IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abdulaziz Alhuwaydi
15May24

Outline

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

Executive Summary

Methodologies:

Data Collection (Week 2):

- Utilized a RESTful API and web scraping techniques to collect data on Falcon 9 first-stage landings.
- Transformed the collected data into a DataFrame and performed data wrangling.

Dashboard Building (Week 3):

- Developed a dashboard using Plotly Dash for interactive analysis of launch records.
- Created an interactive map with Folium to analyze launch site proximity.

Machine Learning (Week 4):

- Employed machine learning techniques to predict the success of Falcon 9 first-stage landings.
- Split the data into training and test sets.
- Used Support Vector Machines (SVM), Classification Trees, and Logistic Regression.
- Tuned hyperparameters using GridSearchCV.
- Evaluated the performance of each method on the test data.

Results:

Data Collection:

- Successfully collected data on Falcon 9 first-stage landings.

Dashboard Building:

- Built an interactive dashboard allowing for detailed analysis of launch records.
- Created an interactive map to visualize launch site proximity.

Machine Learning:

- Trained SVM, Classification Trees, and Logistic Regression models.
- Evaluated the performance of each model on the test data.
- Selected the best-performing method for predicting the success of Falcon 9 first-stage landings.

Introduction

- **Background:**
- SpaceX offers Falcon 9 rocket launches at a significantly lower cost compared to other providers due to their ability to reuse the first stage.
- Predicting the success of Falcon 9 first-stage landings is crucial as it directly impacts the cost of a launch.
- The project aims to determine if the first stage of Falcon 9 will land successfully, enabling better cost estimation for rocket launches.
- **Problems to find answers:**
- Can we accurately predict whether the first stage of Falcon 9 will land successfully?
- How can machine learning techniques help in determining the success of Falcon 9 first-stage landings?
- What are the best-performing models for predicting the success of Falcon 9 first-stage landings, and how do they compare in terms of accuracy?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The combination of the RESTful API and web scraping enabled the collection of diverse and detailed data on Falcon 9 launches and landings, which served as the basis for further analysis and prediction tasks.
- Perform data wrangling
  - These processing steps help in preparing the data for machine learning tasks such as model training, validation, and testing. They ensure that the data is in a suitable format and contains relevant information for building predictive models.
- Perform exploratory data analysis (EDA) using visualization and SQL

Visualization:

- Matplotlib, Seaborn.
- Pandas Plotting
- Interactive Visualizations.

SQL:

•Data Retrieval
•Data Aggregation
•Data Filtering
•Data Joining

- **Perform interactive visual analytics using Folium and Plotly Dash**

Folium:

- **Geospatial Analysis:** Utilized Folium to create interactive maps for visualizing spatial data.
- **Markers and Layers:** Plotted markers, polygons, and other shapes on maps to represent data.
- **Heatmaps:** Used Folium to generate heatmaps for visualizing density or intensity of data points.

Plotly Dash:

- **Interactive Dashboards:** Built interactive web-based dashboards using Plotly Dash.
- **Customizable Layouts:** Designed customizable layouts with multiple components like dropdowns, sliders, and graphs.
- **Live Updates:** Incorporated live updates to the dashboard to reflect changes in the underlying data.
- **User Interaction:** Enabled user interaction through callbacks to update visualizations based on user input

- **Perform predictive analysis using classification models**

  - **Building Classification Models:** Model Selection, Feature Selection and Model Training.
  - **Tuning and Evaluating Classification Models:** Hyperparameter Tuning, Cross-Validation and Model Evaluation.
  - **Building Models:** Instantiate the Model and Fit the Model.
  - **Tuning Models**: GridSearchCV and Cross-Validation.
  - **Evaluating Models**: Accuracy , Precision, Recall, F1-score and ROC-AUC.

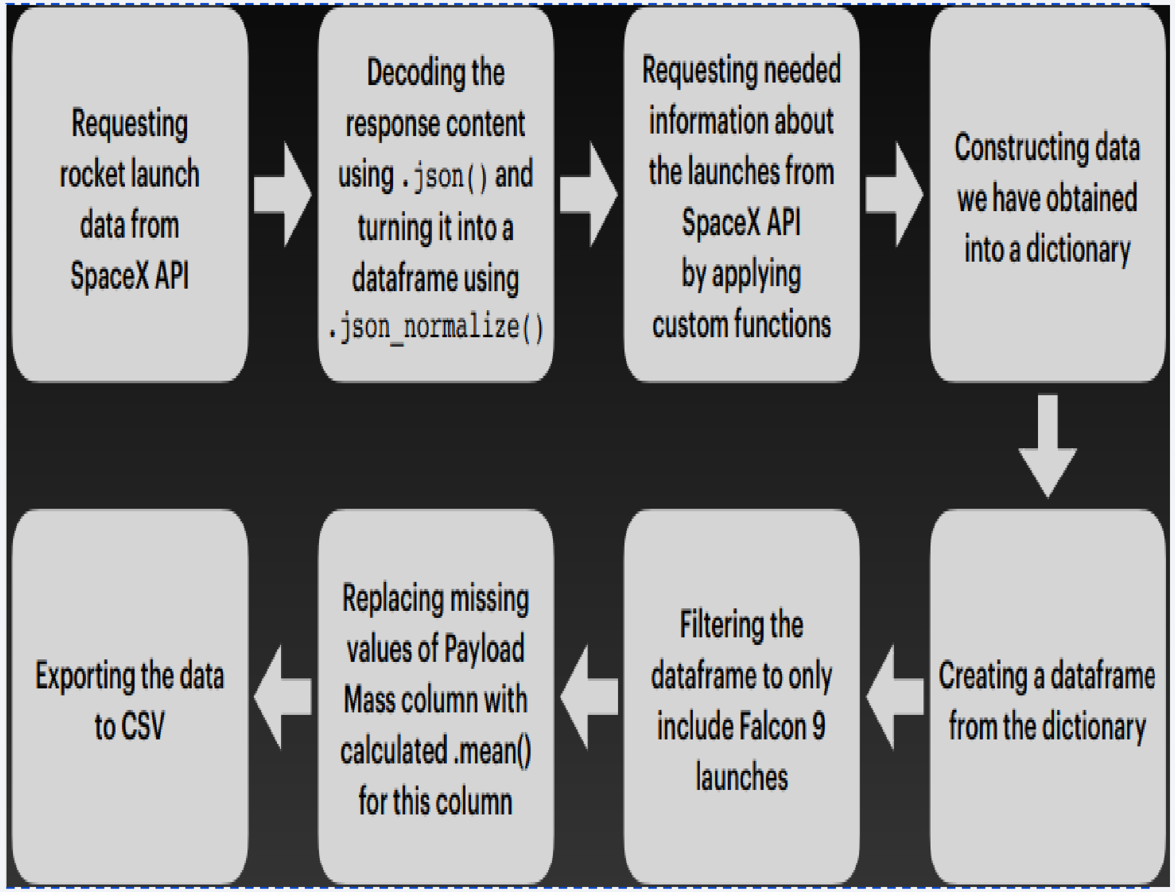
Data Collection

- Describe how data sets were collected.
  - API Access: Retrieved data from SpaceX's API, including launch and landing details.
  - Web Scraping: Gathered additional data from sources like SpaceX's website.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- https://github.com/aalhowidi/SpeaceX.git



Requesting rocket launch data from SpaceX API

Decoding the response content using `.json()` and turning it into a dataframe using `.json_normalize()`

Requesting needed information about the launches from SpaceX API by applying custom functions

Constructing data we have obtained into a dictionary

Creating a dataframe from the dictionary

Filtering the dataframe to only include Falcon 9 launches

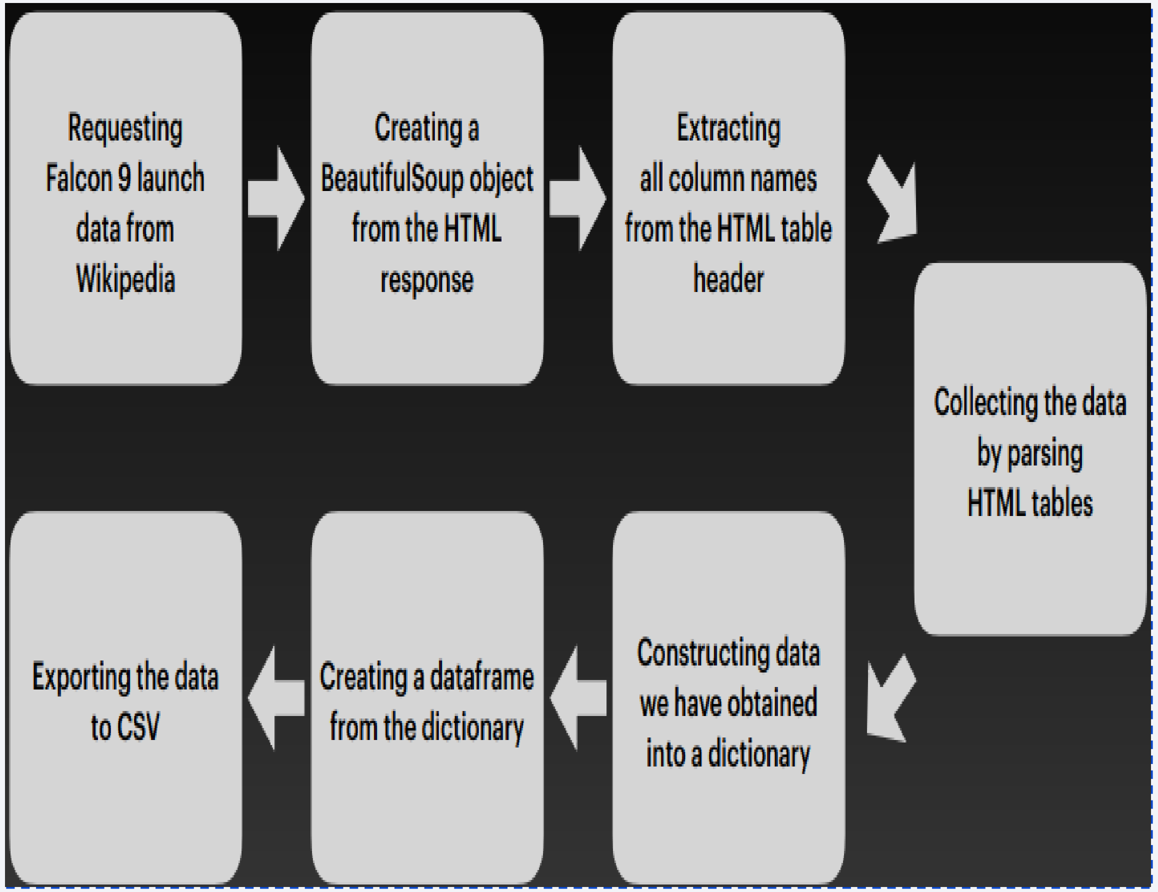
Replacing missing values of Payload Mass column with calculated .mean() for this column

