IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pelumi Ayeni

September 18, 2022

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Executive Summary

- **Summary of methodologies**
  - Data Collection through API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL and Data Visualisation
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Plotly Dash
  - Machine Learning Predictive Analysis

- **Summary of all results**
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

Project Background and Context

SpaceX 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 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Based on public information and machine learning models, we are going to predict if SpaceX will reuse the first stage

Problems We Need Answers For

- What influences the failure or success of a rocket landing?
- What are the effects of each relationship of the rocket variables on the success rate of a successful landing?
- What conditions will allow SpaceX achieve the best rocket landing success rate?

Section 1

Methodology

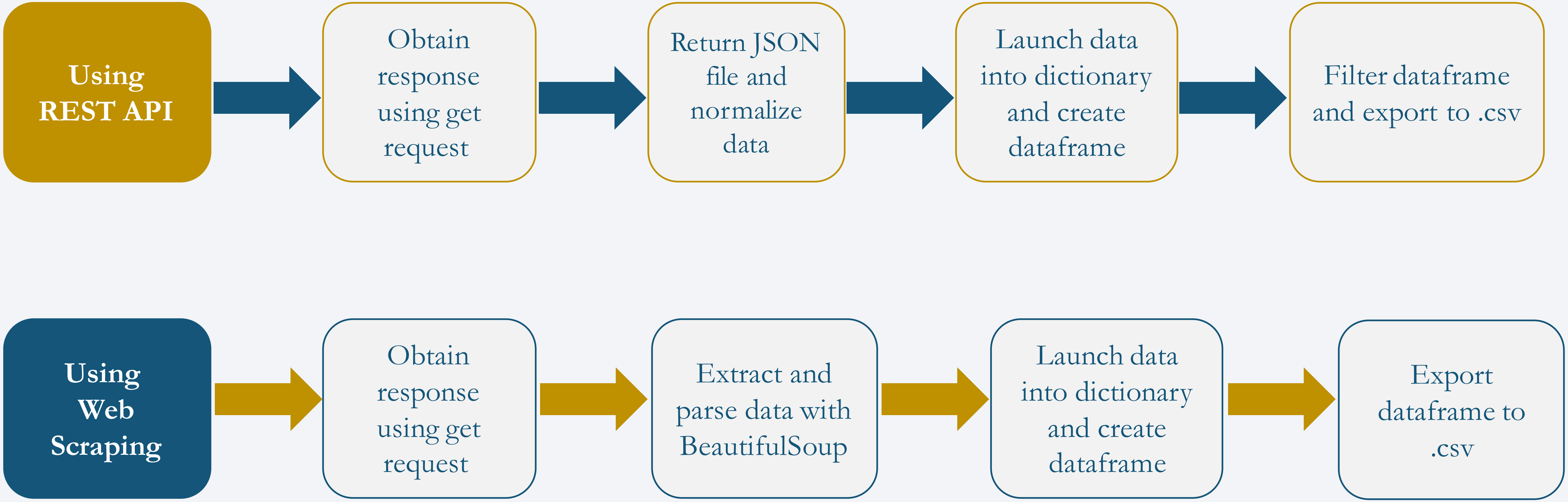
Methodology

Executive Summary

- Data collection methodology
    - SpaceX Rest API
    - Web Scraping from Wikipedia
- Perform data wrangling
    - One Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- 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

The data collection process involved a combination of API requests from SpaceX REST API and Web scraping data from Wikipedia which provided complete information about the launches for a more detailed analysis.



Using REST API

Obtain response using get request

Return JSON file and normalize data

Launch data into dictionary and create dataframe

Filter dataframe and export to .csv

Using Web Scraping

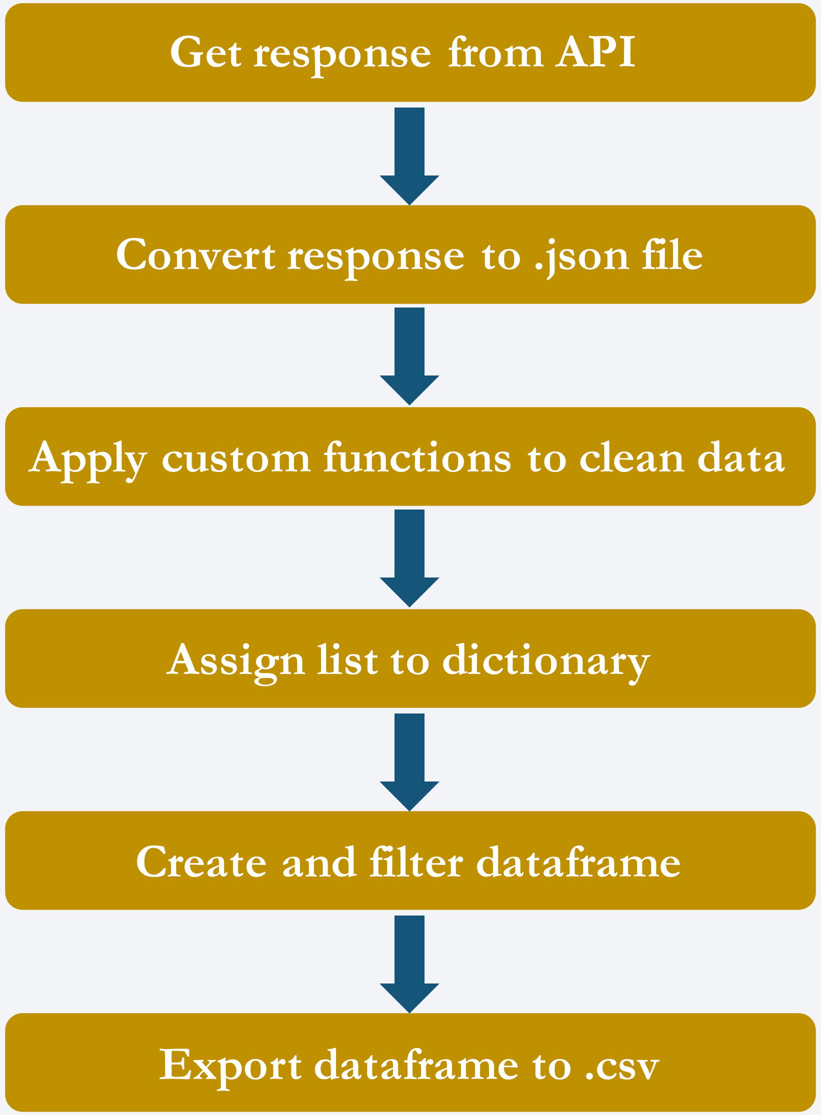
Obtain response using get request

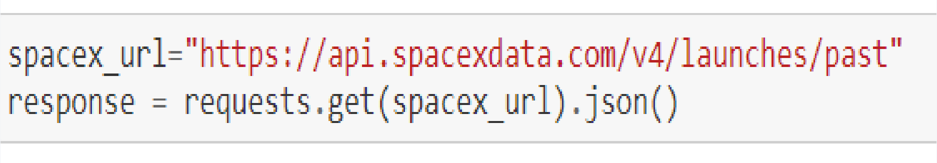
Extract and parse data with BeautifulSoup

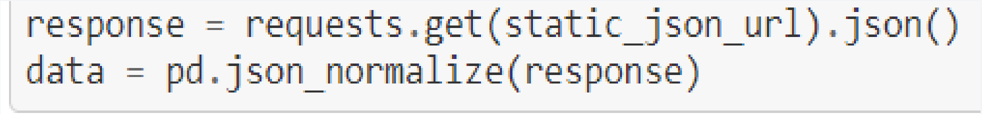
Launch data into dictionary and create dataframe

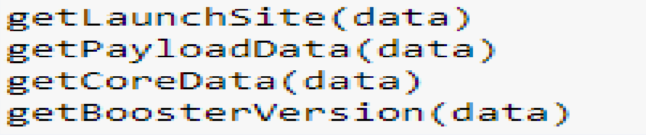
Export dataframe to .csv

Data Collection – SpaceX API

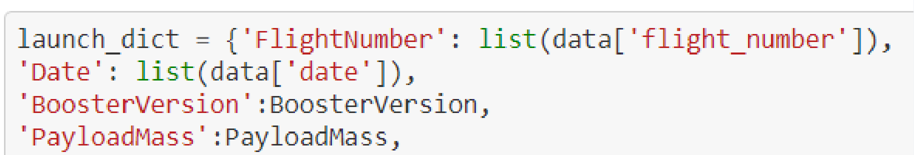

Get response from API
Convert response to .json file
Apply custom functions to clean data
Assign list to dictionary
Create and filter dataframe
Export dataframe to .csv

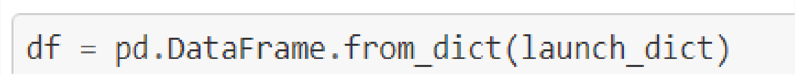

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

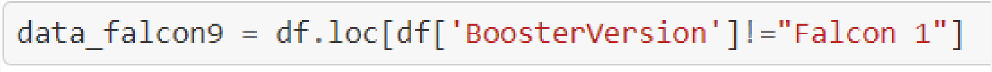

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

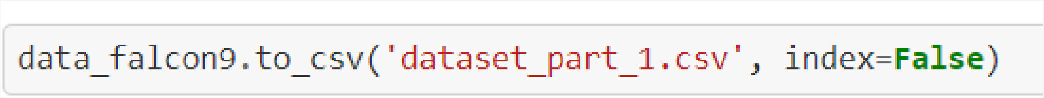

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

[GitHub Link](#)

