IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamad Abdul Rahman
20-09-2021

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API which gives us data about launches including info about Rocket, Core, Capsule, Starlink, Launch Pad and Landing Pad data.
  - API url: “`api.spacexdata.com/v4/launches/past`”
- Perform data wrangling:
  - Dealing with missing values in column by replacing it with mean value of the data in this column

Methodology

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

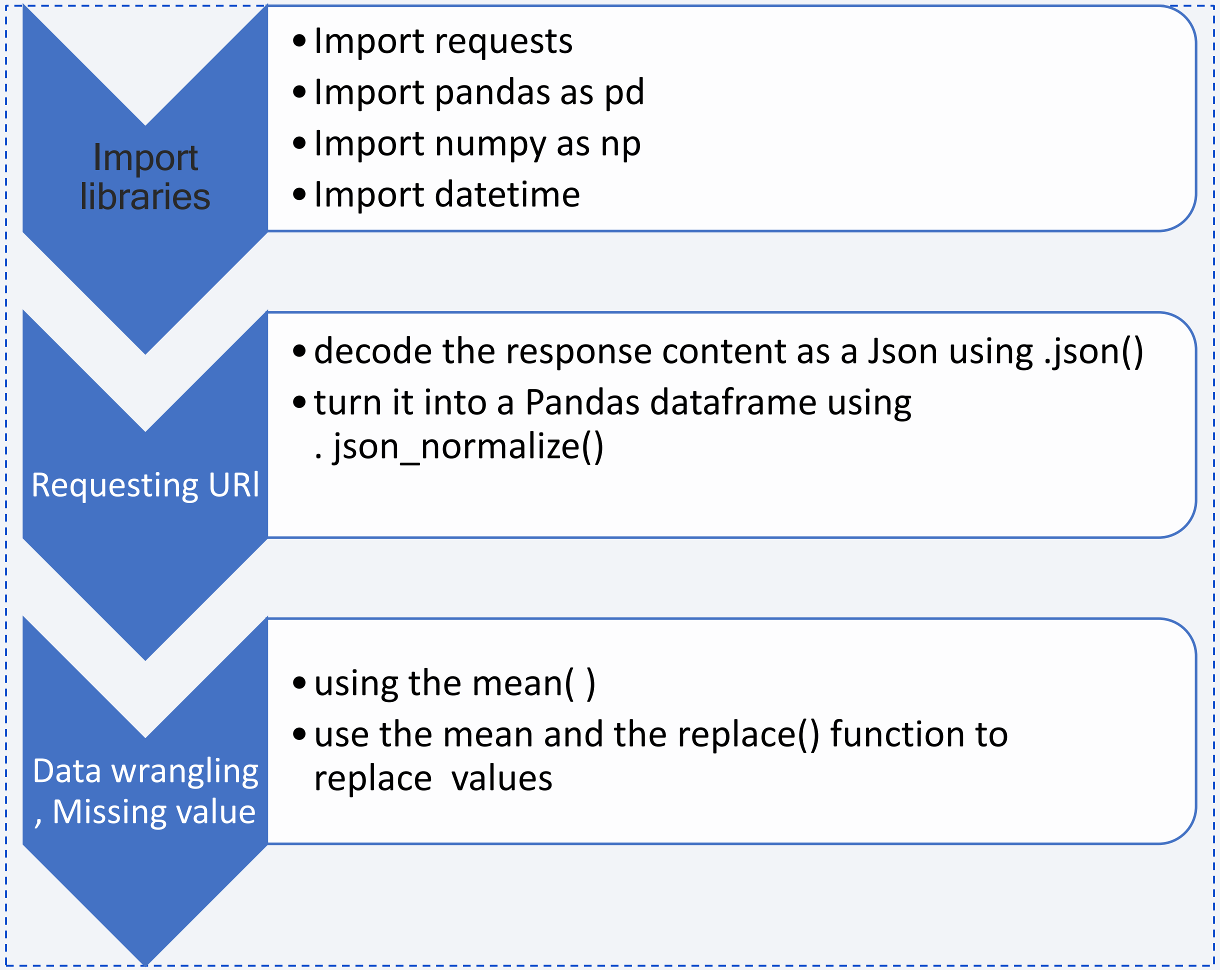
Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection - SpaceX API

1. Import libraries and define auxiliary Functions
2. Filter the dataframe to only include falcon 9 launches
3. Dealing with missing values
4. Request and parse the SpaceX launch data using the GET request

- https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%201/jupyter-labs-spacex-data-collection-api.ipynb



Import libraries

- Import requests
- Import pandas as pd
- Import numpy as np
- Import datetime

Requesting URl

- decode the response content as a Json using .json()
- turn it into a Pandas dataframe using . json_normalize()

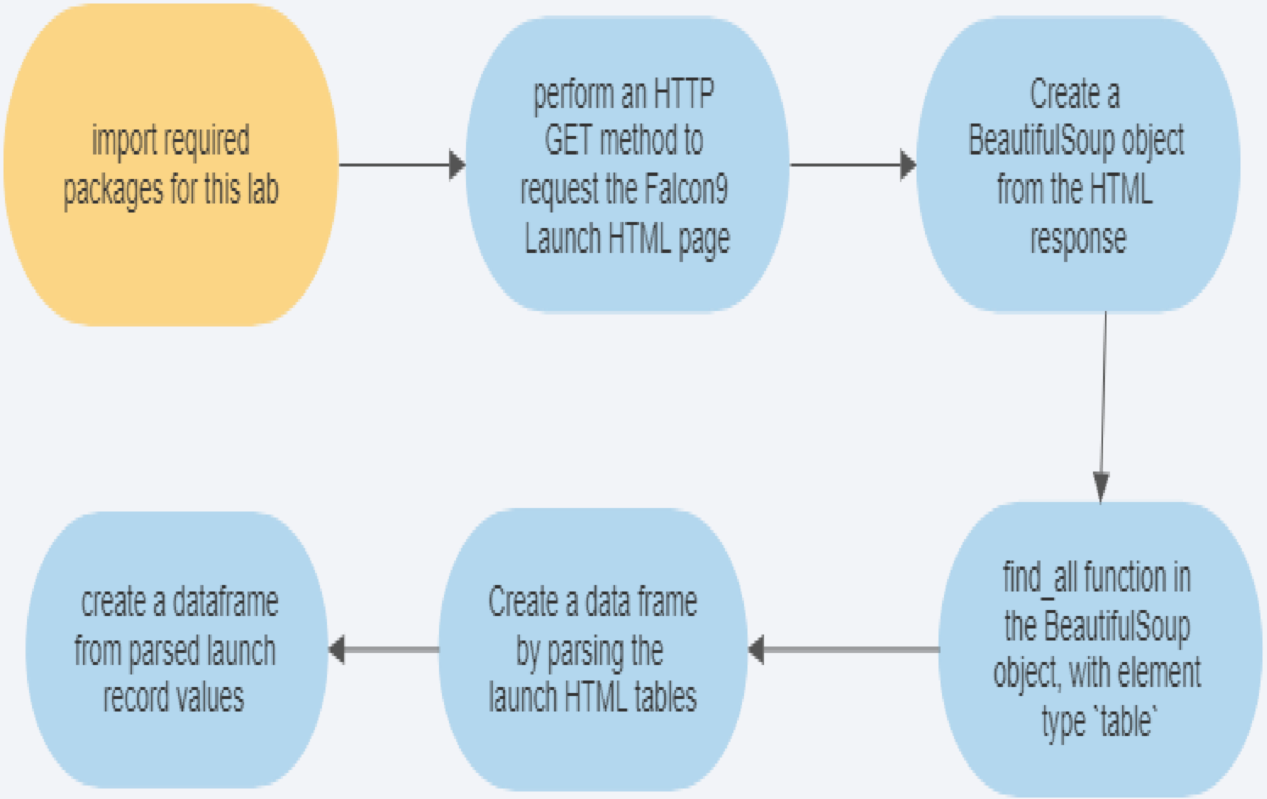
Data wrangling , Missing value

- using the mean( )
- use the mean and the replace() function to replace values

Data Collection - Scraping

1. Request the Falcon9 Launch Wiki page from its URL
2. Extract all column/variable names from the HTML table header
3. Create a data frame by parsing the launch HTML tables

https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%201/jupyter-labs-webscraping.ipynb



import required packages for this lab

perform an HTTP GET method to request the Falcon9 Launch HTML page

Create a BeautifulSoup object from the HTML response

find_all function in the BeautifulSoup object, with element type `table`

Create a data frame by parsing the launch HTML tables

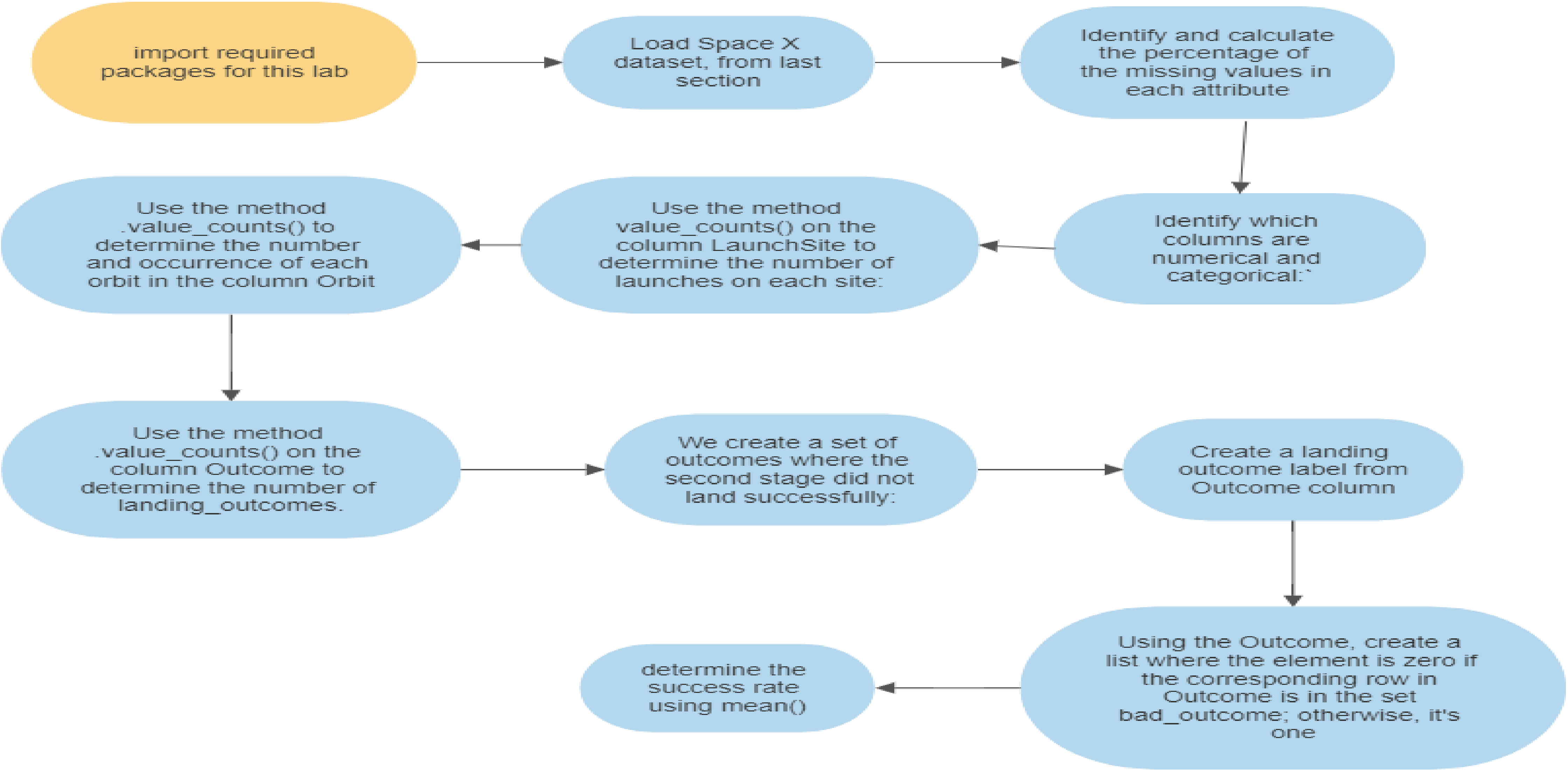
create a dataframe from parsed launch record values

Data wrangling 1

1. Calculate the number of launches on each site
2. Calculate the number and occurrence of each orbit
3. Calculate the number and occurrence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column

https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%201/labs-jupyter-spacex-Data%20wrangling.ipynb

Data wrangling 2



import required packages for this lab

Load Space X dataset, from last section

Identify and calculate the percentage of the missing values in each attribute

Identify which columns are numerical and categorical:`

Use the method value_counts() on the column LaunchSite to determine the number of launches on each site:

Use the method .value_counts() to determine the number and occurrence of each orbit in the column Orbit

Use the method .value_counts() on the column Outcome to determine the number of landing_outcomes.

