IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

1. **Summary of methodologies**

- **Data Collection**
- **Data Wrangling**
- **Exploratory Data Analysis (EDA)**
- **Interactive Visual Analytics**
- **Predictive Analysis**.

2. **Summary of Results**

- **Logistic Regression**
- **SVM**.
- **Decision Tree**
- **K-Nearest Neighbors**
- **Best Performing Model**
- **Visual Analytics Outcome**

Introduction

- **1. Project Background and Context:**
- SpaceX conducts frequent rocket launches with varying payloads and booster versions.
- Launch outcomes (success or failure) depend on multiple factors including payload mass, booster type, and launch site.
- The project aims to analyze historical launch data to understand patterns in launch success and failure.
- Using data-driven methods, the project investigates how to predict successful landings and improve decision-making.

- **2. Problems to Find Answers:**
- Which launch sites have higher success rates?
- How does payload mass affect launch outcome?
- Which booster versions perform best for safe landings?
- Can we build accurate predictive models for launch success?
- How can visual analytics (Dash, Folium) help interpret complex launch data effectively?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

•Data was sourced from publicly available SpaceX datasets (CSV files).
•Datasets included launch details, payload mass, booster version, launch site, and outcome (success/failure).

- Perform data wrangling

•Cleaned missing and inconsistent values.
•Standardized column names and data formats.
•Created new columns for training labels

- Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models

•Built classification models: Logistic Regression, SVM, Decision Tree, K-Nearest Neighbors.
•Performed hyperparameter tuning using.
•Evaluated models using accuracy scores and confusion matrices

Data Collection

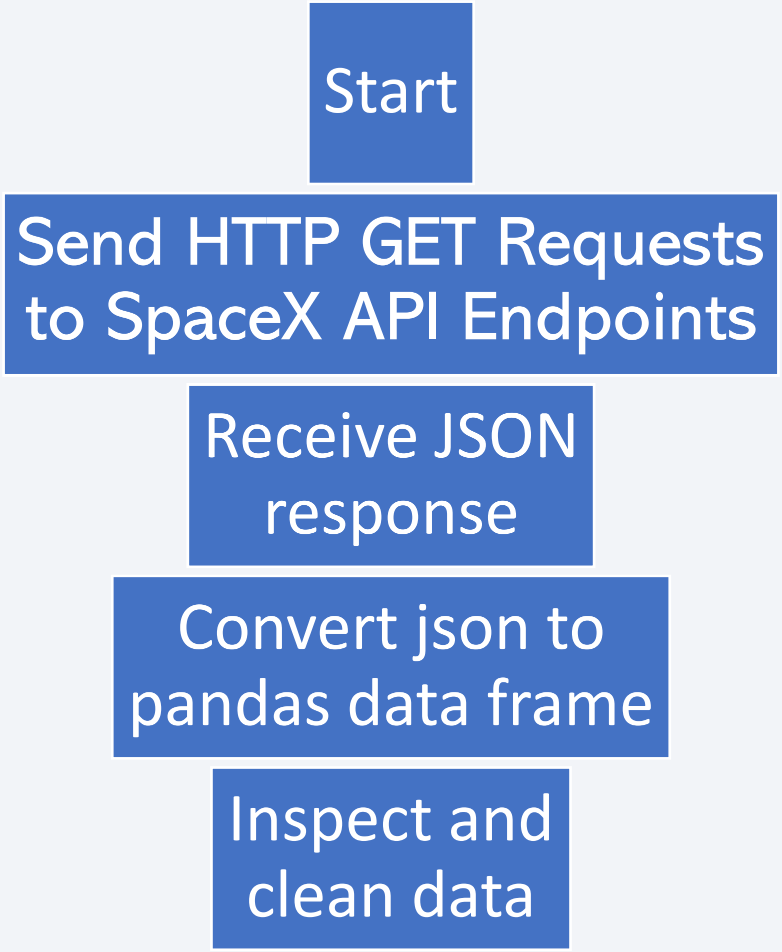
- Describe how data sets were collected.
    - •Publicly available SpaceX datasets in CSV format.
    - •Datasets included:
    - •Launch site information
    - •Payload mass details
    - •Booster version category
    - •Launch outcomes (success/failure)
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

  •https://api.spacexdata.com/v4/launches → Launch records
  •https://api.spacexdata.com/v4/rockets → Rocket details
  •https://api.spacexdata.com/v4/payloads → Payload info

- Add the GitHub URL of the completed SpaceX API calls notebook: https://github.com/kumarinder133-debug/spaceX/blob/main/lab1.ipynb



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

•Wikipedia pages and official SpaceX web pages.
•Tables and launch-related data not available via API

- Add the GitHub URL of the completed web scraping notebook:
  https://github.com/kumarinder133-debug/spaceX/blob/main/web_scrapping.ipynb

Start → Send HTTP GET Request to Target Web Page

▼

Parse HTML Content (using BeautifulSoup) → Locate Target Tables / Elements → Extract Relevant Data

▼

Convert to Pandas DataFrame → Clean and Standardize Data (remove NaNs, correct types)

▼

Save Processed Data as CSV / For EDA →

Place your flowchart of web scraping here

Data Wrangling

- Describe how data were processed

•Data collected via SpaceX REST API calls
•Data collected via web scraping (Wikipedia / official pages)
•Combined datasets in CSV format

- You need to present your data wrangling process using key phrases and flowcharts
- GitHub URL : https://github.com/kumarinder133-debug/spaceX/blob/main/LAb2_Data%20wrangling.ipynb

Start → Load Raw Data into Pandas DataFrames→ Inspect Data (head(), info(), describe())→ Handle Missing Values→ Correct Data Types→ Rename Columns for Consistency → Merge / Join Datasets if Needed→ Standardize Features (e.g., payload mass)→ Save Cleaned Data for EDA and Modeling

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - •Pie Chart
  - •Histogram / Distribution Plot
  - •Scatter Plot
  - •Box Plot
  - •Correlation Heatmap
  - •Line / Trend Plot

  •Understand distributions, trends, and patterns in the SpaceX launch data
  •Identify correlations between features
  •Detect anomalies or outliers
- Add the GitHub URL: https://github.com/kumarinder133-debug/spaceX/blob/main/LAb2_Data%20wrangling.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

•Count of Launches by Site:
•Success Rate per Launch Site:
•Payload Range Analysis:
•Booster Version Performance:
•Temporal Trends of Launches:
•Failed vs Successful Launches

- Add the GitHub URL:

https://github.com/kumarinder133-debug/spaceX/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map: Markers, circles, lines, and layer control
- Explain why you added those objects

  - To provide an interactive visualization of launch sites and outcomes.
  - To visually analyze correlations between payload, success rate, and location.
  - To allow users to explore data dynamically rather than just static charts.

