Winning Space Race with Data Science

Danail Arabadzhiev
October 9, 2024

Outline

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

Executive Summary

- The project involves collecting data using SpaceX's API and web scraping Wikipedia, followed by data wrangling techniques like handling missing values and encoding variables. Exploratory data analysis (EDA) and visual analytics are performed using visualization tools, SQL, Folium, and Plotly Dash, leading to predictive analysis through classification models, including model building, tuning, and evaluation.
- Picking the optimal orbit, payload mass and launch site, as well as the advancement of technology can greatly affect the result of a landing.

Introduction

- The Falcon 9 rocket is designed to have its first stage return and land safely after launch, allowing it to be reused for future missions. This reusability is a key factor in reducing costs, and accurately predicting whether the first stage will land successfully is crucial for planning and budgeting.
- By leveraging historical data, the project aims to build a predictive model that can estimate whether a Falcon 9 first stage will successfully land. This could help SpaceX optimize its launch processes.
- Key questions that we want to see answered during our project are the following:
- What factors influence a successful landing?
- What is the probability of a successful landing under different conditions?
- Can the success rate be improved by adjusting certain parameters?
- How accurate is the current prediction model for Falcon 9 landings?

Section 1

Methodology

Methodology

Executive Summary

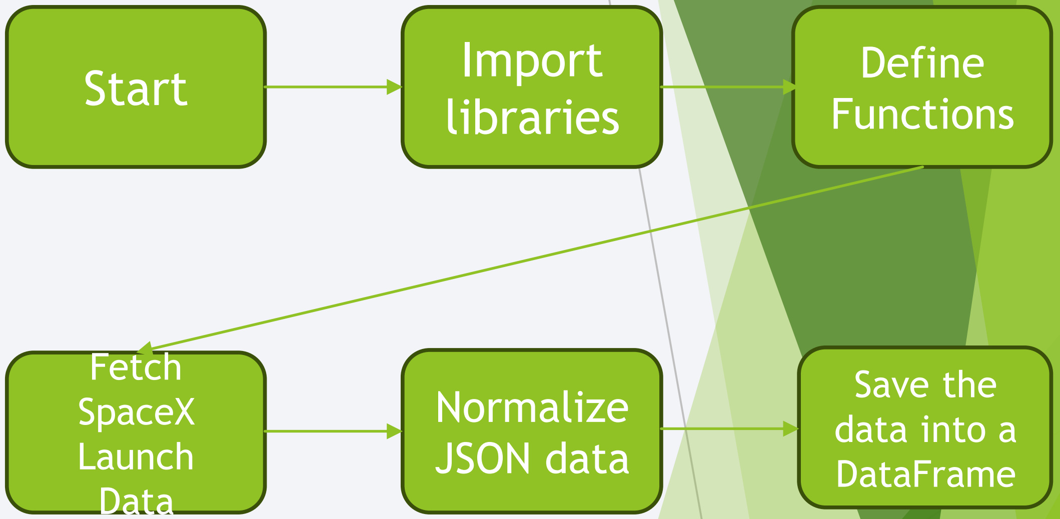
- Data collection methodology:
  - SpaceX's API was used together with the webscraping of a Wikipedia page about the Falcon rockets.
- Perform data wrangling
  - A few techniques were use such as handling missing values, filtering rows, creating new variables, encoding categorical variables, data type conversions and more.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was sourced from SpaceX’s API and Wikipedia.
- SpaceX API: Using Python’s requests library, I fetched the API URL, extracted a JSON file with .json(), and used pandas.json_normalize to convert it into a DataFrame.
- Wikipedia Scraping: I used BeautifulSoup and requests to access and parse the relevant Wikipedia page. By locating the specific table, I extracted and cleaned the required data.

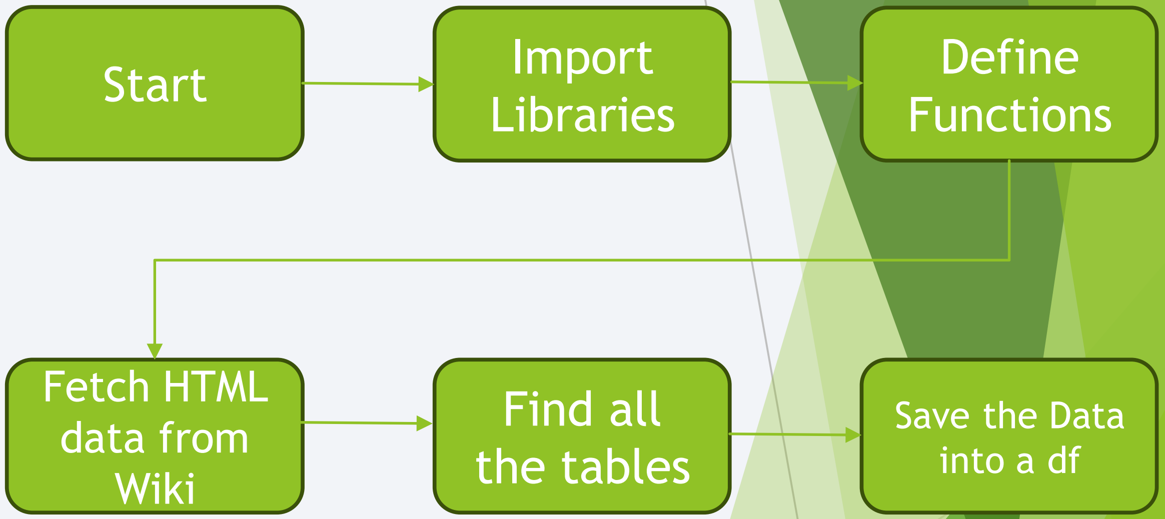
Data Collection – SpaceX API

- Firstly we import a few important libraries like “requests” and “pandas”. I had a few given functions that loop through a few of the APIs pages. After that we fetch the data with the requests library and the get function. Then we normalize the data with pd.json_normalize and save it into a variable as a DataFrame object.
- Here is a GitHub link where all the files on this assignment are situated: https://github.com/Danail-A/capstone_project
- For this you can use jupyter-labs-spacex—data-collection-api.ipynb file.



Data Collection - Scraping

- Firstly I import a few important libraries like requests, pandas, and bs4 for the BeautifulSoup object. Again here the main functions were defined. I again use the requests library and the get function to get the data from the URL given. I use the BeautifulSoup with the text of the URL to get the information from the website. With this I can find all the tables with {html_tables = list(soup.find_all('table’))}. After finding the needed table I fetch the data from it and save it into a dataframe (more on this in the data wrangling section).
- You can use the jupyter-labs-webscraping.ipynb file


Start
Import Libraries
Define Functions
Fetch HTML data from Wiki
Find all the tables
Save the Data into a df

Data Wrangling – SpaceX API

- After loading the data into a DataFrame from the API:
- Trimming Columns: Kept only “rocket,” “payloads,” “launchpad,” “cores,” “flight_number,” and “date_utc.” Rows with multiple cores/payloads were removed, and “date_utc” was converted to datetime format.
- Filtering Data: Extracted information for Falcon 9 boosters, including Booster Version, Payload Mass, Longitude, Latitude, and Outcome.
- Handling Missing Values: Checked for missing data, which was found only in the Payload Mass column; filled these using the mean of other entries.

Data Wrangling - Scraping

- After finding which table to use for the WebScraping:
- **Identify Table Headers:** Located the relevant table and stored its headers using *first_launch_table.find_all('th').* Used the *extract_column_from_header* function to gather all necessary columns.
- **Create Data Dictionary:** Initialized a dictionary with column names as keys and empty lists as values.
- **Iterate Through Table Rows:** Used a predefined function to loop through each row in the table ("wikitable plainrowheaders collapsible"). Verified if the row's header contained a flight number and if it was a digit to confirm it as a valid record.
- **Extract and Store Data:** Extracted relevant information for valid records and assigned it to the correct keys in the dictionary. After the process was complete it was loaded into a DataFrame.

EDA with Data Visualization

- The main charts used in this section were created in order to see the relationship between Flight Number, Payload Mass, Launch Site, Orbit and Year with Success. The reason they were created was in order to find visible correlation between the above mentioned variables.
- edadataviz.ipynb is the file where you can in detail see the specific charts created.

EDA with SQL

- Display the names of the unique launch sites in the space mission – pretty explanatory. It is really needed to know all the launch sites as it might be a major factor.
- Find the payload mass carried by boosters launched by NASA
- Find the first date when there was a successful landing
- Find the total number of successful and unsuccessful mission outcomes
- List the names of the booster versions which have carried the maximum payload mass
- Jupyter-labs-eda-sql-coursera_sqlite.ipynb is the file where you can see my SQL queries.

Build an Interactive Map with Folium

- The objects used in the creation of the map were Map, Markers, MarkerClusters, Circles, PolyLines, MousePosition and Icons.
- Map was used to initiate the map object where everything else was later added on. The reason markers were created at the beginning was to mark a point on the map where there was a launch site. Later they were used to pin-point the shore, town, railroad and highways with the help of the MousePosition indicator. MarkerClusters were used as there were a lot of markers with the same location and it made it hard to see how many of them were successful or not. Quite a useful tool. PolyLines were used to connect two points. Icons were used just for visuals. Circle was used around the marker points to visually create an area around them.
- You can use the lab_jupyter_launch_site_location.ipynb file for more visuals and info.

Build a Dashboard with Plotly Dash

- In this project I have added a dropdown menu and a range slider in order to make the charts interactive. With the dropdown menu you can select a specific launch site and see what is the percentage of successful launches from this specific site in the form of a pie chart. The range slider has a range of minimum of payload to maximum of payload and when it moves the graph below, which is a scatter plot that shows the successful launches per selected side, moves with it.
- These interactions were added because it can easily show the ways these two things interact with the success rate and visually is the easiest way for this information to be understood.
- spacex_dash_app.py is the file where you can check this.

