IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jack McLay

06/09/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Data Analysis w/ Data Visualizations
  - Data Analysis w/ SQL
  - Building an Interactive Map (Folium)
  - Building a Dashboard (Dash)
  - Preforming Predictive Analysis (ML Classification)
- Summary of all results
  - Results include my findings from Exploratory Data Analysis, Interactive Mapping Analysis, and Predictive Analysis

Introduction

- Project background and context
  - The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets. However, the most successful is SpaceX.
  - One reason SpaceX can do this is the rocket launches are relatively inexpensive.
    - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars
    - While other providers cost upwards of 165 million dollars each.
  - Much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- Problems you want to find answers
  - Therefore, the task of this project is to predict if the first stage of the Falcon 9 rocket will land successfully or crash and burn.

Section 1

Methodology

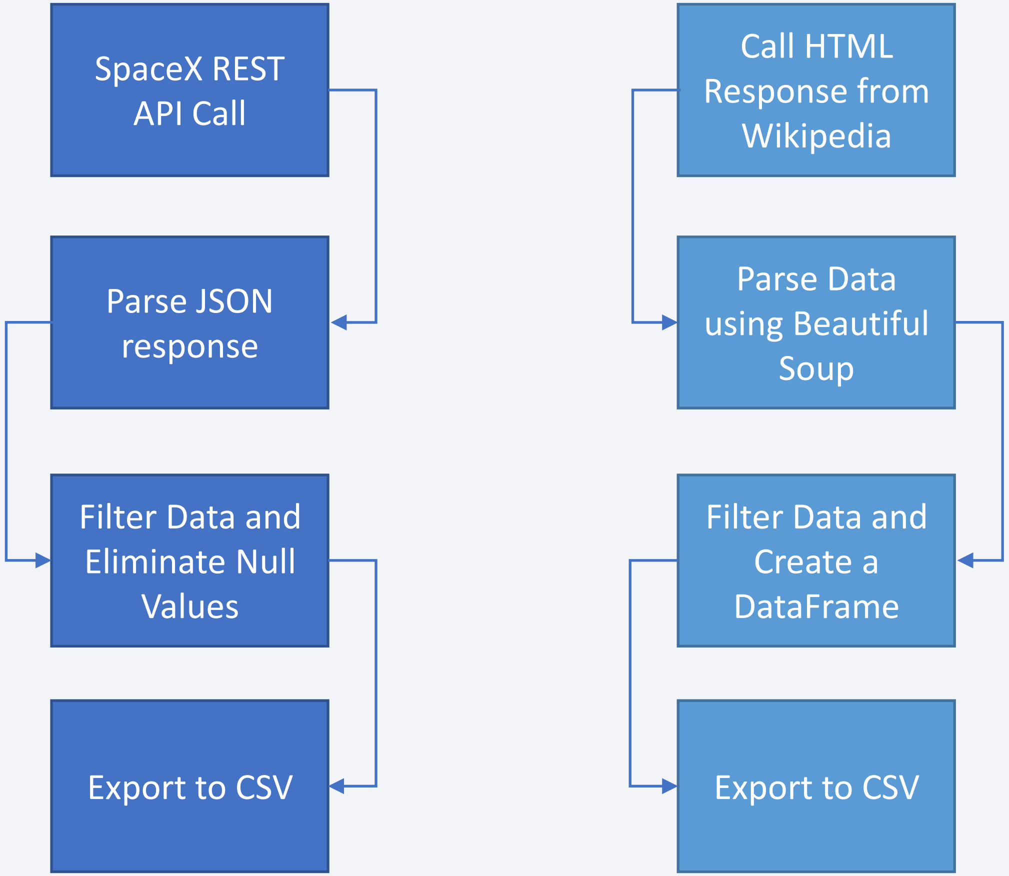
Methodology

Executive Summary

- Data collection methodology:
    - The data was collected from a number of different sources including:
        - SpaceX Rest API
        - Web Scraping from Wikipedia
- Perform data wrangling
    - Scrubbed the data removing null values and irrelevant data, along with One Hot Encoding (Pandas get_dummies) to prepare fields for ML models.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Including Linear Regression, K-Nearest Neighbor, SVM, and Decision Trees

Data Collection

- Describe how data sets were collected.
  - The SpaceX Launch Site data was collected from SpaceX's REST API.
  - This data included details about, the launch pad, rocket used, payload, failures, successes, dates, etc.
  - Our other source of data involved web scrapping data about the Falcon 9 launches from Wikipedia using Beautiful Soup.

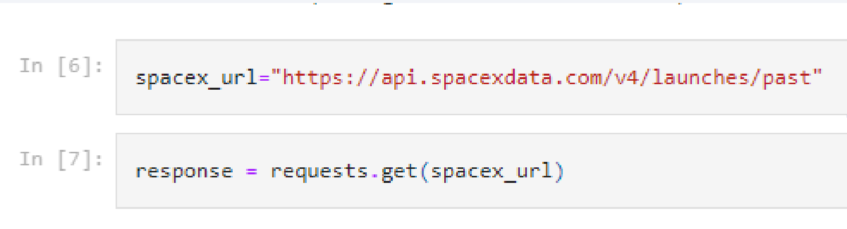

SpaceX REST API Call
Parse JSON response
Filter Data and Eliminate Null Values
Export to CSV
Call HTML Response from Wikipedia
Parse Data using Beautiful Soup
Filter Data and Create a DataFrame
Export to CSV

Data Collection – SpaceX API

- Data collection with SpaceX REST calls and Code examples:

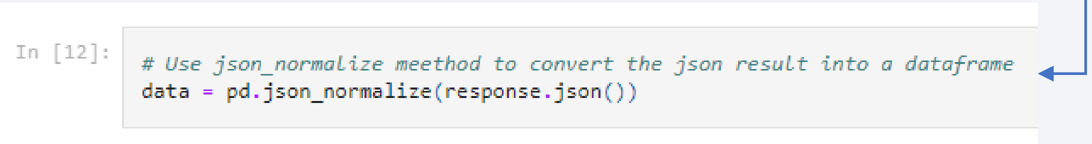
- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. Call the API



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

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

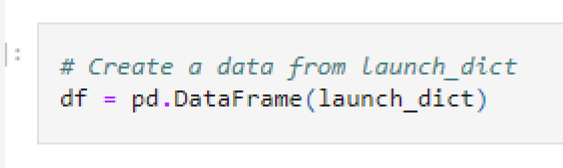
2. Convert to JSON:



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

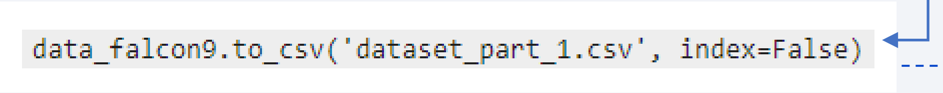
3. Clean Data using Functions: getBoosterVersion(data), getLaunchSite(data), getPayloadData(data), getCoreData(data)

4. Create DataFrame from Cleaned JSON:



```
]: # Create a data from launch_dict
   df = pd.DataFrame(launch_dict)
```

5. Export to a CSV



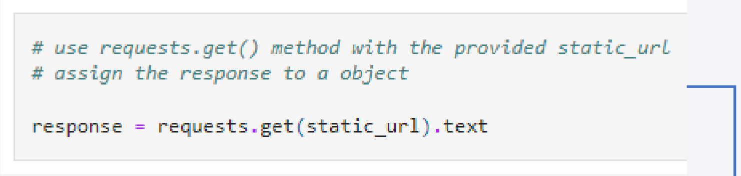
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Web scraping from Falcon 9 Wikipedia with Code examples:

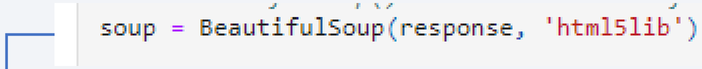
- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

1. Get Response from Static URL:



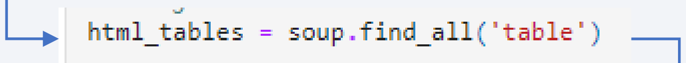
```
# use requests.get() method with the provided static_url
# assign the response to a object

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

2. Create BeautifulSoup Parser:

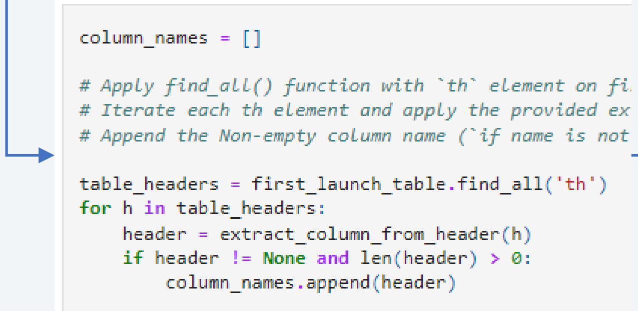


```
soup = BeautifulSoup(response, 'html5lib')
```

3. Find tables and get column names:

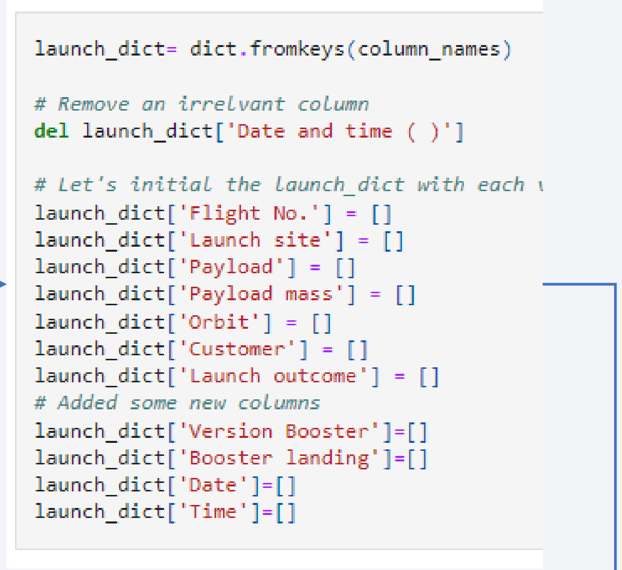


```
html_tables = soup.find_all('table')
```



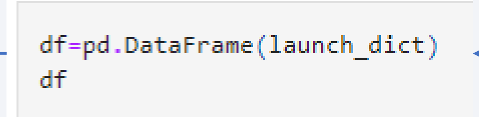
```
column_names = []

# Apply find_all() function with `th` element on fi
# Iterate each th element and apply the provided ex
# Append the Non-empty column name (`if name is not

table_headers = first_launch_table.find_all('th')
for h in table_headers:
    header = extract_column_from_header(h)
    if header != None and len(header) > 0:
        column_names.append(header)
```

4. Create dictionary:

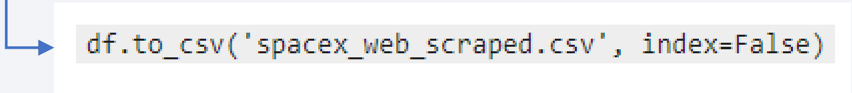


