IBM Developer
SKILLS NETWORK

DATA SCIENCE CAPSTONE PROJECT – SPACE| RACE

Winning Space Race with Data Science

< Name: Hang Le Thi Thuy>
<Date: Dec 30, 2021>

Outline

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

Executive Summary

- Summary of methodologies

- Data Collection

- Data Wrangling

- EDA with SQL and Visualization

- Visual Analytics with Folium lab

- Build an Interactive Dashboard with Ploty Dash

- Summary of all results

- Display the data set

- Explored the data set

- Graph and Dashboard display.

Introduction

- Project background and context

In In the era of science and technology, it is increasingly developing and focusing on development. Several companies have been developing in the field of space travel. However, launching rockets will require a lot of costs as well as other conditions. And data science has become part of this process to help scientists predict and evaluate the next rocket launches to bring success. Specially, Falcon 9 from Space X.

- Problems you want to find answers

After many times rockets launched, we will have the data of previous launches. Base on that, we can build up the method to predict Falcon 9 first stage will land successfully or not.

Section 1

Methodology

Methodology

Data collection methodology:

- Use the API to extract information in the launch data and web scraping to collect Falcon 9 historical launch records. (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

- Perform data wrangling
  - Find some patterns in the data and determine what would be the label for training supervised models.
  - Checked missing data and transform the data to the type that useful to modeling.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Choose input data and output data. Transform that to the suitable type
  - Split the data, build and fit models. Then evaluate model.

Data Collection

- Describe how data sets were collected.
- Present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

1. Requesting rocket launch data from SpaceX API.

2. Requested JSON results and turn it into a Pandas dataframe.

3. Get information about the launches using the IDs given for each launch and the functions. Then assign to dictionary, dataframe.

The GitHub URL

```
# 1
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
# 2
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# 3
getBoosterVersion(data)
```

click to expand output; double click to hide output

the list has now been update

```
BoosterVersion[0:5]
```

```
3]: ['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

Finally lets construct our dataset using the data we have obtained. We we combine the columns into a dictionary.

```
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}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
df.describe()
```

Data Collection – SpaceX API

4. Get the dataframe to only include Falcon 9 launches.

5. Checked missing data and deal with it. Export the data to a CSV file for next sections.

The GitHub URL

```
# 4
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

Now that we have removed some values we should reset the FlgihtNumber column

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

5

```
data_falcon9.isnull().sum()
```

```
# Calculate the mean value of PayloadMass column
mean_payloadmass = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(mean_payloadmass)
```

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

Data Collection - Scraping

1. Get page from HTML and made a beautyfulsoup object.

2. Find all tables and extract column name one by one.

3. Create an empty dictionary with keys then extracted from table rows.

4. Create a dataframe from it and export it to a CSV file.

The GitHub URL

1

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

```
soup = BeautifulSoup(page, 'html5lib')
```

2

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

```
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
```

3

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
```

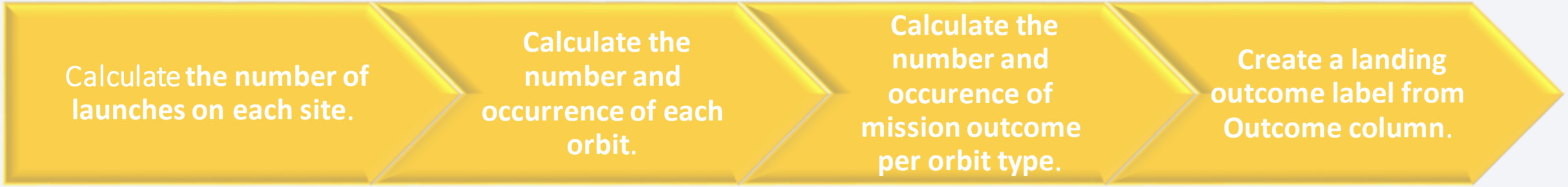
4

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df.head()
```

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

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.



The GitHub URL

EDA with Data Visualization

Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumber and Orbit type, Payload and Orbit type.

Scatter charts

Visualize the launch success yearly trend.

Line Chart

Bar Chart

Visualize the relationship between success rate of each orbit type.

The GitHub URL

EDA with SQL

- 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.

The GitHub URL

Build an Interactive Map with Folium

- To visualize the SpaceX launch dataset using more interactive visual analytics. Took a look in the coordinates data for each site and see that just plain numbers, can't point exactly where those sites are on the map.
- So we marked all launch sites with Map object and an initial center location is NASA Johnson Space Center at Houston, Texas (the main site) with a blue circle. Then add a circle for each launch site in data frame launch_sites.
- Next, we added the launch outcomes for each site, and see which sites have high success rates. Markers for all launch records with a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0) on the map in the MarkerCluster.
- Calculate the distances between a launch site to its proximities using Harverine's formula. Plot distance lines to the proximities to measure patterns.

The GitHub URL

Build a Dashboard with Plotly Dash

- Pie Graph: Display the total o first see which site has the largest success count and easy to check the detailed success rate of the site.
- Scatter Graph: We can visually observe how payload may be correlated with mission outcomes for selected site(s).

The GitHub URL

Predictive Analysis (Classification)

Building

- **Load the** dataframe **then** standardize input data and transform output data into Numpy.
- Split **the** data into **training and test data** with test size is 20%.
- **Choose which type of machine learning might be suitable for** the **problem based on the understanding of data.**
- **Create** methods **object then** create GridSearchCV. Fit **the** object and train the dataset.

Evaluation

- Checked **the accuracy** of each model.
- Display the **confusion matrix** plot.

