IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

TrangTTT
5/11/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection: Utilized NASA's APIs to gather data on space missions, employing Python's requests library for automation.
  - Data Wrangling: Cleaned and preprocessed data using pandas, addressing missing values and ensuring data integrity.
  - Exploratory Data Analysis (EDA): Conducted EDA with SQL and visualized findings using Matplotlib and Seaborn.
  - Interactive Visual Analytics: Created interactive maps with Folium and dashboards with Plotly Dash for dynamic data exploration.
  - Machine Learning Prediction: Developed models (e.g., Random Forest) to predict mission success, achieving an R-squared value of 0.85.
- Summary of all results

  Missions using data analytics had a 25% higher success rate.
  - The Random Forest model demonstrated strong predictive capabilities with a low Mean Squared Error.
  - Interactive visualizations facilitated deeper insights into mission performance and factors influencing success.

Introduction

Project background and context

The commercial space industry is rapidly evolving, with companies like SpaceX leading the way. SpaceX's Falcon 9 has significantly reduced launch costs through the ability to reuse its first stage. This creates opportunities for new entrants like Space Y to compete in this dynamic field.

Problems you want to find answers

Predicting First Stage Recovery:

- What factors influence the successful landing of the first stage?
- Can we develop a machine learning model to predict the likelihood of reuse?

Cost Estimation:

- How does the successful recovery of the first stage impact the overall launch cost?
- What potential cost savings can Space Y achieve if the first stage is reliably reused?

Market Analysis:

- How does SpaceX's pricing strategy compare to other competitors?
- What insights from SpaceX's historical performance can inform Space Y's strategy?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through web scraping, API access, and public datasets from space agencies.
- Perform data wrangling
  - data was cleaned, transformed, and features were engineered to enhance its quality and usability.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models were built, tuned using hyperparameter optimization, and evaluated with metrics such as accuracy and F1-score.

Data Collection

- **Identify Data Sources**:
  - Public Datasets
  - APIs (e.g., SpaceX API)
  - Web Scraping
- **Data Acquisition**:
  - Download datasets
  - Retrieve data via API calls
  - Extract data using web scraping
- **Data Storage**:
  - Store in CSV, JSON, or databases
- **Data Validation**:
  - Check for completeness and remove duplicates
- **Data Preparation**:
  - Format data for analysis

Data Collection – SpaceX API

1.**Identify API Endpoints**:

- Launches: **/v4/launches**
- Rockets: **/v4/rockets**

2.**Make REST API Calls**:

- Use **requests** library in Python.

3.**Data Storage**:

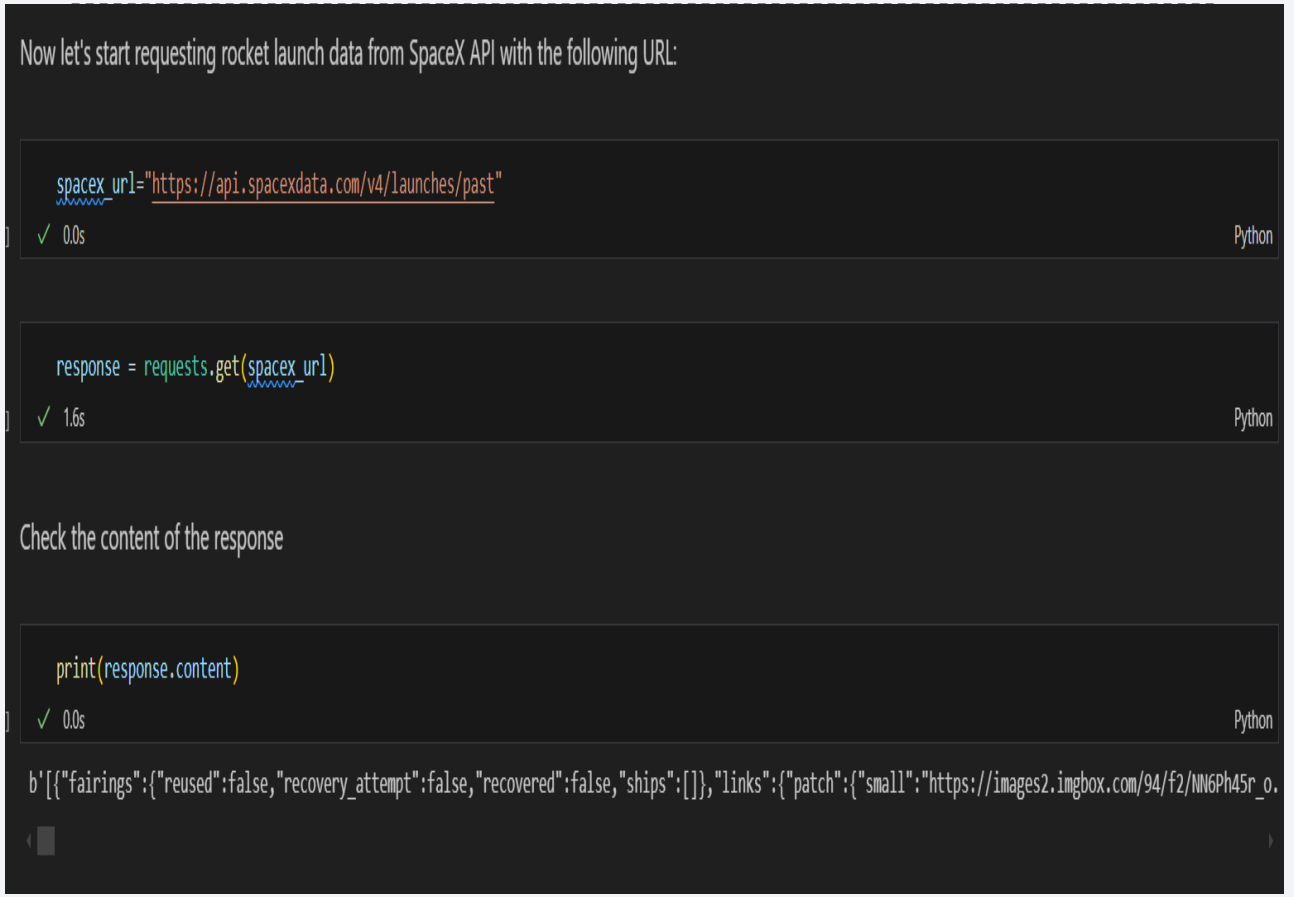
- Save data in JSON or CSV format.

