Since the target feature is categorical data, some of classification models such as Random Forest Classifier, SVC, Logistic Regression, K-Nearest Neighbour and Decision Tree Classifier are trained. They were then fine-tuned using GridSearchCV.

Project show case the application of data science data wrangling, feature engineering, interactive visualization, and machine learning for binary classification tasks.

Introduction

SpaceX has pioneered the commercial space launch sector and aimed to make space travel more reliable and reducing the cost as much as 63% lower than the average. One of the factors that contribute highly cost-effective launches is reusing first stage from the successful missions.

We aim to predict whether the first stage of SpaceX Falcon 9 rocket will land successfully which will be able to determine the cost of the mission. In this project, we will be collecting, preparing SpaceX missions data, wrangling to uncover hidden insights, analyze launch characteristics and success rates.

Furthermore, we will visualize mission geolocations, correlations of features and build supervised machine learning models for predicting missing outcome.

Section 1

Methodology

Methodology

1. Data collection methodology: SpaceX's public API, Web scraping with Beautiful soup from Wikipedia for historical data
2. Data wrangling methodology: Perform feature extraction, Null value handling in Pandas DataFrame. Binary classification and label encoding
3. Perform data analysis (EDA) using visualization with Matplotlib and Seaborn.
4. Perform interactive visual analytics using Folium and Plotly Dash.
5. Perform predictive analysis using classification models: Logistic Regression, Random Forest
6. Classifier, Support Vector Machines (SVM), K-Nearest Neighbors (KNN).
7. Perform Hyperparameter tunning with GridSearchCV and evaluation with confusion matrix.

Data Collection

To build a comprehensive dataset, two main data sources were used.

1. SpaceX Public API

Using SpaceX Public API with Python requests library enables the retrieval of real-time updated data and allows a structured and machine-readable JSON file to be directly loaded into Pandas DataFrame.

2. Web scraping data from Wikipedia

Historical data from Wikipedia HTML content was scraped with the Python BeautifulSoup library and parsed the relevant data into tabular format in a Pandas DataFrame.

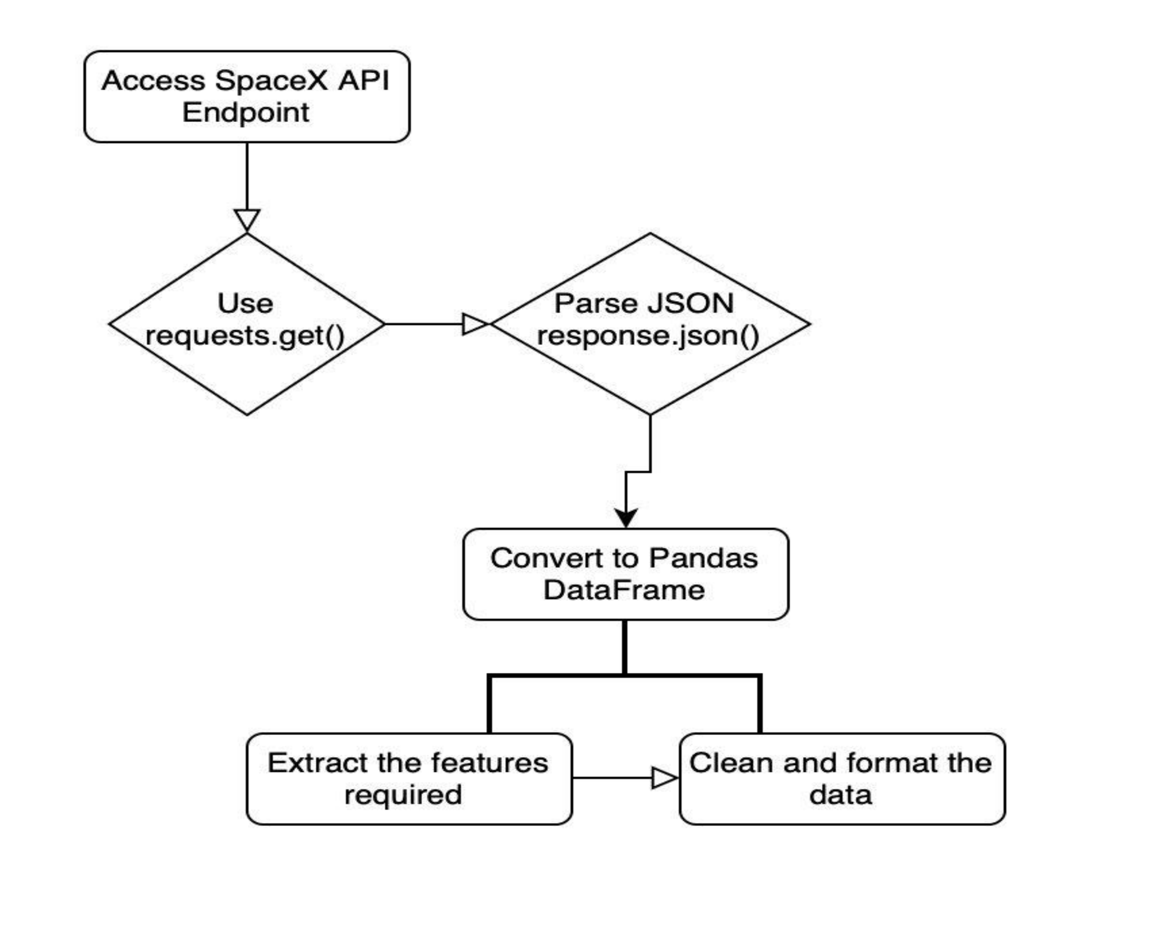
By combining structured data from the API and historical data from web scraping, we ensured the final dataset is suitable for supervised learning.

Data Collection – SpaceX API

Request and parse the SpaceX launch data using the GET request

- API Endpoint Access: Used SpaceX REST API to retrieve the launch data
- Python requests library: performed HTTP GET requests to access API responses in JSON format
- JSON Parsing: Parsed structured JSON data into a Pandas DataFrame
- Extract the features required: 'rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc'
- Data Cleaning: Removed null or irrelevant entries and standardized formats.

Gihub URL: SpaceX data collection with API

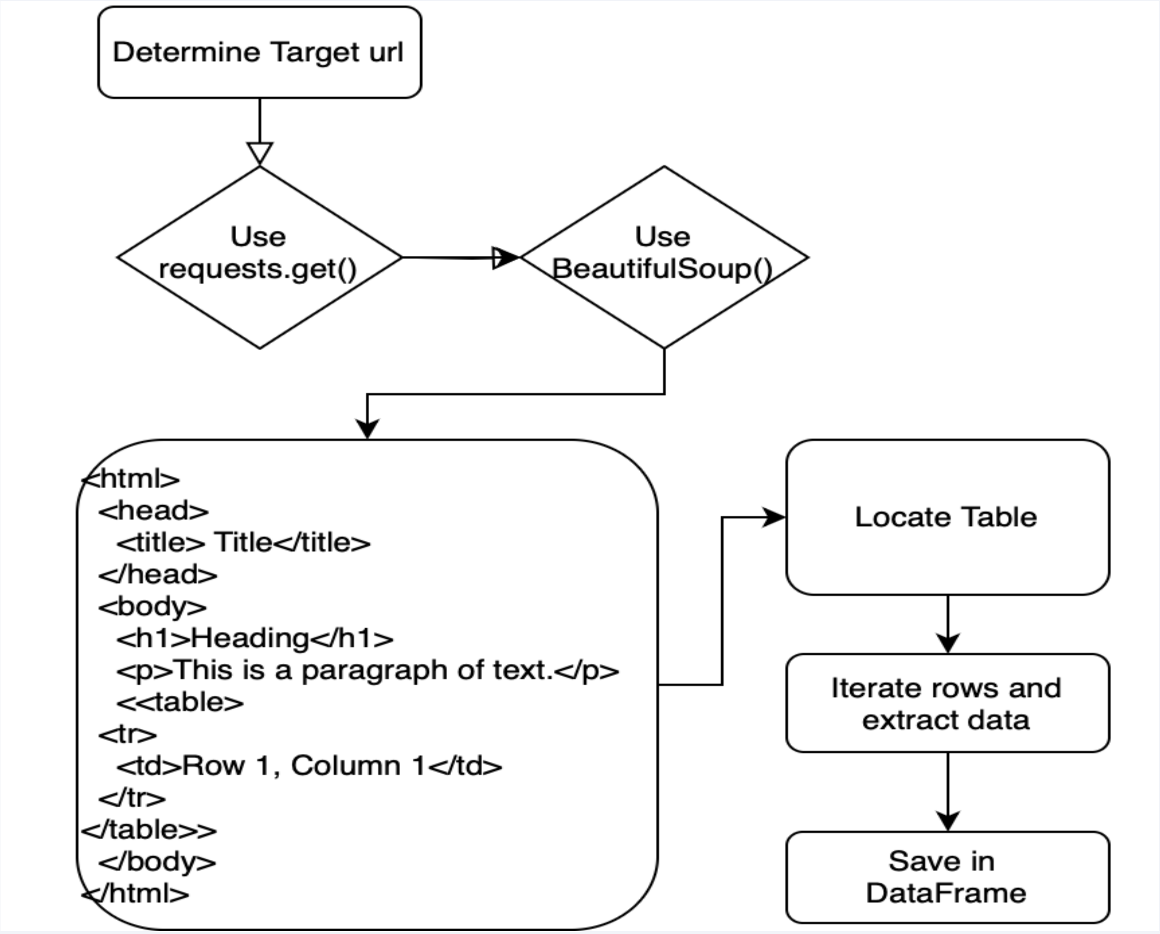

Access SpaceX API Endpoint
Use requests.get()
Parse JSON response.json()
Convert to Pandas DataFrame
Extract the features required
Clean and format the data

Data Collection - Scraping

Scraping HTML content from Wikipedia

- Determine target URL: Wikipedia page on Falcon 9 launches
- Python BeautifulSoup library: scraped HTML tables
- Table Extractions: located the specified tables
- Row Iteration: extracted required data from the iterated rows.
- Data Cleaning: Removed irrelevant columns.
- Creating DataFrame: Save the results to Pandas DataFrame for further wrangling.

Github URL: SpaceX data with Webscraping



Determine Target url

Use requests.get()

Use BeautifulSoup()

```
<html>
  <head>
    <title> Title</title>
  </head>
  <body>
    <h1>Heading</h1>
    <p>This is a paragraph of text.</p>
    <<table>
  <tr>
    <td>Row 1, Column 1</td>
  </tr>
</table>>
  </body>
</html>
```

Locate Table

Iterate rows and extract data

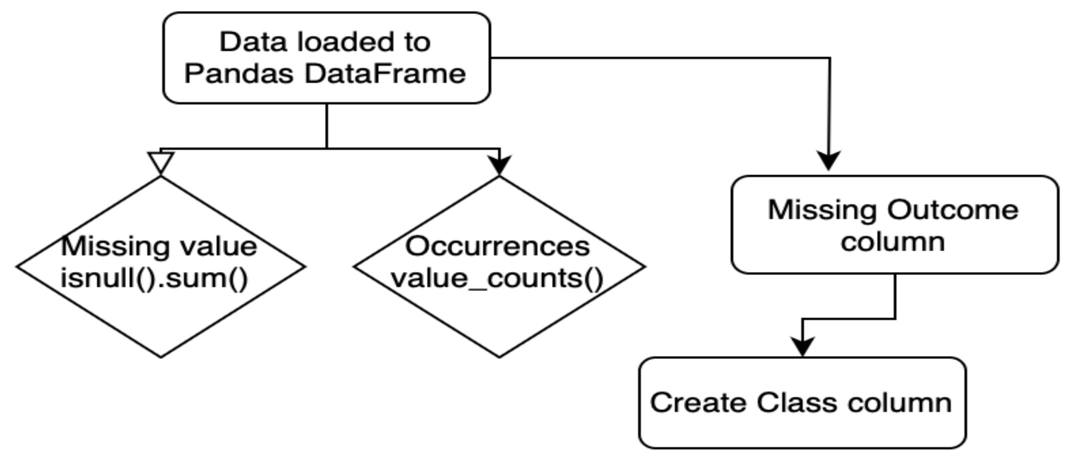
Save in DataFrame

Data Wrangling

Data Wrangling in Pandas DataFrame

- Identified missing values in each attributes
- Performed basic analysis such as number of launches in each site, mission outcome of the launch sites.
- Created new column 'Class': landing class (1: Success and 0: Failure) from mission outcome column.
- Calculated the mean of landing class (Success rate)
- Exported final cleaned dataset and saved as CSV file.

[Github URL: SpaceX data wrangling]


Data loaded to Pandas DataFrame
Missing value isnull().sum()
Occurrences value_counts()
Missing Outcome column
Create Class column

EDA with Data Visualization

Exploratory Data Analysis was performed to obtain preliminary insights about important variables that would affect the success rate and selected features to use in predictions. Several visualizations were used to perform the task.

