IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kevin Christian Rodriguez-Siu
April 08th, 2025

Outline

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

Executive Summary

- In this project, we use Data Science to help determine if the first stage component of Falcon 9 rockets used by SpaceX land successfully or not. A successful prediction can be crucial to determine the cost of a launch and, therefore, has a direct impact in the proper allocation of resources.
- Methods like data collection and web scraping were used to gather relevant information. After a process of data wrangling, exploratory data analysis and visualization techniques were used to analyze what factors contribute to a successful landing. Finally, machine learning algorithms were built and trained on the available data.
- The results of this project include some visualization tools, like maps and a dashboard, as well as a machine learning pipeline that has an overall accuracy of 83.33%, with a 100% percent of accuracy on our testing data for landed components.

Introduction

- 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, making SpaceX a more affordable option.
- A big factor on the savings is that SpaceX can reuse the first stage of the Falcon 9 rocket when they can be retrieved.
- Regardless of the outcome of a mission, the first stage can't always be recovered, and Space X has data on the success of these recoveries.
- **Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.** This is very important for both SpaceX and any other company that wants to bid against them for a rocket launch.
- Our main goal then can be summarized as: **Can we use the available data on Falcon 9 rocket launches to determine if the first stage will land successfully?**

Section 1

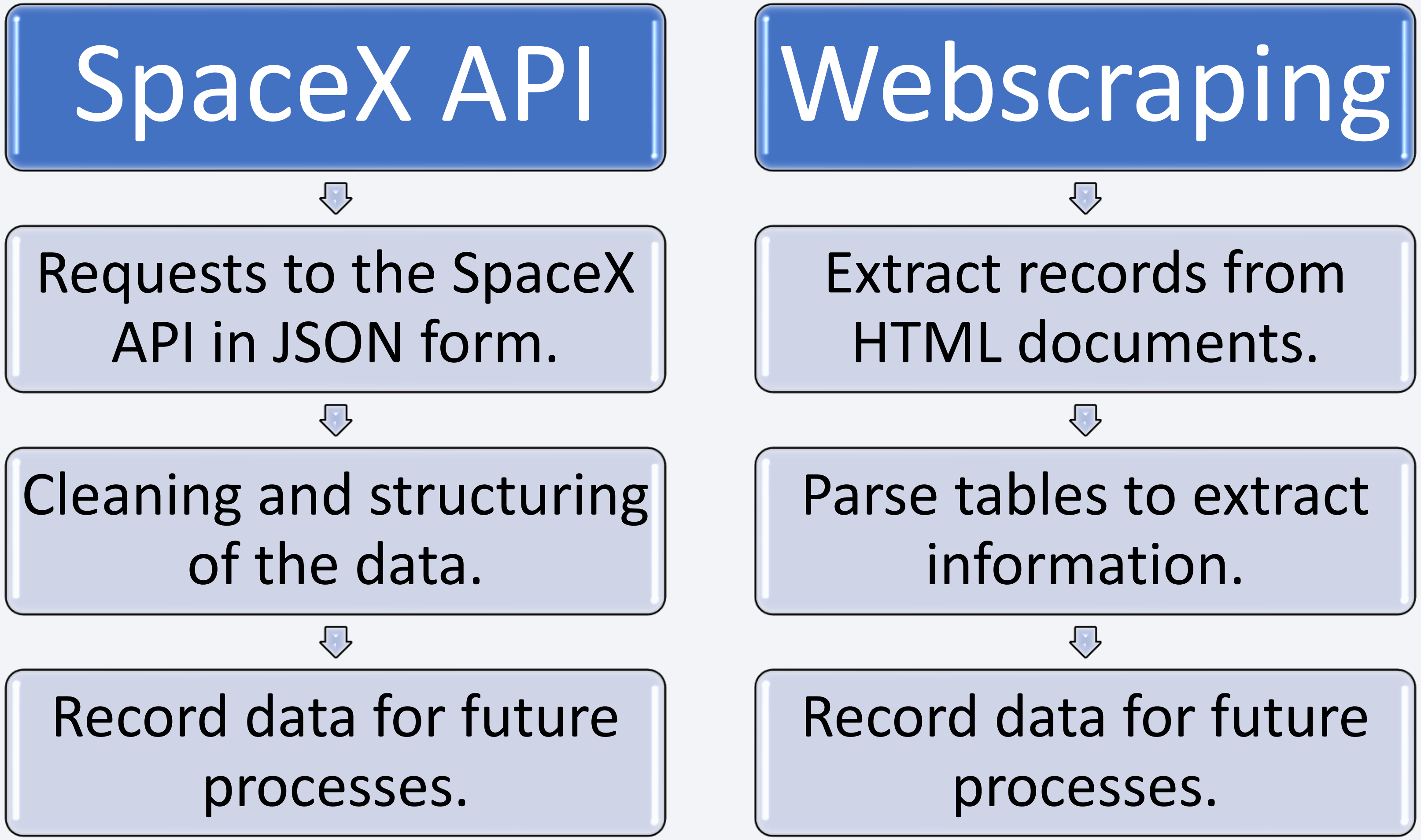
Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Space X has an API that offers data on their rocket launches.
    - Webscraping was also used to collect historical launch records.
- Perform data wrangling
    - Missing values and landing outcomes were determined.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - For this, 4 supervised methods were tested: SVM, Logistic Regression, Classification Trees, and KNN.

Data Collection



SpaceX API

Requests to the SpaceX API in JSON form.

Cleaning and structuring of the data.

Record data for future processes.

Webscraping

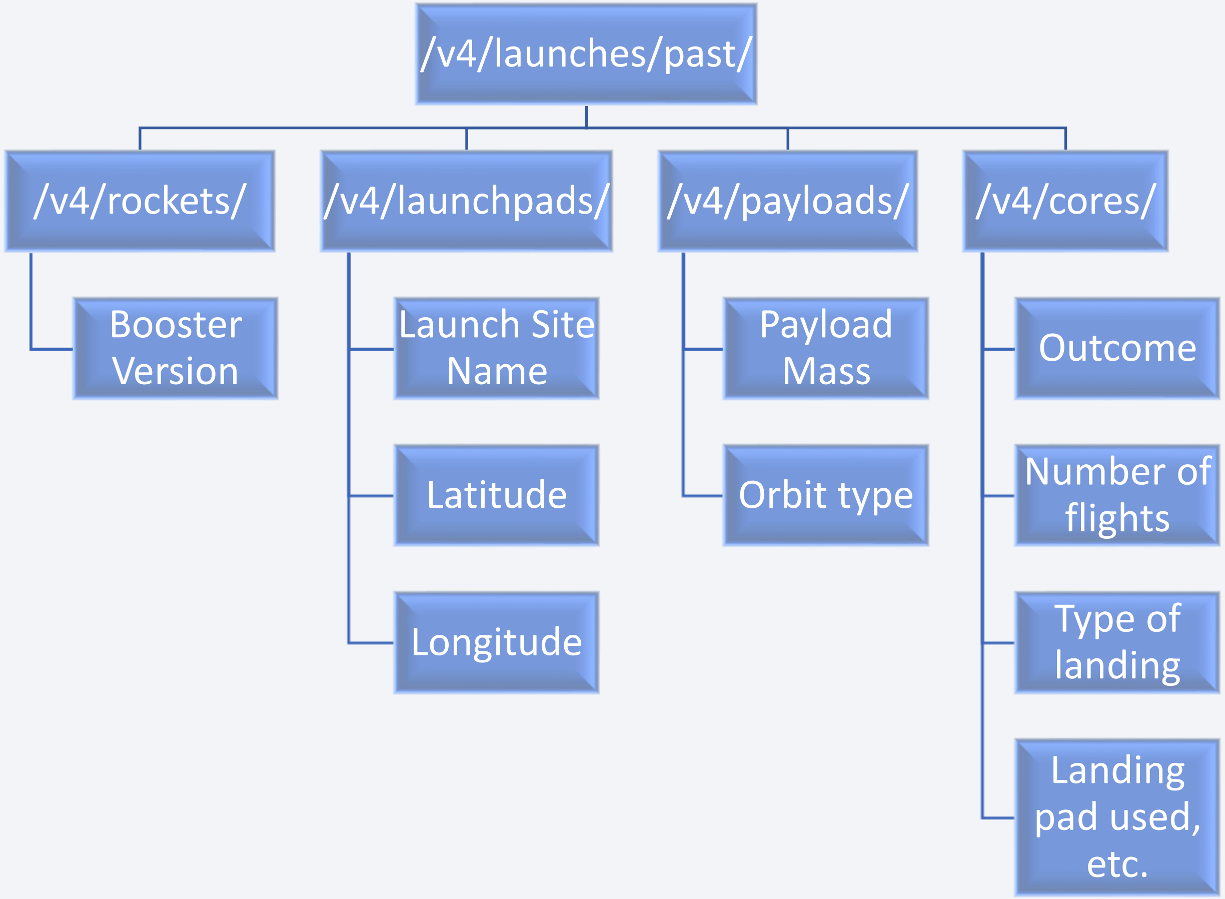
Extract records from HTML documents.

Parse tables to extract information.

Record data for future processes.

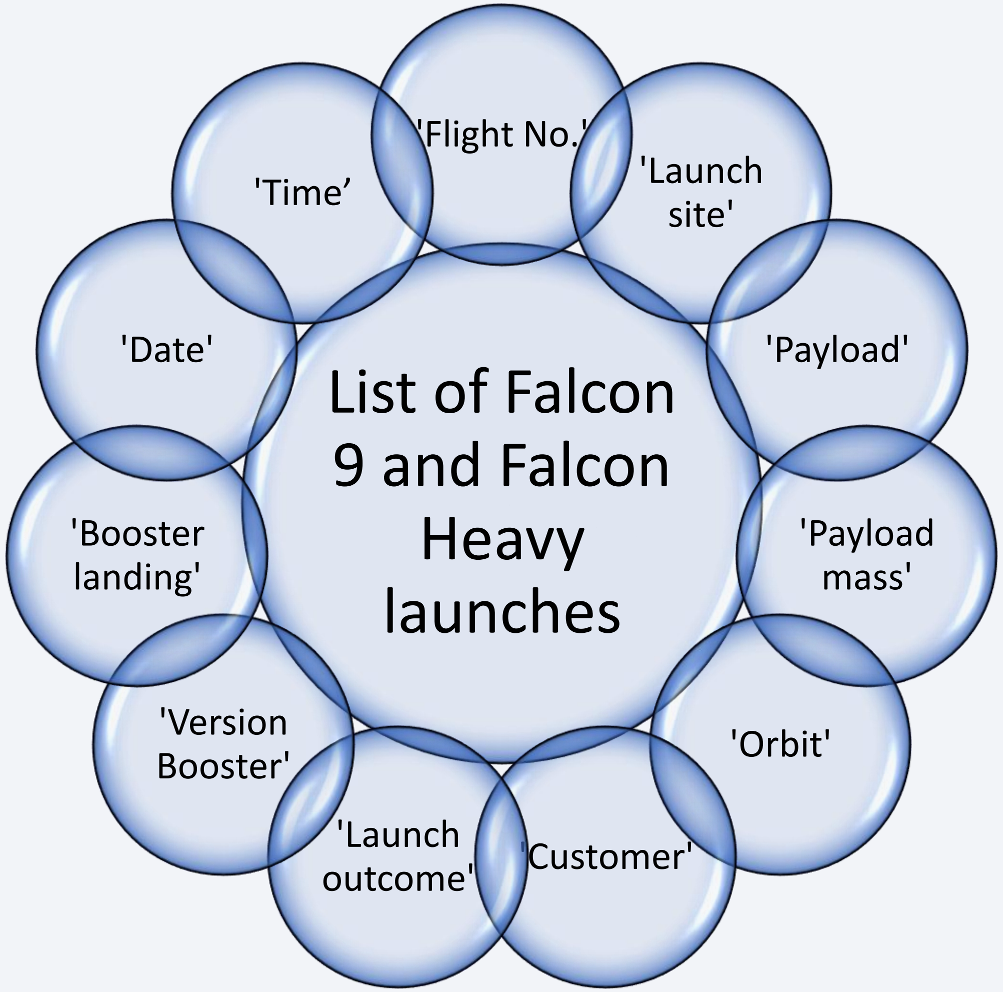
Data Collection – SpaceX API

- The SpaceX REST API was used, specifically to gather information on the specific rockets, launchpads, payloads and cores given a particular launch.
- All the responses are given in JSON format, that is parsed accordingly.
- We then filter only launches using a Falcon 9 rocket and clean up missing values for Payload mass.
- https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/001-jupyter-labs-spacex-data-collection-api.ipynb



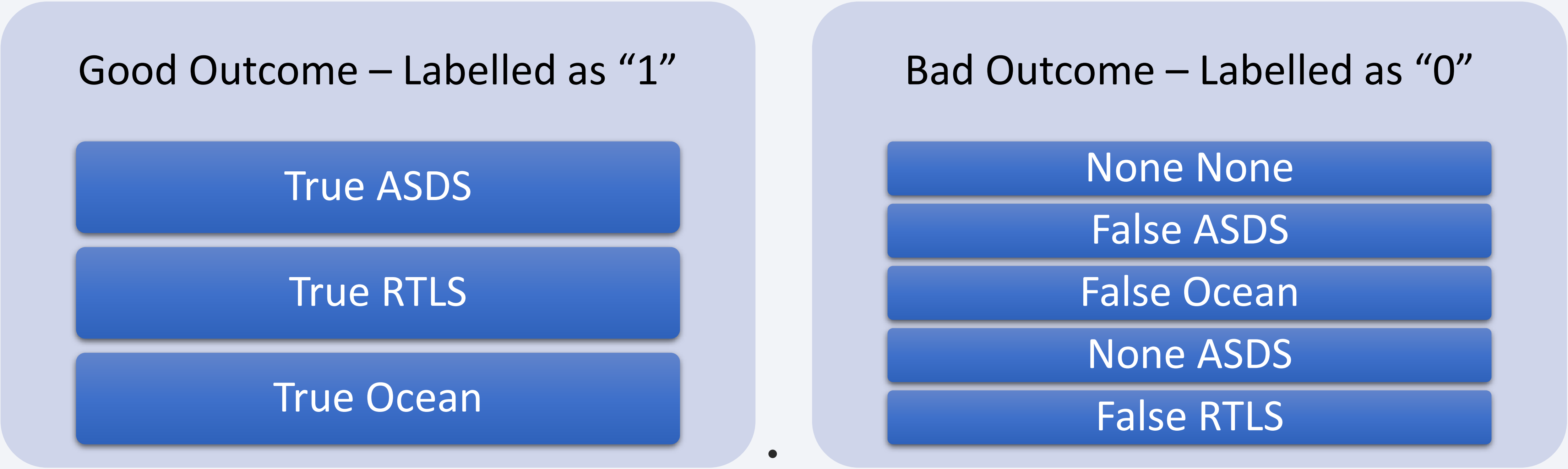
Data Collection - Scraping

- Wikipedia has records of the Falcon 9 launches, including if the booster landed or not.
- Request of this information is given in HTML format, that needs to be parsed to find the appropriate records.
- Table is then parsed using BeautifulSoup and converted into a Pandas DataFrame and a CSV file.
- https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/002-jupyter-labs-webscraping.ipynb