We create a set of outcomes where the second stage did not land successfully:

Create a landing outcome label from Outcome column

Using the Outcome, create a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one

determine the success rate using mean()

EDA with Data Visualization

- 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 due to the fact that SpaceX can reuse the first stage.
- I used catplot and lineplot to understand the relation between many factors such as Flight Number and Launch Site , Payload and Launch Site , success rate of each orbit type and many other relations to improve the success of the mission.
- Please if there is a problem in viewing notebook copy link and past it in :

https://nbviewer.jupyter.org/

- https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%202/eda-dataviz.ipynb

EDA with SQL 1

SQL queries that I performed on SpaceX dataset

- 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. Use a subquery

EDA with SQL 1

- 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

https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%202/eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities

Adding previous objects is to give information about racket launches more clear to those who don’t have geographic background and make the map more informative and clear.

https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%203/launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

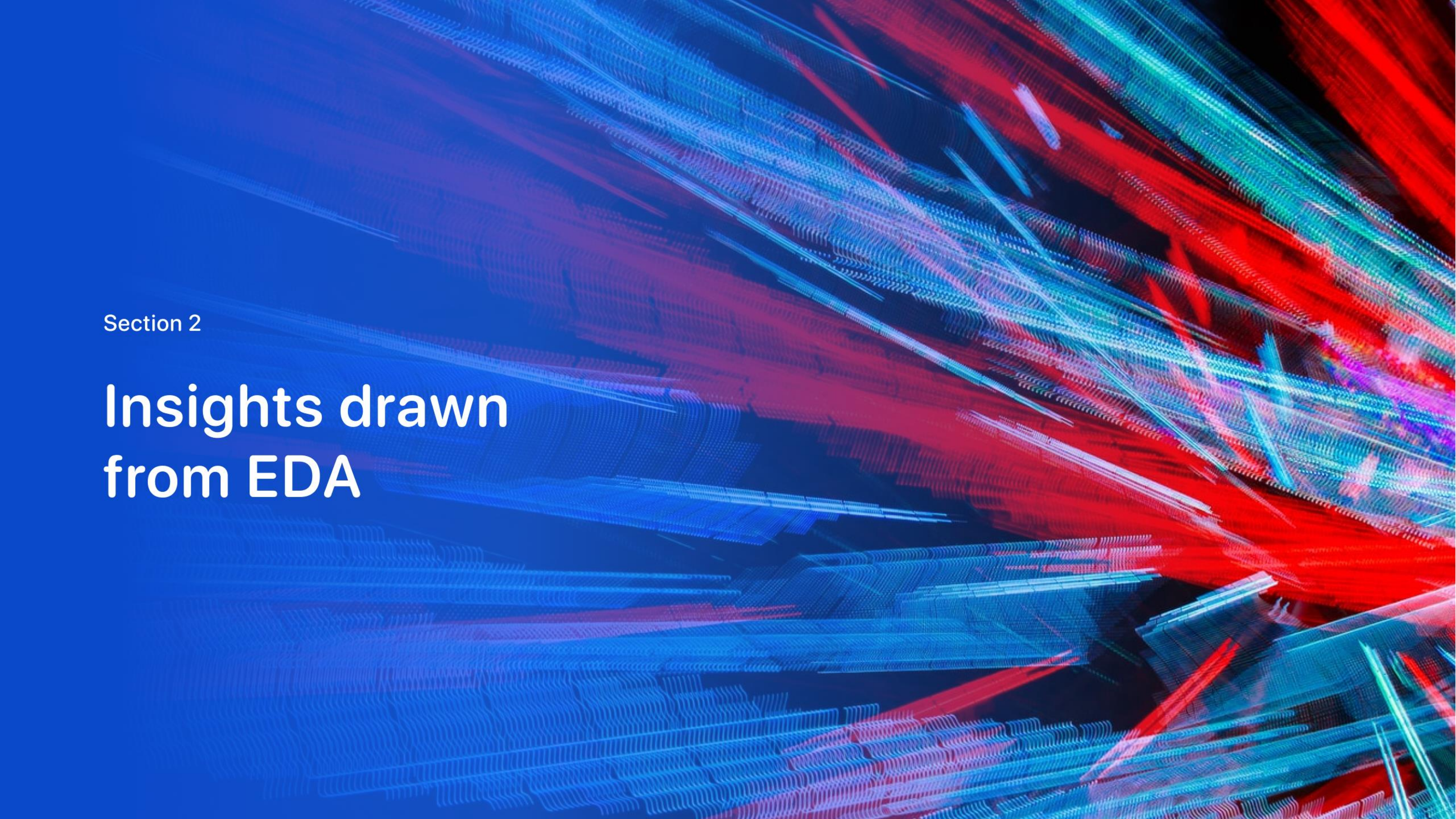
Predictive Analysis (Classification)

- Importing related libraries
- Load data and clean it
- Standardize data
- Split data into training and test set
- Apply different algorithms on training set and then calculate the accuracy on test data using the method **score ( )**

https://github.com/MohamadAbdulrahman/Applied-Data-Science-Capstone-IBM/blob/master/week%204/Machin%20Learning.ipynb

Results

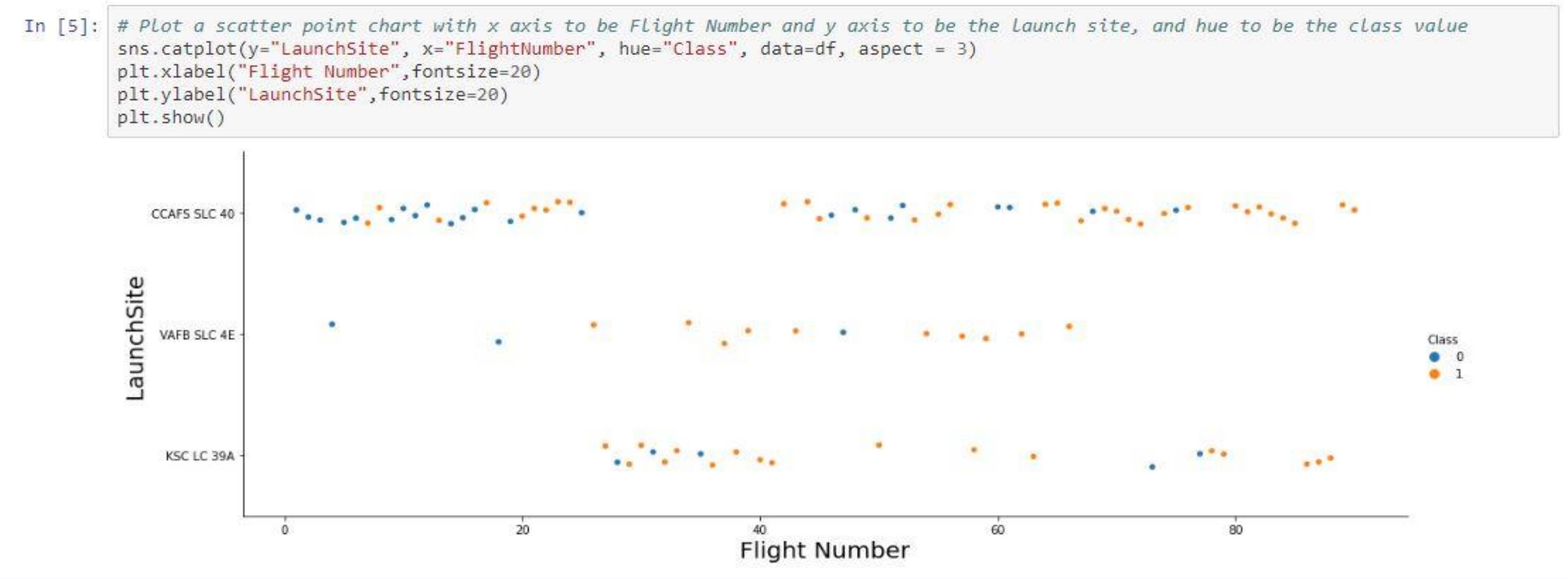
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

Insights drawn from EDA

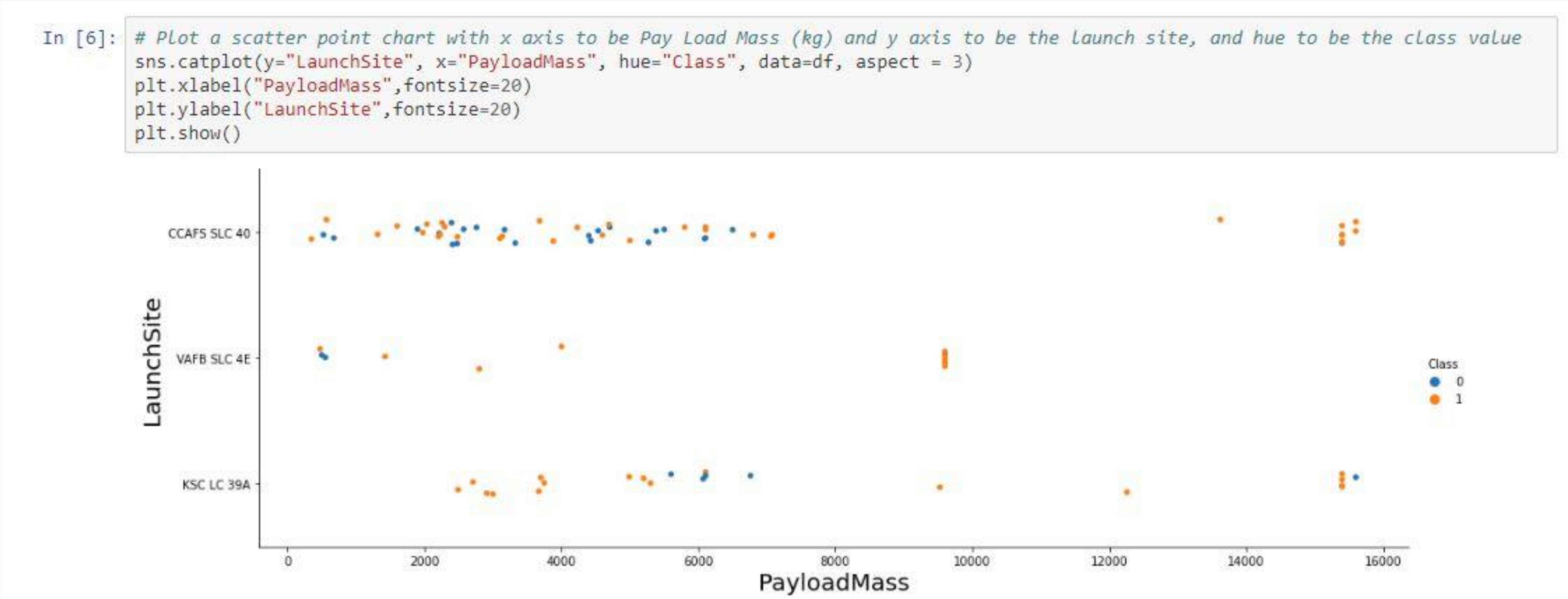
Flight Number vs. Launch Site



```
In [5]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
        sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 3)
        plt.xlabel("Flight Number",fontsize=20)
        plt.ylabel("LaunchSite",fontsize=20)
        plt.show()
```

We can see that CCAFC SLC 40 have lower success rate than other launch sites . Especially at the first launches , then after gaining experience from previous launches , the rate improved

Payload vs. Launch Site



In [6]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 3)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

Like previous chart the most failed launches came from CCAFS SLC 40 , regardless the payload weight .