- Seaborn Categorical plots for
  - FlightNumber vs. PayloadMass
  - FlightNumber vs LaunchSite
  - FlightNumber vs Orbit type
  - Payload Mass vs Launch Site
  - Payload Mass vs Orbit type

- Seaborn Bar plot for the relationship between success rate of each orbit type
- Seaborn Line plot for launch success yearly trend

Github URL: EDA with Data Visualization

EDA with SQL

To better understand the dataset, SQL queries were made using sqlite3.

- All Launch Site Names
- Launch Site Names Begin with 'CCA'
- Total Payload Mass
- Average Payload Mass by F9 v1.1
- First Successful Ground Landing Date
- Successful Drone Ship Landing with Payload between 4000 and 6000
- Total Number of Successful and Failure Mission Outcomes
- Boosters Carried Maximum Payload
- 2015 Launch Records
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Github URL: EDA with SQL

Build an Interactive Map with Folium

With Folium map, we were able to find geographical patterns about launch sites.

- folium.Map() to initialize the map with coordinates to mark all launch sites on a map
- folium.Circle() to add a highlighted circle area in the map.
- folium.Marker() to add an icon showing the name of the area and grouped marker with MarkerCluster() object to show the success/failed launches for each site on the map
- folium.PolyLine() to describe the distances between a launch site to its proximities to coastline, railways, highways and cities.

Github URL: Visualization with Folium

Build a Dashboard with Plotly Dash

Summary of Plots/Graphs and Interactions

1. Launch Site Dropdown (dcc.Dropdown)

Purpose: Allows the user to select either all launch sites or specific launch site and enables to examine differences in success rates and payload outcomes.

2. Success Pie Chart (dcc.Graph)

Purpose: Helps visually compare the performance of different sites' success rate.

3. Payload Range Slider (dcc.RangeSlider)

Purpose: Payload is an important variable in rocket launches and this helps study its relationship to mission outcome.

4. Payload vs. Success Scatter Plot (dcc.Graph)

Purpose: Shows correlation between payload weight and success. Helps identify if heavier/lighter payloads affect mission outcome. Allows comparison across booster versions.

Github URL: SpaceX dashboard with Dash

Predictive Analysis (Classification)

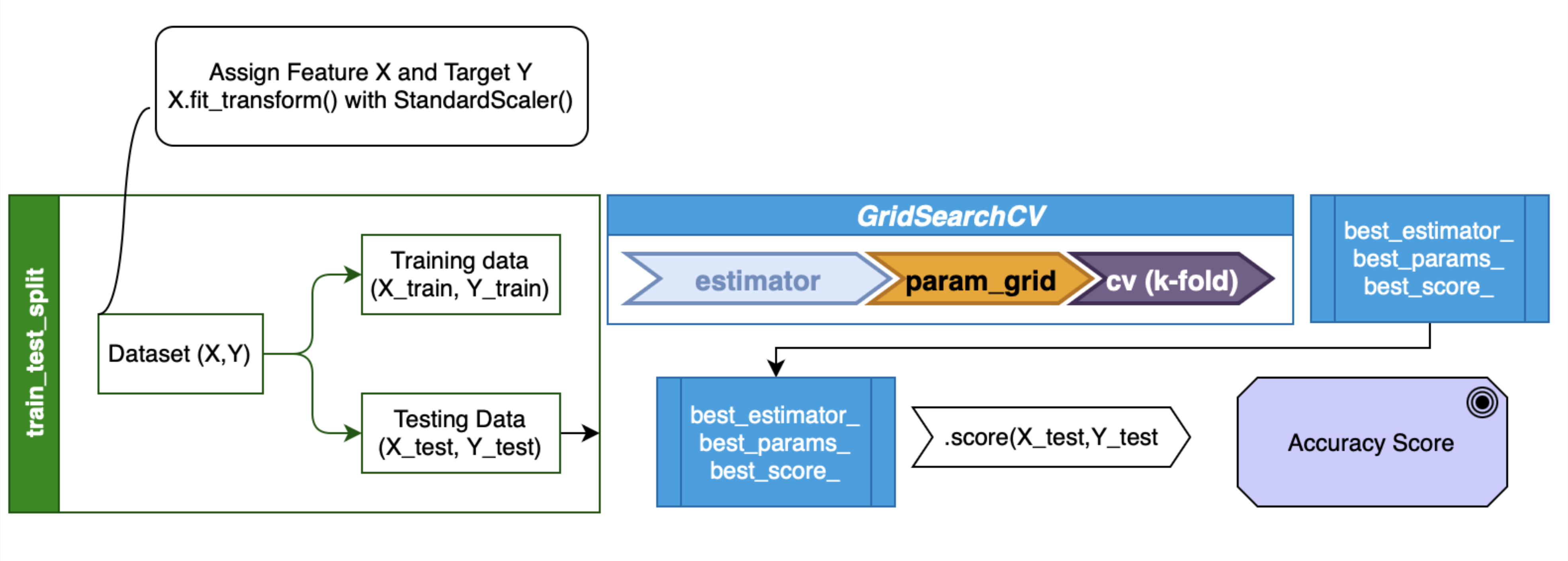
Summary of Model Development Process

1. Assign the data into features (X) and target variable (y) (class = mission success). Features (x) were preprocessed using StandardScaler().
2. Split the dataset into training set (80%) and test set (20%) using train_test_split().
3. Trained and evaluated multiple supervised classification models: Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest Classifier.
4. Applied GridSearchCV for hyperparameter optimization.
5. Used the metrics such as Accuracy, Precision, Recall, F1-score (via classification_report) to compare model performance.
6. Compared cross-validated scores and test set performance.
7. Selected the model with highest accuracy and best-balanced metrics as the final model.

Github URL: SpaceX_Machine Learning Prediction

Predictive Analysis (Classification)

Model Building Flowchart



Results

Exploratory data analysis results

- We found that flight number shows correlation with Payload Mass, Launch Site, and Orbit Type.
- With EDA, we found characteristics of dataset to guide feature selection and model development such as all launch sites, first successful landing date, booster types, including which booster carried maximum payload.

Interactive analytics results

- Line plot indicates a clear upward trend in mission success rate over the years.
- Bar plot for relationships between success rate and orbit type shows ES-L1, GEO, HEO and SSO orbit types have highest success rate among other.

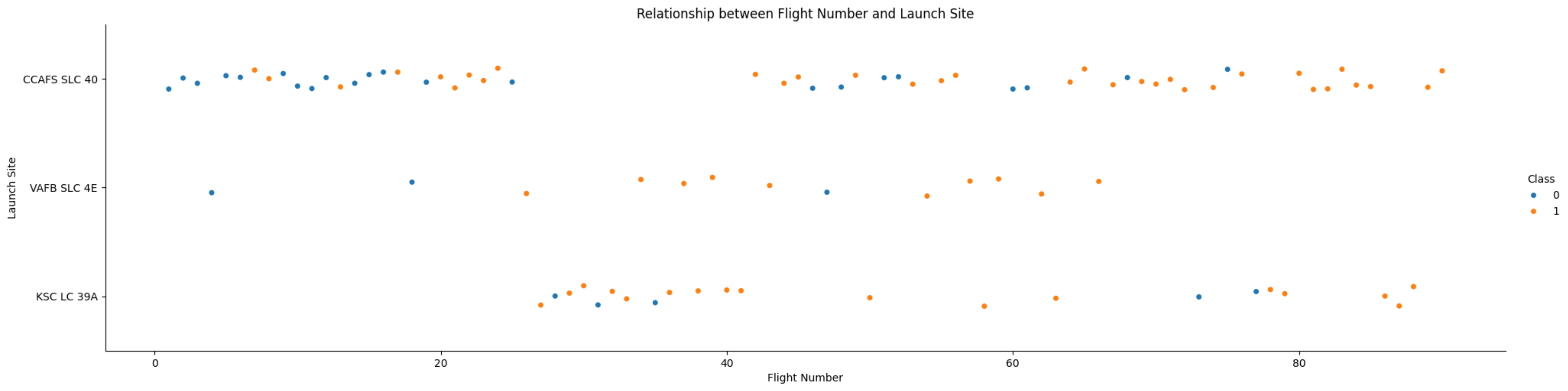
Predictive analysis results

- DecisionTreeClassifier achieved high accuracy score of 0.94 and it was outperforming other models by decreasing False positve in the model.
- 3 other models (LogisticRegression, SVM and KNN) testing accuracy are at 0.83.
- However, target categories are imbalanced and it is essential to use stratified sampling in both train_test_split and cross-validation to ascertain the reliability of models.

Section 2

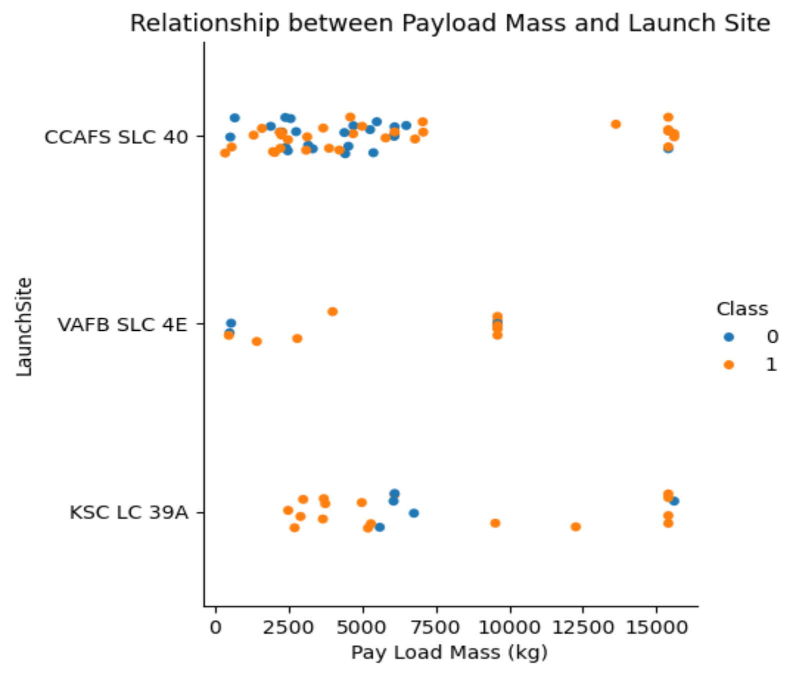
Insights drawn from EDA

Flight Number vs. Launch Site



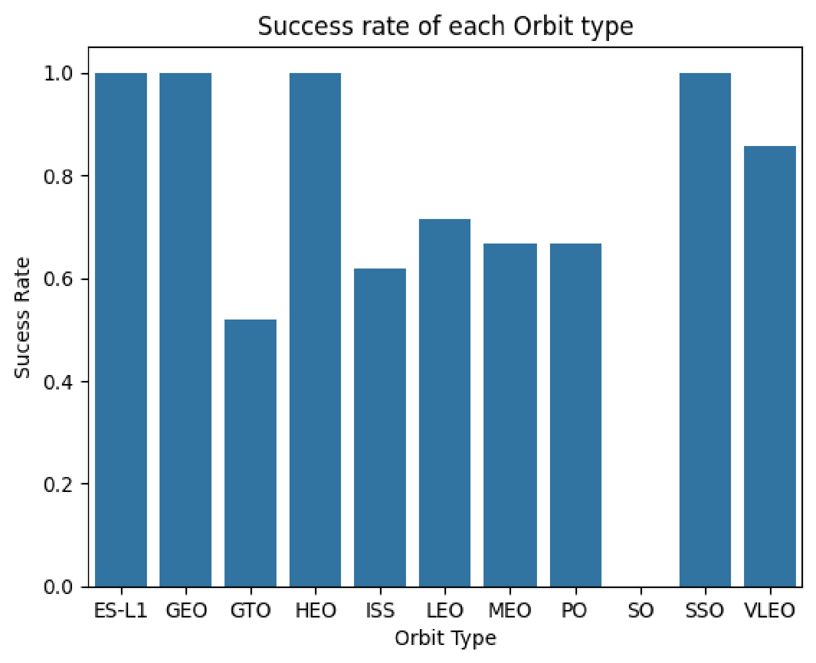
- Overall mission outcome shifted to 'Success' as flight number increased indicating the trend moving to success after initial learning curve.
- CCAFS SLC40 launch site has highest number of launches with a mix of early failure and recent success.
- VAFB SLC 4E has lowest number of launch and KSC LC 39A mostly appears after flight number ~20 and also shows strong performance, with relatively few failures.

