IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

A Rakesh

11/06/2023

Outline

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

Executive Summary

- We have extracted the data from different API.
  - Created custom functions to extract required data from the API.
  - Worked of missing values, removed unnecessary features.
  - Did analysis on the class feature and consolidate it into binary classification.
  - Visualized the data and analyzed the relation between different features.
  - Created different ML models and identified the best accuracy score.
- This is a binary Classification model and we are able to create a model which as with 88.92% success rate.

Introduction

- SpaceX advertises Falcon 9 rocket launches at a cost of 62 million dollars and other providers cost upward of 165 million dollars.
- Much of the savings for SpaceX is from reuse of first stage.
- If we can determine the first stage success land, we can determine the cost of a launch and bid against SpaceX.

Section 1

Methodology

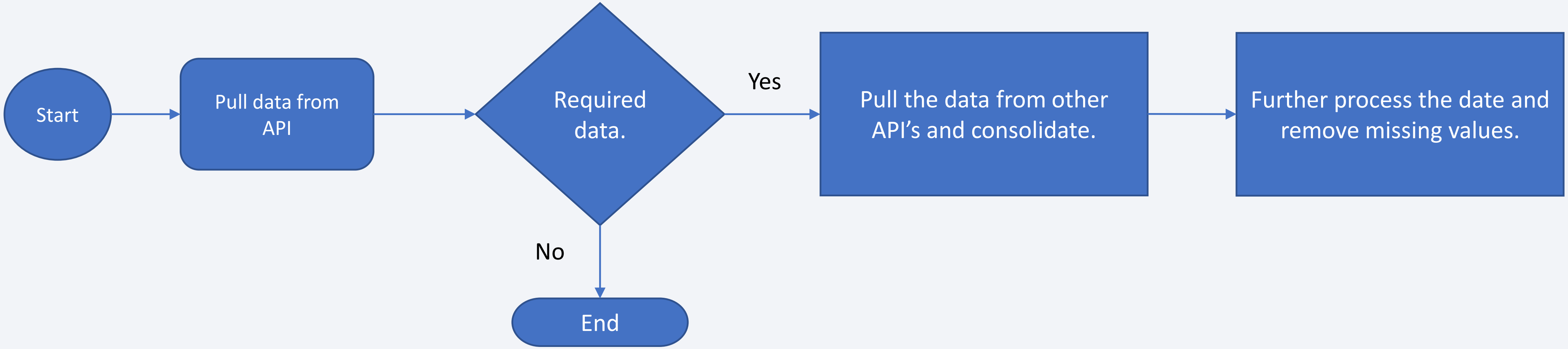
Methodology

Executive Summary

- Data collection methodology:
  - We have collected the data from multiple SpaceX API's, Web scrapping..
- Perform data wrangling
  - Identify the different types of landing class and simplifying the outcome in 2 classes.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Identified different models and used GridSearchCV to identify best parameters.

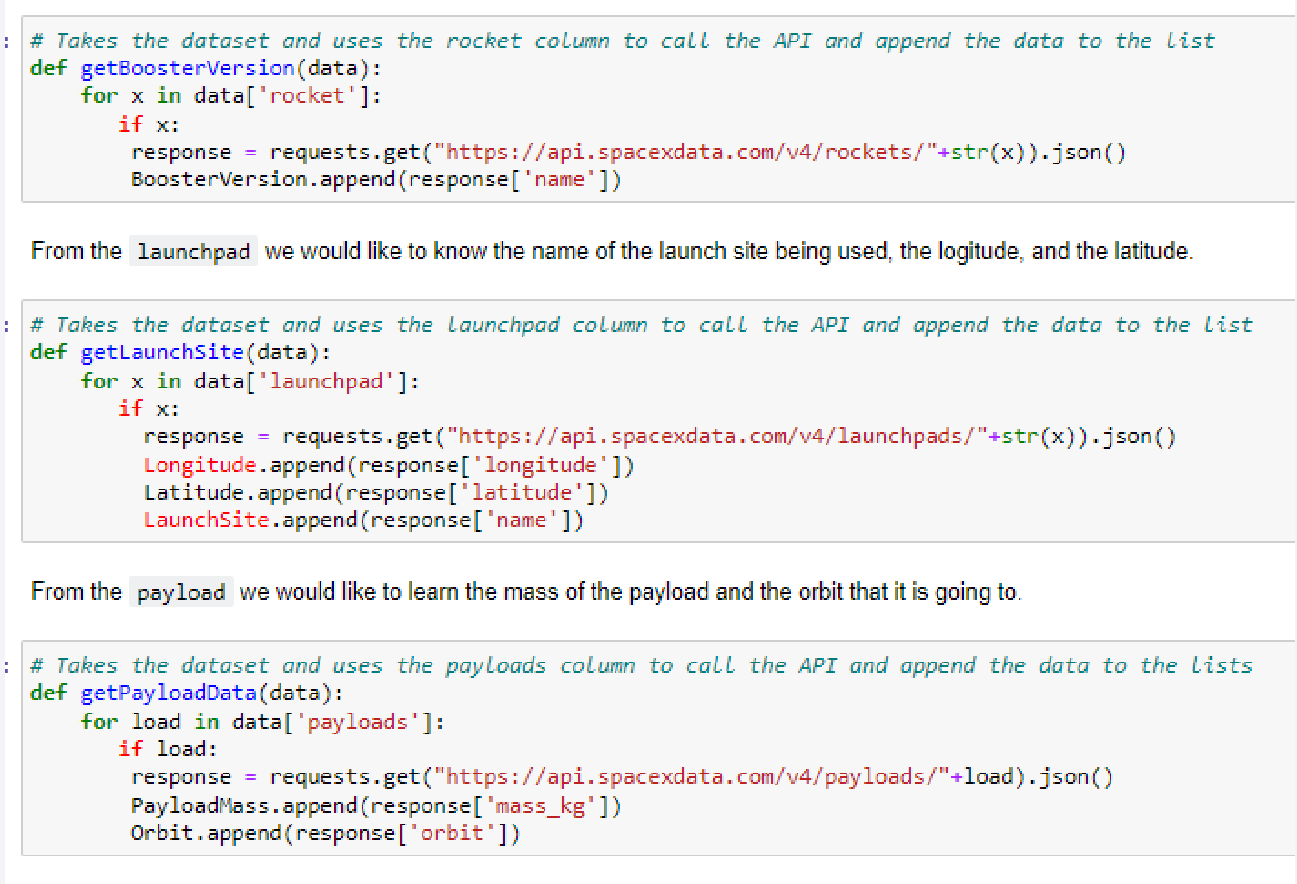
Data Collection

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


Start
Pull data from API
Required data.
Yes
Pull the data from other API's and consolidate.
Further process the date and remove missing values.
No
End

Data Collection – SpaceX API

- This is the data collection with SpaceX REST calls using key phrases.
- Add the GitHub URL of the completed SpaceX API calls notebook (https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W1_1_jupyter-labs-spacex-data-collection-api%20_worked.ipynb)



```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

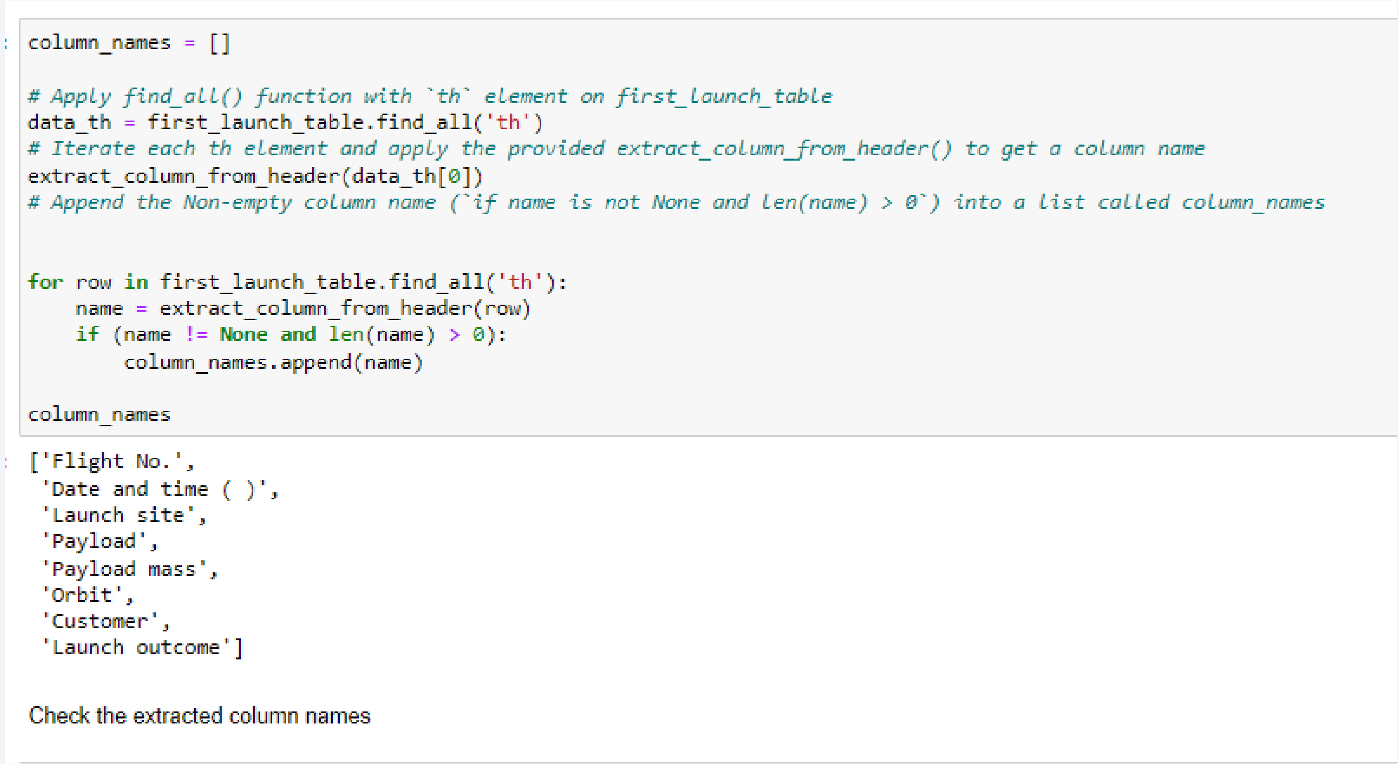
From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

Data Collection - Scraping

- Webscraping process using key phrases and flowcharts

- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W1_2_jupyter-labs-webscraping_worked.ipynb



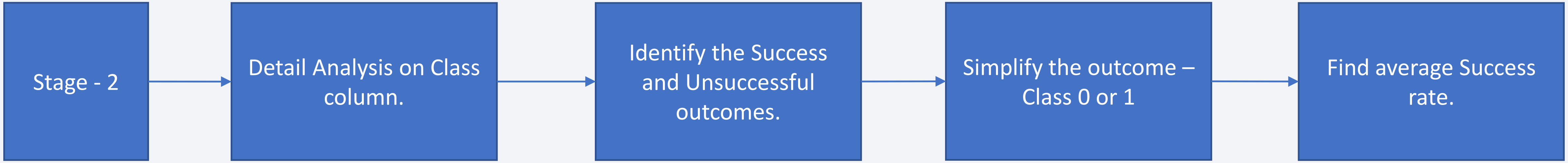
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
data_th = first_launch_table.find_all('th')
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
extract_column_from_header(data_th[0])
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names


for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)

column_names
```

```
['Flight No.',
 'Date and time ( )',
 'Launch site',
 'Payload',
 'Payload mass',
 'Orbit',
 'Customer',
 'Launch outcome']
```

Check the extracted column names

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W1_3_IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite_worked.ipynb



Stage - 2

Detail Analysis on Class column.

