Applied Data Science Capstone

BRIAN EDUARDO TAY SACCACO

Outline

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

Executive Summary

Executive Summary -

Data collection and preparation

- Data was gathered from SpaceX API and Space Wikipedia Page.
- A 'class' column was created to classify successful landings.
- Data exploration was conducted using SQL, visualization tools, folium maps, and dashboards.
- Data was standardized to ensure consistency.
- GridSearchCV was employed to identify the best parameters for machine learning models.

Machine Learning Models

- Four machine learning models were developed:Logistic Regression, Support vector machine, decision tree classifier and K nearest Neighbors
- All models demonstrated similar performance, achieving an accuracy of approximately 83.33%.
- A common issue was the overprediction of successful landings across all models.

Introduction

This report presents a comprehensive analysis of “Space X” launch data with the objective of classifying successful landings. As part of this capstone project, I’m going to assume the role of a data scientist working for a new rocket company called “Space Y”. The analysis aims to leverage advanced data science techniques including regression, machine learning, and classification, to provide actionable insights and enhance the success rate of future rocket landings for “Space Y”.

Methodology

Overview of Data Collection, Wrangling, Visualization, Dashboard and model methods

- **Data collection methodology:** Combined data from SpaceX API and Wikipedia
- **Perform data wrangling:** Classifying true landings as successful and unsuccessful
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Foliom and Plotly Dash**
- **Perform predictive analysis using classification models:** Tuned models using GridSearchCV

Data Collection Overview

IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone at main · briedts/IBM-Data-Science-Capstone-Project

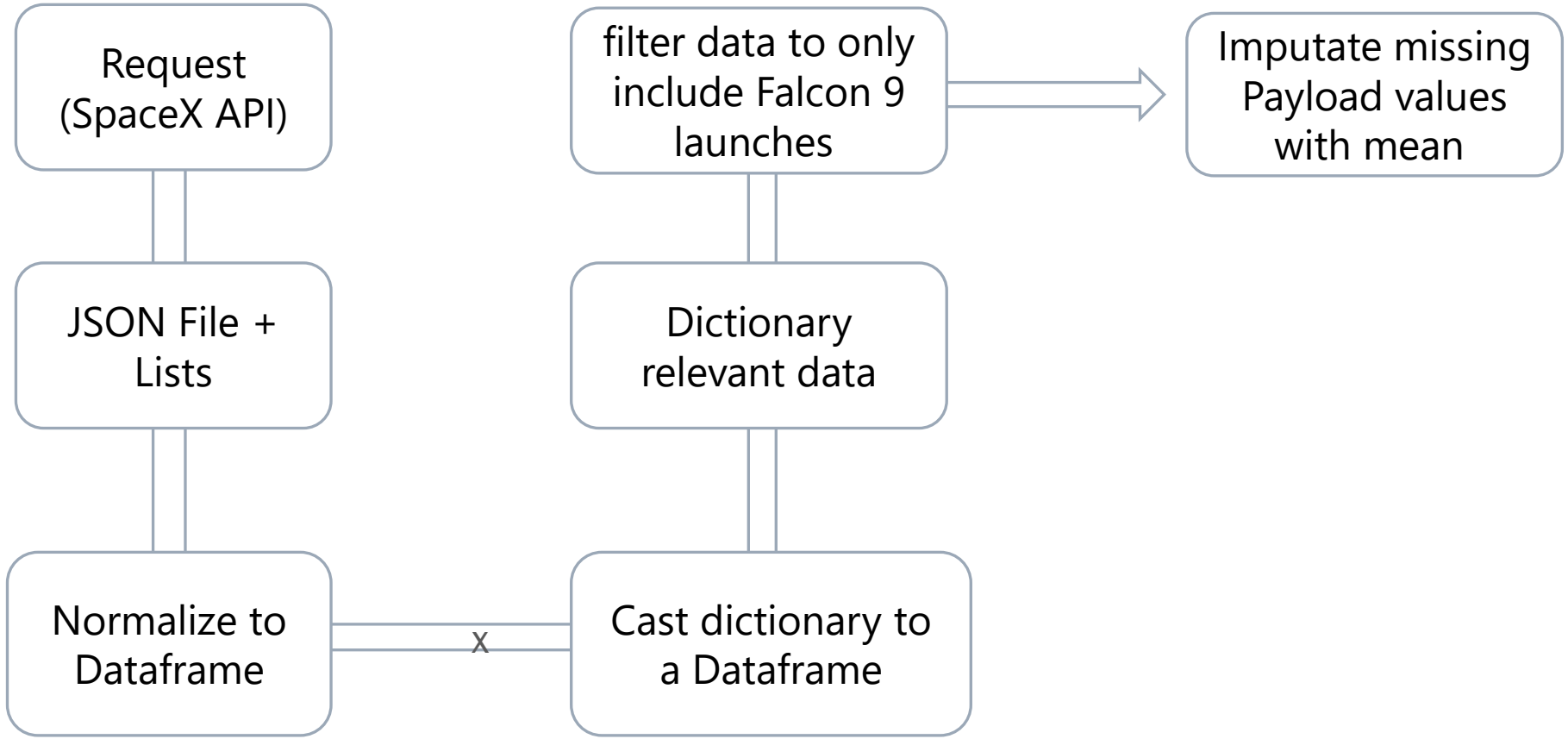
Data was gathered from SpaceX API and Space Wikipedia Page. The next slides will show the flowchart of data collection

- **Space X Data columns:**

FlightNumber, Date, BoosterVersion , PayloadMass, Orbit, Outcome, Flights, GrindFins, Reused Legs, LandingPad, Block ReusedCount, Serial, Longitude, Latitude