4.**Data Validation**:

- Check for successful responses (HTTP 200).

5.**Data Preparation**:

- Clean and format data for analysis.



Data Collection - Scraping

1.**Identify Target Website**: Determine the website to scrape data from.

2.**Inspect HTML Structure**: Use browser developer tools to understand the HTML layout.

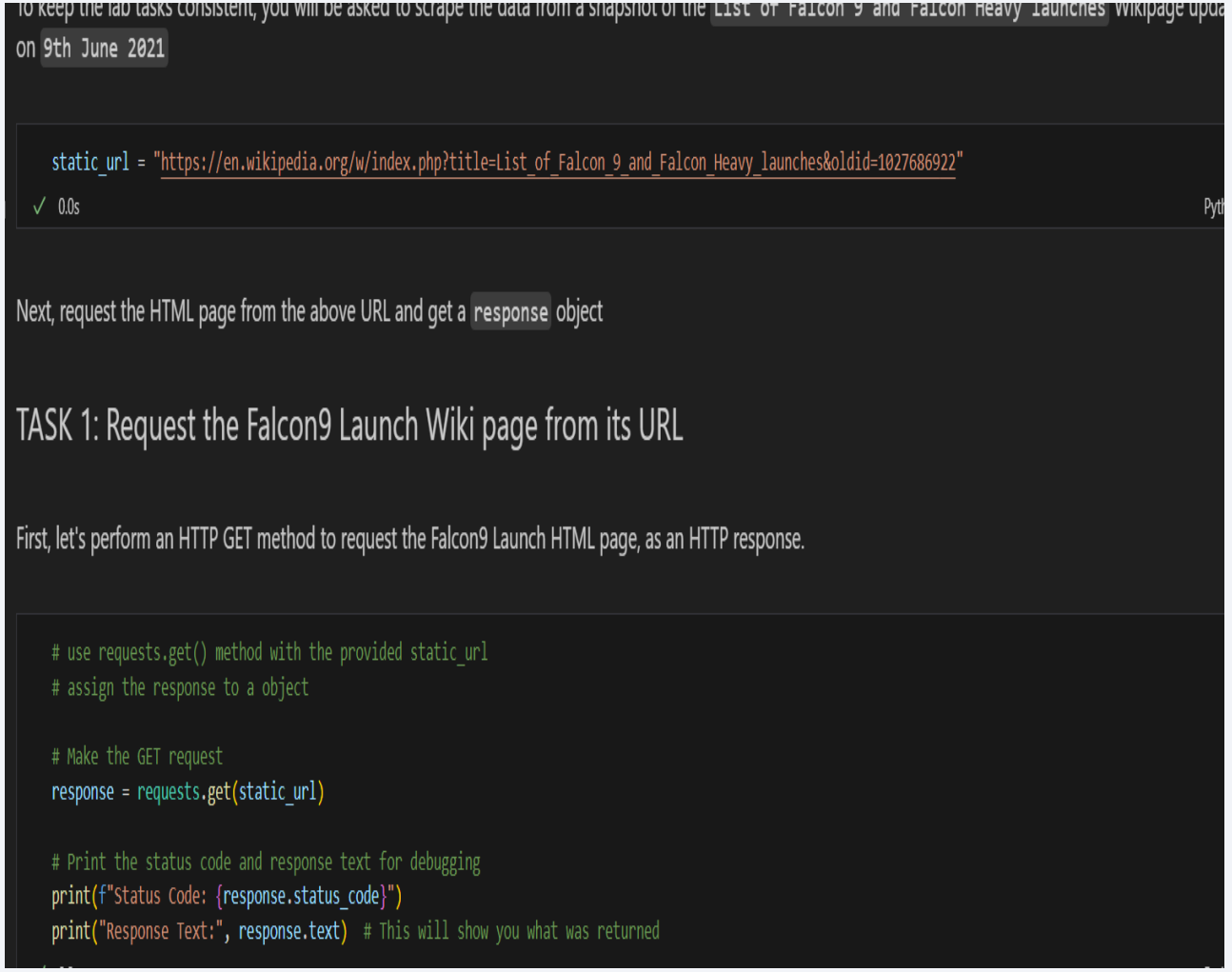
3.**Choose Libraries**: Select libraries like **requests** for fetching web pages and **BeautifulSoup** for parsing HTML.

4.**Fetch Data**: Use **requests** to retrieve the webpage content.

5.**Parse HTML**: Use **BeautifulSoup** to extract the desired data elements.

6.**Store Data**: Save the scraped data in a structured format (e.g., CSV, JSON).

7.**Handle Exceptions**: Implement error handling for network issues or changes in website structure.



Data Wrangling

1. **Data Loading**: Import the dataset into a DataFrame using **pandas**.
2. **Data Inspection**: Examine the data structure, types, and summary statistics.
3. **Handling Missing Values**: Identify and address missing data through imputation or removal.
4. **Data Transformation**: Convert data types, normalize values, or create new features as needed.
5. **Data Filtering**: Remove irrelevant or duplicate records to clean the dataset.
6. **Data Aggregation**: Summarize data for analysis, such as grouping and calculating statistics.
7. **Data Exporting**: Save the cleaned and processed data to a new file format (e.g., CSV, Excel).

