IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ahmed Bakhit
2/26/2023

Outline

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

Executive Summary

Summary of methodologies

- Data Collection with SQL, API, and Web Scraping
- Data Analysis and Wrangling
- Data Visualization via Interactive Maps
- Predictive Analysis Via Machine Learning

Summary of all results

- Data Analysis with Visualizations
- Best model for Predictive Analysis

Introduction

Project background and context

- Our goal is to determine the likelihood of a successful landing for the first stage of the Falcon 9 rocket. This rocket is advertised on SpaceX's website and is priced at 62 million dollars, a significant cost savings compared to other providers whose rockets cost over 165 million dollars each. This cost savings is achieved in part because SpaceX can reuse the first stage of their rocket. By accurately predicting the likelihood of a successful first stage landing, this information can be useful for other companies that may wish to bid against SpaceX for a rocket launch contract.

Problems you want to find answers

- With What factors will the rocket land successfully
- The effect of each relationship of rocket variables on outcome
- Conditions which will aid SpaceX have to achieve the best results

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using Webscarping and API
- Perform data wrangling
  - One-hot encoding was used for 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

- The SpaceX API was utilized to collect data through a GET request. The response content was decoded as Json using the .json() function and then transformed into a pandas dataframe using .json_normalize(). Following this, the data was cleaned and any missing values were identified and subsequently filled in. Additionally, BeautifulSoup was utilized for web scraping from Wikipedia to extract Falcon 9 launch records. The aim was to extract the launch records as an HTML table, parse the table, and convert it into a pandas dataframe for subsequent analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- Here's the link: https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/84670413-8139-4458-8e7d-edbc2ece9368/view?projectid=d6587a5a-40d8-46d6-b04a-2a562a733026&context=cpdaas

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Bl
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	
...	...	...	...	...	...	...	...	...	...	...	...	...	
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	

90 rows × 17 columns

Data Collection - Scraping

- We utilized BeautifulSoup for web scraping to extract Falcon 9 launch records. After extracting the data, we parsed the table and transformed it into a pandas dataframe.
- Here's the link: https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/84670413-8139-4458-8e7d-edbc2ece9368/view?projectid=d6587a5a-40d8-46d6-b04a-2a562a733026&context=cpdaas

