IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Debosmita Chakraborty
11/06/2023

Outline

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

Executive Summary

This report deals with the scenario, methodology, results and findings of the analysis of SpaceX Falcon 9 launch data. For this project, data collection was done using the SpaceX API and web scraping a Wikipedia webpage for Falcon 9 launch details. Then data wrangling and transformation was done, followed by data visualization using Python libraries, Folium map and an interactive dashboard built using Plotly Dash. Classification models of different kinds were trained on the data after choosing the optimum hyperparameters and their performance was then evaluated. The model, EDA and further data visualizations revealed many insights into the various factors involved in a successful launch.

Introduction

- This project was to analyze rocket launches of the SpaceX Falcon 9 rocket and predict whether the first stage of the rocket lands successfully, in which case it can be reused and the cost of the rocket launch can be determined accordingly.
- The purpose of this project is to derive information that can be used by a rival company to bid against SpaceX for a rocket launch.

The main question we are trying to answer is: *"Given the features of the rocket launch such as payload mass, launch site, etc., can we predict if the first stage lands successfully?"*

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:** Data was collected using the SpaceX API and web scraping historical launch records
- **Data wrangling:** Data was processed by dealing with null values, one-hot encoding, etc.
- **Exploratory data analysis (EDA):** EDA was done using SQL and Python libraries such as Matplotlib and Seaborn
- **Interactive visual analytics:** Done using Folium for geographical mapping and Plotly with Dash for interactive dashboarding
- **Predictive analysis using classification models:** Various models were tested with optimal hyperparameters for performance comparison

Data Collection

- Data was collected from primarily 2 sources:

  1. the SpaceX API

  2. a Wikipedia webpage containing historical launch data for the Falcon 9 rocket.

- The API used is: https://api.spacexdata.com/v4/rockets/
- The webpage used is: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- For consistency, a snapshot of the webpage that was updated on 9th June 2021 was used as the source.

Data Collection – SpaceX API

- The requests library was used to make HTTP requests to the API. We passed the relevant URL for past launch data via GET method to get the response which was returned as a JSON file.
- It was then decoded using .json() method and converted into a Pandas dataframe using .json_normalize()
- As data is primarily in the form of rocket ids, we request the API for each id to get information regarding the rocket booster name, payload, launchpads, cores etc.
- The data received is stored in lists, then a dictionary and finally combined to form a new dataframe.
- External reference: [GitHub link to API notebook]

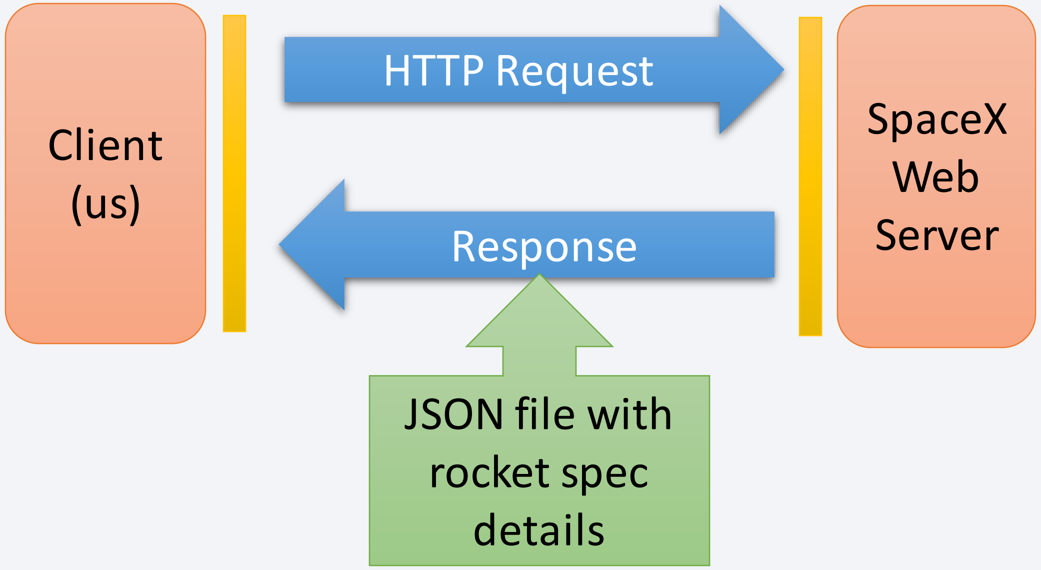


Fig 1: Flowchart of REST API call

Data Collection - Scraping

- The requests library was used to make HTTP request and the HTML response was stored in a BeautifulSoup object.
- We locate the target table and extract the column names while iterating through each <th> element; then store them in a list
- We create an empty dictionary with keys matching the column names from the list
- We iterate through each <td> data element in each table row <tr> (parsing) to extract and append data to the dictionary, which is finally converted into a Pandas dataframe
- External reference: [GitHub link to webscraping notebook]()

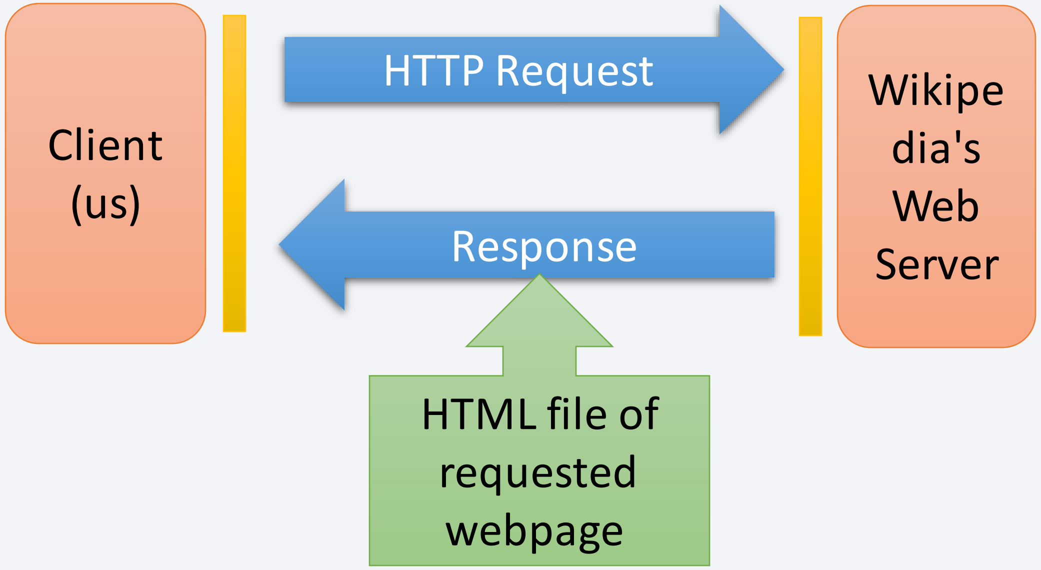
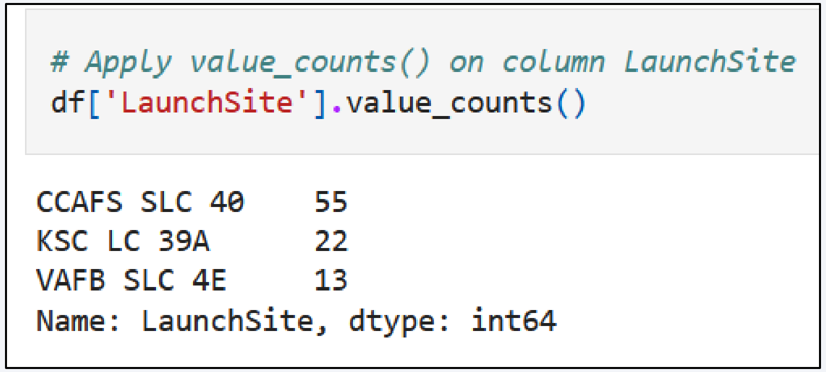


Fig 2: Flowchart for HTTP request

Data Wrangling

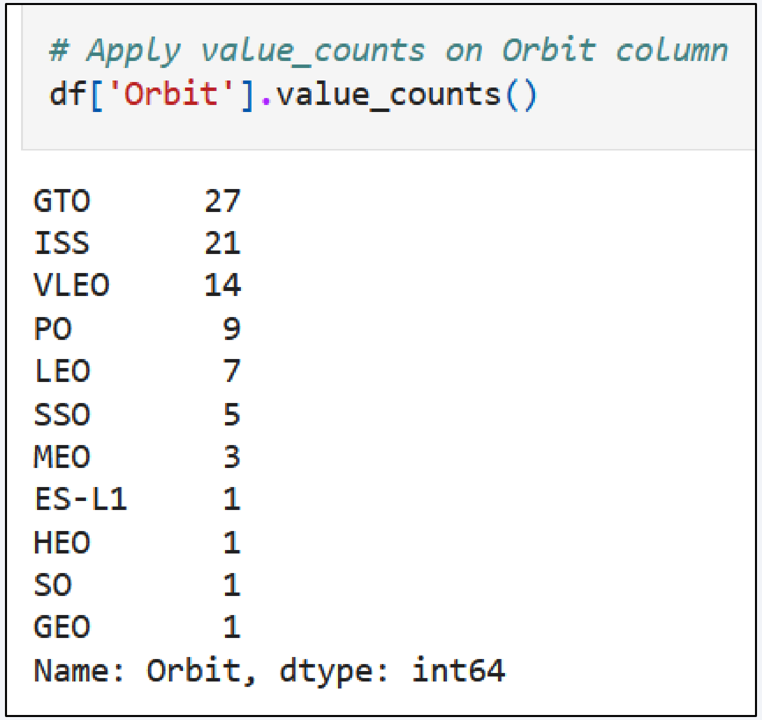
- We first computed the percentage of missing values in each attribute, along with observing the datatype of each.
- We then used value_counts() method to find:
  1. Number of launches on each site (on 'Launch Site' column)
  2. Number and occurrence of each orbit (on 'Orbit' column)
  3. Number and occurrence of mission outcomes (on 'Outcome column') + store in variable landing_outcomes
- From the created landing_outcomes we take a subset bad_outcomes of unsuccessful landings, which is later used to create a list of outcomes- 0 for failure (present in bad_outcome), 1 for success (One-Hot encoding)
- This list represents the classification variable (later added to dataframe as 'class' column). The mean of this column gives the success rate 66.67%
- External reference: GitHub link to data wrangling notebook