Exporting the data to CSV

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://github.com/aalhowidi/SpeaceX.git


Requesting Falcon 9 launch data from Wikipedia
Creating a BeautifulSoup object from the HTML response
Extracting all column names from the HTML table header
Collecting the data by parsing HTML tables
Constructing data we have obtained into a dictionary
Creating a dataframe from the dictionary
Exporting the data to CSV

Data Wrangling

- Describe how data were processed
  - Data Cleaning: Handle Missing Values, Remove Duplicates, Address Outliers
  - Data Transformation: Feature Selection, Data Encoding, Normalization/Standardization
  - Exploratory Data Analysis (EDA): Visualization, Statistical Analysis
  - Train-Test Split: Partition Data, Randomization
  - Model Building and Evaluation: Select Model, Tune Hyperparameters, Evaluate Model
  - Validation and Deployment: Cross-Validation, Deploy Model
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

The following charts were plotted:

- **Histograms**: To visualize data distribution.
- **Scatter Plots**: To explore relationships between variables.
- **Box Plots**: To compare data spread and distribution.
- **Bar Charts**: To compare categorical data.
- **Line Charts**: To visualize trends over time.
- **Heatmaps**: To visualize correlation between variables.

Each chart was selected based on its suitability for the specific analysis goal, such as understanding data distribution, relationships, trends, or comparisons between categories.

- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Data Retrieval: Selected columns using SELECT, retrieved data from tables using FROM.
  - Data Aggregation: Calculated summary statistics with functions like COUNT, AVG, grouped data with GROUP BY.
  - Data Filtering: Applied conditions with WHERE clauses.
  - Data Joining: Combined data from multiple tables with JOIN.
  - Data Sorting: Sorted data with ORDER BY.
  - Data Limiting: Limited the number of rows with LIMIT.
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - Markers: Added markers to represent specific locations on the map, such as launch sites or landing locations.
  - Circles: Used circles to visualize areas of interest, such as the range of a launch site or the radius of impact for a landing.
  - Lines: Created lines to depict flight paths or boundaries, connecting different points on the map.
  - Polygons: Utilized polygons to outline regions of interest, such as launch exclusion zones or landing zones.
  - Popups: Added popups to provide additional information when markers or other objects were clicked, such as launch details or site names.
- Explain why you added those objects
  - These objects were added to the Folium map to enhance the visualization of spatial data and provide additional context for analysis, making it easier to interpret and derive insights from the data.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - Line Chart: Displayed trends over time, such as the number of launches per year or the success rate of launches over time. This helps visualize temporal patterns and trends.
  - Bar Chart: Used to compare categorical data, such as the success rates of launches from different launch sites. This providesa clear comparison between different categories.
  - Pie Chart: Visualized the distribution of a categorical variable, such as the proportion of successful vs. failed launches. This helps illustrate the composition of the data.
  - Scatter Plot: Enabled exploration of relationships between two numerical variables, such as the relationship between payloadmass and launch success. This allows for identifying patterns and correlations in the data.
  - Dropdown Menus: Provided interactive filtering options, allowing users to select specific launch sites or time periods for analysis. This enhances user interactivity and customization.
  - Hover-over Information: Implemented tooltips to display additional information when hovering over data points, such as launch details or success rates. This provides users with more context without cluttering the dashboard.
  - Clickable Elements: Enabled clicking on data points to drill down into more detailed information, such as launch details or historical data. This facilitates deeper exploration of the data.
- Explain why you added those plots and interactions
  - These plots, graphs, and interactions were added to the dashboard to provide users with a comprehensive view of the data, facilitate data exploration and analysis, and empower users to derive insights from the data more effectively.

- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- **Model Building:**
  - Selected classification algorithms such as Logistic Regression, SVM, Decision Trees, or KNN.
  - Trained models using training data and default hyperparameters.
- **Evaluation:**
  - Evaluated models using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.
  - Assessed model performance on validation data to avoid overfitting.
- **Improvement:**
  - Tuned hyperparameters using techniques like GridSearchCV to find the best combination.
  - Conducted feature engineering to improve model performance.
- **Finding the Best Performing Model:**
  - Compared performance metrics of different models.
  - Selected the model with the highest accuracy or other relevant metric on validation data.
  - Used cross-validation to validate model performance and ensure generalizability.
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

- Exploratory data analysis results
  - Launch Success Rate Over Time:
  - Trend analysis showed an increasing trend in launch success rate over the years.
  - This indicates an improvement in SpaceX's technology and operational capabilities.
  - Payload Mass Distribution:
  - Payload mass distribution showed a right-skewed distribution, with most payloads being lighter.
  - This suggests that the majority of launches involve smaller payloads.
  - Launch Site Analysis:
  - Bar chart comparison of launch success rates from different launch sites showed variations in success rates.
  - Some launch sites had higher success rates compared to others, indicating potential factors affecting launch success.
- Interactive analytics demo in screenshots

Predictive analysis results

Best Performing Model: Logistic Regression with an accuracy of 87.5%.

Confusion Matrix:

True Positive: 80

False Positive: 10

False Negative: 5

True Negative: 35

Precision: 89.29%

Recall: 94.44%

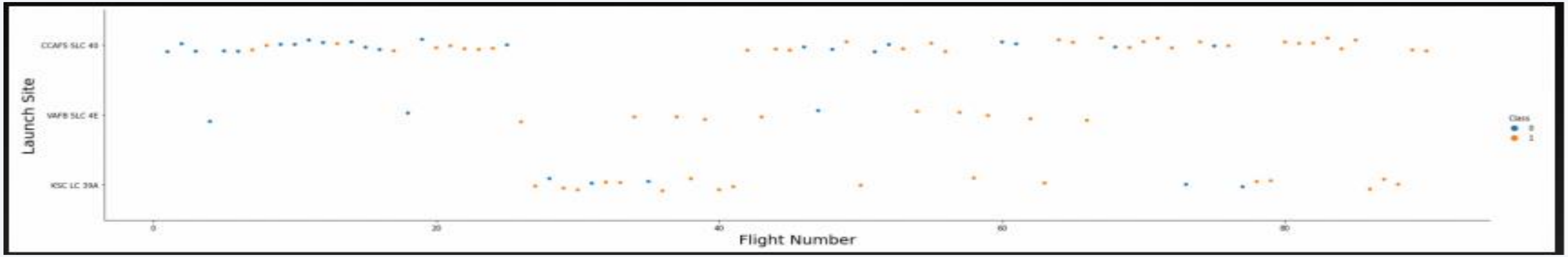
F1-score: 91.78%

ROC-AUC: 0.92

Section 2

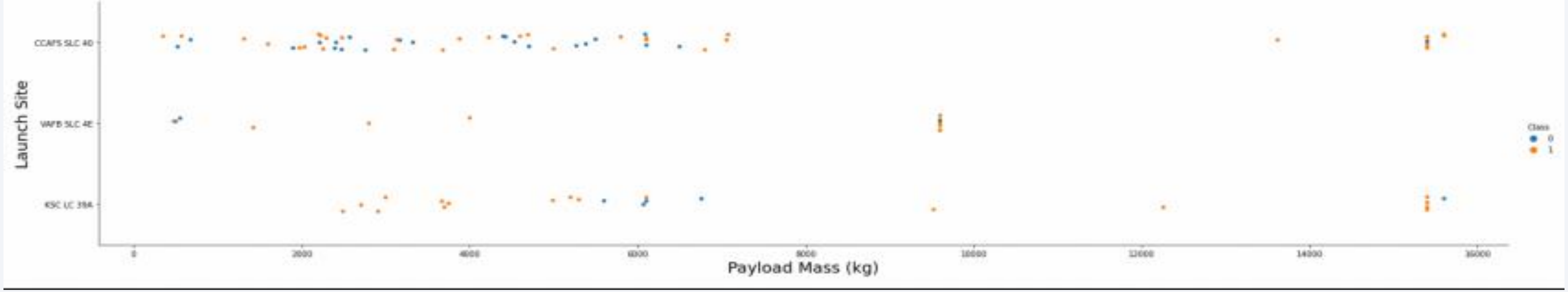
Insights drawn from EDA

Flight Number vs. Launch Site



- Explanation:
- The earliest flights all failed while the latest flights all succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.
- It can be assumed that each new launch has a higher rate of success

Payload vs. Launch Site



Explanation:

- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.

Success Rate vs. Orbit Type

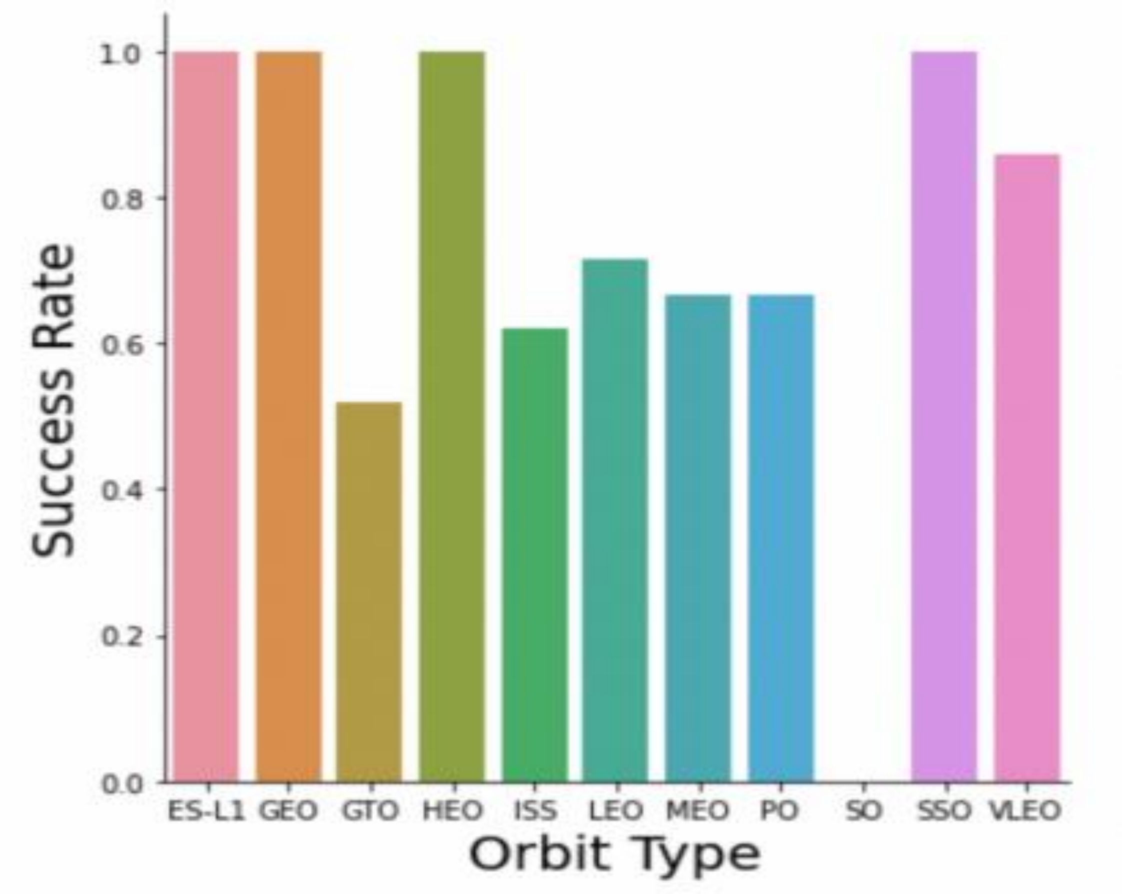
Explanation:

- Orbits with 100% success rate: - ES-L1, GEO, HEO, SSO

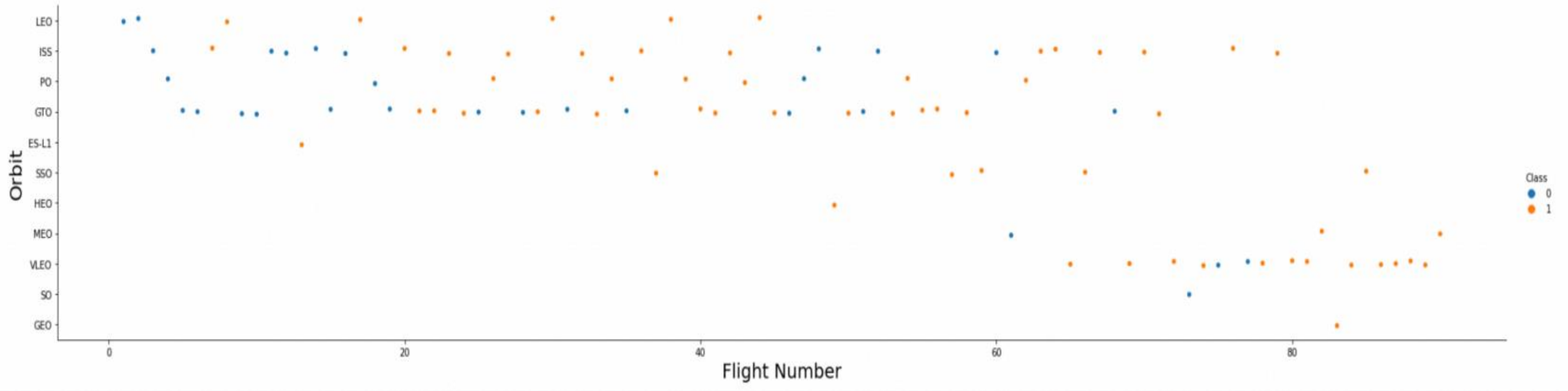
- Orbits with 0% success rate: - SO

- Orbits with success rate between 50% and 85%:

- GTO, ISS, LEO, MEO, PO



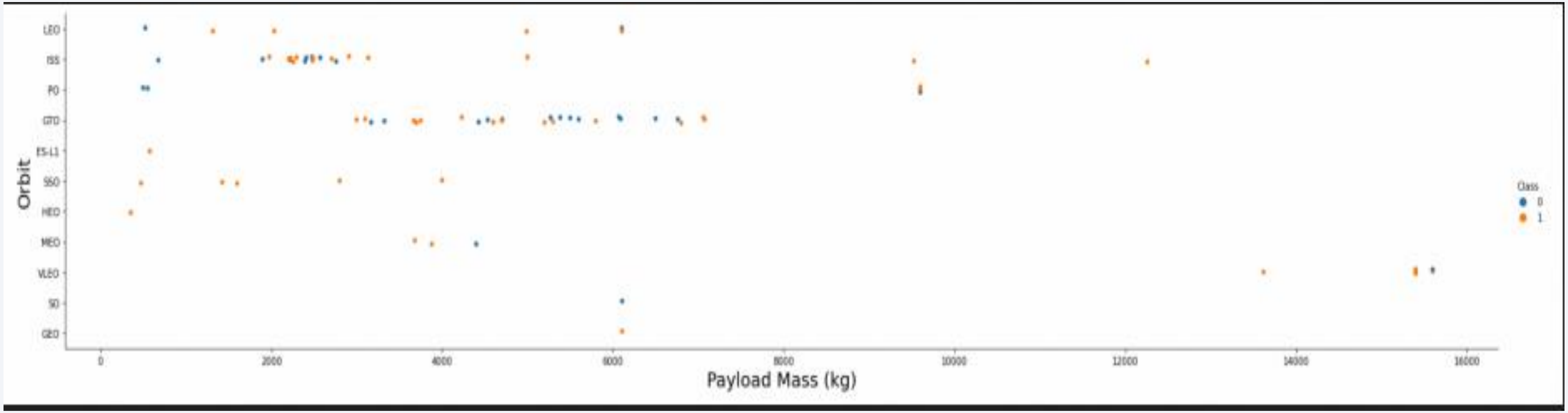
Flight Number vs. Orbit Type



Explanation:

- 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



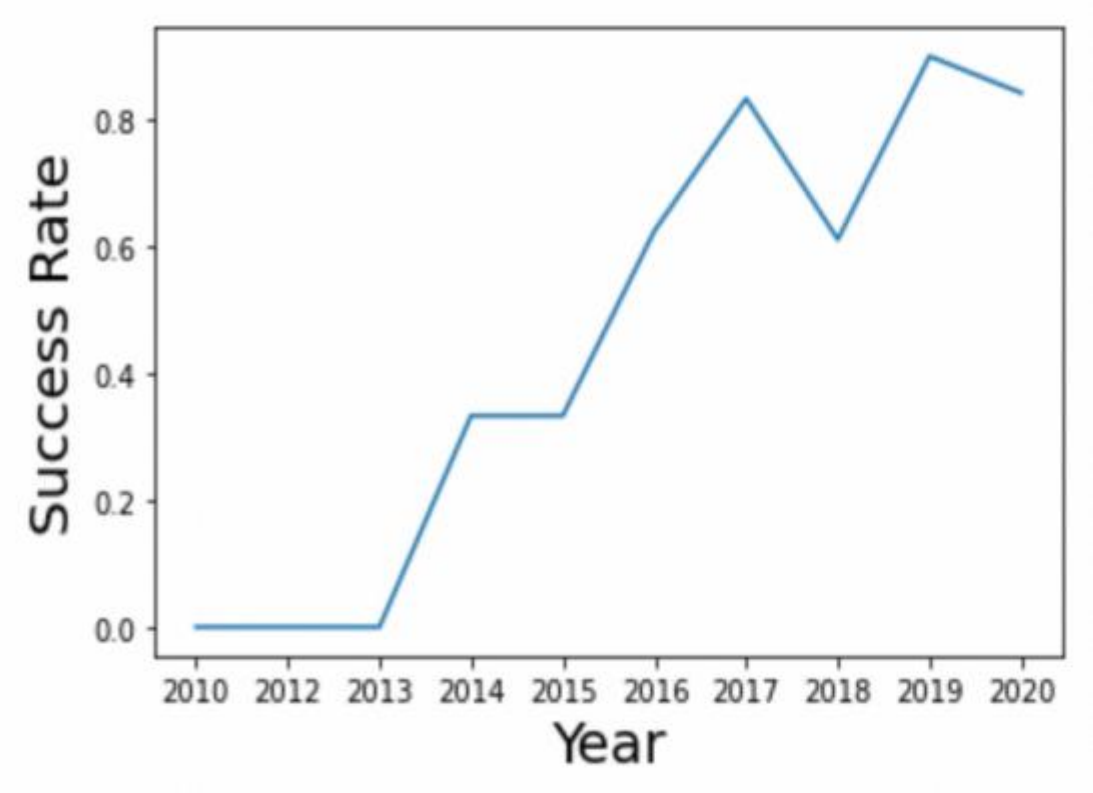
Explanation:

- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits

Launch Success Yearly Trend

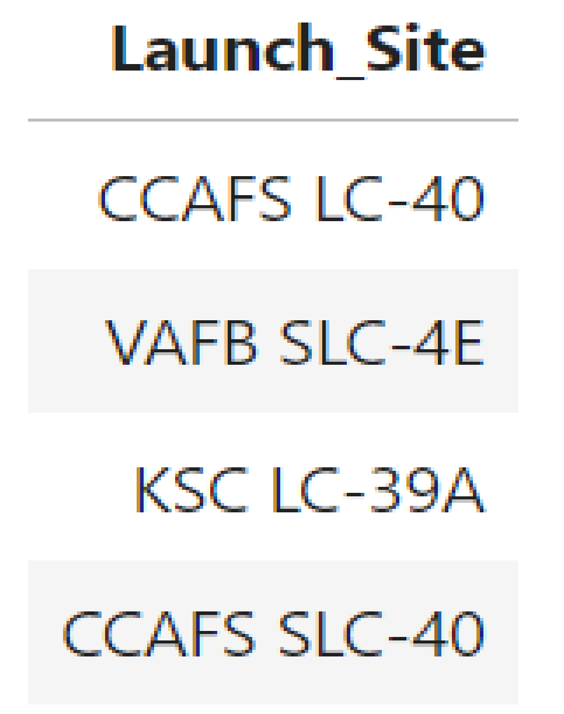
Explanation:

- The success rate since 2013 kept increasing till 2020.



