IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jaymart de Leon
November 16th, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  On its website, SpaceX promotes Falcon 9 rocket launches at 62 million dollars; other suppliers charge upwards of 165 million dollars for each launch, and a large portion of the savings is due to SpaceX's ability to reuse the first stage. So, if we can figure out whether the first stage will land, we can figure out how much a launch will cost. If another business wishes to submit a proposal for a rocket launch against SpaceX, they can use this information. The project's objective is to build a machine-learning pipeline to forecast whether the first stage will successfully land.

- Problems you want to find answers
  - What elements determine whether the rocket will successfully land?
  - The way that different elements interact to affect the likelihood of a successful landing.
  - What are the requirements must be met to guarantee a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was gathered through scraping Wikipedia's website and the SpaceX API.
- Perform data wrangling
  - One-Hot Encoding was utilize to categorical features
- 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

- Describe how data sets were collected.
  - Data was collected using the get request to the SpaceX REST API
  - Next, we decoded the response content as a Json the use of .json() junction call and turn it into a pandas dataframe the use of .json_normalize().
  - I then wiped clean the facts, checked for missing values and fill in missing values where necessary.
  - Further, I finished internet scraping from Wikipedia for Falcon nine launch statistics with BeautifulSoup.
  - The goal became to extract the release data as HTML table, parse the table and convert it to an pandas dataframe for future analysis.

Data Collection – SpaceX API

- Used the get request to the SpaceX REST API calls to gather the data, clean, wrangling the data, and formatting
- GitHub URL: Data Collection

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Data Collection - Scraping

- Applied webscrapping to scrape Falcon 9 launch records with BeautifulSoup
- Then parsed the table and converted it into a pandas dataframe.

- GitHub URL: [WebScrapping]

```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

```
200
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

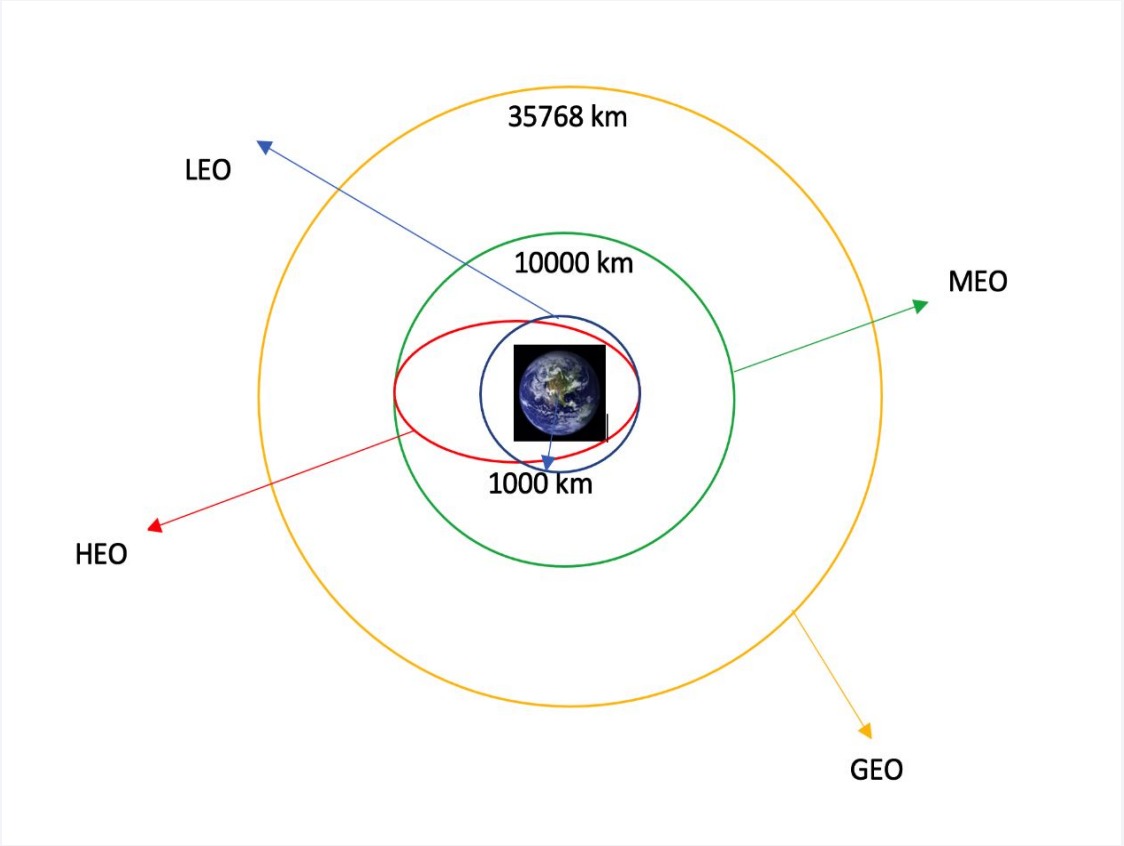
Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
soup.title
```