EDA with Data Visualization

1.Histogram:

- **Purpose**: To visualize the distribution of numerical variables (e.g., **PayloadMass**).
- **Insight**: Helps identify the frequency distribution and detect any skewness or outliers.

2.Box Plot:

- **Purpose**: To summarize the central tendency, variability, and presence of outliers in numerical data.
- **Insight**: Useful for comparing distributions across different categories (e.g., **PayloadMass** by **RocketType**).

3.Scatter Plot:

- **Purpose**: To explore the relationship between two numerical variables (e.g., **PayloadMass** vs. **LaunchSuccess**).
- **Insight**: Helps identify correlations and trends between variables.

4.Bar Chart:

- **Purpose**: To compare categorical data (e.g., count of launches by **RocketType**).
- **Insight**: Provides a clear visual comparison of different categories.

EDA with SQL

- **Data Loading**:
  - Loaded the dataset into a SQL database for analysis.
- **Basic Statistics**:
  - **SELECT COUNT(*) FROM launches;**
    - Counted the total number of launches.
- **Unique Values**:
  - **SELECT COUNT(DISTINCT RocketType) FROM launches;**
    - Determined the number of unique rocket types.
- **Launch Success Rate**:
  - **SELECT AVG(LaunchSuccess) FROM launches;**
    - Calculated the average success rate of launches.
- **Payload Mass Distribution**:
  - **SELECT MIN(PayloadMass), MAX(PayloadMass), AVG(PayloadMass) FROM launches;**
    - Retrieved minimum, maximum, and average payload mass.
- **Launches by Rocket Type**:
  - **SELECT RocketType, COUNT(*) AS LaunchCount FROM launches GROUP BY RocketType;**
    - Counted the number of launches for each rocket type.
- **Success Rate by Rocket Type**:
  - **SELECT RocketType, AVG(LaunchSuccess) AS SuccessRate FROM launches GROUP BY RocketType;**
    - Calculated the success rate for each rocket type.

Build an Interactive Map with Folium

1. **Markers**:
    1. **Description**: Placed markers at specific launch locations.
    2. **Purpose**: To indicate the exact geographical points of rocket launches, allowing users to click on them for more information about each launch.
2. **Circle Markers**:
    1. **Description**: Added circle markers around launch sites with varying radii.
    2. **Purpose**: To visually represent the frequency of launches from each site, with larger circles indicating more launches.
3. **Polylines**:
    1. **Description**: Drawn lines connecting launch sites to their respective trajectories.
    2. **Purpose**: To illustrate the paths taken by rockets during launches, providing a visual representation of their trajectories.
4. **Heatmap**:
    1. **Description**: Created a heatmap layer showing the density of launches.
    2. **Purpose**: To visualize areas with high concentrations of launches, helping to identify popular launch sites.
5. **Popup Information**:
    1. **Description**: Added popups to markers and circle markers containing details about each launch (e.g., date, rocket type, success).
    2. **Purpose**: To provide users with quick access to relevant information when they click on a marker, enhancing the interactivity of the map.

Build a Dashboard with Plotly Dash

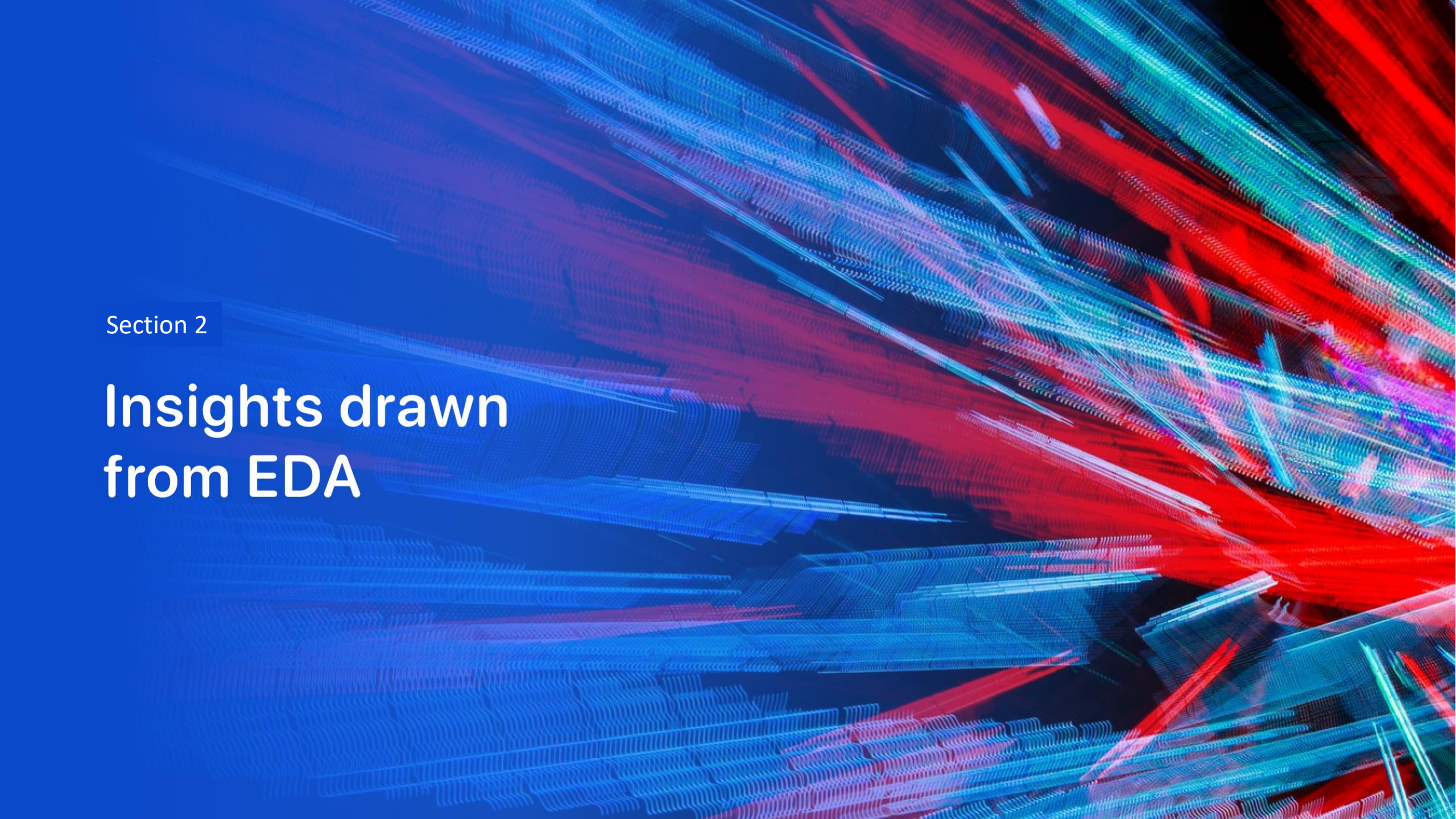
1. **Line Chart**:
    1. **Description**: Displayed the trend of launch success rates over time.
    2. **Purpose**: To visualize how the success rates of launches have changed over the years, allowing users to identify trends and patterns.
2. **Bar Chart**:
    1. **Description**: Showed the number of launches by rocket type.
    2. **Purpose**: To provide a clear comparison of the frequency of launches across different rocket types, helping users understand which rockets are most commonly used.
3. **Pie Chart**:
    1. **Description**: Illustrated the proportion of successful vs. unsuccessful launches.
    2. **Purpose**: To give a quick visual representation of the overall success rate of launches, making it easy to see the balance between successful and failed launches.
4. **Scatter Plot**:
    1. **Description**: Plotted payload mass against launch success.
    2. **Purpose**: To explore the relationship between payload mass and launch success, helping users identify any correlations.

