Data Science Capstone project

Andrew Groves

01/09/2021

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with data visualization
  - EDA with SQL
  - Interactive map with Folium
  - Dashboard
  - Predictive analysis
- Summary of all results
  - Exploratory data analysis results
  - Interactive demo
  - Predictive results

Introduction

- Project background and context

SpaceX advertises that the falcon 9 rocket will cost 62 million dollars however other providers say it will cost 165 million. The difference is that the falcon 9 will land the first stage and reuse it. This is a large saving, and we will be able to predict the total cost of the rocket if we can predict if it will land successfully.

- Problems you want to find answers
- What will influence if the rocket will land or not?
- What is the effect of each relationship to the success of the rocket or not?

Methodology

- Data collection methodology:
  - Using the SpaceX rest API
  - Web scrapping from the internet
- Perform data wrangling
  - The data will be prepared by cleaning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

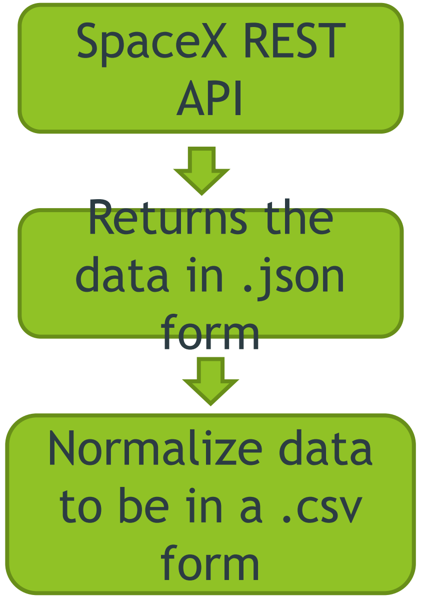
Methodology

Data collection

The main data source is rom the REST API, this can give information on everything from rocket used to landing specifications.

The other way to gain information is from web scrapping, for this purpose we use Wikipedia.

API calling

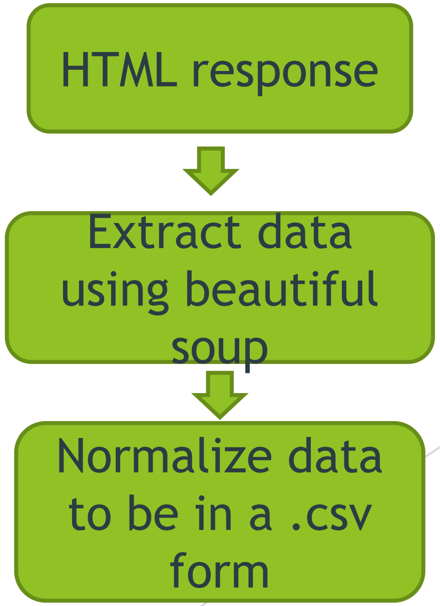


SpaceX REST API

Returns the data in .json form

Normalize data to be in a .csv form

Web Scrapping



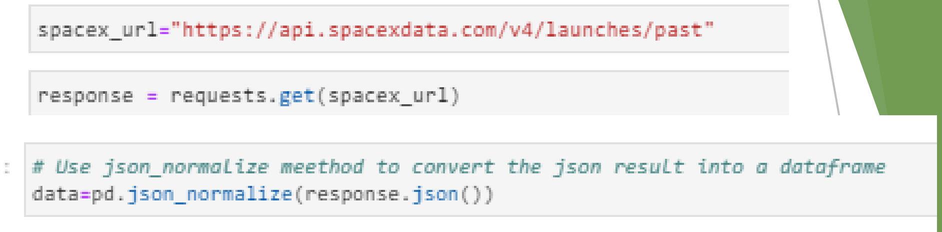
HTML response

Extract data using beautiful soup

Normalize data to be in a .csv form

Data collection – SpaceX API

1. Sets up the API call and fills it inside a data frame

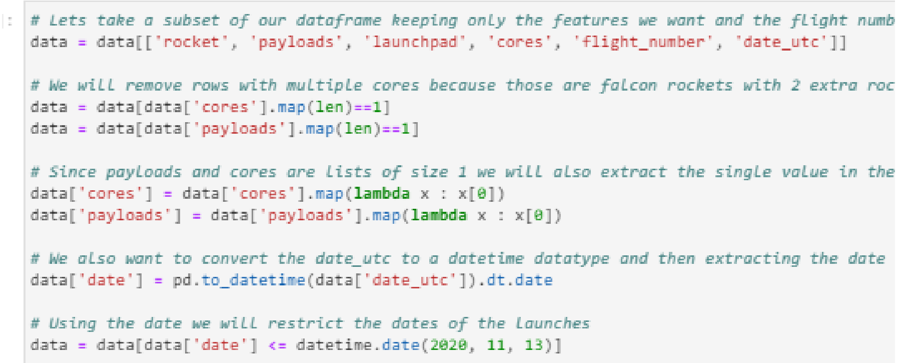


```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

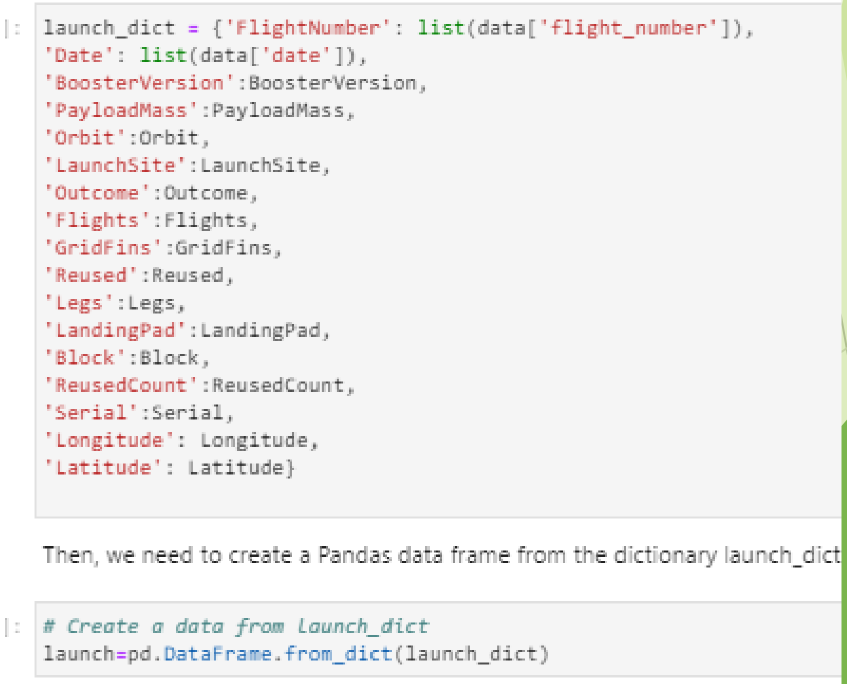
```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

2. Clean the data



```
# Lets take a subset of our dataframe keeping only the features we want and the flight numb
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra roc
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

3. Fills in the data details for each launch



```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

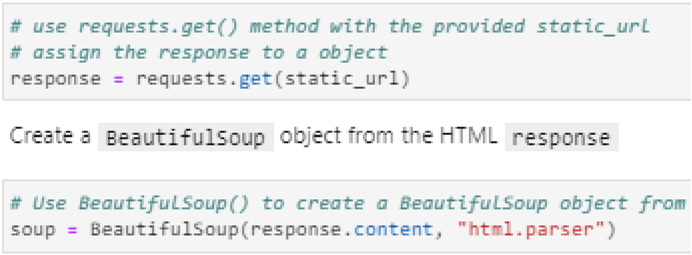
Then, we need to create a Pandas data frame from the dictionary launch_dict

```
# Create a data from launch_dict
launch=pd.DataFrame.from_dict(launch_dict)
```

https://github.com/AndrewJGroves/IBMCaptoneProject

Data collection – Web scraping

1. Get the website content

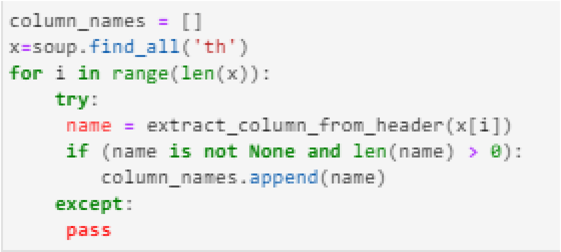


```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

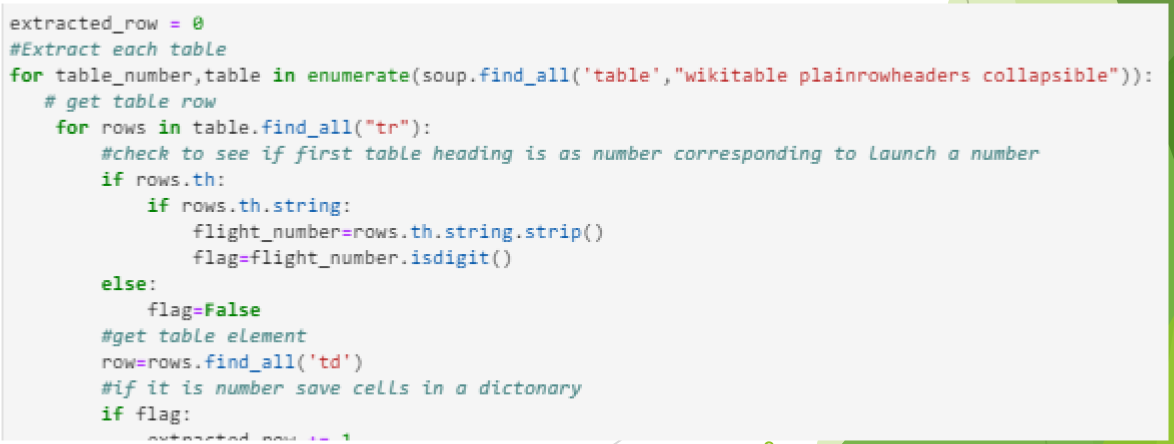
```
# Use BeautifulSoup() to create a BeautifulSoup object from
soup = BeautifulSoup(response.content, "html.parser")
```

2. Find the column names



