IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MMH

5th August 2025

Outline

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

Executive Summary

This report presents a comprehensive data science project analyzing SpaceX launch data, incorporating the full data analysis pipeline—from data collection to predictive modeling. The project began with data extraction from the SpaceX API and web scraping of Wikipedia HTML tables, followed by thorough data wrangling to handle missing values and outliers.

Exploratory Data Analysis (EDA) was conducted using both visualizations and SQL queries to uncover key insights and relationships in the data. Interactive visual tools such as Folium maps and Plotly Dash dashboards provided enhanced data storytelling.

To forecast future outcomes, machine learning classification models were implemented, including Logistic Regression, Decision Trees, K-Nearest Neighbors, and Support Vector Machines. These models were trained, validated using GridSearchCV, and evaluated for accuracy, with Logistic Regression achieving the highest performance at approximately 94%.

Overall, this end-to-end project demonstrates proficiency in real-world data handling, insight generation, and predictive modeling, providing actionable intelligence on SpaceX's launch outcomes.

Introduction

This report outlines the full data analysis process, starting with data collection and cleaning to improve quality. It then explores the data using SQL queries, statistical analysis, and visualizations to uncover patterns and relationships, and interactive dashboards. The data is further analyzed by grouping it based on categorical variables. Finally, predictive models are built and refined to generate deeper insights and forecast future trends.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Data was collected from SpaceX API and HTML data from the web.

- Perform data wrangling
  - During data wrangling, missing data was identified and fixed. Outliers were detected and removed .
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The model was built and fine tuned with GidsearchCV cross validation technique.

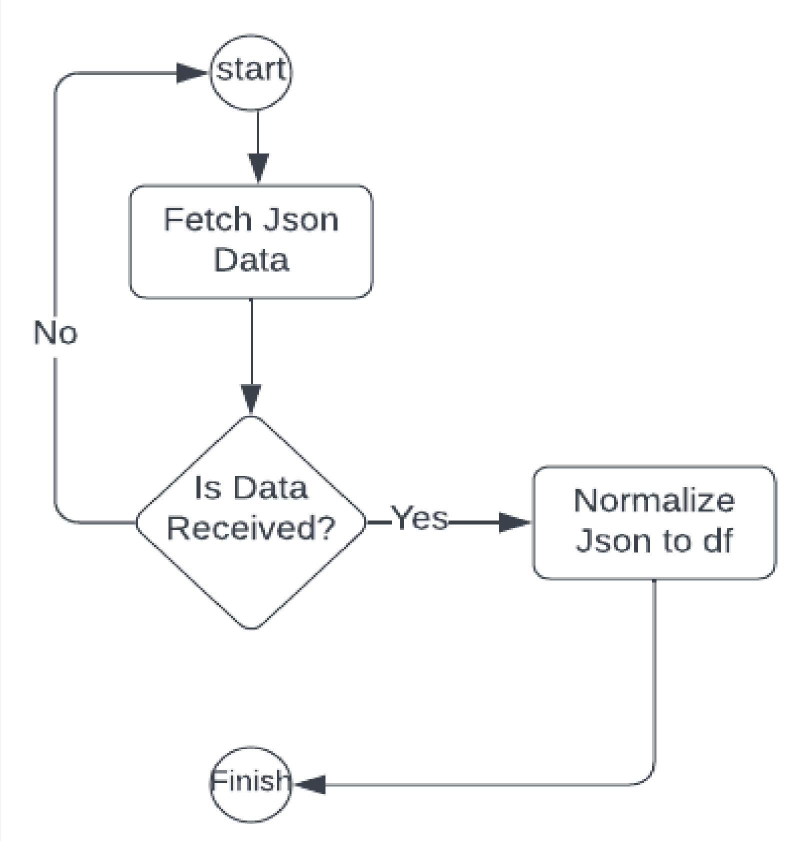
Data Collection

- Data was collected from SpaceX API and HTML from the web. This was preprocessed in order to meet the required accuracy during data analysis phase.

Data Collection – SpaceX API

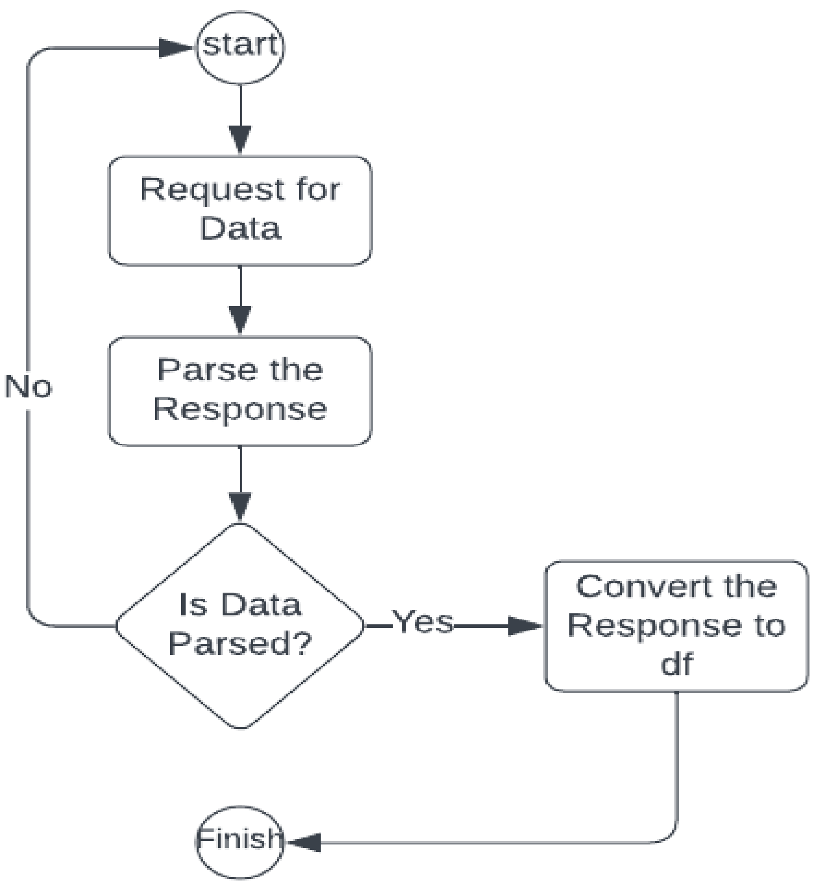
- Data was collected from SpaceX REST API using a get request from the requests object. The resulting json data is normalized and converted into a pandas data frame for further manipulation. This process is illustrated by the flow chart a side.

- The GitHub URL below is for the completed SpaceX API calls notebook; https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Falcon 9 launch records from Wikipedia HTML table were extracted using Beautiful Soup
- The received html data from the web was parsed and converted into a Pandas data frame.
- The GitHub URL points to the completed web scraping notebook; https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- During data wrangling process the following operations were carried out;

1) Calculation of the number of launches on each site
2) Calculation of the number and occurrence of each orbit
3) Calculation of the number and occurrence of mission outcome of the orbits
4) Creation of a landing outcome label from Outcome column
5) The following GitHub URL points to the completed data wrangling related notebooks; https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- The following charts were plotted during the EDA stage;
- Catplot – This was used to display an outlay of *Flight Number* vs *Payload Mass*
- Catplot showing the relationship between *Flight Number* and *Launch Site* – This was used to visualize the relationship between the Flight Number and the Launch Sites.
- Bar graph showing the relationship between success rate of each orbit type.
- Scatter plot for the relationship between Payload Mass and Launch Site. The scatter plot shows whether there is a correlation between payload mass and launch site
- Lineplot showing Visualize launch success yearly trend

- GitHub URL of the completed EDA with data visualization noteboo; https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/edadataviz.ipynb

EDA with SQL

TASK	QUERRY	PURPOSE
1	SELECT DISTINCT Launch_Site FROM SPACEXTABLE	Fetch uique launch sites
2	SELECT * FROM SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5	Display 5 records where launch sites begin with the string 'CCA'
3	select sum(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where customer ='NASA (CRS)'	Display the total payload mass carried by boosters launched by NASA (CRS)
4	SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Booster_Version ='F9 v1.1'	Display average payload mass carried by booster version F9 v1.1
5	SELECT Date FROM SPACEXTABLE where Landing_Outcome = 'Success (ground pad)' order by Date ASC LIMIT 1	List the date when the first succesful landing outcome in ground pad was acheived.

EDA SQL Cont’n

TASK	QUERRY	PURPOSE
6	select * from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000)	List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

7	SELECT COUNT(*) FROM SPACEXTABLE WHERE Mission_Outcome ='Success’
	SELECT COUNT(*) FROM SPACEXTABLE WHERE Mission_Outcome ='Failure'
		List the total number of successful and failure mission outcomes

8	SELECT booster_version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);	List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

EDA SQL CONT’N

TASK	QUERRY	PURPOSE
9	SELECT CASE substr(Date, 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April' WHEN '05' THEN 'May' WHEN '06' THEN 'June' WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September' WHEN '10' THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December' END AS month_name, landing_outcome, booster_version, launch_siteFROM SPACEXTABLEWHERE landing_outcome LIKE '%Failure (drone ship)%' AND substr(Date, 0, 5) = '2015';	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.
10	WITH outcome_counts AS (SELECT landing_outcome, COUNT(*) AS outcome_count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing_outcome),ranked AS (SELECT landing_outcome, outcome_count, RANK() OVER (ORDER BY outcome_count DESC) AS Rank FROM outcome_counts)SELECT *FROM rankedORDER BY rank, landing_outcome;	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.¶

EDA WITH SQL CONT’N

- The Github link bellow points to the completed EDA with SQL Notebook https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

No	Object	Use
1	Markers	Markers were used pinpoint exact coordinates with labels
2	Circles	Was used to add a highlighted circle area with a text label on a specific coordinates
3	Lines	Used to draw travel paths like between launch sites and railways
4	Marker Cluster	Used to add clusters of markers on the map

The GitHub URL of your completed interactive map with Folium map;
https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

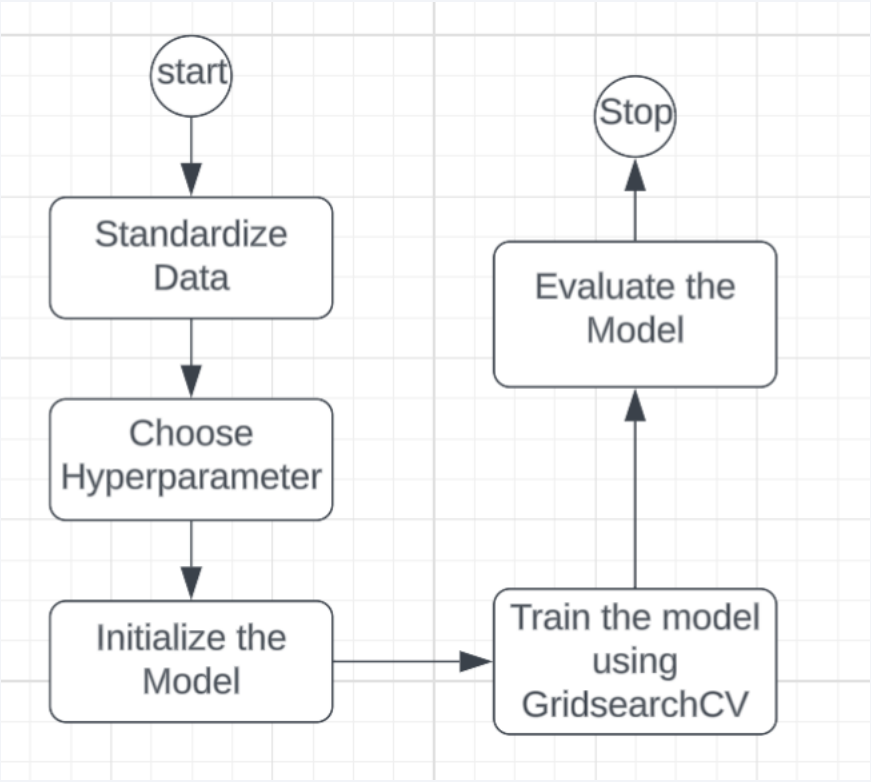
Build a Dashboard with Plotly Dash

NO	PLOT/CHART	USE
1	Line Graph	It was used to display Automobile Sales trend over Recession Period and Yearly Automobile sales for the whole period because it is the best to display trends patterns
2	Bar Graph	Displayed Average Automobile Sales fluctuation over Recession Period and effect of unemployment rate on sales
3	Pie Chart	Expenditure shared by vehicle type, average vehicles sold in a year and budget propositions. The pie chart is the best visualization that can be used for this type of data

- The GitHub URL below points to the completed Plotly Dash lab; https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/DV0101EN-Final-Assign-Part-2-Questions.py

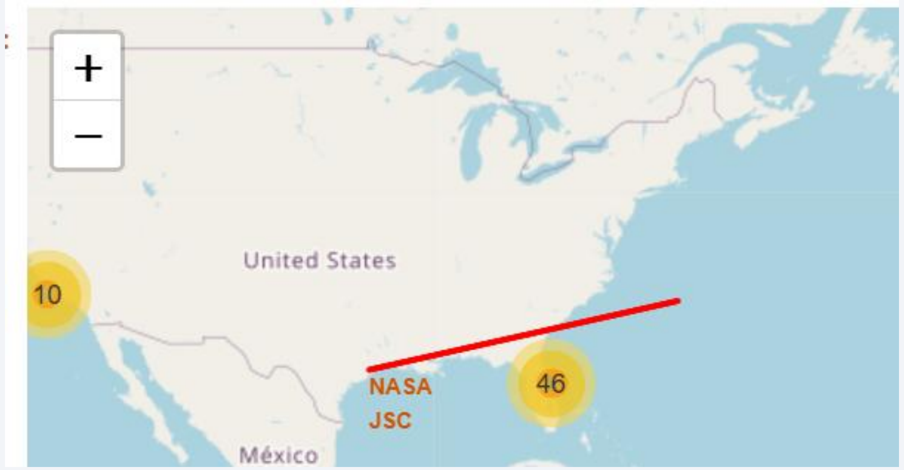
Predictive Analysis (Classification)

- I used standard scalar to standardize the data before model building to ensure that all data is on a similar scale.
- I also split the data through training and testing dataset using train_test_split
- I initialized the objects for all the algorithms for the models used after whilch I Used Gidsearchcv cross validation technique to train the models using the various hyperparameters.
- I evaluated the model by calculating the accuracy of the best estimators from grid search to determine which model performs best on unseen data.
- GitHub URL for completed predictive analysis; https://github.com/TechHavenUG/DS-CAPSTONE/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Results

- Exploratory data analysis revealed some good insights about the data as well as producing visually appealing plots/charts that helped to draw insights about the dataset.
- As shown by the picture below, during Interactive analytics several maps with markers and lines were produced during the lab.



