IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Doris Macht
24th May 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection API
  - Data Collection with Web Scraping
  - Exploratory Data Analysis (EDA) incl. SQL and Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Data Analysis with Interactive Visualization
  - Predictive Analytics results

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - What are the main factors who determine if the rocket will land successfully?
  - Which interaction of various features determine the success rate of a successful landing.
  - Which preconditions must be given to achieve the best possible result?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding was applied to categorical features (Transform data for Machine Learning (ML)
- 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

- Basic Steps
    1. Getting Data from API and Web Page
    2. Load the data into a Pandas dataframe using .json_normalize().
    3. Filter the dataframe as required
    4. Export to flat file (.csv)

Data Collection – SpaceX API

1. Getting Response

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

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

2. Converting Response to .json file and turn it into a dataframe

```
# Decode the response content as json
df_spacex = data_request.json()

# Normalize the json data
data = pd.json_normalize(df_spacex)
```

3. Apply custom functions to clean data

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

4. Assign list to dictionary and create a dataframe

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

5. Filter dataframe and export to flat file

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Github

Data Collection - Scraping

1. Getting Response from HTML
2. Creating BeautifulSoup Object
3. Finding tables
4. Getting column names
5. Creation of dictionary and appending data to keys
6. Converting dictionary to dataframe
7. Dataframe to .csv

```
html_falcon9 = requests.get(static_url)
```

```
soup = BeautifulSoup(html_falcon9.text, 'html.parser')
```

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

```
element = soup.find_all('th')

for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

