IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Wayi Richard Dumba
05/11/2024

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

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- In this study, we will use predictive analytics to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

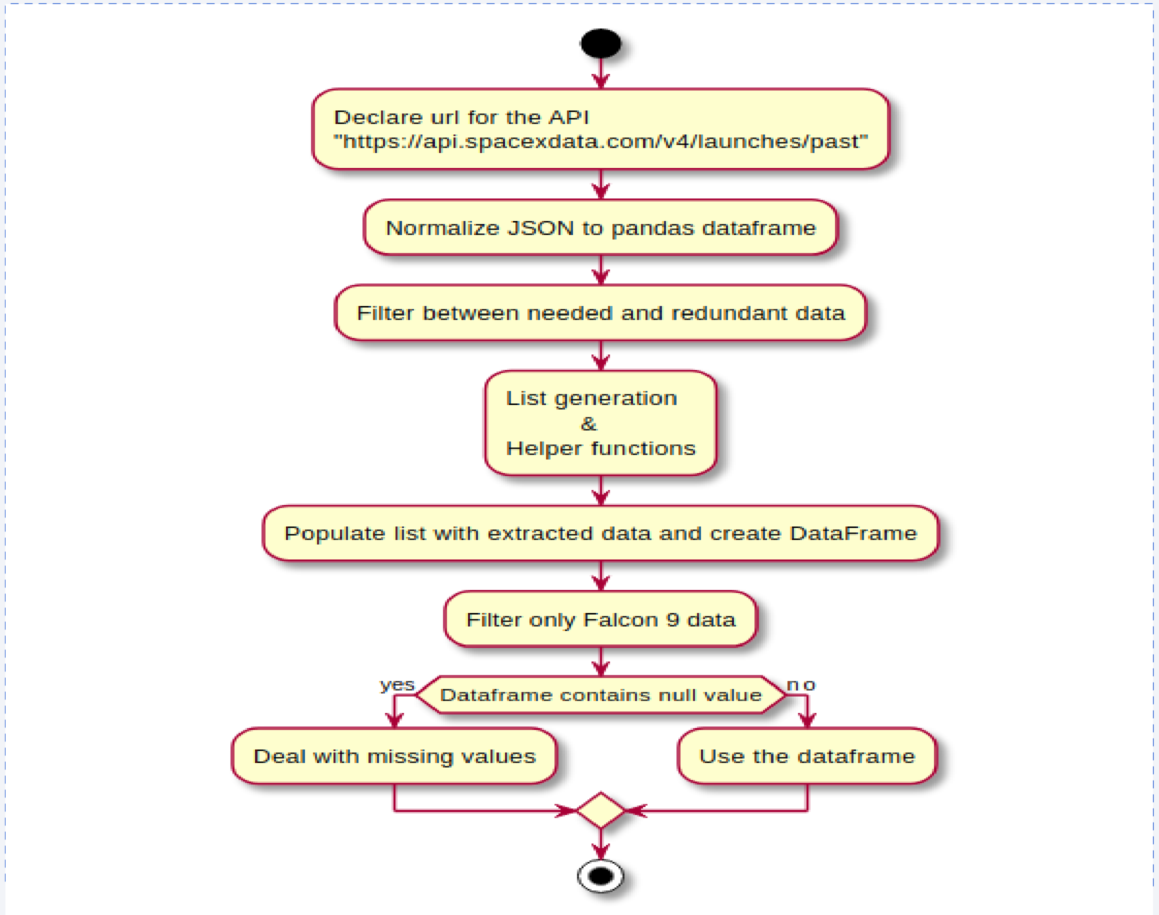
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was gathered in a number of ways.
  - Using a get request to the SpaceX API, data was gathered.
  - The response content was then decoded as JSON using the.json() function call, and converted to a pandas dataframe using the.json_normalize() function.
  - After cleaning the data, we looked for any missing values and, if needed, filled them in.
  - Additionally, we used BeautifulSoup to scrape Wikipedia for Falcon 9 launch records.
  - Extracting the launch records as an HTML table, parsing the table, and converting it to a Pandas dataframe for further analysis was the goal.