```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

5. Convert to a DataFrame then CSV:

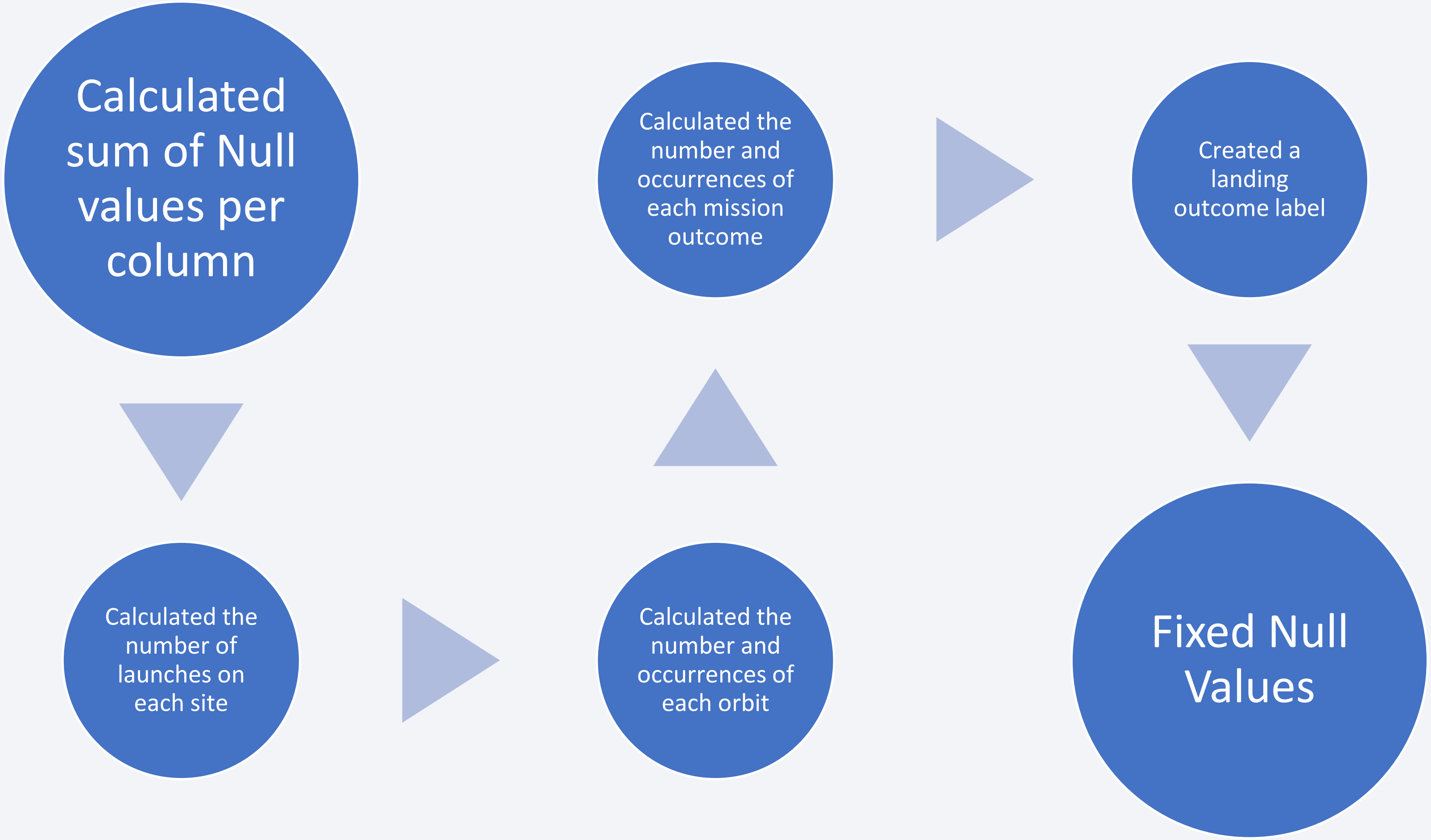


```
df=pd.DataFrame(launch_dict)
df
```



```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling



Calculated sum of Null values per column

Calculated the number of launches on each site

Calculated the number and occurrences of each orbit

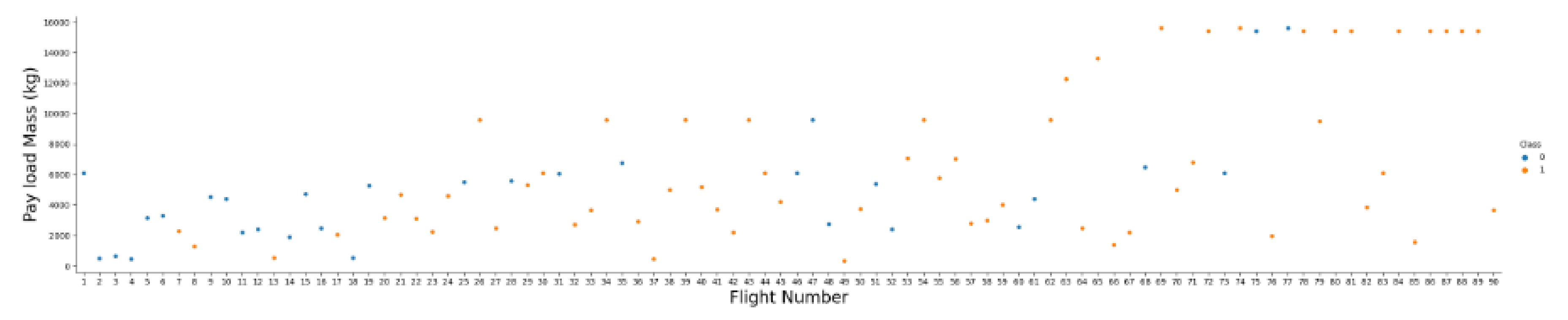
Calculated the number and occurrences of each mission outcome

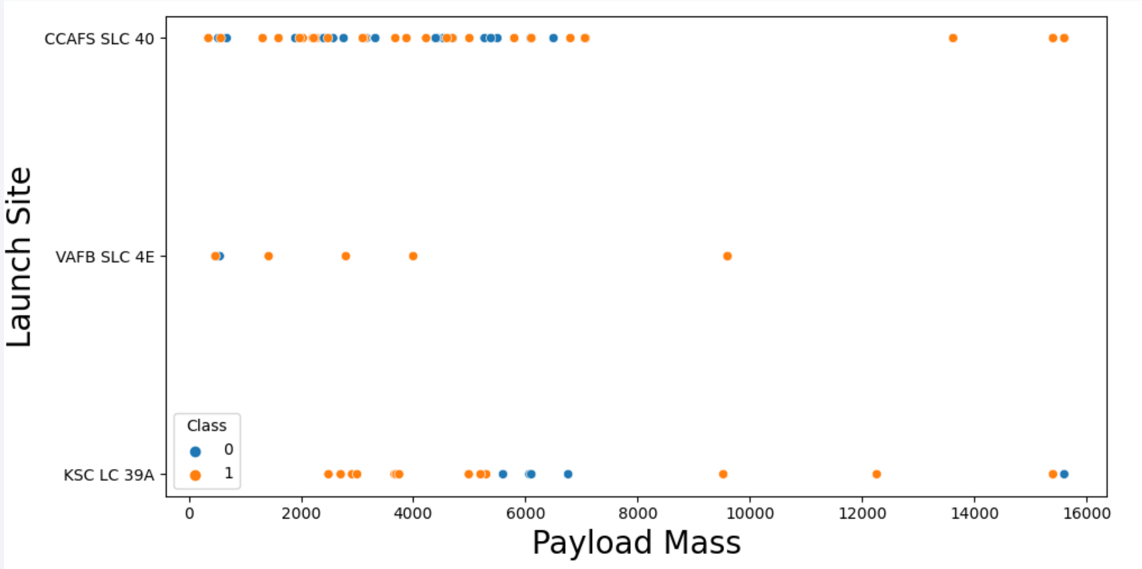
Created a landing outcome label

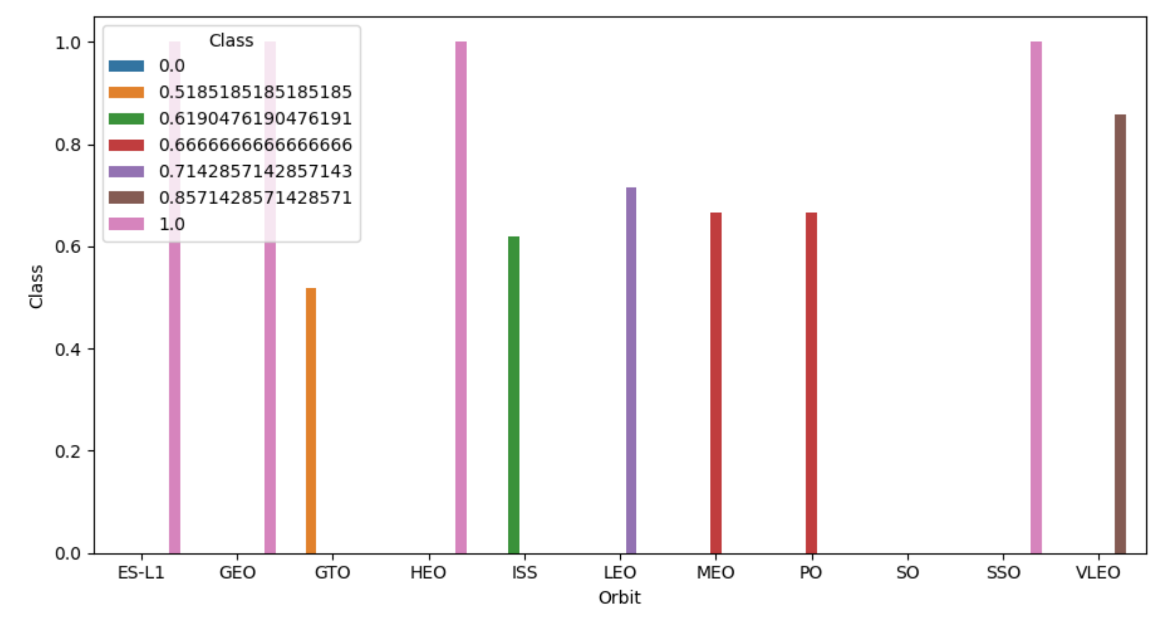
Fixed Null Values

- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

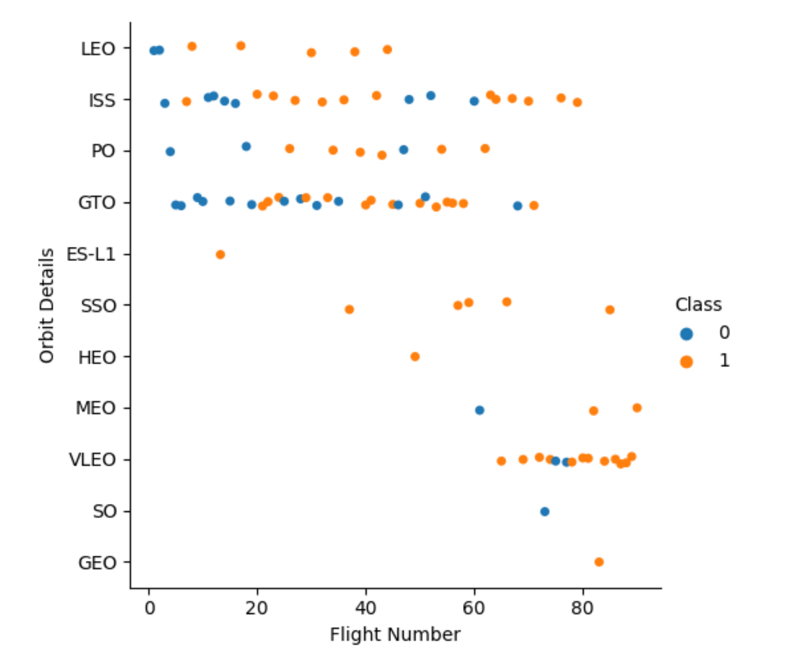


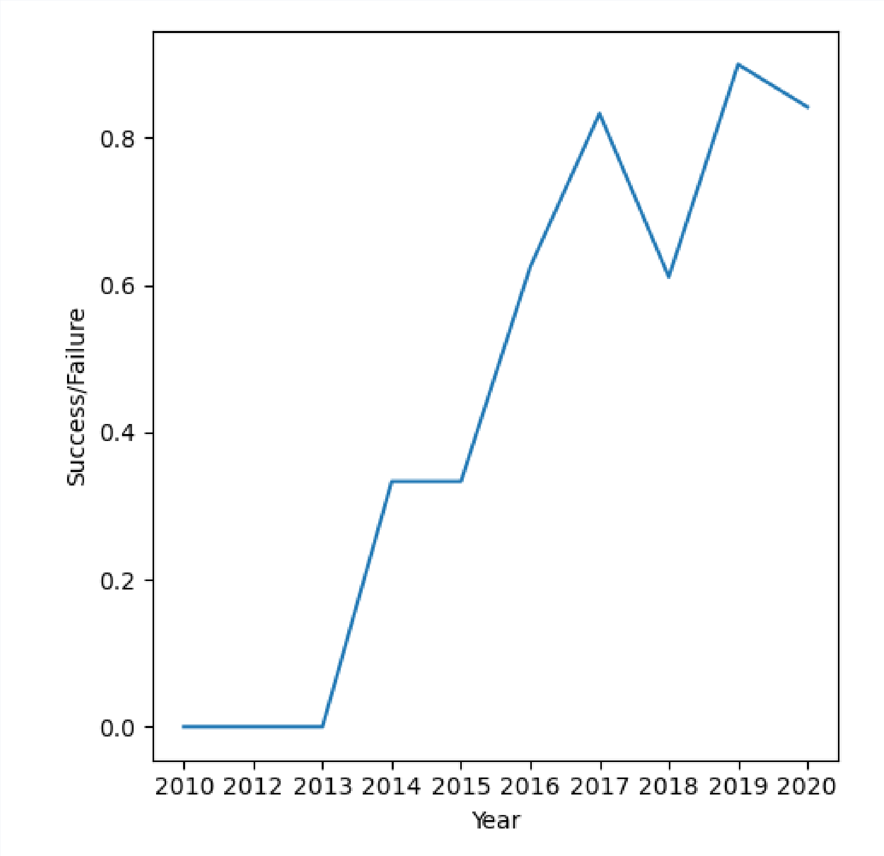


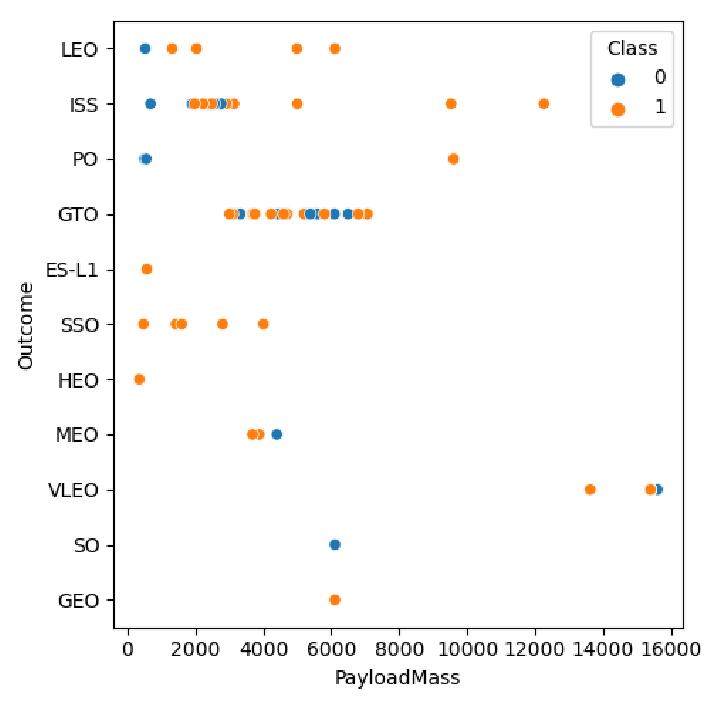


- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/eda-dataviz.ipynb.jupyterlite.ipynb

EDA with Data Visualization







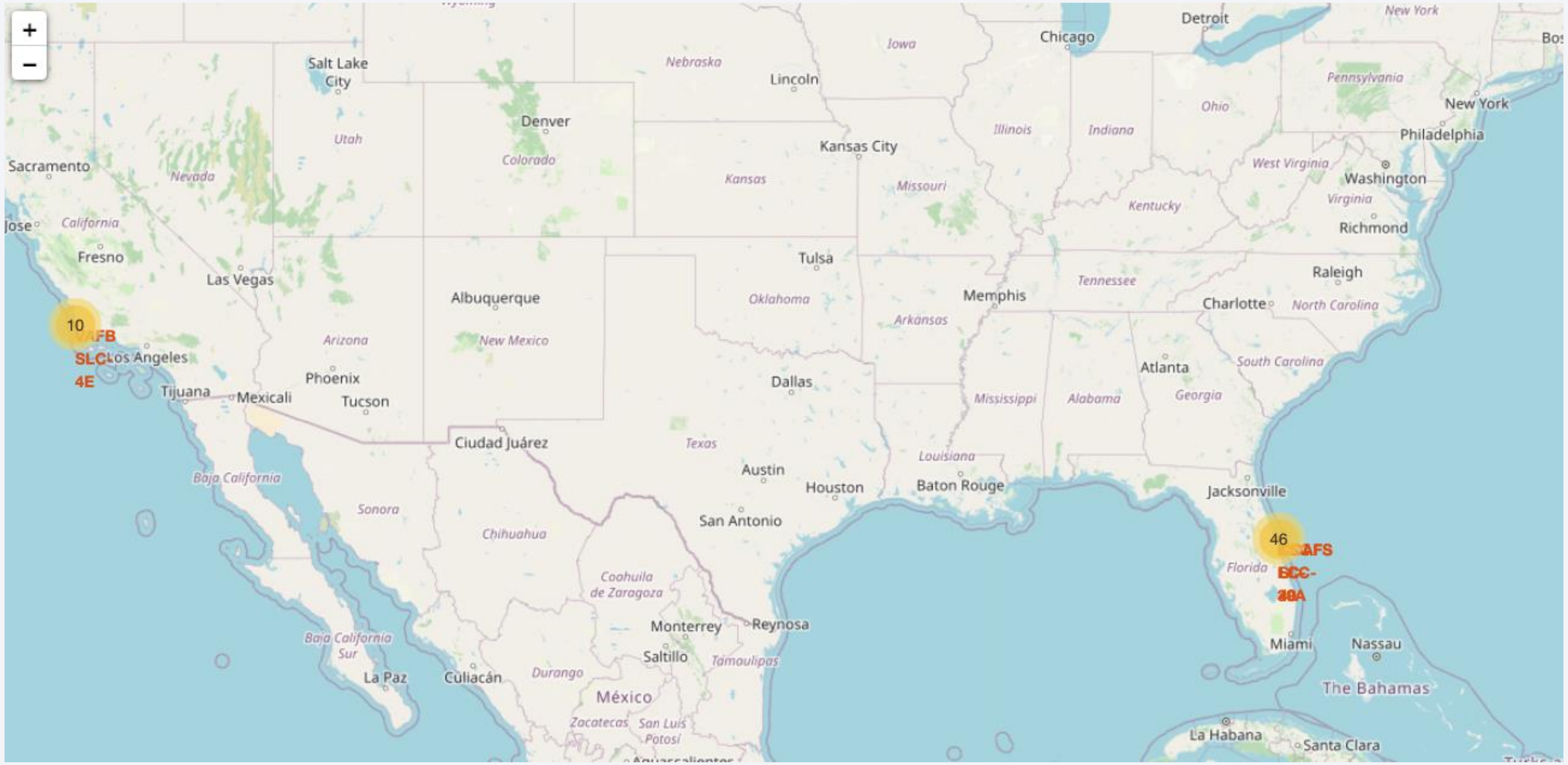
- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- SQL Queries Preformed:
  - 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 successful landing outcome in ground pad was achieved.
  - 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. Use a subquery
  - 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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/eda-sql-coursera_sqllite.ipynb

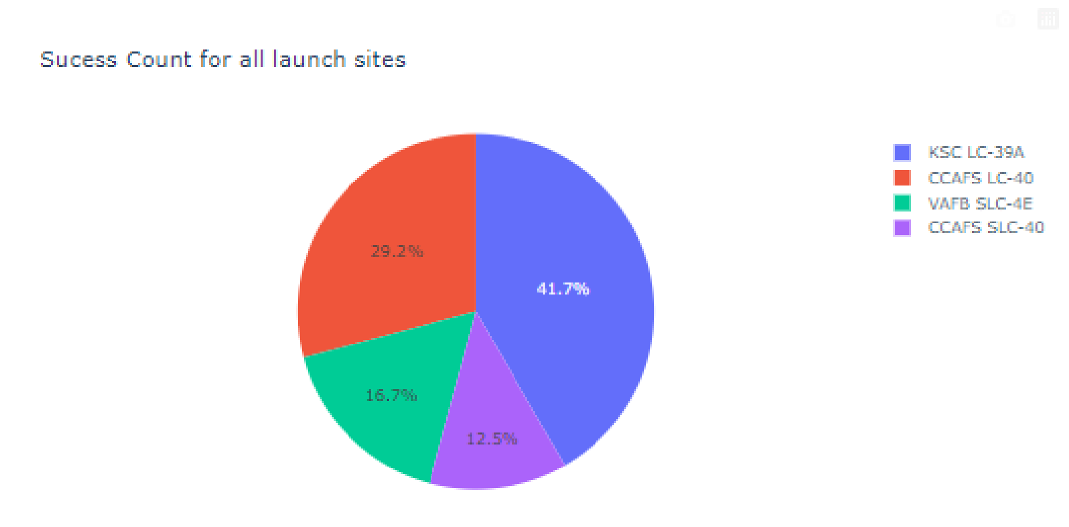
Build an Interactive Map with Folium

