IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Felipe
October 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API and Web Scraping
  - Data Visualization for EDA
  - SQL for EDA
  - Interactive Map with folium library
  - Dashboard with Plotly Dash library
  - Predictive Analysis via ML techniques
- Summary of all results
  - Exploratory Data Analysis Rresults
  - Interactive Map and Dashboard Results
  - Predictive Analysis Results

Introduction

- Project background and context
  - The question is: Will the Falcon 9 first stage land successfully? This is a pivotal issue since the cost of a launch depends largely on it. SpaceX is able to offer a better price and save much money by reusing the first stage of its rockets. In fact, SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, whereas other providers cost upward of 165 million dollars each. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - How is a (un)succesful landing defined?
  - How do other variables influence the outcome of a landing?
  - How can we optimize the landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Selecting only necessary variables or features
  - One Hot Encoding of categorical variables
- Perform exploratory data analysis (EDA) using visualization and SQL

Methodology

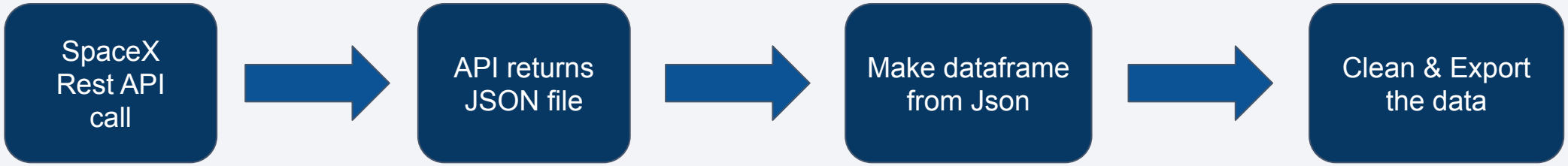
Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
  - Use maps and dashboard to get insights quick insights
- Perform predictive analysis using classification models
  - Fine tune machine learning models with training data, then try them with test data to assess their predictive power

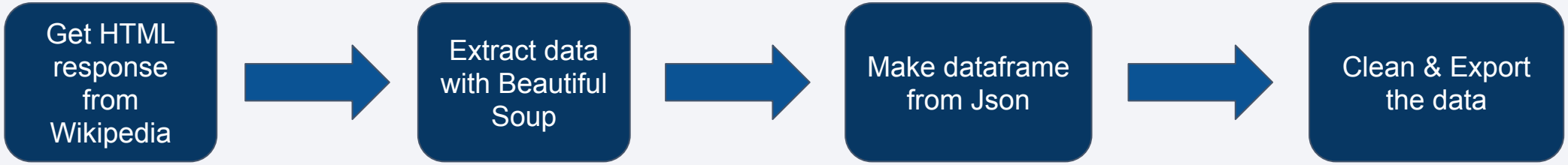
Data Collection

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia

- The Space X REST API URL is api.spacexdata.com/v4



- The Wikipedia URL is: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

1. Get Response from API

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

response = requests.get(spacex_url)
```

2. Convert response to JSON FILE

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

3. Transform data

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

4. Create dictionary with data

```
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}
```

5. Create dataframe

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

6. Filter dataframe

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

7. Export as csv

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

[Link to Github Notebook]

Data Collection – Scraping

1. Get Response from HTML

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

2. Create BeautifulSoup object

```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables

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

4. Get columns names

```
column_names = []

# Apply find_all() function with `th` element
# Iterate each th element and apply the provid
# Append the Non-empty column name (`if name i

for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create dictionary

```
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']=[]
```

6. Add data to keys

```
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:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

7. Create dataframe from dictionary

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

7. Export to file

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

[Link to Github Notebook]

Data Wrangling

- There are lots of instances where the landing outcome is not succesful
  - **Succesful attempts**: True Ocean, True RTLS, True ASDS means the mission has been successful.
  - **Unsuccesful attempts**: False Ocean, False RTLS, False ASDS means the mission was a failure.

- Let us now transform string variables into categorical ones

1. Calculate number of launches for each site

```
df['LaunchSite'].value_counts()

CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
SO        1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

3. Calculate number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

Outcome
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: count, dtype: int64
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes

{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

4. Create landing outcome label from Outcome column

