IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ing. Anna Rývová
2024

Outline

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

Executive Summary

- Summary of methodologies
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with Web Scraping
  - SpaceX Data Wrangling
  - SpaceX Exploratory Data Analysis using SQL
  - Space-X EDA DataViz Using Python Pandas and Matplotlib
  - Space-X Launch Sites Analysis with Folium - Interactive Visual Analytics and Plotly Dash
  - SpaceX Machine Learning Landing Prediction
- Summary of all results
  - EDA results
  - - Interactive Visual Analytics and Dashboards
  - - Predictive Analysis(Classification)

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.

Problems you want to find answers

In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Data was collected through two methods: requesting data from the SpaceX API and web scraping launch data from a Wikipedia page.

- Perform data wrangling

  Data wrangling was then performed to transform and clean the data using Python's pandas library.

- Perform exploratory data analysis (EDA) using visualization and SQL

  We using visualization tools such as Python's matplotlib and seaborn libraries and using SQL queries.

- Perform interactive visual analytics using Folium and Plotly Dash

  Python's interactive visualization packages were used to answer some analytical questions. Folium was used for creating maps while Plotly Dash was used to create interactive data visualizations.

- Perform predictive analysis using classification models

  We used models for logistics regression, support vector machines, k-nearest neighbour and decision tree classifier. Each model was trained, tuned and evaluated to find the best one.

Data Collection

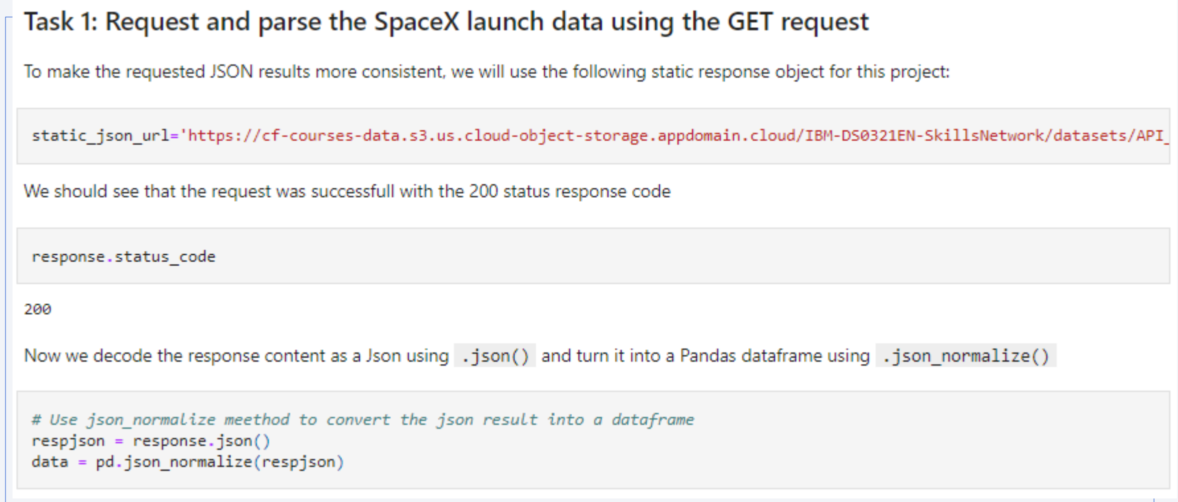
1. Collected data using SpaceX REST API
2. Decode response content as a JSON data format
3. Convert JSON to Pandas dataframe
4. Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia web page
5. Extract only the Falcon 9 launch HTML table using BeautifulSoup
6. Handle missing values
7. Parse the table and converted it into a Pandas dataframe.
8. Export the dataframe to *.csv file

Data Collection – SpaceX API

Data Collection Process

- Collected data using SpaceX REST API
- 2. Decode response content as a JSON data format
- Convert JSON to Pandas dataframe

GitHub URL: 1. Space-X Data Collection API.ipynb



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

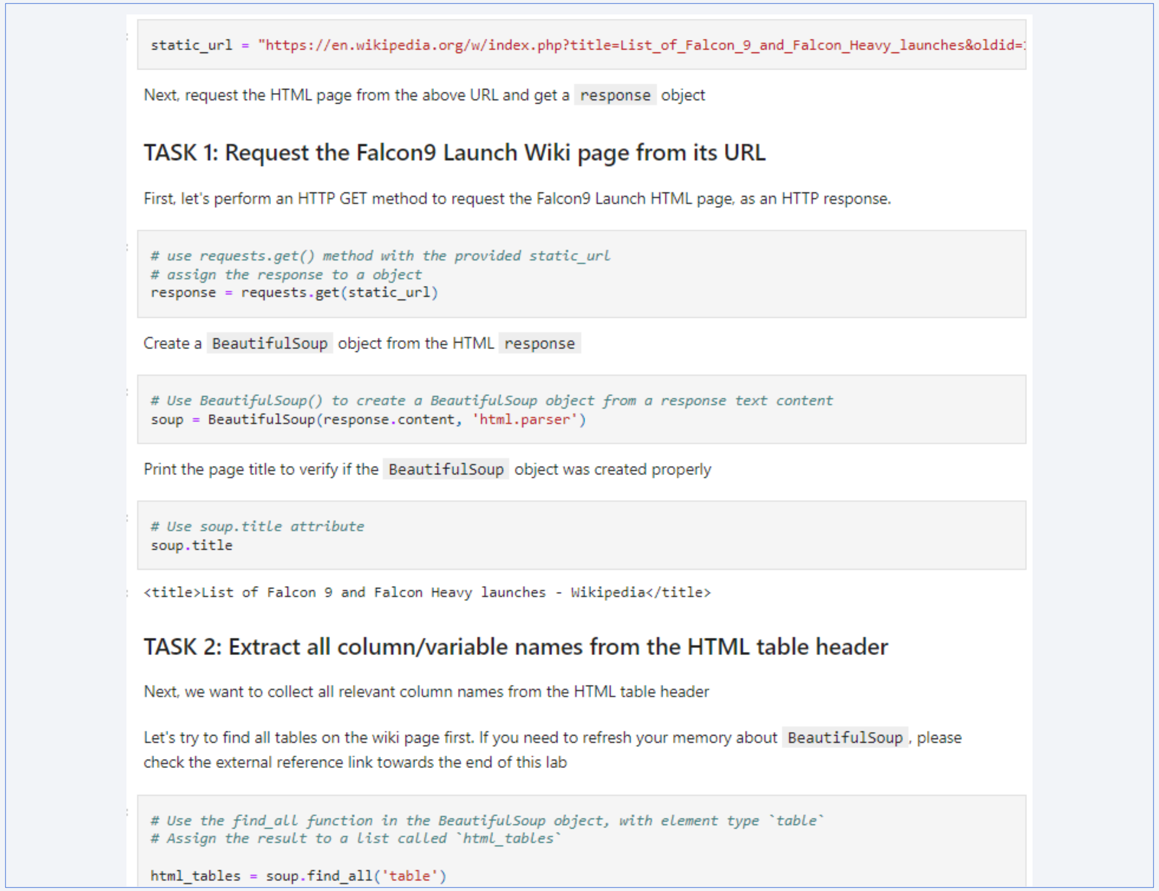
```
# Use json_normalize meethod to convert the json result into a dataframe
respjson = response.json()
data = pd.json_normalize(respjson)
```

Data Collection - Scraping

Web scraping process

- Request rocket launch data from its Wikipedia page
- Create a BeautifulSoap object from a response text content
- Extract all column/variable names from the HTML table header

GitHub URL: [2. Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia.ipynb]()



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform 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)
```

Create a `BeautifulSoup` object from the HTML `response`

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

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
soup.title
```

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

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

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

Data Wrangling

Data wrangling process:

- 1. Import data from a *.csv file to Pandas dataframe
- 2. Calculate the number of launches on each site
- 3. Calculate the number and occurence of each orbit
- 4. Calculate the number and occurence of mission outcome per orbit type
- 5. Create a landing outcome label from Outcome column
- 7. Export to a *.csv file

GitHub URL: 3. Space-X Data Wrangling spacex.ipynb

EDA with Data Visualization

Scatter plots

Scatter plots were used to represent the relationship between two variables. Different sets of features were compared such as Flight Number vs. Launch Site, Payload vs. Launch Site, Flight Number vs. Orbit Type and Payload vs. Orbit Type.

Bar chart

Bar charts were used makes it easy to compare values between multiple groups at a glance. The x-axis represents a category and the y-axis represents a discrete value. Bar charts were used to compare the Success Rate for different Orbit Types

Line chart

Line charts are useful for showing data trends over time. A line chart was used to show Success Rate over a certain number of Years.

GitHub URL: 5. Space-X EDA DataViz Using Pandas and Matplotlib - SpaceX.ipynb

EDA with SQL

The following SQL queries were performed for EDA:

- 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 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_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub URL: [4. Space-X EDA Using SQL.ipynb]

Build an Interactive Map with Folium

Objects were created and added to a Folium map. Marker objects were used to show all launch sites on a map as well as the successful/failed launches for each site on the map. Line objects were used to calculate the distances between a launch site to its proximities.

By adding these objects, following geographical patterns about launch sites are found:

- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

GitHub URL: 6.Space-X Launch Sites Locations Analysis with Folium-Interactive Visual Analytics.ipynb

Build a Dashboard with Plotly Dash

The dashboard application contains two charts:

- A **pie chart** shows the successful launch by each site. This chart shows the distribution of landing outcomes across all launch sites or show the success rate of launches on individual sites.
- A **scatter chart** shows the relationship between landing outcomes on the payload mass of different boosters. The dashboard takes two inputs, namely the site(s) and payload mass. This chart shows how different variables affect the landing outcomes.

GitHub URL: 7. Build an Interactive Dashboard with Ploty Dash - spacex_dash_app.ipynb

Predictive Analysis (Classification)

Predictive Analysis Process:

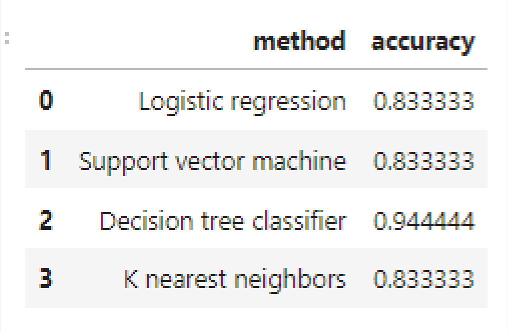
- Import data from *.csv file into Pandas dataframe
- Create a NumPy array from the column Class in data
- Standardize the data
- Split the data X and Y into training and test data
- Create a logistic regression object and a GridSearchCV object, fit the object to find the best parameters, calculate the accuracy on the test data
- Create a support vector machine object and a GridSearchCV object, fit the object to find the best parameters, calculate the accuracy on the test data
- Create a decision tree classifier object and a GridSearchCV object, fit the object to find the best parameters, calculate the accuracy on the test data
- Create a k nearest neighbors object and a GridSearchCV object, fit the object to find the best parameters, calculate the accuracy on the test data
- Compare the test data accuracy score for each of the methods

GitHub URL: 8. SpaceX Machine Learning Prediction.ipynb

Results

- The results of the exploratory data analysis revealed that the success rate of the Falcon 9 landings was 66.66 %.

