Applied Data Science Capstone
SpaceX case study

Luciano Wilhelmsen M Moreno

september 13, 2021

Outline

- Methodologies:
  - Data Collection and wrangling
  - EDA with visualization and with SQL
  - Building map with Folium and Dashboard with Plotly Dash
  - Predictive analysis and Classification
- Results:
  - Exploratory data analysis results
  - Interactive analytics
  - Predictive analysis

Introduction

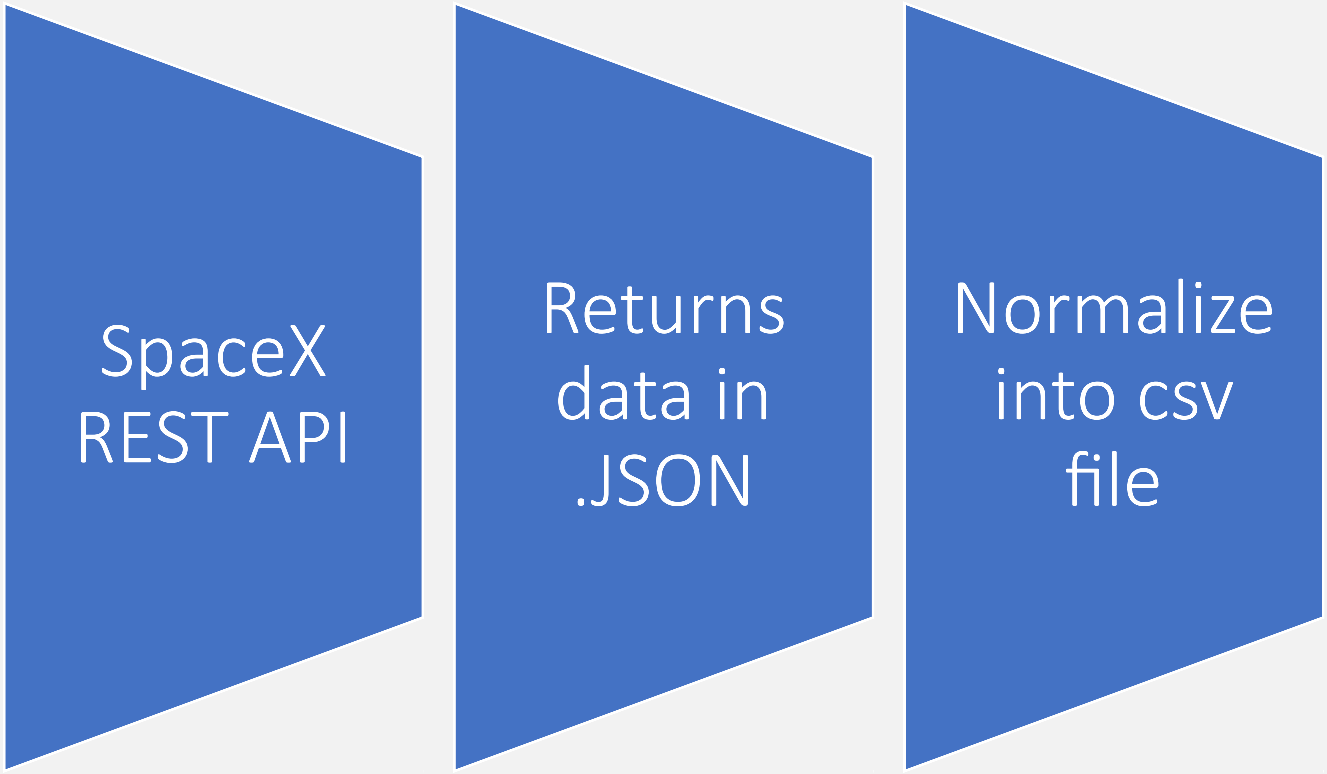
- Our task is predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each.
- Much of the savings is because SpaceX can reuse the first stage. Therefore if we can accurately predict the likelyhood of the first stage rocket landing successfully, we could determine the cost of a launch.
- What variables impact the success rate of a successful landing?

Methodology

- Data Collection with SpaceX Rest API and Web Scrapping from Wikipedia
- Data Wrangling via One Hot Encoding for Machine Learning, dropping irrelevant columns
- Exploratory Data Analysis using visualization and SQL:
  - Plotting Scatter and Bar Graphs shows relations between variables and patterns
- Interactive visual analysis using Folium and Plotly Dash
- Predictive analysis using classification models:
  - Built, tune and evaluate classification models

SpaceX API

- SpaceX REST API includes data for launches, rocket used, payload, launch and landing specs. And landing outcome.



SpaceX REST API

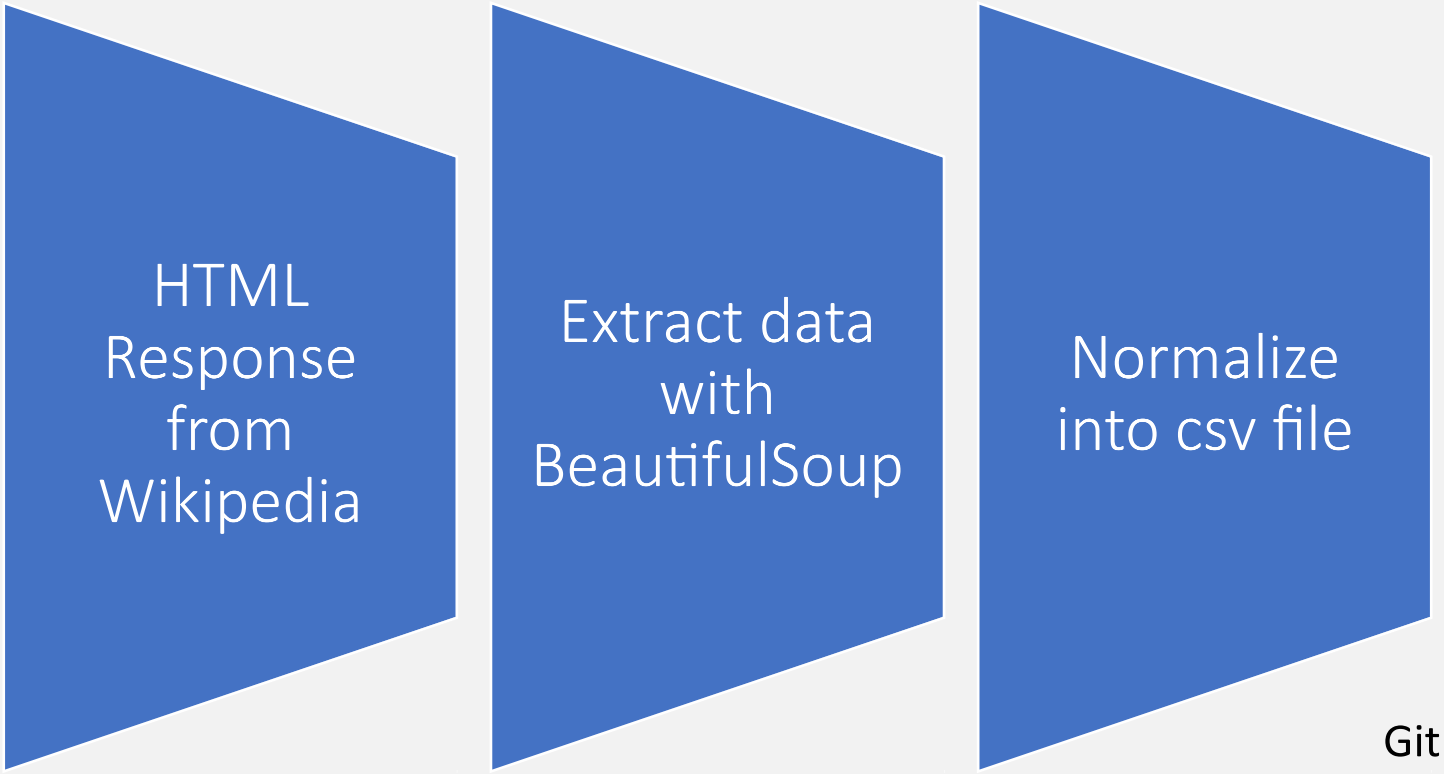
Returns data in .JSON

Normalize into csv file

GitHub

SpaceX Web Scrapping

- Web scrapping Wikipedia with BeautifulSoup to obtain more Falcon 9 Launches data



HTML Response from Wikipedia

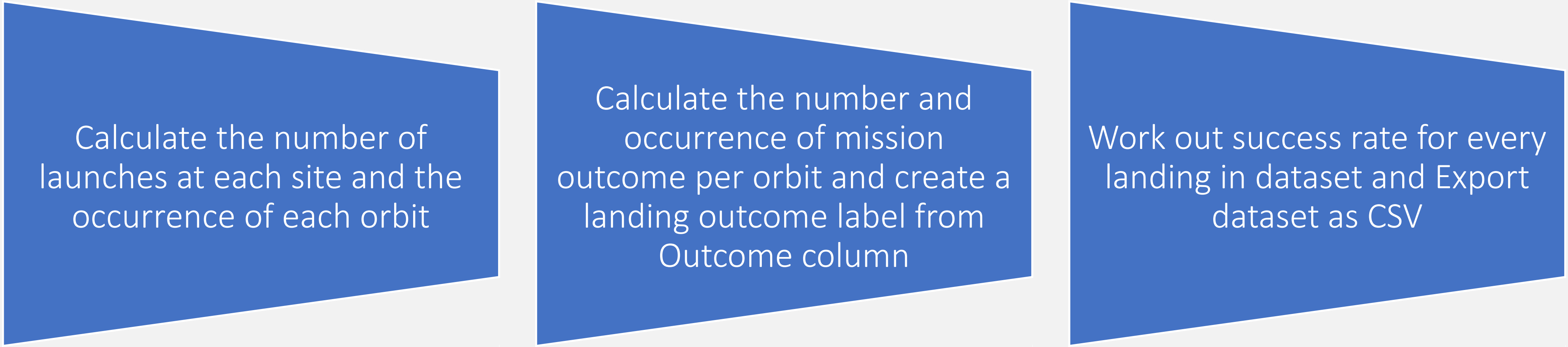
Extract data with BeautifulSoup

Normalize into csv file

GitHub

Data Wrangling

- There were several cases in the data set where the booster didn't land successfully.



Calculate the number of launches at each site and the occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit and create a landing outcome label from Outcome column

Work out success rate for every landing in dataset and Export dataset as CSV

- We converted the outcomes into Training Labels where 1 means the booster landed successfully and 0 means it did not.

