IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alexander Leano
19/03/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection via API and Web Scraping
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and Dashboard
  - Predictive results

Introduction

- Project background and context
  - The project aims to predict if the Falcon 9 first stage will successfully land.
  - Falcon 9 rocket launch cost 62 million dollars when other providers cost upward of 165 million dollars each.
  - The price difference is explained by the fact that SpaceX can reuse the first stage.
  - By determining if the stage will land, It is possible to determine the cost of a launch.
  - This information is interesting for another company if it wants to compete with SpaceX for a rocket launch.
- Problem to be solved
  - Identification of factors that influence the landing outcome.
  - Find the relationship between variables and how they affect the outcome.
  - Increase the probability of landing success by finding the best condition needed.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request data from the SpaceX API (https://api.spacexdata.com/v4/rockets/).
  - Web scrapping from Wikipedia.
- Perform data wrangling
  - Dropped unnecessary columns and labeled them for classification 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
  - Find the best hyper-parameters, compare the different models

Data Collection

- Dataset collected by REST API:
  - URL: https://api.spacexdata.com/v4/rockets/
  - Using of get request, decode the response content as JSON and turn it into a Pandas data frame using the json_normalize() method. Cleaned data, checked and filled in missing values that were needed.
- Web scraping:
  - List of Falcon 9 and Falcon Heavy launches (9th June 2021). URL: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
  - Use of BeatifulSoup for extraction of record as HTML table, parsing and converting to a Pandas data frame for further analysis

Link to workbook document

Data Collection – SpaceX API

1. Getting a response from API

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

2. Convert response to JSON file

```
data=pd.json_normalize(response.json())
```

3. Convert response to JSON file

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

4. Create a 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 a dataframe

```
data=pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Export to file

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

[Link to workbook document]

Data Collection - Scraping

1. Getting a response from HTML

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

2. Create a BeautifulSoup Object

```
soup = BeautifulSoup(response.text, 'html.parser')
```

3. Find all tables

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

4. Get column names

```
column_names=[]
# Apply find_all() function with `th` element on
all_th = first_launch_table.find_all('th')

# Iterate each th element and apply the provided
for row in all_th:
    try:
        name = extract_column_from_header(row)
        # Append the Non-empty column name (`if r
        if (name is not None and len(name)>0):
            column_names.append(name)
    except:
        pass
```

5. Create a dictionary with data

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each
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('tabl
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as numb
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
```

See the notebook for the rest of code

7. Create dataframe

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

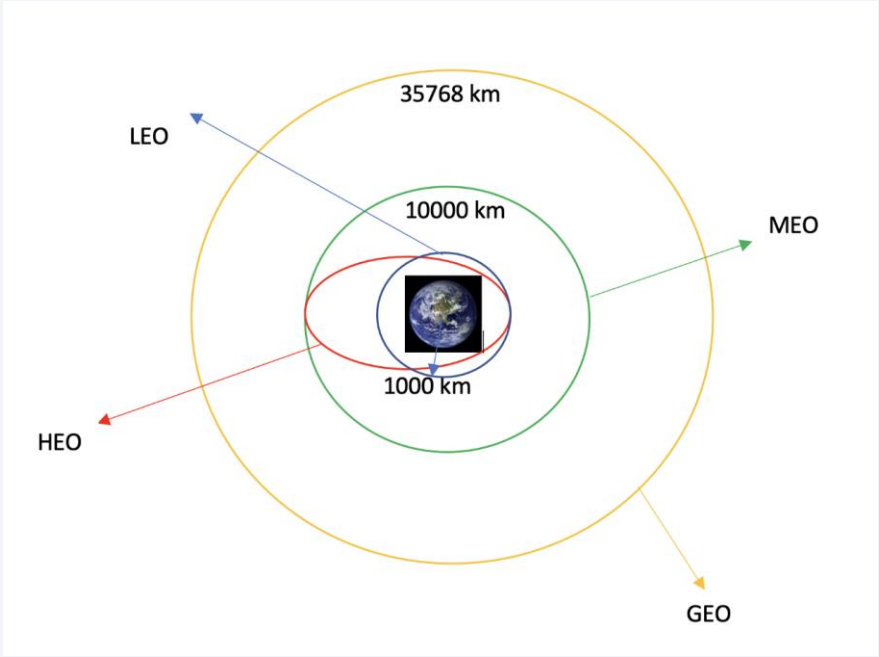
7. Export to file

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

Link to workbook document

Data Wrangling

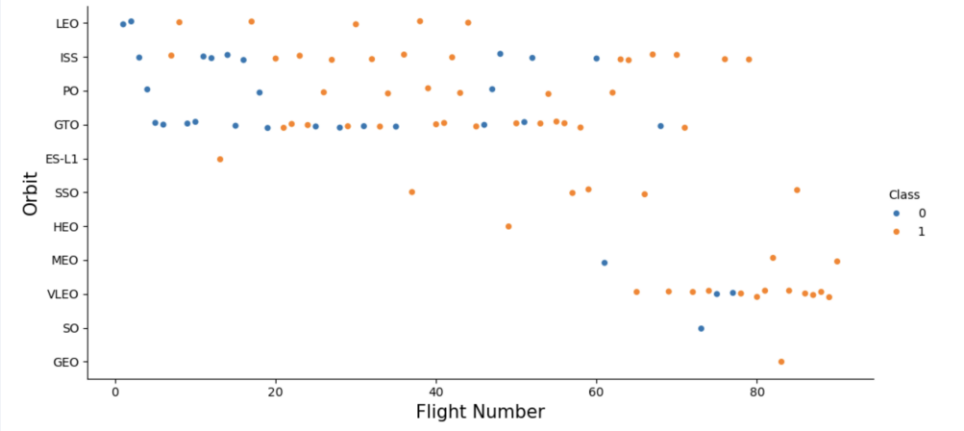
- Data Wrangling is the process of cleaning and unifying complex data sets for easy access and Exploratory Data Analysis (EDA)
- There were several cases where the booster did and did not land successfully:
  - Success: True Ocean, True RTLS, True ASDS
  - Failure: False Ocean, False RTLS, False ASDS
- It was necessary to transform the outcome launch into a new categorical variable.
- Some exploratory analysis was performed to have a better understanding of the occurrence of each orbit.

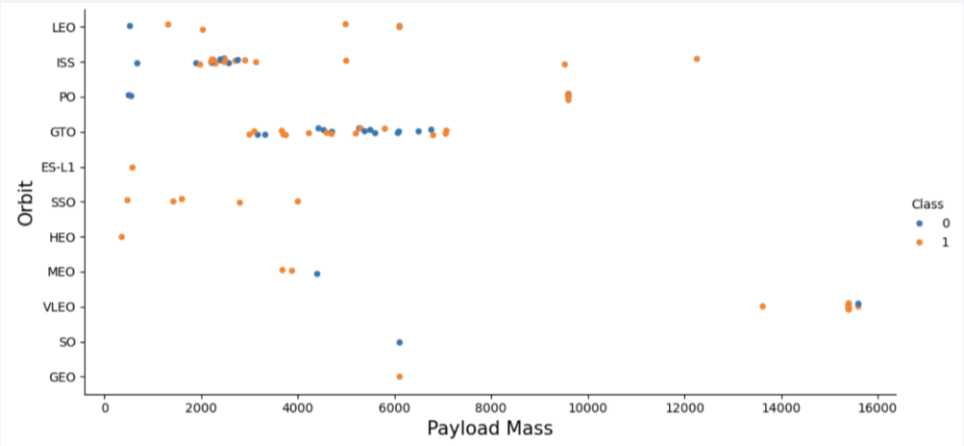


Link to workbook document

EDA with Data Visualization

- Different scatter plots were graphed to find the relationship between variables:
  - Payload and Flight Number
  - Launch Site and Flight Number
  - Payload and Launch Site
  - Orbit and Flight Number
  - Success Rate and Orbit type
  - Orbit Type and Payload Mass
- Scatter plots show which factors most affect the success of the landing outcomes. The correlation.
- Bar graph which shows the relationship between numerical and categoric variables:
  - Success Rate vs Orbit
- Line Graph which shows trends and makes predictions for unseen data:
  - Success rate vs Year





[Link to workbook document]

EDA with SQL

- SQL queries were 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 the ground pad was achieved.
  - List the names of the boosters which have success in drone ships and have payload mass greater than 4000 but less than 6000kg
  - List the total number of successful and failed mission outcomes
  - List the names of the booster versions which have carried the maximum payload mass.
  - List the records that will display the month names, failure outcomes in drone ship, booster versions, and launch site for the months in the 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 workbook document

Build an Interactive Map with Folium

- For visualization of launch sites, a marker with a circle around and a label with the name was added.
- Green and Red markers were added to indicate the launch outcome(success or failure)
- Using the Haversine's formula for distance calculation was used to answer the following question:
  - How close are the launch sites to railways, highways, and coastlines?
  - How close are the launch sites to nearby cities?

  Some lines with labels were added to indicate these distances.

