IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Seitaro Miyashita
4th February 2025

➢ Outline

- ✓ Executive Summary
- ✓ Introduction
- ✓ Methodology
- ✓ Results
- ✓ Conclusion

➢ Executive Summary

✓ Summary of methodologies

-Data Collection through API

-Data Collection with Web Scraping

-Data Wrangling

-Exploratory Data Analysis with SQL

-Exploratory Data Analysis with Data Visualization

-Interactive Visual Analytics with Folium

-Machine Learning Prediction

✓ Summary of all results

-Exploratory Data Analysis result

-Interactive analytics in screenshots

- Predictive Analytics result

➢ Introduction

✓ Project background and context

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

✓ Problems you want to find answers

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

➢ Methodology

Executive Summary

✓Data collection methodology:

Data was collected using SpaceX API and web scraping from Wikipedia.

✓Perform data wrangling

One-hot encoding was applied to categorical features

✓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 collected using various methods**

-Data collection was done using get request to the SpaceX API.

-Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().

-We then cleaned the data, checked for missing values and fill in missing values where Necessary.

-In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.

-The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

➢ Data Collection – SpaceX API

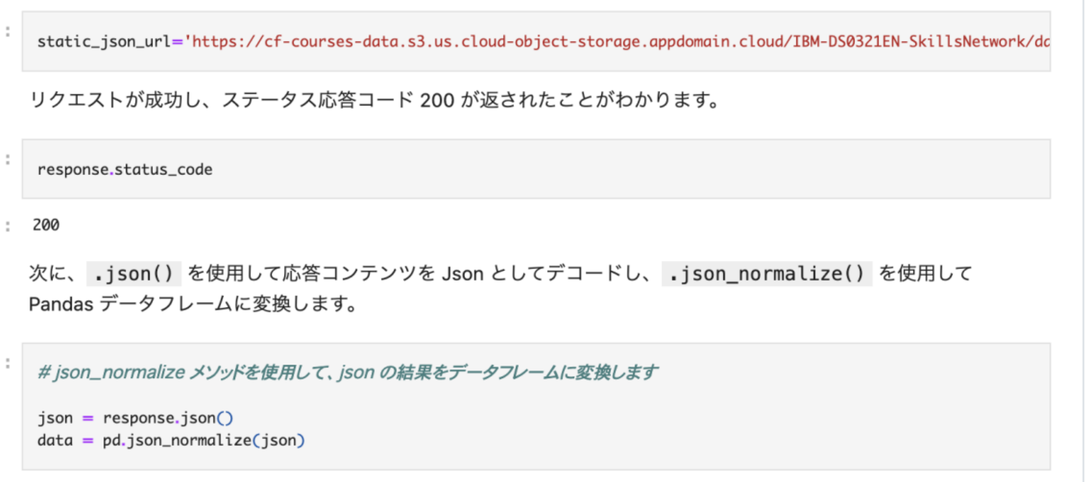
- ✓ We made a GET request to the SpaceX API to collect data. Then, we cleaned the retrieved data and performed basic data wrangling and formatting.

- ✓ The link to the notebook is here:

https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/b7350de69e752c8b8b245c4772533757f1c9fed5/SpaceX%20Data%20Collection%20API%20v2.ipynb

※language is Japanese.

▪Call API



```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/dc
```

リクエストが成功し、ステータス応答コード 200 が返されたことがわかります。

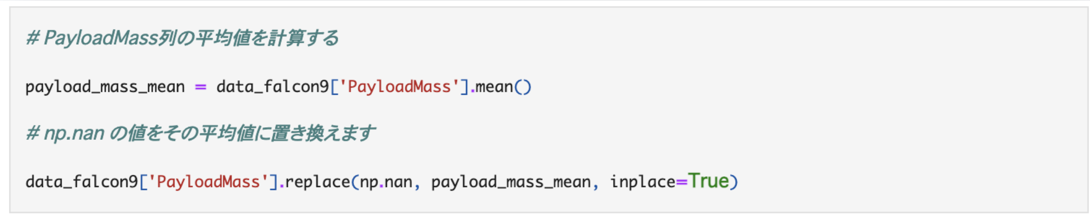
```
response.status_code
```

```
200
```

次に、 `.json()` を使用して応答コンテンツを Json としてデコードし、 `.json_normalize()` を使用して Pandas データフレームに変換します。

```
# json_normalize メソッドを使用して、json の結果をデータフレームに変換します

json = response.json()
data = pd.json_normalize(json)
```

▪Example: replacing missing values



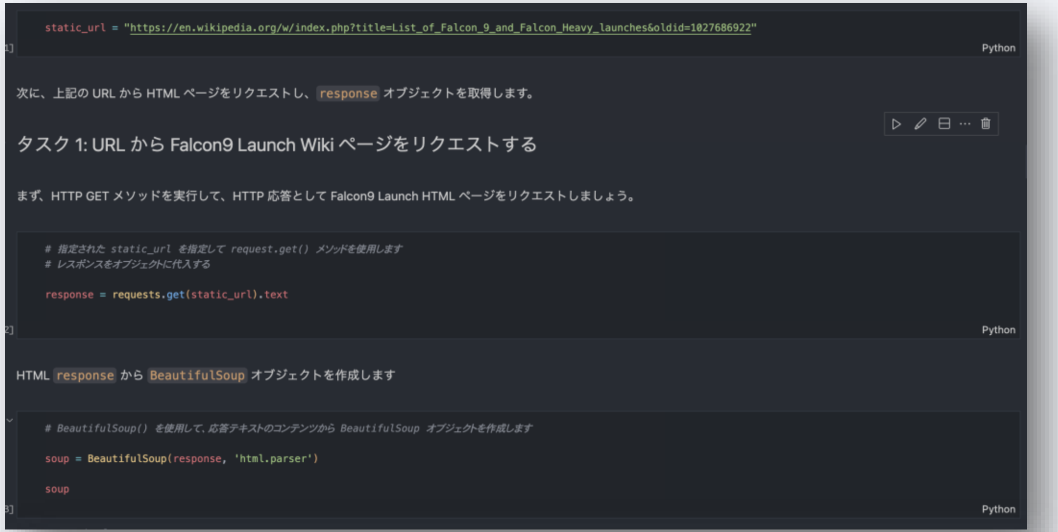
```
# PayloadMass列の平均値を計算する

payload_mass_mean = data_falcon9['PayloadMass'].mean()

# np.nan の値をその平均値に置き換えます

data_falcon9['PayloadMass'].replace(np.nan, payload_mass_mean, inplace=True)
```

➤ Data Collection - Scraping

- ✓ We applied web scraping to scrape Falcon 9 launch records using BeautifulSoup.

- ✓ We parsed the table and converted it into a pandas DataFrame.

- ✓ The link to the notebook is here: https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/5471076e8785e9ca23328f09d95f6be27257f51e/Jupyter%20Labs%20Web%20Scraping.ipynb



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Python

次に、上記の URL から HTML ページをリクエストし、response オブジェクトを取得します。

タスク 1: URL から Falcon9 Launch Wiki ページをリクエストする

まず、HTTP GET メソッドを実行して、HTTP 応答として Falcon9 Launch HTML ページをリクエストしましょう。

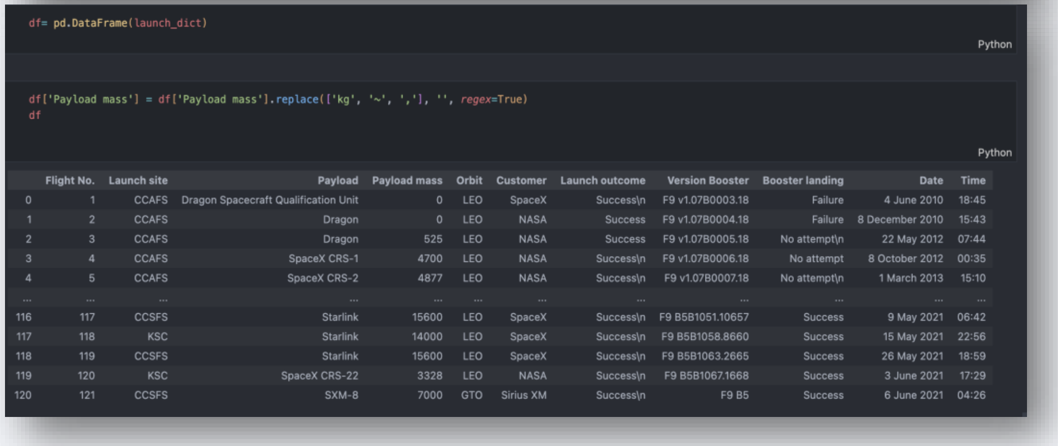
```
# 指定された static_url を指定して request.get() メソッドを使用します
# レスポンスをオブジェクトに代入する

response = requests.get(static_url).text
```

Python

HTML response から BeautifulSoup オブジェクトを作成します

```
# BeautifulSoup() を使用して、応答テキストのコンテンツから BeautifulSoup オブジェクトを作成します

soup = BeautifulSoup(response, 'html.parser')

soup
```

Python



```
df= pd.DataFrame(launch_dict)
```

Python

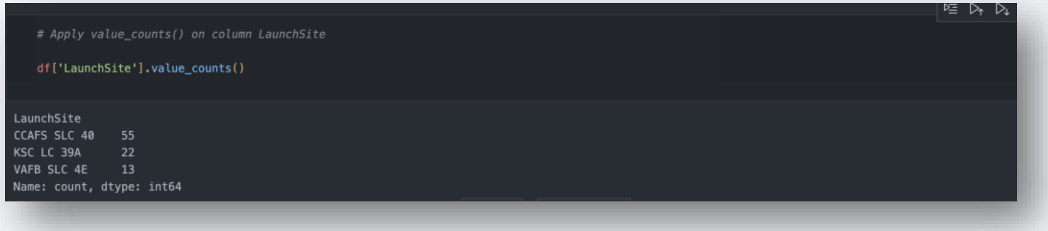
```
df['Payload mass'] = df['Payload mass'].replace(['kg', '~', ','], '', regex=True)
df
```

Python

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0004.18	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525	LEO	NASA	Success	F9 v1.07B0005.18	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4700	LEO	NASA	Success\n	F9 v1.07B0006.18	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4877	LEO	NASA	Success\n	F9 v1.07B0007.18	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...
116	117	CCSFS	Starlink	15600	LEO	SpaceX	Success\n	F9 B5B1051.10657	Success	9 May 2021	06:42
117	118	KSC	Starlink	14000	LEO	SpaceX	Success\n	F9 B5B1058.8660	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15600	LEO	SpaceX	Success\n	F9 B5B1063.2665	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3328	LEO	NASA	Success\n	F9 B5B1067.1668	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7000	GTO	Sirius XM	Success\n	F9 B5	Success	6 June 2021	04:26

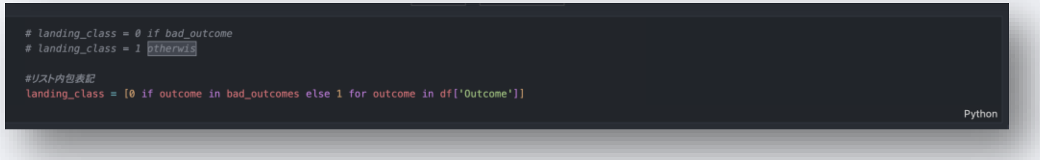
➢ Data Wrangling

- ✓ We performed exploratory data analysis and determined the training labels.
- ✓ We calculated the number of launches at each site, and the number and occurrence of each orbit.
- ✓ We created landing outcome labels from the outcome column and exported the results to a CSV file.
- ✓ The link to the notebook is here: https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/a5dd8ec34a5f6cfe7d36f9b572fc41d683124626/SpaceX%20Data%20Wrangling%20v2.ipynb

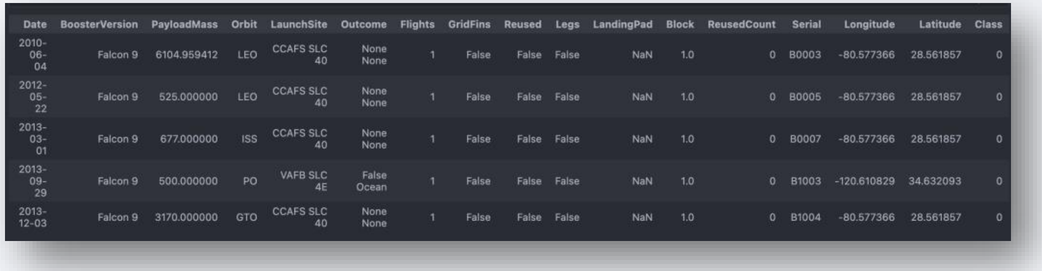


```python
# Apply value_counts() on column LaunchSite

df['LaunchSite'].value_counts()
```

```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```



