IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

S M Rafizul Haque

April 21, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X promotes Falcon 9 rocket launches on its website at a price of $62 million, while other providers charge over $165 million each. A significant part of the cost savings comes from Space X's ability to reuse the first stage. Thus, if we can predict whether the first stage will land, we can estimate the launch cost. This knowledge could be valuable for other companies bidding against Space X for a rocket launch contract. The aim of this project is to develop a machine learning pipeline to forecast the successful landing of the first stage.

- Problems we want to find answers

  - What factors determine whether the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions need to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

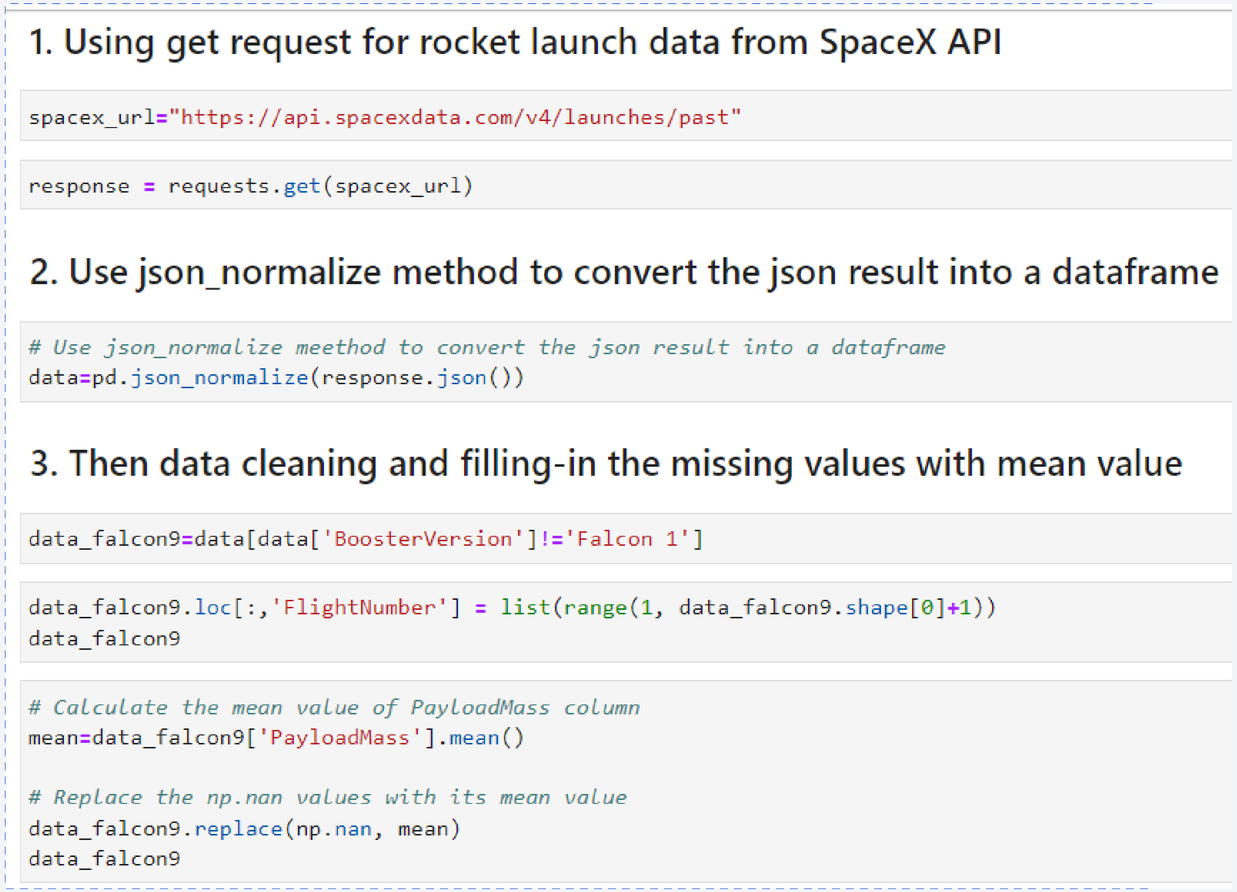
Data Collection

- Data was collected using get request to the SpaceX API.
- Next, the response content was decoded as a JSON using .json() method and was turned into a Pandas data frame using .json_normalize() method.
- Then the data was cleaned, checked for missing values and the missing values were filled-in where necessary.
- Data about Falcon 9 launch records were collected through web scraping from Wikipedia using BeautifulSoup Python library.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a Pandas data frame for future analysis.

Data Collection – SpaceX API

- Get request was sent to SpaceX API to collect data, cleaned the requested data and applied some basic data wrangling and formatting.
- The link to the notebook is:

https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



1. Using get request for rocket launch data from SpaceX API

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

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

2. Use json_normalize method to convert the json result into a dataframe

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

3. Then data cleaning and filling-in the missing values with mean value

```
data_falcon9=data[data['BoosterVersion']!='Falcon 1']
```

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

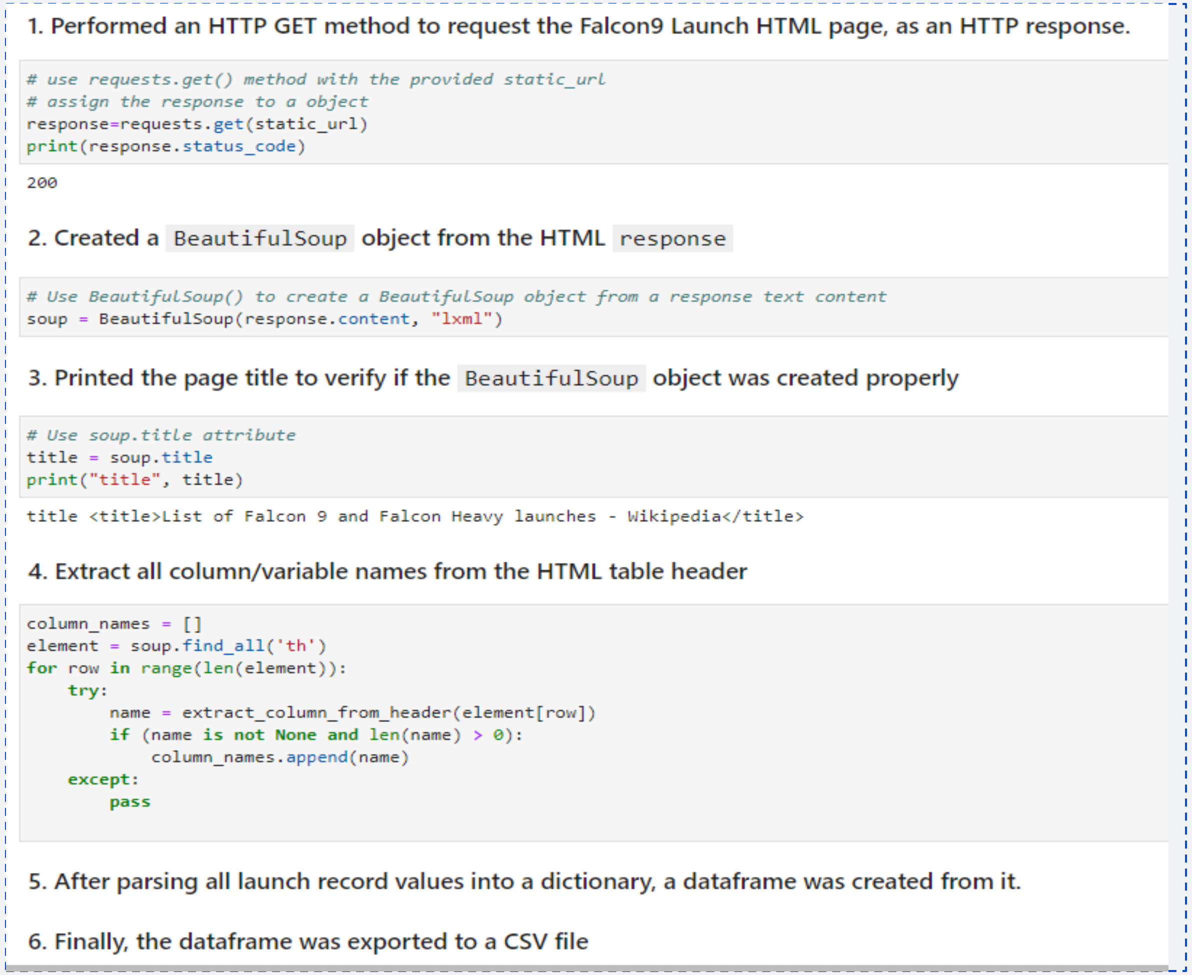
```
# Calculate the mean value of PayloadMass column
mean=data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9.replace(np.nan, mean)
data_falcon9
```

Data Collection - Scraping

- Web scrapping was used to extract Falcon 9 launch records using BeautifulSoup library.
- From the extracted html data, tables were parsed and converted into a Pandas dataframe.
- The link to the notebook is:

https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/0bddd967f8f72e018749ee4fd923bcc612e0f725/jupyter-labs-webscraping.ipynb



1. Performed an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

```
200
```

2. Created a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, "lxml")
```

3. Printed the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
title = soup.title
print("title", title)
```

```
title <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

4. Extract all column/variable names from the HTML table header