Success Rate vs. Orbit Type

The most success rate when launching to are :

ES-L1

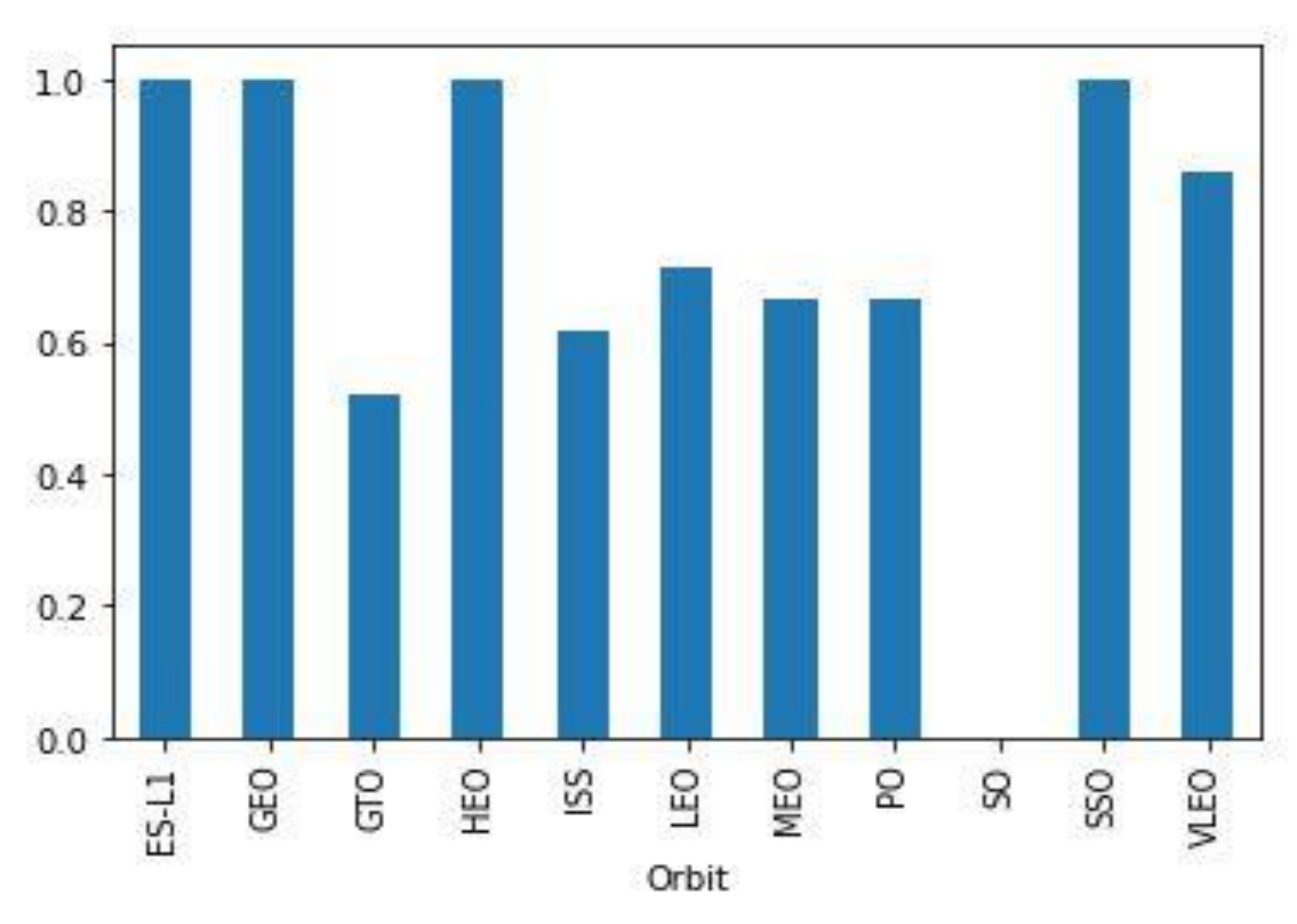
GEO

HEO

SSO

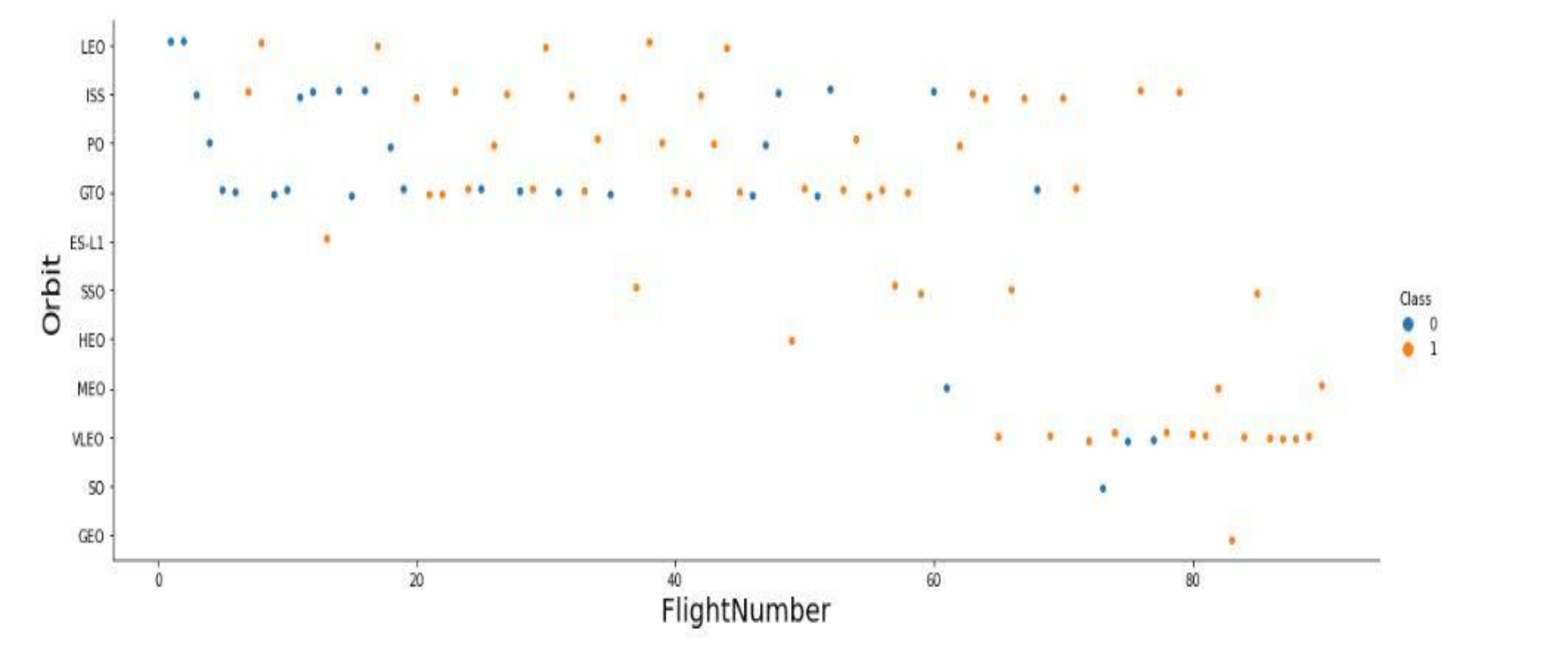
And less success rate are:

SO and GTO



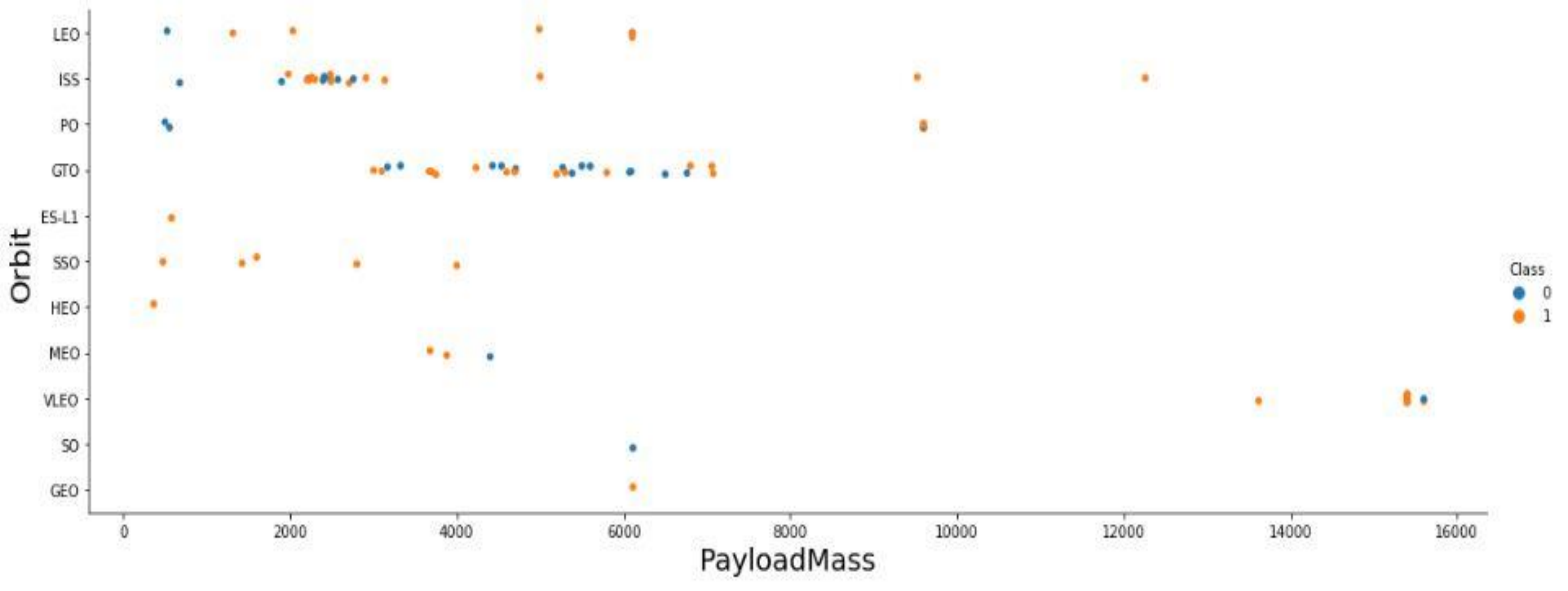
Flight Number vs. Orbit Type

we see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



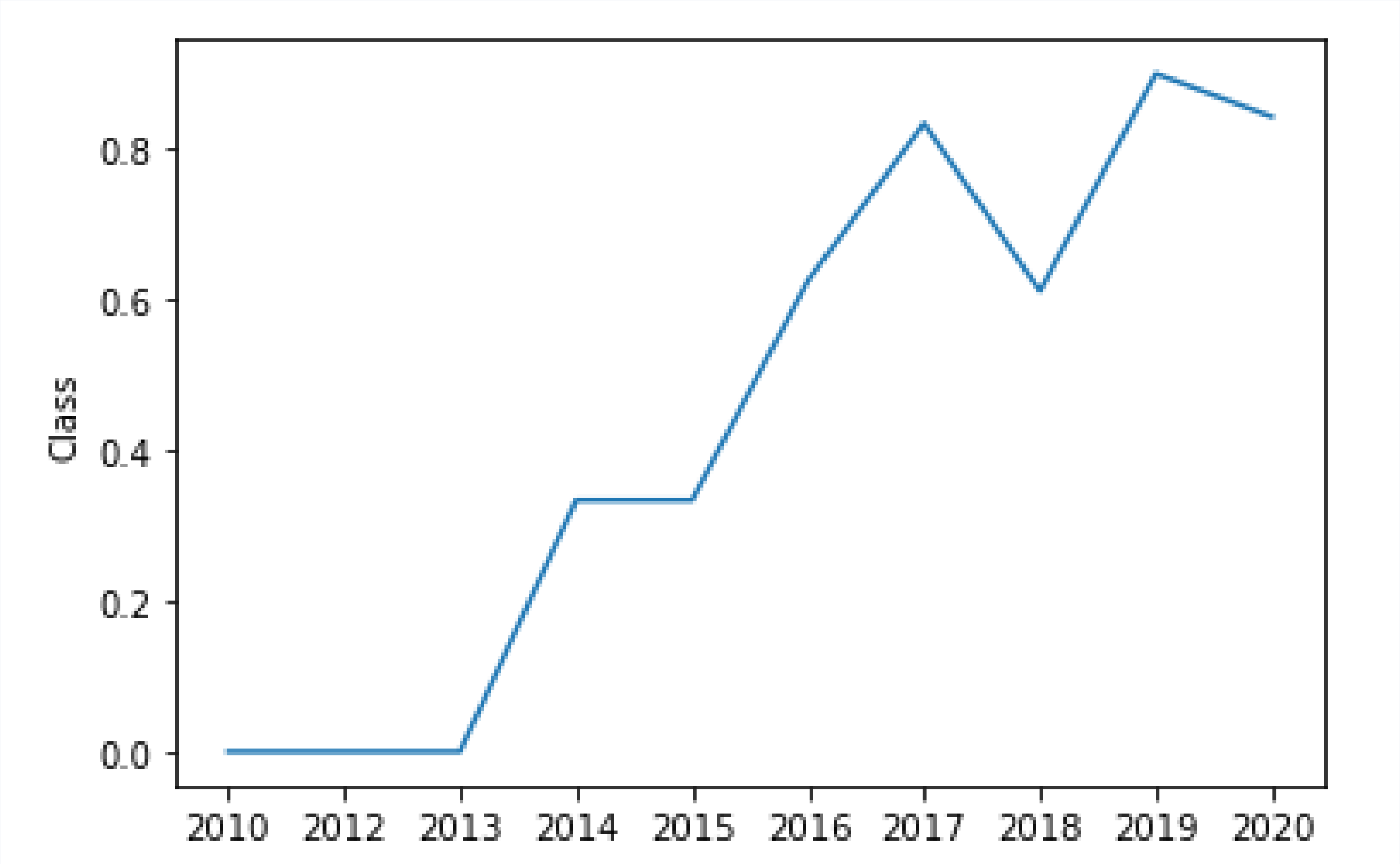
Payload vs. Orbit Type

observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

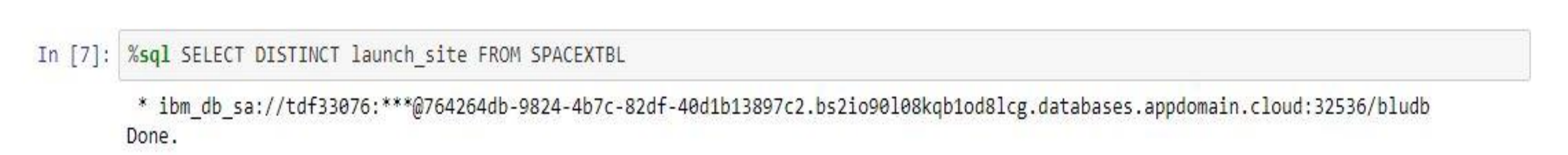


Launch Success Yearly Trend

observe that the success rate since 2013 kept increasing till 2020

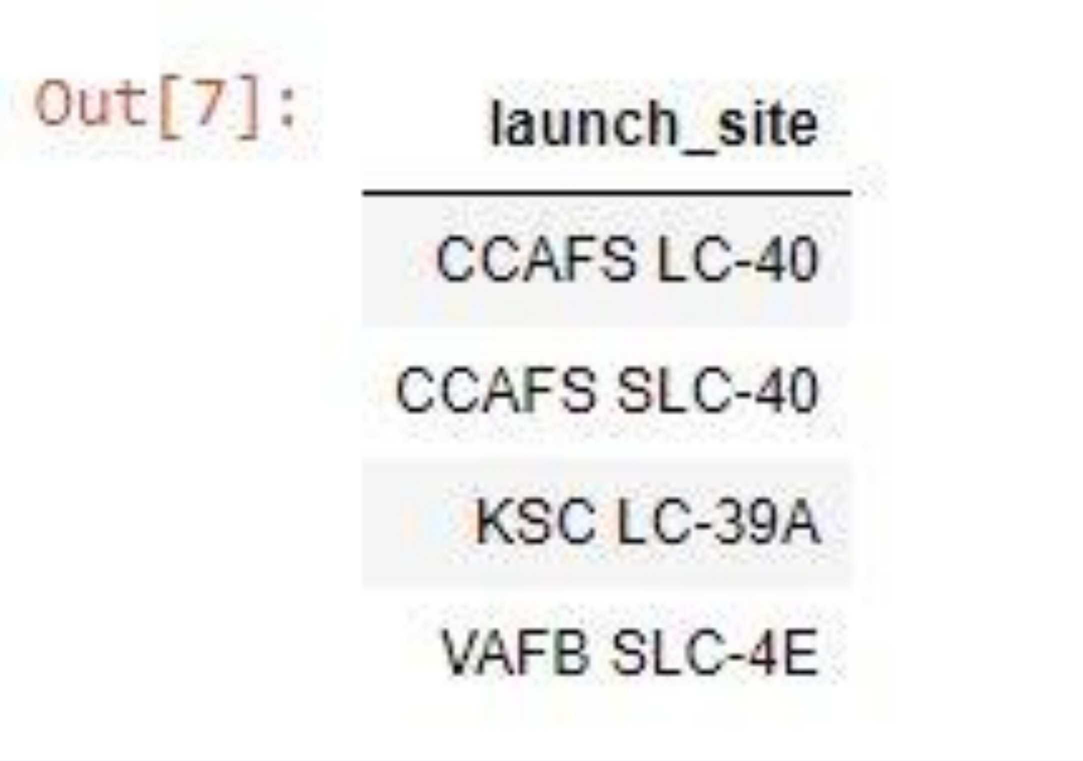


All Launch Site Names



```
In [7]: %sql SELECT DISTINCT launch_site FROM SPACEXTBL

 * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

The idea here is to use keyword Distinct with column (Launch_site)

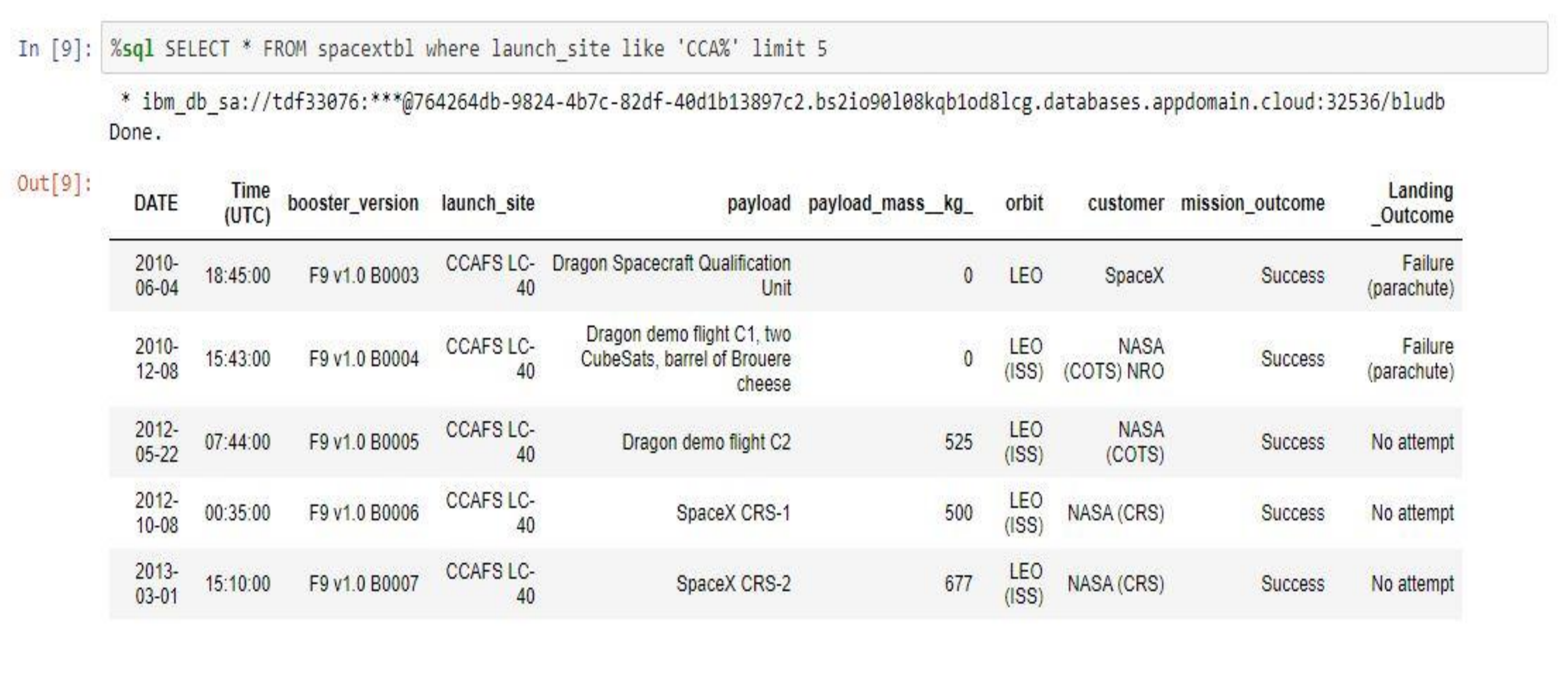


Out[7]:

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

Launch Site Names Begin with 'CCA'

Using of keyword (like) and limit 5 to show only 5 records



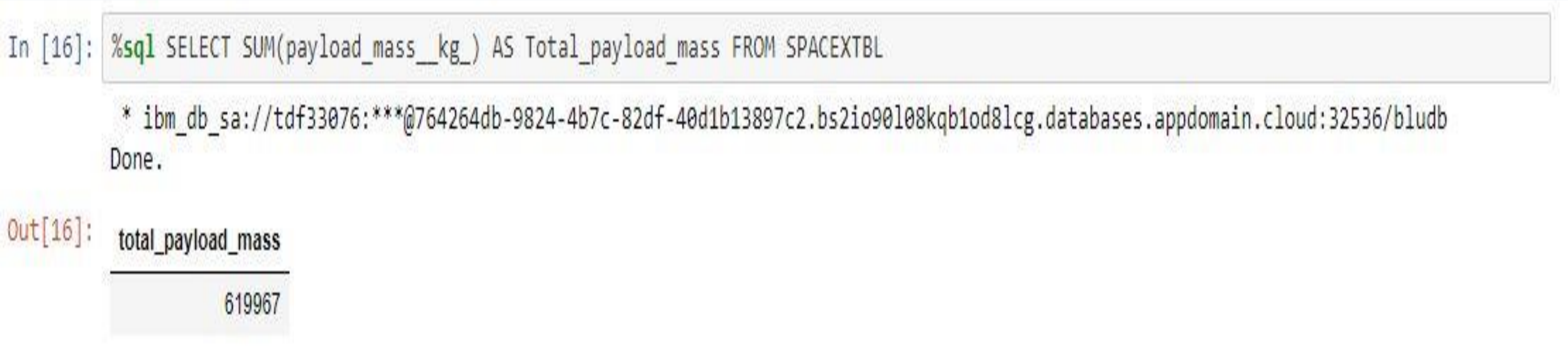
```
In [9]: %sql SELECT * FROM spacextbl where launch_site like 'CCA%' limit 5

 * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[9]:

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

Total Payload Mass

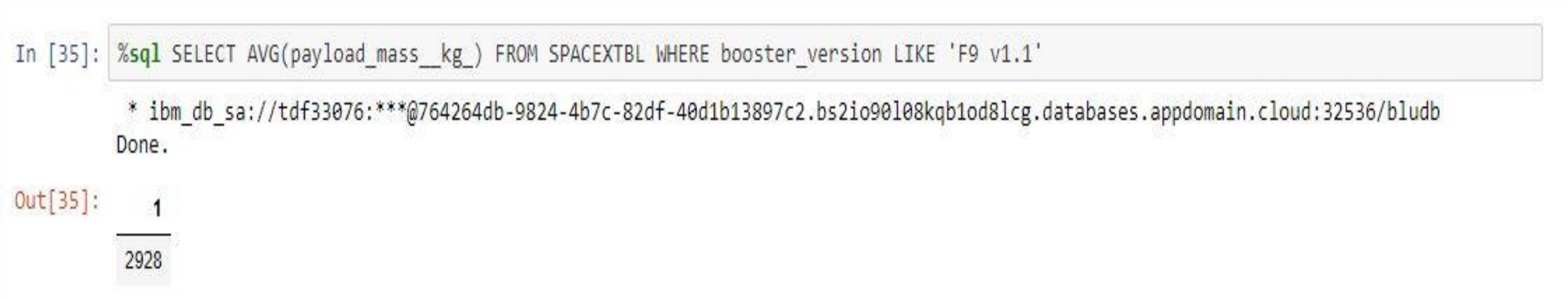
Using of keyword (SUM) with the column



```
In [16]: %sql SELECT SUM(payload_mass__kg_) AS Total_payload_mass FROM SPACEXTBL

          * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
         Done.

Out[16]: total_payload_mass
         ------------------
                     619967
```

Average Payload Mass by F9 v1.1

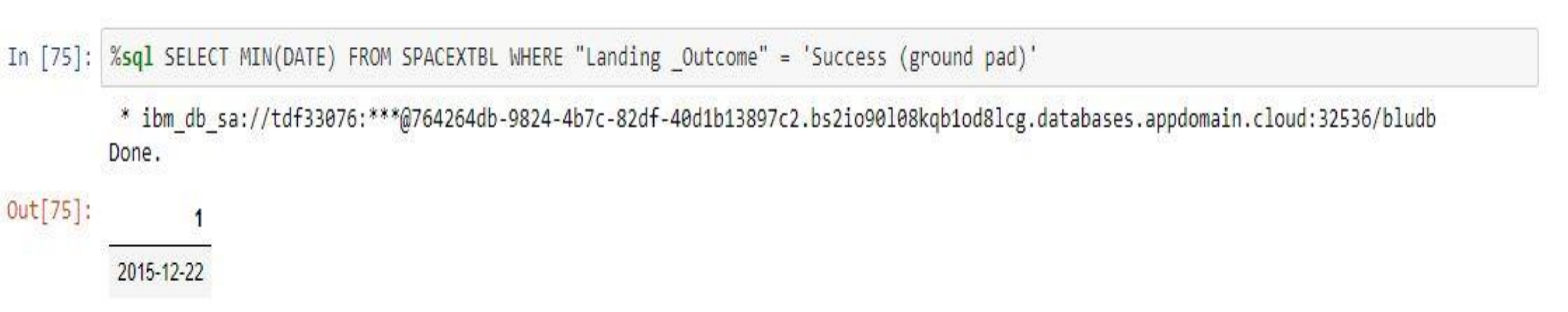
Using of keyword (AVG) and keyword (LIKE)



```
In [35]: %sql SELECT AVG(payload_mass__kg_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1'

          * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
         Done.

Out[35]:    1
         ------
          2928
```

First Successful Ground Landing Date

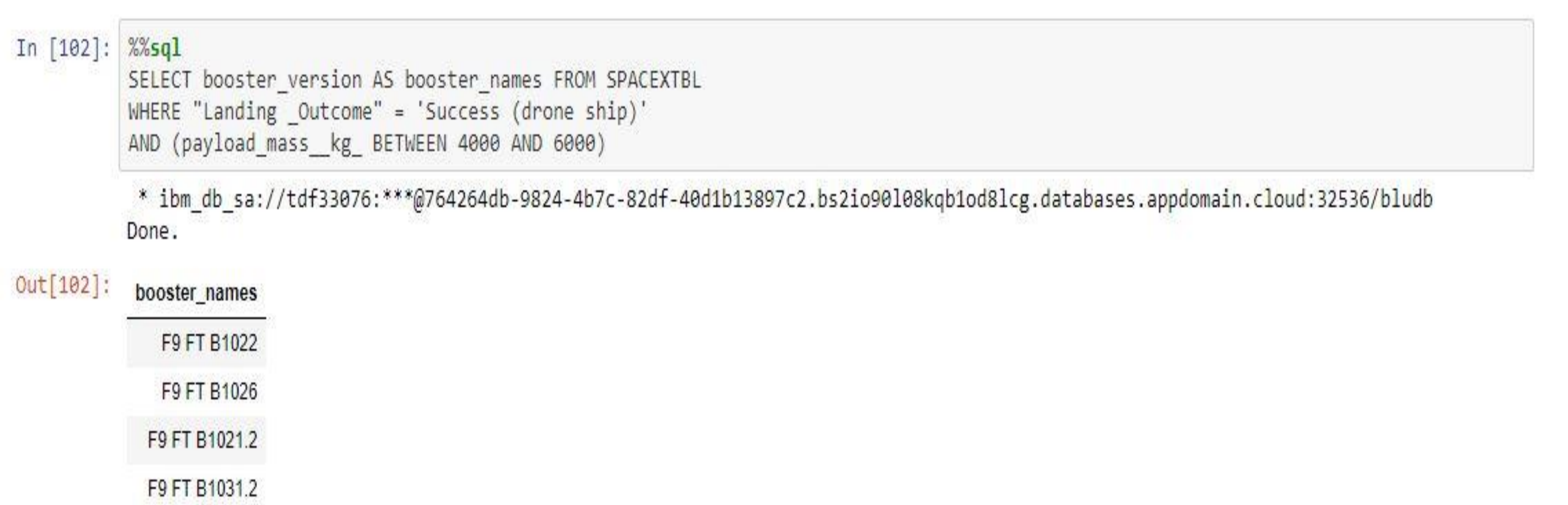
To select first successful ground landing we use keyword (MIN) after getting the result form Landing outcome where the success on ground pad



```
In [75]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)'

          * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
         Done.

Out[75]:          1
         ----------
         2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

The idea is using keyword (BETWEEN)



In [102]:

```
%%sql
SELECT booster_version AS booster_names FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (drone ship)'
AND (payload_mass__kg_ BETWEEN 4000 AND 6000)
```

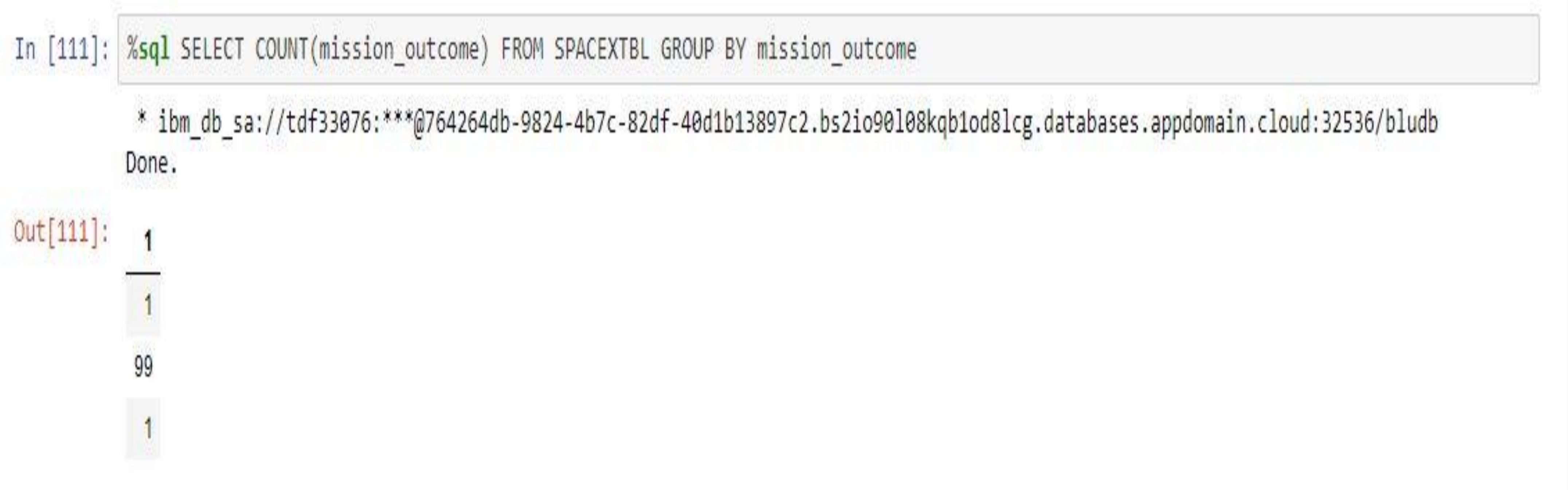
```
 * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[102]:

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

Total Number of Successful and Failure Mission Outcomes

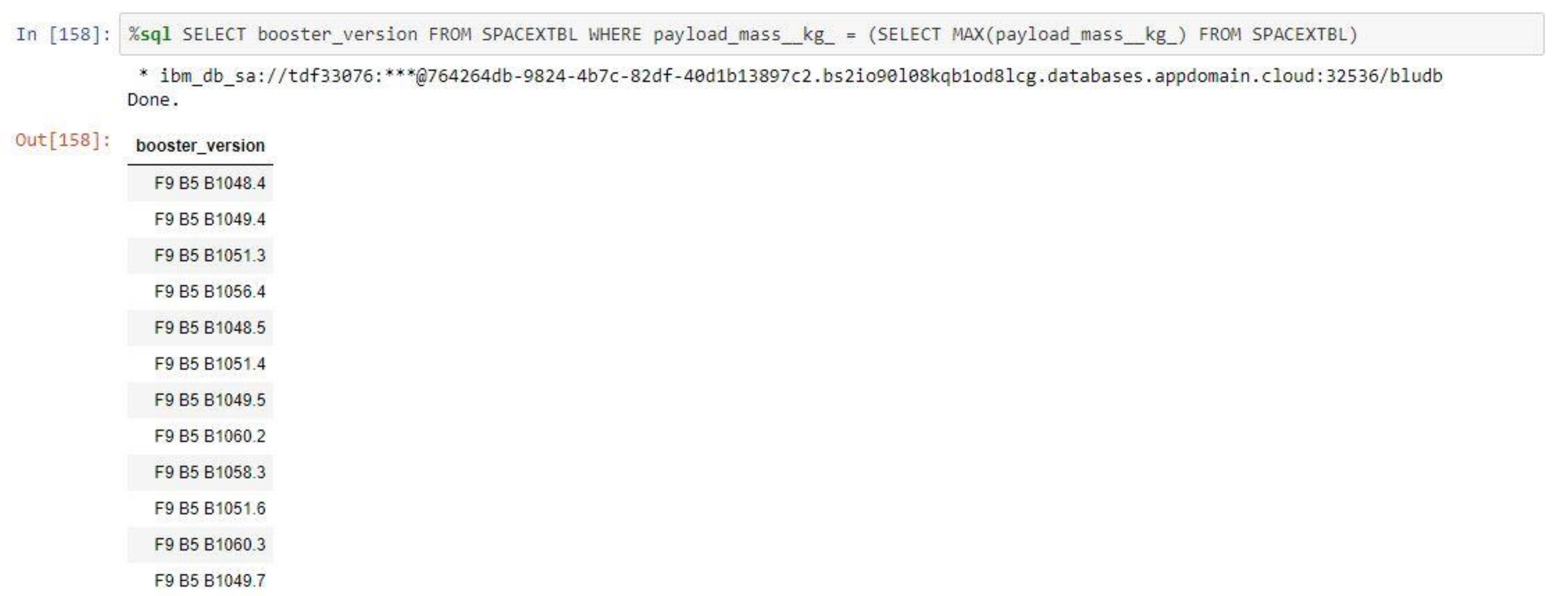
Using of keyword (GROUP BY)



```
In [111]: %sql SELECT COUNT(mission_outcome) FROM SPACEXTBL GROUP BY mission_outcome

 * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[111]:  1
           1
          99
           1
```

Boosters Carried Maximum Payload

The idea here is using subquery

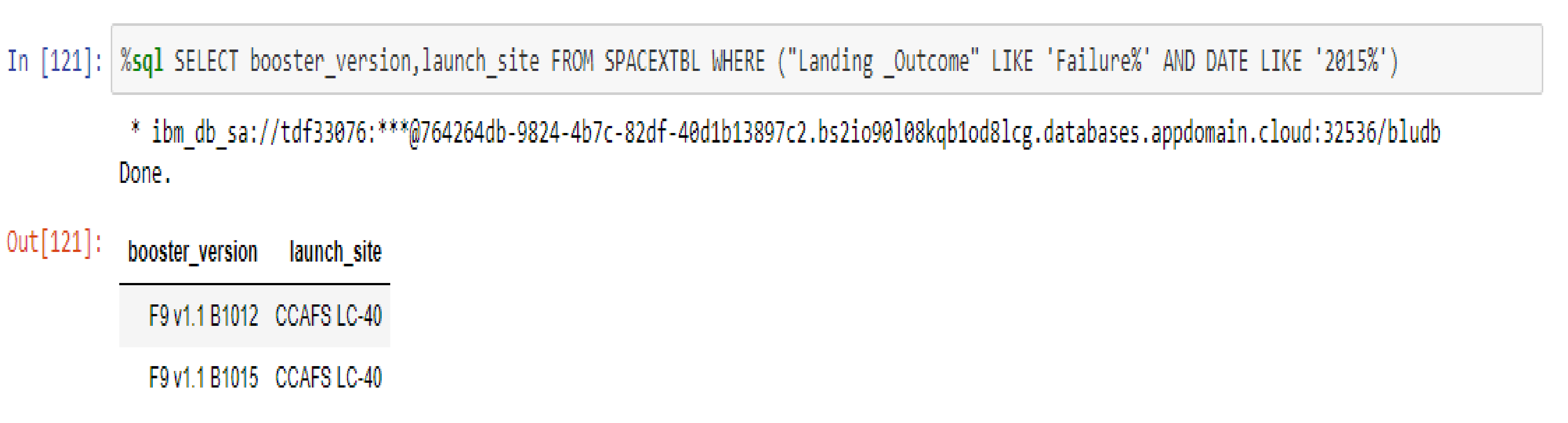


```
In [158]: %sql SELECT booster_version FROM SPACEXTBL WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)

 * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[158]:
```

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

Using keyword (LIKE) with two columns



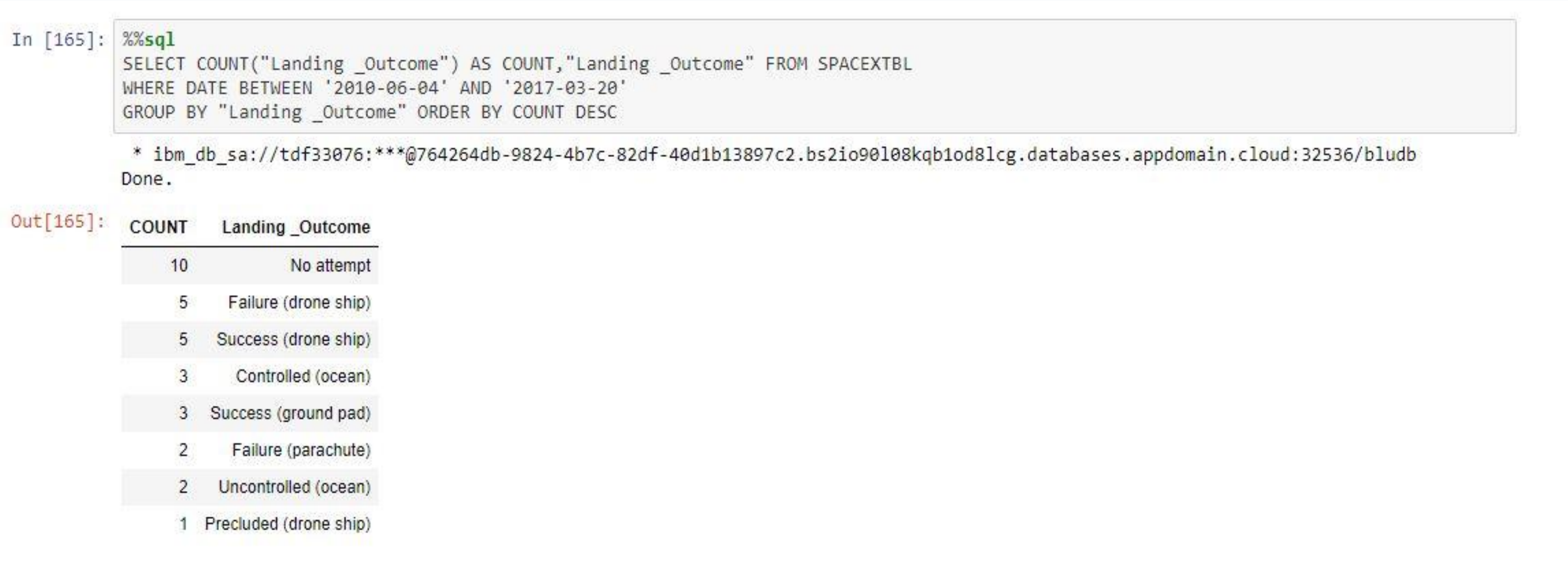
```
In [121]: %sql SELECT booster_version,launch_site FROM SPACEXTBL WHERE ("Landing _Outcome" LIKE 'Failure%' AND DATE LIKE '2015%')

 * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[121]:

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

Using of keywords (AND) , (GROUP BY) and (ORDER BY)



```
In [165]: %%sql
          SELECT COUNT("Landing _Outcome") AS COUNT,"Landing _Outcome" FROM SPACEXTBL
          WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
          GROUP BY "Landing _Outcome" ORDER BY COUNT DESC

           * ibm_db_sa://tdf33076:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
          Done.
```

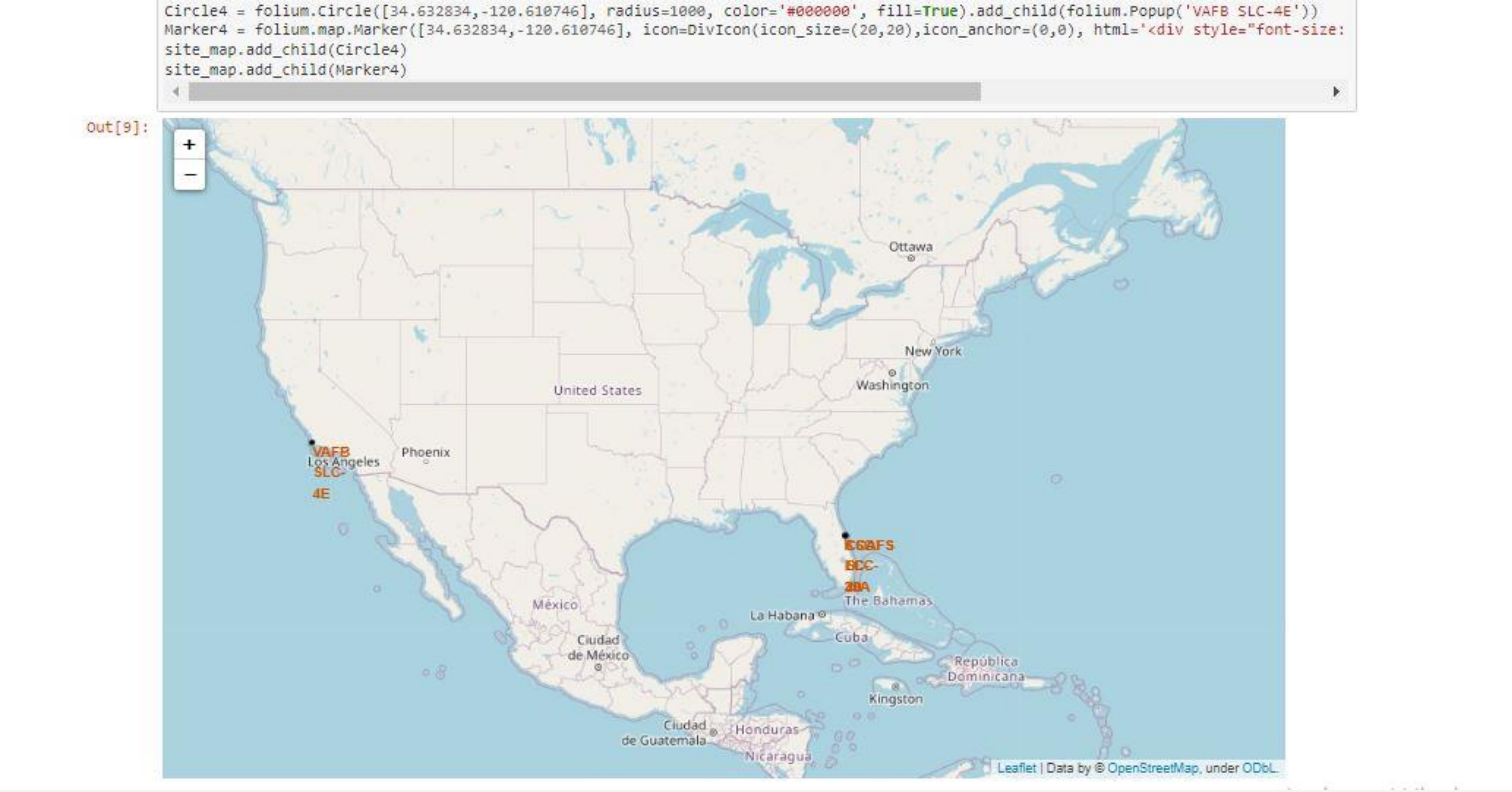
Out[165]:

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

Section 4

Launch Sites Proximities Analysis

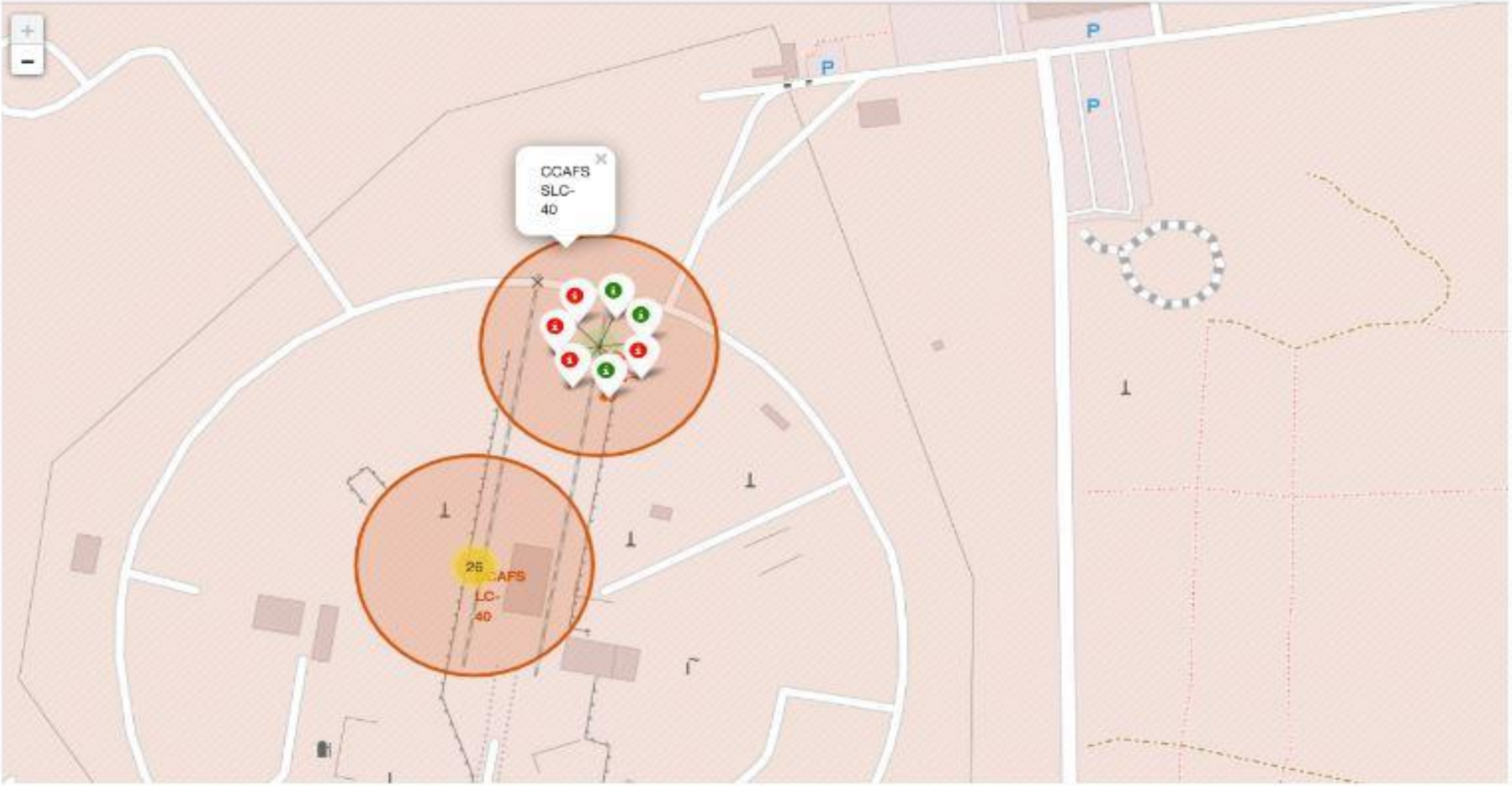
<Folium Map Screenshot 1>