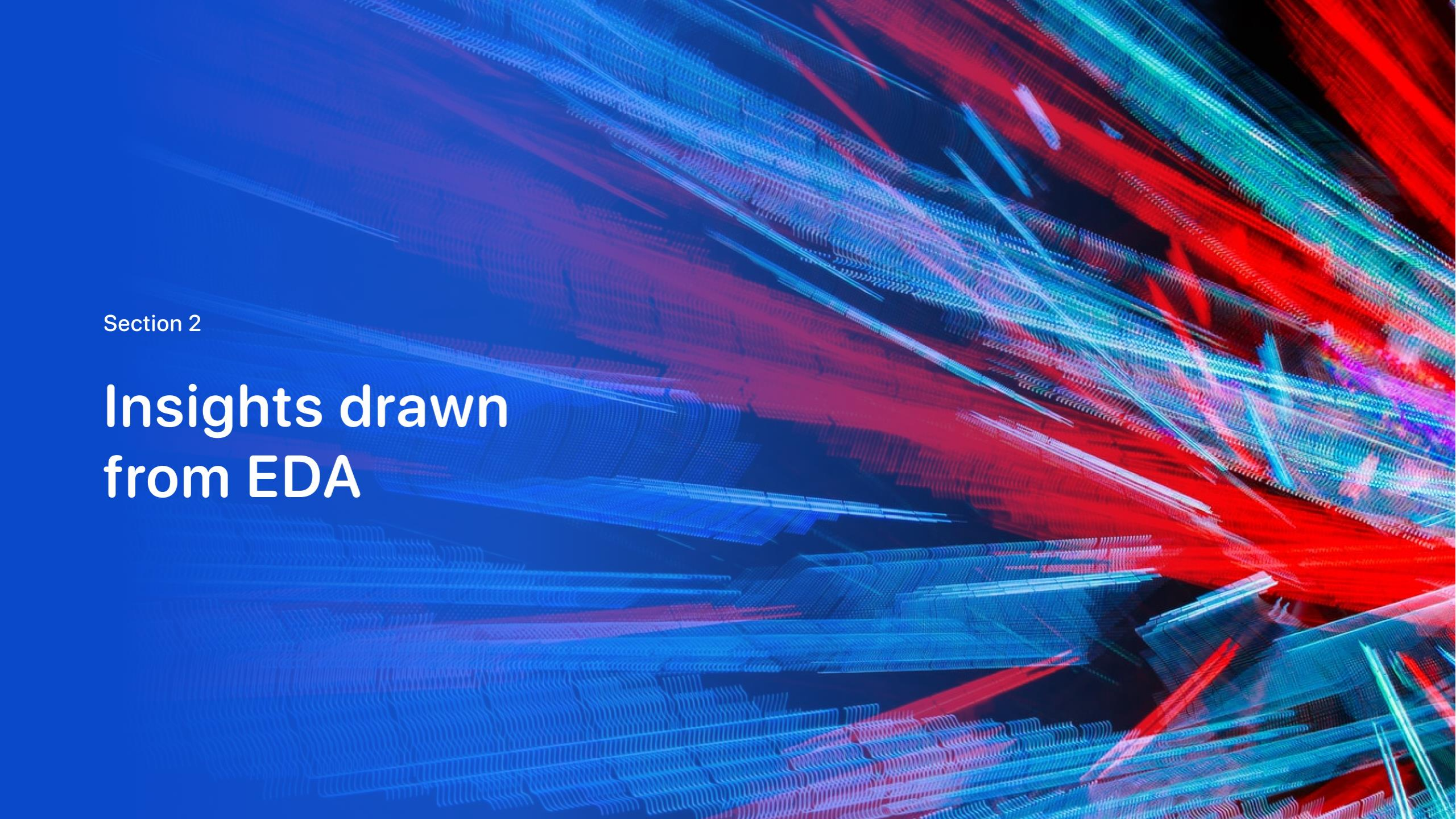
Find the method performs best

- With the best accuracy score. This model is performing to use.

The GitHub URL

Results

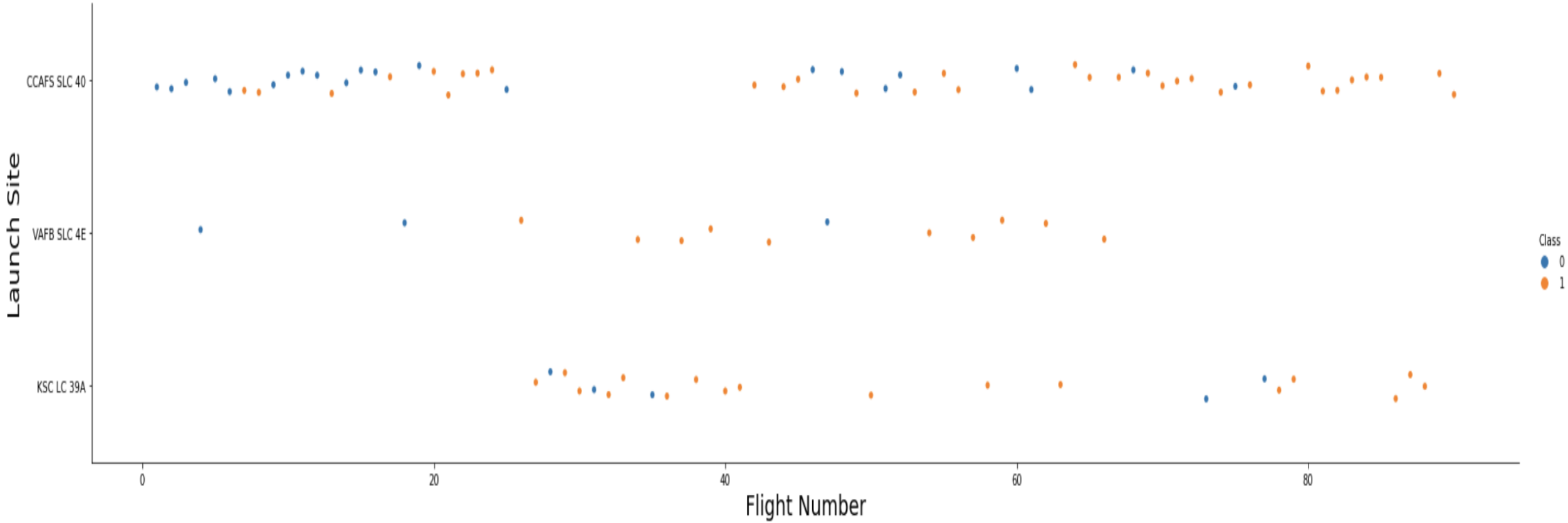
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

