Winning Space Race with Data Science

Mahdi Habibi
November 2023

Outline

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

Executive Summary

Summary of methodologies

Data Handling: Combined data collection, cleaning, and transformation for robust dataset preparation.
Exploratory Analysis: Utilized statistical techniques and visual analytics for in-depth data exploration.
Predictive Modeling: Employed advanced algorithms for predictive analysis, ensuring thorough model evaluation.
Interactive Tools: Integrated Folium for geographic insights and Plotly Dash for dynamic visual dashboards.

Summary of all results

Visual Discoveries: Uncovered key insights and patterns through comprehensive visualizations.
SQL Insights: Derived critical findings and trends from complex SQL queries.
Geospatial Trends: Revealed significant geographical patterns using interactive Folium maps.
Dashboard Interactivity: Enabled insightful user interactions via Plotly Dash.
Predictive Accuracy: Achieved substantial predictive performance, highlighting significant classifications.
Conclusion: Presented overarching findings, implications, and avenues for future exploration.

Introduction

Falcon 9: Revolutionising Space Launches

- **Innovation in Focus**: SpaceX's Falcon 9 rocket, leading the charge in reducing space launch costs.
- **The Power of Reusability**: Emphasising the cost-saving impact of reusing the first stage.

Predictive Challenge: Landing Success

- **Crucial Question**: Can we accurately predict the successful landing of Falcon 9’s first stage?
- **Why It Matters**: Insights into financial implications and competitive dynamics in the space industry.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - **API Utilisation**
  - **Web Scraping**
- Perform data wrangling
  - **Normalisation and Parsing**
  - **Filtering and Cleaning**
  - **Handling Nulls**
- Perform exploratory data analysis (EDA) using visualisation and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - **Model Building**
  - **Tuning and Evaluation**

Data Collection

- **API Utilisation**: Collected SpaceX launch data using the SpaceX REST API, specifically from the endpoint **api.spacexdata.com/v4/launches/past**.
- **Web Scraping**: Employed Python BeautifulSoup package to scrape HTML tables containing Falcon 9 launch records from relevant Wiki pages.

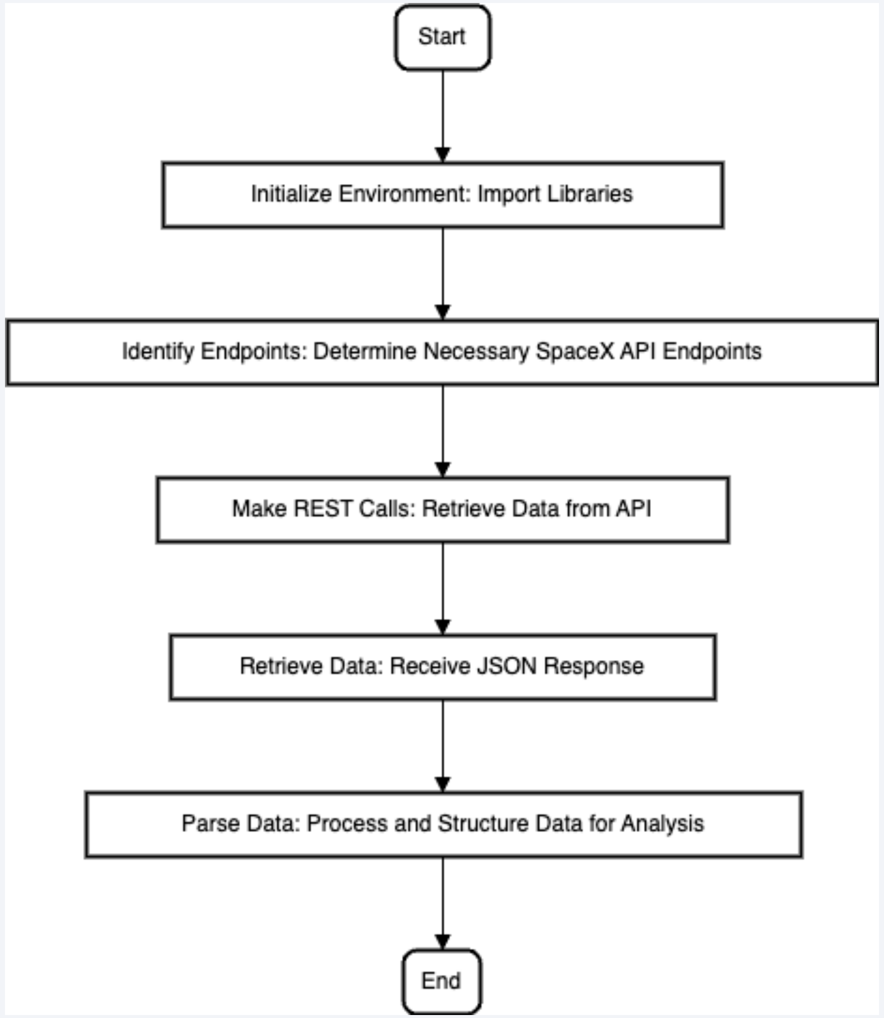
Data Collection – SpaceX API

Overview of Data Collection using SpaceX API

- **Initialisation:** Setting up the environment and importing necessary libraries (e.g., `requests`, `json`).
- **API Endpoint Identification:** Identifying the SpaceX API endpoints relevant for the project (e.g., launches, rockets, etc.).
- **REST Calls:** Making REST calls to SpaceX API.
- **Data Retrieval:** Retrieving data in JSON format.
- **Data Parsing:** Parsing and structuring the data for analysis.

GitHub Repository

- URL: [GitHub Repository for Data Collection with SpaceX API]



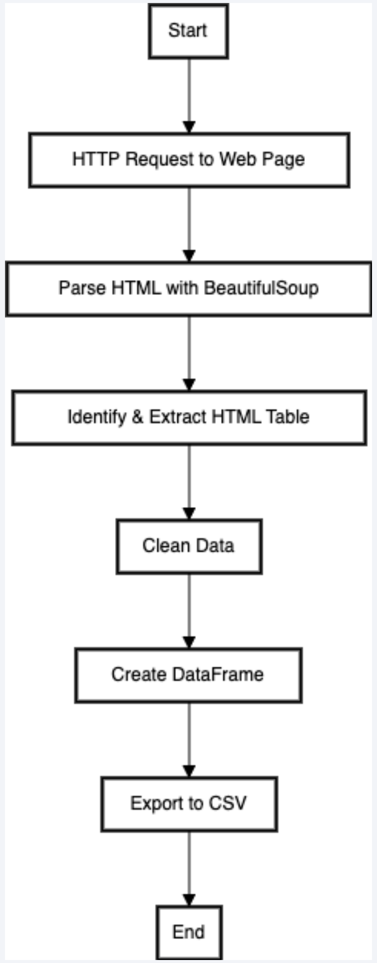
Data Collection - Scraping

Overview of Data Collection using Web Scraping

HTTP Request: Sending a request to access the web page (e.g., Wikipedia page).
BeautifulSoup Parsing: Utilising BeautifulSoup to parse the HTML content of the page.
HTML Table Extraction: Identifying and extracting the relevant HTML table containing data.
Data Cleaning: Cleaning and formatting the extracted data for consistency and readability.
Data Frame Creation: Converting the cleaned data into a structured pandas DataFrame.
CSV Export: Exporting the DataFrame into a CSV file for further use and analysis.

GitHub Repository

URL: [GitHub Repository for Data Collection with Webscraping]



Data Wrangling

Overview of Data Wrangling Process

- **Data Loading:** Importing the raw data using pandas' `read_csv` function.
- **Data Inspection:** Assessing the first few rows of the data with the `head()` method to get an initial understanding.
- **Handling Missing Values:**
  - Identifying missing values using `isnull()` and `sum()`.
  - Filling missing values in 'PUBCHEM_ACTIVITY_OUTCOME' with 'Inactive'.
  - Dropping rows where 'PUBCHEM_CID' is missing.
- **Data Type Conversion:**
  - Converting 'PUBCHEM_CID' from float to integer.
- **Data Filtering:**
  - Filtering the data based on the condition that 'PUBCHEM_ACTIVITY_OUTCOME' must be 'Active'.
- **Data Extraction:**
  - Extracting the 'PUBCHEM_CID' column to get the list of active compounds.
- **Data Export:**
  - Exporting the final list of active compounds into a CSV file using `to_csv`.