```
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

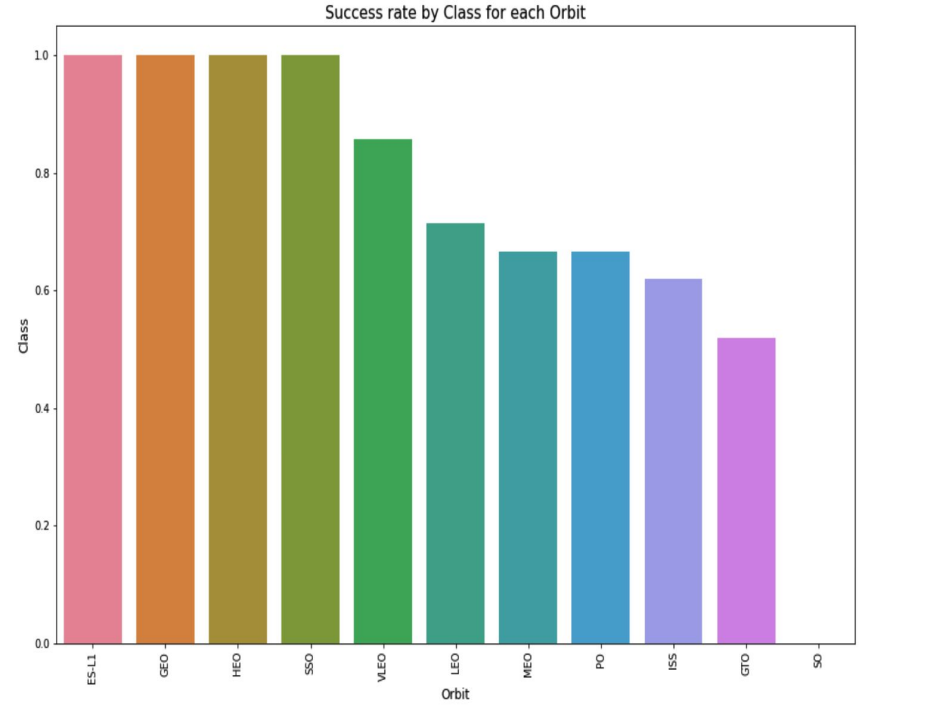
Data Wrangling

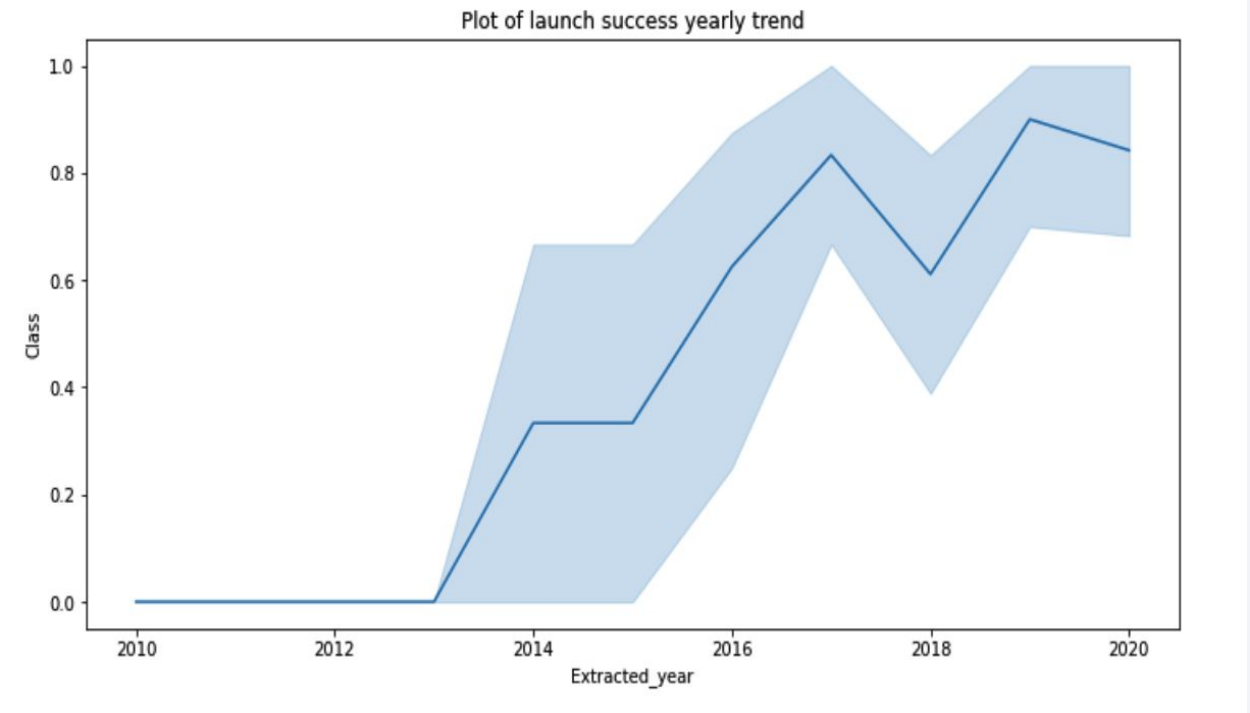
- Performed exploratory analysis and determined the training labels
- Calculated the variety of launches at each website, and wide variety and incidence of each orbits.
- Created landing outcome from outcome column and exported the results to .csv file.
- GitHub URL: [Data Wrangling]



EDA with Data Visualization

- Survey the data visualizing the connection between flight number and payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success by yearly trend
- GitHub URL: [EDA with Visualization]





EDA with SQL

- Loaded the SpaceX dataset into IBM DB2 and using the jupyter notebook to analyze the dataset
- Applied EDA with SQL to get insight from the data. Wrote to find out for instance:
  - The unique names of each lunch site for each space mission
  - Total mass of payload by boosters launched by NASA (CRS)
  - Average mass of payload by booster version F9 v1.1
  - Total successful and failure outcomes for each mission
  - The failed landing outcomes of drone ship, launch site and booster version
- GitHub URL: EDA with SQL

Build an Interactive Map with Folium

- Marked all launch sites and visualize map objects such as markers, circles, lines to mark the success or failure of each of the site on the folium map.
- Assigned the feature launch outcomes to class 0 (failure) and class 1 (success).
- Using the color-labeled marker clusters and identified which launch site have relatively high success rate.
- Calculated the distance of each of the launch site to its proximities. Example of the questions answered:
    - Are the launch sites located next to railways, highways, and coastlines?
    - Do launch sites maintain a certain distance absent from cities?

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly Dash
- Plotted pie charts showing the total launches by certain sites
- A scatter plot showing the relationship between the outcome and the payload mass in kilograms for each booster version
- GitHub URL: space_dash.py

Predictive Analysis (Classification)

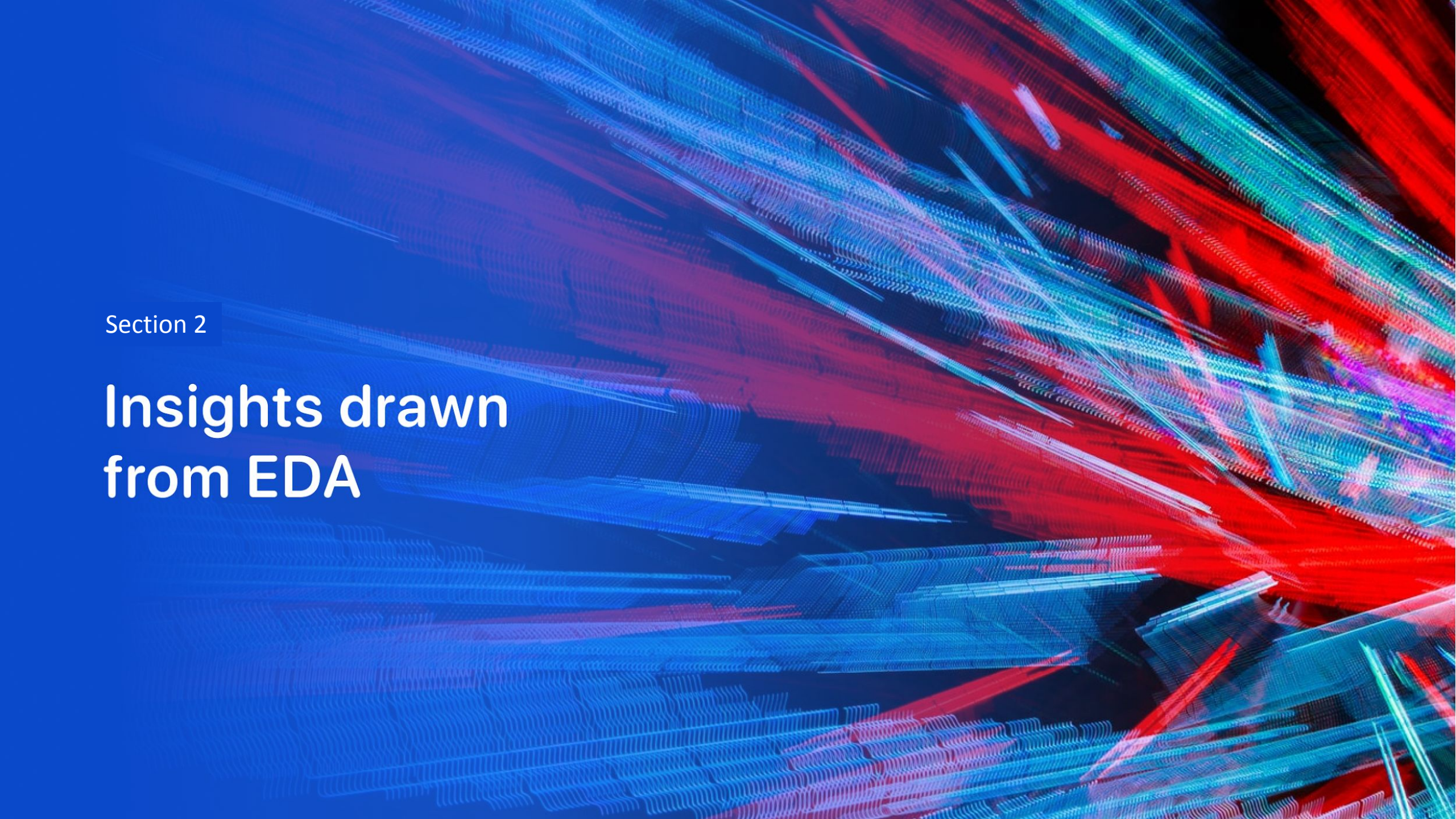
- Transformed the data and divided our data into training and testing using numpy and pandas.
- Utilizing GridSearchCV, constructed various machine learning models and adjusted various hyperparameters.
- Used accuracy as our model's metric, enhanced the model through algorithm tuning and feature engineering.
- Discovered the classification model that performed the best.
- GitHub URL: Machine Learning Predictive Analysis