```
column_names = []
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)
    except:
        pass
```

5. After parsing all launch record values into a dictionary, a dataframe was created from it.

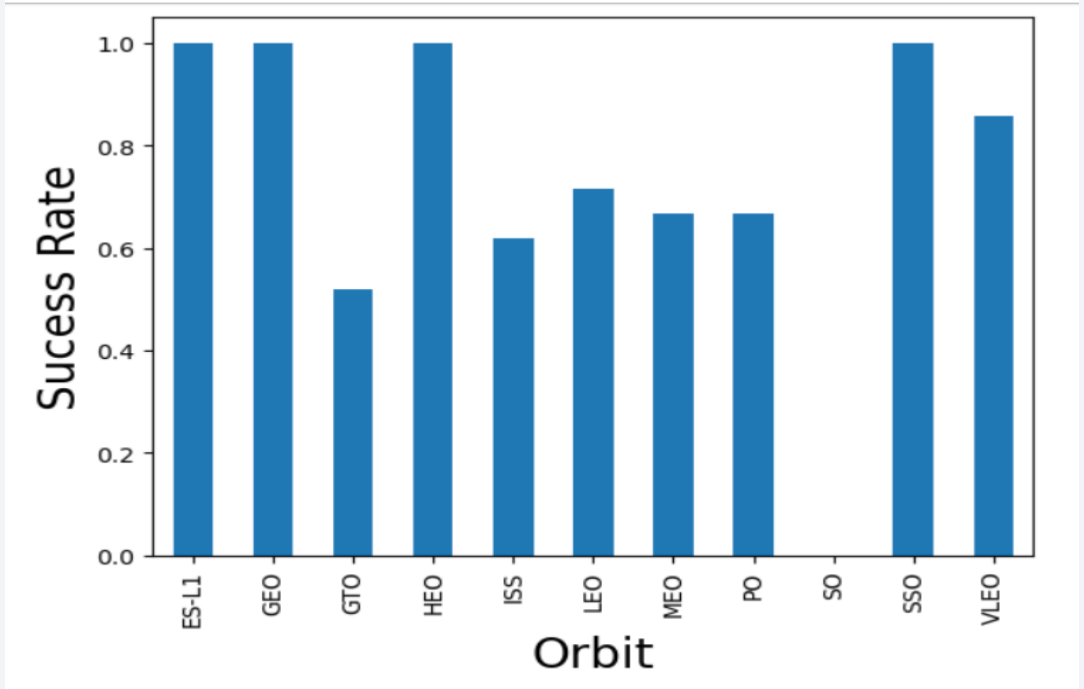
6. Finally, the dataframe was exported to a CSV file

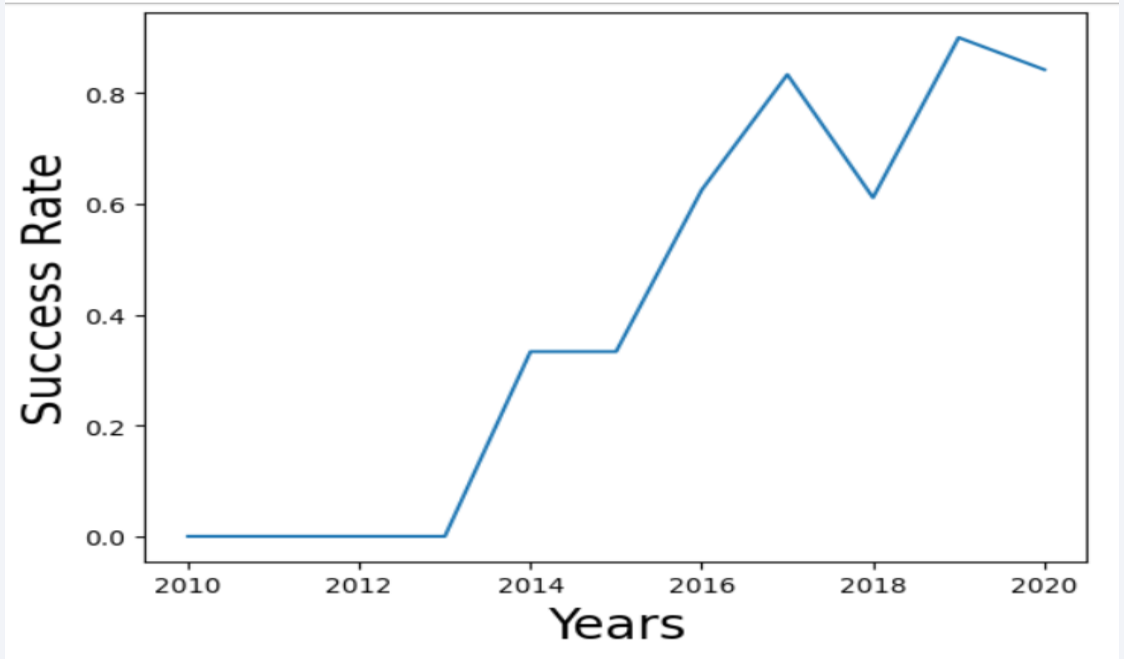
Data Wrangling

- First step of data analysis was to calculate the percentage of the missing values in each attribute
- Data types of the columns, whether numerical and categorical, were identified
- Number of launches on each site was calculated
- Number and occurrence of each orbit were calculated
- Number and occurrence of mission outcome of the orbits were calculated
- Landing outcome label from outcome column was created
- Finally, the dataframe was exported to a csv file.
- The link to the notebook is https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/0bddd967f8f72e018749ee4fd923bcc612e0f725/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, payload and orbit type and the yearly trend of launch success were visualized.





The link to the Notebook is:
https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/0bddd967f8f72e018749ee4fd923bcc612e0f725/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- First, connection with database was established and then the Spacex dataset was converted to a database table.

- SQL commands were executed for EDA to get insights from the data. Results of the following queries were obtained using SQL commands:

  - Names of unique launch sites in the space mission.
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - The date when the first successful landing outcome in ground pad was achieved
  - Total number of successful and failure mission outcomes
  - Names of the booster versions which have carried the maximum payload mass
  - Failed landing outcomes in drone ship, their booster version and launch site names.

- The link to the notebook is https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/0bddd967f8f72e018749ee4fd923bcc612e0f725/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch sites on the map were marked with map objects circle and marker.
- Success or failure of launches were marked for each site on the map. A new column in the dataframe was created to store the marker color based on the success (green) or failure (red) of the launch.
- Using the color-labeled marker clusters, launch sites with relatively high success rate could be identified.
- Distances between a launch site to its proximities were calculated. For example, distance between the coastline point and the launch site was calculated, and a polyline was also drawn between a launch site to the selected coastline point.
- Plotting distance lines to the proximities helped to know the answers of the following questions:
  - Are launch sites in close proximity to railways?
  - Are launch sites in close proximity to highways?
  - Are launch sites in close proximity to coastline?
  - Do launch sites keep certain distance away from cities?

Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly dash was built
- Pie charts showing the total launches by a certain sites were plotted
- Scatter charts showing the relationship between Outcome and Payload Mass (Kg) for different booster versions were plotted.
- The link to the notebook is https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/0bddd967f8f72e018749ee4fd923bcc612e0f725/spacex_dash_app.py.

Predictive Analysis (Classification)

- The data was fetched and loaded as a Pandas dataframe. A new column was created from the "Class" column of the data and was assigned to "Y", which was the target variable.
- Other features of the data were assigned to variable "X" and these were preprocessed with StandardScaler() transformation method.
- X and Y data were split into training and testing sets.
- Different machine learning models were built and different hyperparameters were tuned using GridSearchCV.
- Accuracy was used as the metric for evaluating the models. Some models were improved using feature engineering and algorithm tuning.
- Finally, models were compared to find the best performing classification model.
- The link to the notebook is https://github.com/rafizulku/IBM-Data-Science-Capstone_SpaceX/blob/0bddd967f8f72e018749ee4fd923bcc612e0f725/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

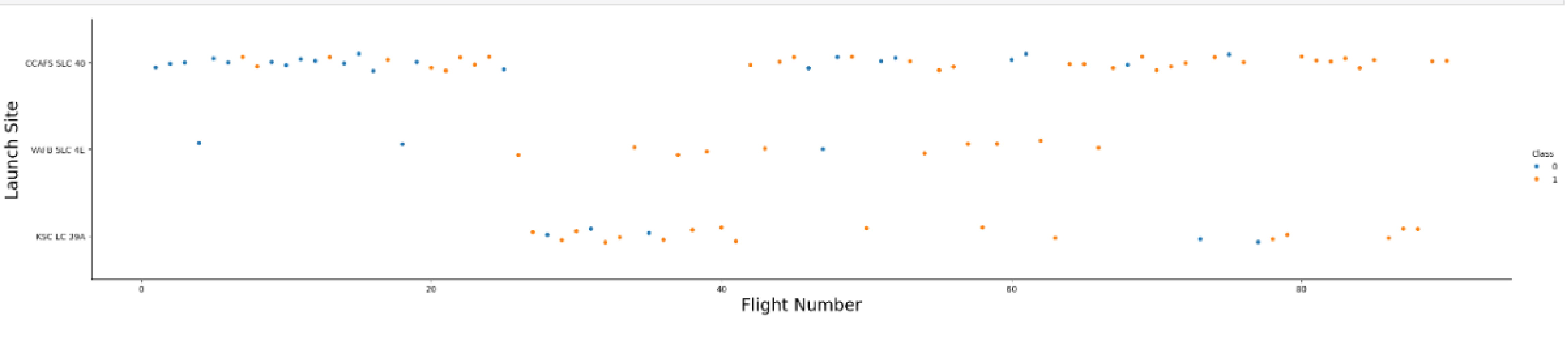
Results

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

Section 2

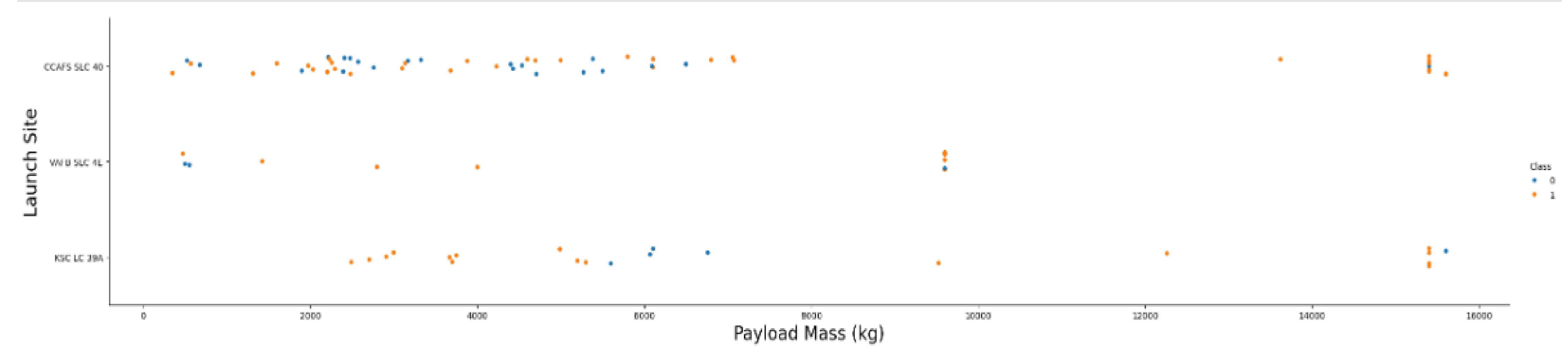
Insights drawn from EDA

Flight Number vs. Launch Site



From this plot, it can be observed that the success rate for a launch site increases with the increasing flight number i.e., the larger the flight number for a launch site, the greater the success rate at a launch site.

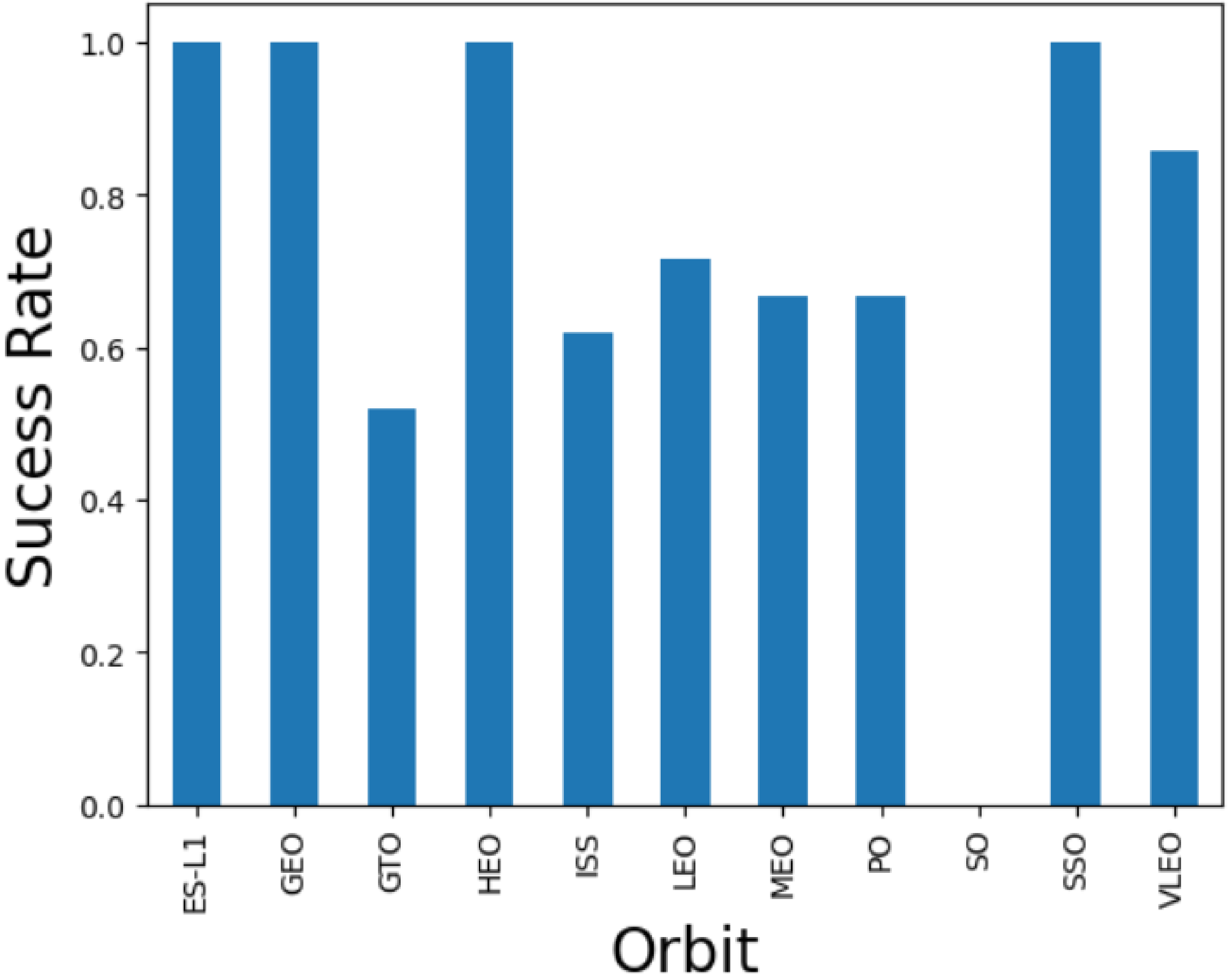
Payload vs. Launch Site



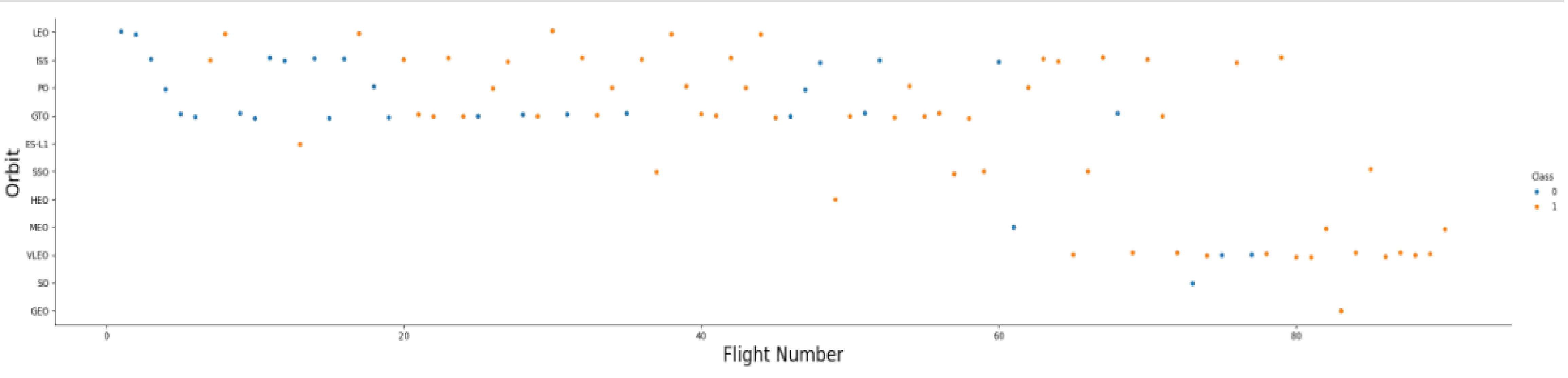
For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000). For two other launch sites, success rates are higher for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type

Success rates are highest and same for orbits ES-L1, GEO, HEO, SSO, while GTO has the lowest success rate and SO has no value for success rate.

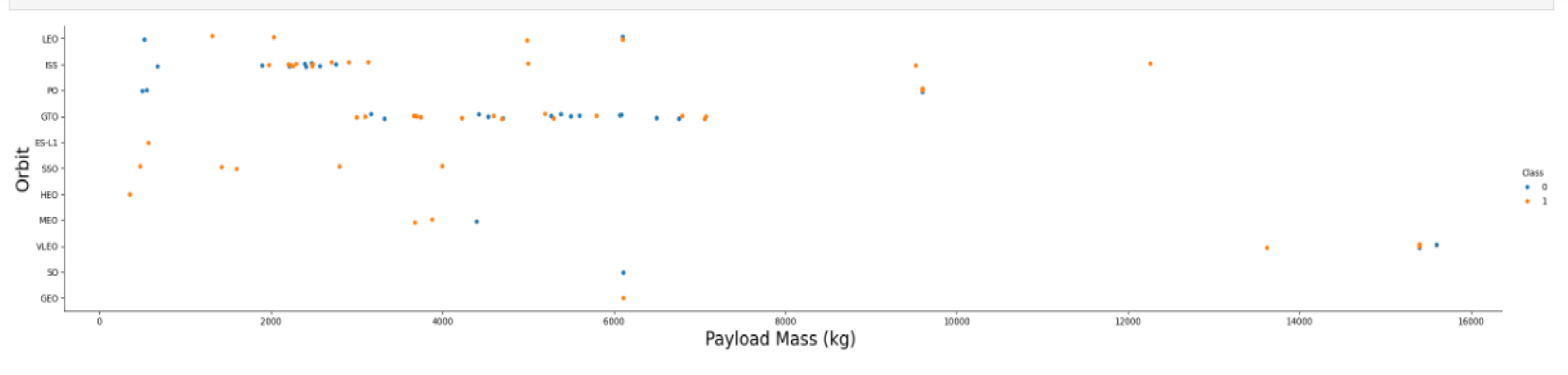


Flight Number vs. Orbit Type



This plot shows that for LEO orbit, success is related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

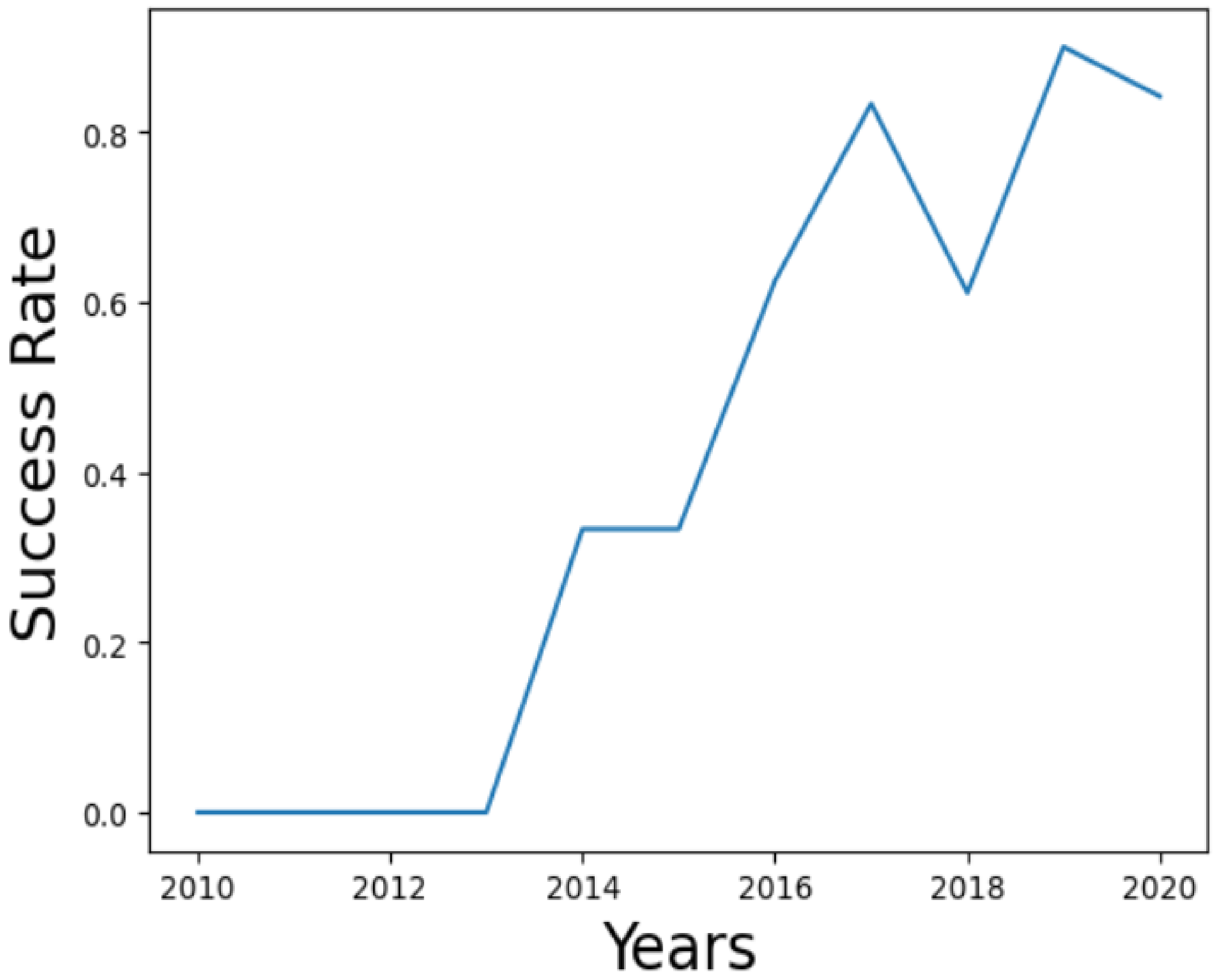
Payload vs. Orbit Type



In Polar, LEO, and ISS orbits with heavy payloads, the likelihood of a successful landing or a positive landing rate is higher. However, distinguishing between positive and negative landing outcomes is challenging for GTO orbits, where both successful and unsuccessful missions occur.

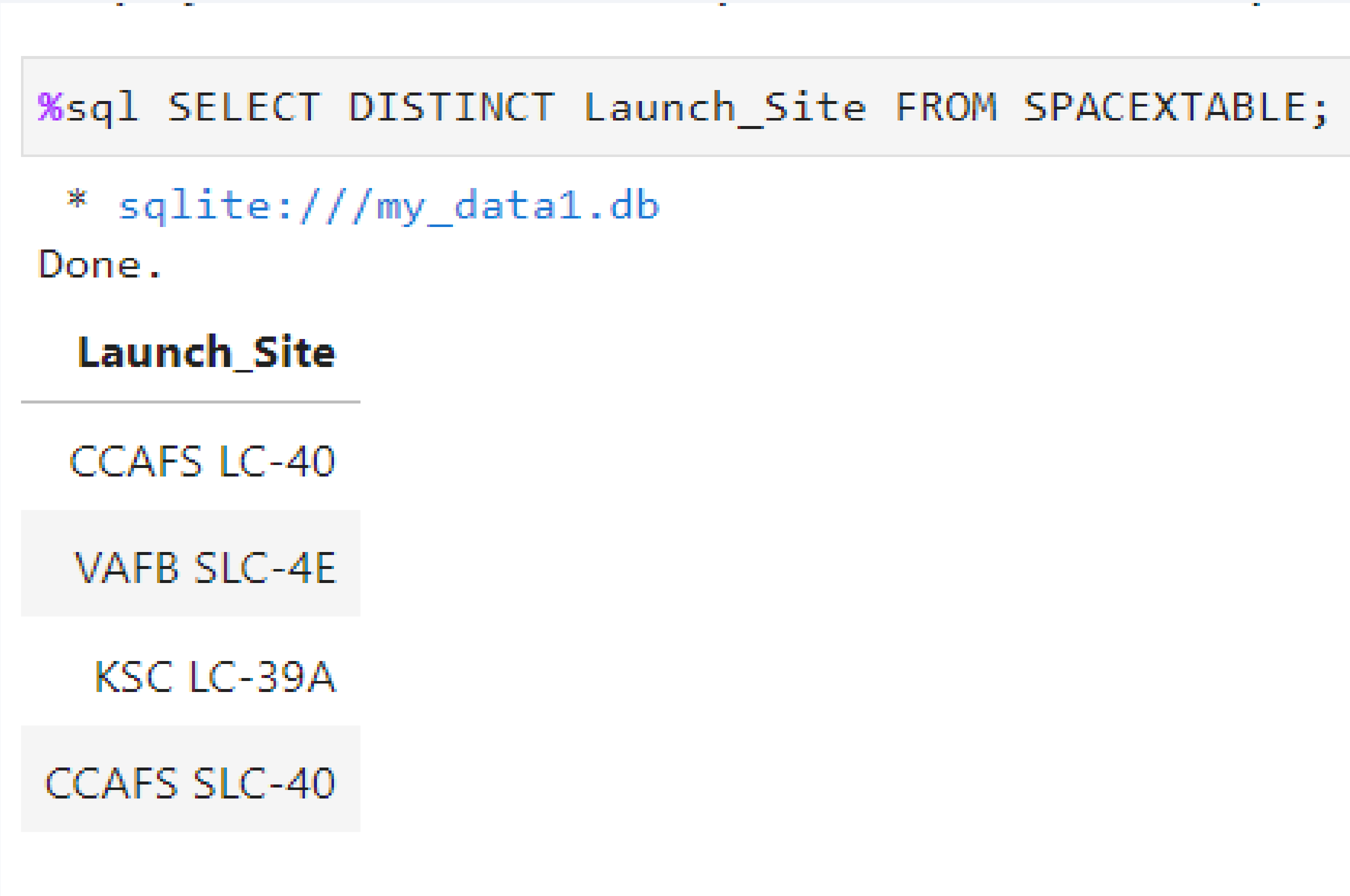
Launch Success Yearly Trend

From this plot, it can be observed that success rate since 2013 kept increasing till 2020, with an exception in 2018.



All Launch Site Names

To get the names of the unique launch sites, “DISTINCT” statement was used in the SQL query

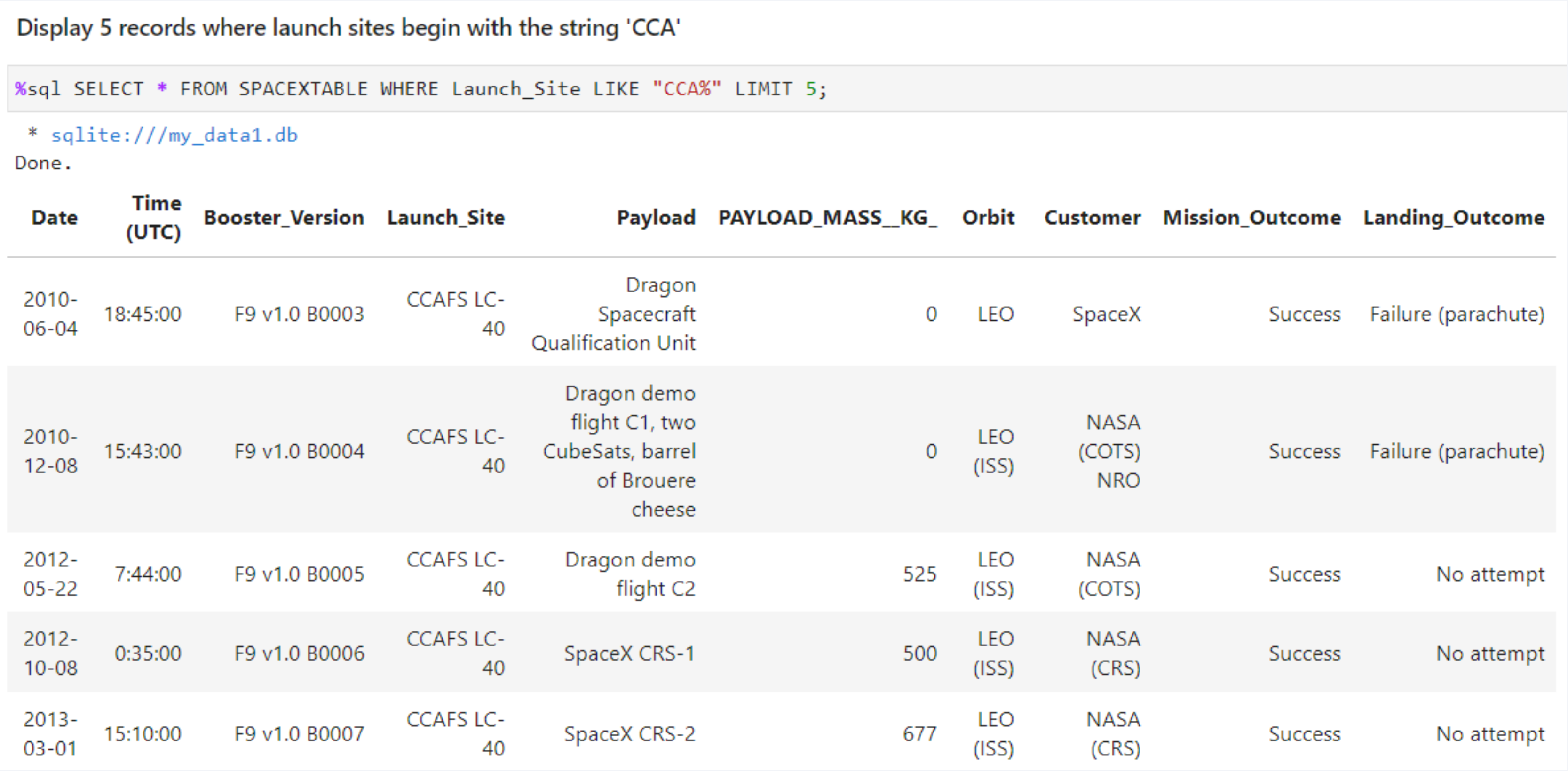