```
column_names = []
x=soup.find_all('th')
for i in range(len(x)):
    try:
     name = extract_column_from_header(x[i])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

3. Fill a dictionary with the information



```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
```

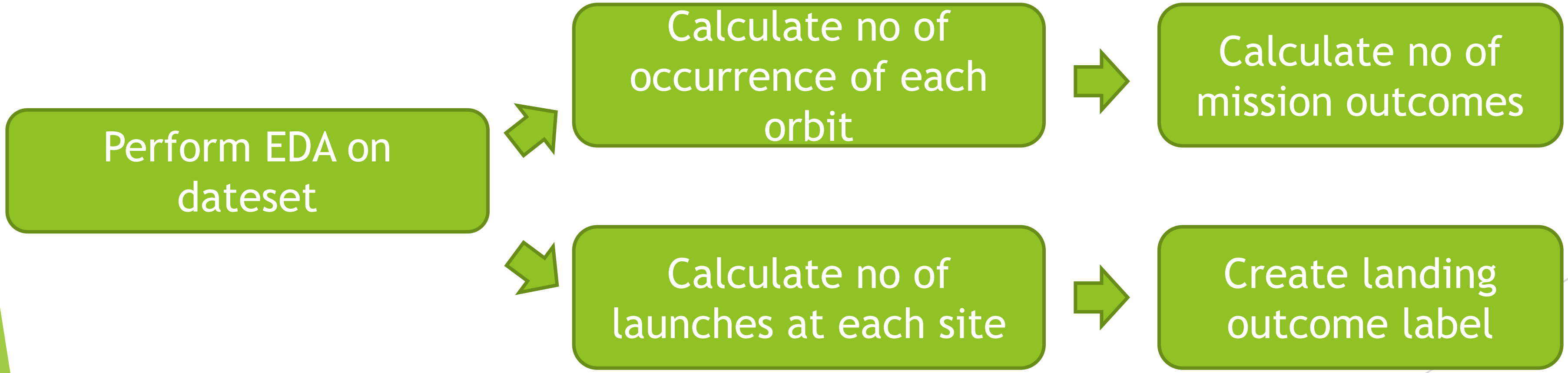
https://github.com/AndrewJGroves/IBMCaptoneProject

Data wrangling

https://github.com/AndrewJGroves/IBMCaptoneProject

- The data contains a lot of information, of interest to us is whether the landing was successful or not. These are then converted into training labels with 1 being successful and 0 being failed.

Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose



Perform EDA on dateset

Calculate no of occurrence of each orbit

Calculate no of mission outcomes

Calculate no of launches at each site

Create landing outcome label

EDA with data visualization

https://github.com/Andrew JGroves/IBMCaptoneProject

Scatter Plots

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload vs Launch Site
- Orbit vs Flight Number
- Payload vs Orbit Type
- Orbit vs Payload Mass

Bar Graph

- Mean vs Orbit

Line Graph

- Success Rate vs Year

EDA with SQL

https://github.com/AndrewJGroves/IBMCaptoneProject

SQL queries used to gain information, the following was used

- The names of the unique launch sites in the space mission
- 5 records where launch sites being with the string KSC
- The total payload mass carried by boosters launched by NASA
- Average payload mass carried by booster F9 v1.1
- List dates where the successful landing outcome in drone ship was achieved.
- List the names of the boosters which have success in ground pad 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, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017
- Ranking the count of successful landing outcomes between the date 2010 06 04 and 2017 03 20 in descending order.

Build an interactive map with Folium

https://github.com/AndrewJGroves/IBMCaptoneProject

- Each launch site is labeled and marked on a interactive map
- Green markers will be successful sites and red unsuccessful
- Find the distance to various landmarks such as railways, coastlines and cities.

Build a Dashboard with Plotly Dash

https://github.com/AndrewJGroves/IBMCaptoneProject

- Dashboard is built flask and dash, various plots are shown
- Pie chart showing the total launches by sites
- Scatter graph showing the outcome and payload mass.

Predictive analysis (Classification)

https://github.com/AndrewJGroves/IBMCaptoneProject

Building

- Split our data into training and test data
- Decide on the machine learning algorithms
- Set parameters and algorithms to GridSearchCV
- Train the dataset

Evaluating

- Check the accuracy of the model
- Tune the parameters
- Plot Confusion matrix

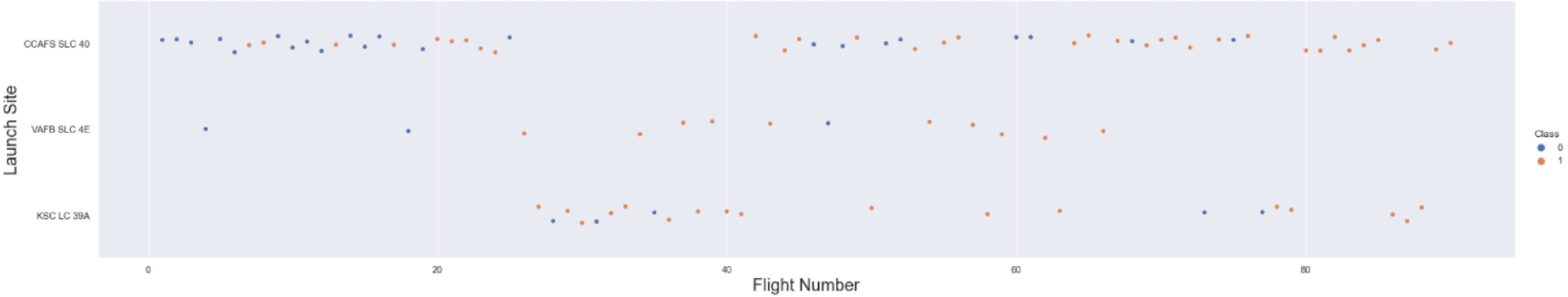
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

