IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

By Motlatsi Moea
Date: 6/7/2024

Outline

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

Executive Summary

The aim of the project was to investigate if predictive models could be developed to predict if SpaceX's launches would succeed or fail, and thus determine the cost of each mission.

To achieve this goal, data about SpaceX's past missions was obtained from SpaceX REST API as well as web scraping the wiki pages.

The data was then processed and cleaned using Python Pandas library, analyzed using SQL, visualized using seaborn and Matplotlib and Folium libraries, as well as creating a dashboard using Plotly and Dash.

Furthermore, the data was used to train a predictive classification algorithm.

What was found was that it would be possible to predict the success or failure of future SpaceX missions using the classification algorithm, thus predicting future costs. However, it is advised that more data be used to train the models for better accuracy

It was also found that SpaceX's mission successes have been steadily increasing. Meaning, it can be expected to remain competitive for the foreseeable future.

Introduction

We live in an age of commercialized space travel with companies such:

- Virgin Galactic
- Blue origin

Top of the food chain is SpaceX because of its ability to reuse the first stage of the Rocket, making the whole process much cheaper.

- Problems you want to find answers
  - Determine the price of each launch Space X will conduct
  - Predict whether the first stage will be reused or not for each launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected from the SpaceX REST API as well as scrapping SpaceX wiki Pages
- Perform data wrangling
    - Python Pandas library was used to clean, process and analyze the data as required
- 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 scikit-learn library was used to build and test out different classification predictive models and the accuracy of the various models compared.

Data Collection

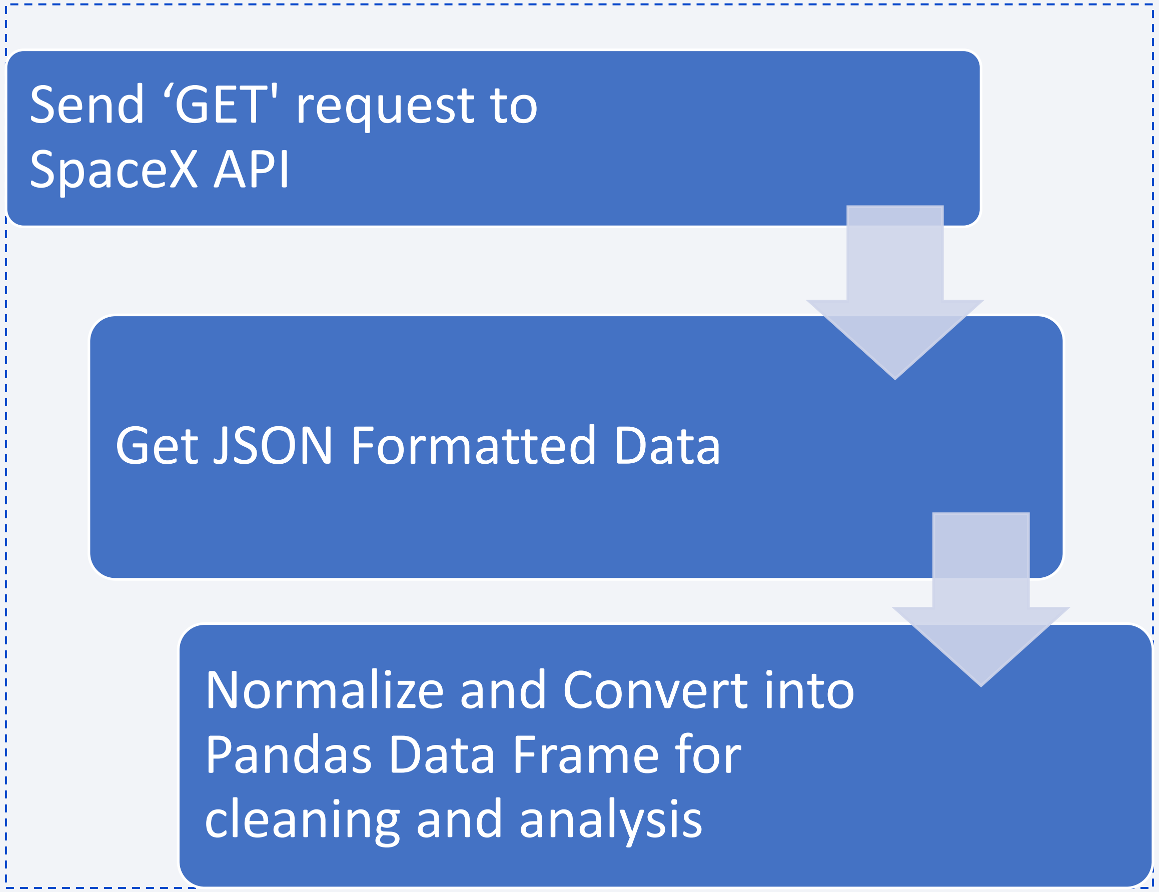
The data for this project was collected using a SpaceX REST API. This provided data about the rocket launches, including:

- Information about rocked used;
- Payload carried;
- Launch & landing specifications
- Landing outcome

Another method that can be used to collect this data is by web scraping some related wiki pages using beautiful soup for Falcon 9 launch data.

Data Collection – SpaceX API

https://github.com/motlatsimoea/final_exam/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



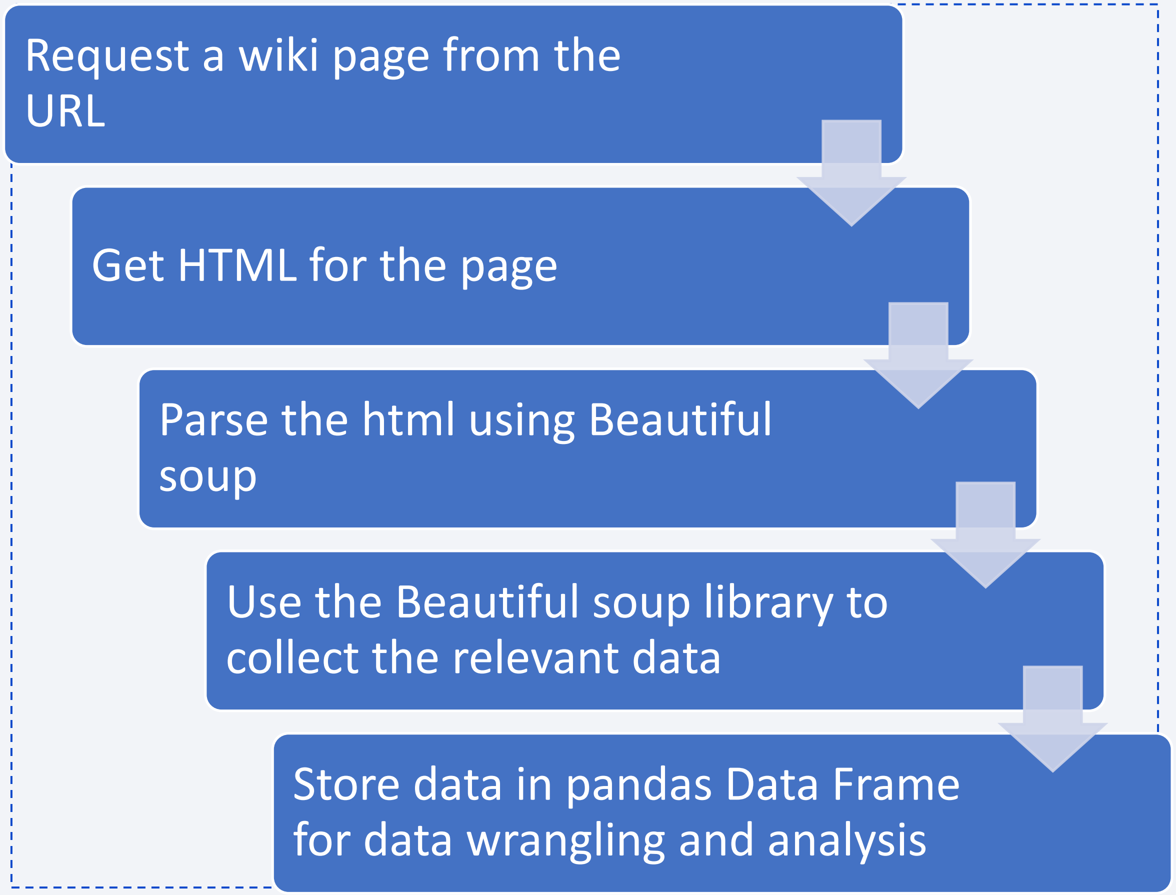
Send ‘GET' request to SpaceX API

Get JSON Formatted Data

Normalize and Convert into Pandas Data Frame for cleaning and analysis

Data Collection - Scraping

https://github.com/motlatsimoea/final_exam/blob/main/jupyter-labs-webscraping.ipynb



Request a wiki page from the URL

Get HTML for the page

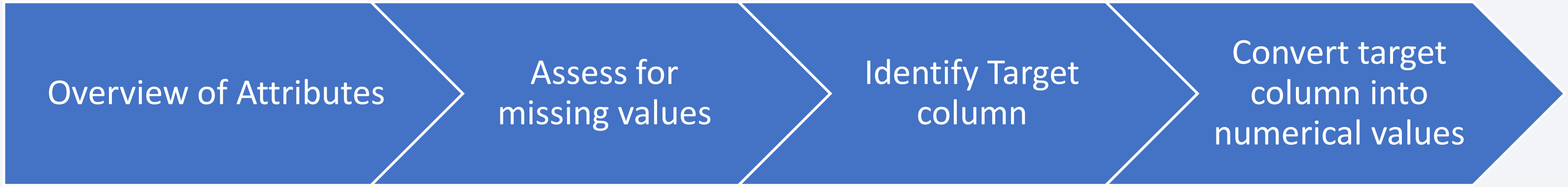
Parse the html using Beautiful soup

Use the Beautiful soup library to collect the relevant data

Store data in pandas Data Frame for data wrangling and analysis

Data Wrangling

- An overview of the data was conducted, in order to get a better understanding of the various attributes of the data, and to identify the target column, which was the 'Outcome'.
- This Target was categorical with multiple categories indicating success and failure outcomes. These needed to be converted into numerical values. (0 for failed and 1 for success.)


Overview of Attributes

Assess for missing values

Identify Target column

Convert target column into numerical values

- https://github.com/motlatsimoea/final_exam/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- The different variables were visualized using scatter plots. The goal was to understand the relationships between the different variable and how they impacted the target column(Outcome).
- The variables were plotted and separated by the 'Outcome' category(0:failure, 1: success)
- Bar plot was also used for the success rate for each Orbit Type

- https://github.com/motlatsimoea/final_exam/blob/main/edadataviz.ipynb

EDA with SQL

With the data stored in a database, it was explored using SQL. The goal was to obtain information that included:

- Different Launch sites used;
- The total payload carried for all missions;
- Average payload carried by Booster version F9 v1.1;
- Total missions that succeeded and failed;
- Sum of each of the landing outcomes;
- Etc.

