IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ricardo Rodriguez>
<8/11/2021>

Outline

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

Executive Summary

- This project is composed of different methods in data science to describe and investigate the success and failure of possible launches of a new space company using public data from SPACEX where we have the characteristics of the payload, orbit and launch site.

- The data was collected by connecting Python to the SPACEX API and performing WEBSCRAPPING using beautiful soup methods in Python. This was followed by a DATA WRANGLING process to classify and transform the successful and unsuccessful launch data.

- Data exploration methods were used, such as descriptive statistics of the variables and comparison of variables to observe which were the most optimal. Finally, graphs were created, and MACHINE LEARNING methods were used to select the best method according to the data used.

- Having as results the characteristics of which launch sites are more adequate and what size the payload should be, and which orbits are the most successful, with the technology of each booster.

Introduction

- The private space race has several companies competing for different objectives, but costs are an issue that began to have relevance where SPACEX implemented the first phase of its rockets to be reusable and is currently trying to develop a fully reusable rocket, with which they have I degree reduce costs from 165 million dollars, losing the rocket in its entirety to an approximate value of 62 million dollars reusing the first phase of the rocket, if a safe landing is achieved.
- In wanting to compete SPACEY this hypothetical company analyzes what factors influence the first stage of the rocket to land safely or successfully, where SPACEX historical data variables are compared and applies several data science methods and MACHINE LEARNING to predict which factors or values of the variables and their combinations are the most favorable to have the most launches and safe landings. This presentation will explore the variables and methods used.

Section 1

Methodology

Methodology

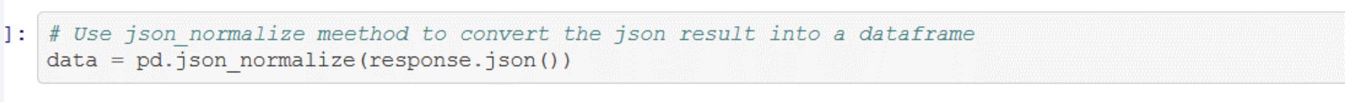
Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

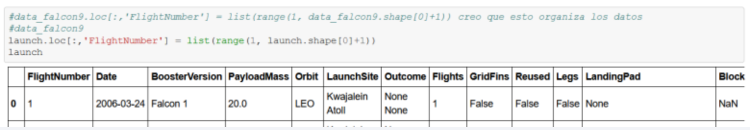
- Getting SPACEX data from its API in json format

- Saving data in tables using normalization method



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

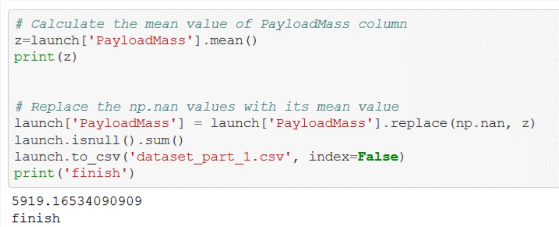
- Reviewing tables and filtering tables



```
#data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1)) creo que esto organiza los datos
#data_falcon9
launch.loc[:,'FlightNumber'] = list(range(1, launch.shape[0]+1))
launch
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN

- Data cleansing

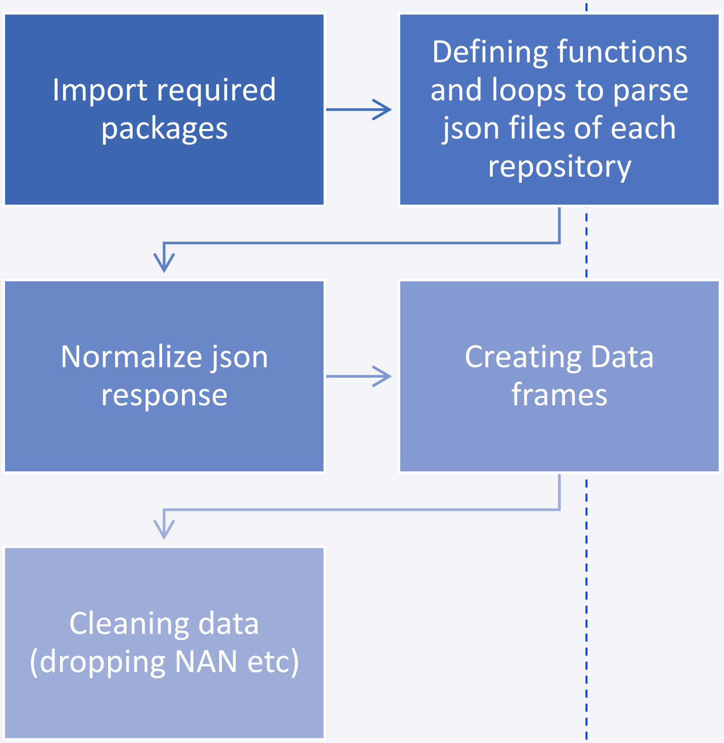


```
# Calculate the mean value of PayloadMass column
z=launch['PayloadMass'].mean()
print(z)