Predictive Analysis (Classification)

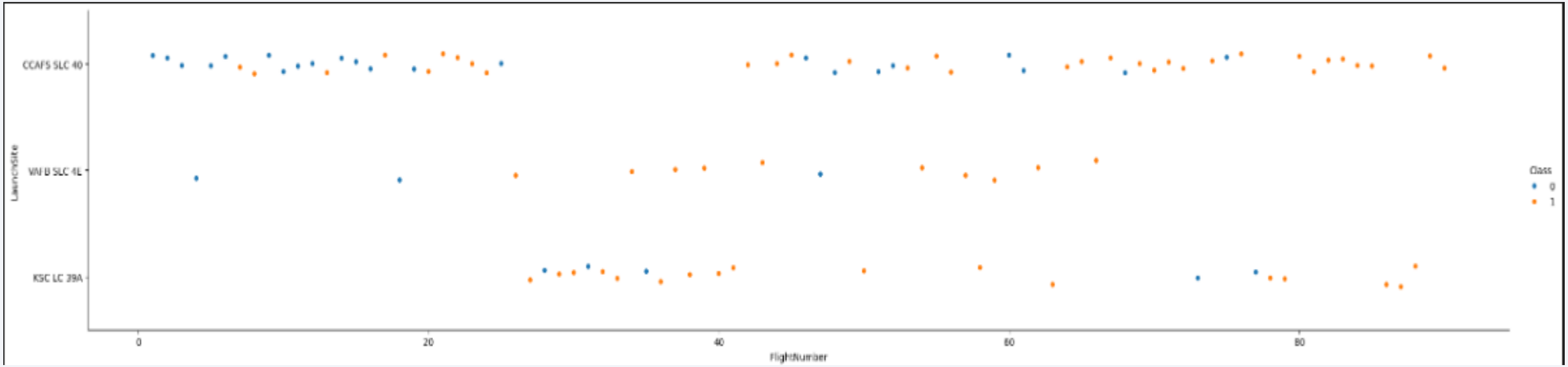
1. **Data Collection**: Gathered the dataset for the classification task.
2. **Data Preprocessing**:
    1. **Cleaning**: Handled missing values and outliers.
    2. **Encoding**: Converted categorical variables using one-hot encoding.
    3. **Splitting**: Divided data into training (80%) and testing (20%) sets.
3. **Model Selection**: Chose various algorithms:
    1. Logistic Regression
    2. Decision Trees
    3. Random Forest
    4. Support Vector Machines (SVM)
4. **Model Training**: Trained each model on the training dataset.
5. **Model Evaluation**:
    1. Used metrics: Accuracy, Precision, Recall, F1 Score, ROC-AUC.
    2. Conducted cross-validation for robustness.