Results

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

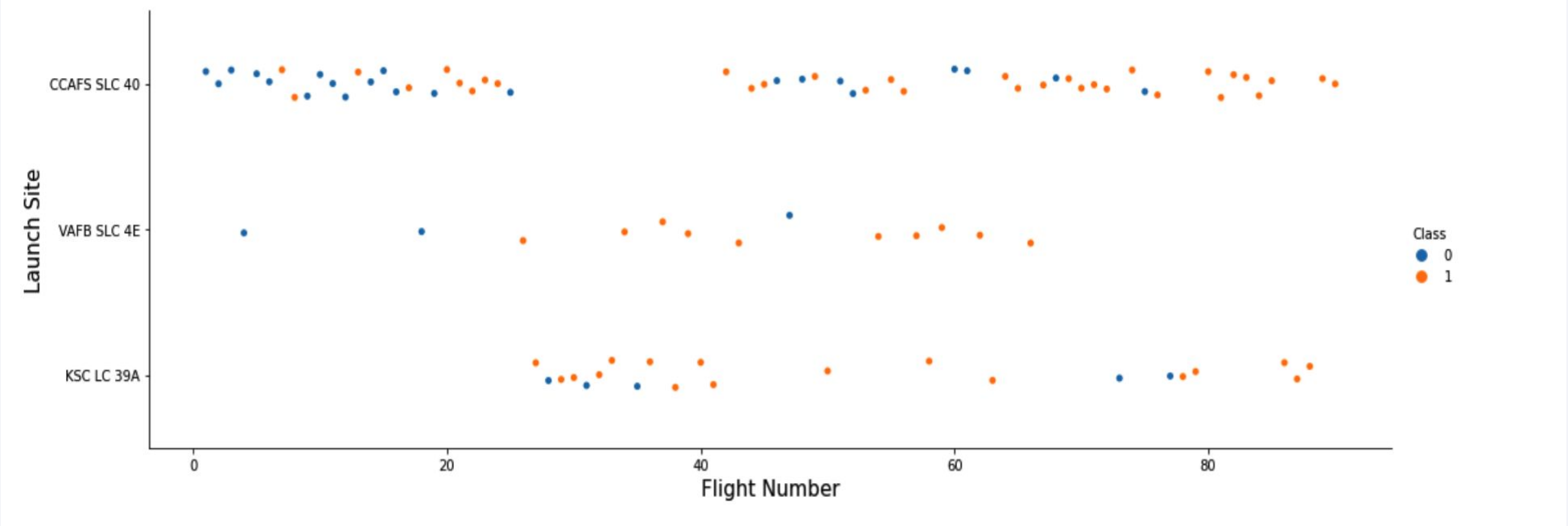
Section 2

Insights drawn from EDA



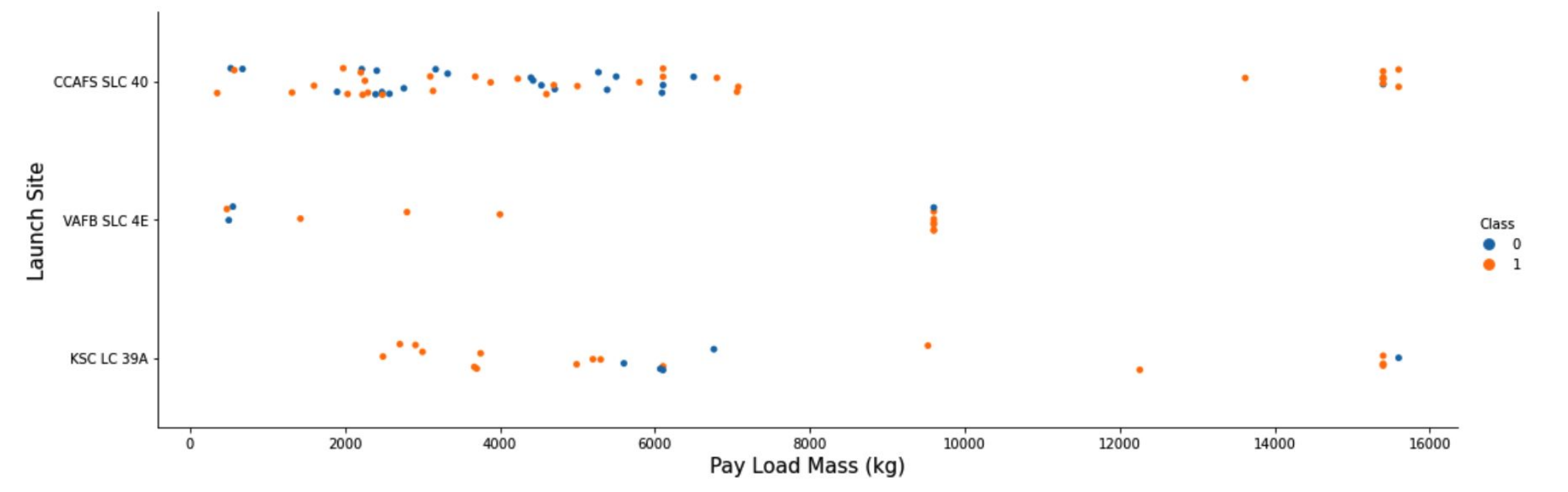
Flight Number vs. Launch Site

- Discovered from the plot that a launch site's success rate was proportional to the flight volume.



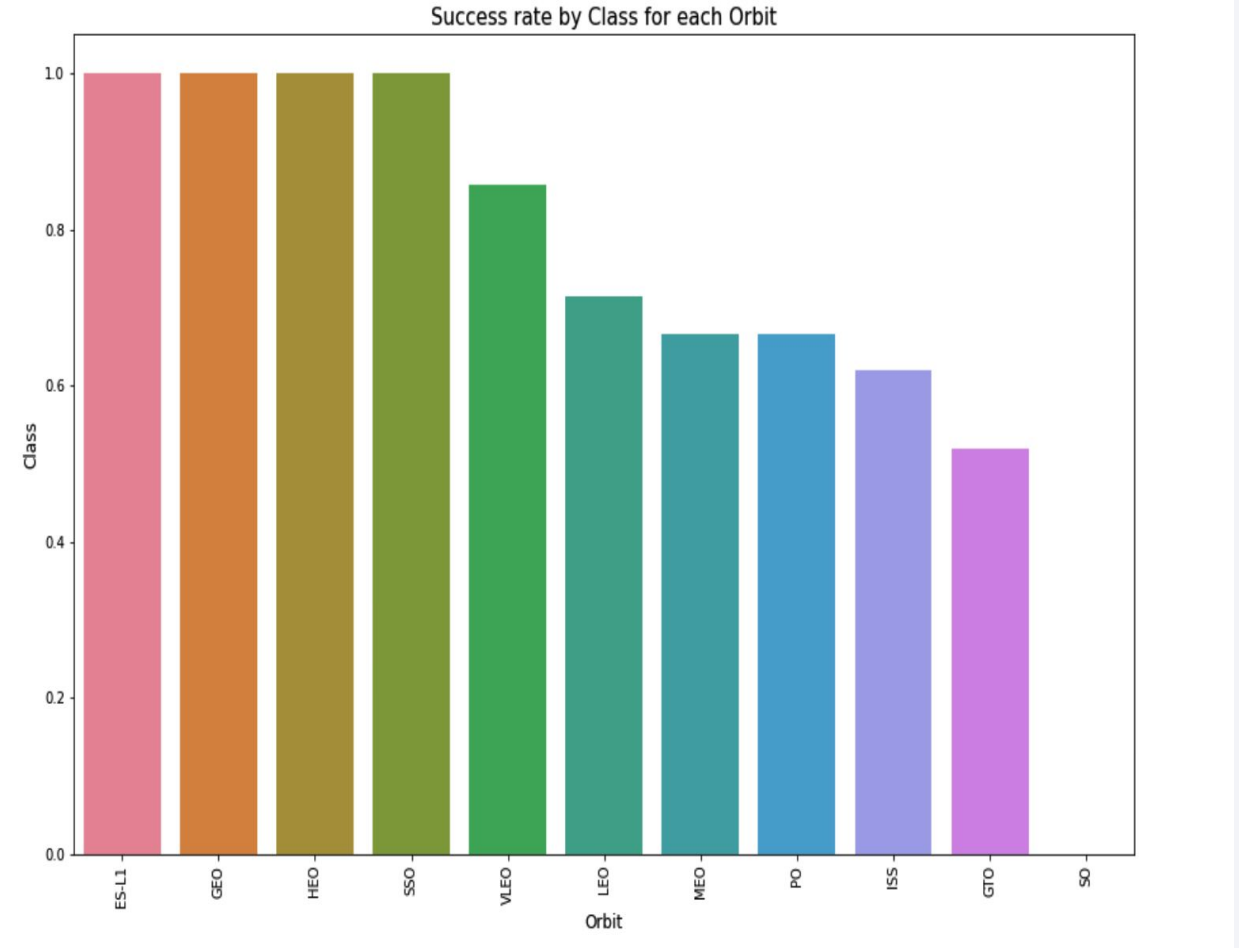
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



