IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Timothy Martin
March 4, 2024

timothymartin456@gmail.com

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through API
  - Data collection with web scraping
  - Data wrangling
  - Exploratory Data Analysis (EDA) using SQL
  - EDA using data visualization
  - Interactive Visual Analytics using Folium
  - Predictive analytics using machine learning
- Summary of all results
  - EDA results
  - Screenshots of interactive analytics
  - Predictive analytics results

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 the launch.

  The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully so that this information can be utilized by Space Y to bid against Space X and break into the market.

- Problems you want to find answers
  - What factors help determine if the first stage will land successfully?
  - Interactions between features which help predict the success rate of a landing.
  - Operating conditions which provide a better chance of successful landings.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data for this analysis was obtained through web scraping Wikipedia and from the SpaceX API
- Perform data wrangling
  - One-hot encoding was used on categorical features for easier analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, SVM, Decision Tree, and KNN models were evaluated
  - Hyperparameters were tuned using Gridsearch cross-validation
  - Score method was used to evaluate accuracy and a confusion matrix was utilized

Data Collection

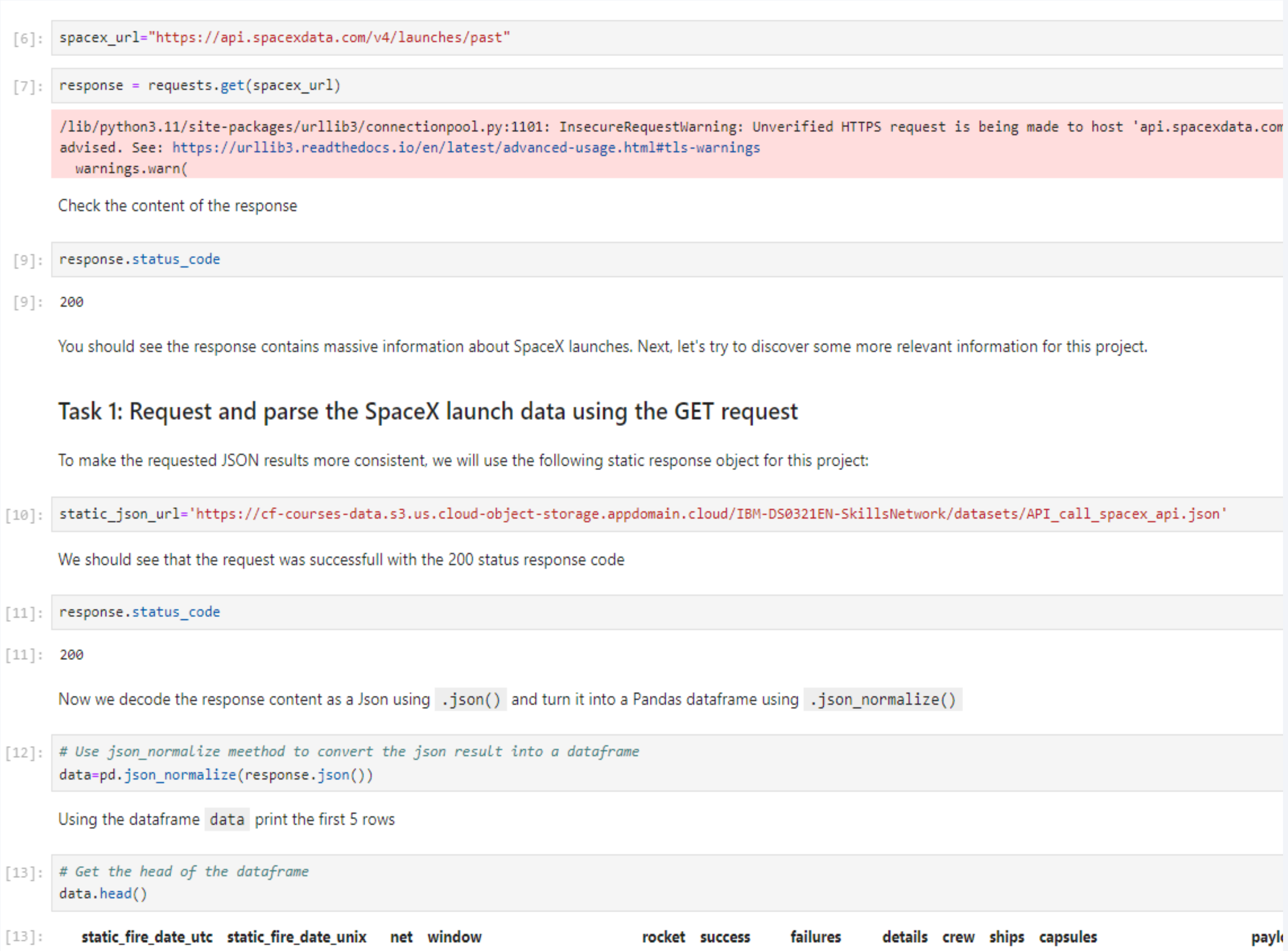
Data was collected from the following locations utilizing the described methods

- SpaceX API
    - ❖Data collection was done using Get request to SpaceX API
    - ❖Data was then decoded as a json using .json() and turned into a dataframe using .json_normalize()
    - ❖Data was then cleaned, and missing values were identified and filled in where necessary.
- Wikipedia
    - ❖Falcon 9 records were collected by web scraping using a HTTP GET request
    - ❖The data was then parsed and stored in a dataframe using BeautifulSoup

Data Collection – SpaceX API

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

- GitHub Link: [tools-for-data/week 1 lab1-data-collection-api.ipynb at main · timothymartin456/tools-for-data (github.com)]()



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

```
[7]: response = requests.get(spacex_url)
```

```
/lib/python3.11/site-packages/urllib3/connectionpool.py:1101: InsecureRequestWarning: Unverified HTTPS request is being made to host 'api.spacexdata.com
advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#tls-warnings
  warnings.warn(
```

Check the content of the response

```
[9]: response.status_code
```

```
[9]: 200
```

You should see the response contains massive information about SpaceX launches. Next, let's try to discover some more relevant information for this project.

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:

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

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

```
[11]: response.status_code
```

```
[11]: 200
```

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

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

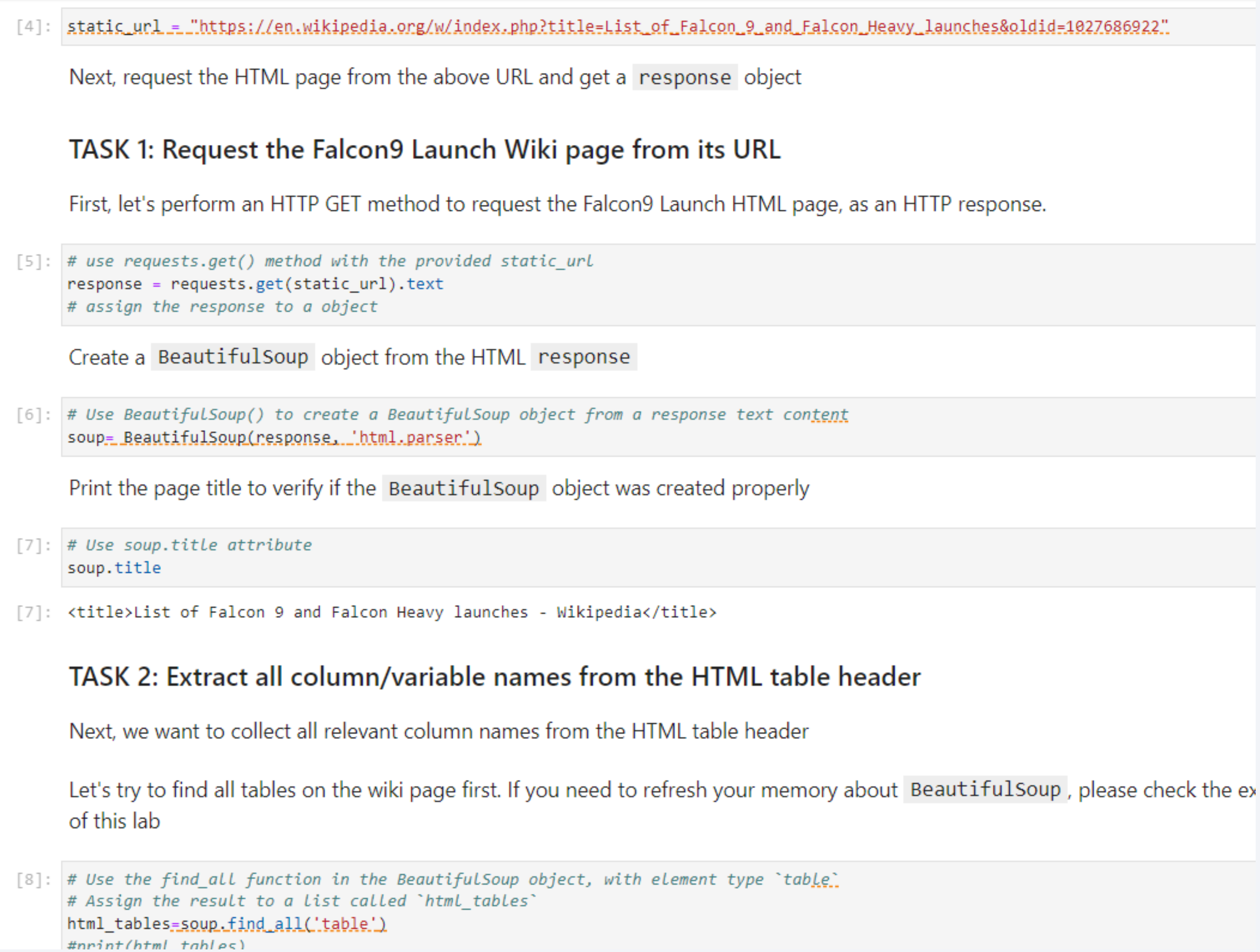
Using the dataframe `data` print the first 5 rows

```
[13]: # Get the head of the dataframe
      data.head()
```

```
[13]:   static_fire_date_utc  static_fire_date_unix  net  window  rocket  success  failures  details  crew  ships  capsules  payl
```

Data Collection - Scraping

- Http GET request was sent to Falcon 9 launch HTML page
- Data was parsed and stored int a pandas dataframe usin BeautifulSoup
- GitHub link: [tools-for-data/Week1 lab 2 data scraping.ipynb at main · timothymartin456/tools-for-data (github.com)]()



```
[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
     response = requests.get(static_url).text
     # assign the response to a object
```

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

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

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

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

[7]: <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

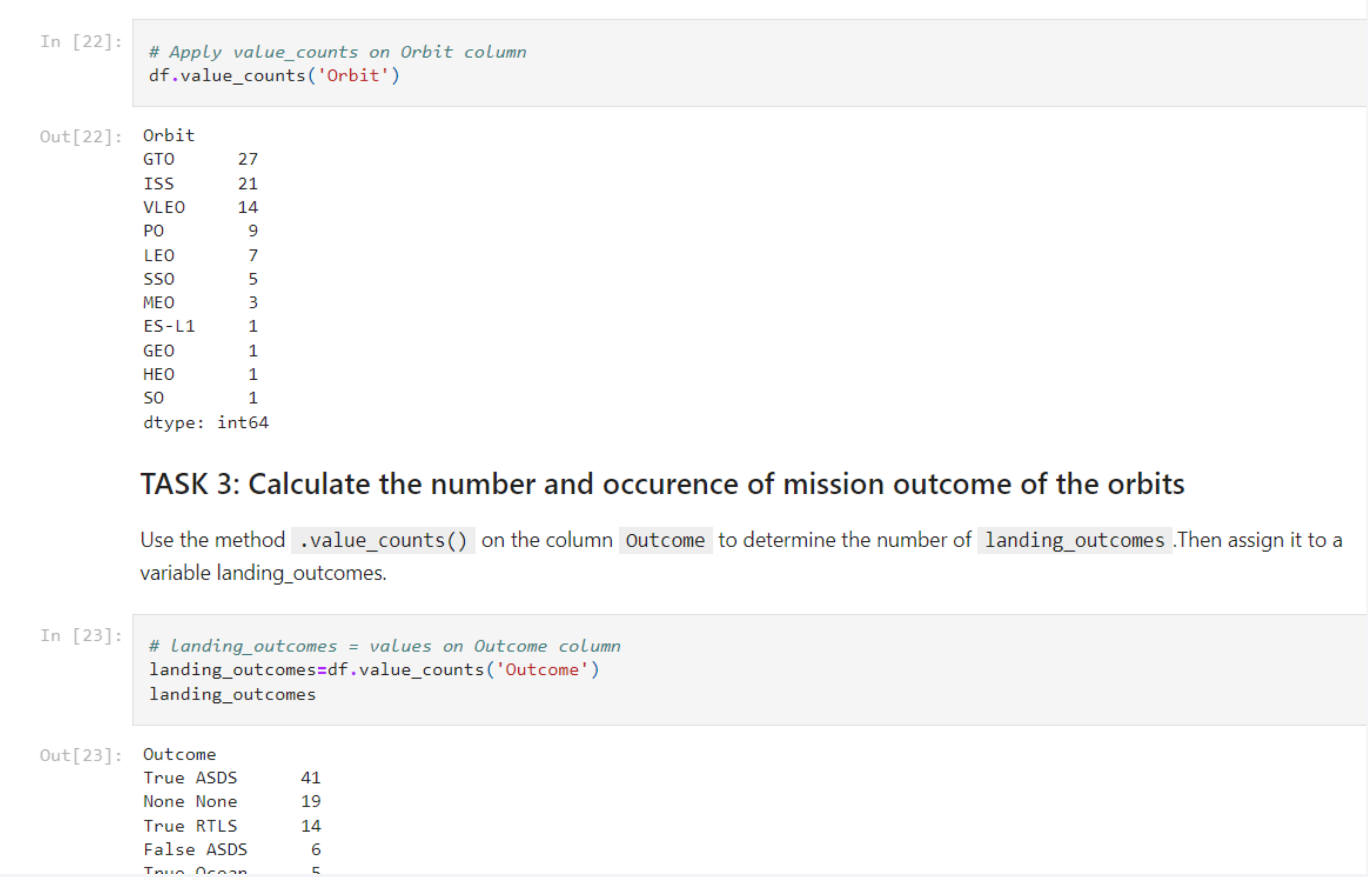
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the ex of this lab

```
[8]: # 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')
     #print(html_tables)
```

Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches on each site, the number and occurrence of each orbit, and the landing outcomes organized by type and frequency.
- Created a landing outcome label from the outcome column and exported the results to a csv.