Payload vs. Launch Site



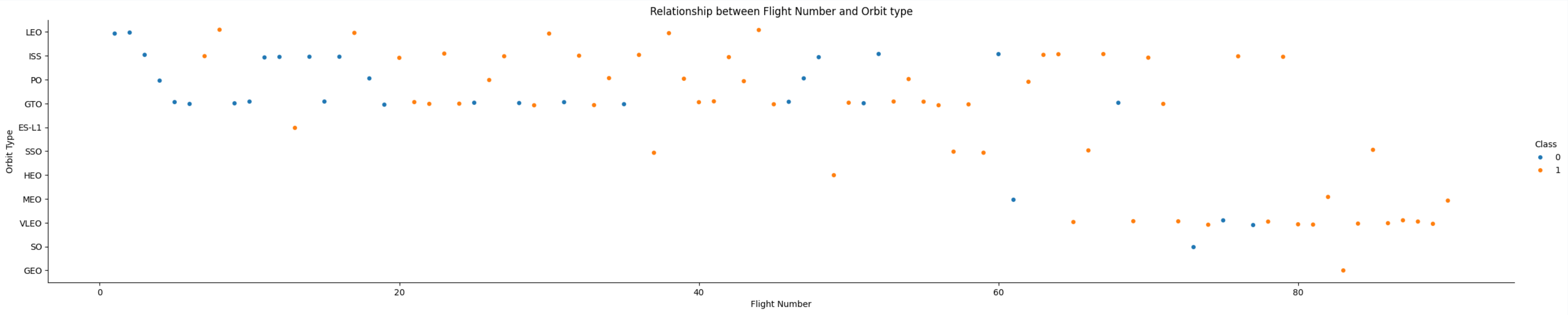
- As established prior, CCAFS SLC40 launch site handled most of earlier launches which resulted in failure and the plot shows many of them are below (~7500 kg) of pay load mass.
- CCAFS SLC40 and KSC LC 39A handled heaviest pay load launches up to (~15,000 kg) with rare failure.
- VAFB SLC 4E has no launch with pay load above (~10,000 kg) which might indicate the possible capacity limitation.

Success Rate vs. Orbit Type



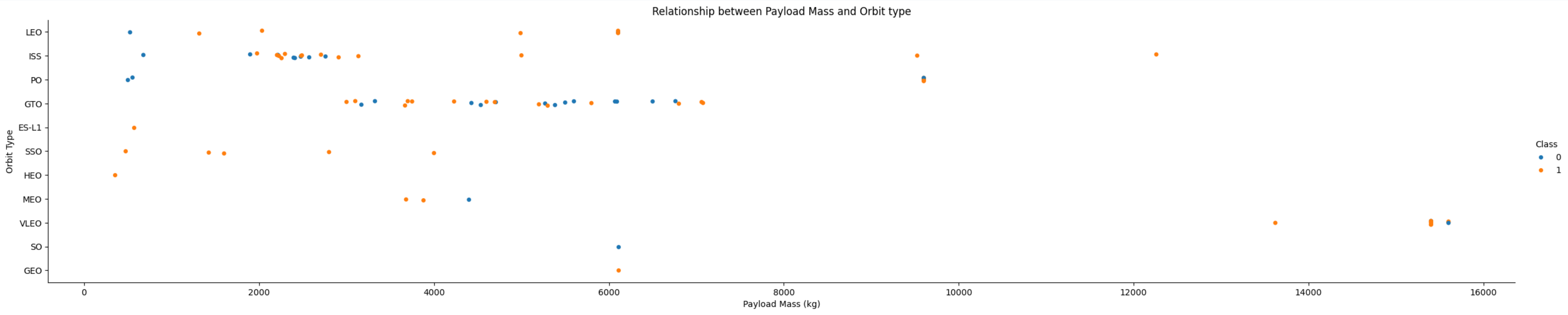
- ES-L1, GEO, HEO, and SSO orbit types show a strong positive correlation with mission success rates.
- GTO (Geostationary Transfer Orbit) exhibits the lowest success rate among all orbit types.
- SO (Sun-Synchronous Orbit) has one failed launch and maintained as lowest success rate.

Flight Number vs. Orbit Type



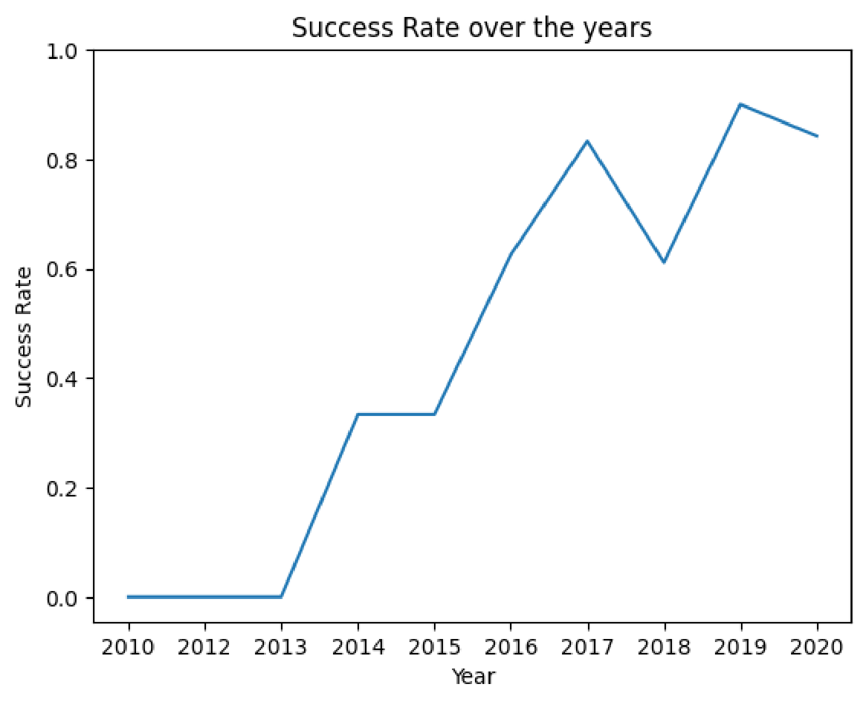
- The first mission was launched to LEO orbit and it ended in failure.
- Up to ~35 flight numbers, launches were limited only to LEO, ISS, PO and GTO.
- A successful mission was carried out to GEO at later stage while there is one single failed launch for SO.

Payload vs. Orbit Type



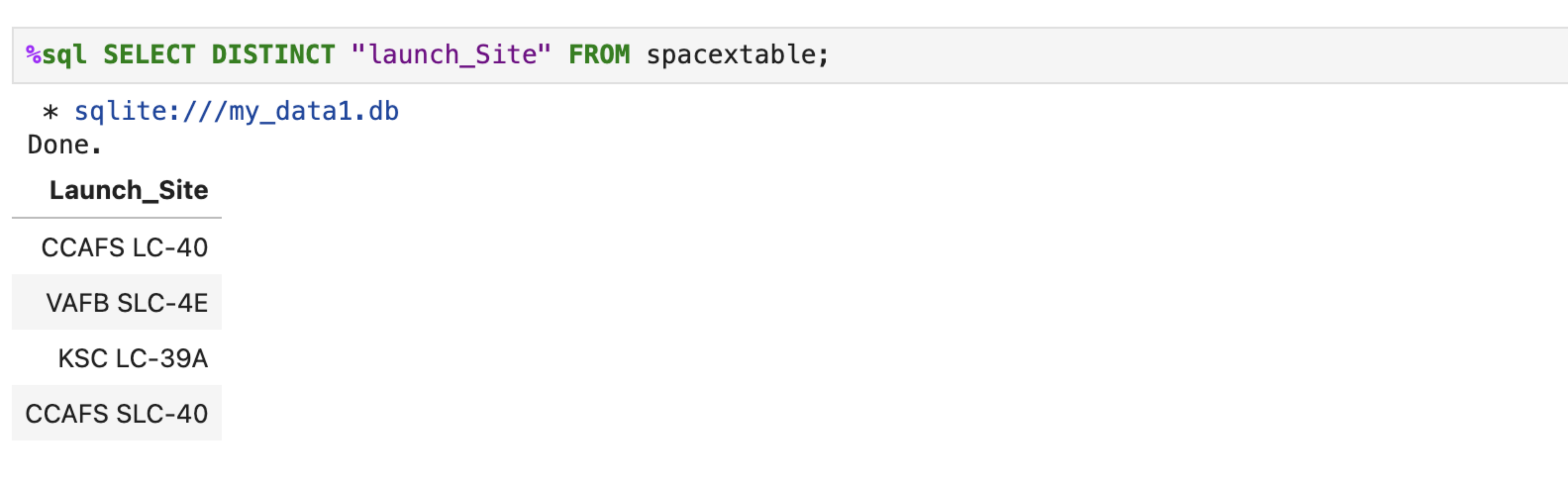
- Highest payload launches with nearly (16,000 kg) are targeted to VLEO.
- The mission with pay load above (~8,000 kg) are relatively rare across all orbit type.
- While there is no strong correlation mission success with payload mass, the number of launches are clearly influenced by the orbit type.

Launch Success Yearly Trend



- Success rate remained flat at 0 from 2010 to 2013 with notable increase at 2013 and again at 2015.
- Highest success rate is approximately at 0.9 in 2019.
- Success rate is trending upward over the year indicating continuous improvement and growth.

All Launch Site Names



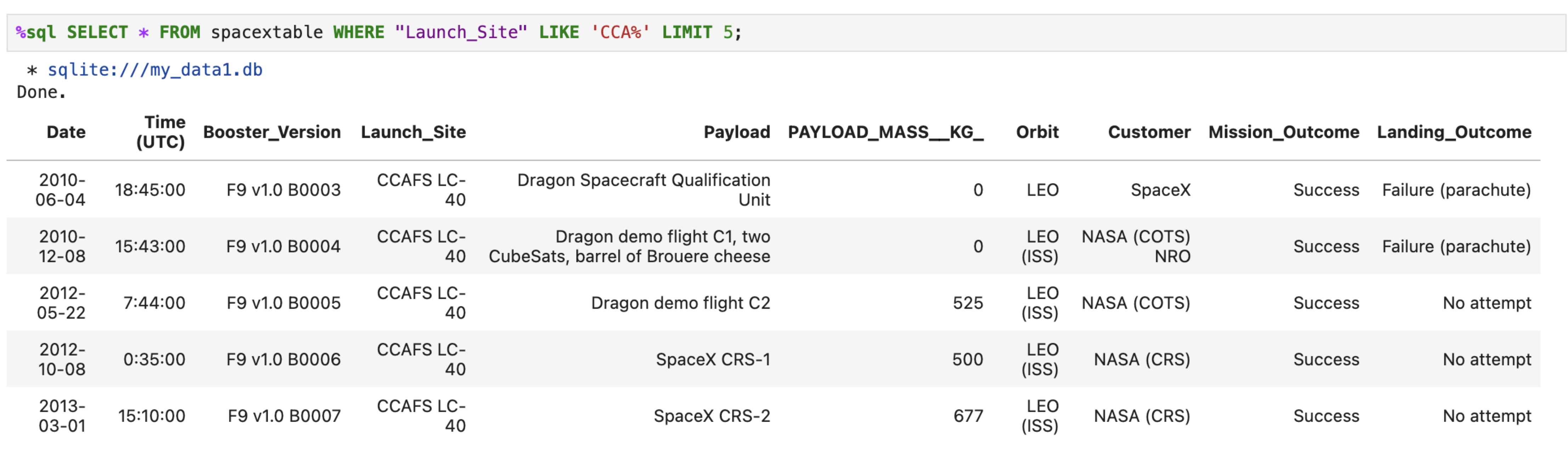
```
%sql SELECT DISTINCT "launch_Site" FROM spacextable;
```

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

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

Query finds all unique Launch Sites with DISTINCT keyword. Retrieves all unique launch site names to understand launch distribution by location.

Launch Site Names Begin with 'CCA'



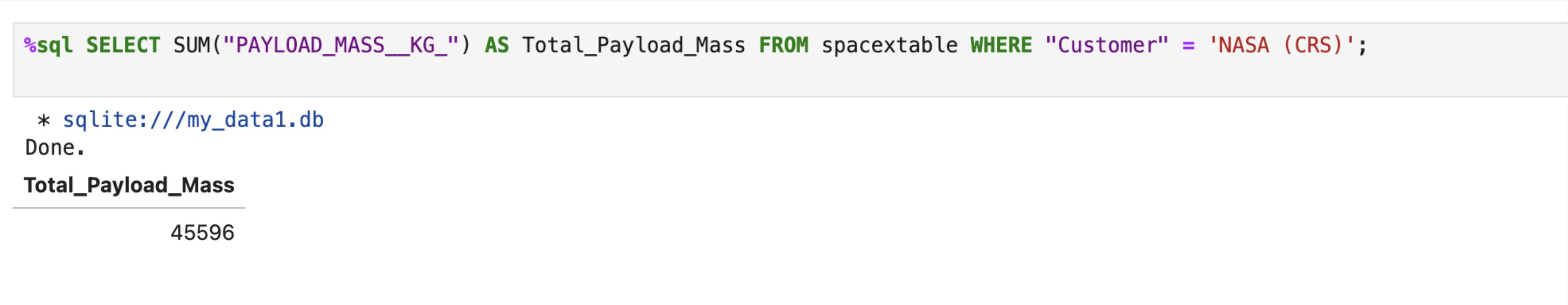
```
%sql SELECT * FROM spacextable WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

```
 * 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

Filters launch sites that **start with** 'CCA', to focus the characteristics of attributes.

