IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

In Hoi Kim
09/06/2023

Outline

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 rate of $62 million, while other providers charge over $165 million per launch. A significant portion of this cost advantage is attributed to Space X's ability to recycle the initial rocket stage. Consequently, if we can forecast the successful landing of the first stage, we can estimate the launch cost. This data could prove valuable if another company wishes to compete with Space X for a rocket launch contract. The objective of this project is to develop a machine learning pipeline for predicting the successful landing of the first rocket stage.

- Problems you want to find answers

What factors determine if the rocket will land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Describe how data was collected
- Perform data wrangling
    - Describe how data was processed
- 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

- Describe how data sets were collected.

1. Data Collection

1. Utilized **GET requests** to access the **SpaceX API** for retrieval.

2. Data Decoding and Transformation

1. Applied **.json()** function to decode the response content.
2. Converted the decoded data into a **pandas dataframe** using **.json_normalize()**.

3. Data Cleaning

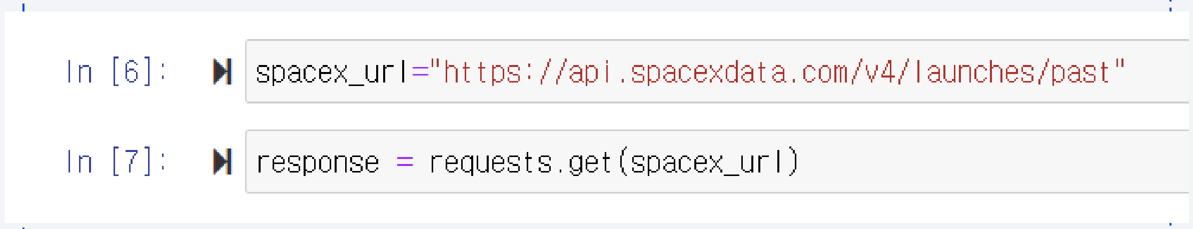
1. Performed data cleaning procedures.
2. Checked for and addressed **missing values**.

4. Web Scraping from Wikipedia

4. Employed **BeautifulSoup** for web scraping.
5. Extracted Falcon 9 launch records from **HTML tables**.
6. Parsed and converted the extracted data into a **pandas dataframe** for further analysis.

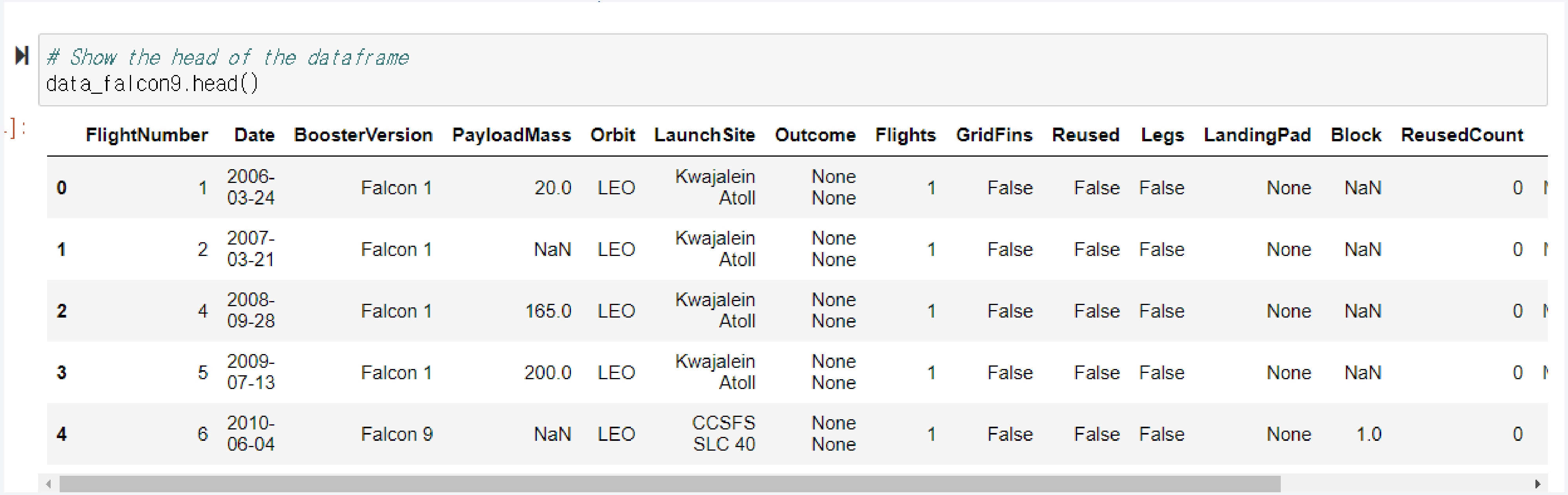
Data Collection – SpaceX API

https://github.com/inhoi/IBM_Capstone/blob/main/Week1_A.ipynb



```
In [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]:  response = requests.get(spacex_url)
```

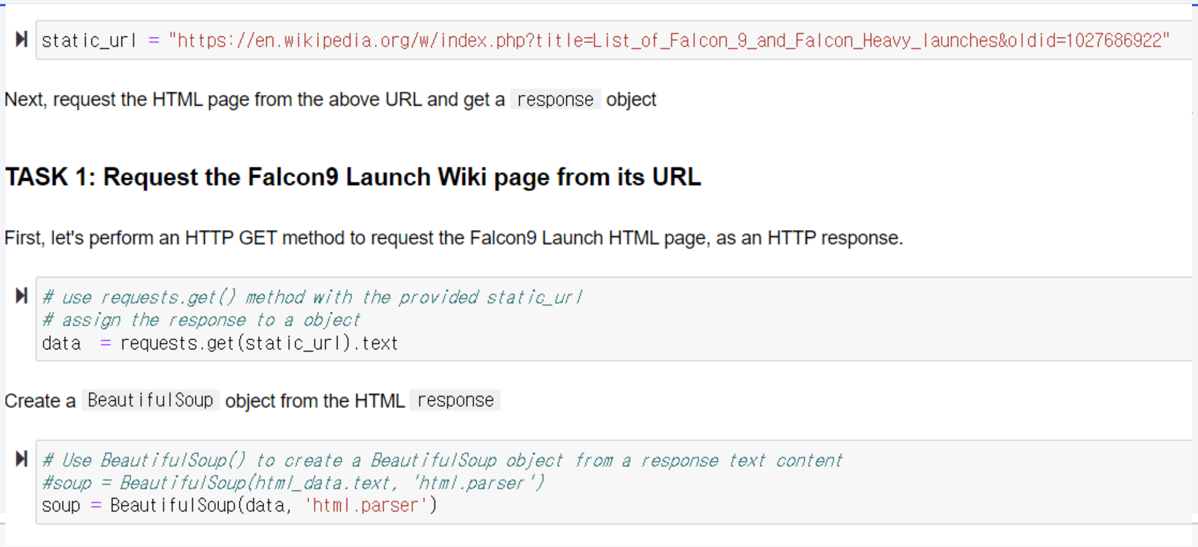


```
# Show the head of the dataframe
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0

Data Collection - Scraping

https://github.com/inhoi/IBM_Capstone/blob/main/Week1_B.ipynb



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

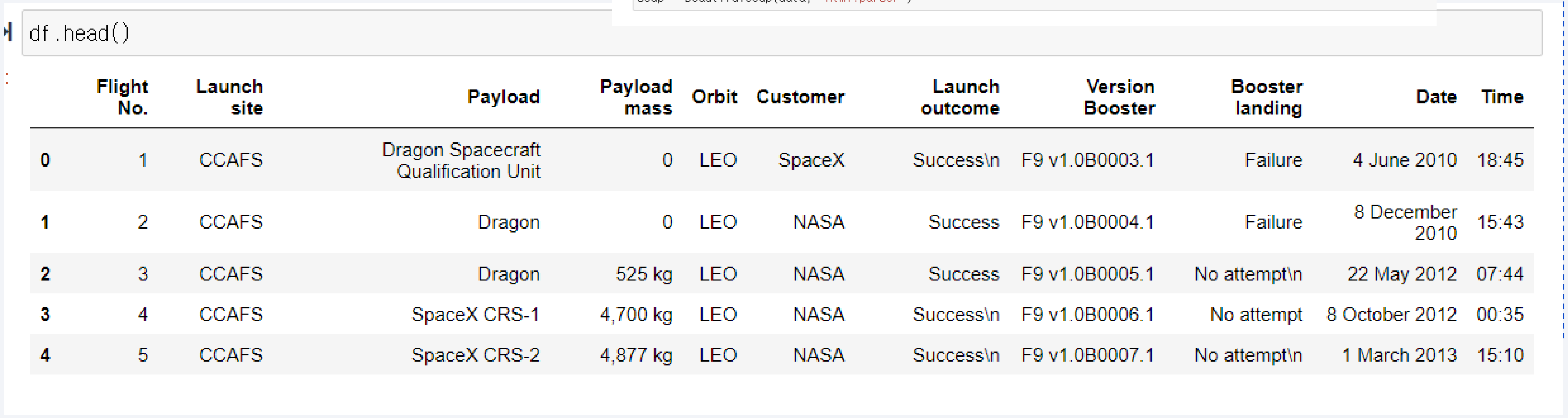
TASK 1: Request the Falcon9 Launch Wiki page from its URL

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
data  = requests.get(static_url).text
```