- GitHub link: [tools-for-data/Week 1 lab 3 data wrangling.ipynb at main · timothymartin456/tools-for-data (github.com)]



```
In [22]:  # Apply value_counts on Orbit column
          df.value_counts('Orbit')
```

```
Out[22]:  Orbit
          GTO      27
          ISS      21
          VLEO     14
          PO        9
          LEO       7
          SSO       5
          MEO       3
          ES-L1     1
          GEO       1
          HEO       1
          SO        1
          dtype: int64
```

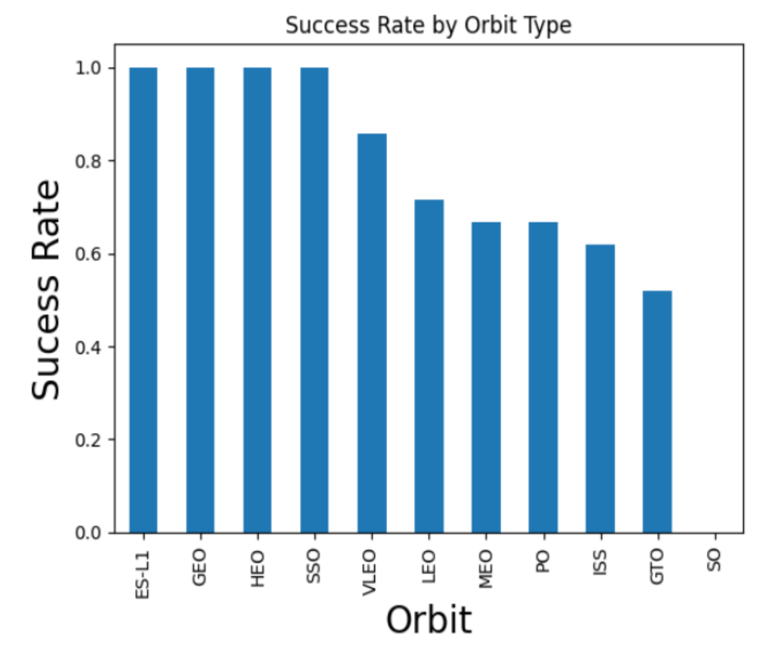
TASK 3: Calculate the number and occurence of mission outcome of the orbits

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

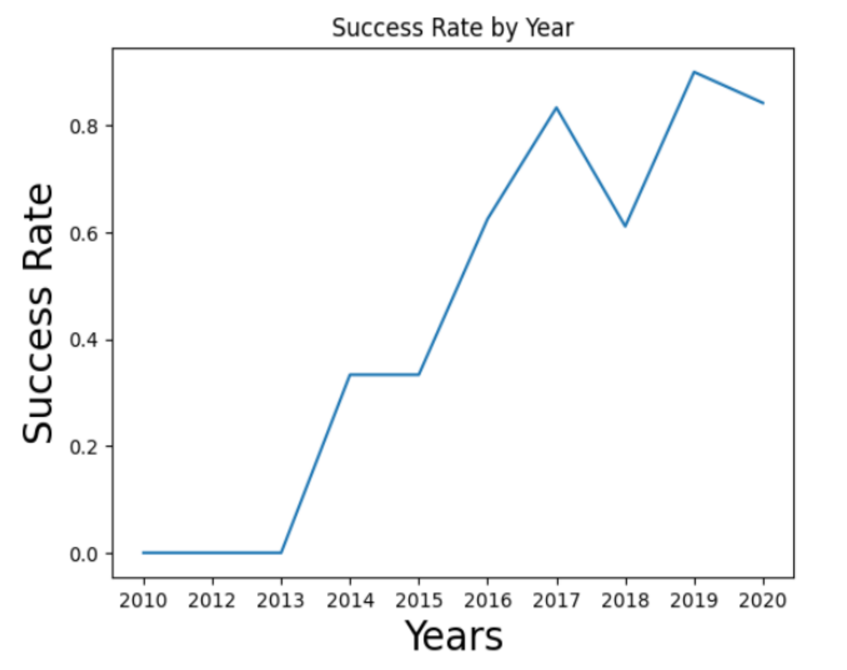
```
In [23]:  # landing_outcomes = values on Outcome column
          landing_outcomes=df.value_counts('Outcome')
          landing_outcomes
```

```
Out[23]:  Outcome
          True ASDS      41
          None None      19
          True RTLS      14
          False ASDS      6
          True Ocean      5
```

EDA with Data Visualization



We visualized the relationship between flight number and launch site, payload and launch site, success rate and orbit, flight number and orbit type, payload mass and orbit, and the yearly trends for successful launches.



GitHub link: tools-for-data/week 2 lab 2 eda with visualization.ipynb at main · timothymartin456/tools-for-data (github.com)

EDA with SQL

- Using Jupyter Notebook the SpaceX dataset was loaded into PostgreSQL and the following queries were performed.
    - The names of unique launch sites
    - 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 successes and failures in the mission outcomes
    - The failed landing outcomes in drone ship, their booster versions and launch site names

- GitHub link: tools-for-data/Week 2 sql.ipynb at main · timothymartin456/tools-for-data (github.com)

Build an Interactive Map with Folium

- Marked all launch sites and added map objects such as markers, lines, and circles to mark the successes and failures of each launch by location on the folium map
- Assigned the feature launch outcomes to class 0 for failures and 1 for successes
- Identified and labeled success rates of launch sites using color-labeled marker clusters
- Calculated the distances between the launch site and important features such as railways, highways, and cities to identify trends.

- GitHub Link: tools-for-data/Week 3 lab 1 folium.ipynb at main · timothymartin456/tools-for-data (github.com)

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Created pie charts showing the total launches by certain sites
- Plotted scatter plots showing the relationship between outcomes and payloads (Kg) for sites chosen by user

- GitHub link: tools-for-data/week 3 ploty spacex dash app.py at main · timothymartin456/tools-for-data (github.com)

Predictive Analysis (Classification)

- Split data into 2 sets using 80% of the data to train the model and 20% to test
- Used different machine learning models and tuned various hyperparameters using GridSearchCV
- With accuracy as the metric; improved the model using feature tuning and hyperparameter tuning
- Found the most accurate classification model out of the 4 tested

- GitHub link: tools-for-data/Week 4 machine learning.ipynb at main · timothymartin456/tools-for-data (github.com)