GitHub

EDA with Data Visualization

- Scatter Graphs show how one variable is affected by another, with their relationship being called correlation:
  - Flight Number vs. Payload Mass
  - Flight nr. Vs. Launch site
  - Payload vs. Site
  - Orbit vs. Flight nr.
  - Payload vs. Orbit
  - Orbit vs. Payload Mass.
- Bar Graphs make easy to compare sets of data between different groups:
  - Orbit vs. Mean (success outcome)
- Line Graphs show data variables and trends very clearly:
  - Success rate vs. Year

GitHub

EDA with SQL

- Tasks performed in SQL query:
  - 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 successful landing outcome in ground pad was acheived.
  - 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 failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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

GitHub

Build of Interactive Map with Folium

- Added Circle Markers around each launch site with labels of the names of the sites, using the Latitude and Longitude coordinates;
- Added Green and Red markers on the map with MarkerCluster() to assign successful or not launches outcomes to the sites.
- Calculated distance from one Launch Site to landmarks around, using the Haversine's formula and coordinates.

GitHub

Build of Dashboard with Plotly Dash

- Added a Pie Chart showing the total launches by certain site or by all sites (select on a dropdown menu):
  - The graph shows proportion of Total Success Launches by all sites or for a certain site the proportion of success (class 1) or not (class 0);
- Added a Scatter Graph with the relation between Outcome (classes 0 or 1) and Payload Mass for different Booster Versions:
  - Possible to choose the range of payload mass displayed on the slider, and
  - the launch site on the dropdown menu.

GitHub

Predictive Analysis

- Load data into NumPy and Pandas, and transform the data
- Split data into Training and Test data sets, check number of test samples
- For each type of machine learning algorithm, set parameters and algorithms to GridSearchCV;
- Fit datasets into the GrisSearchCV objects and train;
- Evaluate each model as follows:

Check for accuracy

Tune hyperparamethers for each algorithm

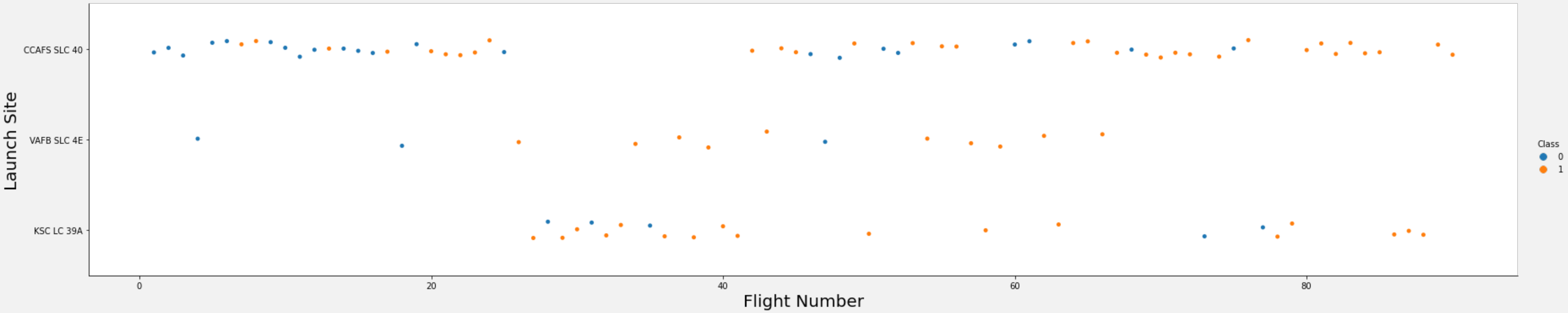
Plot the confusion Matrix

GitHub

Results

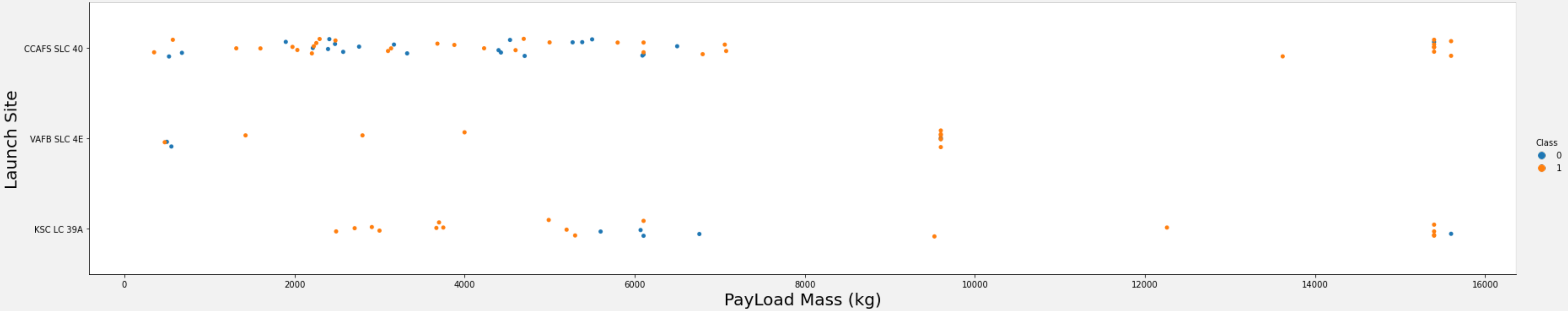
- Exploratory data analysis results

Flight Number vs. Launch Site



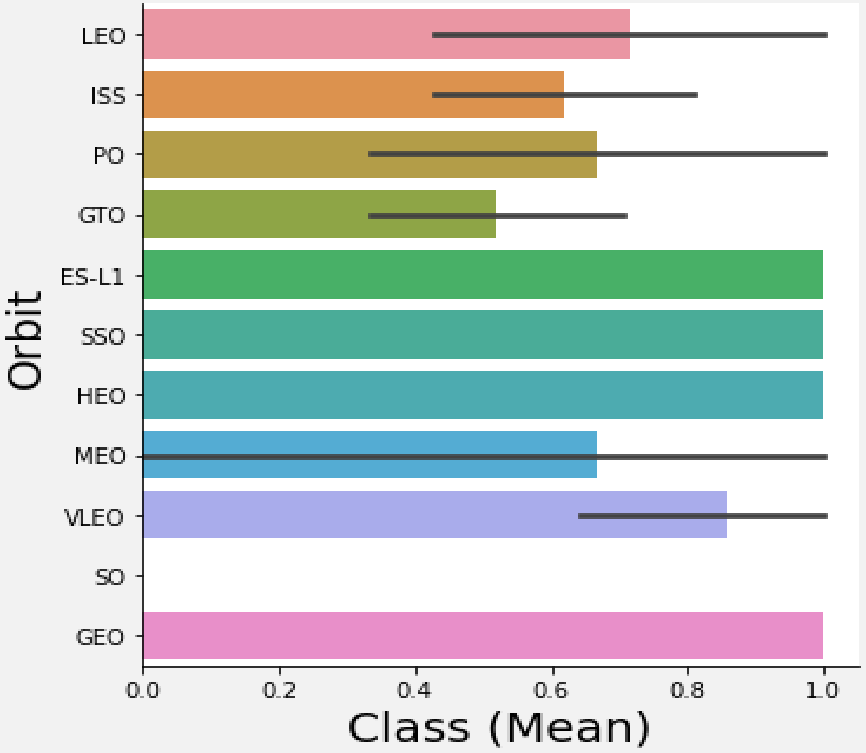
The success rate (class 1) seems to increase with the number of the flight at launch site

Payload Mass vs. Launch Site



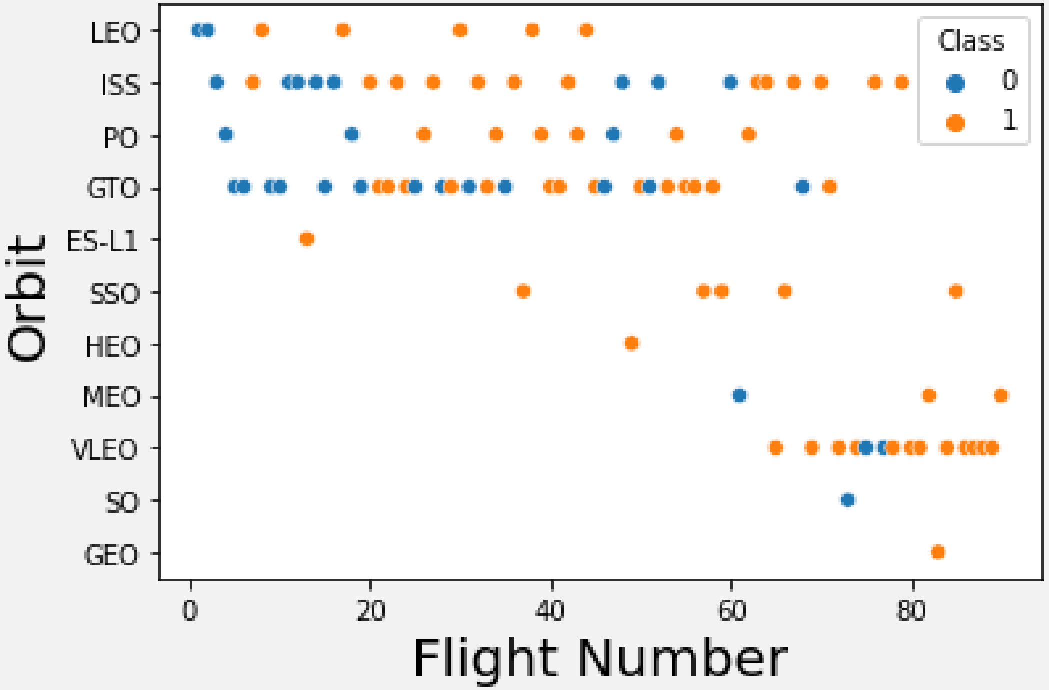
In general, the greater the payload mass, the higher the success rate (class 1). But there were successful launches with low Payload in all the sites.

Success rate vs. Orbit type



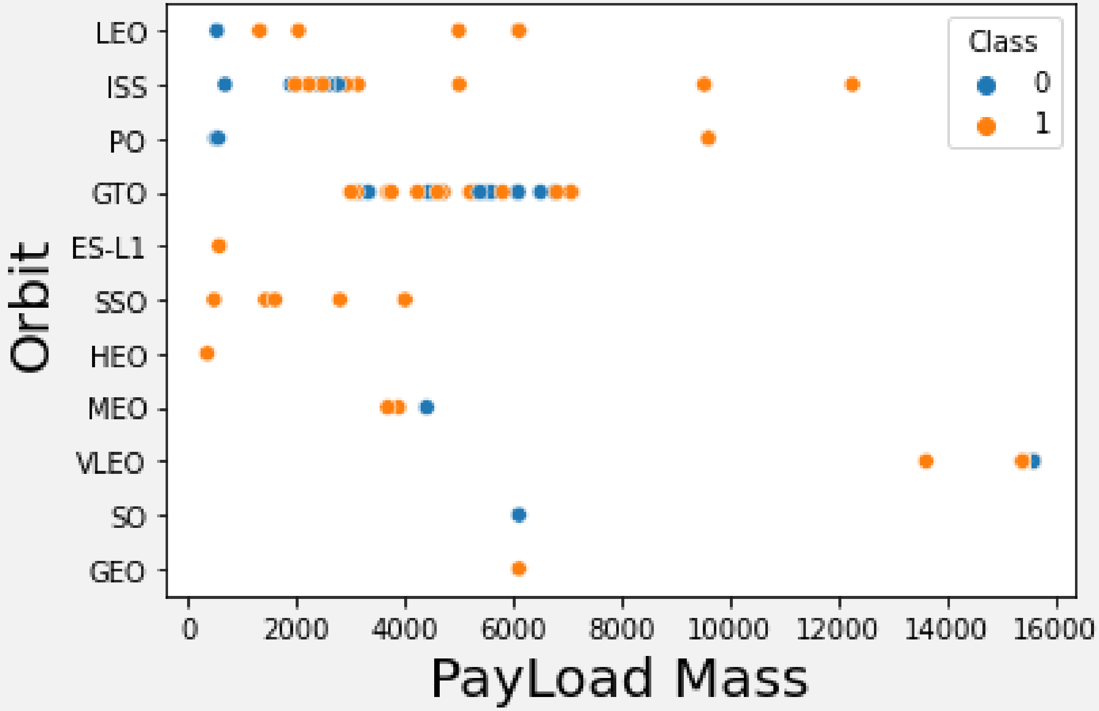
Orbits GEO, HEO, SSO and ES-L1 had the highest success rate (class 1).