- These markers were added to the map to explore why launch sites are close to the equator, coast, railways, highways, and far(ish) from cities/towns.

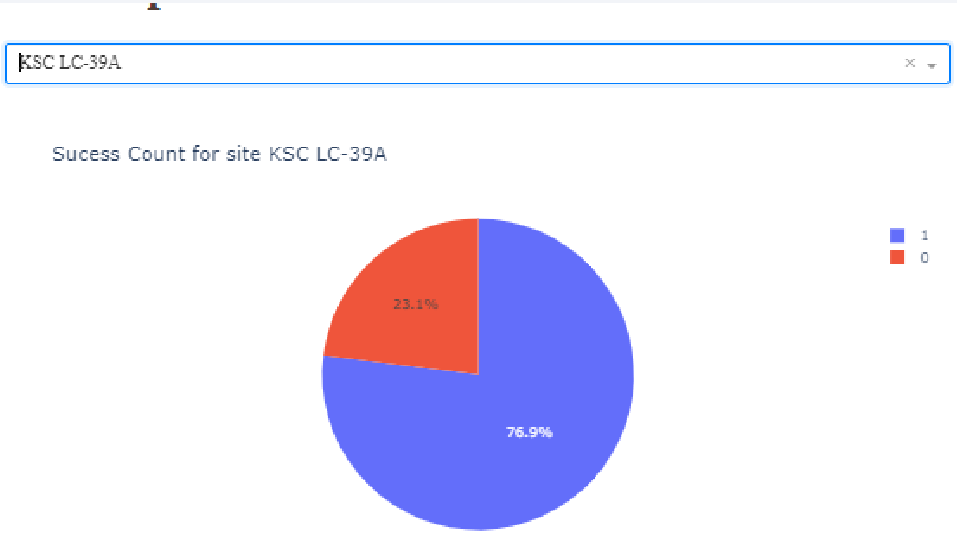


- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb

Build a Dashboard with Plotly Dash



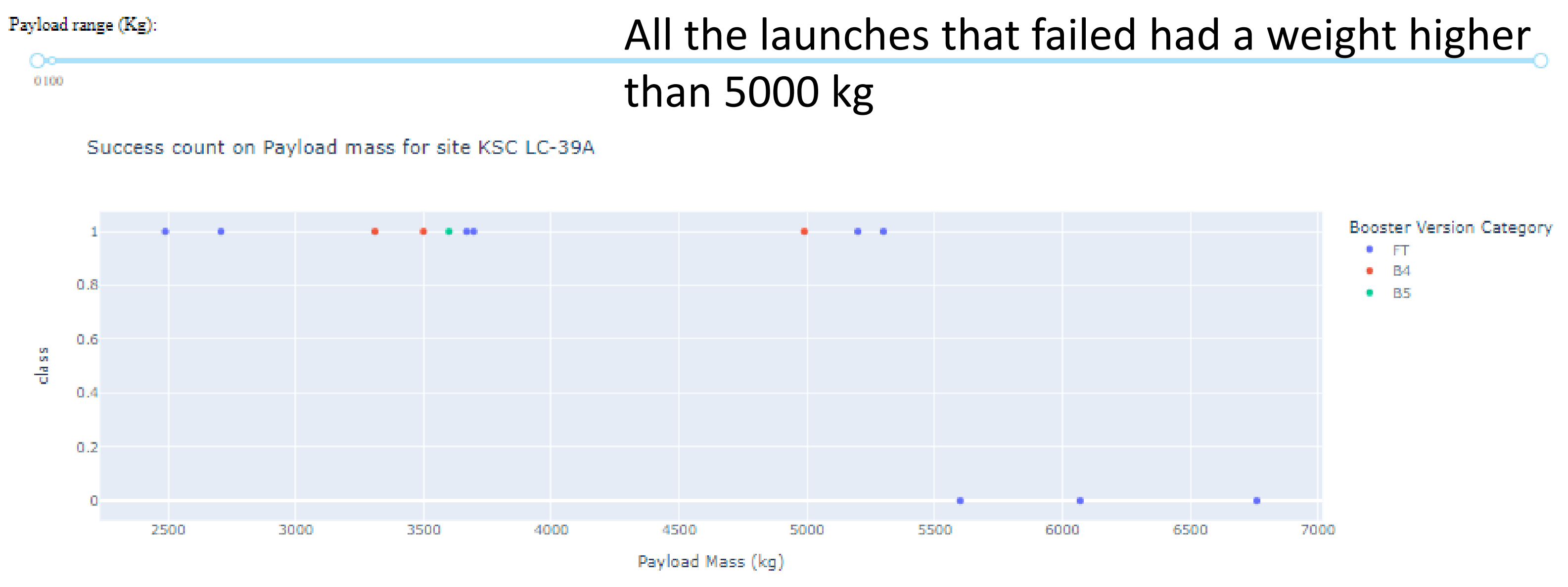
KSC LC-39A had the most successful launches



KSC LC-39A

Out of all the launches at KSC LC-39A 23% failed

- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

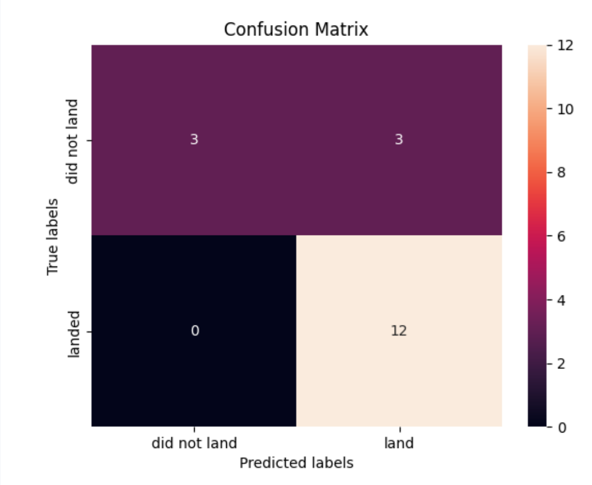


All the launches that failed had a weight higher than 5000 kg

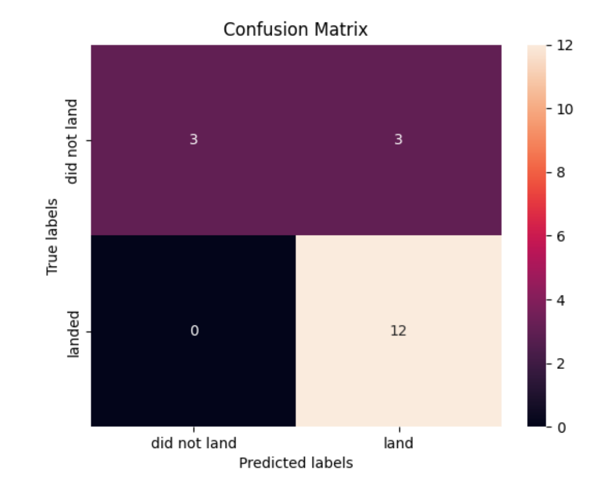
Predictive Analysis (Classification)

- All the machine learning models preformed similarly. With all the accuracies for each model being 83.34%
- Only outlier being the Decision Tree that is slightly different.
- https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

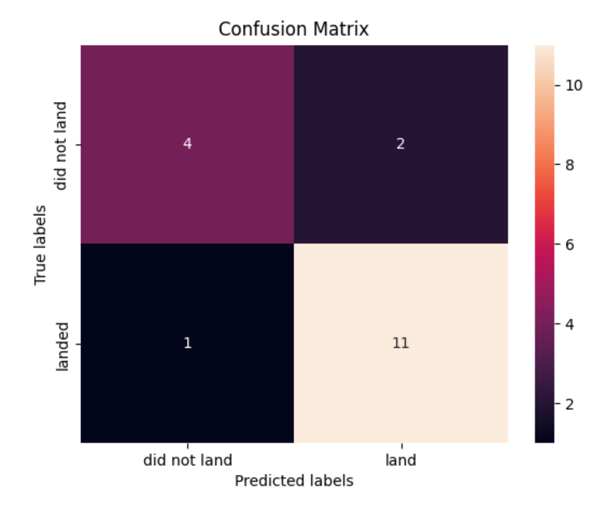
LogReg



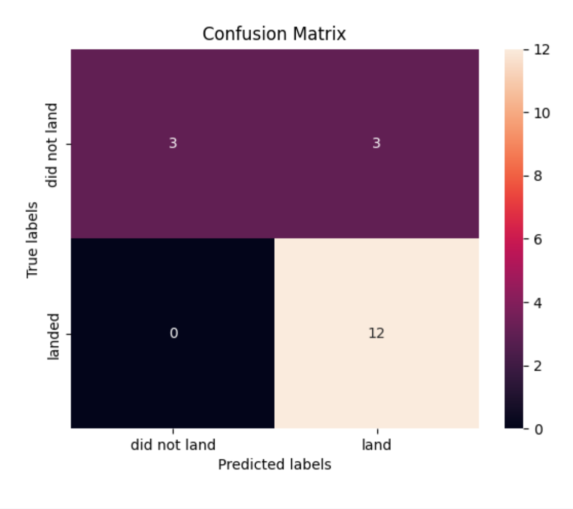
SVM



Decision Tree

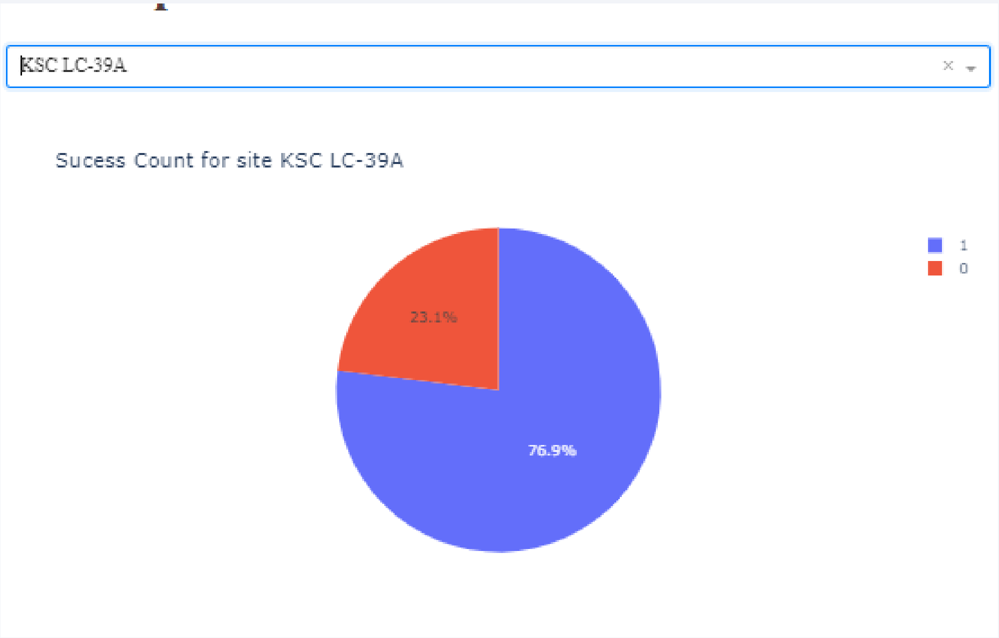


KNN



Results

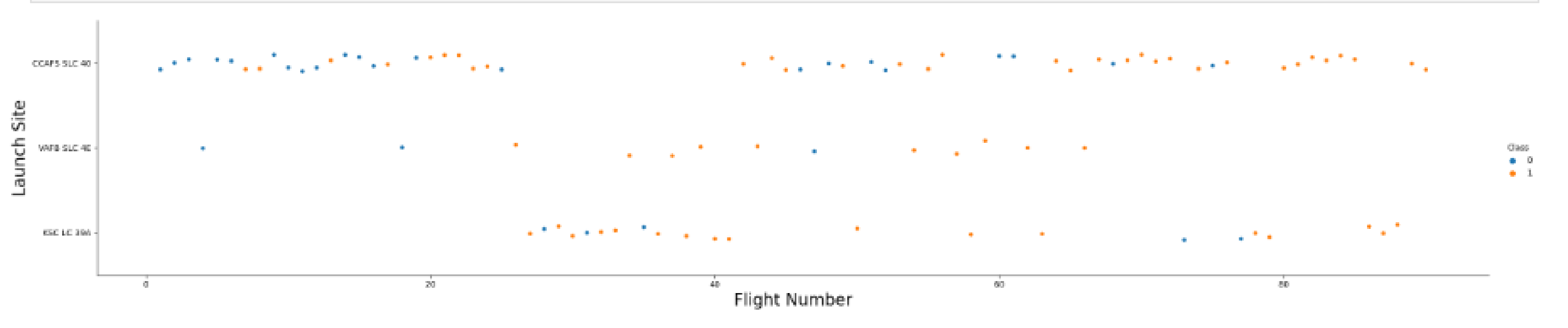
- According to our exploratory data analysis rockets with lower payload weights tend to be more successful than heavier attempts.
- As SpaceX continues to be successful in there missions the cost of space travel will continue to decrease since rockets are more reliable and renewable.
- The most successful launches were from the KSC LC-39A launch site with over 70% of there launches being successful.
- In terms of prediction, KNN, SVM, and Log Regression all preformed well with a decision tree being the worst out of the four.