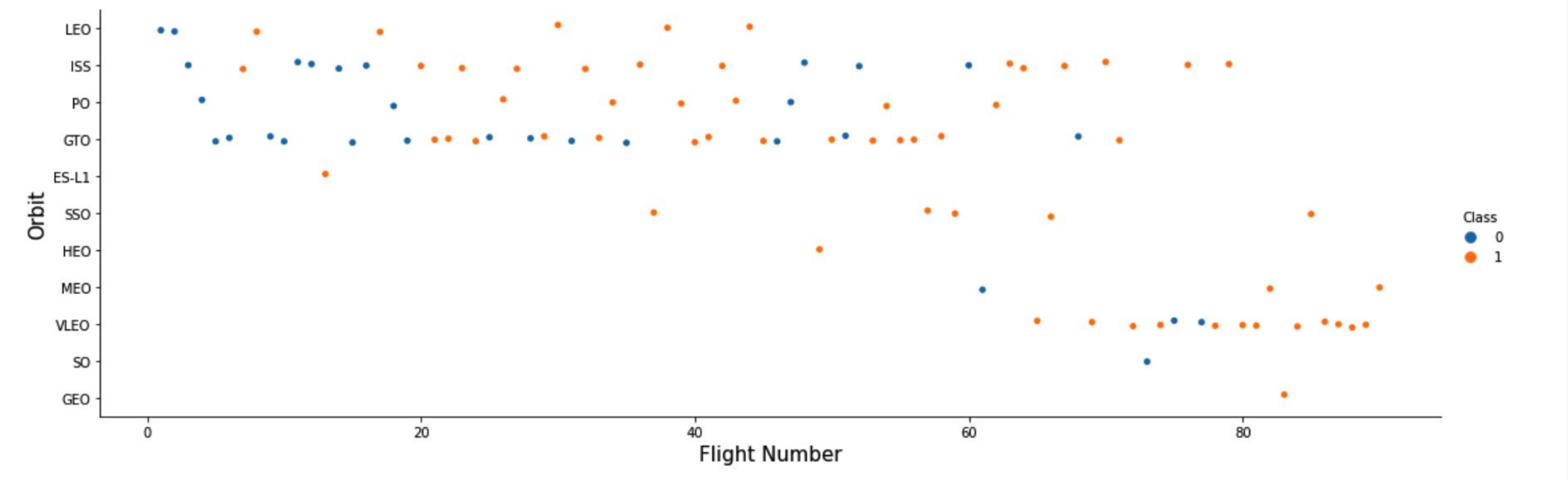
Success Rate vs. Orbit Type

- From the plot that ES-L1, GEO, HEO, SSO, and VLEO all had the highest success rates.



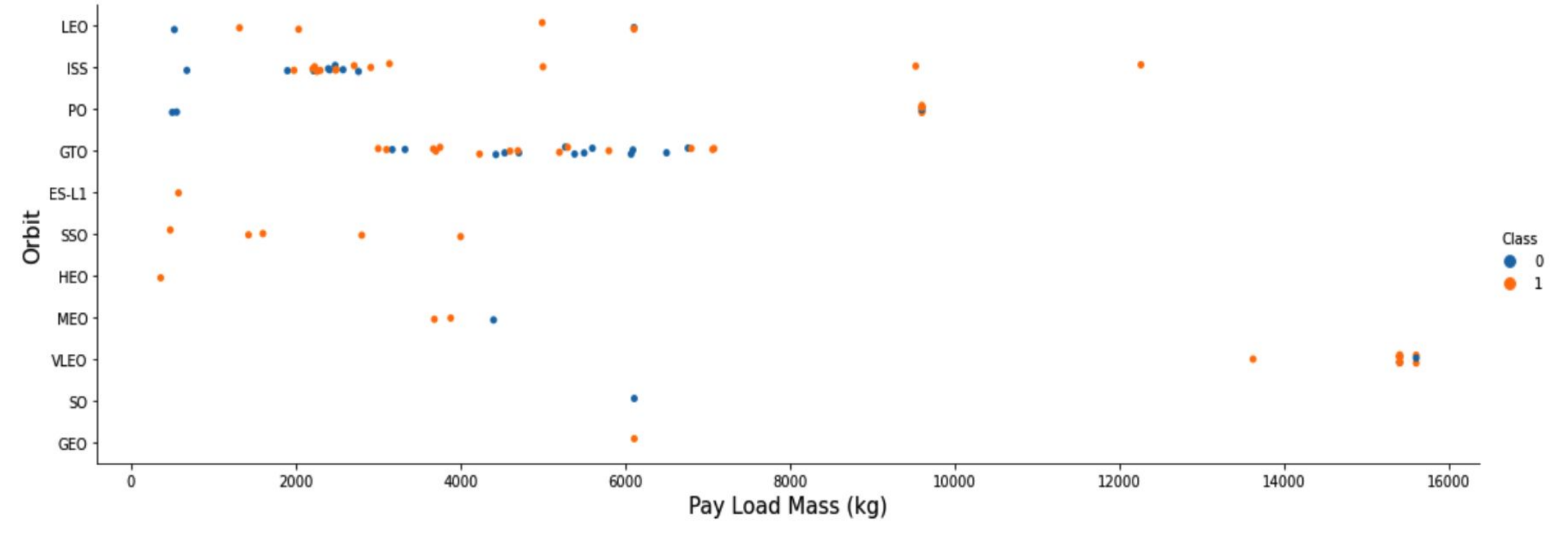
Flight Number vs. Orbit Type

- The relationship between flight number and orbit type is depicted in the plot below.We see that in the LEO orbit, achievement is connected with the number of flights though in the GTO orbit, there is no connection between flight number and the orbit.



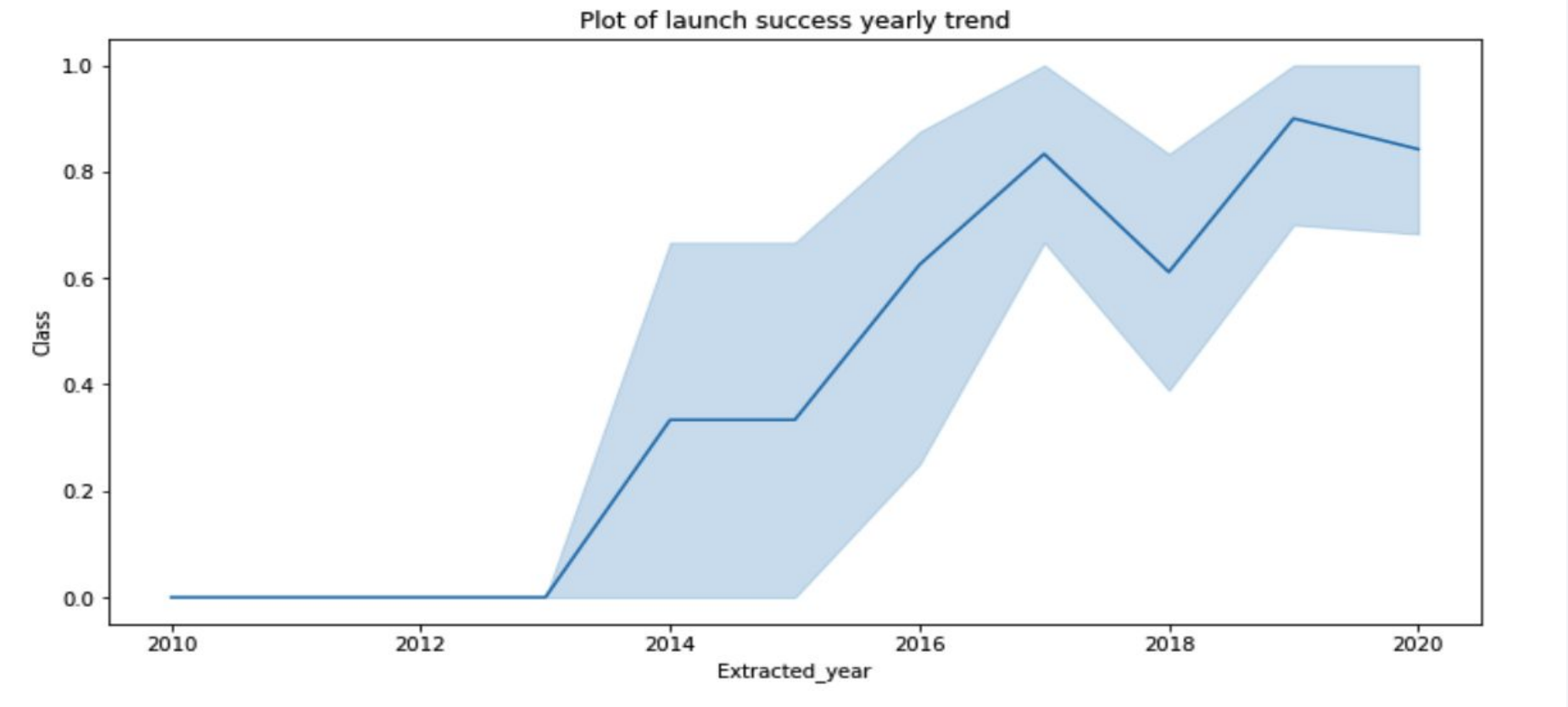
Payload vs. Orbit Type

- Observe that PO, LEO, and ISS orbits are more frequently used for successful landings of heavy payloads.



Launch Success Yearly Trend

- From the plot that the success rate has increased steadily since 2013 until 2020.



All Launch Site Names

- Used the keyword "**DISTINCT**" to display only unique SpaceX data launch sites.

```
%%sql
SELECT DISTINCT Launch_Site
FROM Spacex;
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

- Used the query above to display five records containing the launch site prefix "CCA."

