Winning Space Race with Data Science

Felipe Alvim
22/12/2022

Outline

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

Executive Summary

- This presentation has the objective to demonstrate the knowledge acquired in the course "DevOps e Engenharia de Software da IBM" throughout the data scrapping and mining of SpaceX. Python code will be used to conduct this job, using machine learning tools and libraries to determine correlations between succeeded rockets launches.

- Results acquired suggest that the outcome of a mission may be predicted by some of the collected data features, such as booster version and payload mass.

Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Given our dataset we want to determine if the first stage of the rocket will successfully land.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
  - WebScrapping on public internet sites (Wikipedia)
- Perform data wrangling
  - Data was cleaned
- 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

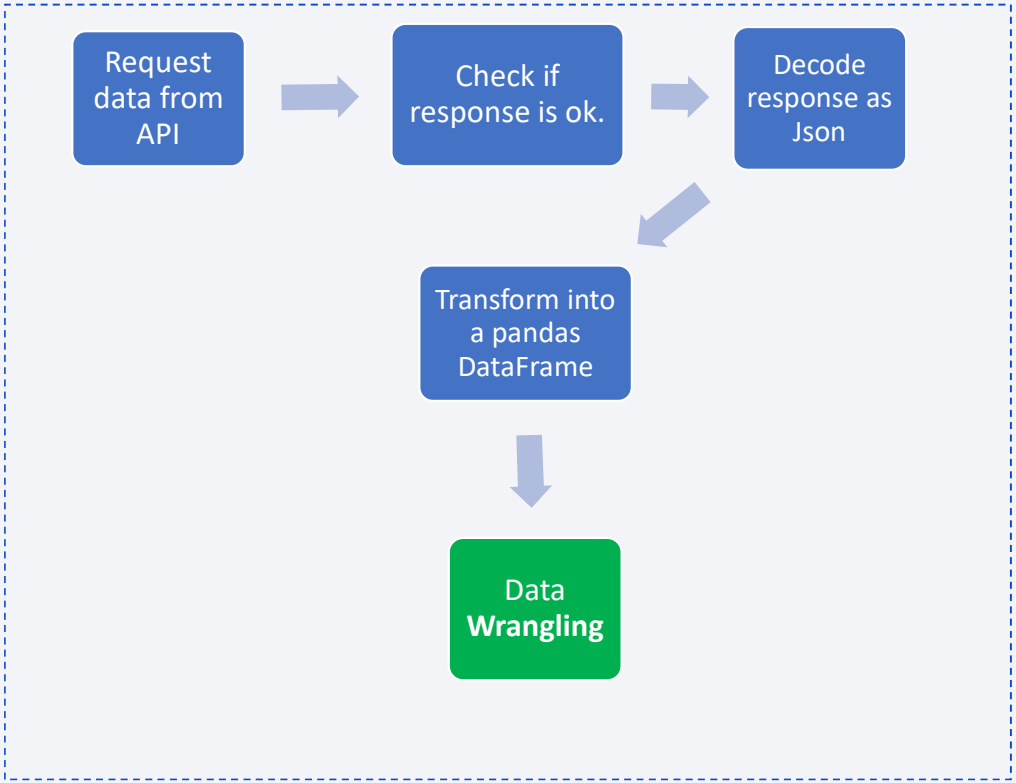
- Data was collected by two main process:
  - Space X API using the GET request
    - https://api.spacexdata.com/v4/launches/past
  - Web Scraping data from Wikipedia using beatifulsoup library
    - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data Collection – SpaceX API

- SpaceX REST API example call:

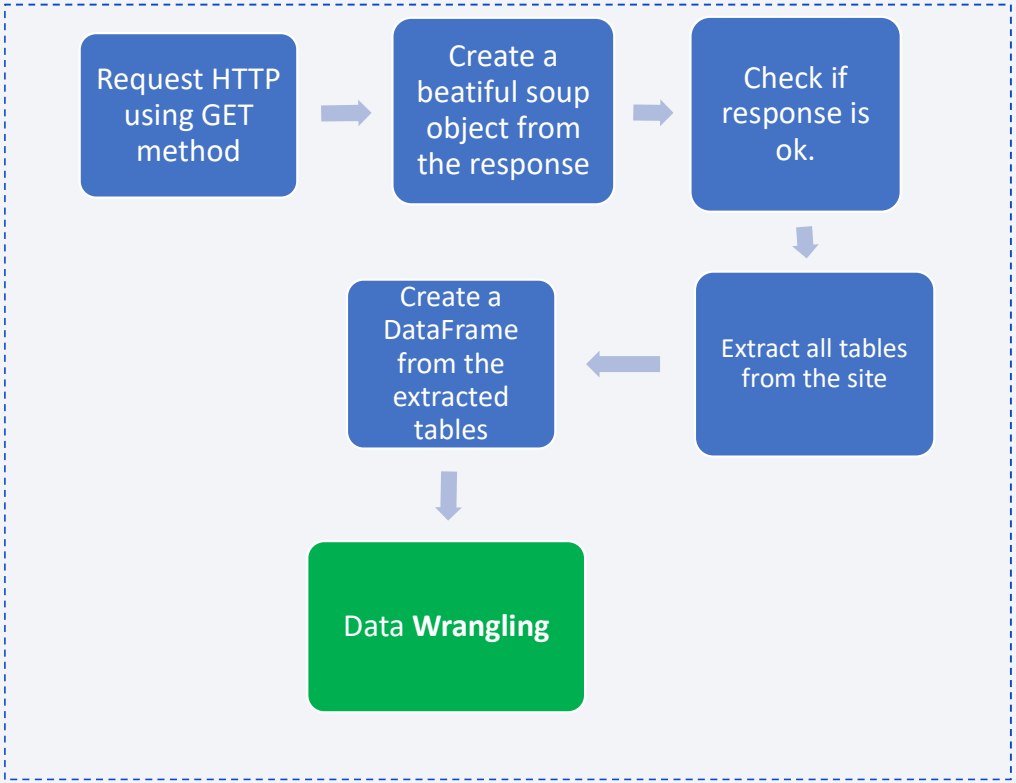
```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

- Flowchart shows the process
- Detailed notebook with all process and responses:
  - https://github.com/fbalvim/DevOps_IBM/blob/d5f20999129cf5d3c126a16caed0d0fcc679f78b/Data%20Collection%20API.ipynb



Data Collection - Scraping

- Main source:
  - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- Flowchart shows the process
- Detailed notebook with all process and responses:
  - https://github.com/fbalvim/DevOps_IBM/blob/d5f20999129cf5d3c126a16caed0d0fcc679f78b/jupyter-labs-webscraping.ipynb



Data Wrangling

- Some relevant features of the acquired data set were selected.
- Data was cleaned by removing rows with multiple cores, formatting date, and filtering date
- Class feature added, showing if a mission was successful or not.
- Data was “one hot encoded” for some orbit analyses
- Link to detailed notebook:
  - https://github.com/fbalvim/DevOps_IBM/blob/d5f20999129cf5d3c126a16caed0d0fcc679f78b/data_wrangling.ipynb
- Ex DataFrame (first 5 lines):

