IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Stevy Makoumbou
2024-03-07

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with SpaceX API
  - Data collection with Web scrapping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) using SQL
  - EDA using Data Visualization
  - Interactive Visualization with Folium
  - Interactive Visualization with Dashboard
  - Predictive Analysis with Machine Learning
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive Analytic demo in screenshots
  - Predictive Analysis results

Introduction

- Project background and context

  On its website, SpaceX lists Falcon 9 rocket launches at $62 million. Other providers charge upwards of $165 million for a single launch, but SpaceX's reusable first stage significantly reduces costs. Therefore, if an alternate company bids against SpaceX for a rocket launch, it needs to determine the cost of the launch. This project aims to create a machine-learning pipeline to predict the success of the first stage.

- 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 need to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was obtained using SpaceX API and web-scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was done on 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
  - Logistics Regression, SVM, Decision Tree, and KNN models were used
  - Hyper parameters were tuned using GridSearch cross-validation
  - Accuracy on test data was calculated using the score method and the confusion matrix was plotted.

Data Collection

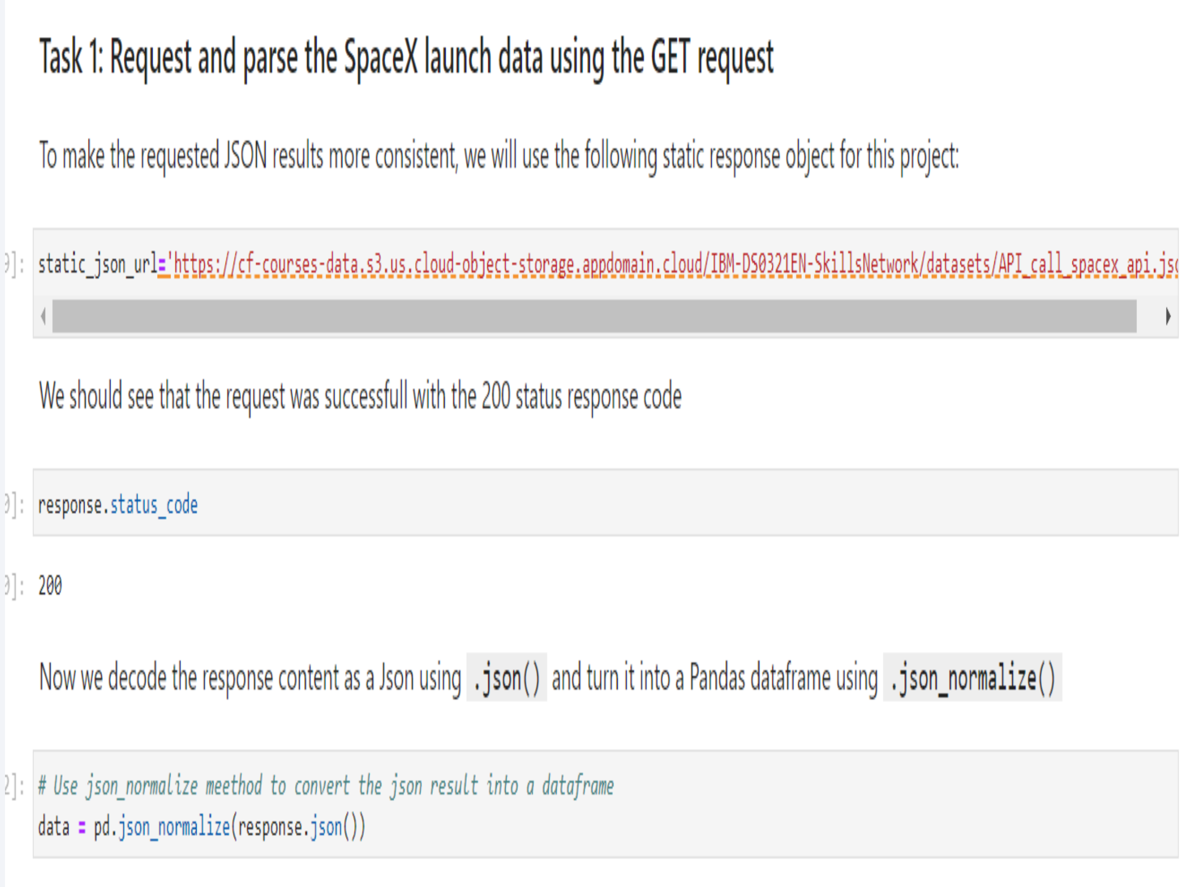
- The following steps were involved in Data Collection:
  - SpaceX API:
    - Data was collected by sending a GET request to SpaceX API
    - The data was decoded as JSON using .json() and turned into a Data Frame using .json normalize()
    - The data was then cleaned, missing values were identified and filled where necessary
  - Web-Scraping:
    - Falcon 9 launch records were also collected by web-scraping from Wikipedia
    - An HTTP GET request was sent to Falcon 9 launch HTML page
    - The data was parsed and stored as a Data Frame using BeautifulSoup.

Data Collection – SpaceX API

- A GET request was sent to the SpaceX API.
- The data was decoded using .json() and converted to Data Frame using .json_normalize()

- GitHub URL :

https://github.com/stevy24/Data-collection-with-SpaceX-API



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

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

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

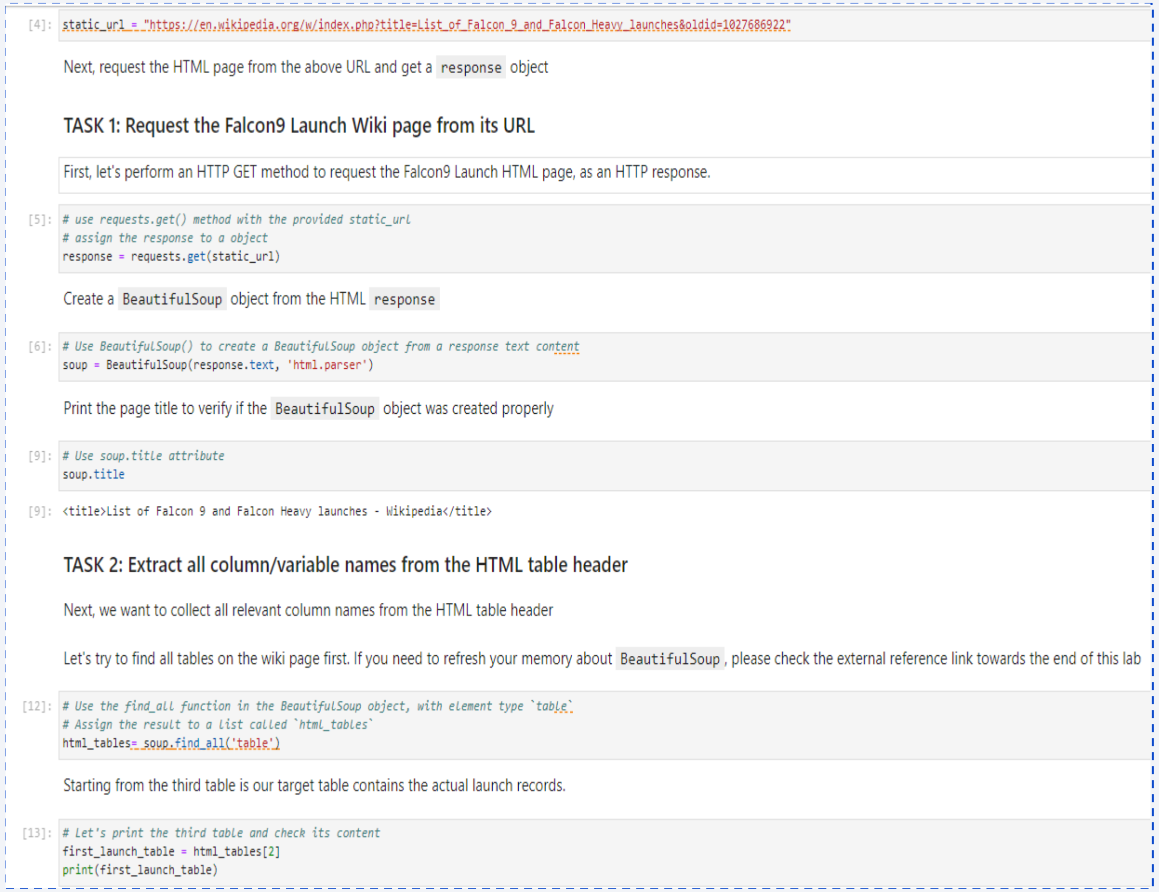
Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Data Collection - Scraping

- HTTP GET request was sent to Falcon 9 Launch HTML page.
- Data was parsed and stored as a Data Frame using BeautifulSoup.
- GitHub URL :

https://github.com/stevy24/Data-collection-with-Web-scrapping



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

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[5]: # use requests.get() method with the provided static_url
     # assign the response to a object
     response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
[6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
     soup = BeautifulSoup(response.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[9]: # Use soup.title attribute
     soup.title

