Data Science Capstone project

By: Eduardo Valtierra Díaz Infante

August 31, 2021

Outline

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

Executive Summary

First, we obtain data based on the **API** process on Xspace webpage and secondly through **webscrapping** the information that is in the Wikipedia Spacex launches web page, those two data groups are the ones we are going to use to obtain the variables and processed information through different methods: complete missing data, fix and clean columns are some steps with **data wrangling.** Then with the **dataframes** created is to analize them with **SQL queries** to obtain very precise information in a hand , and in other hand a **EDA,** this is done through **graphs and plots**, including a **Dashboard**, after the analysis **feature engineering** is performed to select the data that are **representative** for the responders of the main question, the **predictive model**. Meanwhile, **maps** were created to perform a geographic analysis of certain characteristics of the launches to later begin the final process of creating **objects** according to each **classification method**, train, test and compare them to select the most suitable for this case.

Introduction

In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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.

- We want to find data about launches, including information about the rockets and its landing outcome.
- In a second step is to obtain tables, plots and specific data that contain valuable information about launches and sites.
- Our goal is create a classification model to predict whether SpaceX will attempt to land a rocket or not.

Methodology

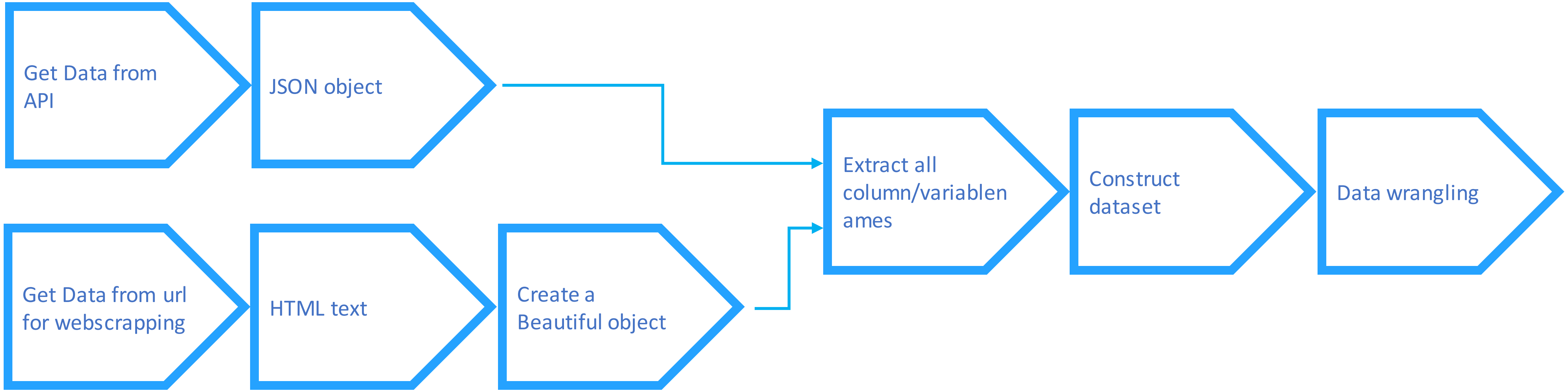
- Data collection methodology:
  - Describe how data were collected
- Perform data wrangling
  - Describe how data were processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Methodology

Data collection

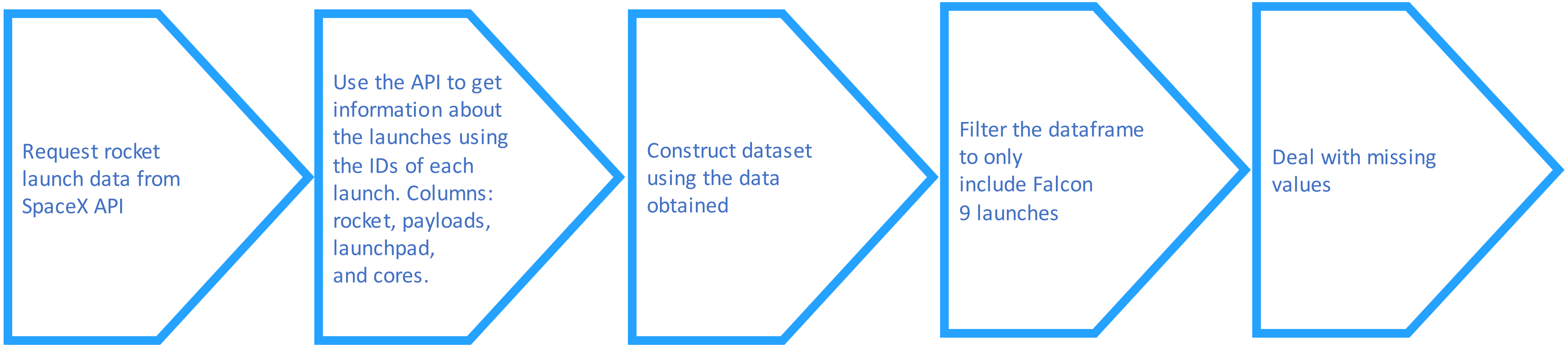
Data was collected by two methods:

Using an API from Xpace webpage and with web scrapping a webpage (wikipedia Xpace webpage) method.



Get Data from API
JSON object
Get Data from url for webscrapping
HTML text
Create a Beautiful object
Extract all column/variablen ames
Construct dataset
Data wrangling

Data collection – SpaceX API



Request rocket launch data from SpaceX API

Use the API to get information about the launches using the IDs of each launch. Columns: rocket, payloads, launchpad, and cores.

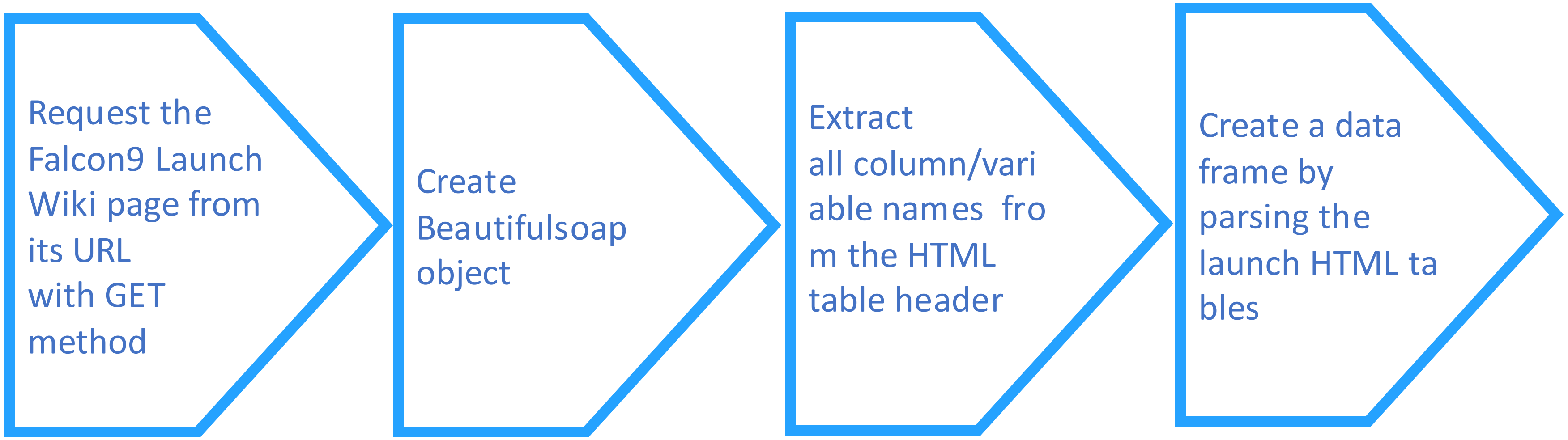
Construct dataset using the data obtained

Filter the dataframe to only include Falcon 9 launches

Deal with missing values

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D1-labs-spacex-data-collection.ipynb

Data collection – Web scraping



Request the Falcon9 Launch Wiki page from its URL with GET method

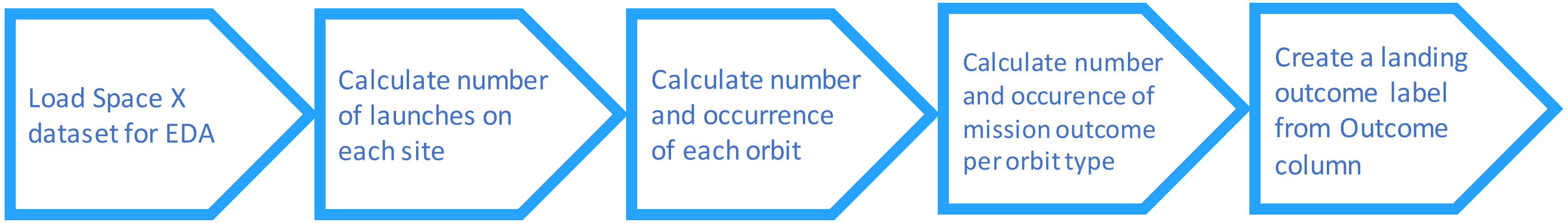
Create Beautifulsoap object

Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D2-labs-webscraping.ipynb

Data wrangling



Load Space X dataset for EDA

Calculate number of launches on each site

Calculate number and occurrence of each orbit

Calculate number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Outcome column will represent the classification variable that represents the outcome of each launch. If the value is '0', the first stage did not land successfully; '1' means the first stage landed successfully.

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D3-spacex-Data%20wrangling.ipynb

EDA with data visualization

8 charts and plots were created:

- A **scatter plot** to see how the FlightNumber (indicating the continuous launch attempts) and Payload variables would affect the launch outcome
- A **scatter plot** to visualize the relationship between Flight Number and Launch Site
- A **scatter plot** to visualize the relationship between Payload and Launch Site
- A **bar chart** to visualize the relationship between success rate of each orbit type
- A **scatter plot** to visualize the relationship between FlightNumber and orbit type
- A **scatter plot** to visualize the relationship between Payload and orbit type
- A **line plot** visualize the launch success yearly trend

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D5-jupyter-labs-eda-dataviz.ipynb

EDA with SQL

10 Tasks with a SQL Query each one:

- The names of the unique launch sites in the space mission
- The 5 records where launch sites begin with the string 'CCA'
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- List of date when the first successful landing outcome in ground pad was achieved
- List of names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List of total number of successful and failure mission outcomes
- List of names of the booster_versions which have carried the maximum payload mass, with a subquery
- List of failed landing_outcomes in drone ship, their booster versions, and launch site names for the year 2015
- Rank of 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

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D4-labs-eda-sql-coursera.ipynb

Build an interactive map with Folium

A **circle** and a **maker objects** for each Launch site (list of 4 sites) were created to see its exact position, a **maker cluster** for each site with 56 launches grouped within to show the real distribution of launches for site, a **mouse position object** to show coordinates of the mouse pointer and with these create **line objects** to display the distance to each of the 4 points (road, train road, coast and city) with a **maker** showing distances to a launch site.

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ebcff91e29445ac9370ff440de53839c2bceec7e/D6-

lab_jupyter_launch_site_location.ipynb

and with nbviewer to view maps:

https://nbviewer.jupyter.org/github/lalovaltierra/AppliedDSCapstone/blob/ebcff91e29445ac9370ff440de53839c2bceec7e/D6-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Charts added are 2 and interactions components are 2:

Drop-down Input to select **Launch site**, **Range Slider** to Select **Payload** range, a callback function to render a **success-pie chart** based on selected site dropdown, and a callback function to render a **success vs payload-scatter plot.**

For Drop-down input:

- If **all sites** are selected, we will use all rows to render and return a pie chart graph to show the **total success** launches.
- If a **specific launch site** is selected, you need to filter the data to include only data for the selected site. Then, render and return a pie chart graph to show the **success count and failed count** for the **selected site**.