Results

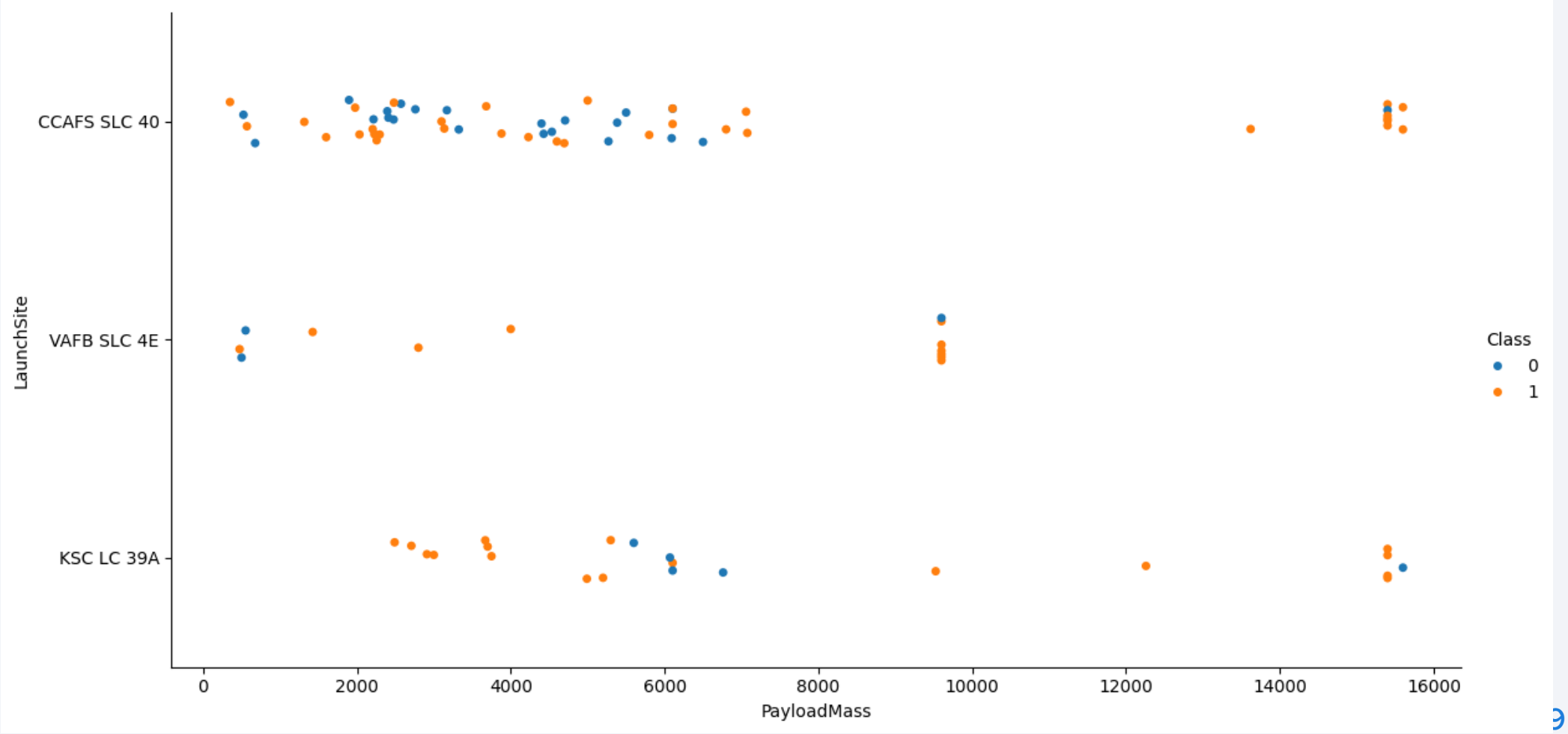
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