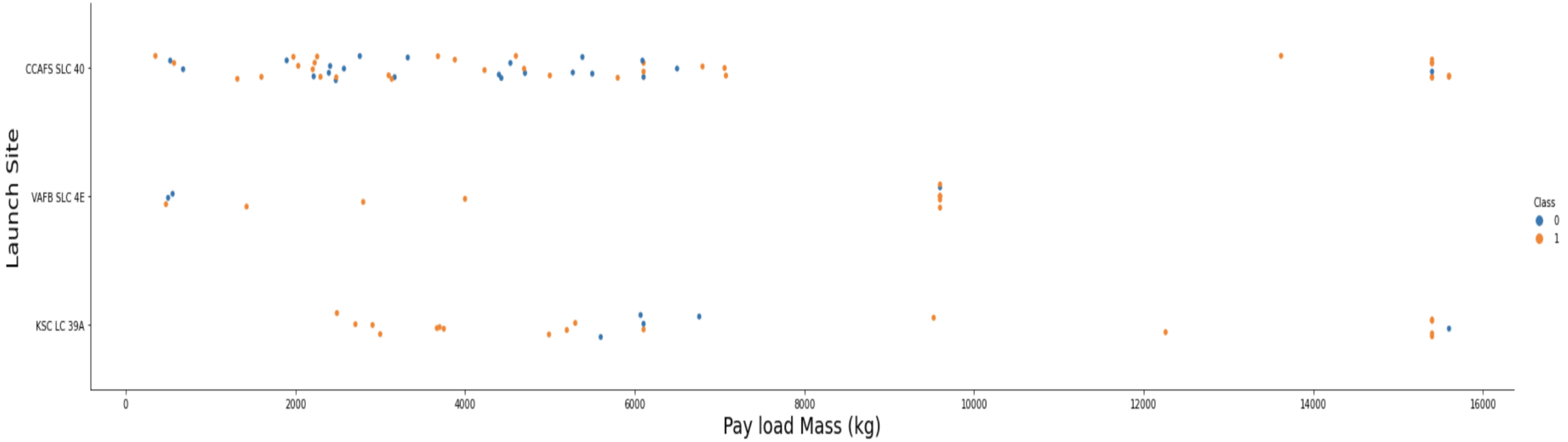
Insights drawn from EDA

Flight Number vs. Launch Site



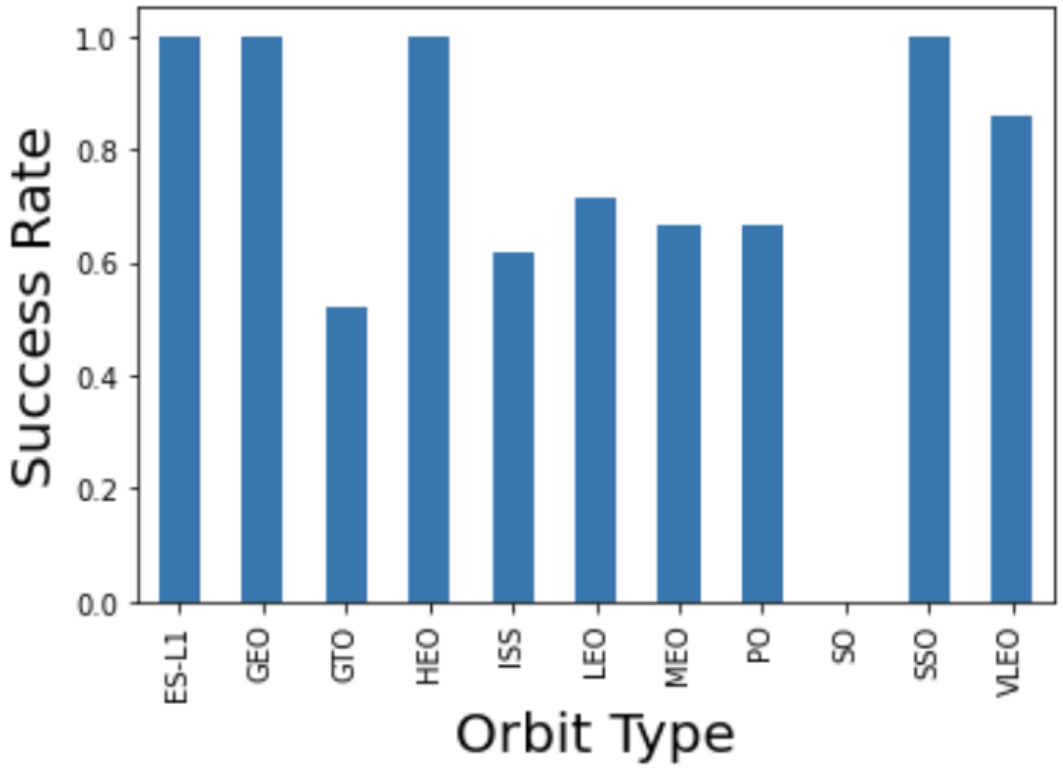
We can see that if the site has much flight that have more rate success.

Payload vs. Launch Site



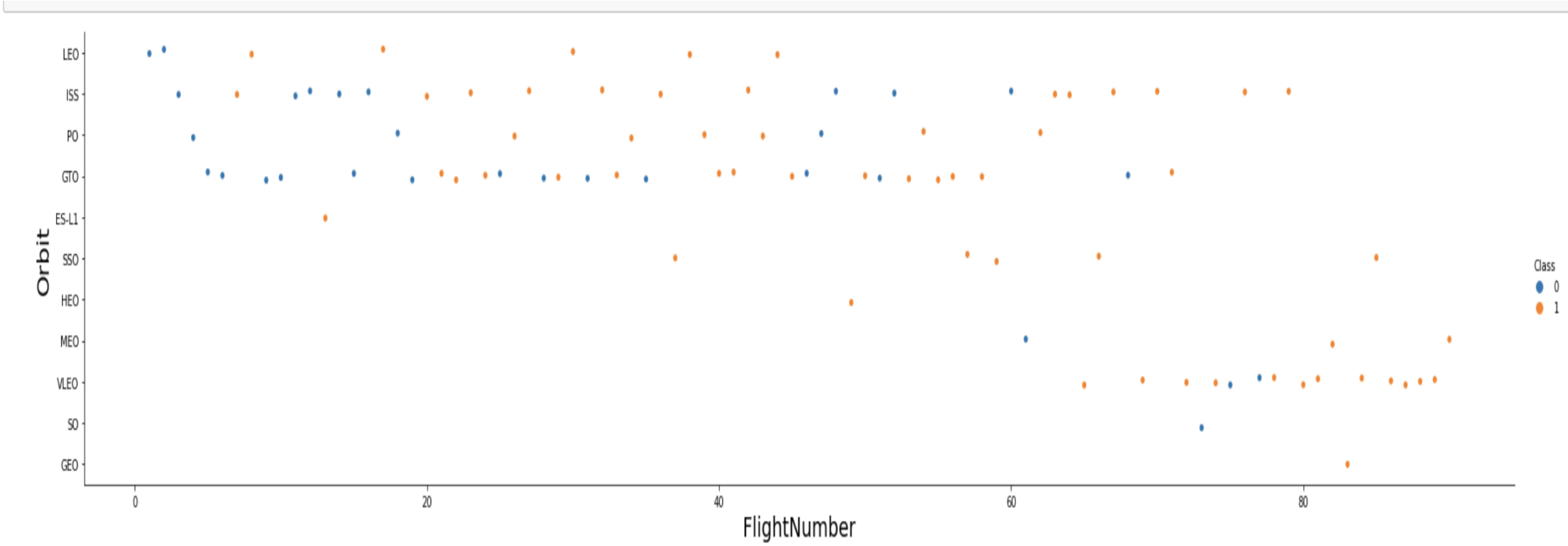
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



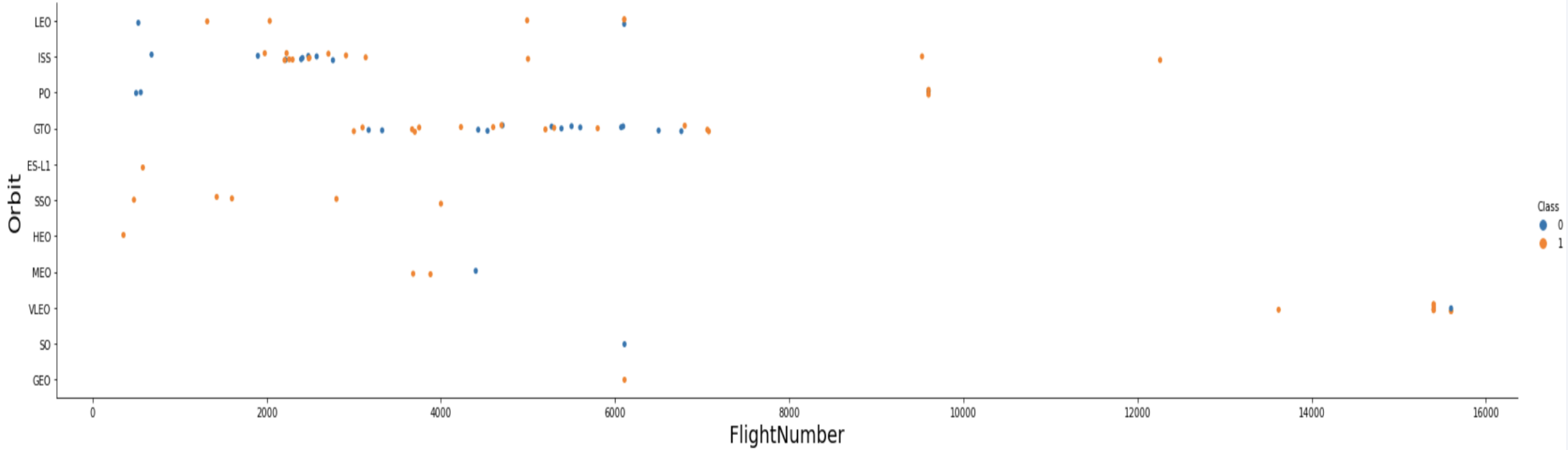
Orbits are ES-L1, GEO, HEO, SSO have high sucess rate.

Flight Number vs. Orbit Type



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.