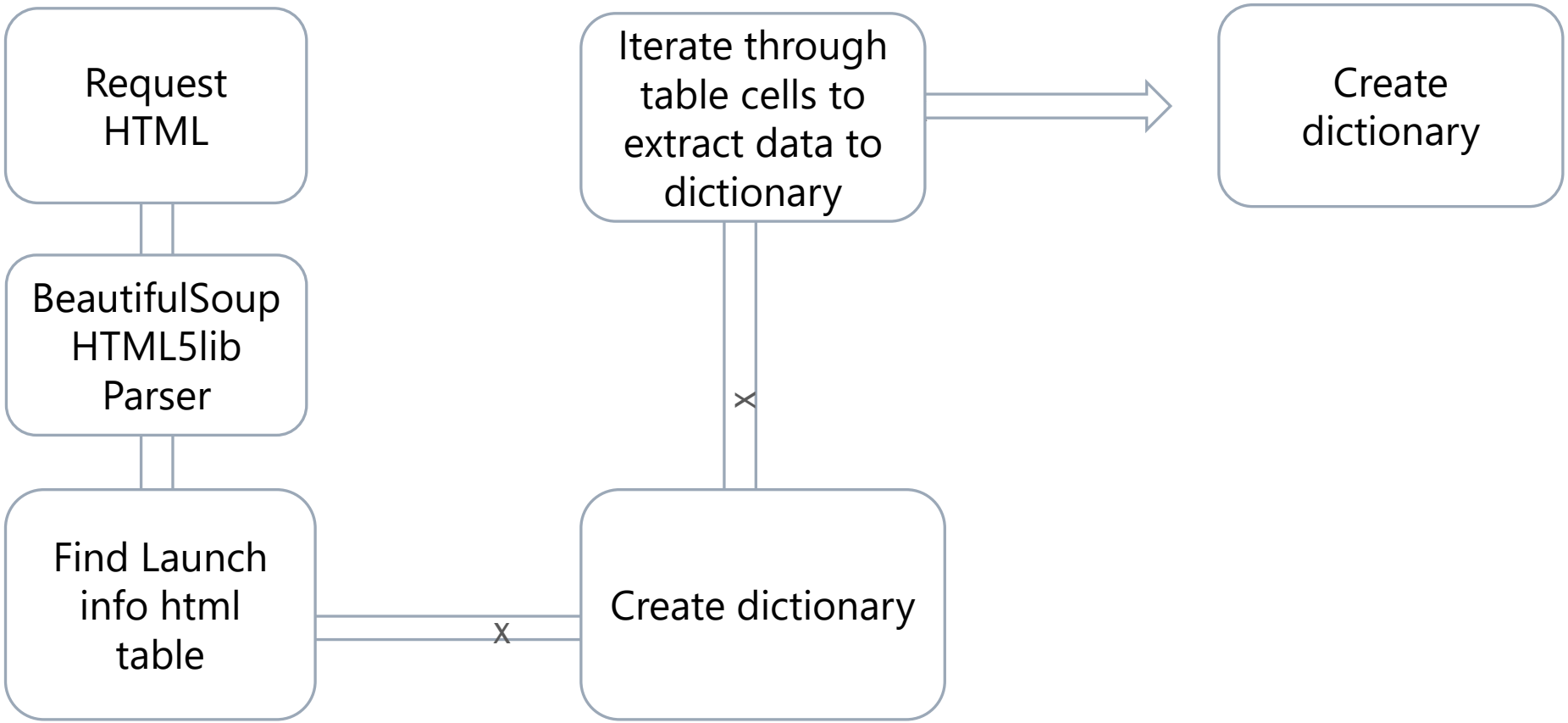
- **Wikipedia Webscrape Data Columns:**

Flight N°, Launch Site, Payload, Payload Mass , Orbit, Costumer, Launch Outcome, Version Booster, Booster Landing, Date, Time



Data collection - SpaceXAPI

Github url: [IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/Data Collection API.ipynb at main · briedts/IBM-Data-Science-Capstone-Project]



Data collection – Web Scrapping

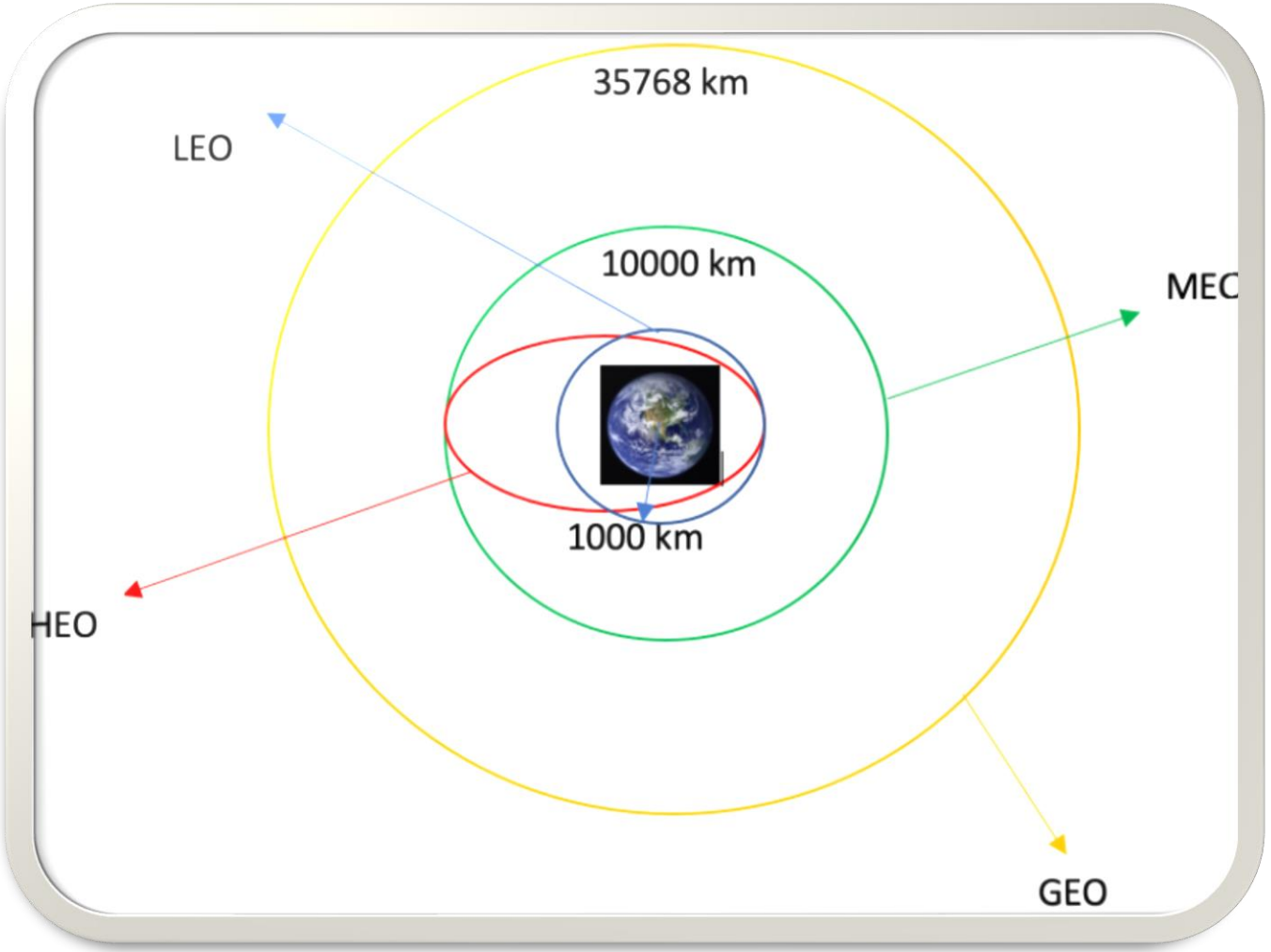
Github url: [IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/Data Collection with Web Scraping.ipynb at main · briedts/IBM-Data-Science-Capstone-Project]

Data Wrangling

Exploratory data analysis was performed to determine the training labels. The number of launches at each site, as well as the number and occurrence of each orbit, were calculated.

The landing outcome label was created from the outcome column, and the results were exported to a CSV file.

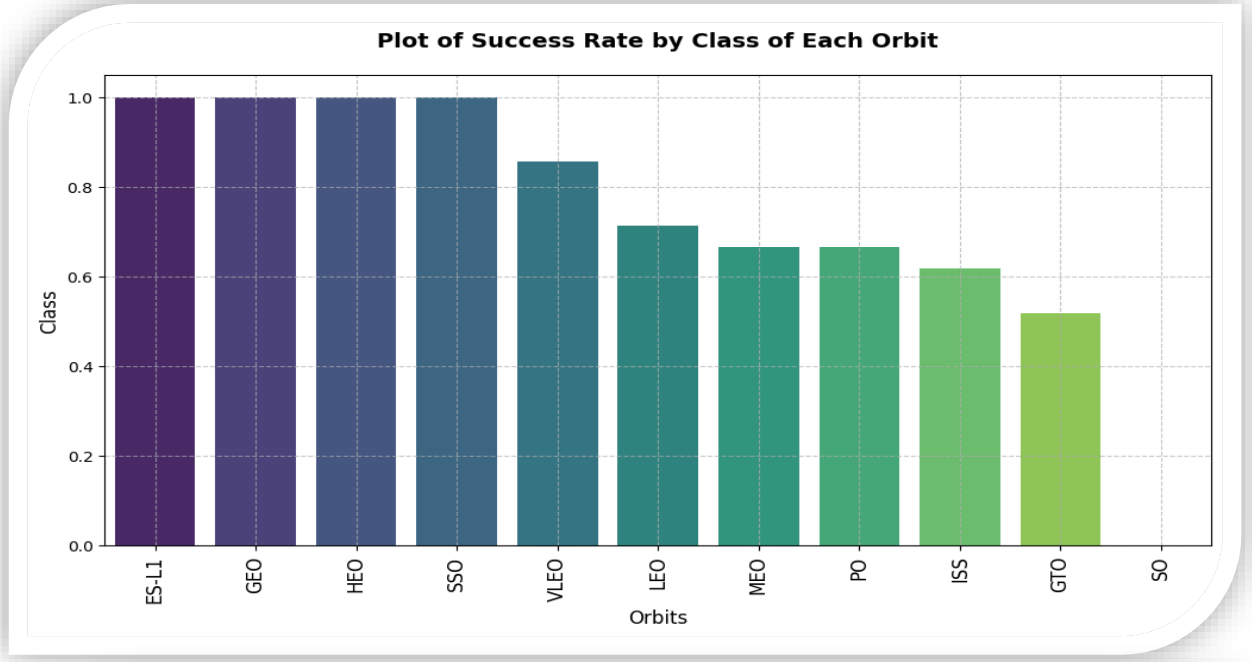
[IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/Data Wrangling.ipynb at main · briedts/IBM-Data-Science-Capstone-Project]

