Winning Space Race with Data Science

Sidney Thollon
Feb-24

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API and Web Scraping
  - Data Wrangling
  - EDA with SQL
  - EDA 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

- Project background and context
  - In this project, we predict if the Falcon 9 first stage will land successfully. 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. The goal of the project is to create a machine learning algorithm to predict the booster landing success probability of a launch.
- Problems you want to find answers
  - Assess the impact of payload mass, launch site selection, frequency of flights, and chosen orbits on the success rate of first-stage landings.
  - Analyze successful landings over time to discern any patterns or trends that have emerged.
  - Determine the most effective predictive model for binary classification of landing success to enhance decision-making and future launch planning.

Section 1

Methodology

Methodology

Executive Summary

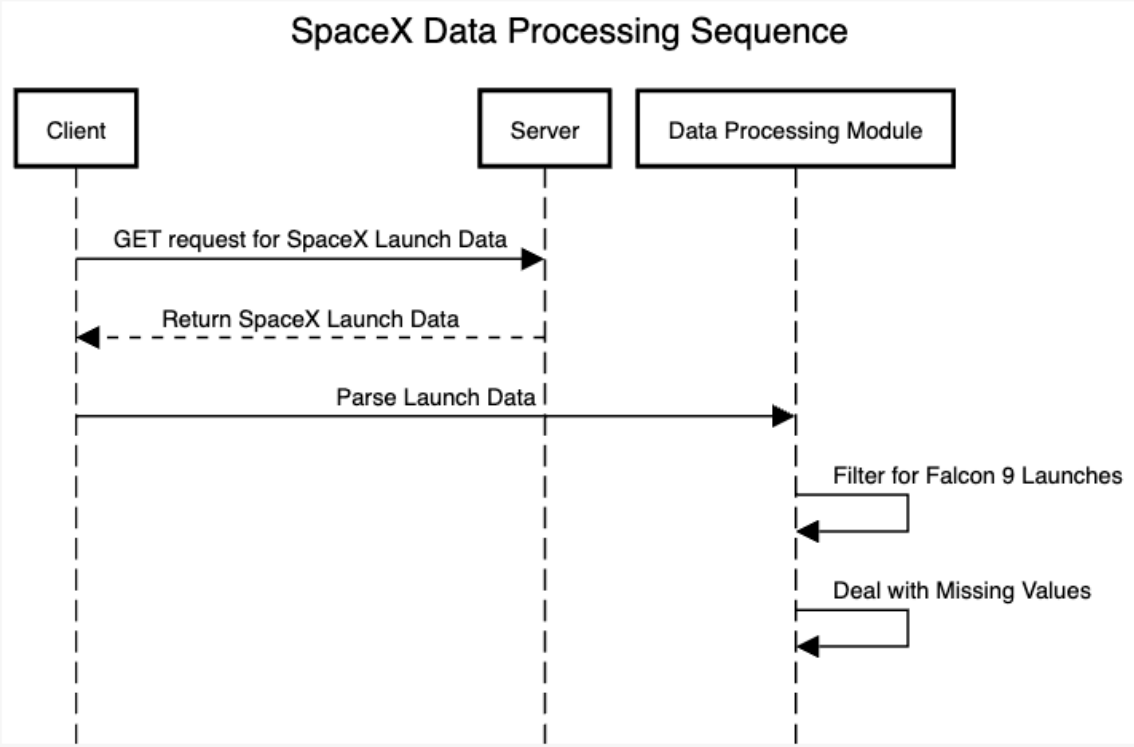
- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, the JSON response content was converted into a Pandas DataFrame using .json_normalize().
  - Data cleansing and preprocessing was performed.
  - As an alternative, data was collected using web scraping techniques from Wikipedia for Falcon 9 launch records using the BeautifulSoup librairy.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a Pandas DataFrame for the project.

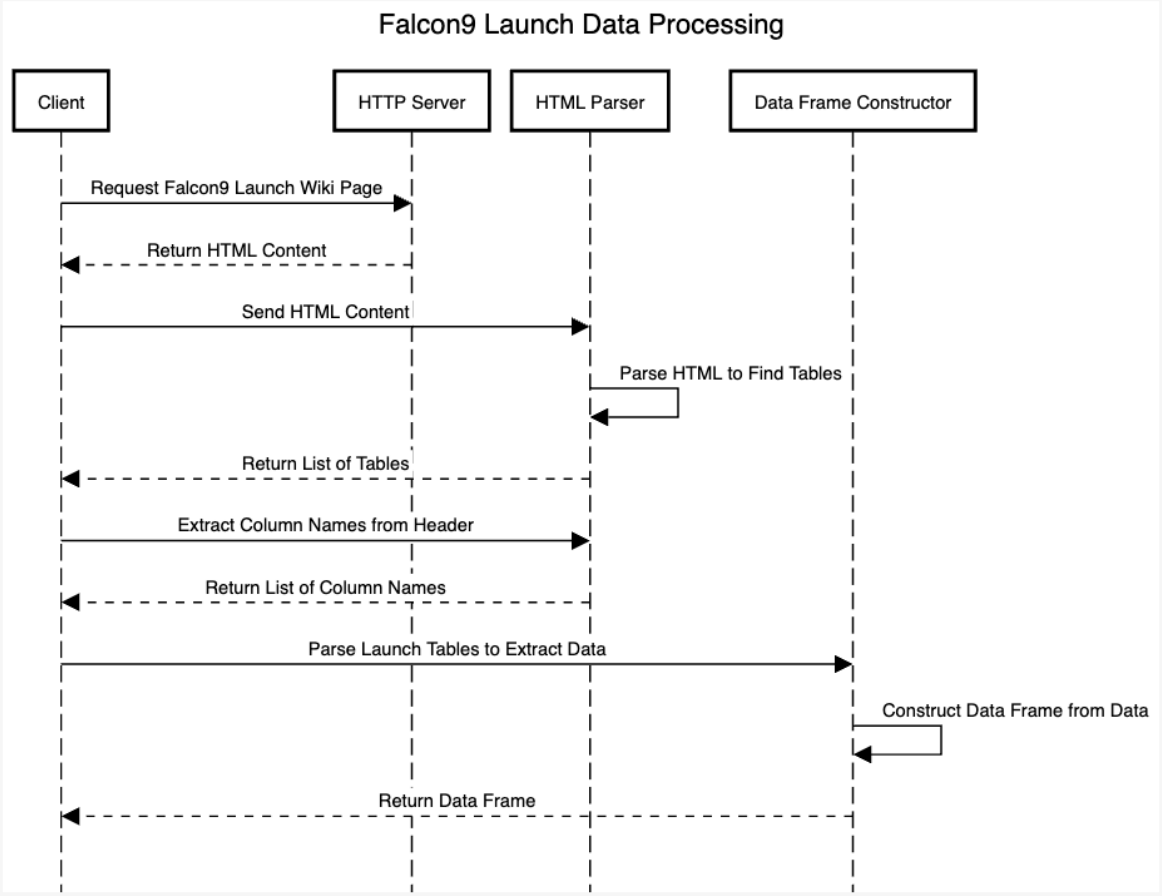
Data Collection – SpaceX API

- Request and parse the SpaceX launch data using the GET request
- Filter the dataframe to only include Falcon 9 launches
- Dealing with Missing Values
- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