Data Wrangling

- The data collected was then pre-processed, dealing with missing values and some preliminary analysis were done:
  - Number of launches on each site and number and occurrence of each orbit was calculated.
  - Number and occurrence of each mission outcome was determined.
- Then, landings were labeled according to their mission outcome:



Good Outcome – Labelled as “1”

- True ASDS
- True RTLS
- True Ocean

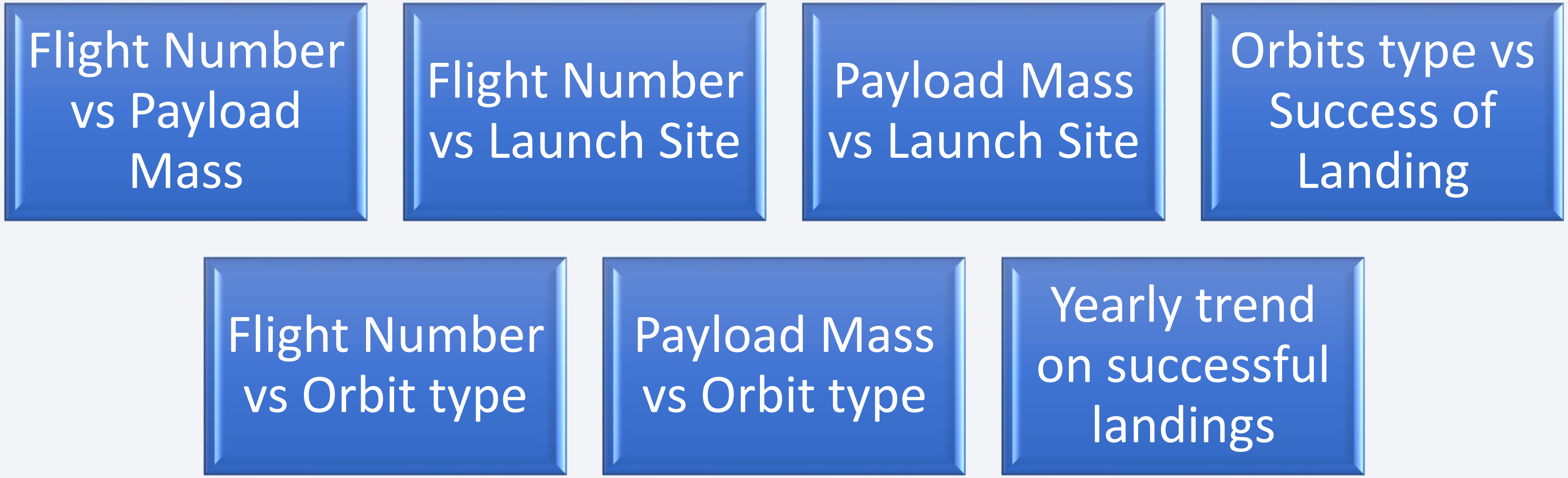
Bad Outcome – Labelled as “0”

- None None
- False ASDS
- False Ocean
- None ASDS
- False RTLS

https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/003-labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- We used Matplotlib to visualize the relationships between the data to find, if possible, where correlations were apparent. These will be shown in detail in the next section.



Flight Number vs Payload Mass

Flight Number vs Launch Site

Payload Mass vs Launch Site

Orbits type vs Success of Landing

Flight Number vs Orbit type

Payload Mass vs Orbit type

Yearly trend on successful landings

- With these graphs, we could also determine what features are the most impactful in the success of the landings, allowing us to do Feature Engineering.

- https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/005-edadataviz.ipynb

EDA with SQL

- SQL was also used to execute queries to review some of the features present in the database and determine both **specific ranges of numerical features** and **unique values of categorical features**.
- Some of the queries done include:
  - Number of unique launch sites, as well as records of specific launch sites depending on their name.
  - Total and average payload mass carried by specific customers or booster versions.
  - Date of the first successful landing outcome in a ground pad.
  - Booster versions with successful landing in drone ships within a range of payload mass.
  - Total number of successful and failure mission outcomes.
  - Boosters that have carried the maximum payload, etc.
- https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/004-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Using Folium, a map was created to showcase the locations of the launch sites, as well as markers that indicated both successful and failure of landings. This was done to determine visually if specific launch sites had better ratios of success.

- Additionally, the map allowed to see the proximity of launch sites with other geographical elements like the coastline, highways, railways or cities. This helped determine if the launch sites had specific places closer that could have an impact in the success of landings.

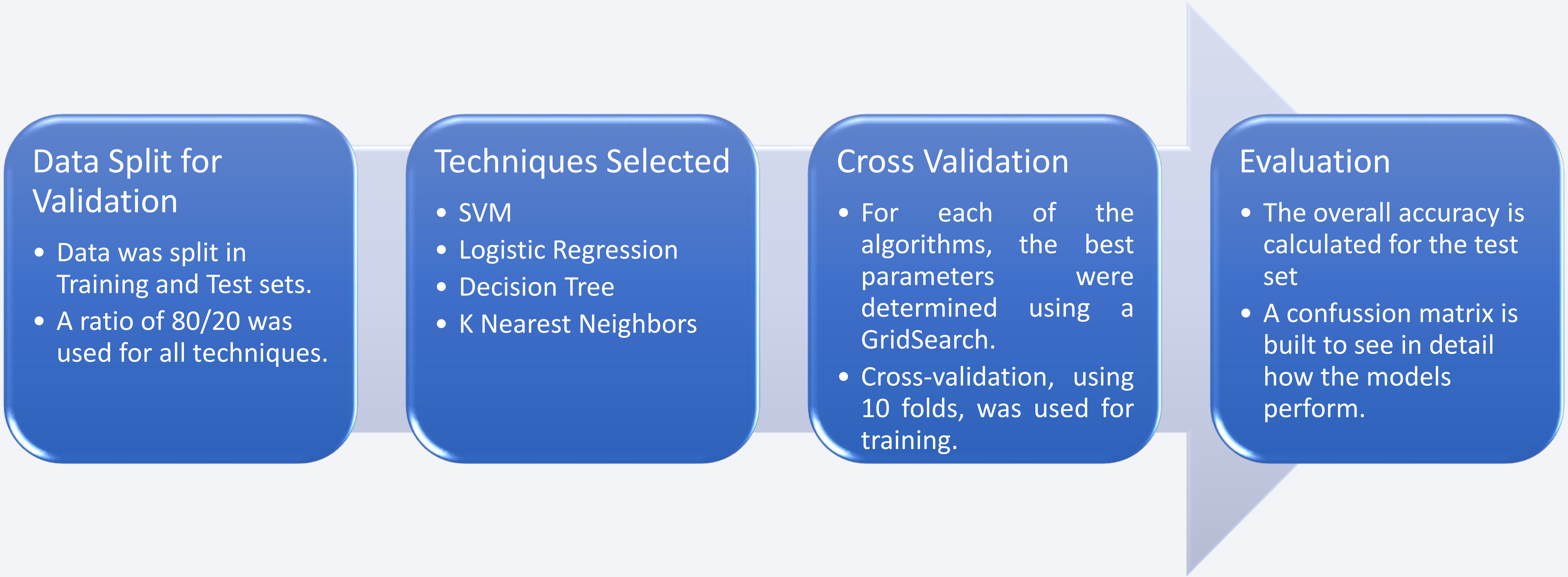
- The resulting maps are interactive and can be navigated, allowing for a better exploration of additional characteristics.

- https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/006-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Using Plotly Dash, an interactive dashboard was created that has the following components:
  - A pie chart showing successful launches, both for all launch sites and for each launch site separately
  - A scatter plot that shows the correlation between successful launches within a range of payload masses, both for all launch sites and for each launch site separately.
- These plots and interactions helped us determine which sites have best chance of success, as well as visualizing if the mass of the payload was crucial to the success of the landing.
- https://github.com/thekcrs/DataScienceCapstone/tree/main/Plotly%20Dashboard

Predictive Analysis (Classification)



Data Split for Validation

- Data was split in Training and Test sets.
- A ratio of 80/20 was used for all techniques.

Techniques Selected

- SVM
- Logistic Regression
- Decision Tree
- K Nearest Neighbors

Cross Validation

- For each of the algorithms, the best parameters were determined using a GridSearch.
- Cross-validation, using 10 folds, was used for training.

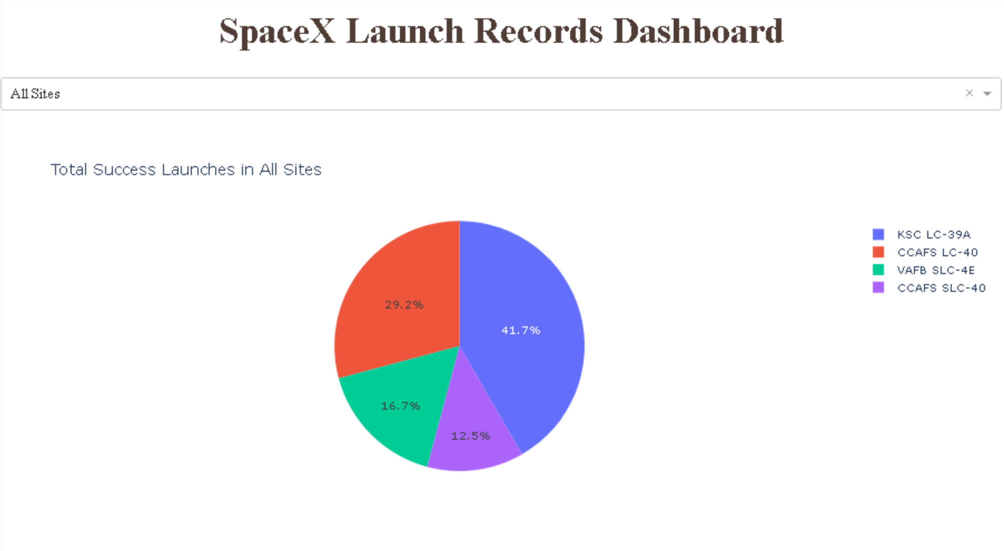
Evaluation

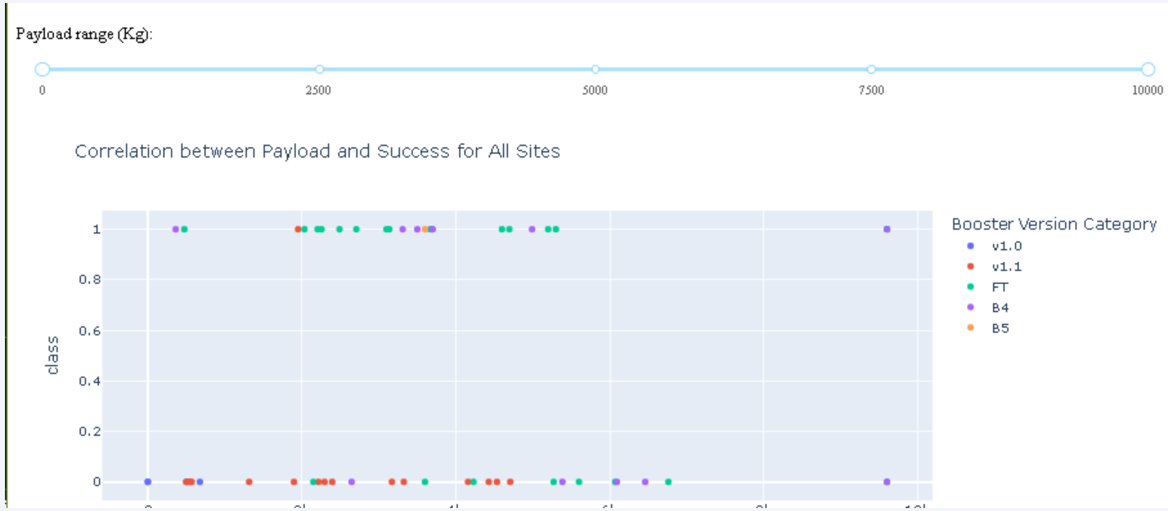
- The overall accuracy is calculated for the test set
- A confussion matrix is built to see in detail how the models perform.

- https://github.com/thekcrs/DataScienceCapstone/blob/main/Jupyter%20Notebooks/008-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- In our analysis of the data, the parameters that were seeing as the more impactful for the success of the landing were: 'FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial'





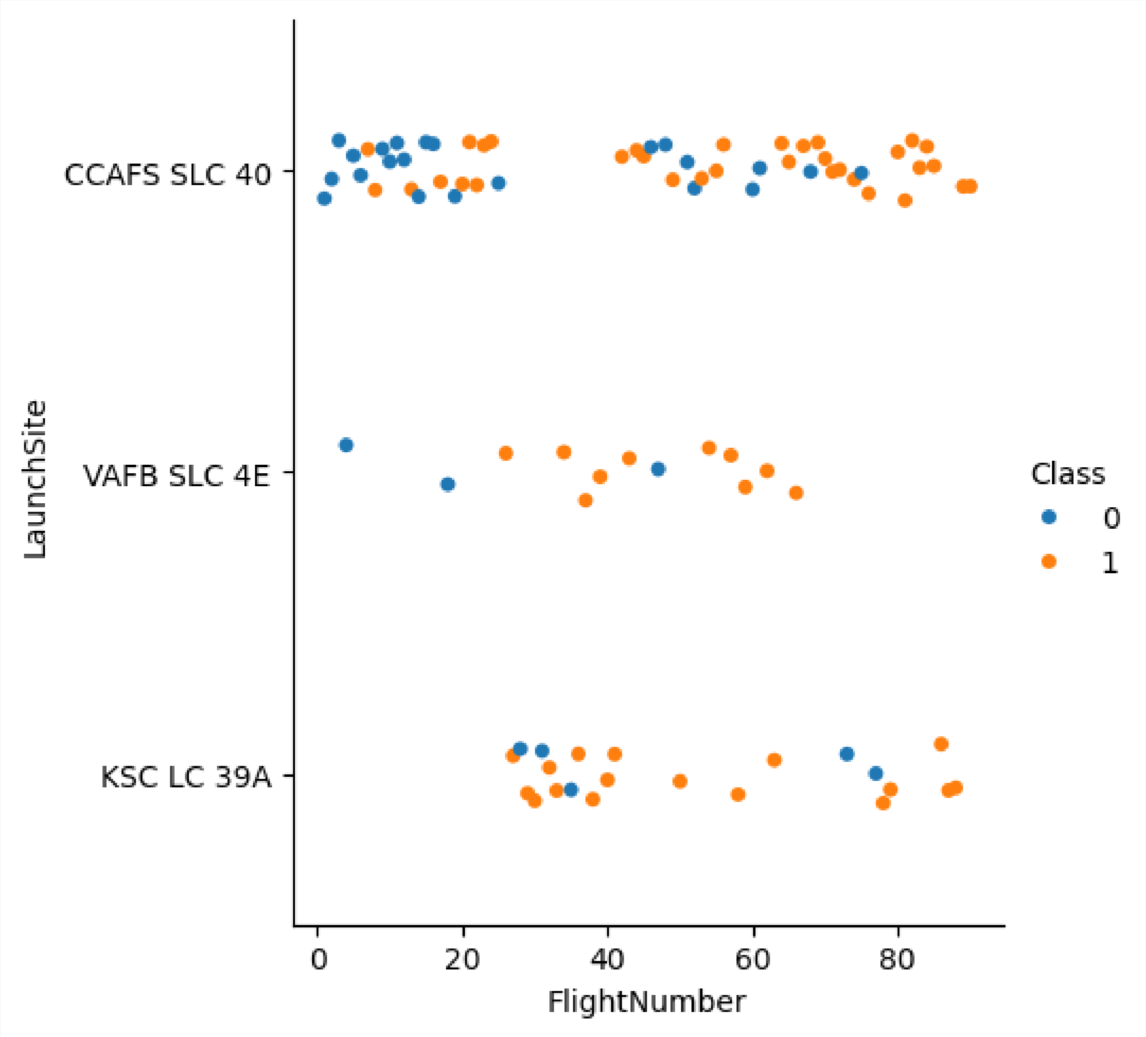
- Our models reach an 83.33% of general accuracy, with a 100% percent of accuracy on our testing data for landed components. The difference between accuracy happens mainly when mislabeling components that didn’t land.