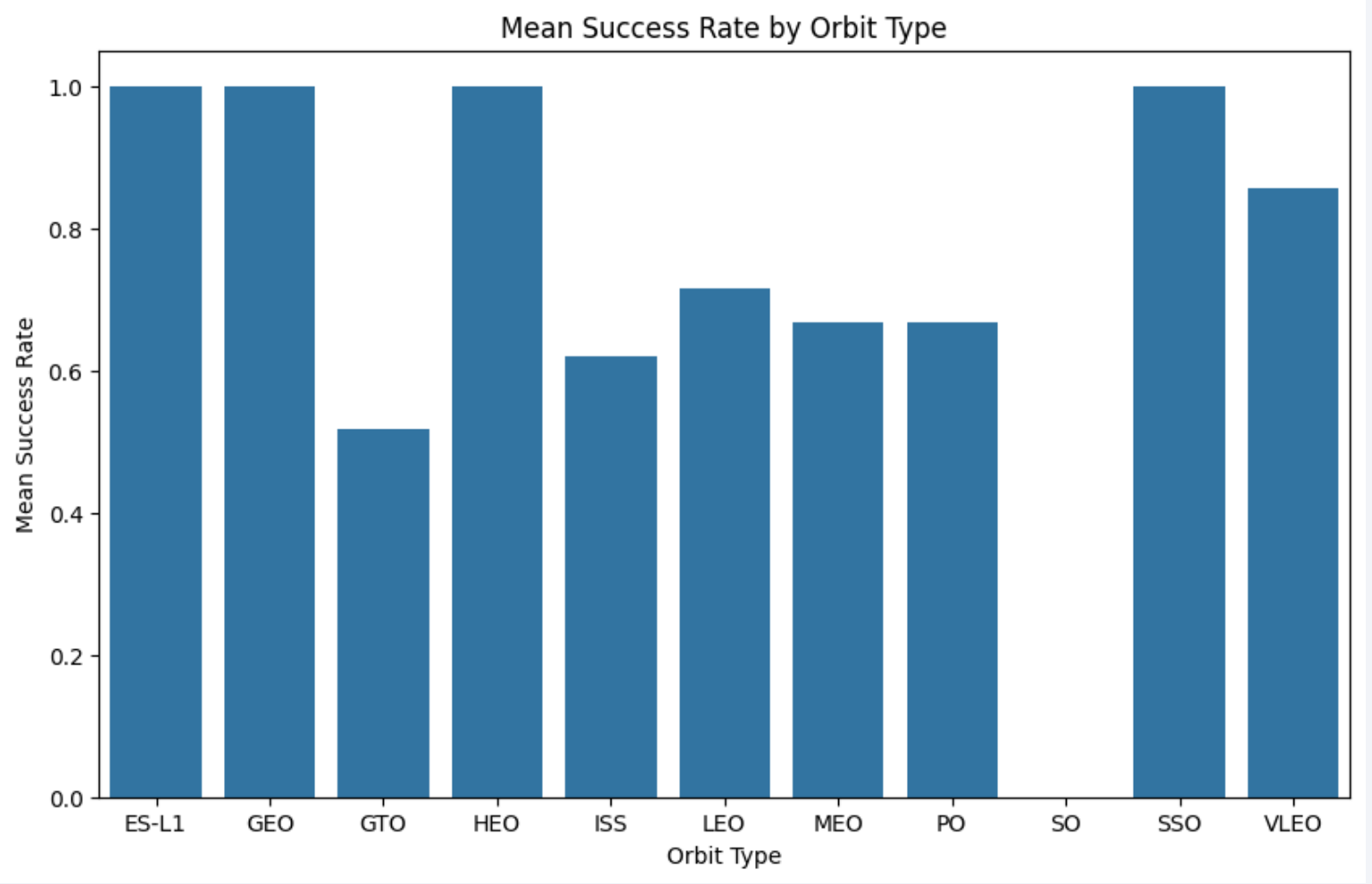
Flight Number vs. Launch Site



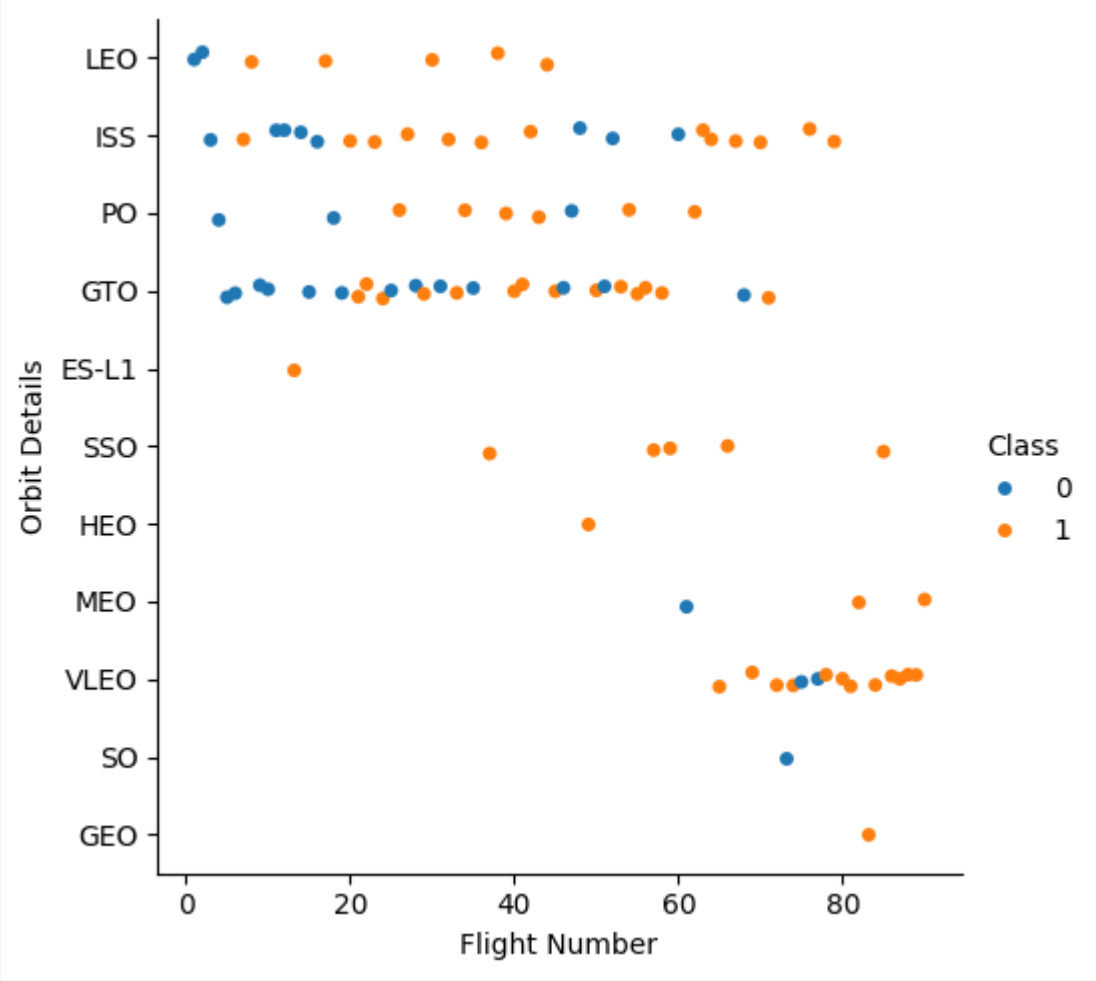
Payload vs. Launch Site



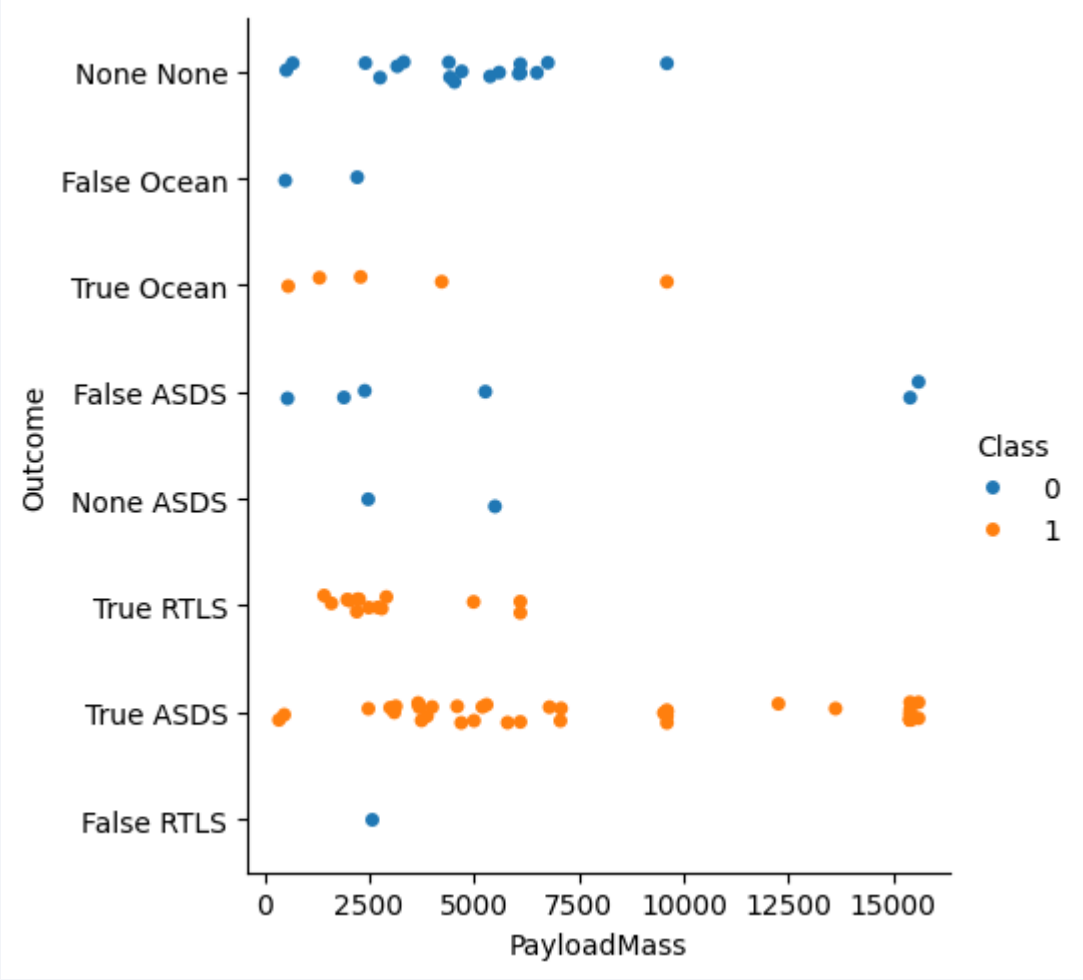
Success Rate vs. Orbit Type