# Replace the np.nan values with its mean value
launch['PayloadMass'] = launch['PayloadMass'].replace(np.nan, z)
launch.isnull().sum()
launch.to_csv('dataset_part_1.csv', index=False)
print('finish')
```

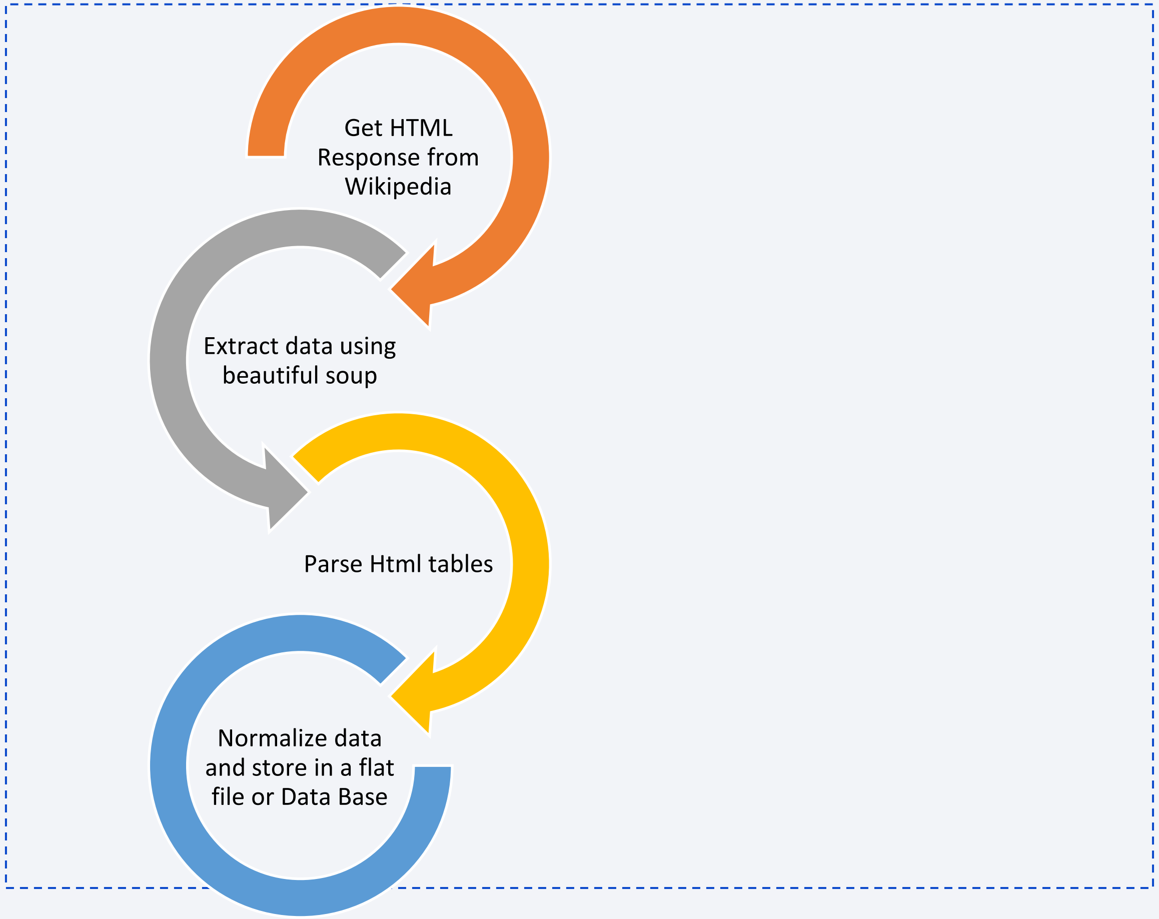
```
5919.16534090909
finish
```

- [Data Collection]()


Import required packages
Defining functions and loops to parse json files of each repository
Normalize json response
Creating Data frames
Cleaning data (dropping NAN etc)

Data Collection - Scraping

- Installing and importing relevant packages
- Defining global variables and element extraction functions
- Defining data from wikipedia
- Extracting tables by searching for specific tags
- Creating data frame using html tables



Get HTML Response from Wikipedia

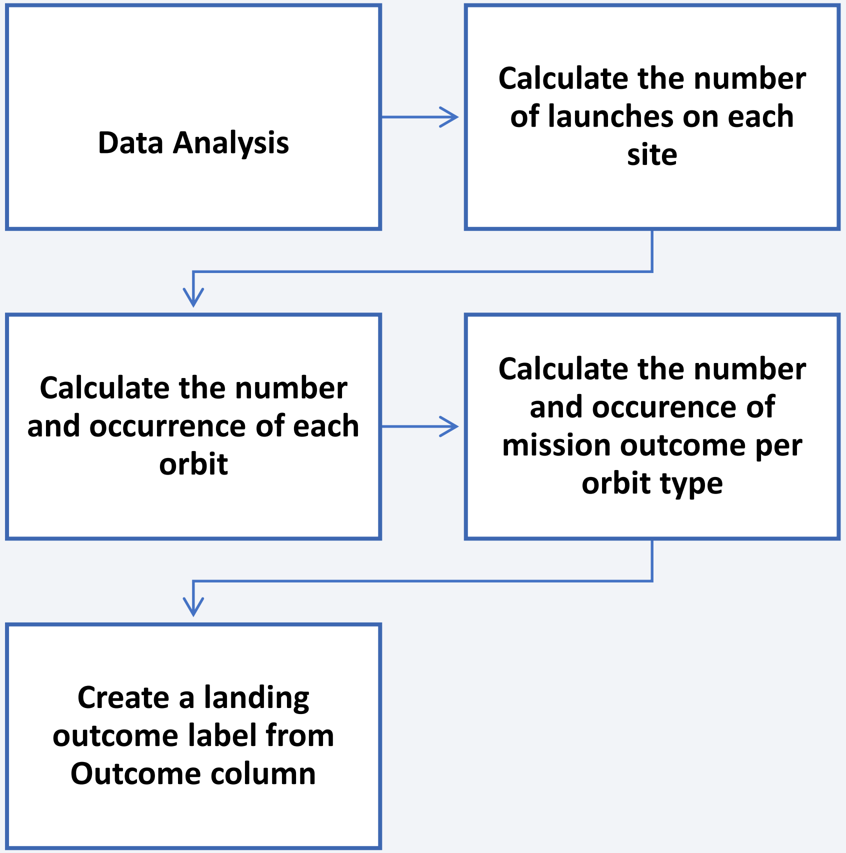
Extract data using beautiful soup

Parse Html tables

Normalize data and store in a flat file or Data Base

Data Wrangling

- from 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. We mainly convert those outcomes into Training Labels with 1 value means the booster successfully landed 0 value means it was unsuccessful



Data Analysis

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

- GitHub URL

EDA with Data Visualization

- Scatter graphs or dots:
- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit
- Orbit VS. Payload Mass
- Mean VS. Orbit(bar plot)
- Success Rate VS. Year (line plot)

The scatter plots shows the correlation between two variables

The bar plot shows the relationship between multiple variables independents vs a depend variable

The line plot is good to show time series or temporal data

GitHub URL

EDA with SQL

SQL queries performed:

- 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
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- GitHub URL

Build an Interactive Map with Folium

Interactive map using folium methods.

- Create a list of Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.
- Assigned the dataframe launch_outcomes(failures, successes) to Classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()
- GitHub URL

Build a Dashboard with Plotly Dash

- **Graphs and plots**

Pie Chart showing the total launches by a certain site/all sites

display relative proportions of binary classes of data of success or failure filtered by site.