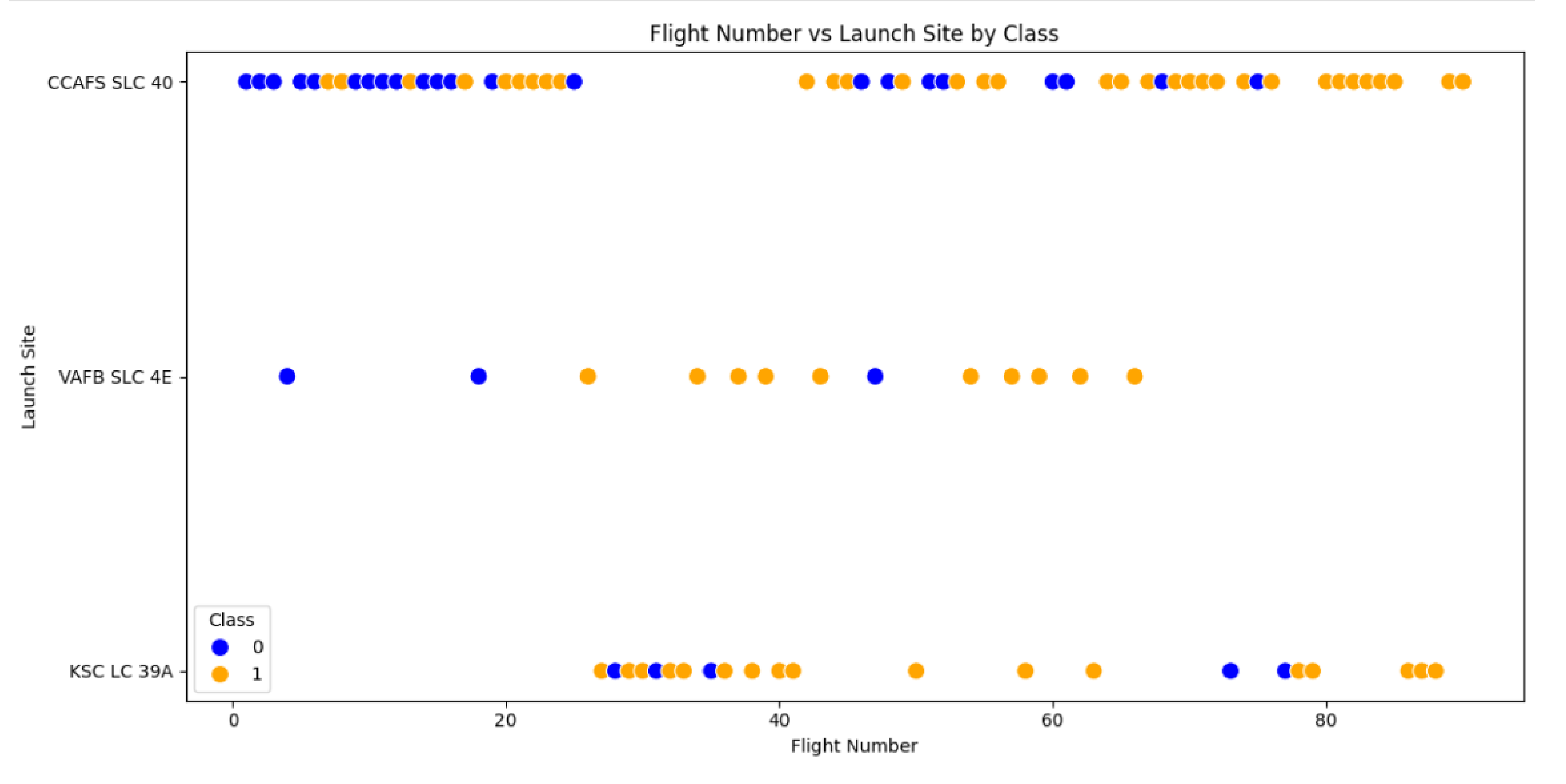
Results Cont'n

- During Logistic Regression model, Tree Regression, and KNN all have the same accuracy of approximately 94% and the SVM had accuracy of approximately 89%.

