IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Maksim Shek
13.02.2024

Outline

Executive Summary

TL;DR

The main goal of this project is to develop a predictive model to ascertain whether the Falcon 9's first stage will land successfully. Utilizing SpaceX's comprehensive launch data, we analyzed all Falcon-9 launches to date. This model addresses a critical question with significant implications for cost efficiency and sustainability in space travel: “Will it land?” By predicting the outcome of the first stage landings, our model aims to enhance the feasibility of reusable rockets, marking a pivotal step towards reducing launch costs and fostering sustainable space exploration practices.

Introduction

Space travel is no longer a form of fantasy, it has almost become a reality! There is no doubt that our future is tightly bound with space, and the paradigm has shifted from "Is it really possible?" to "How can we build a thriving business around it?". The biggest problem for commercial space travel, and the reason it is not yet a widespread reality, is the fact that current rockets are single-use, which makes this industry unaffordable for most people and hence not profitable. One solution is to reuse the first stage, but while some rockets land seamlessly, others give us more of a pyrotechnical show. Can we address this issue?

To answer this question, we will examine data from a cutting-edge company that has been leading the space industry for the last few years – SpaceX. They are definitely learning from their mistakes, and so are we. After inspecting the data, we have identified the most promising concepts and ideas to build our business around.

Section 1

Methodology

Methodology

Executive Summary

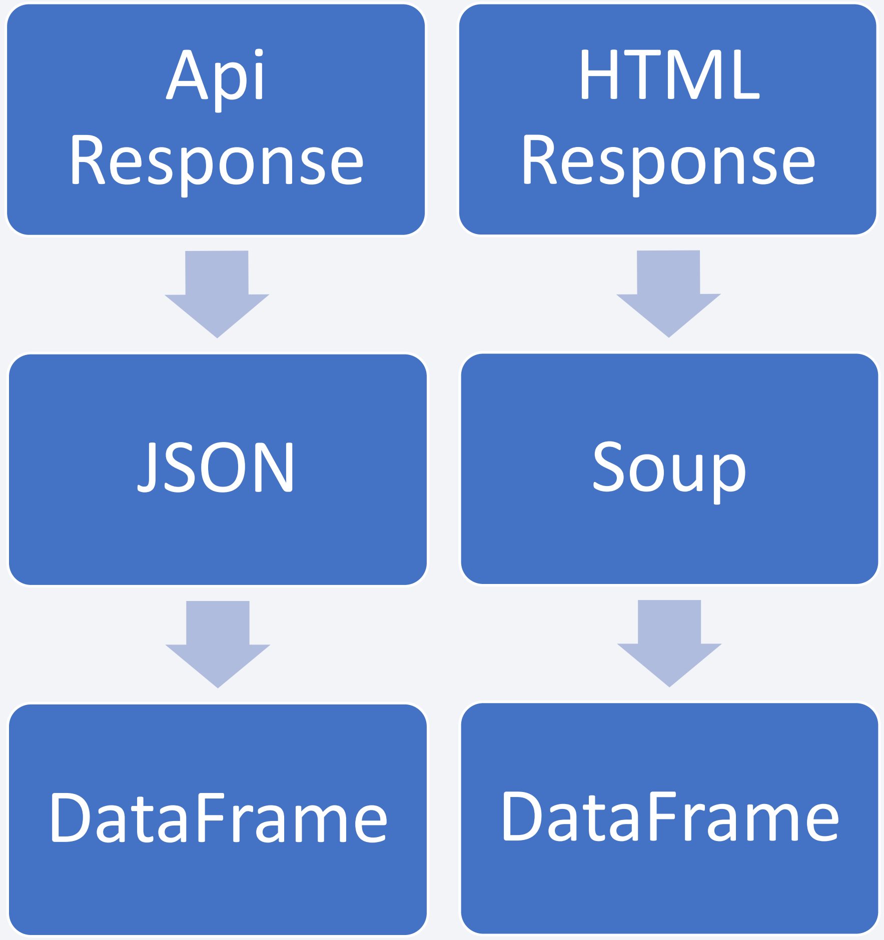
- Data collection methodology:
    - Rocket Launch Data from SpaceX API
    - Web Scraping Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling
    - Characteristic extraction
    - Data Filtering
    - Dealing with missing values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Create and train several models
    - Fine tune them
    - Evaluate and choose best one

Data Collection

For this analysis, two primary data sources were utilized: the SpaceX API and the Wikipedia page detailing Falcon Heavy launches.

Api is convienient and offers data nicely wrapped in json. So is already usable when received, we can simply serialize it and make dataframe out of it using just 2 rows of code.

Web Scraping Wikipedia: While the API offered structured data, additional information on Falcon Heavy launches was sourced through web scraping Wikipedia. Using Beautiful Soup, we parsed the HTML content to extract relevant tables, specifically focusing on launch details not covered by the API. The process involved identifying the correct table tags, extracting table rows, and then applying data transformations to align with the structure of our existing dataset.

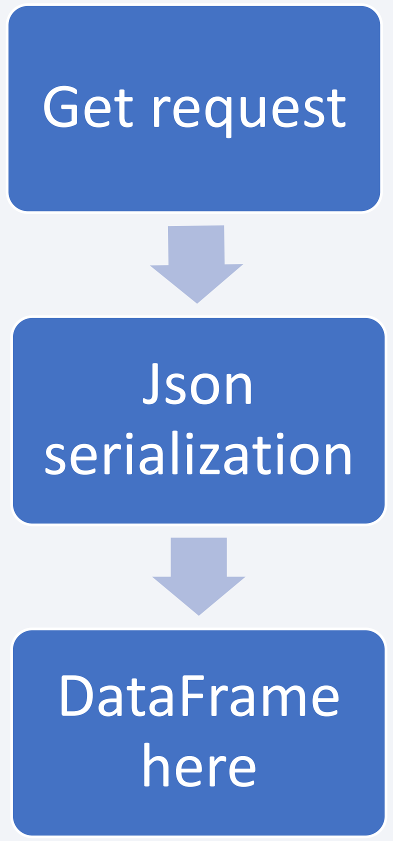


Data Collection – SpaceX API

To get data from SpaceX API we follow four simple steps

1. Use the Get method from the requests library which allows us to “query” data from the server
2. The Chosen endpoint contains historical data about SpaceX Rockets. Docs are here. We can select what data to receive, but we want to gather all, so we are not going to specify anything
3. Most Api communicates via JSON or XML. Luckily, here we have JSON. So we can use JSON method to convert it to the dictionary
4. After this, we can serialize a list of dicts to DataFrame using pd.json_serialize method, Simple as that

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/jupyter-labs-spacex-data-collection-api.ipynb

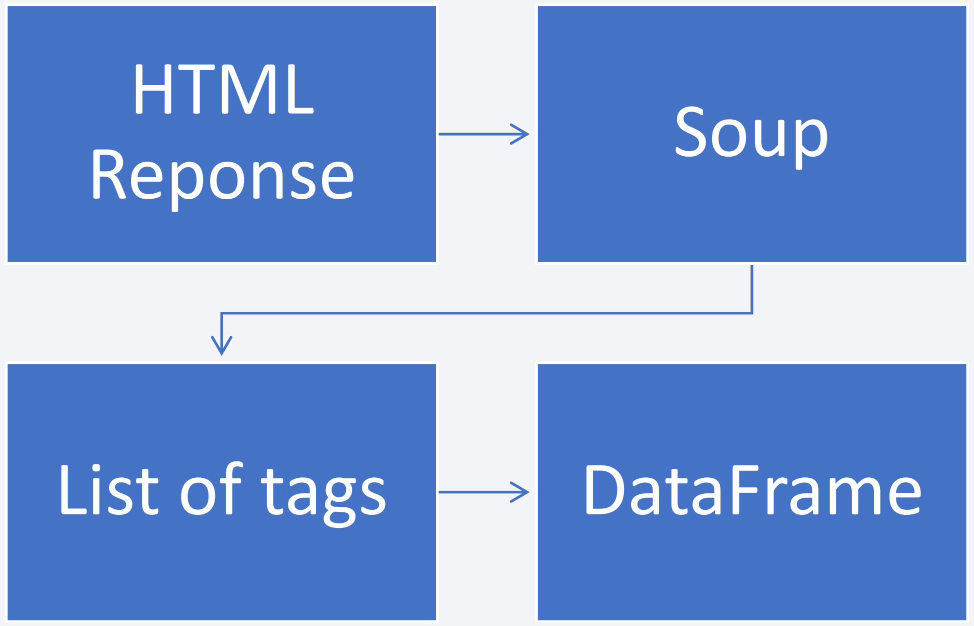


Data Collection - Scraping

Web Scraping is sometimes a hard and meticulous process. Luckily for us, we need only scrape tables, so

1. “Get” data from wiki page
2. Create tag soup using BeautifulSoup library
3. Find all tables by tag
4. Gather headers from <th> tags
5. Gather records row by row <tr> tags
6. Convert to Dataframe

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/jupyter-labs-webscraping.ipynb



Data Wrangling

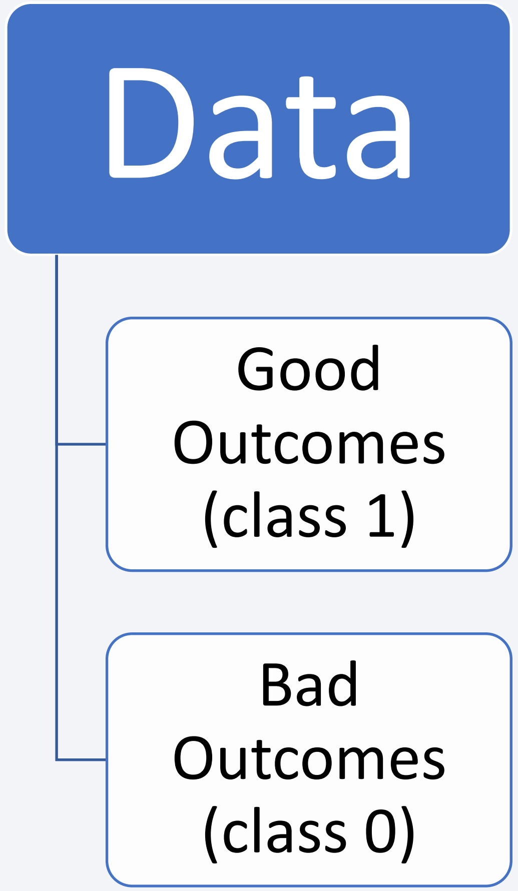
This step involves data exploration, data cleaning, and a bit of data transformation.

After investigation, we've found out that we have only missing values in the LandingPad feature, and it is all due to a lack of data. But this absence of data is actually meaningful. If it doesn't land, then the outcome is negative.

Since the final goal of our project is to find out if it lands, so we create a target class due to the provided data.

We have bad outcomes and good ones and label records, respectively.

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

Due to the fact that most of our features are categorical, we do not have much choice In terms of visuals. Catplots and barplots are the most appropriate plot type for this.

