IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Breeze Vu
December 11, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive Analytics result and Dashboards
  - Predictive Analytics result

Introduction

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars while other providers cost upward of 165 million dollars each. Much of the savings is because Space X can reuse the first stage of the launch by re-land the rocket to be used on the next mission. 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.

The problems included:

- Identifying all factors that influence the landing outcome,
- Identifying the relationship between different features and how they are affecting the outcome,
- Indicates the best conditions to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

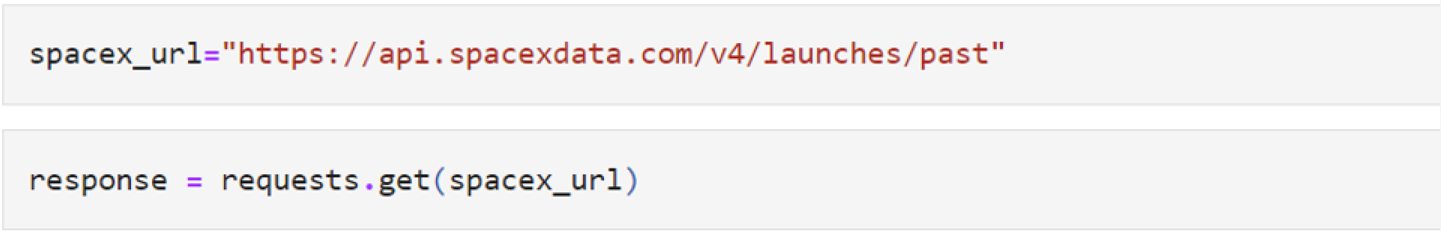
- Data was collected using SpaceX REST API and Web Scrapping
- Perform data wrangling using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data was divided in training and test data sets and evaluated by four different classification models. The accuracy of each model was evaluated using different combinations of parameters.

Data Collection

- Data was collected using a variety of methods. As mentioned, here the dataset is collected using SpaceX API and Web Scraping.
- For SpaceX API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe using json_normalize(). Next, we then cleaned the data, checked for missing values and fill in missing values where necessary.
- For Web Scrapping, we will use the BeautifulSoup 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

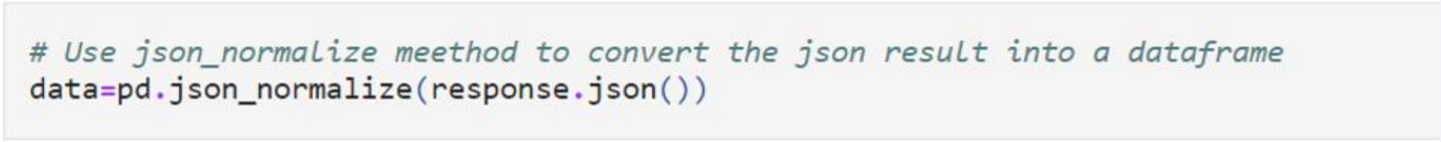
- Get request for rocket launch data using API



```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

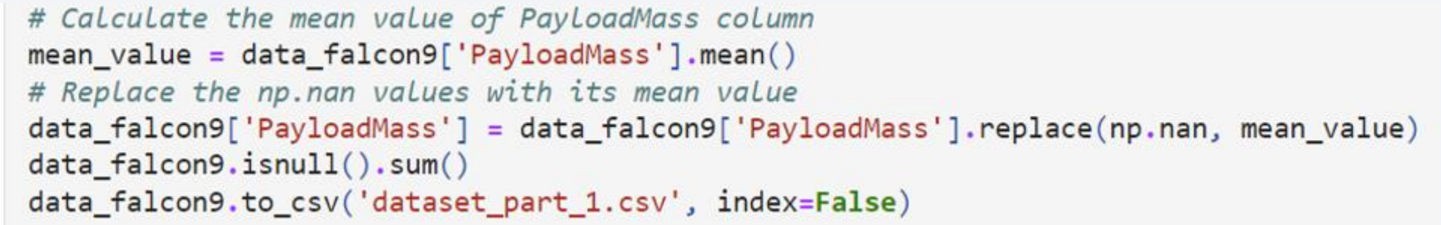
```
response = requests.get(spacex_url)
```

- Use json_normalize method to convert json result to dataframe



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

- Dealing with missing values and export to file

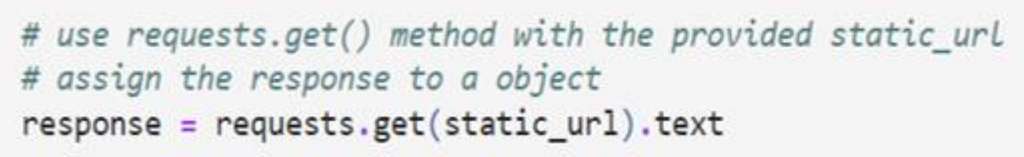


```
# Calculate the mean value of PayloadMass column
mean_value = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean_value)
data_falcon9.isnull().sum()
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

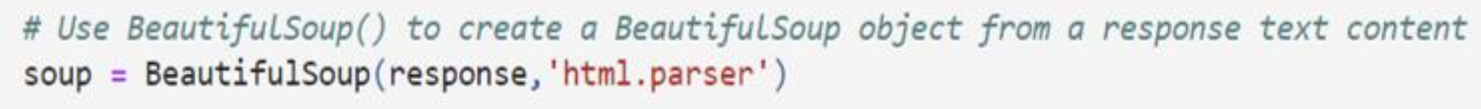
Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL



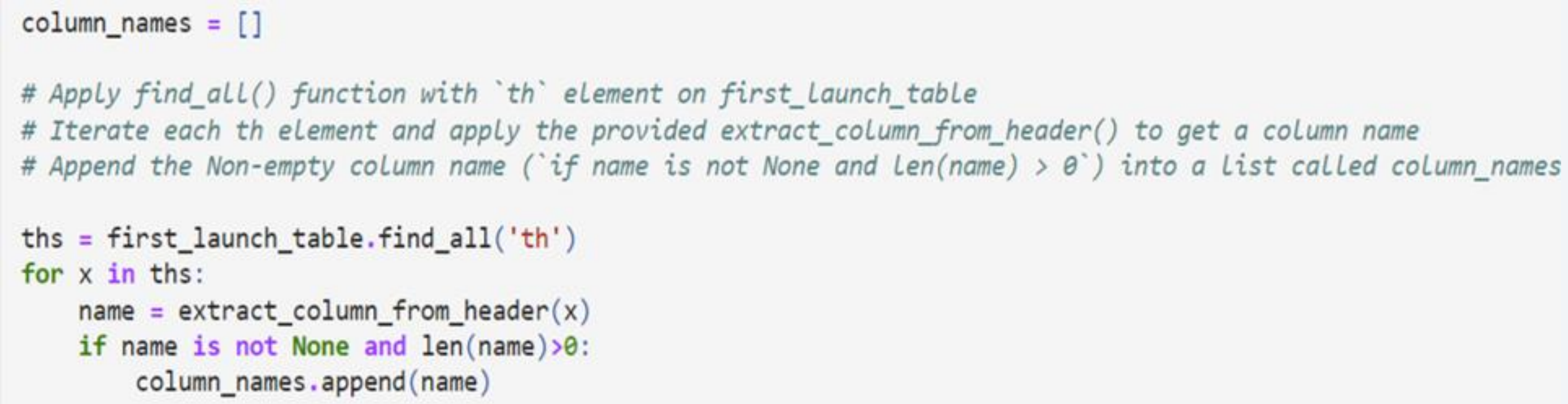
```python
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

- Create a BeautifulSoup object from the HTML response



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

- Extract all column names from the HTML table header

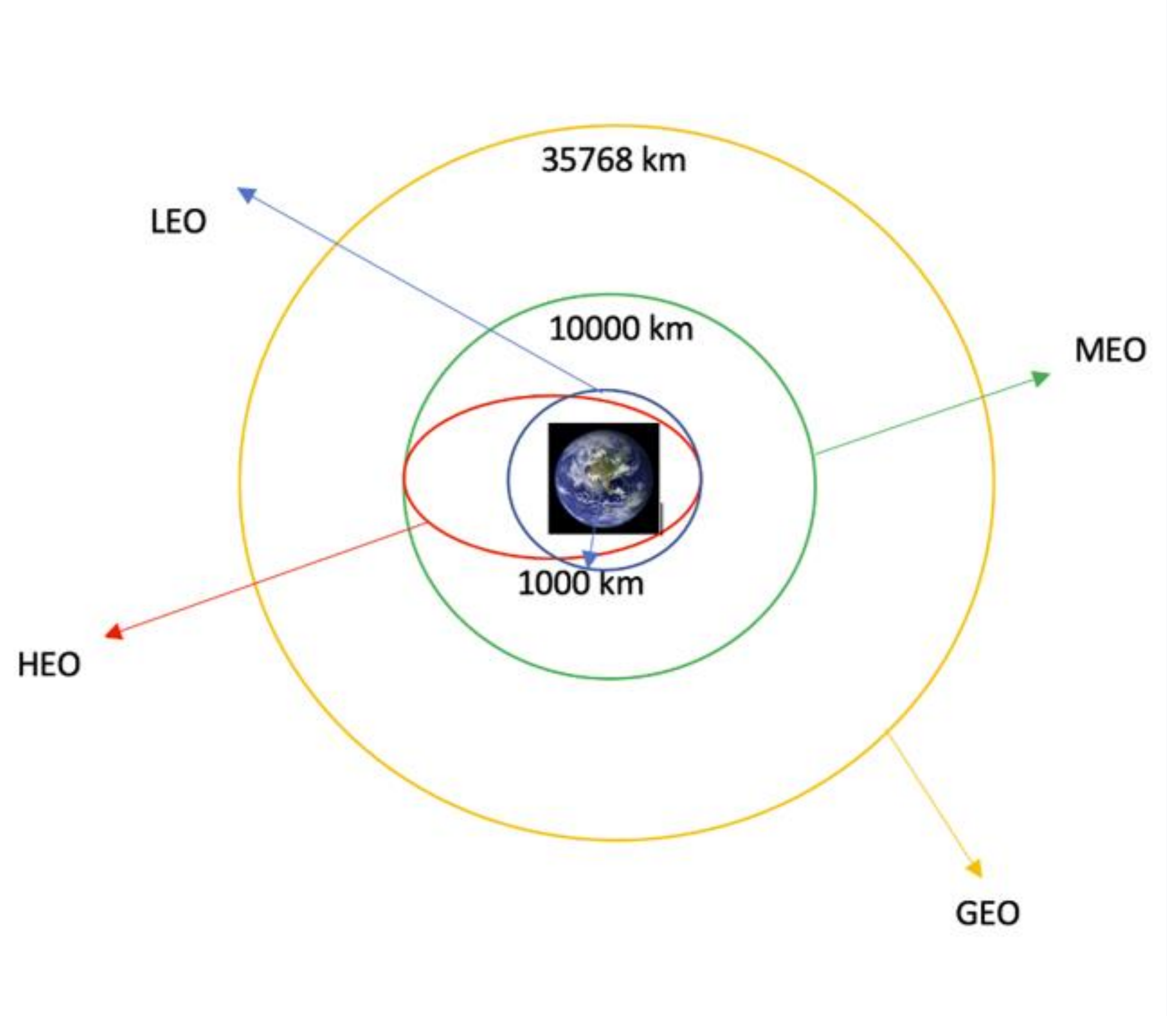