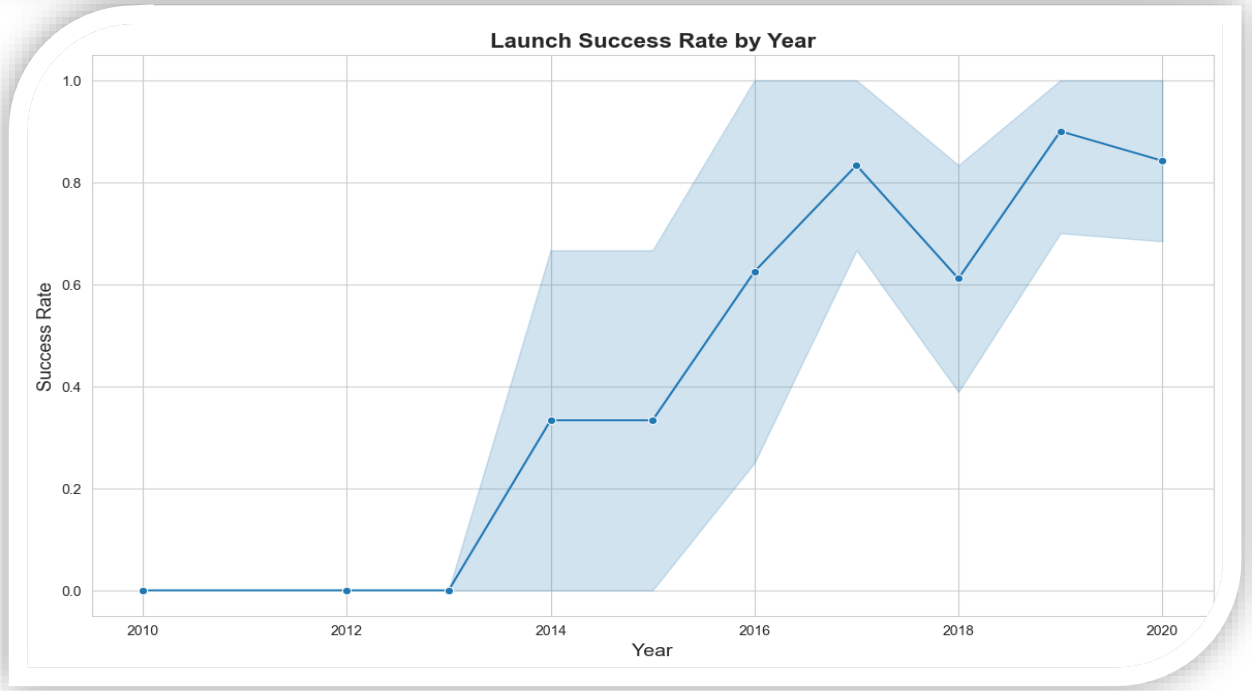


EDA With Data Visualization

The data was explored by visualizing the relationships between flight number and launch site, payload and launch site, success rates of each orbit type, flight number and orbit type, and the yearly trend of launch success.

[IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/EDA with Data Visualization.ipynb at main · briedts/IBM-Data-Science-Capstone-Project]





03/03/2025

EDA with SQL

The SpaceX datased was loaded into a PostgreSQL database without leaving jupyter notebook to get insights

IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/EDA with SQL.ipynb at main · briedts/IBM-Data-Science-Capstone-Project

EDA with SQL

The SpaceX datased was loaded into a PostgreSQL database without leaving jupyter noteboot to get insights

IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/EDA with SQL.ipynb at main · briedts/IBM-Data-Science-Capstone-Project

Loaded data set into PostgreSQL database

Queried using SQL Python integration

Queries were made to get a better understanding of the dataset

Queries information about launch site names mision outcomes, various costumer, booster versions and landing outcomes

Build an interactive map with Folium

[IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/Interactive Visual Analytics with Folium.ipynb at main · briedts/IBM-Data-Science-Capstone-Project]

Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.

This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

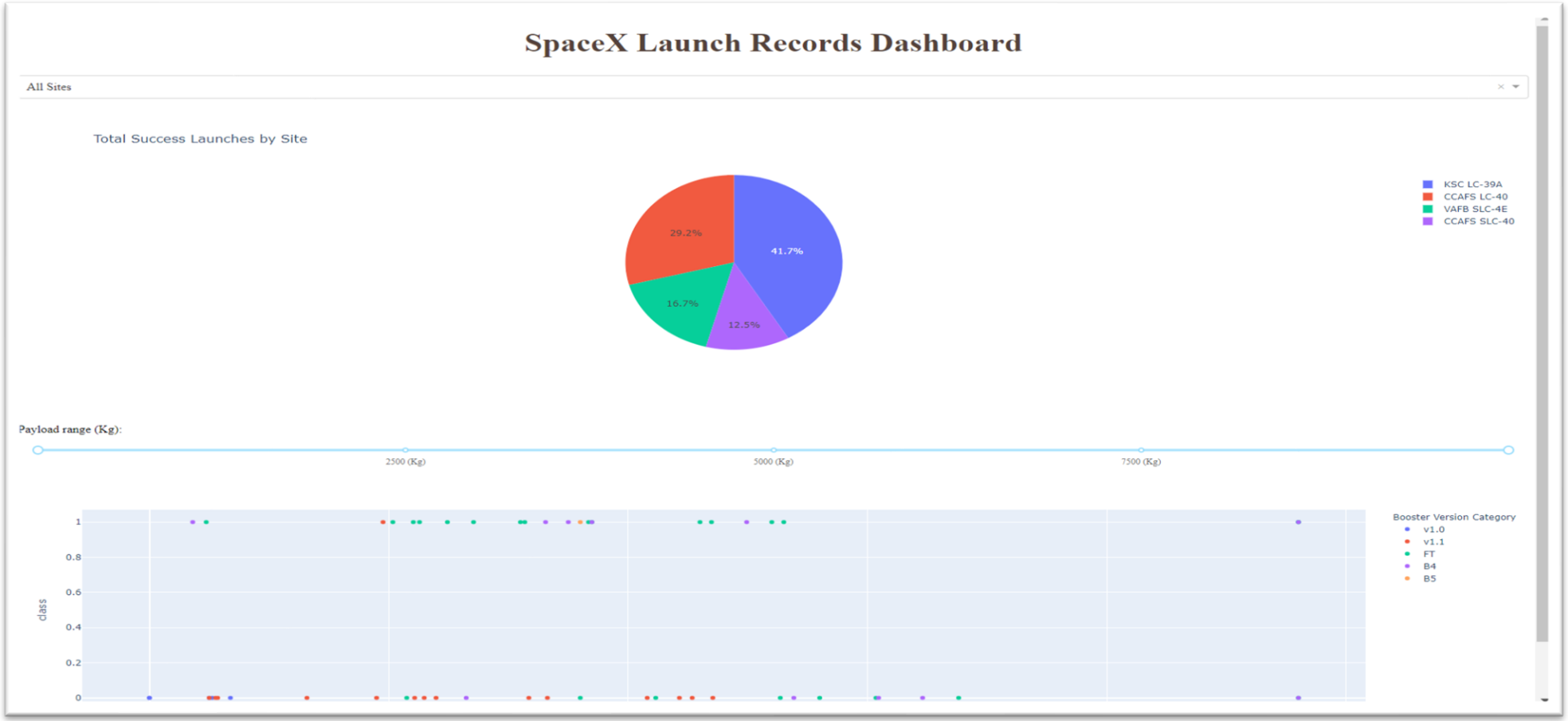
Predictive análisis (Classification)