Flight Number vs. Orbit type



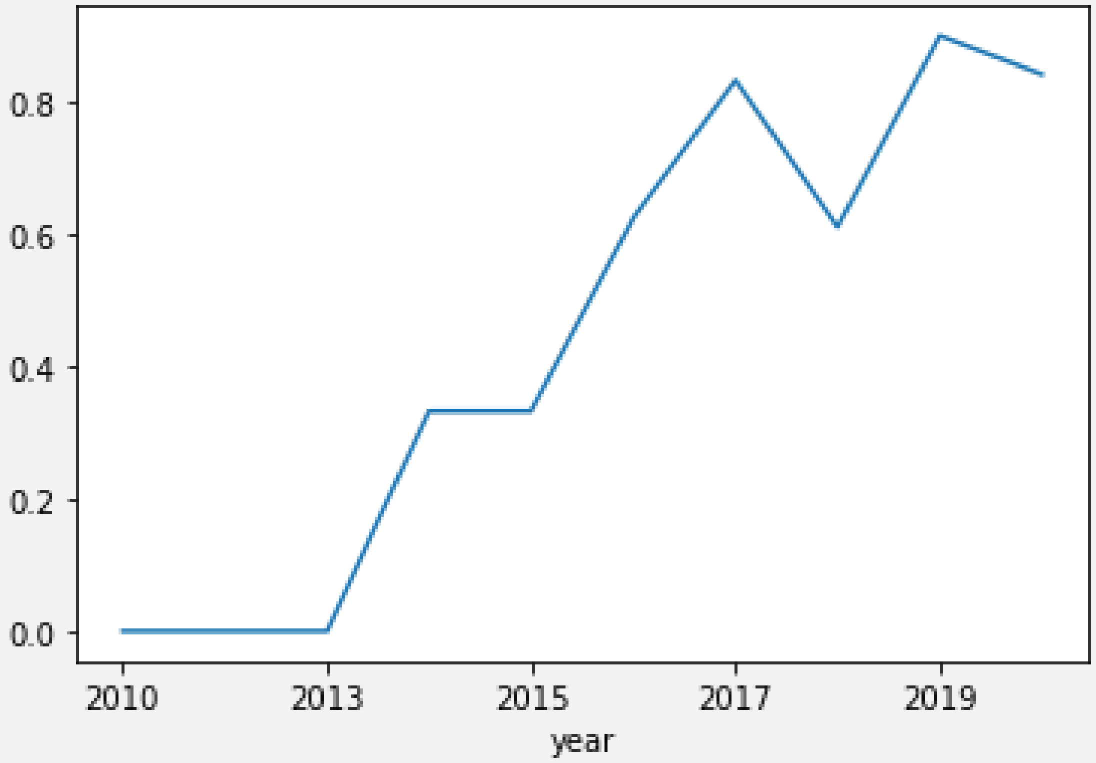
Success (class 1) seems to be related with the increase in Flight Number in diferent Orbits. That's true for Orbit LEO, but not as clear in ISS Orbit, for example.

Payload vs. Orbit type



Payload increase seems to be related with the success in Orbit LEO, for example, but that is not true for GTO.

Launch Success Yearly trend

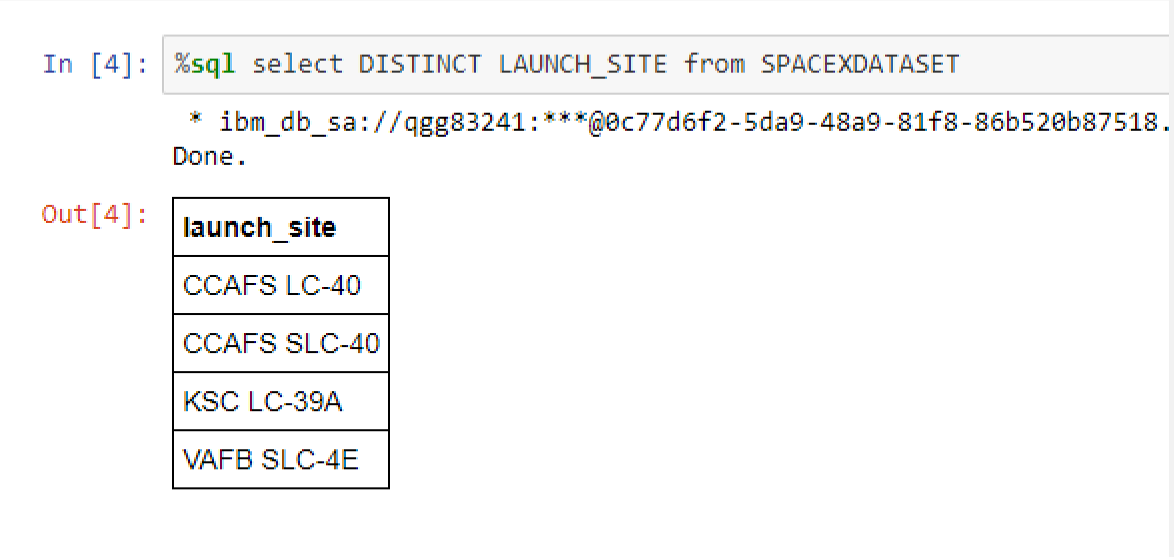


We can observe that success rate kept increasing from 2013 untill 2020.

Results

- Exploratory data analysis with SQL results

All Launch Site names



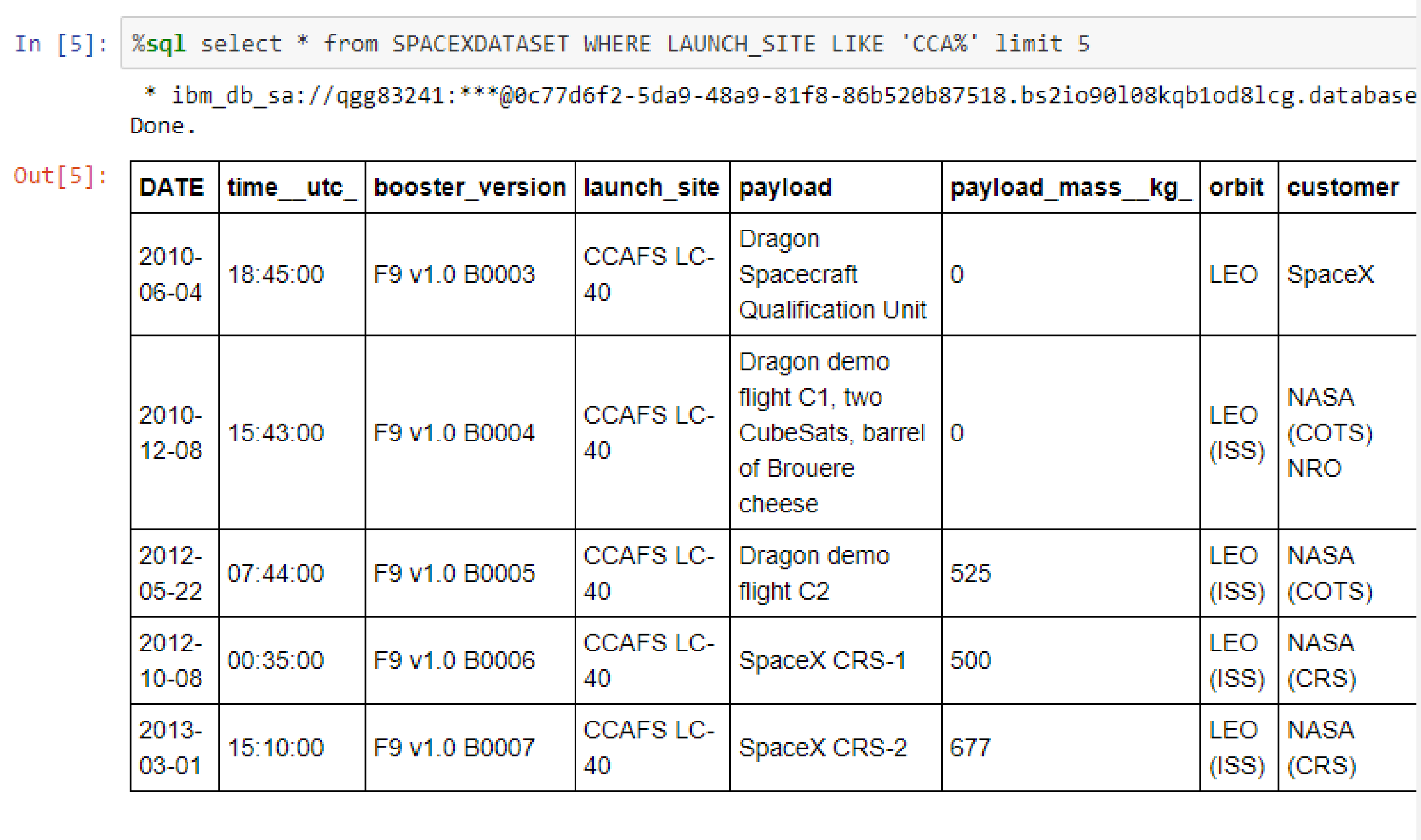
```
In [4]: %sql select DISTINCT LAUNCH_SITE from SPACEXDATASET

 * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.
Done.
```

Out[4]:

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

Use DISTINCT to show unique values in the column.