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

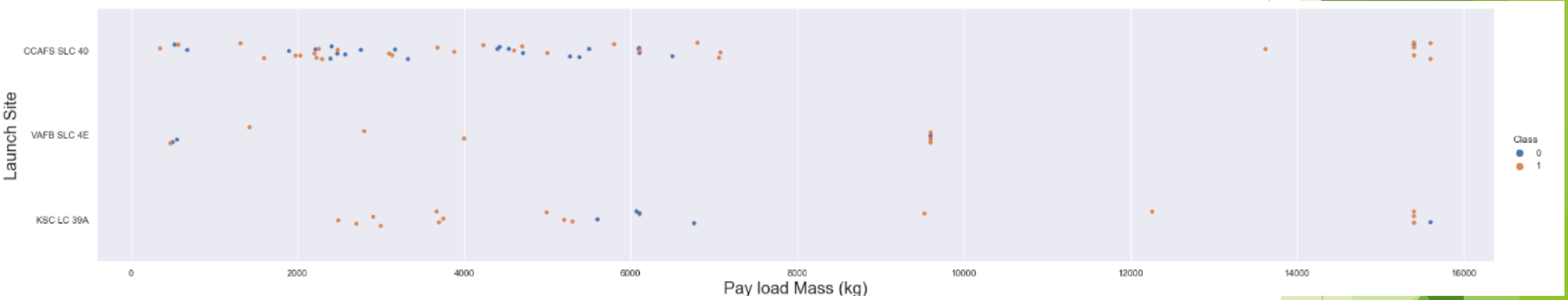
EDA with Visualization

Flight Number vs. Launch Site



As the number of flights increased the more successes there are

Payload vs. Launch Site

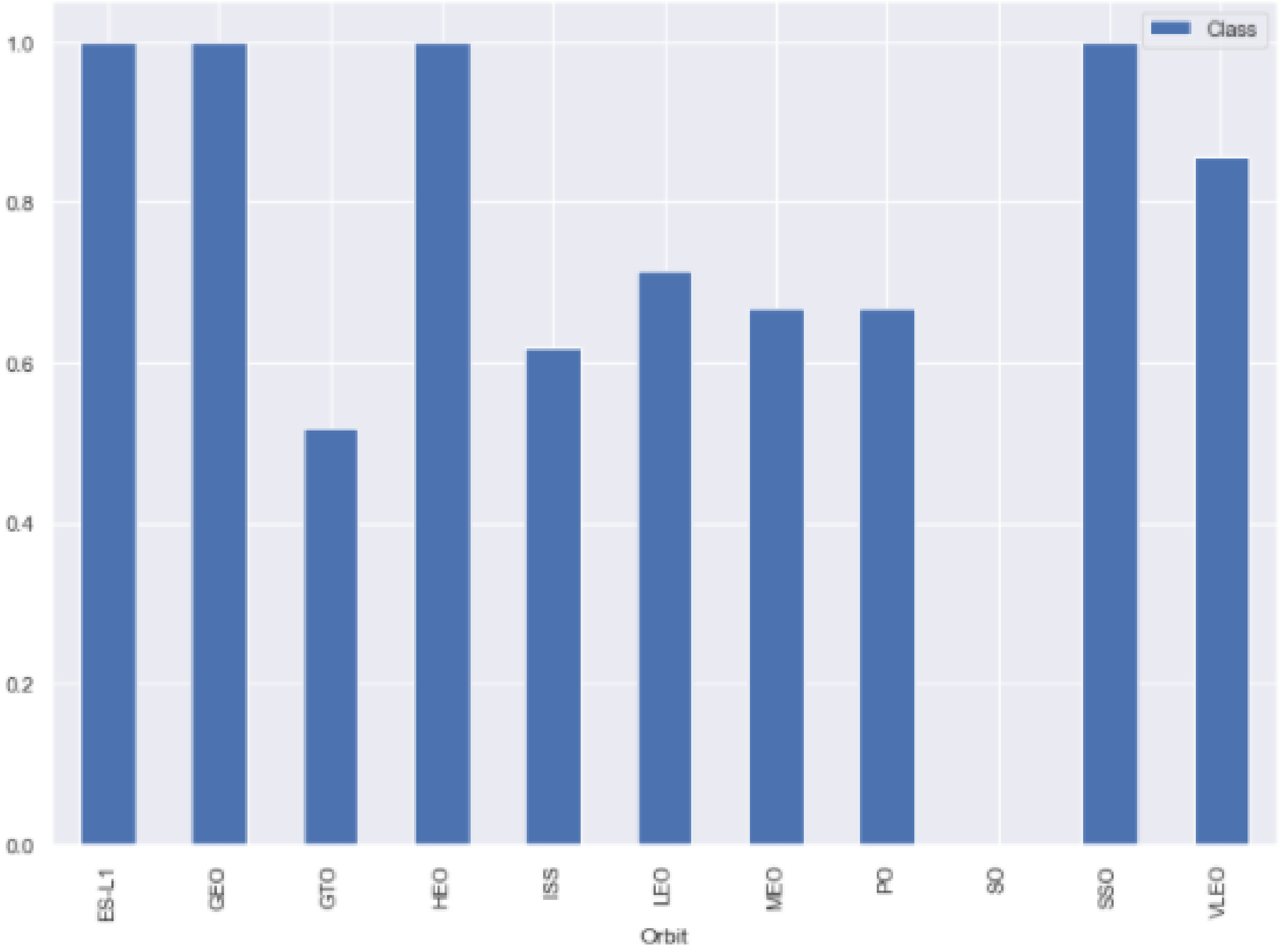


The higher the mass the more likely a success, however there is not enough information to say this for sure.

Success rate vs. Orbit type

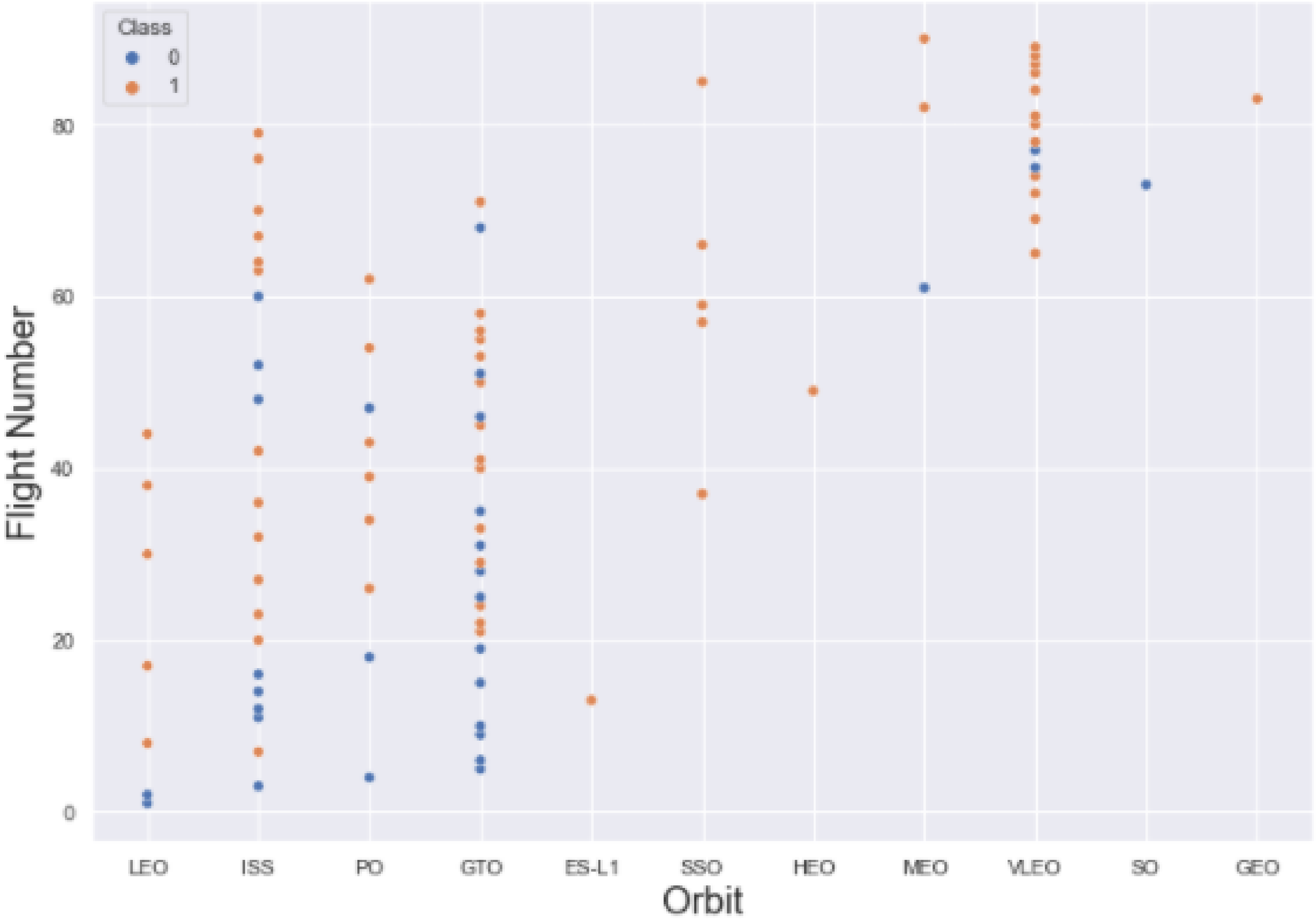
4 orbit types have the best chance of success according to this graph.