```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'



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

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5;
```

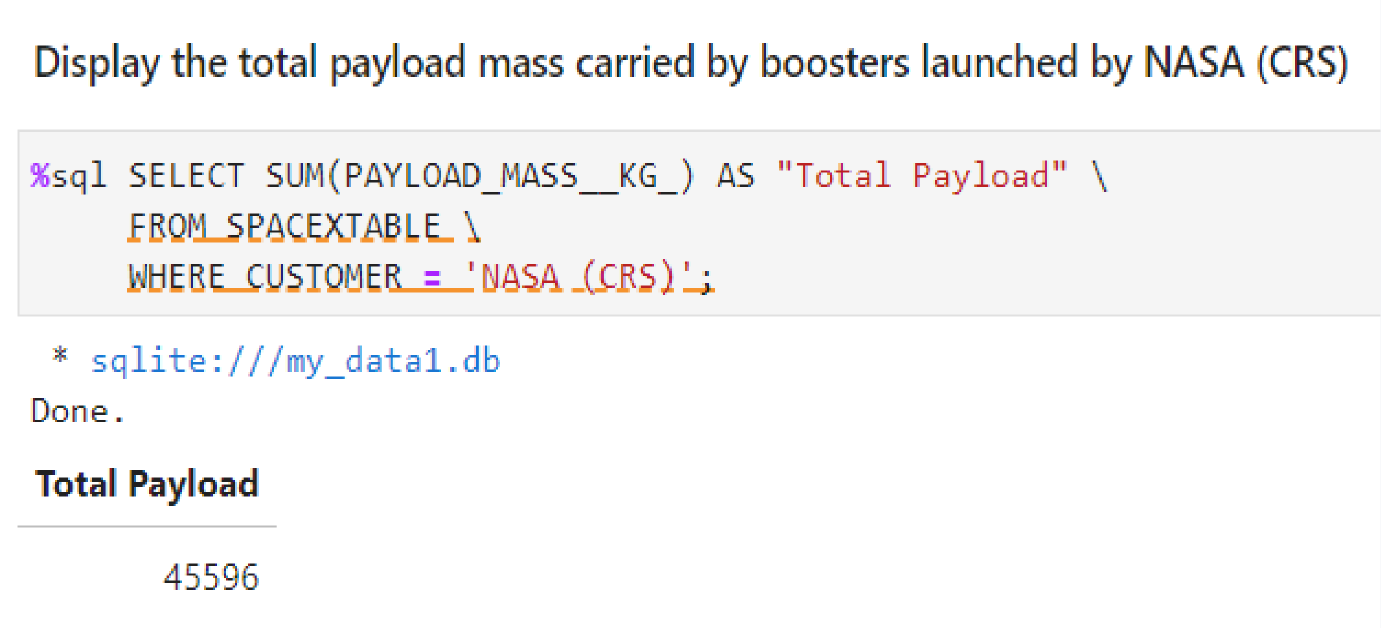
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

To get the names of the launch sites that begin with the string “CCA”, “LIKE ‘CCA%’ ” was used. To get only 5 records, “LIMIT 5” was used in the SQL query.

Total Payload Mass

Total payload mass carried by boosters launched by NASA (CRS) was calculated using the query in the figure where, “SUM ()” function was used to compute the total.



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

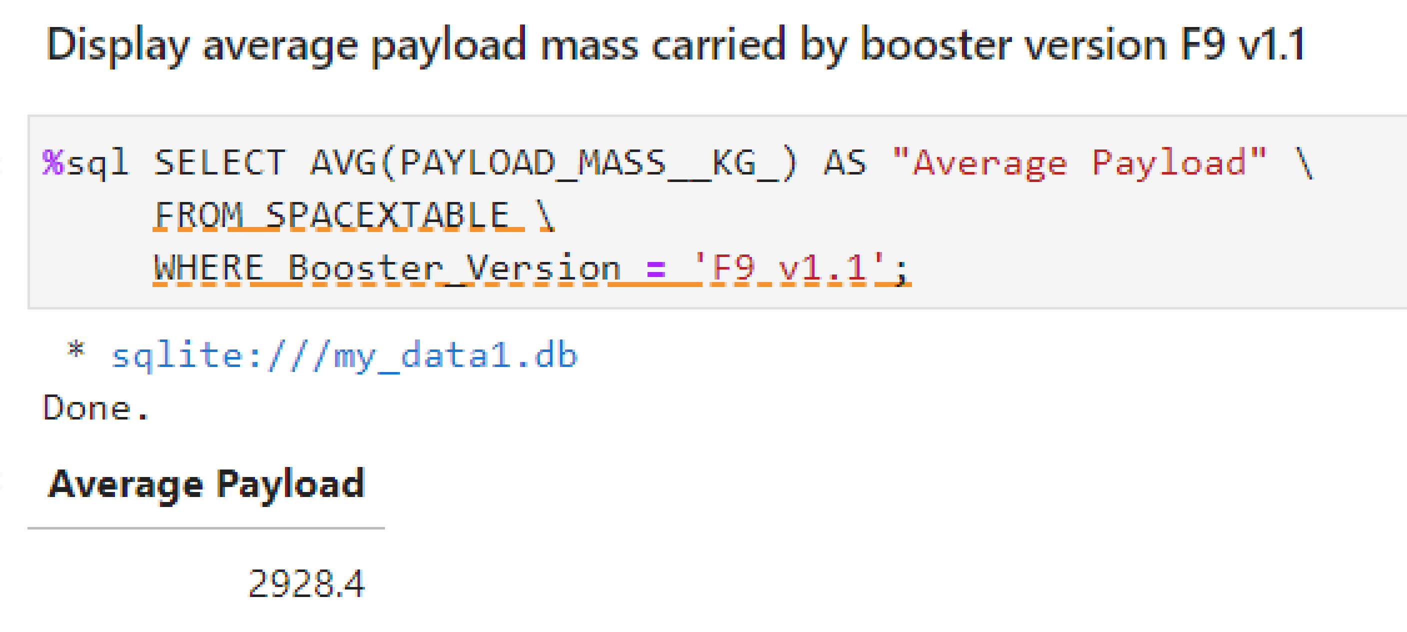
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload" \
     FROM SPACEXTABLE \
     WHERE CUSTOMER = 'NASA (CRS)';
```

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

Total Payload
45596

Average Payload Mass by F9 v1.1

Average payload mass carried by booster F9 v1.1 was calculated with the query in the figure using “AVG ()” function to compute the average



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

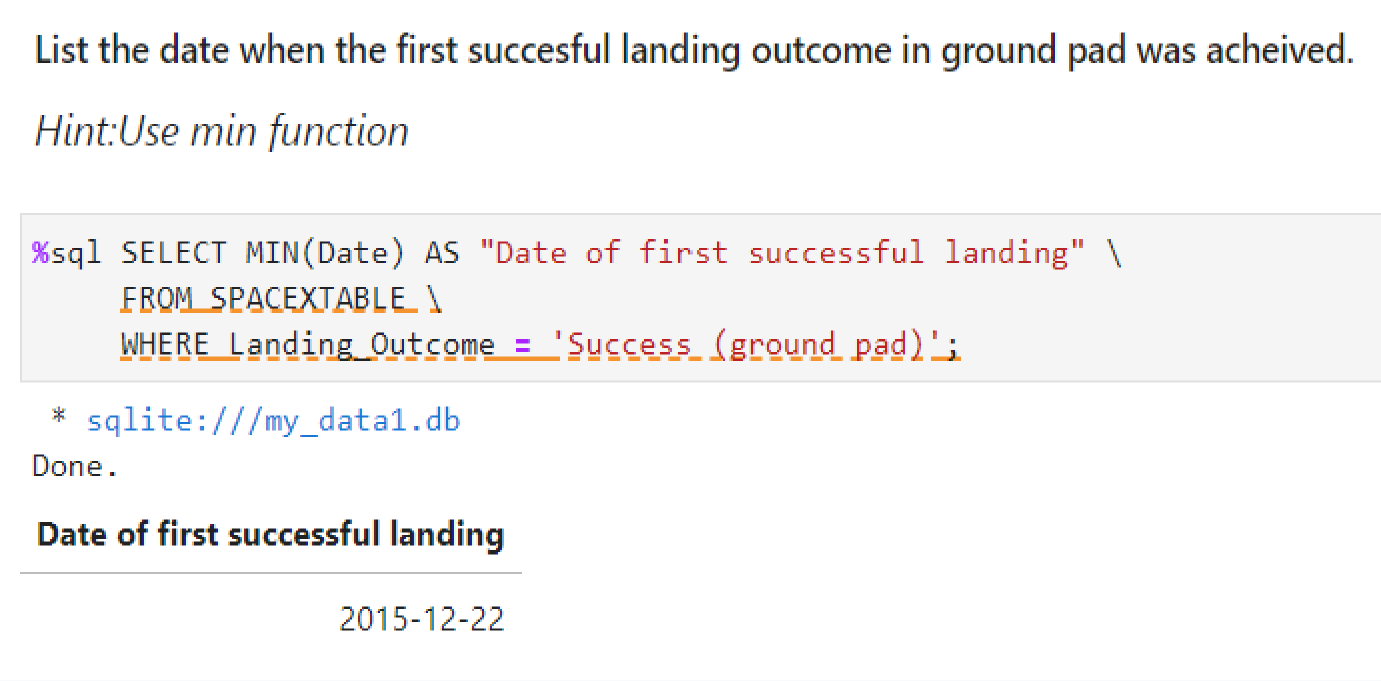
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload" \
     FROM SPACEXTABLE \
     WHERE Booster_Version = 'F9 v1.1';
```

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

Average Payload
2928.4

First Successful Ground Landing Date

Date of first successful landing outcome in ground pad was obtained using the “MIN()” function in the SQL query.



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

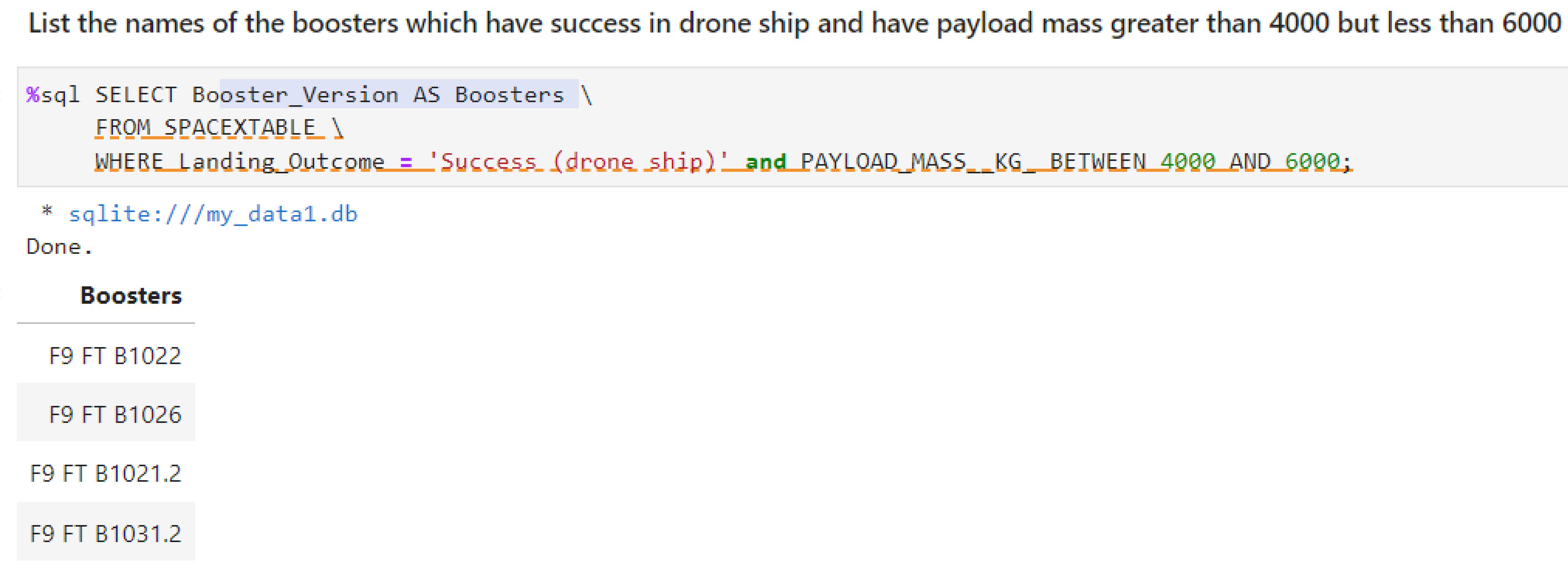
Hint:Use min function