Section 2

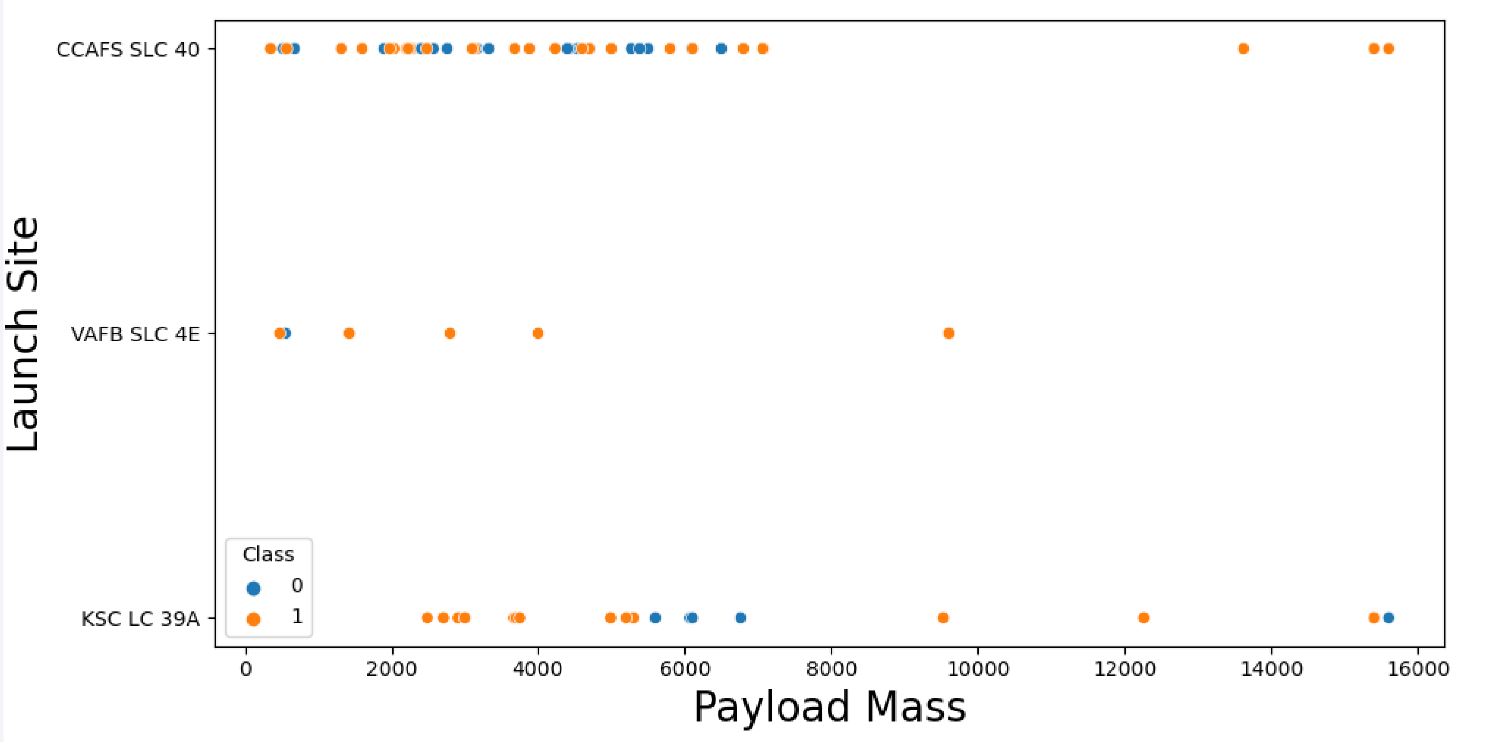
Insights drawn from EDA

Flight Number vs. Launch Site



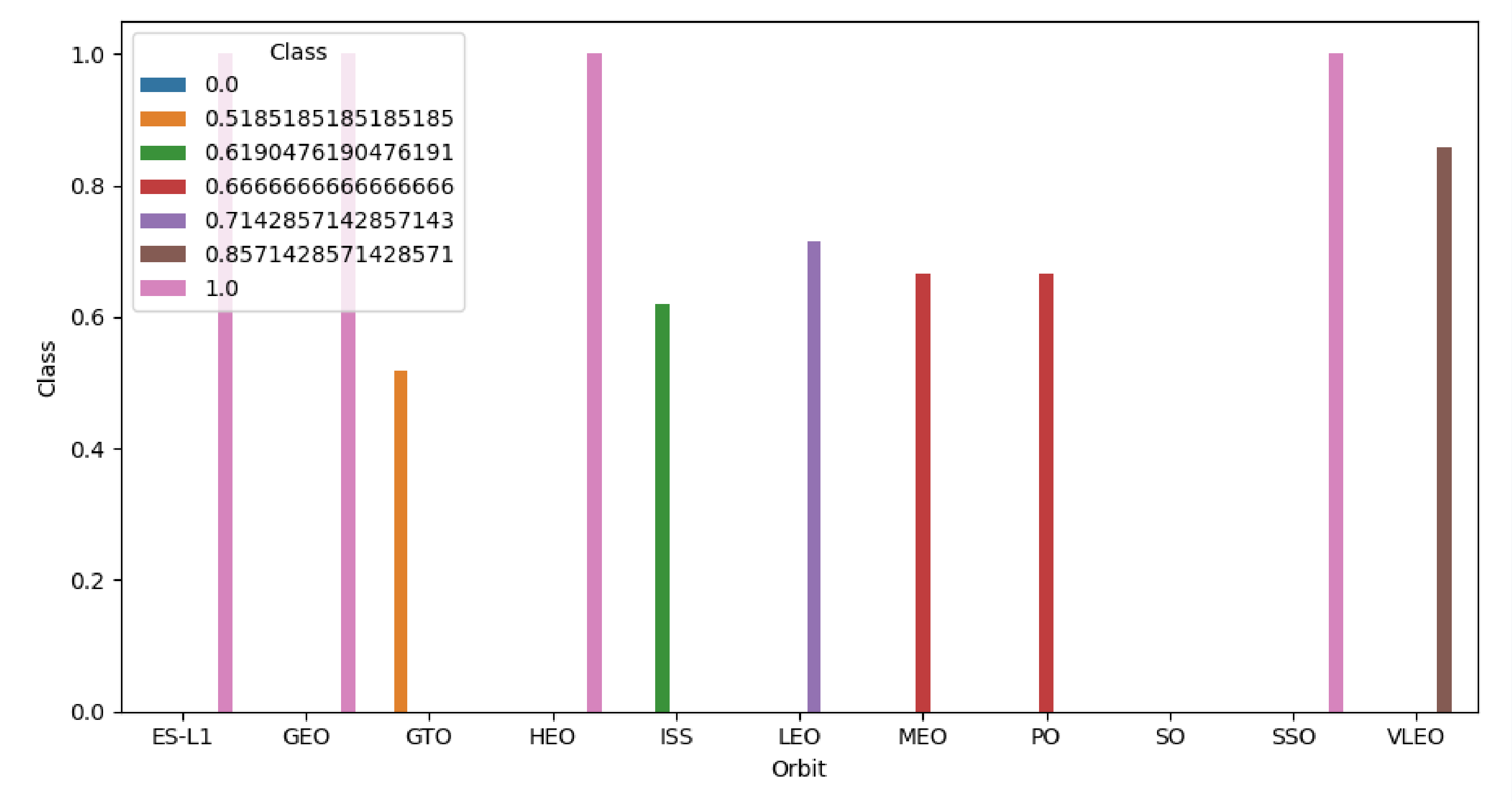
- There have been significantly more launches from CCAFS SLC 40 than any other site.
- With a gap in flight numbers between 25 and 40.

Payload vs. Launch Site



- CCAFS SLC 40 launched with a majority of there payload weight under 8,000 kg.
- While the other two launch sites have more diverse launch weights.

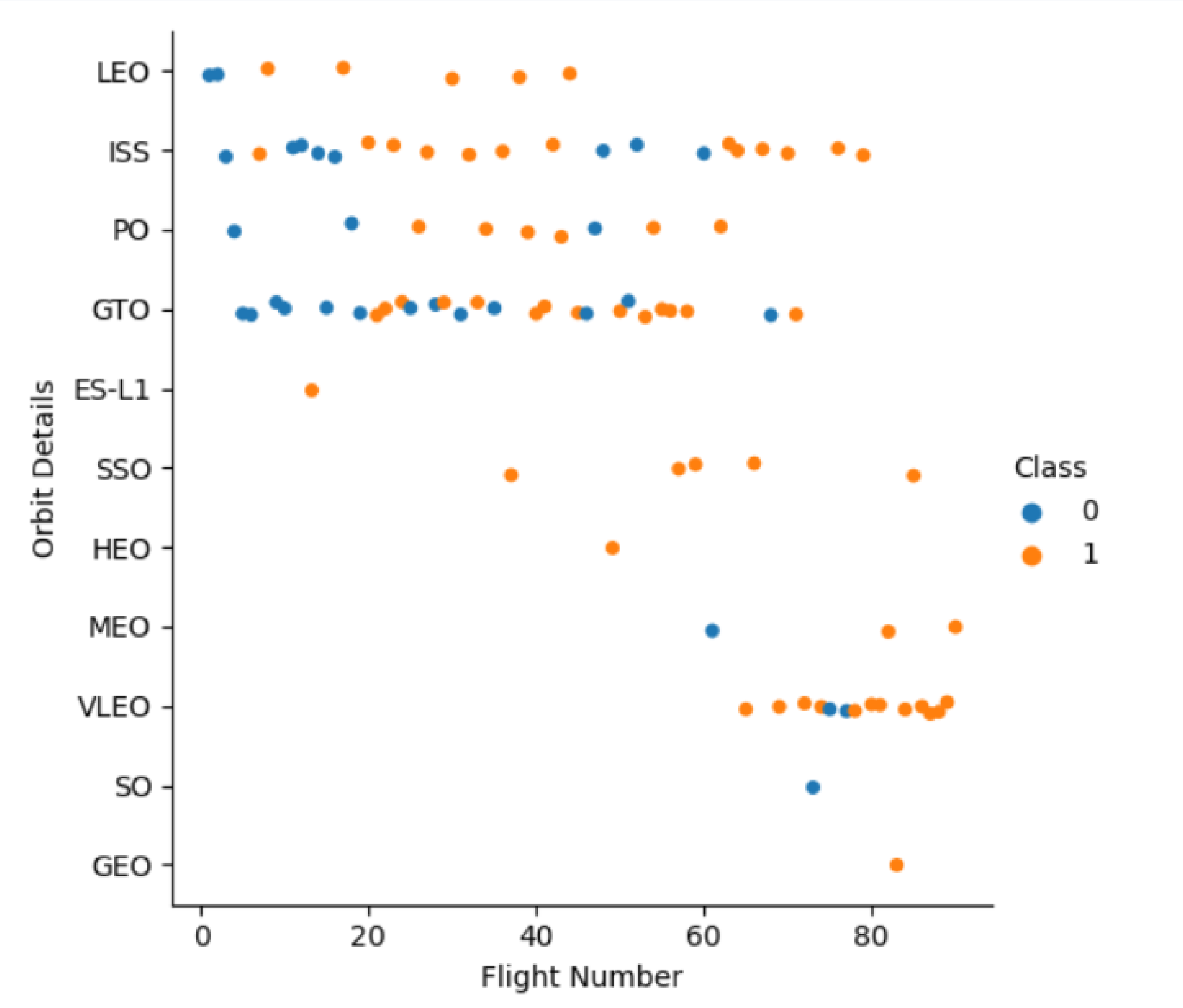
Success Rate vs. Orbit Type



- ES-L1, GEO, HEO, and SSO orbit had the highest average success across the board.
- While GTO had the lowest success average.

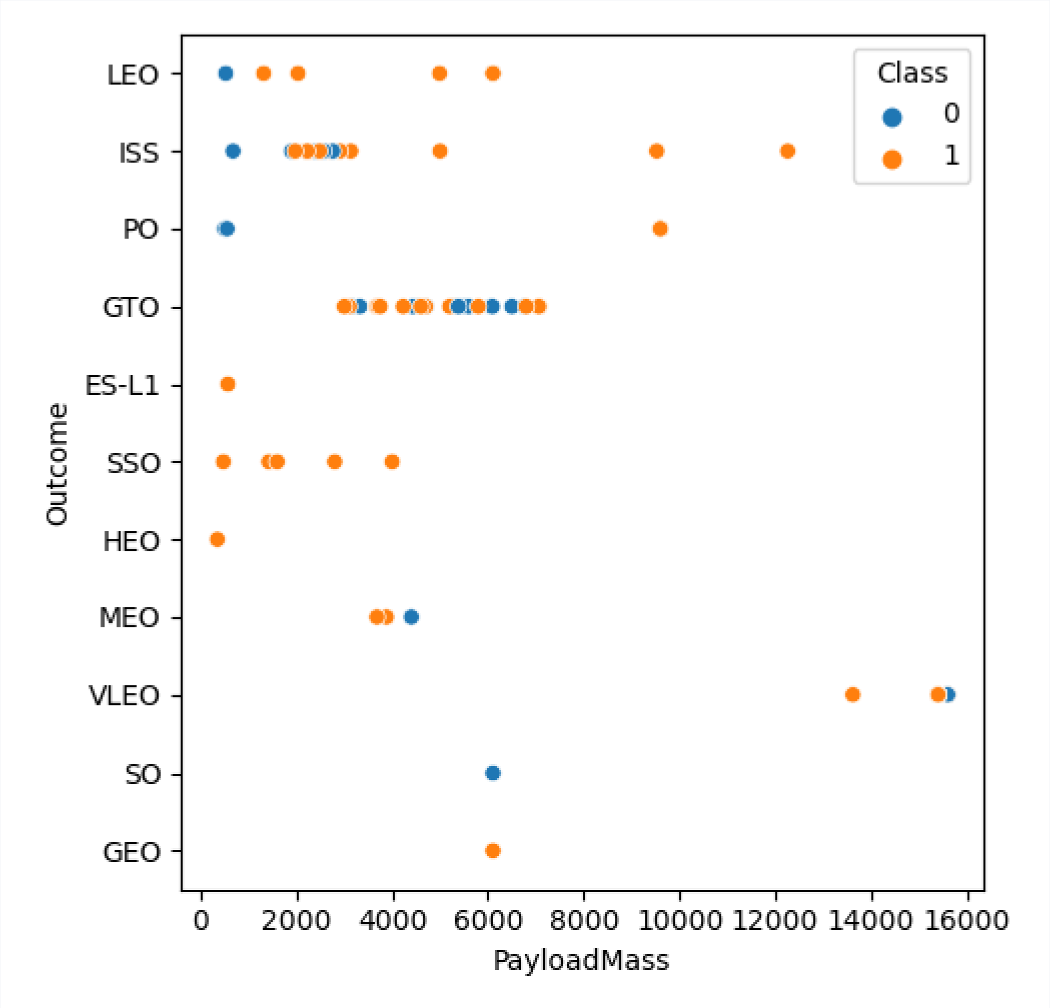
Flight Number vs. Orbit Type

- GEO, SO, and VELO all have flight numbers greater than 60.
- LEO, ISS, PO, and GTO all have flight numbers ranging from 0-80



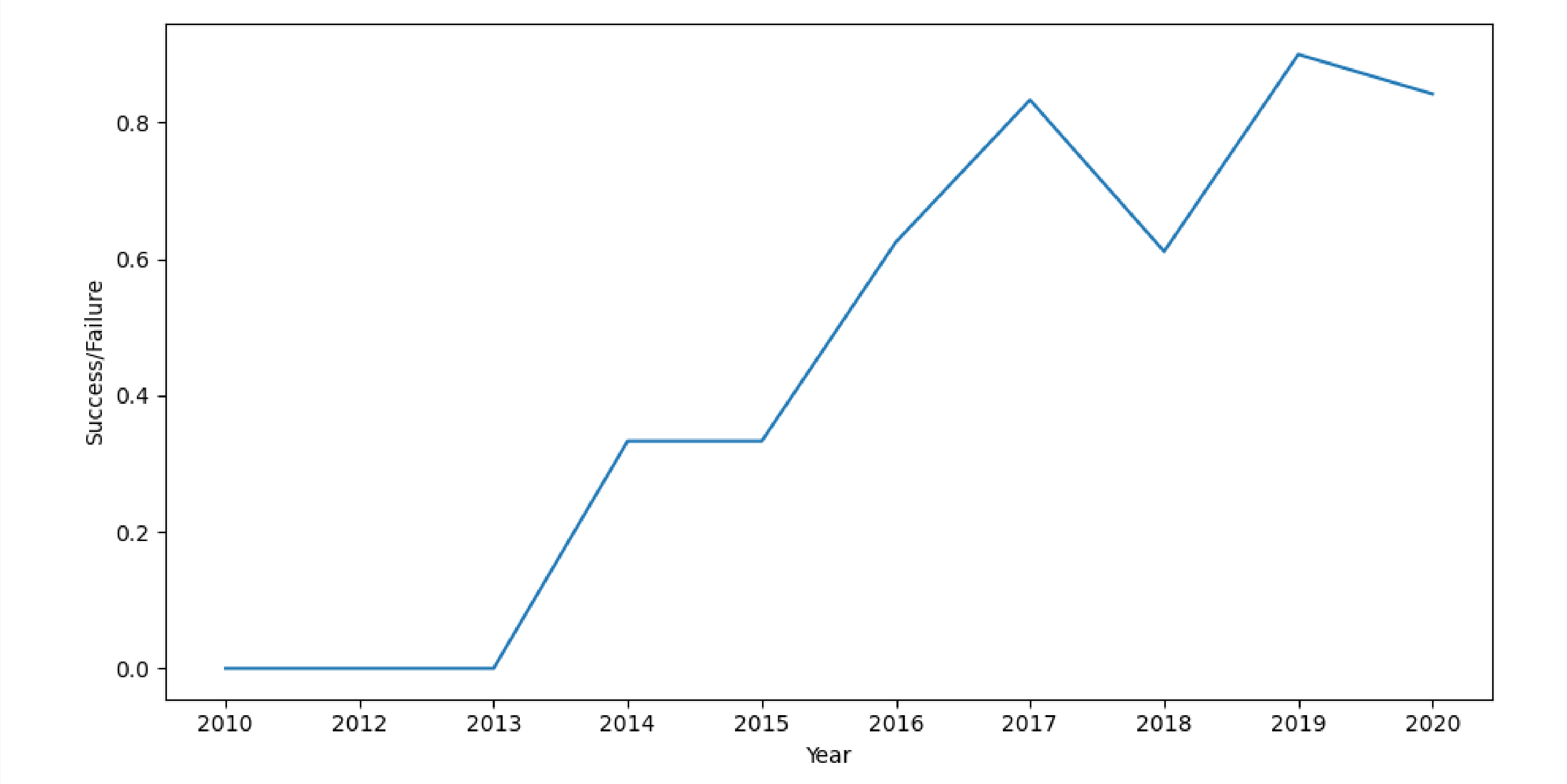
Payload vs. Orbit Type

- There is a correlation between GTO orbits and the payload mass being between 3,500 - 8,000 kgs.
- There is also a correlation between and ISS orbit and a payload mass between 2,000 and 4,000 kgs.



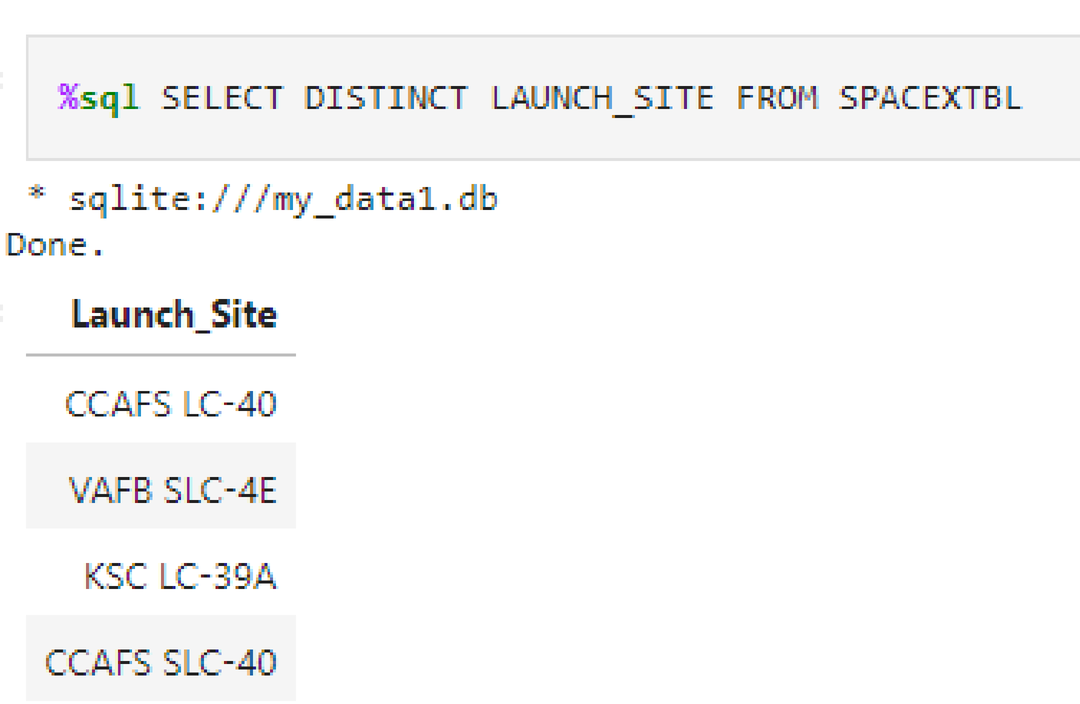
Launch Success Yearly Trend

- Over the course of 10 years launch success rates have increased and continue to trend in that direction.



All Launch Site Names

- Find the names of the unique launch sites



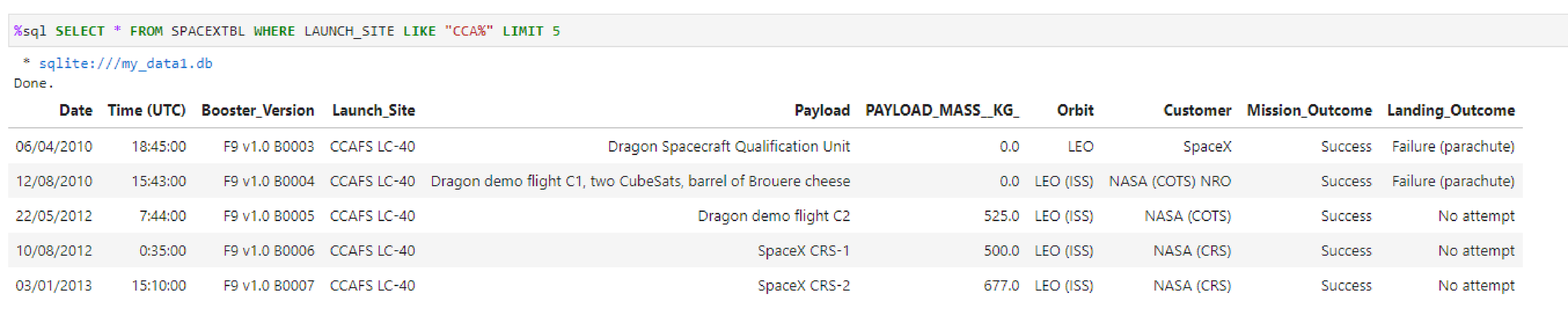
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