```
launch_dict['Booster landing'].append(booster_landing)
```

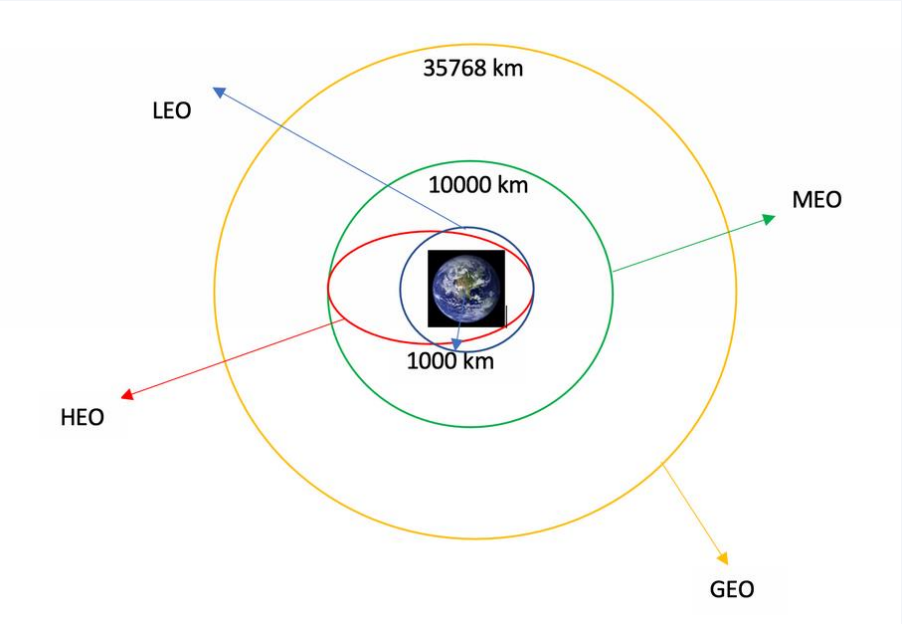
```
df=pd.DataFrame(launch dict)
```

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

Github

Data Wrangling

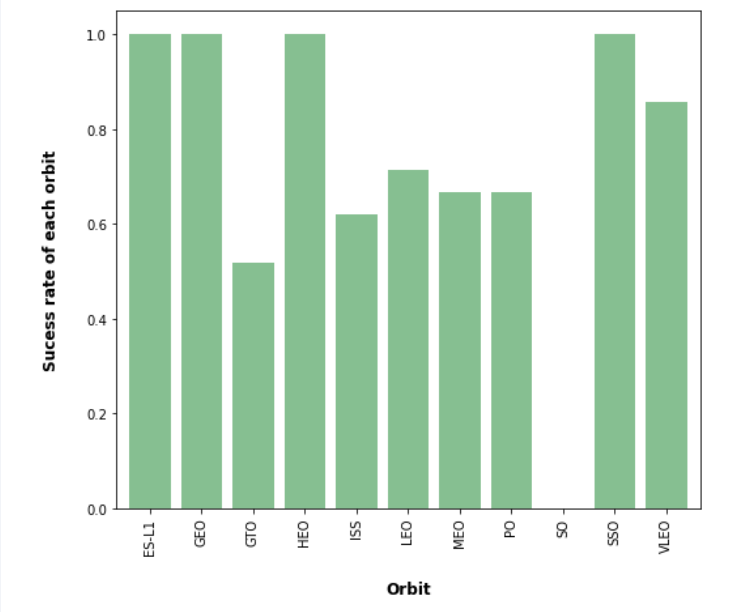
- Perform exploratory Data Analysis and determine Training labels
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.



Github

EDA with Data Visualization

- Perform exploratory Data Analysis using Pandas and Matplotlib
- Explore the data by visualizing the relationship between different features like number and Launch Site, payload and launch site etc. by using different chats.



Github

EDA with SQL

- We use IBM's Db2 for Cloud to store the data.
- We applied EDA with SQL to get insight from that data and wrote queries to find out more about it:
  - Display the names of the unique launch sites in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'.
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery.
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

Github

Build an Interactive Map with Folium

- Visualize data on an interactive leaflet map
  - Using latitude and longitude for each launch site