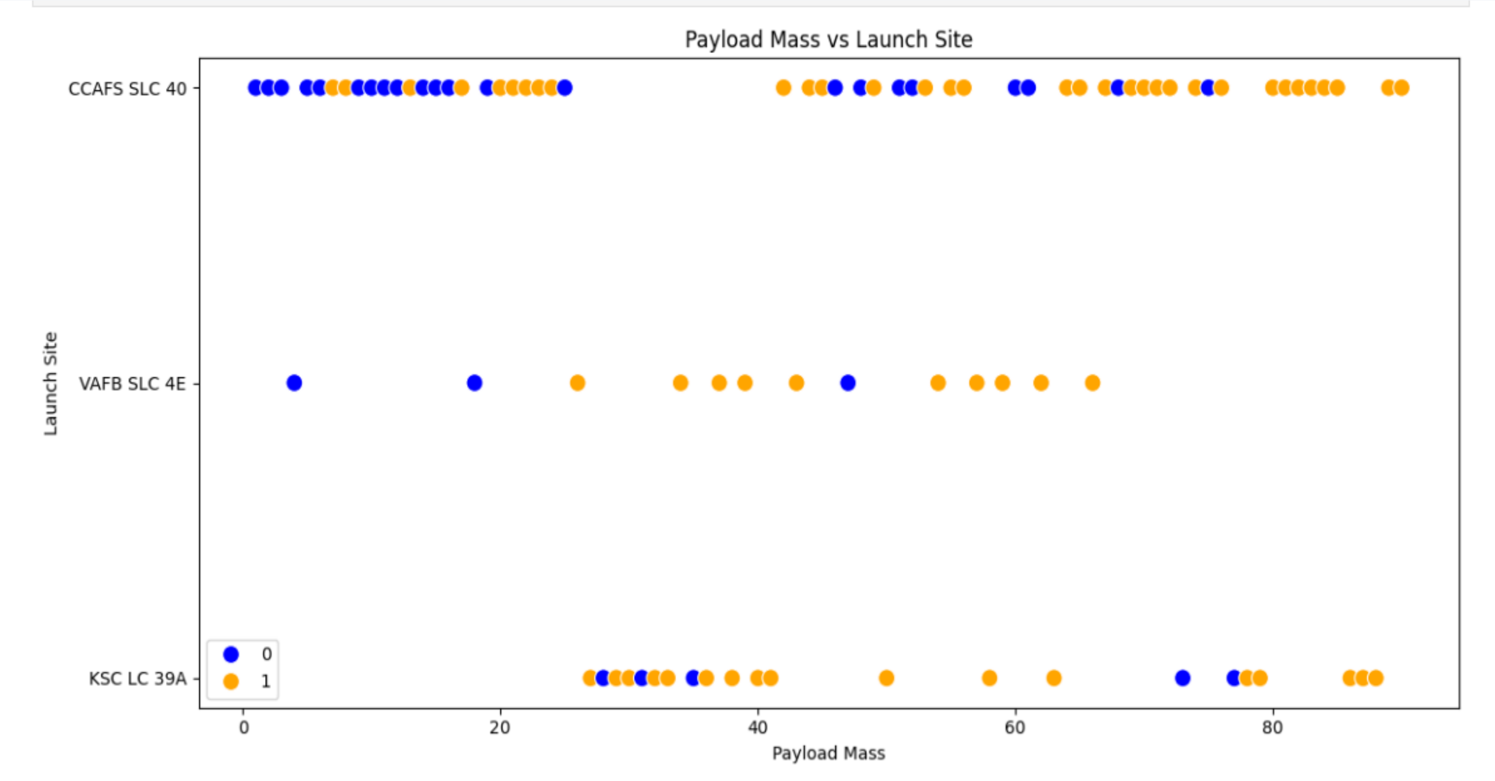
Section 2

Insights drawn from EDA

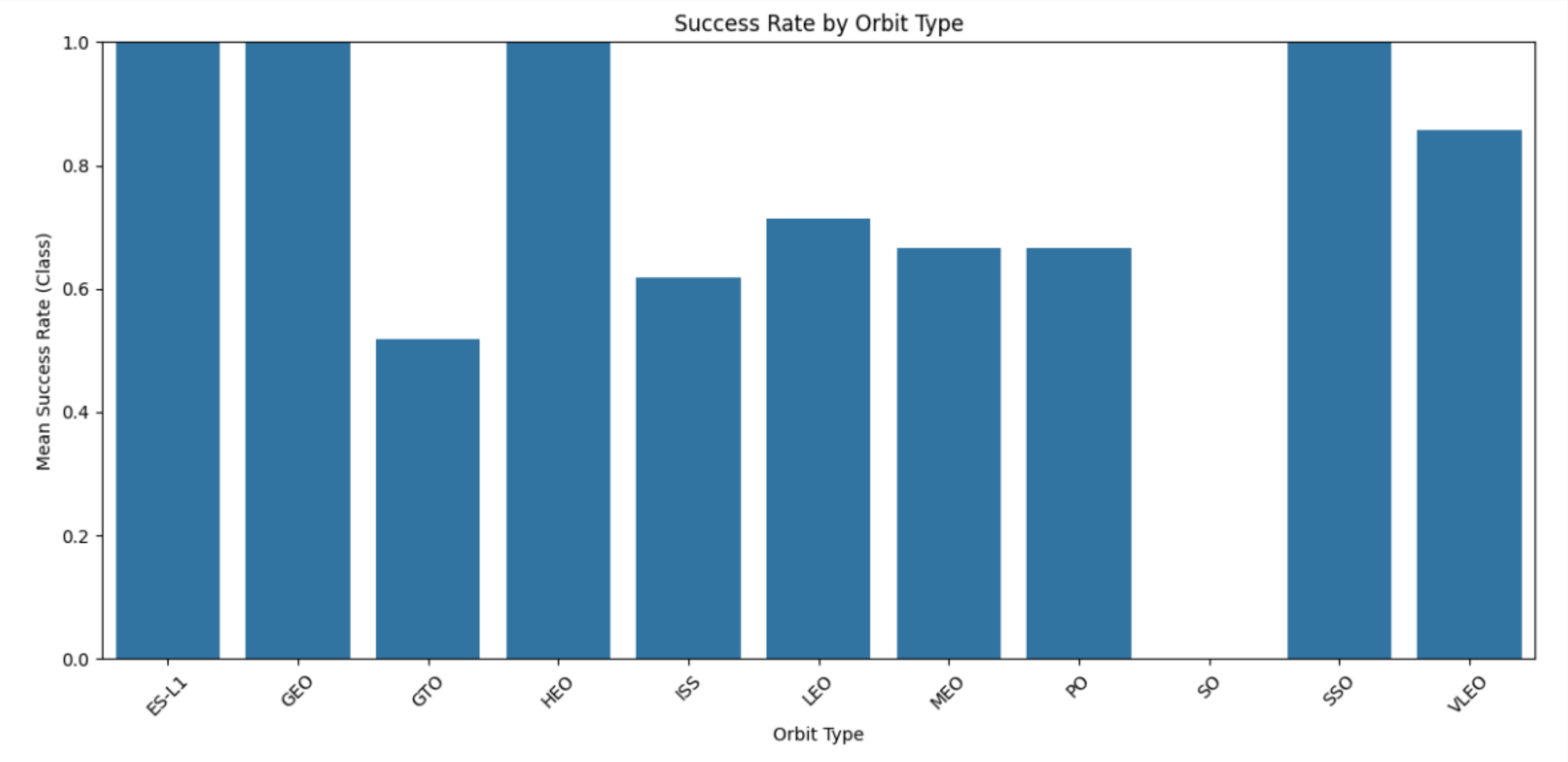
Flight Number vs. Launch Site



Payload vs. Launch Site

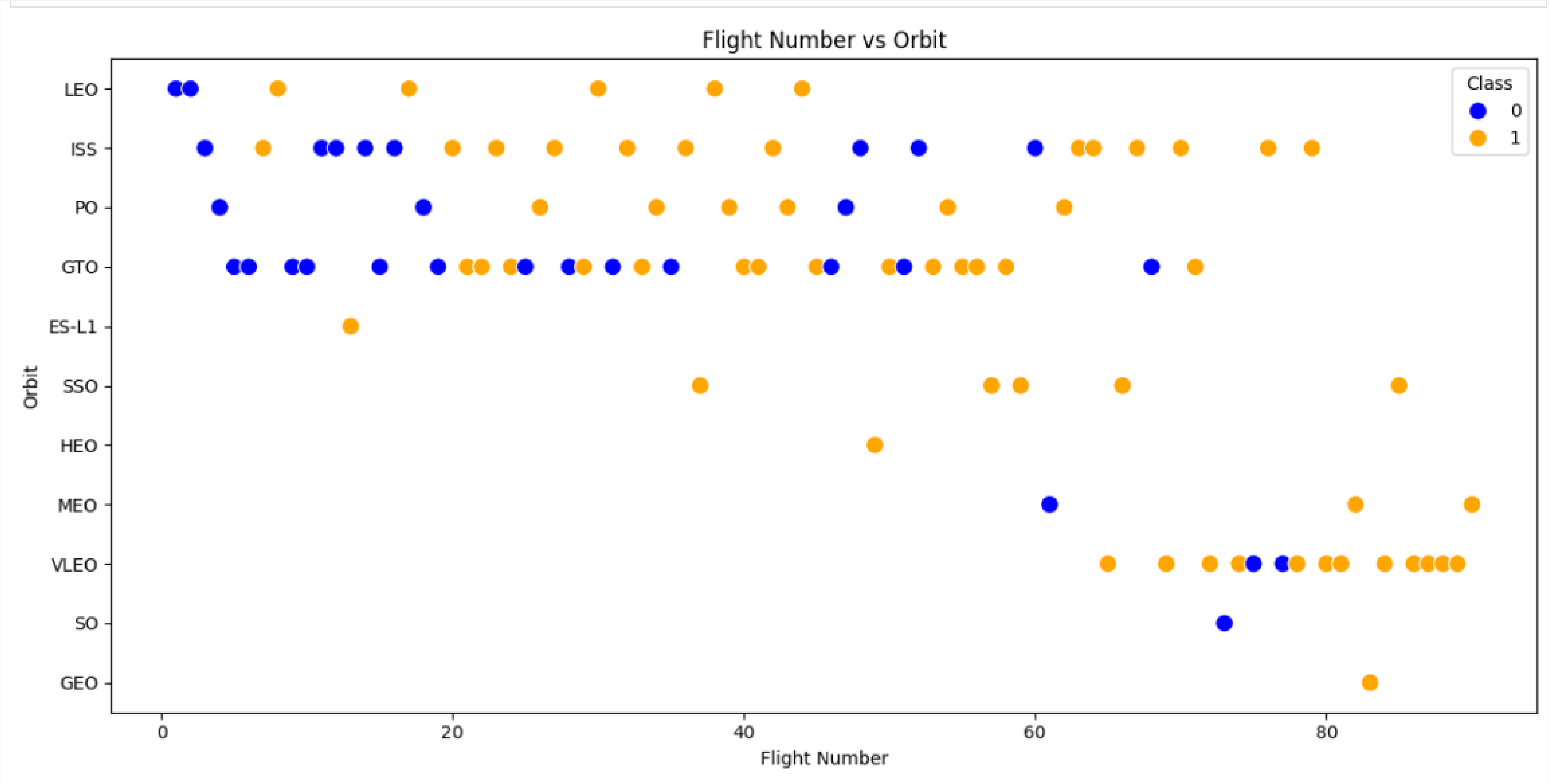


Success Rate vs. Orbit Type

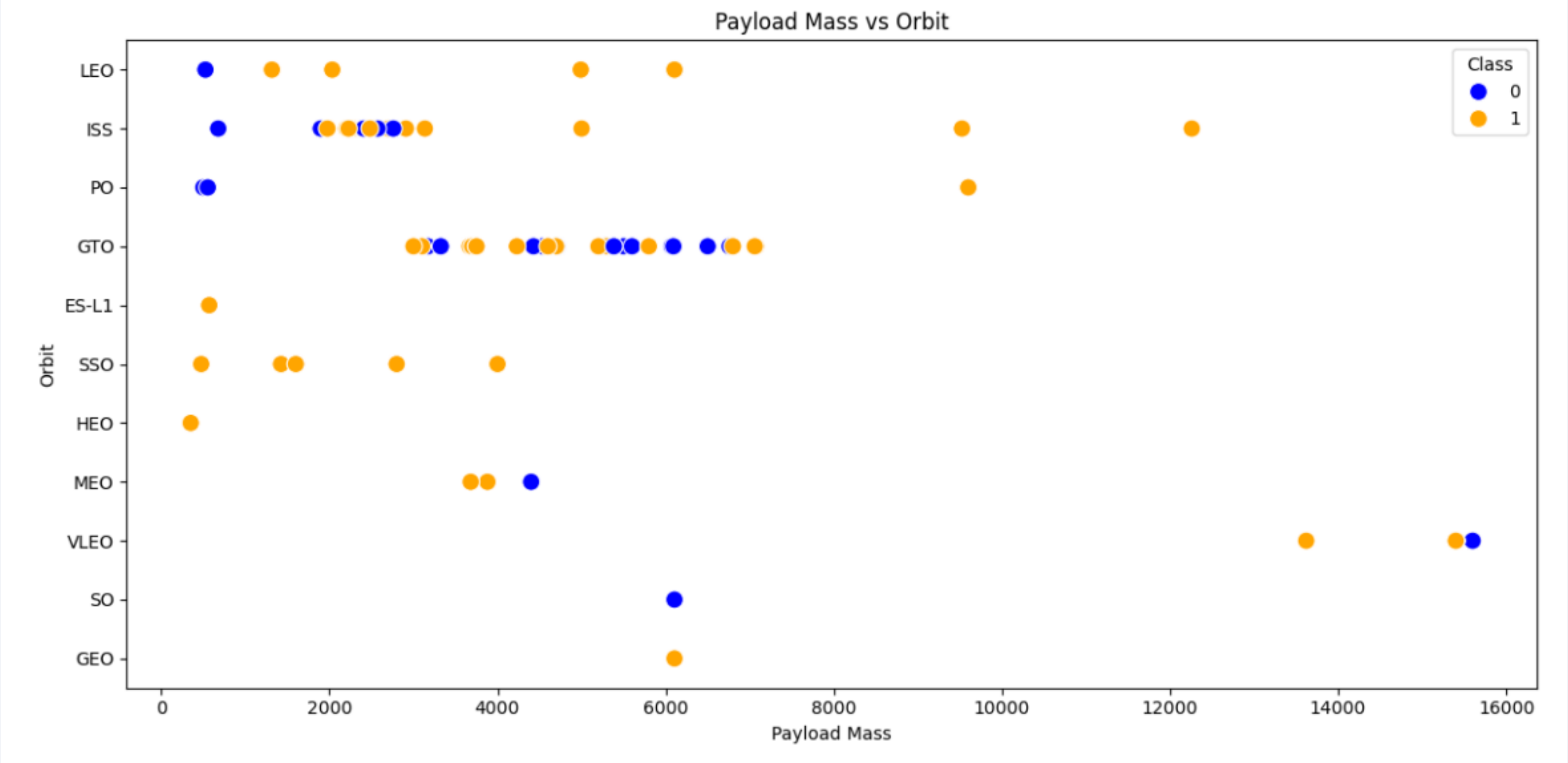


- ES-L1,GEO,HEO AND S50 have highest success rates per Orbit

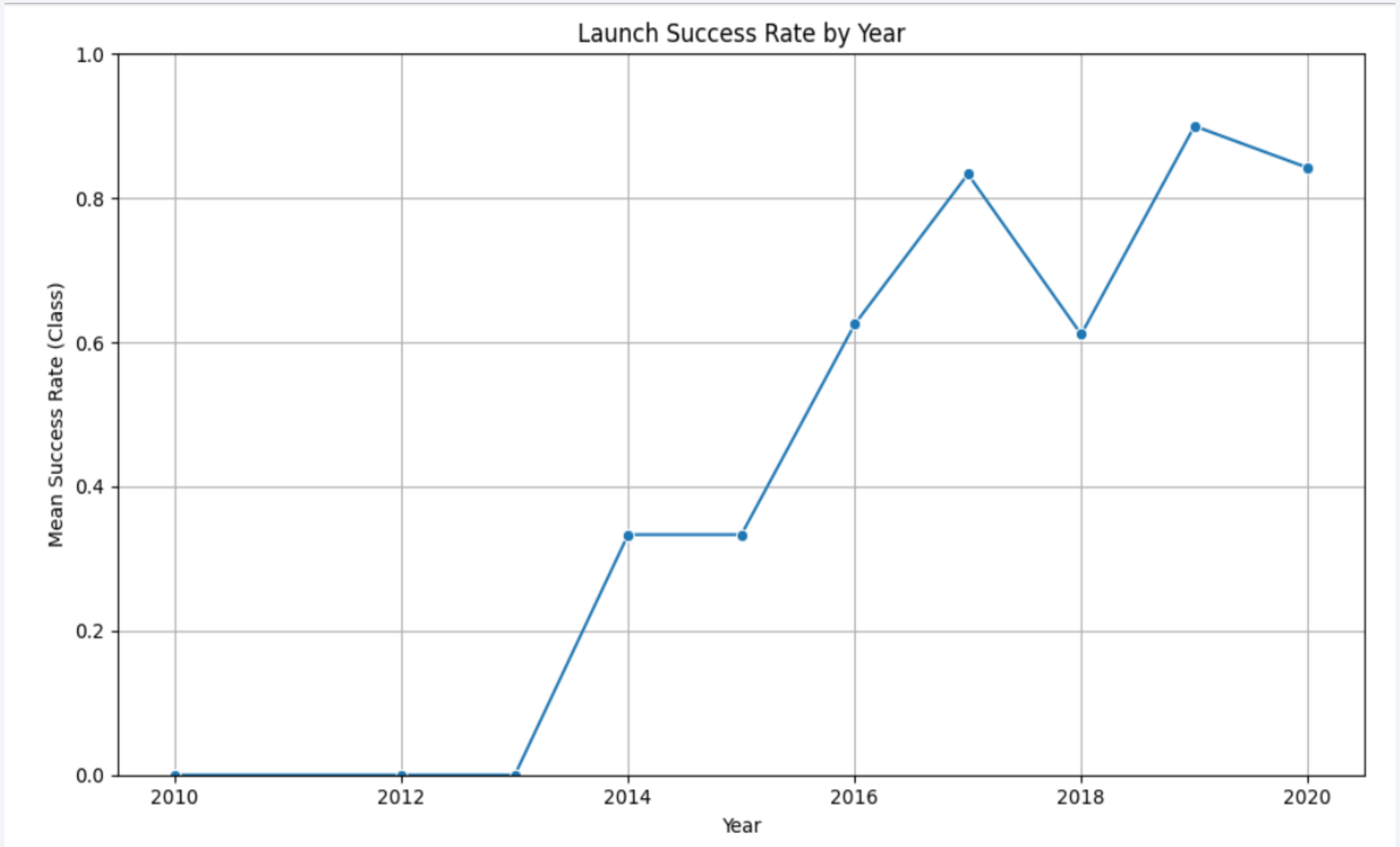
Flight Number vs. Orbit Type



Payload vs. Orbit Type

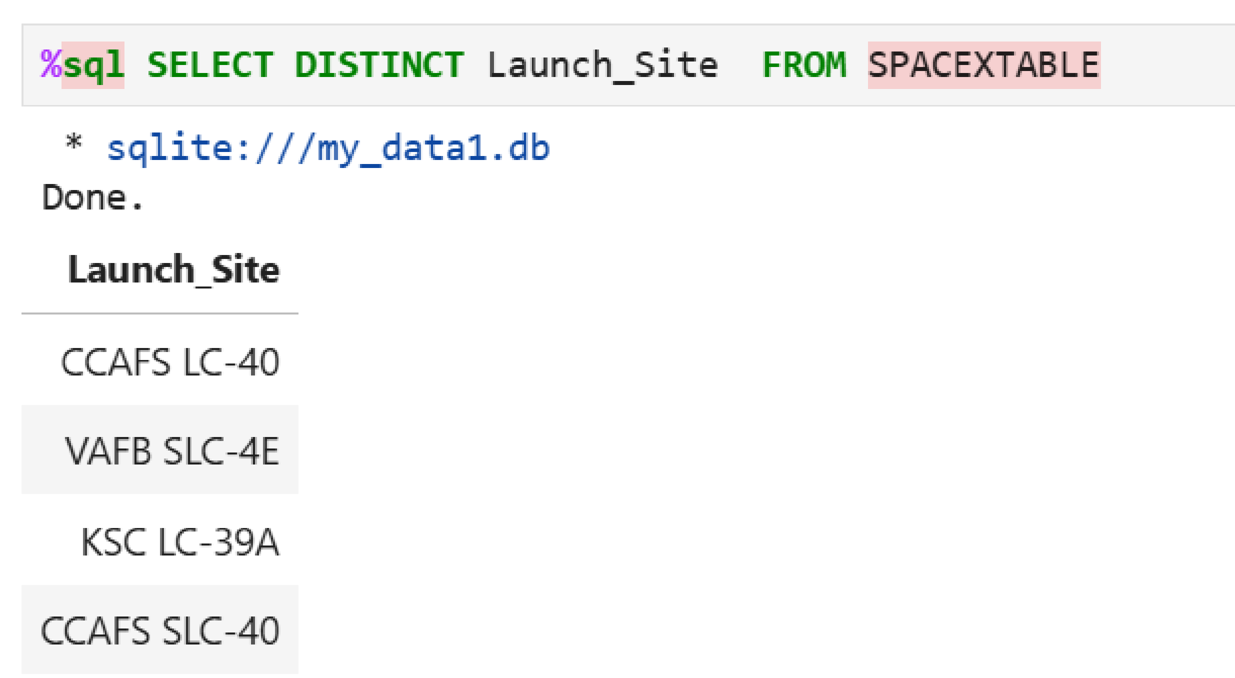


Launch Success Yearly Trend



- Generally success rates has increased over the years

All Launch Site Names



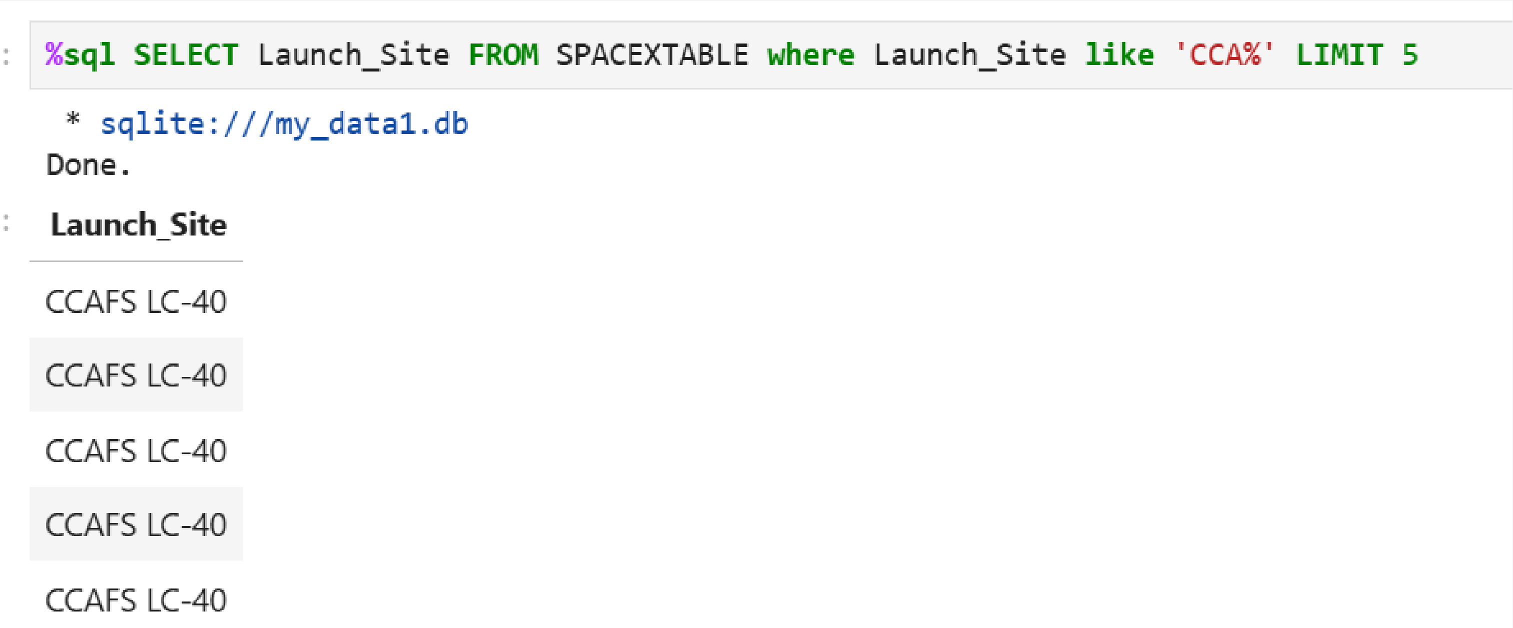
```
%sql SELECT DISTINCT Launch_Site  FROM SPACEXTABLE
```

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

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

- The distinct keyword is used to select unique launching sites

Launch Site Names Begin with 'CCA'



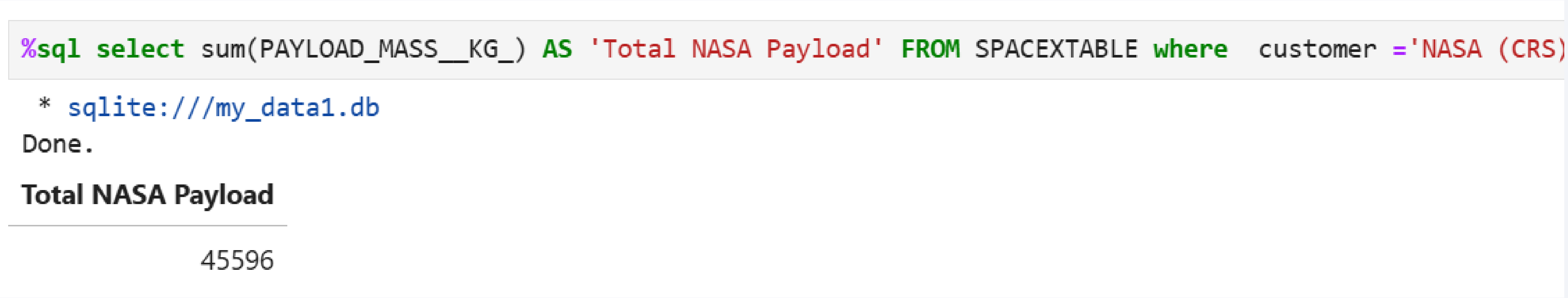
```
%sql SELECT Launch_Site FROM SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5
```

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