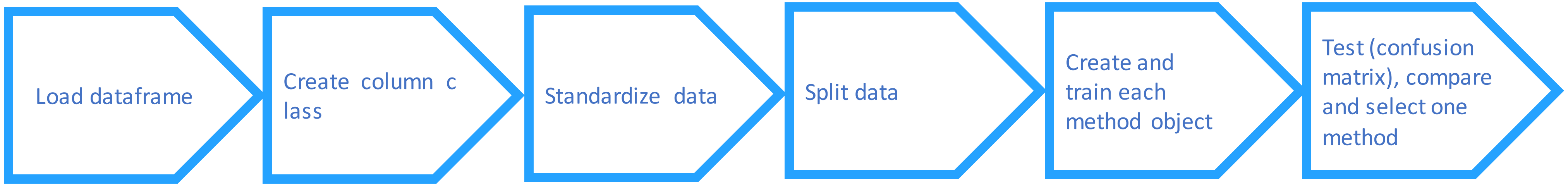
For range slider

- If **all sites** are selected, render a scatter plot to display all **values for payload mass and class**. In addition, the point **color** needs to be set to the **booster version**.
- If a **specific launch s**ite is selected, you need to filter data, and render a scatter chart to show **values payload mass and class** for the **selected site**, and **color**-label the point using **booster version** likewise.

URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D7-spacex_dash_app.py

Predictive analysis (Classification)

First, the necessary information is obtained, then it´s processed so that it has the ideal characteristics to work on it, it´s split into the training data and the testing data. The objects of each of the models proposed to be used are created: **KNN, SVM, Classification Trees and Logistic Regression**. Tools are used to obtain the **hyperparameters** and it´s trained with the training data. Once the ideal parameters have been obtained, the object is tested with the testing data and a **confusion matrix** is generated for each one. The accuracy´s levels and the information of the matrixes will be compared to select the most efficient method for this case.


Load dataframe
Create column c lass
Standardize data
Split data
Create and train each method object
Test (confusion matrix), compare and select one method

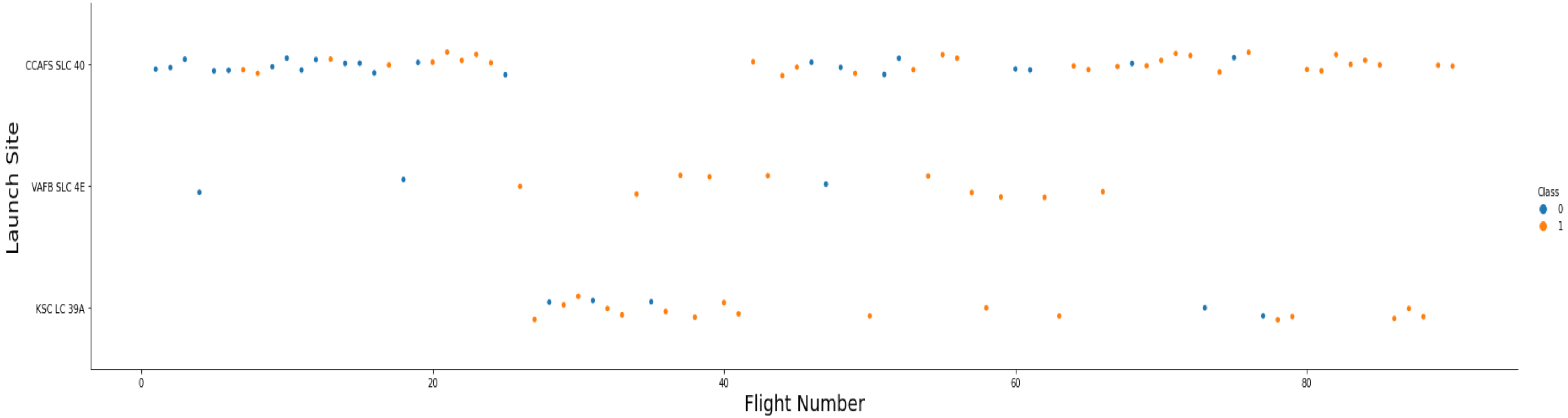
URL: https://github.com/lalovaltierra/AppliedDSCapstone/blob/ec6f6cf40805c203ec714a09a63aad8588217969/D7-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

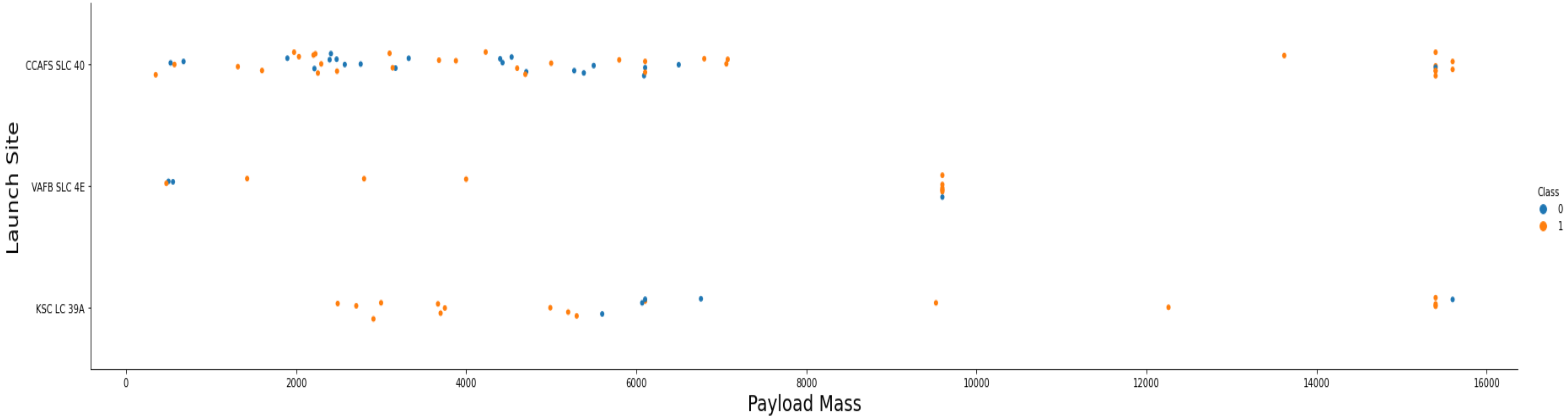
EDA with Visualization

Flight Number vs. Launch Site



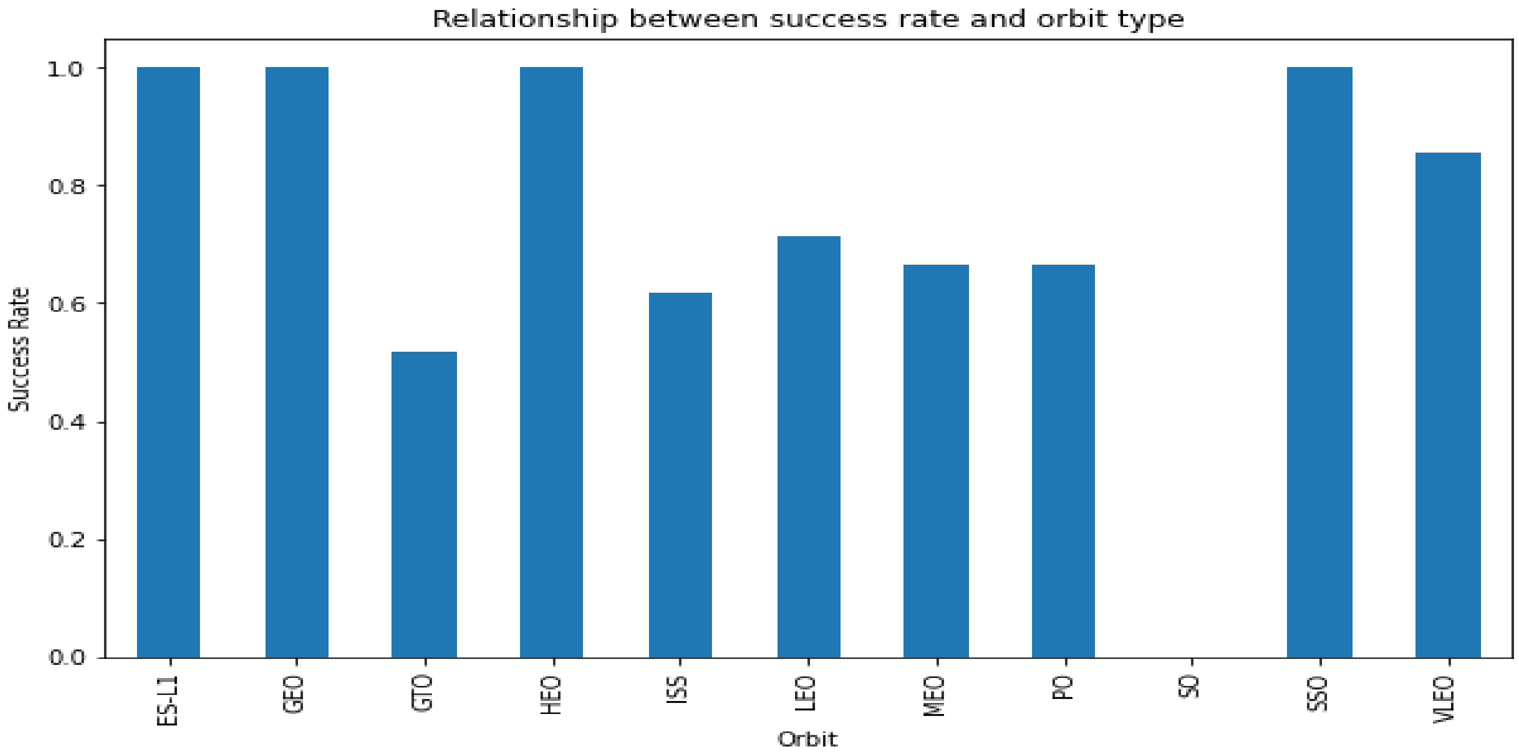
The plot shows that the CCAFS LC-40 site is the one with the highest number of launches.