In [10]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_c
```

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

In [11]:
```
response.status_code
```

Out[11]: 200

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

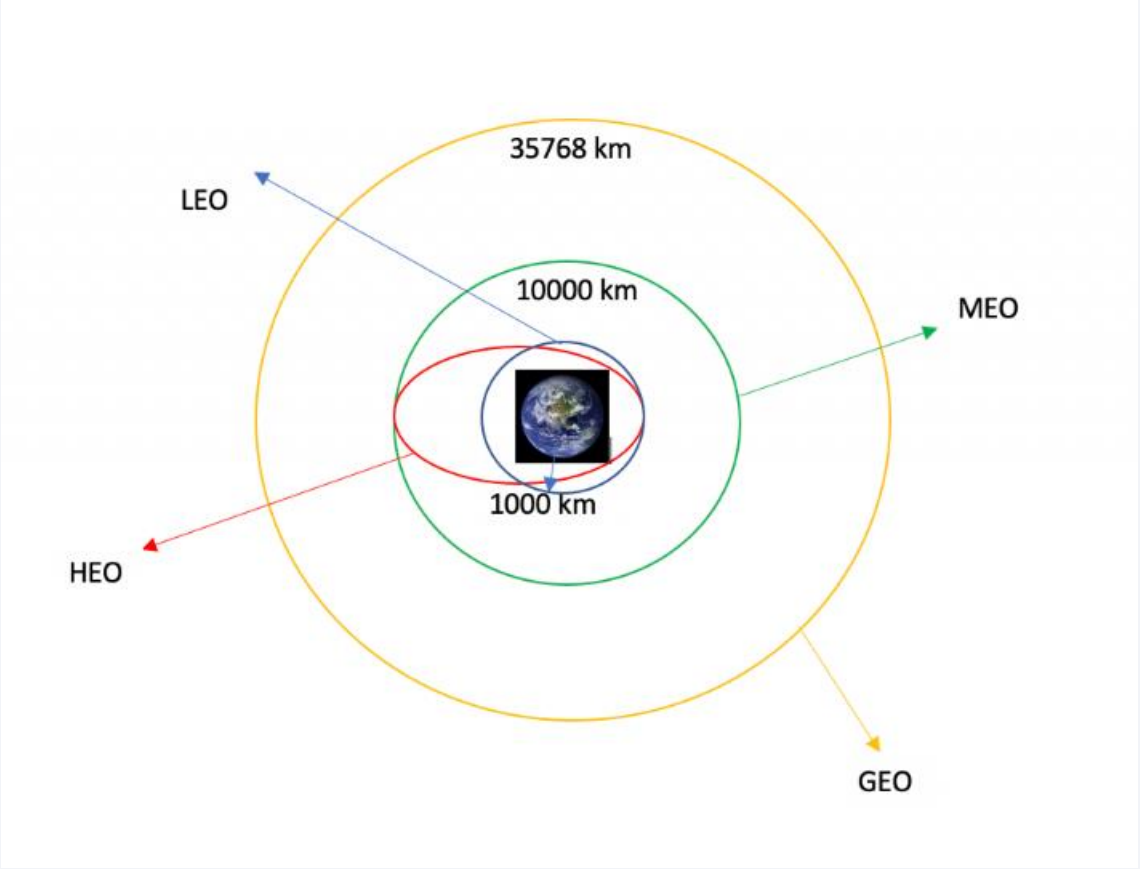
In [16]:
```
# Use json_normalize meethod to convert the json result into a dataframe
jlist = requests.get(static_json_url).json()
data = pd.json_normalize(jlist)
```

Using the dataframe `data` print the first 5 rows

In [17]:
```
# Get the head of the dataframe
data.head()
```

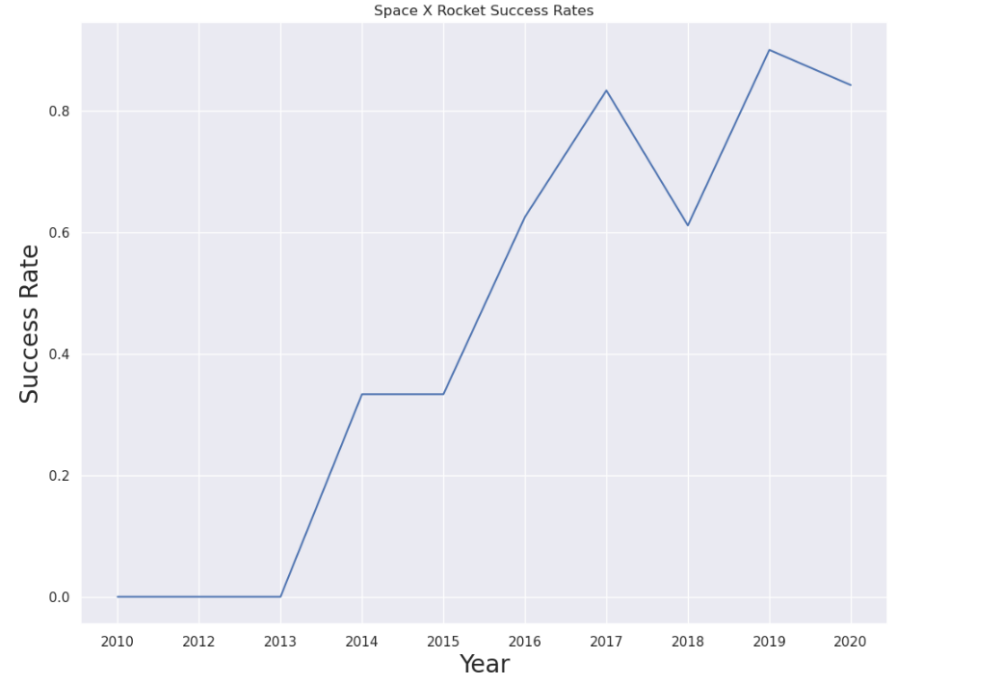
Data Wrangling

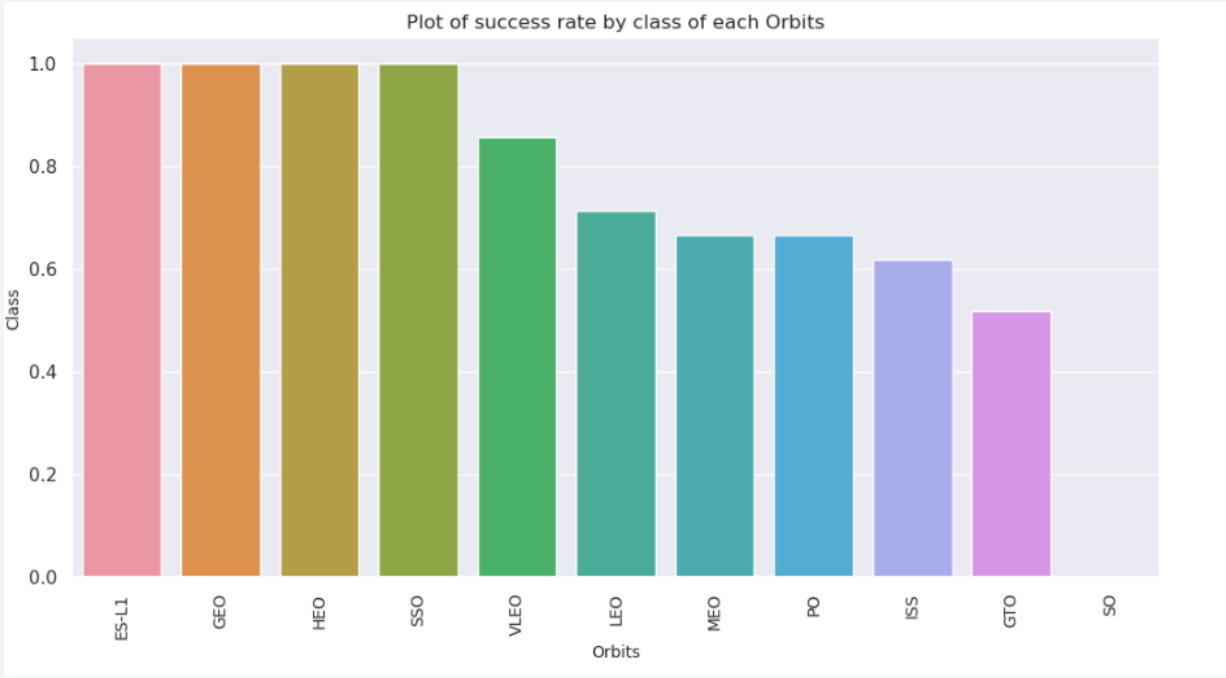
- After conducting exploratory data analysis, we identified the training labels. Our analysis involved calculating the number of launches that occurred at each site, as well as the number and frequency of launches for each orbit type. We then created a landing outcome label based on the outcome column, and finally exported the results to a csv file.
- Here's the link : https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/b3e03f7d-5794-4f7b-ab01-025b237af860/view?projectid=d6587a5a-40d8-46d6-b04a-2a562a733026&context=cpdaas



EDA with Data Visualization