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

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

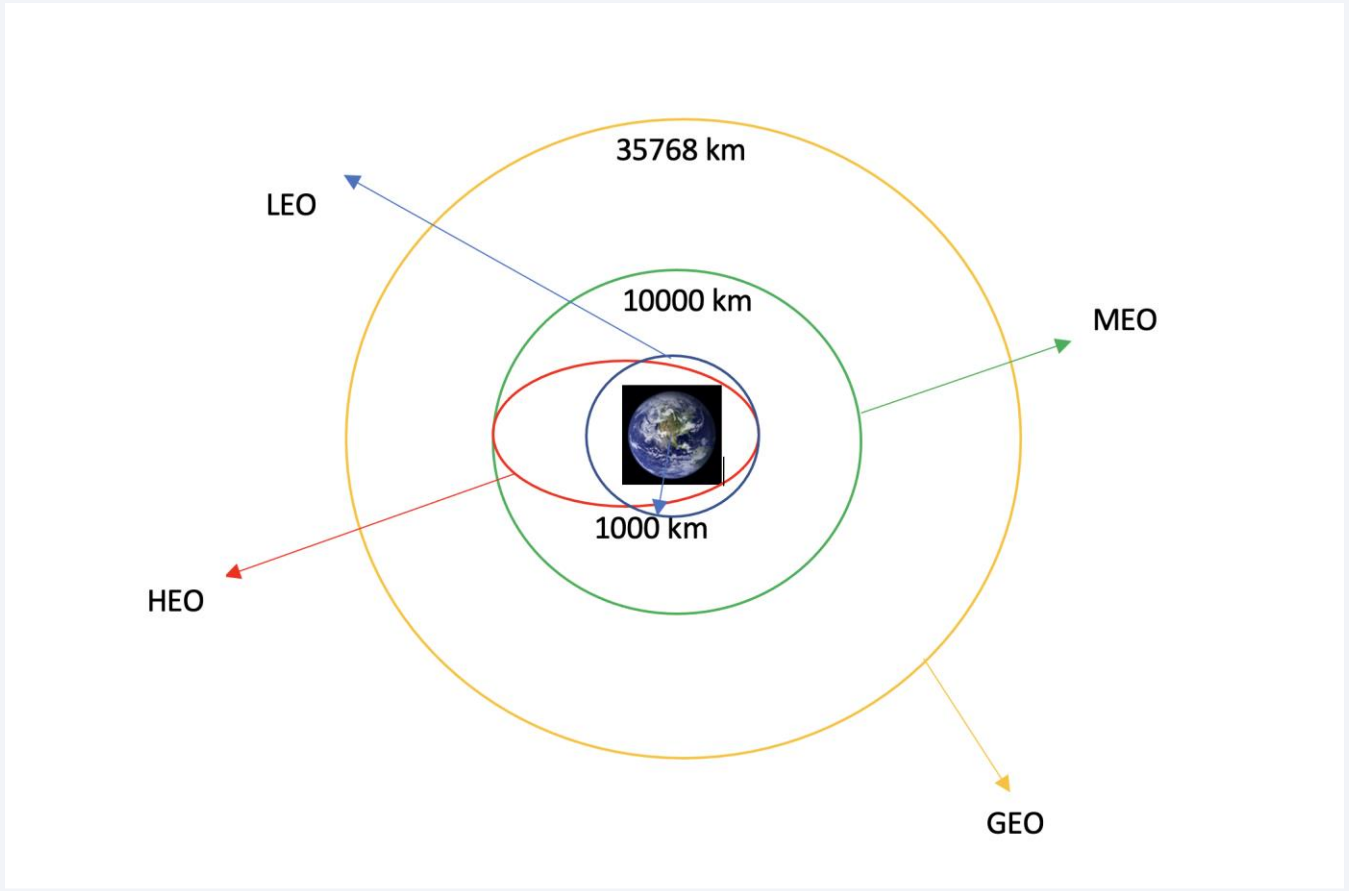


```
df.head()
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

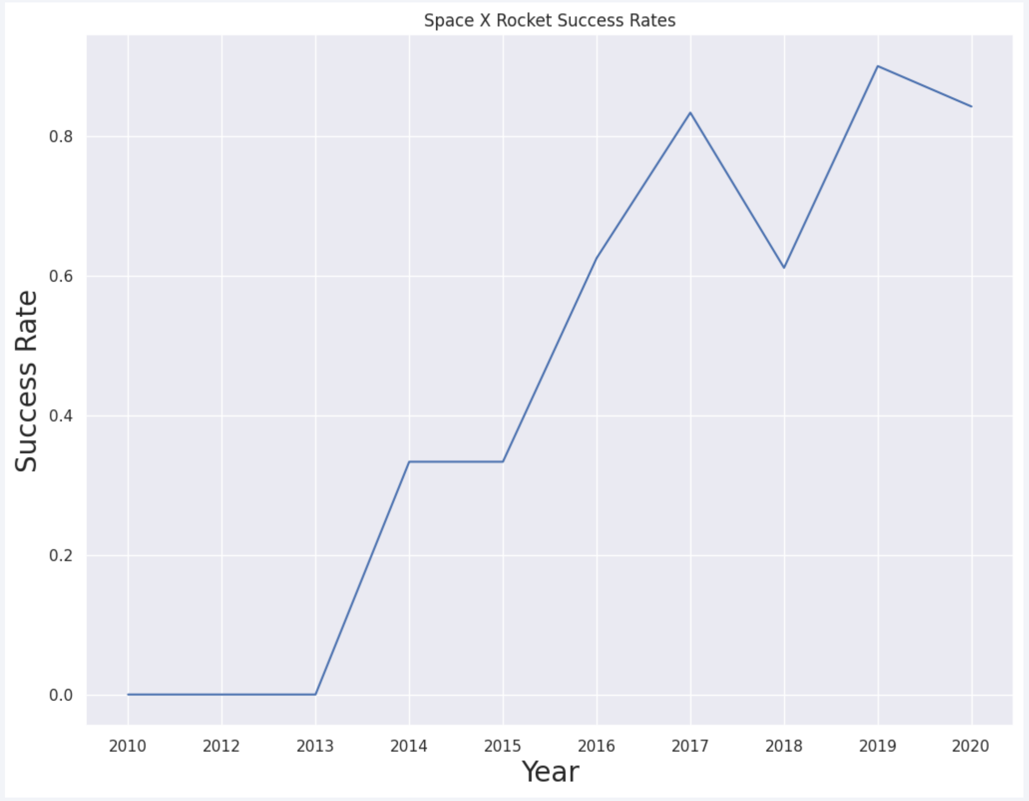
Data Wrangling

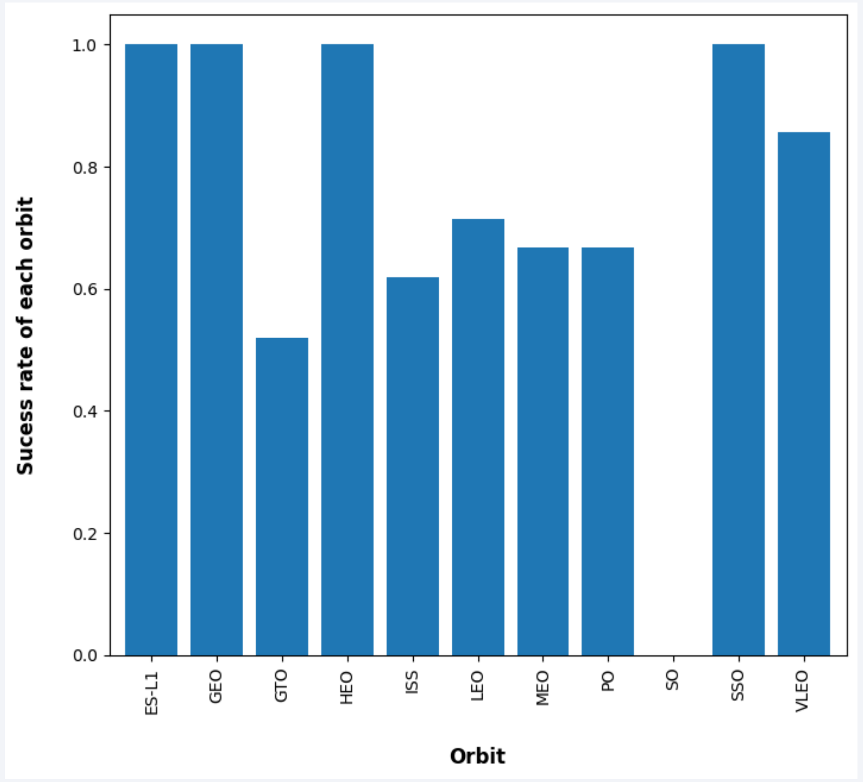
https://github.com/inhoi/IBM_Capstone/blob/main/Week1_C.ipynb



EDA with Data Visualization

Visualizing data by the relationship between flight number and launch site, success rate of each orbit type, flight number and orbit type by yearly trend.
https://github.com/inhoi/IBM_Capstone/blob/main/Week2_B.ipynb





EDA with SQL

- I used Live SQL site to perform this because my SQL database server was having problem, and I couldn’t login. I uploaded all screenshots in Github.

1. The identities of distinct launch locations in the space mission.
2. The cumulative payload mass transported by NASA's CRS-launched boosters.
3. The mean payload mass transported by the F9 v1.1 booster variant.