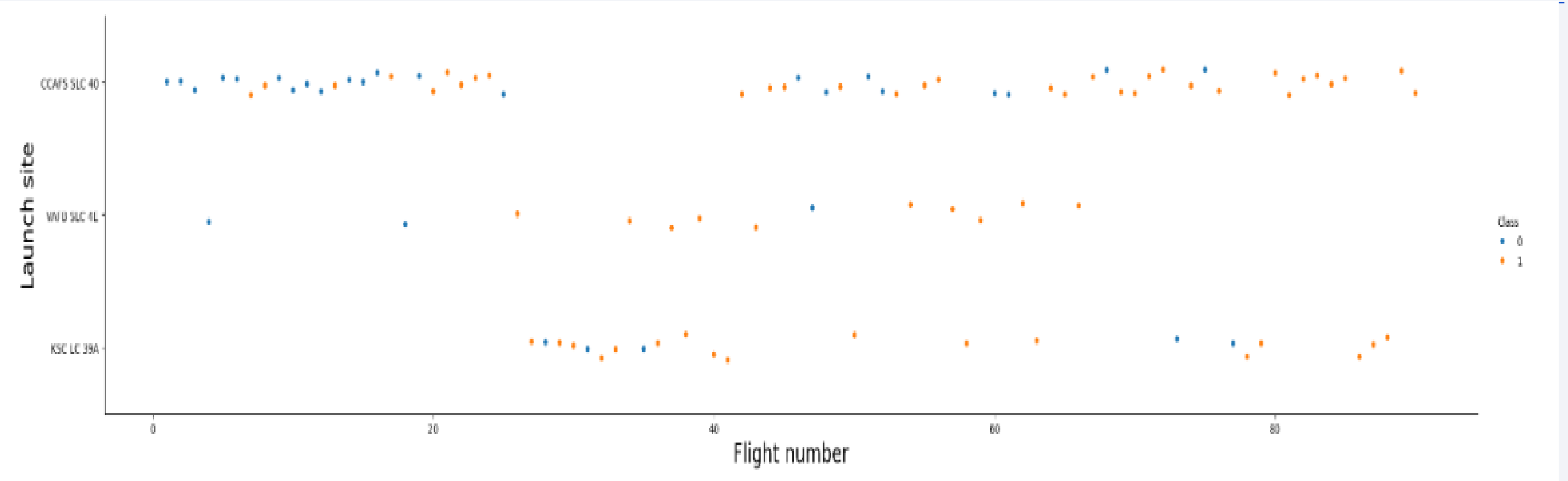
Results

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

Section 2

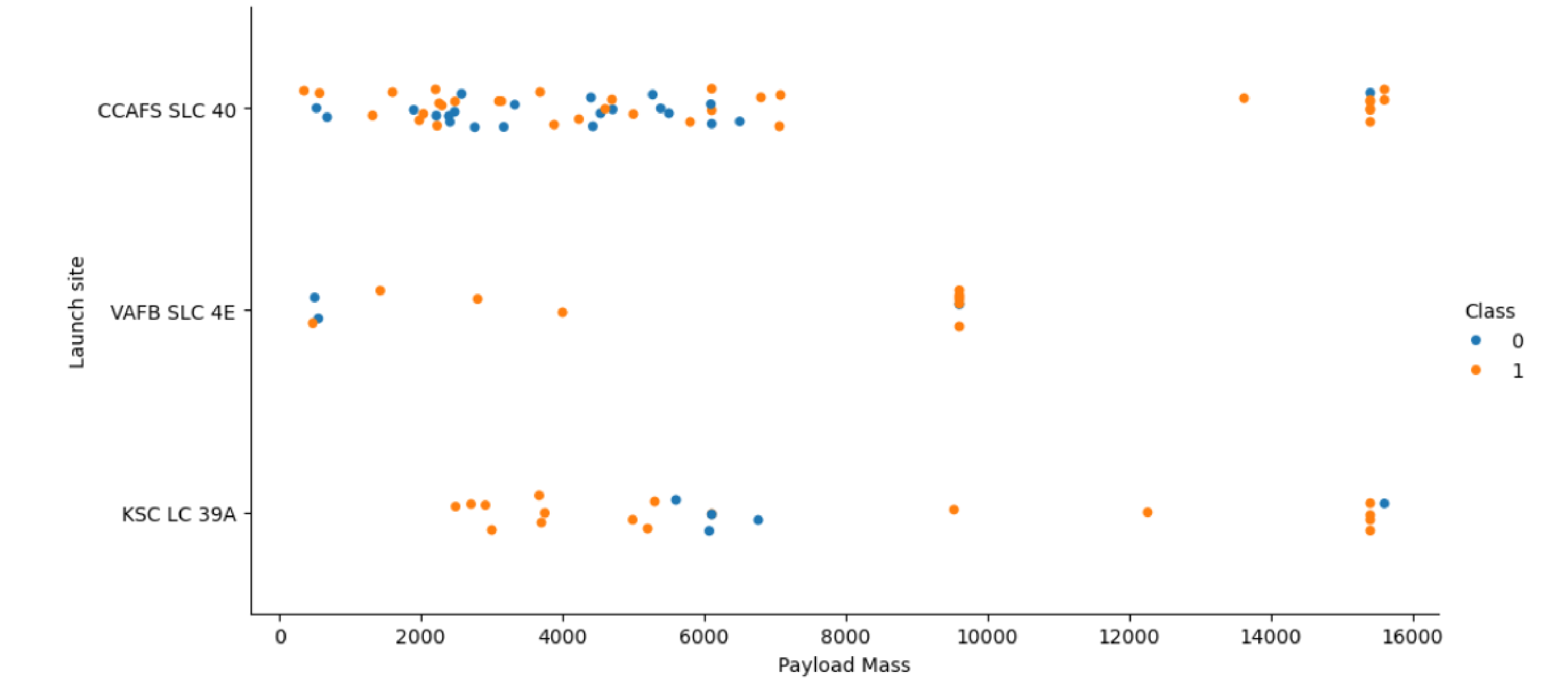
Insights drawn from EDA

Flight Number vs. Launch Site



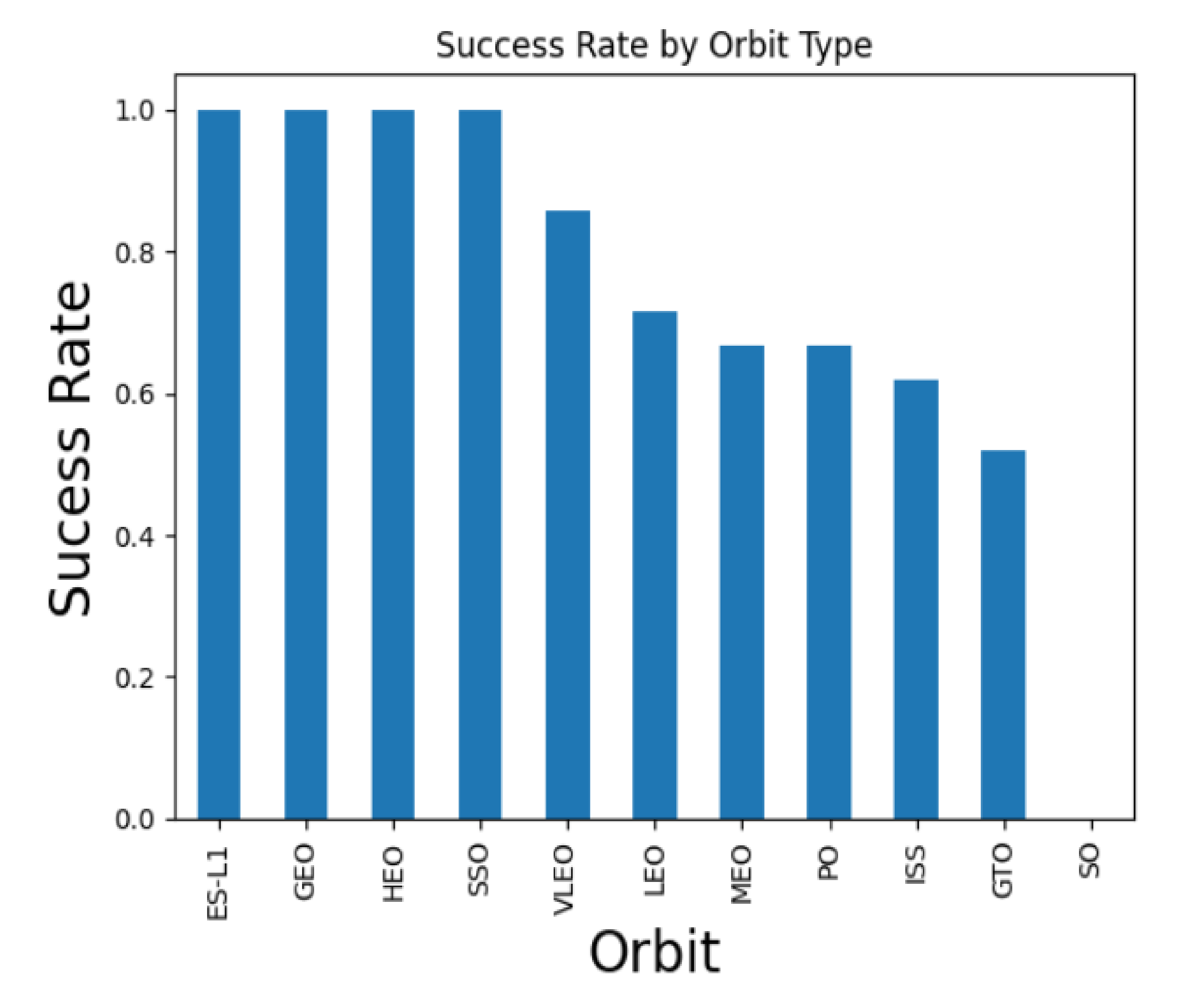
- Observe from the plot, as a site increases the number of launches they attempt, there is an increase in the number of successes. This would imply more launches helps improve the odds the next launch will be a success.

Payload vs. Launch Site



- Launch site CCAFS SLC 40 Seems to have greater success as payloads increase
- Launch site VAFB SLC 4E only launches payloads less than 10,000 kg
- Launch site KSC LC 39A doesn't launch rockets below 2,000 kg

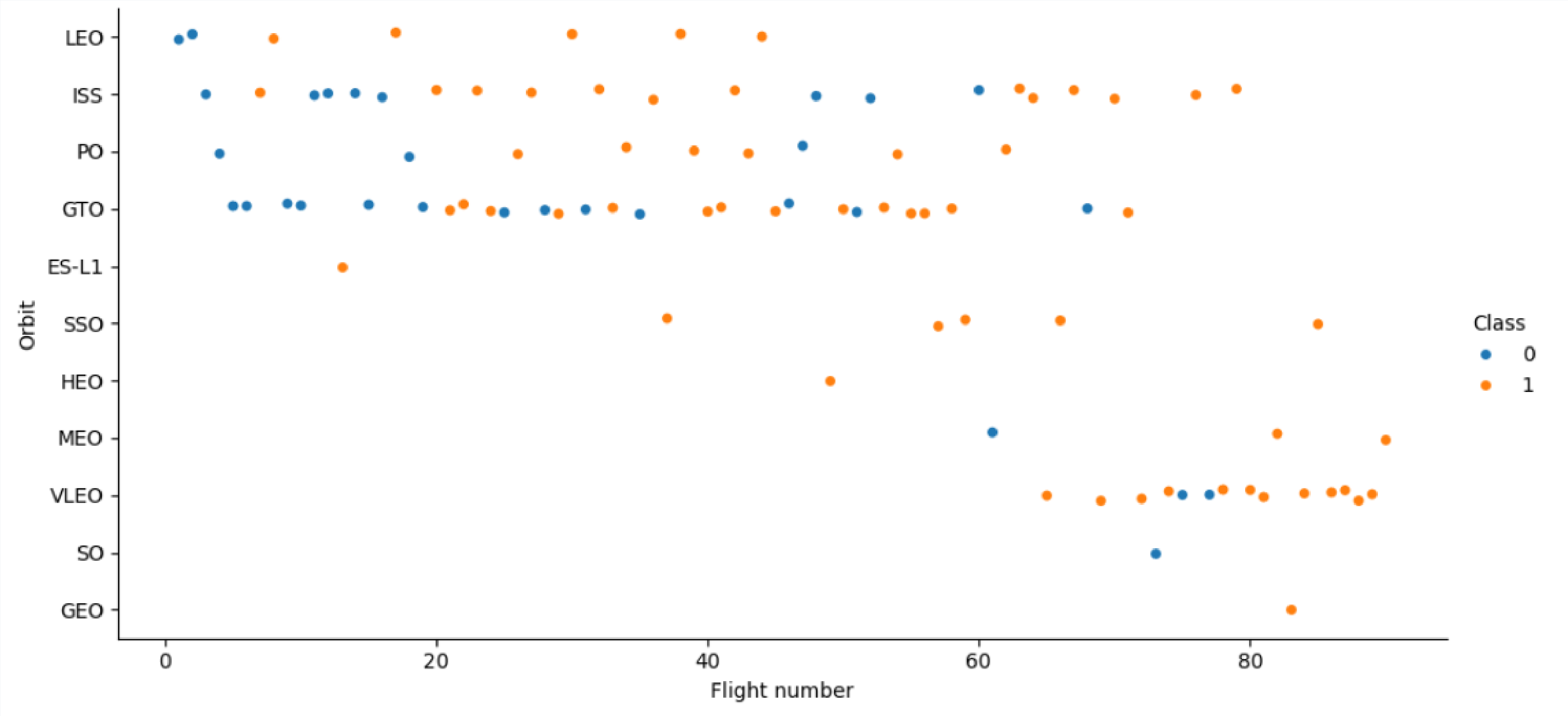
Success Rate vs. Orbit Type



- The top 4 successful orbits are ES-L1, GEO, HEO, and SSO.
- Sun Synchronous Orbit (SO) has the worst success rate by far

Flight Number vs. Orbit Type

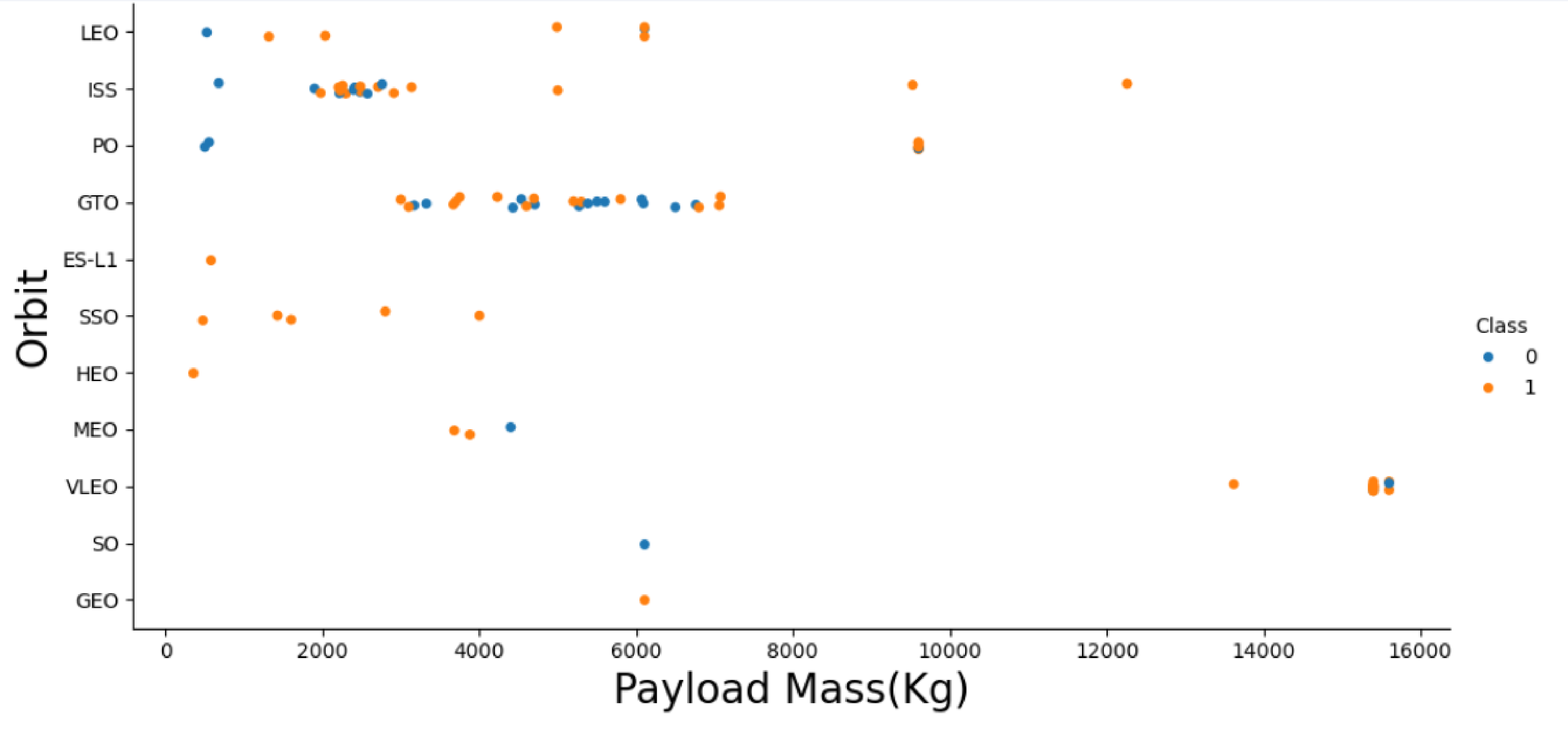
- From the graph it appears that an increase in the number of flights increases your successes



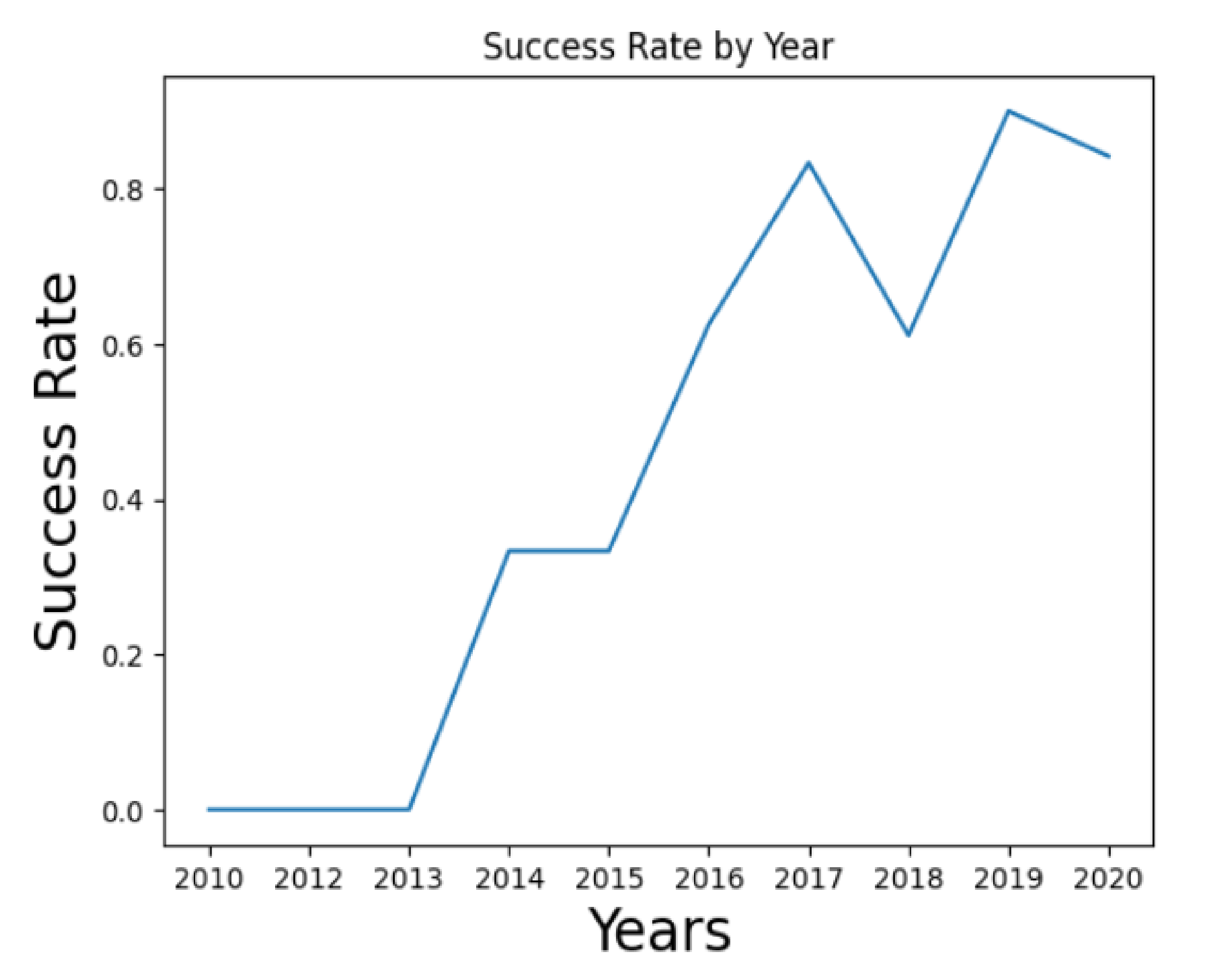
- GTO seems to not completely follow this trend so more analysis may be necessary
- Some sites lack enough data to confirm this trend for all sites (i.e. ES-L1, GEO, etc.)