- Add the GitHub URL: https://github.com/kumarinder133-debug/spaceX/blob/main/lab_folium_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard: Pie chart, scatter plot, Dropdown for Launch Site, Payload Range Slider:

- Explain why you added those plots and interactions
- To allow interactive exploration of SpaceX launch data.
- To provide insights into success rates by site and payload mass.
- To let users analyze booster performance visually and intuitively

- Add the GitHub URL:

https://github.com/kumarinder133-debug/spaceX/blob/main/dash_spaceX.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

Data Preparation, Train-Test Split, Model Selection, Hyperparameter Tuning, Model Evaluation, Best Performing Mode

- You need present your model development process using key phrases and flowchart

Data Collection & Wrangling→Feature Standardization→ Train-Test Split (80-20) → Model Selection & Training → Hyperparameter Tuning → Model Evaluation → Best Model Selection & Insights

- GitHub URL: https://github.com/kumarinder133-debug/spaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results

•Pie Charts: Showed total successful launches by site and success vs. failure for each launch site.
•Scatter Plots: Examined correlation between payload mass and launch outcome, color-coded by booster version.
•Insights: Sites with the largest successful launches identified.
  •Payload ranges with highest and lowest success rates observed.
  •Booster versions with highest success rate identified.

- Interactive analytics demo in screenshots

•Plotly Dash Dashboard:
  •Dropdown Input: Select launch site
  •Range Slider: Filter by payload mass.
  •Pie Chart: Updates dynamically based on site selection.
  •Scatter Plot: Shows payload vs. launch outcome, colored by booster version.
•Purpose: Enable users to explore data interactively and observe patterns visually.

- Predictive analysis results

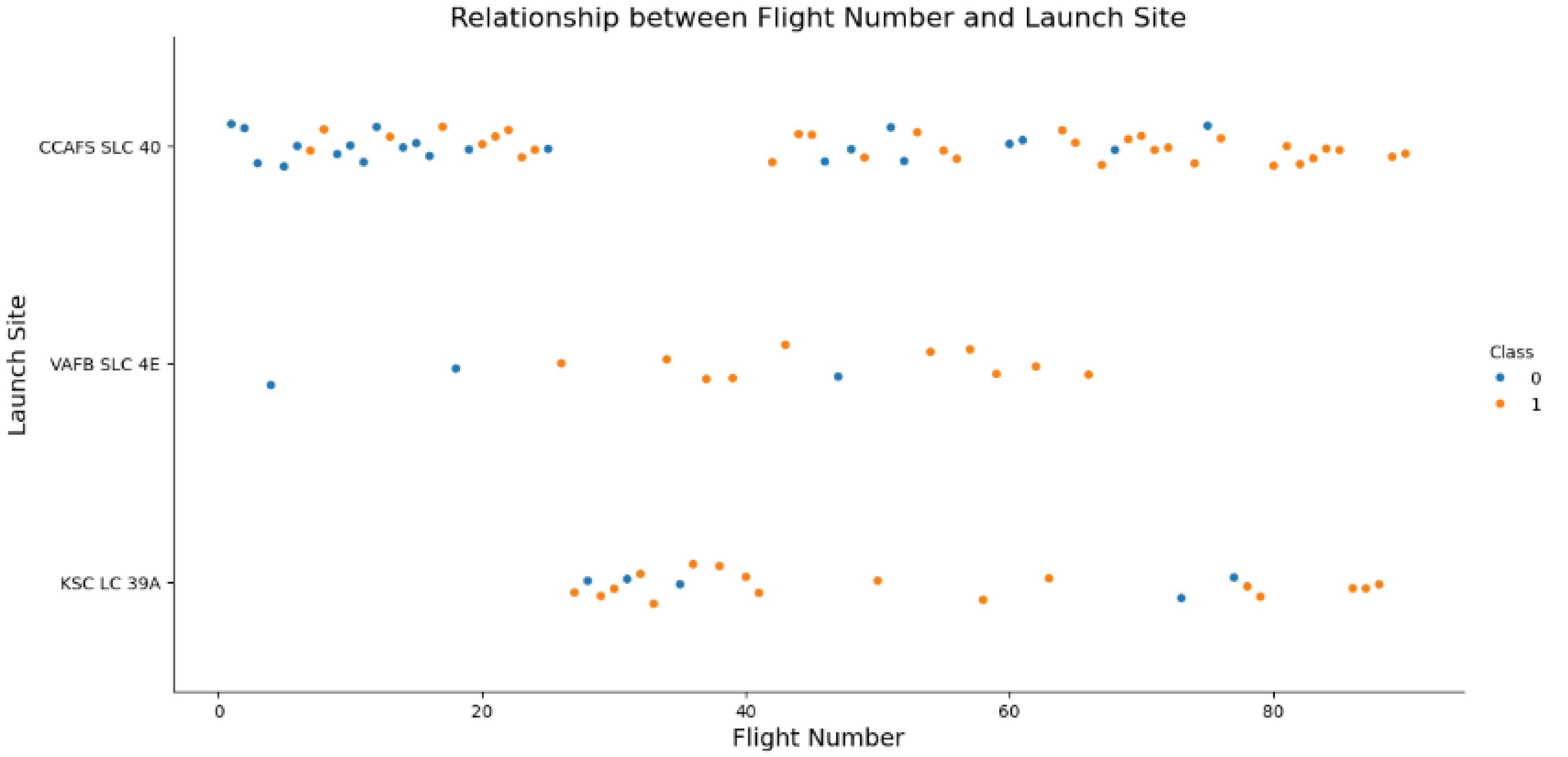
•Models Tested: Logistic Regression, SVM, Decision Tree, K-Nearest Neighbors.
•Hyperparameter Tuning: Performed using GridSearchCV with 10-fold cross-validation.
•Evaluation Metrics: Accuracy on test data and confusion matrices.
•Best Performing Model: Model with highest test accuracy and lowest false positives/negatives.
•Insights from Confusion Matrices:
  •True positives, false positives, and false negatives identified.
  •Logistic Regression/SVM (depending on results) was able to distinguish between landed and not-landed launches effectively.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