- The predictive analysis results showed that the Decision Tree algorithm was the best classification method with an accuracy of 94 %.



	method	accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.944444
3	K nearest neighbors	0.833333

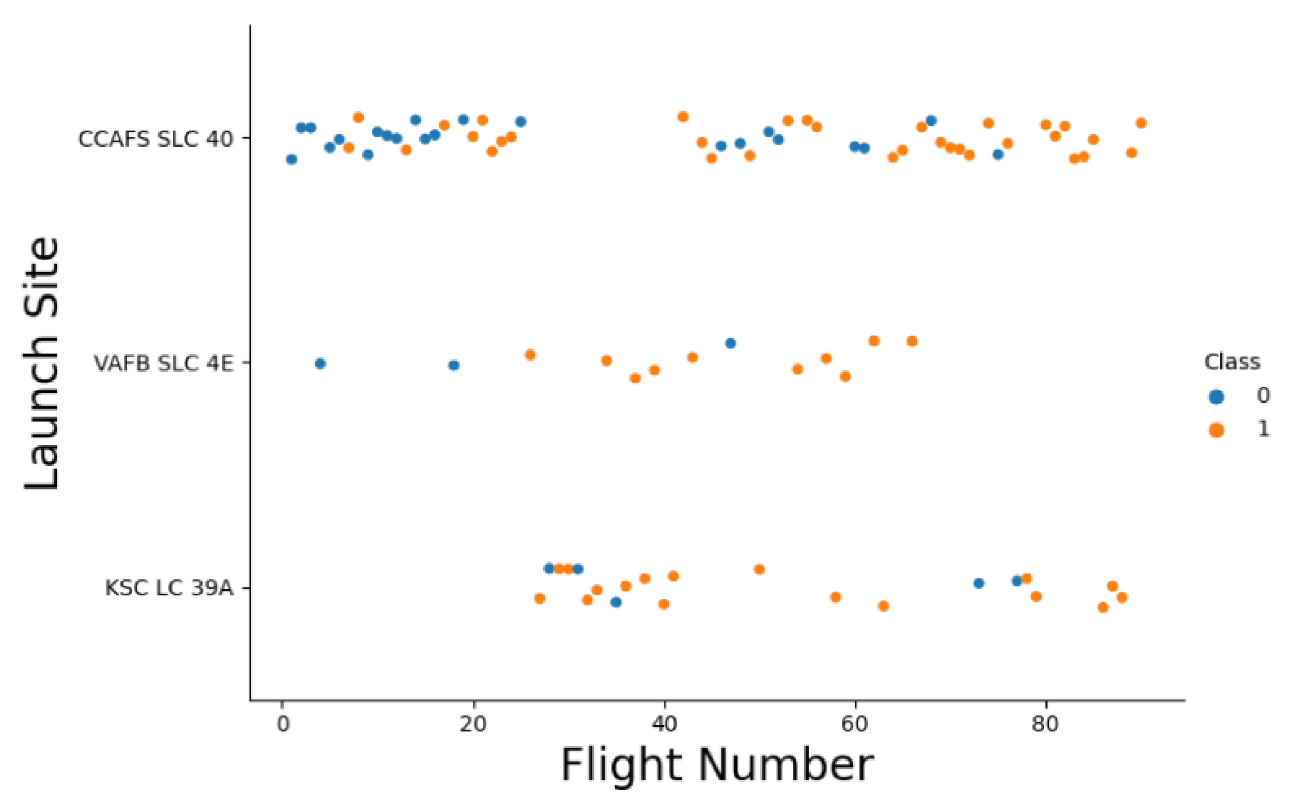
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Blue dots - unsuccessful launches
Orange dots - successful launches

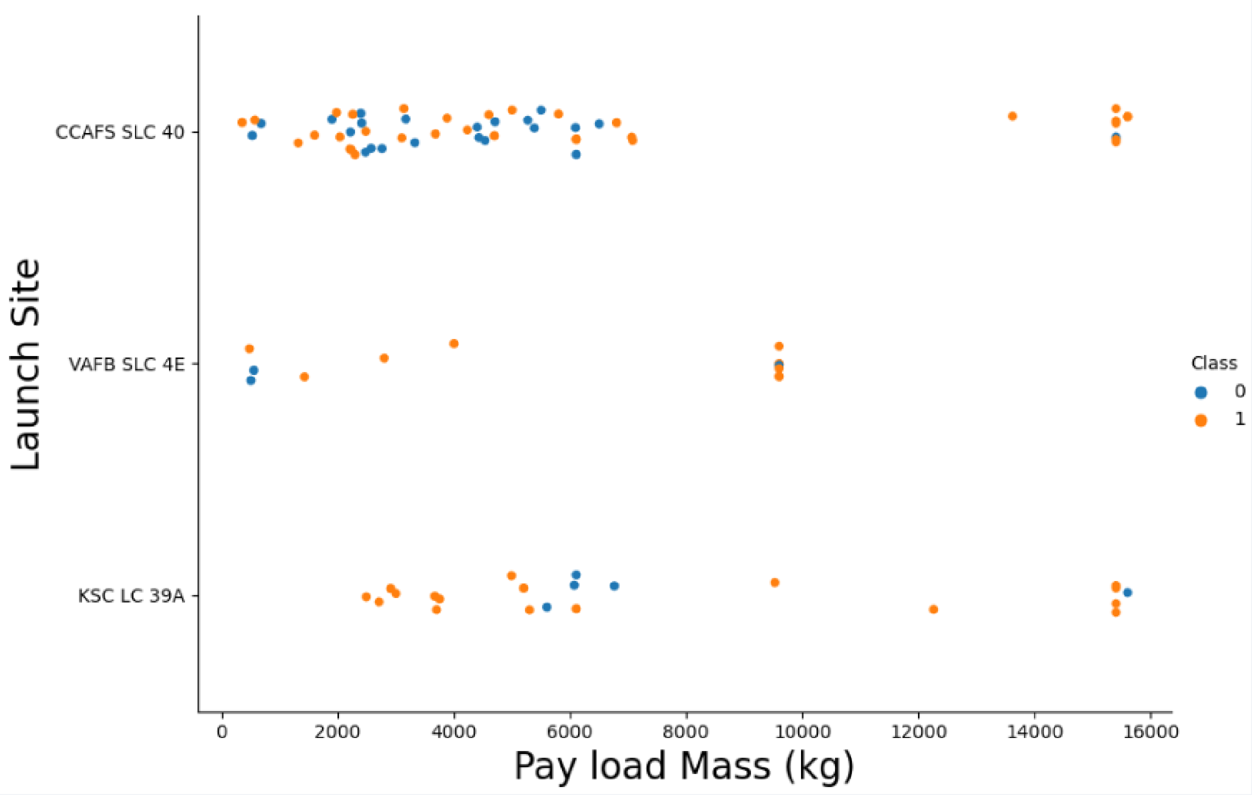
We see, as the **flight number increases in each of the 3 launch sites, so does the success rate.** The success rate for the VAFB SLC 4E launch site is 100 % after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100 % success rate after 80th flight. **All three sites have most of the failed landings for Flight Number below 20.**



Payload vs. Launch Site

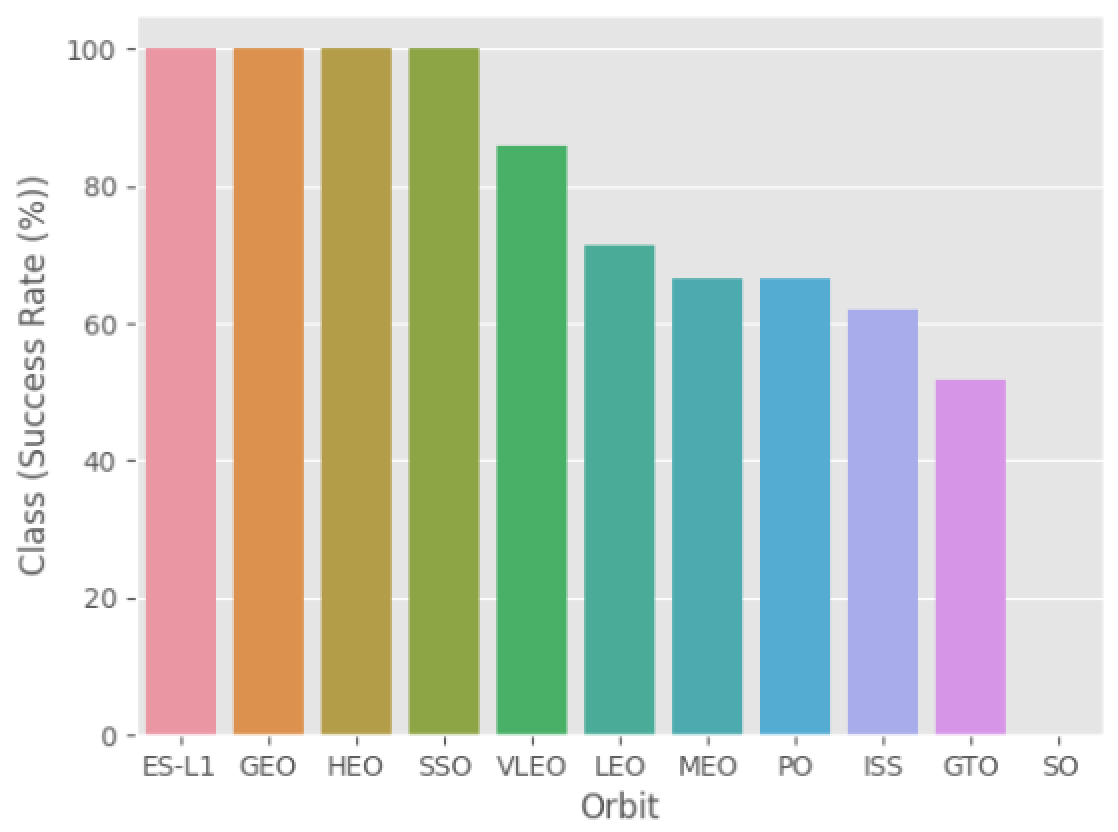
Blue dots - unsuccessful launches
Orange dots - successful launches

- For the VAFB-SLC launchsite there are no rockets launched for payload mass greater then 10000 kg.
- Most of the unsuccessful landings is for payload mass les then 10000 kg.
- If payload mass is greater than 10000 kg, almost all of landings are successful.