Launch Site names that Begin with ‘CCA’



```
In [5]: %sql select * from SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' limit 5
```

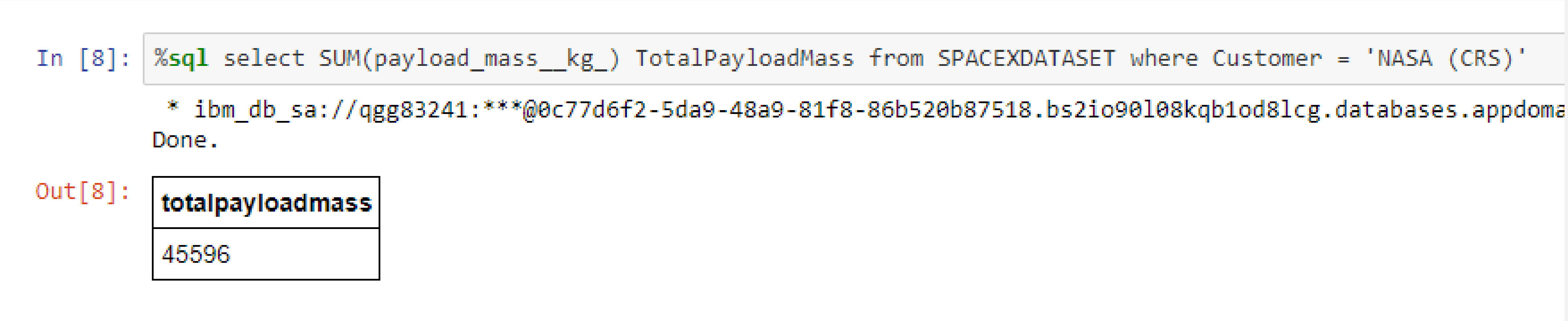
```
 * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
Done.
```

Out[5]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Use LIKE with 'CCA%' will show only results that begin with the letters CCA (% symbol at the end). Limit 5 at the end of the query to show only five results.

Total Payload Mass by Customer NASA (CRS)



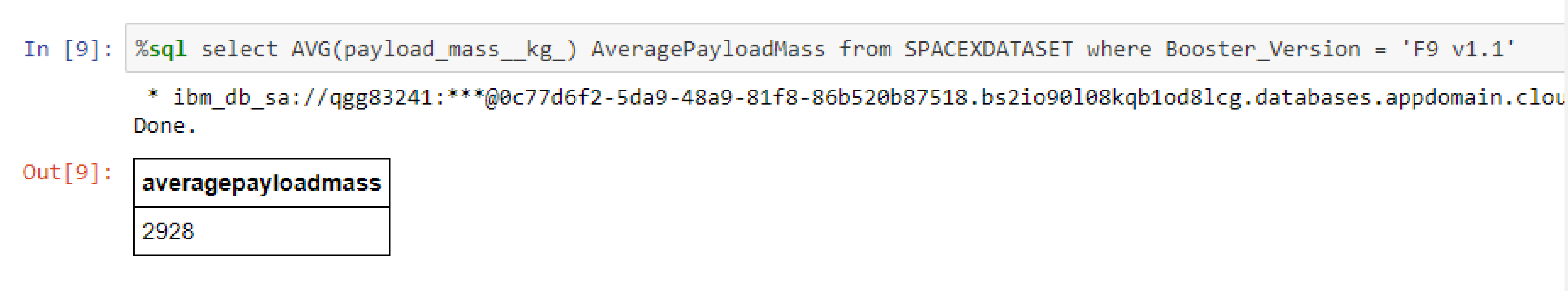
```
In [8]: %sql select SUM(payload_mass__kg_) TotalPayloadMass from SPACEXDATASET where Customer = 'NASA (CRS)'

         * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdoma
        Done.
```

Out[8]:

totalpayloadmass
45596

Use of SUM to get the total of values in the column Payload Mass
and WHERE to filters the dataset to only get NASA (CRS) on Customer column.

Average Payload Mass carried by F9 v1.1 booster



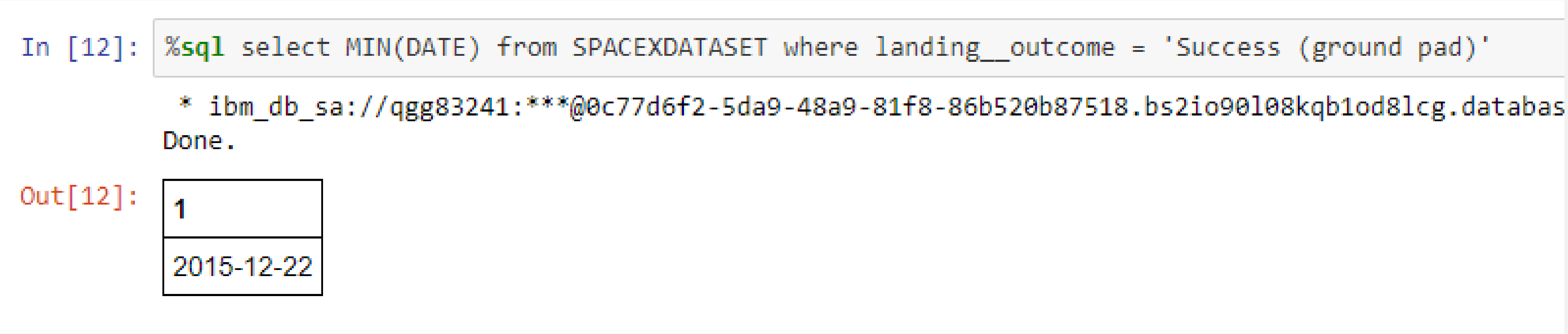
```
In [9]: %sql select AVG(payload_mass__kg_) AveragePayloadMass from SPACEXDATASET where Booster_Version = 'F9 v1.1'

 * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.clou
Done.
```

Out[9]:

averagepayloadmass
2928

Use of AVG to get the average of values in the column Payload Mass
and WHERE to filters the dataset to only get F9 v1.1 on Booster_Version column.

First Successful Ground Landing Date



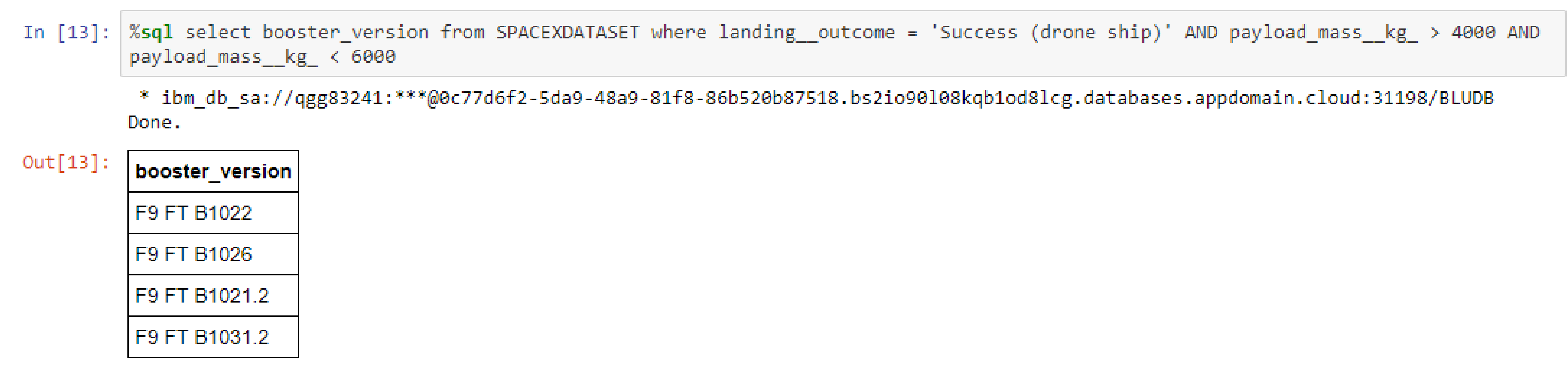
```
In [12]: %sql select MIN(DATE) from SPACEXDATASET where landing__outcome = 'Success (ground pad)'

          * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databas
         Done.
```

Out[12]:

1
2015-12-22

Use of MIN to get the minimum value in the column date and
WHERE to filters the dataset to only get Success (drone ship) on Landing_Outcome
column.

Successful Drone Ship Landing with Payload between 4000 and 6000

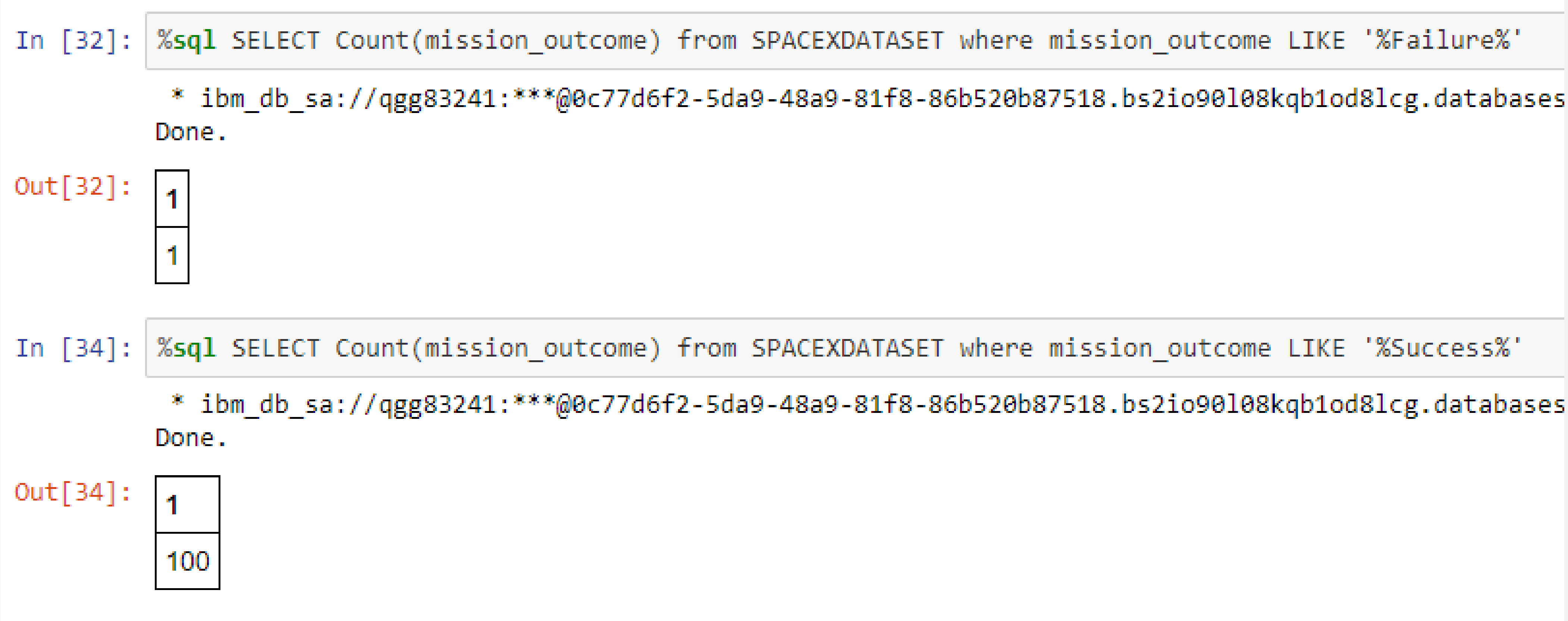


```
In [13]: %sql select booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND
         payload_mass__kg_ < 6000

          * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
         Done.

Out[13]:
```

