IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MOATAZ MUSTAFA
Feb-17-2023

Outline

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

Executive Summary

- **Summary of methodologies**

❖ **Data Collection and Processing:**
  - ✓ Utilized SpaceX REST API and web scraping techniques for comprehensive data collection.
  - ✓ Engineered a success/fail outcome variable through meticulous data wrangling.

❖ **Exploratory Data Analysis (EDA):**
  - ✓ Explored the dataset through SQL analysis and data visualization techniques.
  - ✓ Investigated factors such as payload, launch site, flight number, and yearly trends.

❖ **Interactive Analytics and Visualization:**
  - ✓ Leveraged Folium for interactive visualizations to enhance data exploration.
  - ✓ Investigated launch site success rates and their proximity to geographical markers.

❖ **Predictive Modeling:**
  - ✓ Developed machine learning models, including logistic regression, support vector machine (SVM), decision tree, and K-nearest neighbor (KNN).

- **Summary of all results**

❖ **Exploratory Data Analysis:**
  - ✓ Highlighted the improvement in launch success rates over time.
  - ✓ Identified specific launch sites with high success rates, emphasizing KSC LC-39A.
  - ✓ Recognized specific orbits (ES-L1, GEO, HEO, SSO) with a 100% success rate.

❖ **Interactive Analytics and Visualization:**
  - ✓ Provided screenshots showcasing interactive analytics and key visualizations.
  - ✓ Visualized launch sites with the highest success rates and successful payload ranges.

❖ **Predictive Analytics:**

1. All models performed similarly on the test set, with a slight advantage for the decision tree model.

Introduction

- **Project background and context**

SpaceX, a pioneering force in the space industry, revolutionizes space travel affordability with its innovative Falcon 9 rocket. While competitors incur costs upwards of $165 million per launch, SpaceX's ability to reuse the first stage slashes expenses to $62 million. This pivotal cost reduction hinges on predicting the successful landing of the first stage. The project's primary goal is to establish a machine learning pipeline capable of forecasting the first stage's landing outcome.

- **Problems you want to find answers**

✓ Identify the factors influencing a successful rocket landing.

✓ Explore the intricate interactions among features determining landing success.

✓ Determine the necessary operating conditions for a successful landing program.

✓ Analyzing how payload mass, launch site, number of flights, and orbits impact first-stage landing success.

✓ Examining the trend of successful landings over time.

✓ Identifying the optimal predictive model for successful landings (binary classification).

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data acquisition involved utilizing both SpaceX's API and web scraping from Wikipedia
- Perform data wrangling
    - The dataset underwent filtering, handling of missing values, and the application of one-hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Predict landing outcomes through the implementation of classification models. Perform model tuning and evaluation to identify the optimal model and parameters.

Data Collection

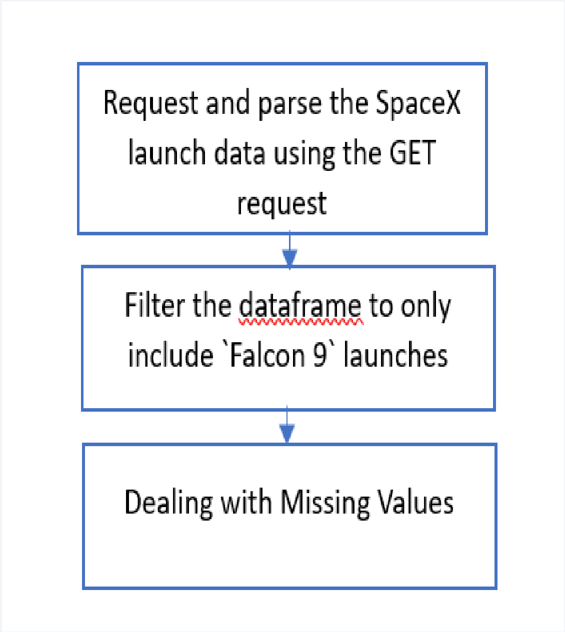
Data collection involved employing multiple methods. Initially, we utilized a GET request to access the SpaceX API. Following this, the response content was decoded into JSON format using the .json() function call, and the data was transformed into a Pandas dataframe using .json_normalize(). Subsequently, we conducted data cleaning, addressing missing values as needed.

Additionally, we conducted web scraping from Wikipedia to gather Falcon 9 launch records. Using BeautifulSoup, we targeted the launch records presented as an HTML table. The table was then parsed and converted into a Pandas dataframe for subsequent analysis.

Data Collection – SpaceX API

- **Initiate SpaceX REST Calls:** Request Falcon 9 launch data through SpaceX REST API.
- **HTML Parsing with BeautifulSoup:** Create a BeautifulSoup object to parse the HTML response.
- **Header Extraction:** Extract column names from the HTML table header for data organization.
- **Data Retrieval from HTML Tables:** Collect relevant data by parsing the HTML tables containing launch information.
- **Dictionary Creation:**Formulate a dictionary structure from the acquired data.
- **Dataframe Construction:** Build a Pandas dataframe using the created dictionary for structured data representation.

1. **Data Export:** Conclude the process by exporting the gathered data to a CSV file for further analysis and reference.



- The GitHub URL of the completed SpaceX API calls: https://github.com/buluk85/Capstone-SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

❖ In outlining our web scraping process, the journey unfolds through the following key steps:

✓ **Exploratory Data Analysis (EDA):**Conduct an in-depth EDA to determine data labels and gain foundational insights.

✓ **Quantitative Calculations:** Calculate essential metrics, including the number of launches for each site, the occurrence of different orbits, and the frequency of mission outcomes per orbit type.

✓ **Binary Landing Outcome Creation:**Introduce a binary landing outcome column, establishing it as the dependent variable. And export the enriched dataset to a CSV file for further analysis.

✓ **Landing Outcome Classification:** Acknowledge the variability in landing success.

✓ **True Ocean:** Signifies a mission outcome with a successful landing in a specific region of the ocean.

✓ **False Ocean:** Represents an unsuccessful landing in a specific ocean region.

✓ **True RTLS (Return To Launch Site):** Indicates a successful landing on a ground pad.

✓ **False RTLS:**Denotes an unsuccessful landing on a ground pad.

✓ **True ASDS (Autonomous Spaceport Drone Ship):** Conveys a successful landing on a drone ship.

✓ **False ASDS:**Indicates an unsuccessful landing on a drone ship.

❖ **Outcome Conversion:** Convert outcomes into binary values, where 1 signifies a successful landing, and 0 signifies an unsuccessful landing. This comprehensive approach, will effectively communicate the intricacies of our web scraping methodology during the presentation.

Data Wrangling

❖ In illustrating our data wrangling process, we engaged in the following crucial steps:

✓ **Exploratory Data Analysis (EDA):** Executed EDA to discern the training labels, gaining critical insights into the dataset.

✓ **Quantitative Analysis:** Conducted a quantitative analysis by calculating the count of launches at each site and quantifying the number and occurrence of various orbits.

✓ **Landing Outcome Label Creation:** Formulated a landing outcome label based on the existing outcome column, defining the success or failure of the landing.

✓ **Exporting Processed Data:** Concluded the data wrangling phase by exporting the results, including the newly created landing outcome labels, to a CSV file for subsequent analysis.

This streamlined presentation ensures clarity in conveying the key phases of our data wrangling process.

❖ the GitHub URL of your completed data wrangling related notebooks: https://github.com/buluk85/Capstone-SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

❖ In summarizing our data visualization approach, we strategically employed various chart types to unravel key insights:

✓ **Scatter Plots for Relationship Exploration:** Utilized scatter plots to visually assess relationships between variables. This technique is particularly beneficial for identifying potential relationships that could be pertinent for machine learning applications.

✓ **Bar Charts for Comparisons:** Employed bar charts to showcase comparisons among discrete categories. These charts effectively illustrate relationships between different categories and their corresponding measured values.