However we don't know when the flights took place, or how many flights



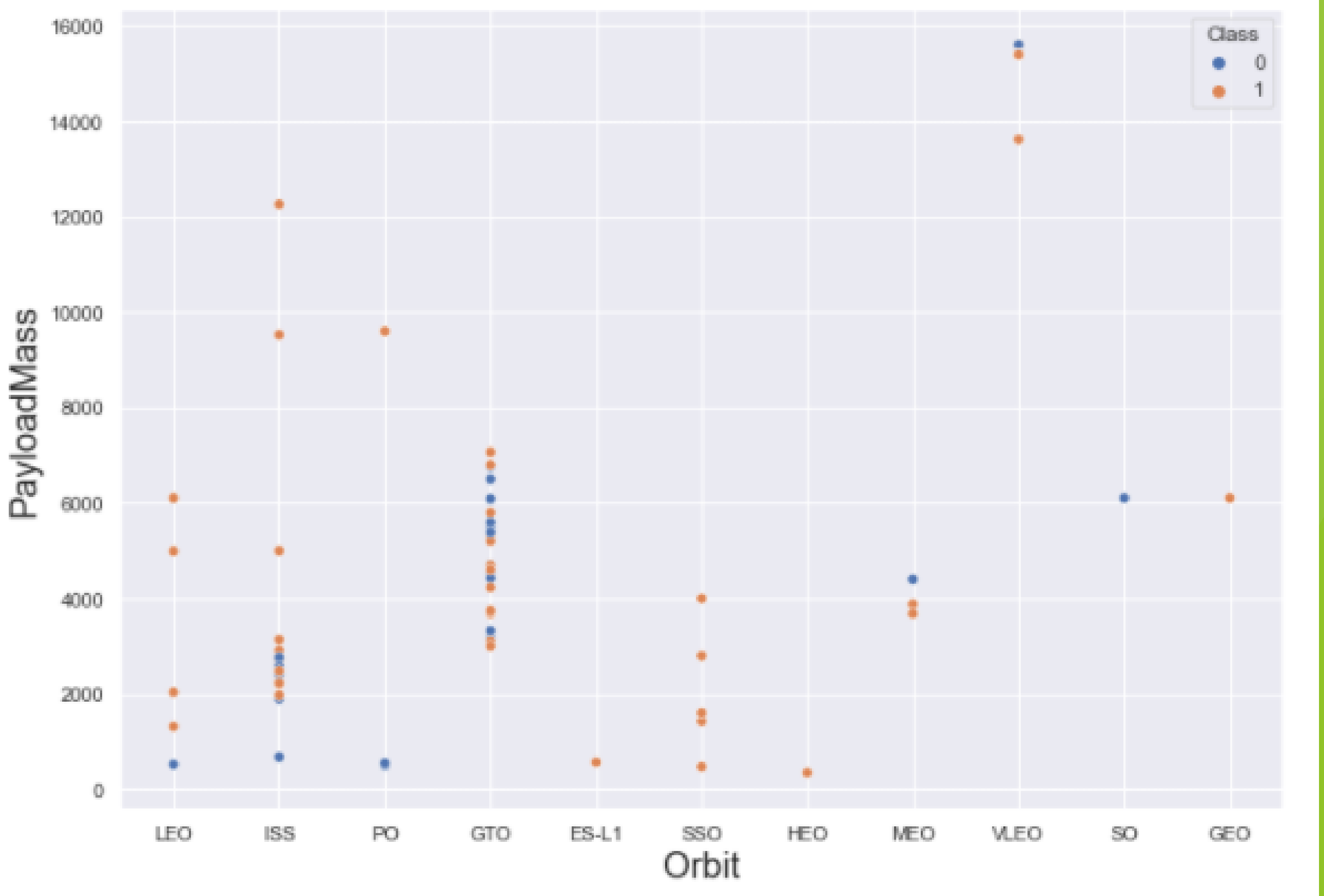
Flight Number vs. Orbit type

We now show that some of the bigger success rate from previous plots don’t have many launchs



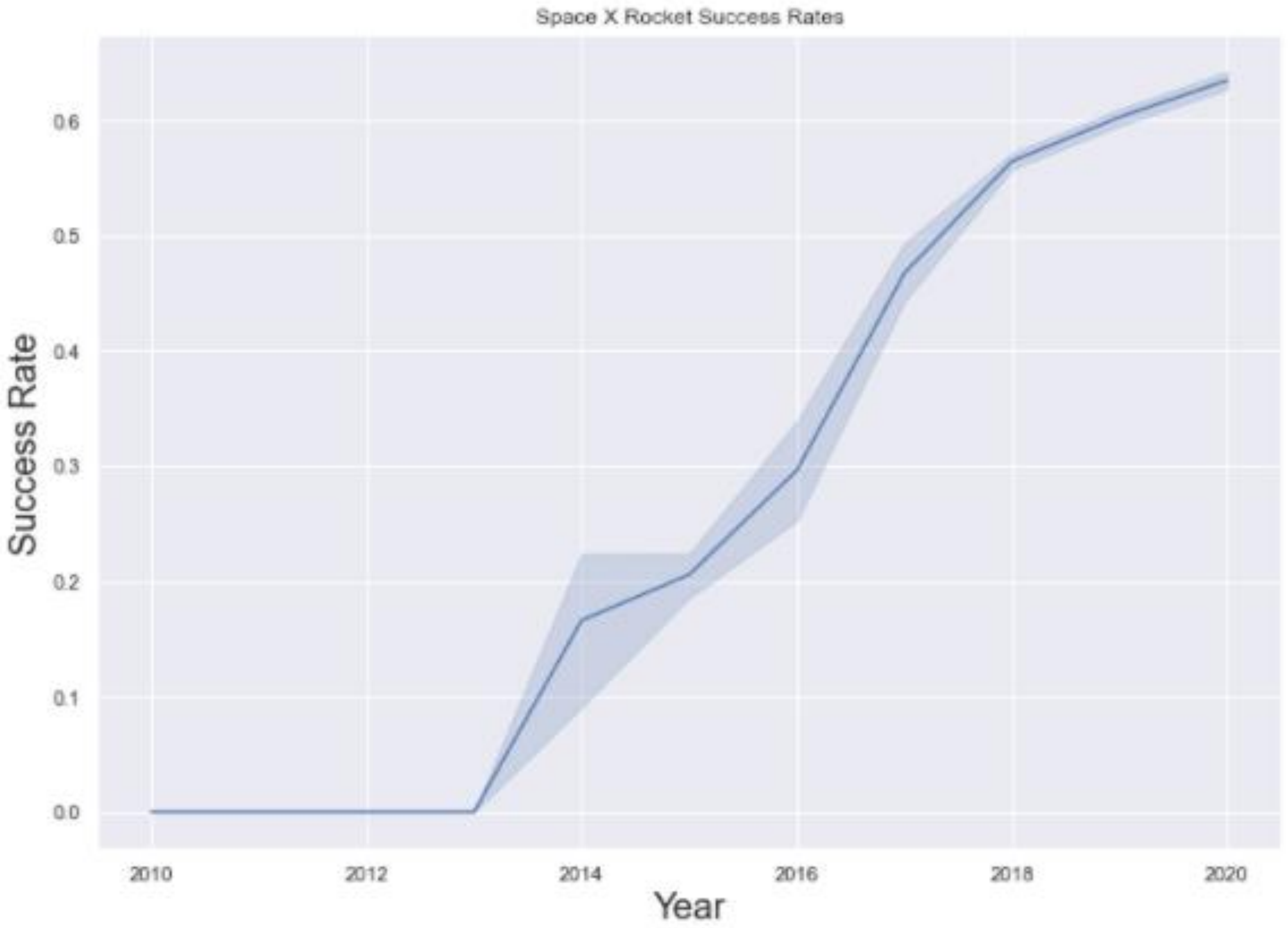
Payload vs. Orbit type

There is not enough information to draw good conclusions about success rate. This is because we don’t know when the orbit would happen