Payload vs. Orbit Type

- As payloads increase the success rates for LEO, ISS and PO seem to increase
- Payload size seems to have no marked affect on GTO



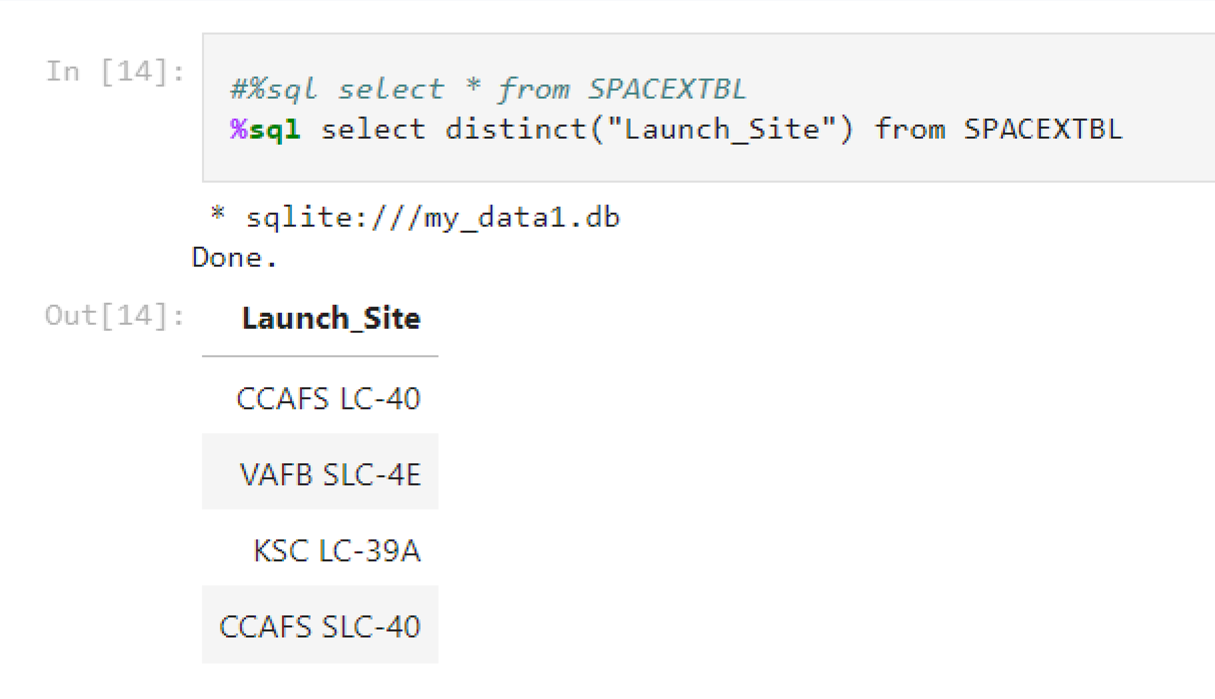
Launch Success Yearly Trend



- There seems to be an overall increase in successes over time with a dip in 2018
- This data agrees with the other figures we've already observed

All Launch Site Names

- There are only 4 unique launch sites we can launch from



In [14]:

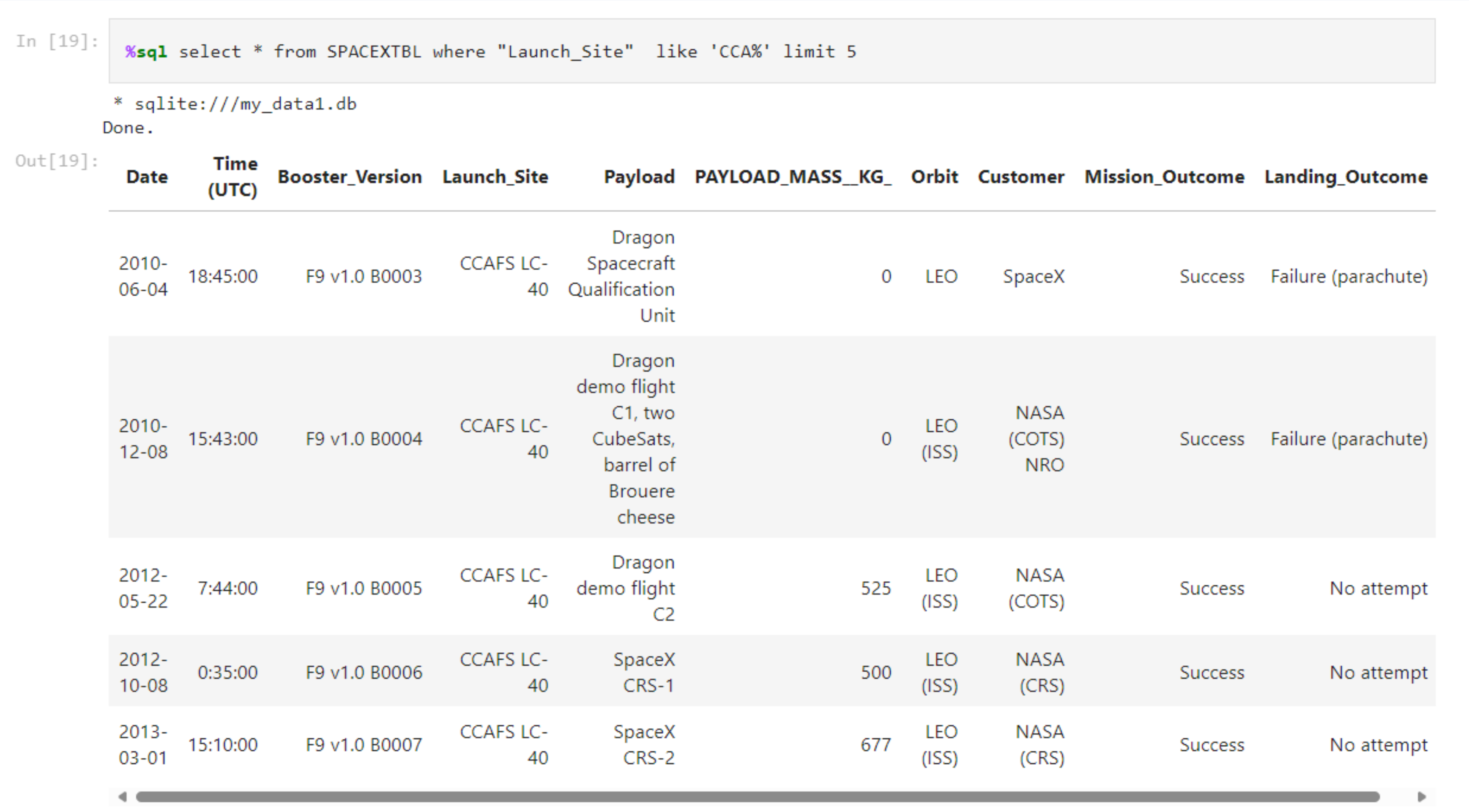
```
#%sql select * from SPACEXTBL
%sql select distinct("Launch_Site") from SPACEXTBL
```

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

Out[14]:

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

Launch Site Names Begin with 'CCA'



```
In [19]:  %sql select * from SPACEXTBL where "Launch_Site"  like 'CCA%' limit 5

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

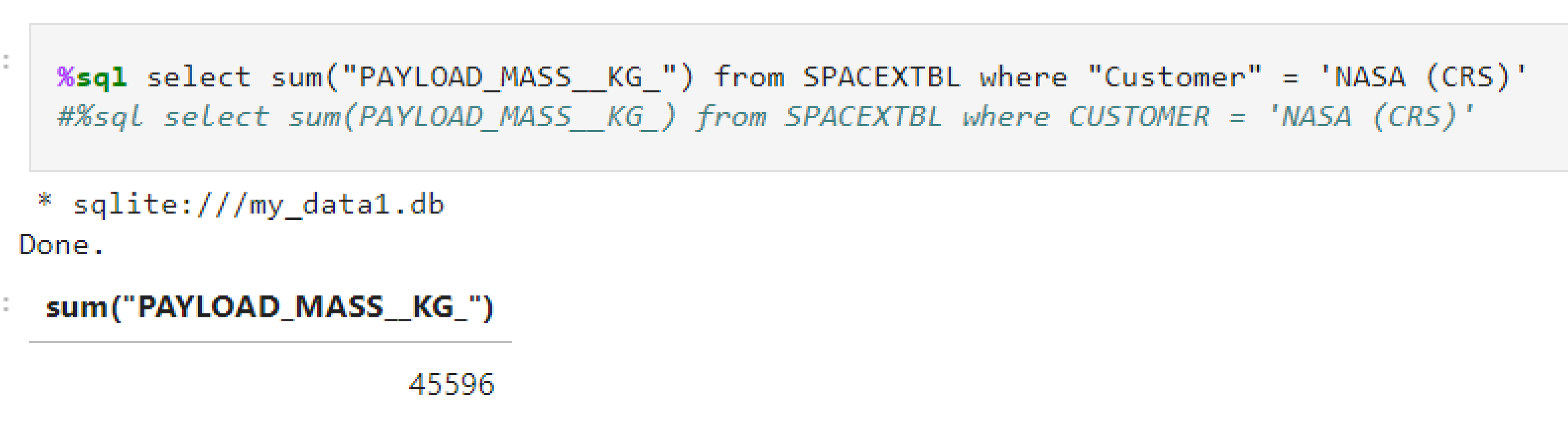
Out[19]:

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

- We can pull as many records as we need from any specific launch site
- Here we see 5 from CCAFS as an example

Total Payload Mass

- The following query told us the total payload carried by boosters from NASA is 45596 Kg



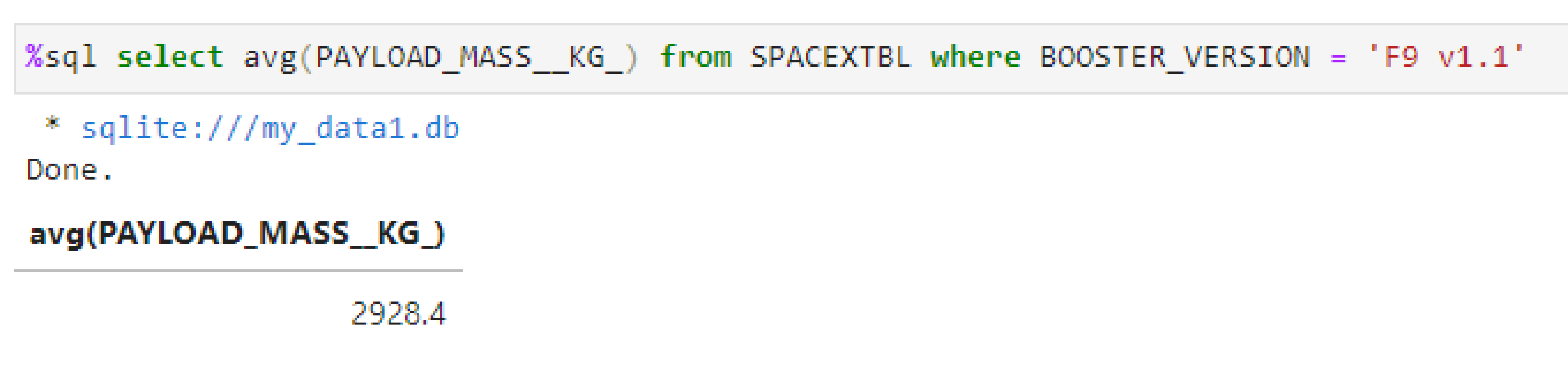
```
%sql select sum("PAYLOAD_MASS__KG_") from SPACEXTBL where "Customer" = 'NASA (CRS)'
#%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
 * 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 booster version F9 v1.1 below and found it was 2928.4 Kg



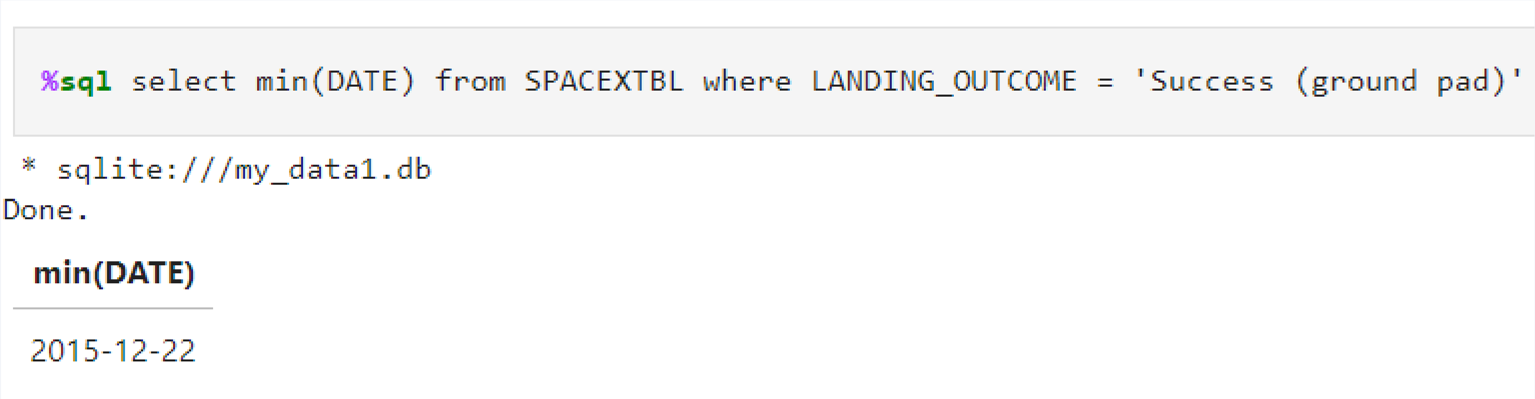
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Through the SQL search it can be seen the first successful ground landing was on December 22, 2015