✓ **Specific Visualizations:** Explored the data intricacies through targeted visualizations, including:

- ➢ Relationship between flight number and launch site.
- ➢ Payload and launch site dynamics.
- ➢ Success rate variation across different orbit types.
- ➢ Interaction between flight number and orbit type.
- ➢ Examined the launch success trend over the years.

Each chart type was chosen purposefully to shed light on specific aspects of the data, fostering a comprehensive understanding of the relationships within the dataset.

❖ The GitHub URL of your completed EDA with data visualization notebook:https://github.com/buluk85/Capstone-SpaceX/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

❖ In SQL, we seamlessly integrated the SpaceX dataset into a SQL Server database, unleashing the power of EDA through targeted queries. The following SQL queries provided invaluable insights:

✓ Unique Launch Sites: Identified the names of unique launch sites in the space mission.

✓ NASA CRS Payload Analysis: Calculated the total payload mass carried by boosters launched by NASA (CRS).

✓ Average Payload Mass for F9 v1.1: Determined the average payload mass carried by booster version F9 v1.1.

✓ Mission Outcomes: Computed the total number of successful and failure mission outcomes.

✓ Failed Drone Ship Landings: Isolated failed landing outcomes on drone ships, providing details on their associated booster versions and launch site names.

These SQL queries not only enabled us to extract specific information but also facilitated a comprehensive exploratory data analysis, enhancing our understanding of the SpaceX mission dataset.

❖ The GitHub URL of your completed EDA with SQL notebook:https://github.com/buluk85/Capstone-SpaceX/blob/main/jupyter-labs-eda-sql-edx_sqllite.ipynb

Build an Interactive Map with Folium

In the creation of our Folium map, we strategically incorporated various map objects to enhance visualization and analysis. The following map objects were utilized:

- Markers for Launch Sites: Placed markers on all launch sites, providing a visual reference for their locations.
- Markers, Circles, and Lines for Launch Outcomes: Incorporated markers, circles, and lines to visually represent the success or failure of launches for each site. This approach offered an intuitive way to distinguish outcomes on the map.
- Class Assignment for Launch Outcomes: Assigned launch outcomes (failure or success) to classes 0 and 1, respectively. This classification allowed for effective color-coding of markers, aiding in quick identification of success rates.
- Color-Labeled Marker Clusters: Leveraged color-labeled marker clusters to discern launch sites with relatively high success rates. This visual representation facilitated a quick overview of success trends.
- Distance Calculation: Calculated distances between launch sites and their proximities to answer crucial questions, such as:

➢ Proximity to railways, highways, and coastlines.

➢ The spatial relationship between launch sites and nearby cities.

By incorporating these map objects and features, our Folium map not only presented launch site locations but also provided insightful visualizations that aided in the analysis of success rates and spatial considerations.

Build an Interactive Map with Folium (Con)

- Distance Calculation: Calculated distances between launch sites and their proximities to answer crucial questions, such as:
  - ➢ Proximity to railways, highways, and coastlines.
  - ➢ The spatial relationship between launch sites and nearby cities.

By incorporating these map objects and features, our Folium map not only presented launch site locations but also provided insightful visualizations that aided in the analysis of success rates and spatial considerations.

- The GitHub URL of the completed interactive map with Folium map: https://github.com/buluk85/Capstone-SpaceX/blob/main/Interactive%20Visual%20Analytics%20and%20Dashboard_Folium.ipynb

Build a Dashboard with Plotly Dash

In crafting our dashboard, we strategically incorporated diverse plots, graphs, and interactive features to offer a rich and insightful user experience. The key elements include:

✓ Dropdown List with Launch Sites:

➢ Purpose: Enables users to select either all launch sites or a specific launch site.

➢ Interactivity: Enhances user control and focus on specific launch site data.

✓ Pie Chart Showing Successful Launches:

➢ Purpose: Visualizes the proportion of successful and unsuccessful launches as a percentage of the total.

➢ Interactivity: Provides an at-a-glance overview of success rates, aiding quick insights.

✓ Slider of Payload Mass Range:

➢ Purpose: Allows users to dynamically select a payload mass range for focused analysis.

➢ Interactivity: Enhances customization and focus on specific payload mass ranges.

Build a Dashboard with Plotly Dash (CONT)

✓ Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version:

➢ Purpose: Illustrates the correlation between payload mass and launch success, differentiated by booster versions.

➢ Interactivity: Facilitates in-depth analysis and comparison of payload mass impact on launch success across different booster versions.

By incorporating these interactive elements, our dashboard not only presents essential data but also empowers users to customize their exploration, gaining nuanced insights into the relationships between launch sites, success rates, payload masses, and booster versions.

❖ The GitHub URL of the completed Plotly Dash lab: https://github.com/buluk85/Capstone-SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

❖ In the journey of developing our classification model, we meticulously followed key steps to ensure robustness and accuracy:

➢ Data Loading and Transformation:

✓ Loaded and transformed the dataset using NumPy and Pandas.

✓ Split the data into training and testing sets for comprehensive model evaluation.

➢ Model Construction:

✓ Constructed various machine learning models to explore the dataset comprehensively.

✓ Utilized GridSearchCV to systematically tune hyperparameters for enhanced model performance.

➢ Metric Selection: Chose accuracy as the primary metric for model evaluation.

- Model Improvement: Enhanced the model through meticulous feature engineering and algorithm tuning.

➢ Best Performing Model Identification: Evaluated multiple models and identified the best-performing classification model.

❖ The GitHub URL of the completed predictive analysis lab: https://github.com/buluk85/Capstone-SpaceX/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

In summarizing our results across different analytical dimensions, we arrived at insightful findings:

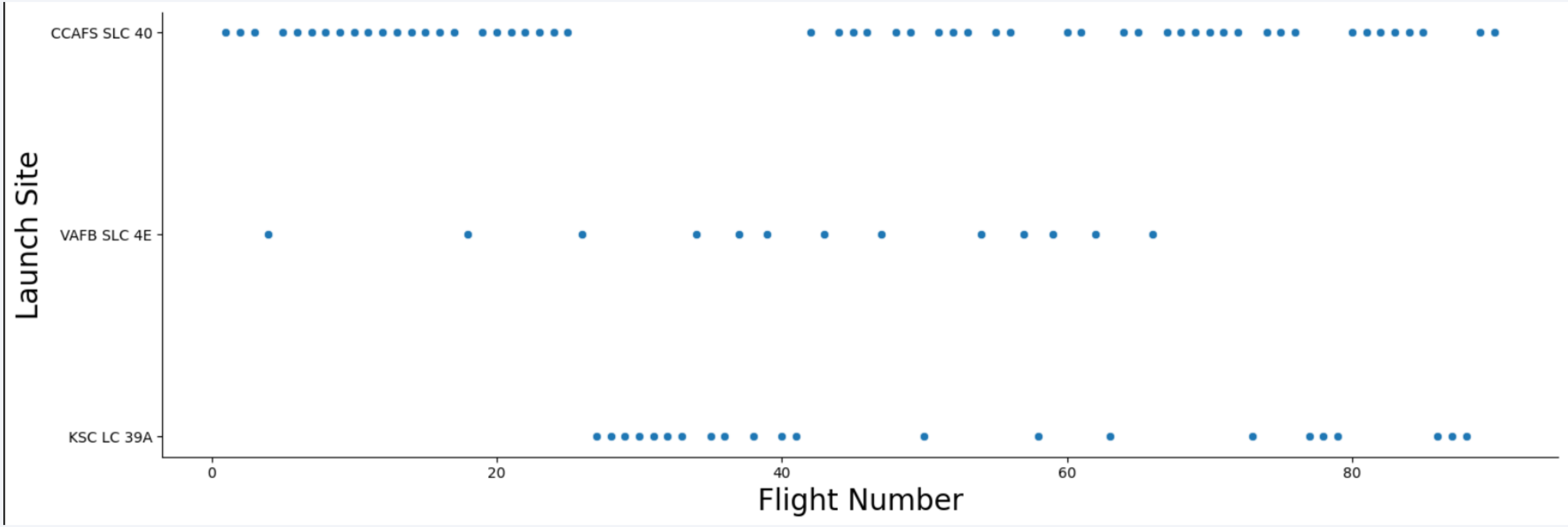
- ❑ **Exploratory Data Analysis:**
- ✓ Launch success has demonstrated a noticeable improvement over time.
- ✓ KSC LC-39A emerges as the landing site with the highest success rate.
- ✓ Orbits ES-L1, GEO, HEO, and SSO exhibit an impressive 100% success rate.
- ❑ **Visual Analytics:**
- ✓ Most launch sites are strategically positioned near the equator and in close proximity to the coast.
- ✓ Launch sites are strategically located—sufficiently far from potential damage sources (city, highway, railway) in the event of a failed launch, while remaining close enough to facilitate logistical support.
- ❑ **Predictive Analytics:**
- ✓ The Decision Tree model emerged as the most effective predictive model for the dataset, showcasing its prowess in discerning patterns and predicting outcomes