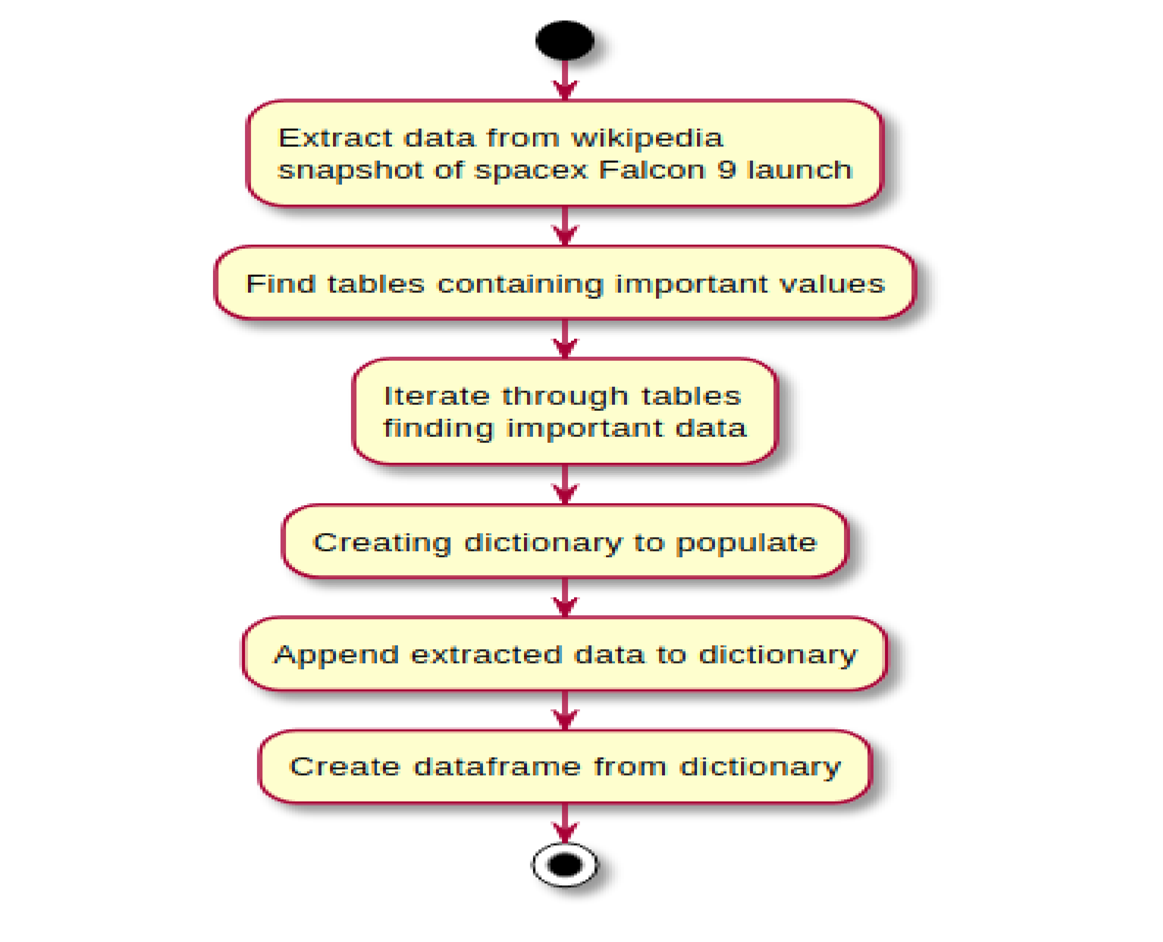
Data Collection – SpaceX API

- We collected data using the SpaceX API's get request, cleaned the requested data, and performed some simple formatting and data wrangling.
- The link to the notebook is below:
- https://github.com/wayirichard/IBM-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



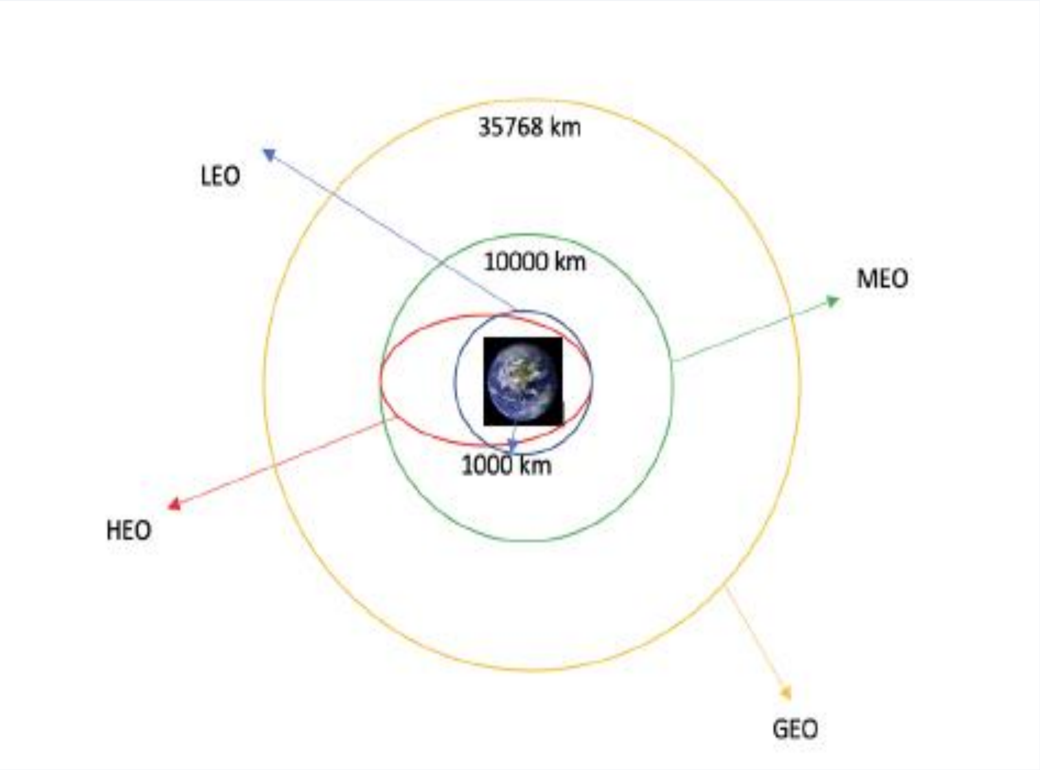
Data Collection - Scraping

- We used BeautifulSoup to perform web scraping on Falcon 9 launch records.
- The table was parsed, and a pandas dataframe was created.
- The link to the notebook is below:
- https://github.com/wayirichard/IBM-Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb



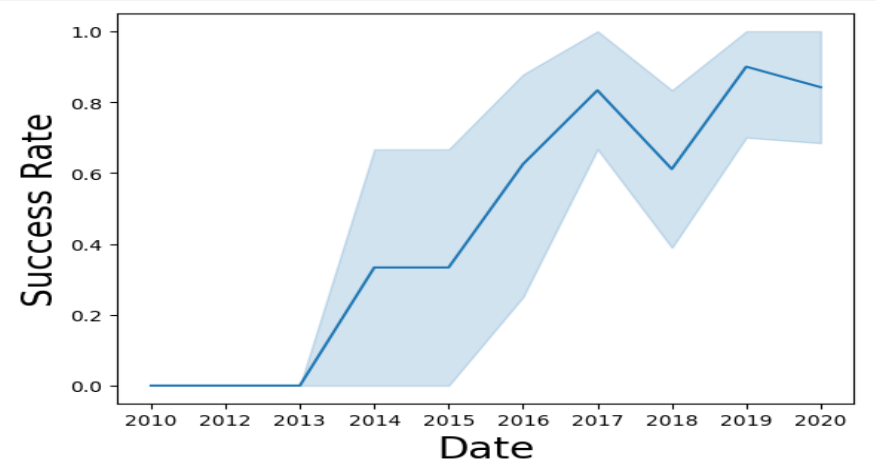
Data Wrangling

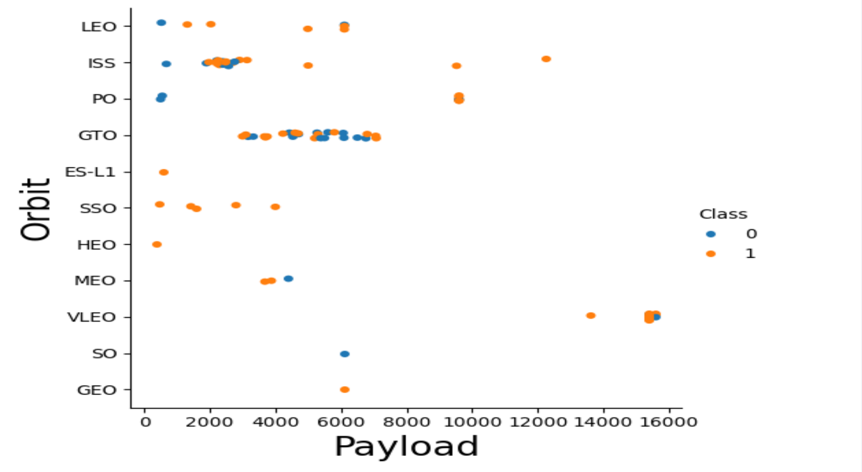
- We identified the training labels by conducting exploratory data analysis.
- We determined the number of launches at each location as well as the number of occurrences for each orbit.
- After creating a landing outcome label from the outcome column, we exported the results to a CSV file.
- The link to the note book: https://github.com/wayirichard/IBM-Capstone-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- The link to the notebook is: https://github.com/wayirichard/IBM-Capstone-Project/blob/main/edadataviz.ipynb





EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - 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 failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- The link to the notebook is: https://github.com/wayirichard/IBM-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We labeled every launch location and included map elements like markers, circles, and lines to indicate whether a launch was successful or unsuccessful for each location on the folium map.
- The feature launch outcomes (success or failure) were allocated to classes 0 and 1.For example, one for success and zero for failure.
- Through the use of color-labeled marker clusters, we were able to determine which launch sites have a favorable success rate.
- The distances between a launch site and its proximities were determined. We addressed a few questions, such as:
- Launch sites are located close to highways, railroads, and coastlines.
- Does the launch site maintain a specific distance from urban areas?
- The link to the notebook is: https://github.com/wayirichard/IBM-Capstone-Project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version
- The address to the web-App code is: https://github.com/wayirichard/IBM-Capstone-Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tuned different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model
- The link to the notebook is: https://github.com/wayirichard/IBM-Capstone-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

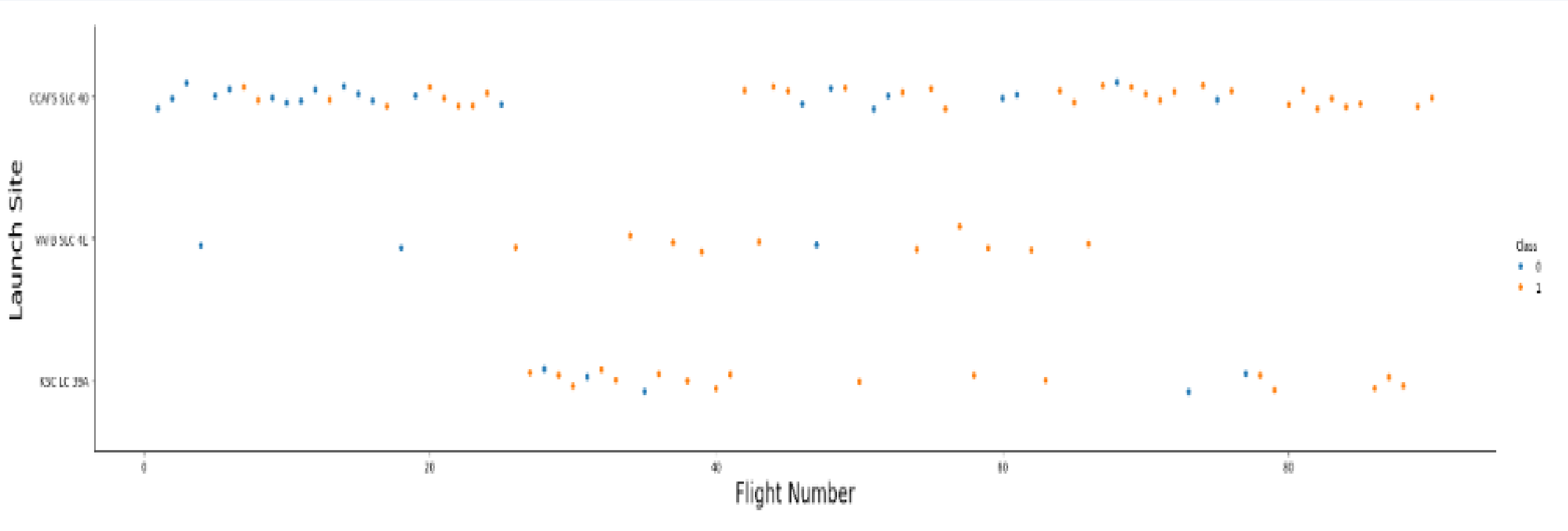
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