```
 * 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'

- Find 5 records where launch sites begin with `CCA`



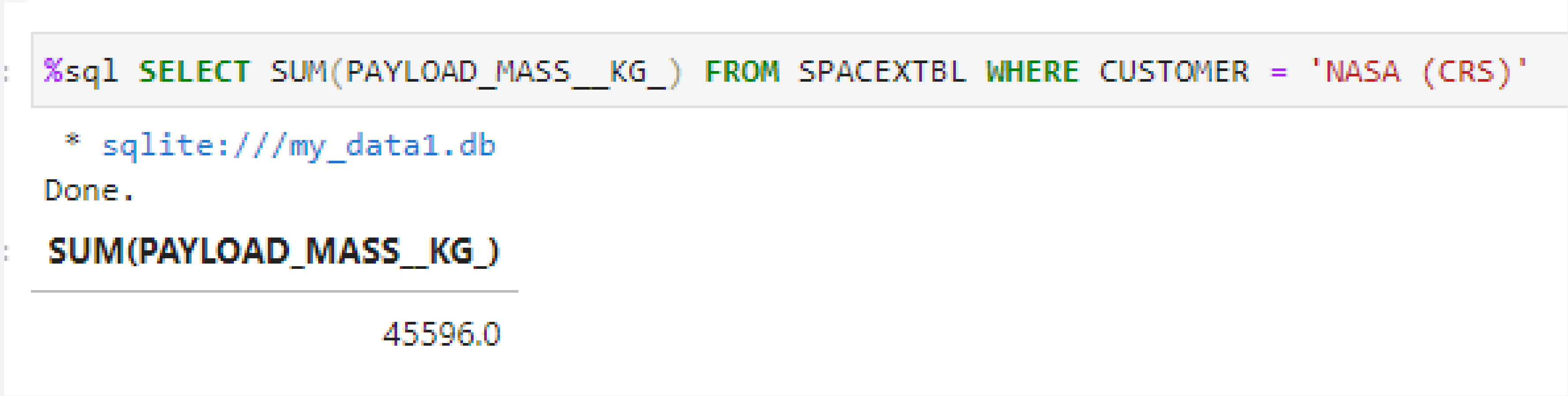
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE "CCA%" LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



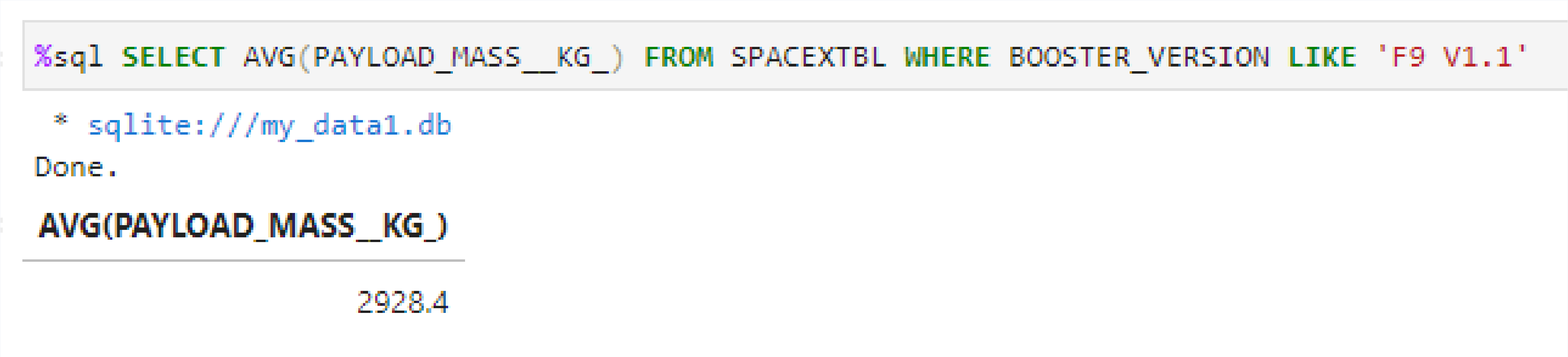
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1



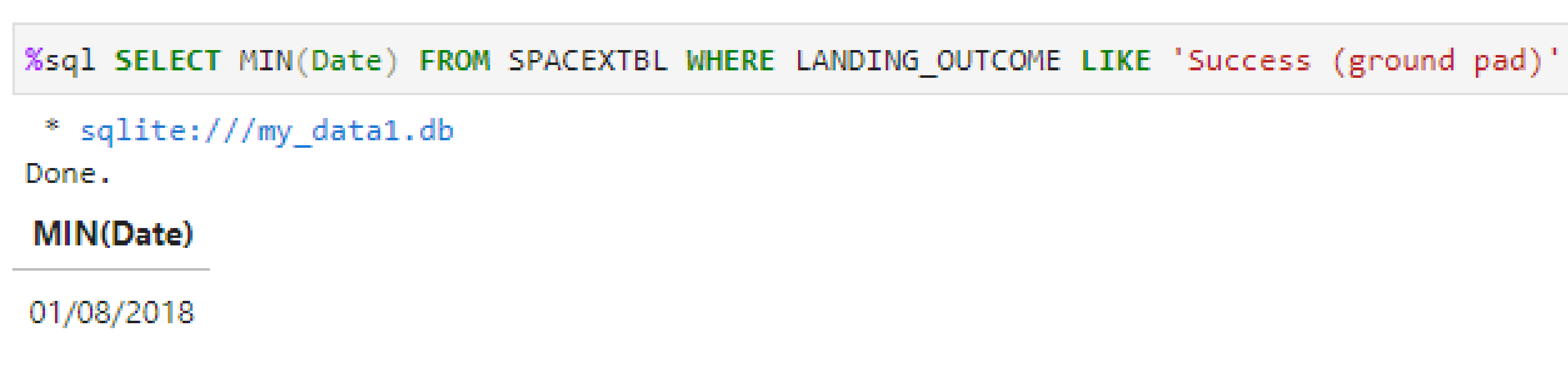
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 V1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



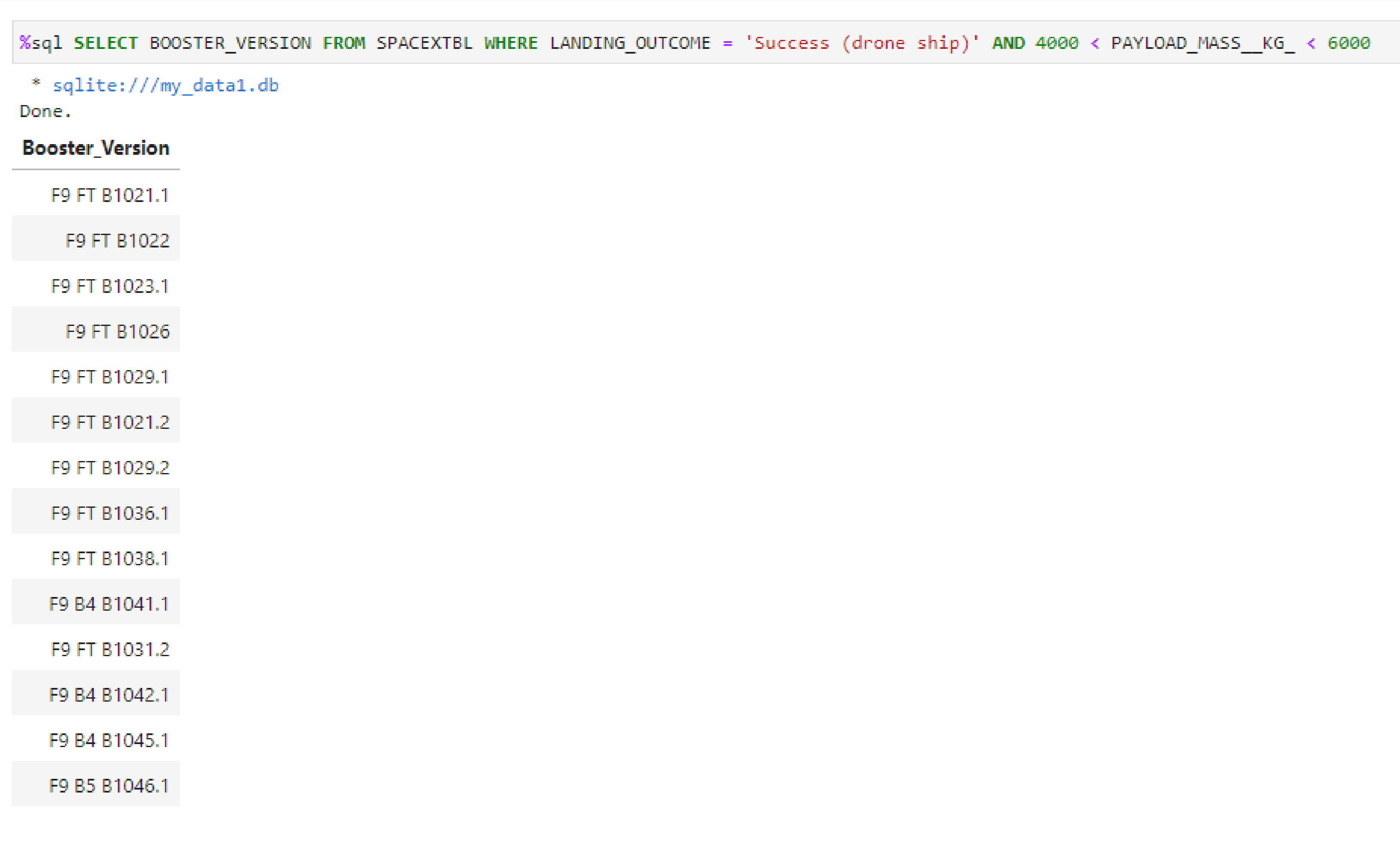
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Success (ground pad)'
```

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

MIN(Date)
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



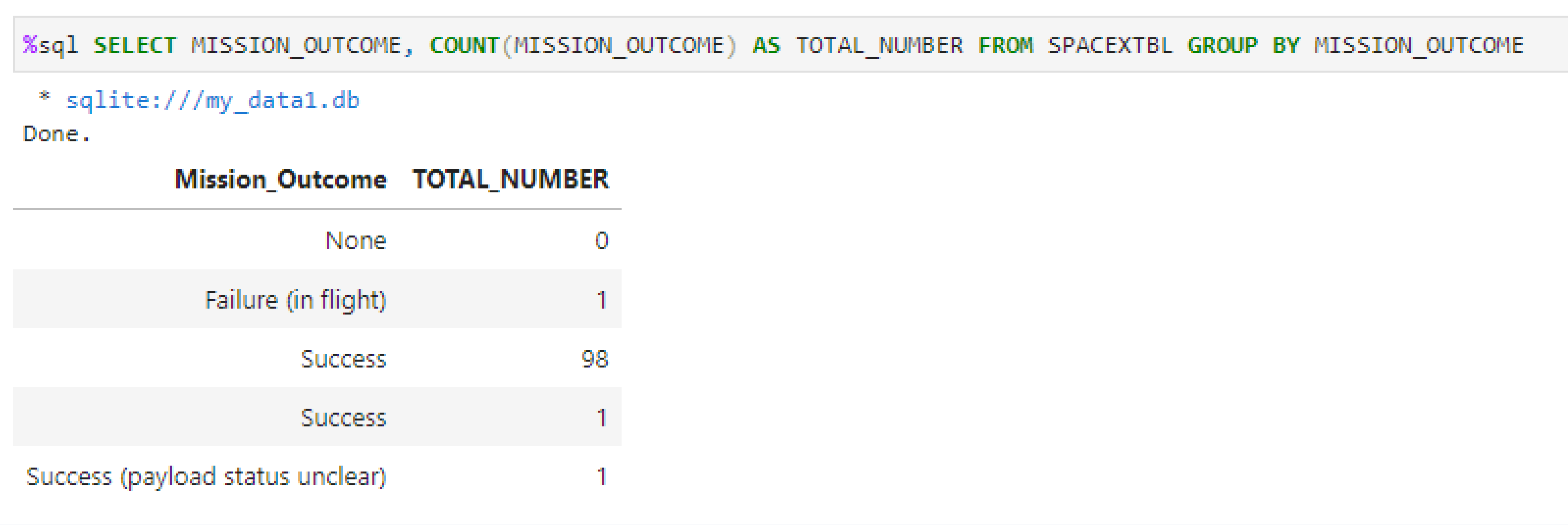
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND 4000 < PAYLOAD_MASS__KG_ < 6000
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



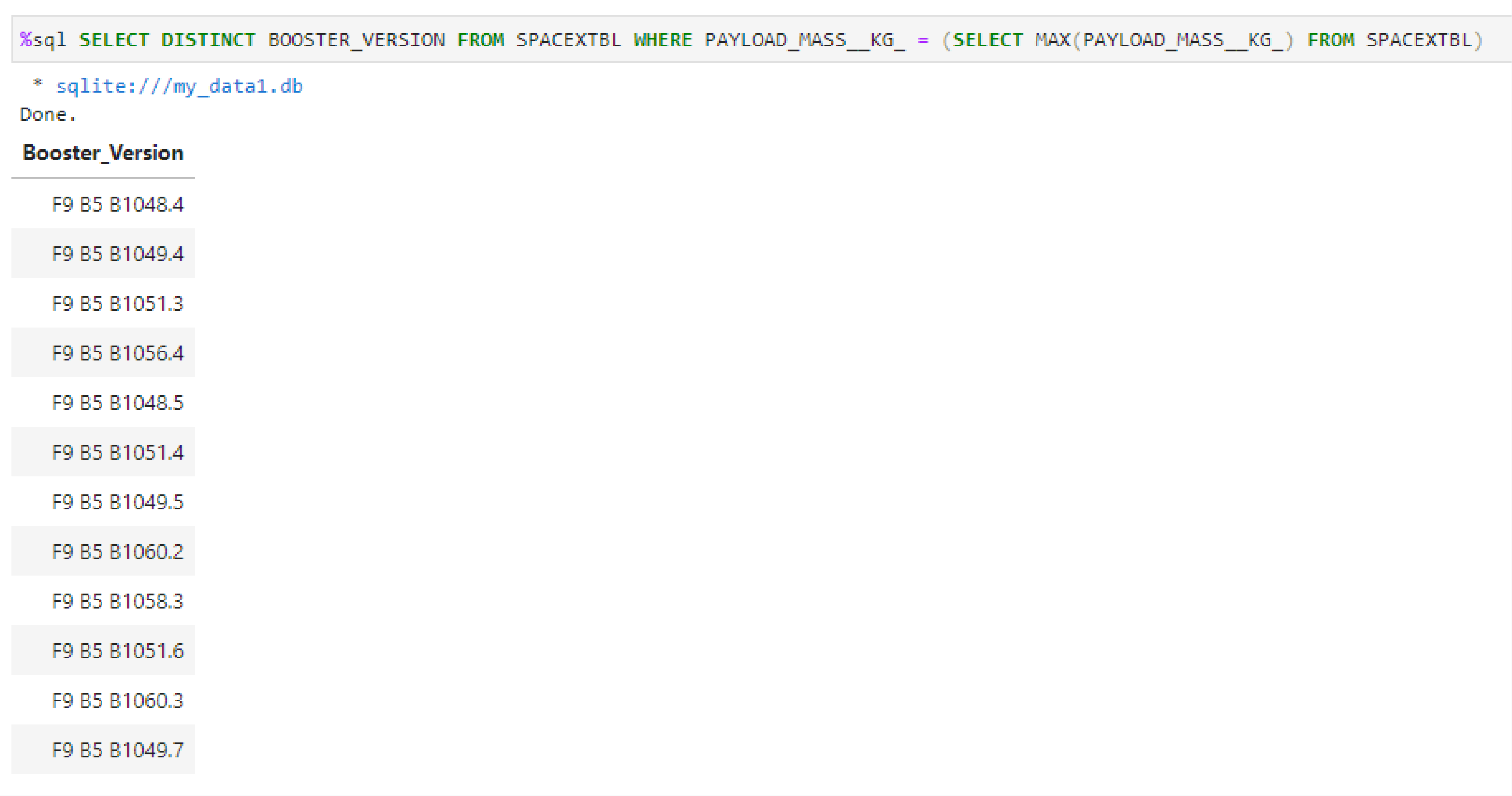
```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