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

Select booster_version, using WHERE to filters the dataset to only get Success (drone ship) on Landing_Outcome column, plus AND clause to filter Payload Mass

Total number of Successful and Failure Mission Outcomes



```
In [32]: %sql SELECT Count(mission_outcome) from SPACEXDATASET where mission_outcome LIKE '%Failure%'

          * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases
         Done.
```

Out[32]:

1
1

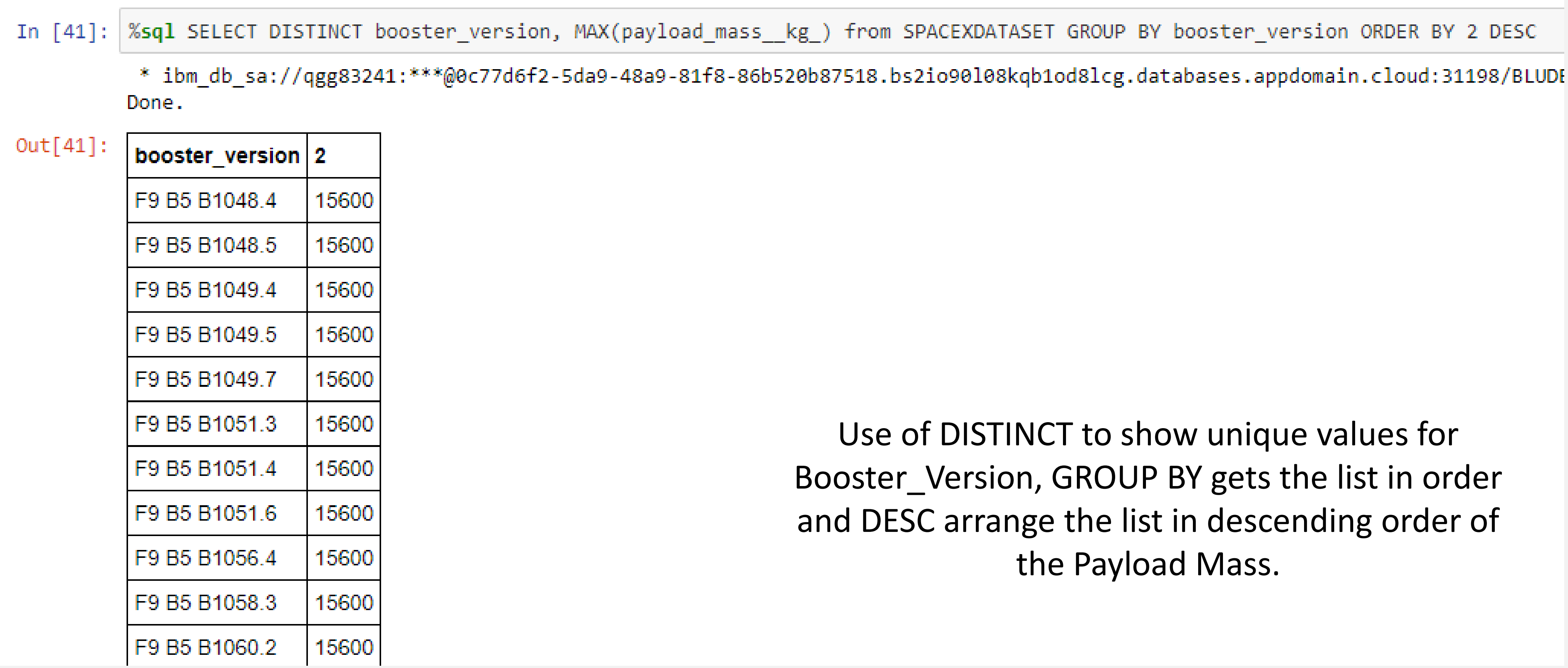
```
In [34]: %sql SELECT Count(mission_outcome) from SPACEXDATASET where mission_outcome LIKE '%Success%'

          * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases
         Done.
```

Out[34]:

1
100

Use of SELECT and then Count the number of outcomes with WHERE and LIKE to filter the results.

Boosters Carried Maximum Payload



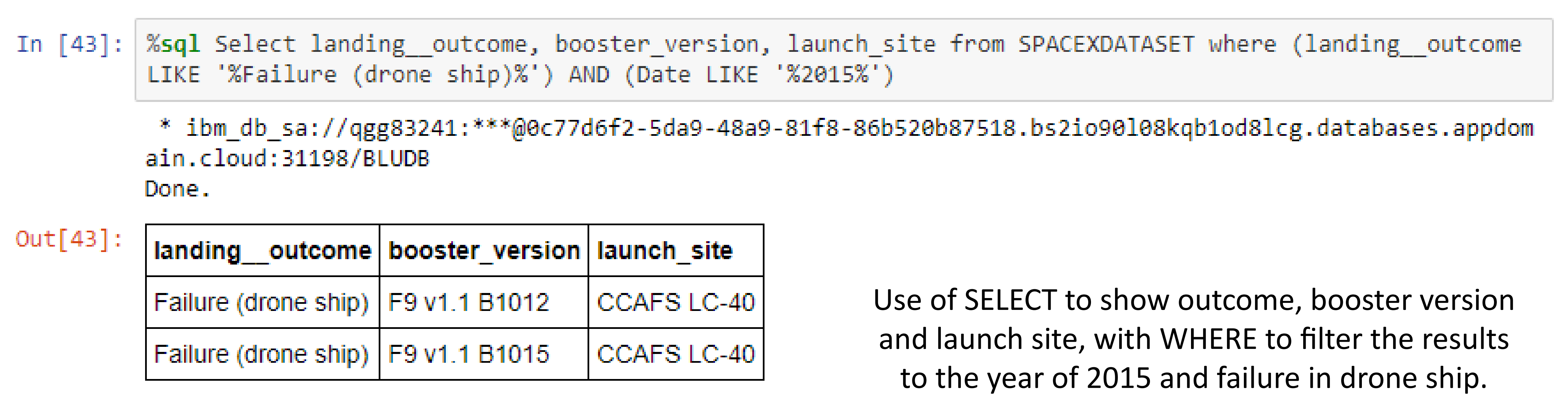
```
In [41]: %sql SELECT DISTINCT booster_version, MAX(payload_mass__kg_) from SPACEXDATASET GROUP BY booster_version ORDER BY 2 DESC

          * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDE
         Done.
```

Out[41]:

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

Use of DISTINCT to show unique values for Booster_Version, GROUP BY gets the list in order and DESC arrange the list in descending order of the Payload Mass.

2015 Launch Records



```
In [43]: %sql Select landing__outcome, booster_version, launch_site from SPACEXDATASET where (landing__outcome
         LIKE '%Failure (drone ship)%') AND (Date LIKE '%2015%')
```

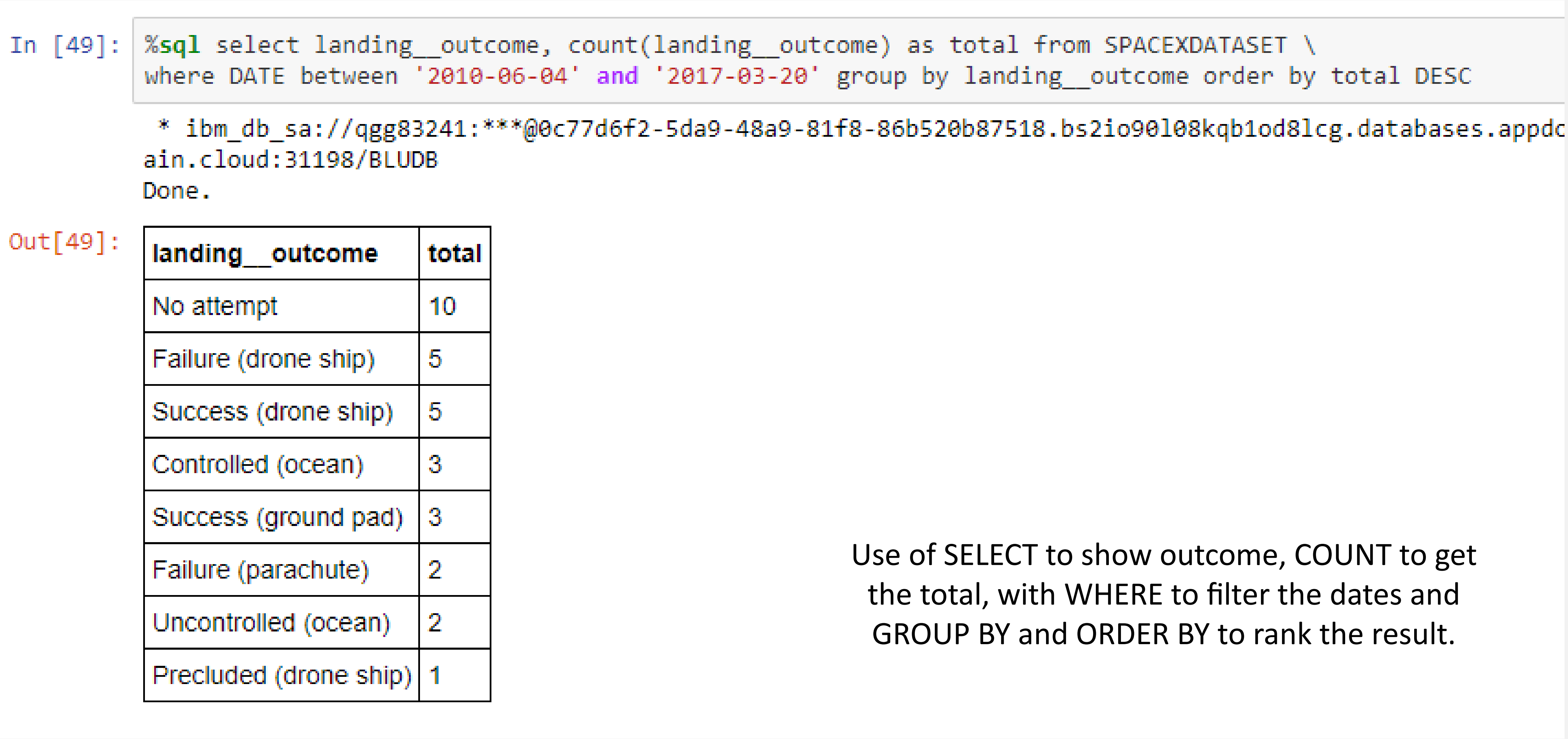
```
 * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdom
ain.cloud:31198/BLUDB
Done.
```

Out[43]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Use of SELECT to show outcome, booster version and launch site, with WHERE to filter the results to the year of 2015 and failure in drone ship.