- https://github.com/motlatsimoea/final_exam/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- A map was created using Folium to visually display the launch sites.
- Markers circles were used to mark the different launch sites on the map;
- Cluster markers were used to show the successful and failed missions for each site;
- The closest highway, city, railway and coastlines were highlighted on the map using lines.
- https://github.com/motlatsimoea/final_exam/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

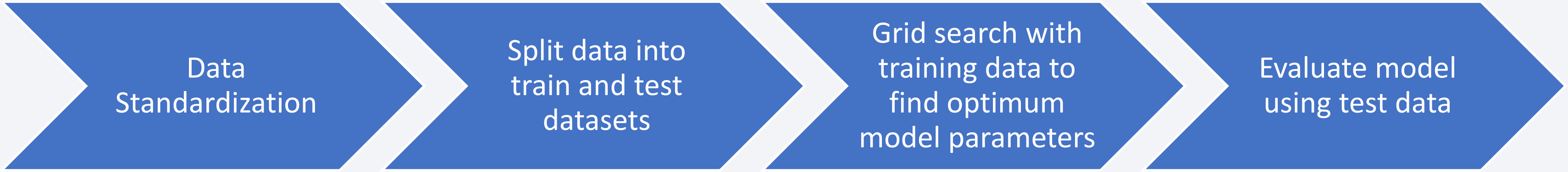
A dashboard was created using dash. This Dashboard included:

- A pie chart giving a summary of the success rate for each launch site;
- There is a dropdown menu to allow the user to see results for a specific site
- A scatter plot that shows the relationship between the payload size and the Outcome.
- A slider was added to allow user to adjust the range of the payload for visualization.

- https://github.com/motlatsimoea/final_exam/tree/main/dash

Predictive Analysis (Classification)

- The data was standardized so the impact of all features is properly assessed.
- Data was then split into test and train data. 20% of the data as for testing
- For the various models tested, Grid search validation was used with various options for parameters to find the optimal ones using training data
- The models were evaluated using test data and a confusion matrix was visualized to assess model accuracy


Data Standardization

Split data into train and test datasets

Grid search with training data to find optimum model parameters

Evaluate model using test data

- https://github.com/motlatsimoea/final_exam/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

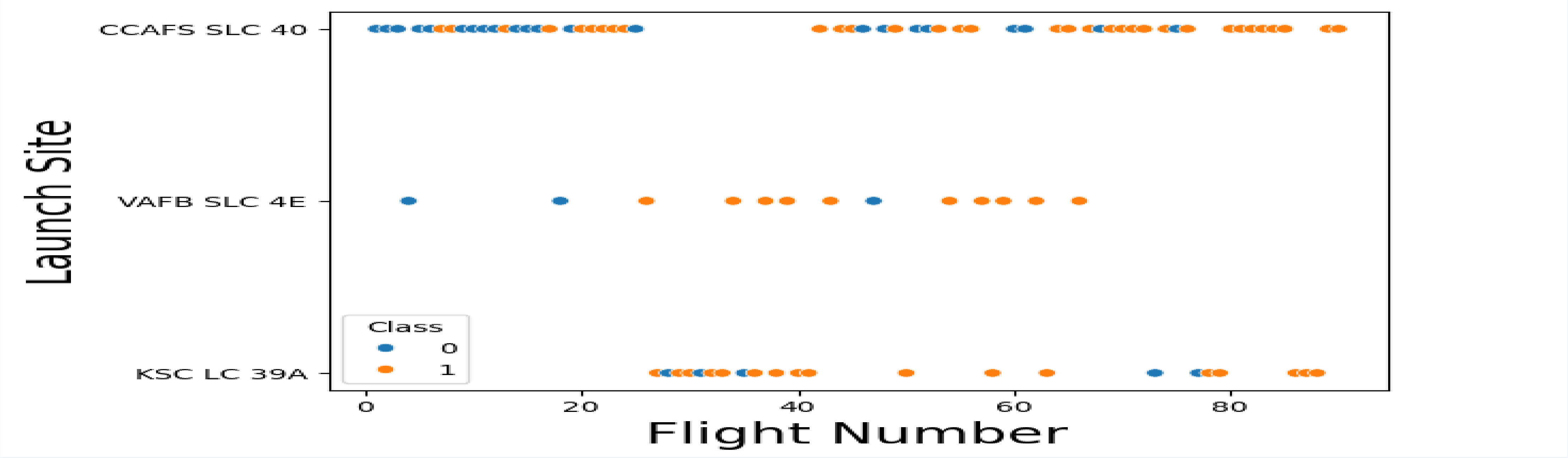
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

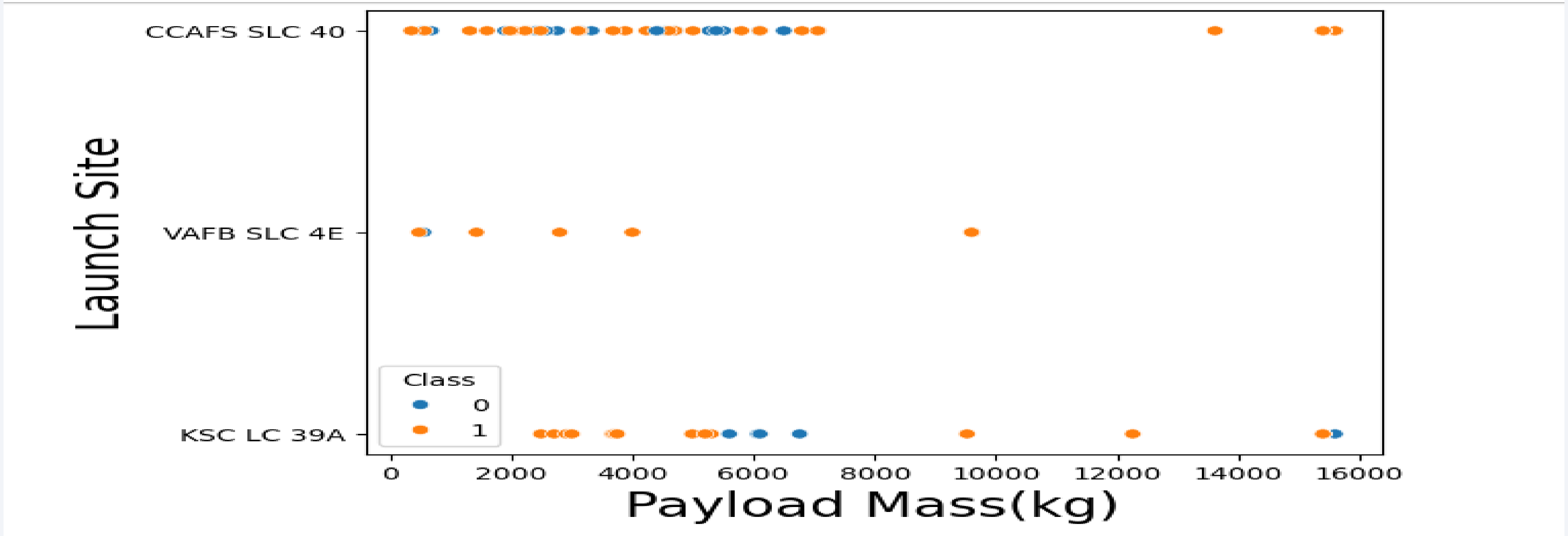
Insights drawn from EDA

Flight Number vs. Launch Site



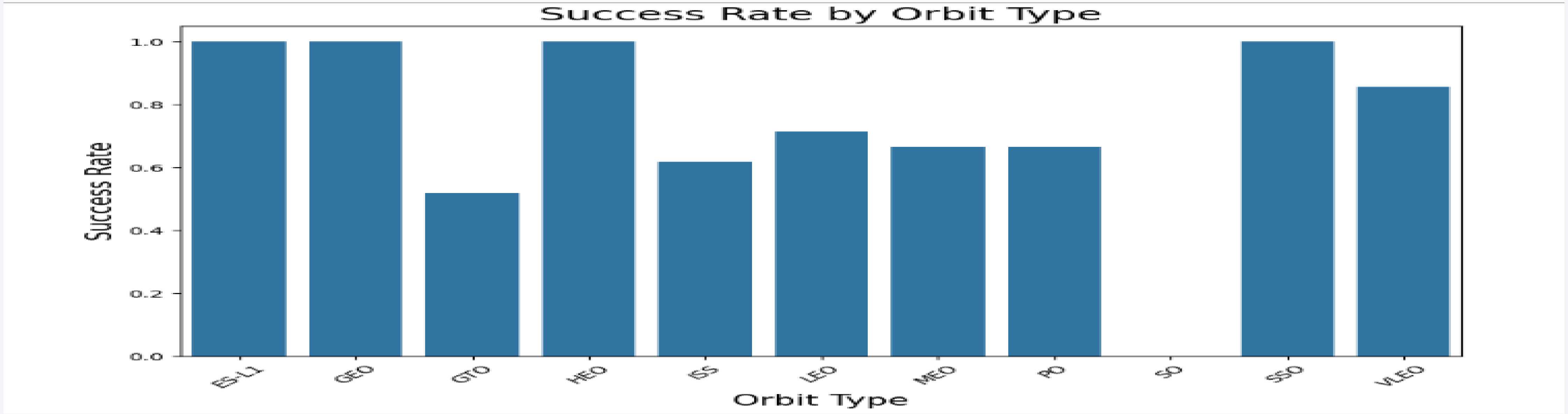
- From the plot, it can be observed that CCAFS SLC 40 is used for most launches;
- VAFB SLC 4E has the least number of launches;
- At least the last 5 launches from all sites were successful. In some cases, more.
- VAFB SLC 4E seems to have the highest success ratio, although also least number of launches