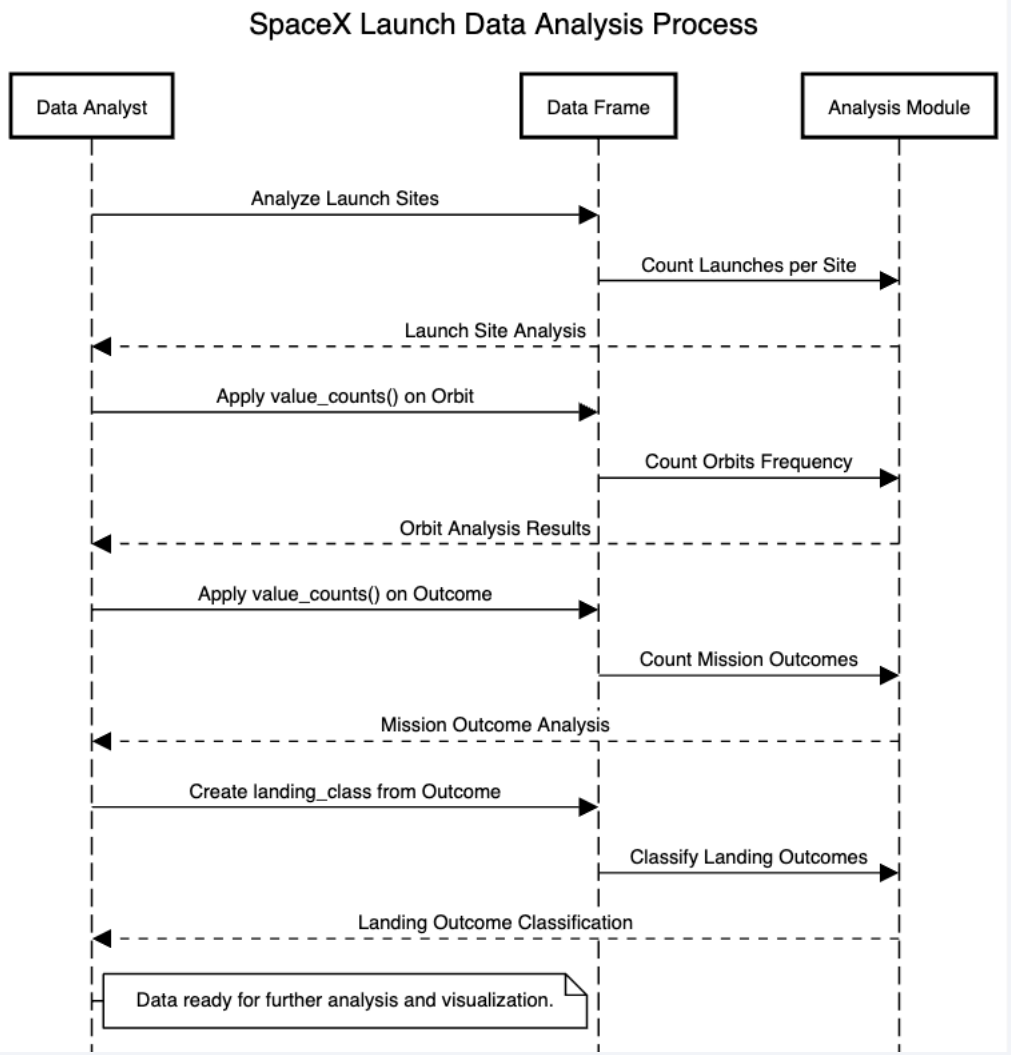
- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables
- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

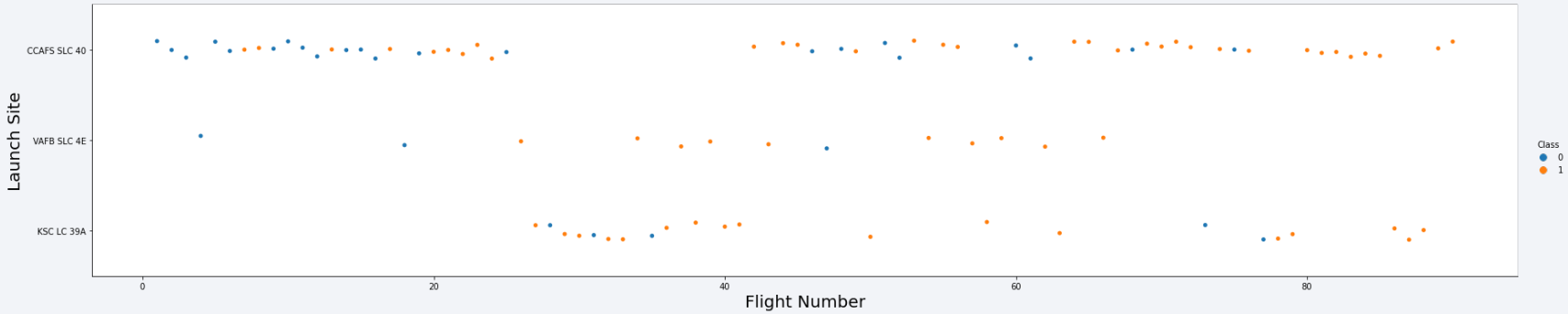
1. **Analyzing Launch Sites:** Investigate the launch sites to understand the distribution and frequency of launches from each site.
2. **Orbit Analysis:** Utilize .value_counts() on the Orbit column to analyze the frequency of each orbit type utilized in launches.
3. **Mission Outcome Analysis:** Apply .value_counts() on the Outcome column to categorize and count the different outcomes of missions, identifying successful and unsuccessful landings.
4. **Landing Outcome Labeling:** Create a binary classification for landing outcomes, where 0 represents unsuccessful landings, and 1 represents successful landings, based on predefined criteria of what constitutes a bad outcome.

- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- **Flight Number vs. Payload Mass:** To assess if experience and payload size influence launch success.
- **Flight Number vs. Launch Site:** To check for patterns in launch success across different sites.
- **Payload vs. Launch Site:** To see if certain sites are preferred for heavier payloads.
- **Success Rate by Orbit Type:** To identify which orbits yield higher success.
- **Flight Number vs. Orbit Type:** To explore if success in specific orbits is linked to experience.
- **Payload vs. Orbit Type:** To determine the impact of payload mass on success in each orbit.
- **Yearly Success Trend:** To visualize changes in success rate over time.
- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb



EDA with SQL

- **Identified Unique Launch Sites:** Extracted distinct launch site names to understand SpaceX's operational locations.
- **Filtered Launch Sites with 'CCA':** Selected records of launch sites starting with 'CCA' to focus on a specific area.
- **Summed NASA (CRS) Payload Mass:** Calculated the total payload mass for NASA (CRS) missions, highlighting their contribution.
- **Averaged Payload for F9 v1.1 Booster:** Determined the average payload mass for the Falcon 9 v1.1 booster.
- **Found First Successful Ground Pad Landing:** Identified the date of SpaceX's first successful ground pad landing.
- **Listed Boosters with Successful Drone Ship Landings:** Named boosters that achieved successful drone ship landings within a specific payload range.
- **Counted Mission Outcomes:** Tallied successful and failed missions to gauge SpaceX's success rate.
- **Identified Boosters with Maximum Payload:** Found boosters that carried the maximum payload mass, showcasing their capacity.
- **Analyzed 2015 Drone Ship Landing Failures:** Focused on failed drone ship landings in 2015, pinpointing challenges.
- **Ranked Landing Outcomes by Count:** Ranked landing outcomes by frequency between specific dates to assess performance over time.
- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Markers** were employed to pinpoint the exact locations of launch sites. Each marker was color-coded to represent the launch outcome, with distinct colors for successes and failures. This visualization technique facilitates the immediate identification of each site's performance history.
- **Circles** were utilized to highlight areas of interest around the launch sites, such as proximity to infrastructure (railways, highways) and natural features (coastlines). The radius of each circle was chosen to reflect distances of relevance, allowing for the assessment of strategic positioning relative to these features.
- **Lines** were drawn to connect launch sites with nearby points of interest, such as cities, to visually represent the distance between them. This was particularly useful in evaluating whether launch sites are strategically located at safe distances from populated areas to mitigate risk in the event of a launch failure.
- The addition of these objects to the folium map was driven by the need to visually convey complex spatial relationships and outcomes in an intuitive and interactive manner. By integrating these map objects, the map not only serves as a tool for geographical orientation but also as an analytical instrument to assess the success rates of launch sites and their strategic positioning in relation to critical infrastructure and natural features.
- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Creation of an interactive dashboard using Plotly Dash to visualize launch success rates and analyze the impact of payload on mission outcomes.