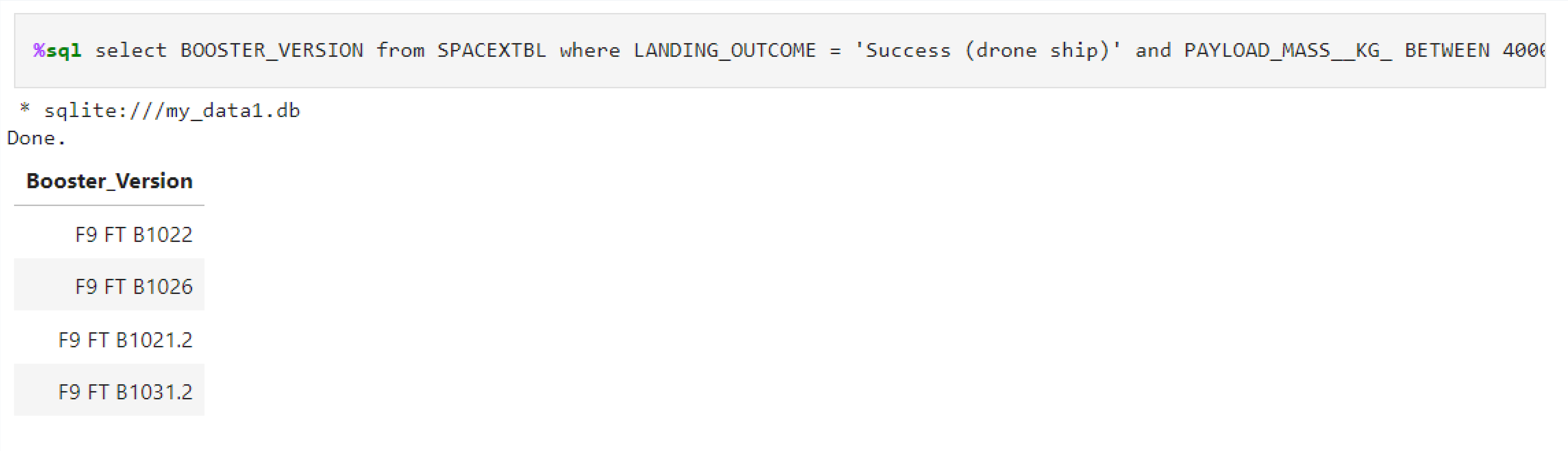
```
%sql select min(DATE) from SPACEXTBL where LANDING_OUTCOME = 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query process also revealed the names of boosters which have successfully landed on a drone ship with a payload between 4000 Kg and 6000 Kg. These are shown below.

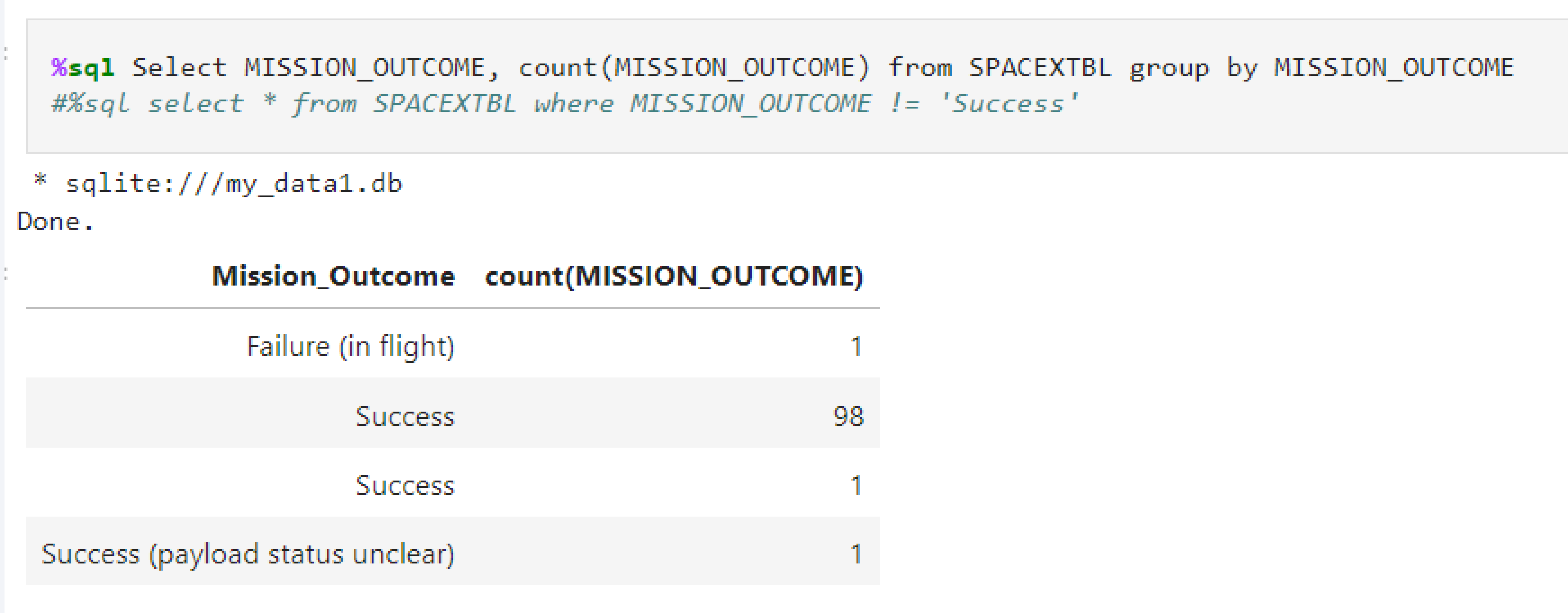


```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 400
```

```
 * 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) from SPACEXTBL group by MISSION_OUTCOME
#%sql select * from SPACEXTBL where MISSION_OUTCOME != 'Success'
```

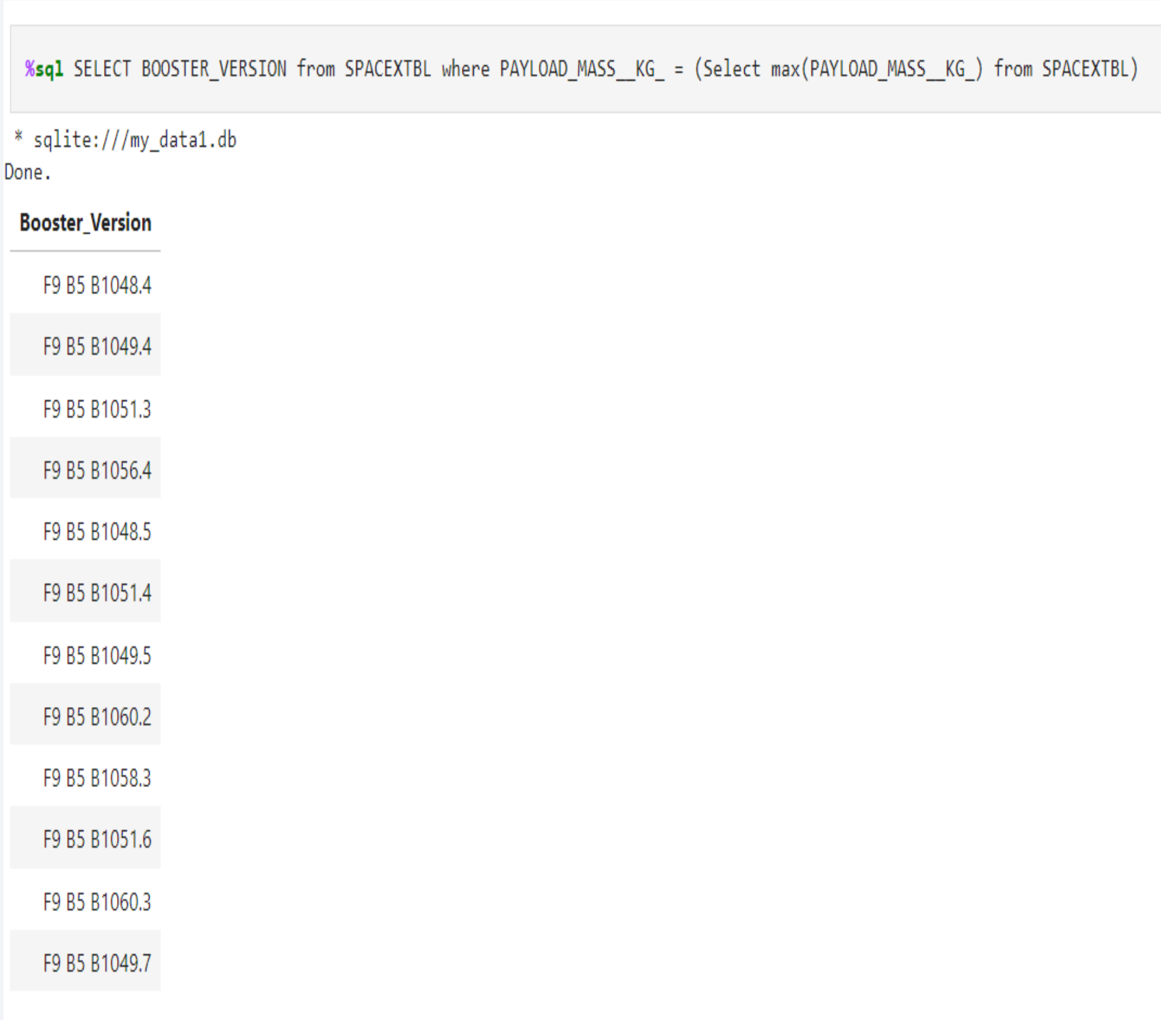
```
 * sqlite:///my_data1.db
Done.
```

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

- Using SQL and the search shown above, we can see that while landing outcomes had mixed success, the missions were widely successful.

Boosters Carried Maximum Payload

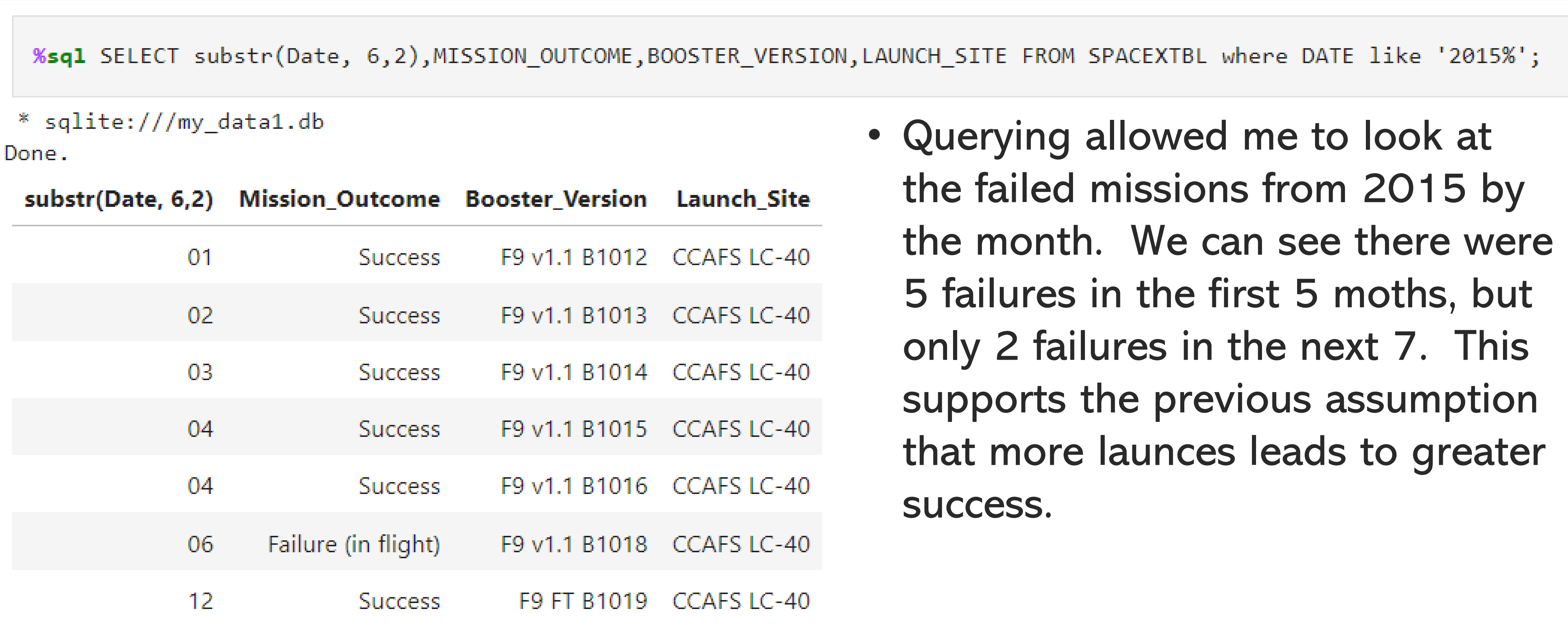
- Continuing to query revealed which boosters can support the max payload. The 11 results on the right would imply only B5 rockets can carry the max payload.



