IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kotireddy Kotireddy

24th March 2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

- EDA results
- Interactive analytics
- Predictive analysis

Introduction

Project background and context

SpaceX is Rocket launching service provider. At first stage of the rocket launching SpaceX re-useses its Technology.Because of this Technology rocket launching cost is 62 million dollars; other service providers cost is 165 million dollars each. In this project we will predict if the Falcon 9 first stage will land successfully.

We will use data analysis tools to load a dataset from SpaceX website, after that we clean it and find whether the SpaceX rocket launching is successfully in its first stage or not.

Problems you want to find answers

The project task is to predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - ➢ SpaceX Rest API
  - ➢ Web Scrapping from Wikipedia

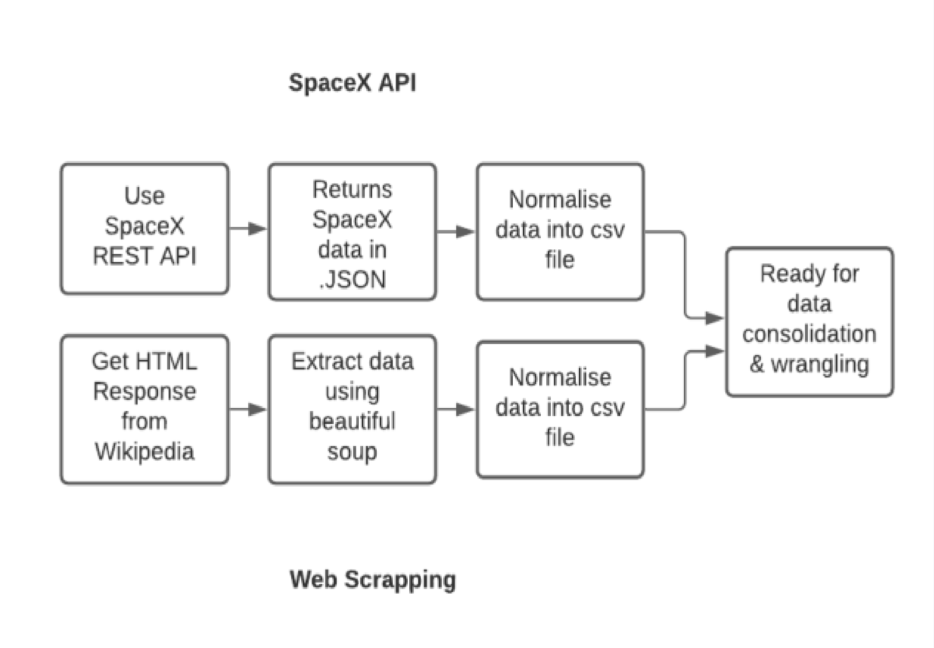
Perform data wrangling

- ➢ One Hot Encoding data fields for Machine Learning and data cleaning of null values and 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
  - ➢ LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered,
- launch specifications, landing specifications, and landing outcome.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.



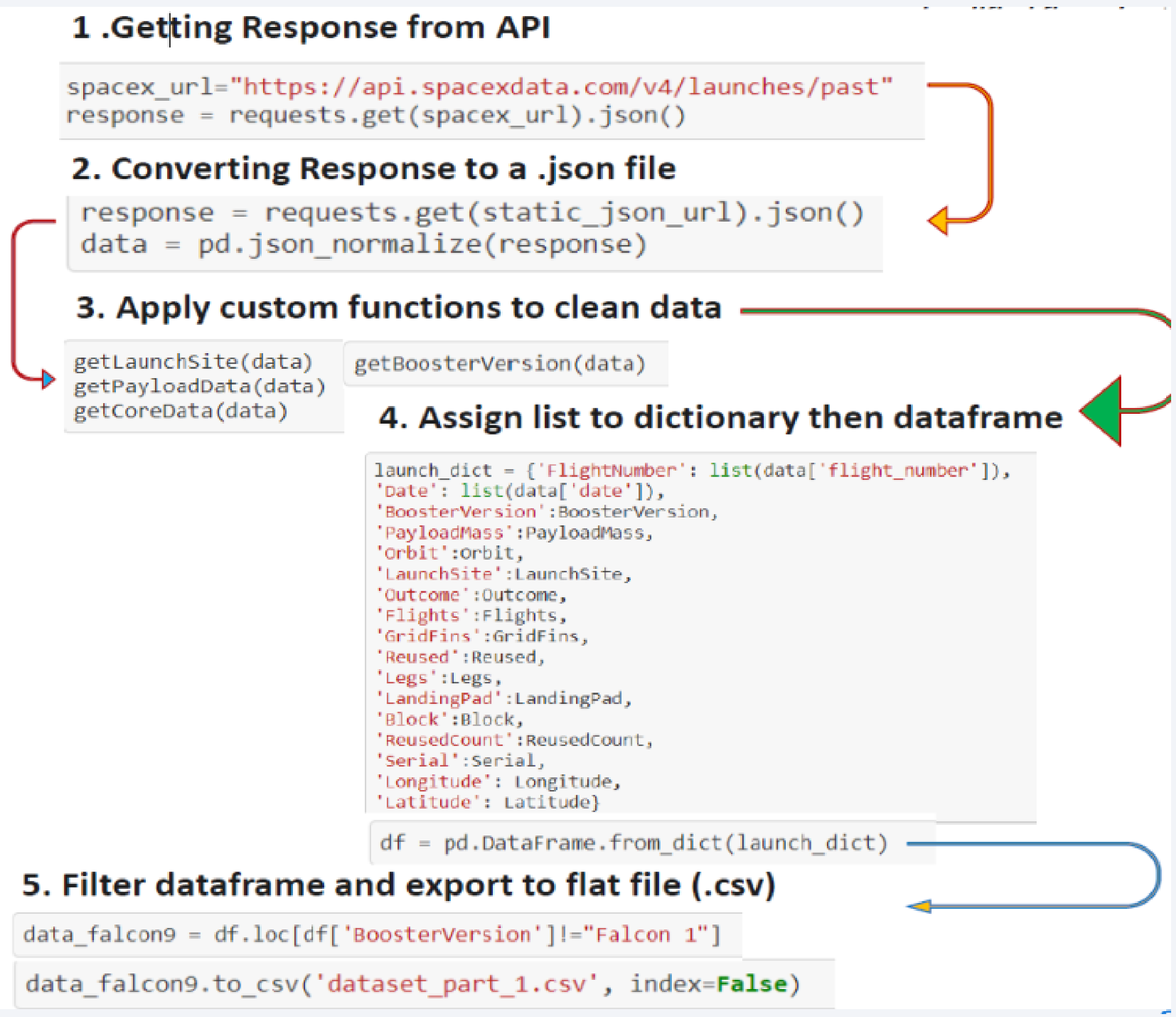
SpaceX API
Use SpaceX REST API
Returns SpaceX data in .JSON
Normalise data into csv file
Get HTML Response from Wikipedia
Extract data using beautiful soup
Normalise data into csv file
Ready for data consolidation & wrangling
Web Scrapping

Data Collection – SpaceX API

Data collection with SpaceX REST calls

URL to the API Notebook

https://github.com/kkoti30/DS0720EN/blob/d21144e3e9610d5dfc826119686983d36c4f8502/Data%20Collection%20API.ipynb



1 .Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2. Converting Response to a .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
getBoosterVersion(data)
```

4. Assign list to dictionary then dataframe

```
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}
```

```
df = pd.DataFrame.from_dict(launch_dict)
```

5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

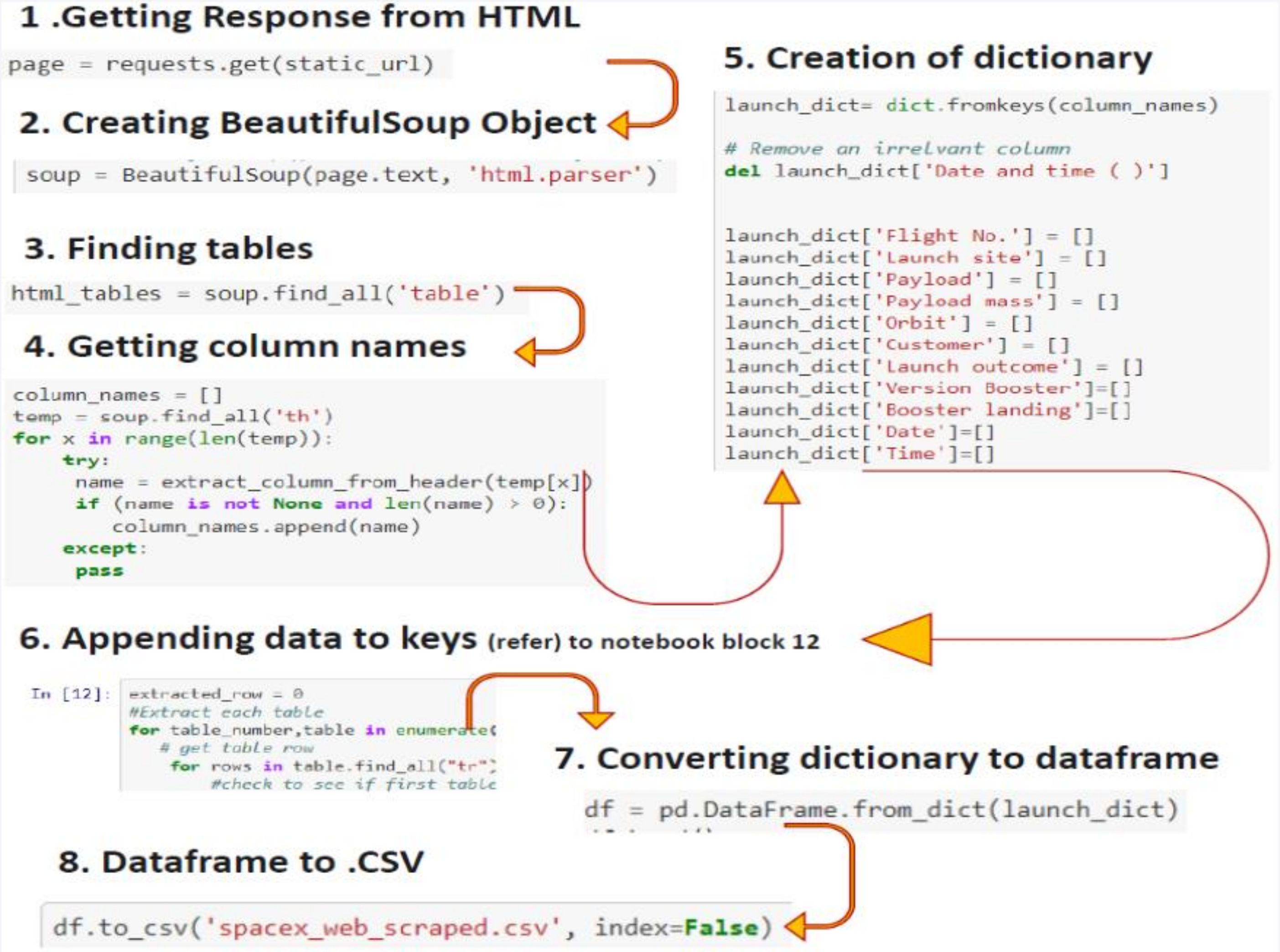
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

Web Scrapping from Wikipedia

GitHub URL:

https://github.com/kkoti30/DS0720EN/blob/d21144e3e9610d5dfc826119686983d36c4f8502/Web%20Scraping.ipynb



1 .Getting Response from HTML

```
page = requests.get(static_url)
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Finding tables

```
html_tables = soup.find_all('table')
```

4. Getting column names

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

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']


launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys (refer) to notebook block 12

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
            # get table row
             for rows in table.find_all("tr"
                 #check to see if first table
```

7. Converting dictionary to dataframe

