IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jose R. Heras
October 23, 2023

Outline

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

Executive Summary

- SpaceY is a new player on the rocket industry and wants to compete against SpaceX.
- SpaceX advertises Falcon 9 rockets as only costing 69 millions dollars compared to the 195 million dollars advertised by other providers. As result SpaceX has become a key player on the industry.
- One of the key of reasons of such difference on price is because SpaceX reuses the components used in Stage 1 which tend to be the largest component compared to the other stages.
- By making use of mission data such as payload mass and desired orbit, a model was able to calculate and produce a report, with an accuracy level of 83.33%, on wether the first stage rocket booster will complete a successful landing.
- Thanks to this analysis, SpaceY will be able to make more accurate decission and as result bid more intelligently against SpaceX.

Introduction

- As part of the Applied Data Science Capestone course this report has been made using the various stages taught during the course.
- As main objective, I take upon the role of a Data Scientist working for new company in the rocket industry, SpaceY.
- Using various techniques and methodologies it is my objective to produce models and reports that will aid SpaceY to make more informed bids against SpaceX.

Introduction

- SpaceX has become a key player in the rocket industry mainly due to their advertised price of 69 millions dollar, which is accomplished due to them reusing the Stage 1 Rocket.
- The Stage 1 Rocket, being one of the biggest components, tends to take a big part of the overall budget. By recovering this component SpaceX is able to reduce the overall cost.
- As impressive as it, recovering the first stage is no easy task. SpaceX is not always able to do so and that is mainly due to circumstances where parameters such as payload, orbit, location, and others, play an major role in the recovery.
- It is the goal of this report to create a prediction model that will take into account those parameters to decide whether the first stage will make a successful landing and as result help SpaceY to take less risky bids.

Section 1

Methodology

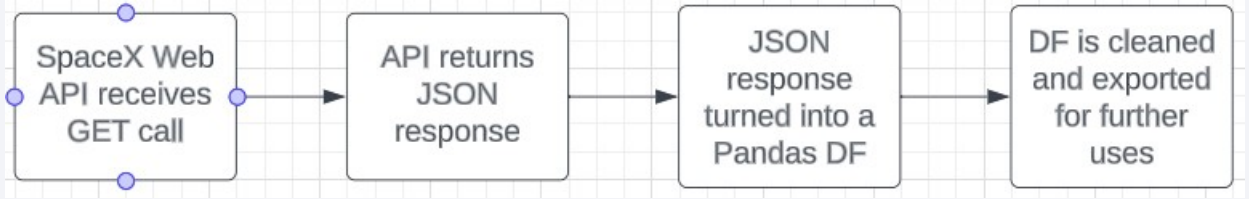
Methodology

Executive Summary

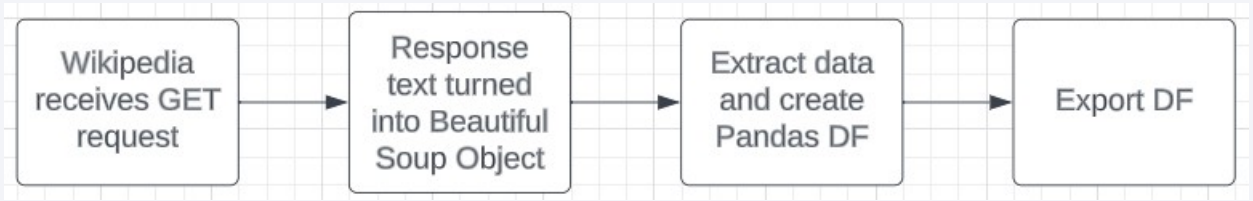
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Selecting only necessary columns for our model
  - One Hot Encoding for models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

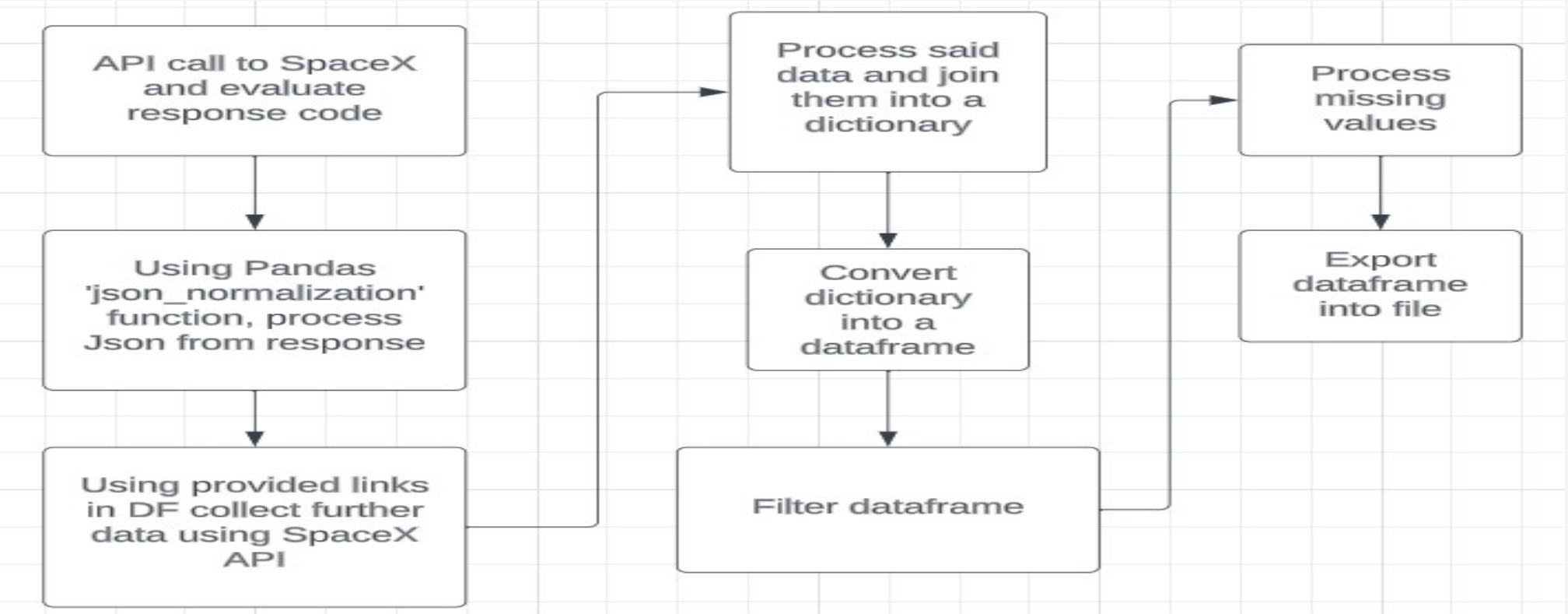
- SpaceX API and web scrapping Wikipedia serve as data source for our purpose
- SpaceX API provides us with data on rockets, payload, launchpad, cores, flight number, and date
- Space X API URL is : https://api.spacexdata.com/v4



- Data scrapped from Wikipedia includes: Flight No., Date and time, Launch site, Payload, Payload mass, Orbit, Customer and Launch outcome.
- Wikipedia URL : https://en.wikipedia.org/w/index.phptitle=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