Total Payload Mass by boosters from NASA



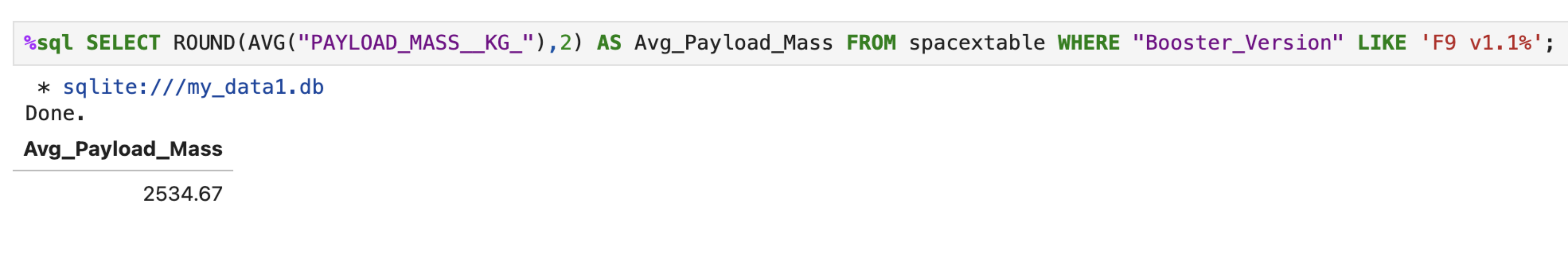
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM spacextable WHERE "Customer" = 'NASA (CRS)';
```

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

Total_Payload_Mass
45596

Query is filtered to provide insight into NASA’s payload capacity.

Average Payload Mass by F9 v1.1



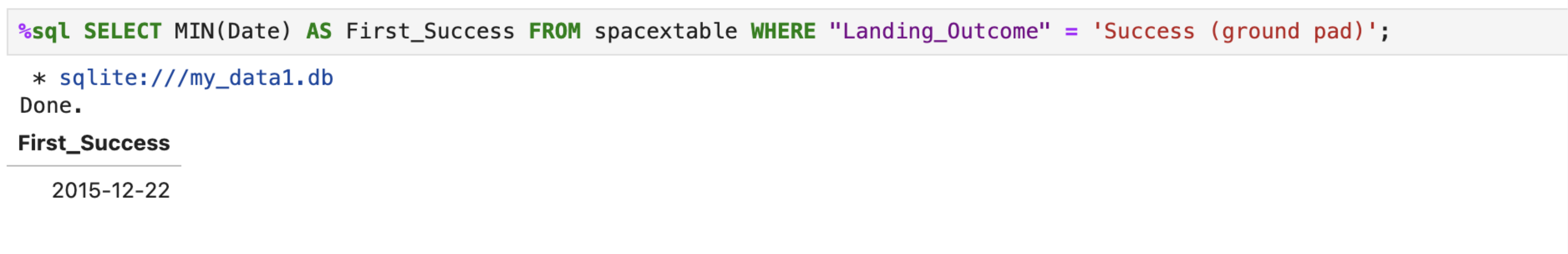
```
%sql SELECT ROUND(AVG("PAYLOAD_MASS__KG_"),2) AS Avg_Payload_Mass FROM spacextable WHERE "Booster_Version" LIKE 'F9 v1.1%';
```

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

Avg_Payload_Mass
2534.67

Computes the **average payload mass** of **Falcon 9 v1.1** booster missions.

First Successful Ground Landing Date



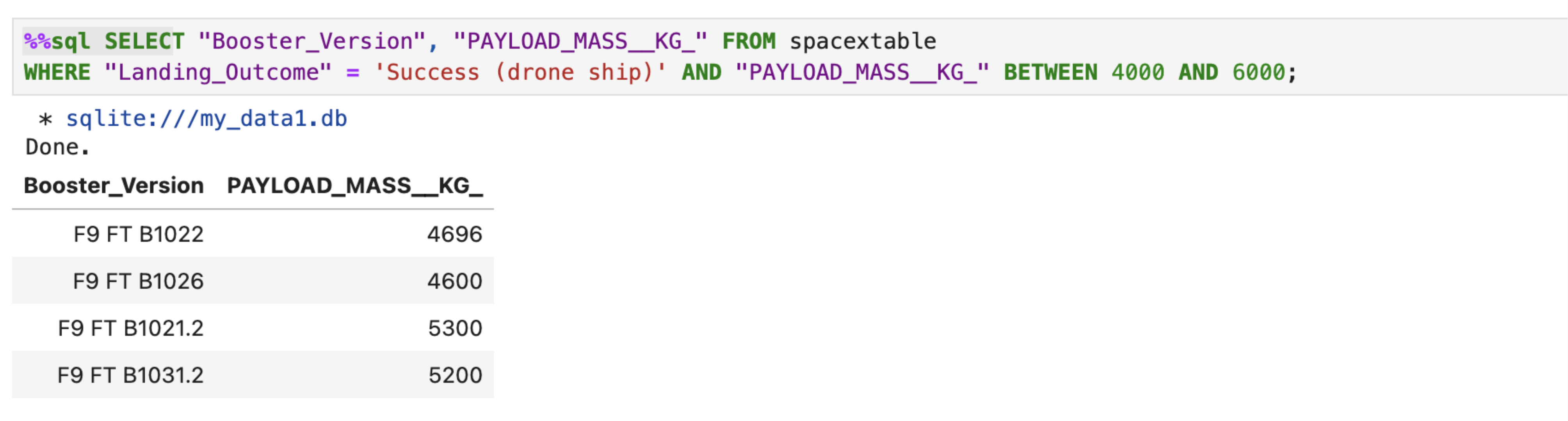
```
%sql SELECT MIN(Date) AS First_Success FROM spacextable WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

First_Success
2015-12-22

Identifies the **earliest date** a **booster successfully landed** on ground.

Successful Drone Ship Landing with Payload between 4000 and 6000



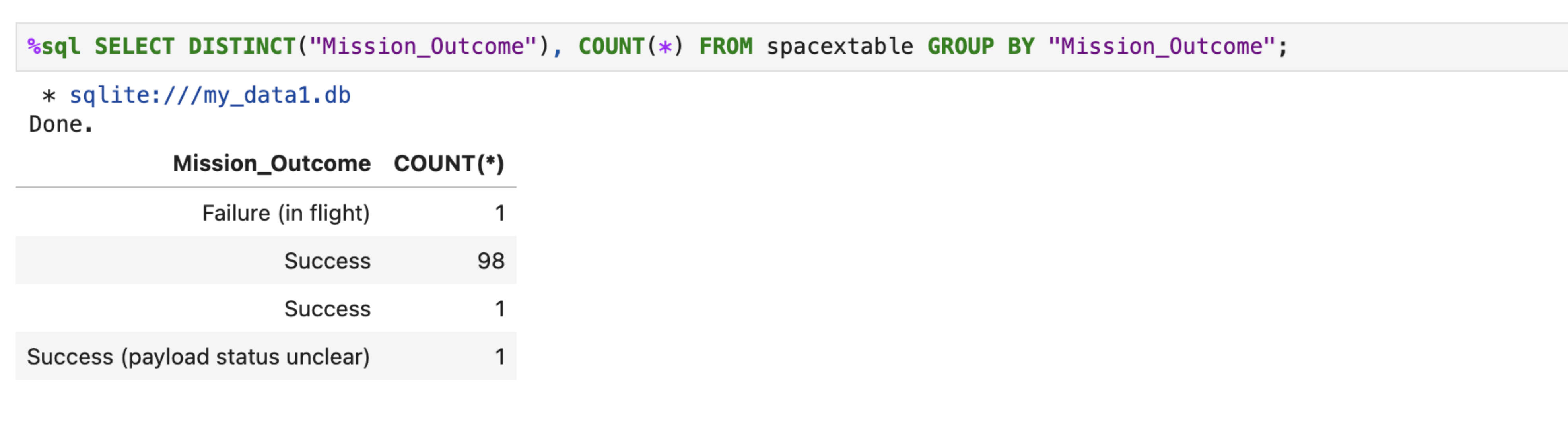
```
%%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM spacextable
WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Filters data to analyze safe recoveries in the range.

Total Number of Successful and Failure Mission Outcomes



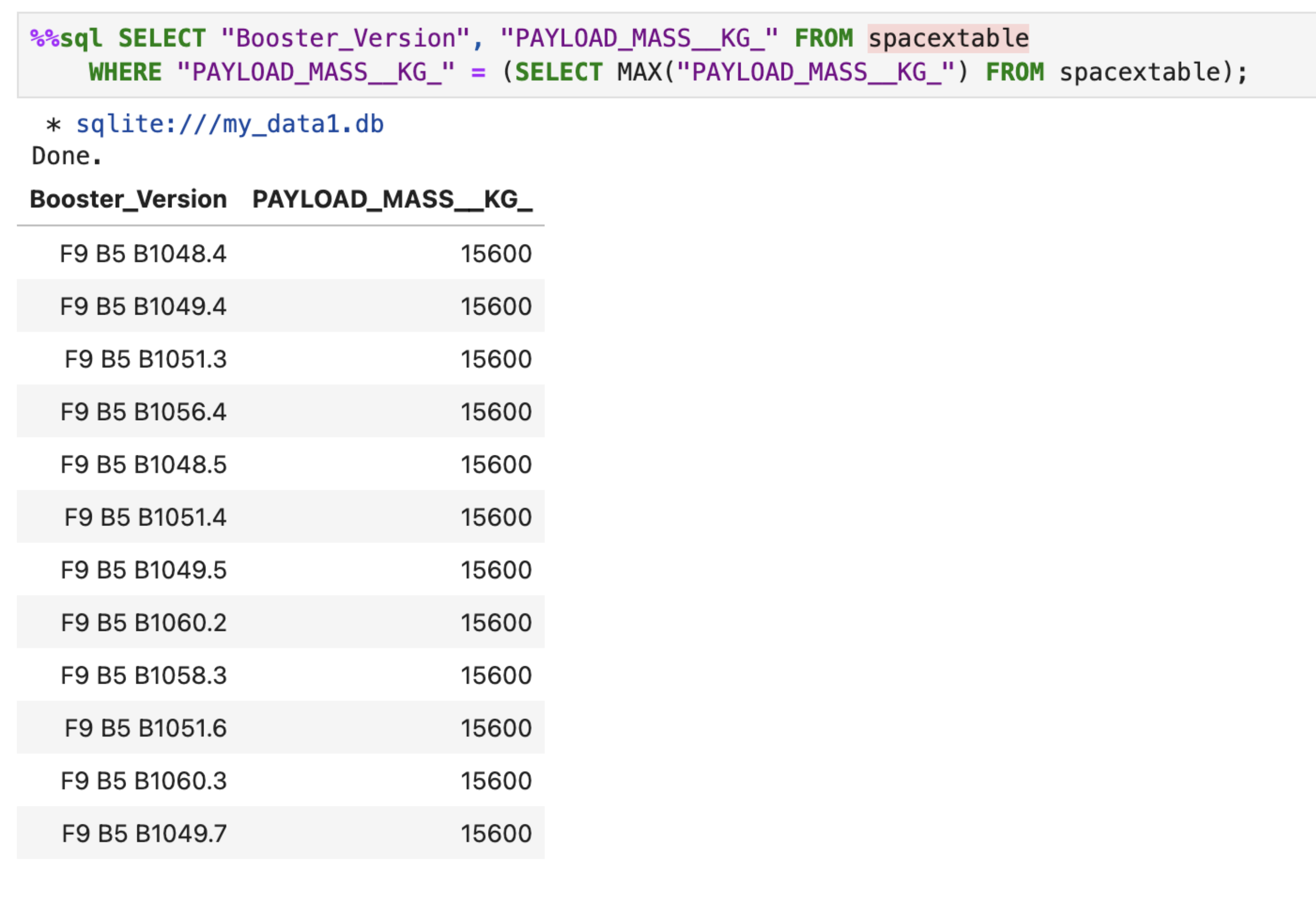
```
%sql SELECT DISTINCT("Mission_Outcome"), COUNT(*) FROM spacextable GROUP BY "Mission_Outcome";
```

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

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

Show case basic success rate breakdown.

Boosters Carried Maximum Payload



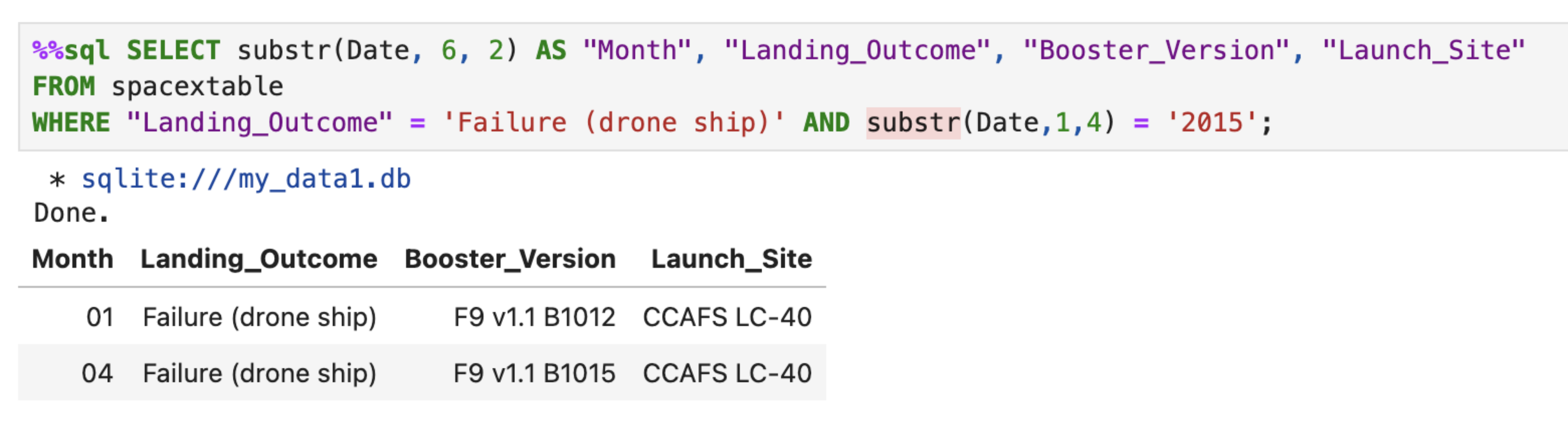
```
%%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM spacextable
    WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM spacextable);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Retrieves booster(s) that carried the heaviest payload and identifies high performing boosters.