```python
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

ths = first_launch_table.find_all('th')
for x in ths:
    name = extract_column_from_header(x)
    if name is not None and len(name)>0:
        column_names.append(name)
```

- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/jupyter-labs-webscraping.ipynb

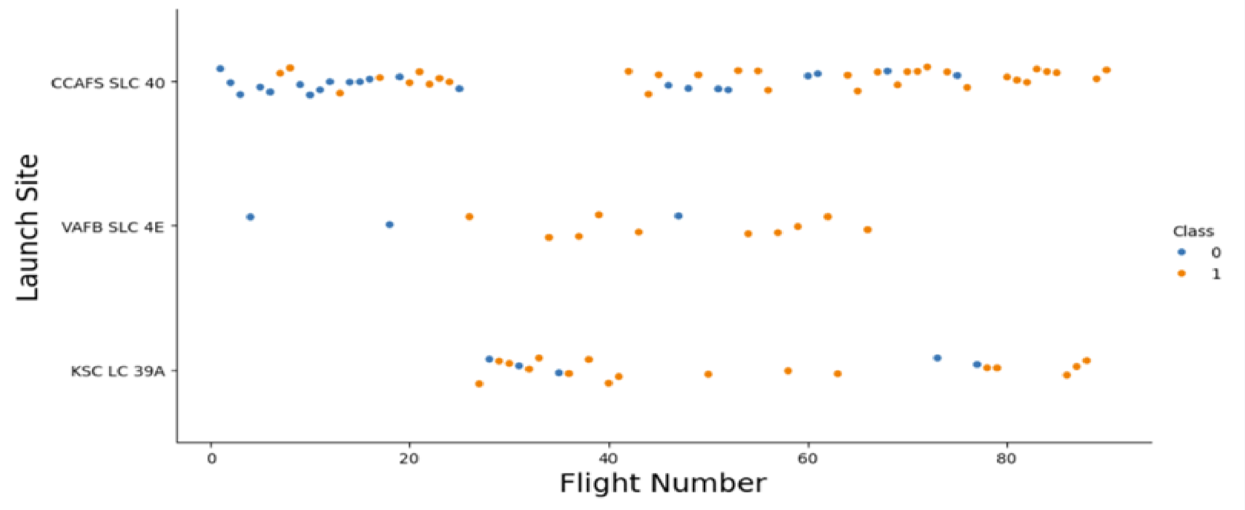
Data Wrangling

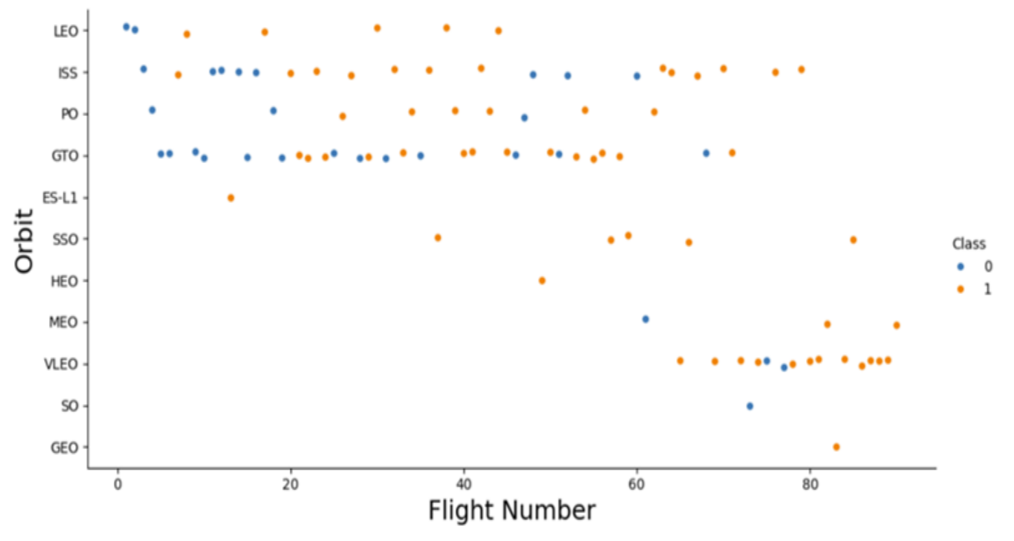
- First, some Exploratory Data Analysis (EDA) was performed on the dataset.
- Then, the number of launches at each site as well as the number and occurrence of each orbits were calculated.
- Finally, the landing outcome label was created from Outcome column and exported to file.
- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

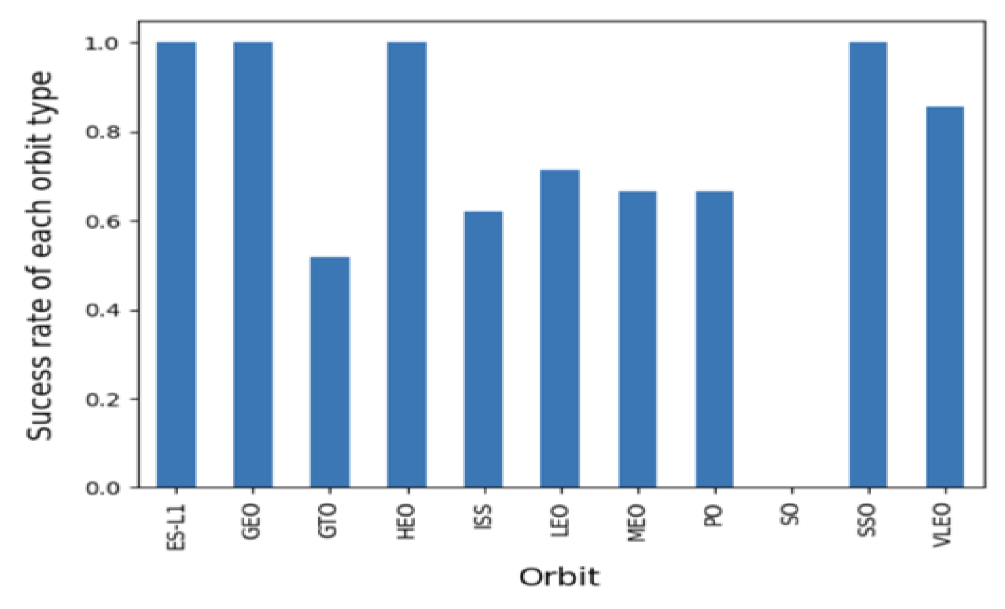
- We explored the data by visualizing the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumber and Orbit type, Payload and Orbit type, success rate of each orbit type, and the launch success yearly trend.
- First, we started by using scatter charts to find the relationship between the attributes.





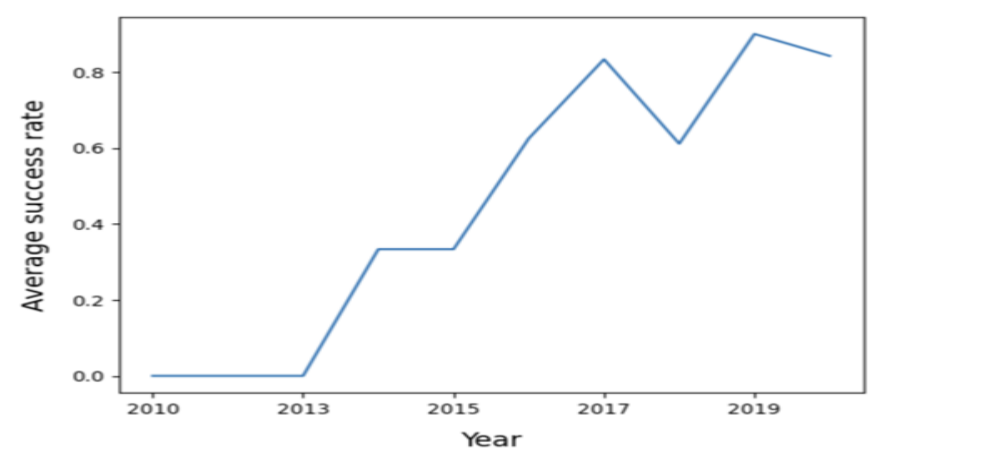
- For the relationship between Flight Number and Launch Site, we see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- As for the relationship between FlightNumber and Orbit type, you should see that in the LEO orbit the Success appears related to the number of flights. On the other hand, there seems to be no relationship between flight number when in GTO orbit.

EDA with Data Visualization



- Next, we will use other visualization tools such as bar charts and line charts for further analysis.
- The bar charts is one of the easiest way to interpret the relationship between the attributes. In this case, we will use the bar chart to determine which orbits have the highest rate of success.
- The line charts shows a trends or pattern of the attribute over time. In this case, it used to determine the average yearly trend of successful launches. You can observe that the success rate since 2013 continues to increase until 2020.

- Finally, we use Feature Engineering to predict success in future modules by creating dummy variables for the categorical columns.
- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb



EDA with SQL

- To implement Exploratory Data Analysis using SQL, we performed some of the following queries:
  - Display the names of the unique launch sites in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'.
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - 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.
- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- First, we took the coordinates (latitude and longitude) at each launch site and added a circle marker around each launch site with a label with the name of the launch site.
- Then, we create markers for all launch records. If a launch was successful (class=1), then we use a Green marker and if a launch was failed, we use a Red marker (class=0).
- Next, we used the Haversine formula to calculated the distances between a launch site to its proximities.
- Finally, once we have drawn the distance lines from the launch sites to their proximities, we can easily answer the questions:
  - Are launch sites in close proximity to railways?
  - Are launch sites in close proximity to highways?
  - Are launch sites in close proximity to coastline?
- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

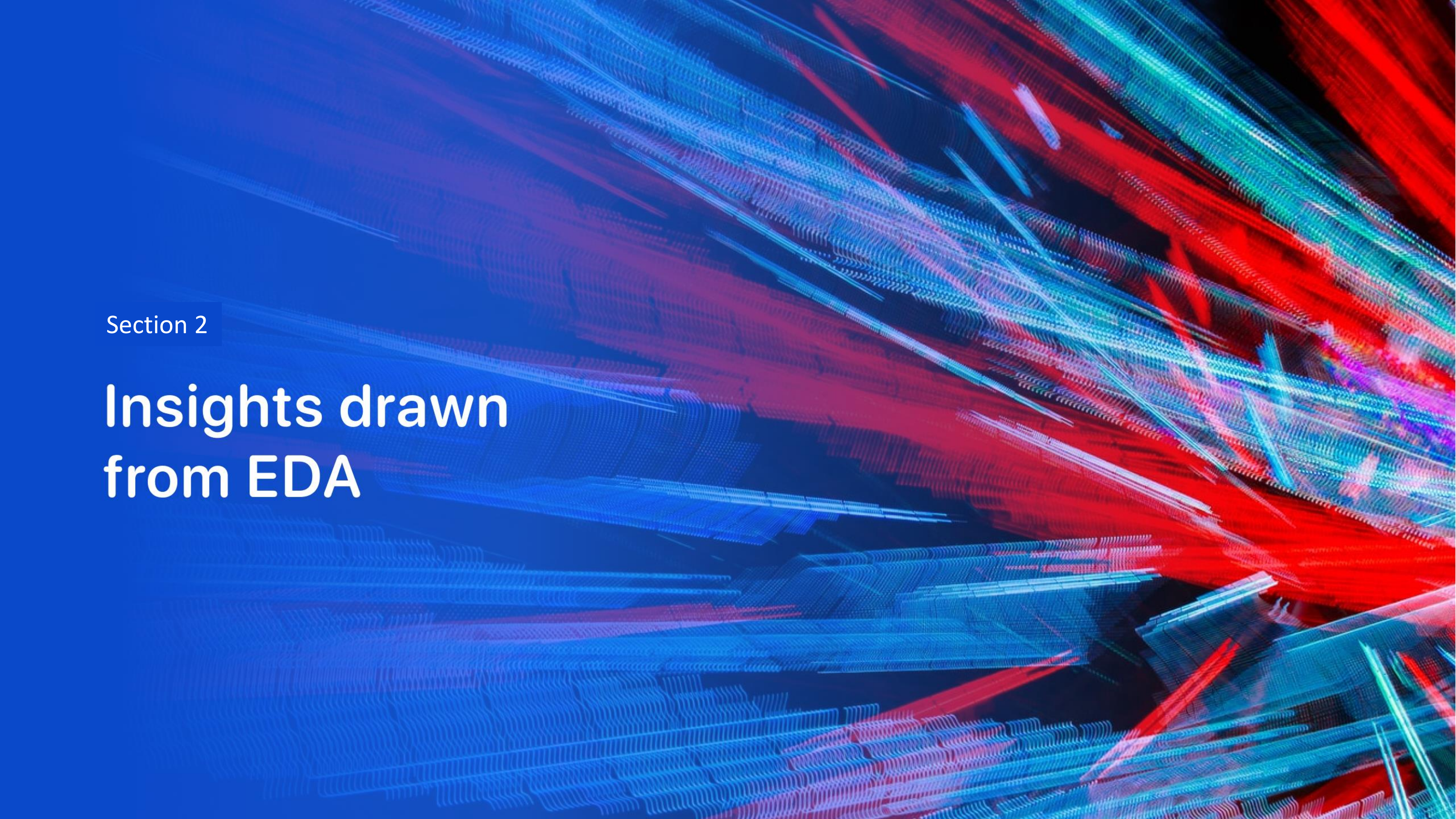
- We built an interactive dashboard with Plotly Dash that allows users to see data when they need it.
- We have plotted pie charts to show the total number of successful launches for a certain launch site or sites.
- For the selected site(s), we also plotted scatter charts showing the relationship between Outcome and Payload Mass (Kg) for different boosters.
- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- The development process that was applied to find the best performing classification model was:
  - Prepare Data
    - Load the data using numpy and pandas, then transform and split our data into training and testing datasets.
  - Build the Model
    - Build different machine learning models (SVM, Logistic Regression, Classification Trees and KNN) and tune different hyperparameters using GridSearchCV.
  - Evaluate the Model
    - Get the best hyperparameters for each model type,
    - Compute the accuracy for each model with test dataset,