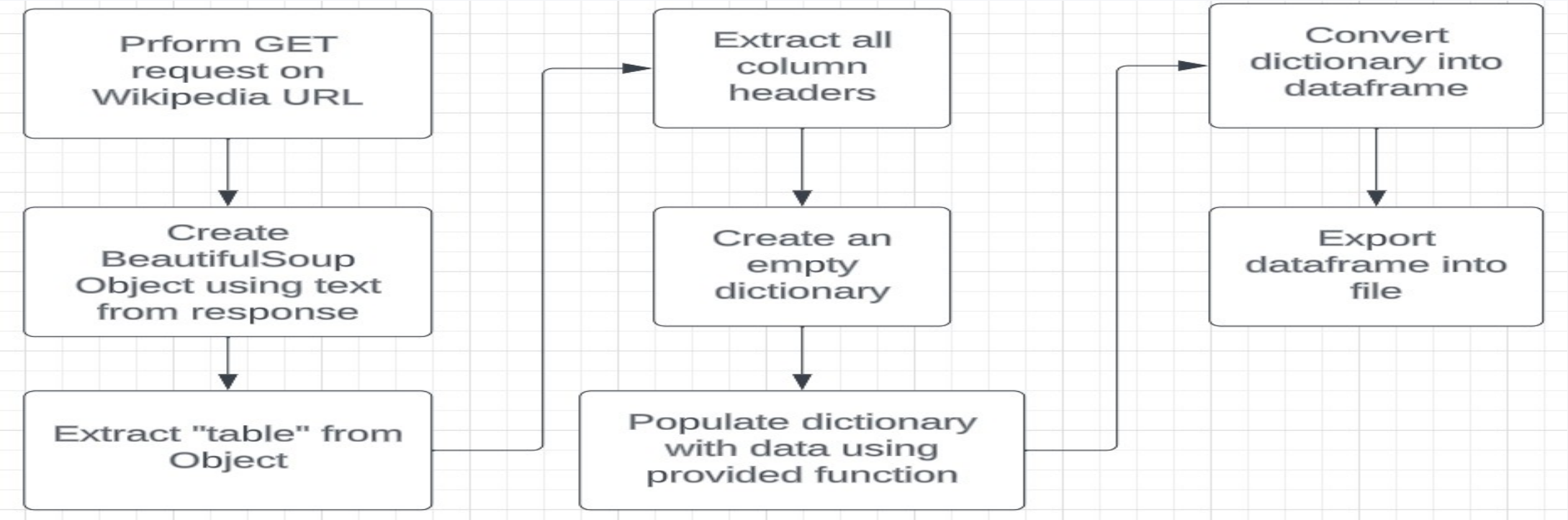


Data Collection – SpaceX API



GitHub URL

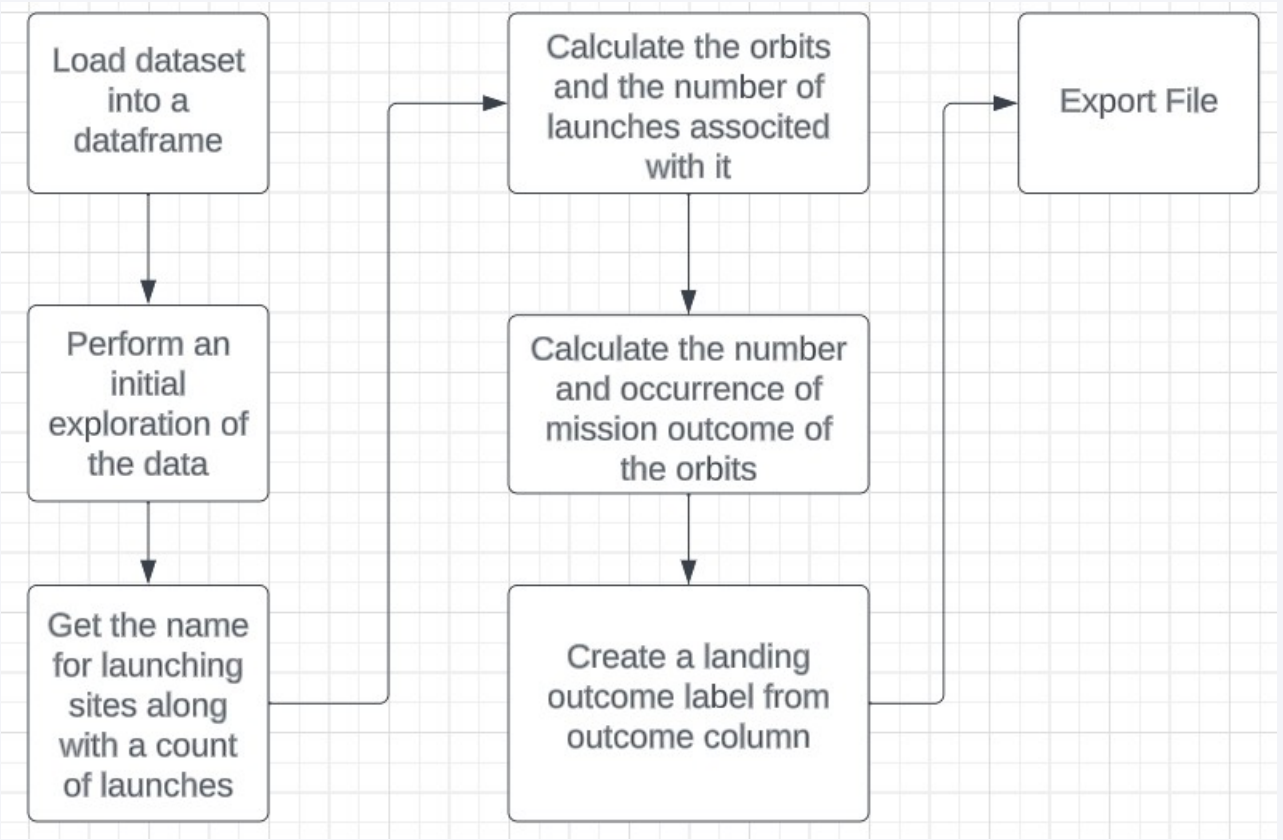
Data Collection - Scraping



GitHub URL

Data Wrangling

- The data includes both successful and non-successful cases of landing.
- For our purpose our data wrangling will concentrate on turning those outcomes into training labels of ‘1’ for successful landing and ‘0’ for non-successful.



GitHub URL

EDA with Data Visualization

- Our EDA and Data Visualization includes the following graphs:
  - Flight Number vs Payload Mass (Scatter)
  - Flight Number vs Launch Site (Scatter)
  - Launch Site vs Payload Mass (Scatter)
  - Orbit and Class (Bar Graph)
  - Flight Number vs Orbit (Scatter)
  - Payload Mass vs Orbit (Scatter)
  - Success Rate vs Year (Line)
- Scatter plot were mainly use to determine relationship between those parameters
- Bar plots were intended to find relationship between numerical and categorical variables
- Our line graph is intended to give us a insight into launch trends in yearly periods

GitHub URL

EDA with SQL

- Performed SQL were intended to provide us with more insight on the data such as:
  - Getting names of unique sites in the space mission
  - Display 5 records where launch sites begin with the string ‘CCA’
  - Display 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 groud pad was achieved
  - List boosters’ names 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 maximun payload mass.