Scatter Graph showing the relationship with Outcome and Payload Mass in Kg for the different Booster filtered by site and a step slider for every 2000 kg

GitHub URL

Predictive Analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCVobjects and train our dataset.

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix for each model

FINDING THE BEST PERFORMING MODEL

- The model with the best accuracy score wins the best model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook and a bar plot. GitHub URL

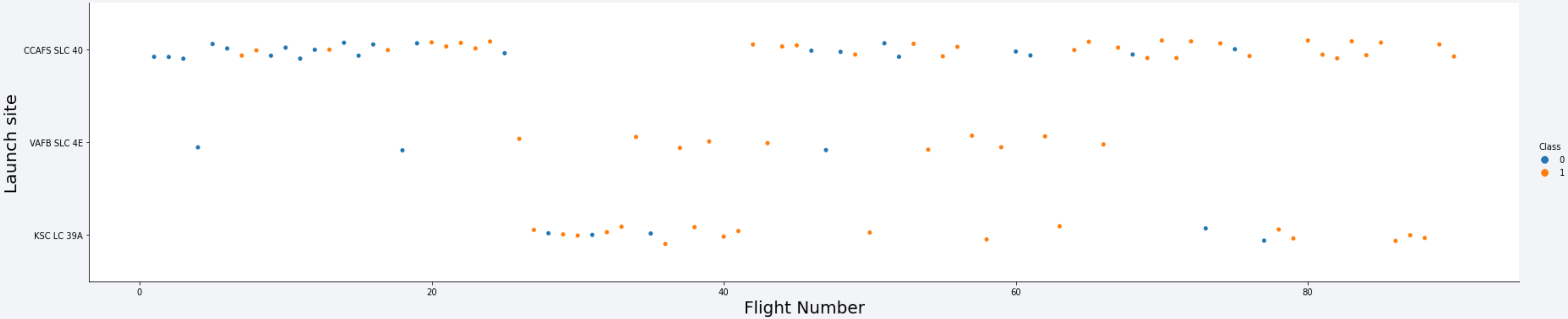
Results

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

Section 2

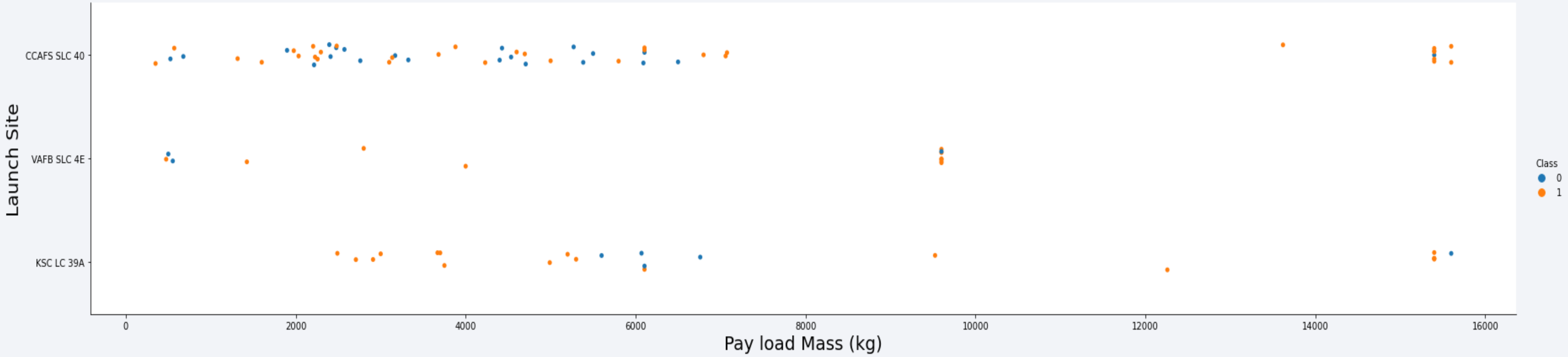
Insights drawn from EDA

Flight Number vs. Launch Site



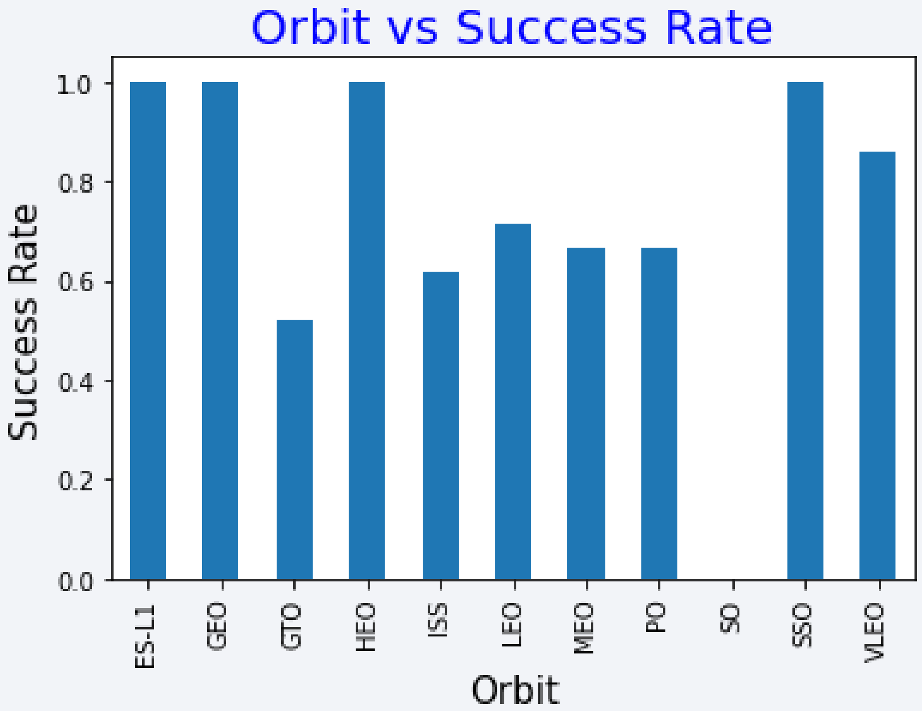
- The launch site ‘CCAFS SLS 40’ has the highest number of unsuccessful launches but is the most widely used site and the site ‘VAFB SLS 4E’ has the highest number of successful launches in proportion to the number of launches made but is the site with the fewest launches of the three sites.

Payload vs. Launch Site



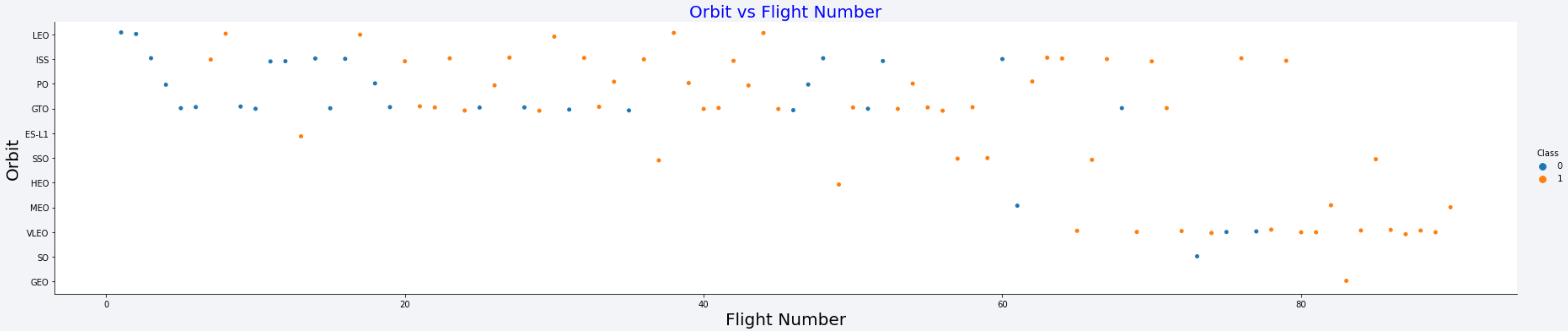
- Most launches are carried out with less than 8,000 kilos of payload, but these throws concentrate most of the failures and there are very few throws up of 8000 kilos and these in their proportion are more successful, more data is needed to define if the weight influences the failures.

Success Rate vs. Orbit Type



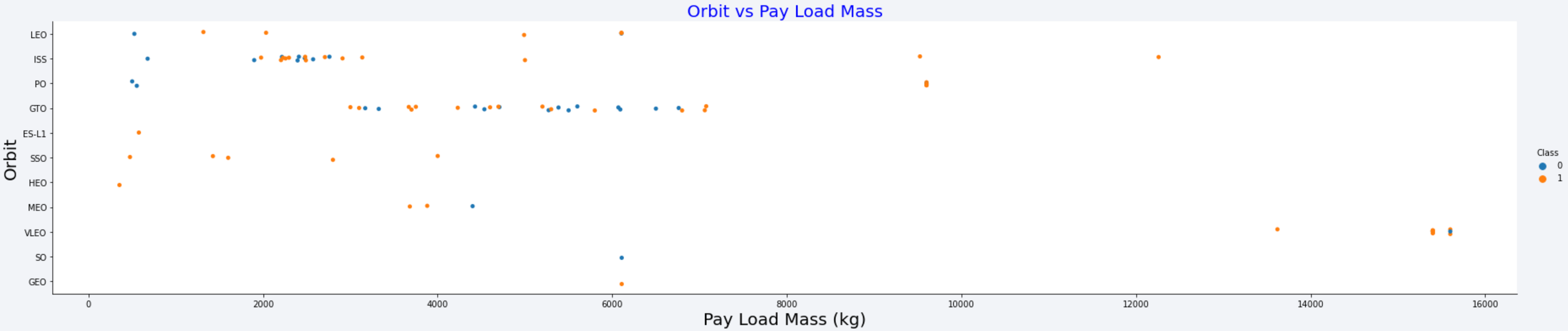
- The most successful orbits are “ES-L1, GEO, HEO and SSO” , but the least successful orbits are “GTO, ISS and SO”, but the least successful orbits in turn have the highest launches

Flight Number vs. Orbit Type



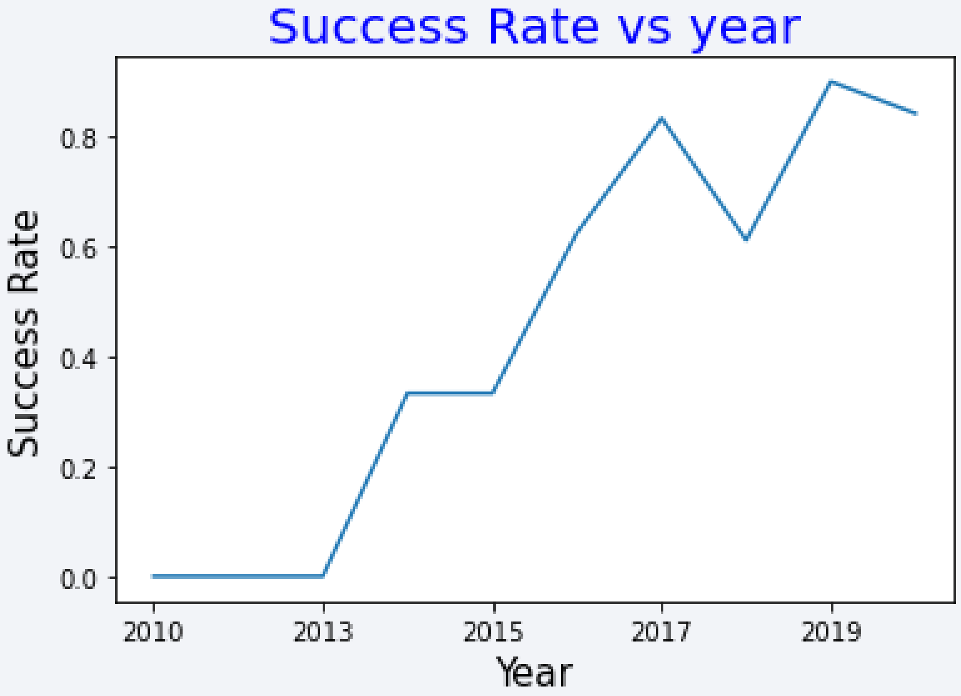
- The most successful orbit according to the latest launches is ‘VLEO’, followed by the orbit ‘ISS’ showing a correlation of the higher the number of flights, the higher the success rate, probably due to the correction of errors in the technology used.

Payload vs. Orbit Type



- By payload, the orbits “ISS and GTO” seem to be more successful, but at the same time they concentrate more failures.

Launch Success Yearly Trend



- As seen in the graph, the year 2013 shows a change in the trend of mission success but there is a drop in 2018, which is corrected in 2019.

All Launch Site Names

- SQL QUERY:

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXDATASET;
```