    - Improve the model using feature engineering and algorithm tuning.
  - Compare to find the best Model
    - Compare models according to their accuracy. The model with the best accuracy will be chosen.
- The link to the notebook is: https://github.com/mrthanhvu/testrepo/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

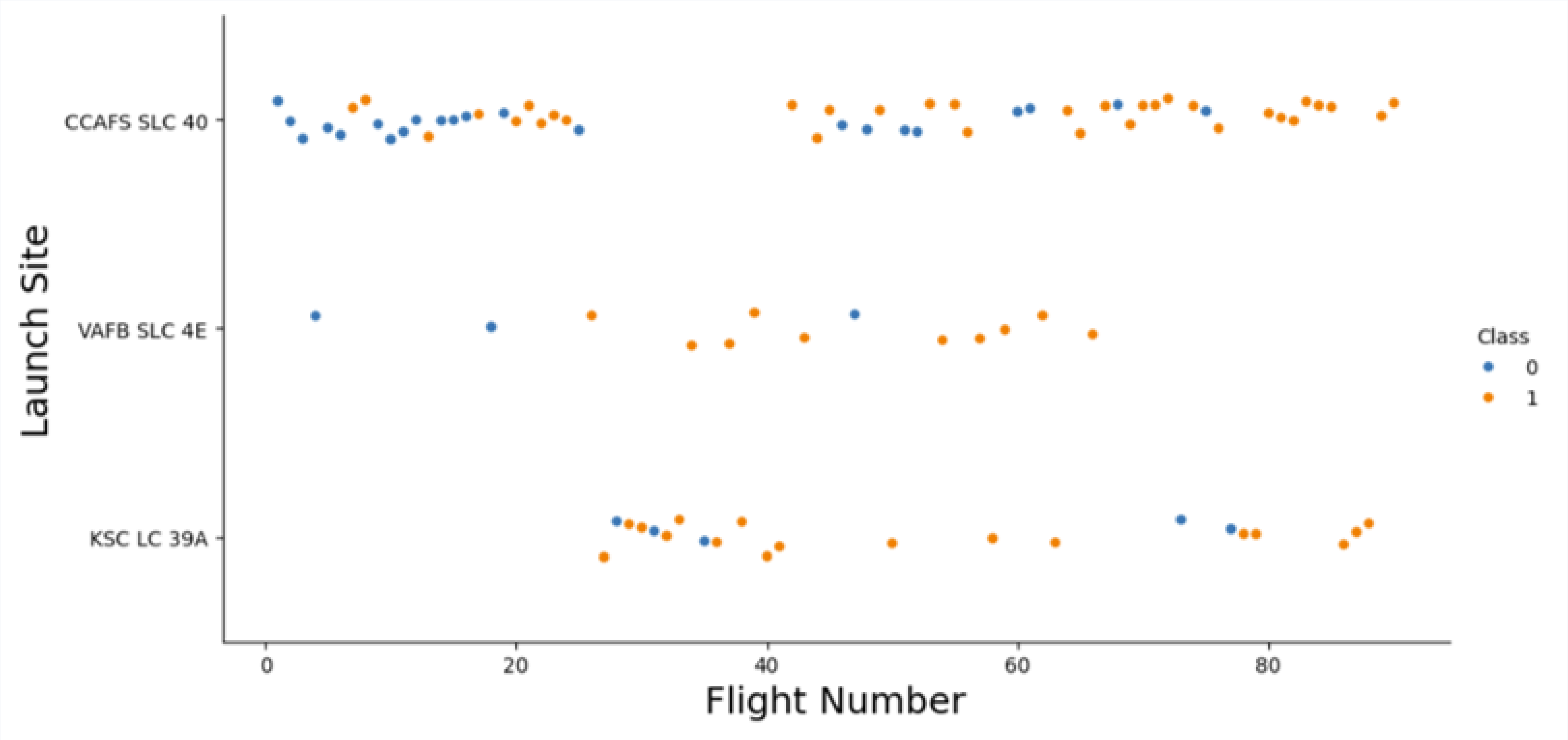
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

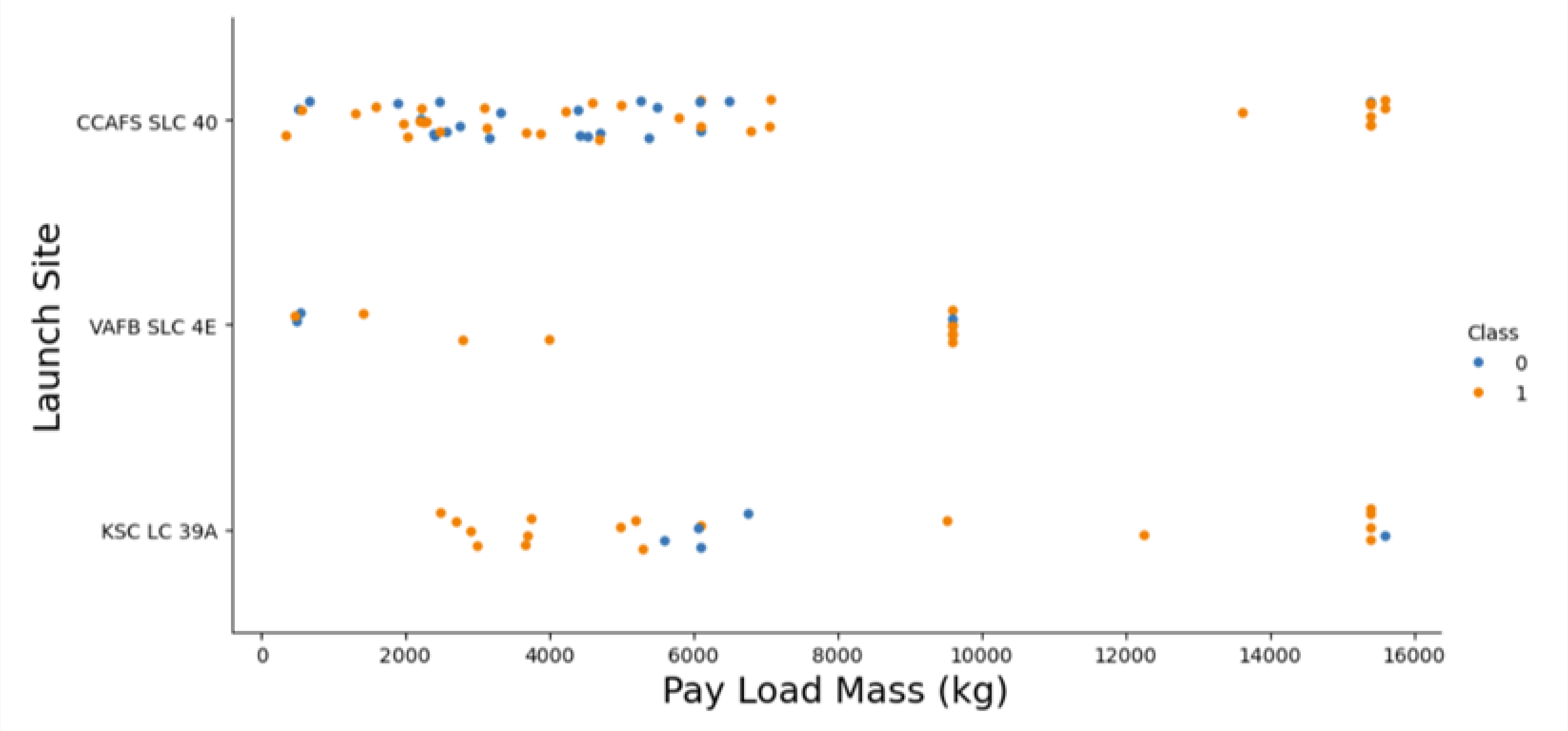
Insights drawn from EDA

Flight Number vs. Launch Site



- This figure depicts the relationship between Flight Number and Launch Site. From the scatter chart, we realized that the larger the number of flights at the launch site, the higher the success rate at the launch site.

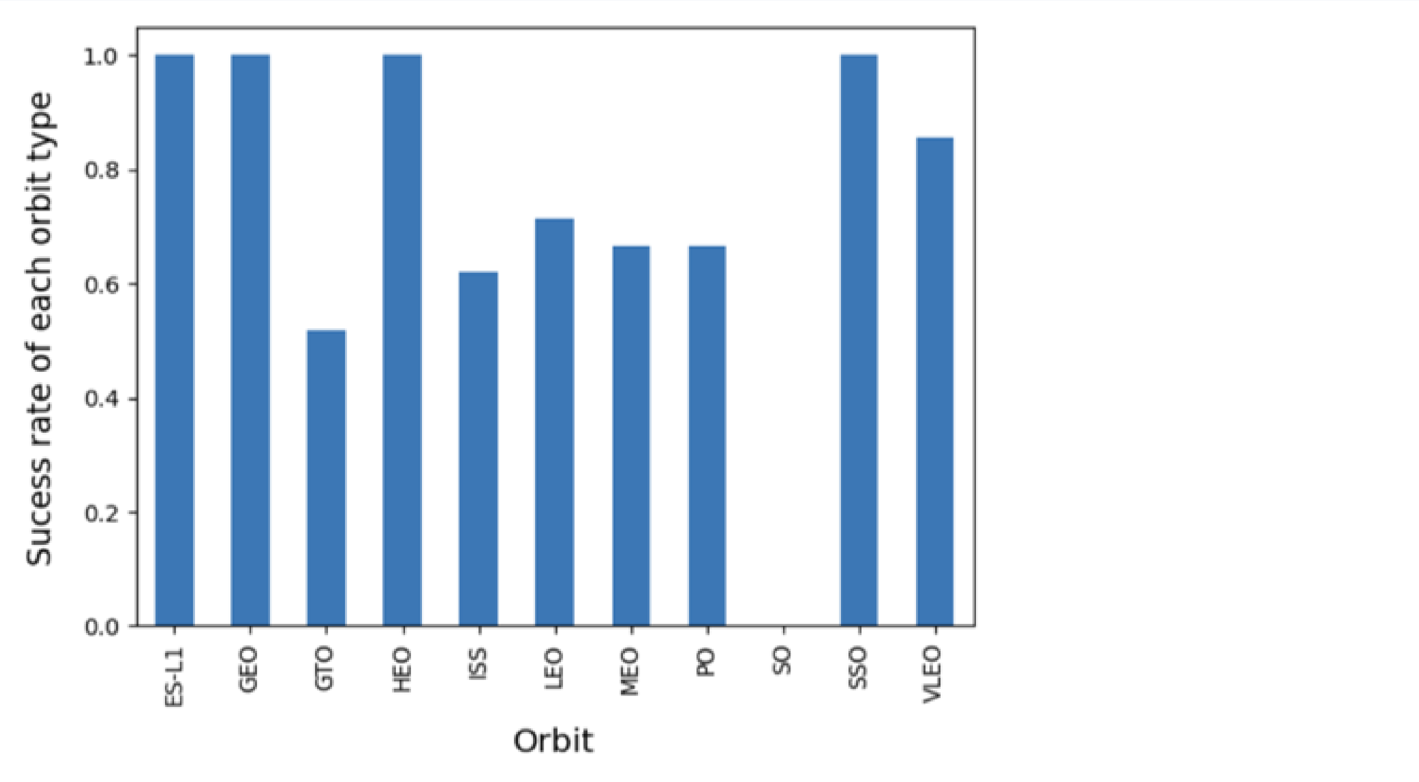
Payload vs. Launch Site



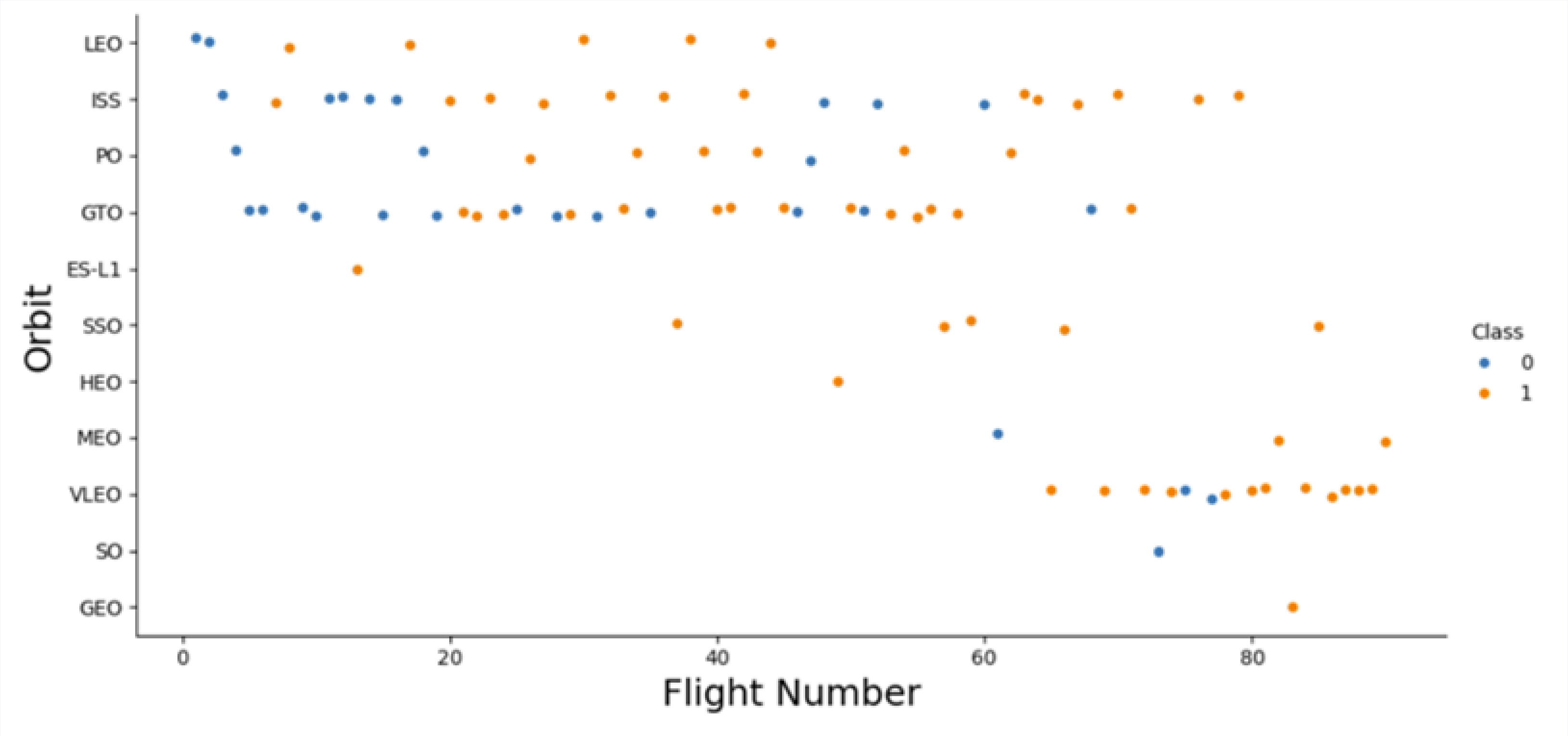
- This figure depicts the relationship between Payload and Launch Site. From the scatter chart, we realized that for the VAFB-SLC 4E launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type

This figure depicts the relationship between success rates and orbit types. From the bar chart, we realized that the success rate with different types of orbits is different. Some orbits have a success rate of 100% such as SSO, HEO, GEO and ES-L1 while others have a success rate of just 50% to 60% (GTO, ISS, etc.) or even 0% (SO).

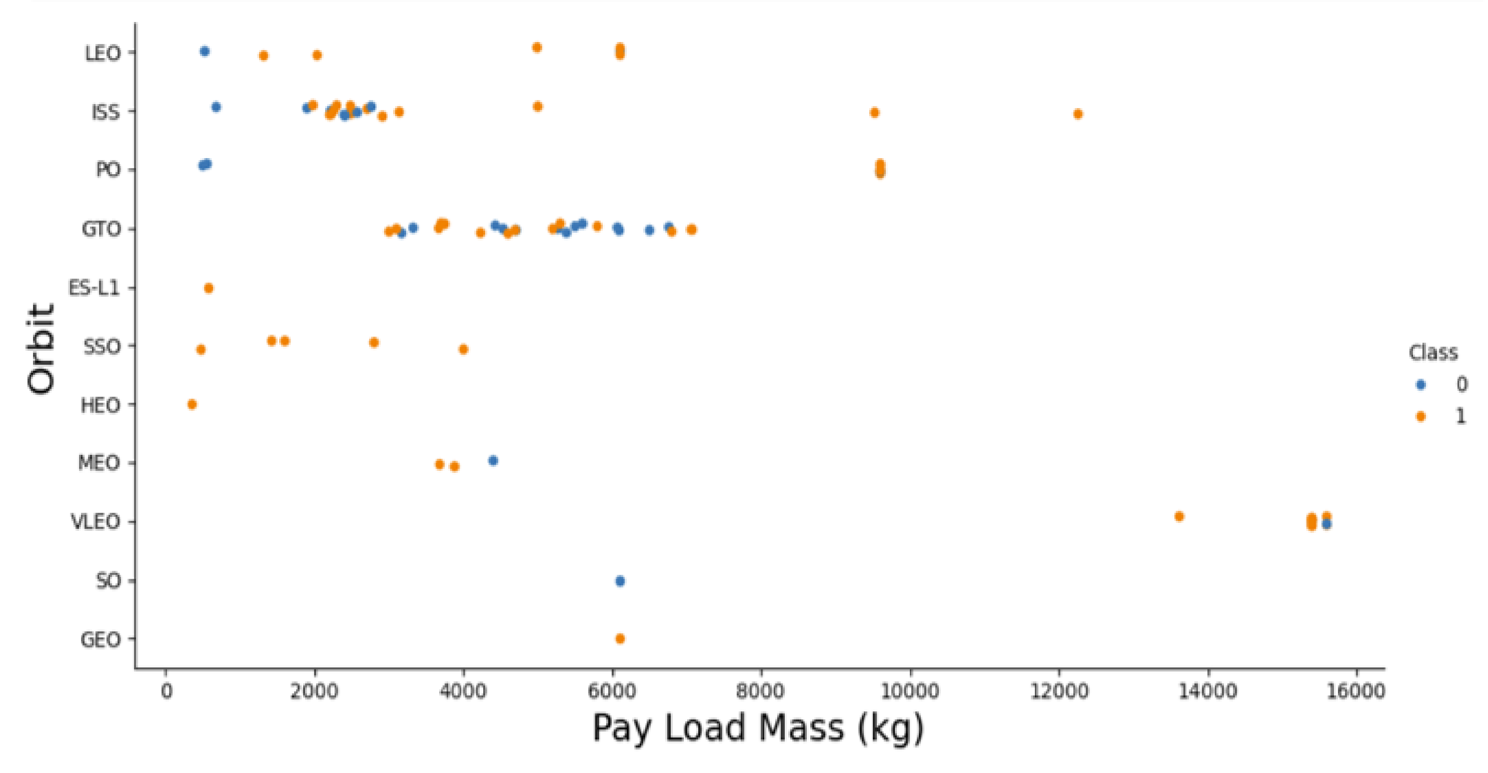


Flight Number vs. Orbit Type



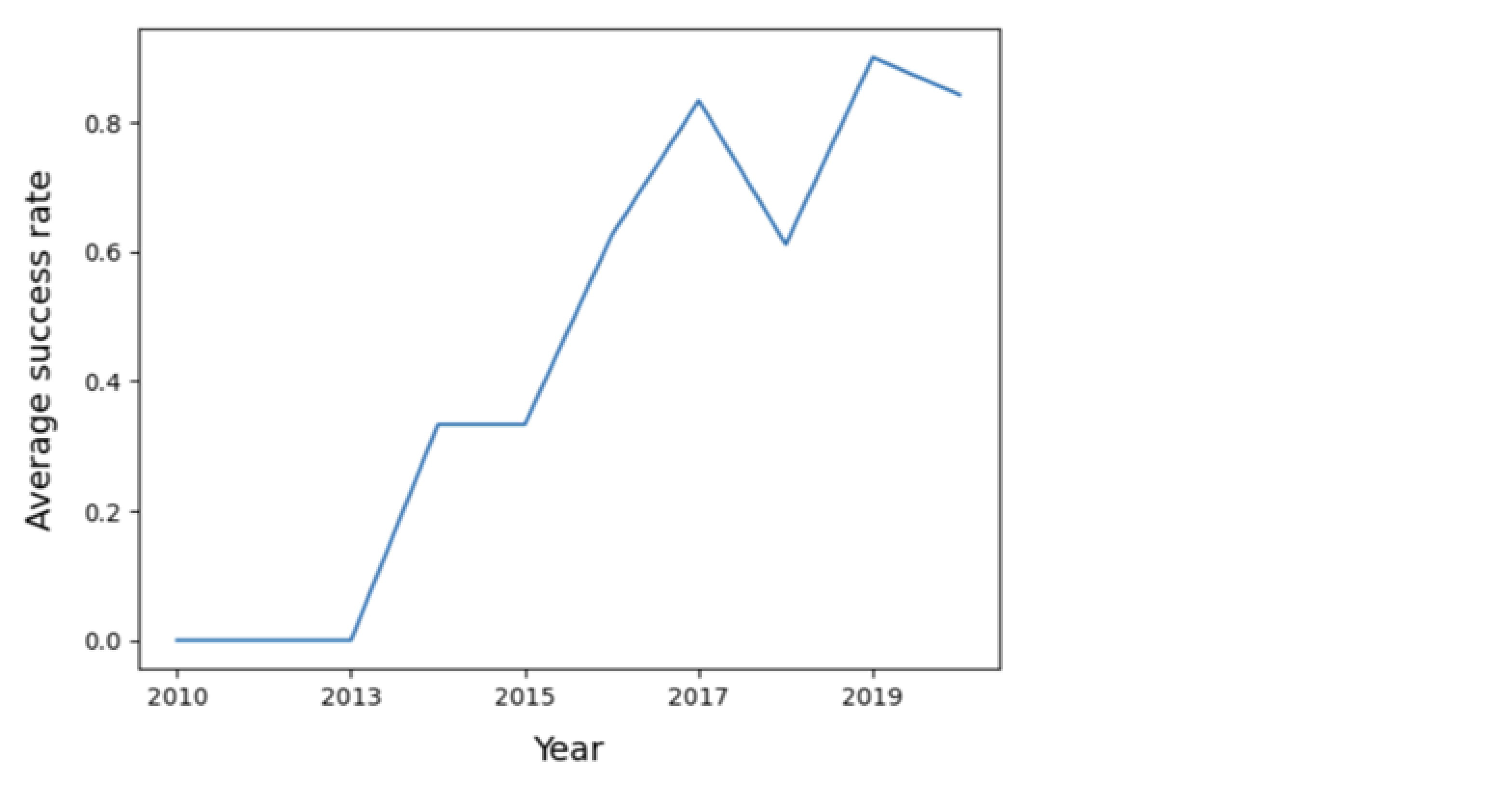
- This figure depicts the relationship between FlightNumber and Orbit type. From the scatter chart, we realized that in the LEO orbit the success appears related to the number of flights. On the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



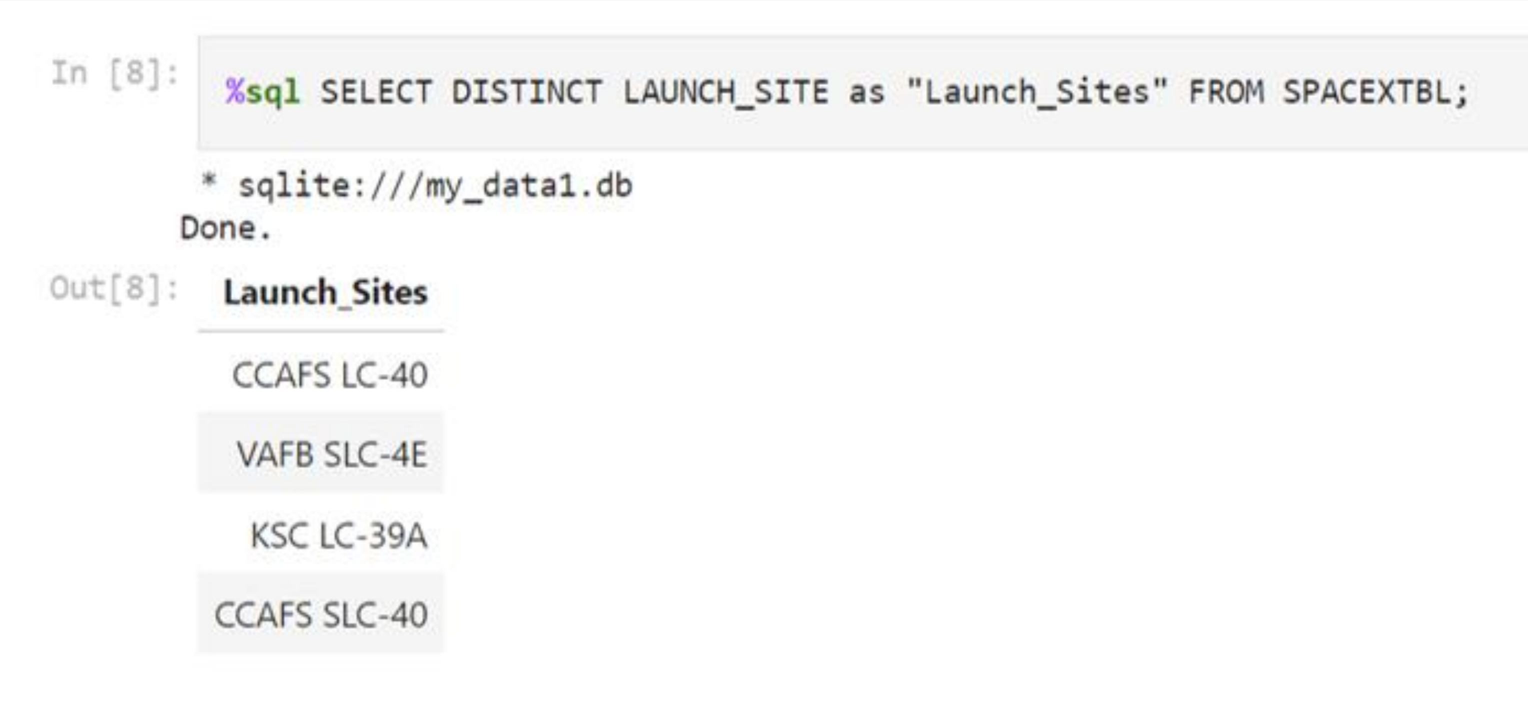
This figure depicts the relationship between Payload and Orbit type. From the scatter chart, we realized that with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.

Launch Success Yearly Trend



- This figure depicts the launch success yearly trend. From the line chart, we realized that the success rate since 2013 continues to increase until 2020.

All Launch Site Names



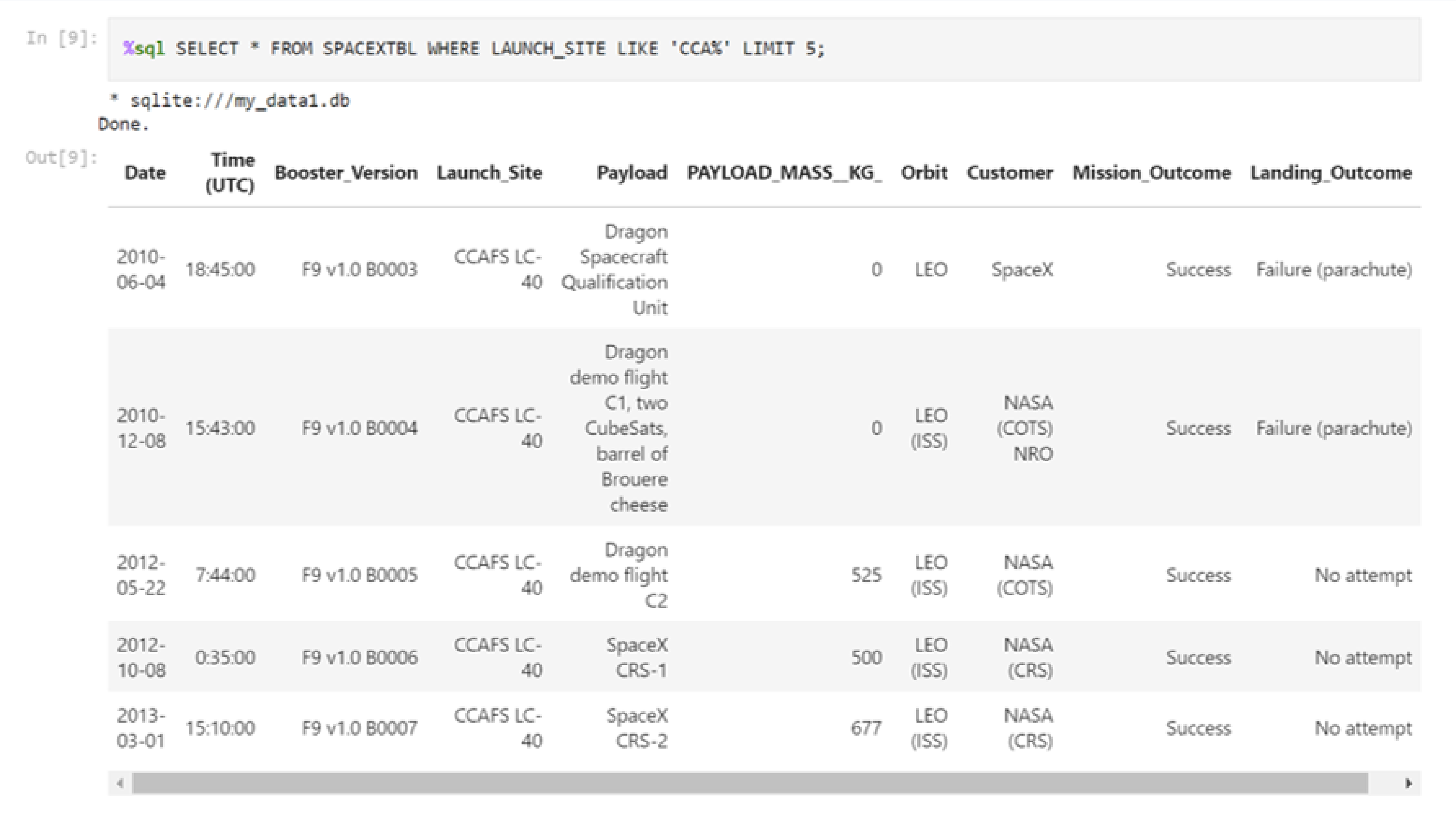
```
In [8]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;