```
df = pd.DataFrame.from_dict(launch_dict)
```

8. Dataframe to .CSV

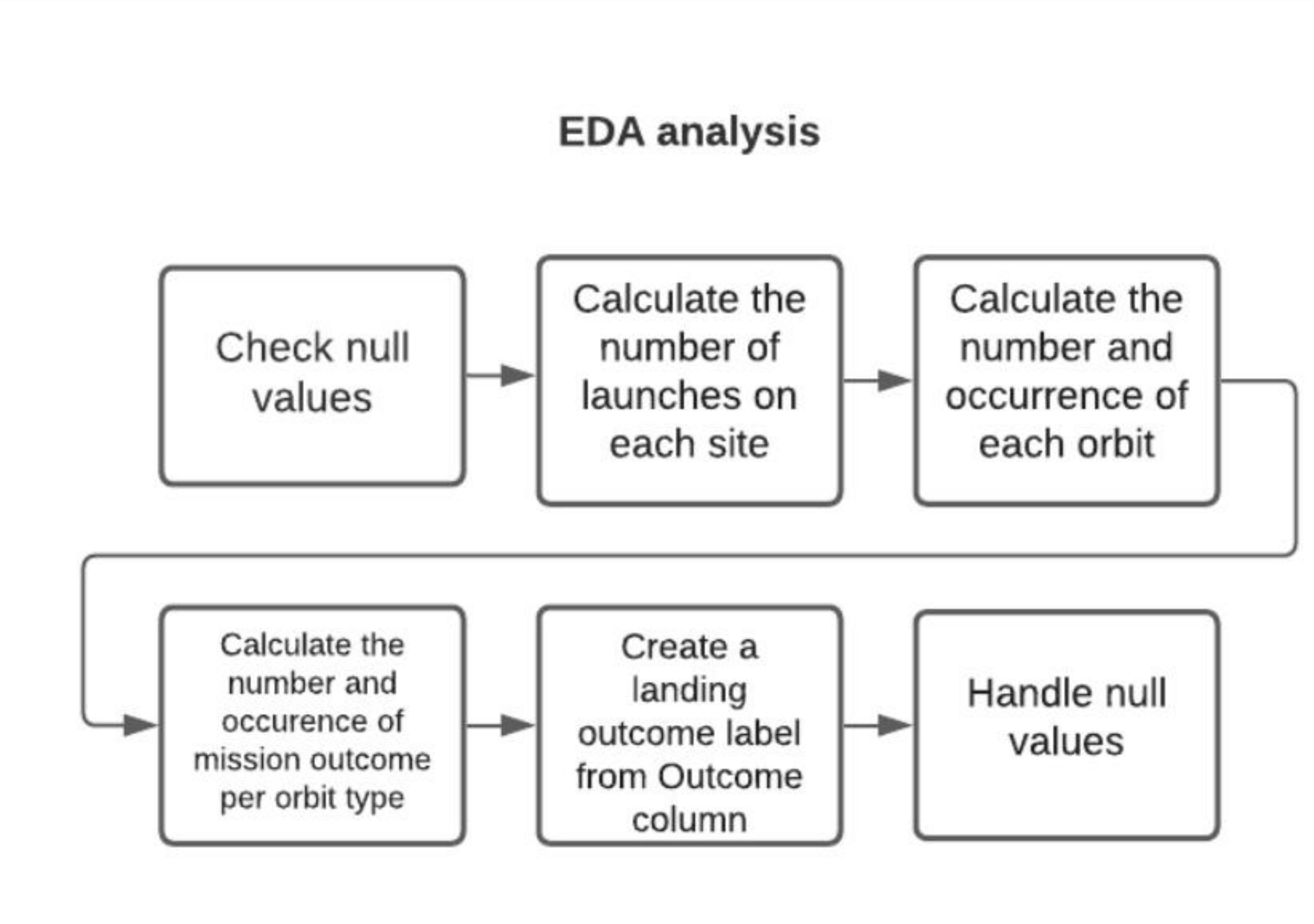
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Flow Chart of Data Wrangling

GitHub URL

https://github.com/kkoti30/DS0720EN/blob/d21144e3e9610d5dfc826119686983d36c4f8502/Web%20Scraping.ipynb



EDA analysis

Check null values

Calculate the number of launches on each site

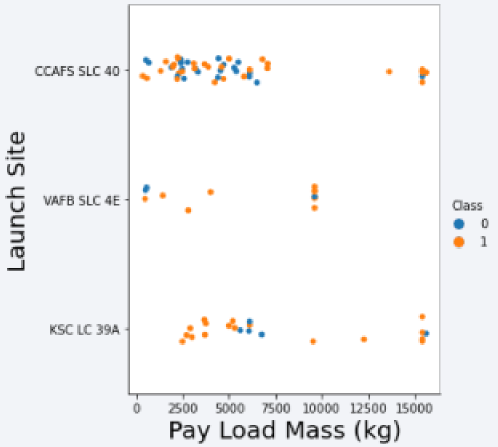
Calculate the number and occurrence of each orbit

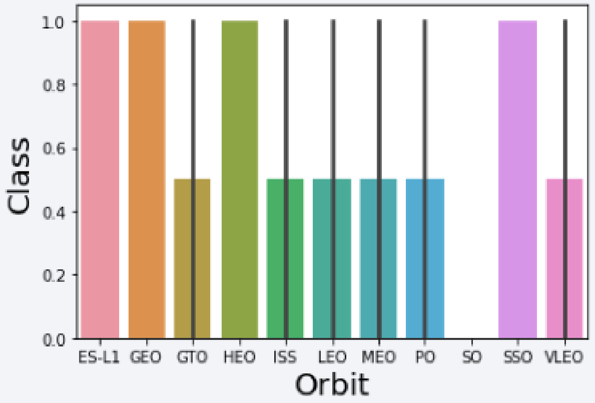
Calculate the number and occurence of mission outcome per orbit type

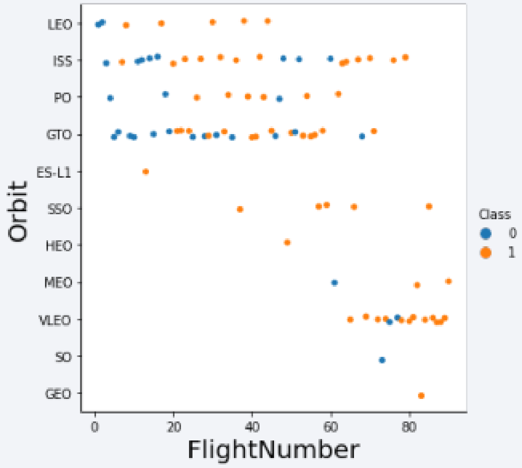
Create a landing outcome label from Outcome column

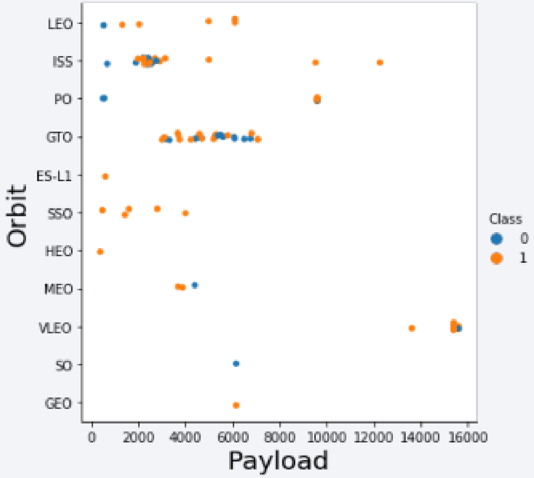
Handle null values

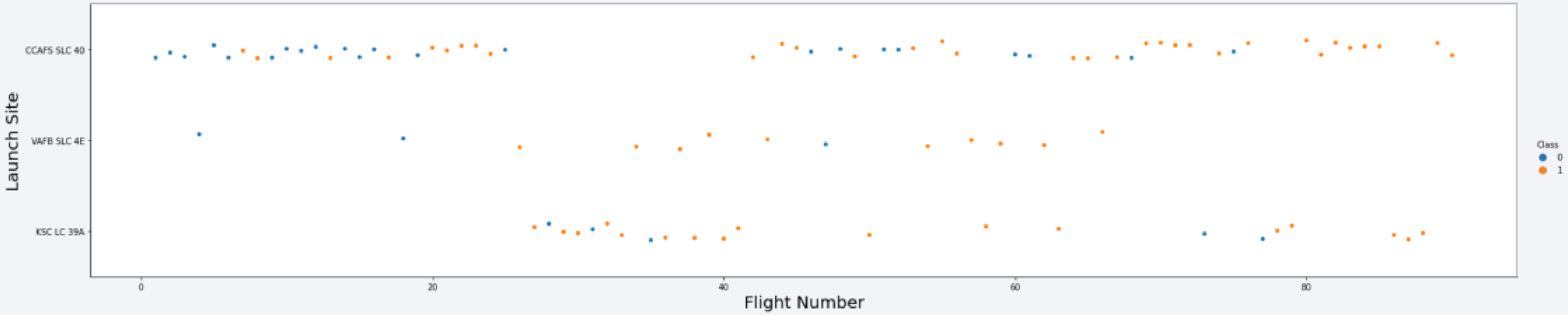
EDA with Data Visualization

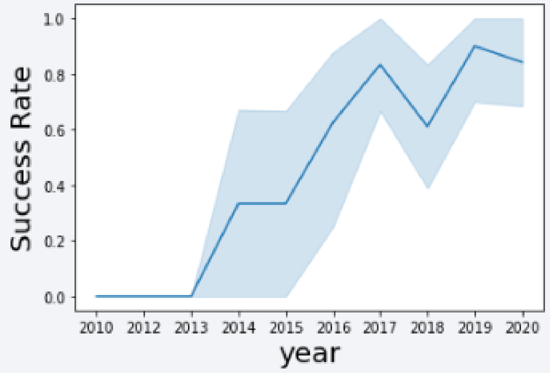












https://github.com/kkoti30/DS0720EN/blob/d21144e3e9610d5dfc826119686983d36c4f8502/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

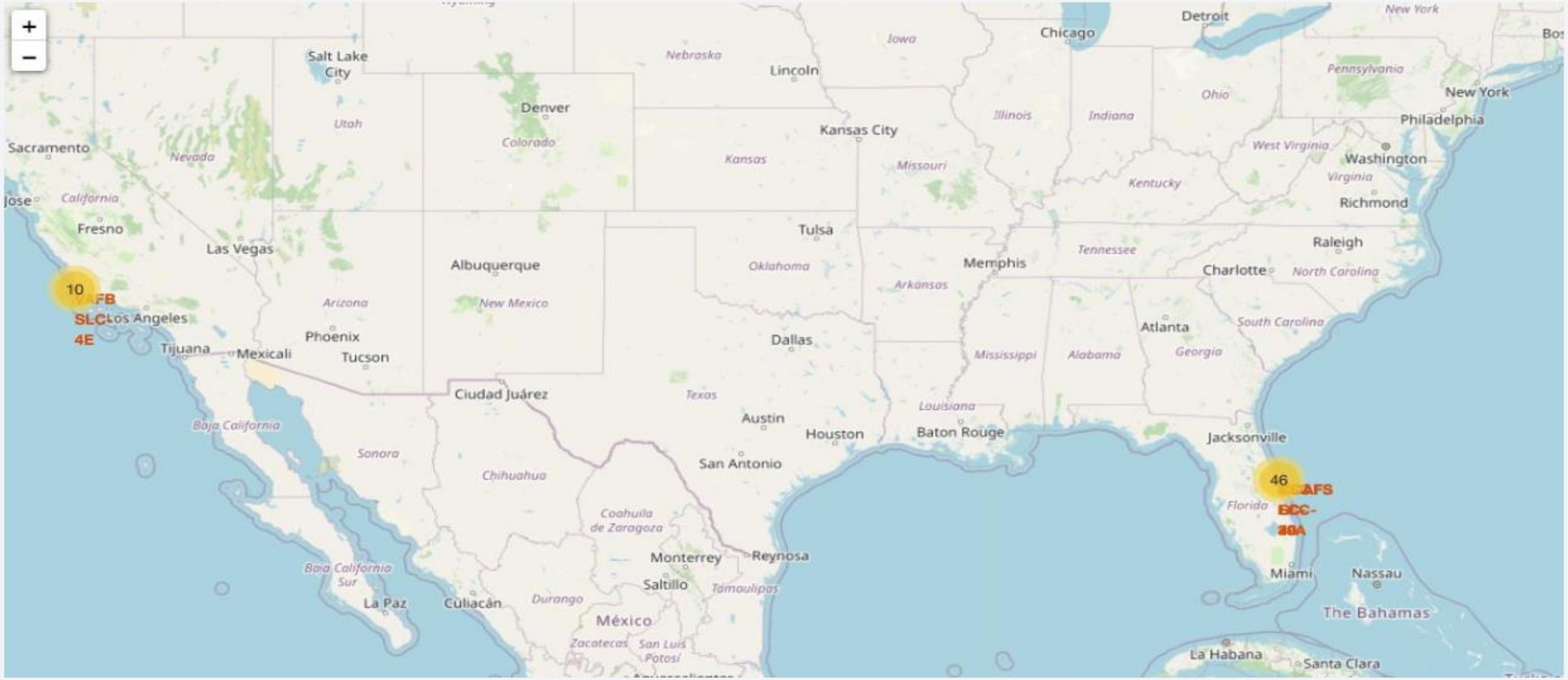
SQL queries performed include:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass.
- Listing 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.

Github URL:

https://github.com/kkoti30/DS0720EN/blob/d21144e3e9610d5dfc826119686983d36c4f8502/EDASQL%20(2).ipynb

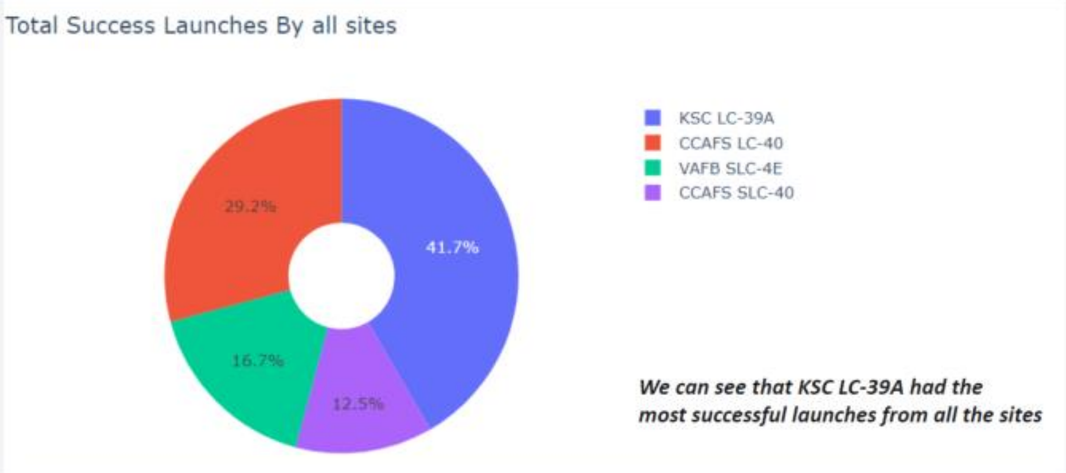
Build an Interactive Map with Folium



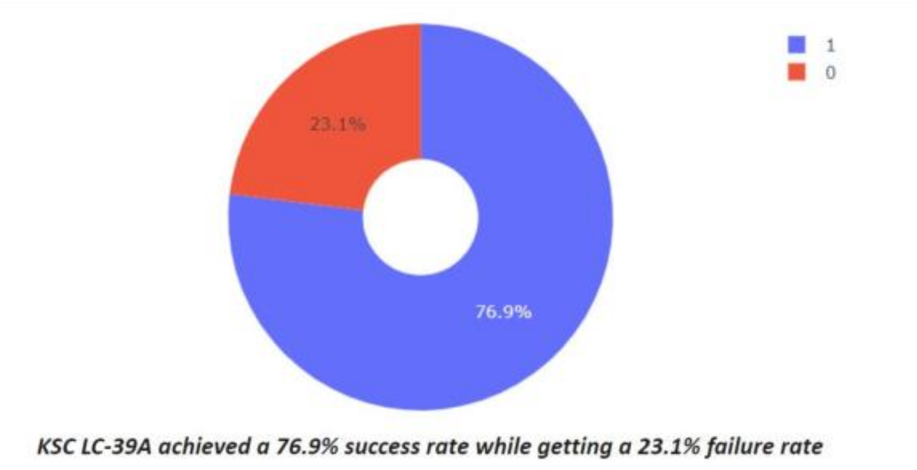
Map markers have been added to the map with aim to finding an optimal location for building a launch site

https://github.com/kkoti30/DS0720EN/blob/d21144e3e9610d5dfc826119686983d36c4f8502/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

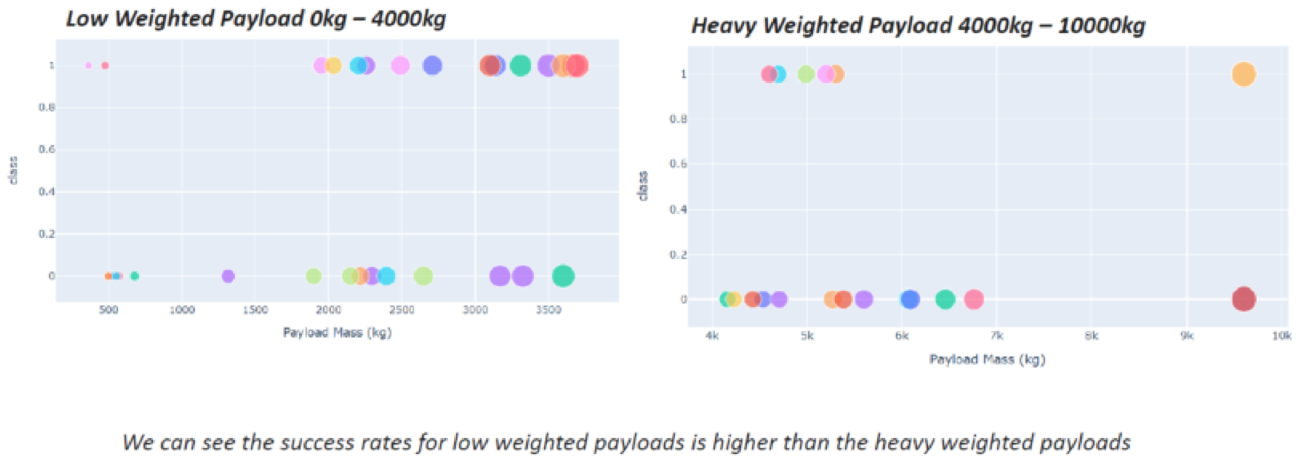
Build a Dashboard with Plotly Dash



We can see that KSC LC-39A had the most successful launches from all the sites



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

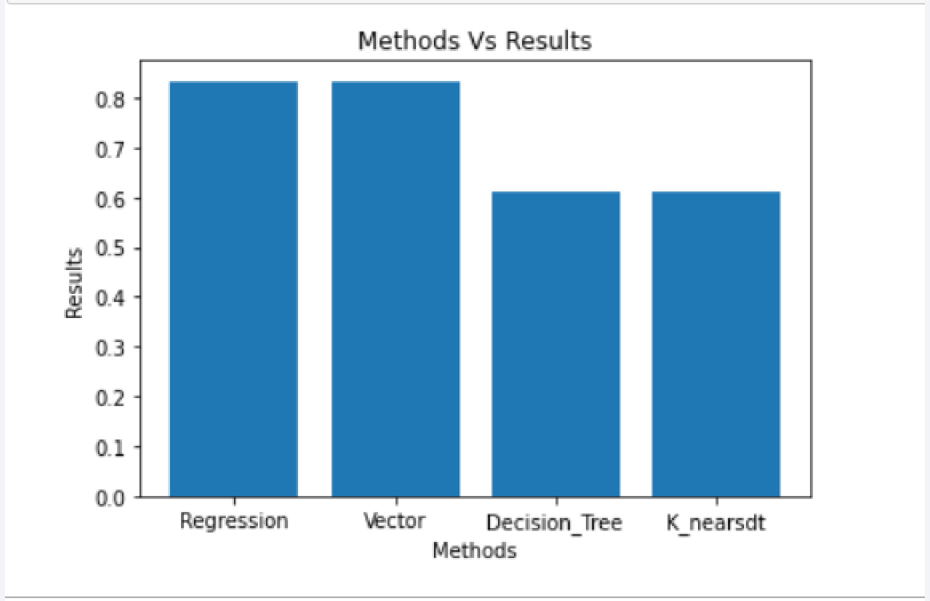


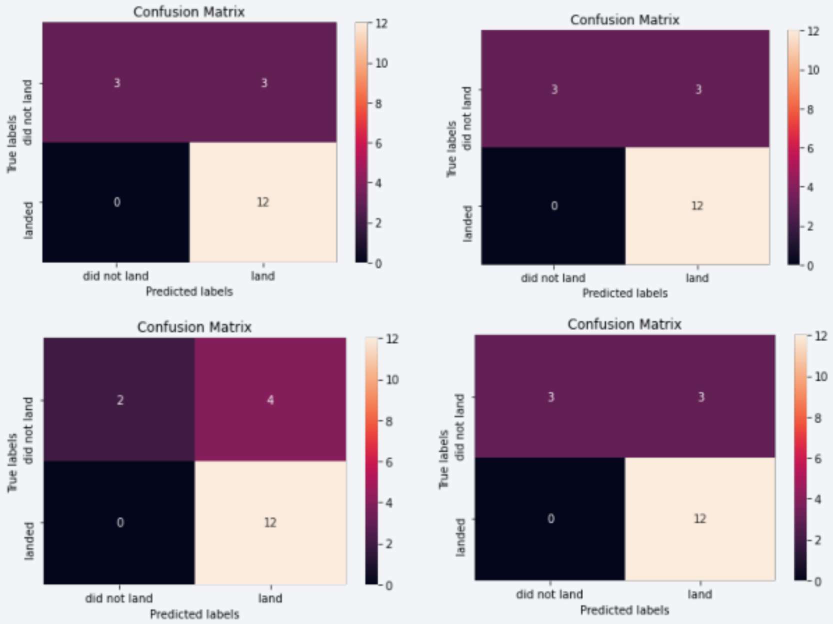
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

https://github.com/kkoti30/DS0720EN/blob/07e7d6e2fc32be6511edcd96483f46629d14ad5/Dashboards%20with%20Poly.ipynb

Predictive Analysis (Classification)

Regression Model and Vector Models have 83% Accuracy





https://github.com/kkoti30/DS0720EN/blob/07e7d6e2fc32be6511edccd96483f46629d14ad5/SpaceX-ML-Prediction.ipynb

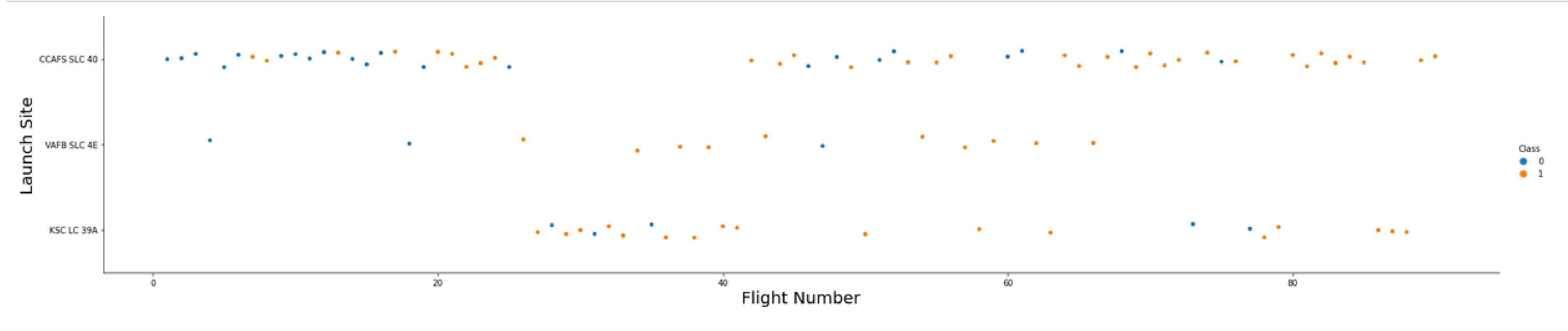
Results

- Regression and Vector models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

Section 2

Insights drawn from EDA

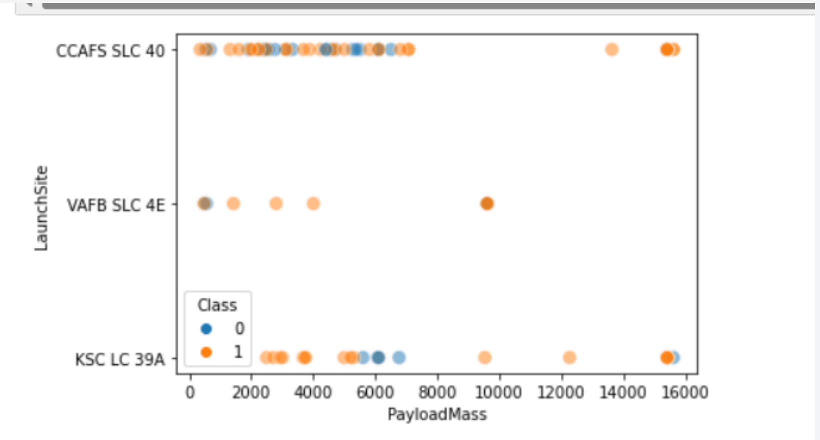
Flight Number vs. Launch Site



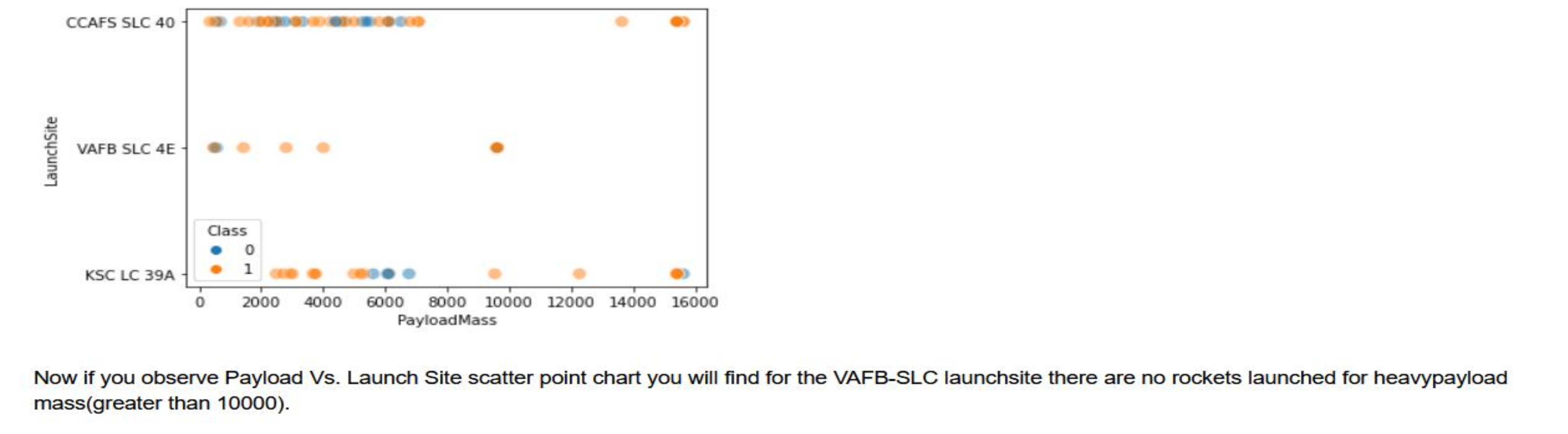
Maximum number of the flights are launched through the Launch site CCAFS LC-40

Payload vs. Launch Site

Show a scatter plot of Payload vs. Launch Site



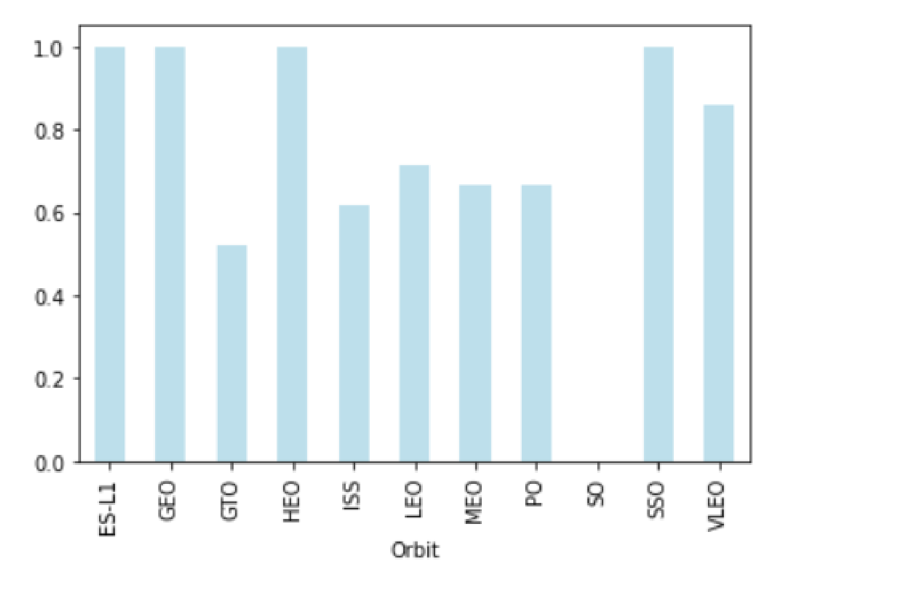
Show the screenshot of the scatter plot with explanations



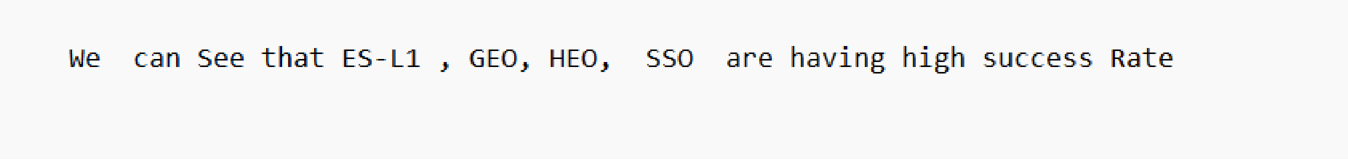
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

Show a bar chart for the success rate of each orbit type



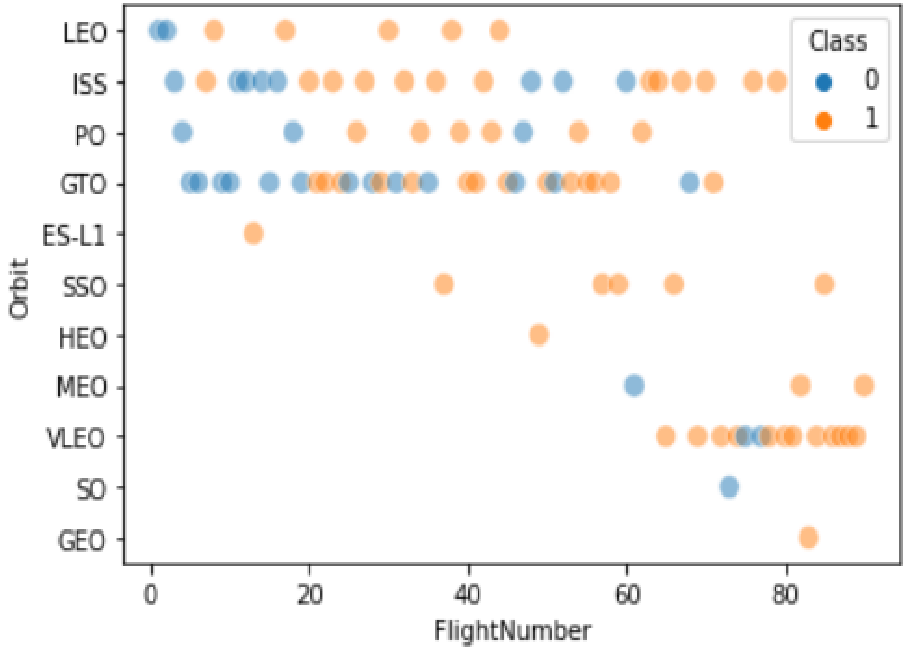
Show the screenshot of the scatter plot with explanations