Section 2

Insights drawn from EDA

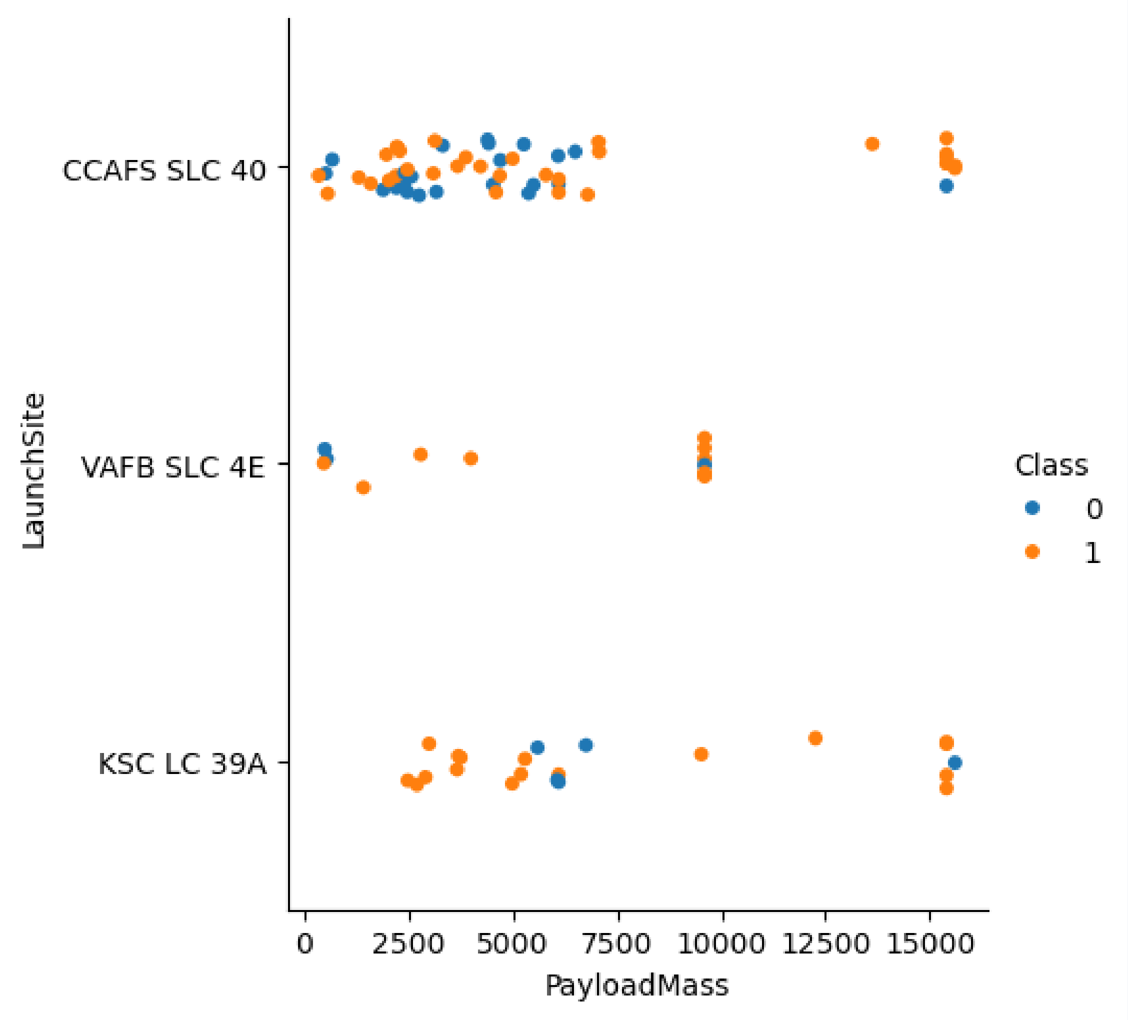
Flight Number vs. Launch Site

- Here we see a scatter plot showing the relationship between Flight Number vs. Launch Site.
- As we can see, as time progresses (flight number is higher), the ratio of success tends to be higher, regardless of what Launch Site is being used.
- CCAFS SLC 40 does have the most failures but also has the most launches and most of the failures have happened in past launches, as opposed of the recent ones.



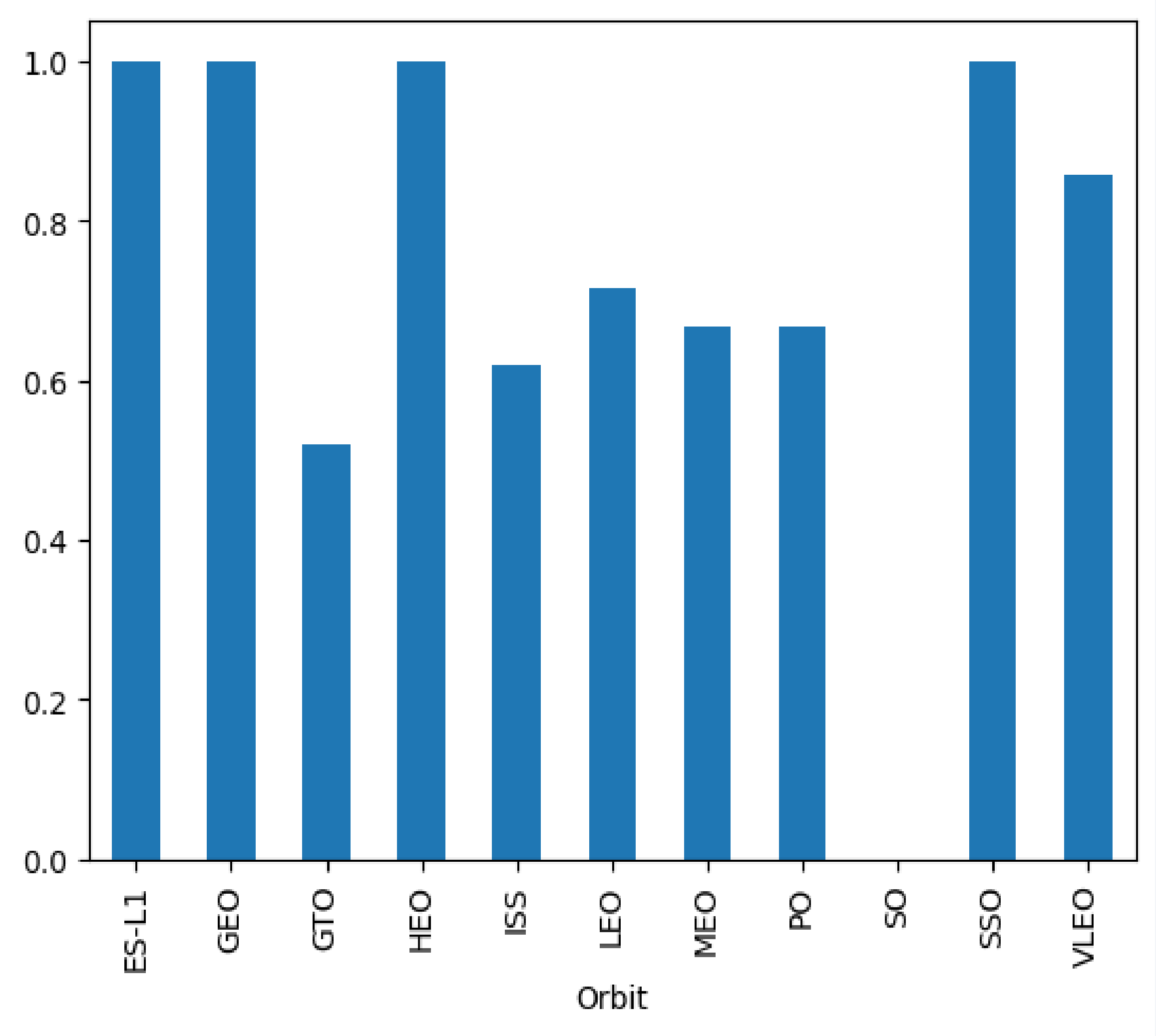
Payload vs. Launch Site

- Here we see a scatter plot showing the relationship between Payload Mass vs. Launch Site.
- For the VAFB-SLC launch site there are no rockets launched with a heavy payload mass (greater than 10000).
- For the other launch sites, their failures tend to happen for payloads less than 10000.



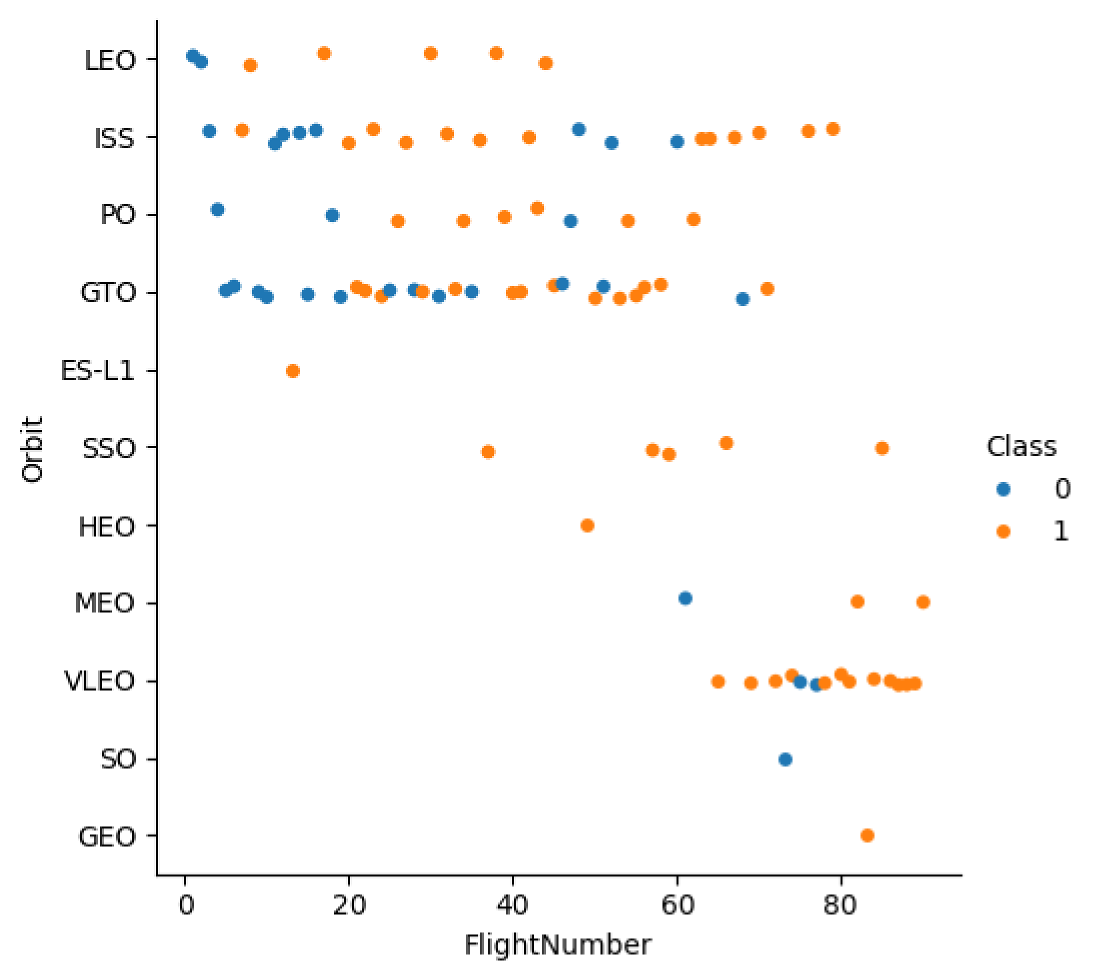
Success Rate vs. Orbit Type

- Here we see a bar plot showing the relationship between the Success Rate vs the Orbit Type.
- Orbit Types do seem to have some impact in the success of the landing.
- ES-L1, GEO, HEO and SSO show a 100% of success rate.
- VLEO shows a good rate higher than 80%.
- GTO, ISS, MEO, LEO and PO show moderated success. SO doesn't have any successful landing.