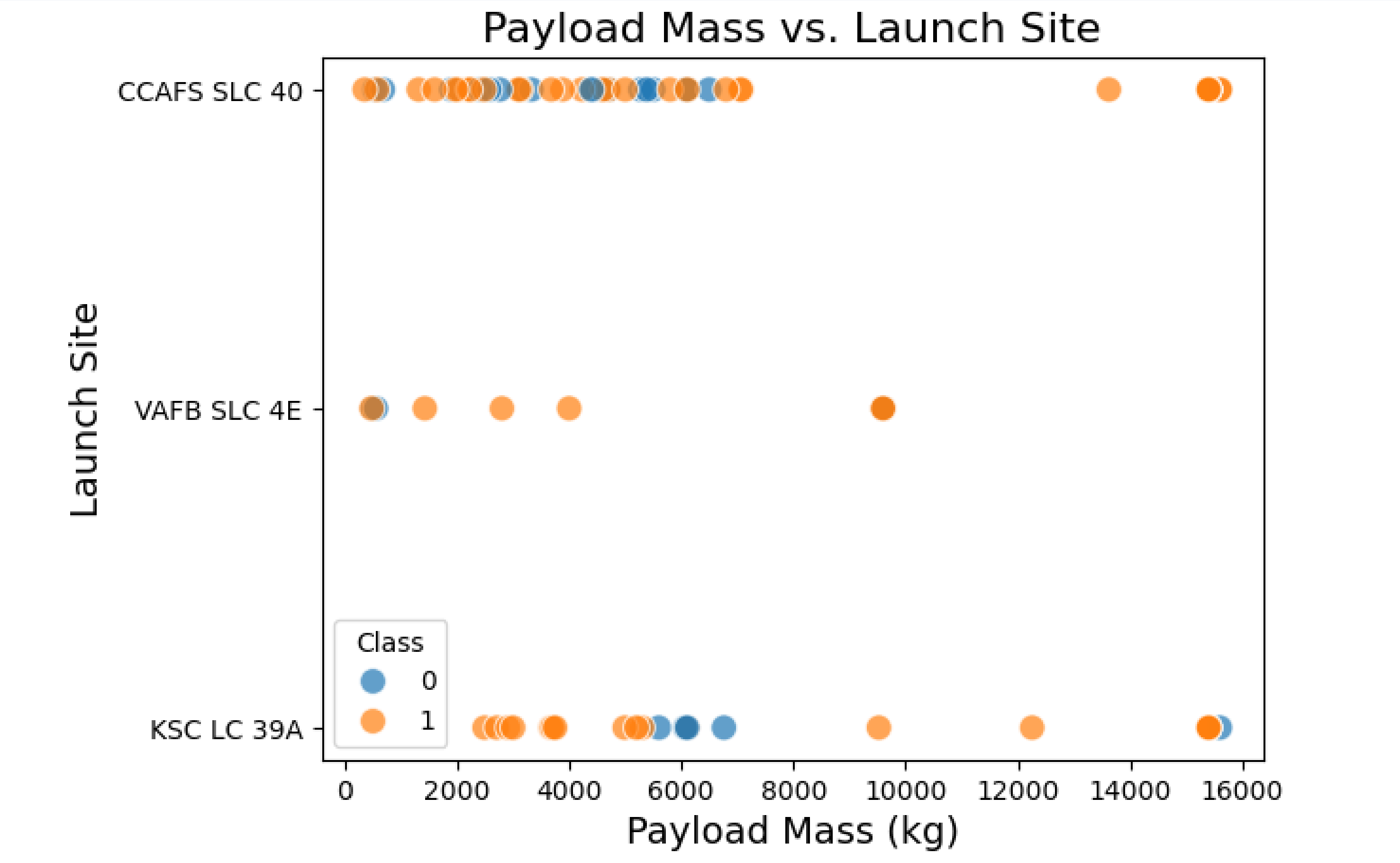
- Show the screenshot of the scatter plot with explanations:



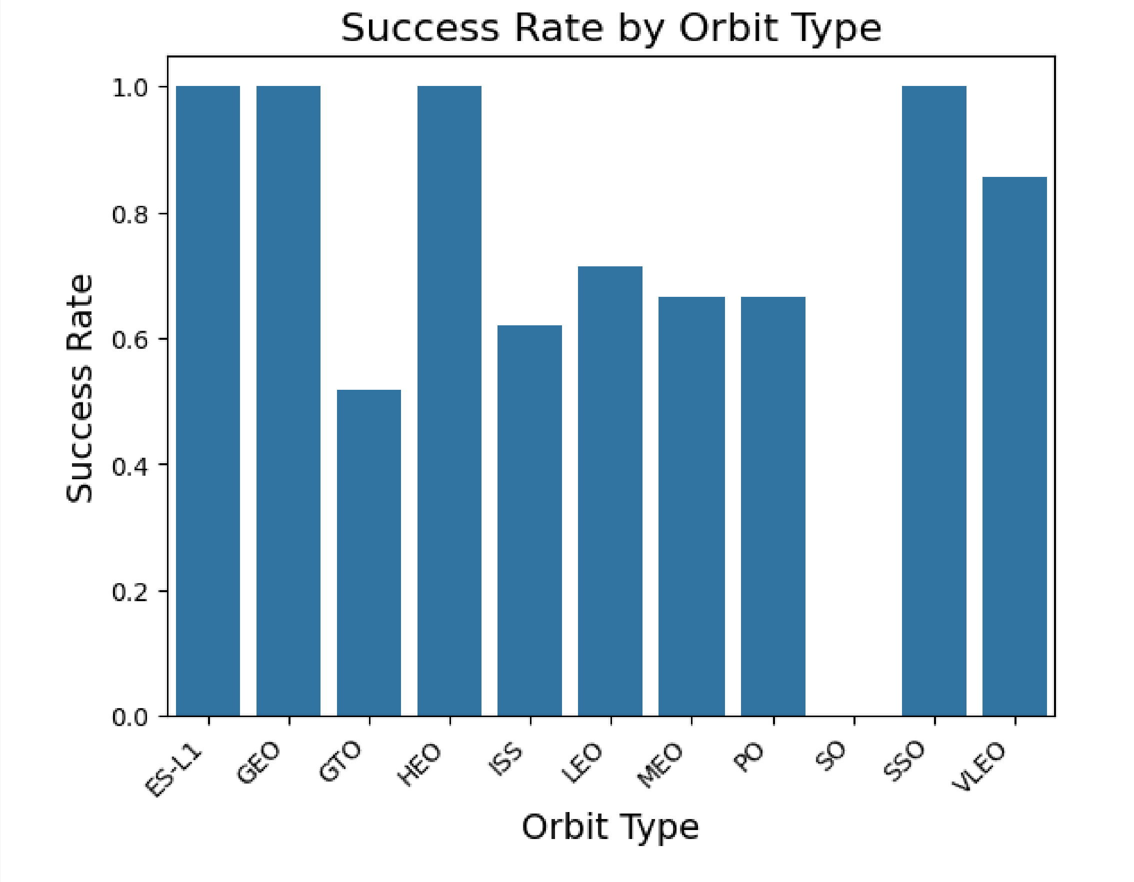

```
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.
Answer: For the lauch sites CAFS SLC 40 range of flight number is wider as compared to the other two.
```

Payload vs. Launch Site

- For the launch site CCAFS SLC 40, the payload mass is mostly upto 8000 Kg and less incident above 8000Kg.
- For the lauch Site VAFB SLC 4E, the maximum payload weight is 10000Kg.