[9]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
[12]: # Use the find_all function in the BeautifulSoup object, with element type `table`
      # Assign the result to a list called `html_tables`
      html_tables= soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
[13]: # Let's print the third table and check its content
      first_launch_table = html_tables[2]
      print(first_launch_table)
```

Data Wrangling

- Calculated the number of launches on each site, the number and occurrence of each orbit, and the types and respective number of landing outcomes.

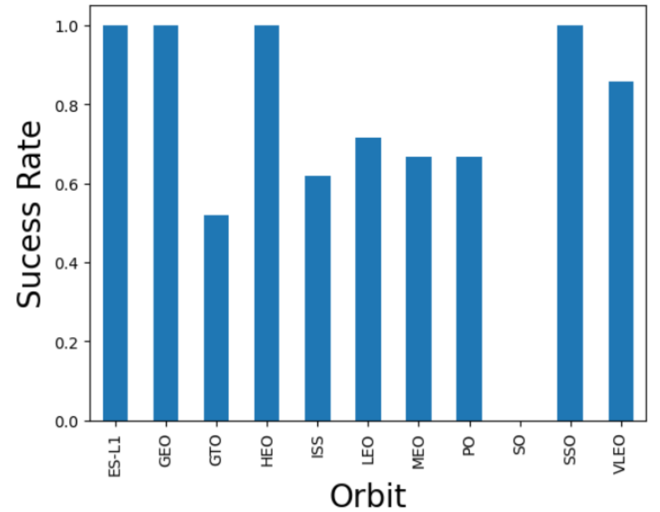
- Created a landing outcome label from the Outcome column

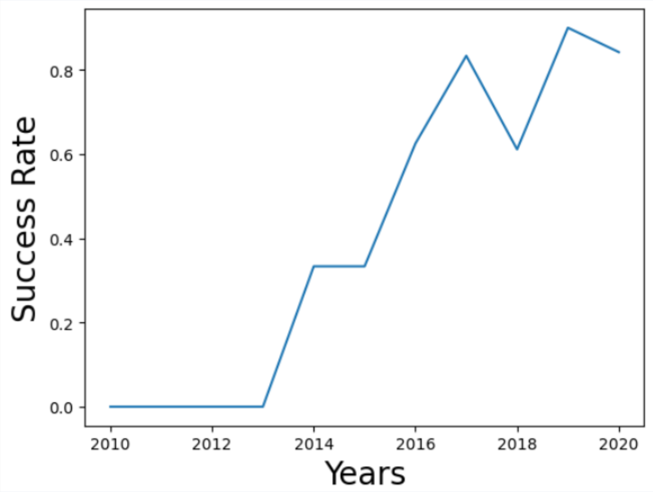
- GitHub URL :

https://github.com/stevy24/Data-Wrangling

EDA with Data Visualization

- We visualized the relationship between flight number and launch site, payload and launch site, success rate and orbit, number of flights and orbit, payload mass and orbit, and launch success yearly trend.





- GitHub URL :

https://github.com/stevy24/EDA-with-Data-Visualization

EDA with SQL

- SQL table was loaded in Jupyter, and the following SQL queries were performed:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failed mission outcomes
  - The failed landing outcomes in drone ship, their booster version, and launch site names.

- GitHub URL :

https://github.com/stevy24/EDA-with-SQL

Build an Interactive Map with Folium

- We marked all launch sites and added map objects such as markers, circles, and lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to classes 0 and 1. i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rates.
- We calculated the distances between the launch site and its proximities like railways, highways, and cities.
- GitHub URL :

https://github.com/stevy24/Interactive-Visualization-with-Folium

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash
- We plotted pie charts showing the total launches by certain sites
- We plotted a scatter graph showing the relationship between Outcome and Payload Mass (Kg) for the different booster versions.
- GitHub URL :

https://github.com/stevy24/Interactive-Visualization-with-Dashboard

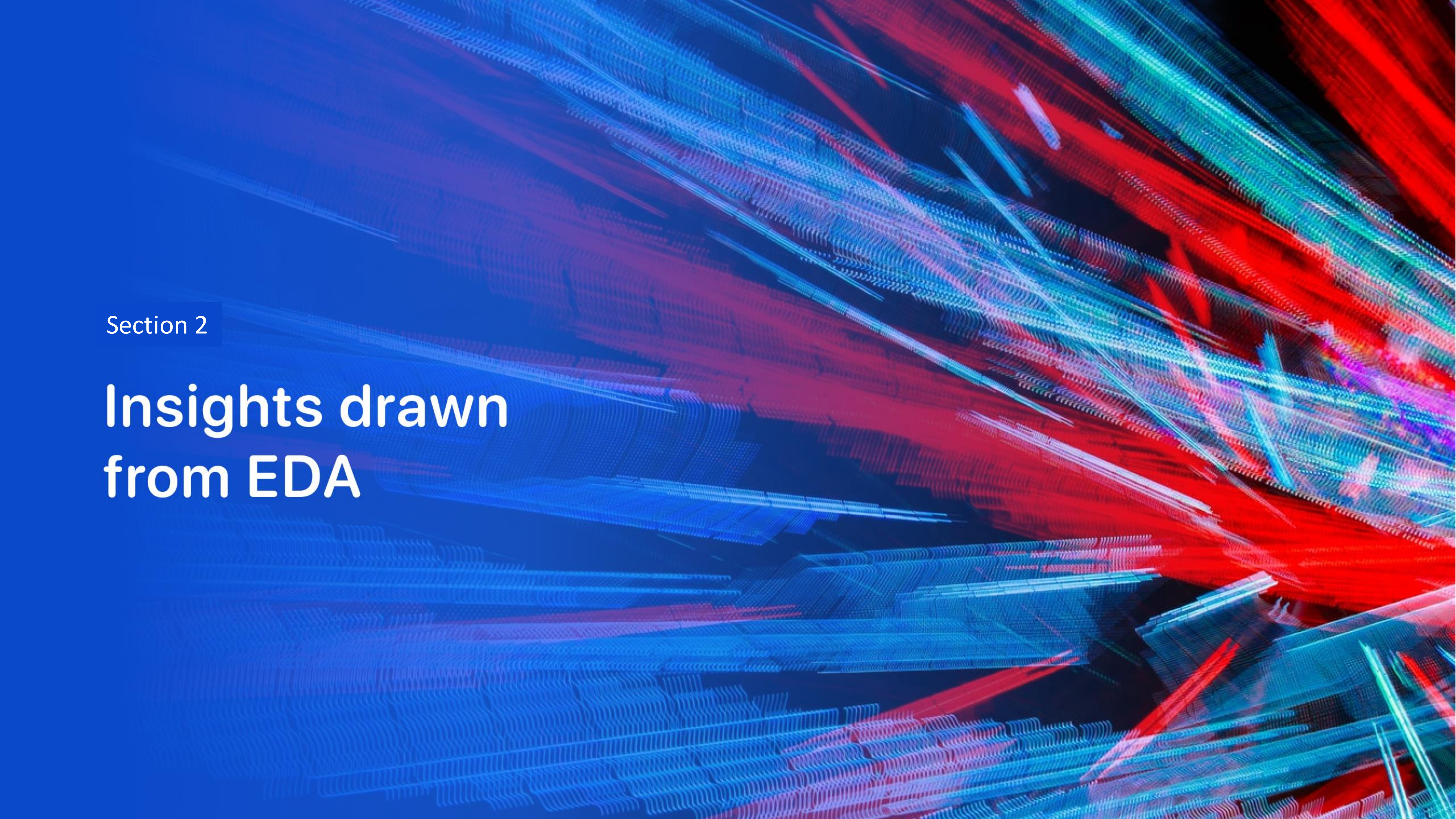
Predictive Analysis (Classification)

- We split the dataset into training and testing sets.
- We built different machine-learning models and tuned different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model and improved the model using feature engineering and hyperparameter tuning.