1. A dropdown menu to select between different launch sites or view aggregate data, enabling users to compare the success rates across sites or examine a specific site in detail.
2. A pie chart to visualize the success and failure counts, providing a quick and clear representation of launch outcomes.
3. A range slider to filter the data by payload mass, allowing for an exploration of the correlation between payload size and launch success.
4. A scatter plot to display the relationship between payload and launch outcome, with color-coded points indicating different booster versions, offering insights into how these variables interact.

- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/spacex_dash_app.py

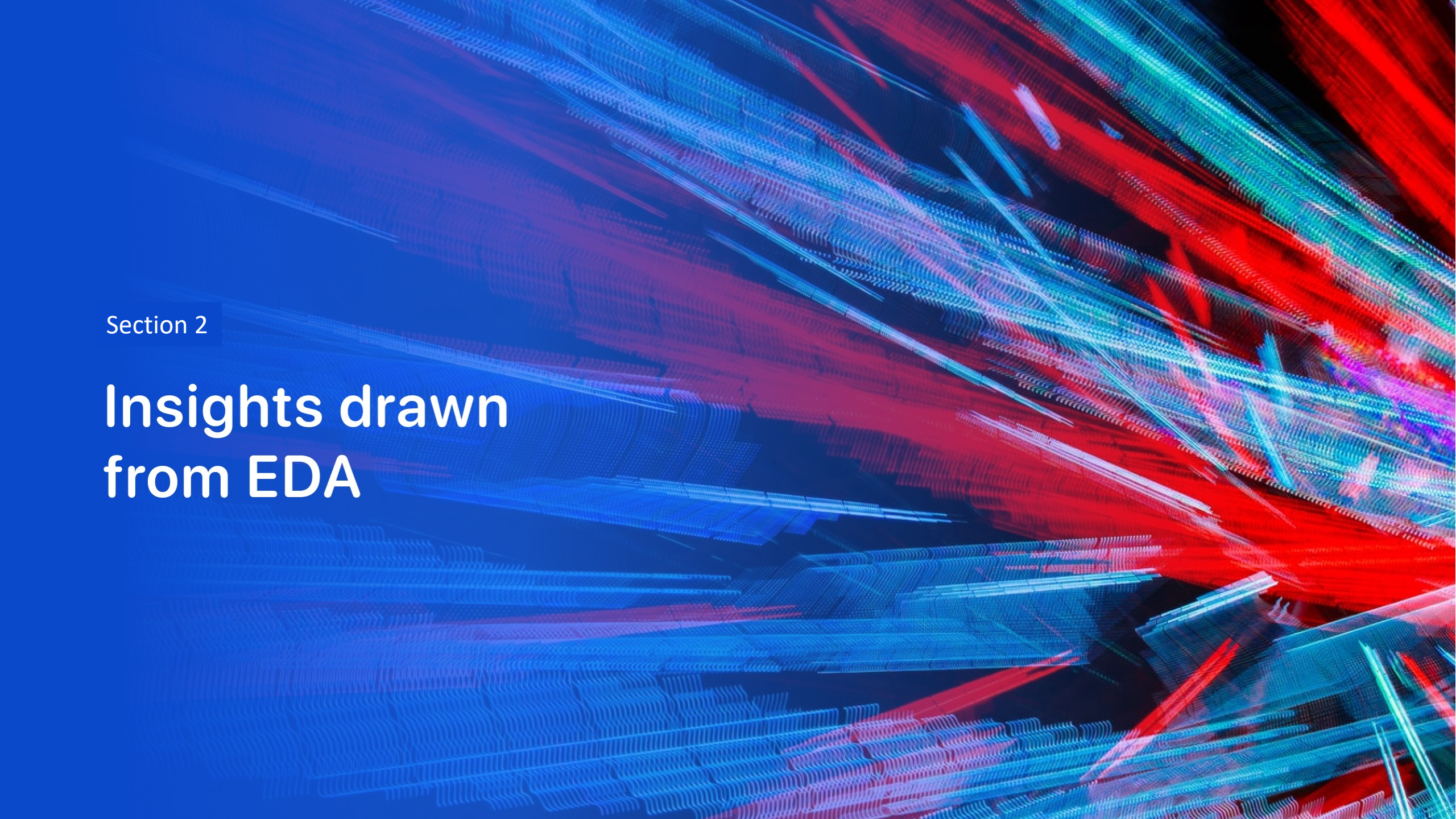
Predictive Analysis (Classification)

- **Data Preparation:** Converted target variable 'Class' to a NumPy array. Standardized feature variables for uniformity.
- **Model Training:** Split data into training (80%) and testing (20%) sets. Employed cross-validation to optimize model parameters.
- **Model Selection:** Utilized GridSearchCV to tune hyperparameters for multiple models: Logistic Regression, Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbors (KNN)
- **Model Evaluation:** Assessed model performance using accuracy scores. Analyzed confusion matrices to identify false positives and negatives.
- **Optimization:** Refined models by selecting the best hyperparameters from GridSearchCV.
- **Best Model Identification:** Compared models based on validation scores. Decision Tree identified as the best model with specific hyperparameters set. Evaluated the best model on the test data to confirm performance.

- https://github.com/SLTResearch/ibm_ds_certificate/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