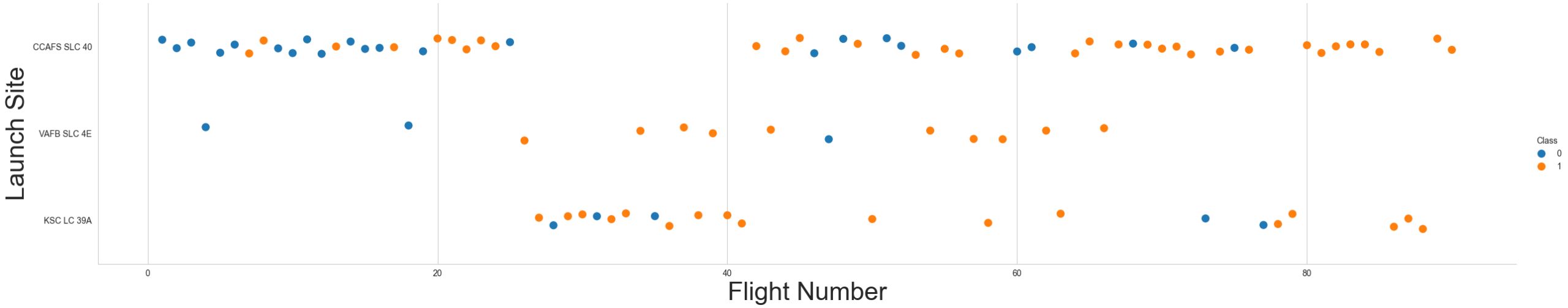
[IBM-Data-Science-Capstone-Project/IBM-Data-Science-Capstone/Machine Learning Prediction.ipynb at main · briedts/IBM-Data-Science-Capstone-Project]

.

The data was loaded using numpy and pandas, transformed, and split into training and testing sets. Various machine learning models were built, and different hyperparameters were tuned using GridSearchCV. Accuracy was used as the metric for the models, with improvements achieved through feature engineering and algorithm tuning. The best performing classification model was identified based on these efforts.

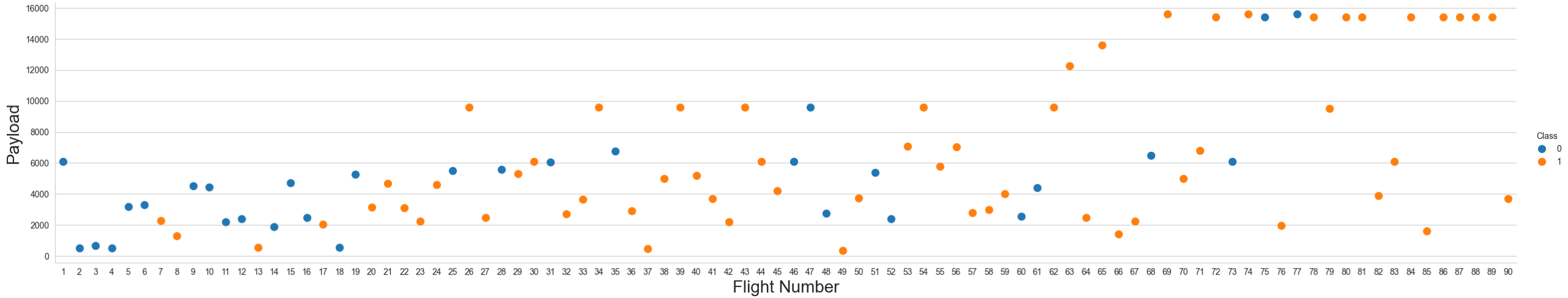
Results


SpaceX Launch Records Dashboard
All Sites
Total Success Launches by Site
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
2500 (Kg)
5000 (Kg)
7500 (Kg)
class
1
0.8
0.6
0.4
0.2
0
Booster Version Category
v1.0
v1.1
FT
B4
B5



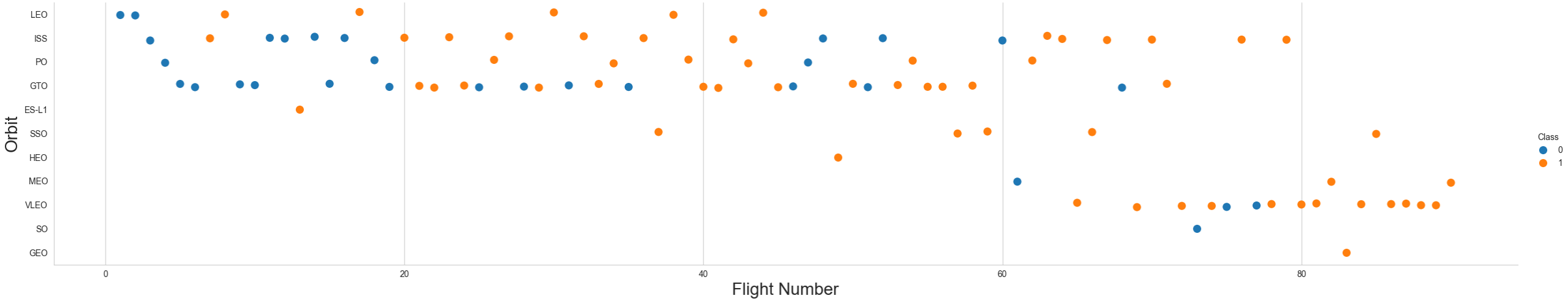
Flight Number Vs Launch Site

The plot reveals that an increased number of flights at a launch site correlates with a higher success rate.



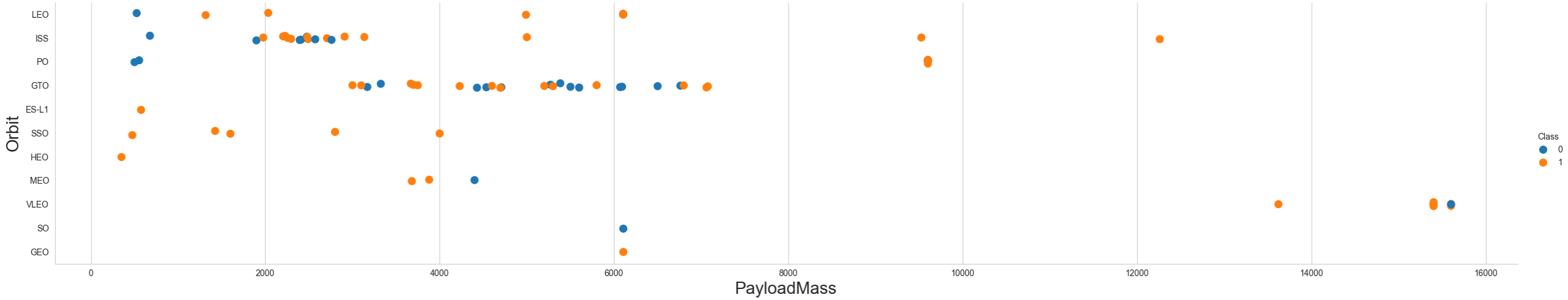
Payload vs Launch site

The payload mass predominantly ranges between 0 and 6000 kg. Additionally, different launch sites appear to utilize varying payload masses



Flight Number vs. Orbit Type

The plot below illustrates the relationship between Flight Number and Orbit type. In the LEO orbit, a higher number of flights corresponds to greater success. Conversely, in the GTO orbit, no correlation exists between the flight number and the orbit.