Data Wrangling



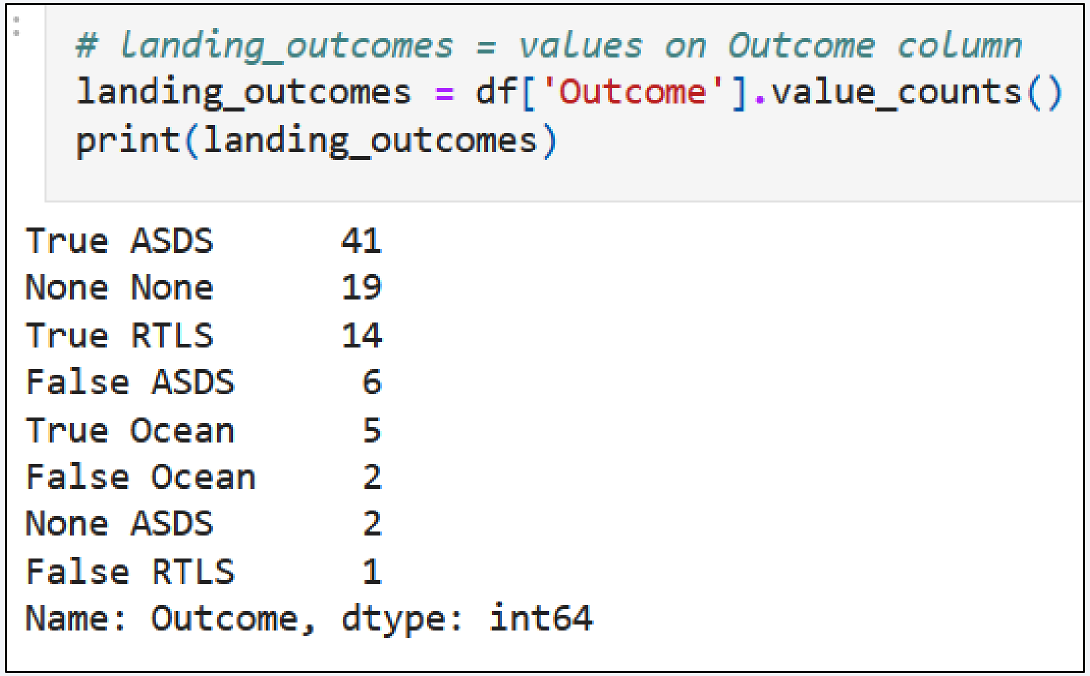
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
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
```



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

Fig 3: Some Data Wrangling results

EDA with Data Visualization

- Using Matplotlib and Seaborn, we created various charts and plots to visualise the relationship between various attributes
- We used scatter plots to map the relationship between Flight Number and Payload Mass, Flight Number and Launch Site, Payload Mass and Launch Site, Flight Number and Orbit type and Payload Mass and Orbit type
- We use a bar plot to visualize the relationship between Success rate and Orbit type, according to Payload Mass
- We use a line chart of average success rate vs year to get the launch success yearly trend
- We also do feature engineering by selecting important features for modelling, using get_dummies method for one-hot encoding and type converting all values to float64
- External reference: GitHub link to EDA notebook

EDA with SQL

SQL queries were performed to obtain data on:

- Unique launch sites
- Launch sites that start with 'CCA'
- Total payload mass
- Average payload mass for booster version F9 v1.1
- Date of first successful landing outcome in ground pad
- Boosters successful in drone ship with payload mass from 4000 to 6000kg
- Total number of successful and failed missions
- Booster versions that have carried the maximum payload mass
- Months, failure outcomes, booster versions and launch sites for 2015
- Count of successful landing outcomes between 04-06-2010 and 20-03-2017 in descending order.

External reference: GitHub link to EDA with SQL notebook

Build an Interactive Map with Folium

- We used the Folium library to analyze the impact of the launch sites' features such as location and proximity, with the help of an interactive map
- In a Folium map we created markers as circle icons with labels to pinpoint locations such as launch sites and cities, and lines to connect them and find the distance between them.
- We marked:
  1. All launch sites,
  2. The successful and failed launches for each site with green icon for success and red for failure
  3. The distances between a launch site and the coastline and a nearby city
- External reference: GitHub link to Folium viz notebook

Build a Dashboard with Plotly Dash

- We then built an interactive dashboard using Plotly Dash for real-time visual analytics. It consists of a dropdown list and a range slider to interact with a pie chart and a scatter point chart.
- The launch site dropdown list is for selecting one or all launch sites
- The range slider selects a range of payload mass from 0 to 10000kg
- The pie chart shows the successful launches by site. For one site, it shows the success and failure launches
- The scatter plot has x-axis as payload mass and y-axis as launch outcome for the selected site(s). The payload mass input from range slider is used.