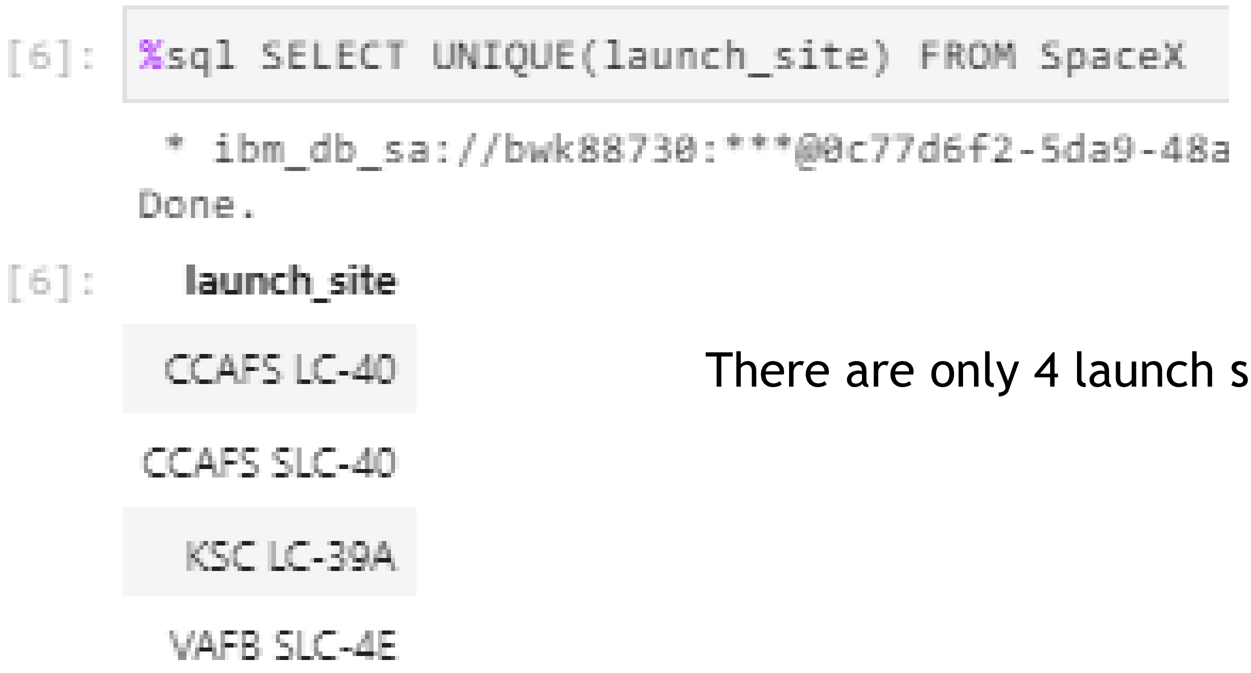
Launch success yearly trend

This shows that as time went on, the success rate increases



EDA with SQL

All launch site names



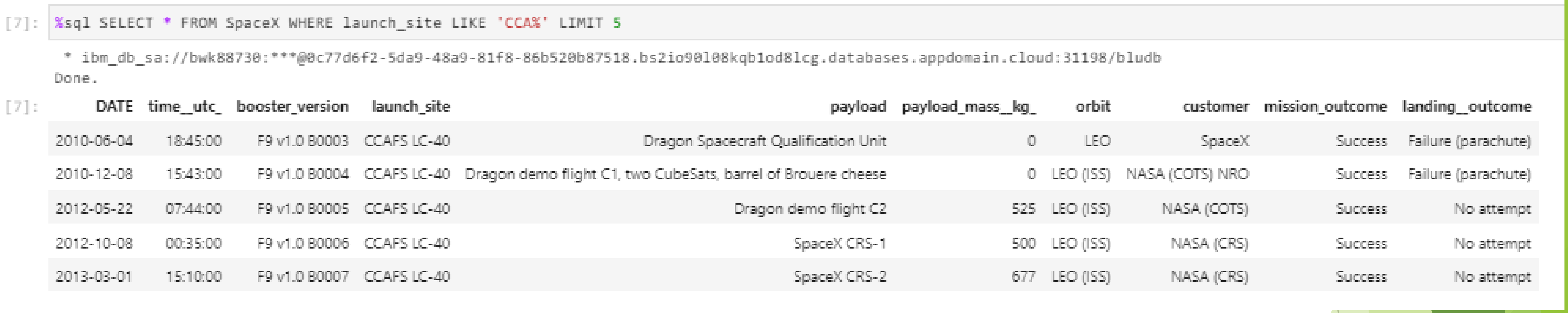
```
[6]: %sql SELECT UNIQUE(launch_site) FROM SpaceX

 * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a
Done.
```

[6]:

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

There are only 4 launch sites used by SpaceX

Launch site names begin with `CCA`



```
[7]: %sql SELECT * FROM SpaceX WHERE launch_site LIKE 'CCA%' LIMIT 5
```

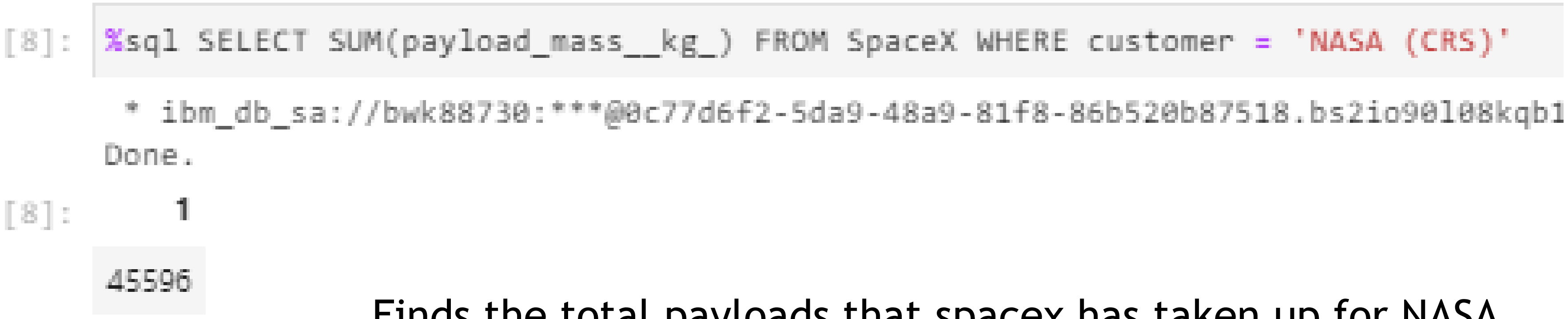
```
 * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

[7]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Lists the first 5 launches from CCA site

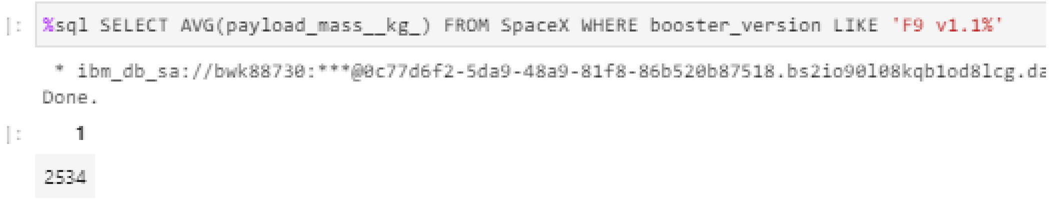
Total payload mass



```
[8]: %sql SELECT SUM(payload_mass__kg_) FROM SpaceX WHERE customer = 'NASA (CRS)'

      * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
     Done.

[8]:      1
     45596
```

Finds the total payloads that spacex has taken up for NASA

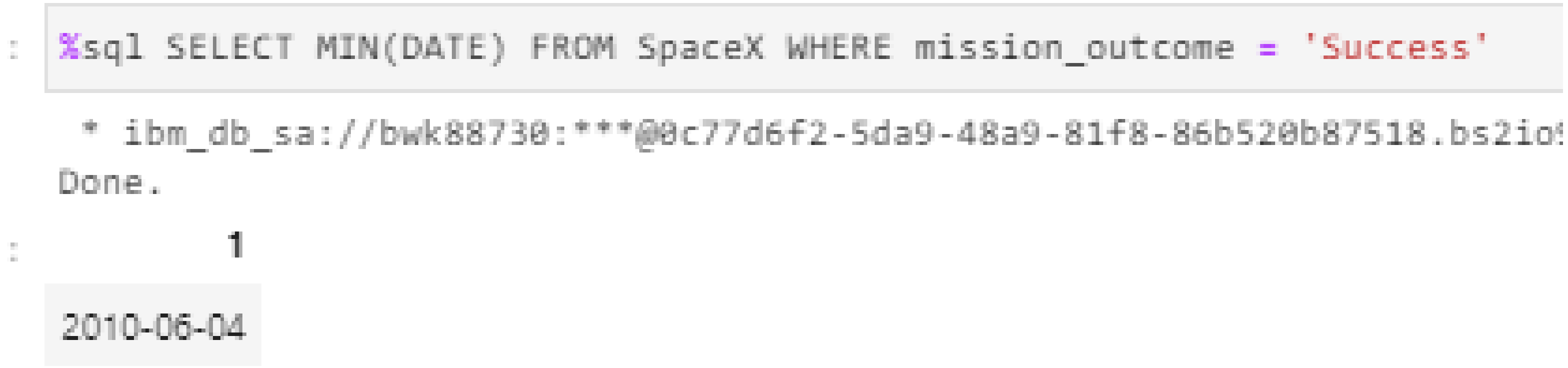
Average payload mass by F9 v1.1



```
]: %sql SELECT AVG(payload_mass__kg_) FROM SpaceX WHERE booster_version LIKE 'F9 v1.1%'

    * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.da
   Done.

]:     1
   2534
```

2534 is the average payload mass carried by booster version F9 v1.1

First successful ground landing date



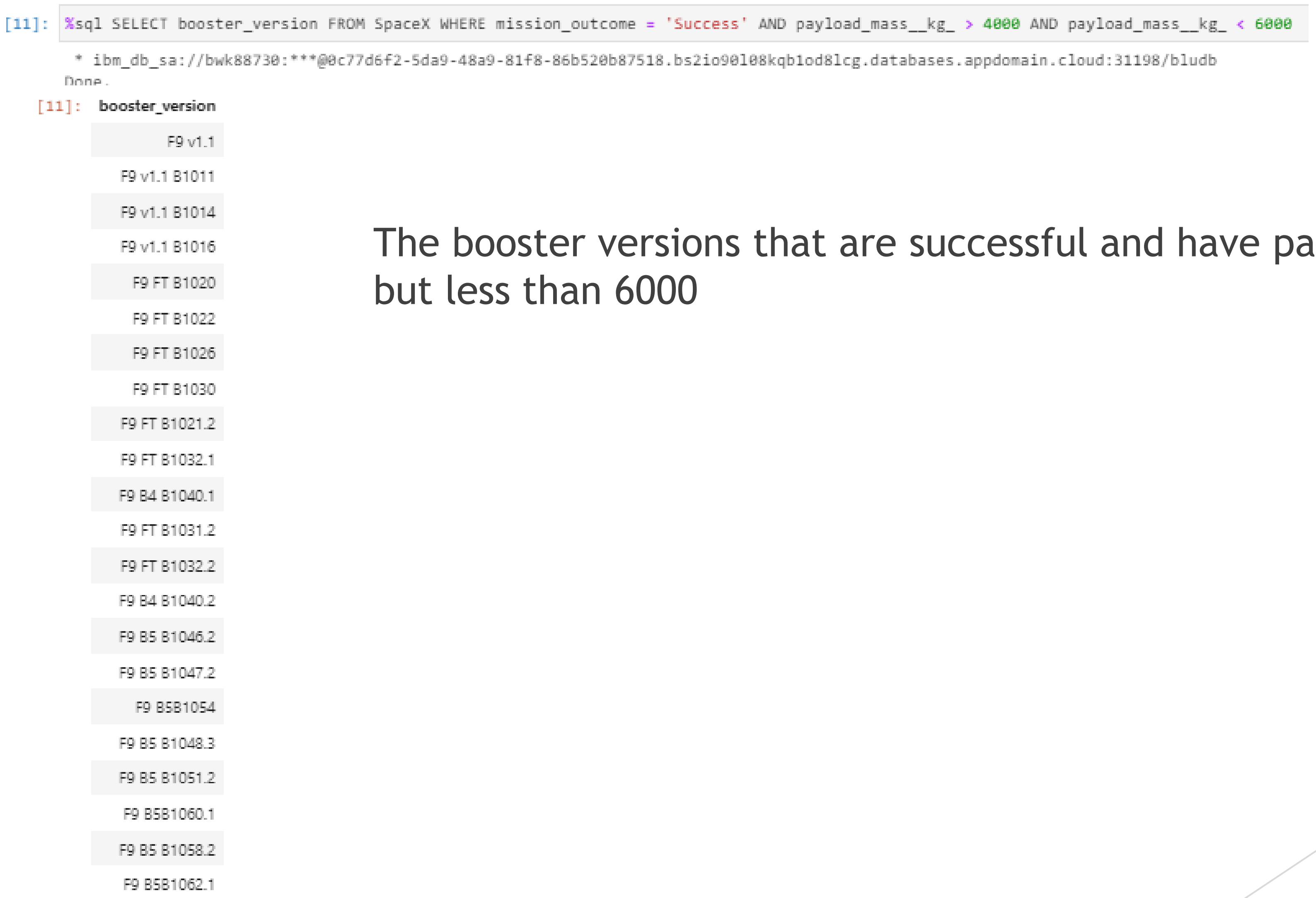
```
%sql SELECT MIN(DATE) FROM SpaceX WHERE mission_outcome = 'Success'
```