After careful data investigation, we found out that

- Two platforms are used for heavy launches, and one is not.
- The success rate was gradually increasing over time.
- Least successful rate were on GTO and SO orbit Launches
- Most successful is SSO orbit

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Ad-hoc analysis

- Total mass carried by NASA is 107010kg
- Average mass carried by Falcon 9 is 2535 kg
- First successful ground landing was performed in 2015
- In 2015 there were 2 failure by landing on drone ship

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

To see whole thing we need first to place markers on Launch Sites.

Other valuable information is proximity of sites including coastal line, cities, railways and highways.

So I add lines and little but of annotations to show distances for closest to Launch Site objects.

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Two primary graphs are presented: the first depicts the success ratio at a selected site compared or at all sites, and the second illustrates the relationship between success and payload mass.

These graphs are closely related and enable us to explore the connection between the mass of the payload and the success rate at a specific site. For instance, nearly all heavy launches from the first site achieved success, whereas only about half from the second site were successful (this is speculative).

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/spacex_dash_app.py

Predictive Analysis (Classification)

We employed a straightforward model along with grid search, achieving satisfactory results with an accuracy exceeding 80% on the test set. The tree model, however, delivered the least impressive performance among all our models, while other models demonstrated a comparable level of effectiveness.

There's considerable scope for further enhancement. One potential improvement involves adopting the normalized logloss metric, which could offer a deeper insight into the misclassification errors of our models.

Another area of improvement concerns the models themselves. For achieving a more precise and sophisticated model, and where representativeness is not a primary concern, we might opt for a gradient boosting method combined with random forest. My personal preference in this context is XGBoost.

Lastly, additional fine-tuning efforts could prove beneficial. For instance, utilizing Optuna for more extensive option exploration could be advantageous. Surprisingly, even KNN could yield unexpected benefits if we experiment with different distance metrics.

https://github.com/T1r3sh/IBM-assignments/blob/main/Capstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