Identify the Success and Unsuccessful outcomes.

Simplify the outcome – Class 0 or 1

Find average Success rate.

EDA with Data Visualization

- We have used below plots:
    - Scatter:
        - To find relation between Flight number and Launch site.
        - To find relation between Pay load and Launch site
        - To find relation between Flight number and Orbit.
        - To find relation between Pay load and Orbit.
    - Bar chart:
        - To find the success rate of different Orbits.
    - Line chart
        - To find success rate in different years and the trend.
- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W2_2_IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite_worked.ipynb

EDA with SQL

- With the help of SQL queries we try to find below information:
  - Different types of Launch sites.
  - Different payloads carried by a specific Customer.
  - Average payload carried my F9 v1.1 booster.
  - Date of first successful Landing.
  - Different success drones with specific Pay load.
  - Total number of Successful and Failures.
  - Booster versions which carried the highest Pay load.
- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W2_1_jupyter-labs-eda-sql-coursera_sqllite_WORKED.ipynb

Build an Interactive Map with Folium

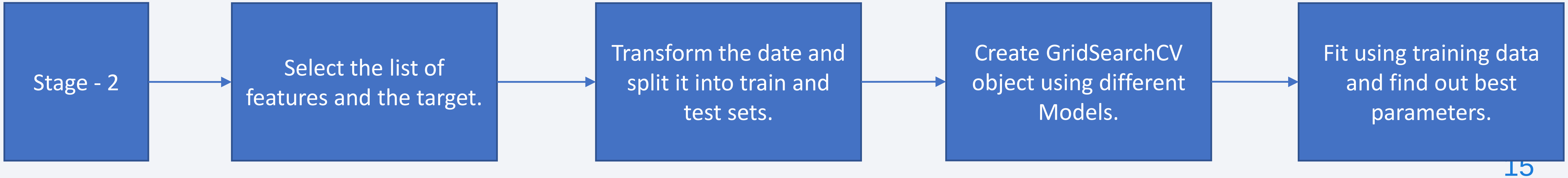
- We have added below in folium map:
  - Circle for each launch site in data frame launch sites.
  - Marker coordinate with a icon showing its name.
- We have added those objects for below analysis:
  - Mark all launch sites on a map
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site to its proximities
- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W3_1_IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite_worked.ipynb

Build a Dashboard with Plotly Dash

- We have created below plots:
  - Pie Plot:
    - To view the different launch sites and their success rate.
    - If we select a location from the list of locations. We get a chart for those specific location.
  - Scatter plot:
    - To find the relation on Pay load and Class (success rate).
    - If we select a location from the list of locations. We get a chart for those specific location.
    - We also create a slider which can be used to change the Pay load value and view the impact.
- Instead of creating multiple charts we could create a single chart for multiple anlaysis.
- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W3_2_Dash%20exercise_worked.ipynb

Predictive Analysis (Classification)

- Standardize the data
- Split into training data and test data
- Find best Hyper parameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data.
- https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W4_1_IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_Worked.ipynb



Stage - 2

Select the list of features and the target.

Transform the date and split it into train and test sets.

Create GridSearchCV object using different Models.

Fit using training data and find out best parameters.

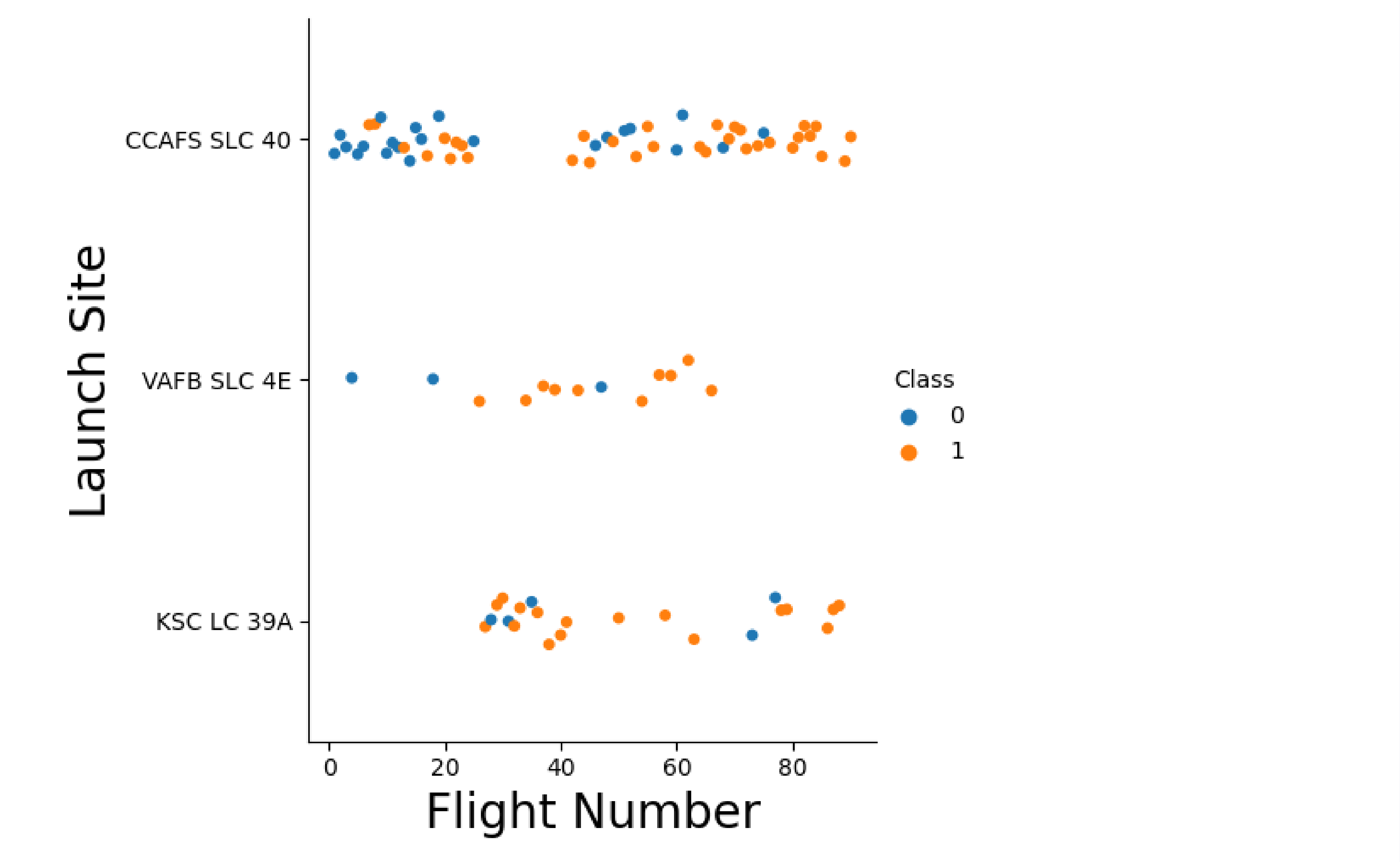
Results

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

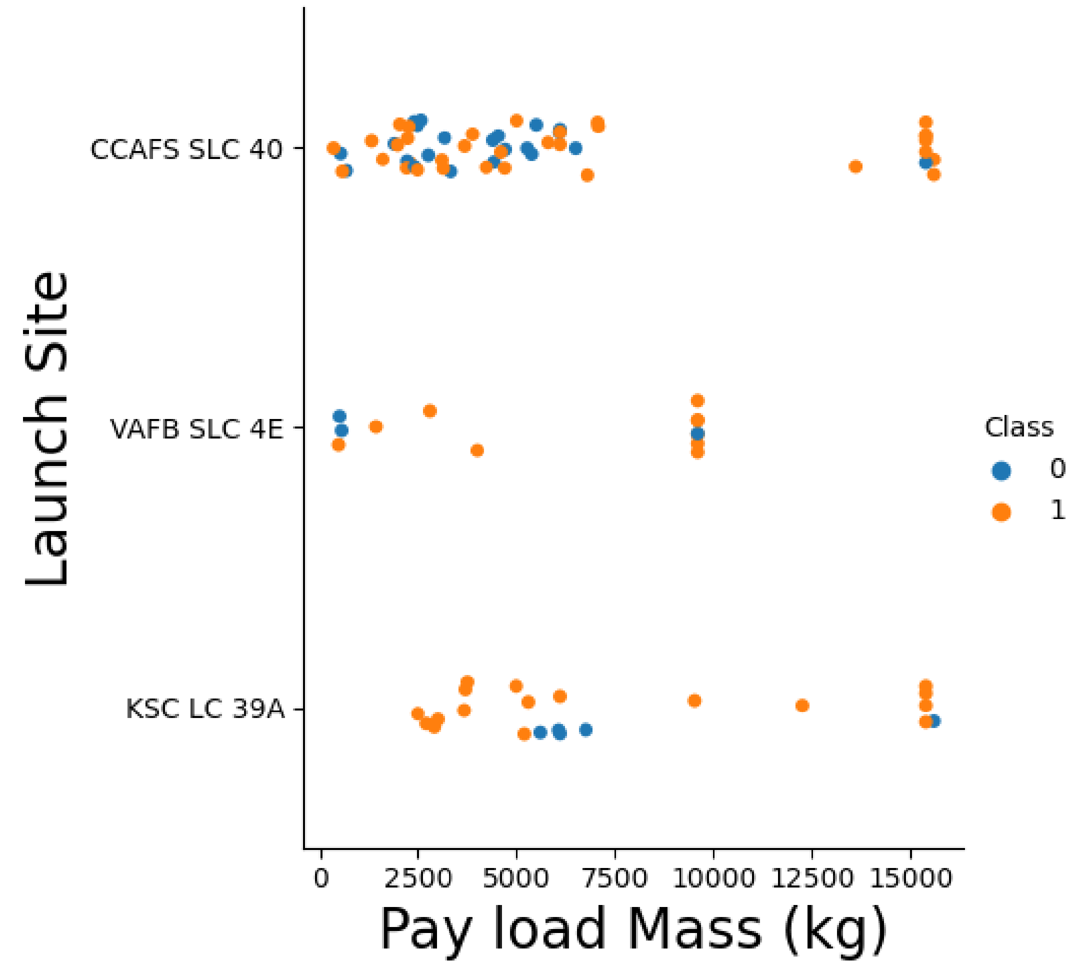
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