Mission_Outcome	TOTAL_NUMBER
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



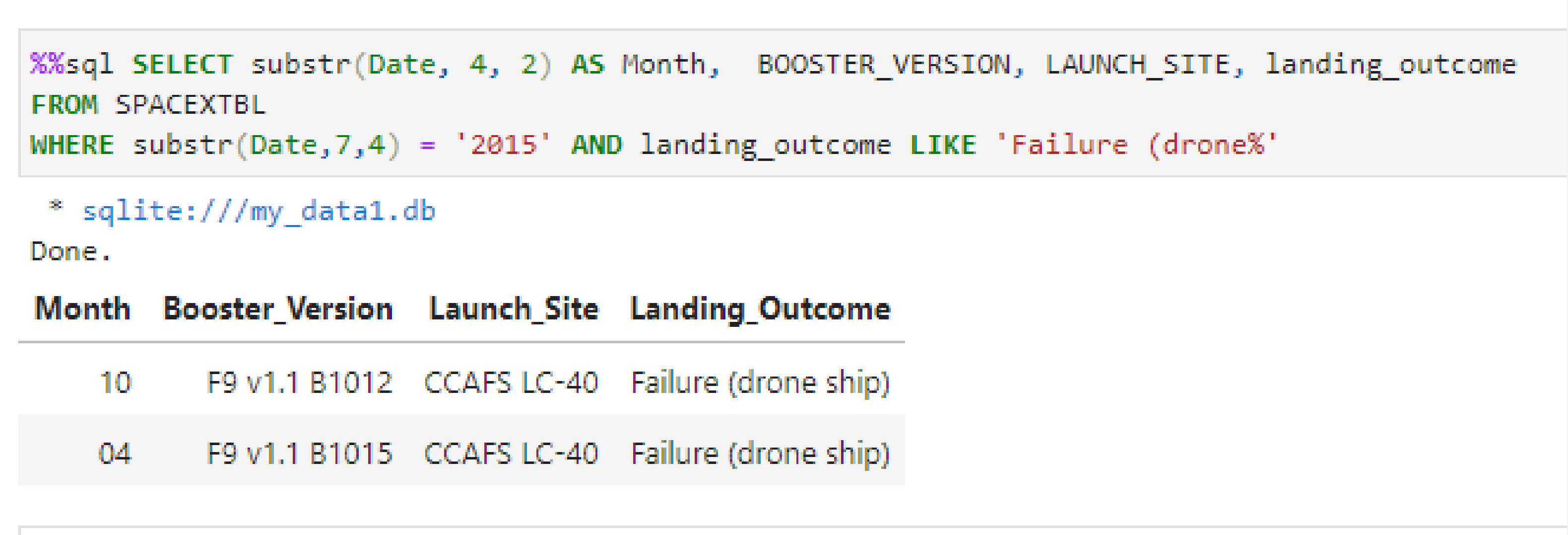
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



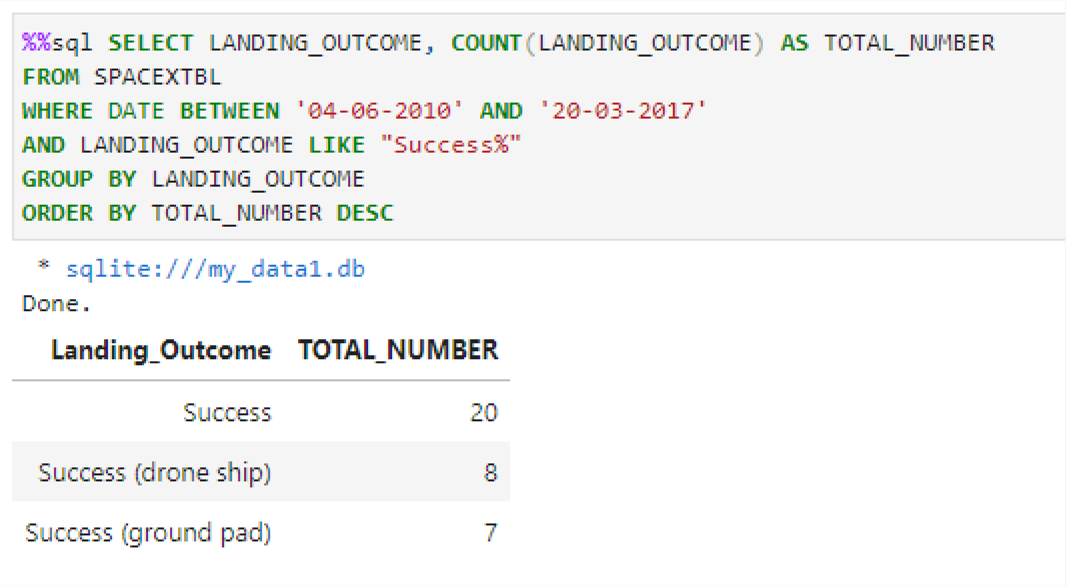
```
%%sql SELECT substr(Date, 4, 2) AS Month,  BOOSTER_VERSION, LAUNCH_SITE, landing_outcome
FROM SPACEXTBL
WHERE substr(Date,7,4) = '2015' AND landing_outcome LIKE 'Failure (drone%'
```

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

Month	Booster_Version	Launch_Site	Landing_Outcome
10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

