IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Subba Reddy Alla
06/19/2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics
  - Predictive Analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo
  - Predictive analysis results

Introduction

- Background and context of the project
- The **commercial space race** is gaining momentum, with firms such as SpaceX, Blue Origin, Rocket Lab, and Virgin Galactic transforming access to space.
- **SpaceX's Falcon 9** is an affordable launch option, mainly due to its **reusable first stage**.
- A Falcon 9 launch costs around **$62M**, while competitors charge **$165M+**.
- Savings arise from the **successful recovery** of the **first stage**, which is vital for launch cost efficiency.

Questions to address

- Can we **forecast the successful landing of the Falcon 9 first stage**?
- If affirmative, we can **assess launch costs** and gauge **SpaceX's efficiency**.
- This allows for **competitive analysis** for new players (e.g., our hypothetical company "Space Y") to enter the market.
- Rather than relying on physics-based simulations, we'll employ **machine learning** and **public data** to predict outcomes for the first stage.

Section 1

Methodology

Methodology

Data Collection Method:

Retrieved data from the SpaceX REST API (api.spacexdata.com/v4/launches/past) and scraped Wikipedia Falcon 9 records using BeautifulSoup.

Data Wrangling:

Parsed launch data with requests and .json(), converted JSON to DataFrame with json_normalize, and obtained additional info using rocket/payload/launchpad IDs. Filtered for Falcon 9 launches and replaced nulls in PayloadMass with the mean, leaving LandingPad nulls for one-hot encoding.

Exploratory Data Analysis (EDA):

Analyzed data distribution, missing values, and landing success patterns.

Interactive Visual Analytics:

Mapped launch and landing sites using Folium and created dashboards with Plotly Dash.

Predictive Analysis:

Trained machine learning models (LogReg, SVM, Decision Tree, etc.), tuned with GridSearchCV, and evaluated accuracy, precision, recall, and F1 score.

Data Collection

Using SpaceX Rest API

GET request is sent using Python's library

Response is in JSON format

Converted into flat table using json_normalize

Enriching data with Additional API calls

Some fields (e.g., rocket, launchpad, payload, core) are referenced by ID.

Additional API calls are made to other endpoints to retrieve detailed information for these IDs.

Using Web Scraping

Falcon 9 launch data is also scraped from Wikipedia using BeautifulSoup.

HTML tables are parsed and converted into Pandas dataframes.

Data Wrangling

The collected data is cleaned and transformed:

Filter out non-Falcon 9 launches (e.g., Falcon 1).

Handle missing values (e.g., replace nulls in PayloadMass with the column's mean).

Leave certain nulls (e.g., LandingPad) intact for later processing like one-hot encoding.

Data Collection – SpaceX API

DATA COLLECTION: REQUEST & PARSE THE SPACEX LAUNCH DATA USING THE GET REQUEST

DATA COLLECTION: FILTER THE DATAFRAME TO ONLY INCLUDE LAUNCHES

DATA WRANGLING: MISSING ISSUES

GitHub URL:
https://github.com/voyager3000/IBMDataScience/blob/main/01_Data_Collection.ipynb

Data Collection - Scraping

Request a HTTP response using GET method

Extract all names from HTML table header

Create a data frame by parsing the launch HTML tables

- GitHub URL: https://github.com/voyager3000/IBMDataScience/blob/main/02_Web_Scraping.ipynb

Data Wrangling

01

Load data from CSV file

02

Calculate the number of launches on each site

03

Calculate the number and occurrence of each orbit

04

Calculate the number and occurrence of mission outcome of the orbits

05

Create a landing outcome label from Outcome column

- GitHub URL: https://github.com/voyager3000/IBMDataScience/blob/main/03_Data_Wrangling.ipynb

EDA with Data Visualization

- Scatterplots were used to depict the following:
  - FlightNumber vs. PayloadMass
  - Flight Number and Launch Site
  - FlightNumber and Orbit type
  - Payload Mass and Orbit type
- Bar chart was used to depict the following:
  - Success rate of each orbit type
- Line chart was used to depict the following:
  - Launch success yearly trend

GitHub URL:

https://github.com/voyager3000/IBMDataScience/blob/main/05_EDA_Visualization.ipynb

EDA with SQL

The following are SQL activities done:

- Table created: SPACEXTABLE
- Identified unique launch sites: DISTINCT
- Records for launch sites beginning with 'CCA': LIKE 'CCA%'
- Total payload mass recorded by NASA (CRS): SUM("PAYLOAD_MASS__KG_")
- Average payload mass for the booster version F9 v1.1: AVG("PAYLOAD_MASS__KG_")
- Date of the first successful landing outcome on the ground pad: MIN("Date")

GitHub URL: https://github.com/voyager3000/IBMDataScience/blob/main/04_EDA_SQL.ipynb

Build an Interactive Map with Folium

Blue circle at NASA Johnson's Space Center's coordinate to highlight this location

Circles were added to denote proximity zones

Lines were added to denote distances between nearby features

- GitHub URL: https://github.com/voyager3000/IBMDataScience/blob/main/06_Visual_Analytics_Folium.ipynb

Build a Dashboard with Plotly Dash

- Created a Pie chart to denote the total successful launches count for all sites
- Added a slider feature for the user to select the payload range
- Added a Scatter chart to show the correlation between payload and launch success

GitHub URL:

https://github.com/voyager3000/IBMDataScience/blob/main/08_Visual_Analytics_Plotly.py

Predictive Analysis (Classification)