  - Using the color-labeled marker clusters to identify which launch sites have relatively high success rates
- Answering questions for example:
  - Are launch sites next to railways, highways or coastlines?
  - How far away are launch sites to cities?

Github

Build a Dashboard with Plotly Dash

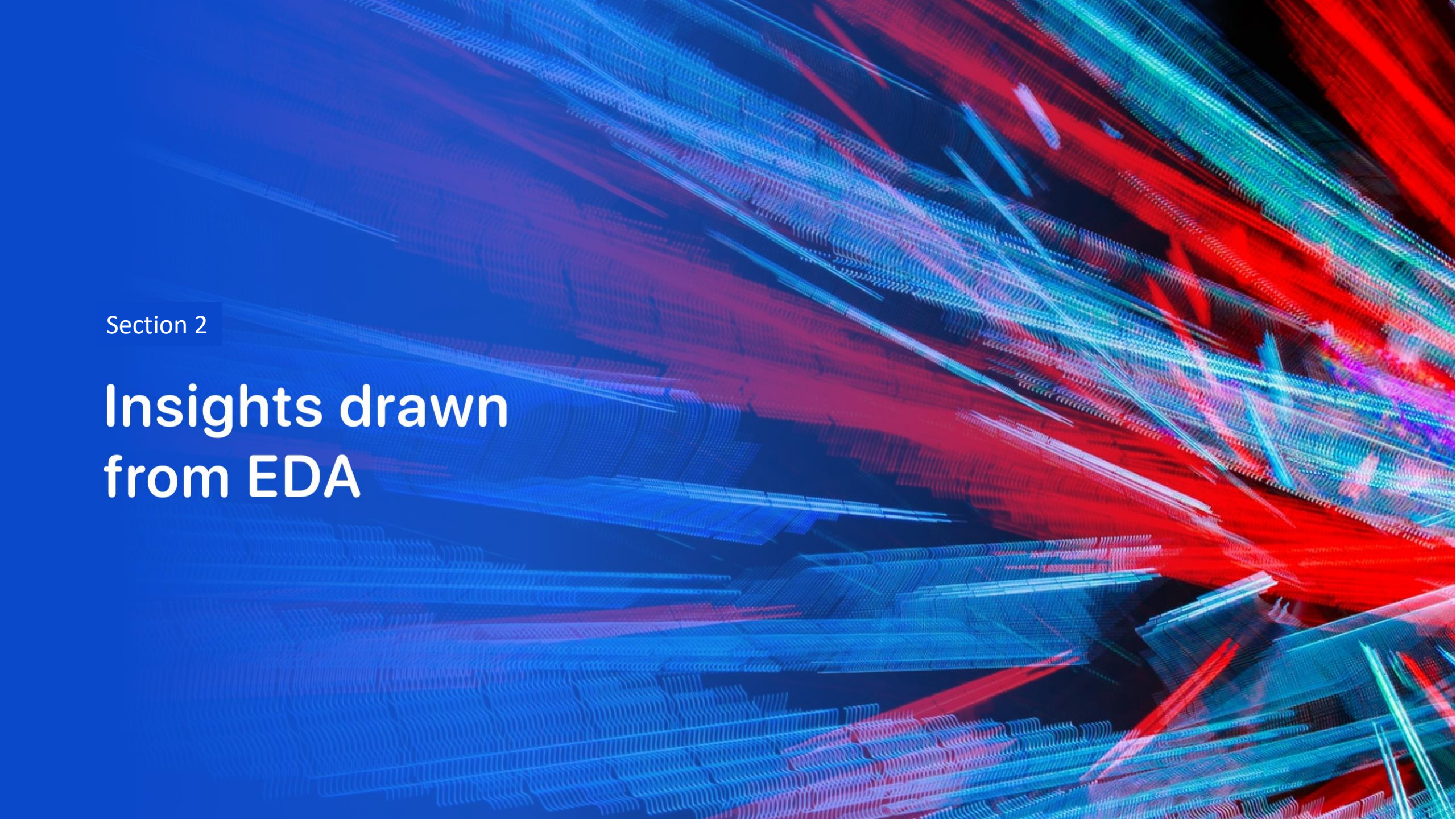
- Interactive Dashboard building with Plotly Dash.
- Plotting Pie charts to show total launches by a certain site.
- Plotting Scatter Graph to show correlation between Outcome and Payload Mass (Kg) for different booster versions.

Predictive Analysis (Classification)

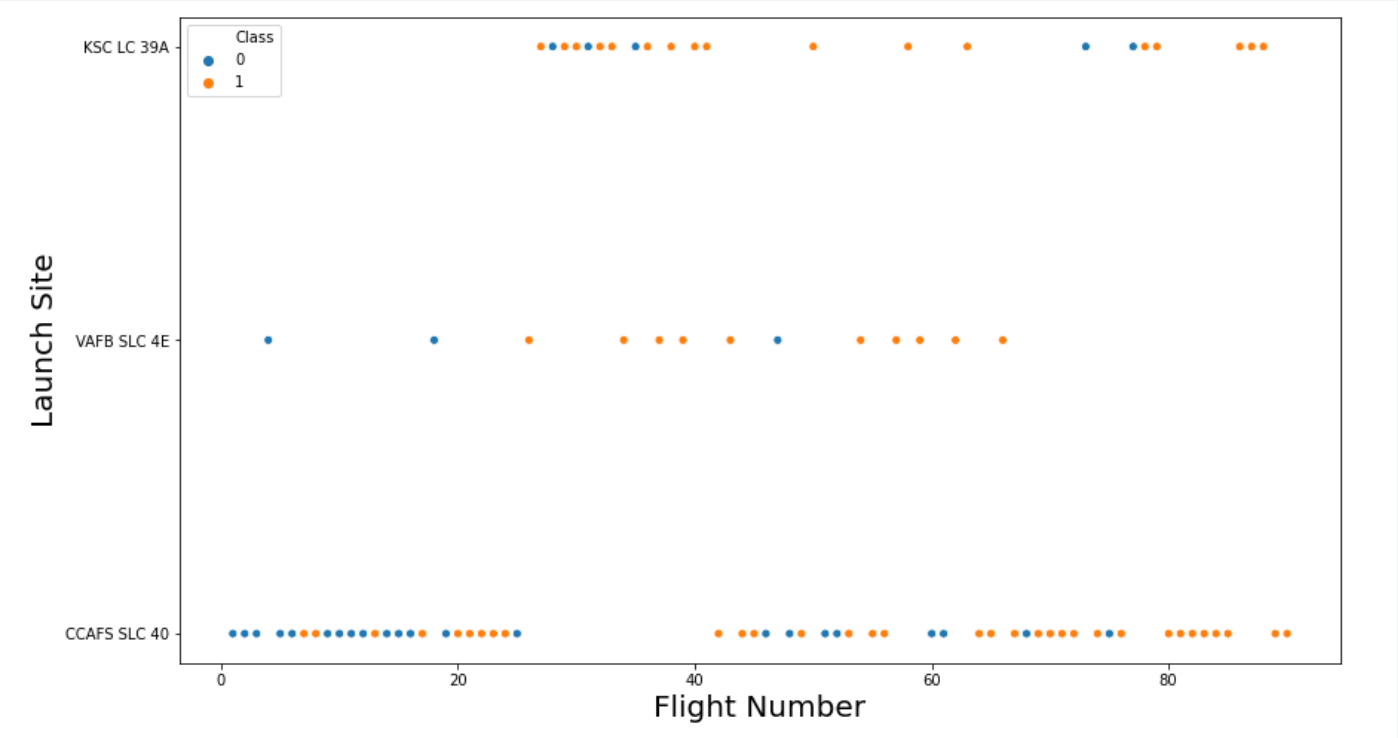
- Perform EDA and determine Training Labels to
  - Create columns for Classification
  - Standardization of data
  - Splitting data into training and test data
- Find Hyperparameters for SVM, Classification Trees and Logistic Regression to find the method what performs best under the use of test data

Results

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

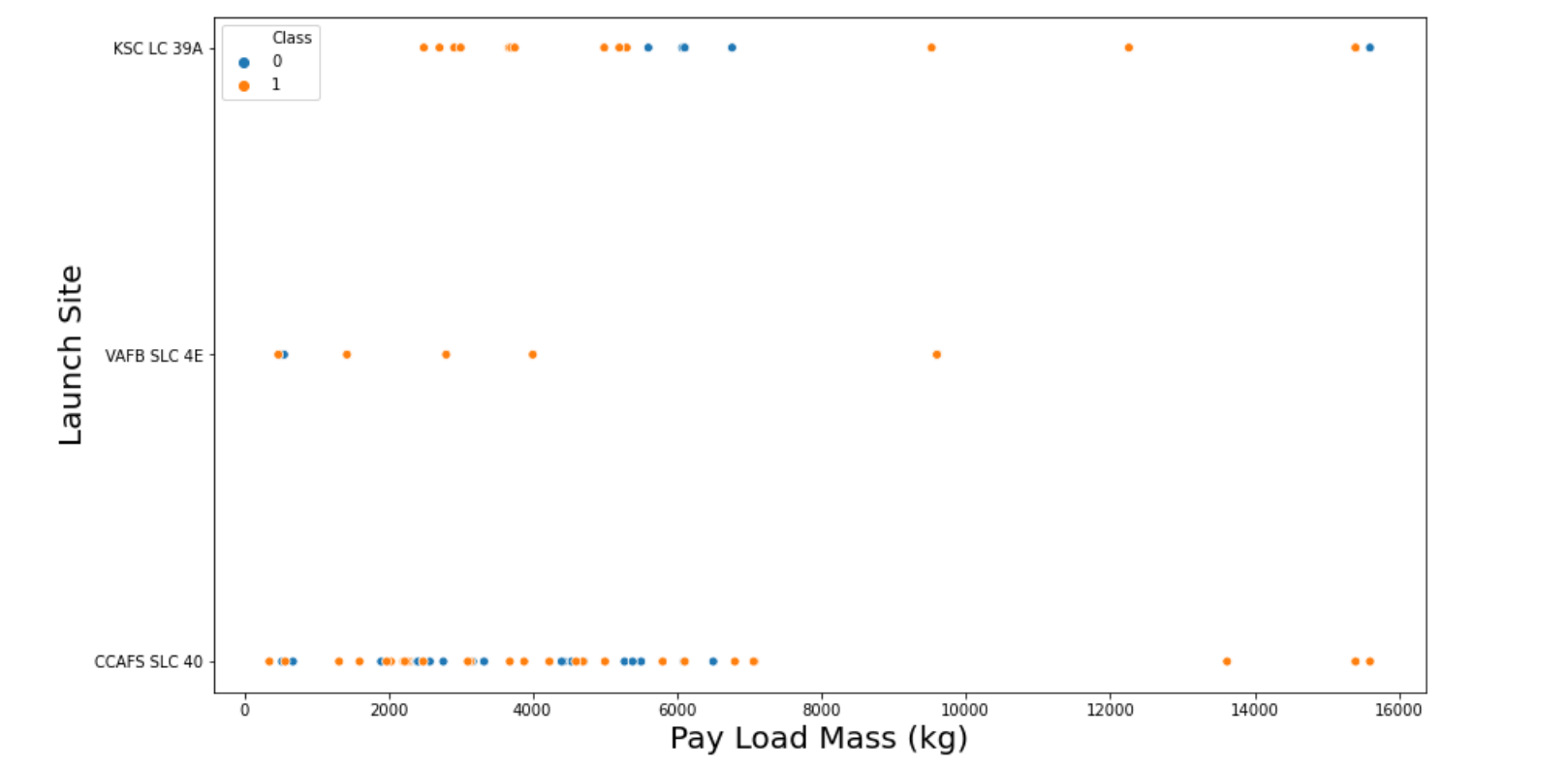

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



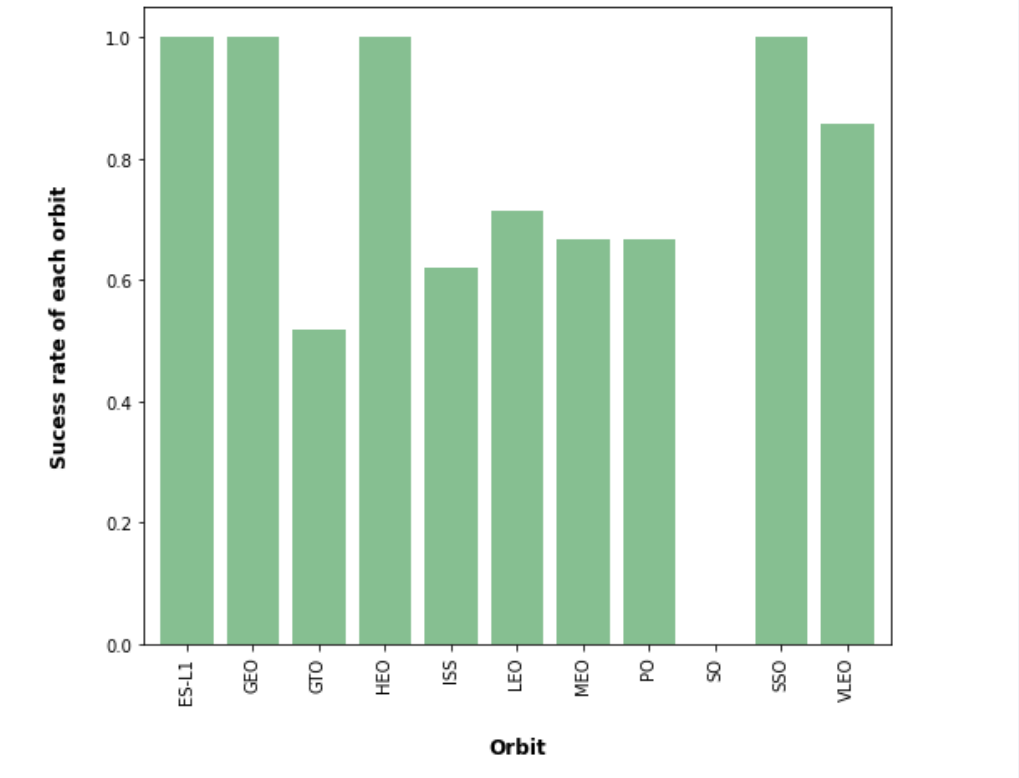
- With more flight numbers (after 40) higher the success rate for the Rocket is increasing.