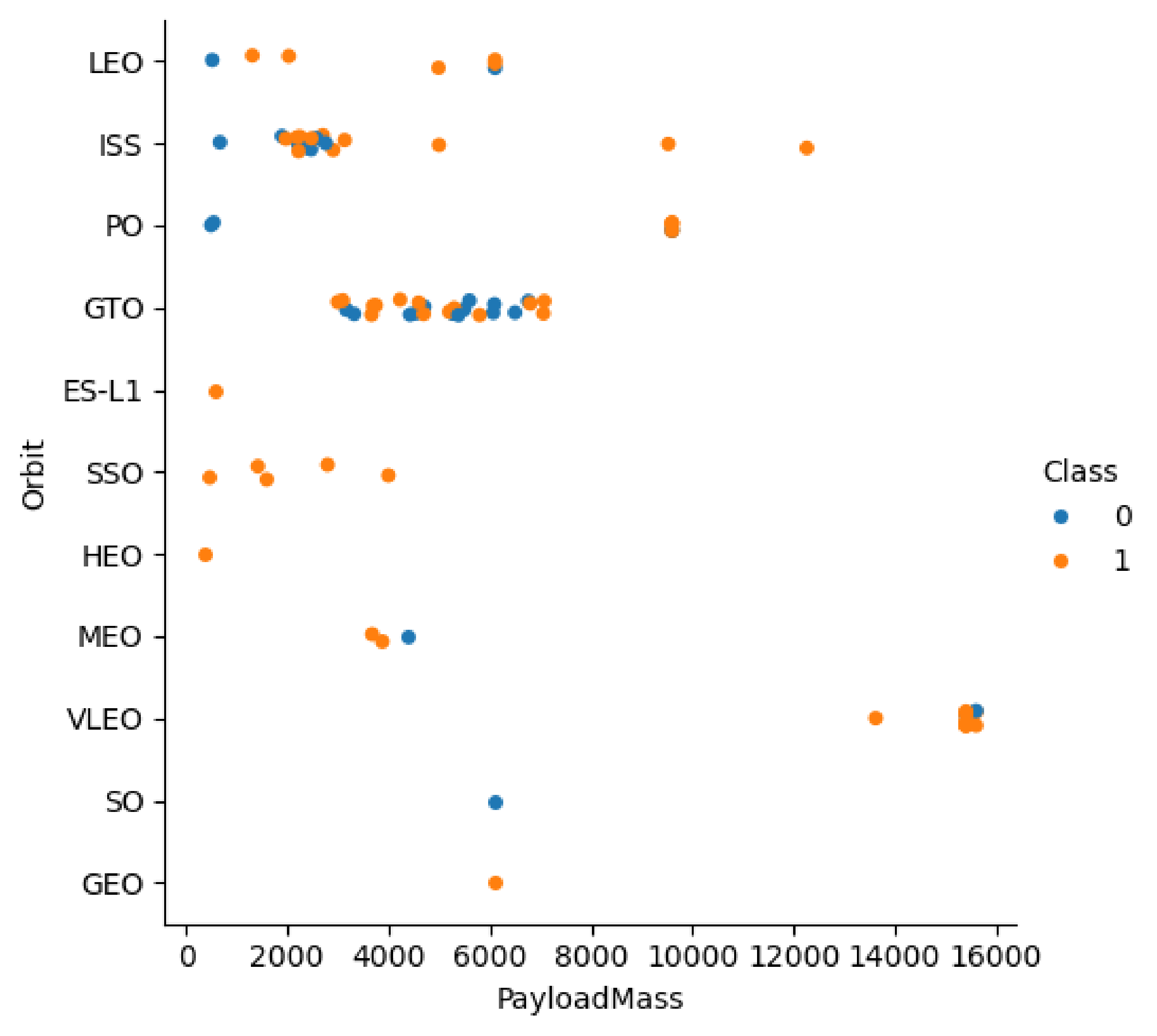
Flight Number vs. Orbit Type

- Here we see a scatter plot showing the relationship between Flight Number vs. Orbit Type.
- In some cases, like the LEO and PO orbits, success seems to be related to the number of flights.
- Conversely, in the GTO or ISS orbits, there appears to be no relationship between flight number and success.
- The low rate of success of SO is also explained as it seems only 1 flight has been done for this orbit.



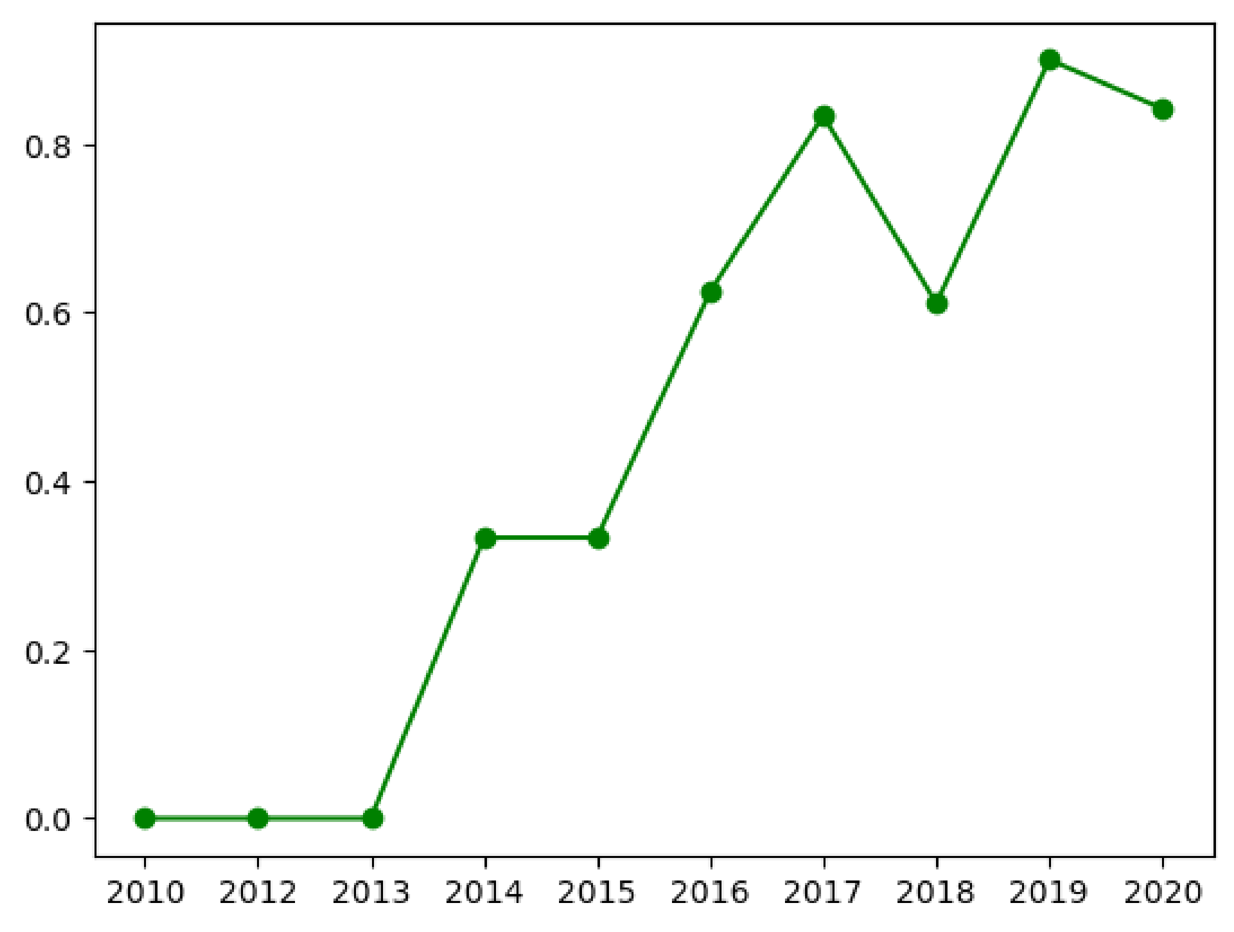
Payload vs. Orbit Type

- Here we see a scatter plot showing the relationship between Payload Mass vs. Orbit Type.
- With heavy payloads the successful landing or positive landing rate are more for PO, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.
- LEO and SO also have very small range of payload values.

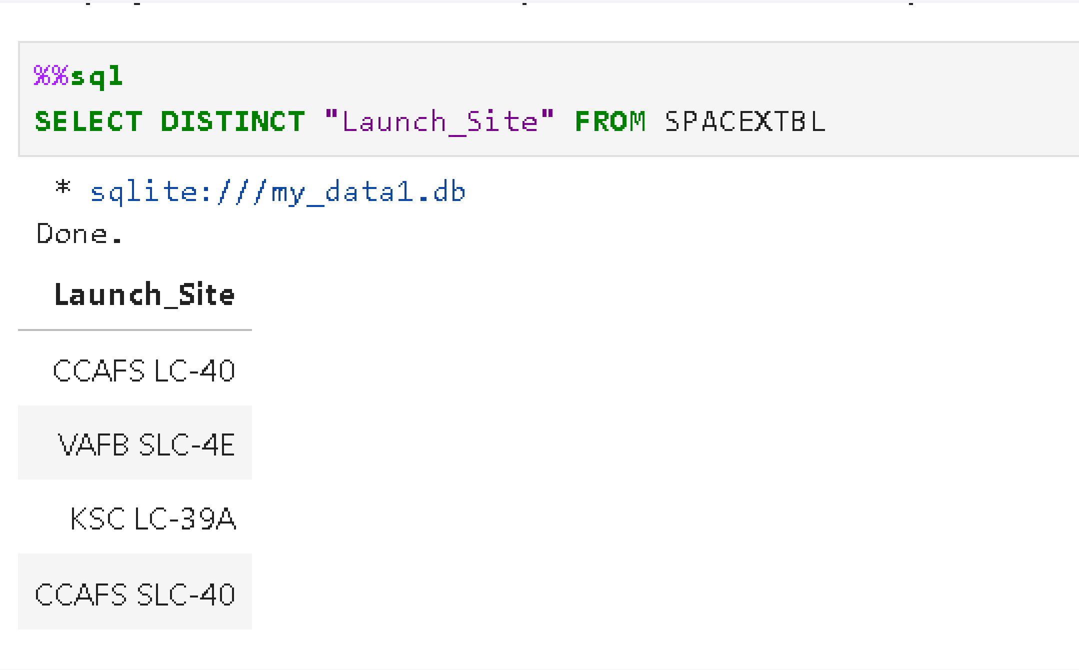


Launch Success Yearly Trend

- Here we see a line plot showing the trend between Launch Success vs. year of launch.
- The success rate has been increasing as the years go by, which is also confirmed by the flight numbers we saw earlier.
- Starting in 2013, there has been an increase until 2017. Then there was a drop in 2018, but a recovery in 2019, with a very small decrease in 2020.



All Launch Site Names



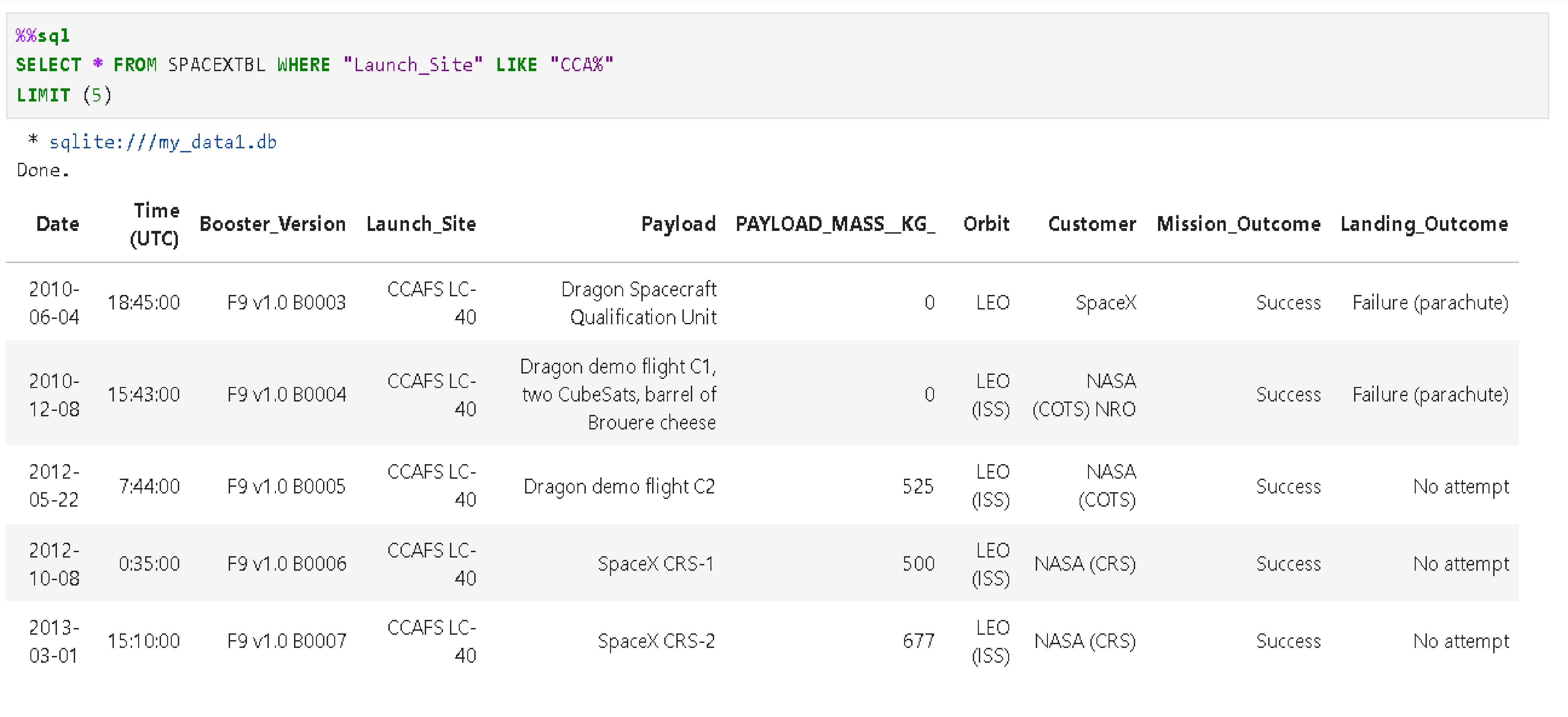
```
%%sql
SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
```

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

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

- Here, we see the four unique launch site names from our extracted data.
- While launches can come from multiple places, we will only be working with data from these 4, with 2 of them coming from the same facility: CCAFS.