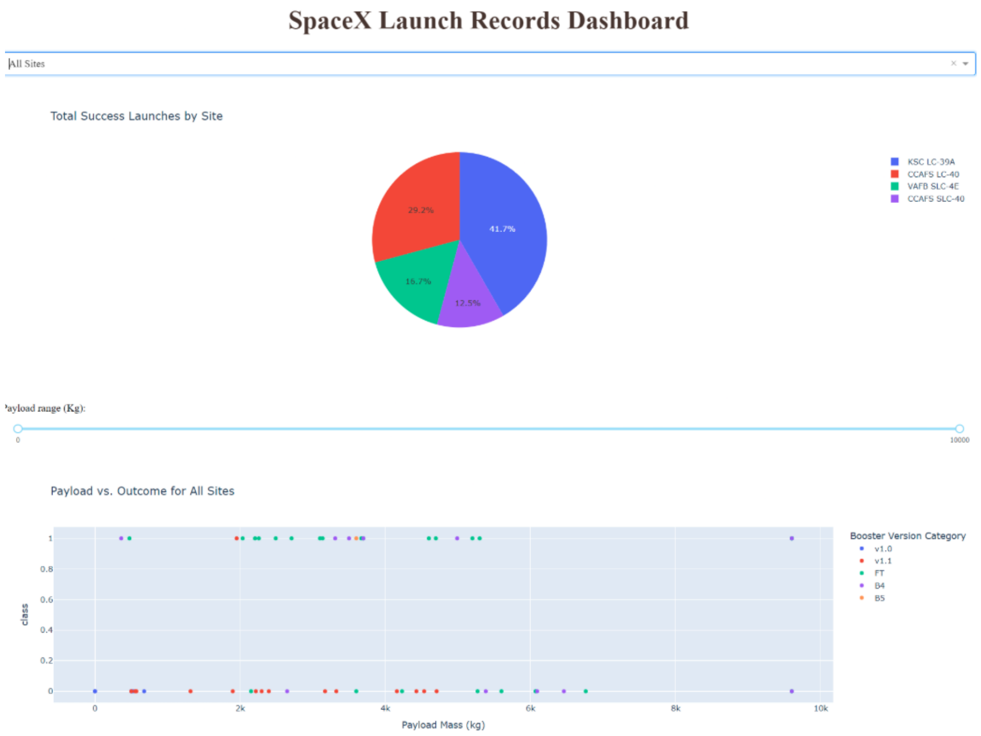
Created a pandas data frame → Split data into training vs. training sets → Trained models → Evaluated the models → Compared models to evaluate their accuracy scores

GitHub URL:

https://github.com/voyager3000/IBMDataScience/blob/main/07_Predictive_Analytics.ipynb