Payload vs. Launch Site



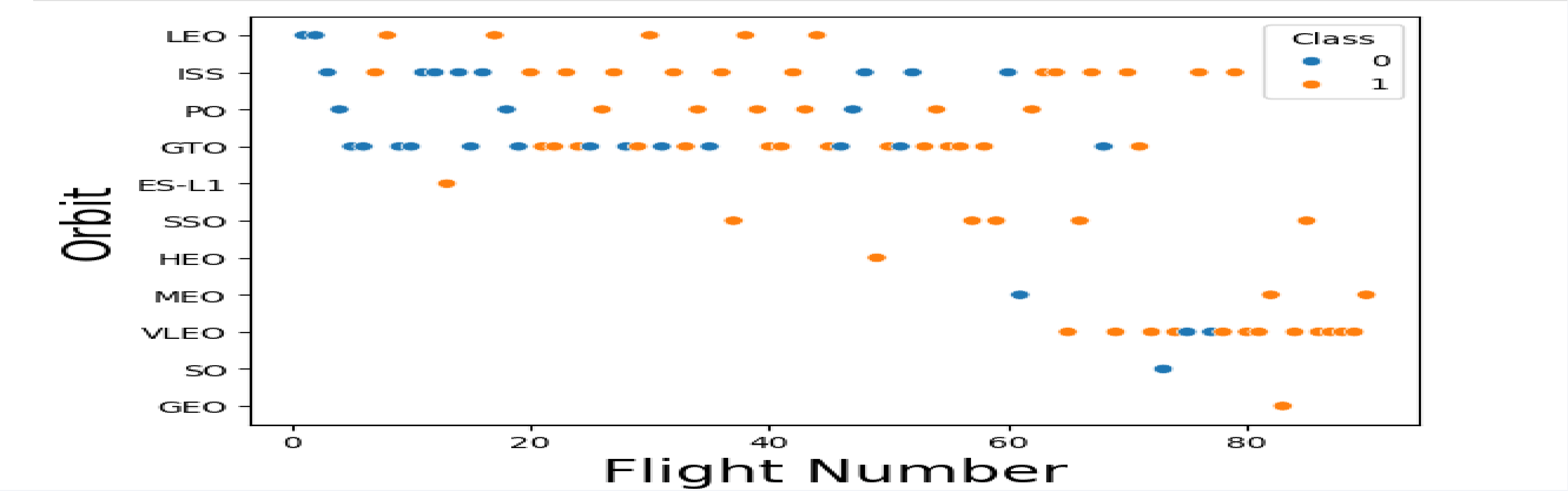
- CCAFS SLC 40 launch site seems to have a mix of failures and successes for payload between 0 and 8000kg. For the larger loads of over 12000kg, the launches have been successful;
- VAFB SLC 4E seems to be used for lighter payload between 0 and 10000kg, resulting in the all but one launch being successful.

Success Rate vs. Orbit Type



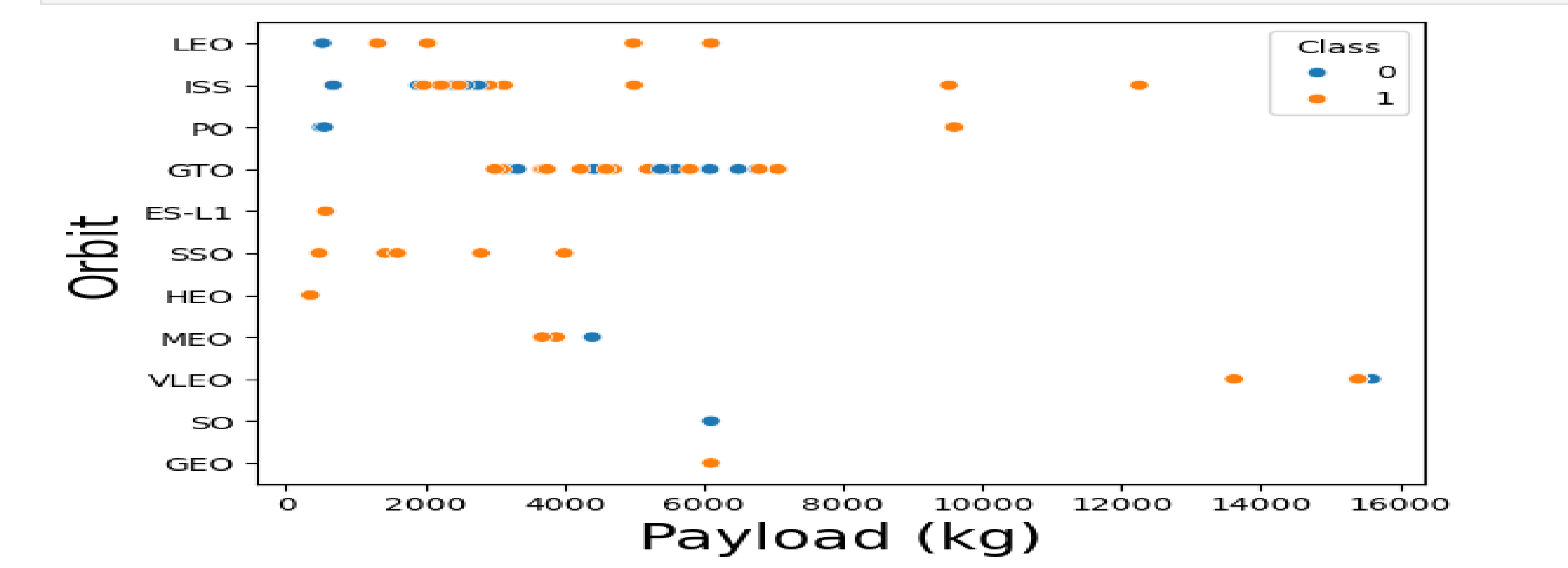
- ES-L1, GEO, HEO, SSO have the highest success rate of 100%
- Followed by LEO with around 70%, MEO and PO just below that and the lowest being GTO at around 50%.

Flight Number vs. Orbit Type



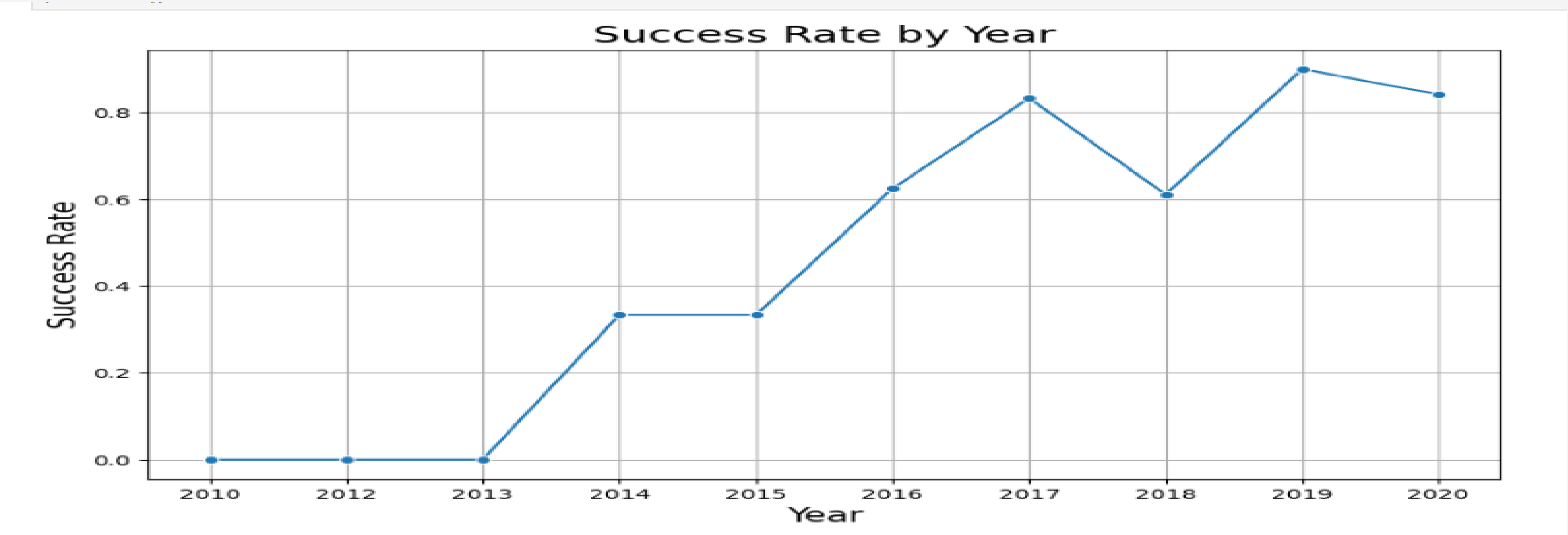
- As the flight number increases, the amount of successful travel to different obits increases;
- ISS and GTO have been visited more often than other obits;
- ES-L1 and GEO only had one flight each. These were successful.

Payload vs. Orbit Type



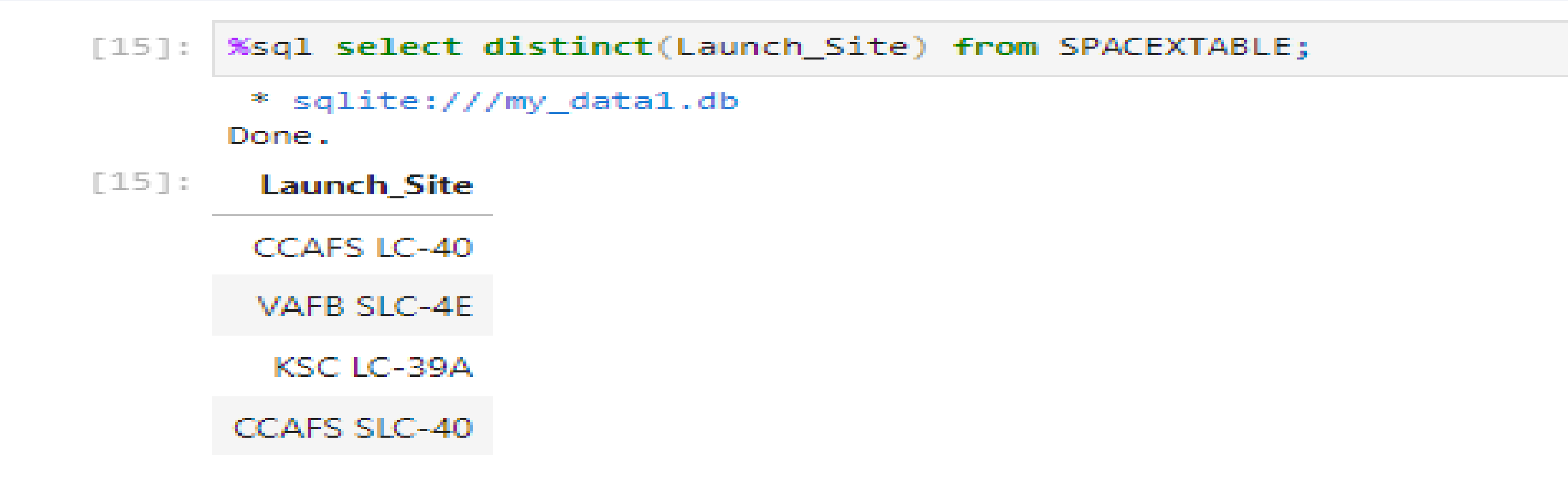
- Payloads between 0 and 4000kg were taken to SSO with all missions being successful;
- GTO had payloads between 3000kg and less than 8000kg and were a mix of failures and success.