4. The combined count of both successful and unsuccessful mission results.
5. Instances of unsuccessful landings on drone ships, including their booster versions and launch site designations.

- https://github.com/inhoi/IBM_Capstone/blob/main/Week2_A.ipynb

Build a Dashboard with Plotly Dash

- Build dashboard using Plotly
- Plotting Scatter chart to see relationship between several features
- Plotting Pie chart to see total launches
- https://github.com/inhoi/IBM_Capstone/blob/main/Week3_A.ipynb

Predictive Analysis (Classification)

- Load dataset, standardize, split data to train and test data sets, set parameters for GridsearchCV.
- Check accuracy for each model, do hyperparameter tuning, plot confusion matrix
- Find best classification model
- https://github.com/inhoi/IBM_Capstone/blob/main/Week4.ipynb

Results

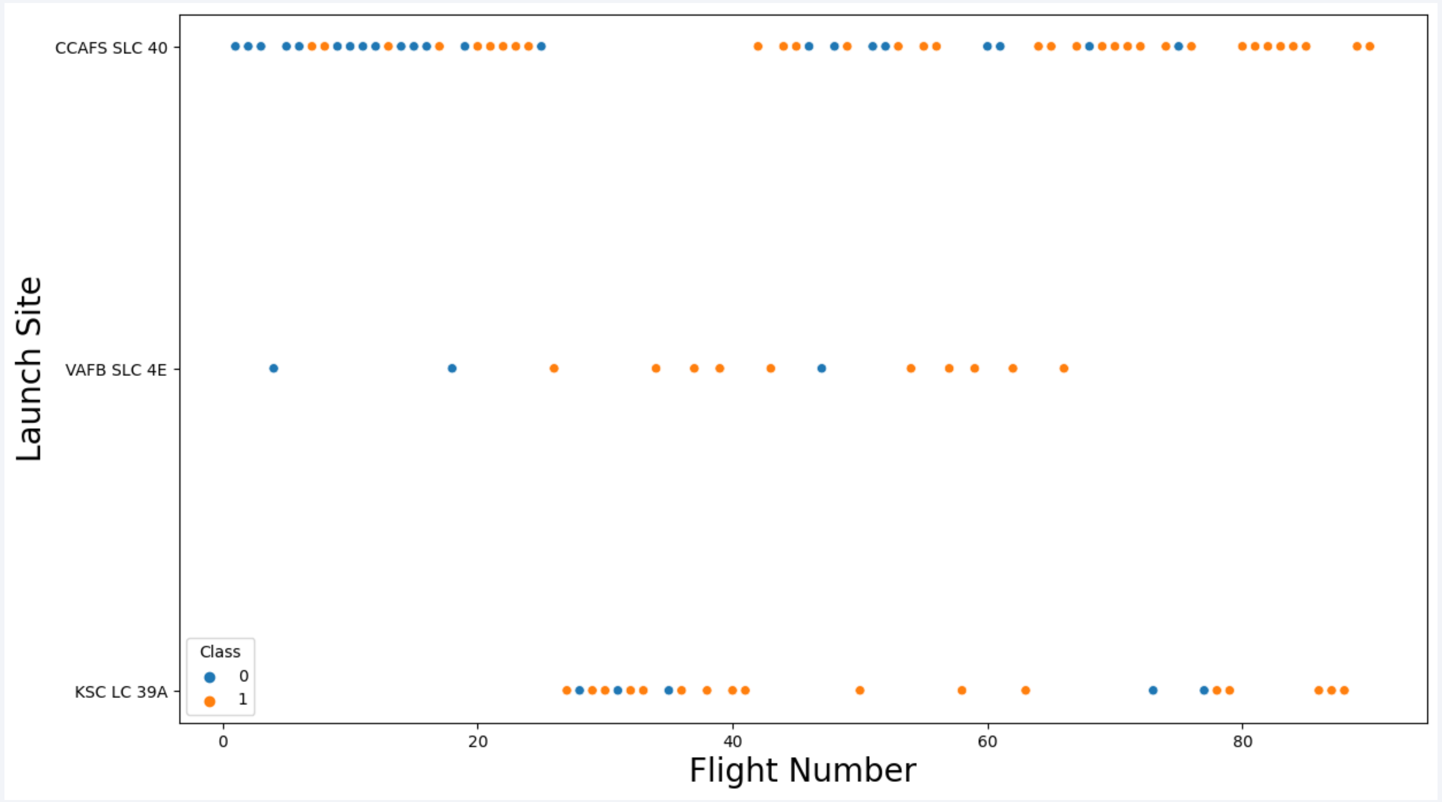
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

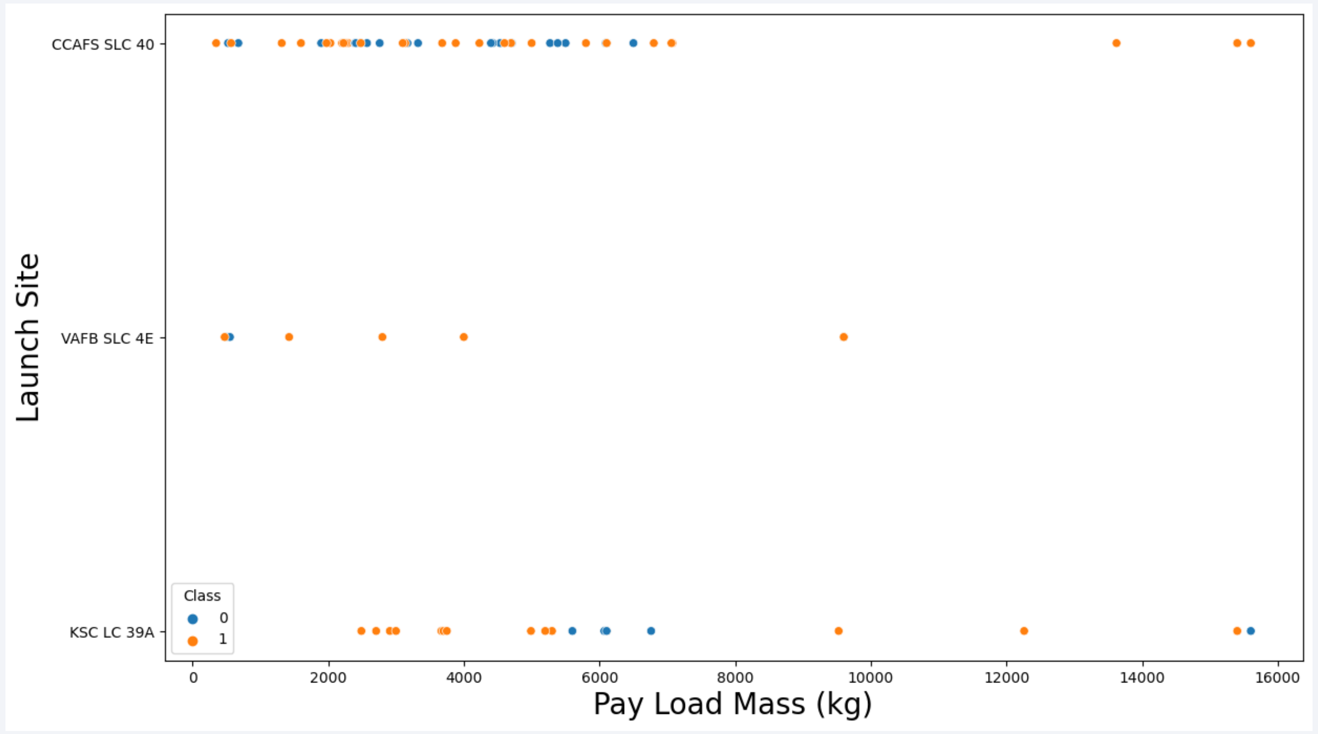