GitHub Repository

URL: GitHub Repository for Data Wrangling Process

EDA with Data Visualization

GitHub Repository
URL: [GitHub Repository for EDA with Data Visualization]

Objective: Perform Exploratory Data Analysis and Feature Engineering using `Pandas` and `Matplotlib`.

Key Steps and Findings:

- **Launch Site Analysis:**
  - Scatter plot and Cat plot: Relationship between `FlightNumber` and `LaunchSite`.
  - Findings: Different launch sites have varying success rates.
- **Payload and Launch Site Relationship:**
  - Scatter plot: `PayloadMass` vs. `LaunchSite`.
  - Key Observation: VAFB-SLC lacks heavy payload launches.
- **Orbit Type Success Rates:**
  - Bar chart: Success rate for each orbit type.
  - Findings: Identification of orbits with higher success rates.
- **Flight Number and Orbit Type:**
  - Scatter plot: `FlightNumber` vs. `Orbit`.
  - Insights: Correlation between flight number and success rate in different orbits.
- **Payload Mass and Orbit Type:**
  - Scatter plot: `PayloadMass` vs. `Orbit`.
  - Conclusion: Heavy payloads have more success in certain orbits.
- **Yearly Trend of Success:**
  - Line chart: Launch success yearly trend.
  - Trend Analysis: Increasing success rate over the years.

EDA with SQL

GitHub Repository

URL: GitHub Repository for EDA with SQL

Objective: Perform EDA using SQL queries to gain insights from the SpaceX dataset.

Key Steps and Findings:

- **Launch Site Analysis**:
  - Distinct Launch Sites: Identified unique launch sites in space missions.
  - Key Finding: Multiple launch sites with varying characteristics.
- **Launch Site Record Exploration**:
  - Records with 'CCA': Found records where launch sites begin with 'CCA'.
  - Observation: Specific pattern in launch site naming and categorisation.
- **Payload Analysis by Booster for NASA (CRS)**:
  - Total Payload Mass: Calculated total payload mass carried by boosters for NASA (CRS).
  - Insight: Helps in understanding payload capacity for specific customers.
- **Booster Version Payload Analysis**:
  - Average Payload Mass: Determined for booster version F9 v1.1.
  - Conclusion: Provides an average benchmark for payload capabilities of the booster.
- **First Successful Landing Outcome**:
  - Date of First Success: Identified when the first successful landing on the ground pad was achieved.
  - Milestone: Marks an important achievement in SpaceX's journey.
- **Successful Boosters in Specific Conditions**:
  - Boosters with Specific Success: Listed boosters with success in drone ship landings and specific payload mass criteria.
  - Finding: Highlights the capability of certain boosters under defined conditions.
- **Mission Outcome Analysis**:
  - Success and Failure Counts: Listed the total number of successful and failure mission outcomes.
  - Overview: Provides a general success-failure ratio.
- **Max Payload Booster Versions**:
  - Boosters with Max Payload: Identified the names of the booster versions carrying the maximum payload mass.
  - Key Insight: Understanding the maximum payload capacity achieved.
- **Detailed 2015 Drone Ship Failures Analysis**:
  - Failure Records in 2015: Displayed specific records with failure landing outcomes in drone ships for the year 2015.
  - Detailed Review: Allows a focused analysis on a particular year and failure type.
- **Landing Outcomes Ranking (2010-2017)**:
  - Ranked Landing Outcomes: Ranked the count of different landing outcomes between specific dates.
  - Trend Analysis: Provides a descending order view of landing outcomes.

Build an Interactive Map with Folium

Summary of Map Objects Added to the Folium Map:

- **Markers**: Placed at various locations to represent specific points of interest or data points.
- **Circles**: Used to visualise a certain radius or range around specific locations.
- **Polylines (Lines)**: Connects different markers, representing pathways, routes, or boundaries.

Reason for Adding These Objects:

- **Markers**: To highlight and provide information about specific locations, such as schools, restaurants, or landmarks.
- **Circles**: To illustrate proximity or coverage areas from a particular point, which can be useful in understanding accessibility or reach.
- **Polylines**: To show routes, connections, or boundaries between different locations, aiding in visualising relationships or paths.

GitHub Repository

URL: [GitHub Repository for Interactive Map with Folium]

Build a Dashboard with Plotly Dash

SpaceX Launch Dashboard Summary

1. Interactive Elements:

- Launch Site Dropdown: Filter data by launch sites.
- Payload Range Slider: Select payload mass range.

2. Visualisations:

- Success Pie Chart: Visualise launch success rates.
- Payload Scatter Chart: Explore payload mass vs. launch success correlation.

Purpose:

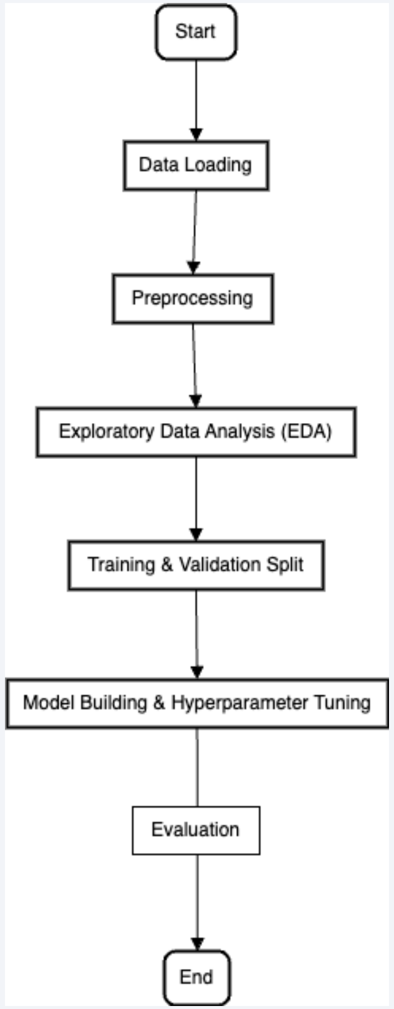
- Enhance user engagement and enable in-depth analysis of SpaceX launch data through interactive filters and informative visualisations.

GitHub Repository
URL: GitHub Repository for Plotly Dash

Predictive Analysis (Classification)

Key Phrases:

- Data Loading & Preprocessing
- Exploratory Data Analysis (EDA)
- Training & Validation Split
- Model Building & Hyperparameter Tuning
- Evaluation using Confusion Matrix & Accuracy Score
- Selection of Best Performing Model



GitHub Repository

URL: [GitHub Repository for Classification]

Results

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

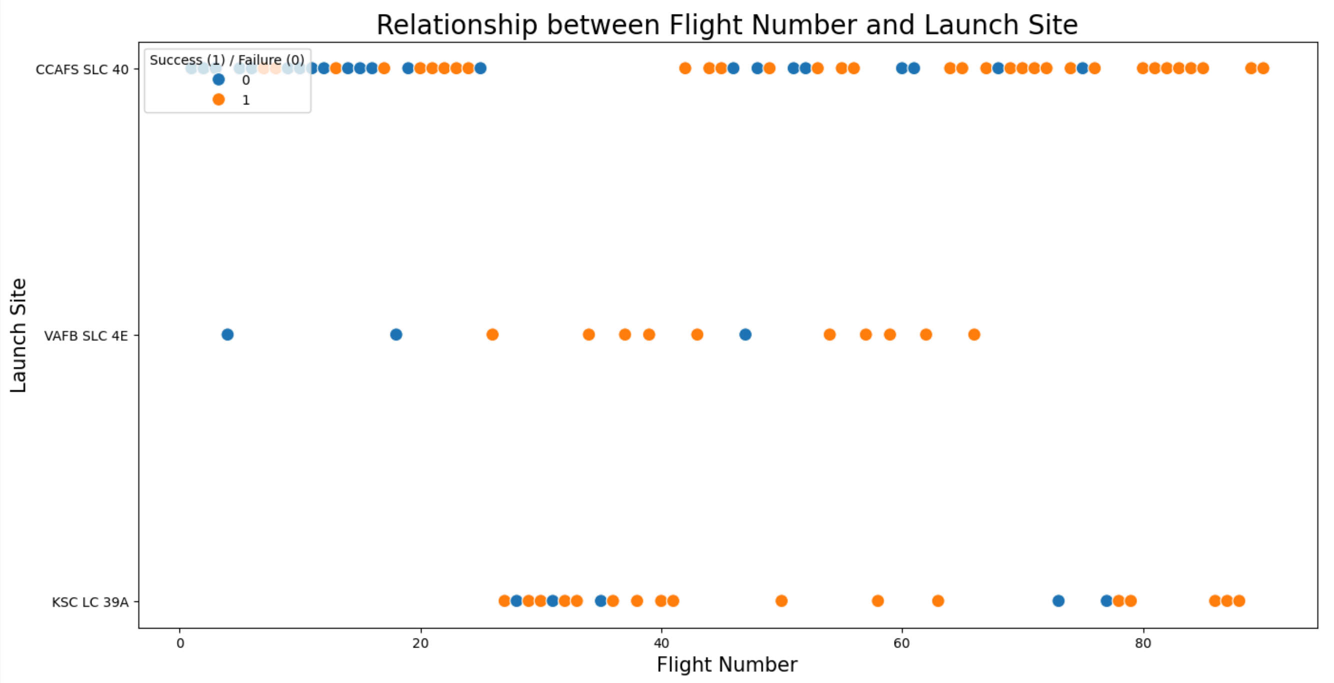
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