Predictive Analysis (Classification)

- The way the machine learning models were build is pretty straightforward. As in the previous steps there was a normalization step where I used one-hot encoding to convert the categorical data into numeric so the data was ready to be fed to the models. The models used are Logistic regression, Support Vector Machines, Decission tree, and KNN as they are good for classification prediction.
- The way the models were build was with GridSearchCV. The data set was preprocessed using *preprocessing.StandardScaler() and fit_transform.* Then it was split into training and testing with train_test_split with test_size=0.2. GridSearchCV was used to calculate the best values for the parameters which we use from a predefined list of parameters. Then the train data is fit to the best parameters in order to provide a prediction on the test data. Weirdly all the models give the same predictions with accuracy of around 83% with Decision Trees providing a best score of 90.18% on the training data.
- SpaceX_Machine Learning Prediction_Part_5.ipynb is the file for this

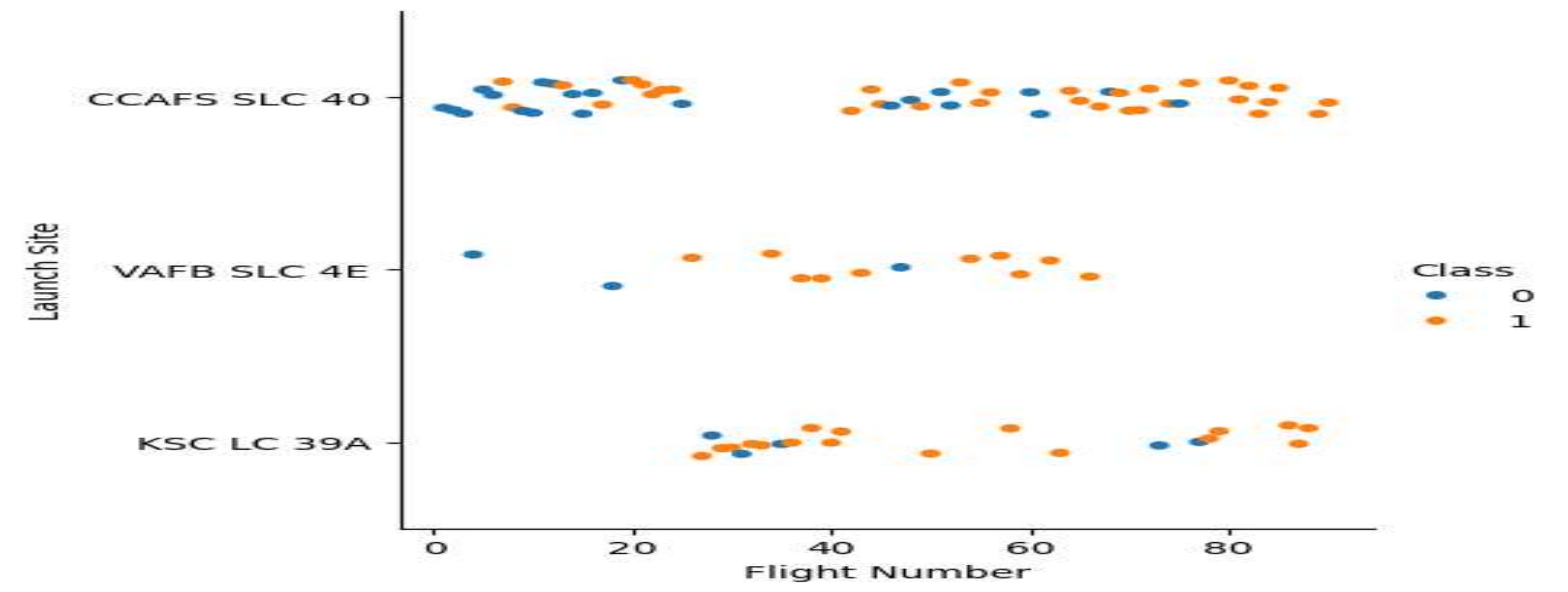
Results

- One of the important conclusion that could be seen from the EDA is that with the larger the Payload Mass and the larger the number of flights, the better the chance of a successful landing. Later we can see that for some orbits they Payload mass is an important factor as their success landings are getting more and more prominent with larger payload mass. Also with years success rate tends to get much better especially since 2013.
- All the methods used bring the same result being 83% on the test data. Decision trees bring the best result at 90% for the training data.

Section 2

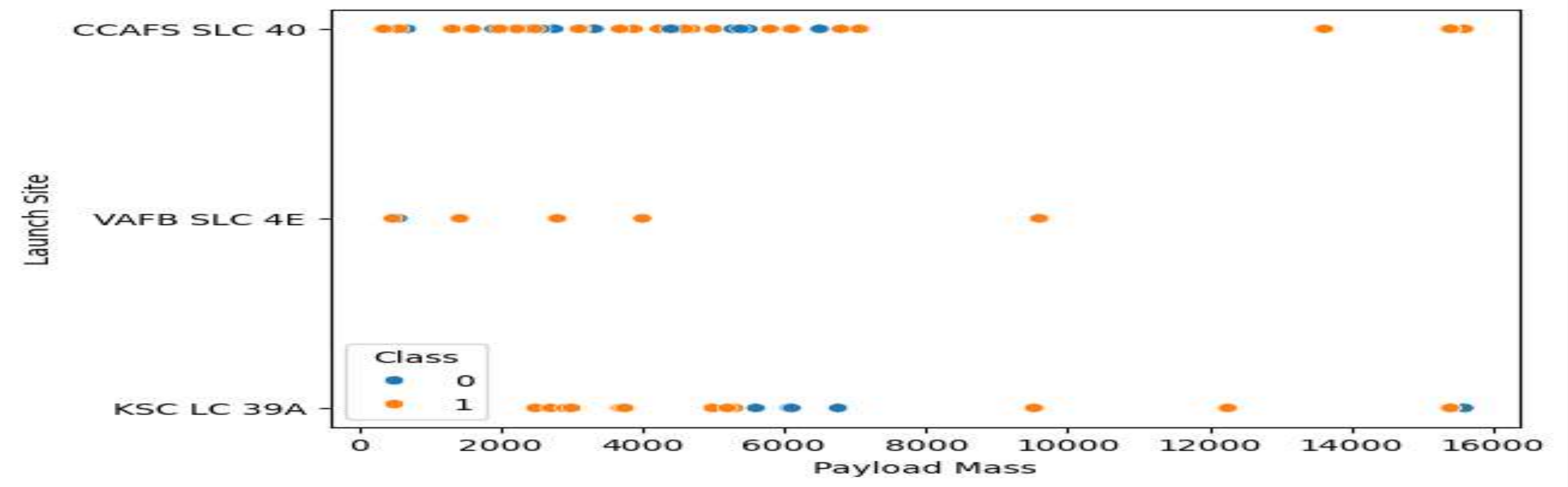
Insights drawn from EDA

Flight Number vs. Launch Site



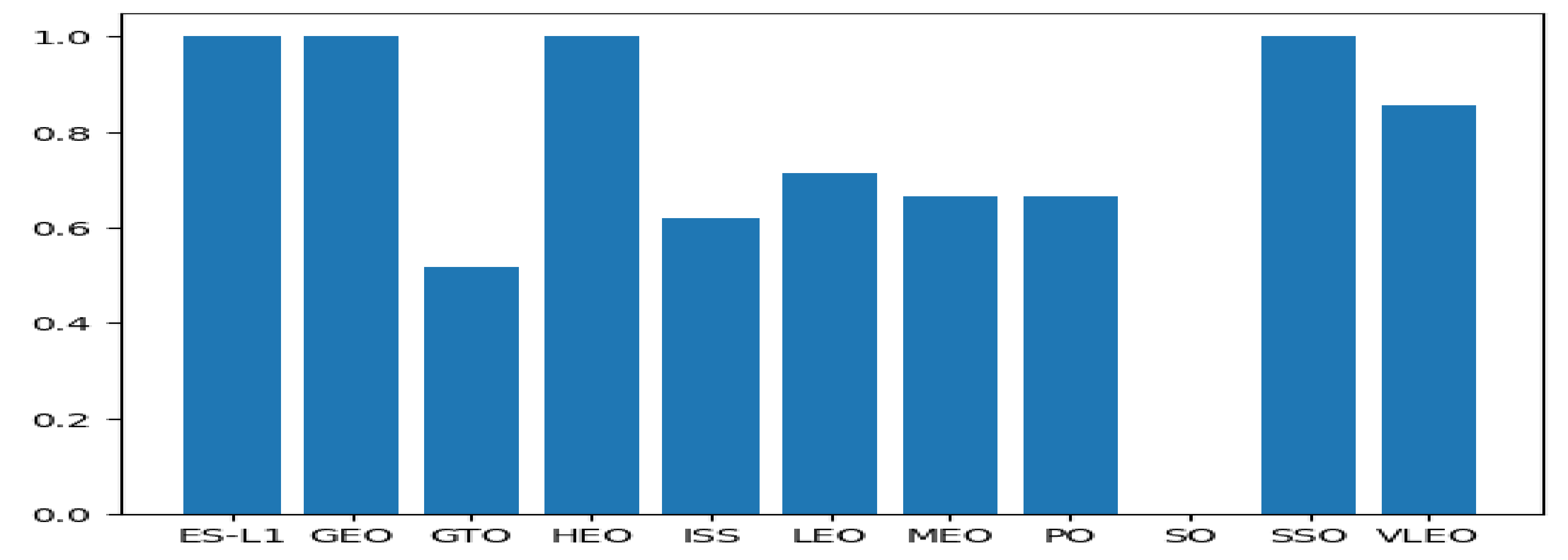
- From the above scatter plot we can clearly see that the higher the flight number the better the success rate for each site. We can conclude that there is a positive correlation between the two.

Payload vs. Launch Site



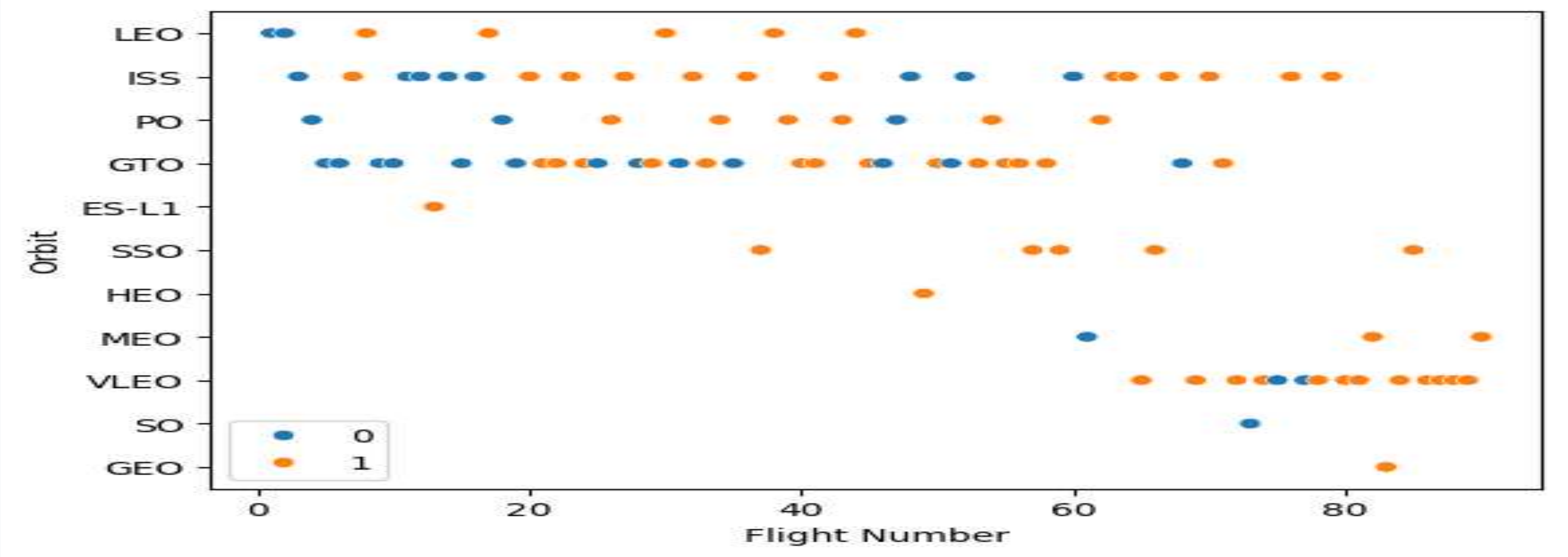
- What we can see is that for the CCAFS SLC 40 launch site the higher the Payload mass the more successful the landings. However, we cannot say the same for the other sites as we do not have a clear indications based on the few samples.