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

- The like key word is used to filter out the launch sites while the limit keyword is used to limit the number of rows to display.

Total Payload Mass



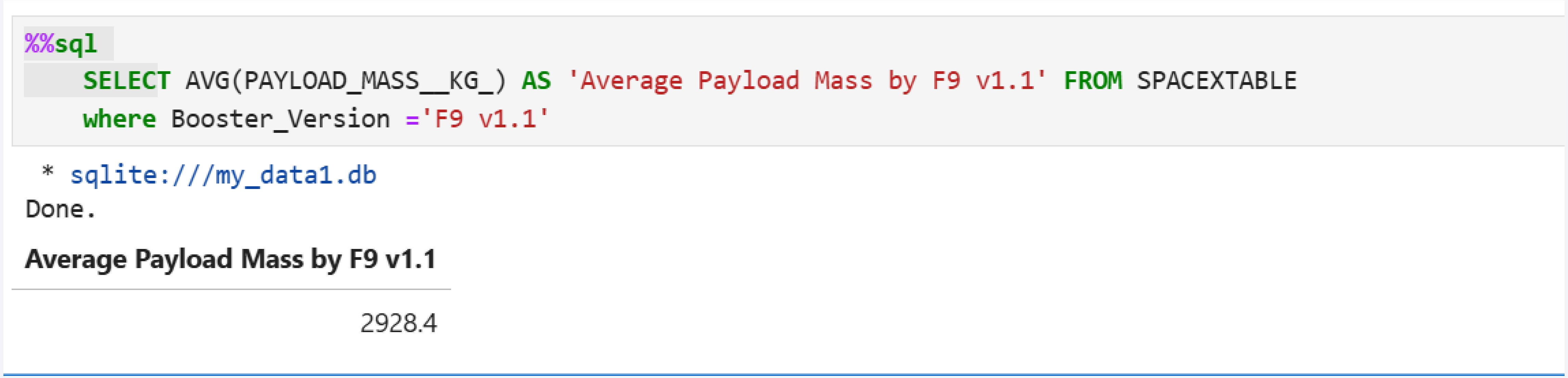
```
%sql select sum(PAYLOAD_MASS__KG_) AS 'Total NASA Payload' FROM SPACEXTABLE where  customer ='NASA (CRS)
```

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

Total NASA Payload
45596

- Sum function is used to calculate the sum of NASA payloads

Average Payload Mass by F9 v1.1



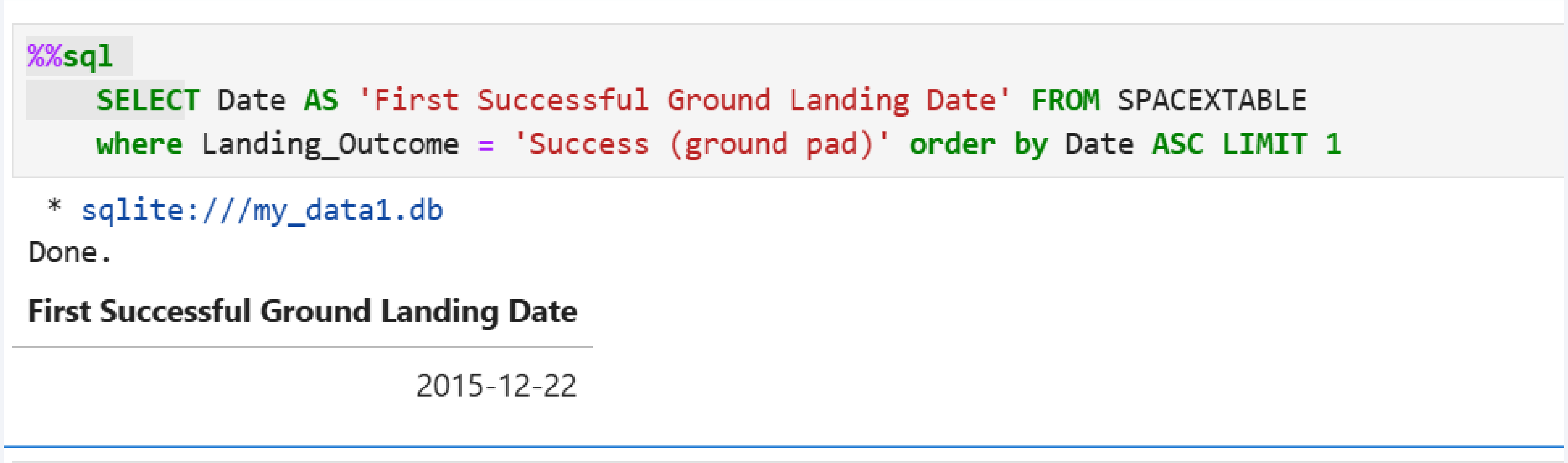
```
%%sql
    SELECT AVG(PAYLOAD_MASS__KG_) AS 'Average Payload Mass by F9 v1.1' FROM SPACEXTABLE
    where Booster_Version ='F9 v1.1'
```

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

Average Payload Mass by F9 v1.1
2928.4

- AVG payload is used to calculate average payload

First Successful Ground Landing Date



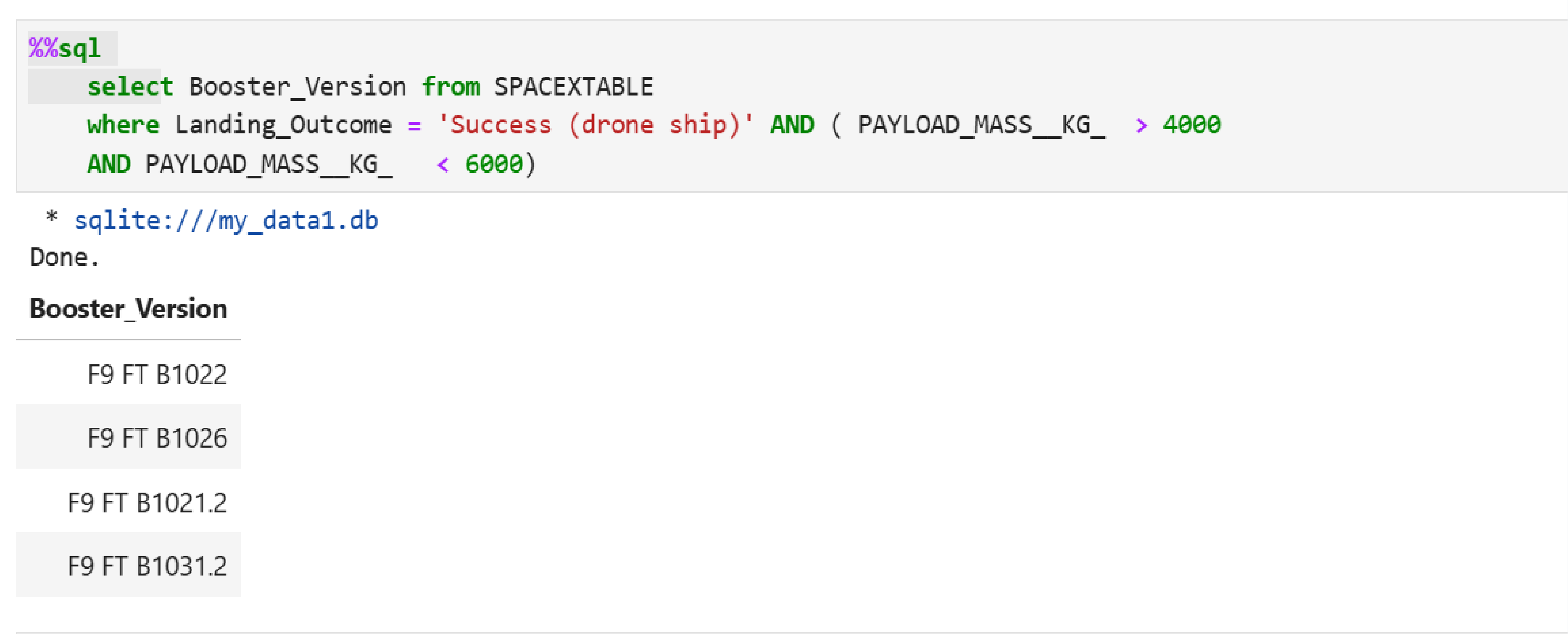
```
%%sql
    SELECT Date AS 'First Successful Ground Landing Date' FROM SPACEXTABLE
    where Landing_Outcome = 'Success (ground pad)' order by Date ASC LIMIT 1
```