```
In [42]: # Show the head of the dataframe
         dfspacex.head()
```

Out[42]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.047721
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857

EDA with Data Visualization

- Charts were plotted in order to try to visualize if we could answer some of our questions about the success rate:
  - Scatter plot PayloadMass x FlightNumber – marker showing class
  - Scatter plot LaunchSite x FlightNumber – marker showing class
  - Scatter plot LaunchSite x PayloadMass – marker showing class
  - Bar chart Success rate x Orbit
  - Scatter plot Orbit x FlightNumber – marker showing class
  - Scatter plot Orbit x PayloadMass – marker showing class
  - Line chart Success rate x Year
- Link to detailed notebook:
  - https://github.com/fbalvim/DevOps_IBM/blob/d5f20999129cf5d3c126a16caed0d0fcc679f78b/eda-dataviz.ipynb

EDA with SQL

- SQL Querys successfully performed:
  - 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)
  - 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
- Link to detailed notebook:
  - https://github.com/fbalvim/DevOps_IBM/blob/d5f20999129cf5d3c126a16caed0d0fcc679f78b/eda-sql-coursera.ipynb

Build an Interactive Map with Folium

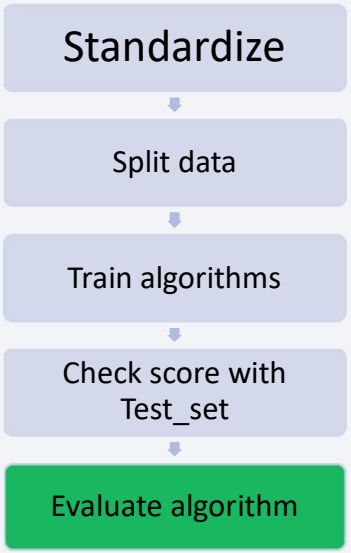
- Circles and markers were crated to show launch location on the map.
- Clusters were formed to show which launches were successful.
- Lines and markers were created to show distance between places.
- Link to detailed notebook:
  - https://nbviewer.org/github/fbalvim/DevOps_IBM/blob/49cd2566ebbc9a50fb2d4374a4993074d0d534d2/launch_site.ipynb

Build a Dashboard with Plotly Dash

- A drop-down list with the launch sites was added to the dashboard.
- A pie chart was added to the dashboard showing results based on the drop-down list.
- A slider was added to the dashboard in order to filter the selected payload mass.
- A Scatter plot Class x payload with marker filtered by the drop-down list and the payload mass filtered by the slider was added to the dashboard.
