IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Teran Oung
August 16, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Webscraping
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization & SQL
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis
  - Interactive Analytics
  - Predictive Analytics

Introduction

- SpaceX is a leading company in space-industry rocket launches. They are able to innovate by keeping rocket launches relatively inexpensive – with Falcon 9 rocket launches costing only 62 million dollars – by reusing the first stage between launches.

- The goal of this project is to:
  - Identify factors that affect the landing outcome
  - Determine the relationship between different rocket variables in landings
  - Find ideal conditions to achieve a successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API and Web Scrapping
- Perform data wrangling
  - One-Hot Encoding for classification
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- With this study, we worked with SpaceX launch data that is gathered from an API (SpaceX REST API), and web scrapping from Wikipedia
- For REST API, we start by using the ‘get’ request and decode the response as ‘json’
- Once the data is transformed, we can normalize the data by cleaning the data and exporting

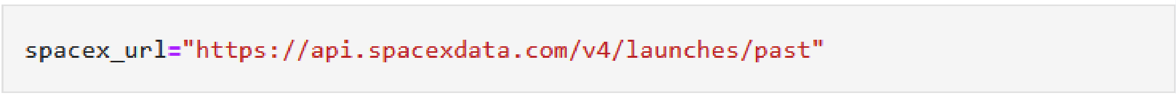
- For web scrapping, we can extract the data as an HTML table, parse the table, and convert it to a pandas dataframe for more analysis

Data Collection – SpaceX API

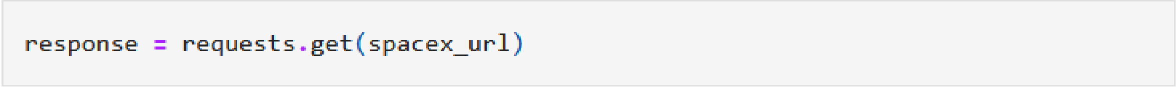
Get request for launch data using API

Use json_normalize to convert json to dataframe

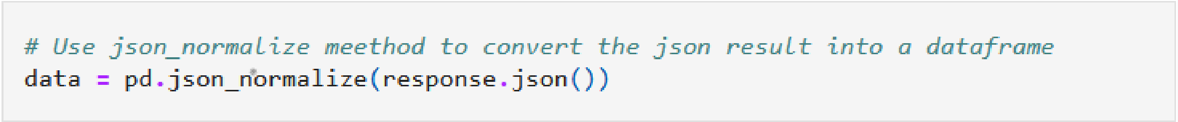
Clean data and fill missing values



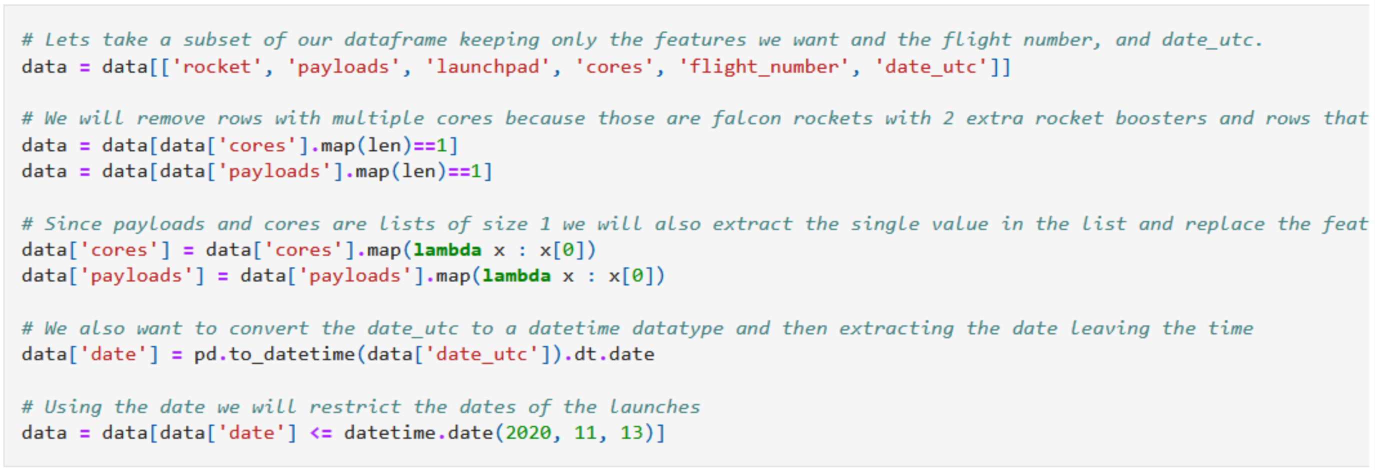
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```



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



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



```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feat
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

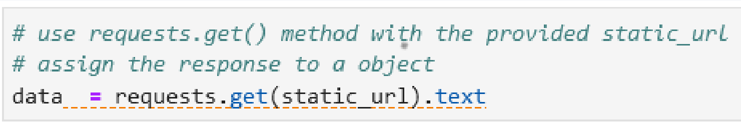
Source: https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/1-Data%20Collection%20API.ipynb

Data Collection - Scraping

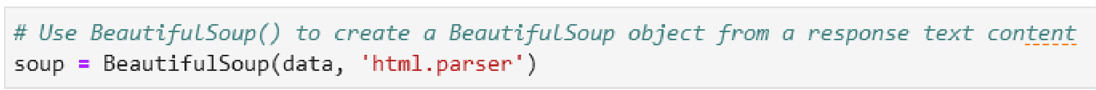
Get request for data from URL

Creates a BeautifulSoup object from the response

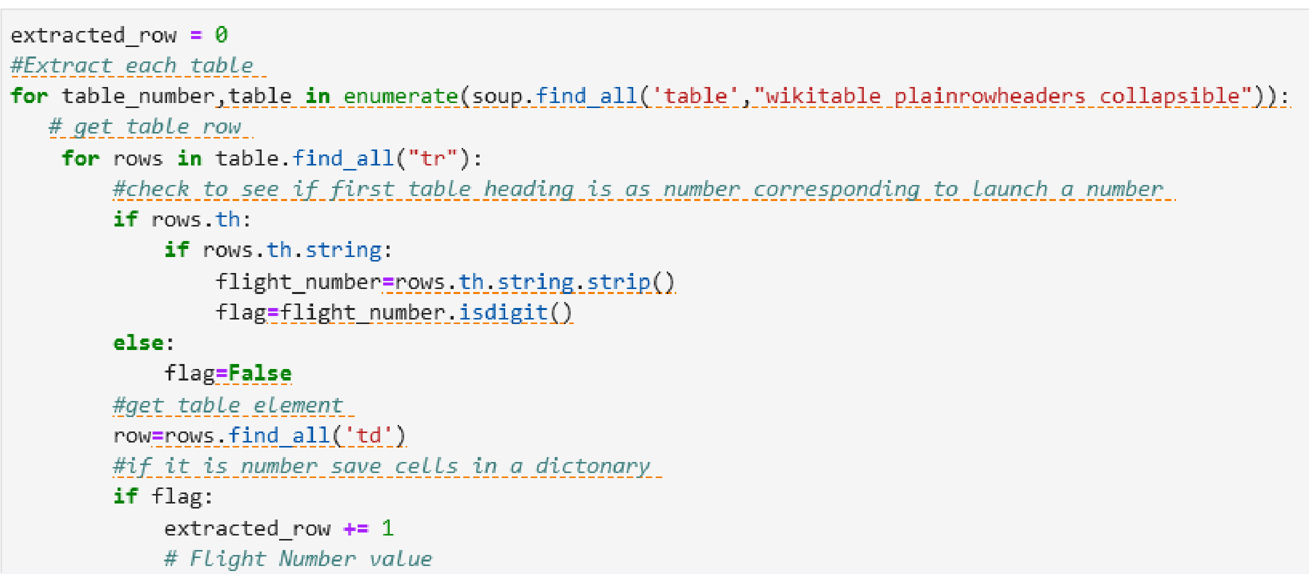
Extracts and creates table from the response data



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



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



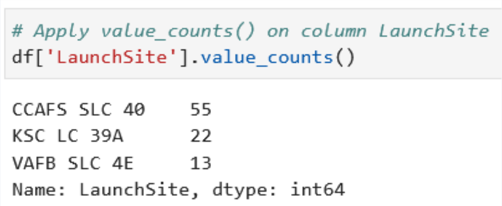
```
extracted_row = 0
#Extract each table 
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row 
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number 
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element 
        row=rows.find_all('td')
        #if it is number save cells in a dictonary 
        if flag:
            extracted_row += 1
            # Flight Number value
```

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/2-Data%20Collection%20with%20Web%20Scrapping.ipynb

Data Wrangling

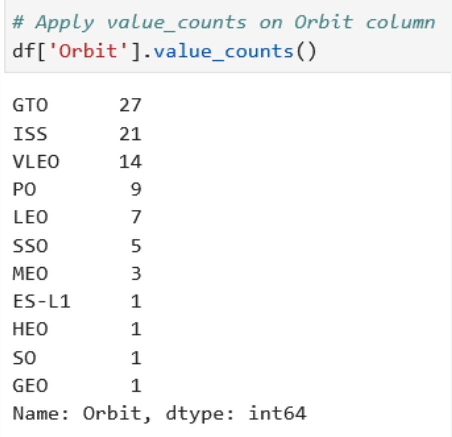
- Data Wrangling is the process of transforming and structuring data into a desired format.

Identify and define each launch



```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

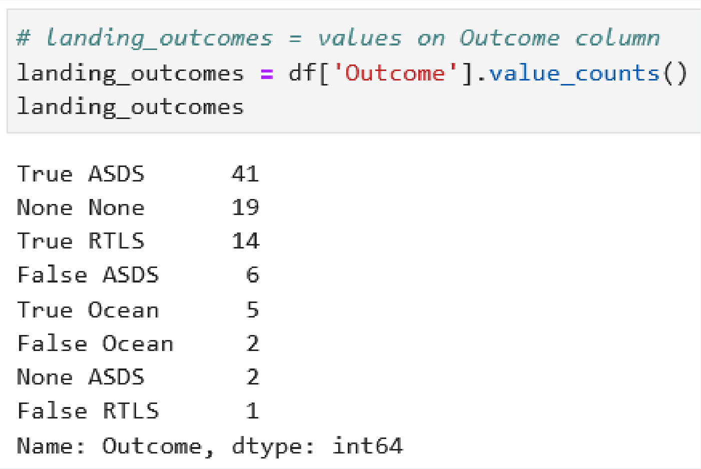
```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```



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

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

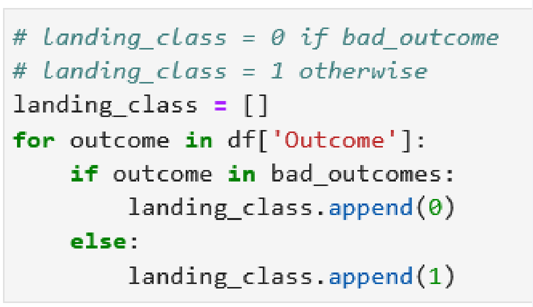
Identify each mission outcome



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

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

Create a landing outcome label



```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/3-Data%20Wrangling.ipynb

EDA with Data Visualization

- Scatterplots: shows relationship/correlation between two variables
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Flight Number vs Orbit Type
  - Payload vs Orbit Type
- Bar Graphs: shows relationship between numeric and categorical variables
  - Orbit Type vs Success Rate
- Line Graphs: shows variables and their trends
  - Year vs Success Rate

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/5-EDA%20with%20Visualization.ipynb