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

Out[8]:

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

- Find the names of the unique launch sites.
- We used DISTINCT to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

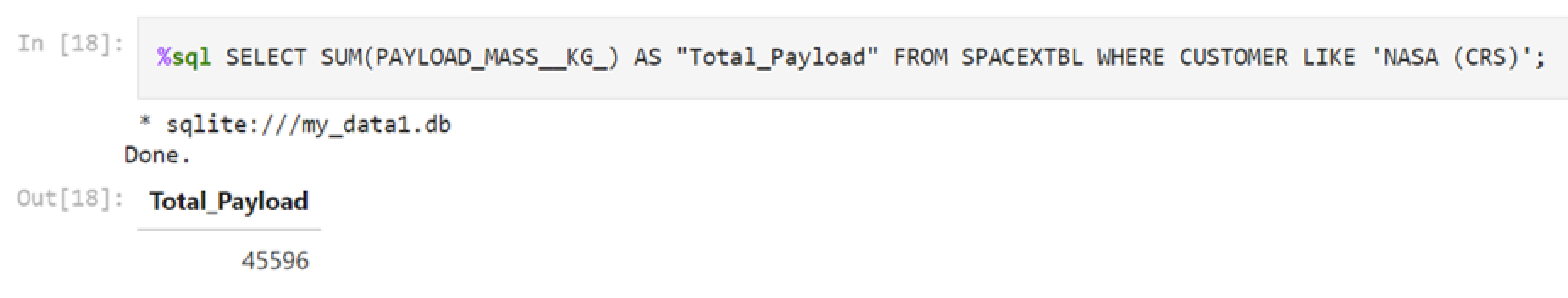


```
In [9]:  %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;

          * sqlite:///my_data1.db
         Done.
Out[9]:
```

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

- Find 5 records where launch sites begin with `CCA`.
- We used the query above to display 5 records where launch sites begin with `CCA`.

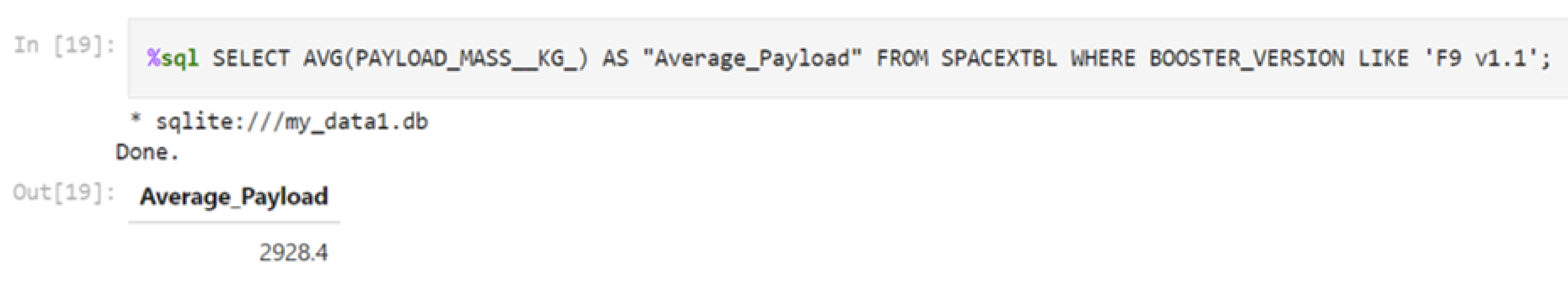
Total Payload Mass



```
In [18]:  %sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total_Payload" FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA (CRS)';

           * sqlite:///my_data1.db
          Done.

Out[18]:  Total_Payload
                  45596
```

- Calculate the total payload carried by boosters from NASA.
- We used the query above to calculate the total payload carried by NASA's boosters as 45596.

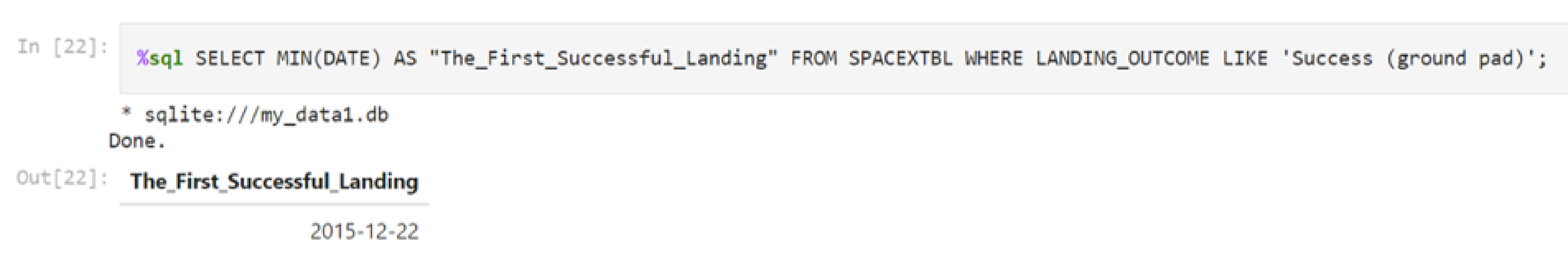
Average Payload Mass by F9 v1.1



```
In [19]:  %sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average_Payload" FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1';

           * sqlite:///my_data1.db
          Done.

Out[19]:   Average_Payload
          ----------------
                    2928.4
```

- Calculate the average payload mass carried by booster version F9 v1.1
- We used the query above to calculate the average payload mass carried by booster version F9 v1.1 as 2928.4.

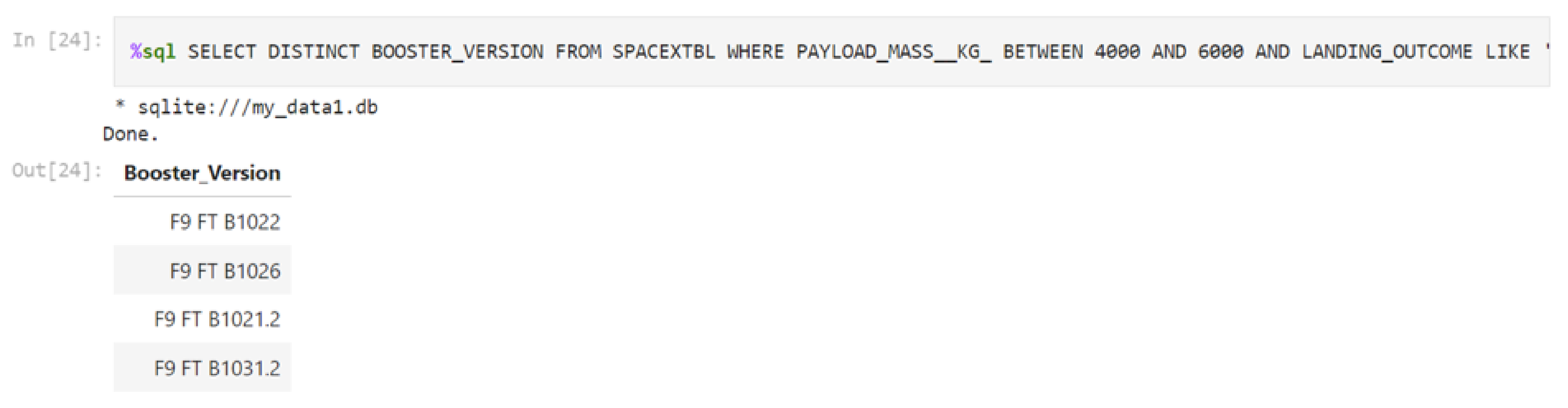
First Successful Ground Landing Date



```
In [22]:  %sql SELECT MIN(DATE) AS "The_First_Successful_Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Success (ground pad)';

           * sqlite:///my_data1.db
          Done.

Out[22]:  The_First_Successful_Landing
                            2015-12-22
```

- Find the dates of the first successful landing outcome on ground pad.
- We used the query above and found the date of the first successful landing outcome on ground pad was 12/22/2015.

Successful Drone Ship Landing with Payload between 4000 and 6000



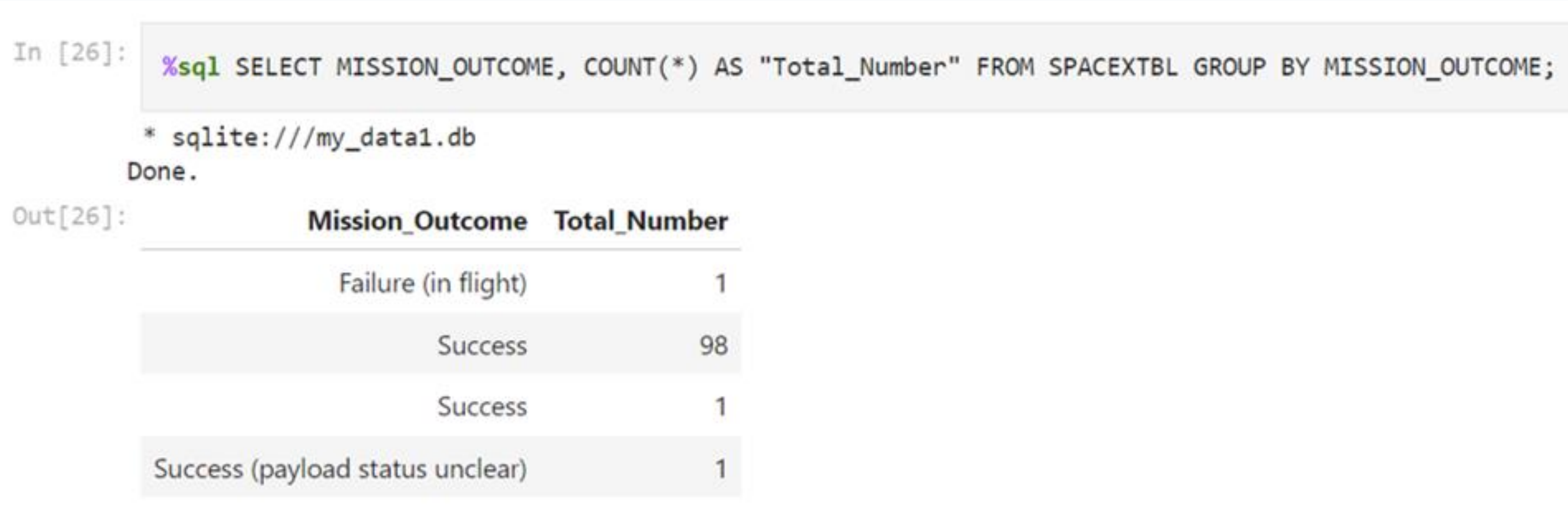
```
In [24]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING_OUTCOME LIKE '

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

Out[24]:

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

- We used WHERE to filter for boosters that successfully landed on the drone ship and applied AND to determine successful landing with payload masses greater than 4000 but less than 6000, and finally obtained the results as shown above.

Total Number of Successful and Failure Mission Outcomes



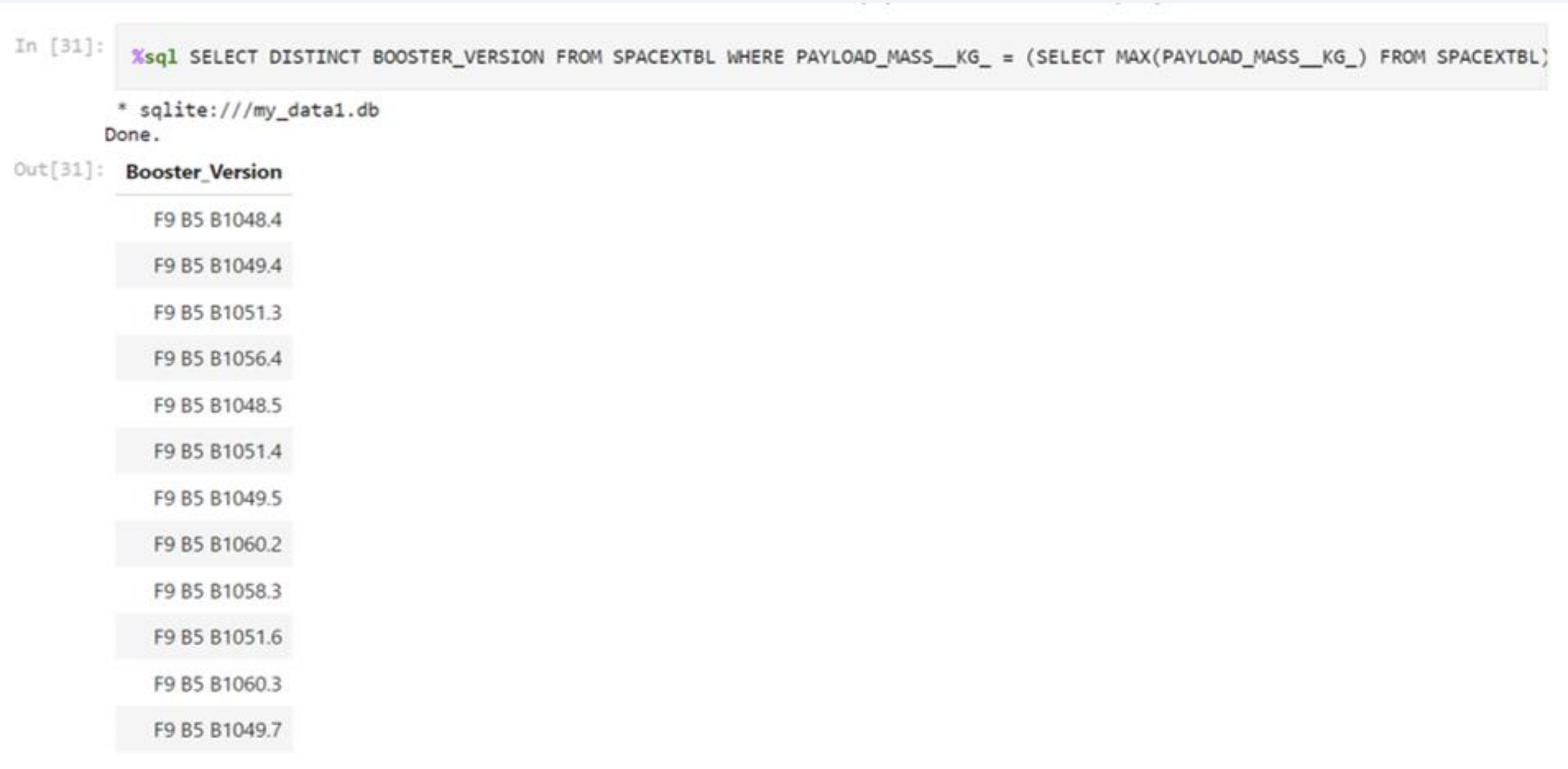
```
In [26]: %sql SELECT MISSION_OUTCOME, COUNT(*) AS "Total_Number" FROM SPACEXTBL GROUP BY MISSION_OUTCOME;

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

Out[26]:

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

- Calculate the total number of successful and failure mission outcomes.
- We used the query above and found success = 100 and failure = 1.