Payload vs. Launch Site



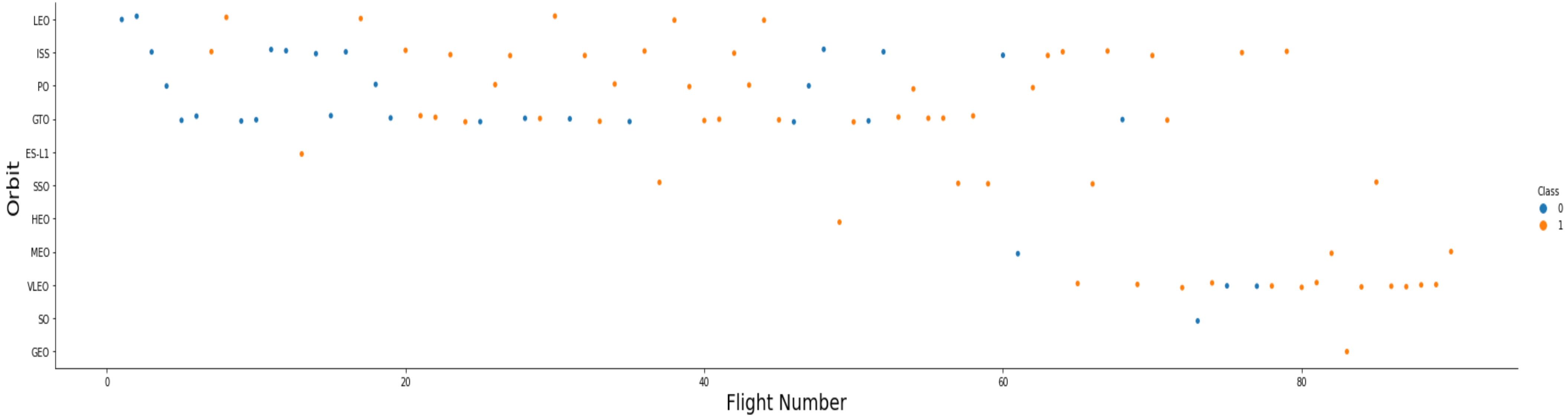
The plot shows that the CCAFS LC-40 site is the one with the highest number of launches with the highest load. Most launches have lower payload mass.

Success rate vs. Orbit type



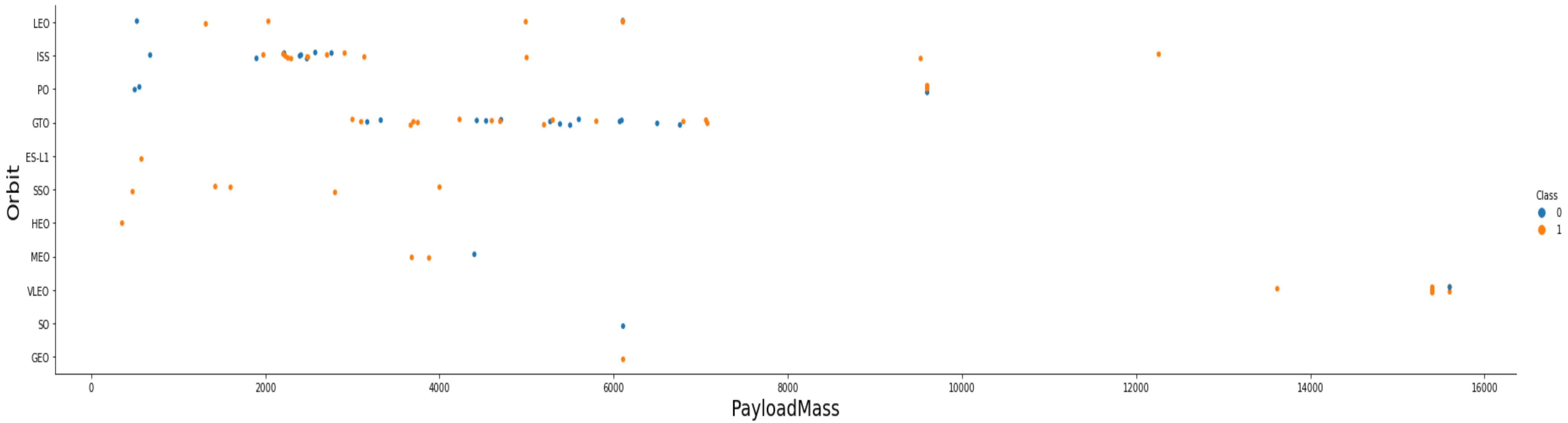
The graph shows there are 4 of 11 orbits where the launches are 100% successful, most launches with a success greater than 50% and only one orbit has 0% successful launches.

Flight Number vs. Orbit type



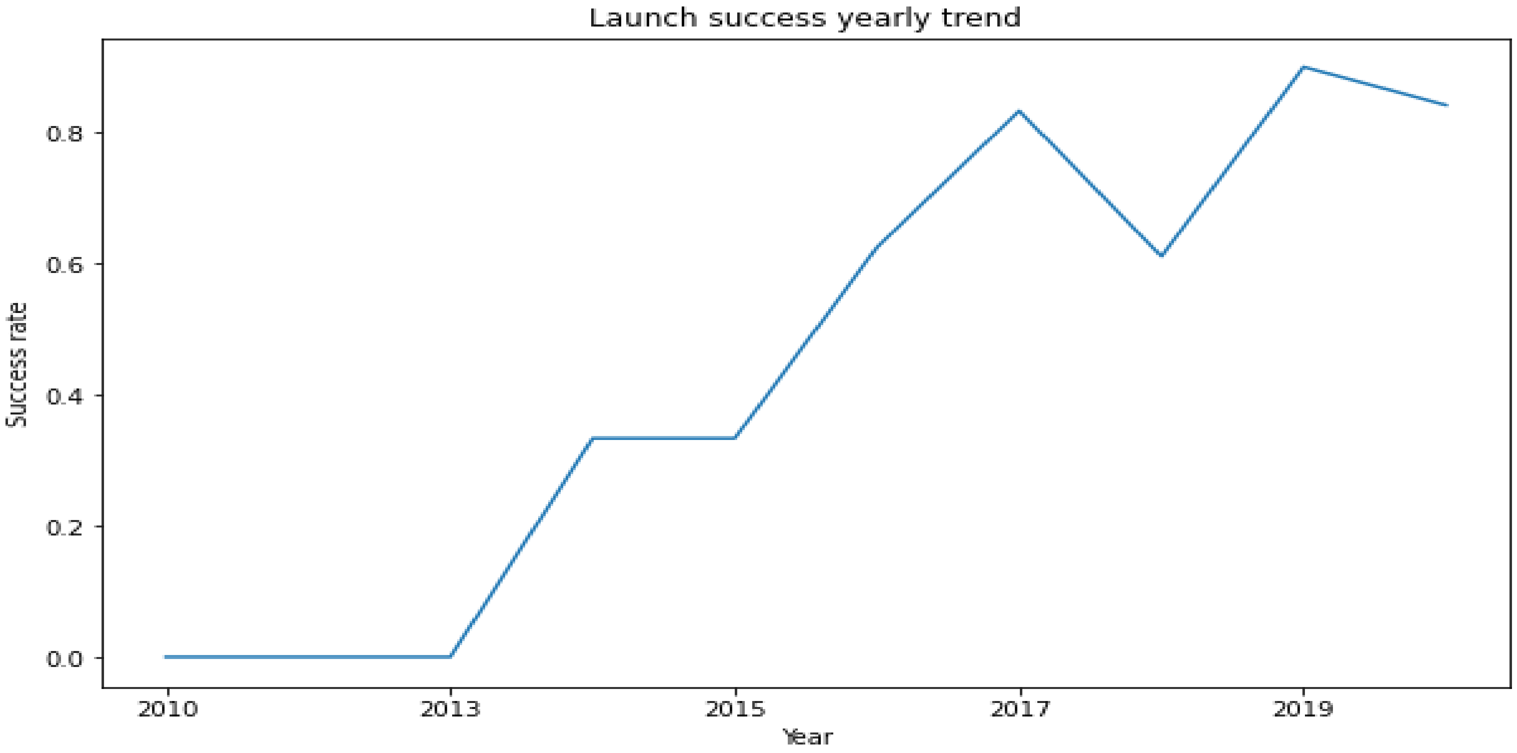
The plot shows that most of the launches have been in 4 orbits and the majority were successful, and around flight number 60 other orbits began to be used, one in particular VLEO.

Payload vs. Orbit type



The plot shows that the VLEO orbit is the one that has launches with the greatest payload mass, in general the payload mass is distributed equally or at least in ranges versus the orbit, with exception of orbits with few launches.

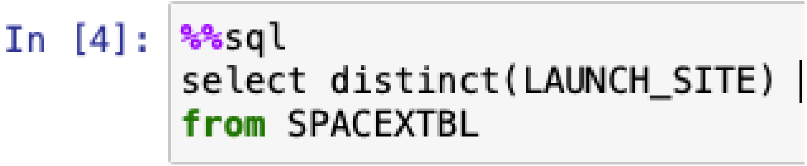
Launch success yearly trend



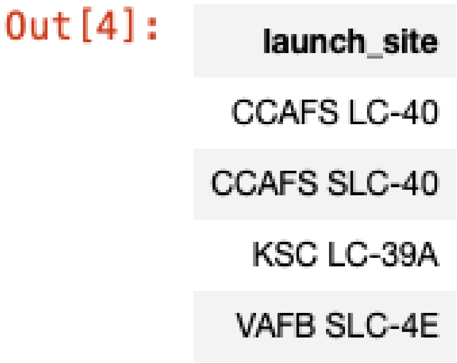
The graph shows that the success of the launches has been increasing year after year, even though in the year 2017-2018 there were failed launches and later the graph was seized again.

EDA with SQL

All launch site names



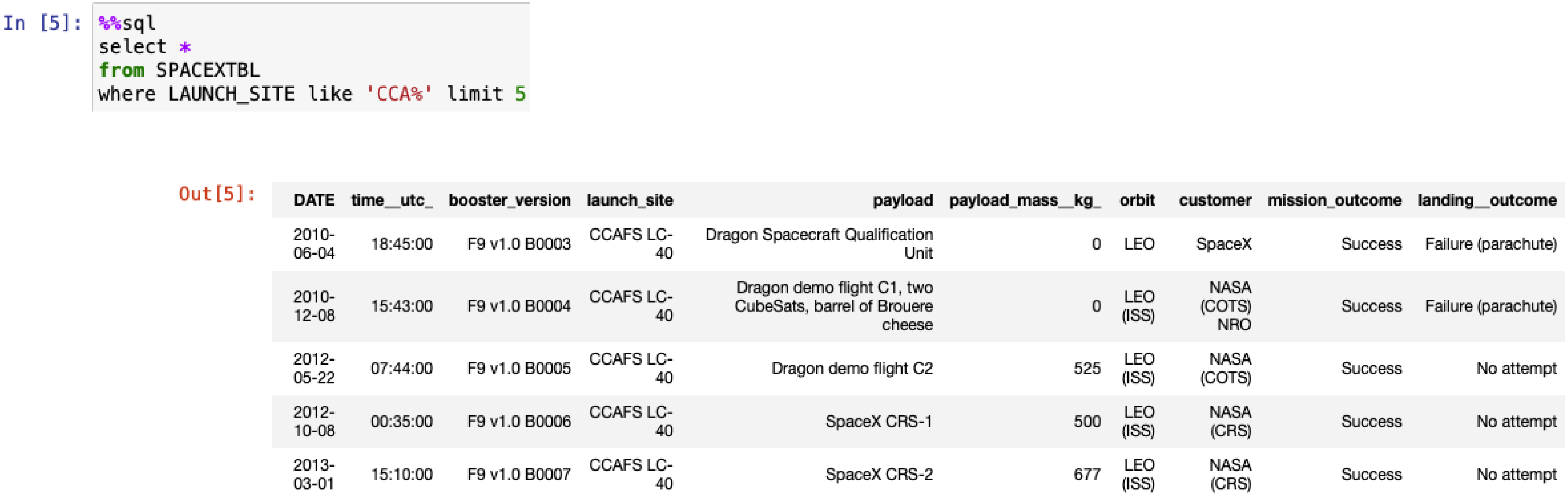
```
In [4]: %%sql
        select distinct(LAUNCH_SITE)
        from SPACEXTBL
```



```
Out[4]:   launch_site
          CCAFS LC-40
          CCAFS SLC-40
          KSC LC-39A
          VAFB SLC-4E
```

There are only 4 launch sites.

