IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Phil Chen
03/16/2024

Outline

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

Executive Summary

- Summary of methodologies

We acquired data of interest through SpaceX API and web-scraping on Wikipedia. After cleaning and organizing the data, we conduct an exploratory data analysis (EDA) to revealed trends and correlations among factors related to launch outcome through data visualization. Subsequently, we built and evaluated several machine learning models (Logistic Regression, SVM, KNN, and Decision Tree) and selected the best model to predict future launch outcomes.

- Summary of all results
  - SpaceX launches its rockets at 4 sites near the coast.
  - KSC LC-39A launch site has highest percentage of success rate compared to other launch sites.
  - Rockets with Booster version FT or B4 are likely to have a successful landing.
  - Few launches for rockets with payloads over 6000kg.
  - Decision Tree Classifier performs the best at predicting landing outcome.

Introduction

- Project background:

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - The percentage of success for all launch sites.
  - The percentage of success / failure for different launch sites.
  - Correlation between payload and success for all launch sites.

Section 1

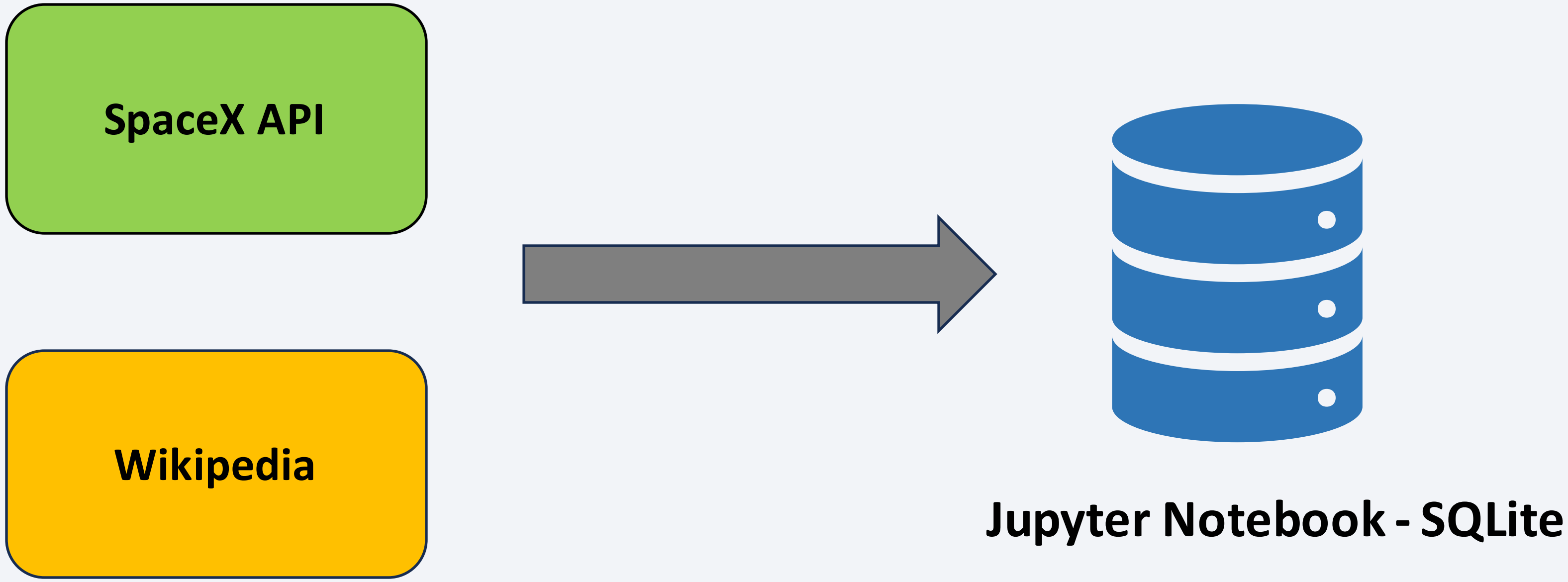
Methodology

Methodology

- Data collection methodology:
  - Download data using SpaceX API and Scrap data from Wikipedia
- Perform data wrangling
  - Clean raw data, handle missing values and standardize data.
  - Create new features in preparation of predictive model training.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Split the dataset into train and test model, tune the model with hyperparameters
  - Evaluate model performance using accuracy and confusion matrix.

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts



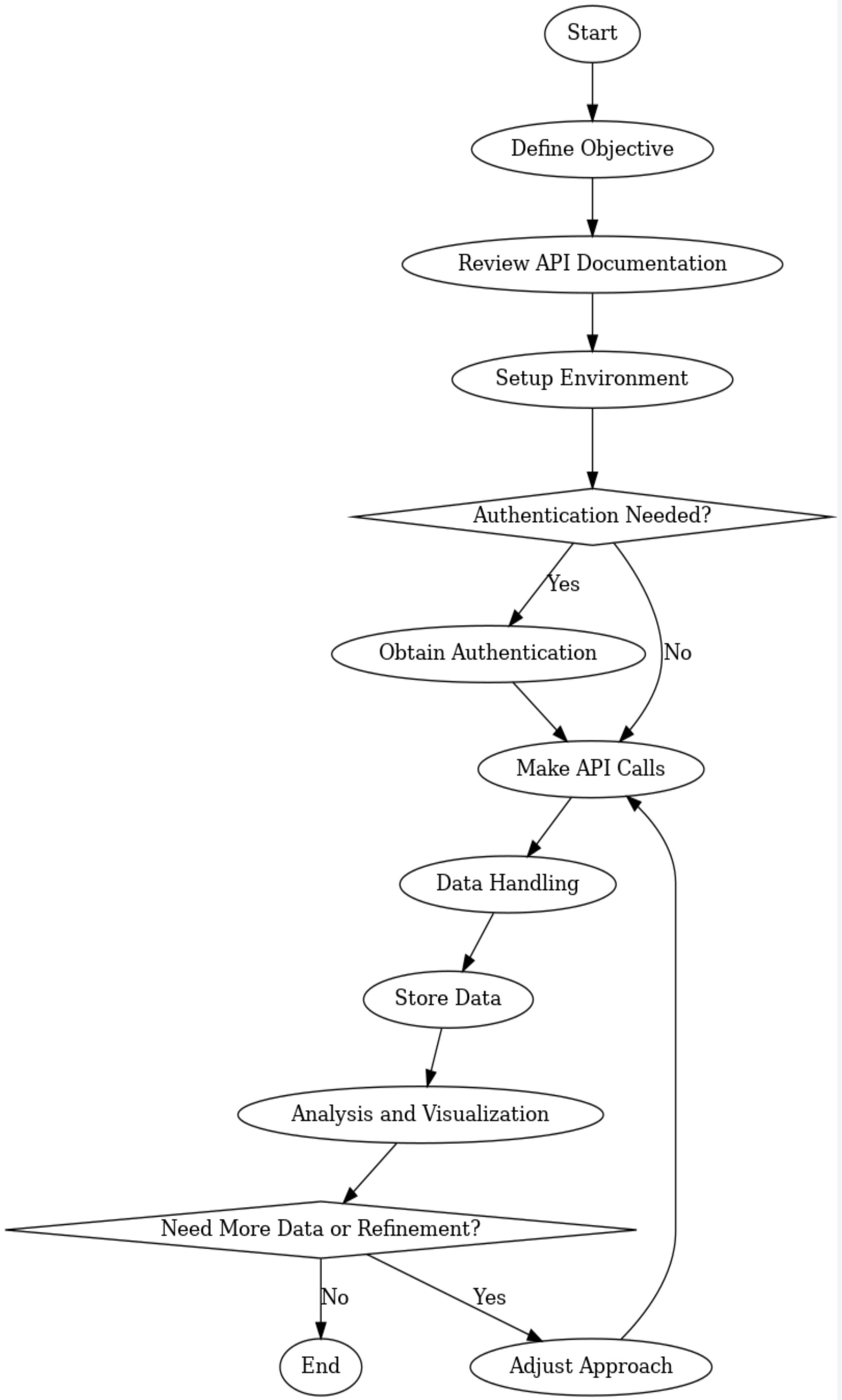
SpaceX API

Wikipedia

Jupyter Notebook - SQLite

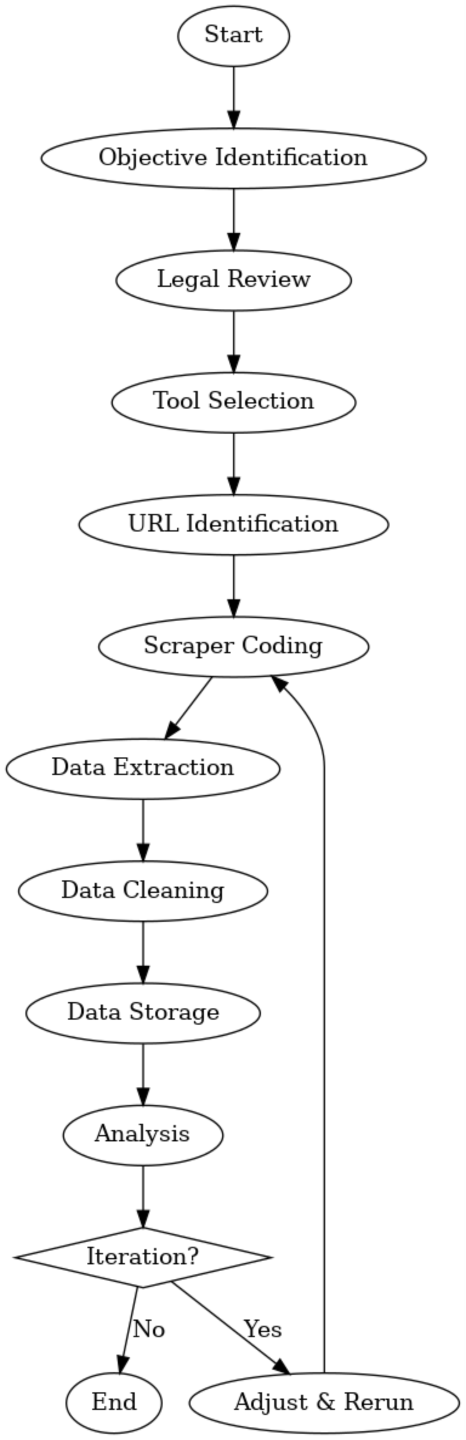
Data Collection – SpaceX API

- **Key phrases**
  - ○ Objective Definition: Identify target SpaceX data.
  - ○ API Docs: Review SpaceX API documentation.
  - ○ Environment Setup: Prepare for HTTP requests.
  - ○ Authentication: Check and apply if needed.
  - ○ API Requests: Execute GET calls to fetch data.
  - ○ Data Processing: Parse and handle API responses.
  - ○ Data Storage: Choose storage method for data.
- [Check the completed SpaceX API calls notebook at GitHub]()



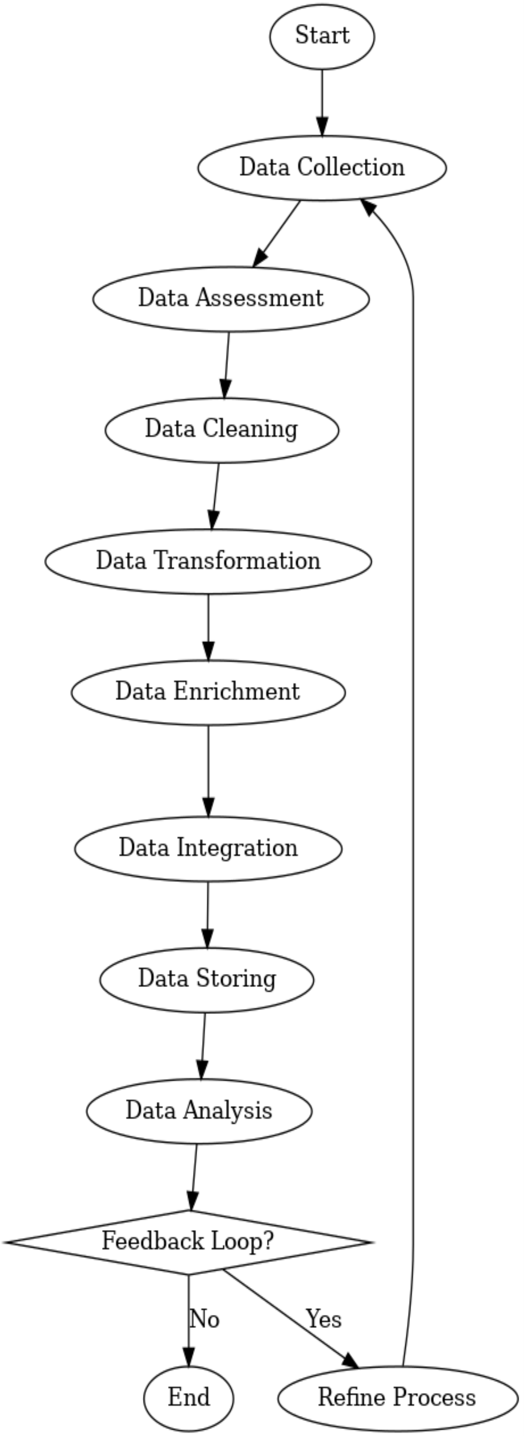
Data Collection - Scraping

- **Key Phrases:**
  - **Objective Identification**: Define what data to scrape.
  - **Legal Review**: Check website's legal restrictions on scraping.
  - **Tool Selection**: Choose libraries/tools (e.g., BeautifulSoup, Scrapy).
  - **URL Identification**: Determine the URLs of the pages to scrape.
  - **Scraper Coding**: Write script to extract the desired data.
  - **Data Extraction**: Run the script to collect data.
  - **Data Cleaning**: Refine the raw scraped data.
  - **Data Storage**: Save the cleaned data for analysis.
  - **Analysis**: Analyze the data for insights.
  - **Iteration**: Modify and rerun as needed.
- Check the completed web scraping notebook at GitHub



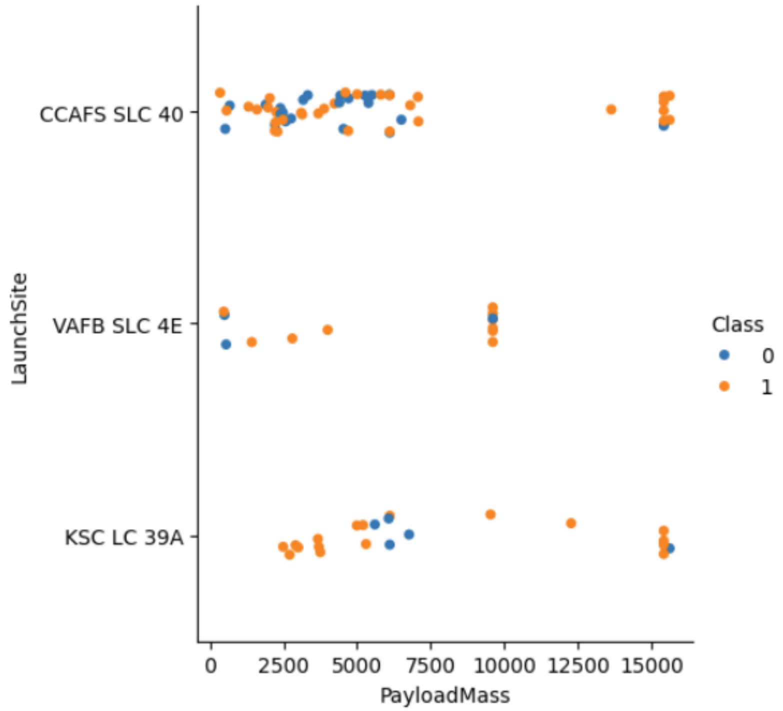
Data Wrangling

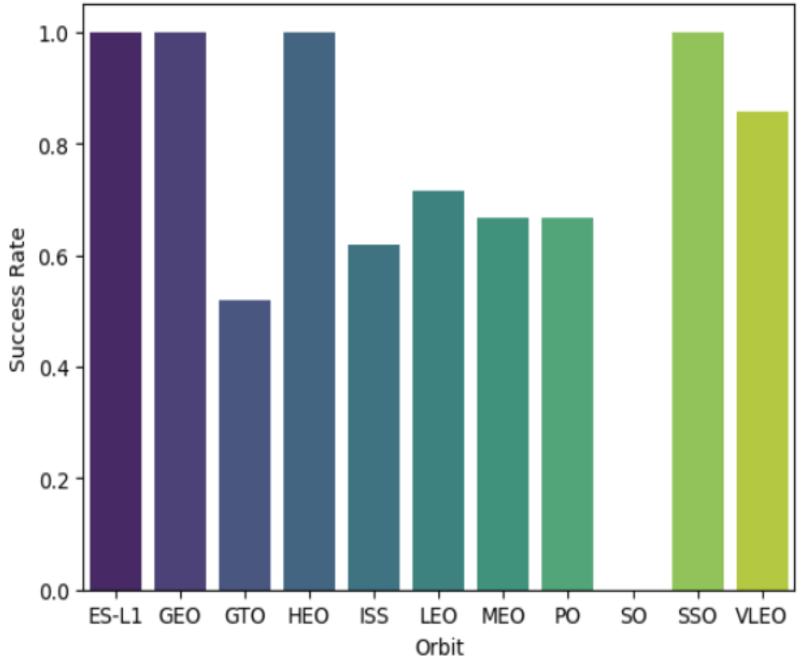
- Key phrases:
  - Data Collection: Gather raw data from various sources.
  - Data Assessment: Evaluate quality and structure.
  - Data Cleaning: Correct inaccuracies, remove duplicates.
  - Data Transformation: Normalize, format, and restructure.
  - Data Enrichment: Enhance data with additional sources.
  - Data Integration: Merge data from different sources.
  - Data Storing: Save wrangled data for analysis.
  - Data Analysis: Analyze to extract insights.
  - Feedback Loop: Refine processes based on outcomes
- Check the completed SQL notebook, and data wrangling notebook at GitHub.