```
landing_class = [0 if x in bad_outcomes else 1 for x in df['Outcome']]
```

5. Export to file

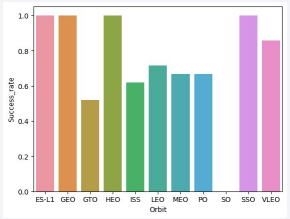
```
df.to_csv("dataset_part_3.csv", index=False)
```

[Link to Github Notebook]

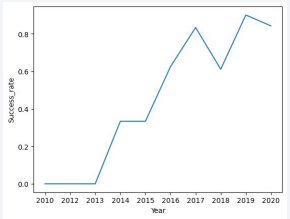
EDA with Data Visualization

- Scatter Graphs [X vs Y]
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Orbit vs. Flight Number
  - Payload vs. Orbit Type
  - Orbit vs. Payload Mass

- Bar Graph [X vs Y]
  - Orbit vs. Success rate



- Line Graph [X vs Y]
  - Year vs. Success rate



Link to Github Notebook

EDA with SQL

We performed 10 SQL queries to gather some basic insights. The complete list is as follows:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first succesful landing outcome in ground pad was acheived.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
9. List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
10. 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 Github Notebook

Build an Interactive Map with Folium

- **The folium map object is a map centered on NASA Johnson Space Center at Houson, Texas**
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name *(folium.Circle, folium.map.Marker)*.
  - Red circles at each launch site coordinates with label showing launch site name *(folium.Circle, folium.map.Marker, folium.features.DivIcon).*
  - The grouping of points in a cluster to display multiple and different information for the same coordinates *(folium.plugins.MarkerCluster).*
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. *(folium.map.Marker, folium.Icon).*