Launch site names begin with `CCA`



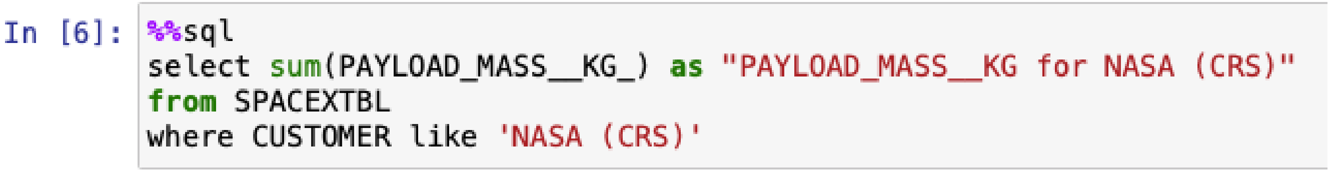
```
In [5]: %%sql
        select *
        from SPACEXTBL
        where LAUNCH_SITE like 'CCA%' limit 5
```

Out[5]:

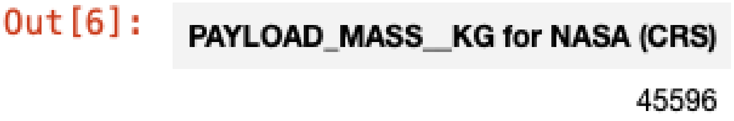
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

5 launches made from sites whose name begins with CCA are shown, all with successful mission and with LEO orbit.

Total payload mass



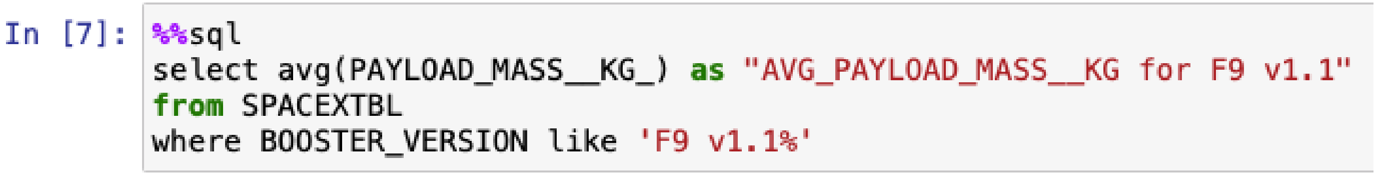
```
In [6]: %%sql
        select sum(PAYLOAD_MASS__KG_) as "PAYLOAD_MASS__KG for NASA (CRS)"
        from SPACEXTBL
        where CUSTOMER like 'NASA (CRS)'
```



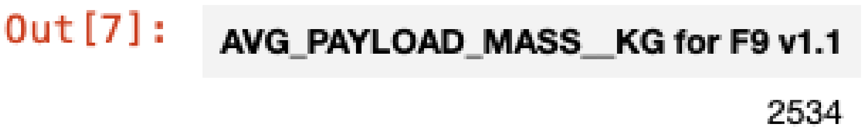
```
Out[6]: PAYLOAD_MASS__KG for NASA (CRS)
                                  45596
```

The total payload mass launched for NASA is shown.

Average payload mass by F9 v1.1



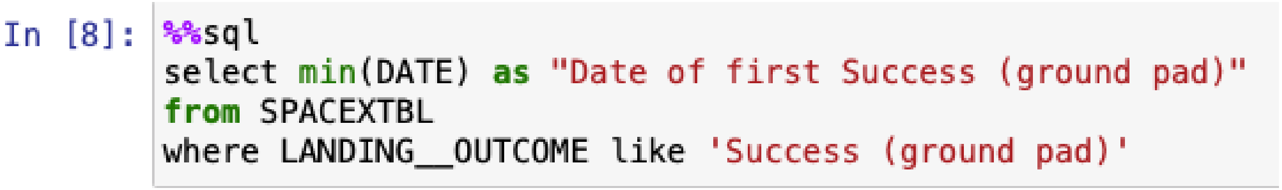
```
In [7]: %%sql
        select avg(PAYLOAD_MASS__KG_) as "AVG_PAYLOAD_MASS__KG for F9 v1.1"
        from SPACEXTBL
        where BOOSTER_VERSION like 'F9 v1.1%'
```



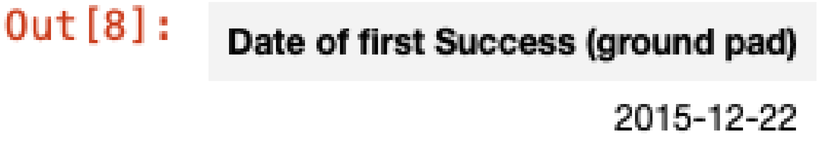
```
Out[7]: AVG_PAYLOAD_MASS__KG for F9 v1.1
                                    2534
```

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

First successful ground landing date



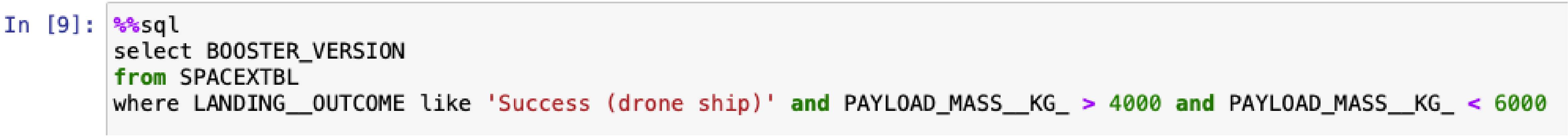
```
In [8]: %%sql
        select min(DATE) as "Date of first Success (ground pad)"
        from SPACEXTBL
        where LANDING__OUTCOME like 'Success (ground pad)'
```



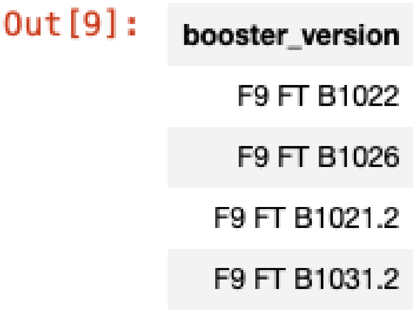
```
Out[8]: Date of first Success (ground pad)
                                2015-12-22
```

The date of the first lauch ground landing with success.

Successful drone ship landing with payload between 4000 and 6000



```
In [9]: %%sql
        select BOOSTER_VERSION
        from SPACEXTBL
        where LANDING__OUTCOME like 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

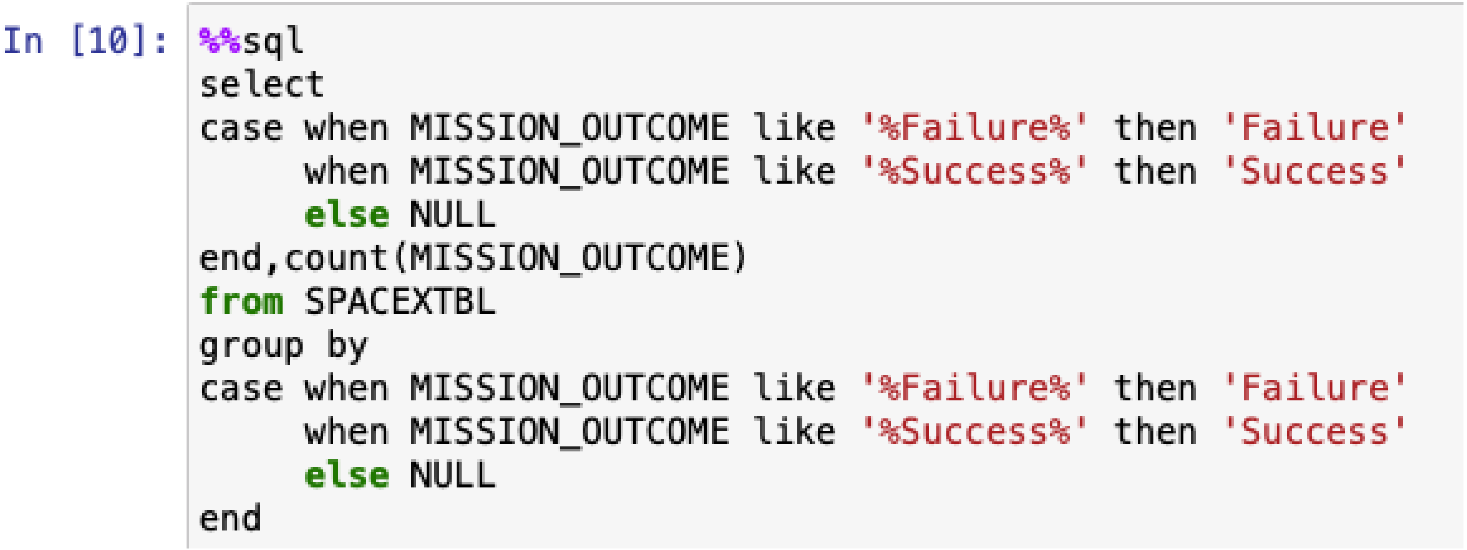


Out[9]:

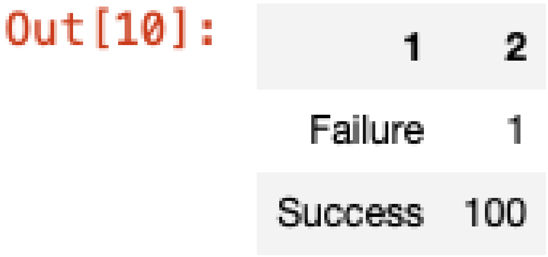
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

These are booster versions in launches with droneship landing success with payload mass between 4000 and 6000 kg. There are just 4 versions.

Total number of successful and failure mission outcomes



