IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Stephanie Goodman
June 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API
  - Data Collection using Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Visualization
  - Interactive Visual Analytics using Folium
  - Machine Learning Prediction
- Summary of all results
  - Results of Exploratory Data Analysis
  - Demonstration of Analysis Through Visualization
  - Results of Predictive Analysis

Introduction

- Space X: Cost Effective Solutions
  - According to official SpaceX marketing, the Falcon 9 rocket costs approximately $62 million per launch.
  - Compared to other providers Falcon 9 launches are over 90% cheaper than competitors, who average $165 million + per launch.
  - To decrease costs, SpaceX plans to reuse the first stage.
  - Our goal is to determine the cost of a launch by using data science methods to determine whether the first stage of a launch will be a success. A launch is considered successful if the rocket lands in the expected location with no damage.
- Issues to be Addressed:
  - What factors impact whether a launch will be successful?
  - What variables impact the success rate of launches? Which variables affect launches most?
  - What are the ideal conditions for a SpaceX rocket to have the highest probability of success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from the SpaceX Wikipedia page.
  - Data was collected via HTML web scraping through the SpaceX Rest API.
- Perform data wrangling
  - One-hot encoding was applied to categorical features to clean the dataset.
- 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

- Publicly available SpaceX launch data was gathered from the SpaceX REST API
  - The REST API provided launch data that included variables like payload delivered, launch specifications, and outcomes
  - Data was obtained from the API using a .get() request.
  - The response was decoded as a .json() object, and was transformed to a pandas dataframe using the .json_normalize() method
  - Additional data was collected using web scraping methods on the SpaxeX Wikipedia page using the BeautifulSoup Python library.

Data Collection – SpaceX API

- Using a get request, we were able to use the SpaceX API to collect data and complete some basic data cleaning and formatting.

```
In [33]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [34]: response = requests.get(spacex_url)
```

- Link to the GitHub notebook further detailing this process: https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/Data_Collection_API.ipynb

Data Collection - Scraping

- Using BeautifulSoup, we applied web scraping to the Falcon 9 Launches page on Wikipedia to obtain launch records