- But there's no clear pattern to decide if the Flight Number is dependent on Launch Site for a success launch.

Payload vs. Launch Site



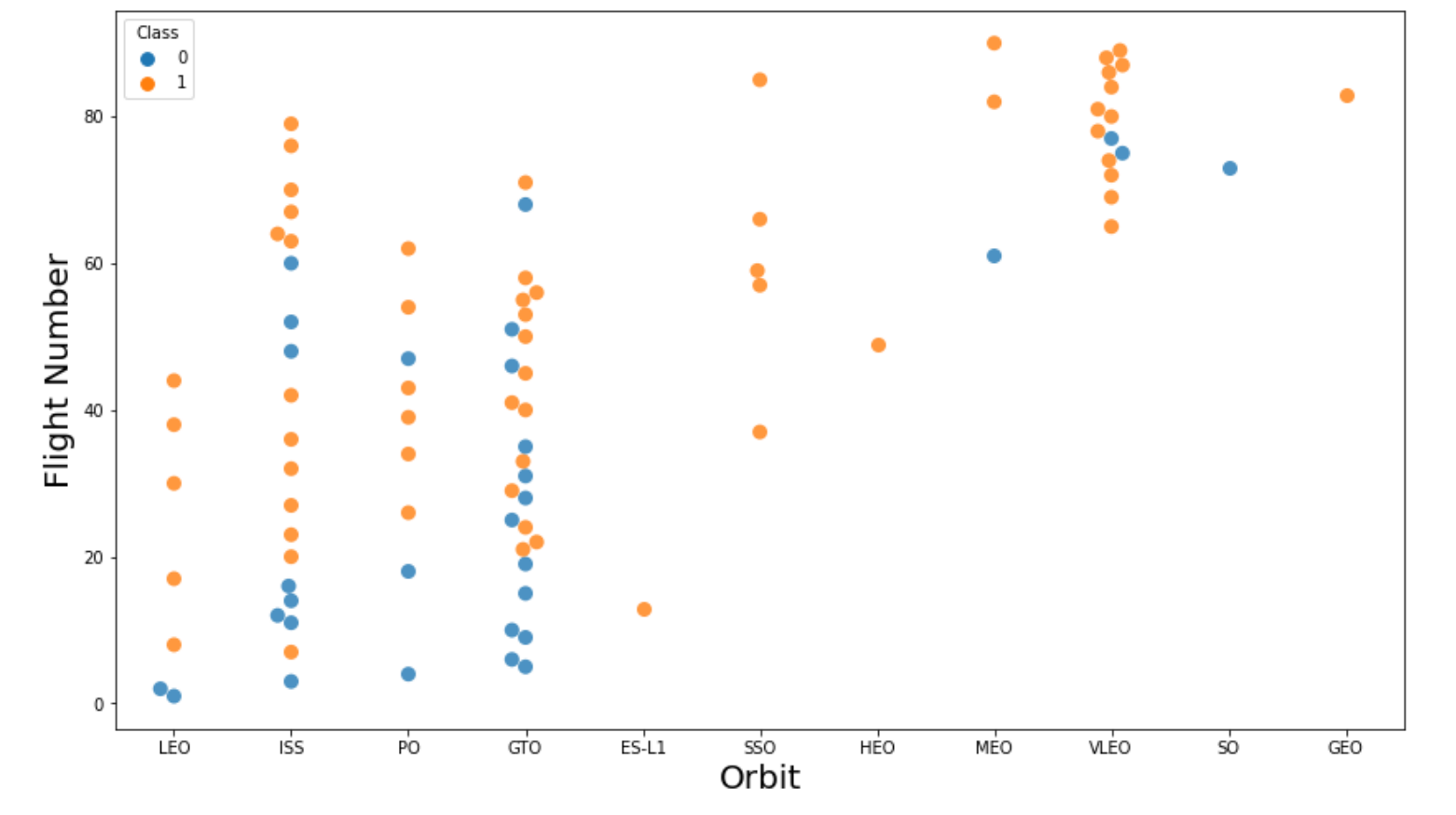
– The greater the payload mass (greater than 8000) higher the success rate for the Rocket.
– But there's no clear pattern to decide if the Launch Site is dependent on Pay Load Mass for a success launch.

Success Rate vs. Orbit Type



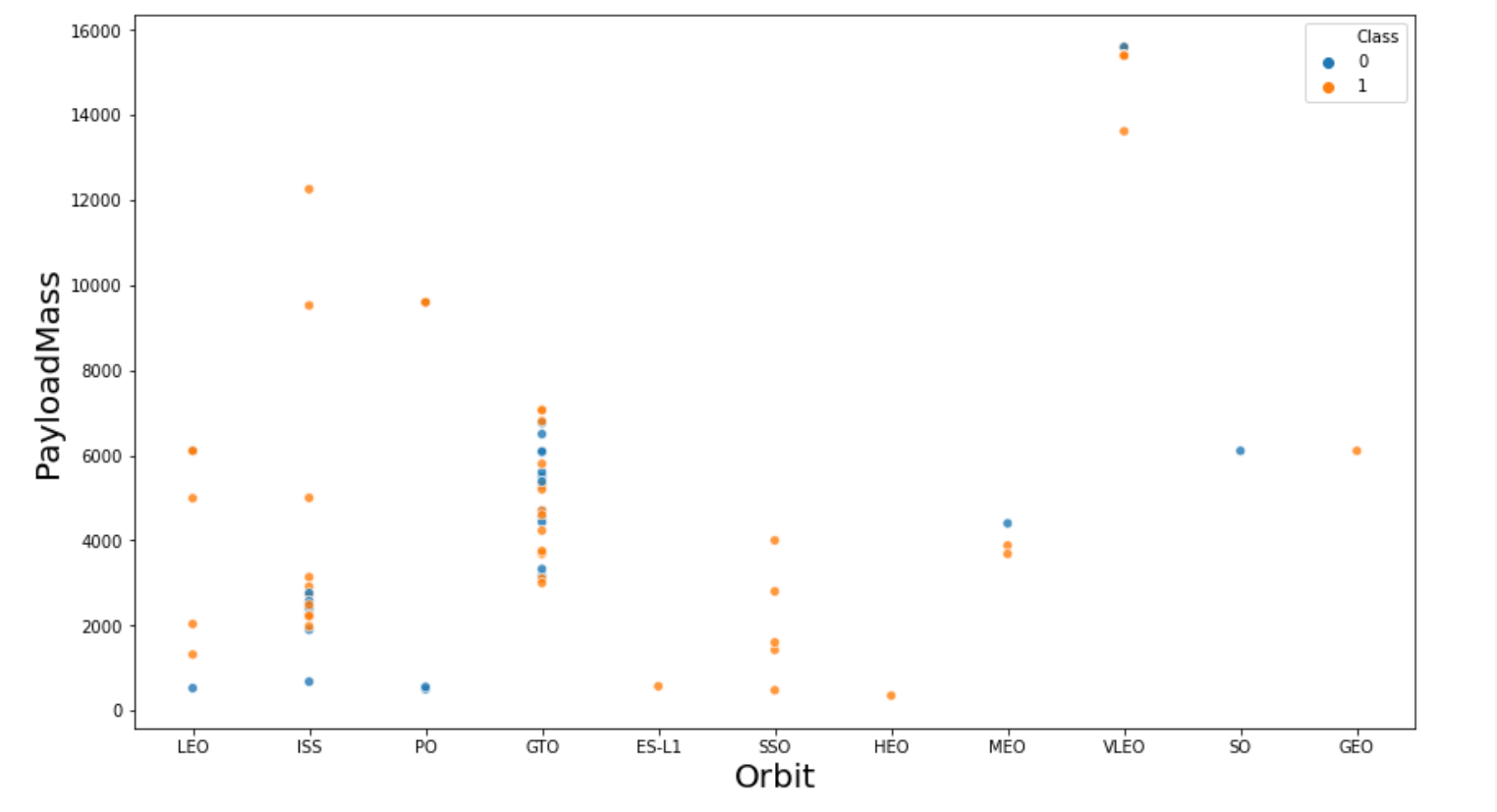
ES-L1, GEO, HEO, SSO has highest Success rates. SO has poorest.

Flight Number vs. Orbit Type



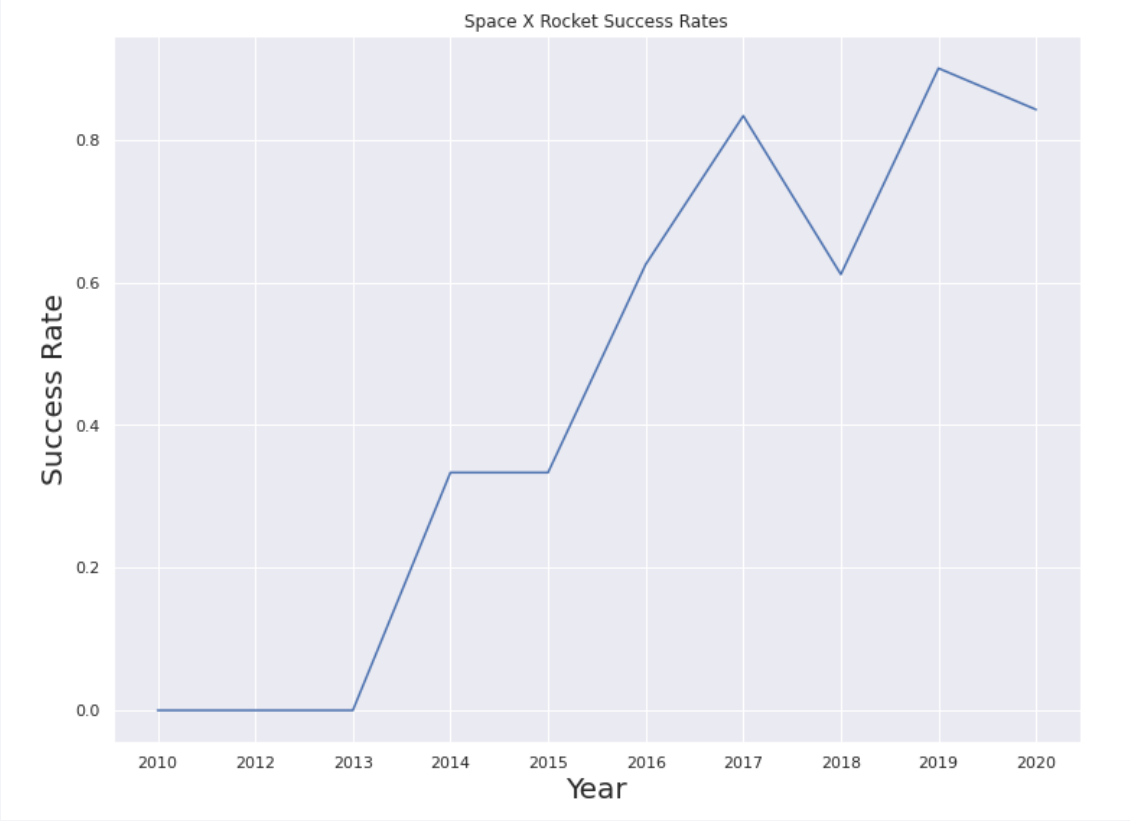
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However, for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.

Launch Success Yearly Trend



You can observe that the success rate since 2013 kept increasing till 2020.

All Launch Site Names

- Under using the key word DISTINCT it is possible to show only unique launch sites from the SpaceX data.

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

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

Launch Site Names Begin with 'CCA'

- We use SQL to display 5 records where launch sites begin with ‘CCA’.