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them . *(folium.map.Marker, folium.PolyLine, folium.features.DivIcon)*
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings

Link to Github Notebook

Build a Dashboard with Plotly Dash

- **Dashboard has dropdown, pie chart, rangeslider and scatter plot components**
  - Dropdown allows a user to choose the launch site or all launch sites *(dash_core_components.Dropdown).*
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component *(plotly.express.pie).*
  - Rangeslider allows a user to select a payload mass in a fixed range *(dash_core_components.RangeSlider).*
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass *(plotly.express.scatter)*

Link to Github Notebook

Predictive Analysis (Classification)

- **Data preparation**
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- **Model preparation**
  - Selection of machine learning algorithms
  - List of hyperparameters to try
  - Get hyperparameters for each algorithm with GridSearchCV
  - Training GridSearchModel models with training dataset to obtain parameters
- **Model evaluation**
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- **Model comparison**
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen

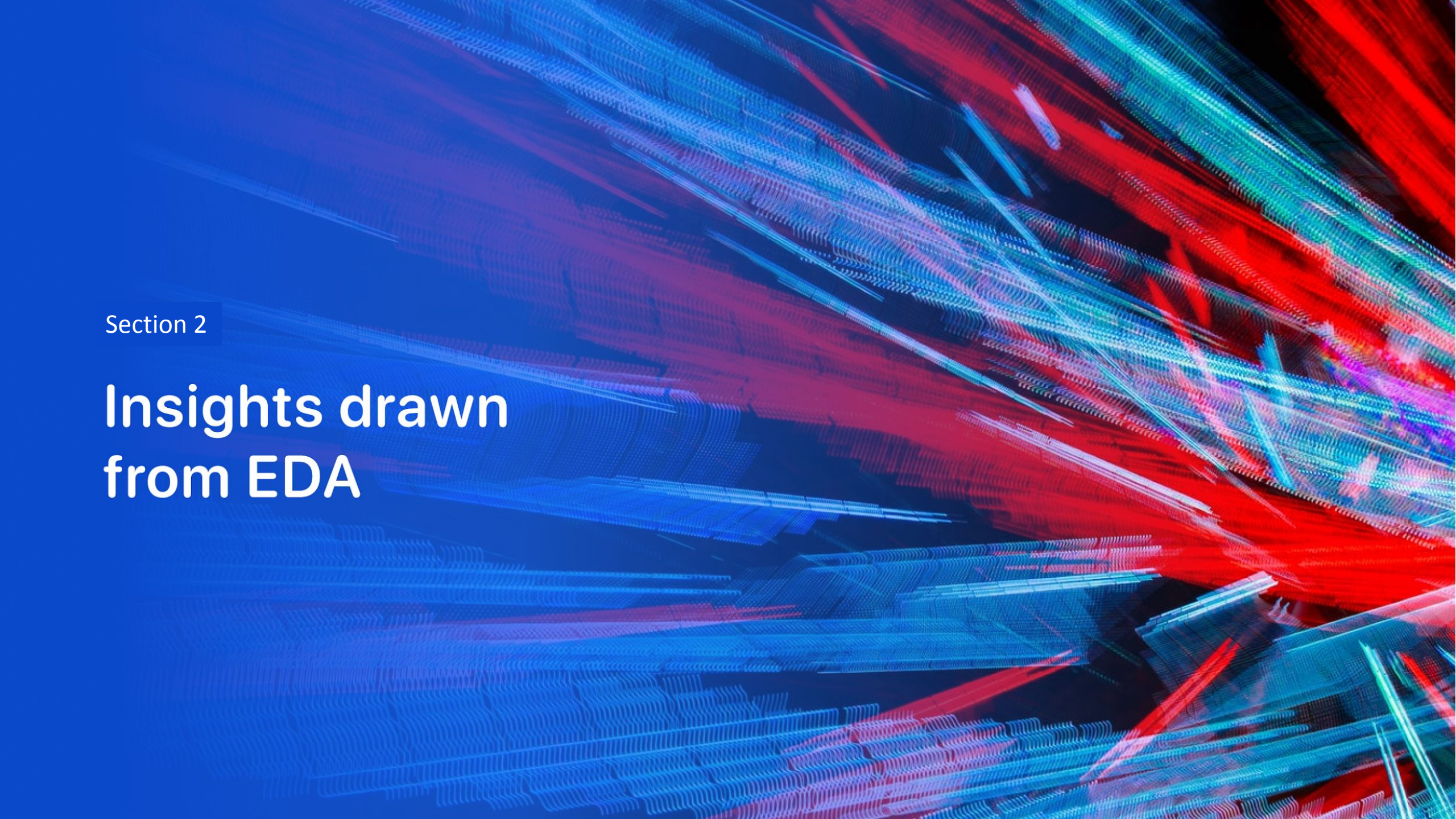
Link to Github Notebook

Results

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

Section 2

Insights drawn from EDA

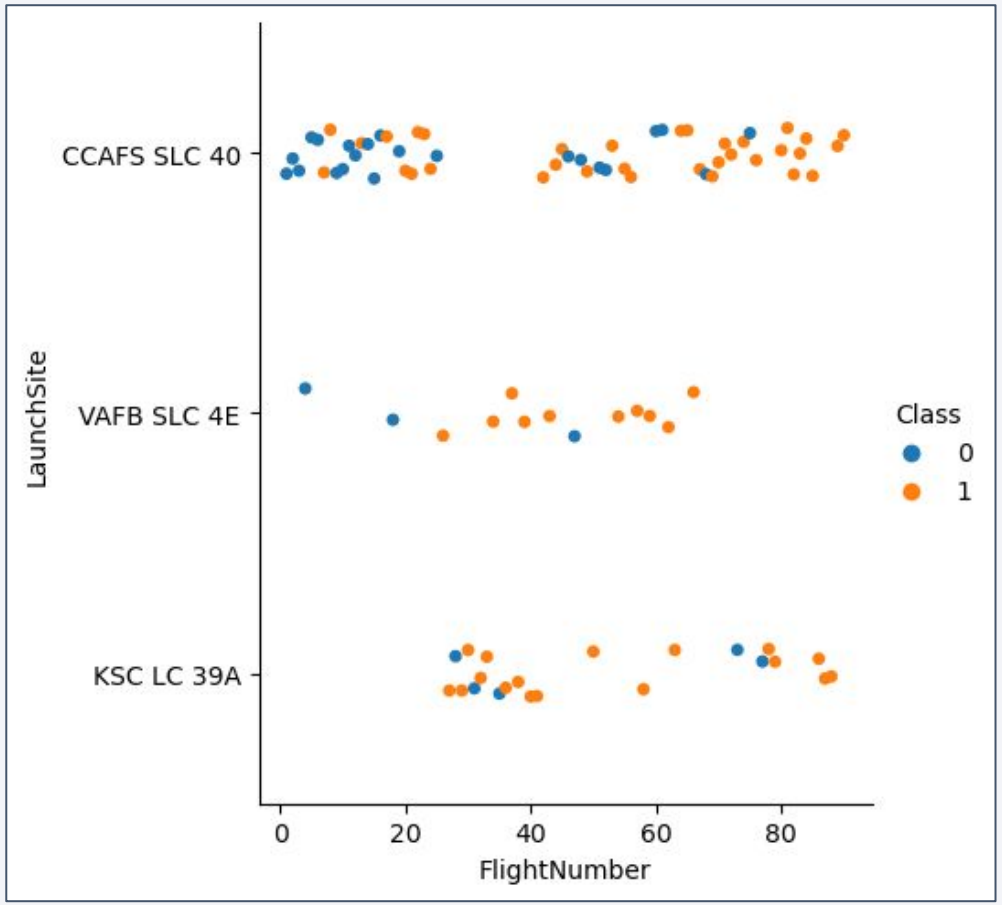


Flight Number vs. Launch Site

Observations

- As the flight number increases the success rate tends to increase this is particularly for launch site *ccafs slc 40*