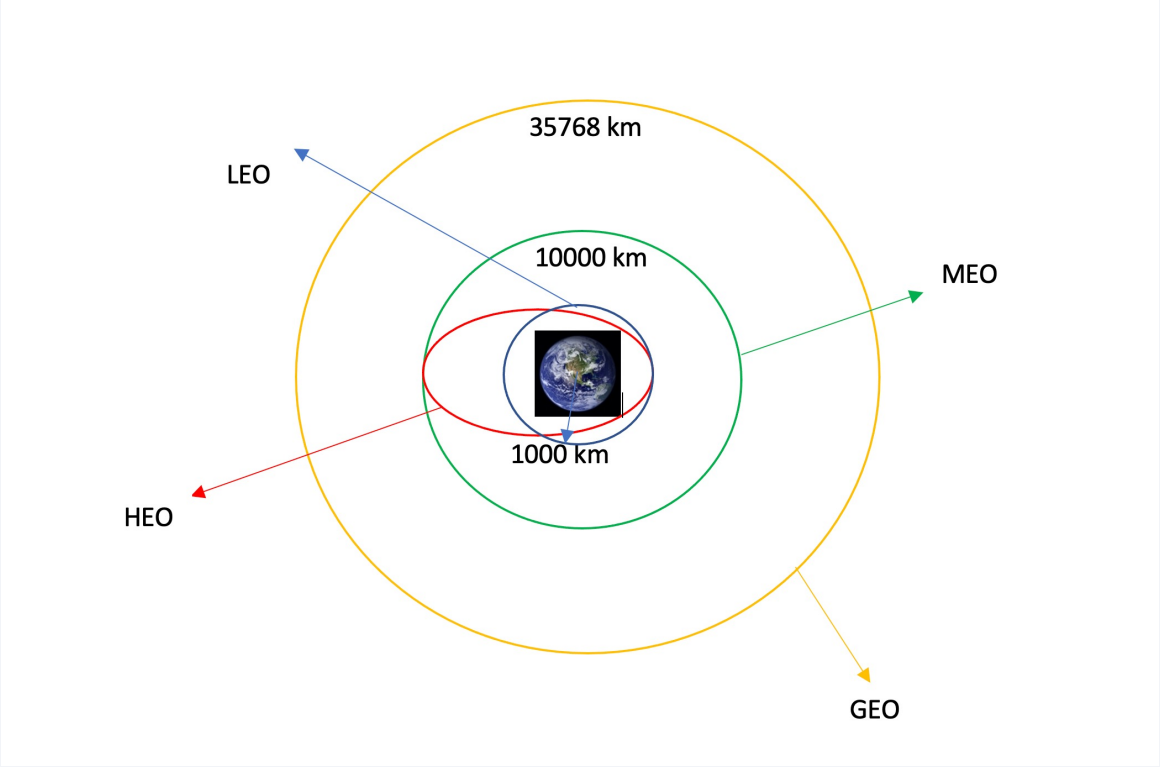
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heav
```

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

- Link to the GitHub notebook further detailing this process: https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/Data_Collection_WebScraping.ipynb

Data Wrangling

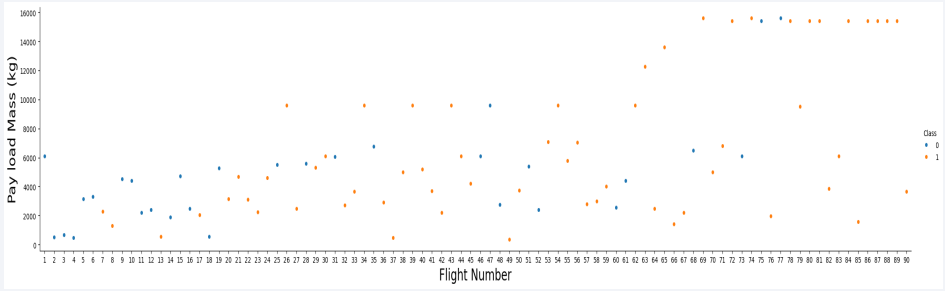
- Launch outcomes varied based on landing location and overall success. To decrease confusion, outcomes were converted to 0 for failure, and 1 for success.
- Calculations were performed to determine number of launches per site, number and occurrence of each orbits. A diagram detailing common orbit types used at SpaceX is included on this slide.
- Results were exported to a .csv file.
- Link to the GitHub notebook further detailing this process: https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/Data_Wrangling.ipynb



EDA with Data Visualization

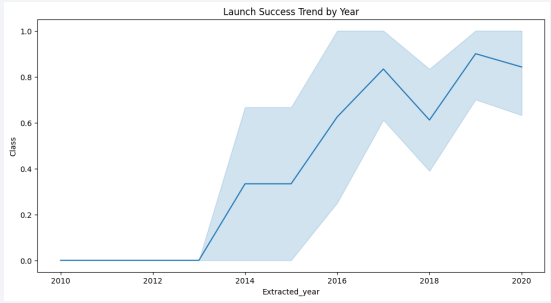
Scatter plots were used to compare different quantitative variables, including

Flight Number vs Payload Mass:



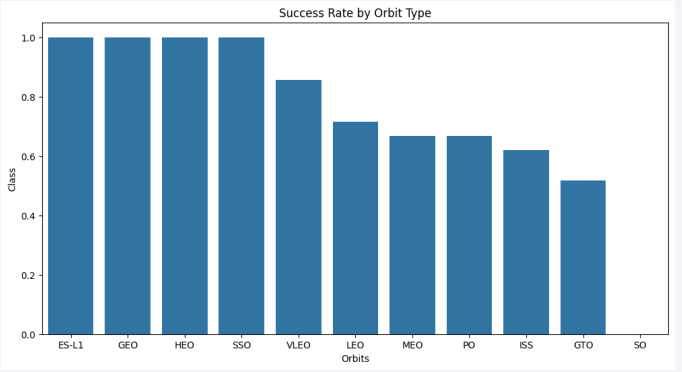
Line Graphs were used to display variables and trends over time:

Success Rate by Year



Bar graphs were used to compare categories and discrete values, such as:

The relationship between Success Rate and Orbit Type



Link to the GitHub notebook further detailing this process:

https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/EDA_with_Visualization.ipynb

EDA with SQL

- Using SQLAlchemy, queries were performed on the dataset to create tables that provide further context for launch sites, site outcomes, and larger generalizations.

In [25]:
```
%%sql
    SELECT MIN(Date) AS first_landing_success
    FROM SPACEXTABLE
    WHERE Landing_Outcome LIKE '%Success%';
```

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

Out[25]:

first_landing_success
2015-12-22

- Link to the notebook on GitHub further detailing this process: https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/EDA_With_SQL.ipynb

Build an Interactive Map with Folium

- Launch data was incorporated into an interactive map with Folium through this process:
  - Latitude and Longitude coordinates were taken at each launch site
  - A circle marker was added around each launch site with a label for each launch site name.
  - Using Green and Red markers, launch outcomeswere color coordinated (0 = Red, unsuccessful, 1 = Green, successful).
  - The distance between a launch site and other varied landmarks was calculated using Haversine's Formula. This was done so to determine various trends regarding launch site location.
    - Lines were drawn on the map to measure distance from landmarks
- Link to the notebook on GitHub further detailing this process:

https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/Interactive_Visual_Analysis_with_Folium.ipynb

Build a Dashboard with Plotly Dash

- Using Plotly Dash, an interactive dashboard was created.
  - This dashboard was hosted on a local URL for ease of access.
  - Features provided for interactivity included dropdown menus to select year or launch site.
  - Pie charts displayed total launches by specific sites
  - Scatter graphs displayed the relationship between Payload Mass (Kg) and Outcomes based on specific booster versions
- Link to the notebook on GitHub further detailing this process: https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/Interactive_Dashboard.py

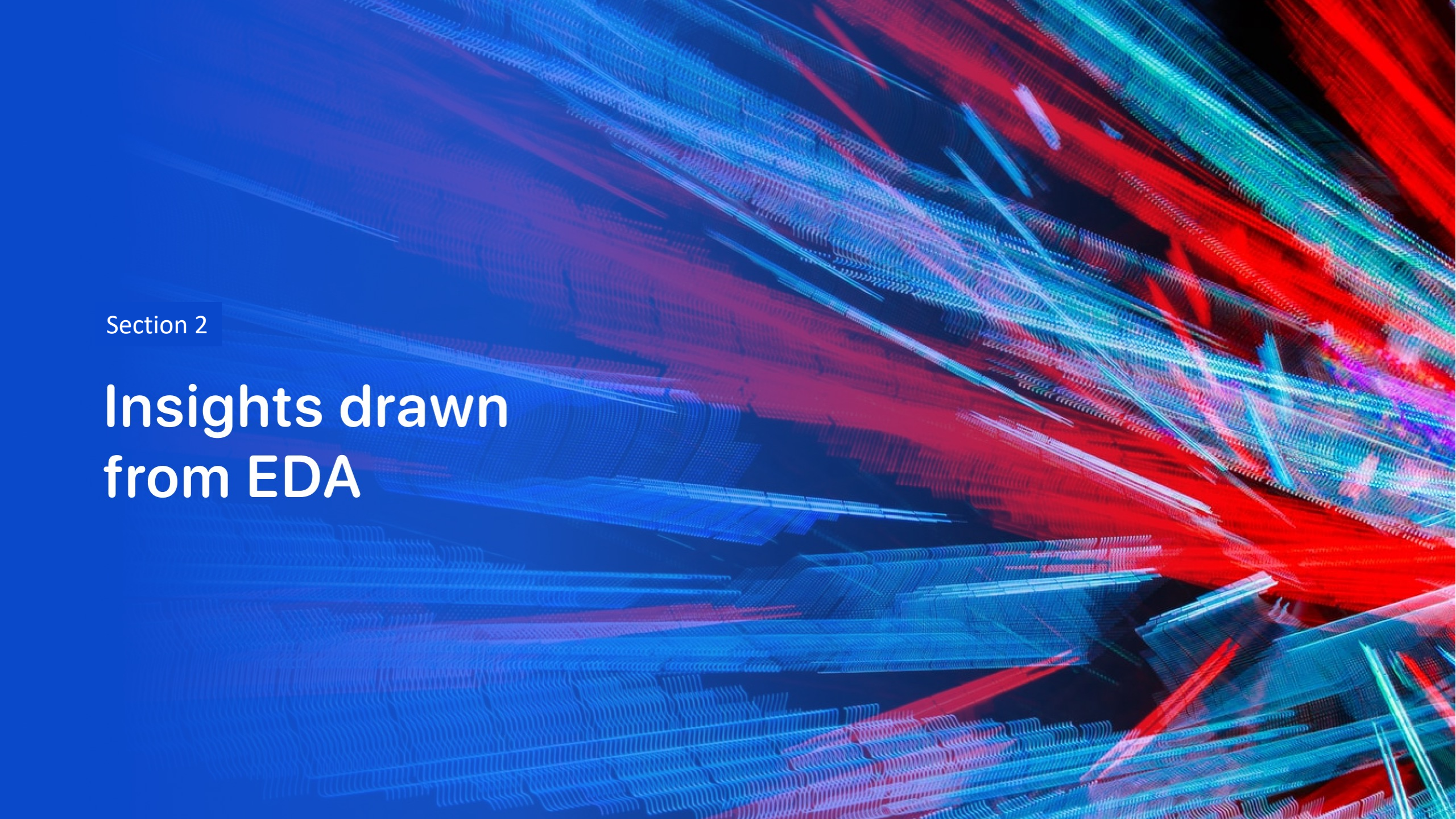
Predictive Analysis (Classification)

- Building the Model: data was loaded, transformed, split into training and test datasets, machine learning algorithms were chosen, parameters were set, and datasets were fit into GridSearchCV objects to train the dataset.