Payload vs. Orbit Type

With heavy payloads, successful landings are more frequent in the PO, LEO, and ISS orbits.

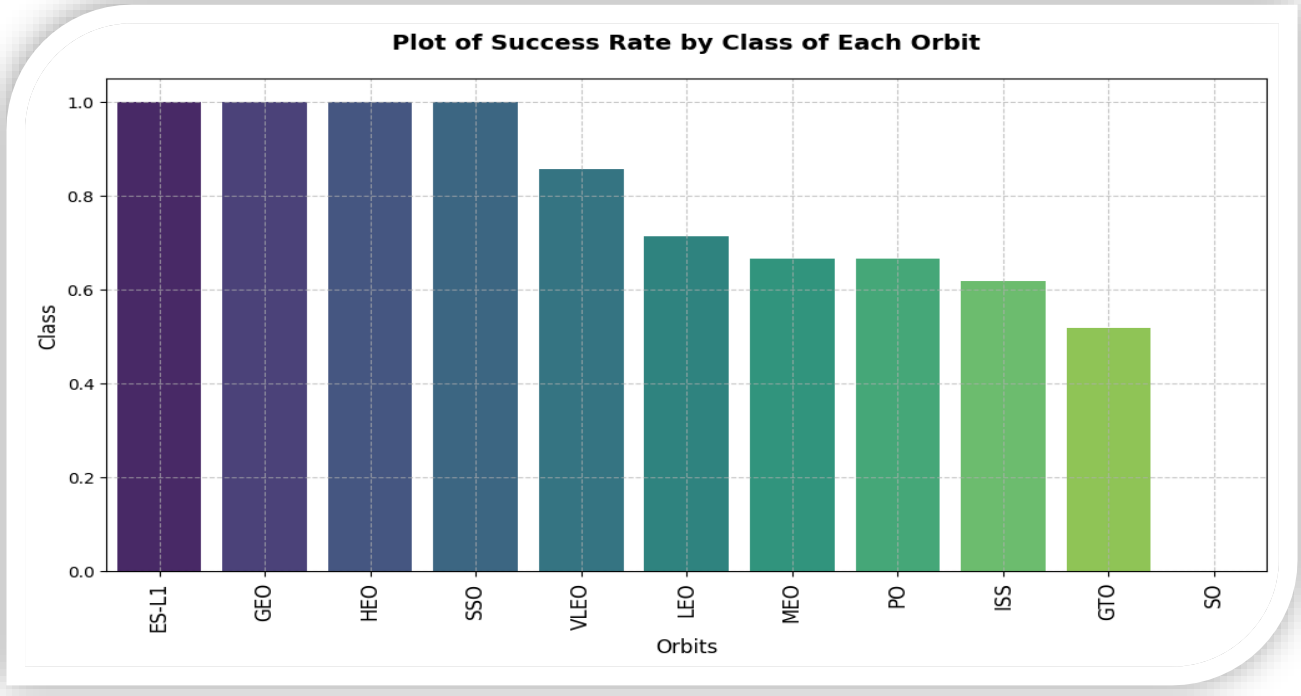
Succes rate vs Orbit Type

ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis) SSO (5) has 100% success rate

VLEO (14) has decent success rate and attempts

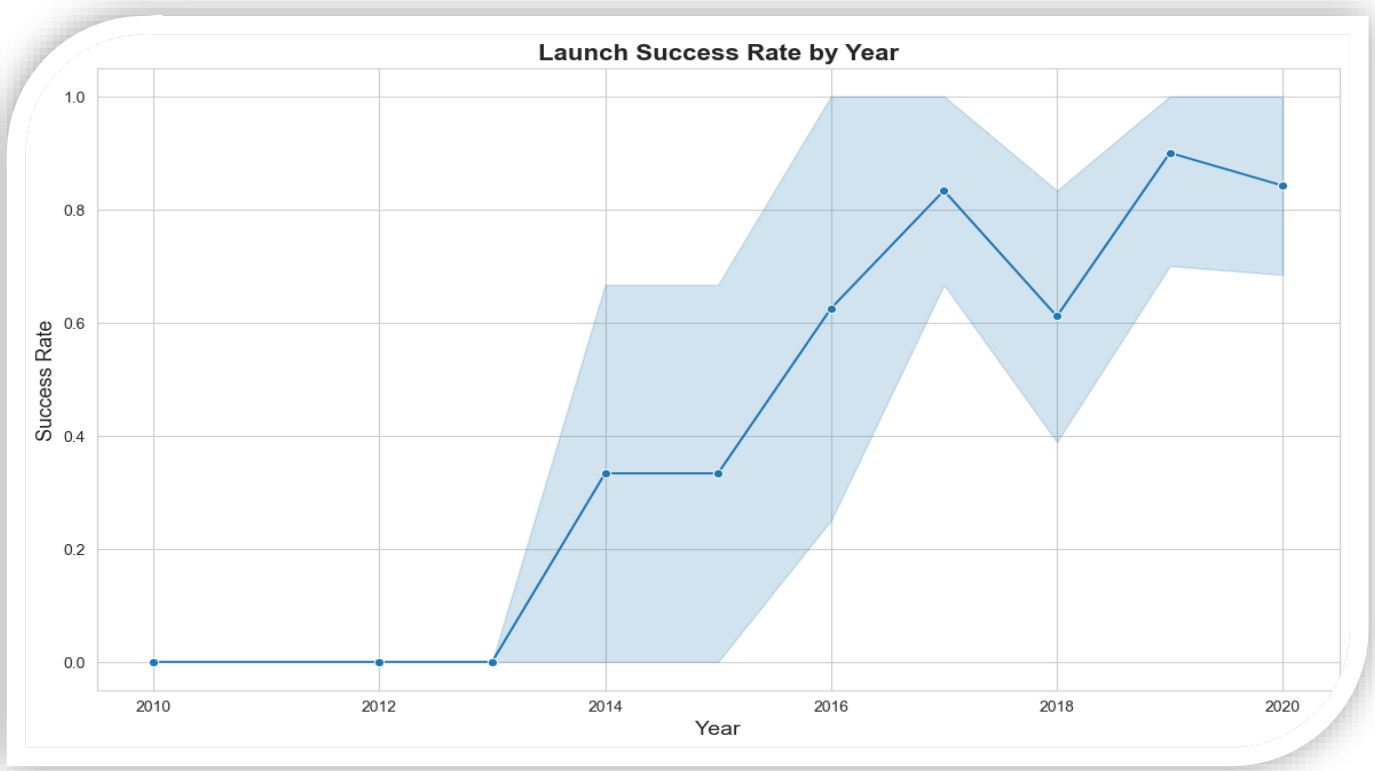
SO (1) has 0% success rate

GTO (27) has the around 50% success rate but largest sample



Launch Success Yearly Trend

The plot indicates a continuous increase in success rates from 2013 to 2020.



All Launch Site Names

The keyword DISTINCT was used to display only unique launch sites from the SpaceX data.

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

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

Launch Site Names Begin with 'CCA'

this query above to display 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	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 was carried by boosters from NASA as 45596 using the query below

Display the total payload mass carried by boosters launched by NASA (CRS)

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Total Payload Mass

total payload was carried by boosters from NASA as 45596 using the query below

Display the total payload mass carried by boosters launched by NASA (CRS)

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

the average payload mass was carried by booster version F9 v1.1 as 2928.4

Display average payload mass carried by booster version F9 v1.1

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

The first successful landing on a ground pad occurred on December 22, 2015.

In [14]:

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The WHERE clause was utilized to filter for boosters that successfully landed on a drone ship. Additionally, the AND condition was applied to identify successful landings with a payload mass greater than 4000 but less than 6000.