EDA with SQL

- Continuing with out list
  - List the records which will display the month names, failure landing outcomes in drone ships, booster versions, launch site for the months in year 2015
  - Rank the count landing outcomes(such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20

GitHub URL

Build an Interactive Map with Folium

- Folium map centers on displaying launch location along with representation of successful and non-successful launch at sites
  - Red circles represent the launch site locations
  - Markers represent the amount of launches that have taken place at that location
  - Grouping points used to cluster various marker located at the same coordinates
  - On those market a Green marker represents a successful launch while a red one represent non-successful ones
  - There are marker to represent location of plotted locations along with lines which represents the distance between lunch site and the marker

GitHub URL

Build a Dashboard with Plotly Dash

- Our Dashboard contains various elements and graphs such as:
  - Drop downs which enable us to select ‘Launch Site’, by default it start at ‘All Sites’
  - Pie Charts which shows the all successful launches for a location and all successful and non-successful for ‘All Sites’
  - Slider used to select the payload mass based on weight by modifying it allows to change the value used in the scatter plot
  - Scatter chart which shows correlation between payload and launch success

GitHub URL

Predictive Analysis (Classification)

- Built:
  - Import ML algorithms from SciKit Library
  - Load the dataframe
  - Standardize the data
  - Separate data in two groups (Training, Testing)
- Modeling
  - Create object for each machine learning algorithm
  - Set the parameters to be used with GridSearchCV (Varies on algorithm)
  - Train the model using training data
- Evaluation
  - GridSearchCV will give us the best hyper parameters for that model
  - Load test data into model and get results
  - Plot confusion matrix and evaluate results
- Improving
  - Compare results between all models using its accuracy score
  - Model with best results will better serve our purpose

GitHub URL

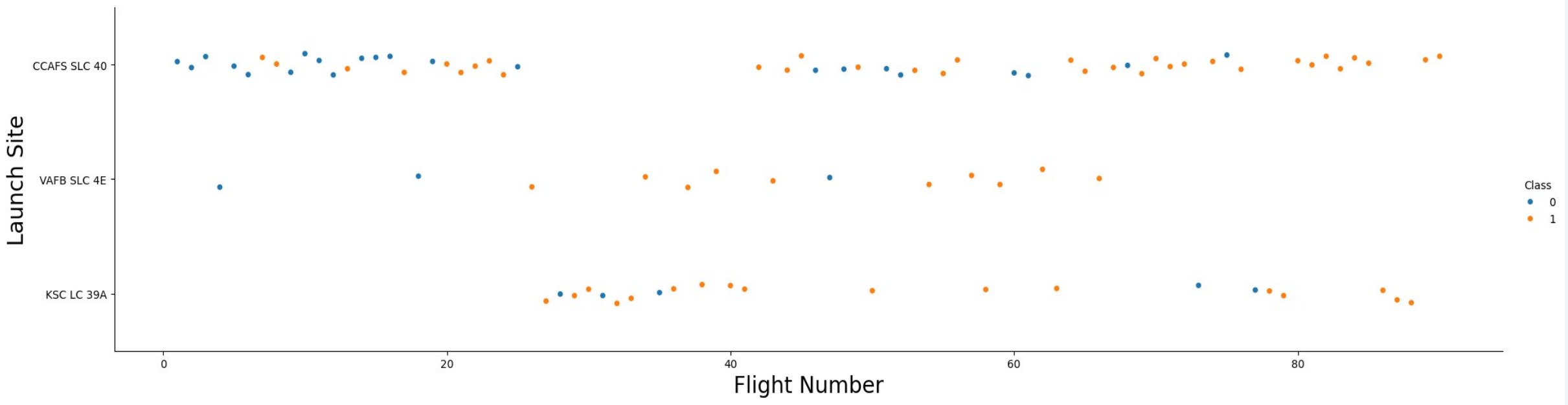
Results

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

Section 2

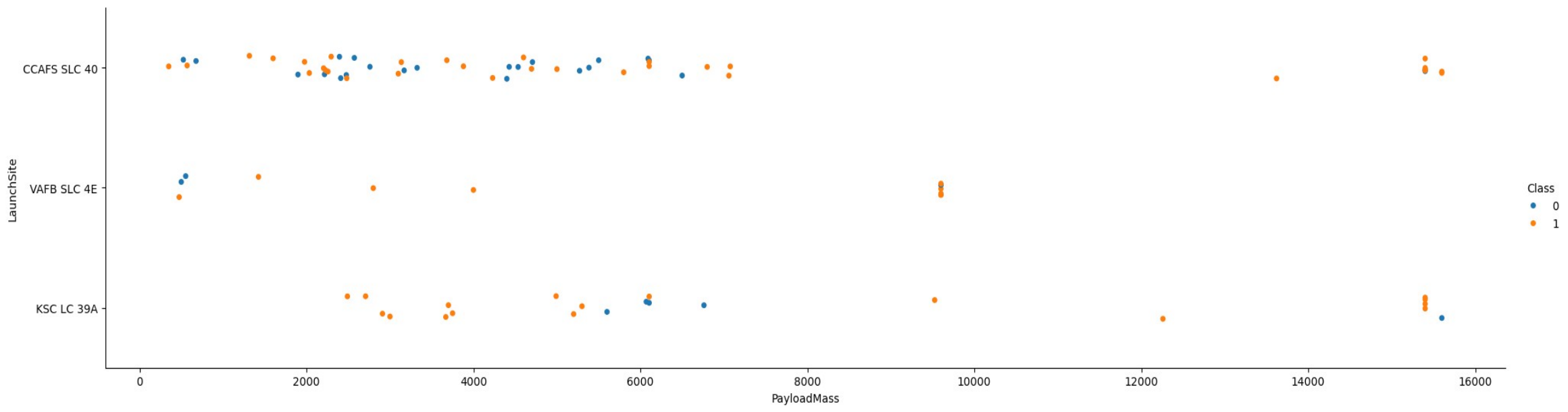
Insights drawn from EDA

Flight Number vs. Launch Site



- The graph let us see that with an increase in the number of flights the number of successful launches is increasing as well.