- Model Evaluation: Accuracy metrics were reviewed, best hyperparameters were determined, and a Confusion Matrix was plotted for each model
- Model Improvement: The model was improved using feature engineering and algorithm tuning
- The best performing classification model was defined as the model with the best accuracy score.
- Link to the notebook on GitHub further detailing this process:

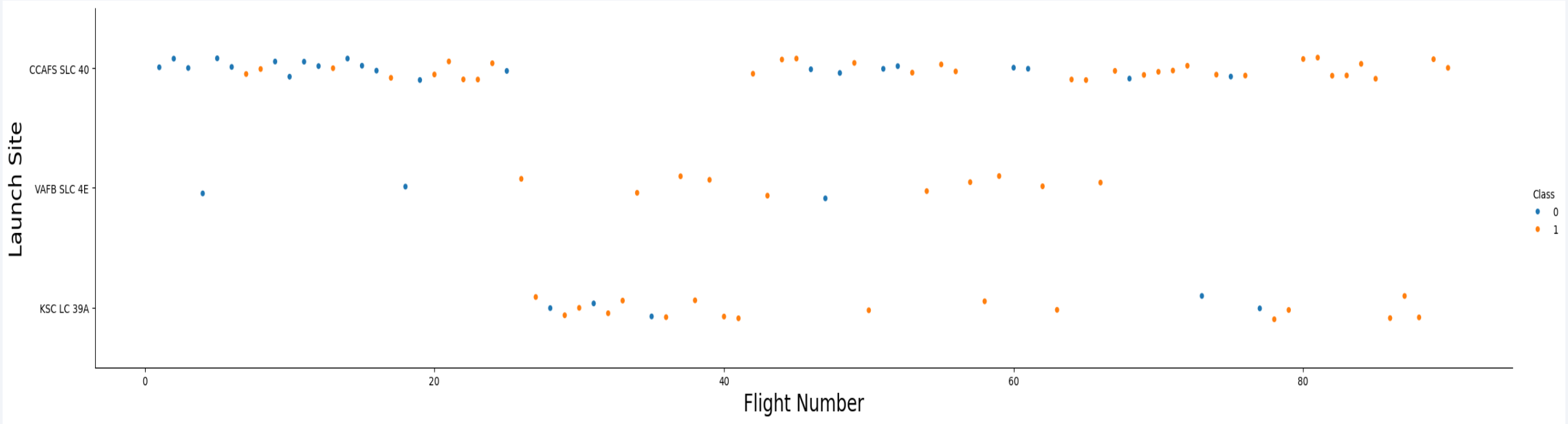
https://github.com/stephaniegoodman/spacex_flightanalysis/blob/e80658124600ae5d3bc37c6c6e45107e9f5044ba/Interactive_Dashboard.py

Results

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

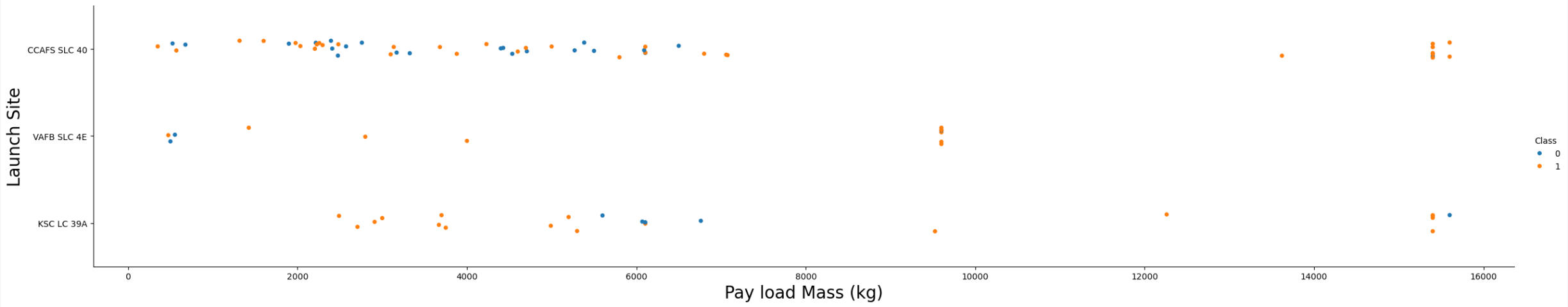

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



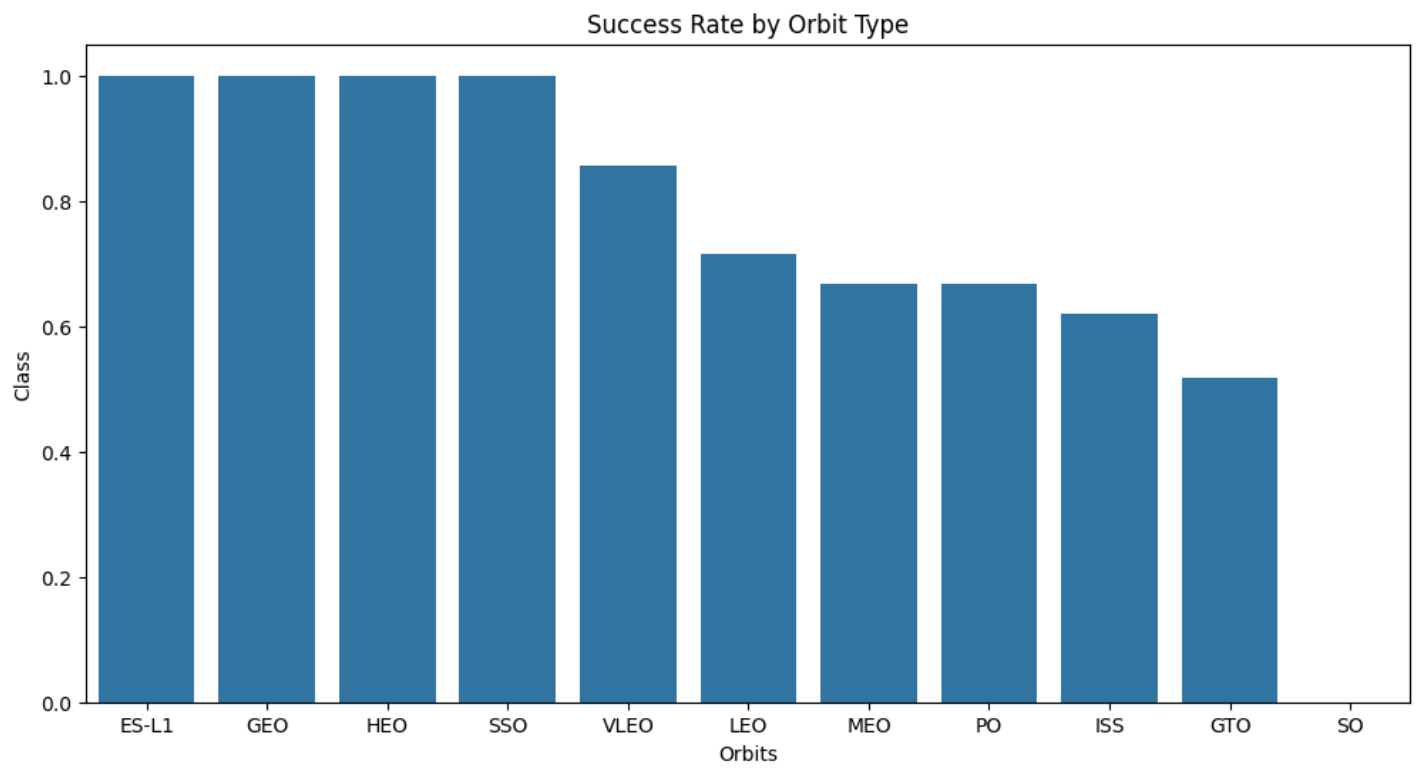
Higher numbers of flights are correlated with higher success rates.

Payload vs. Launch Site



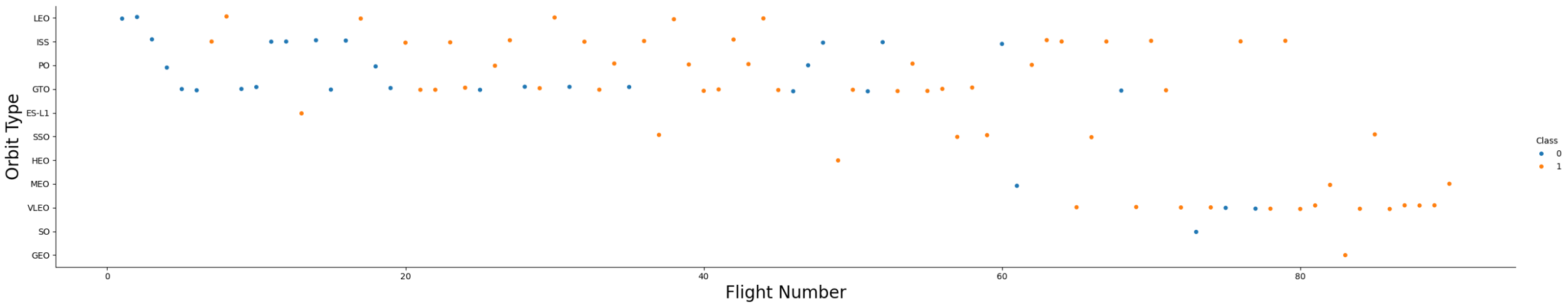
For launch site CCAFS SLC 40, higher payload mass is associated with higher rocket success rate. There are not clear enough patterns among the other two launch sites to suggest a relationship between launch site and payload mass.

Success Rate vs. Orbit Type



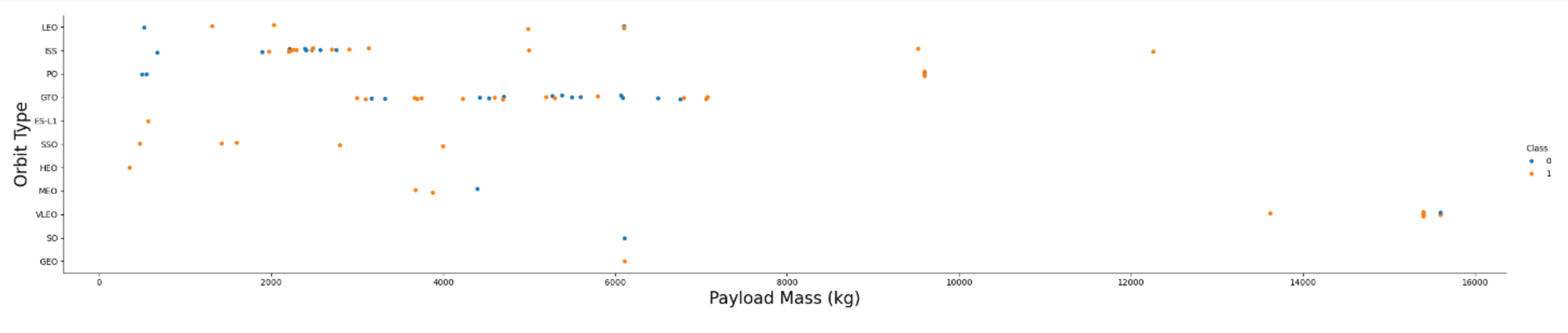
Orbit types ES-L1, GEO, HEGO, and SSO have the highest success rates of all orbit types.

Flight Number vs. Orbit Type



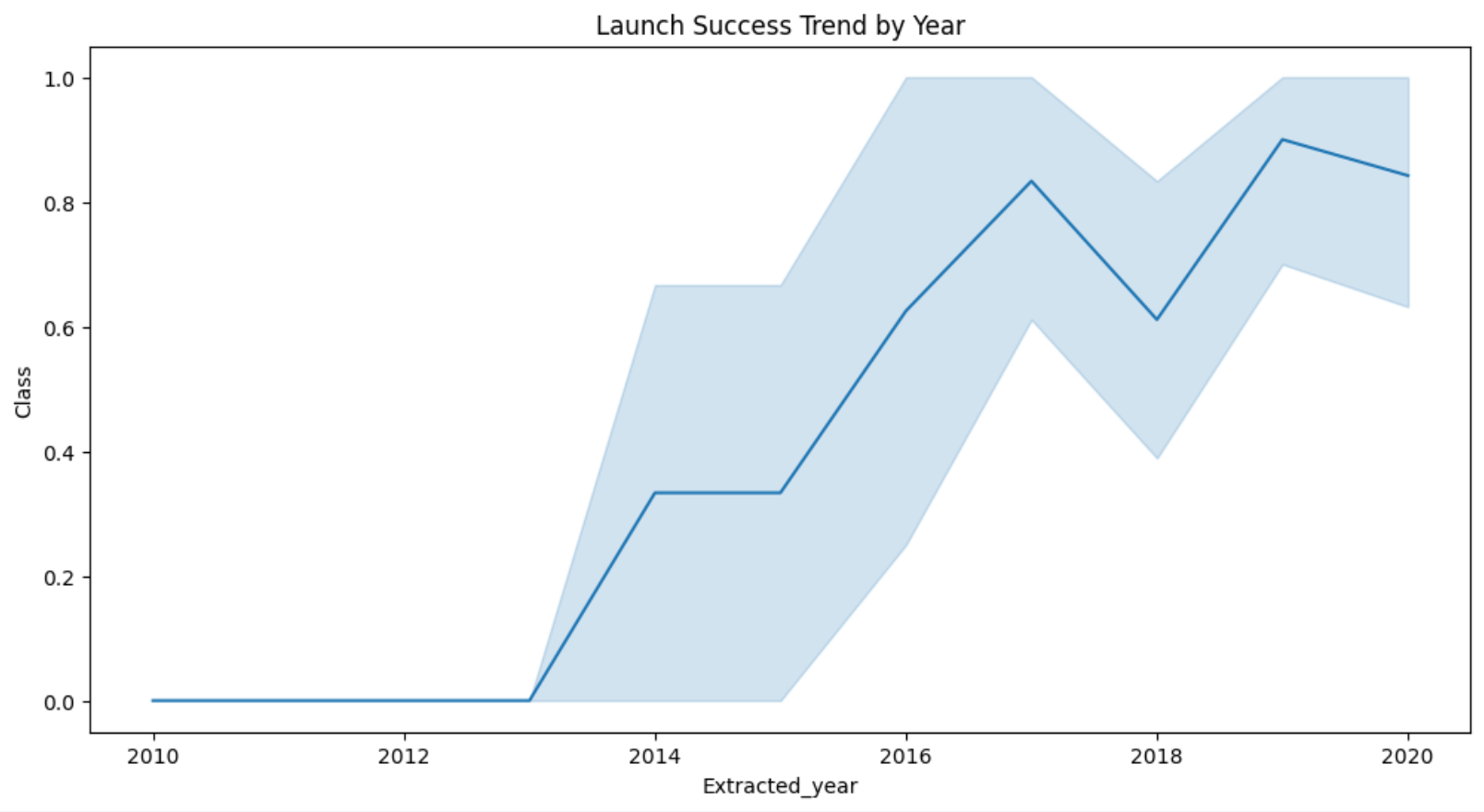
In the LEO orbit, success is correlated with flight number. For other orbit types, there is no clear relationship between flight number and orbit type.

Payload vs. Orbit Type



Landings are more likely to be successful with heavy payloads for PO, LEO, and ISS orbits.

Launch Success Yearly Trend



Success rate increased from 2013-2020, with one notable peak in 2016.

All Launch Site Names

- Using a SQL query including a DISTINCT operator in the SELECT clause resulted in a list of all unique launch site names