```
 * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io
Done.
```

```
         1
2010-06-04
```

The date when the first successful landing outcome was done by SpaceX

Successful drone ship landing with payload between 4000 and 6000



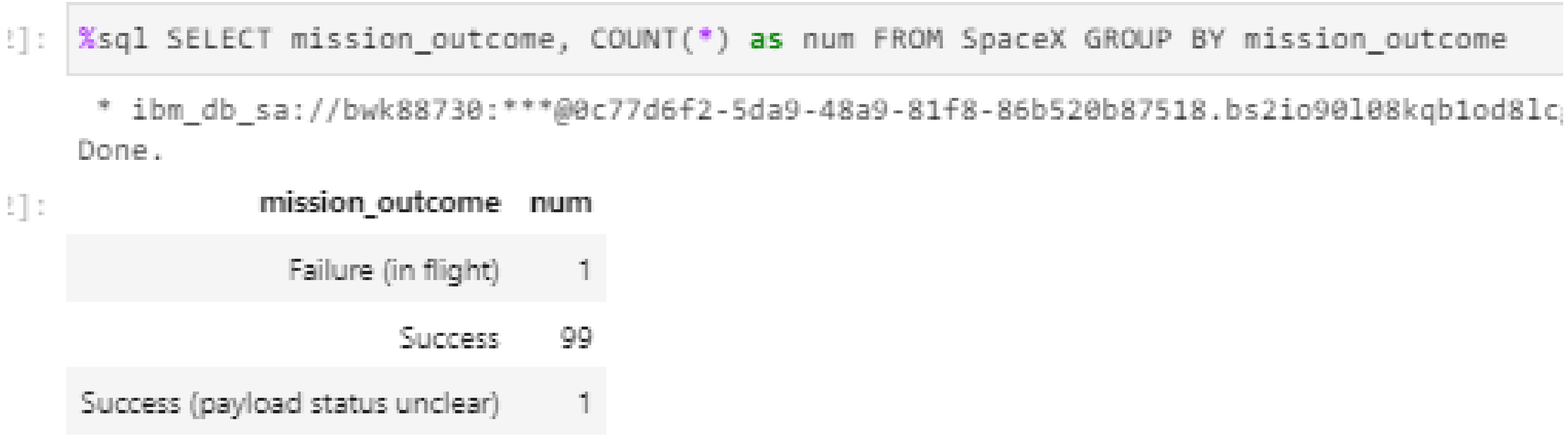
```
[11]: %sql SELECT booster_version FROM SpaceX WHERE mission_outcome = 'Success' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000

 * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

[11]:

booster_version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5B1054
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

The booster versions that are successful and have payload mass greater than 4000 but less than 6000

Total number of successful and failure mission outcomes



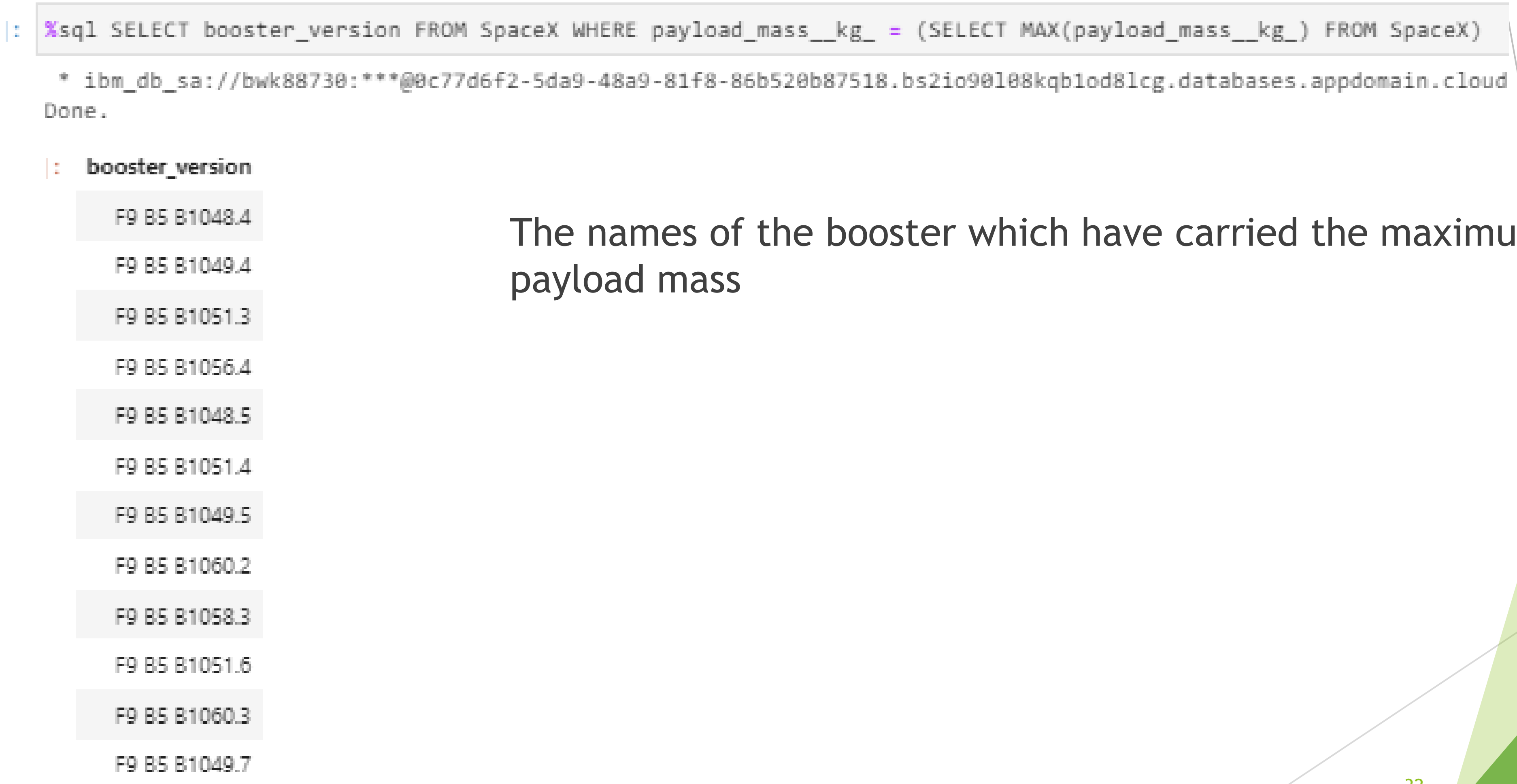
```
%sql SELECT mission_outcome, COUNT(*) as num FROM SpaceX GROUP BY mission_outcome
```

```
 * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lc
Done.
```

mission_outcome	num
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

The number of successful and failure mission outcomes

Boosters carried maximum payload



```
%sql SELECT booster_version FROM SpaceX WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SpaceX)
```