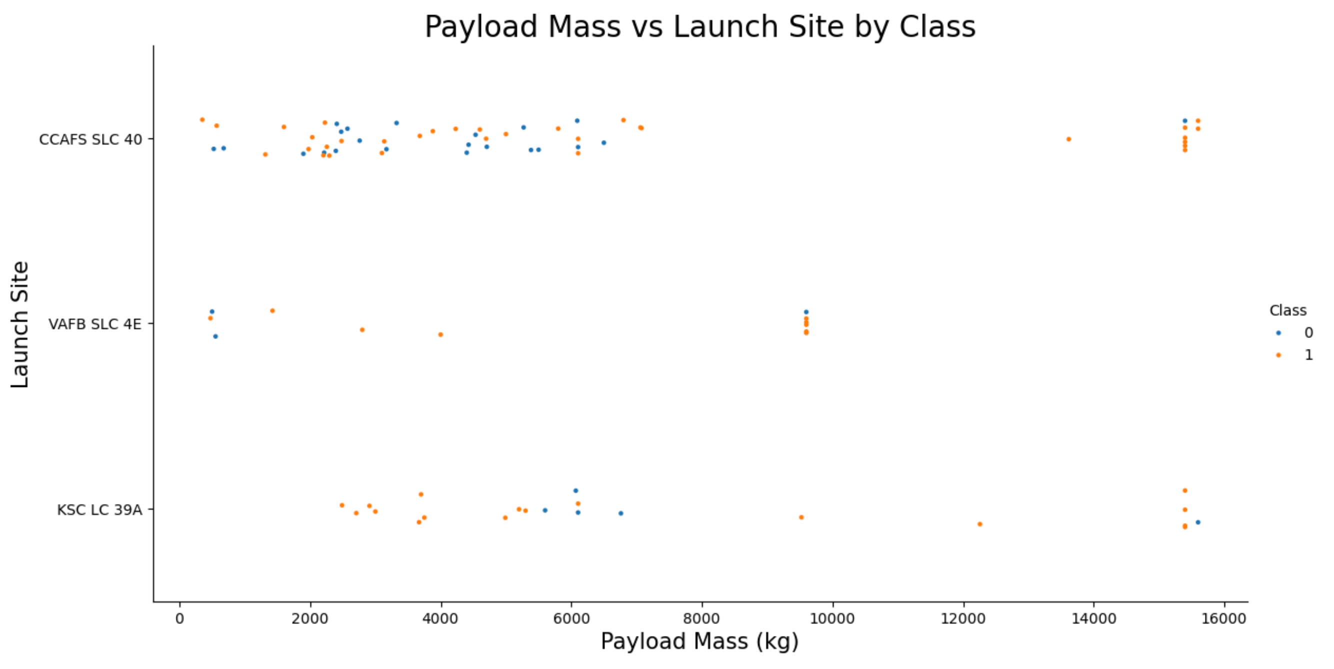
Scatter plot and Cat plot: Relationship between `FlightNumber` and `LaunchSite`.

Findings: Different launch sites have varying success rates.



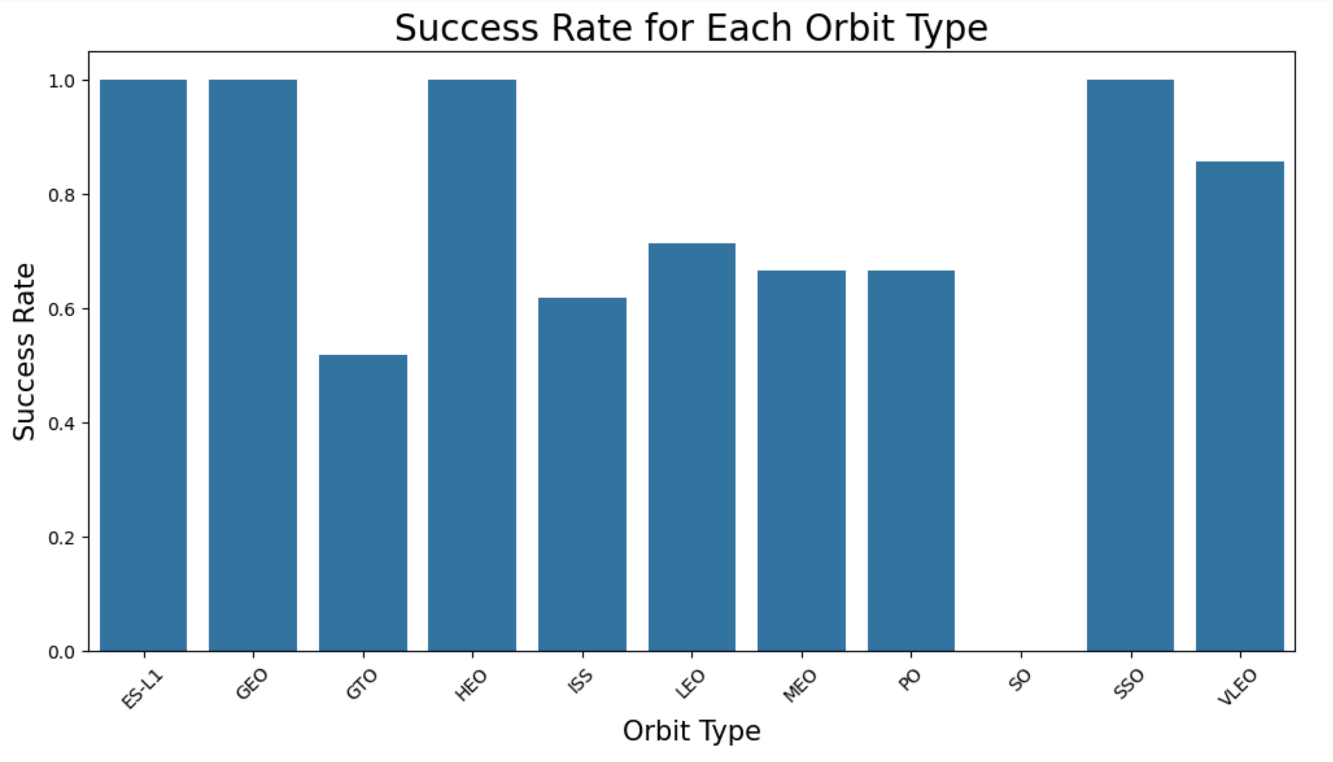
Payload vs. Launch Site

- **Scatter plot:** `PayloadMass` vs. `LaunchSite`.
- **Key Observation:** VAFB-SLC lacks heavy payload launches.