```python
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwis

#リスト内包表記
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```



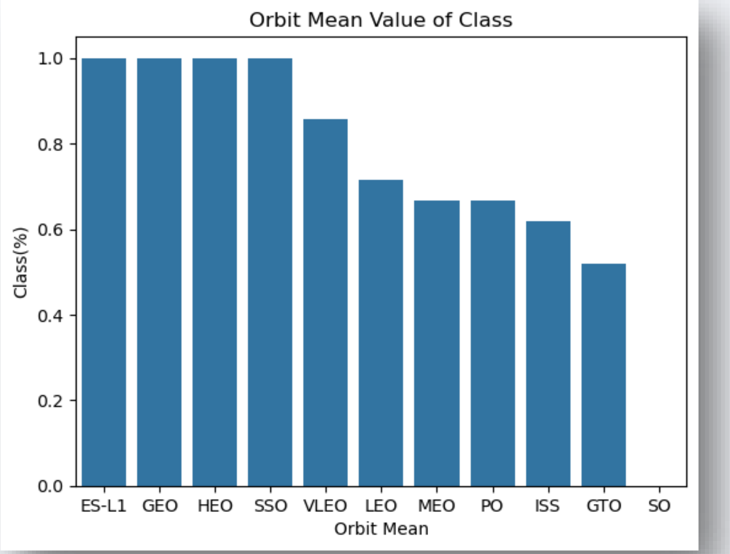
Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

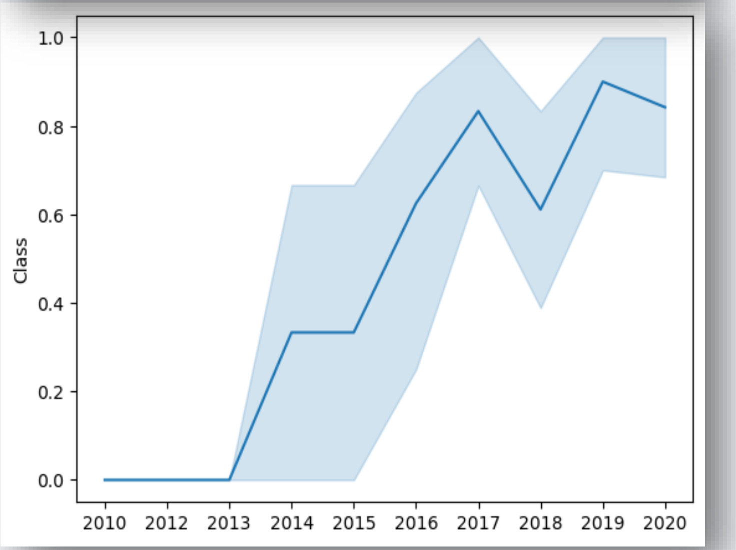
➢ EDA with Data Visualization

- ✓ We explored the data by visualizinghe relationship between flight number and launch Site, payload andlaunch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

- ✓ The link to the notebook is here:

[https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/c2a764edca0b630c58b608e82c303d83d59dd94d/Jupyter%20Labs%20EDA%20Data viz%20v2.ipynb](https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/c2a764edca0b630c58b608e82c303d83d59dd94d/Jupyter%20Labs%20EDA%20Dataviz%20v2.ipynb)





➢ EDA with SQL

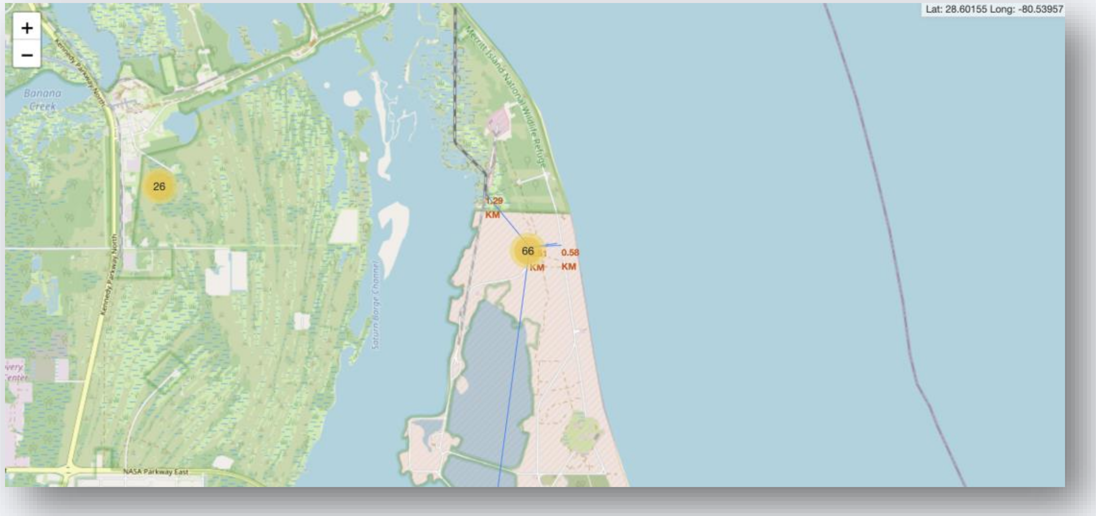
- ✓ We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- ✓ We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

-The names of the unique launch sites in the space mission.
-The total payload mass carried by boosters launched by NASA (CRS).
-The average payload mass carried by[1] the booster version F9 v1.1
-The total number of successful and failed mission outcomes.
-The failed landing outcomes on the drone ship, their booster version, and launch site names.

- ✓ The link to the notebook is here: https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/c2a764edca0b630c58b608e82c303d83d59dd94d/Jupyter%20Labs%20EDA%20SQL.ipynb

➢ Build an Interactive Map with Folium

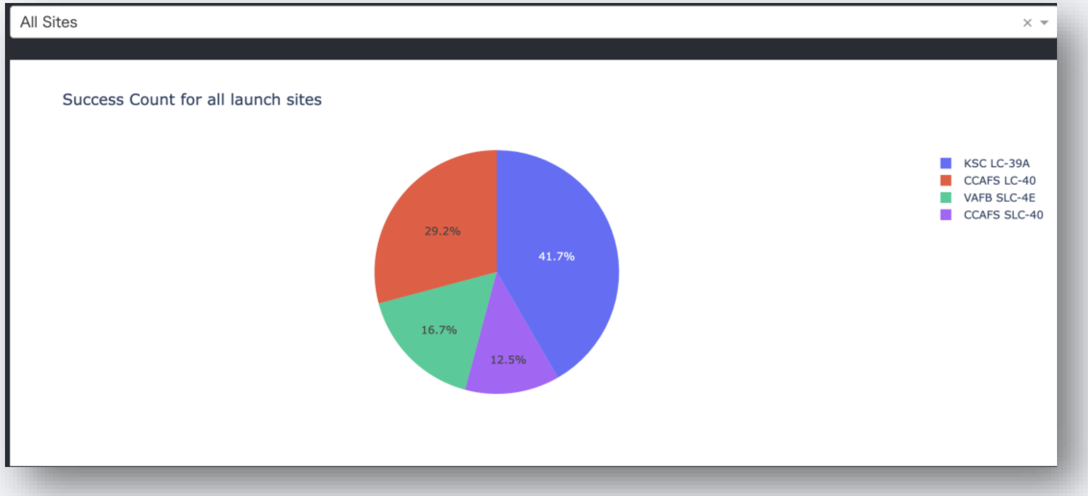
- ✓ We marked all launch sites, and added map objects such as markers, circles, lines tomark the success or failure of launches for each site on the folium map.
- ✓ •We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 or failure, and 1 for success.
- ✓ Using the color-labeled marker clusters, we identified which launch sites have
- ✓ relatively high success rate.
- ✓ We calculated the distances between a launch site to its proximities.



- ✓ The link to the notebook is here: https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/cb8b0ef359bc77bb40e58dca75f44b8e3aa2cbe3/Jupyter%20Lab%20Launch%20Site%20Location%20v2.ipynb

➢ Build a Dashboard with Plotly Dash

- ✓ We built an interactive dashboard using Plotly Dash.
- ✓ We plotted pie charts showing the total launches by site.
- ✓ We plotted a scatter plot showing the relationship between Outcome and Payload Mass (kg) for the different booster versions.
- ✓ The link to the notebook is here: https://github.com/Seitaro1012/IBM-Data-Science-SpaceX/blob/7ba68525c3b4ea42d82ce2985b0c0c6db933e172/SpaceX%20Dashboard%20with%20Plotly%20Dash.ipynb



All Sites

➢ Predictive Analysis (Classification)

- ✓ We loaded the data using NumPy and pandas, transformed the data, and split it into training and testing sets.
- ✓ We built different machine learning models and tuned their hyperparameters using GridSearchCV.
- ✓ We used accuracy as the evaluation metric for our models and improved performance through feature engineering and algorithm tuning.
- ✓ We identified the best-performing classification model.

➢ Results

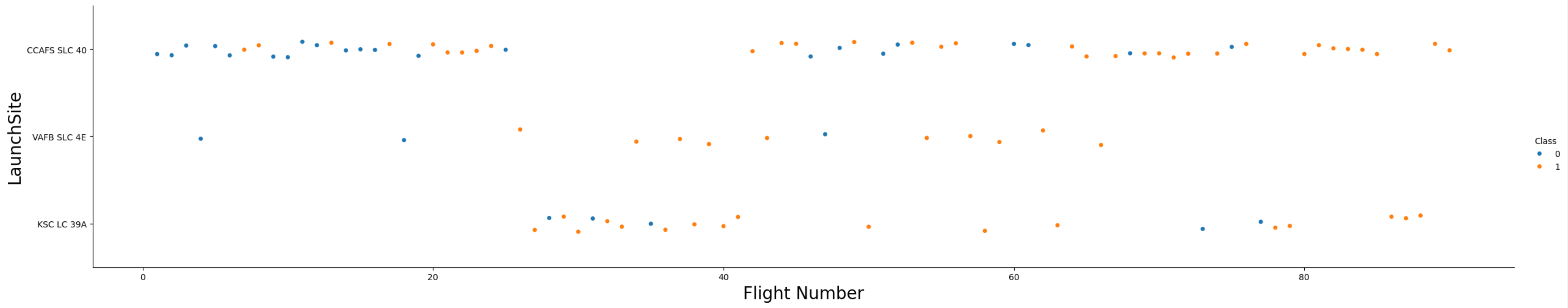
- ✓ Exploratory data analysis results
- ✓ Interactive analytics demo in screenshots
- ✓ Predictive analysis results