```
%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



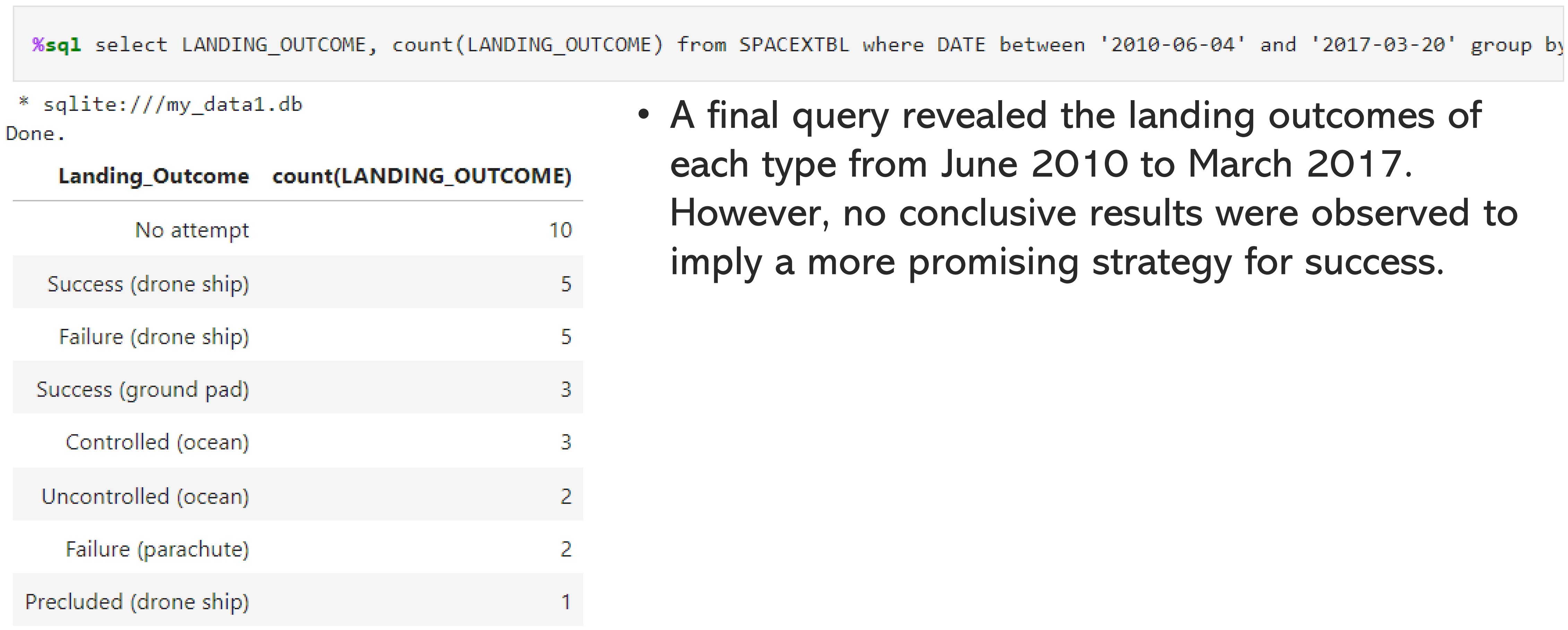
```
%sql SELECT substr(Date, 6,2),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where DATE like '2015%';
```

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

substr(Date, 6,2)	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

- Querying allowed me to look at the failed missions from 2015 by the month. We can see there were 5 failures in the first 5 moths, but only 2 failures in the next 7. This supports the previous assumption that more launces leads to greater success.