Success Rate vs. Orbit Type

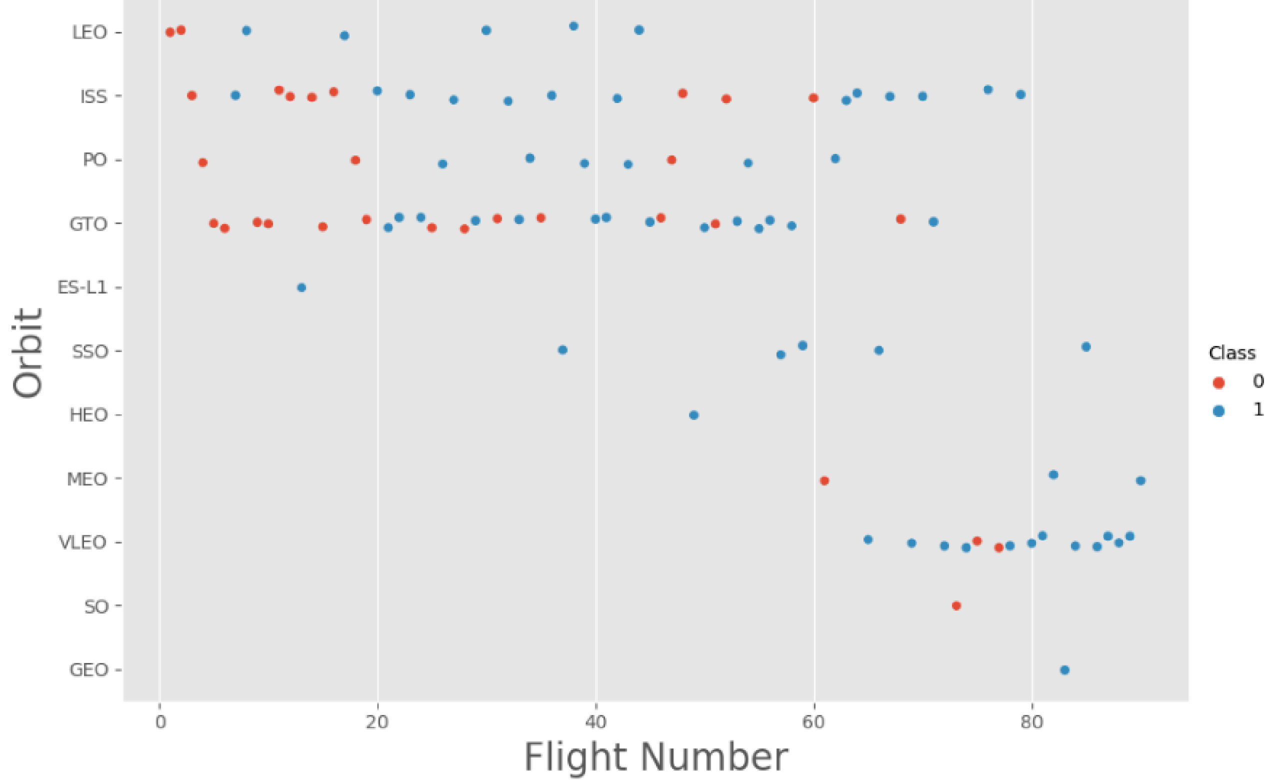
- Orbits SSO, HEO, GEO, and ES-L1 have 100% success rates.
- SO orbit has 0% success rate.
- All remaining orbits have a success rate between 50 and 100 %.



Flight Number vs. Orbit Type

Orange dot – unsuccessful landing
Blue dot – successful landing

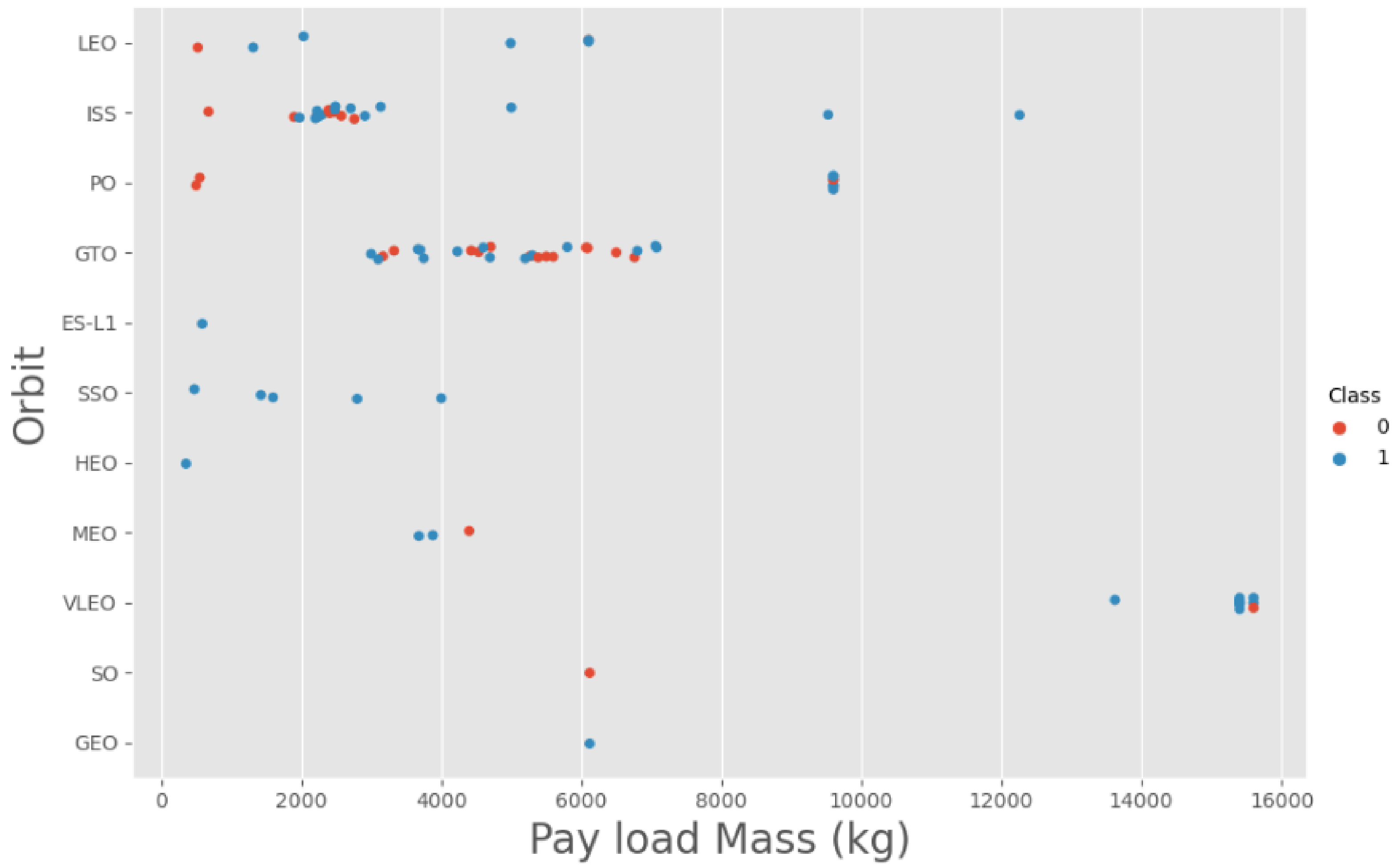
- LEO orbit - unsuccessful landing has very low flight number.
- SSO orbit - only successful landing
- ES-L1, HEO, GEO orbits - only successful landings, but we have very little data
- GTO, PO orbit - all flight with number lower than 20 was unsuccessful
- All orbits - all flights with number greater than 80 is successful



Payload vs. Orbit Type

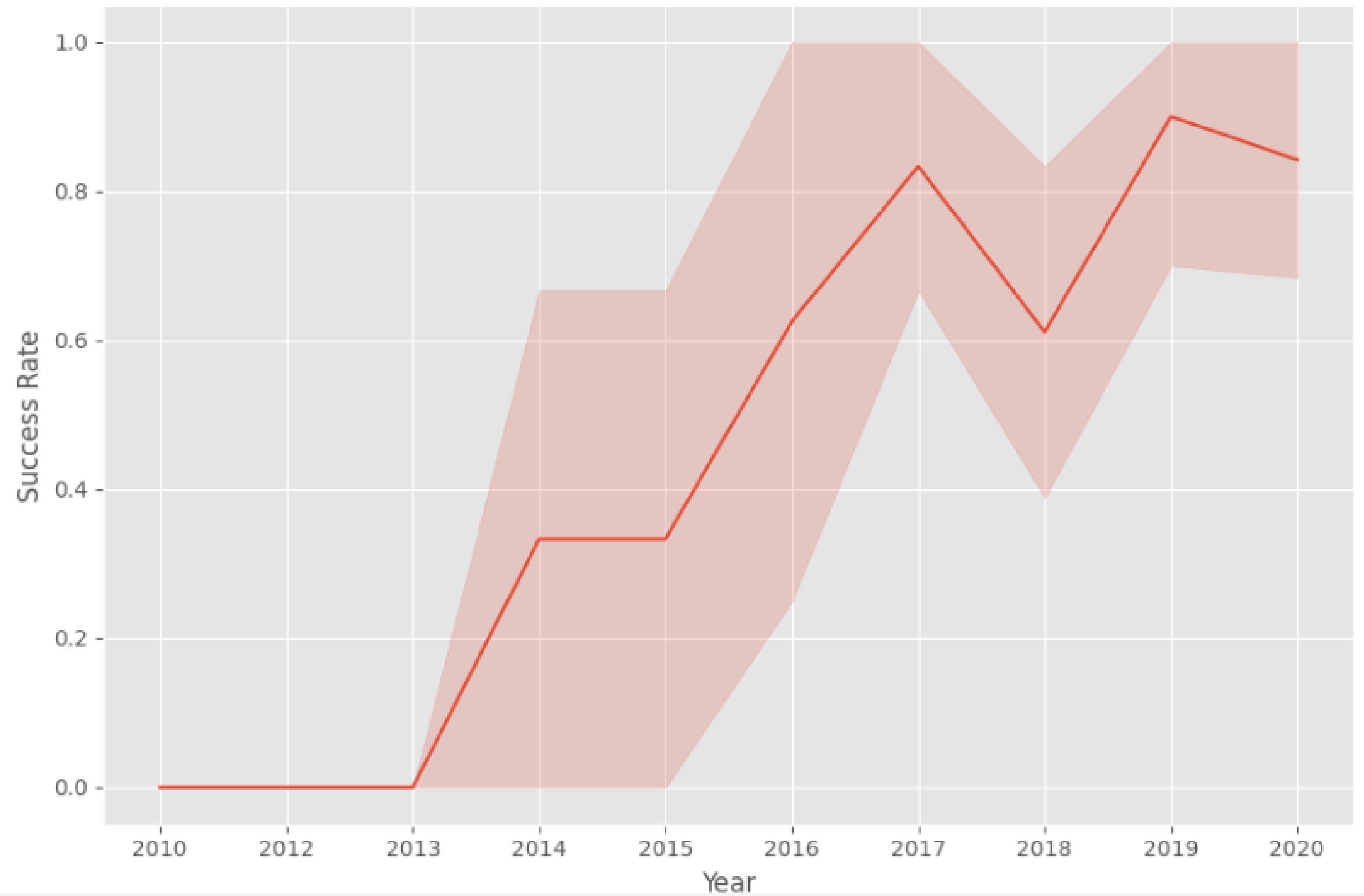
Orange dot – unsuccessful landing
Blue dot – successful landing

For a payload greater than 8000 kg, almost all landings are successful.