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

- Using the distinct function you can obtain a list excluding duplicate sites.

Launch Site Names Begin with 'CCA'

- SQL QUERY:

```
SELECT *

FROM SPACEXDATASET

WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Using the function like, 'CCA%' the request is obtained but I limit the table to 5 results to get the first 5 lines

Total Payload Mass

- SQL QUERY:

```
SELECT CUSTOMER, SUM(PAYLOAD_MASS__KG_)

FROM SPACEXDATASET

WHERE CUSTOMER ='NASA (CRS)'

GROUP BY CUSTOMER;
```

customer	2
NASA (CRS)	45596

- Using the sum function, grouping by the 'CUSTUMER' field and conditioning with the text 'NASA (CRS)'.

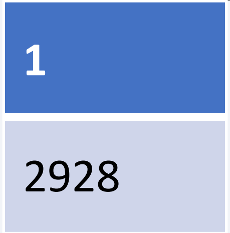
Average Payload Mass by F9 v1.1

- SQL QUERY:

```
SELECT AVG(PAYLOAD_MASS__KG_)

FROM SPACEXDATASET

WHERE BOOSTER_VERSION ='F9 v1.1’ ;
```



1
2928

- The average payload is calculated and filtered with 'F9 v1.1’using the field booster version

First Successful Ground Landing Date

- SQL QUERY:

```
SELECT MIN(DATE)
FROM SPACEXDATASET
WHERE LANDING__OUTCOME='Success (ground pad)';
```

1
2015-12-22

- To find the first date use the suggested function 'MIN' filtering the date with the field 'landing outcome' and make it 'successful '.

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL QUERY:

```
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_

FROM SPACEXDATASET

WHERE LANDING__OUTCOME='Success (drone ship)' AND (PAYLOAD_MASS__KG_
BETWEEN 4000 AND 6000) ;
```

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

- In the condition Y is used to get the result, among the desired values

Total Number of Successful and Failure Mission Outcomes

- SQL QUERY:

```
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME)as COUNTER

FROM SPACEXDATASET

GROUP BY MISSION_OUTCOME;
```

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

- to calculate the total number of successful and unsuccessful missions, the 'COUNT' function was used, and the results were grouped with the field MISSION_OUTCOME', to have the value of success and failure.

Boosters Carried Maximum Payload

- SQL QUERY:

```
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_

FROM SPACEXDATASET

WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET );
```

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

- with a sub-query you can obtain the maximum load value of each booster

2015 Launch Records

- SQL QUERY:

```
SELECT BOOSTER_VERSION, LANDING__OUTCOME, LAUNCH_SITE
FROM SPACEXDATASET
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND (DATE BETWEEN '2015-01-01' AND '2015-
12-31’);
```

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

- Using a combination of conditions with AND where it is limited to the bugs and dates requested

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

- SQL QUERY:

```
SELECT LANDING__OUTCOME, DATE

FROM SPACEXDATASET

WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20')