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
```

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

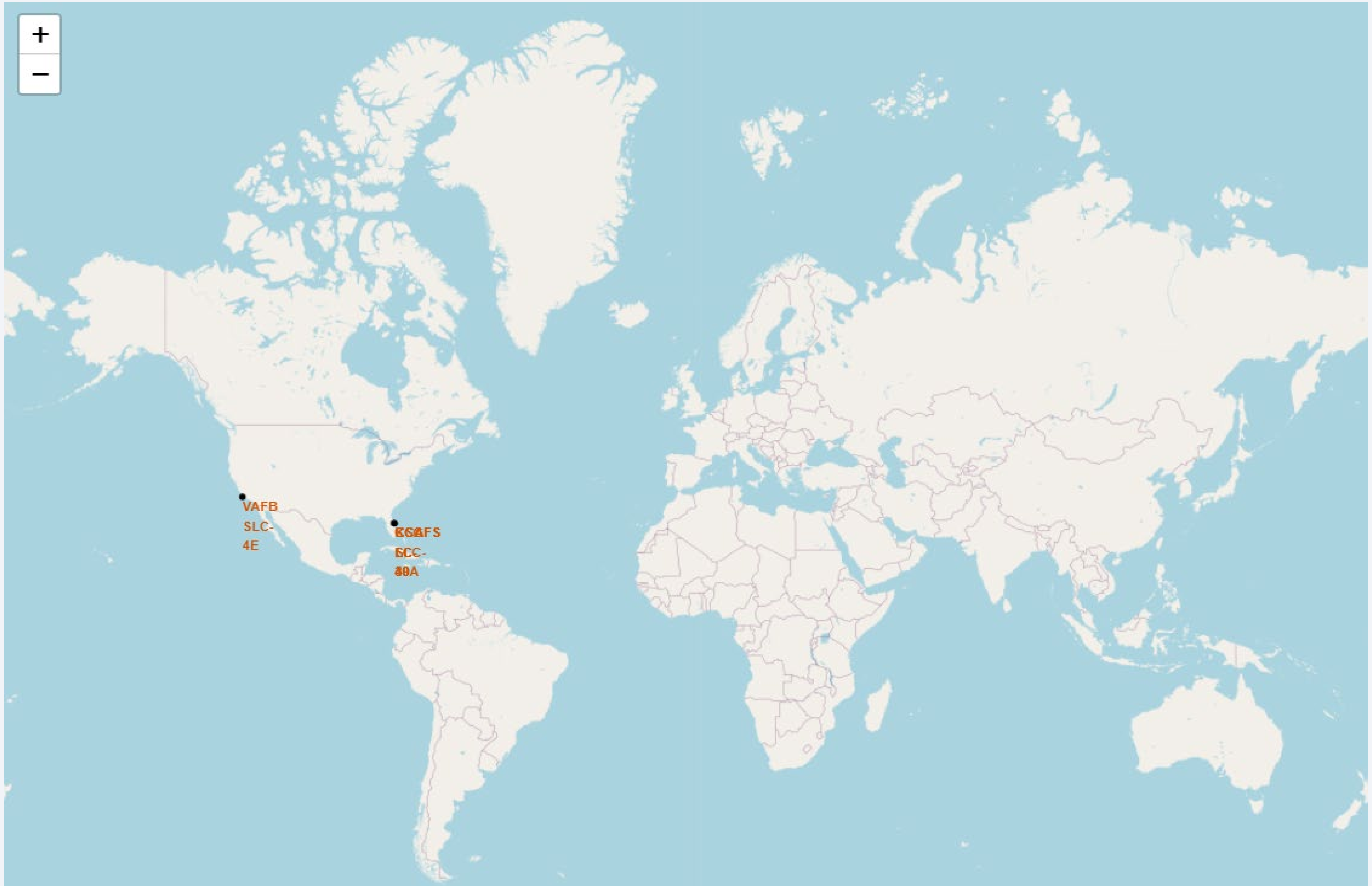
Landing_Outcome	count(LANDING_OUTCOME)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- A final query revealed the landing outcomes of each type from June 2010 to March 2017. However, no conclusive results were observed to imply a more promising strategy for success.

Section 3

Launch Sites Proximities Analysis

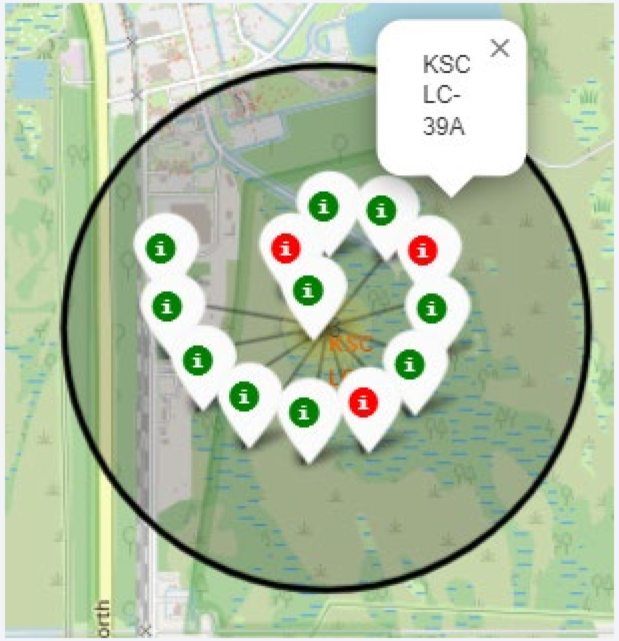
Space X Launch Site Locations

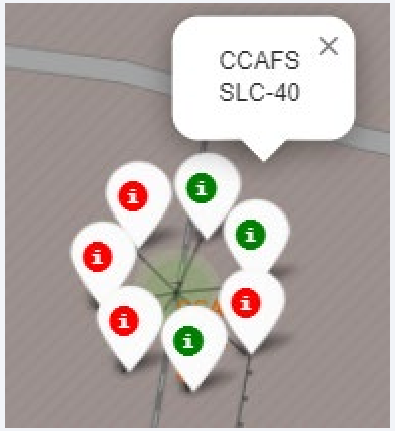


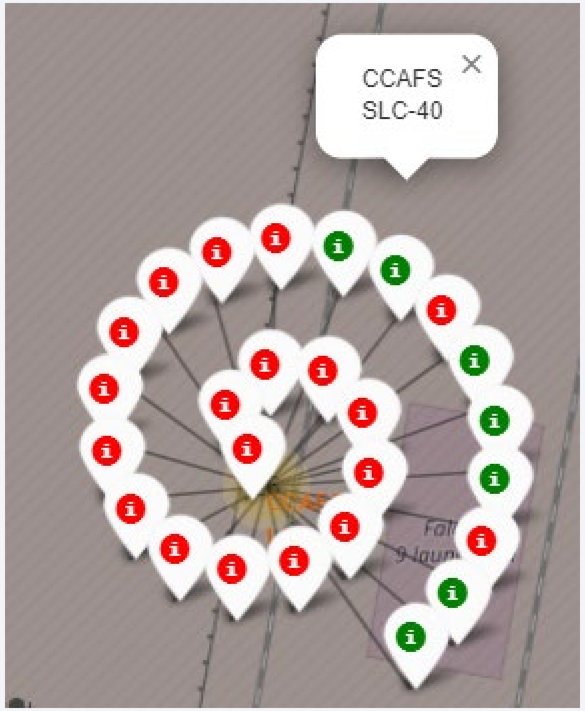
- All Space X Launch sites are located on the coasts of North America.
- Specifically, One is in California (VAFB SLC 4E) and the other 3 are in Florida

Success/Failures By Launch Site

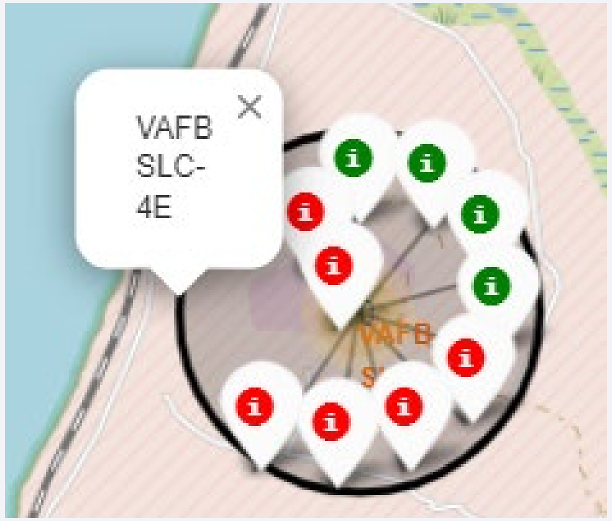
Florida Launch sites





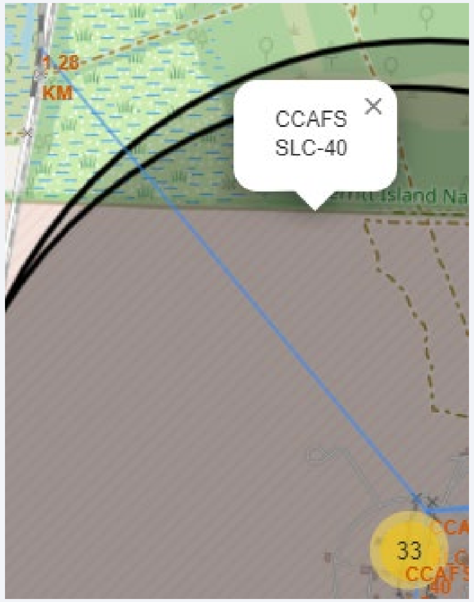


California launch site

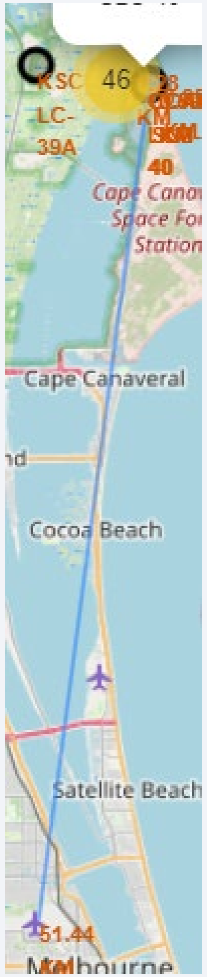


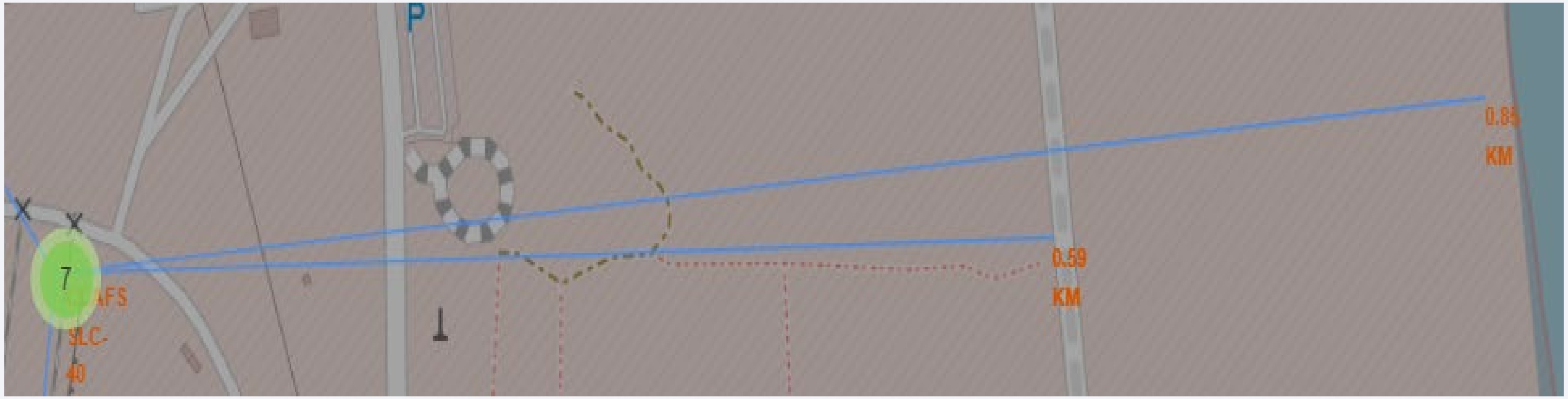
Green markers signify successful landings, and red markers show failures

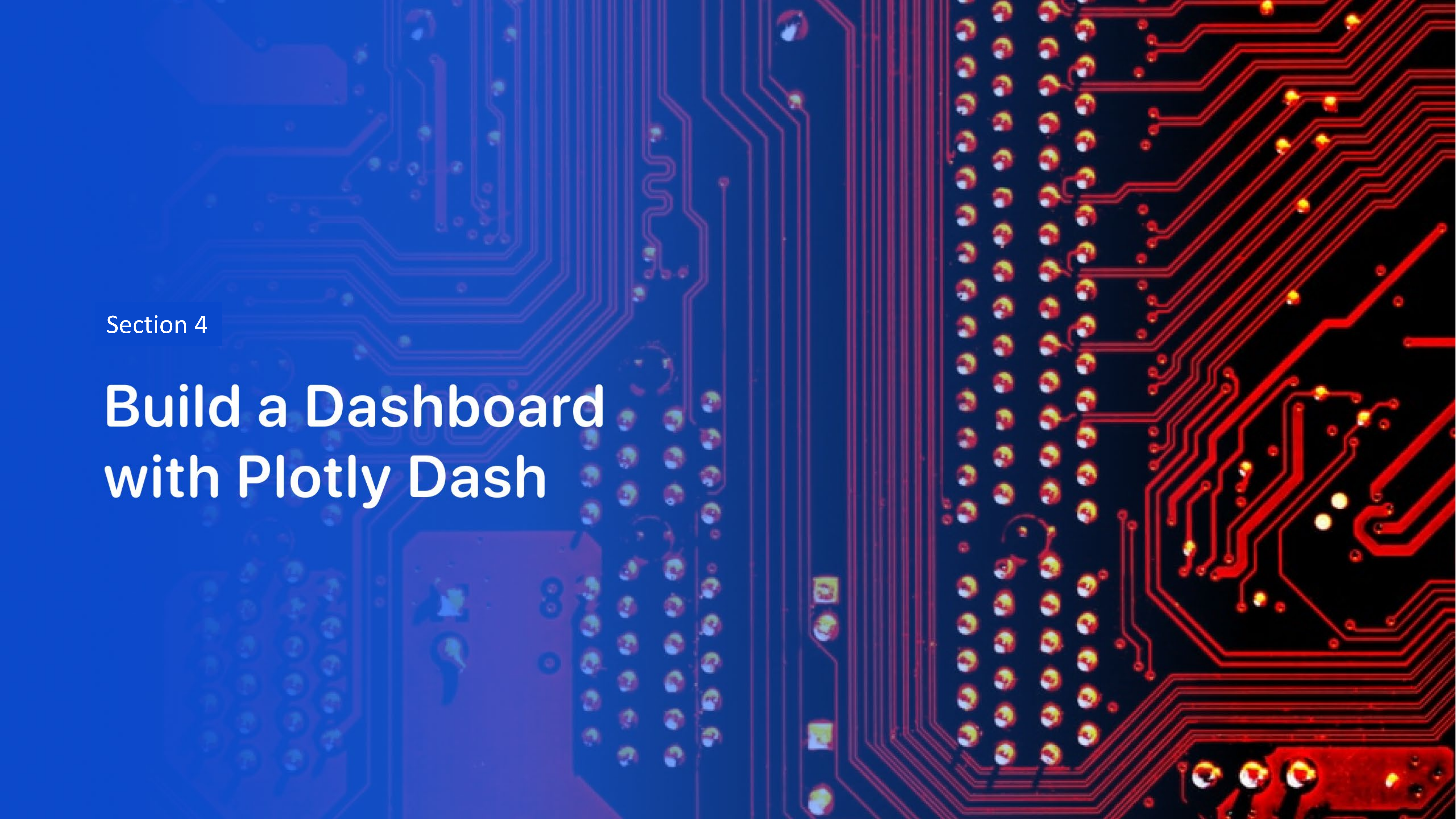
Launch Site Proximity from Landmarks



- Launch site appear to be built near Coastlines and Highways. Here less than a Km from each. (shown above)
- Near railroads also seems to be a preferred characteristic for launch sites. The picture to the left shows the nearest railroad is just over a Km away.
- Inversly launch sites appear to try and avoid major cities. On the right you can see the nearest city is over 50 Km away



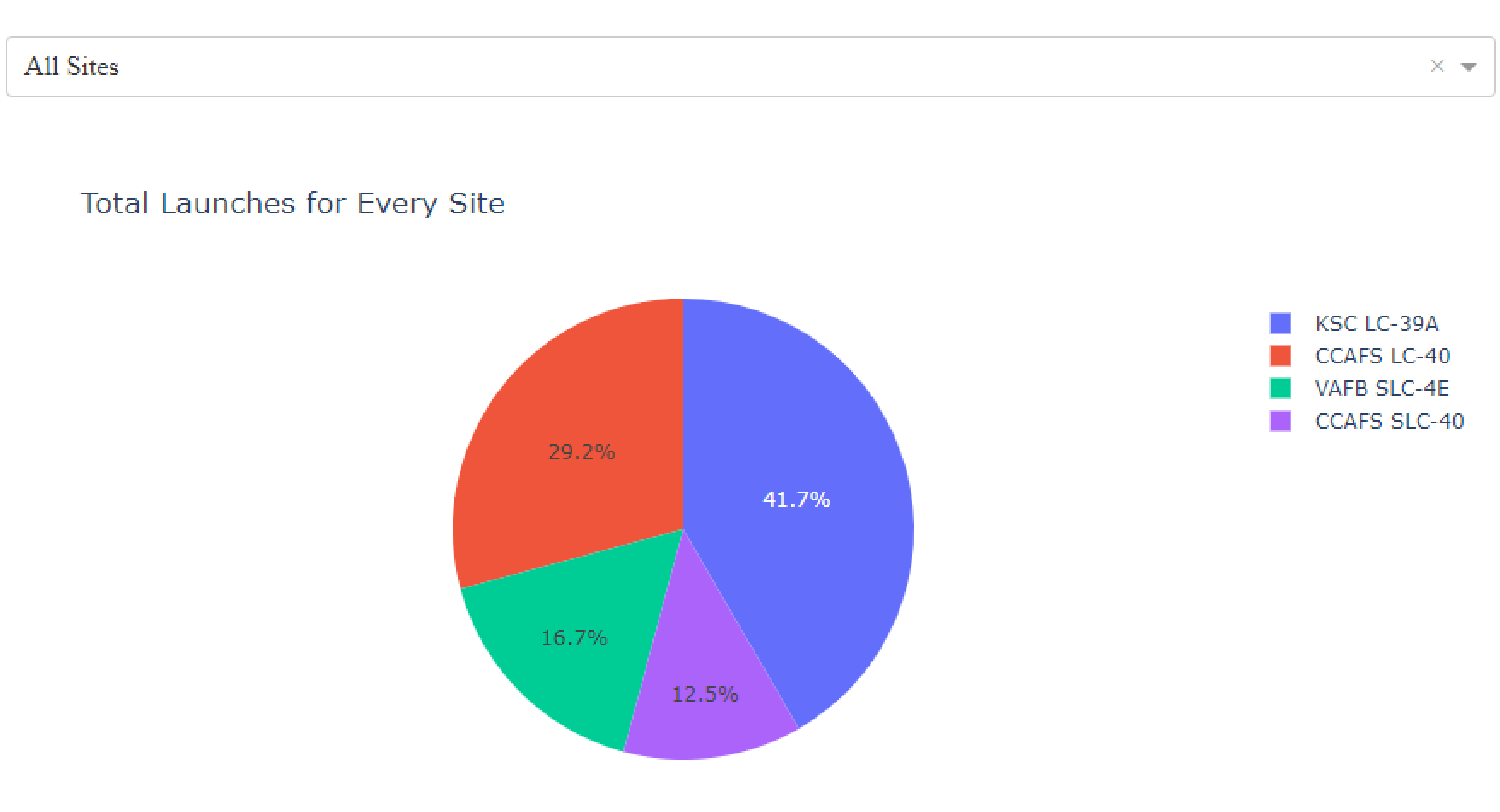




Section 4

Build a Dashboard with Plotly Dash

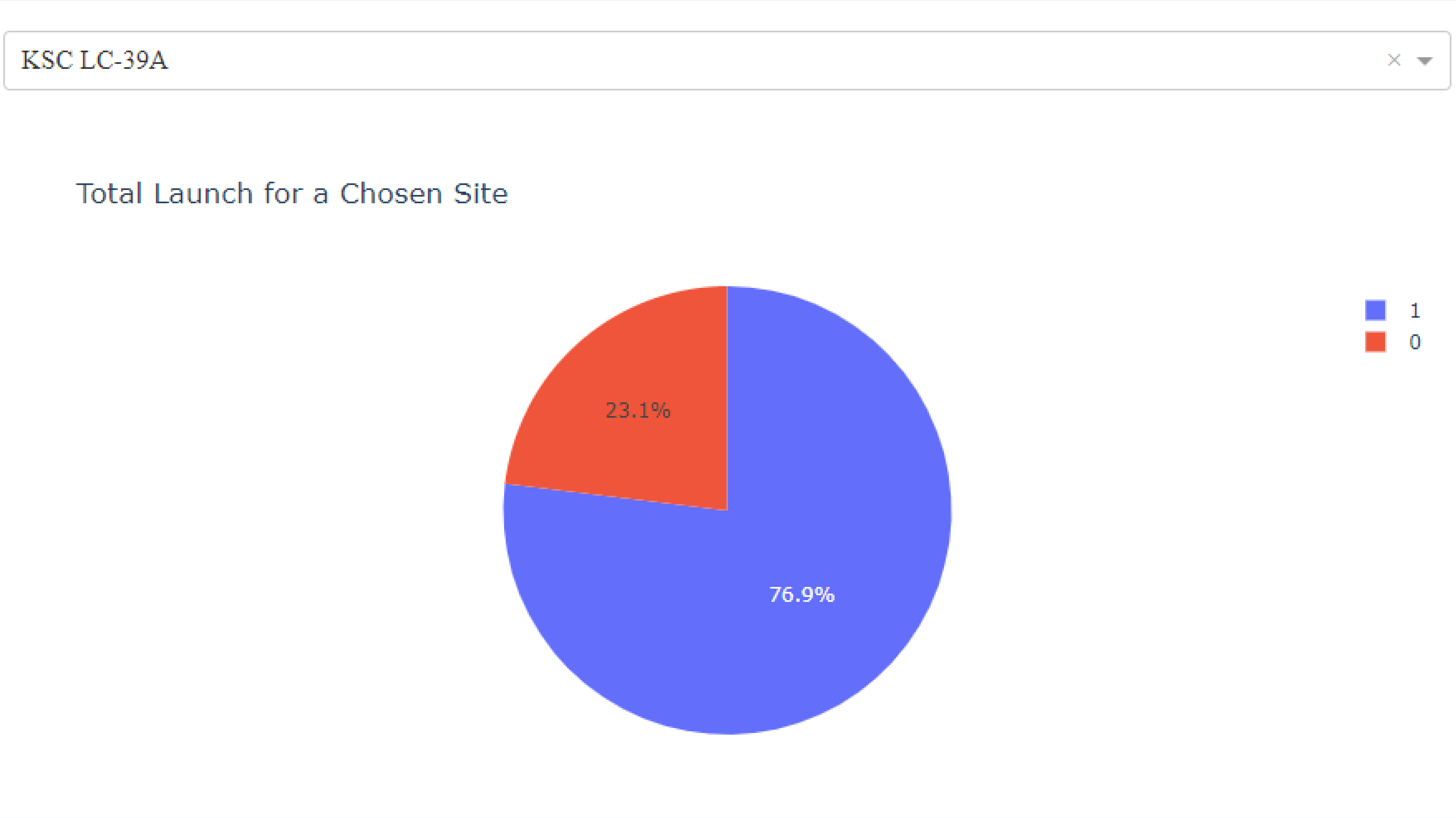
Total Successful Launches Across All Sites



All Sites