- We found the best-performing classification model.
- GitHub URL :

https://github.com/stevy24/Predictive-Analysis-with-Machine-Learning

Results

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

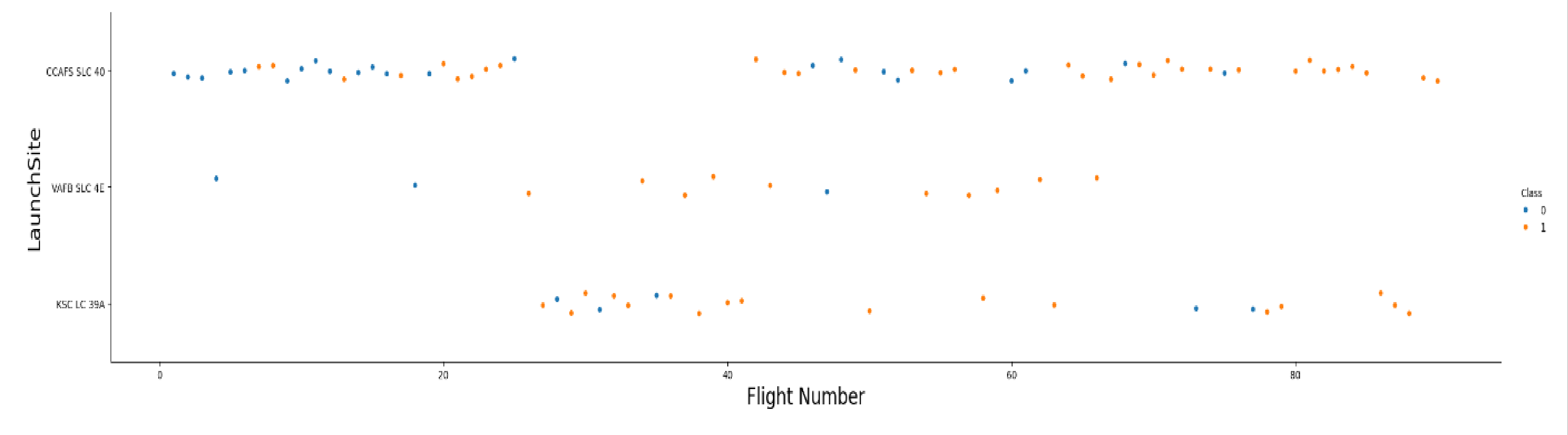


Section 2

Insights drawn from EDA

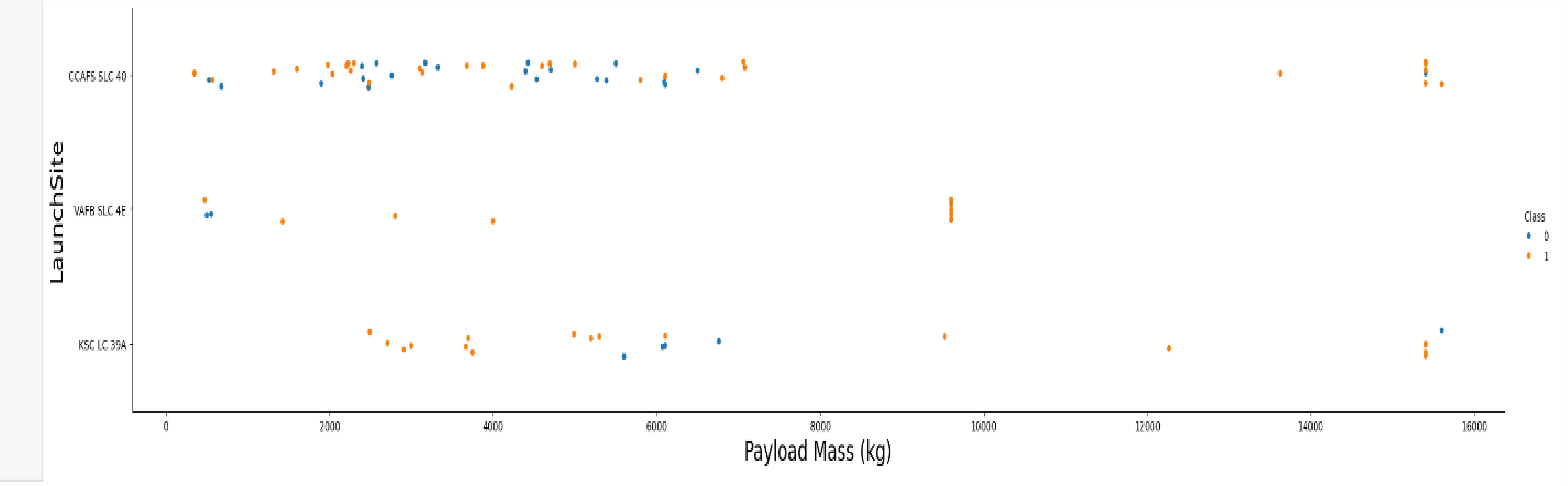
Flight Number vs. Launch Site

- From the scatter plot of flight Number vs. Launch Site we see that the greater the number of flights at a launch site, the greater the success rate at the launch site.



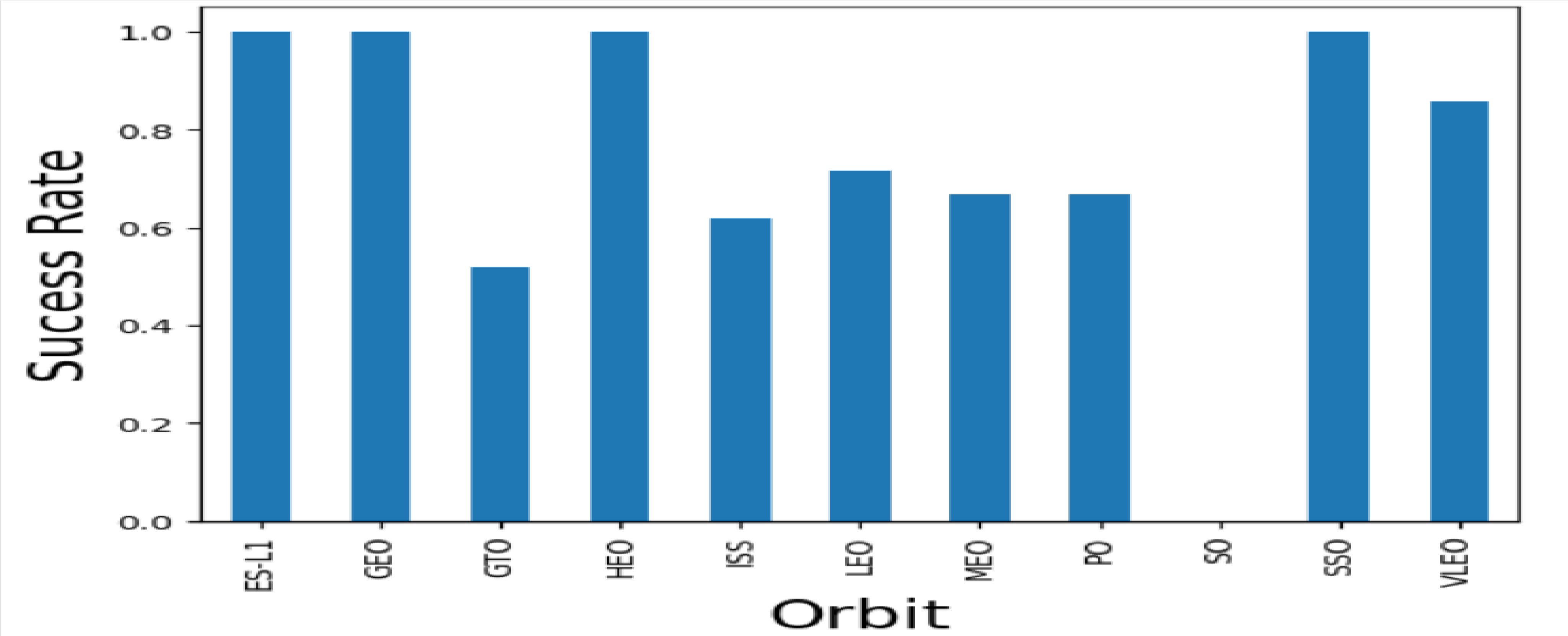
Payload vs. Launch Site

- For CCSFS SLC 40, as payload mass increases the first stage is more likely to land.
- For the VAFB-SLC launch site there are no rockets launched for payload > 10,000 kg
- For the KDC LC 39A launch site there are no rockets launched for payload < 2,000 kg



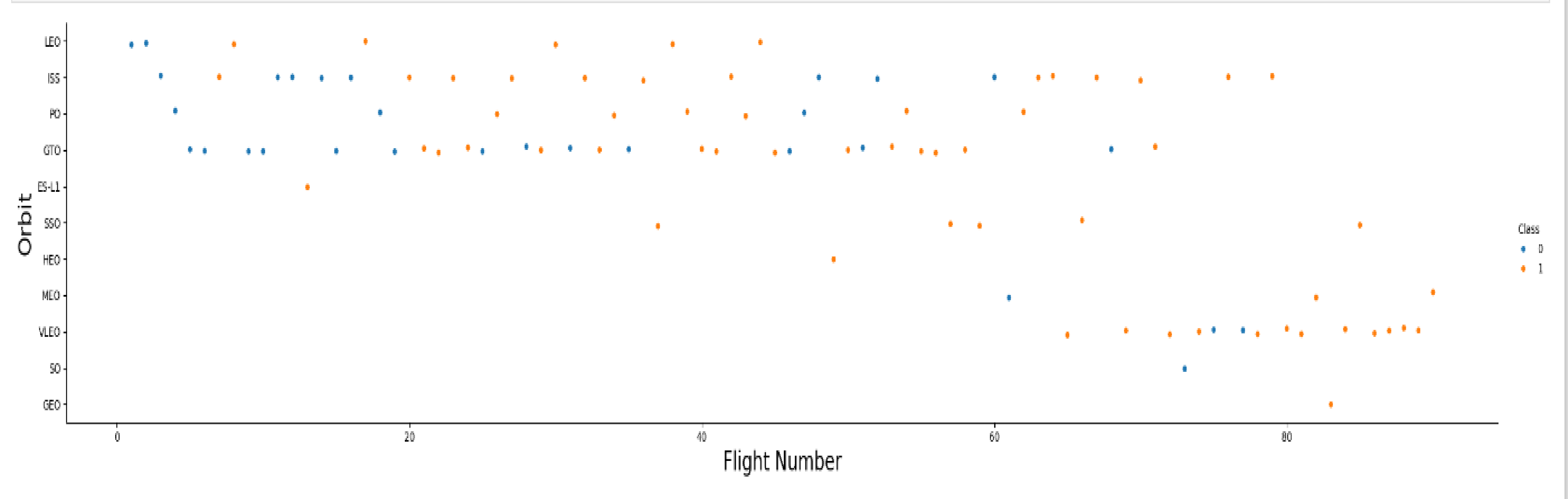
Success Rate vs. Orbit Type

- From the bar chart, we see that ES-L1, GEO, HEO, and SSO orbits have the highest success rate.



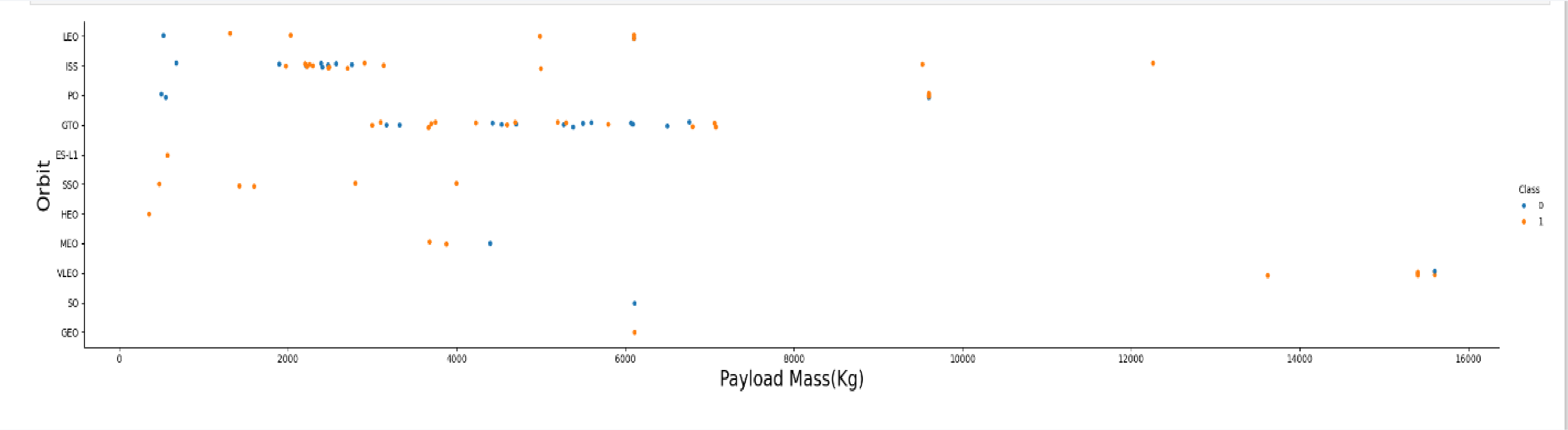
Flight Number vs. Orbit Type

- From the scatter plot of Flight number vs. Orbit type, we see that LEO's success rate is related to flight number, whereas GEO's success rate seems to be independent of flight number.



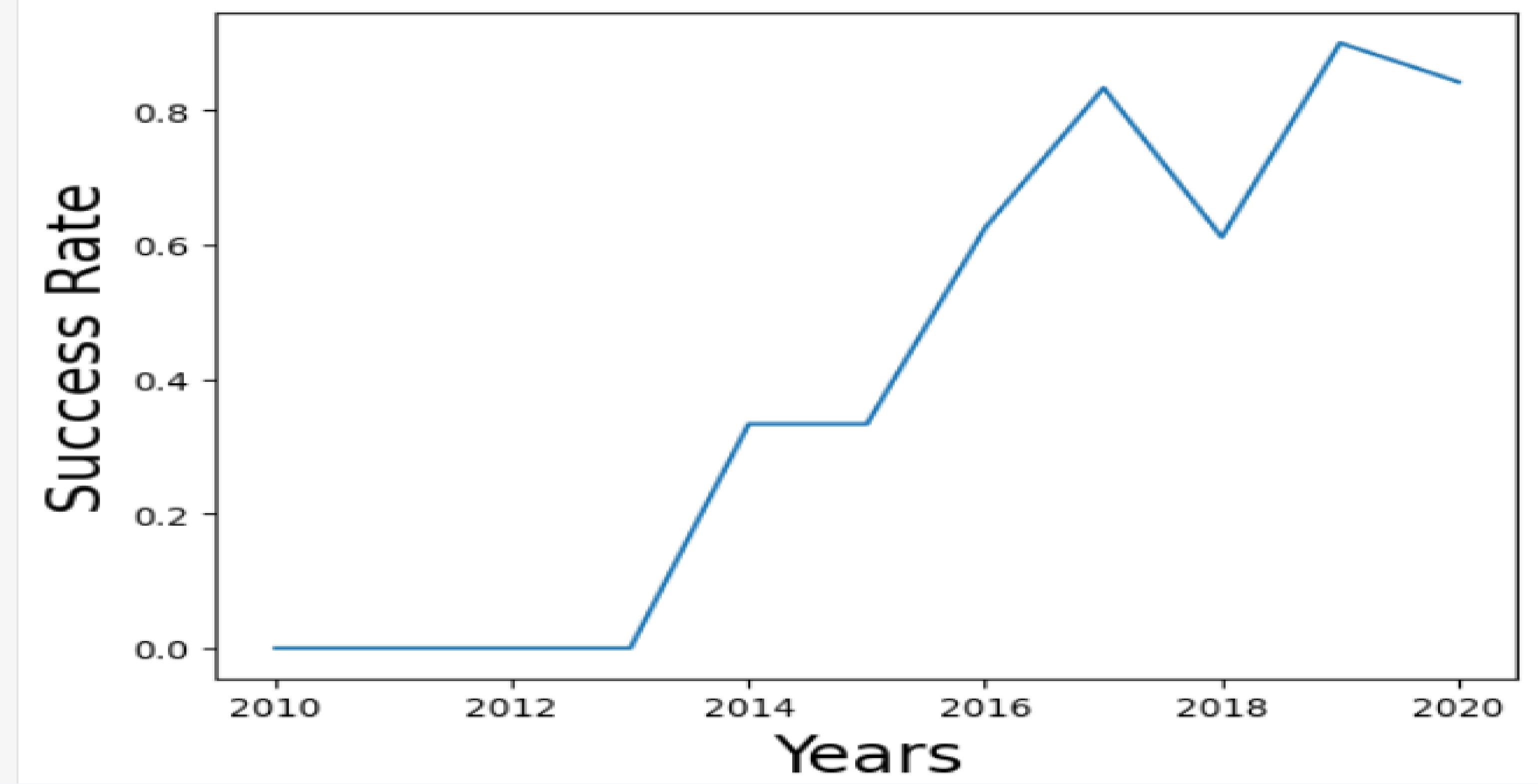
Payload vs. Orbit Type

- We see that for heavier payloads, the success rate increases for LEO, ISS, and Polar orbits
- There seems to be no observable relation between GTO orbit and payload mass.



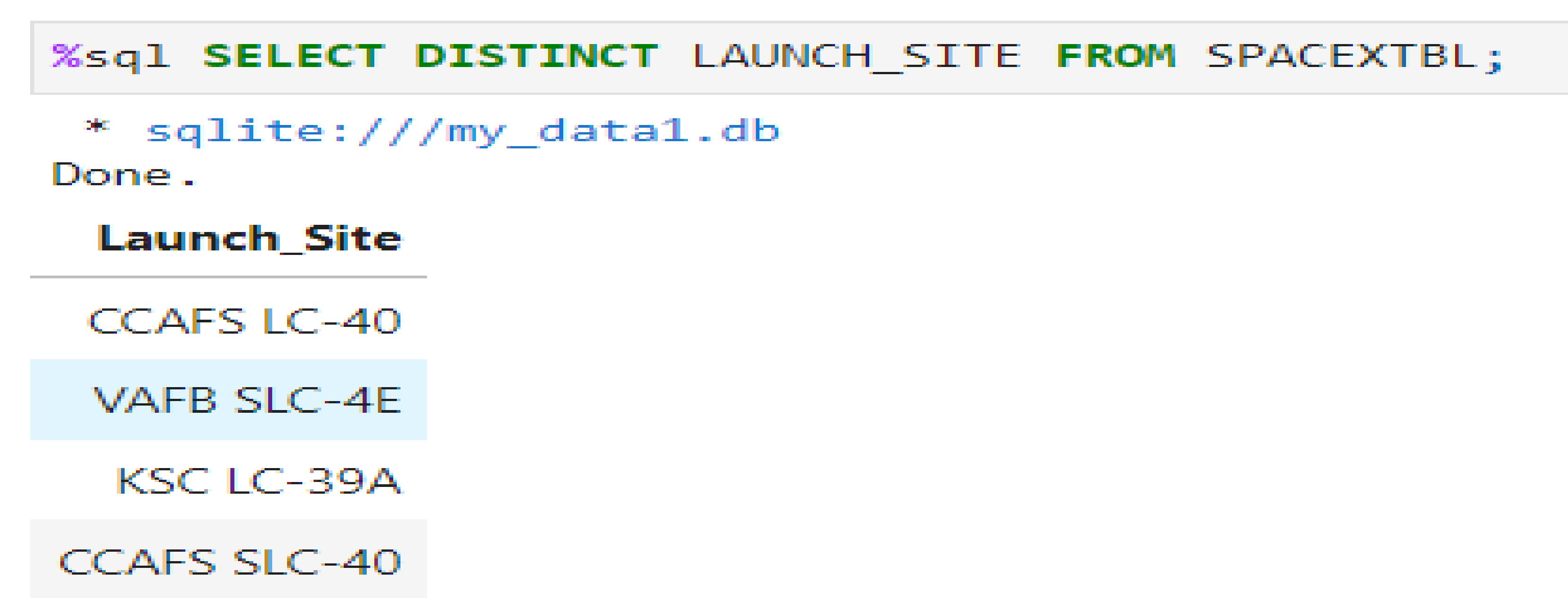
Launch Success Yearly Trend

- The Line Chart shows that the success rate increased from 2013 to 2020 with minor fluctuations throughout the period.
- We see a major plunge in the success rate for the year 2018.



All Launch Site Names

- We use the keyword DISTINCT to select unique launch sites from the table.



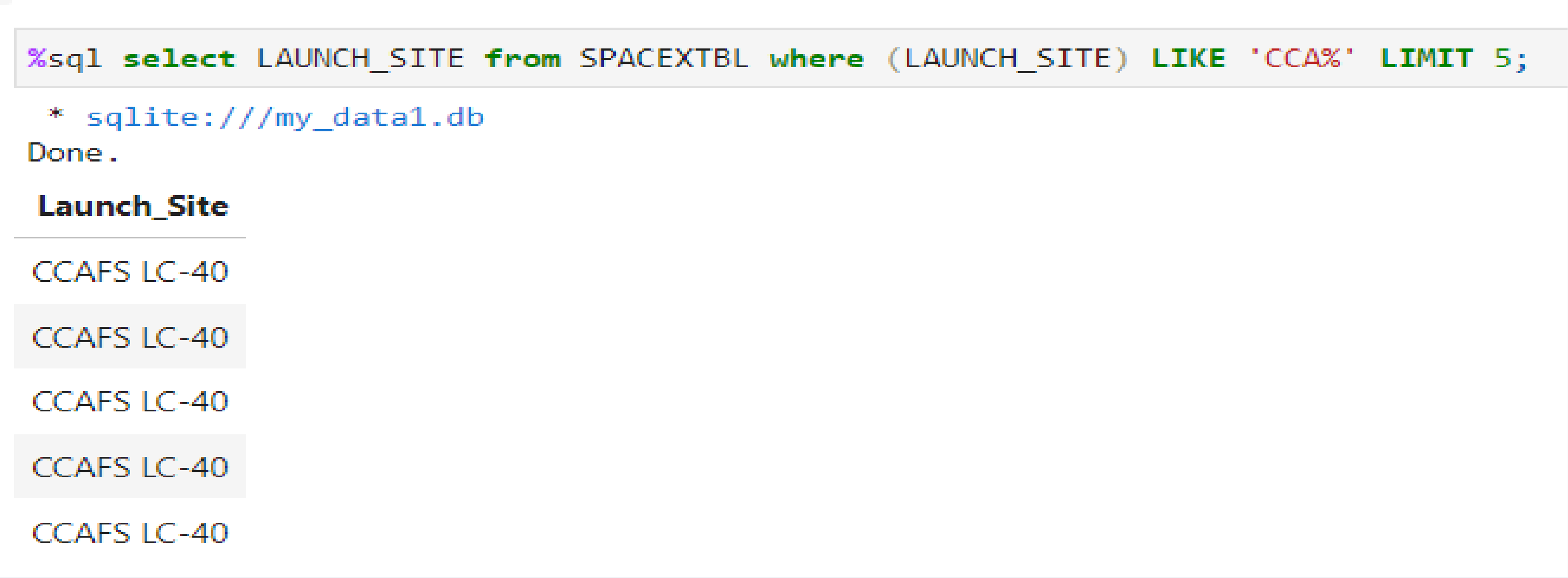
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

- The keyword LIKE is used to specify that the launch site name begins with CCA.
- The limit keyword is used to query only 5 records.



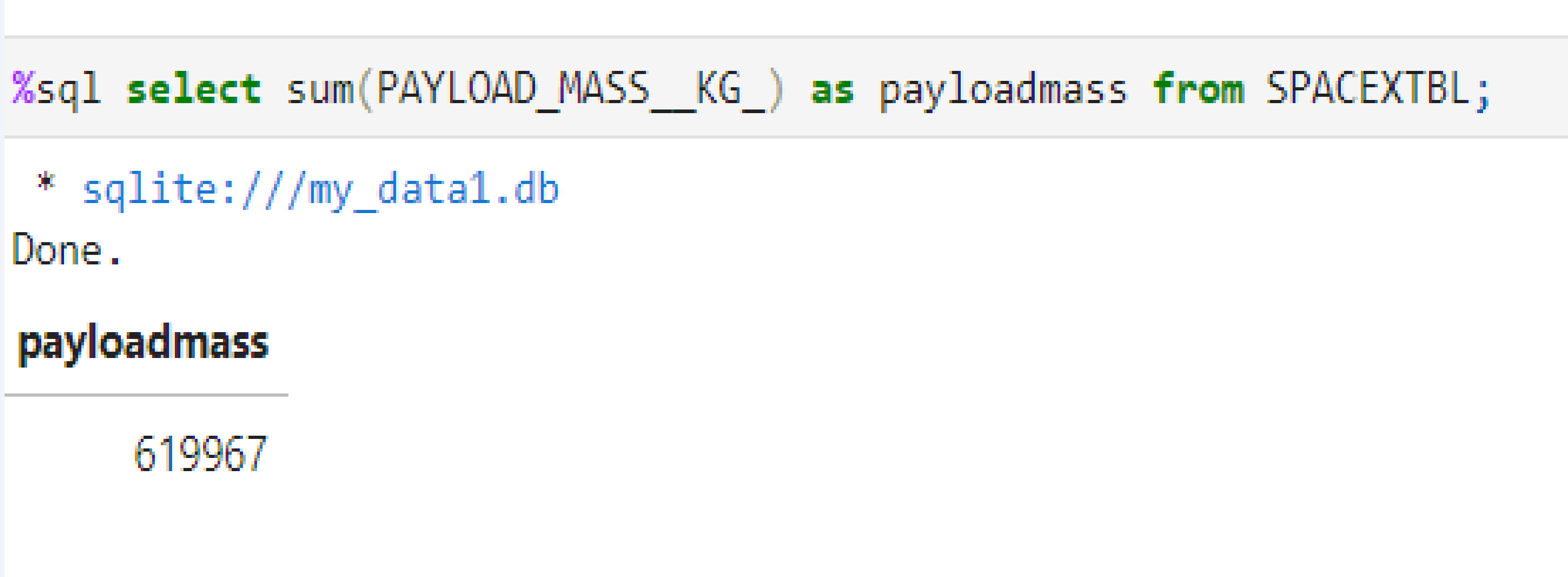
```
%sql select LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- The total payload mass for customer NASA (CSR) is 45,596 kg.
- The function SUM is used to obtain the sum of payload mass.
- The WHERE clause is used to specify that the customer should be NASA (CRS)