EDA with Data Visualization

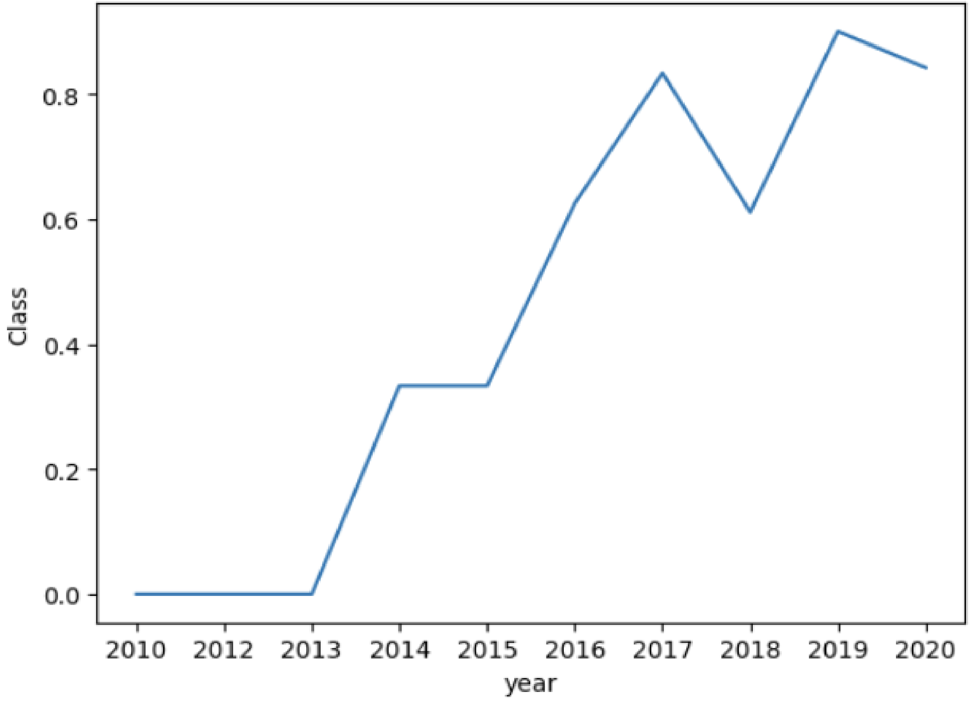
- Scatter plot shows the relationship between launch sites and their payload mass(kg). VAFB SLC 4E had no launch with payload mass over 10000 kg.
- Bar chart shows the relationship between success rate and orbit type. ES-L1, GEO, HEO and SSO see higher success rates.
- Check the completed exploring and preparing data notebook at GitHub

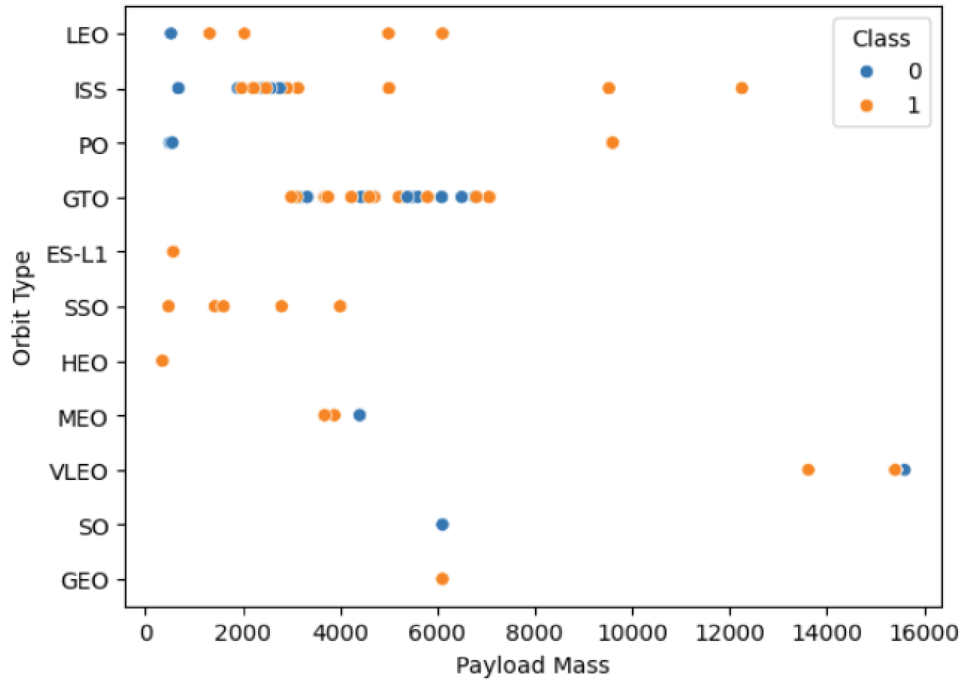




EDA with Data Visualization

- Line chart shows the average launch success trend from 2010 to 2020.
- Scatter plot shows the relationship between payload and orbit type. With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. For GTO, we cannot distinguish this well.
- Check the completed exploring and preparing data notebook at GitHub





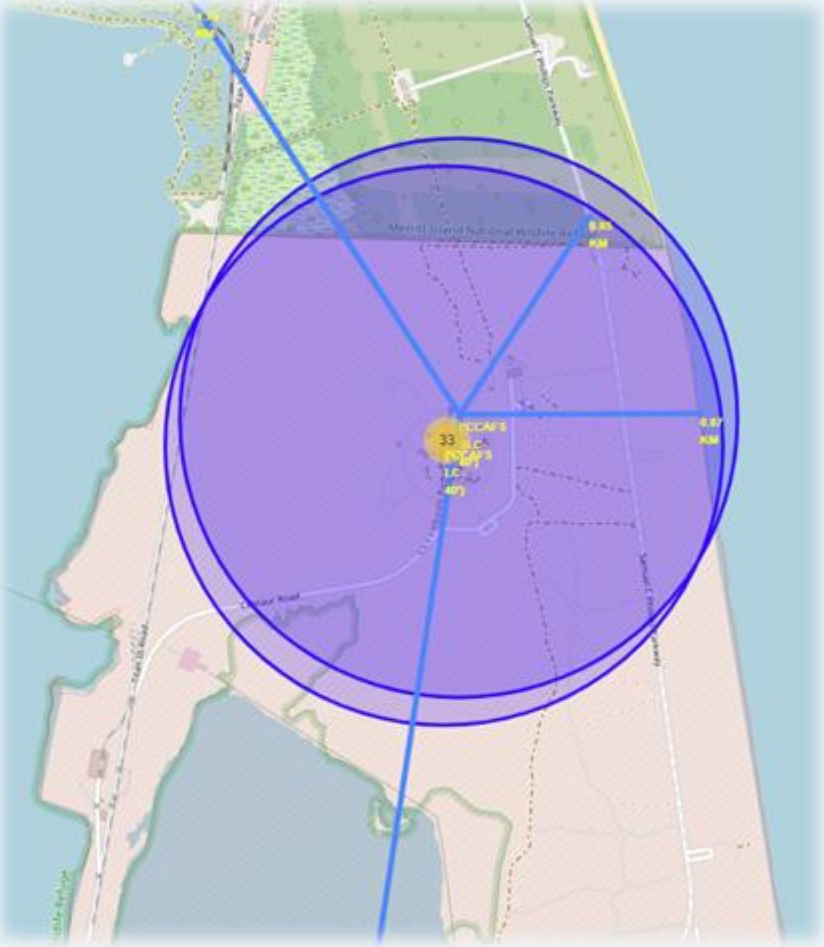
EDA with SQL

- Setup
  - Downloaded dataset CSV.
  - Installed sqlalchemy and ipython-sql.
  - Created SQLite database and table SPACEXTBL.
- Data Preparation
  - Imported SpaceX data into SPACEXTBL.
  - Created SPACEXTABLE to eliminate blank rows.
- Check the completed SQL notebook at GitHub.

- Queries
  - Queried unique launch sites.
  - Selected first 5 'CCA' launch sites.
  - Calculated NASA (CRS) total payload mass.
  - Averaged payload for F9 v1.1 booster.
  - Found date of first successful ground pad landing.
  - Listed boosters with successful drone ship landings, 4000-6000 kg payload.
  - Counted successful vs. failure mission outcomes.
  - Identified boosters with maximum payload mass.
  - Listed 2015 failure outcomes on drone ship, by month.
  - Ranked landing outcomes by count, descending order.

Build an Interactive Map with Folium

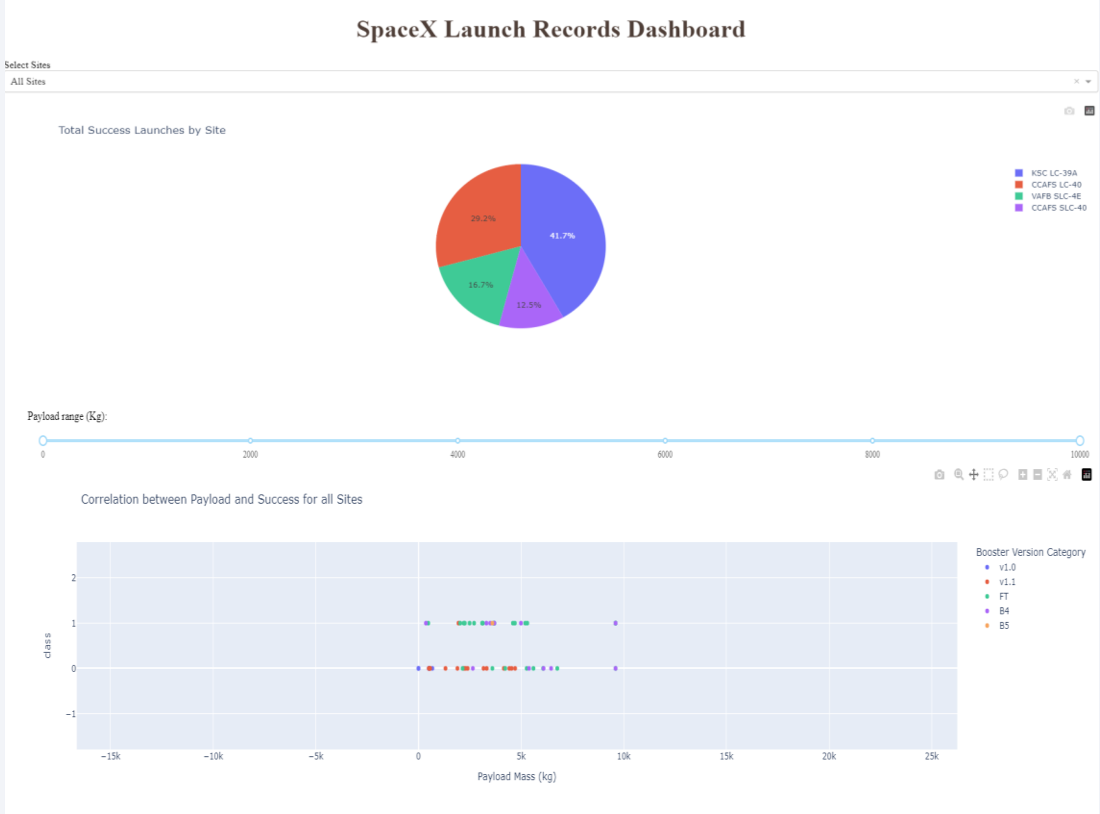
- Launch Sites Marked: Identified with markers and circles for geographic visualization.
- Launch Outcomes: Color-coded markers within clusters to assess success rates.
- Proximity Analysis: Distance lines to cities, railways, highways, and coastlines evaluate strategic placement.
- Key Insights:
- Equator Proximity: Maximizes launch efficiency.
- Coastal Locations: Offers safety and recovery benefits.
- Infrastructure Balance: Ensures logistic feasibility while maintaining safety distances from populated areas.
- See [the completed interactive map with Folium map]() at GitHub



Build a Dashboard with Plotly Dash

- Features:
  - ○ Launch Site Filter: Dropdown for site-specific or aggregate data viewing.
  - ○ Success Rate Visualization: Pie chart comparing launch outcomes.
  - ○ Payload Range Selector: Slider to explore various payload masses.
  - ○ Success Correlation Graph: Scatter plot linking payload mass to launch success.
- Impact:
  - ○ Enhances user interaction and data discovery.
  - ○ Simplifies success rate assessment and comparison.
  - ○ Facilitates targeted payload mass analysis.
  - ○ Demonstrates the relationship between payload and success visually.

- Check the completed Plotly Dash lab at GitHub



SpaceX Launch Records Dashboard

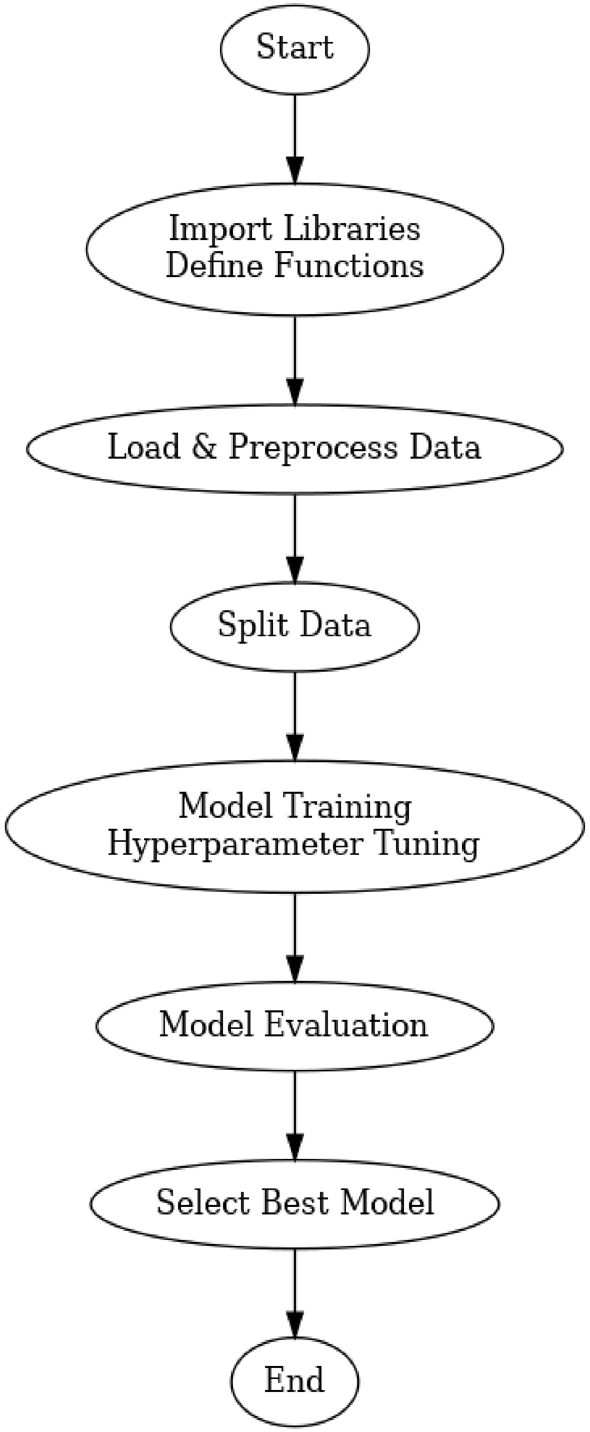
Select Sites

All Sites

Payload range (Kg):

Predictive Analysis (Classification)

- Model Development Highlights
  - Setup: Imported ML libraries; standardized data from CSVs.
  - Splitting: Partitioned data into training/test sets.
  - Tuning / Training: Utilized GridSearchCV on Logistic Regression, SVM, Decision Tree, KNN.
  - Evaluation: Measured model success with accuracy scores and confusion matrices.
  - Selection: Decision Tree emerged as top model based on accuracy.
- Key Achievements:
  - Optimized model parameters.
  - Established model accuracy benchmarks.
  - Prioritized Decision Tree Classifier for superior performance.
- Strategy:
  - Ensured best model choice through comparative analysis.
  - Maximized performance with hyperparameter tuning.
  - Validated findings with detailed evaluation techniques.
- Check the completed predictive analysis lab at GitHub.