Boosters Carried Maximum Payload



In [31]:

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

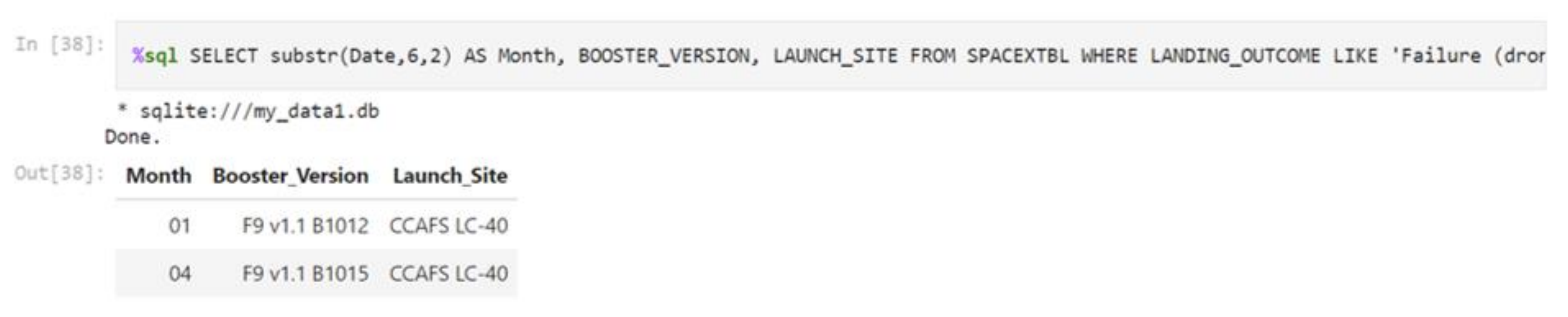
```
 * sqlite:///my_data1.db
Done.
```

Out[31]:

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

- We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass, and finally obtained the results as shown above.

2015 Launch Records



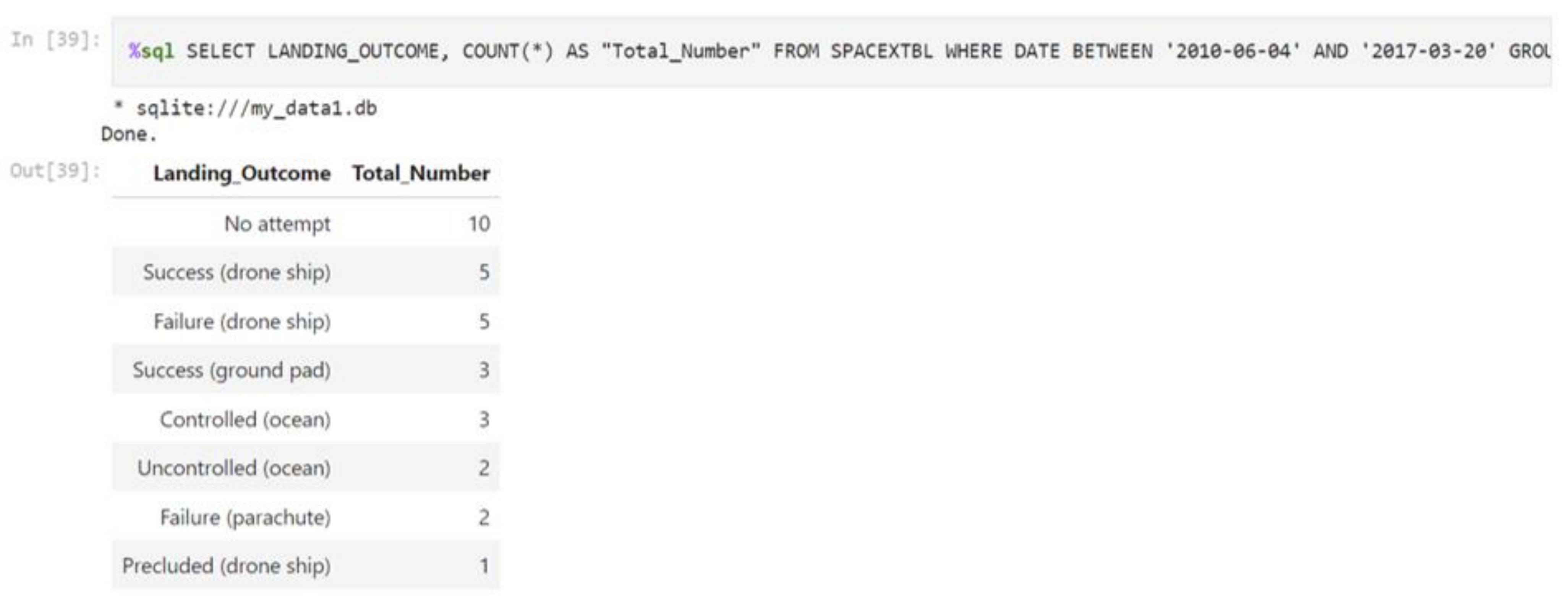
```
In [38]:  %sql SELECT substr(Date,6,2) AS Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Failure (dror

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

Out[38]:

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- We used the query above to return the month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. The substr() function process date in order to take month or year - substr(Date,6,2) shows month, substr(Date,0,5) shows year.

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



In [39]:

```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS "Total_Number" FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROU
```

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

Out[39]:

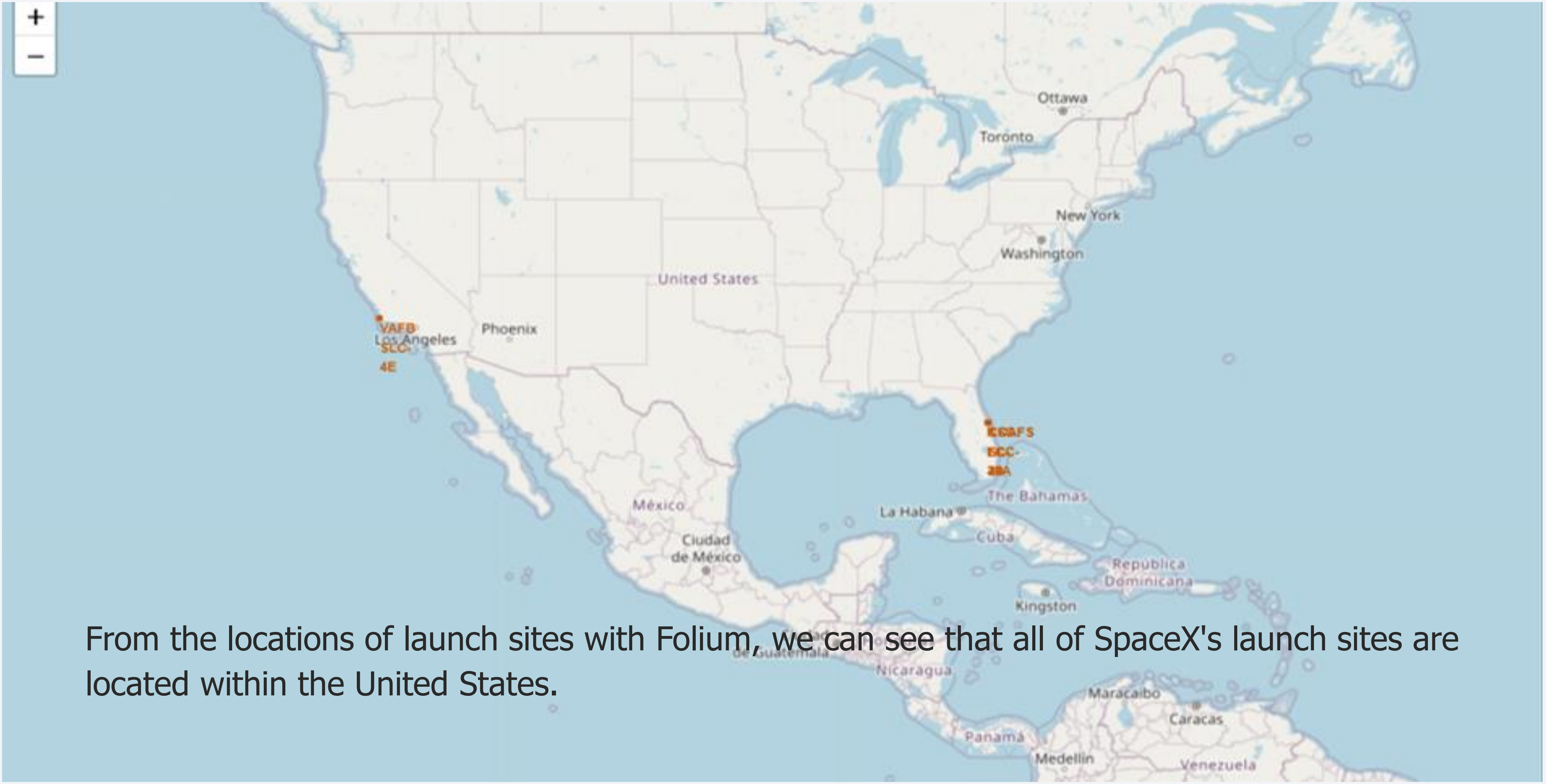
Landing_Outcome	Total_Number
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- We used the query above to return landing outcomes and their count where mission was successful and date is between 2010-06-04 and 2017-03-20. The GROUP BY clause groups results by LANDING_OUTCOME and ORDER BY Total_Number DESC shows results in decreasing order.

Section 3

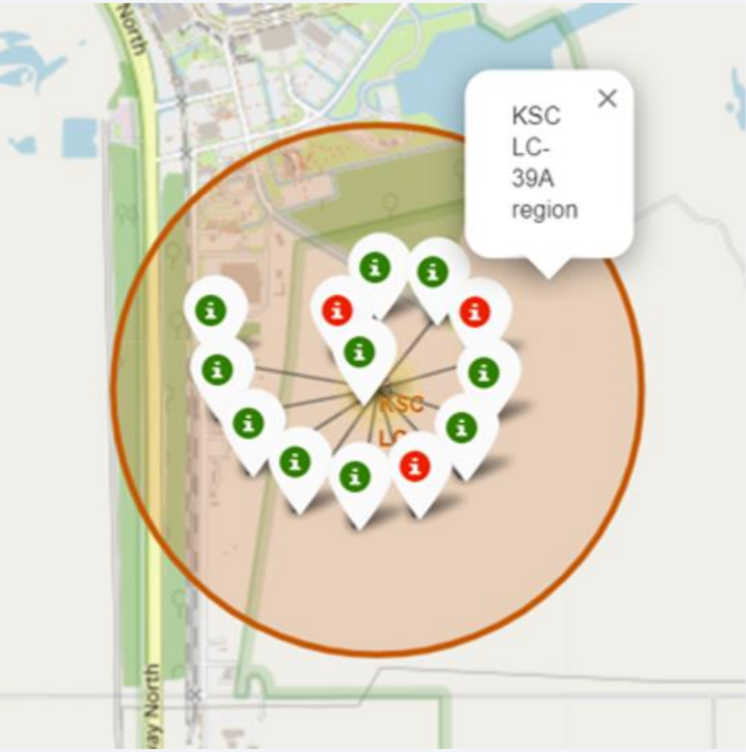
Launch Sites Proximities Analysis

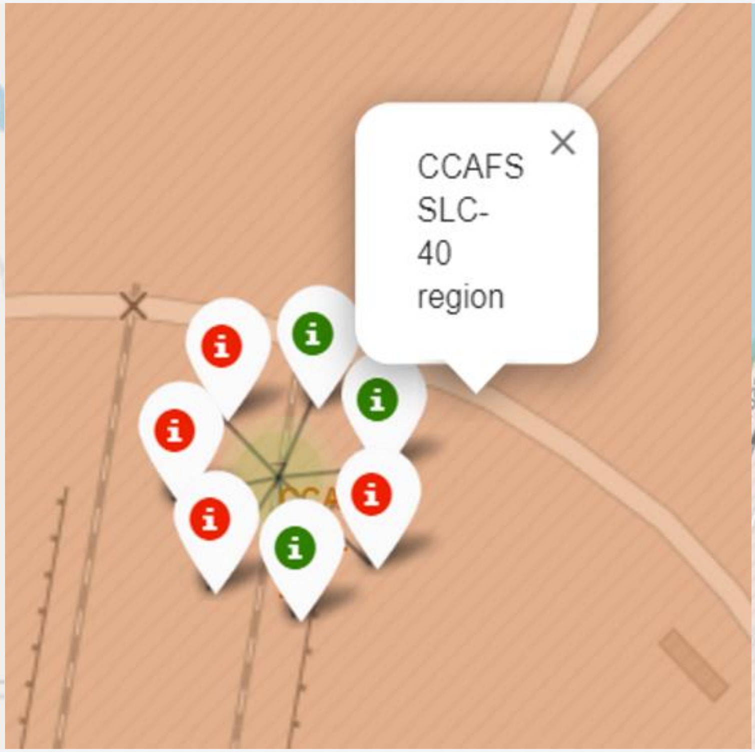
Launch Sites Locations with Folium

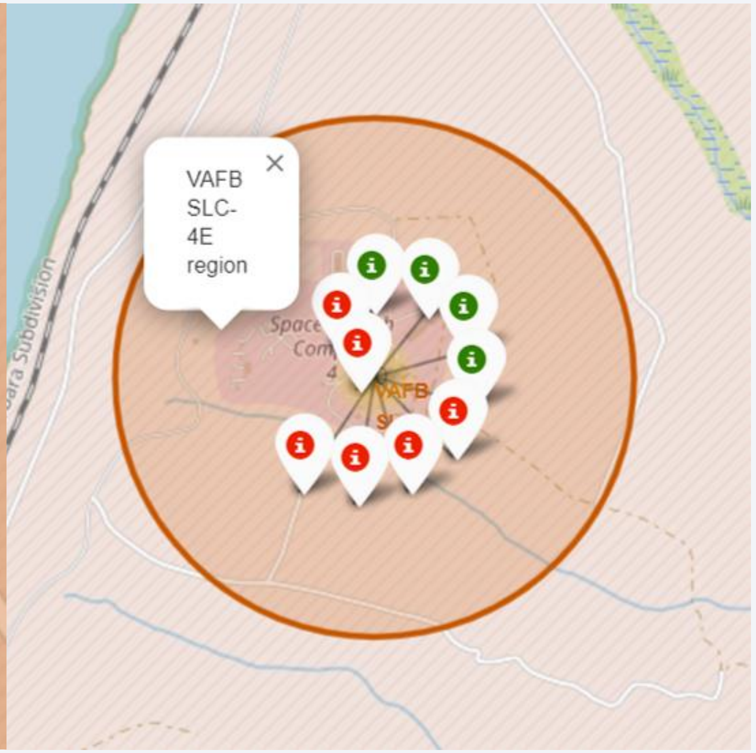


From the locations of launch sites with Folium, we can see that all of SpaceX's launch sites are located within the United States.

Launch Sites vs. Color-Labeled Markers

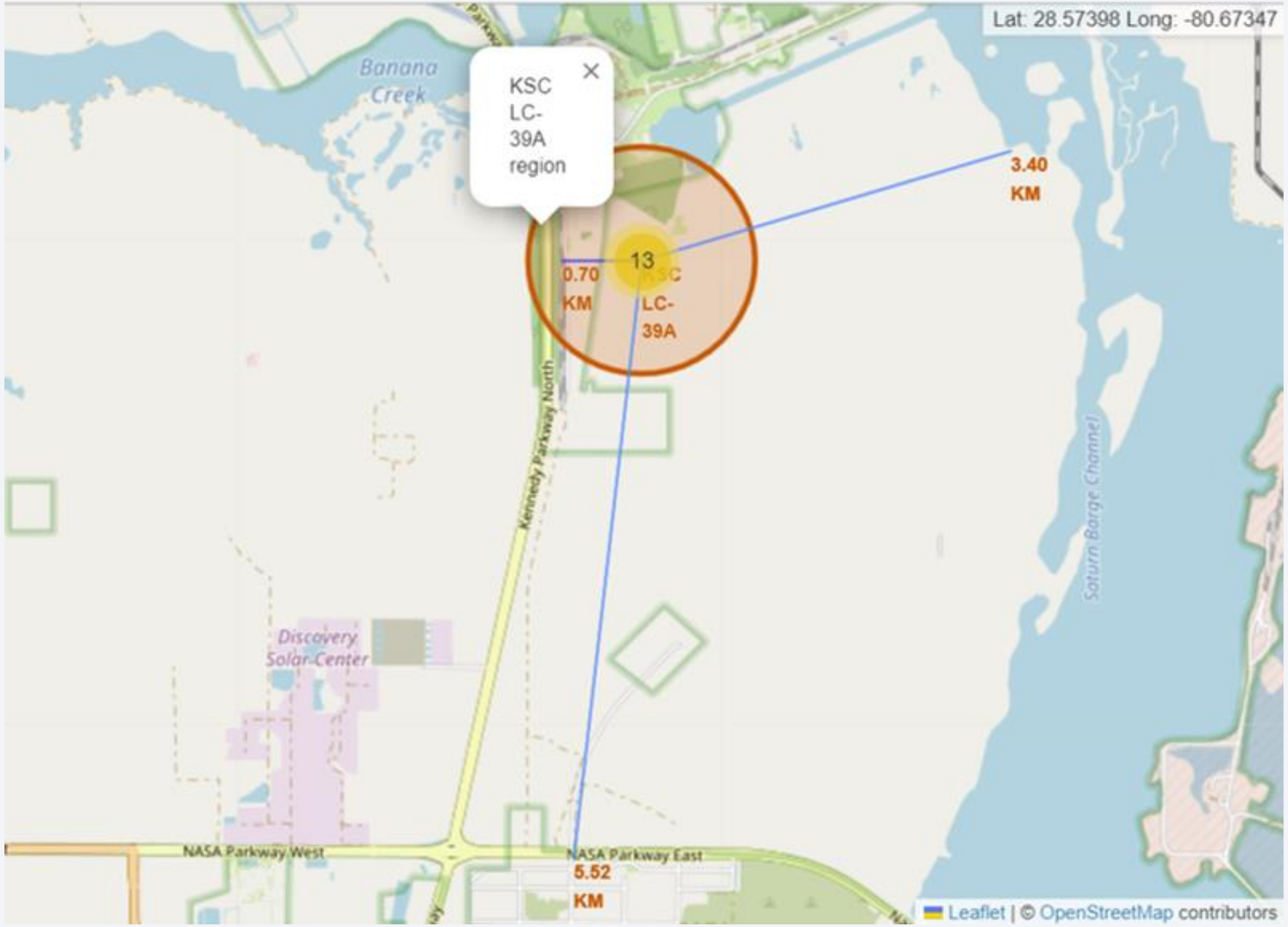






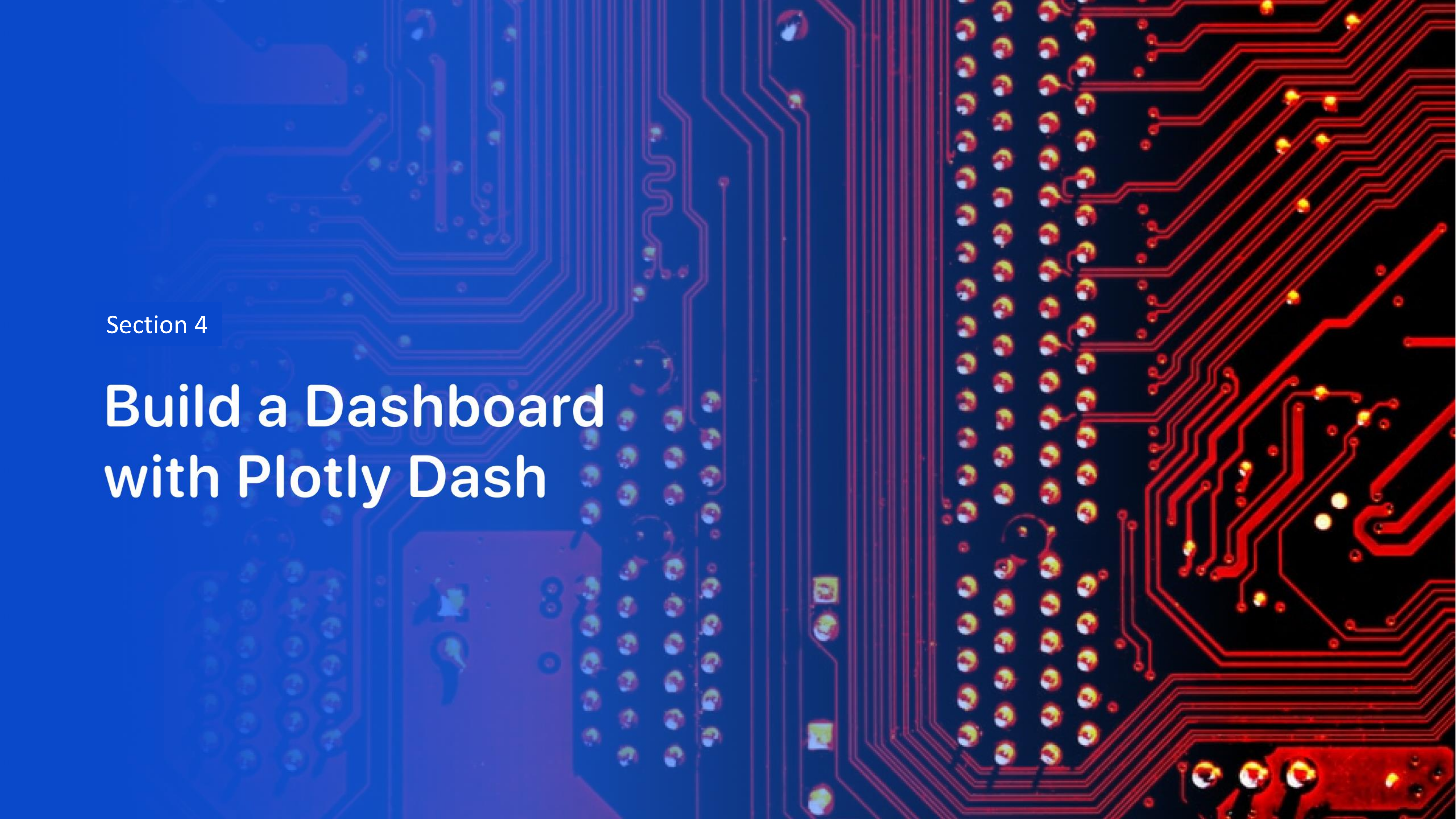
Green marker represents successful launches. Red marker represents unsuccessful launches. From the color-labeled markers in marker clusters, we can easily determine which launch sites have the higher launch success rate. We note that KSC LC-39A is the site with a higher launch success rate.

Launch Sites vs. Proximities



For the launch site selected (KSC LC-39A), we can now answer questions regarding its proximities:

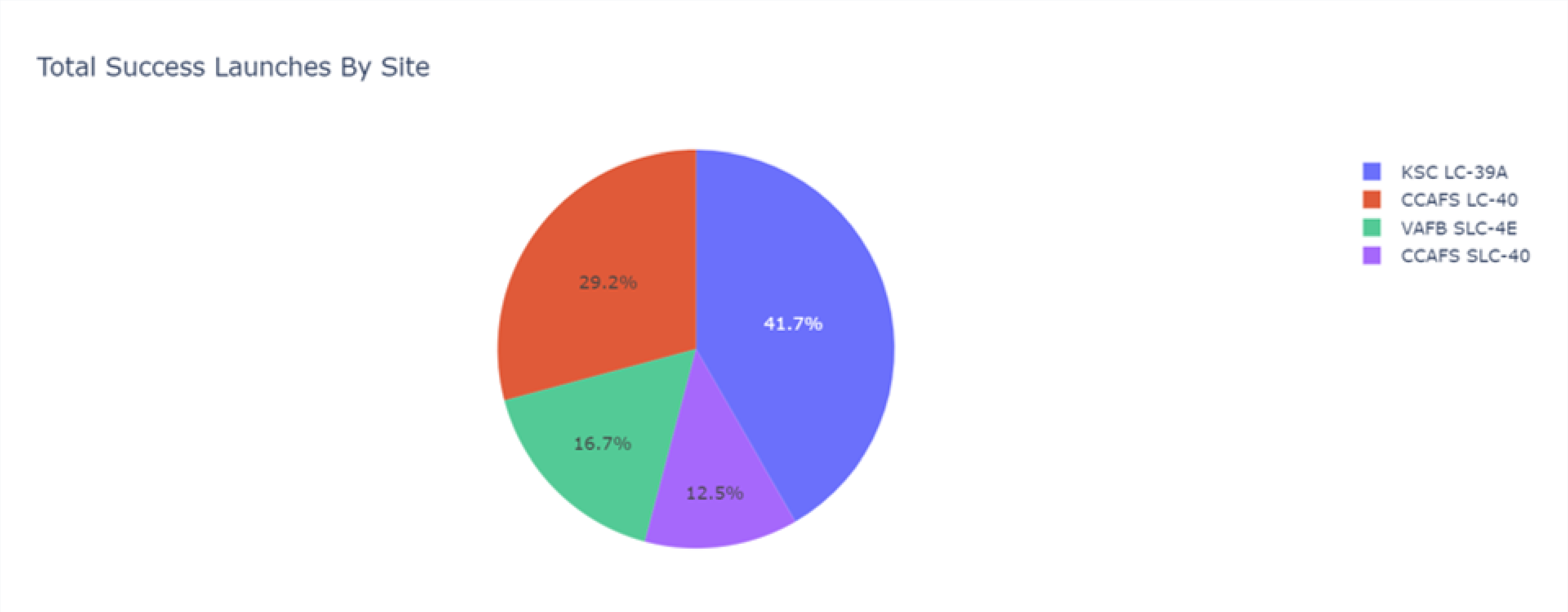
- Is KSC LC-39A in close proximity to railways?
  - YES
- Is KSC LC-39A in close proximity to highways?
  - YES
- Is KSC LC-39A in close proximity to coastline?
  - YES



Section 4

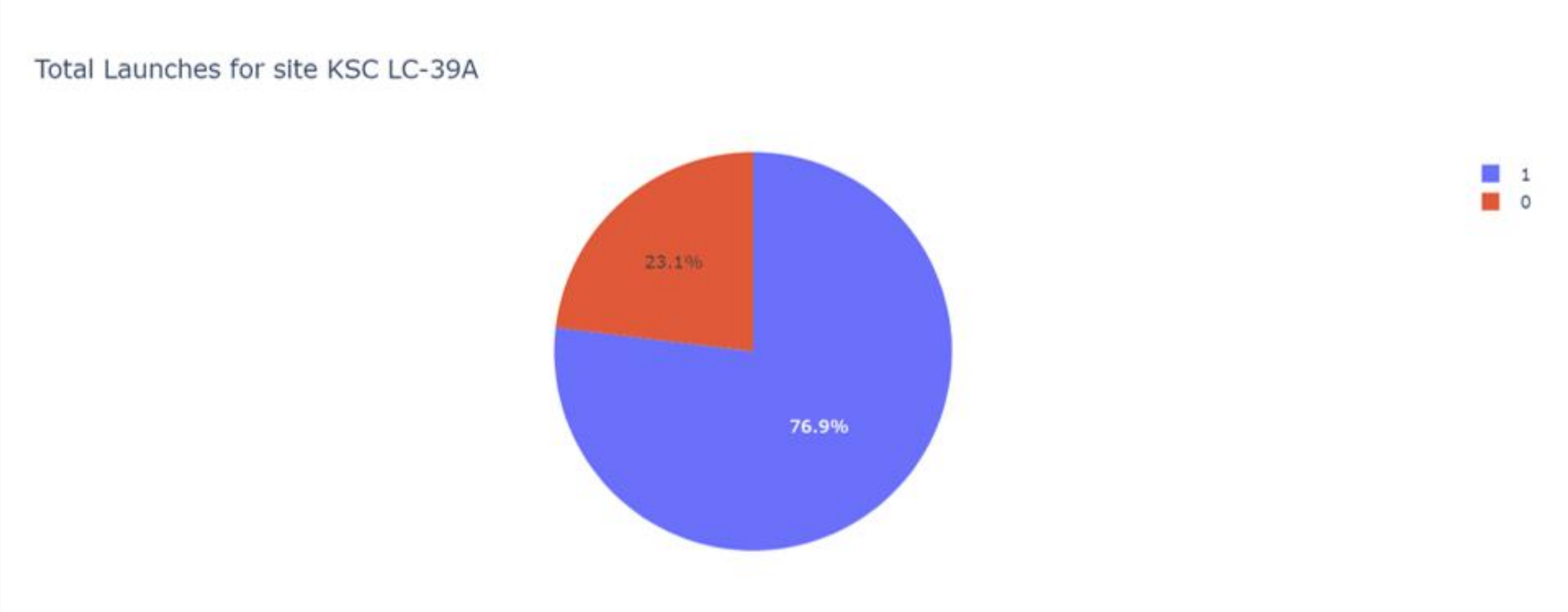
Build a Dashboard with Plotly Dash

Total Success Launches By all Sites



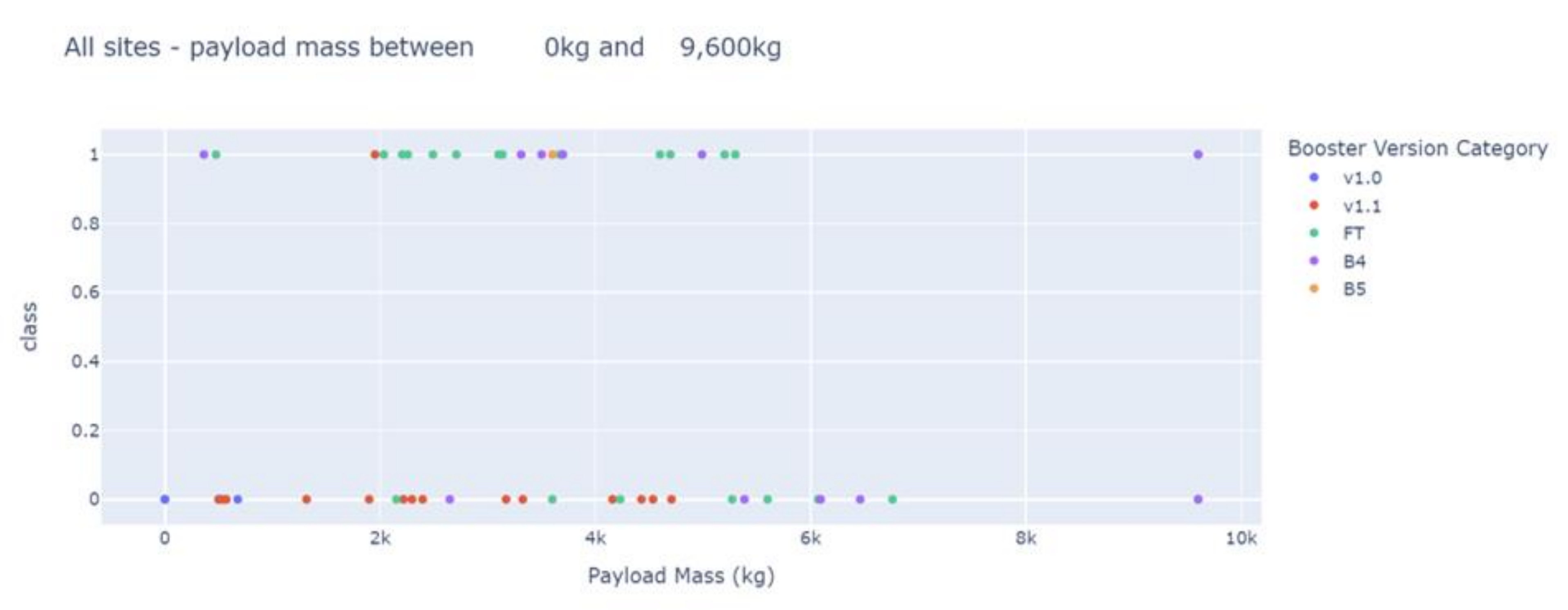
- From the pie chart, we can see that KSC LC-39A had the largest number of successful launches from all sites.