- Our data exploration involved visualizing several aspects of the data. We examined the relationship between flight number and launch site, as well as between payload and launch site. We also analyzed the success rates of each orbit type and the relationship between flight number and orbit type. Additionally, we investigated the yearly trend in launch success. Link: Data Visulization - IBM Watson Studio





EDA with SQL

- In this analysis, we loaded the SpaceX dataset into a PostgreSQL database within the Jupyter notebook environment. We utilized SQL for exploratory data analysis and gained insight from the data by writing queries. Some of the queries included determining the unique names of launch sites, the total payload mass carried by boosters launched by NASA (CRS), the average payload mass carried by booster version F9 v1.1, the total number of successful and failed mission outcomes, and identifying failed landing outcomes in drone ship, along with their corresponding booster version and launch site names.
- Link: https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/792b7204-c492-4492-87fa-6b25f057ea83/view?projectid=d6587a5a-40d8-46d6-b04a-2a562a733026&context=cpdaas

Build an Interactive Map with Folium

- In this analysis, we utilized folium maps to mark all launch sites, and added map objects such as markers, circles, and lines to indicate the success or failure of launches for each site. We assigned the launch outcomes (failure or success) to class 0 and 1, respectively. By using color-labeled marker clusters, we were able to identify launch sites that had a relatively high success rate. Additionally, we calculated the distances between each launch site and its proximities, and answered questions such as whether launch sites were near railways, highways, and coastlines, and whether they kept a certain distance away from cities.
- Link: https://eu-de.dataplatform.cloud.ibm.com/analytics/notebook5a-40d8-46d6-b04a-2a562a733026&context=cpdaass/v2/e0b9ecc1-fa05-45b8-beed-0079b0c57e61/view?projectid=d6587a

Build a Dashboard with Plotly Dash

We utilized Plotly Dash to develop an interactive dashboard that included:

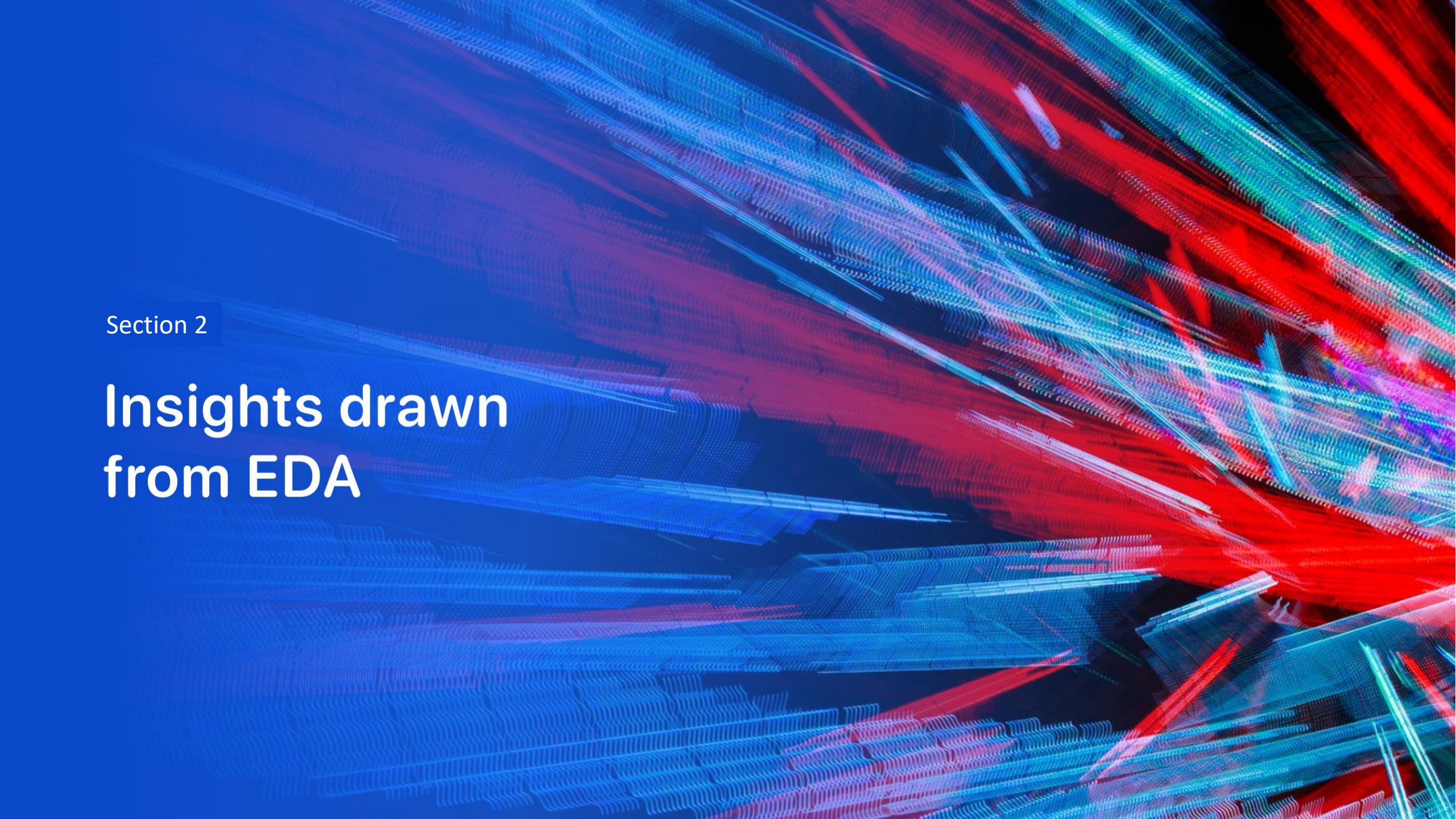
- Pie charts showcasing the overall number of launches for specific sites.
- A scatter plot that depicted the correlation between Outcome and Payload Mass (Kg) for different booster versions.

Predictive Analysis (Classification)

- In this analysis, we loaded the data using numpy and pandas, transformed it, and split it into training and testing sets. We then constructed various machine learning models and tuned different hyperparameters using GridSearchCV. The accuracy metric was utilized to evaluate the performance of each model, and we refined the models by utilizing feature engineering and algorithm tuning. Finally, we identified the best-performing classification model.