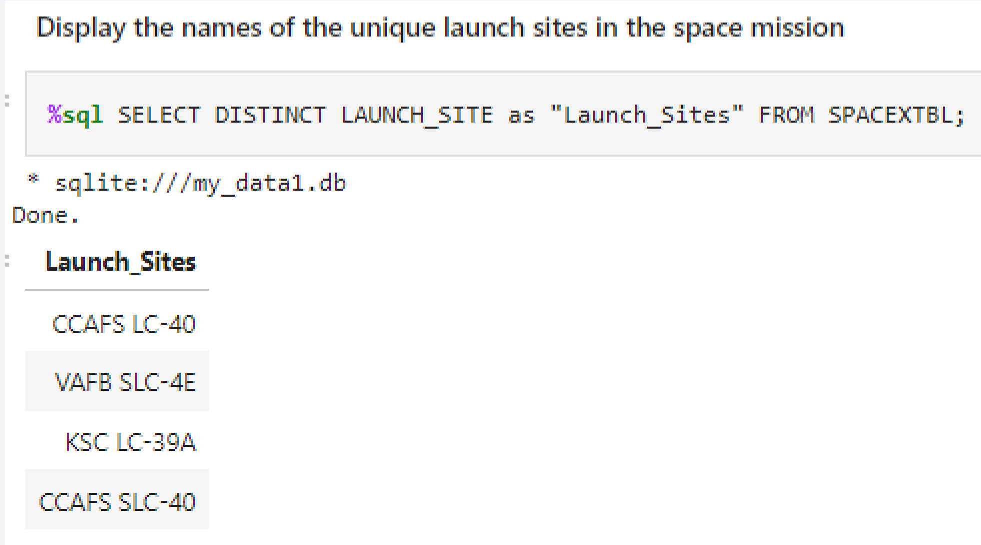


Launch Success Yearly Trend

We see an increasing trend successfully landing since 2013 with a decrease in 2017 - 2018 and 2019 – 2020 years



All Launch Site Names



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

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

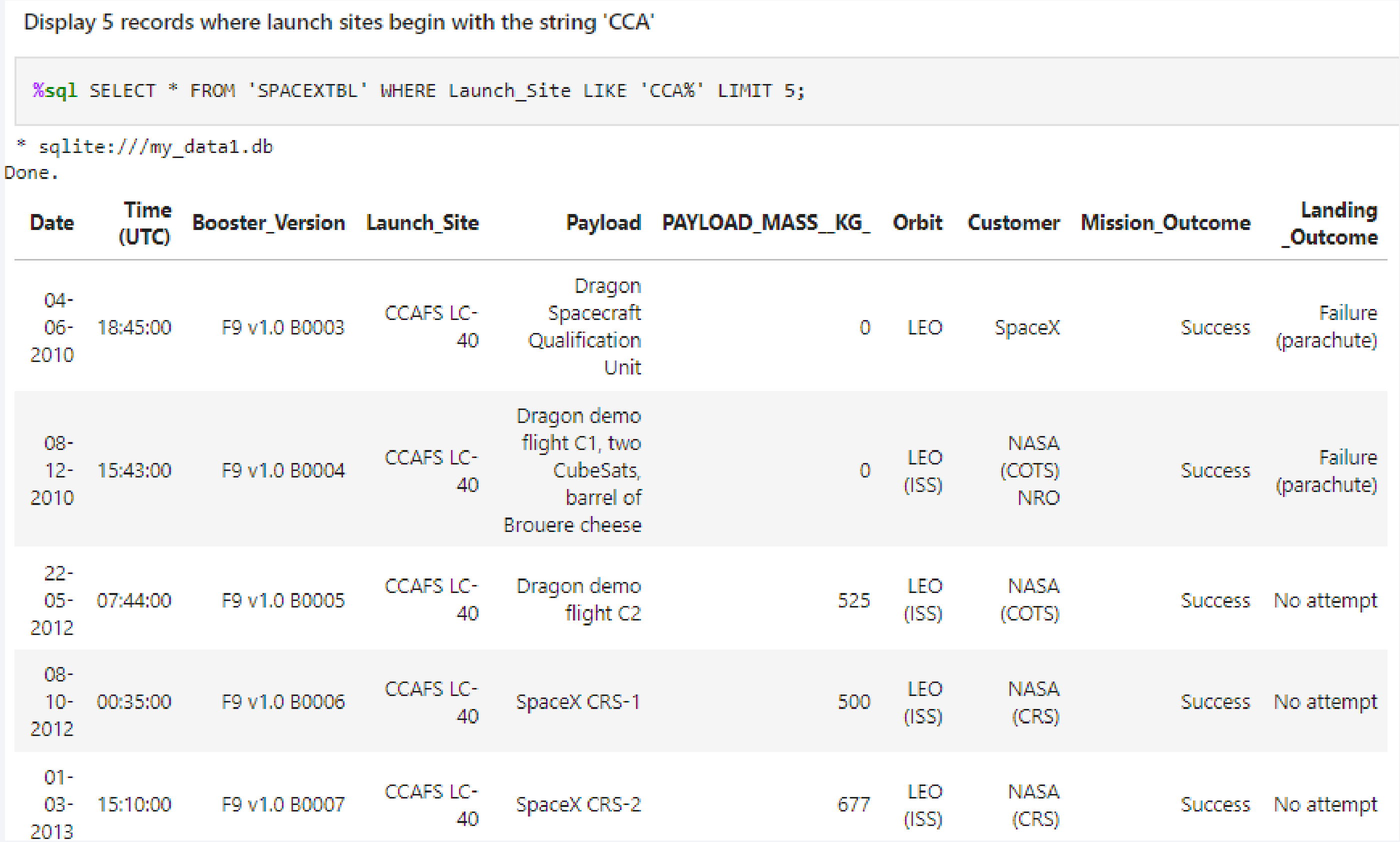
```
 * sqlite:///my_data1.db
Done.
```

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

Used SELECT DISTINCT statement to return only the unique launch sites from the LAUNCH_SITE column of the SPACEXTBL table

Launch Site Names Begin with 'CCA'

Used LIKE command started with % wildcard in WHERE clause to select and display a table of all records where launch sites begin with the string 'CCA'



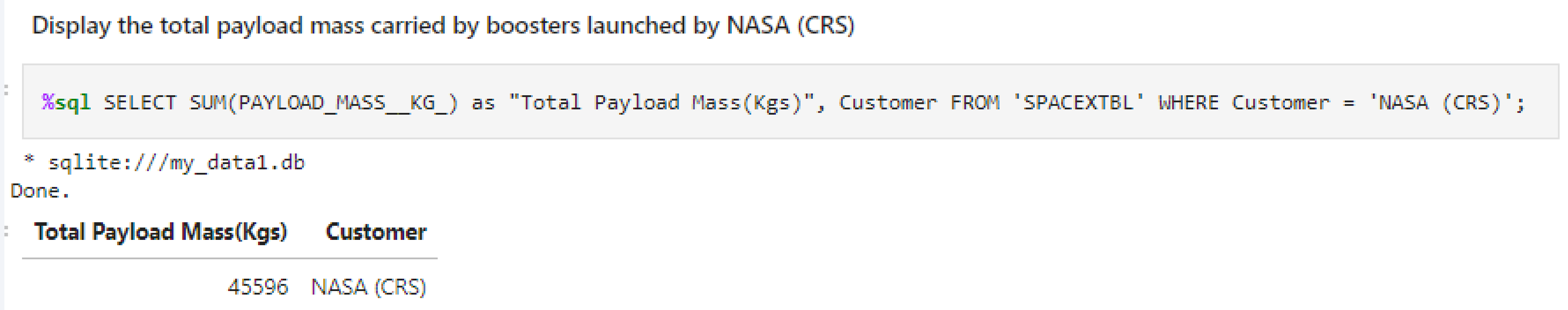
Display 5 records where launch sites begin with the string 'CCA'

```
%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

Total Payload Mass



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

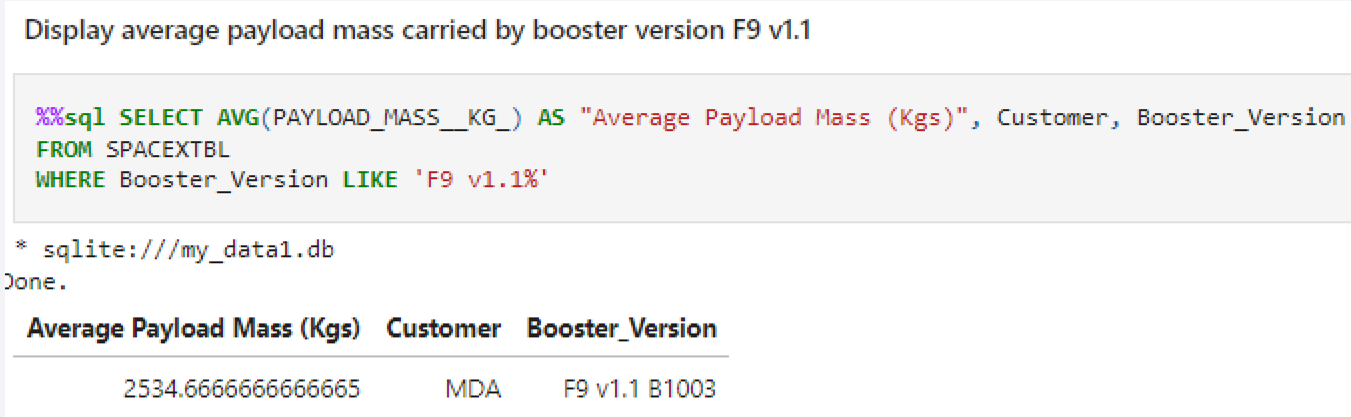
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Used the SUM() function to return and display the total sum of PAYLOAD_MASS_KG column for customer '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 (Kgs)", Customer, Booster_Version
FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1%'
```

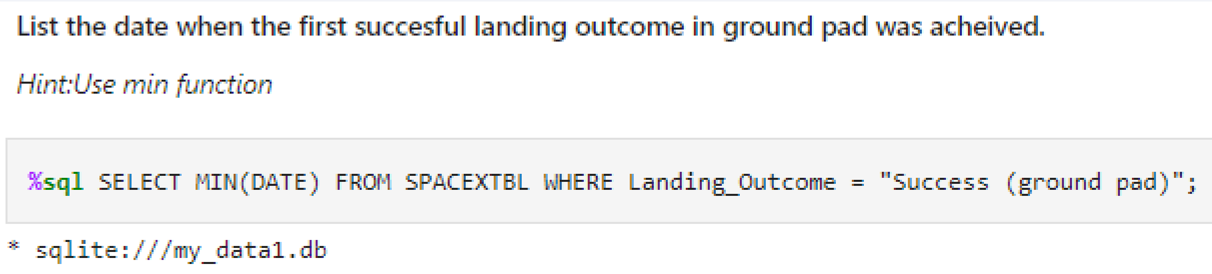
```
 * sqlite:///my_data1.db
Done.
```

Average Payload Mass (Kgs)	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

- The AVG() function was used to the calculate the average payload the average payload mass carried by booster version name started 'F9 v1.1'

- The WHERE clause was used to filter results so that the calculations were only performed on booster_versions only if they were name started '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

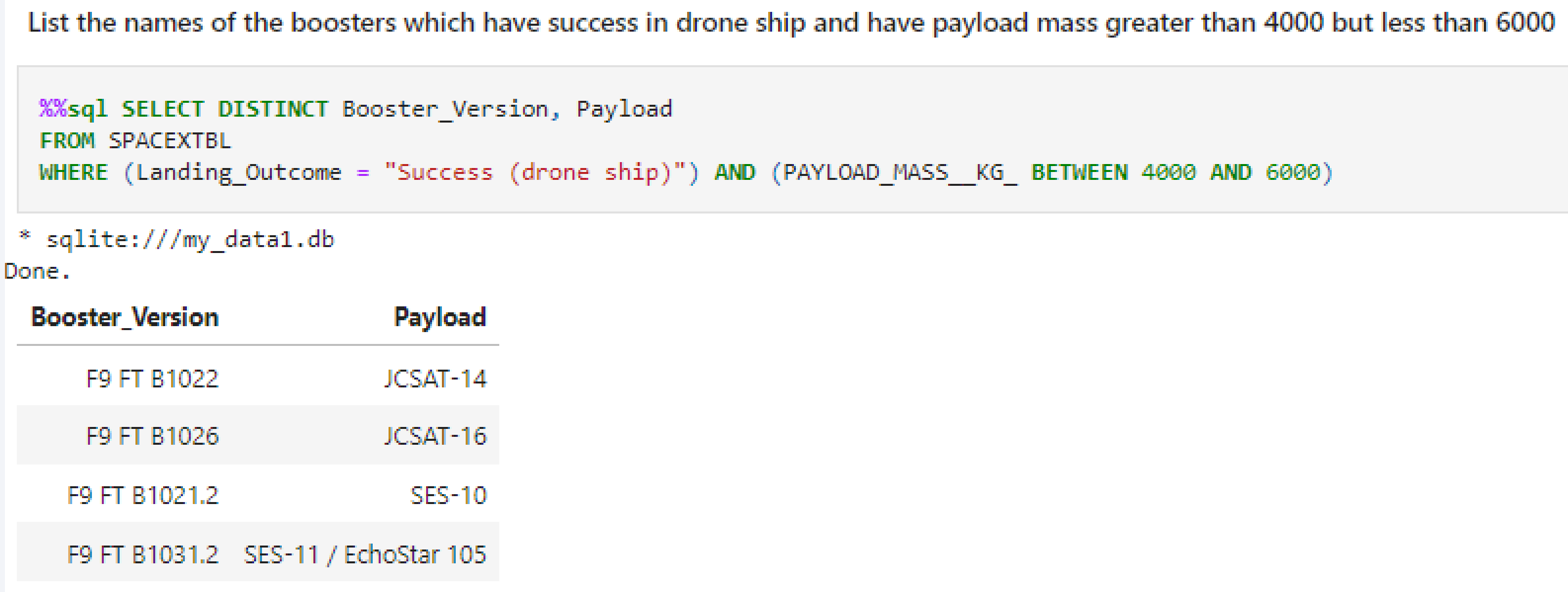
```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome = "Success (ground pad)";
```