```
%sql SELECT MIN(Date) AS "Date of first successful landing" \
     FROM SPACEXTABLE \
     WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Date of first successful landing
2015-12-22

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 AS Boosters \
     FROM SPACEXTABLE \
     WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

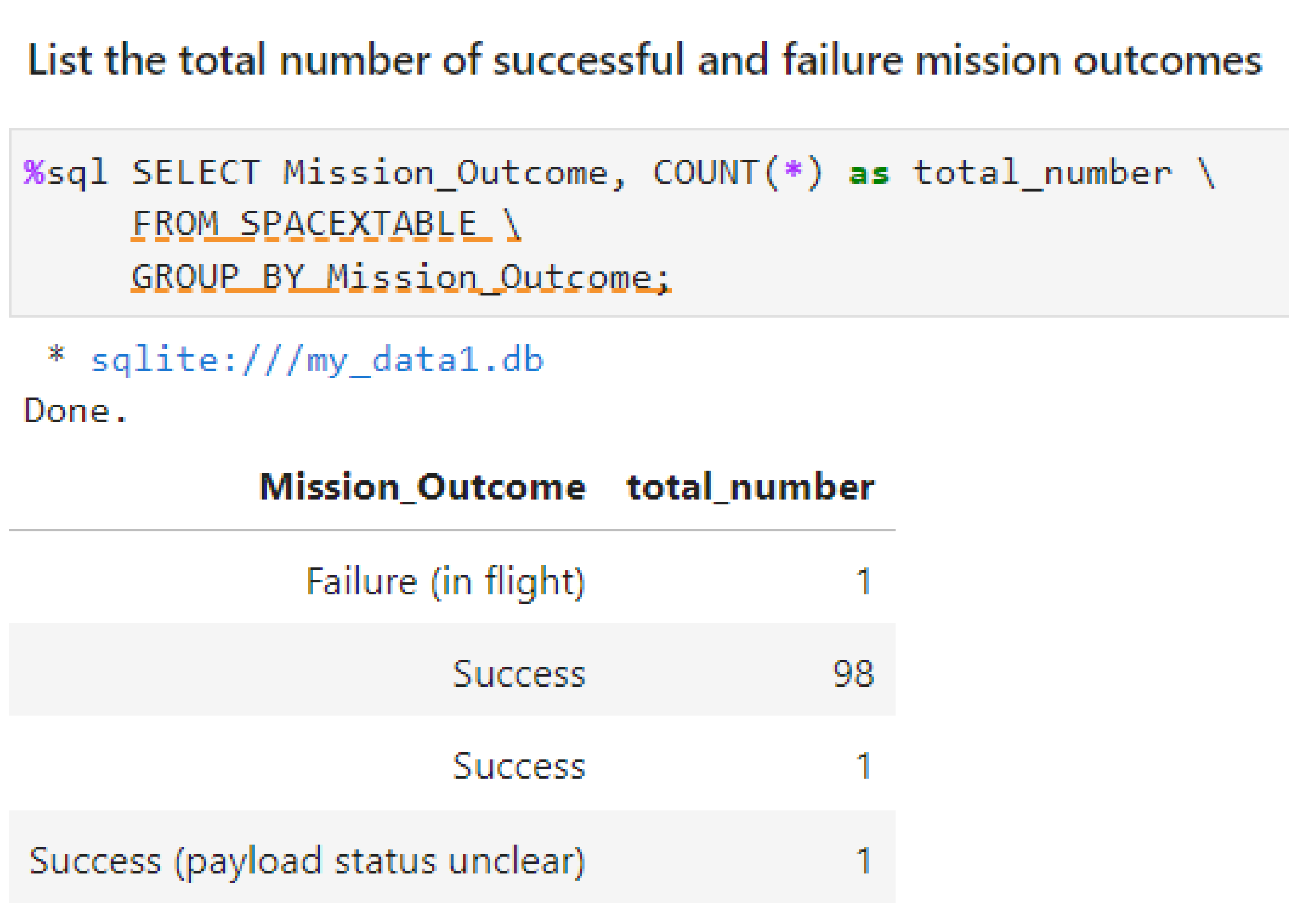
```
 * sqlite:///my_data1.db
Done.
```

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

“WHERE” clause has been used to filter the boosters which have success in drone ship and “BETWEEN” operator has been used to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

To calculate the total number of successful and failure mission outcomes, “COUNT ()” function and “GROUP BY” statement were used in the SQL query.



List the total number of successful and failure mission outcomes

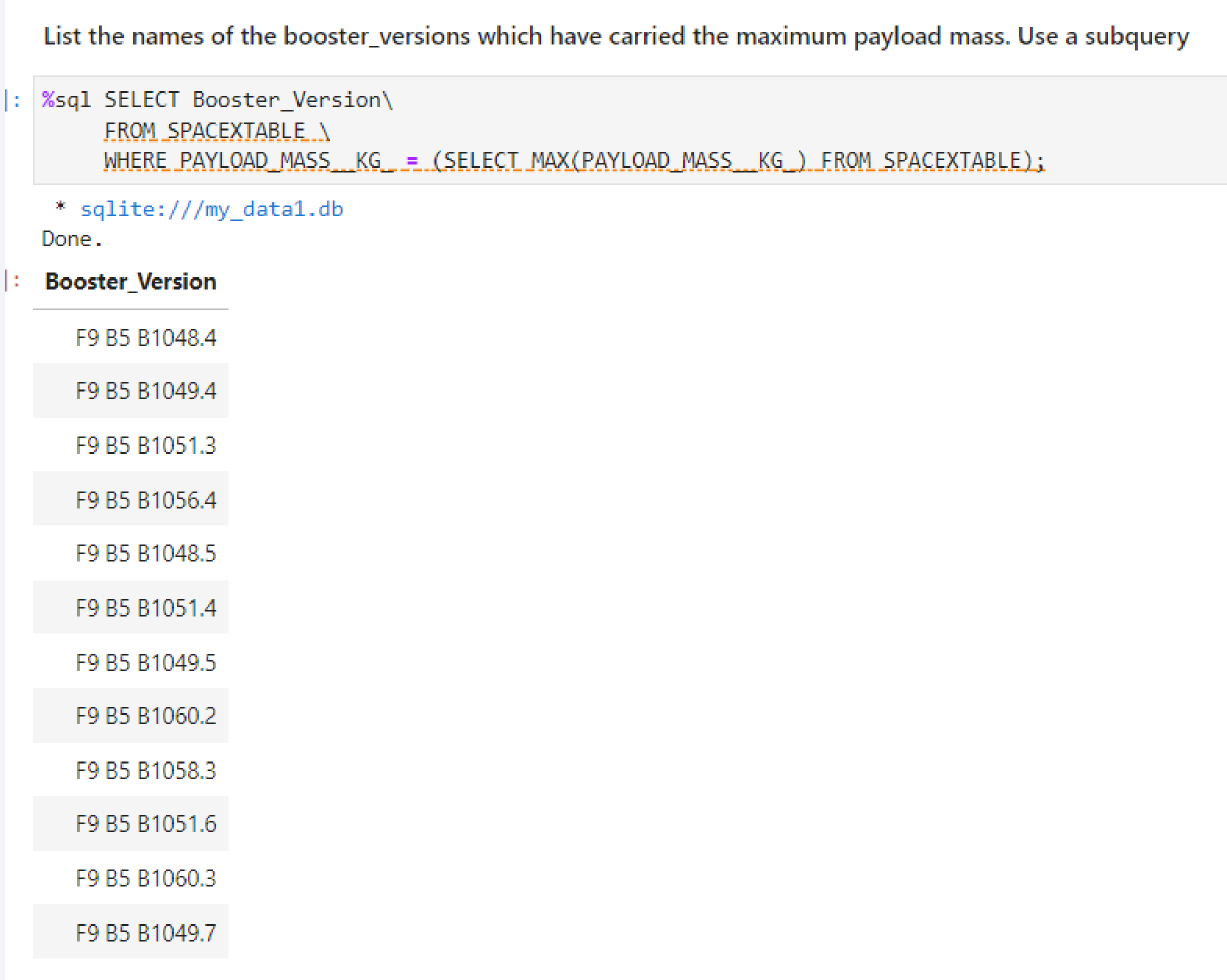
```
%sql SELECT Mission_Outcome, COUNT(*) as total_number \
     FROM SPACEXTABLE \
     GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

Names of the booster versions which have carried the maximum payload mass have been determined using a subquery in the “WHERE” clause and MAX() function.



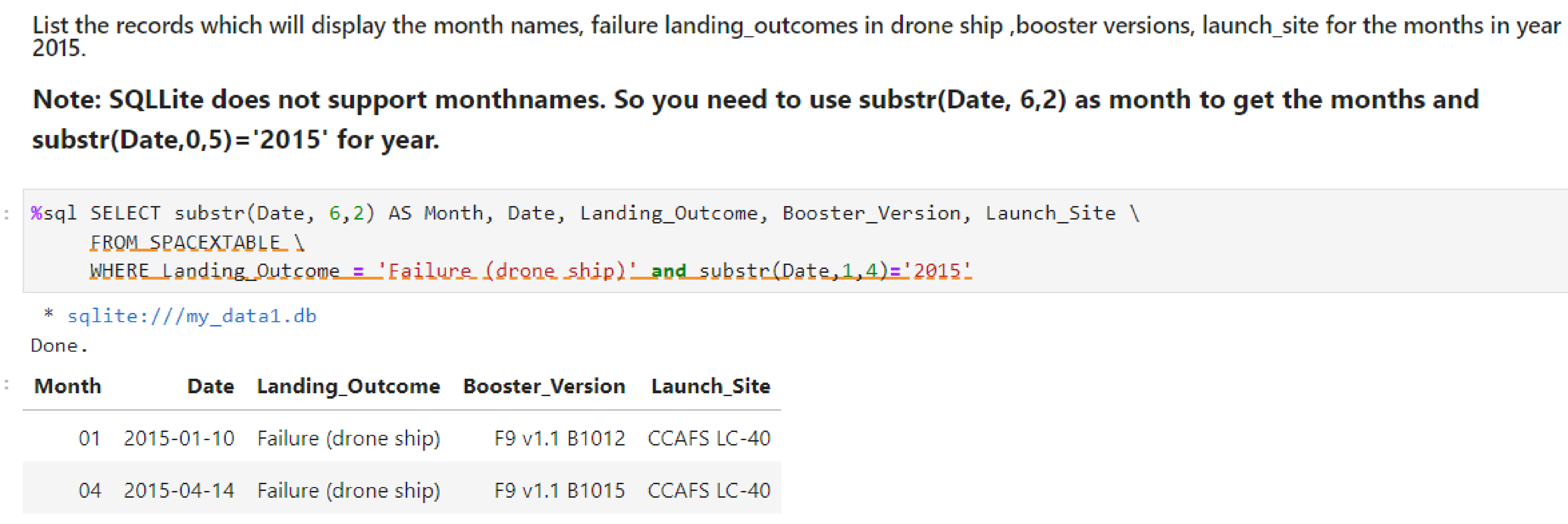
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
]: %sql SELECT Booster_Version\
        FROM SPACEXTABLE \
        WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