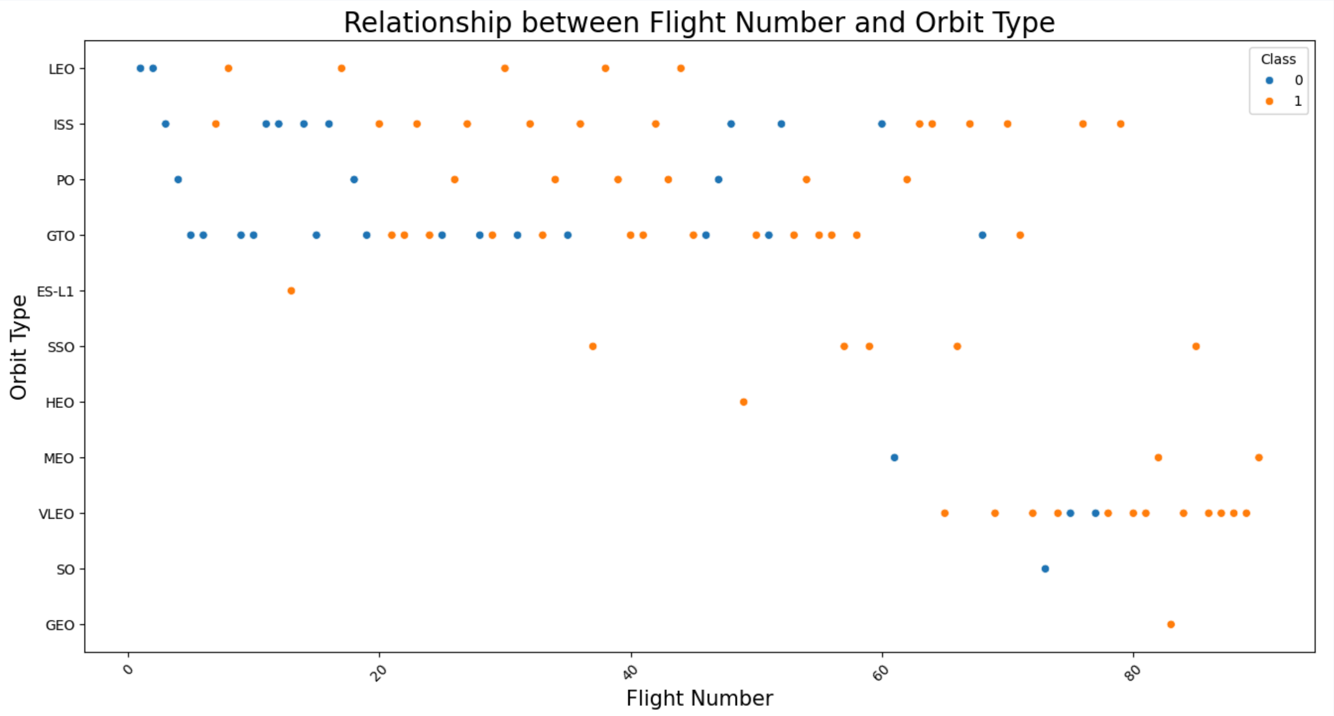
Success Rate vs. Orbit Type

- **Bar chart:** Success rate for each orbit type.
- **Findings:** Identification of orbits with higher success rates.



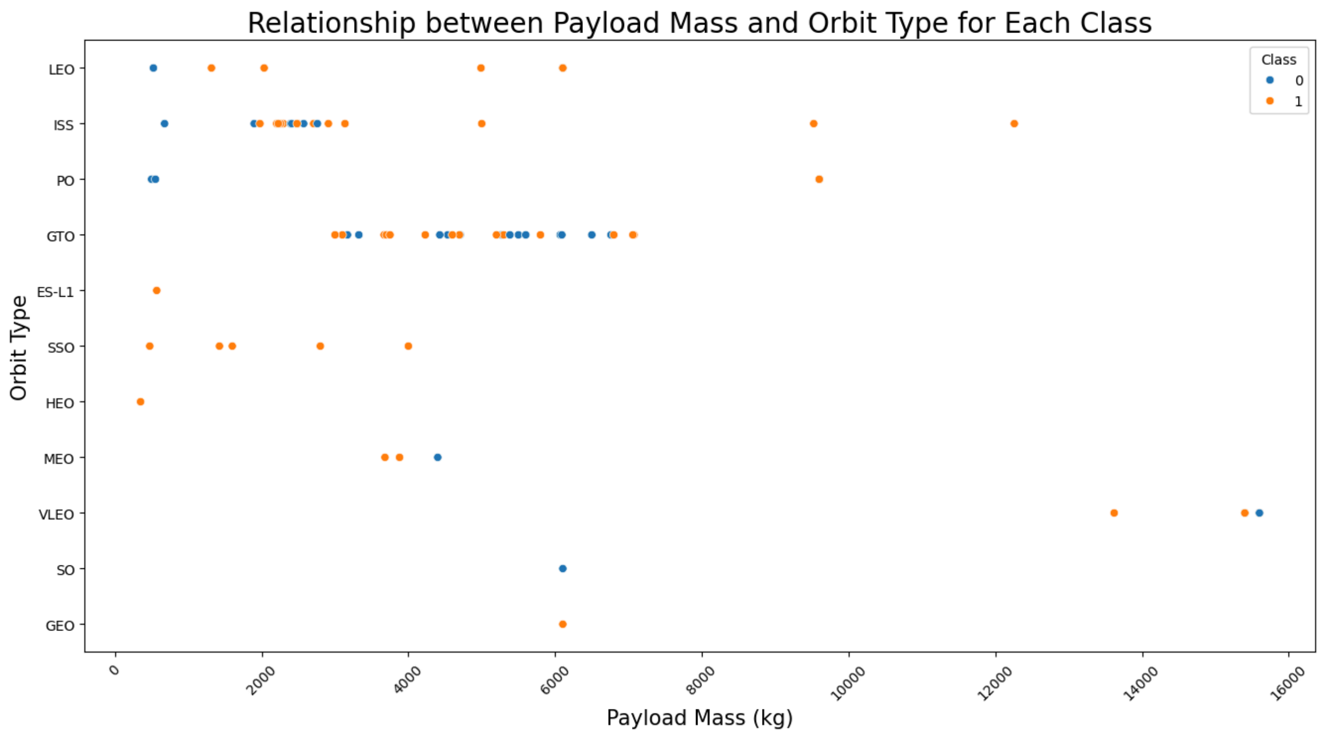
Flight Number vs. Orbit Type

- **Scatter plot:** `FlightNumber` vs. `Orbit`.
- **Insights:** Correlation between flight number and success rate in different orbits.



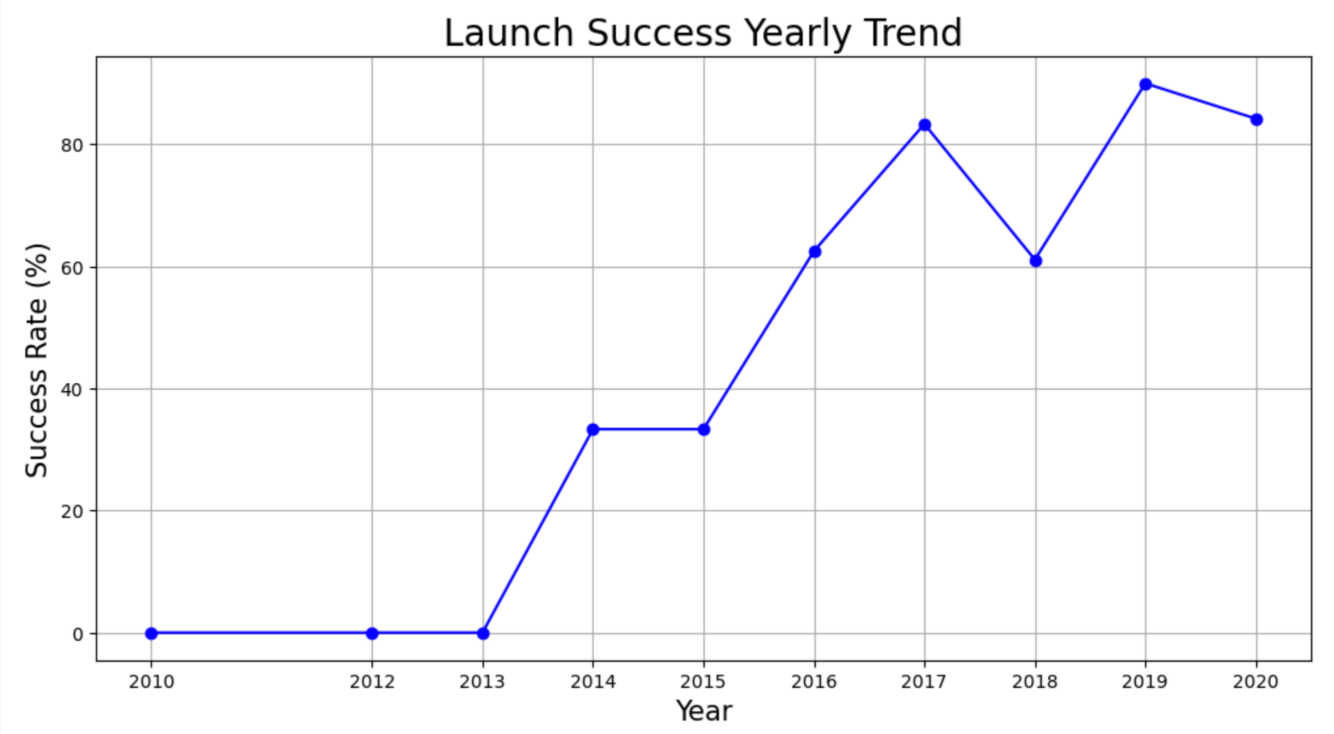
Payload vs. Orbit Type

- **Scatter plot:** `PayloadMass` vs. `Orbit`.
- **Conclusion:** Heavy payloads have more success in certain orbits.