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

- The MIN(DATE) function was used to find the date of the first successful landing outcome on ground pad

- The WHERE clause ensured that the results were filtered to match only when the Landing_Outcome column is 'Success (ground pad)'

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

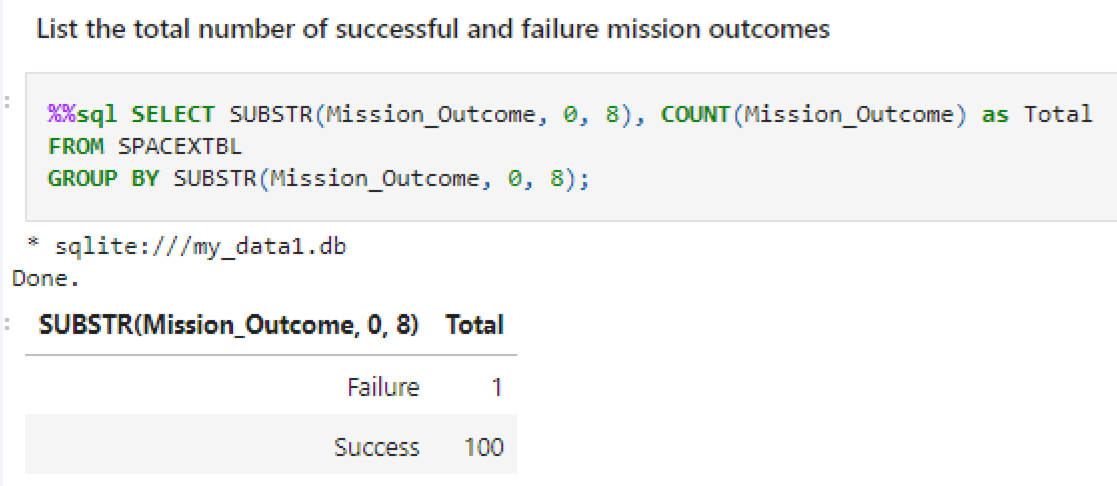
```
%%sql SELECT DISTINCT Booster_Version, Payload
FROM SPACEXTBL
WHERE (Landing_Outcome = "Success (drone ship)") AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

- The BETWEEN clause was used to retrieve only those results of payload mass greater than 4000 but less than 6000
- The WHERE clause filtered the results to include only boosters which successfully landed on drone ship

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
%%sql SELECT SUBSTR(Mission_Outcome, 0, 8), COUNT(Mission_Outcome) as Total
FROM SPACEXTBL
GROUP BY SUBSTR(Mission_Outcome, 0, 8);
```

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

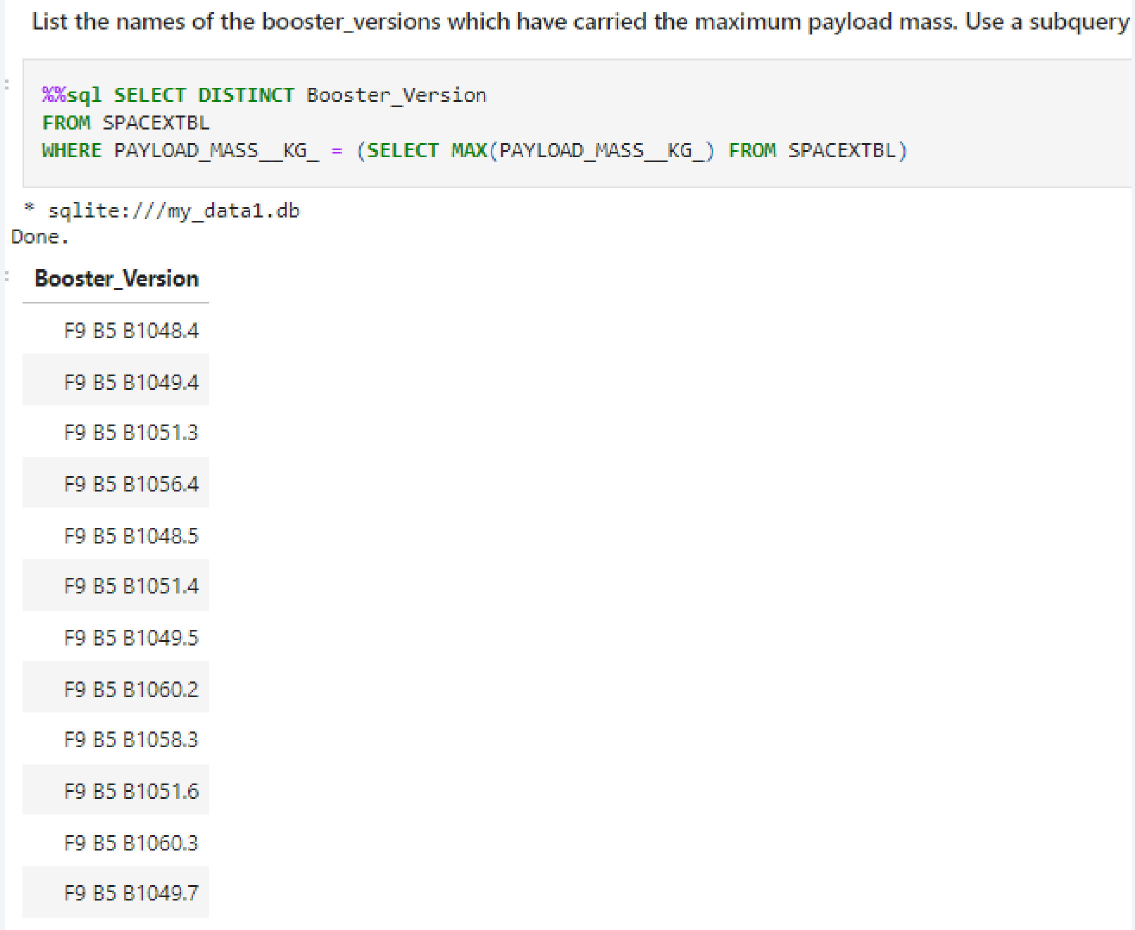
SUBSTR(Mission_Outcome, 0, 8)	Total
Failure	1
Success	100

- The SUBSTR() function is used to group by 'Success' or 'Failure'.

- The COUNT() function is used to count the number of occurences of different mission outcomes with the help of the GROUPBY clause applied to the Mission_Outcome column. A list of the total number of successful and failure mission outcomes is returned. There have been 100 successful mission outcomes out of 101 missions.

Boosters Carried Maximum Payload

The MAX() function was used in a subquery to retrieve a list of boosters which have carried the maximum payload mass



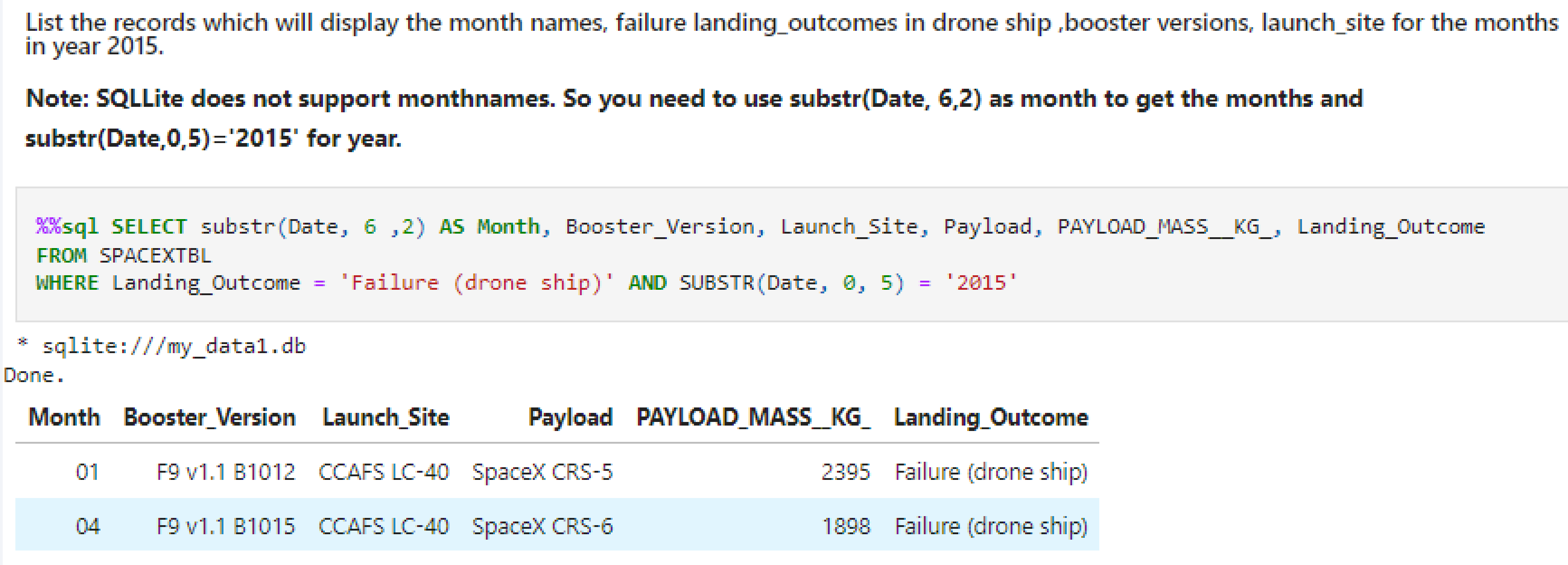
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

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

```
%%sql SELECT substr(Date, 6 ,2) AS Month, Booster_Version, Launch_Site, Payload, PAYLOAD_MASS__KG_, Landing_Outcome
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)' AND SUBSTR(Date, 0, 5) = '2015'
```