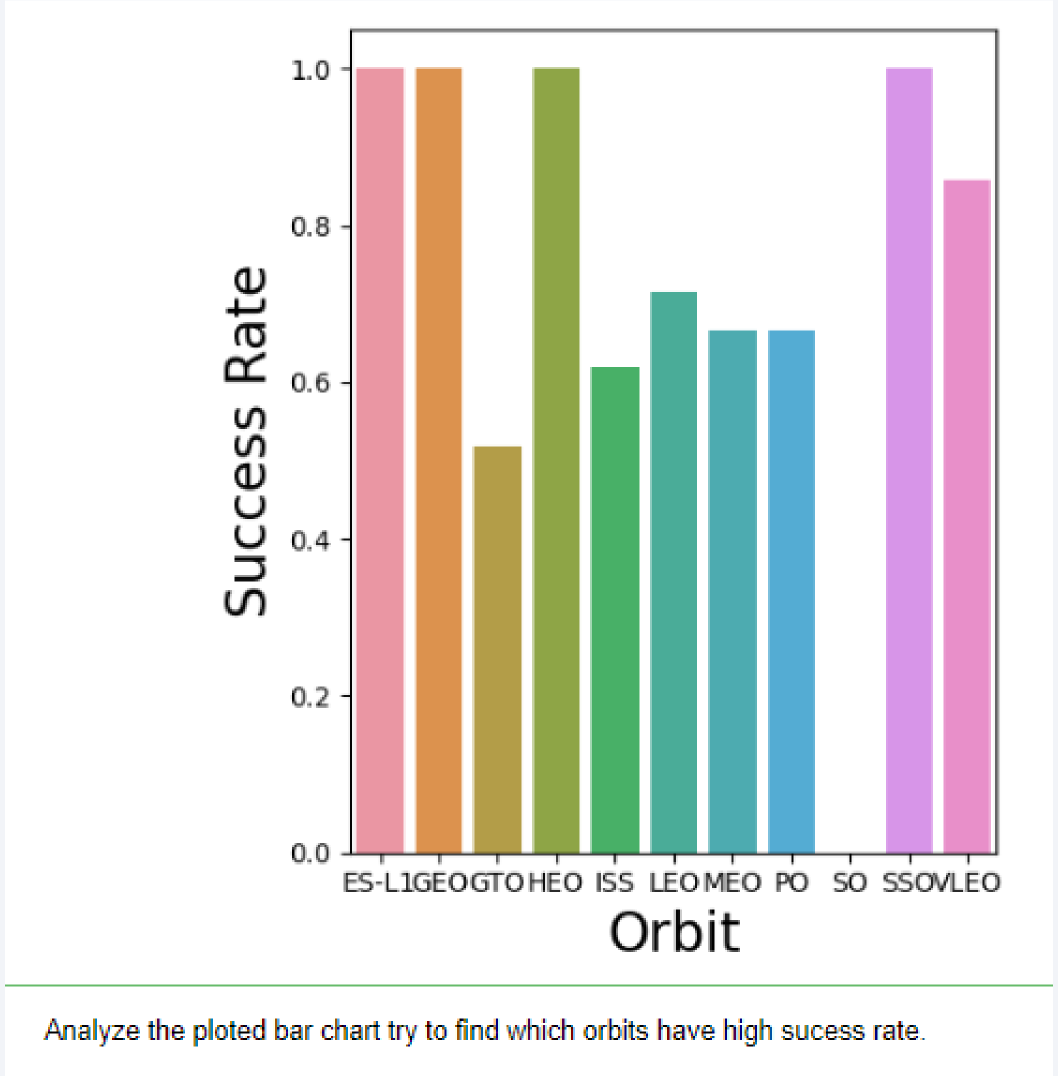


Payload vs. Launch Site



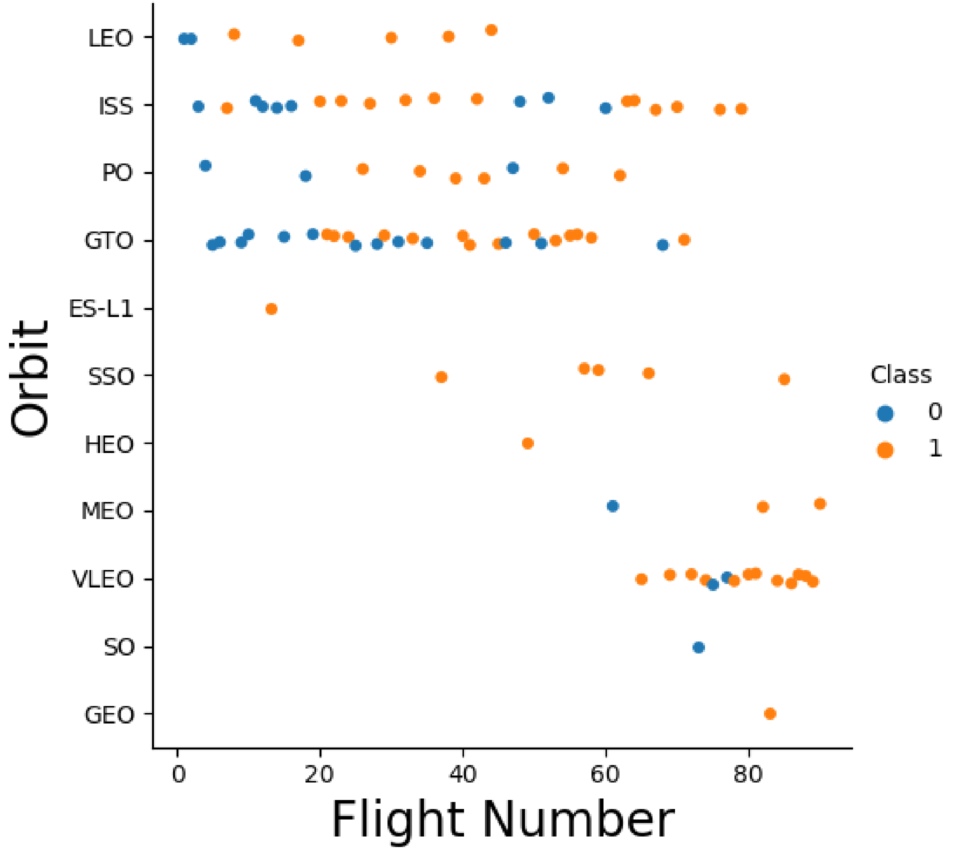
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



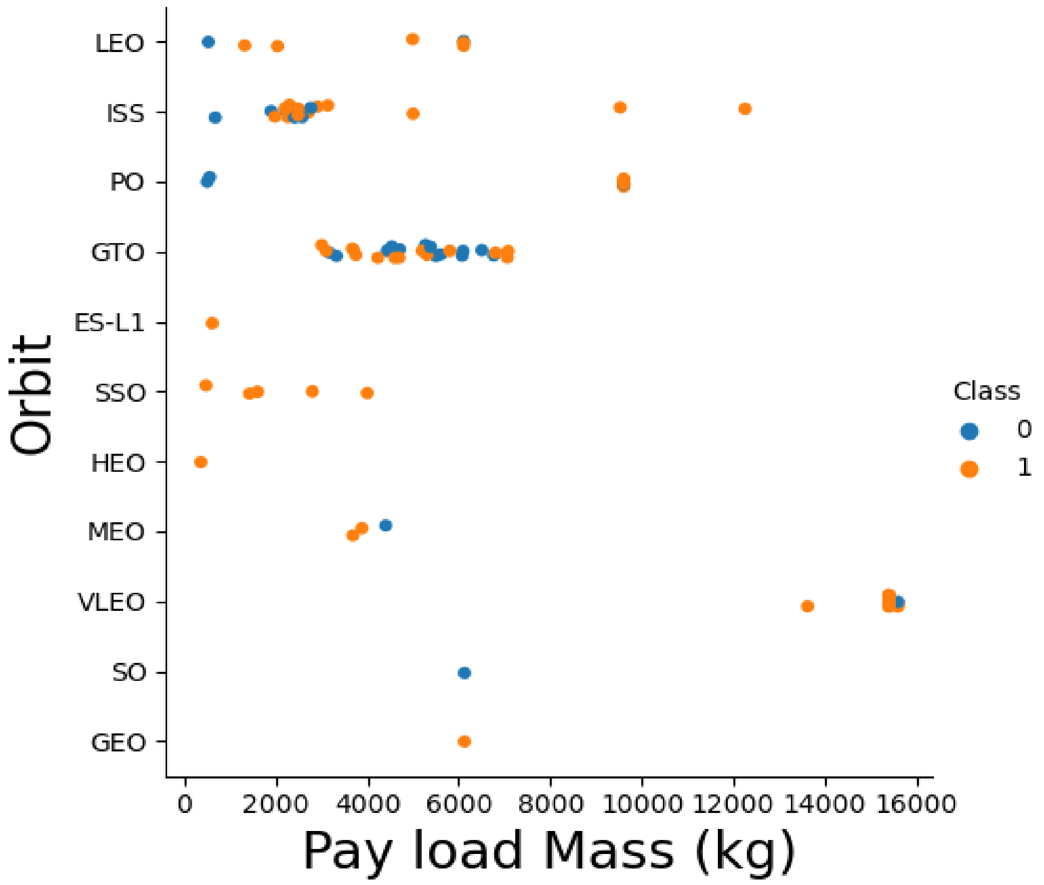
Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight Number vs. Orbit Type



You should 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.

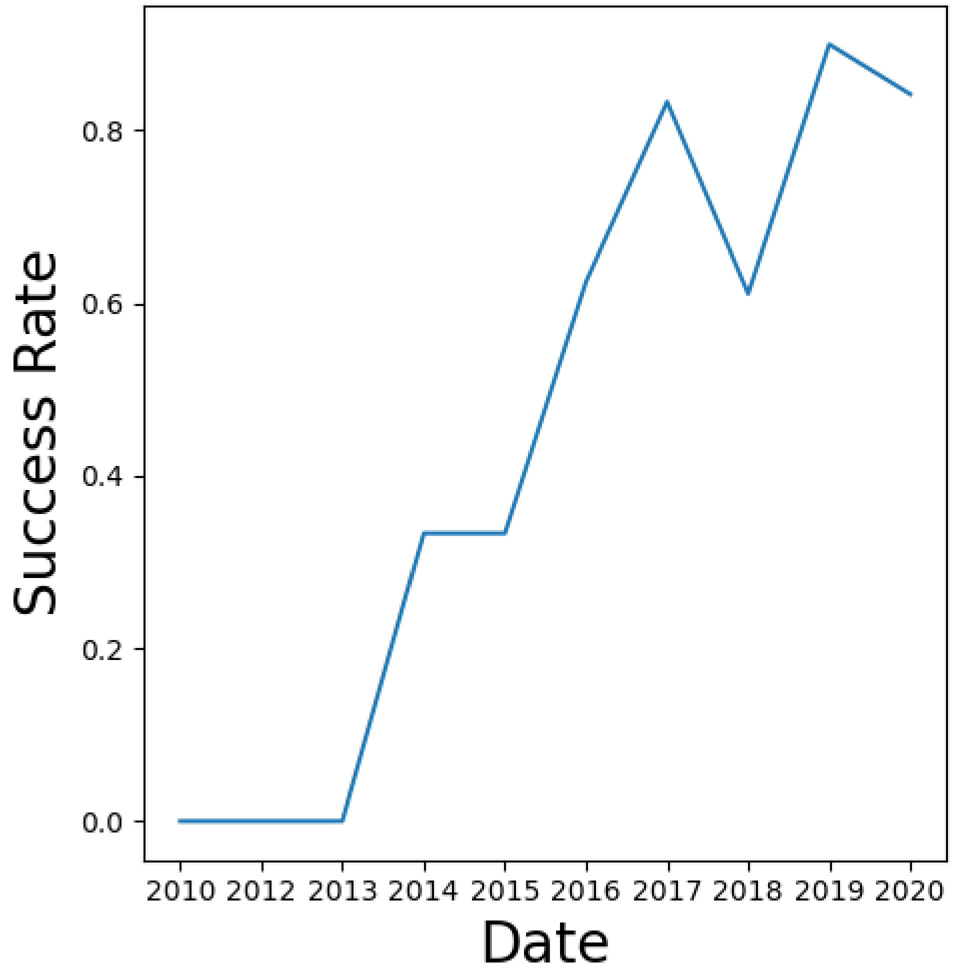
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