Section 2

Insights drawn from EDA

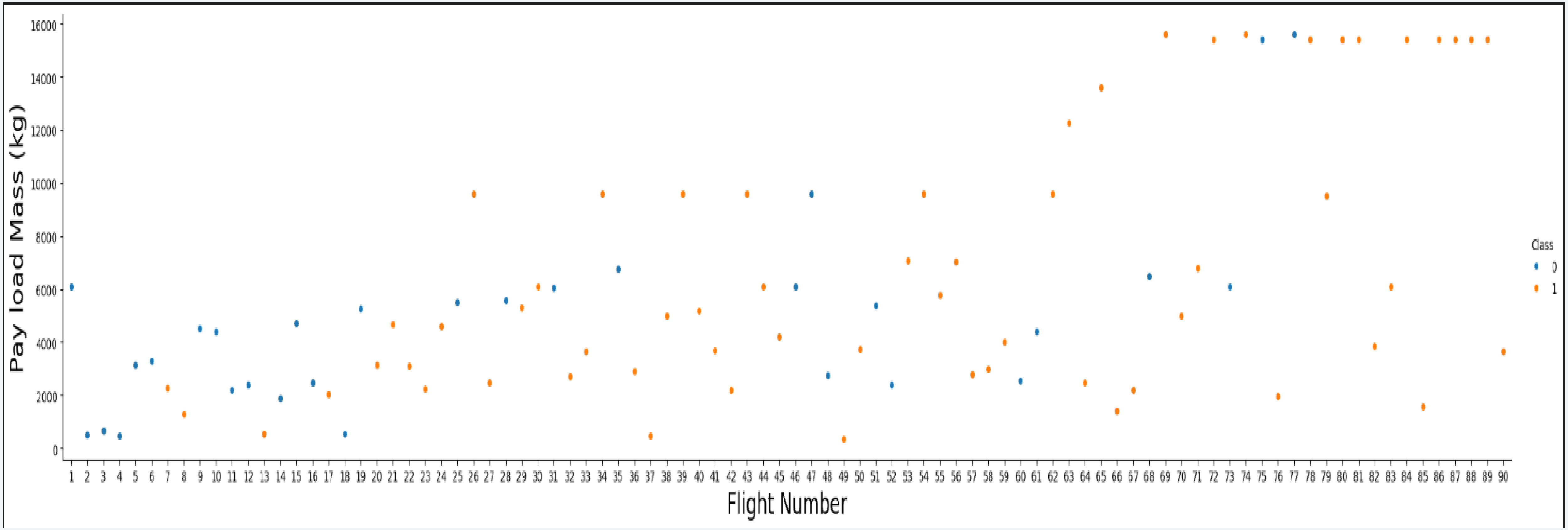
Flight Number vs. Launch Site

- Based on the below screenshot, we observe a positive correlation between the number of flights at a launch site and its success rate. As the flight amount increases, the success rate at the launch site also tends to rise.



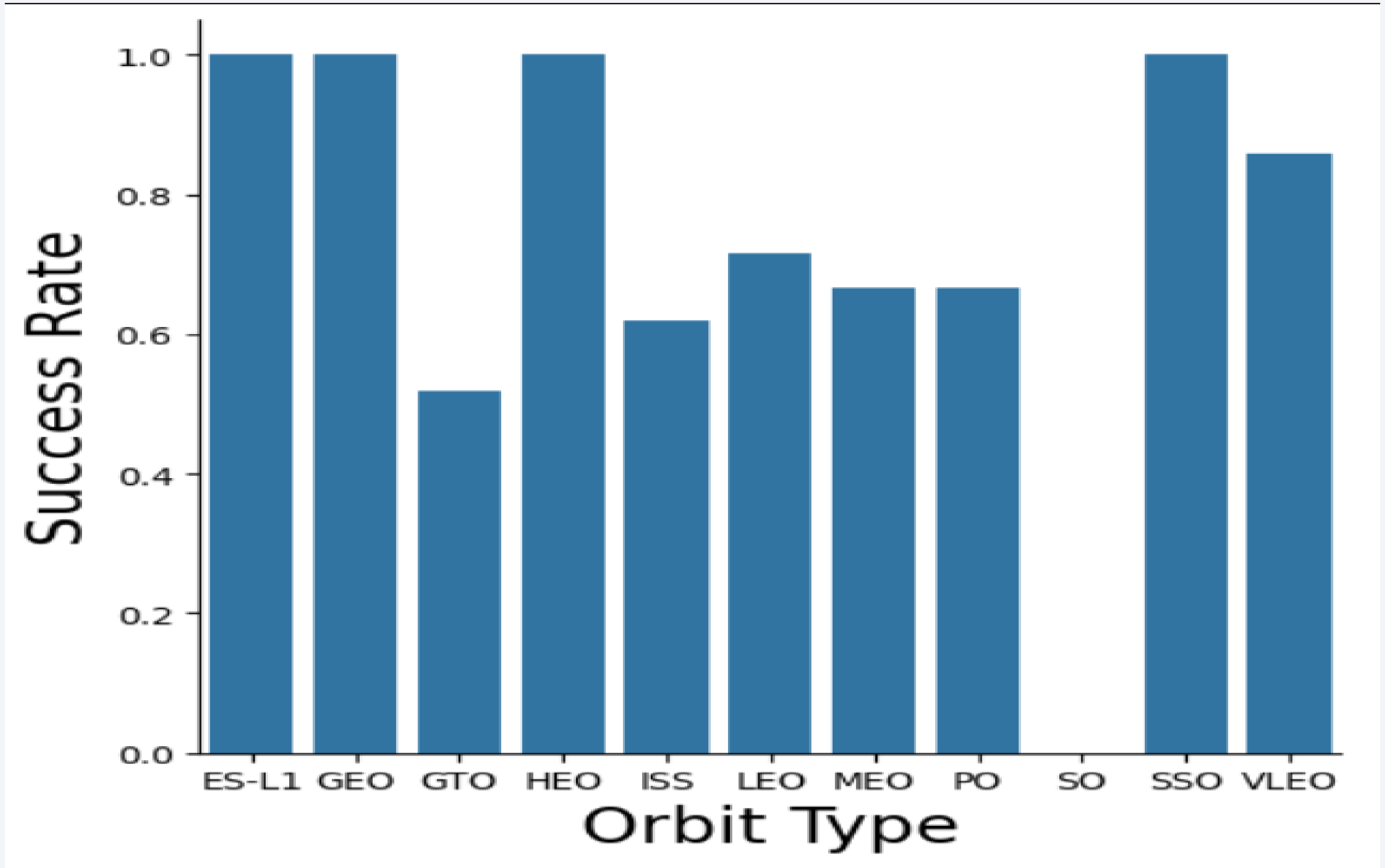
Payload vs. Launch Site

We see that different launch sites have different success rates. CCAFS LC-40 has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.