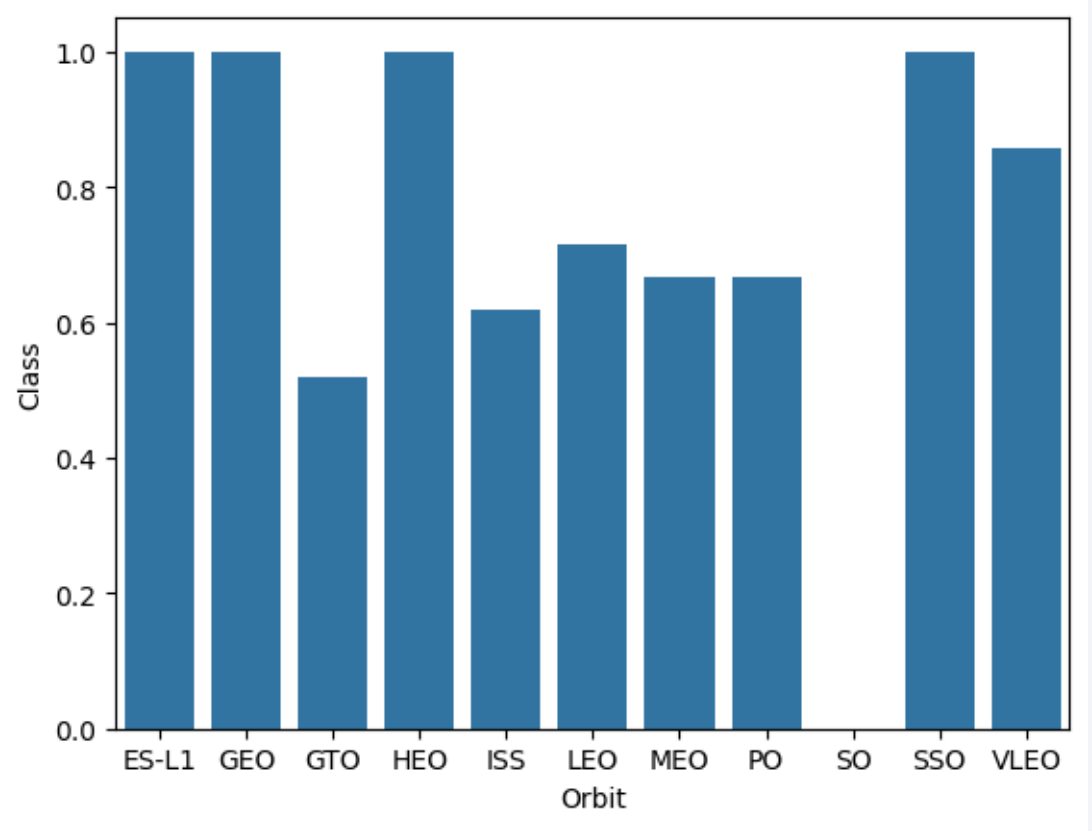
Payload vs. Launch Site



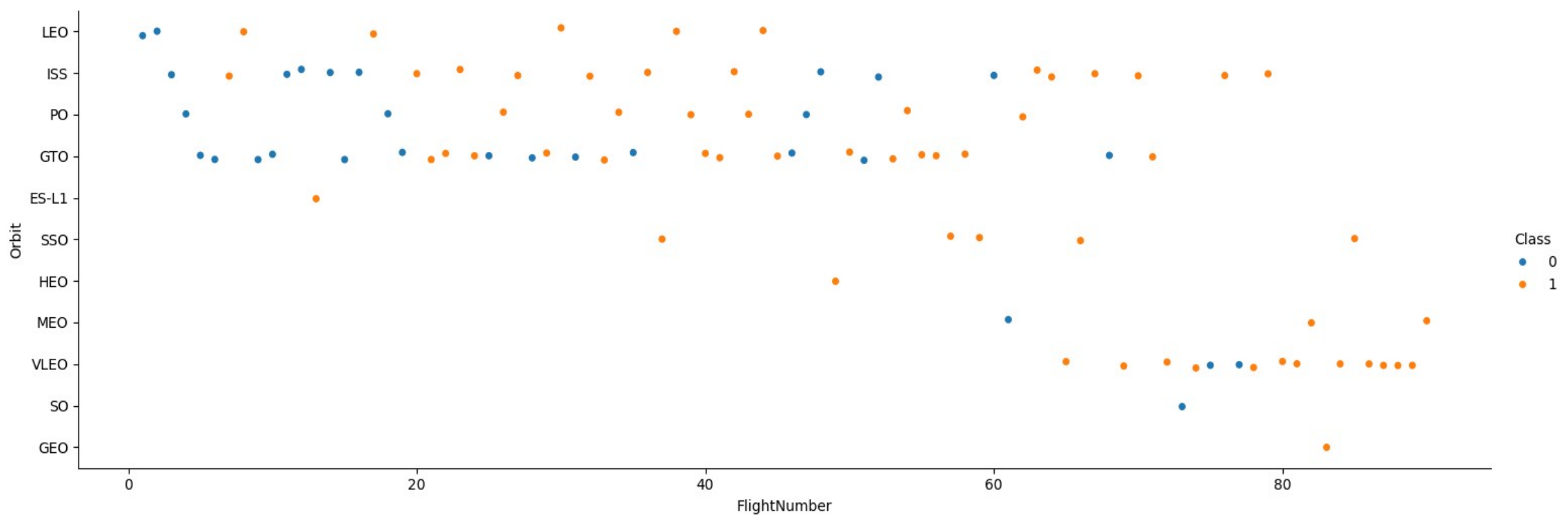
- It seems that most launches tend to have payload that are on the lower side of the scale. As a result there is a mixture of successful and non-successful launches. It is important to notice that mission with heavier payload are less common but with an overall higher success rate

Success Rate vs. Orbit Type

- Our graph lets us see that there are a group of orbits that have greater success rate compared
- This may be due to the number of mission on those orbits along with other variables
- Further analysis of the data may be need it to get a better insights

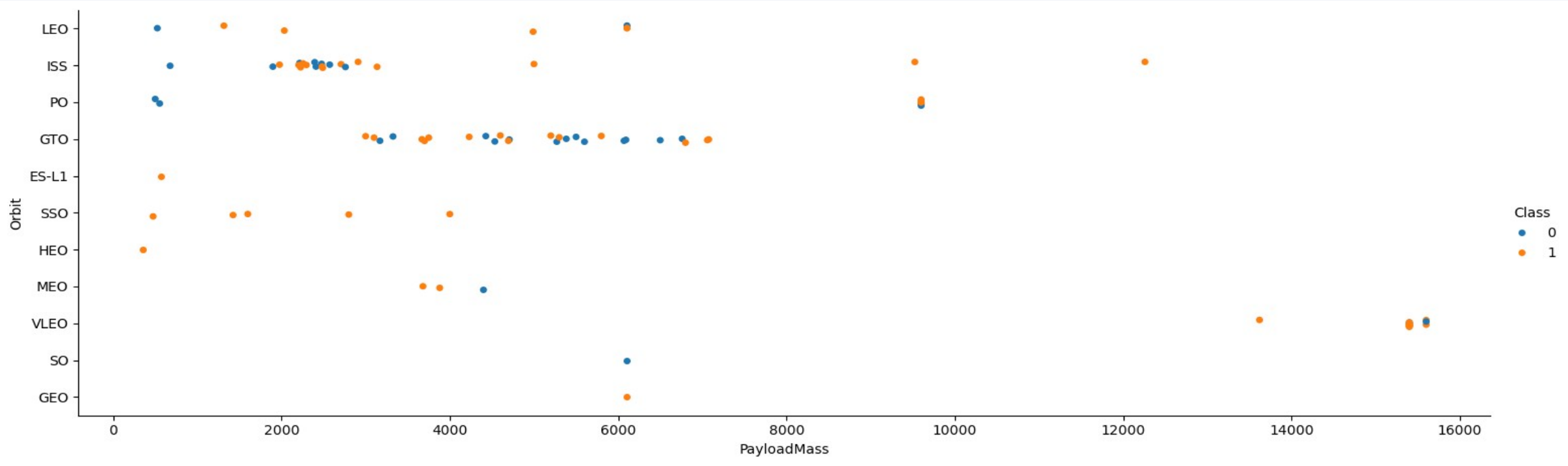


Flight Number vs. Orbit Type



- In this chart we can observe that success rate has gone up along with number of flights
- I could attribute this change due to gained knowledge on how to work with those orbits

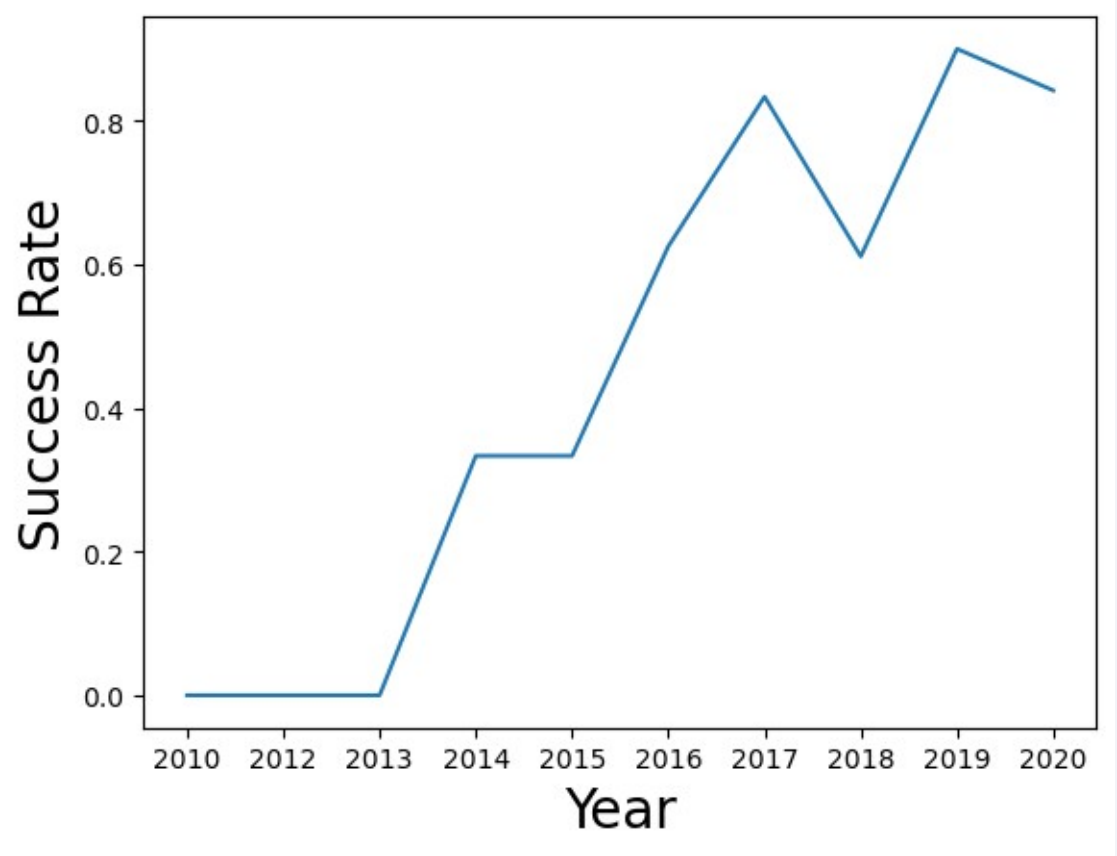
Payload vs. Orbit Type



- Payload mass tend to shift to lower side of the scale on most missions
- In overall mission with lower payload tend to have higher rate of success on specific orbits such as ‘SSO’