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



```
%%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
AND LANDING_OUTCOME LIKE "Success%"
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

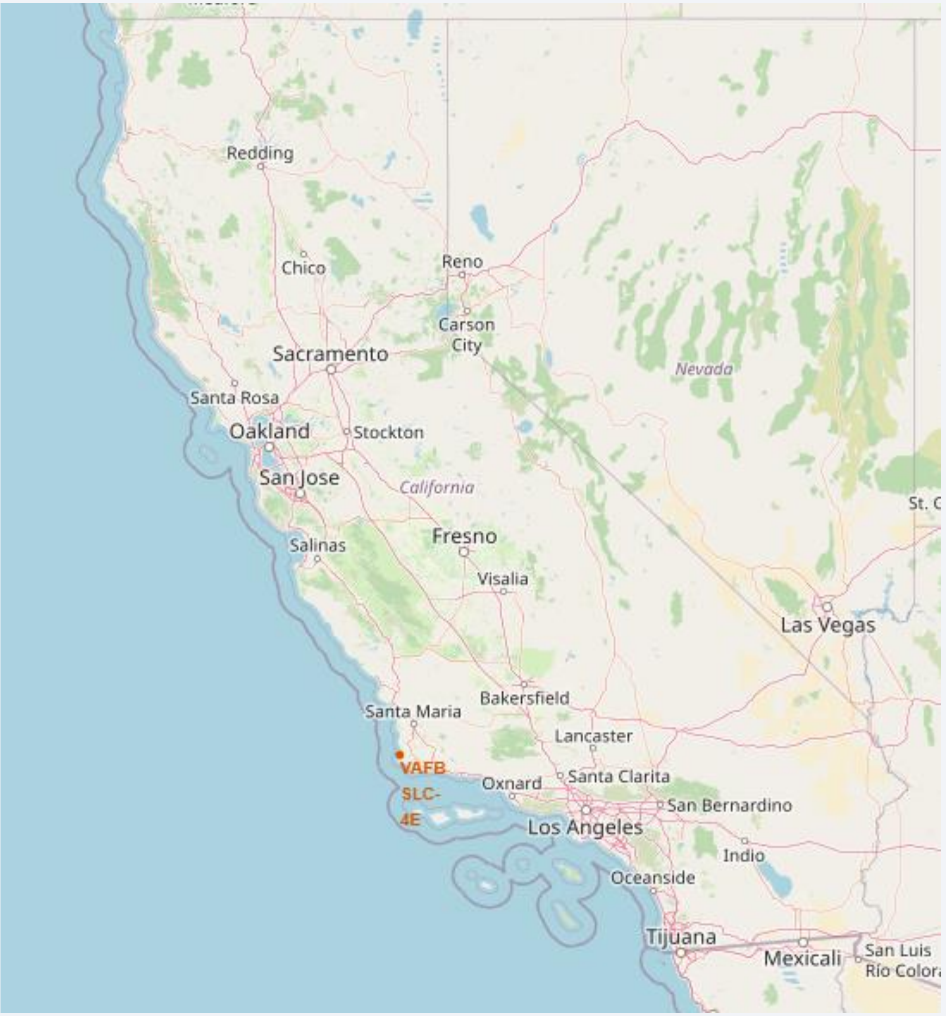
```
 * sqlite:///my_data1.db
Done.
```

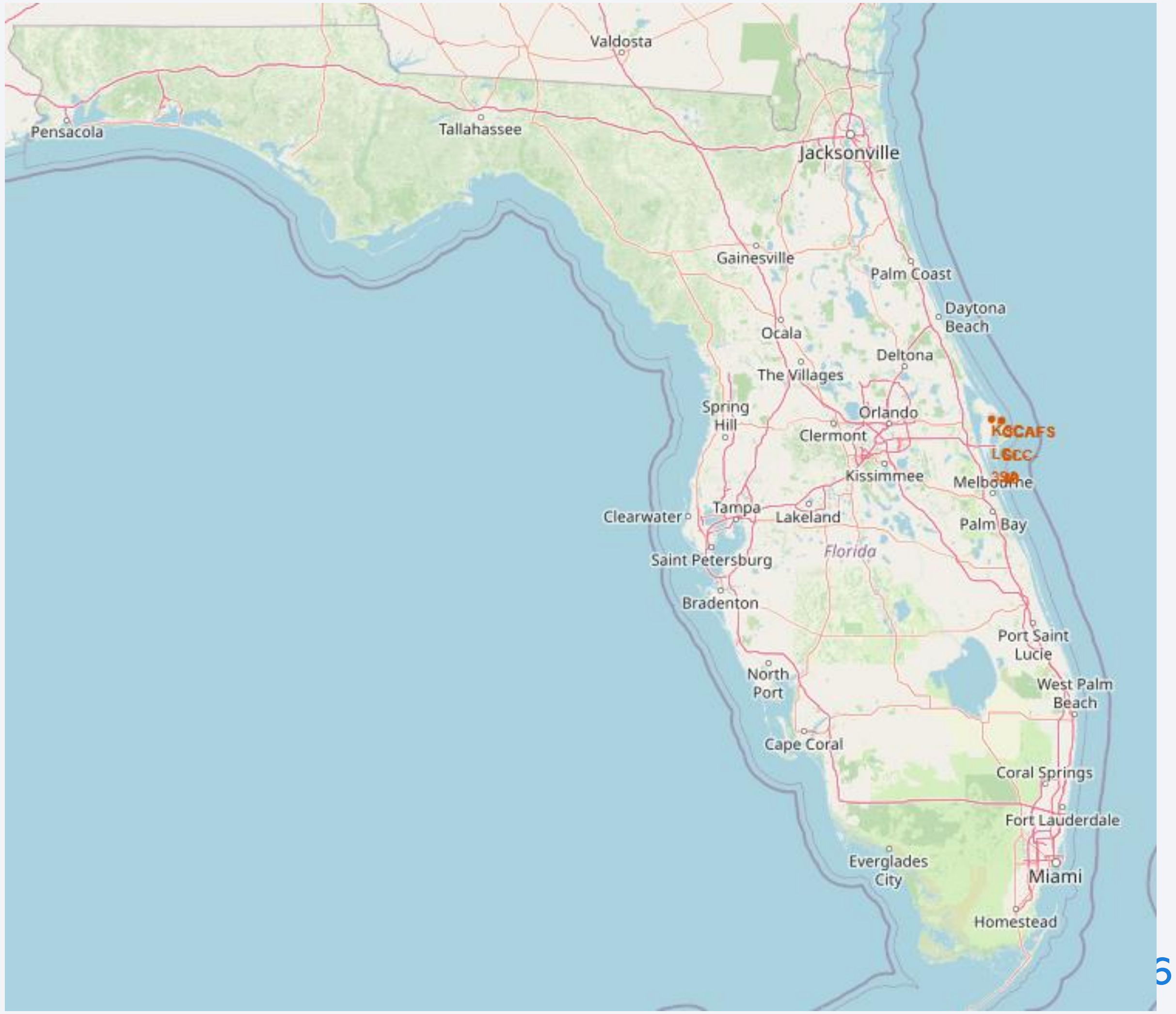
Landing_Outcome	TOTAL_NUMBER
Success	20
Success (drone ship)	8
Success (ground pad)	7

Section 3

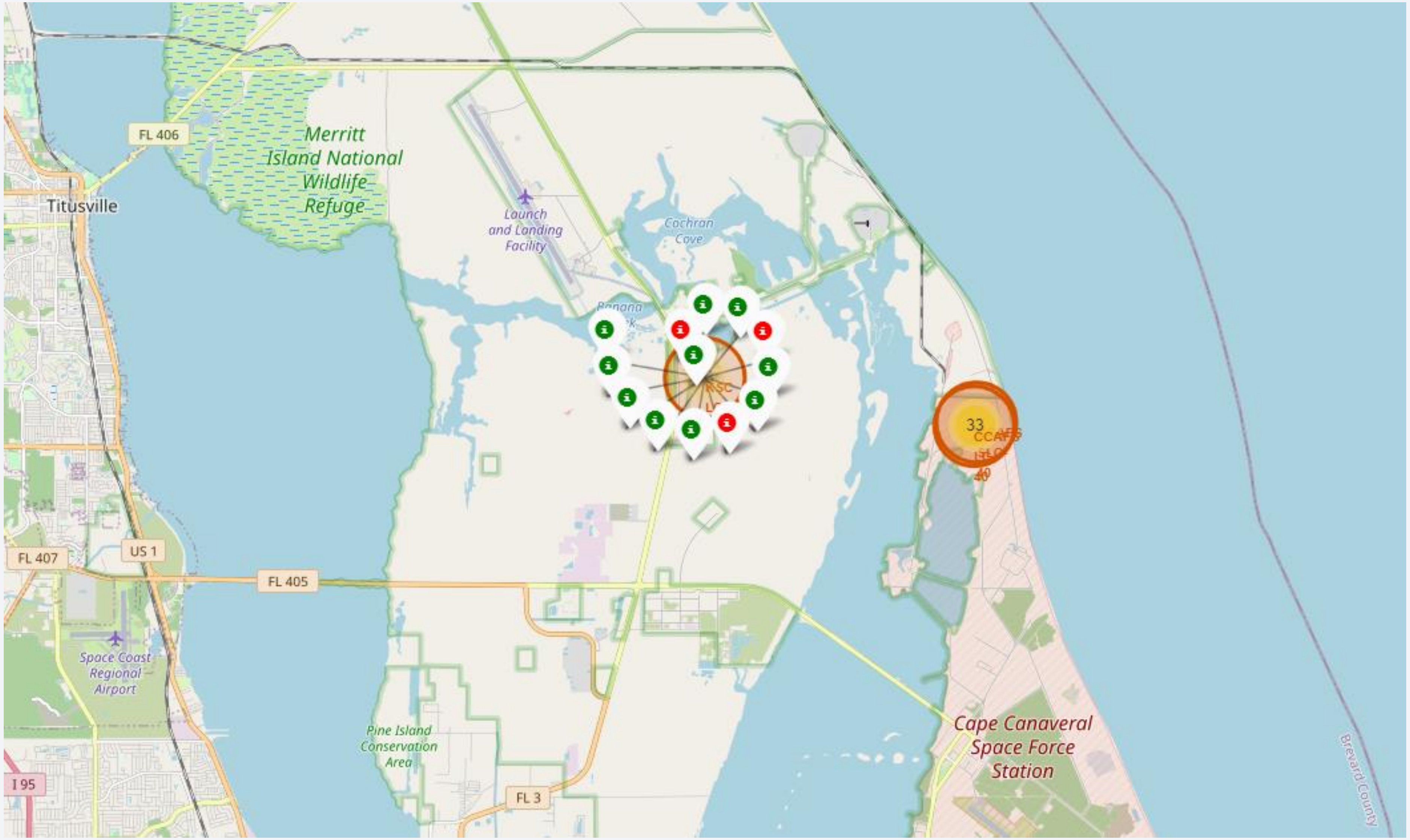
Launch Sites Proximities Analysis

All Launch Site Location on a Global Map

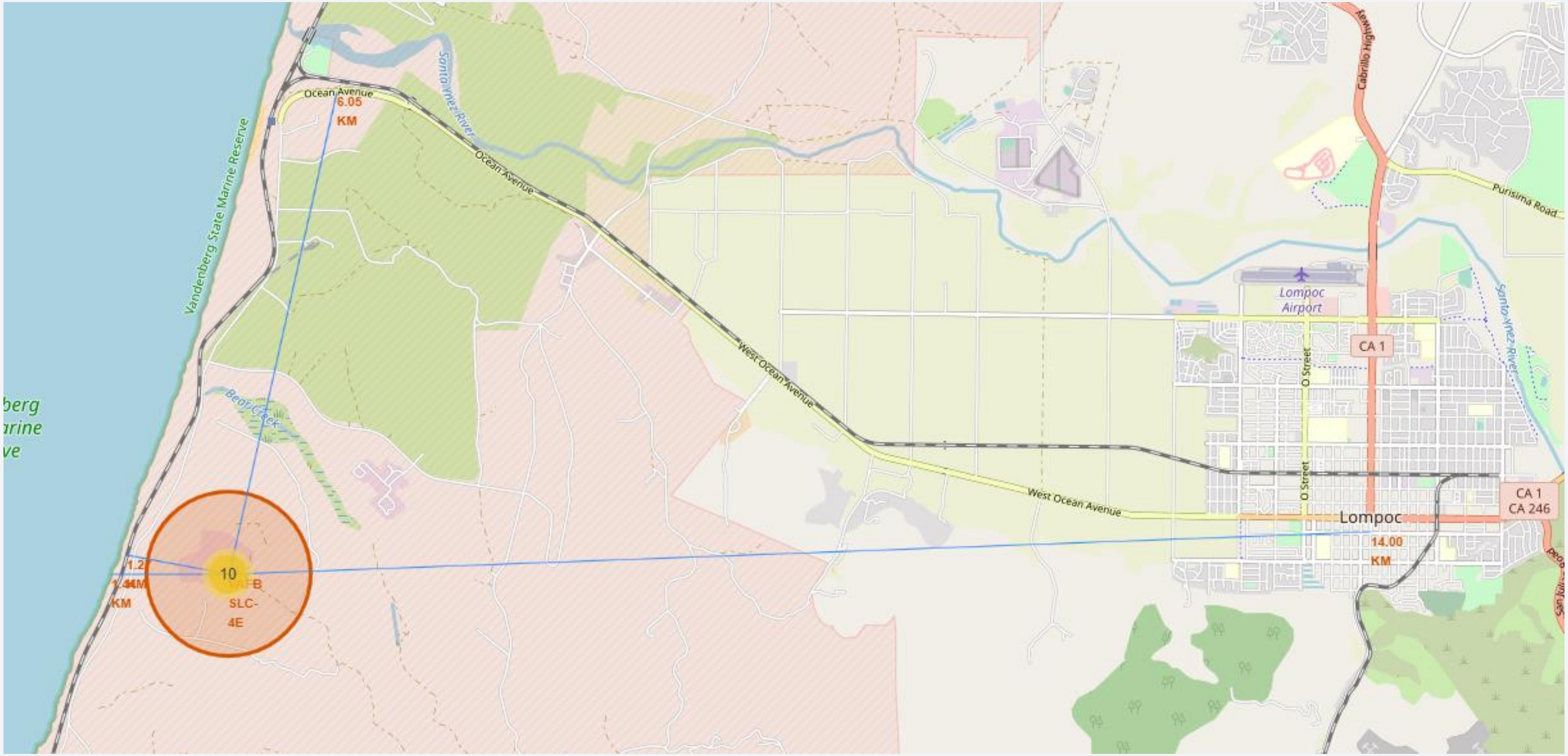


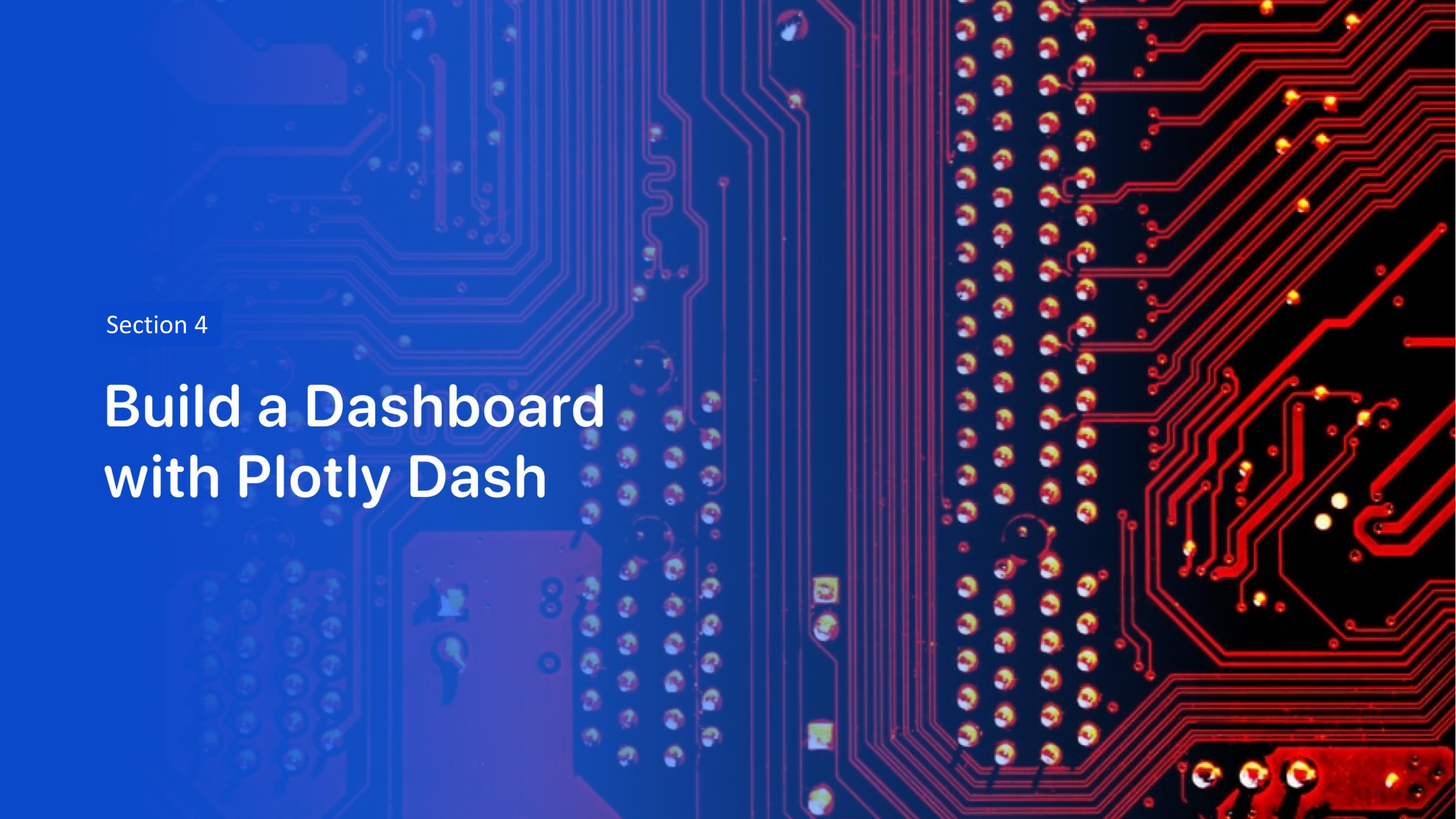


Color-Labeled Launch Outcomes on the Map


FL 406
Merritt Island National Wildlife Refuge
Titusville
Launch and Landing Facility
Cochran Cove
Banana Creek
KSC LC
33
CCAFS LC 40
CCAFS SLC 40
FL 407
US 1
FL 405
Space Coast Regional Airport
Pine Island Conservation Area
I 95
FL 3
Cape Canaveral Space Force Station
Brevard County

Launch Site Proximity to Railway, Highway, Coastline, with Distance Calculated and Displayed

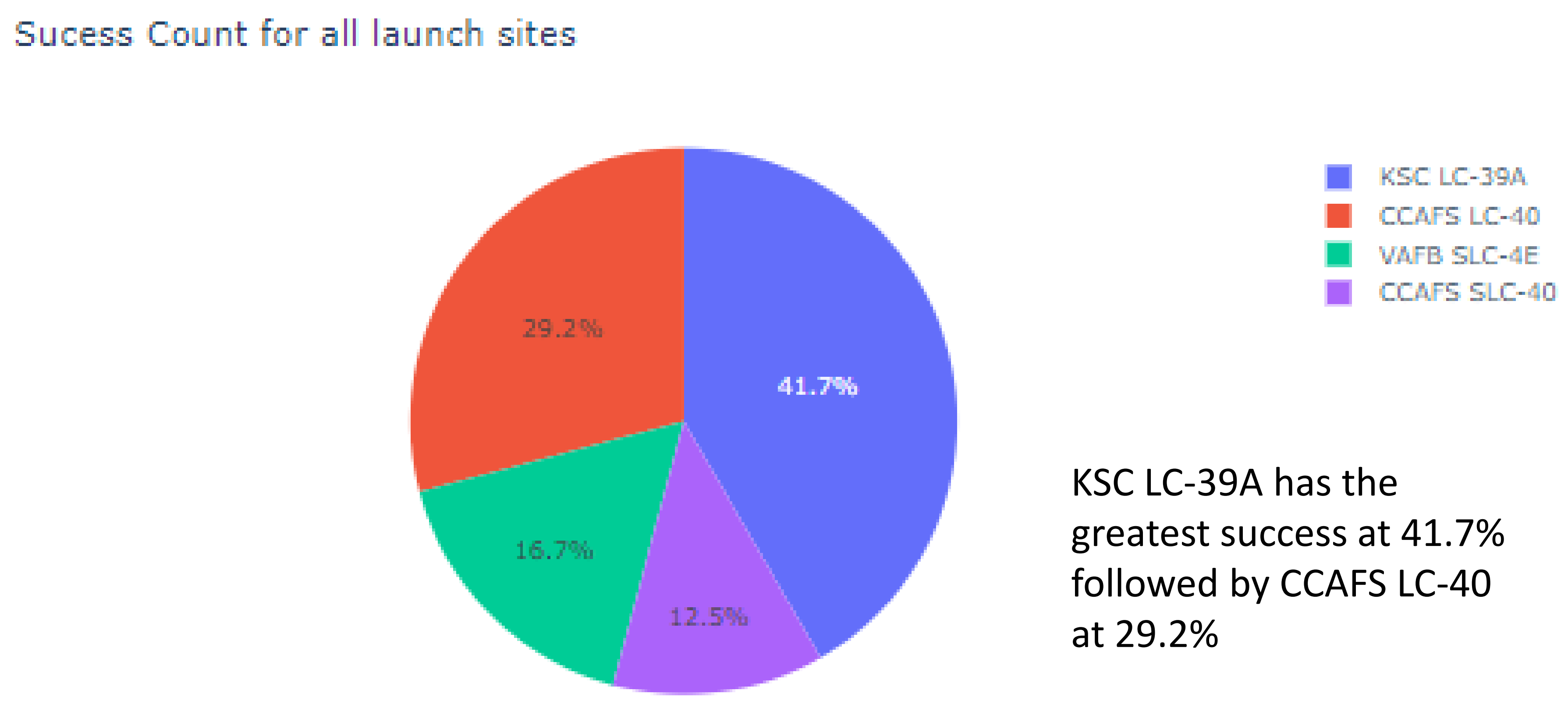




Section 4

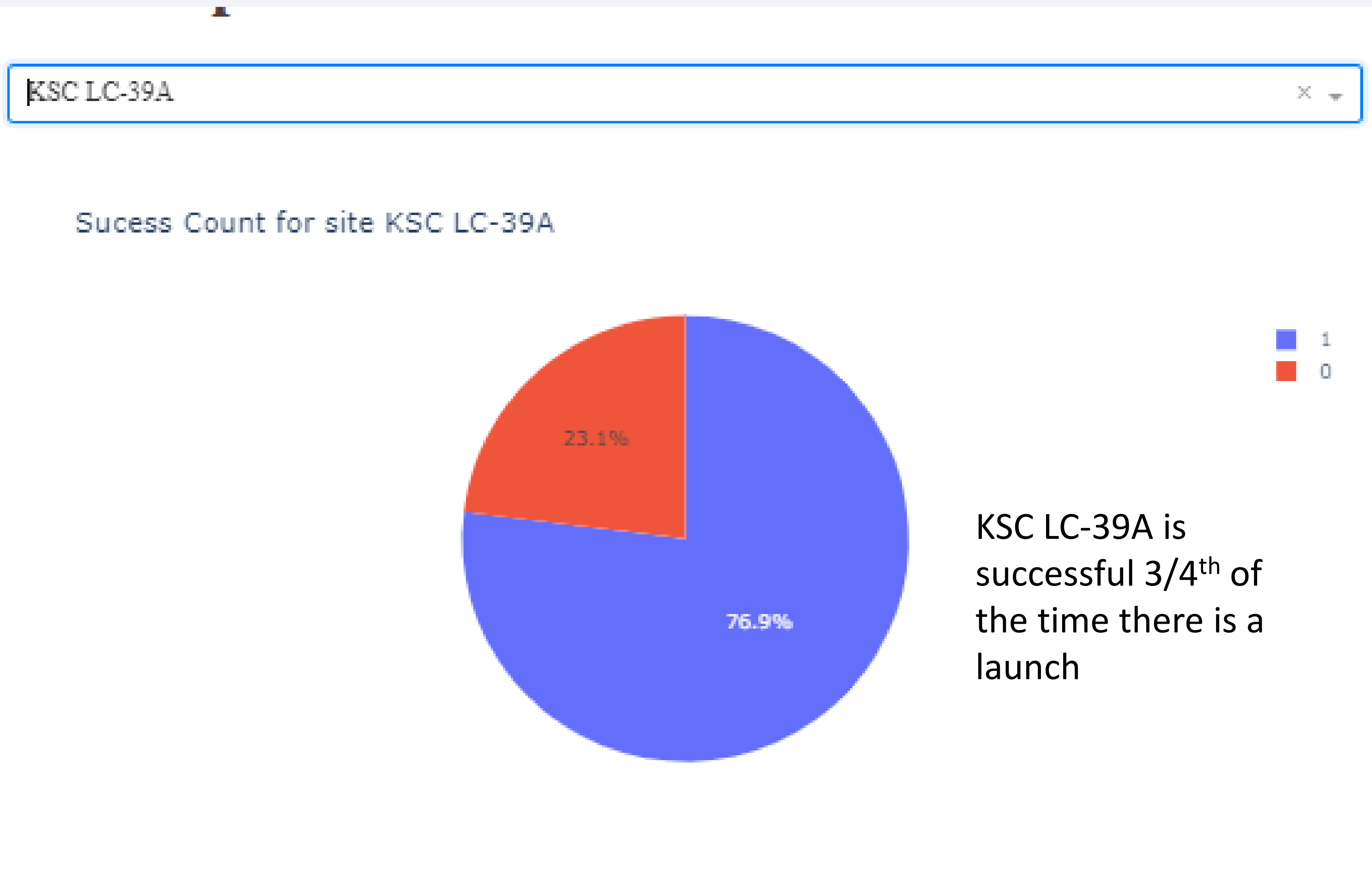
Build a Dashboard with Plotly Dash

Launch Success Percentage for all Sites



KSC LC-39A has the greatest success at 41.7% followed by CCAFS LC-40 at 29.2%

Launch Site with the Highest Success Percentage

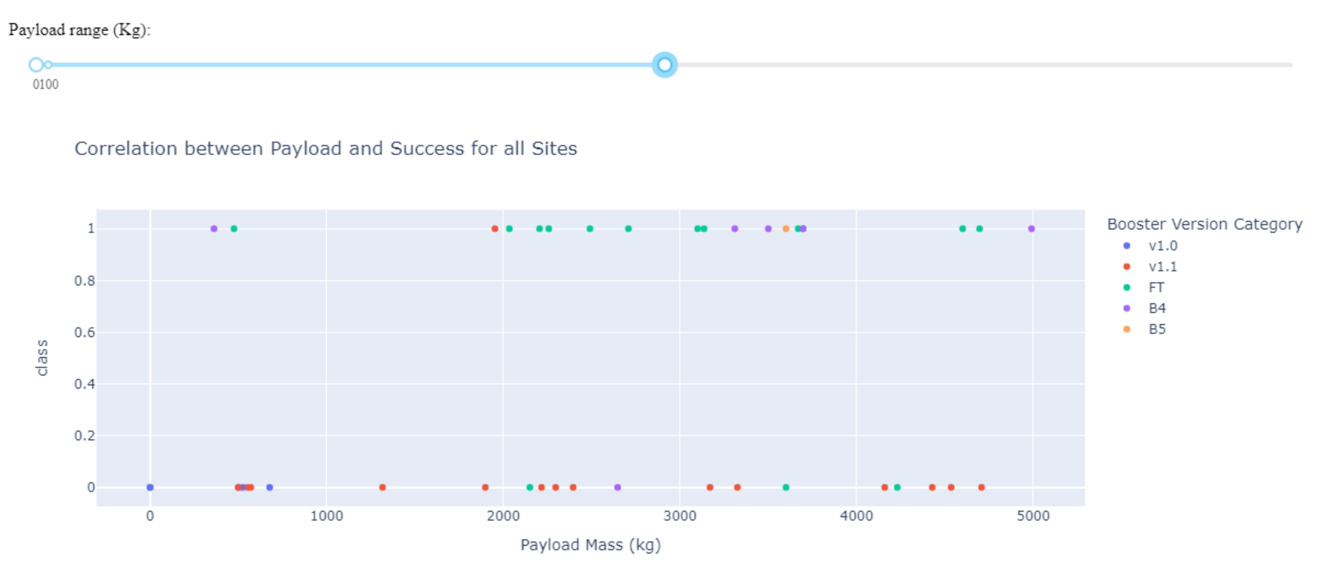


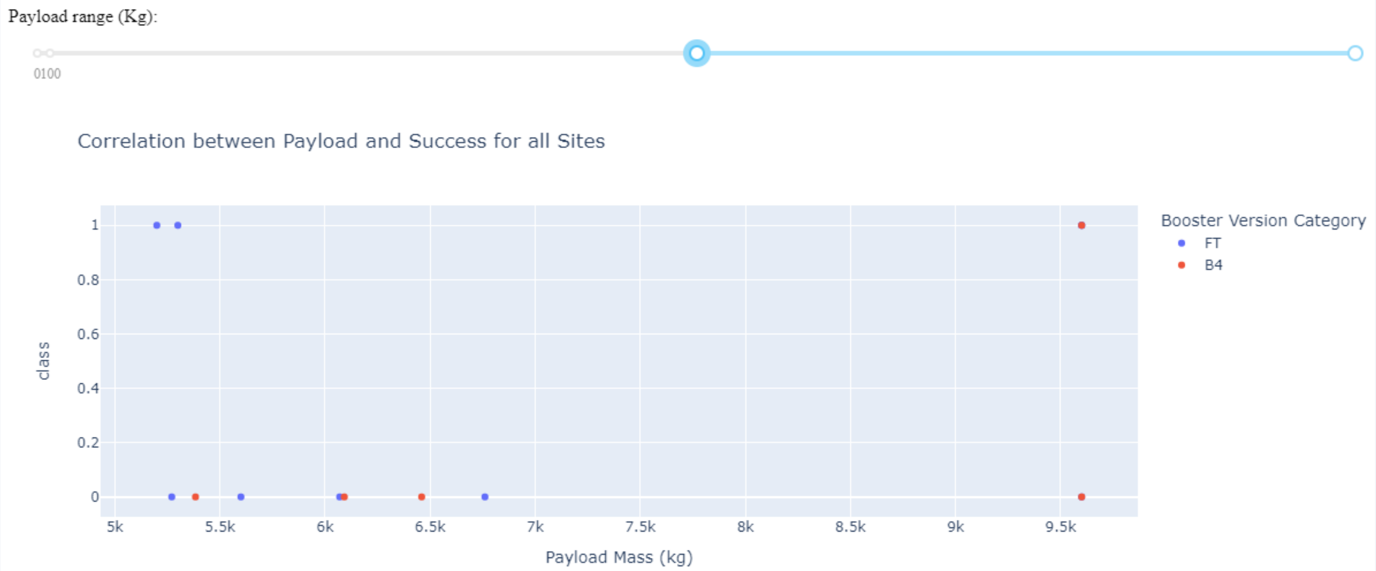
KSC LC-39A

KSC LC-39A is successful 3/4th of the time there is a launch

Payload Mass (kg) vs. Launch Outcome

There is an even split of failure and success across all sites from payloads between 0 and 5,000 kgs.





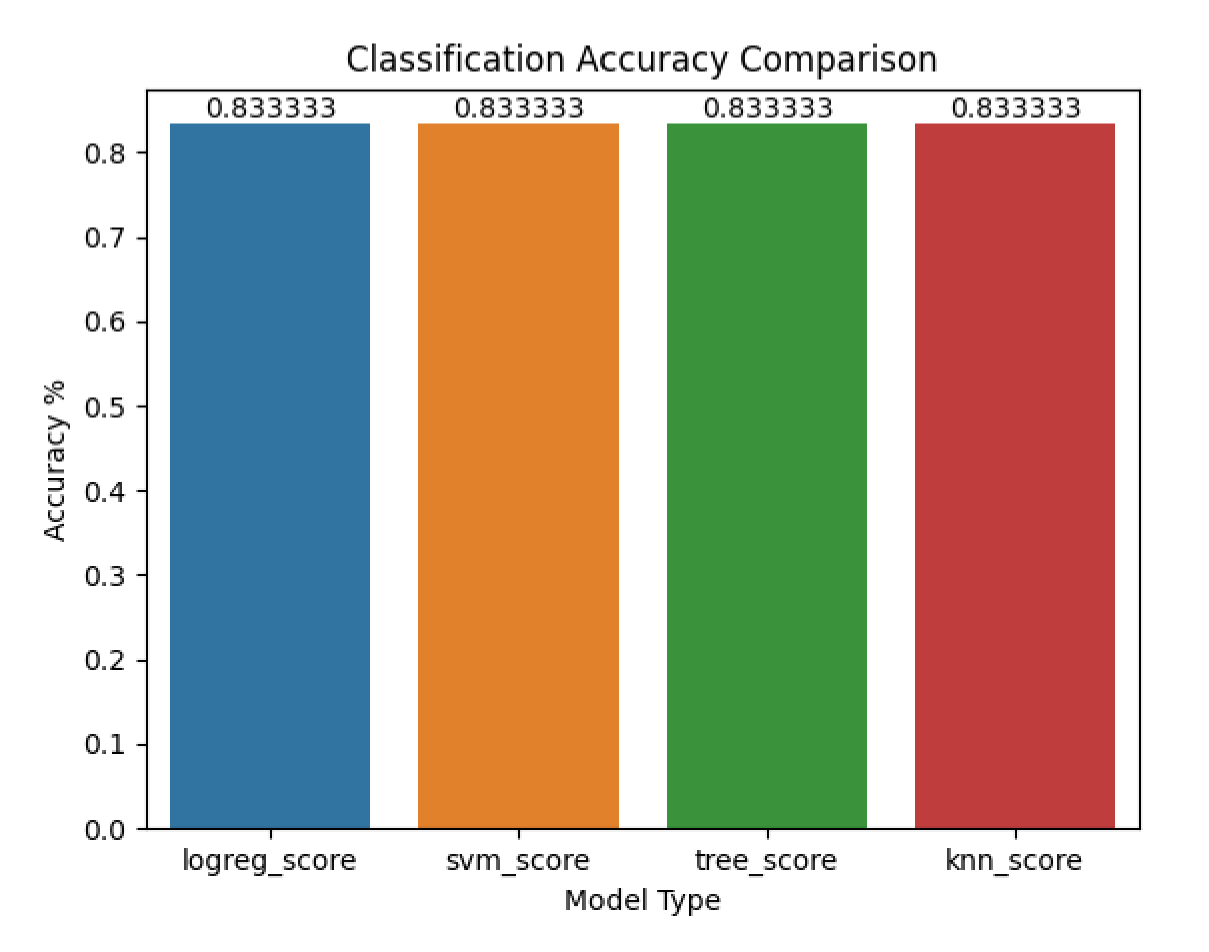
However, the number of failures greatly out weight the successes once the weight begins to exceed 5,000 kgs

Section 5

Predictive Analysis (Classification)

Classification Accuracy