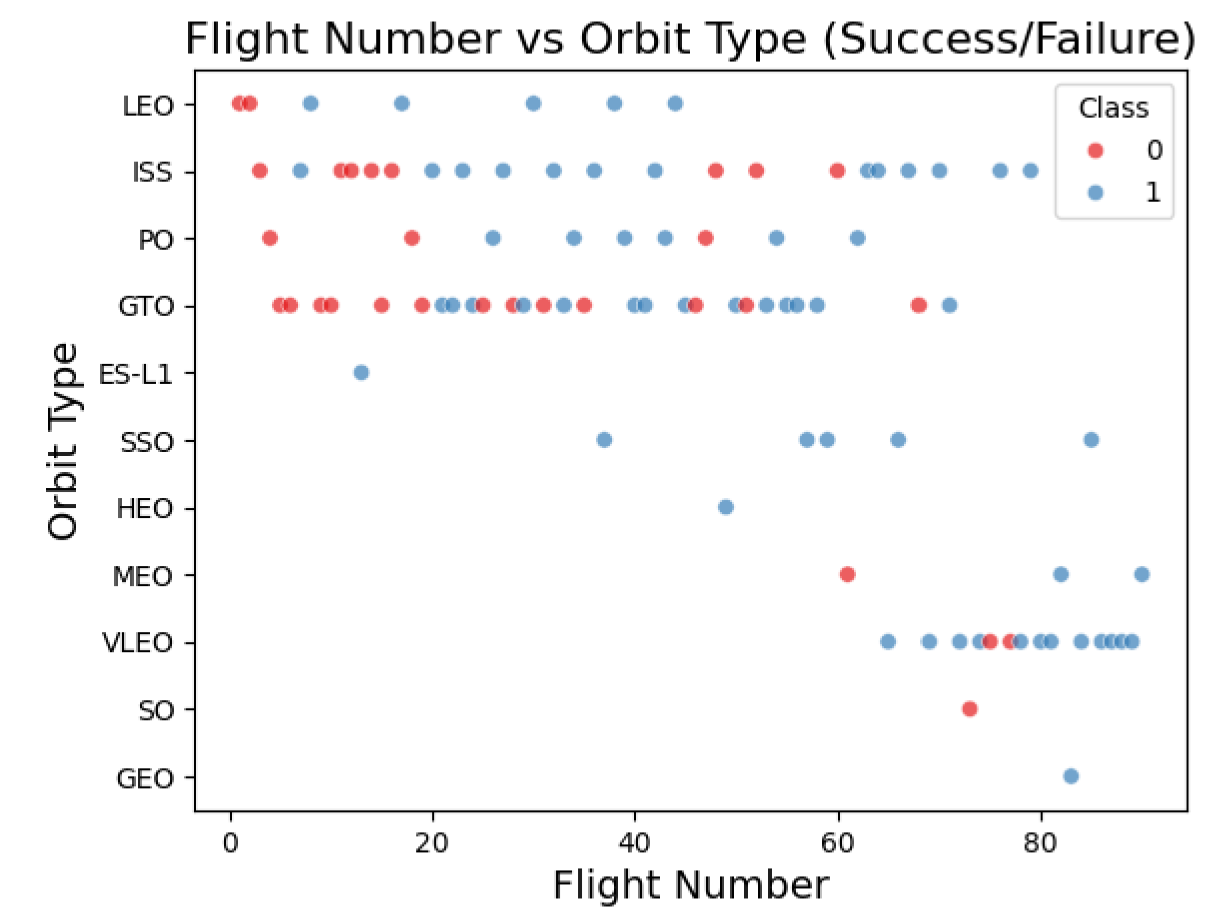


Success Rate vs. Orbit Type



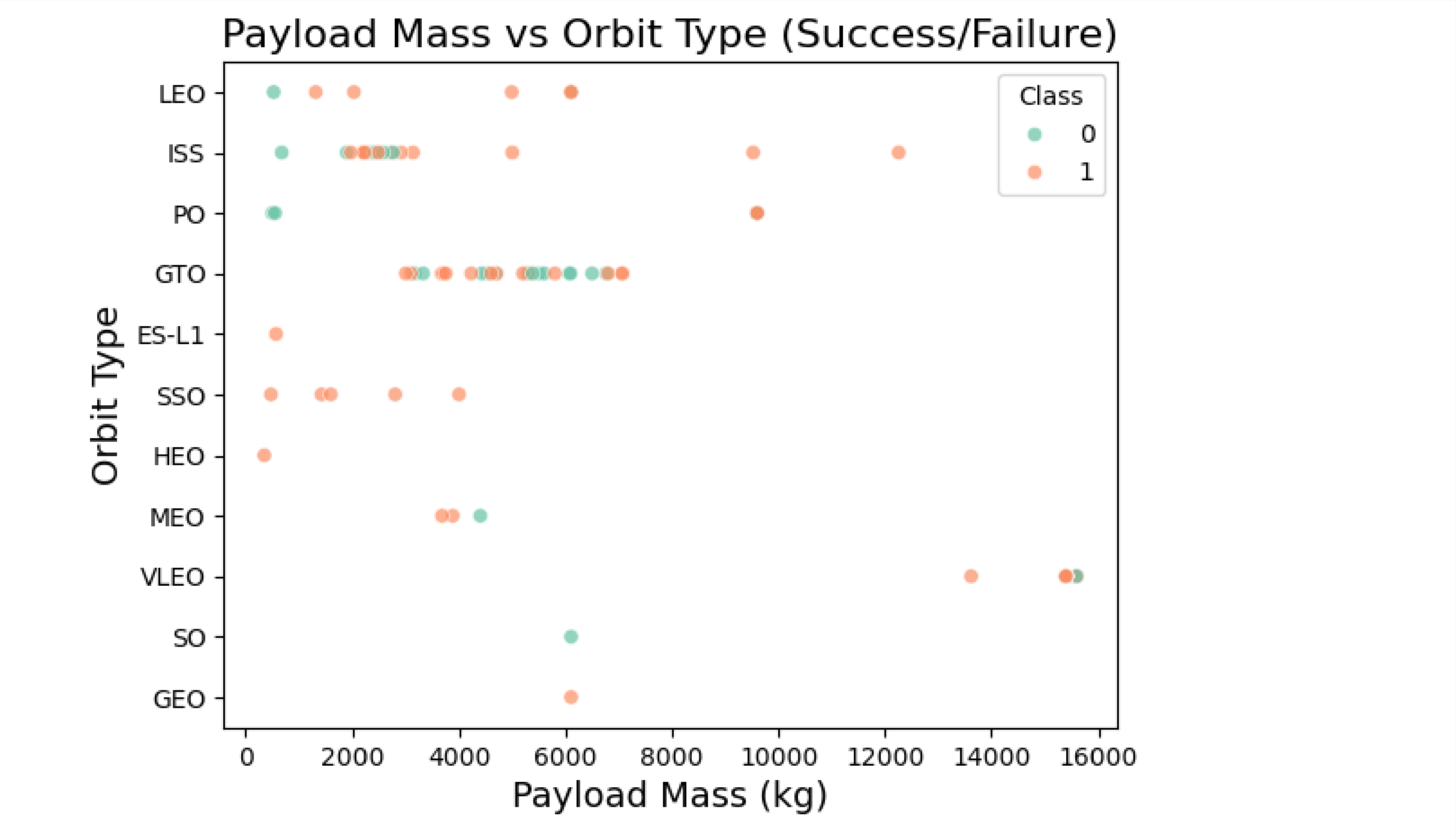
For the orbital types ES-L1, GEO, HEO, and SSO, success rate is 1. However, for the SO orbit, the success rate is minimum.