```
%%sql
SELECT * FROM SPACEX
WHERE LAUNCH_SITE LIKE 'CCA%'
Limit 5;
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
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

- Using the function SUM to calculate the total in the PAYLOAD_MASS_KG__ column and WHERE to filter the data to get the customer 'NASA (RCS)'.

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX
WHERE CUSTOMER = 'NASA (CRS)';
```

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- Under using the function AVG we get the average in the column PAYLOAD_MASS_KG__ and the key word WHERE filters the results for only the BOOSTER_VERSION ‘F9 v1.1”.

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by booster version F9 v1.1" FROM SPACEX
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Average Payload Mass by booster version F9 v1.1
2928

First Successful Ground Landing Date

- Under using the function MIN, we find the minimum date in the DATE column and filter with WHERE the data on LANDING_OUTCOME with values ‘Success (ground pad).

```
%%sql
SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX
WHERE Landing__Outcome = 'Success (ground pad)';
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Selection of only BOOSTER_VERSION with WHERE clause LANDING_OUTCOME = ‘Success (drone ship) AND PAYLOAD_MASS_KG__ is between 4000 and 6000.

```
%%sql
SELECT BOOSTER_VERSION FROM SPACEX
WHERE LANDING__OUTCOME = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000;
```

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

Total Number of Successful and Failure Mission Outcomes

- Using a case clause within a sub query for getting both success and failure counts with one query.

```
%%sql
SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission",
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission"
FROM SPACEX;
```

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

- Under using the function MAX, we find the maximum payload in the column PAYLOAD_MASS_KG__ with a sub query and we filter with WHERE the data on BOOSTER_VERSION which had that maximum payload.

```
%%sql
SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- We list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

```
%%sql
SELECT month(DATE) as Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX
WHERE year(DATE) = '2015' AND
LANDING__OUTCOME = 'Failure (drone ship)';
```

MONTH	booster_version	launch_site
1	F9 v1.1 B1012	CCAFS LC-40
4	F9 v1.1 B1015	CCAFS LC-40

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

- We 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.

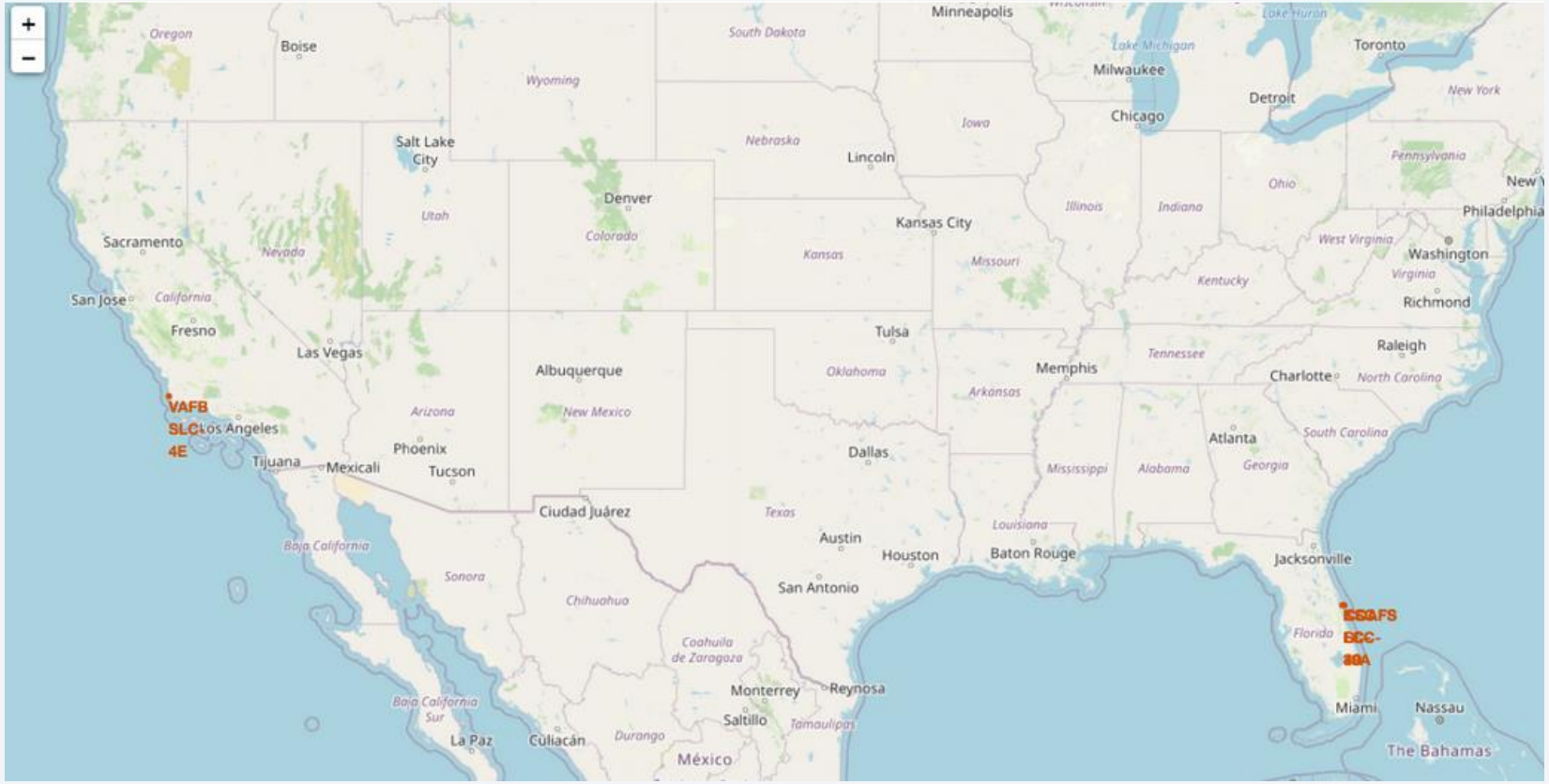
```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

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

Section 3

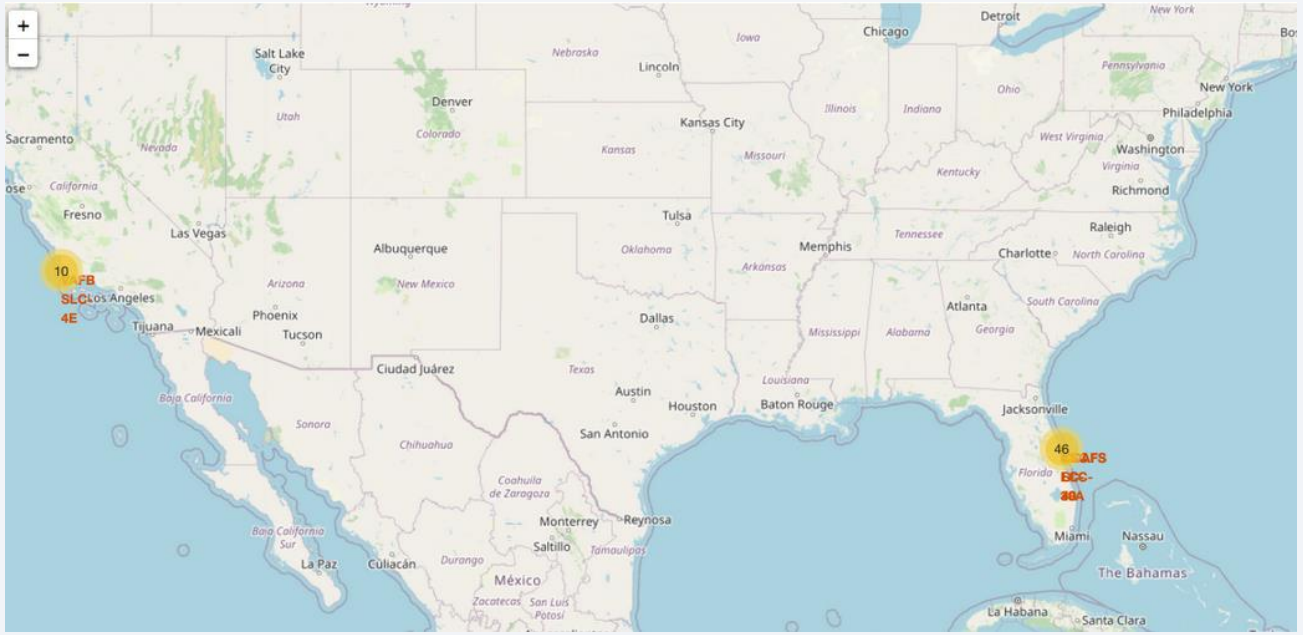
Launch Sites Proximities Analysis

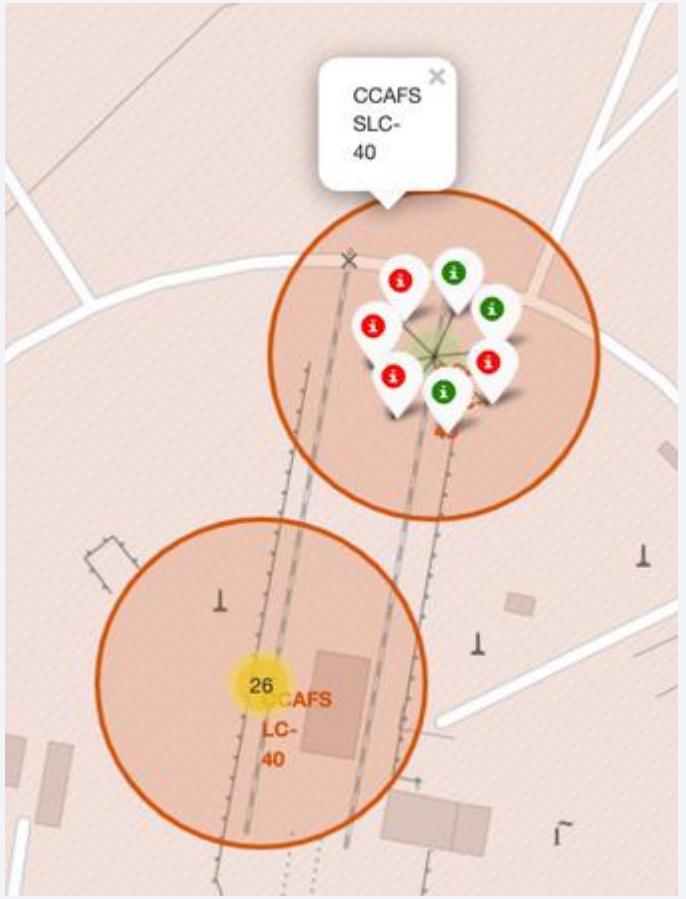
All Launch Sites on Folium



We can see that SpaceX launch sites are near the coastline of the United States (i.e., Florida and California Regions).