```
In [14]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

Out[14]:

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

Launch Site Names Begin with 'CCA'

- Using the LIKE operator in the WHERE clause, as well as a LIMIT clause in this SQL query returned 5 records where launch sites begin with ‘CCA’

```
In [15]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5
 * sqlite:///my_data1.db
Done.
```

Out[15]:

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

- The aggregate function SUM() combined with a WHERE clause specifying customer type as NASA returns the total payload carried by boosters from NASA

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

In [19]:
```
%%sql
    SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass
    FROM SPACEXTABLE
    WHERE Customer = 'NASA (CRS)';
```

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

Out[19]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- The aggregate function AVG() combined with a WHERE clause specifying booster version returns the average payload mass carried by booster version F9 v1.1

In [23]:

```
%%sql
    SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass
    FROM SPACEXTABLE
    WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

Out[23]:

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

- The use of the MIN() function combined with the LIKE operator in the WHERE clause allows us to determine the date of the first successful landing

In [25]:

```
%%sql
    SELECT MIN(Date) AS first_landing_success
    FROM SPACEXTABLE
    WHERE Landing_Outcome LIKE '%Success%';
```

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

Out[25]:

first_landing_success
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The use of inequality symbols < > and a specified WHERE clause returns the list of successful booster versions where the payload is between 4000 and 6000 kilograms

In [33]:
```
%%sql
    SELECT Booster_Version
    FROM SPACEXTABLE
    WHERE Landing_Outcome = 'Success (drone ship)'
        AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

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

Out[33]:

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failed Mission Outcomes

- The use of the COUNT() function and GROUP BY clause returns the total number of successful and failed mission outcomes.

In [35]:

```
%%sql
    SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS total_of_outcome
    FROM SPACEXTABLE
    GROUP BY MISSION_OUTCOME;
```

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

Out[35]:

Mission_Outcome	total_of_outcome
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The combination of the DISTINCT operator in the SELECT clause, and the use of a subquery in the WHERE clause returns the list of unique boosters with the maximum payload in the entire dataset.

In [38]:
```
%%sql
    SELECT DISTINCT Booster_Version
    FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ =
        (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

Out[38]:

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

- The use of the substr() function allows for a result that includes important launch data in the year 2015.

In [41]:

```
%%sql
    SELECT Landing_Outcome, Booster_Version, Launch_Site,
    FROM SPACEXTABLE
    WHERE Landing_Outcome = 'Failure (drone ship)'
        AND substr(Date,0,5)='2015';
```

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

Out[41]:

Landing_Outcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The combination of the aggregate COUNT() function, the WHERE, GROUP BY, and ORDER BY clauses returns a ranked lists of landing outcomes between June 4th 2010 and March 20th 2017.

In [39]:

```
%%sql
    SELECT Landing_Outcome, COUNT(Landing_Outcome) AS total_landings
    FROM SPACEXTABLE
    WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY total_landings DESC
```

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

Out[39]:

Landing_Outcome	total_landings
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

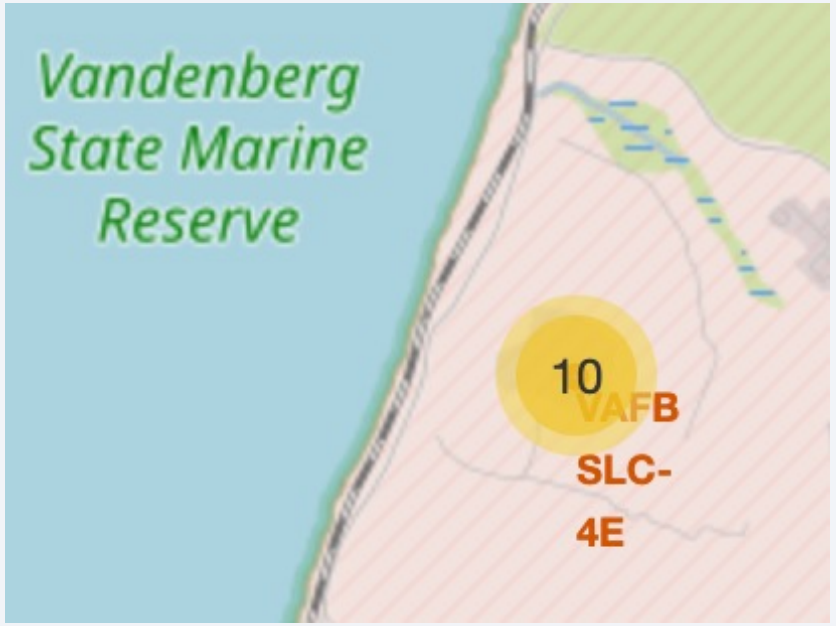

Section 3
Launch Sites
Proximities Analysis

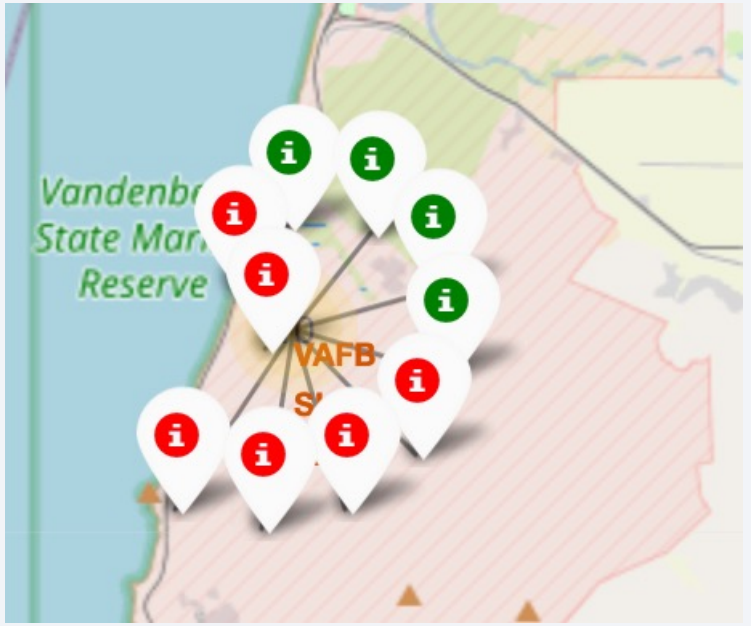
Global Map Markers for All Launch Sites

- The two primary launch sites are on opposite coasts of the United States.

Color Coded Launch Outcomes

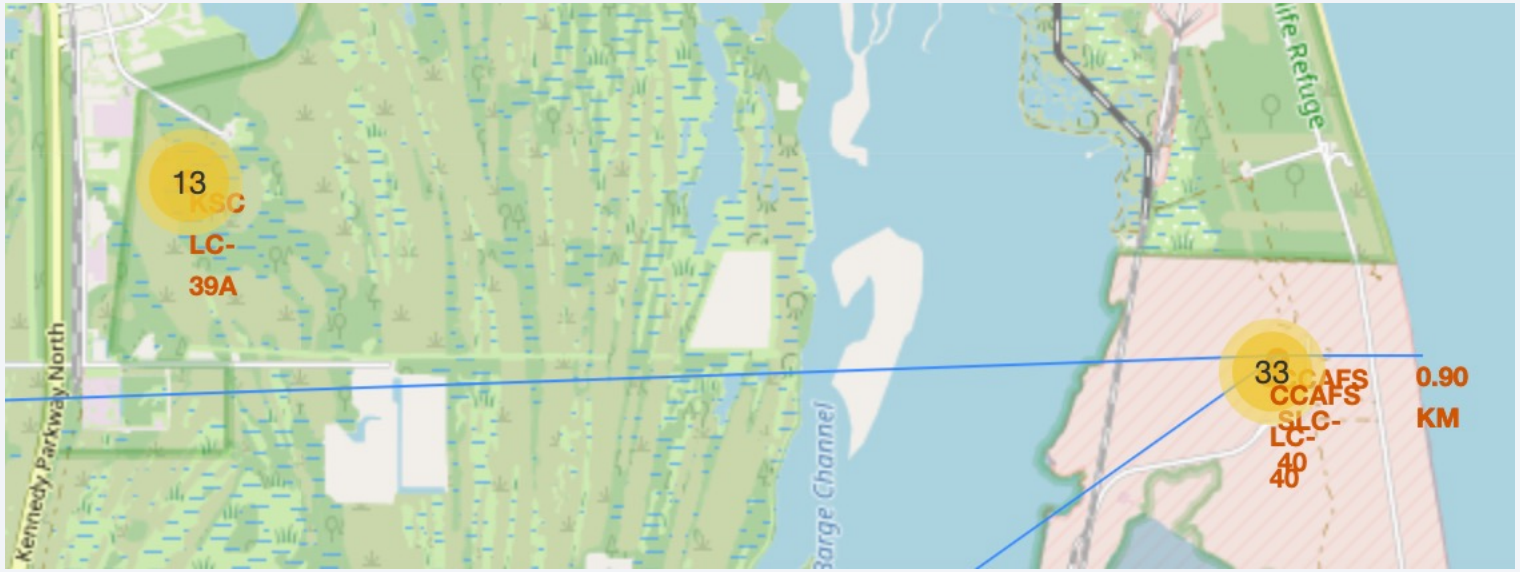
- Using the interactivity provided by Folium maps, we can navigate to specific launch sites and view color coded outcomes. Successful landings are in green, unsuccessful landings are in red.

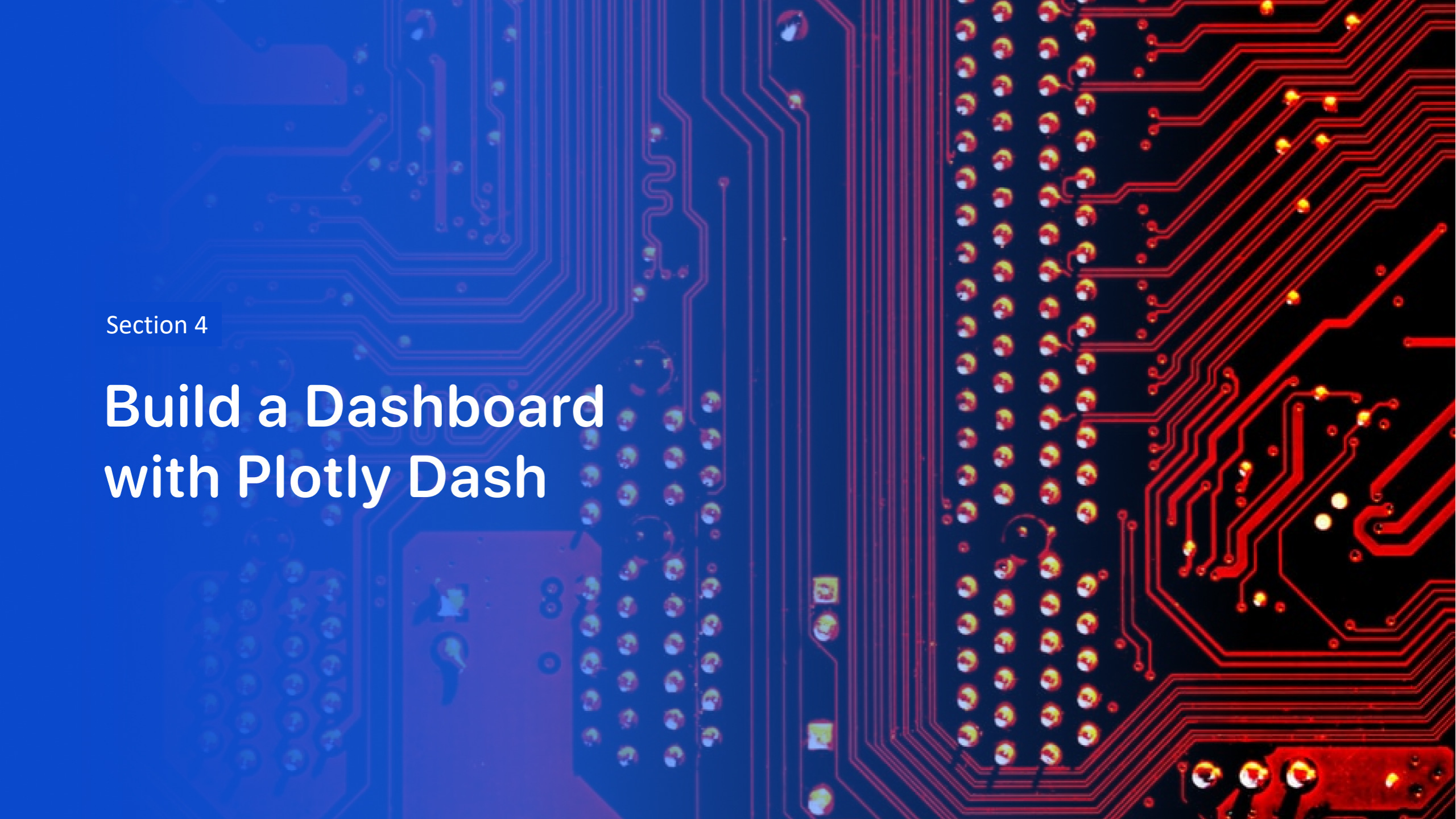




Launch Site Proximity to Notable Areas

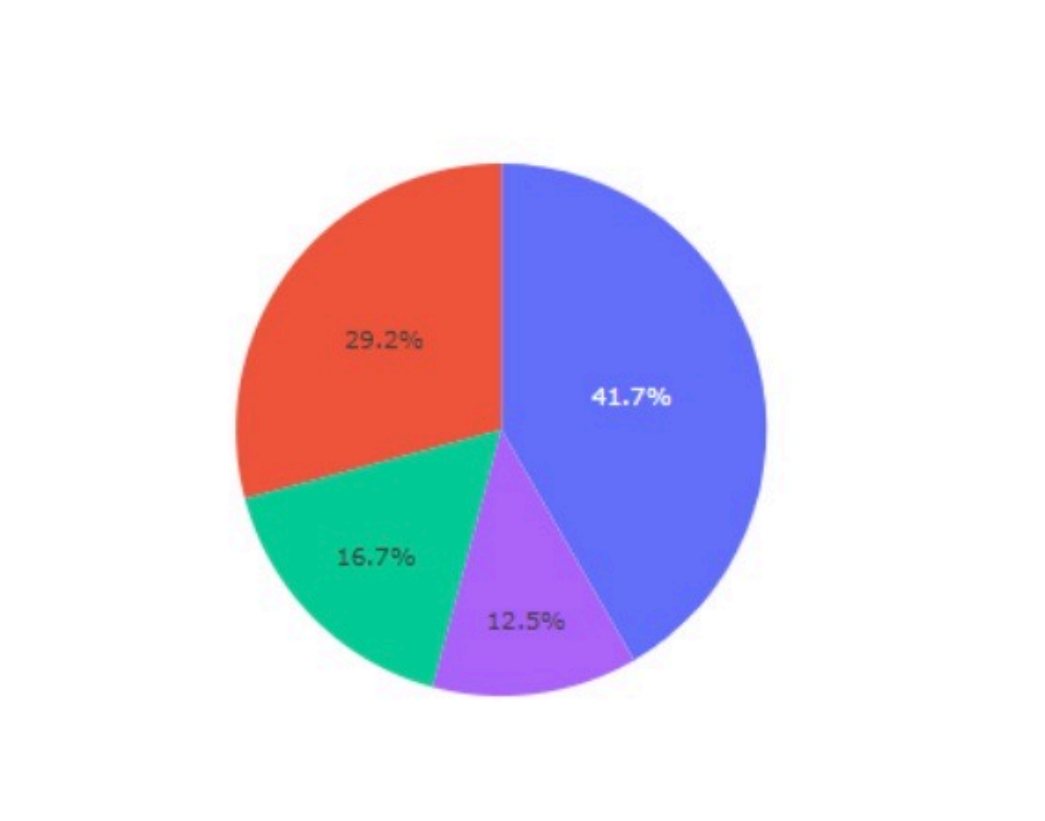