- Link to detailed notebook:
  - https://github.com/fbalvim/DevOps_IBM/blob/49cd2566ebbc9a50fb2d4374a4993074d0d534d2/DASH.py

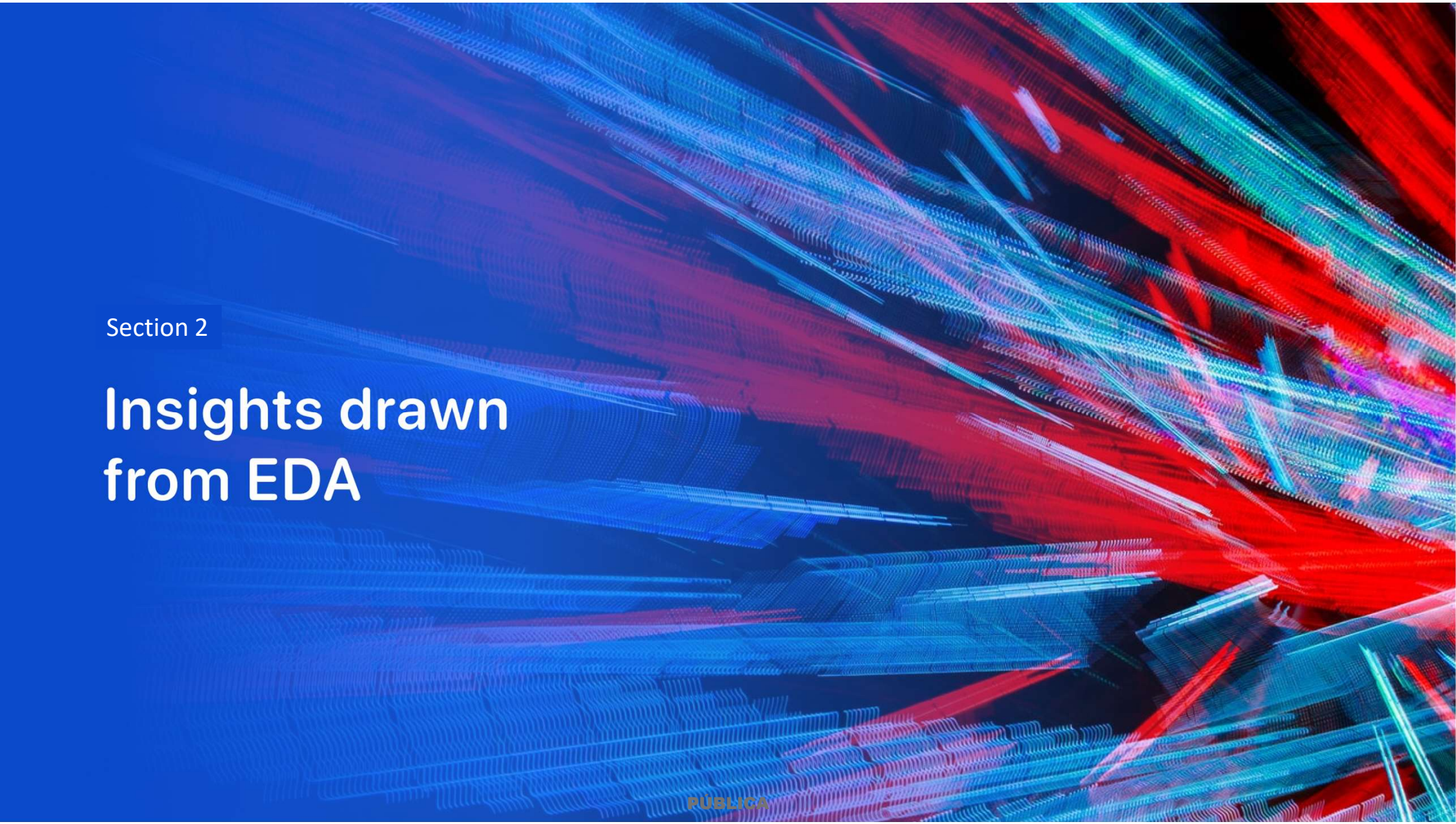
Predictive Analysis (Classification)

- Data was Standardize by the method StandardScaler() from sklearn
- Data was split in Test_set and Train_set
- algorithms from sklearn were used:
  - LogisticRegression
  - support vector machine
  - decision tree
  - k nearest
- performance was checked on the train_set, and confirmed on the test_set.
- Link to detailed notebook:
  - https://github.com/fbalvim/DevOps_IBM/blob/49cd2566ebbc9a50fb2d4374a4993074d0d534d2/Machine_Learning_Predictio.jupyterlite.ipynb

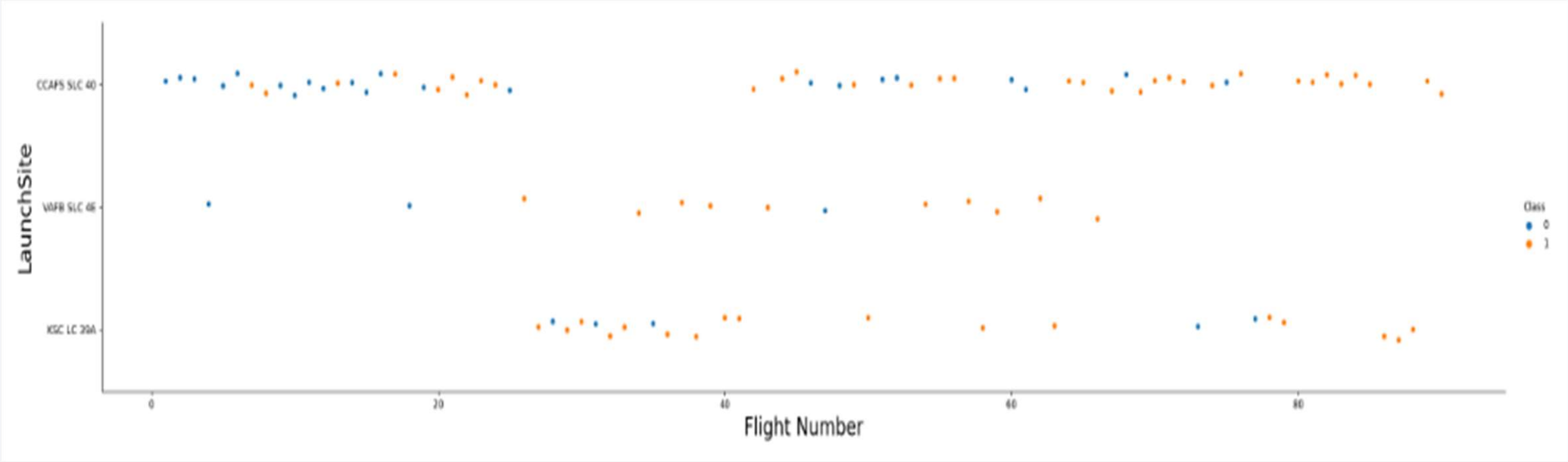


Results

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

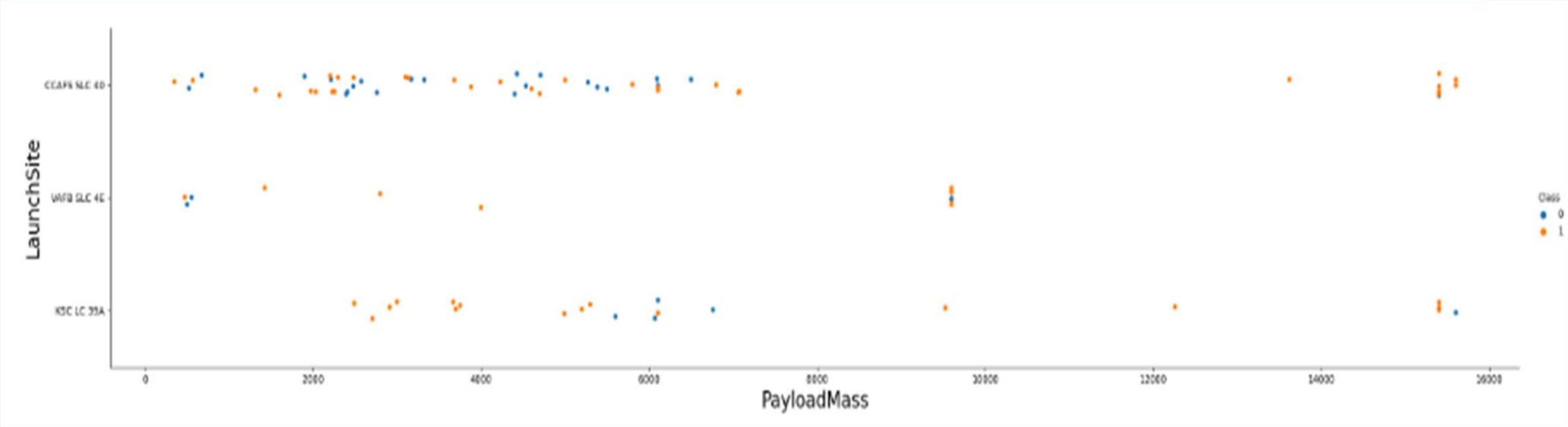

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



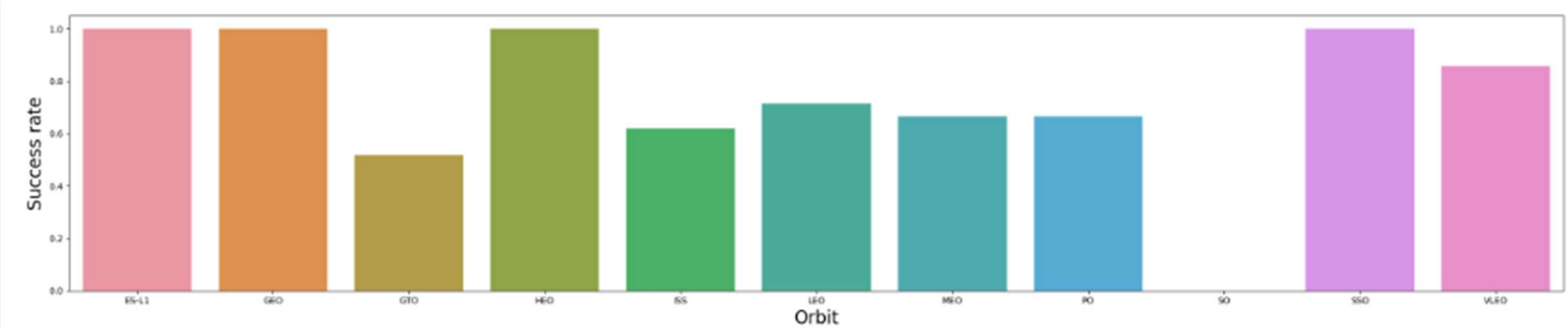
- We can see that as the flight number increases the success rate usually increases for all sites.

Payload vs. Launch Site



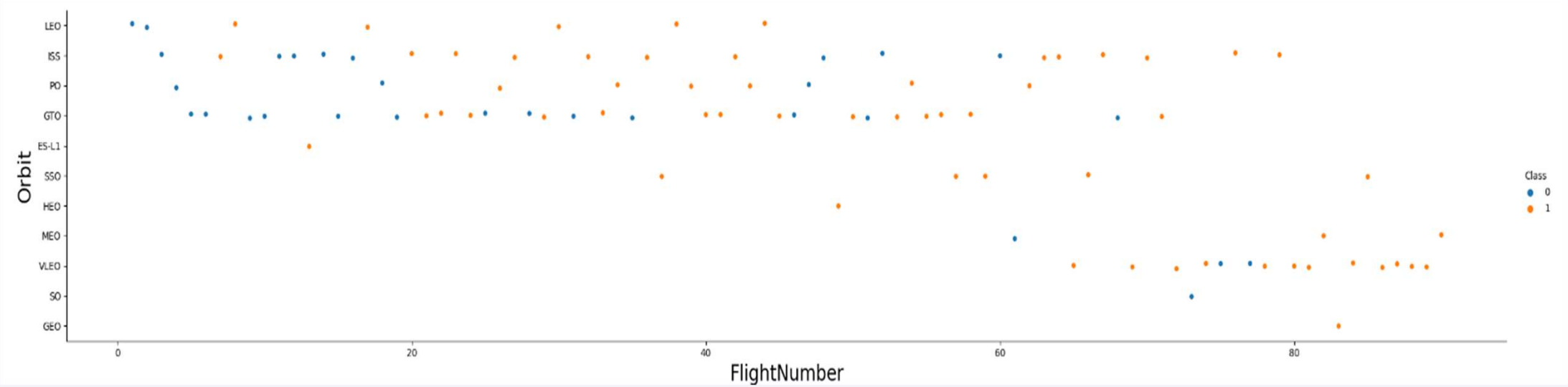
There is a clear separation here, high payload mass are much more likely to be successful.

Success Rate vs. Orbit Type



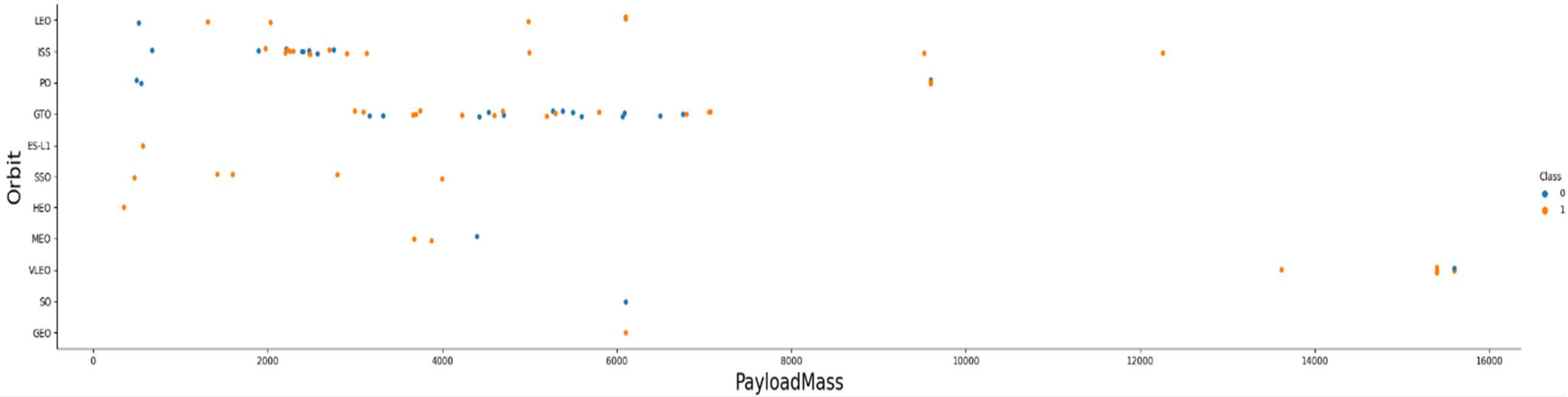
- Orbits GTO and SO have the lowest success rate.
- ES-L1, GEO and SSO have the highest success rate

Flight Number vs. Orbit Type



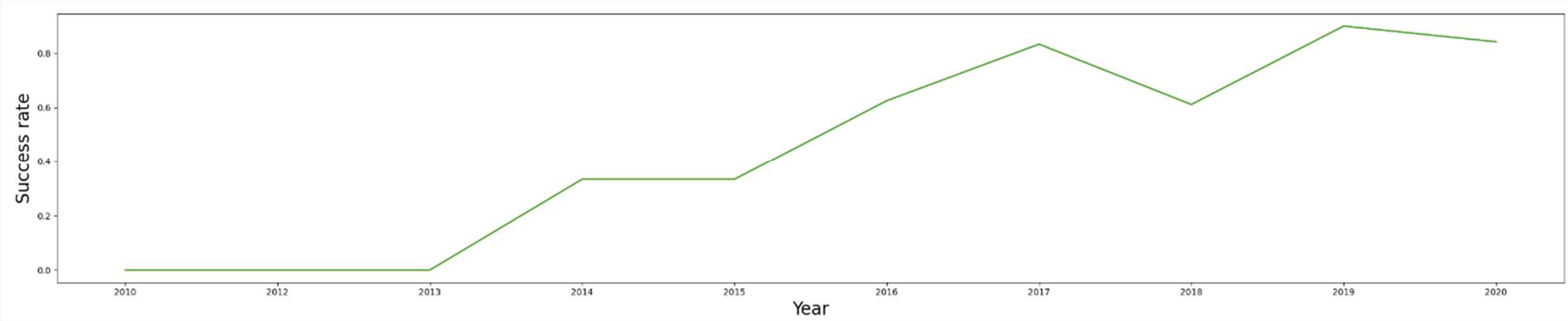
- Clearly success rate increases with the flight number
- the aim of the launches (orbits) seems to have changed over time.

Payload vs. Orbit Type



- The mass is probably correlated with the payload mass, as orbits increase in distance the launch probably needs more fuel.

Launch Success Yearly Trend



- Again, success rate seems to increase over time (flight number).

All Launch Site Names

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

In [9]:
```
%sql select launch_site from SPACEXTBL group by launch_site
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[9]:

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

- SQL query and result

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select launch_site,count(launch_site) as "count" from SPACEXTBL group by launch_site
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

launch_site	count
CCAFS LC-40	26
CCAFS SLC-40	34
KSC LC-39A	25
VAFB SLC-4E	16

- SQL query and result

Total Payload Mass

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

```
In [23]: %sql select sum(payload_mass__kg_) as "total payload mass (kg)" from SPACEXTBL where customer = 'NASA (CRS)'
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[23]:

total payload mass (kg)
45596

- SQL query and result

Average Payload Mass by F9 v1.1

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

In [5]:
```
%sql select avg(payload_mass__kg_) as "average mass(kg)" from SPACEXTBL where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[5]:

average mass(kg)
2928

- SQL query and result

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [6]:
```
%sql select min(DATE) as "Date" from SPACEXTBL where landing__outcome like 'Success%'
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[6]:

Date
2015-12-22

- SQL query and result

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [8]:
```
%sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ < 6000 and payload_mass__kg_ > 4000
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[8]:

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

- SQL query and result

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [15]:
```
%sql select mission_outcome,count(mission_outcome) as "count" from SPACEXTBL  group by mission_outcome
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[15]:

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

- SQL query and result

Boosters Carried Maximum Payload

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

In [16]:
```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[16]:

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

- SQL query and result

2015 Launch Records

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

In [26]:
```
%%sql
select month("DATE") as "month of 2015", landing__outcome, booster_version, launch_site from SPACEXTBL
where year("DATE") = 2015 and landing__outcome like 'Fail%drone%'
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUD
Done.
```

Out[26]:

month of 2015	landing__outcome	booster_version	launch_site
1	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
4	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- SQL query and result