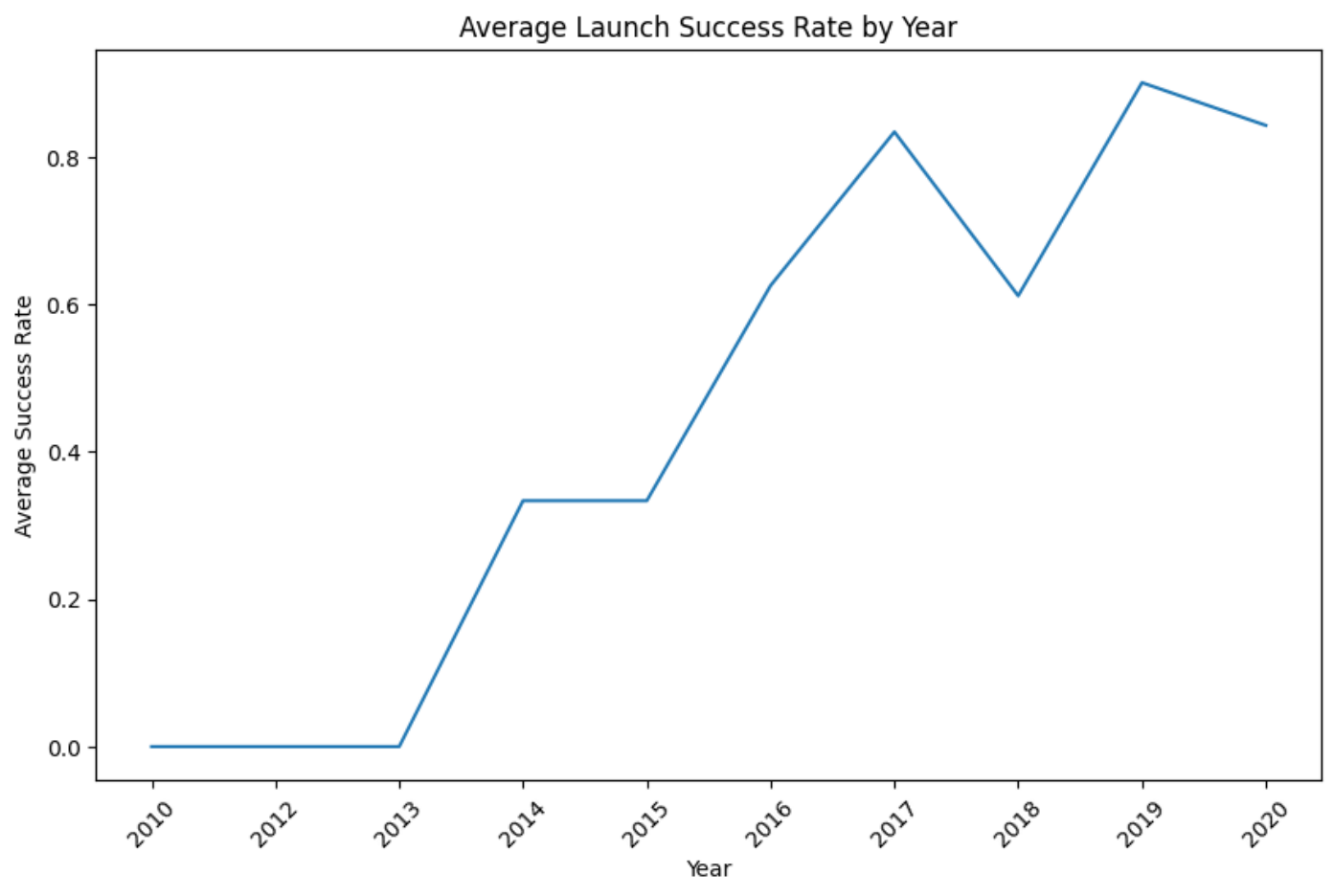
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

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

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

✓ 0.0s

* sqlite:///my_data1.db

Done.