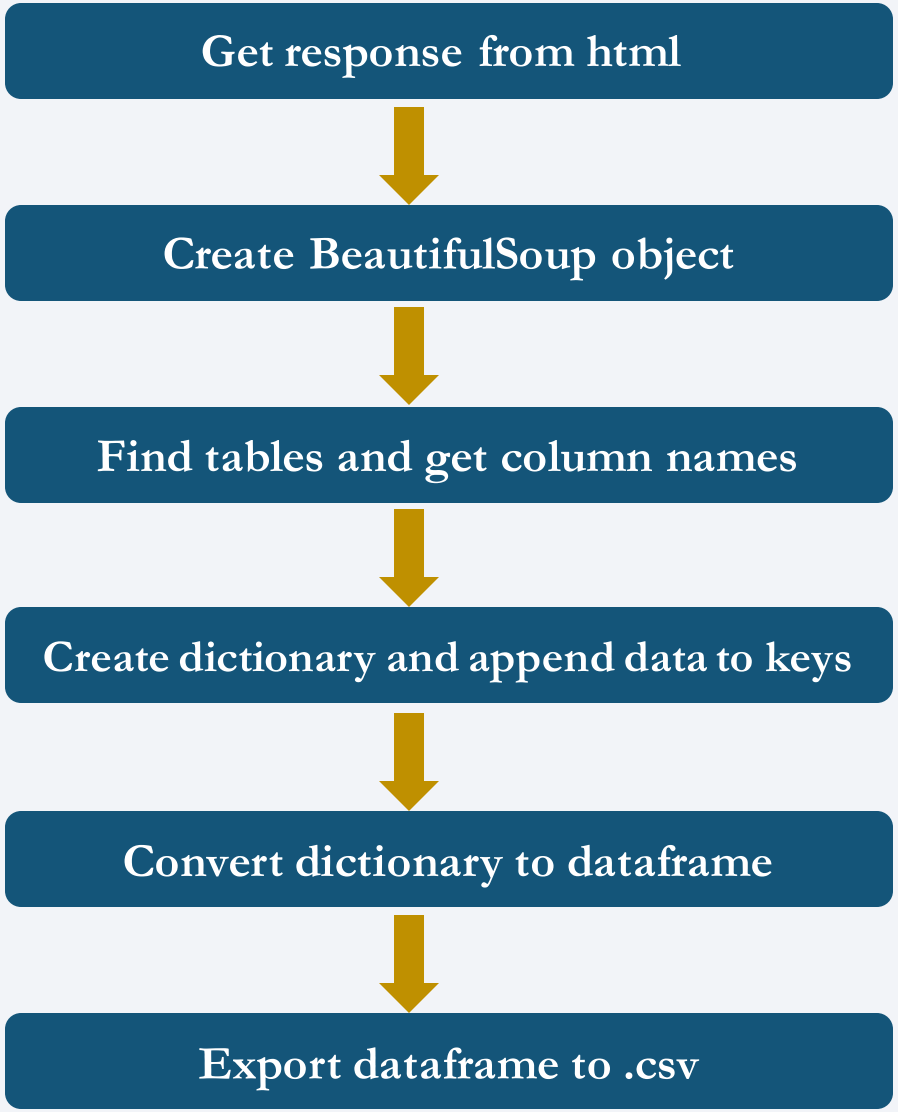

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
```


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


```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```


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

Data Collection - Scraping



Get response from html

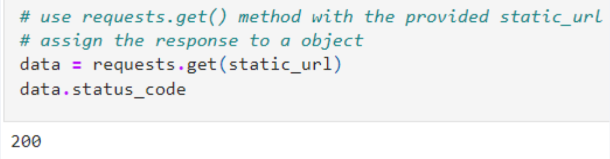
Create BeautifulSoup object

Find tables and get column names

Create dictionary and append data to keys

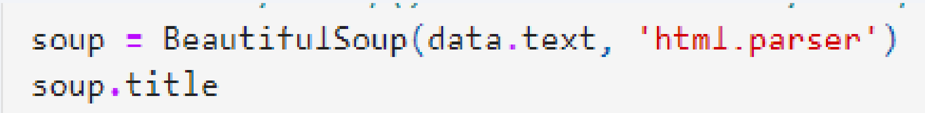
Convert dictionary to dataframe

Export dataframe to .csv

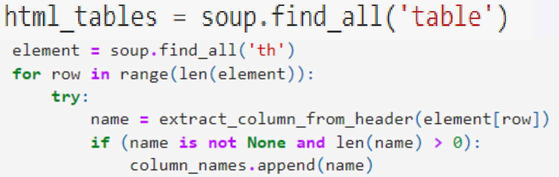


```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url)
data.status_code
```

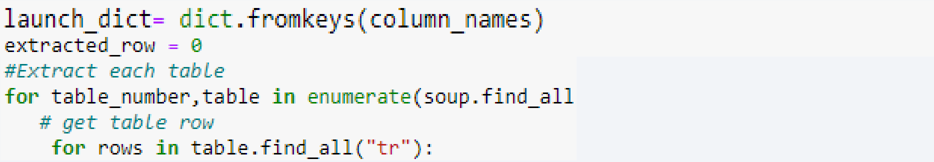
```
200
```



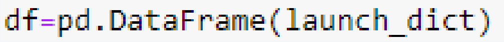
```
soup = BeautifulSoup(data.text, 'html.parser')
soup.title
```



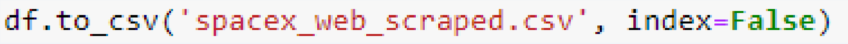
```
html_tables = soup.find_all('table')
element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
```



```
launch_dict= dict.fromkeys(column_names)
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
```



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



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

GitHub Link

Data Wrangling

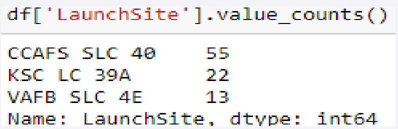
In the dataset, there are several instances where the booster did not land successfully. The outcomes are as defined below -

Successful outcomes - True Ocean, True RTLS, True ASDS

Failed outcomes - False Ocean, False RTLS, False ASDS

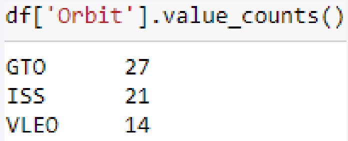
These outcomes will be converted into Training Labels where 1 means success and 0 means failure

Calculate number of launches at each site



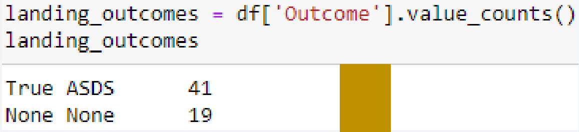
```
df['LaunchSite'].value_counts()

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

Calculate number and occurrence of each orbit



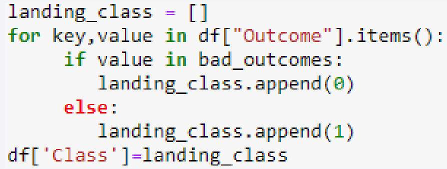
```
df['Orbit'].value_counts()

GTO      27
ISS      21
VLEO     14
```

Calculate number and occurrence of mission outcome per orbit type



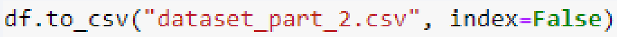
```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

True ASDS      41
None None      19
```

Create landing outcome label from Outcome column



```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

Export dataset as .csv



```
df.to_csv("dataset_part_2.csv", index=False)
```

GitHub Link

EDA with Data Visualization

Scatter Graphs

Scatter plots show the relationship between variables and this is called correlation. The following variables were plotted -

- Flight Number v Payload Mass
- Flight Number v Launch Site
- Payload v Launch Site
- Flight Number v Orbit
- Payload Mass v Orbit

Bar Graph

Bar graphs show the relationship between numeric and categorical variables. The variables below were plotted –

- Success Rate v Orbit

Line Graph

Line graphs clear trends in data variables and often aid in making future predictions. The following were plotted –

- Success Rate v Year

GitHub Link

EDA with SQL

SQL 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 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.
- List the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch sites 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.

GitHub Link

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houston, Texas
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle, folium.map.
  - Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster)
  - Markers to show successful and unsuccessful landings. **Green** for successful landing and **Red** for unsuccessful landing (folium.map.Marker, folium.Icon)
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)

- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

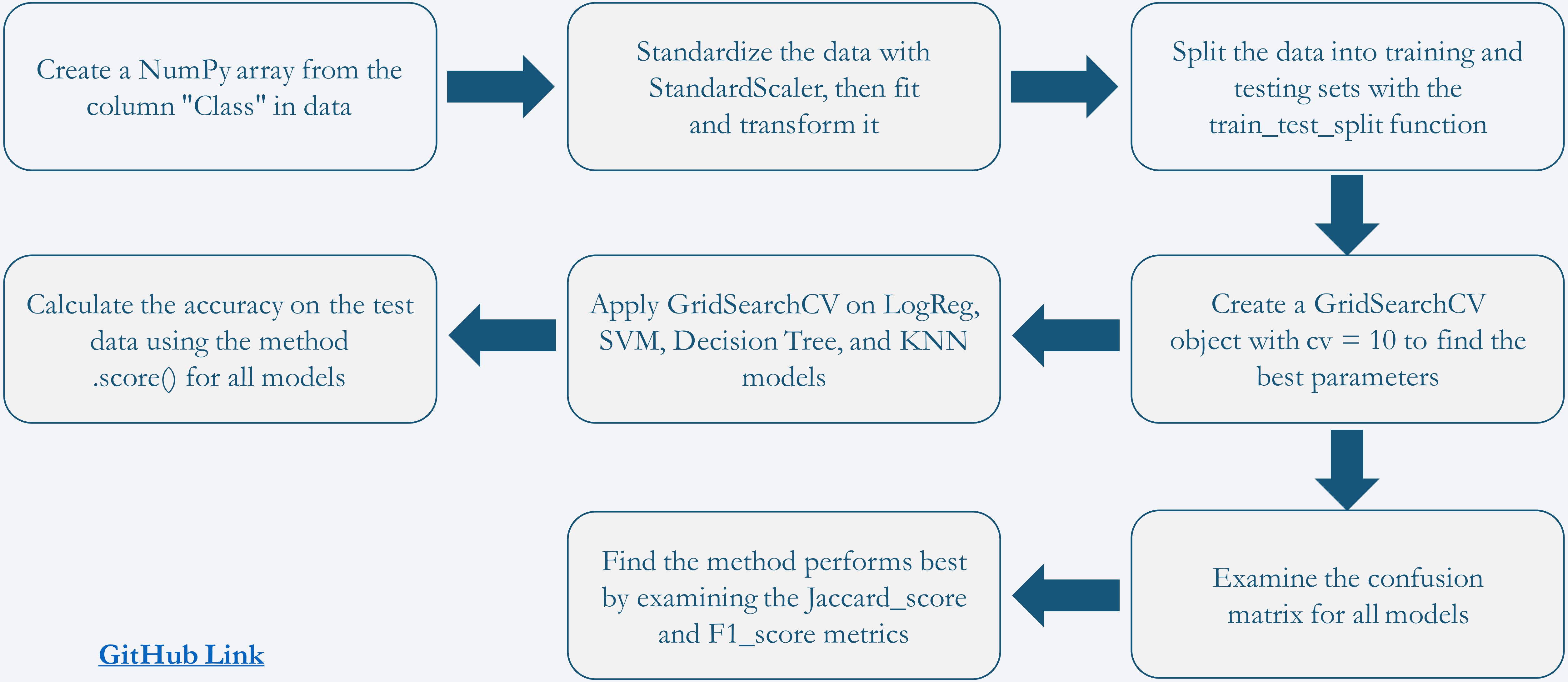
GitHub Link

Build a Dashboard with Plotly Dash

- Launch Sites Dropdown List:
  - Included a dropdown list to enable Launch Site selection.
- Pie Chart showing Success Launches (All Sites/Certain Site):
  - Included a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected.
- Slider of Payload Mass Range:
  - Included a slider to select Payload range.
- Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions:
  - Included a scatter chart to show the correlation between Payload and Launch Success.