Rank Landing Outcomes Between two dates



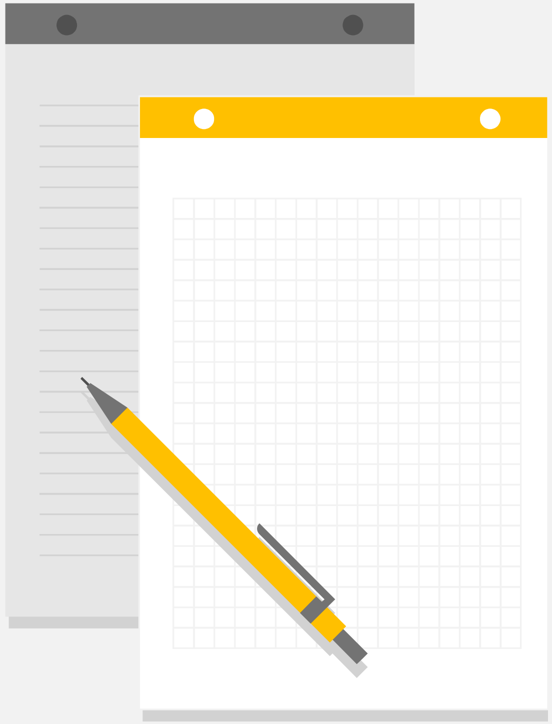
```
In [49]: %sql select landing__outcome, count(landing__outcome) as total from SPACEXDATASET \
         where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome order by total DESC

          * ibm_db_sa://qgg83241:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdo
         ain.cloud:31198/BLUDB
         Done.
```

Out[49]:

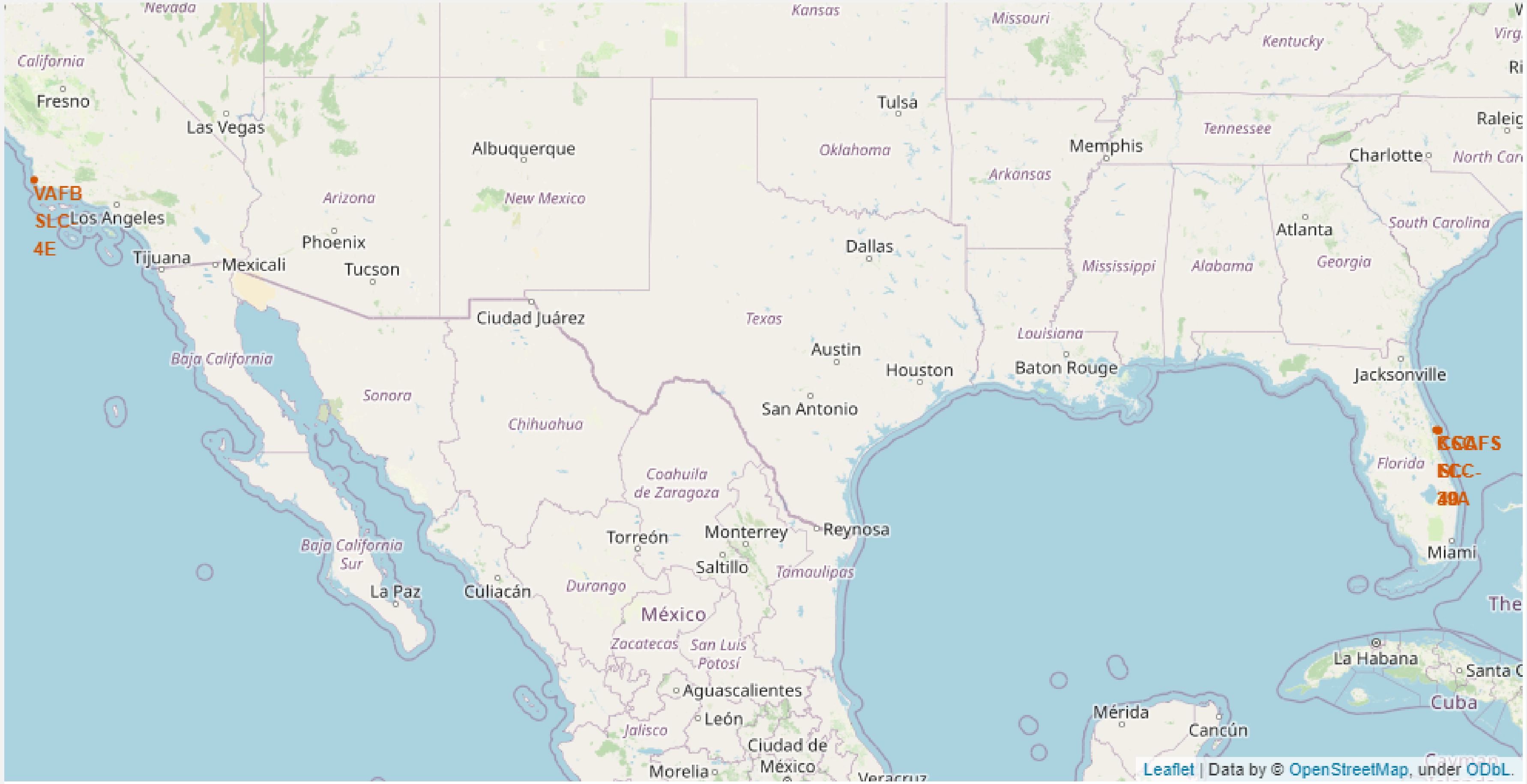
landing__outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Use of SELECT to show outcome, COUNT to get the total, with WHERE to filter the dates and GROUP BY and ORDER BY to rank the result.