Launch Site Names Begin with 'CCA'



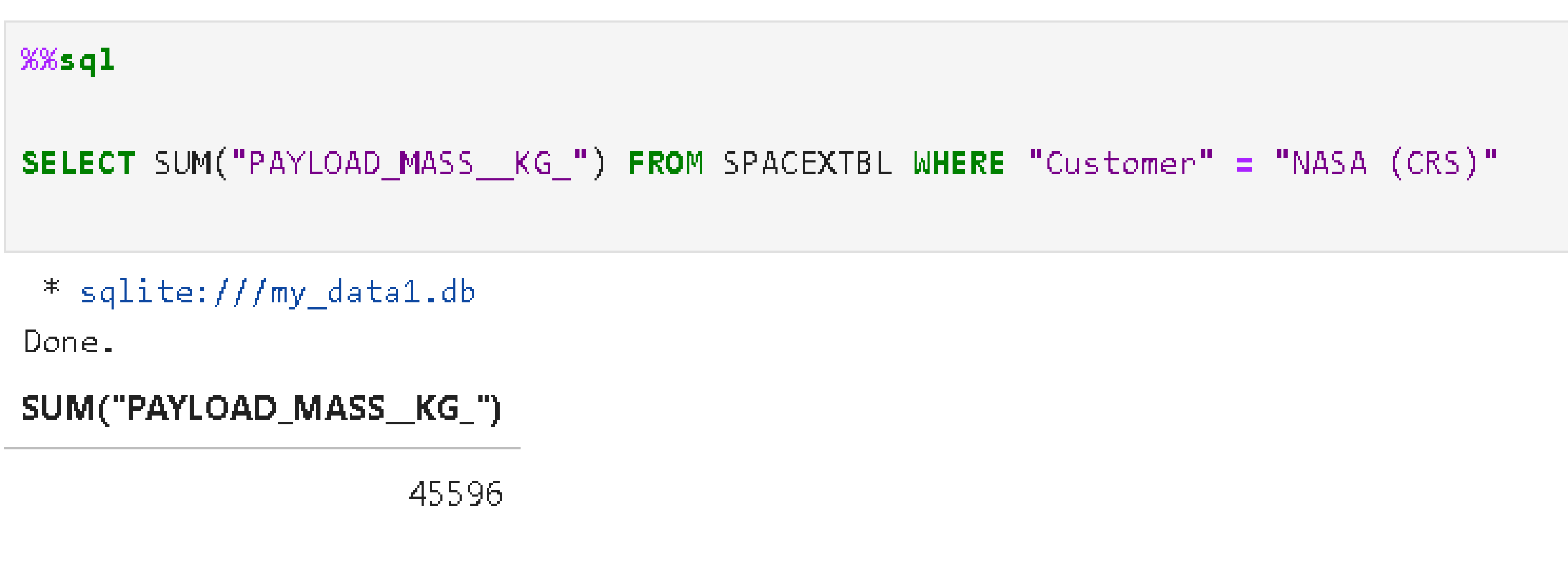
```
%%sql
SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE "CCA%"
LIMIT (5)
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Here, we can calculate the total payload carried by boosters from NASA.
- Filtering by customer, we can determine the total of their payloads with a sum.



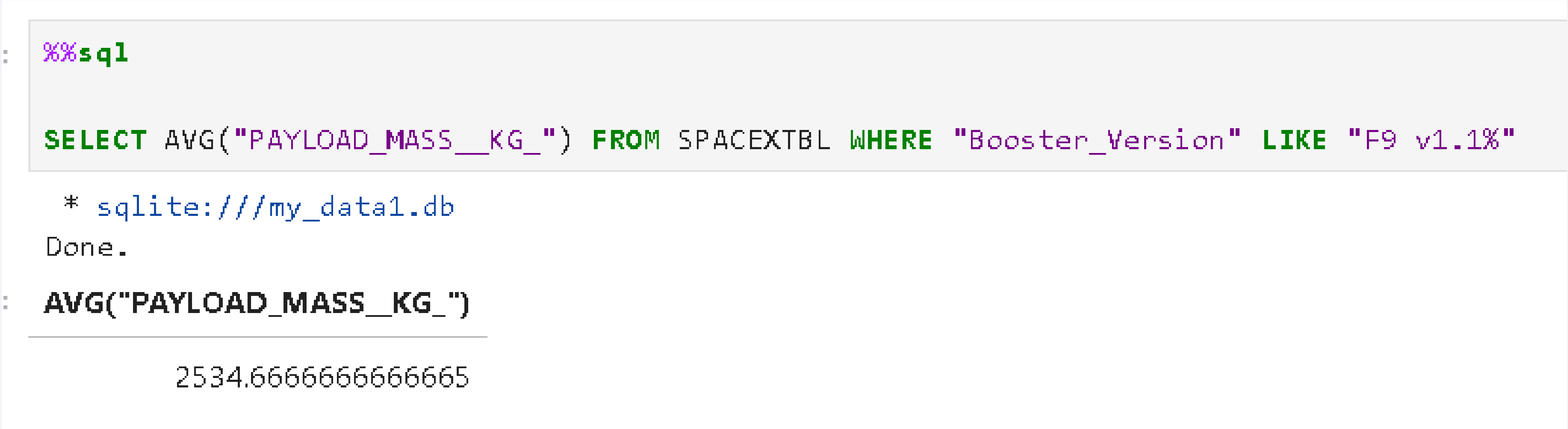
```
%%sql

SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Customer" = "NASA (CRS)"
```

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

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Here, we have calculated the average payload mass carried by booster version F9 v1.1.
- Seeing what is the expected average of particular booster versions can be helpful in determine if they will land successfully or not.



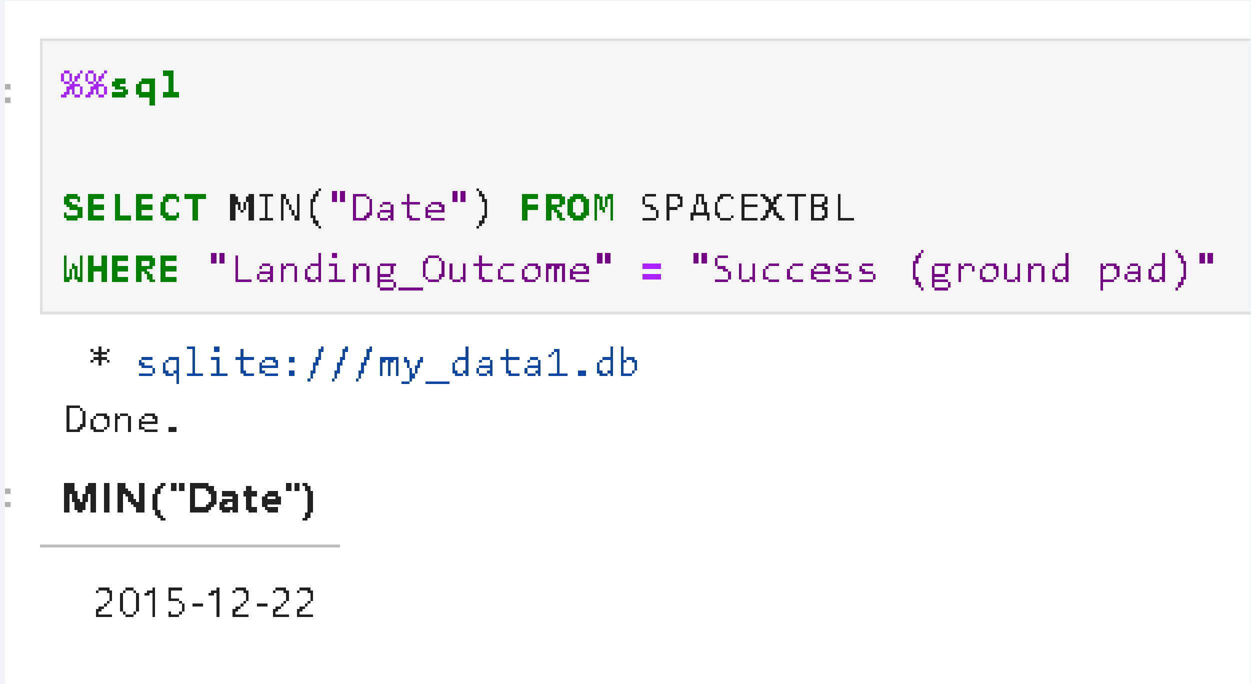
```
%%sql

SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" LIKE "F9 v1.1%"
```

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

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- The first successful Ground landing date happened on December 22, 2015.
- This aligns with the general trend of the data, where success started increasing in 2013.

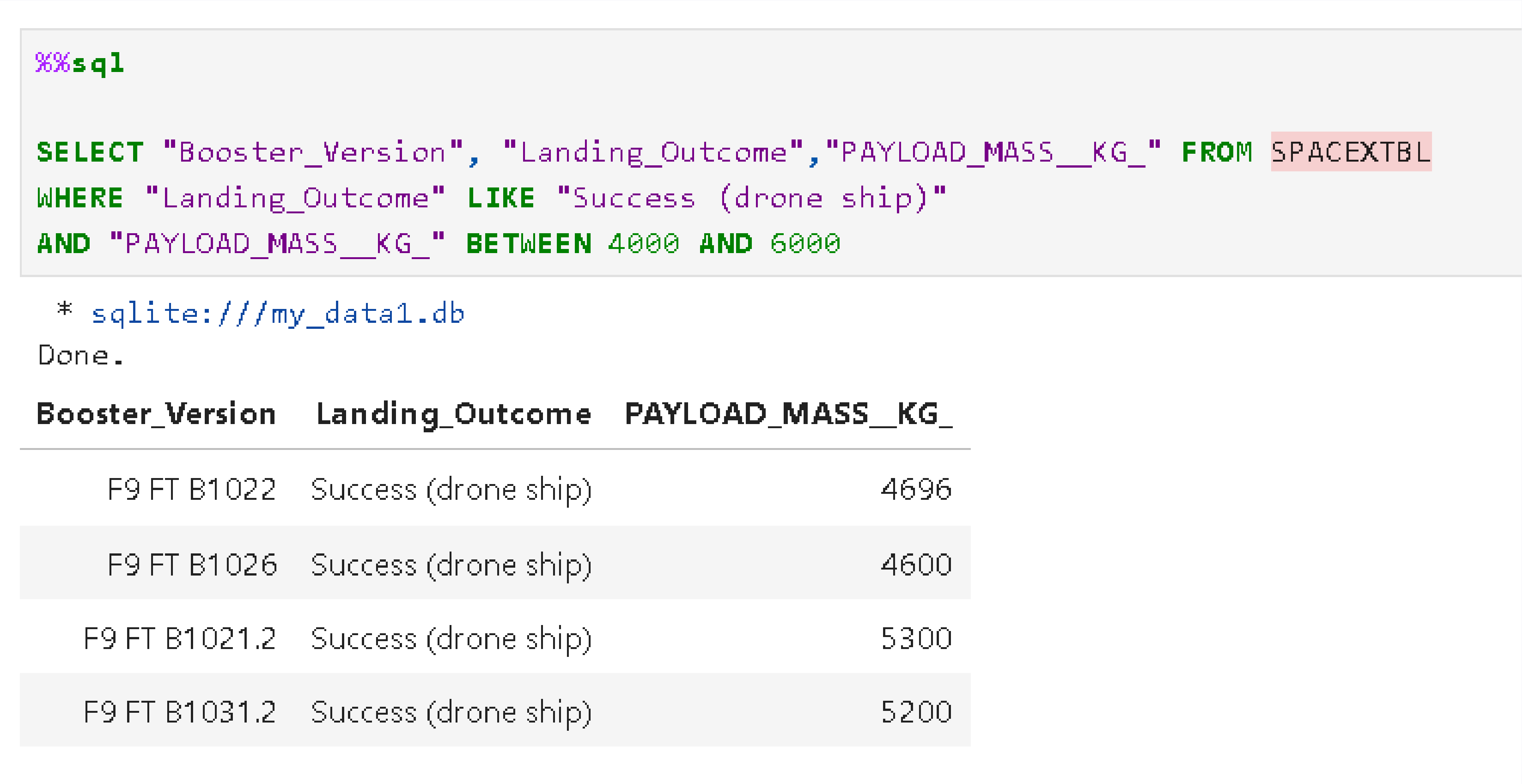


```
%%sql

SELECT MIN("Date") FROM SPACEXTBL
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