Results

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

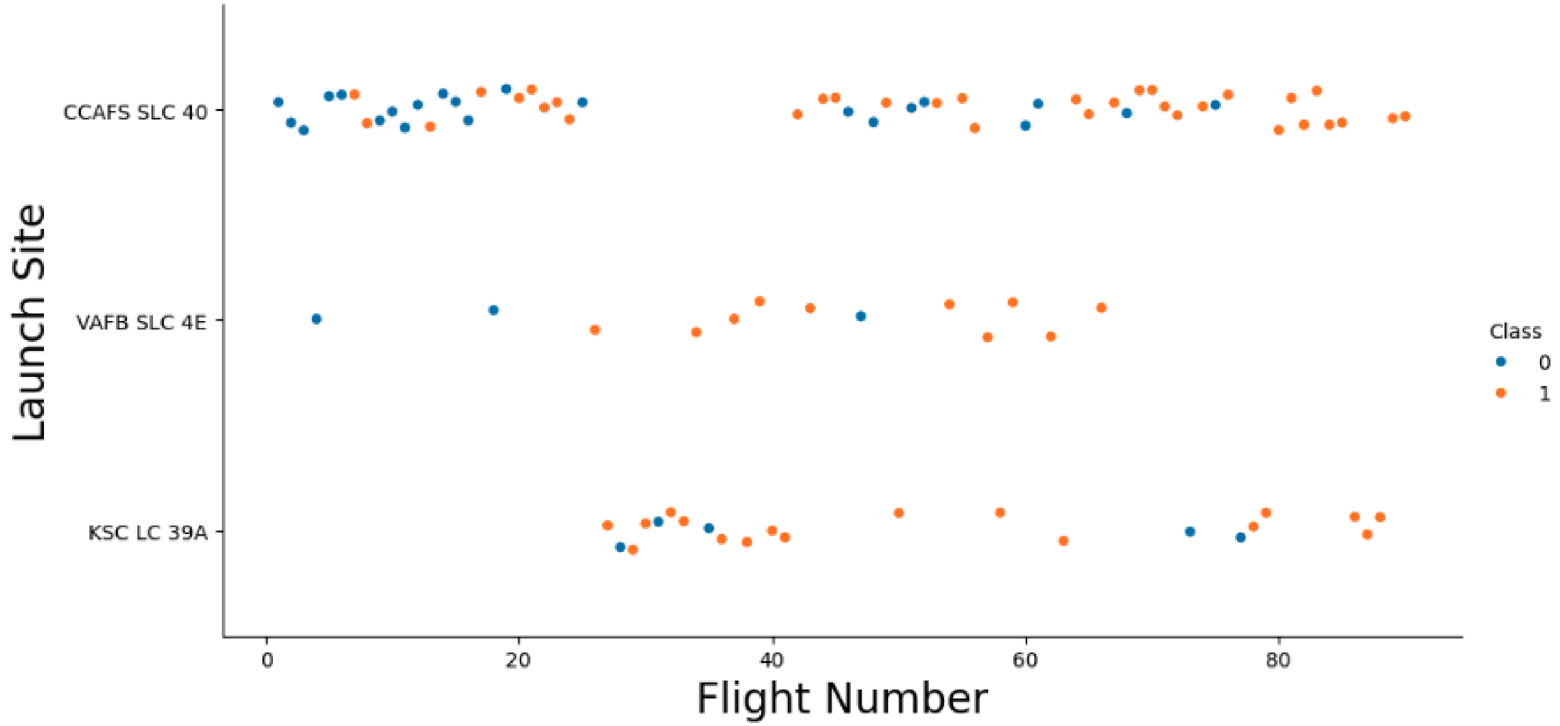


Section 2

Insights drawn from EDA

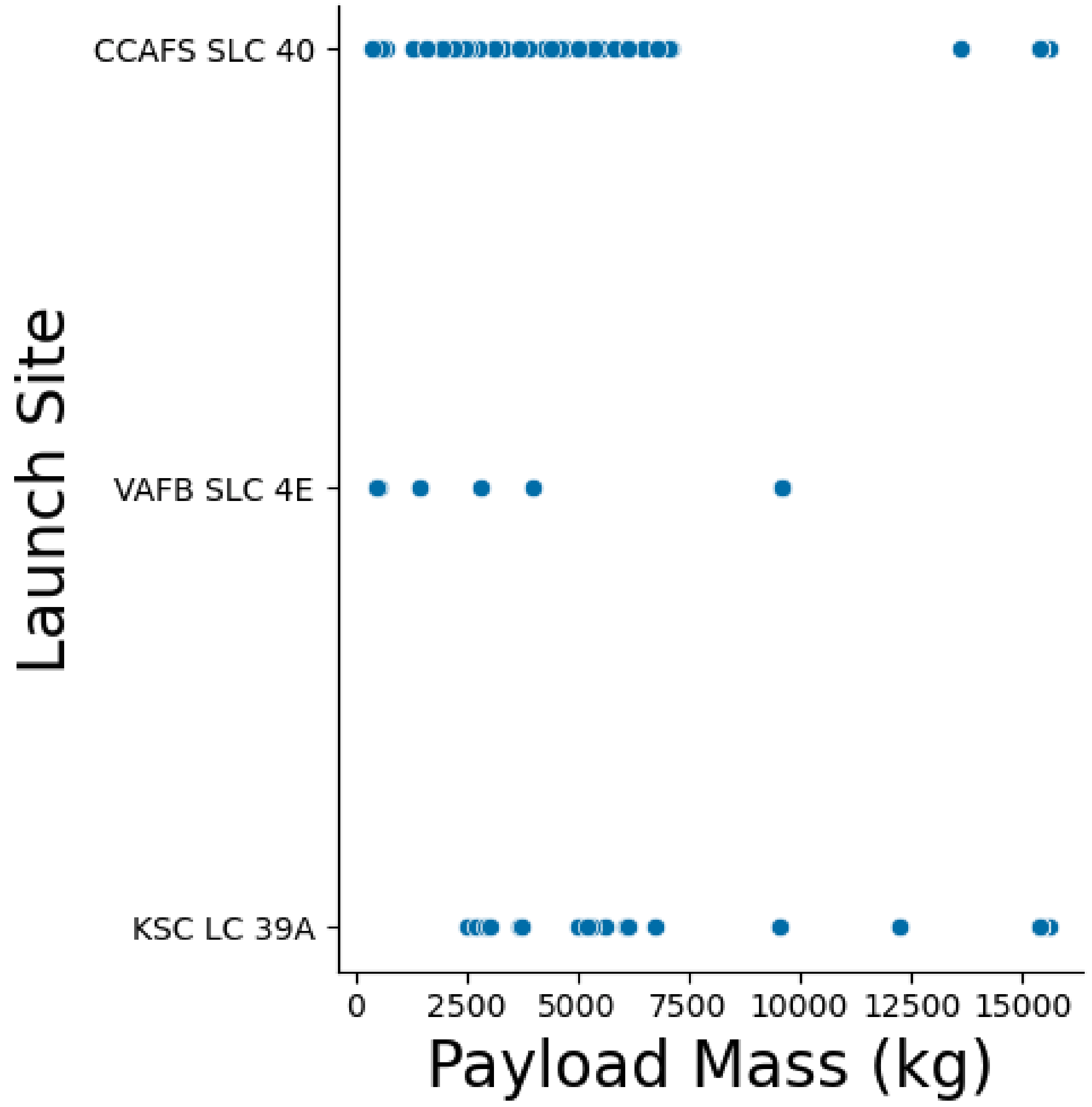
Flight Number vs. Launch Site

- **Analysis**:
- There was a higher success rate with the flights shown in orange
- The flights shown in blue had lower success rates
- CCAFS SLC 40 & KSC LC 39A had a mix of success & failed flights however the VAFB SLC 4E has been fairly successful



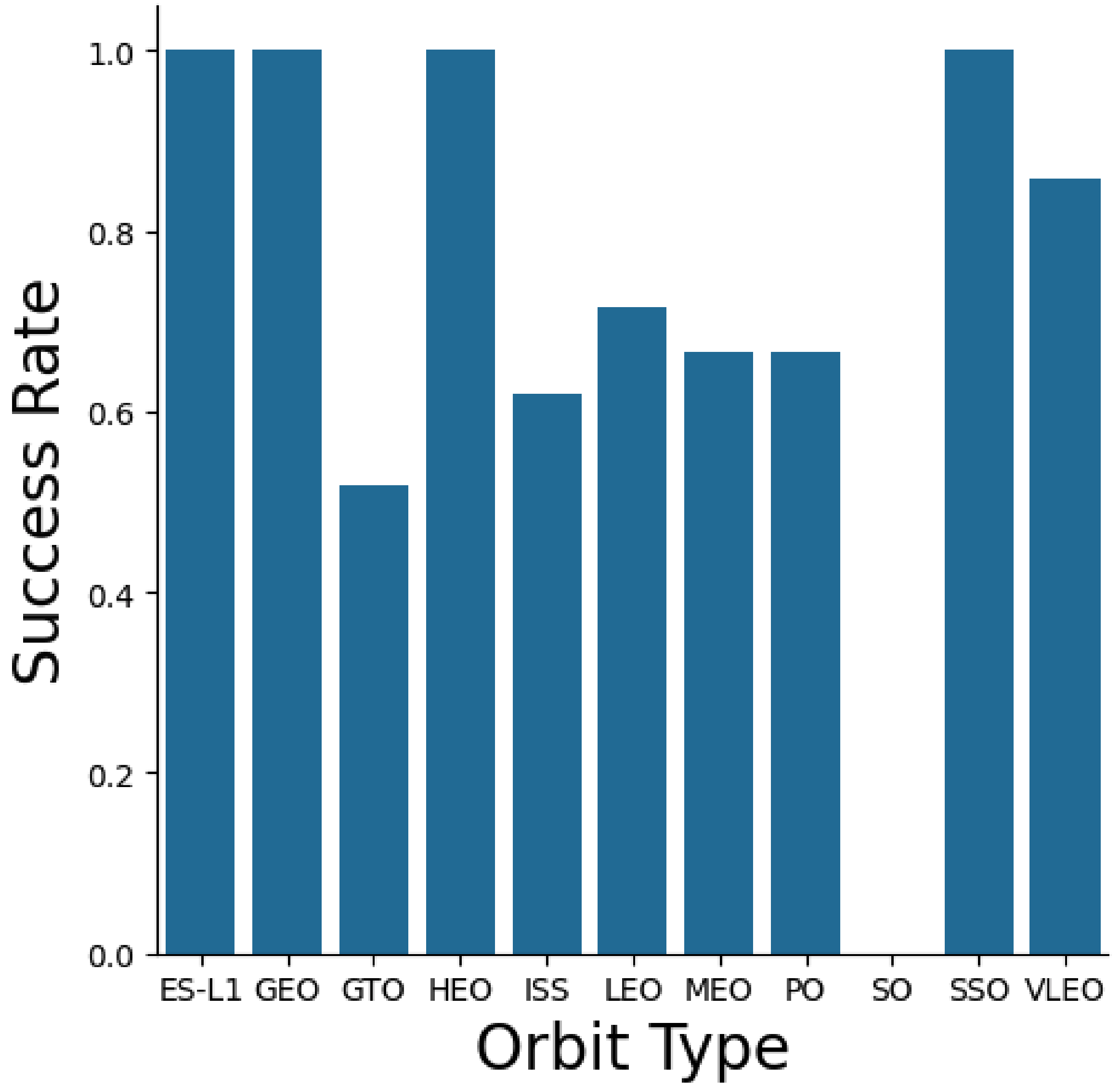
Payload vs. Launch Site

- The launch site - CCAFS SLC 40 seems to carry either a payload mass of less than 8000 kg or more than 15000 kg
- The launch site – VAFB SLC 4E seems to have a consistent payload of 9500 kg