Link to workbook document

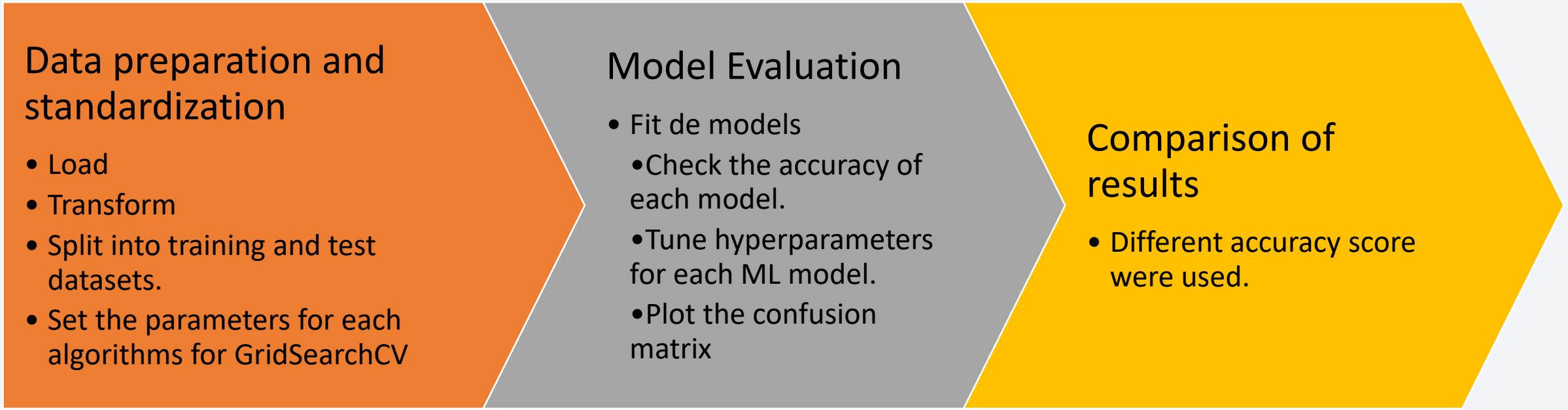
Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly Dash was built to plot graphs with different data. The components were:
  - Dropdown: allows users to choose all launch sites or a specific one (dash_core_components.Dropdown)
  - Pie chart: shows the total successful launches count for all sites and the success and failed counts for a specific site if it was chosen with the dropdown component. (plotly.express.pie)
  - RangeSlider: Allows users to select a payload mass in a fixed range (dash_core_components.RangeSlider).
  - Scatter chart: Shows the relationship between Success and Payload Mass (plotly.express.scatter)

Link to code

Predictive Analysis (Classification)

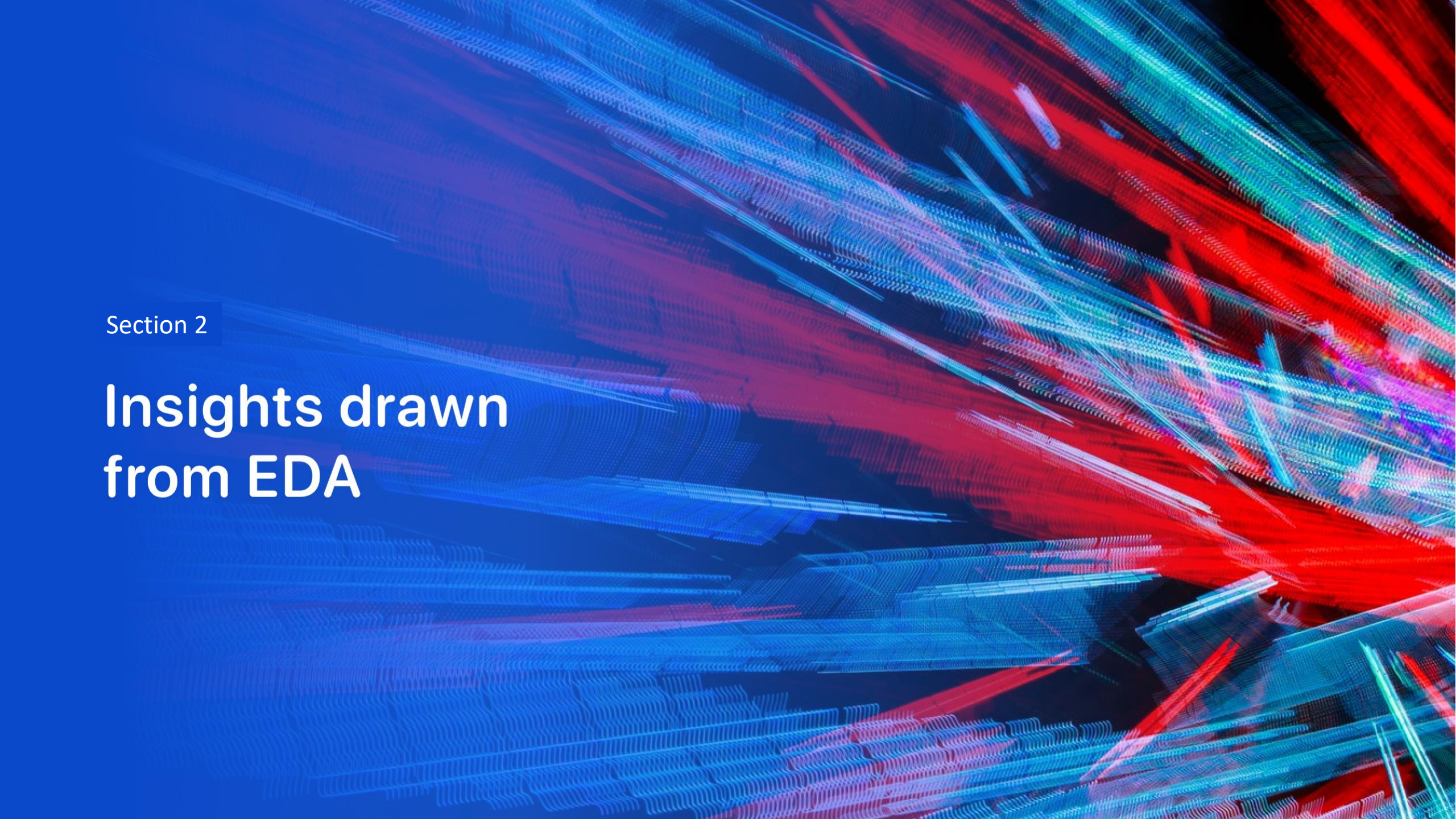
- Classification models used:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K-nearest Neighbors



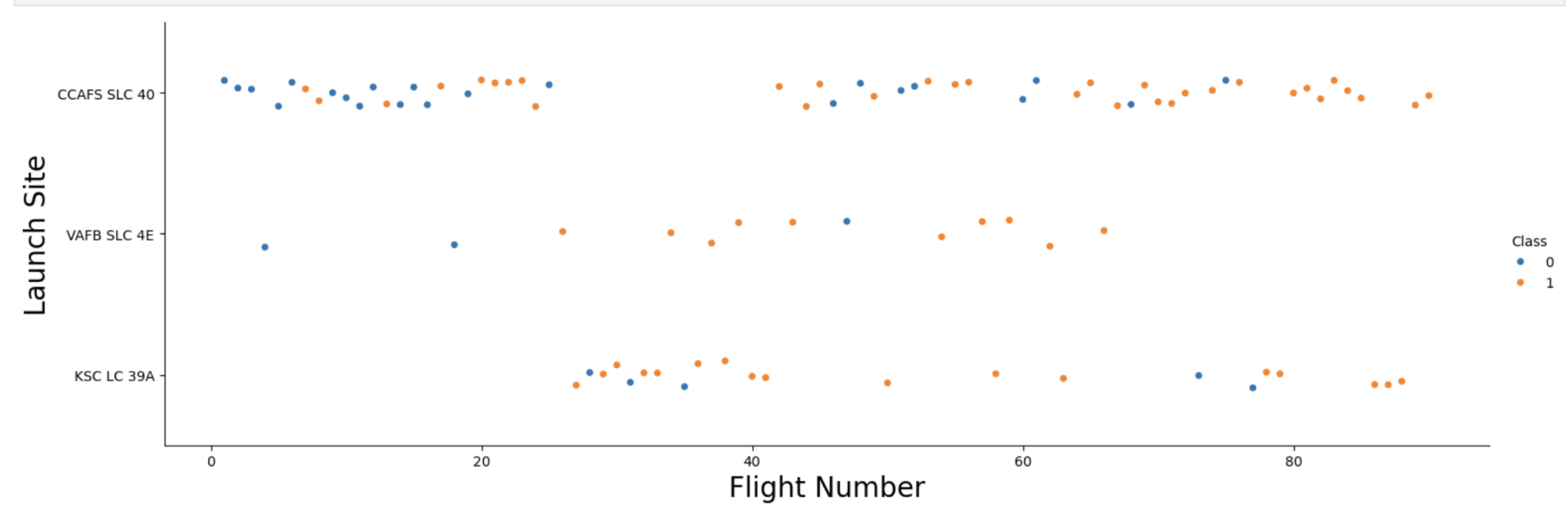
Link to workbook document

Results

- The results will be categorized into 3 main results:
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results