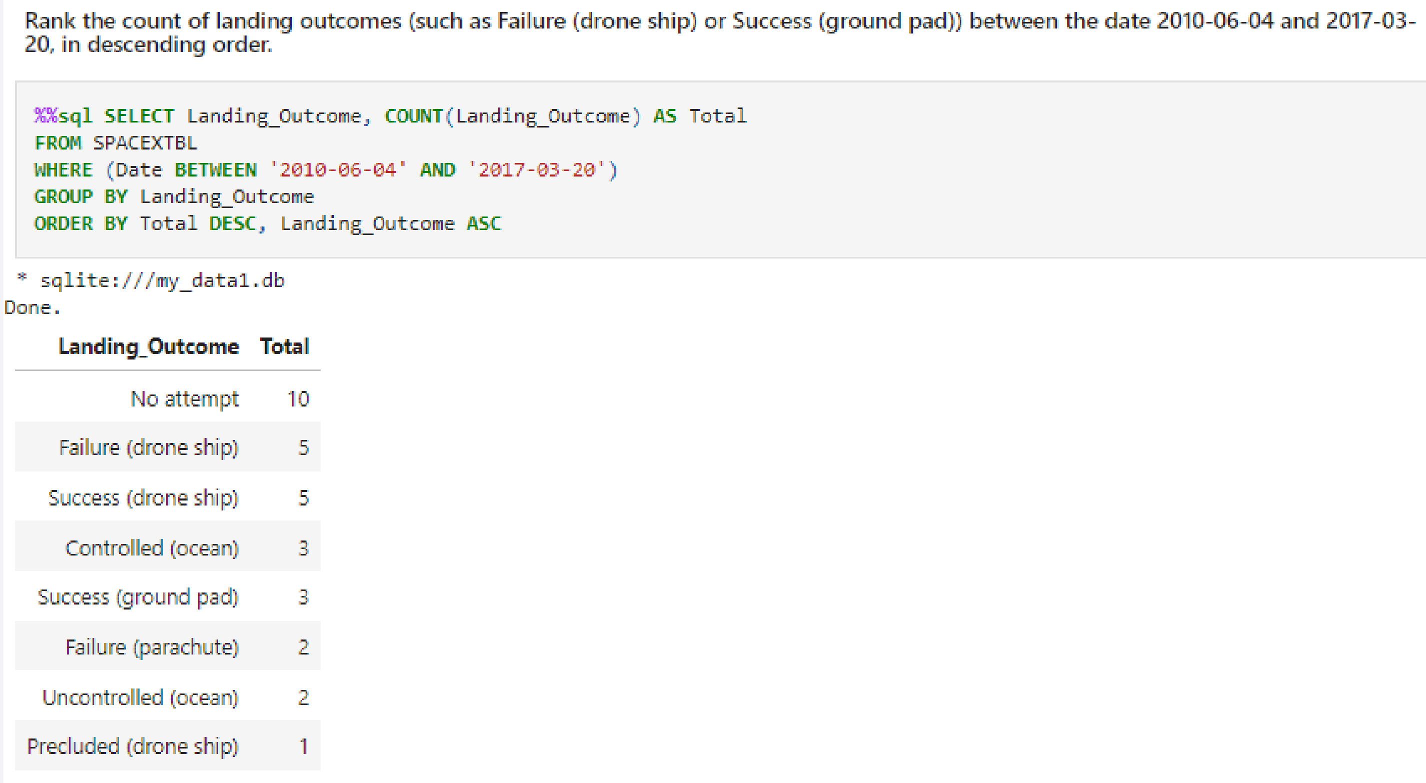
```
 * sqlite:///my_data1.db
Done.
```

Month	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Failure (drone ship)

- The SUBSTR() function is used to extract month and year from date
- WHERE Landing_Outcome = 'Failure (drone ship)' is used to select failure landing in drone ship

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

- The COUNT() function was used to count the different landing outcomes.
- The WHERE and BETWEEN clauses filtered the results to only include results between 2010-06-04 and 2017-03-20.
- The GROUP BY clause ensure that the counts were grouped by their outcome.
- The ORDERBY and DESC clauses were used to sort the results by descending order.