```
We  can See that ES-L1 , GEO, HEO,  SSO  are having high success Rate
```

Flight Number vs. Orbit Type

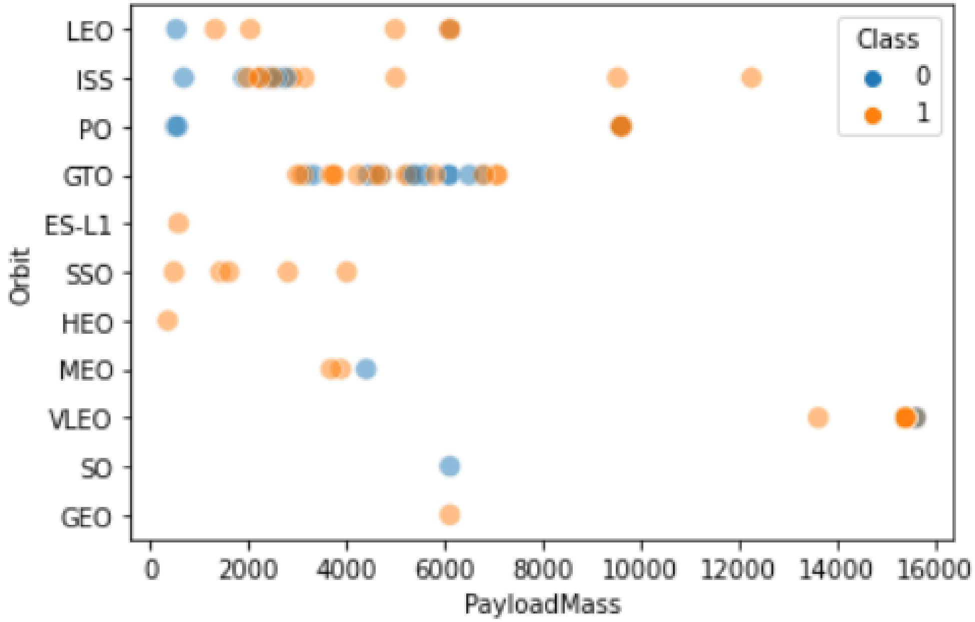
Show a scatter point of Flight number vs. Orbit type



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

Show a scatter point of payload vs. orbit type



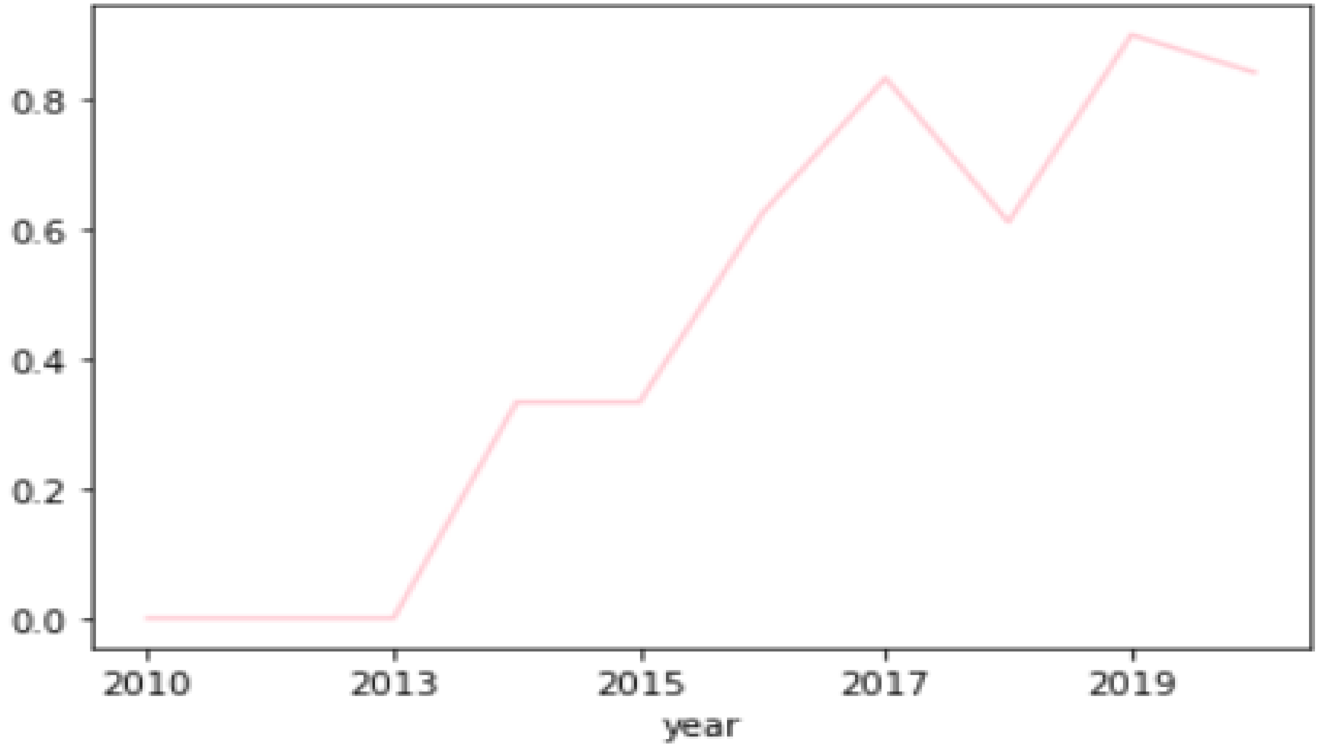
Show the screenshot of the scatter plot with explanations

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

Show a line chart of yearly average success rate

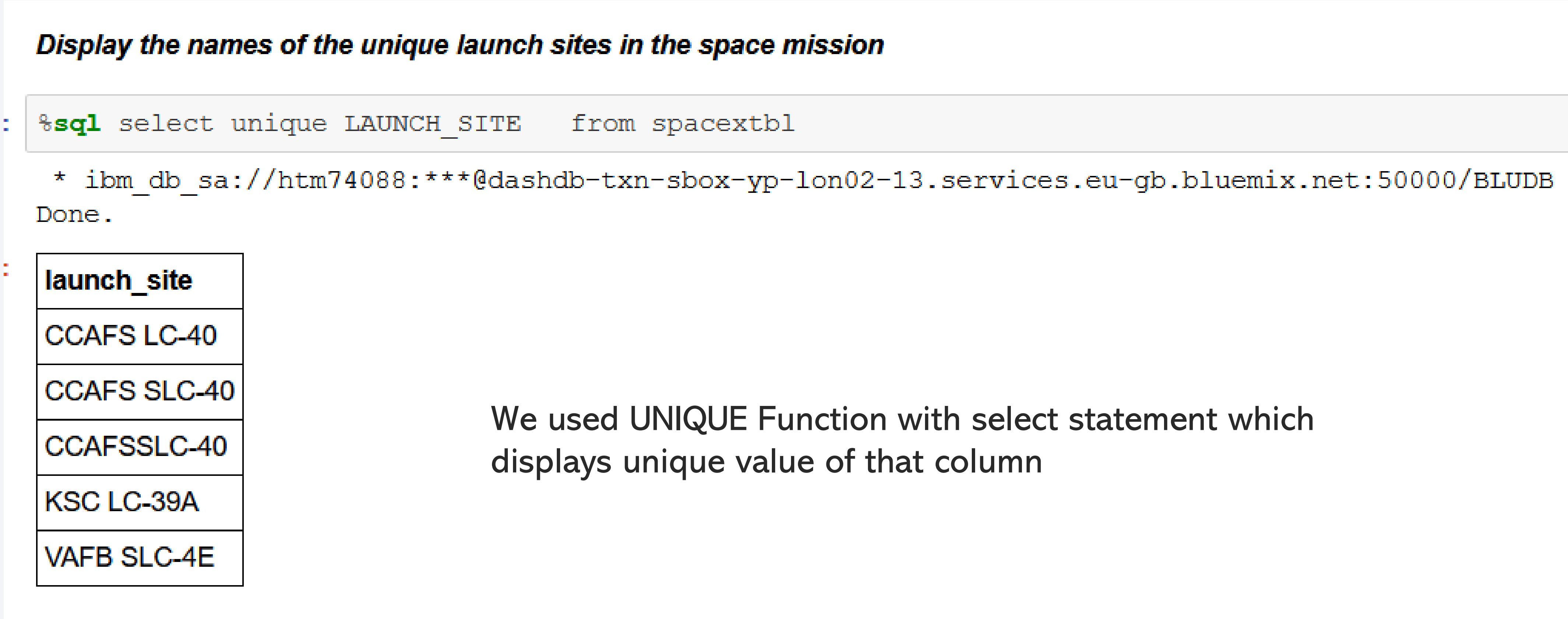


Show the screenshot of the scatter plot with explanations

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



Display the names of the unique launch sites in the space mission