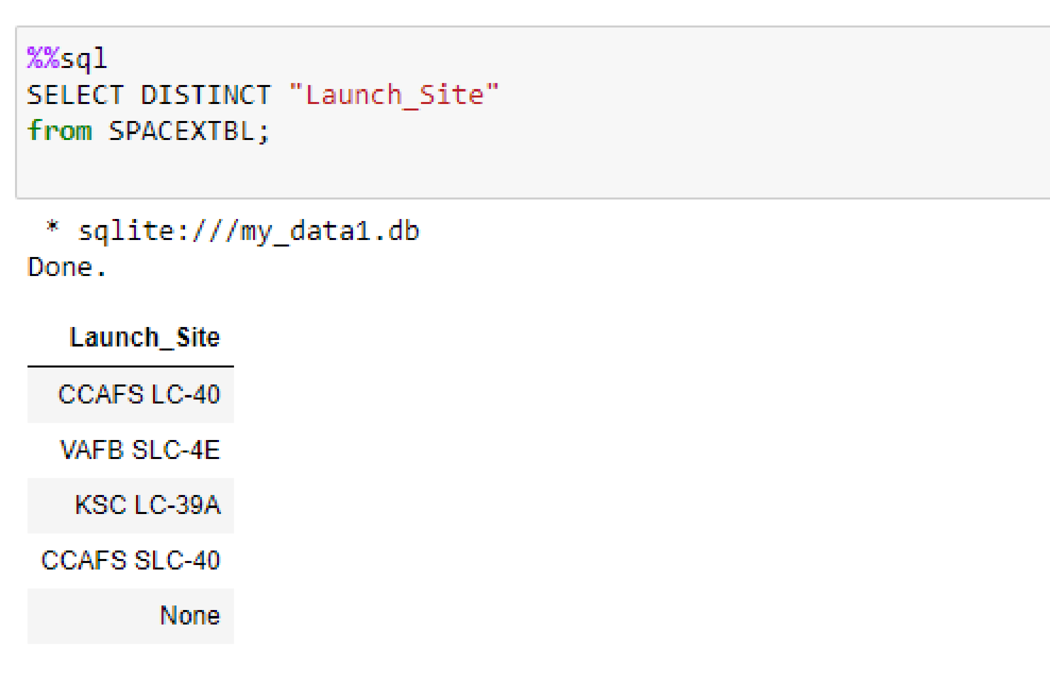
Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites
- Distinct keyword is used in this query.



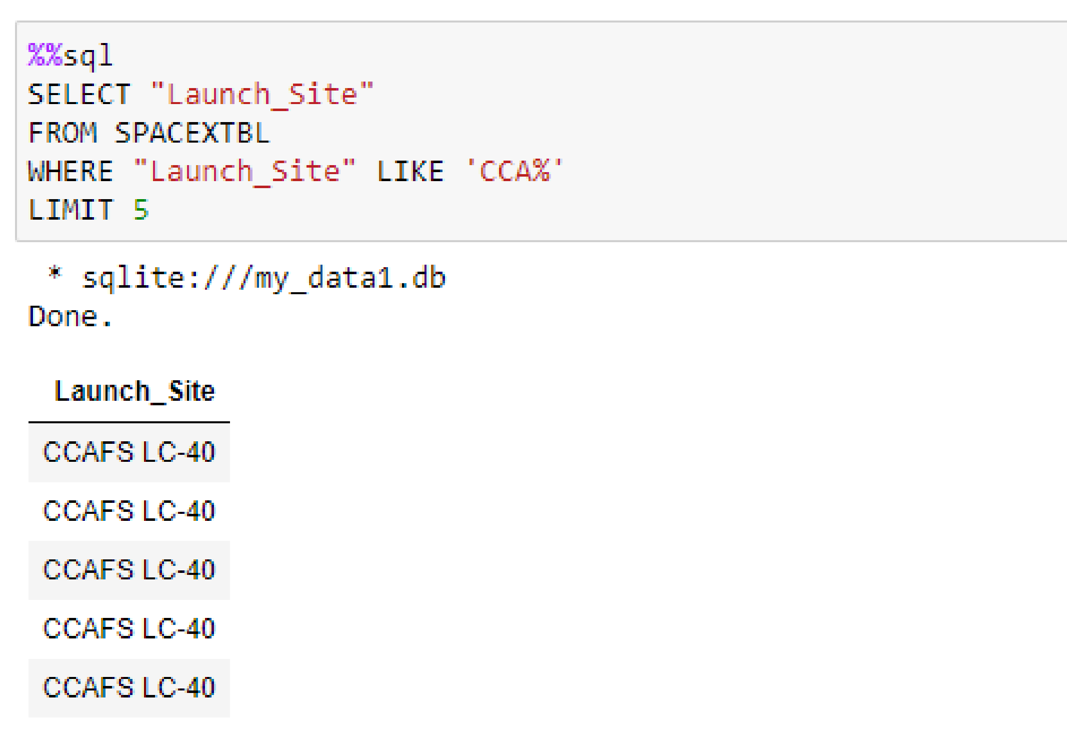
```
%%sql
SELECT DISTINCT "Launch_Site"
from SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- WHERE keyword is used to find launch sites starting with ‘CCA’



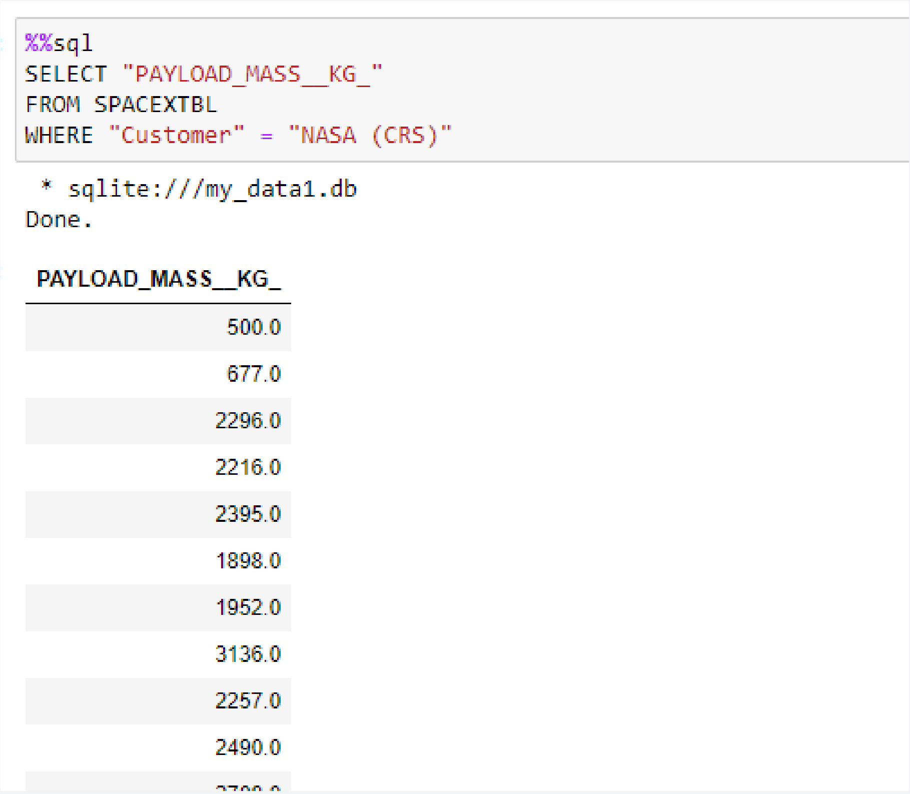
```
%%sql
SELECT "Launch_Site"
FROM SPACEXTBL
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



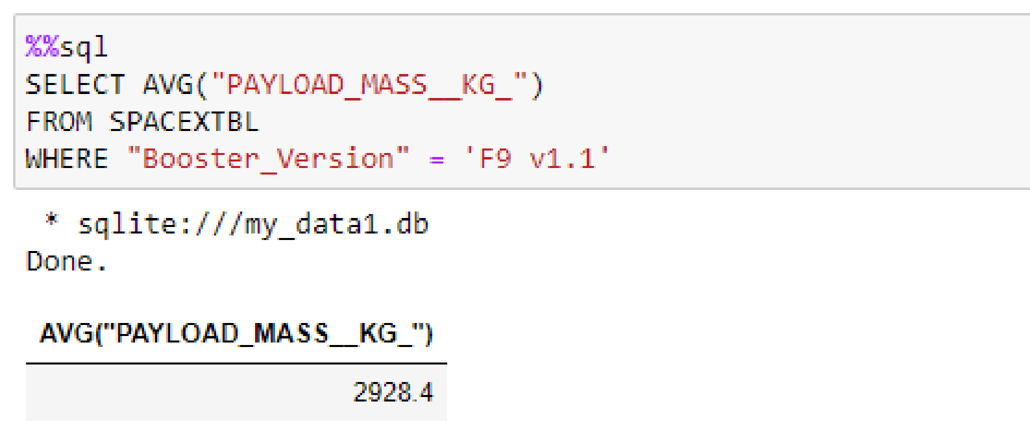
```
%%sql
SELECT "PAYLOAD_MASS__KG_"
FROM SPACEXTBL
WHERE "Customer" = "NASA (CRS)"
```

```
 * sqlite:///my_data1.db
Done.
```

PAYLOAD_MASS__KG_
500.0
677.0
2296.0
2216.0
2395.0
1898.0
1952.0
3136.0
2257.0
2490.0
2700.0

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1



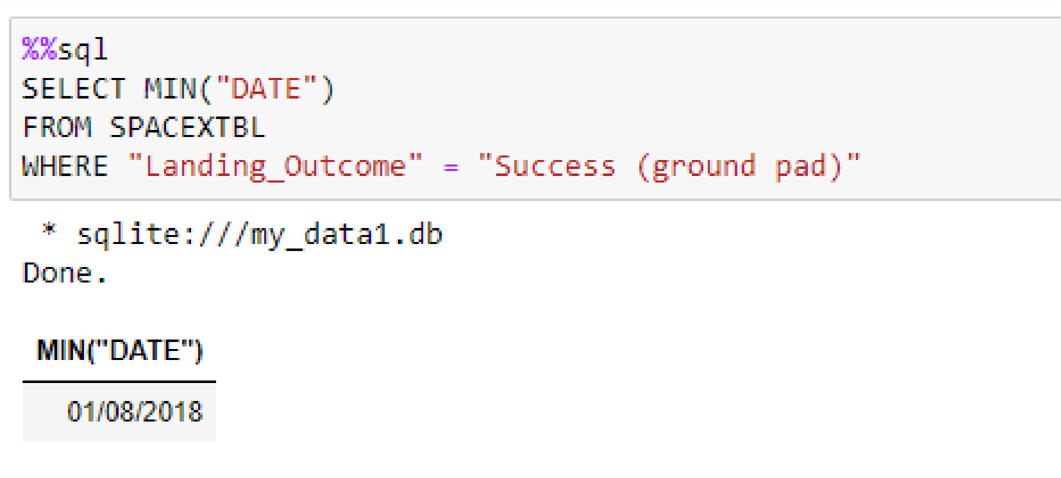
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_")
FROM SPACEXTBL
WHERE "Booster_Version" = 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