EDA with SQL

- With SQL, we can get a better understanding of the data:
  - 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. 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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/4-EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We can build an interactive map with Folium to coordinate each launch site with markers:
  - Red circles at each launch site, with labels
  - Markers for each landing – green for successful landings and red for failed landings
  - Lines to show the distance between launch sites and key locations near them
- These objects give us a better visual understanding of some geographical patterns about the launch sites

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/6-Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We can build an interactive dashboard to allow users to customize plots and graphs based on what sites and payloads they want to evaluate.
- Customizations include:
  - Dropdowns to select specific launch sites
  - Pie chart showing total successes and success rates by site
  - Range slider to specify payload mass ranges
  - Scatterplot showing correlation between success rates and payload mass

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/7-Interactive%20Visual%20Analytics%20with%20Ploty.py

Predictive Analysis (Classification)

- Building a model
  - Load the datasets
  - Split the dataset into training and test data
  - Set the parameters and fit it to the dataset
- Evaluating the model
  - Get hyperparameters for each
  - Check the accuracy
  - Plot a confusion matrix
- Finding the best model
  - Compare accuracy scores for all models

Source:
https://github.com/Grighund/AppliedDataScienceCapstone/blob/main/8-Predictive%20Analysis.ipynb

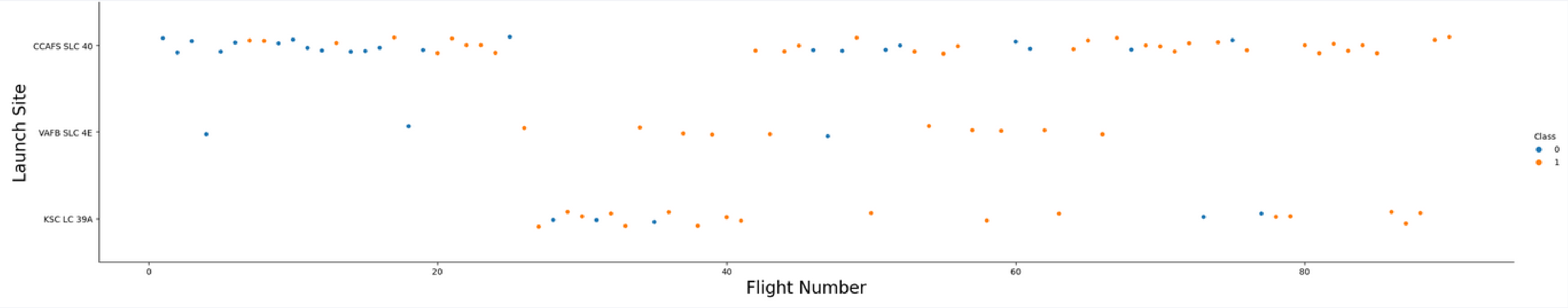
Results

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

Section 2

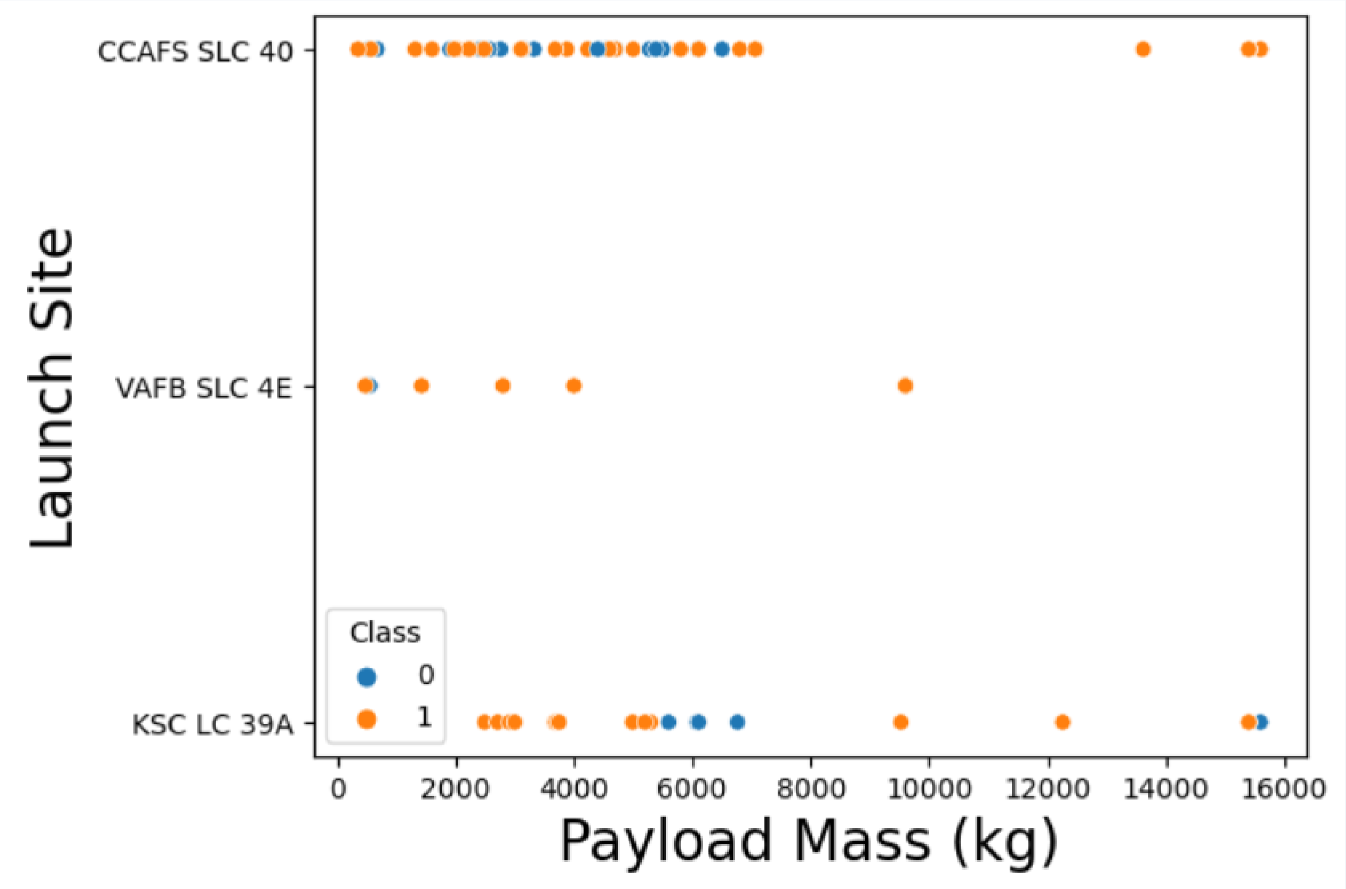
Insights drawn from EDA

Flight Number vs. Launch Site



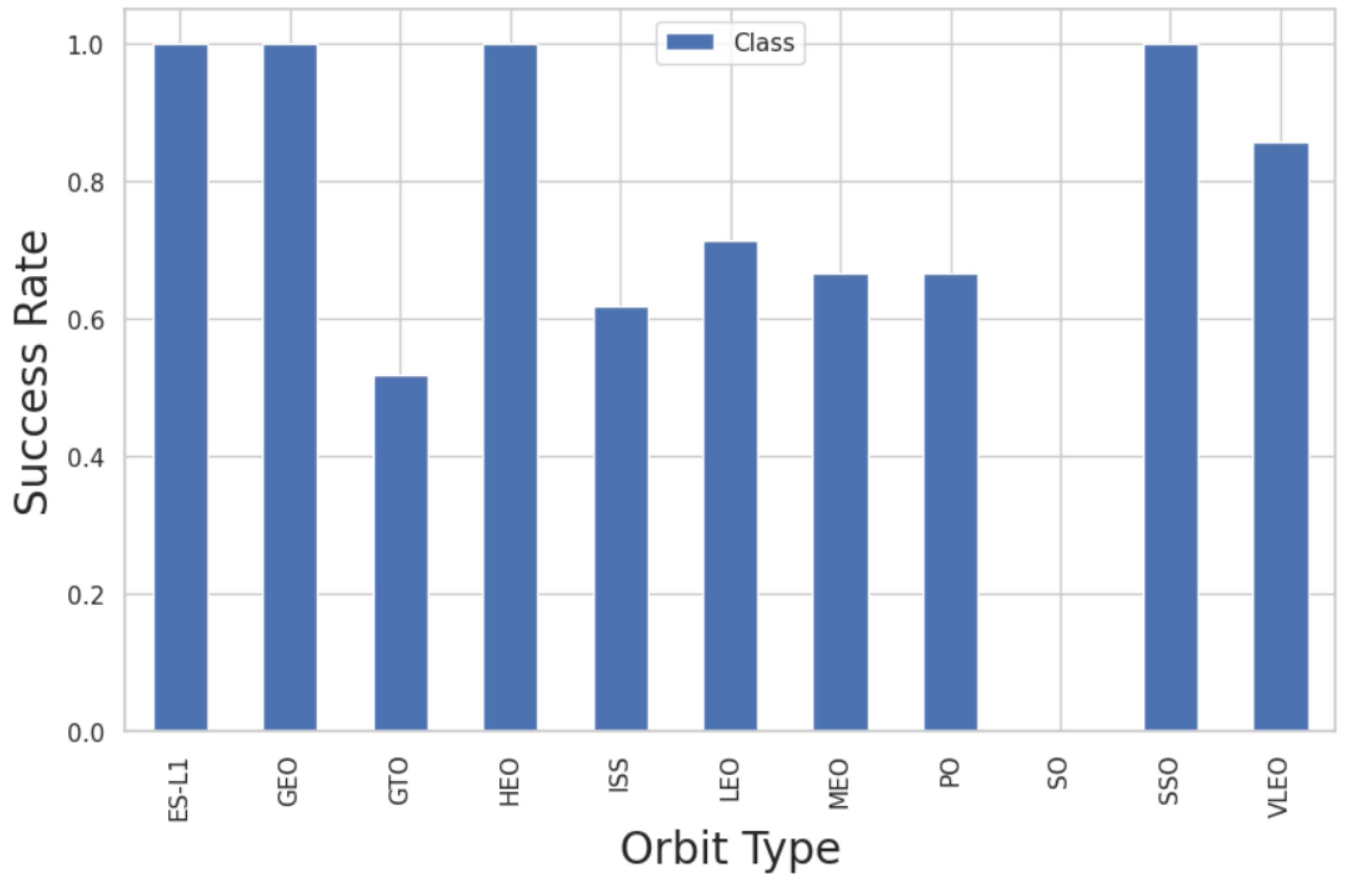
- The Flight Number vs Launch Site scatterplot shows that as we develop more flights for each Launch Site, we start to see more successes (Class 1) as opposed to failures (Class 0)