Success Rate vs. Orbit Type



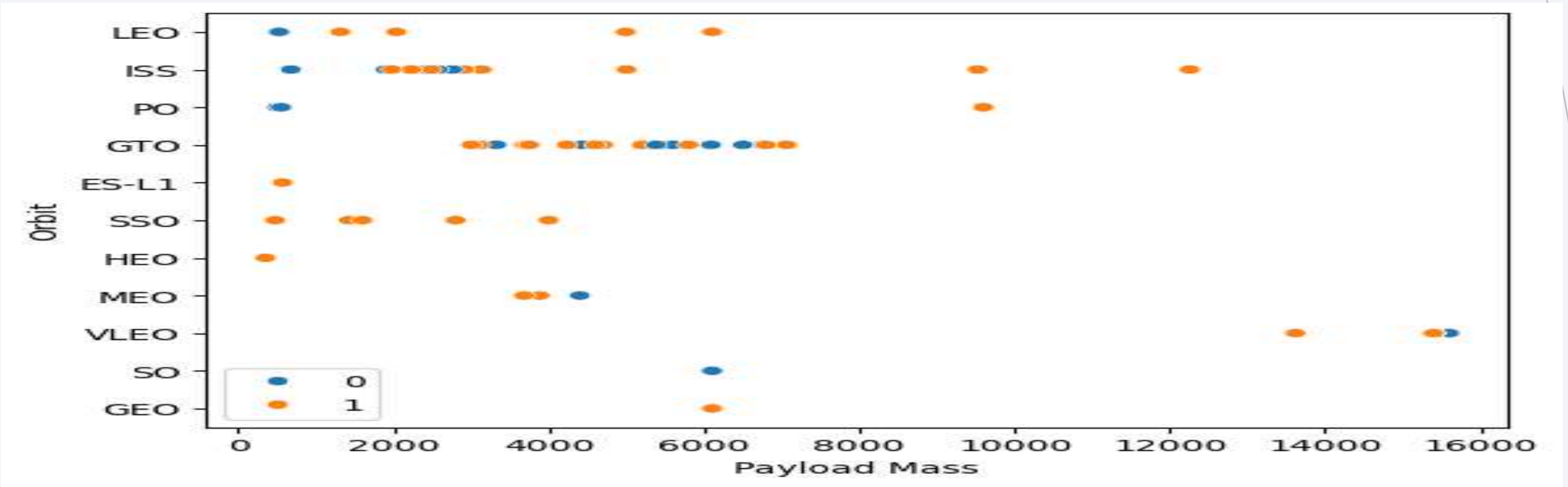
- We can observe that for some orbits like ES-L1, GEO, HEO and SSO have a success rate close to 1. Just from this information there isn’t much more to be drawn. Later we can see that for some of them we just have too little data (like 1 sample).

Flight Number vs. Orbit Type



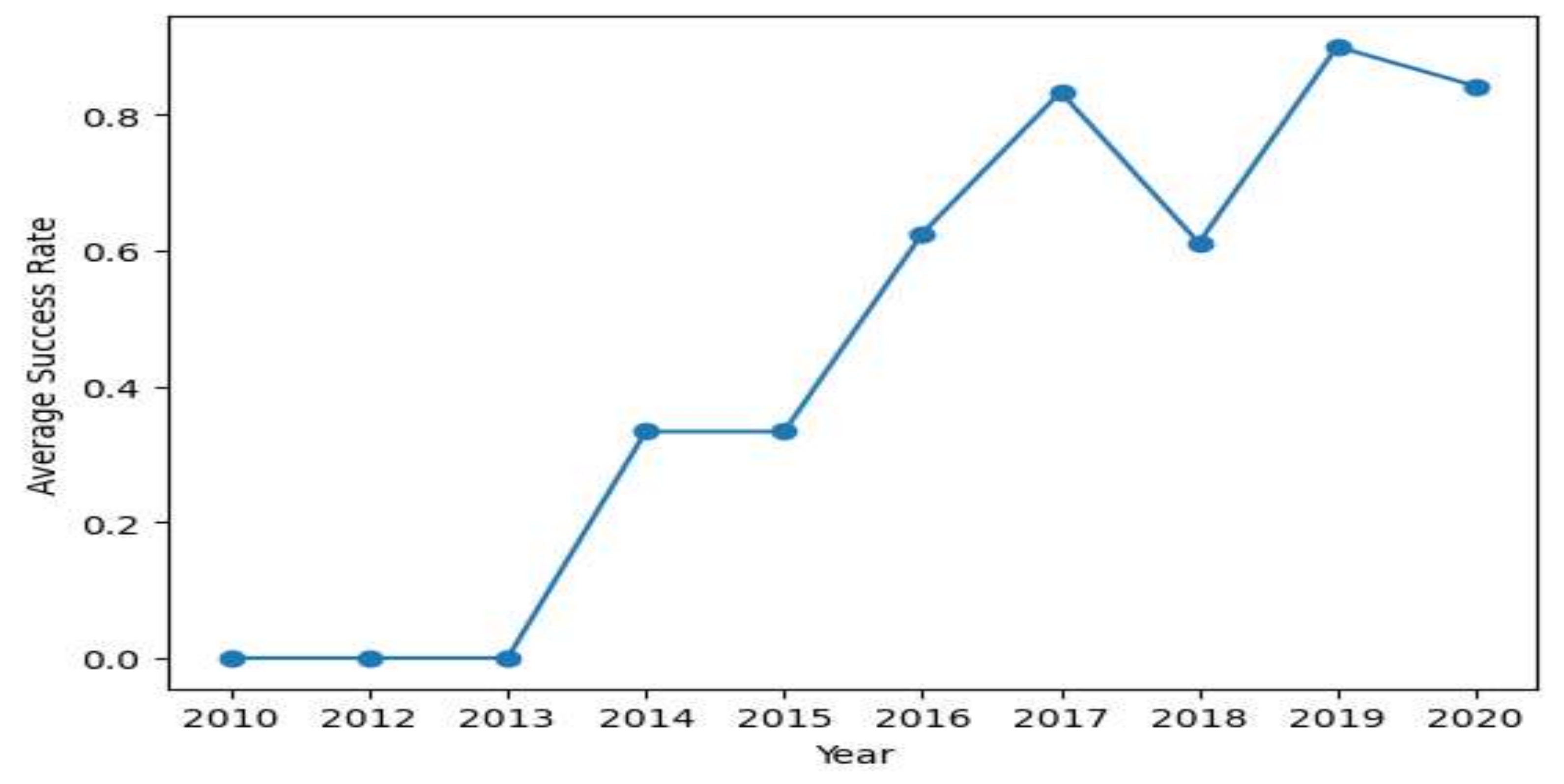
- We can observe that for some orbits like VLEO, ISS and LEO the more the flight number increases the better the odds. Still, we cannot conclude that for all the orbits.

Payload vs. Orbit Type



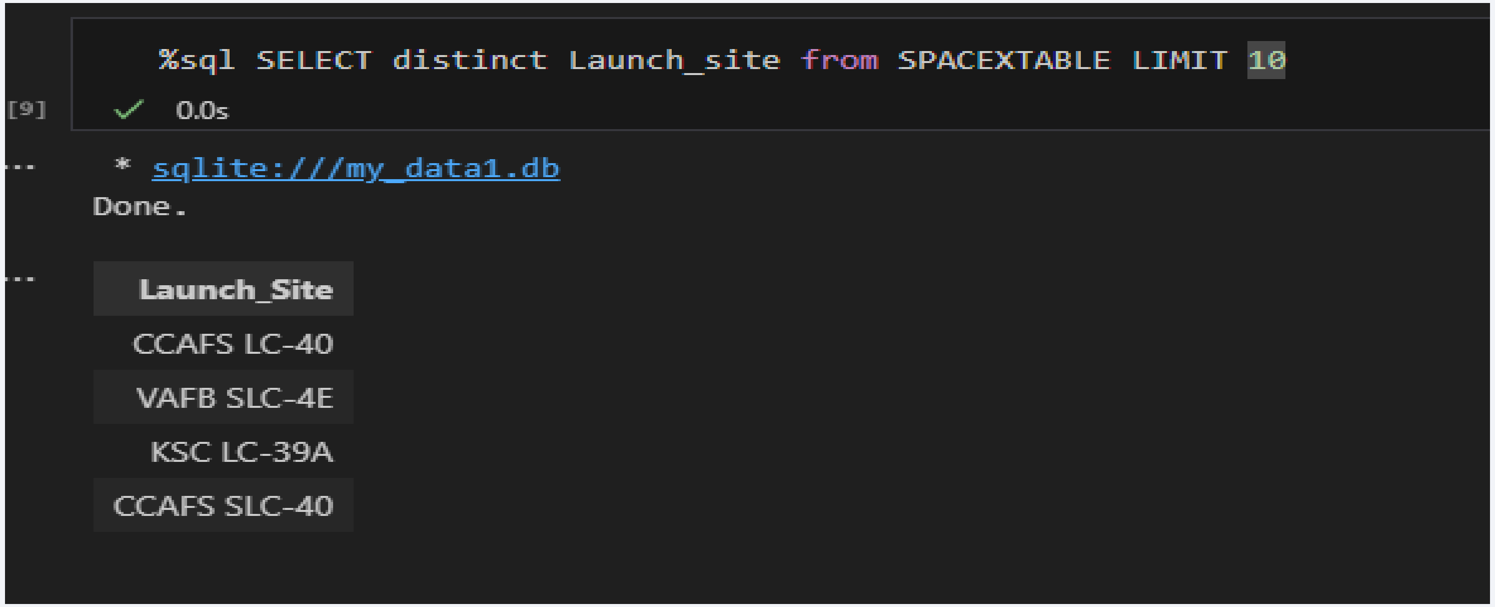
- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



- As beforementioned we can see a clear uptrend starting from 2013. This probably is due to the advancement of technology.