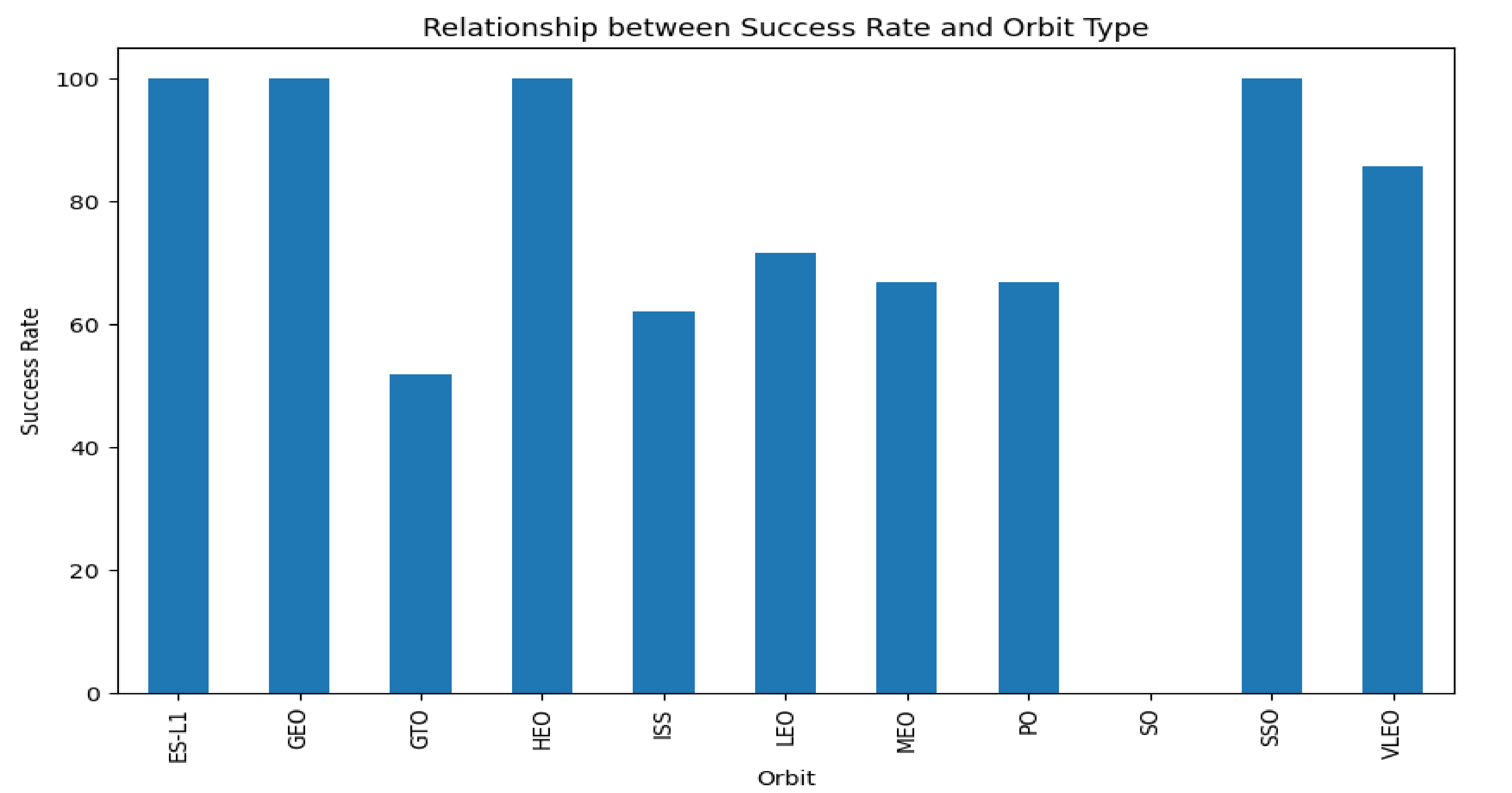


Payload vs. Launch Site

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

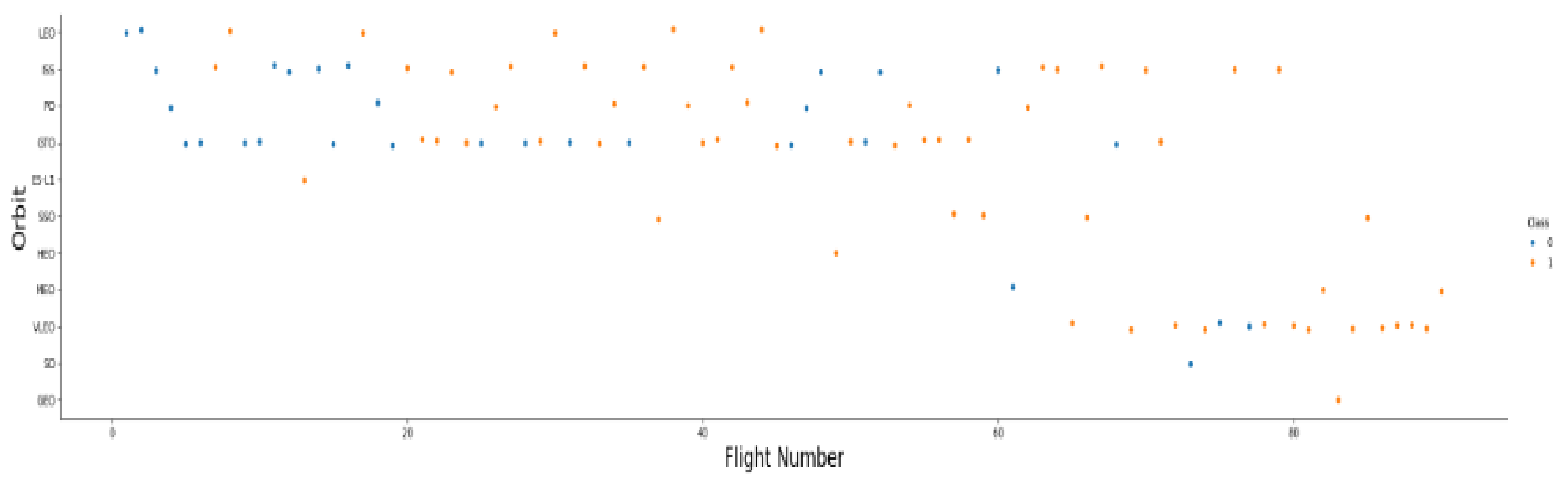
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



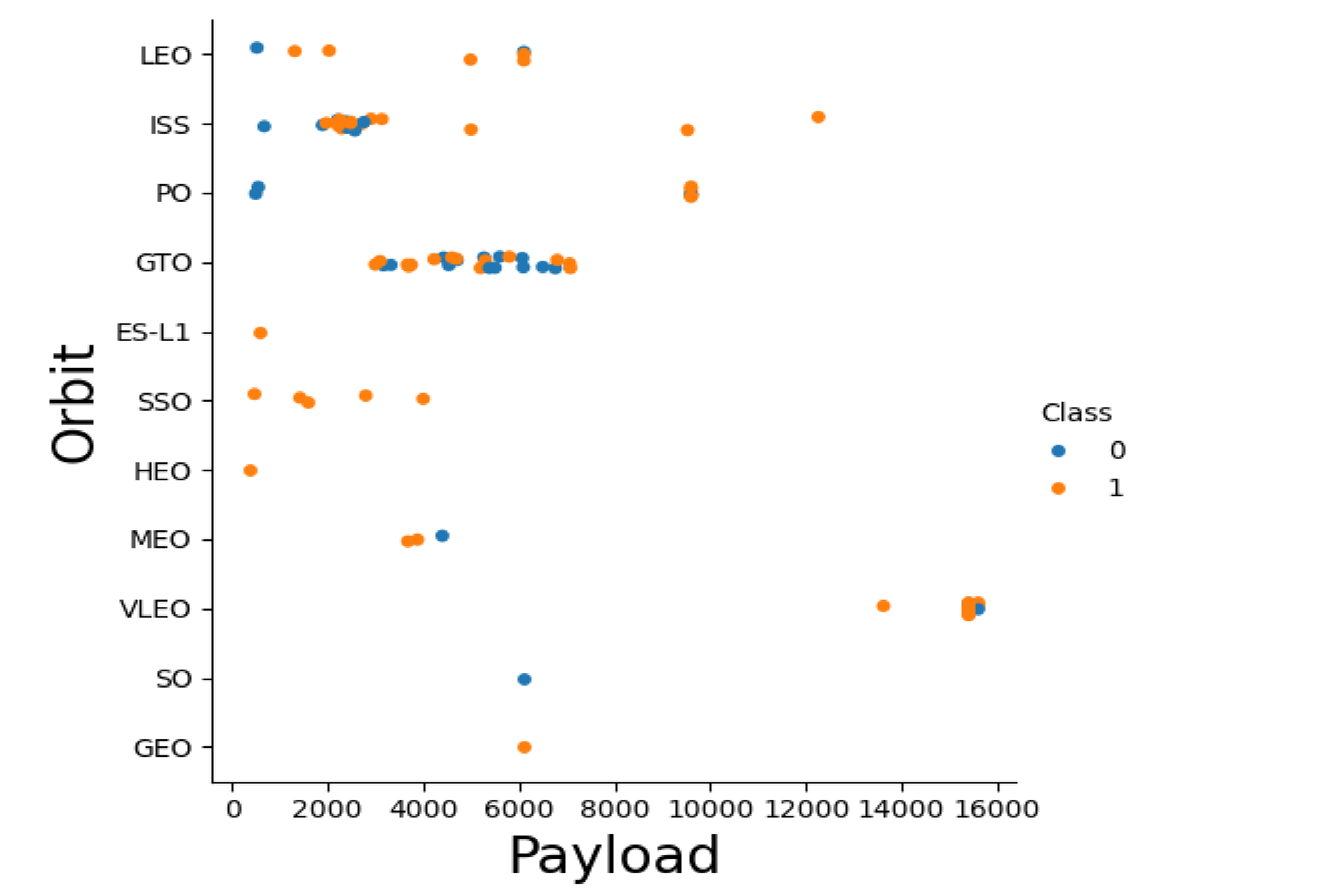
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