GitHub Link

Predictive Analysis (Classification)



Create a NumPy array from the column "Class" in data

Standardize the data with StandardScaler, then fit and transform it

Split the data into training and testing sets with the train_test_split function

Create a GridSearchCV object with cv = 10 to find the best parameters

Apply GridSearchCV on LogReg, SVM, Decision Tree, and KNN models

Calculate the accuracy on the test data using the method .score() for all models

Examine the confusion matrix for all models

Find the method performs best by examining the Jaccard_score and F1_score metrics

GitHub Link

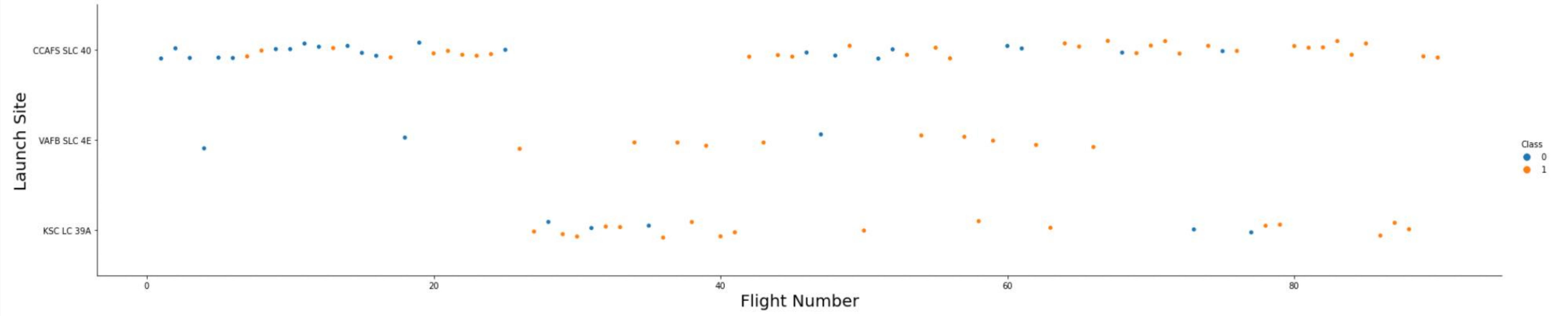
Results

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

Section 2

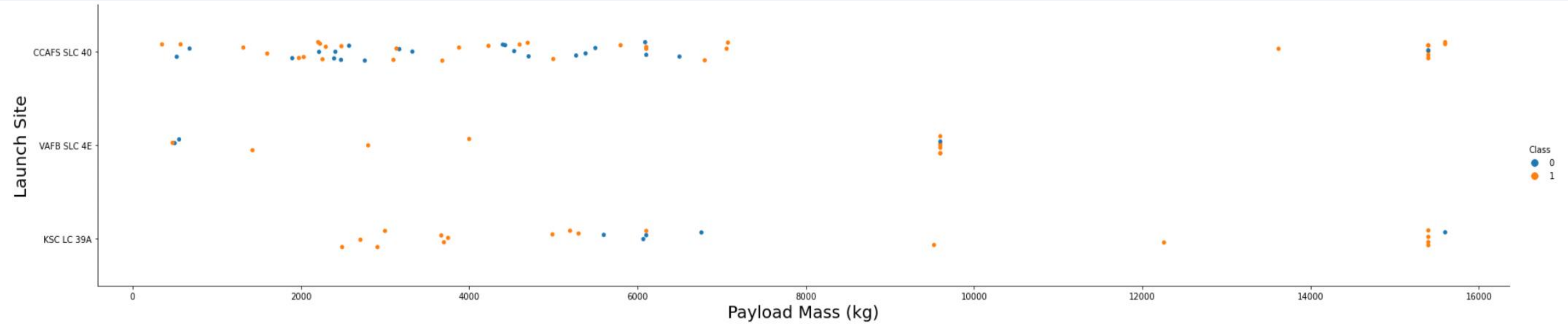
Insights drawn from EDA

Flight Number vs. Launch Site



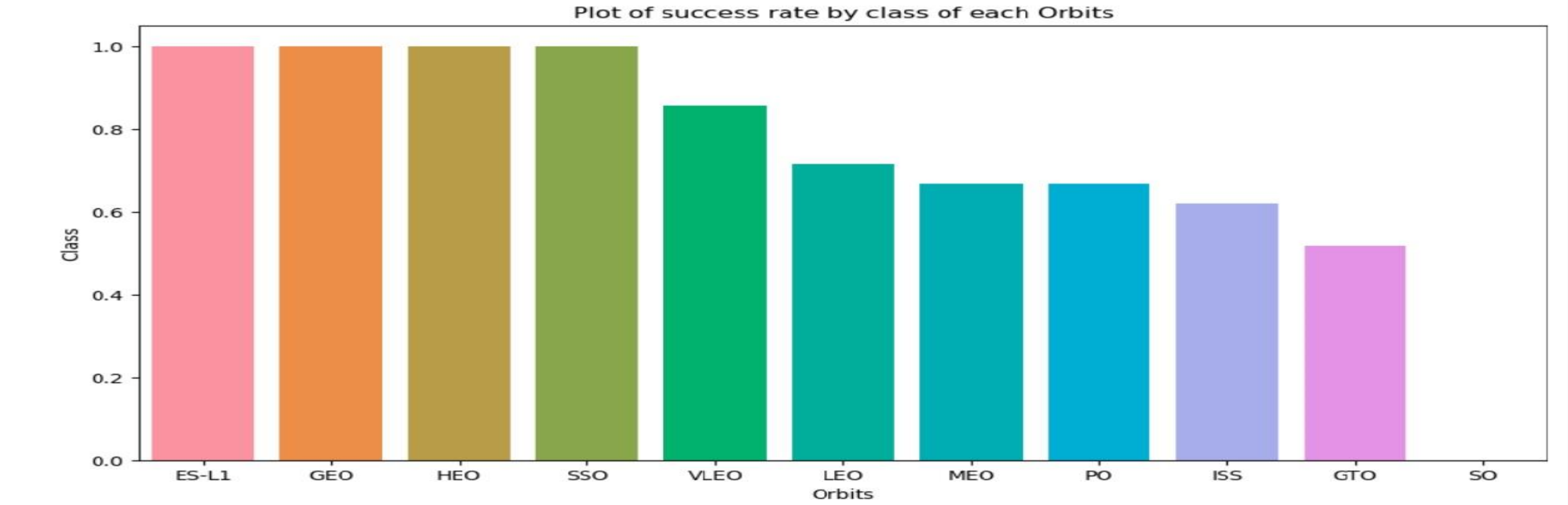
- The earliest flights all failed while the latest flights all succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.
- It can be assumed that each new launch has a higher rate of success

Payload vs. Launch Site



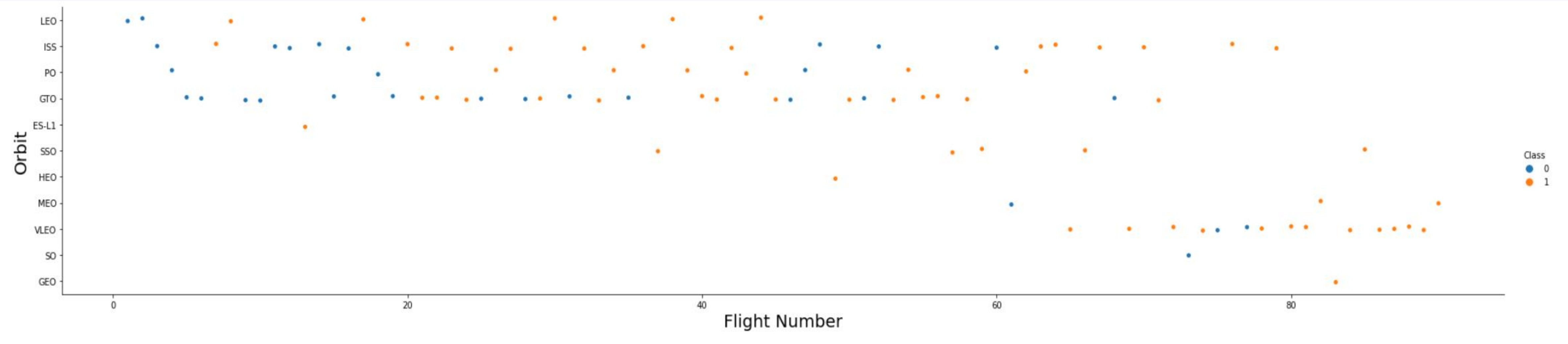
- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg as well

Success Rate vs. Orbit Type



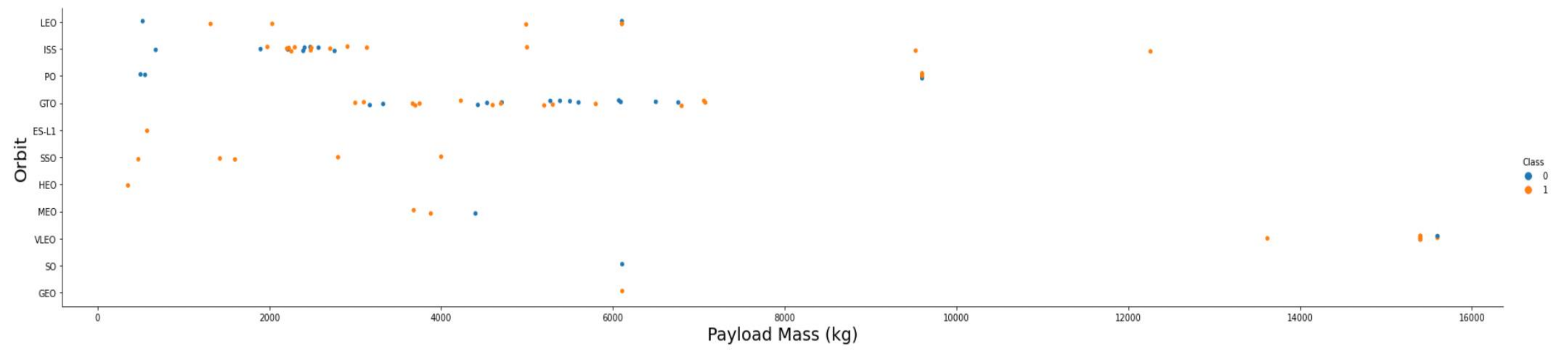
- Orbits with 100% success rate are ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate – SO
- Orbits with success rate between 50% and 85% are GTO, ISS, LEO, MEO, PO