Section 2

Insights drawn from EDA

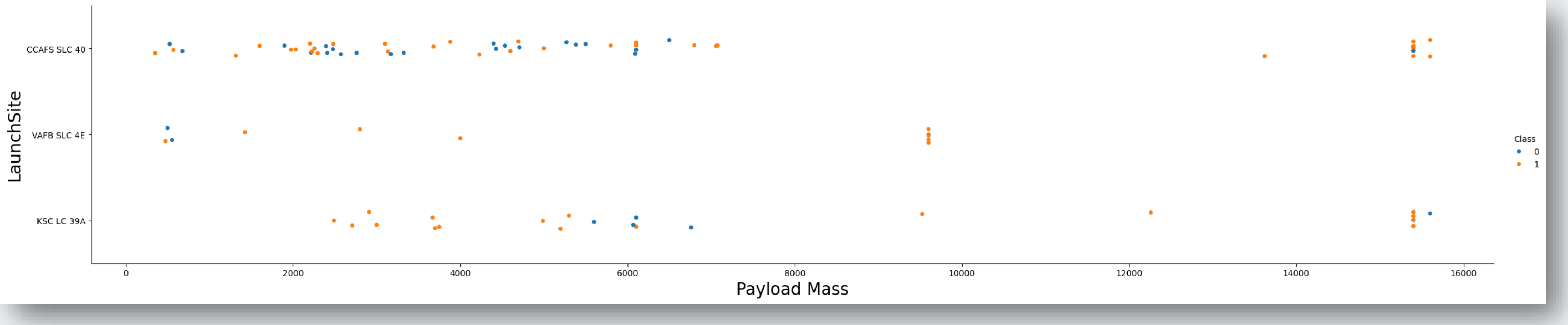
➢ Flight Number vs. Launch Site

✓ From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



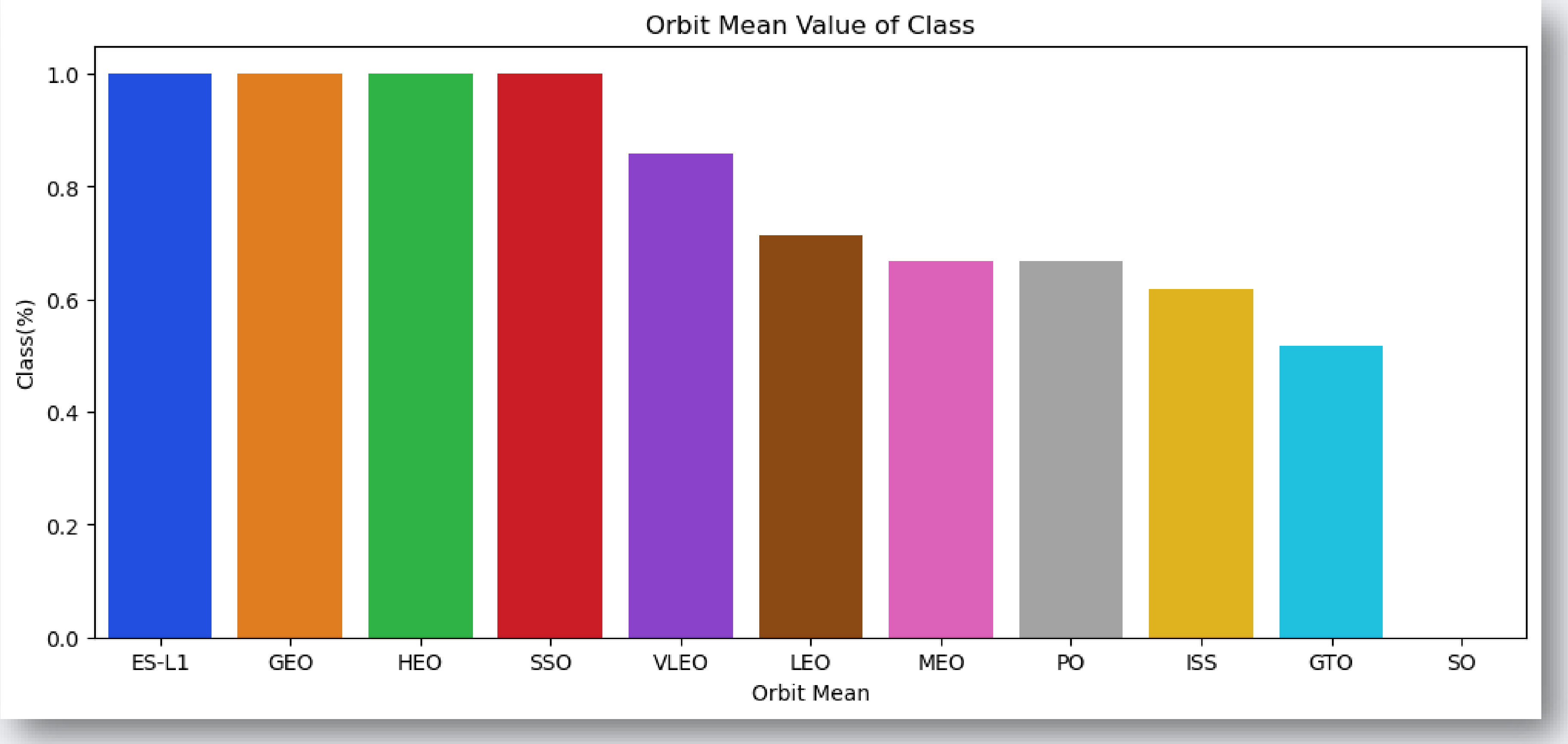
➢ Payload vs. Launch Site

✓ The greater the payload mass for launch site CCAFS 40 the higher the success rate for the rocket.



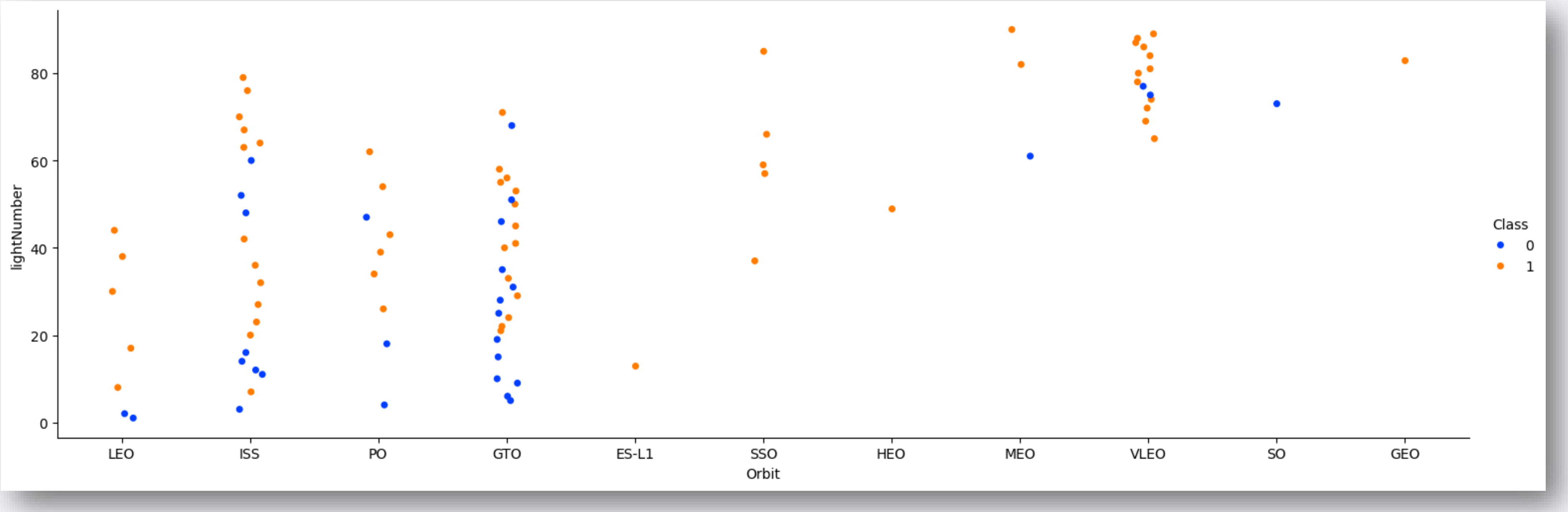
➢ Success Rate vs. Orbit Type

✓ From the plot, we can see that ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.



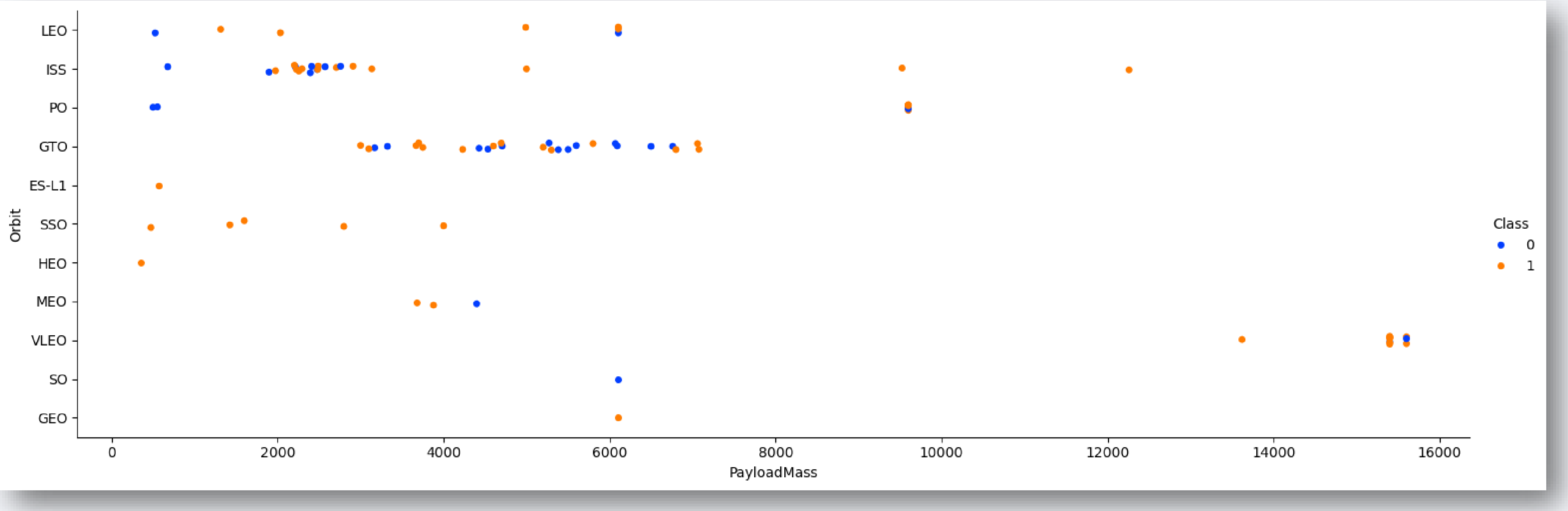
➢ Flight Number vs. Orbit Type

✓ The plot below shows the relationship between Flight Number and Orbit type. We observe that for LEO orbits, success appears correlated with the number of flights, whereas for GTO orbits, there appears to be no such correlation.



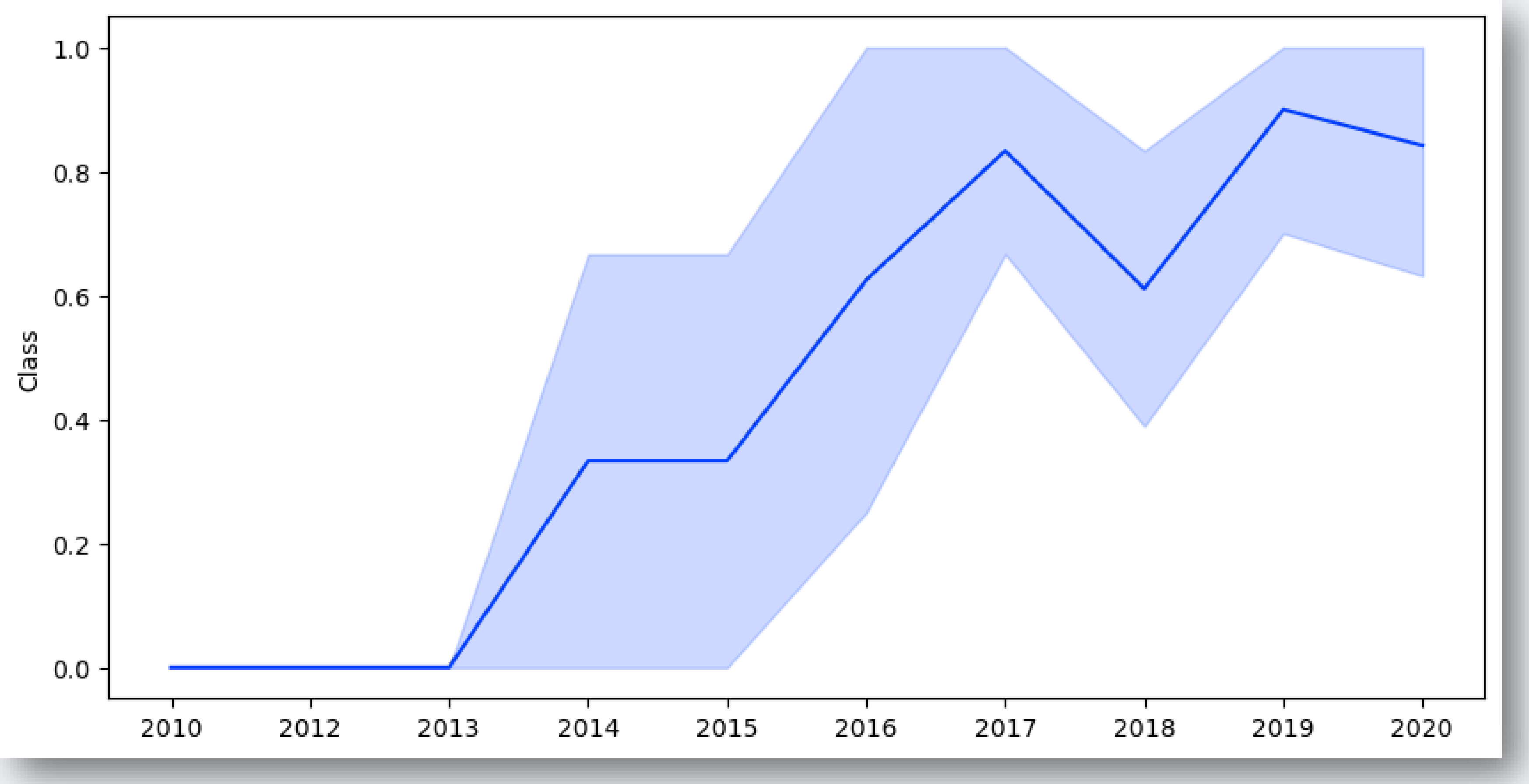
➤ Payload vs. Orbit Type

✓ We can observe that successful landings with heavy payloads are more frequent for PO, LEO, and ISS orbits.