All Launch Site Names

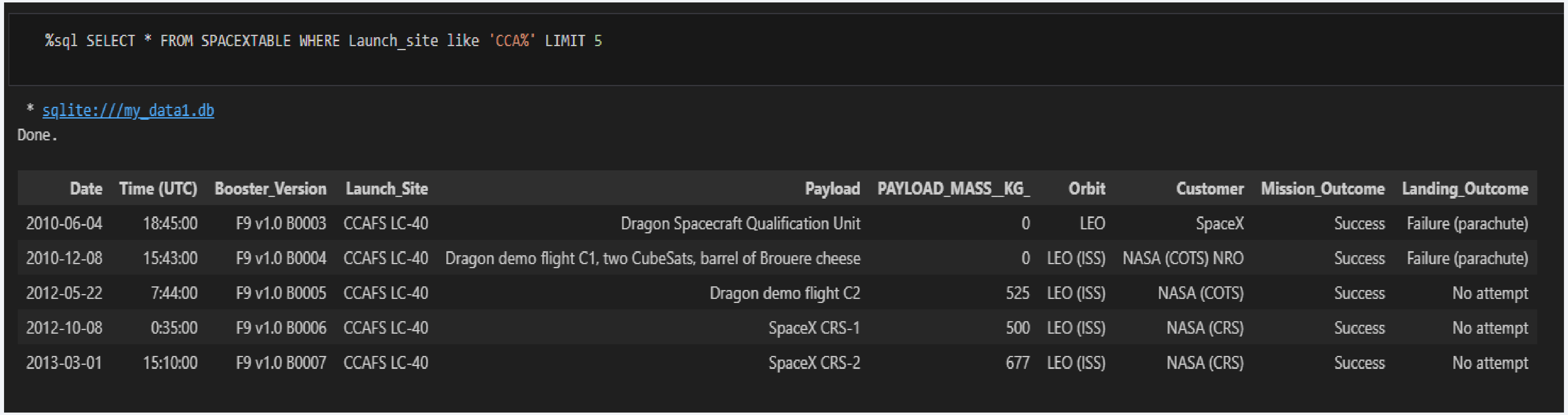


```
   %sql SELECT distinct Launch_site from SPACEXTABLE LIMIT 10
[9] ✓ 0.0s
…  * sqlite:///my_data1.db
   Done.
```

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

- Here we can see all launch sites.

Launch Site Names Begin with 'CCA'

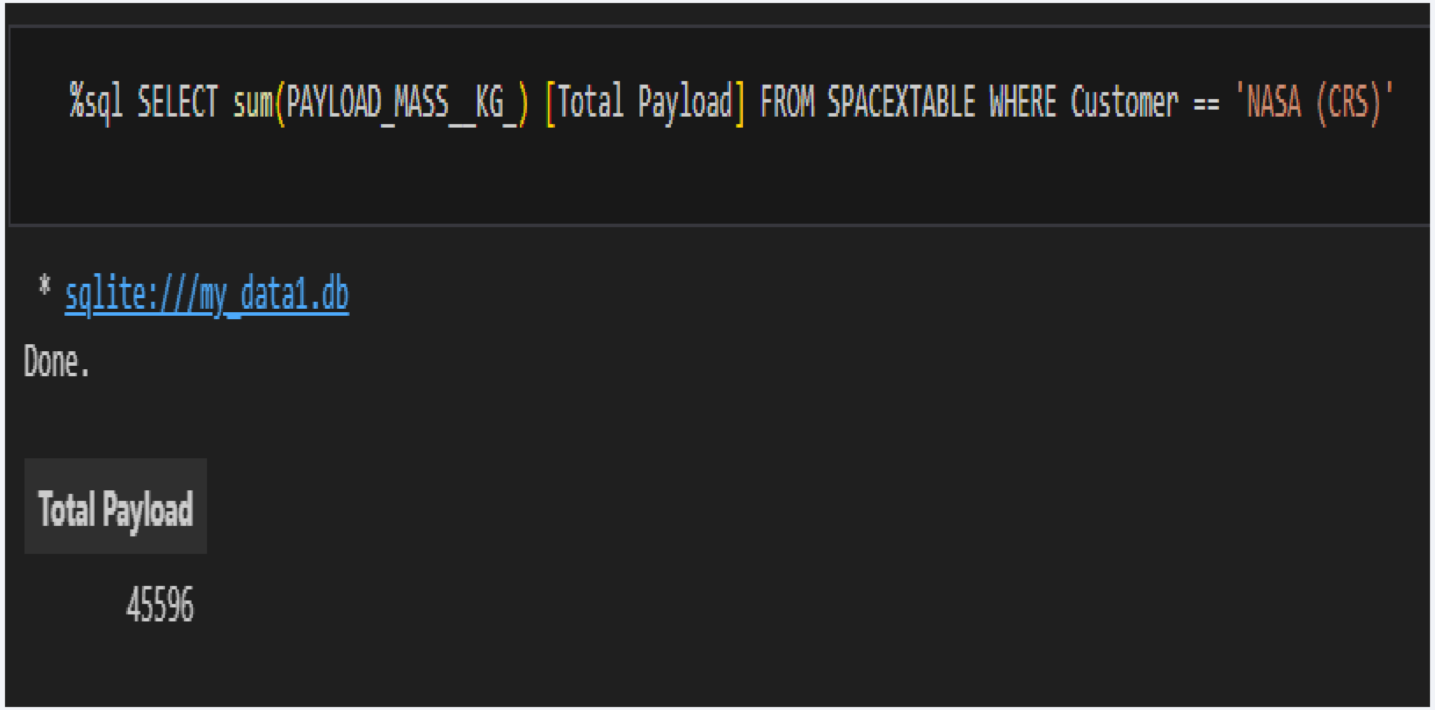


```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_site like 'CCA%' LIMIT 5
```

```
 * 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

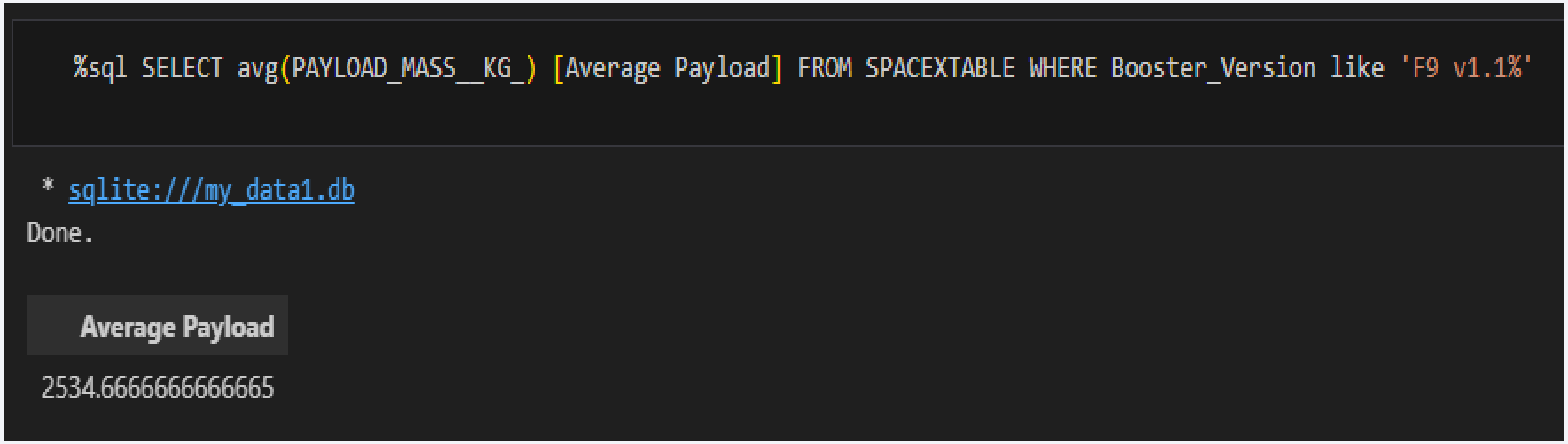


```
%sql SELECT sum(PAYLOAD_MASS__KG_) [Total Payload] FROM SPACEXTABLE WHERE Customer == 'NASA (CRS)'
```

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

Total Payload
        45596
```

Average Payload Mass by F9 v1.1

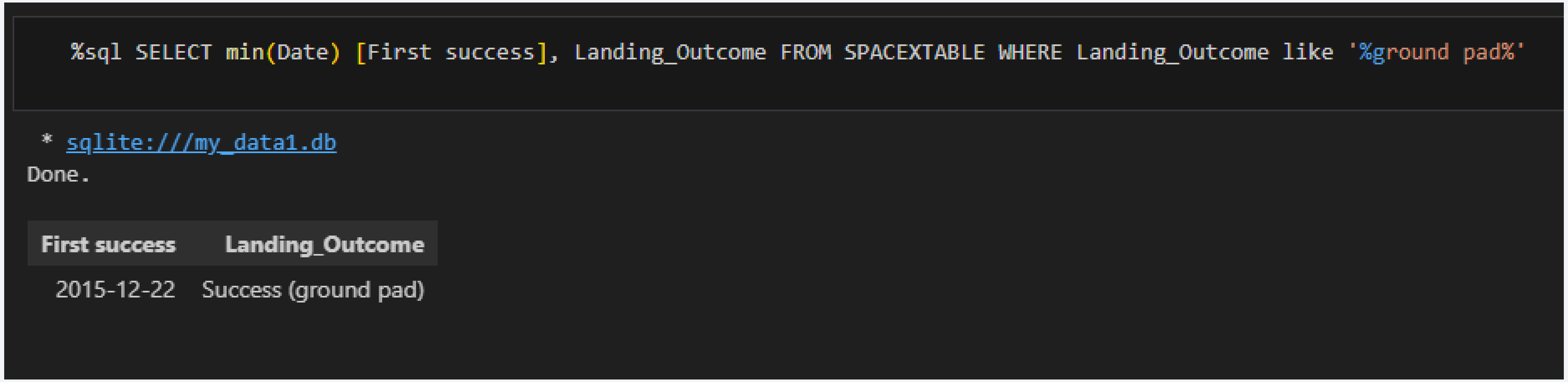


```
%sql SELECT avg(PAYLOAD_MASS__KG_) [Average Payload] FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'
```