- External reference: GitHub link to Plotly app

Predictive Analysis (Classification)

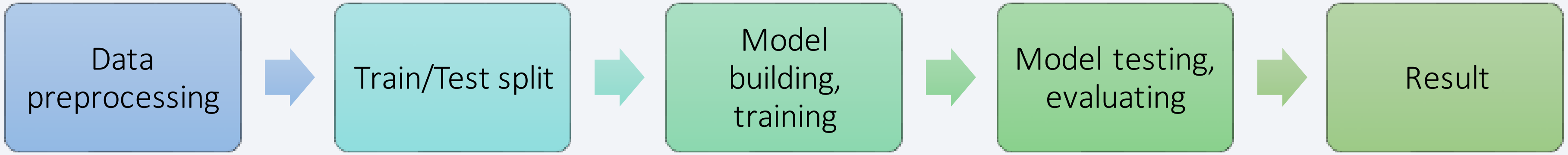


Fig 4: Flow chart for classification

- We then used the Scikit-learn library to build and evaluate classification models. Many models were created and compared.
- The target variable was stored in a NumPy array Y and the data was standardized using StandardScaler() before being split into training and testing sets
- Logistic Regression, Support Vector Machine (SVM), Decision tree and K nearest neighbors (KNN) were the models created, then evaluated using the accuracy score and the confusion matrix

Predictive Analysis (Classification)

- GridSearchCV was used for choosing the best hyperparameters for modeling
- All of the models performed the same and give the same result, as their confusion matrices and accuracy scores were identical.
- External reference: GitHub link to ML prediction notebook

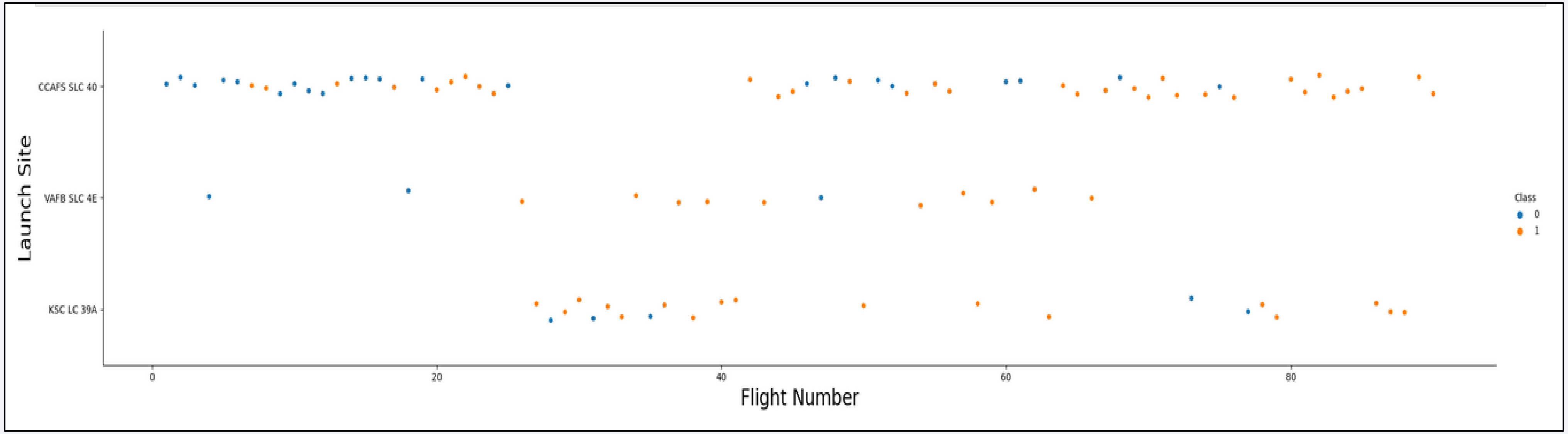
Results

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

Section 2

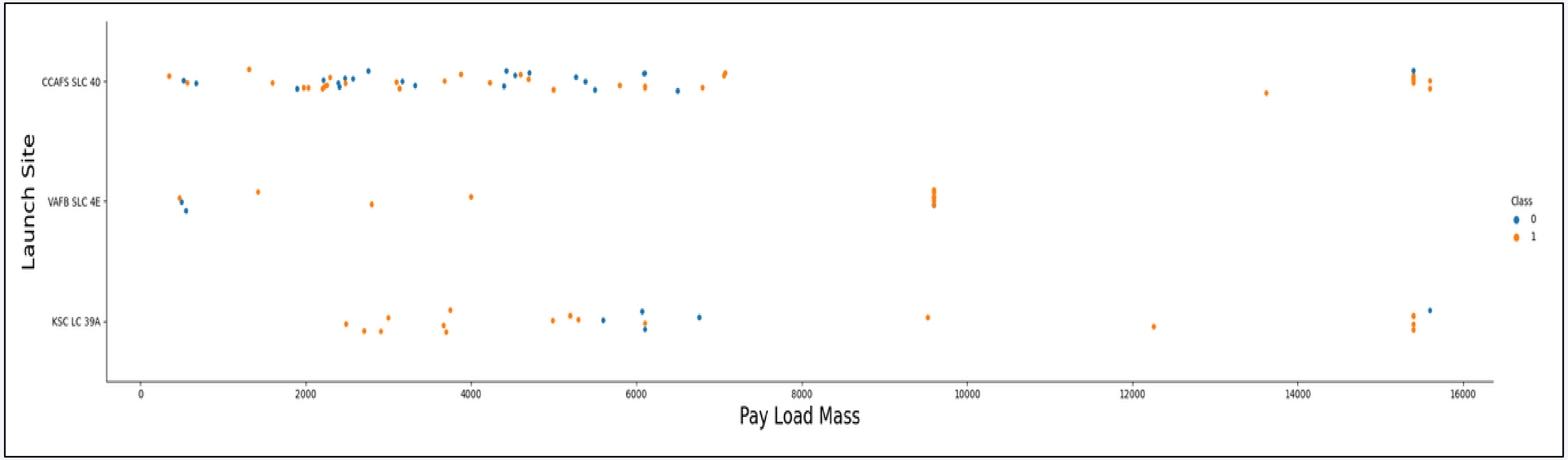
Insights drawn from EDA

Flight Number vs. Launch Site



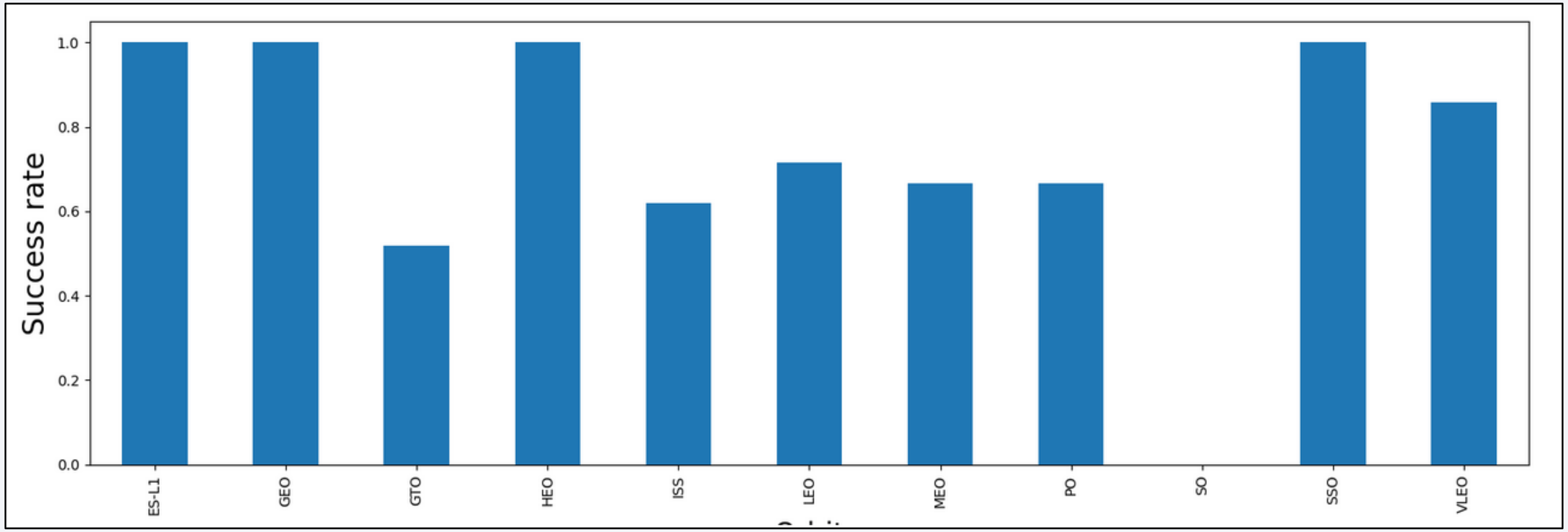
From the scatter plot we see that for an increase in flight number, the successful launches increase for all 3 launch sites. However, some launch sites simply have more number of launches (data points) than others.

Payload vs. Launch Site



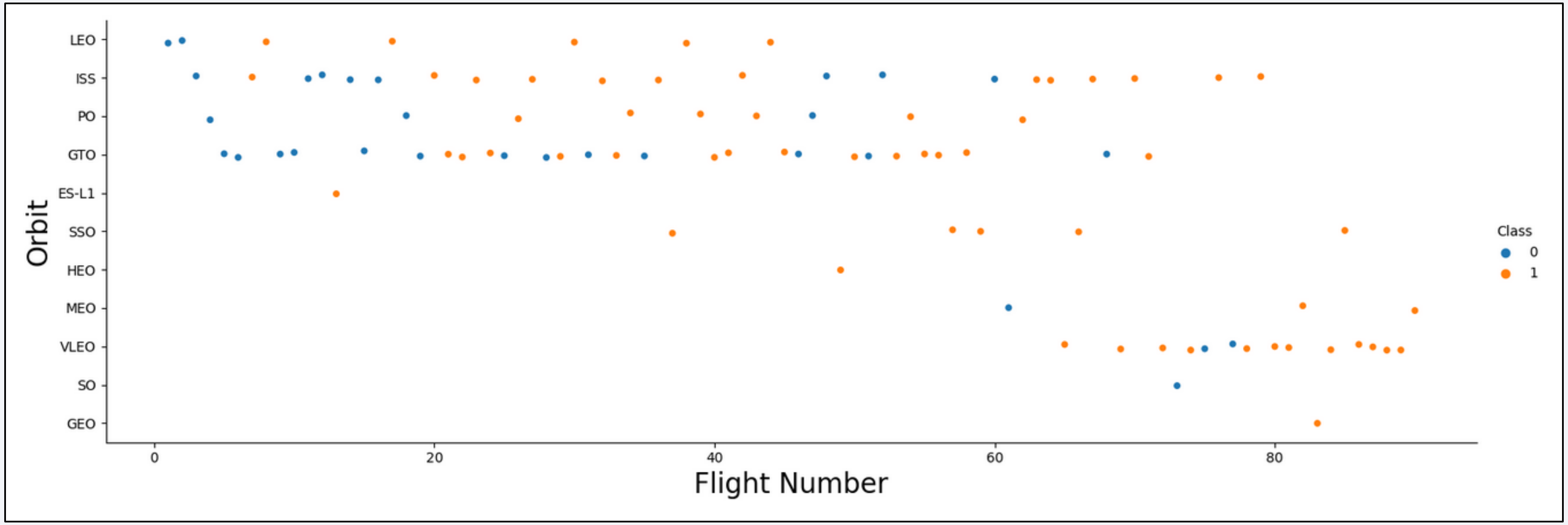
From the scatter plot we see that for an increase in payload mass, the successful launches increase for all 3 launch sites. However, the VAB SLC 4E launch site has no payload heavier than 10000kg whereas KSC LC 39A seems to perform well for both lighter (2000-6000kg) and heavier (10000kg+) payloads.

Success Rate vs. Orbit Type



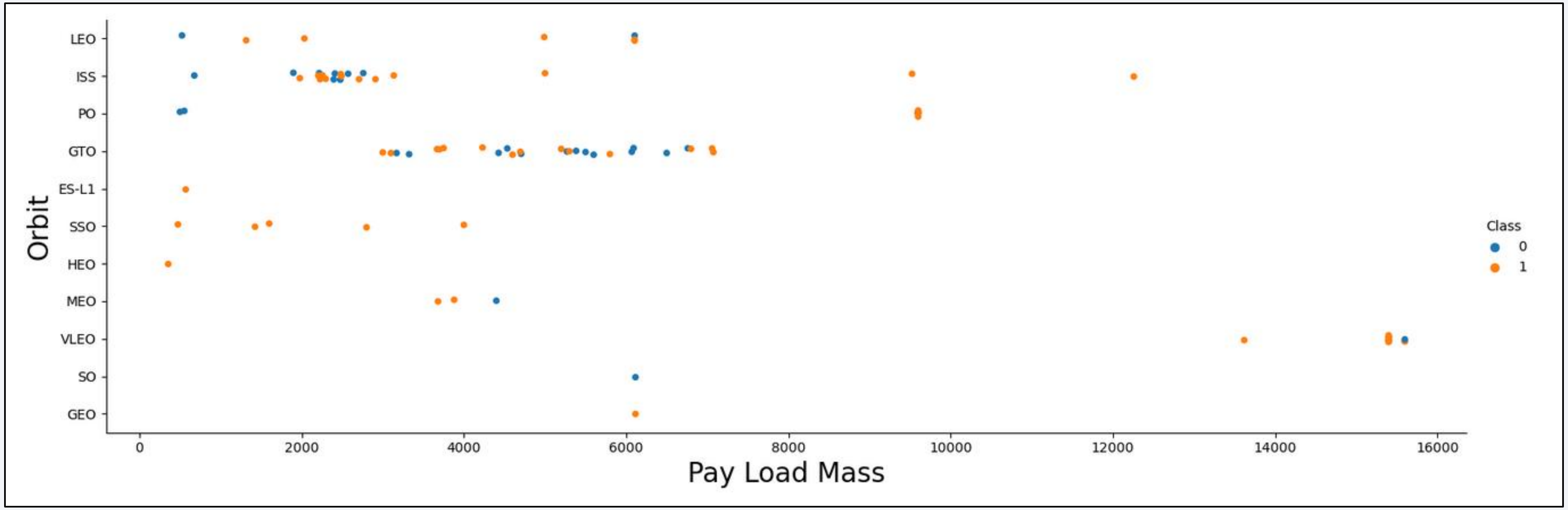
From the bar chart, we see that the orbits ESL-1, GEO, HEO and SSO have the highest success rate of 1 or 100%, while GTO and ISS the lowest at around 60%. However, SO's success rate could not be determined.

Flight Number vs. Orbit Type



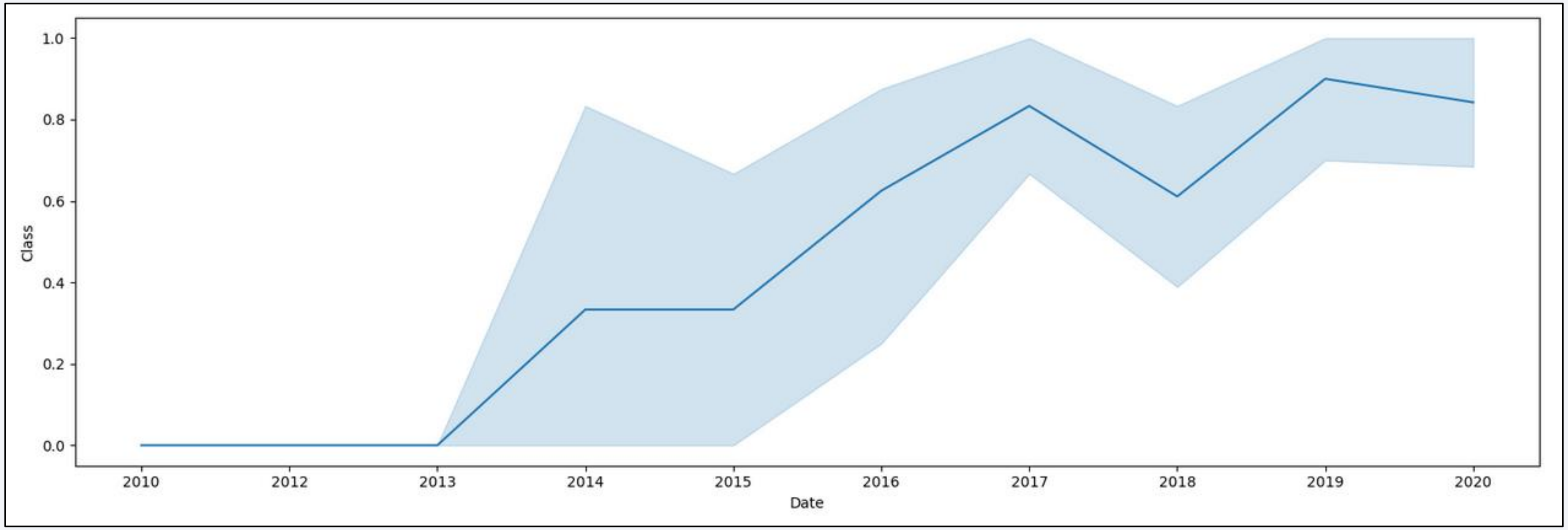
From the scatter plot, we see that overall mid- to high values of flight number give successful launches for all orbits, however, from SSO onwards there is no data for lower flight numbers. LEO shows this trend, yet GTO has failed launches even for higher flight numbers.

Payload vs. Orbit Type



From the scatter plot, we see that higher values of payload mass give successful launches for orbits LEO, Polar and ISS, however, SSO has no data for higher payload mass. GTO has both successful and failed launches for low and high payload mass.

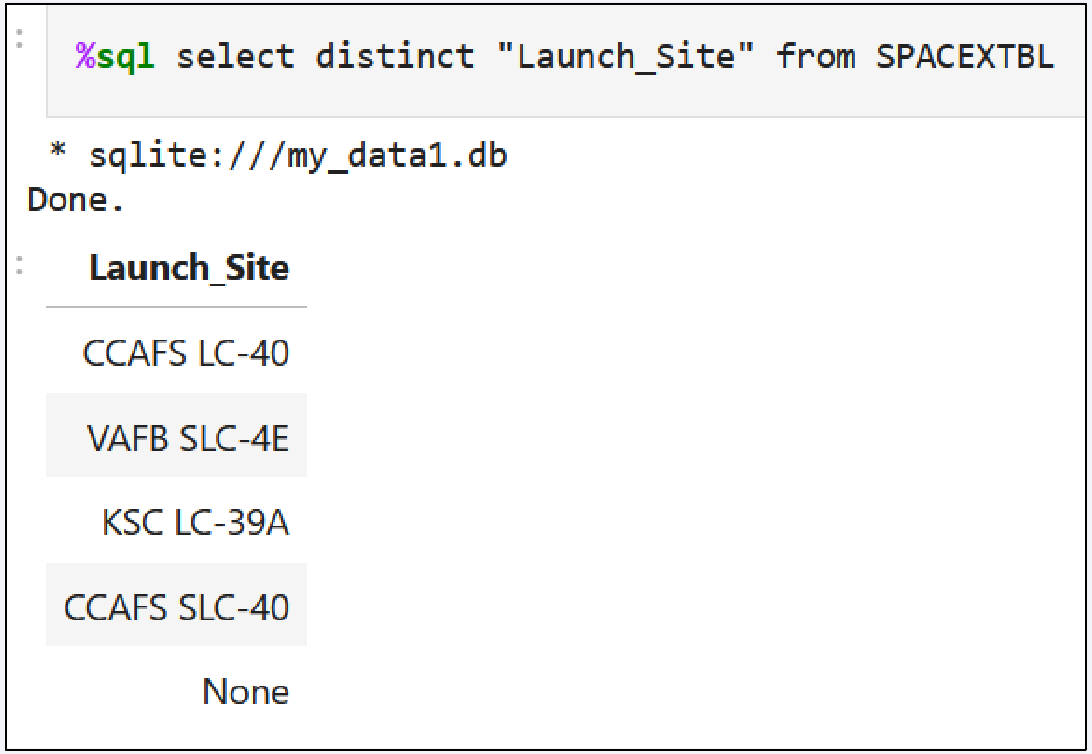
Launch Success Yearly Trend



The line chart shows that the yearly average launch success rate has steadily increased from 2013 onwards.

All Launch Site Names

SQL query to get all launch site names from "Launch_Site" column of spacextbl table using 'distinct'

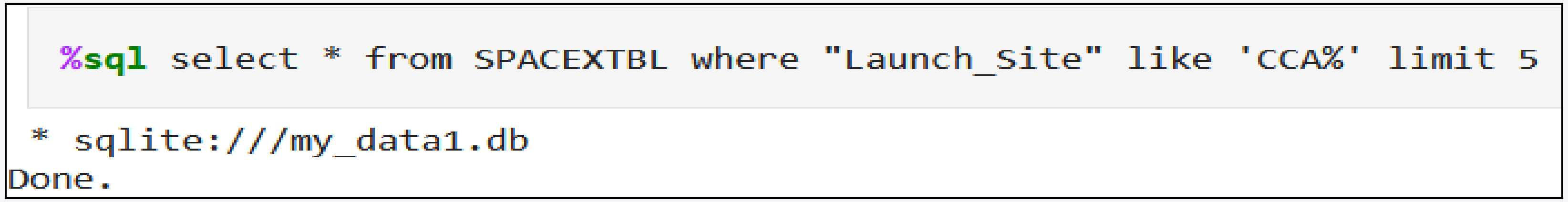