Launch Sites Proximities Analysis

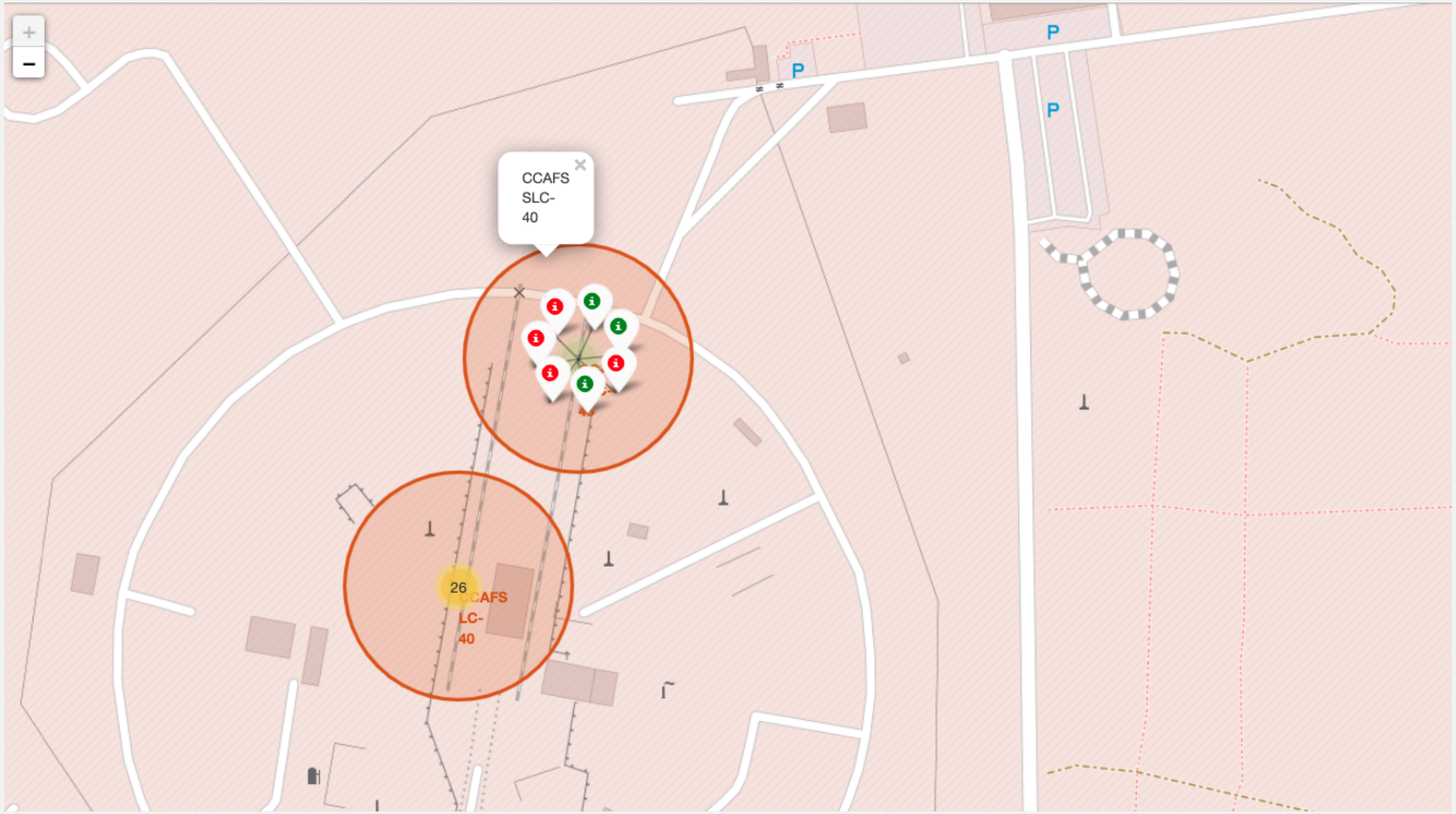


Circle and label launch sites



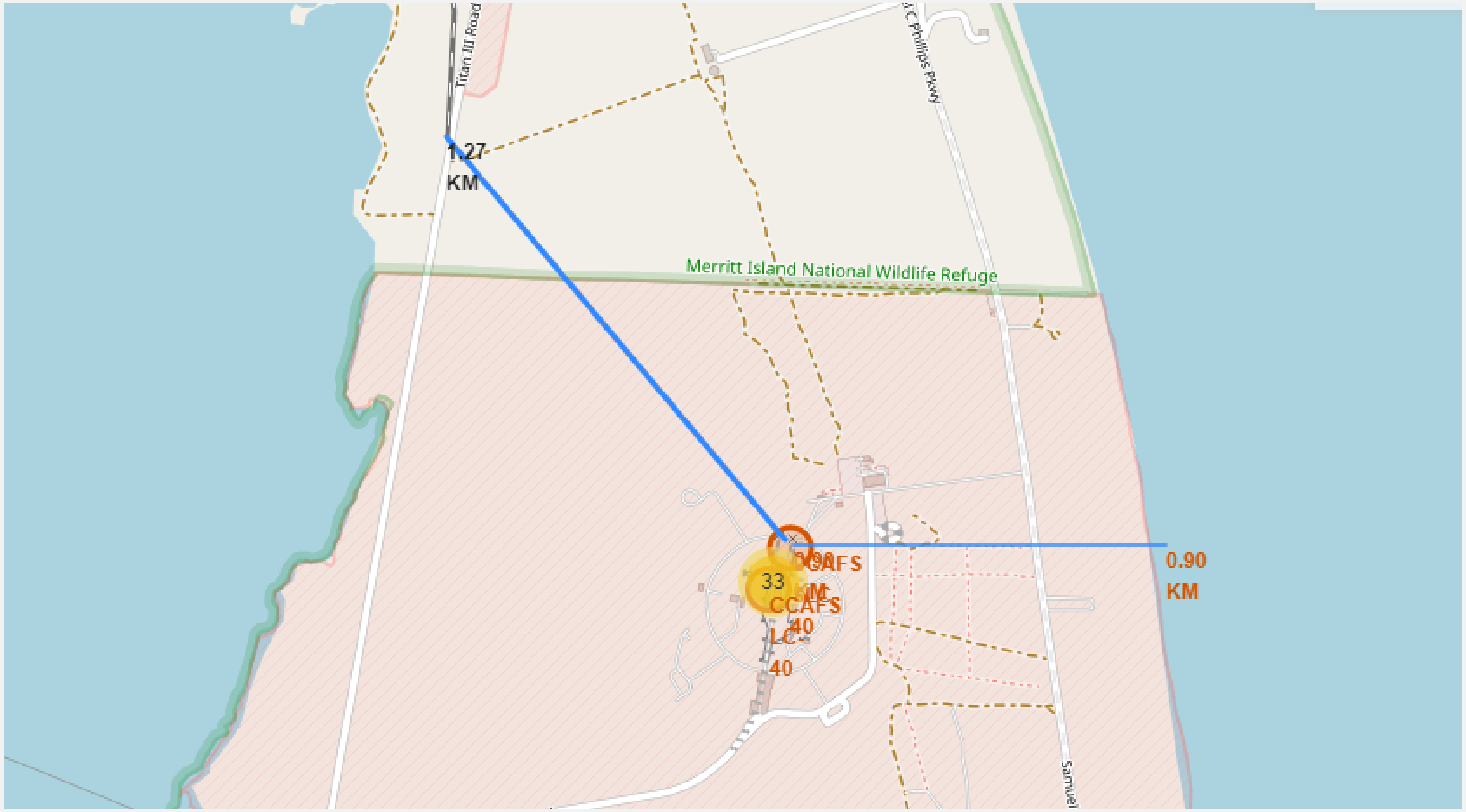
We can see that the launch sites are close to the coast and in the South of the country.

Success launches per site



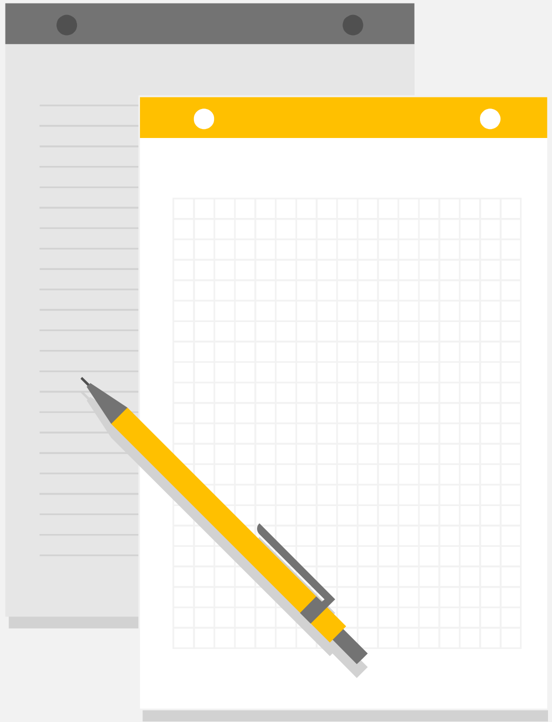
We can explore what launch sites had more success by clicking on the number of launches per site.

Proximities of the launch site

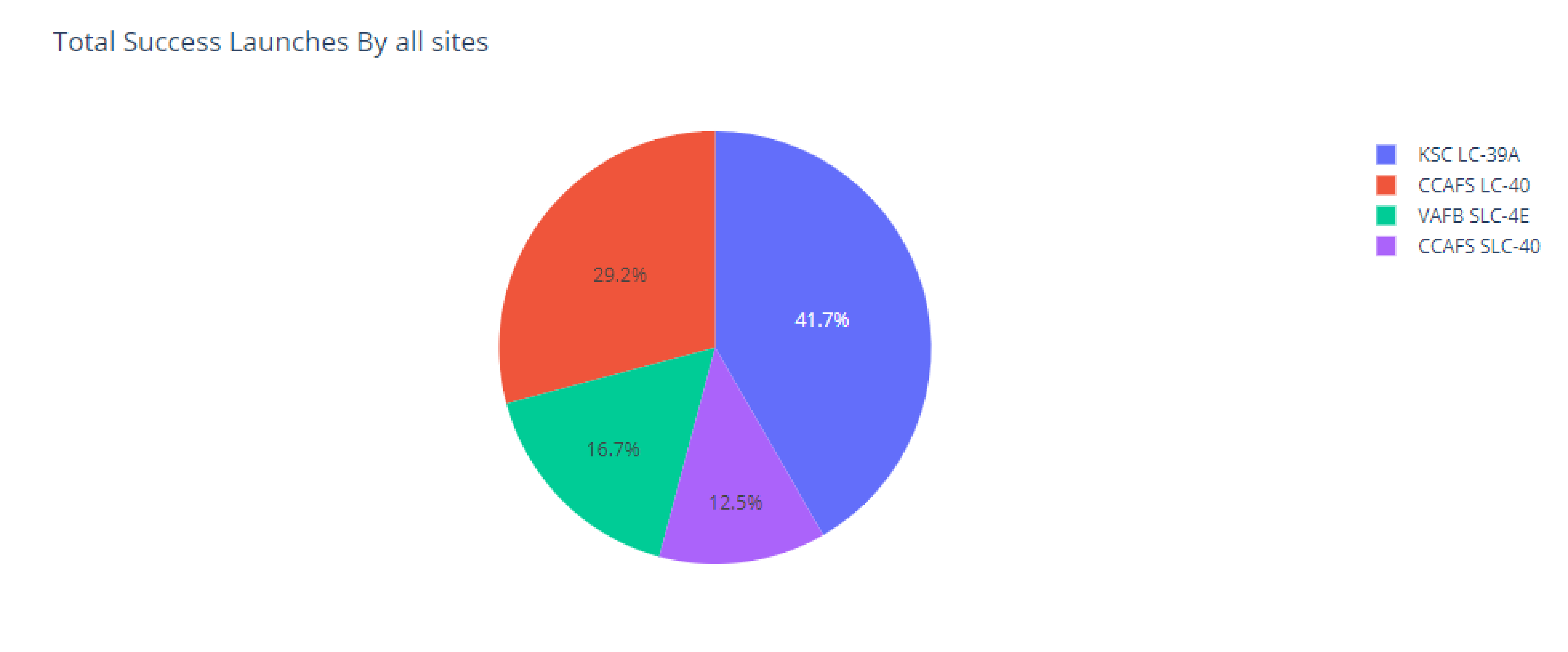


We can measure proximity of launch sites to the coast and to the nearest railway.