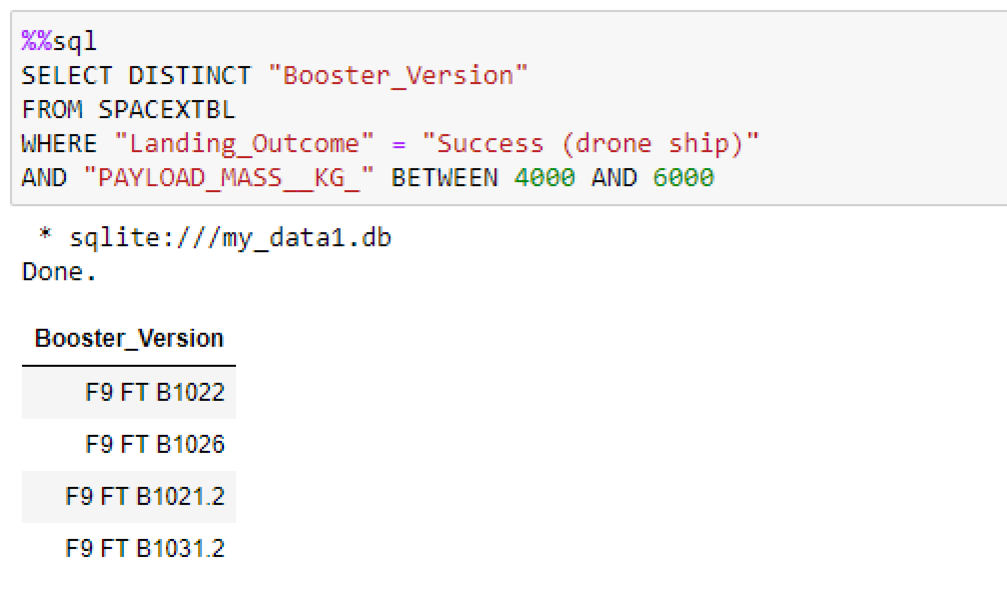
```
%%sql
SELECT MIN("DATE")
FROM SPACEXTBL
WHERE "Landing_Outcome" = "Success (ground pad)"
```

```
 * sqlite:///my_data1.db
Done.
```

MIN("DATE")
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



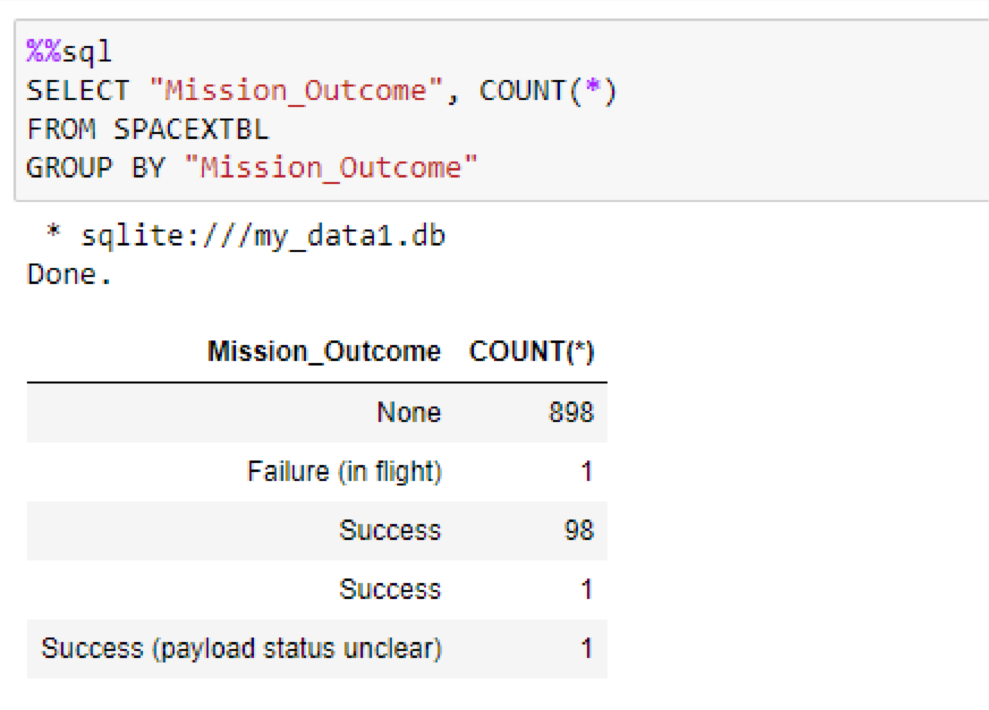
```
%%sql
SELECT DISTINCT "Booster_Version"
FROM SPACEXTBL
WHERE "Landing_Outcome" = "Success (drone ship)"
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



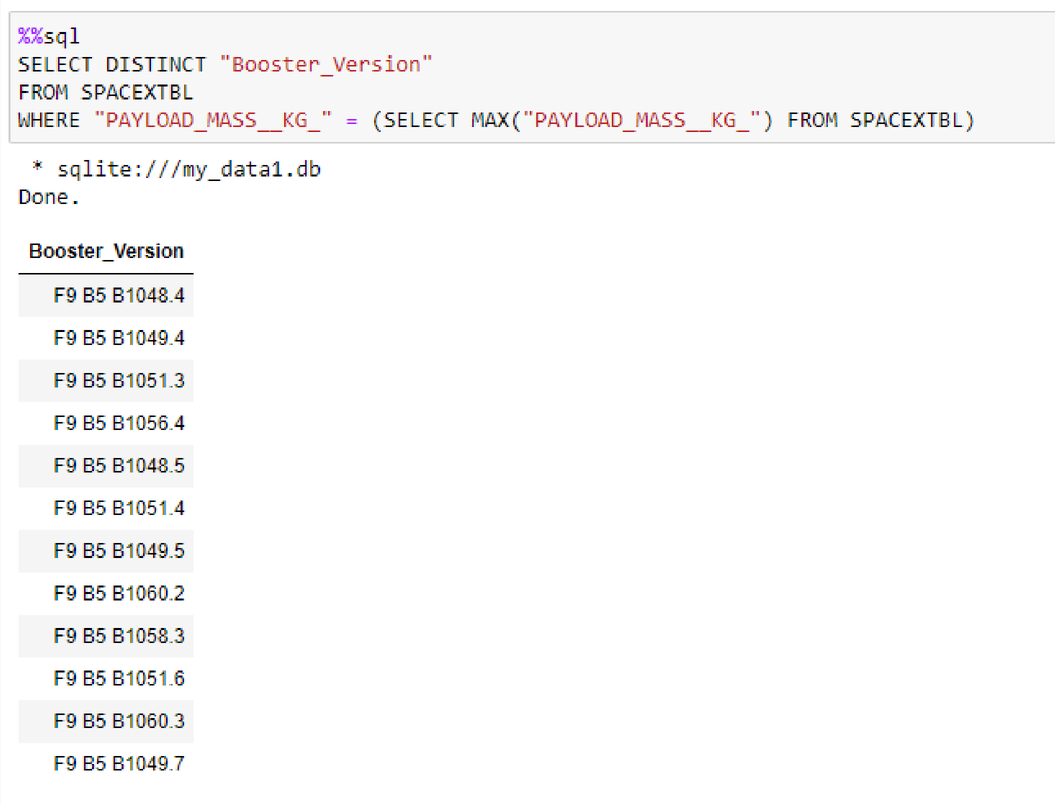
```
%%sql
SELECT "Mission_Outcome", COUNT(*)
FROM SPACEXTBL
GROUP BY "Mission_Outcome"
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	COUNT(*)
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



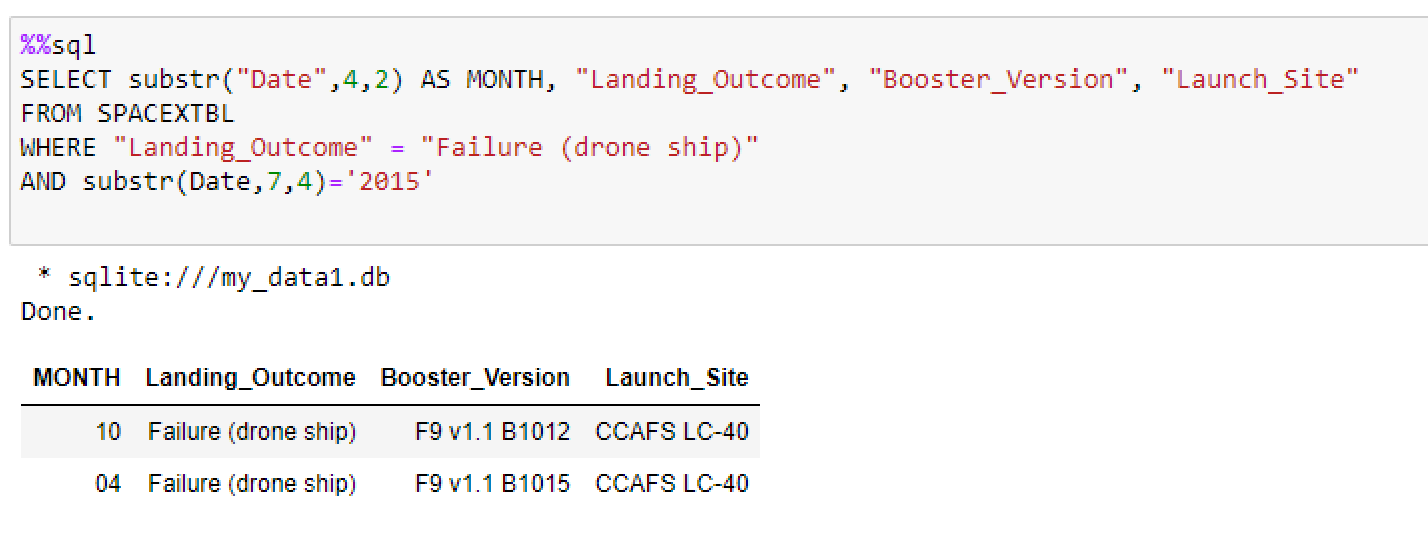
```
%%sql
SELECT DISTINCT "Booster_Version"
FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

```
 * sqlite:///my_data1.db
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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