Flight Number vs. Launch Site

- Larger the flight amount shows the greater success rate of launch site



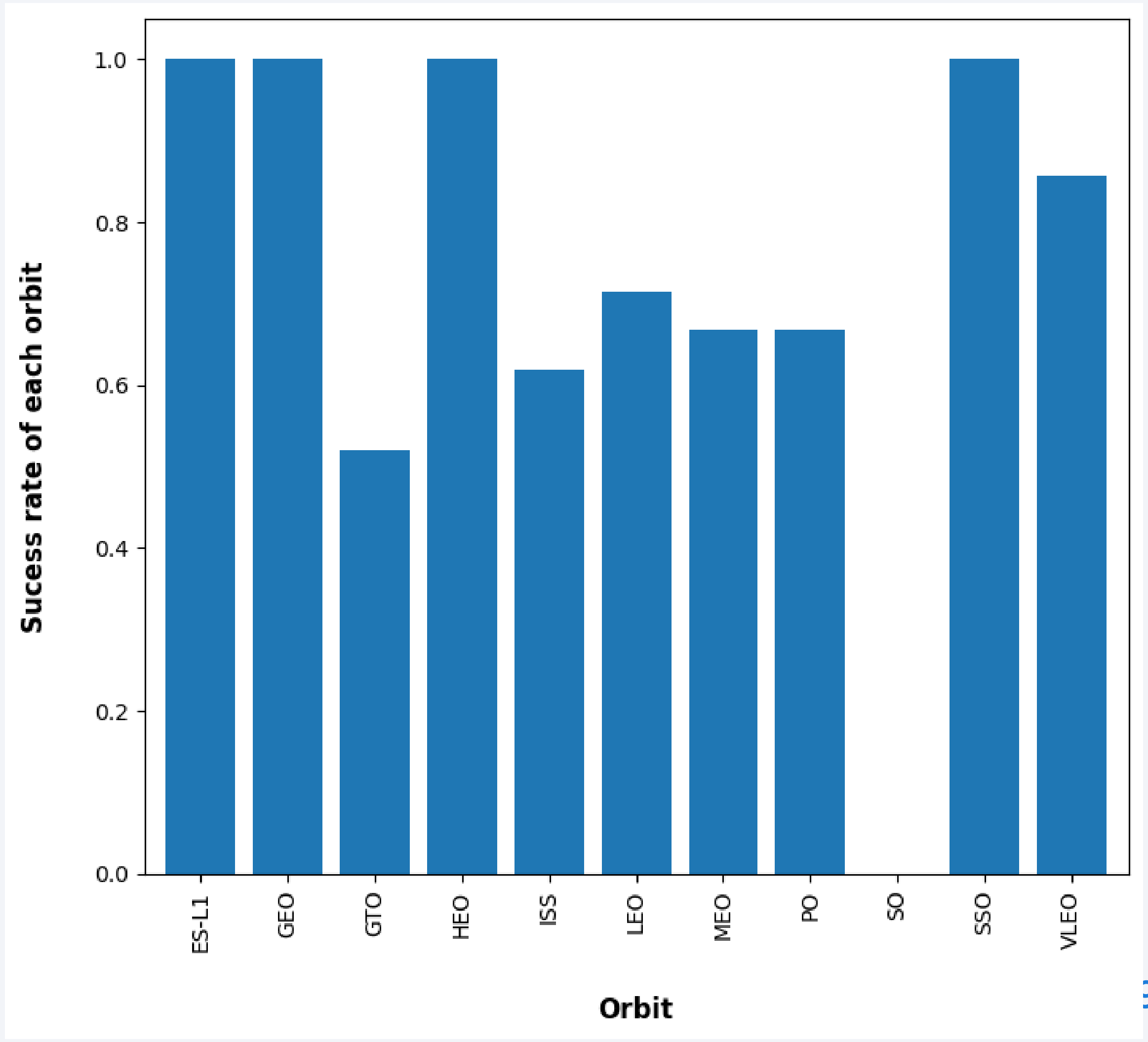
Payload vs. Launch Site

- Greater payload mass shows the higher success rate for launch site



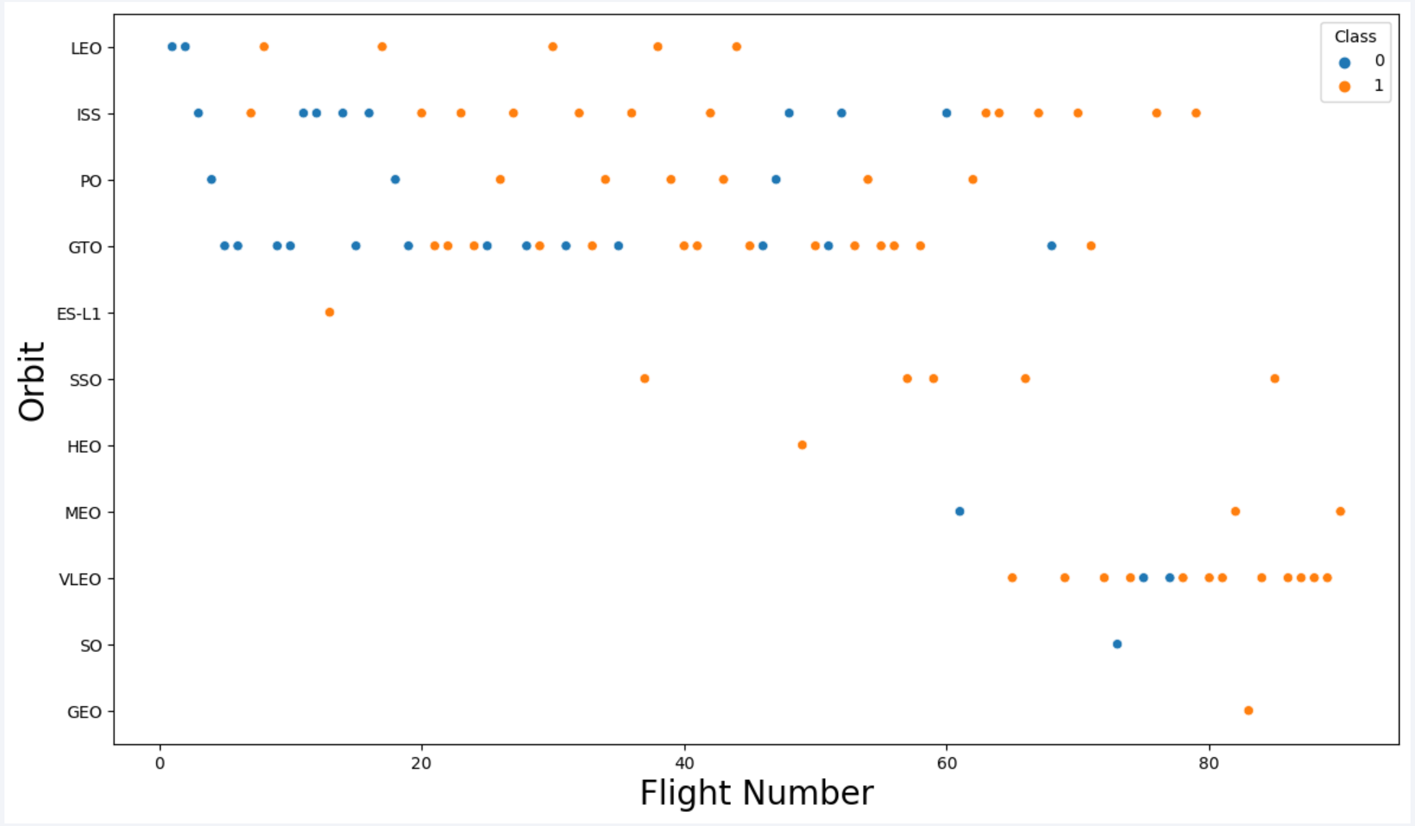
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO shows the most success rate



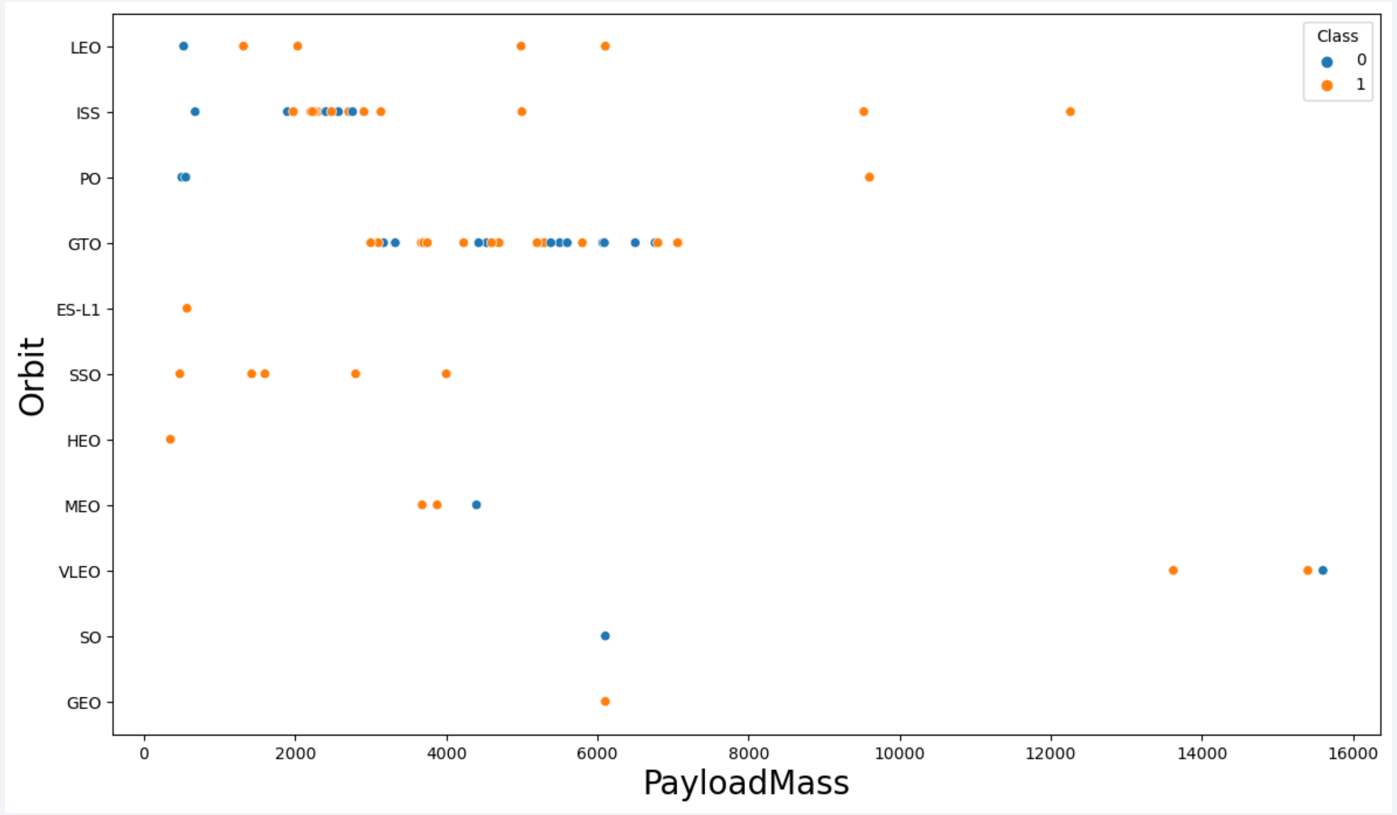
Flight Number vs. Orbit Type

- Seems no relationship between flight number and GTO orbit but LEO orbit shows increasing success



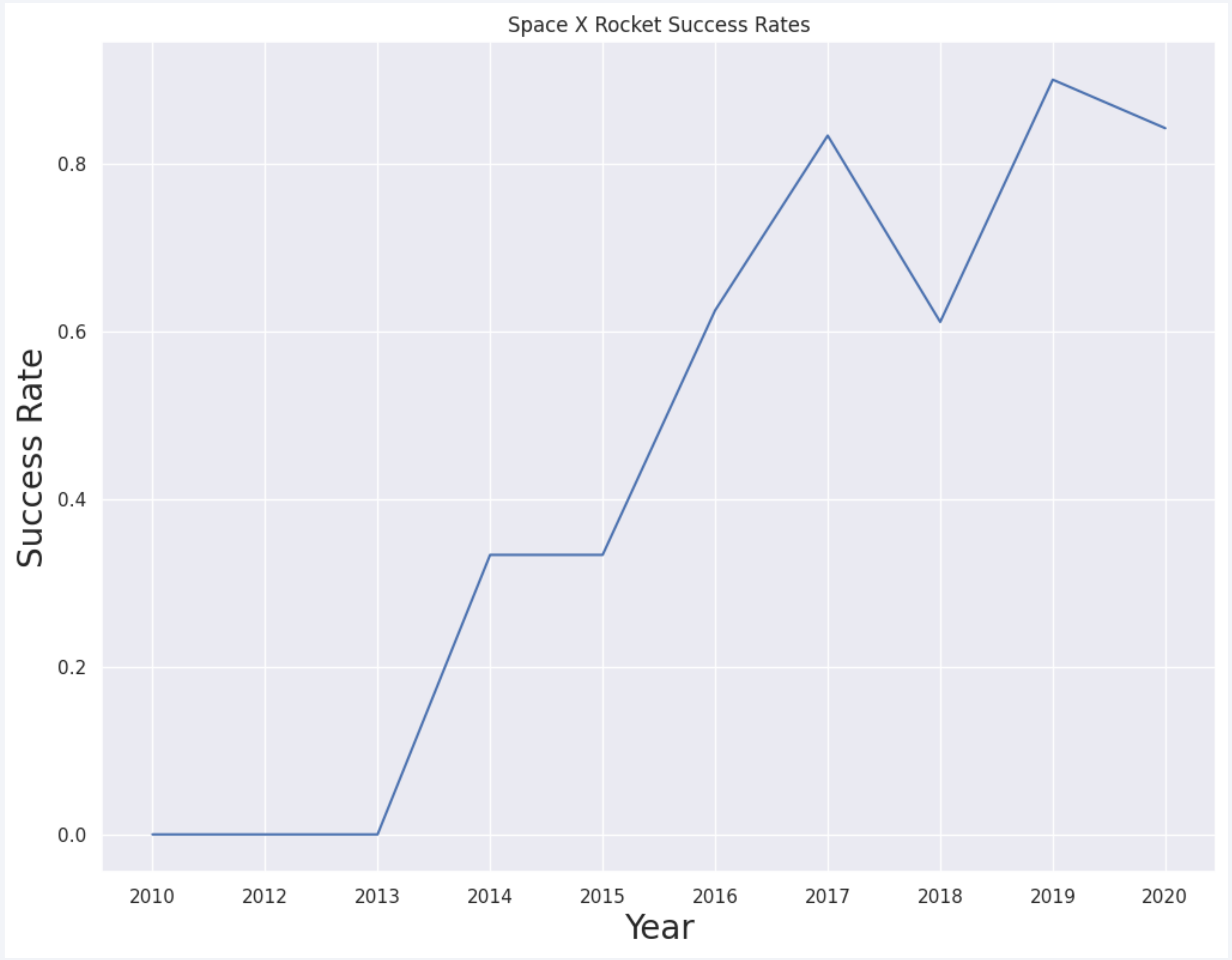
Payload vs. Orbit Type

- Heavy payload have negative influence on MEO, GTO, VLEO orbits, but positive on LEO and ISS orbits



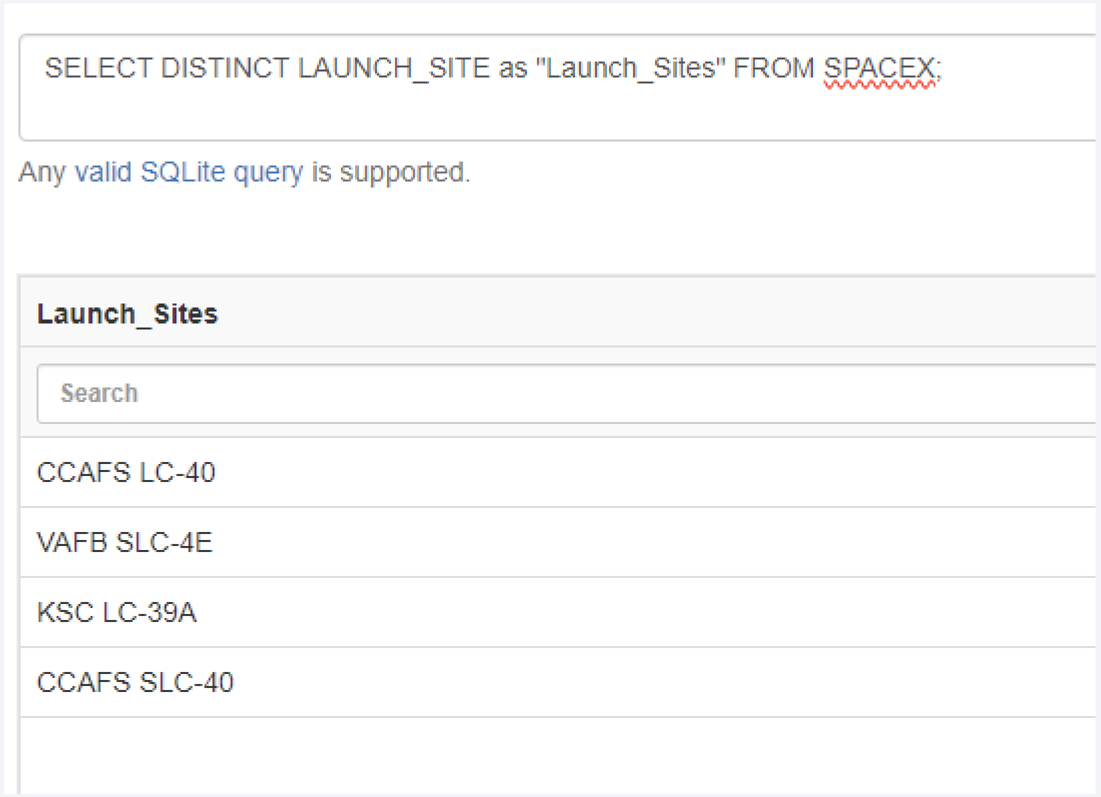
Launch Success Yearly Trend

- Success rate after 2013 shows keep increasing till 2020



All Launch Site Names

- Using Distinct to show unique names



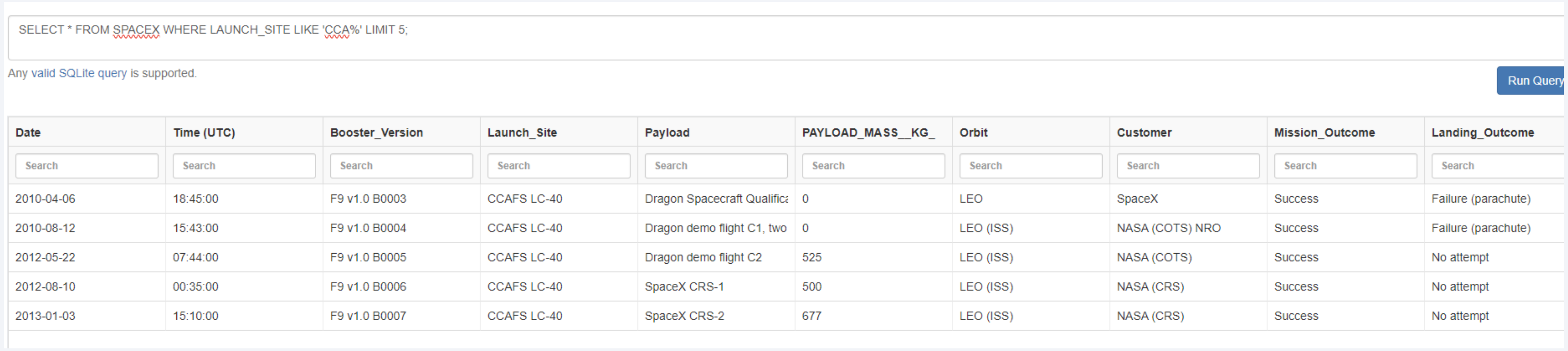
```
SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