```
%sql select distinct "Launch_Site" from SPACEXTBL
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

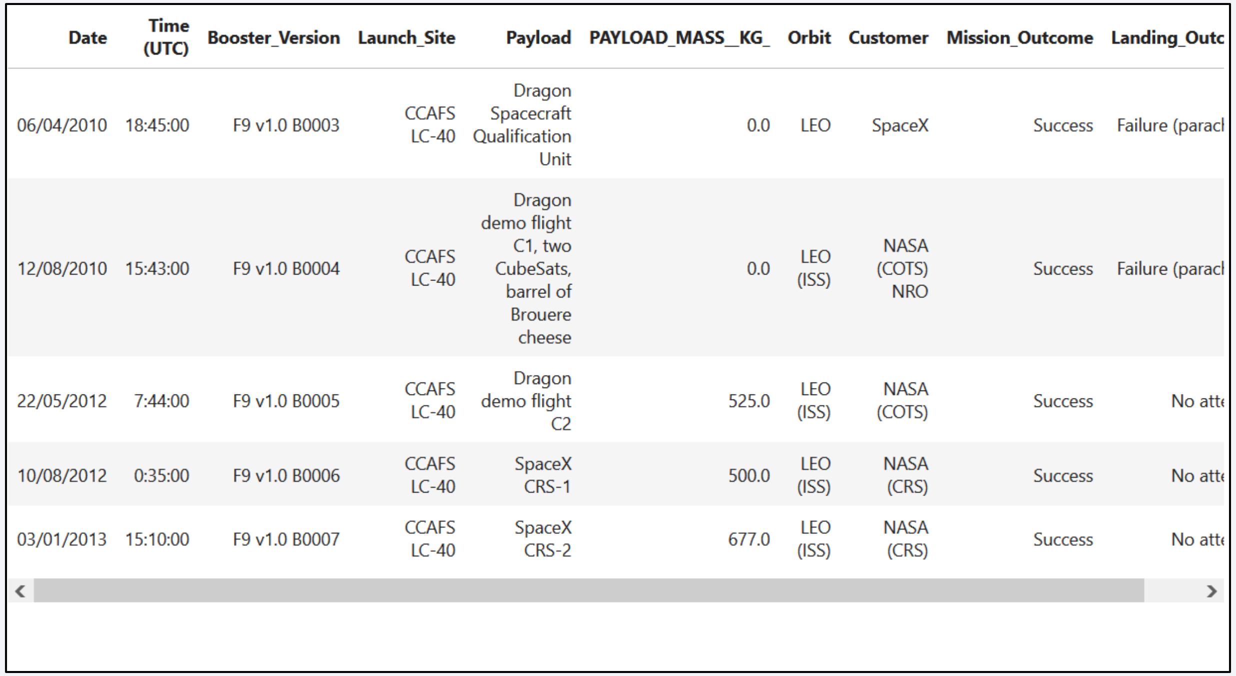
Launch Site Names Begin with 'CCA'



```
%sql select * from SPACEXTBL where "Launch_Site" like 'CCA%' limit 5
```

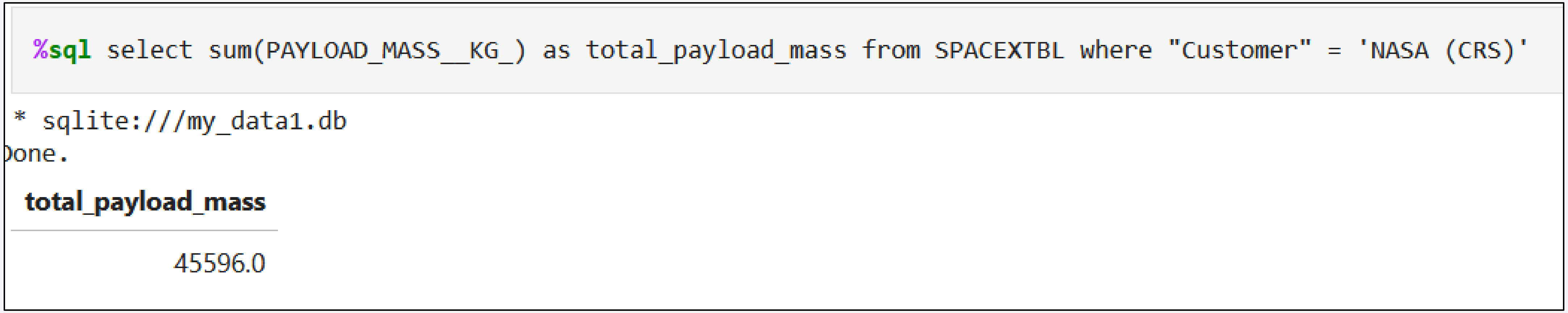
```
 * sqlite:///my_data1.db
Done.
```

SQL query for 5 launch sites starting with CCA using 'limit' and wildcard character



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
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 (paracl
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 atte
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No atte
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass



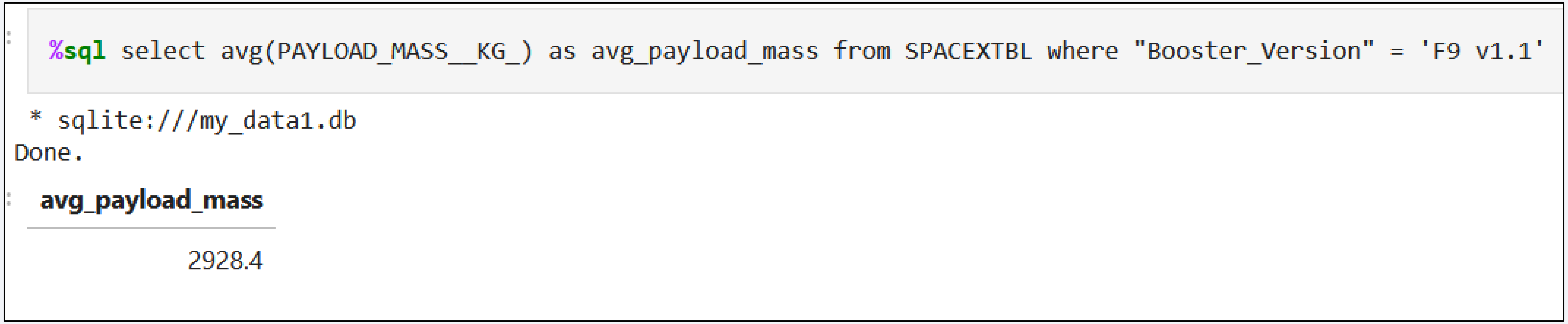
```
%sql select sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTBL where "Customer" = 'NASA (CRS)'
```

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

total_payload_mass
45596.0

SQL query for total payload mass using sum() function and 'where' clause

Average Payload Mass by F9 v1.1



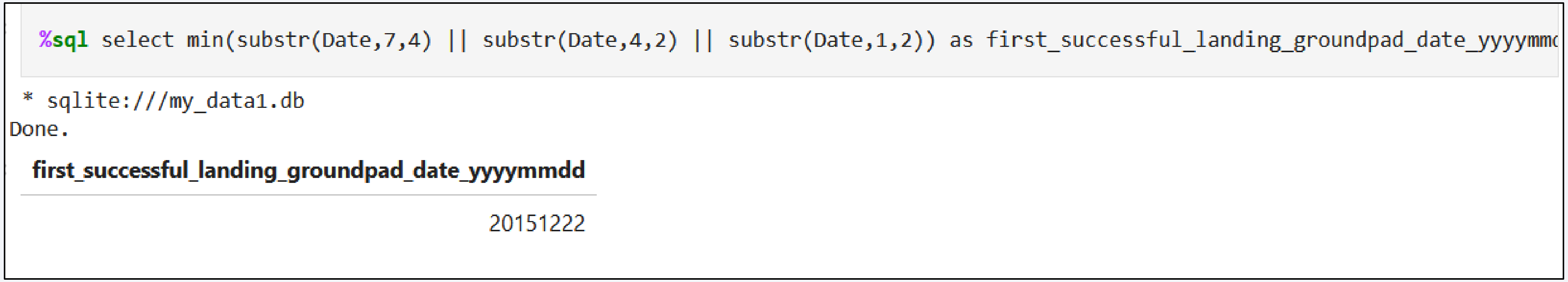
```
%sql select avg(PAYLOAD_MASS__KG_) as avg_payload_mass from SPACEXTBL where "Booster_Version" = 'F9 v1.1'
```

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

avg_payload_mass
2928.4

SQL query for average payload mass for F9 v1.1 using avg() function and where clause