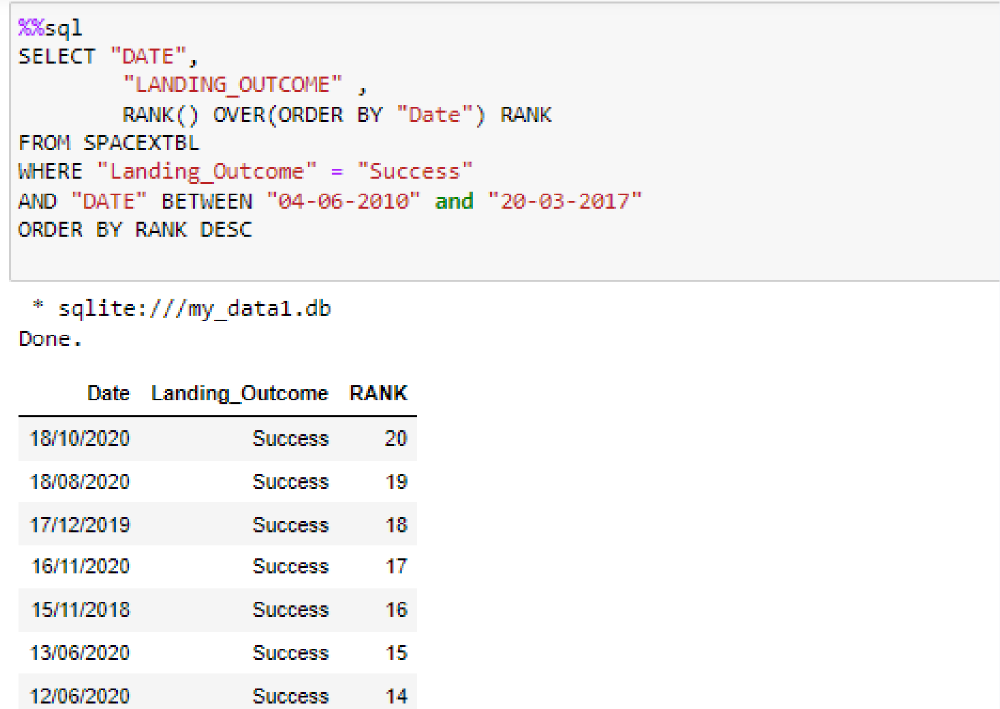
```
%%sql
SELECT substr("Date",4,2) AS MONTH, "Landing_Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTBL
WHERE "Landing_Outcome" = "Failure (drone ship)"
AND substr(Date,7,4)='2015'
```

```
 * sqlite:///my_data1.db
Done.
```

MONTH	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- 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



```
%%sql
SELECT "DATE",
        "LANDING_OUTCOME" ,
        RANK() OVER(ORDER BY "Date") RANK
FROM SPACEXTBL
WHERE "Landing_Outcome" = "Success"
AND "DATE" BETWEEN "04-06-2010" and "20-03-2017"
ORDER BY RANK DESC
```

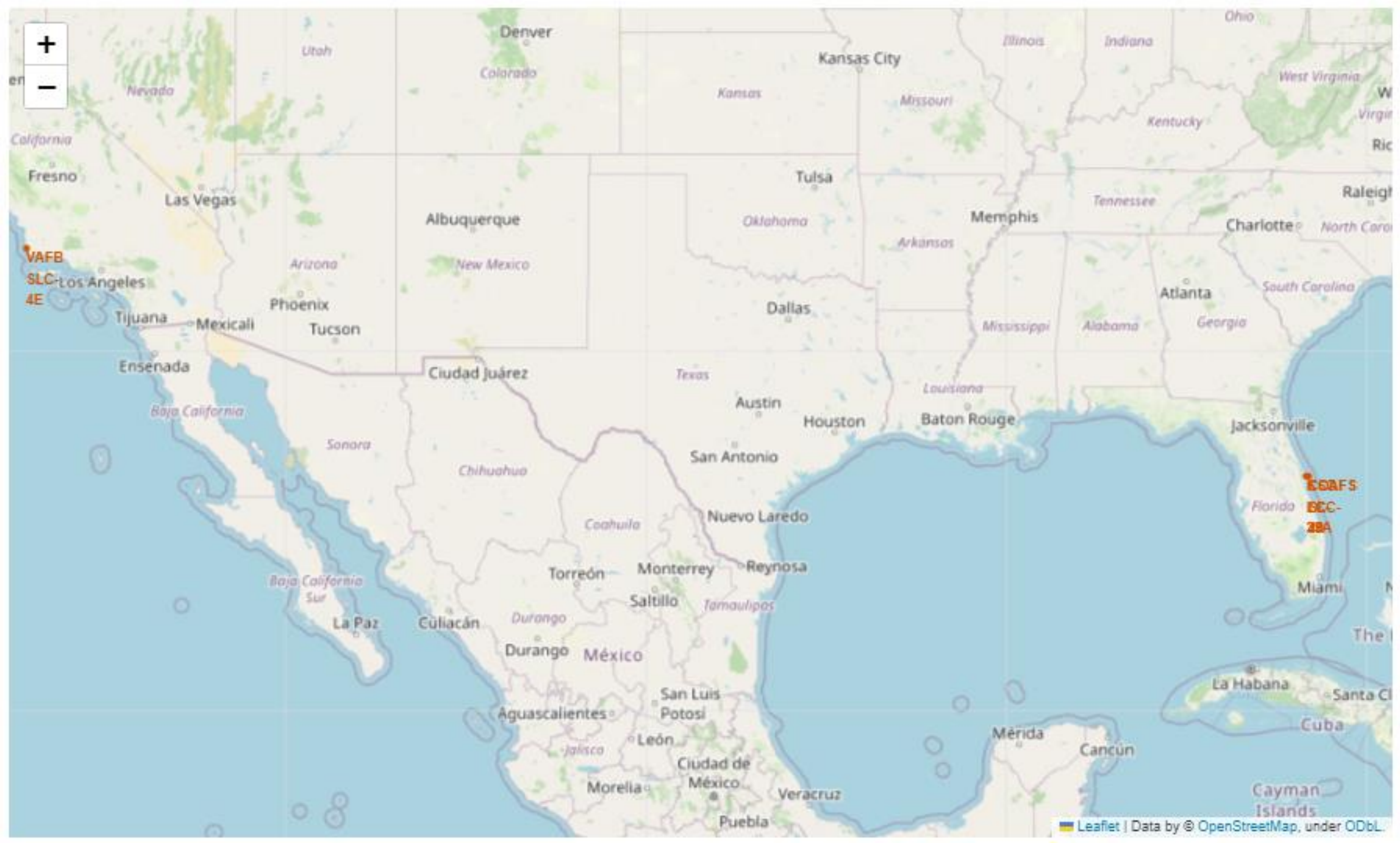
```
 * sqlite:///my_data1.db
Done.
```

Date	Landing_Outcome	RANK
18/10/2020	Success	20
18/08/2020	Success	19
17/12/2019	Success	18
16/11/2020	Success	17
15/11/2018	Success	16
13/06/2020	Success	15
12/06/2020	Success	14