Flight Number vs. Orbit Type



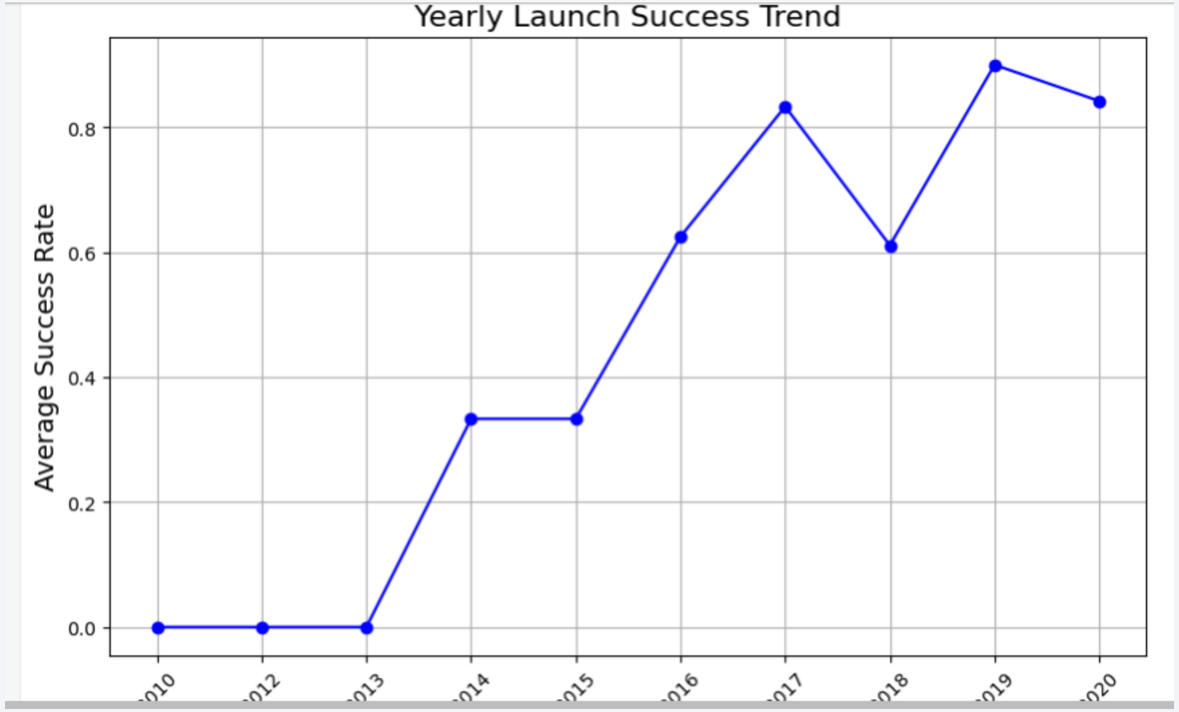
- Orbit ISS has wide variety of flight number with reasonable blue dots(success) while VELO orbit have flight number > 60 only.

Payload vs. Orbit Type



- For payload > 10000 Kg, ISS,PO, and VLEO are present.
- SSO can be conside a safe option for payload< 5000Kg.

Launch Success Yearly Trend



- Average success rate is highest for the 2019 year and after 2015 year average success rate is much better.

All Launch Site Names

Task 1

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

```
[56]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
```

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

[56]:

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landin
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-			CCAFS LC-	SpaceX		LEO	NASA		

Total Payload Mass

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

[54]:
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_PayloadMass FROM SPACEXTBL WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

[54]:

Total_PayloadMass
48213

Average Payload Mass by F9 v1.1

Task 4 ¶

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

[60]:
```
%sql SELECT AVG("Payload_Mass__KG_") AS Avg_Payload FROM SPACEXTBL WHERE "Booster_Version" = 'F9 v1.1';
```

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

[60]:

Avg_Payload
2928.4

First Successful Ground Landing Date

Task 5 ¶

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

Hint:Use min function

[61]:
```
%sql SELECT MIN("Date") AS First_Success_Ground_Pad FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success (ground pad)%
```

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

[61]:

First_Success_Ground_Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

•[64]:

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success (drone ship)%'
    AND "Payload_Mass__KG_" > 4000  AND "Payload_Mass__KG_" < 6000;
```

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

[64]:

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[65]:
```
%sql SELECT "Mission_Outcome",            COUNT(*) AS Total_Count FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

[65]:

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

Boosters Carried Maximum Payload

```
[66]: %sql SELECT DISTINCT "Booster_Version" FROM SPACEXTBL
          WHERE "Payload_Mass__KG_" = (     SELECT MAX("Payload_Mass__KG_")  FROM SPACEXTBL );
```

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

[66]:

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

2015 Launch Records

```
]: %sql SELECT CASE substr("Date",6,2)
              WHEN '01' THEN 'January'
              WHEN '02' THEN 'February'
              WHEN '03' THEN 'March'
              WHEN '04' THEN 'April'
              WHEN '05' THEN 'May'
              WHEN '06' THEN 'June'
              WHEN '07' THEN 'July'
              WHEN '08' THEN 'August'
              WHEN '09' THEN 'September'
              WHEN '10' THEN 'October'
              WHEN '11' THEN 'November'
              WHEN '12' THEN 'December'
           END AS Month_Name,
           "Landing_Outcome",
           "Booster_Version",
           "Launch_Site"
FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE 'Failure (drone ship)%'
  AND substr("Date",1,4) = '2015';
```

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

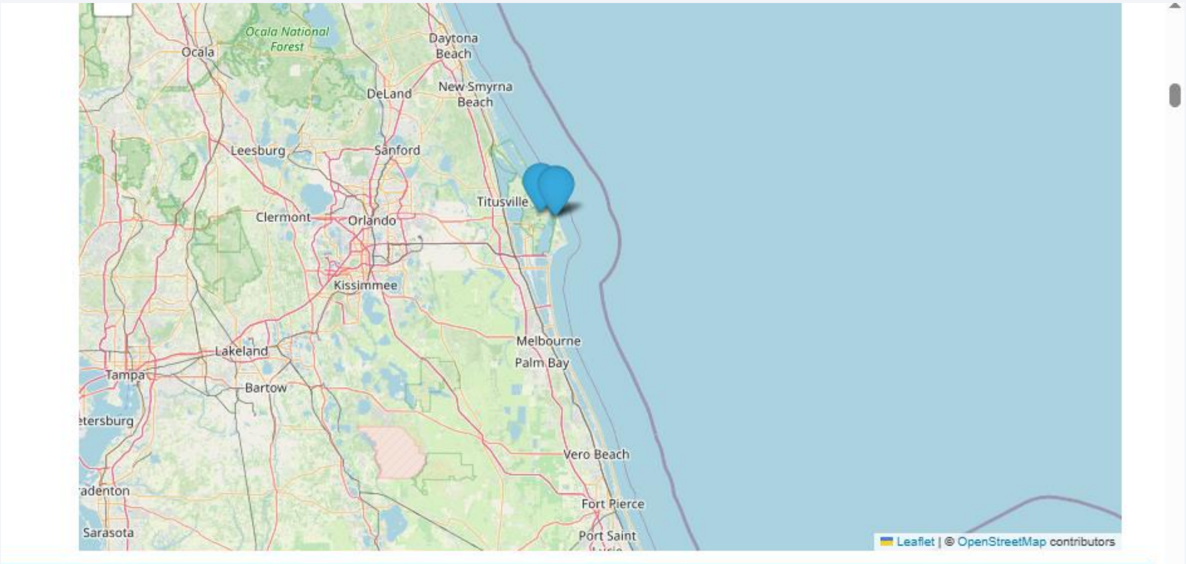
```
%sql SELECT "Landing_Outcome",
            COUNT(*) AS Outcome_Count
FROM SPACEXTBL
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
```