First Successful Ground Landing Date



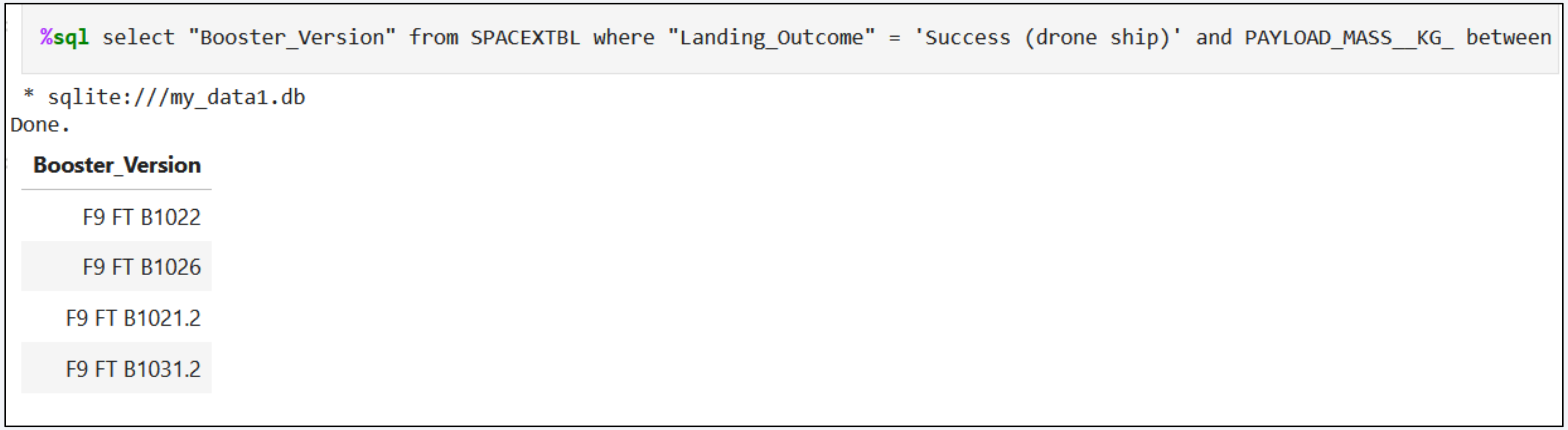
```
%sql select min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)) as first_successful_landing_groundpad_date_yyyymmc
```

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

first_successful_landing_groundpad_date_yyyymmdd
20151222

SQL query for first successful ground landing date using substr() method and concatenation, then min() method for earliest date

Successful Drone Ship Landing with Payload between 4000 and 6000



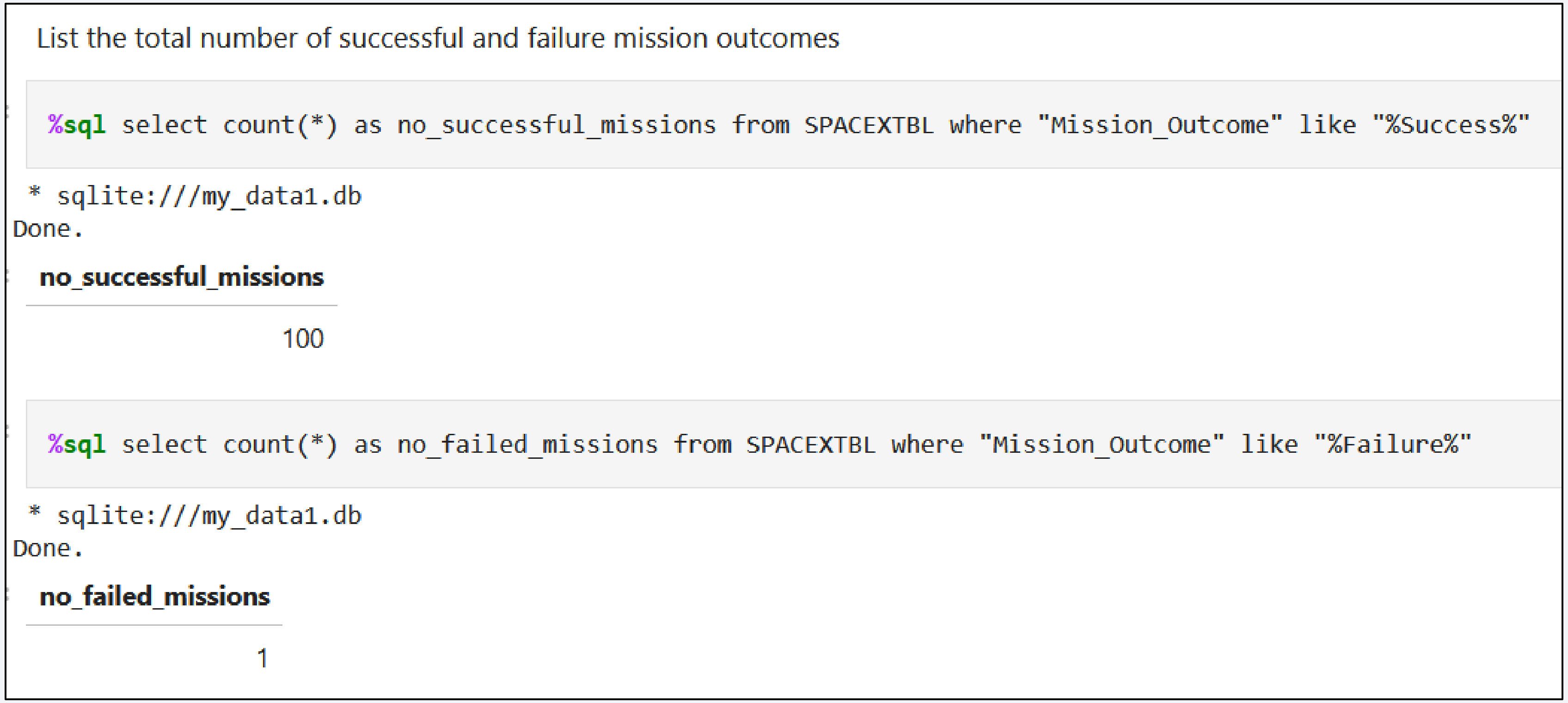
```
%sql select "Booster_Version" from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between
```

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

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

SQL query for booster versions with landing outcome successful(drone ship) and payload mass in given range using 'between' and 'where' clause.

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
%sql select count(*) as no_successful_missions from SPACEXTBL where "Mission_Outcome" like "%Success%"
```

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

no_successful_missions
100

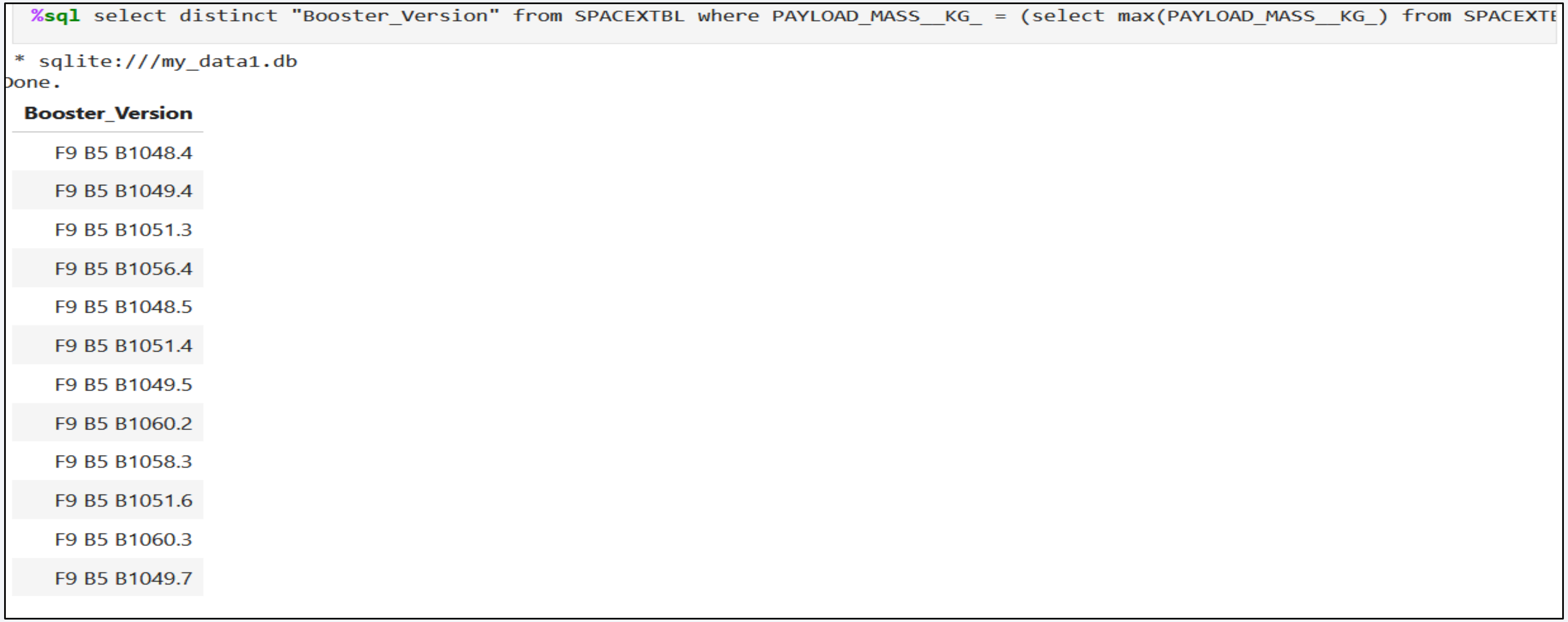
```
%sql select count(*) as no_failed_missions from SPACEXTBL where "Mission_Outcome" like "%Failure%"
```

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

no_failed_missions
1

SQL query for total of successful and failed missions using count() method and wildcard character

Boosters Carried Maximum Payload



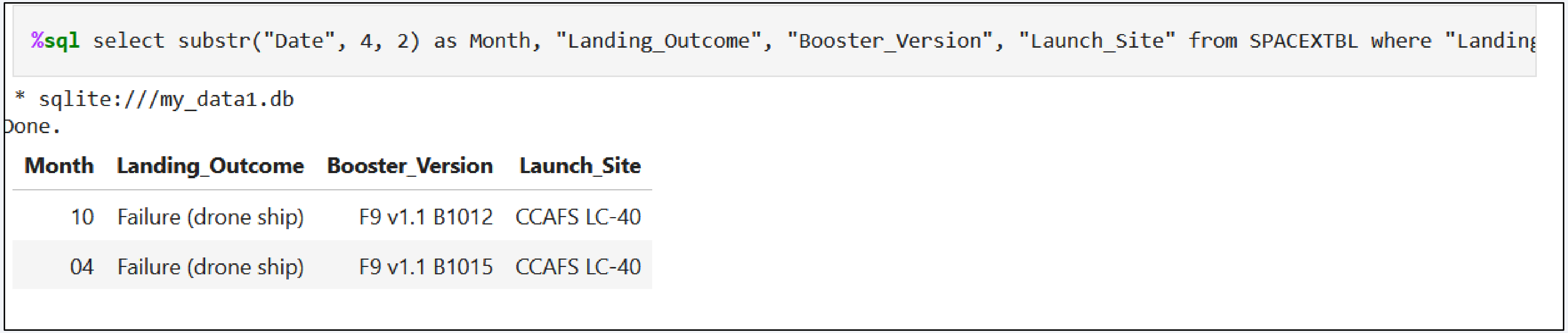
```
%sql select distinct "Booster_Version" from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTE
```

```
 * 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

SQL query for booster versions with maximum payload with subquery. The subquery returns the maximum payload mass using max() method, which is used in the 'where' clause of the outer query to match the booster versions.