Section 2
Insights drawn from EDA

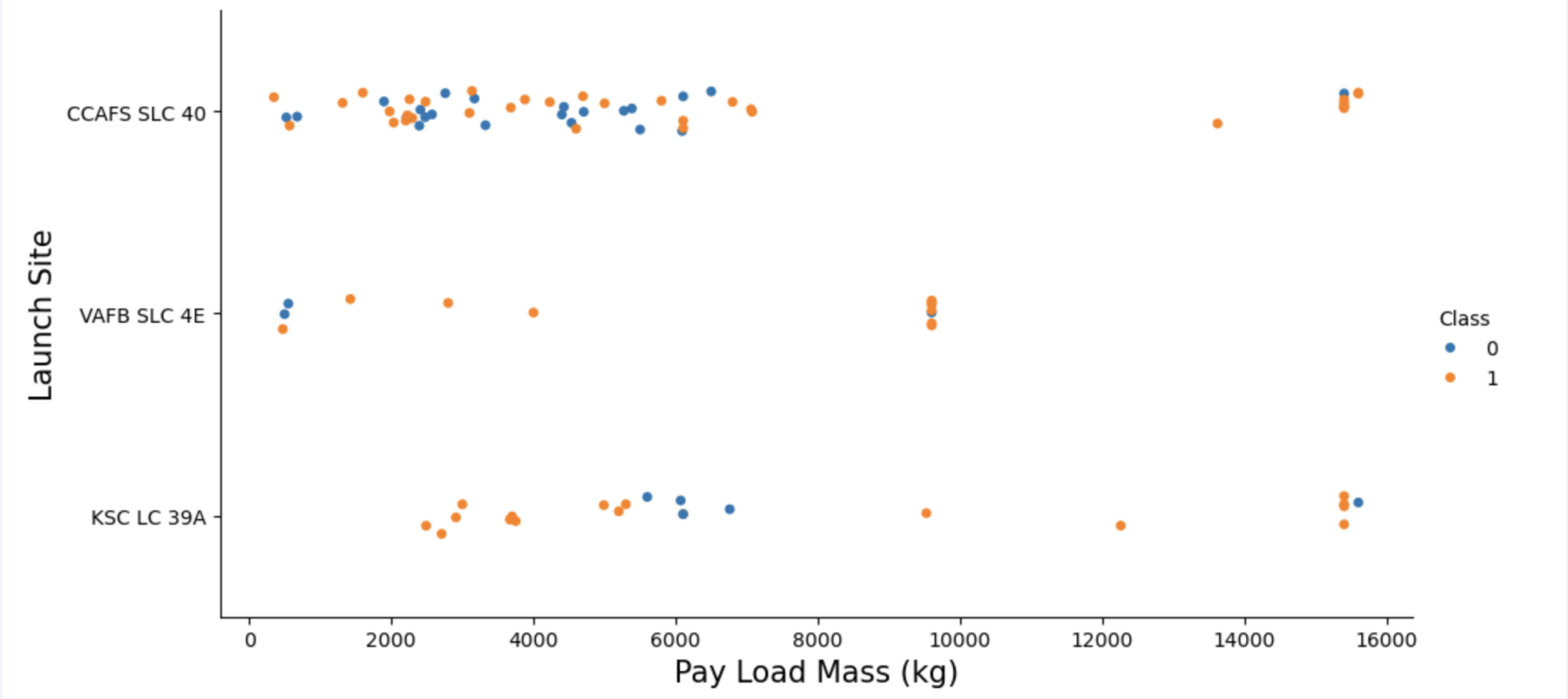
Flight Number vs. Launch Site



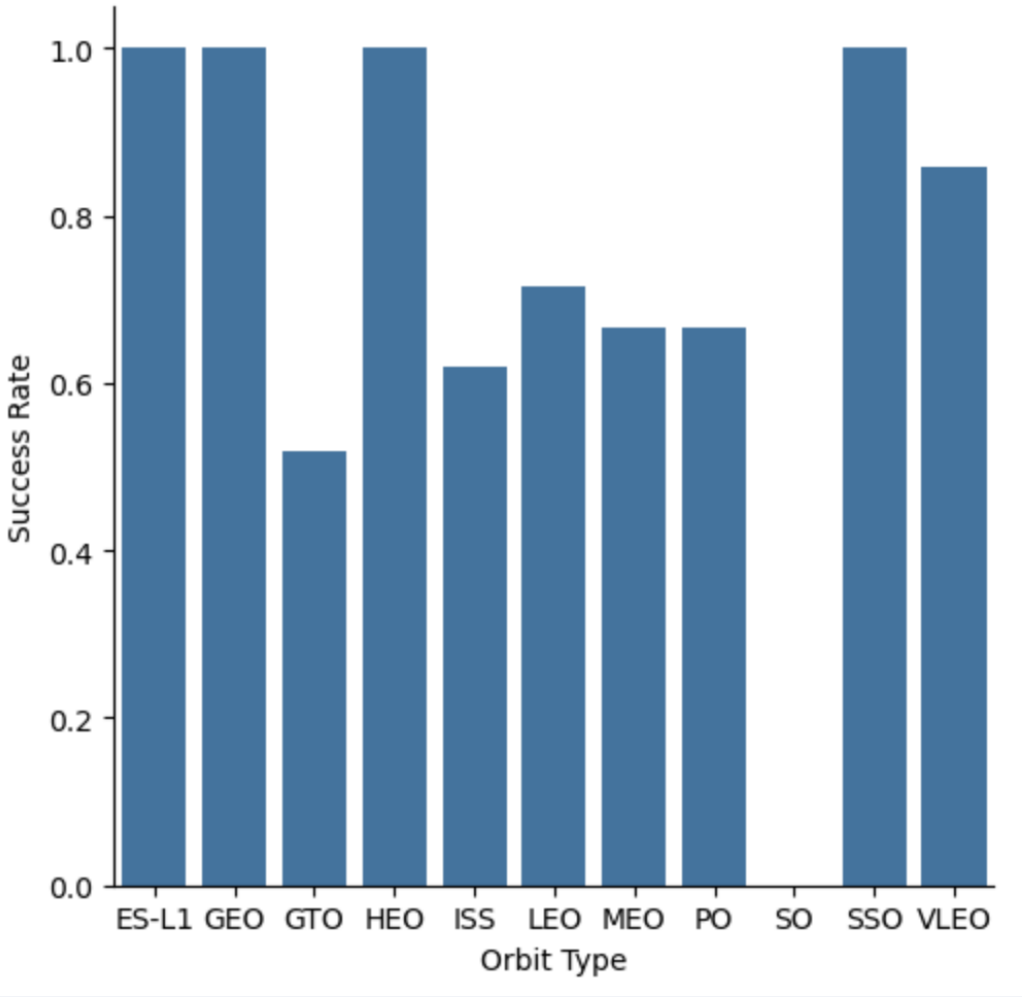
- The earliest flights all failed while the latest flights all succeeded in each Launch Site.
- The general success rate improved with each launch.
- CCAFS SLC 40 launch site has around the half total number of launches and the most recent ones were successful
- VAFB SLC 4E and KSC LC 39A have higher success rates.

Payload vs. Launch Site

- VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10.000kg)
- Payloads over 8,000 kg have an excellent success rate.



Success Rate vs. Orbit Type



	Orbit	Qty. of Lauch
0	ES-L1	1
1	GEO	1
2	GTO	27
3	HEO	1
4	ISS	21
5	LEO	7
6	MEO	3
7	PO	9
8	SO	1
9	SSO	5
10	VLEO	14

Orbits with a 100% success rate have only one launch:

ES-L1, GEO, HEO

While there is one 100% success rate with more launches and is better to do statistics:

SSO

Orbits with 0% success rate and just 1 launch is:

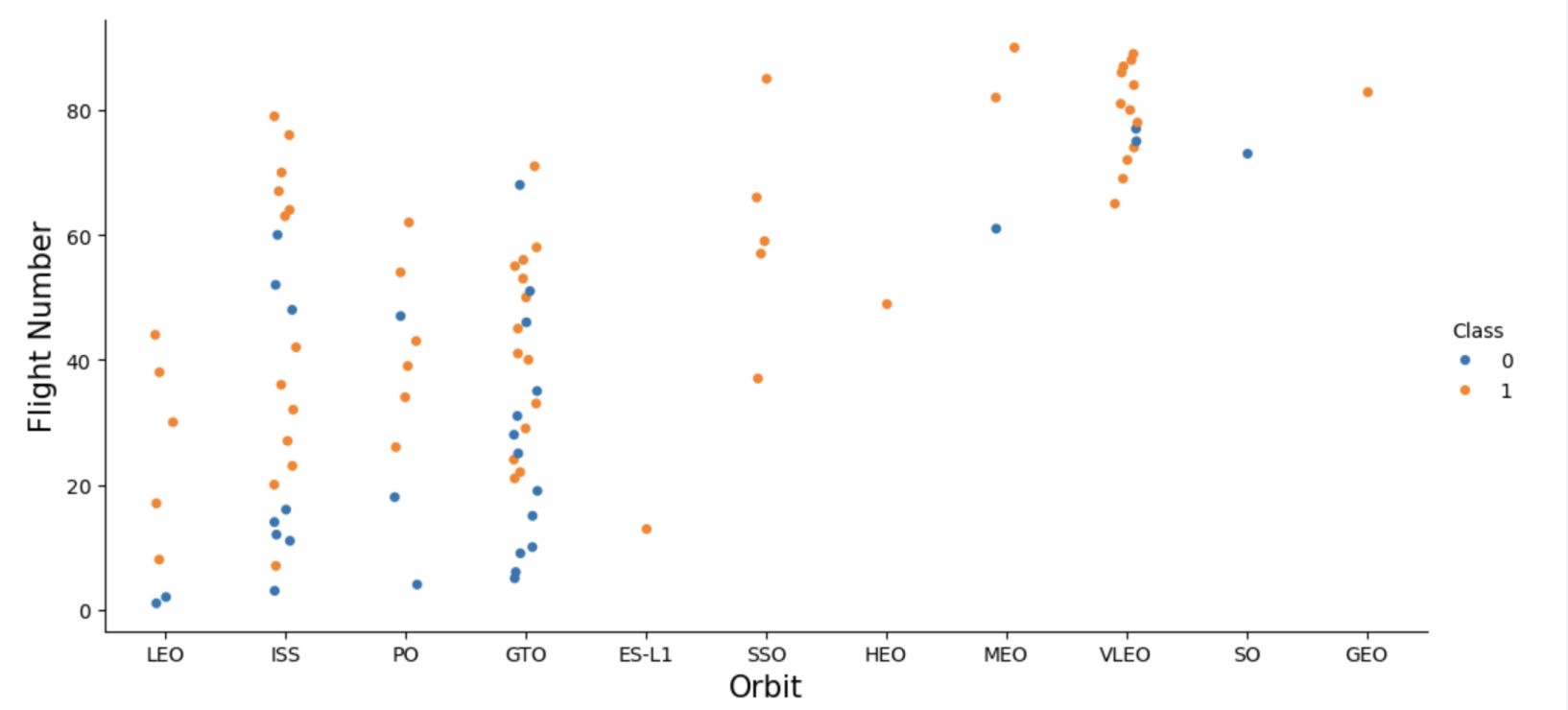
SO

Orbits with a success rate between 50% and 85%:

GTO, ISS, LEO, MEO, PO

Flight Number vs. Orbit Type

- LEO orbit success appears related to the number of flights;
- There seems to be no relationship between flight numbers when in GTO orbit.

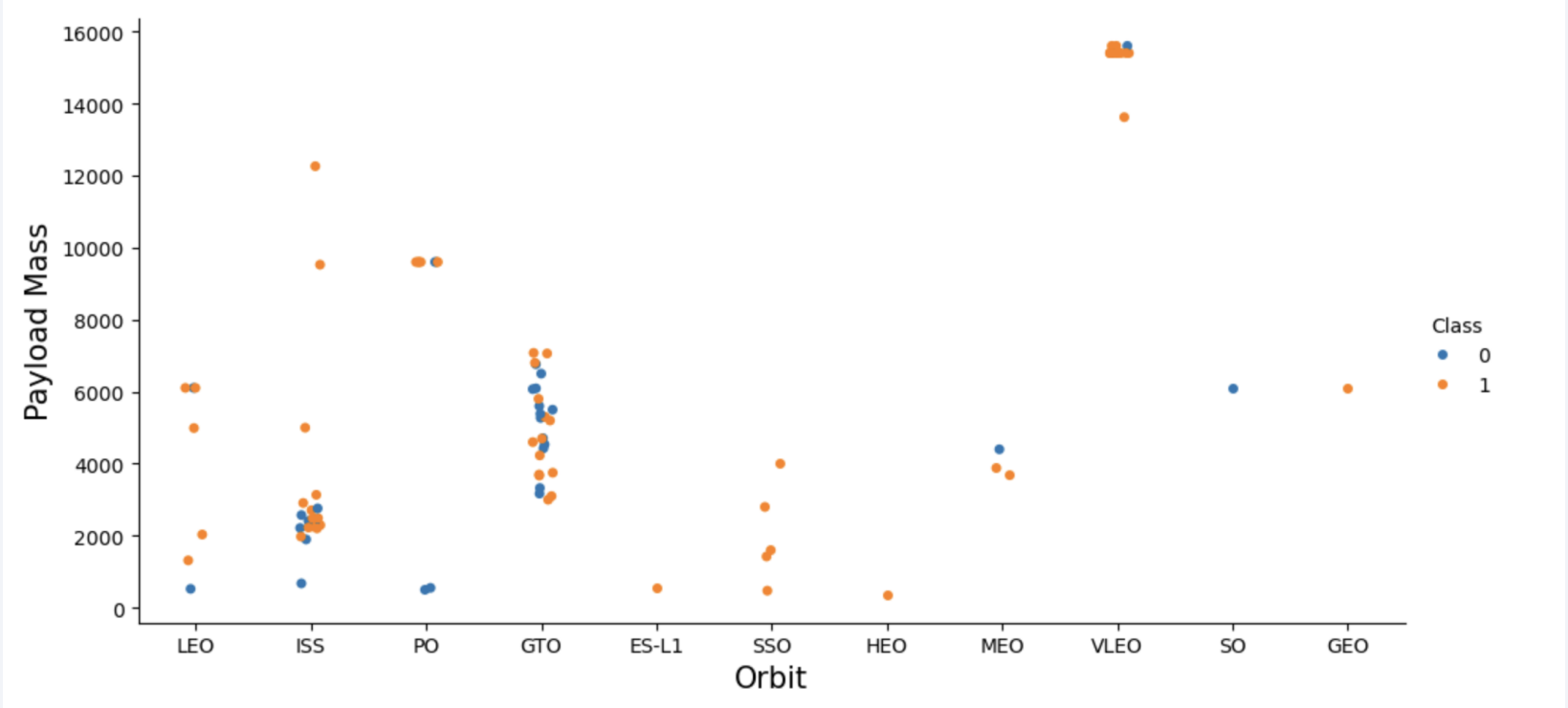


Payload vs. Orbit Type

- Heavy payloads for a successful landing or positive landing rate are more likely for LEO, ISS, and Polar. And in the opposite for MEO and VLEO orbits.

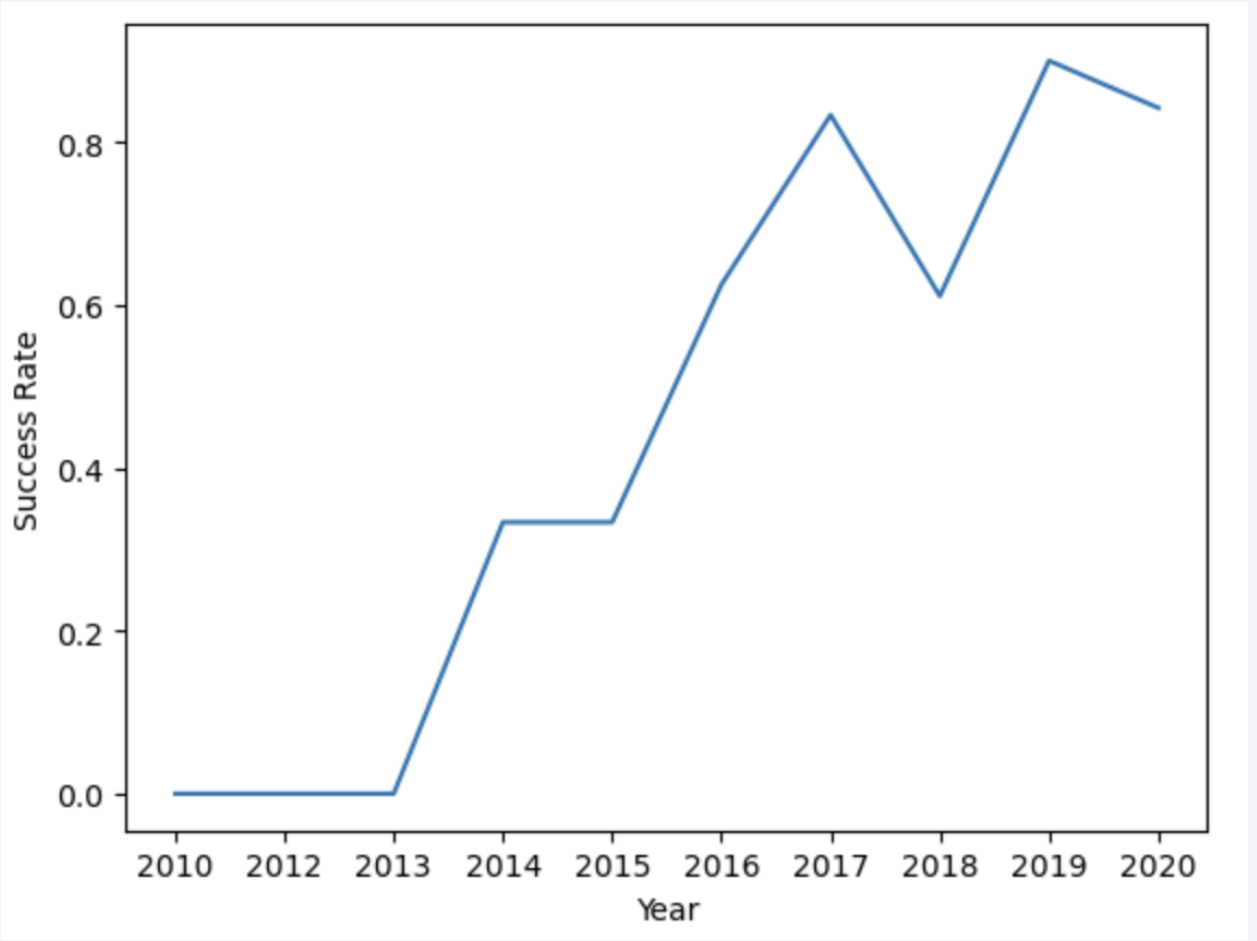
- For GTO cannot be distinguished this well as both positive landing rate and negative landing (unsuccessful mission), both there here.