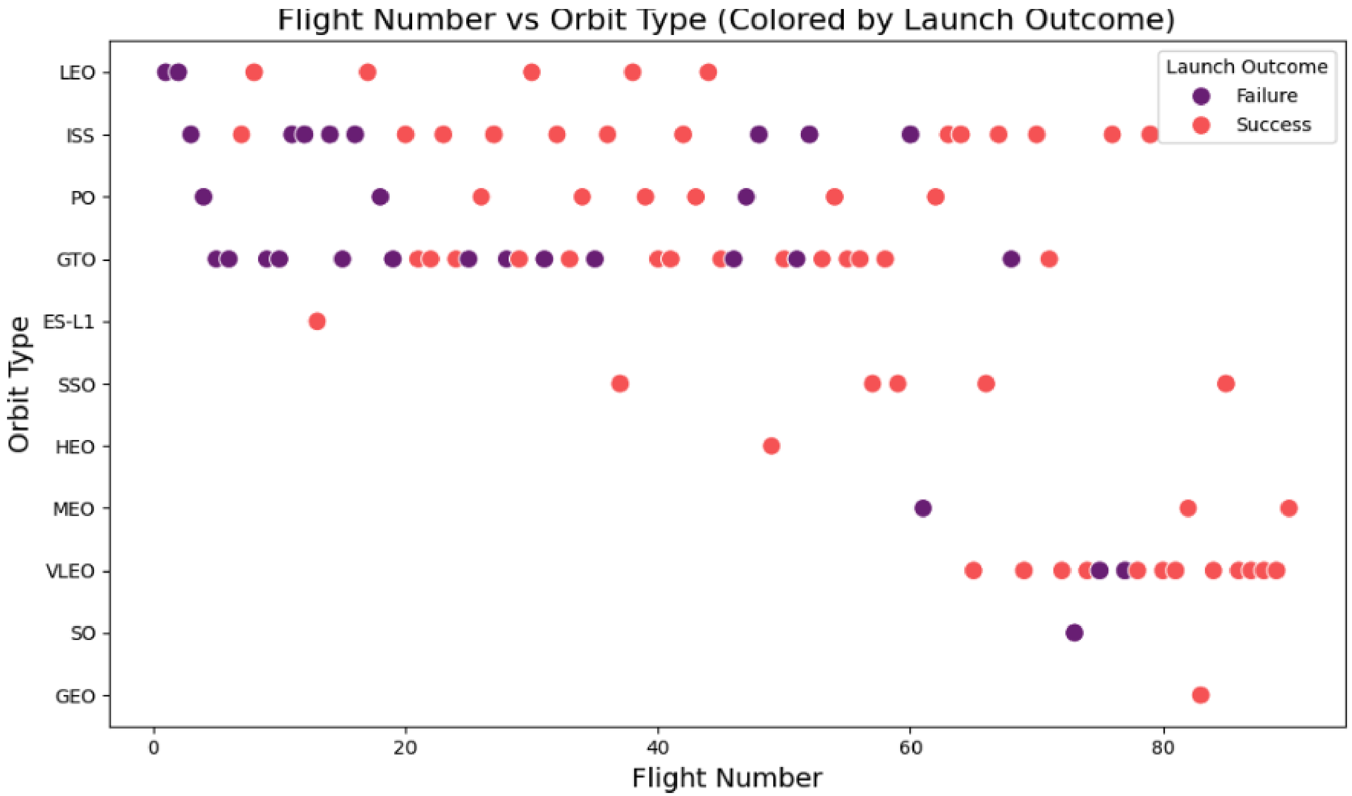
Success Rate vs. Orbit Type

- The orbit type ES-L1, SSO, HEO & GEO orbits seem to have the highest success rate
- The orbit type SO has no success rate



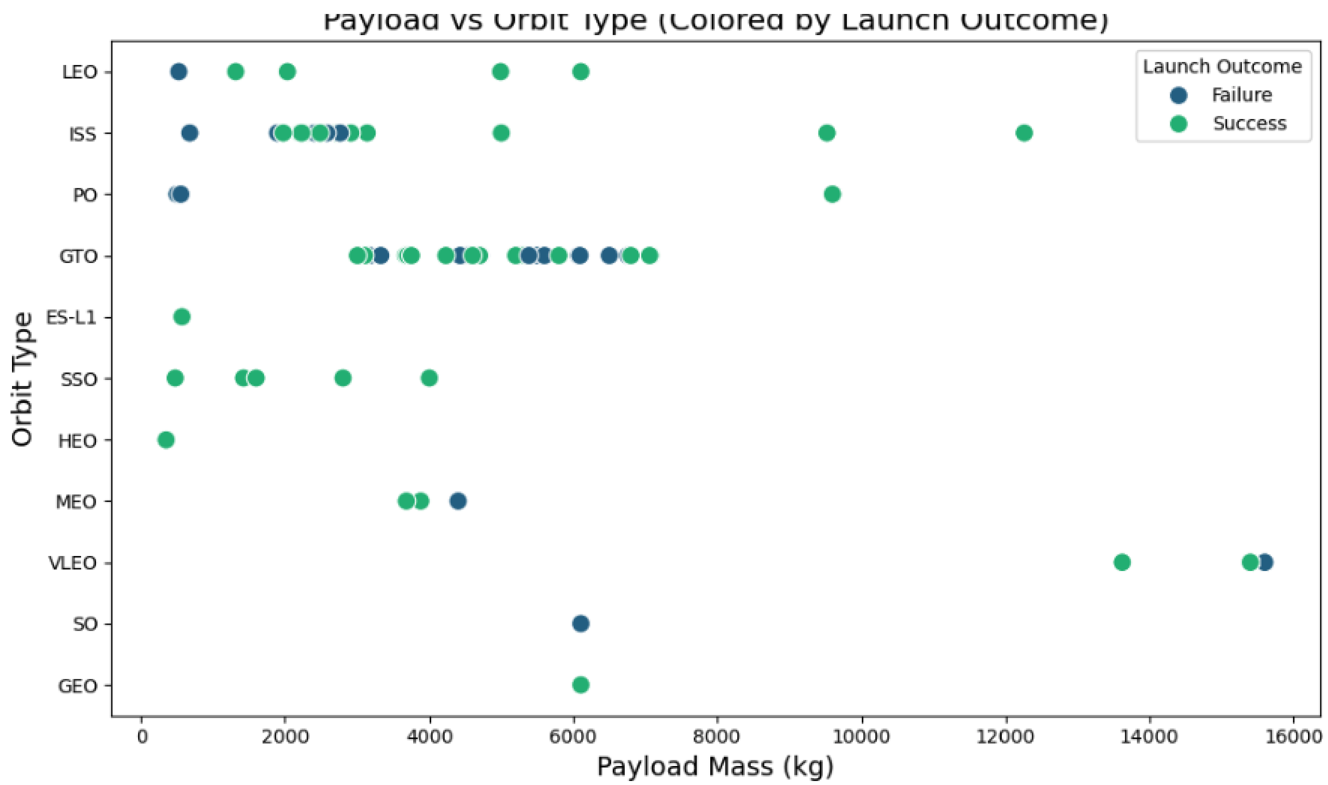
Flight Number vs. Orbit Type

- There were higher flights in the VLEO orbit
- There are more flights taking place in GTO & ISS orbit