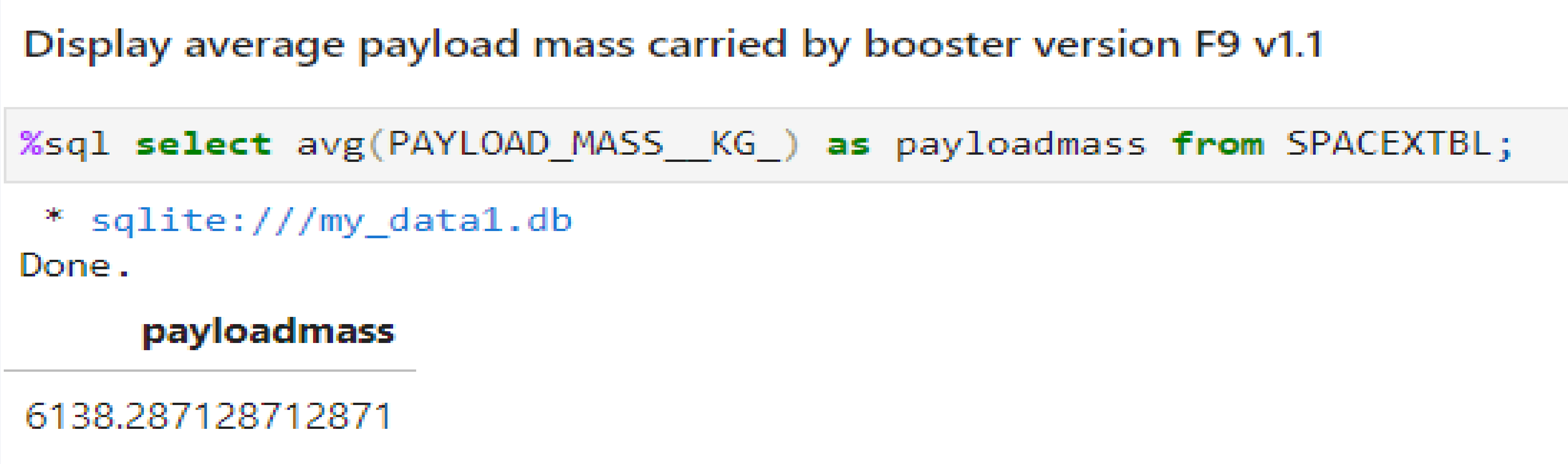
```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

payloadmass
619967

Average Payload Mass by F9 v1.1

- The average payload mass for rockets with booster version F9 v1.1 is 2928.4 kg.
- The function AVG is used to calculate the average of payload mass.
- The WHERE clause is used to specify that the booster version should be F9 v1.1



Display average payload mass carried by booster version F9 v1.1

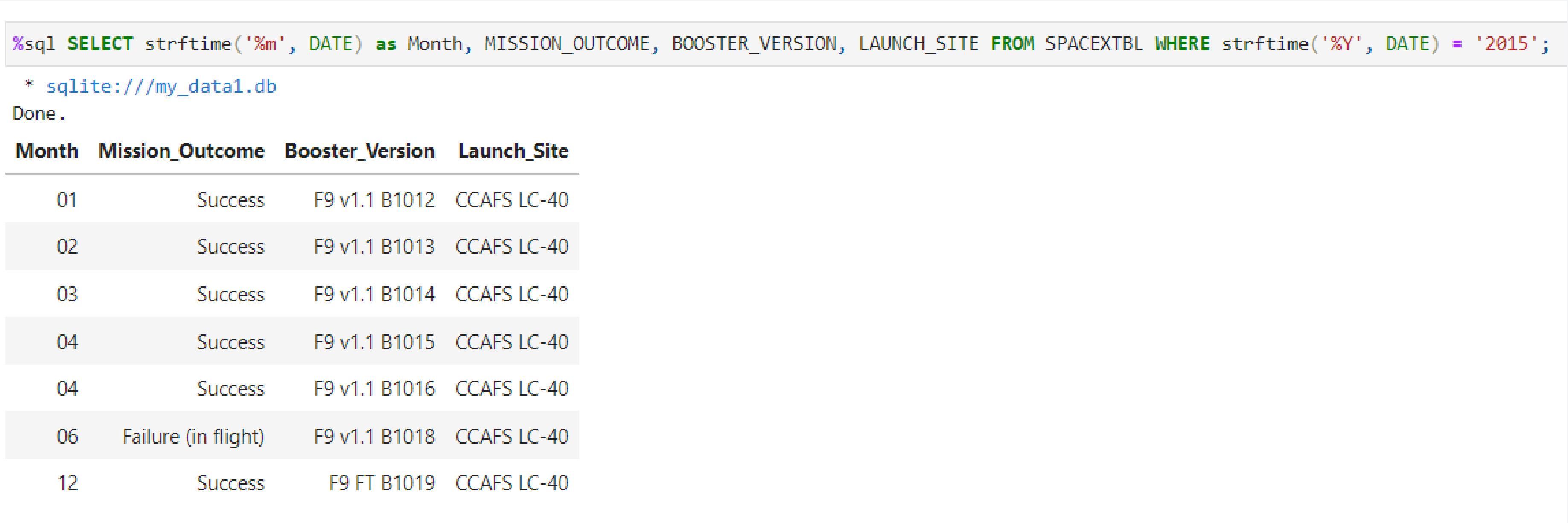
```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