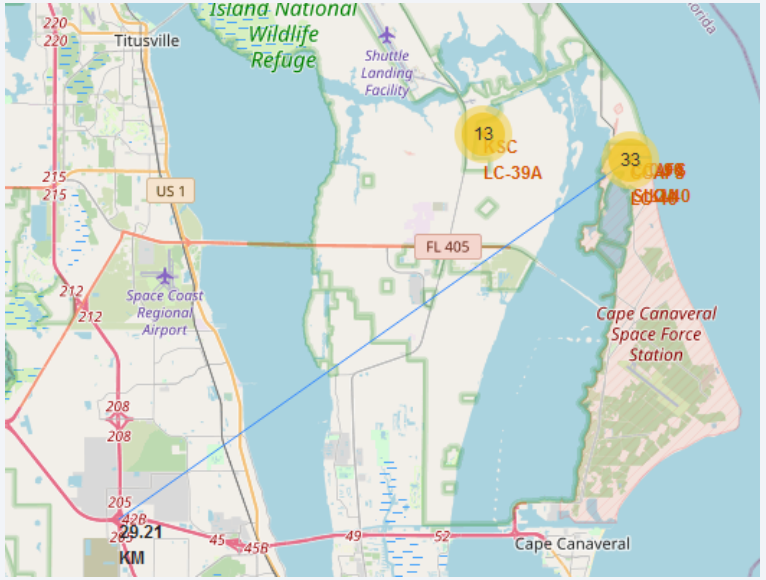
Markers showing launch sites with color labels

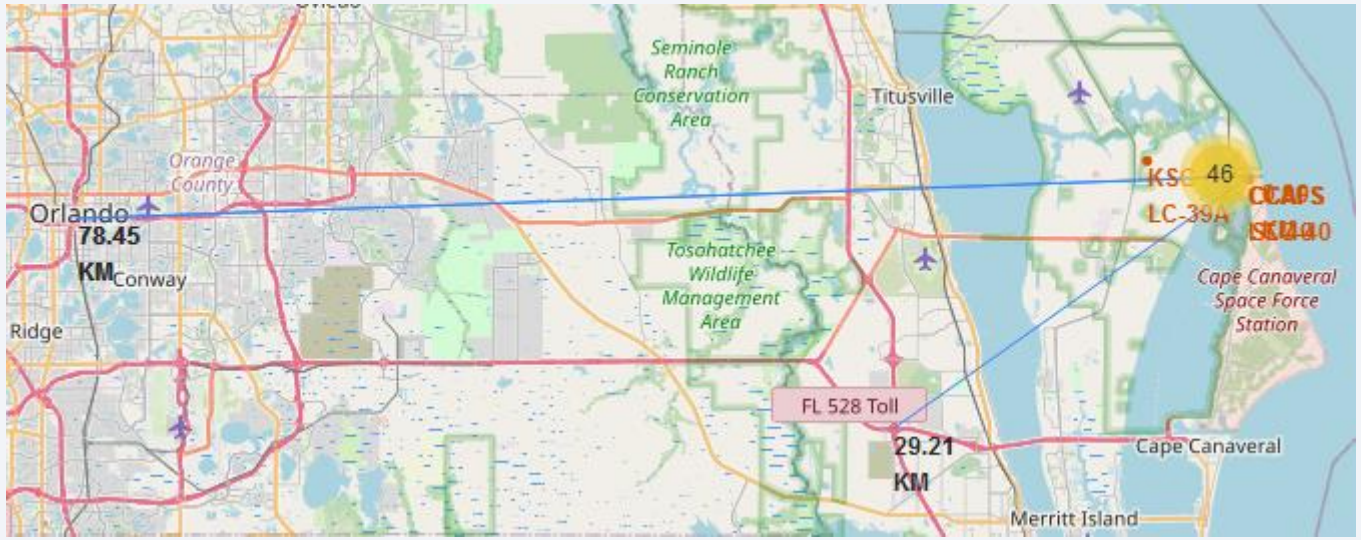




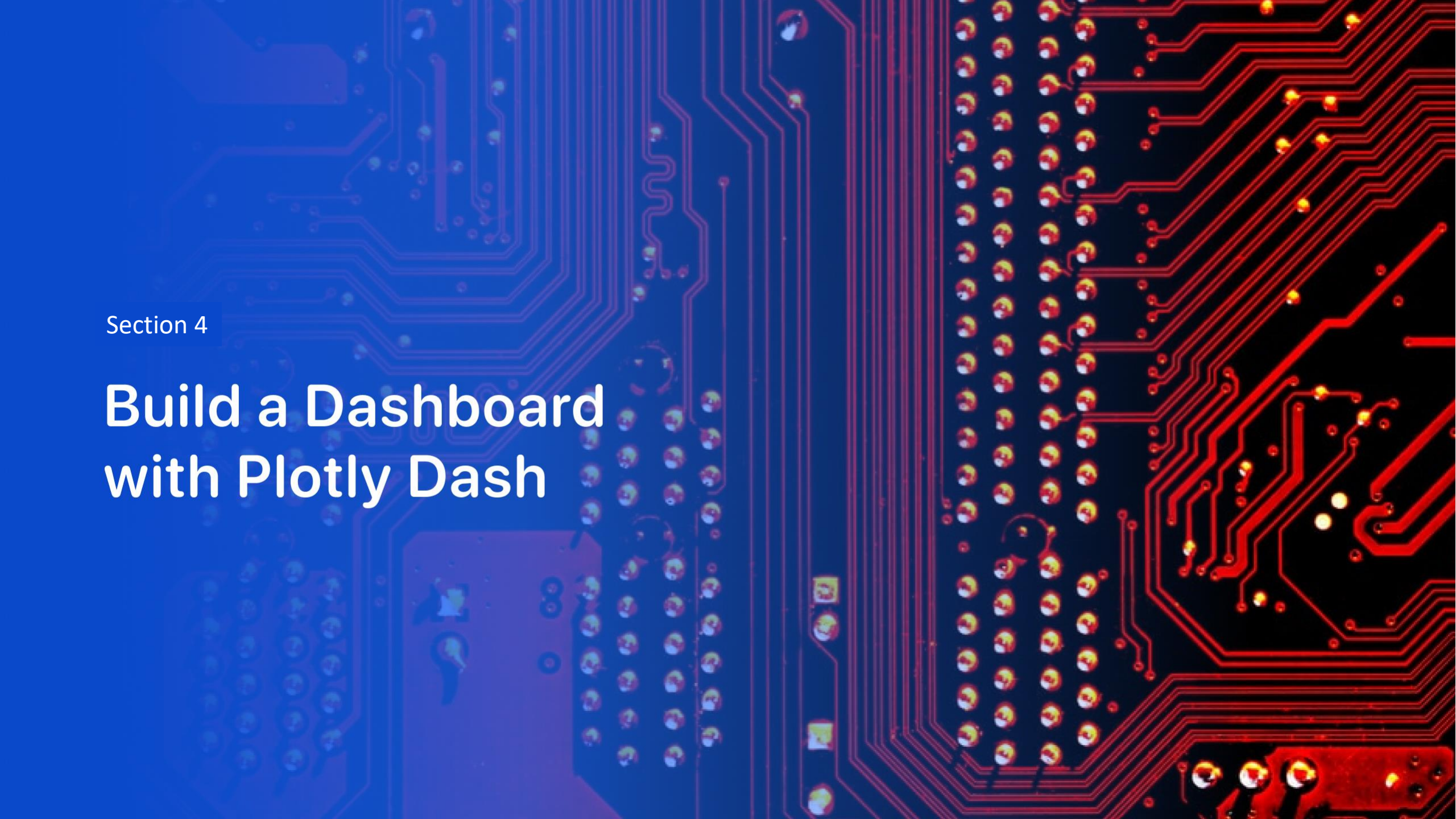
Green Marker shows successful Launches and Red Marker shows Failures.