Flight Number vs. Orbit Type



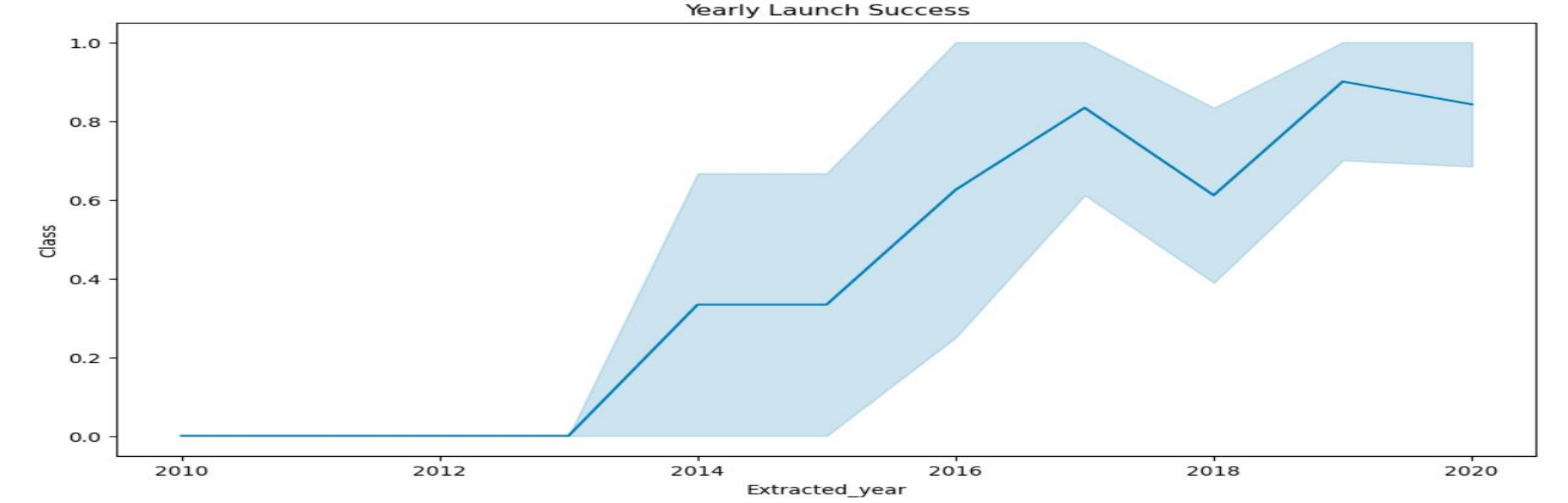
- In the LEO orbit the Success appears related to the number of flights and there seems to be no relationship between flight number when in GTO orbit

Payload Mass vs. Orbit Type



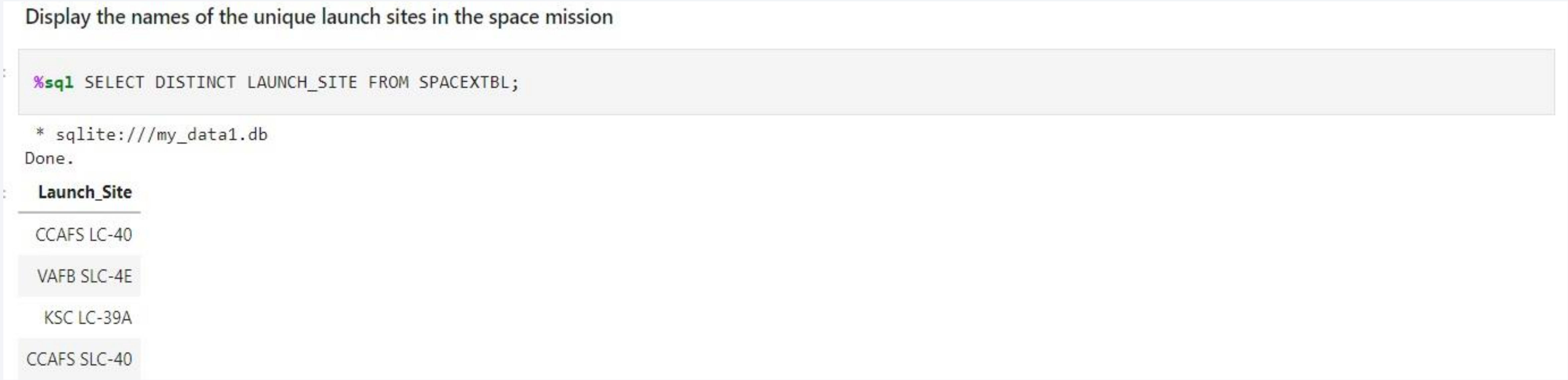
- Heavy payloads have a negative influence on GTO orbit and a positive influence on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



- The success rate received a significant boost in 2013 and rapidly increased till 2020

All Launch Site Names



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

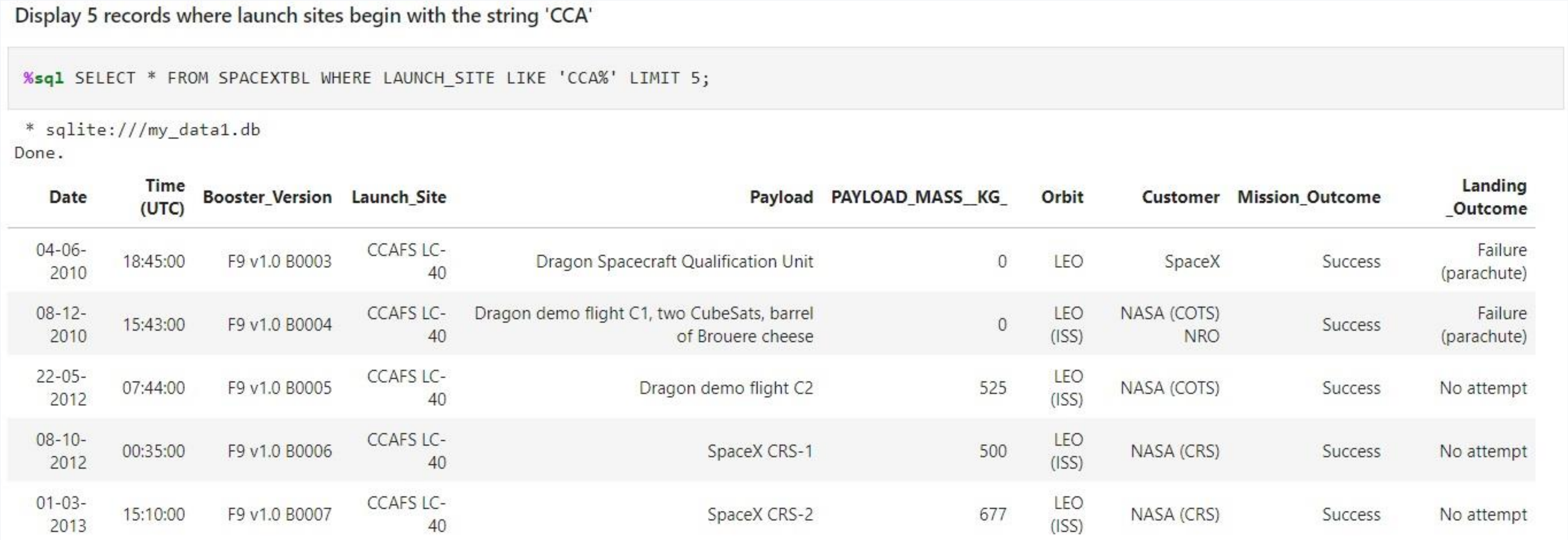
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

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

- The keyword Distinct was used to show unique launch sites from the SpaceX data

Launch Site Names Begin with 'CCA'



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

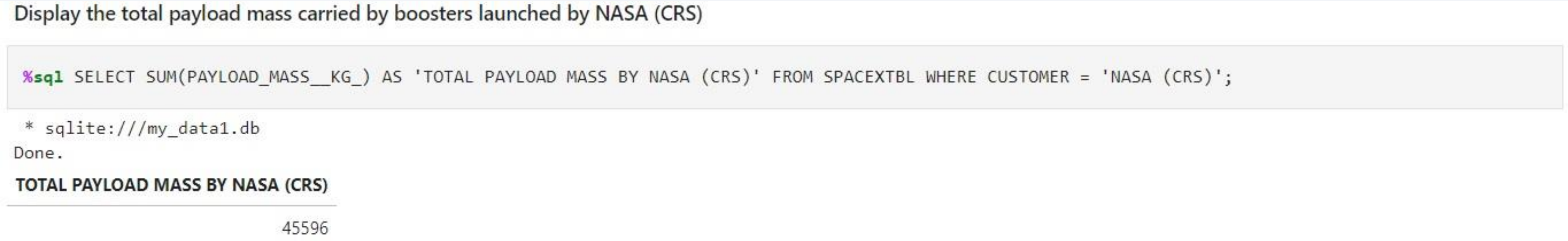
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The keywords LIKE 'CCA%' and LIMIT 5 were entered into the query following the WHERE clause to generate 5 launch sites beginning with CCA

Total Payload Mass



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

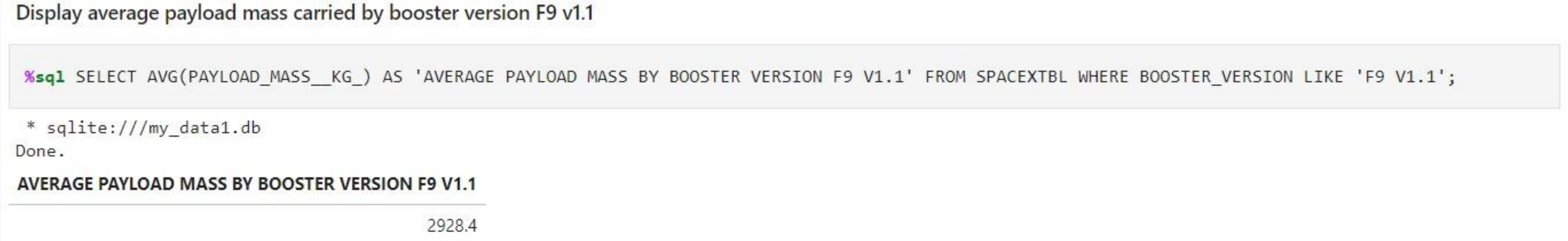
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS 'TOTAL PAYLOAD MASS BY NASA (CRS)' FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

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

TOTAL PAYLOAD MASS BY NASA (CRS)
45596

- The WHERE clause was used to filter the query to return total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1