payloadmass
6138.287128712871

First Successful Ground Landing Date

- The first successful ground landing occurred on 22 December 2015.
- The function MIN is used to find the earliest date.
- The WHERE clause is used to specify that the landing outcome is a successful ground pad landing.



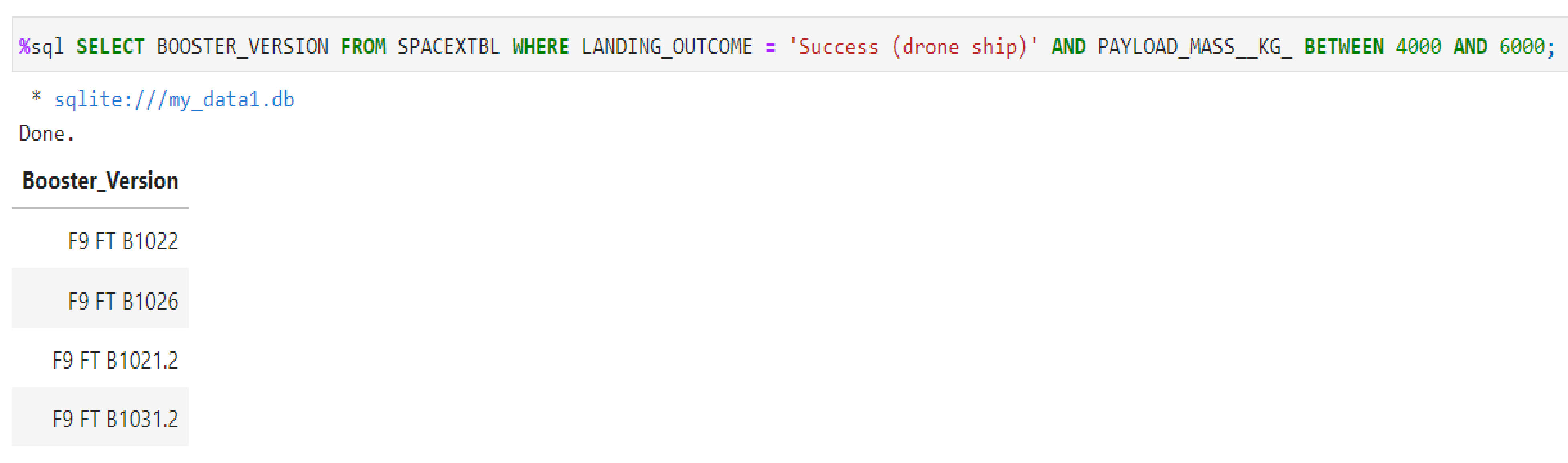
```
%sql SELECT strftime('%m', DATE) as Month, MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE strftime('%Y', DATE) = '2015';
```

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

Month	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters that have successfully landed on a drone ship and had payload mass greater than 4000 but less than 6000 –F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2
- The WHERE clause specifies that the landing was a successful drone ship landing.
- The AND clause adds another condition that the payload mass is between 4000 and 6000



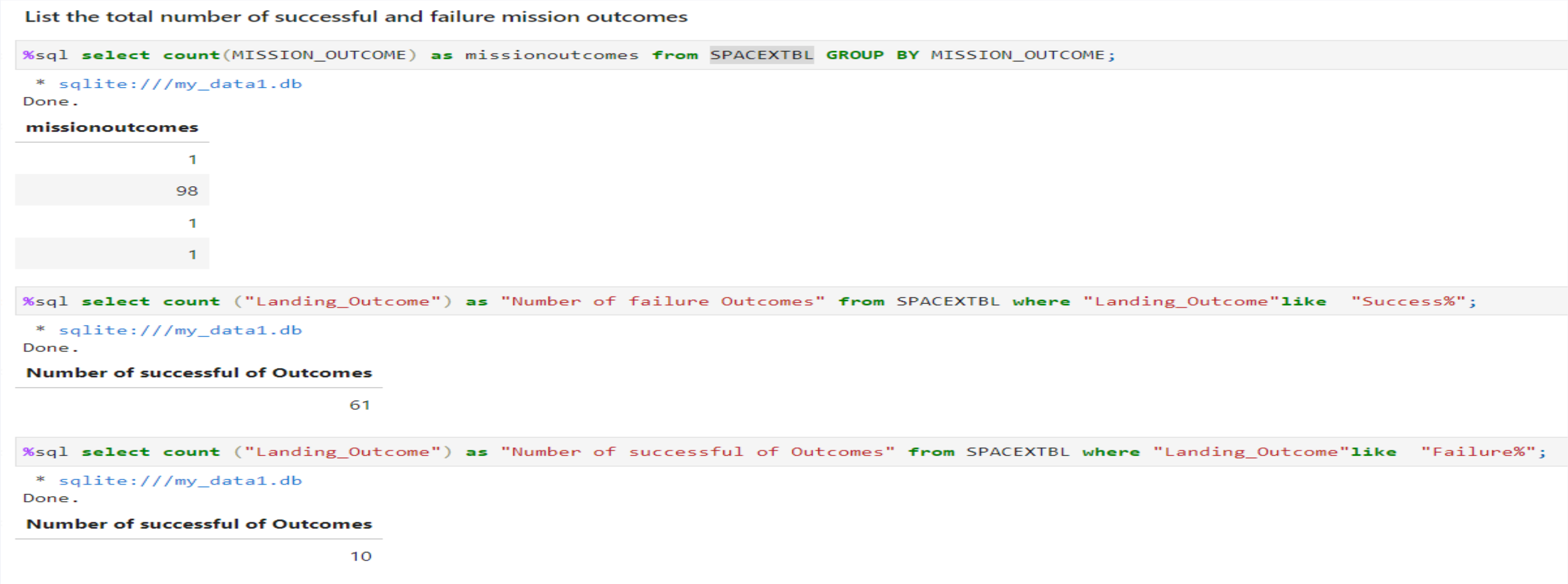
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * 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

- Total Number of Successful Mission Outcomes : 61
- Total Number of Failure Mission Outcomes : 10
- The COUNT function gives the number of landings.
- The LIKE clause is used to specify whether the outcome is a success or a failure.



List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

```
%sql select count ("Landing_Outcome") as "Number of failure Outcomes" from SPACEXTBL where "Landing_Outcome"like  "Success%";
```

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

Number of successful of Outcomes
61

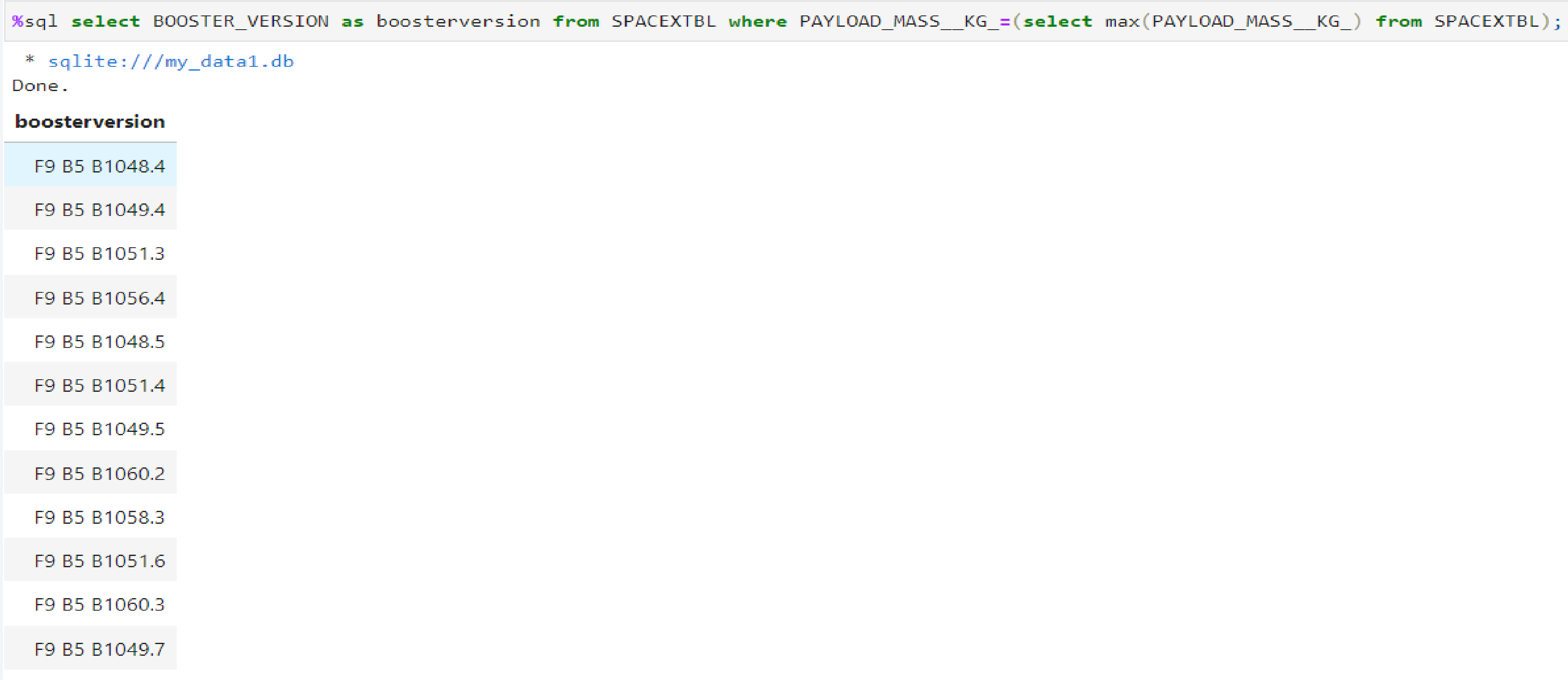
```
%sql select count ("Landing_Outcome") as "Number of successful of Outcomes" from SPACEXTBL where "Landing_Outcome"like  "Failure%";
```

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

Number of successful of Outcomes
10

Boosters Carried Maximum Payload

- The WHERE clause is used to specify that the booster version with the maximum payload mass is to be selected
- The MAX function gives the maximum payload mass.