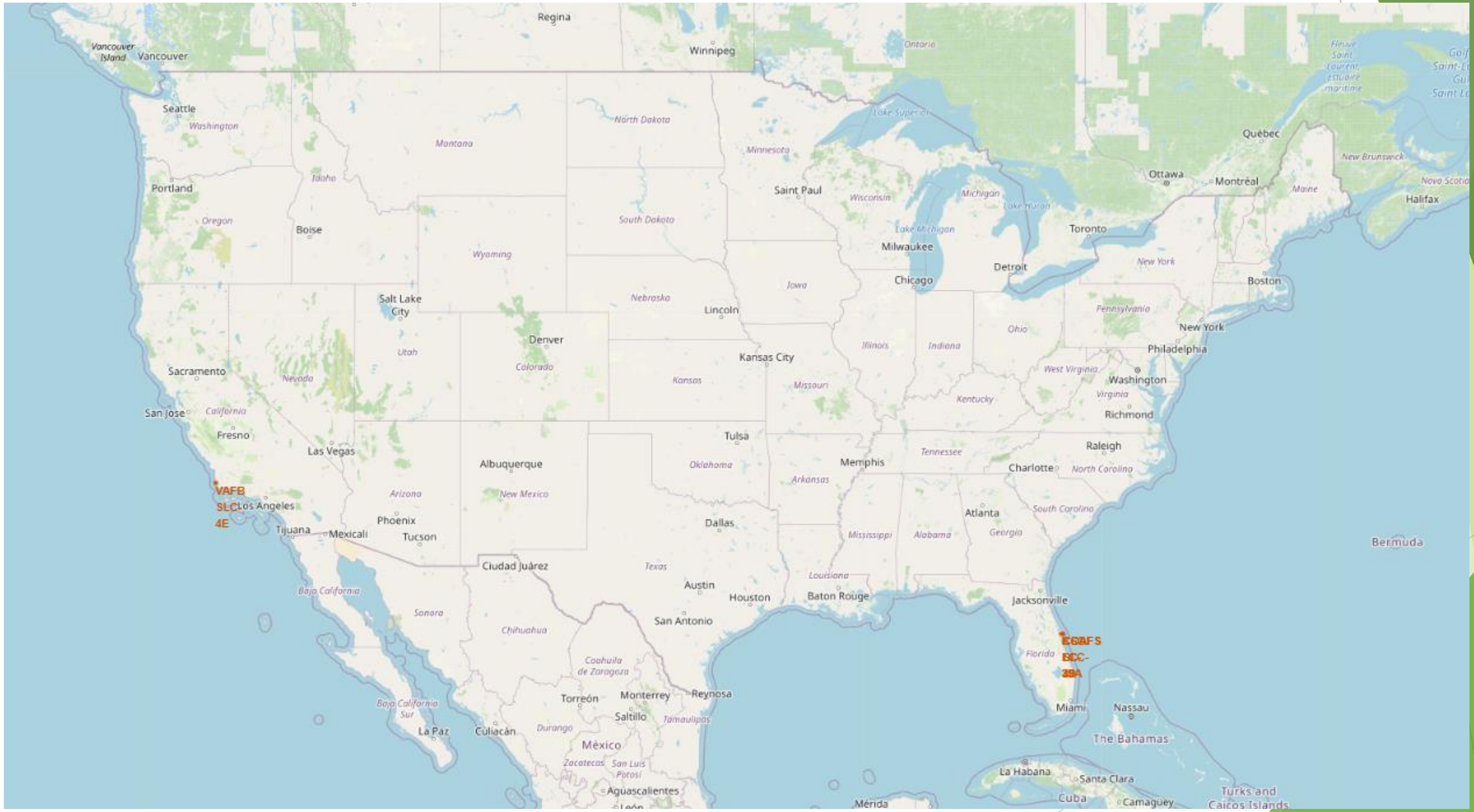
```
 * ibm_db_sa://bwk88730:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud
Done.
```

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

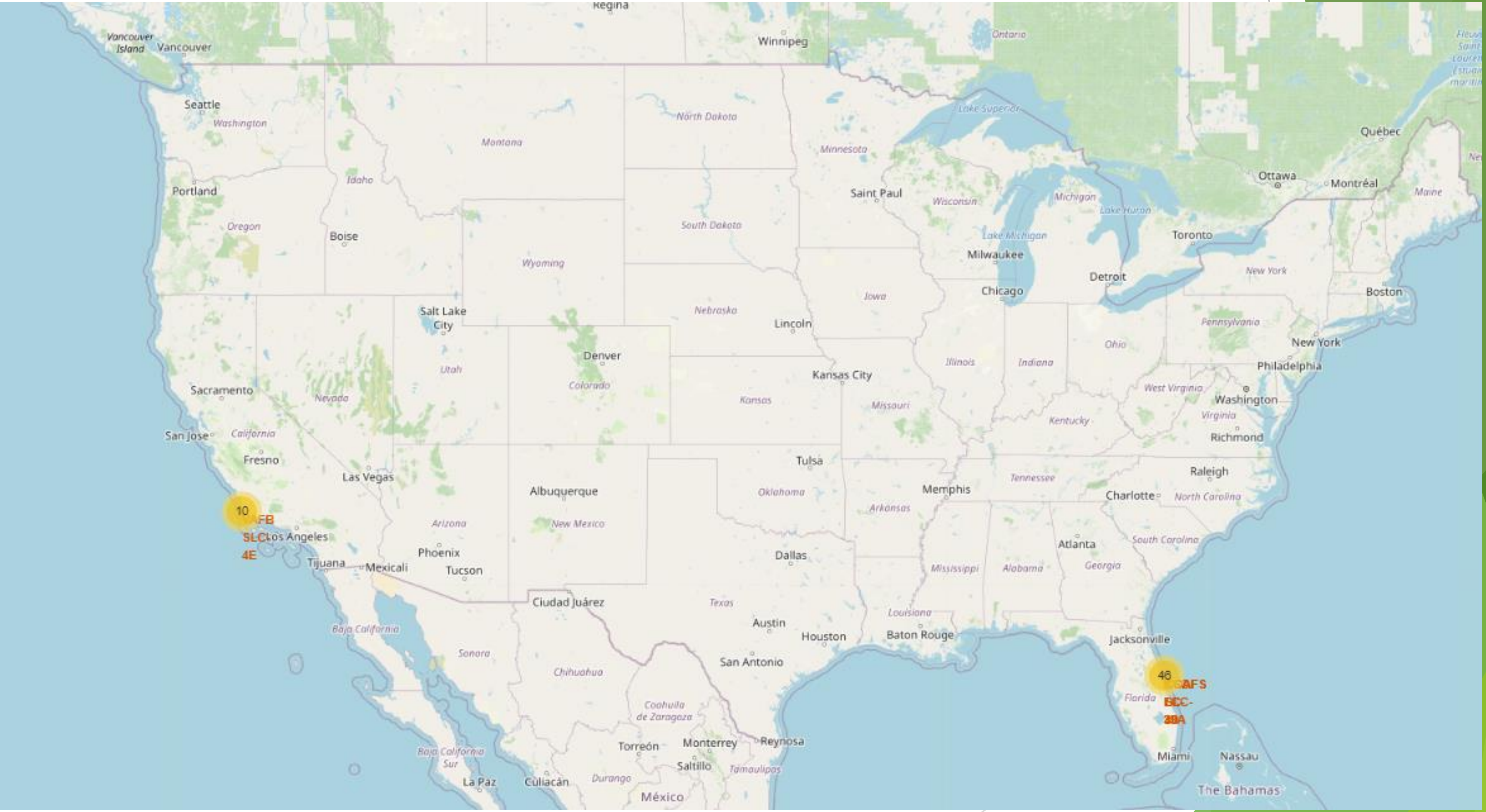
The names of the booster which have carried the maximum payload mass