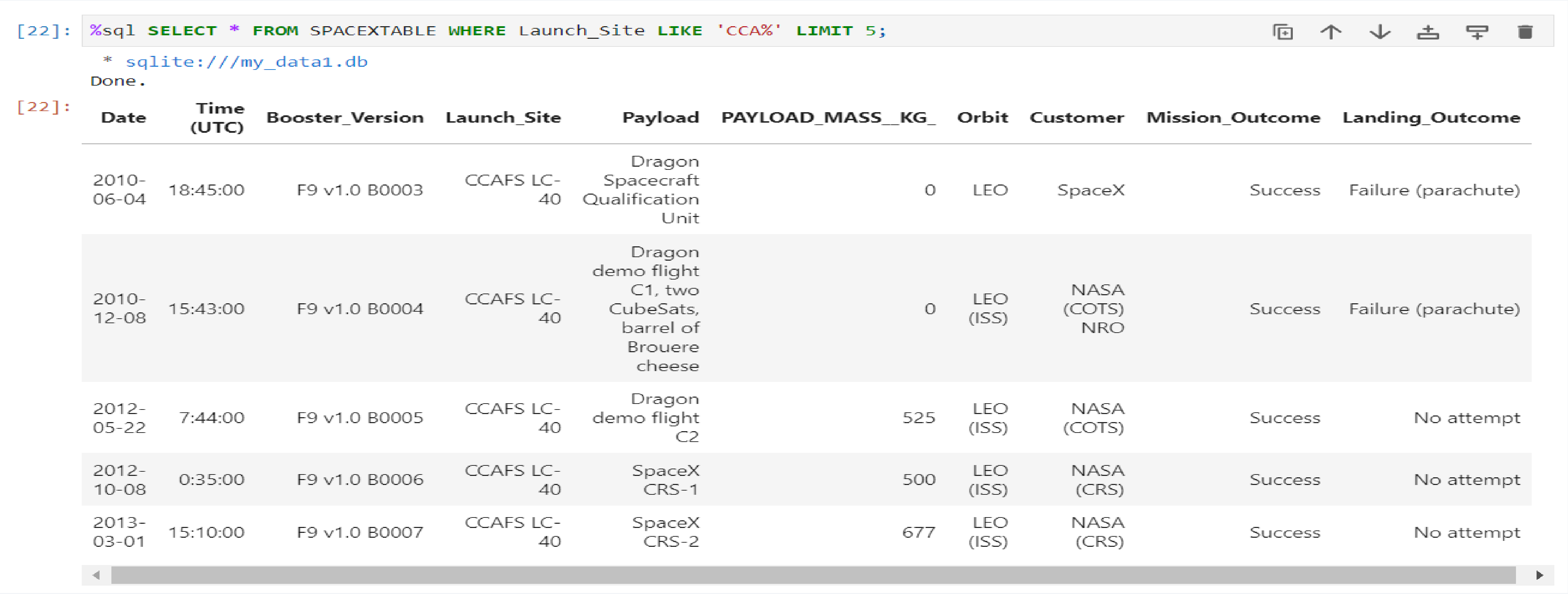
All Launch Site Names

Explanation: Displaying the names of the unique launch sites in the space mission.



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

Launch Site Names Begin with 'CCA'



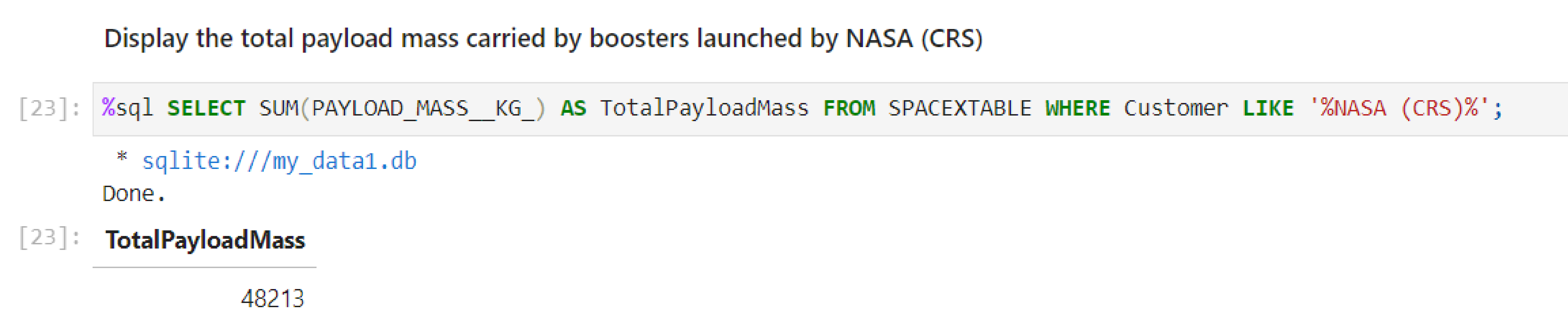
```
[22]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

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

[22]:

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

Explanation: Displaying 5 records where launch sites begin with the string 'CCA'

Total Payload Mass



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

```
[23]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTABLE WHERE Customer LIKE '%NASA (CRS)%';
```

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

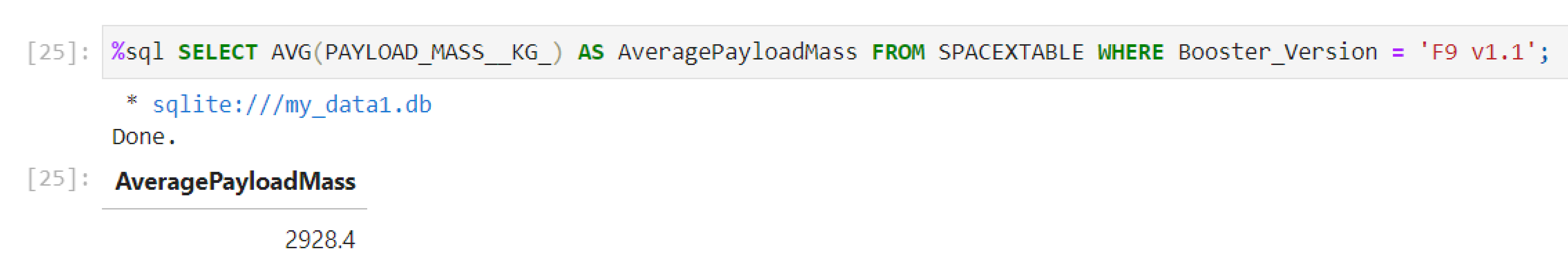
[23]:

TotalPayloadMass
48213

Explanation: Displaying the total payload mass carried by boosters launched by NASA (CRS).

Average Payload Mass by F9 v1.1

Explanation: Displaying average payload mass carried by booster version F9 v1.1.



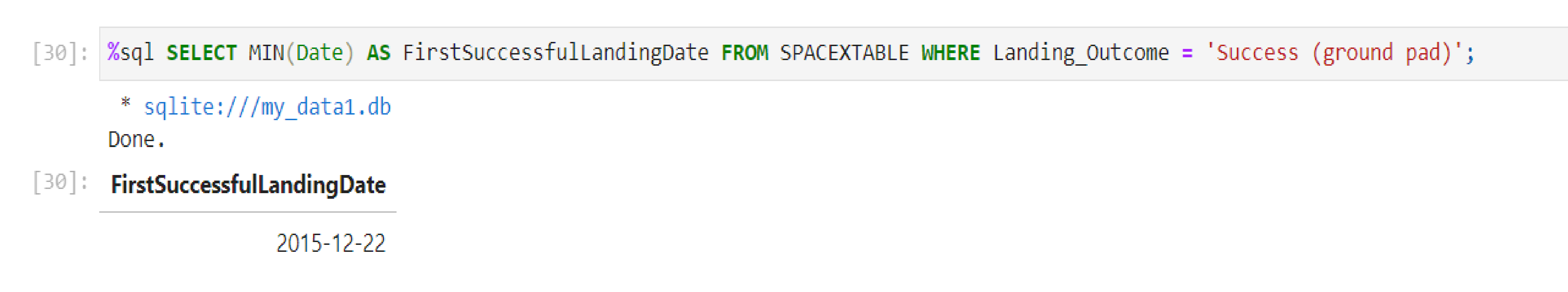
```
[25]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';

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

[25]:

AveragePayloadMass
2928.4

First Successful Ground Landing Date

Explanation: Listing the date when the first successful landing outcome in ground pad was achieved



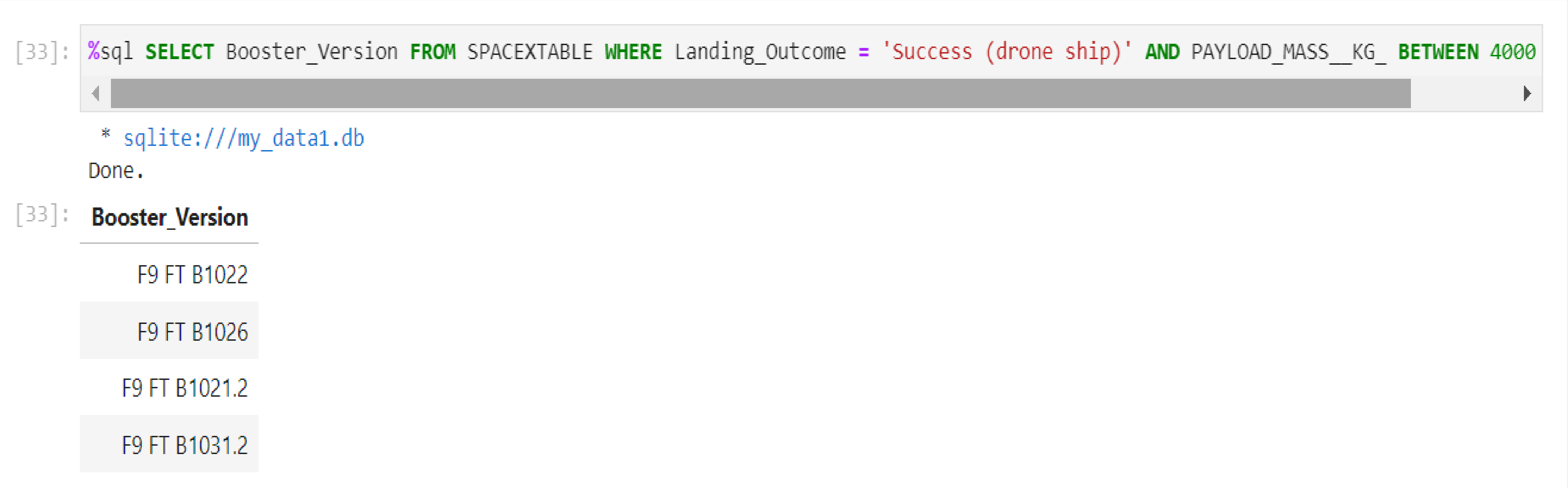
```
[30]: %sql SELECT MIN(Date) AS FirstSuccessfulLandingDate FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';

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

[30]:

FirstSuccessfulLandingDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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