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

First Successful Ground Landing Date
2015-12-22

- Order by ASC limit 1c is used to get the first landing date

Successful Drone Ship Landing with Payload between 4000 and 6000

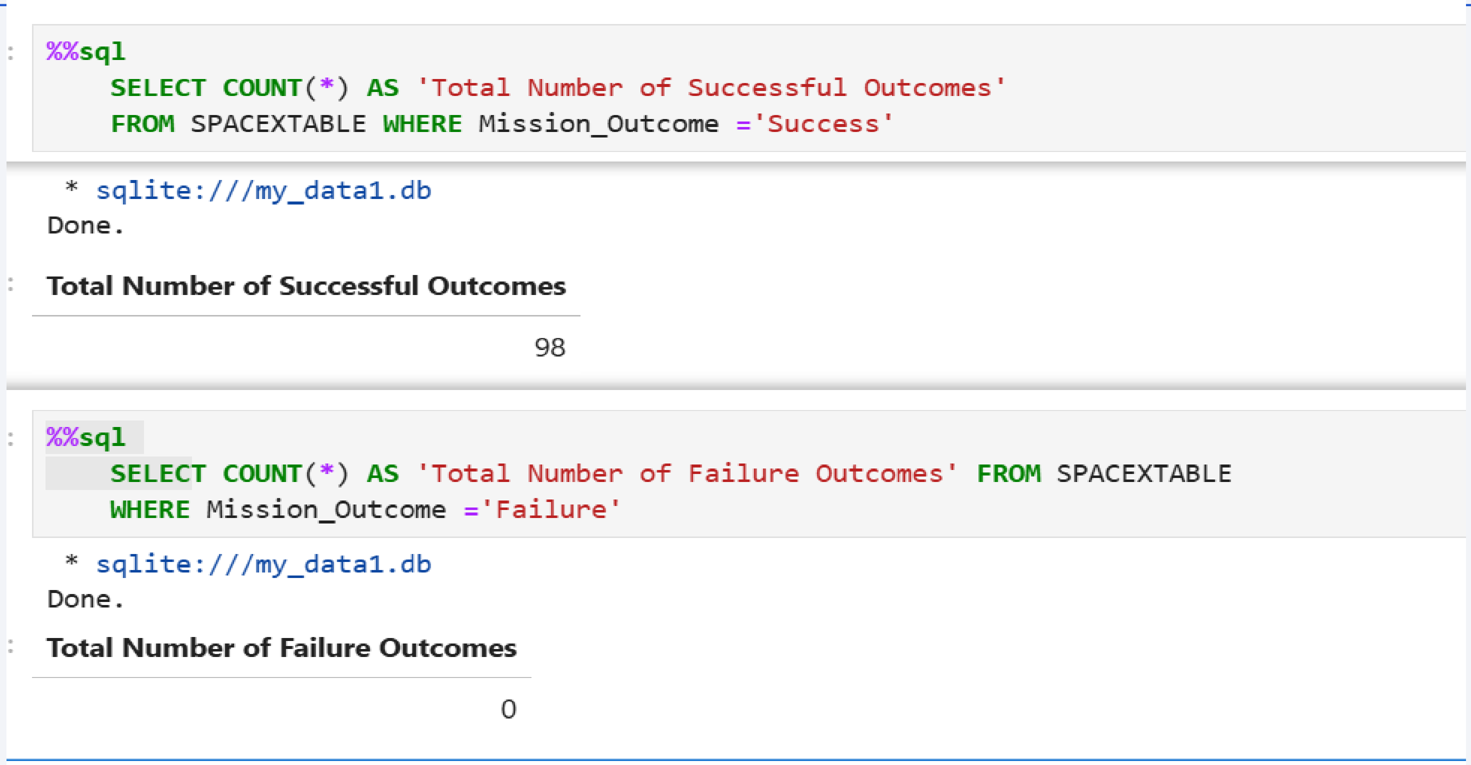


```
%%sql
    select Booster_Version from SPACEXTABLE
    where Landing_Outcome = 'Success (drone ship)' AND ( PAYLOAD_MASS__KG_  > 4000
    AND PAYLOAD_MASS__KG_   < 6000)
```

```
 * 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



```
%%sql
    SELECT COUNT(*) AS 'Total Number of Successful Outcomes'
    FROM SPACEXTABLE WHERE Mission_Outcome ='Success'
```

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

Total Number of Successful Outcomes
98

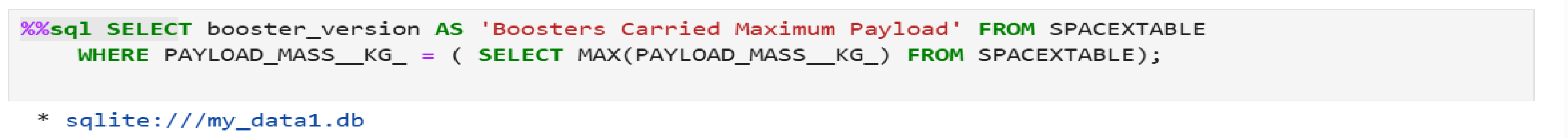
```
%%sql
    SELECT COUNT(*) AS 'Total Number of Failure Outcomes' FROM SPACEXTABLE
    WHERE Mission_Outcome ='Failure'
```

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

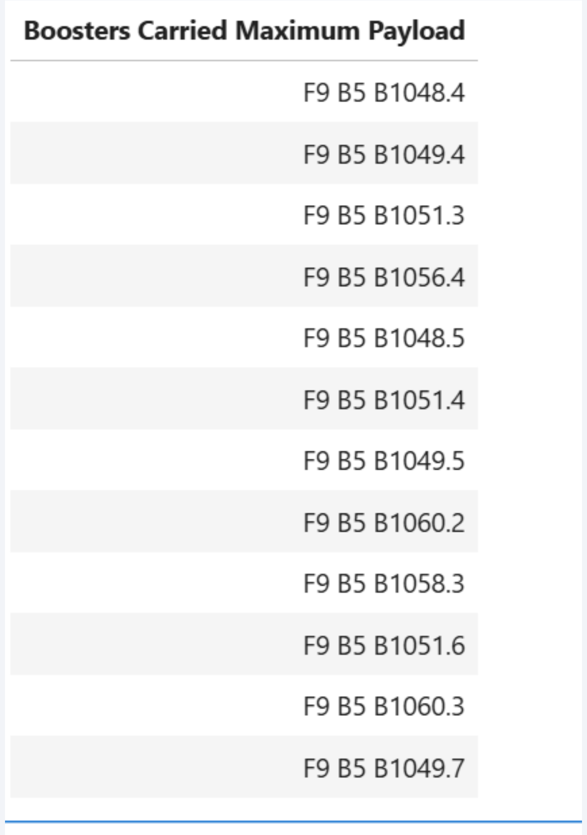
Total Number of Failure Outcomes
0

- Count function is used to calculate the number of successful and failure mission outcomes

Boosters Carried Maximum Payload



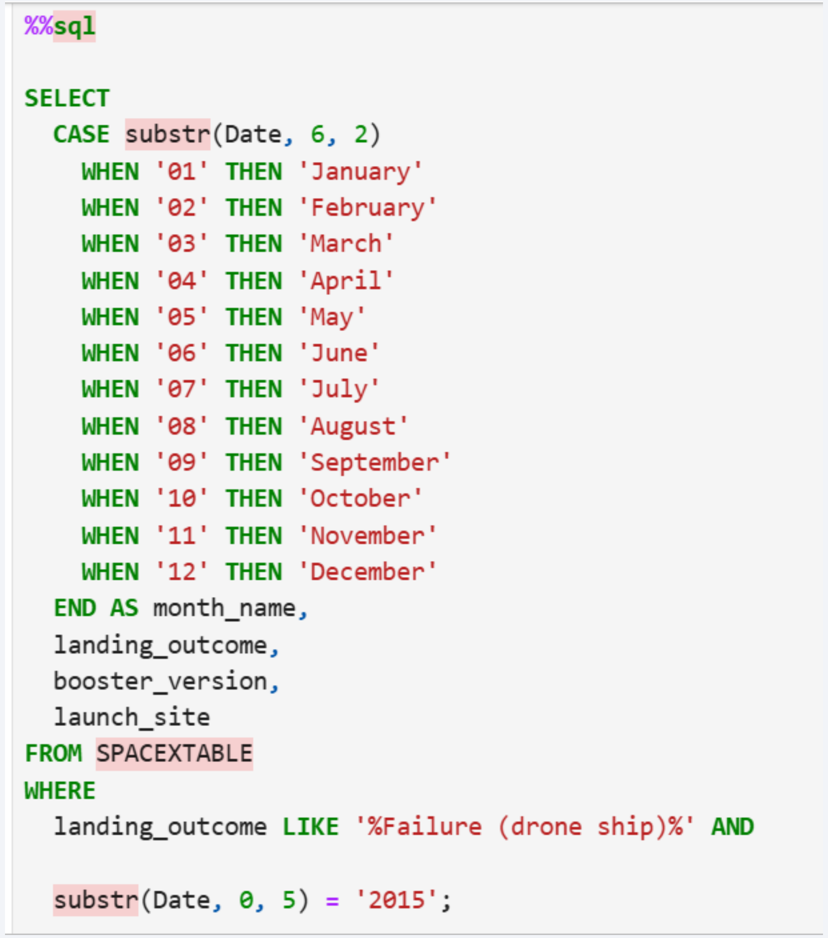
```
%%sql SELECT booster_version AS 'Boosters Carried Maximum Payload' FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = ( SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

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



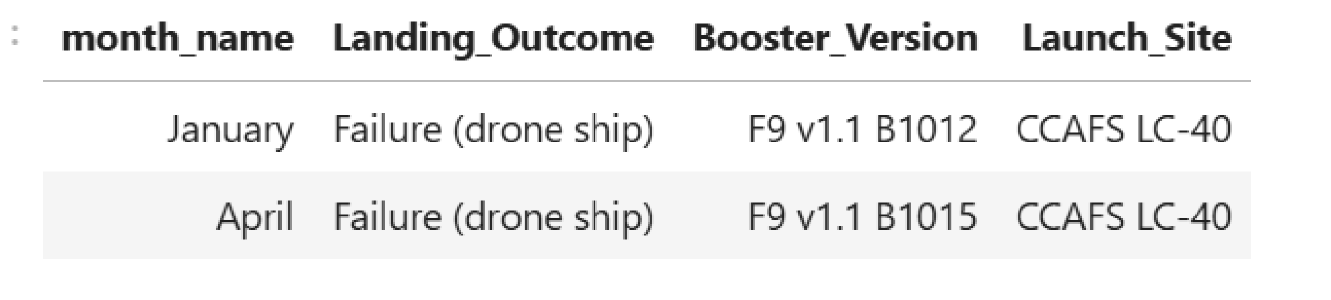
Boosters Carried Maximum Payload
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- Booster versions that have carried the maximum payload mass, using a subquery

2015 Launch Records



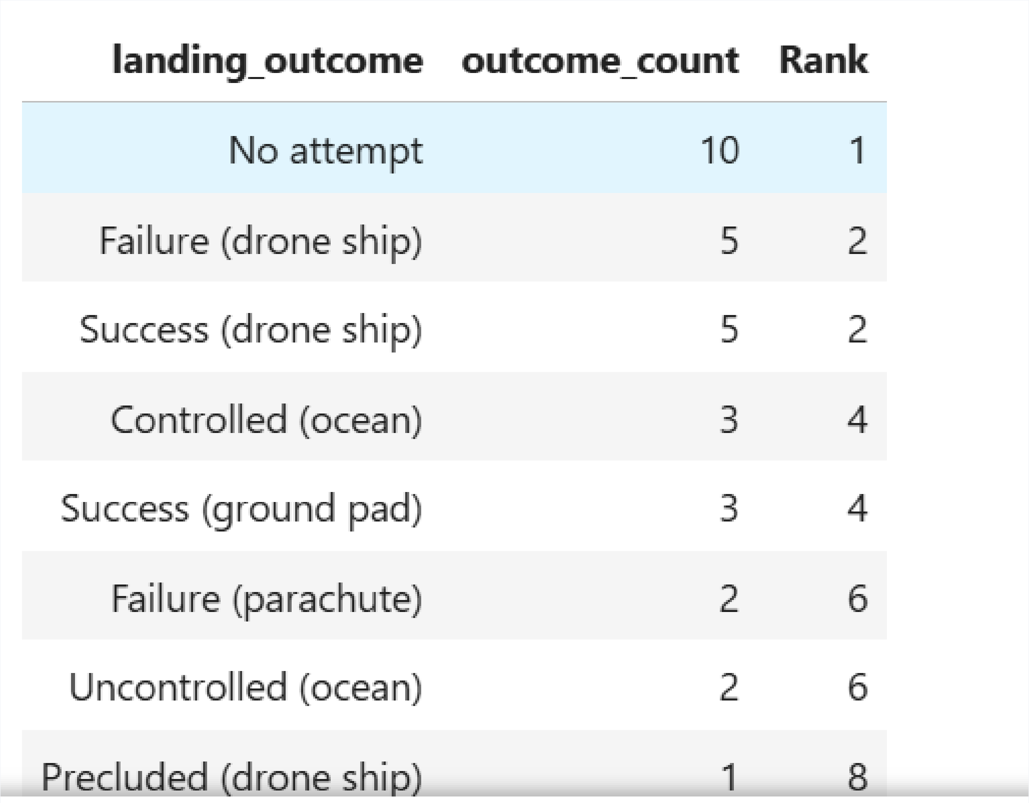
```
%%sql

SELECT
  CASE substr(Date, 6, 2)
    WHEN '01' THEN 'January'
    WHEN '02' THEN 'February'
    WHEN '03' THEN 'March'
    WHEN '04' THEN 'April'
    WHEN '05' THEN 'May'
    WHEN '06' THEN 'June'
    WHEN '07' THEN 'July'
    WHEN '08' THEN 'August'
    WHEN '09' THEN 'September'
    WHEN '10' THEN 'October'
    WHEN '11' THEN 'November'
    WHEN '12' THEN 'December'
  END AS month_name,
  landing_outcome,
  booster_version,
  launch_site
FROM SPACEXTABLE
WHERE
  landing_outcome LIKE '%Failure (drone ship)%' AND

  substr(Date, 0, 5) = '2015';
```



month_name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- List of the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