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

Average Payload
2534.6666666666665

First Successful Ground Landing Date

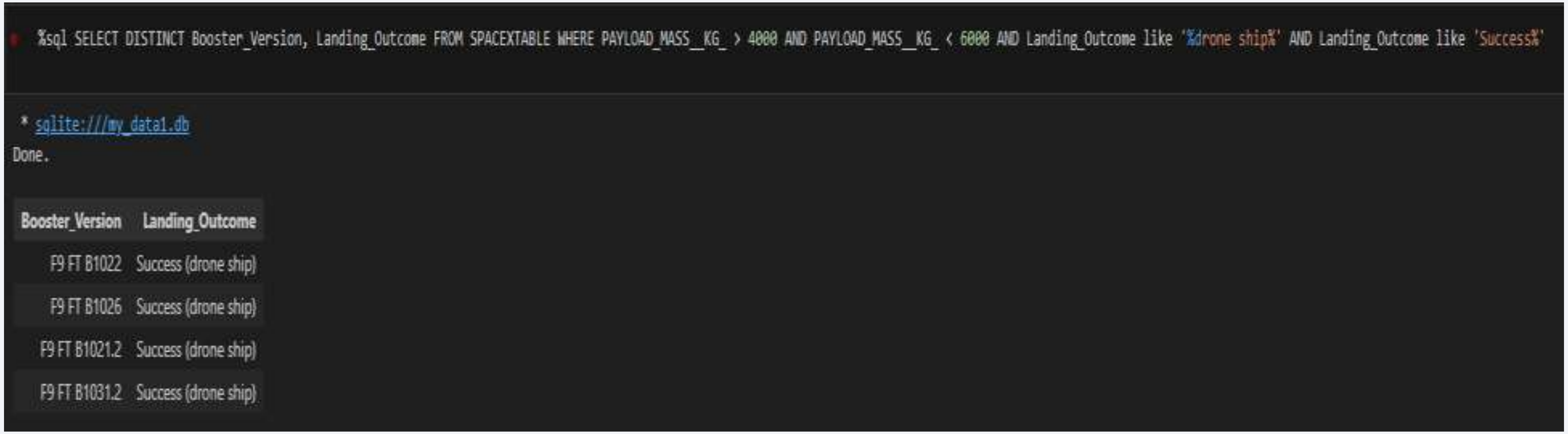


```
%sql SELECT min(Date) [First success], Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome like '%ground pad%'
```

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

First success	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

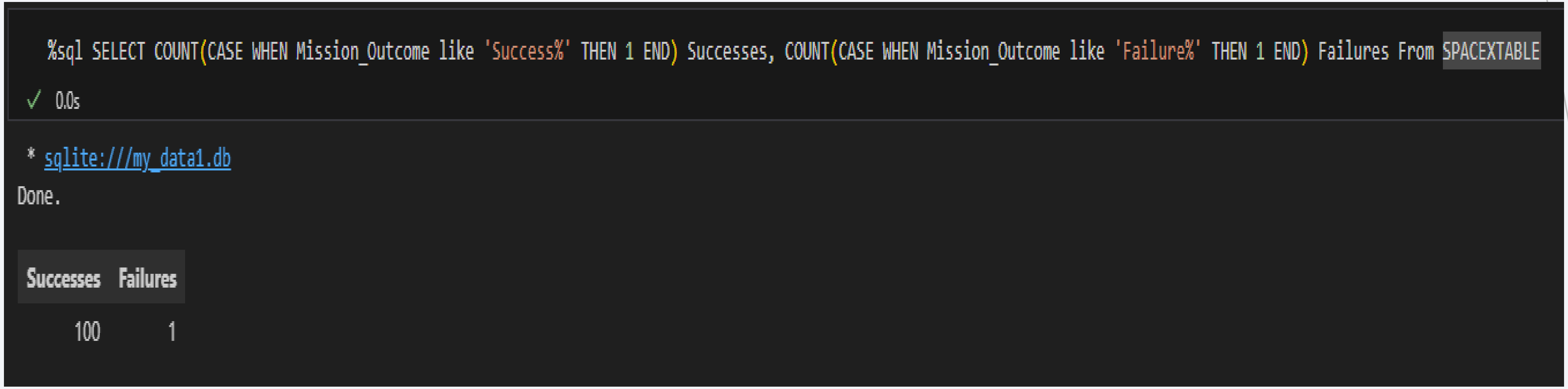


```
%sql SELECT DISTINCT Booster_Version, Landing_Outcome FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000 AND Landing_Outcome like '%drone ship%' AND Landing_Outcome like 'Success%'
```

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

Booster_Version	Landing_Outcome
F9 FT B1022	Success (drone ship)
F9 FT B1026	Success (drone ship)
F9 FT B1021.2	Success (drone ship)
F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes



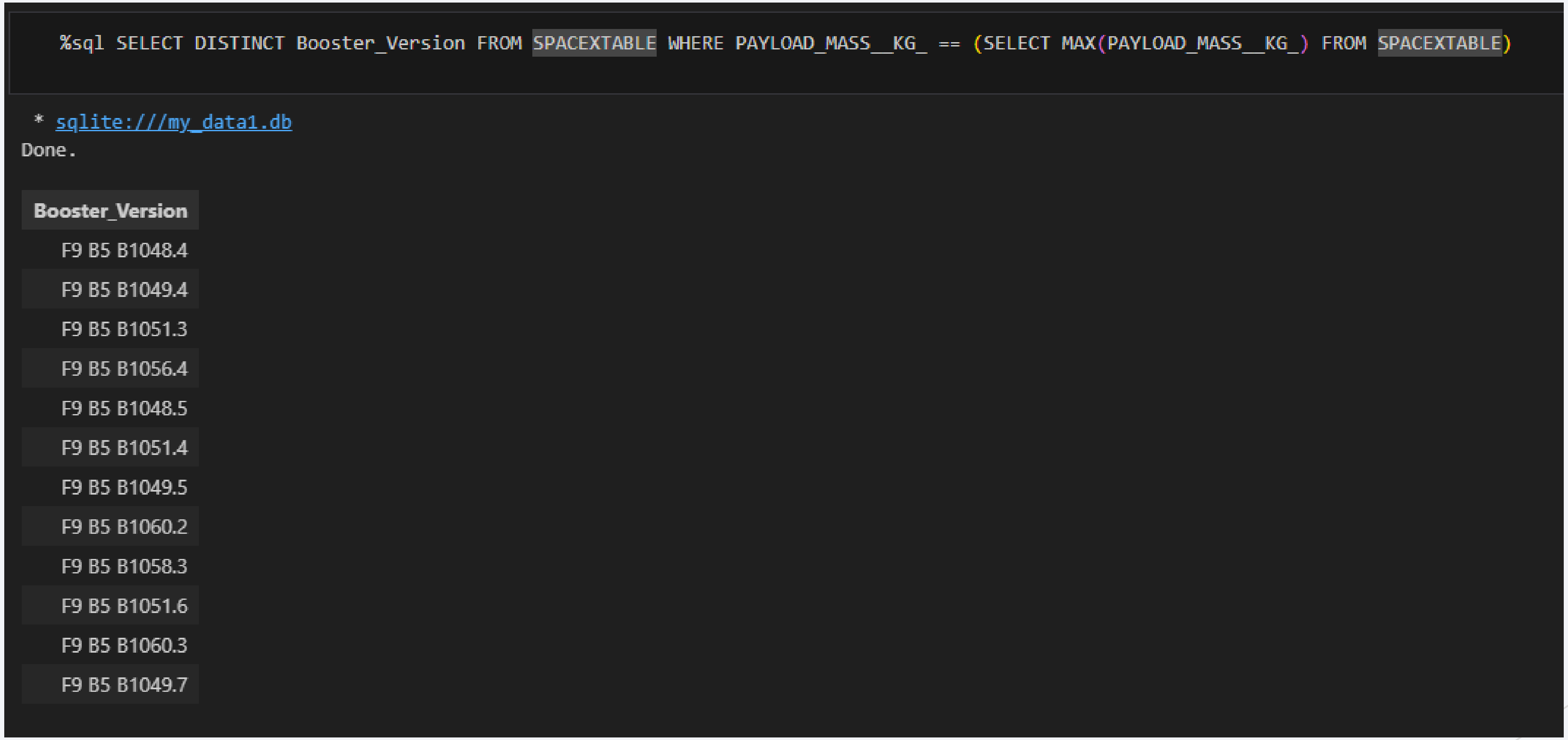
```
%sql SELECT COUNT(CASE WHEN Mission_Outcome like 'Success%' THEN 1 END) Successes, COUNT(CASE WHEN Mission_Outcome like 'Failure%' THEN 1 END) Failures From SPACEXTABLE
```

✓ 0.0s

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

Successes	Failures
100	1

Boosters Carried Maximum Payload

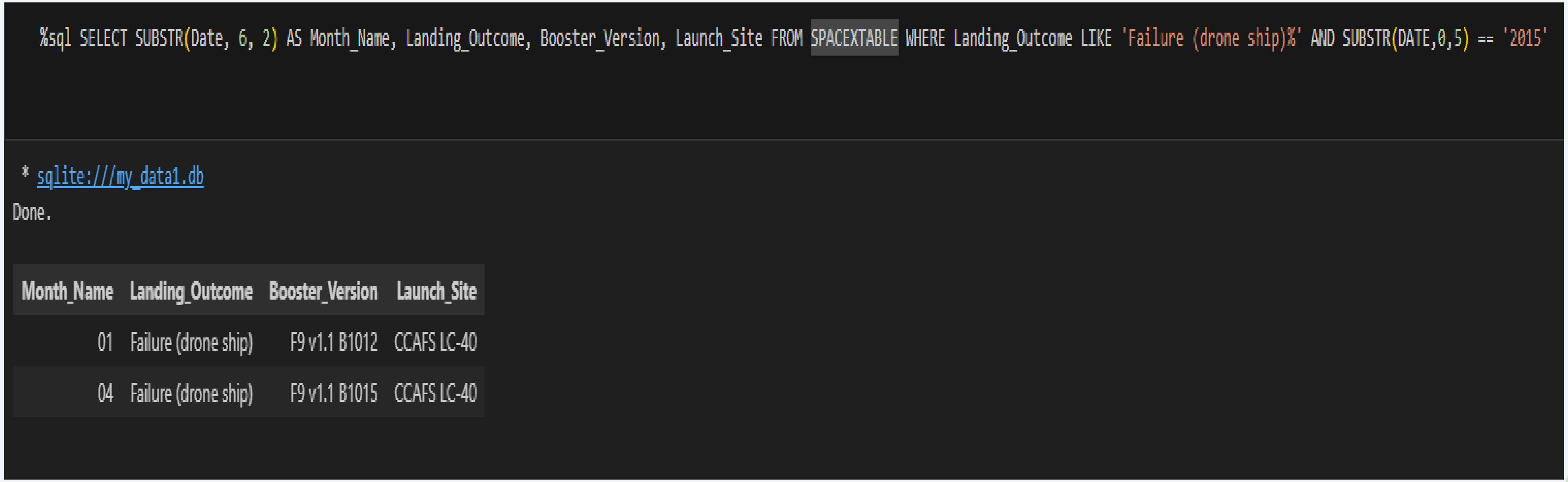


```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