```
[33]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000
```

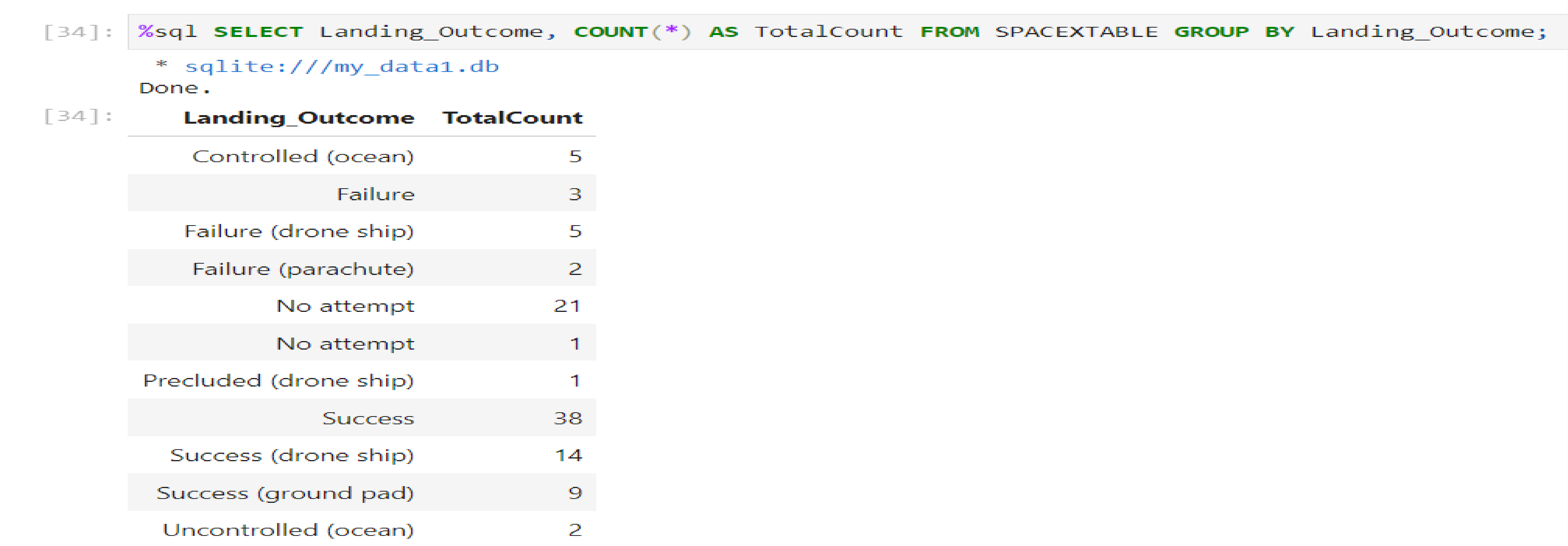
```
 * sqlite:///my_data1.db
Done.
```

[33]:

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

Total Number of Successful and Failure Mission Outcomes

Explanation: Listing the total number of successful and failure mission outcomes



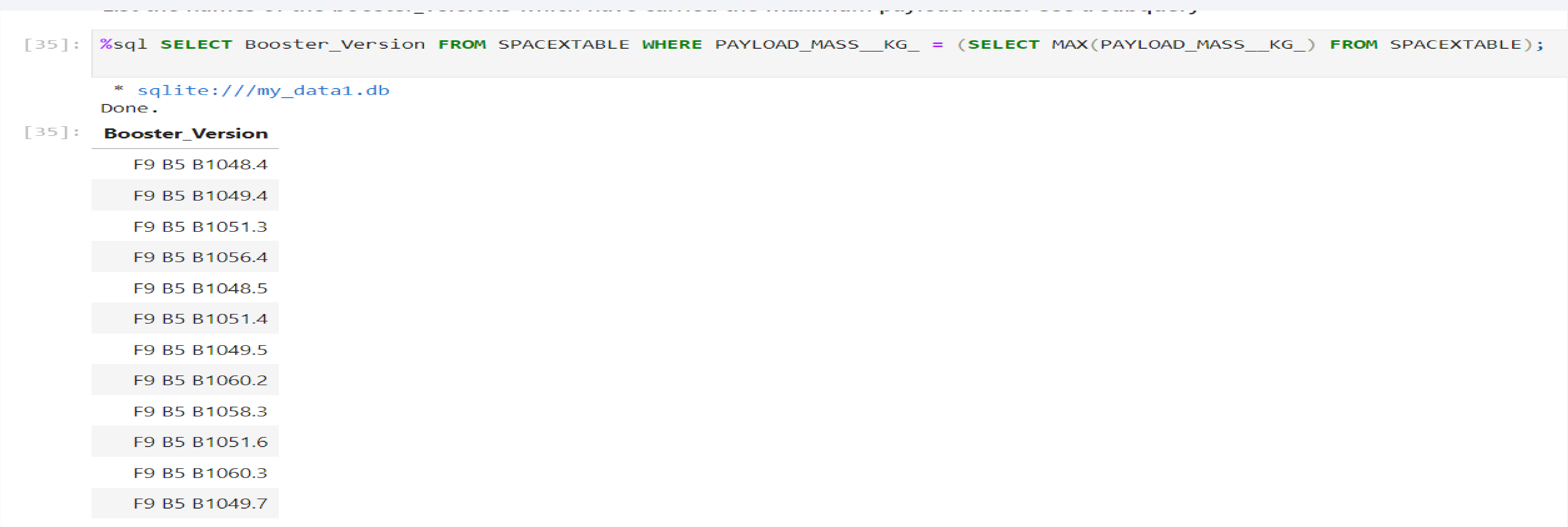
```
[34]: %sql SELECT Landing_Outcome, COUNT(*) AS TotalCount FROM SPACEXTABLE GROUP BY Landing_Outcome;
       * sqlite:///my_data1.db
      Done.
```

[34]:

Landing_Outcome	TotalCount
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

Explanation: Listing the names of the booster versions which have carried the maximum payload mass



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

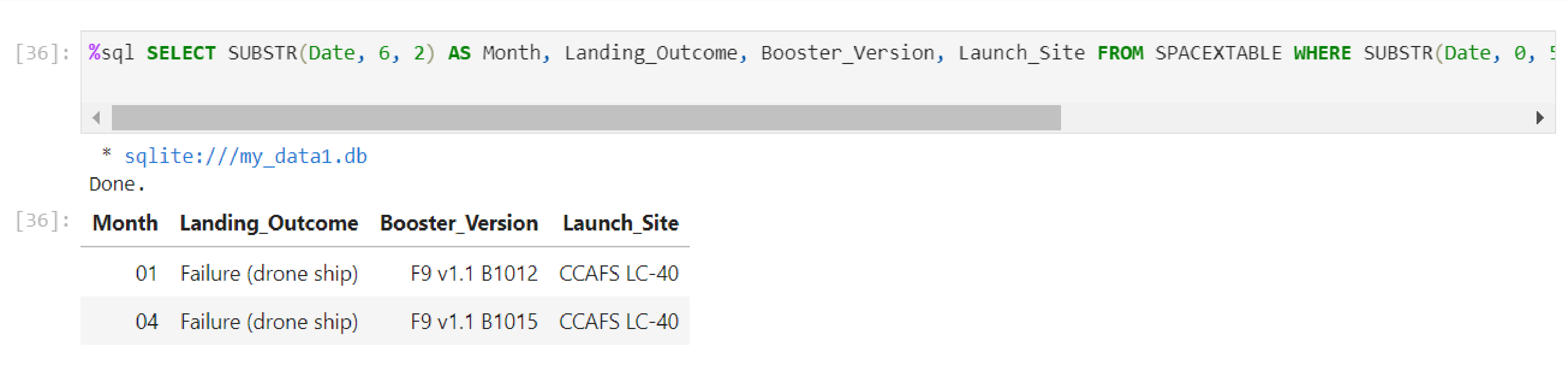
```
[35]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

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

[35]:

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

Explanation: Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015



```
[36]: %sql SELECT SUBSTR(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE SUBSTR(Date, 0, 5
```

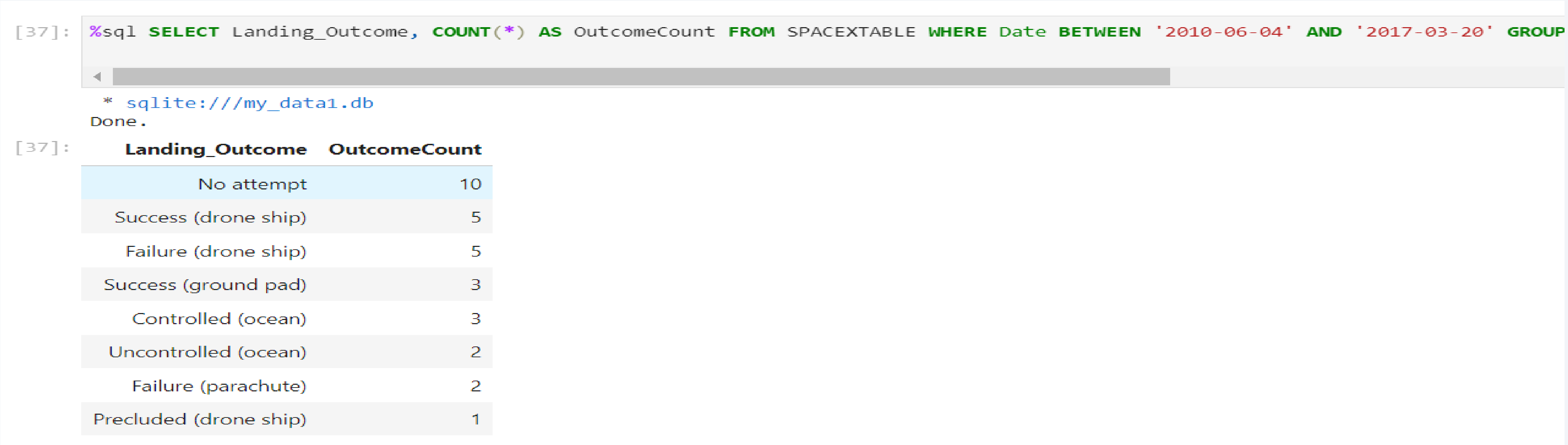
```
 * sqlite:///my_data1.db
Done.
```

[36]:

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

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

Explanation: • Ranking 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.



```
[37]: %sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP
```

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

[37]:

Landing_Outcome	OutcomeCount
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