Any valid SQLite query is supported.

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

Launch Site Names Begin with 'CCA'

- Using Limit 5 to show 5 records



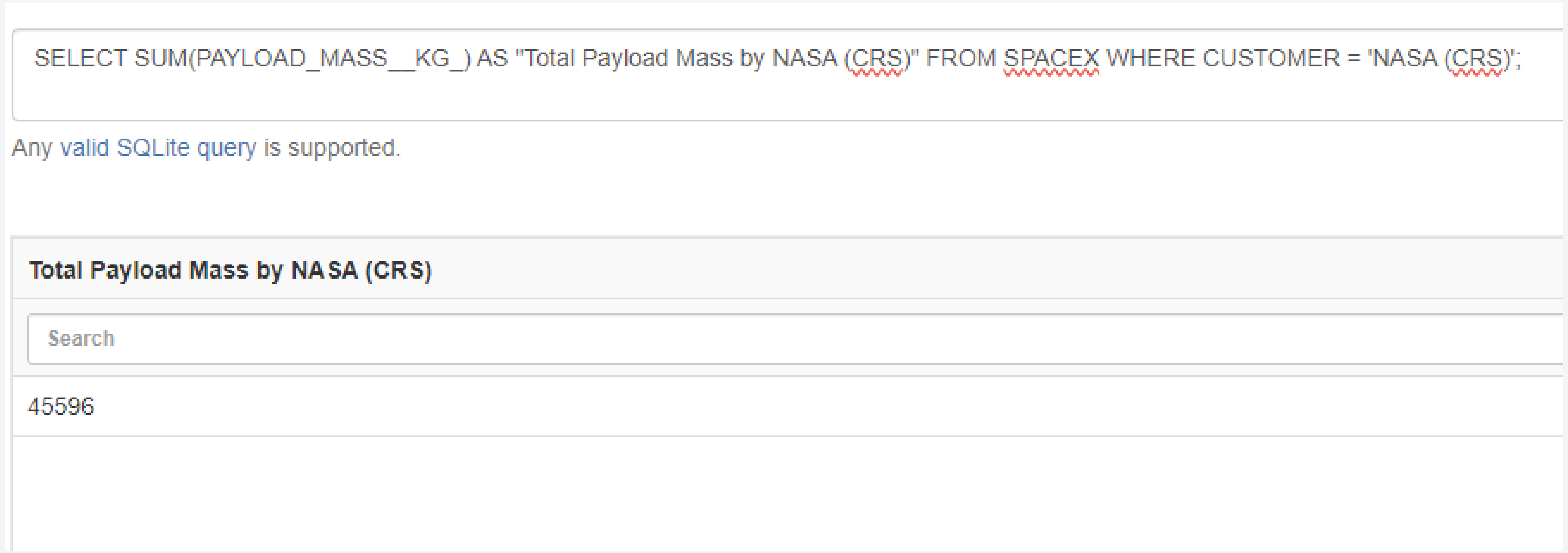
```
SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

Any valid SQLite query is supported.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualifica	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Use SUM function to calculate total payload mass



```
SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

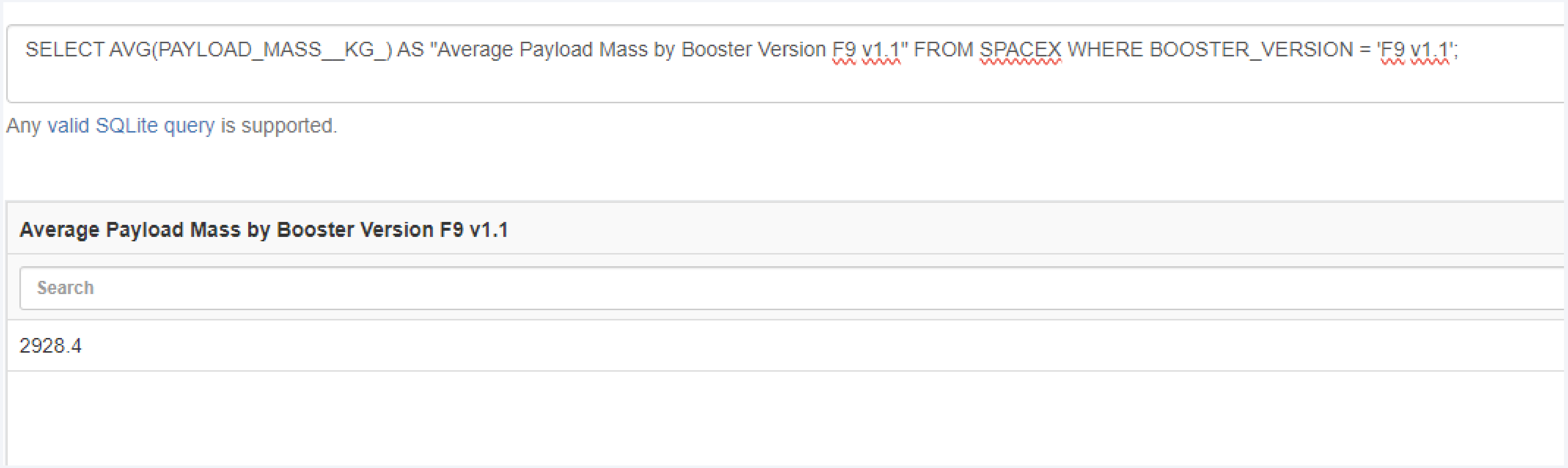
Any valid SQLite query is supported.

```
Total Payload Mass by NASA (CRS)

45596
```