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

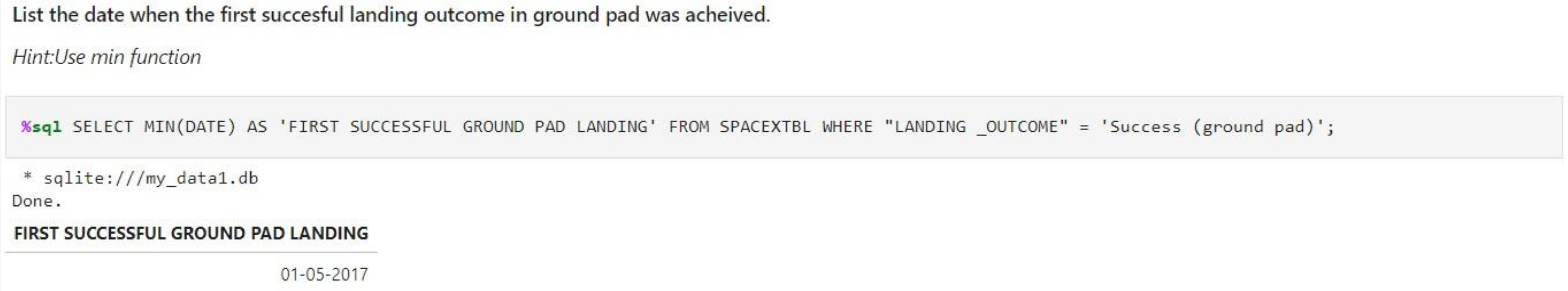
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS 'AVERAGE PAYLOAD MASS BY BOOSTER VERSION F9 V1.1' FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 V1.1';
```

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

AVERAGE PAYLOAD MASS BY BOOSTER VERSION F9 V1.1
2928.4

- The WHERE clause was used to filter the query to return the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date



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

Hint:Use min function

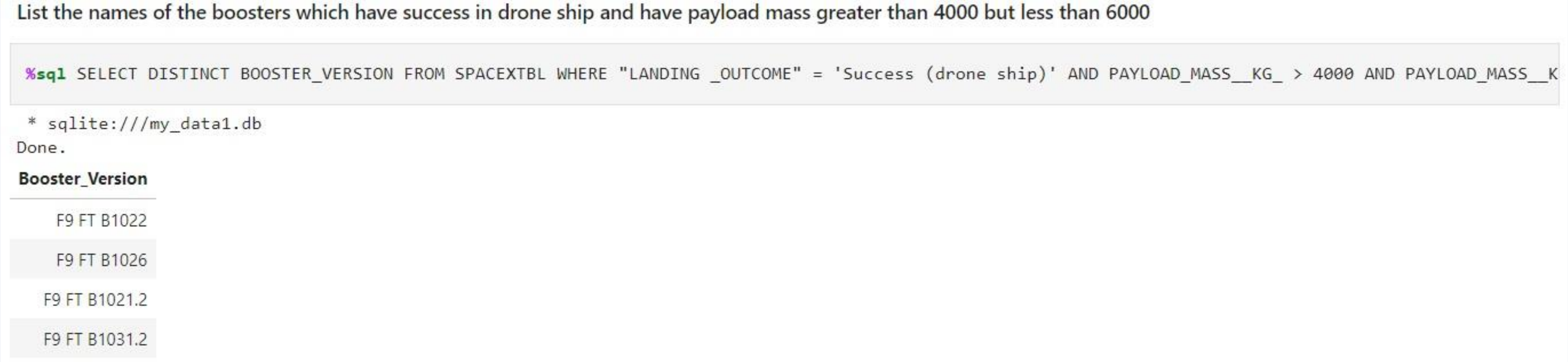
```
%sql SELECT MIN(DATE) AS 'FIRST SUCCESSFUL GROUND PAD LANDING' FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (ground pad)';
```

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

FIRST SUCCESSFUL GROUND PAD LANDING
01-05-2017

- The MIN function returns the oldest date from the SpaceX and the WHERE clause filtered successful landings. The above query generates the oldest successful landing date from the data

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

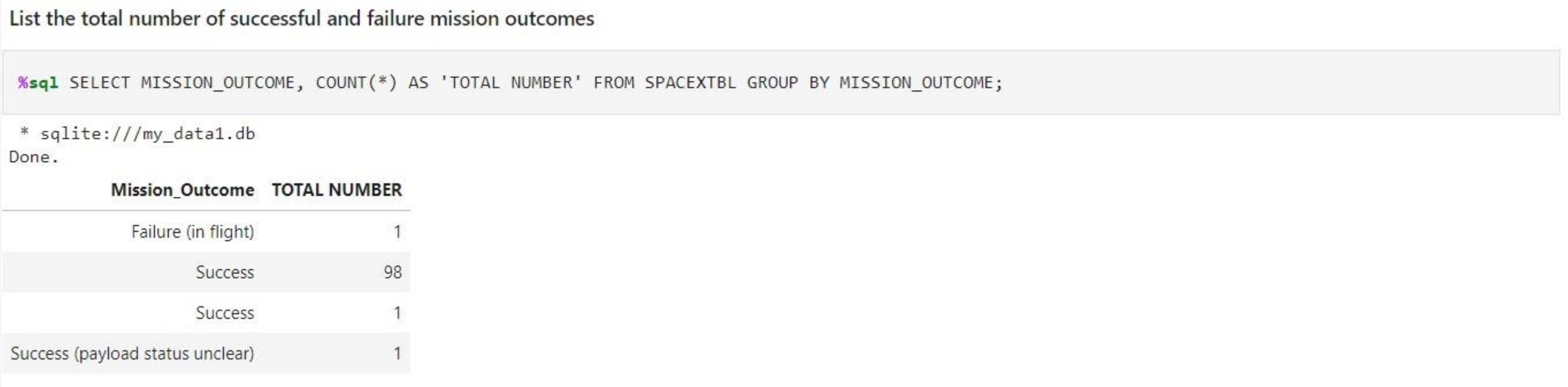
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__K
```

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

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

- The above query returns unique booster versions with successful landings and payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

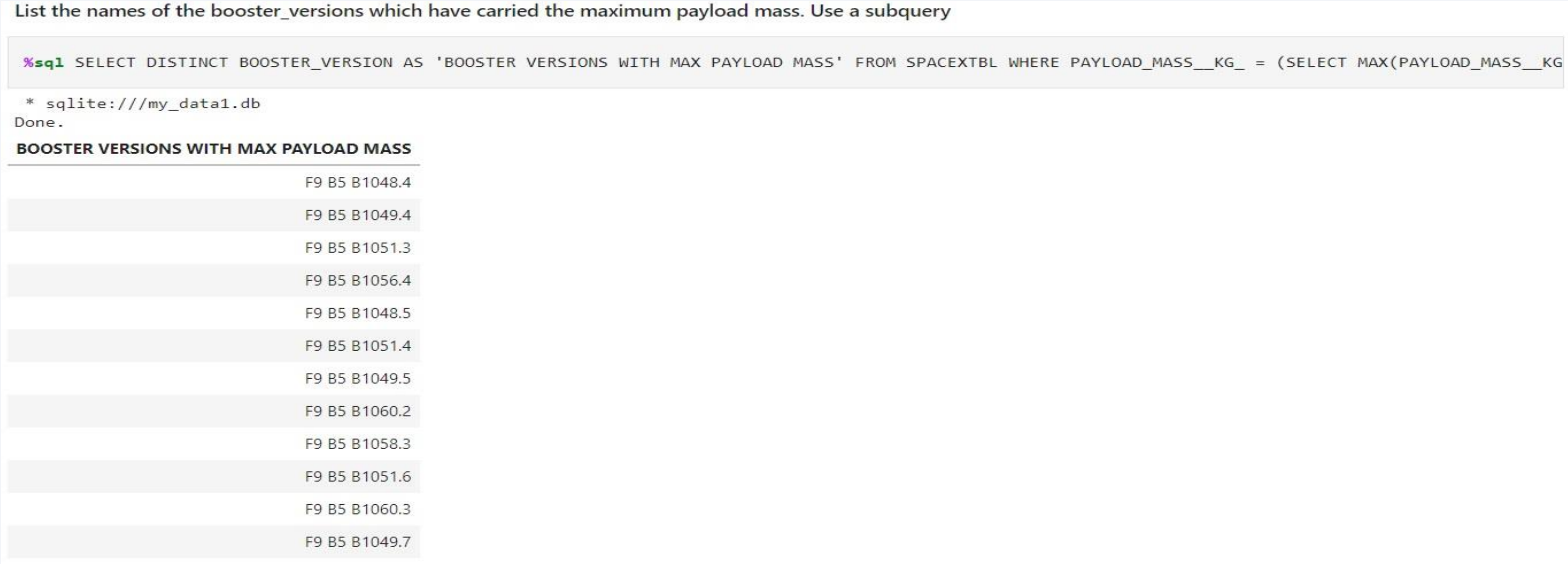
```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS 'TOTAL NUMBER' FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

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

- The query above shows the number of successful and failed missions

Boosters Carried Maximum Payload



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

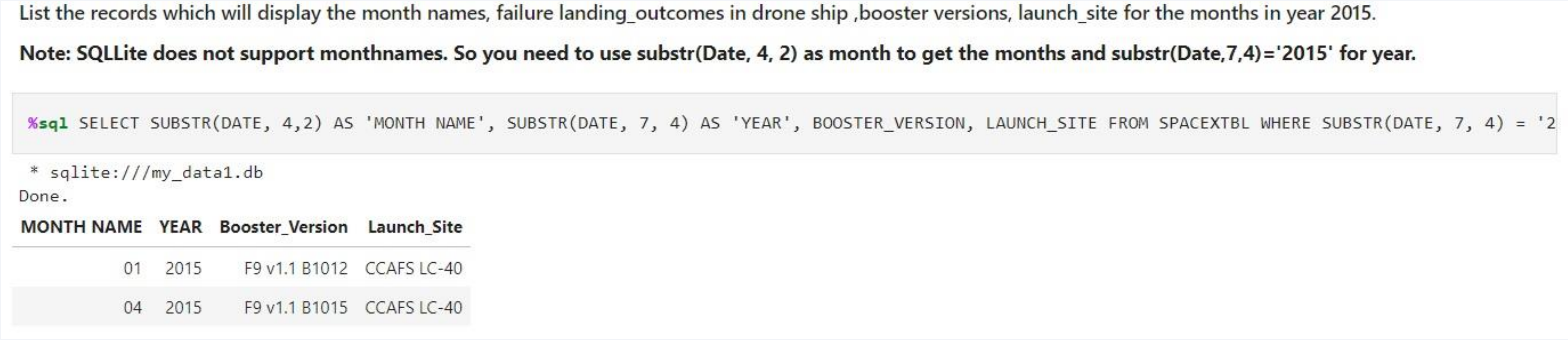
```
%sql SELECT DISTINCT BOOSTER_VERSION AS 'BOOSTER VERSIONS WITH MAX PAYLOAD MASS' FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG
```

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

BOOSTER VERSIONS WITH MAX PAYLOAD MASS
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

- Using the with MAX function, a subquery was written to filter data by returning only the heaviest payload mass
- The main query uses subquery results and returns unique booster versions with the heaviest payload mass.

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.

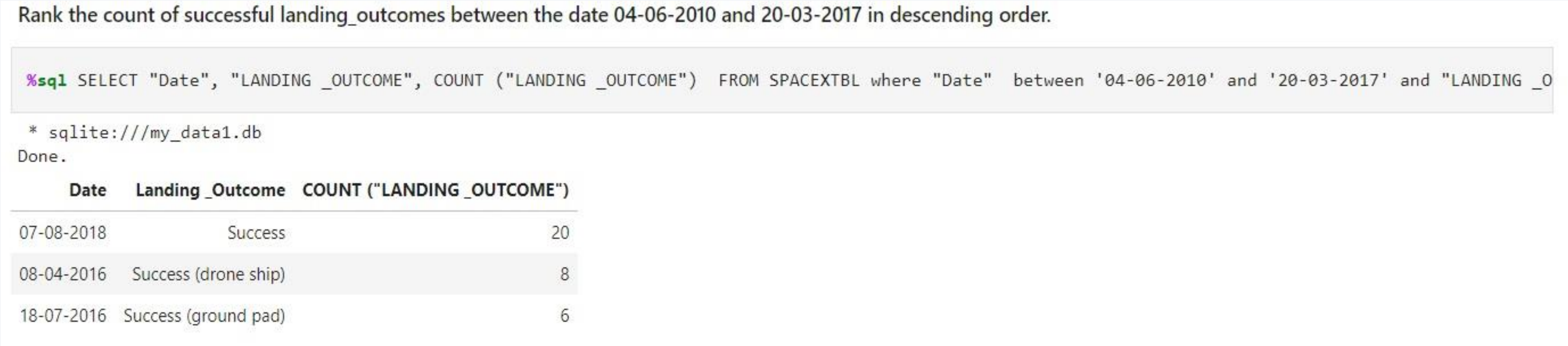
```
%sql SELECT SUBSTR(DATE, 4,2) AS 'MONTH NAME', SUBSTR(DATE, 7, 4) AS 'YEAR', BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE SUBSTR(DATE, 7, 4) = '2
```

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

MONTH NAME	YEAR	Booster_Version	Launch_Site
01	2015	F9 v1.1 B1012	CCAFS LC-40
04	2015	F9 v1.1 B1015	CCAFS LC-40

- The above query returns the month, booster version and launch site with failed landing outcomes in year 2015
- Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows the month and Substr(DATE,7, 4) shows the year.

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



Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT "Date", "LANDING _OUTCOME", COUNT ("LANDING _OUTCOME")  FROM SPACEXTBL where "Date"  between '04-06-2010' and '20-03-2017' and "LANDING _O
```