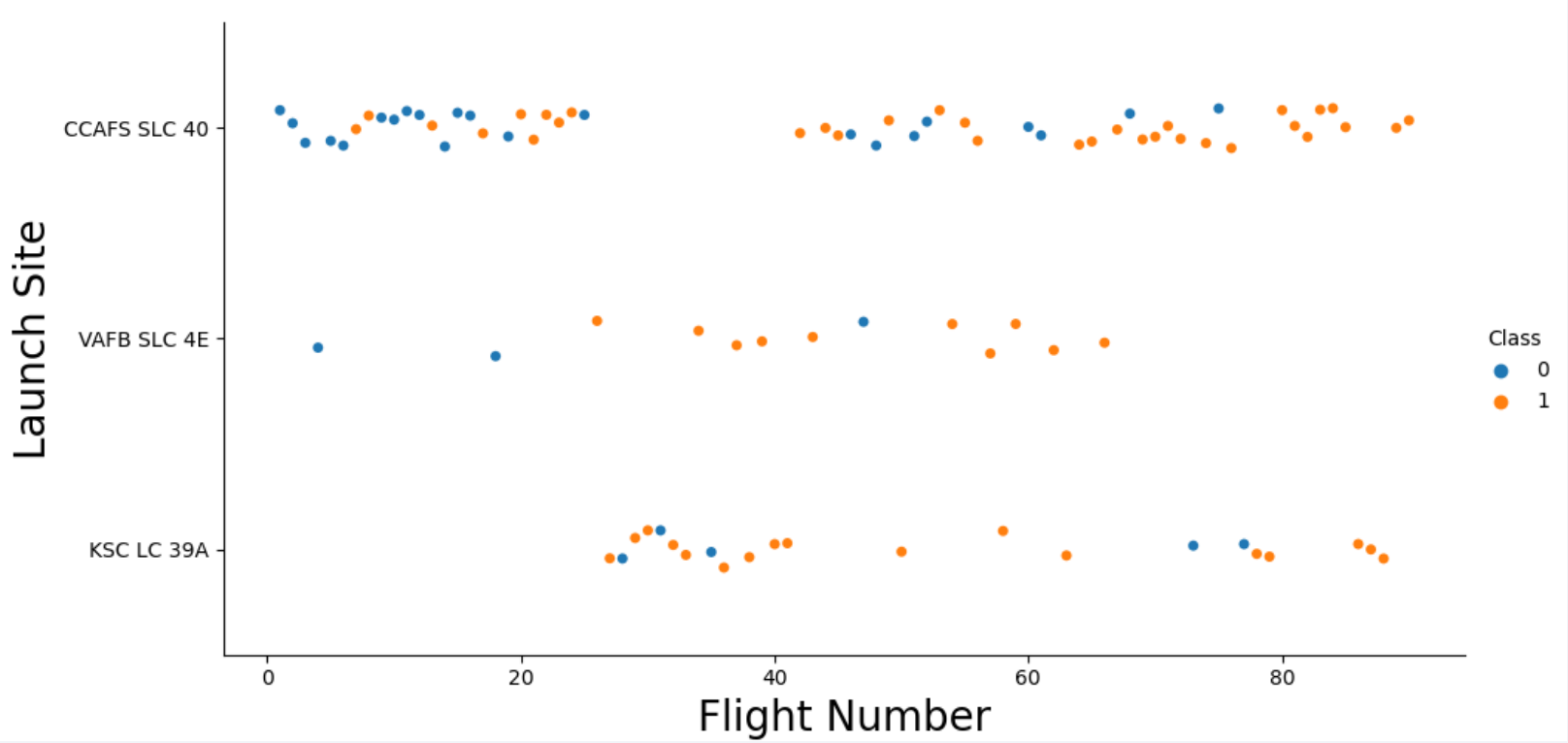
Results

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


Section 2
Insights drawn from EDA

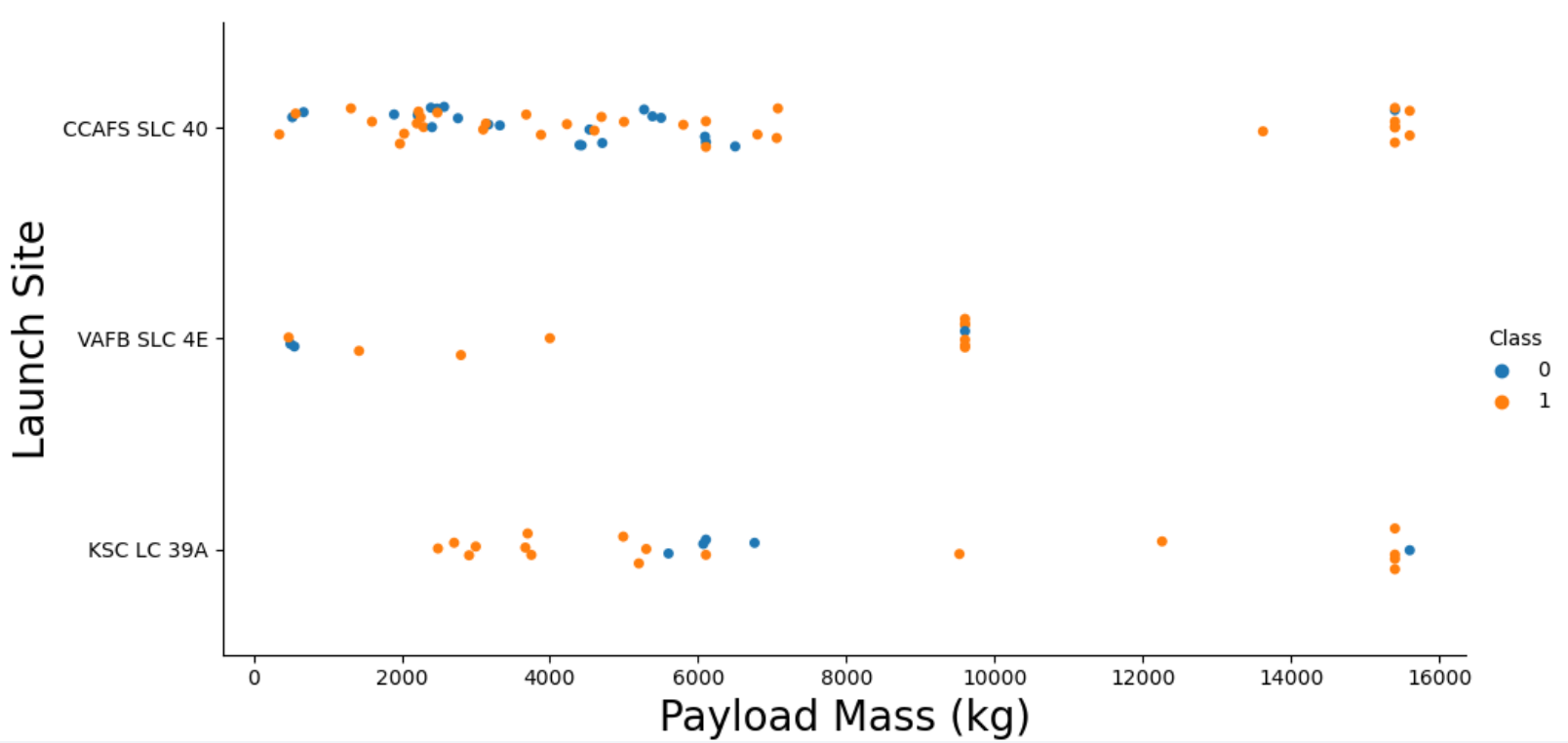
Flight Number vs. Launch Site

- As the Flight Number increase per Launch Site, the greater the Success Rate



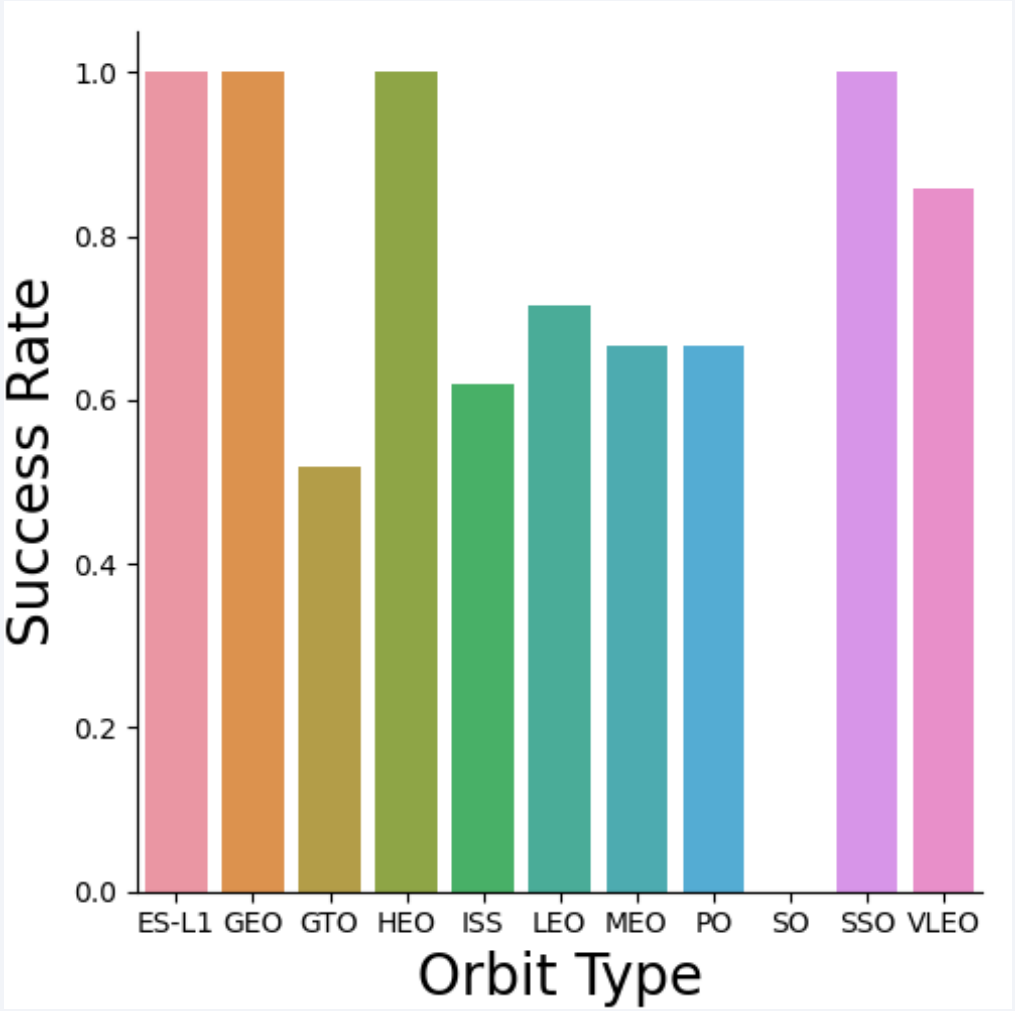
Payload vs. Launch Site

- Surprisingly, the heavier the payload is, the greater is the success rate. Could be due to the fact that first launched included lighter payloads.