```
%%sql
SELECT Launch_Site
FROM Spacex
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Using the query below, determined that the total payload carried by NASA boosters was 256163.

```
%%sql
SELECT SUM(Payload_Mass__Kg_) AS Total_PayloadMass
FROM Spacex;
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

total_payloadmass
256163

Average Payload Mass by F9 v1.1

- Calculated that the booster version F9 v1.1 carried an average of 3676 kilograms of payload

```
%%sql
SELECT AVG(Payload_Mass__Kg_) AS Average_PayloadMass
FROM Spacex
WHERE Booster_Version = 'F9 v1.1';
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

average_payloadmass
3676

First Successful Ground Landing Date

- Observed that the 5th of January 2017 marked the first successful landing on the ground pad.

```
%%sql
SELECT MIN(Date) AS First_Successful_Launch
FROM Spacex
WHERE Landing__Outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

first_successful_launch
2017-01-05

Successful Drone Ship Landing with Payload between 4000 and 6000

- Utilized the “**AND**” condition to determine a successful landing with a payload mass greater than 4000 but less than 6000 and used the “**WHERE**” clause to search for boosters that had landed on the drone ship.

```
%%sql
SELECT Booster_Version
FROM Spacex
WHERE Landing__Outcome = 'Success (drone ship)'
AND (Payload_Mass__Kg_ > 4000)
AND (Payload_Mass__Kg_ < 6000);
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

booster_version
F9 FT B1022
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Used a wildcard such as subquery and **“%”** to filter for "**WHERE**" Mission Outcome was successful or failure.

```
%%sql
SELECT COUNT(*) AS Tota_Successful_Mission,
        (
            SELECT COUNT(*)
            FROM Spacex
            WHERE Mission_Outcome LIKE '%Failure%'
        ) AS Total_Failure_Missioon
FROM Spacex
WHERE Mission_Outcome LIKE '%Success%'
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

tota_successful_mission	total_failure_missioon
45	0

Boosters Carried Maximum Payload

- Utilizing a subquery in the “**WHERE**” clause and the “**MAX()**” function, we were able to identify the booster that carried the most payload.

```
%%sql
SELECT Booster_Version, Payload_Mass__Kg_
FROM Spacex
WHERE Payload_Mass__Kg_ =
                        (
                            SELECT MAX(Payload_Mass__Kg_)
                            FROM spacex
                        )
ORDER BY Booster_Version;
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600

2015 Launch Records

- Used a combinations of the **WHERE** clause, **LIKE**, **“%”**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

```
%%sql
SELECT Landing__Outcome, Booster_Version, Launch_Site
FROM Spacex
WHERE YEAR(Date) = 2015
AND Landing__Outcome LIKE '%Failure (drone ship)%'
```

```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

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

- Selected the landing_outcome and **“COUNT”** of the landing outcomes from the data and **“RANK”** count of each of the landing outcomes and **“ORDER BY”** greatest to least and used **“WHERE”** to filter **“YEAR”** **“BETWEEN”** 2010 and 2017.
- Applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
%%sql
SELECT Landing__Outcome,
        COUNT(*) AS Total_Number ,
        RANK() OVER(ORDER BY COUNT(*) DESC) AS RANK
FROM Spacex
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing__Outcome
ORDER BY Total_Number DESC;
```