Launch Site distance to landmarks

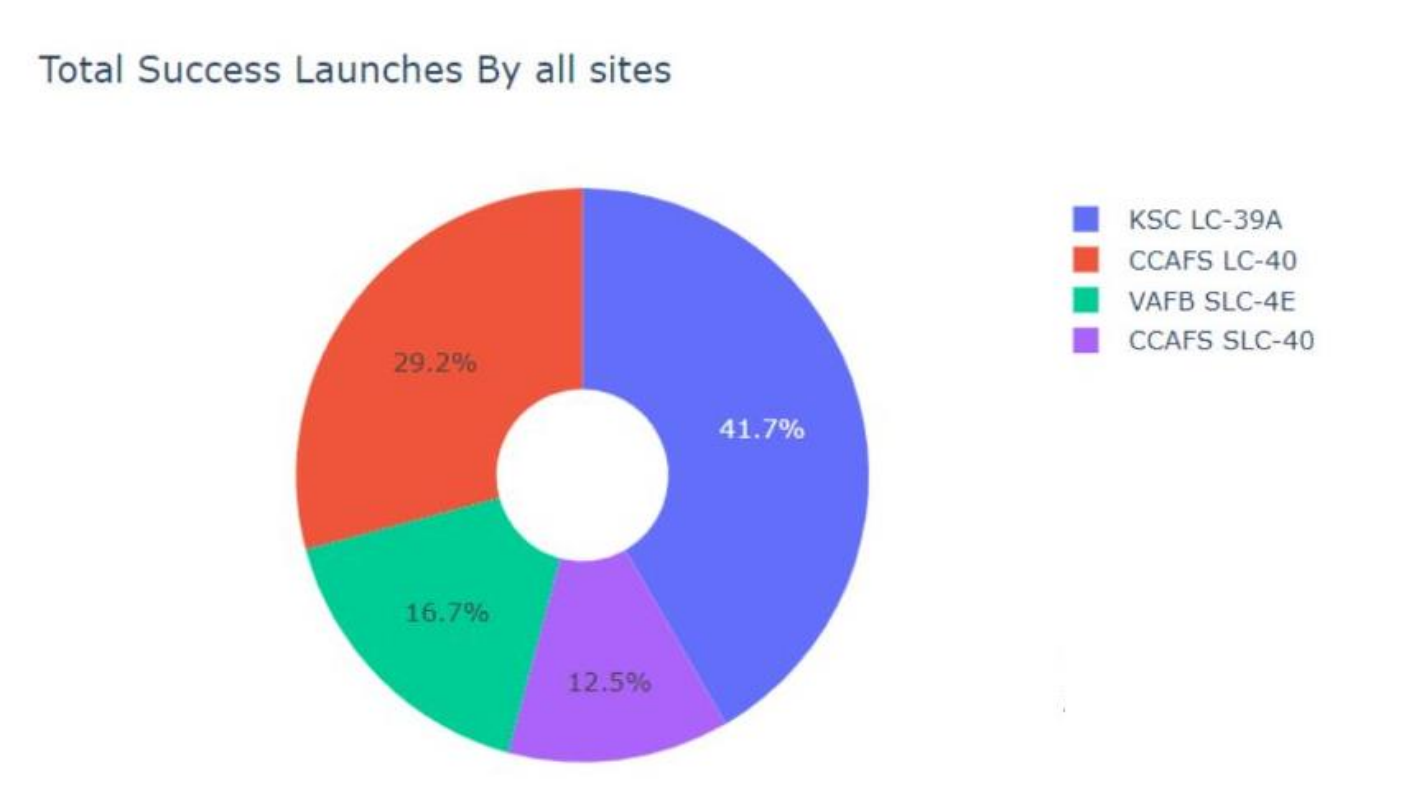




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

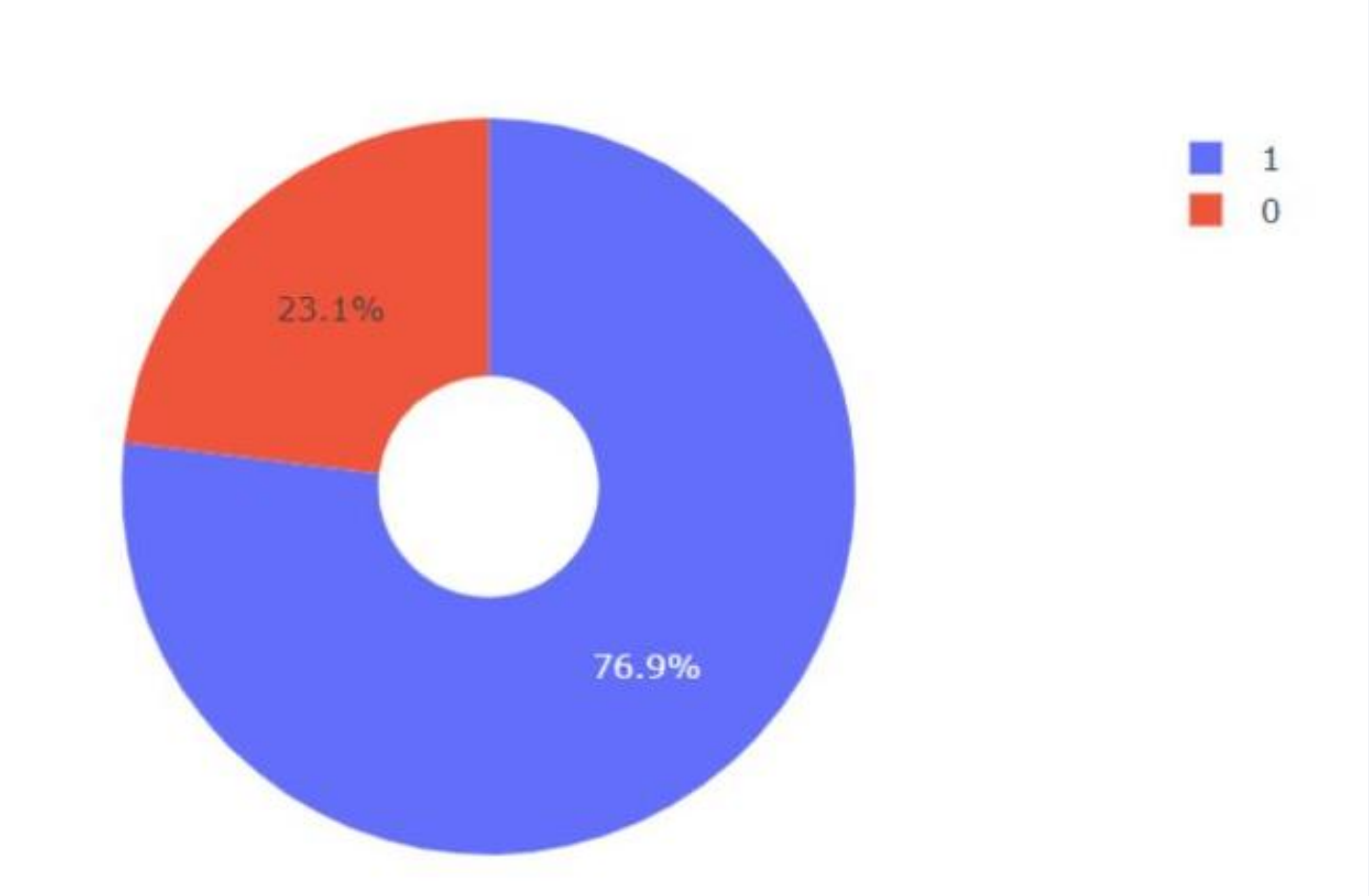

Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



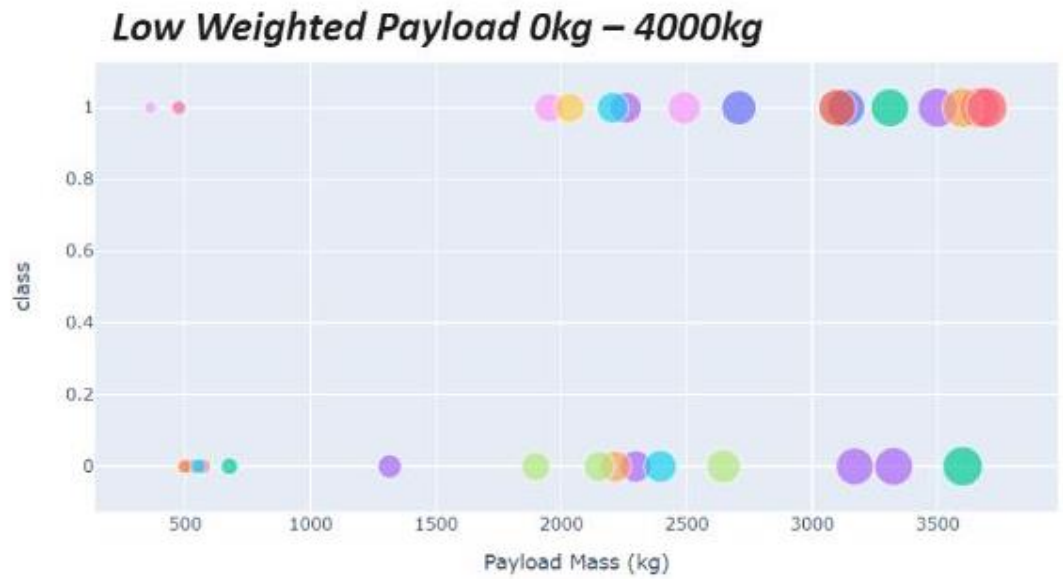
We can see that KSC LC-39A had the most successful launches from all the sites.