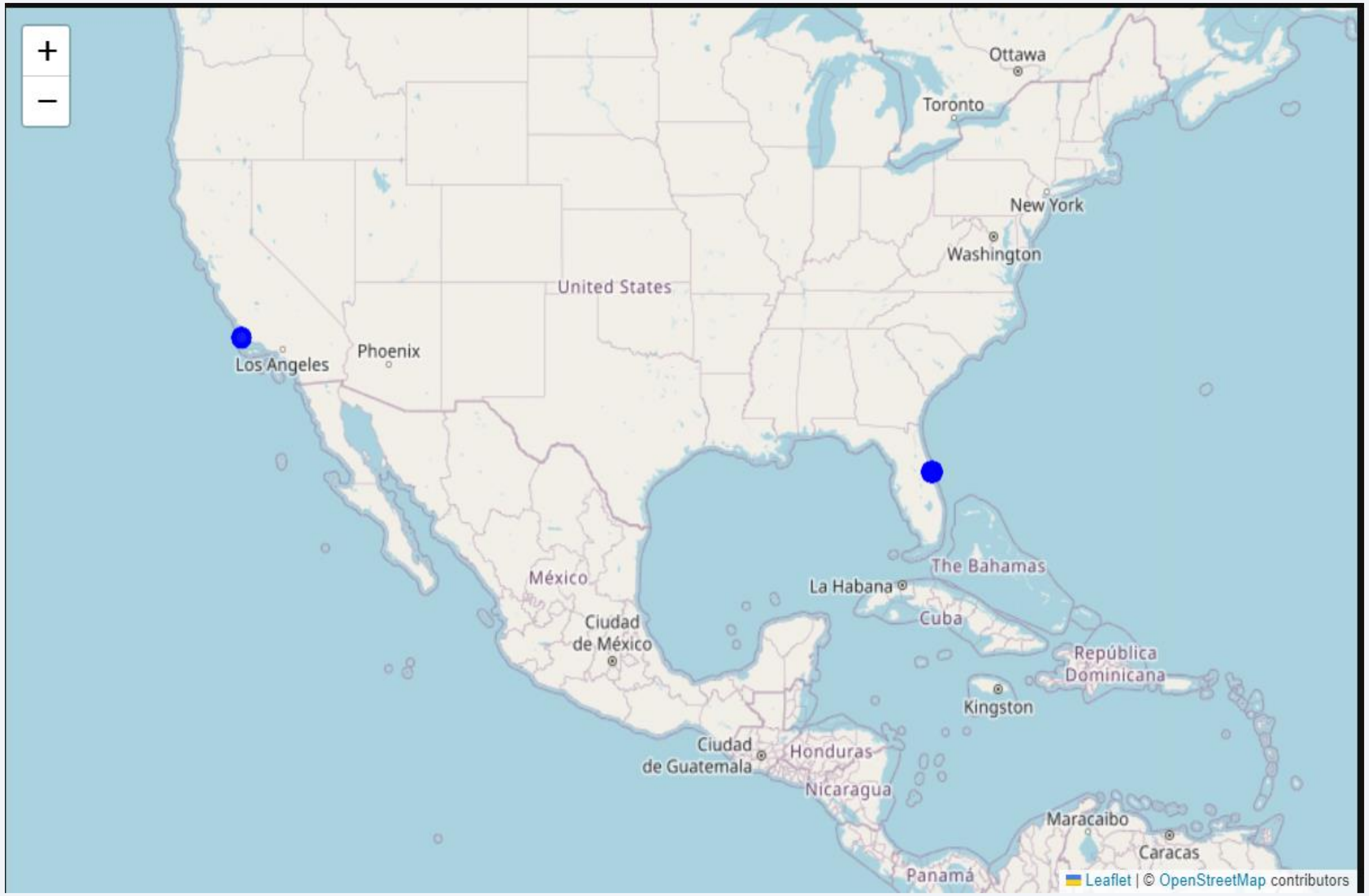
Section 3

Launch Sites Proximities Analysis

All launch sites' location markers on a global map

Explanation:

- Most of Launch sites are in proximity to the Equator line. The land is moving faster at the equator than any other place on the surface of the Earth. Anything on the surface of the Earth at the equator is already moving at 1670 km/hour. If a ship is launched from the equator it goes up into space, and it is also moving around the Earth at the same speed it was moving before launching. This is because of inertia. This speed will help the spacecraft keep up a good enough speed to stay in orbit.
- All launch sites are in very close proximity to the coast, while launching rockets towards the ocean it minimizes the risk of having any debris dropping or exploding near people



Color-labeled launch records on the map

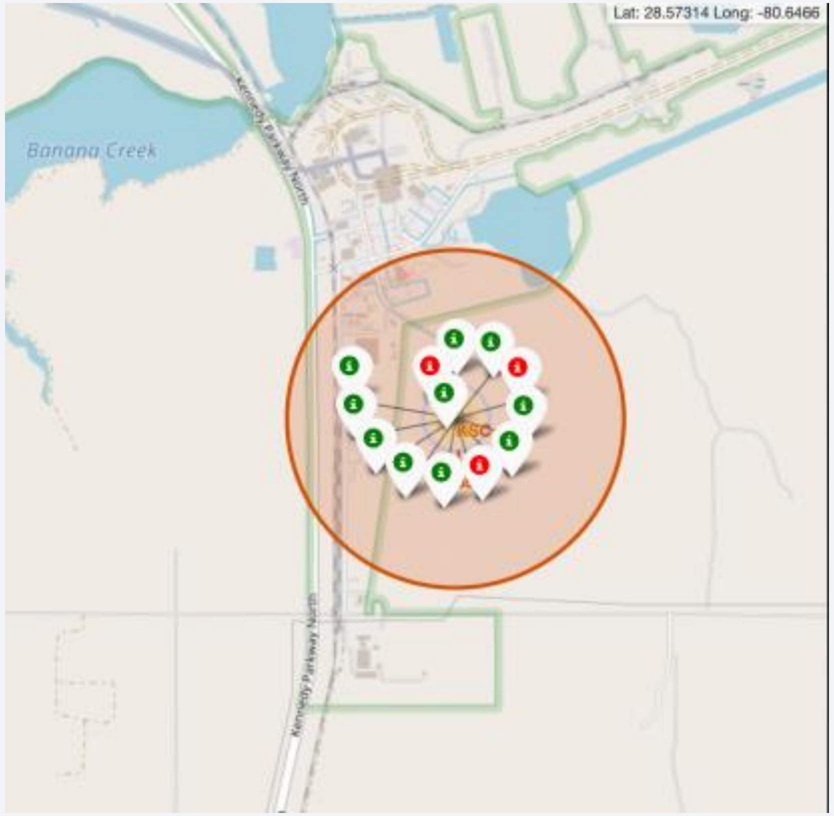
Explanation:

- From the color-labeled markers we should be able to easily identify which launch sites have relatively high success rates.

 - Green Marker = Successful Launch

- Red Marker = Failed Launch

- Launch Site KSC LC-39A has a very high Success Rate



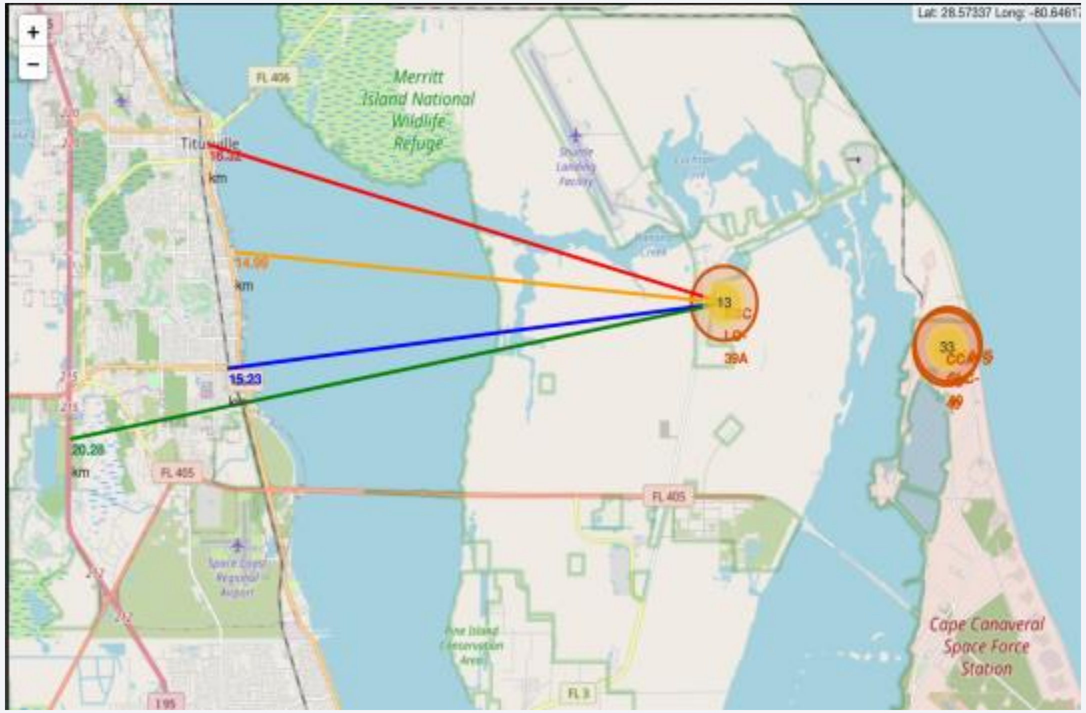
Distance from the launch site KSC LC-39A to its proximities

Explanation:

- From the visual analysis of the launch site KSC LC-39A we can clearly see that it is: - relative close to railway (15.23 km) - relative close to highway (20.28 km) - relative close to coastline (14.99 km)

- Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km).