```
 * 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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date, 6,2) AS Month, Date, Landing_Outcome, Booster_Version, Launch_Site \
     FROM SPACEXTABLE \
     WHERE Landing_Outcome = 'Failure (drone ship)' and substr(Date,1,4)='2015'
```

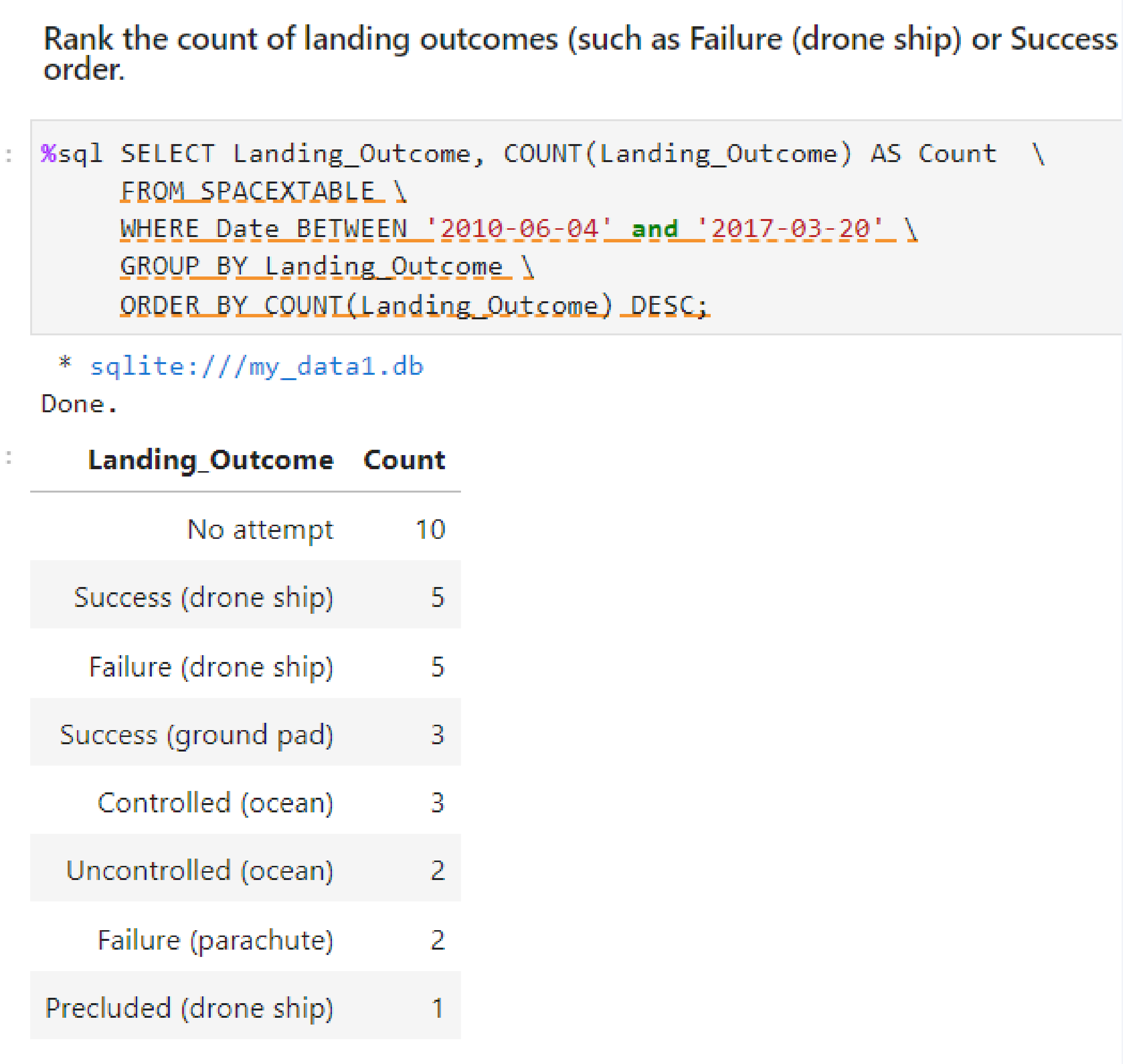
```
 * sqlite:///my_data1.db
Done.
```

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

Month names, failure landing outcomes in drone ship, booster versions, launch sites for the months in year 2015 have been displayed with the query in the figure using “AND” condition and substr() function the “WHERE” clause.

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

- Landing outcomes and the "COUNT" of landing outcomes were selected from the data and "WHERE" clause was used to filter the landing outcomes "BETWEEN" 2010-06-04 to 2017-03-20.
- "GROUP BY" clause was used to group the landing outcomes and "ORDER BY" clause was used to sort the grouped landing outcome in descending order.



Rank the count of landing outcomes (such as Failure (drone ship) or Success
order.