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

In [31]:
```
%%sql
select DATE, landing__outcome from SPACEXTBL
    where landing__outcome like 'Fail%drone%' and DATE between '2010-06-04' and '2017-03-20'
    order by date desc
```

```
 * ibm_db_sa://scd83041:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[31]:

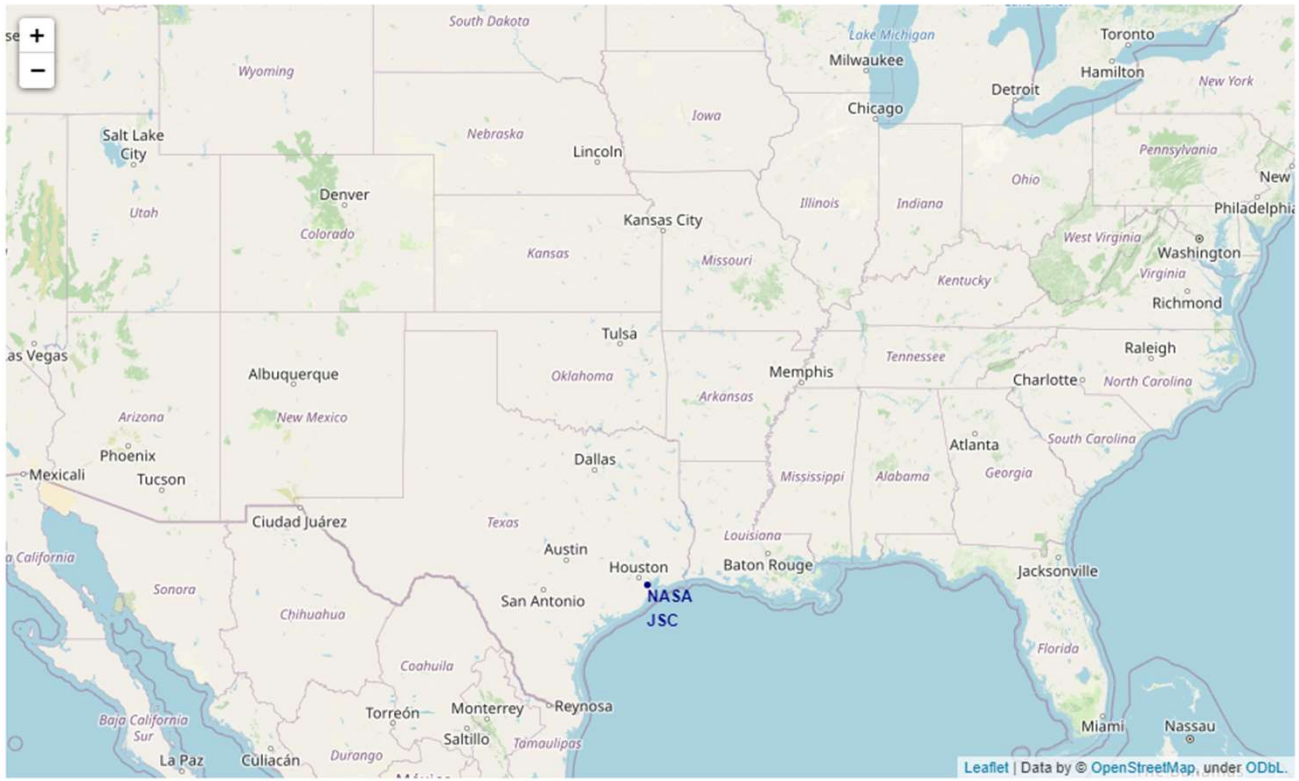
DATE	landing__outcome
2016-06-15	Failure (drone ship)
2016-03-04	Failure (drone ship)
2016-01-17	Failure (drone ship)
2015-04-14	Failure (drone ship)
2015-01-10	Failure (drone ship)

Section 3

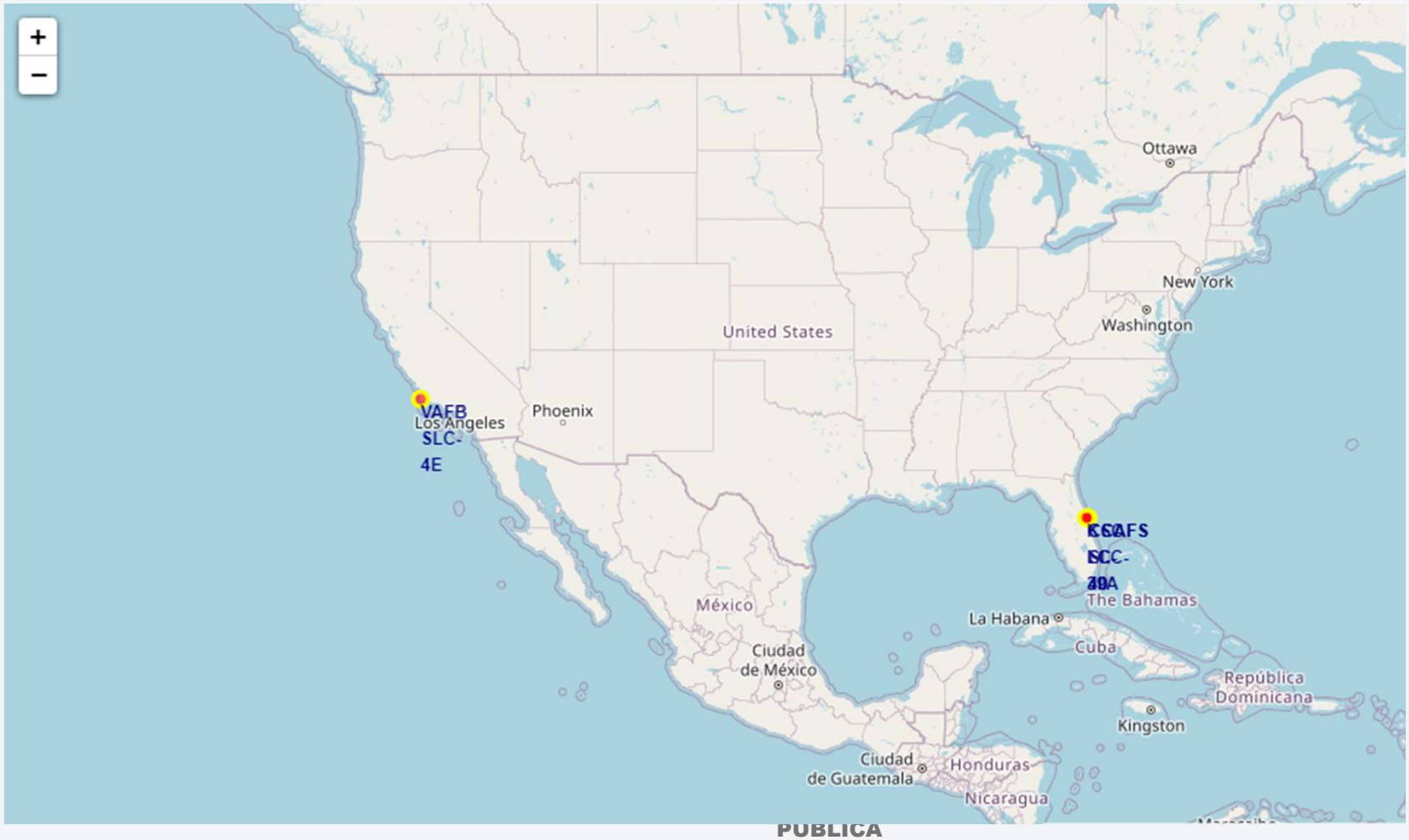