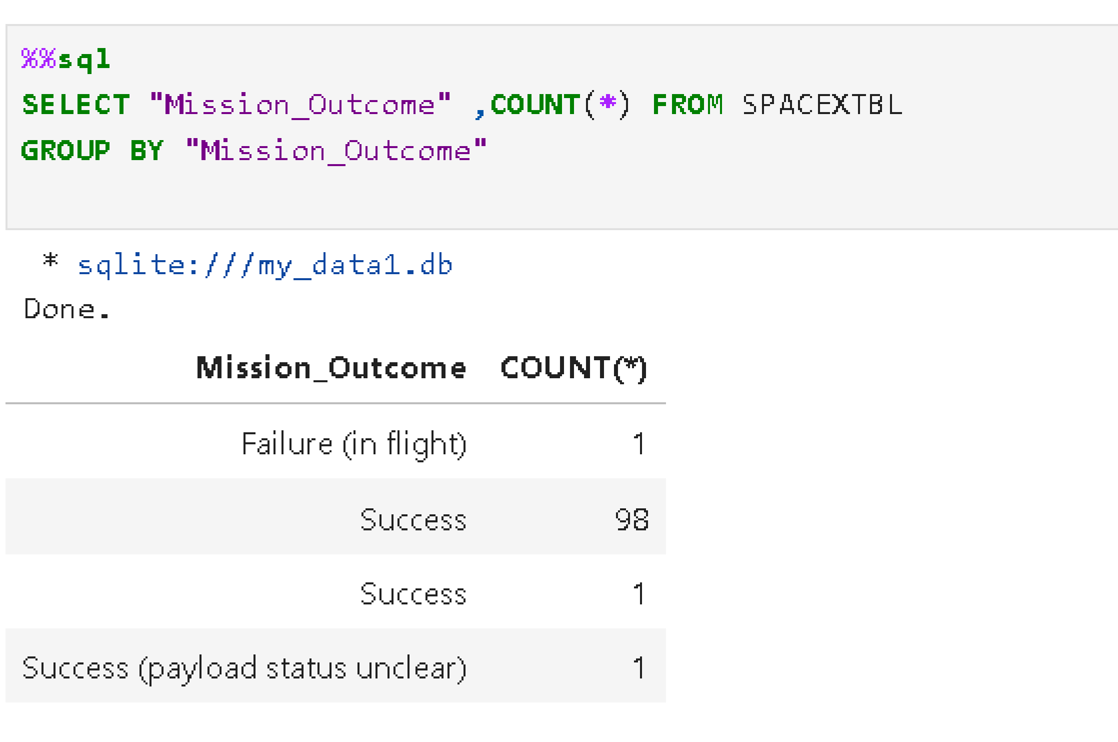
```
%%sql

SELECT "Booster_Version", "Landing_Outcome","PAYLOAD_MASS__KG_" FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE "Success (drone ship)"
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- Here, we can see the total number of successful and failure mission outcomes
- As we can see, most of the missions are successful, with only 1 failure. There seems to be no relationship in the overall mission success and if the landing of the component was successful.



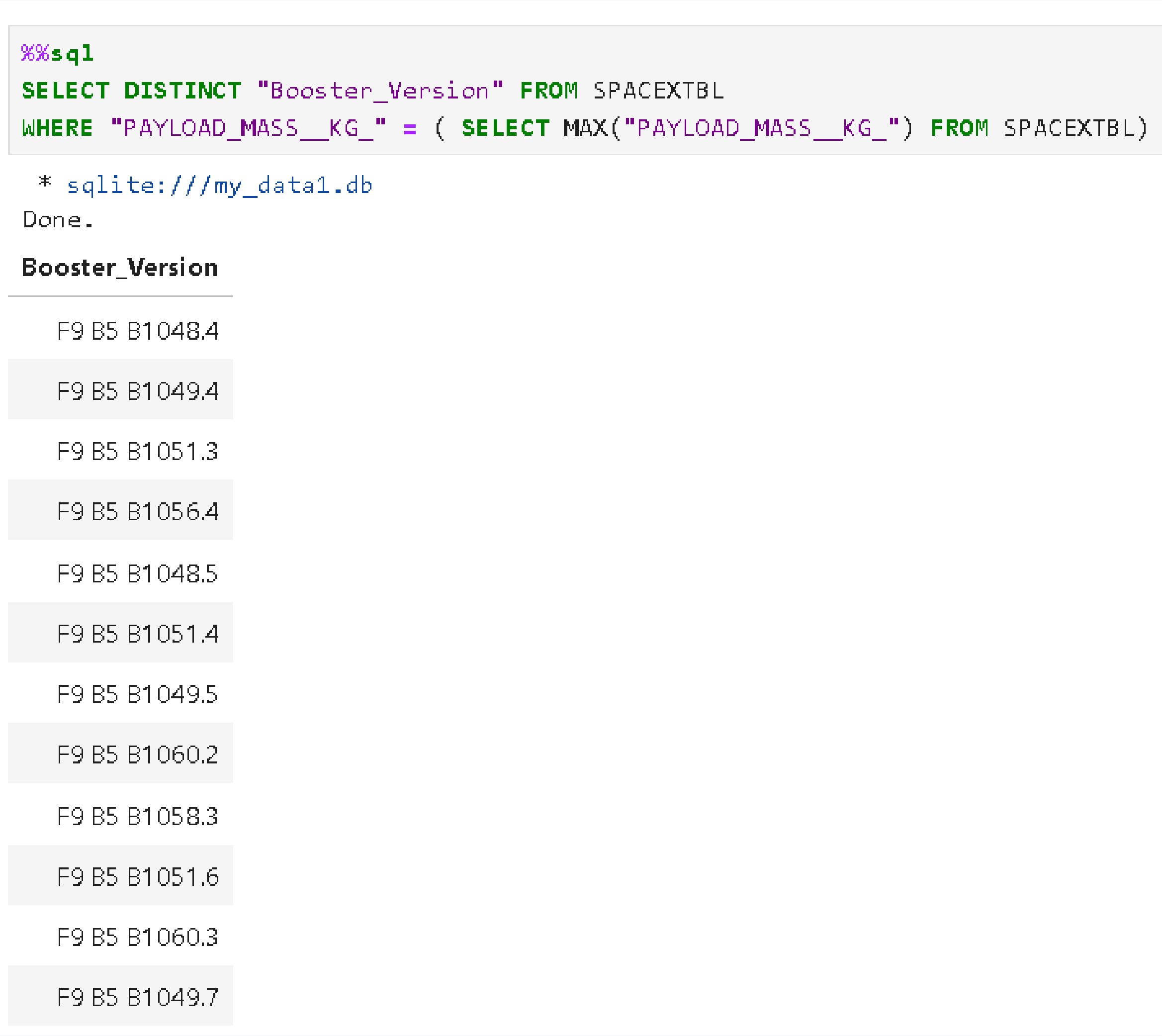
```
%%sql
SELECT "Mission_Outcome" ,COUNT(*) FROM SPACEXTBL
GROUP BY "Mission_Outcome"
```

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

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

Boosters Carried Maximum Payload

- Here, we can see the names of the booster versions which have carried the maximum payload mass.
- Multiple versions carry the maximum payload often.



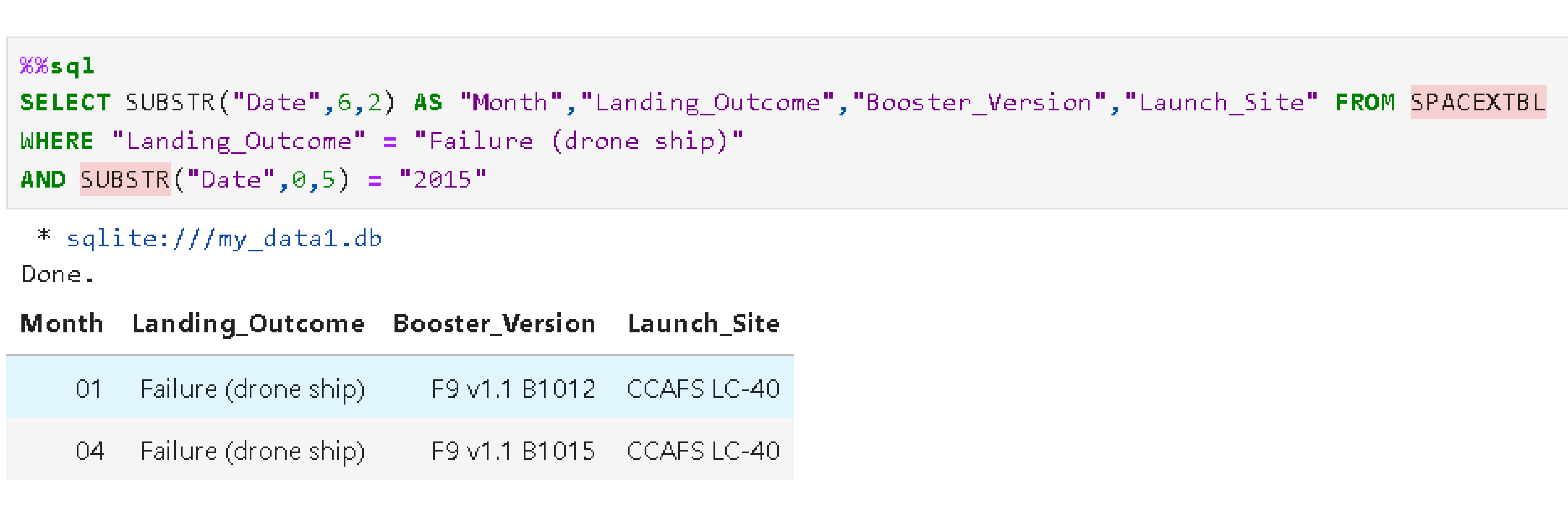
```
%%sql
SELECT DISTINCT "Booster_Version" FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = ( SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

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

- Here, we cans ee the failed landing outcomes in a drone ship, their booster versions, and launch site names for in year 2015.
- All of them came from CCAFS LC-40 site and had similar booster versions.



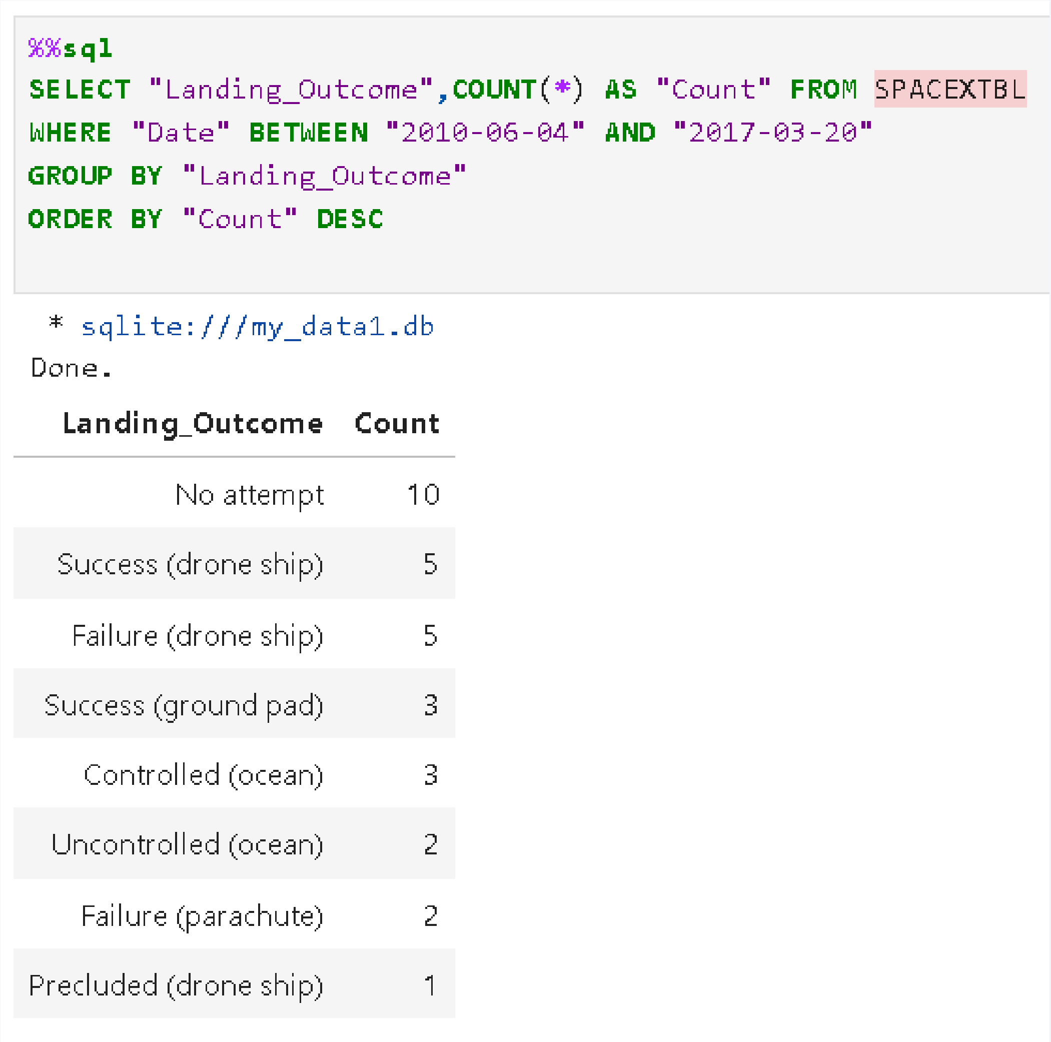
```
%%sql
SELECT SUBSTR("Date",6,2) AS "Month","Landing_Outcome","Booster_Version","Launch_Site" FROM SPACEXTBL
WHERE "Landing_Outcome" = "Failure (drone ship)"
AND SUBSTR("Date",0,5) = "2015"
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Here, we see the rank of the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.
- No attempts to recover were common, followed by successful and failed landings in drone ships.
- The least occurrent event was the Precluded (drone ship) one, with only one count in all the time frame of the query.