- Link: https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/77feb23a-eaa2-47b2-9f1f-afc33805dc13/view?projectid=d6587a5a-40d8-46d6-b04a-2a562a733026&context=cpdaas

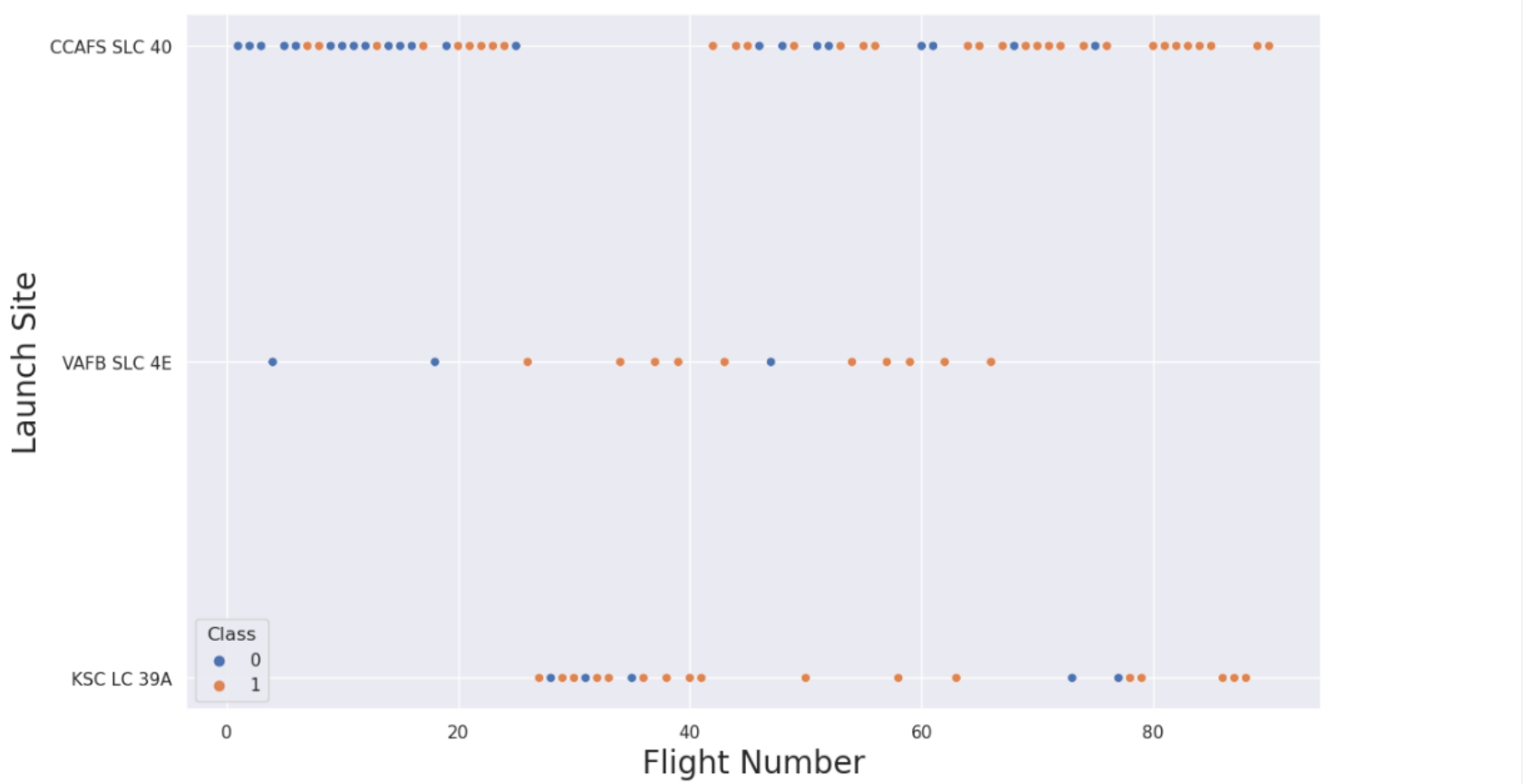
Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

Based on the plot, it was observed that as the number of flights increases at a launch site, so does the success rate.

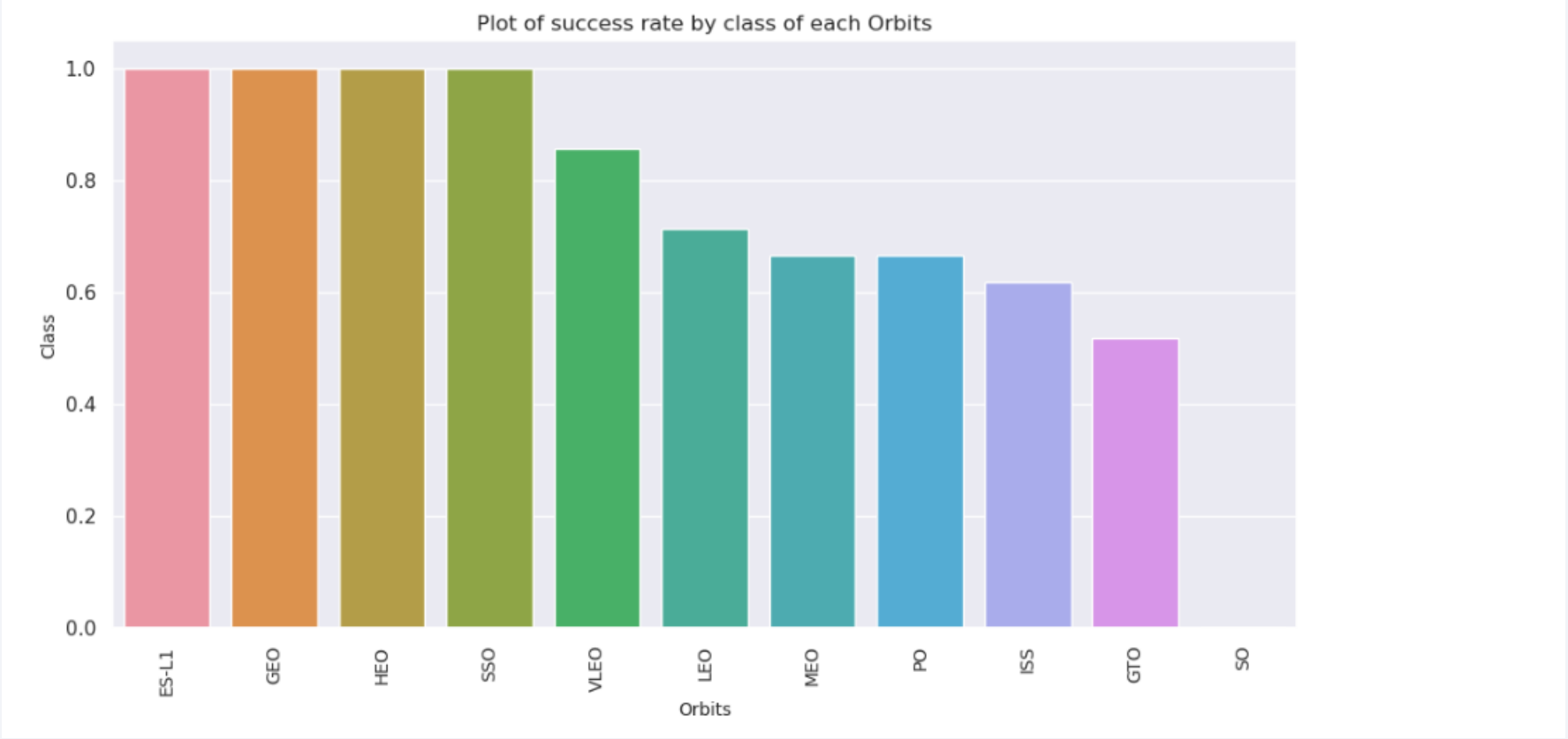


Payload vs. Launch Site

- For the launch site CCAFS SLC 40, the higher the payload mass the higher the success rate for the rocket

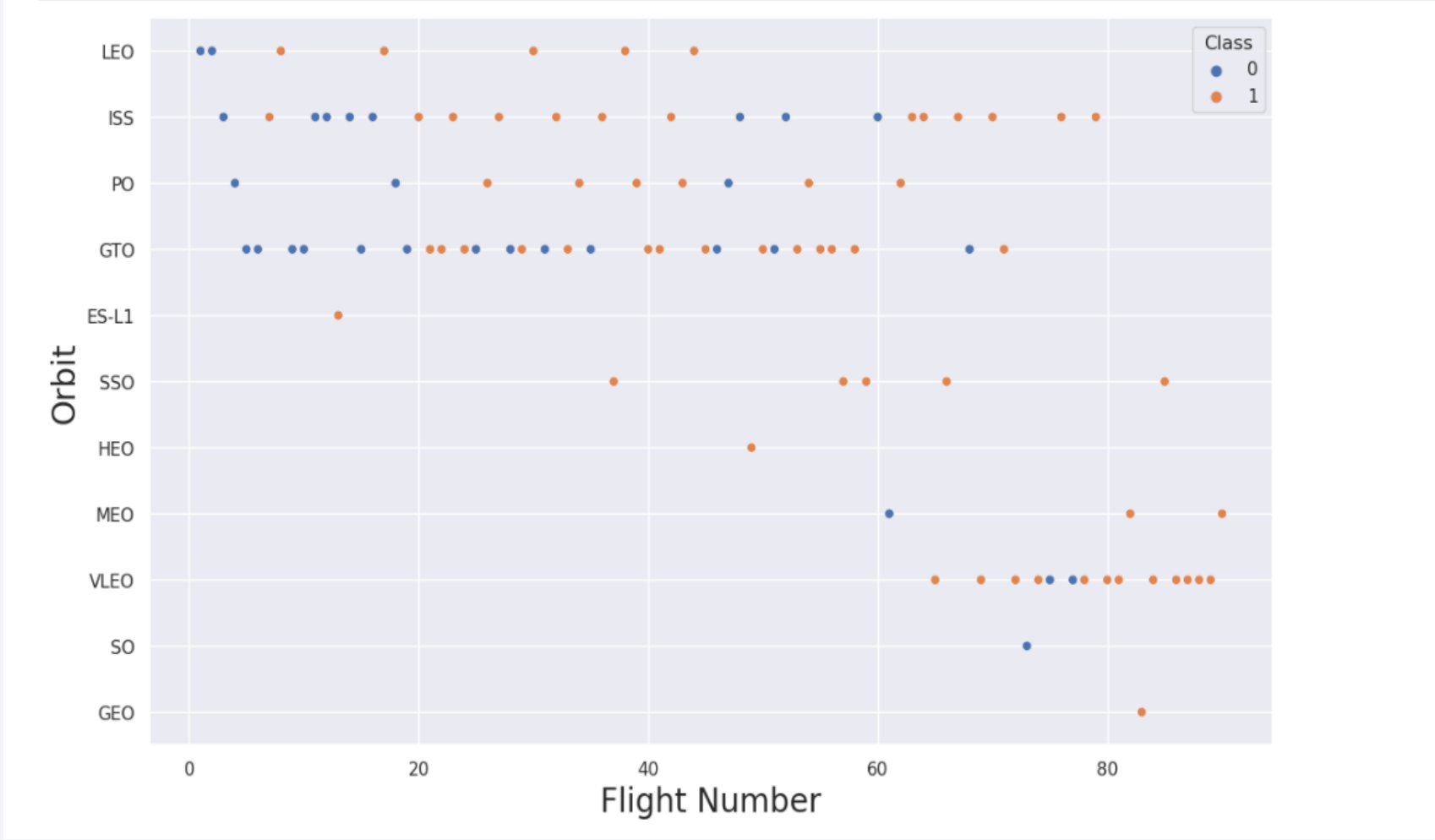
Success Rate vs. Orbit Type

- We can observe the success rate by decreasing order



Flight Number vs. Orbit Type

- The plot depicted below displays the correlation between Flight Number and Orbit type. It was noticed that for the LEO orbit, success is linked to the number of flights, while for the GTO orbit, there is no discernible association between flight number and the orbit.

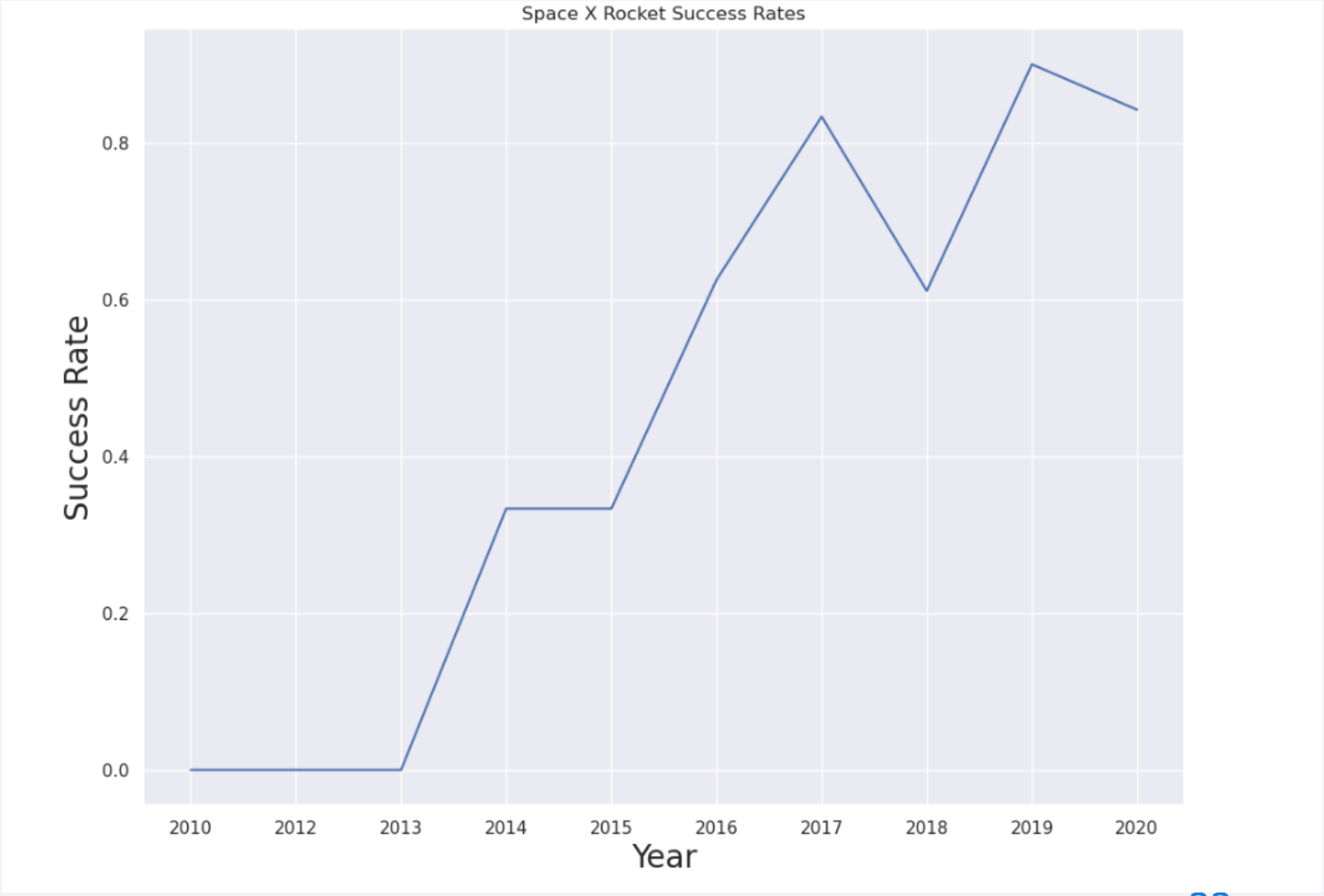


Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

Launch Success Yearly Trend

- We can observe that the success rate is increasing from 2013



All Launch Site Names

Display the names of the unique launch sites in the space mission