Launch Success Yearly Trend



- After the first 3 years, the number success rate of the launches has been rising linearly with a some notable drops in 2018 and 2020.

All Launch Site Names



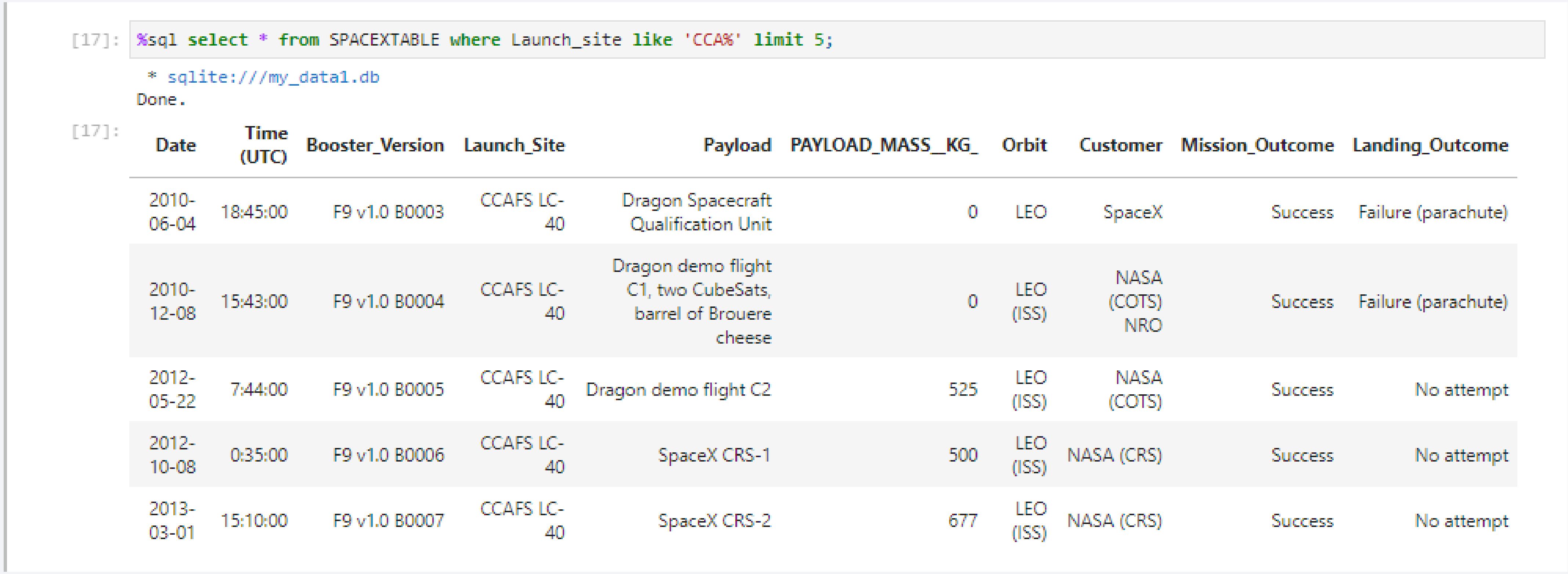
```
[15]: %sql select distinct(Launch_Site) from SPACEXTABLE;

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

[15]:

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

- This was the SQL query that was used to retrieve the unique launch sites.

Launch Site Names Begin with 'CCA'



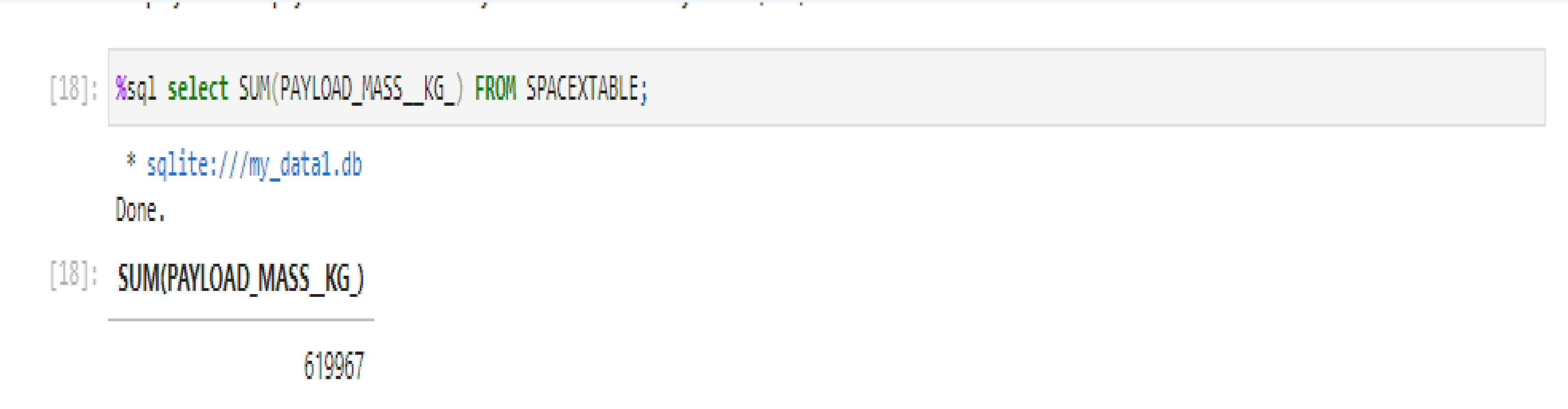
```
[17]: %sql select * from SPACEXTABLE where Launch_site like 'CCA%' limit 5;

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

[17]:

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

- This is the SQL query and the top 5 results that came with it.

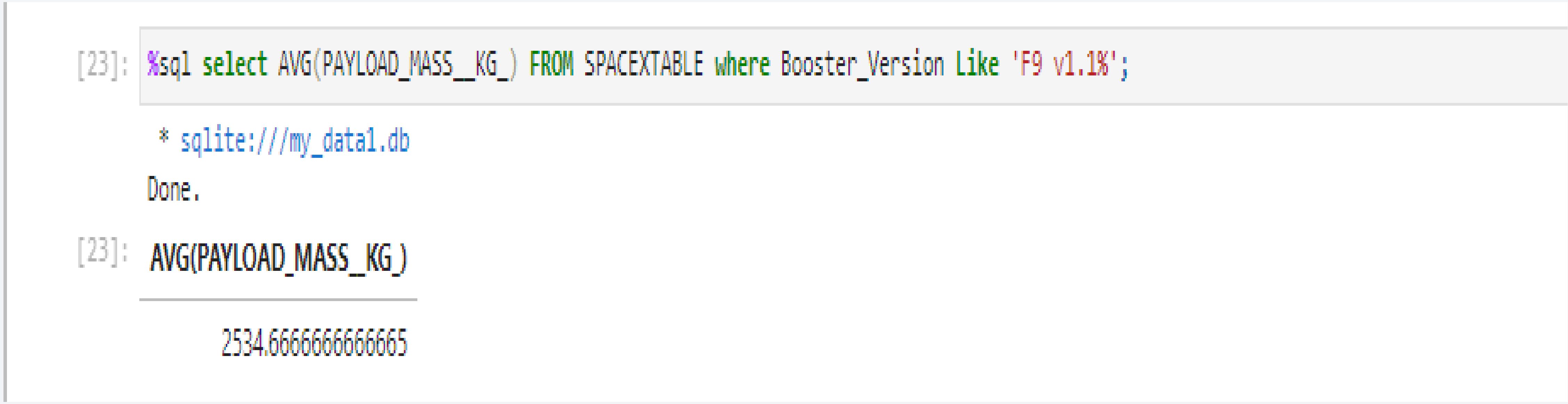
Total Payload Mass



```
[18]: %sql select SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;

 * sqlite:///my_data1.db
Done.

[18]: SUM(PAYLOAD_MASS_KG_)
      ---------------------
                     619967
```

- The query to calculate the total payload mass is shown above with the resulting value.

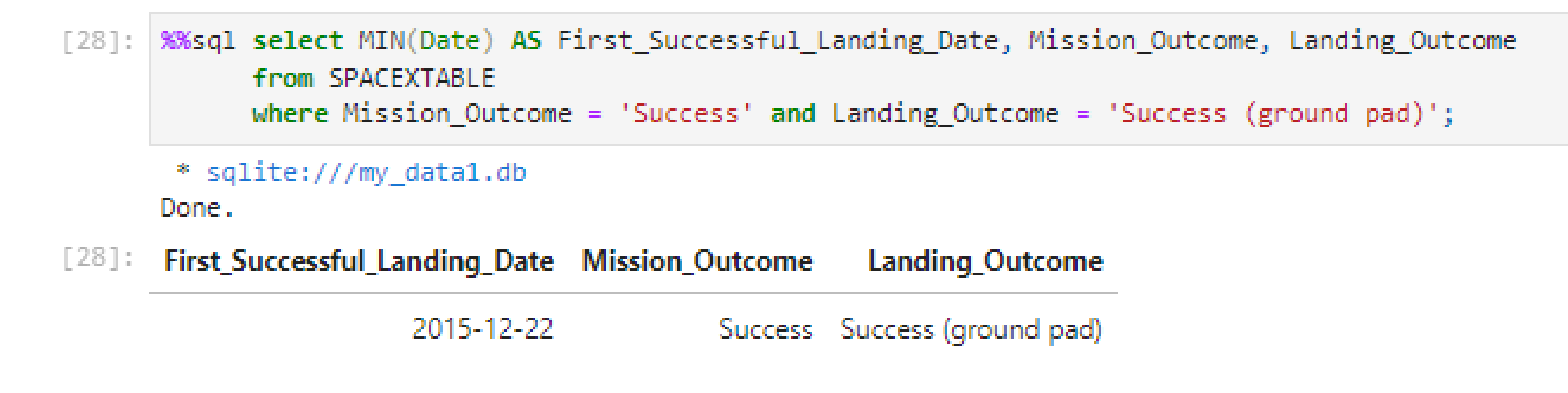
Average Payload Mass by F9 v1.1



```
[23]: %sql select AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Booster_Version Like 'F9 v1.1%';

 * sqlite:///my_data1.db
Done.

[23]: AVG(PAYLOAD_MASS__KG_)
      2534.6666666666665
```

- The average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date



```
[28]: %%sql select MIN(Date) AS First_Successful_Landing_Date, Mission_Outcome, Landing_Outcome
          from SPACEXTABLE
          where Mission_Outcome = 'Success' and Landing_Outcome = 'Success (ground pad)';
```