2015 Launch Record for Failed Outcomes in Drone Ship



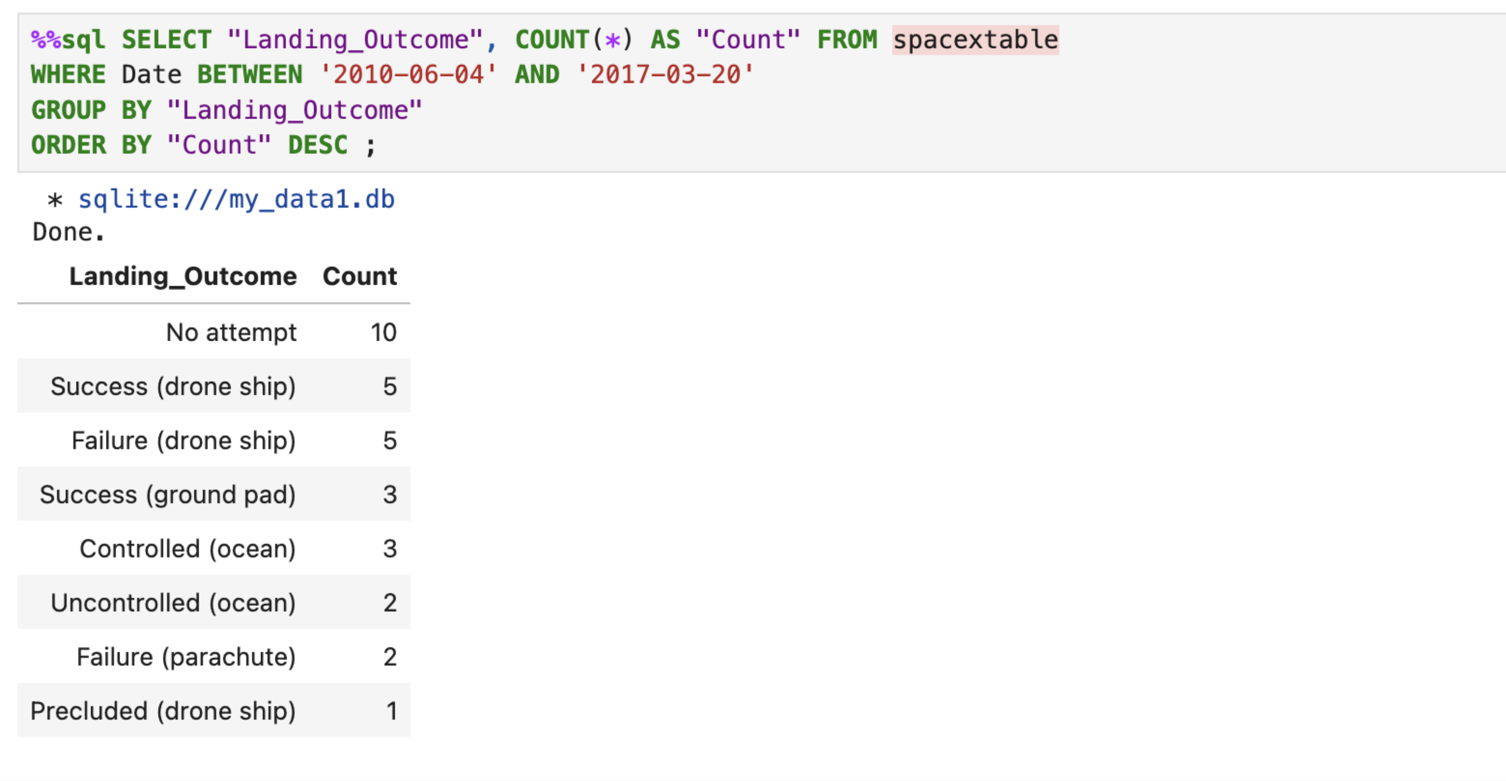
```
%%sql SELECT substr(Date, 6, 2) AS "Month", "Landing_Outcome", "Booster_Version", "Launch_Site"
FROM spacextable
WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr(Date,1,4) = '2015';
```

```
 * 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

Retrieves data that is useful for investigating operational issues in that year.

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