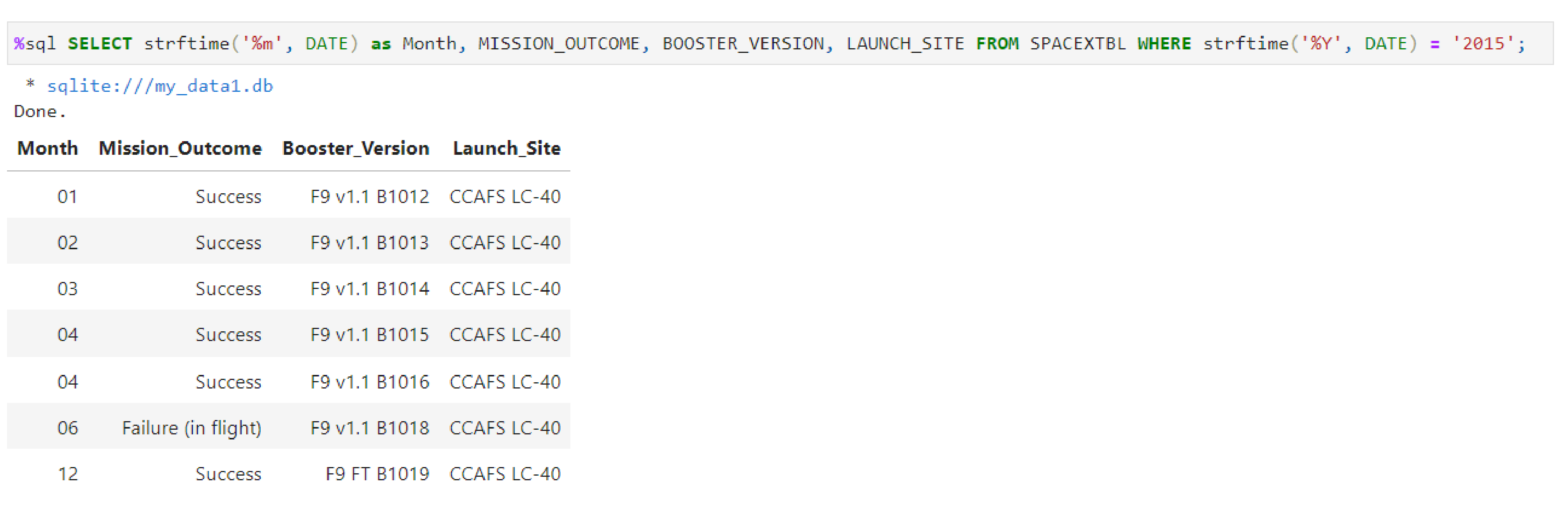
```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

boosterversion
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

- The strftime( ) function formats a date-time value based on a specific format.
- The WHERE clause specifies that the date should be 2015 and the landing outcome should be a drone ship failure.



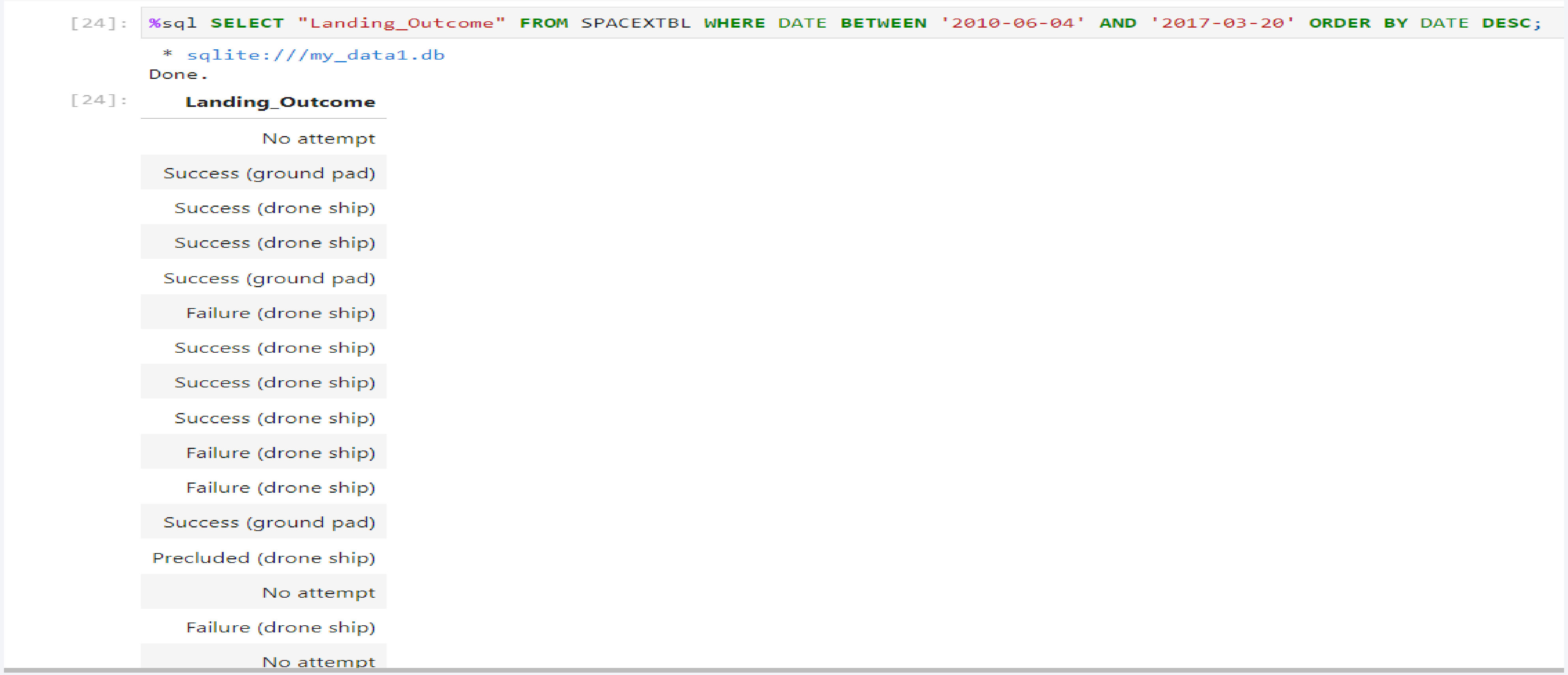
```
%sql SELECT strftime('%m', DATE) as Month, MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE strftime('%Y', DATE) = '2015';
```

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

Month	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order



```
[24]: %sql SELECT "Landing_Outcome" FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

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

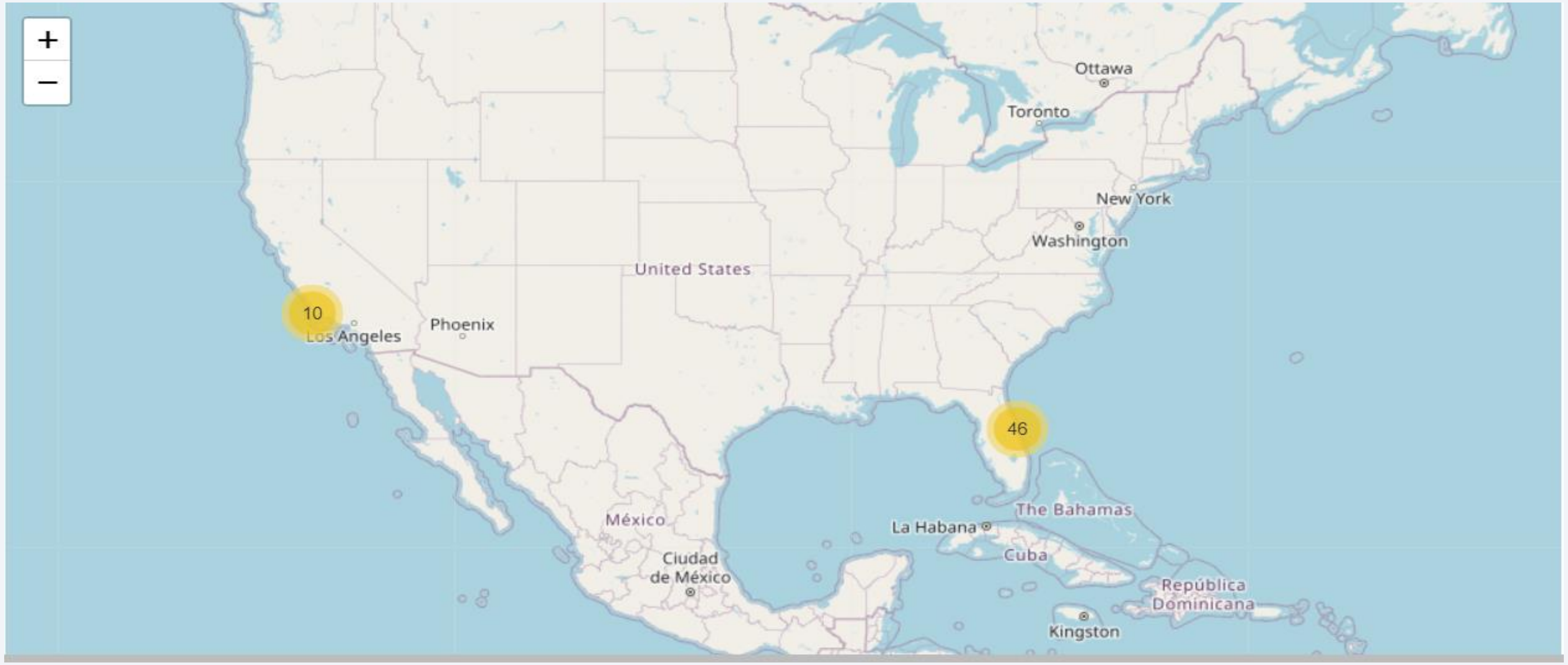
[24]:

Landing_Outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
No attempt
Failure (drone ship)
No attempt

Section 3

Launch Sites Proximities Analysis

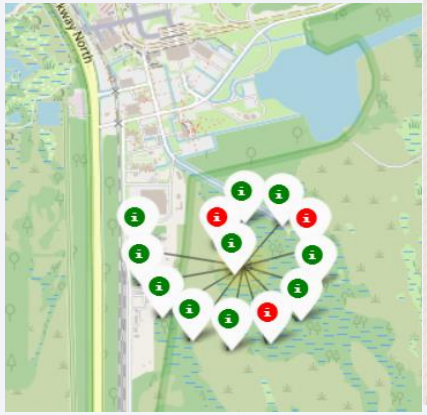
Space X Launch Sites on the Global Map

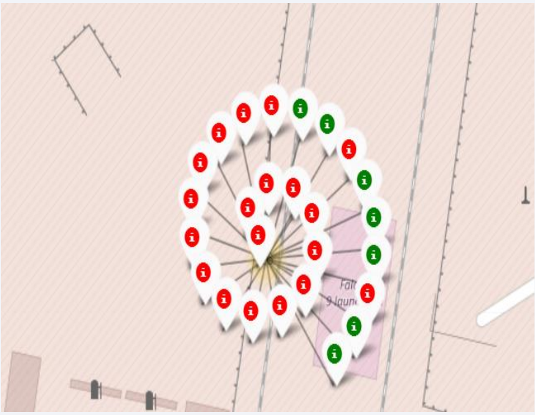


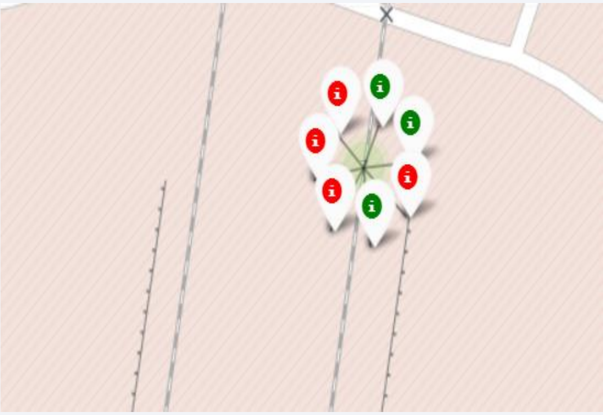
- The SpaceX Launch Sites are on the coasts of Florida and California in America
- Only the launch site VAFB SLC –4E is in California; the rest are in Florida