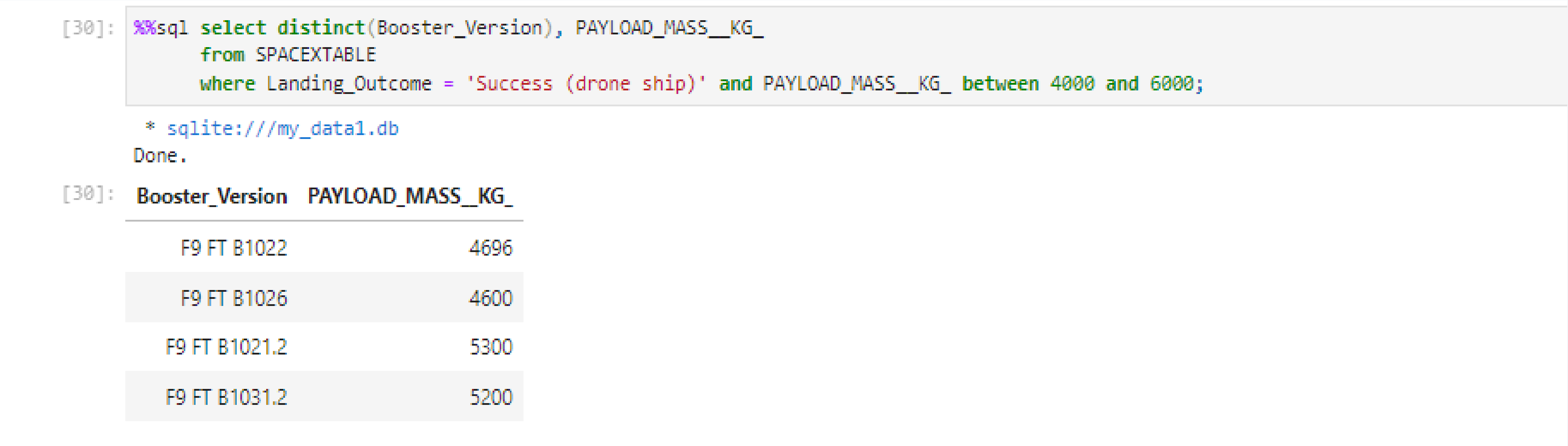
```
 * sqlite:///my_data1.db
Done.
```

[28]:

First_Successful_Landing_Date	Mission_Outcome	Landing_Outcome
2015-12-22	Success	Success (ground pad)

- Query giving the date for the first successful ground landing

Successful Drone Ship Landing with Payload between 4000 and 6000



```
[30]: %%sql select distinct(Booster_Version), PAYLOAD_MASS__KG_
          from SPACEXTABLE
          where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

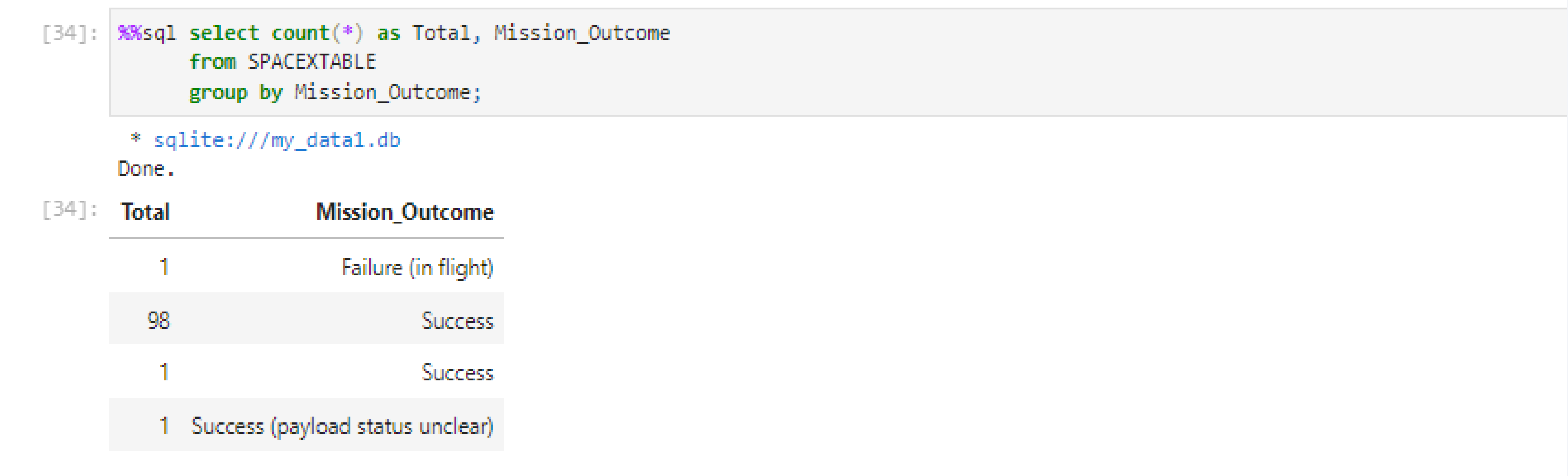
```
 * sqlite:///my_data1.db
Done.
```

[30]:

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

- These are the 4 boosters the landed successfully with the Payload range.

Total Number of Successful and Failure Mission Outcomes



```
[34]: %%sql select count(*) as Total, Mission_Outcome
          from SPACEXTABLE
          group by Mission_Outcome;
```

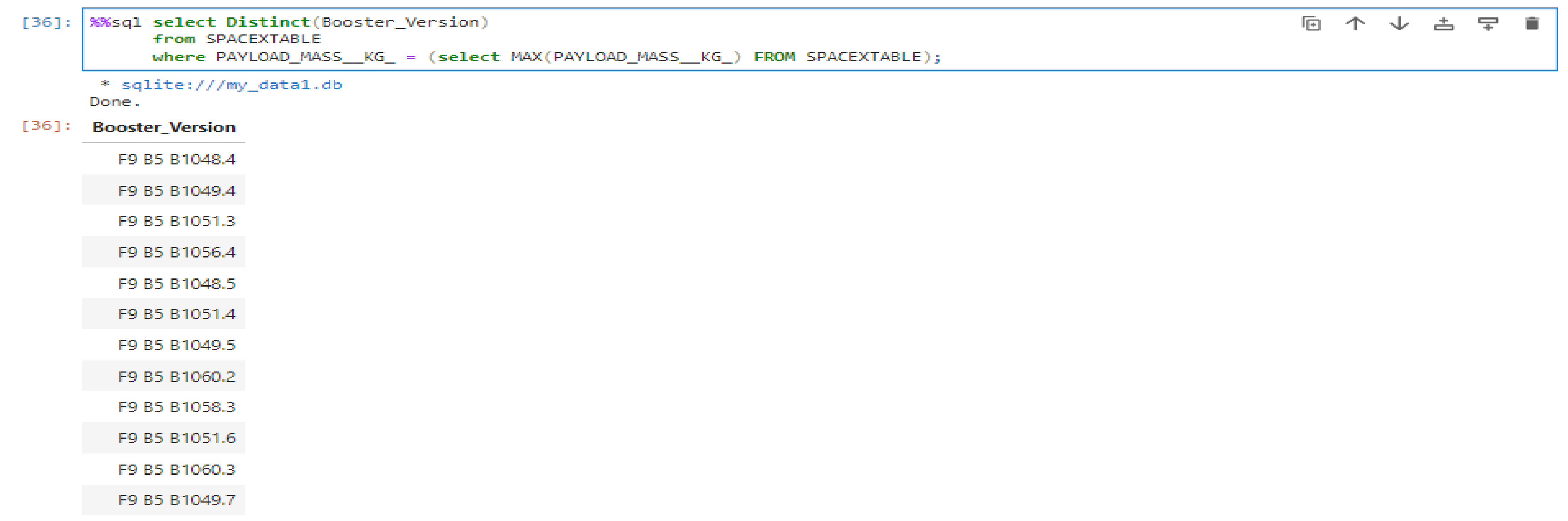
```
 * sqlite:///my_data1.db
Done.
```

[34]:

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

- The query above shows the results. There might need to be some data cleaning to dealing with second and third results

Boosters Carried Maximum Payload



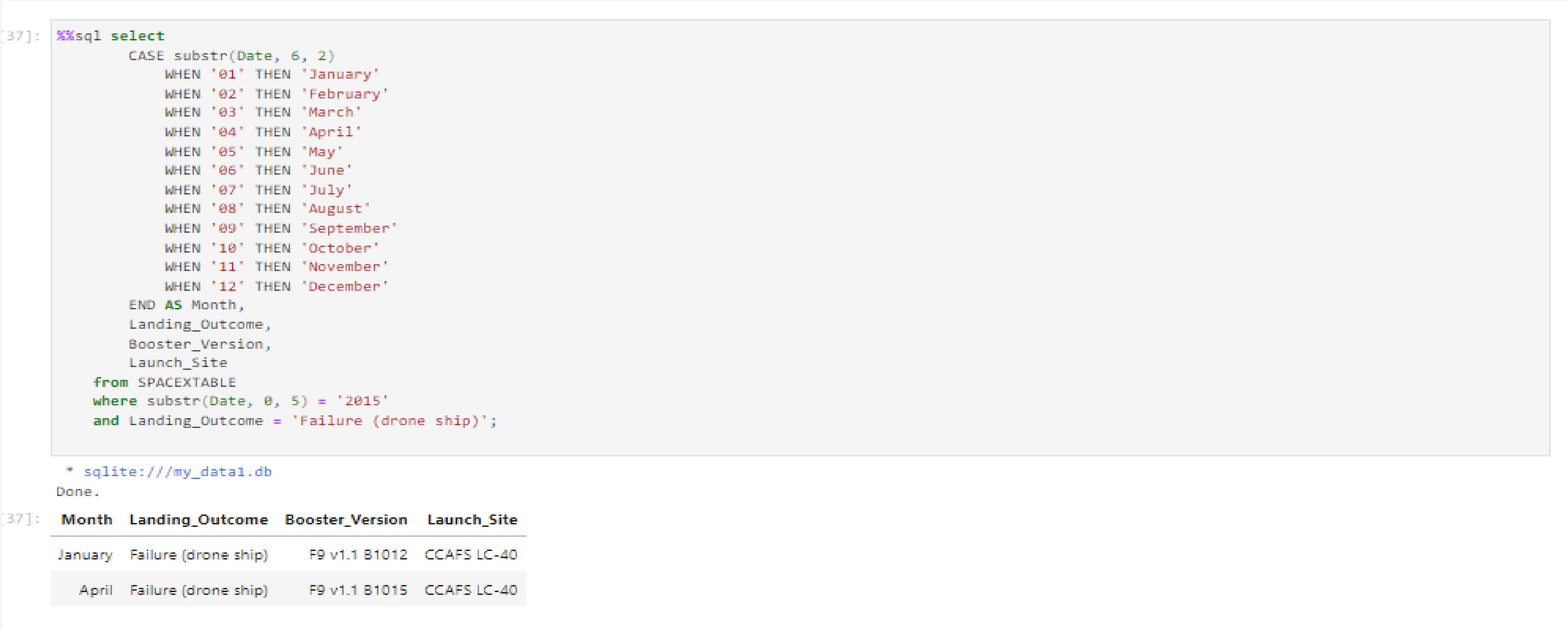
```
[36]: %%sql select Distinct(Booster_Version)
             from SPACEXTABLE
             where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

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

[36]:

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

- There are about 12 booster versions that have carried the maximum payload.