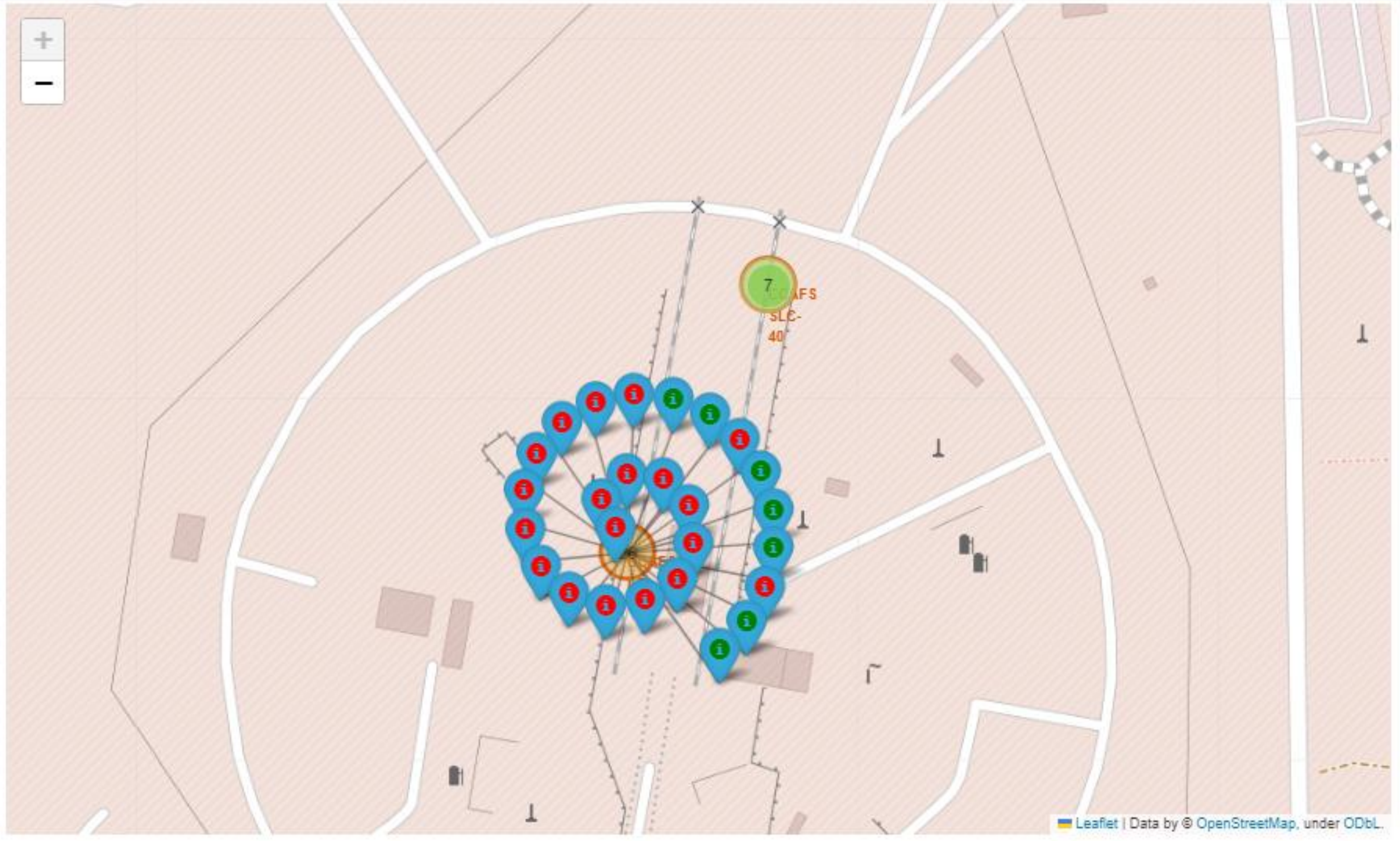
Section 3

Launch Sites Proximities Analysis

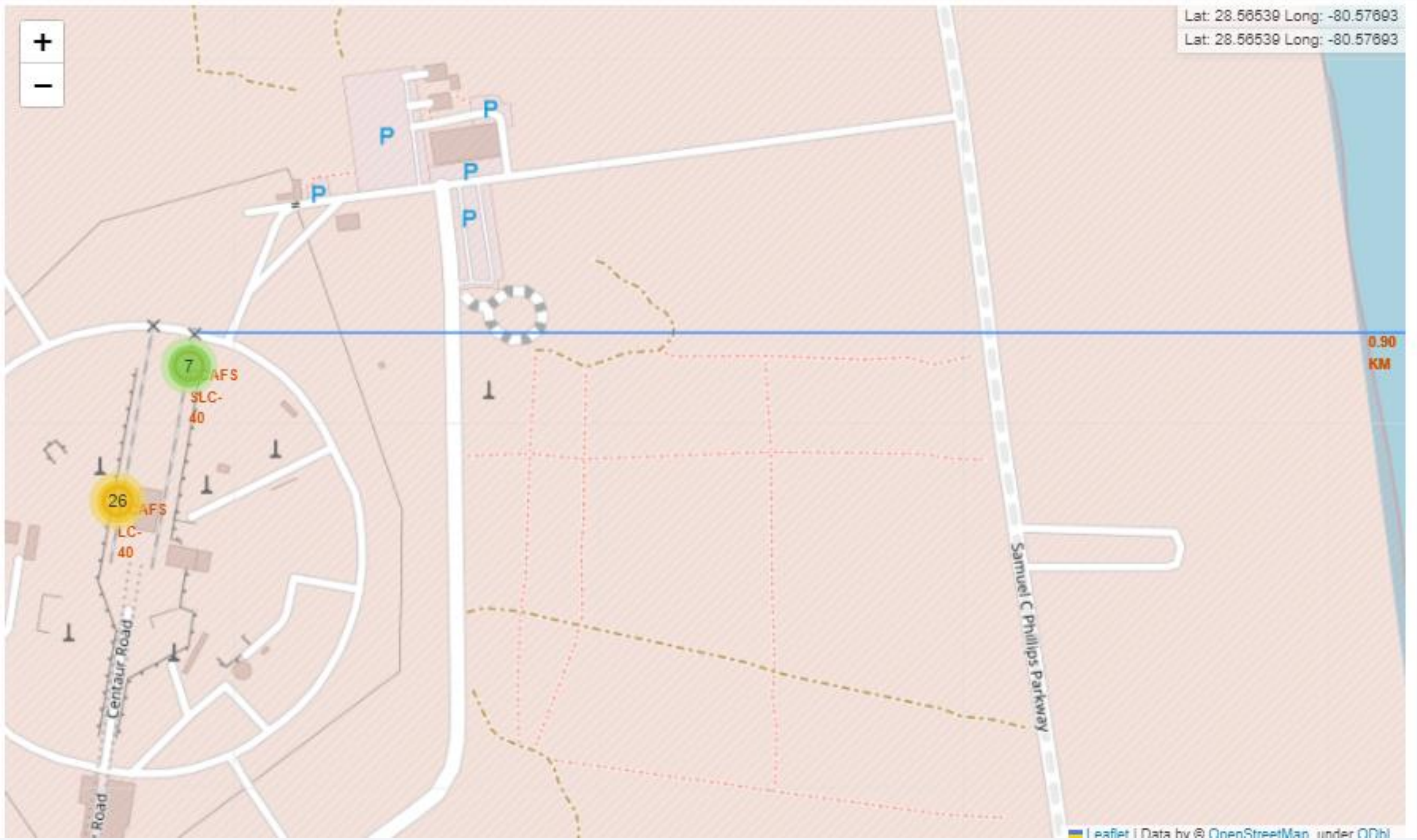
Launch Sites Locations Analysis with Folium

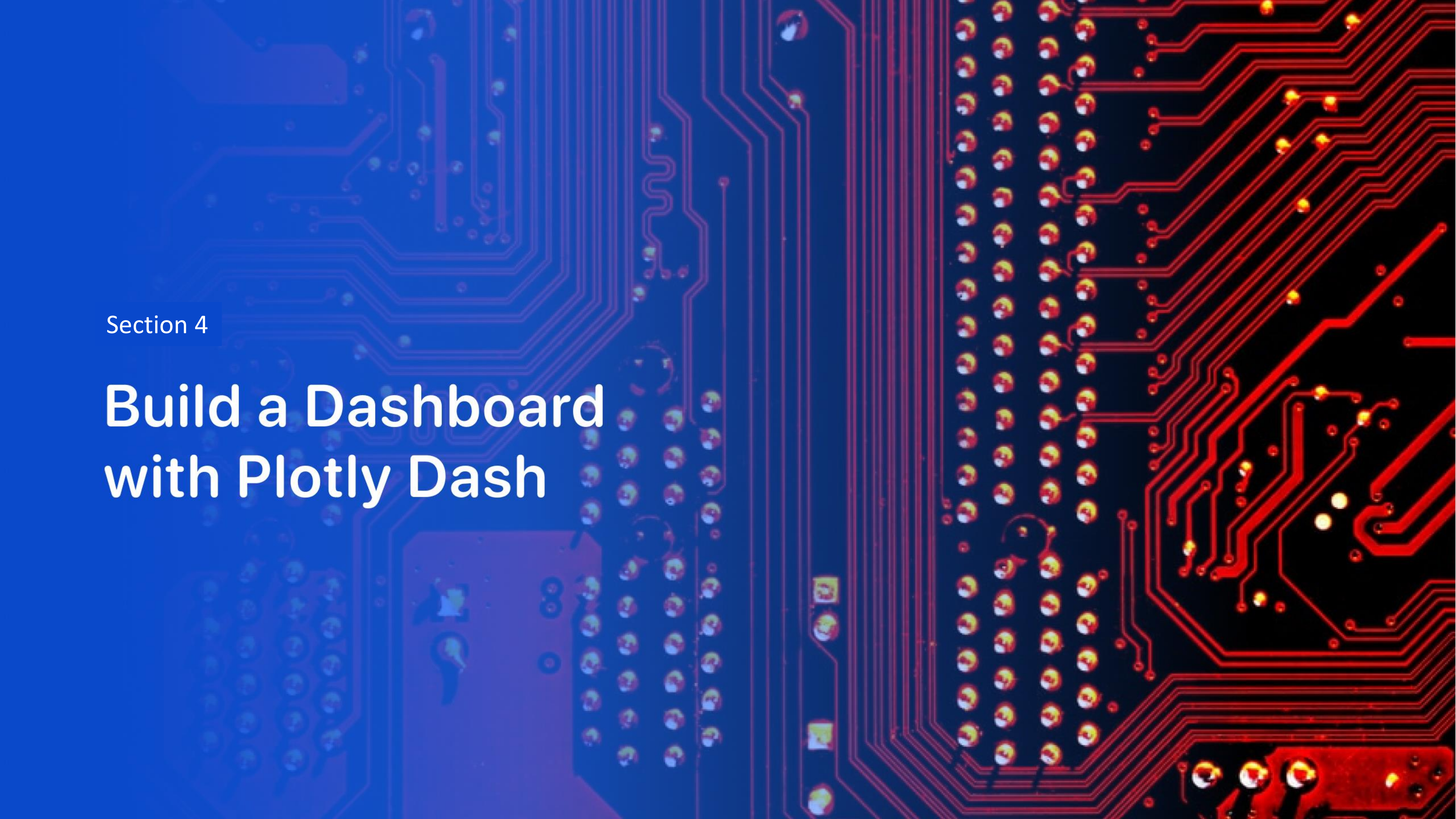


Marker_cluster in Folium Map



Folium Map - distances between a launch site to its proximities

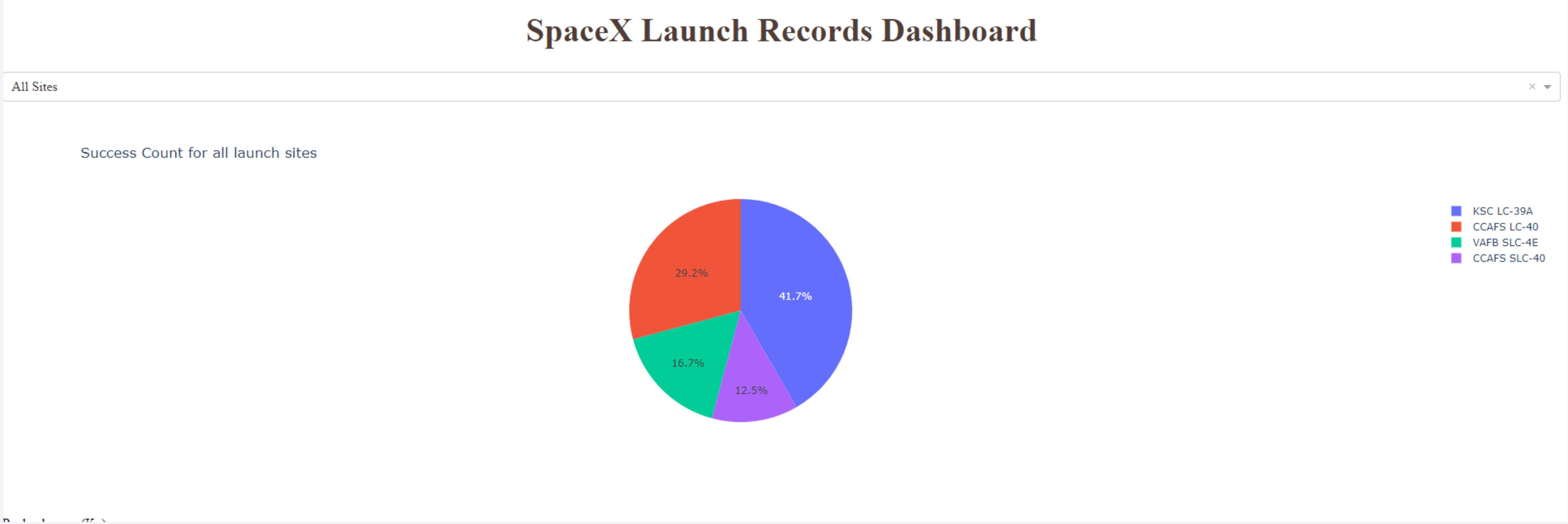

+
−
Lat: 28.56539 Long: -80.57693
Lat: 28.56539 Long: -80.57693
P
P
P
P
P
0.90 KM
7
CCAFS SLC-40
26
CCAFS LC-40
Centaur Road
Road
Samuel C Phillips Parkway
Leaflet | Data by © OpenStreetMap, under ODbL