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(Landing_Outcome) AS Total
FROM SPACEXTBL
WHERE (Date BETWEEN '2010-06-04' AND '2017-03-20')
GROUP BY Landing_Outcome
ORDER BY Total DESC, Landing_Outcome ASC
```

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

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

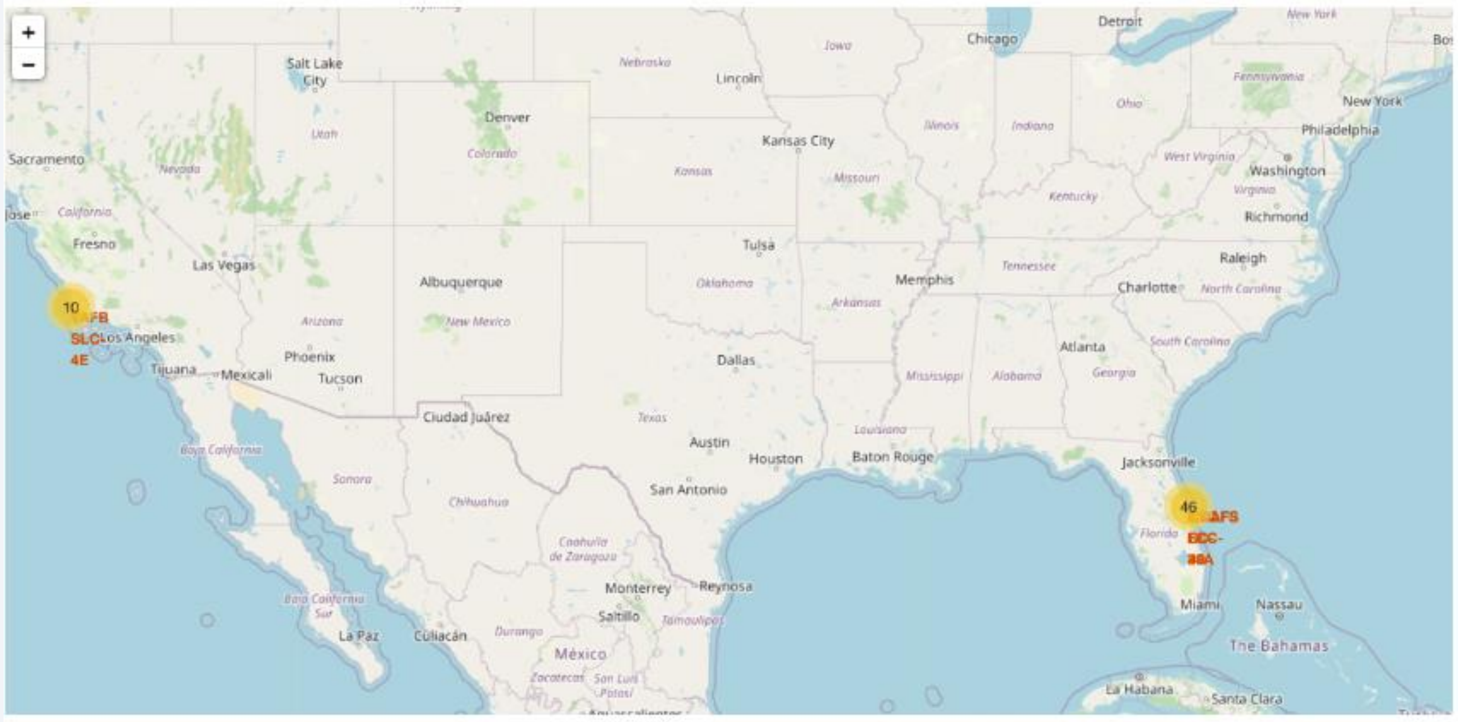
Section 3

Launch Sites Proximities Analysis

SpaceX Launch Sites Locations

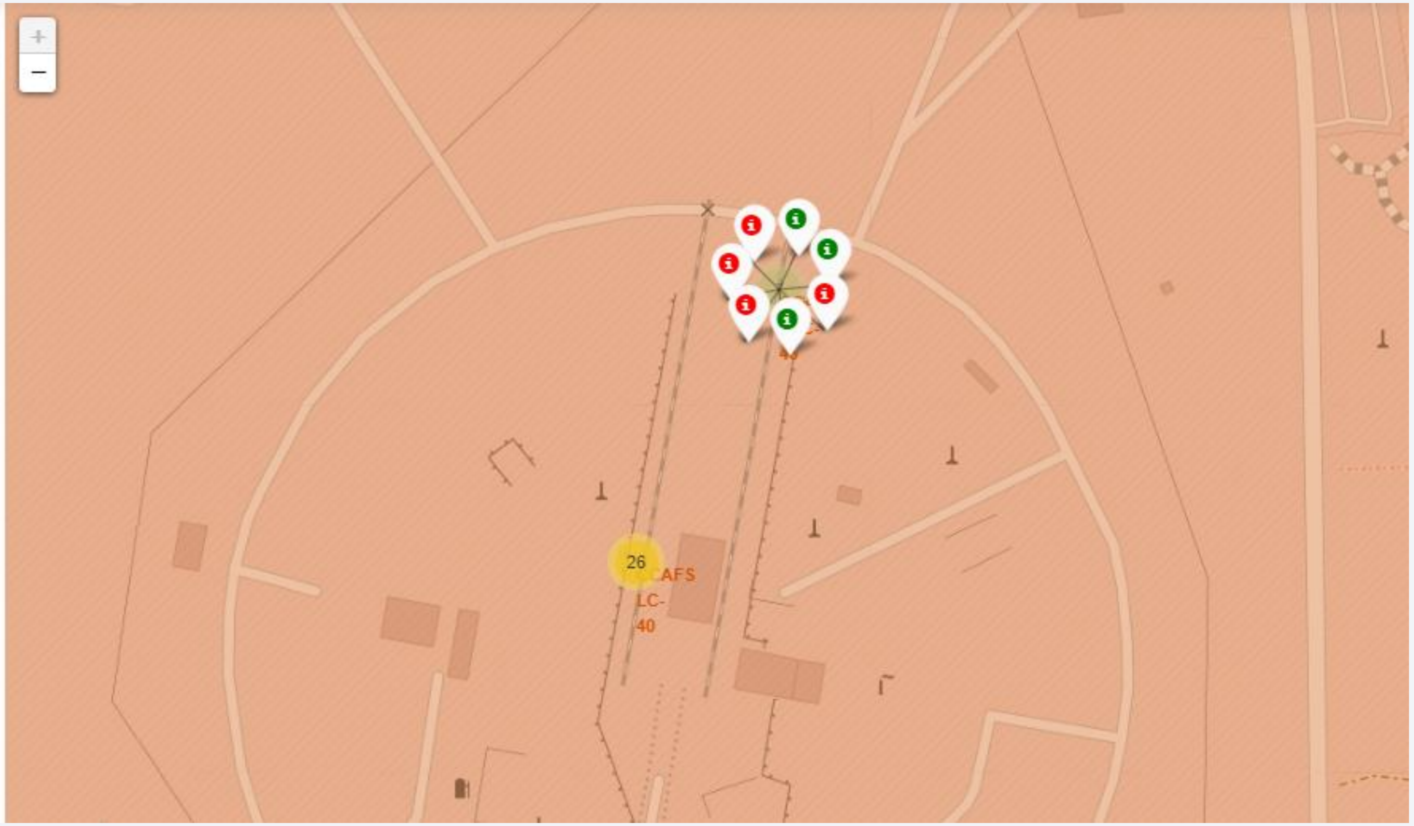
- The yellow markers are indicators of where the locations of all the SpaceX launch sites are situated in the US.

- All launch sites have been strategically placed near the coast and equator.



Success or Failure?

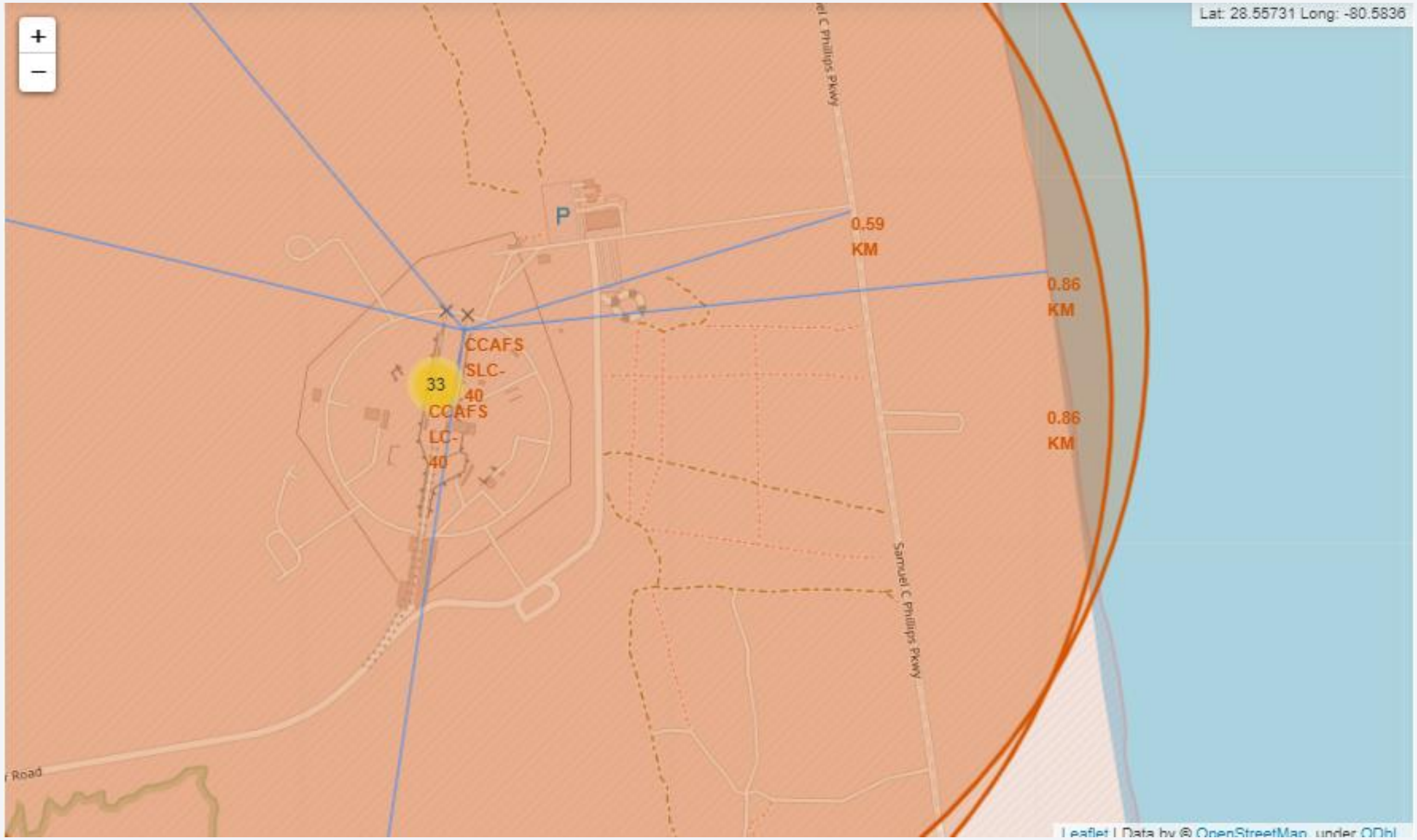
When we zoom in on a launch site and click on the launch site we can display marker clusters of successful landings (green) or failed landing (red).

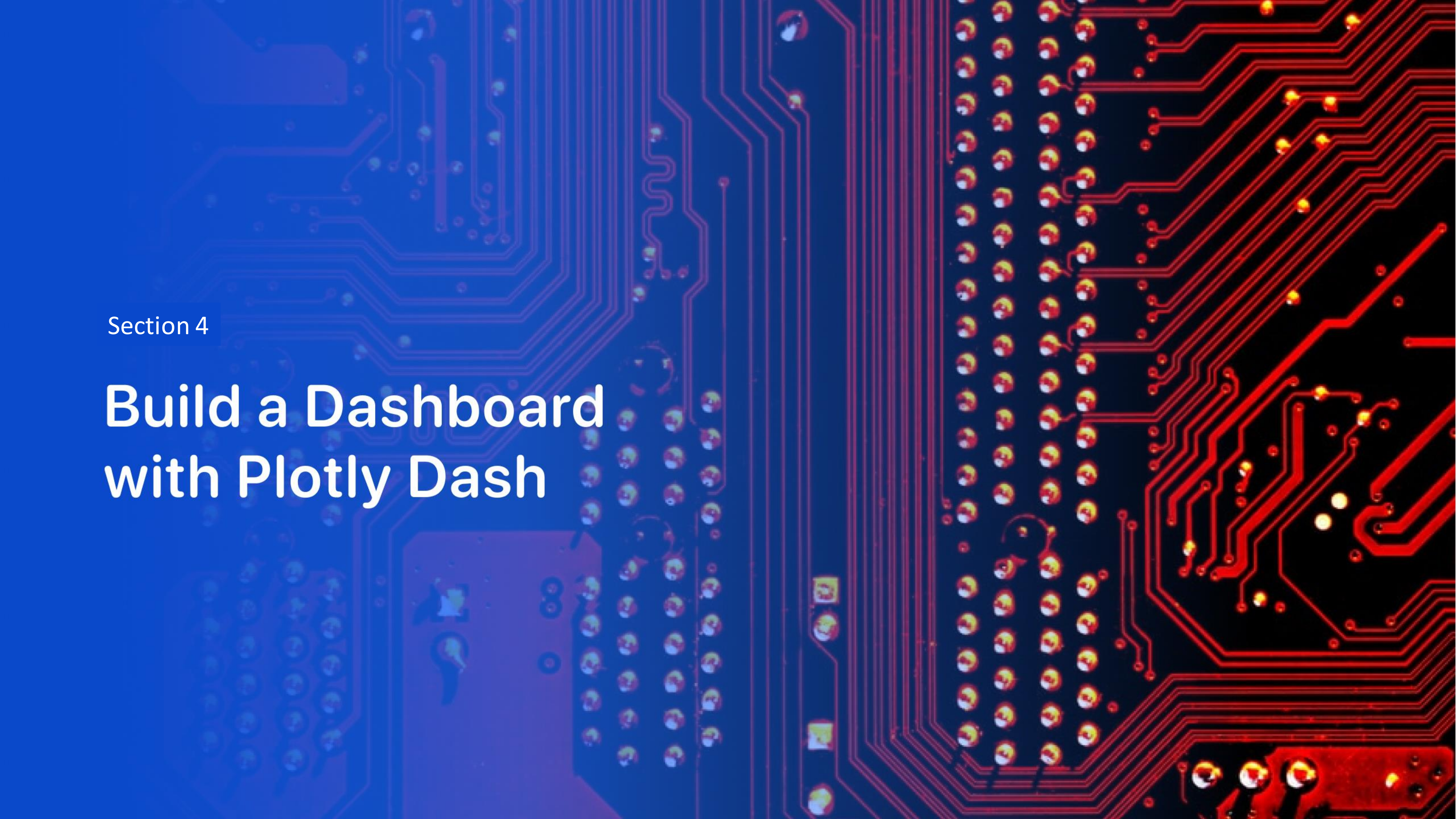


Launch Site Proximities

The generated map shows that the selected launch site:

- is close to a highway for transportation of personnel and equipment
- Is close to the coastlines for launch failure testing.
- maintain a certain distance from the cities (see notebook)

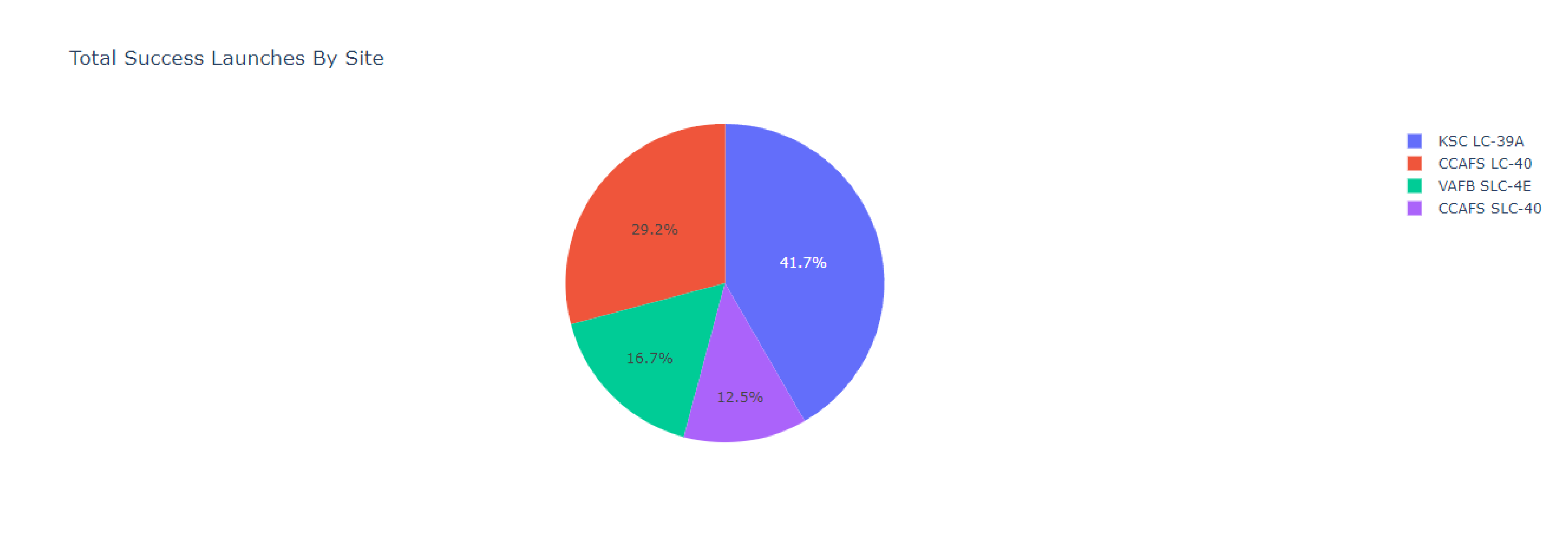




Section 4

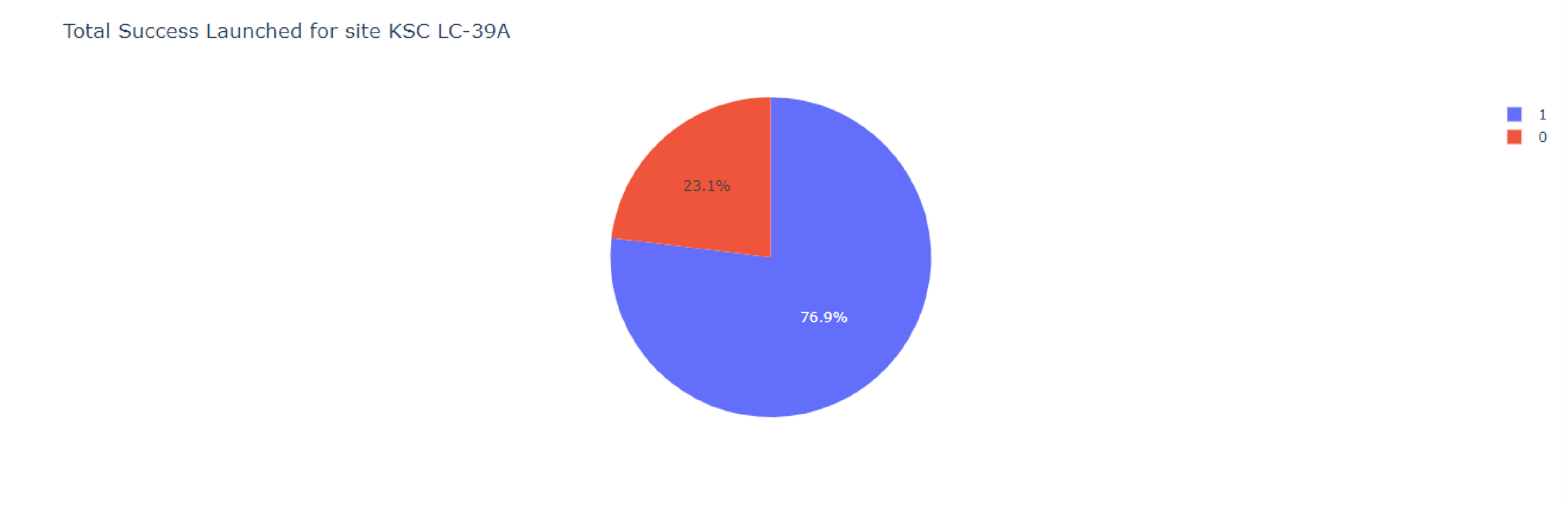
Build a Dashboard with Plotly Dash

Total Successful Launches By Site



- Launch site KSC LC-39A has the highest launch success rate at 42 %.
- Launch site SLC-40 has the lowest success rate at 13 %.

Launch Site With Highest Success Ratio

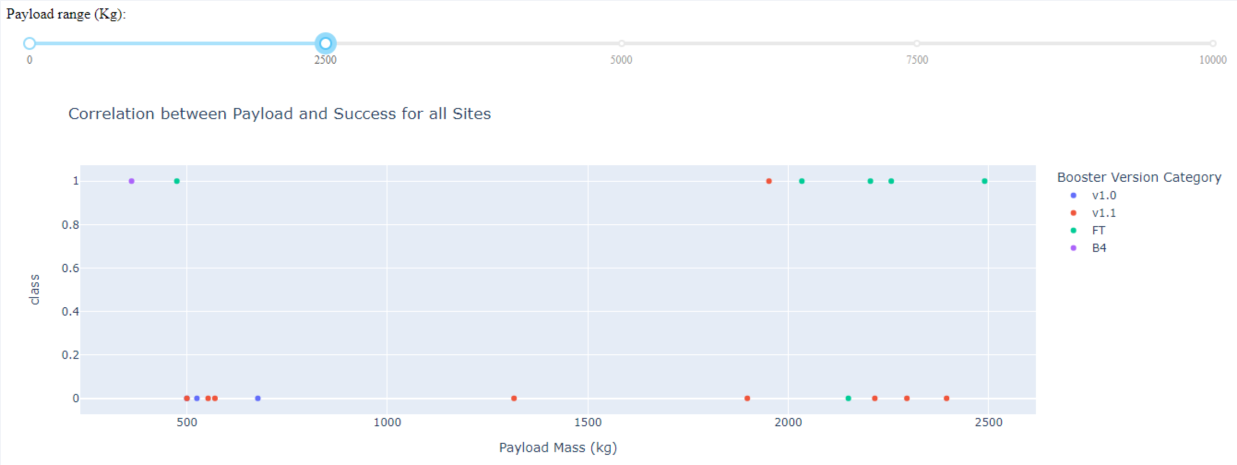


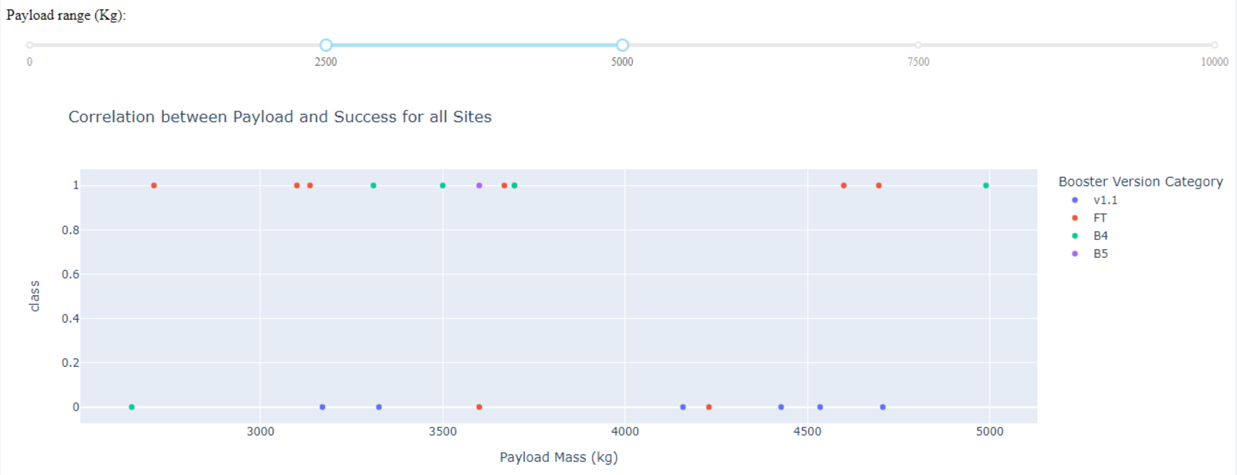
The KSLC-39A has the highest success rate with 76.9 %.

Payloads vs Launch Outcome