Average Payload Mass by F9 v1.1

- Use AVG function to get average



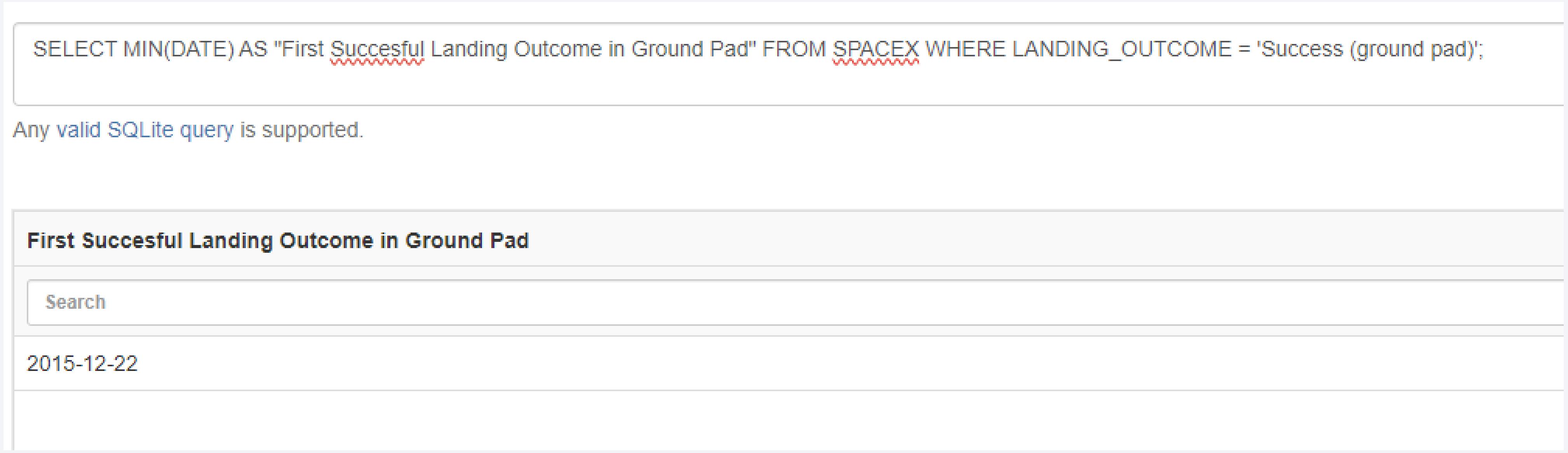
```
SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Any valid SQLite query is supported.

Average Payload Mass by Booster Version F9 v1.1
2928.4

First Successful Ground Landing Date

- Use MIN function to get minimum date and WHERE function to filter



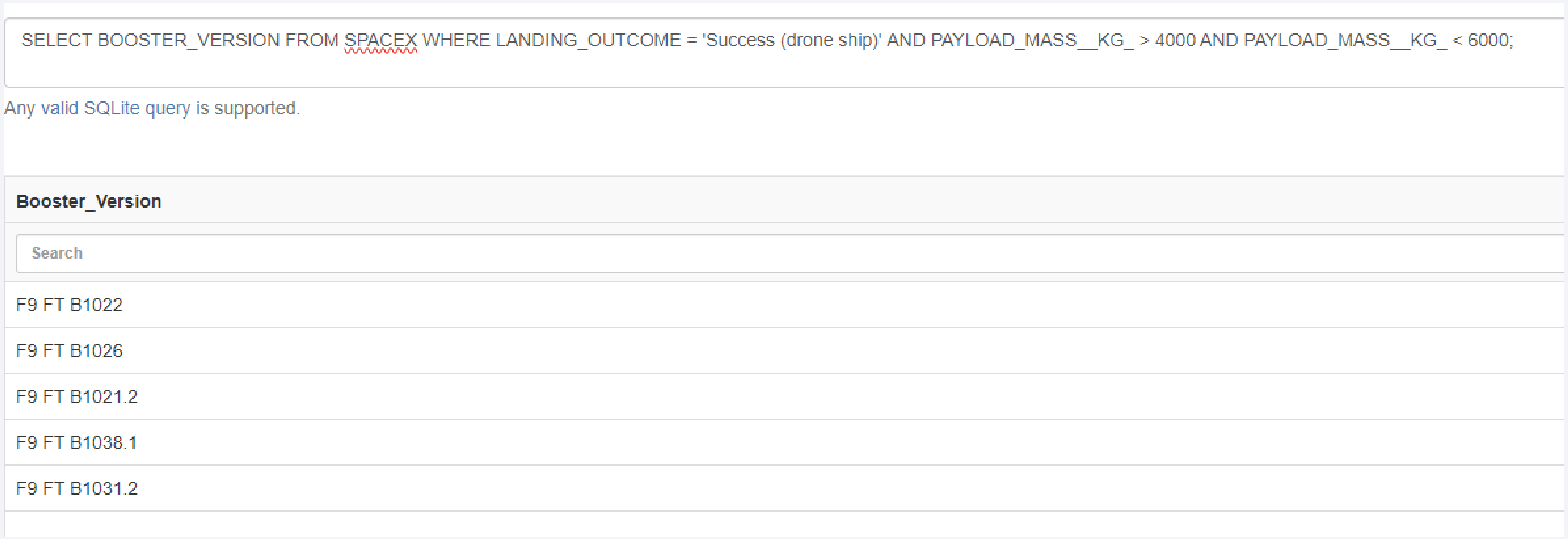
```
SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

Any valid SQLite query is supported.

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Use WHERE clause to filter dataset and AND to give additional condition



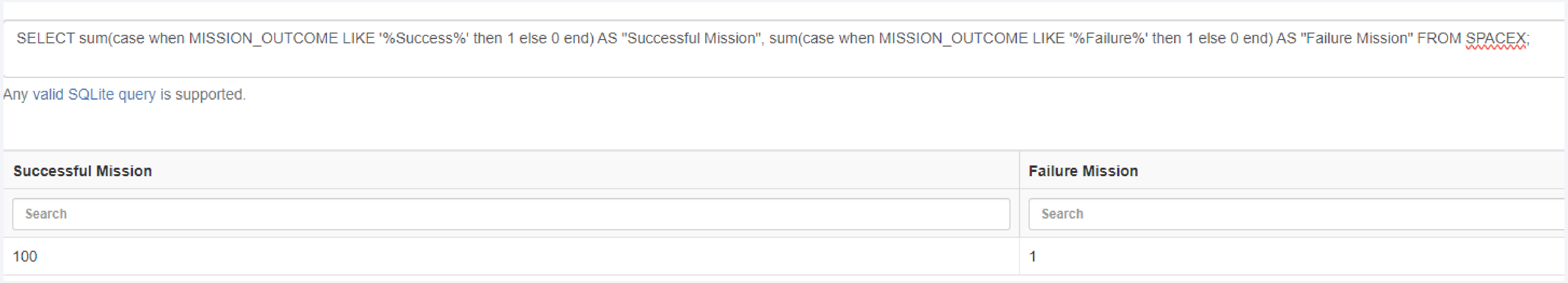
```
SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Any valid SQLite query is supported.

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1038.1
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Use CASE clause with subqueries to get both success and failure



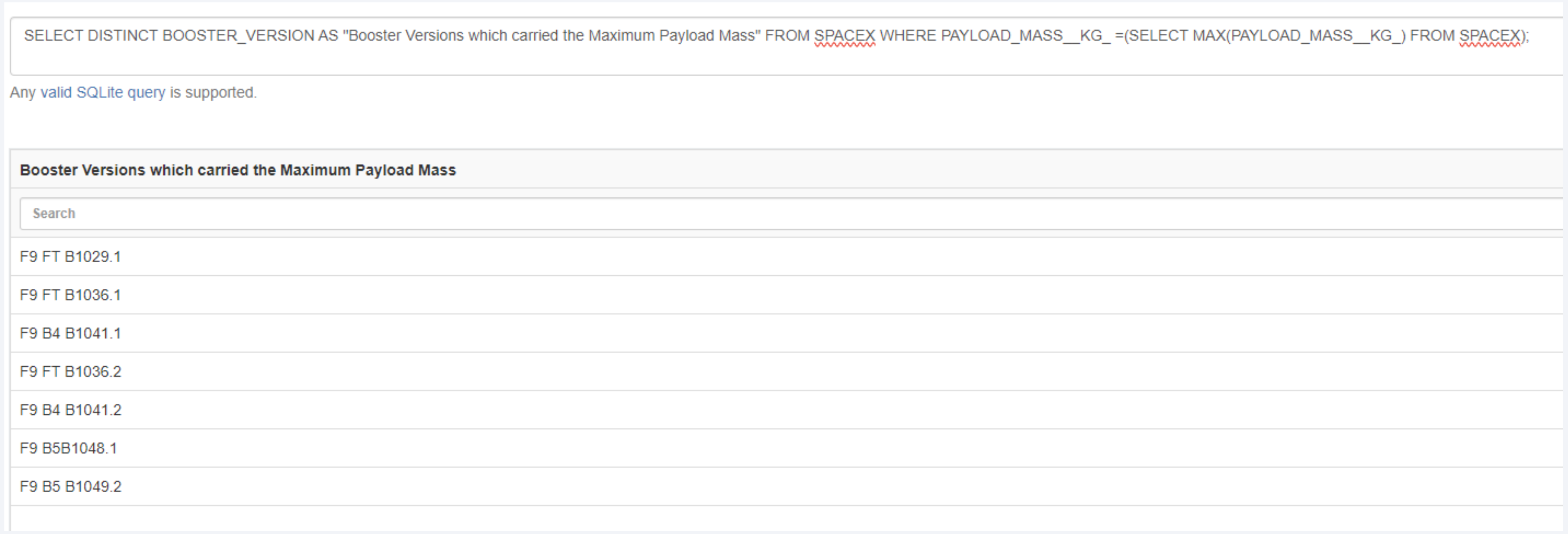
```
SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" FROM SPACEX;
```

Any valid SQLite query is supported.

Successful Mission	Failure Mission
Search	Search
100	1

Boosters Carried Maximum Payload

- Use MAX function for maximum payload and WHERE clause to filter Booster Version



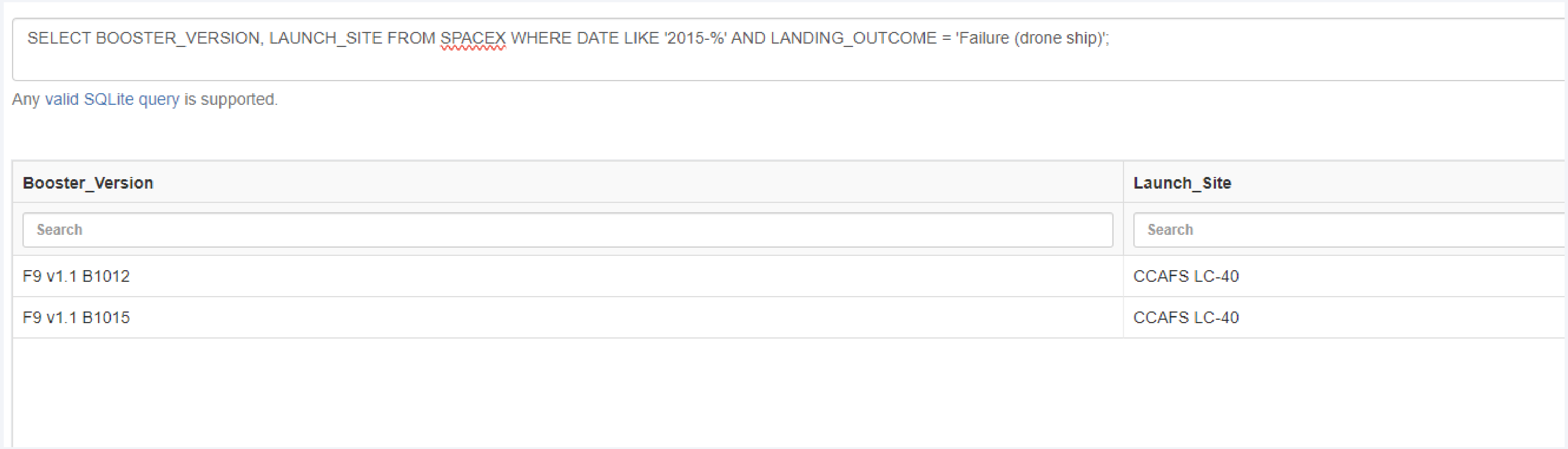
```
SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