```
%sql select unique LAUNCH_SITE   from spacextbl
```

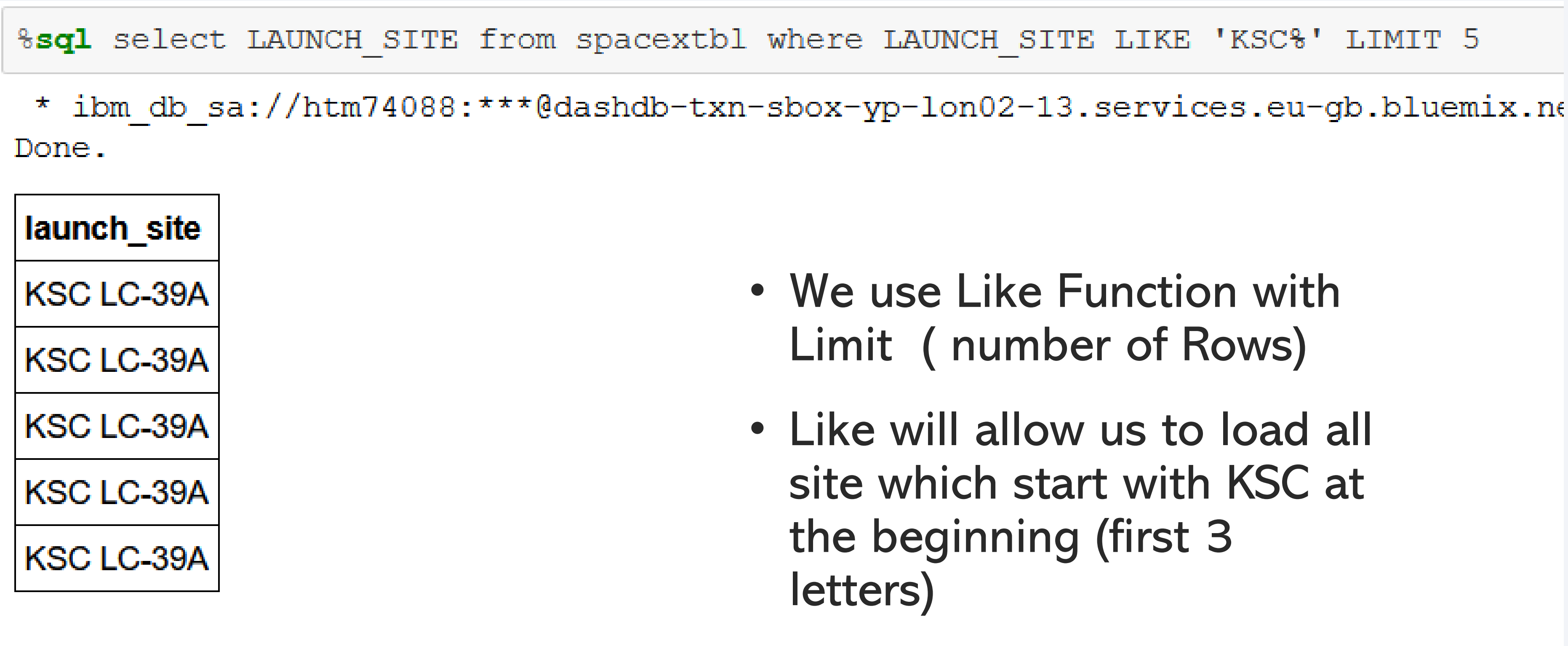
```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

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

We used UNIQUE Function with select statement which displays unique value of that column

Launch Site Names Begin with 'KSC'

Find 5 records where launch sites' names start with `KSC`



```
%sql select LAUNCH_SITE from spacextbl where LAUNCH_SITE LIKE 'KSC%' LIMIT 5
```