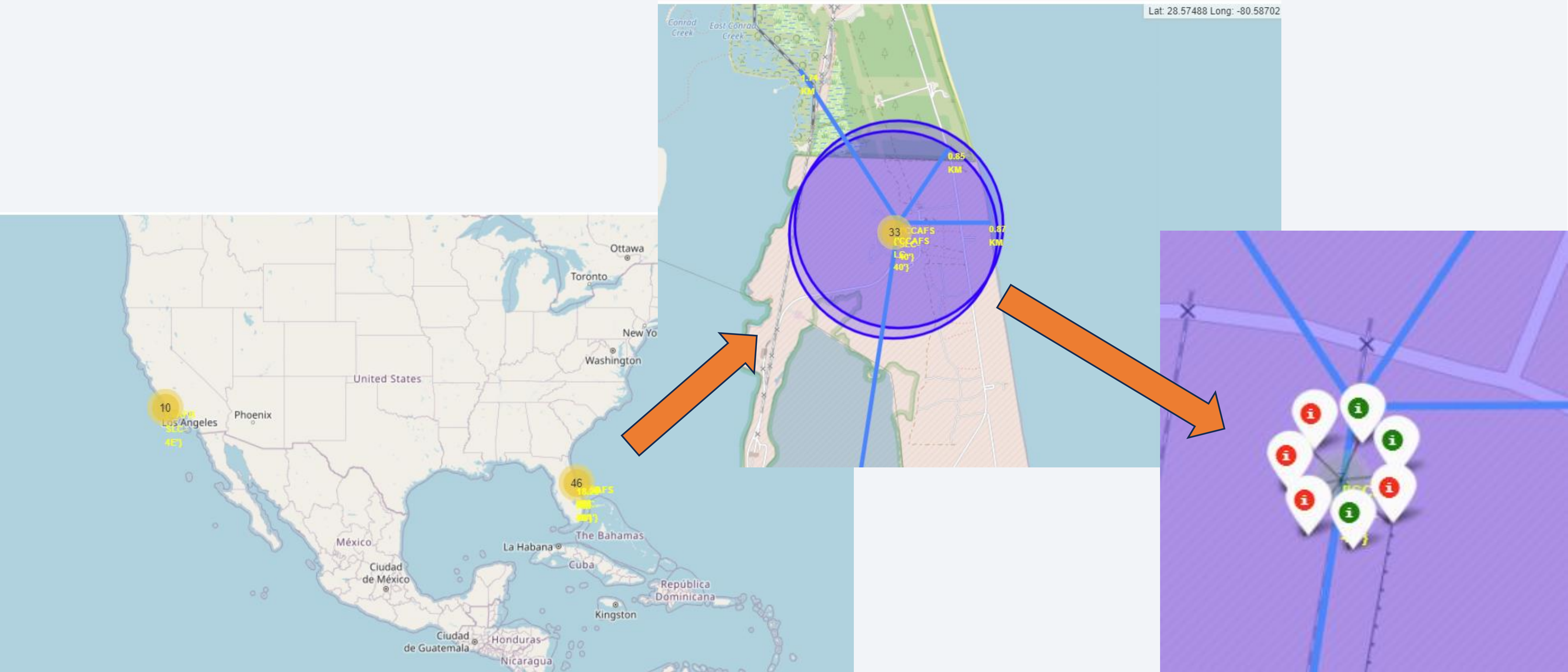


Results

- Exploratory data analysis results
    - SpaceX launches its rockets at 4 sites near the coast.
    - KSC LC-39A launch site has highest percentage of success rate compared to other launch sites.
    - Rockets with Booster version FT or B4 experience more success landing.
    - Rockets with payload mass over 6000kg see fewer successful landings.

Results

- Interactive analytics demo in screenshots



Results

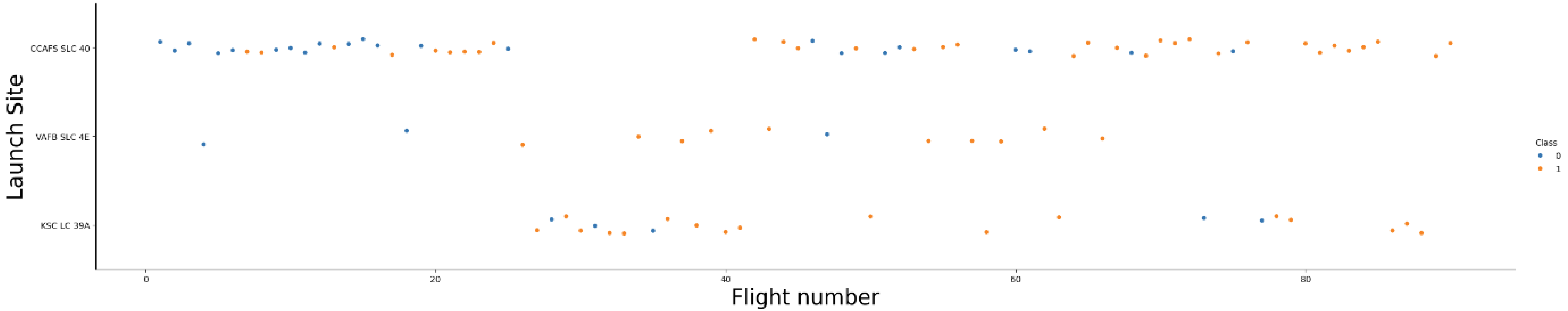
- Predictive analysis results
  - Decision Tree Classifier performs the best at predicting landing outcome.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

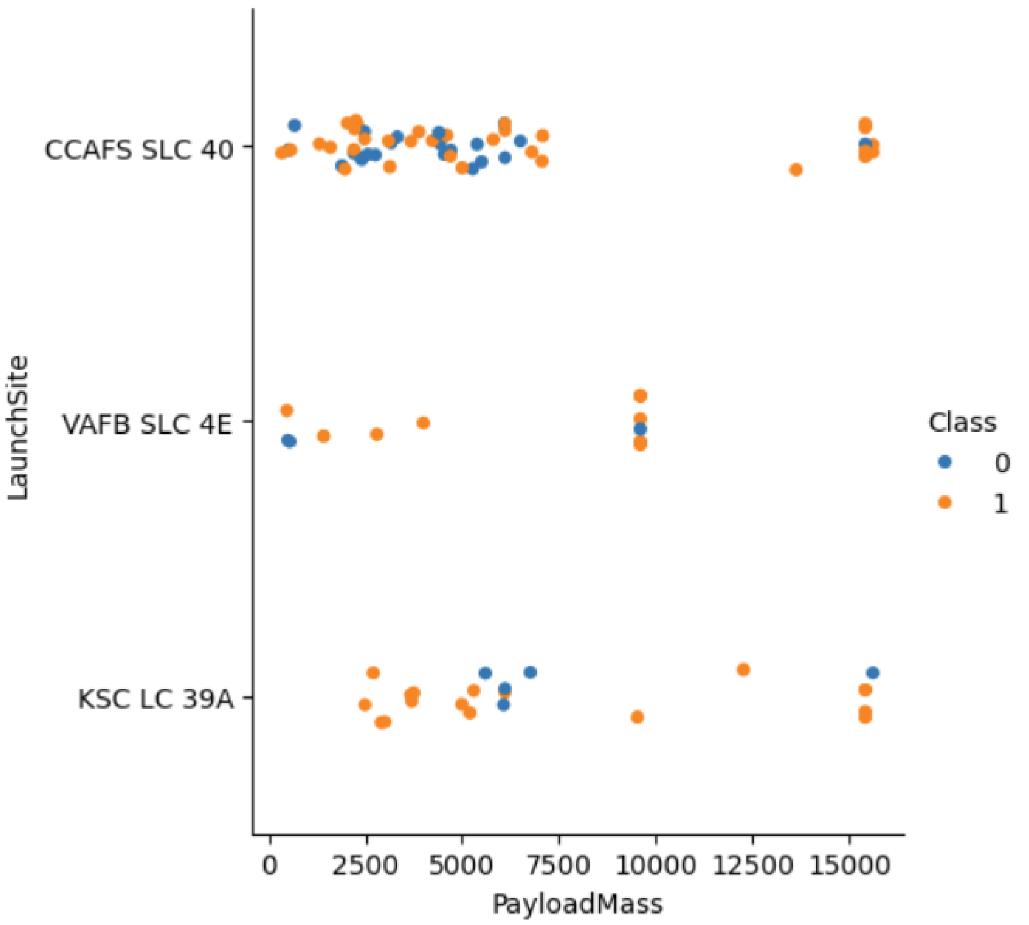
- Scatter plot of Flight Number vs. Launch Site



More launches took place at the CCAFS SLC 40 launch site after flight number 40. No rocket launches at VAFB SLC 4E launch site after Flight number 60.

Payload vs. Launch Site

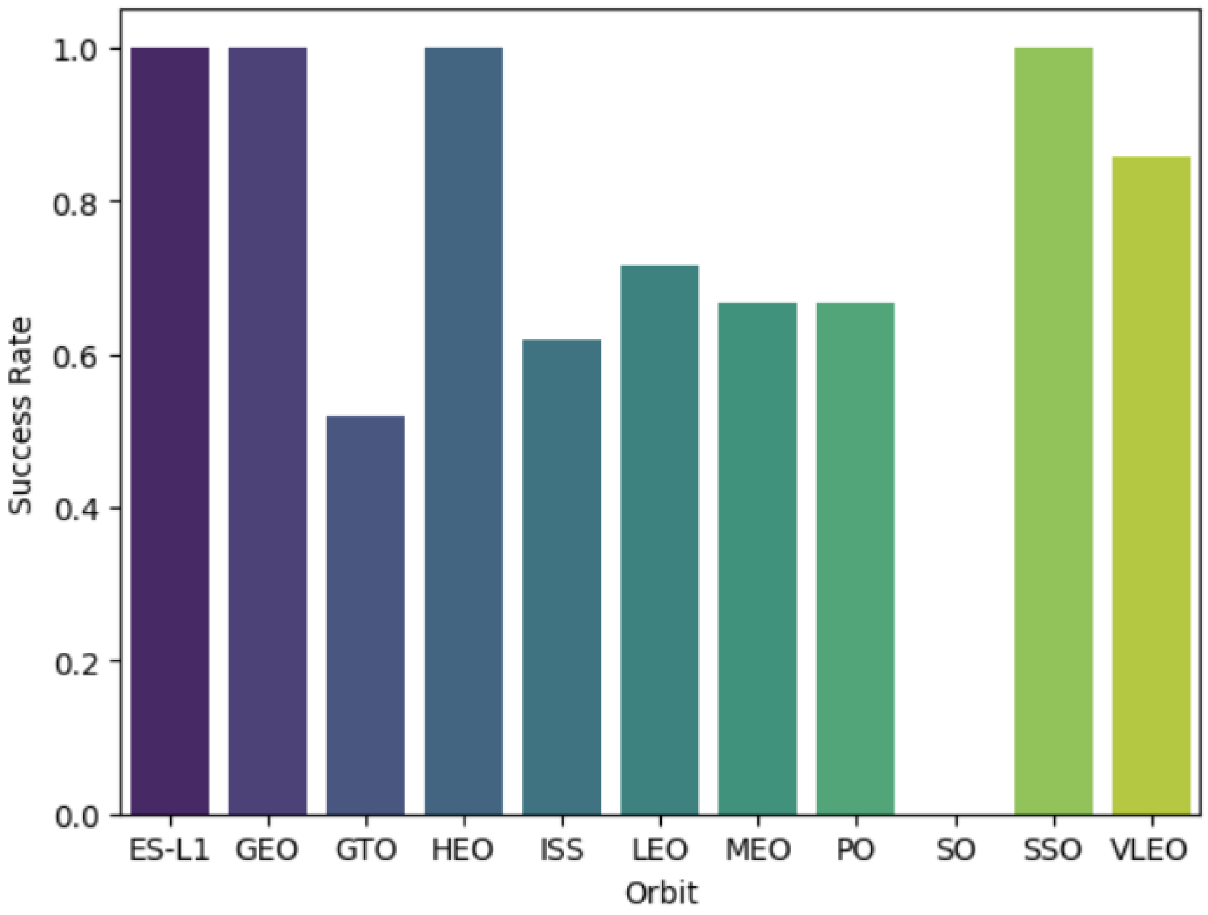
- Scatter plot of Payload vs. Launch Site



For the VAFB-SLC 4E launchsite, there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type

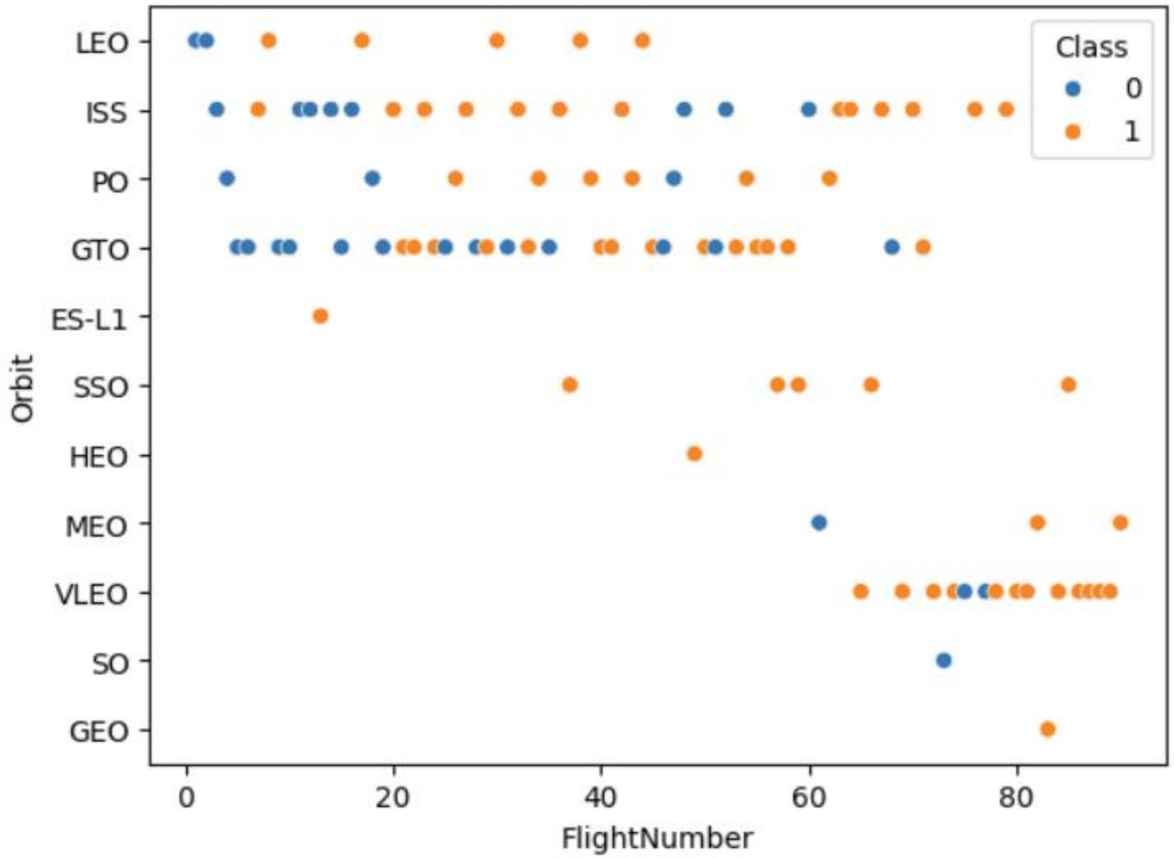
- Bar chart for the success rate of each orbit type



Bar chart shows the relationship between success rate and orbit type. ES-L1, GEO, HEO, and SSO see higher success rates.