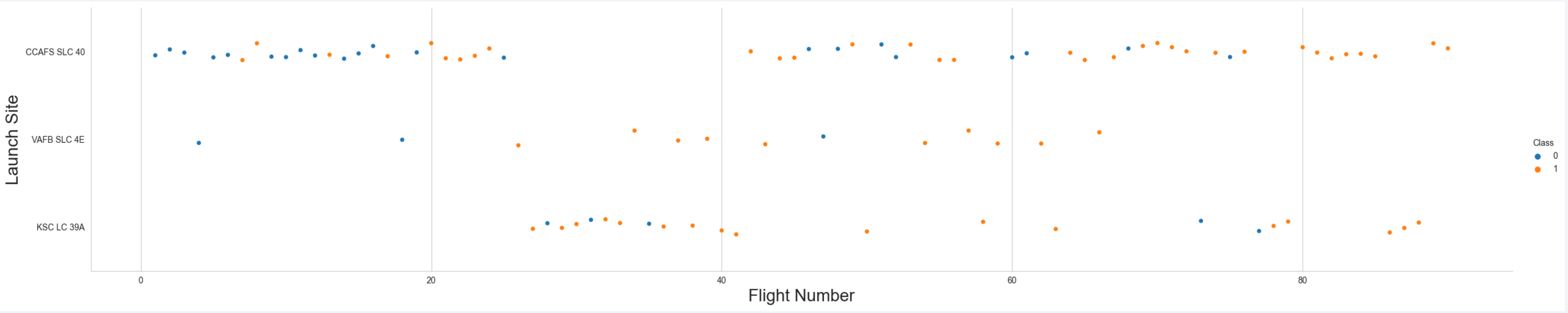
Results

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

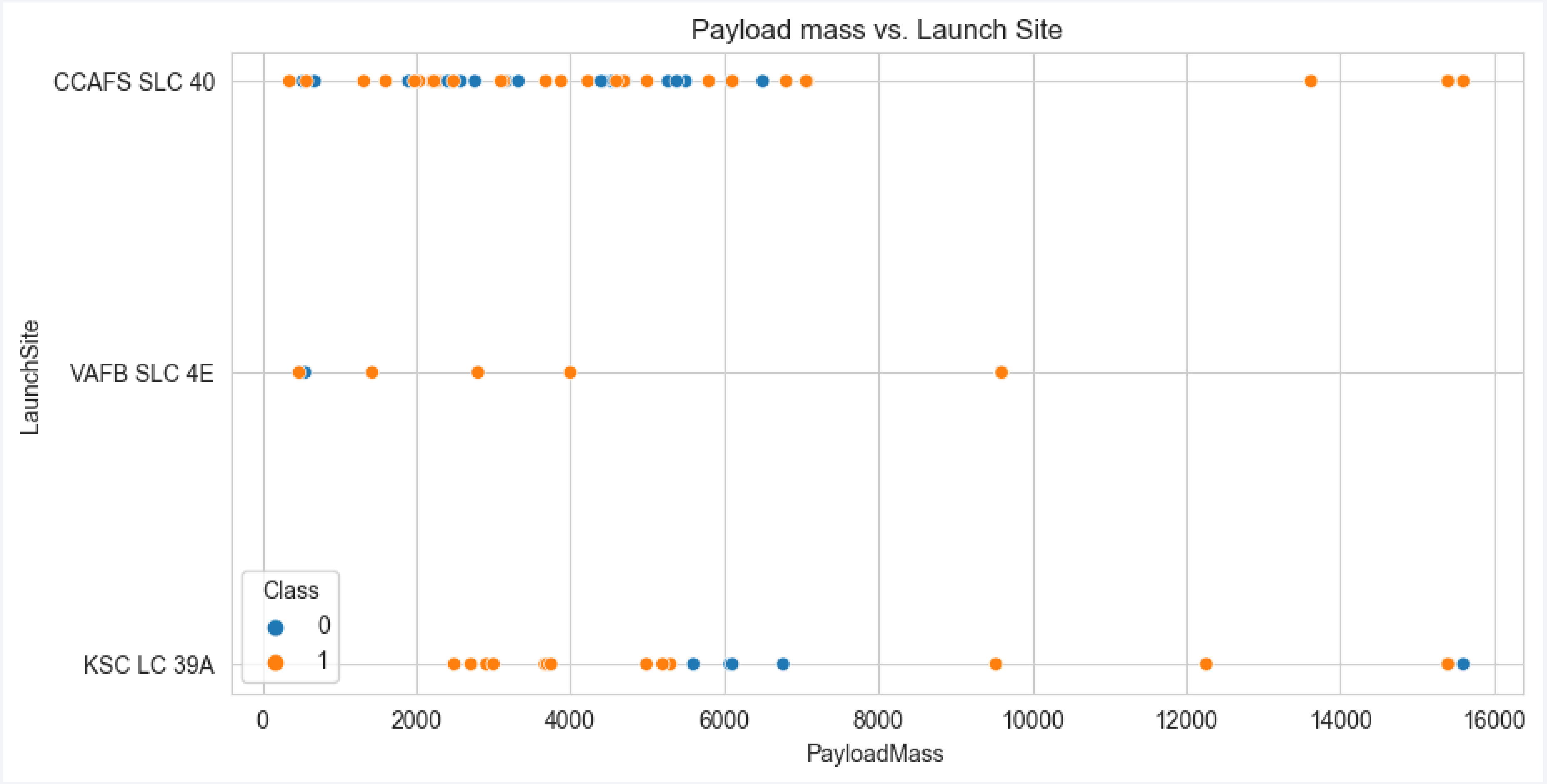
Section 2

Insights drawn from EDA

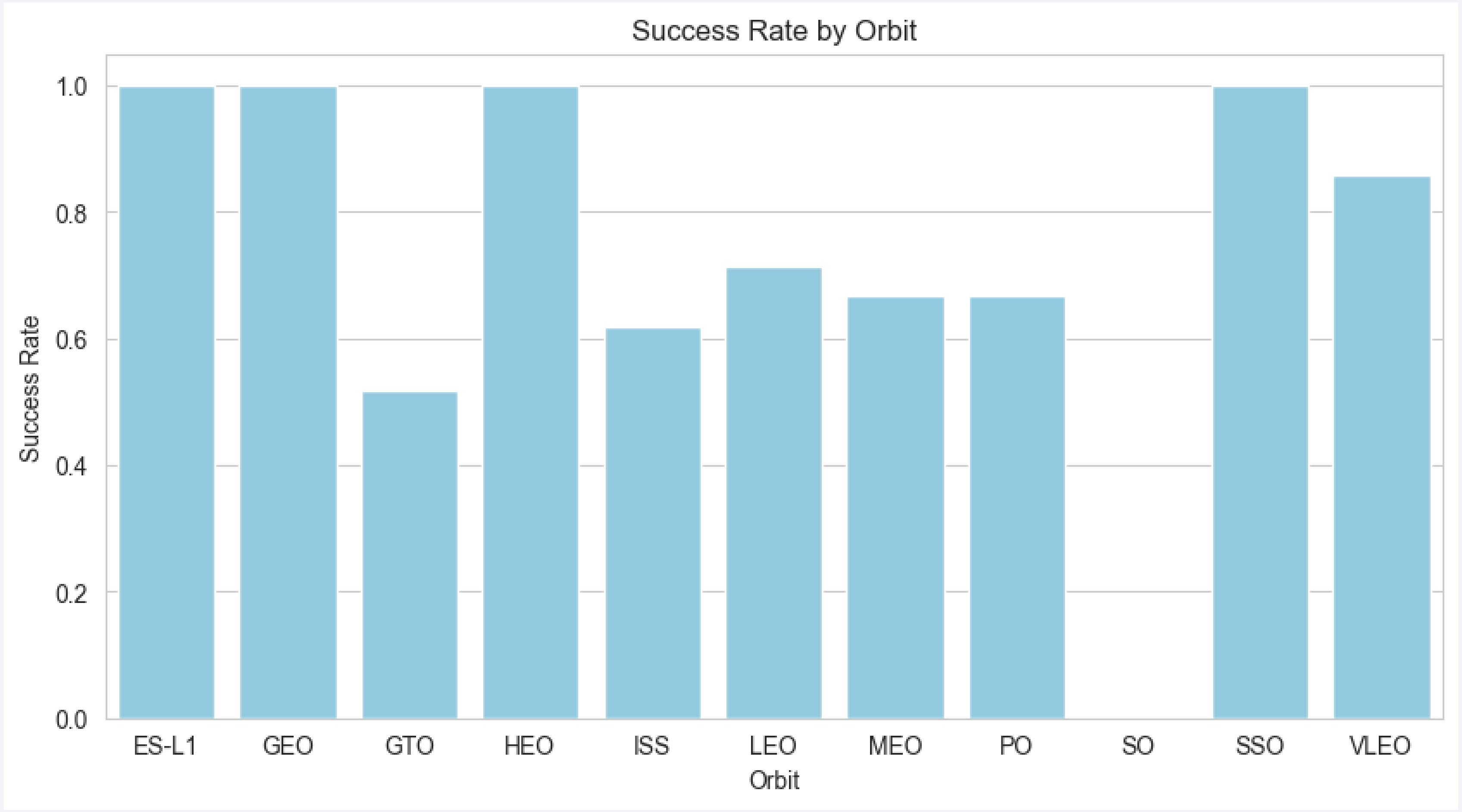
Flight Number vs. Launch Site



Payload vs. Launch Site



Success Rate vs. Orbit Type



Flight Number vs. Orbit Type

- Show a scatter point of
  Flight number vs. Orbit type

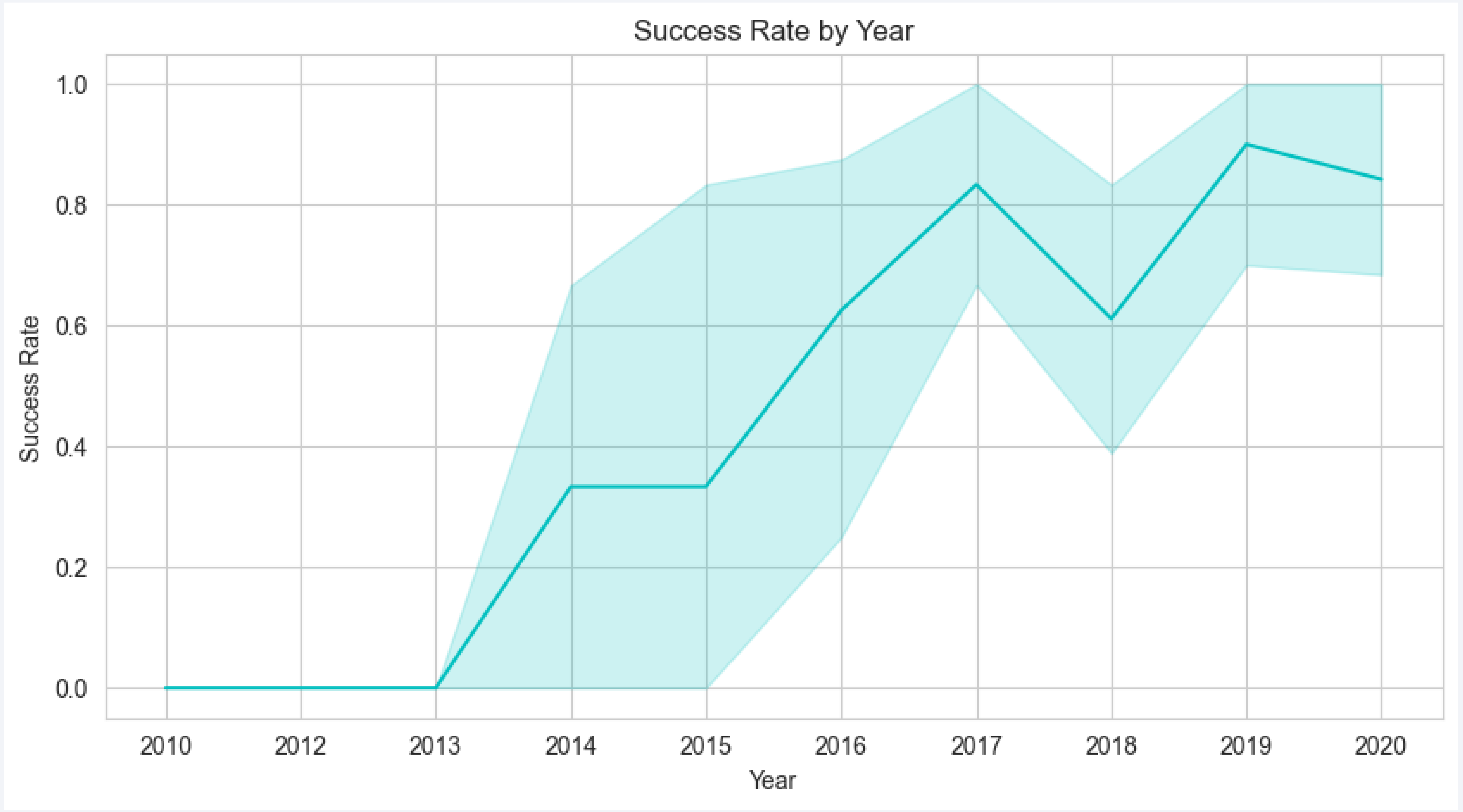
- Show the screenshot of the
  scatter plot with explanations

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

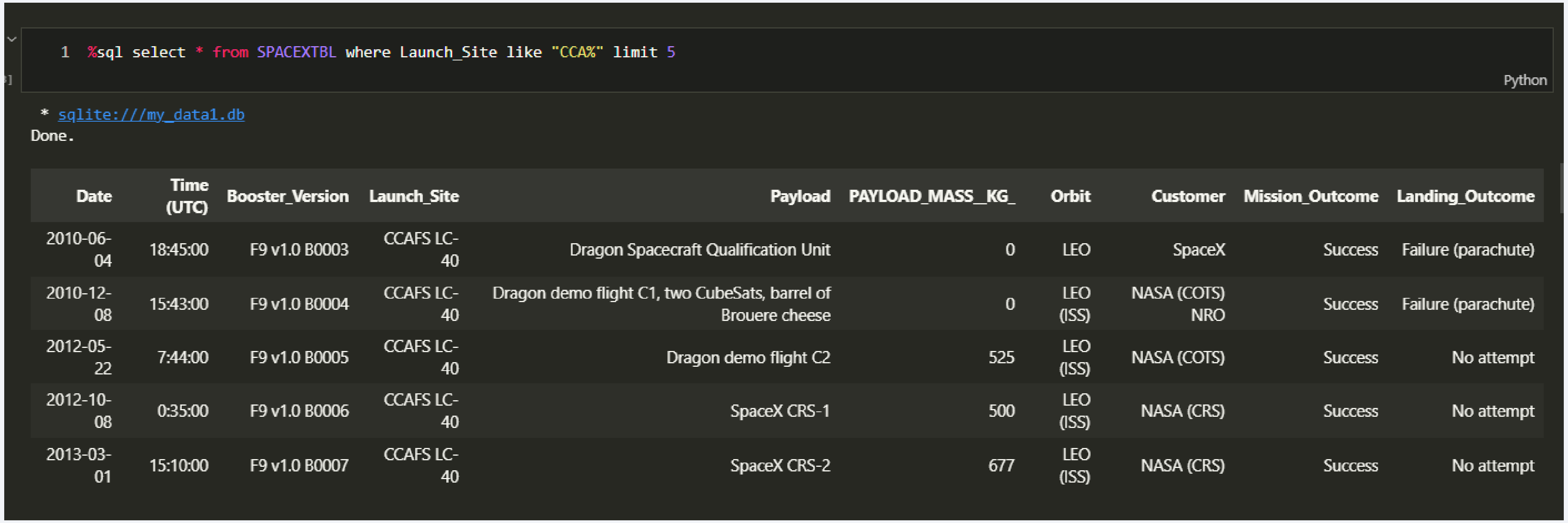
Launch Success Yearly Trend



All Launch Site Names

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

Launch Site Names Begin with 'CCA'



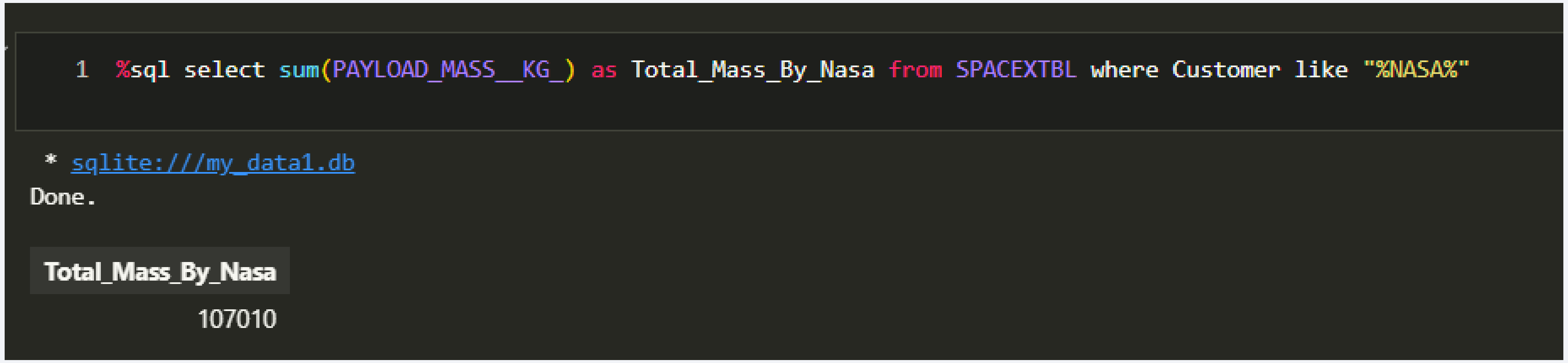
```
%sql select * from SPACEXTBL where Launch_Site like "CCA%" limit 5
```

Python

```
 * 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

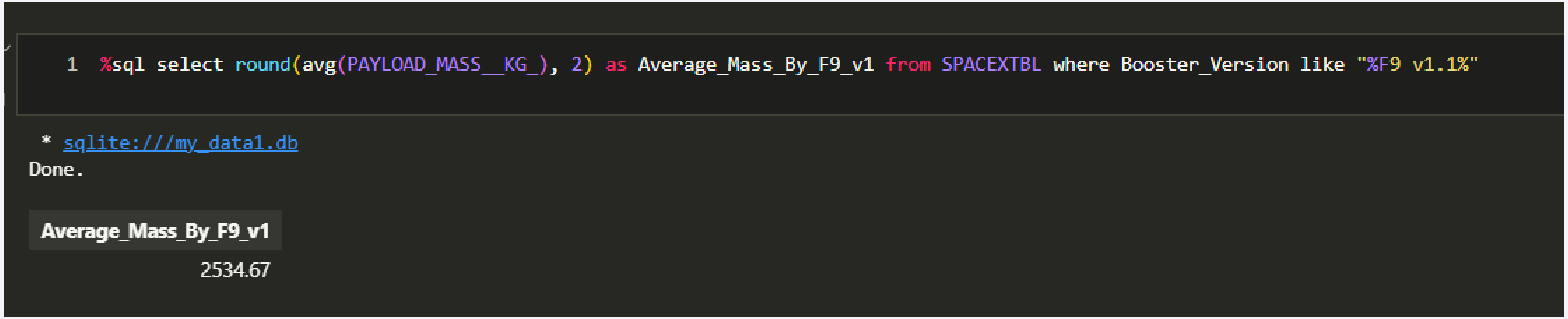