- Around flight number 70 the success rate noticeably high

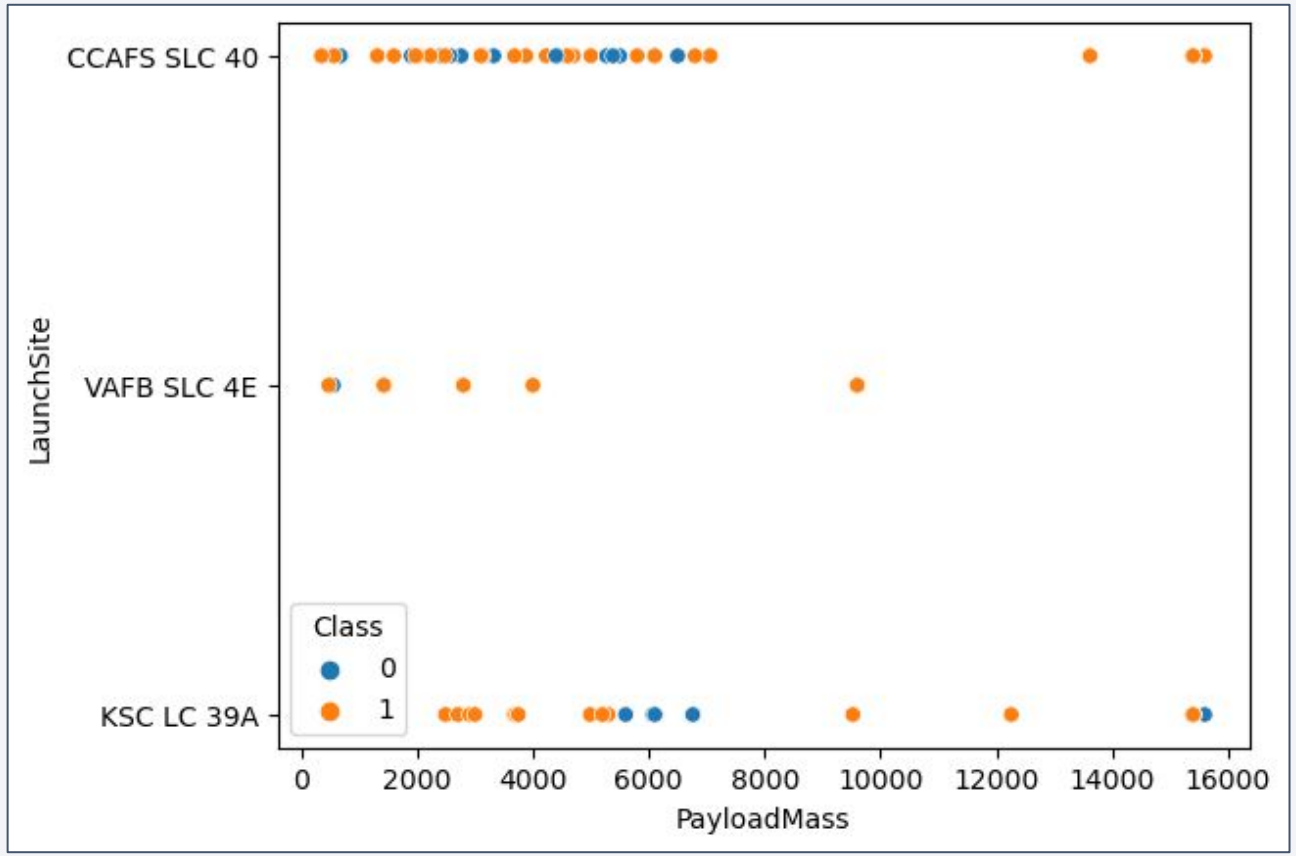


Payload vs. Launch Site

Observations

- Launch site *ccafs slc 40* seems heavily dependent on a high Payload mass to get a high success rate, of around 7000 kg.

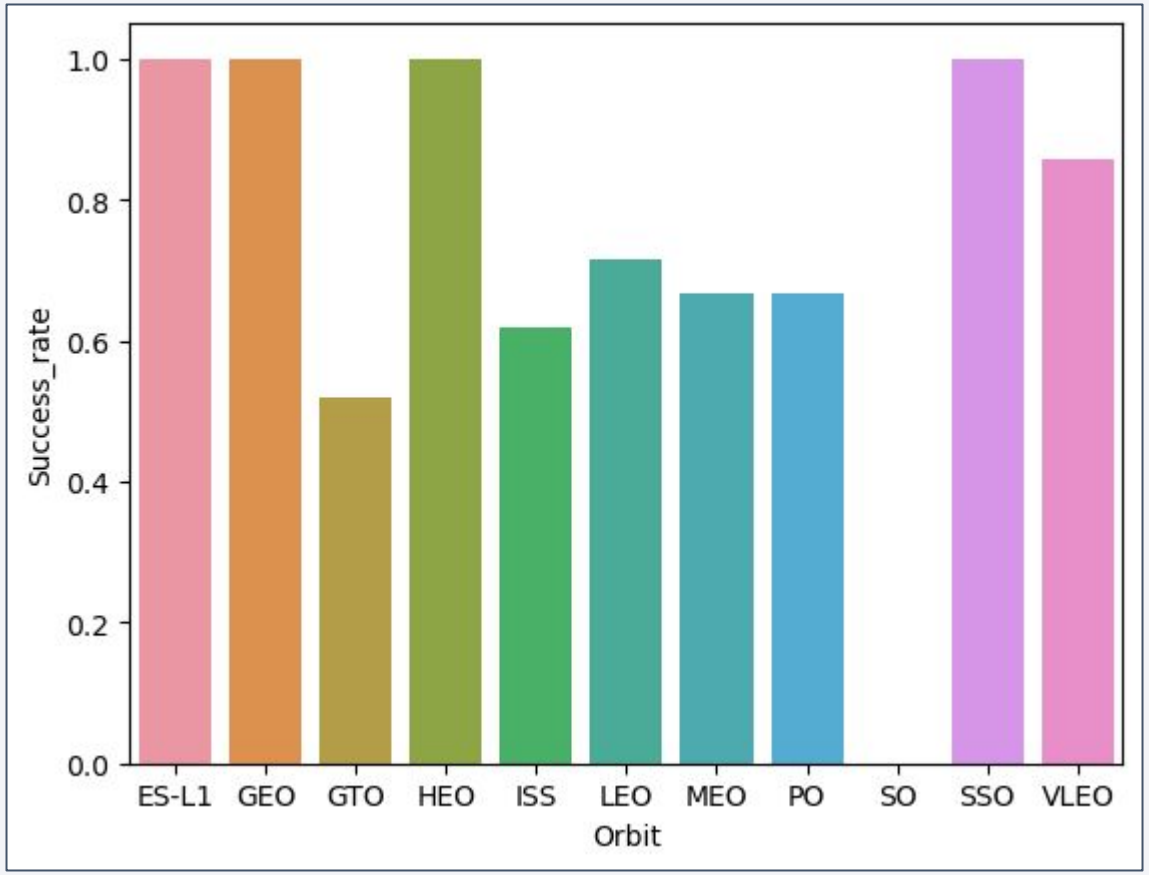
- That is not the case for the other launch sites whose success rates are good, both with low and high payload mass.



Success Rate vs. Orbit Type

Observations

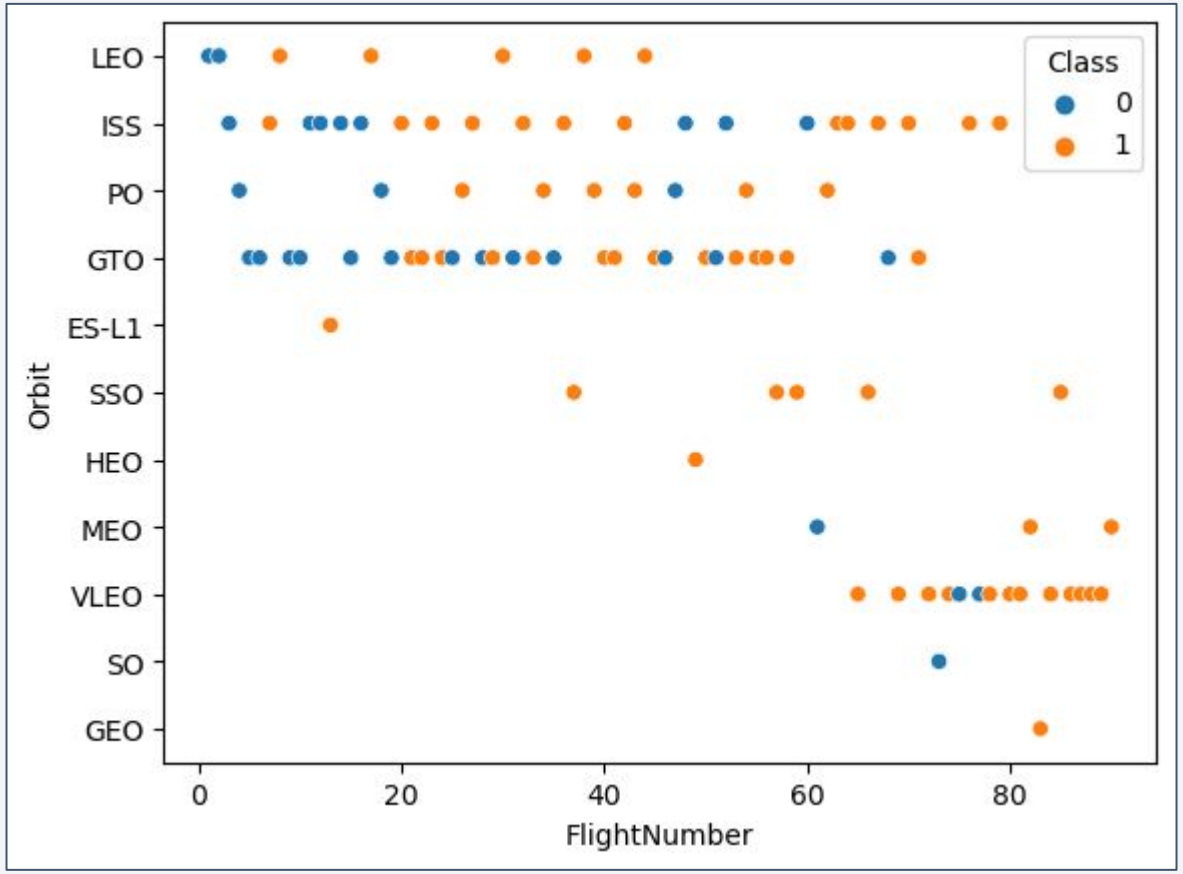
- It is obvious from the chart that the orbits with the highest success rate are *ES-L1, GEO, HEO, SSO,* with a perfect score.



Flight Number vs. Orbit Type

Observations

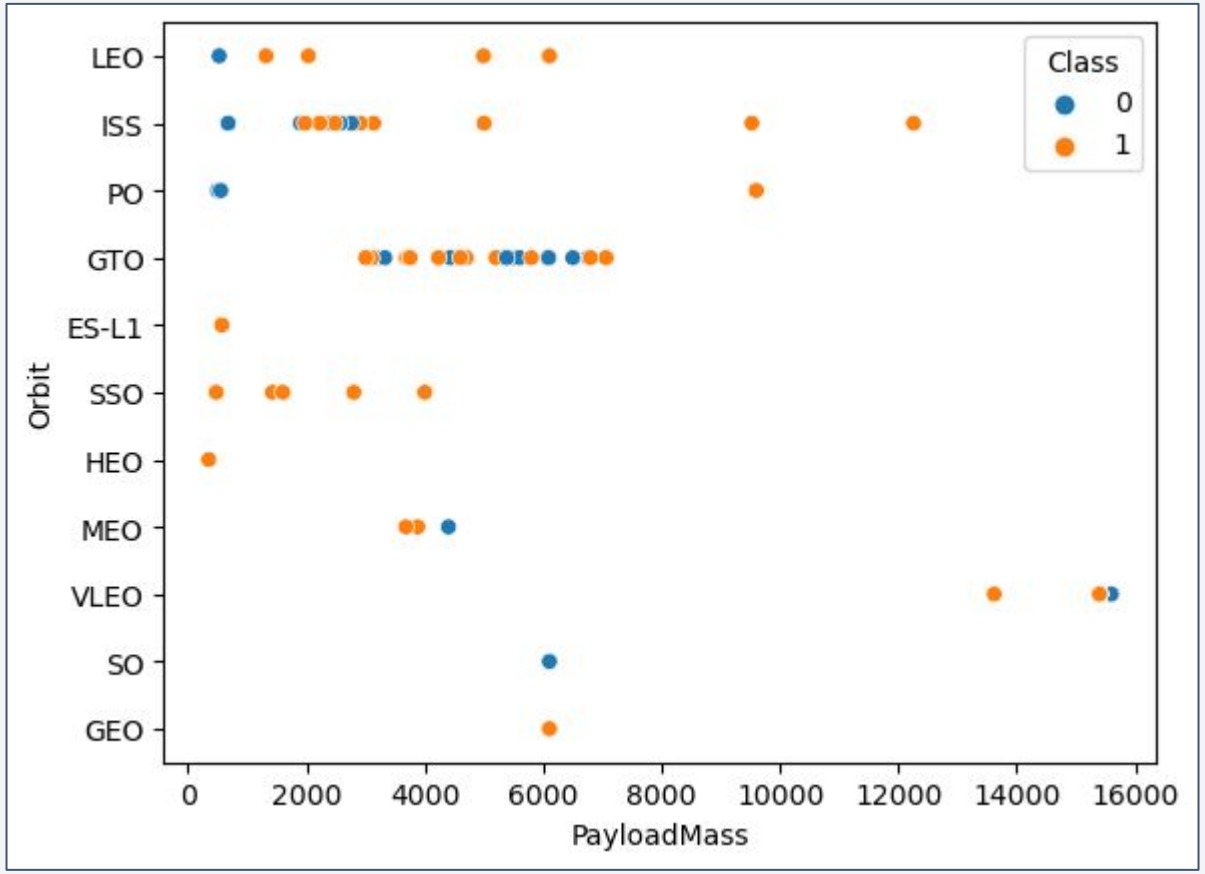
- As previously observed in another chart, the success rate increases as the number of flights increases.
- That's very obvious for the *LEO* orbit, to some extent for *PO* and *ISS* orbits, whereas it's pretty much absent for *GTO* orbit.
- The launches to the other orbits seem to have benefited from previous experience.



Payload vs. Orbit Type

Observations

- The *LEO* orbit improves its success rate as the payload increases
- The same applies to some extent to *ISS* orbit
- The *GTO* orbit success rate is equally mixed everywhere.

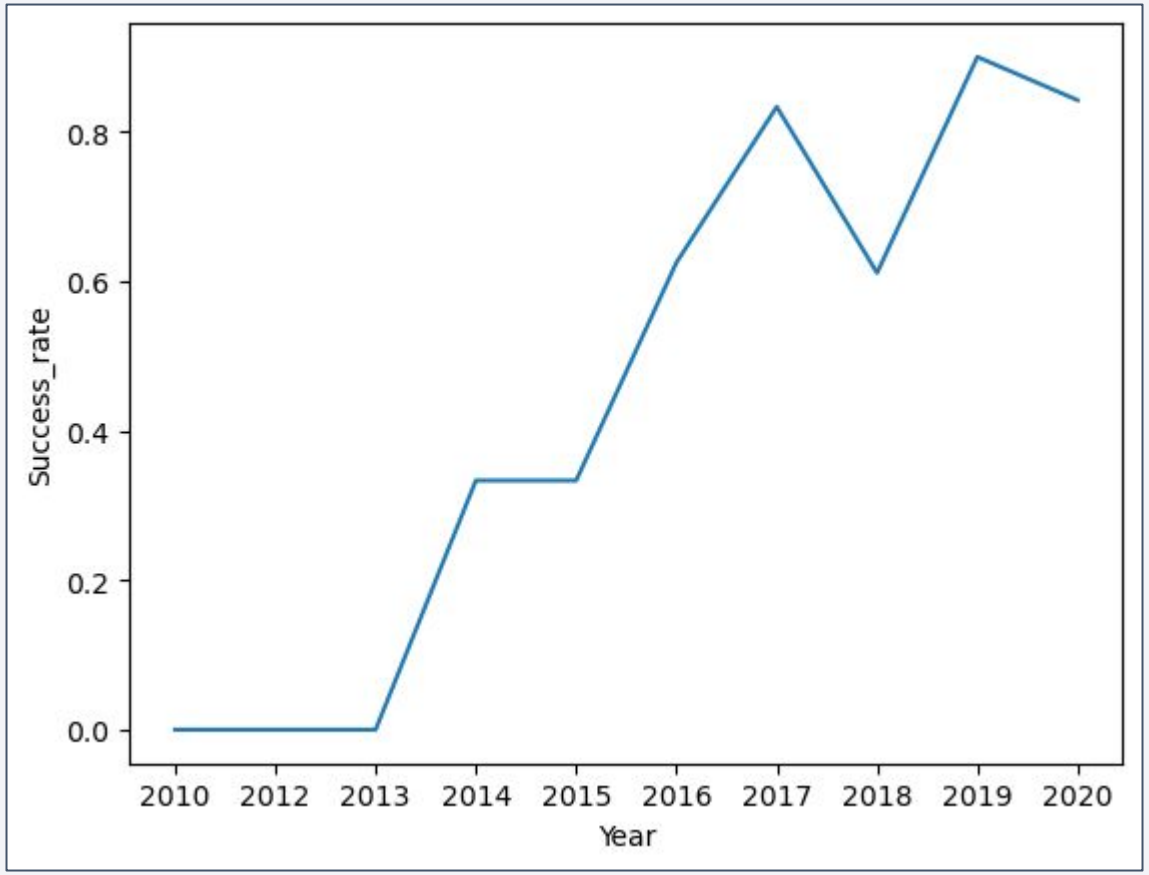


Launch Success Yearly Trend

Observations

- The trend since 2013 is markedly positive, reaching a first peak of over 80% in 2017 and a second of over 90% in 2019.

- There was a setback in 2018 when the success rate fell to under 60%



All Launch Site Names

QUERY