```
In [10]: %%sql
select
case when MISSION_OUTCOME like '%Failure%' then 'Failure'
     when MISSION_OUTCOME like '%Success%' then 'Success'
     else NULL
end,count(MISSION_OUTCOME)
from SPACEXTBL
group by
case when MISSION_OUTCOME like '%Failure%' then 'Failure'
     when MISSION_OUTCOME like '%Success%' then 'Success'
     else NULL
end
```

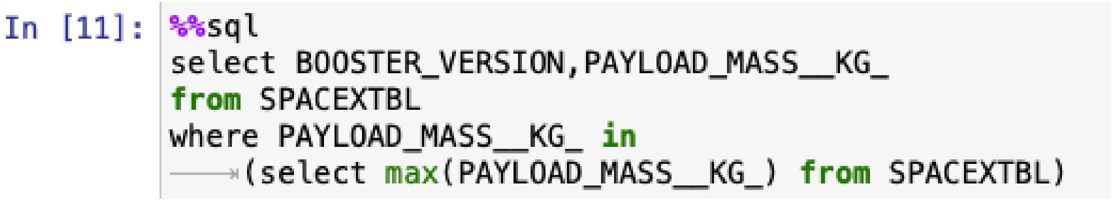


Out[10]:

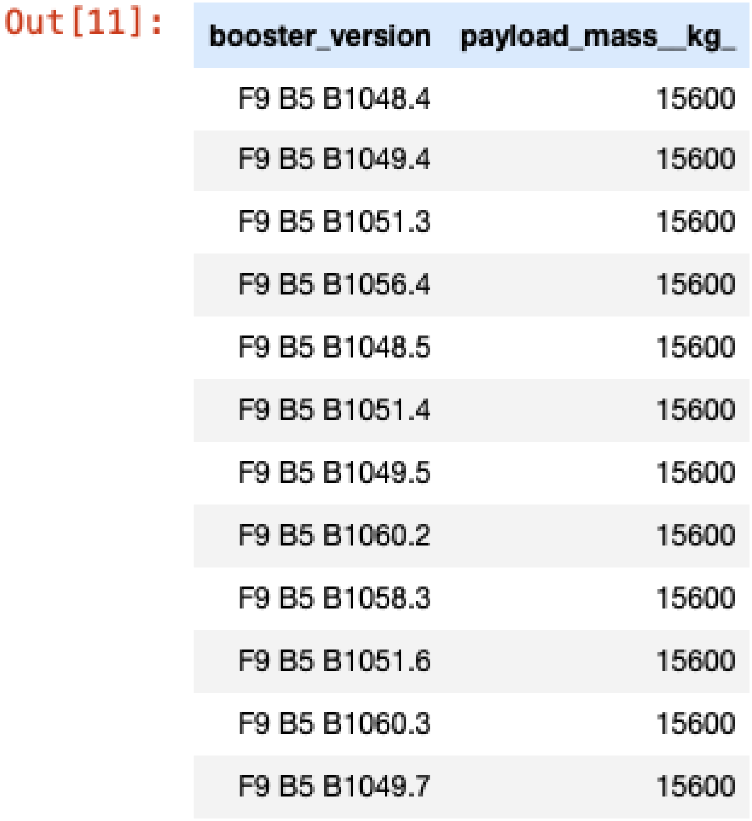
1	2
Failure	1
Success	100

The data shows that there are 100 successful mission outcomes and only 1 failed.

Boosters carried maximum payload



```
In [11]: %%sql
         select BOOSTER_VERSION,PAYLOAD_MASS__KG_
         from SPACEXTBL
         where PAYLOAD_MASS__KG_ in
             (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

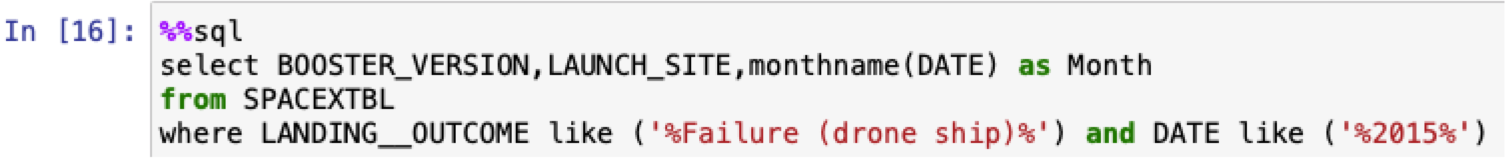


Out[11]:

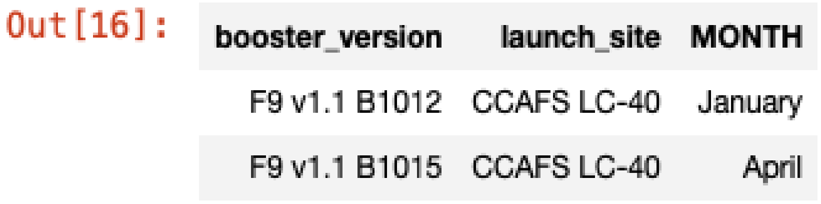
booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

The data shows that there are 12 boosters that carried the maximum payload mass (15600 kg)

2015 launch records



```
In [16]: %%sql
         select BOOSTER_VERSION,LAUNCH_SITE,monthname(DATE) as Month
         from SPACEXTBL
         where LANDING__OUTCOME like ('%Failure (drone ship)%') and DATE like ('%2015%')
```

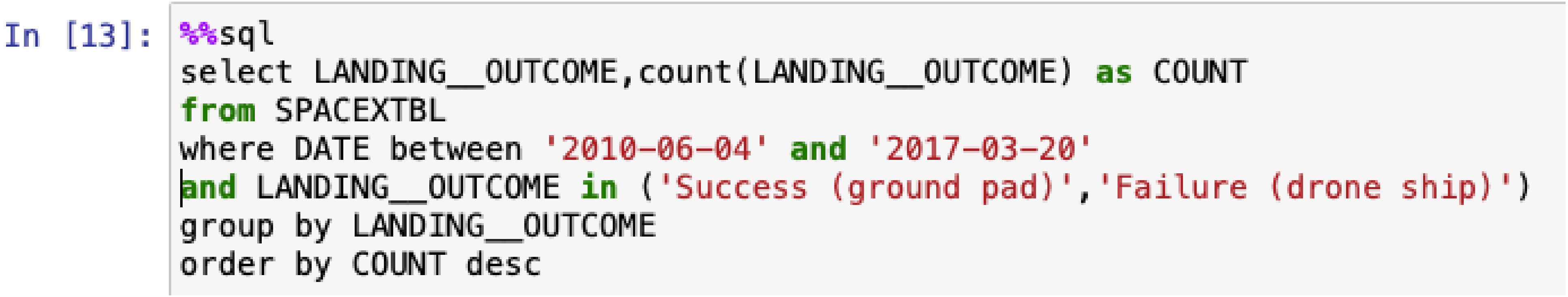


Out[16]:

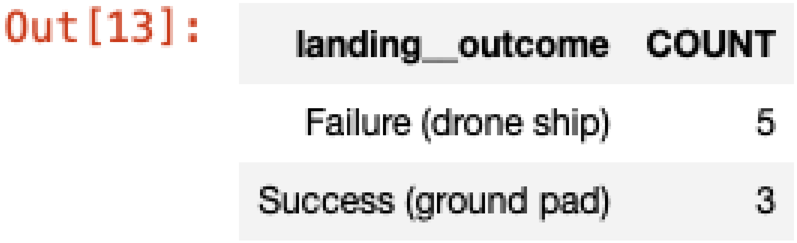
booster_version	launch_site	MONTH
F9 v1.1 B1012	CCAFS LC-40	January
F9 v1.1 B1015	CCAFS LC-40	April

In January and April there were a couple of launches(site and booster version) with failed landing_outcomes in drone ship.

Rank success count between 2010-06-04 and 2017-03-20



```
In [13]: %%sql
         select LANDING__OUTCOME,count(LANDING__OUTCOME) as COUNT
         from SPACEXTBL
         where DATE between '2010-06-04' and '2017-03-20'
         and LANDING__OUTCOME in ('Success (ground pad)','Failure (drone ship)')
         group by LANDING__OUTCOME
         order by COUNT desc
```



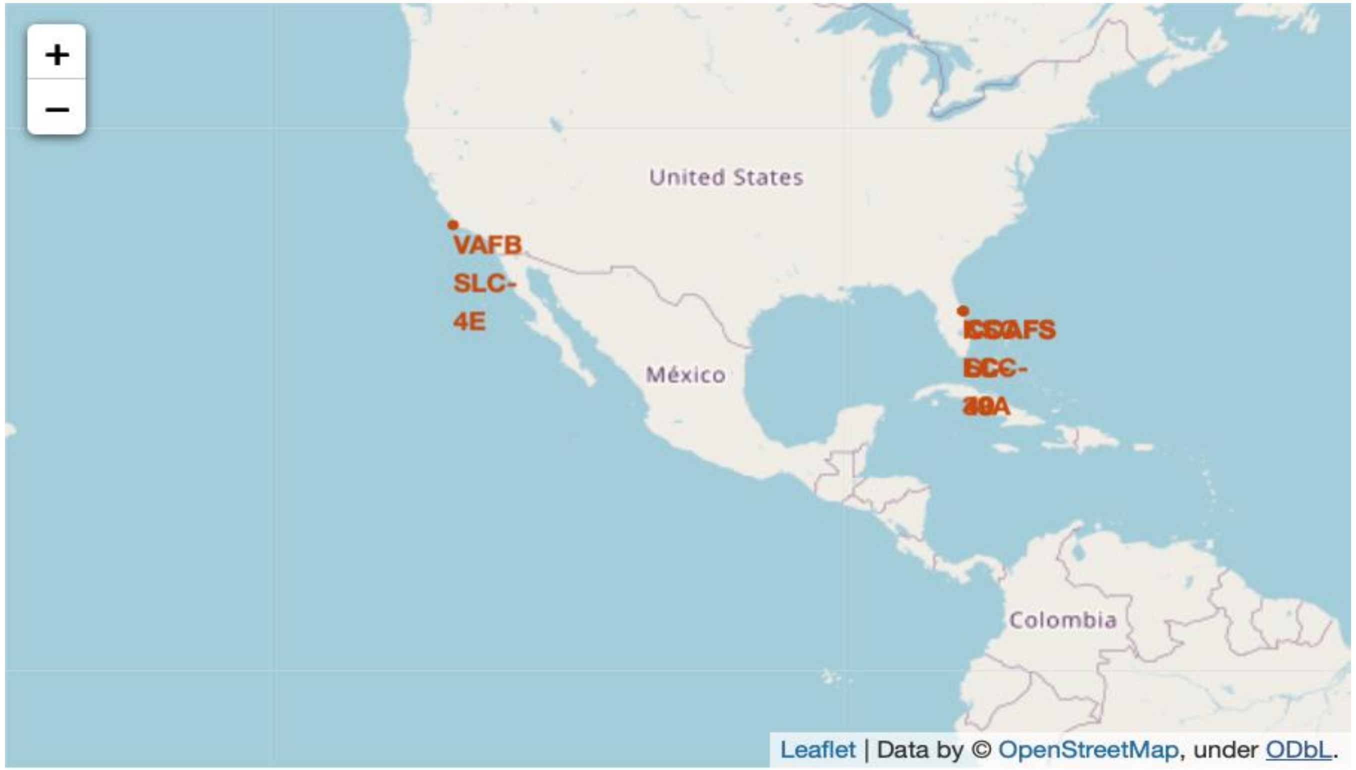
Out[13]:

landing__outcome	COUNT
Failure (drone ship)	5
Success (ground pad)	3

There are more failed landing outcomes(drone ship), 5, than successful (ground pad), 3, between 2010-06-04 and 2017-03-20.