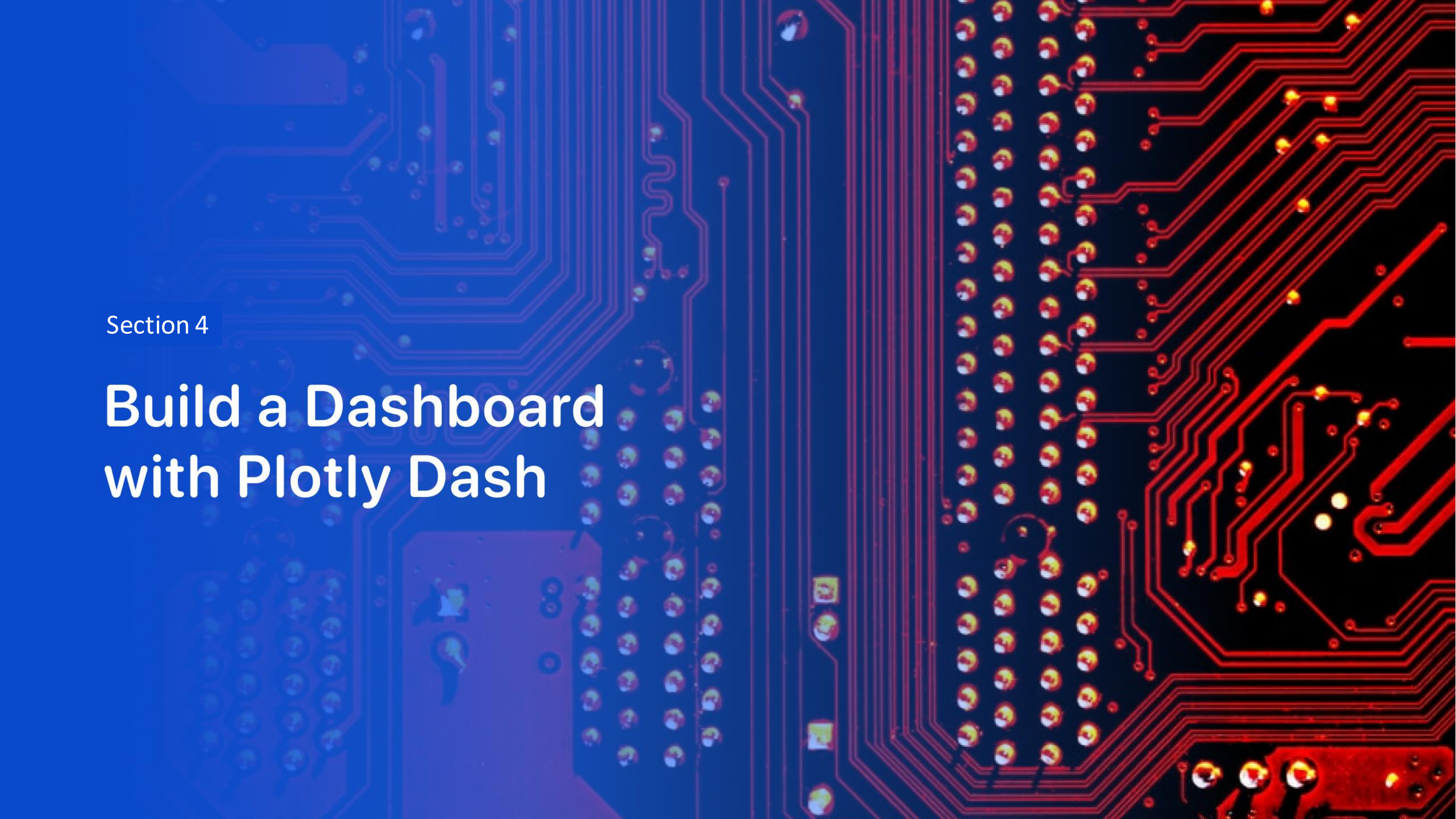
- Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas



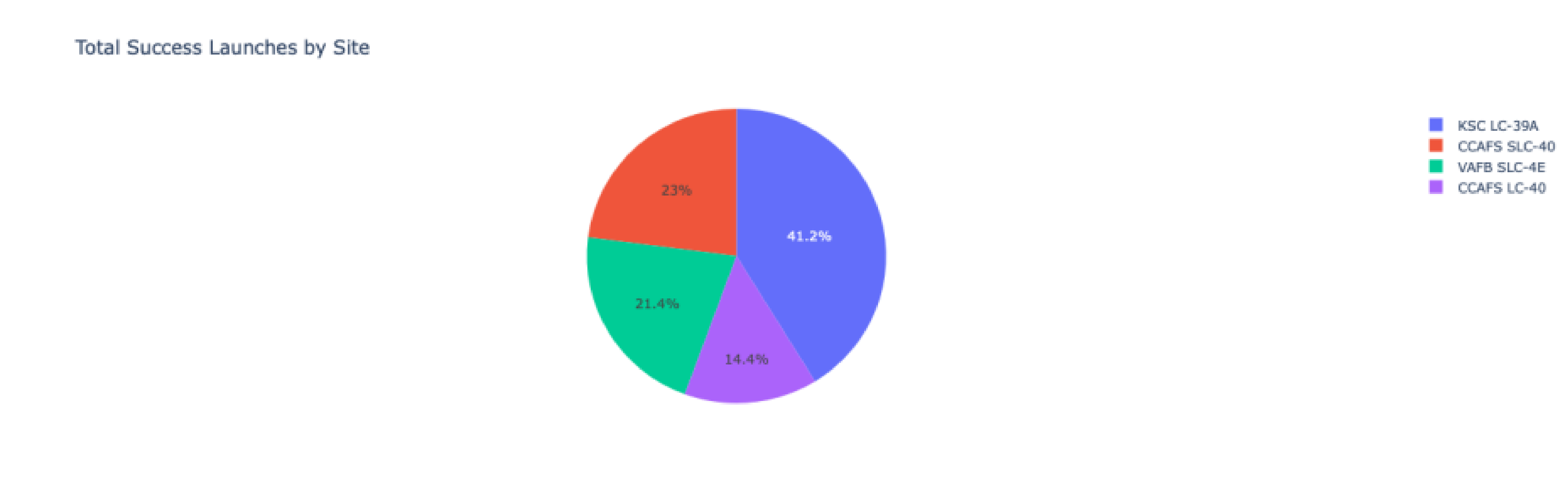
Lat: 28.57337 Long: -80.6461
+
−
FL 406
Merritt Island National Wildlife Refuge
Titusville
16.32 km
14.99 km
15.23 km
20.28 km
13
LC-39A
33
CCAFS SLC-40
FL 405
FL 405
FL 3
I 95
Cape Canaveral Space Force Station