```
%sql select distinct(Launch_Site) from spacextable
```

Explanation

- Using the SQL language we ask the computer to give us every launch site in the “Launch_Site” column without repeating their names.

Unique Launch Sites

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

Launch Site Names Begin with 'CCA'

QUERY

```
%sql select * from spacextable WHERE launch_site like '%CCA%' limit 5
```

Launch Sites beginning with ‘CCA’

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

Explanation

- Using the SQL language we ask the computer to give us the first five rows where the “Launch_Site” column starts with ‘CCA’.

Total Payload Mass

QUERY

```
%%sql select sum(payload_mass__kg_) as [Total Payload Mass (Kg)] from spacextable
where "Customer" = 'NASA (CRS)'
```

Total Payload Mass

Total Payload Mass (Kg)
45596

Explanation

- Using the SQL language we ask the computer to give us the total payload mass in Kg carried by boosters launched by NASA (CRS).

Average Payload Mass by F9 v1.1

QUERY

```
%%sql select avg(payload_mass__kg_) as [Average Payload Mass (Kg)]
from spacextable
where booster_version = 'F9 v1.1'
```

Average Payload Mass by F9 v1.1

Average Payload Mass (Kg)
2928.4

Explanation

- Using the SQL language we ask the computer to give us the average payload mass in Kg carried by F9 v1.1 boosters.

First Successful Ground Landing Date

QUERY

```
%%sql select min(date) as [Date of First Succesful Landing Outcome (ground pad)]
from spacextable
where landing_outcome = 'Success (ground pad)'
```

Date

Date of First Succesful Landing Outcome (ground pad)
2015-12-22

Explanation

- Using the SQL language we ask the computer to give us the date when the outcome of a landing on a ground pad was successful for the first time.

Successful Drone Ship Landing with Payload between 4000 and 6000

QUERY

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

List of Booster Versions

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

Explanation

- Using the SQL language we ask the computer to give us the of Booster Versions which have landed successfully on drone ships with payload between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

QUERY

```
%%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTABLE WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS,
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTABLE WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Success and Failure

SUCCESS	FAILURE
100	1

Explanation

- Using the SQL language we ask the computer to create two columns summarizing the outcomes of missions as success or failure.

Boosters Carried Maximum Payload

QUERY

```
%%sql select distinct(booster_version) from spacextable
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextable)
```

List of Booster Versions

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

Explanation

- Using the SQL language we ask the computer to list the booster version having carried the maximum payload.

2015 Launch Records

QUERY

```
%%sql select substr(Date,6,2) as Month, landing_outcome, booster_version, launch_site from spacextable
where substr(Date,1,4)='2015'
and landing_outcome like '%Failure%'
order by Month
```

List of Booster Versions

Month	Landing_Outcome	Booster_Version	Launch_Site
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

Explanation

- Using the SQL language we ask the computer to show us the month, landing outcome, booster version and launch site in 2015 when the landing outcome was a failure.

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

QUERY

```
%%sql select(select count(*) from spacextable
            where Landing_Outcome like '%Success%'
            and Date between '2010-06-04' and '2017-03-20') as Success,

(select count(*) from spacextable
 where Landing_Outcome like '%Failure%'
and Date between '2010-06-04' and '2017-03-20') as Failure
```

List of Booster Versions

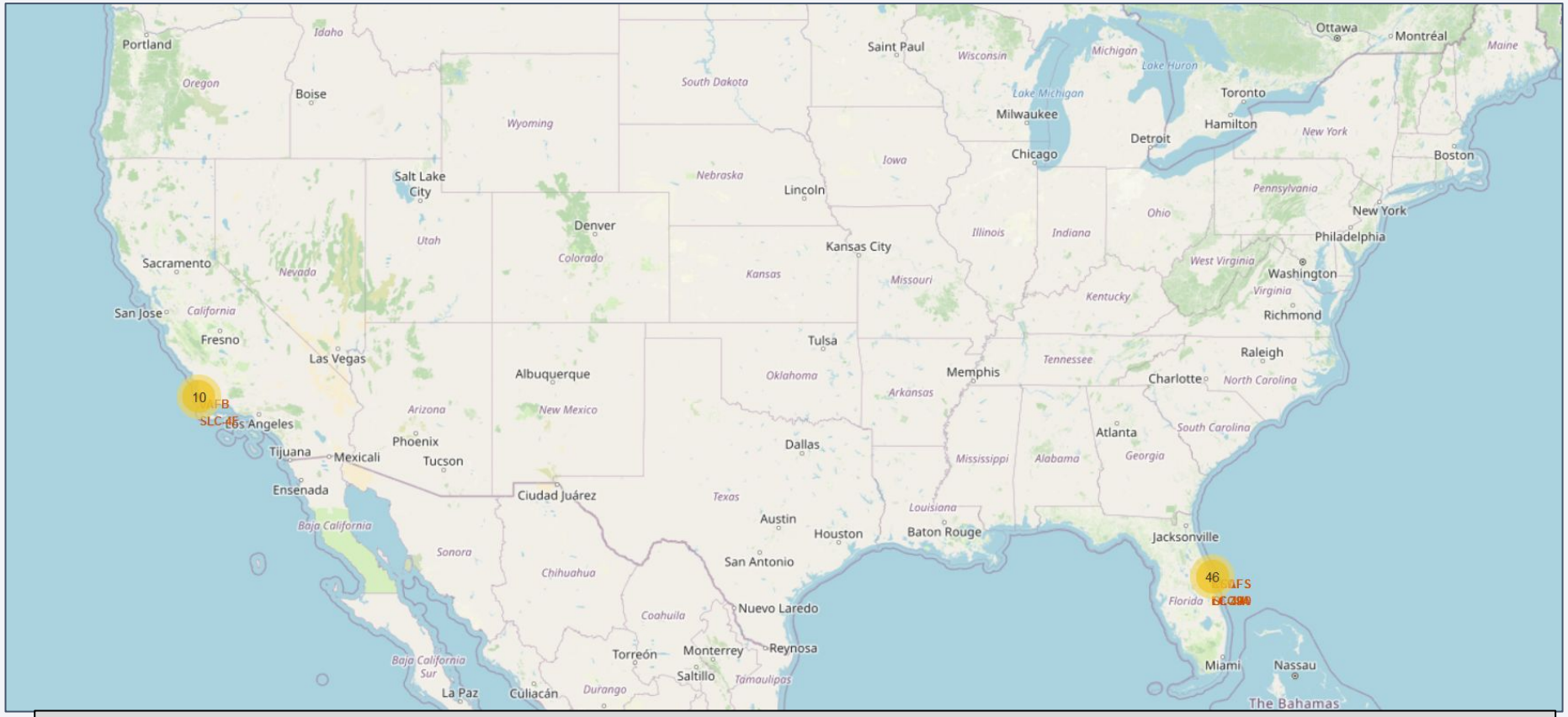
Success	Failure
10	6

Explanation

- Using the SQL language we ask the computer to create two columns summarizing the outcomes of landings as success or failure, for a time frame between 04/06/2010 and 20/03/2017