- This map utilizes distance lines, markers, and distance circles to display the relative proximity to the nearest highways, cities, railways, and coastlines to each respective launch site.




Section 4
Build a Dashboard with Plotly Dash

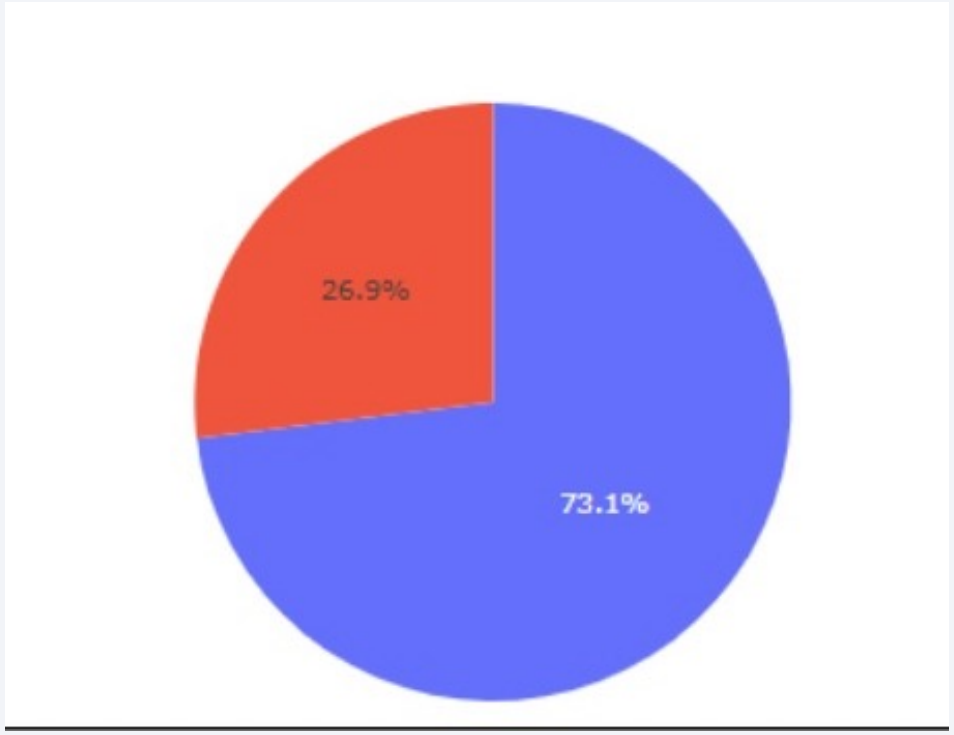
Success by Launch Sites

- This pie chart details the percentage of successful launches by launch site
- KSC LC-39A was the most successful launch site.



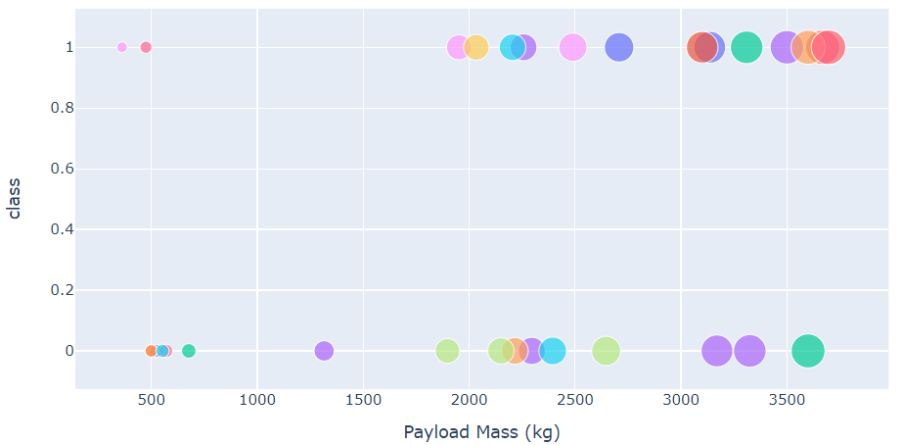
Launch Site KSC LC-39A Success Ratio

- KSC LC-39A had the highest success ratio (1 = successful launch, 0 = unsuccessful launch) at 72.1%



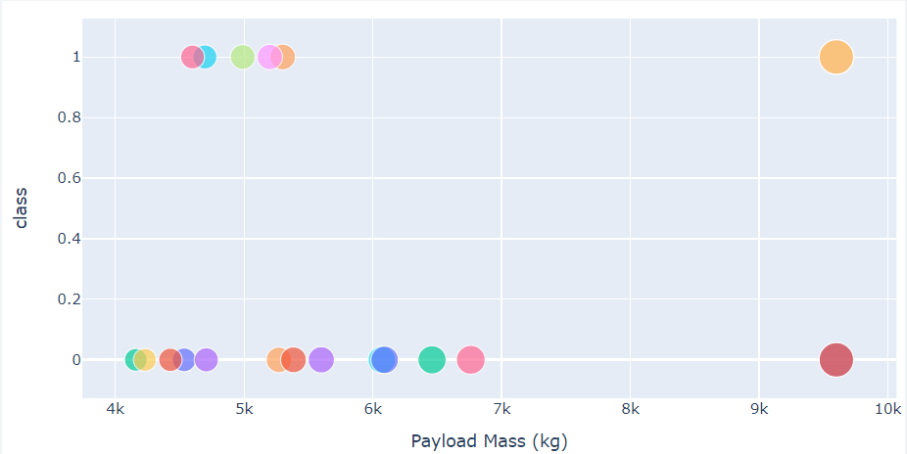
Payload vs. Launch Outcome

- Using the slider function in the Plotly Dashboard, we can investigate the relationship on payload vs launch outcome.
- As displayed below, the success rate is higher when a payload weight is lower; a negative relationship



Low Payload Weight

(0-40kg)



High Payload Weight

(4k-10k kg)

Section 5

Predictive Analysis (Classification)

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_)
```