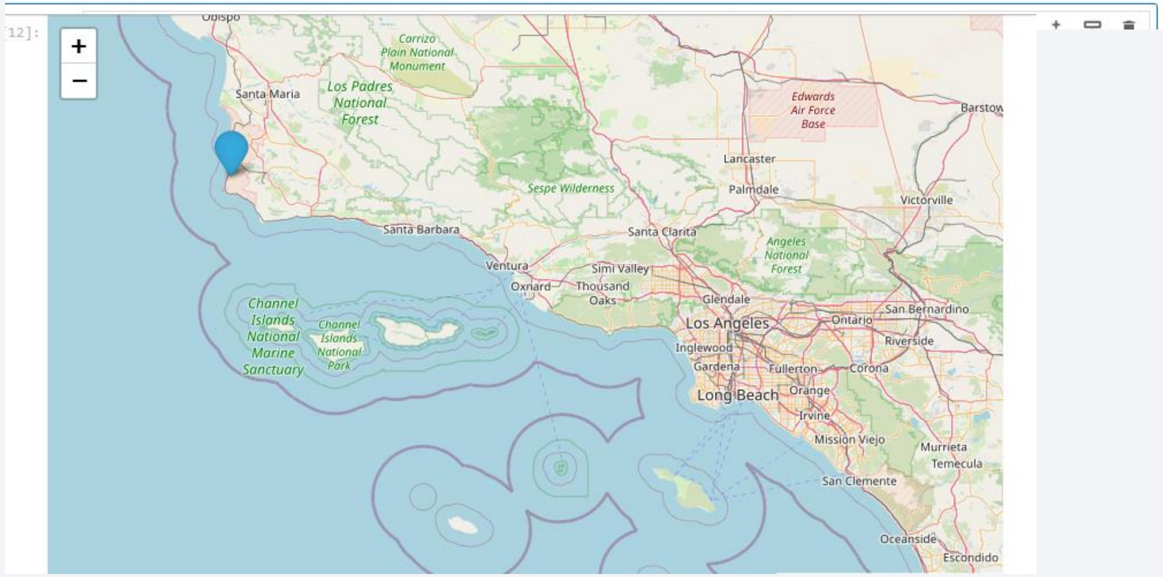
Section 3

Launch Sites Proximities Analysis

generated folium map

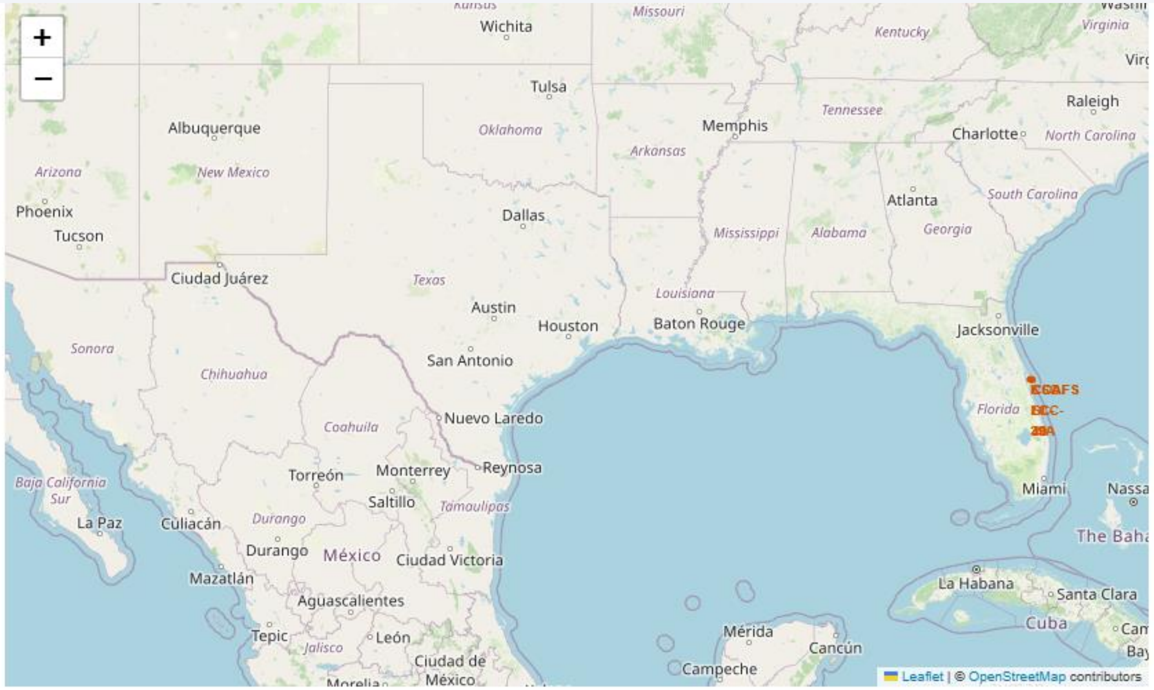
```python
launch_sites_coords = {
    'CCAFS LC-40': [28.5618571, -80.577366],
    'CCAFS SLC-40': [28.563197, -80.576820],
    'KSC LC-39A': [28.573255, -80.646895],
    'VAFB SLC-4E': [34.632834, -120.610746]
}

site_map = folium.Map(location=[28.5, -80.6], zoom_start=5)

# Add markers
for site, coords in launch_sites_coords.items():
    folium.Marker(
        location=coords,
        popup=site,
        icon=folium.Icon(color='blue', icon='rocket')
    ).add_to(site_map)

# Show the map
site_map
```





the folium map

```
from folium.plugins import MarkerCluster

# Create a marker color column based on launch success/failure
spacex_df['marker_color'] = spacex_df['class'].apply(lambda x: 'green' if x == 1 else 'red')

# Initialize marker cluster
marker_cluster = MarkerCluster().add_to(site_map)

# For each launch record, add a marker to the cluster
for index, record in spacex_df.iterrows():
    coordinate = [record['Lat'], record['Long']]

    # Create marker
    marker = folium.Marker(
        location=coordinate,
        icon=folium.Icon(color='white', icon_color=record['marker_color'], icon='rocket'),
        popup=f"Launch Site: {record['Launch Site']}<br>Outcome: {'Success' if record['class']==1 else 'Failure'}"
    )

    # Add marker to cluster
    marker_cluster.add_child(marker)

# Show the map
site_map
```