Flight Number vs. Orbit Type

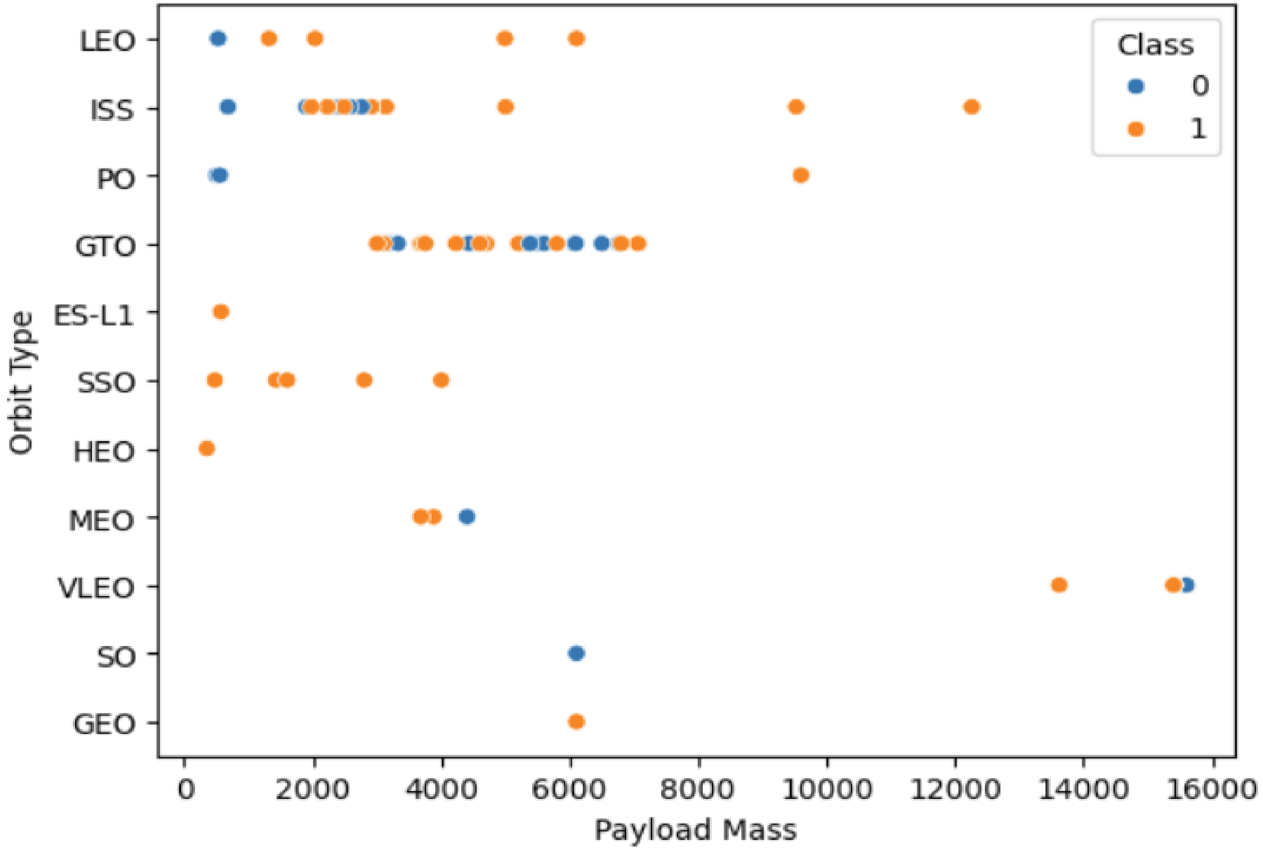
- Show a scatter point of Flight number vs. Orbit type



In the LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

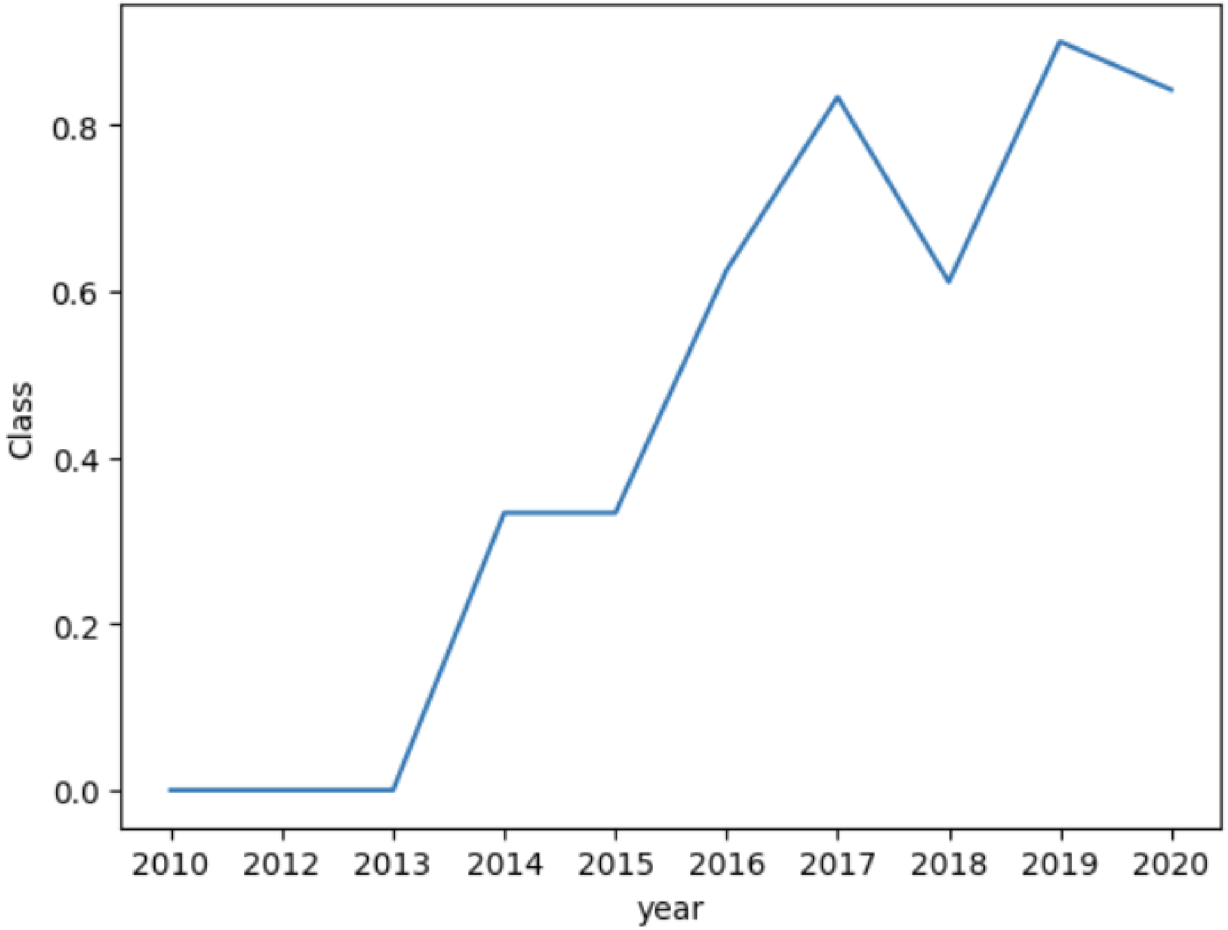


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

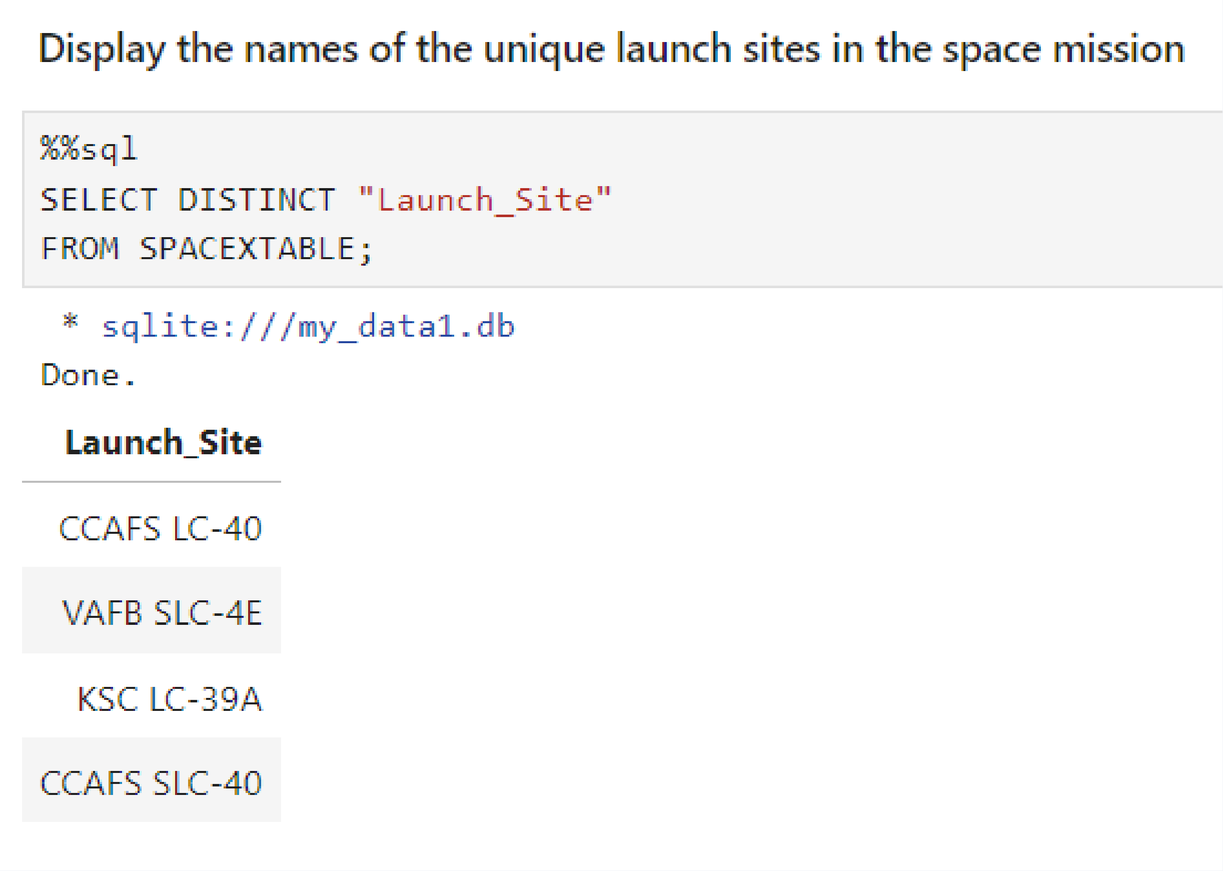
- Show a line chart of yearly average success rate



The success rate since 2013 kept increasing utill 2017 (stable in 2014) and after 2015 it started increasing.

All Launch Site Names

- The SQL query lists unique SpaceX launch sites from the SPACEXTABLE, eliminating duplicates and providing an overview of the company's launch locations.



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

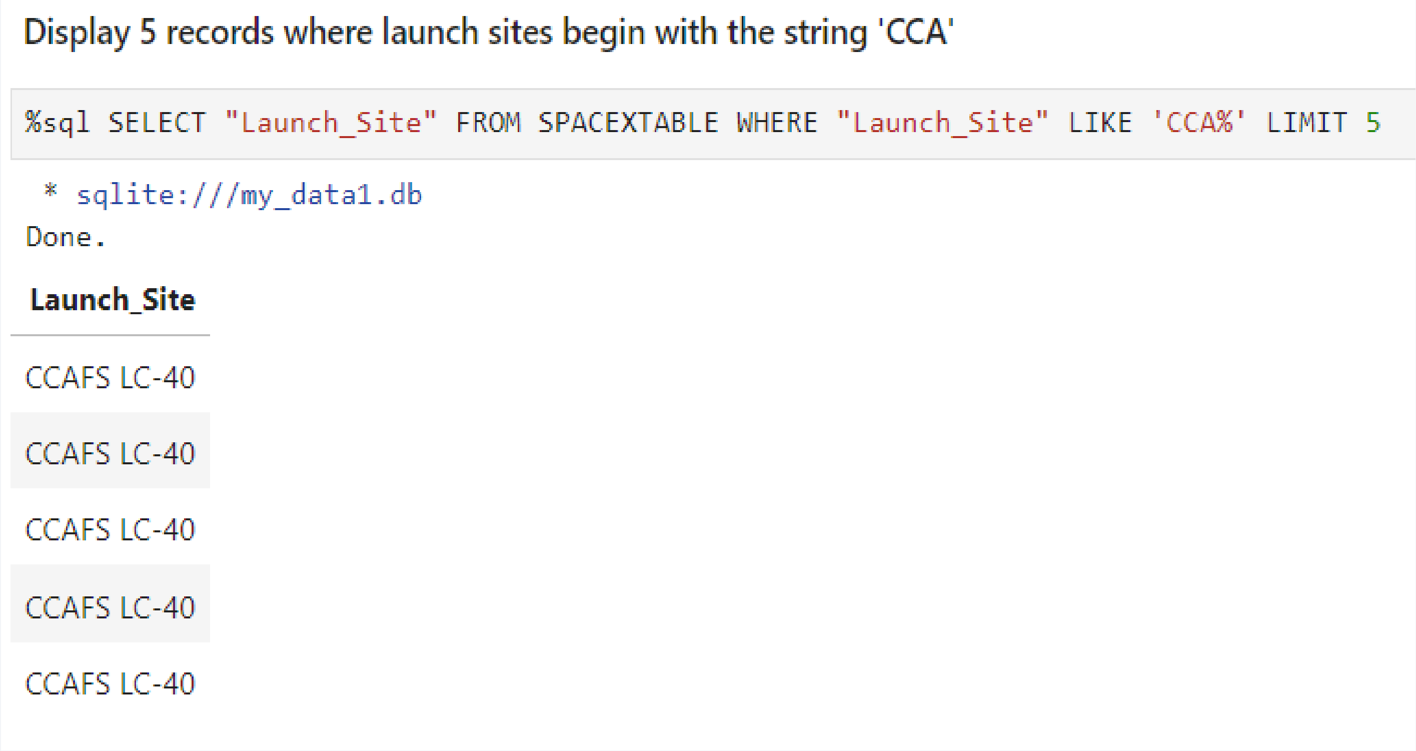
```
%%sql
SELECT DISTINCT "Launch_Site"
FROM SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'

- The query returns up to 5 launch sites from SPACEXTABLE that start with 'CCA', showcasing specific SpaceX launch locations matching this pattern.



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

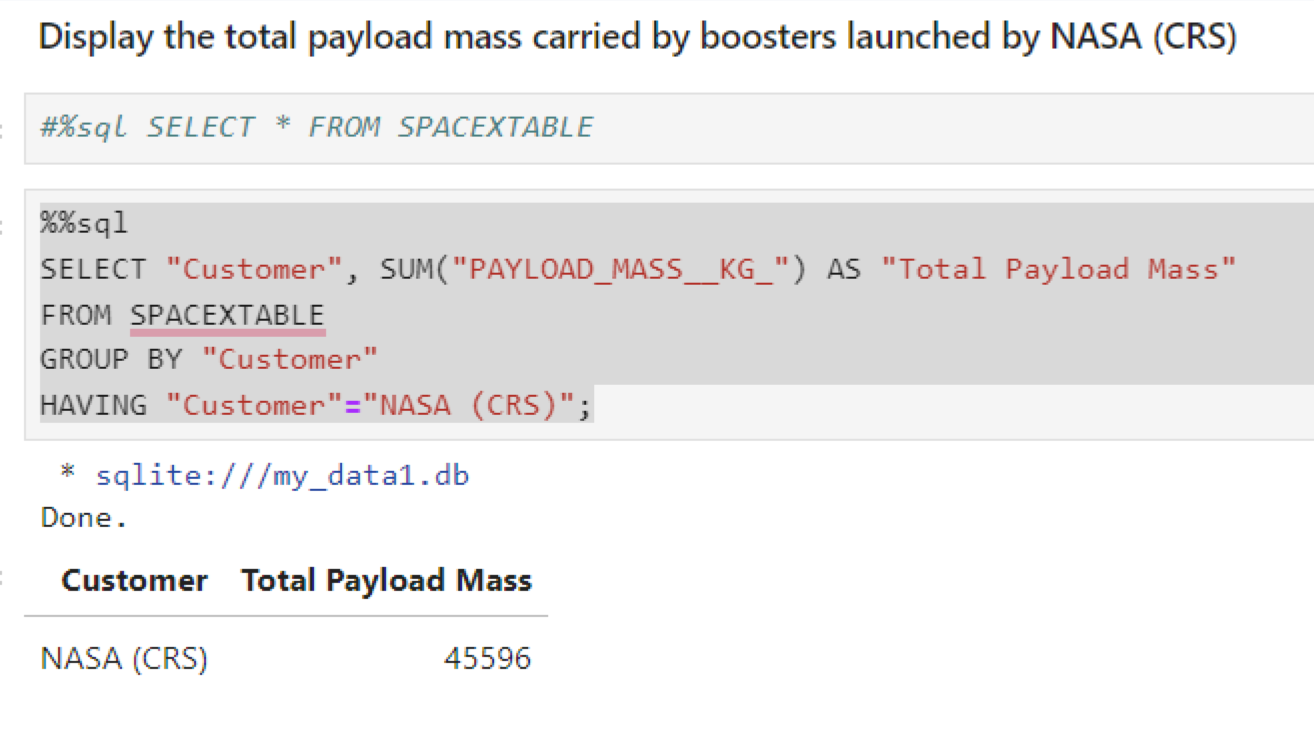
```
%sql SELECT "Launch_Site" FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- The query calculates the total payload mass SpaceX has launched for NASA (CRS), summing up the weights for all relevant missions.