Section 4

Build a Dashboard with Plotly Dash

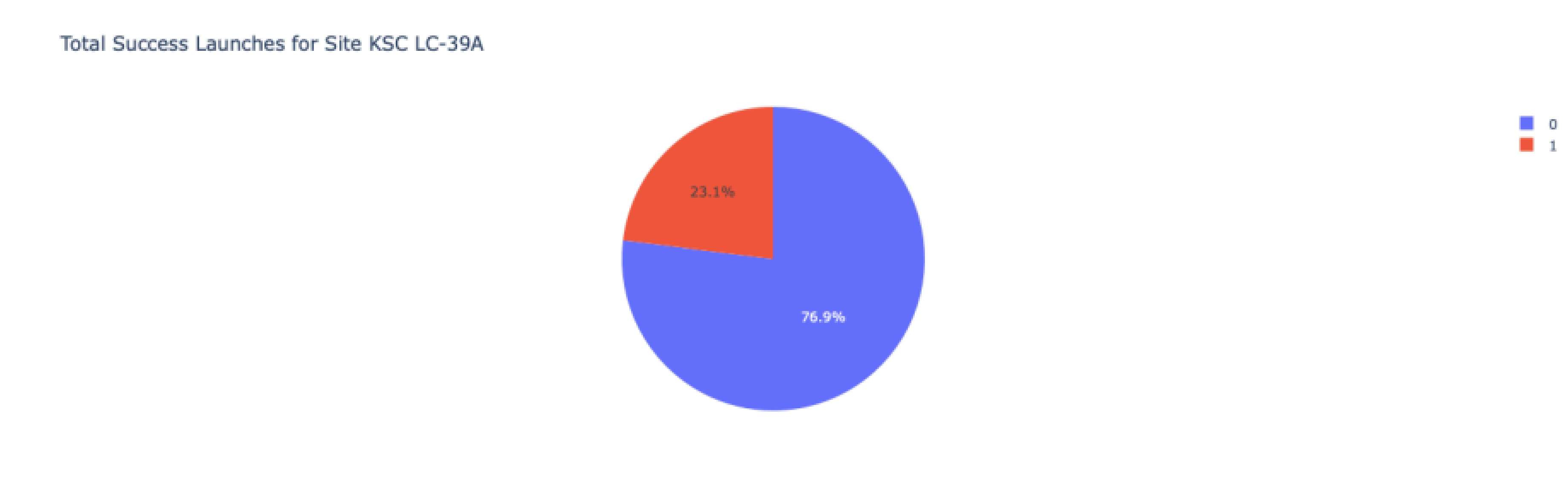


Launch success count for all sites



Explanation: The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches

Launch site with highest launch success ratio

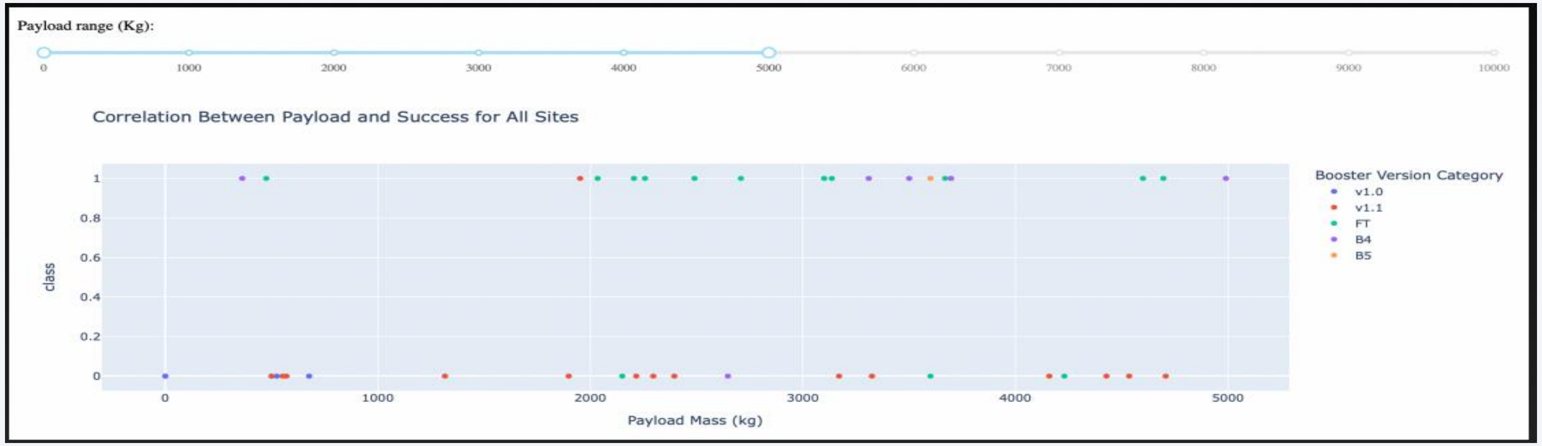


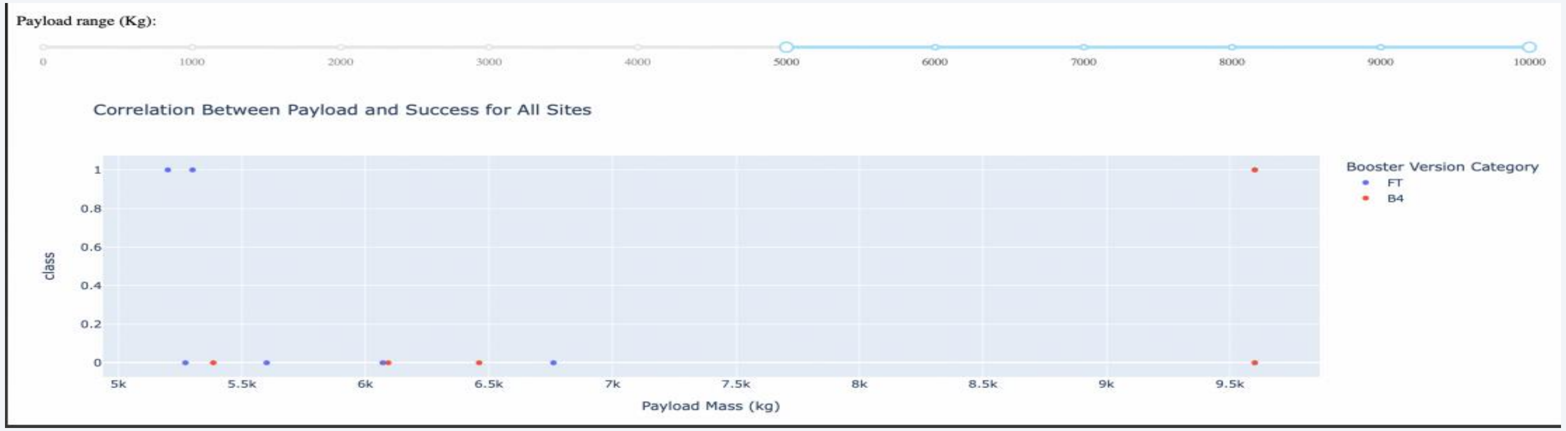
Explanation: KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings

Payload Mass vs. Launch Outcome for all sites

Explanation:

The charts show that payloads between 2000 and 5500 kg have the highest success rate





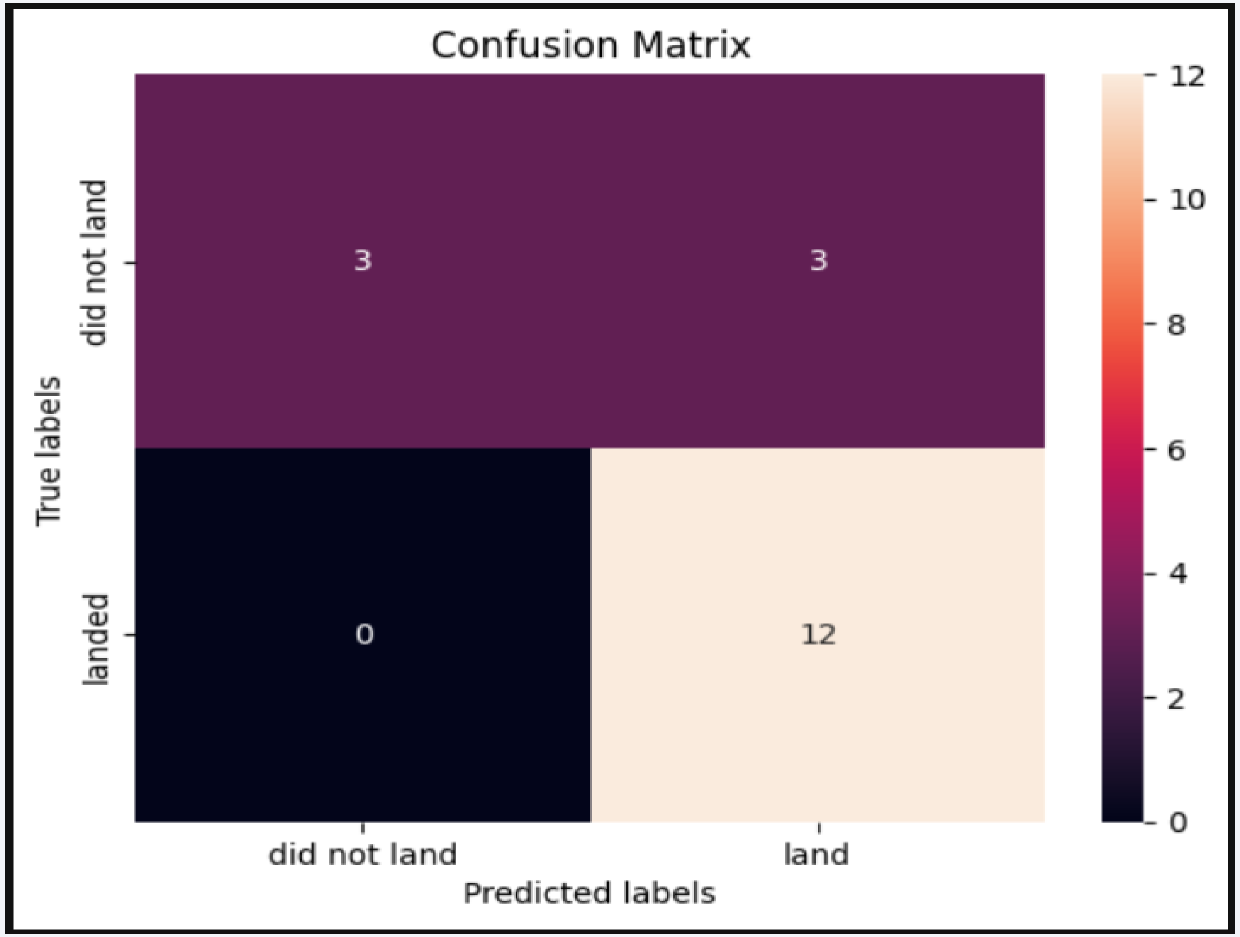
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Explanation:

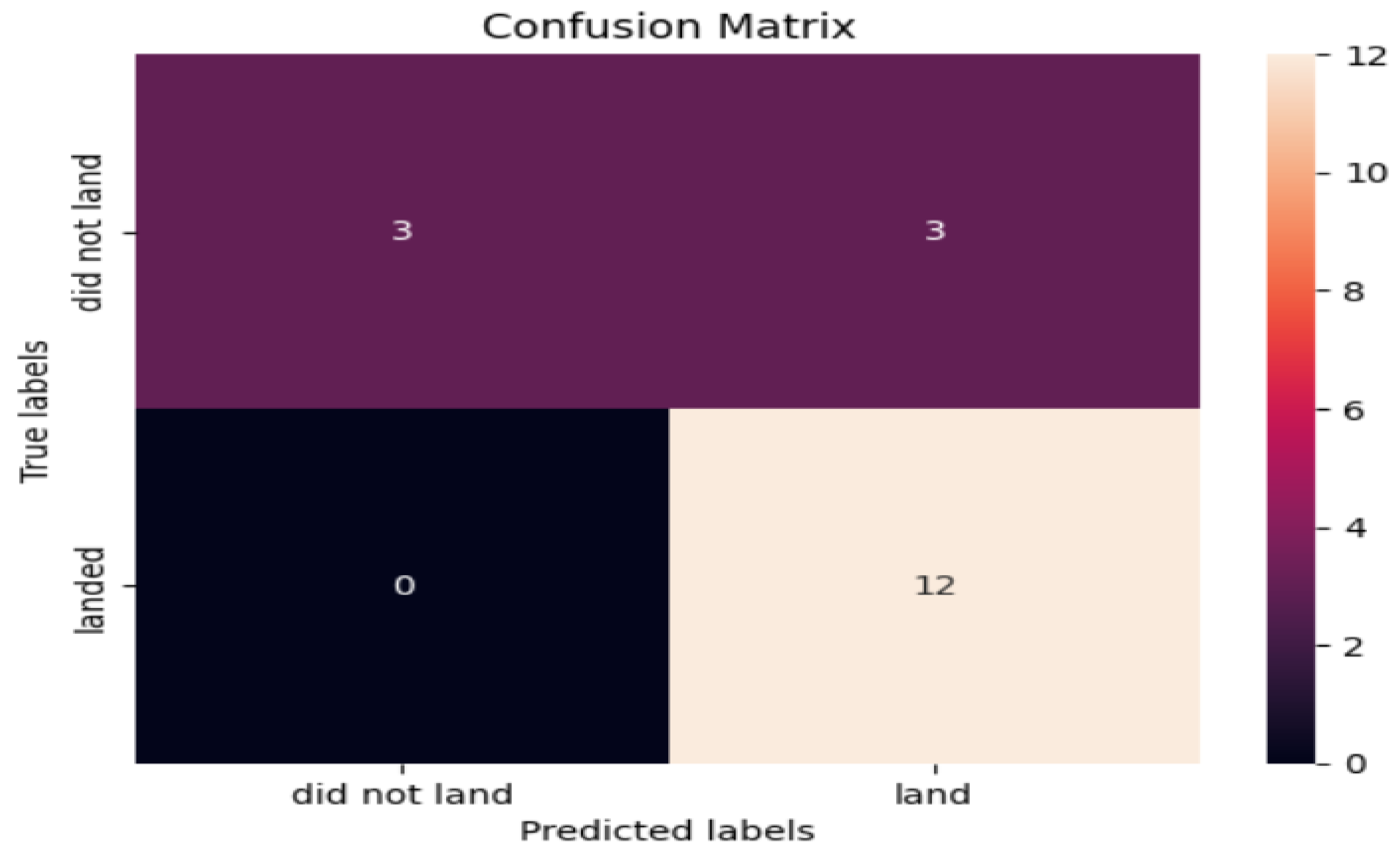
- Based on the scores of the Test Set, we can not confirm which method performs best.
- Same Test Set scores may be due to the small test sample size (18 samples). Therefore, we tested all methods based on the whole Dataset.
- The scores of the whole Dataset confirm that the best model is the Decision Tree Model. This model has not only higher scores, but also the highest accuracy



Confusion Matrix

Explanation:

- Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives



Conclusions

Decision Tree Model is the best algorithm for this dataset.

- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!