In [15]:
```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
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

A wildcard character, such as '%', was utilized to filter for **WHERE** Mission Outcome was either a success or a failure.

List the total number of successful and failure mission outcomes

In [16]:
```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

Out[16]:

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

	failureoutcome
0	1

Boosters Carried Maximum Payload

The booster that carried the maximum payload was identified using a subquery in the WHERE clause combined with the MAX() function.

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

A combination of the WHERE clause, LIKE, AND, and BETWEEN conditions was utilized to filter for failed landing outcomes on drone ships, along with their booster versions and launch site names, for the year 2015.

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

Landing outcomes and their counts were selected from the data using the **WHERE** clause to filter for outcomes between 2010-06-04 and 2010-03-20. The **GROUP BY** clause was applied to group the landing outcomes, and the **ORDER BY** clause was used to sort the grouped outcomes in descending order.

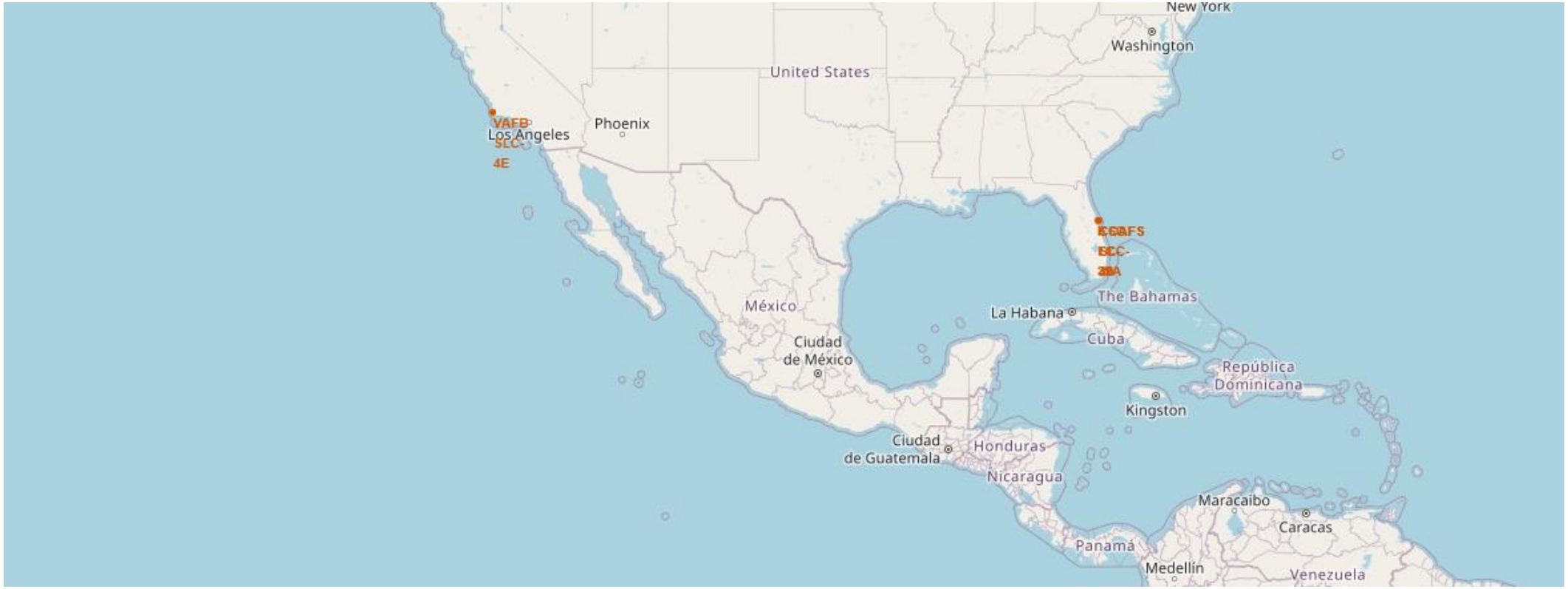
Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