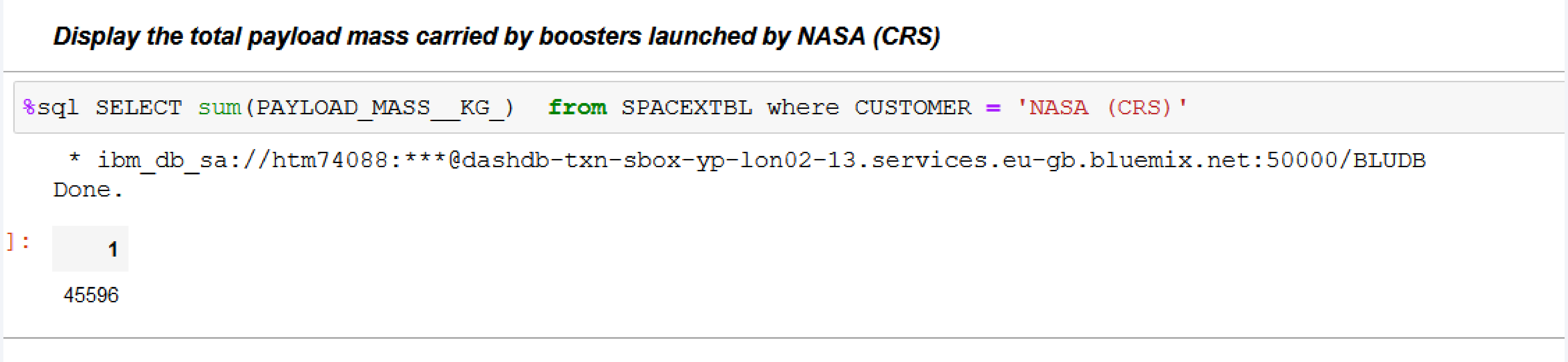
```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.ne
Done.
```

launch_site
KSC LC-39A
KSC LC-39A
KSC LC-39A
KSC LC-39A
KSC LC-39A

- We use Like Function with Limit (number of Rows)
- Like will allow us to load all site which start with KSC at the beginning (first 3 letters)

Total Payload Mass

Calculate the total payload carried by boosters from NASA



Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT sum(PAYLOAD_MASS__KG_)  from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

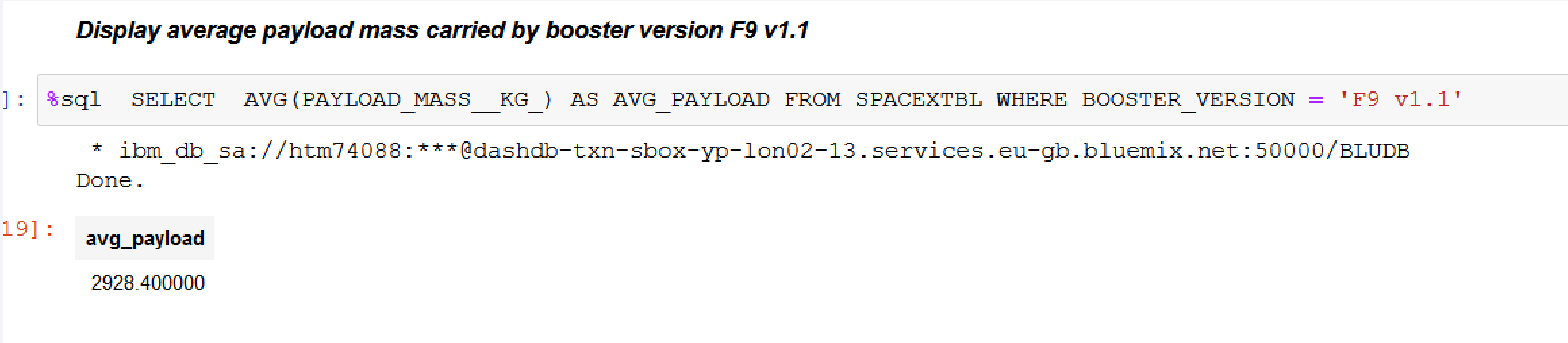
]:

1
45596

In our Quarry using SUM function with a where condition of the requirement column

Average Payload Mass by F9 v1.1

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



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

```
]: %sql  SELECT  AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

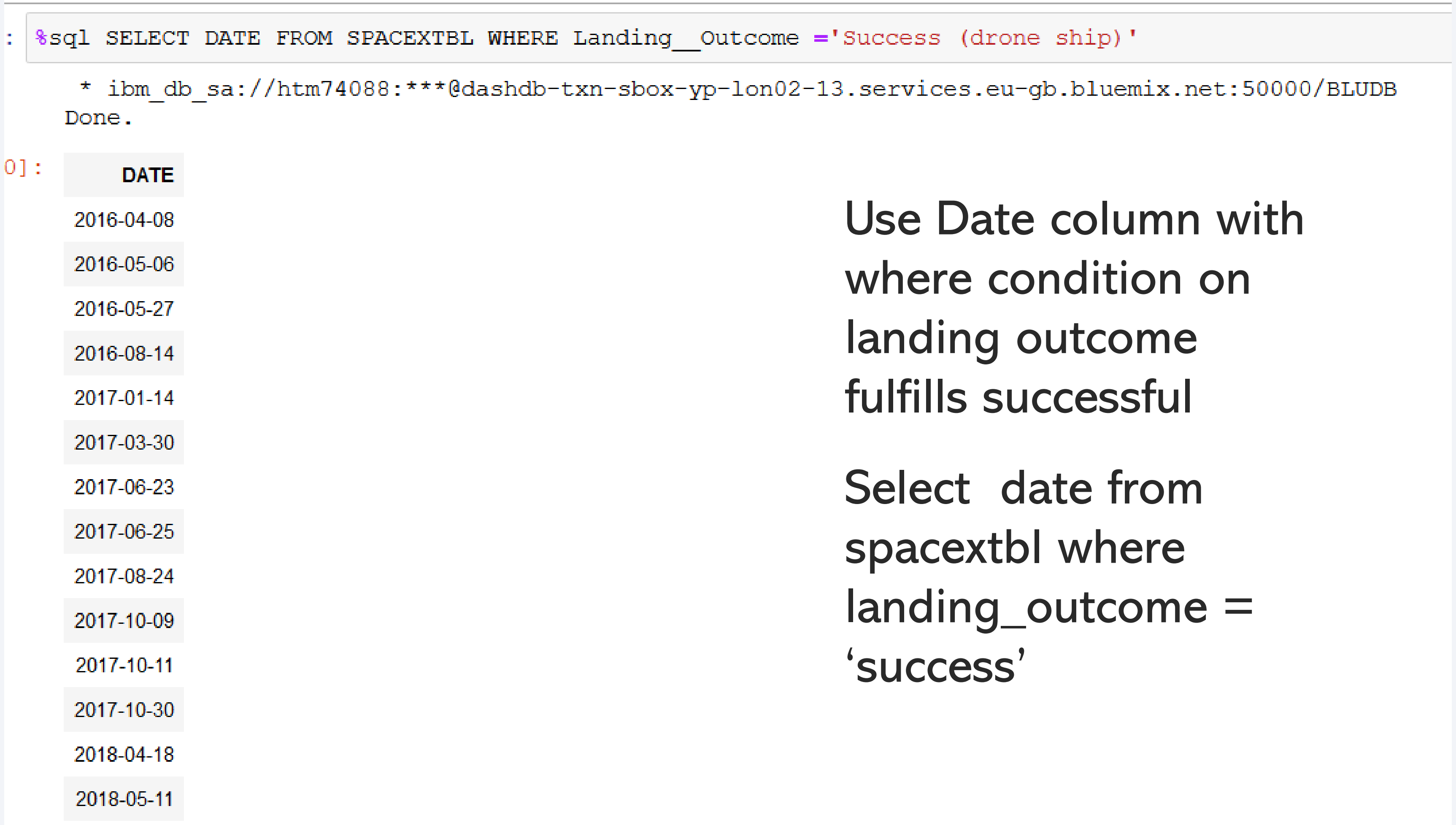
19]:

avg_payload
2928.400000

- Used AVG function on PYLOAD_MASS_KG column with where condition on column Booster_version

First Successful Ground Landing Date

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



```
%sql SELECT DATE FROM SPACEXTBL WHERE Landing__Outcome ='Success (drone ship)'
```

```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

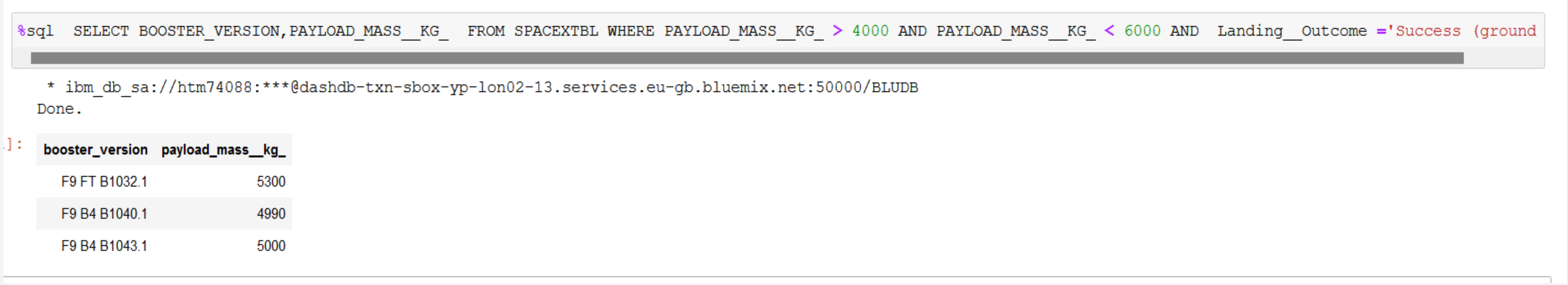
DATE
2016-04-08
2016-05-06
2016-05-27
2016-08-14
2017-01-14
2017-03-30
2017-06-23
2017-06-25
2017-08-24
2017-10-09
2017-10-11
2017-10-30
2018-04-18
2018-05-11

Use Date column with where condition on landing outcome fulfills successful

Select date from spacextbl where landing_outcome = ‘success’

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 BOOSTER_VERSION,PAYLOAD_MASS__KG_  FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000 AND  Landing__Outcome ='Success (ground
```

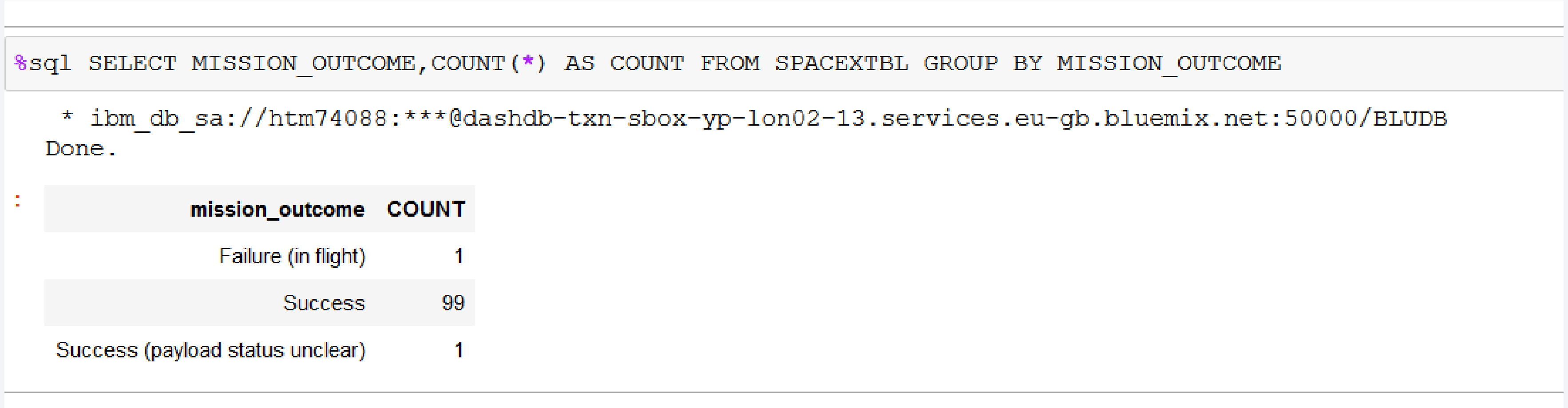
```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

booster_version	payload_mass__kg_
F9 FT B1032.1	5300
F9 B4 B1040.1	4990
F9 B4 B1043.1	5000

- We use logical operator functions to get greater than and less than value of a column

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes



```
%sql SELECT MISSION_OUTCOME,COUNT(*) AS COUNT FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

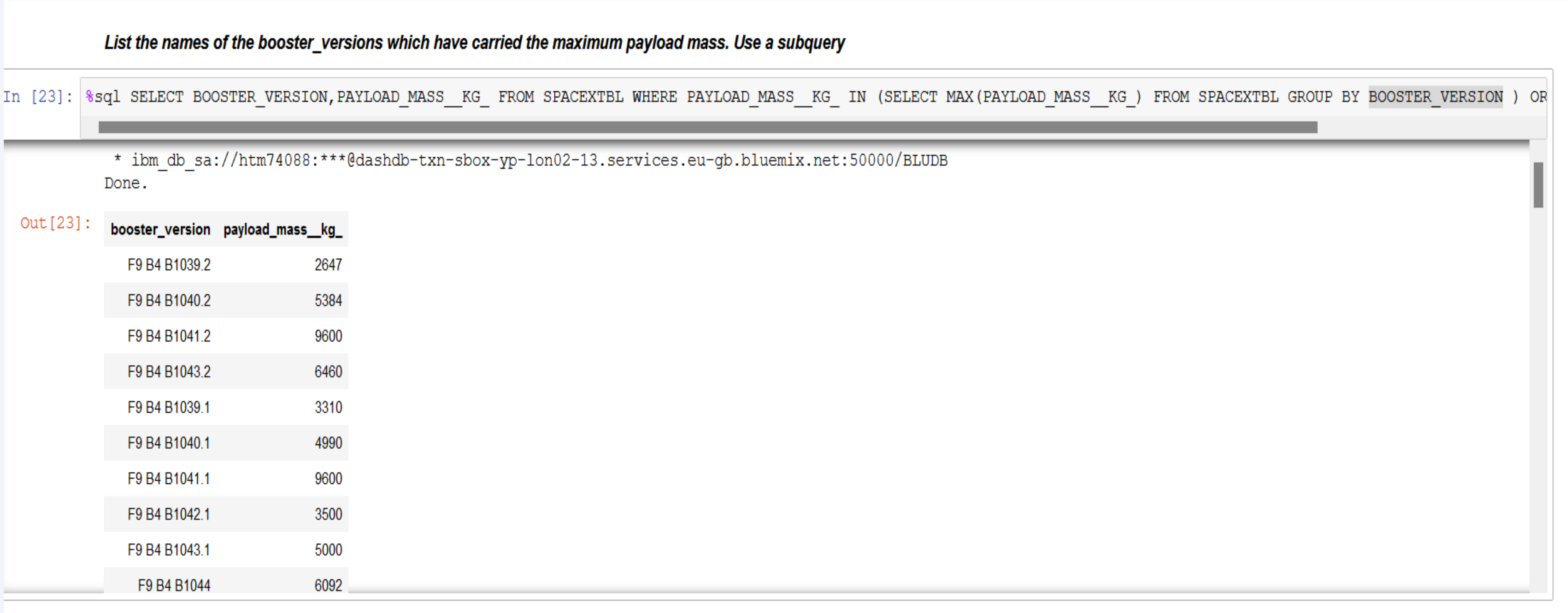
```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