2015 Launch Records



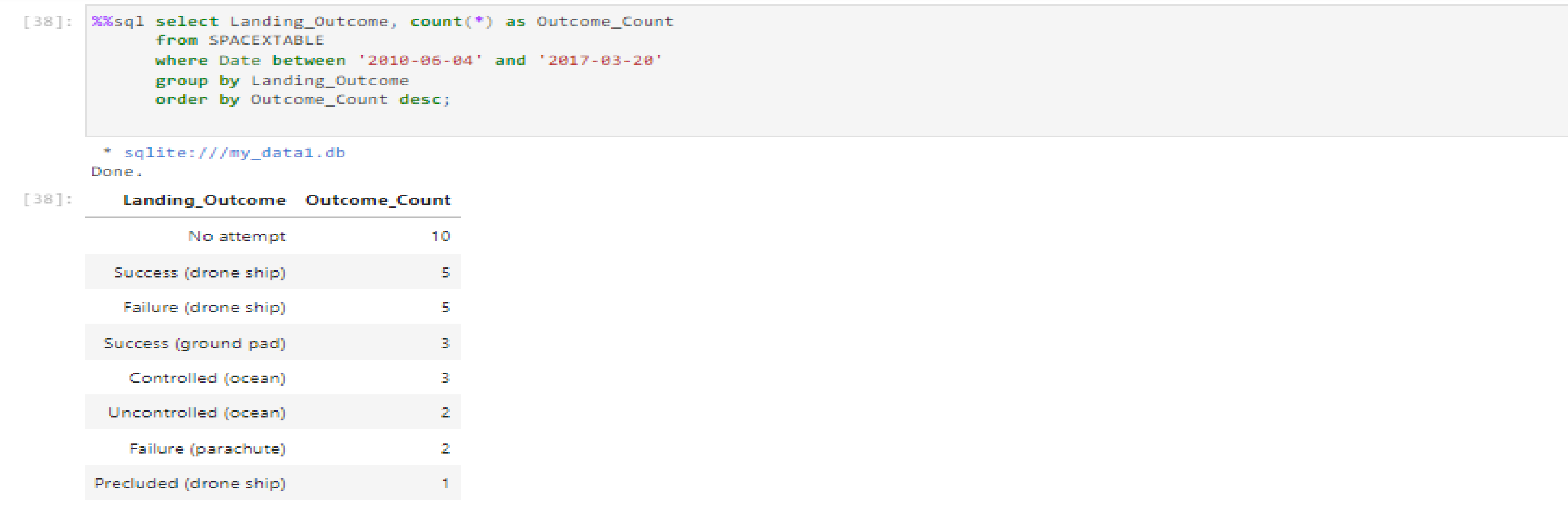
```
%%sql select
        CASE substr(Date, 6, 2)
            WHEN '01' THEN 'January'
            WHEN '02' THEN 'February'
            WHEN '03' THEN 'March'
            WHEN '04' THEN 'April'
            WHEN '05' THEN 'May'
            WHEN '06' THEN 'June'
            WHEN '07' THEN 'July'
            WHEN '08' THEN 'August'
            WHEN '09' THEN 'September'
            WHEN '10' THEN 'October'
            WHEN '11' THEN 'November'
            WHEN '12' THEN 'December'
        END AS Month,
        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.
```

Month	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

- There were only two records that were found to meet this criteria where the landing outcome was Failure landing outcome in drone ships

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



```
[38]: %%sql select Landing_Outcome, count(*) as Outcome_Count
            from SPACEXTABLE
            where Date between '2010-06-04' and '2017-03-20'
            group by Landing_Outcome
            order by Outcome_Count desc;
```

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

[38]:

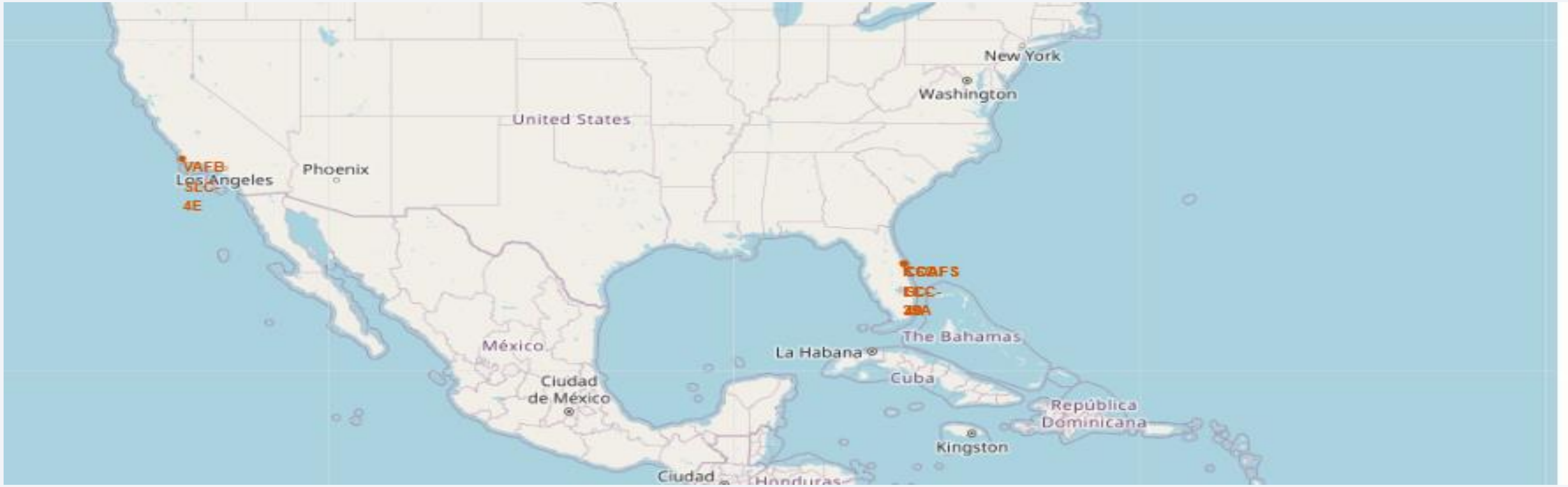
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- The result for the query is shown above. The outcome are in descending order.