Section 4

Build a Dashboard with Plotly Dash

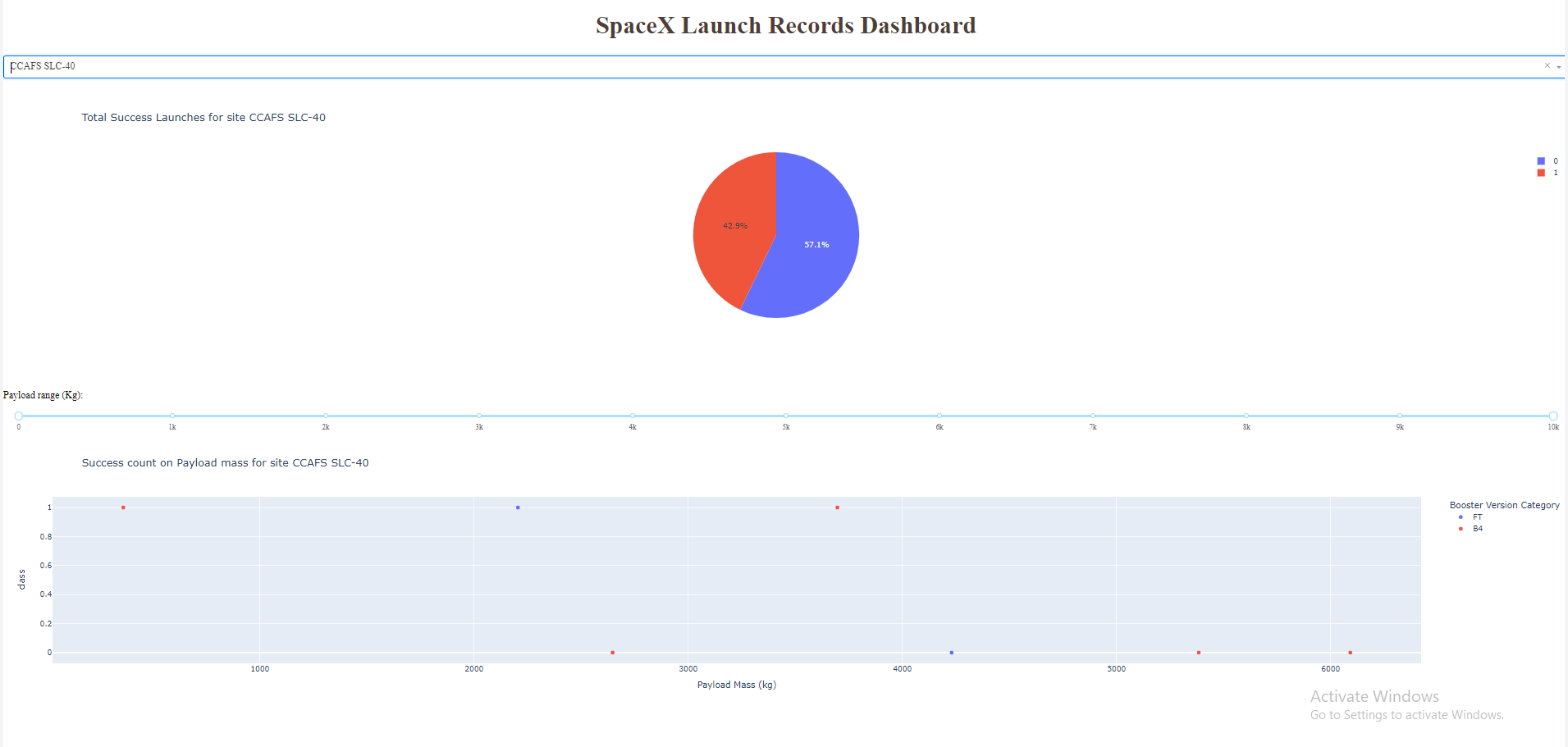
Success count for all launch sites:



SpaceX Launch Records Dashboard

All Sites

SpaceX Launch Records Dashboard



SpaceX Launch Records Dashboard

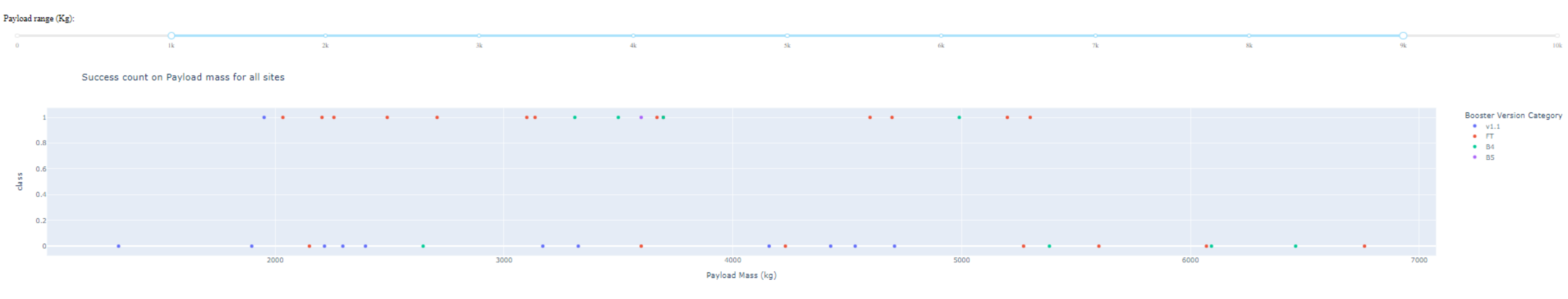
CCAFS SLC-40

Total Success Launches for site CCAFS SLC-40

Payload range (Kg):

Success count on Payload mass for site CCAFS SLC-40

Success Count of Payload mass for all sites:

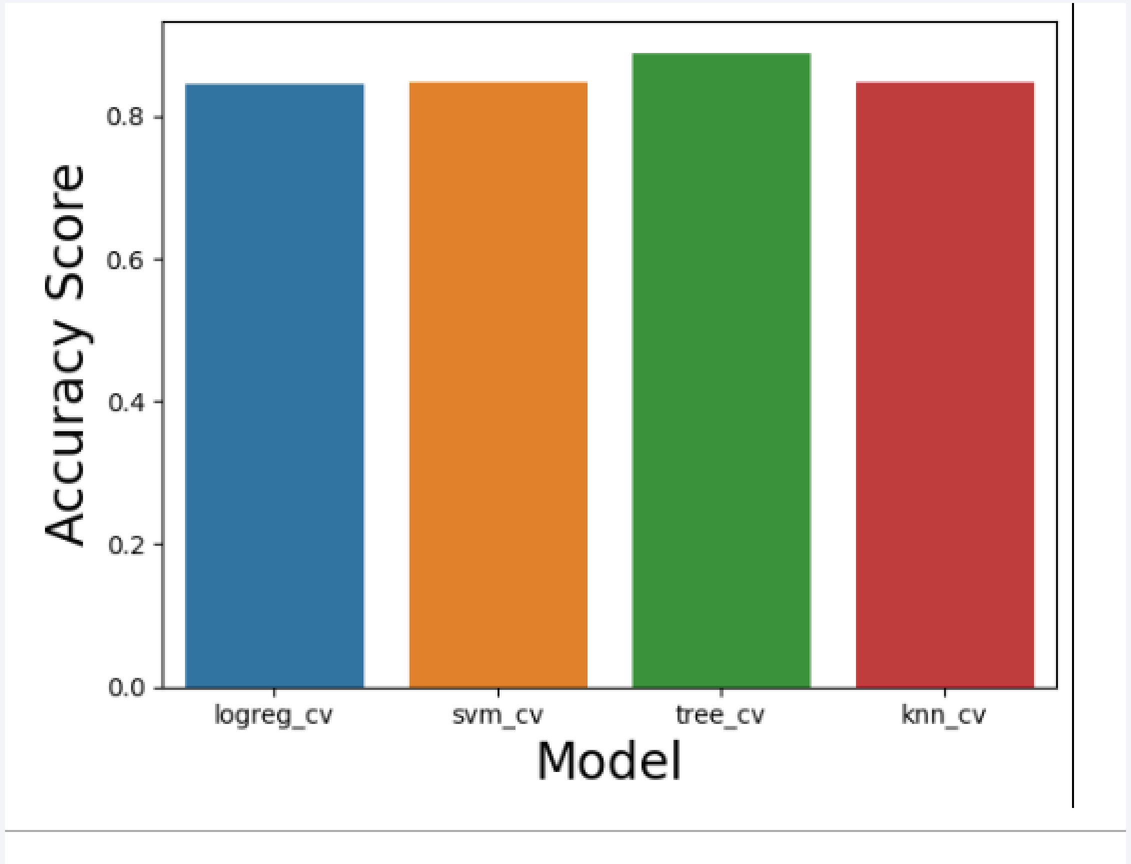


Payload range (Kg):

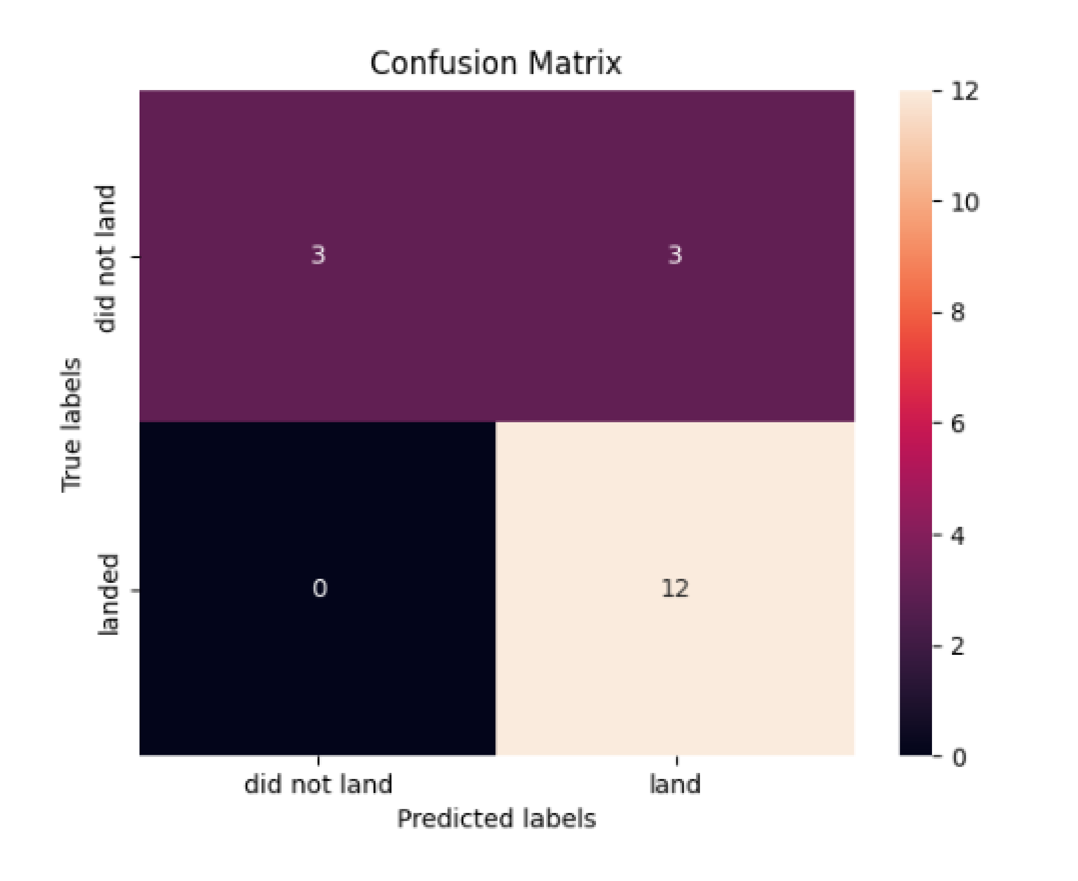
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- Collected data from different source:
  - API’s, Web Scrapping, Web hosted .. etc.
- Used different methods to clean the data.
- Summarize the Class into 2 types.
- Used different charts for analysis and find relations
- Created multiple ML models.
- Used GridSearchCV to identify best parameters.

Appendix

- Below are the links of files which I have created during my analysis.
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W1_1_jupyter-labs-spacex-data-collection-api%20_worked.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W1_2_jupyter-labs-webscraping_worked.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W1_3_IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite_worked.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W2_1_jupyter-labs-eda-sql-coursera_sqllite_WORKED.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W2_2_IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite_worked.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W3_1_IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite_worked.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W3_2_Dash%20exercise_worked.ipynb
  - https://github.com/Rakesh-Arsid/Capstone_Project/blob/main/W4_1_IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_Worked.ipynb

Thank you!