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



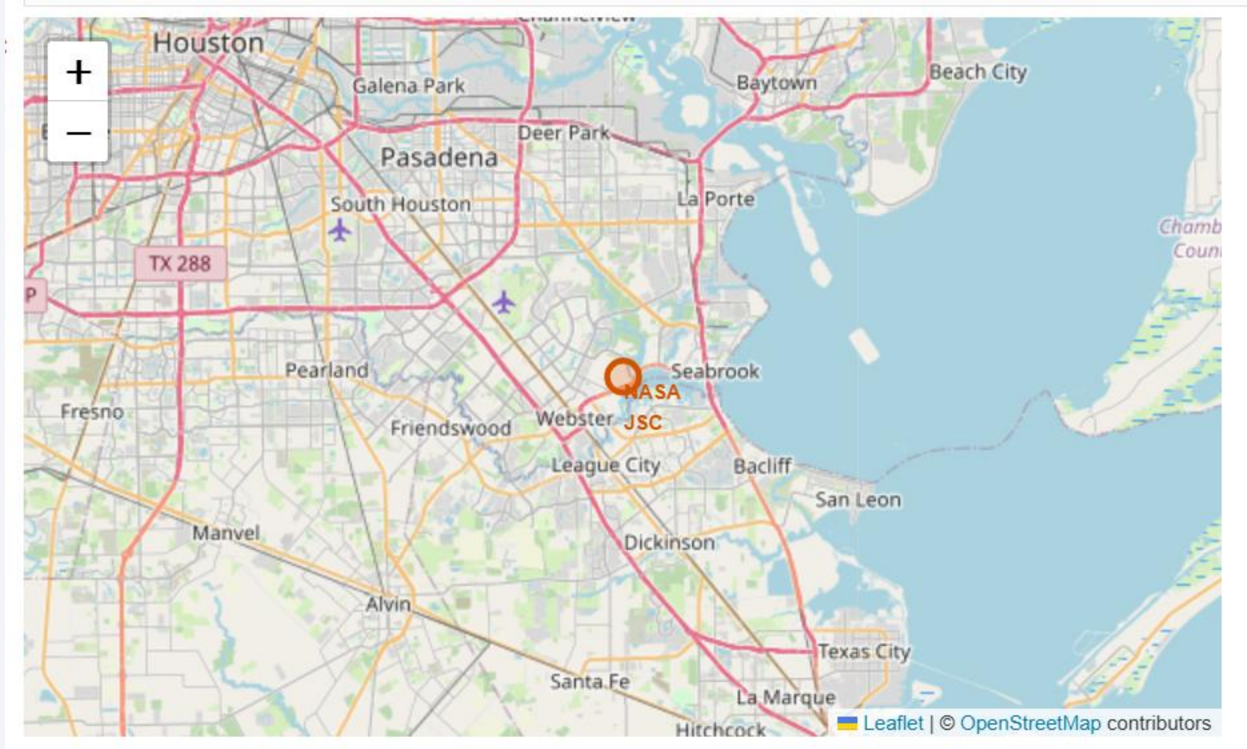
landing_outcome	outcome_count	Rank
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Controlled (ocean)	3	4
Success (ground pad)	3	4
Failure (parachute)	2	6
Uncontrolled (ocean)	2	6
Precluded (drone ship)	1	8