Launch Success Yearly Trend

- Since 2013 the number of success rate has been steadily increasing
- There are only a few gap where success rate has either stop growing or started to go down
- Perhaps those could be attribute to periods where new technology was being test or other factors



All Launch Site Names

- Our result is obtained by doing the following:
  - Selecting the Column containing the launch site names
  - Using ‘Distinct’ to tell our program to display one iteration of whatever values are stored in the column

```
%sql select distinct Launch_Site from SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5    "SPACEXTABLE": Unknown word.
```

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- We use keyword “where” to express a condition which in our case is any string that start with ‘CCA’
- This is expressed by using keyword ‘like’ followed by ‘CCA%’
- Next keyword ‘limit’ returns only the specified amount of records(In our case 5)

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__kG_) as total_mass from SPACEXTABLE where Customer like 'NASA%'

 * sqlite:///my_data1.db
Done.

total_mass
99980
```

- Obtain the total sum of all mission related to NASA
- Obtaining the total can be achieved by using ‘sum’ in the column containing the values
- This return the total sum of all values stored in that column

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as AVG_Payload_Mass from SPACEXTABLE where Booster_Version like 'F9 v1.1%'

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

AVG_Payload_Mass
2534.6666666666665

- Query returns the average of the values contained on the given column.
- Values in that column correspond to the record that match the criteria passed by the where constraint.

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTABLE where Mission_Outcome = 'Success'
```

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

min(Date)
2010-04-06

- Query return the value corresponding to the record filtered by the condition
- Min function returns the lowest value which in our case it is the first date with a success record

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select distinct Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)'and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

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

- Query returns the records that match the corresponding values passed to condition
- It makes uses of keyword such as “and”, “< and >”

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) as Mission_Count from SPACEXTABLE group by Mission_Outcome
```

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

Mission_Outcome	Mission_Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Query returns a table which contains a mission outcome along with a count
- It makes use of “group by” to group different mission outcomes and count function to return a count of how many member are in that group.

Boosters Carried Maximum Payload

- Query returns booster version that have carried the max amount of payload mass
- It uses a sub query which obtains the value corresponding to the max value store in payload mass
- Outer query takes that value and select records that match that given value

```
%%sql
select distinct Booster_Version
from SPACEXTABLE
where PAYLOAD_MASS__KG_ = (
    select max(PAYLOAD_MASS__KG_)
    from SPACEXTABLE
)
```

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Query returns records that match the given criteria in the where condition
- It makes uses of cases to match the corresponding number to a given month name
- Return a table where the month number has been replaced with a month name corresponding to the date given

```
%%sql
select
    case substr(Date, 6, 2)
        when '01' then 'January'
        when '02' then 'February'
        when '03' then 'March'
        when '04' then 'April'
        when '05' then 'May'
        when '06' then 'June'
        when '07' then 'July'
        when '08' then 'August'
        when '09' then 'September'
        when '10' then 'October'
        when '11' then 'November'
        when '12' then 'December'
    end as 'Month',
    Landing_Outcome as 'Landing Outcome',
    Booster_Version as 'Booster Version',
    Launch_Site as 'Launch Site'
from SPACEXTABLE
where substr(Date, 0, 5) = '2015'
and Landing_Outcome = 'Failure (drone ship)'
```

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

Month	Landing Outcome	Booster Version	Launch Site
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Query return a count of landing between the given dates.
- Ranking is organized from highest to lowest using ‘order by’