```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Mass_By_Nasa from SPACEXTBL where Customer like "%NASA%"
```

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

Total_Mass_By_Nasa
107010

Average Payload Mass by F9 v1.1

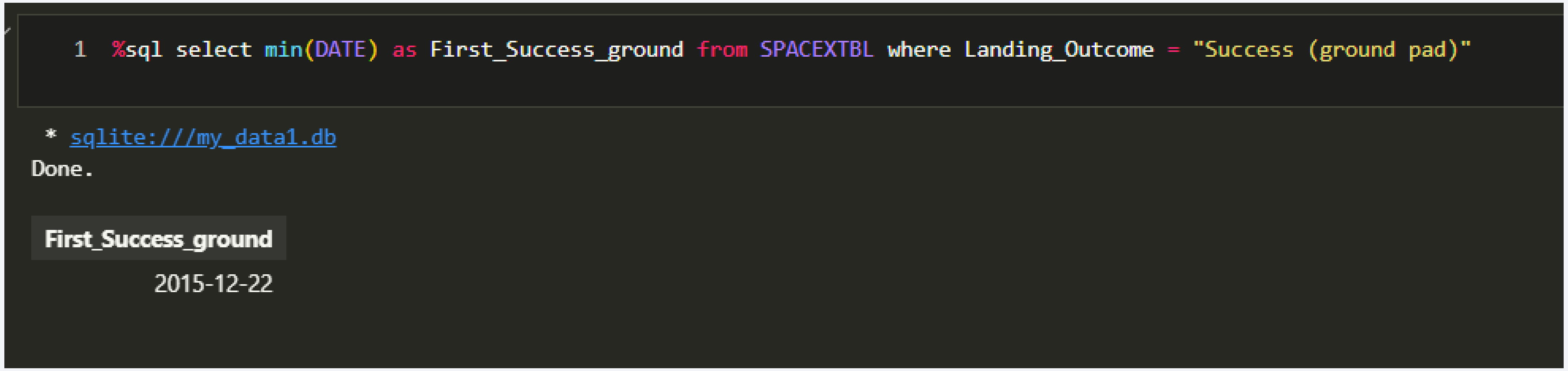


```
%sql select round(avg(PAYLOAD_MASS__KG_), 2) as Average_Mass_By_F9_v1 from SPACEXTBL where Booster_Version like "%F9 v1.1%"
```

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

Average_Mass_By_F9_v1
2534.67

First Successful Ground Landing Date

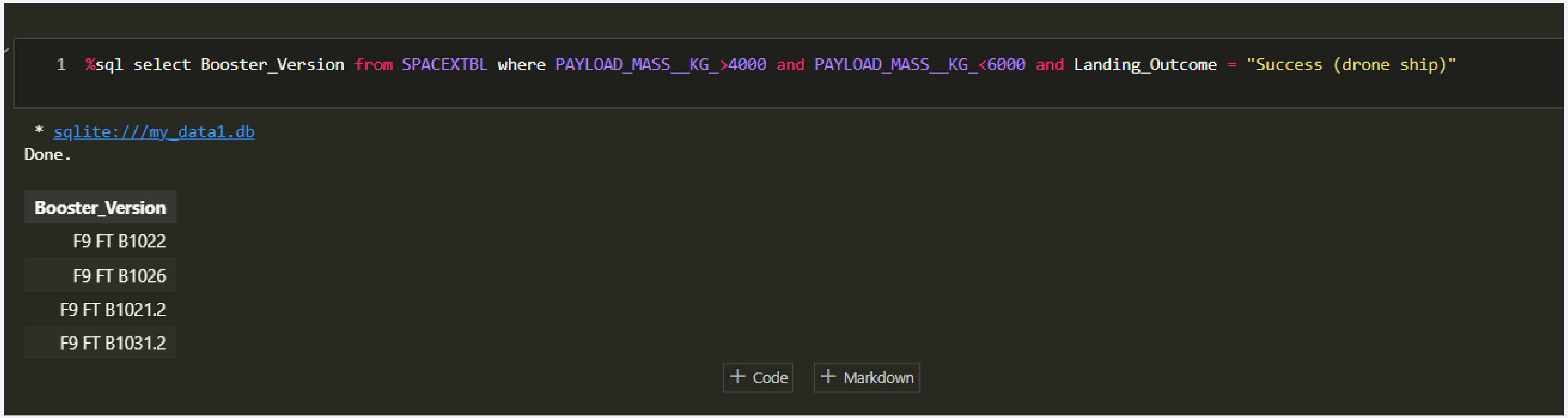


```
%sql select min(DATE) as First_Success_ground from SPACEXTBL where Landing_Outcome = "Success (ground pad)"
```

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

First_Success_ground
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

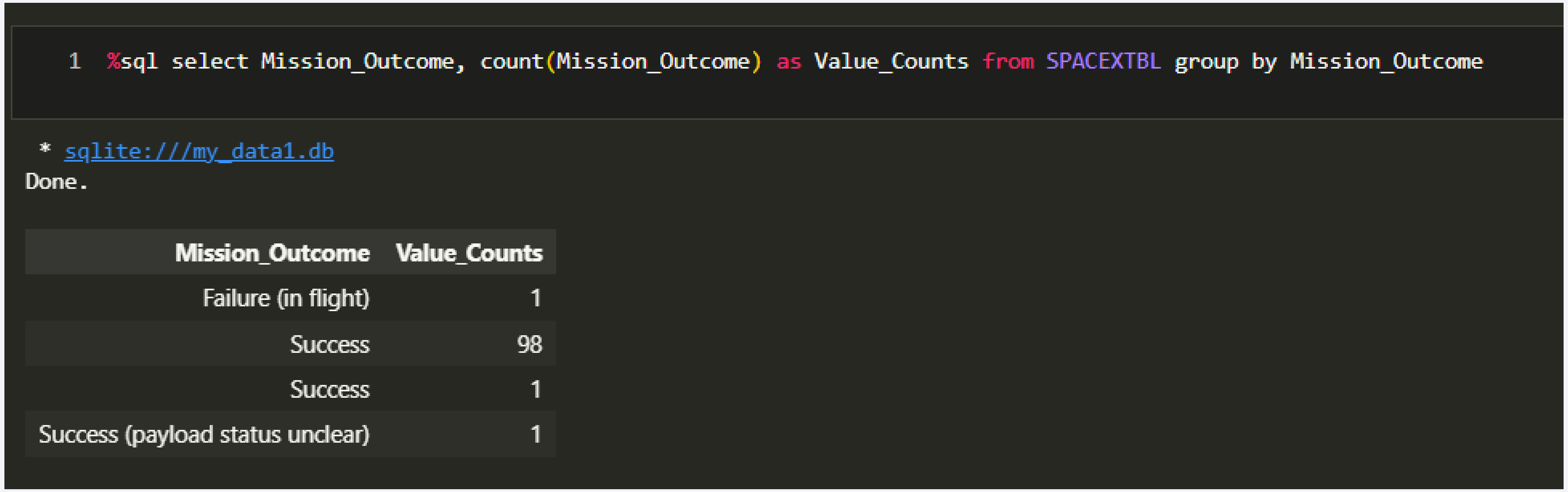


```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000 and Landing_Outcome = "Success (drone ship)"
```

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

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

Total Number of Successful and Failure Mission Outcomes

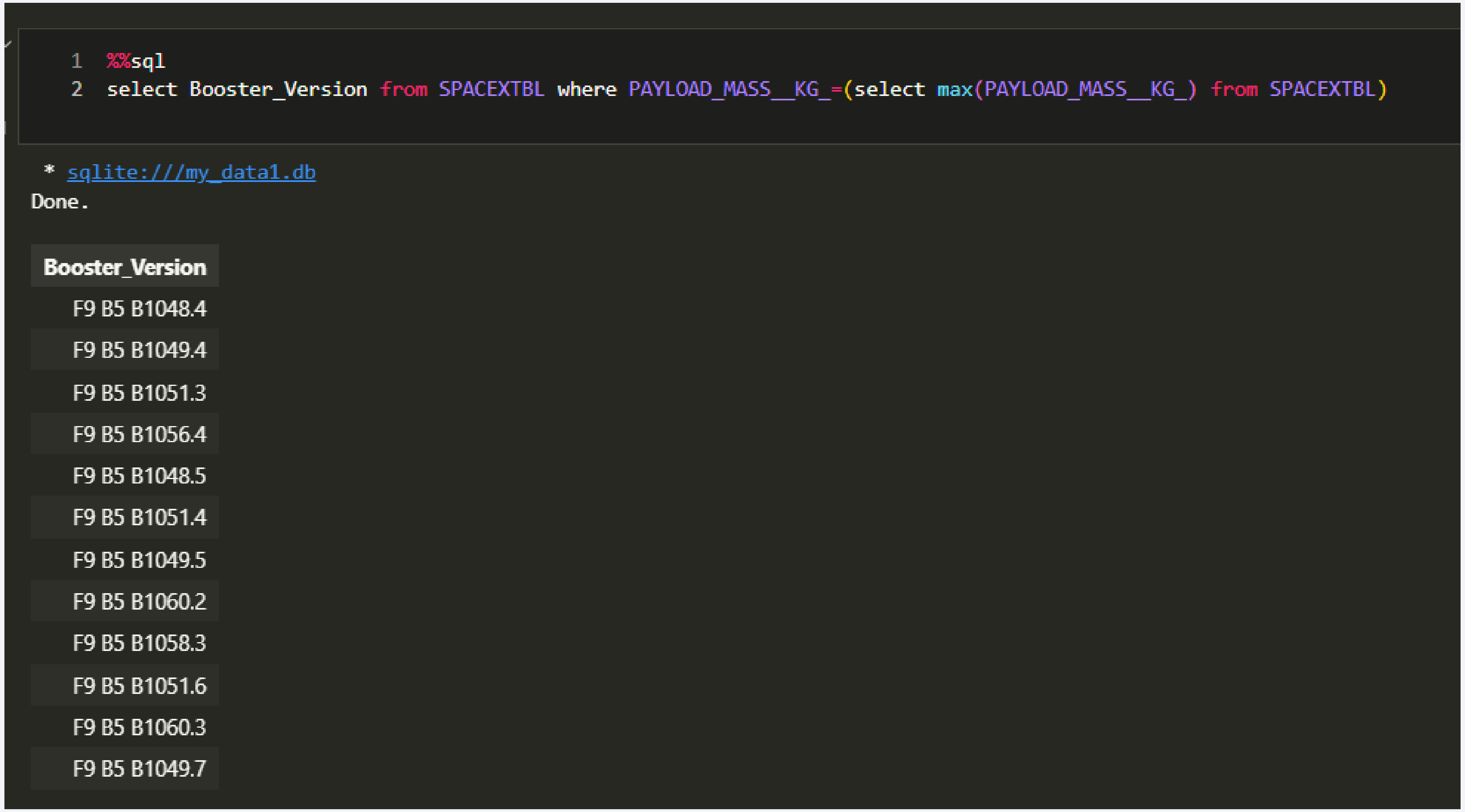


```
%sql select Mission_Outcome, count(Mission_Outcome) as Value_Counts from SPACEXTBL group by Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