```
%%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 ;
```

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

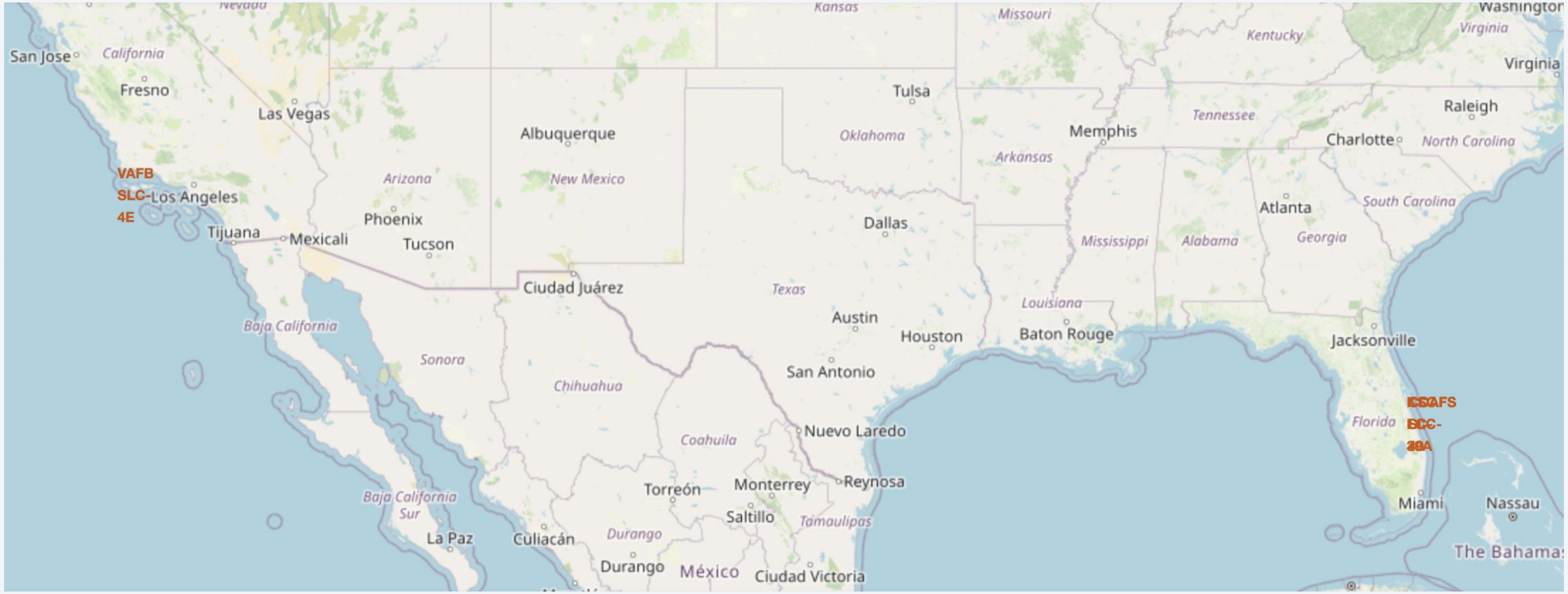
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Ranks different **landing outcomes** showing landing performance evolution.

Section 3

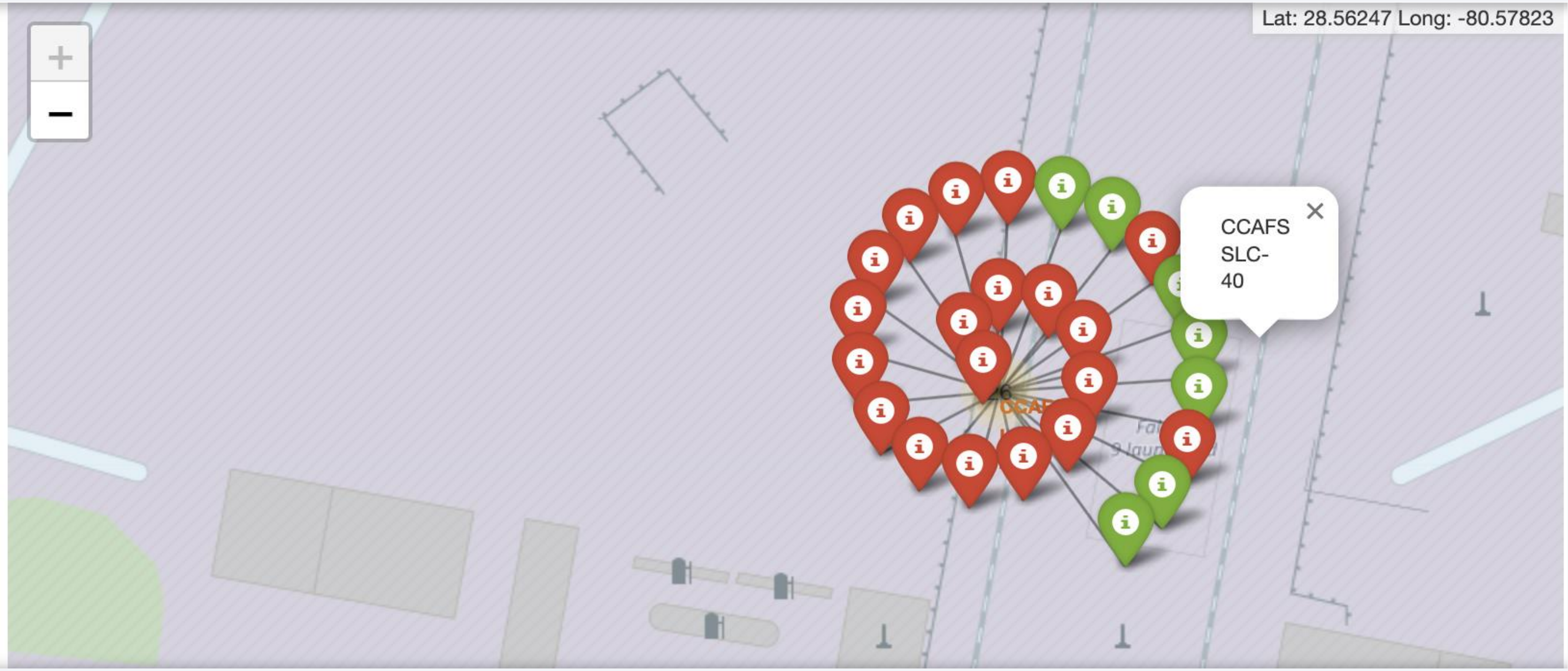
Launch Sites Proximities Analysis

Map showing all Launch Sites



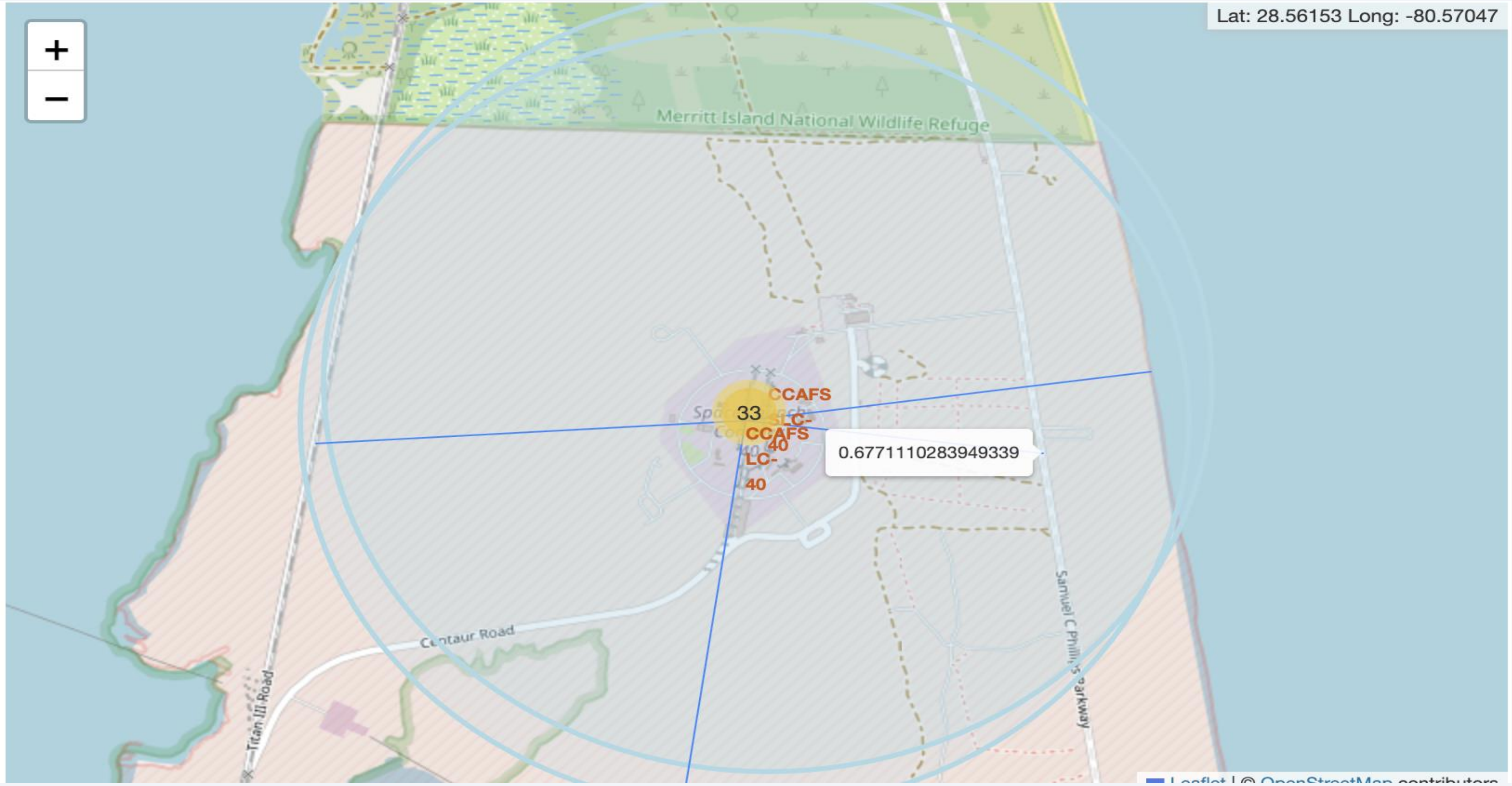
- All launch sites are located near coastlines likely for safety and logistical reasons.
- Site locations near Equator line can help boost velocity resulting efficiency in fuel consumption.

Color Labeled Outcomes at Launch Site

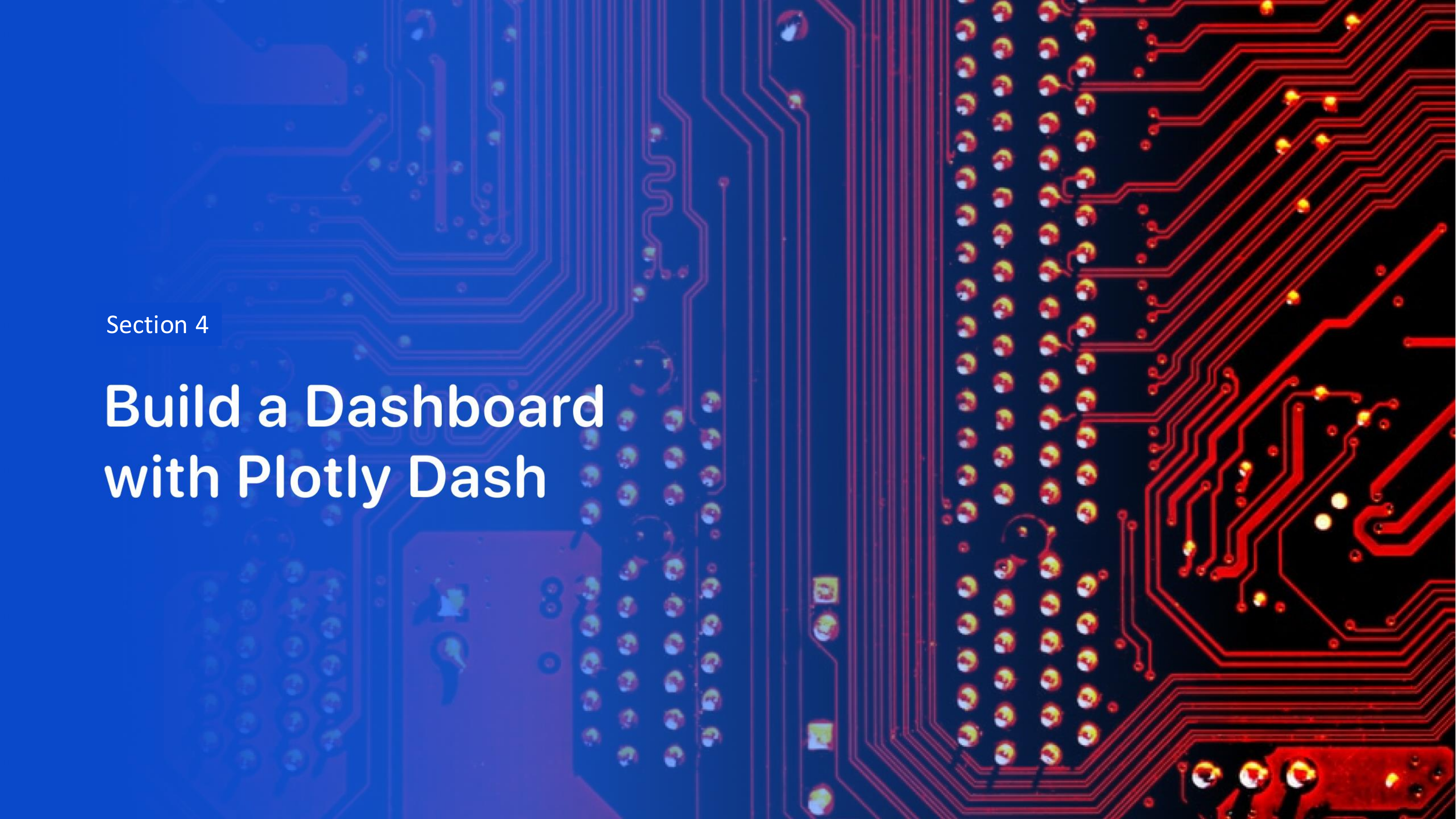


- Mission outcomes are color coded as green (Success) and red (Failure) for launch site CCAF SLC-40

Launch Sites and Proximities


+
−
Lat: 28.56153 Long: -80.57047
Merritt Island National Wildlife Refuge
33
CCAFS
LC-
CCAFS
40
LC-
40
0.6771110283949339
Centaur Road
Titan III Road
Samuel C Phillips Parkway
Leaflet | © OpenStreetMap contributors

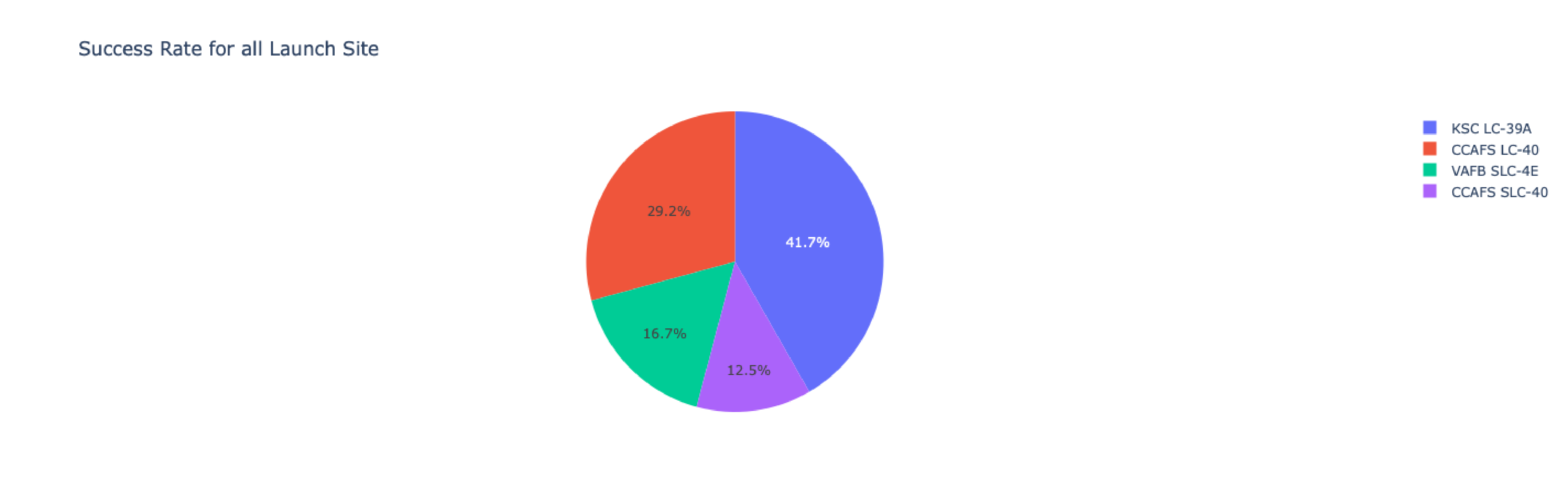
- Launch Sites are within close proximity of railways, highways and coastline, however, far from the city generally to minimize the risks.



Section 4

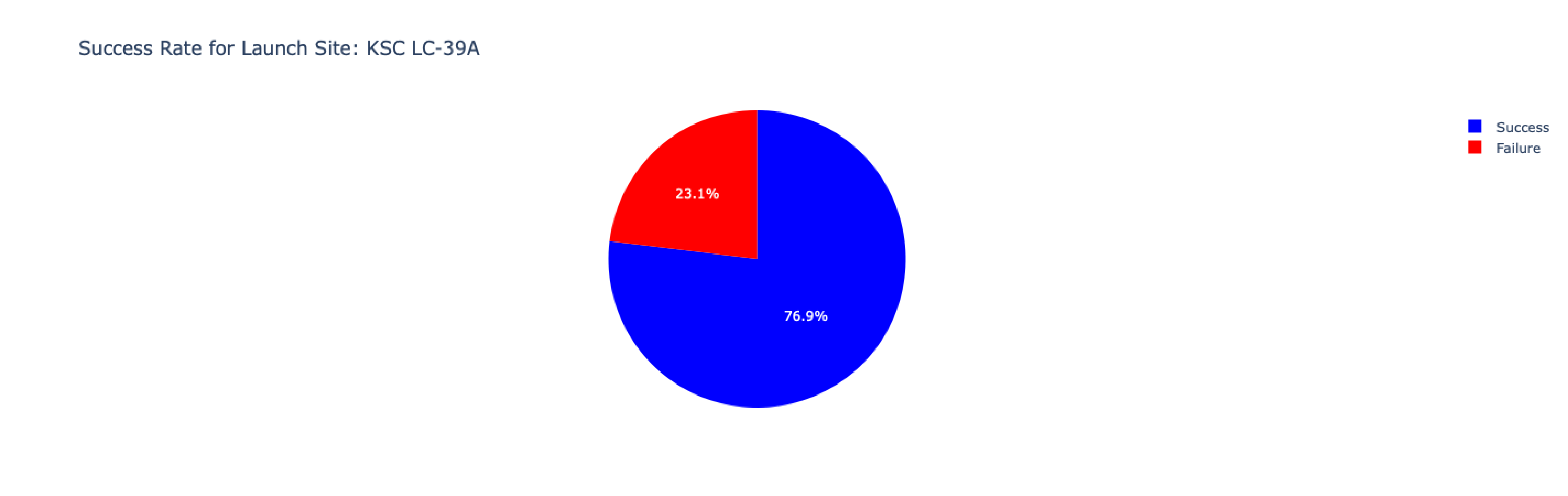
Build a Dashboard with Plotly Dash

Total Success Percentage of all Launch Sites



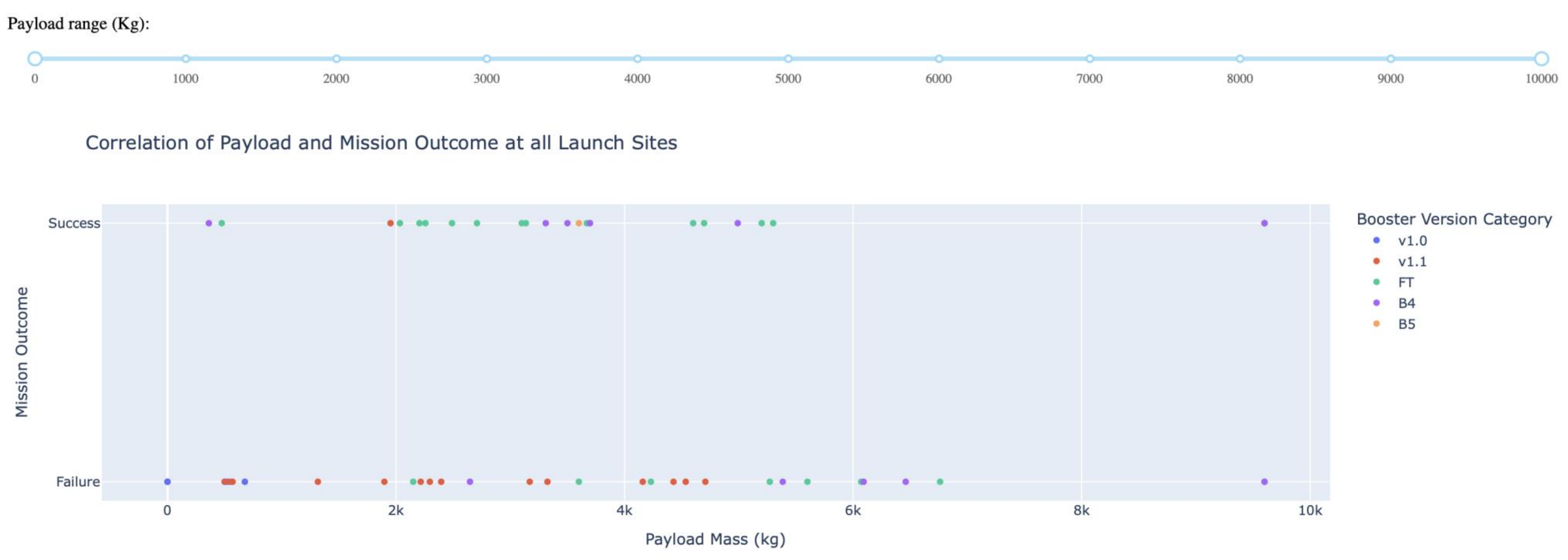