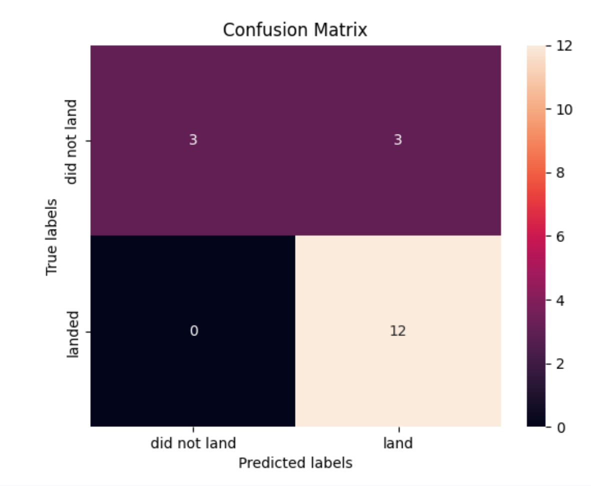
- All models have an equal accuracy in terms of determining success or failure of a landing.



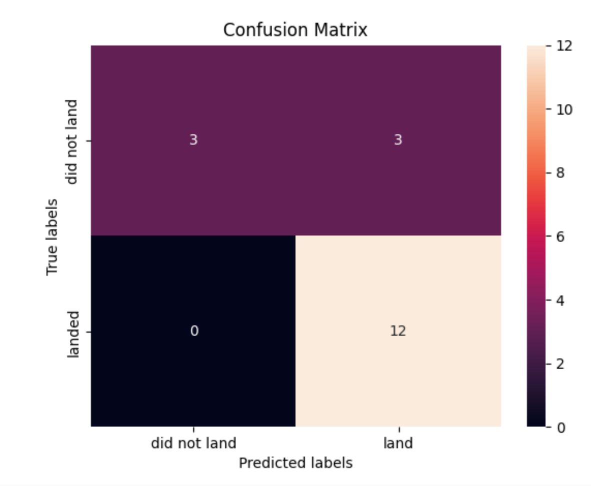
Confusion Matrix

- Our confusion Matrix show that all the classification models except for Decision Tree provide the same classification results.
- Therefore, any of the models except for a Decision Tree could be effective in predicting landing success.

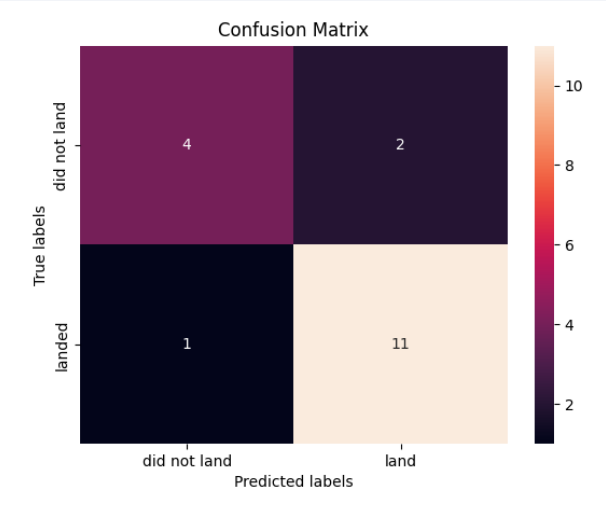
LogReg



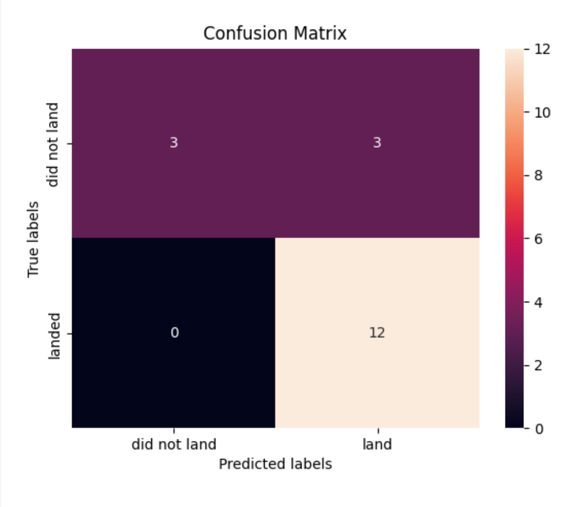
SVM



Decision Tree



KNN

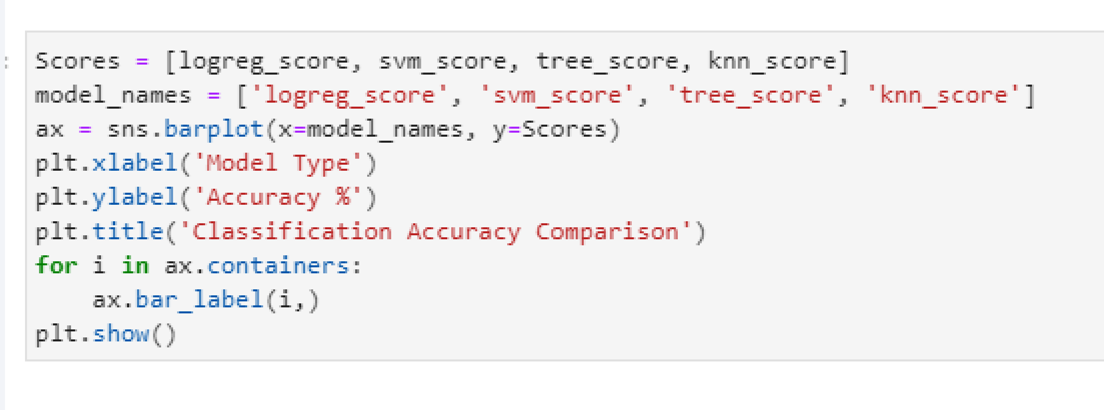


Conclusions

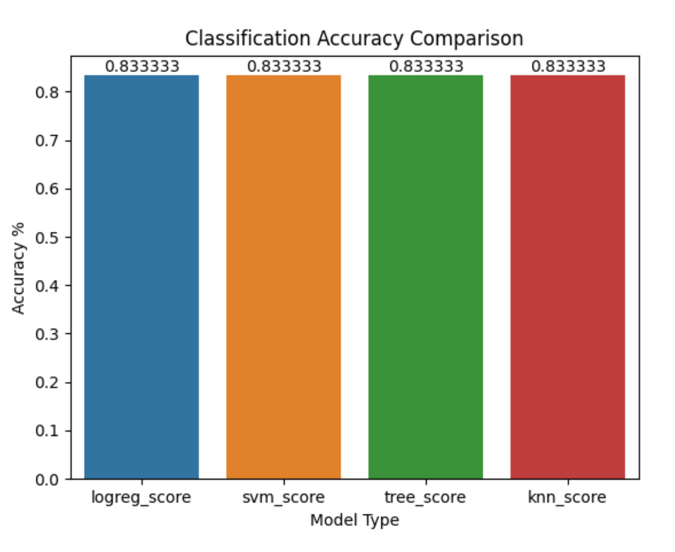
- KNN, SVM, and Log Regression were the best at predicting landing accuracy for this data.
- Per the exploratory data analysis, we can determine that payload mass is a large factor in determining the outcome of a landing.
- SSO, ES L1, HEO, and GEO orbits lead to the most successful landings.
- While rockets launched from KSC LC-39A are the most successful at landing, mostly due to the number of launches with low payloads.
- Overtime as rockets become more advanced the amount of successful landing should increase as a result.

Appendix

- I created this Bar chart to compare accuracy
- Created Data Sets:
    - https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/spacex_web_scraped.csv
    - https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/spacex_launch_dash.csv
    - https://github.com/jackmclay/Applied-Data-Science-Capstone/blob/main/dataset_part_1.csv



```
Scores = [logreg_score, svm_score, tree_score, knn_score]
model_names = ['logreg_score', 'svm_score', 'tree_score', 'knn_score']
ax = sns.barplot(x=model_names, y=Scores)
plt.xlabel('Model Type')
plt.ylabel('Accuracy %')
plt.title('Classification Accuracy Comparison')
for i in ax.containers:
    ax.bar_label(i,)
plt.show()
```



Thank you!