Success/Failed Launches for each Launch Site

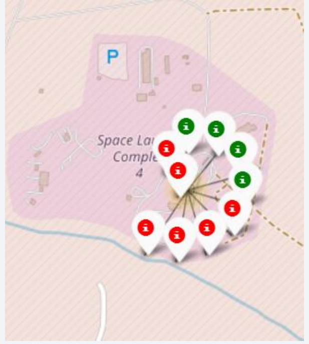
- Florida Launches







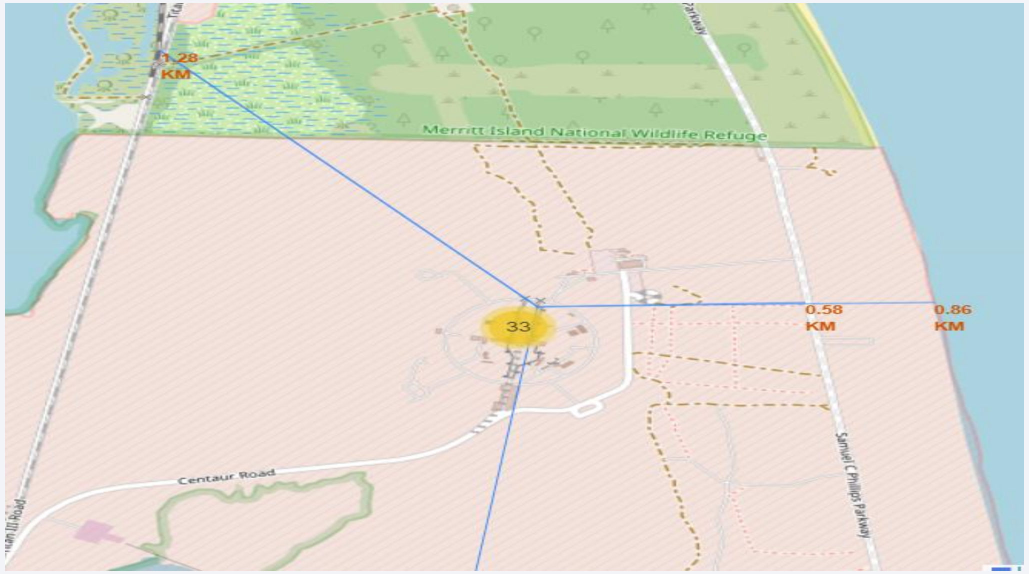
- California Launches

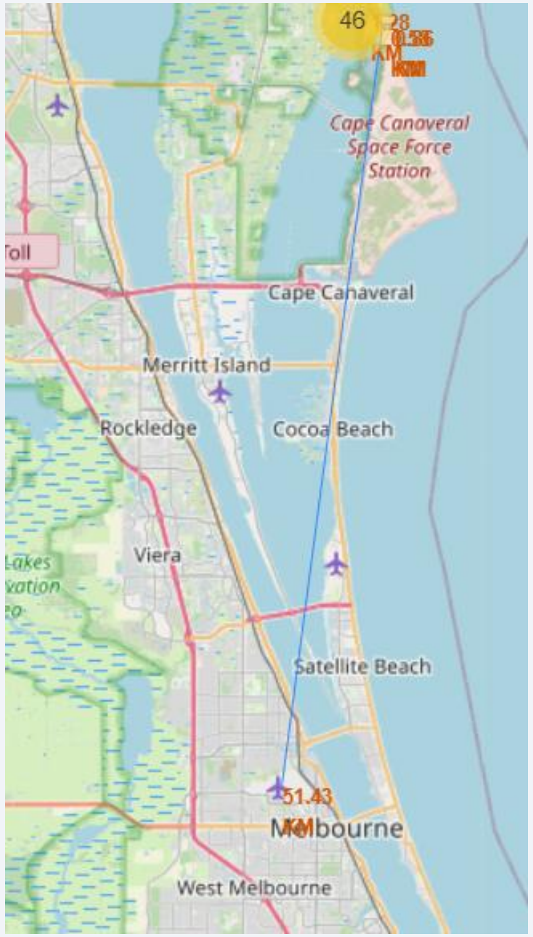


Success   Failed

Launches Sites Proximity From Landmarks

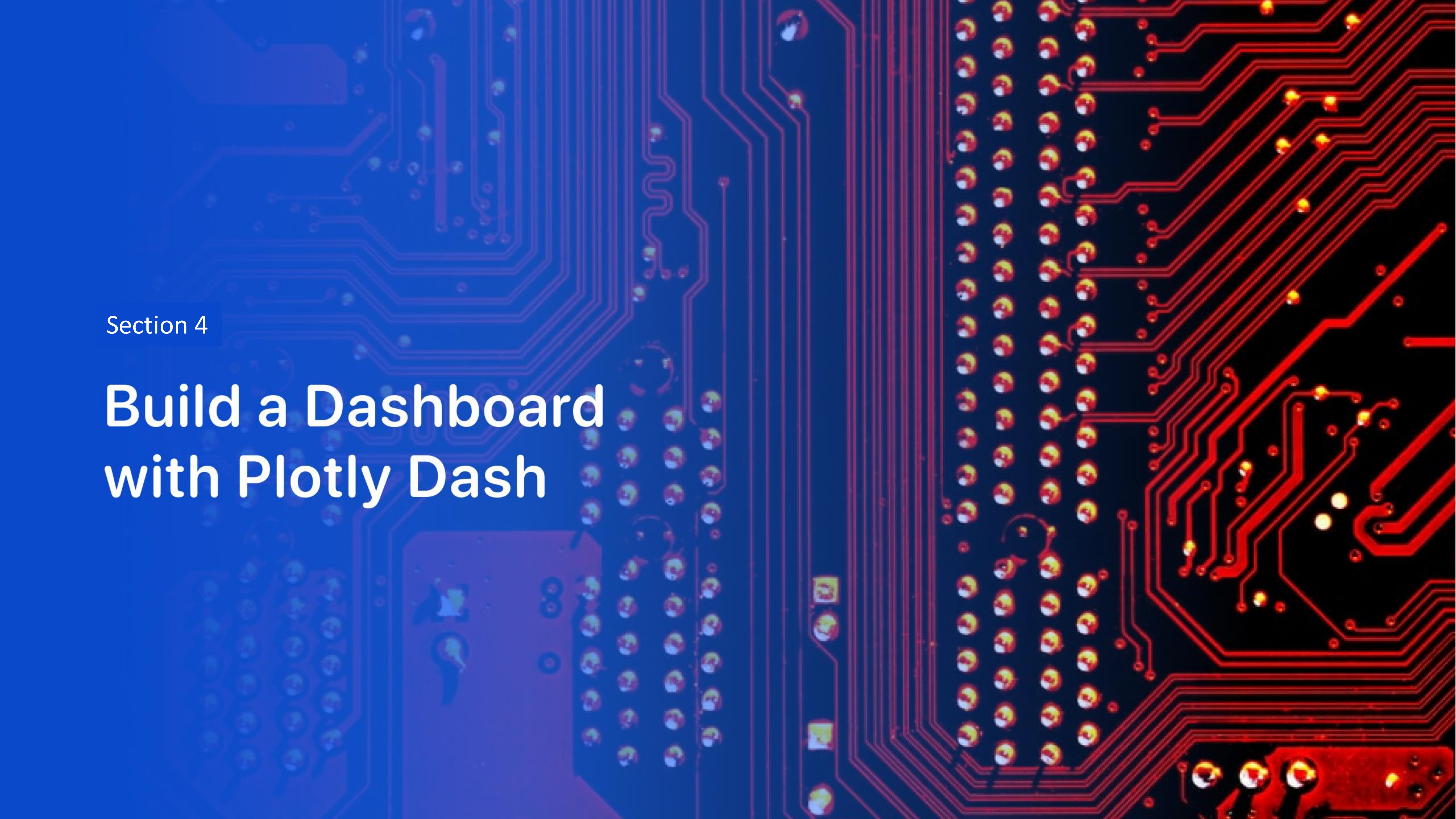
- Distance to the closest railway 1.28 km
- Distance to the closest Highway 0.58 km
- Distance to the closest coastline 0.84 km
- Distance to the closest city 51.43 km