```
 * 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

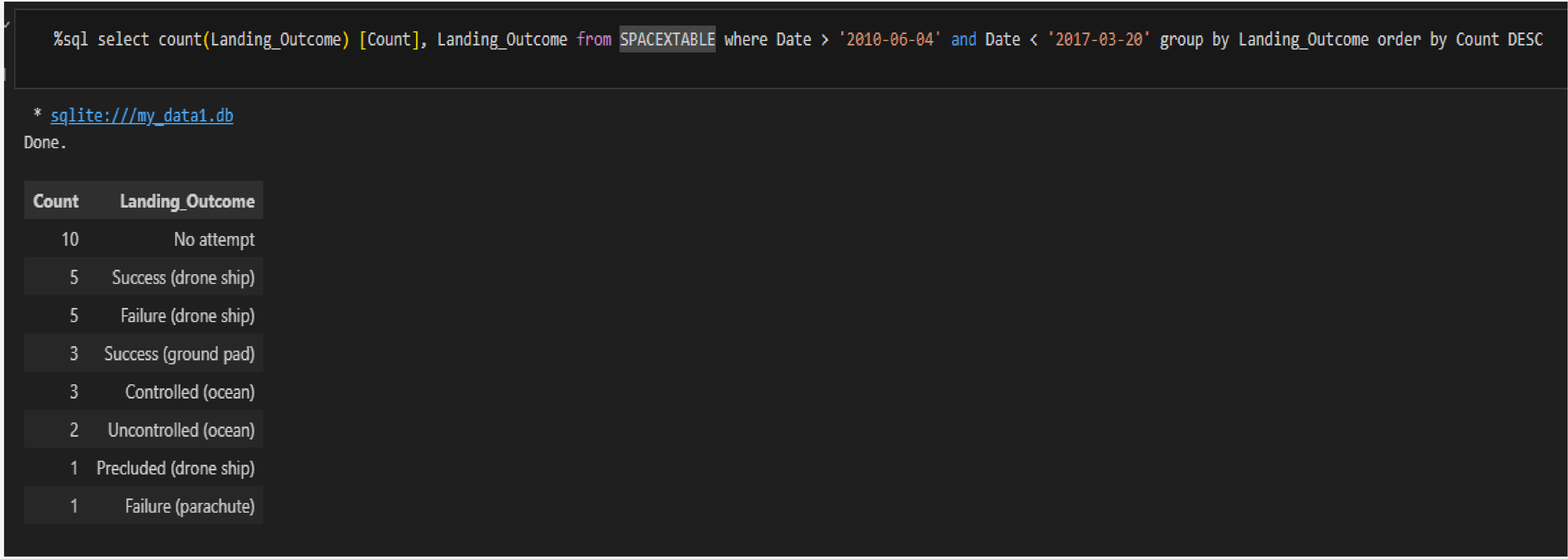


```
%sql SELECT SUBSTR(Date, 6, 2) AS Month_Name, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Failure (drone ship)%' AND SUBSTR(DATE,0,5) == '2015'
```

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

Month_Name	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



```
%sql select count(Landing_Outcome) [Count], Landing_Outcome from SPACEXTABLE where Date > '2010-06-04' and Date < '2017-03-20' group by Landing_Outcome order by Count DESC
```

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

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

Section 3

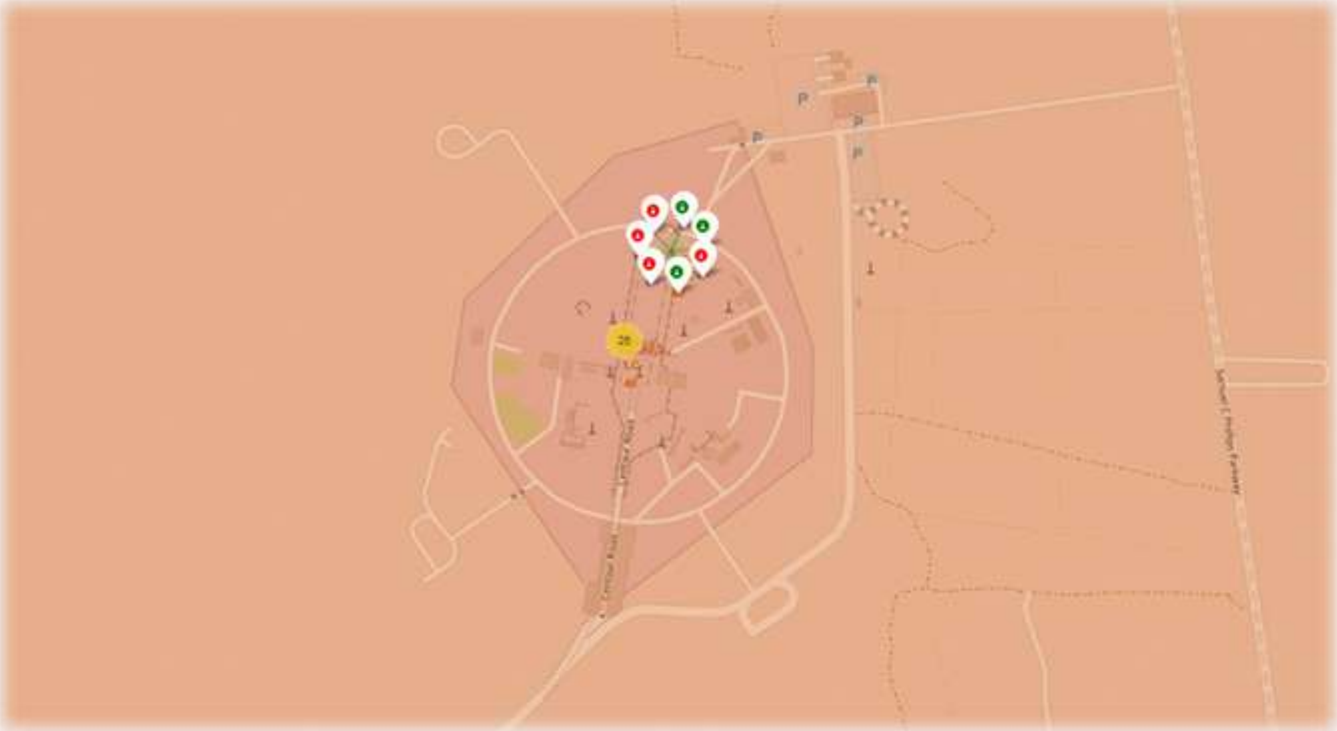
Launch Sites Proximities Analysis

Launch Sites’ Location

+
−
Kansas City
Las Vegas
Dallas
Nuevo Laredo
Baton Rouge
Durango
Saltillo
La Paz

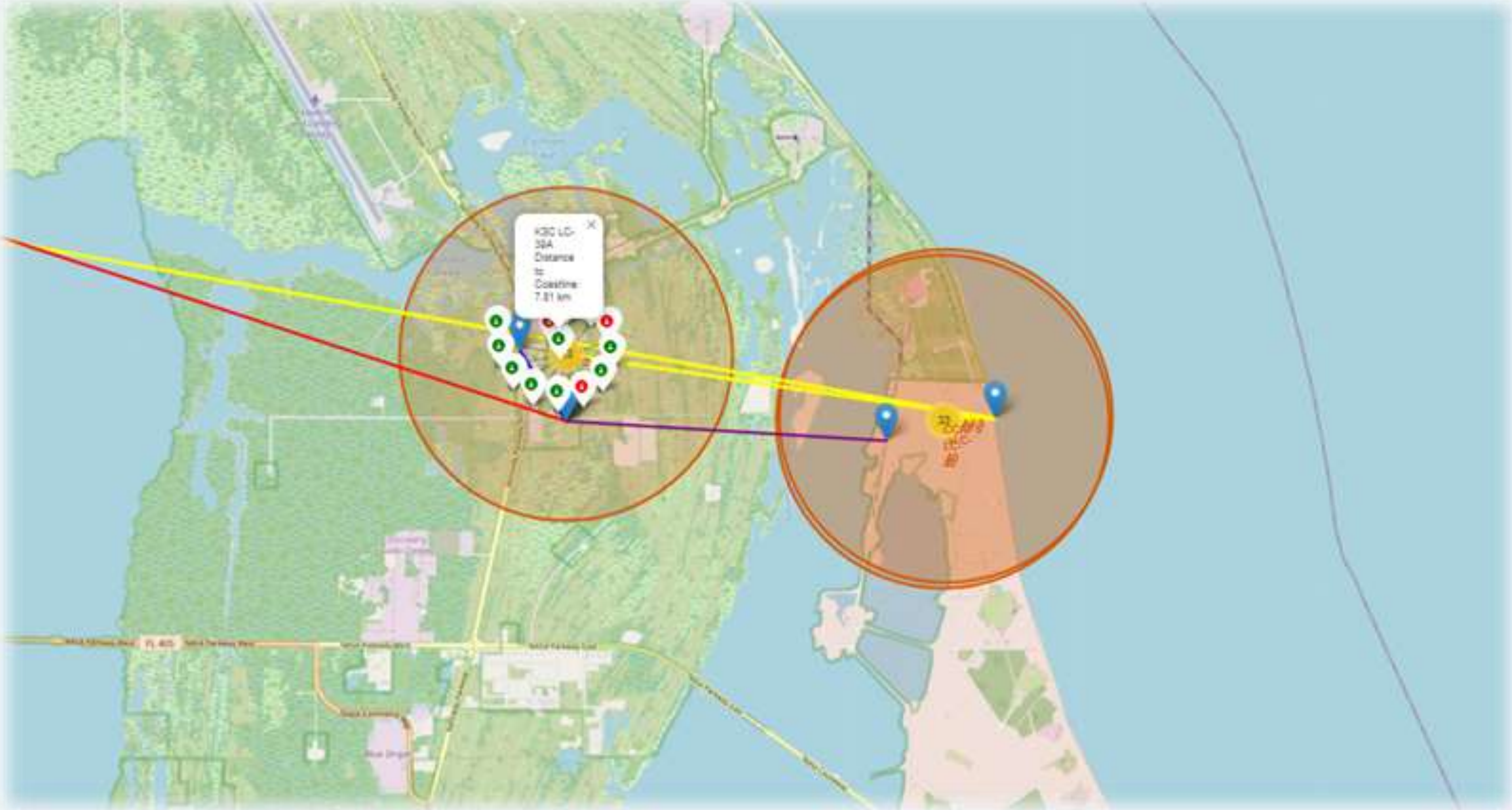
- From what we can barely see on the map 3 of the 4 sites are situated on the east coast and one on the west coast. Also all of the 4 are in a very close proximity to the ocean. The reason for that being is safety.