Any valid SQLite query is supported.

Booster Versions which carried the Maximum Payload Mass
F9 FT B1029.1
F9 FT B1036.1
F9 B4 B1041.1
F9 FT B1036.2
F9 B4 B1041.2
F9 B5B1048.1
F9 B5 B1049.2

2015 Launch Records

- Use DATE LIKE 2015 to find records of 2015



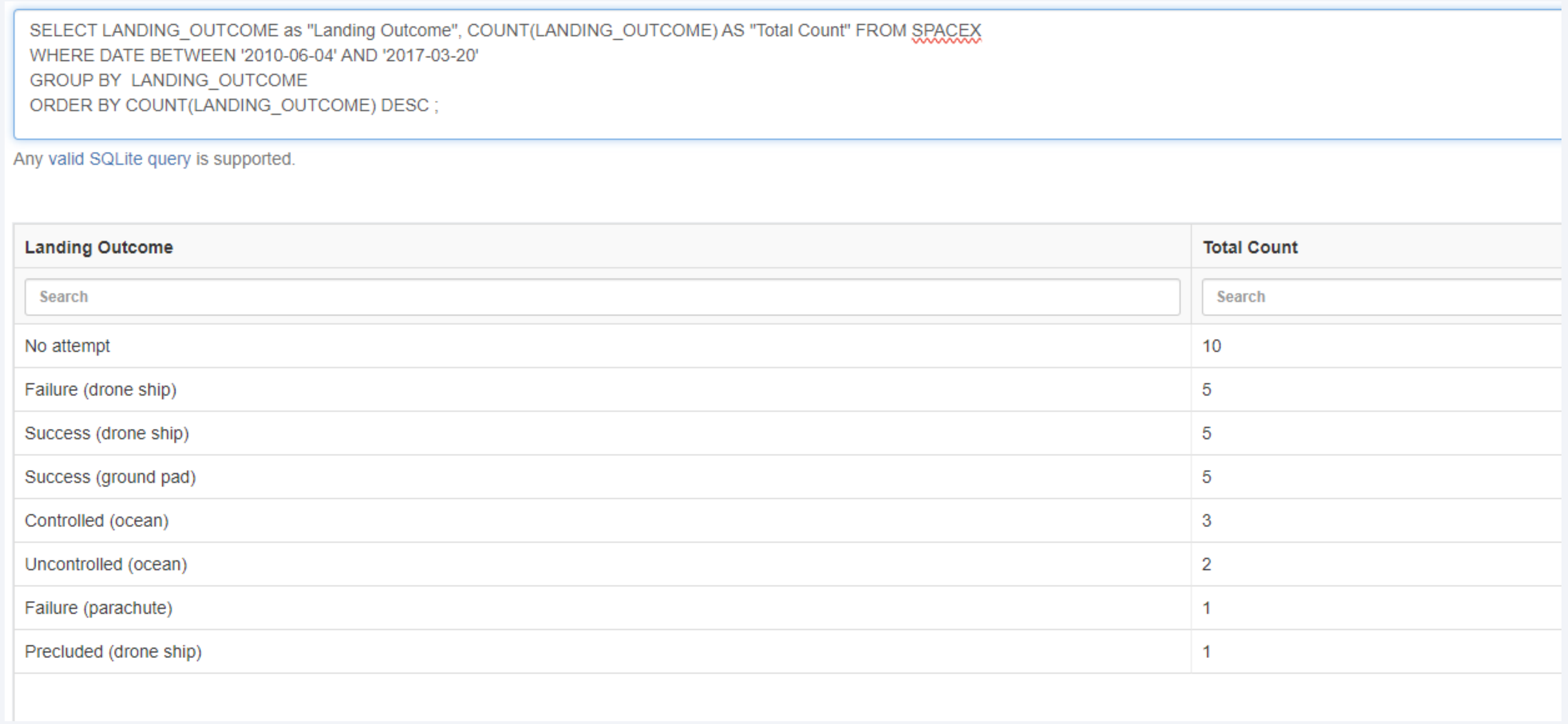
```
SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND LANDING_OUTCOME = 'Failure (drone ship)';
```

Any valid SQLite query is supported.