```
%%sql
SELECT "Landing_Outcome",COUNT(*) AS "Count" FROM SPACEXTBL
WHERE "Date" BETWEEN "2010-06-04" AND "2017-03-20"
GROUP BY "Landing_Outcome"
ORDER BY "Count" DESC
```

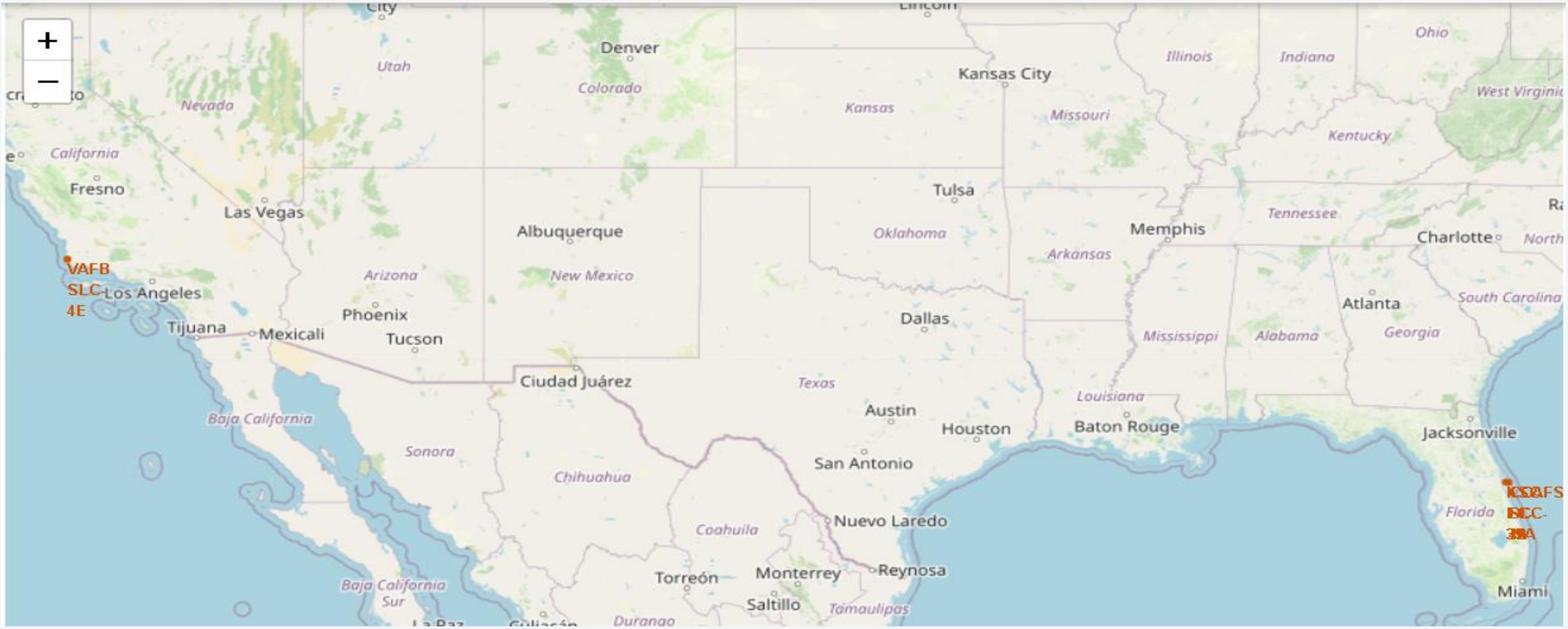
```
 * sqlite:///my_data1.db
Done.
```

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

Section 3

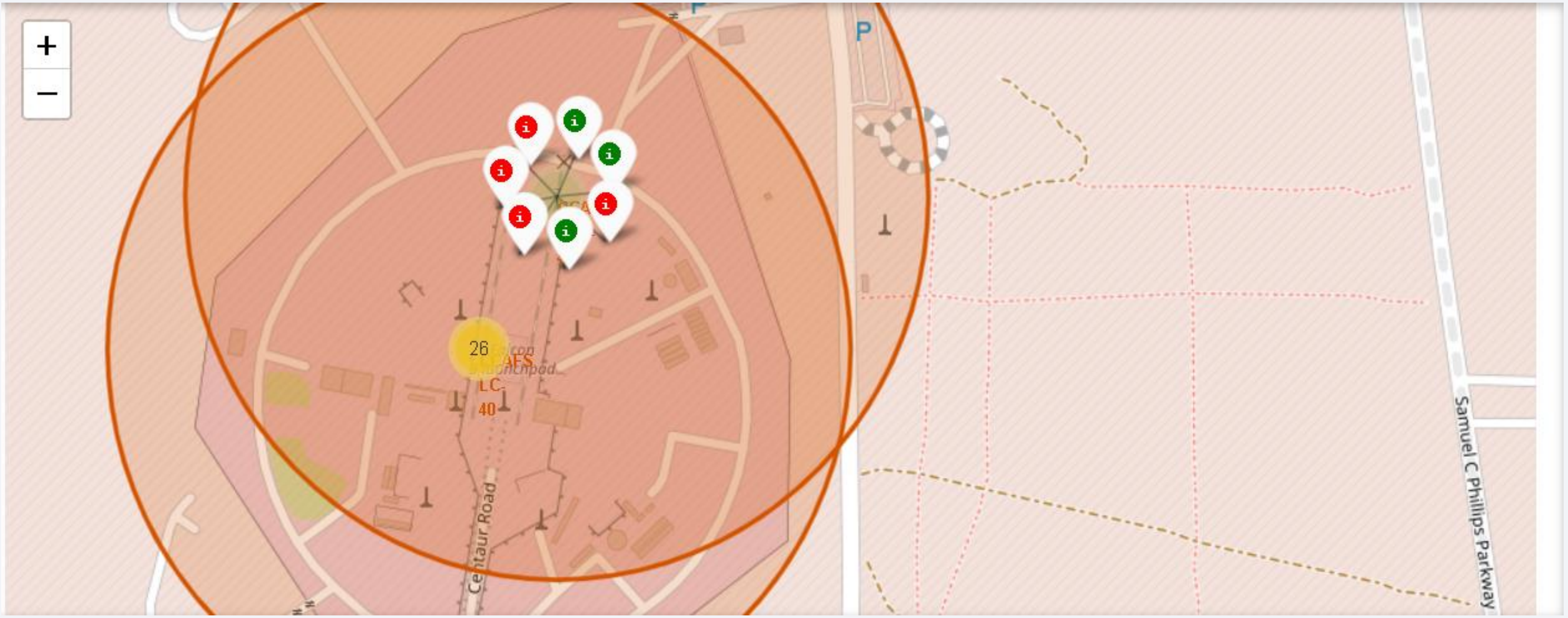
Launch Sites Proximities Analysis

Launch Sites Map: General Location



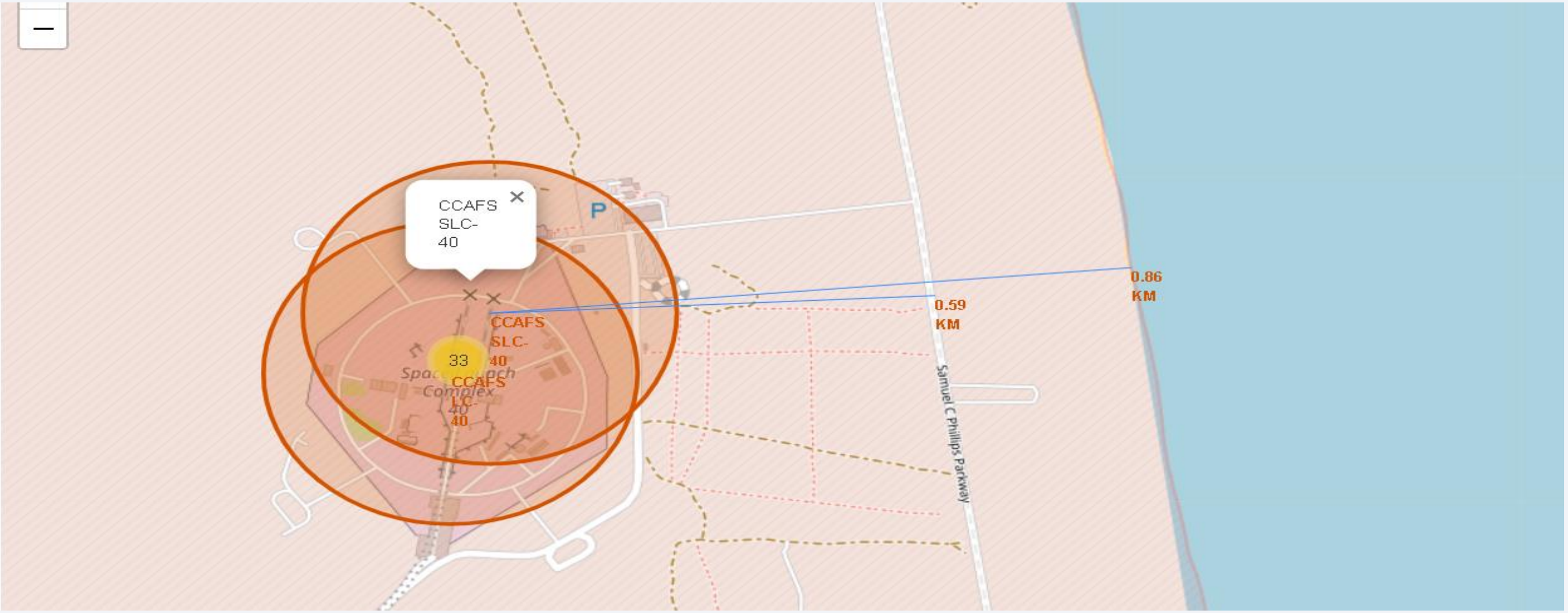
- We can see in the map the location markers of all launch sites' location markers on a global map.
- As seen, all of them are in the United States, specifically in the states of Florida and California.

Launch Sites Map: Success and Failure Ratios.

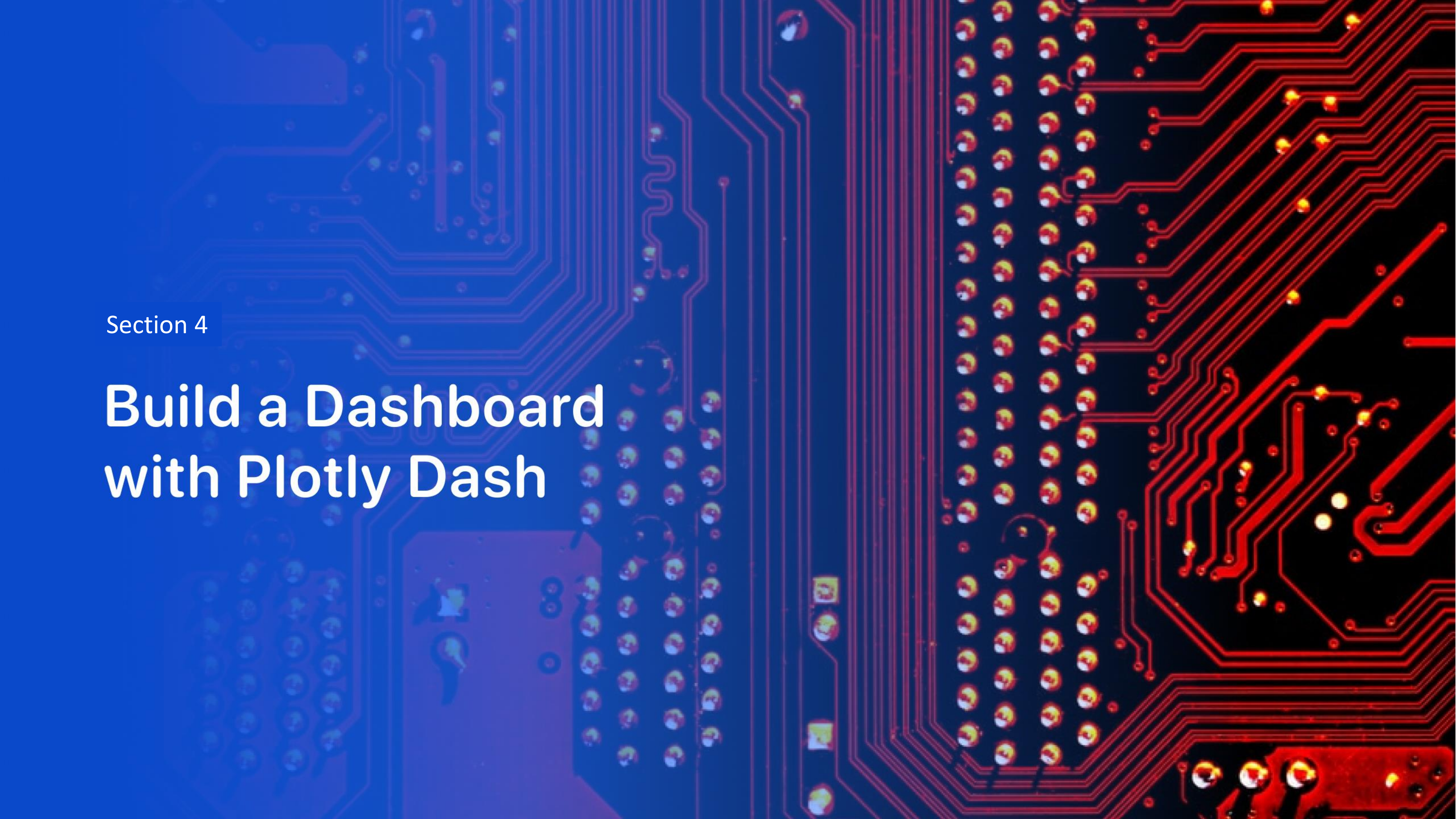


- In the map, we have also labelled what launches were successful and or failed per every launch site. Red corresponds to failure and Green to a success.
- As we see, while the ratio varies, all sites present both successful and failed recoveries.

Launch Sites: Geographical Occurrences.



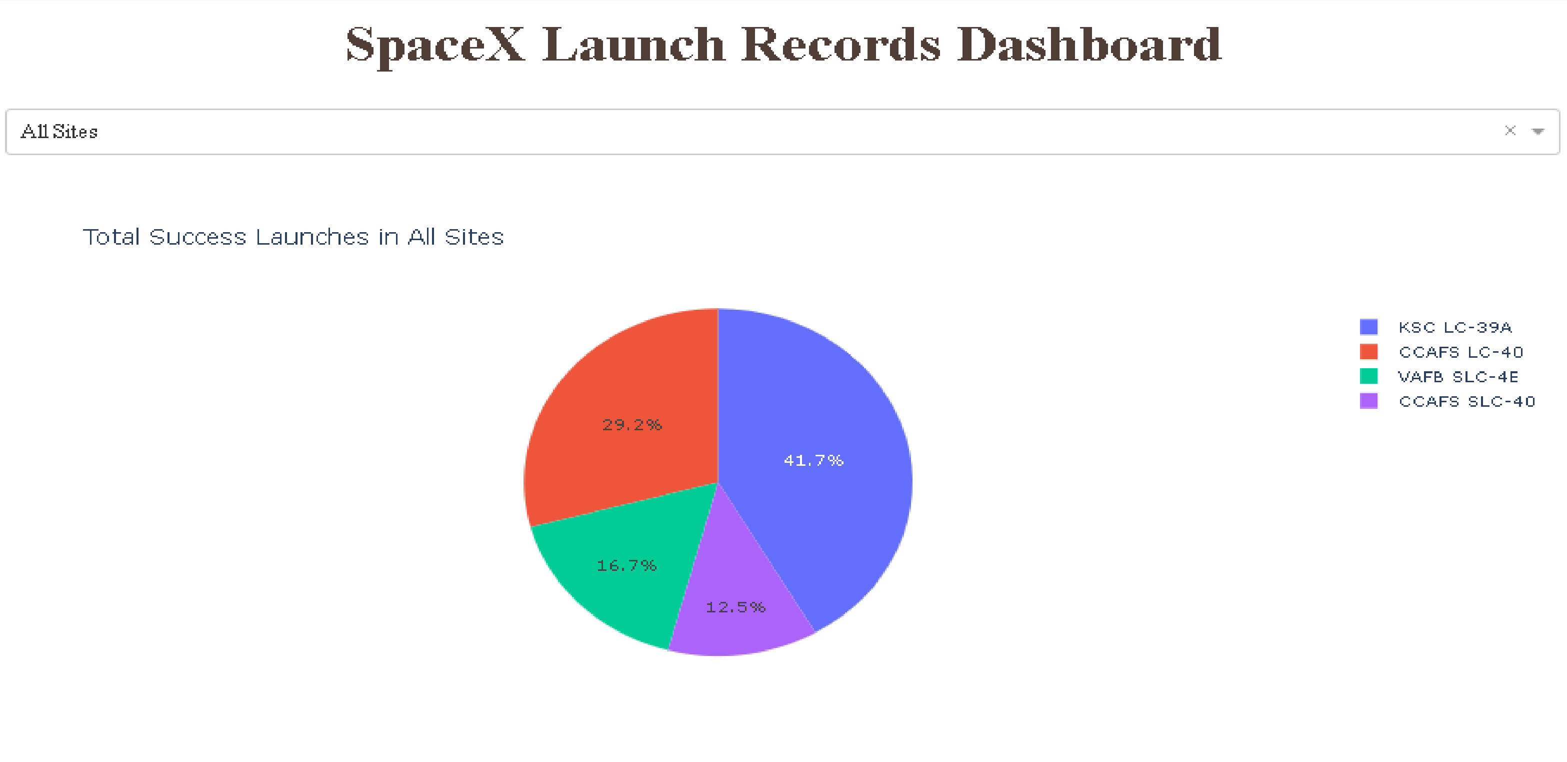
- In the map, some distances have also been calculated to show proximities of launch sites to other geographical points like railways, highways, or the coastline, with distance calculated and displayed
- As we can see, launch sites tend to be pretty close to the coastline and highways.



Section 4

Build a Dashboard with Plotly Dash

Pie Chart: Success of all Sites:

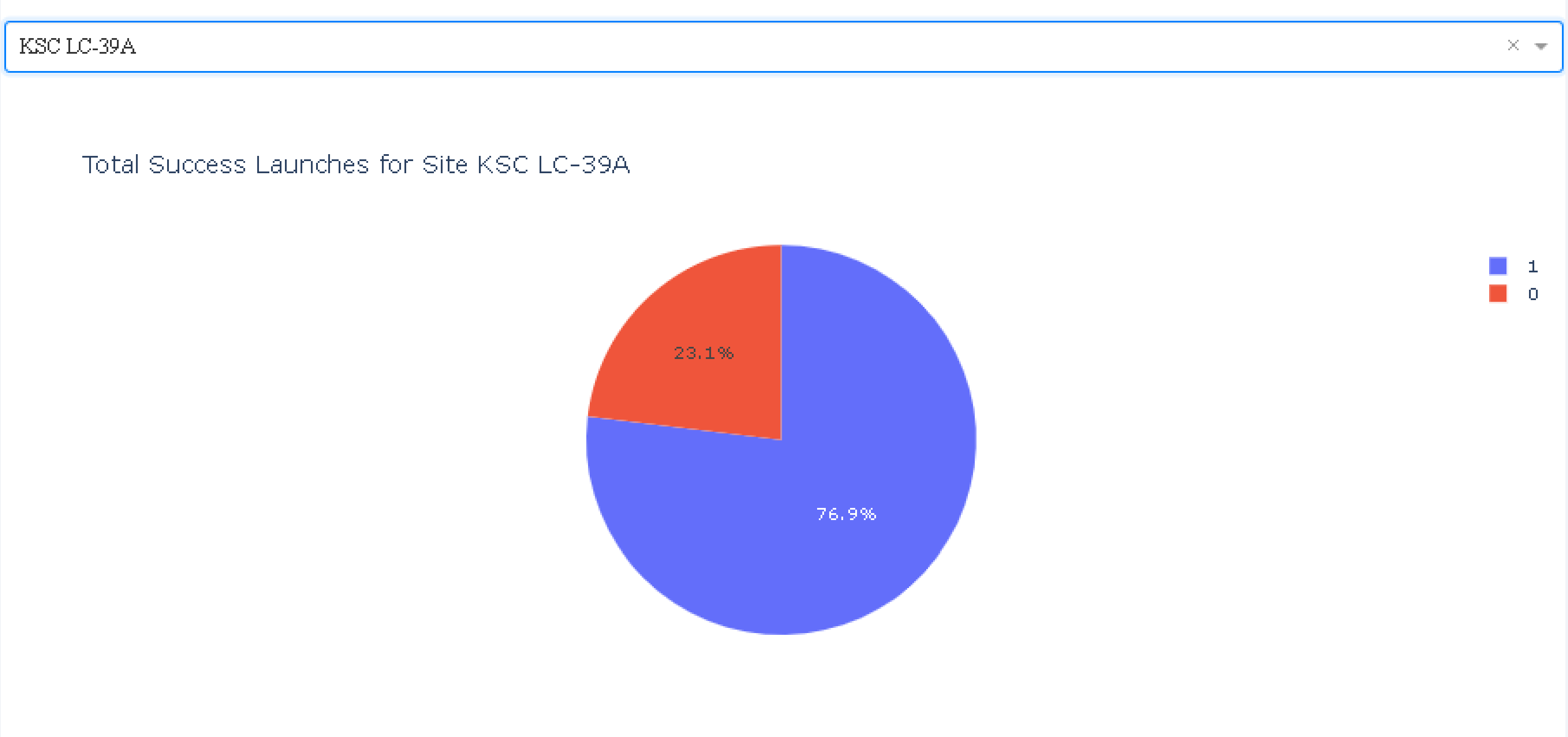


SpaceX Launch Records Dashboard

All Sites

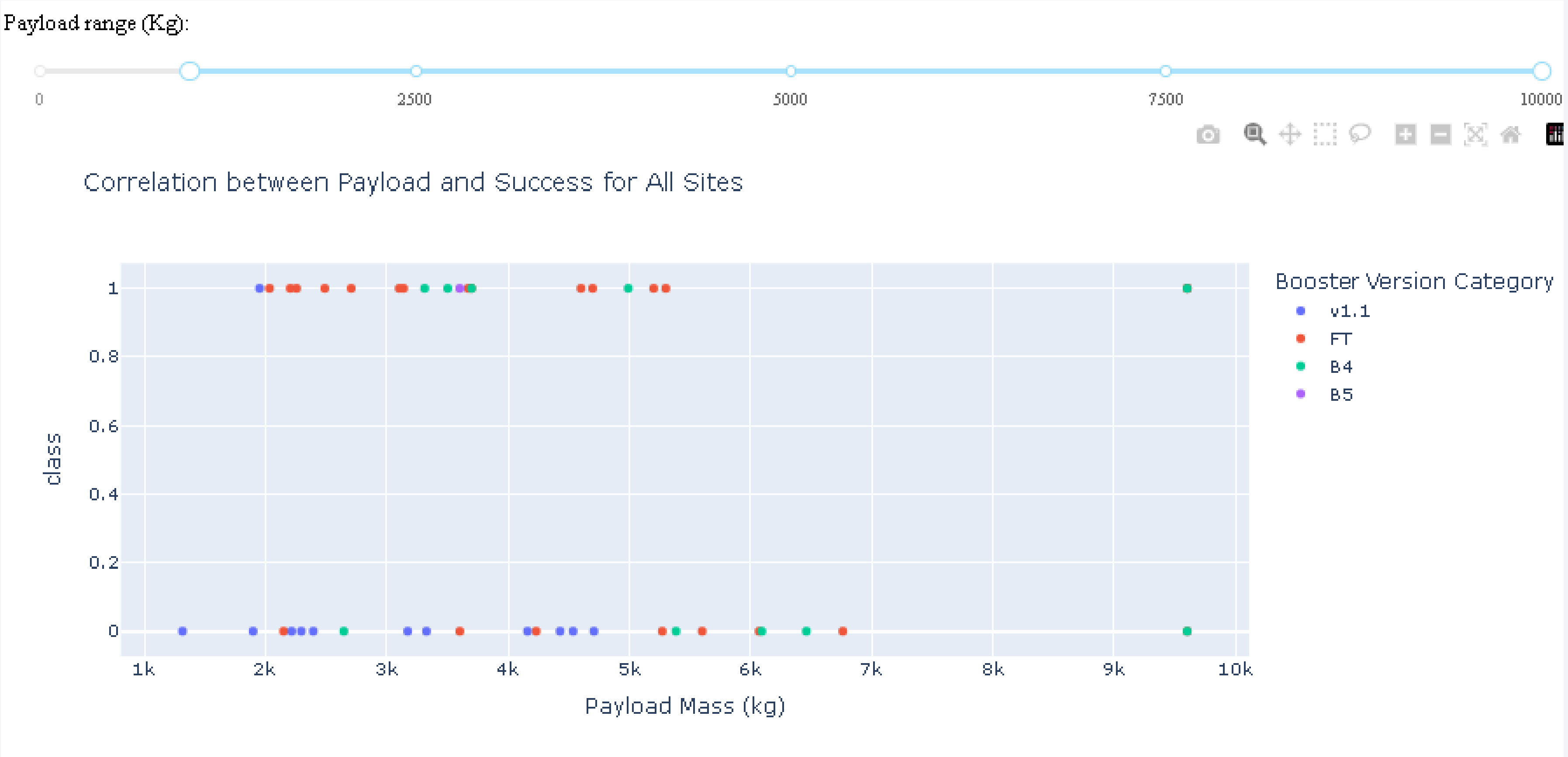
- As seen, the Launch Site with the most successful launches is KSC LC-39A, followed by CCAFS LC-40, VAFB SLC-4E and CCAFS SLC-40.
- None of them seems to overly dominate on successful launches, aside from the first.

Pie Chart: Success ratio of KSC LC-39 A



- Using the dashboard, we also see the success break down of the most successful launch site. Over 3/4ths of launches are successful here.

Relation between Payload and Booster Version