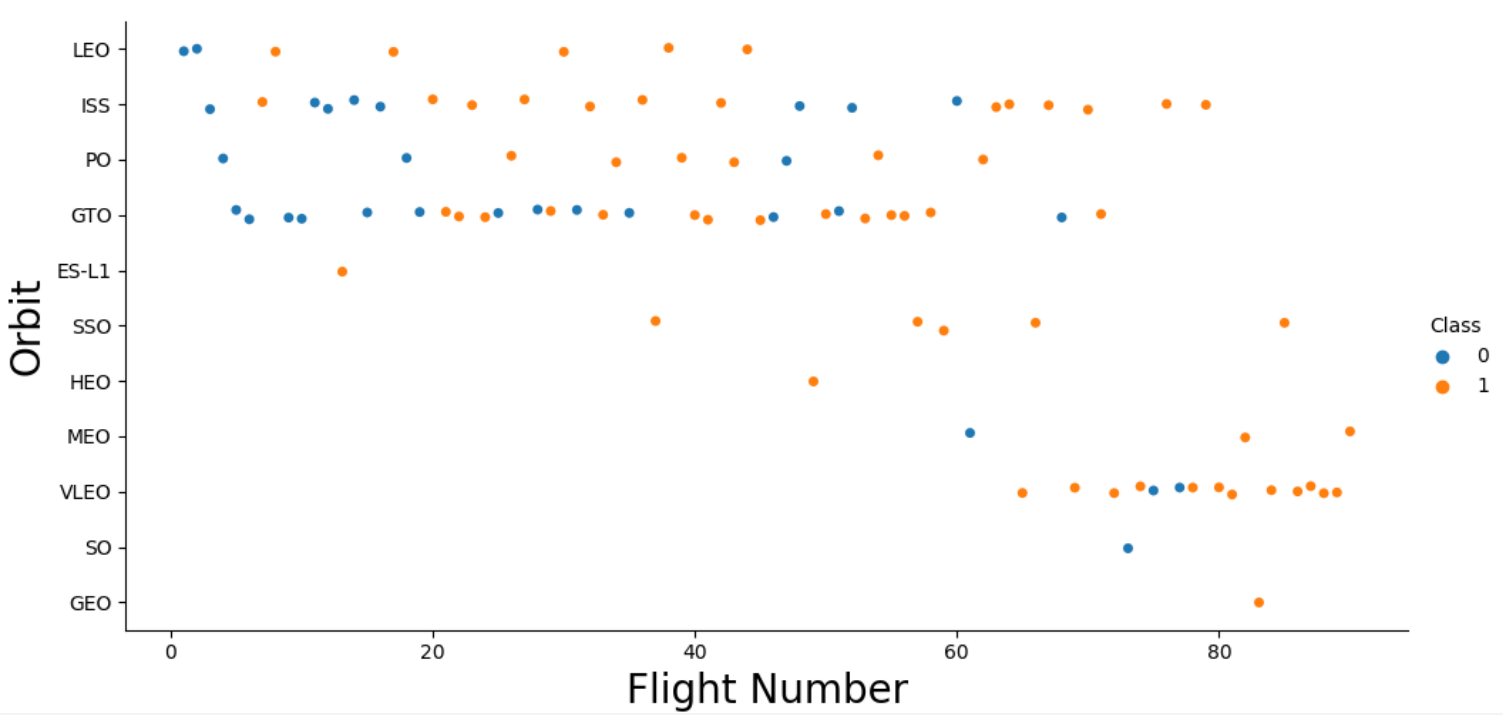
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, have a success rate of 100%.
- GTO exhibits a success rate below 60%, probably due to its farther distance to Earth than some lower-earth orbits.



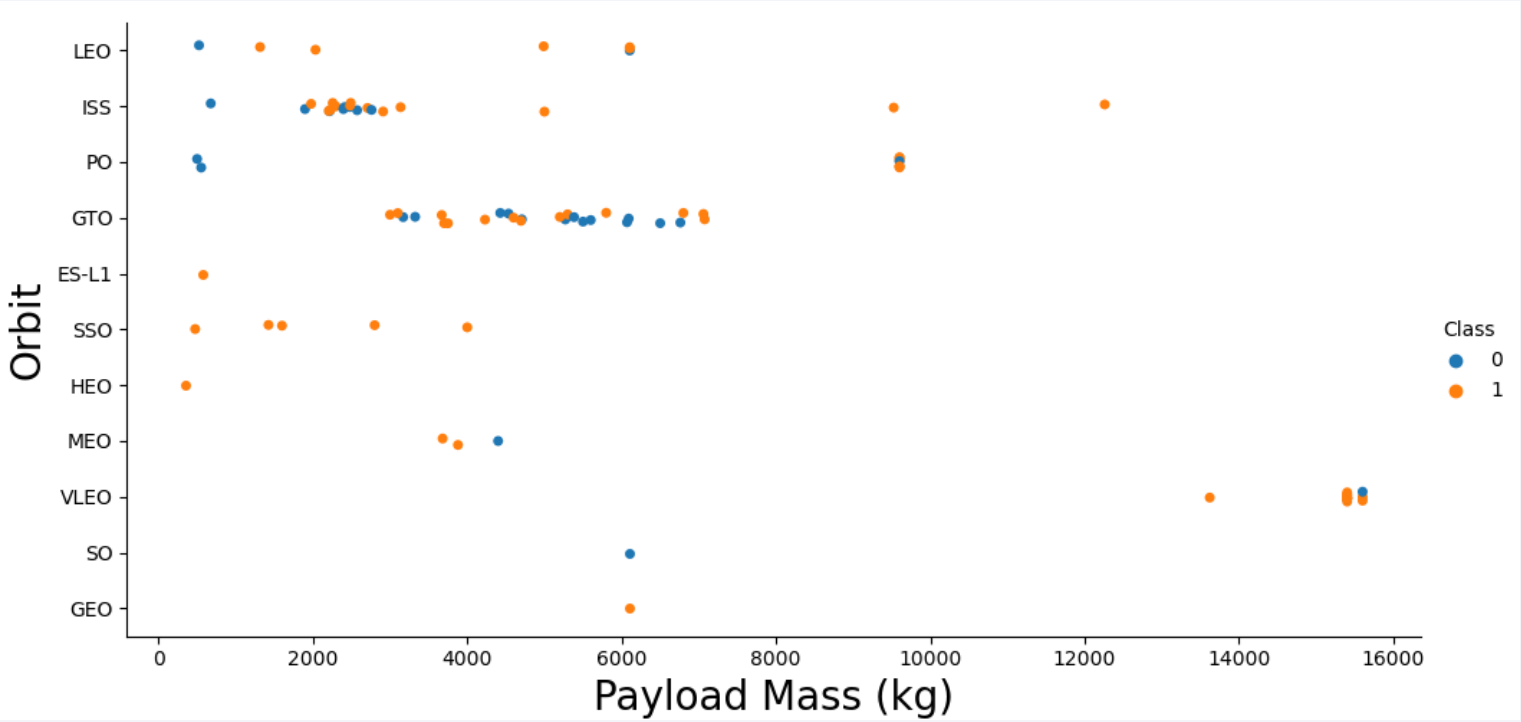
Flight Number vs. Orbit Type

- Logically, first flights were to lower orbits while most recent flights were to farther orbits like Medium Earth and Geostationary orbits.