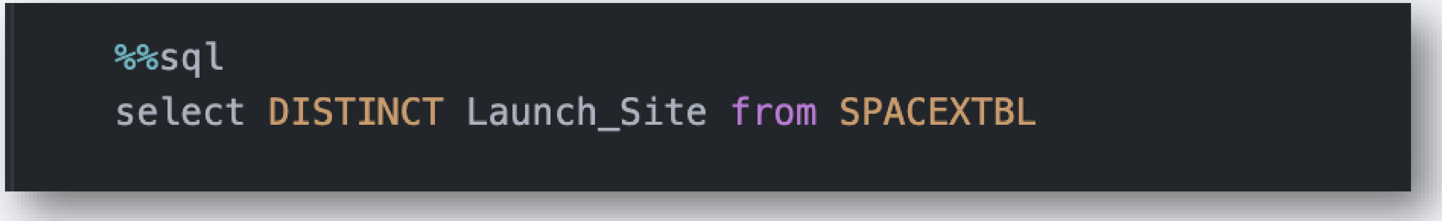
➢ Launch Success Yearly Trend

✓ From the plot, we can observe that the success rate has generally increased from 2013 to 2020.



➢ All Launch Site Names

✓ We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

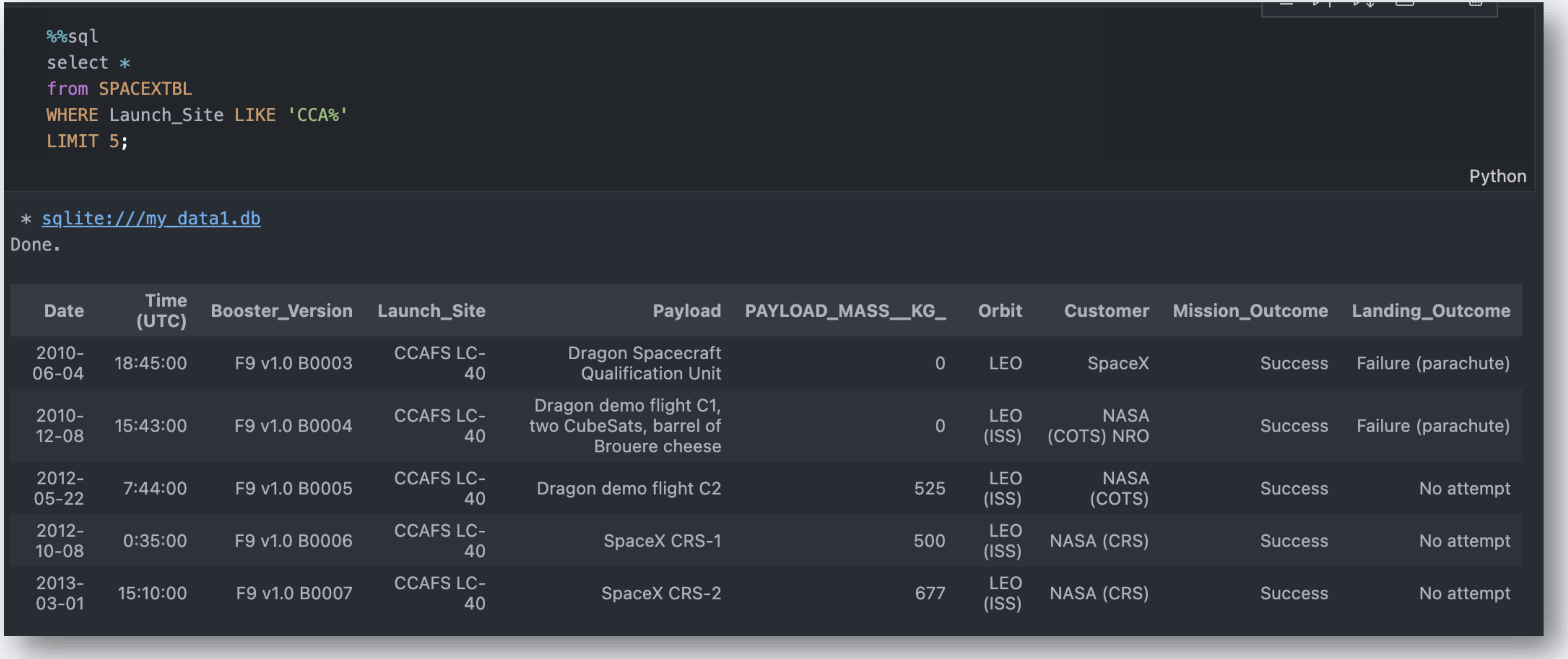


```
%%sql
select DISTINCT Launch_Site from SPACEXTBL
```

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

➢ Launch Site Names Begin with 'CCA'

✓ •We used the query below to display 5 records where launch sites begin with `CCA`



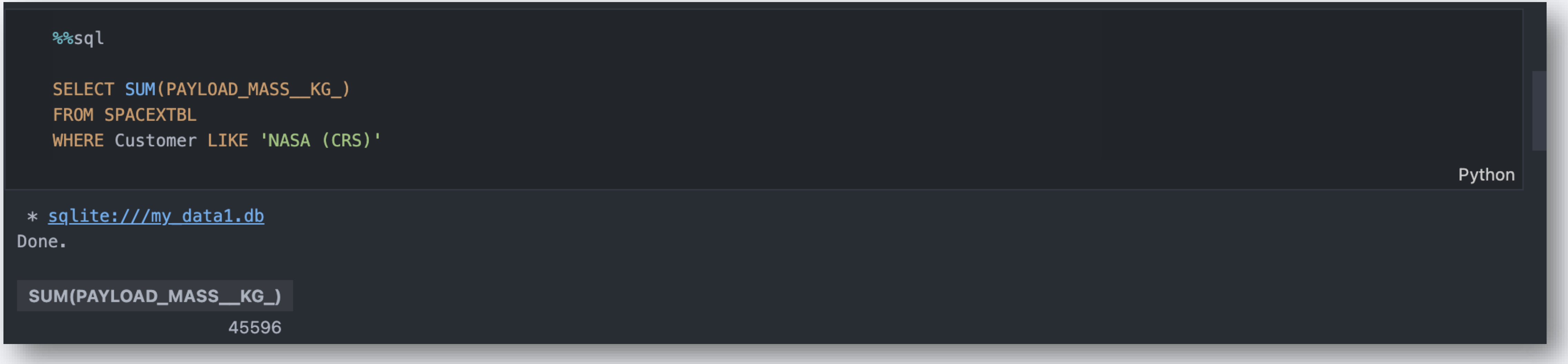
```
%%sql
select *
from SPACEXTBL
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

```
* 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

✓ We calculated the total payload mass carried by NASA boosters to be 45,596 kg using the following query.



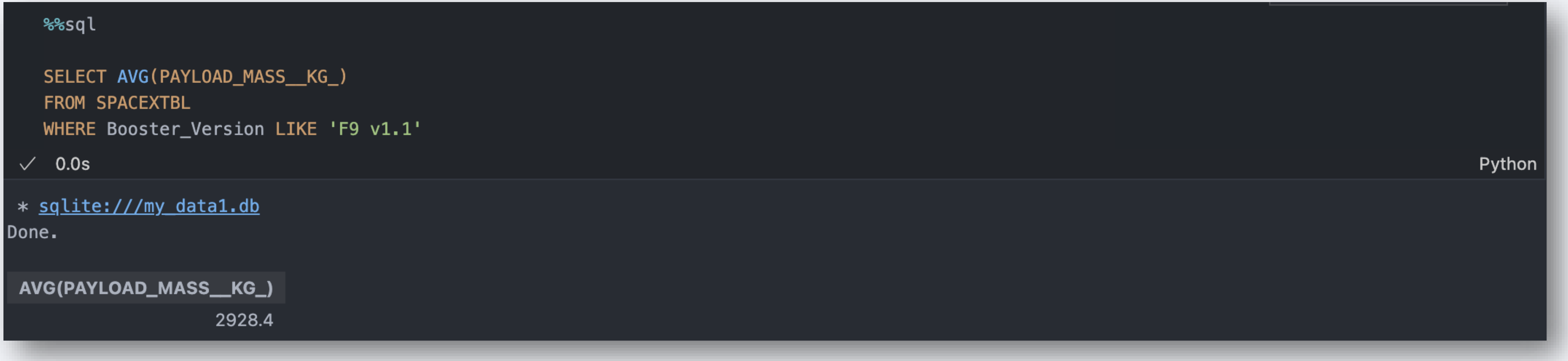
```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer LIKE 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

➢ Average Payload Mass by F9 v1.1

✓ We calculated the average payload mass carried by the F9 v1.1 booster version to be 2928.4 kg.

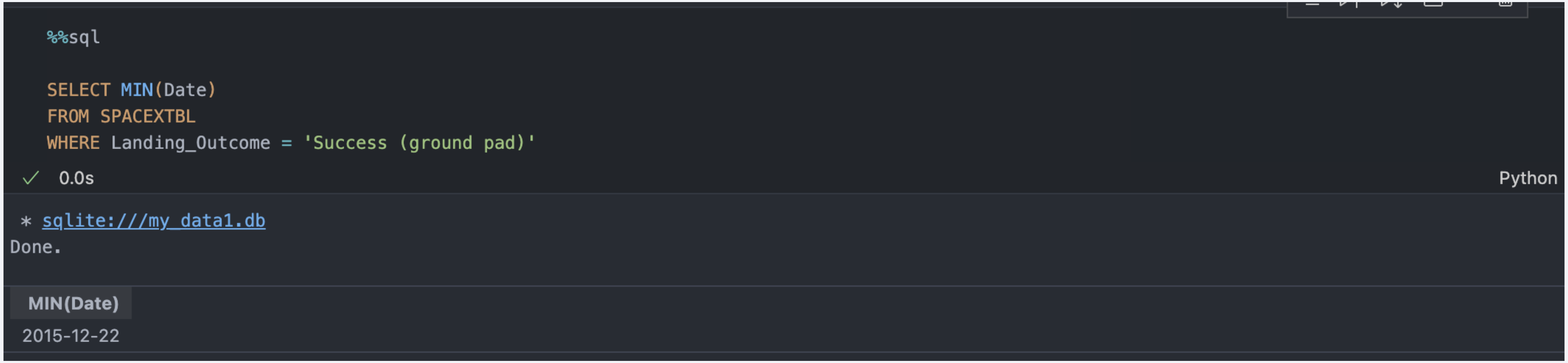