```
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Count  \
    FROM SPACEXTABLE \
    WHERE Date BETWEEN '2010-06-04' and '2017-03-20' \
    GROUP BY Landing_Outcome \
    ORDER BY COUNT(Landing_Outcome) DESC;
```

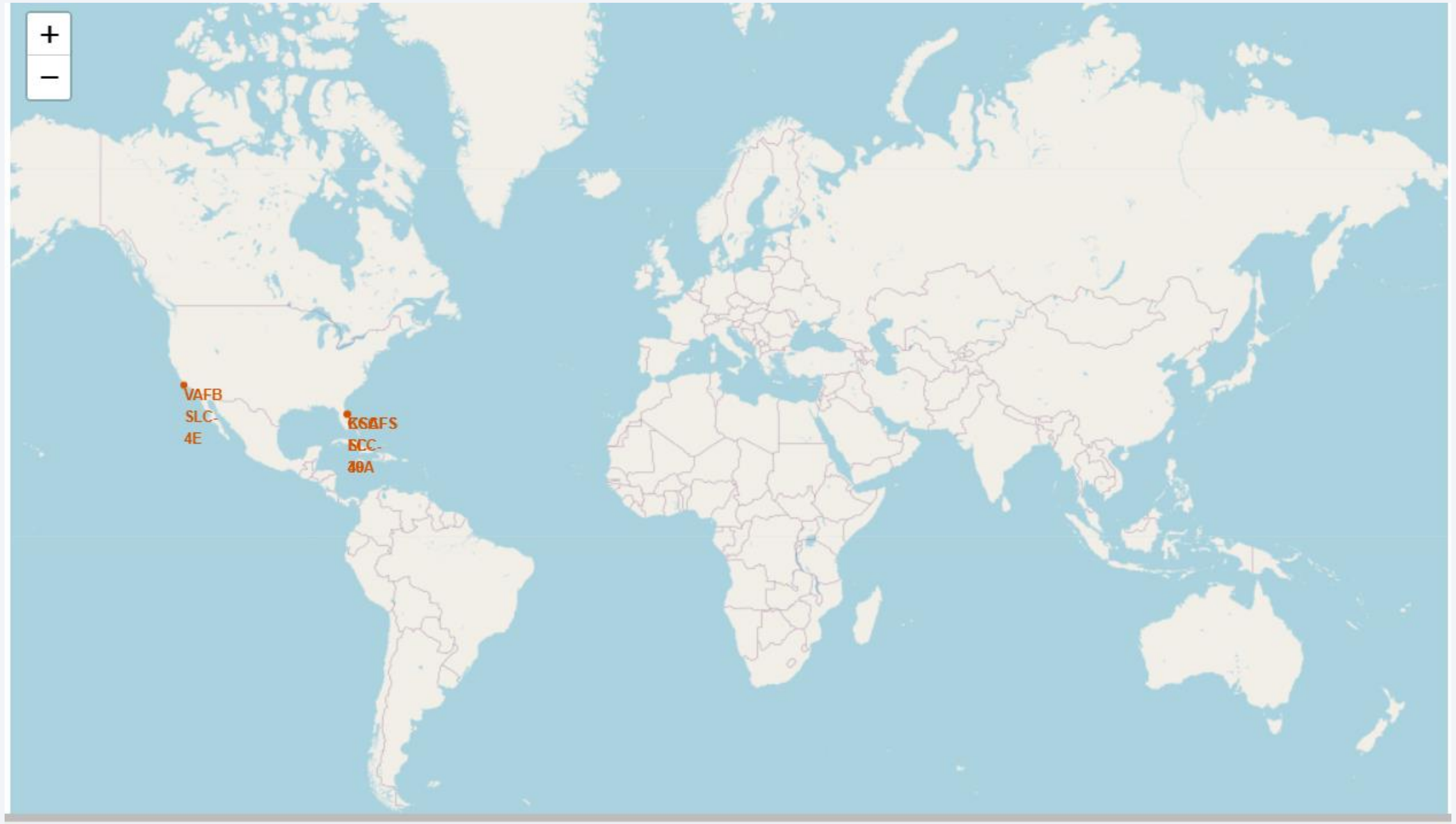
```
 * sqlite:///my_data1.db
Done.
```

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

Section 3

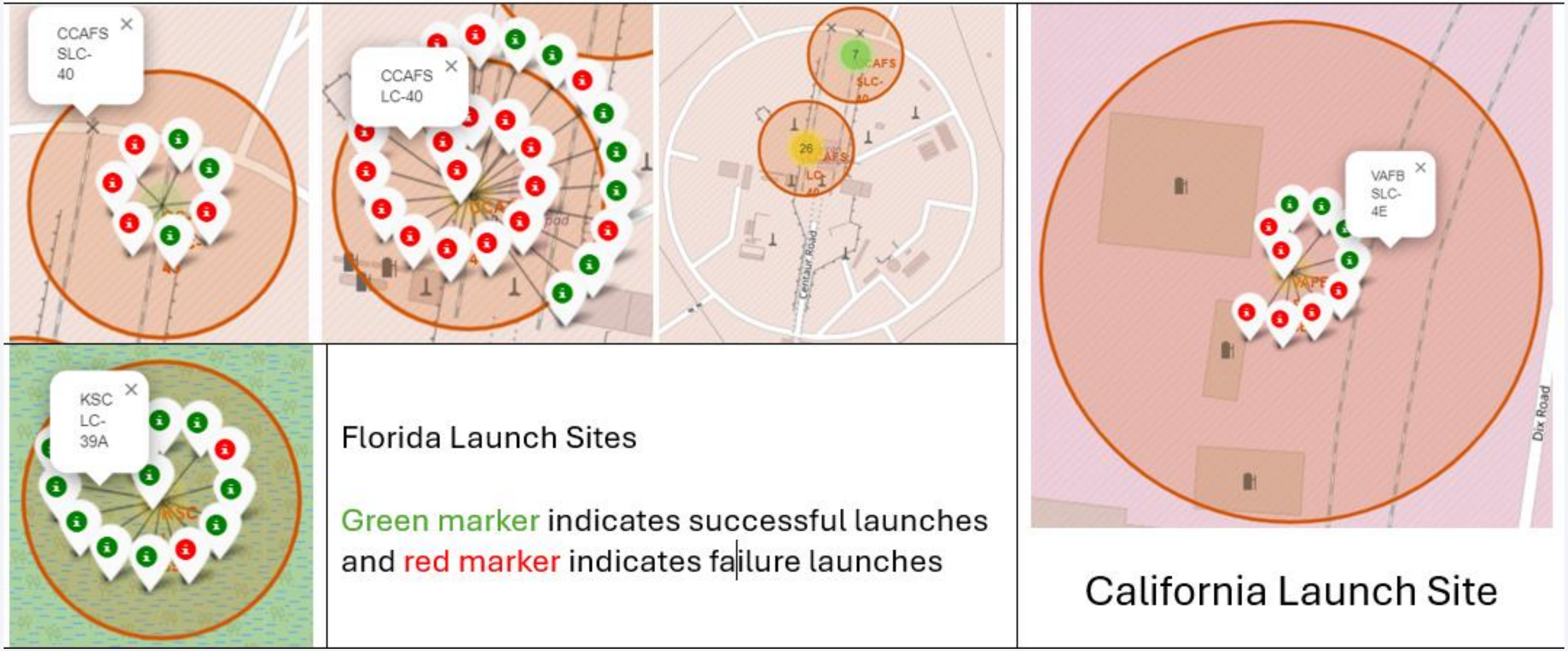
Launch Sites Proximities Analysis

All launch sites in the global map



All SpaceX launch sites are in two states of East and West coasts of USA, Florida and California, respectively.

Markers indicating launch sites with color labels



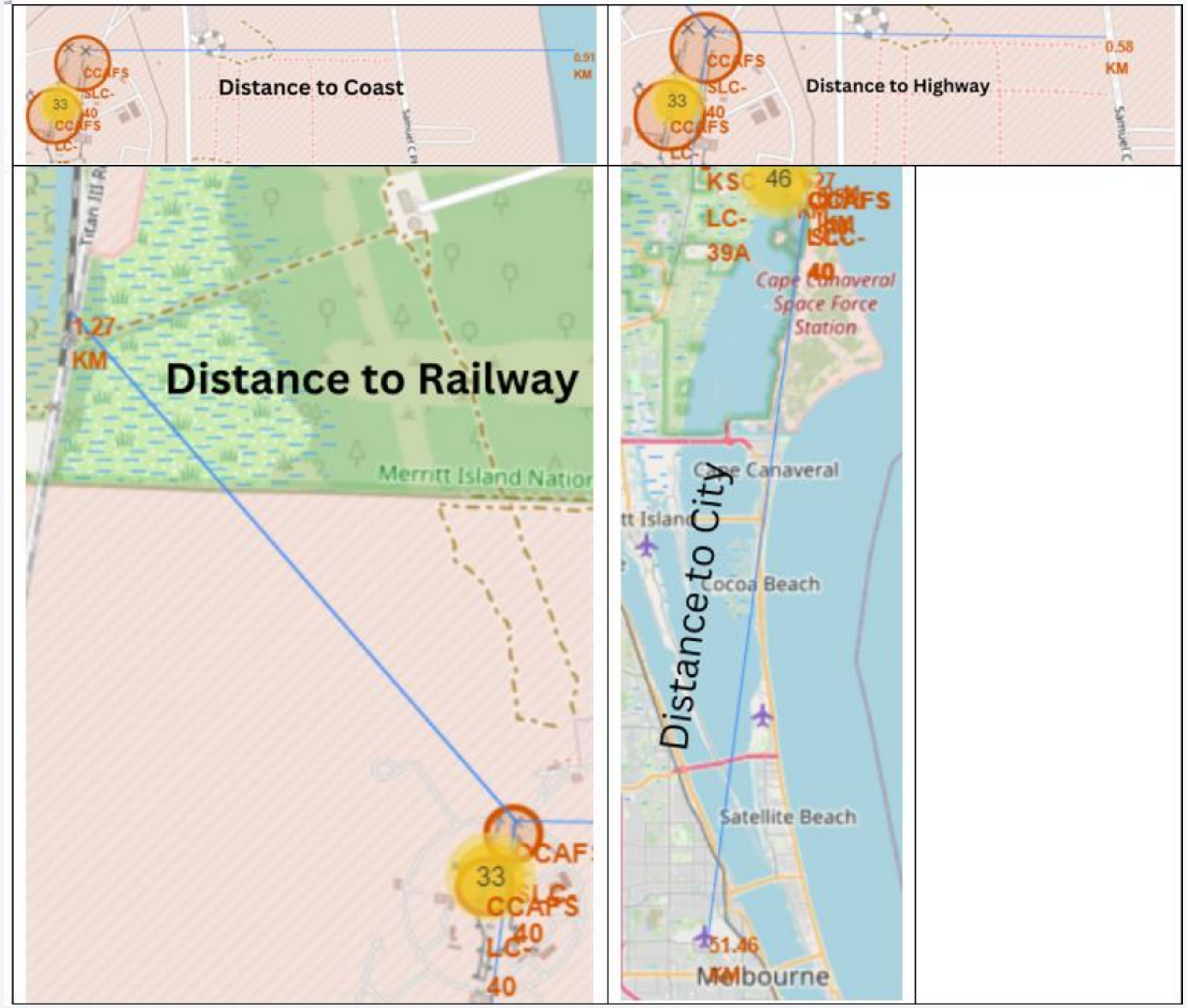
Florida Launch Sites

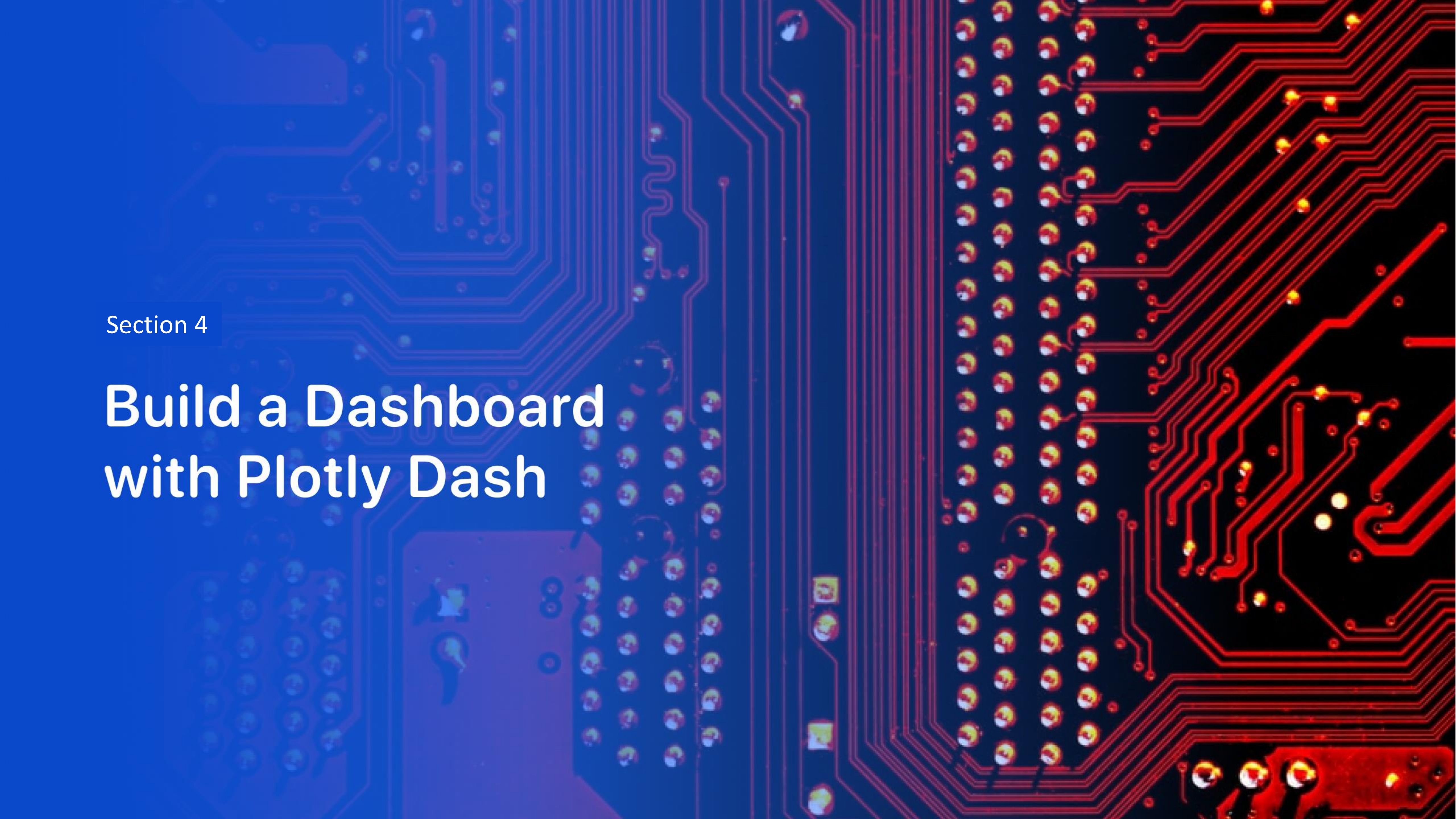
Green marker indicates successful launches and red marker indicates failure launches

California Launch Site

Distances from a launch site to landmarks

Distances from the launch sites to different landmarks such as closest point of coastline, highway, railway and closest city were calculated and shown on the maps using markers.

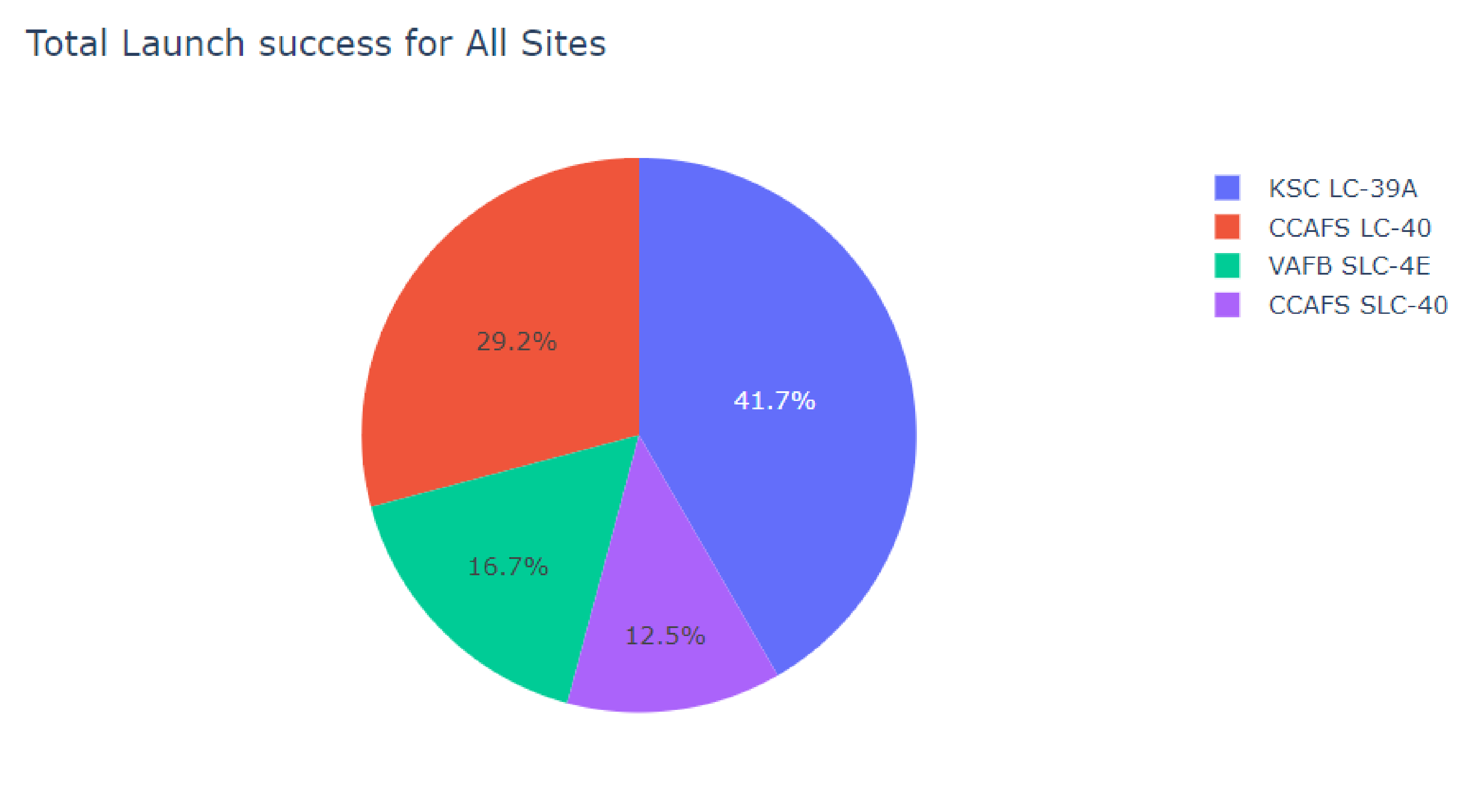




Section 4

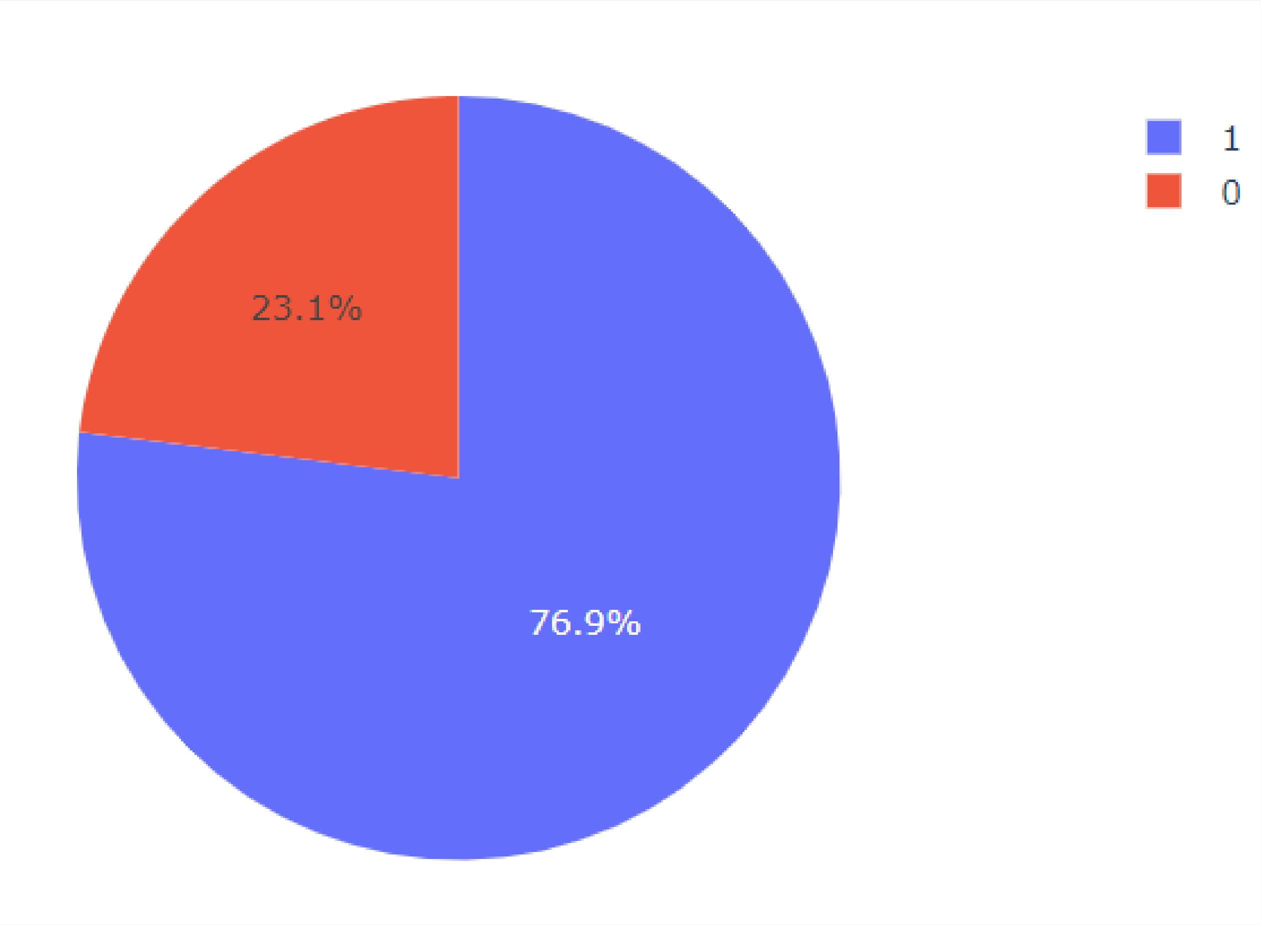
Build a Dashboard with Plotly Dash

Total Launch Success for All Sites in a Pie Chart



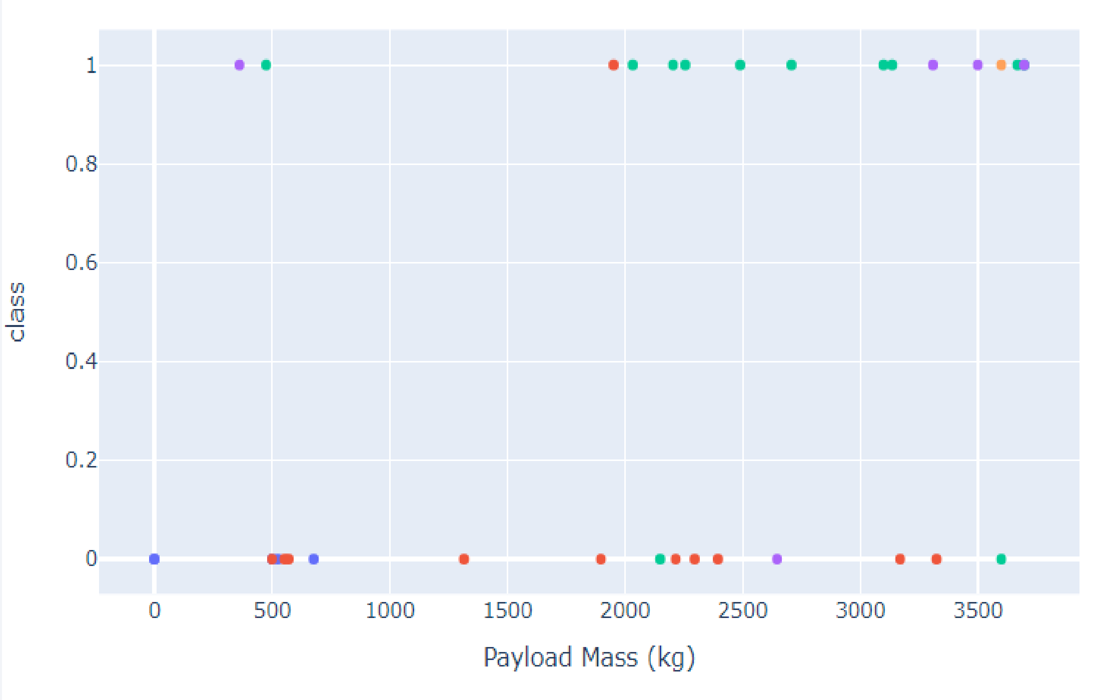