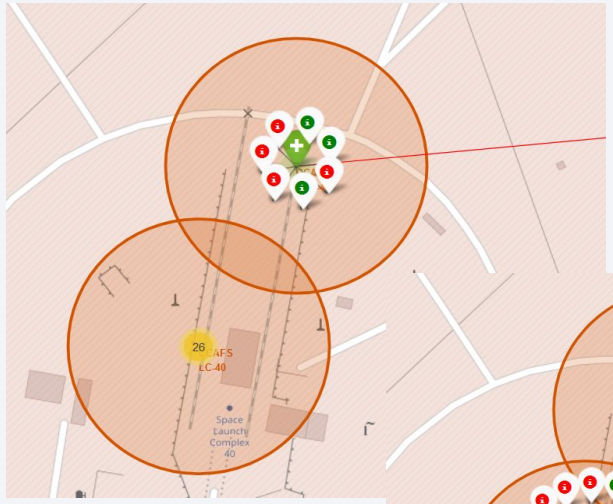

Section 3
Launch Sites
Proximities Analysis

Map–Launch Sites

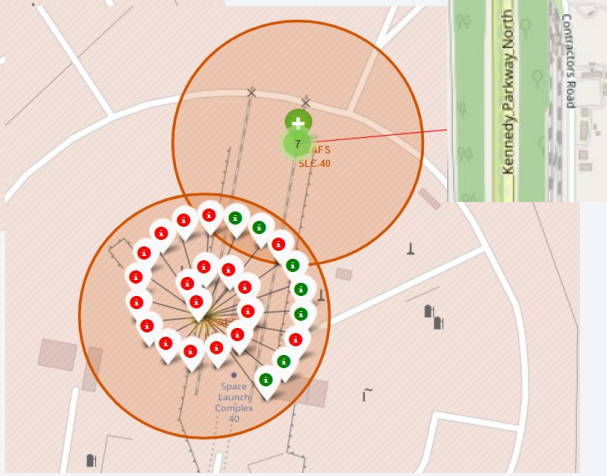


Spacex has launch sites on both coastlines but three of four are located on the east coast in Florida. The other is in South California.