- Orbits that only has 1 occurrence should be excluded until have more date.



Launch Success Yearly Trend

- From 2013 until 2017 the success rate keep increasing.



All Launch Site Names

- The keyword DISTINCT was used to show unique launch sites.

[144]:

```
QUERY = """
SELECT DISTINCT Launch_Site
FROM space_x_tbl
"""
pd.read_sql_query(QUERY,conn)
```

[144]:

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- A query to show 5 launch records that start with ‘CCA’ was executed.

```
QUERY = """
SELECT *
FROM space_x_tbl
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5
"""
pd.read_sql_query(QUERY,conn)
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by boosters from NASA was calculated. It is a total of 45.596Kg.

```
QUERY = """
SELECT SUM(PAYLOAD_MASS__KG_) AS "Total_Payload_Mass by NASA (CRS)"
FROM space_x_tbl
WHERE Customer="NASA (CRS)"
"""
pd.read_sql_query(QUERY,conn)
```

[11]:

	Total_Payload_Mass by NASA (CRS)
0	45596

Average Payload Mass by F9 v1.1

- For the booster version F9 v1.1, the average payload mass carried was calculated as 2.928,4

[170]:

```
QUERY = """
SELECT AVG(PAYLOAD_MASS__KG_) AS Mean_Payload
FROM space_x_tbl
WHERE Booster_Version like "%F9 v1.1%"
"""
pd.read_sql_query(QUERY,conn)
```

[170]:

	Mean_Payload
0	2534.666667

First Successful Ground Landing Date

- The MIN() function was used to find the result from the “Landing_Outcome” column.
- It was observed that the first landing outcome on a ground pad was on the: 22nd of December of 2015.

```
QUERY = """
SELECT MIN(Date) AS First_Succesful_Landing_Outcame_Ground_Pad
FROM space_x_tbl
WHERE Landing_Outcome = "Success (ground pad)"
"""
pd.read_sql_query(QUERY,conn)
```

[29]:

	First_Succesful_Landing_Outcame_Ground_Pad
0	2015-12-22

Successful Drone-Ship Landing with Payload between 4000 and 6000

- The busters which have a successful landing in drone-ship and with a payload between 4000kg and 6000 kg were queried.

```
QUERY = """
SELECT Booster_Version
FROM space_x_tbl
WHERE LANDING_OUTCOME = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ <6000
"""
pd.read_sql_query(QUERY,conn)
```

[30]:

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The total success and failure mission outcomes number was queried.
- There were 100 success and 1 failure.

```
QUERY = """
SELECT Mission_Outcome,COUNT(*) AS Number_of_Successful_And_Failure_Mission
FROM space_x_tbl
GROUP BY Mission_Outcome
"""
pd.read_sql_query(QUERY,conn)
```

[175]:

	Mission_Outcome	Number_of_Successful_And_Failure_Mission
0	Failure (in flight)	1
1	Success	98
2	Success	1
3	Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The boosters which have carried the maximum payload mass were queried. A subquery was needed to find the maximum payload first.

```
QUERY = """
SELECT Booster_Version
FROM space_x_tbl
WHERE PAYLOAD_MASS__KG_ = (
SELECT MAX(PAYLOAD_MASS__KG_)
FROM space_x_tbl
)
"""
pd.read_sql_query(QUERY,conn)
```

[185]:

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

2015 Launch Records

- 2015 was a year when the success rate started to increase.
- There were a couple of failed landing outcomes for a drone ship in the 2015 year which were listed.

```
QUERY = """
SELECT SUBSTR(Date,6,2) AS Month, Date, Booster_Version , Launch_Site
FROM space_x_tbl
WHERE Landing_Outcome = 'Failure (drone ship)'
AND SUBSTR(Date,0,5) = '2015'
"""
pd.read_sql_query(QUERY,conn)
```

[199]:

	Month	Date	Booster_Version	Launch_Site
0	01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40
1	04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40

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

- A rank of the count of kind of landing outcomes between 2010-06-04 and 2017-03-20 dates were queried in descending order.
- There was a great number of no attempts of landing during these years.