It can be observed that KSC LC-39A has the highest number of successful launches among all sites.

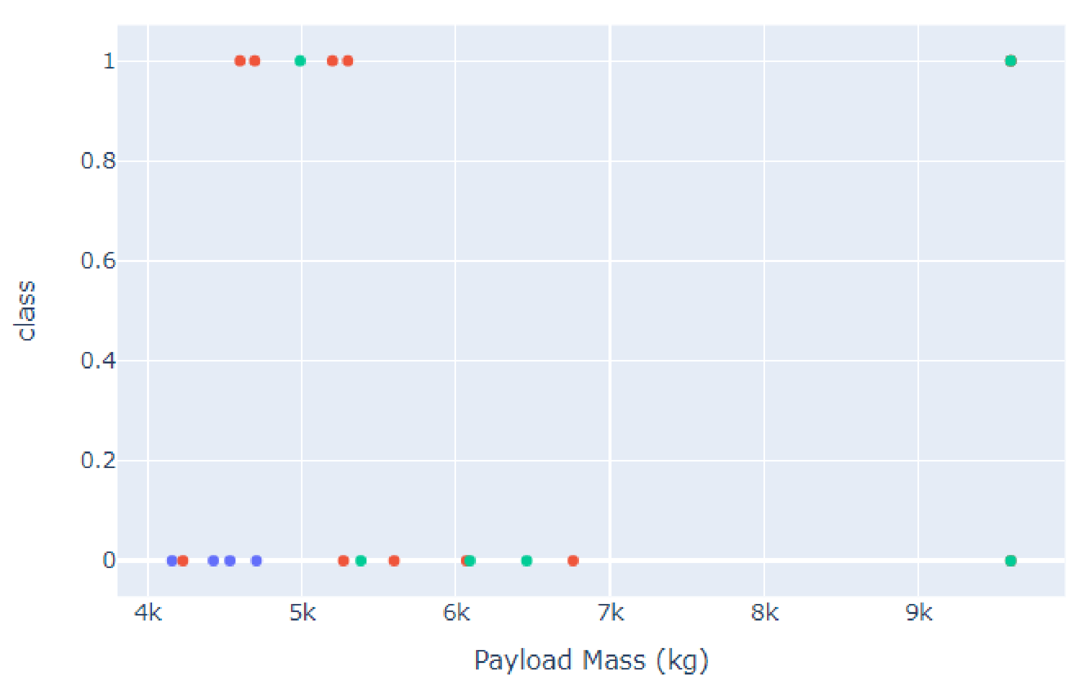
Pie Chart showing the Launch Site with Highest Success Ratio



KSC LC-39A has the highest launch success ratio, which is 76.9%

Scatter Plot of Payload vs Launch Outcome for All Sites with Different Payload





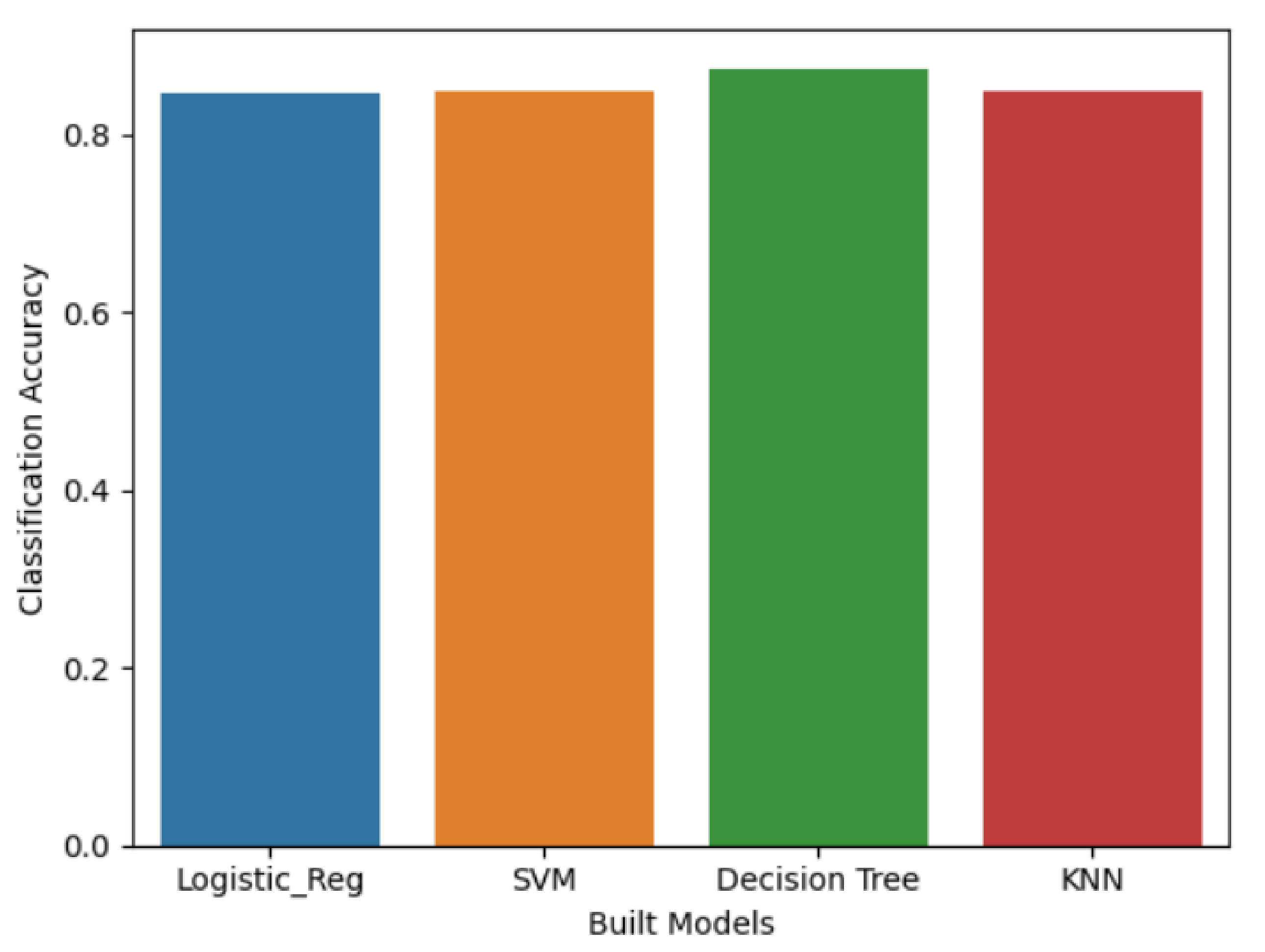
It can be observed that the success rates of low weighted payloads are higher than that of heavy weighted payloads

Section 5

Predictive Analysis (Classification)

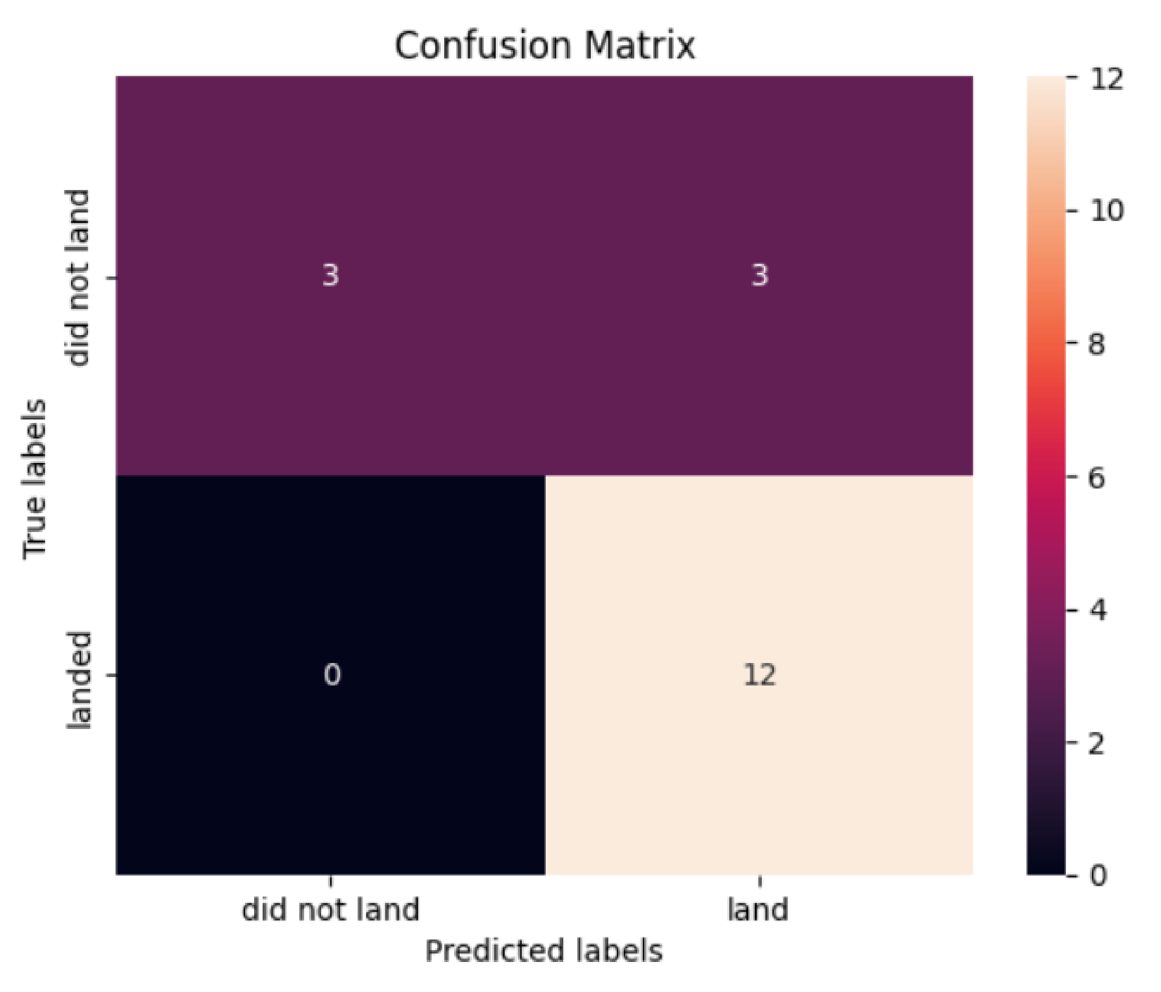
Classification Accuracy

It is evident from the bar plot that Decision Tree has the highest classification accuracy.



Confusion Matrix

The confusion matrix of the Decision Tree classifier indicates its ability to differentiate between classes. The primary issue lies in false positives, where unsuccessful landings are incorrectly labeled as successful by the classifier.



Conclusions

It can be concluded that:

- The larger the flight number at a launch site, the higher the success rate at a launch site.
- Launch success rate started to increase from 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the highest success rates.
- KSC LC-39A had the most successful launches among all sites.
- Decision tree classifier is the best machine learning algorithm for this prediction task.

Thank you!