Launch Success Yearly Trend

- **Line chart:** Launch success yearly trend.
- **Trend Analysis:** Increasing success rate over the years.



All Launch Site Names

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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

Launch Site Names Begin with 'CCA'

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

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

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as Average_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

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

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) as First_Successful_Landing FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

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

First_Successful_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE 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

```
%sql SELECT Mission_Outcome, COUNT(*) as Total_Outcomes FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

```
 * 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

```
%sql SELECT \
    substr(Date, 6, 2) as Month, \
    Booster_Version, \
    Launch_Site, \
    Landing_Outcome \
FROM \
    SPACEXTABLE \
WHERE \
    substr(Date, 0, 5) = '2015' \
    AND Landing_Outcome = 'Failure (drone ship)'
```

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

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

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

```
%sql SELECT \
    Landing_Outcome, \
    COUNT(Landing_Outcome) as Outcome_Count \
FROM \
    SPACEXTBL \
WHERE \
    Date BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY \
    Landing_Outcome \
ORDER BY \
    Outcome_Count DESC;
```

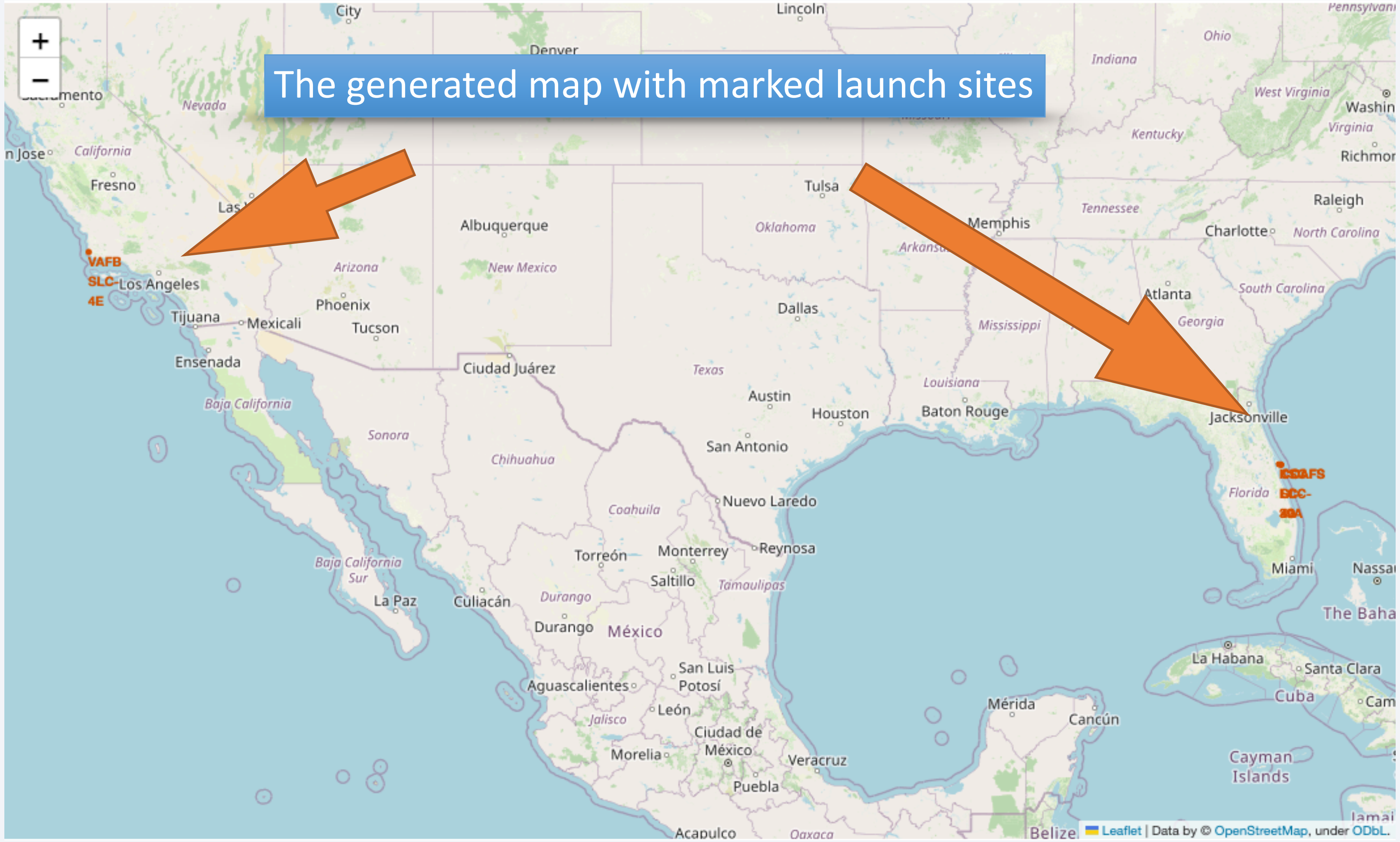
```
 * sqlite:///my_data1.db
Done.
```

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

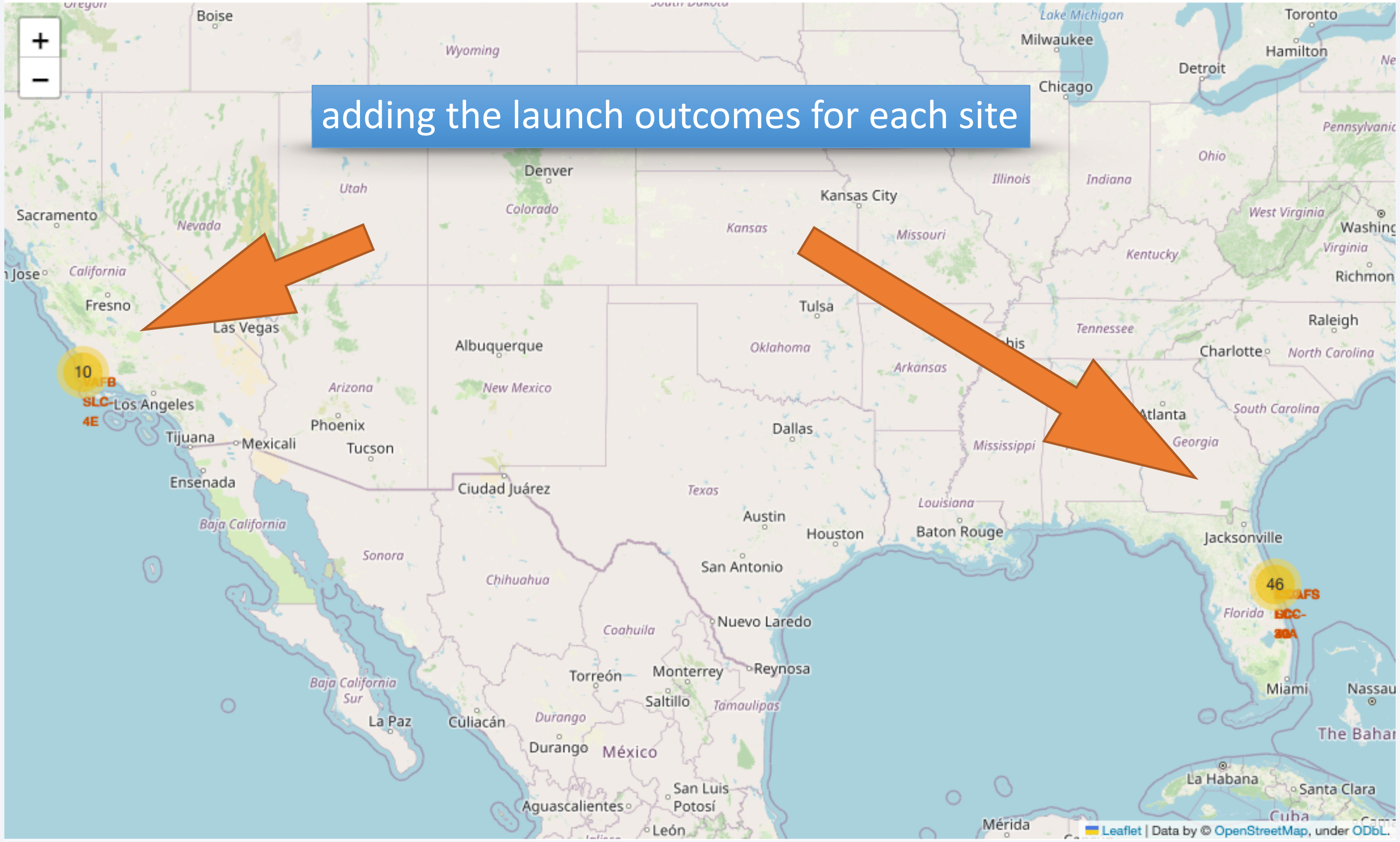
Section 3

Launch Sites Proximities Analysis

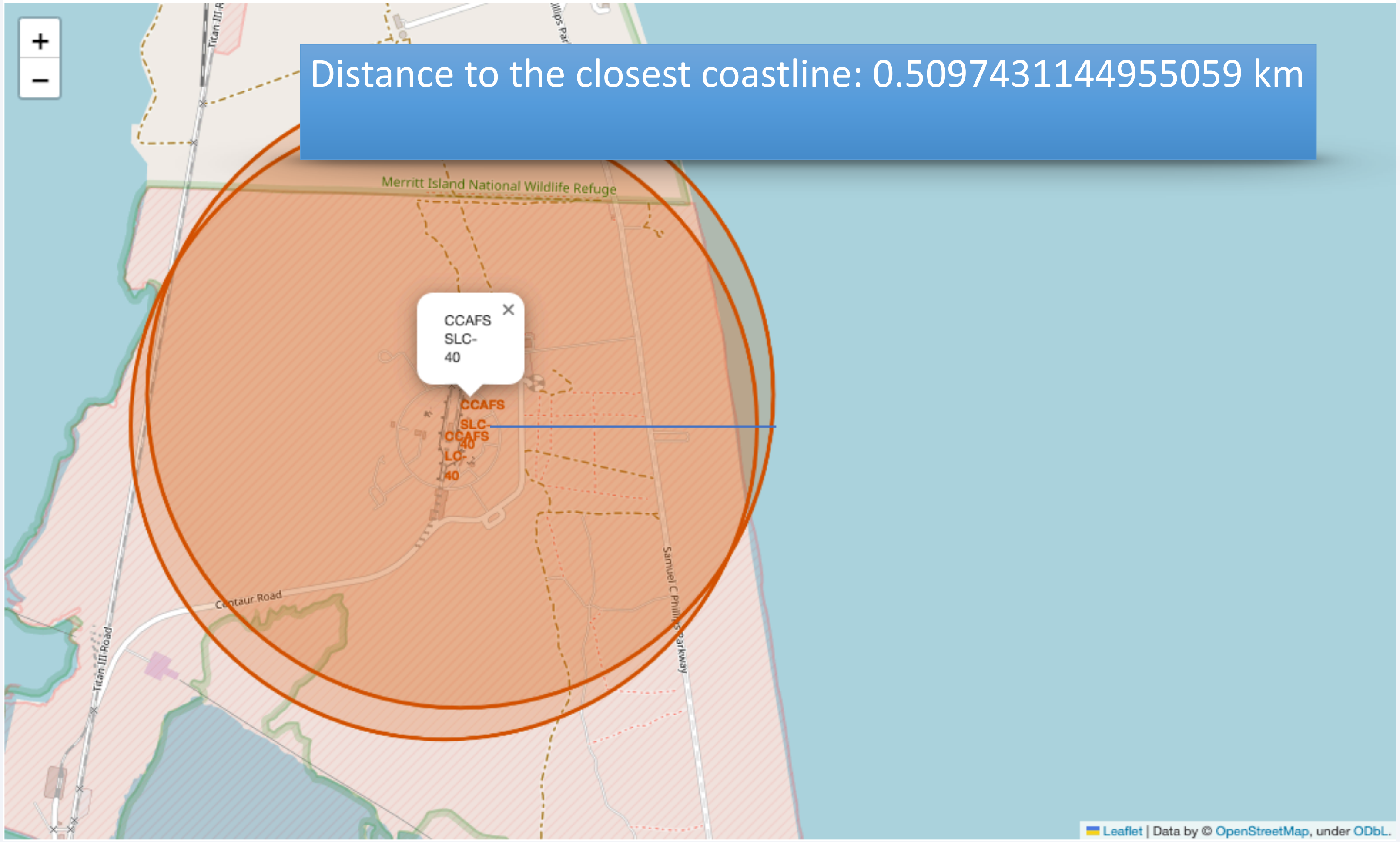
all launch sites’ location markers



color-labeled launch outcomes



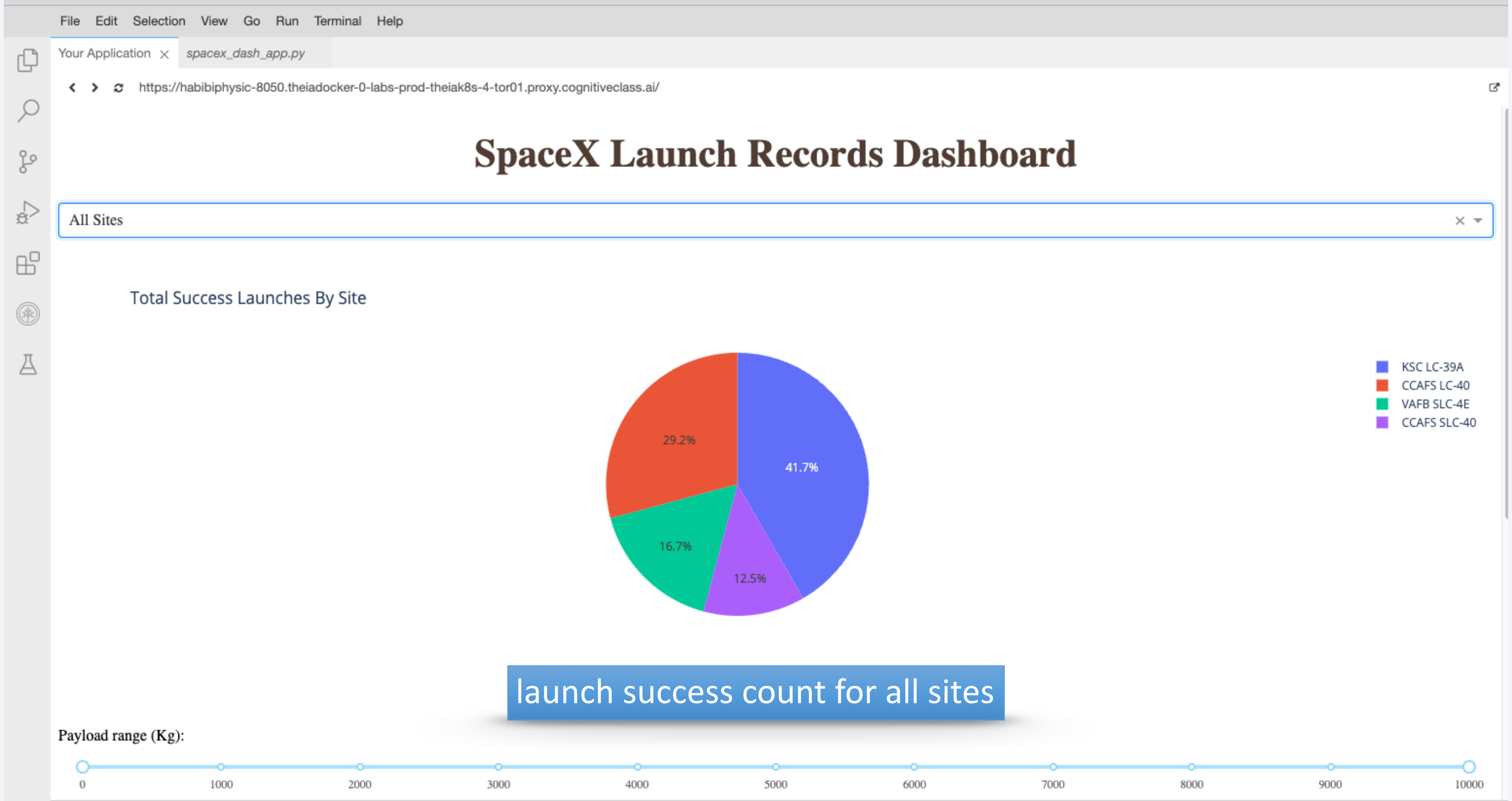
selected launch site to its proximities



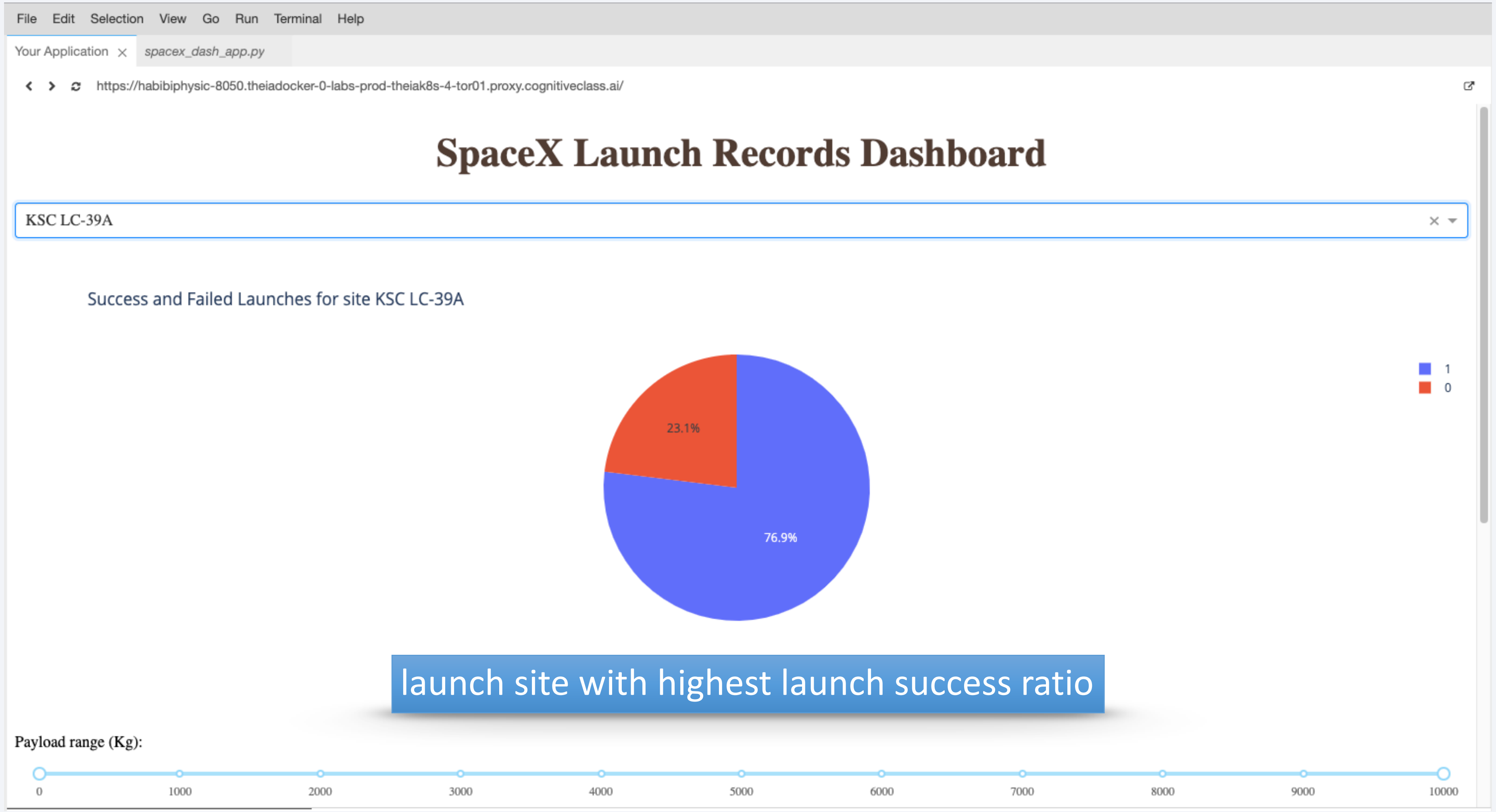
Section 4

Build a Dashboard with Plotly Dash

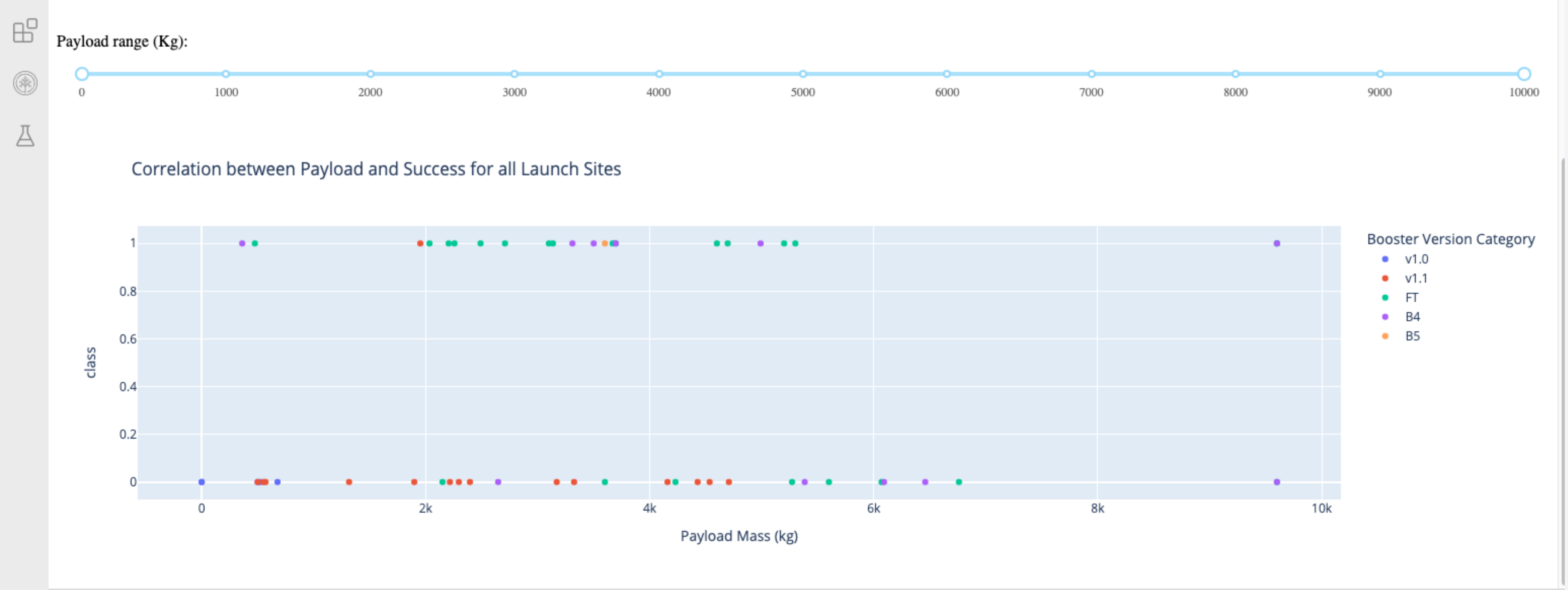