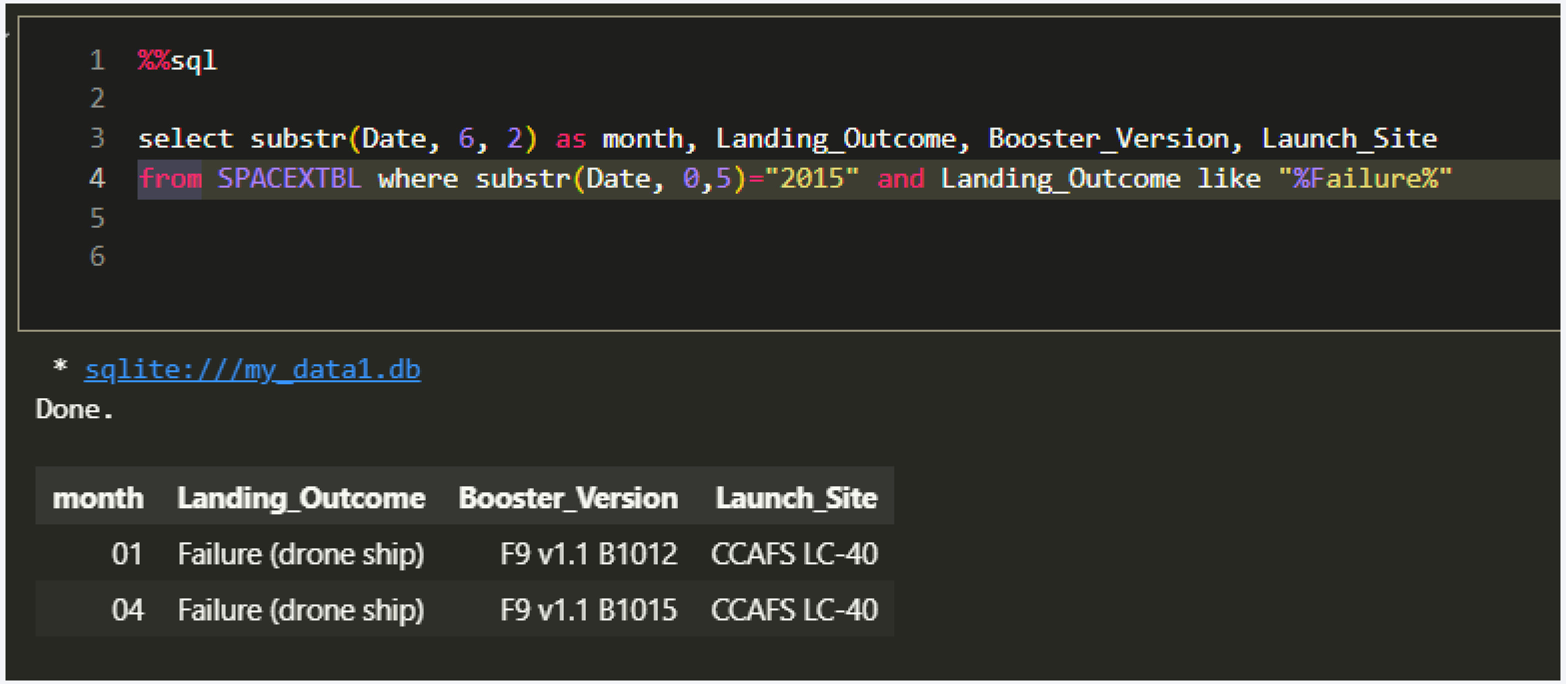


```
%%sql
select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * 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



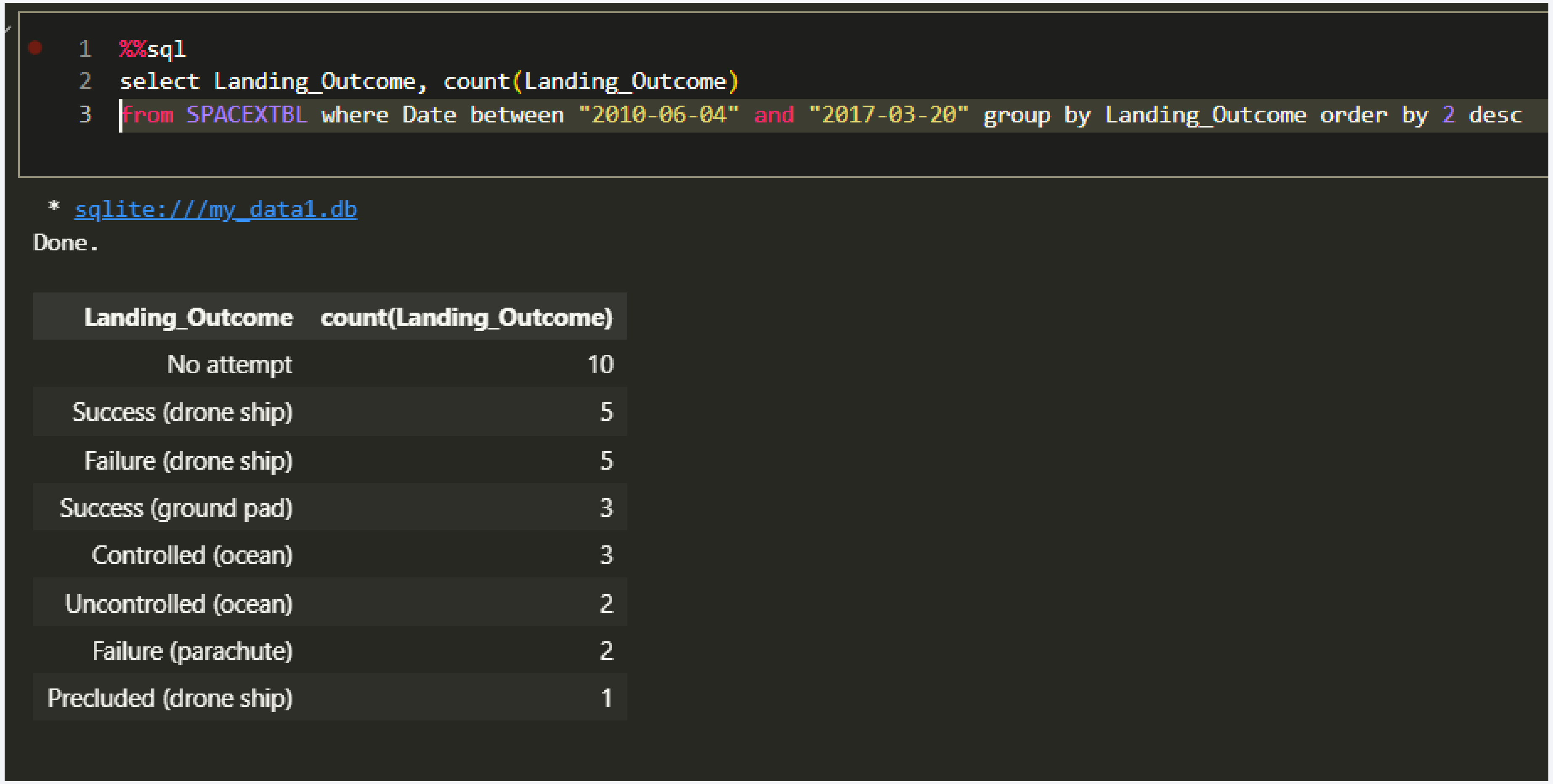
```
%%sql

select substr(Date, 6, 2) as month, Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTBL where substr(Date, 0,5)="2015" and Landing_Outcome like "%Failure%"
```

```
 * 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



```
%%sql
select Landing_Outcome, count(Landing_Outcome)
from SPACEXTBL where Date between "2010-06-04" and "2017-03-20" group by Landing_Outcome order by 2 desc
```

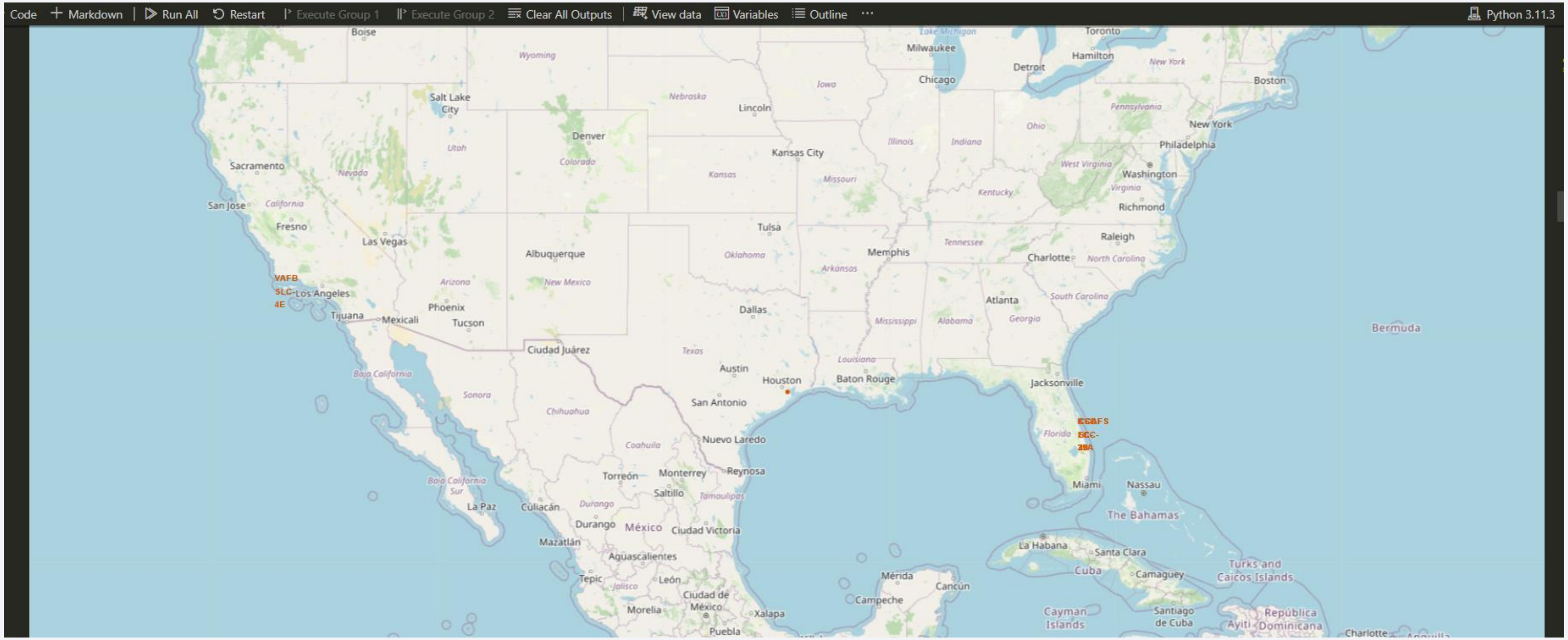
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count(Landing_Outcome)
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