Launch Outcome by Site

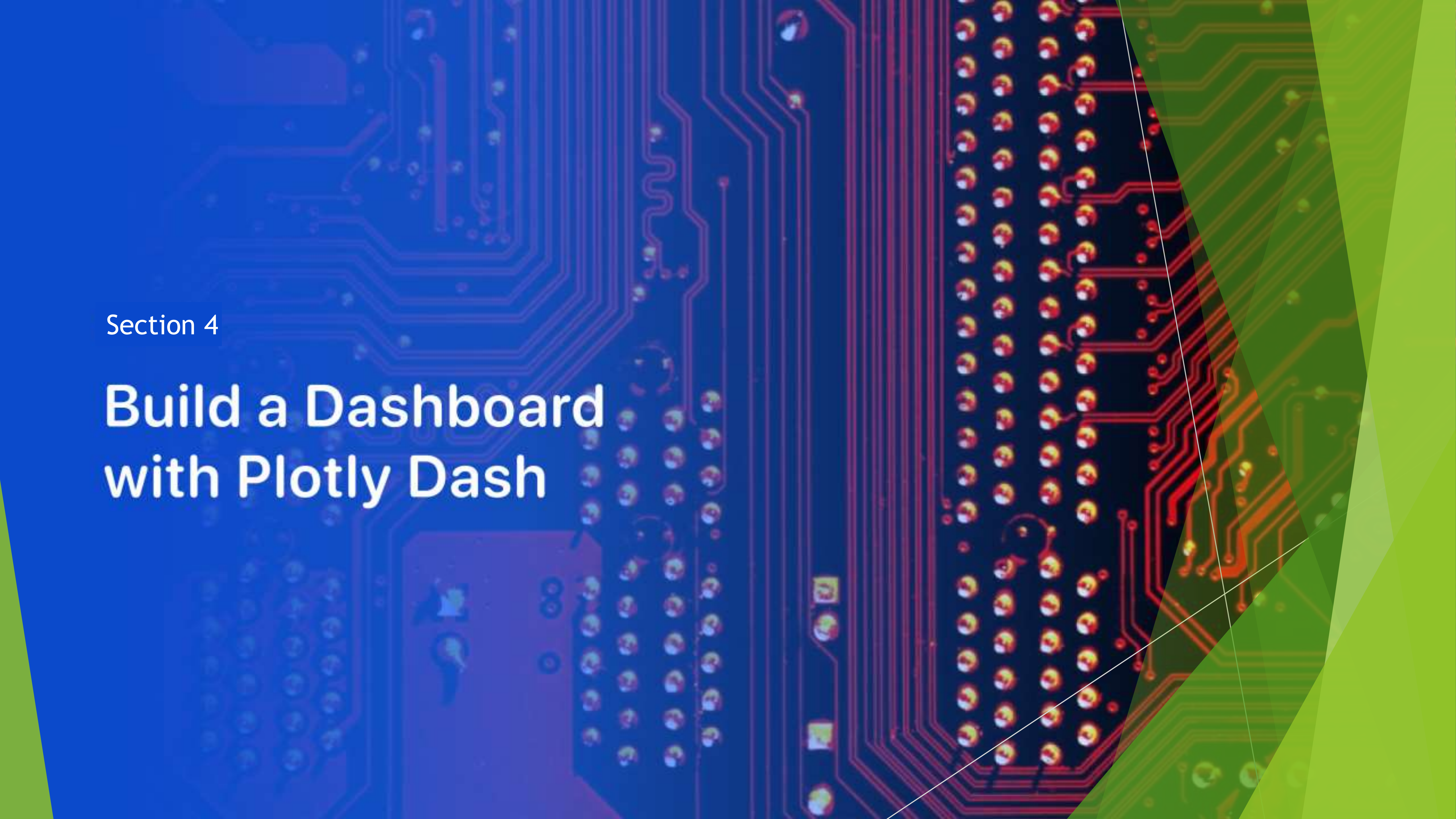


- Here we can see clustered all the data we have on each site. What is important and useful is that we can see how many trials are on each site and by clicking on the number we can see how many of them are successful and how many have failed.

Proximity Map



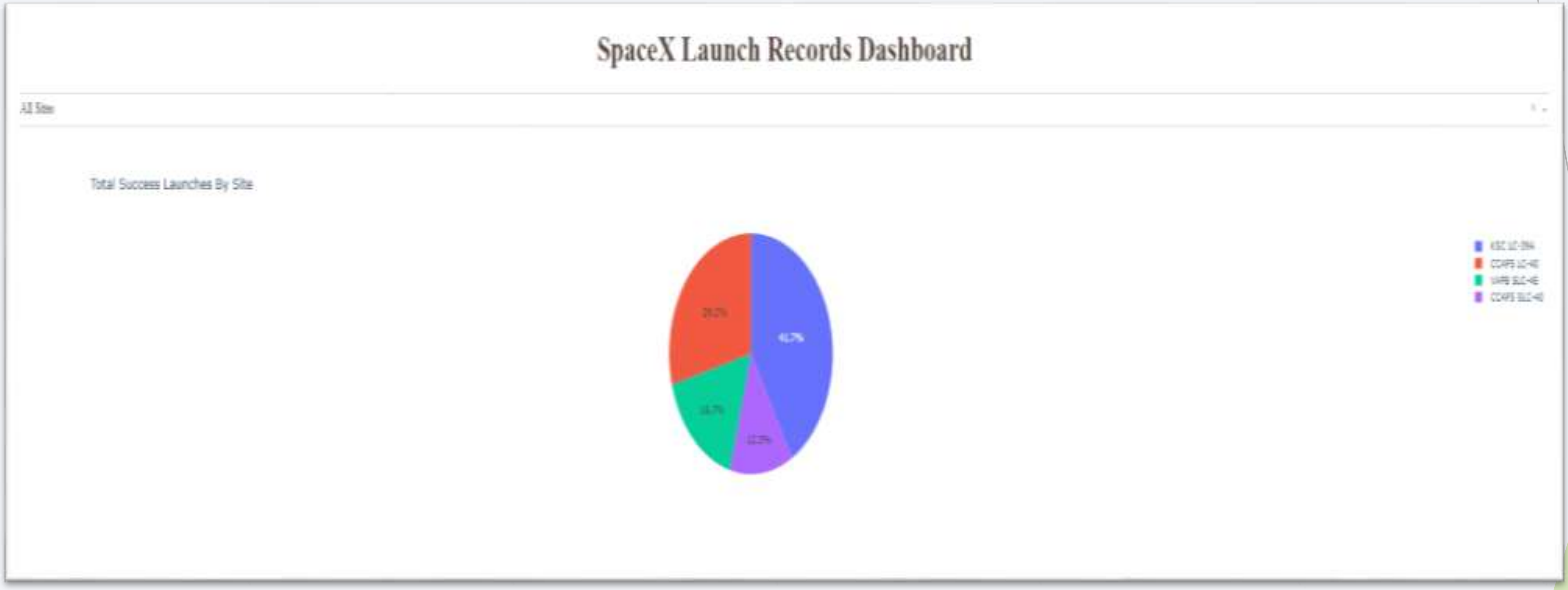
- From this map we can see the close proximity of the sites to things like cities, highways, railroads and coastlines. All of them are in close proximity to the coast and also to a railroad and highway. They are relatively distant from cities. All seems reasonable as for the safety measures beforementioned.



Section 4

Build a Dashboard with Plotly Dash

Total Success Launches By Site



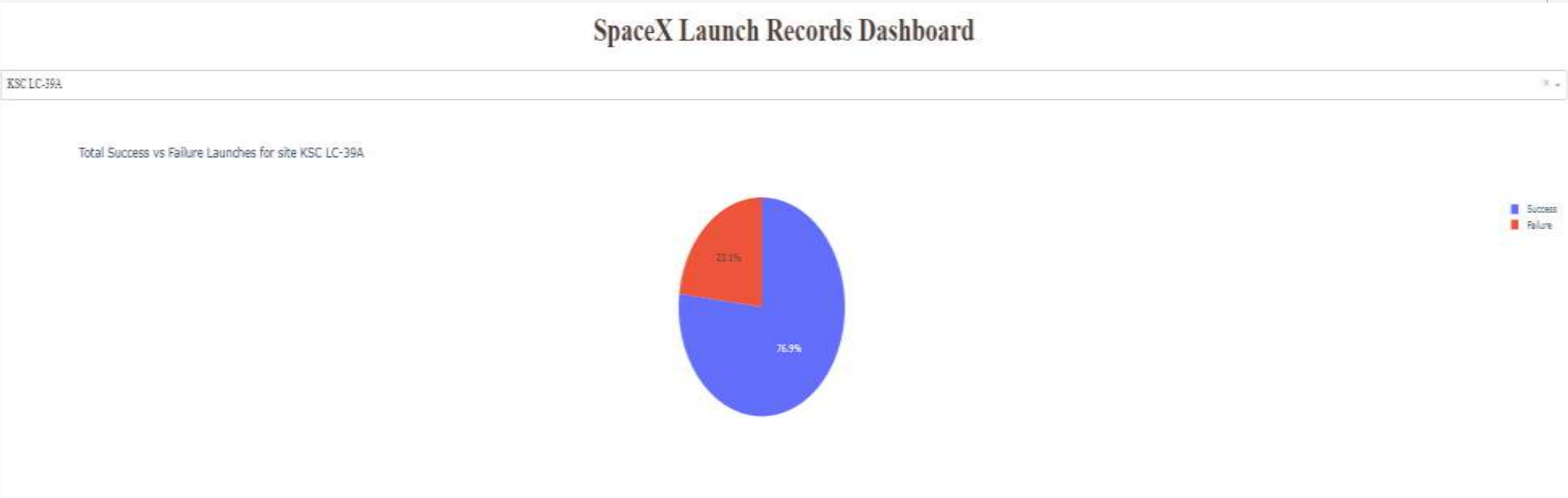
SpaceX Launch Records Dashboard

All Sites

Total Success Launches By Site

- We can see from this screenshot that KSC LC-39A is the launch site with the most successes.