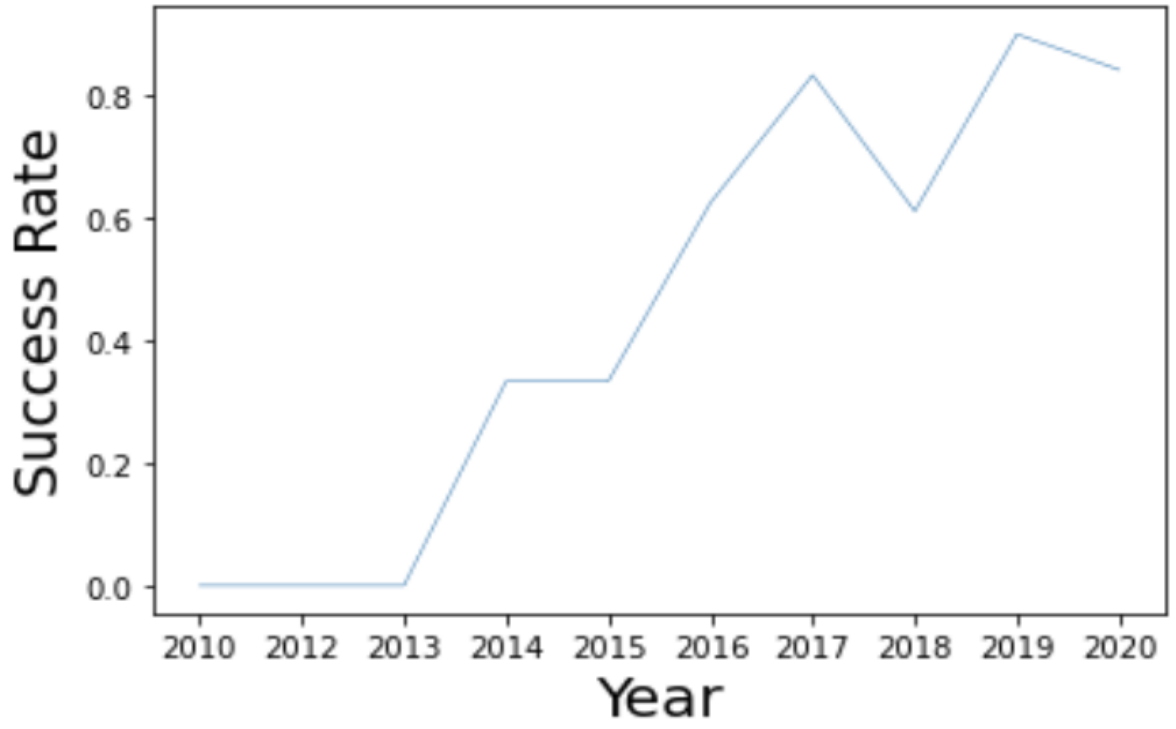
Payload vs. Orbit Type



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

Used "Unique" to find the unique launch sites

```
: %sql select Unique(LAUNCH_SITE) from SPACEXTBL
   * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9
Done.
```

[7]:

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

Launch Site Names Begin with 'CCA'

Used "limit" with condition is site name "like" the word "CCA%"

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnr
Done.
```

]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Total Payload Mass is 619967.

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd
Done.
```

9]:

payloadmass
619967

Average Payload Mass by F9 v1.1

- Average Paylod Mass by F9 v1.1 is 6138.

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c
Done.
```

payloadmass
6138

First Successful Ground Landing Date

- First Successful Ground Landing on 4th June, 2010.

```
: %sql select min(DATE) from SPACEXTBL
   * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb
  Done.
1]:          1
     2010-06-04
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters are F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2 which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
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

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databas
Done.
```

]:

missionoutcomes
1
99
1

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

boosterversion
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