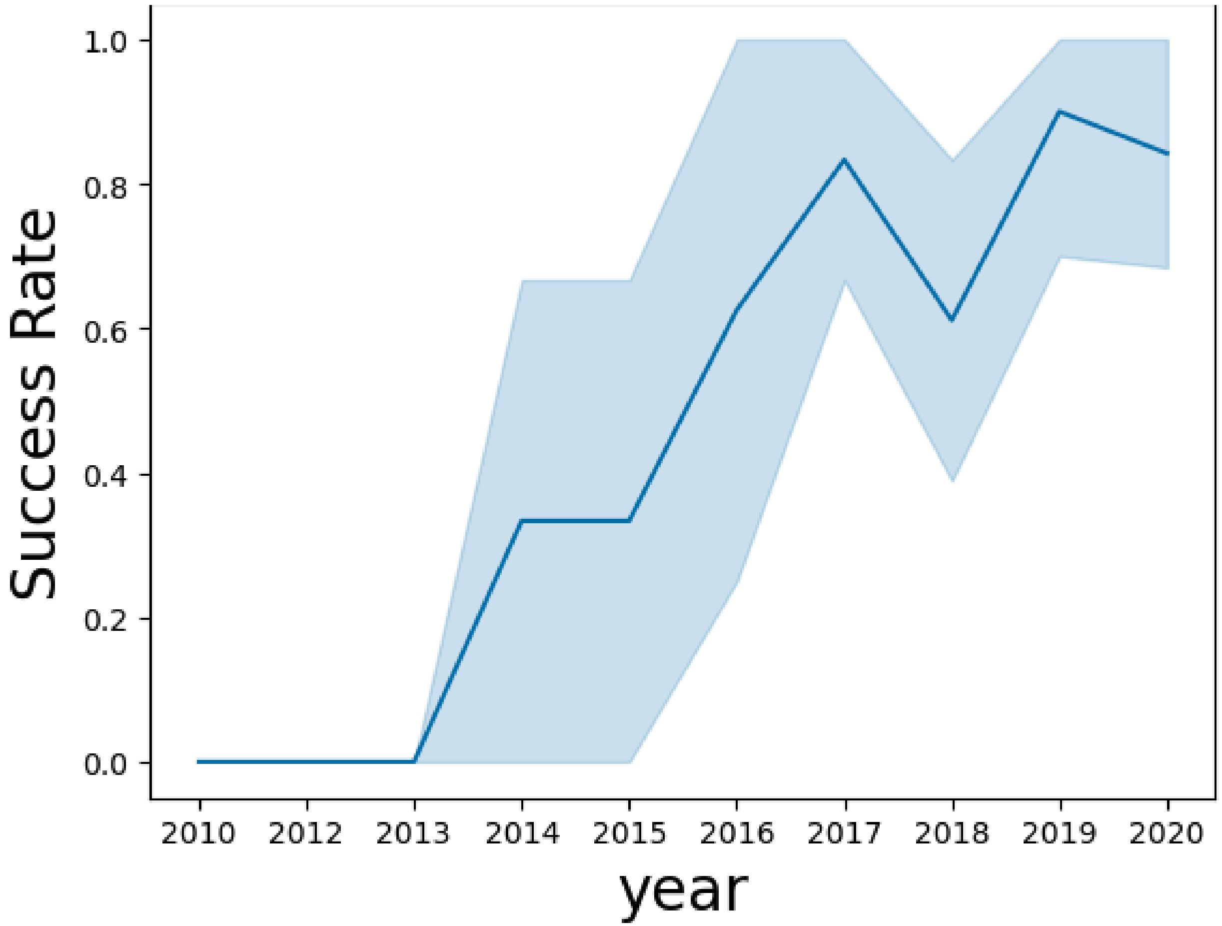
Payload vs. Orbit Type

- The payload between 2000 kg to 4000 kg is successful in the ISS orbit
- The payload between 3000 kg to 8000 kg is more successful in the GTO orbit



Launch Success Yearly Trend

- The success rate has been on an upwards trend since 2013.
- There was a decrease in the success rate in 2018, and there is also a dip between the years of 2019 & 2020



All Launch Site Names

- When the following query is executed in the dataset - SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE, the results are 5 unique names that are presented

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

Launch Site Names Begin with 'CCA'

The following query displays the launch sites that start with the ‘CCA’ in the data set

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

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

```
* sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- Total payload mass carried = 45596 KG
- The below query displays the total payload mass carried by the boosters launched by NASA

Task 3

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

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 = 2928.4 KG
- Below query display the calculation of the Average payload mass carried by F9 v1.1

Task 4

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

In [25]:
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[25]:

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- Below is the query to find the first successful landing outcome on the ground pad. And it happened on "December 22nd 2015"

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [26]:
```
%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[26]:

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Below query lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Query: SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [27]:
```
LE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[27]:

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

Total Number of Successful and Failure Mission Outcomes

- total # of successful and failure mission outcomes = 98
- Below is the query to calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

In [28]:

```
%sql SELECT "Mission_Outcome", COUNT(*) AS "Total" FROM SPACEXTABLE WHERE "Mission_Outcome" IN ('Success', 'Failu
```

```
* sqlite:///my_data1.db
Done.
```

Out[28]:

Mission_Outcome	Total
Success	98

Boosters Carried Maximum Payload

Here the query I used to get this result:

```
SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
```

Here is the list the boosters that have carried the maximum payload mass

F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Please see the query used to find the list of failed landing outcomes and the result displaying the details of the drone ship, booster versions, and launch site names for the year 2015

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [30]:

```
%%sql

SELECT
    CASE
        WHEN substr("Date", 6, 2) = '01' THEN 'January'
        WHEN substr("Date", 6, 2) = '02' THEN 'February'
        WHEN substr("Date", 6, 2) = '03' THEN 'March'
        WHEN substr("Date", 6, 2) = '04' THEN 'April'
        WHEN substr("Date", 6, 2) = '05' THEN 'May'
        WHEN substr("Date", 6, 2) = '06' THEN 'June'
        WHEN substr("Date", 6, 2) = '07' THEN 'July'
        WHEN substr("Date", 6, 2) = '08' THEN 'August'
        WHEN substr("Date", 6, 2) = '09' THEN 'September'
        WHEN substr("Date", 6, 2) = '10' THEN 'October'
        WHEN substr("Date", 6, 2) = '11' THEN 'November'
        WHEN substr("Date", 6, 2) = '12' THEN 'December'
        ELSE 'Unknown'
    END AS "Month_Name",
    "Mission_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM
    SPACEXTABLE
WHERE
    substr("Date", 0, 5) = '2015';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[30]:

Month_Name	Mission_Outcome	Booster_Version	Launch_Site
January	Success	F9 v1.1 B1012	CCAFS LC-40
February	Success	F9 v1.1 B1013	CCAFS LC-40
March	Success	F9 v1.1 B1014	CCAFS LC-40
April	Success	F9 v1.1 B1015	CCAFS LC-40
April	Success	F9 v1.1 B1016	CCAFS LC-40
June	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40

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

Here is the Query to Rank the counts (in descending order) of each type of landing outcomes happened between the date 2010-06-04 and 2017-03-20

Task 10

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.