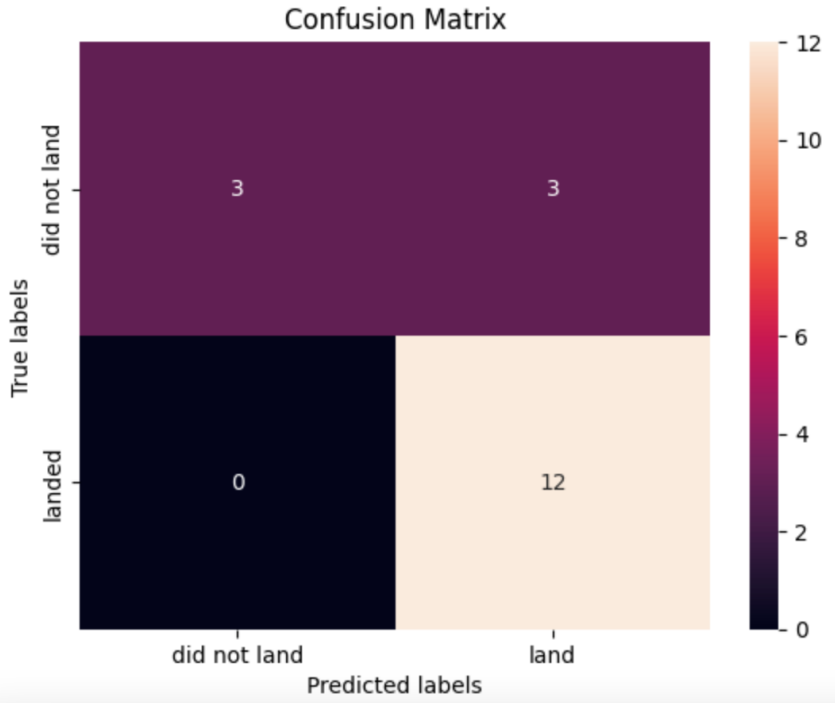
```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

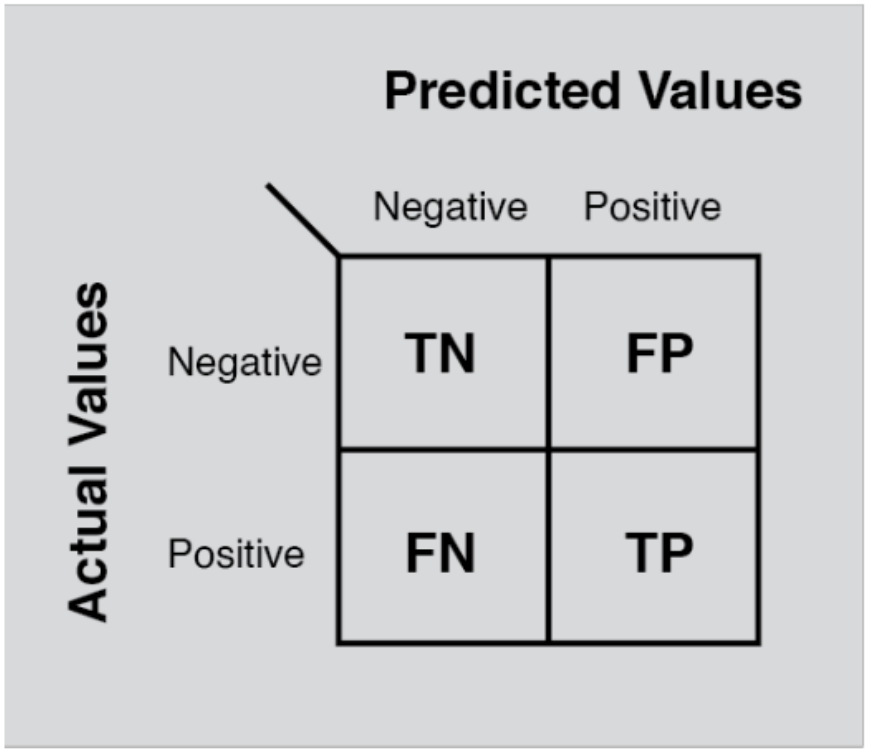
- When comparing all models against each other, the DecisionTree model has the highest accuracy, with a score of 0.87.

Decision Tree Confusion Matrix

- In the case of the Decision Tree model, the biggest area of concern is Type I errors, also known as false positives. This is when a successful landing is predicted, but the actual landing is unsuccessful.

```
In [30]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```





Conclusions

Given the analysis performed and the development of predictive machine learning models, we can conclude that:

- Success rates have improved in the years 2013-2020
- The most successful orbits are ES-L1, GEO, HEO, SSO, VLEO
- Smaller payloads have more successful launches
- Higher flight numbers have more successful launches
- KSC LC-39A is the most successful launch site, with a success ratio of 72.1%
- The Decision Tree model is the most effective to predict future launch success

Appendix

- Here is the link to the GitHub repository with all completed notebooks, code files, and this presentation in pdf format: https://github.com/stephaniegoodman/spacex_flightanalysis

Thank you!