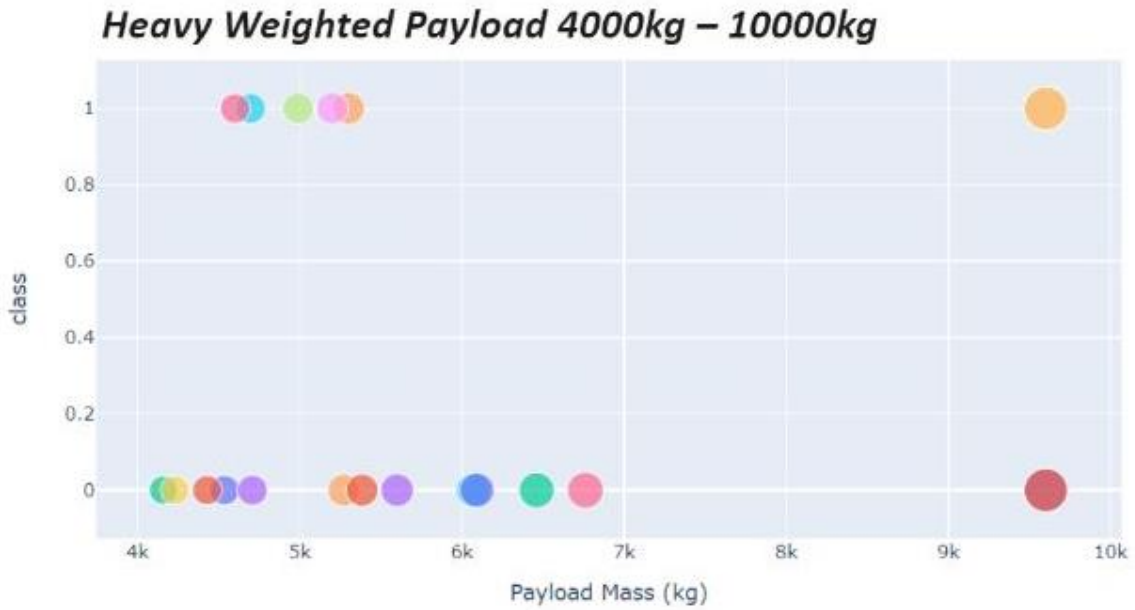
<Dashboard Screenshot 2>



KSC LC-39A achieved a 76,9% success rate while getting a 23,1% failure rate.

<Dashboard Screenshot 3>





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads.

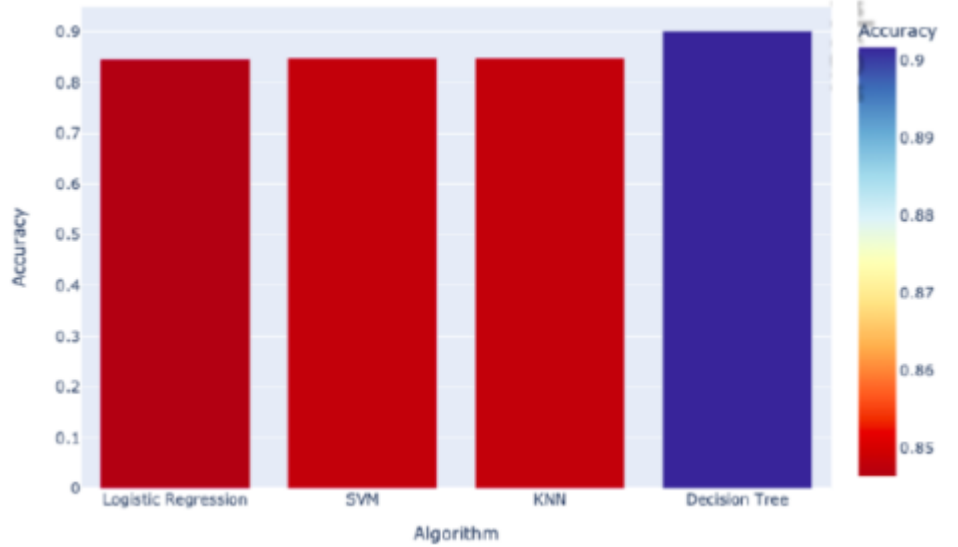
Section 5

Predictive Analysis (Classification)

Classification Accuracy

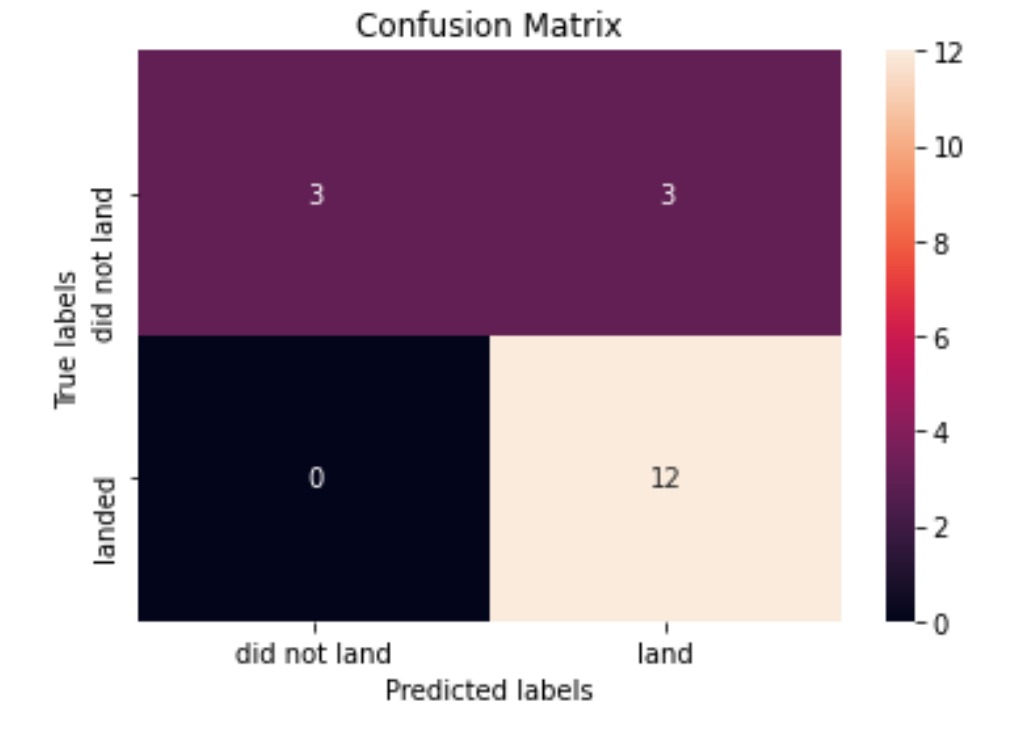
```
algorithms = {'KNN':knn_cv.best_score_,'Decision Tree':tree_cv.best_score_,'Logistic Regression':logreg_cv.best_score_,'SVM':svm_cv.best_score_}
best_algorithm = max(algorithms, key= lambda x: algorithms[x])

print('The method which performs best is \"',best_algorithm,'\" with a score of',algorithms[best_algorithm])
```

```
The method which performs best is " Decision Tree " with a score of 0.8857142857142858
```



The decision tree classifier is the model with the highest classification accuracy.

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!