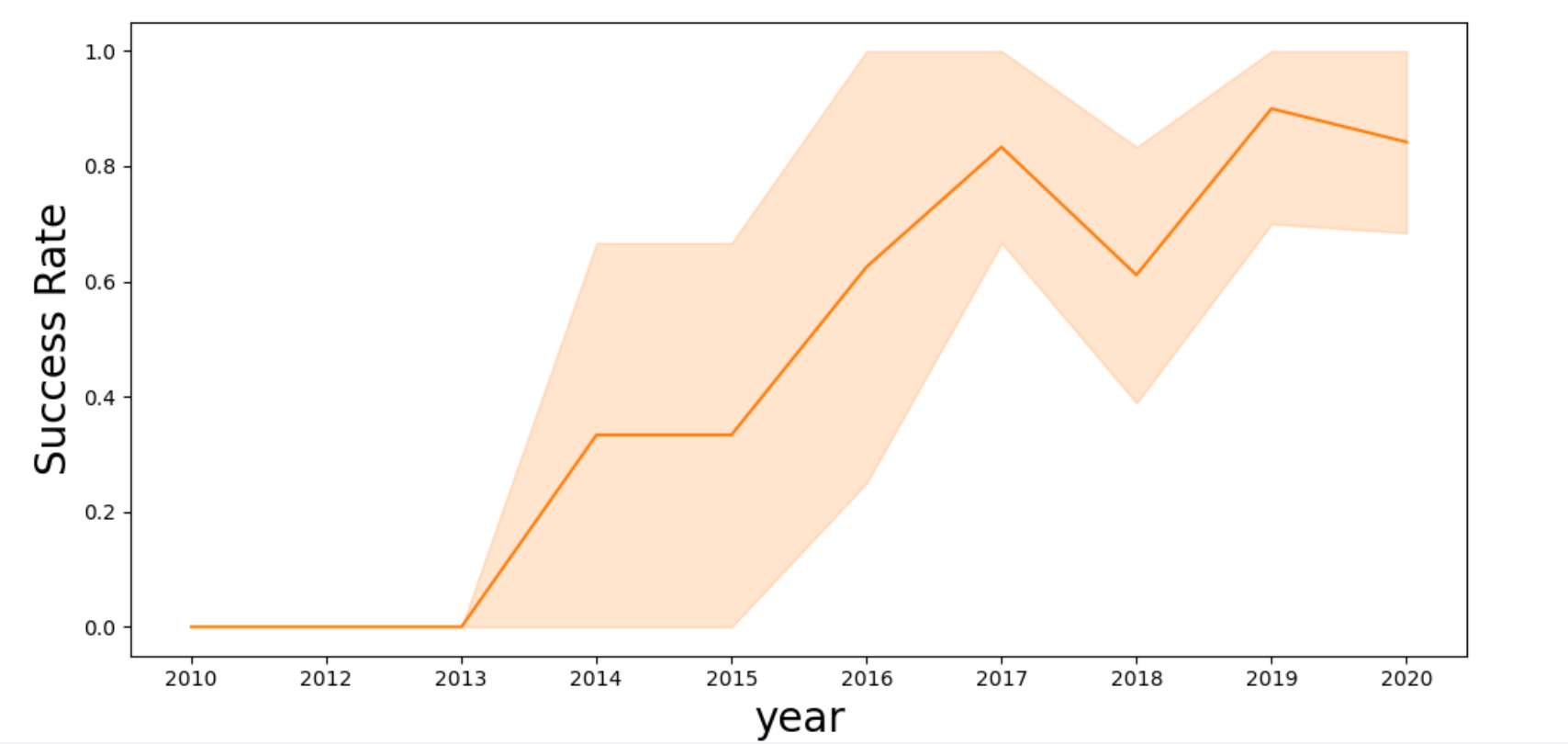
Payload vs. Orbit Type

- Lower payload mass is usually sent to lower orbit. GTO and GEO for example did not have flights with payload mass below 3 tons.



Launch Success Yearly Trend

- Success rate has improved since 2013 over time despite a small drop in 2017-2018, recovered in the following years



All Launch Site Names

- There were 4 unique launch sites:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40
- SQL query below:

Task 1

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

In [9]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

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

Out[9]:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`
- SQL Query:

Task 2

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

In [10]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[10]:

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 total payload carried by boosters from NASA is **45,596 Kg**

Task 3

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

In [11]:
```
%sql SELECT SUM(Payload_Mass__KG_) AS Total_Payload_Mass  FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
```

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

Out[11]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is **2,928.4 Kg**

Task 4

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

In [13]:
```
%sql SELECT AVG(Payload_Mass__KG_) AS Average_Payload_Mass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

Out[13]:

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

- The first successful landing in ground pad was on 12/22/2015.

Task 5

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

Hint:Use min function

```
[9]: %sql SELECT MIN(Date) AS First_Successful_Landing FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
 * sqlite:///my_data1.db
Done.
```

[9]:

First_Successful_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:
  - JSCAT-14
  - JSCAT-16
  - SES-10, SES-11 / EchoStar 105

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

```
[10]: %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.
```

[10]:

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes is as follows:
  - Failure in flight: 1
  - Success: 99
  - Success (payload status unclear): 1

Task 7

List the total number of successful and failure mission outcomes

```
[12]: %sql SELECT MISSION_OUTCOME, COUNT(*) AS total_number FROM SPACEXTBL GROUP BY MISSION_OUTCOME
 * sqlite:///my_data1.db
Done.
```

[12]:

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

Boosters Carried Maximum Payload

- 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

Task 8

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

In [17]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Payload_Mass__KG_ = (SELECT MAX(Payload_Mass__KG_) FROM SPACEXTE
```

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

Out[17]:

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

- List the failed landing_outcomes in drone ship, their booster versions, launch site names, and month in 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.

```
[15]: %sql SELECT SUBSTR(Date, 6, 2) AS MONTH, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Failure (drone ship)' AND SUBSTR(Date, 1, 4) = '2(
```

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

[15]:

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

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

- Ranked 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.

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.

[16]:
```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY Outcome_Count DESC;
```

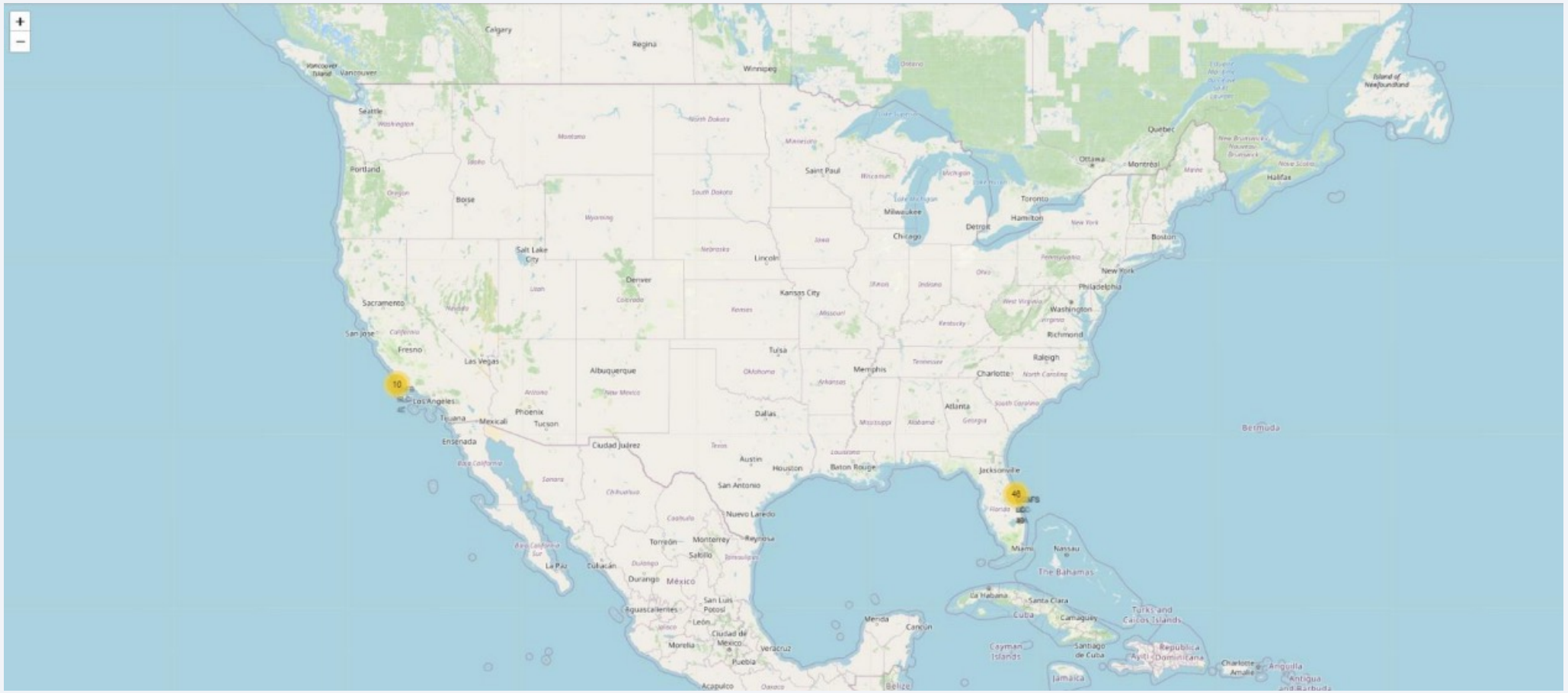
```
 * sqlite:///my_data1.db
Done.
```

[16]:

Landing_Outcome	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

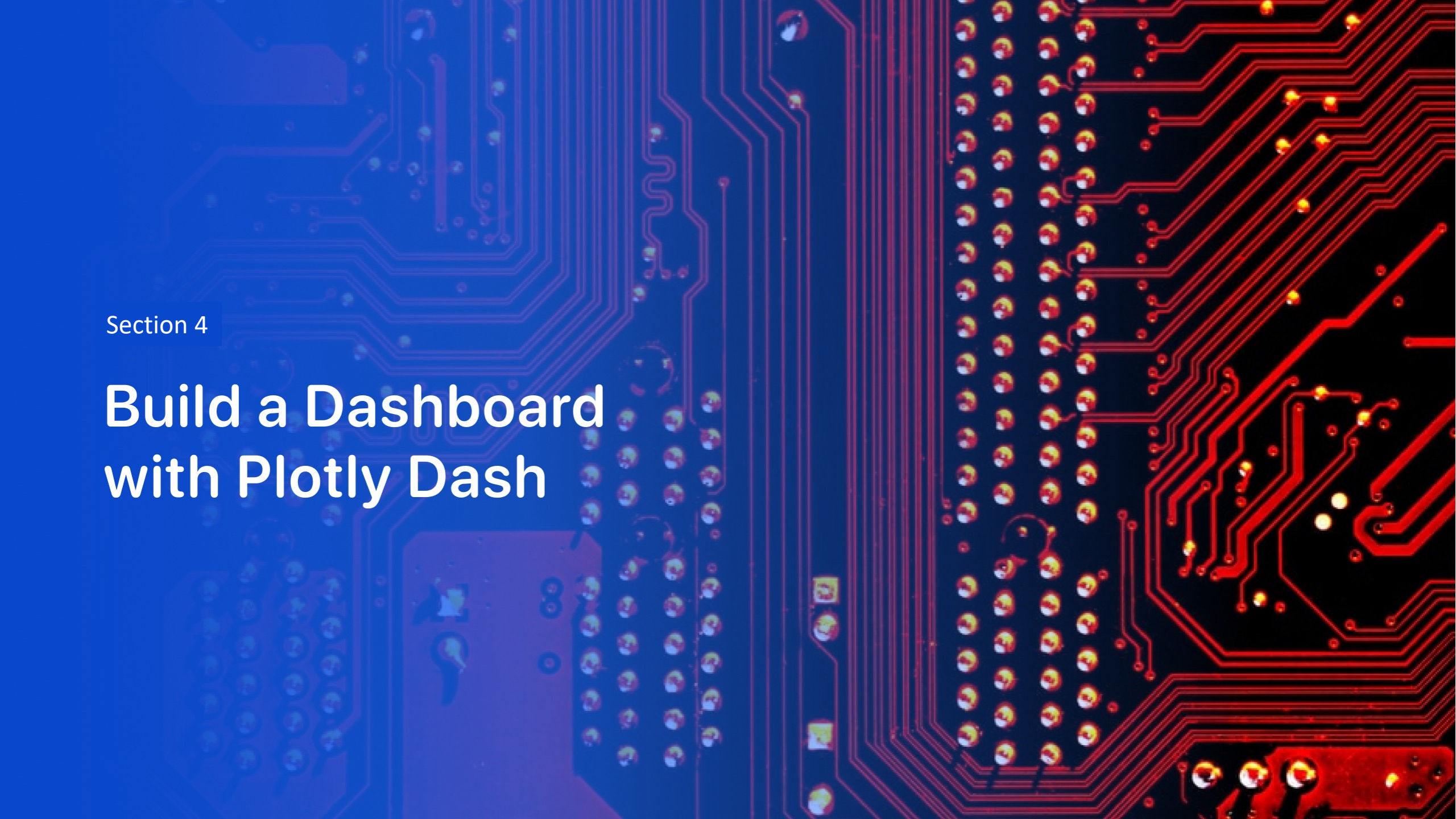
Launch Sites Map



Launch Outcomes Map

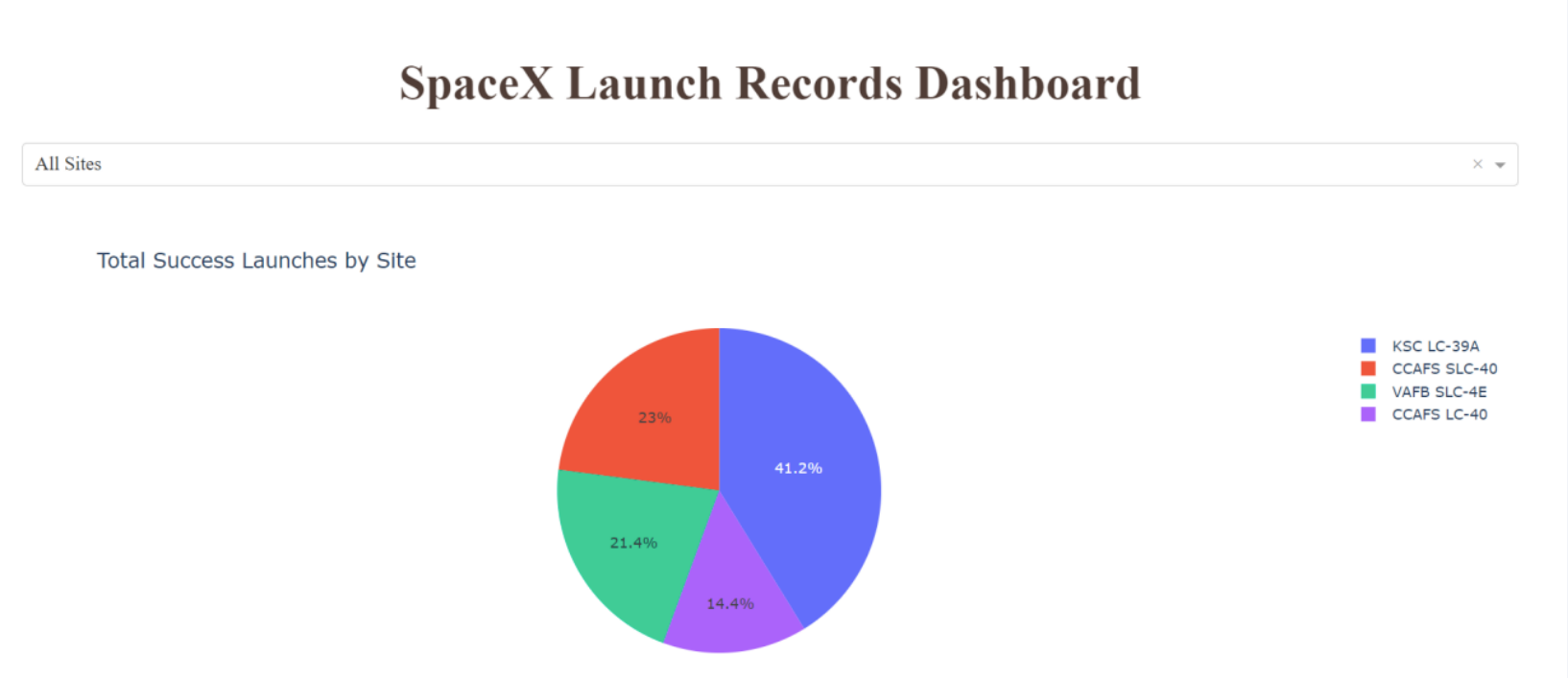
Launch Site Environment

- CCAFS SLC-40
  - 860 m from nearest coastline
  - 21.96 km from nearest railway
  - 23.23 km from nearest city
  - 26.88 km from nearest highway