Payload vs. Launch Site



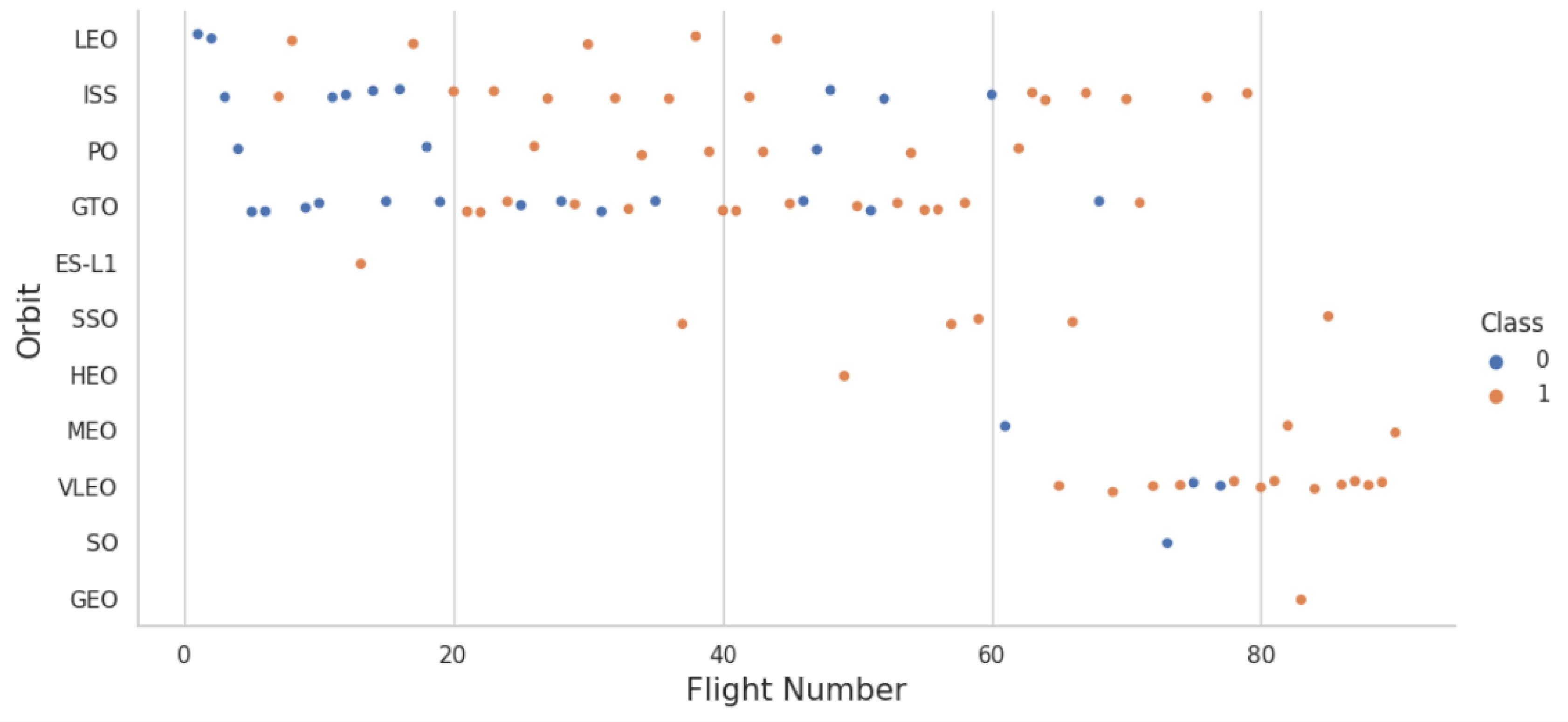
- The Payload vs Launch Site plot shows that heavier payloads may have a higher tendency to succeed
- We can also observe that the VAFB SLC did not launch many heavier rockets

Success Rate vs. Orbit Type



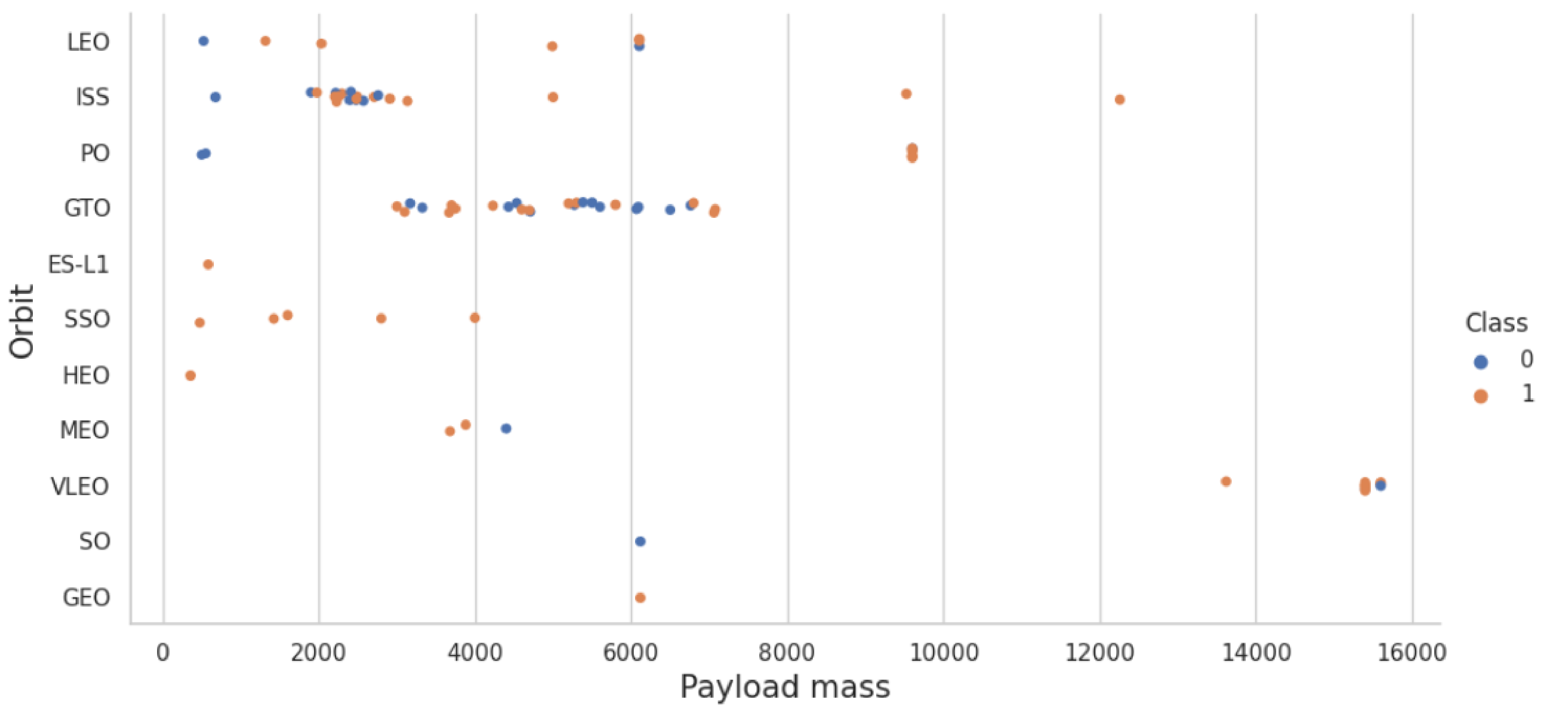
- The bar chart shows which Orbit types have the highest success rates.
- However, the chart does not show the number of occurrences

Flight Number vs. Orbit Type



- We can observe from this that for some orbit types, success rate is correlated with the number of flights, as seen in LEO orbits. However for some, such as GTO, it may not indicate much relationship.

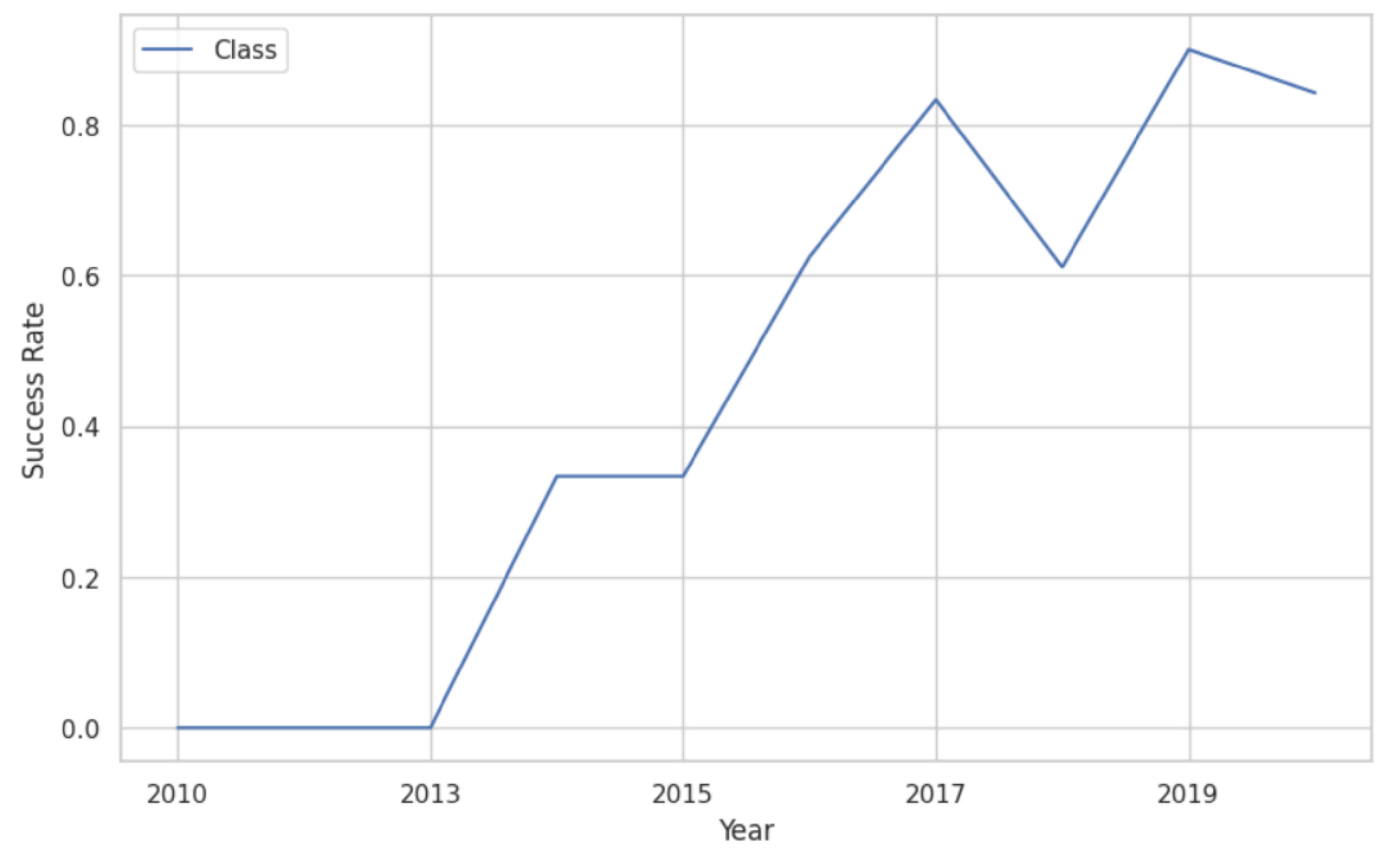
Payload vs. Orbit Type



- For some orbit types, heavier payload may increase the success rates – as seen with LEO and ISS.