- Ranking of 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

Section 3

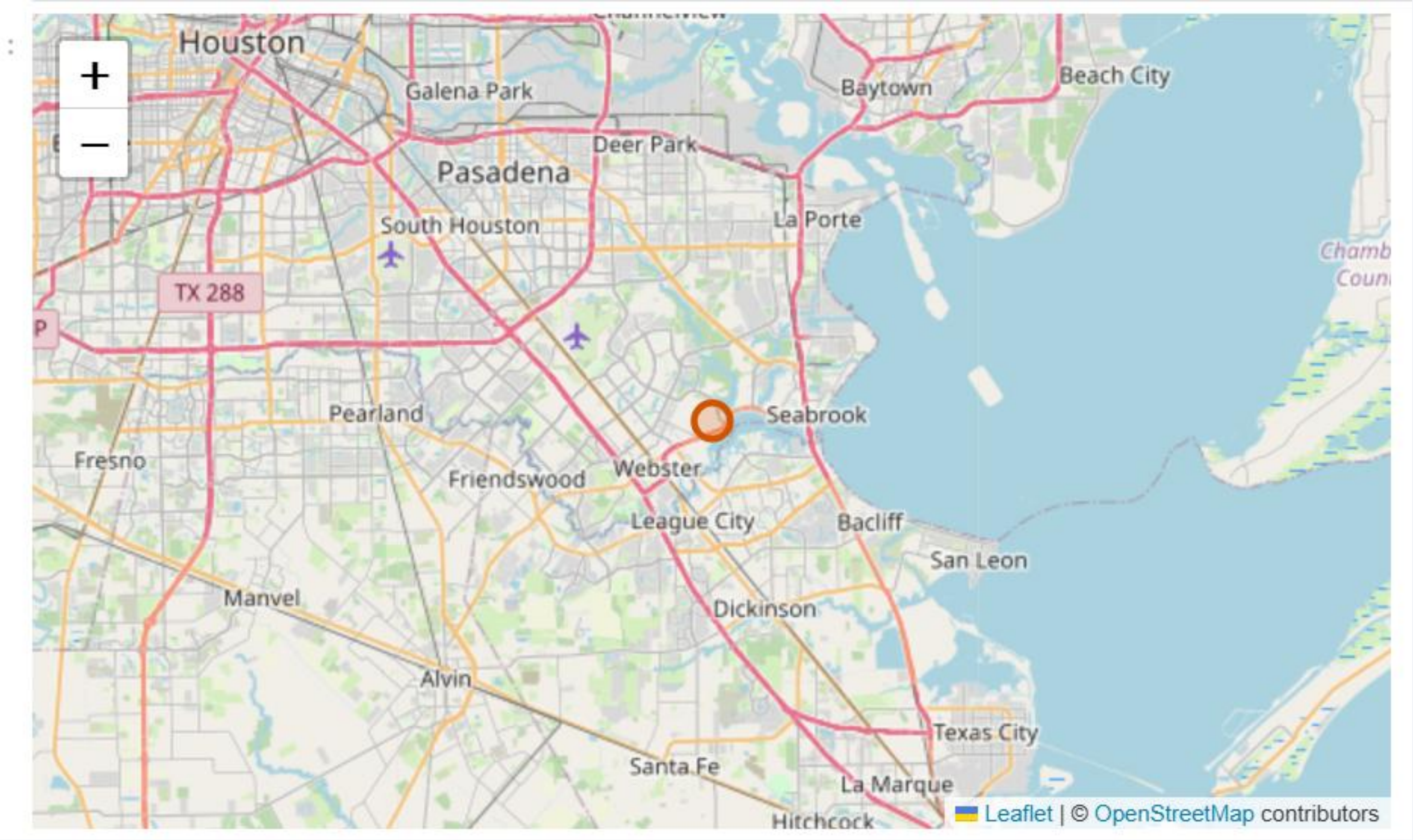
Launch Sites Proximities Analysis

All Launch Sites Map

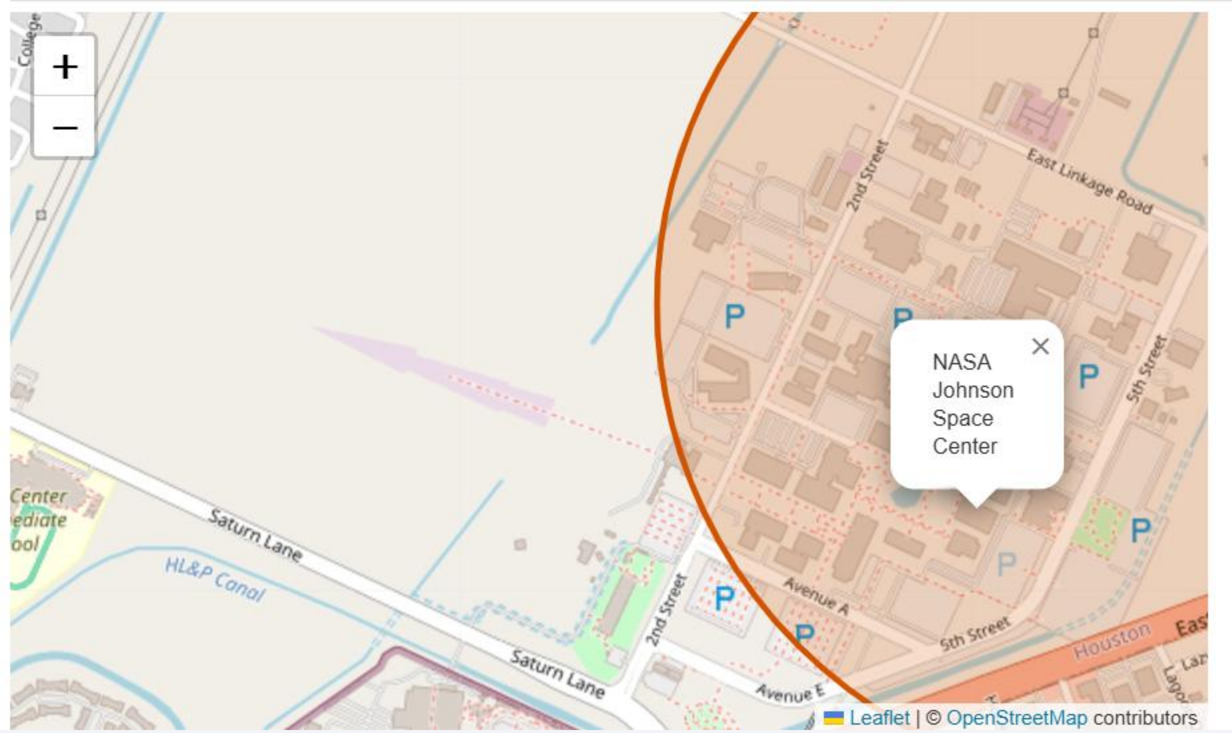


- This is a global map showing all the launch sites centered at NASA

Color-labeled launch outcomes on the map

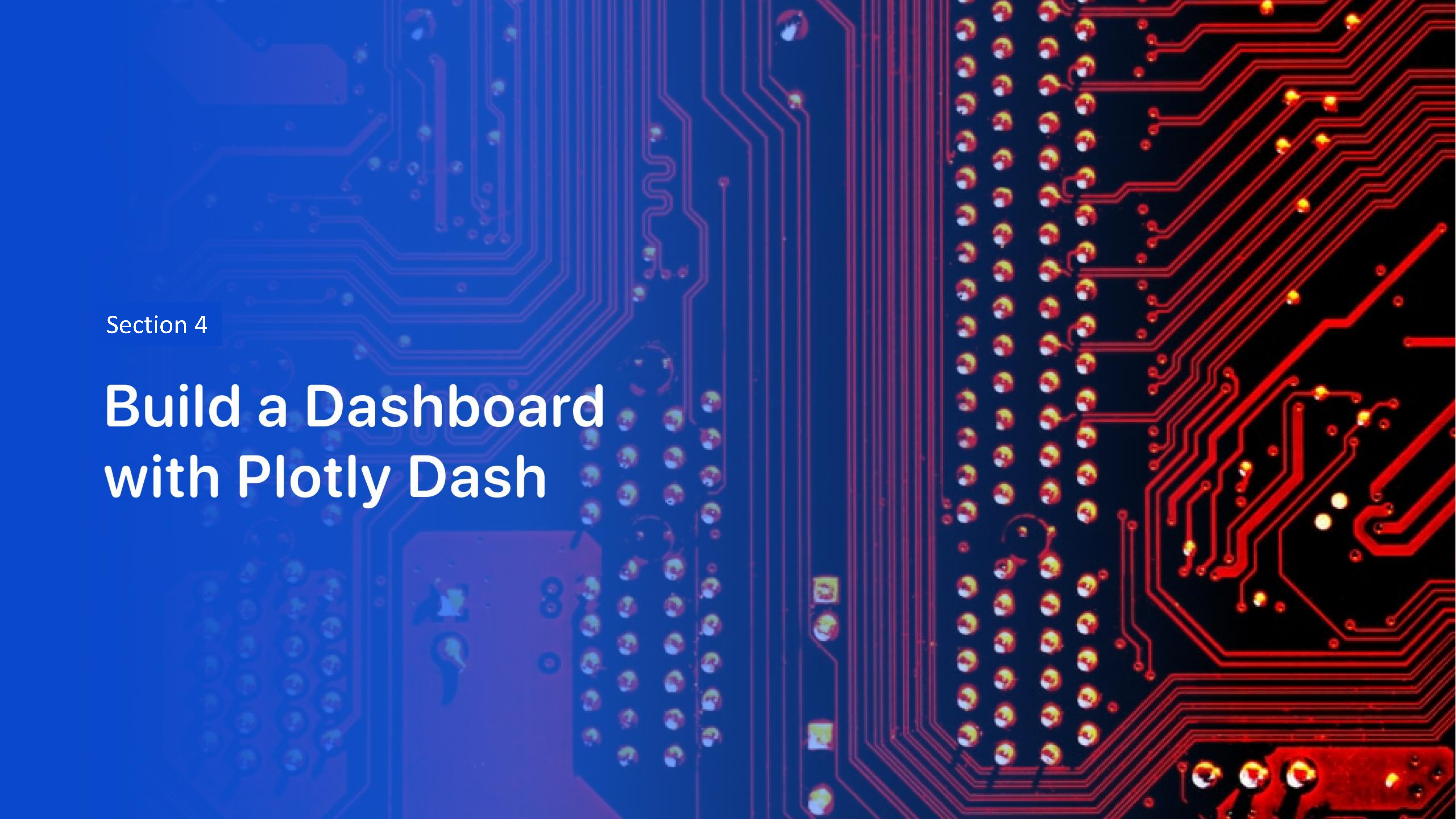


Launch Sites Proximities Map

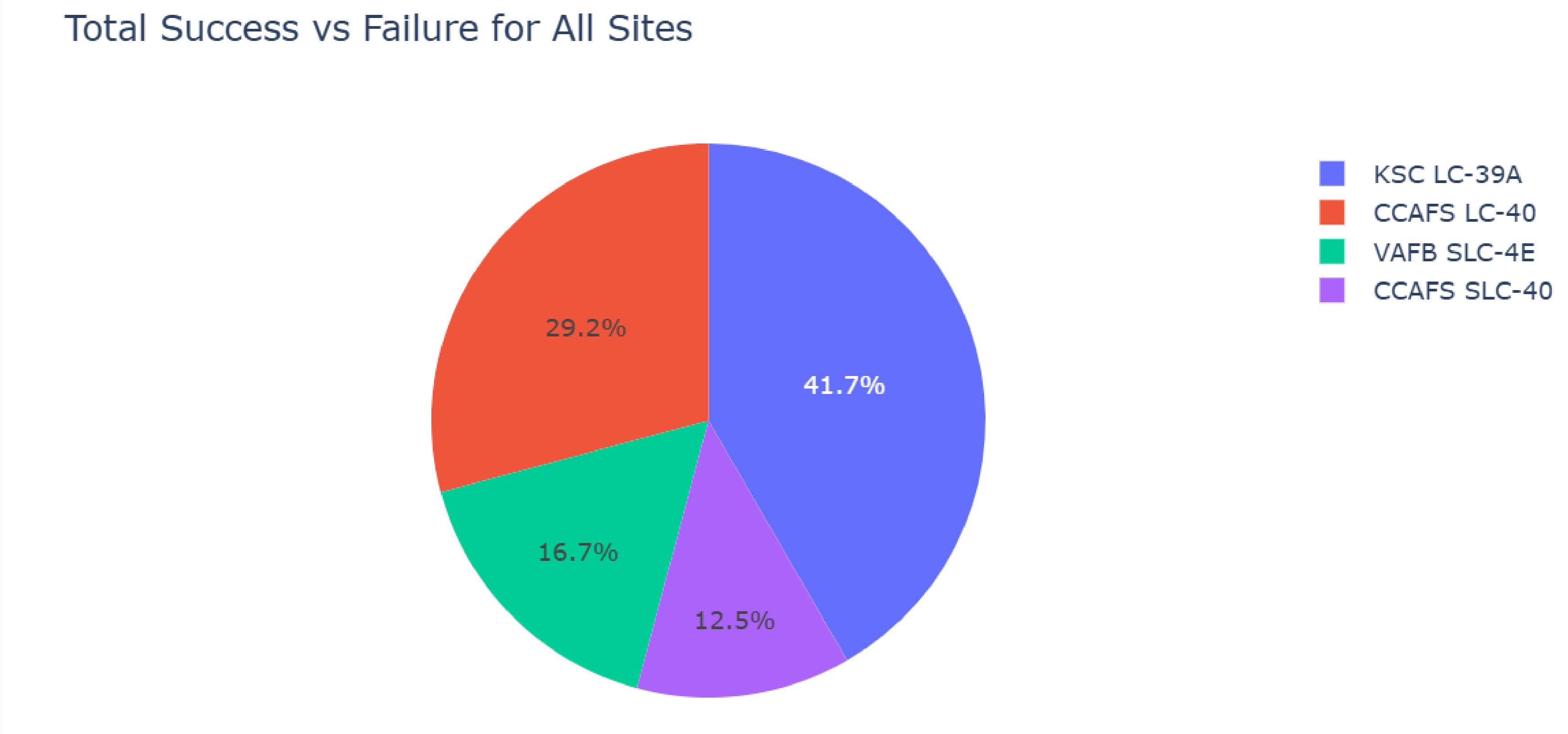


Section 4

Build a Dashboard with Plotly Dash

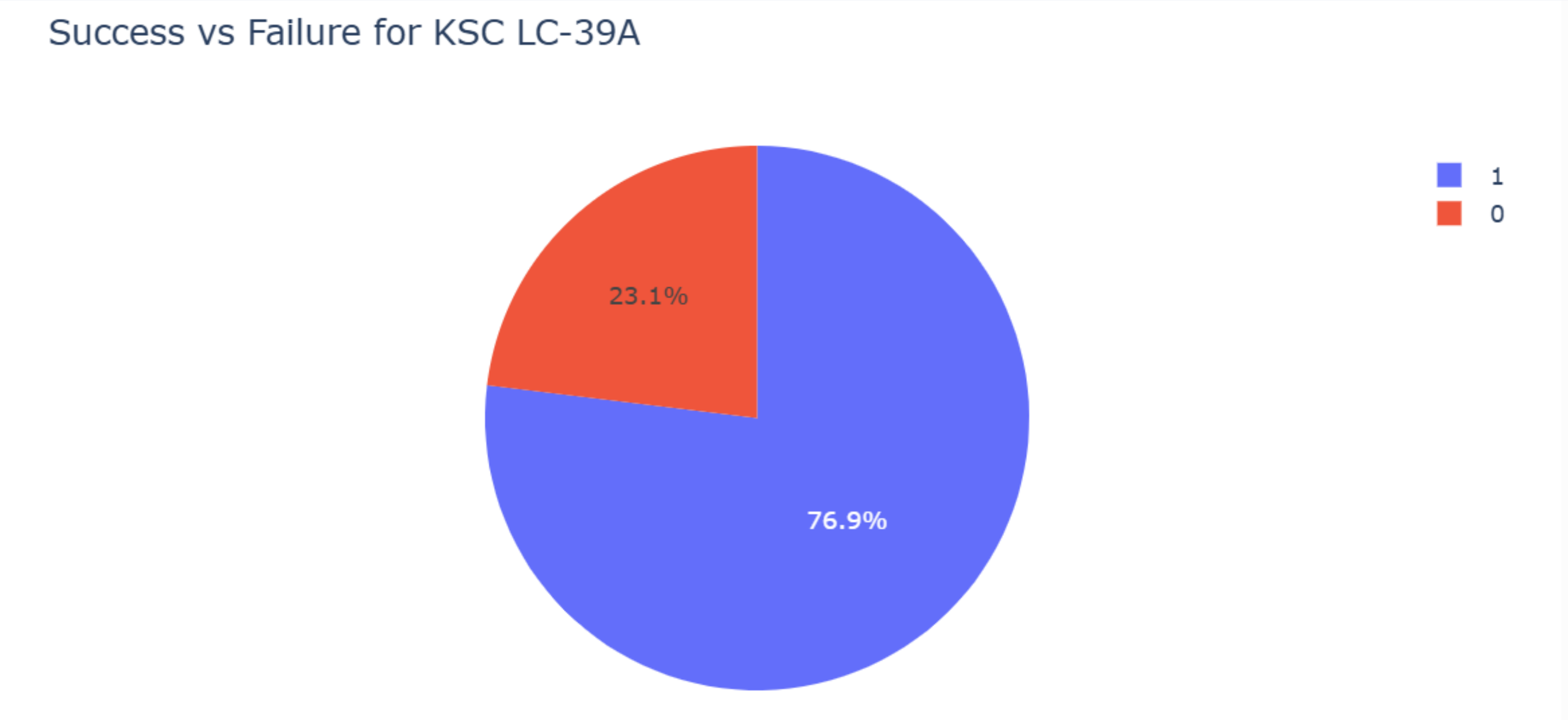


Launch Success Sites



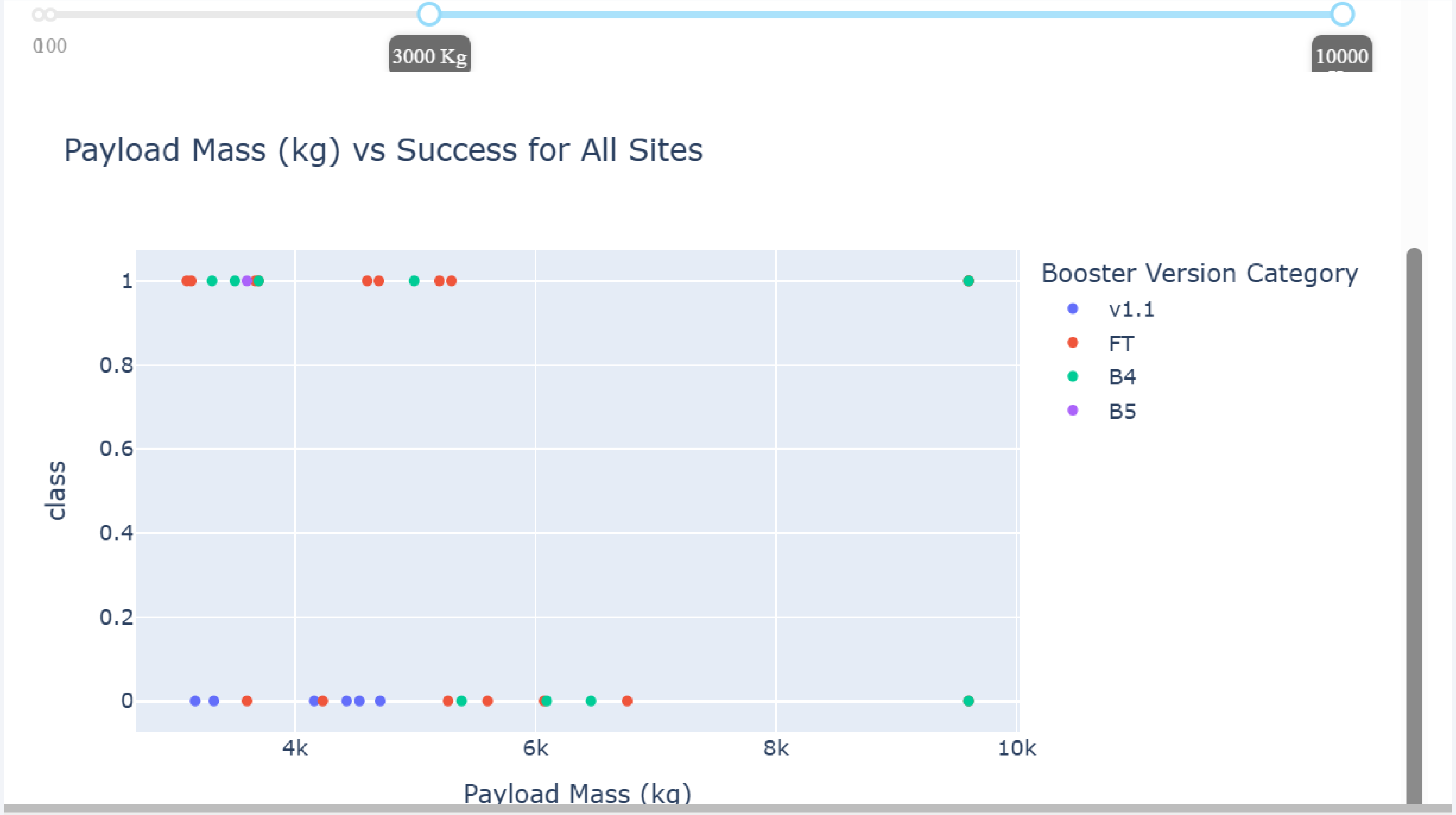
KSC LC-39A has the highest success rate while CCAFS SLC-40 has the lowest.