```
%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 used the query below 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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass

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

We calculated the total payload carried by boosters from NASA as 45596 using the above

Total Payload Mass by NASA (CRS)
22007

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 below

Display average payload mass carried by 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
3676

First Successful Ground Landing Date

- We can observe the first successful land date below

```
%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
2017-01-05

Successful Drone Ship Landing with Payload between 4000 and 6000

- To identify boosters that have successfully landed on the drone ship, we employed the WHERE clause and then utilized the AND condition to specify a successful landing with a payload mass between 4000 and 6000.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 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 B1031.2

Total Number of Successful and Failure Mission Outcomes

- To filter for cases where the MissionOutcome was either a success or a failure, we employed the wildcard character '%'.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

Successful Mission
45

Failure Mission
0

Boosters Carried Maximum Payload

- We observe Boosters with Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

We observe below the 2015 Launch Records

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

In [18]:
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

We retrieved the Landing outcomes and the COUNT of landing outcomes from the dataset and filtered for the landing outcomes that occurred between 2010-06-04 and 2010-03-20 using the WHERE clause. We then grouped the landing outcomes using the GROUP BY clause and arranged them in descending order using the ORDER BY clause.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

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

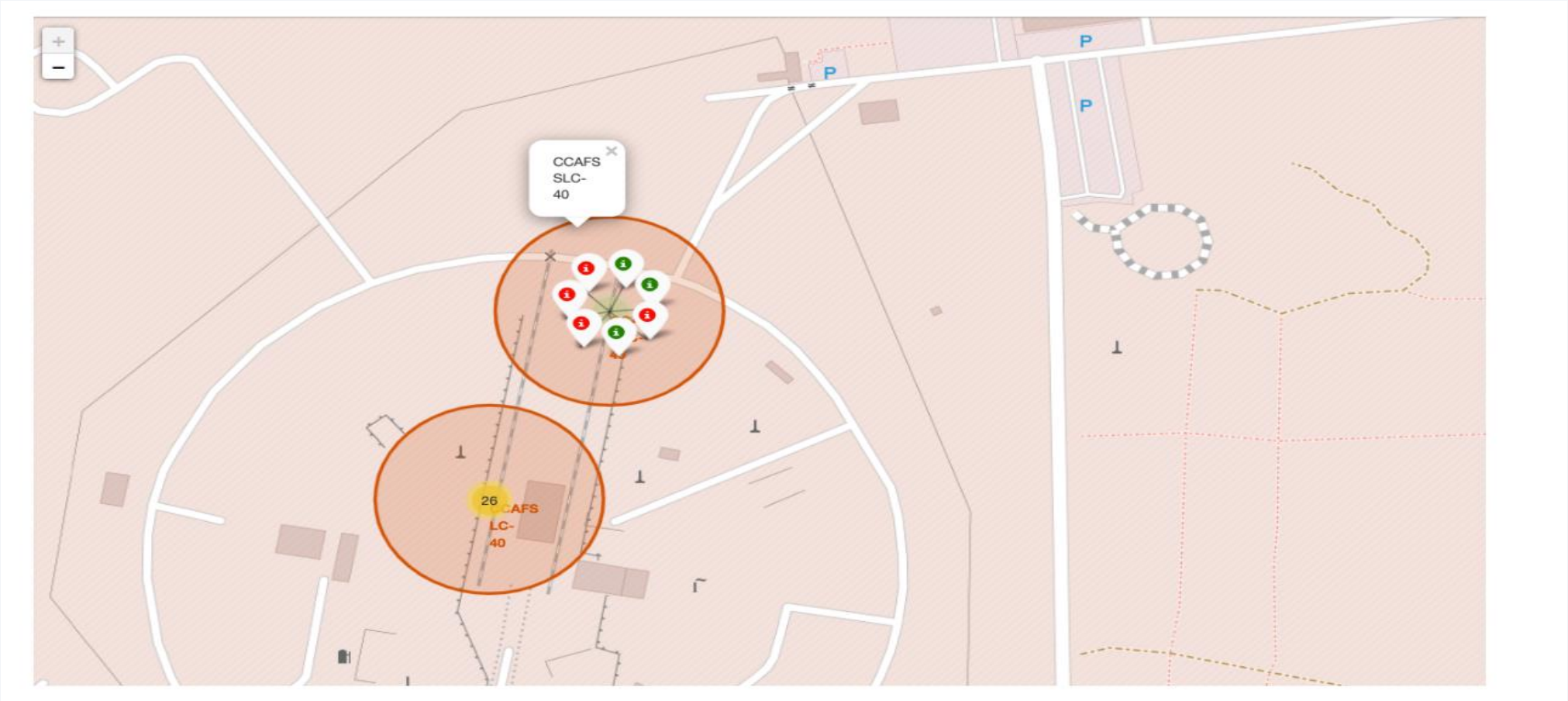
Section 3

Launch Sites Proximities Analysis

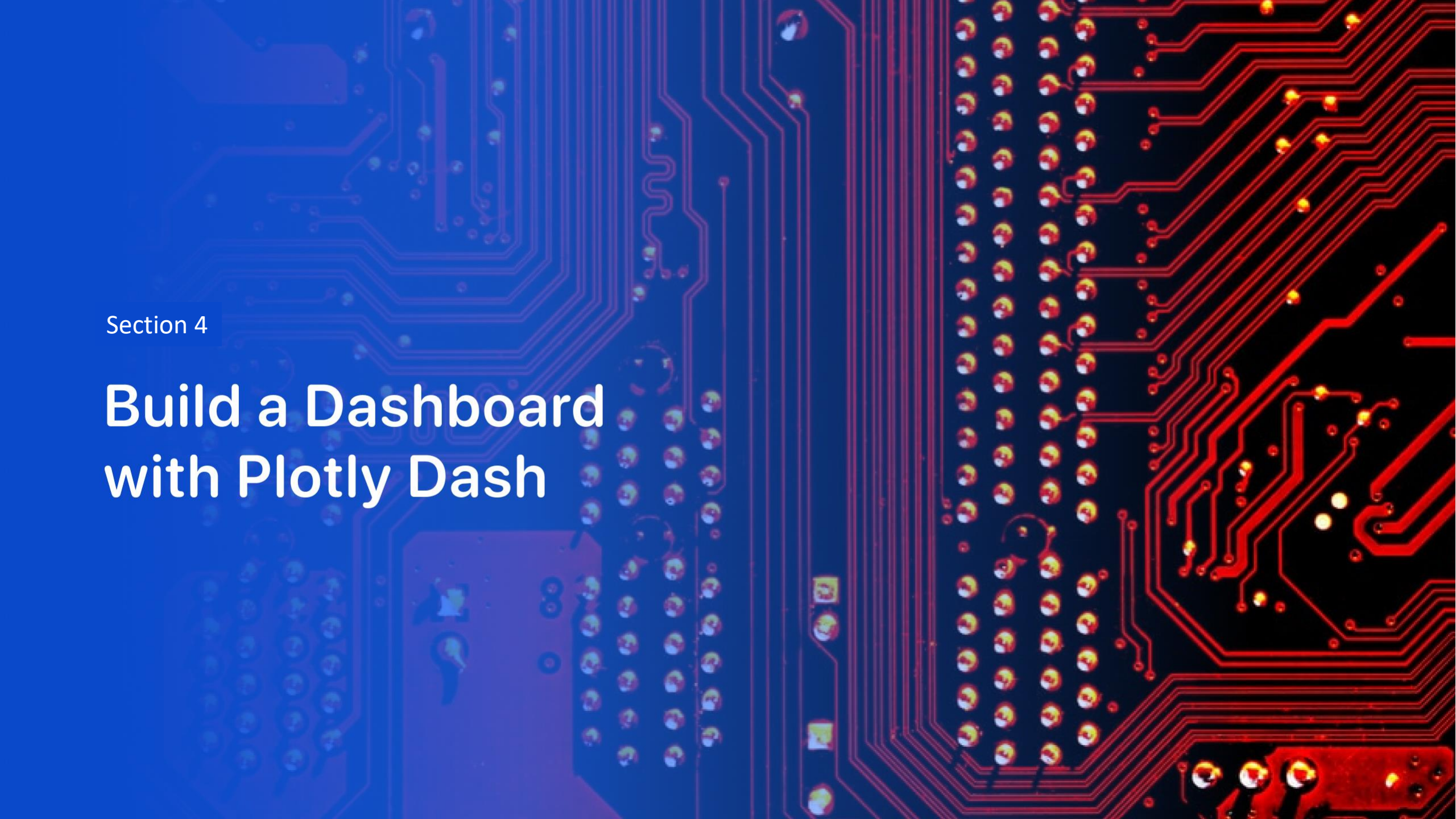
Launch Sites With Global Map Markers

It is observed that the markers are only in the US Coasts

Launch sites with color labels

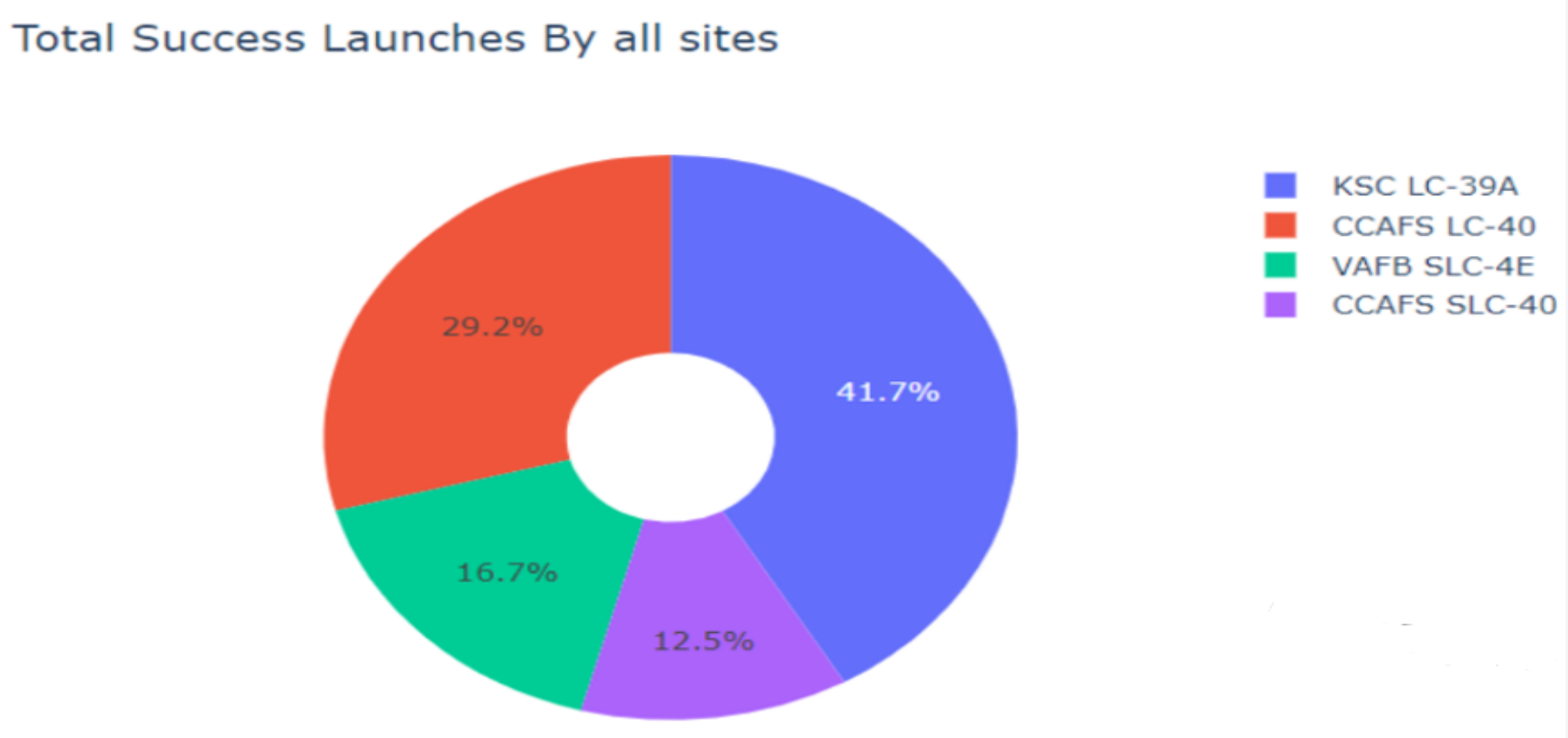


Launch Site with distance to Shore


Section 4
Build a Dashboard with Plotly Dash

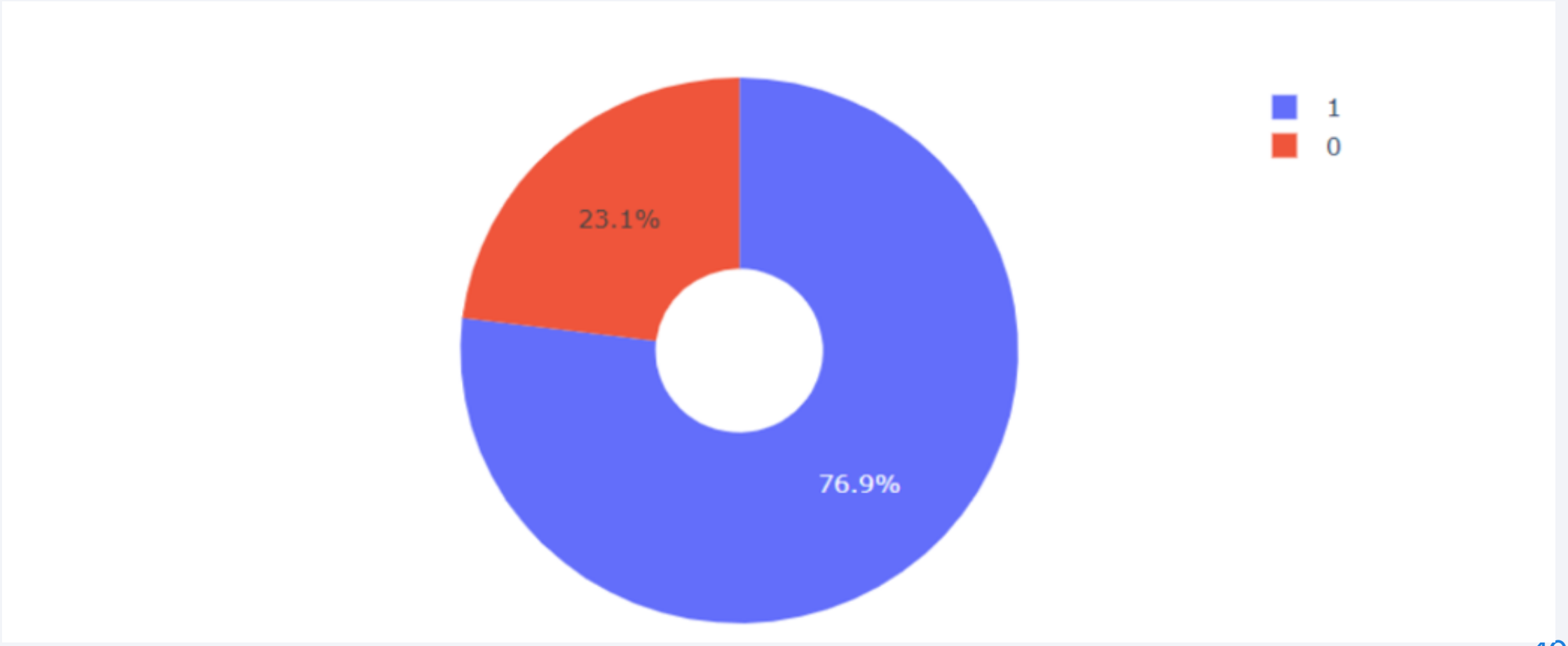
Success Percentage Pie Chart

KSC LC 38A has the most success



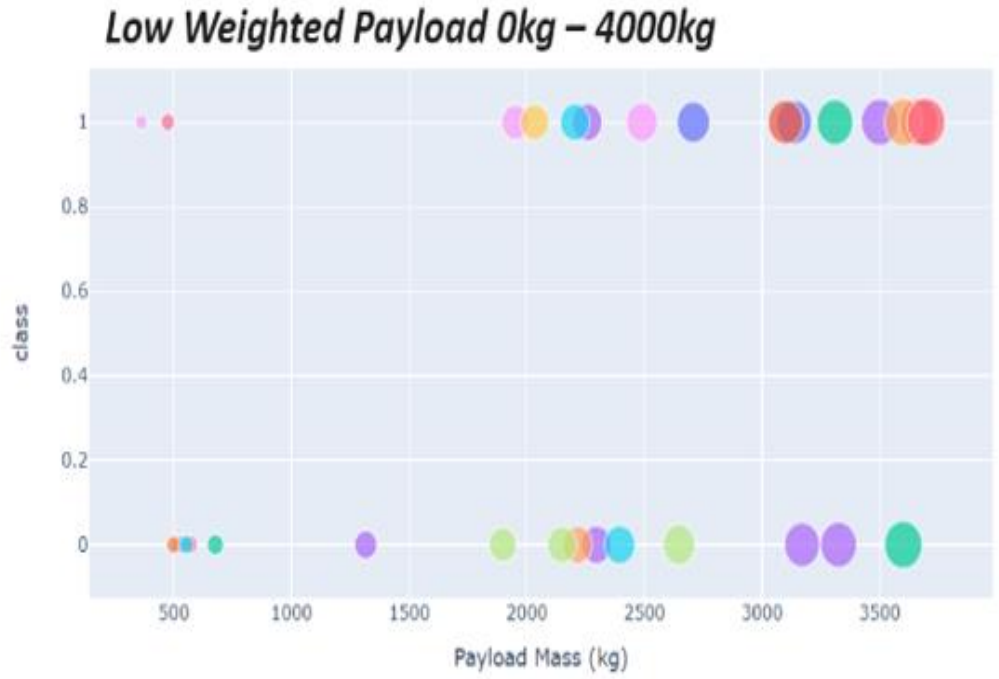
Highest Launch success ratio

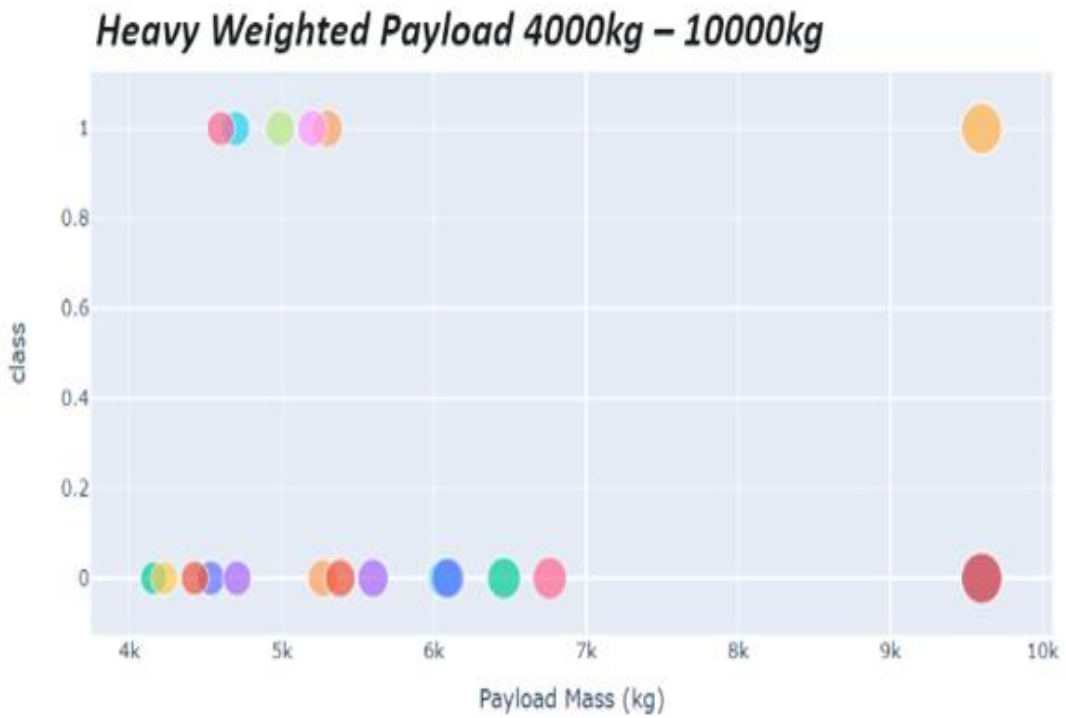
Pie chart with the most success ratio



Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

It is observed that low weighted payload has more success than heavy weighted payload





Section 5

Predictive Analysis (Classification)

Classification Accuracy

Below is the algorithm that has the best accuracy

Find the method performs best:

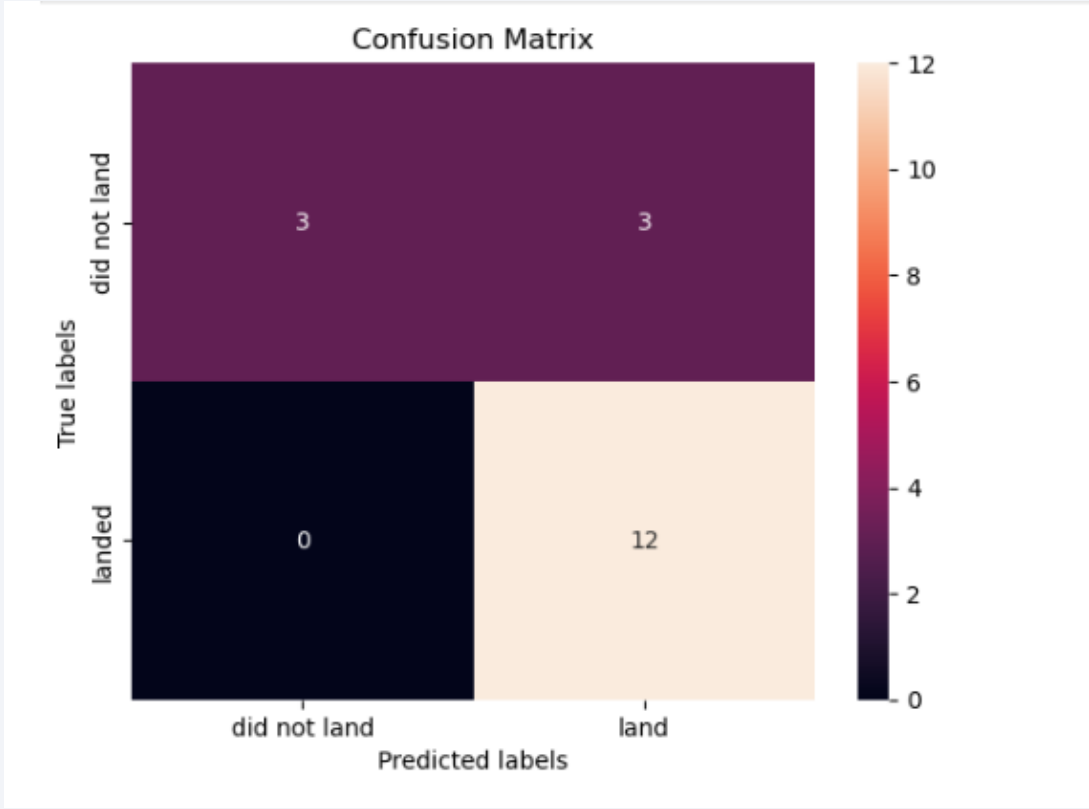
```
algorithms = {'KNN':knn_cv.best_score_,'Decision Tree':tree_cv.best_score_,'Logistic Regression':logreg_cv.best_score_,'SVM':svm_cv.best_score_}
best_algorithm = max(algorithms, key= lambda x: algorithms[x])

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 0.8767857142857143
```

Confusion Matrix

- The decision tree classifier's confusion matrix indicates that it can differentiate between the various classes. However, it has a significant issue with false positives, which refers to the classifier marking an unsuccessful landing as a successful one



Conclusions

- There is a positive correlation between the number of flights at a launch site and the success rate.
- The launch success rate has been increasing since 2013 up until 2020.
- Orbits such as ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.
- Among all launch sites, KSC LC-39A had the highest number of successful launches.
- The Decision tree classifier was found to be the most effective machine learning algorithm for this particular task.

Thank you!