Build a Dashboard with Plotly Dash

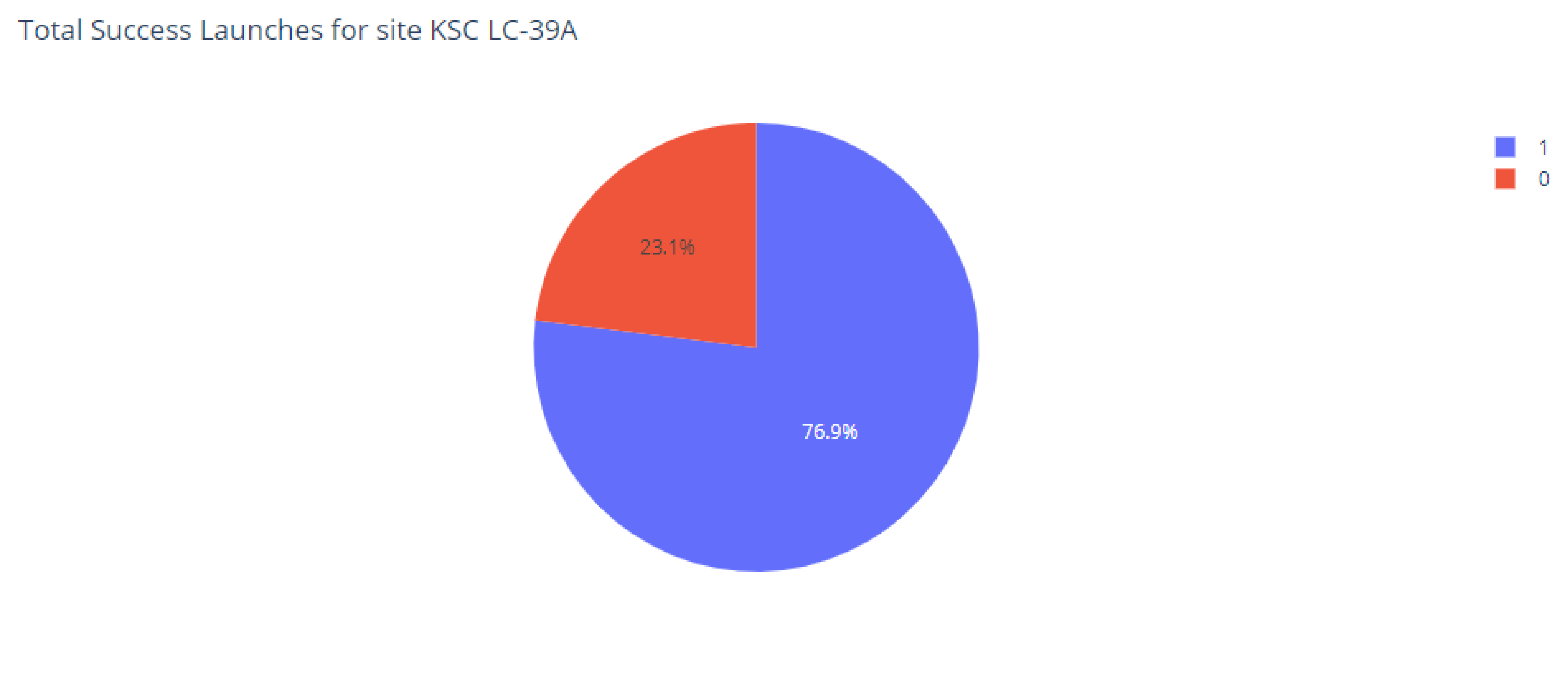


Pie chart of success percentage by launch site



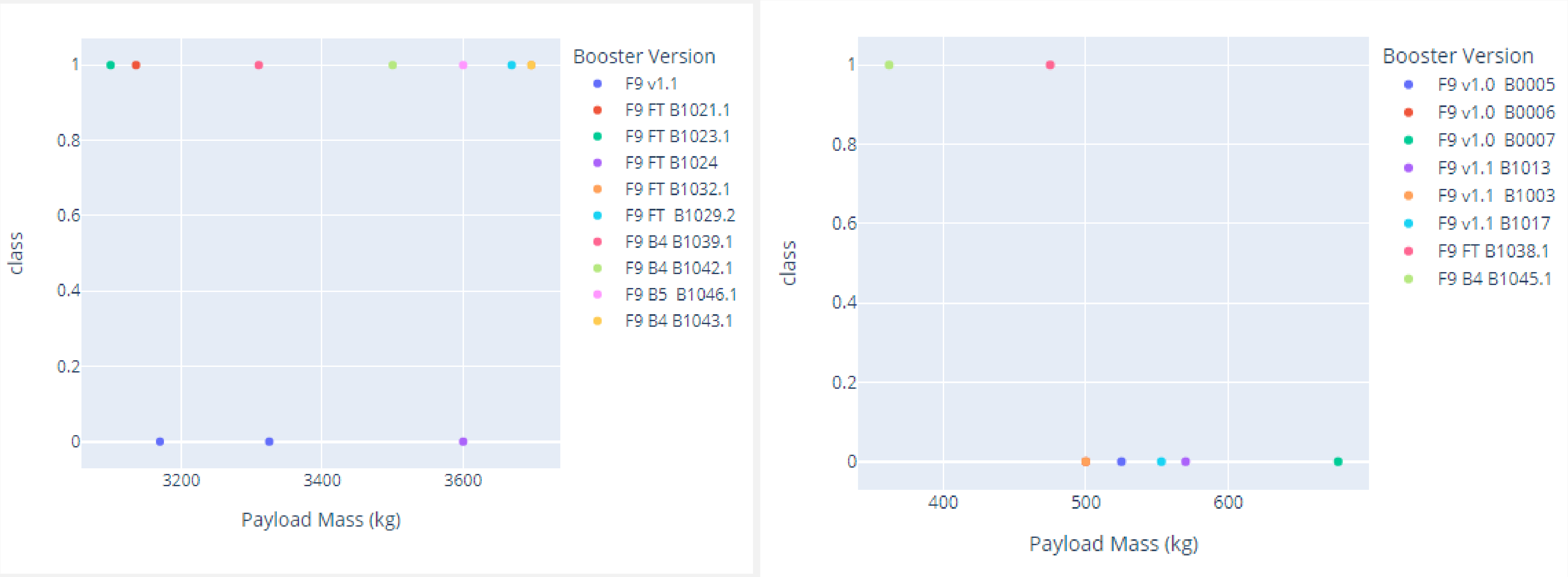
From all the sites, the one that had most successful launches was KSC LC-39A

Pie chart of success percentage by KSC LC-39A launch site



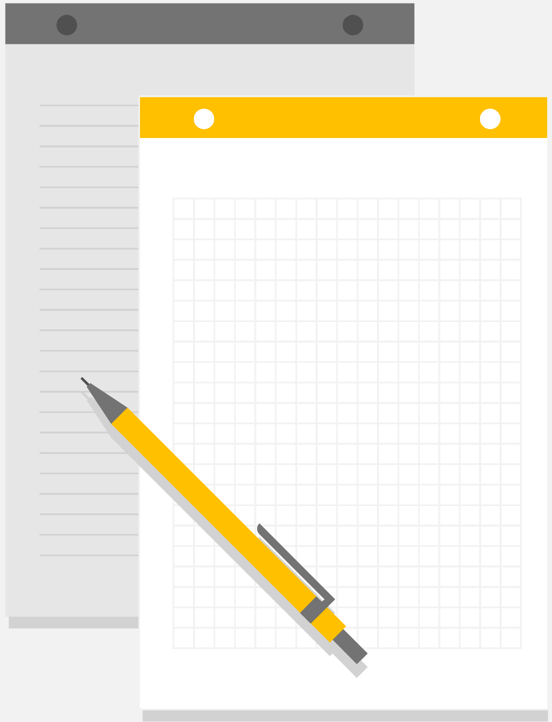
KSC LC-39A is a launch site with 76,9% of success launches rate

Payload vs. Launch Outcome scatter plots for all sites

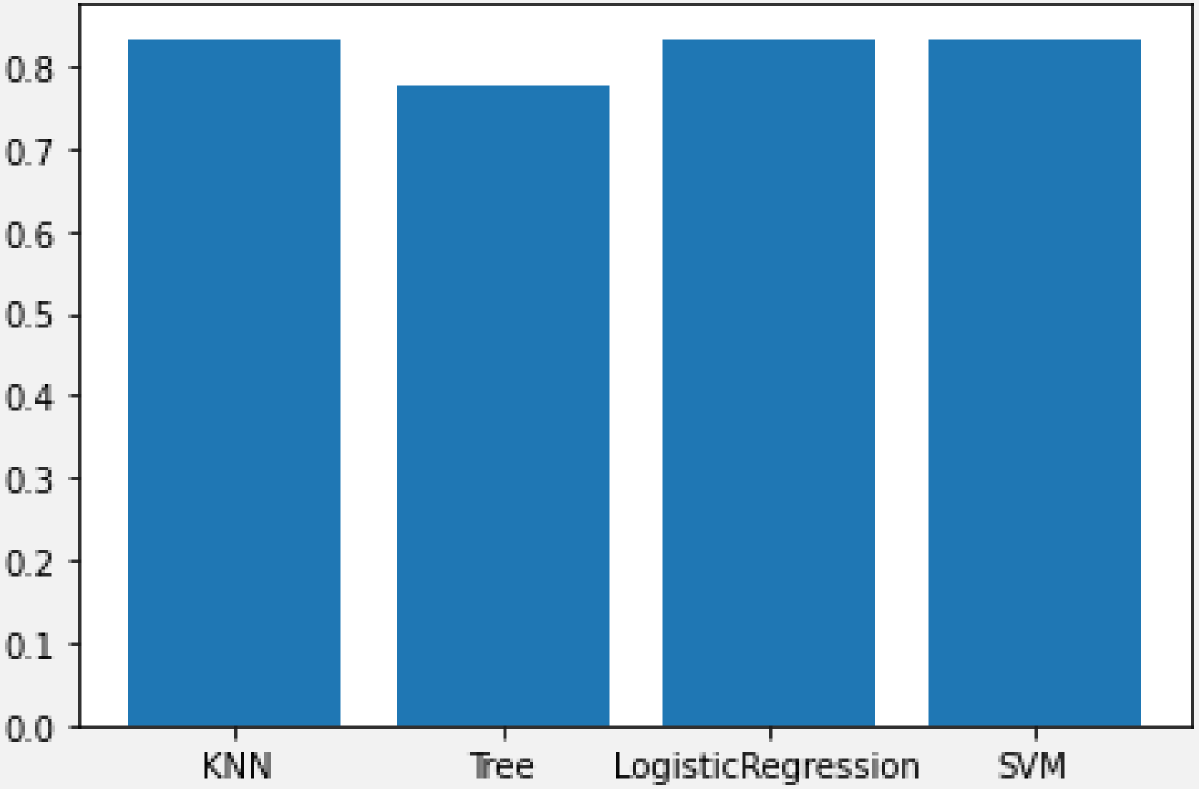


We can see that with payload between 3000kg and 4000kg the success rate were higher than with payload between 0 and 1000kg.

Predictive analysis (Classification)

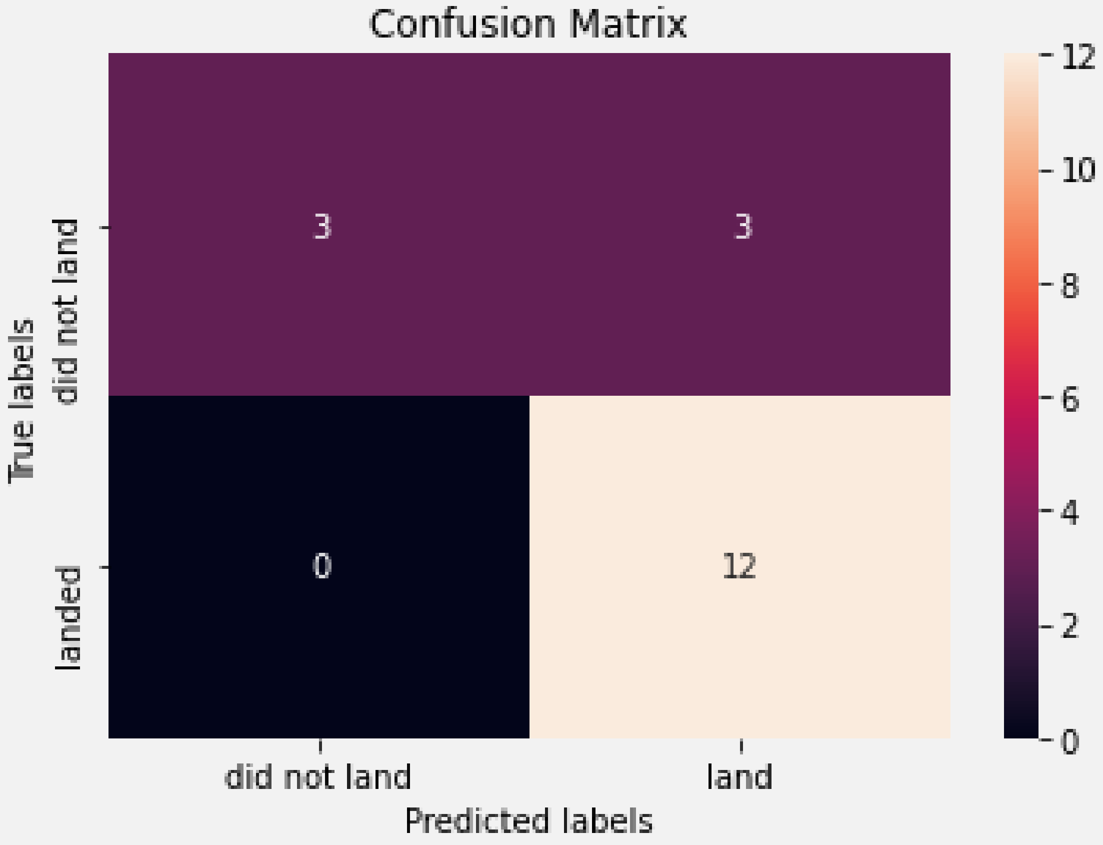


Classification Accuracy



The models are very close from each other in the accuracy found.

Confusion Matrix



All models have the same confusion matrices.
The problem that we see in the models are false positives.

Conclusion

- The machine learning models used have almost the same accuracy, returning some false positives, but no false negatives
- Some payload ranges perform better than others
- The success rates for SpaceX launches increases over time
- KSC LC-39A had the most successful launches from the all sites
- Some Orbits also have Success rate higher than other