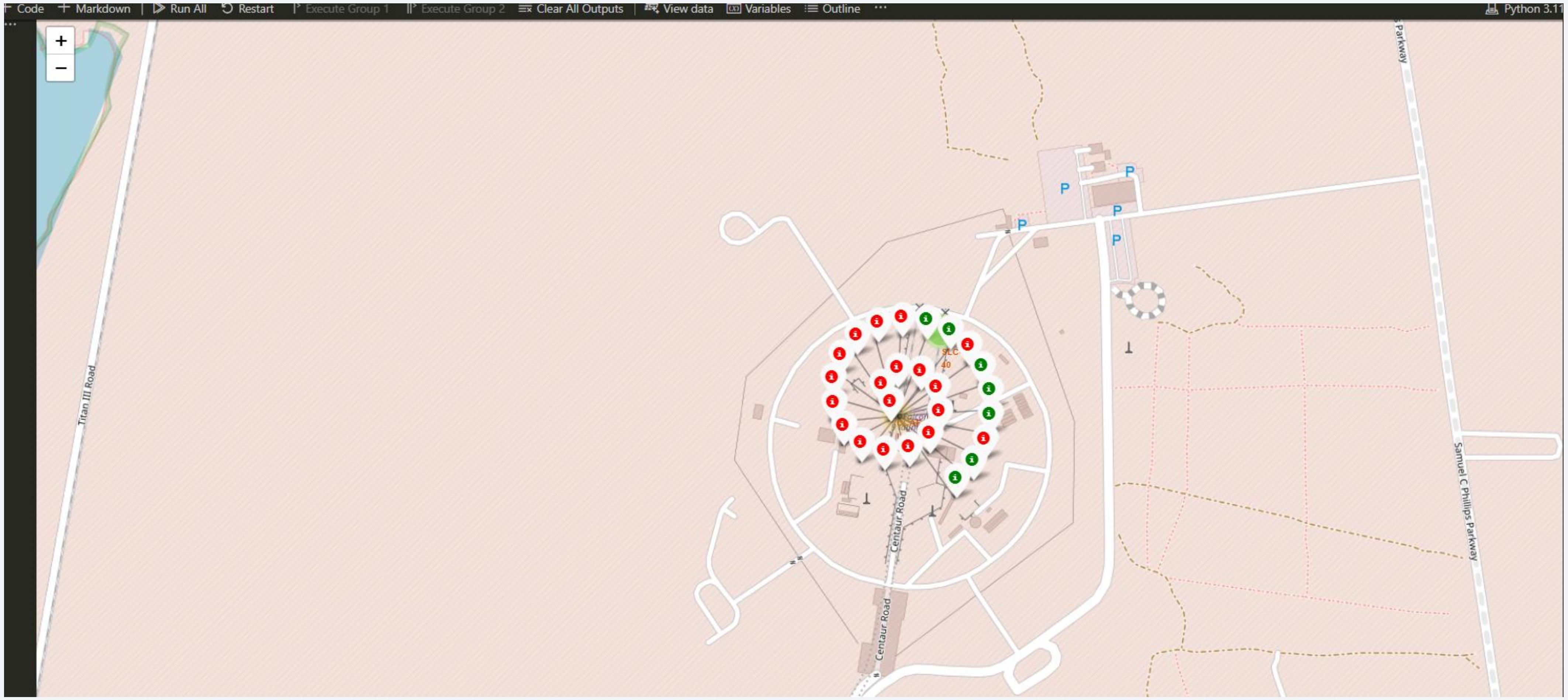
Section 3

Launch Sites Proximities Analysis

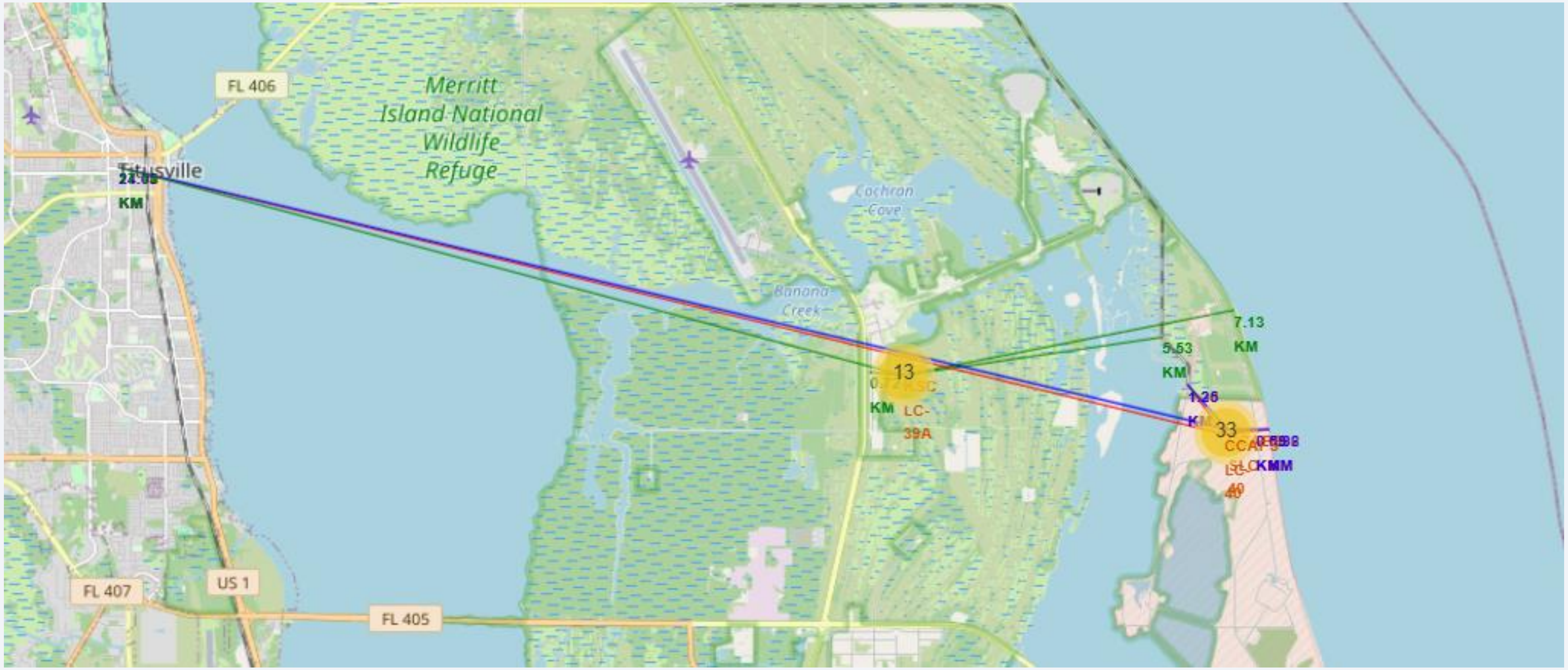
Launch Site Locations



Fail/Success Markers

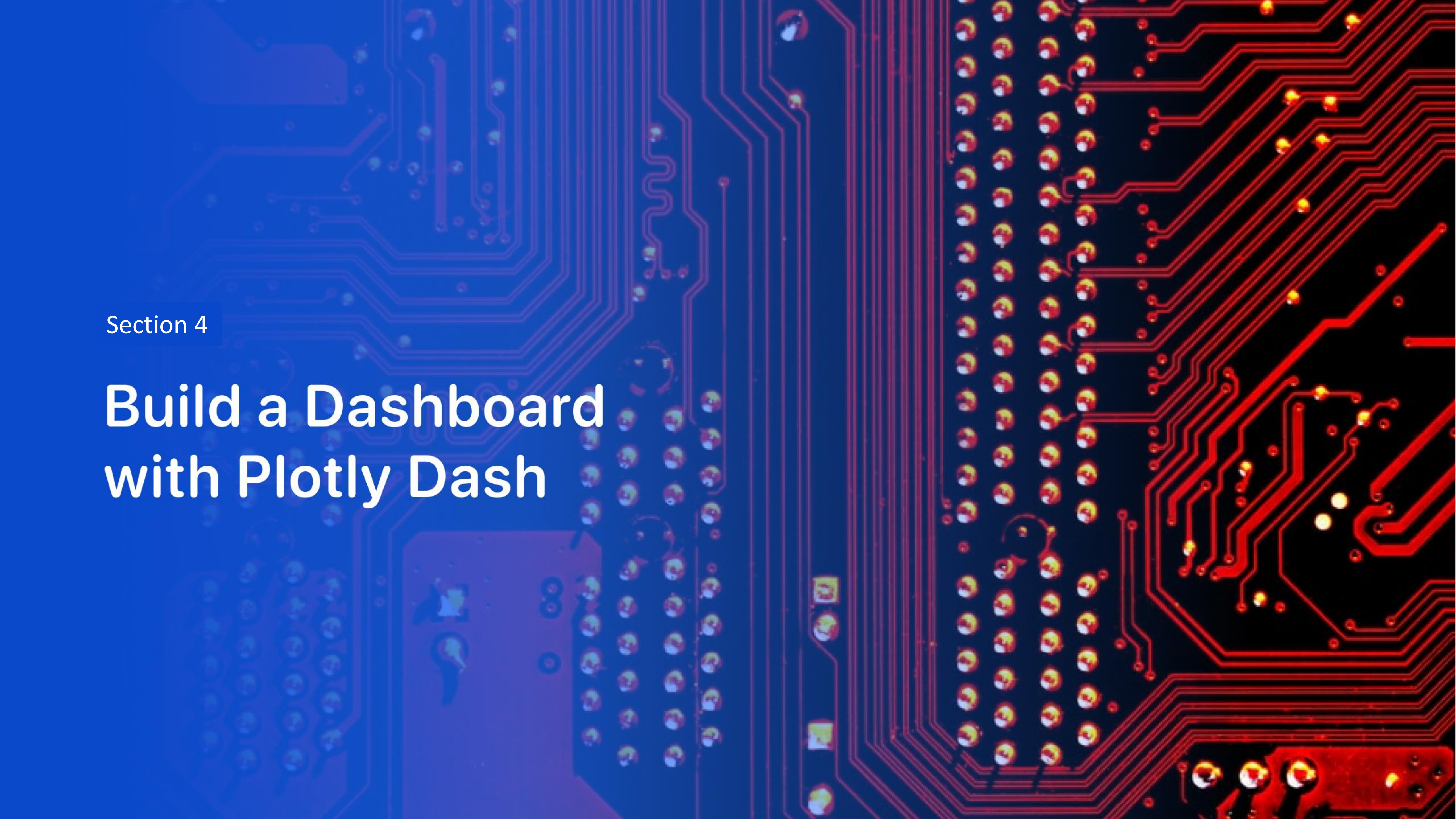


Proximity Map

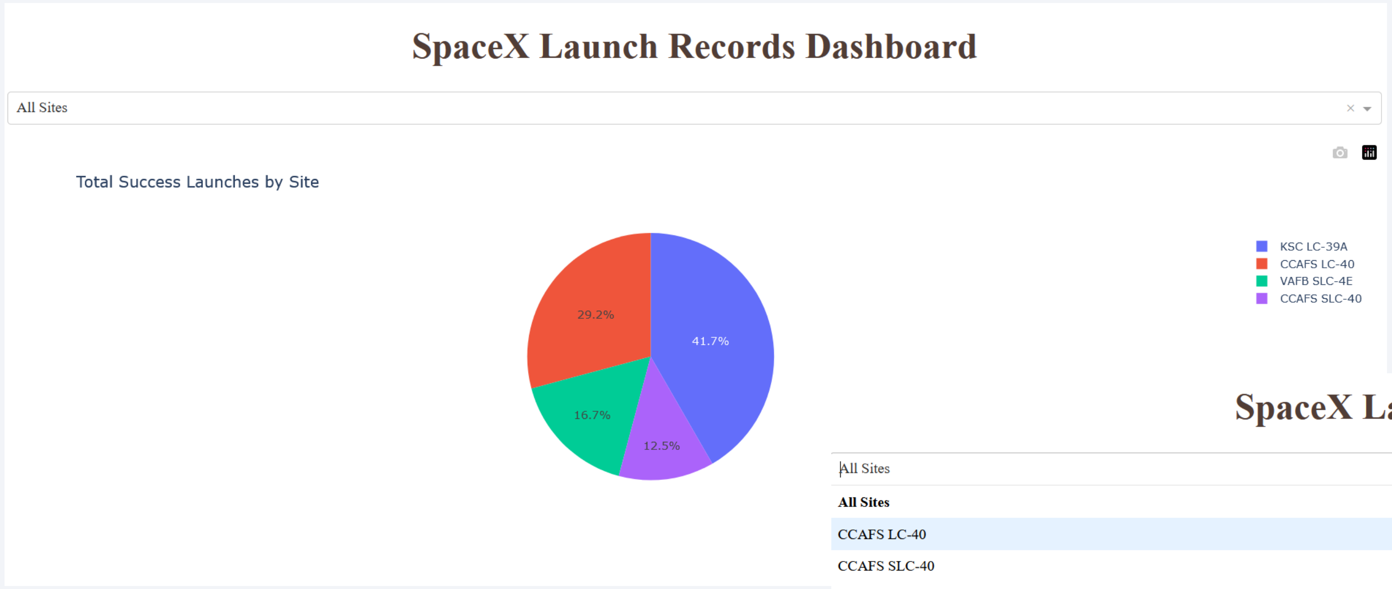


Section 4