```
QUERY = """
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Count_Of_Landing_Output
FROM space_x_tbl
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Count_Of_Landing_Output DESC
"""
pd.read_sql_query(QUERY,conn)
```

[202]:

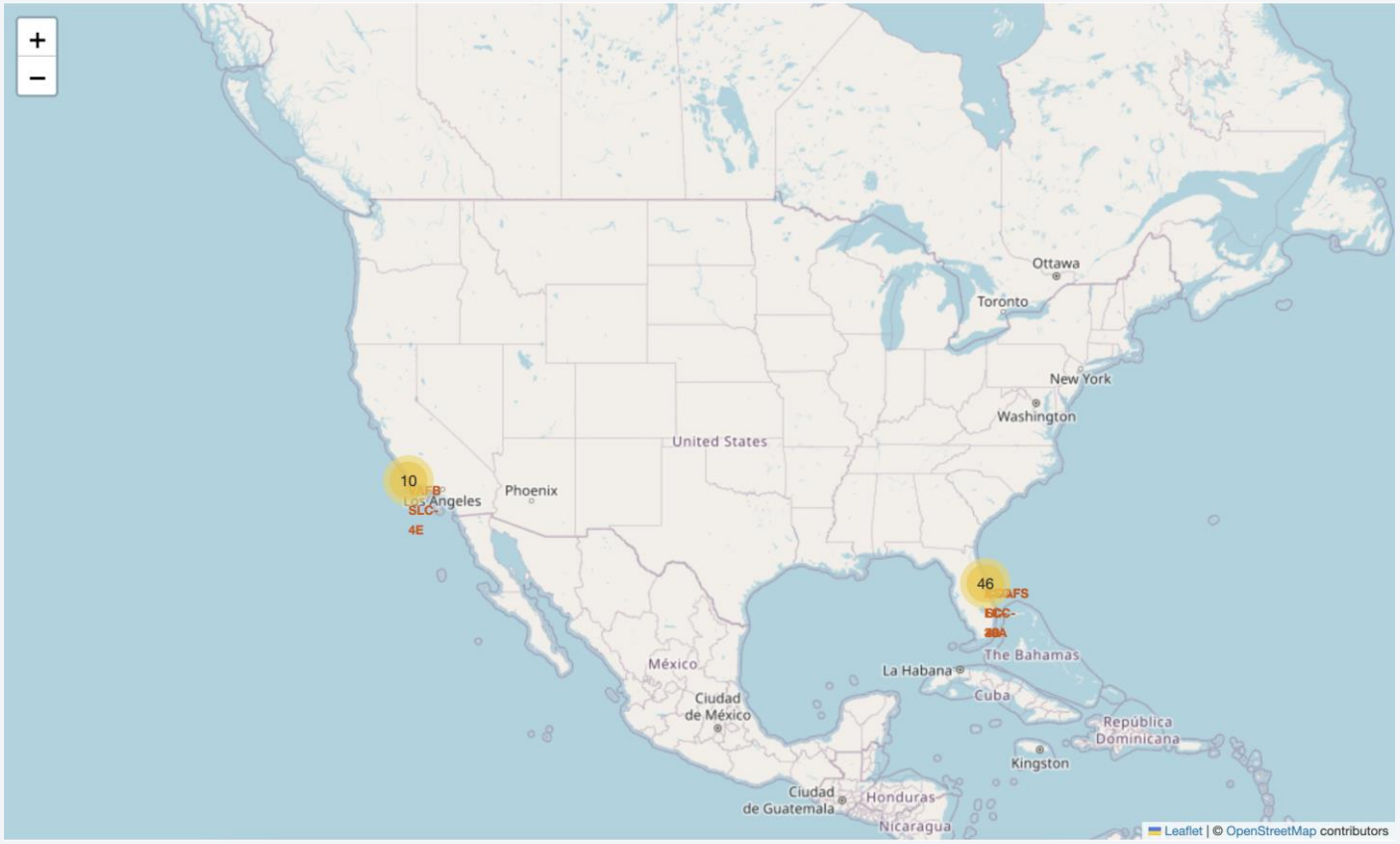
	Landing_Outcome	Count_Of_Landing_Output
0	No attempt	10
1	Success (drone ship)	5
2	Failure (drone ship)	5
3	Success (ground pad)	3
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Failure (parachute)	2
7	Precluded (drone ship)	1

Section 3

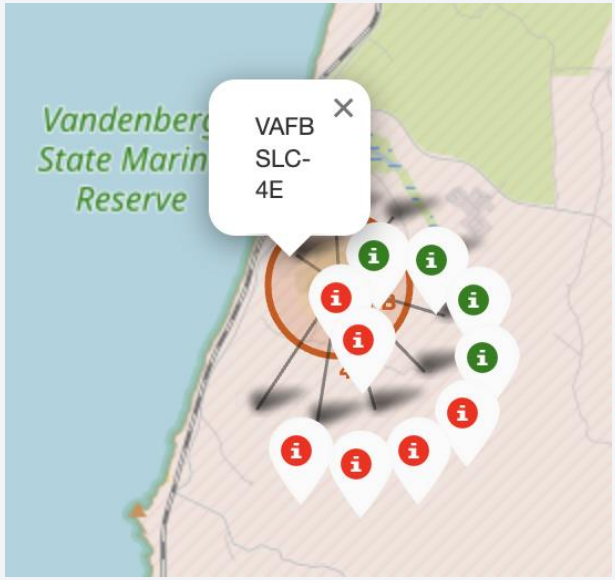
Launch Sites Proximities Analysis

Folium Map - Launch Sites

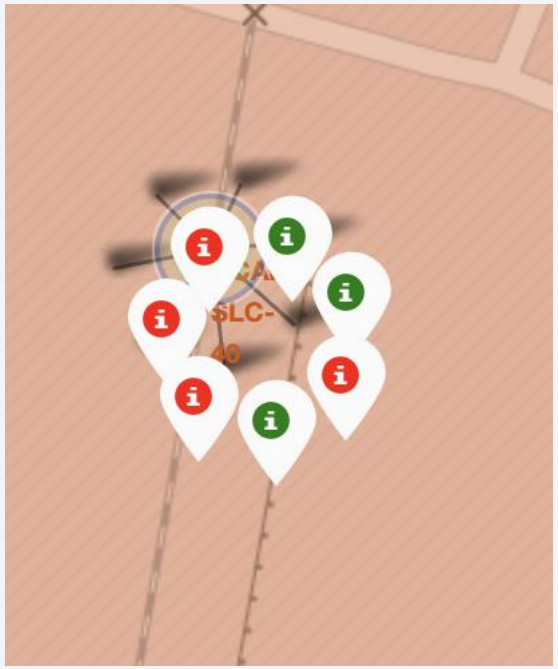
- As can be depicted the launch sites are most close to the Ecuador line inside the USA territory. At the same time close to the cost in both sides of the county.

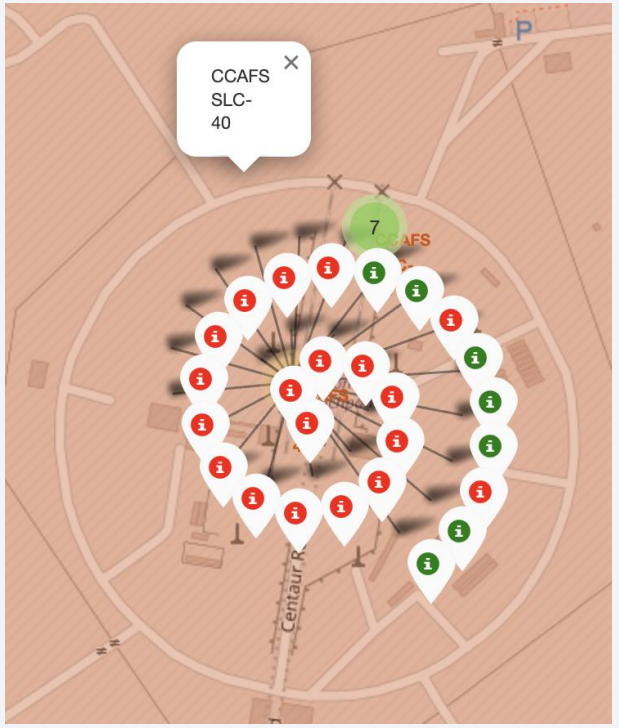


Folium Map – Launch sites with color marker



California Launch Site



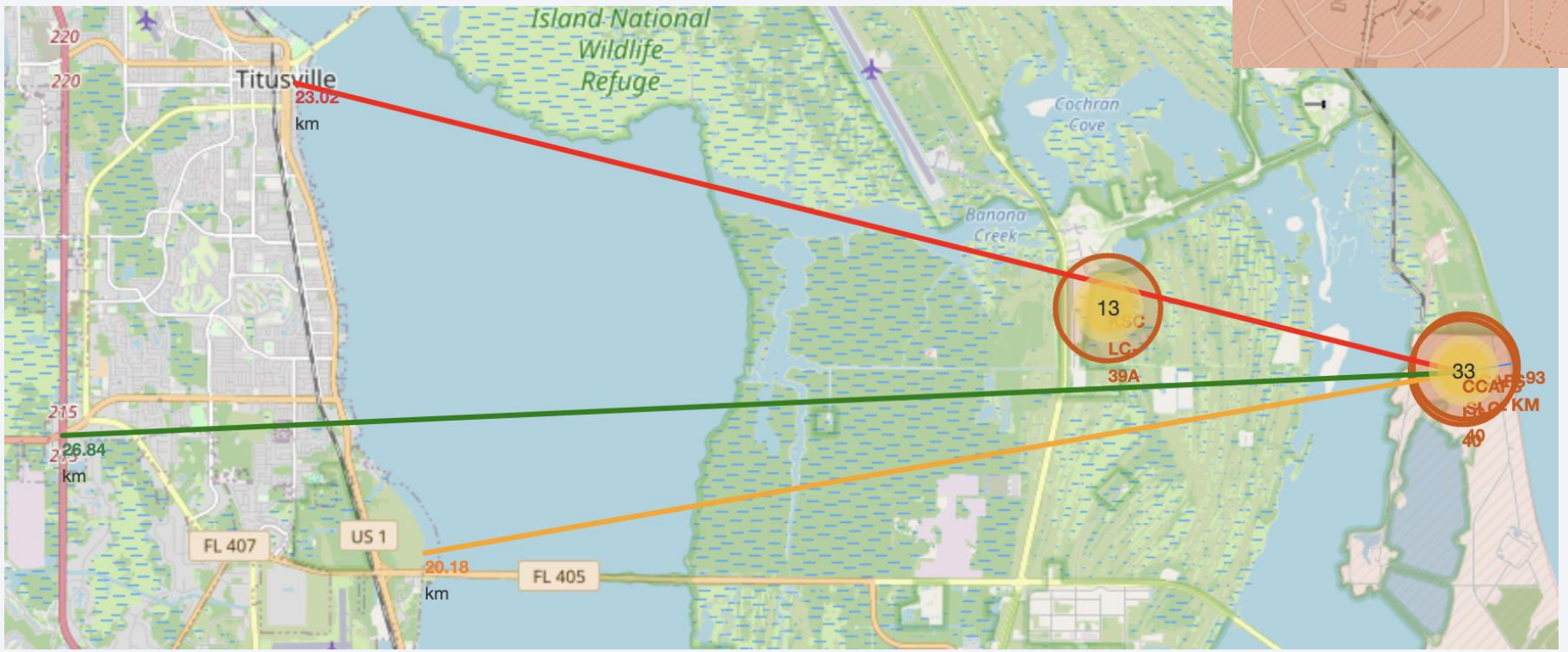


Florida Launch Sites

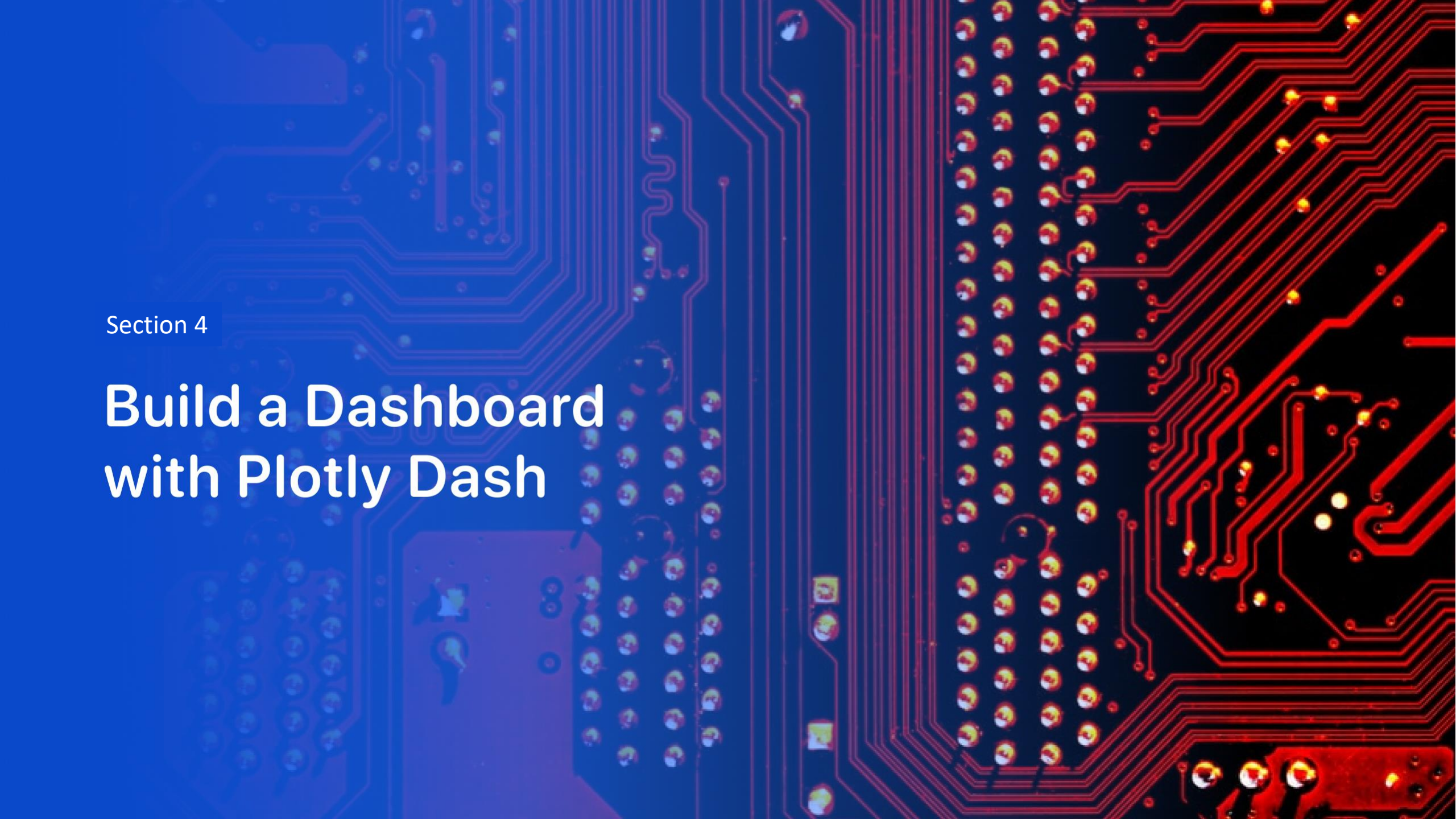
Green Marker shows successful launches and Red Marker shows failures.

<Folium Map Screenshot 3>

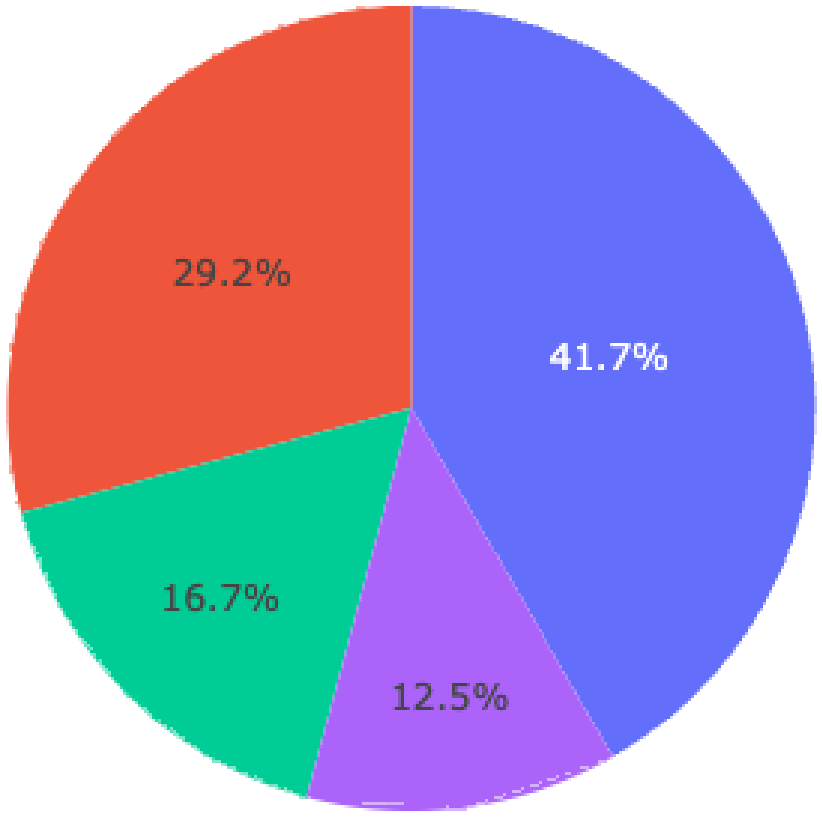
- It was taken as a reference to a launch site with the highest number of launches.
- The close railways, highways, and cities are more than 20km far away.