Launch Sites Proximities Analysis

Folium Map Showing NASA Johnson Space

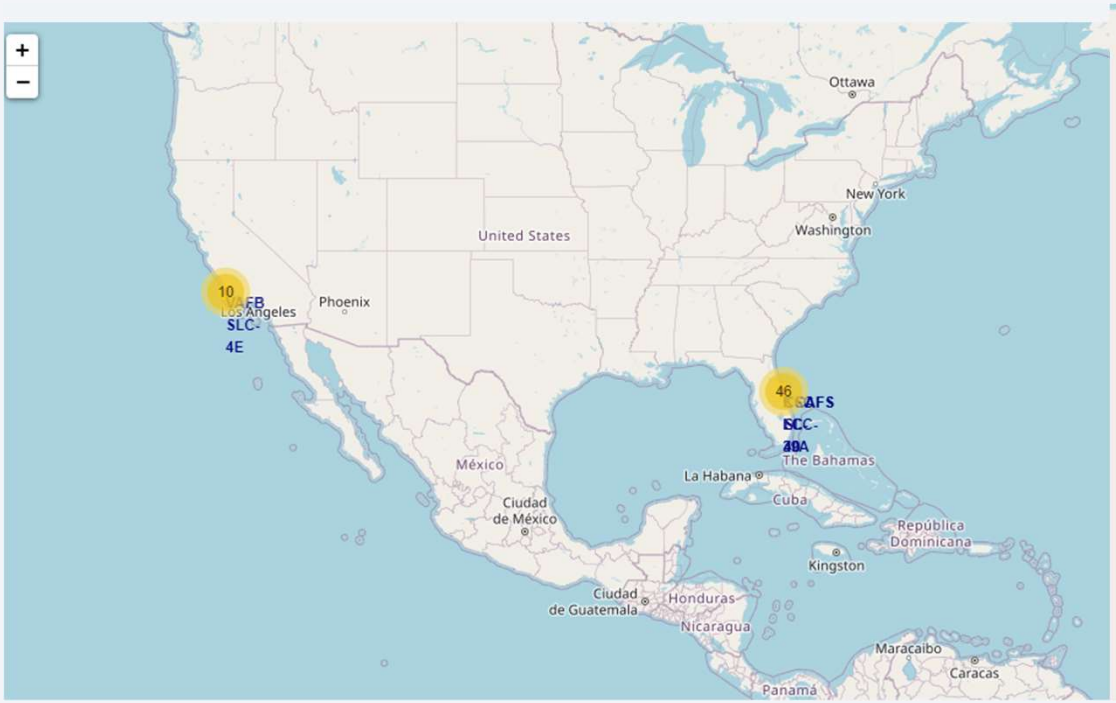
Out[8]:

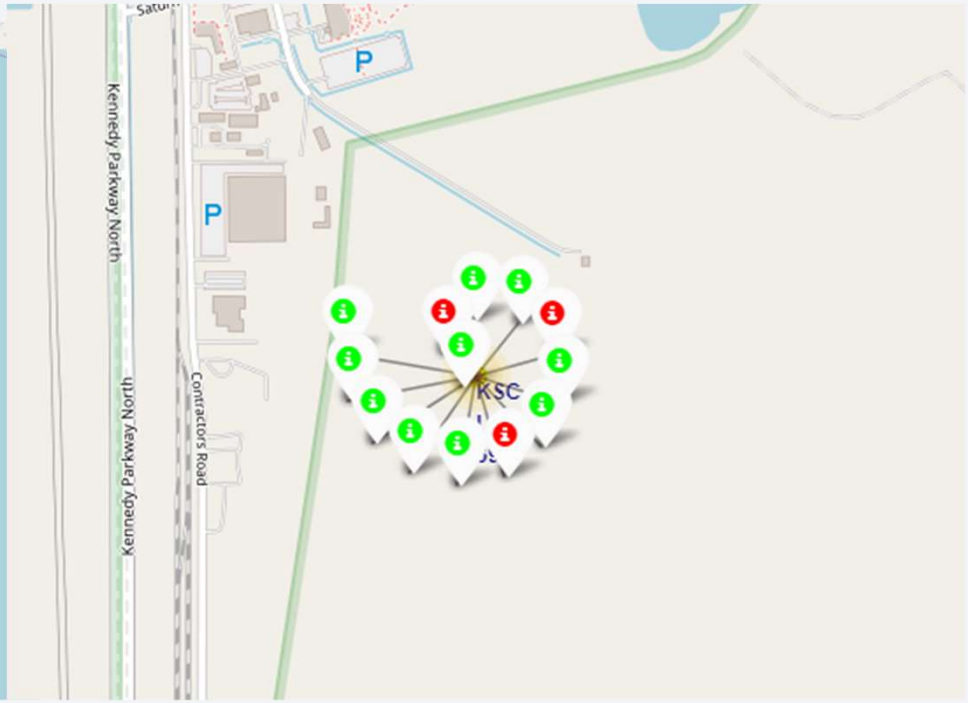


Folium Map Showing Launch sites

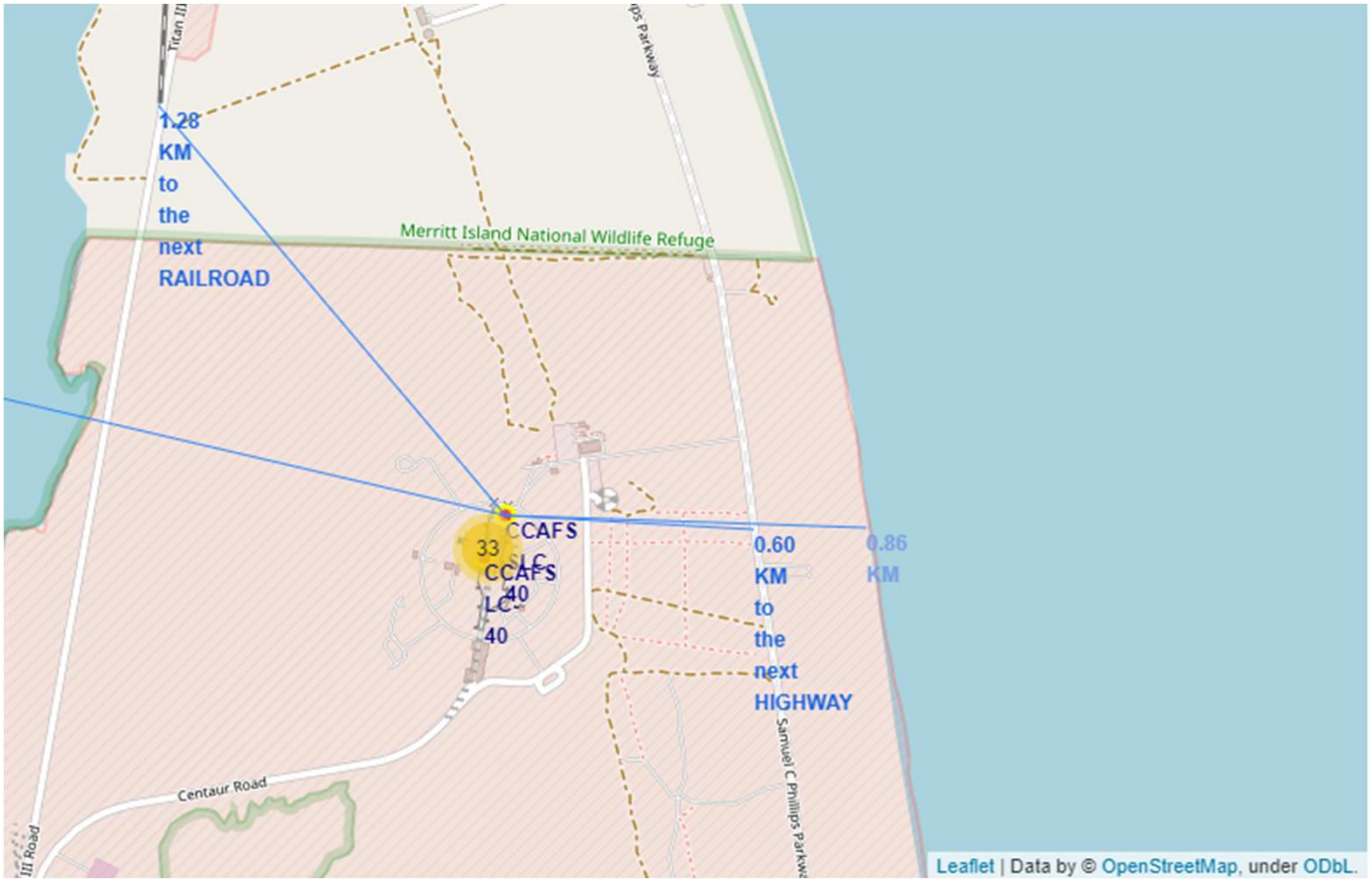


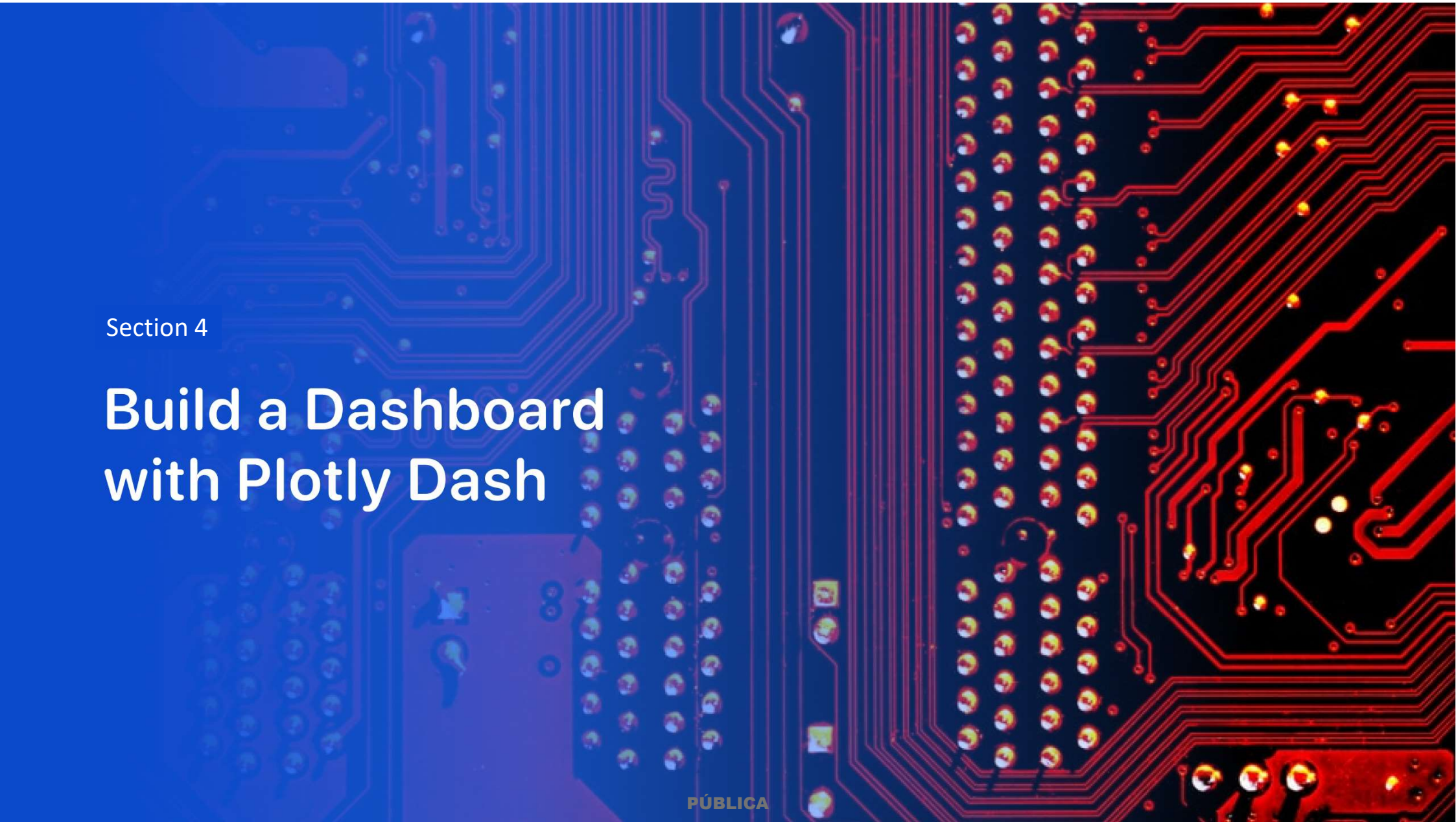
Folium Map with marker clusters



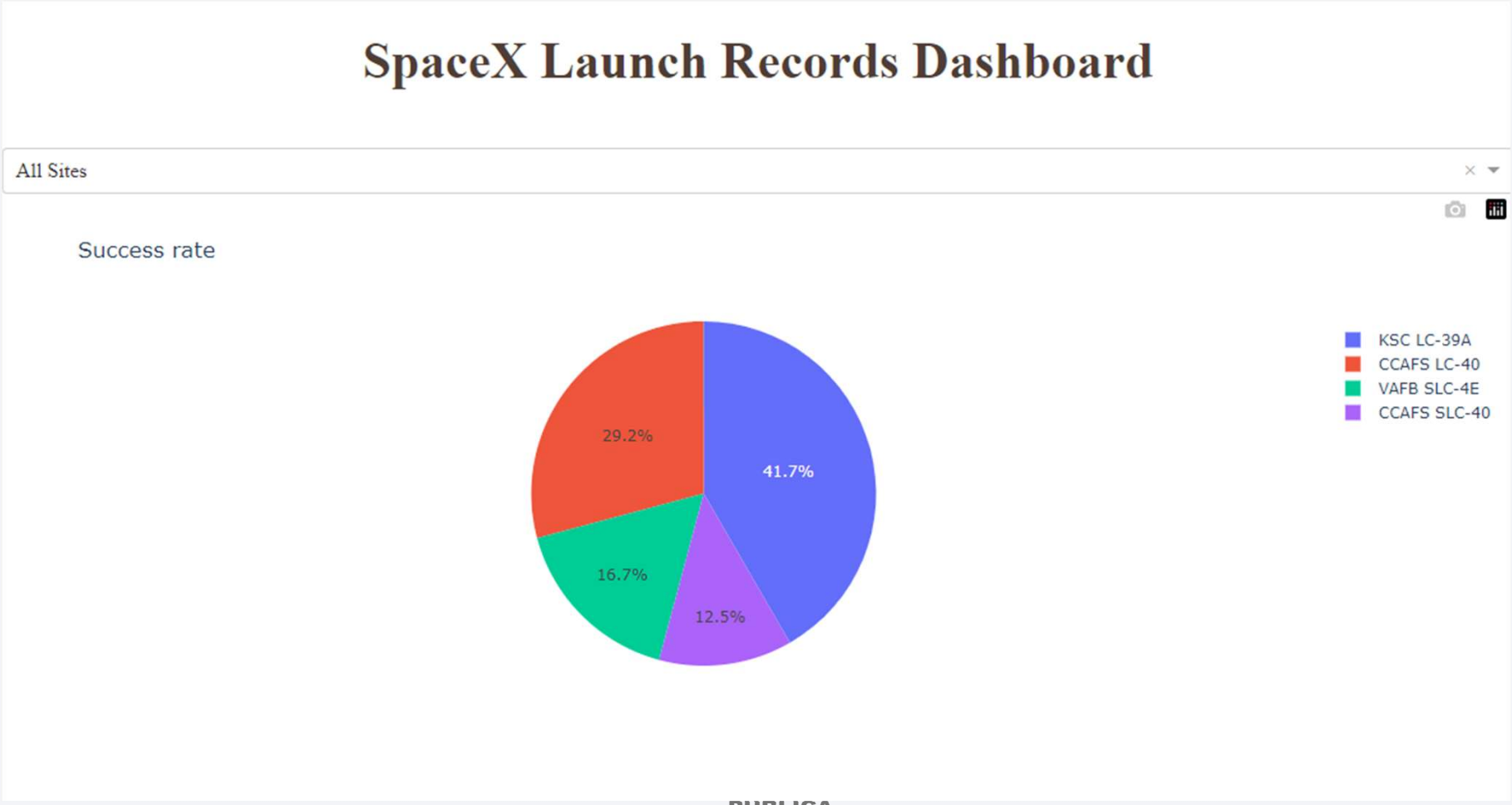


Folium Map Showing distance to the next railroad

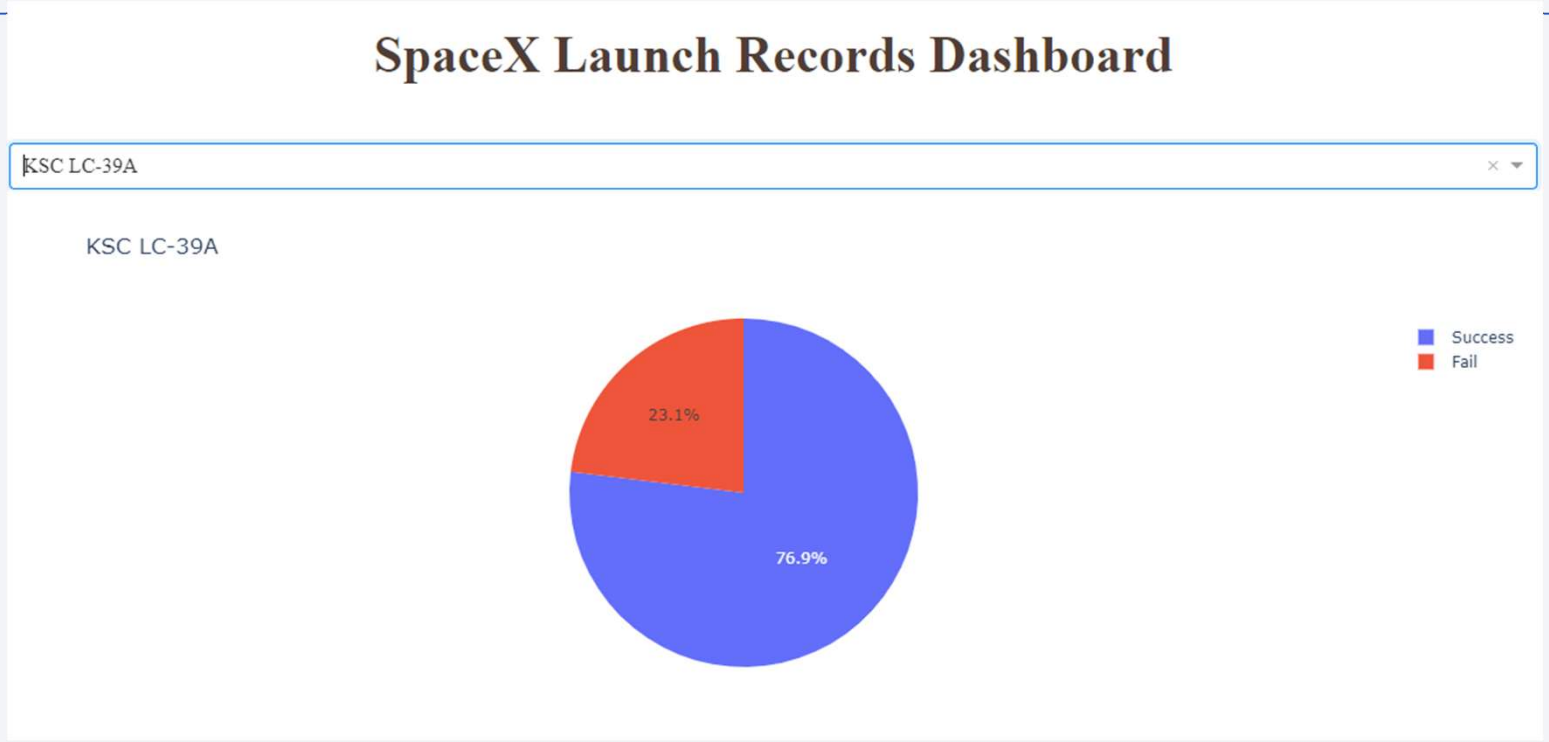




Dashboard – Success rate of all sites

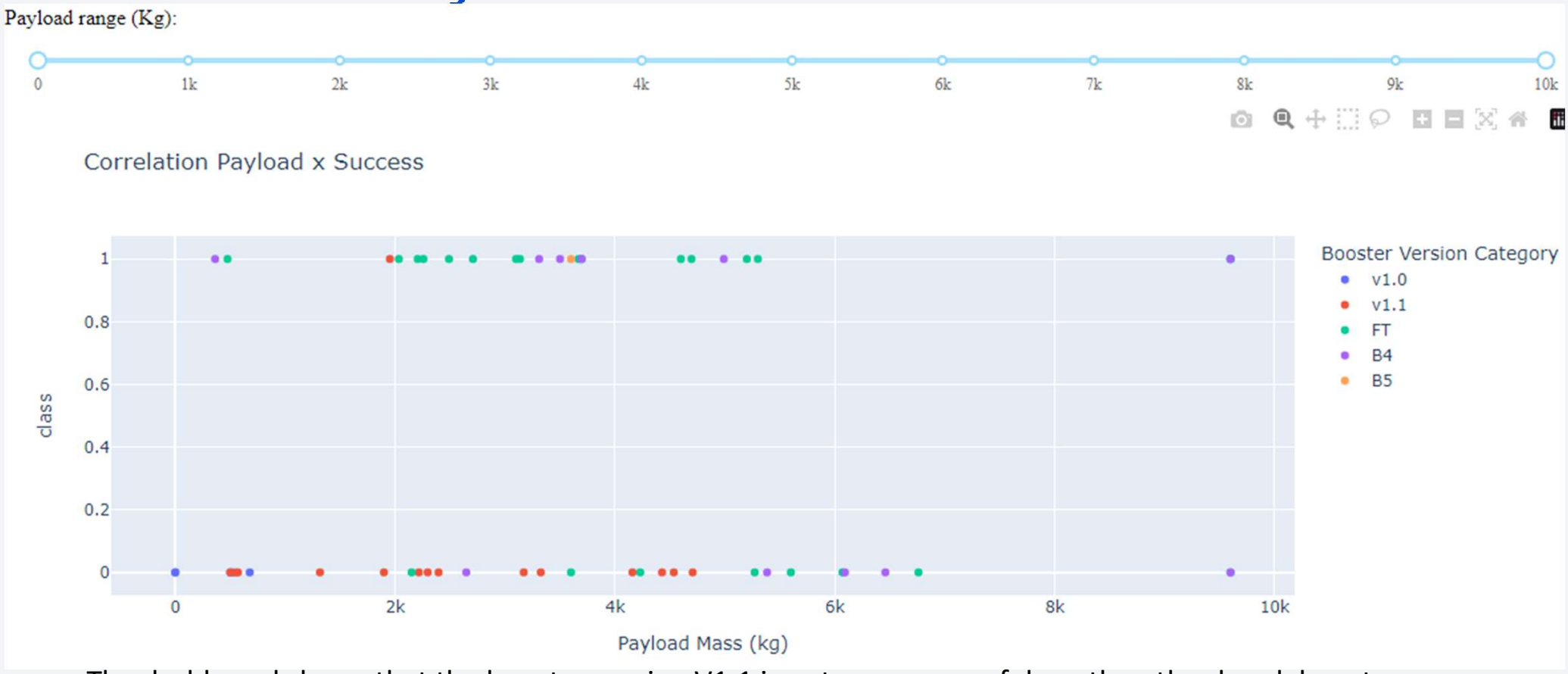


Dashboard - highest success rate