KSC LC-39A vs. Total Launches



- KSC LC-39A is the launch site with the highest successful launch rate of all sites (success = 76.9%, failure = 23.1%).

Payload vs. Launch Outcome By all Sites

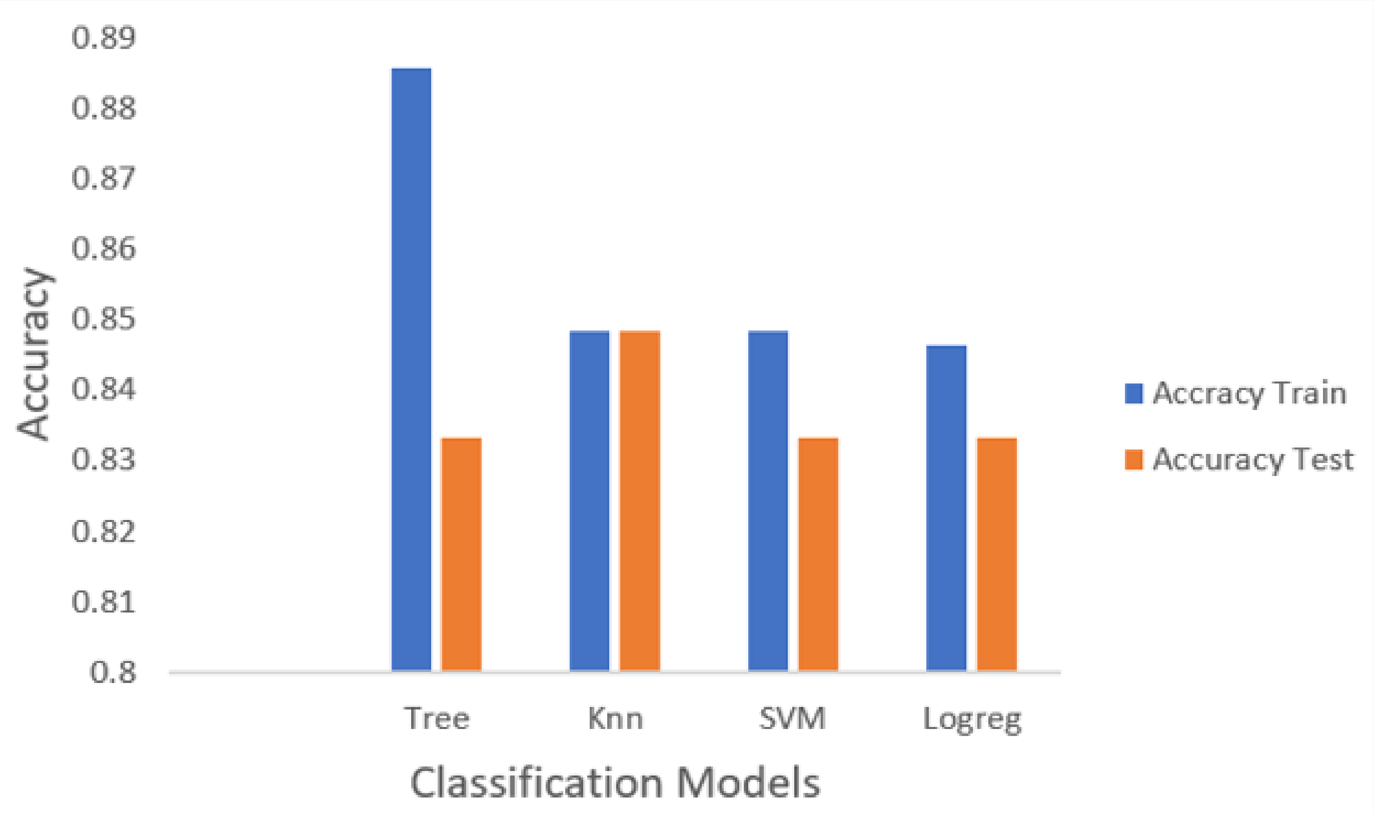


- From the scatter chart, we can see that the low payload range (0 - 5000 kg) has a higher launch success rate than the high payload range (5000 - 10000 kg).

Section 5

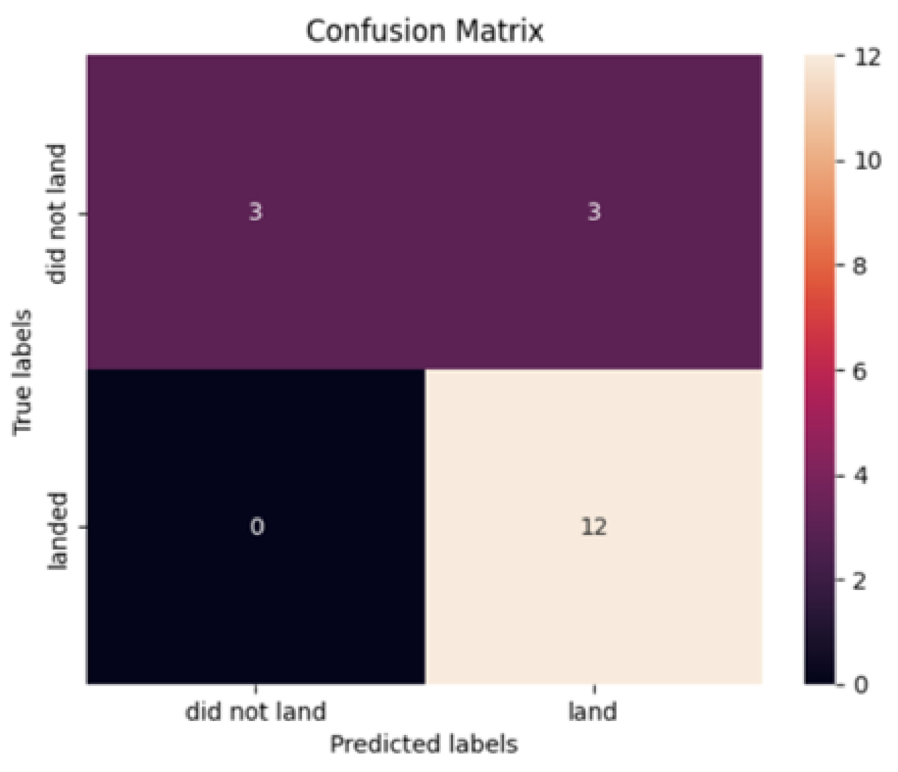
Predictive Analysis (Classification)

Classification Accuracy



- There have been 4 classification models built and tested: SVM, Logistic Regression, Decision Tree, and KNN.
- The accuracy of the 4 models is shown in the screenshot on the left.
- The model with the highest classification accuracy is Decision Tree, which has an accuracy of over 88%.

Confusion Matrix



- In the confusion matrix of the decision tree classifier, the true positive and true negative values are much larger than the false positive and false negative values. This is consistent with the accuracy of the decision tree classification model as presented above.

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally the low payload range (0 - 5000 kg) perform better than the high payload range (5000 - 10000 kg).
- The orbits with higher success rates are GEO, HEO, SSO, ES-L1.
- With current data, we cannot explain why some launch sites have better launch success rates than others. However, it is reality and KSC LC-39A is the best launch site with a successful launch rate of 76.9%.
- Finally, for this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the used models is almost the same. We choose Decision Tree Algorithm because it has a better train accuracy (over 88%).

Appendix

- Any relevant content such as Python code snippets, SQL queries, charts, Notebook outputs or datasets, etc. you can see in my Github Url: https://github.com/mrthanhvu/testrepo

Thank you!