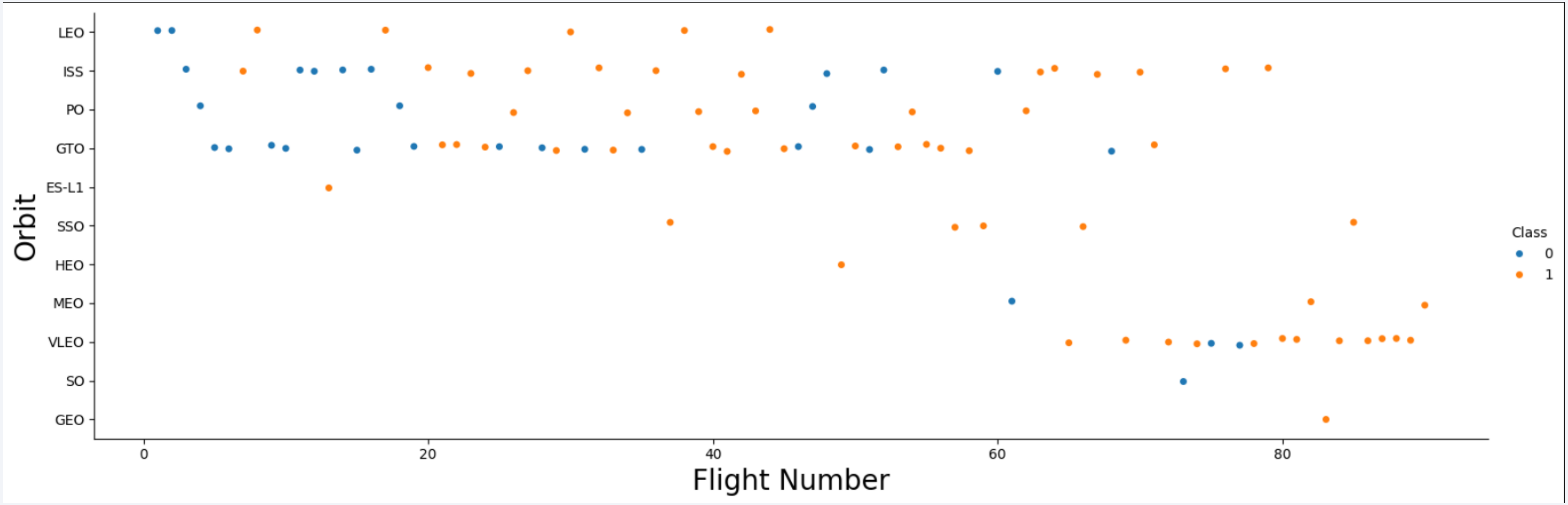
Success Rate vs. Orbit Type

The chart highlights that ES-L1, GEO, HEO, SSO, and VLEO orbits consistently exhibit the highest success rates.



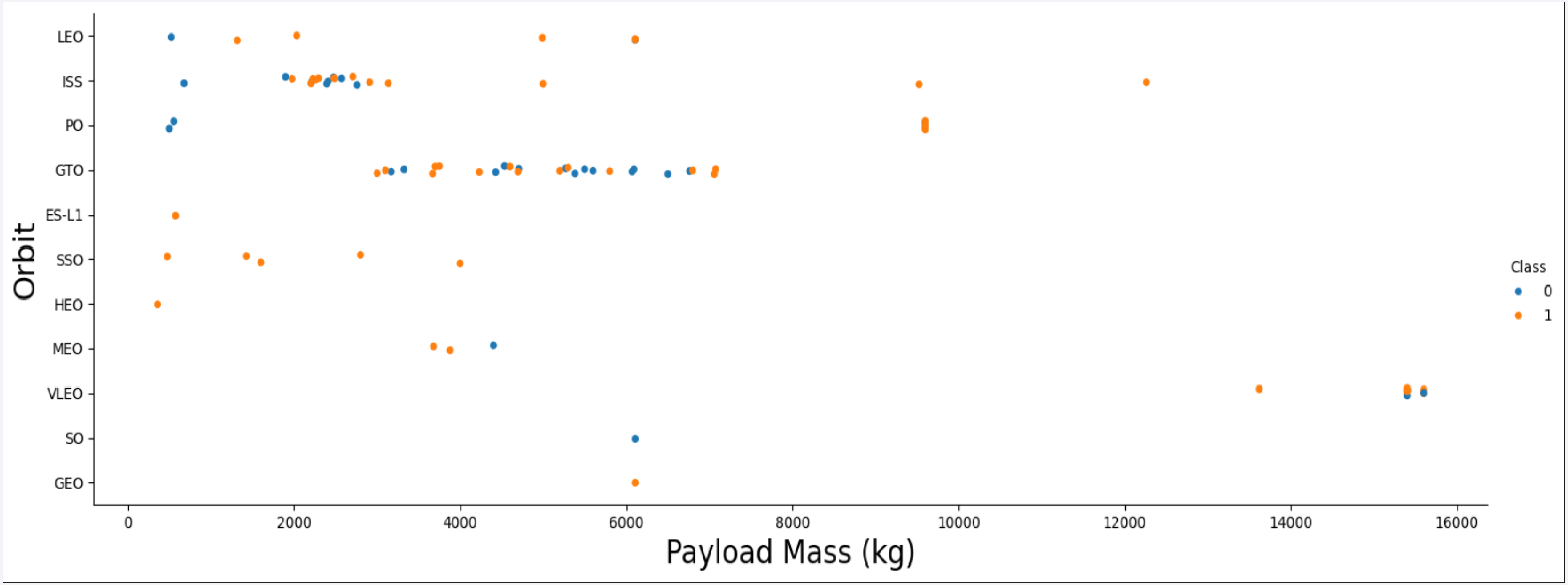
Flight Number vs. Orbit Type

- The chart below indicates that in the LEO orbit, success seems to be correlated with the number of flights. However, in the GTO orbit, there appears to be no discernible relationship between flight number and success.