```
%%sql

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

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

AVG(PAYLOAD_MASS__KG_)
                2928.4
```

➢ First Successful Ground Landing Date

- ✓ We observed that the first successful landing on a ground pad occurred on December 22, 2015.



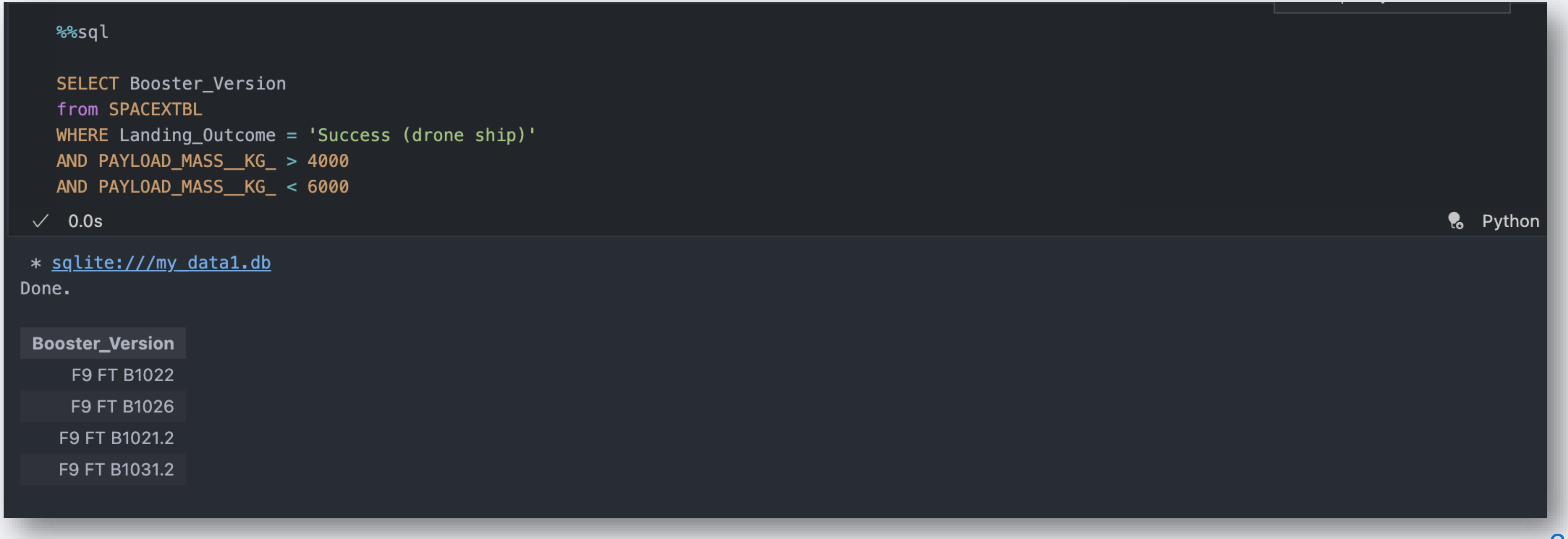
```
%%sql

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

✓ We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000



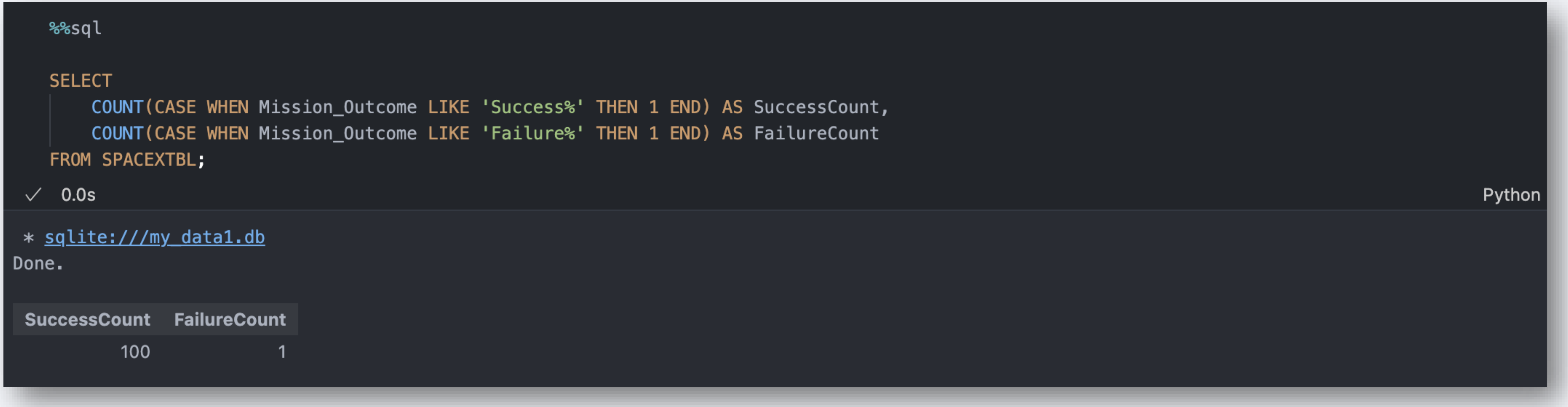
```
%%sql

SELECT Booster_Version
from SPACEXTBL
WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000
```

```
* 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

✓ We used a wildcard, such as '%', to filter the WHERE clause for successful or failed mission outcomes.



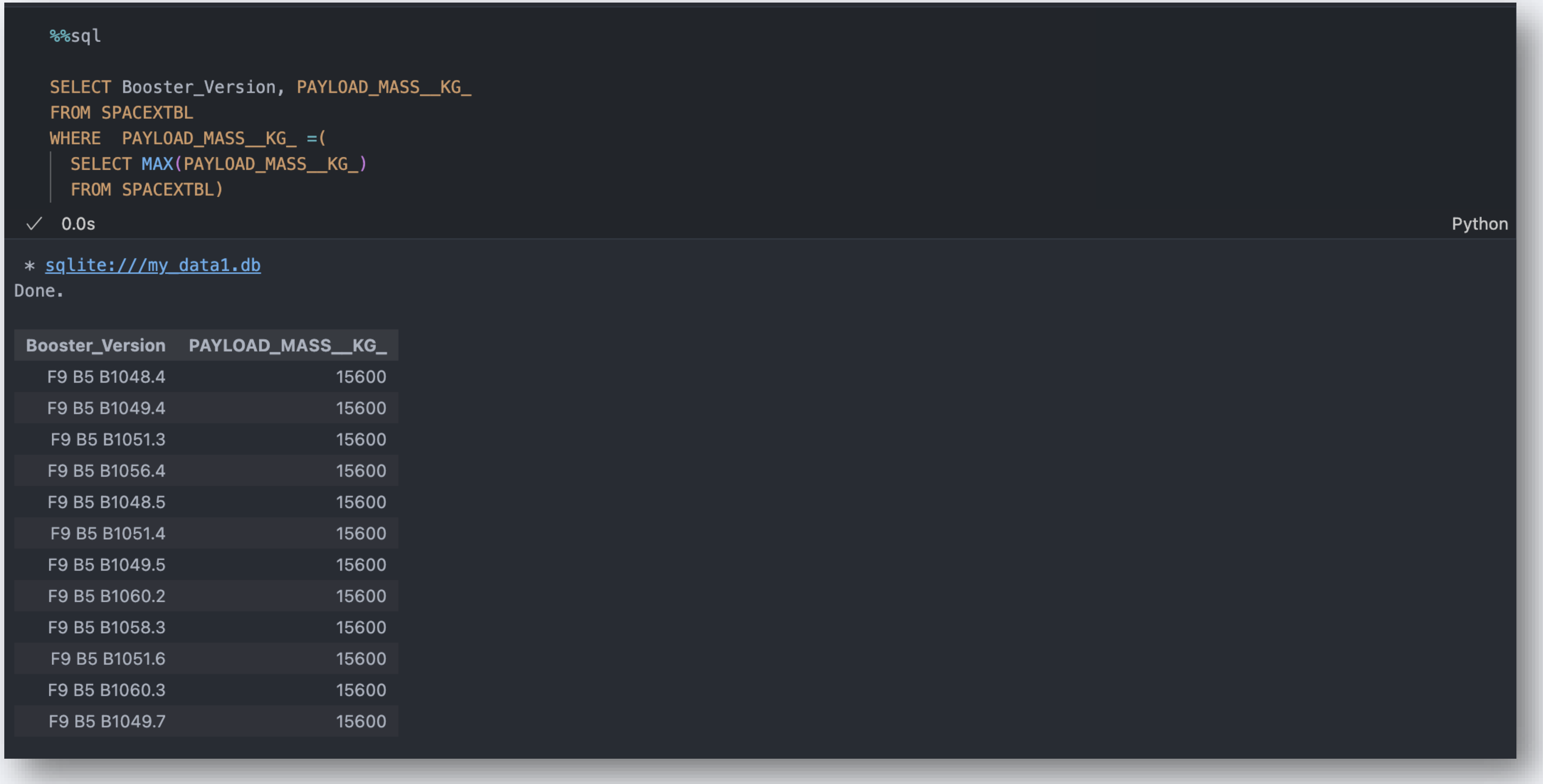
```
%%sql

SELECT
    COUNT(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 END) AS SuccessCount,
    COUNT(CASE WHEN Mission_Outcome LIKE 'Failure%' THEN 1 END) AS FailureCount
FROM SPACEXTBL;
```

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

SuccessCount	FailureCount
100	1

➢ Boosters Carried Maximum Payload

✓ We identified the booster that carried the maximum payload using a subquery in the WHERE clause and the MAX() function.



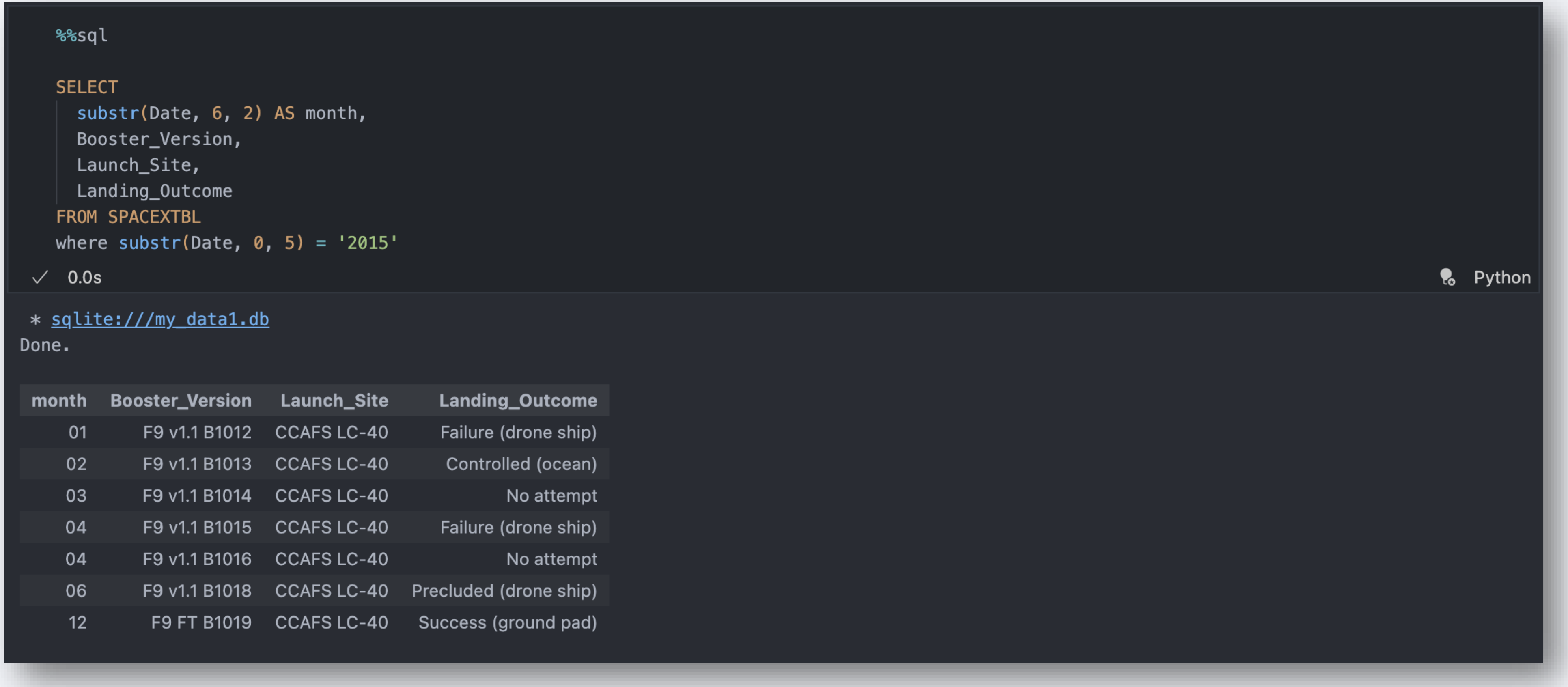
```
%%sql

SELECT Booster_Version, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE  PAYLOAD_MASS__KG_ =(
  SELECT MAX(PAYLOAD_MASS__KG_)
  FROM SPACEXTBL)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

➤ 2015 Launch Records

✓ The month was extracted using substr(Date, 6, 2), and the year was extracted using substr(Date, 0, 4).