Best Success Rate Site

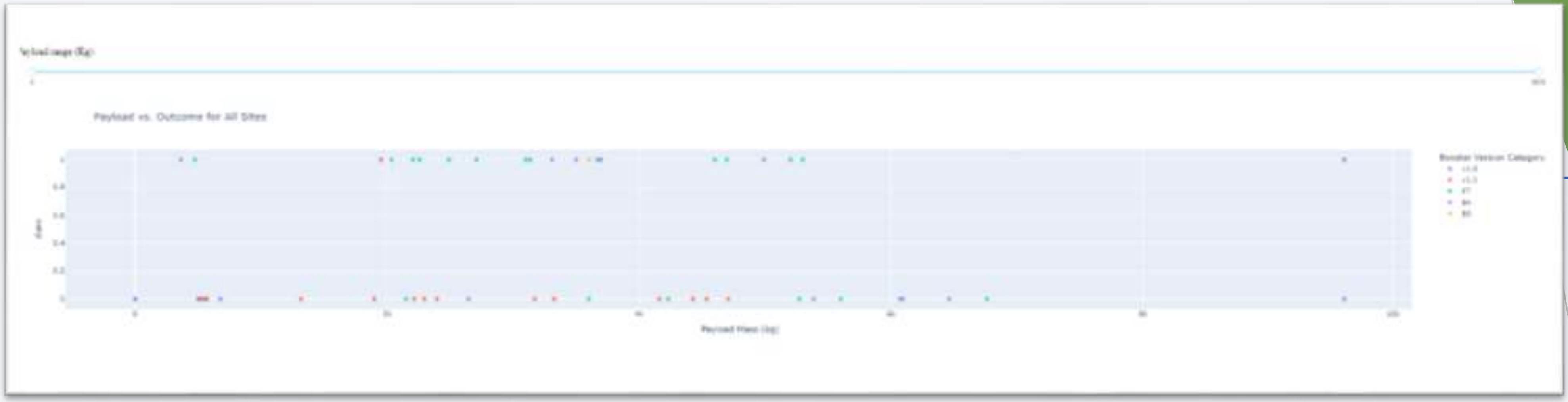


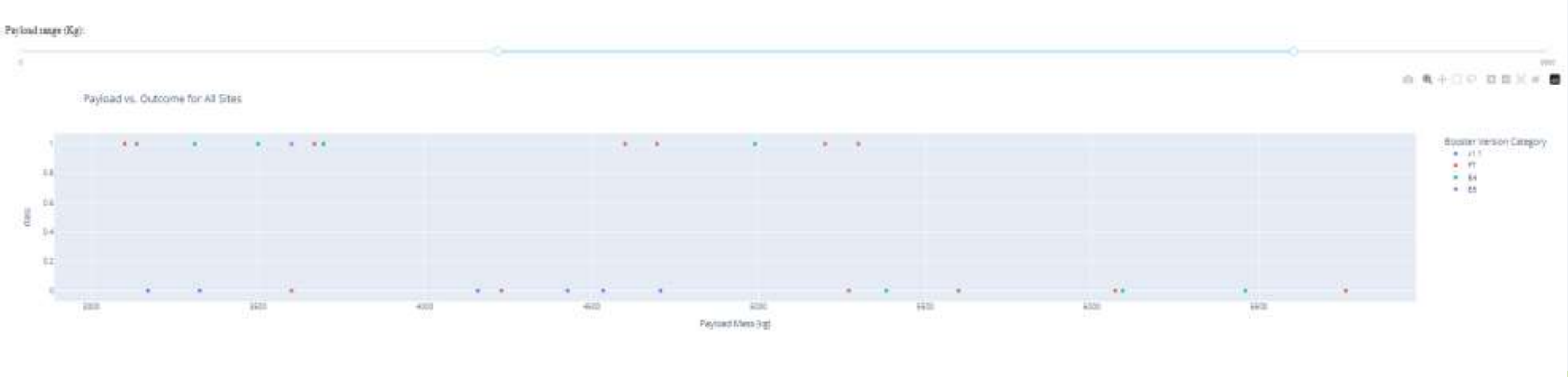
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success vs Failure Launches for site KSC LC-39A

- Here we can see the chart for the site with best success rate. KSC LC-39A is the most reliable site by much.



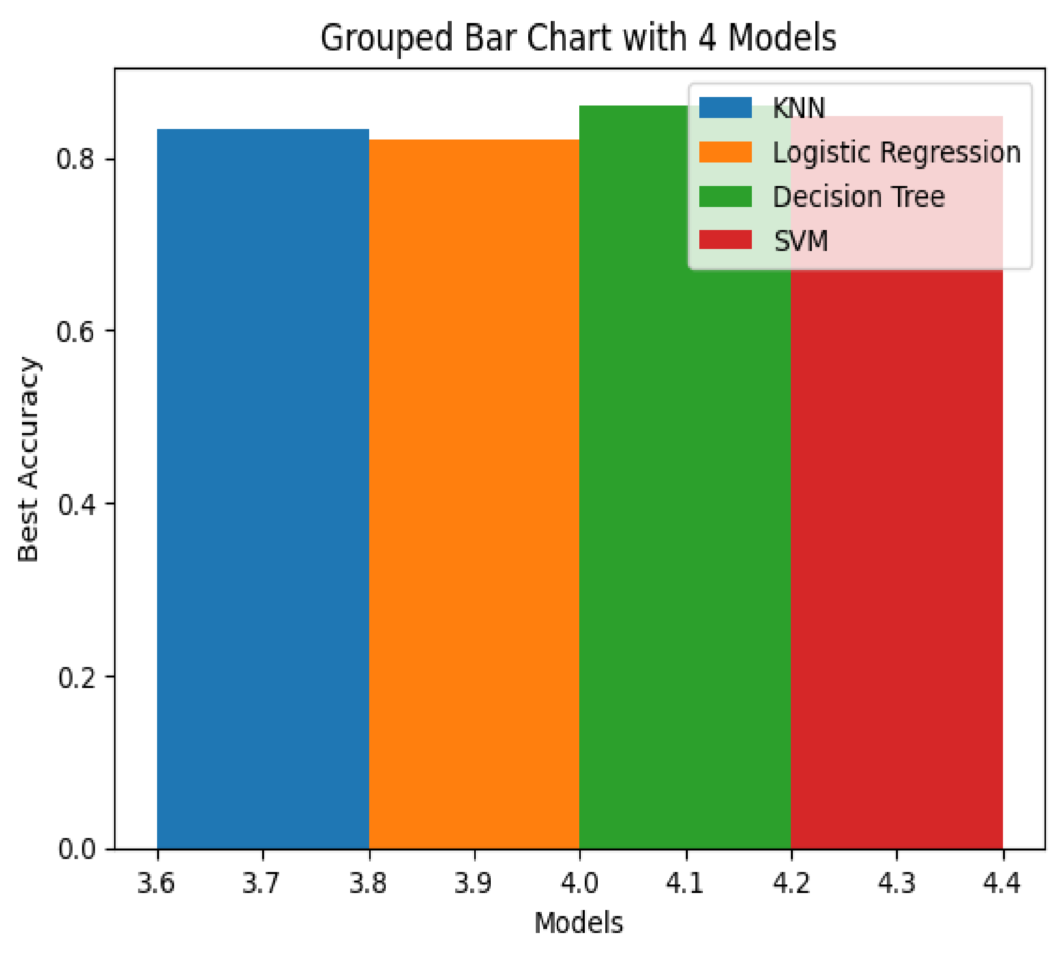


- The things we can see from these two charts of Payload vs. Outcome is that the booster version FT has most of the successful landings. Also we can see that most of the trials are withing the 2k to 4k payload mass and there is no clear correlation between payload mass and success.

Section 5

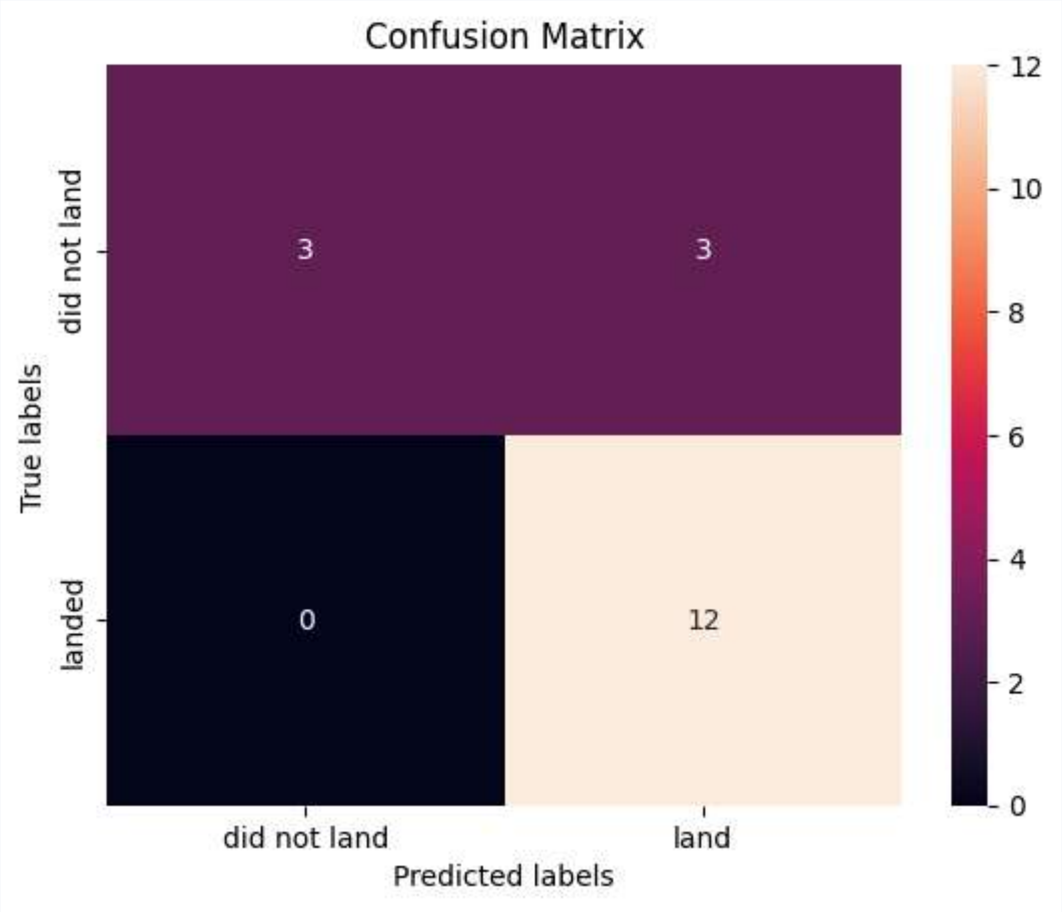
Predictive Analysis (Classification)

Classification Accuracy



- Although all the 4 models have yielded the same confusion matrix with the same accuracy on the testing data, decision trees has yielded above 90% accuracy as a best score.

Confusion Matrix



- This is the confusion matrix for the decision tree.
- True Postive - 12 (True label is landed, Predicted label is also landed)
- False Postive - 3 (True label is not landed, Predicted label is landed)
- This model predicts landed when it shouldn’t. Incorrectly predicting positive results might be bad in the future as incorrectly predicting negative results might at least

Conclusions

- There is a close relation between the flight numbers and success. Also in some orbits the Payload Mass might have a positive correlation with success as well.
- With the improvement of technology the success rate has risen drastically after the year 2013. It has reached a good rate of around 80-90%.
- KSC LC-39A is the most reliable site by much. It has an above 70% success rate. This might be due to the fact that the data we have on this site is after 2017 only which might affect the current results.
- The classification models that I have used yield similar confusion matrixes. The good thing is that they give a 83% accuracy score on the testing data.
- Picking the optimal orbit, payload mass and launch site, as well as the

Thank you!