```
%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE Landing__Outcome = 'Failure (drone shi
p)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ddc07064:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud
Done.
```

]:

1	mission_outcome	booster_version	launch_site
6	Success	F9 v1.0 B0003	CCAFS LC-40
12	Success	F9 v1.0 B0004	CCAFS LC-40
5	Success	F9 v1.0 B0005	CCAFS LC-40
10	Success	F9 v1.0 B0006	CCAFS LC-40
3	Success	F9 v1.0 B0007	CCAFS LC-40
9	Success	F9 v1.1 B1003	VAFB SLC-4E
12	Success	F9 v1.1	CCAFS LC-40
1	Success	F9 v1.1	CCAFS LC-40
4	Success	F9 v1.1	CCAFS LC-40
7	Success	F9 v1.1	CCAFS LC-40

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

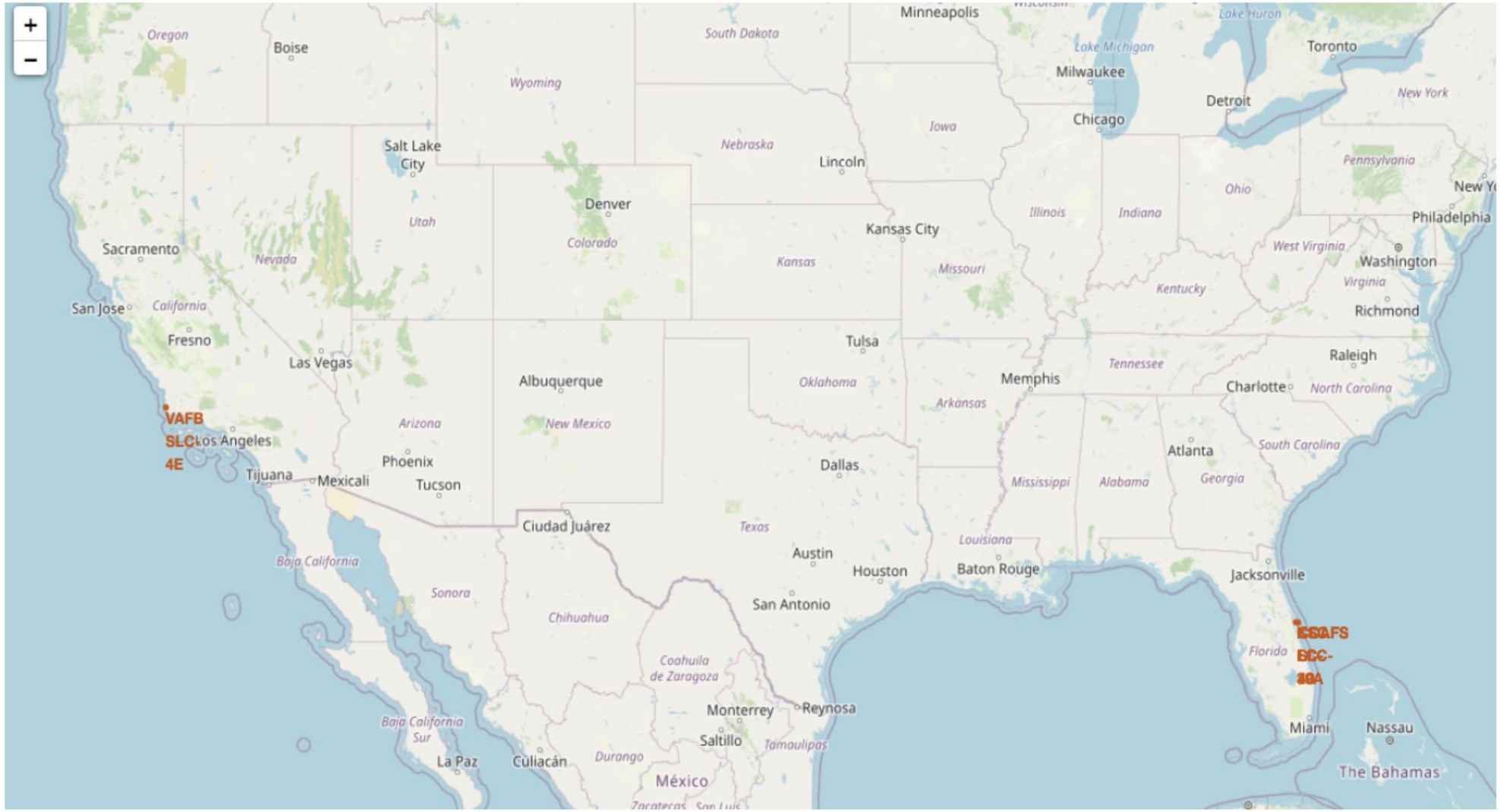
- sql SELECT LANDING__OUTCOME
- FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC

landing__outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
No attempt
Failure (drone ship)
No attempt
Controlled (ocean)
Failure (drone ship)
Uncontrolled (ocean)
No attempt
No attempt
Controlled (ocean)
Controlled (ocean)
No attempt
No attempt
Uncontrolled (ocean)
No attempt
No attempt
No attempt
Failure (parachute)
Failure (parachute)

Section 4

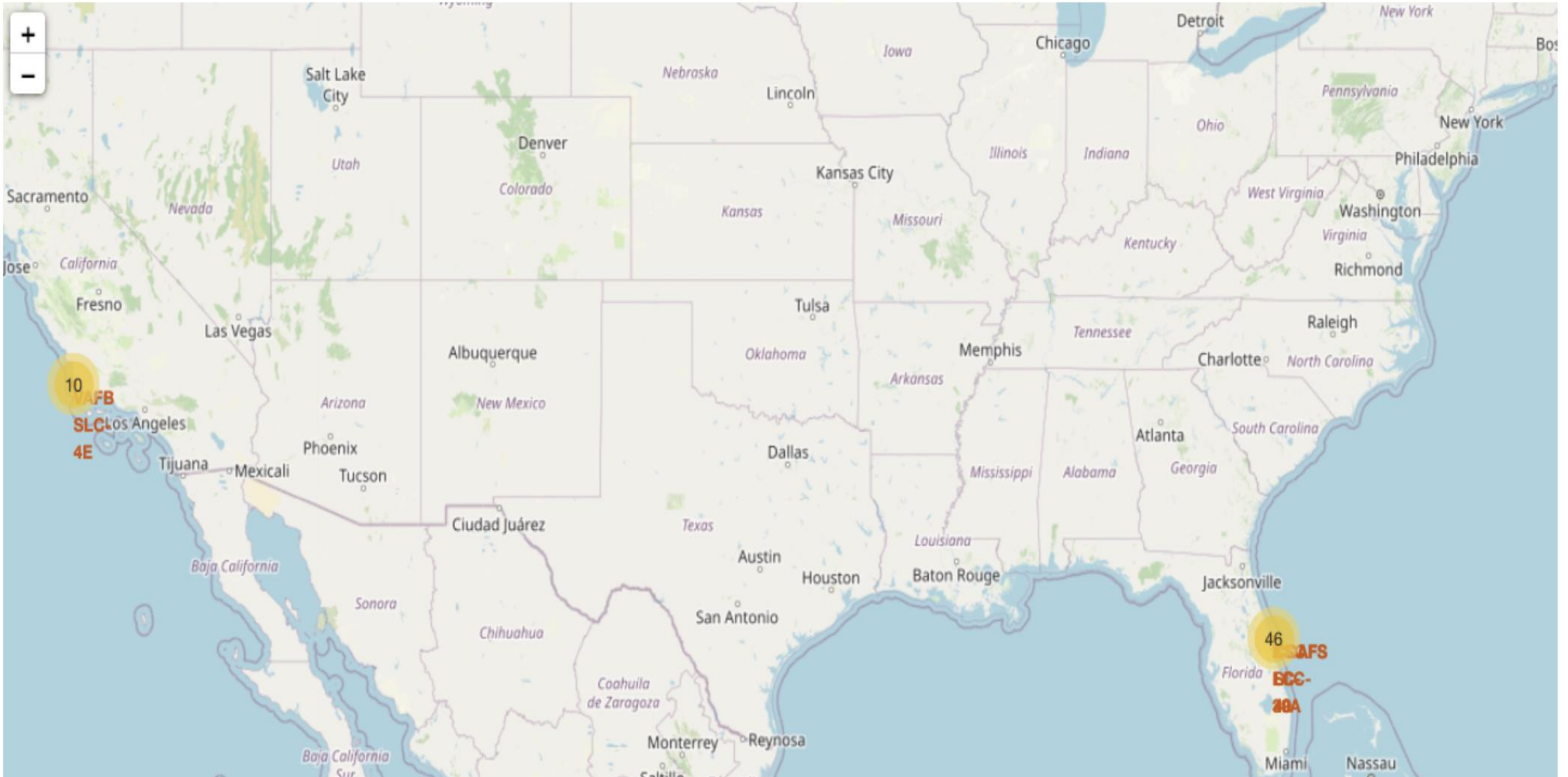
Launch Sites Proximities Analysis

All launch sites on a map

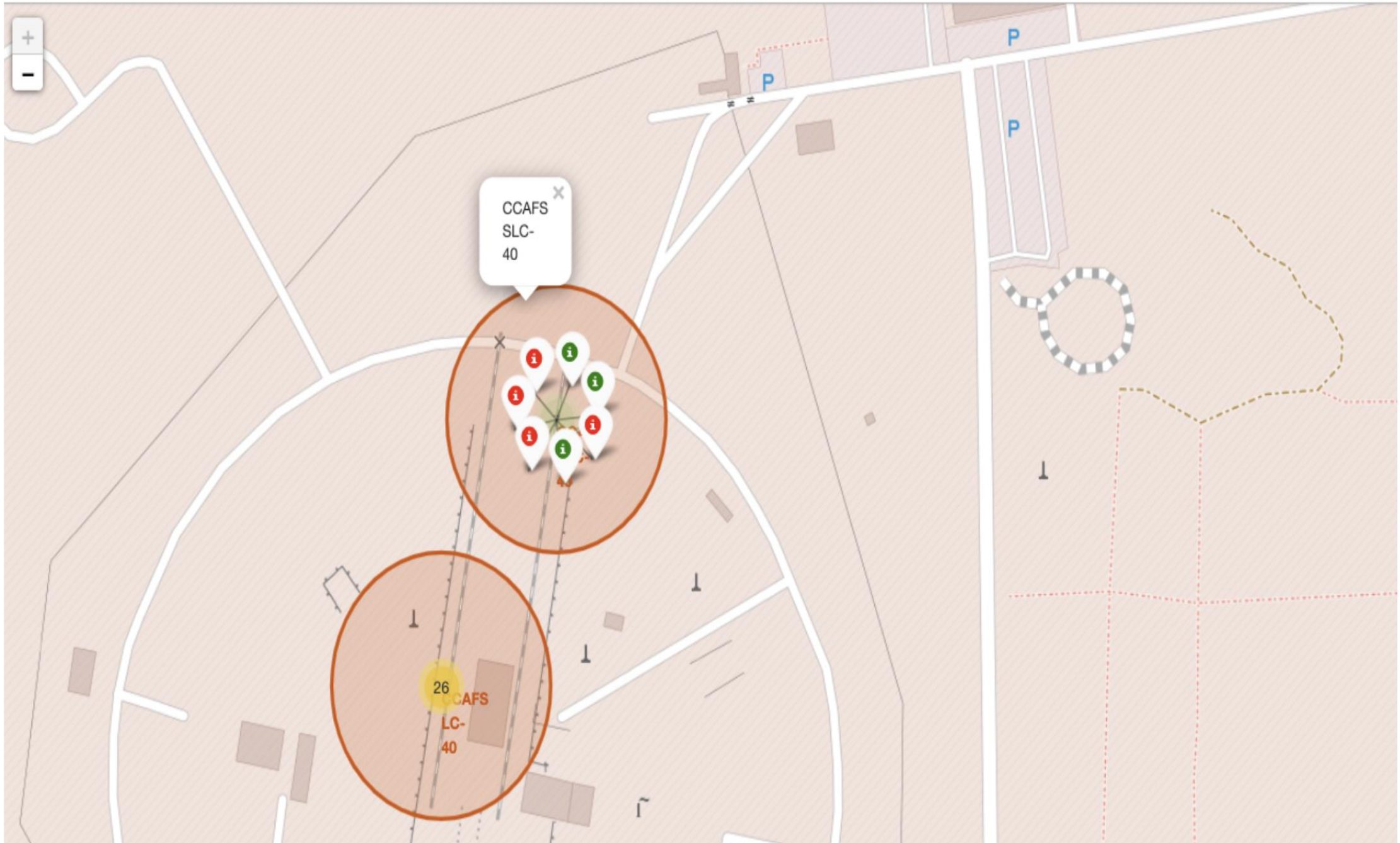


It seems all launch sites in proximity to the Equator line and very close proximity to the coast

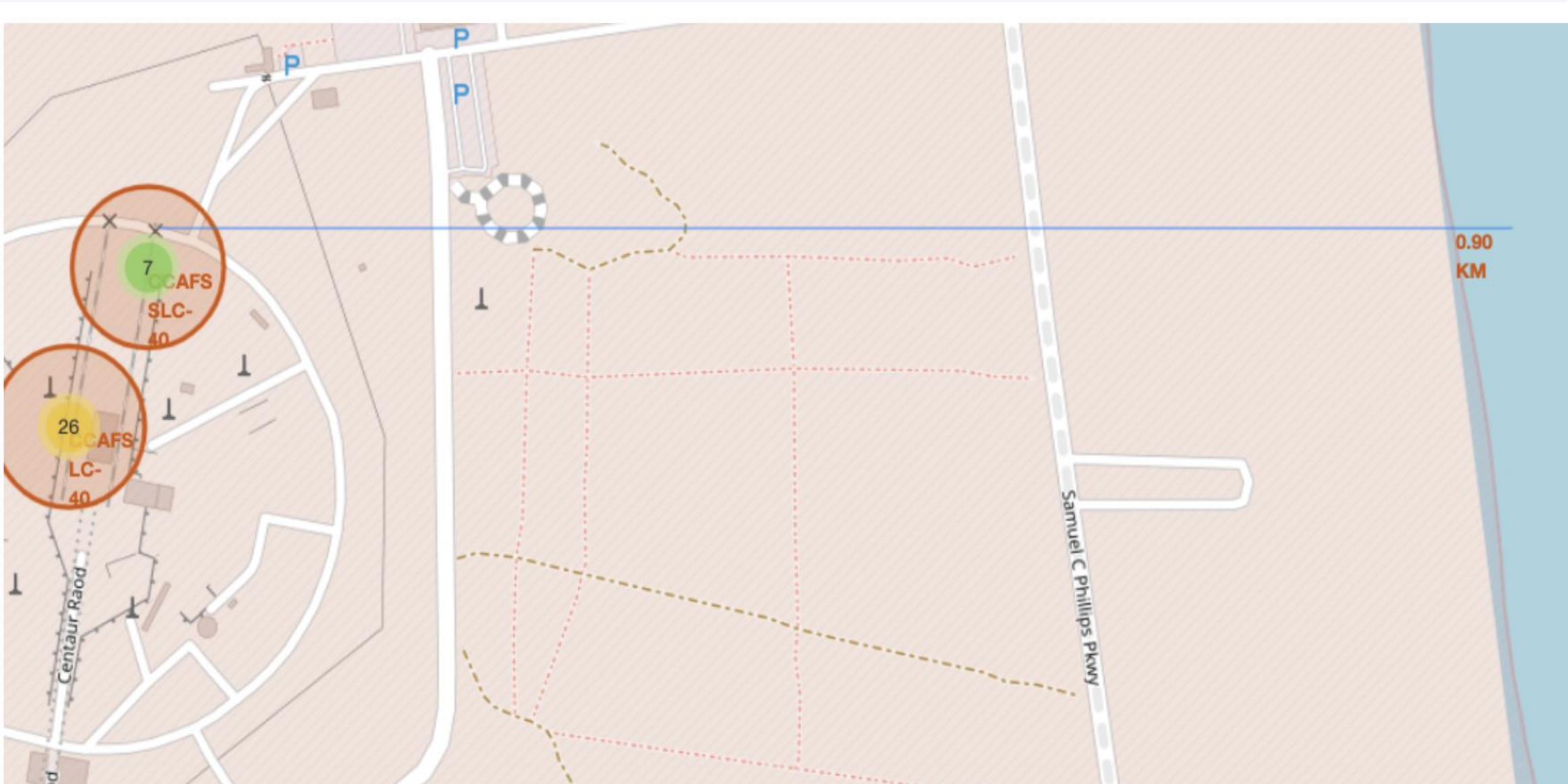
The success/failed launches for each site on the map