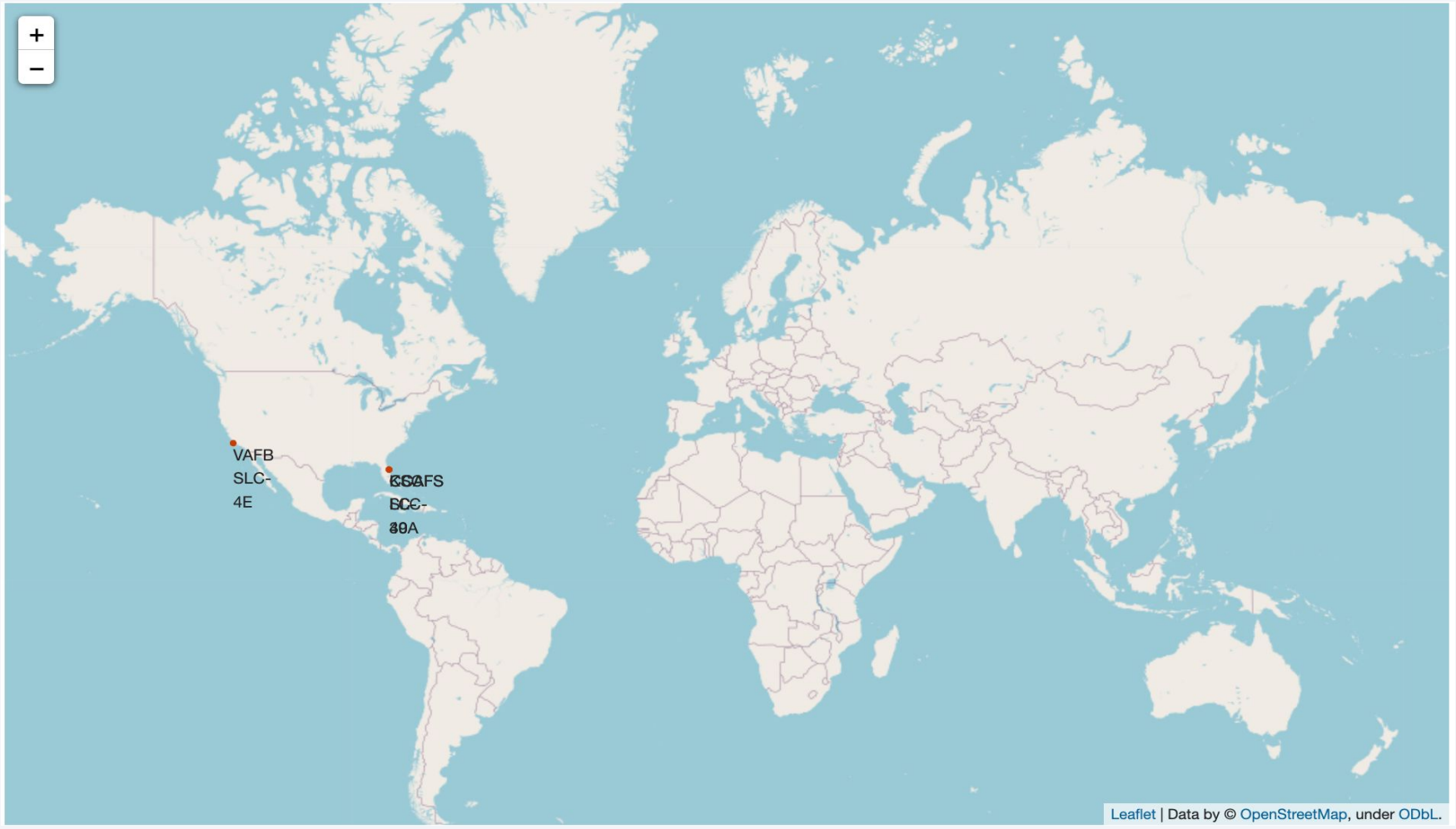
```
 * ibm_db_sa://cby17810:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

landing__outcome	total_number	RANK
No attempt	7	1
Failure (drone ship)	2	2
Success (drone ship)	2	2
Success (ground pad)	2	2
Controlled (ocean)	1	5
Failure (parachute)	1	5


Section 3
Launch Sites
Proximities Analysis

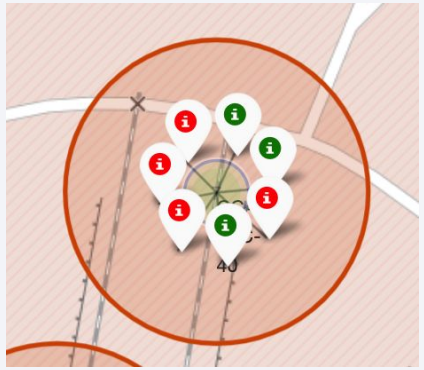
Location of all the launch sites global markers

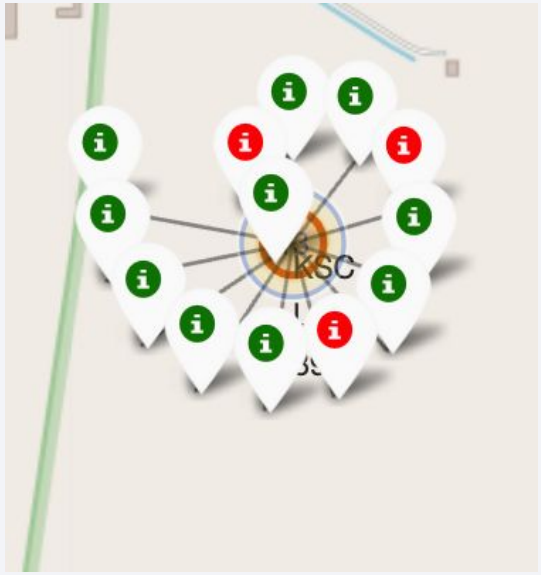
All the launch sites are located in the US along the coasts of California and Florida



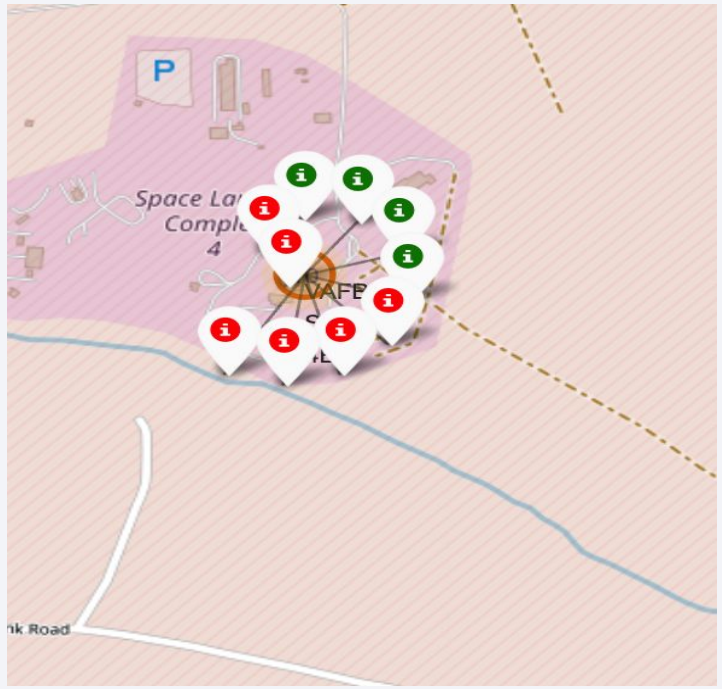
Location of the launch site with markers and color labels

Florida Launch Sites



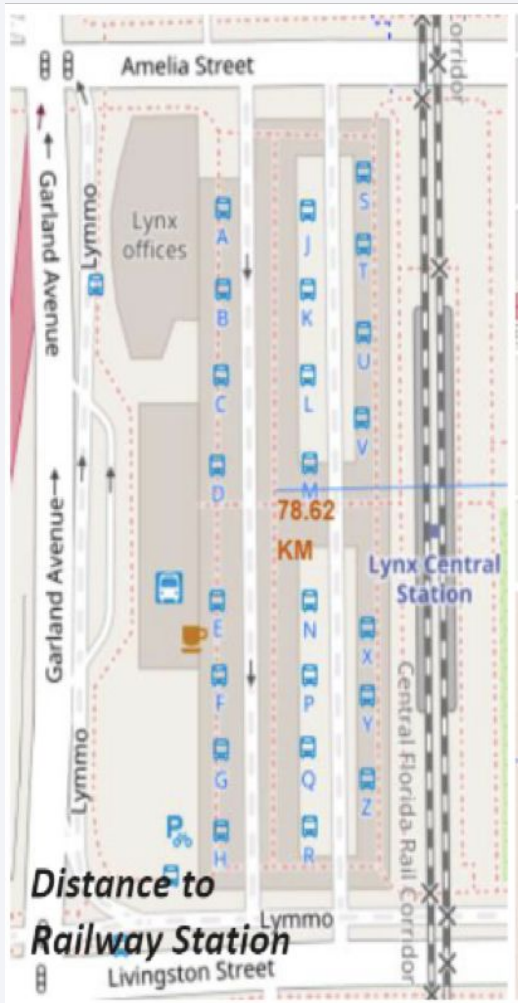


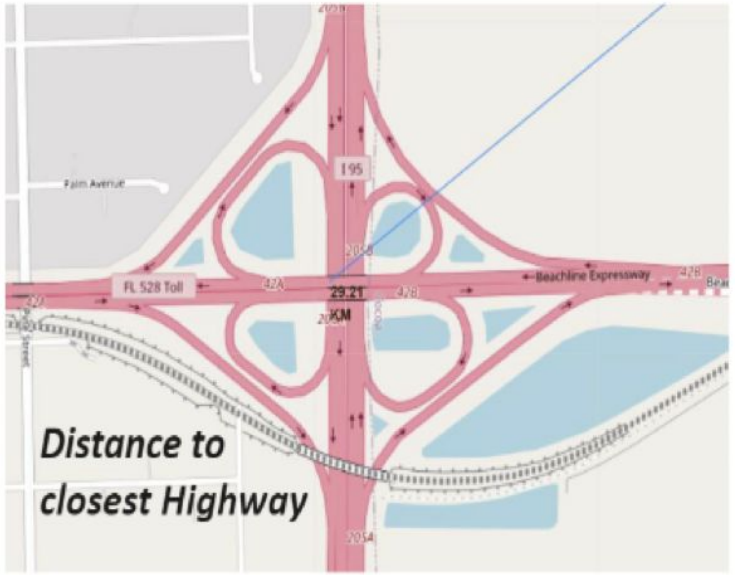
California Launch Sites

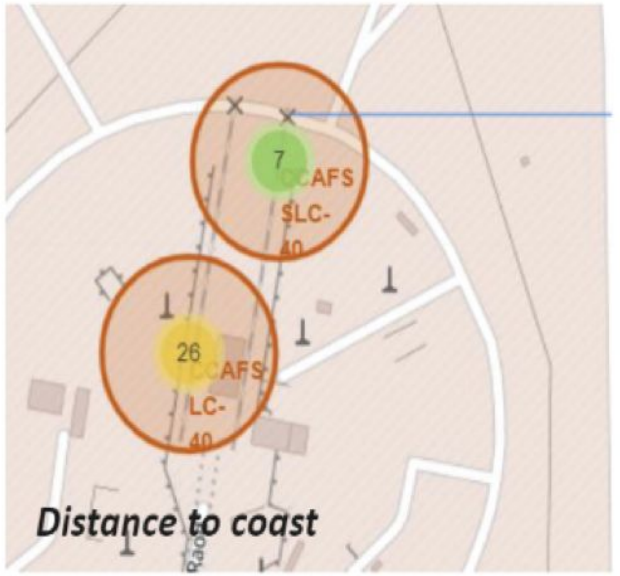


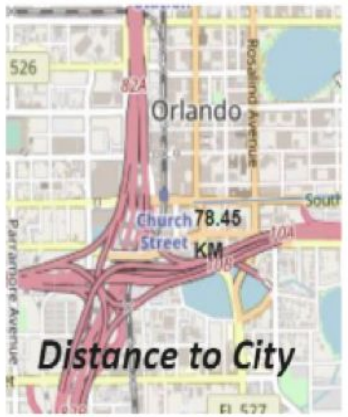
Green Shows *Successful* and Red Shows *Failed* launches

Railway, Highway, Coastline, and City distance to the Launch Site

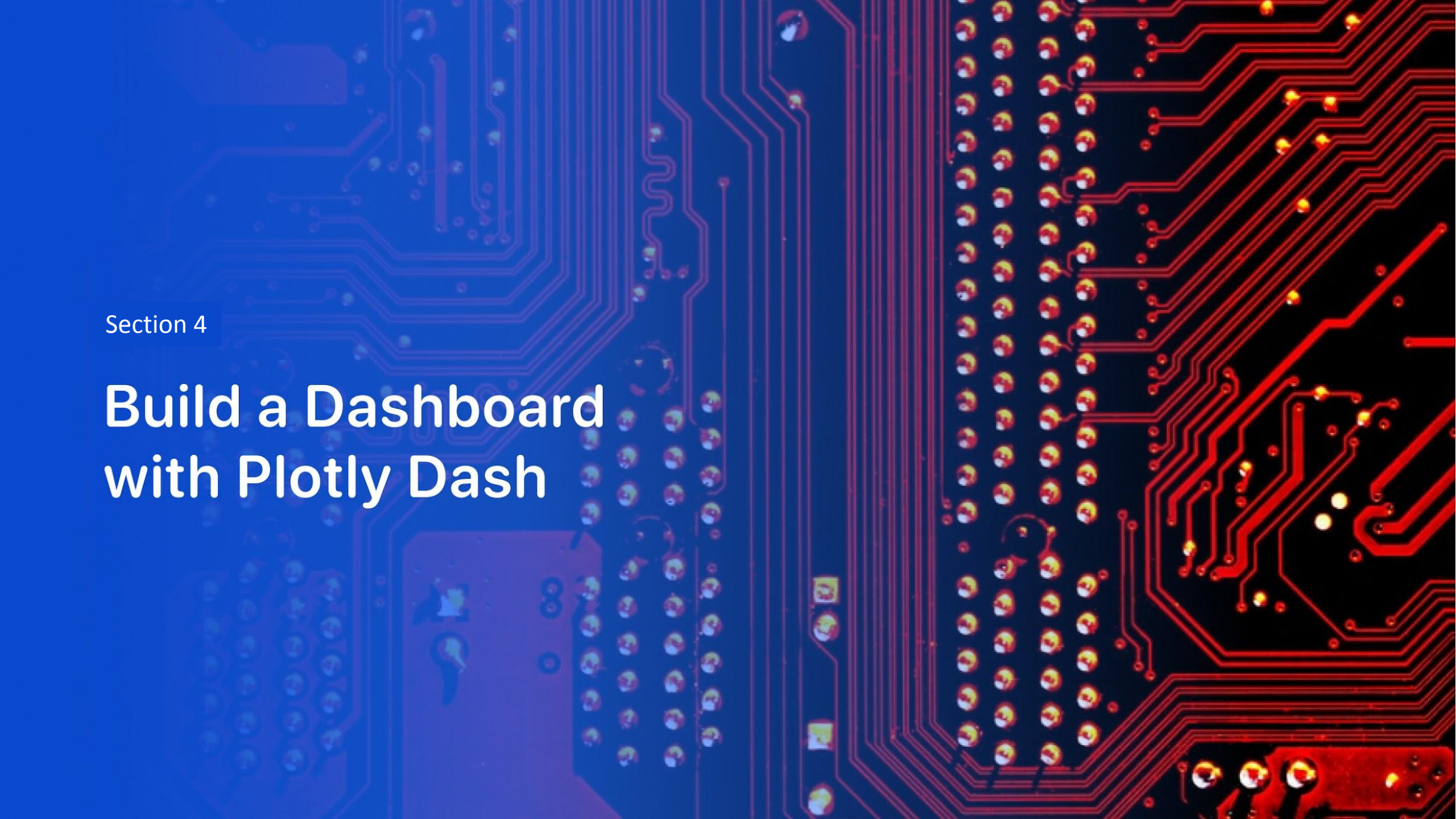




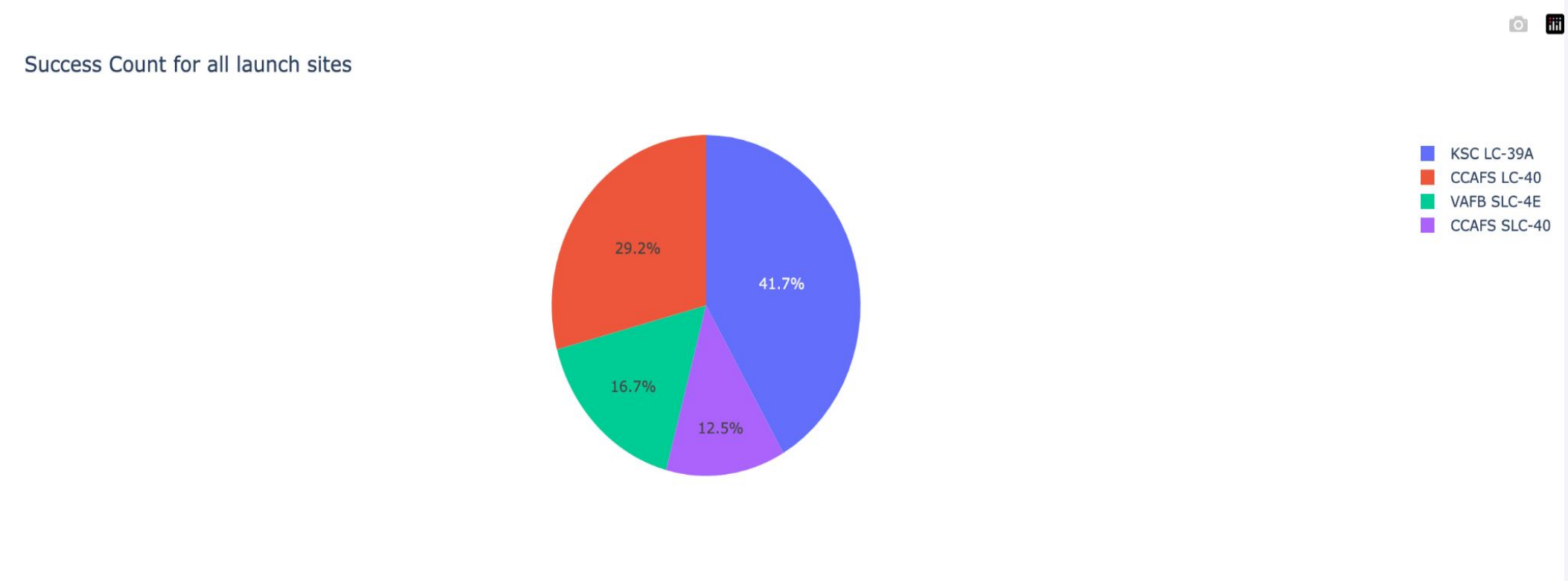