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

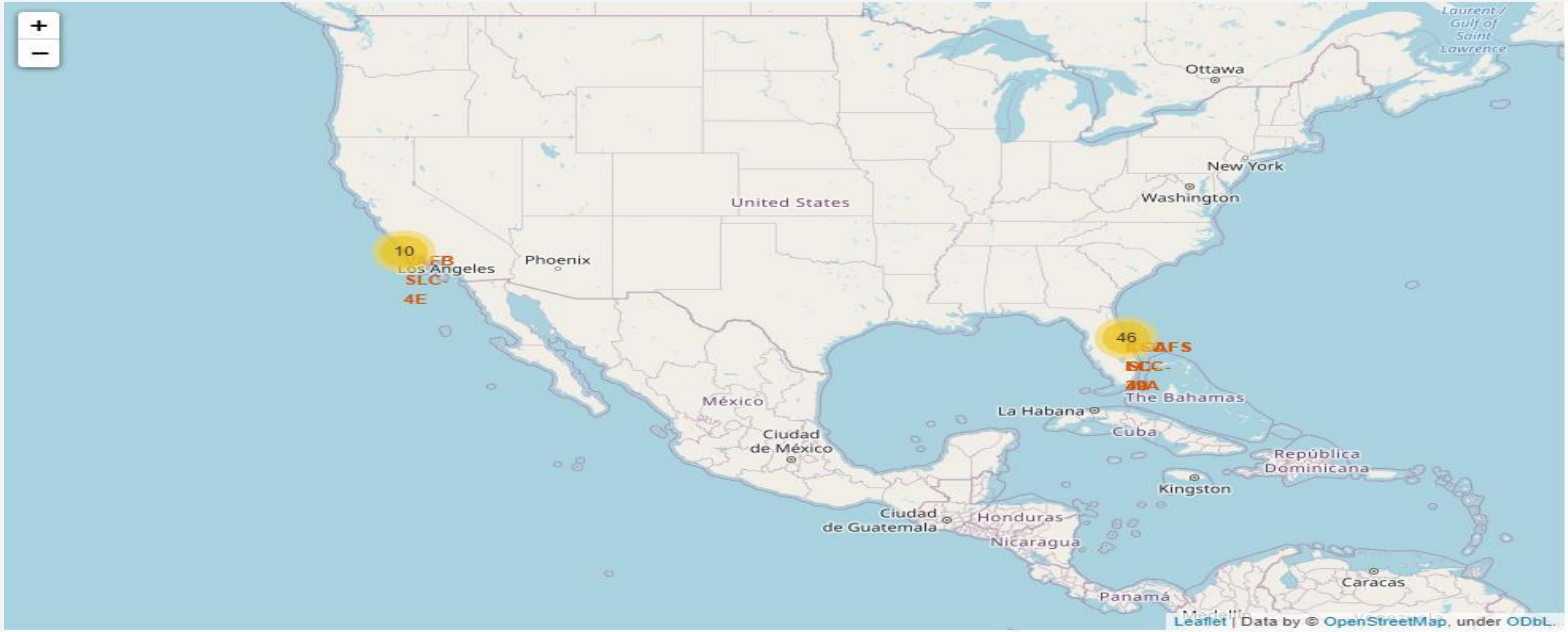
Date	Landing _Outcome	COUNT ("LANDING _OUTCOME")
07-08-2018	Success	20
08-04-2016	Success (drone ship)	8
18-07-2016	Success (ground pad)	6

- The above query returns the number of successful landing outcomes between 04/06/2010 and 20/03/2017
- The GROUP BY clause groups results by landing outcome
- The ORDER BY COUNT DESC shows results in decreasing order.