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

```
#%sql SELECT * FROM SPACEXTABLE
```

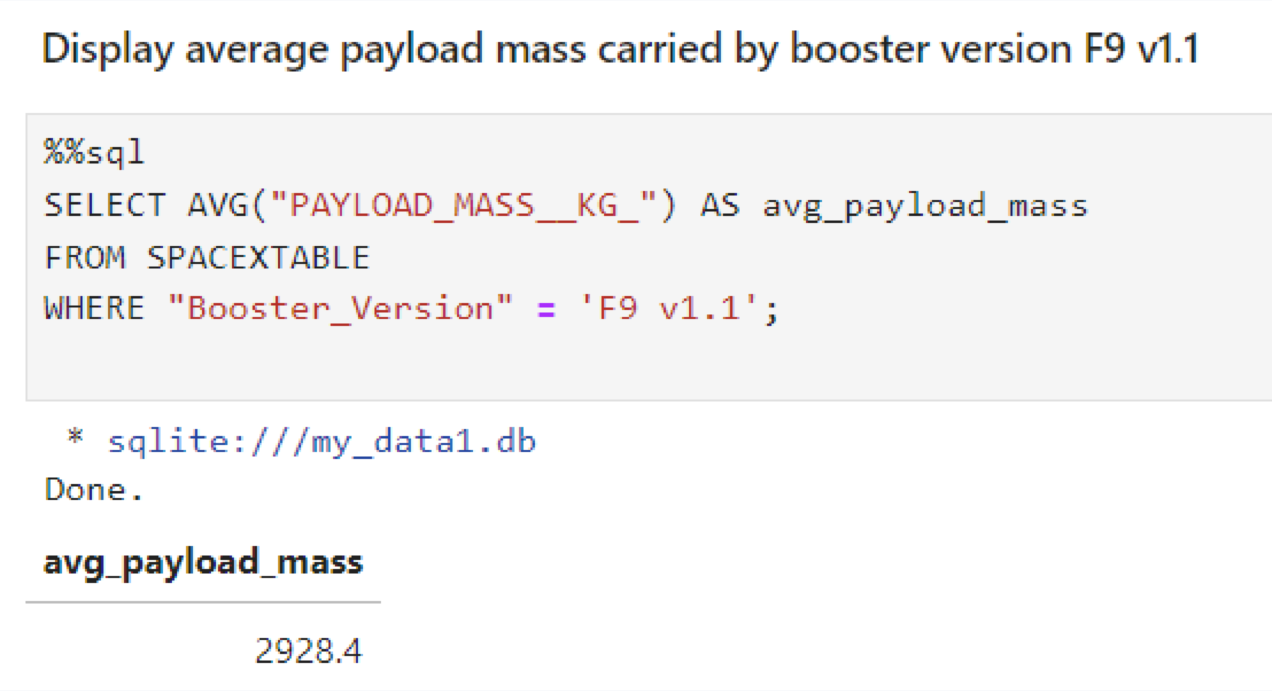
```
%%sql
SELECT "Customer", SUM("PAYLOAD_MASS__KG_") AS "Total Payload Mass"
FROM SPACEXTABLE
GROUP BY "Customer"
HAVING "Customer"="NASA (CRS)";
```

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

Customer	Total Payload Mass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- The query determines the average payload mass launched by the F9 version 1.1 booster.



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

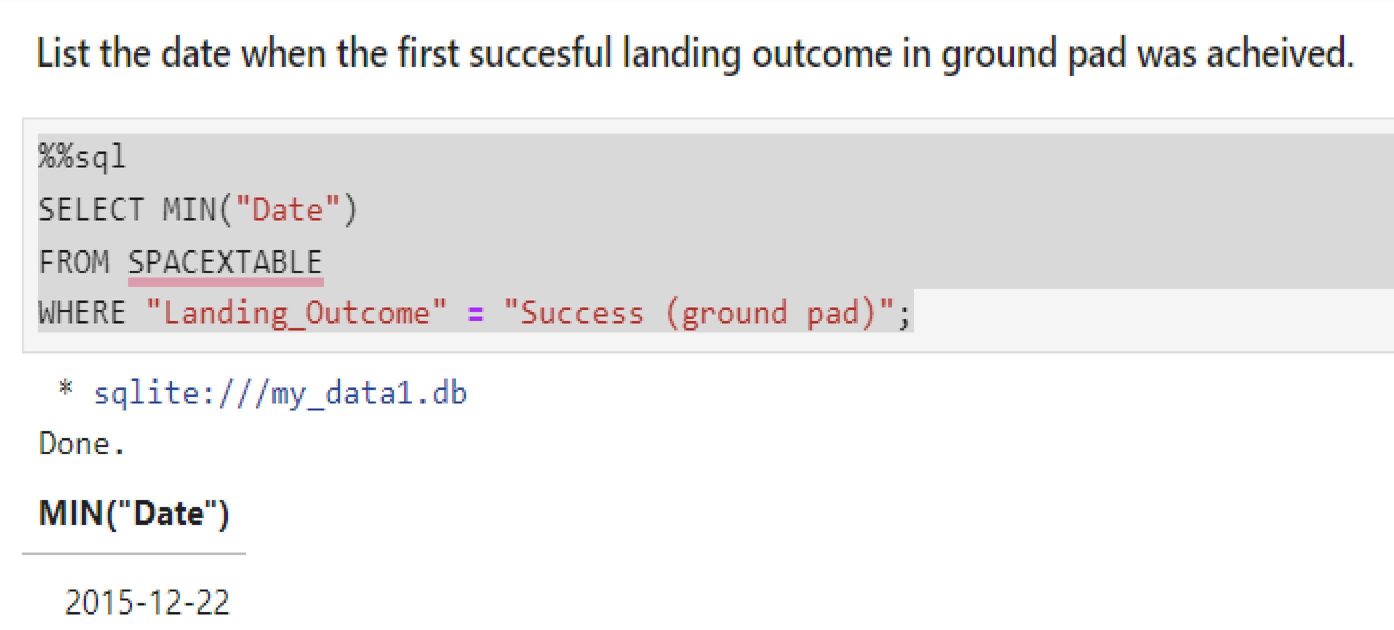
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS avg_payload_mass
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

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

avg_payload_mass
2928.4

First Successful Ground Landing Date

- The query finds the earliest date a SpaceX landing was successfully achieved on a ground pad.



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

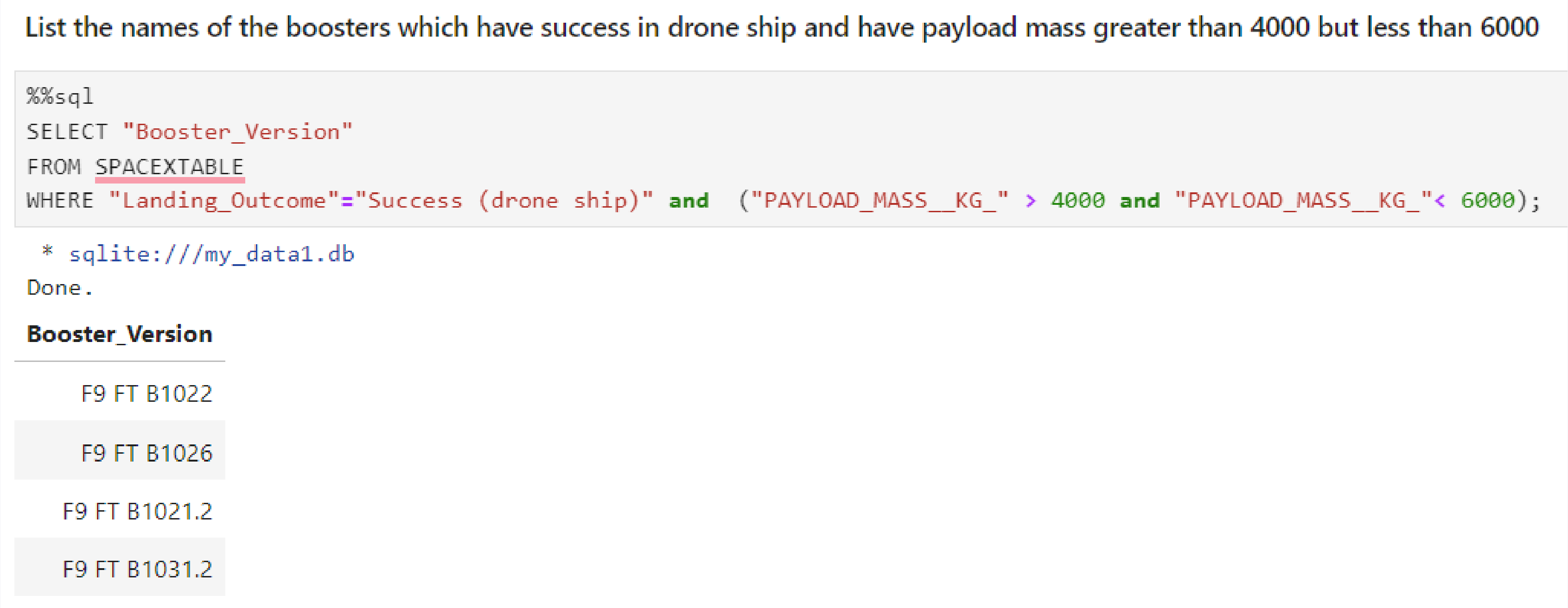
```
%%sql
SELECT MIN("Date")
FROM SPACEXTABLE
WHERE "Landing_Outcome" = "Success (ground pad)";
```

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

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query lists booster versions that successfully landed on a drone ship carrying payloads between 4000 and 6000 kg.



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

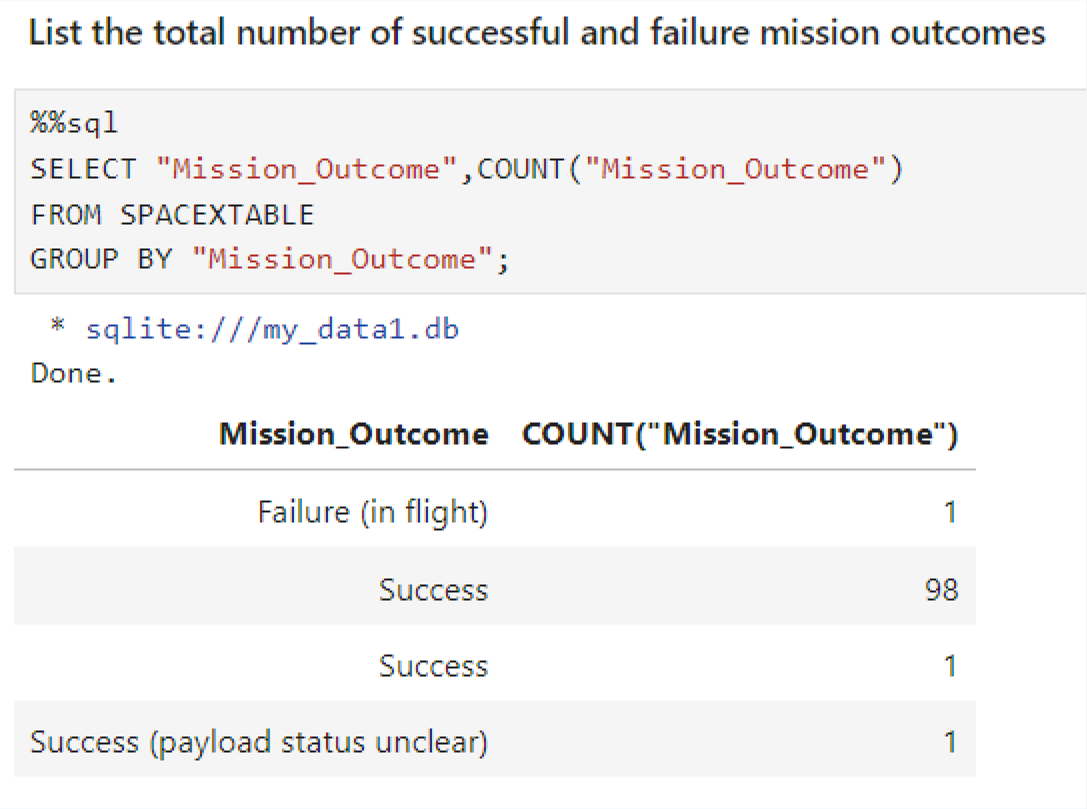
```
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome"="Success (drone ship)" and  ("PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_"< 6000);
```

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

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

Total Number of Successful and Failure Mission Outcomes

- The query counts the occurrences of each type of mission outcome in the SPACEXTABLE, grouping the results by the outcome type.



List the total number of successful and failure mission outcomes

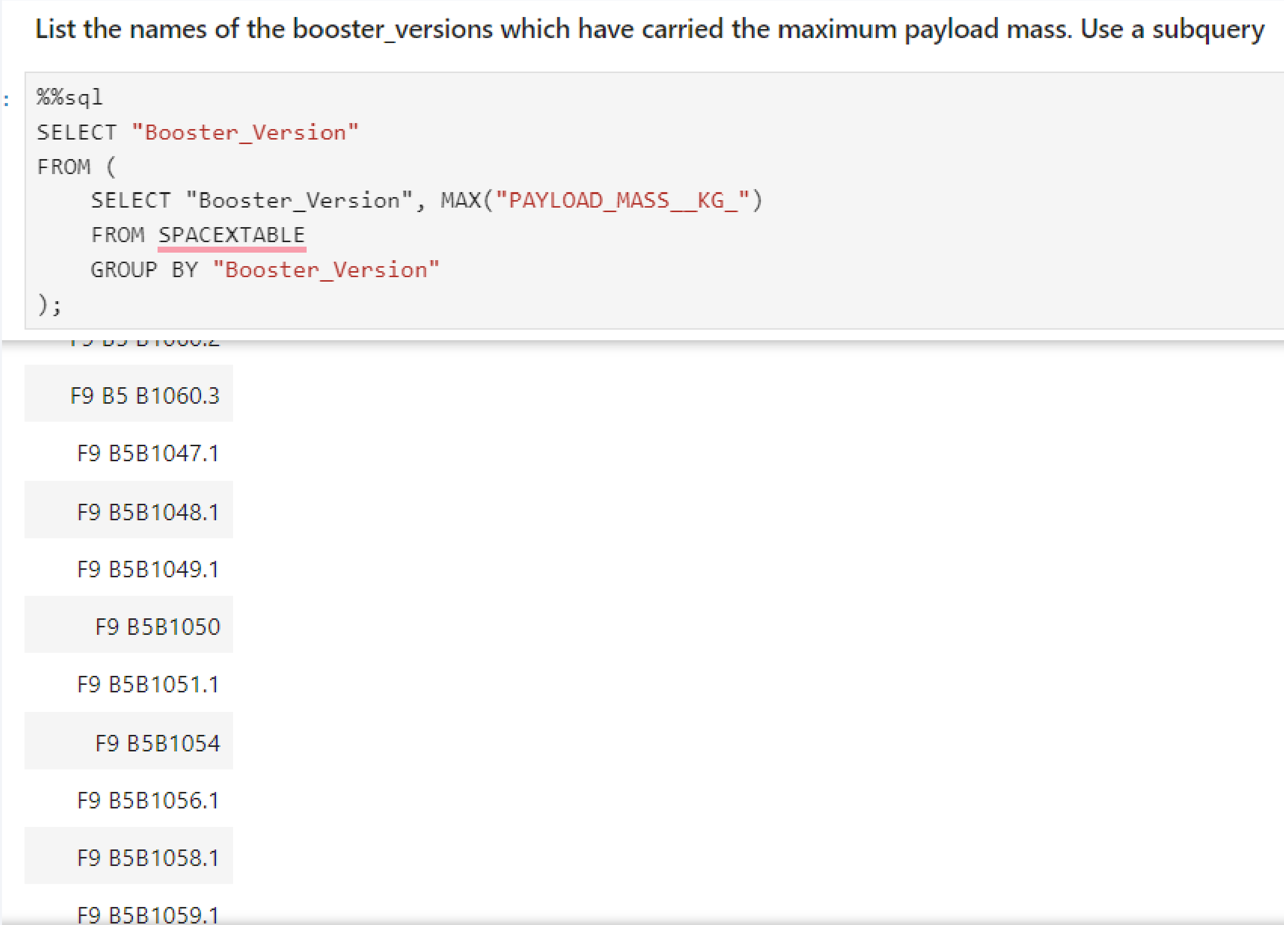
```
%%sql
SELECT "Mission_Outcome",COUNT("Mission_Outcome")
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

- The query identifies booster versions along with the maximum payload mass they have carried, grouped by booster version.



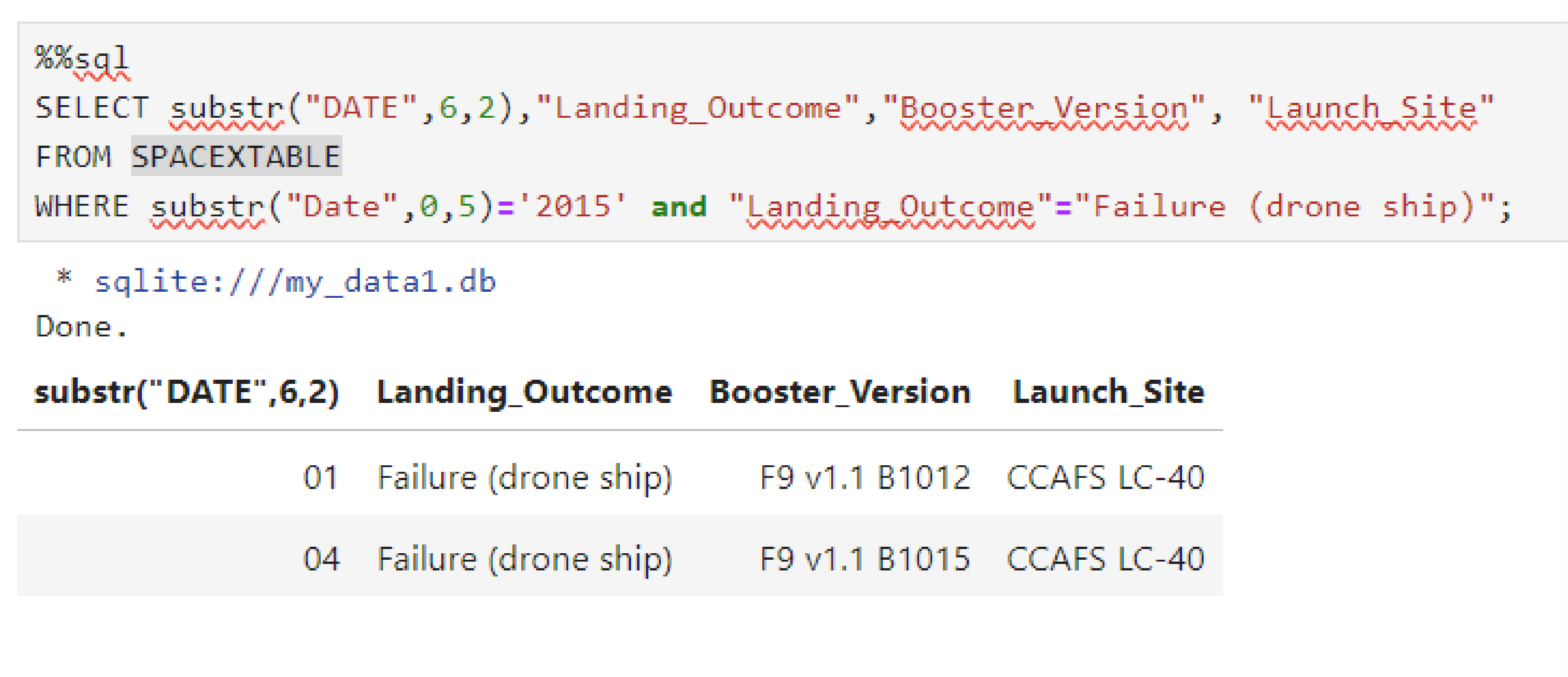
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql
SELECT "Booster_Version"
FROM (
    SELECT "Booster_Version", MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
    GROUP BY "Booster_Version"
);
```

```
 F9 B5 B1060.2
 F9 B5 B1060.3
  F9 B5B1047.1
  F9 B5B1048.1
  F9 B5B1049.1
    F9 B5B1050
  F9 B5B1051.1
    F9 B5B1054
  F9 B5B1056.1
  F9 B5B1058.1
  F9 B5B1059.1
```