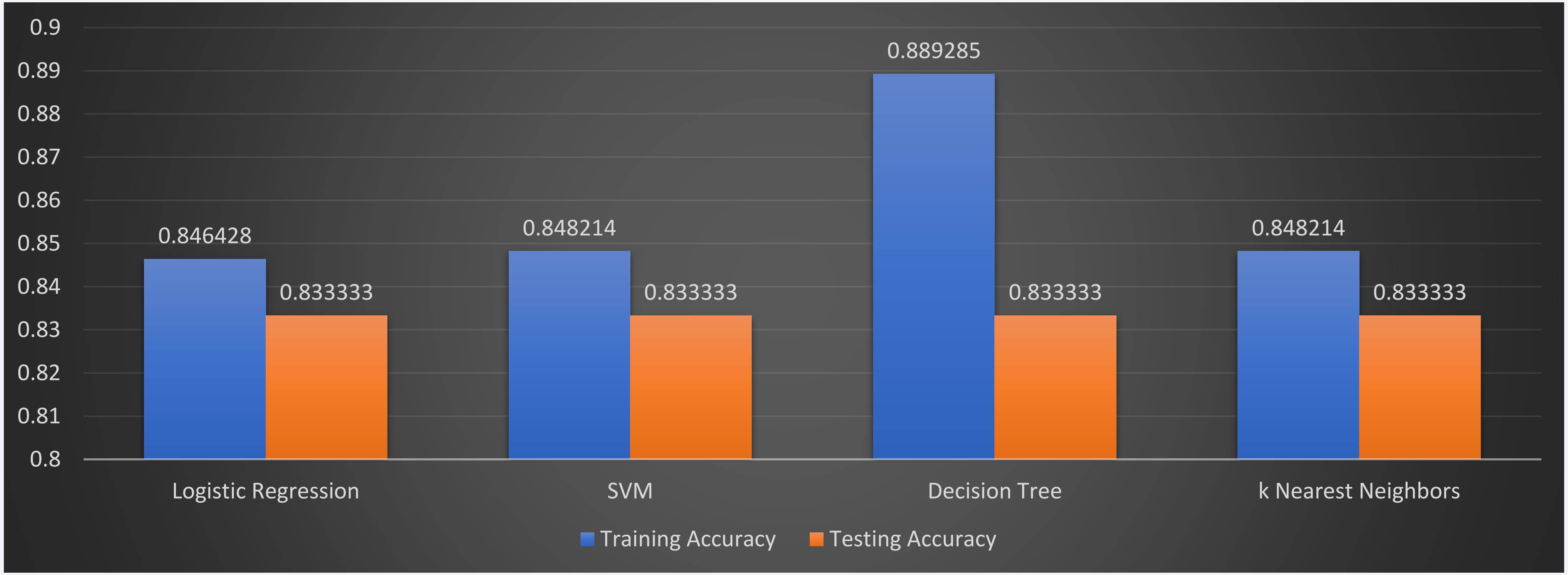


- As we can see, the FT Booster Version seems to have the most success in the selected payload ratio. V1.1 seems to have the least.

Section 5

Predictive Analysis (Classification)

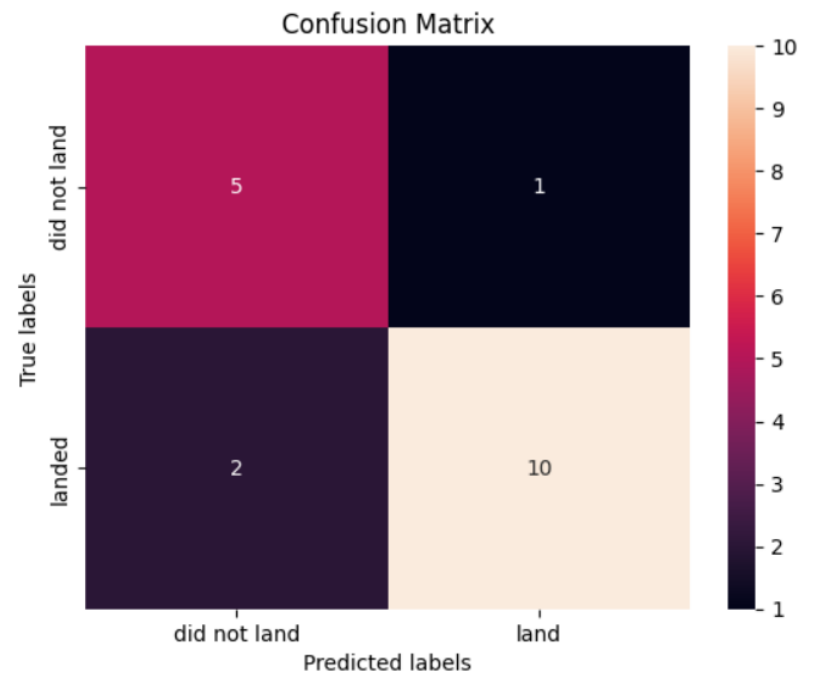
Classification Accuracy

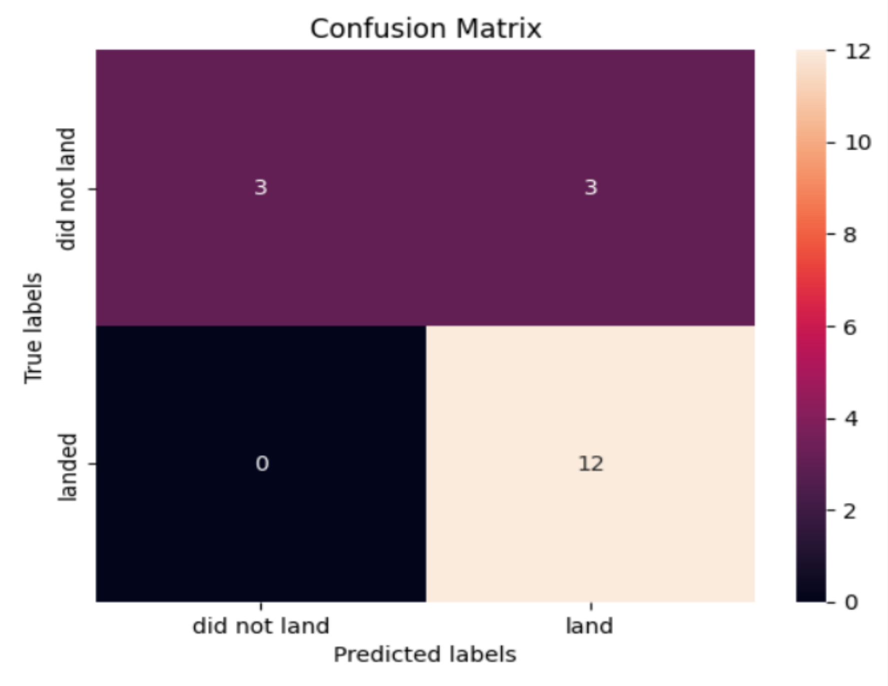


- Here we see a comparison of overall accuracy for all four trained models, both in training and testing phases.
- All of them have similar testing accuracies of 83.33%, but Decision Tree has better training accuracy (88.93%).

Confusion Matrix

- Here we see the Confusion Matrices of both Decision Tree and k Nearest Neighbors, which is the same as the other two.





- As seen, Decision Tree misses 2 successful landings, but the other models don't. It would be more costly to miss a successful land than prepare for recovery of something that will fail.

Conclusions

- Overall, it is possible to predict with a good success ratio if we will find a successful landing or not, which makes this viable for future analysis.
- All the visualization and data analysis were very crucial in determining the most impactful factors for successful landings.
- All techniques present an overall accuracy of 83.33%. We recommend using any of the three with 100% success in labelling a successful landing.
- For future training, Logistic Regression or KNN could be more favorable since they trained faster than SVM.
- We could also try to determine if other geographical features or even weather and seasonal conditions could factor in the recovery. This will be explored in a future iteration of this project.

Appendix

- For any inquiries about this project, one can reach to the author via email:
  - krodsiu@umich.edu
- All relevant information like code, datasets and this report can be found at the repository:
  - https://github.com/thekcrs/DataScienceCapstone/
- Don't hesitate in reaching out if you have any suggestions or just want to talk about ideas. Let's keep learning together! :)

Thank you!