Section 3

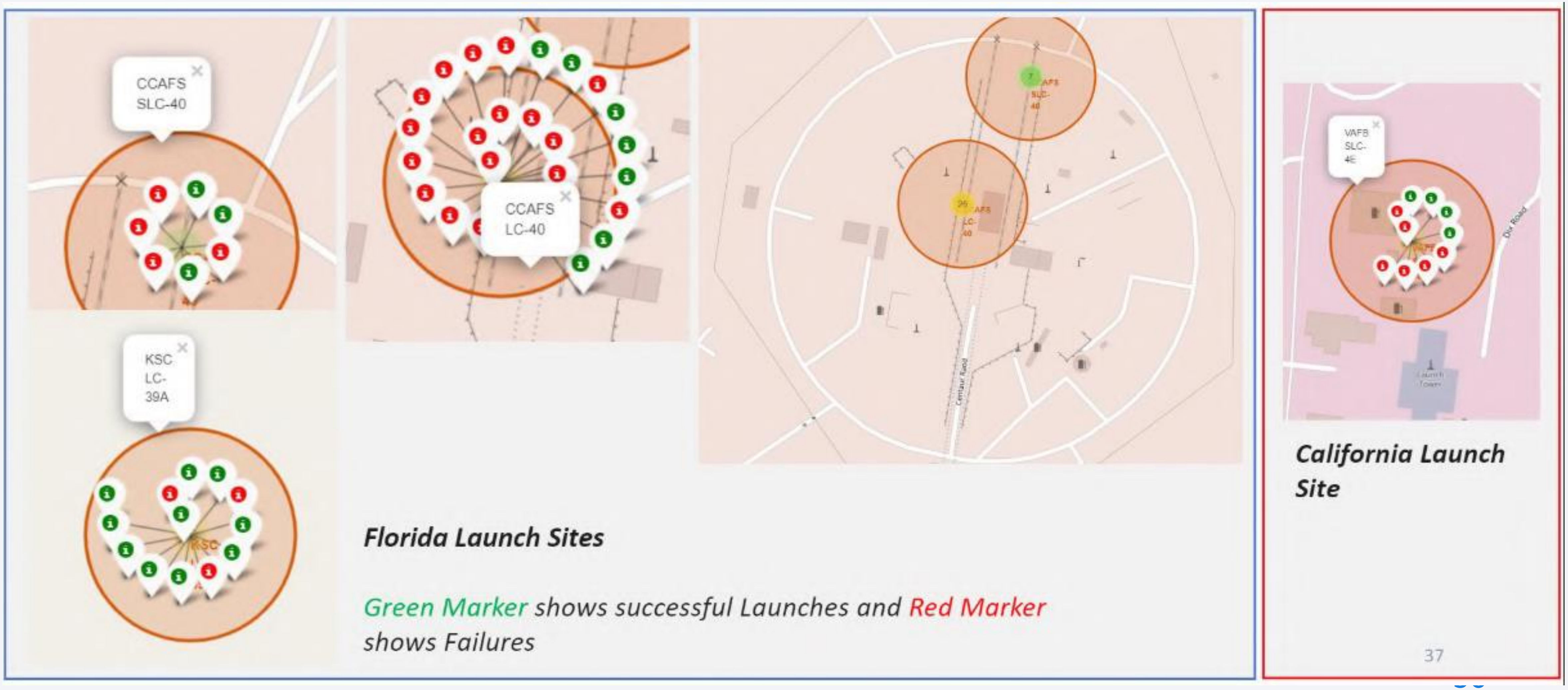
Launch Sites Proximities Analysis

Folium Map – Launch Sites



- All SpaceX launch sites are on the coast of the United States

Folium Map – Colour Coded Launch Markers

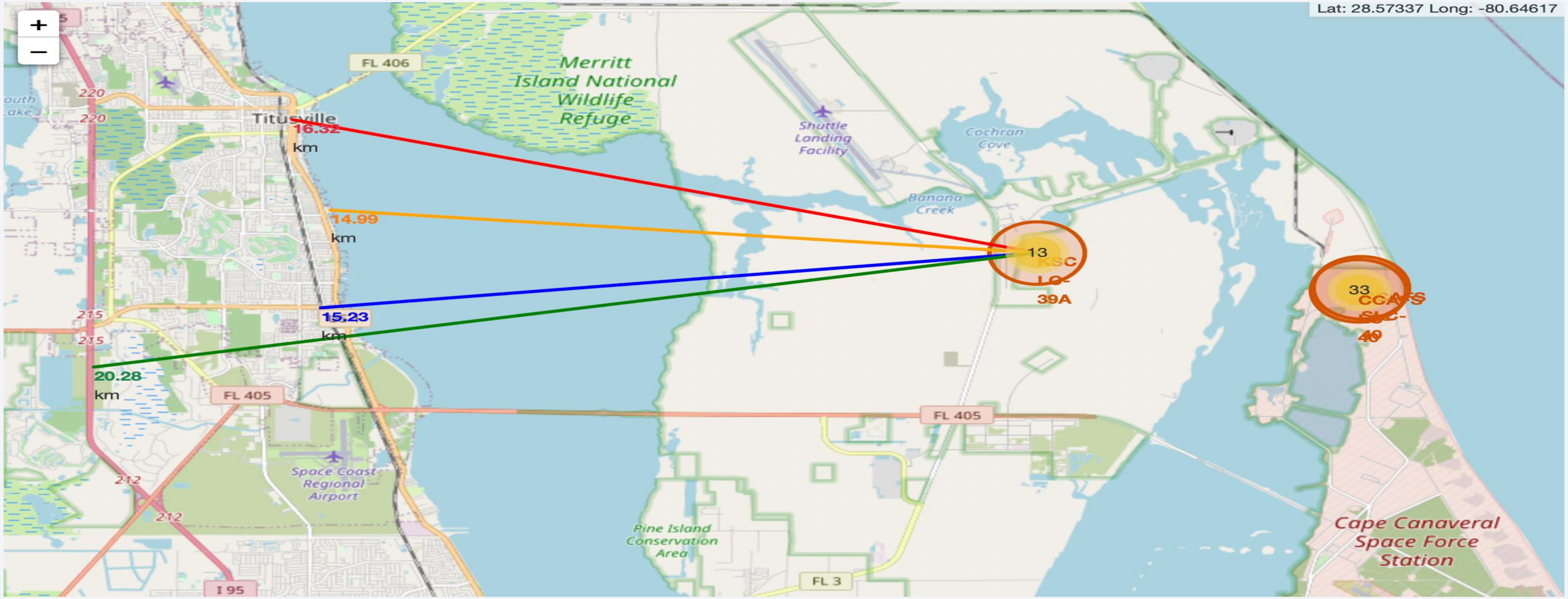


Florida Launch Sites

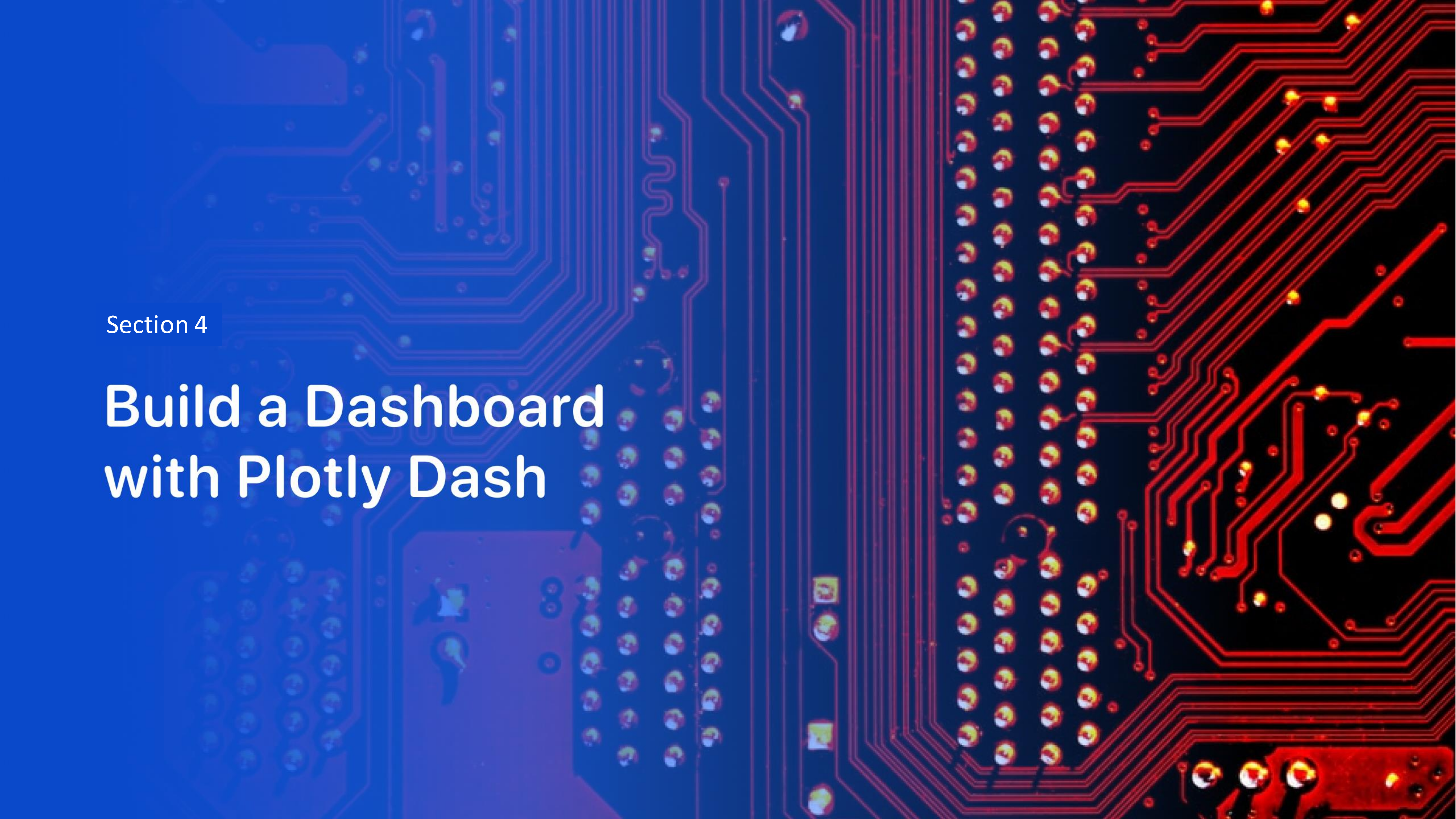
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

Folium Map - Distance Between KSC LC-39A and Its Proximities



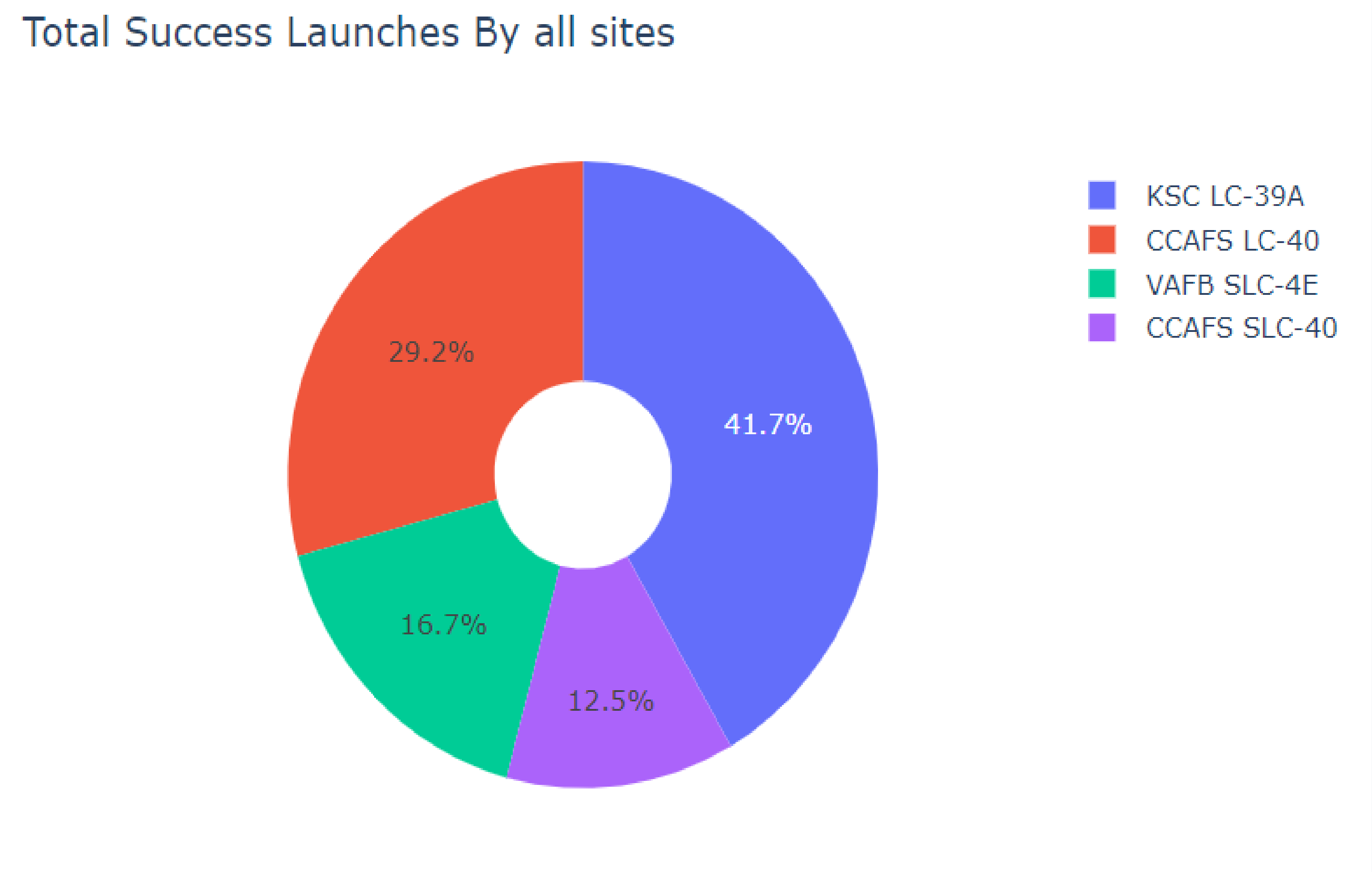
- Is KSC LC-39A in close proximity to railways ? Yes
- Is KSC LC-39A in close proximity to highways ? Yes
- Is KSC LC-39A in close proximity to coastline ? Yes
- Is KSC LC-39A far from cities ? No