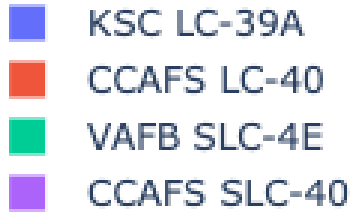


Meanwhile, the cost line is 0,93 km far away.


Section 4
Build a Dashboard with Plotly Dash

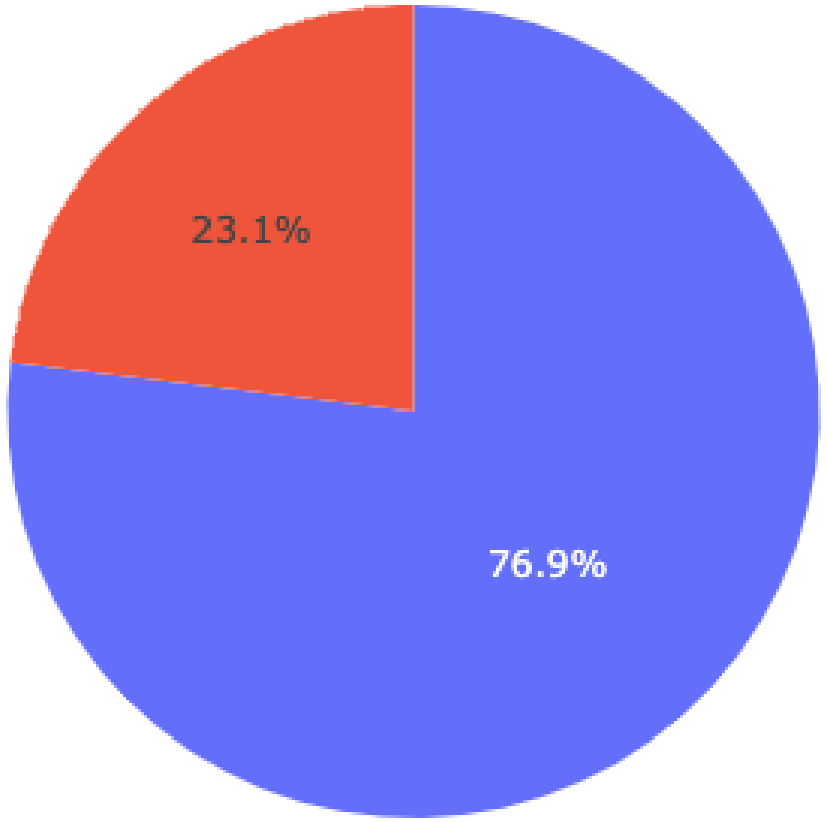
Total success launches by site





- KSC LC-39A had the most successful launches from all the sites.

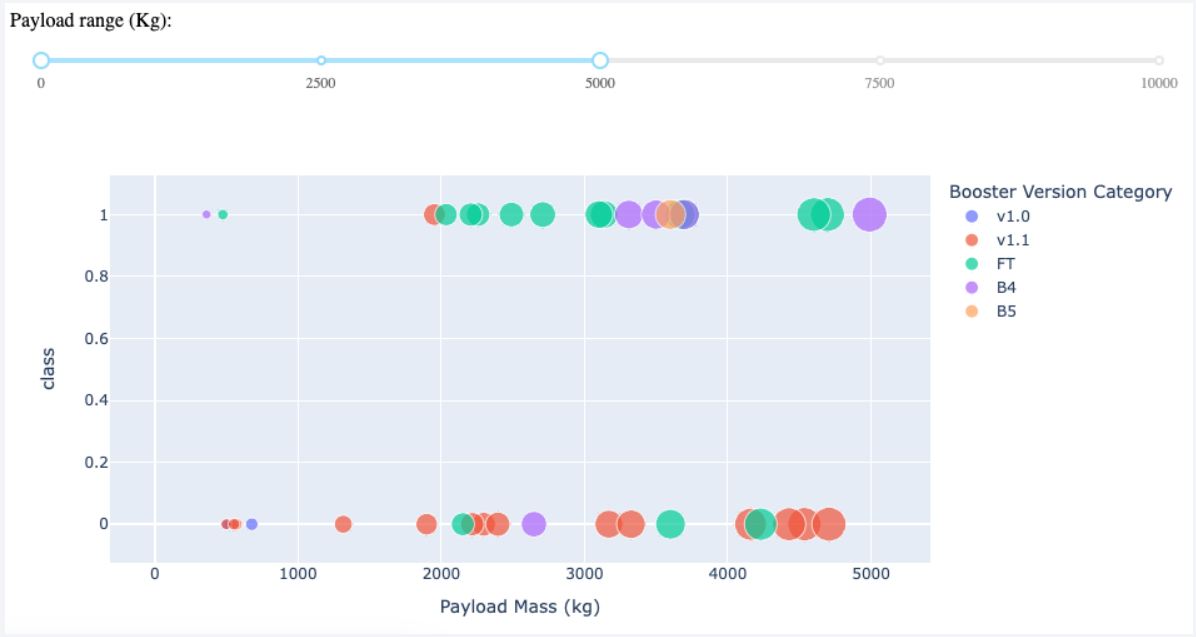
Launch-success ratio for KSC LC-39A



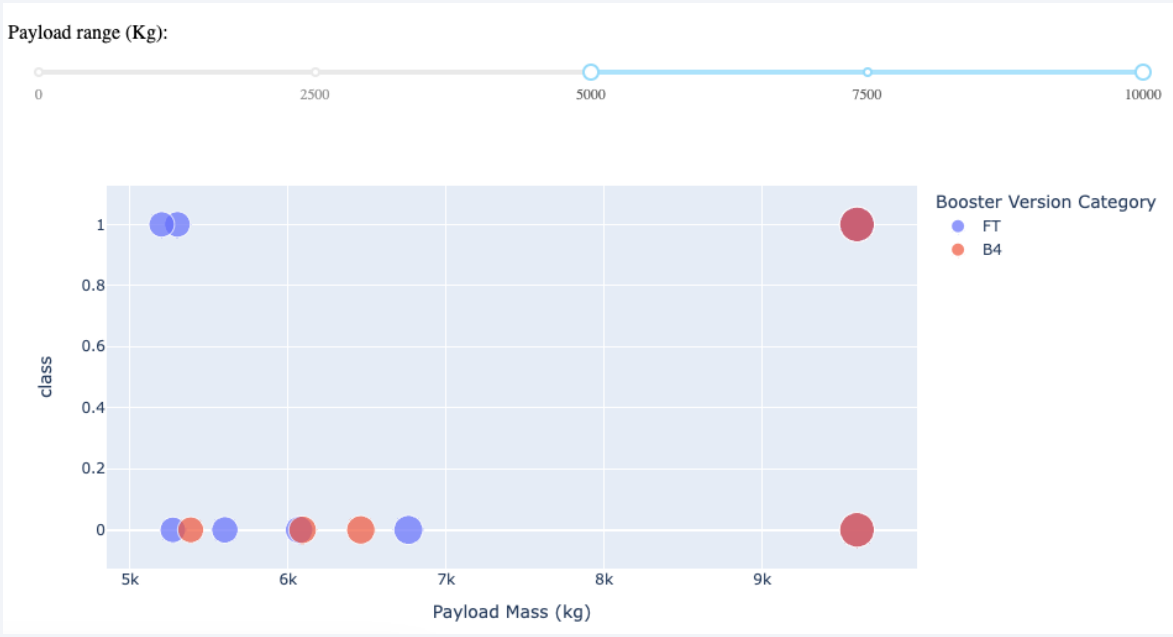
- KSC LC-39 who has the highest percentage of success, achieved a 76,9% success rate while getting a 23.1% failure rate.

Launch Outcome vs Payload

- Can be depicted that low-weight payloads have a success rate higher than heavy-weight payloads.



Low-weight Payload (0kg – 5.000kg)



Heavy-weight Payload (5.000kg-10.000kg)

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- For different models, the accuracy was scored :
  - LogReg: Logistic regression
  - SVM: Support Vector Machine
  - Tree: Decision tree
  - KNN: K-nearest neighbors

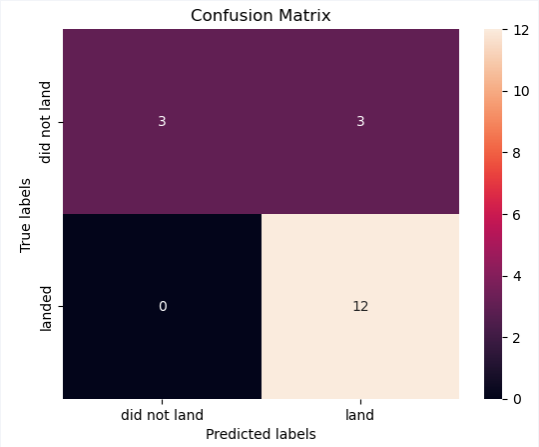
	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.785714	0.800000
F1_Score	0.888889	0.888889	0.880000	0.888889
GridSearchCV.score()	0.833333	0.833333	0.833333	0.833333

All the performed methods were similar.
The preferred to keep working in the future is the KNN:

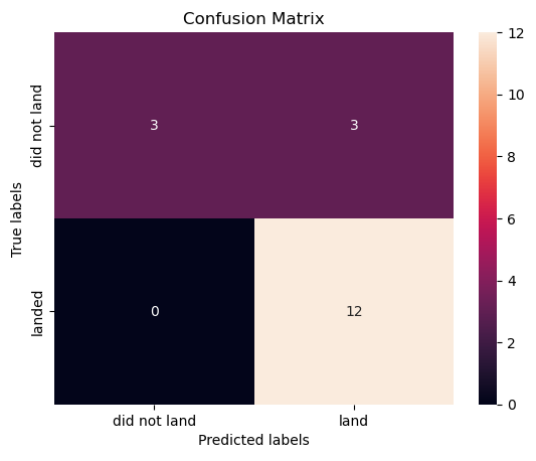
```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