```
%%sql
select Landing_Outcome, count(Landing_Outcome) as Total
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
order by Total desc
```

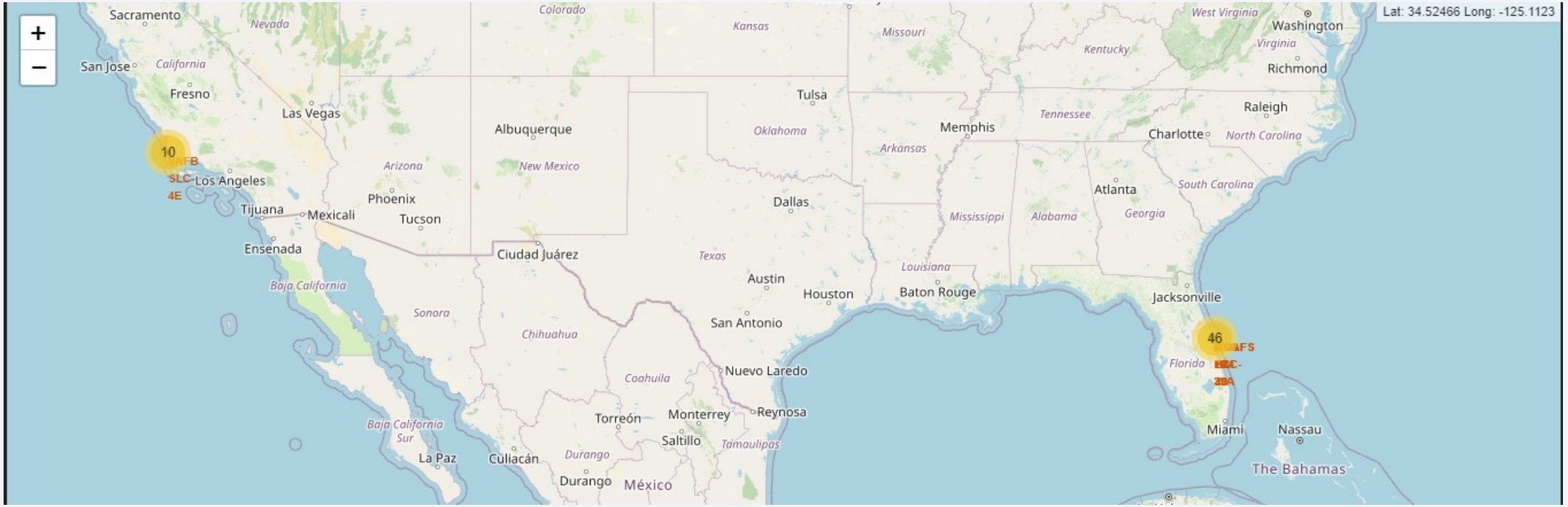
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Total
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

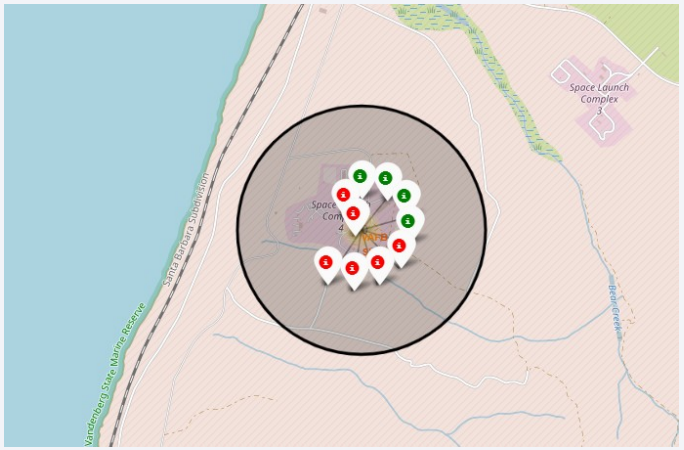
Launch Sites Proximities Analysis

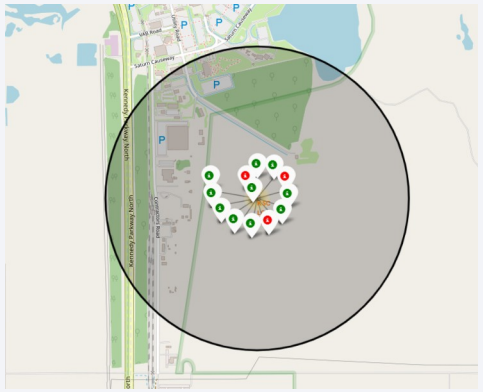
Launch Locations

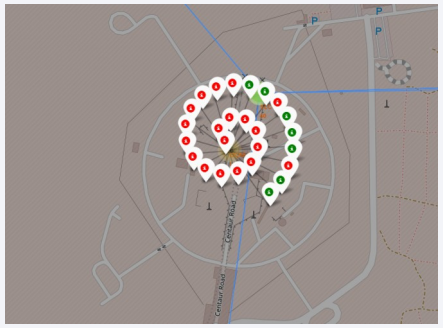


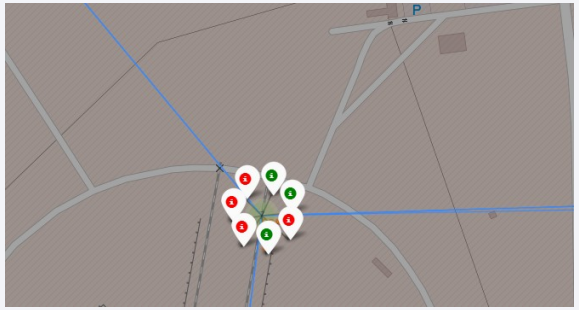
One things that stands out while looking at the map is the fact that most, if not all, launches have taken place at locations near the coast.

Lunches outcomes with color code



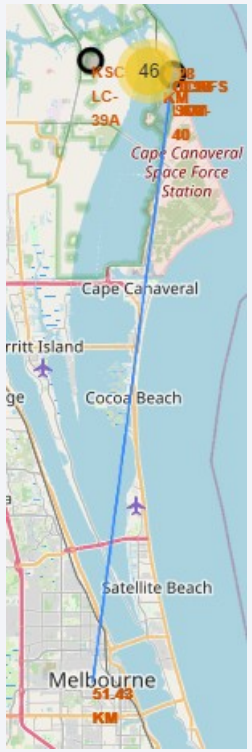




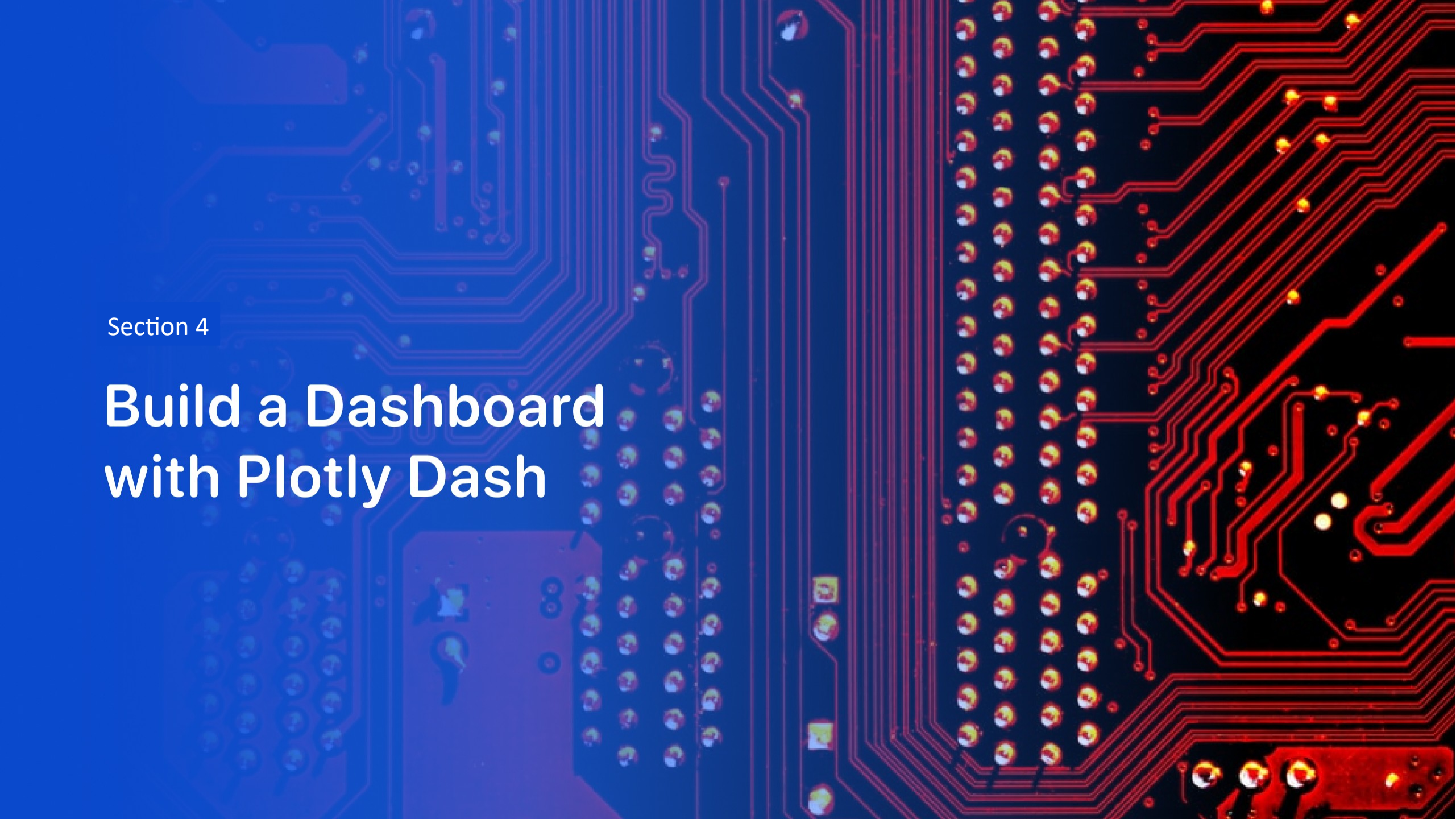


Most of all locations, except for one, seen to have a similar ratio of successful vs non-successful mission attempts.

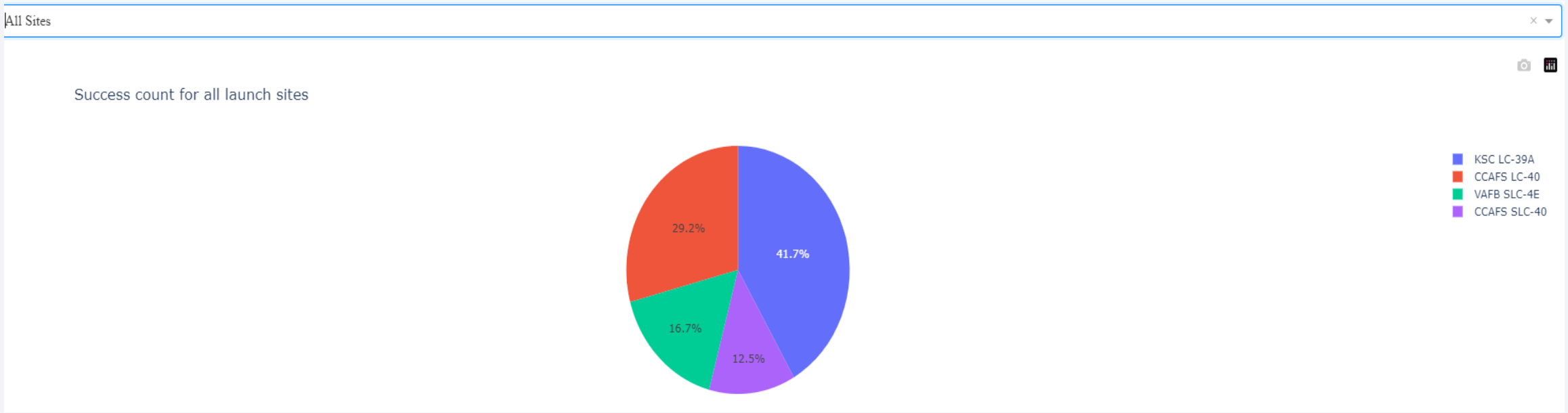
Launch site and its proximity



Launch sites are located near infrastructure such as highways, railways and coastlines. In the other hand it is apparent that it keeps certain distance away from cities. This is understandable due to danger that launches may bring.