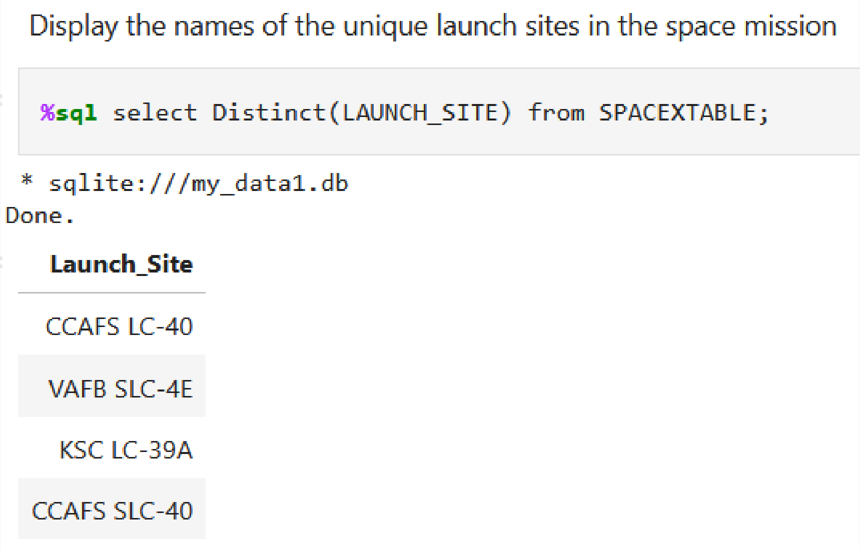
- However for orbits like GTO, we see both successes and failures

Launch Success Yearly Trend



- We can see that since 2013, we have seen a general positive trend toward more successful launches

All Launch Site Names



Display the names of the unique launch sites in the space mission

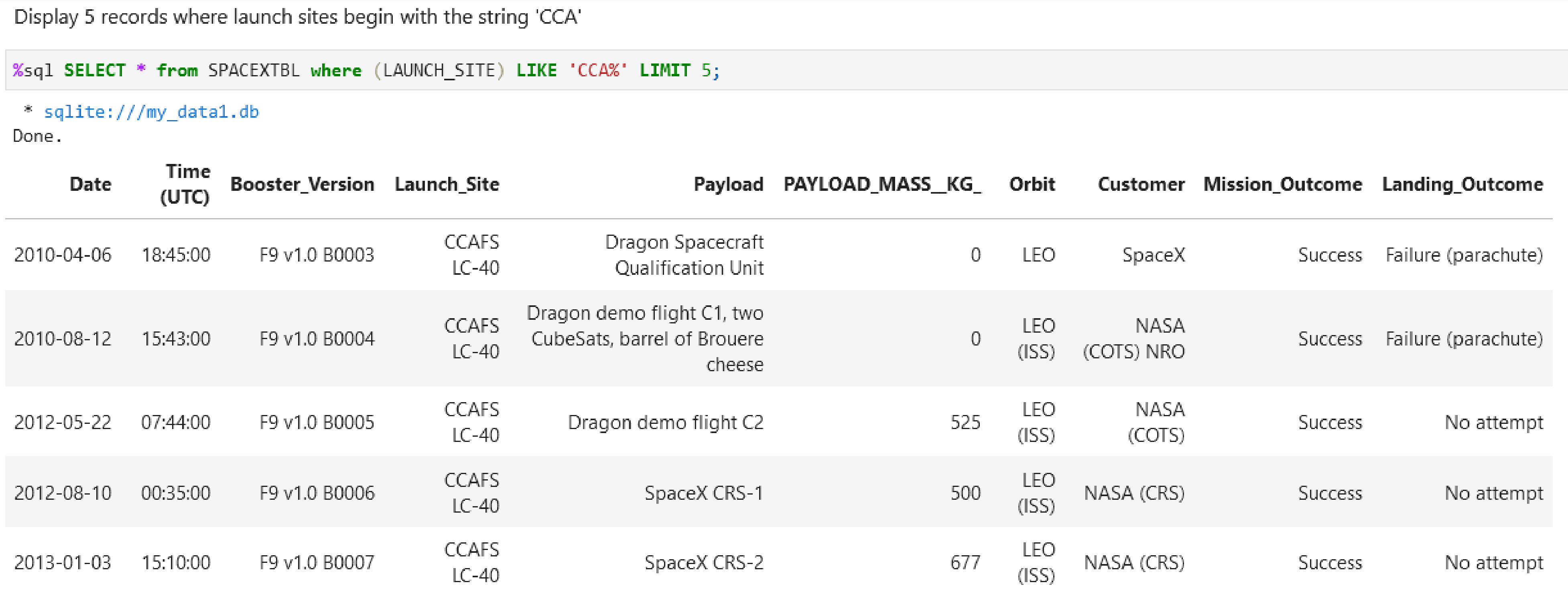
```
%sql select Distinct(LAUNCH_SITE) from SPACEXTABLE;
```

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

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

- We can use the Distinct query to display unique launch site names, without repeats

Launch Site Names Begin with 'CCA'



Display 5 records where launch sites begin with the string 'CCA'

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

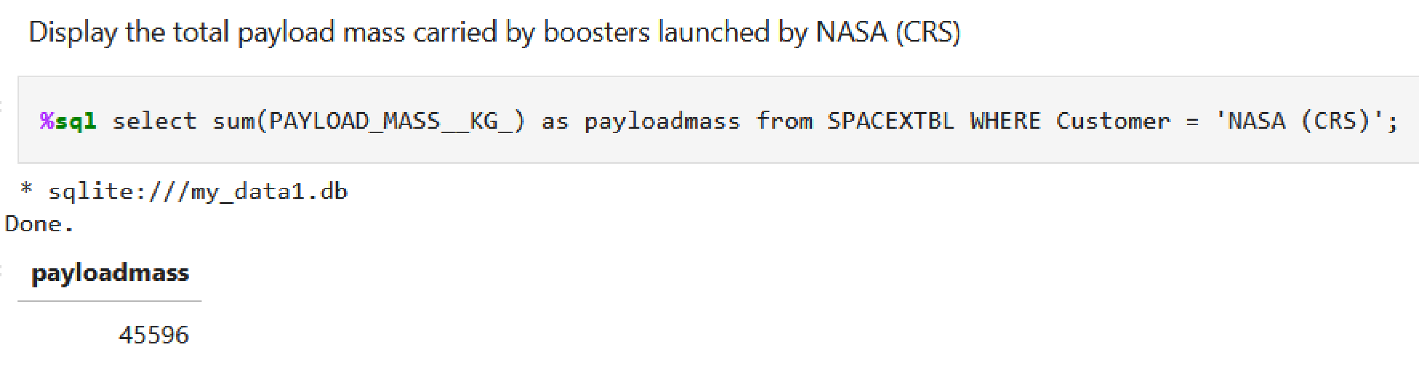
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The query shows to first 5 record details for any launch site beginning with “CCA” in the Launch Site title

Total Payload Mass

- The below query finds the total payload mass for all boosters with the customer name “NASA (CRS)”



Display the total payload mass carried by boosters launched by NASA (CRS)

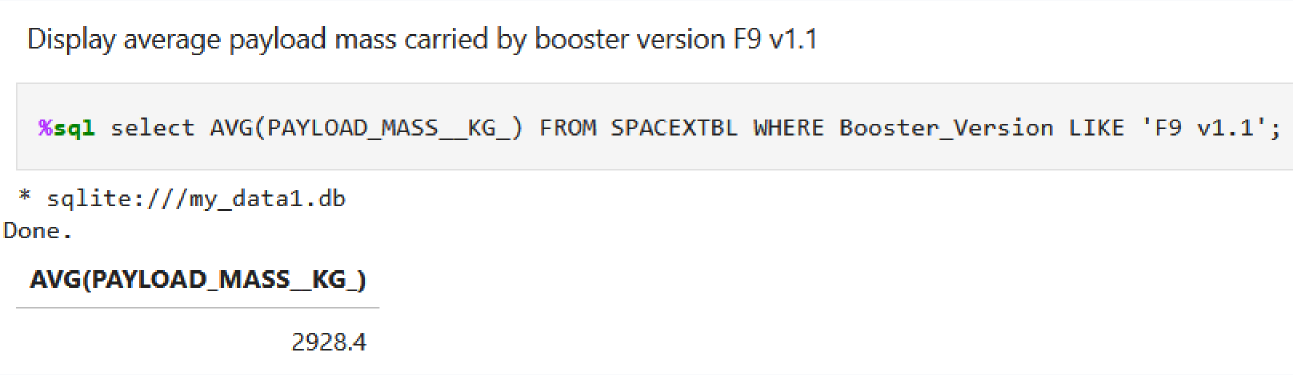
```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

payloadmass
45596

Average Payload Mass by F9 v1.1

- Calculates the average payload mass for all boosters version with the string “F9 v1.1”



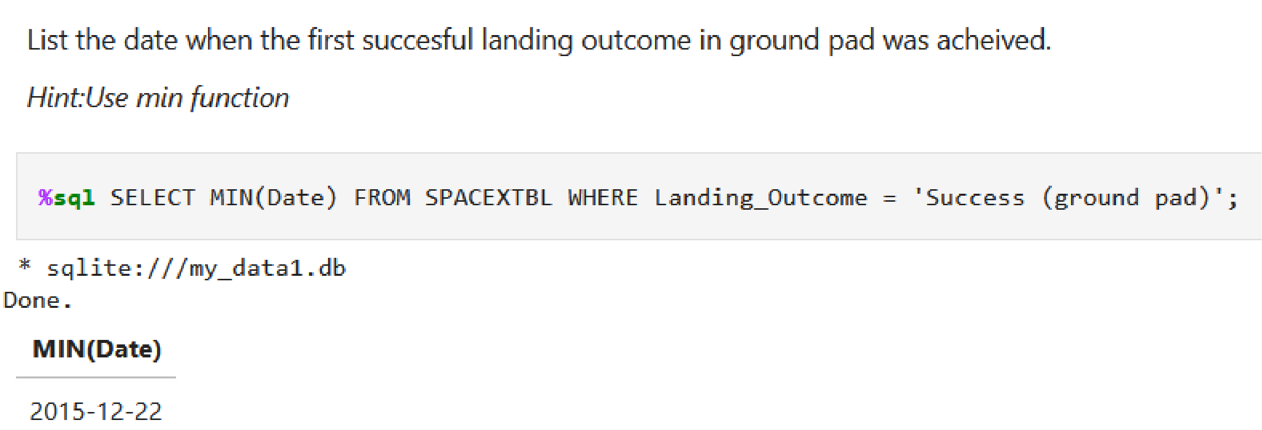
Display average payload mass carried by booster version F9 v1.1

```
%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



List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

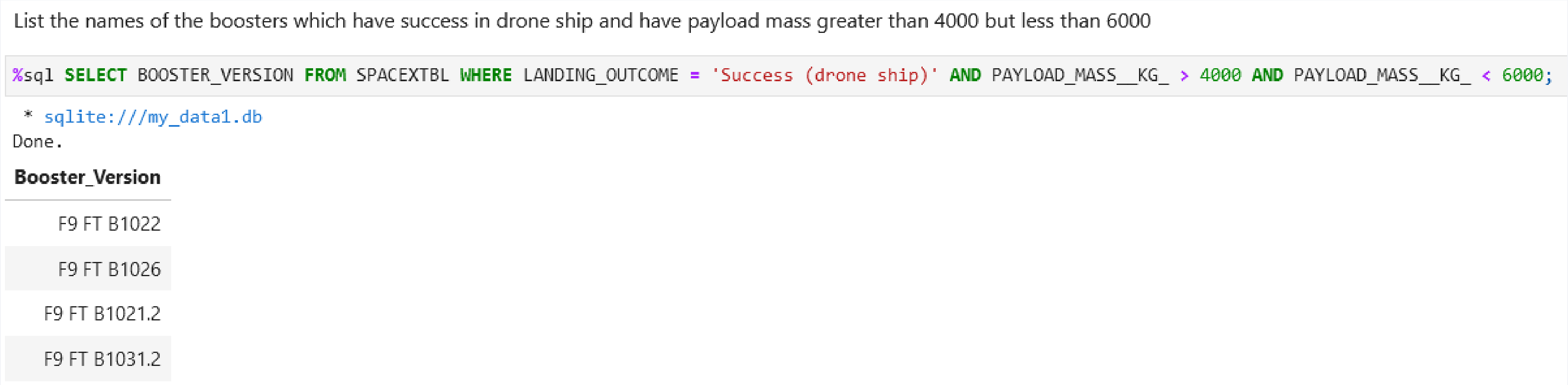
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

MIN(Date)
2015-12-22

- The query finds a successful landing outcome on a ground pad, and displays the earliest date using the MIN function