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)
```

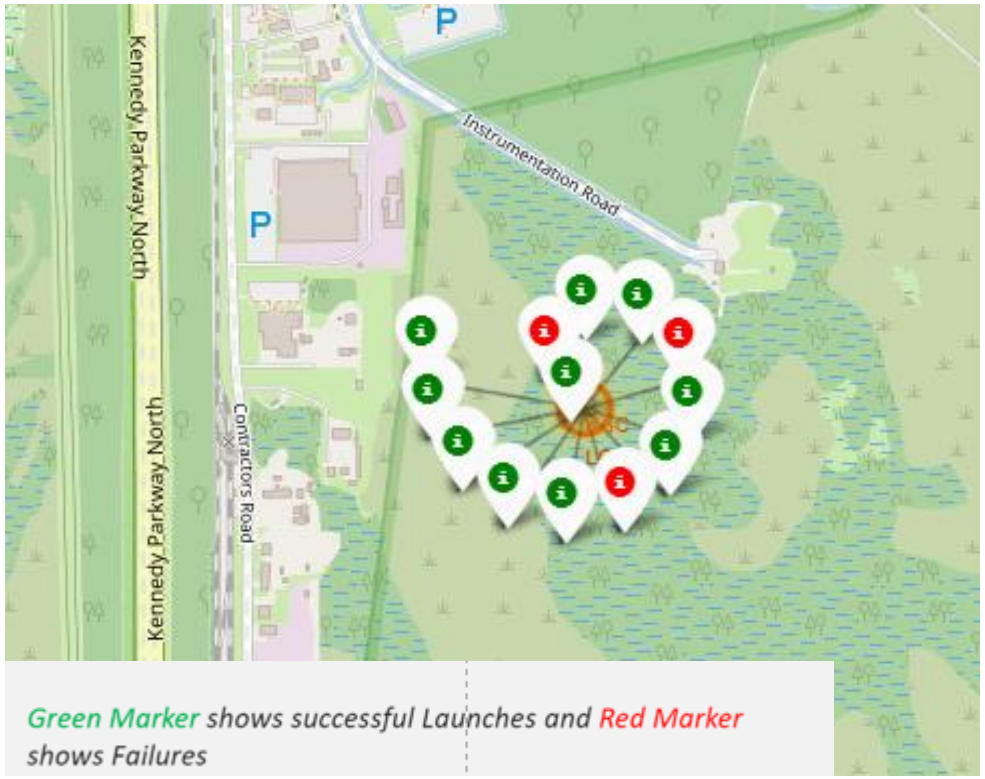
Out[19]:

	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

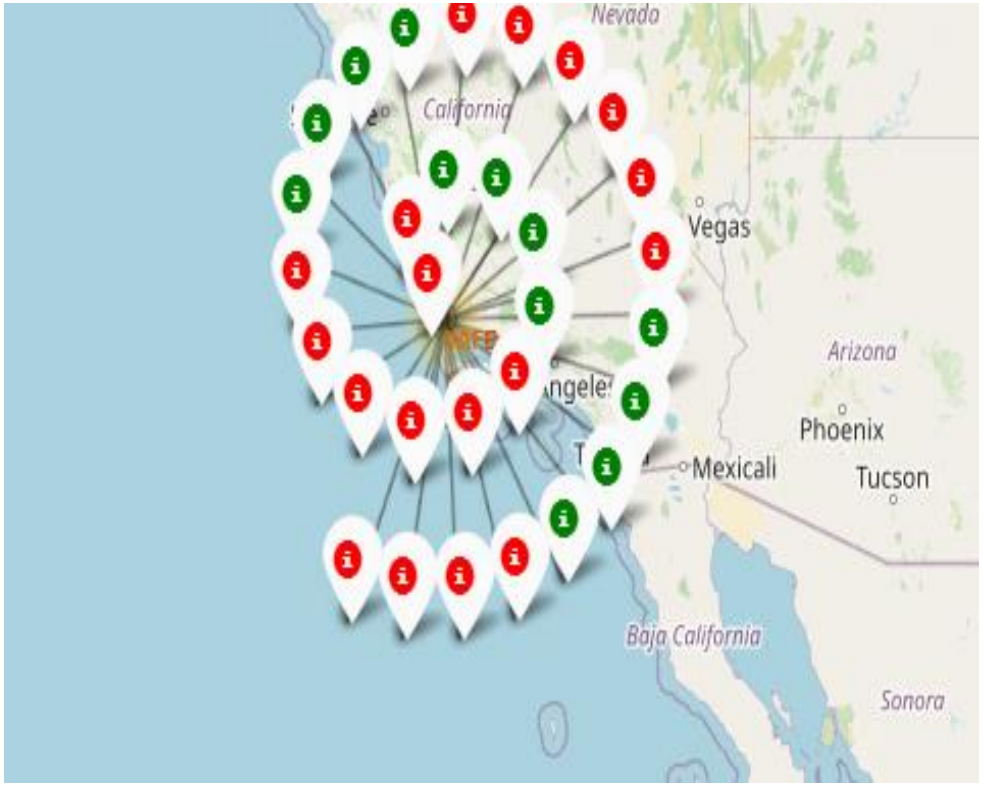
All launch sites global map markers



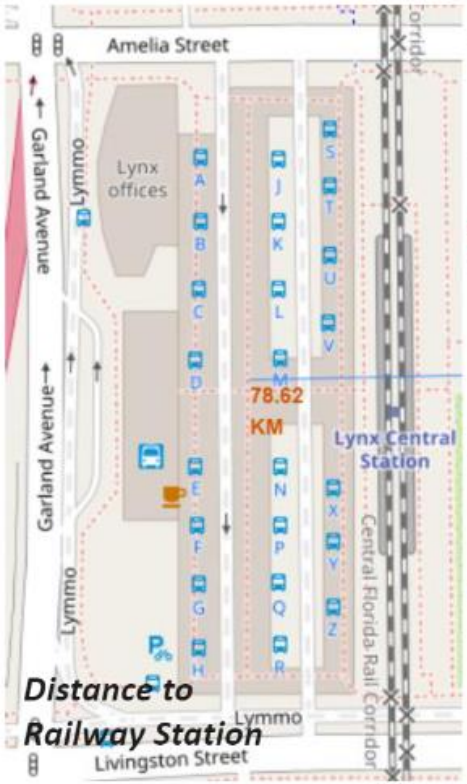
Markers showing launch sites with color labels

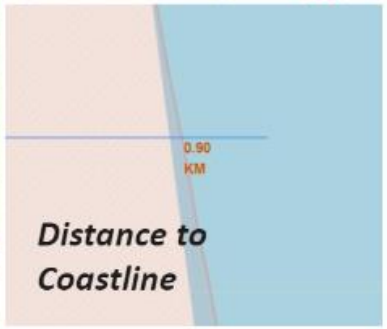


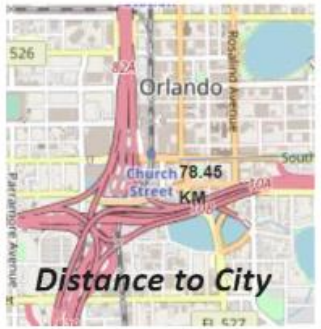
Green Marker shows successful Launches and Red Marker shows Failures



Launch Site distance to landmarks

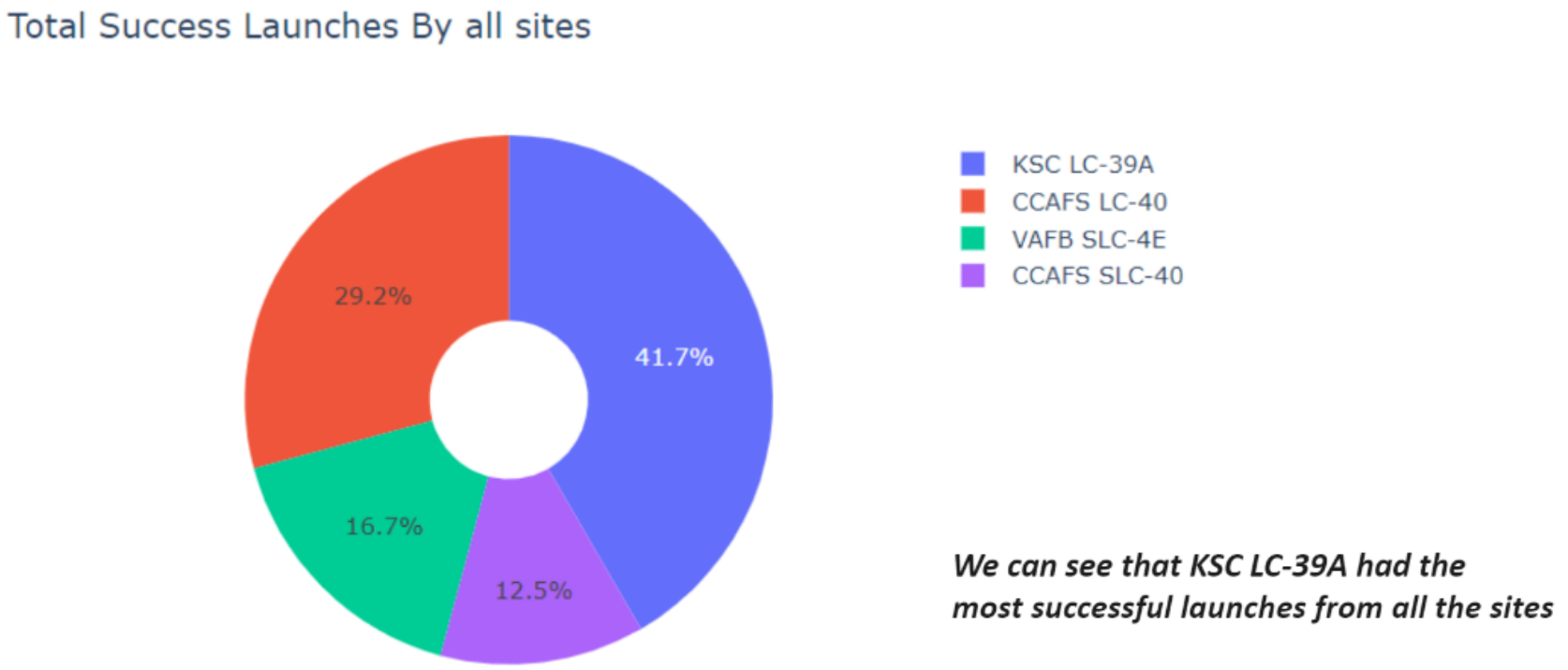






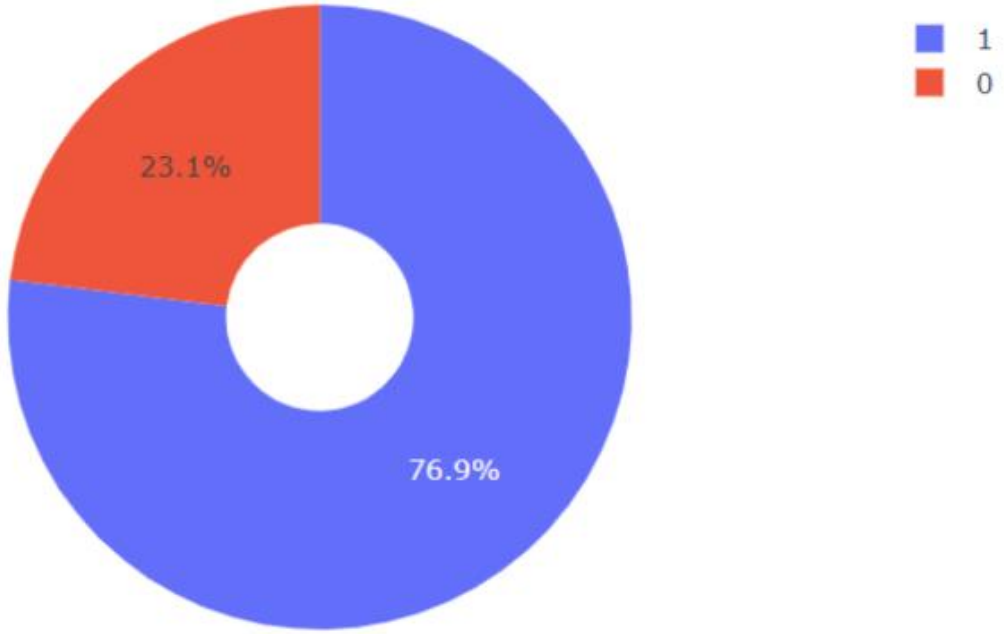
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Successful launches Across launch Sites



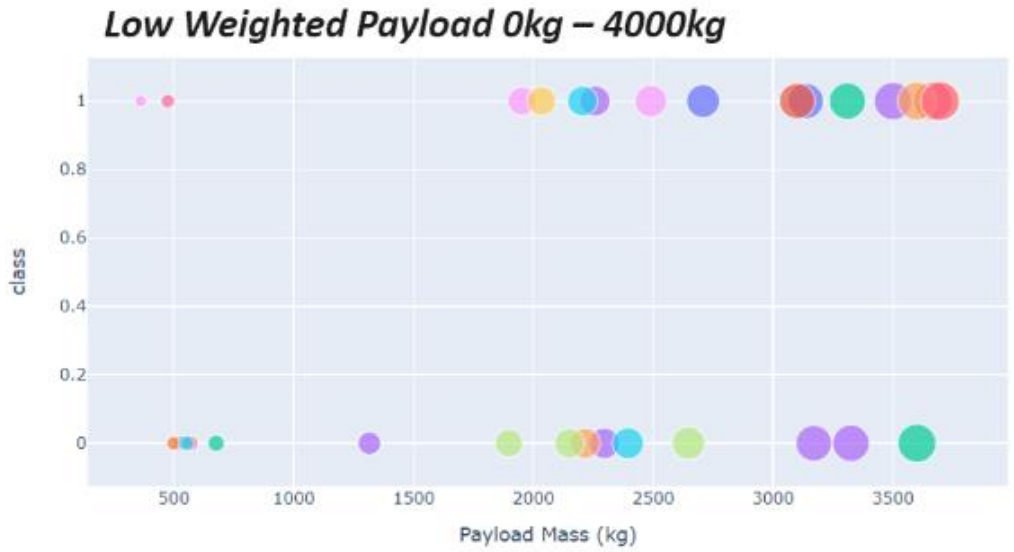
We can see that KSC LC-39A had the most successful launches from all the sites

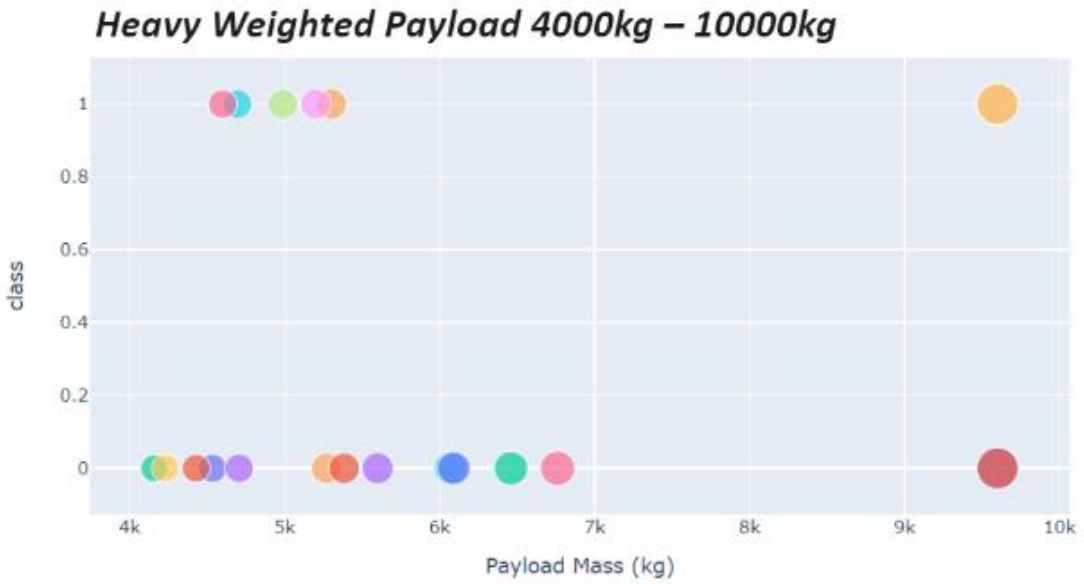
Pie chart showing the highest success ratio



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

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





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