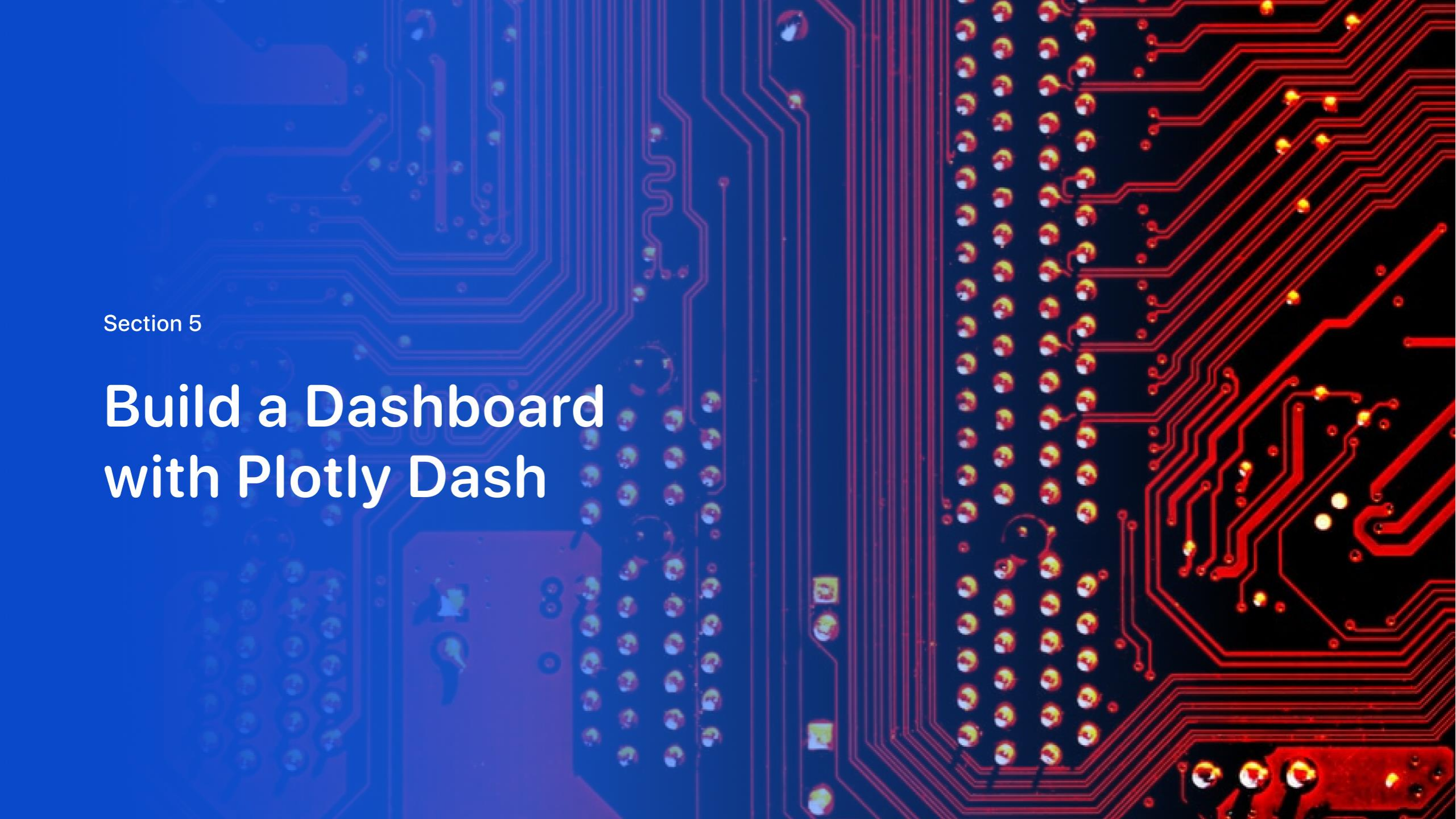


The success/failed launches for each site on the map



The distances between a launch site to its proximities

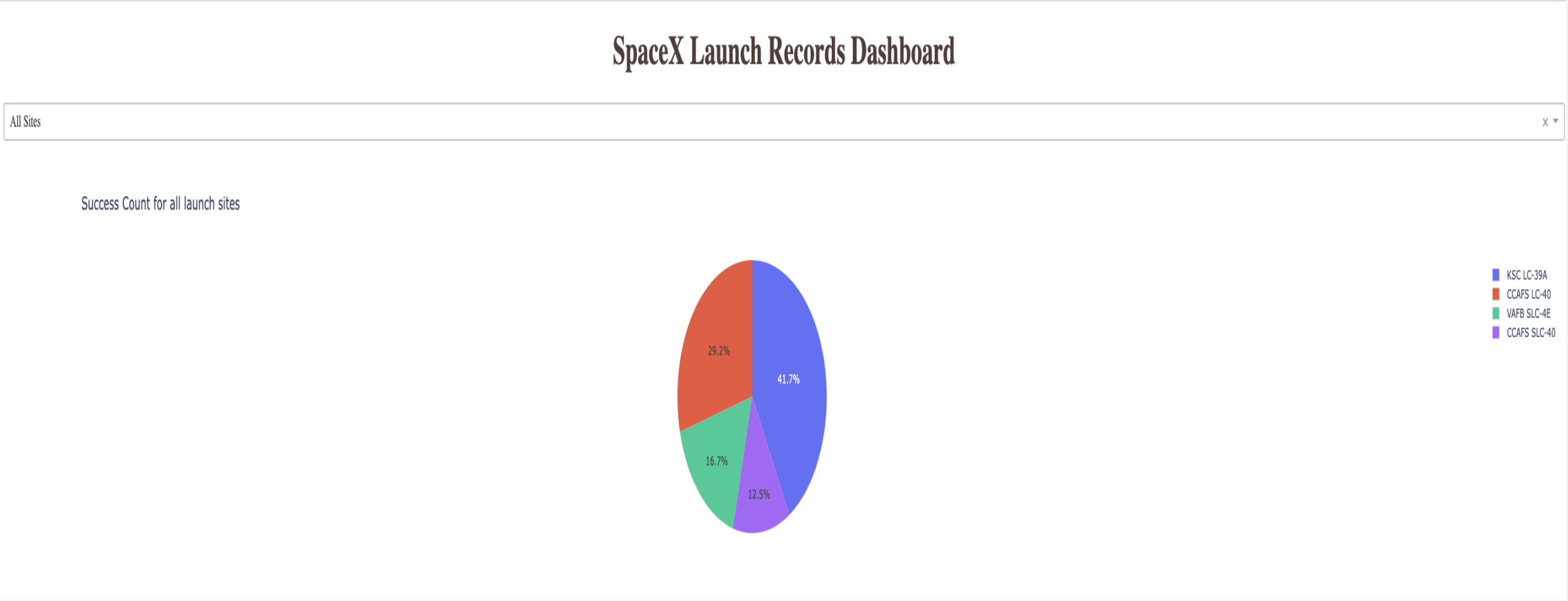




Section 5

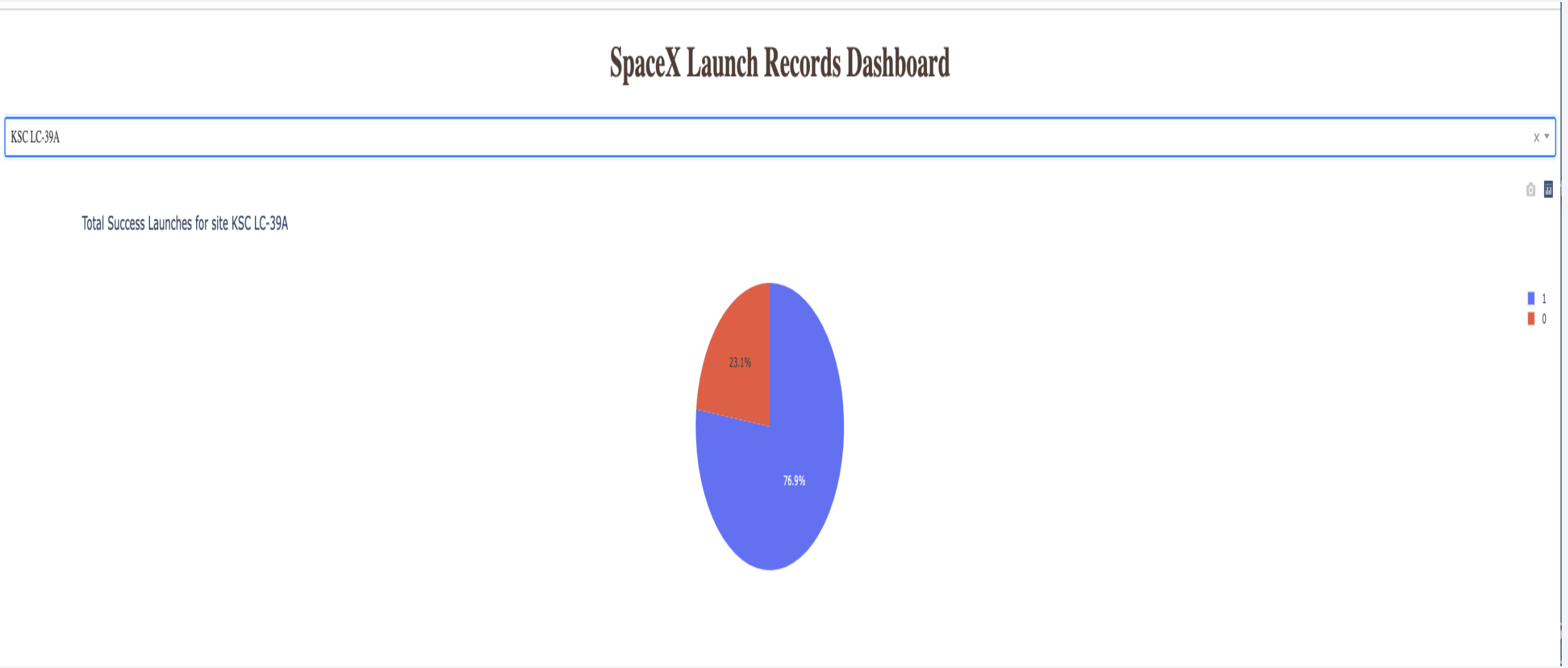
Build a Dashboard with Plotly Dash

Launch success count for all sites in a piechart



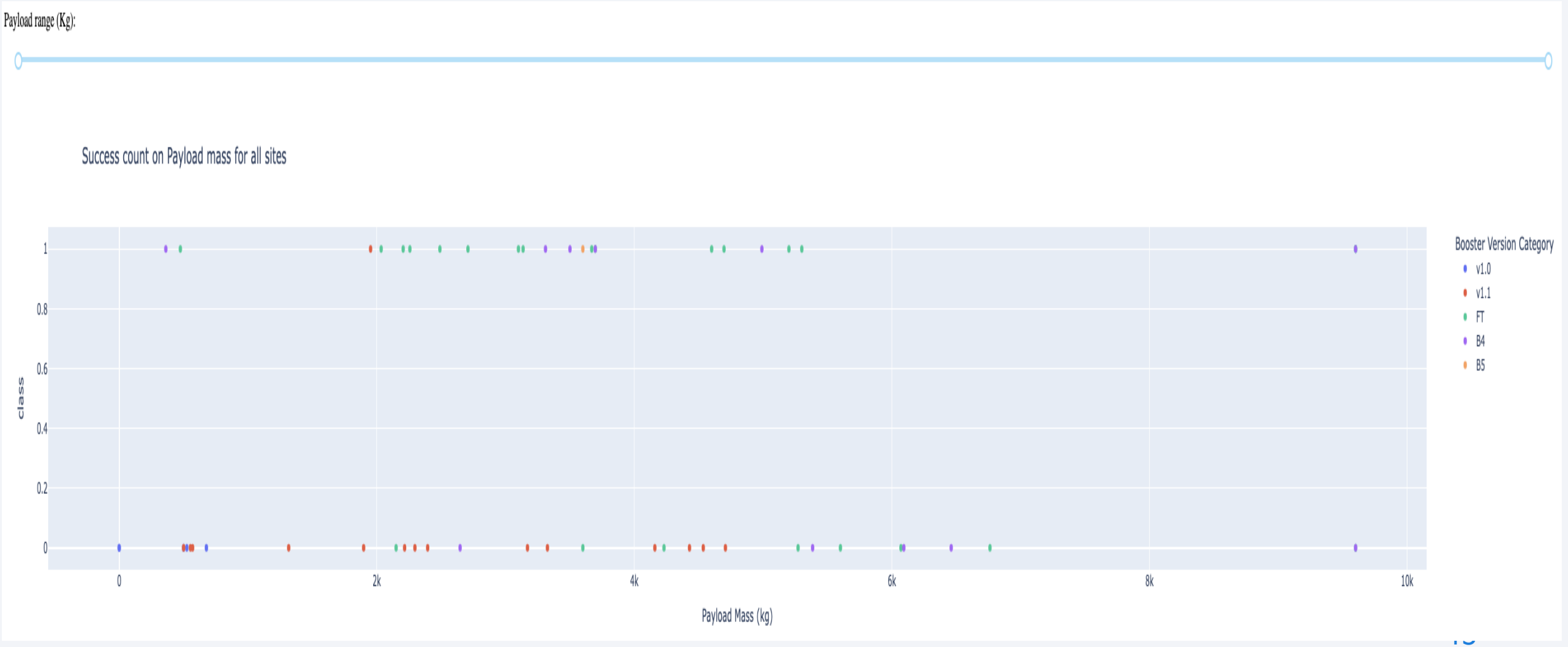
We can see that launch site KSC LC – 39A has the highest success count.

The piechart for the launch site (KSC LC – 39A)



We can see that this site has 76.9 % success rate.

The scatter plot for all site between payload and success rate

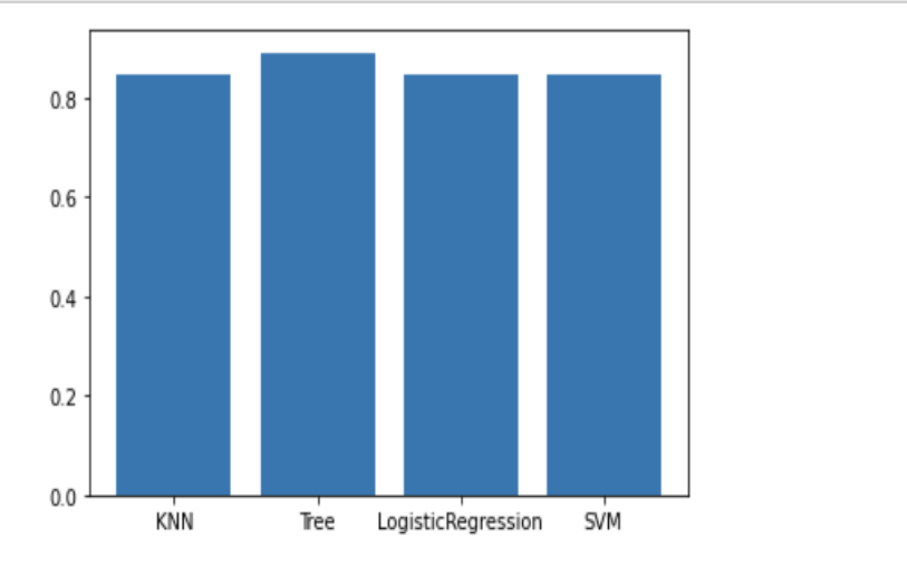


With the payload mass is lower we can see that has higher success rate.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