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

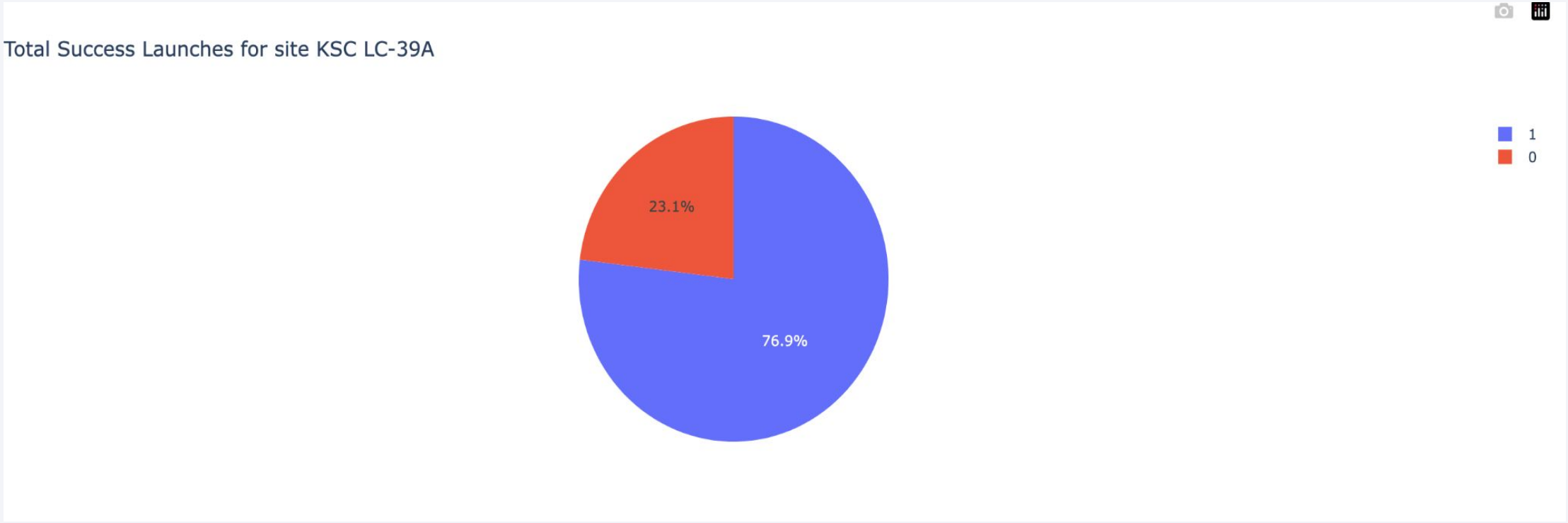

Section 4
Build a Dashboard with Plotly Dash

The pie chart shows the success of each of the launch site by percentage



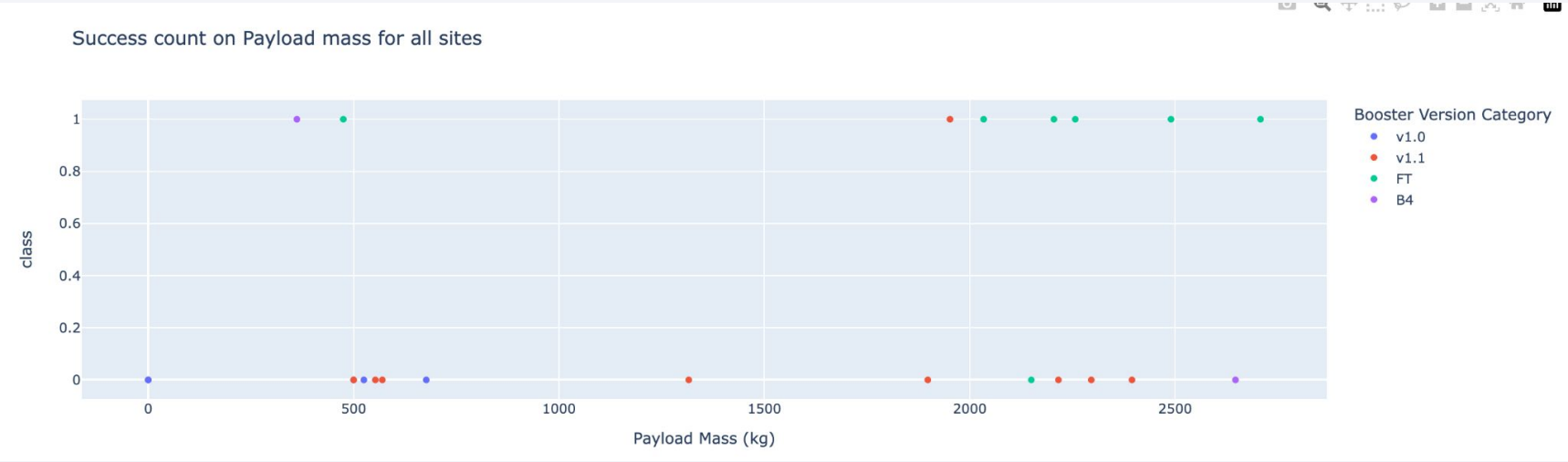
The pie chart shows the launch site with the highest launch success ratio

KSC LC-39A achieved a 76.9% success rate and 23.1% failure rate



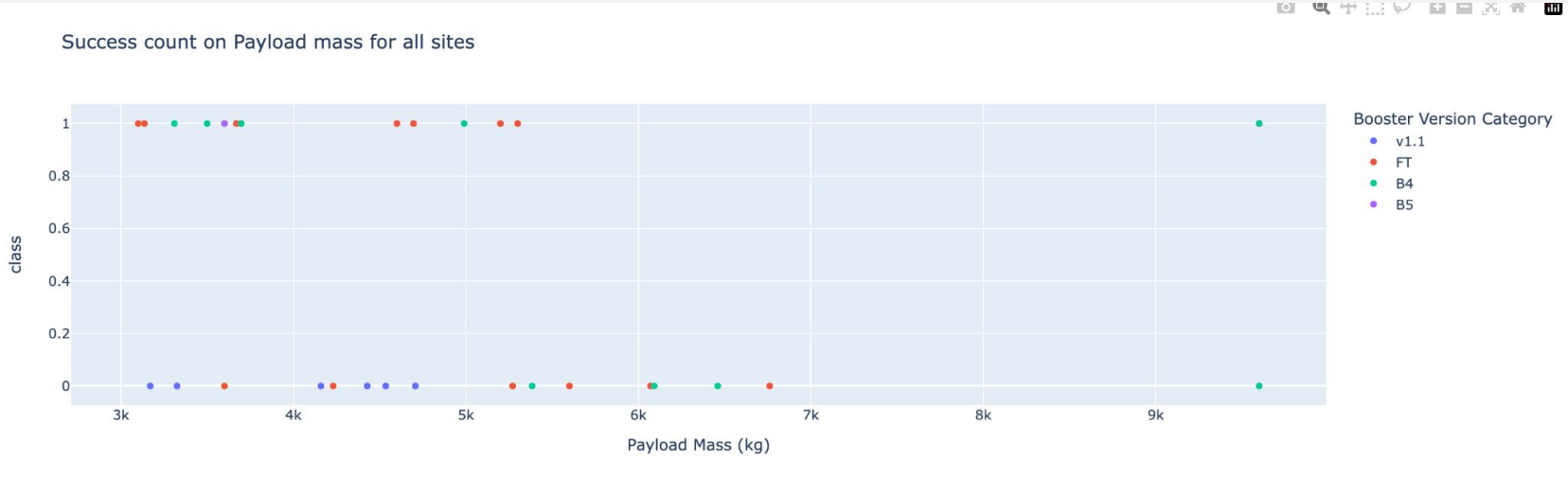
The scatter plots shows Payload vs Launch Outcome for all sites, with different payload mass selected by the range slider

These two scatter plots shows the success rate of the payload ranging from 0 - 3000 kg is higher than 3000 - 1000 kg



Payload between 0 kg to 3000 kg

Payload between 3000 kg to 10000 kg



Section 5

Predictive Analysis (Classification)

Classification Accuracy

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

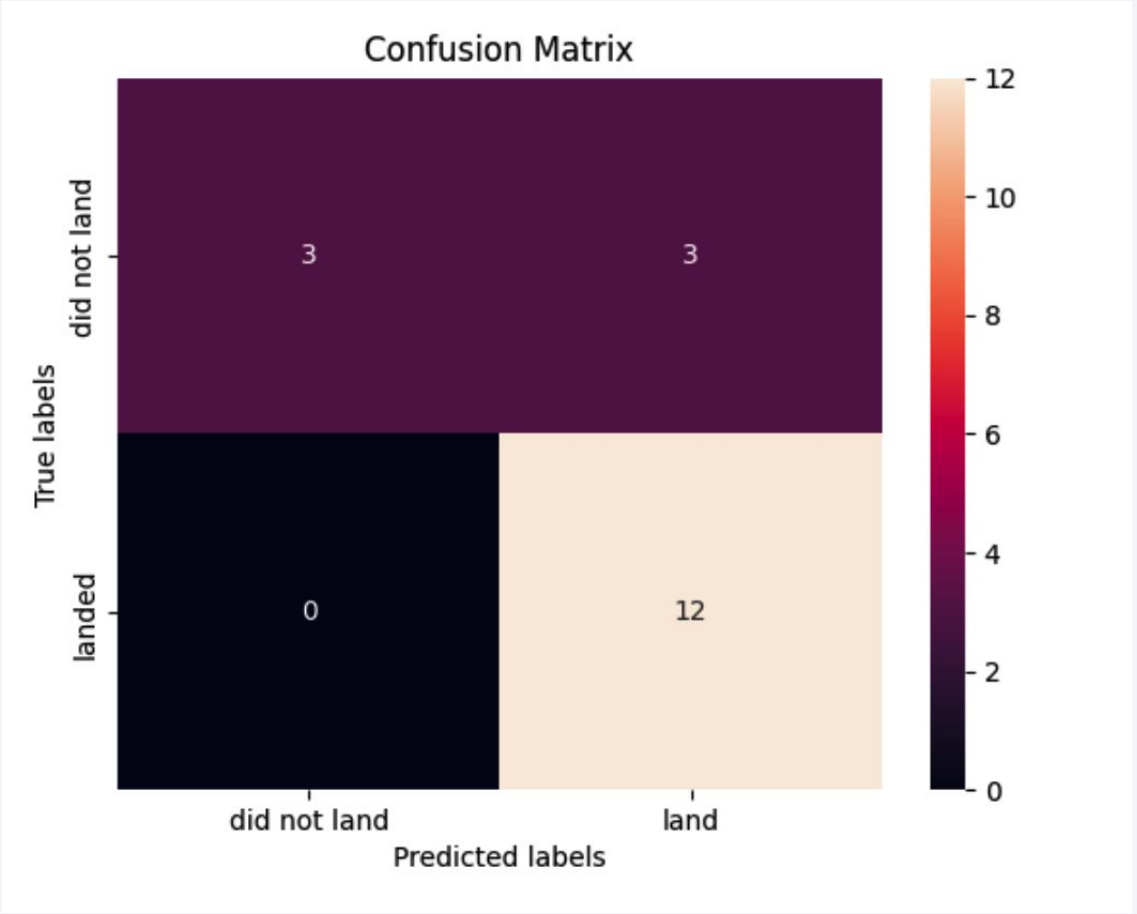
```
Models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(Models, key=Models.get)
print('The best model is', bestalgorithm,'with a score of', Models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('The best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('The best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('The best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('The best params is :', svm_cv.best_params_)
```

```
The best model is DecisionTree with a score of 0.8732142857142856
The best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitte
r': 'random'}
```

Confusion Matrix

- The decision tree classifier confusion matrix demonstrates its ability to differentiate between the various classes. The biggest issue is false positives or failed landing is considered successful landing by the classifier.



Conclusions

In Conclusion:

- The greatest success rate at a lunch site are the ones with larger flight amount.
- From 2013 to 2020, the launch success rate has increased.
- The most successful orbits were the ES-L1, GEO, HEO, SSO, and VLEO ones.
- Launches at KSC LC-39A were the most successful of any site.
- For this job, the best machine learning algorithm is the Decision Tree Classifier.

Thank you!