Section 4

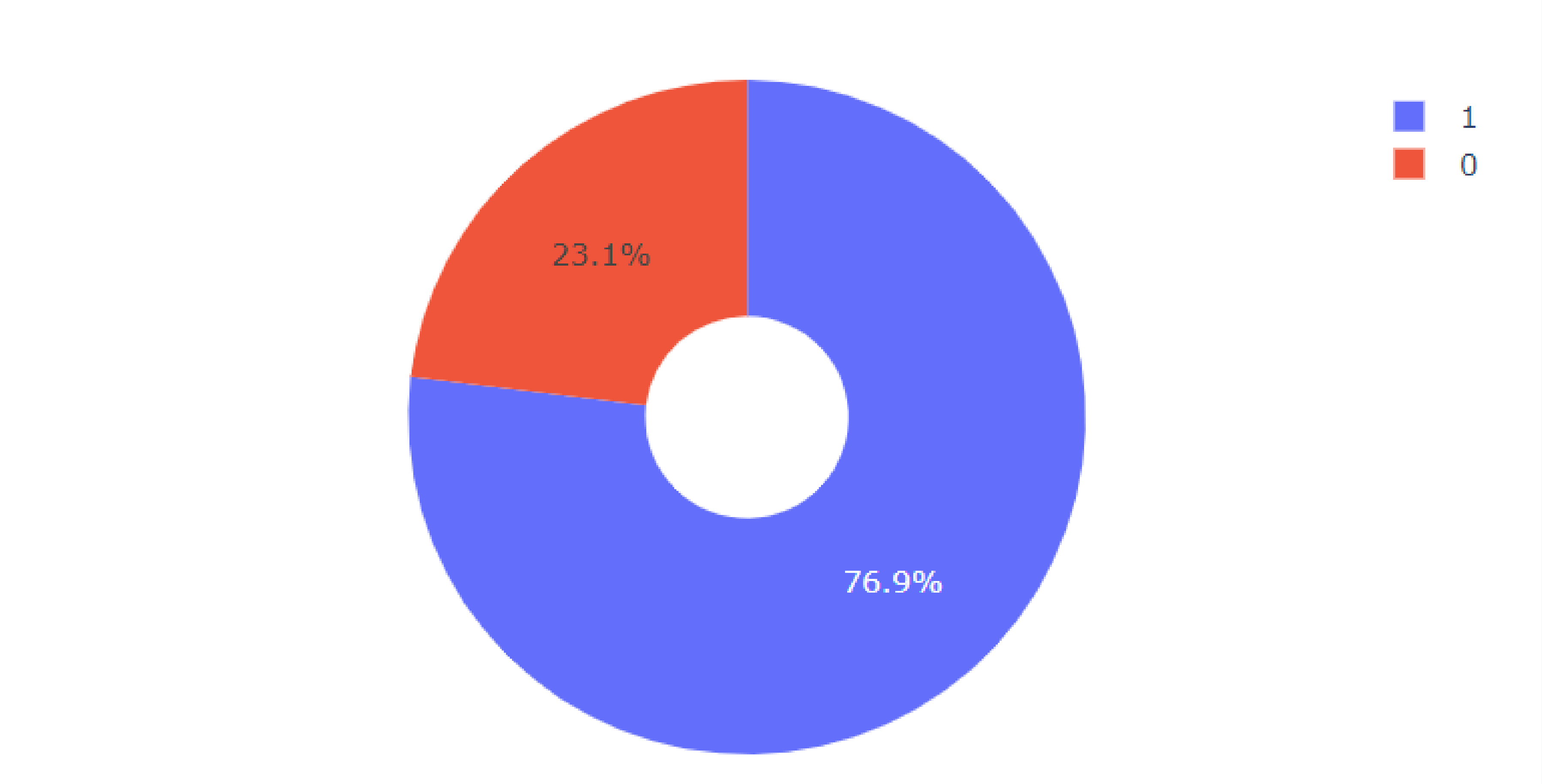
Build a Dashboard with Plotly Dash

Successful Launches by Site



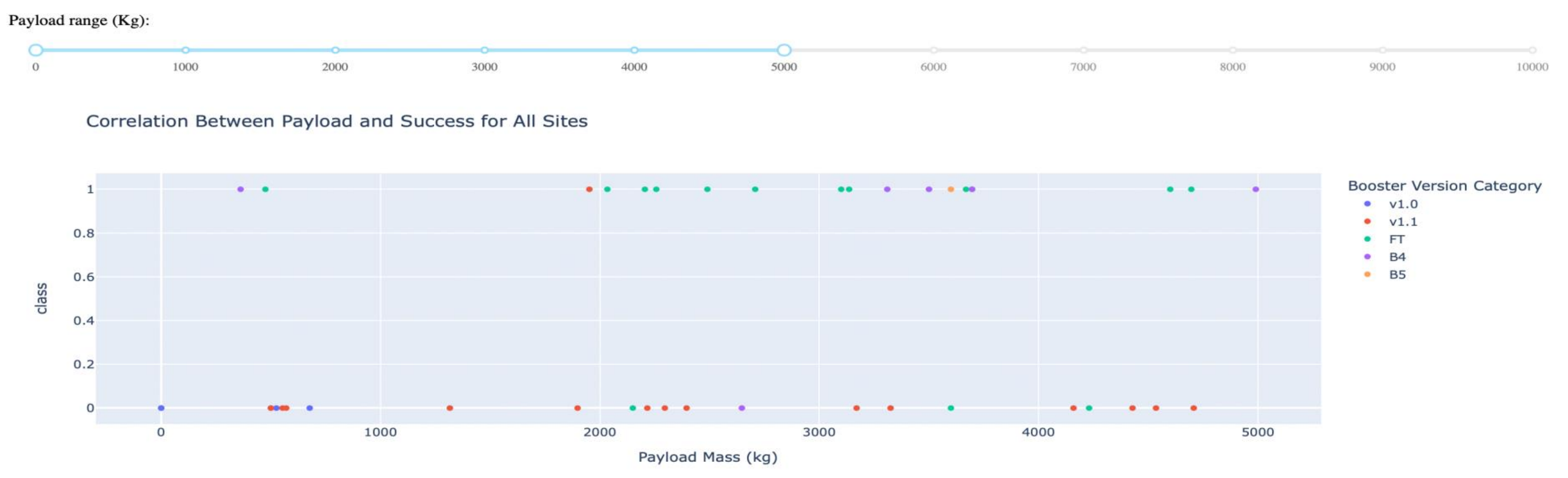
- KSC LC 39A has the best success rate of launches.

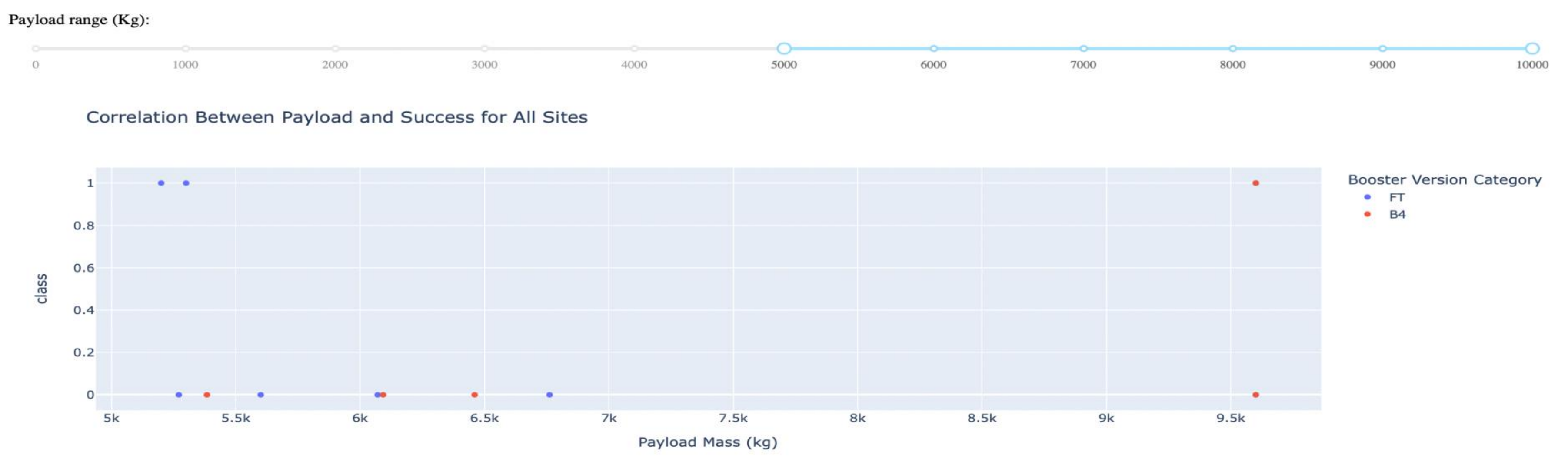
Most Successful Launch Site – Success Ratio



- KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

Payload vs Launch Outcome Scatter Plot





- Low weighted payloads have a better success rate than the heavy weighted payloads
- Payloads between 2000 and 5500 kg have the highest success rate.

Section 5

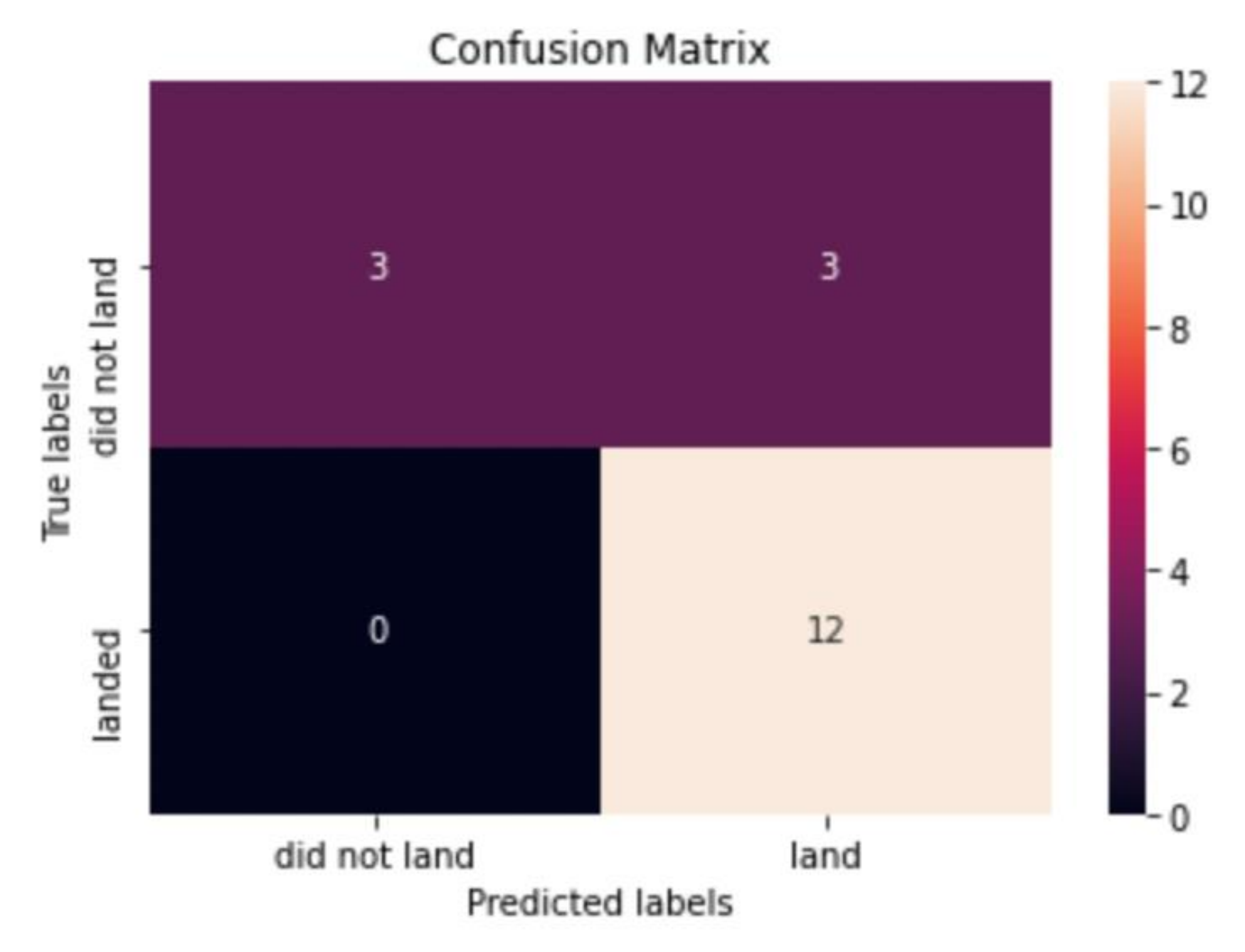
Predictive Analysis (Classification)

Classification Accuracy

Method	Accuracy Train	Accuracy Test
Logistic Regression	0.8472222222222222	0.8333333333333334
SVM	0.8472222222222222	0.8333333333333334
Decision Tree	0.8888888888888888	0.8333333333333334
KNN	0.8472222222222222	0.8333333333333334

- The accuracy scores on the test data are basically the same. The decision tree can be deemed to be the best performing method though with its score of approximately 0.89 when using the train dataset.

Confusion Matrix



- All the methods have the same test accuracy scores so the confusion matrix is the same
- The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier

Conclusion

- The Decision Tree Model is the best algorithm for this dataset because of its superior train accuracy
- Low weighted payloads perform better than the heavy weighted payloads
- The success rate of launches is proportional to time spent on missions
- KSC LC-39A has the highest success rate of the launches – 76.9%
- Orbits GEO, HEO, SSO and ES-L1 have the best success rates.

Thank you!