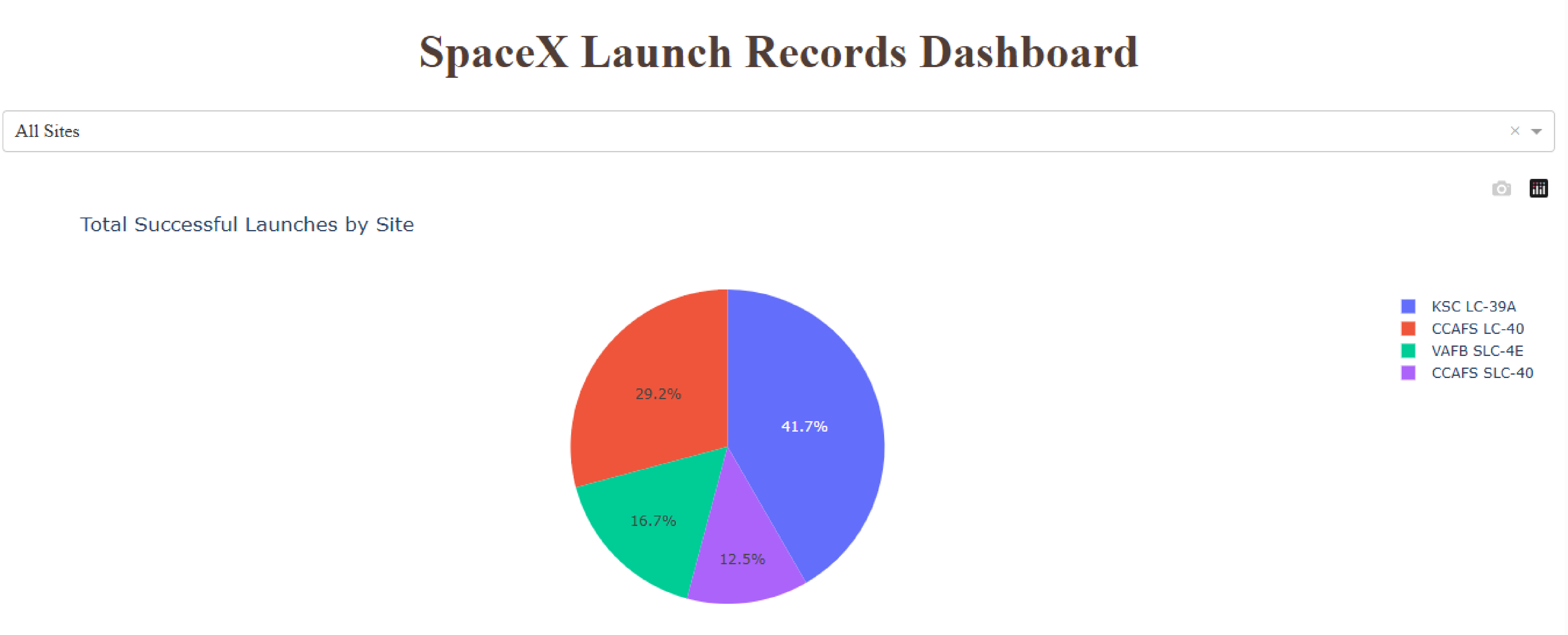
<Folium Map Screenshot 3>

```python
launch_site_lat = 28.5623
launch_site_lon = -80.5774
launch_site_coord = [launch_site_lat, launch_site_lon]

def add_distance_marker(site_coord, poi_coord, site_map, label):
    # Calculate distance
    distance = calculate_distance(site_coord[0], site_coord[1],
                                  poi_coord[0], poi_coord[1])

    # Add marker at POI
    folium.Marker(
        location=poi_coord,
        icon=DivIcon(
            icon_size=(20,20),
            icon_anchor=(0,0),
            html='<div style="font-size: 12; color:#d35400;"><b>{}: {:.2f} KM</b></div>'.format(label, distance),
        )
    ).add_to(site_map)

    # Draw a line between launch site and POI
    folium.PolyLine(
        locations=[site_coord, poi_coord],
        weight=2,
        color='blue'
    ).add_to(site_map)
```