Classification Accuracy

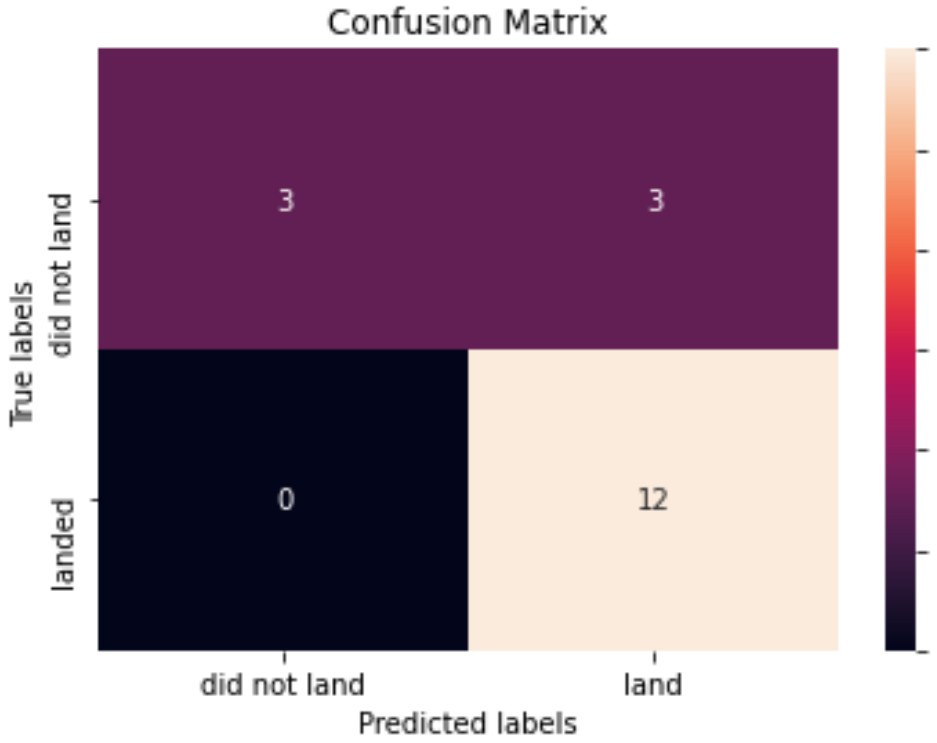
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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

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

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

Confusion Matrix EDIT

The confusion matrix for the decision tree classifier demonstrates its ability to distinguish between different classes. However, the primary issue identified is the high number of false positives, where unsuccessful landings are incorrectly classified as successful. Addressing this misclassification is crucial for enhancing the accuracy and reliability of the model.



Conclusions EDIT

It can be concluded that:

- A higher number of flights at a launch site corresponds to a greater success rate.
- Launch success rates began to increase from 2013 to 2020.
- Orbits ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.
- KSC LC-39A recorded the most successful launches among all sites.
- The Decision Tree classifier proved to be the most effective machine learning algorithm for this task.