ORDER BY DATE DESC;
```

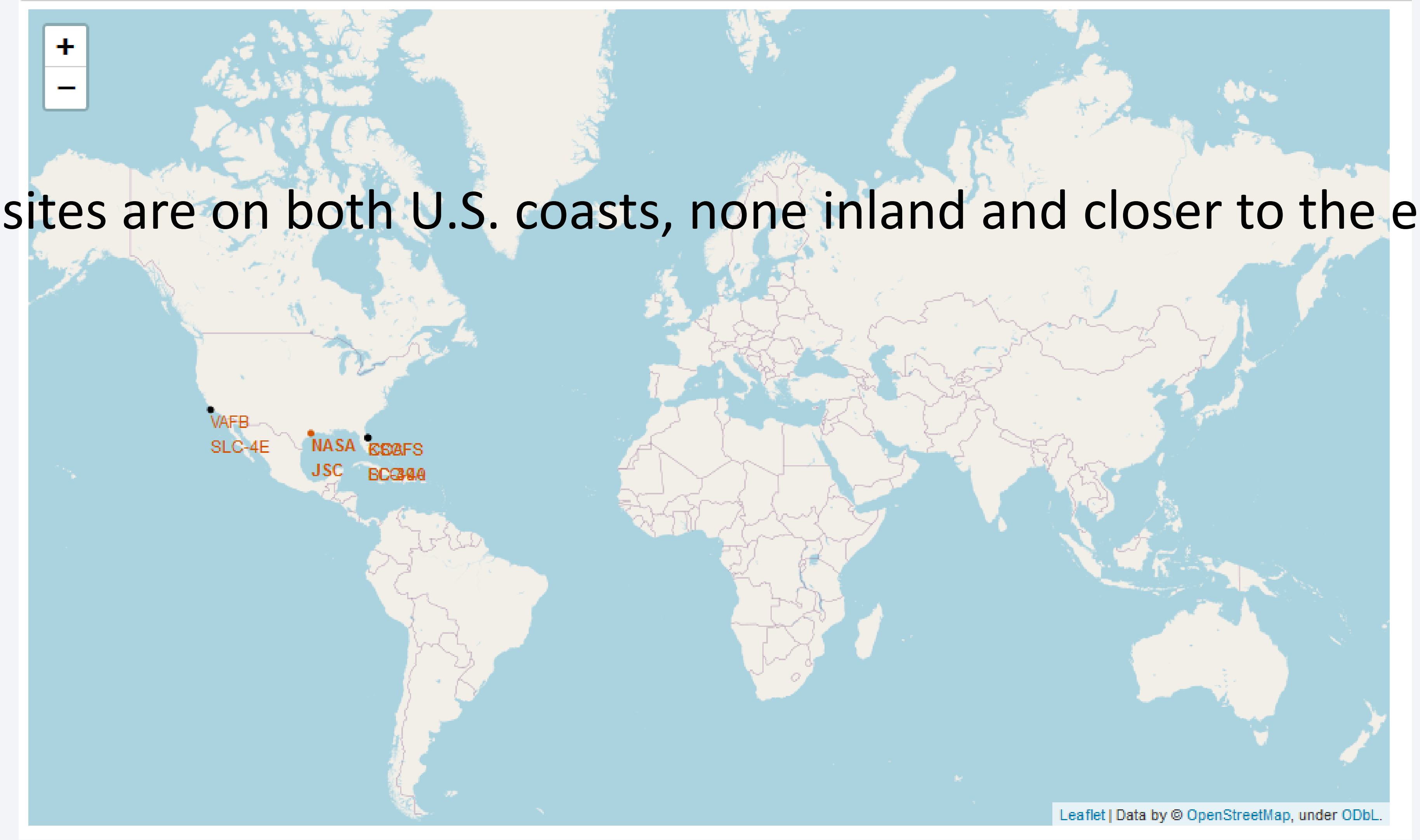
landing__outcome	DATE
No attempt	2017-03-16
Success (ground pad)	2017-02-19
Success (drone ship)	2017-01-14
Success (drone ship)	2016-08-1
……	……….
Failure (parachute)	2010-06-04

- Present your query result with a short explanation here

Section 4

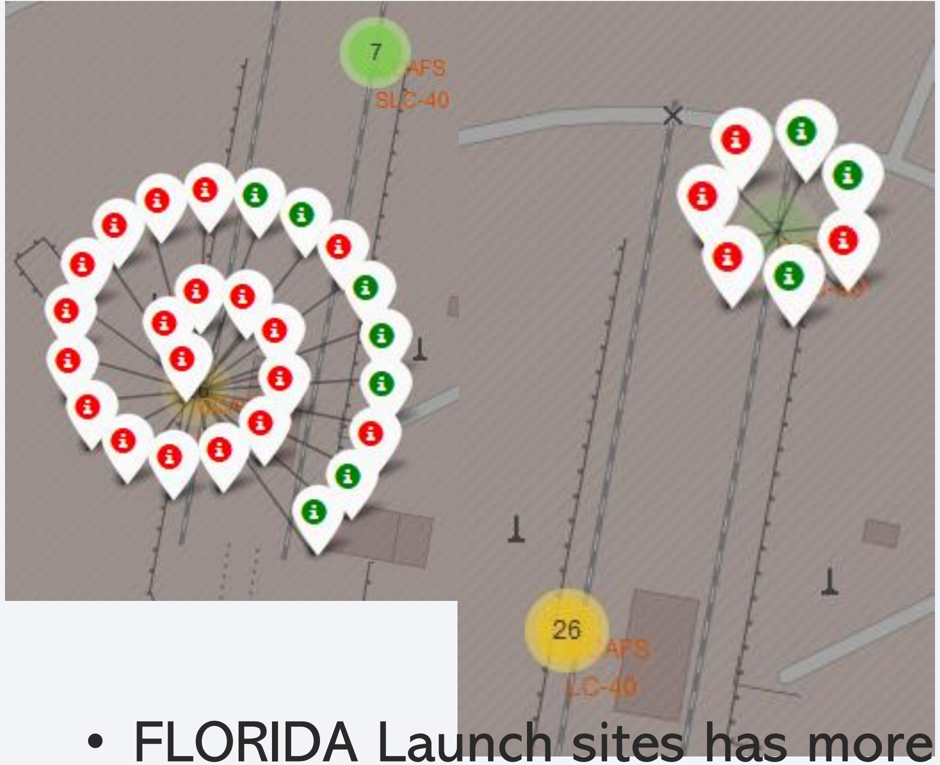
Launch Sites Proximities Analysis

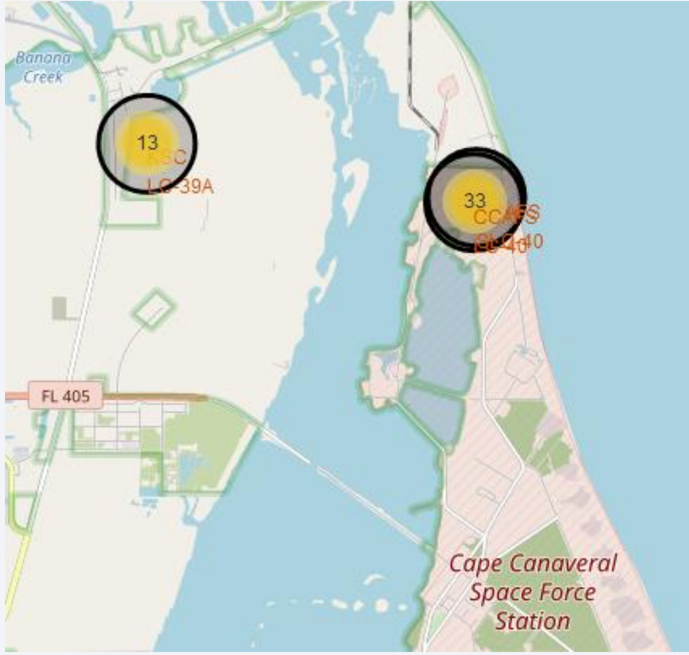
Global map with NASA and SPACEX launch sites

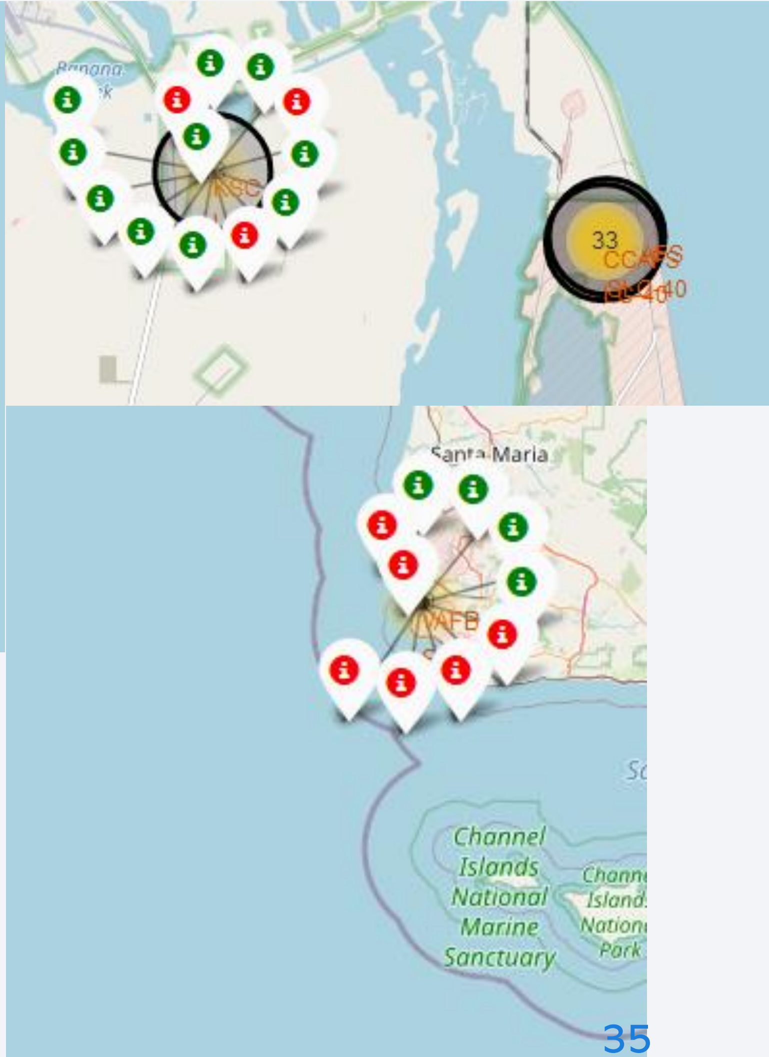


- launch sites are on both U.S. coasts, none inland and closer to the equator.

Color markers at different launch sites

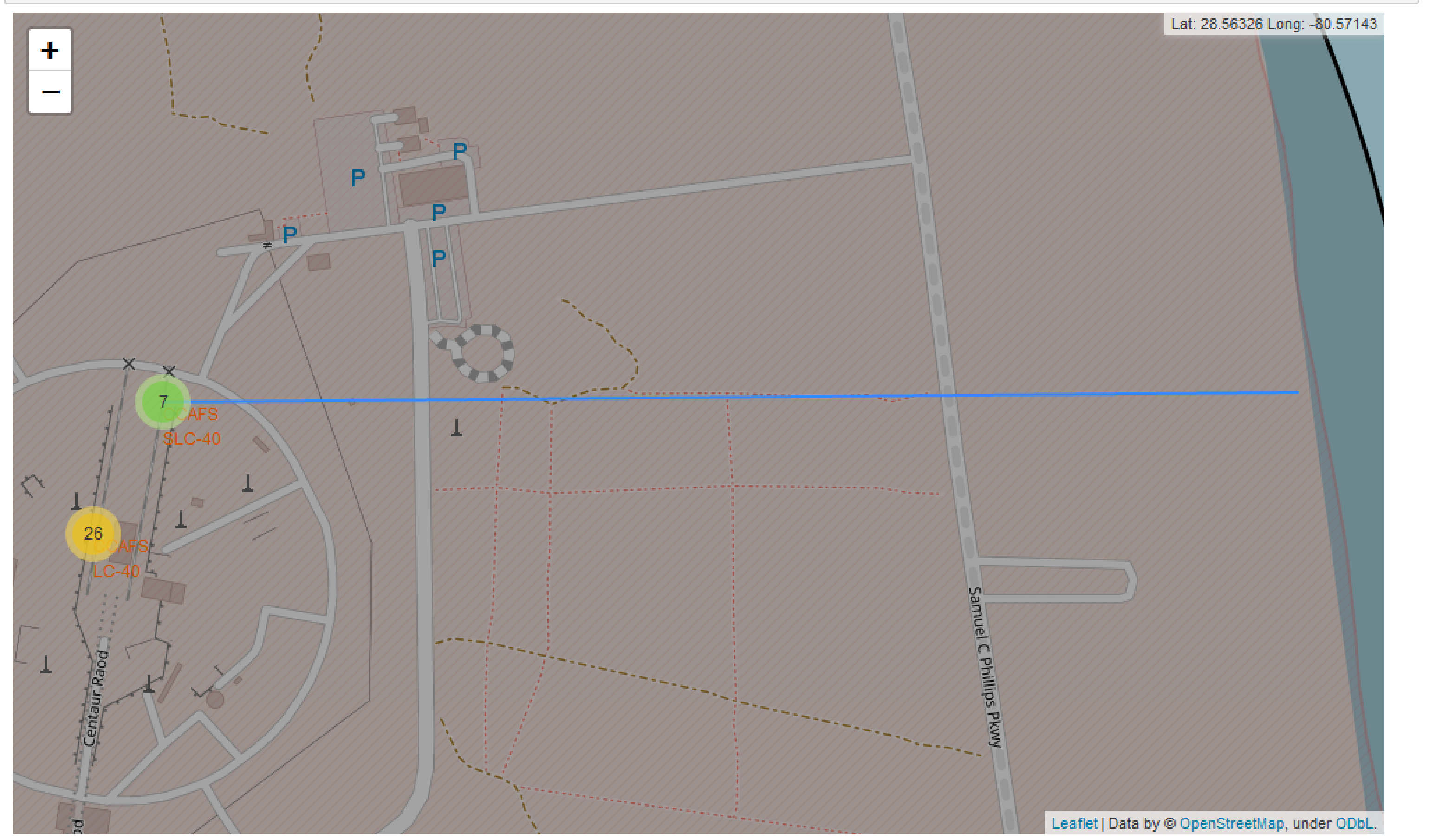