2015 Launch Records



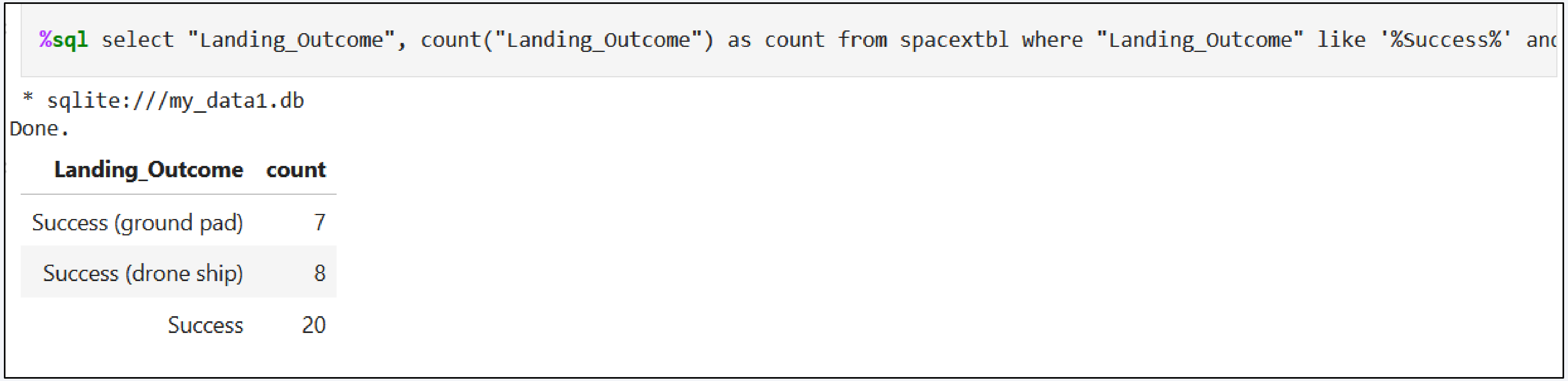
```
%sql select substr("Date", 4, 2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTBL where "Landing
```

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

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

SQL query to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 with 'where' clause and substr() method to find year, month

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



```
%sql select "Landing_Outcome", count("Landing_Outcome") as count from spacextbl where "Landing_Outcome" like '%Success%' and
```

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

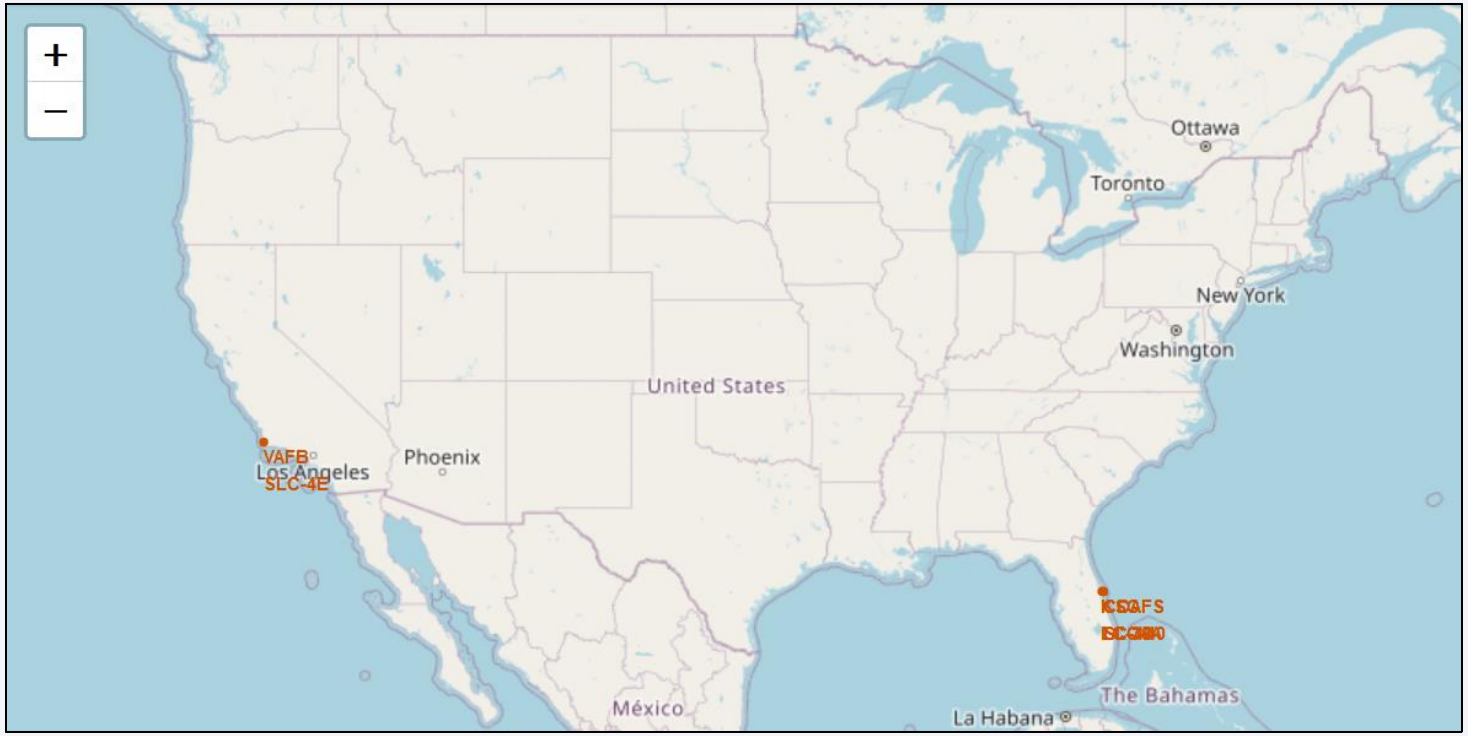
Landing_Outcome	count
Success (ground pad)	7
Success (drone ship)	8
Success	20

SQL query to rank the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order using count() method, 'where' clause, 'order by' and 'group by'

Section 3

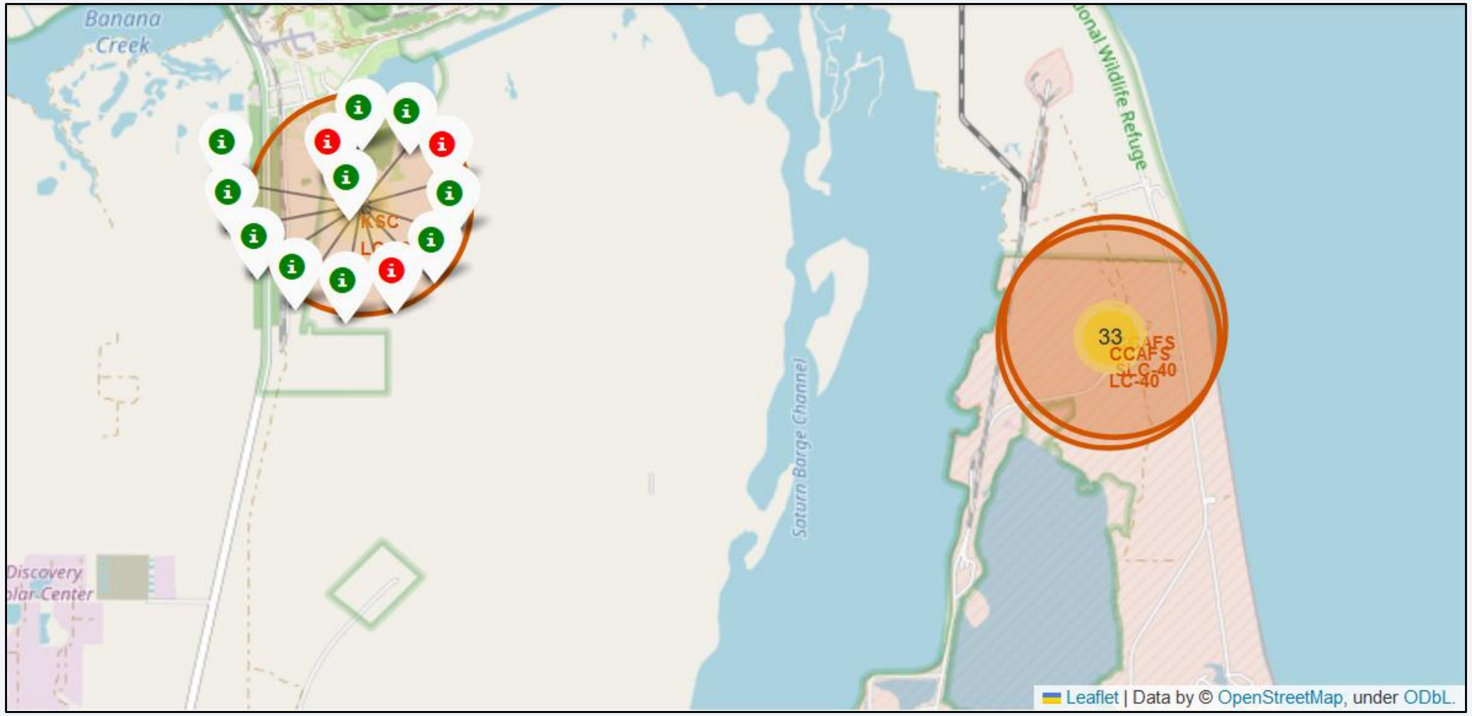
Launch Sites Proximities Analysis

Launch site locations on geographical map



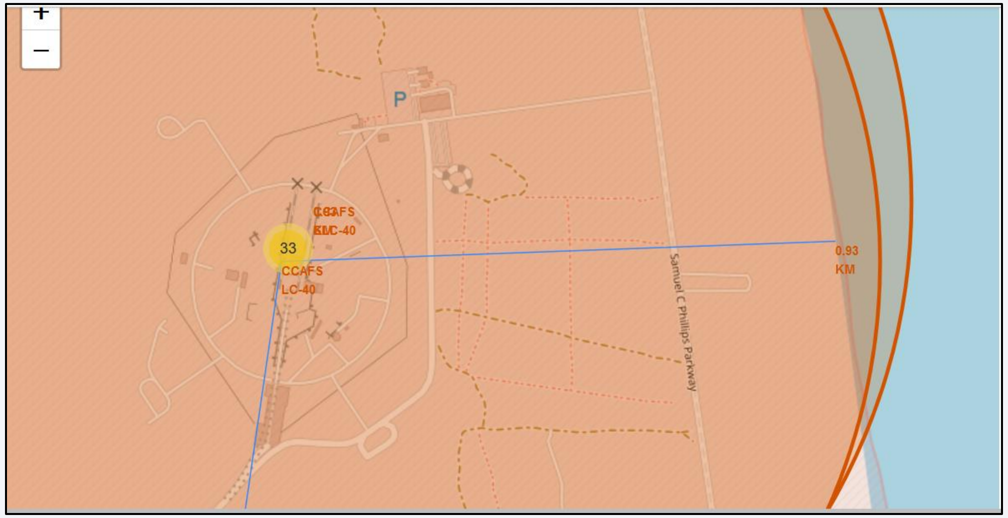
Folium map showing the locations of all launch sites. It is found that they are all close to the Equator, as well as near coastlines.