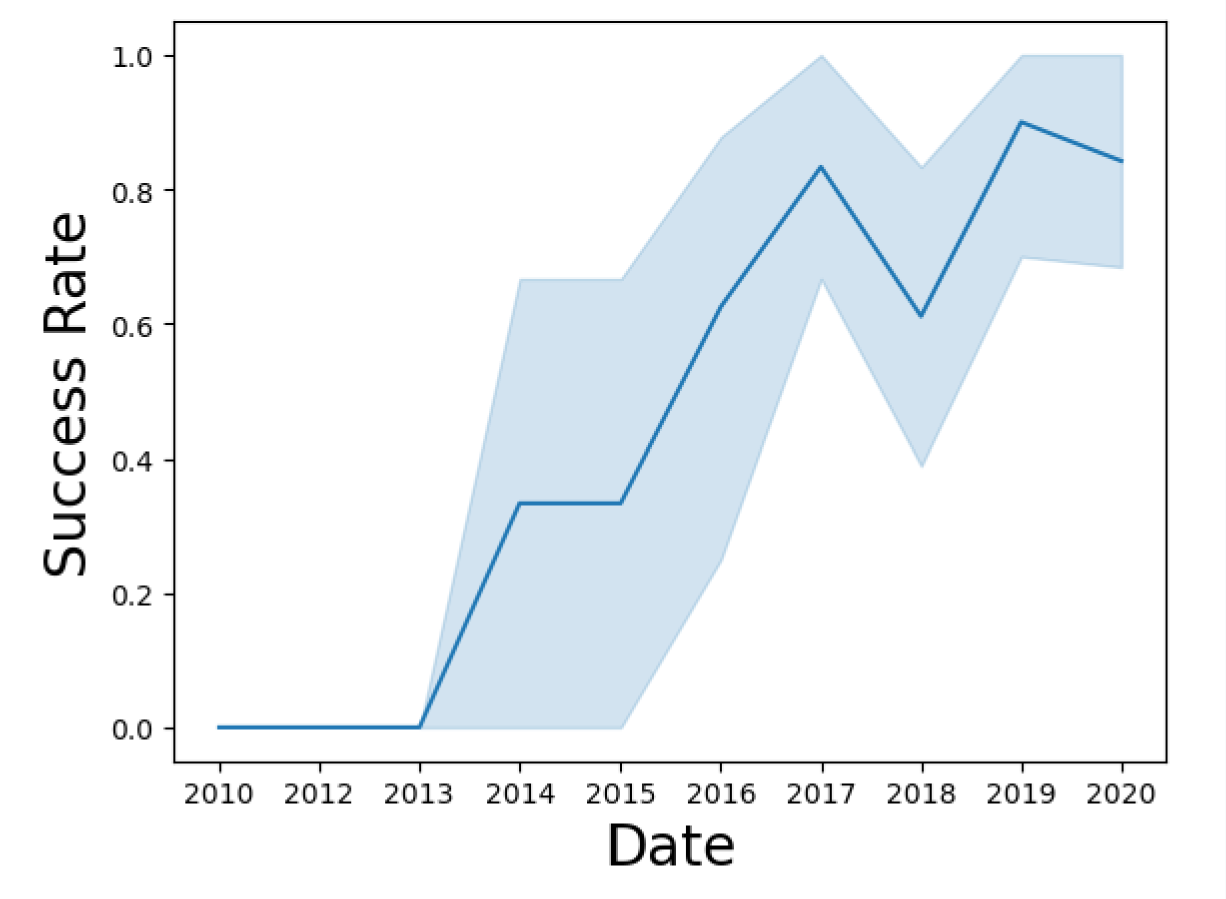
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020



All Launch Site Names

- We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- We used the query below to display 5 records where launch sites begin with `CCA`

```
%sql select * from SPACEXTBL 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_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%sql select Booster_Version from SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' and
PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like '%' to filter for WHERE MissionOutcome was a success or a failure.

```
%sql select count(Mission_Outcome) from SPACEXTBL WHERE Mission_Outcome = 'Success' or Mission_Outcome = 'Failure (in flight)'
```

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

count(Mission_Outcome)
99

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Booster_Version
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

- We 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

```
%sql SELECT SUBSTR(Date,6,2) AS Month, Booster_Version, Launch_site FROM SPACEXTBL WHERE Landing_Outcome
LIKE 'Failure%drone%' AND SUBSTR(Date,0,5) = '2015'
```

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

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- We 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 Numbers FROM SPACEXTBL
WHERE (Landing_Outcome LIKE 'Success%' OR Landing_Outcome LIKE 'Failure%') AND Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome ORDER BY Numbers DESC;
```