Successful Drone Ship Landing with Payload between 4000 and 6000



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

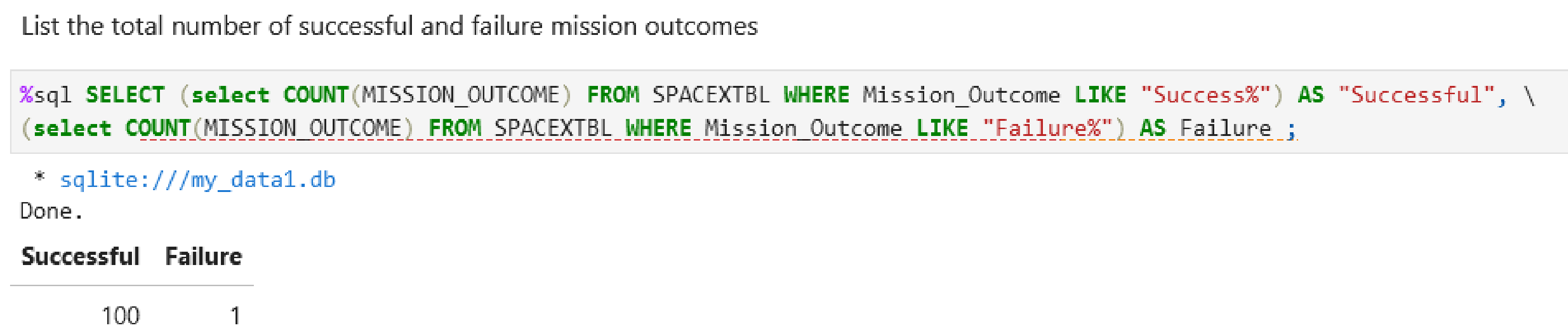
```
 * sqlite:///my_data1.db
Done.
```

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

- This finds the booster versions that match multiple criteria:
  - Successful landing with drone ship
  - Payload mass greater than 4000 kg
  - Payload mass also less than 6000 kg

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes, grouping any mission outcomes beginning with the string “Success” vs “Failure”



List the total number of successful and failure mission outcomes

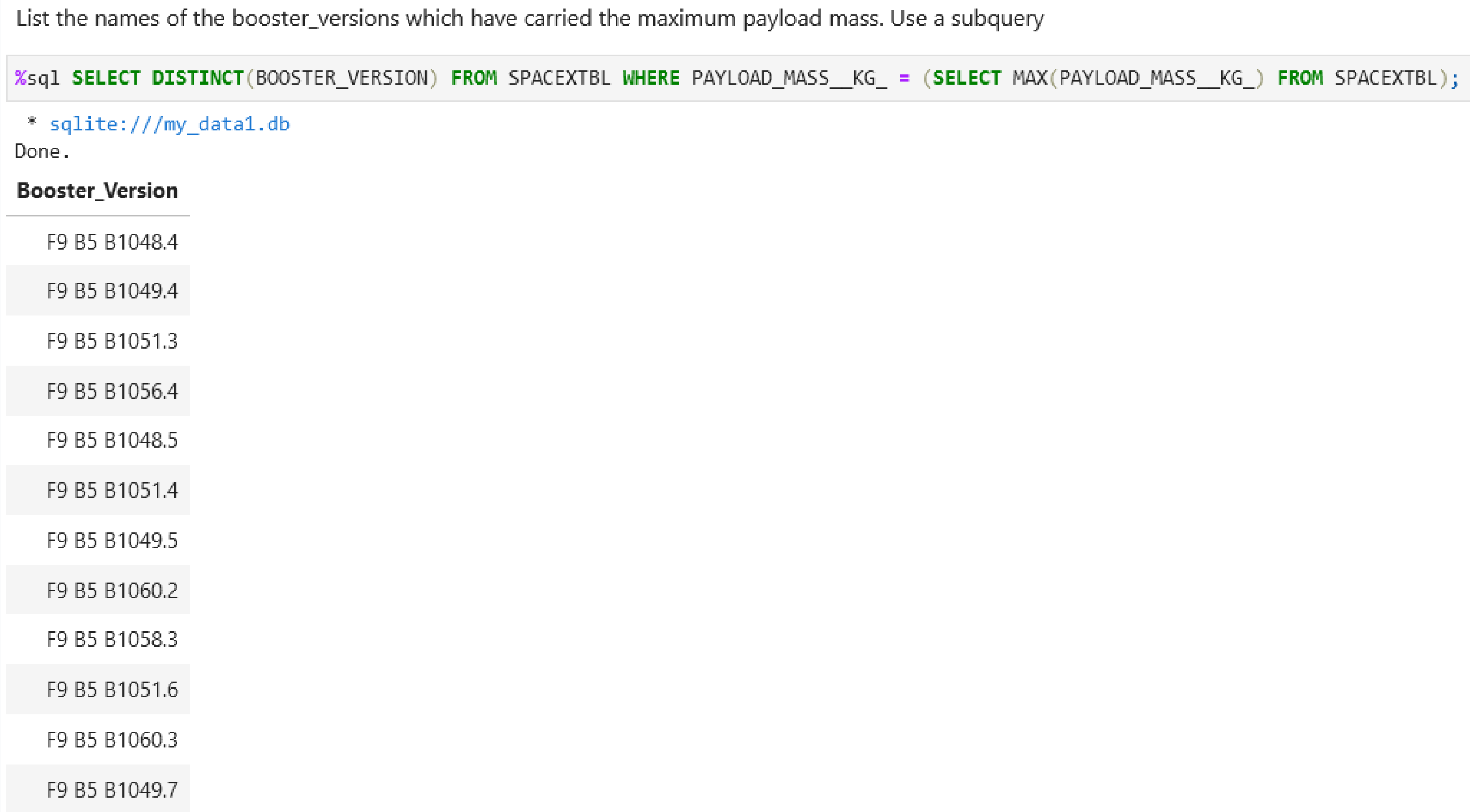
```
%sql SELECT (select COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE Mission_Outcome LIKE "Success%") AS "Successful", \
(select COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE Mission_Outcome LIKE "Failure%") AS Failure ;
```

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

Successful	Failure
100	1

Boosters Carried Maximum Payload

- We can find the booster versions with the maximum payload mass using the WHERE and MAX function



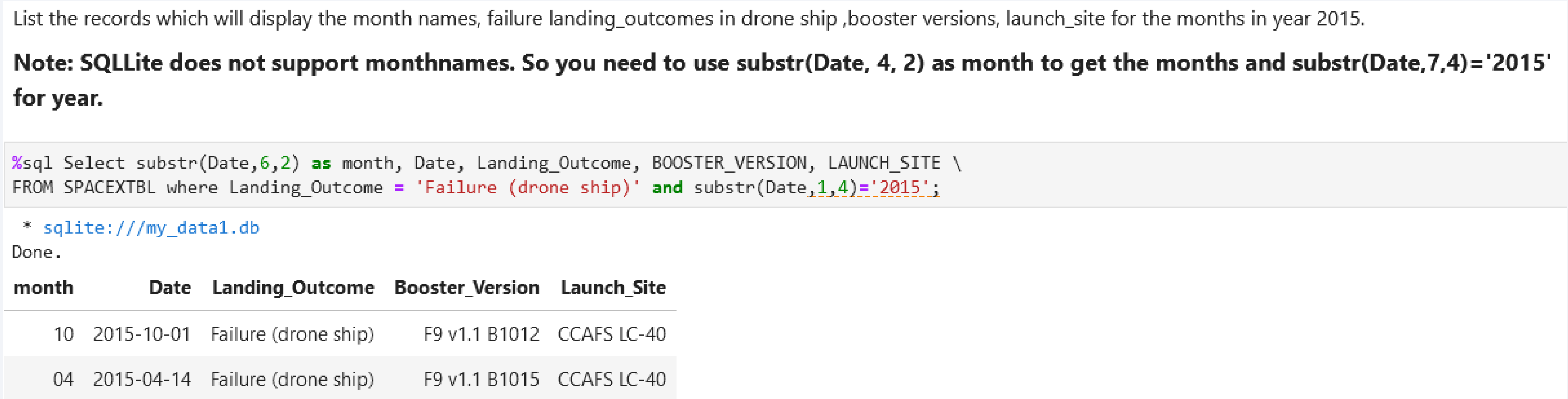
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%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 records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

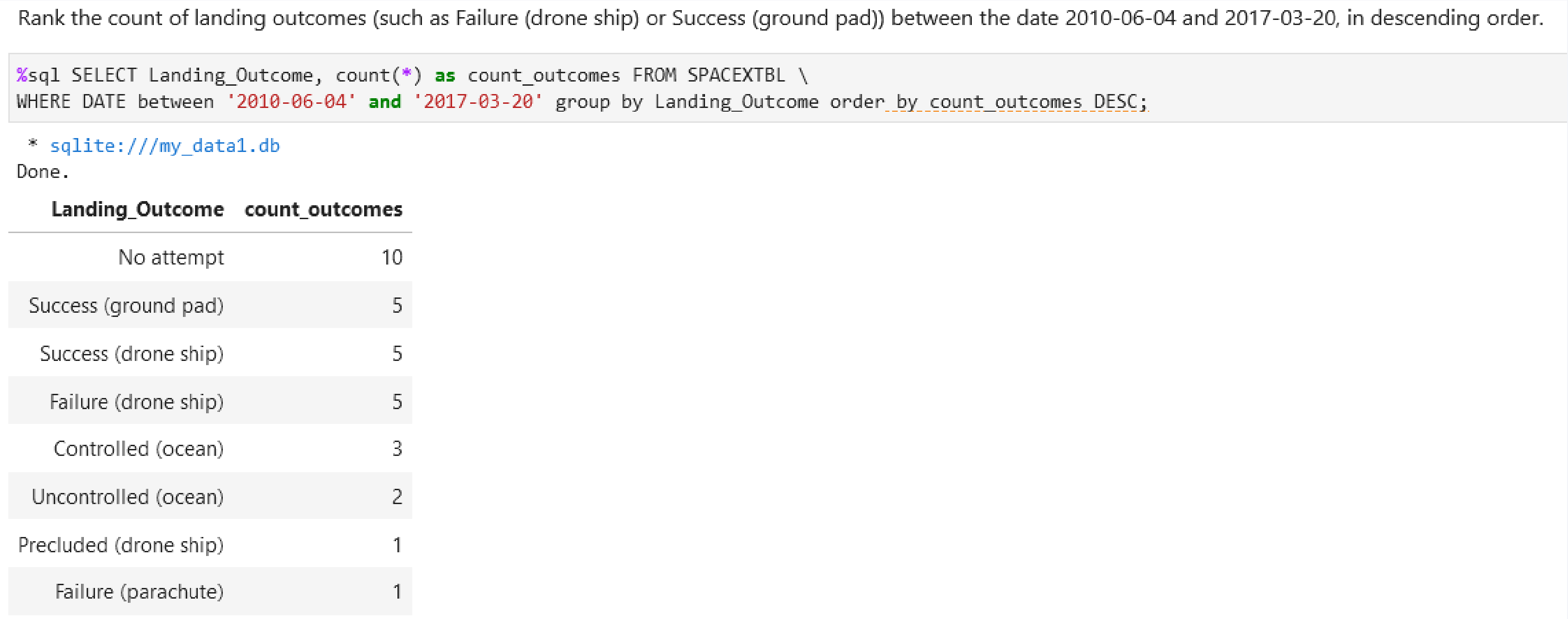
```
%sql Select substr(Date,6,2) as month, Date, Landing_Outcome, BOOSTER_VERSION, LAUNCH_SITE \
FROM SPACEXTBL where Landing_Outcome = 'Failure (drone ship)' and substr(Date,1,4)='2015';
```