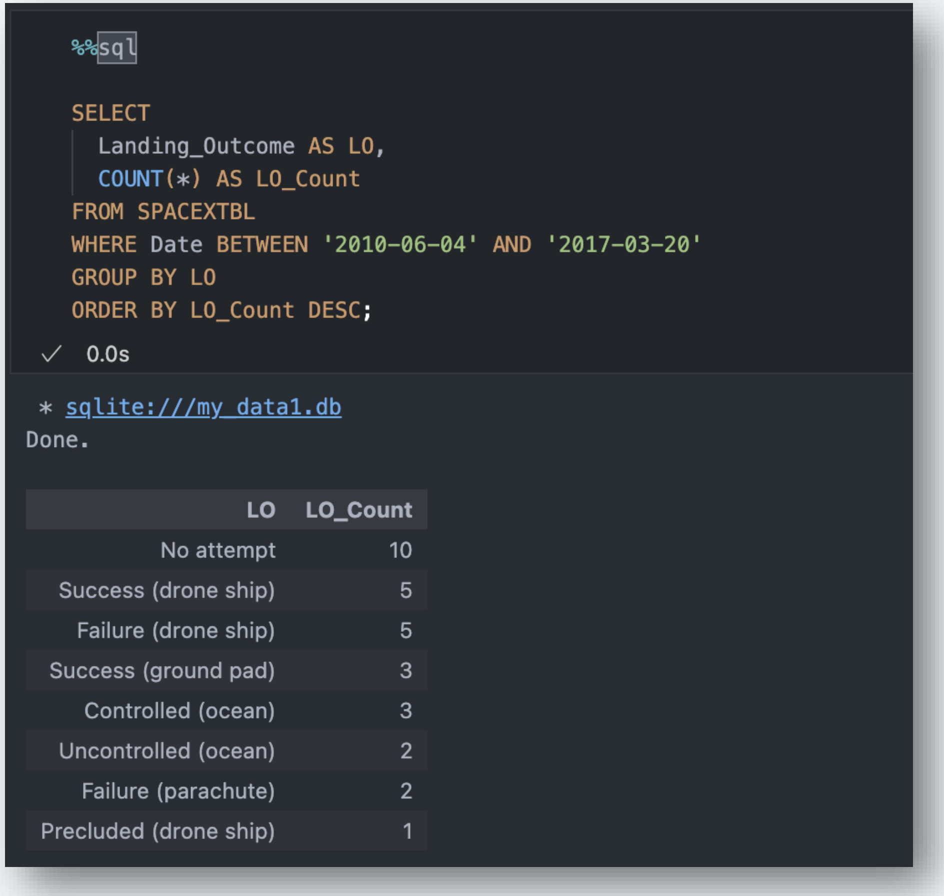
```
%%sql

SELECT
  substr(Date, 6, 2) AS month,
  Booster_Version,
  Launch_Site,
  Landing_Outcome
FROM SPACEXTBL
where substr(Date, 0, 5) = '2015'
```

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

month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
02	F9 v1.1 B1013	CCAFS LC-40	Controlled (ocean)
03	F9 v1.1 B1014	CCAFS LC-40	No attempt
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1016	CCAFS LC-40	No attempt
06	F9 v1.1 B1018	CCAFS LC-40	Precluded (drone ship)
12	F9 FT B1019	CCAFS LC-40	Success (ground pad)

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

- ✓ We selected landing outcomes and the count of each landing outcome from the data, filtering the results using a WHERE clause to include landing outcomes between March 20, 2010, and June 4, 2010.
- ✓ We then grouped the results by landing outcome and ordered them in descending order based on the count.



```
%%sql

SELECT
  Landing_Outcome AS LO,
  COUNT(*) AS LO_Count
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LO
ORDER BY LO_Count DESC;
```

✓ 0.0s

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

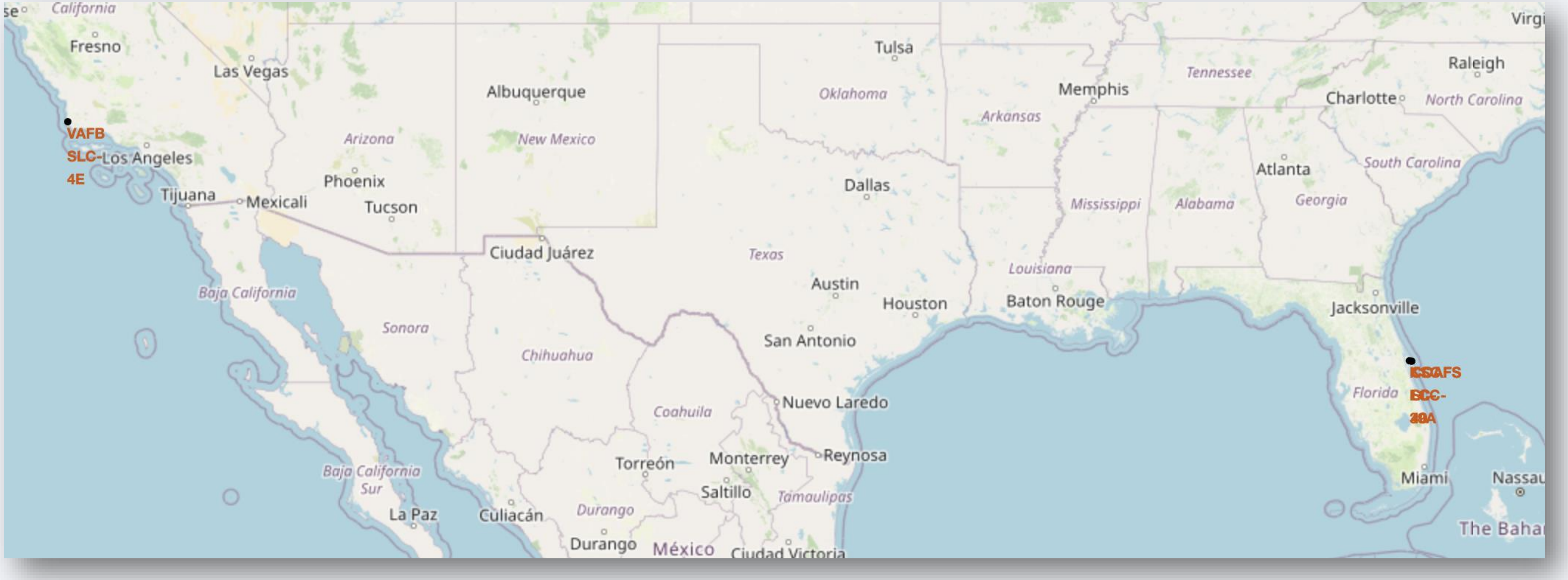
LO	LO_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

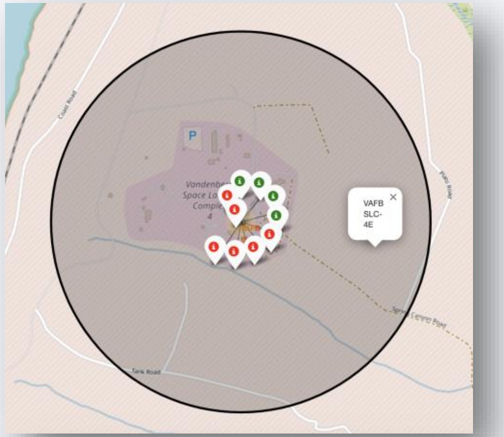
➢ All launch sites global map markers

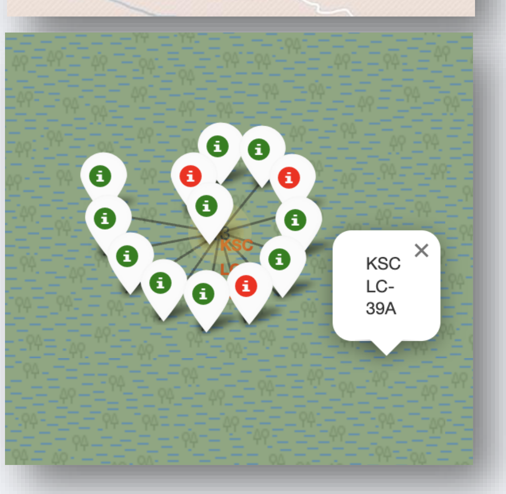
✓ Space launch sites are located on the coasts of the United States, specifically in Florida and California.

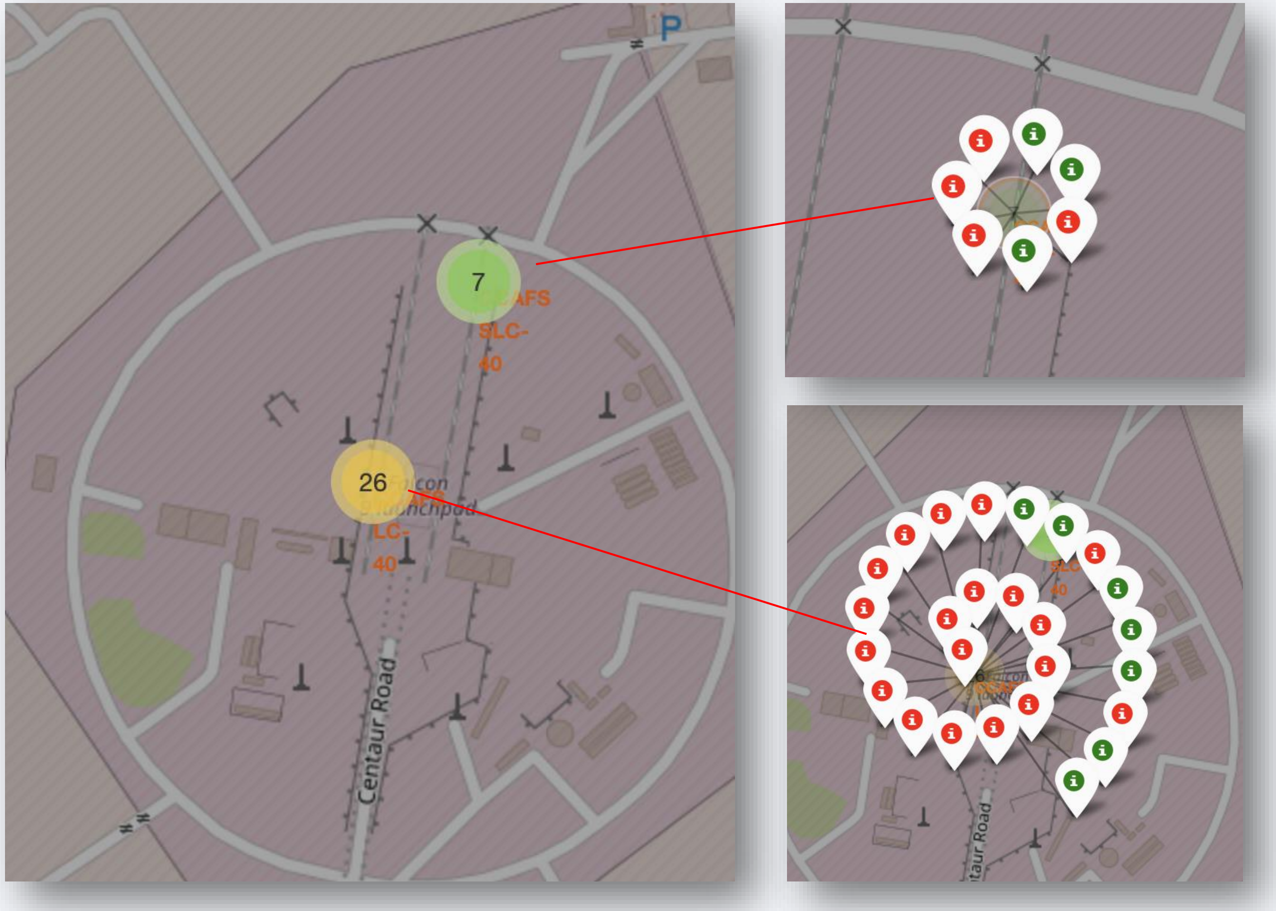


➢ Markers showing launch sites with color labels

- ✓ Green markers indicate success, and red markers indicate failure.
- ✓ KSC LC-39A has the highest success rate.