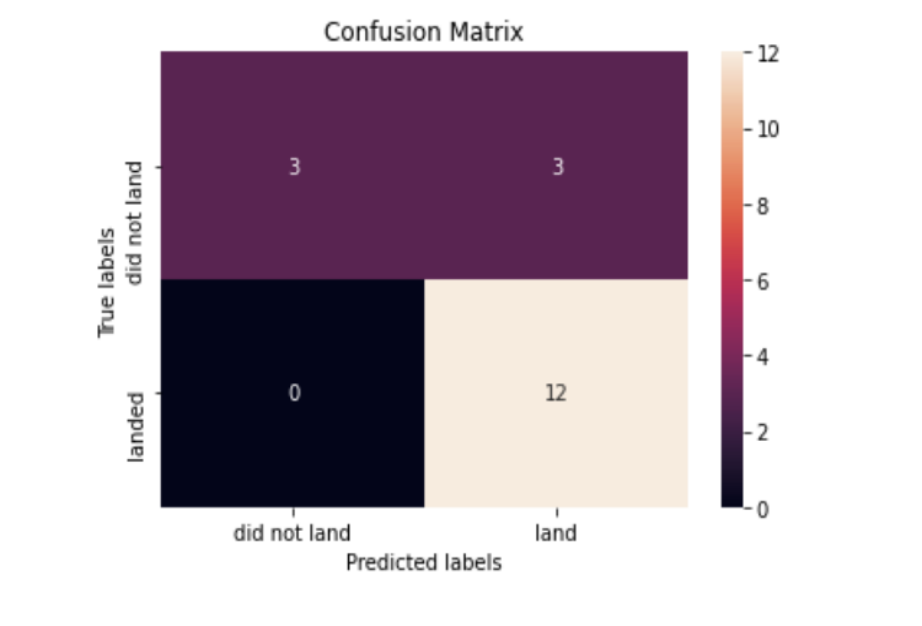


We can see the accuracy scores of models quite closed. But we can find the model Decision Tree with the higher accuracy score.

```
score_method = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_, 'SVM': svm_cv.best_score_}
bestscore = max(score_method, key=score_method.get)
print('Best method is',bestscore,', score is',score_method[bestscore])
```

```
Best method is Tree , score is 0.8910714285714286
```

Confusion Matrix



Conclusions

- The Tree Decision is good method to predict.
- With the payload mass is lower we can see that has higher success rate.
- Launch site KSC LC – 39A has the highest success count.

Thank you!