- Launch site KSC LC-39 has the highest most successful launches of the 4 sites.

KSC LC-39A Drill Down

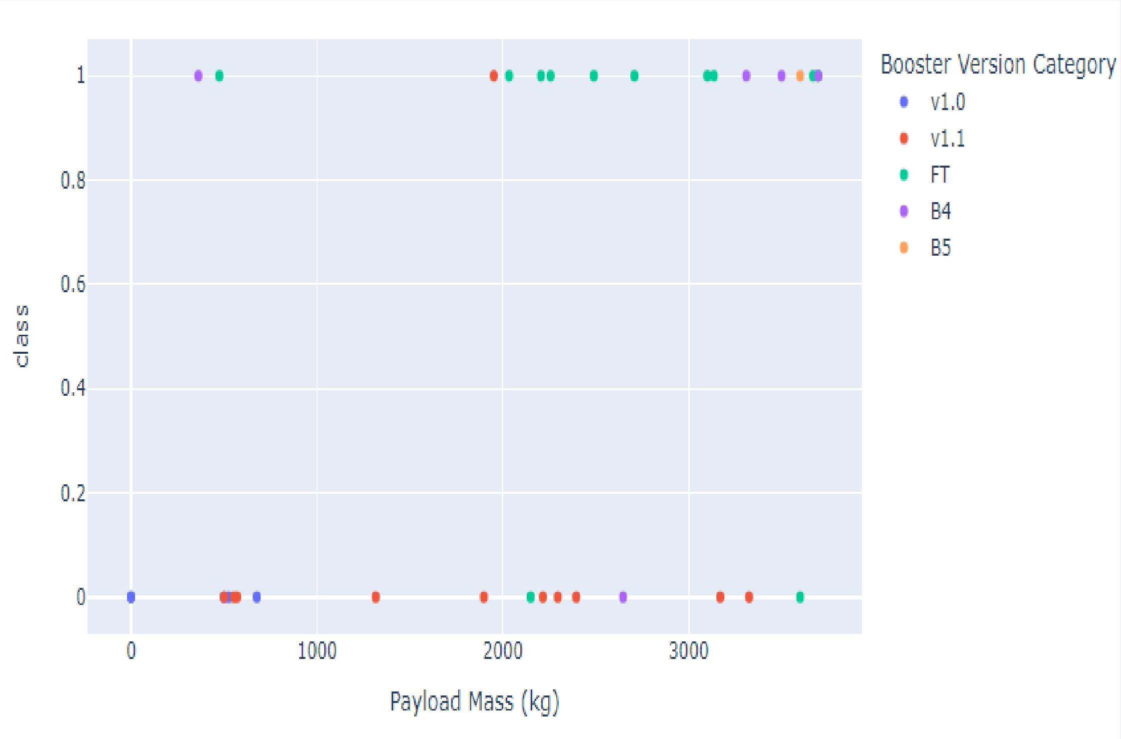


KSC LC-39A

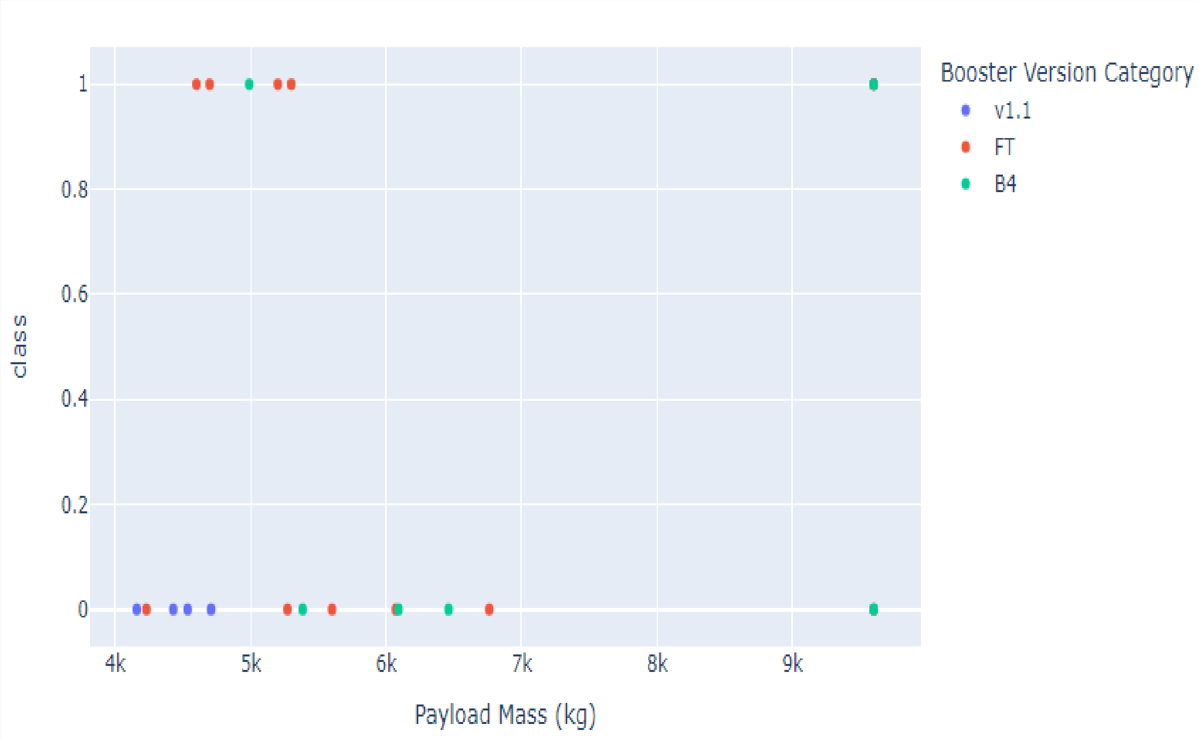
- Even though KSC LC 39-A had the most successes, their overall success percent is only about 77%. There is still room for improvement.

Low vs Heavy Payload Comparison

Low Weight (0Kg – 4,000Kg)



Heavy Weight (4,000Kg – 10,000Kg)



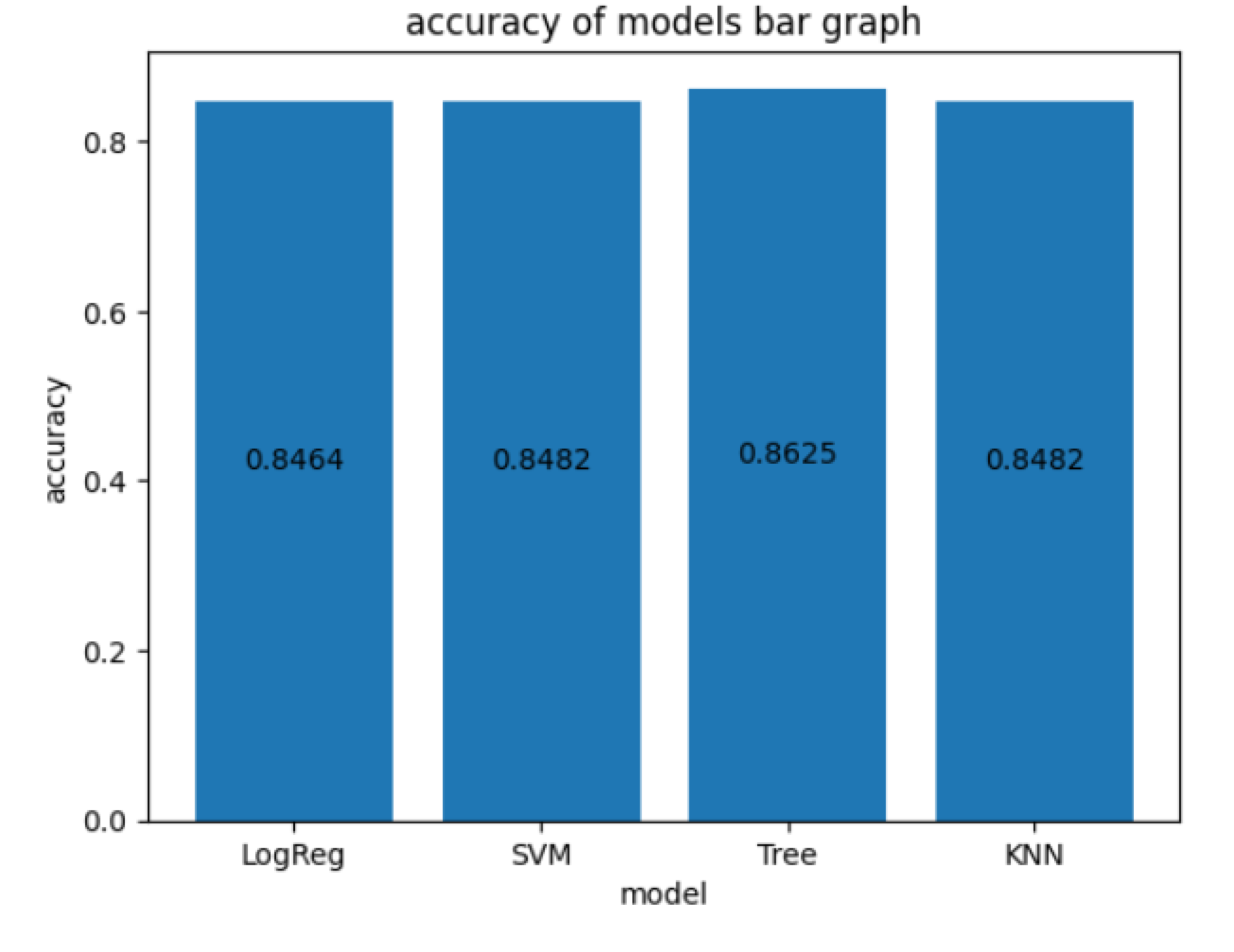
There seems to be a higher success rate for Lighter loads than heavy loads.

Section 5

Predictive Analysis (Classification)

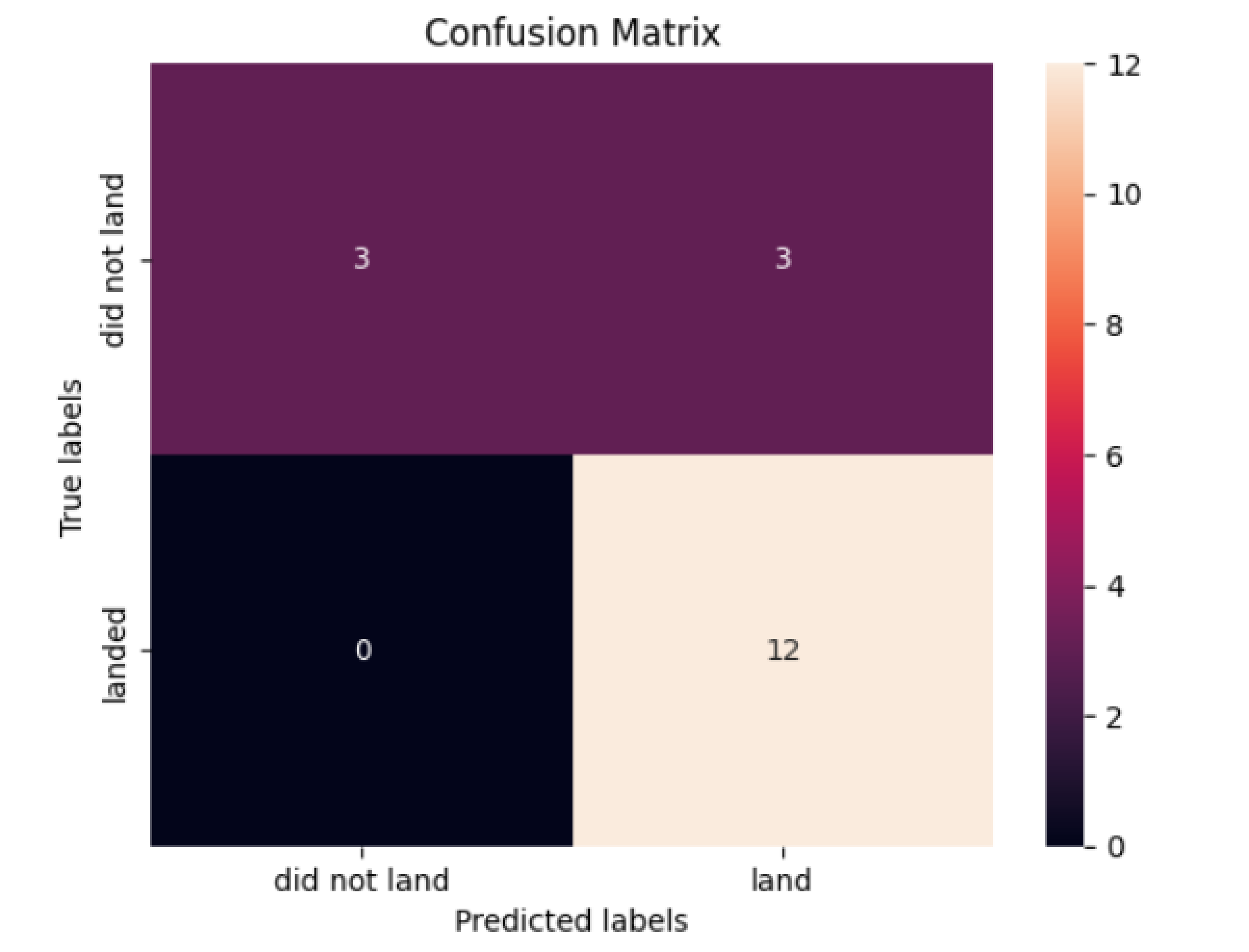
Classification Accuracy

- The Best model appears to be the Decision Tree Model with an accuracy of 86.25%



Confusion Matrix

- The confusion matrix for the Decision Tree can help us identify major strengths and weaknesses in our model that appeared best.



In this case We predicted 3 times that we would land incorrectly showing this model is prone to occasionally provide a false-positive

Conclusions

- The more times a site attempts a launches, the better their outcomes.
- The orbits that yield the highest success rates are ES-L1,GEO, HEO, and SSO.
- If we have a heavier payload, we should use a Leo, ISS, or POLAR orbit to increase our chances for success.
- There has been a steady increase in the success rates since 2013, with a few drops like in 2018. In general though there has been a positive trend over time.
- The site with the most successful launches is KSC LC-39
- In ordert o make predictions we should use the Decision tree model for the most accurate results
- When building our facilities we should try to find a location close to highways, train tracks, and coastlines, but not too close to highly populated areas like cities

Thank you!