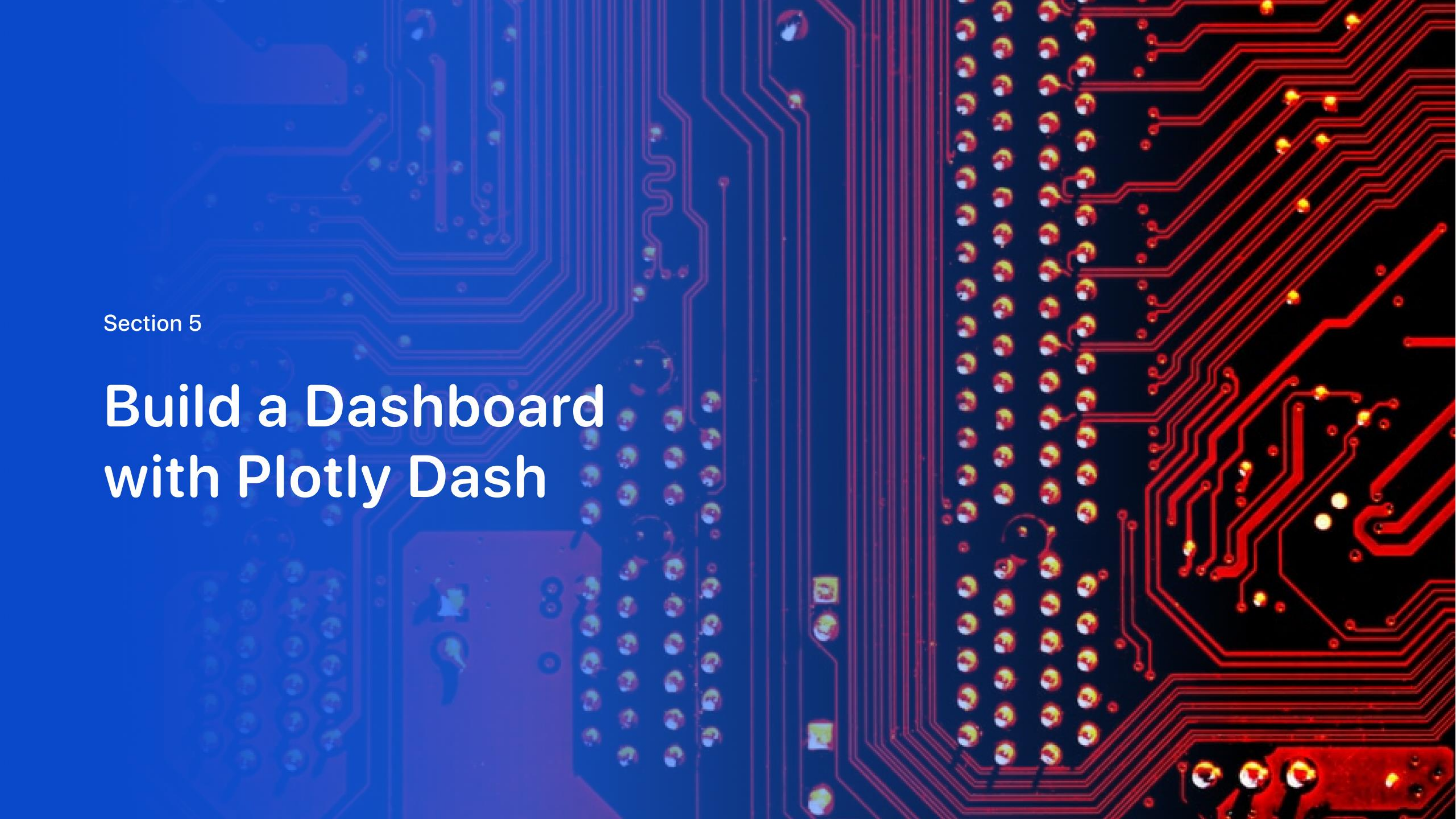



Banana Creek
KSC
33
CCAFS
SLC-40
Santa Maria
VAFB
Channel Islands National Marine Sanctuary
Channel Islands National Park

- FLORIDA Launch sites has more successful launches against California site

Map with polyline from launch site

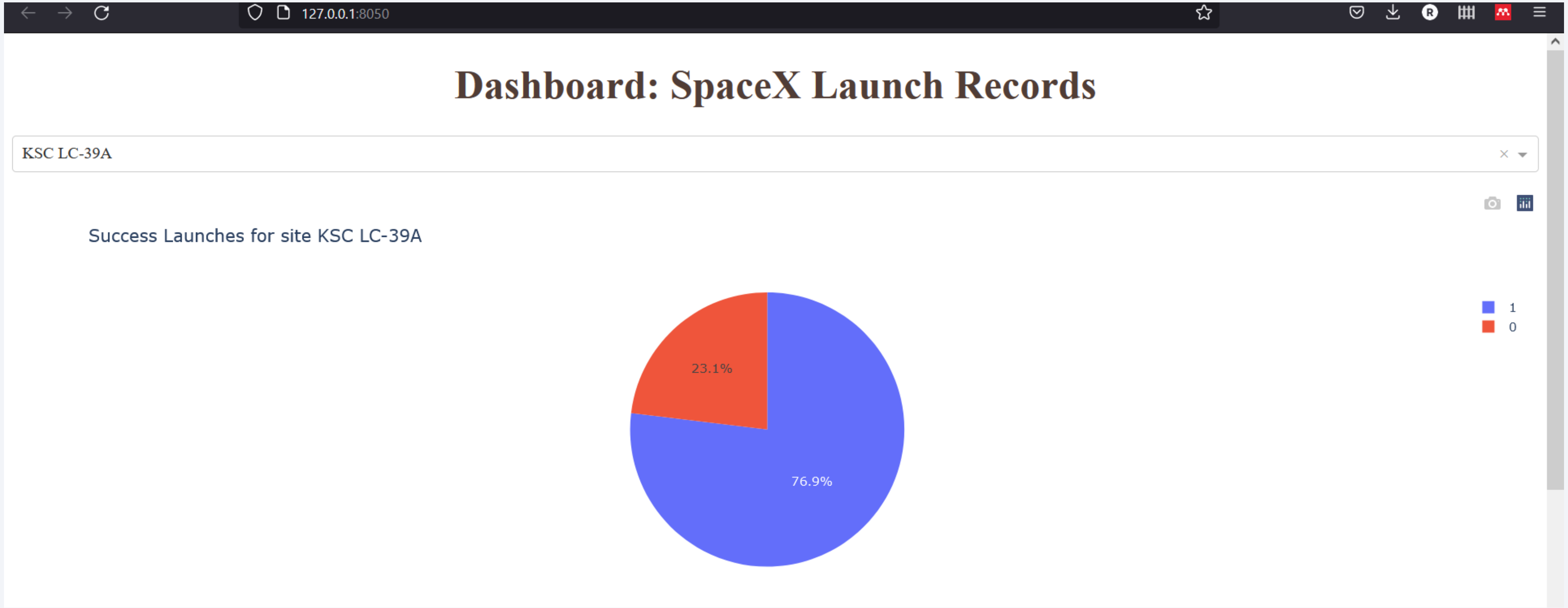

+
−
Lat: 28.56326 Long: -80.57143
P
P
P
P
P
7
AFS
SLC-40
26
AFS
LC-40
Centaur Raod
Samuel C Phillips Pkwy
Leaflet | Data by © OpenStreetMap, under ODbL.



Section 5

Build a Dashboard with Plotly Dash

DASHBOARD: Pie chart for the launch site with highest launch success ratio



Dashboard: SpaceX Launch Records

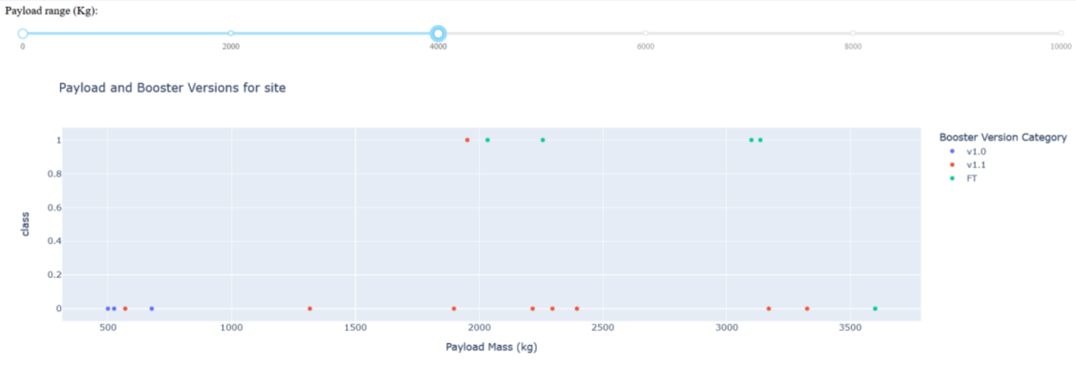
KSC LC-39A

Success Launches for site KSC LC-39A