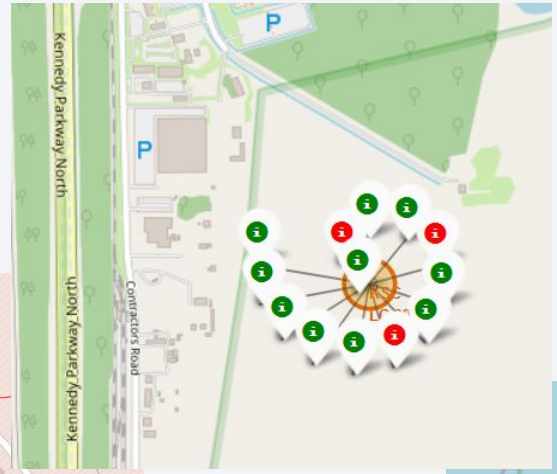
Map–Color-labeled launch outcomes



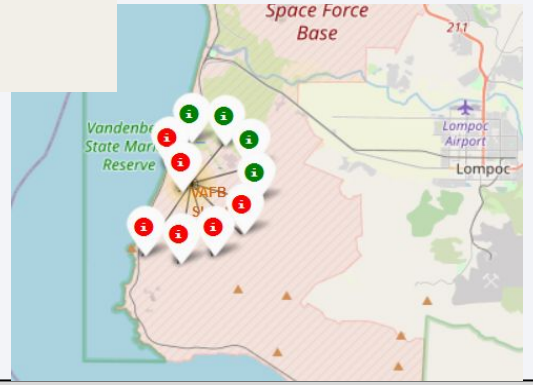
1



2



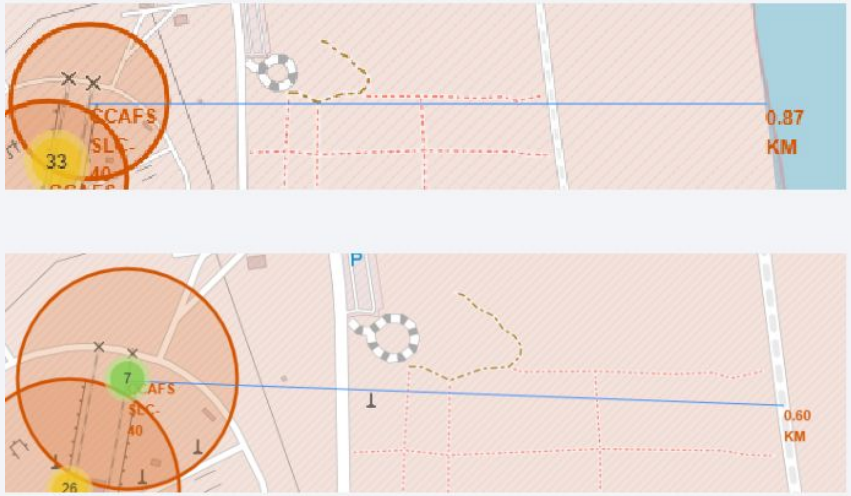
3

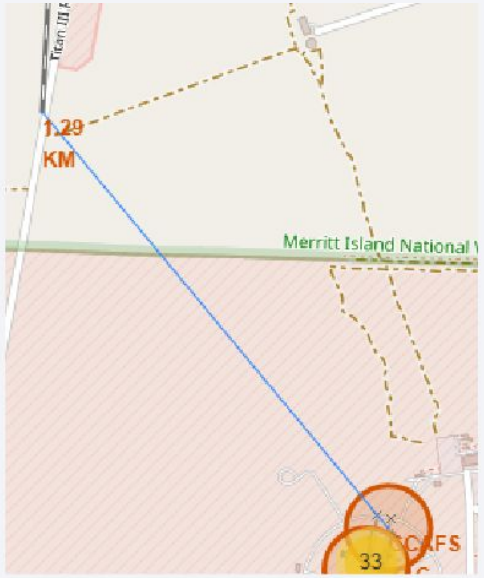


4

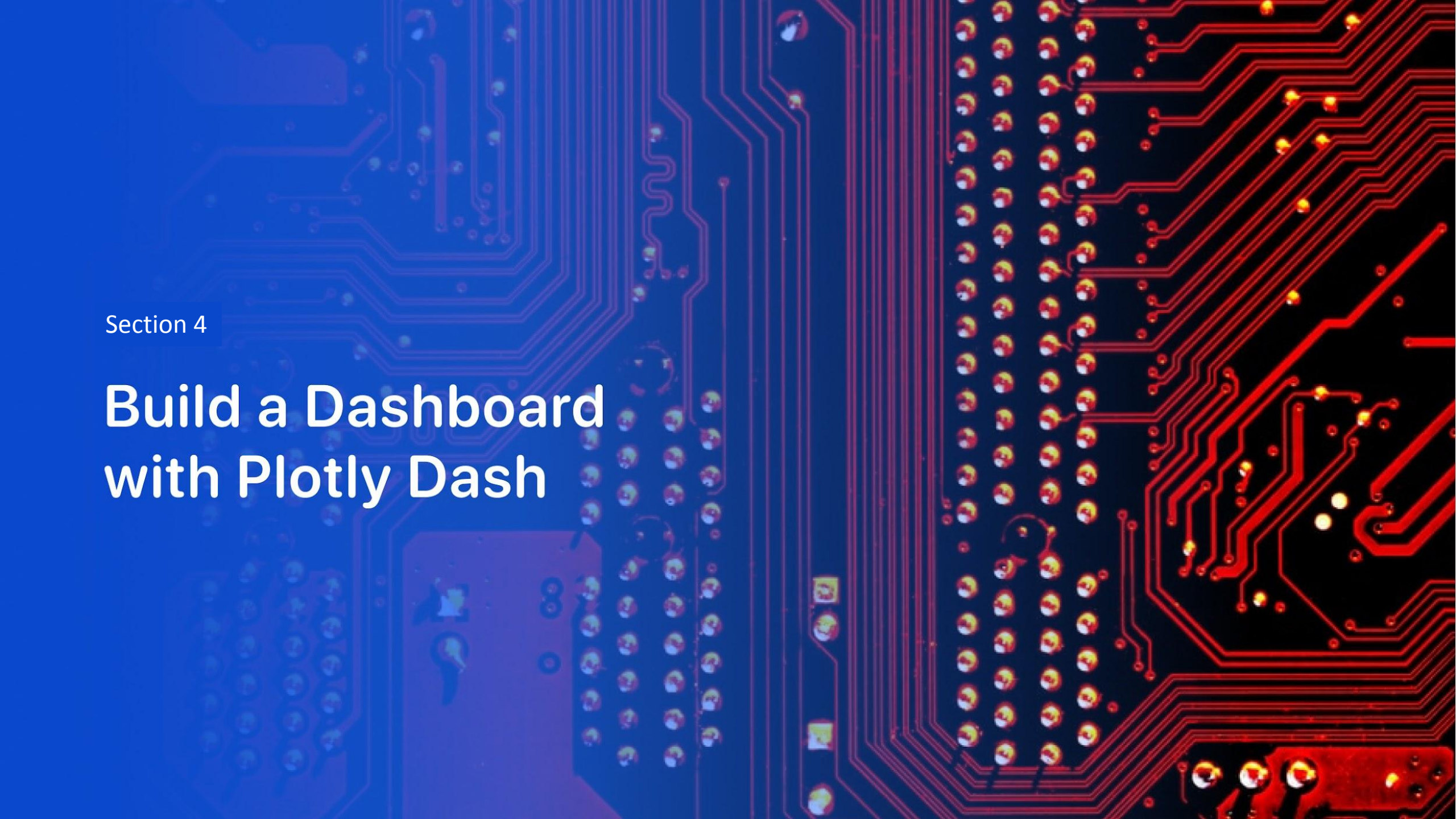
Green represents successful launches whereas red represents unsuccesful ones. 1) ccafs slc-40, 2) ccafs lc-40, 3) KSC LC 39-A, 4) VAFB SLC-4E. Clearly, KSC LC 39-A has the best success rate.