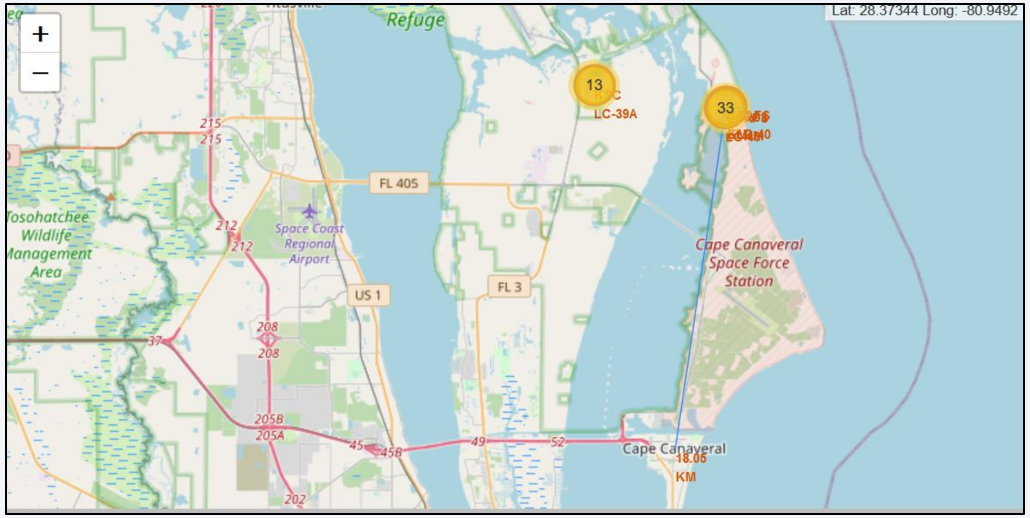
Color coded Launch outcomes for site KLC LC-39A



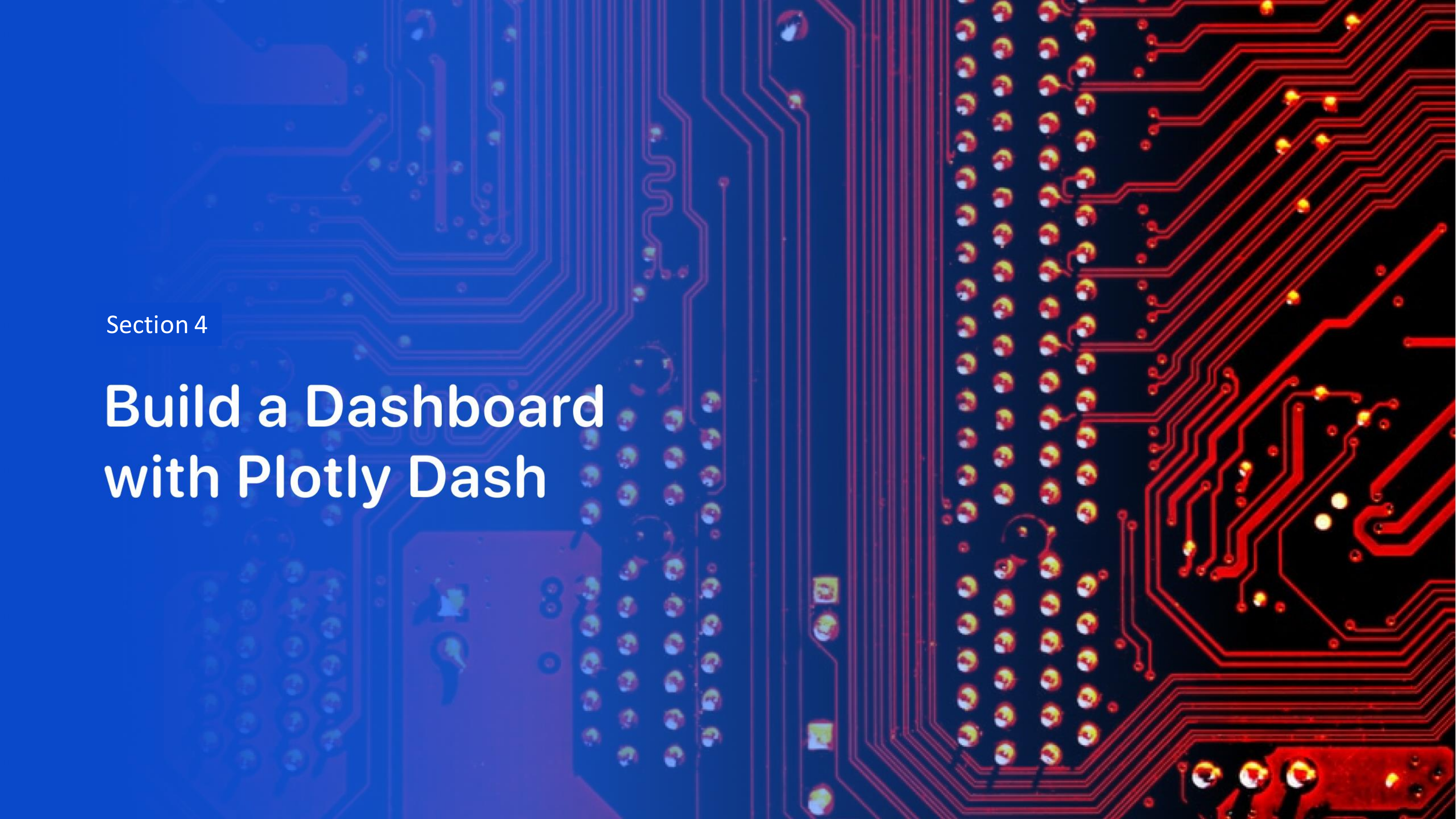
Color coded icons (red for failure, green for success) corresponding to each launch from a selected launch site, here KLC LC-39A has 3 failures and 10 successful launches. This was obtained by zooming in on a cluster such as the one on the right, showing 30 launches in total

Distances from a launch site to its proximities





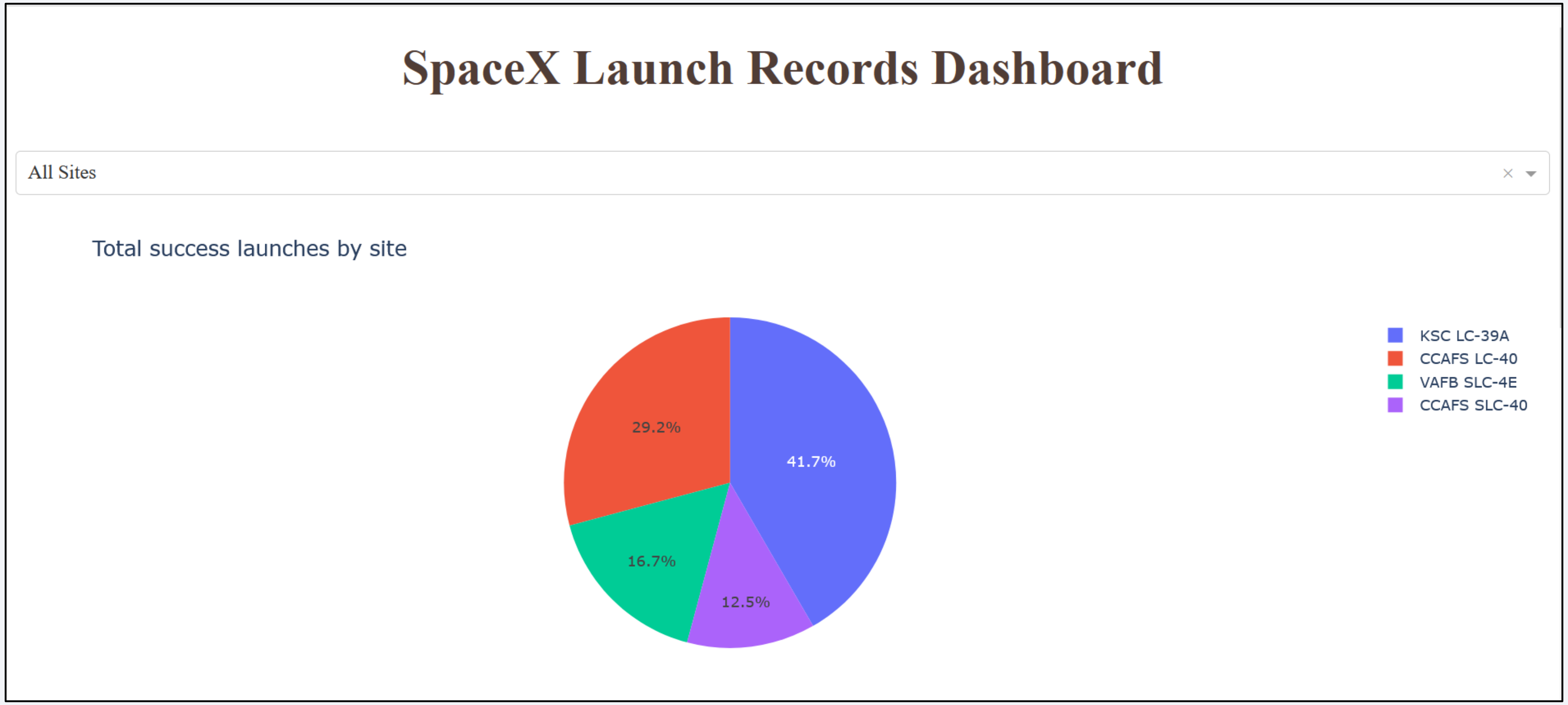
Lines depicting calculated distance from the selected launch site CCAFS LC-40 to its proximities- one a point on the nearest coastline (0.93 km), another a nearby city Cape Canaveral (18.05 km) This shows that launch sites are in close proximity to places such as railways, highways, cities and coastlines