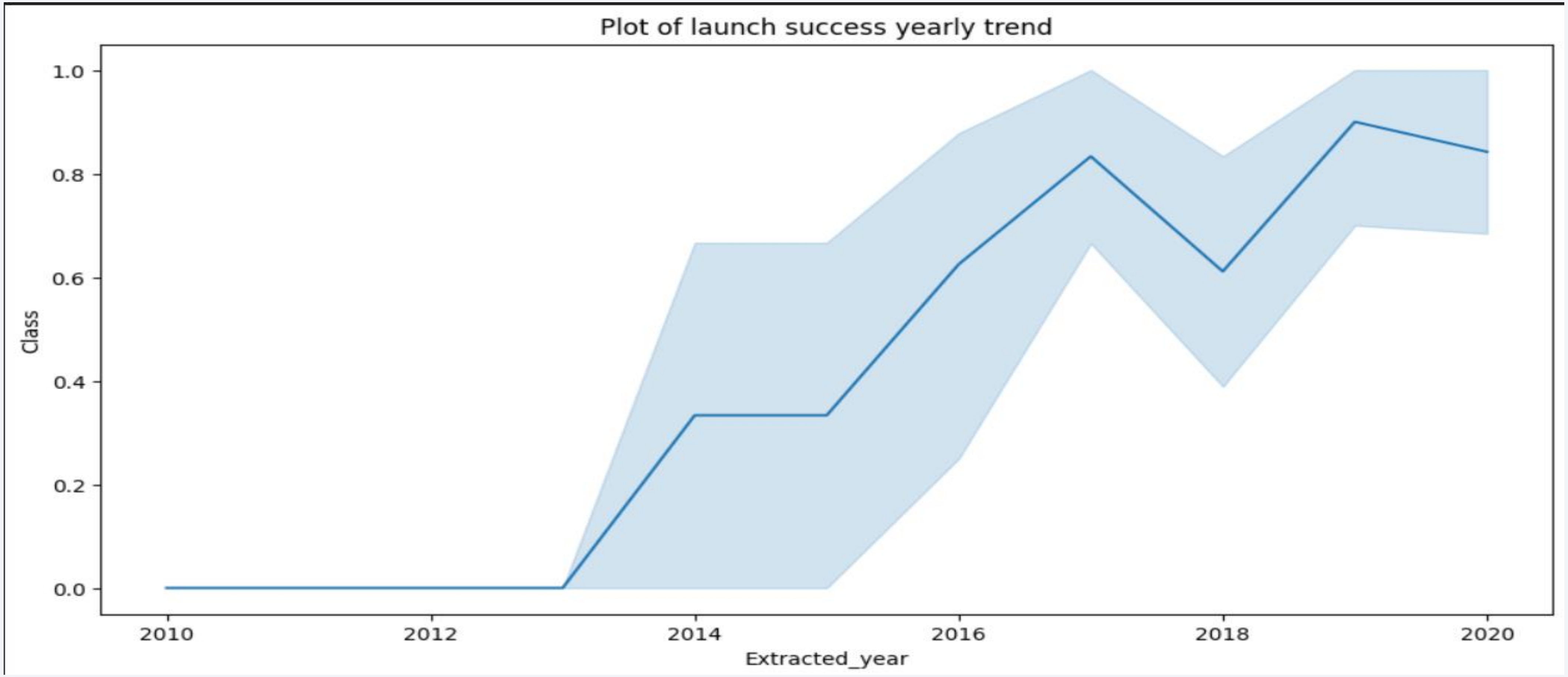
Payload vs. Orbit Type

- The chart below suggests that heavier payloads tend to result in a higher successful landing or positive landing rate, particularly for Polar, LEO, and ISS orbits. However, distinguishing this trend for GTO is challenging, as both positive landing rates and negative landing (unsuccessful mission) instances are present in this orbit.



Launch Success Yearly Trend

- Based on the chart below, we can observe a continuous increase in the success rate from 2013 to 2017, with stability noted in 2014. Following 2015, the success rate experienced a noticeable upward trend.



All Launch Site Names

- In our analysis, we utilized the keyword DISTINCT to selectively display only the unique launch sites from the SPACEXTBL data, providing a clearer and focused

```
select_table_query = '''
SELECT DISTINCT LaunchSite
FROM SPACEXTBL;
'''

# Execute the create table query
df = pd.read_sql_query(select_table_query, con)

# Display the DataFrame
print(df)
```

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

Launch Site Names Begin with 'KSC'

- In our presentation, we employed the keyword LIKE 'KSC%' to selectively showcase launch sites that commence with 'KSC' from the SpaceX dataset. This allowed us to refine the display and highlight specific launch site details with precision.

```
Start_with_KSC = '''
        SELECT top 5 *
        FROM SPACEXTBL
        WHERE LaunchSite LIKE 'KSC%';
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(Start_with_KSC, con)

# Display the DataFrame
print(df)
```

```
         Date      Time  BoosterVersion  LaunchSite           Payload  \
0  2017-02-19  14:39:00   F9 FT B1031.1  KSC LC-39A  SpaceX CRS-10
1  2017-03-16   6:00:00     F9 FT B1030  KSC LC-39A    EchoStar 23
2  2017-03-30  22:27:00  F9 FT  B1021.2  KSC LC-39A         SES-10
3  2017-05-01  11:15:00   F9 FT B1032.1  KSC LC-39A        NROL-76
4  2017-05-15  23:21:00     F9 FT B1034  KSC LC-39A  Inmarsat-5 F4

   PayloadMassKG      Orbit    Customer MissionOutcome      LandingOutcome
0           2490  LEO (ISS)  NASA (CRS)        Success  Success (ground pad)
1           5600        GTO    EchoStar        Success           No attempt
2           5300        GTO         SES        Success  Success (drone ship)
3           5300        LEO         NRO        Success  Success (ground pad)
4           6070        GTO   Inmarsat        Success           No attempt
```

Total Payload Mass

- In the course of our analysis, we utilized the Sum function to meticulously compute the total payload carried by boosters associated with NASA, specifically those categorized under 'NASA (CRS),' from the SpaceX dataset.

```
NASA_CRS = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SPACEXTBL
        WHERE Customer = 'NASA (CRS)'
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(NASA_CRS, con)

# Display the DataFrame
print(df)
```

```
   Total_PayloadMass
0              45596
```

Average Payload Mass by F9 v1.1

- we employed the average function to calculate the mean payload mass carried by booster version F9 v1.1, specifically focusing on instances where the Booster Version is 'F9 v1.1' in the SpaceX dataset.

```
avg_booster = '''
        SELECT avg(PayloadMassKG) as Avg_PayloadMass
        FROM SPACEXTBL
        WHERE BoosterVersion = 'F9 v1.1';
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(avg_booster, con)

# Display the DataFrame
print(df)
```

[42]

```
   Avg_PayloadMass
0             2928
```

First Successful Ground Landing Date

- we applied the minimum function to identify the dates associated with the initial successful landing outcomes on drone ships.

```
seccussful_landing_date  = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SPACEXTBL
        WHERE LandingOutcome = 'Success (ground pad)';
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(seccussful_landing_date, con)

# Display the DataFrame
print(df)
```

```
  FirstSuccessfull_landing_date
0                    2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- we compiled a list featuring the names of boosters that exhibited a successful landing on a drone ship while simultaneously carrying a payload mass greater than 4000 kg but less than 6000 kg.

```
BoosterVersion_name = '''
        SELECT BoosterVersion
         FROM SPACEXTBL
         WHERE LandingOutcome = 'Success (drone ship)'
             AND PayloadMassKG between 4000 and 6000;
         '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(BoosterVersion_name, con)

# Display the DataFrame
print(df)
[44]
```

```
...    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

- we utilized the count function to systematically compute the total number of both successful and unsuccessful mission outcomes.

```
missionOutcomeRate = '''
        select (SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SPACEXTBL
        WHERE MissionOutcome LIKE 'Success%')SuccessOutcome ,
        (SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SPACEXTBL
        WHERE MissionOutcome LIKE 'Failure%')FailureOutcome;
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(missionOutcomeRate, con)

# Display the DataFrame
print(df)
```

[46]

```
   SuccessOutcome  FailureOutcome
0             100               1
```

Boosters Carried Maximum Payload

- we compiled a list featuring the names of boosters that have carried the maximum payload mass. This was achieved by employing a subquery in conjunction with the max function, allowing us to present this information with precision and clarity.

```python
mae_of_max_payload = '''
        SELECT BoosterVersion,PayloadMassKG
        FROM SPACEXTBL
        WHERE PayloadMassKG = (
                        SELECT MAX(PayloadMassKG)
                        FROM SPACEXTBL
                        )
        ORDER BY BoosterVersion;
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(mae_of_max_payload, con)

# Display the DataFrame
print(df)
```

```
   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

- we showcased a curated list of records that displays the month names, successful landing outcomes on ground pads, associated booster versions, and launch sites specifically for the months within the year 2017. This was achieved through the application of the year and month functions, allowing us to present this information with clarity and relevance.

```
order_by_month = '''
        SELECT YEAR(Date) as Year , month(Date) as Month,BoosterVersion, LaunchSite, LandingOutcome
        FROM SPACEXTBL
        WHERE LandingOutcome = 'Success (ground pad)'
        AND YEAR(Date) = '2017'
        order by month(Date);
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(order_by_month, con)

# Display the DataFrame
print(df)
```

```
   Year  Month  BoosterVersion     LaunchSite        LandingOutcome
0  2017      2   F9 FT B1031.1    KSC LC-39A  Success (ground pad)
1  2017      5   F9 FT B1032.1    KSC LC-39A  Success (ground pad)
2  2017      6   F9 FT B1035.1    KSC LC-39A  Success (ground pad)
3  2017      8   F9 B4 B1039.1    KSC LC-39A  Success (ground pad)
4  2017      9   F9 B4 B1040.1    KSC LC-39A  Success (ground pad)
5  2017     12  F9 FT  B1035.2  CCAFS SLC-40  Success (ground pad)
```

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

- we thoughtfully ranked the count of landing outcomes, including Failure (drone ship) or Success (ground pad), within the specified date range from 2010-06-04 to 2017-03-20. The ranking was performed in descending order, providing a nuanced perspective on the distribution of landing outcomes during this timeframe.