2015 Launch Records

- The query lists the month, landing outcome, booster version, and launch site for all failed drone ship landings in 2015.



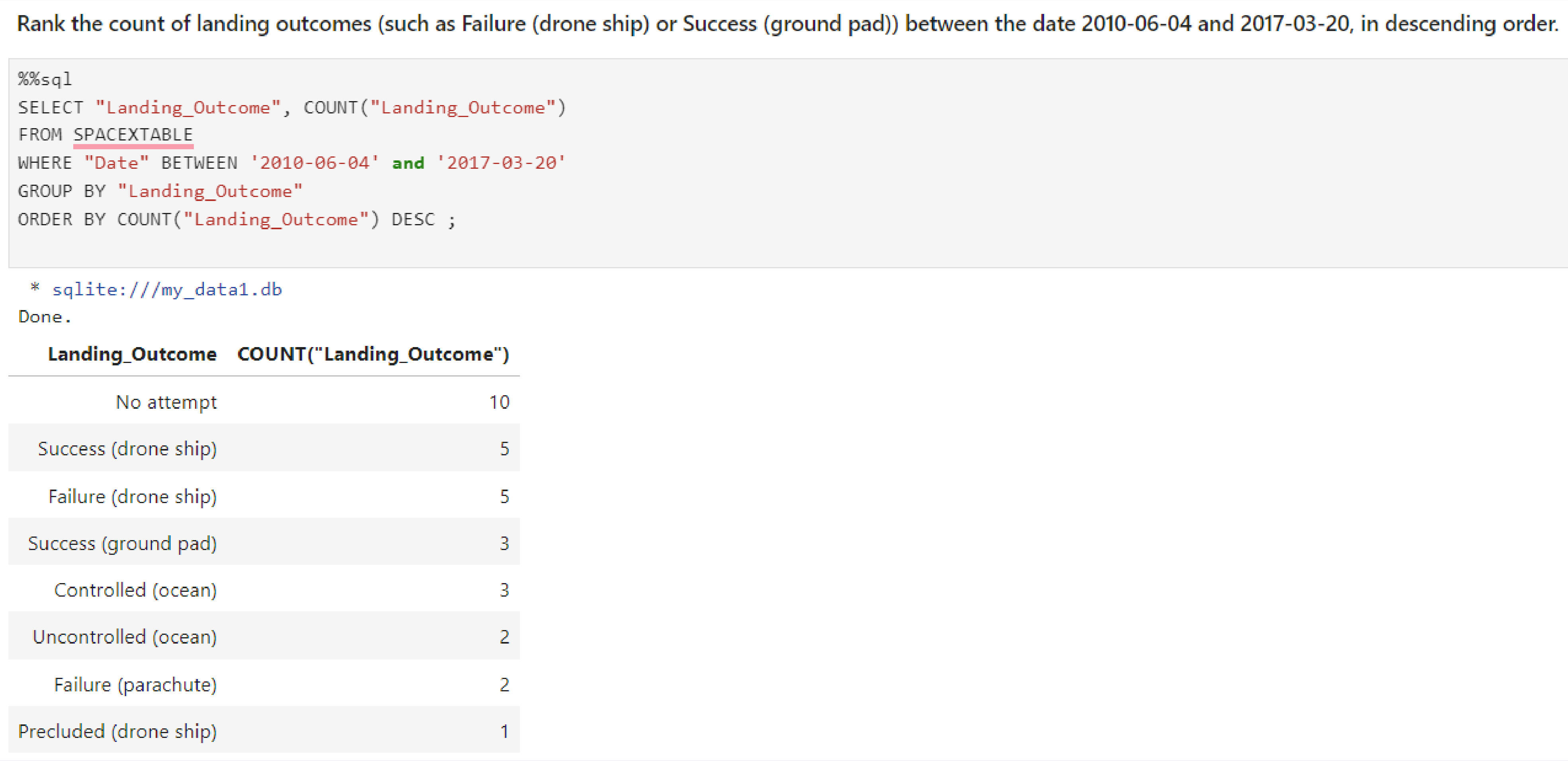
```
%%sql
SELECT substr("DATE",6,2),"Landing_Outcome","Booster_Version", "Launch_Site"
FROM SPACEXTABLE
WHERE substr("Date",0,5)='2015' and "Landing_Outcome"="Failure (drone ship)";
```

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

substr("DATE",6,2)	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

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

- The query counts landing outcomes for missions between June 4, 2010, and March 20, 2017, grouping and ordering them by frequency in descending order.