Section 3

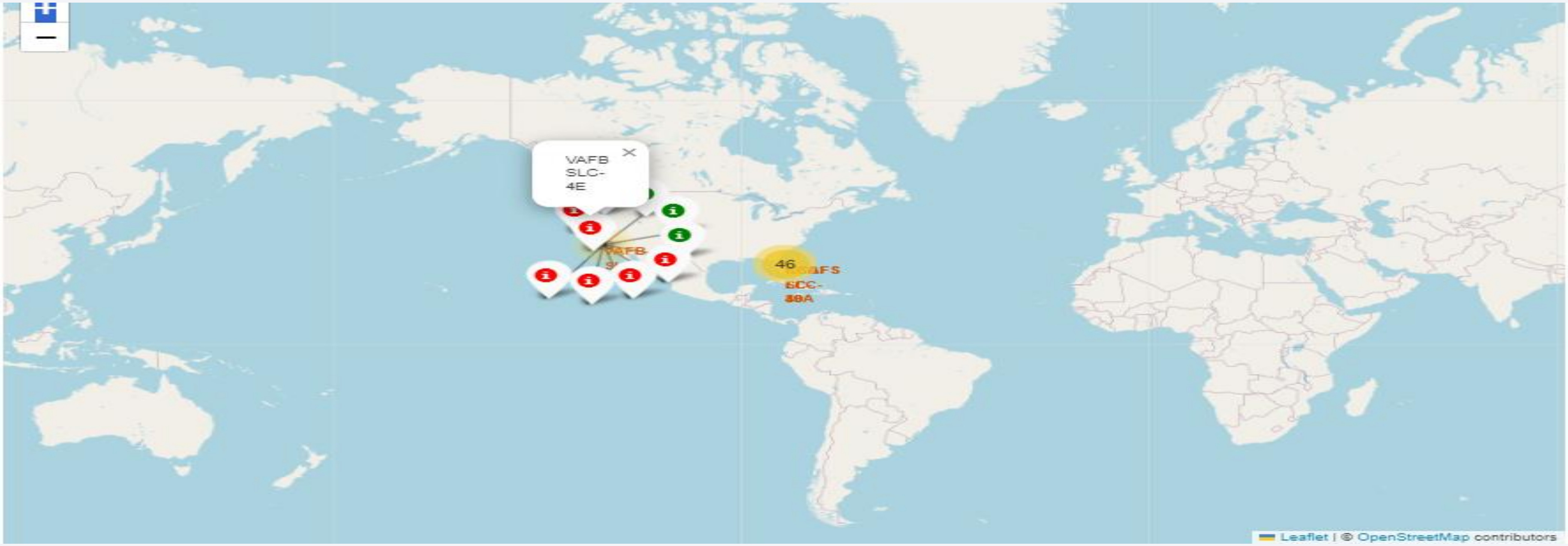
Launch Sites Proximities Analysis

Launch Sites on a global map



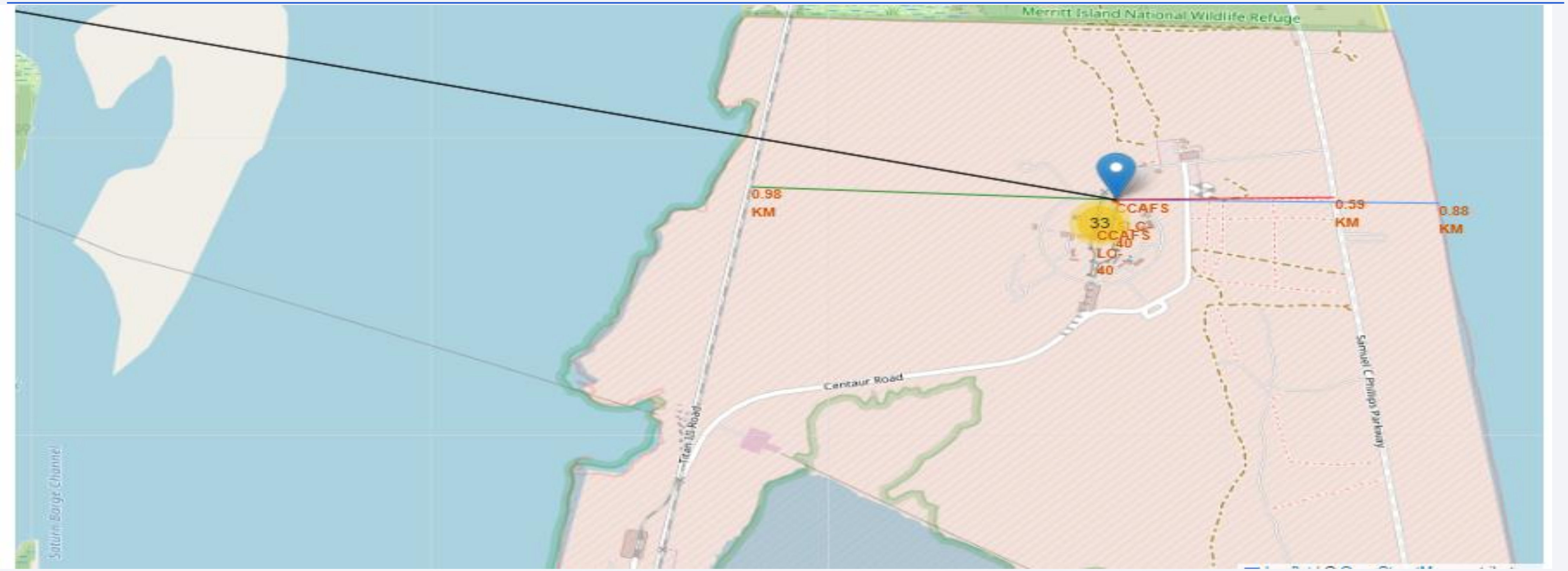
- They are located in remote areas
- All the sites seem to be located near the ocean;
- None of the sites appear to be near the equator;

Success and failure indicators

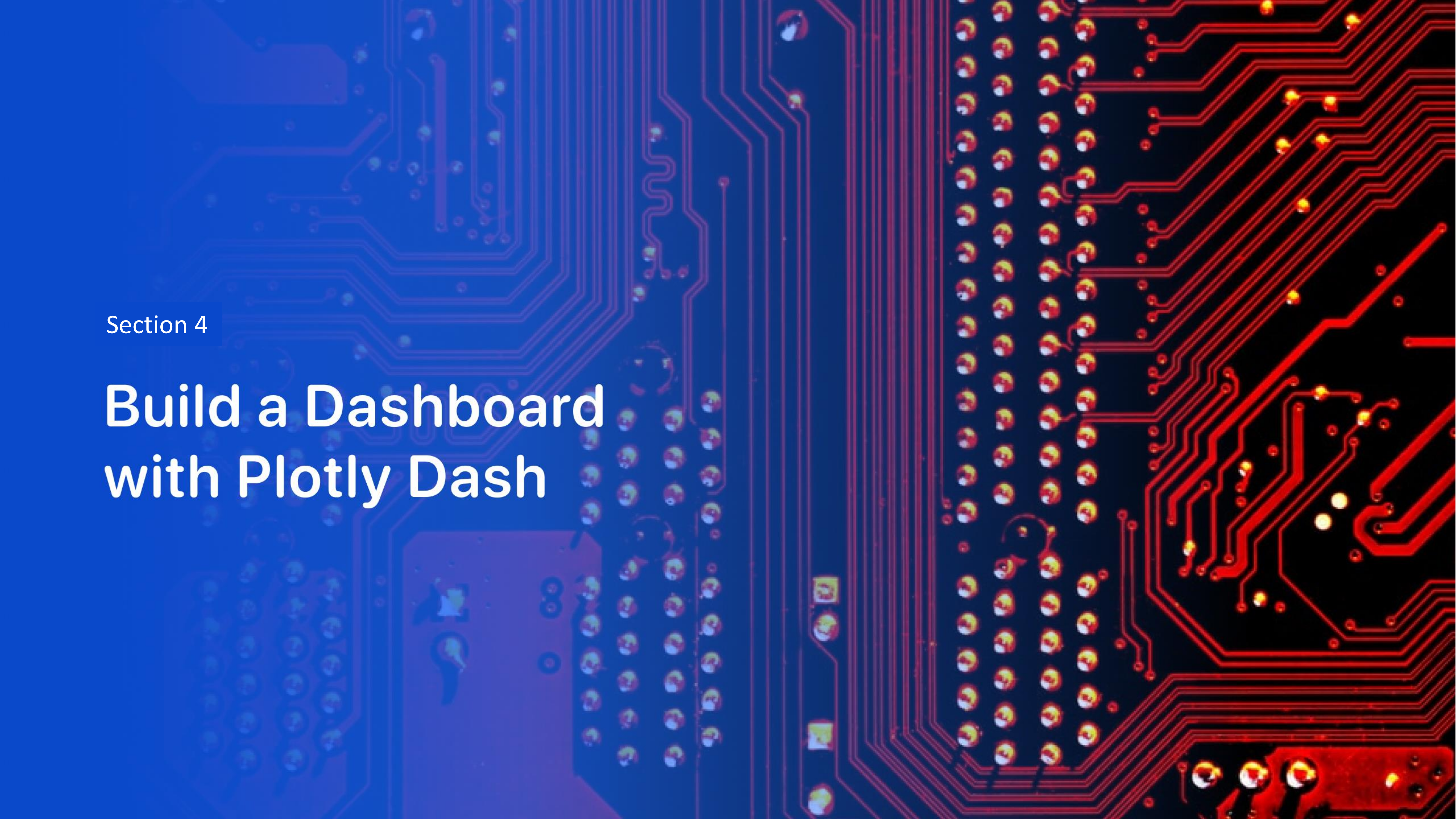


- The image shows one of the sites(VAFB SL 4E);
- In this image, we can see the number of successful launches conducted here(green) and those which failed(red);
- The same information can be found by clicking on the various sites as well

Distances of key areas from CCAFS SLC-40 Site



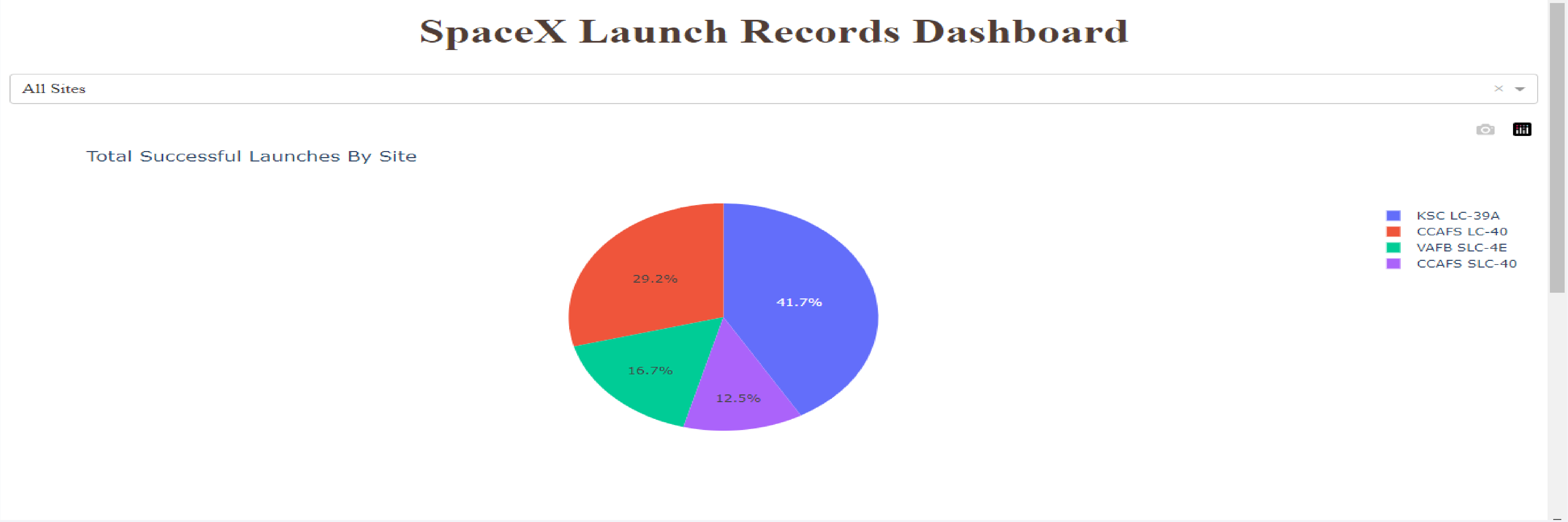
- The image shows the distance of the nearest railway, highway, coastal area and city from the CCAFS SLC-40 site;
- The site is relatively close to a working highway, railroad and coastal line, but very far from closest city.



Section 4

Build a Dashboard with Plotly Dash

Pie Chart Showing success for various Sites

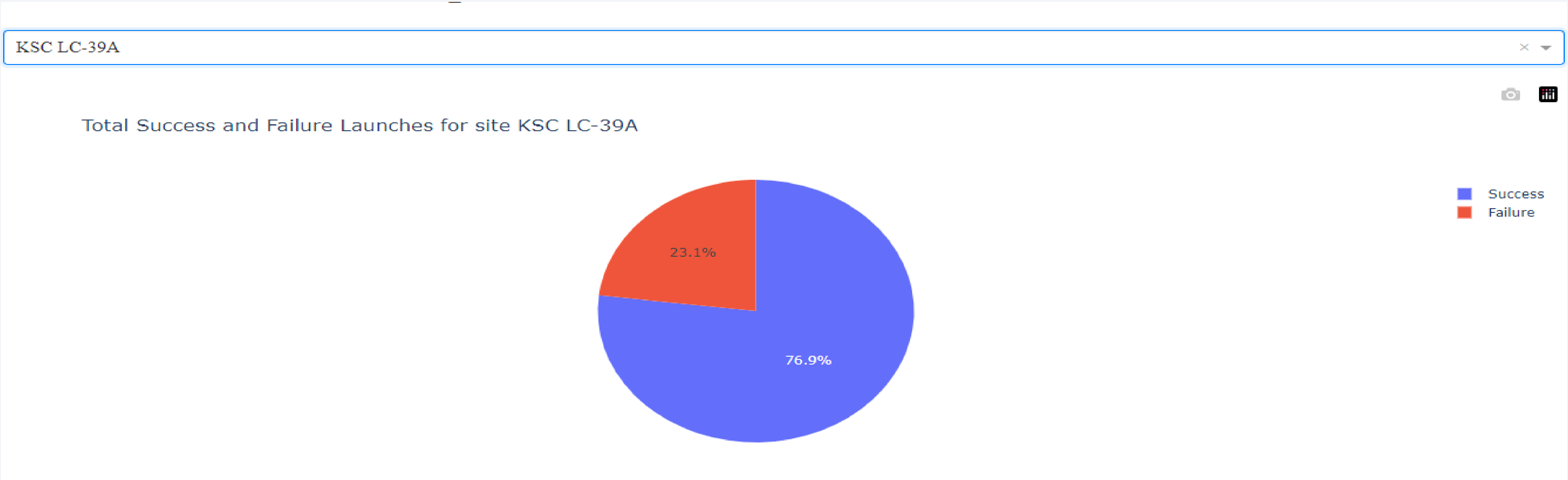


SpaceX Launch Records Dashboard

All Sites

- This dashboard image shows the success rate of all the Launch sites on a pie chart.
- KSC LC-39A boasts the overall highest success rate at 41.7% and CCAFS SLC-40 the lowest at 12.5%