- The launch success rate for payloads 0 - 2500 kg is slightly lower than that of payloads 2500-5000 kg. There is in fact not much difference between the two.

- The booster version that has the largest success rate, in both weight ranges is the v1.1.

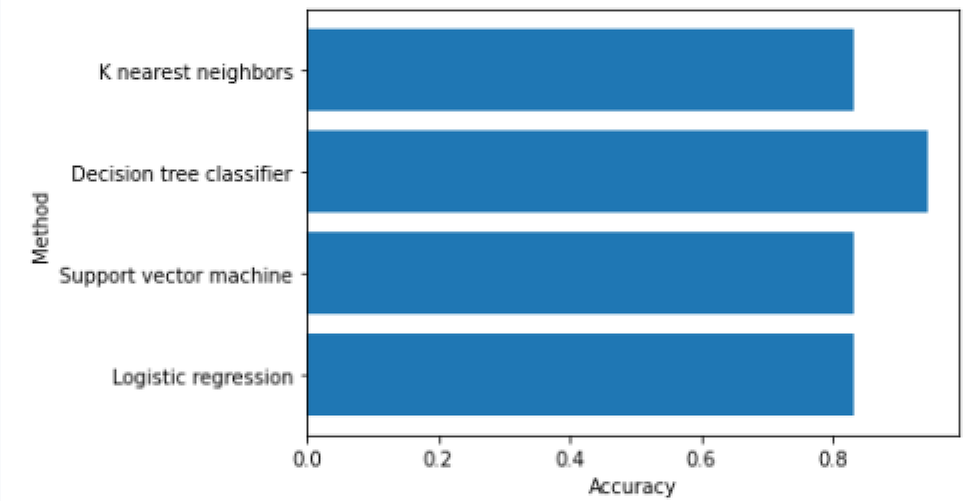


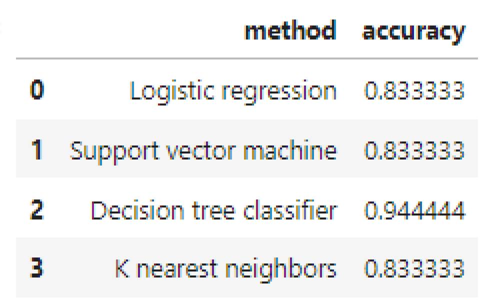


Section 5

Predictive Analysis (Classification)

Classification Accuracy



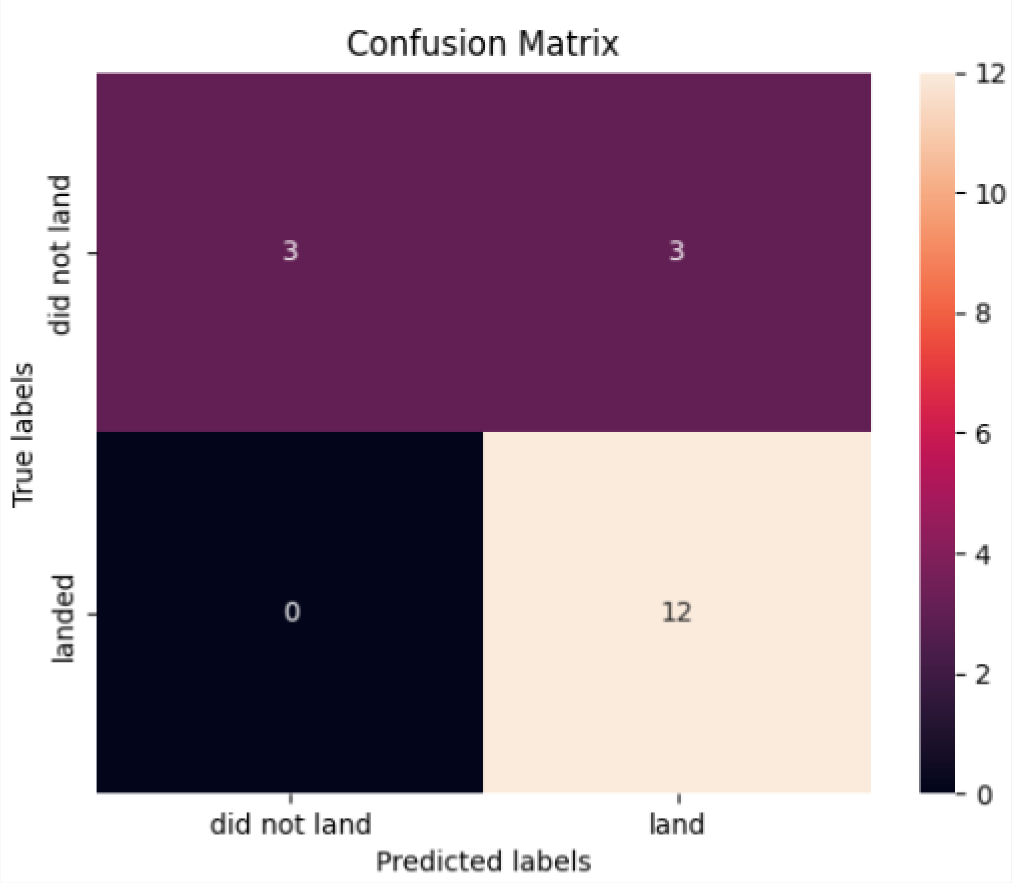


	method	accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.944444
3	K nearest neighbors	0.833333

The Decision Tree classifier had the best accuracy at 94 %.

Confusion Matrix

- All the 4 classification models had the same confusion matrixes.
- The major problem is false positives for all models.



Conclusions

- The analysis showed that there is a positive correlation between number of flights and success rate as the success rate has improved over the years.
- There are certain orbits like SSO, HEO, GEO, and ES-L1 where launches were the most successful. The success rate for the VAFB SLC 4E launch site is 100 % after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100 % success rates after 80th flight.
- Success rate can be linked to payload mass as the lighter payloads generally proved to be more successful than the heavier payloads.
- The launch sites are strategically located near highways and railways, but far away from cities for safety.
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100 %, with SO orbit having the lowest success rate at ~50 %. Orbit SO has 0 % success rate.
- The best predictive model to use for this dataset is the Decision Tree Classifier as it had the highest accuracy with 94 %.

Appendix

GitHub Repository: SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction

Thank you!