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("Landing_Outcome")
FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY COUNT("Landing_Outcome") DESC ;
```

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

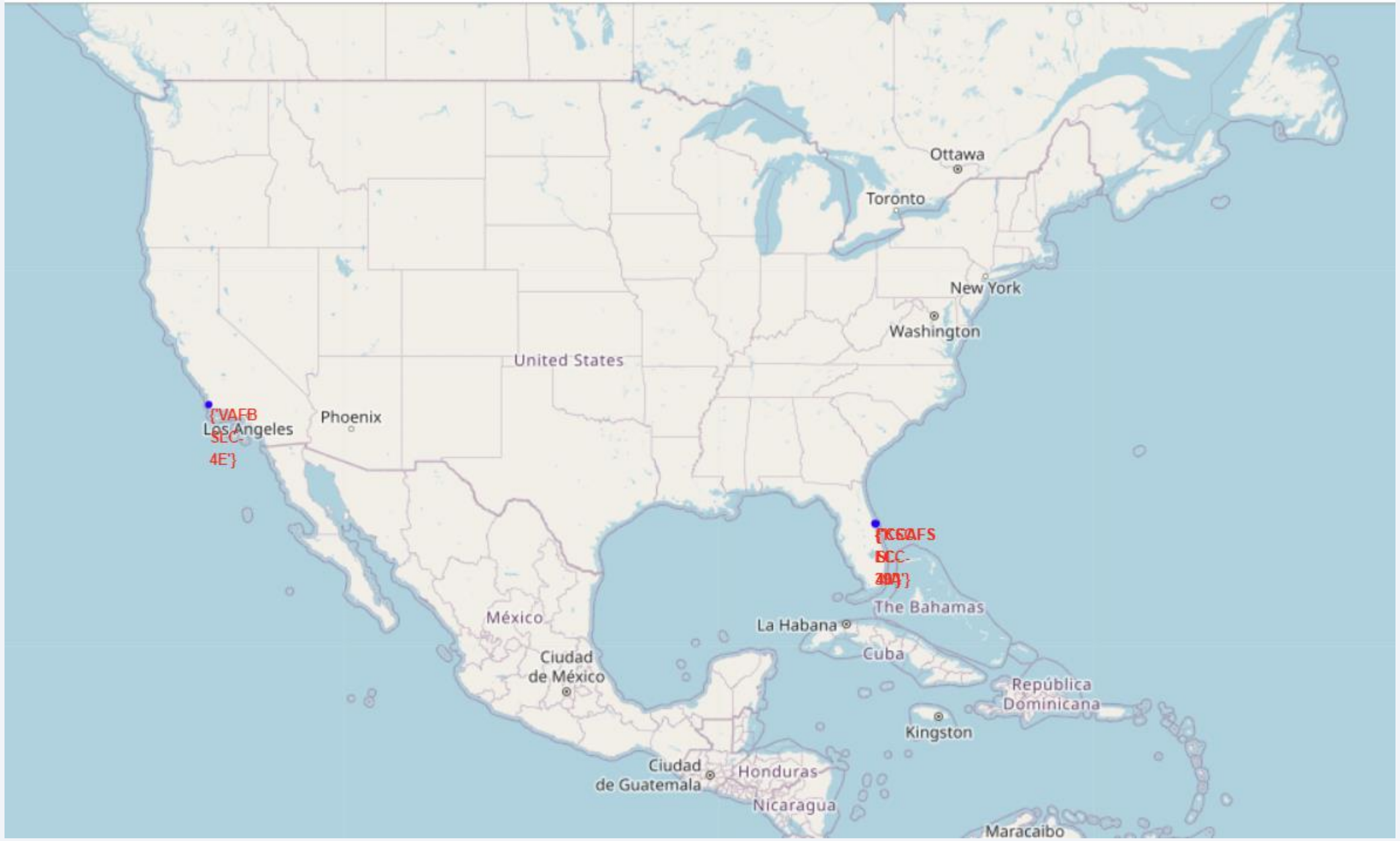
Landing_Outcome	COUNT("Landing_Outcome")
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

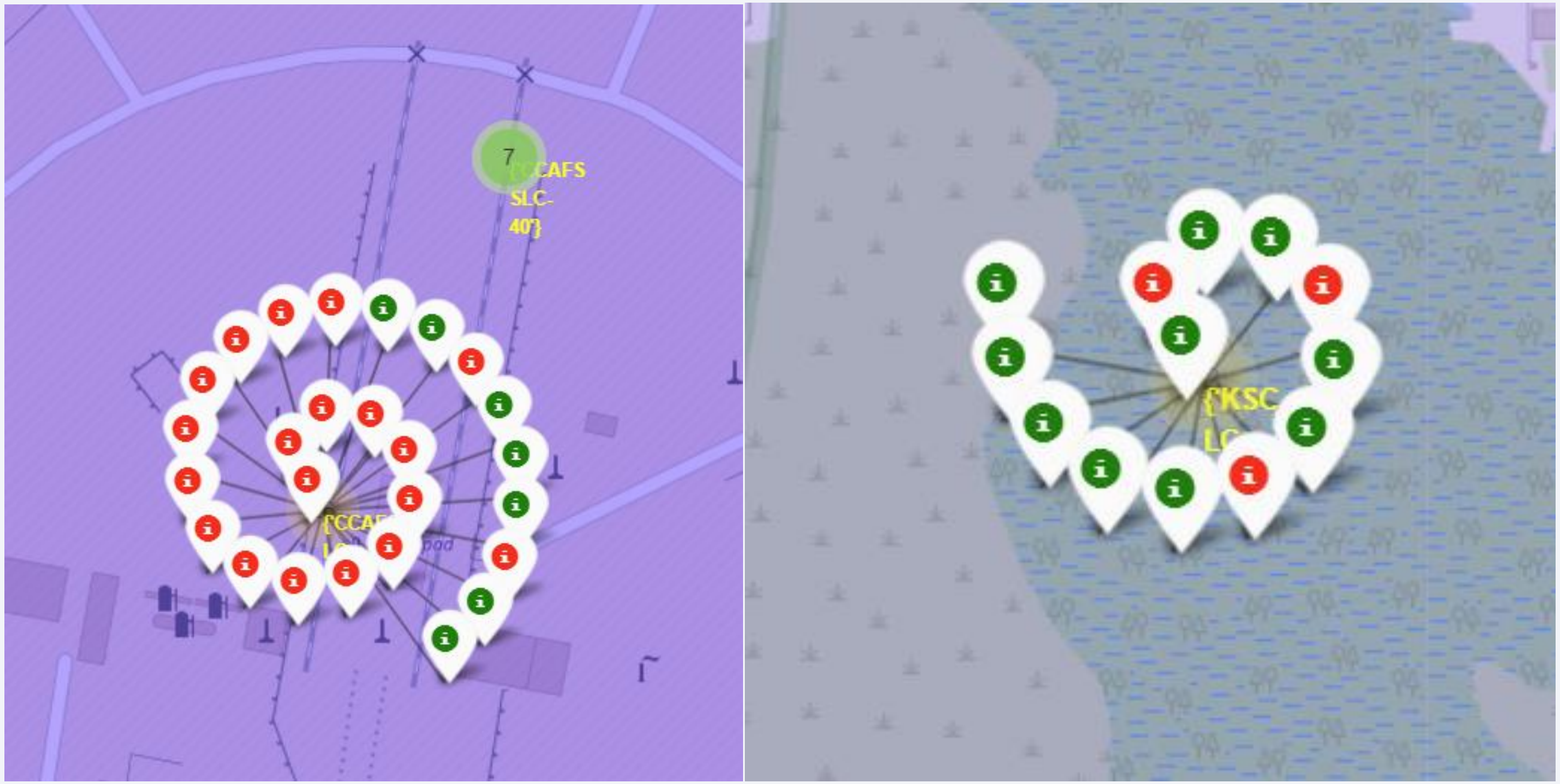
Mark launch sites on the map

- Create a Folium map centered at NASA's coordinates, then iterates over launch sites, adding a circle and marker for each. The circle highlights the area, and the marker displays the site's name on click, visually representing SpaceX launch locations.
- All launch sites are in proximity to the Equator line and the coast.



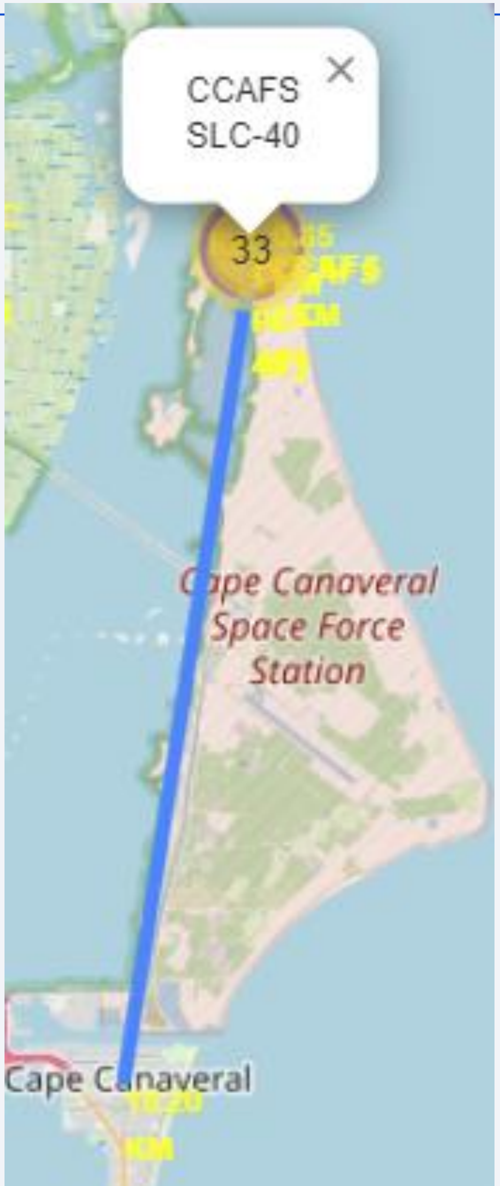
Color-labeled launch outcomes

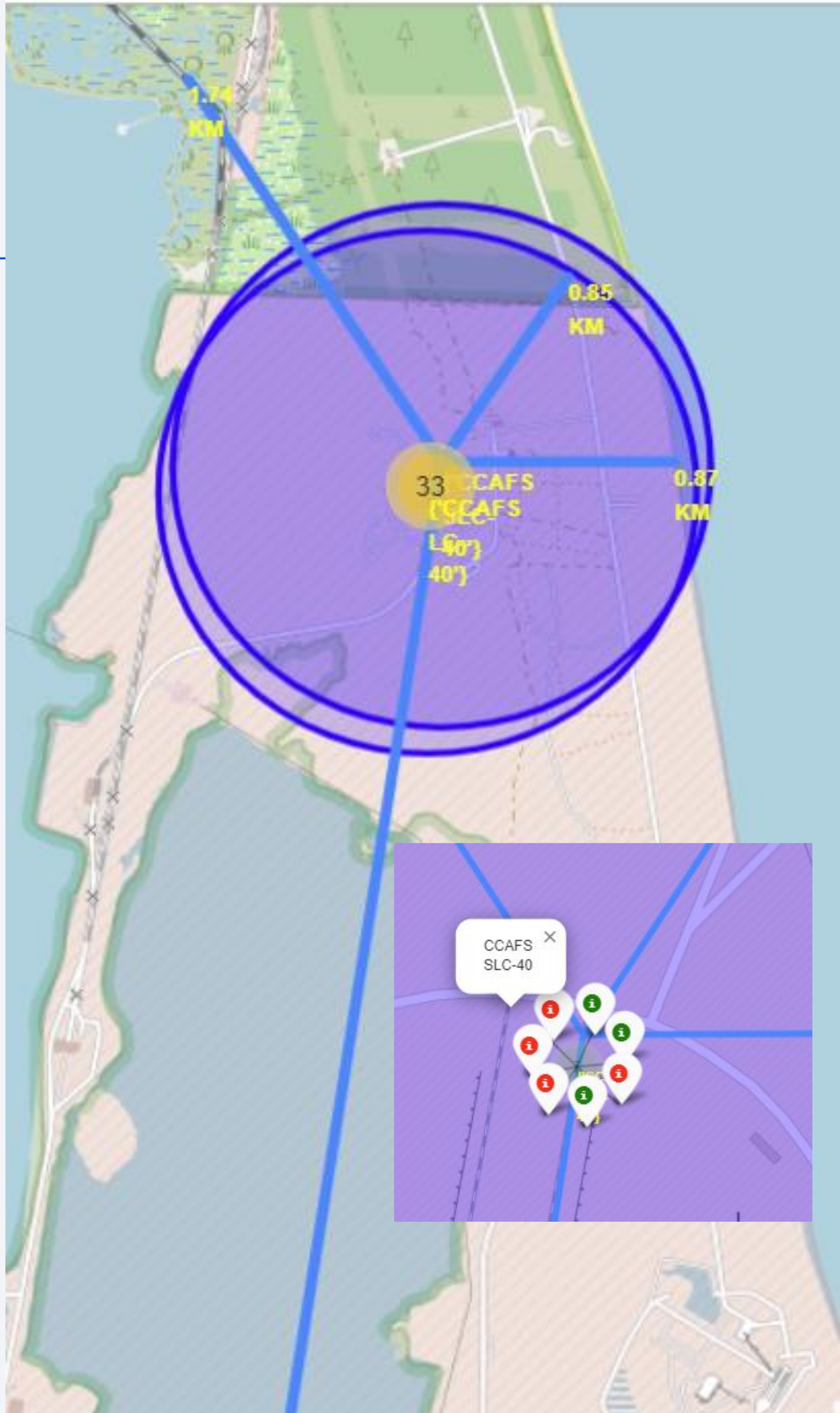
- From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

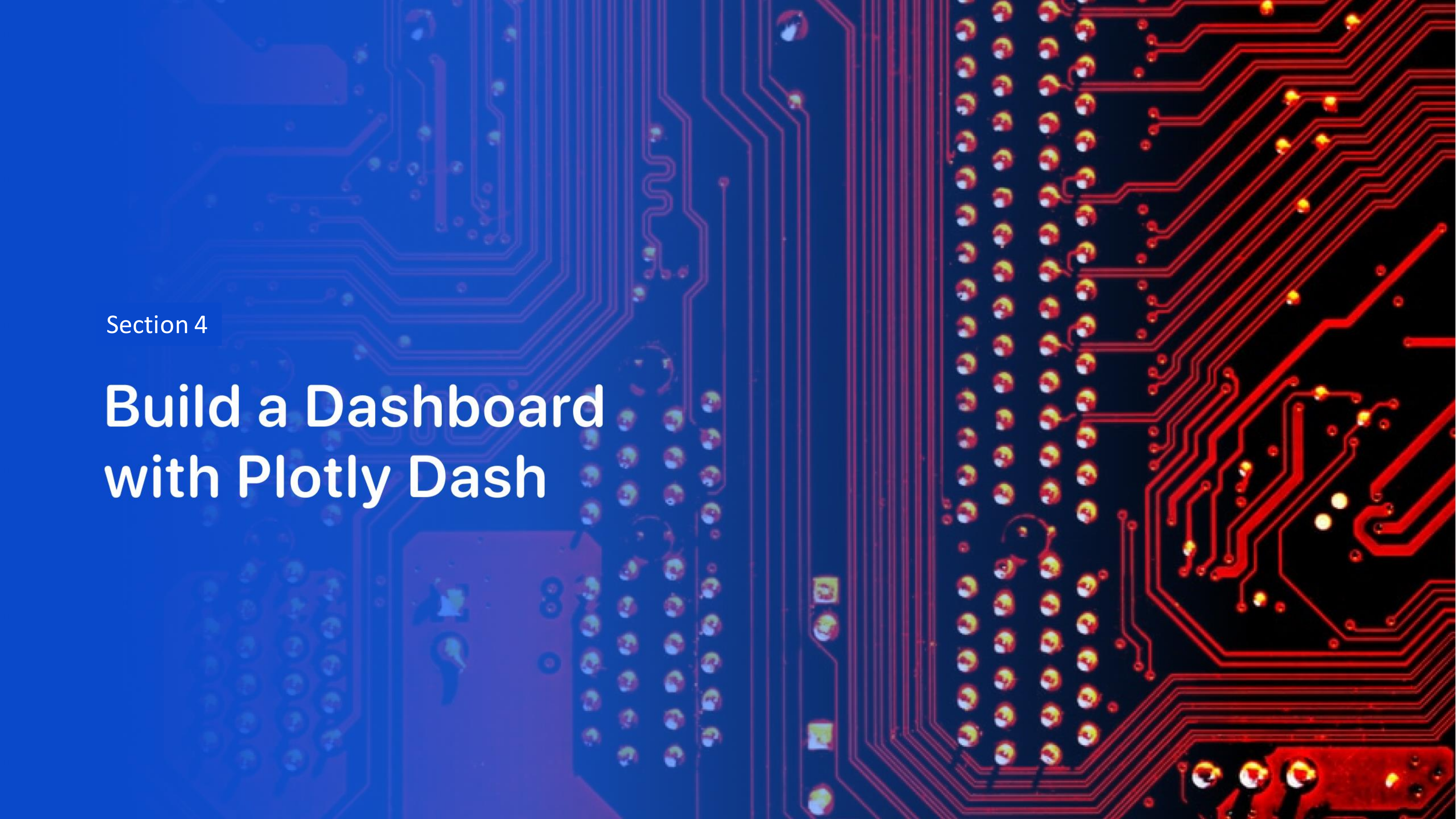


Distance to proximities

- Distance between CCAFS SLC-40 launch site and it proximities :
- Coast: 0.87 km
- Samuel C Phillips Parkway: 0.85 km
- NASA Railroad:1.74 km
- Cape Canaveral: 18.20 km





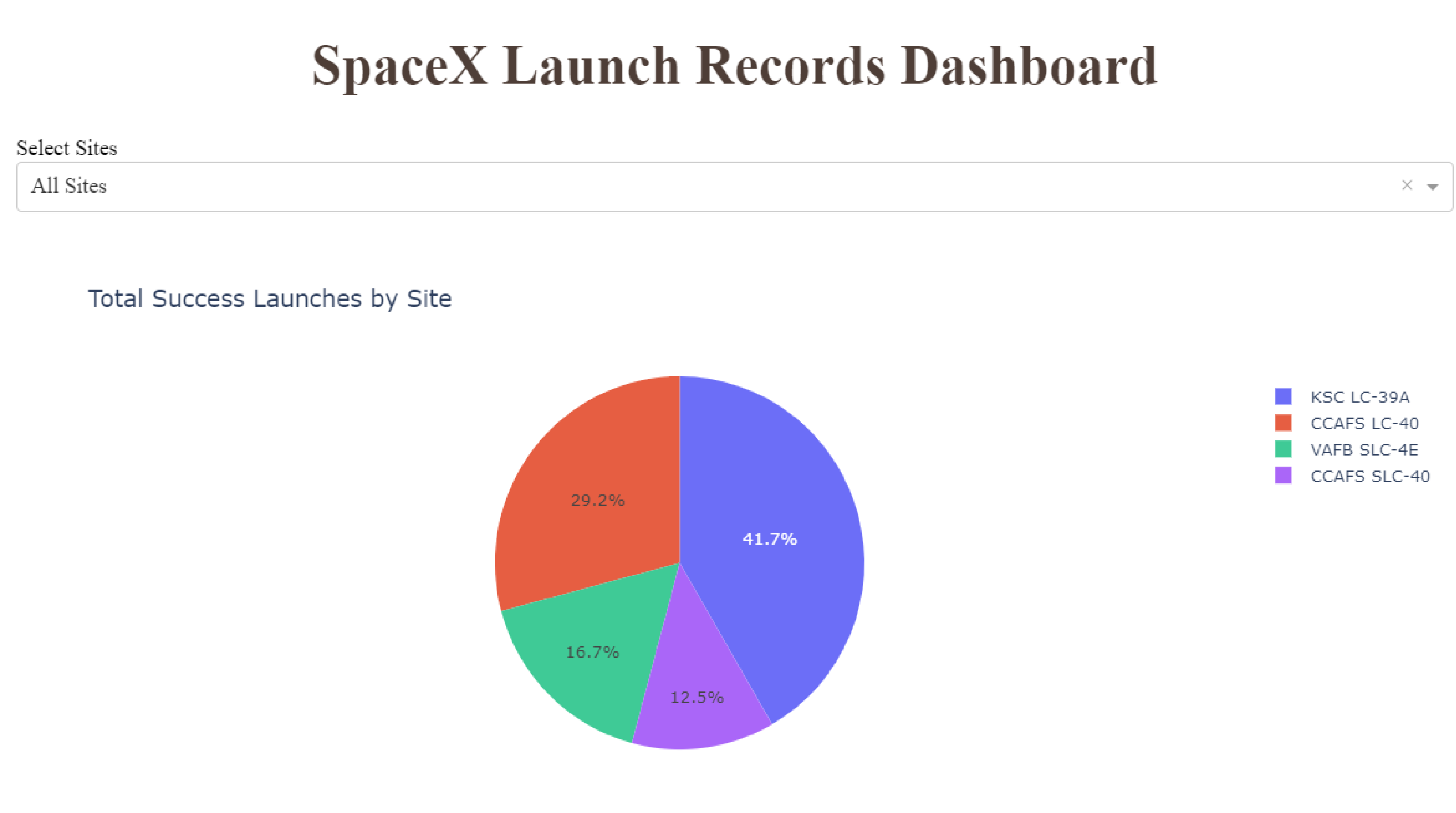


Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

- KSC LC-39A launch site accounts for 41.7% of successful launches.



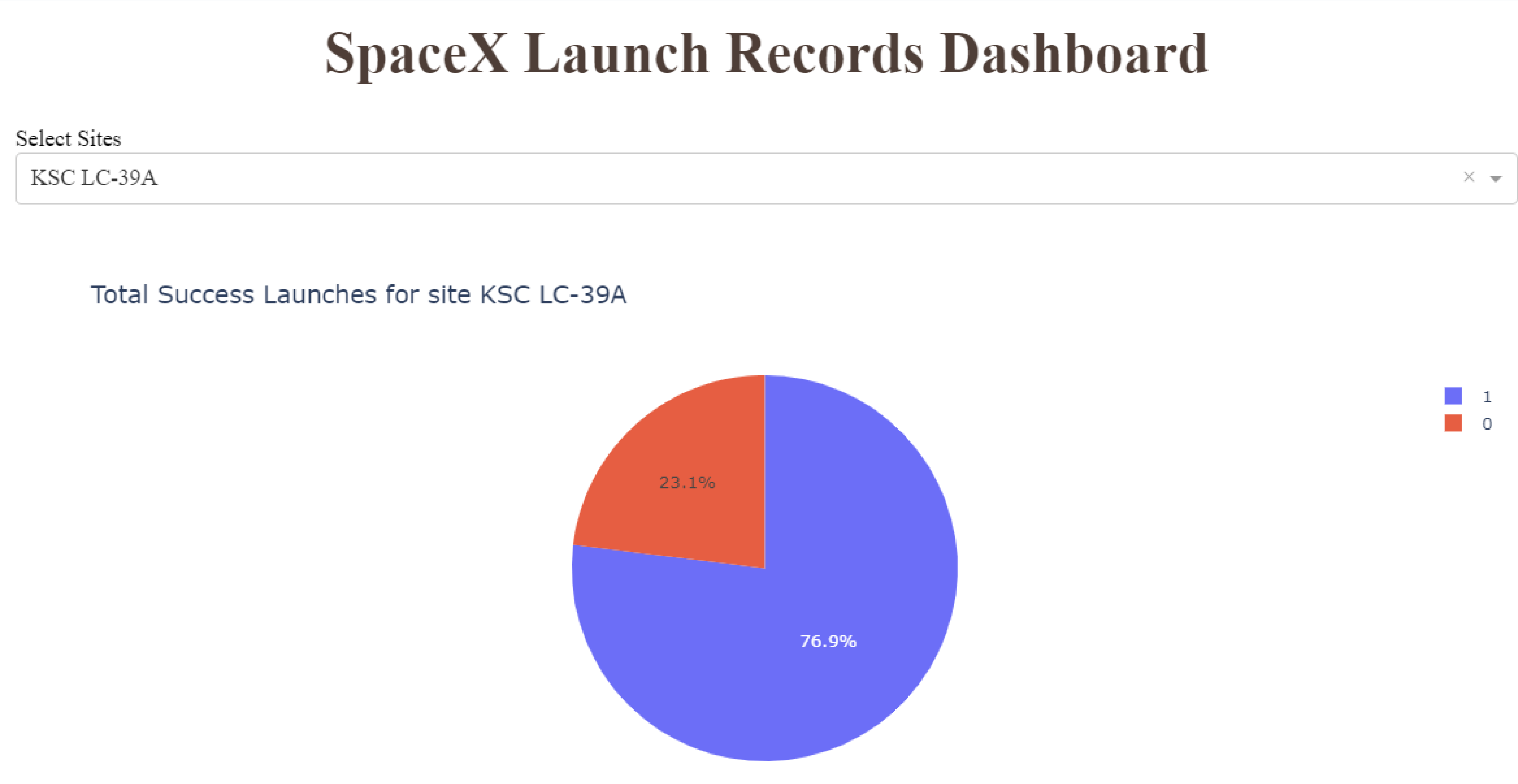
SpaceX Launch Records Dashboard

Select Sites

All Sites

Launch site with highest launch success ratio

- KSC LC-39A has the highest launch success ratio of 76.9% compared to other launch sites.



SpaceX Launch Records Dashboard

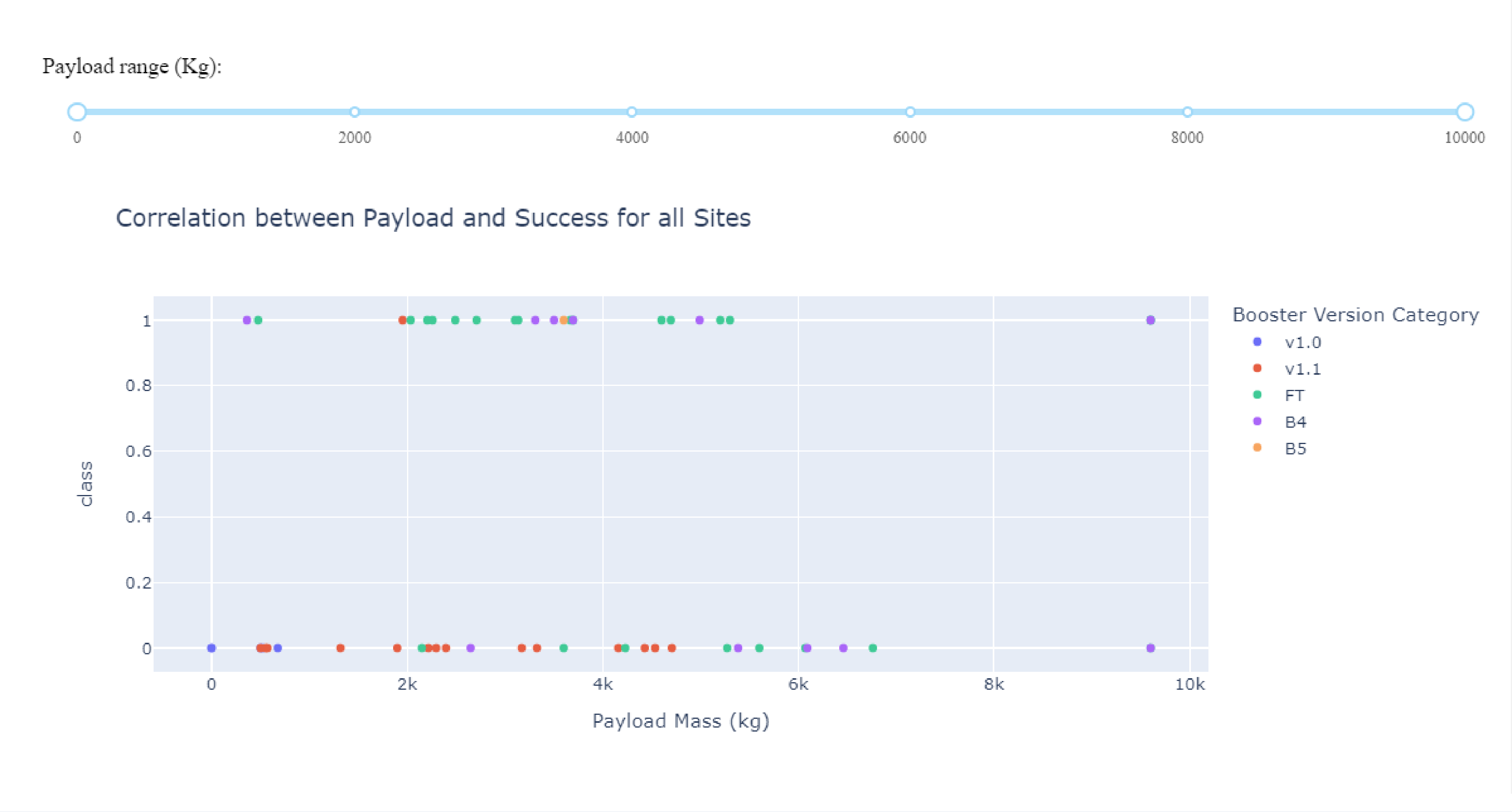
Select Sites

KSC LC-39A

Total Success Launches for site KSC LC-39A

Payload vs. Launch Outcome - all sites

- Rockets with Booster version FT or B4 are likely to have a successful landing.

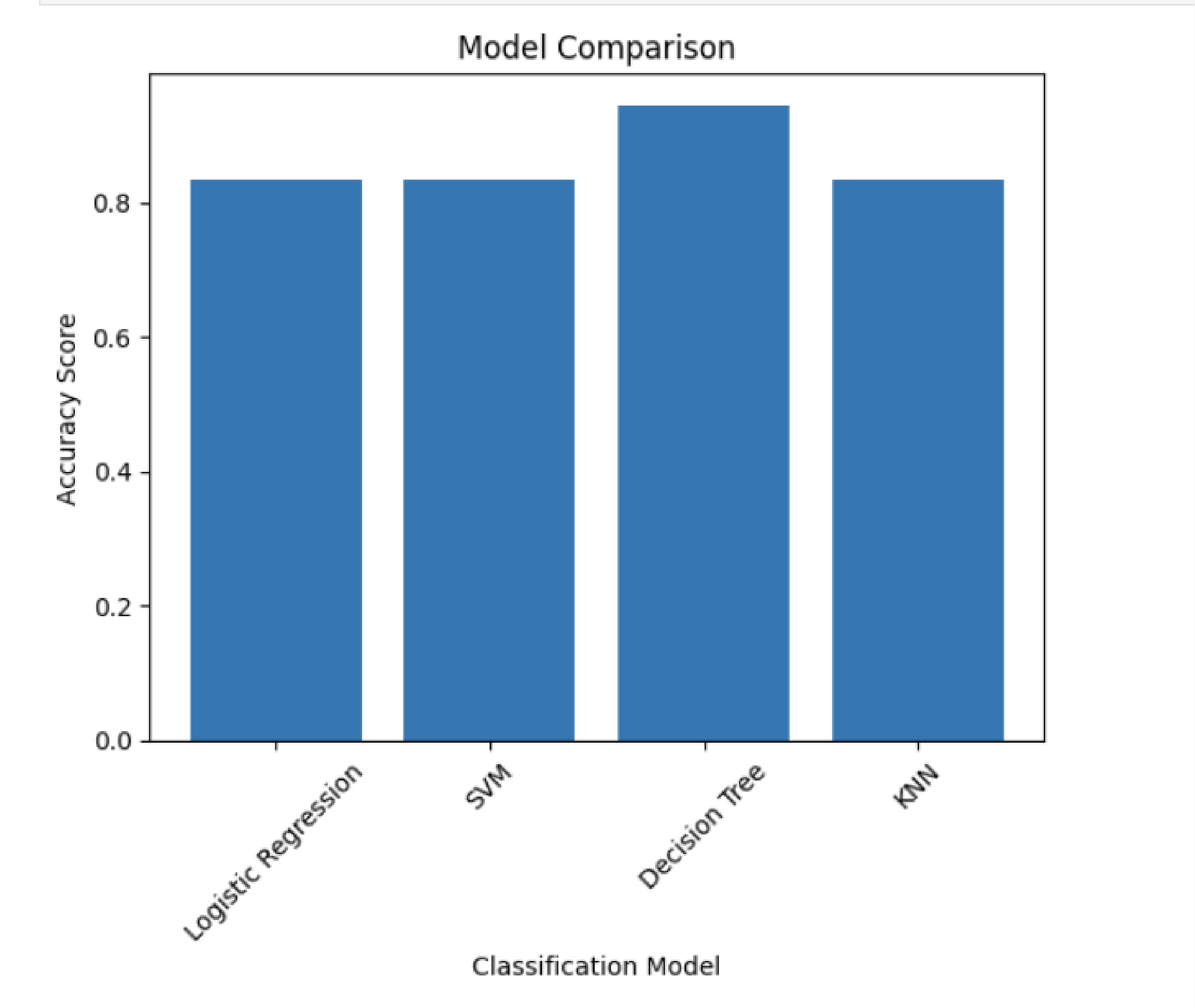


Section 5

Predictive Analysis (Classification)

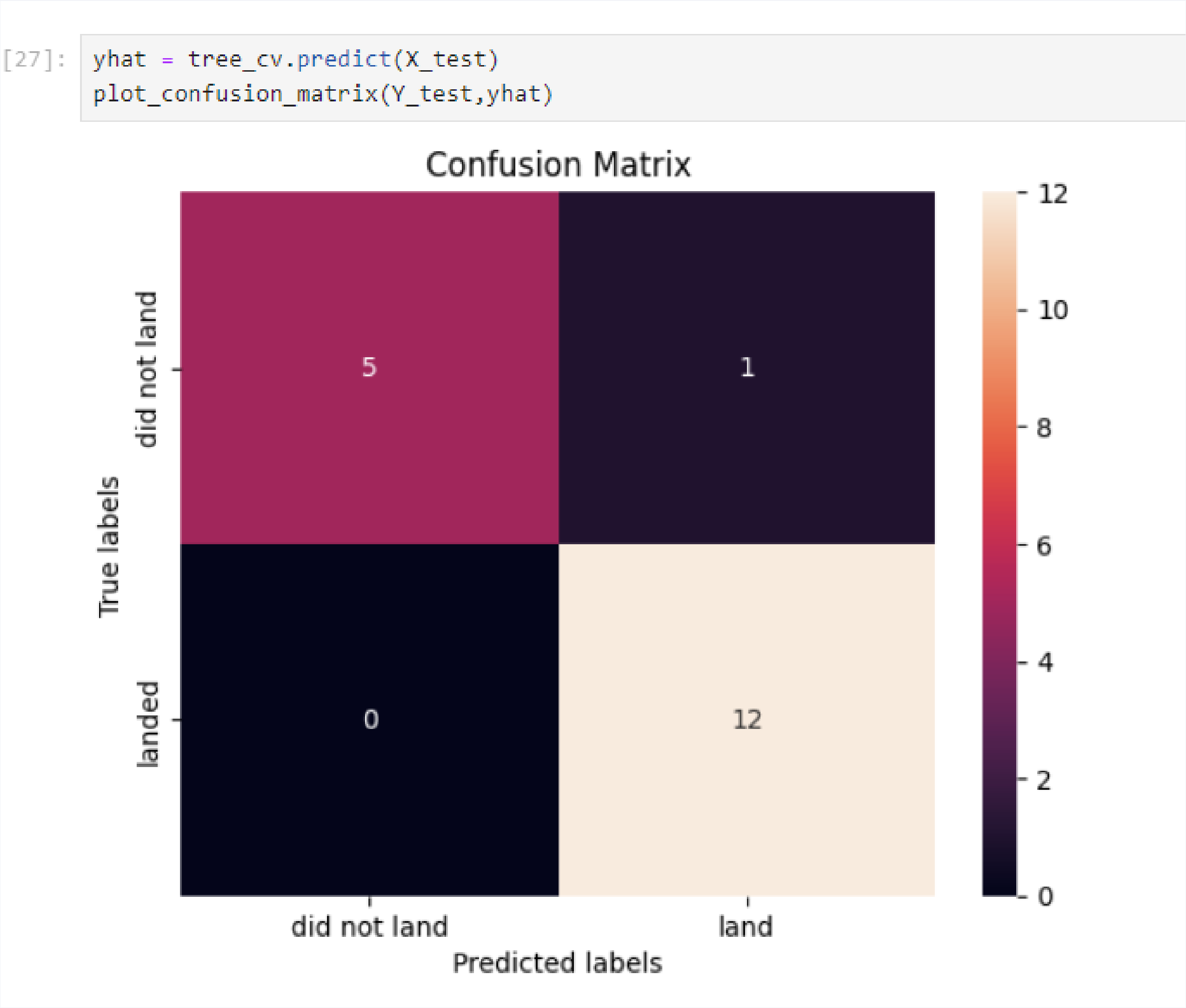
Classification Accuracy

- Decision Tree Classifier yields a higher accuracy on the test data.



Confusion Matrix- Decision Tree

- Out of 18 test sample, Decision Tree Classifier correctly predicted 12 successful landings and 5 failures, with 1 Type II error (False Negative).



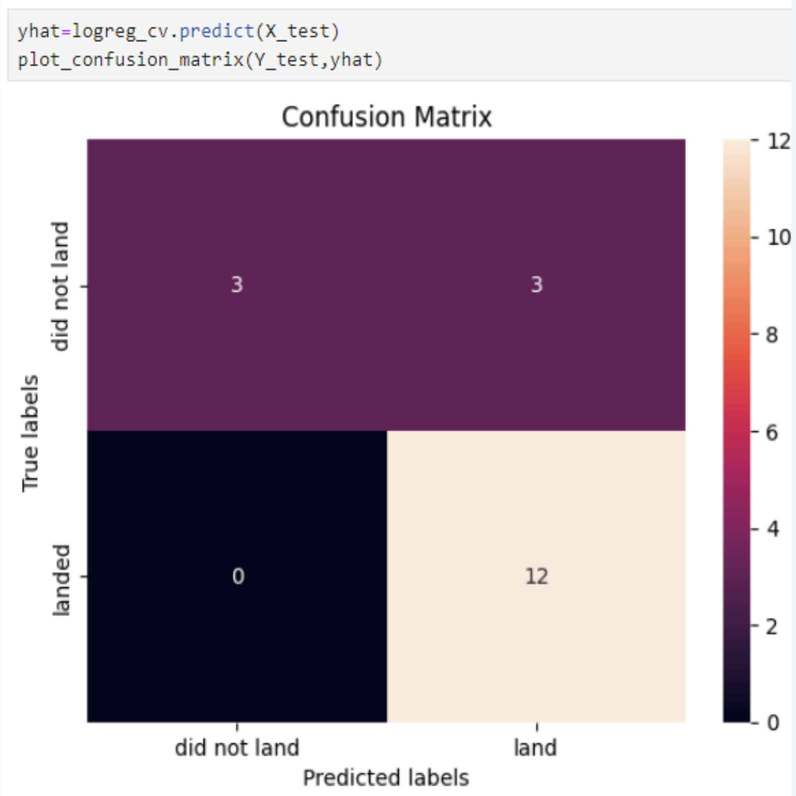
```
[27]: yhat = tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```