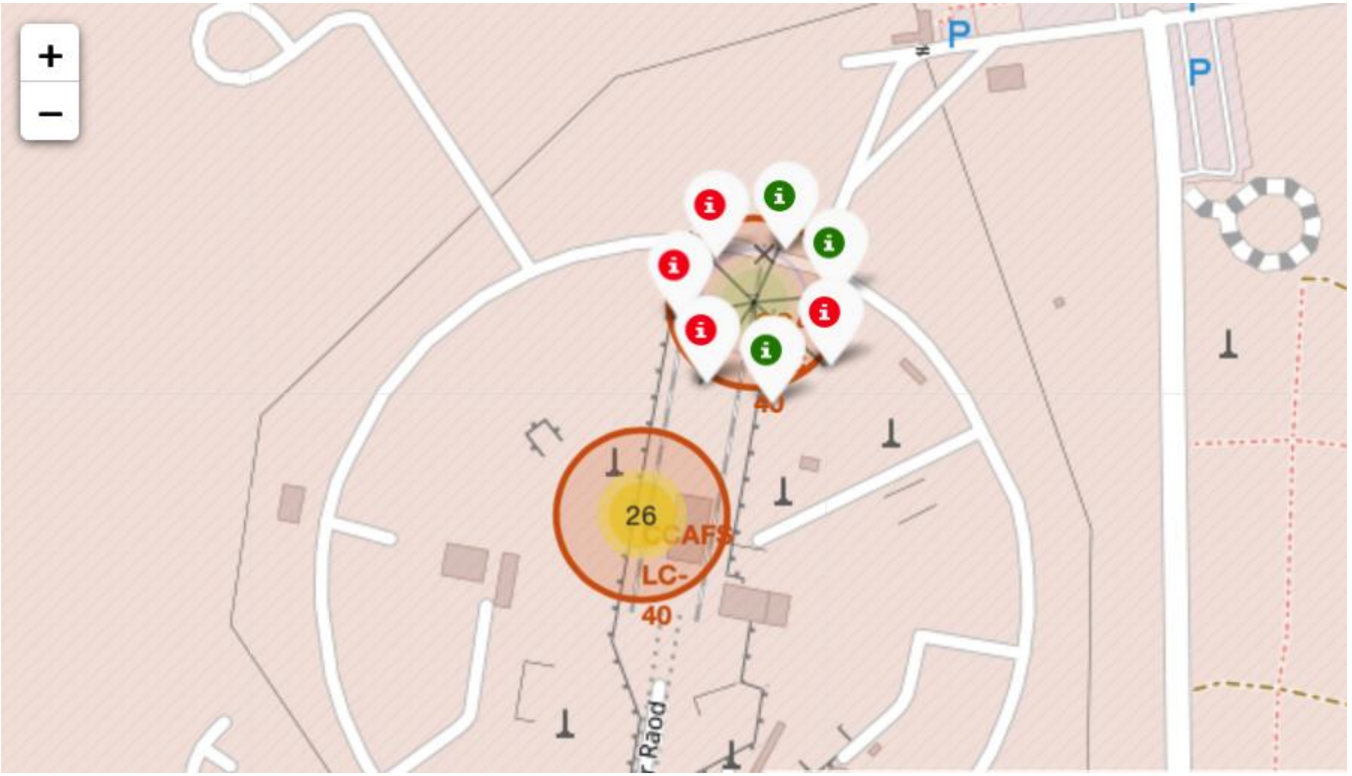
Interactive map with Folium

Launch sites and Equator line



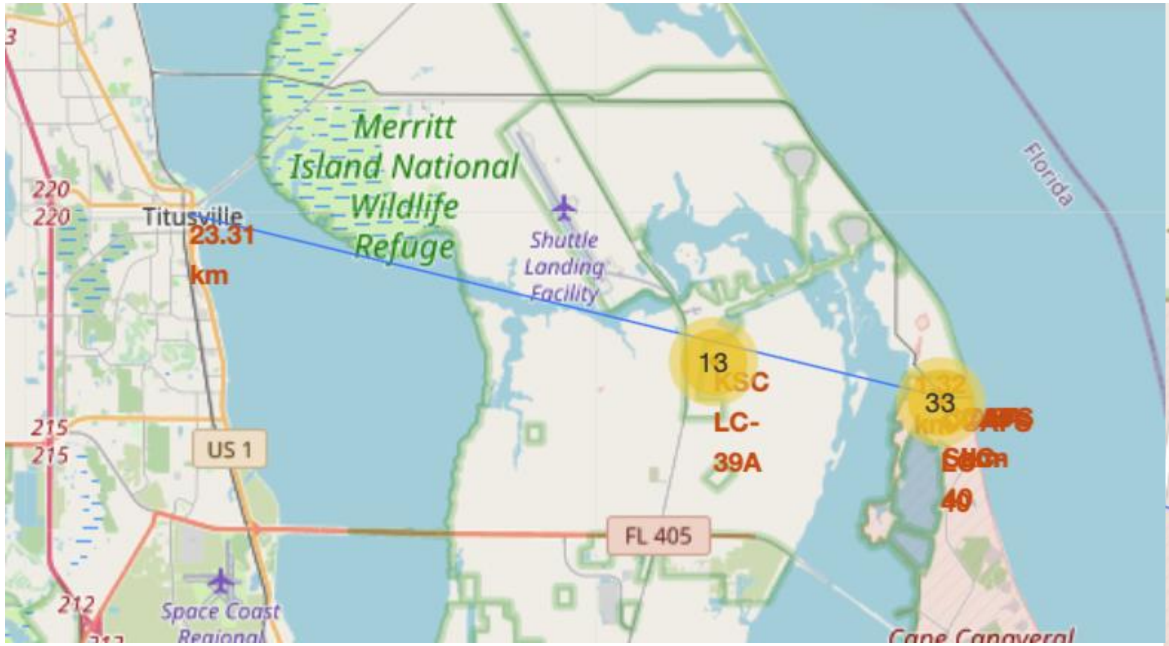
The sites are close to the coasts at both ends of the US, as far south as possible within its territory and therefore relatively close to the Equator line.

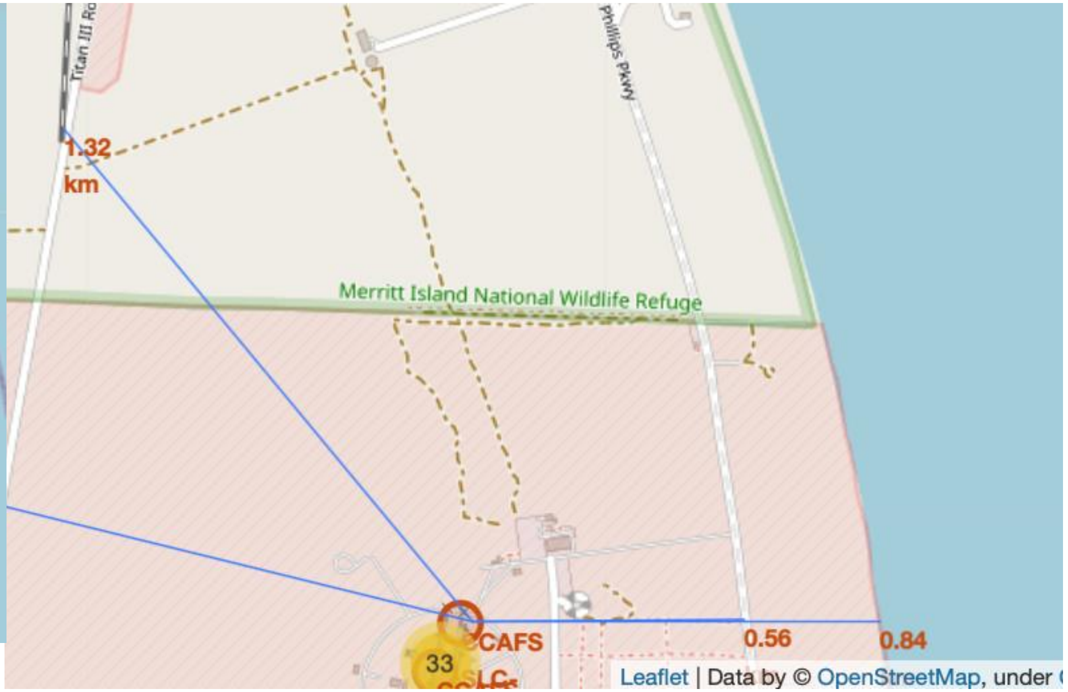
Successful Launch site



The Launches site with the most successful launches is CCAFS SLC-40, with 26.

Points of interest near Launch Site

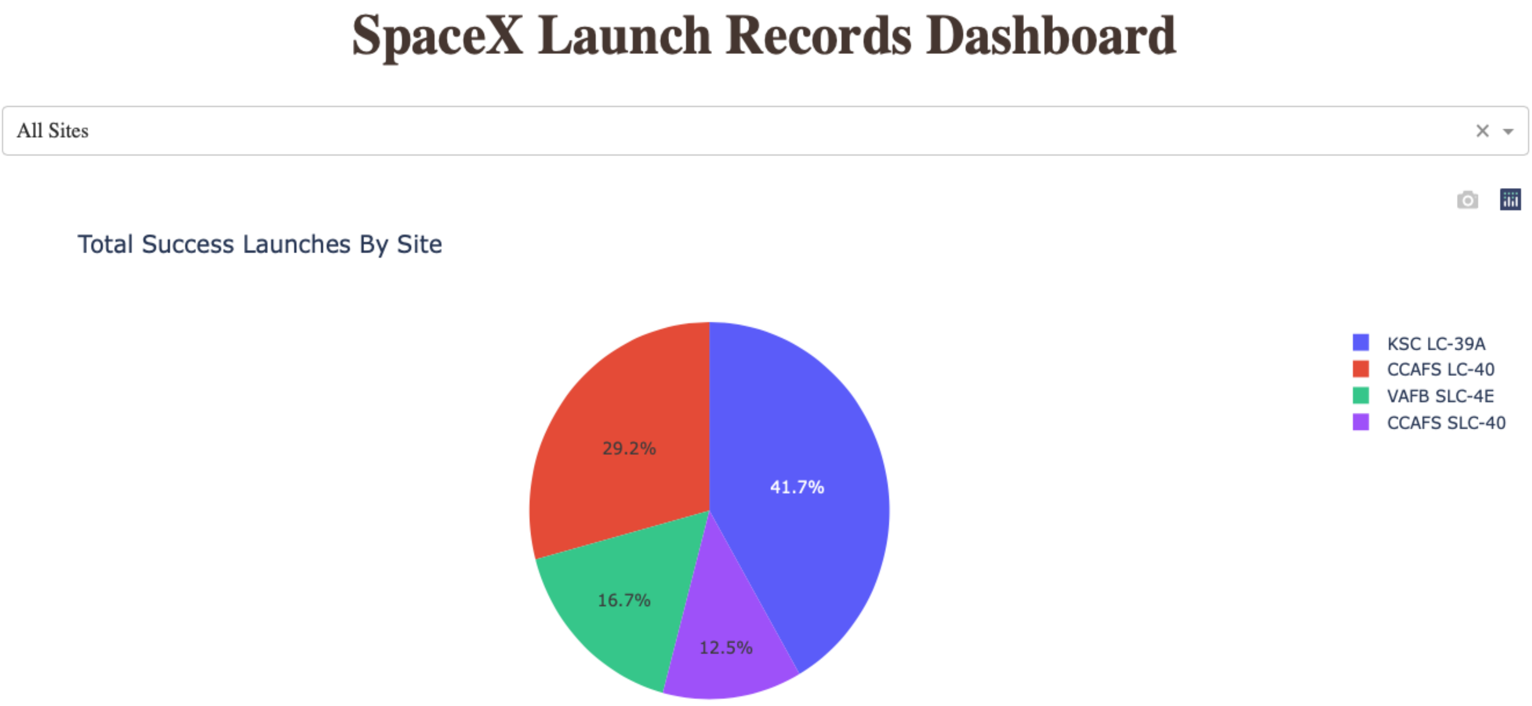



Titan III Rd
Phillips Pkwy
1.32 km
Merritt Island National Wildlife Refuge
CAFS
0.56
0.84
33
Leaflet | Data by © OpenStreetMap, under

It can be seen that the communication routes such as roads, railways and the coast are relatively close to the sites, less than 1km, however cities and centers with high population density are at a very great distance, in this case more than 20km.