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

- We use count function with Group by on mission outcome column

Boosters Carried Maximum Payload

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



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
In [23]: %sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL GROUP BY BOOSTER_VERSION ) OR
```

```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

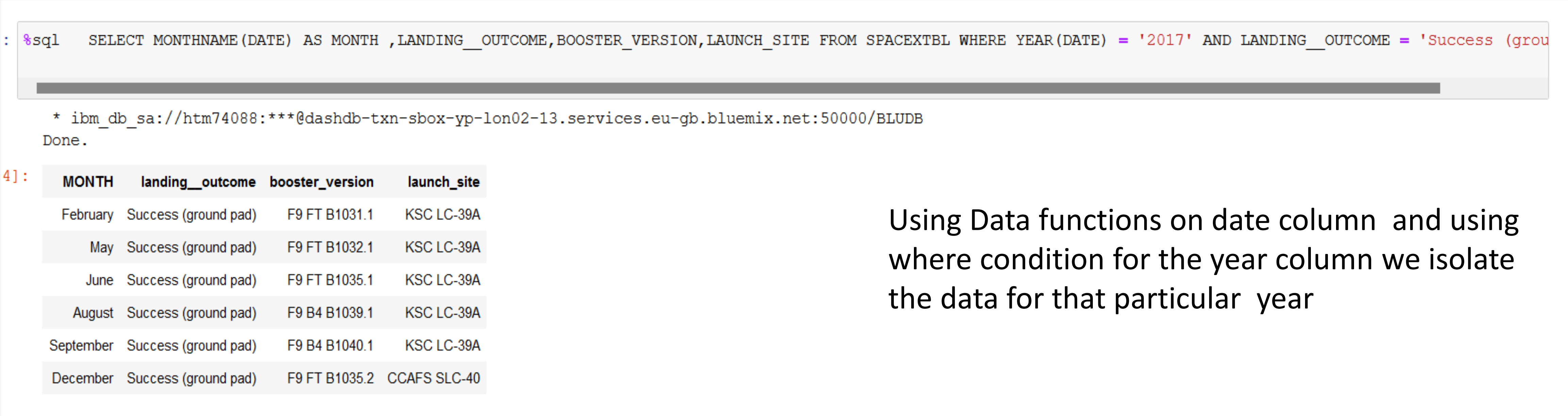
Out[23]:

booster_version	payload_mass__kg_
F9 B4 B1039.2	2647
F9 B4 B1040.2	5384
F9 B4 B1041.2	9600
F9 B4 B1043.2	6460
F9 B4 B1039.1	3310
F9 B4 B1040.1	4990
F9 B4 B1041.1	9600
F9 B4 B1042.1	3500
F9 B4 B1043.1	5000
F9 B4 B1044	6092

- %sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL GROUP BY BOOSTER_VERSION) ORDER BY BOOSTER_VERSION

2015 Launch Records

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch site for the months in year 2017



```
: %sql    SELECT MONTHNAME(DATE) AS MONTH ,LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL WHERE YEAR(DATE) = '2017' AND LANDING__OUTCOME = 'Success (grou
```

```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

4]:

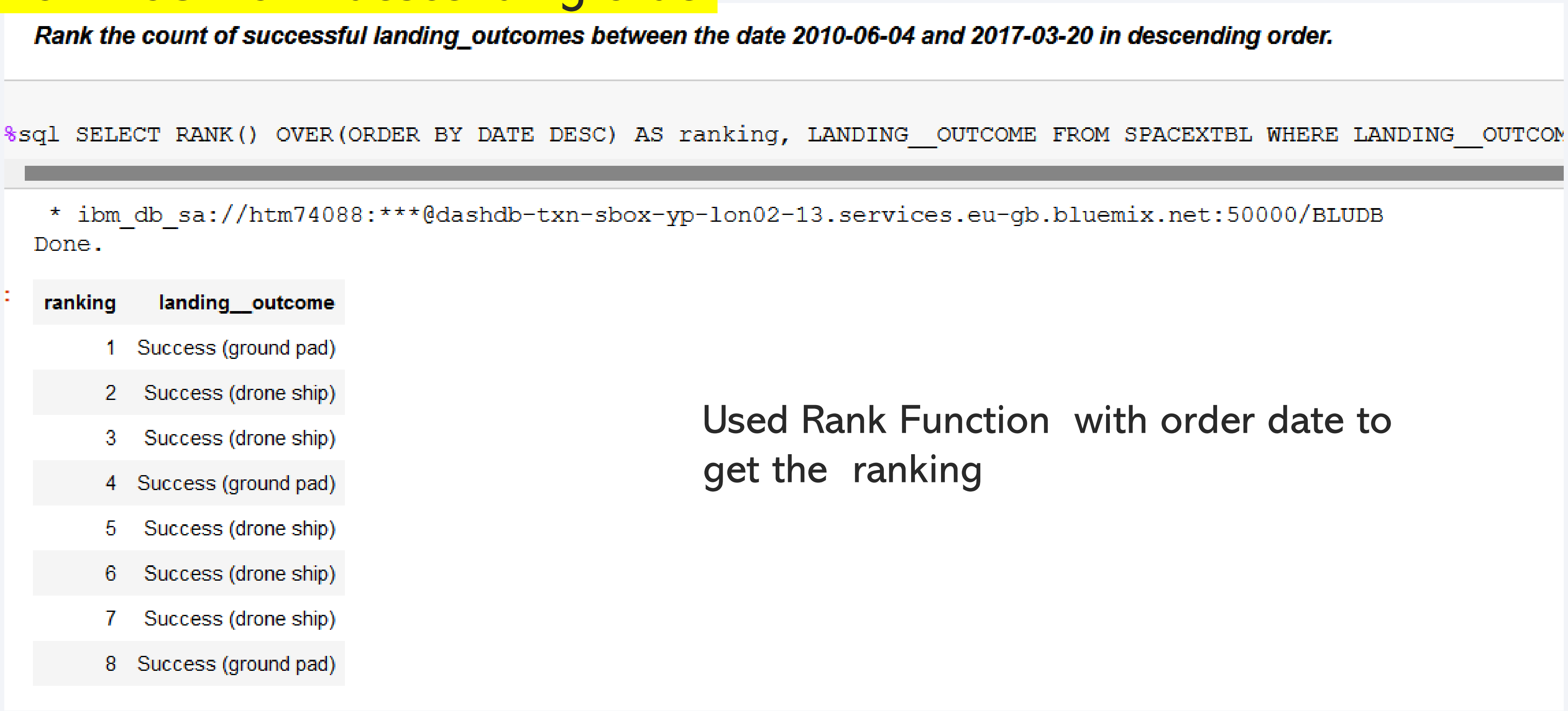
MONTH	landing__outcome	booster_version	launch_site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

Using Data functions on date column and using where condition for the year column we isolate the data for that particular year

%sql   SELECT MONTHNAME(DATE) AS MONTH ,LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL WHERE YEAR(DATE) = '2017' AND LANDING__OUTCOME = 'Success (ground pad)'

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

Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order



Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

```
%sql SELECT RANK() OVER(ORDER BY DATE DESC) AS ranking, LANDING__OUTCOME FROM SPACEXTBL WHERE LANDING__OUTCOM
```

```
 * ibm_db_sa://htm74088:***@dashdb-txn-sbox-yp-lon02-13.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

ranking	landing__outcome
1	Success (ground pad)
2	Success (drone ship)
3	Success (drone ship)
4	Success (ground pad)
5	Success (drone ship)
6	Success (drone ship)
7	Success (drone ship)
8	Success (ground pad)

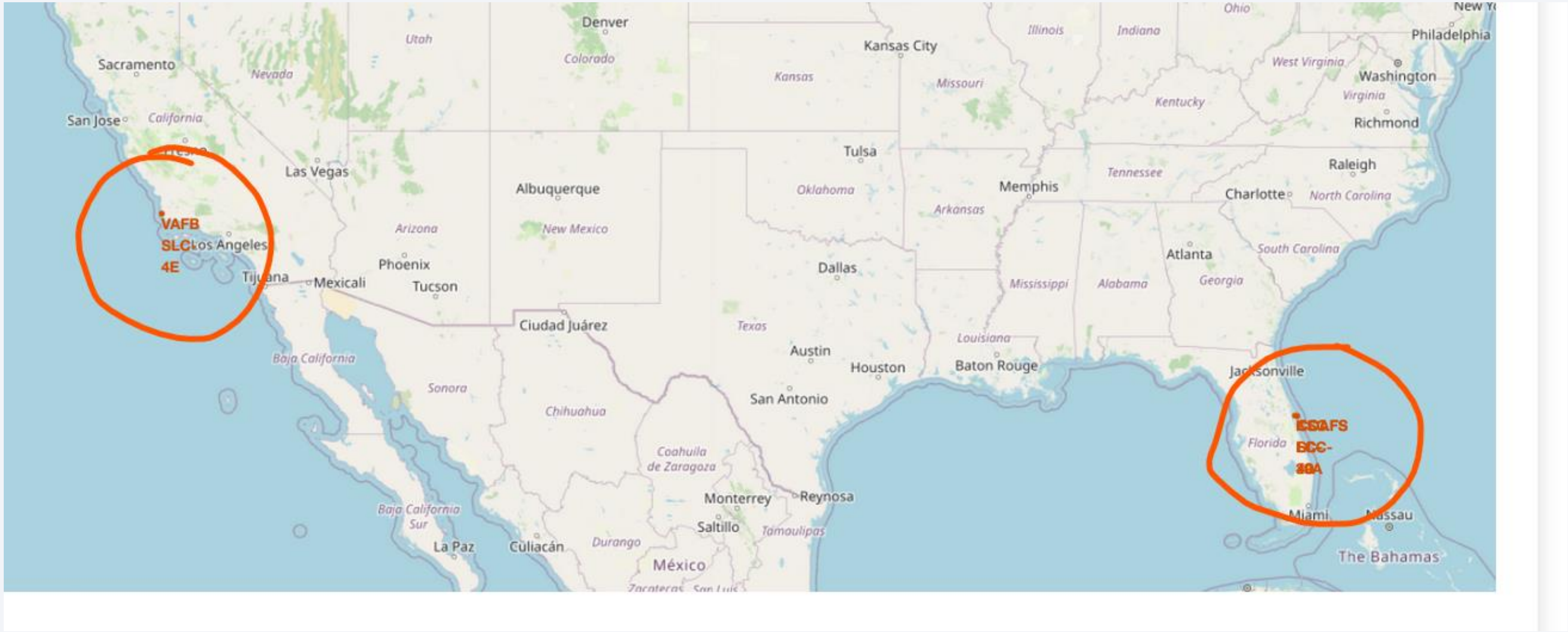
Used Rank Function with order date to get the ranking

%sql SELECT RANK() OVER(ORDER BY DATE DESC) AS ranking, LANDING__OUTCOME FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20'

Section 3

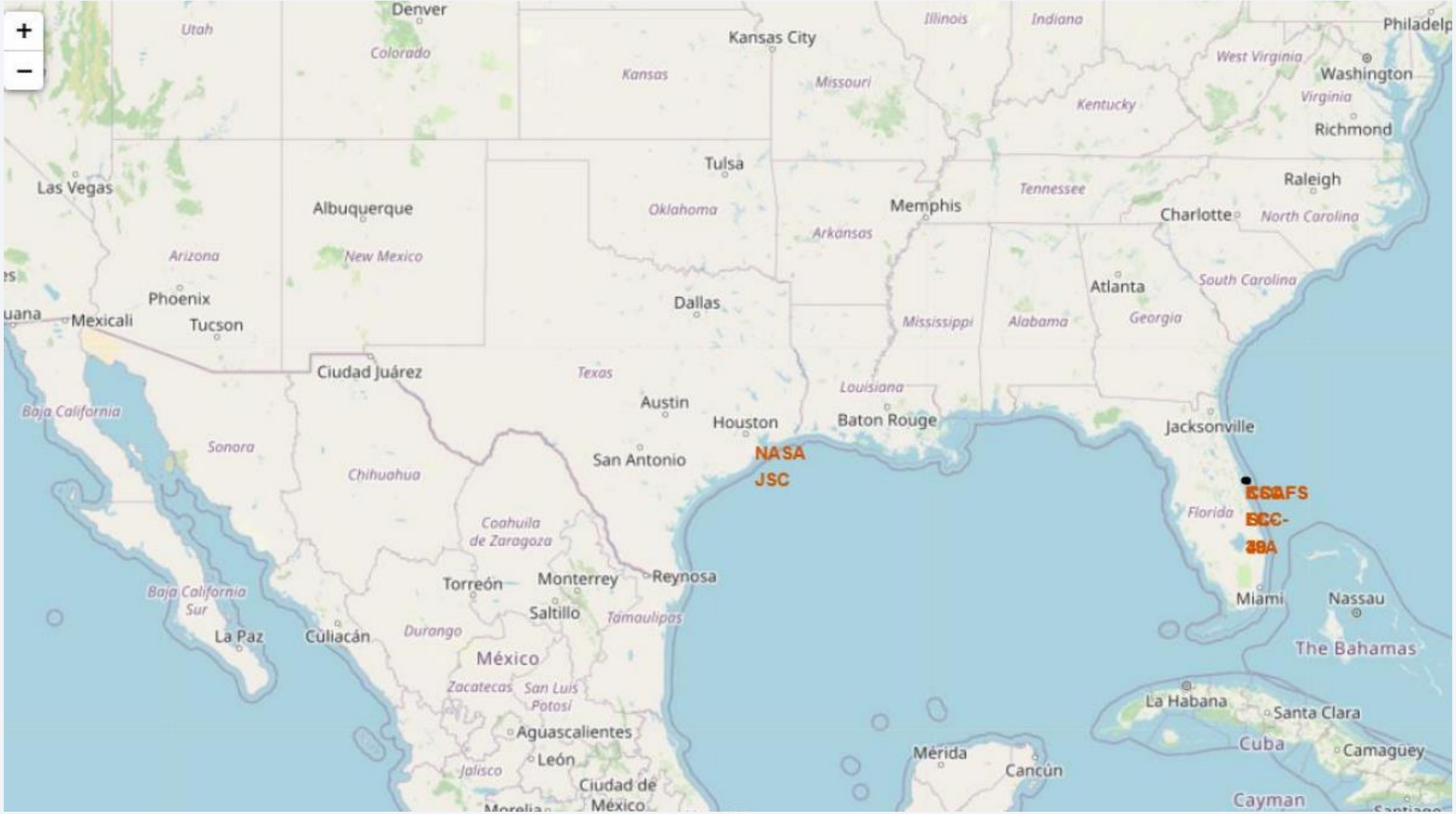
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