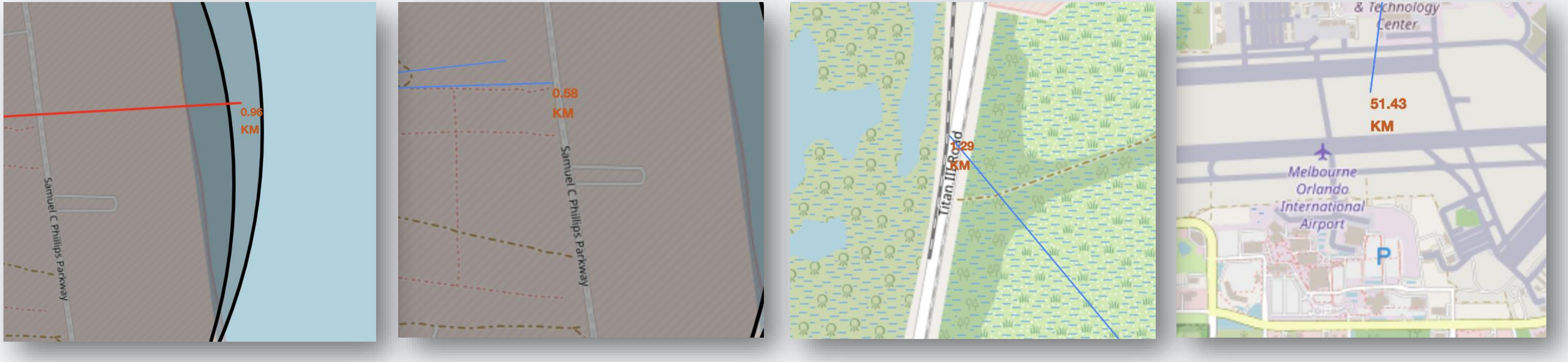




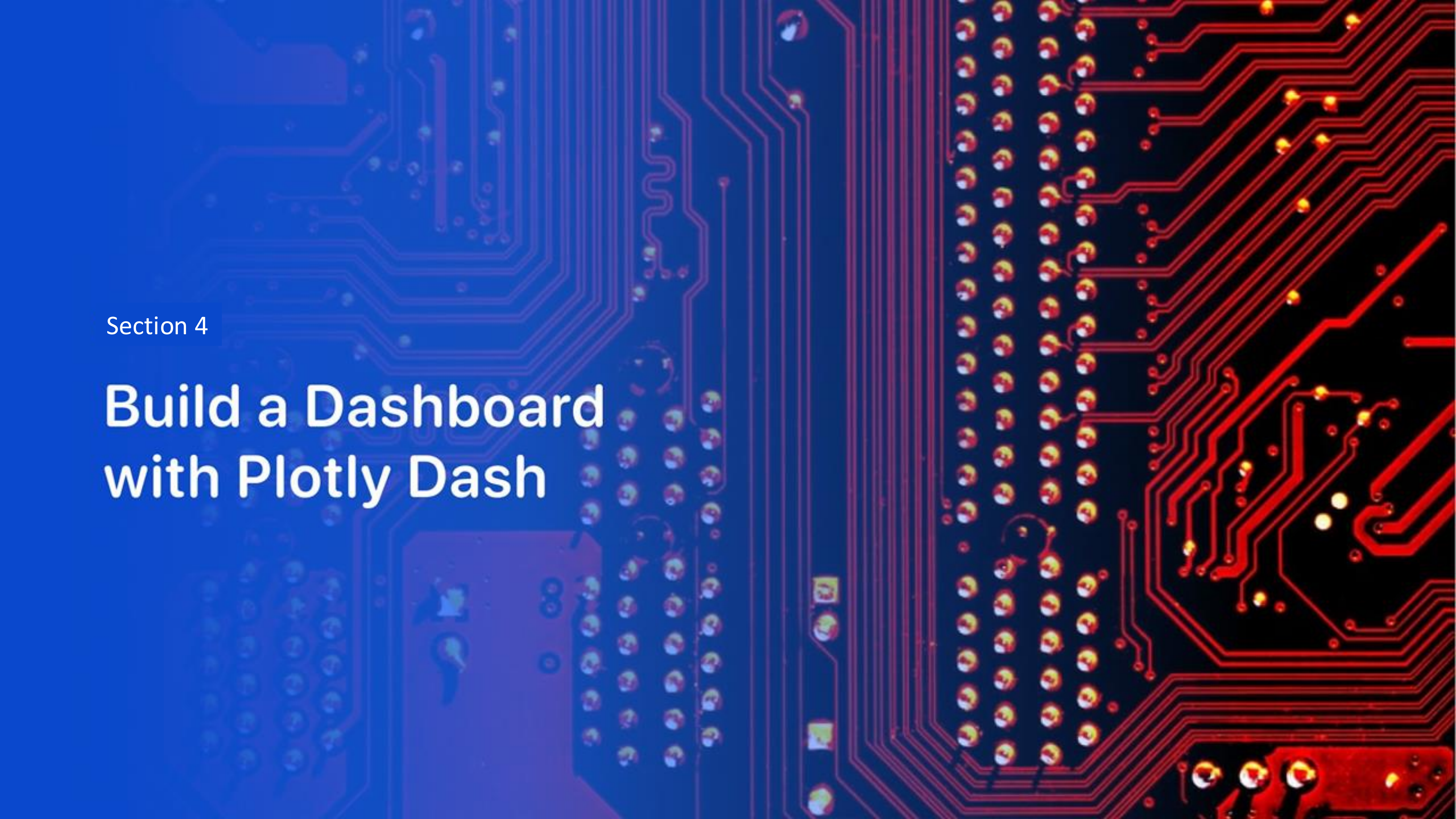


➢ Launch Site distance to landmarks

✓ Calculated distances between major facilities in the surrounding area.



- Are launch sites in close proximity to railways? → No
- Are launch sites in close proximity to highways? → No
- Are launch sites in close proximity to coastline? → No
- Do launch sites keep certain distance away from cities? → Yes

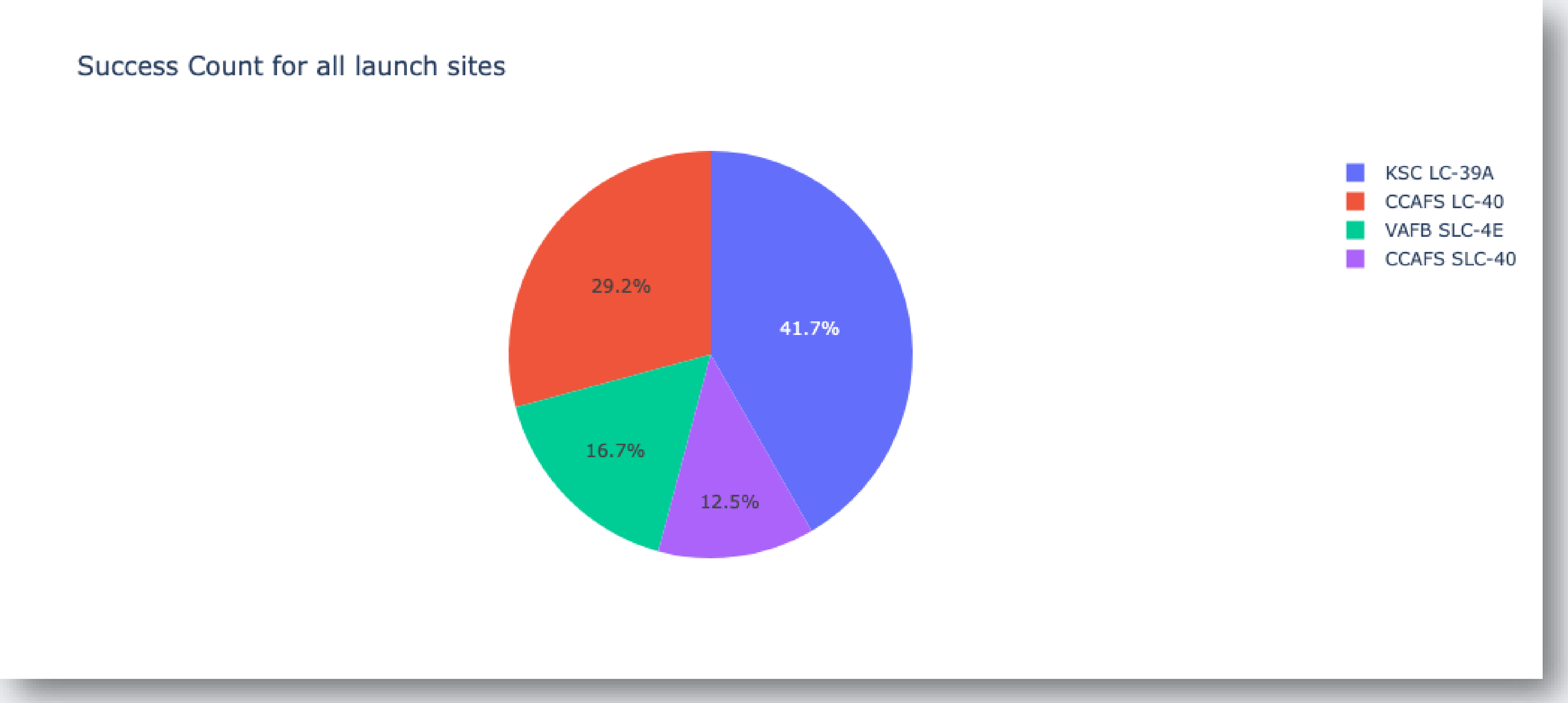


Section 4

Build a Dashboard with Plotly Dash

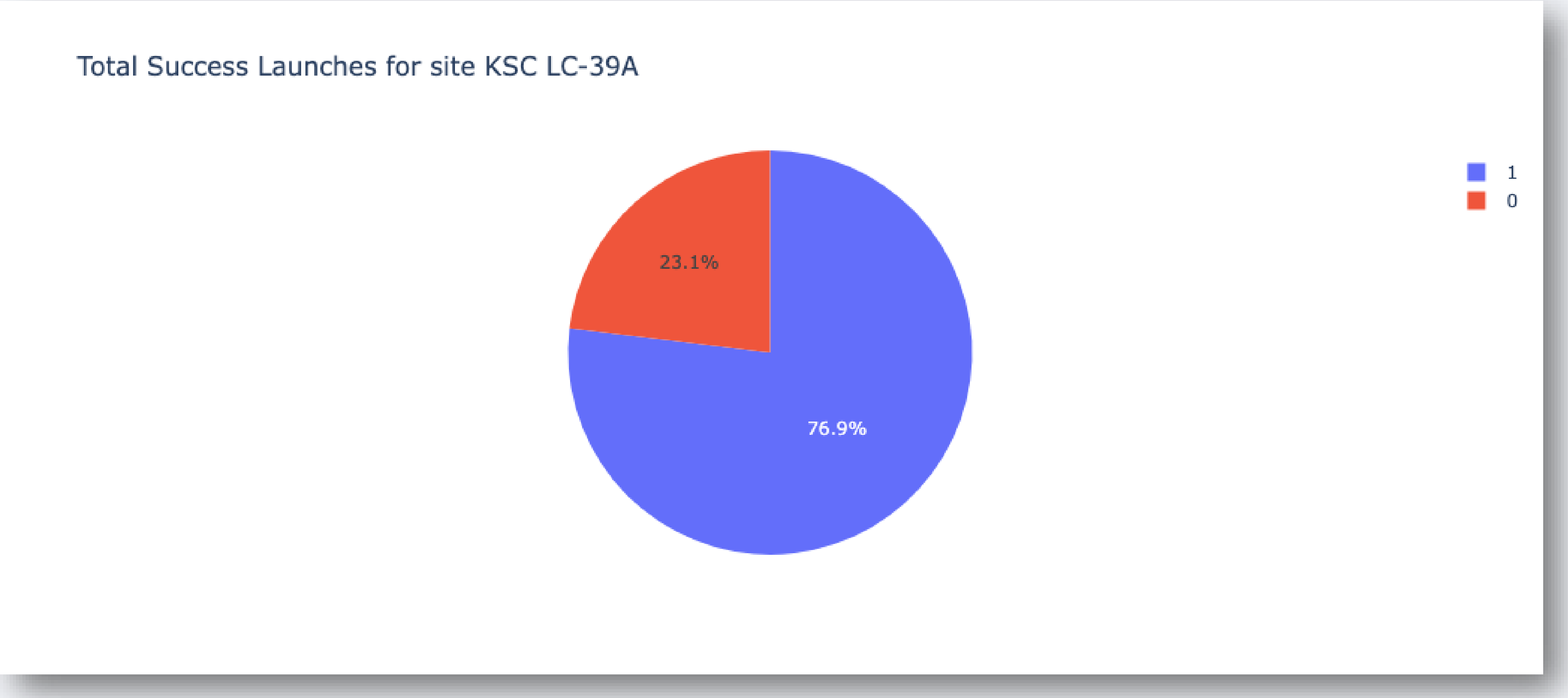
➢ Success percentages by launch site (pie chart)

✓ KSC LC-39A was observed to have the highest number of successful launches among all the sites.