Section 4

Build a Dashboard with Plotly Dash

Pie chart of Launch Success count for All sites



SpaceX Launch Records Dashboard

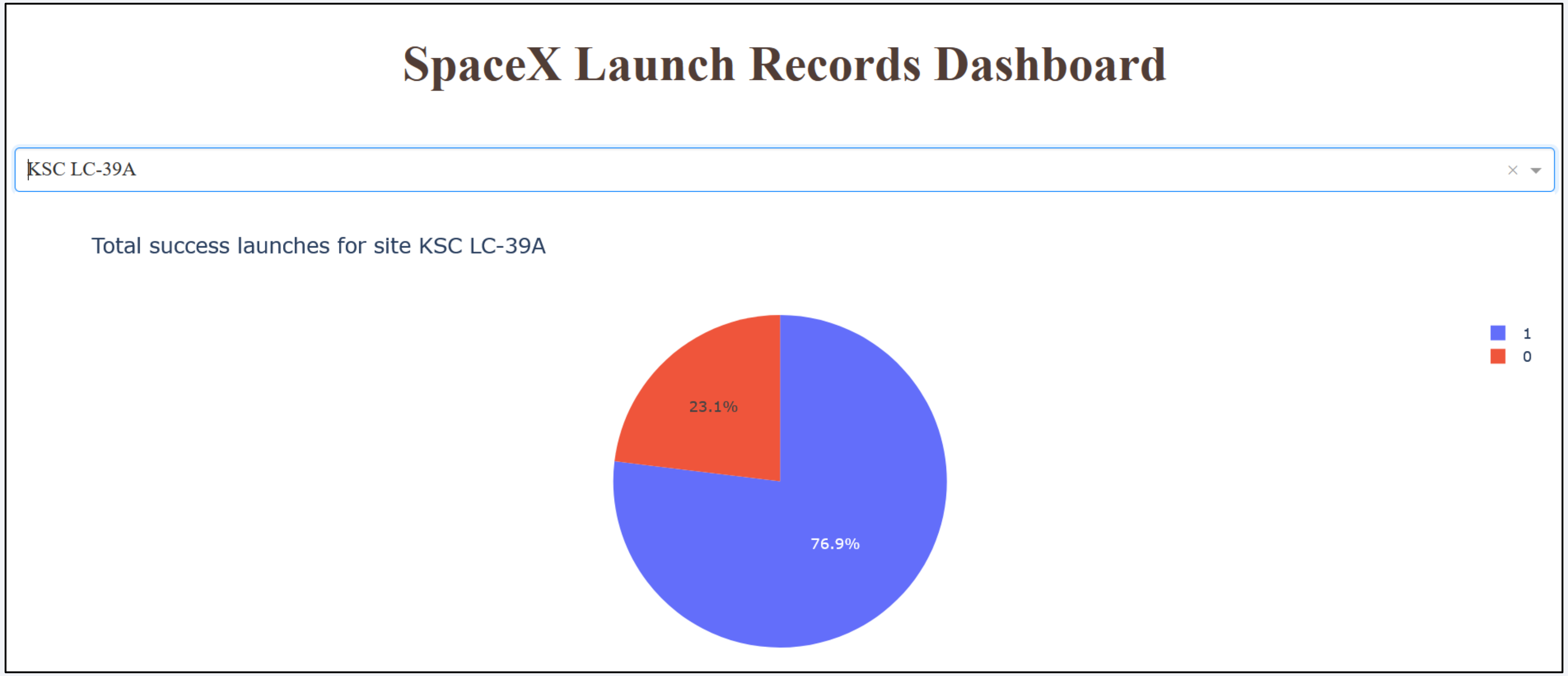
All Sites

Total success launches by site

Pie chart showing successful launches of all launch sites, i.e. total count of class column for each launch site, as fraction of total set of successful launches.

We see that KSC LC-39 A has the highest percentage share of successful launches at 41.7%, while CCAFS SLC-40 has the lowest.

Pie chart of launch site with highest launch success ratio



SpaceX Launch Records Dashboard

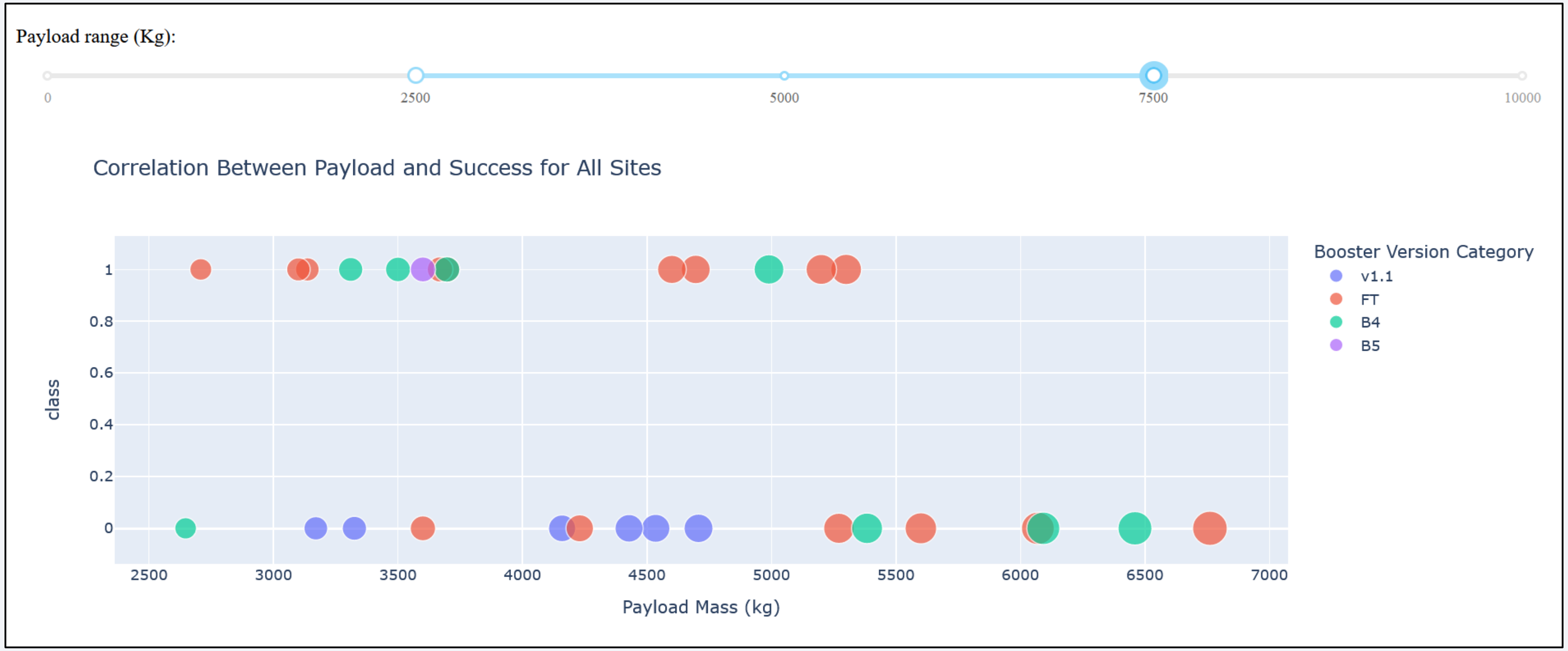
KSC LC-39A

Total success launches for site KSC LC-39A

Pie chart of success to failure ratio for KSC LC – 39A, which is the launch site with the highest launch success ratio among all the sites.

Here we see that of all the past launches at this site, 76.9% have been successful

Scatter plot of Payload mass vs Launch outcome for all sites



Payload range (Kg):

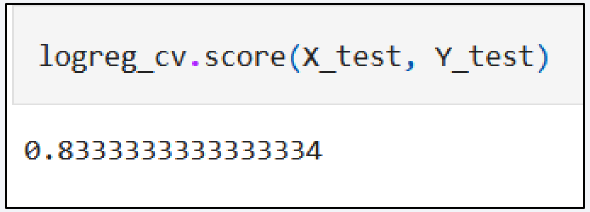
- Scatter plot of Payload vs. Launch Outcome scatter plot for all sites within a chosen payload range 2500-7500 kg as selected in the range slider, color coded by booster version

- On examining different payload ranges, it was found that lighter payloads (<5000kg) have greater success rates than heavier ones for all boosters. For heavy payloads, however, data was there on only FT and B4 versions. The FT booster has the highest success rate among all versions.

Section 5

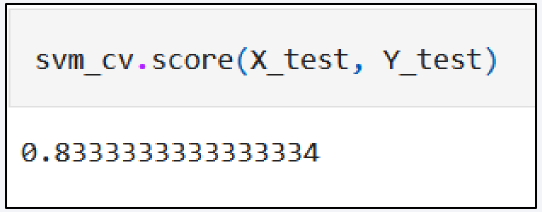
Predictive Analysis (Classification)

Classification Accuracy



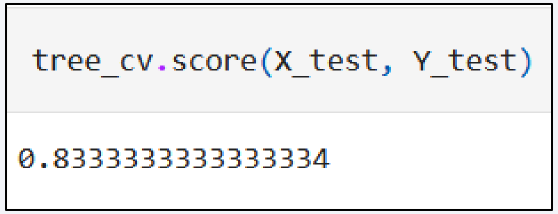
```
logreg_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```



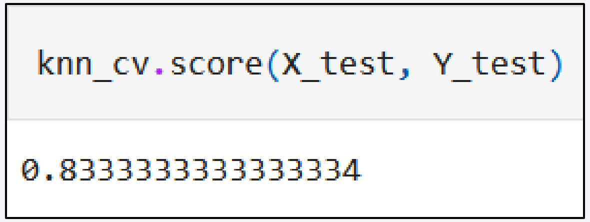
```
svm_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```



```
tree_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

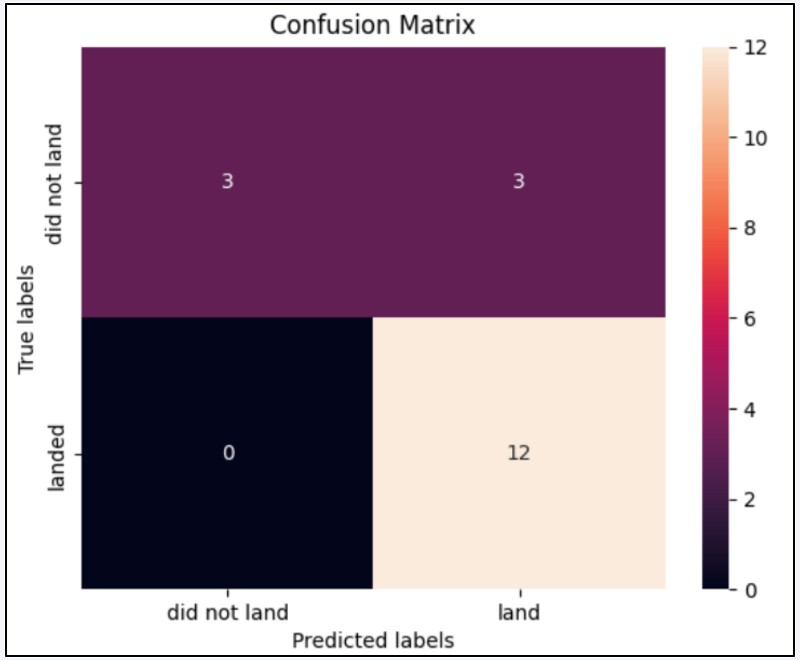


```
knn_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

- The model accuracy for all built classification models is the same, i.e., 0.833333333334
- All models have the same confusion matrix
- Hence, there is no single best performing model and we can choose any one of the four for our modeling and prediction purposes. We go with Decision trees as it has a slightly higher accuracy 0.887 for training data as compared to the other models (~0.84)

Confusion Matrix



Confusion matrix of the decision tree classification model. The columns are for model predictions and rows are for actual values. The intersections of matching labels are for True Positives and True Negatives, while mismatching labels give False Positives(Model says landed but actually didn't) and False Negatives (Model says didn't land but actually did). The quantities of each category are given inside each of the boxes and colored according to the colorbar.

Conclusions

- We were able to build a classification model of 83.34% accuracy in predicting whether or not the first stage of a Falcon 9 rocket would land successfully.
- The yearly average launch success rate has been increasing since 2013
- Several factors influence the outcome of a launch, with varying levels of impact.
- Certain orbits such as ESL-1, GEO, HEO and SSO, higher flight numbers, launching from the KSC LC 39A launch site and lower payload mass are associated with higher success rates
- However, higher values of payload mass give successful launches for orbits LEO, Polar and ISS

Thank you!