```
select_table_query = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SPACEXTBL
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC;
        '''
# Execute the SQL query and create a DataFrame
df = pd.read_sql_query(select_table_query, con)

# Display the DataFrame
print(df)
```

[56]

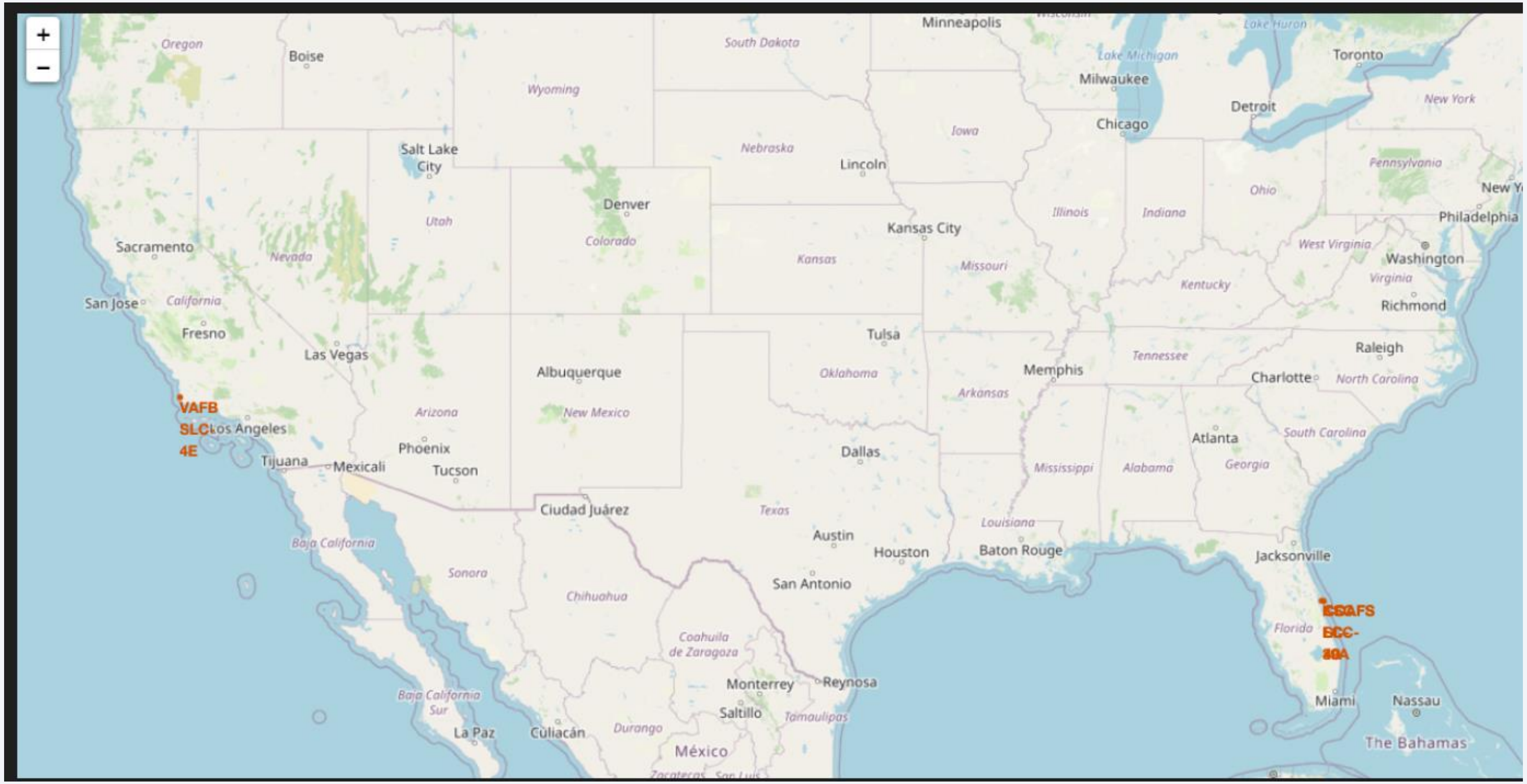
```
           LandingOutcome
0              No attempt  10
1      Failure (drone ship)   5
2      Success (drone ship)   5
3     Success (ground pad)   3
4        Controlled (ocean)   3
5      Uncontrolled (ocean)   2
6       Failure (parachute)   2
7    Precluded (drone ship)   1
```

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

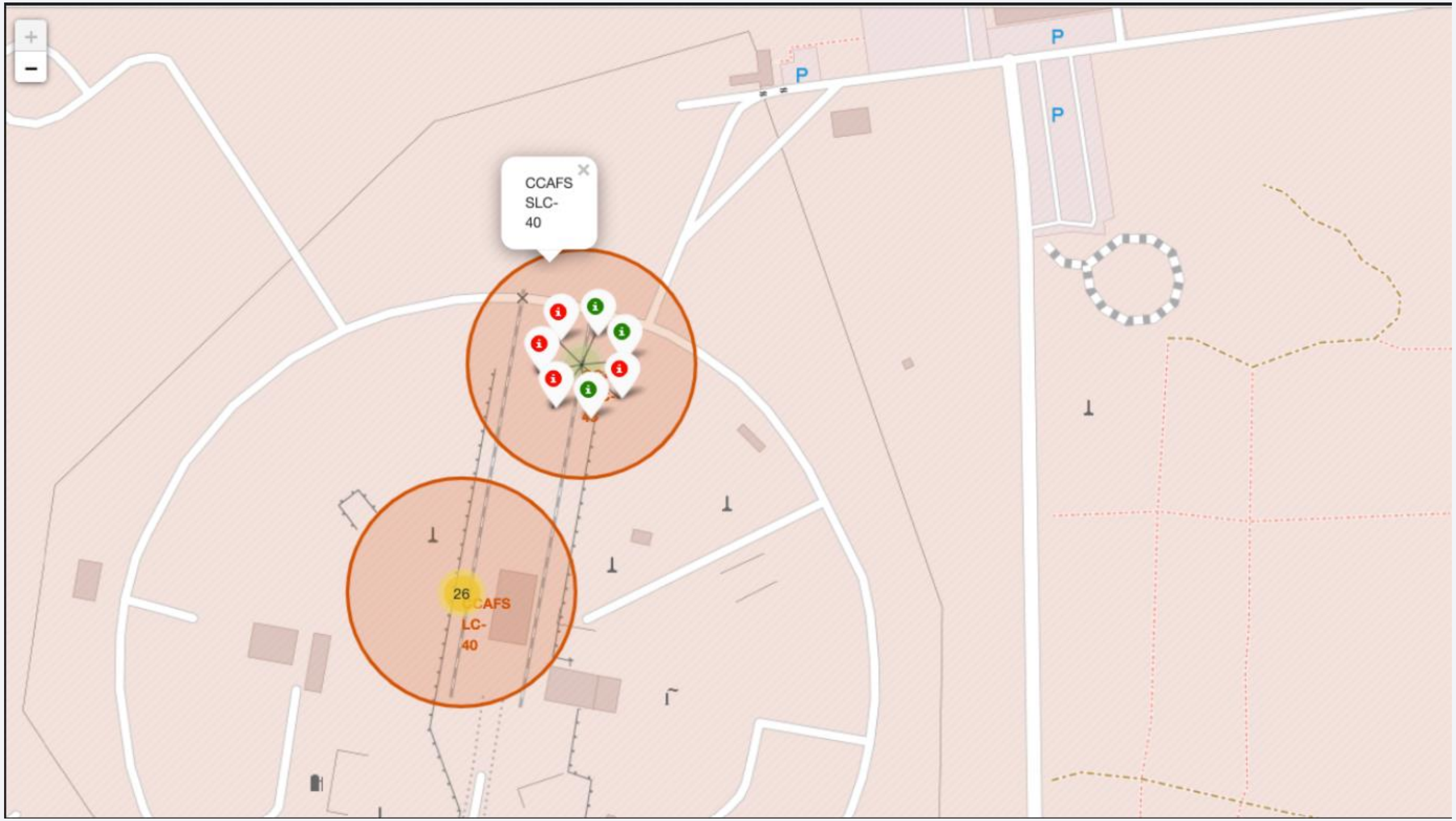