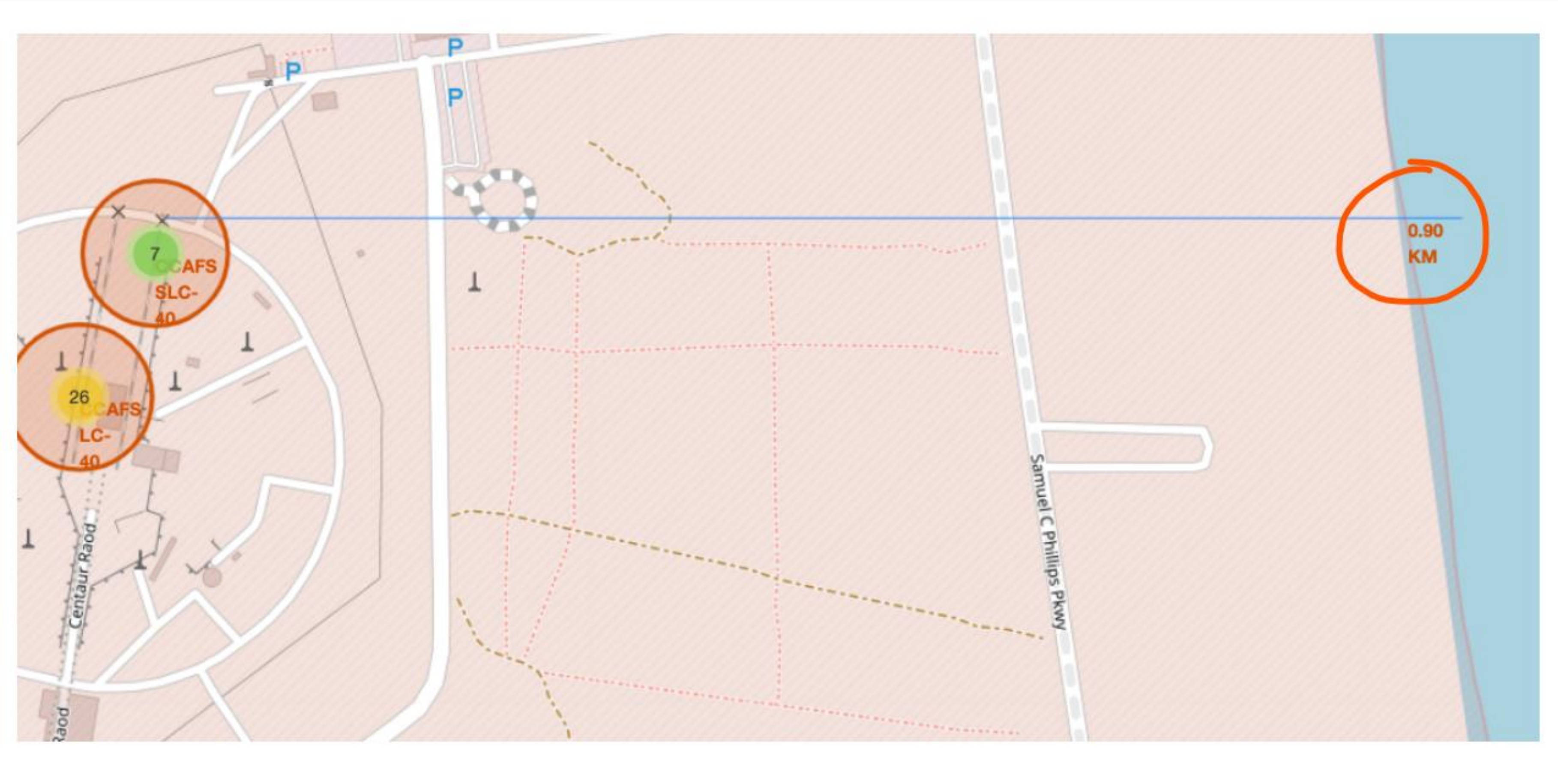


- We use folium package to create folium maps and we use marker to pick the launch sites from the data

<Folium Map Screenshot 2>


+
−
NASA
JSC
CCAFS
SLC-
39A
Denver
Utah
Colorado
Kansas
Kansas City
Missouri
Illinois
Indiana
Kentucky
West Virginia
Washington
Philadelp
Virginia
Richmond
Las Vegas
Albuquerque
Tulsa
Oklahoma
Arkansas
Memphis
Tennessee
Charlotte
Raleigh
North Carolina
South Carolina
Arizona
New Mexico
Phoenix
Tucson
Mexicali
Dallas
Mississippi
Alabama
Atlanta
Georgia
Ciudad Juárez
Texas
Austin
Houston
Louisiana
Baton Rouge
Jacksonville
San Antonio
Baja California
Sonora
Chihuahua
Florida
Coahuila de Zaragoza
Torreón
Monterrey
Reynosa
Saltillo
Tamaulipas
Miami
Nassau
Baja California Sur
La Paz
Culiacán
Durango
México
The Bahamas
Zacatecas
San Luis Potosí
La Habana
Santa Clara
Aguascalientes
Cuba
Camagüey
Mérida
Cancún
Jalisco
León
Ciudad de México
Morelia
Cayman
Santiago

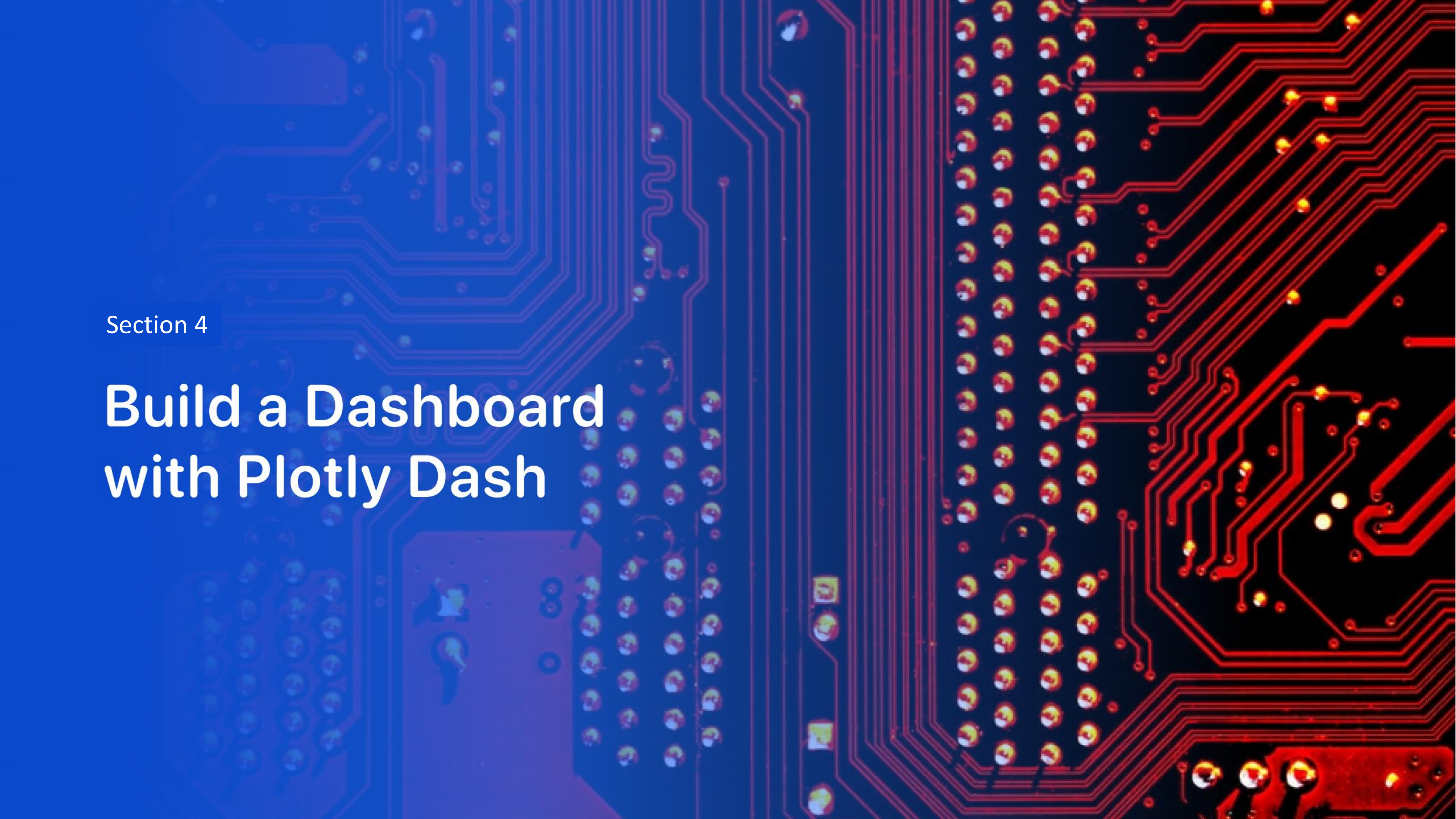
<Folium Map Screenshot 3>



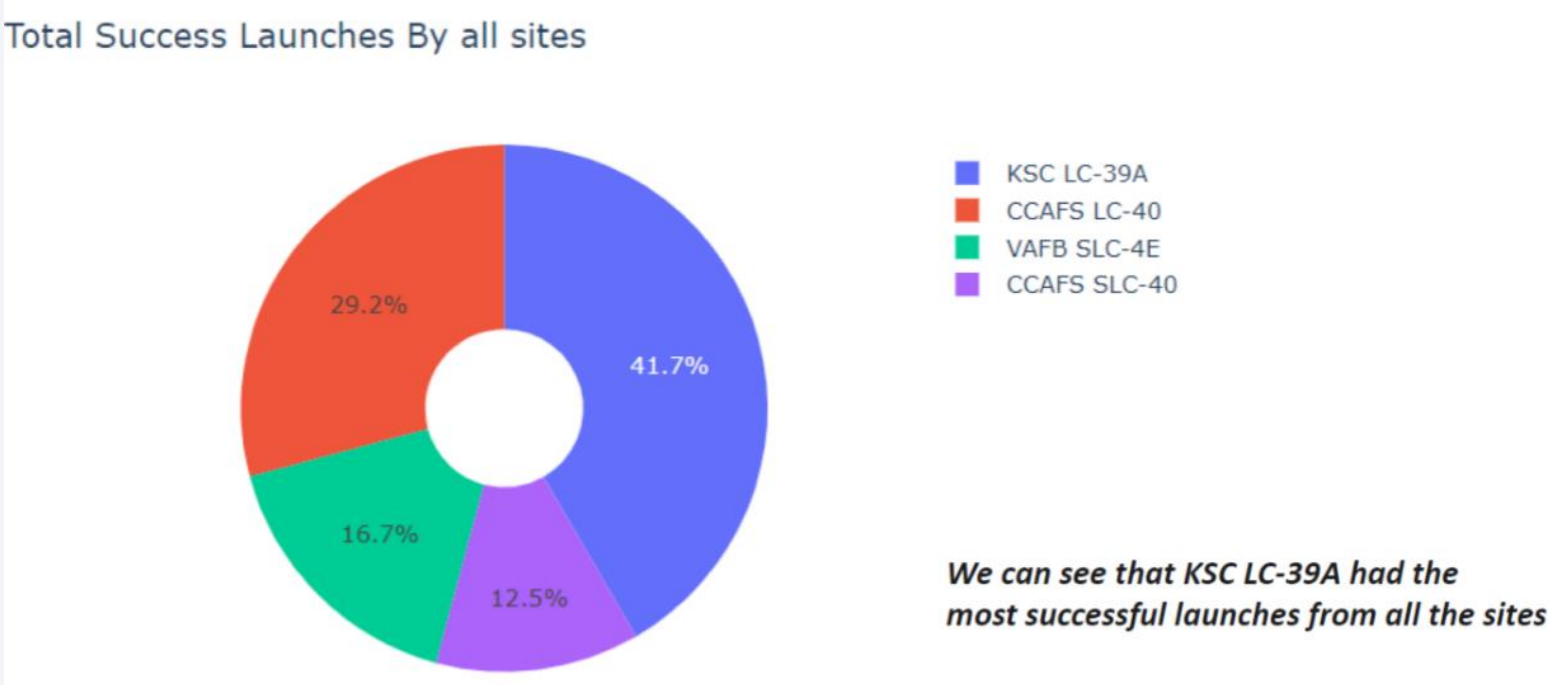
P
P
P
7
CCAFS
SLC-
40
26
CCAFS
LC-
40
Centaur Raod
Raod
Samuel C Phillips Pkwy
0.90
KM