Build a Dashboard with Plotly Dash

Launch success in ALL sites



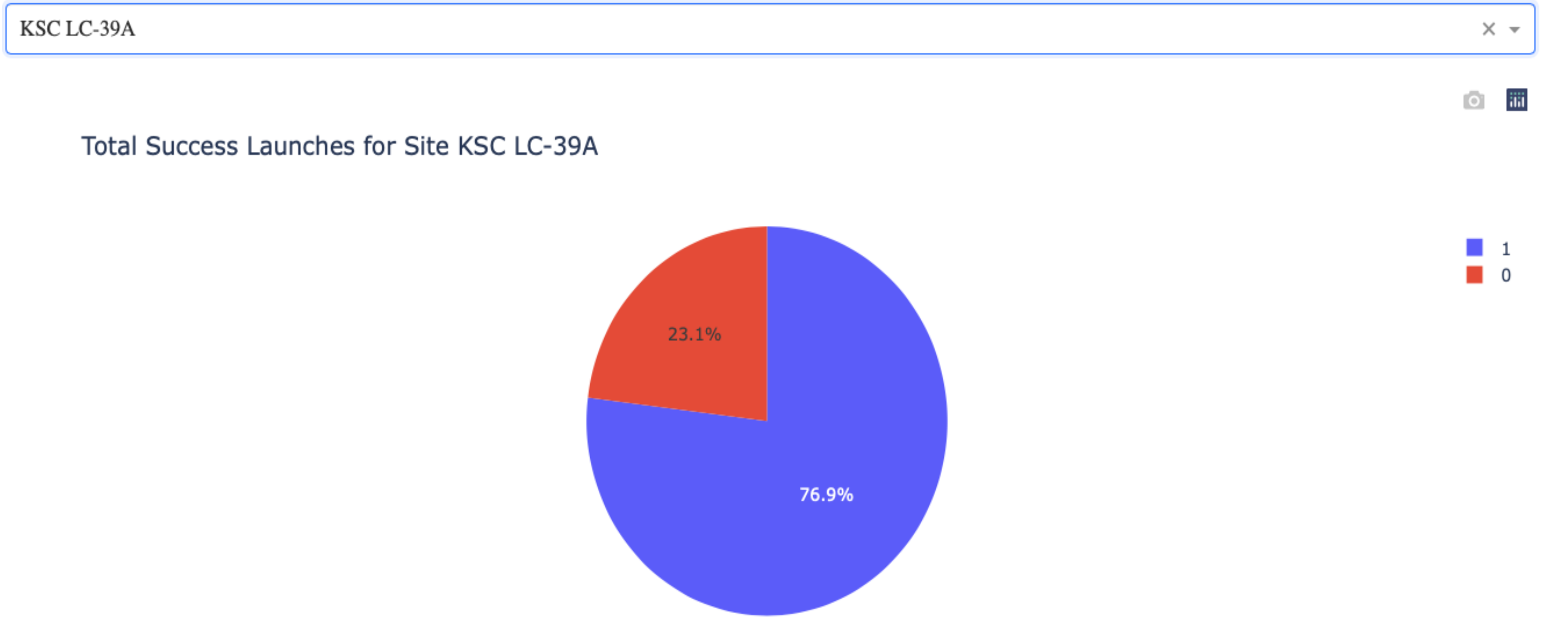
SpaceX Launch Records Dashboard

All Sites

Total Success Launches By Site

Of the 4 sites, KSC LC-39A is the one with the highest number of successful launches, with more than 41% of the total.

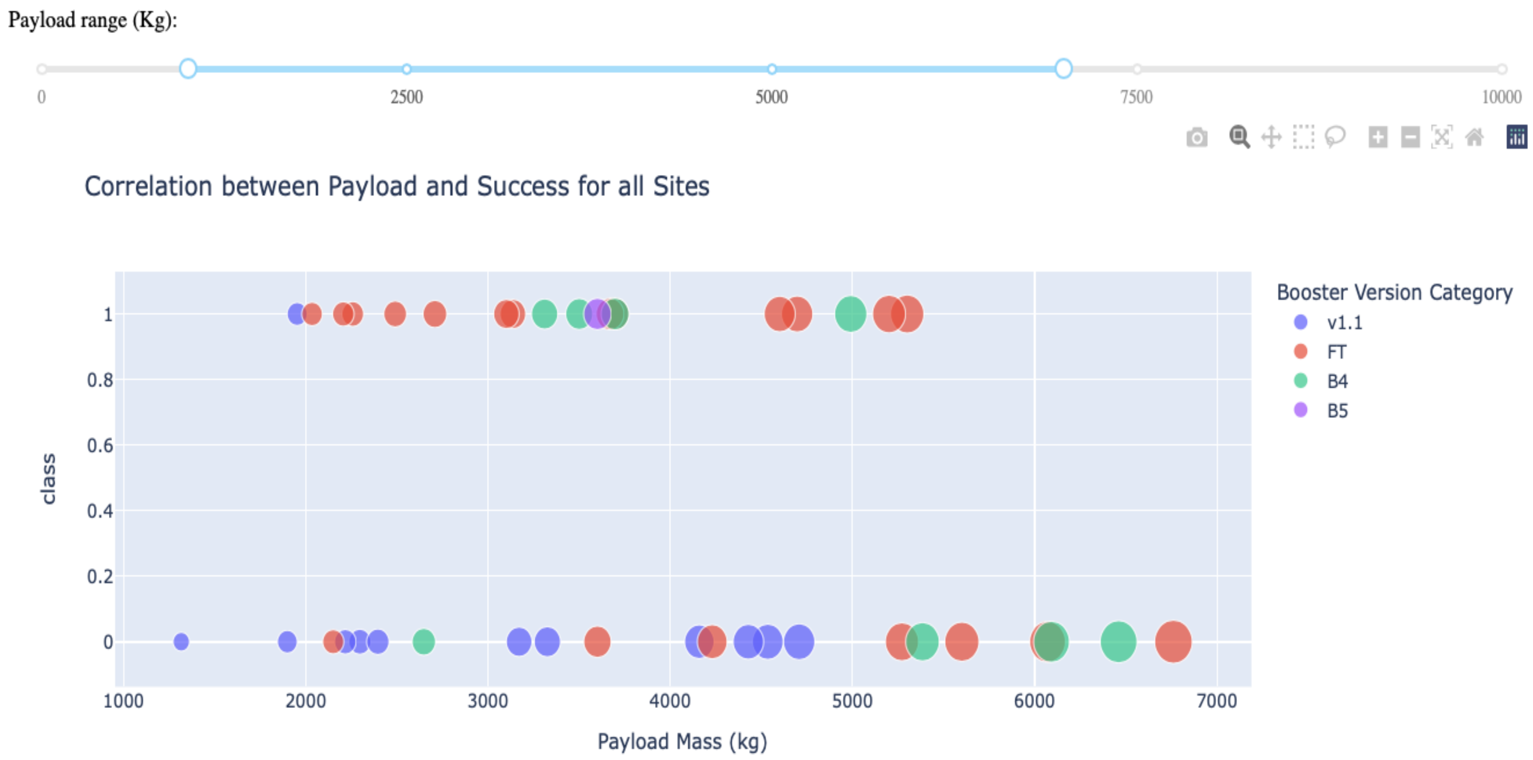
Successful site



KSC LC-39A

KSC LC-39A, the site with the highest number of successful launches, it can be seen that most of the launches were successful with more than 76% success and the rest of failures.

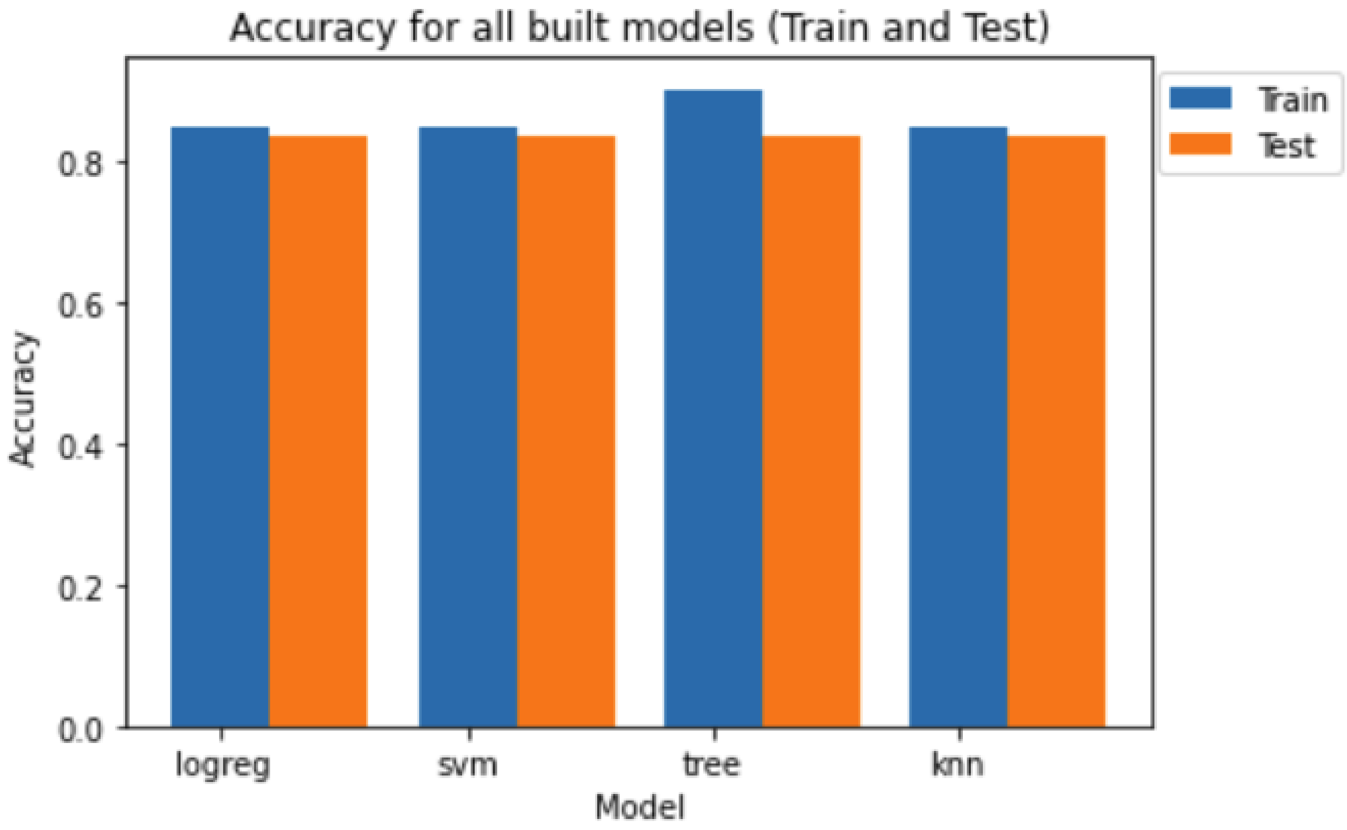
Payload Mass vs Success



In the range of 1000 to 7000 kg of payload mass it can be seen only 4 types of booster and most of the launches have not been successful and the range of payload mass is more reduced near 2000 and 5500.