1.28 KM
Merritt Island National Wildlife Refuge
33
0.58 KM
0.86 KM
Centaur Road
Samuel C Phillips Parkway



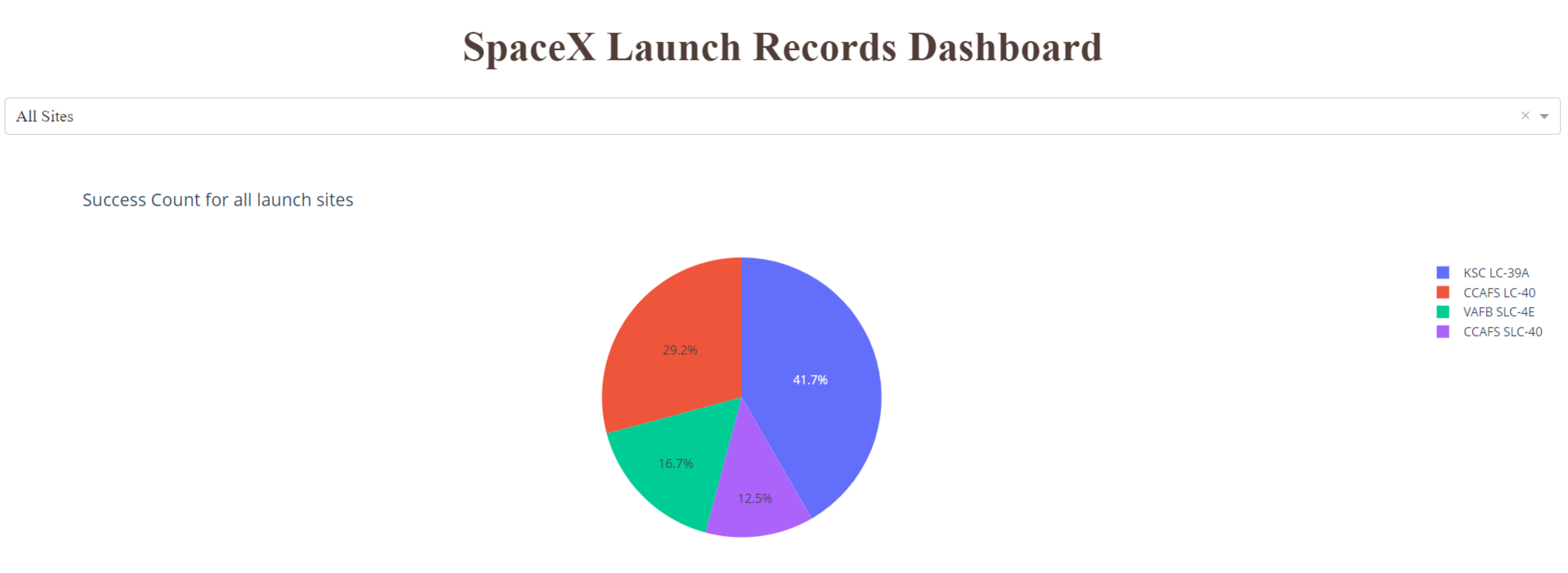
Section 4

Build a Dashboard with Plotly Dash



Pie Chart of Total Success Launches by all sites

- Launch Site KDC LC-39 A has the Greatest Launch Success Rate



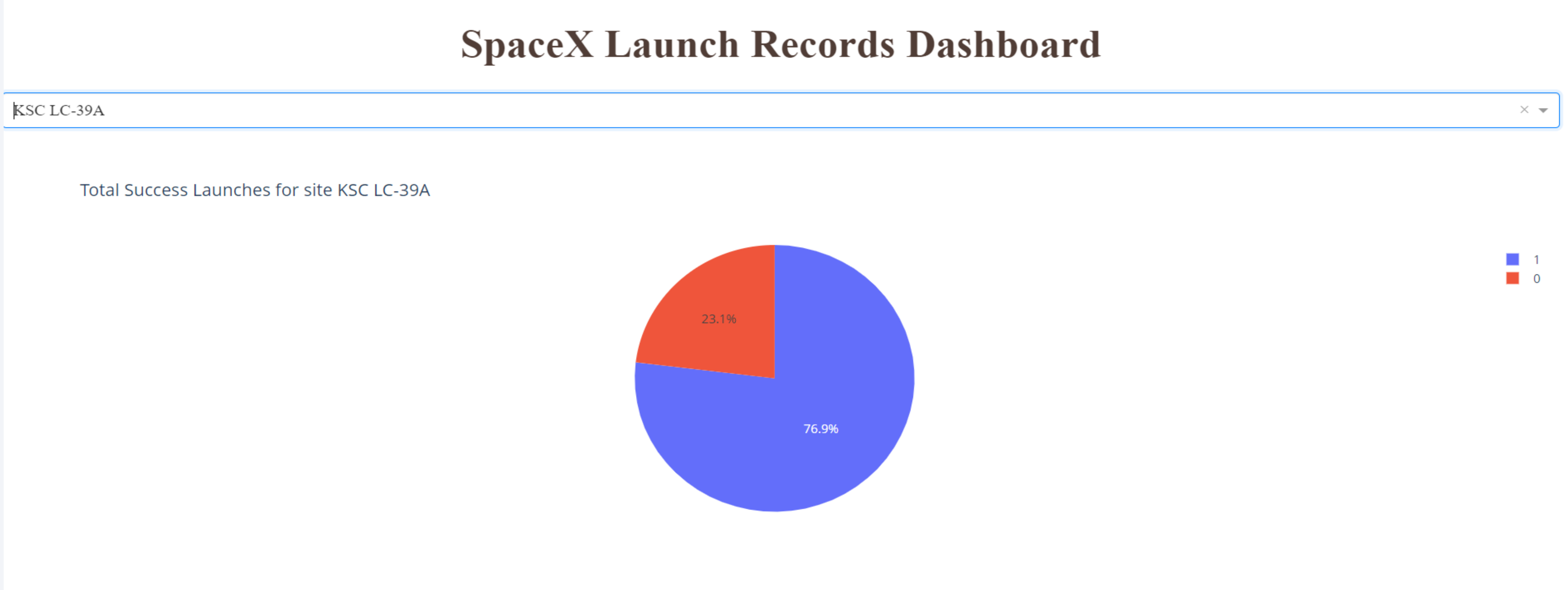
SpaceX Launch Records Dashboard

All Sites

Success Count for all launch sites

Pie Chart of Success /Failure Launch Rates of KSC LC-39A

- We see that the launch site KSC LC-39A achieved a launch success rate of 76.9%

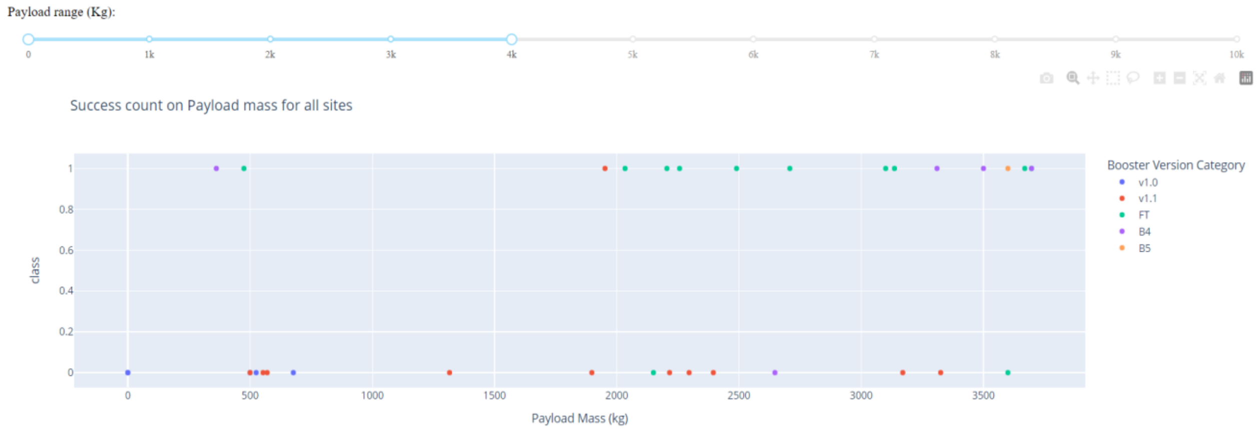


SpaceX Launch Records Dashboard

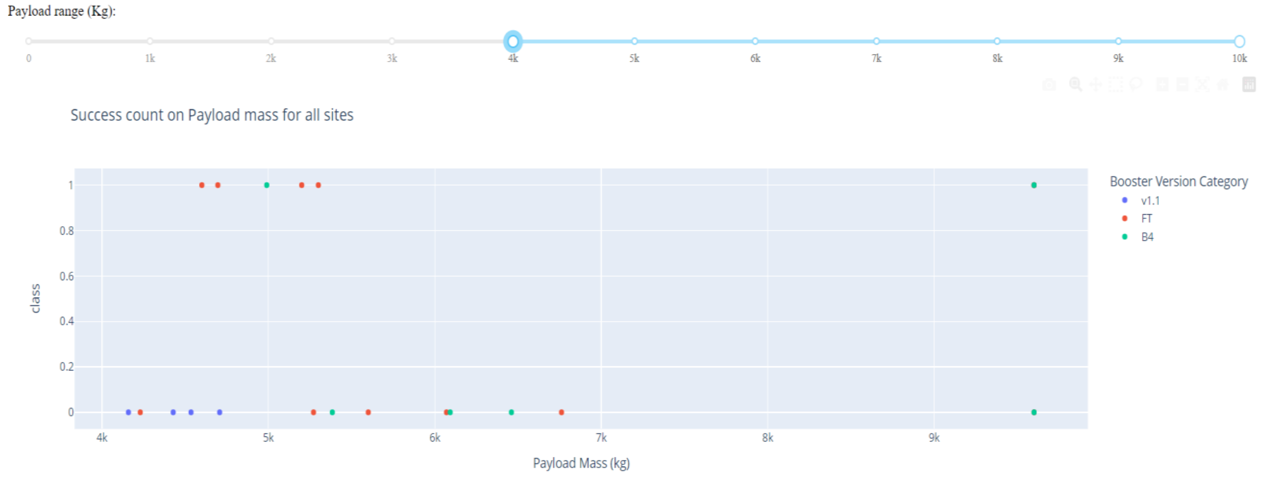
KSC LC-39A

Payload vs. Launch Outcome scatter plot for all sites

- Low Payload Weight 0 – 4000 kg



- Heavy Payload Weight 4000 –10,000 kg
- Success Rate for Heavier Payload is lesser than that for Lighter Payload

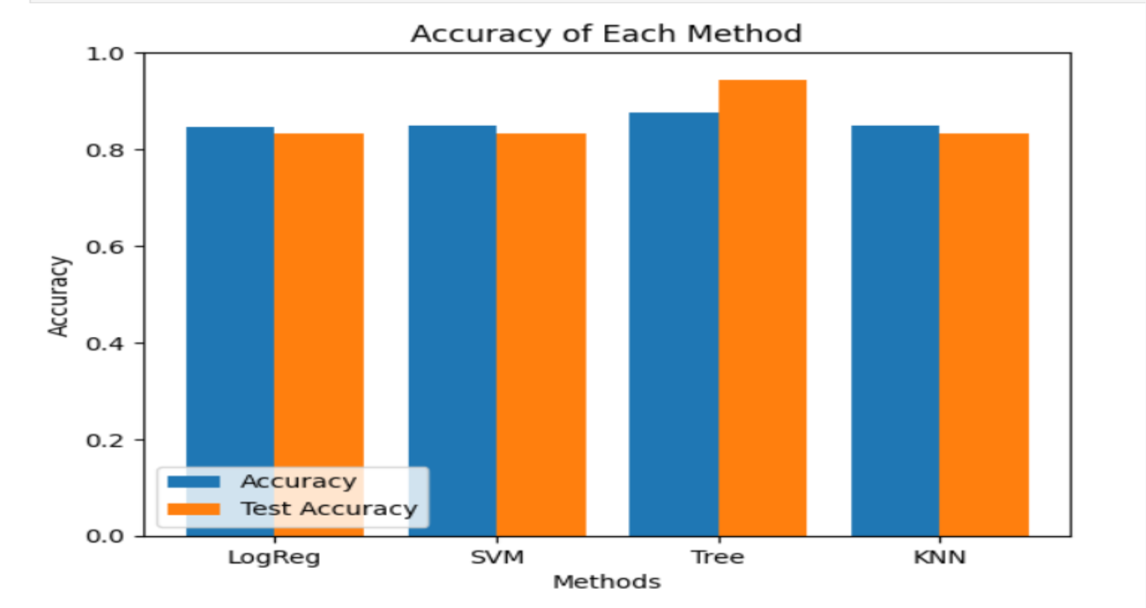


Section 5

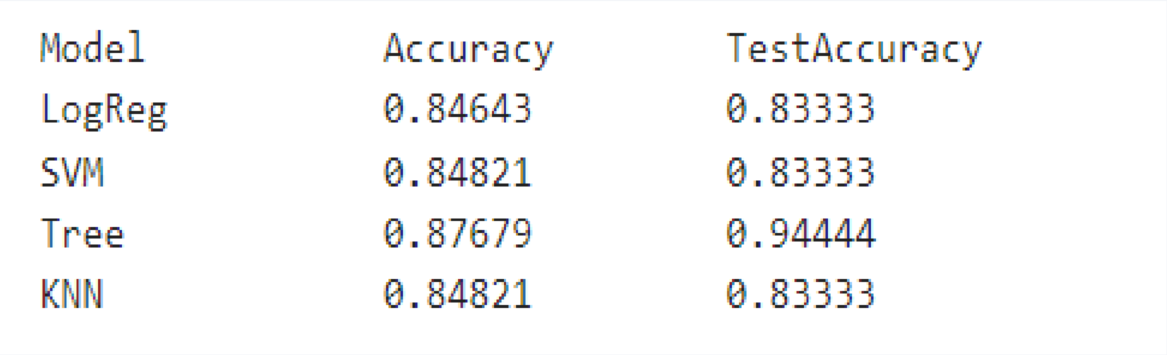
Predictive Analysis (Classification)

Classification Accuracy

- Visualization of the built model accuracy for all built classification models, in a bar chart



- The Tree model has the highest classification accuracy

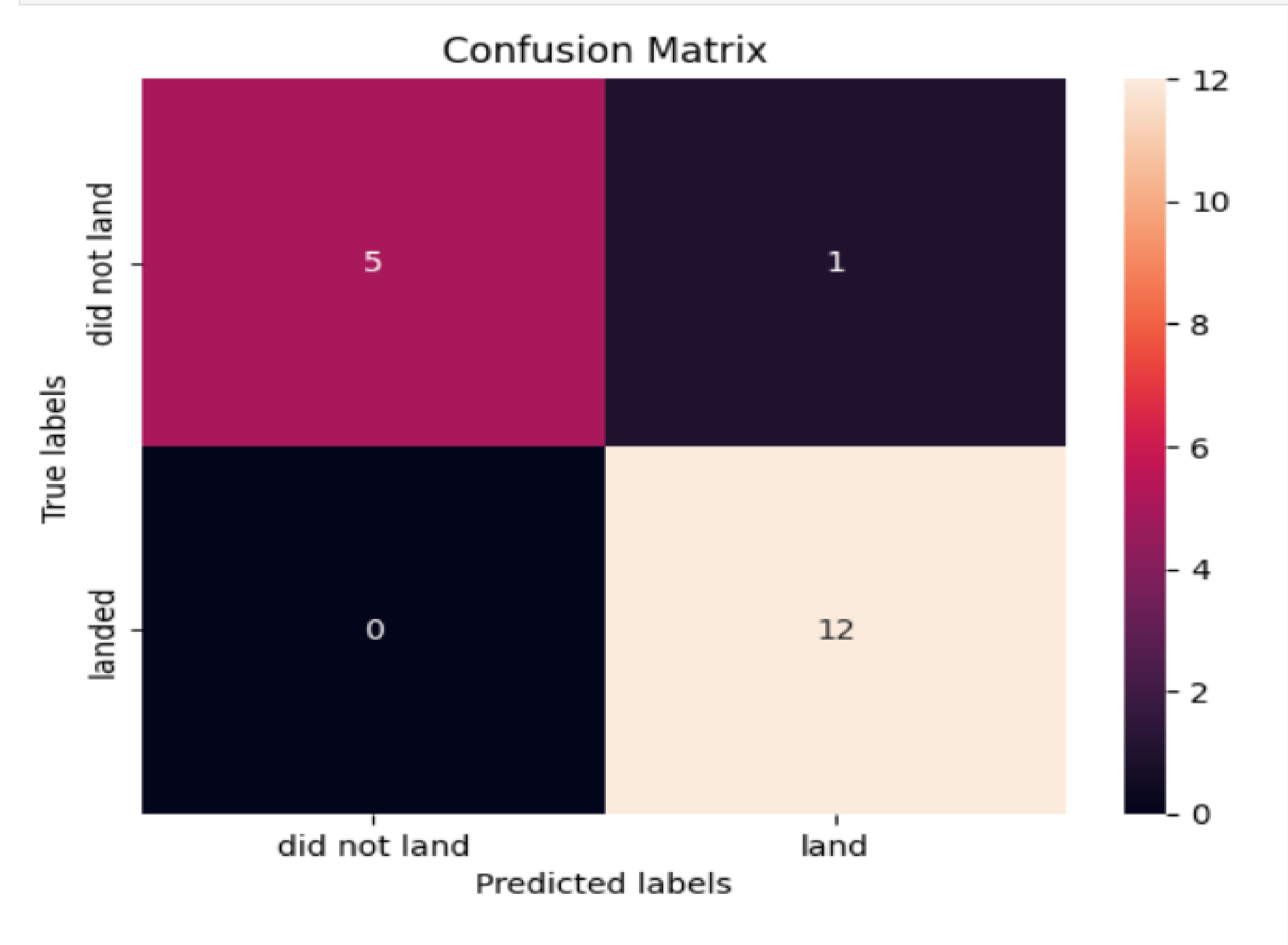


```
Model          Accuracy       TestAccuracy
LogReg         0.84643        0.83333
SVM            0.84821        0.83333
Tree           0.87679        0.94444
KNN            0.84821        0.83333
```

Confusion Matrix

- The Best Model is the Decision Tree Classifier.

- The confusion matrix for the Decision Tree Classifier shows that the classifiers give a high number of false positives, i.e., the rocket did not fail successfully, but the classifiers predicted that it was not successful.



Conclusions

- The greater the number of flights at a launch site, the greater the success rate at the launch site.
- ES-L1, GEO, HEO, and SSO orbits have the highest success rate.
- For heavier payloads, the success rate increases for LEO, ISS, and Polar orbits.
- Launch success rate increased from 2013 to 2020 with minor fluctuations throughout the period.
- Launch Site KDC LC-39 A has the best Launch Success Rate
- The Decision Tree Classifier is the best machine learning algorithm to predict SpaceX Falcon 9 rocket's first stage will land successfully.

Thank you!