Section 4
Build a Dashboard with Plotly Dash

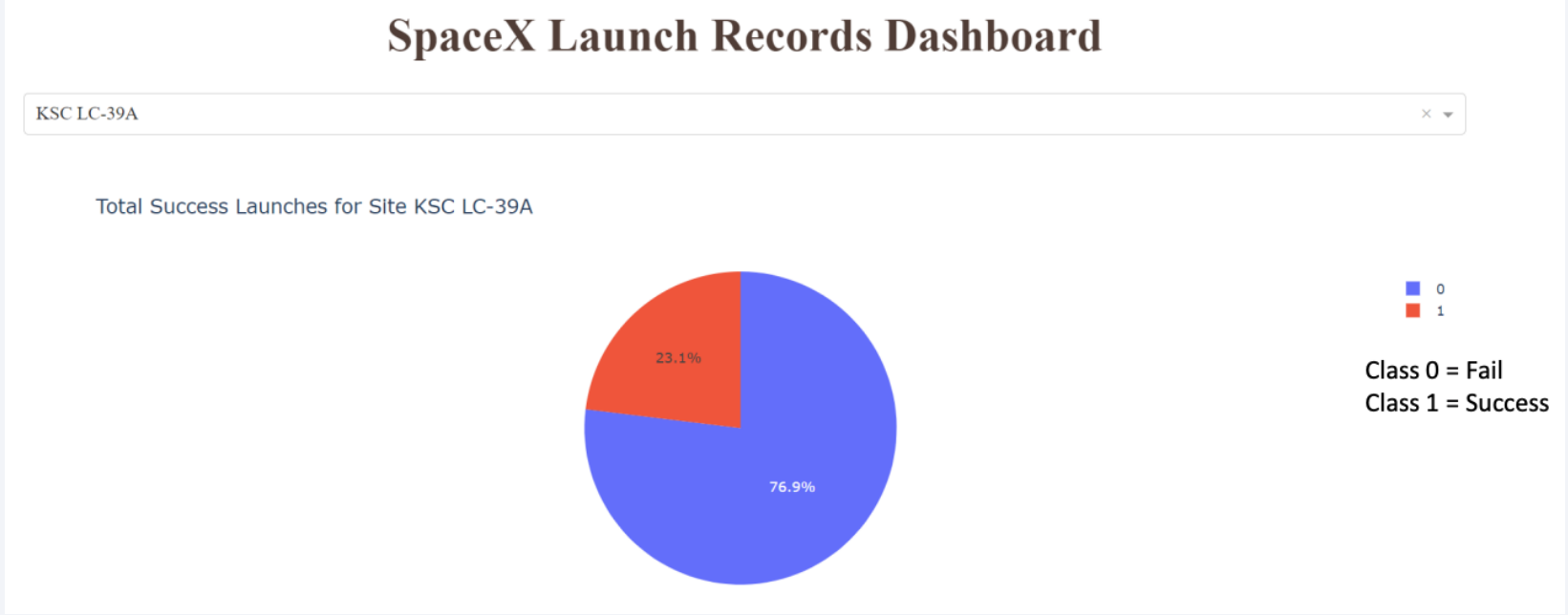
Launch Success by Site

- The most successful site is KSC LC-39A



Most Successful Launch Site Detail

- Successful launches 76.9% of the time from the site KSC LC-39A



Payload vs. Launch Outcome

- Concentration of successful launches for payloads between 2,000 and 4,000 Kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Logistic Regression

```
In [10]: print("tuned hyperparameters :(best parameters) ",logreg_cv.best_params_)
         print("accuracy :", logreg_cv.best_score_)
```

```
tuned hyperparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

- SVM

```
In [14]: print("tuned hyperparameters :(best parameters) ",svm_cv.best_params_)
         print("accuracy :",svm_cv.best_score_)
```

```
tuned hyperparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

- Tree

```
In [18]: print("tuned hyperparameters :(best parameters) ",tree_cv.best_params_)
         print("accuracy :",tree_cv.best_score_)
```

```
tuned hyperparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_sam
ples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.8732142857142856
```

- KNN

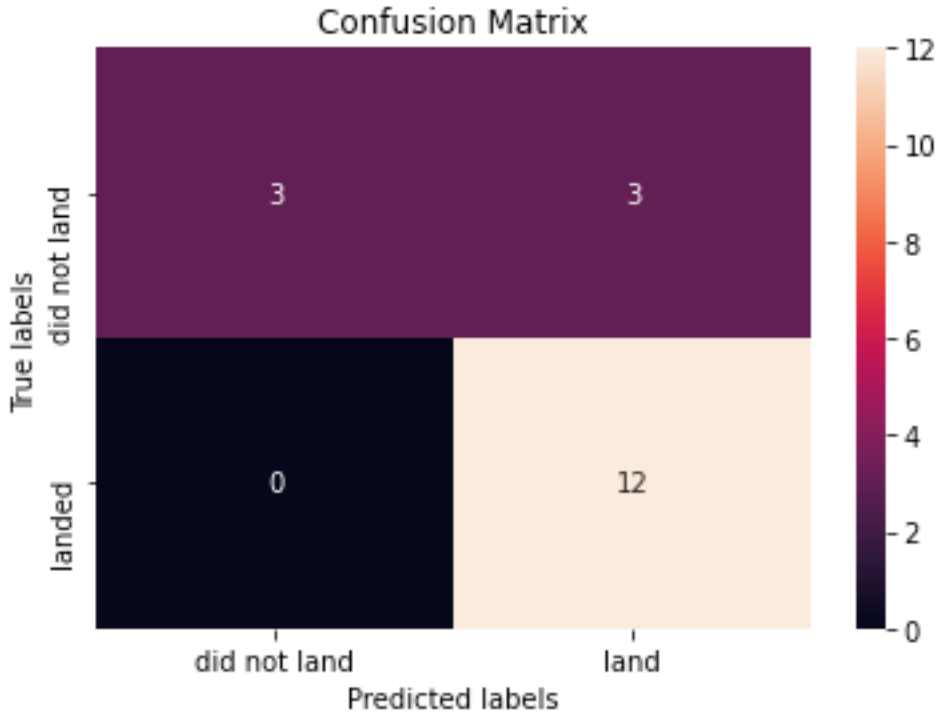
```
In [14]: print("tuned hyperparameters :(best parameters) ",svm_cv.best_params_)
         print("accuracy :",svm_cv.best_score_)
```

```
tuned hyperparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

Confusion Matrix

- Best performing model is KNN

```
In [24]: yhat_knn = knn_cv.predict(X_test)
         plot_confusion_matrix(Y_test, yhat_knn)
```



Conclusions

- **Data Insights:**
  - Launch success rates have shown an upward trend over time.
  - A clear correlation exists between launch success and variables such as flight number, payload mass, and orbit types.
- **Launch Site:**
  - KSC LC-39A stands out with the highest success rate among all evaluated launch sites.
  - Launch sites are predominantly located near coastlines and close to the equator, optimizing for payload efficiency and safety considerations.
- Model Performance:
  - All models demonstrated comparable accuracy on the test data.
  - The KNN model exhibited superior performance during training.
  - Predictive accuracy for launch success exceeds 80%, indicating a reliable classification capability.

Thank you!