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

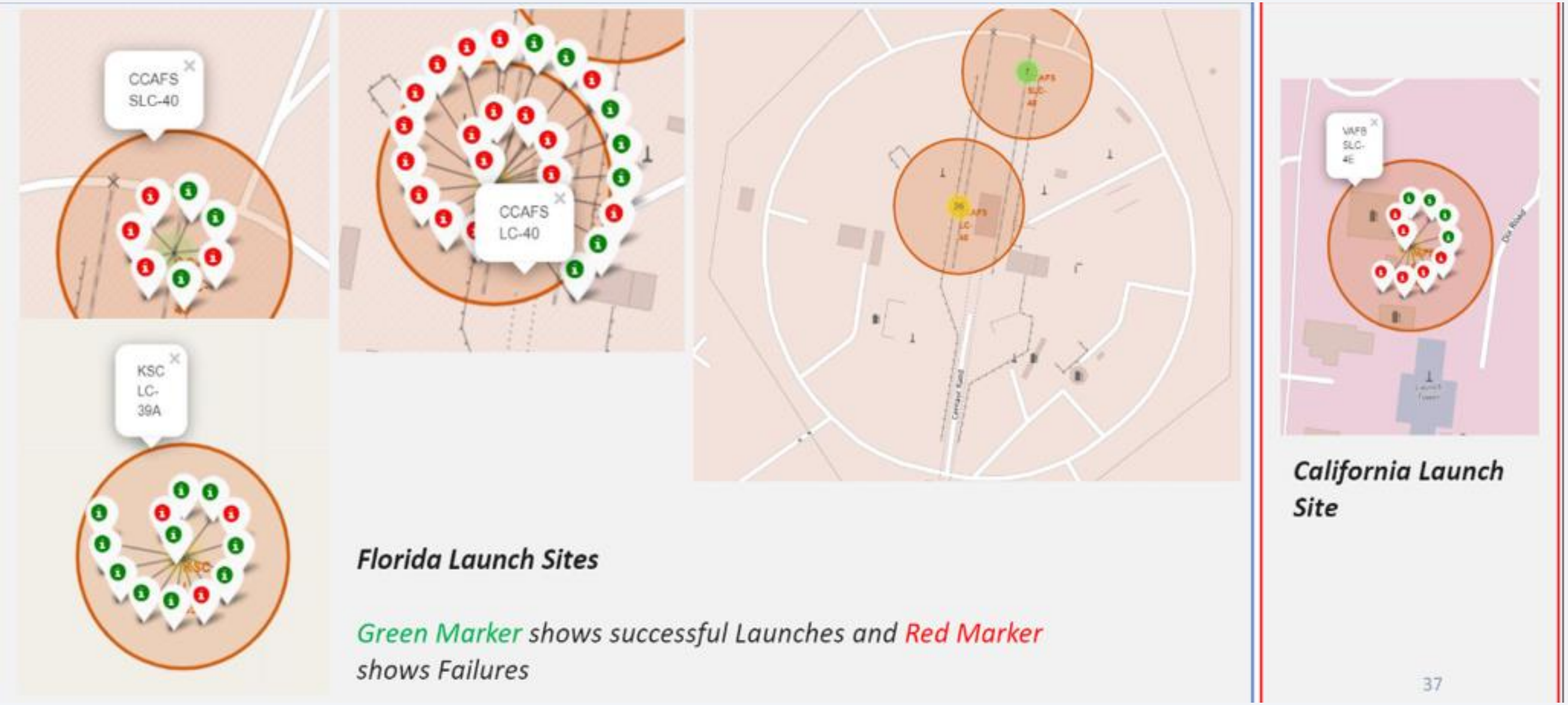
Landing_Outcome	Numbers
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Failure (parachute)	2