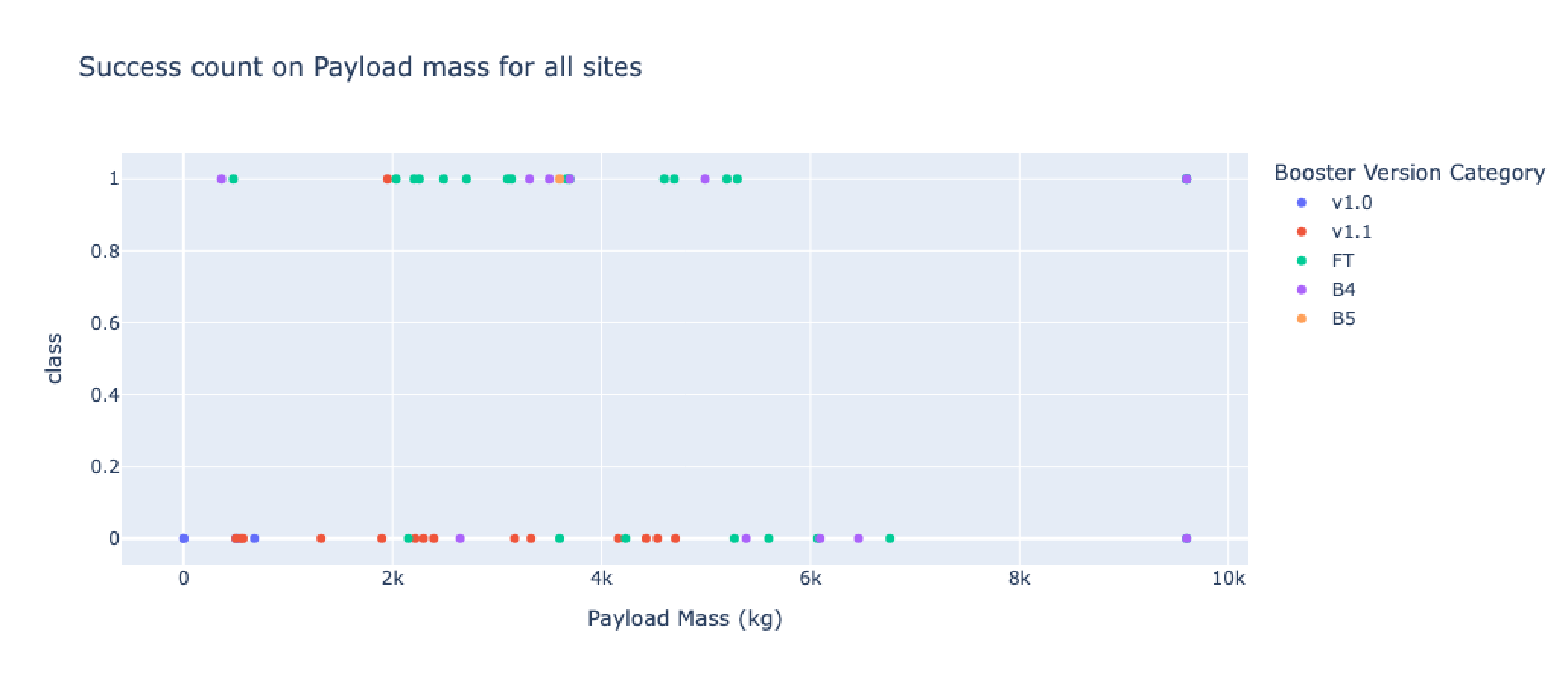
➢ Launch site with the highest launch success ratio

- ➢ KSC LC-39A achieved a 76.9% success rate and a 23.1% failure rate.



➢Payload vs. Launch Outcome scatter plot for all sites (filterable by payload range)

✓ Success rates were observed to be higher for launches with low payload mass than for those with high payload mass.

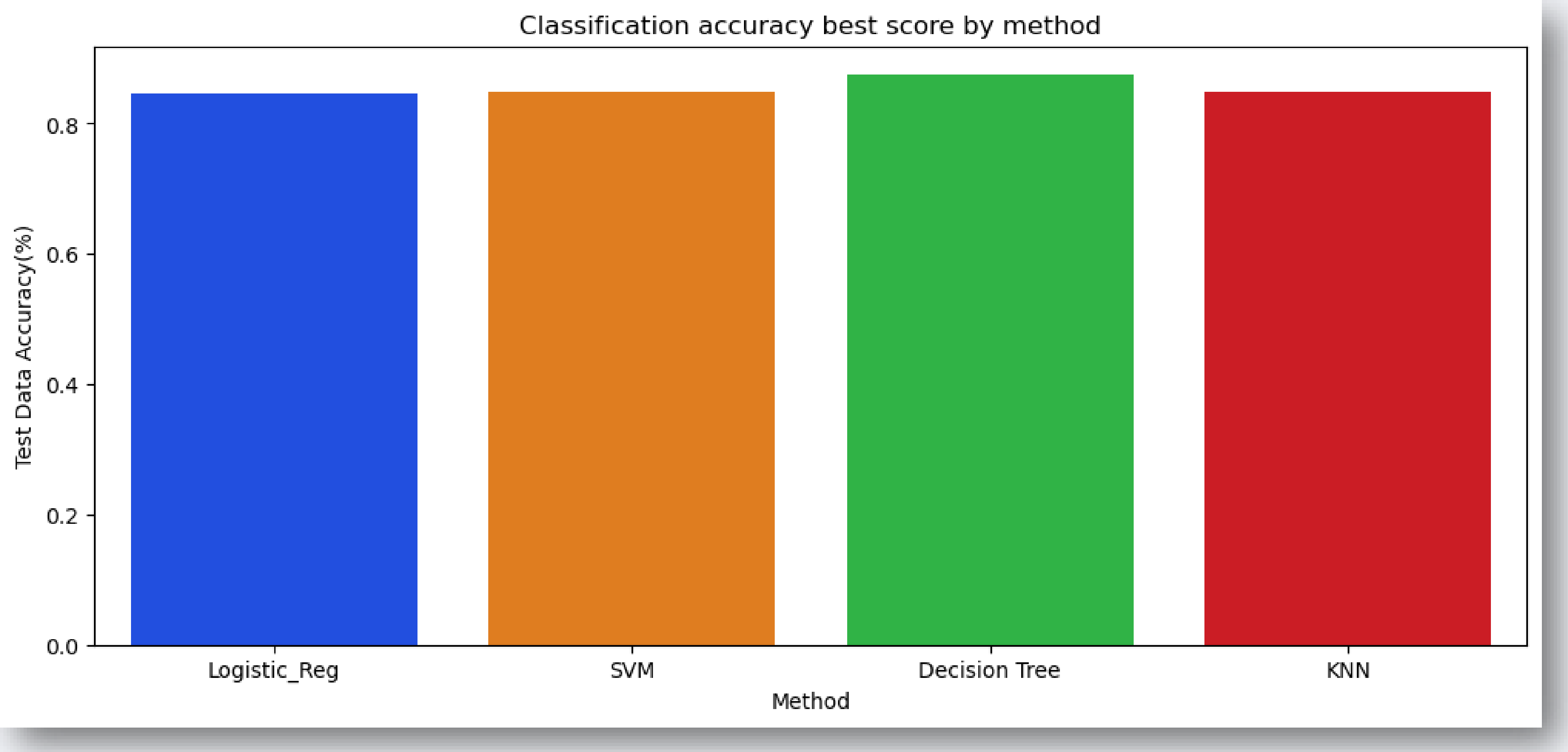


Section 5

Predictive Analysis (Classification)

➢ Classification Accuracy

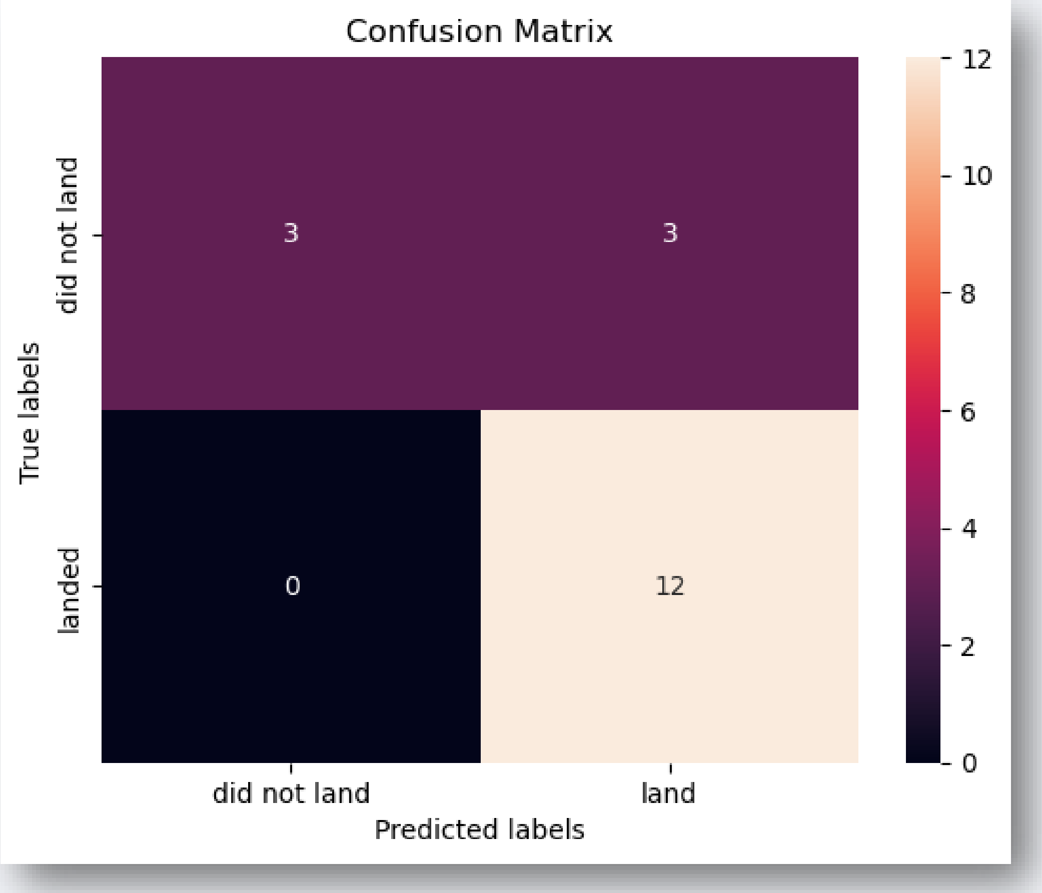
✓ The decision tree classifier achieved the highest classification accuracy.



➢ Confusion Matrix

- ✓ The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

- ✓ However, the major issue is the false positives; that is, unsuccessful landings marked as successful by the classifier.



➢ Conclusions

We conclude the following:

- ✓ A higher number of launches at a site correlates with a higher success rate at that site.
- ✓ The launch success rate generally increased from 2013 to 2020.
- ✓ ES-L1, GEO, HEO, SSO, and VLEO orbits exhibited the highest success rates.
- ✓ KSC LC-39A had the most successful launches of any site.
- ✓ The decision tree classifier proved to be the most effective machine learning algorithm for this task.

Thank you!