- it's evident that SpaceX launch sites are strategically distributed along both the east and west coasts of the United States.



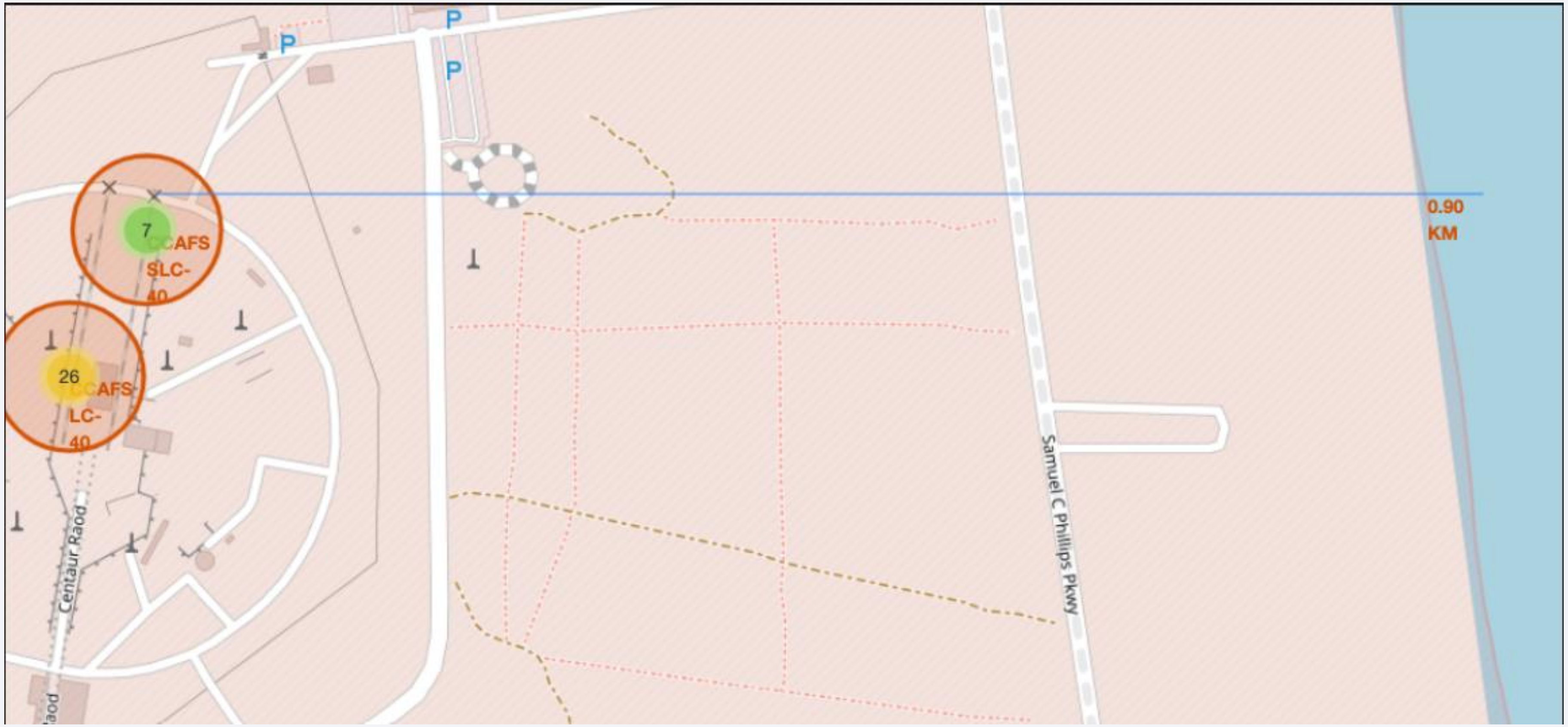
SpaceX-All launch sites

<Folium Map Screenshot 2>

- Green markers denote successful launches.
- Red markers indicate unsuccessful launches.



<Folium Map Screenshot 3>

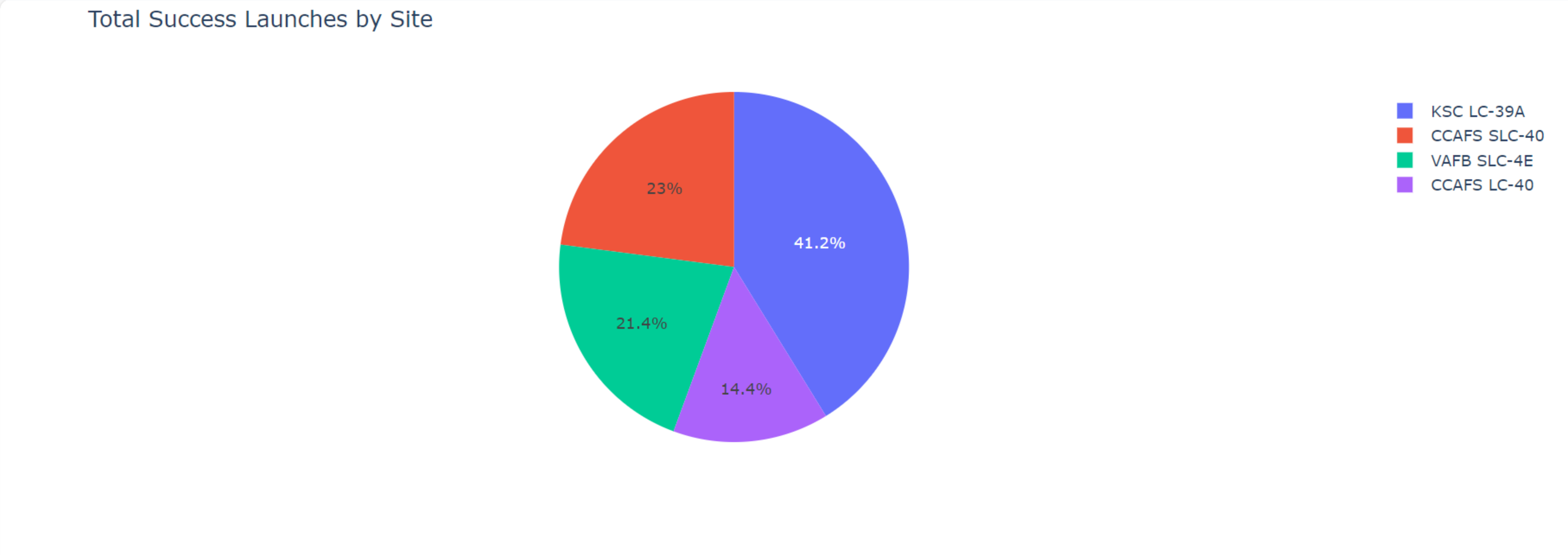


Section 4

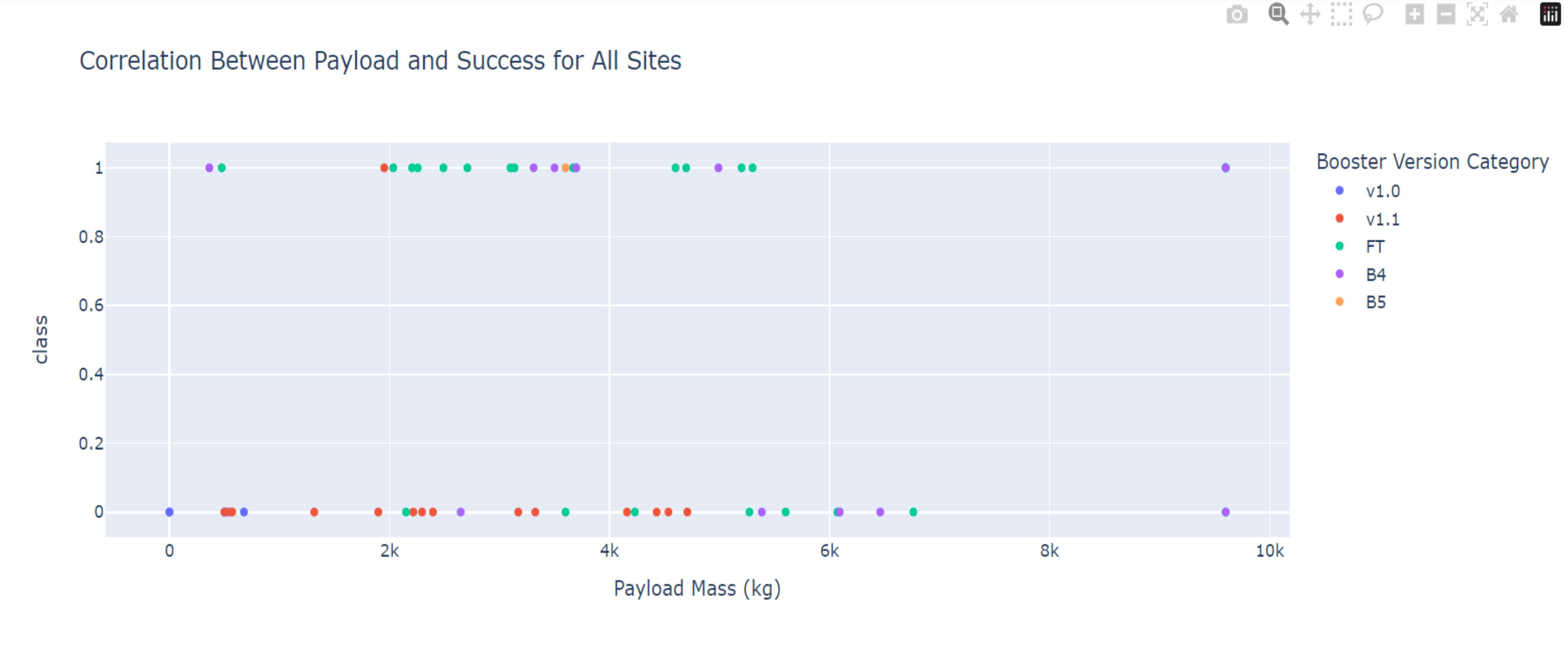
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- It's notable that KSC LC 39-A stands out with the highest number of successful launches compared to other sites



<Dashboard Screenshot 2>



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