<Folium Map Screenshot 3>

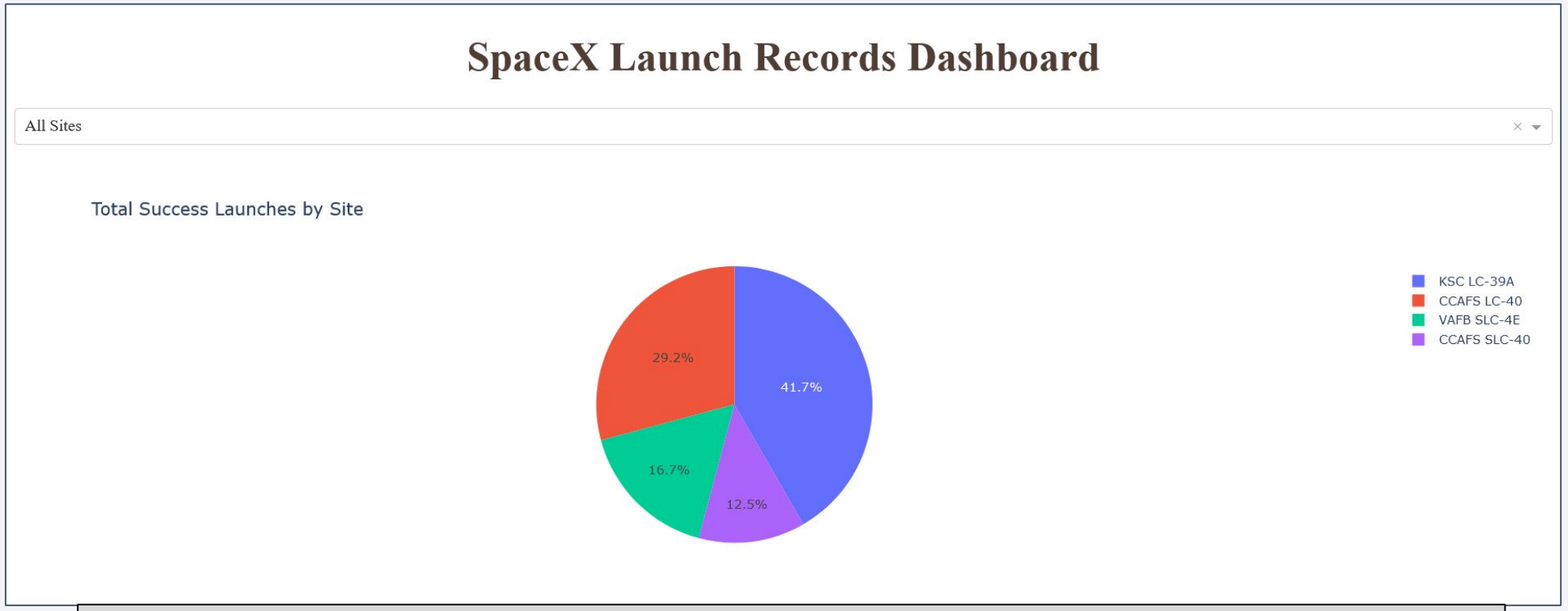




Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

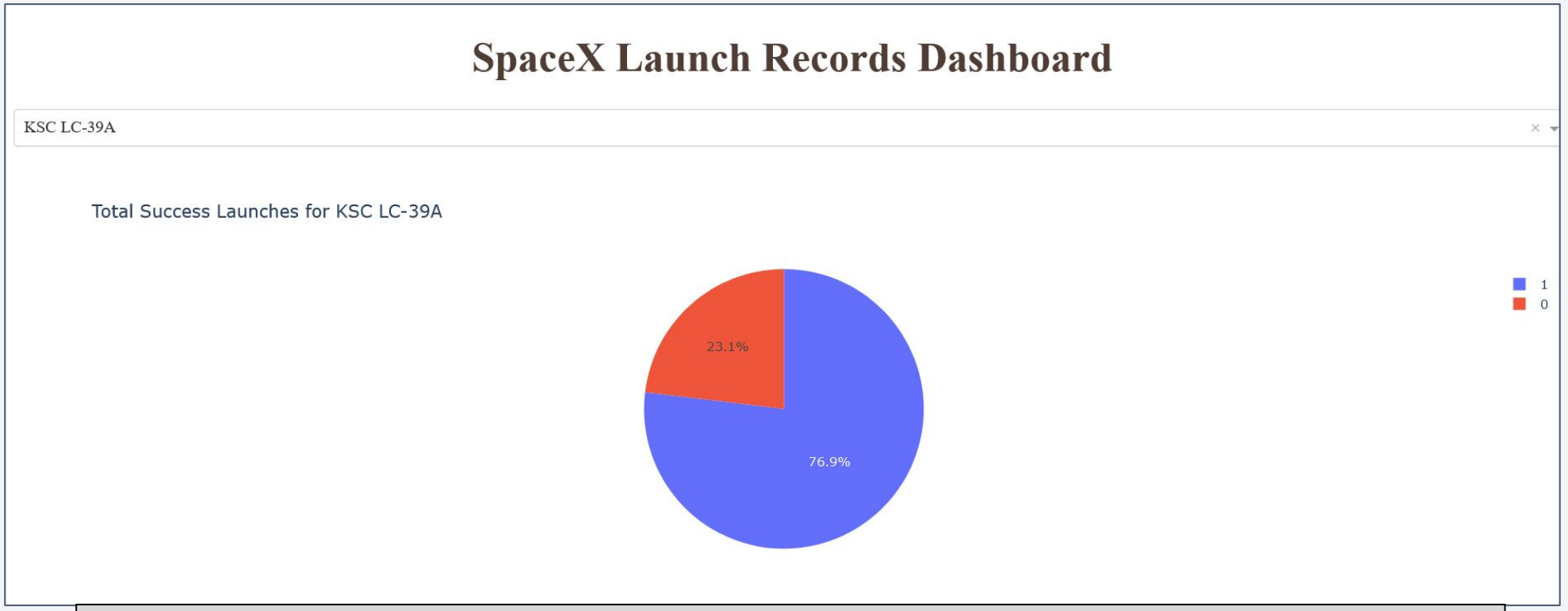

Section 4
Build a Dashboard with Plotly Dash

Total Success Launches for all Sites



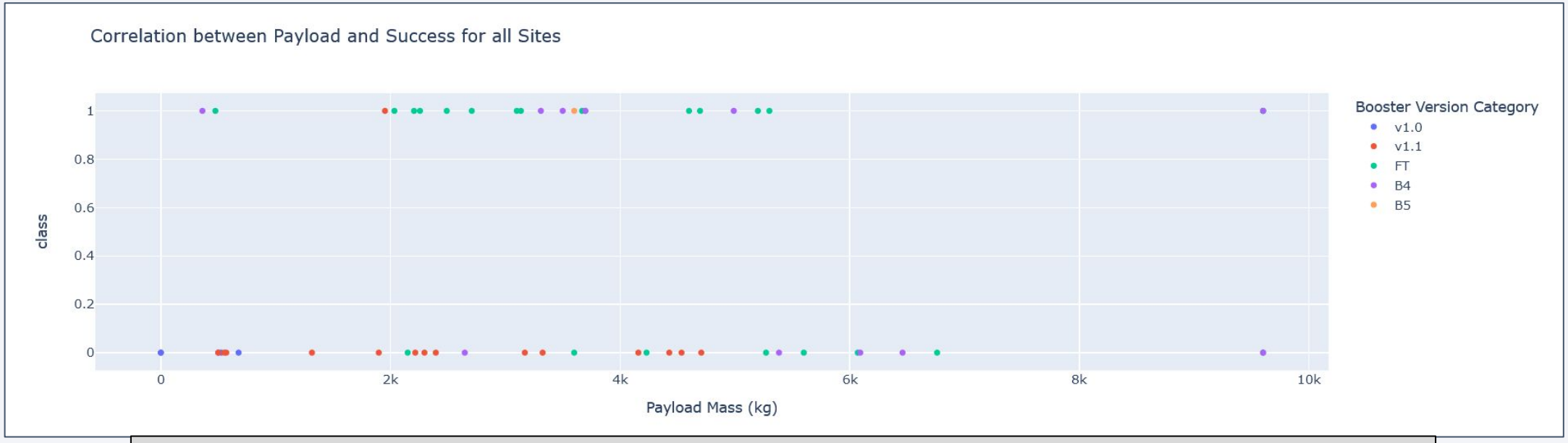
According to the pie chart, the KSC LC-39A launch site has the highest participation in successful launches of all four, with a score 41.7%. This is followed in second place by CCAFS LC-40, with 29.2%.

Total success launches for Site KSC LC-39A



Now, 76.9% of the launches from the KSC LC-39A site have had success and 23.1% of them have failed.

Correlation between Payload and Success for all Sites



Up to a payload mass of about 6,000 kg there is no clear correlation between success and failure, but beyond that point failure becomes the norm.

Section 5

Predictive Analysis (Classification)

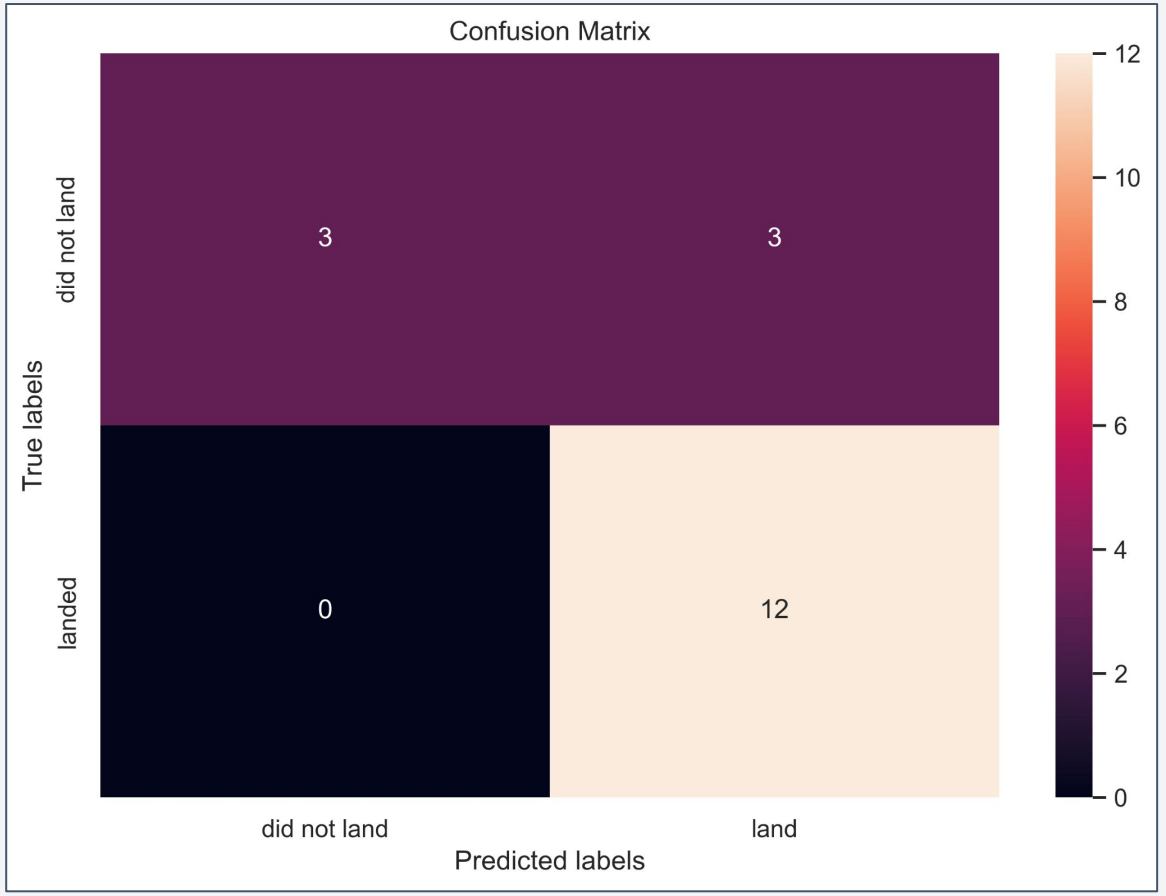
Classification Accuracy

- All of the models have the same accuracy score on the test data. Things are a bit different with the train data, though.
- The SVM model gets the best score using the train data, but not by much.

Confusion Matrix

- Since the accuracy score on the test data is identical for every model, it is no surprise that the confusion matrix is likewise identical for all of them.
- It turns out these models have no false negatives, but do have three false positives each. That's their major problem, It is, however, far from fatal.

TN	FP
FN	TP



Conclusions

- As more experience is gained the launch success rate tends to increase
- Beyond payload mass of 6000 kg, failure is almost certain.
- HEO, GEO, SSO, ES-L1 are the orbits with highest success rate.
- Still some orbits seem to depend more on the payload mass than the rest, that is true for LEO in particular.
- Some launch sites have a far better success rate than others, but we cannot make a confident guess based on the data available.
- The Support vector Machine algorithm turned to be the best on the train data and as good as any other on the test data at predicting the launch outcome based on some previously chosen variables.

Thank you!