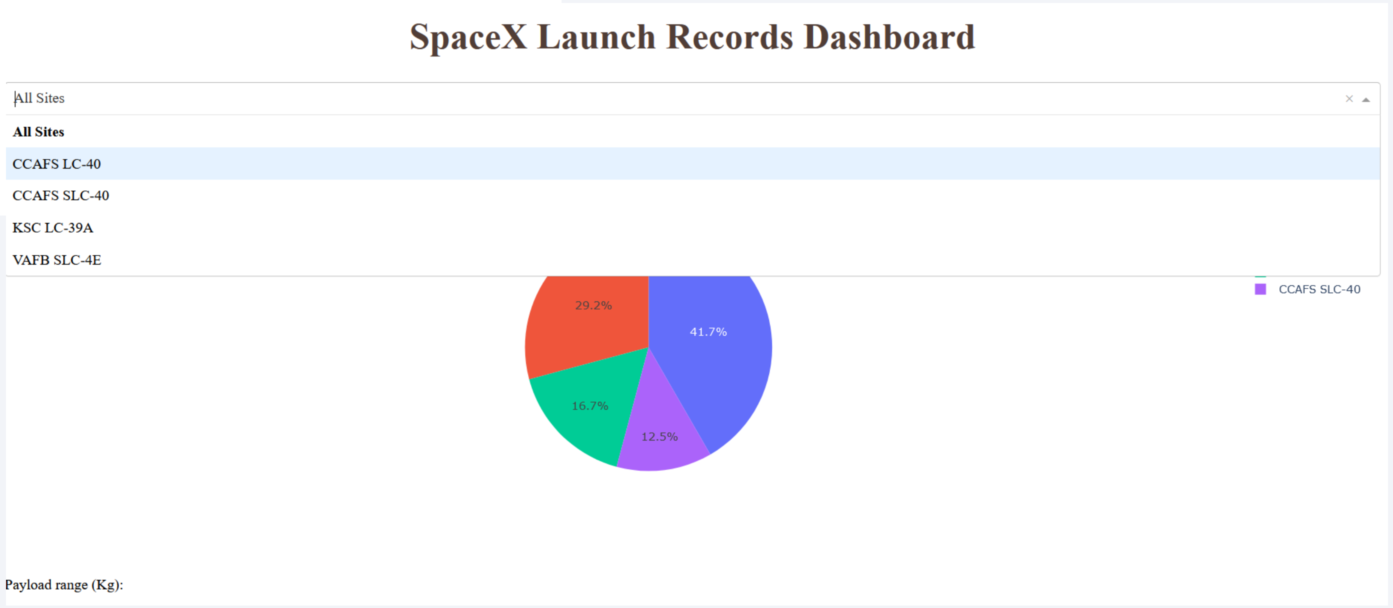
Build a Dashboard with Plotly Dash



Pie Chart


SpaceX Launch Records Dashboard

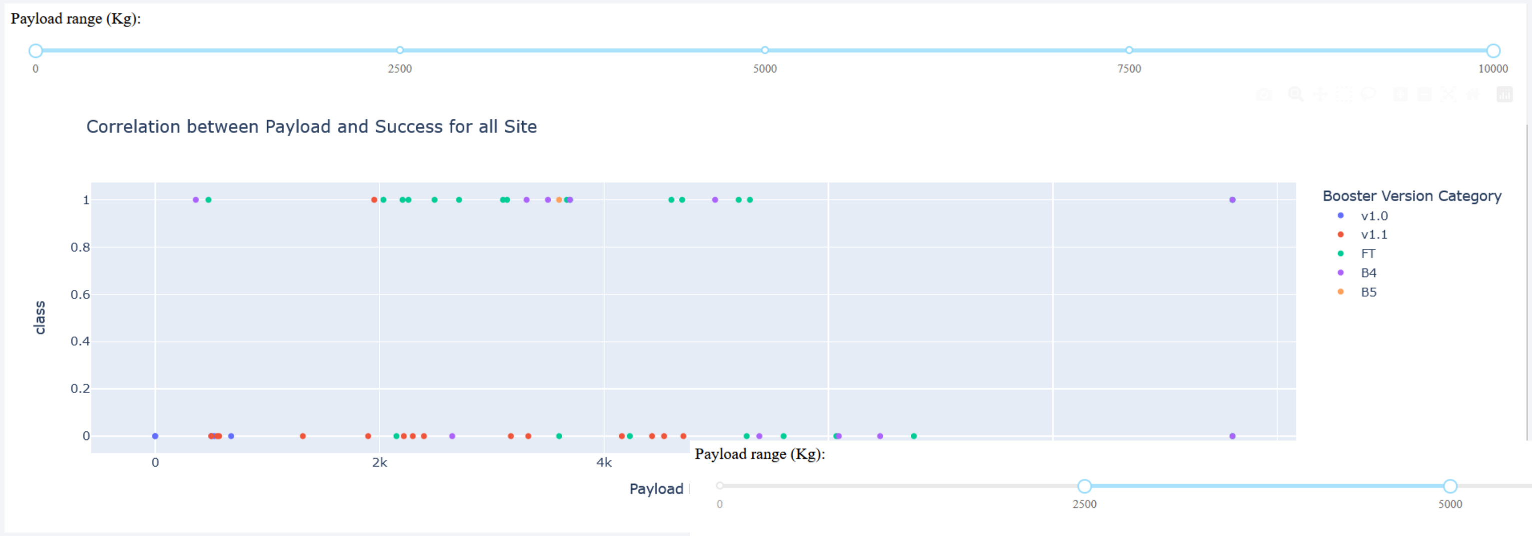
All Sites

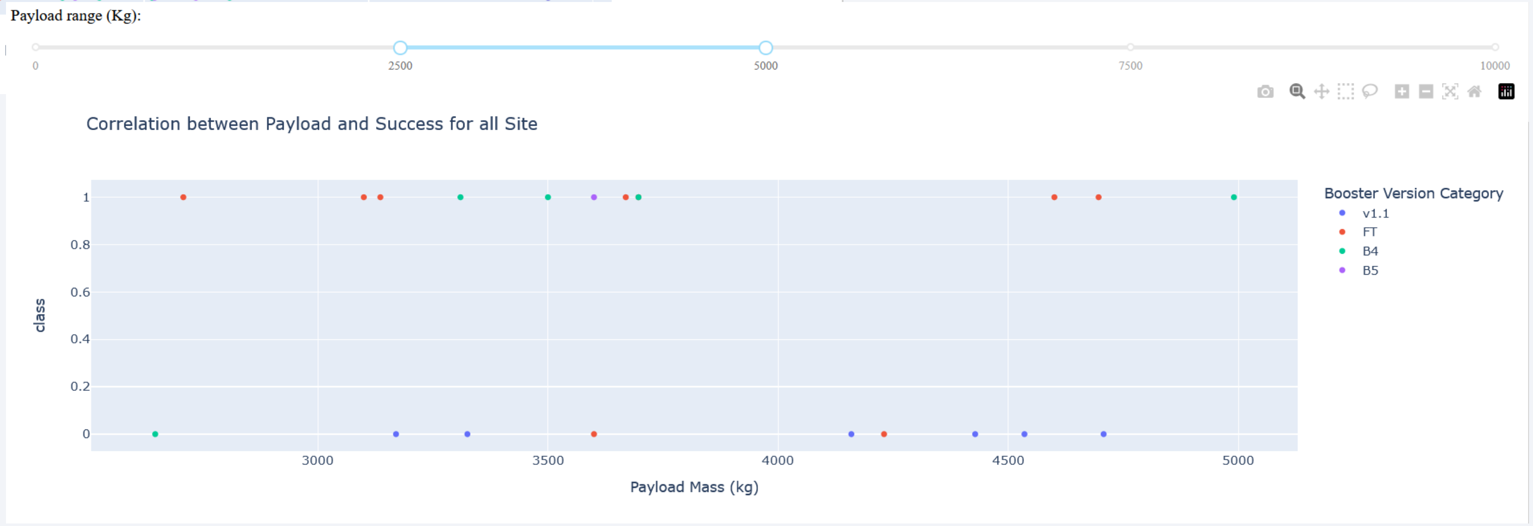

SpaceX Launch Records Dashboard

All Sites

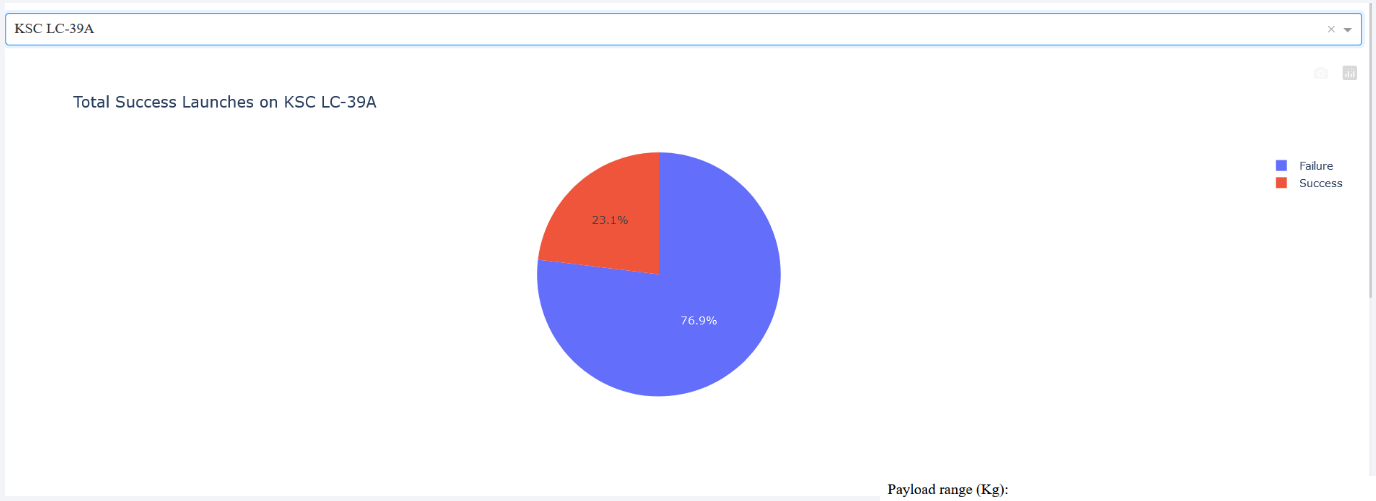
Payload range (Kg):