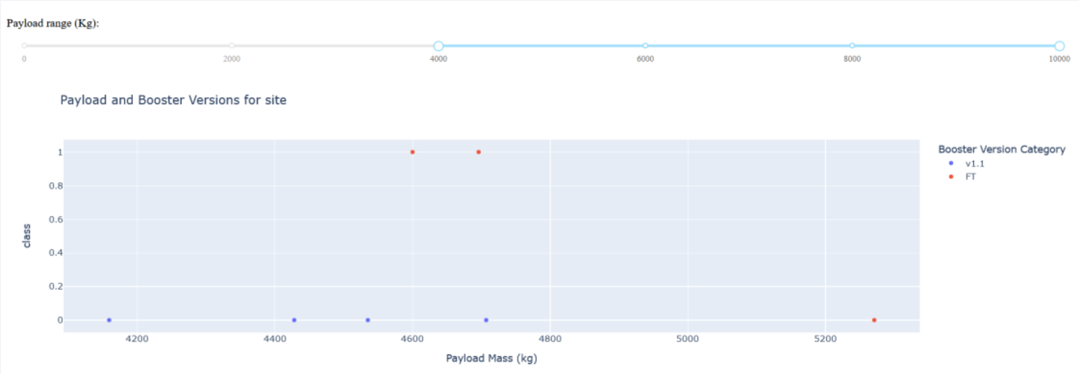
- The site launch KSC LC-39A has the highest launch success ratio with a 76.9% against a 23,1 % of failure ratio

Payload vs. Launch Outcome scatter plot for different sites, with different payload selected in the range slider

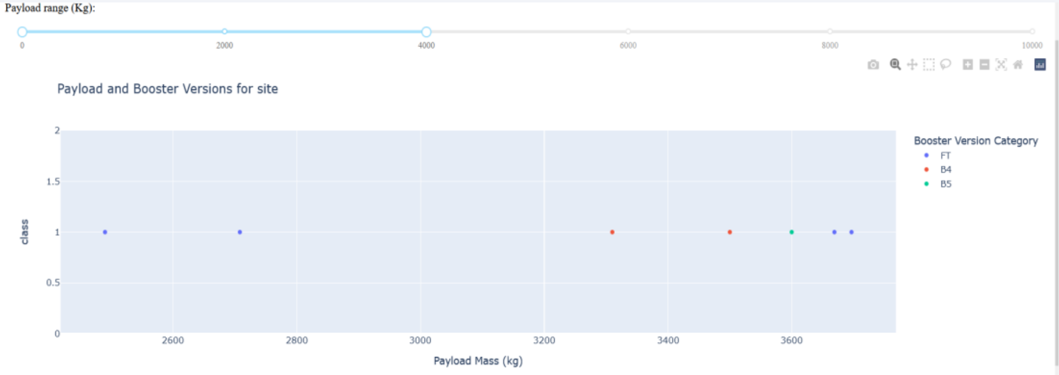
CCAFC LS-40 0 to 4000 kg has mixed outcomes



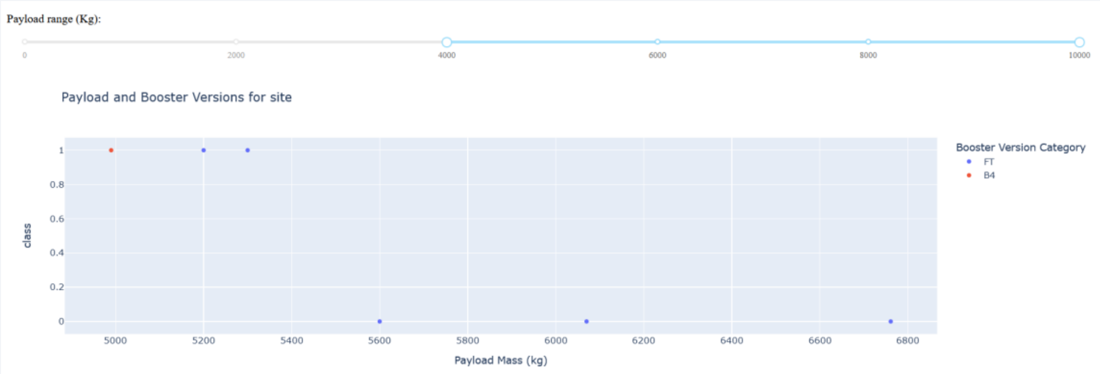
CCAFC LS-40 4000 to 10000 kg has more success outcomes



KSC LC-39A 0 to 4000 kg has mixed outcomes



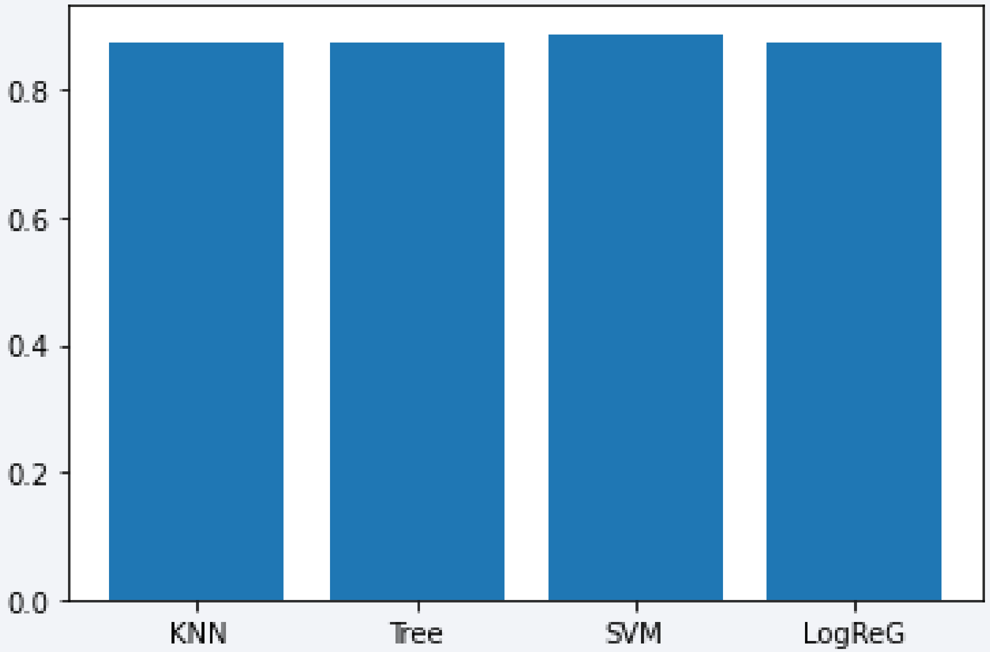
KSC LC-39A 4000 to 10000 kg has more success outcomes



Section 6

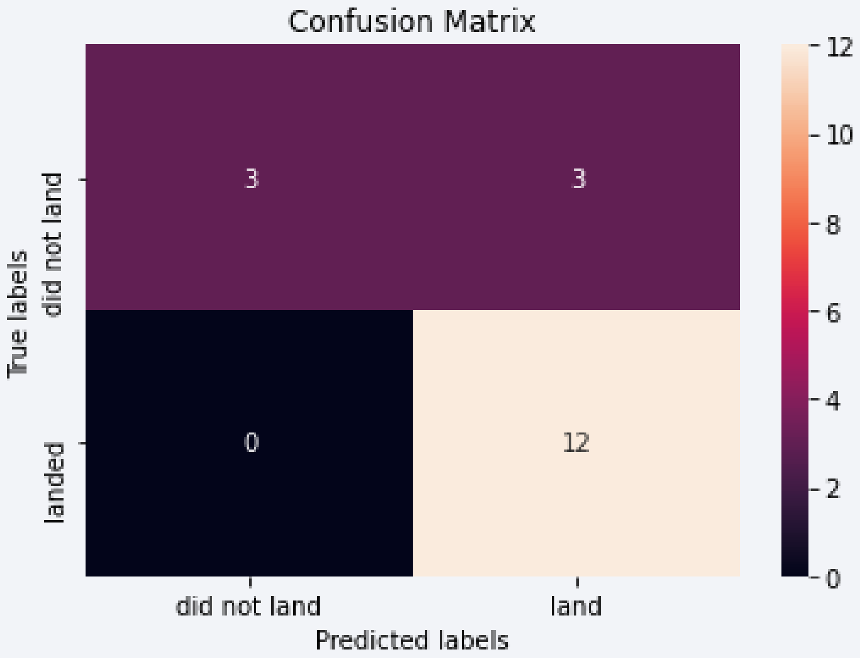
Predictive Analysis (Classification)

Classification Accuracy



SVM model has the highest classification accuracy with a score of 0,88

Confusion Matrix for SVM Model



- In the confusion matrix, we see that SVM can distinguish between the different classes. Majority are true positive values .

Conclusions

- High load launches are more likely to be successful.
- Florida launches also tend to be more successful, perhaps because they are closer to the equator.
- It is difficult to differentiate which model is more accurate since the values are very close to each other, perhaps with more variables we can more clearly differentiate which models are more accurate.
- Data exploration with SQL is a little more effective than with pandas.

Thank you!