- The dashboard shows that the highest success rate is from the launch site KSC LC-39-A

Dashboard Payload vs. Launch Outcome

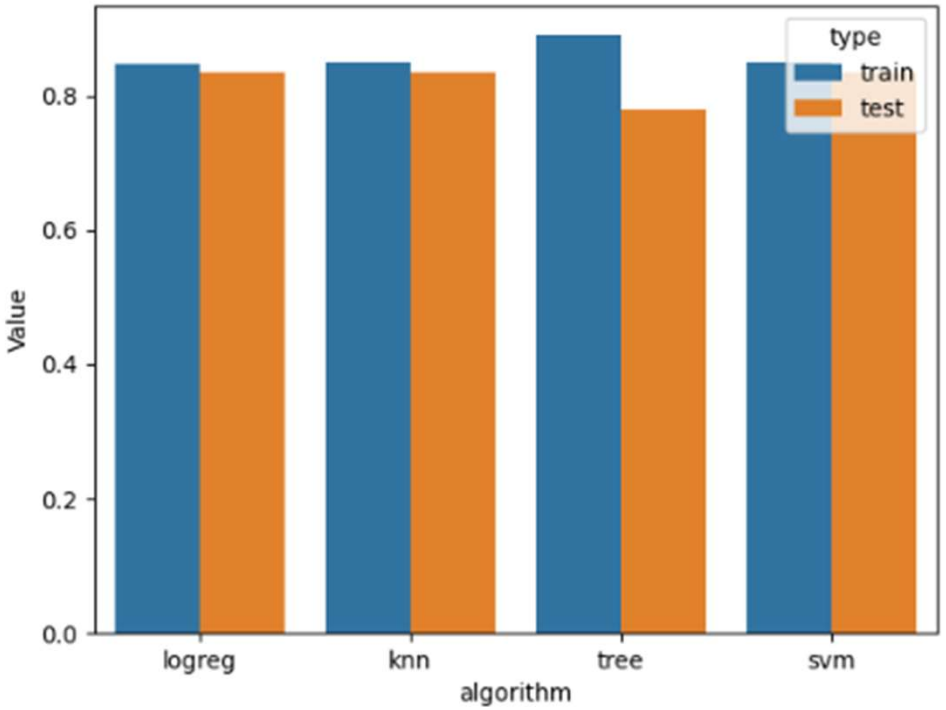


- The dashboard shows that the booster version V1.1 is not very successful, on the other hand, booster version FT seems to have a great success rate .

Section 5

Predictive Analysis (Classification)

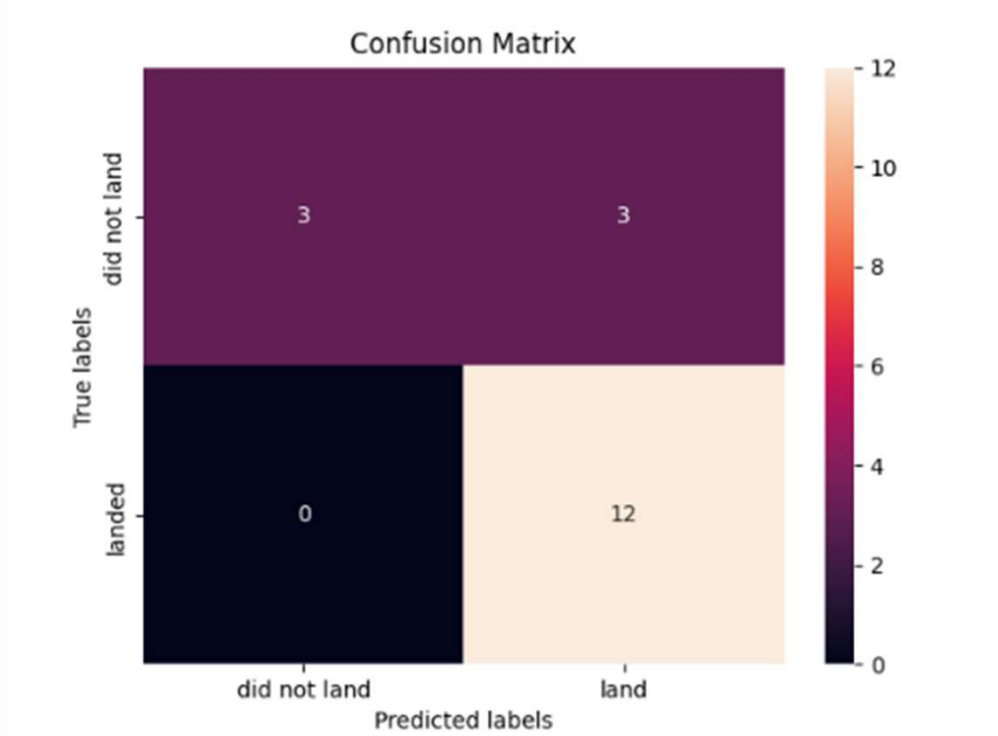
Classification Accuracy



algorithm	type	Value
logreg	train	0.846429
logreg	test	0.833333
knn	train	0.848214
knn	test	0.833333
tree	train	0.889286
tree	test	0.777778
svm	train	0.848214
svm	test	0.833333

- Regression, SVM and KNN had the same results of score on the test data.
- The test set is too small, and we cannot differentiate performance on the algorithms.

Confusion Matrix



- SVM seems to predict well missons that landed, but false positives is an issue.

Conclusions

- Success rate seems to increase over time (launch attemps)
- The data set in the machine learning is small, and a best algorithm could not be choosed
- KSC LC-39A is the launch point with the highest success rate.
- Launch sites have a great distance from cities but not from highway and railroads.

Thank you!