Interactive map with Folium

Main map

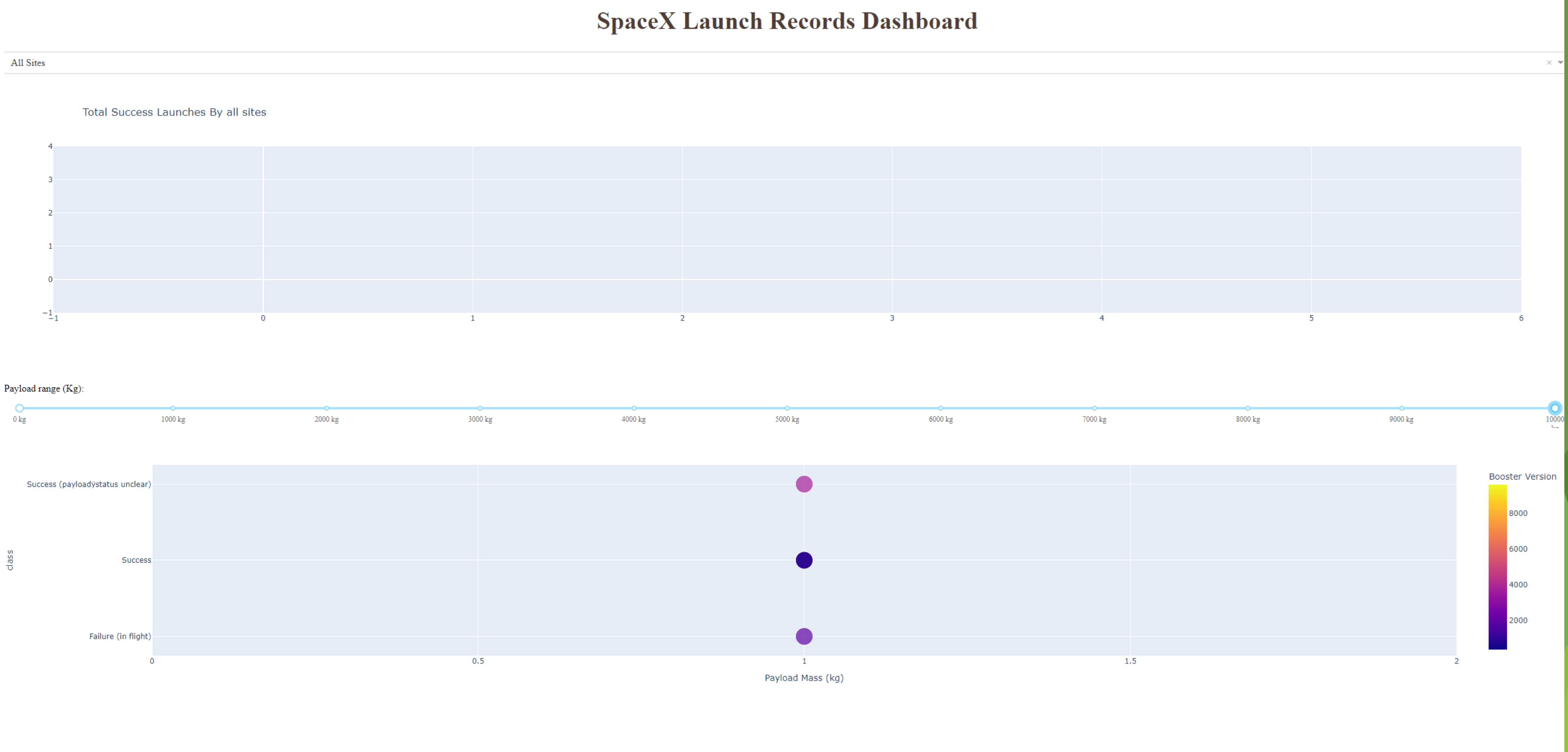

VAFB
SLC-4E
CCAFS
KSC
LC-
39A
Regina
Winnipeg
Vancouver
Seattle
Portland
Boise
Salt Lake City
Denver
Sacramento
San Jose
Fresno
Las Vegas
Los Angeles
Tijuana
Mexicali
Phoenix
Tucson
Albuquerque
Ciudad Juárez
Saint Paul
Lincoln
Kansas City
Tulsa
Dallas
Austin
Houston
San Antonio
Baton Rouge
Memphis
Milwaukee
Chicago
Detroit
Toronto
Ottawa
Montréal
Québec
Halifax
Boston
New York
Philadelphia
Washington
Richmond
Raleigh
Charlotte
Atlanta
Jacksonville
Miami
Nassau
The Bahamas
Bermuda
La Habana
Santa Clara
Cuba
Camagüey
Turks and Caicos Islands
Mérida
Monterrey
Saltillo
Reynosa
Torreón
Culiacán
La Paz
México
Aguascalientes
León

Number of Launchs



Build a Dashboard with Plotly Dash

Dashboard



SpaceX Launch Records Dashboard

All Sites

Total Success Launches By all sites

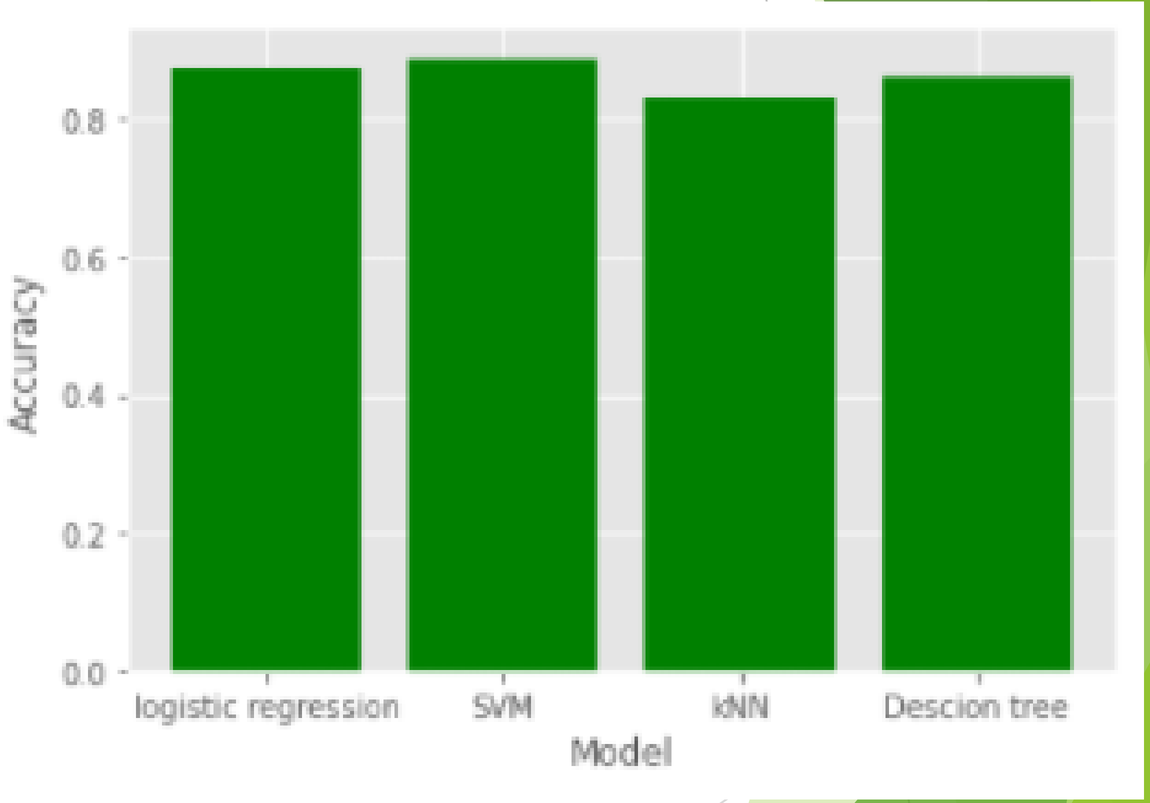
Payload range (Kg):

Predictive analysis (Classification)

Classification Accuracy

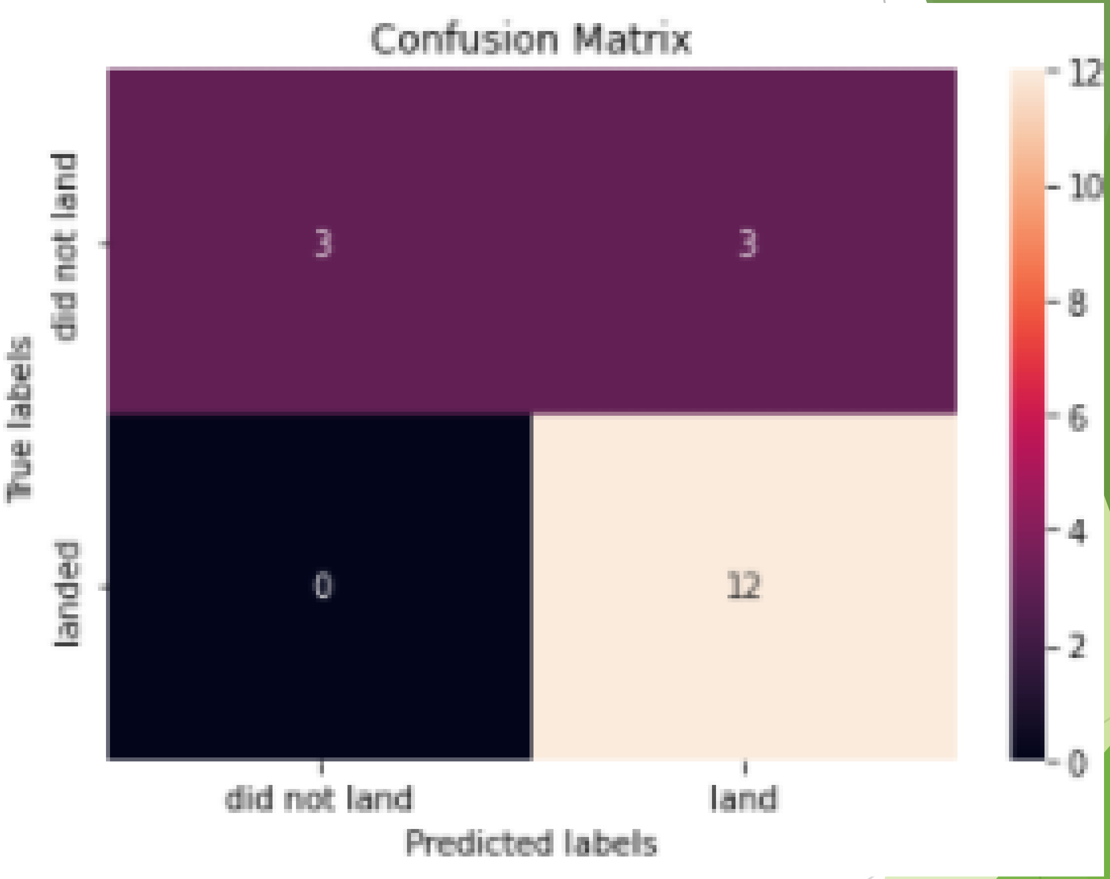
All models have a very similar accuracy score

The best accuracy for the test data is the SVM model so this is what will be used.



Confusion Matrix

This shows that only 3 items where mislabeled as land when that was not true



CONCLUSION

- The SVM model was the best machine learning for this data set
- The success rates for SpaceX is dependent on the year