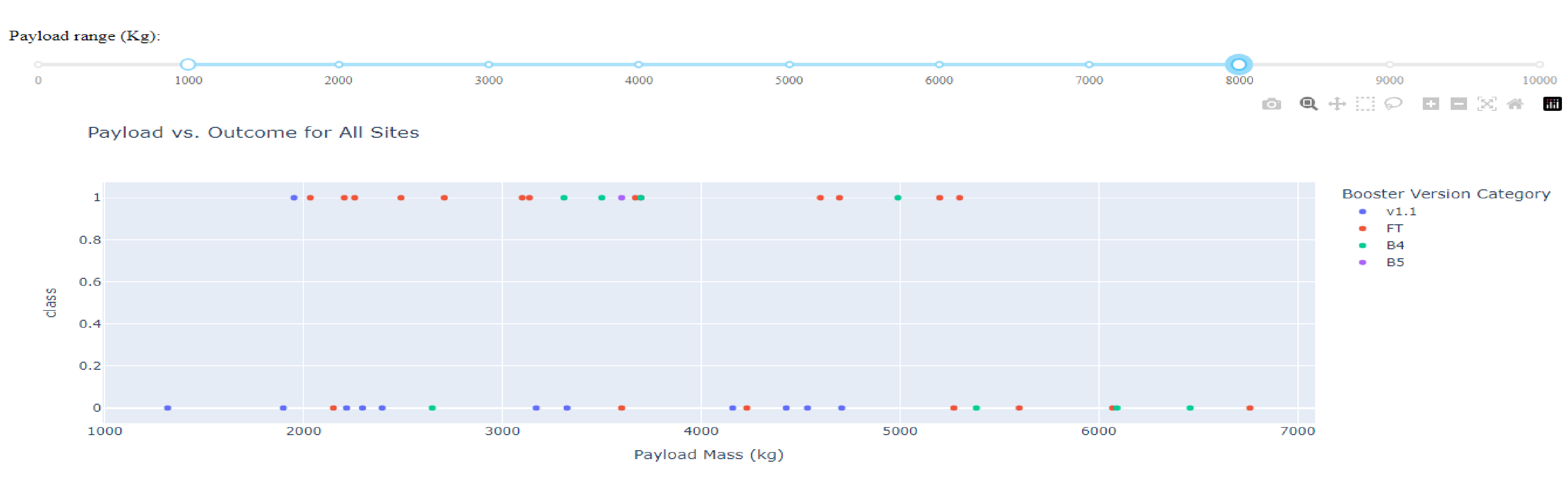
Launch site with the highest Success ratio



KSC LC-39A

The pie chart above shows KSC LC-39A, which has the highest success ratio of all the launch sites used.

Launch outcomes for different Boosters at various Payloads

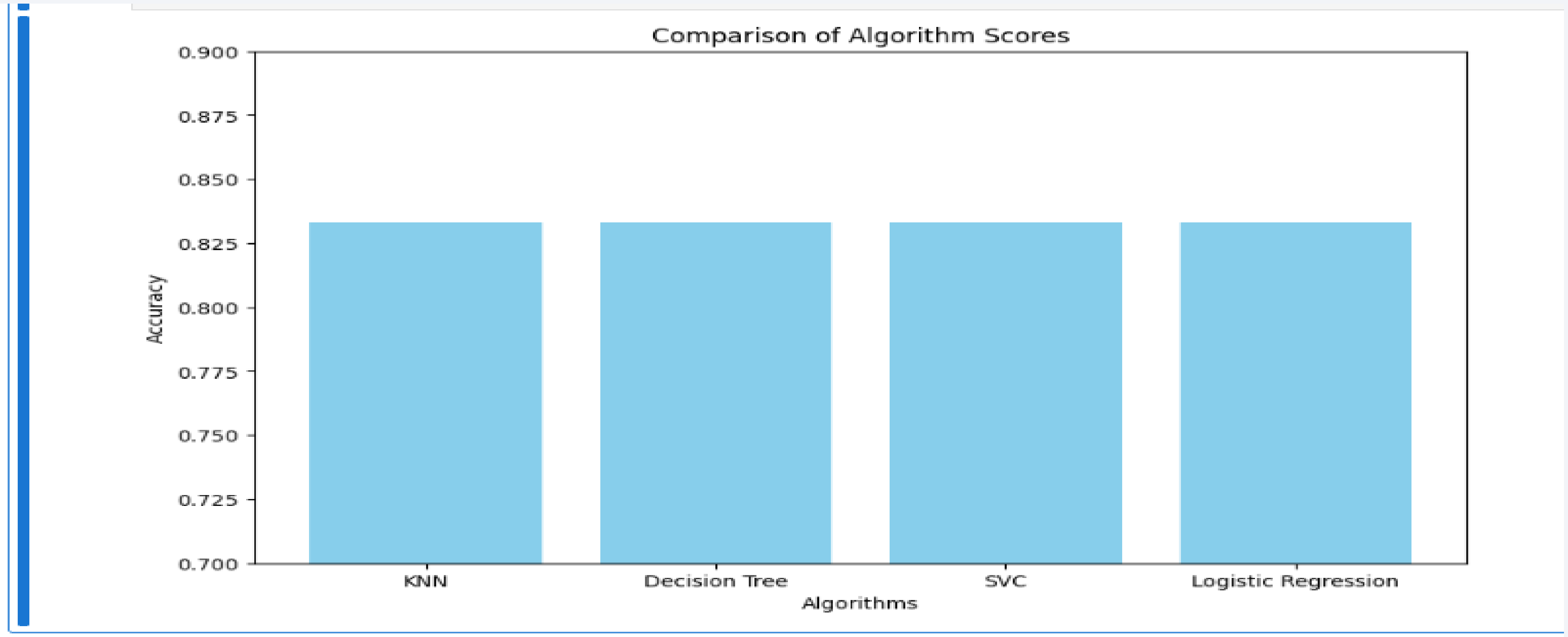


- For the particular Payload ranges provided, the Booster with the highest success is the FT booster.
- The v1.1 seems to have the highest failures

Section 5

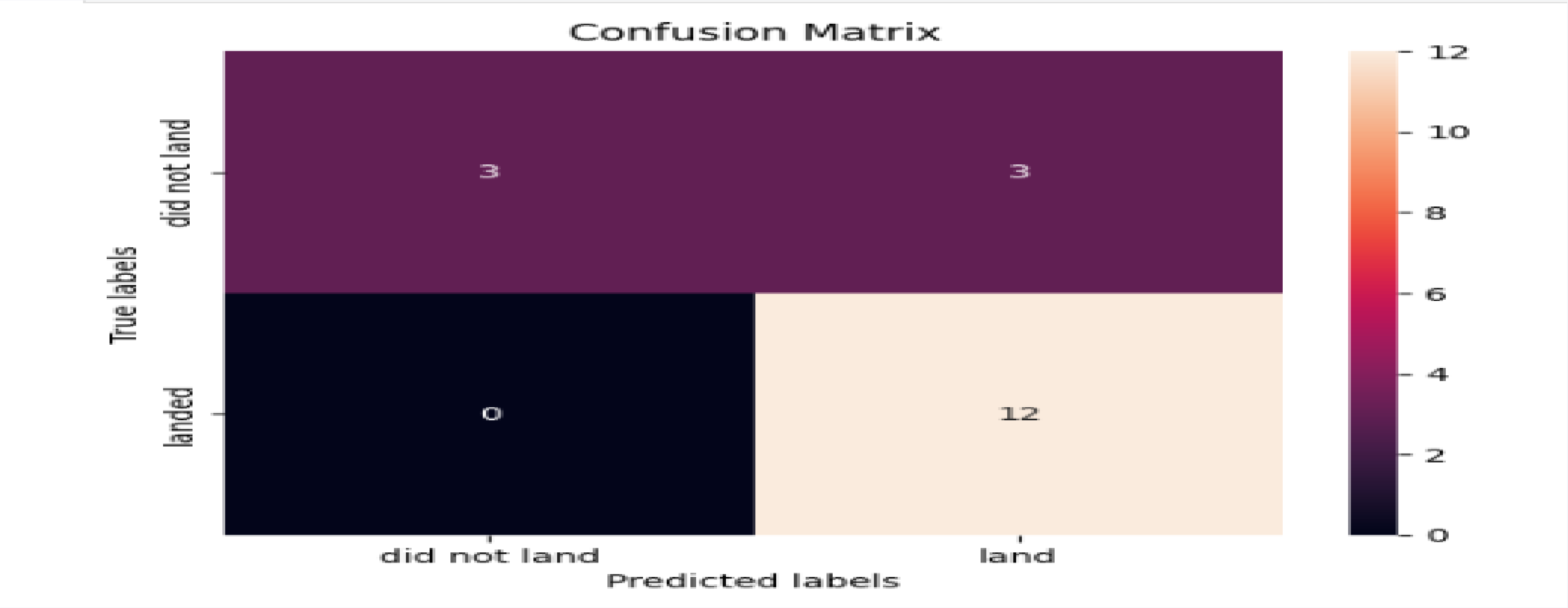
Predictive Analysis (Classification)

Classification Accuracy



- When the models were tested on the test set, they all seem to have similar accuracy score.
- Decision trees can be visually easy to explain visually so it would be used.

Confusion Matrix



- The algorithm seems to have 100% ability to detect the landing outcome.
- There seems to be 50% chance of misclassifying the ‘did not land’ instances as landing.

Conclusions

- The algorithms could be used predict if a launch will be a success or not. However, the ability to detect failed launched needs to be improved
- The Dataset used to train the models needs to be increased to improve model accuracy.
- SpaceX seems to have a lot more successful launches with time. This means we might expect their costs to remain competitive in the future.

Appendix

- SQL queries: https://github.com/motlatsimoea/final_exam/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb
- Visualzation charts: https://github.com/motlatsimoea/final_exam/blob/main/edadataviz.ipynb
- Code for model training: https://github.com/motlatsimoea/final_exam/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb
- Rest of the files: https://github.com/motlatsimoea/final_exam/tree/main/dash
- Code for dashboard: https://github.com/motlatsimoea/final_exam/blob/main/dash.txt

Thank you!