In [31]:

```
%%sql

SELECT
    "Landing_Outcome",
    COUNT(*) AS "Count"
FROM
    SPACEXTABLE
WHERE
    "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    "Landing_Outcome"
ORDER BY
    COUNT(*) DESC;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[31]:

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

Section 3

Launch Sites Proximities Analysis

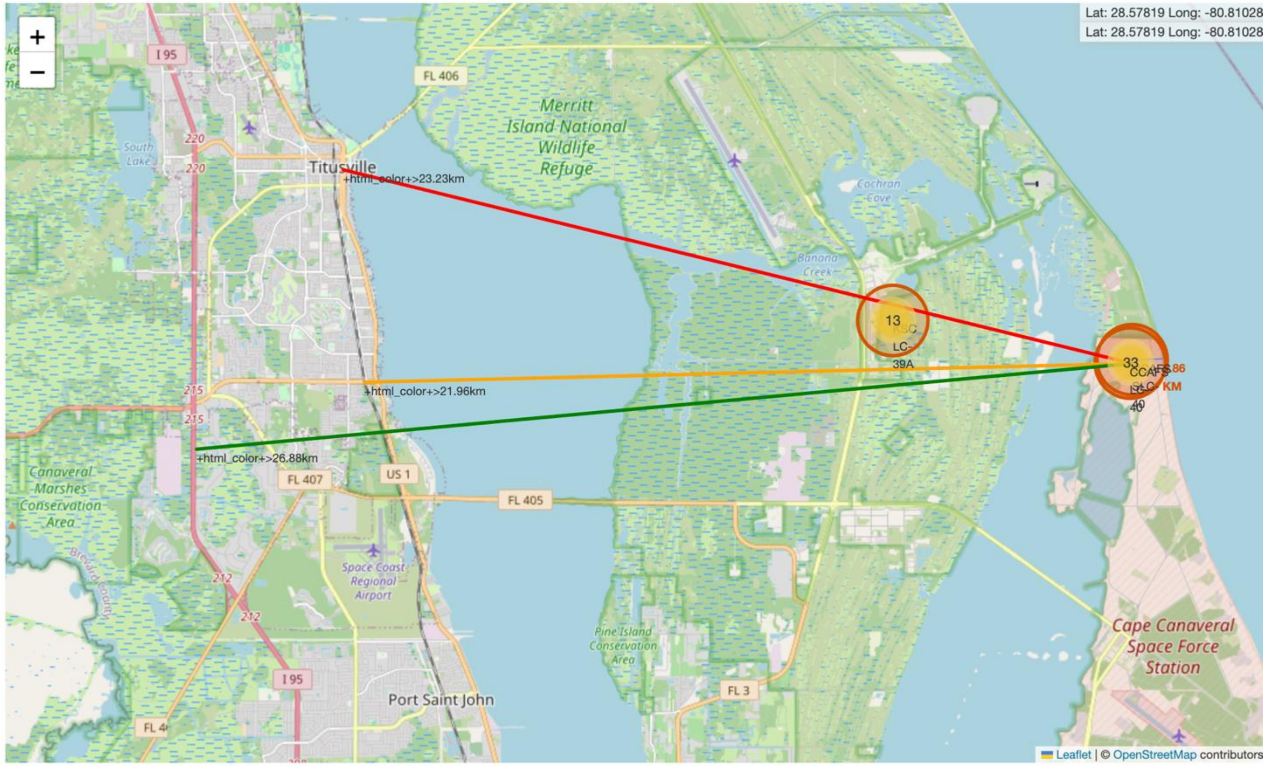
Launch Site Analysis

- I have notice two Important elements
  - All the launch sites are closest to the coast for obvious reasons “Public Safety”
  - And the launch sites in Florida are comparatively closer to the equator making them favorable for the rockets needing to go into the Equatorial Orbits
- Below is the screenshot of the Launch Sites on the global map.

Launch Outcomes

- In this screenshots, we can see 3 different launch sites with Outcomes displayed using Red and Green markers
- Launch site CCAFS SLC-40 seem to have close to 45% success rate (3/7).
- Lauch site CCAFS LC-40 seem to have very less success outcomes (7/26).
- On the other hand, Launch site KSC LC-39A seem to have (10/13) successful launch outcomes resulting in highest success rate site of 76%

Transport Proximities



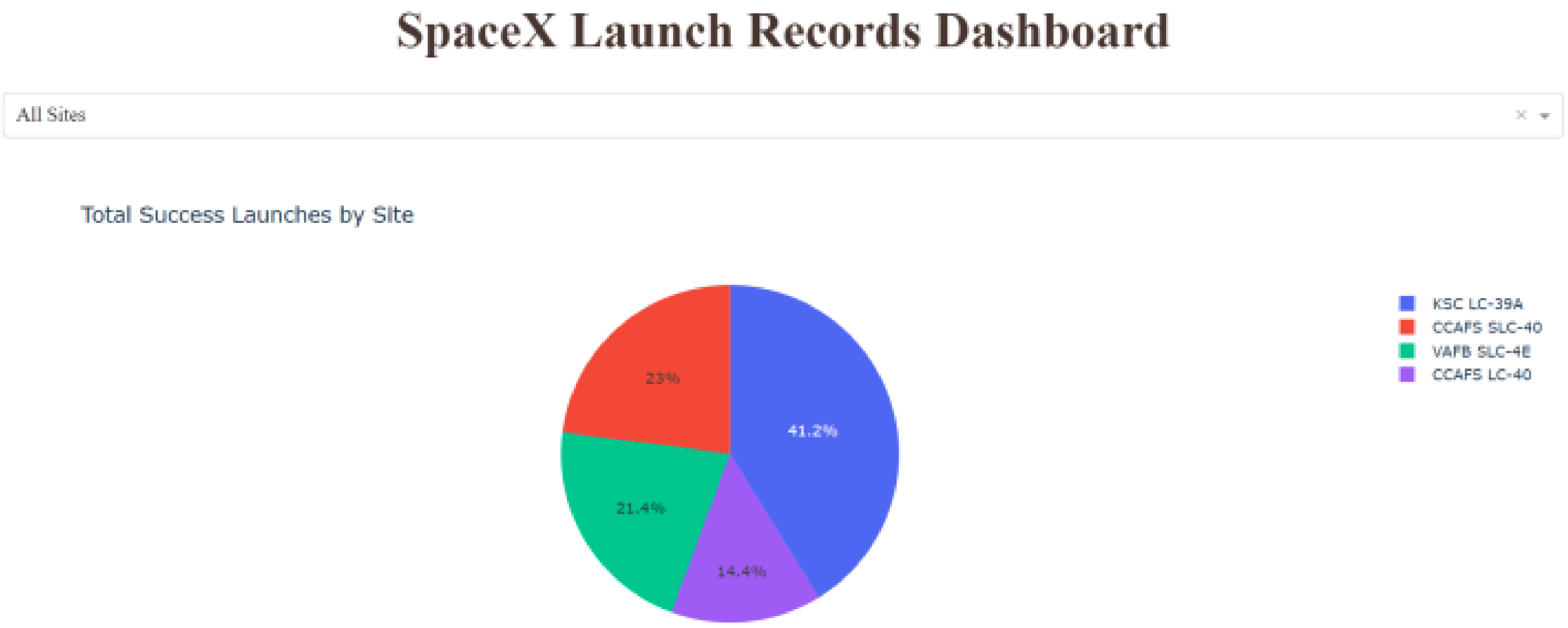
- Based on this screenshot, most successful launch sites CCAFS SLC-40 and CCAFS LC-40 are situated in a very convenient location.
  - Less than 1KM from Coastline
  - 26.9 KM from the I95 that goes from Maine to Key West covering all North America's east coast
  - 22 KM from the closest railway line
  - And 23.3 KM from the closest City

Section 4

Build a Dashboard with Plotly Dash

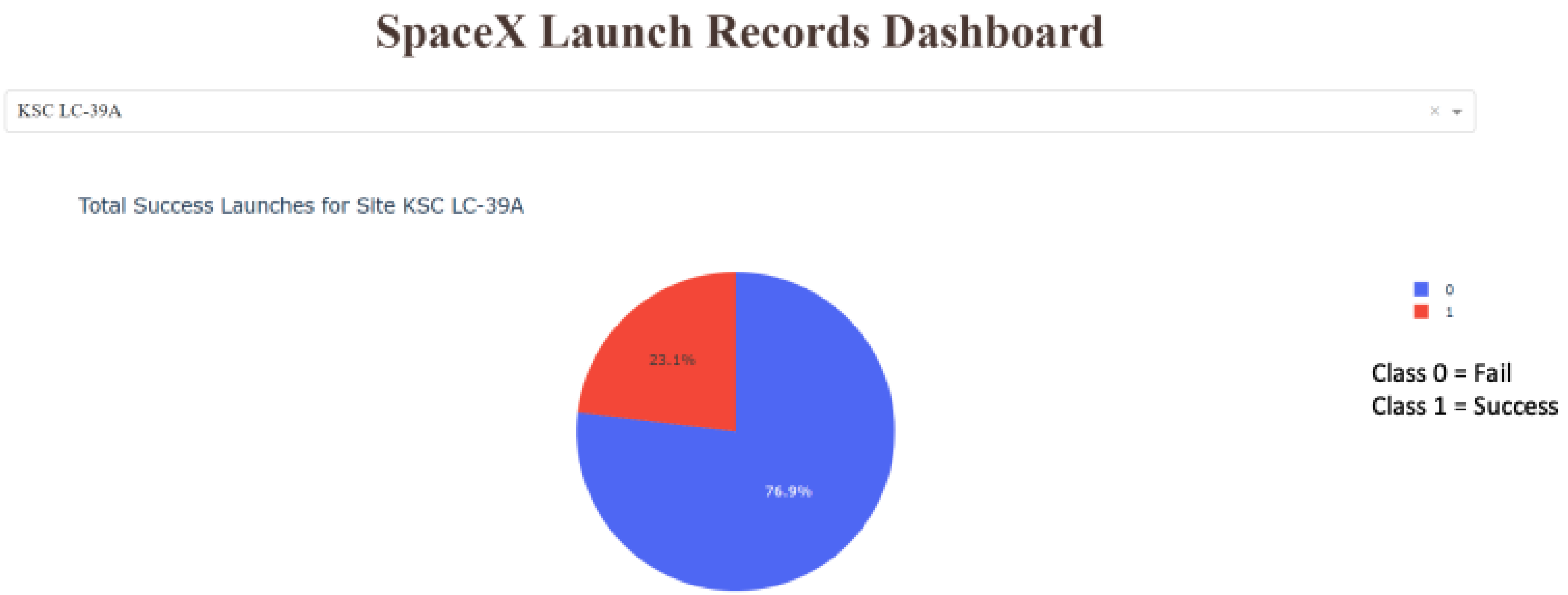
Successful Launches by Site

- Based on this Pie Chart, we can understand that KSC LC-39A seem to have more successful launches in comparison to other Sites, covering 41% success rate.



Highest Launch Success Ratio

Based on the previous chart, we have identified that KSC LC-39A seem to be successful launch site. The below pie will explain the ratio of failed launches to successful ones. Only 3 failed of 13 launches, Resulting in 76.9% Success rate.



Sweet Spot for Payload Mass and Successful Launch (by Booster Version)

- Below is the Payload to Launch Outcomes Scatter Plot by Booster Version.
  - FT Version seem to more successful in comparison to other boosters.
  - B4 seem to hold the record for carrying the heaviest payload (9k+ KG)
- And perfect sweet spot for payload to success launch is between 2k to 4k KG
- Perfect combination would be FT booster with payload mass between 2000 and lower 3000 KG

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Although all the models performed and had same scores.
- Decision Tree model have subtly outperformed the other based on the Best model score = 0.9017857142857144

TASK 12

Find the method performs best:

In [30]:

```python
accuracy = [svm_cv_score, logreg_score, knn_cv_score, tree_cv_score]
accuracy = [i * 100 for i in accuracy]

method = ['Support Vector Machine', 'Logistic Regression', 'K Nearest Neighbour', 'Decision Tree']
models = {'ML Method':method, 'Accuracy Score (%)':accuracy}

ML_df = pd.DataFrame(models)
ML_df
```

Out[30]:

	ML Method	Accuracy Score (%)
0	Support Vector Machine	83.333333
1	Logistic Regression	83.333333
2	K Nearest Neighbour	83.333333
3	Decision Tree	83.333333

In [31]:

```python
from sklearn.metrics import jaccard_score, f1_score

# Examining the scores from Test sets
jaccard_scores = [
                  jaccard_score(Y_test, logreg_yhat, average='binary'),
                  jaccard_score(Y_test, svm_yhat, average='binary'),
                  jaccard_score(Y_test, tree_yhat, average='binary'),
                  jaccard_score(Y_test, knn_yhat, average='binary'),
                 ]

f1_scores = [
             f1_score(Y_test, logreg_yhat, average='binary'),
             f1_score(Y_test, svm_yhat, average='binary'),
             f1_score(Y_test, tree_yhat, average='binary'),
             f1_score(Y_test, knn_yhat, average='binary'),
            ]

accuracy = [logreg_score, svm_cv_score, tree_cv_score, knn_cv_score]

scores_test = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]), index=['Jaccard_Score', 'F1_Score', '
scores_test
```

Out[31]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

In [32]:

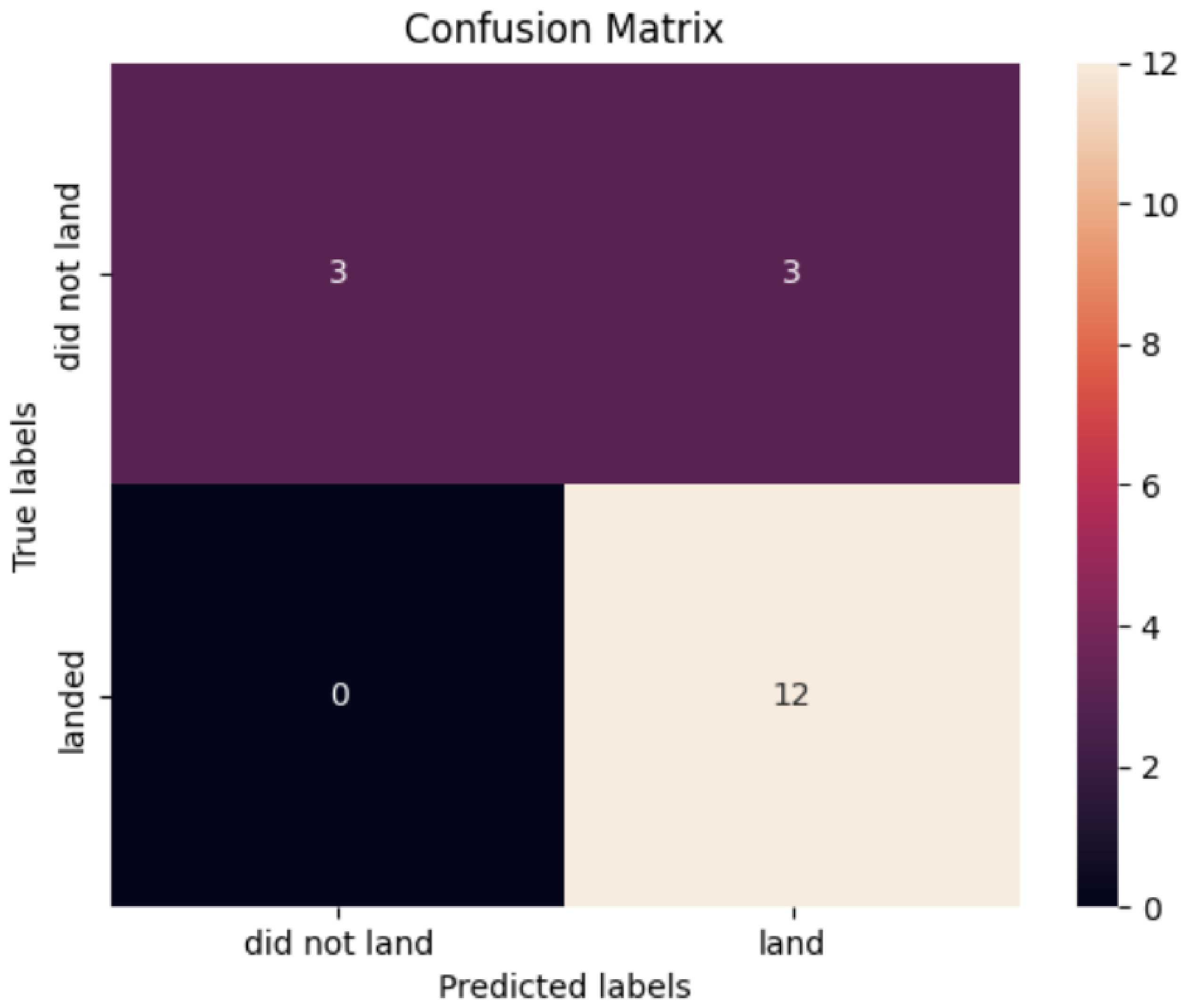
```python
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

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 0.9017857142857144
Best params is : {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_sample
s_split': 2, 'splitter': 'best'}
```

Confusion Matrix

- All the confusion matrices were identical
- Confusion Matrix Outputs:
  - 12 True positive
  - 3 True negative
  - **3 False positive**
  - 0 False Negative
- **Precision**= .80
- **Recall**= 1
- **F1 Score**=.89
- **Accuracy**=.833



Conclusions

- Findings from the Project:
  - KSC LC-39A is highest launch success rate site or 79%
  - All the launch sites are close to Coastlines for public safety.
  - Launch sites close to equator can take boost from rotational speed of earth, resulting in lesser fuel and lower cost
  - Orbits ES-L1, GEO, HEO and SSO have highest success rate
  - B4 carried heaviest payload and 2000 to 4000 KG seem to be perfect payload for a successful launch
- Larger the dataset is better would be the predictive analytics as the findings can be generalized

Thank you!