Section 3

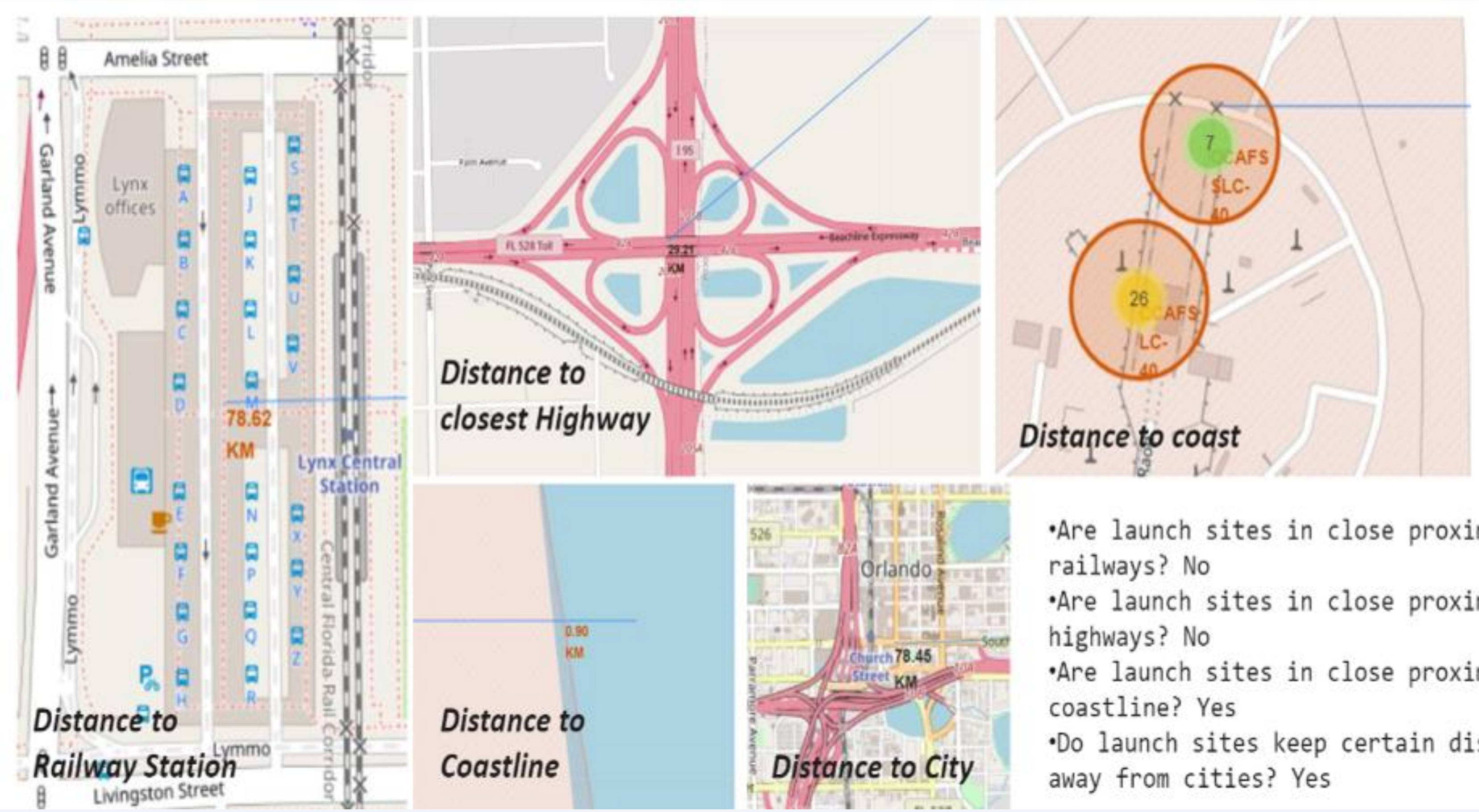
Launch Sites Proximities Analysis

Map of America Showing the launch sites

Successful vs Failure launch sites



Proximity of highway, coastline and railway station to the launch site

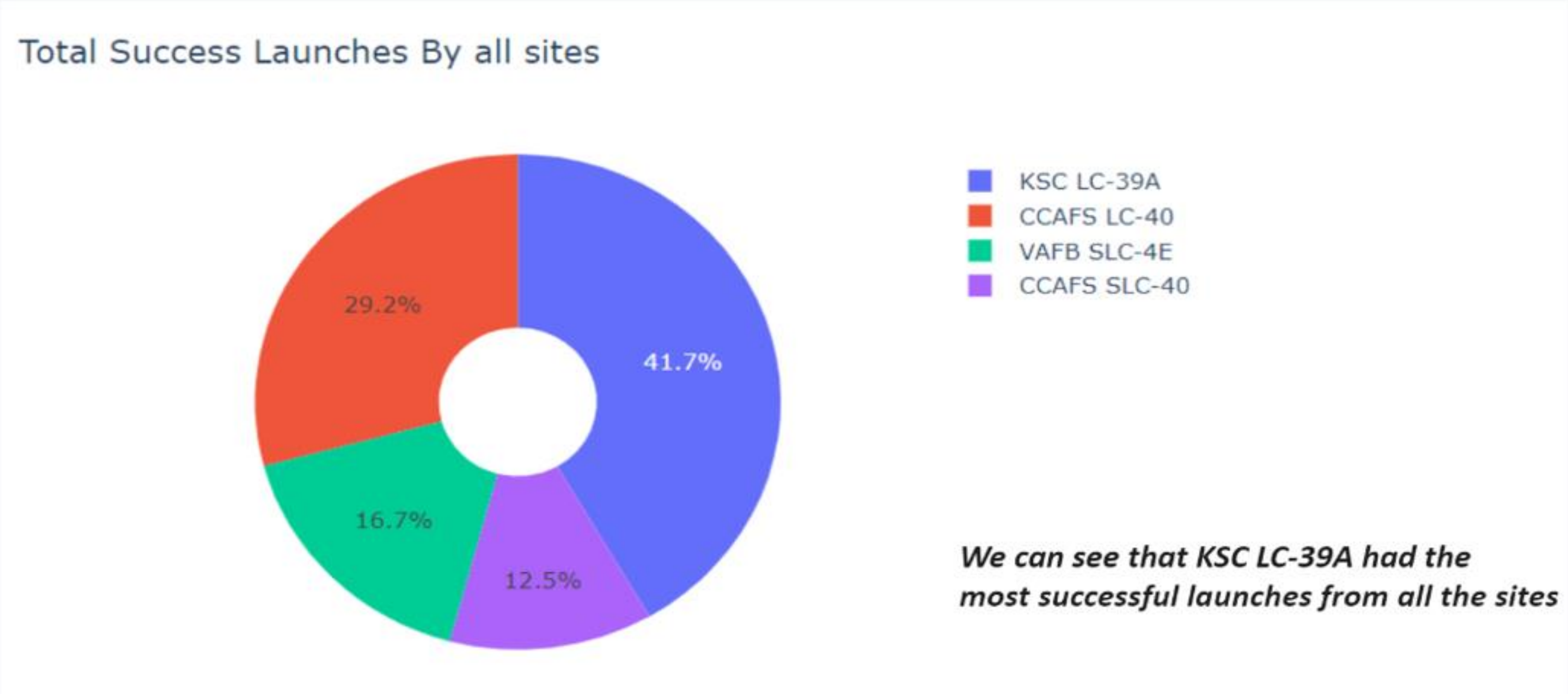


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

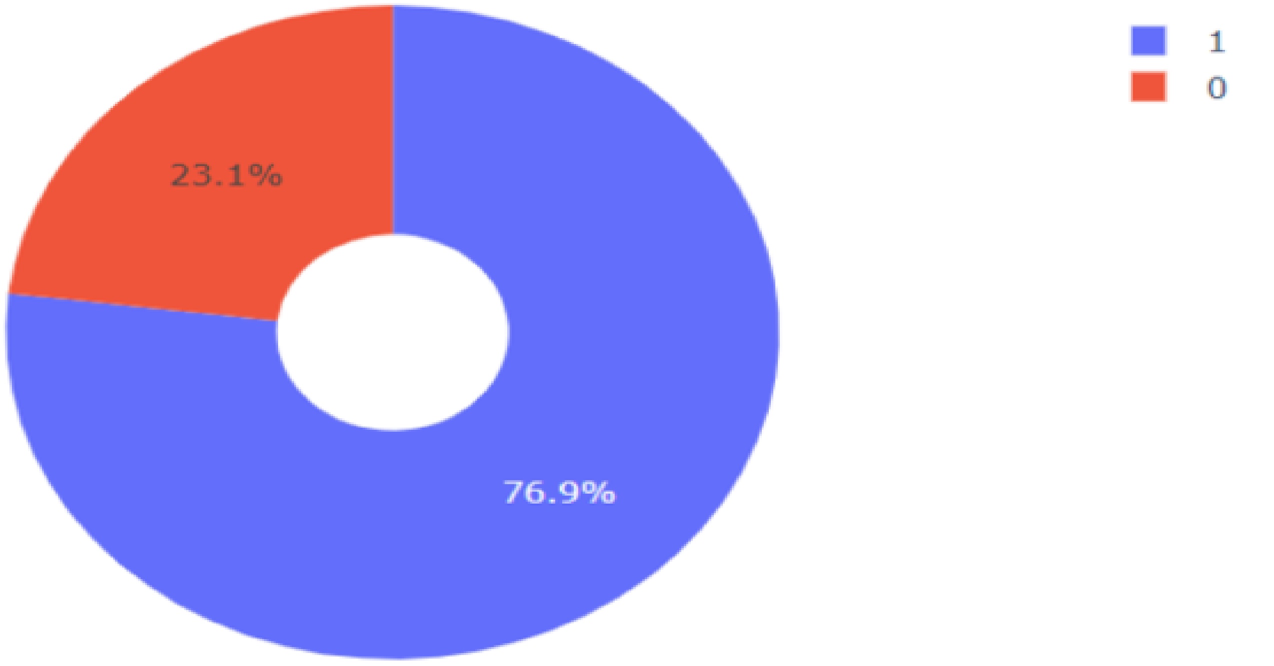
Section 4

Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

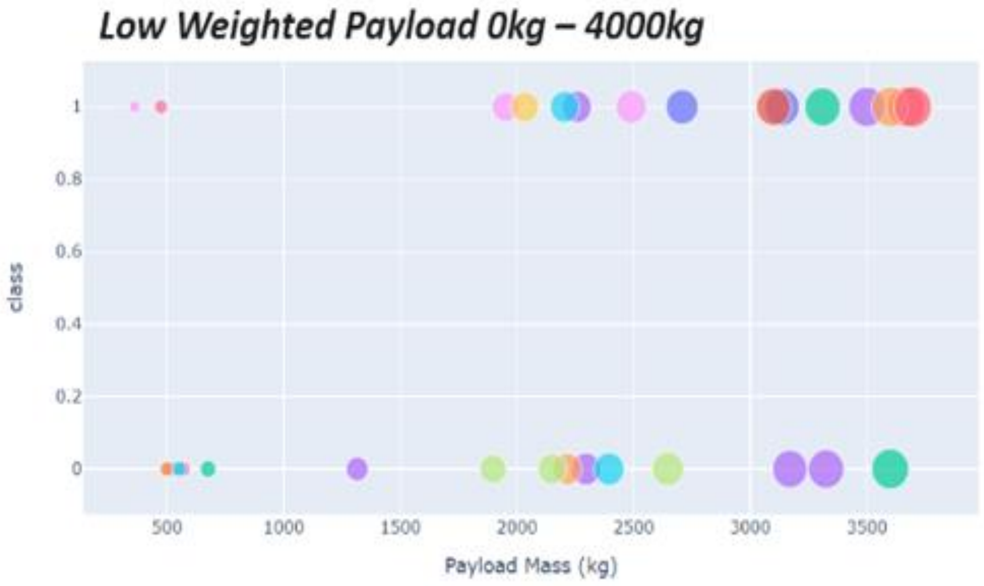


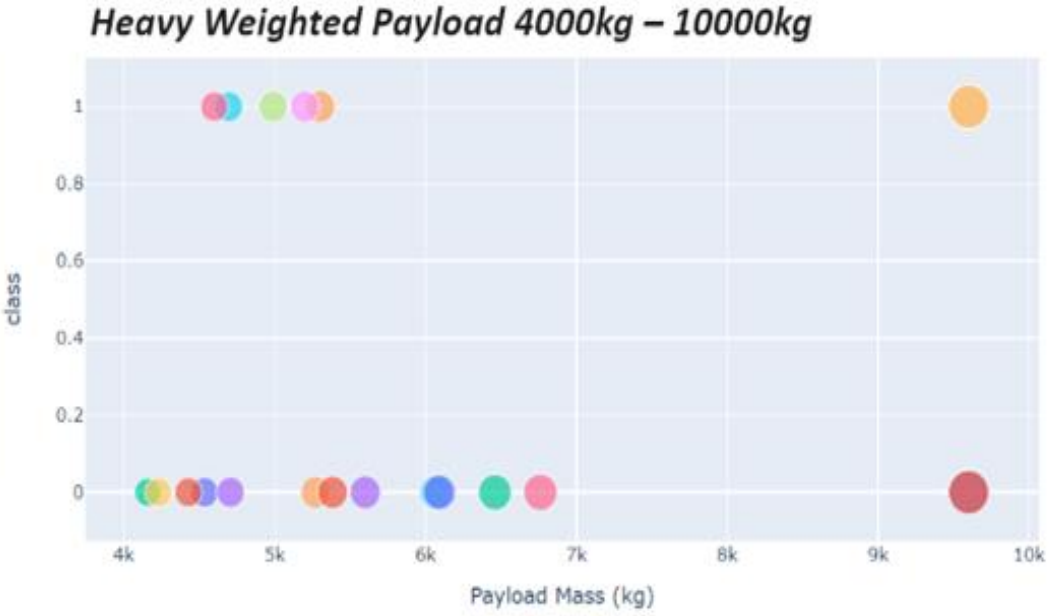
Pie chart showing the Launch site with the highest launch success ratio



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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

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

Find the method performs best:

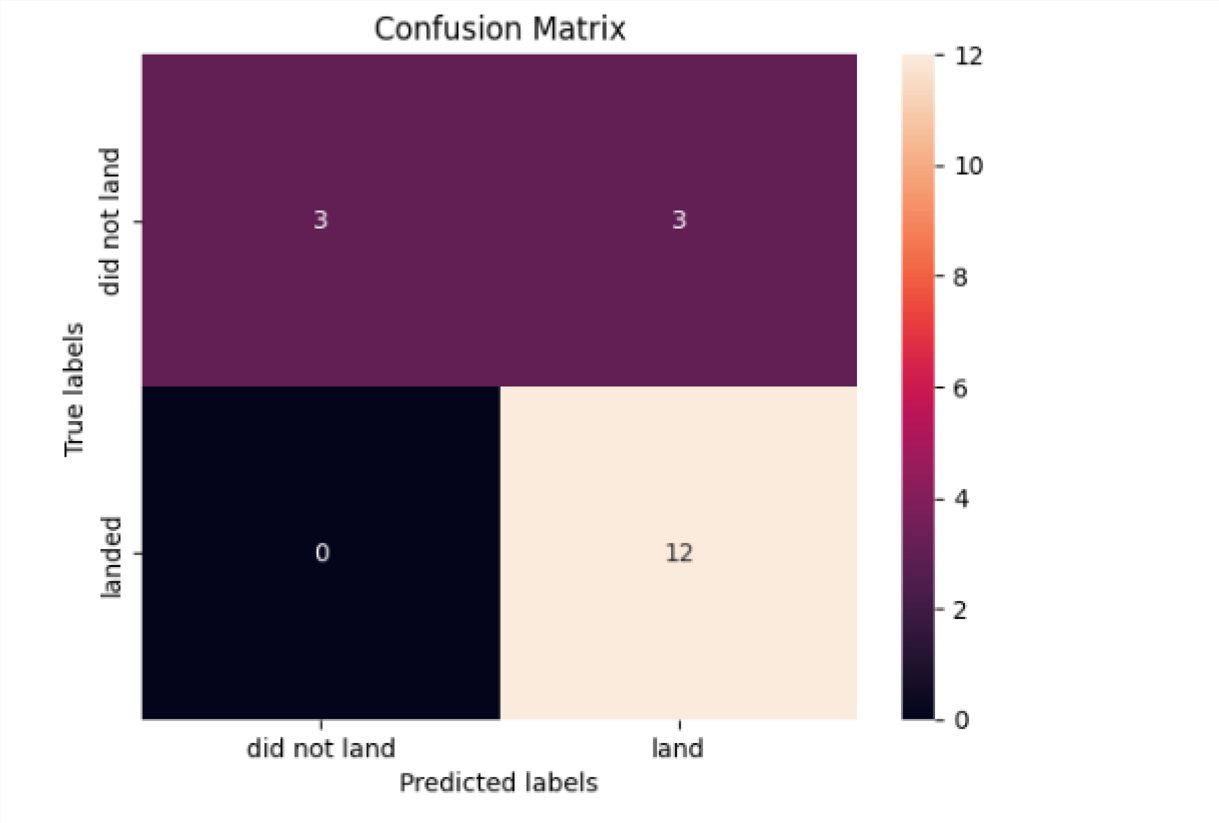
[32]:

```
import pandas as pd

# Assuming knn_cv.best_score_, svm_cv.best_score_, etc., are defined
predictors = {
    'Algorithm': ['KNN', 'SVM', 'Logistics Regression', 'Decision Tree'],
    'Prediction Accuracy': [knn_cv.best_score_, svm_cv.best_score_, logreg_cv.best_score_, tree_cv.best_score_]
}

# Convert dictionary to DataFrame
pred_accuracy = pd.DataFrame(predictors)

# Display the DataFrame
print(pred_accuracy)
```

```
              Algorithm  Prediction Accuracy
0                   KNN             0.848214
1                   SVM             0.848214
2  Logistics Regression             0.846429
3         Decision Tree             0.858929
```

Confusion Matrix

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



Conclusions

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

Thank you!