- KSC LC-39A has the highest success rate among other launch sites while CCAFS SLC-40 has the lowest success rate of. Lower rate of CCAFS SLC-40 could be influenced by earlier failure during learning curve.

Success Rate of KSC LC-39A



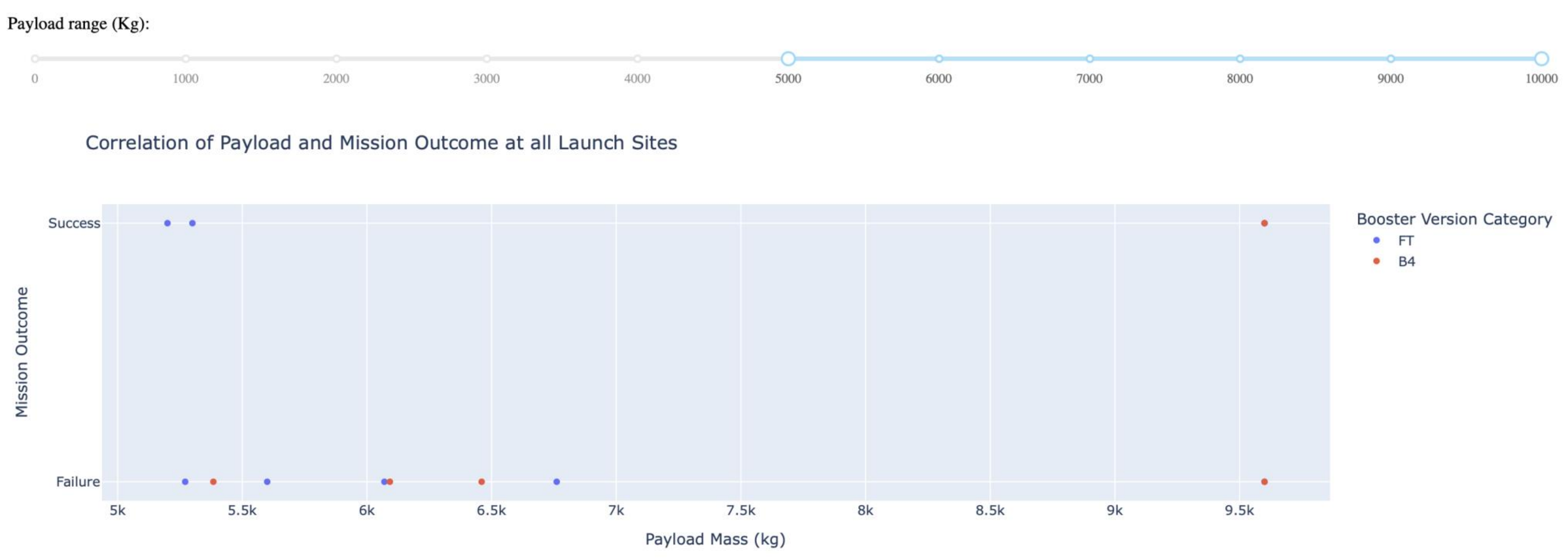
- From dashboard's dropdown menu, each launch site can be filtered and further analyze it's success and failure rate ratio.
- KSC LC-39A leads with 76.9% success rate while CCAFS LC-40 has lowest rate at 26.9%.

Correlation of Payload and Mission Outcome



- In general, no strong linear correlation is observed between payload and mission outcome.
- Booster version v.10 and v1.1 are primarily used in lower payload mass from ~500 kg to ~4500 kg.
- Booster B5 was used in single mission with payload at 3600 kg which resulted in success.

Higher Payload and Booster Versions

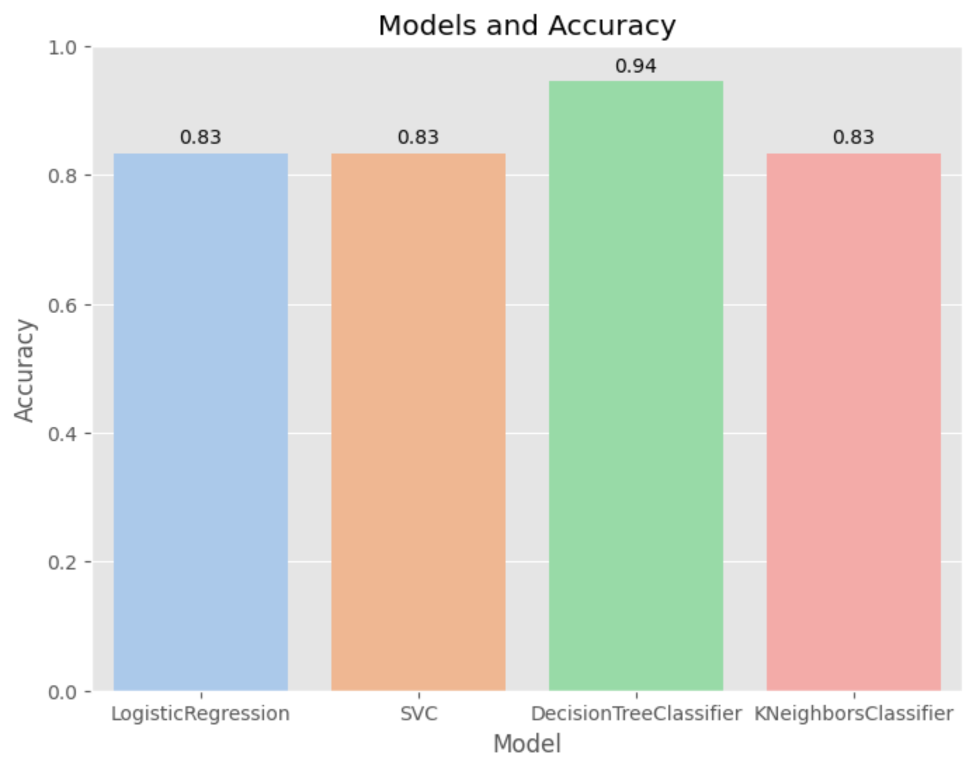


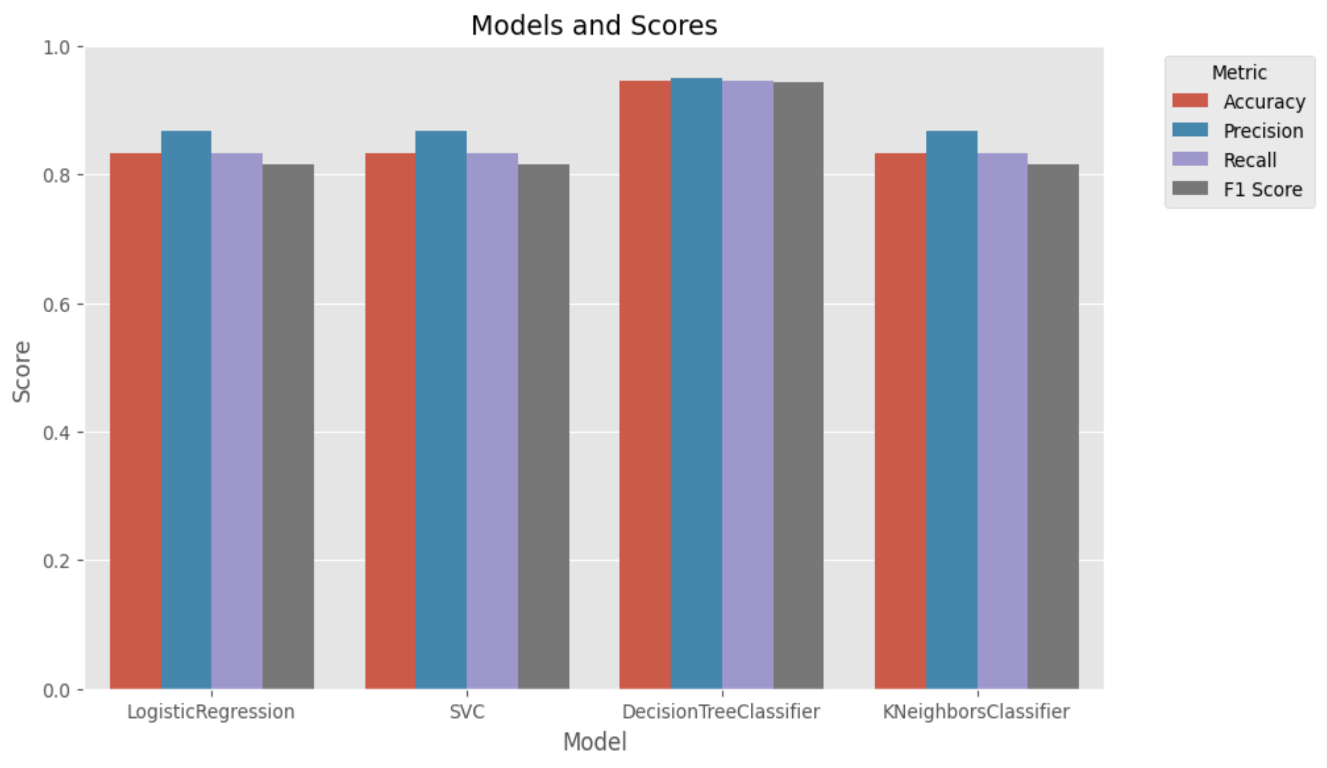
- From payload mass of ~5000 kg to ~10,000 kg, it is observed that only FT and B4 versions were used.
- Missions within payload mass ~ 5000 kg to ~7000 kg shows a higher failure rate compared to success.

Section 5

Predictive Analysis (Classification)

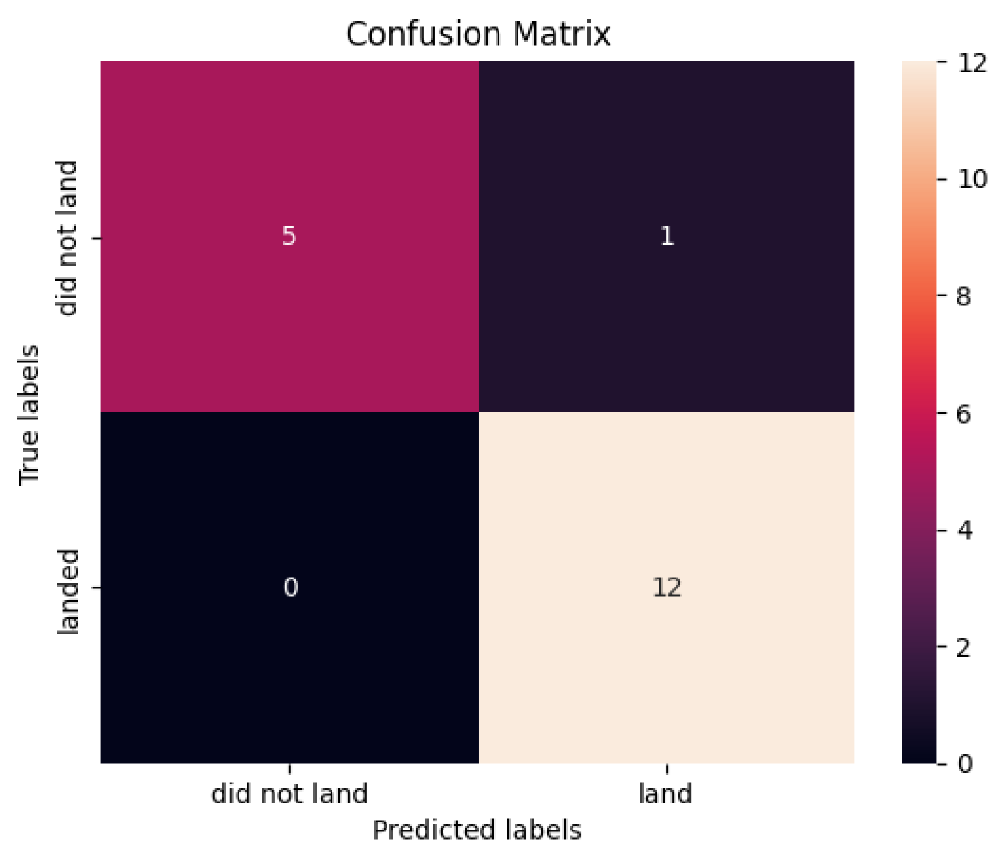
Classification Accuracy



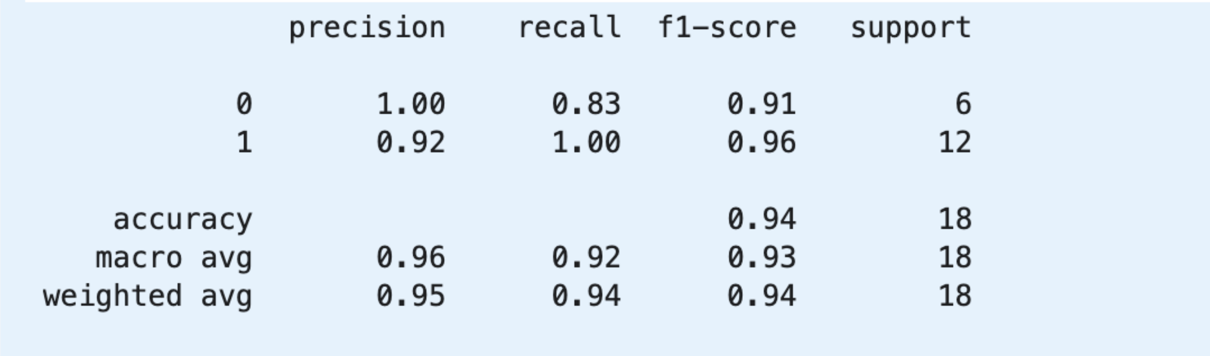


- DecisionTreeClassifier achieved high accuracy score of 0.94 and it was outperforming other models by decreasing False positive in the model.
- 3 other models (LogisticRegression, SVM and KNN) testing accuracy are at 0.83.

Confusion Matrix



Decision Tree Classifier report

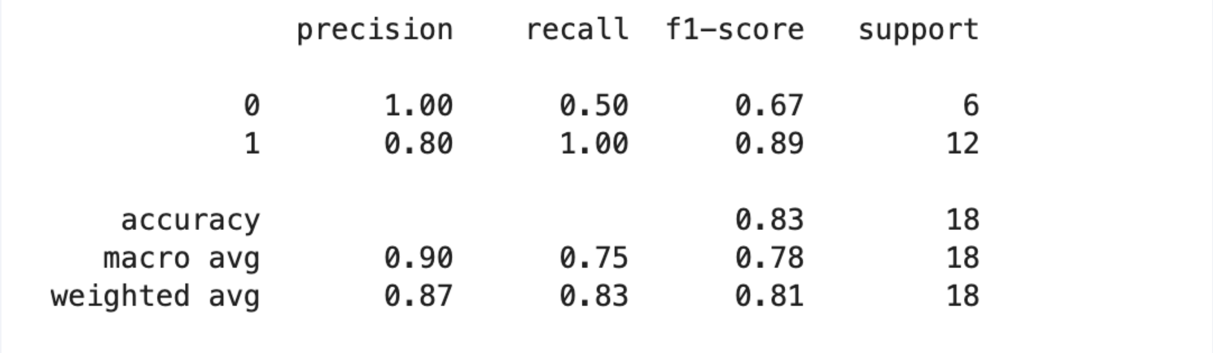