```

Confusion Matrix

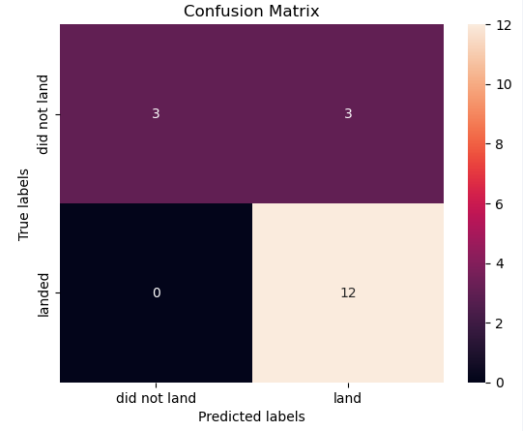
Logistic regression



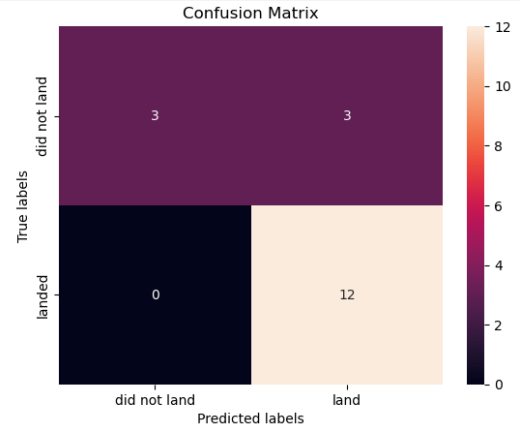
Support Vector Machine



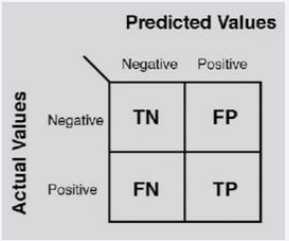
Decision Tree



KNN: K-nearest neighbors



- As the accuracy tests were similar, the confusion matrices were also similar except for the decision tree which had some problems.
- The biggest problem of these models is these false positives, i.e. unsuccessful landing marked as successful landing by the classifier.



Conclusions

- It was possible to perform extraction, transformation, and exploratory data analysis successfully to have a better understanding of the problem.
- Four Machine Learning models were launched to find a way to predict the launch outcome and a high accuracy was achieved but the false positives are the biggest problem of the models.
- Low-weighted payloads (below 5000kg) performed better than heavy-weighted payloads.
- The increase in success rate from 2013 ongoing is quite promising.
- The site KSC LC-39A was identified as the most successful launch site with a 76,9%
- The GEO, HEO, and SSO orbits were correctly identified as having the highest success rates.
- The SSO orbit has a 100% success rate with 5 launches, more launches are needed to make statistics, but it generates good expectations.

Thank you!