```
Circle4 = folium.Circle([34.632834,-120.610746], radius=1000, color='#000000', fill=True).add_child(folium.Popup('VAFB SLC-4E'))
Marker4 = folium.map.Marker([34.632834,-120.610746], icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size:
site_map.add_child(Circle4)
site_map.add_child(Marker4)
```

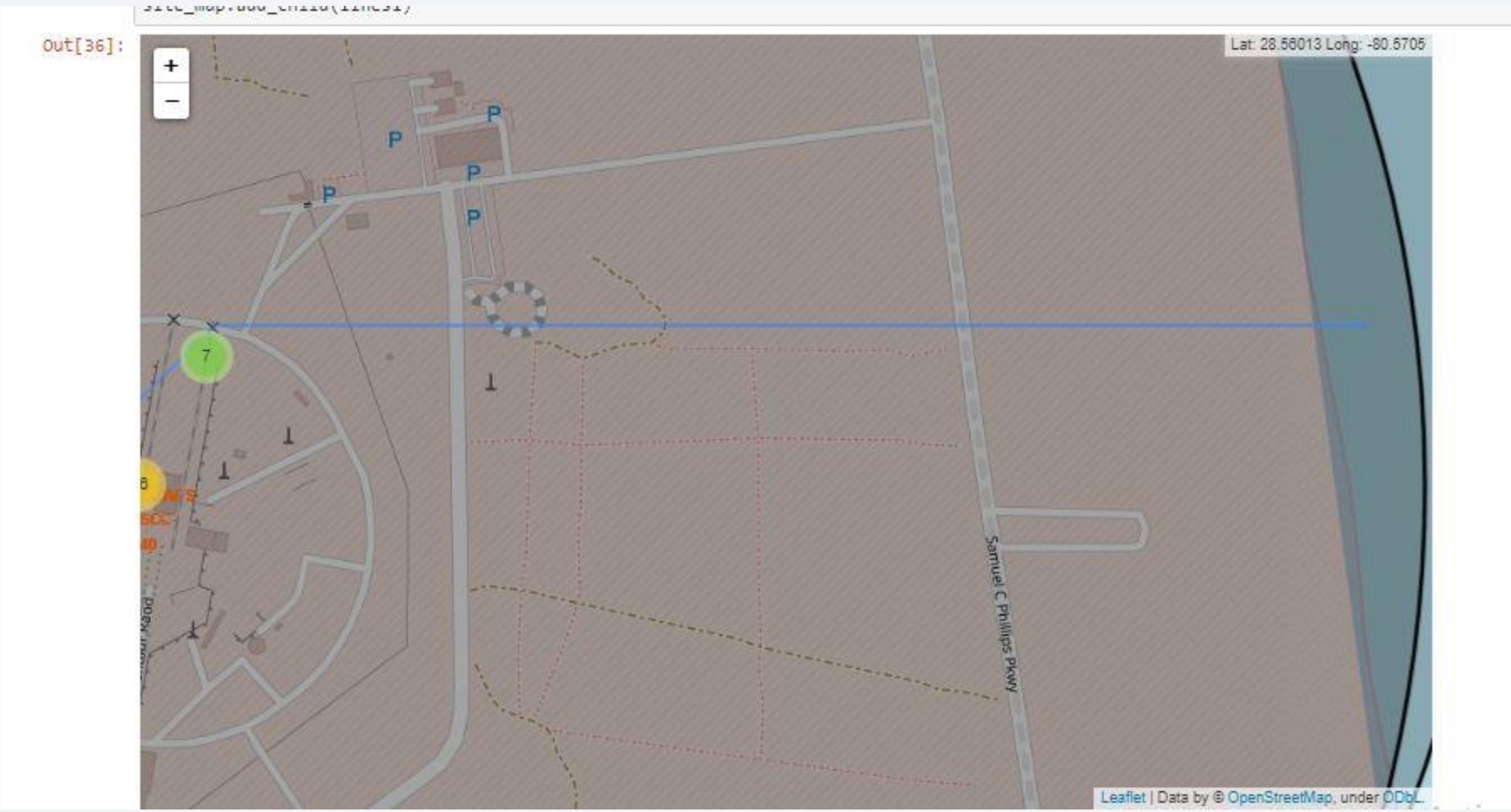
Out[9]:

<Folium Map Screenshot 2>



From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

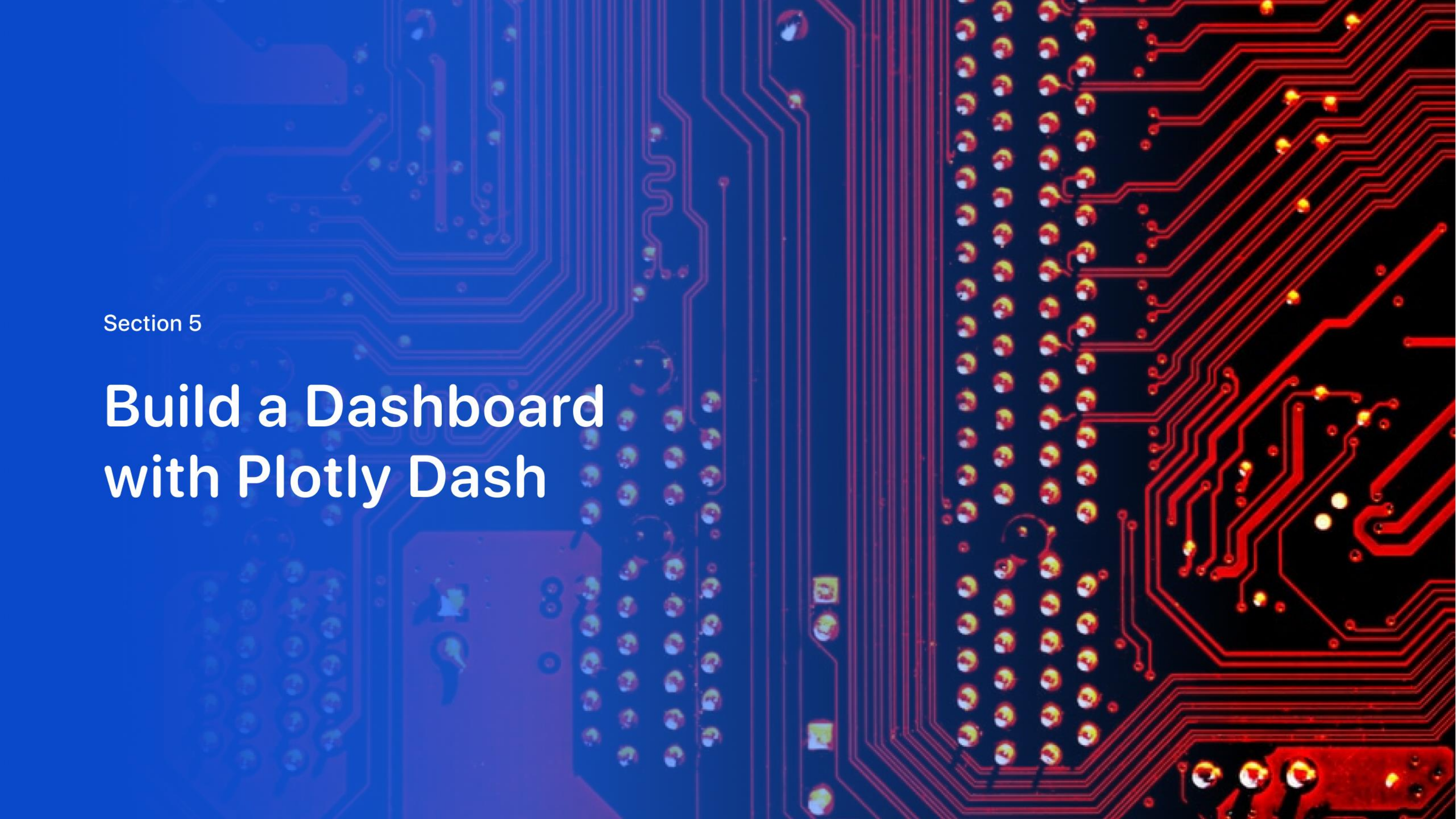
<Folium Map Screenshot 3>