Section 4

Build a Dashboard with Plotly Dash



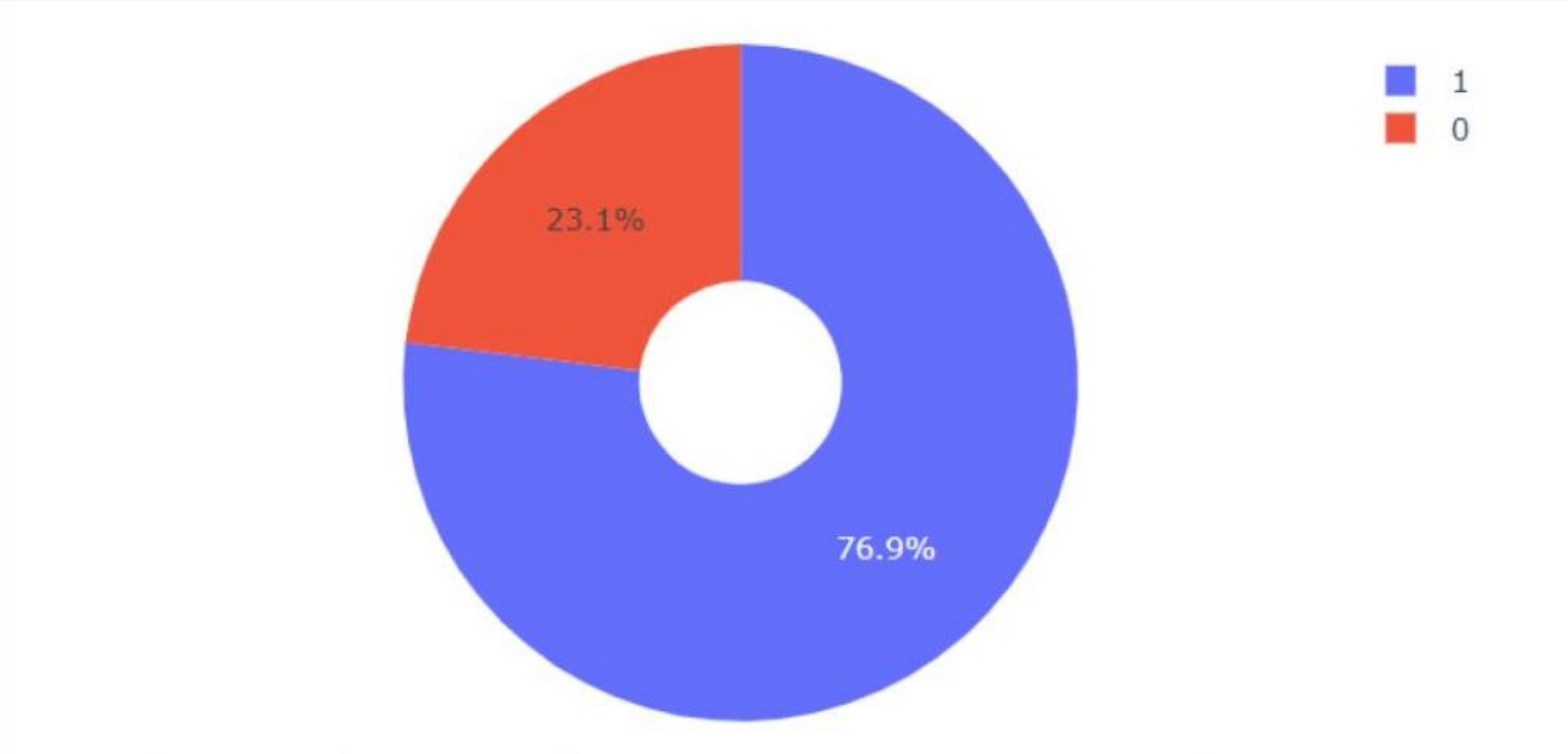
<Dashboard Screenshot 1>



Total Success Launches By all sites

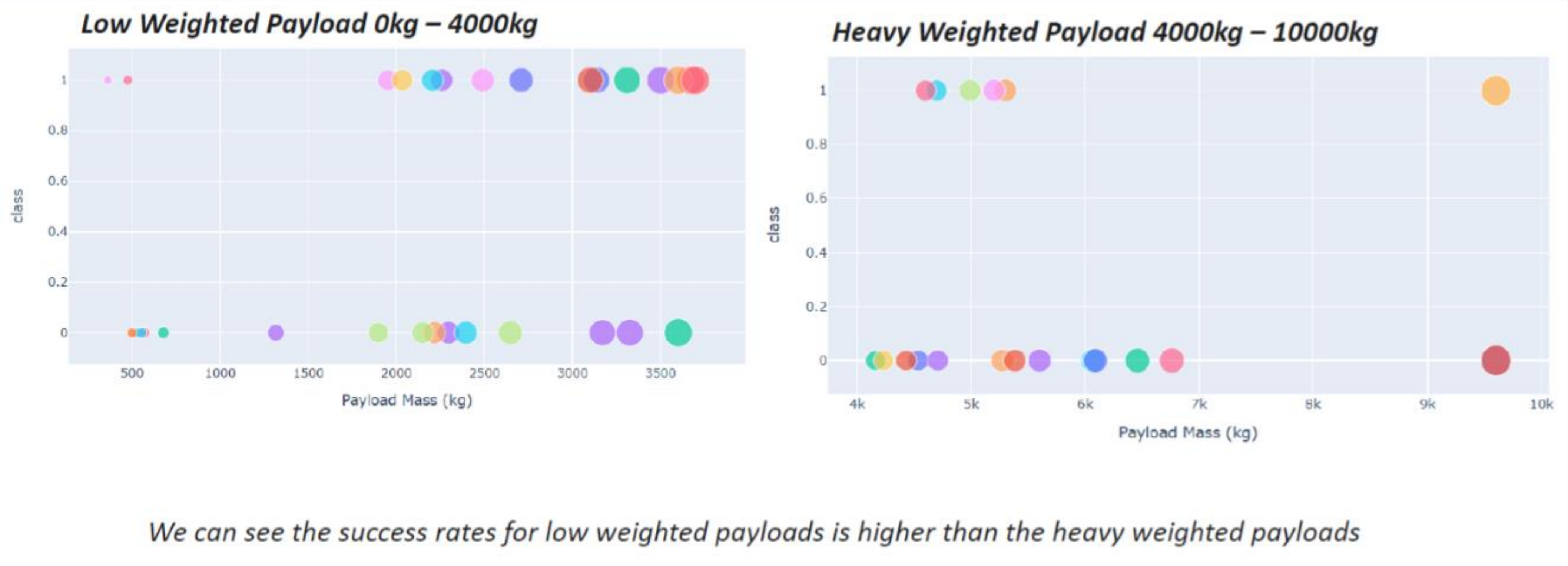
We can see that KSC LC-39A had the most successful launches from all the sites

<Dashboard Screenshot 2>



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Dashboard Screenshot 3>



Low Weighted Payload 0kg – 4000kg

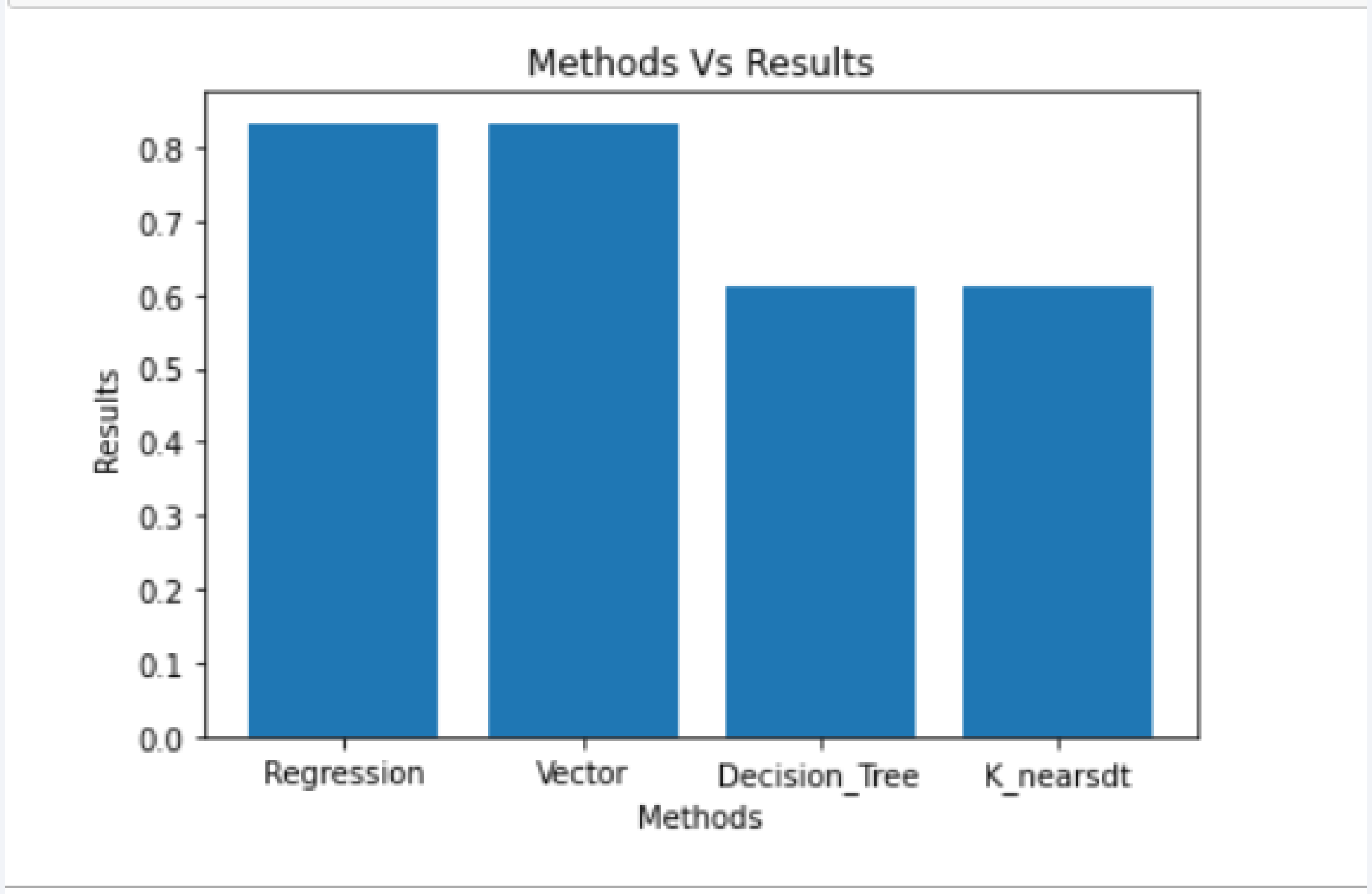
Heavy Weighted Payload 4000kg – 10000kg

We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

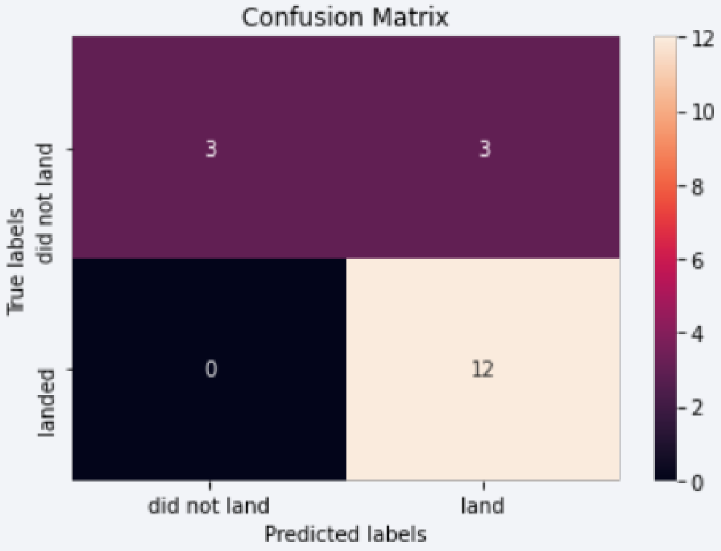
Section 5

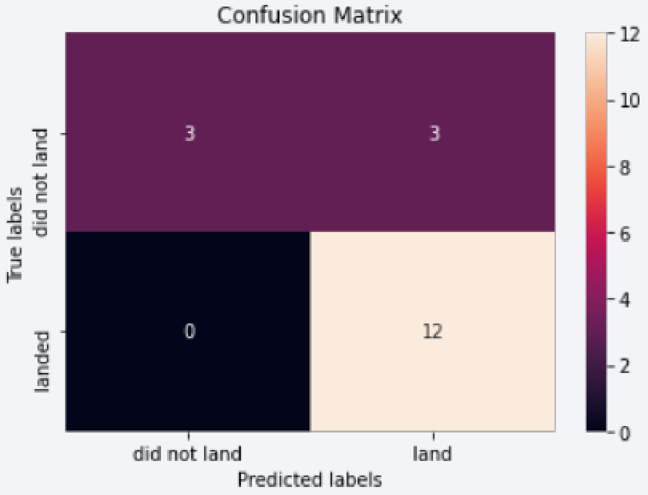
Predictive Analysis (Classification)

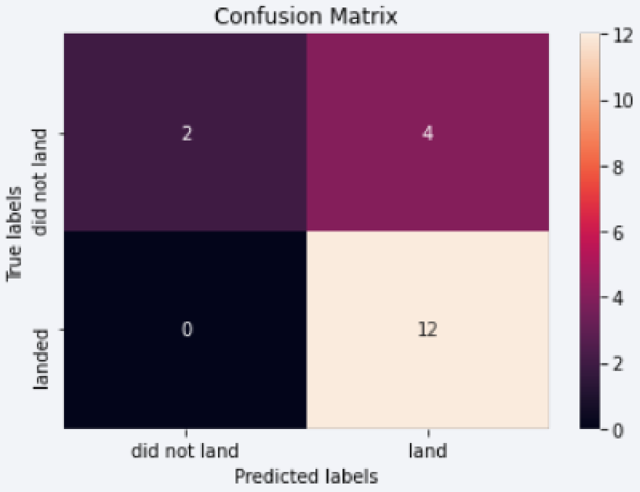
Classification Accuracy

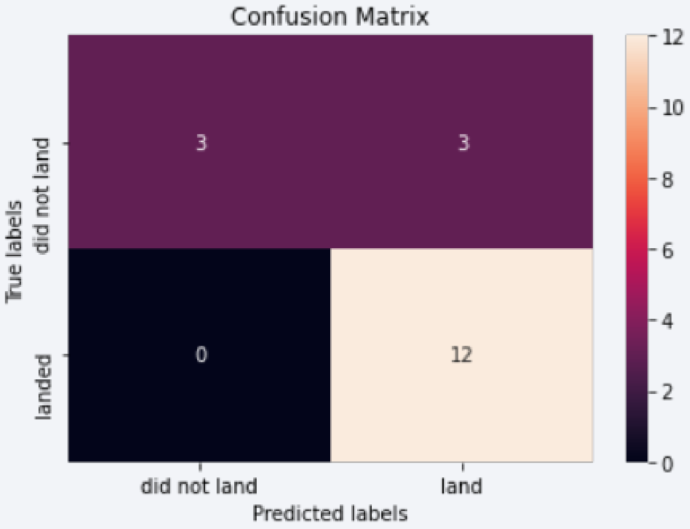


Confusion Matrix









Conclusions

- Regression and Vector models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

Thank you!