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- Use COUNT function and WHERE clause to filter date between 2010 to 2017. Also, used GROUP BY and ORDER BY to group by landing outcome and order in descend



```
SELECT LANDING_OUTCOME as "Landing Outcome", COUNT(LANDING_OUTCOME) AS "Total Count" FROM SPACEX
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY  LANDING_OUTCOME
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

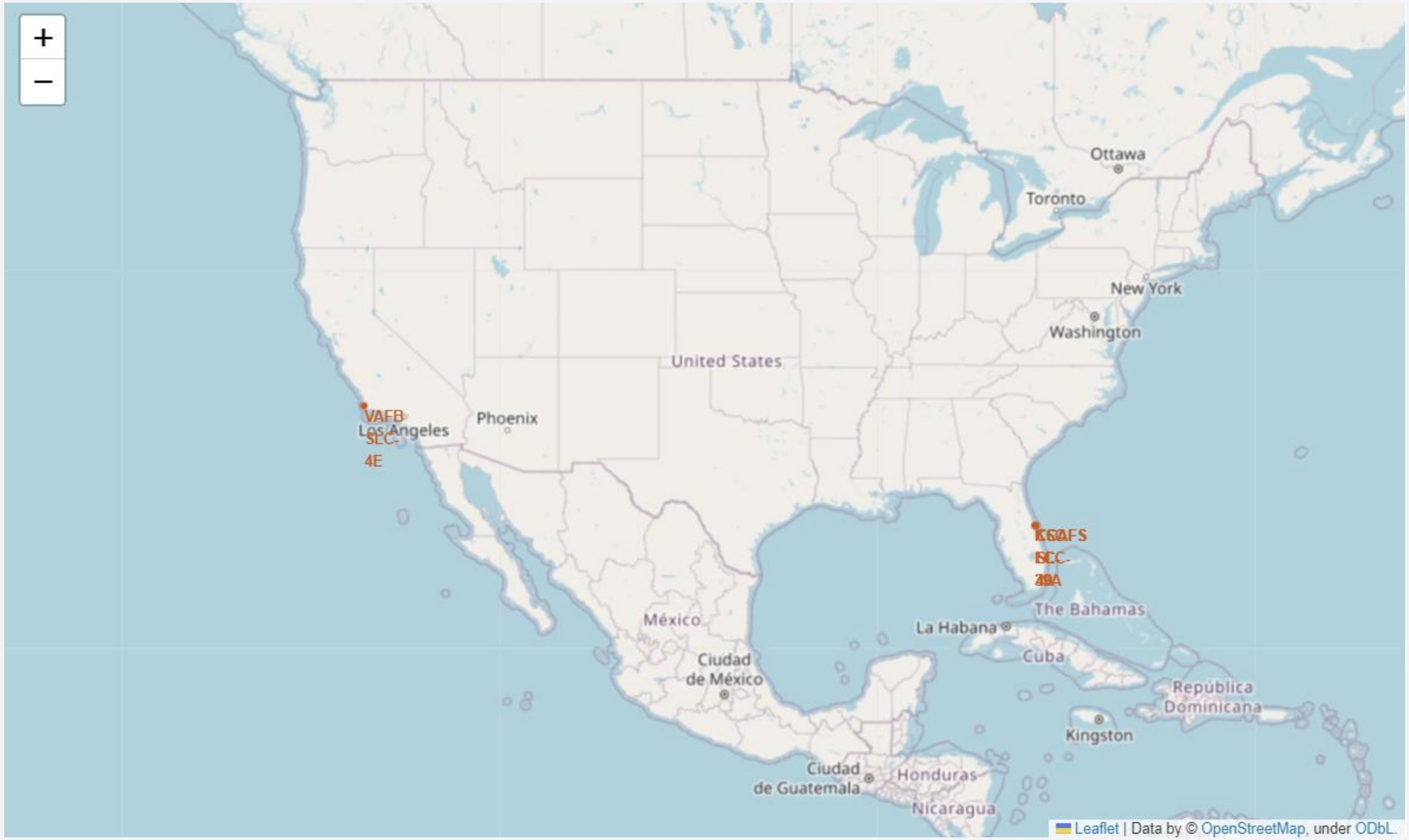
Any valid SQLite query is supported.

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

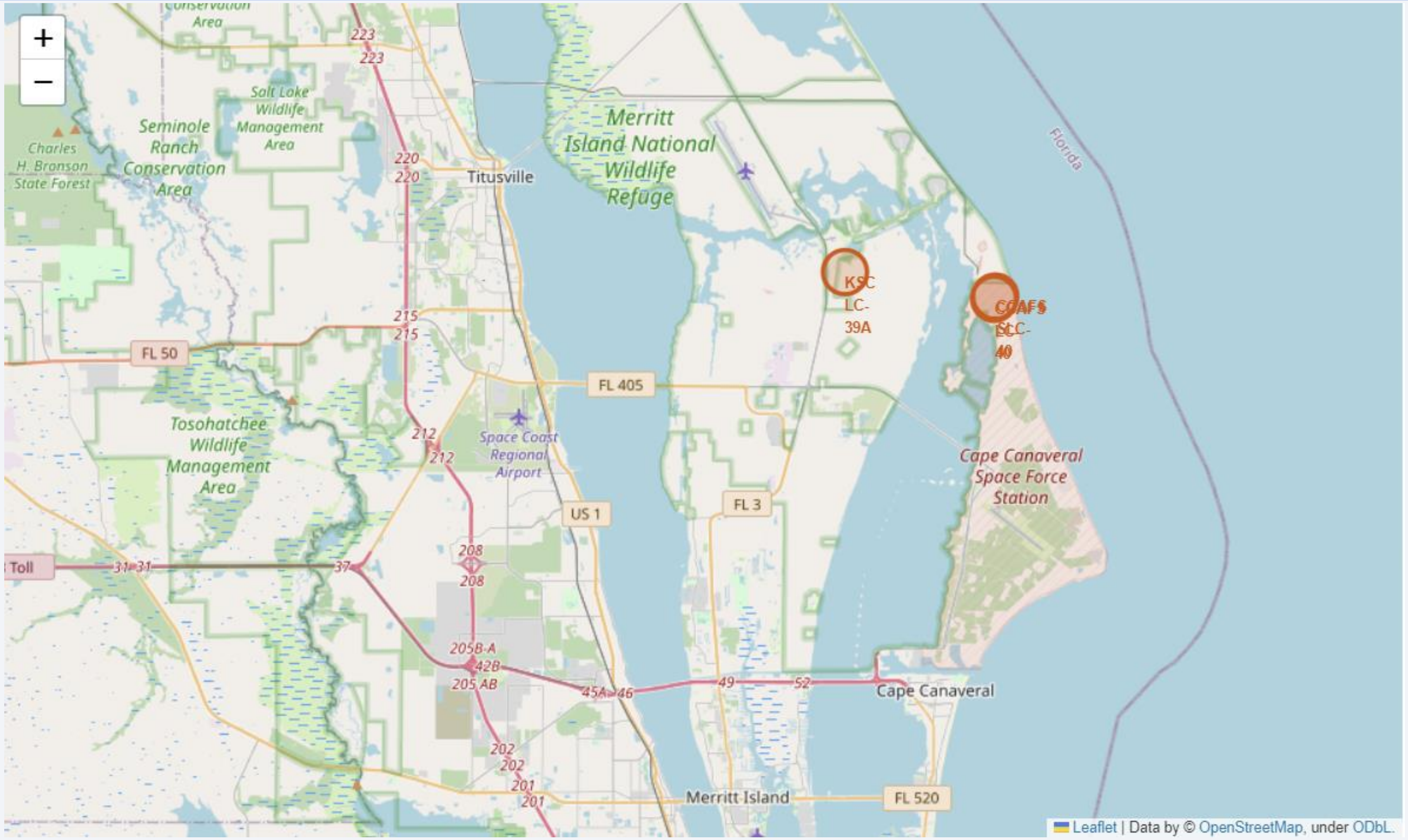
Section 3

Launch Sites Proximities Analysis

<All Launch Sites in world map>


+
−
VAFB SLC-4E
Los Angeles
Phoenix
United States
Ottawa
Toronto
New York
Washington
CCAFS SLC-40
KSC LC-39A
The Bahamas
La Habana
Cuba
México
Ciudad de México
República Dominicana
Kingston
Ciudad de Guatemala
Honduras
Nicaragua
Leaflet | Data by © OpenStreetMap, under ODbL.

<Showing Launch sites and landmarks>

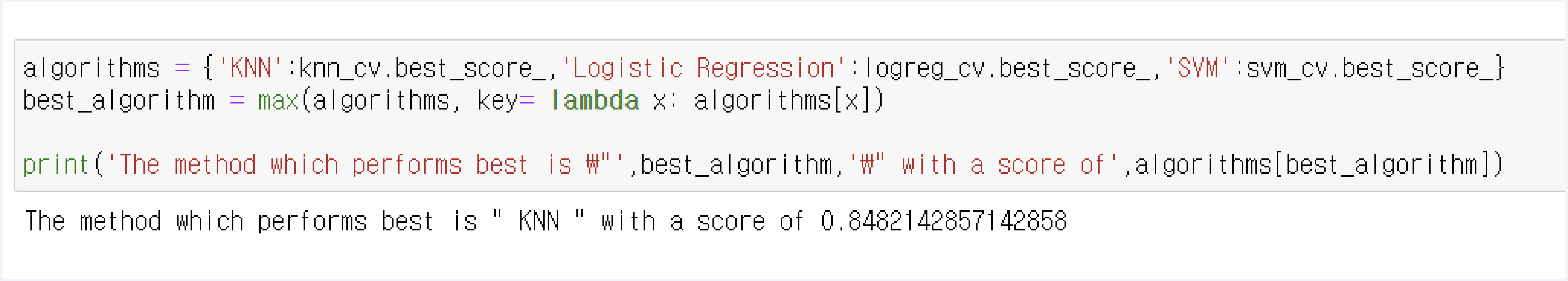

KSC LC-39A
CCAFS SLC-40
Titusville
Merritt Island National Wildlife Refuge
Cape Canaveral Space Force Station
Cape Canaveral
Merritt Island
Space Coast Regional Airport
Leaflet | Data by © OpenStreetMap, under ODbL.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- KNN classifier is the model with the highest classification accuracy



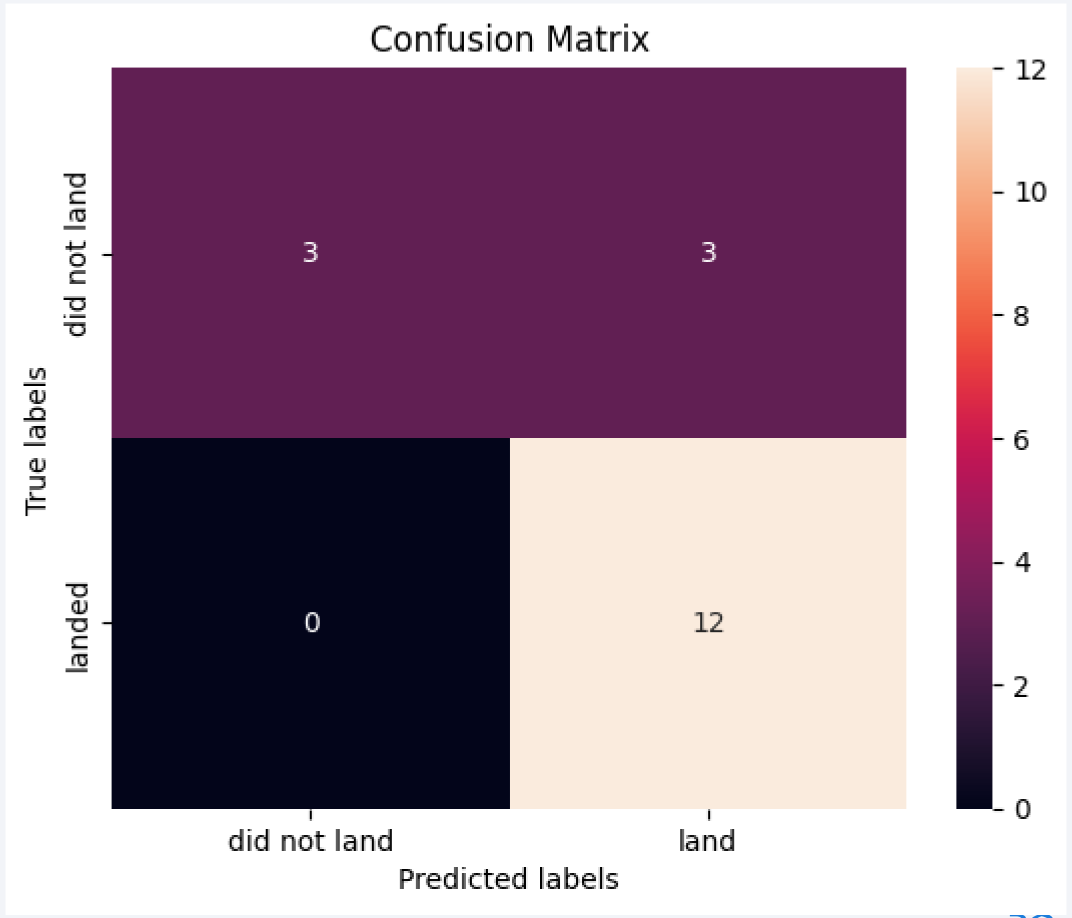
```
algorithms = {'KNN':knn_cv.best_score_,'Logistic Regression':logreg_cv.best_score_,'SVM':svm_cv.best_score_}
best_algorithm = max(algorithms, key= lambda x: algorithms[x])

print('The method which performs best is ₩"',best_algorithm,'₩" with a score of',algorithms[best_algorithm])
```

```
The method which performs best is " KNN " with a score of 0.8482142857142858
```

Confusion Matrix

- It shows unsuccessful landing marked as successful landing.



Conclusions

- Larger flight amount at launch site, the greater success rate.
- Success rate started to increase from 2013 to 2020.
- KNN classifier shows the best classification model
- Couldn’t make dashboard due to problem of import library.

Thank you!