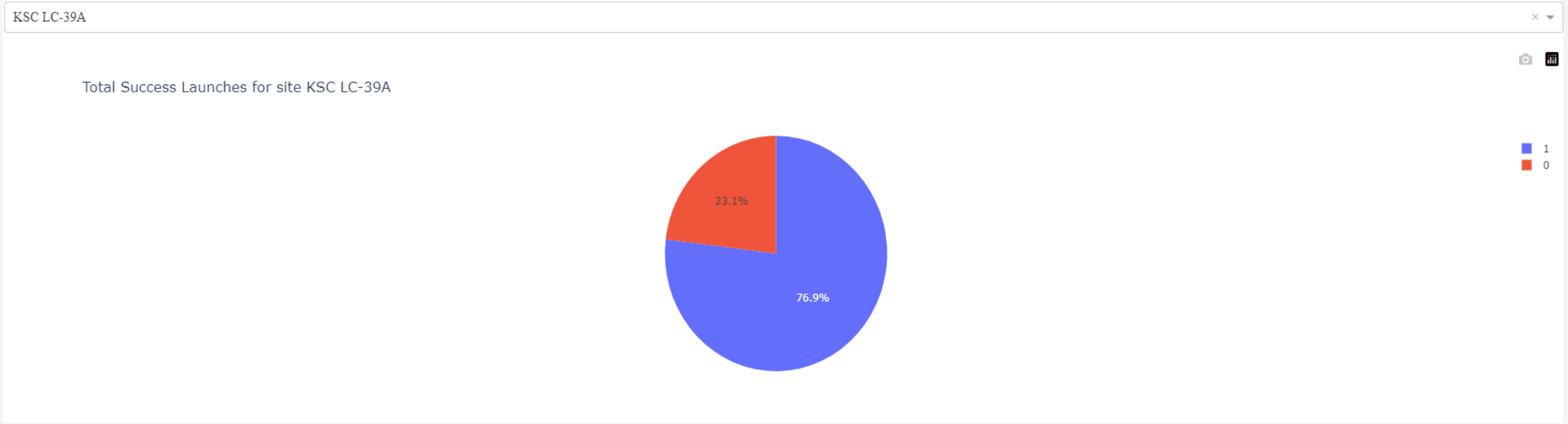

Section 4
Build a Dashboard with Plotly Dash

Percentage of success launch for all locations



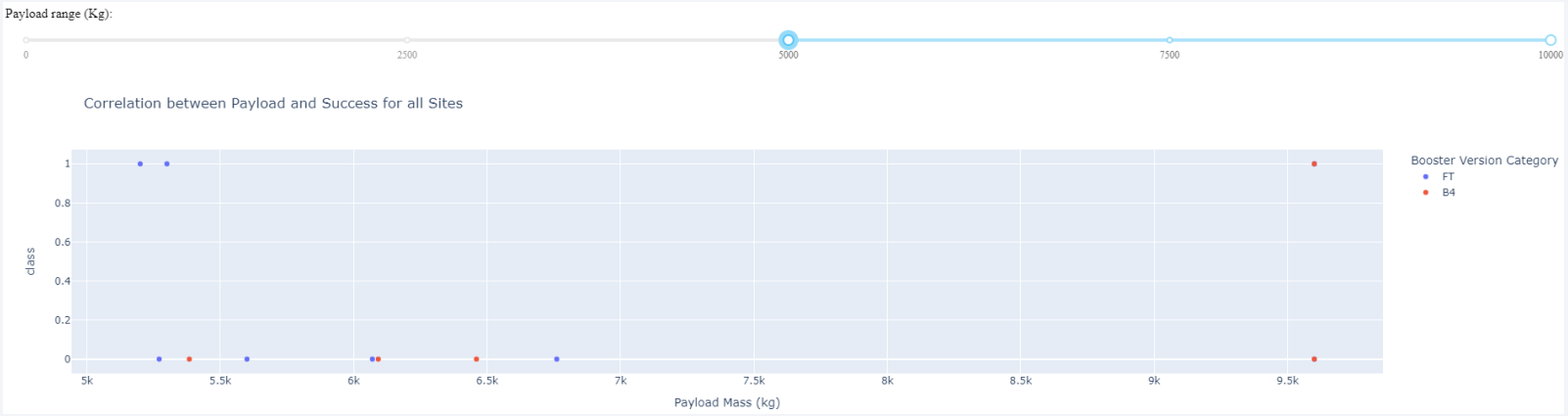
Our pie chart tells us that location KSC LC-39A has a greater percentage of success rate compared to the other locations.

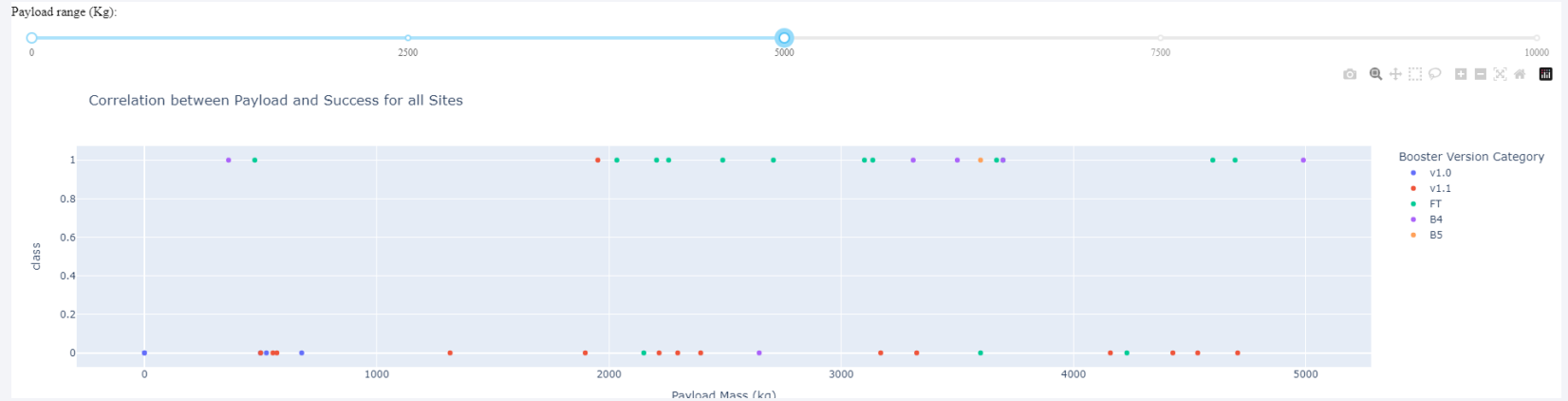
Percentage of success rate for location KSC LC-39A



In a more detailed view of KSC LC-39A we can see that the percentage of success rate for it is 76.9%.

Correlation between Booster Type and Payload





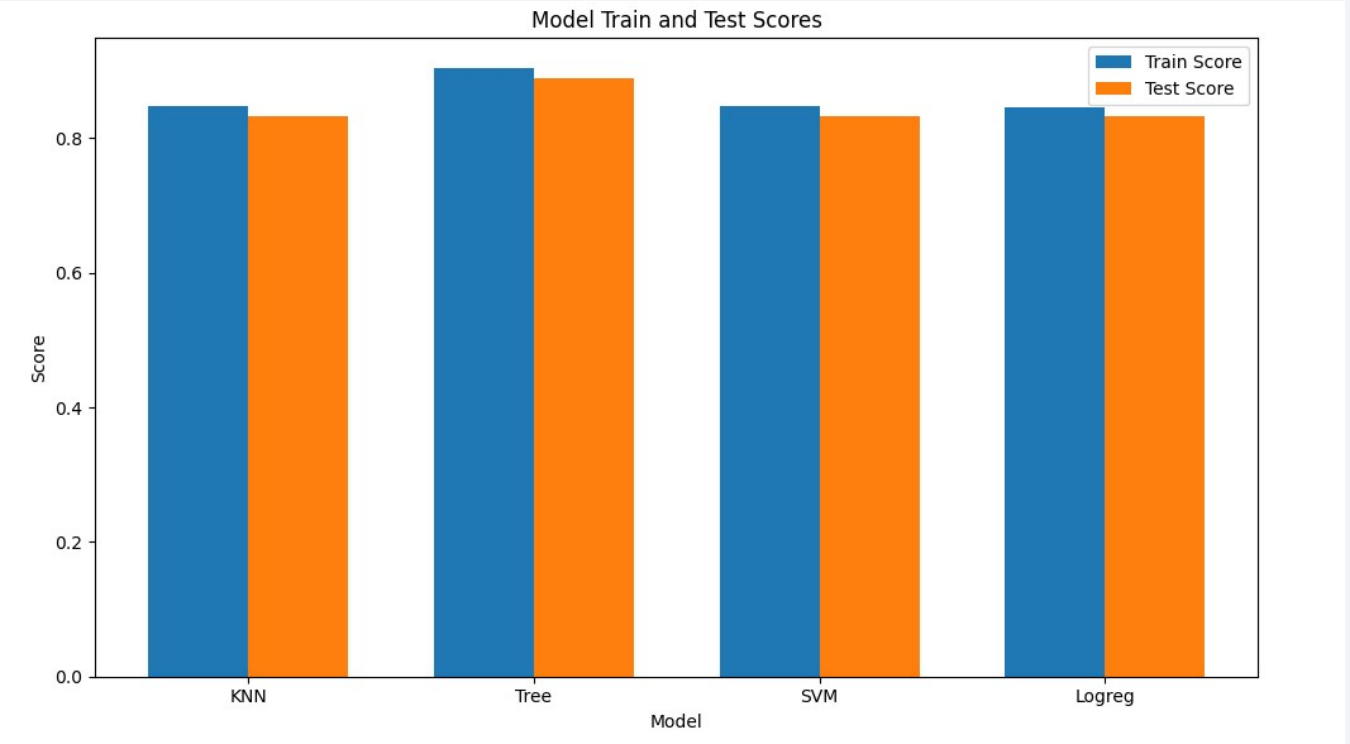
From the graph we can see that on mission with low payload Booster Type FT is more commonly used while on the other hand for mission with higher mass Booster Type B4 is preferred.

Section 5

Predictive Analysis (Classification)

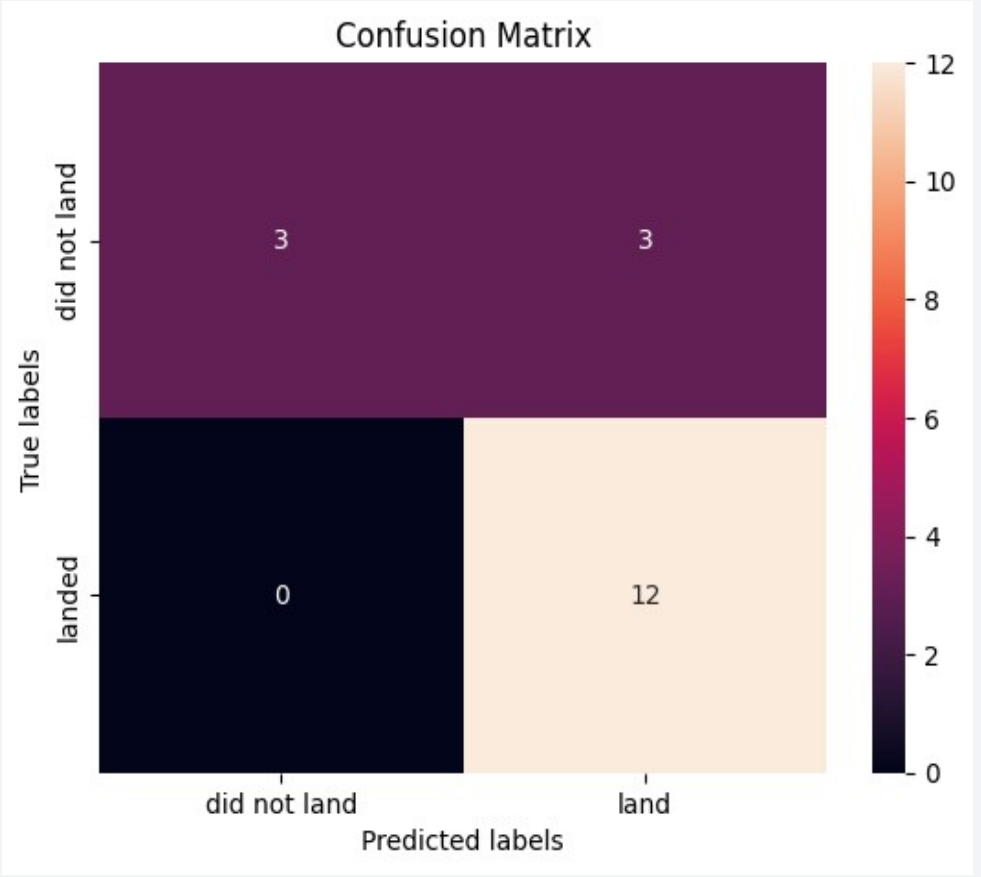
Classification Accuracy

- According to our graph it seem that in overall all models get about the same score on “Train”
- The same thing could be said on accuracy on the “Test”
- If I were to choose, “Tree” would be my choice due to having and slightly higher score on overall.



Confusion Matrix

- Confusion Matrix belonging to “Decision Tree Model”
- In terms of what needs to be check are the ‘False Positive’ given by our model



Conclusions

- Mission success rate seems to be dependent on factors such as orbits, payload, booster type.
- In terms of number we could argue that the higher the number of launches the higher the score will get. This could be attribute to try and error meaning the more we do the more we learn and fix the errors.
- A mission success also seems to be dependent on which orbit are we working on. Some orbits have higher success rate compared to others.
- Payload seems to be a factor as well based on our data payload of greater weight seems to be more successful. Perhaps this is due to the economic value of such and the need of it to be successful.
- Location seem to be a factor as well. Based on our Folium map a greater number of mission occur at one location.
- Taking into consideration all these parameters we developed our model and conclude that a “Decision Tree” will better fit our goal mainly due to its overall higher score.

Thank you!