```
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_)
```

57]

```
Best model is DecisionTree with a score of 0.875
Best params is : {'criterion': 'gini', 'max_depth': 14, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
```

Confusion Matrix

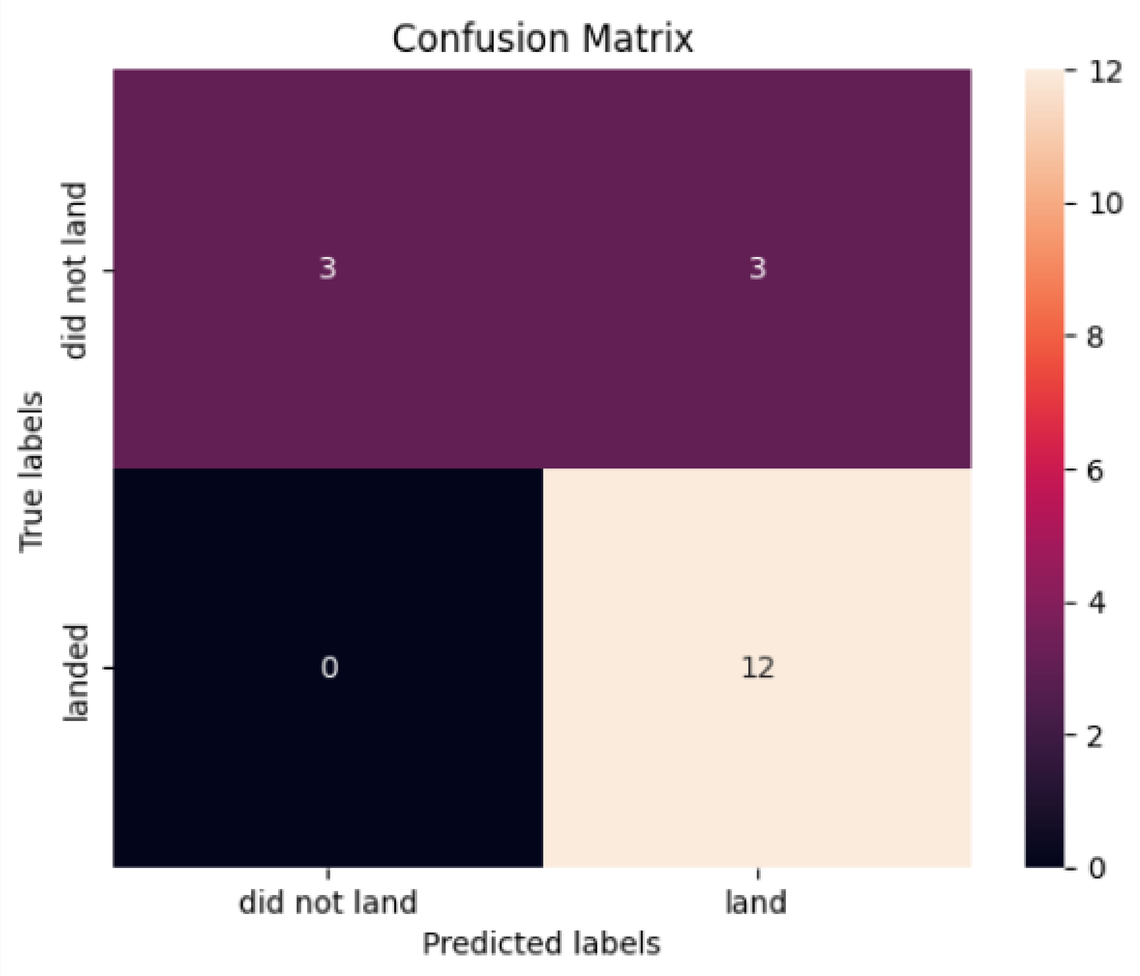
- Confusion Matrix Outputs:

✓ 12 True positive

✓ 3 True negative

✓ 3 False positive

✓ 0 False Negative



Conclusions

- **Flight Amount Impact:** A noticeable correlation is observed between the volume of flights at a launch site and its success rate. The larger the flight amount, the greater the success rate.
- **Temporal Success Trend:** From 2013 to 2020, there is a consistent increase in the launch success rate, highlighting a positive trend in SpaceX's operational efficiency.
- **Orbit Success Rates:** Orbits ES-L1, GEO, HEO, SSO, and VLEO consistently demonstrated the highest success rates, showcasing their reliability in successful launches.
- **Launch Site Performance:** KSC LC-39A emerges as the leading launch site, boasting the highest number of successful launches compared to others.
- **Optimal Machine Learning Model:** The Decision Tree classifier stands out as the most effective machine learning algorithm for predicting landing outcomes in this specific task.

In summary, the project provides valuable insights into the factors influencing launch success, temporal trends, orbit-specific performance, and the optimal model for predictive analytics. These findings contribute to a holistic understanding of SpaceX's mission success dynamics.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!