launch success count for all sites



launch site with highest launch success



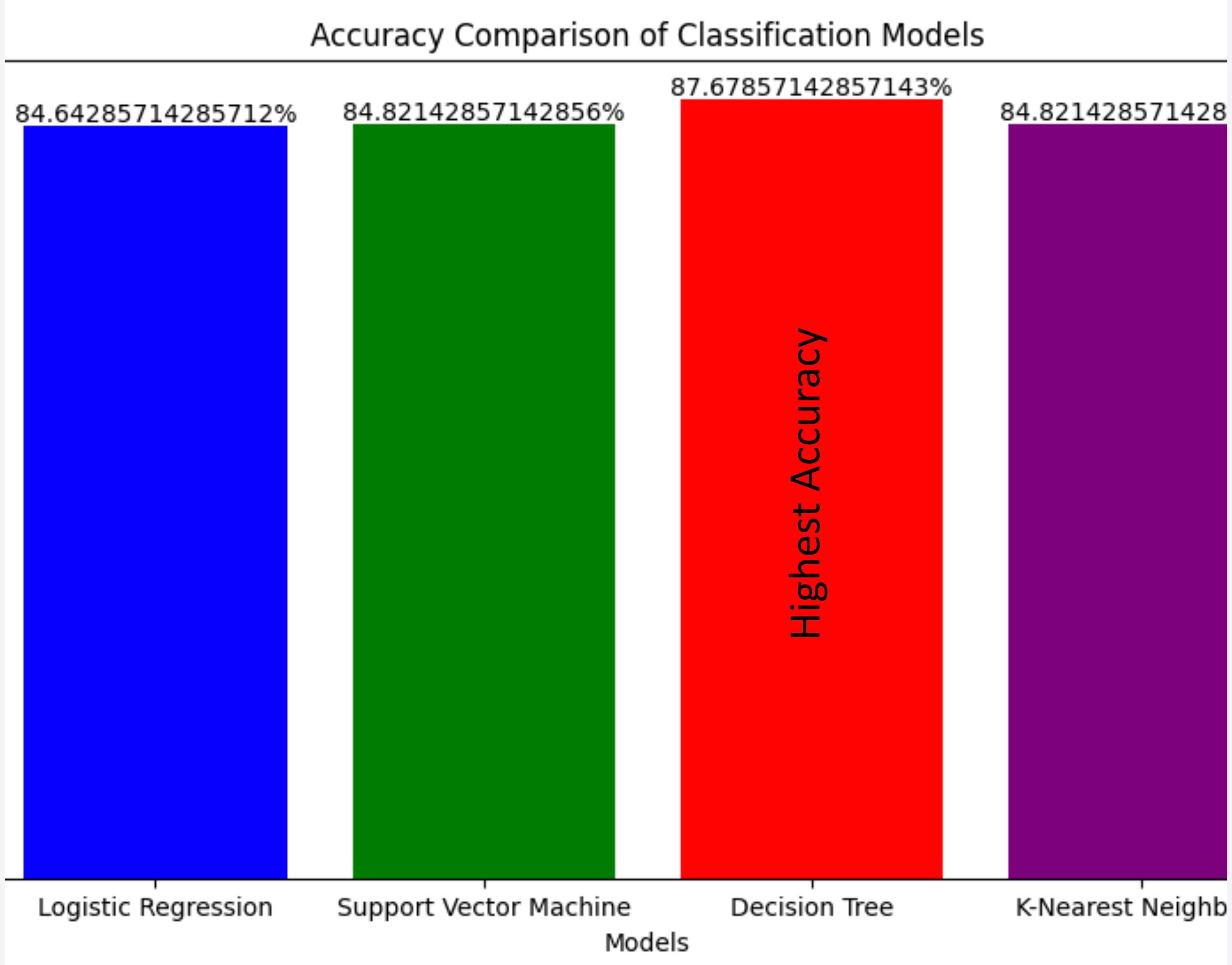
Payload vs. Launch Outcome



Section 5

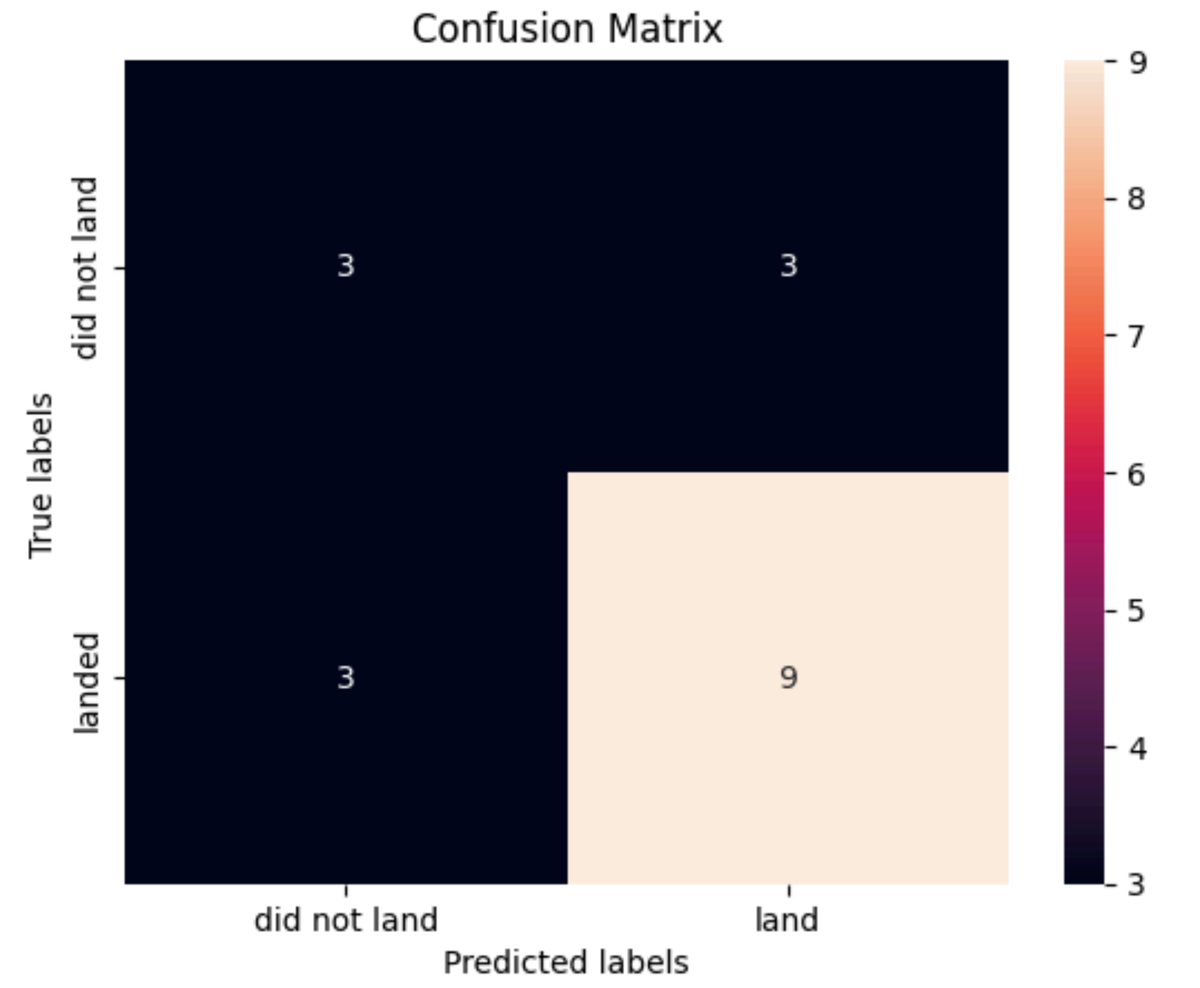
Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix

Confusion matrix of the best performing model (Decision Tree) - with the highest correlation between predicted and the true label



Conclusions

- Increasing success rate over the years
- KSC LC-39A launch site has the highest launch success ratio record
- B4 booster has the most payload among other boosters
- FT has the best operation among other boosters in launch success

Appendix

```
# Create an app layout
app.layout = html.Div(children=[
    html.H1('SpaceX Launch Records Dashboard',
            style={'textAlign': 'center', 'color': '#503D36', 'font-size': 40}),
    dcc.Dropdown(id='site-dropdown',
                 options=[
                     {'label': 'All Sites', 'value': 'ALL'},
                     {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
                     {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'},
                     {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
                     {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'},
                 ],
                 value='ALL',
                 placeholder="Select a Launch Site here",
                 searchable=True
                 ),
    html.Br(),
    html.Div(dcc.Graph(id='success-pie-chart')),
    html.Br(),
    html.P("Payload range (Kg):"),
```

Appendix

```python

# TASK 2: Add a callback function to render success-pie-chart based on selected site dropdown
@app.callback(
    Output(component_id='success-pie-chart', component_property='figure'),
    Input(component_id='site-dropdown', component_property='value')
)
def get_pie_chart(entered_site):
    if entered_site == 'ALL':
        fig = px.pie(spacex_df, names='Launch Site',
                     values='class',
                     title='Total Success Launches By Site')
        return fig
    else:
        filtered_df = spacex_df[spacex_df['Launch Site'] == entered_site]
        fig = px.pie(filtered_df, names='class',
                     title=f'Success and Failed Launches for site {entered_site}')
        return fig
```

Thank you!