Payload

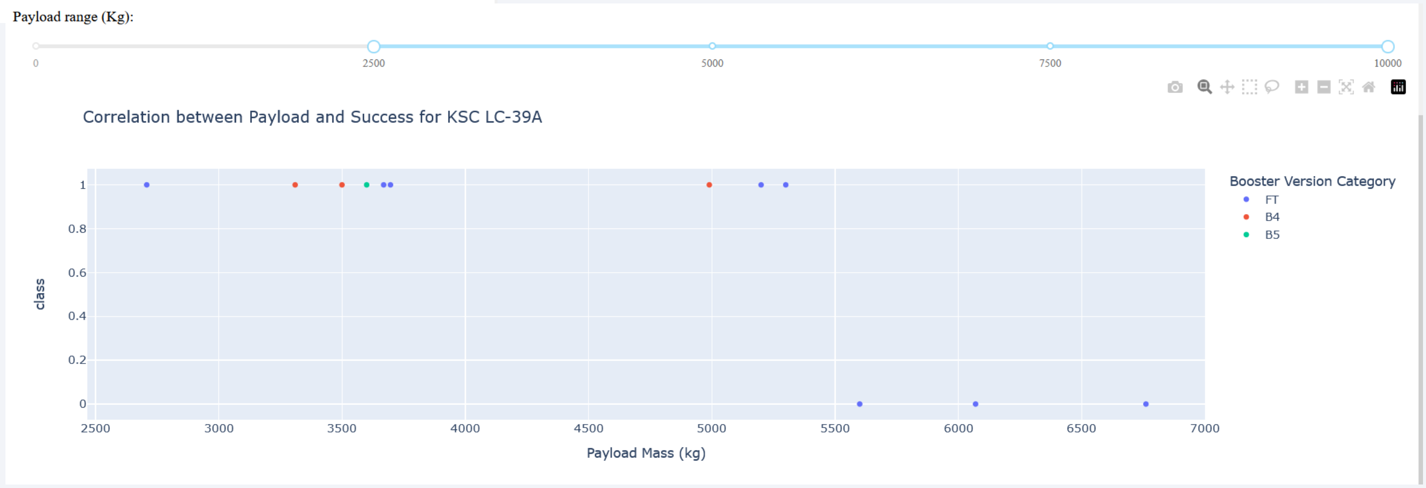

Payload range (Kg):
0
2500
5000
7500
10000
Correlation between Payload and Success for all Site
class
1
0.8
0.6
0.4
0.2
0
0
2k
4k
Payload
Booster Version Category
v1.0
v1.1
FT
B4
B5


Payload range (Kg):
0
2500
5000
7500
10000
Correlation between Payload and Success for all Site
class
1
0.8
0.6
0.4
0.2
0
3000
3500
4000
4500
5000
Payload Mass (kg)
Booster Version Category
v1.1
FT
B4
B5

Other Options



KSC LC-39A

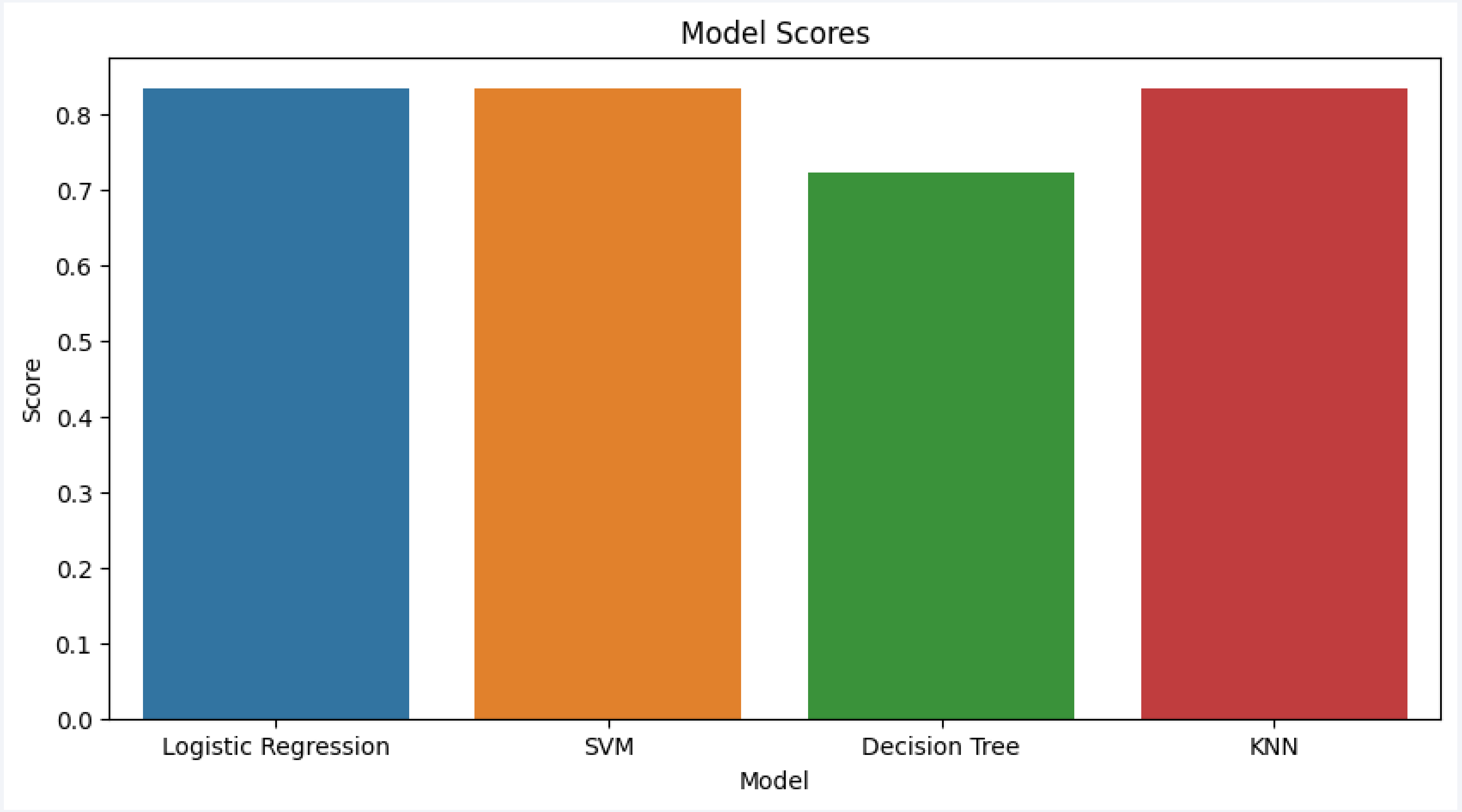


Payload range (Kg):

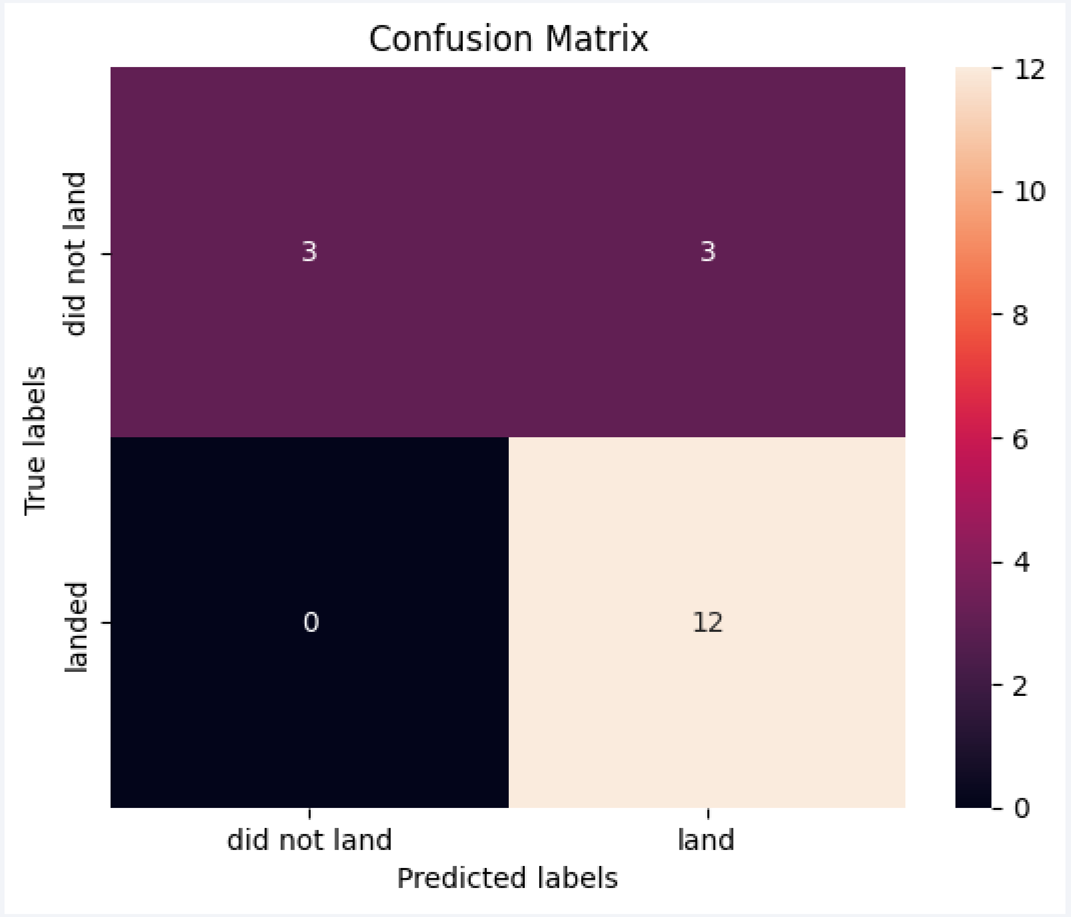
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

This was a really long journey, but in the end, we did what we wanted.

After all the hardest part is to deliver thought to your audience!

Setting questions is more of a fine fantasy that can become real, and my job is to make it look real. After all, space is not as far as it used to be; maybe it is only a small showcase project now, but who knows what we will see in the next decade.

So long.

Appendix

Some scripts are all in notebooks. All notebooks can be found at my git in IBM_assignments repo at capstone dir. You are welcome.

Thank you!