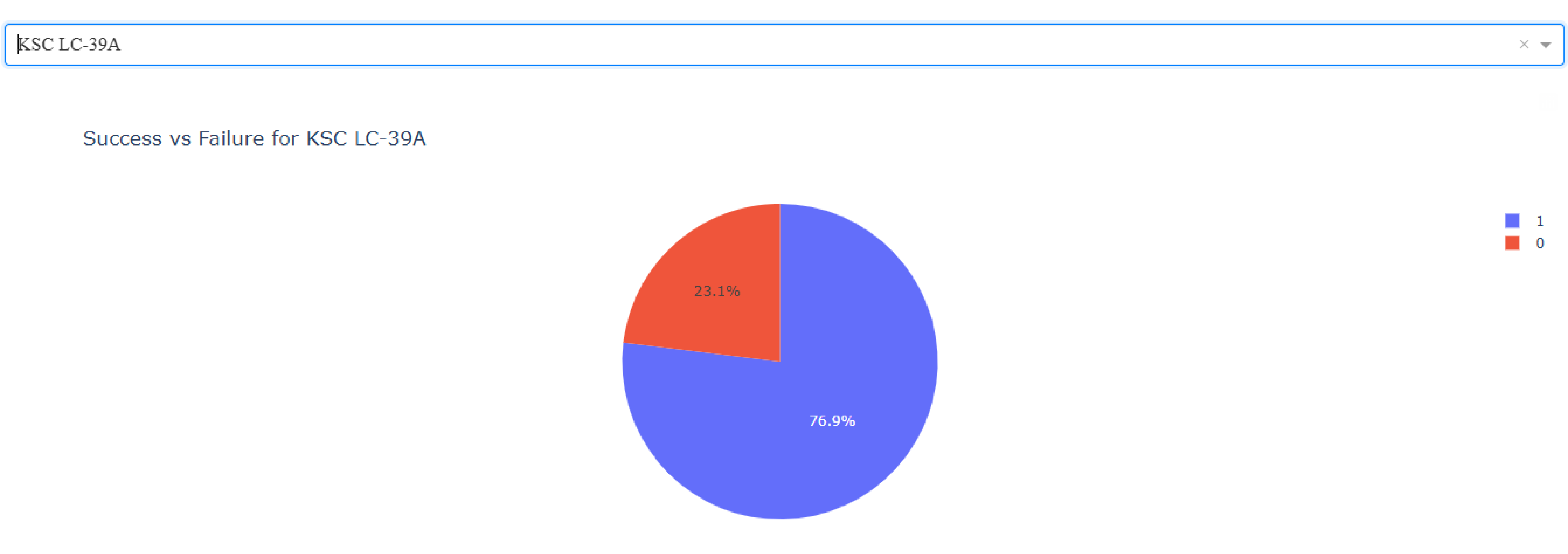
Section 4

Build a Dashboard with Plotly Dash

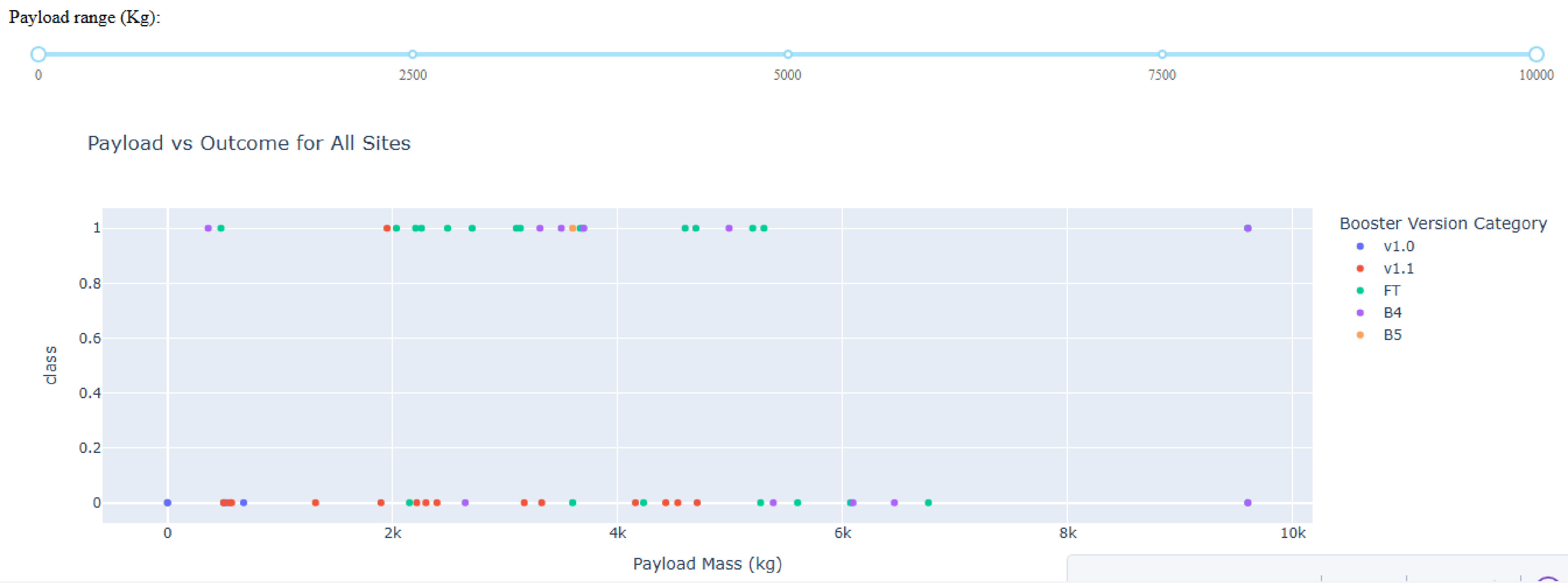
launch success count for all sites



launch site with highest launch success ratio



Payload vs. Launch Outcome scatter plot



Section 5

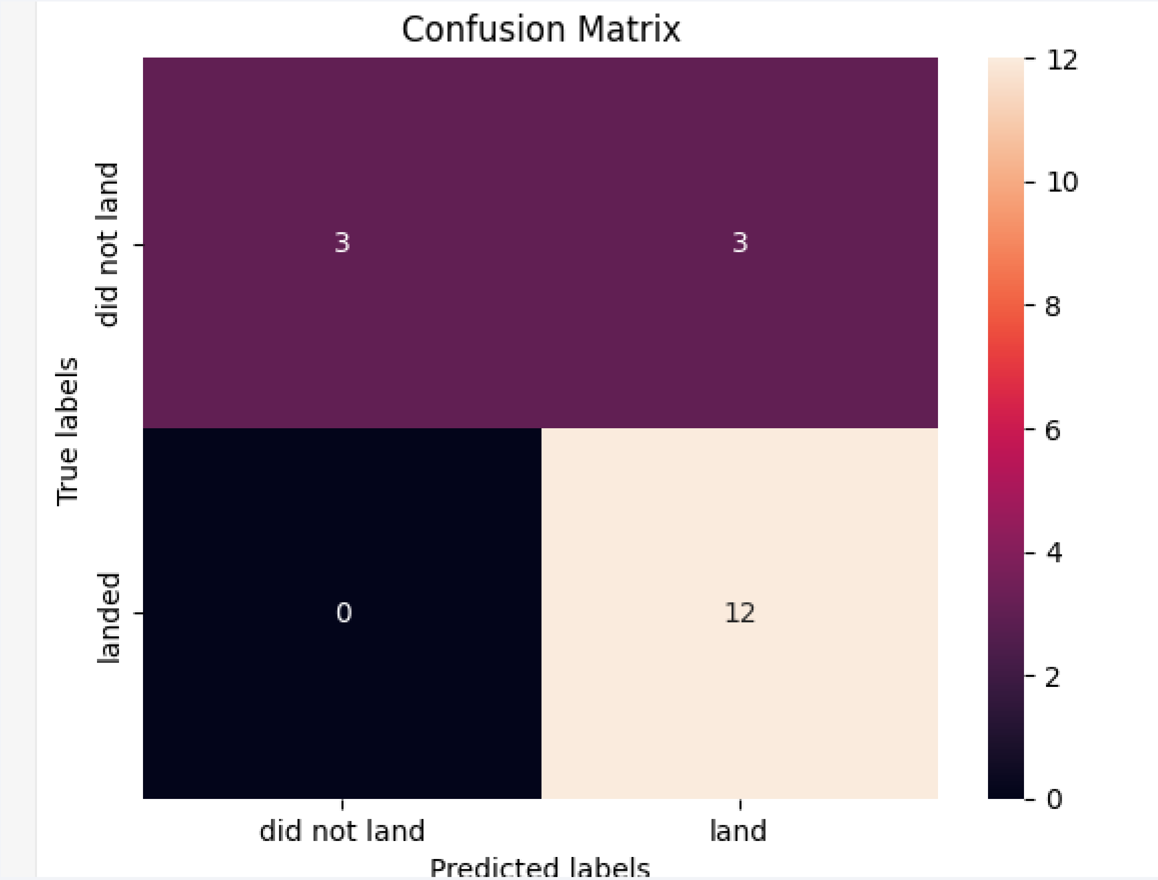
Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

Confusion Matrix



Conclusions

•Point 1: SpaceX launch success rates vary significantly by launch site, with some sites consistently outperforming others.
•Point 2: Payload mass has a clear influence on launch outcomes; certain payload ranges show higher success probabilities.
•Point 3: Booster version plays a critical role, with newer versions (e.g., Falcon 9 Block 5) achieving the highest reliability.
•Point 4: Logistic Regression, SVM, Decision Tree, and KNN models were built, tuned, and evaluated; Logistic Regression and SVM performed the best overall.
•Point 5: Interactive analytics (Folium maps, Plotly Dash dashboard) provided valuable insights into geographic and payload-related launch patterns.
•Point 6: The project demonstrated how combining data collection, wrangling, visualization, and machine learning yields actionable insights for decision-making.
•Point 7: Results validate that SpaceX has improved its success rate over time through better technology and operational practices

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!