Launch Site with highest success ratio

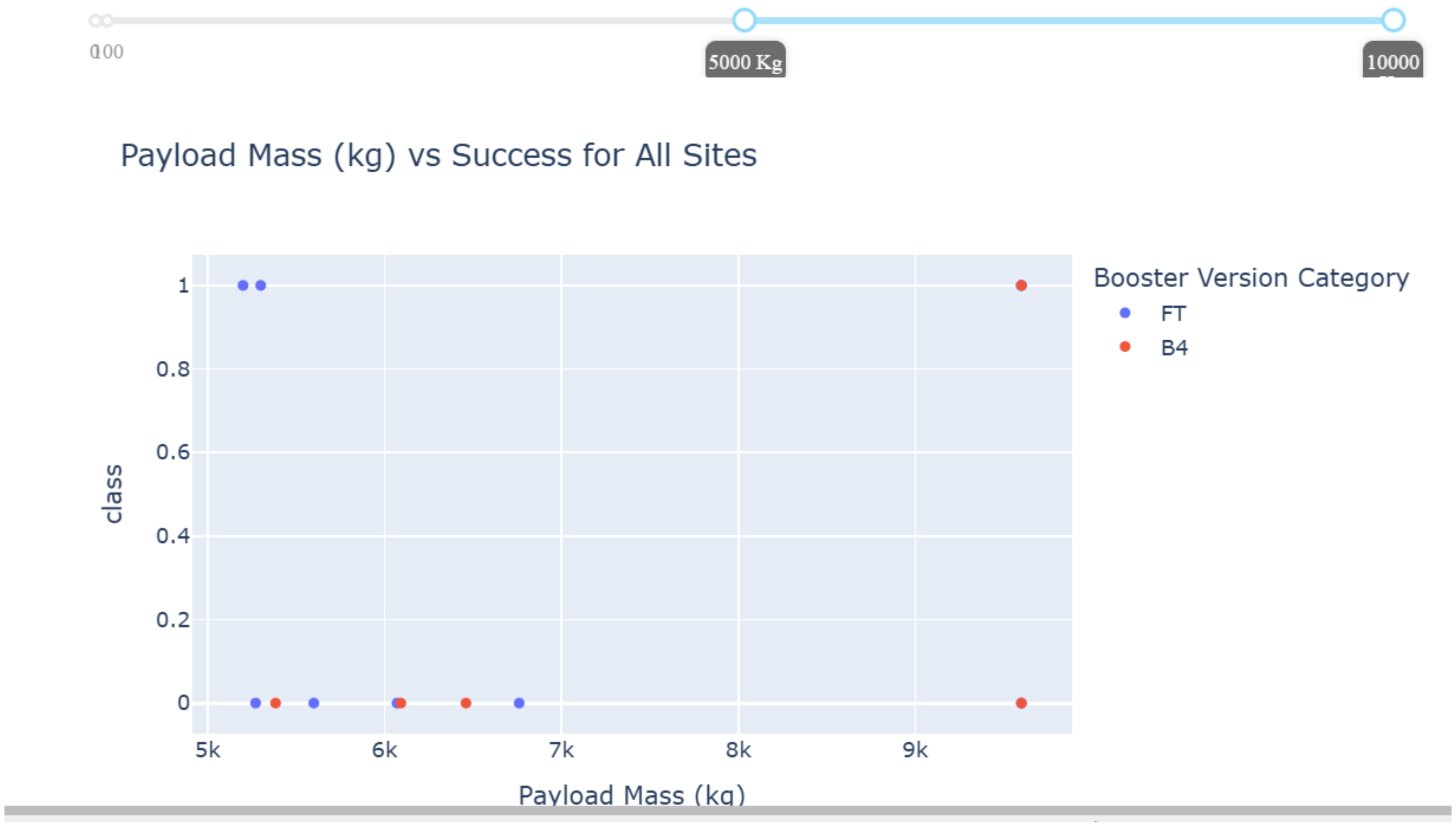


- This launch site has a success rate of approximately 77% which is fairly good.

Payload vs. Launch Outcome scatter



At a payload of 3000kg the FT booster version has the highest success rate and V1.1 has the lowest success rate

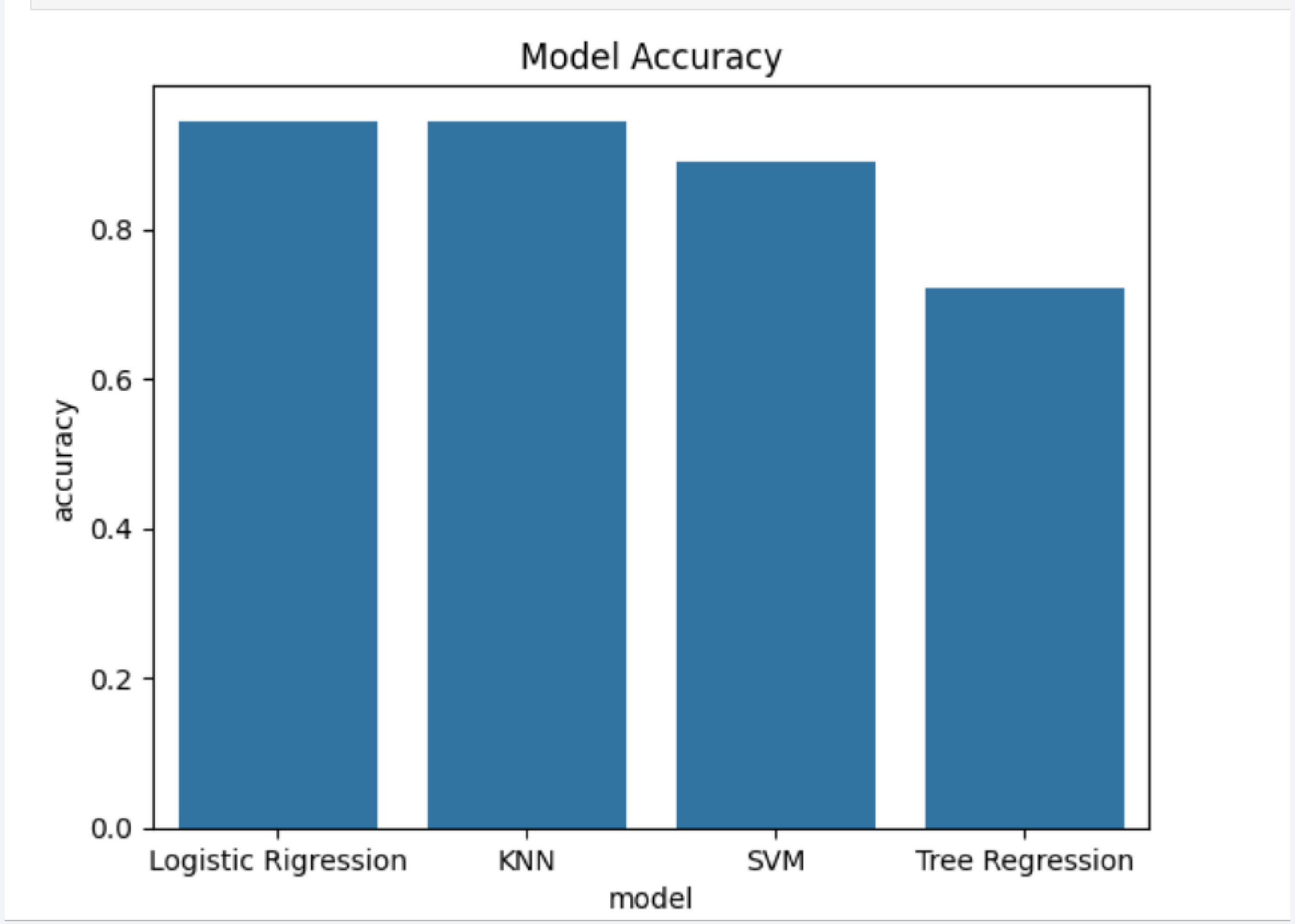


At a payload of 5000 kg, FT booster version has the highest success rate while B4 has the lowest failure rate.

Section 5

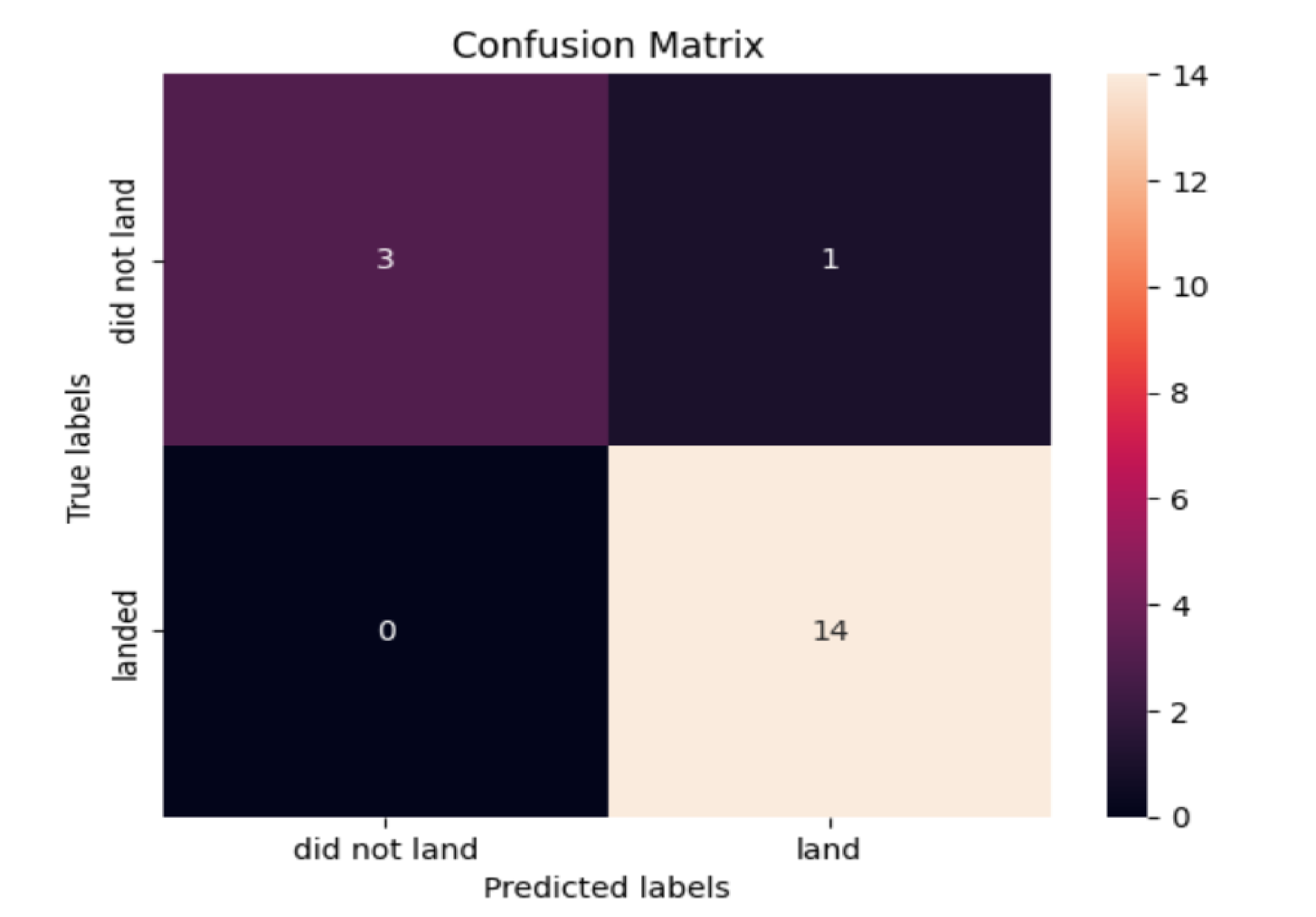
Predictive Analysis (Classification)

Classification Accuracy



- Logistic Regression has the highest accuracy

Confusion Matrix



- The logistic Regression model has 14 correct positives, 1 false positive, 3 true negatives and 0 false negatives.

Conclusions

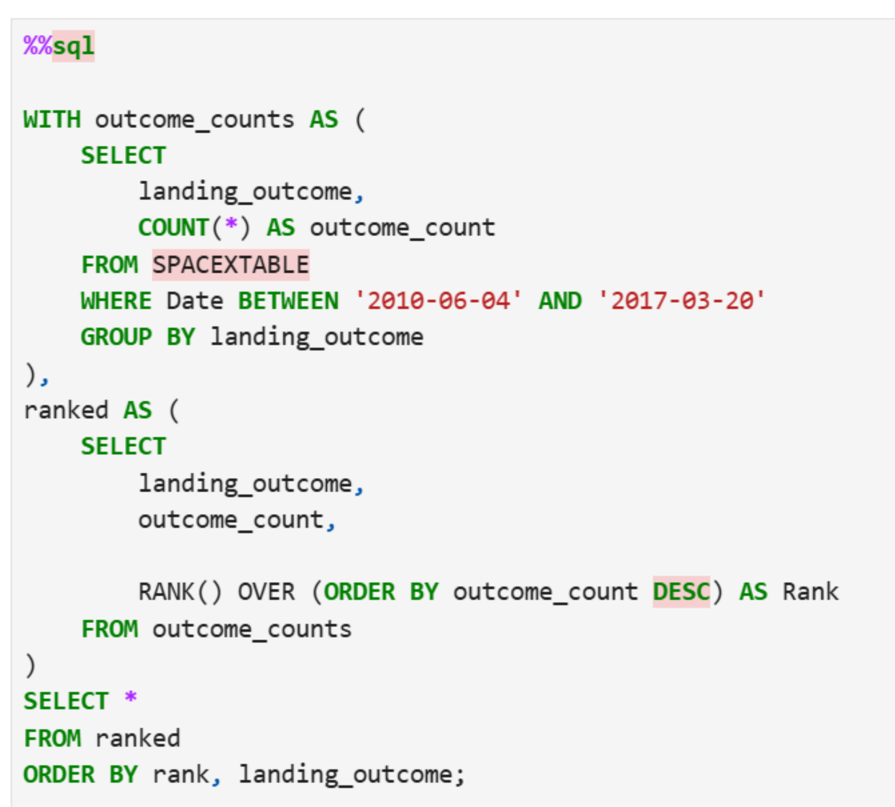
This capstone project successfully applied the core principles of data science to analyze SpaceX launch records. Key conclusions include:

- **KSC LC-39A** emerged as the most reliable launch site, while **FT booster versions** showed higher success rates, especially for heavier payloads.
- There is a clear upward trend in launch success over time, indicating operational improvements by SpaceX.
- Classification models validated the predictability of launch outcomes, with Logistic Regression delivering the best performance.
- Visual and spatial analysis through Folium and Plotly enhanced understanding of geographic and categorical factors influencing mission success.

The integration of data acquisition, wrangling, EDA, interactive visualization, and machine learning illustrates a complete data science workflow and delivers insights that could support strategic decisions in aerospace launch operations.

Appendix

- Rank SQL Query



```
%%sql

WITH outcome_counts AS (
    SELECT
        landing_outcome,
        COUNT(*) AS outcome_count
    FROM SPACEXTABLE
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY landing_outcome
),
ranked AS (
    SELECT
        landing_outcome,
        outcome_count,

        RANK() OVER (ORDER BY outcome_count DESC) AS Rank
    FROM outcome_counts
)
SELECT *
FROM ranked
ORDER BY rank, landing_outcome;
```

Thank you!