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

month	Date	Landing_Outcome	Booster_Version	Launch_Site
10	2015-10-01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- The query finds records with Failures in drone ships and displays:
  - Month number (and date for reference)
  - Landing outcome
  - Booster version
  - Launch site

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.

```
%sql SELECT Landing_Outcome, count(*) as count_outcomes FROM SPACEXTBL \
WHERE DATE between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count_outcomes DESC;
```

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

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

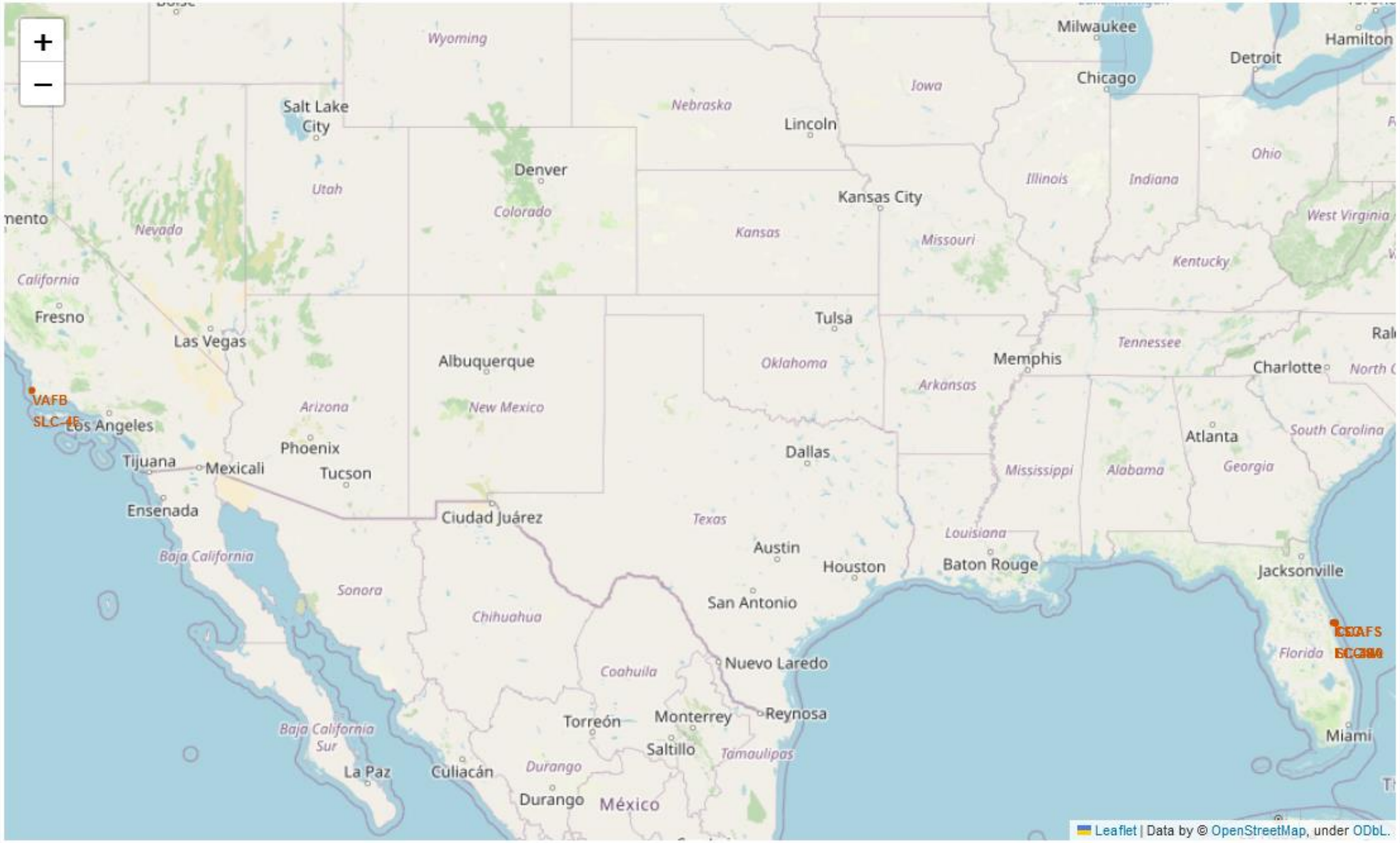
- We can count the number of occurrences for each landing outcome between the dates 6/4/2010 and 3/20/2017
- We also ranked or ordered by the number of occurrences in descending order

Section 3

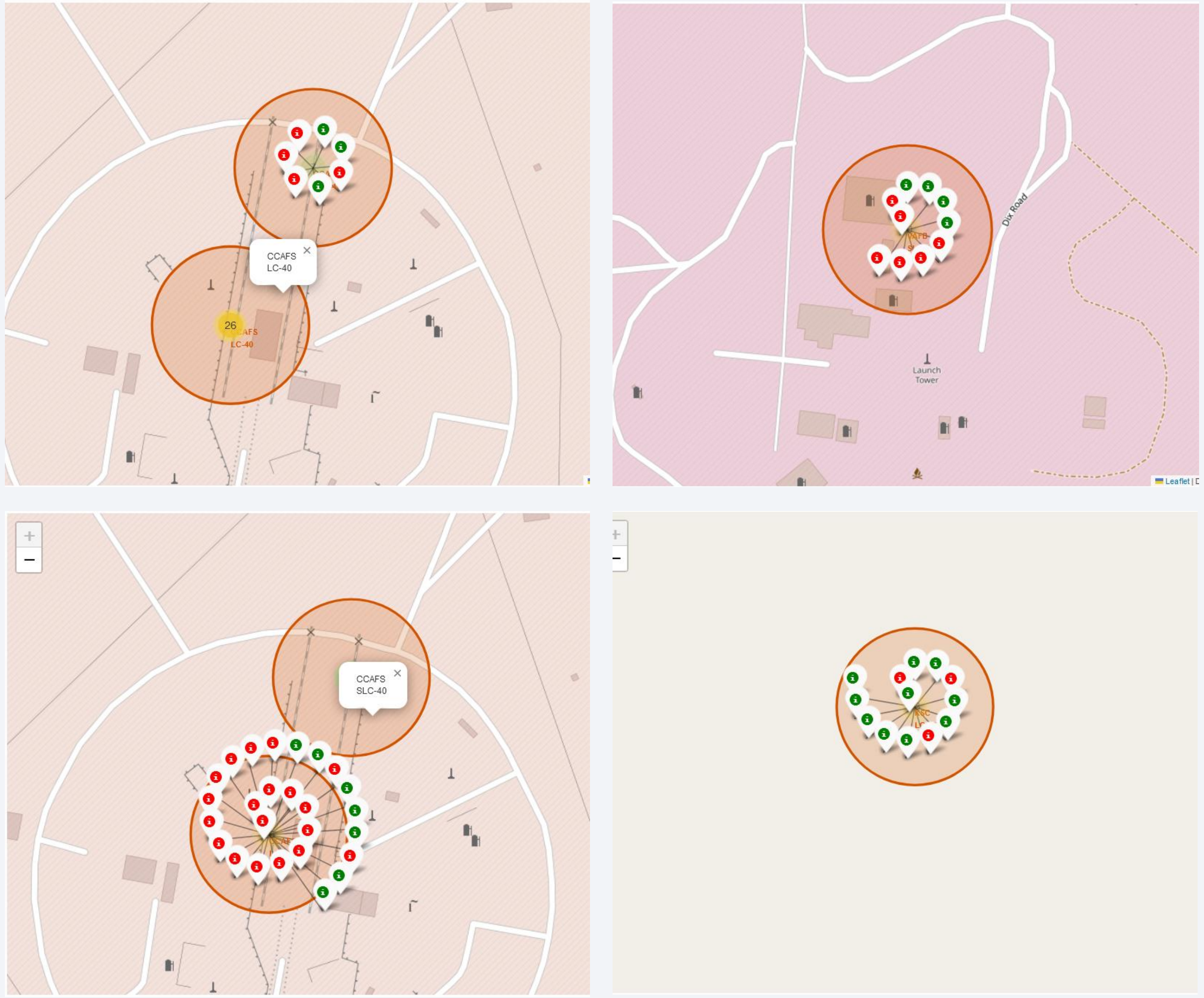
Launch Sites Proximities Analysis

Folium Map – Launch Site Locations

- We can create a map with marked locations for all the SpaceX launch sites

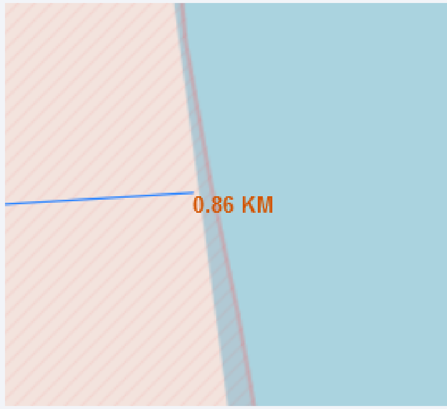


Folium Map – Success/Failure Markers



- We can add markers for each launch attempt at each location
- Green markers indicate successful launches
- Red markers indicate failures

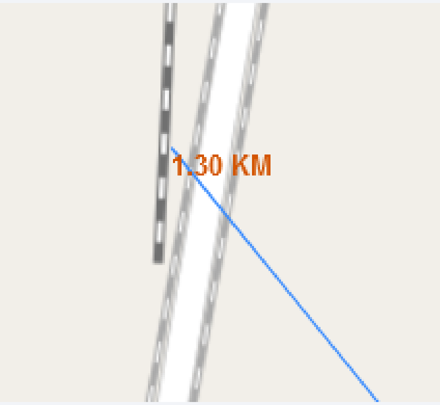
Folium Map – Distances from CCAFS SLC-40