```
              precision    recall  f1-score   support

           0       1.00      0.83      0.91         6
           1       0.92      1.00      0.96        12

    accuracy                           0.94        18
   macro avg       0.96      0.92      0.93        18
weighted avg       0.95      0.94      0.94        18
```

SVM Classifier report



```
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       0.80      1.00      0.89        12

    accuracy                           0.83        18
   macro avg       0.90      0.75      0.78        18
weighted avg       0.87      0.83      0.81        18
```

- Compared to other classifiers , Decision Tree Classifier was able to increase both precision and recall which positively impacted it's overall accuracy score.

Conclusions

1. Decision Tree Performance comparison
   The Decision Tree Classifier achieved the highest accuracy 0.94 significantly outperforming Logistic Regression, SVM, and KNN at 0.83 .This suggests the model was better at capturing patterns in the data.

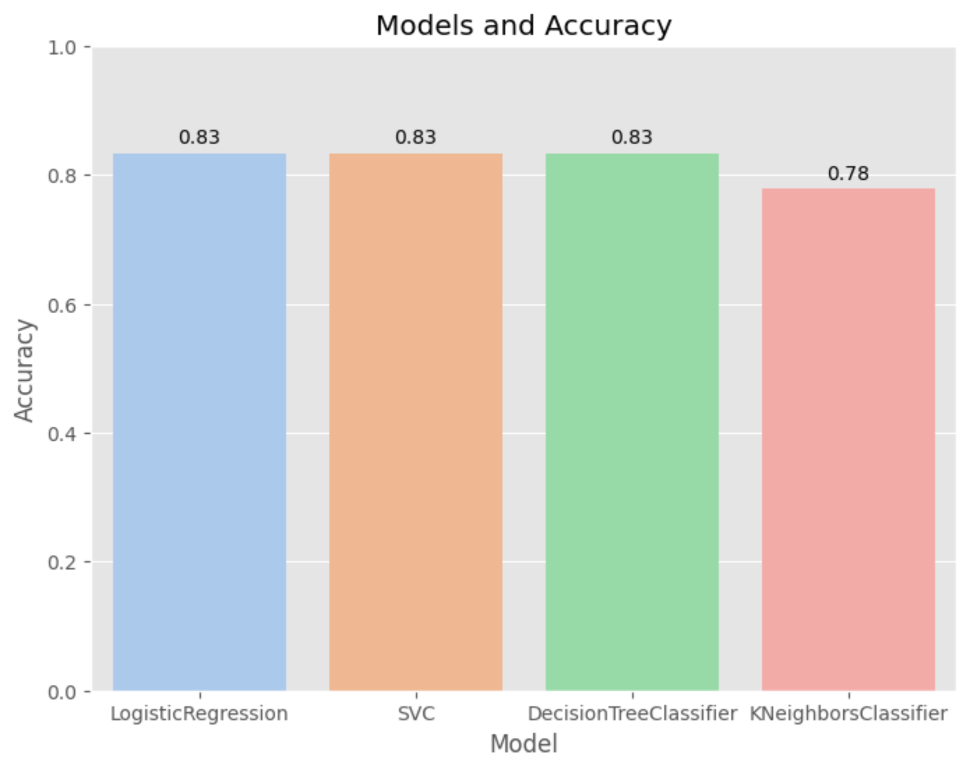
2. Precision and Recall Improvement
   Decision Tree improved both precision and recall.

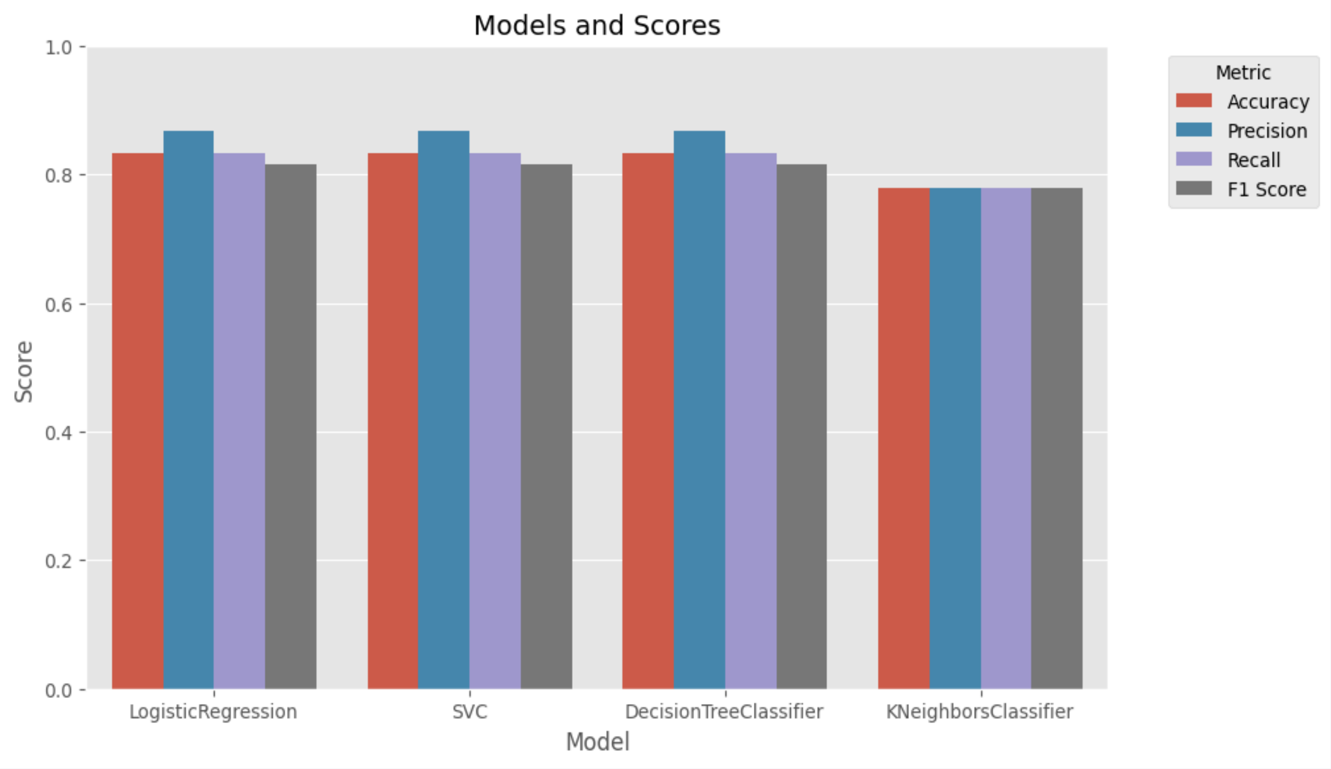
3. Risk of Overfitting
   As target variable is imbalanced and it was not handled, all models are at risk of overfitting.

4. Use of Stratified Validation
   Model's performance result after applying train_test_split with stratify=y and StratifiedKFold are presented in the following slide.

5. Model Robustness through Cross-Validation
   Although accuracy of some models was reduced, using cross_val_score helps validate the consistency of model performance and reliability.

Model Performance after Stratified Validation





- After applying Stratified Validation, the accuracy of Decision Tree Classifier and KNN model has decreased, it suggest that previous results were likely due to overfitting imbalanced target variable and limited dataset size.
- Hyperparameter tunning of models should be revisited for better performance specially for KNN model.

Appendix

- Github URL: Data Science Project Repository

Thank you!