Predictive analysis (Classification)

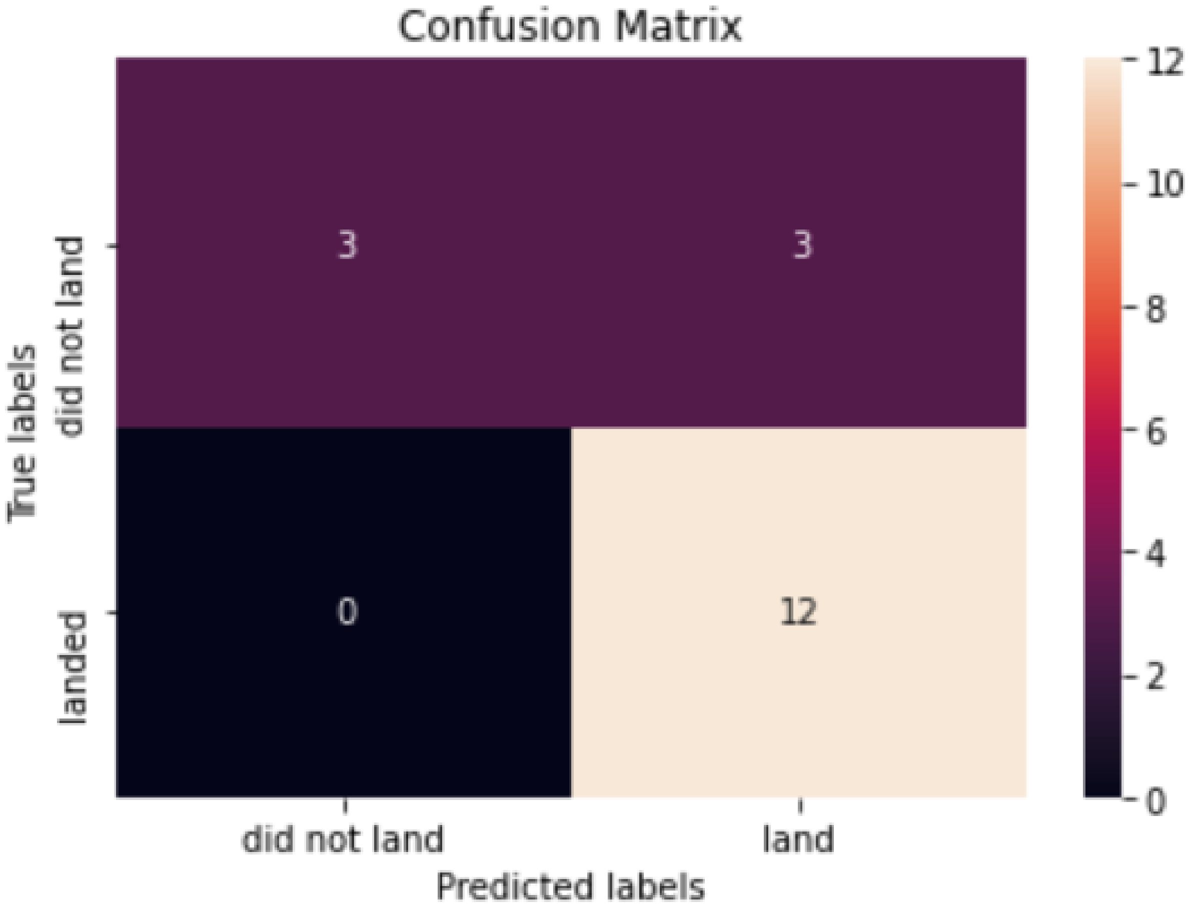
Classification Accuracy



The Tree classification model has the highest classification accuracy.

Confusion Matrix

The best option is tree classification model because it has the best accuracy and the false negatives ar zero and false positive are minor than true positives.

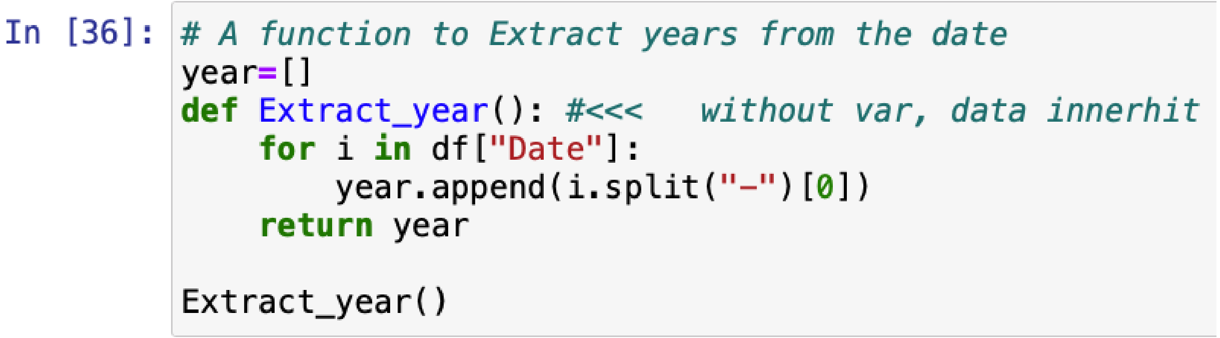


CONCLUSION

- **Tree classification model** is (apparently) the best option.
- It can be seen in the barchart in all cases the accurancy is similar, and in testing the levels of false negative and positive are the same, it must be considered the case of tree classification model has a random parameter (splitter) which makes the results vary in each iteration so it can generate a result to be the best option but at the next time could be not.
- It´s too important the variable **random_state** in general way and **splitting** in particualry case of **tree classification model** because depends on them this model to have the best accuracy of all.
- Another point important is the level of false positive because it could be a number significant and the **probability** of correct prediction of successful launch is **80%**.

APPENDIX

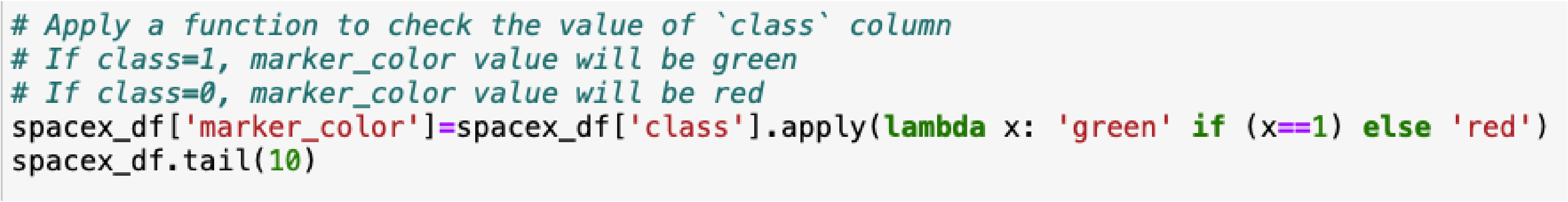
A detail found, a declared function with an unused argument can cause errors (jupyter-labs-eda-dataviz notebook)



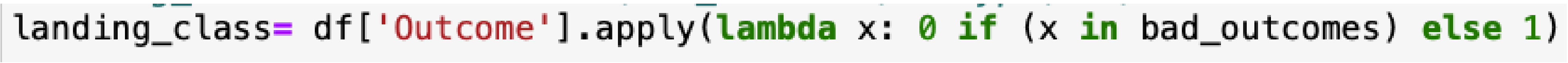
```
In [36]: # A function to Extract years from the date
         year=[]
         def Extract_year(): #<<<   without var, data innerhit
             for i in df["Date"]:
                 year.append(i.split("-")[0])
             return year

         Extract_year()
```

Use of lambda, two examples



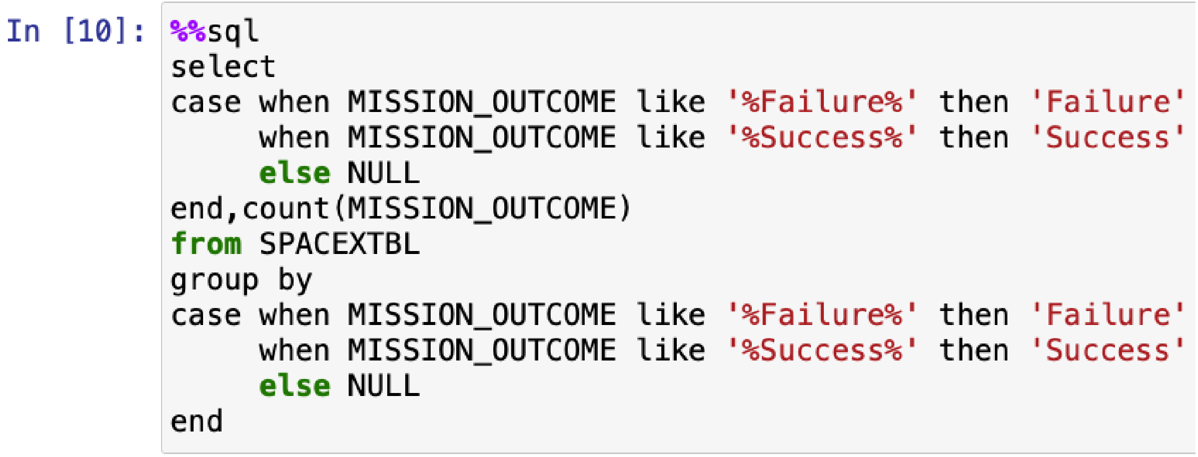
```
# Apply a function to check the value of `class` column
# If class=1, marker_color value will be green
# If class=0, marker_color value will be red
spacex_df['marker_color']=spacex_df['class'].apply(lambda x: 'green' if (x==1) else 'red')
spacex_df.tail(10)
```



```
landing_class= df['Outcome'].apply(lambda x: 0 if (x in bad_outcomes) else 1)
```

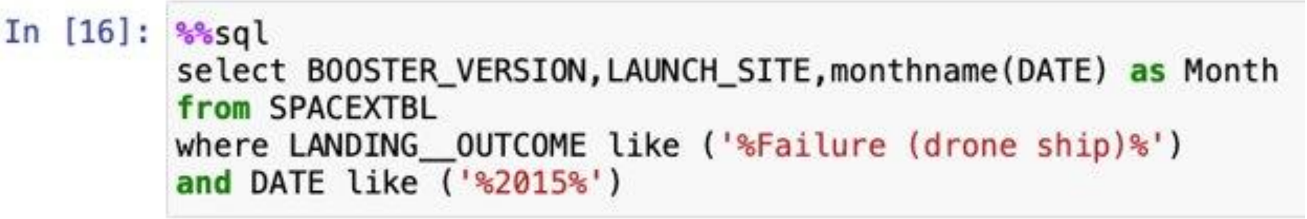
APPENDIX (cont.)

Use of CASE within SQL Query



```
In [10]: %%sql
         select
         case when MISSION_OUTCOME like '%Failure%' then 'Failure'
              when MISSION_OUTCOME like '%Success%' then 'Success'
              else NULL
         end,count(MISSION_OUTCOME)
         from SPACEXTBL
         group by
         case when MISSION_OUTCOME like '%Failure%' then 'Failure'
              when MISSION_OUTCOME like '%Success%' then 'Success'
              else NULL
         end
```

Use of monthname() within SQL Query



```
In [16]: %%sql
         select BOOSTER_VERSION,LAUNCH_SITE,monthname(DATE) as Month
         from SPACEXTBL
         where LANDING__OUTCOME like ('%Failure (drone ship)%')
         and DATE like ('%2015%')
```