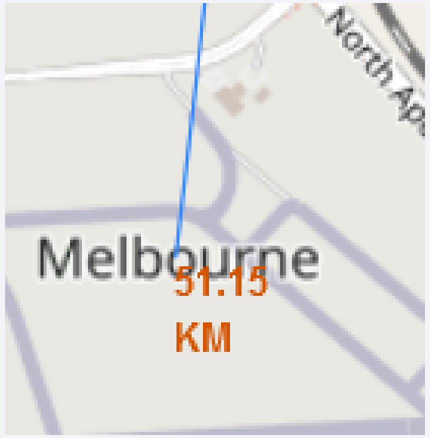
Distance to coastline

Distance to highway

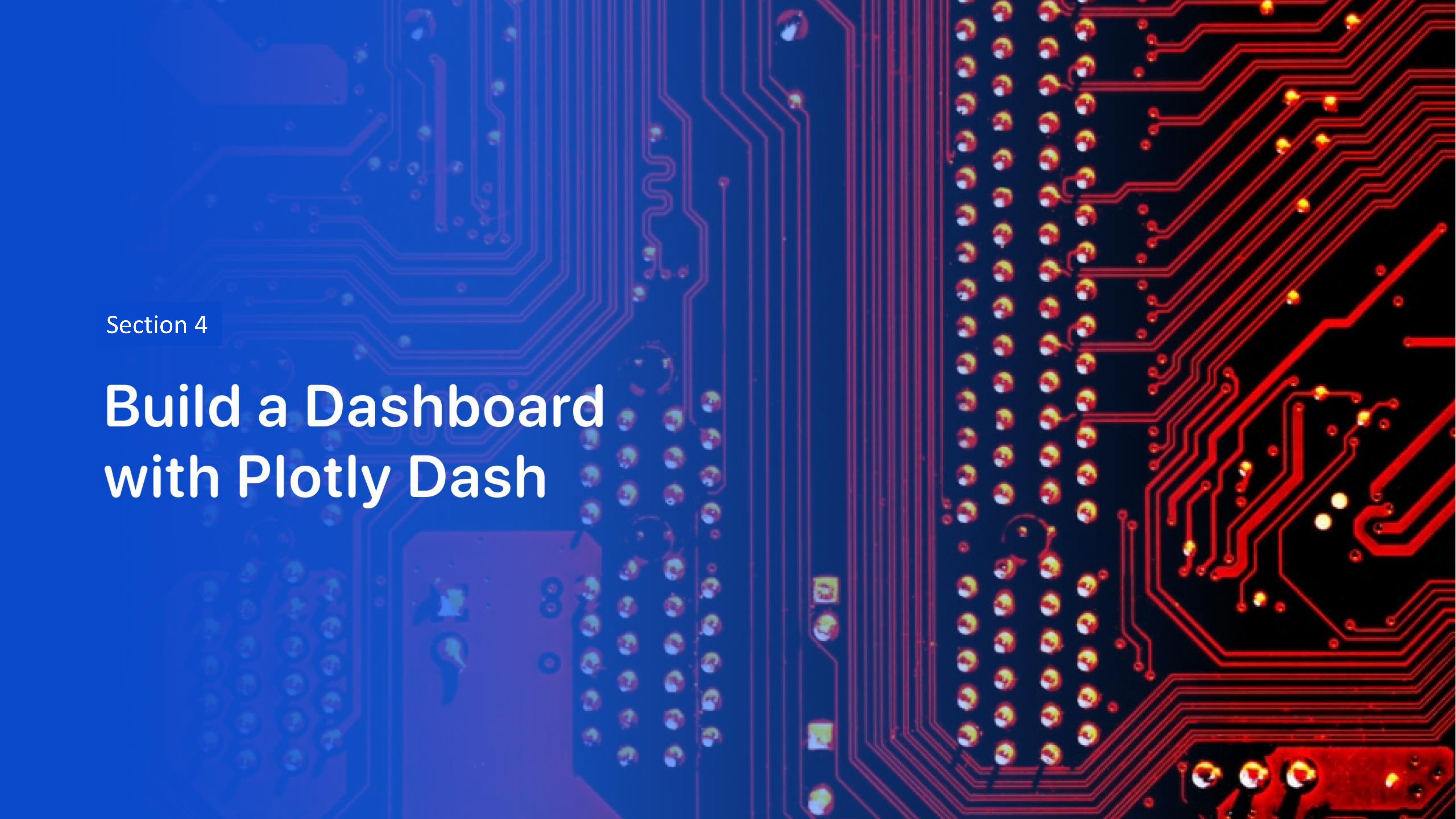
- Are launch sites in close proximity to railways?
  - Yes, approximately 1.3 KM away
- Are launch sites in close proximity to highways?
  - Yes, approximately 0.59 KM away
- Are launch sites in close proximity to coastline?
  - Yes, approximately 0.86 KM away
- Do launch sites keep certain distance away from cities?
  - Yes, the nearest major city (Melbourne) is approximately 51.15 KM away



Distance to railway



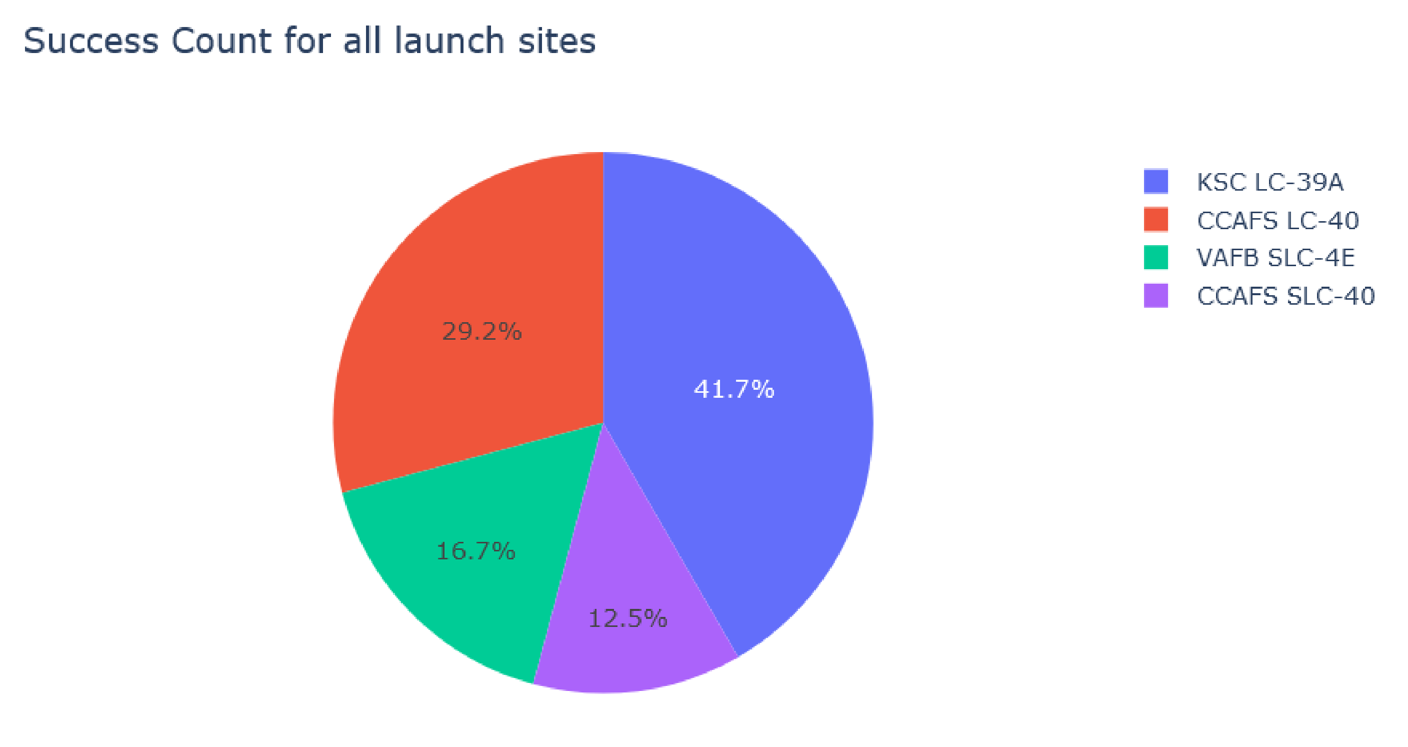
Distance to city



Section 4

Build a Dashboard with Plotly Dash

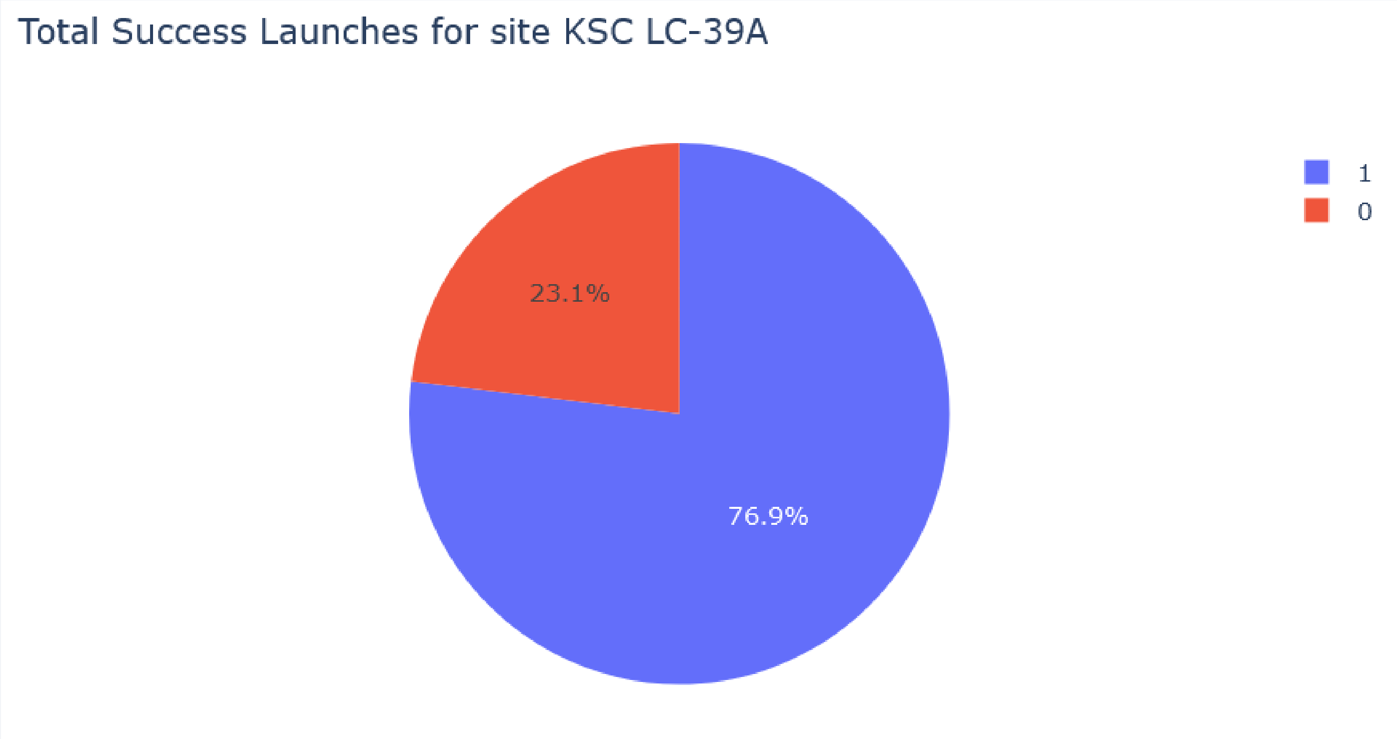
Launch Successes for All Sites



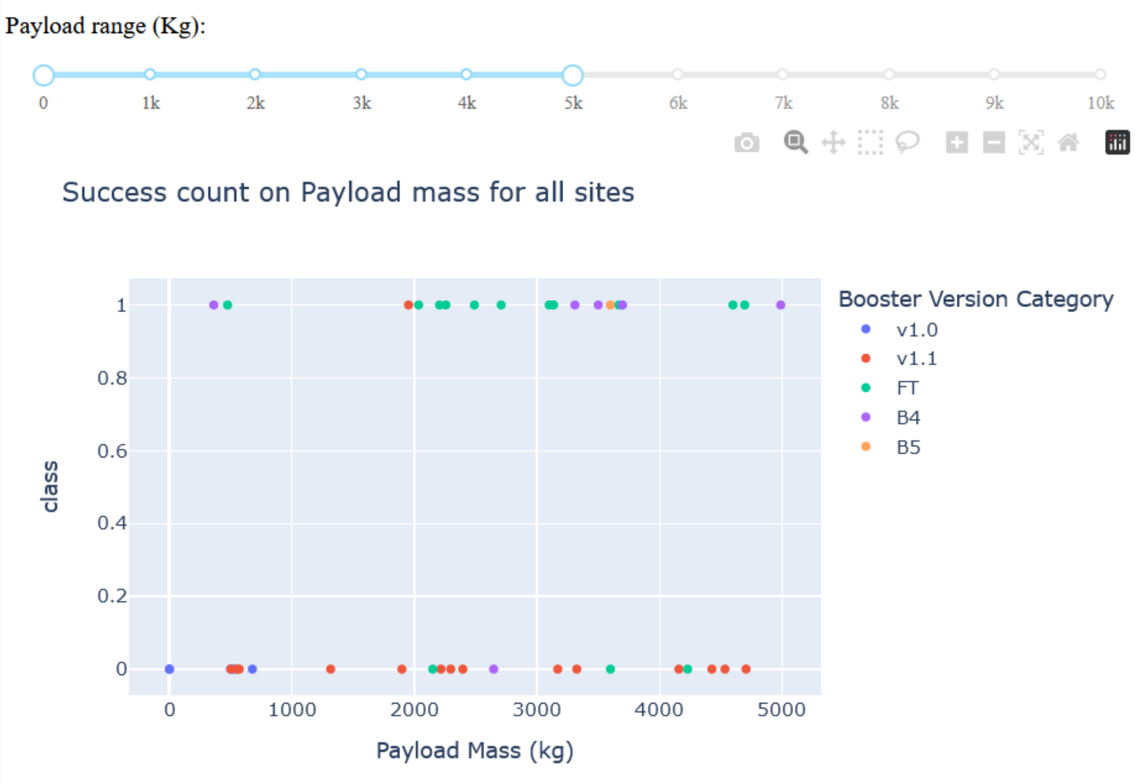
- This chart shows the most successful launches for all sites
- The KSC LC-39A site has the most successes out of all the different sites