```
site_map.add_child(lines1)
```

Out[36]:

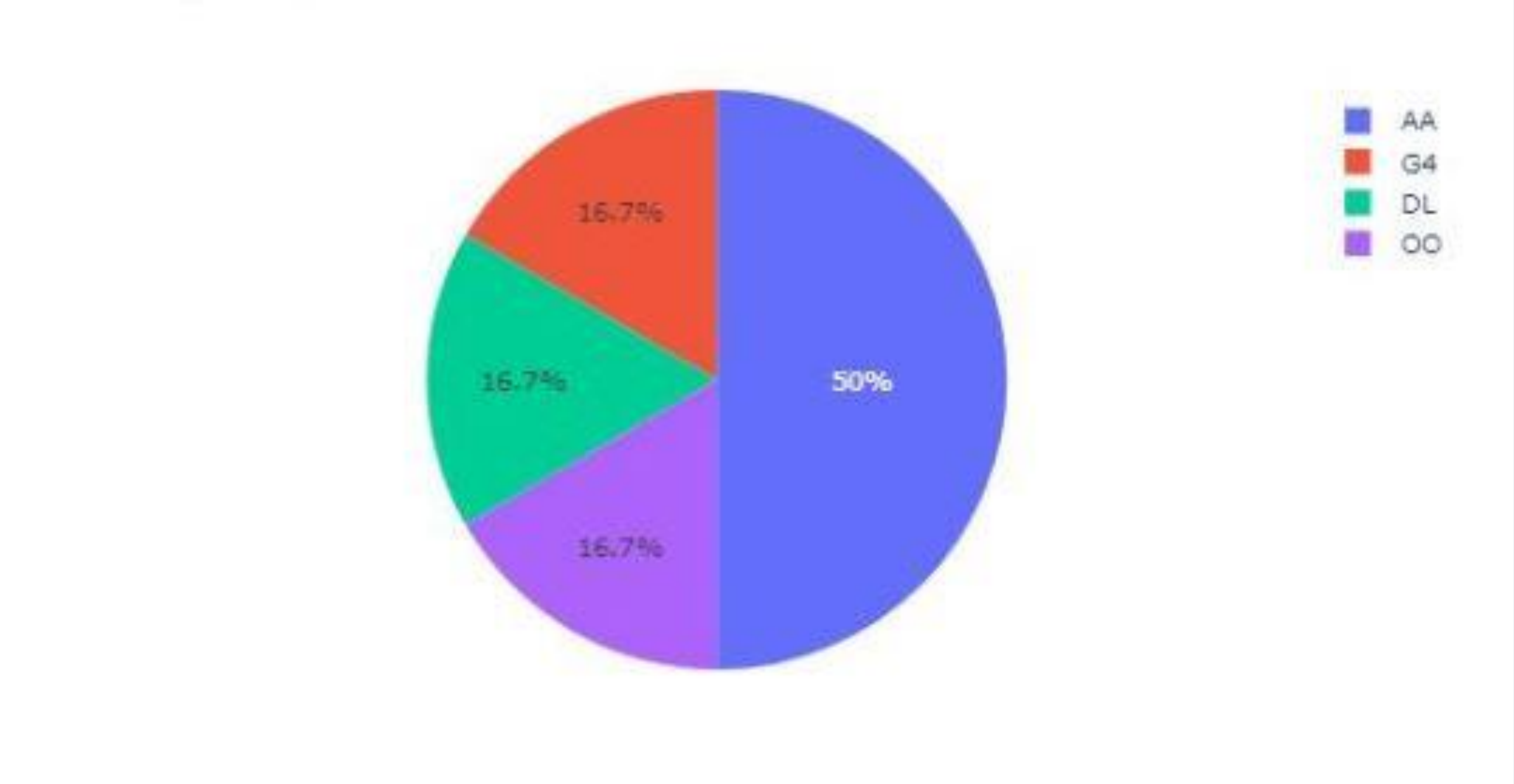
+
−
Lat: 28.56013 Long: -80.5705
P
P
P
P
P
7
Samuel C Phillips Pkwy
Leaflet | Data by © OpenStreetMap, under ODbL.



Section 5

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

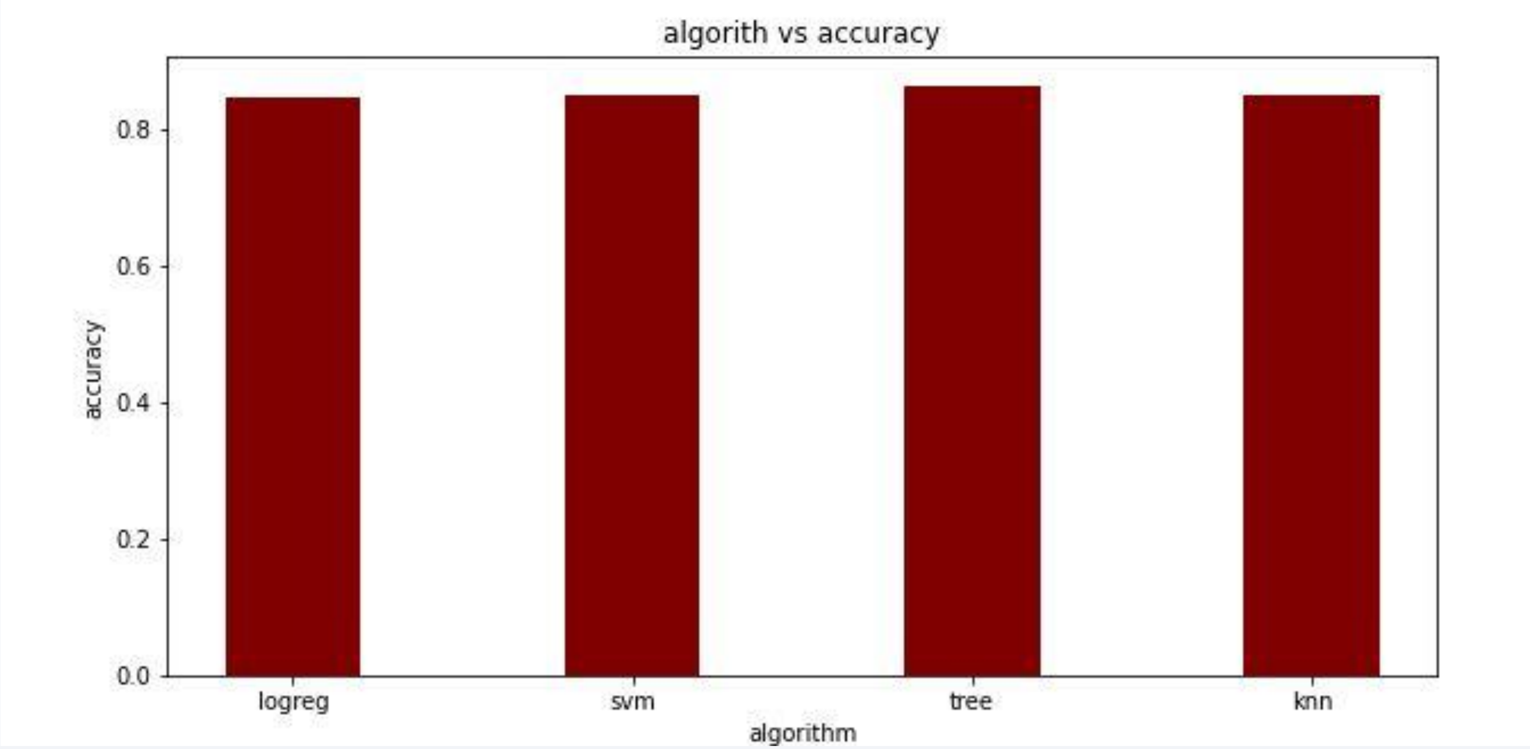

16.7%
16.7%
16.7%
50%
AA
G4
DL
OO

Section 6

Predictive Analysis (Classification)

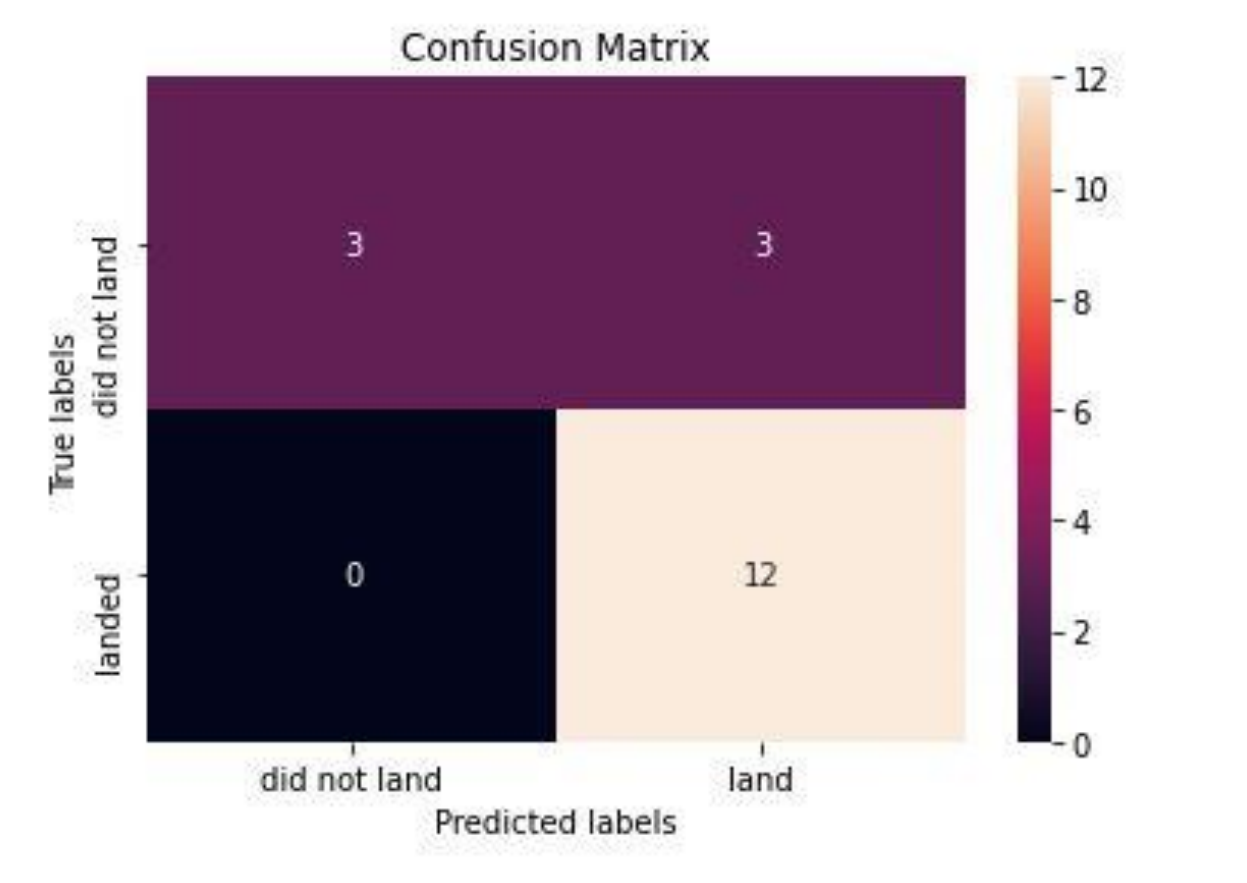
Classification Accuracy

as we see, all algorithms have the same results , so in this case we can you use any one of them



Confusion Matrix

we see that all algorithms give the same result and can distinguish between the different classes. We see that the major problem is false positives.



Conclusions

- Since we have a lot of info about data, we see that all algorithms give almost same accuracy
- The accuracy is high but I think still need improvements
- We have to clean data and remove correlation between some columns

Thank you!