Conclusions

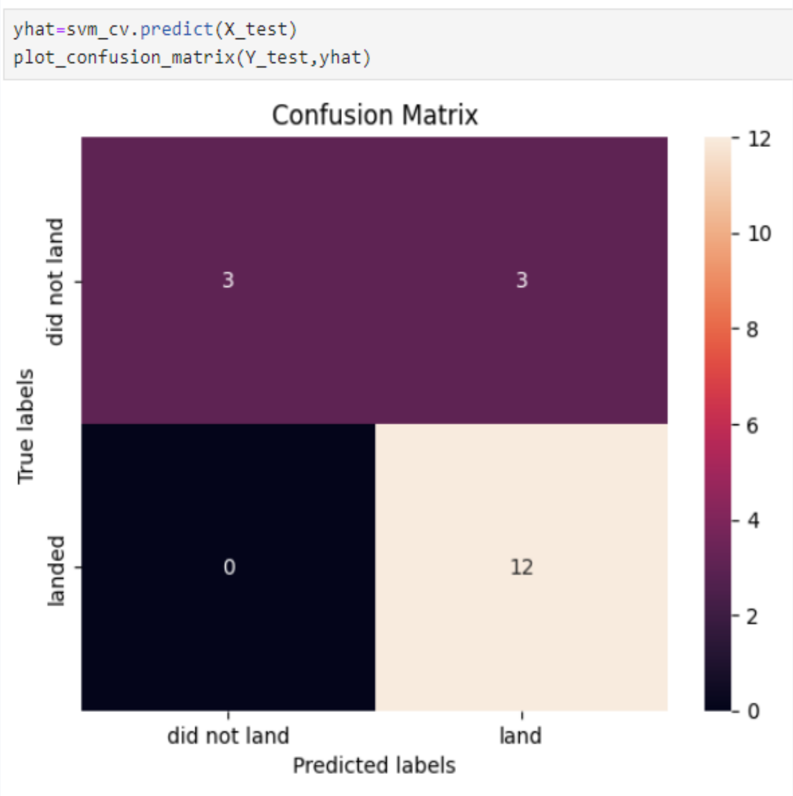
- KSC LC-39A launch site accounts most of successful launches and has highest percentage of success rate compared to other launch sites.
- Rockets with Booster version FT or B4 are likely to have a successful landing.
- Few rocket launches with payloads over 6000kg.
- Decision Tree Classifier performs the best at predicting landing outcome.
- The dataset provided has a small sample size (90 observations); therefore the accuracy of each model may change when train and test on a larger data.

Appendix

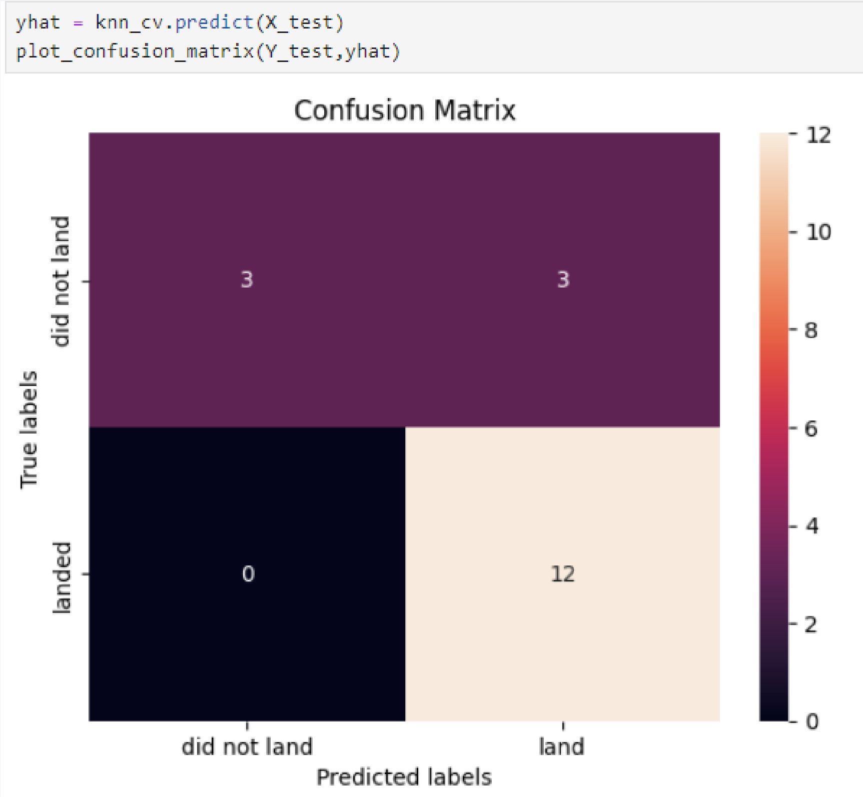
- Confusion matrices for logistic regression, SVM, and KNN show the same result.



```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Thank you!