KSC LC-39A Success Rates

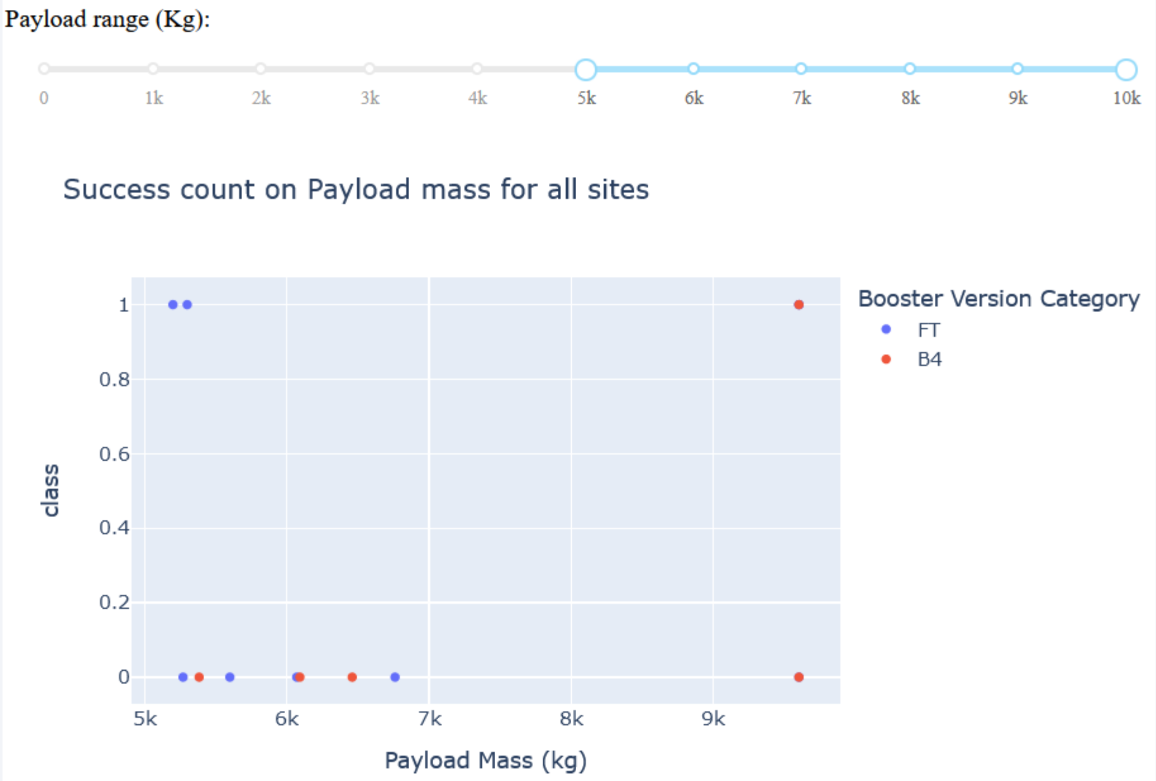
- Looking at the launch site KSC LC-39A, we can dive deeper to look at the success rates
  - 76.9% Success
  - 23.1% Failure



Success Counts by Payload



- Payloads between 0k and 5k have a relatively high success rate, particularly with the FT booster



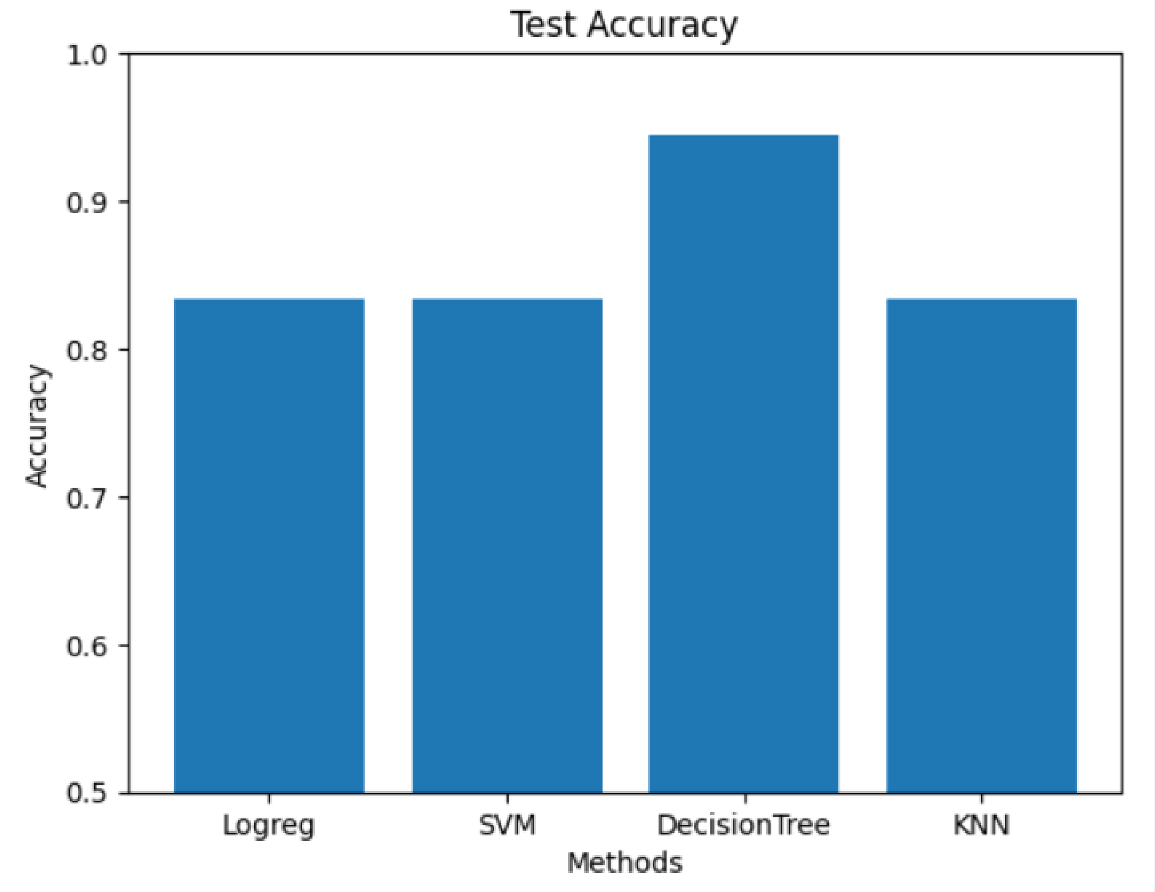
- Payloads between 5k and 10k have a low success rate

Section 5

Predictive Analysis (Classification)

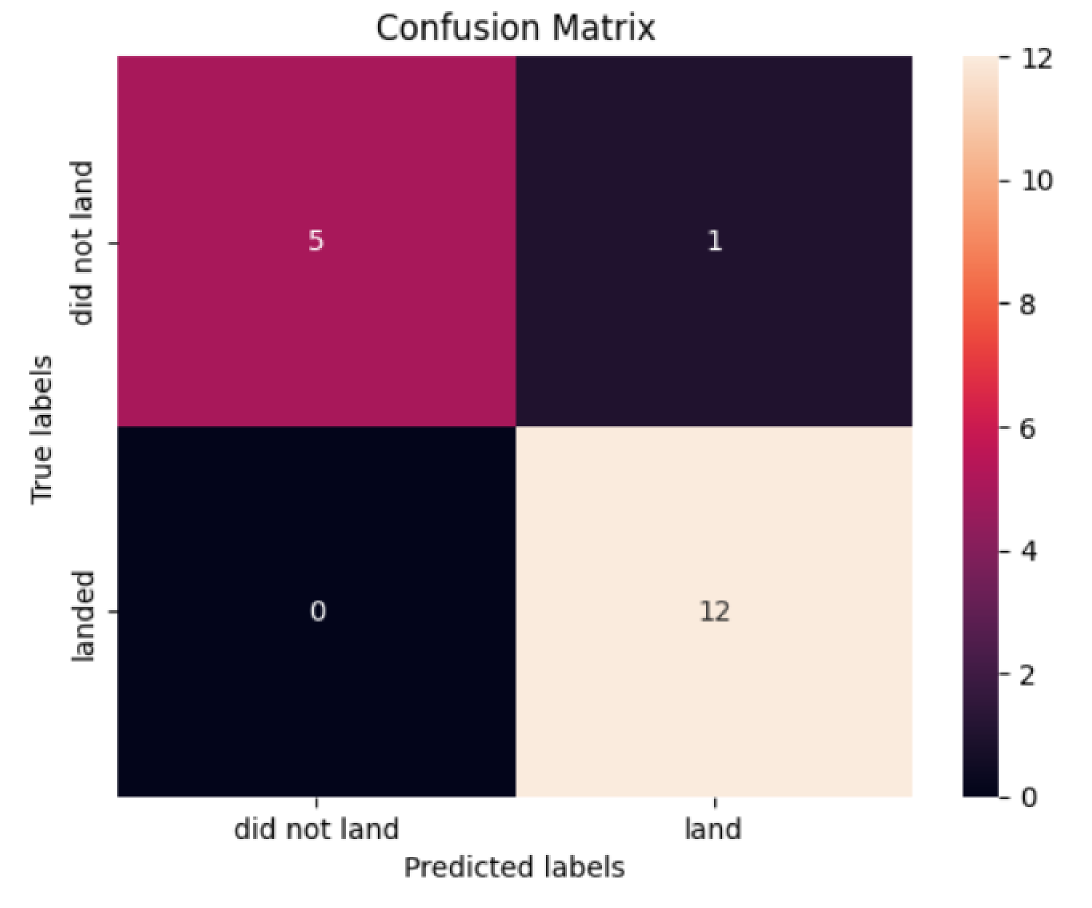
Classification Accuracy

- The accuracy of the difference methods were quite similar, but the Decision Tree method showed slightly better accuracy results



Confusion Matrix

- The confusion matrix for the Decision Tree method showed that it can predict the performance.
- However, there is a still a chance of False-Positives (unsuccessful landings being marked as successful)



Conclusions

- The success of a mission can depend on several factors, including launch site, oribit, payload mass, and number of launches.
- Some of the most successful orbits by success rate include ES-L1, GEO, HEO, and SSO
- Payload mass can impact the success rate, with lower payload generally having more successes
- The most successful launch site is KSC LC-39A
- The Decision Tree model was the most accurate for machine learning classifications

Appendix

- For code, queries and output details, please see the attachments at:

https://github.com/Grighund/AppliedDataScienceCapstone

Thank you!