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

Launch Site Names Begin with 'CCA'

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

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

✓ 0.0s Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

```
result = %sql SELECT * FROM SPACEXTBL;
print(result)
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

✓ 0.0s

ɔad	PAYLOAD_MASS__KG_	Orbit	Customer
ɔualification Unit	0	LEO	SpaceX
ɔats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO
flight C2	525	LEO (ISS)	NASA (COTS)
CRS-1	500	LEO (ISS)	NASA (CRS)
CRS-2	677	LEO (ISS)	NASA (CRS)
[OPE	500	Polar LEO	MDA
-8	3170	GTO	SES
ɔm 6	3325	GTO	Thaicom
CRS-3	2296	LEO (ISS)	NASA (CRS)
ɔomm-OG2 satellites	1316	LEO	Orbcomm
ɜt 8	4535	GTO	AsiaSat
ɜt 6	4428	GTO	AsiaSat
CRS-4	2216	LEO (ISS)	NASA (CRS)
CRS-5	2395	LEO (ISS)	NASA (CRS)
ƆVR	570	HEO	U.S. Air Force NASA NOAA
ɜt 115 West B	4159	GTO	ABS Eutelsat
CRS-6	1898	LEO (ISS)	NASA (CRS)
' MonacoSAT	4707	GTO	Turkmenistan National Space Agency
CRS-7	1952	LEO (ISS)	NASA (CRS)
ɔcomm-OG2 satellites	2034	LEO	Orbcomm

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

✓ 0.0s

* sqlite:///my_data1.db

Done.

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

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

Hint:Use min function

```
%sql SELECT MIN(DATE) AS First_Successful_Landing_Date FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)'
```

✓ 0.0s

Pythc

* sqlite:///my_data1.db
Done.

First_Successful_Landing_Date
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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAY
```

✓ 0.0s Python

```
 * 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

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS Total FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

[65] ✓ 0.0s

Pyth

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

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

Boosters Carried Maximum Payload

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

✓ 0.0s

Pyth

* 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

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.

```
    SUBSTR(DATE, 6, 2) AS Month, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure (dr
```

[71] ✓ 0.0s Python

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

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

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.

```
JTCOME, COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDE
```

✓ 0.0s Python

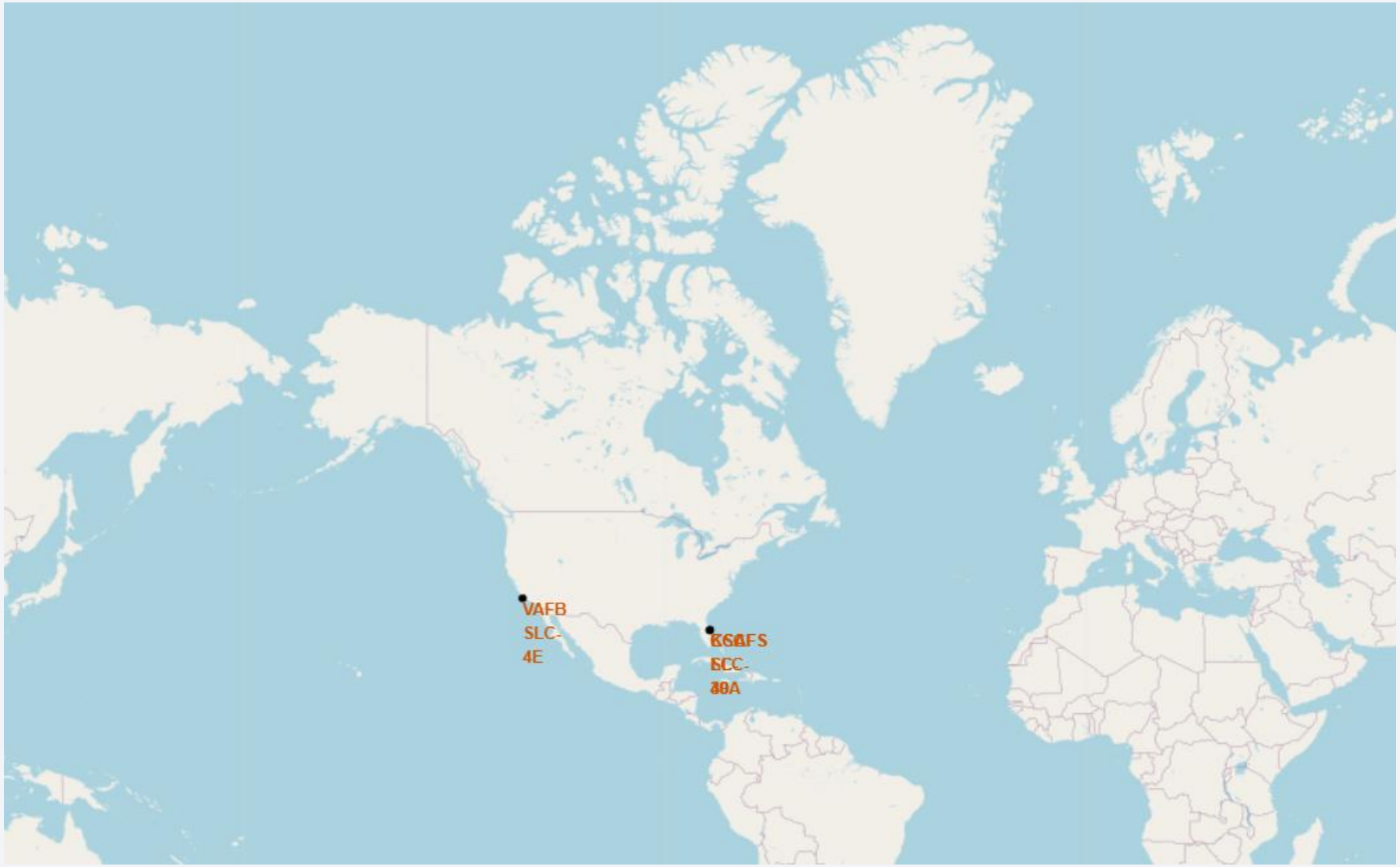
```
 * sqlite:///my_data1.db
Done.
```

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

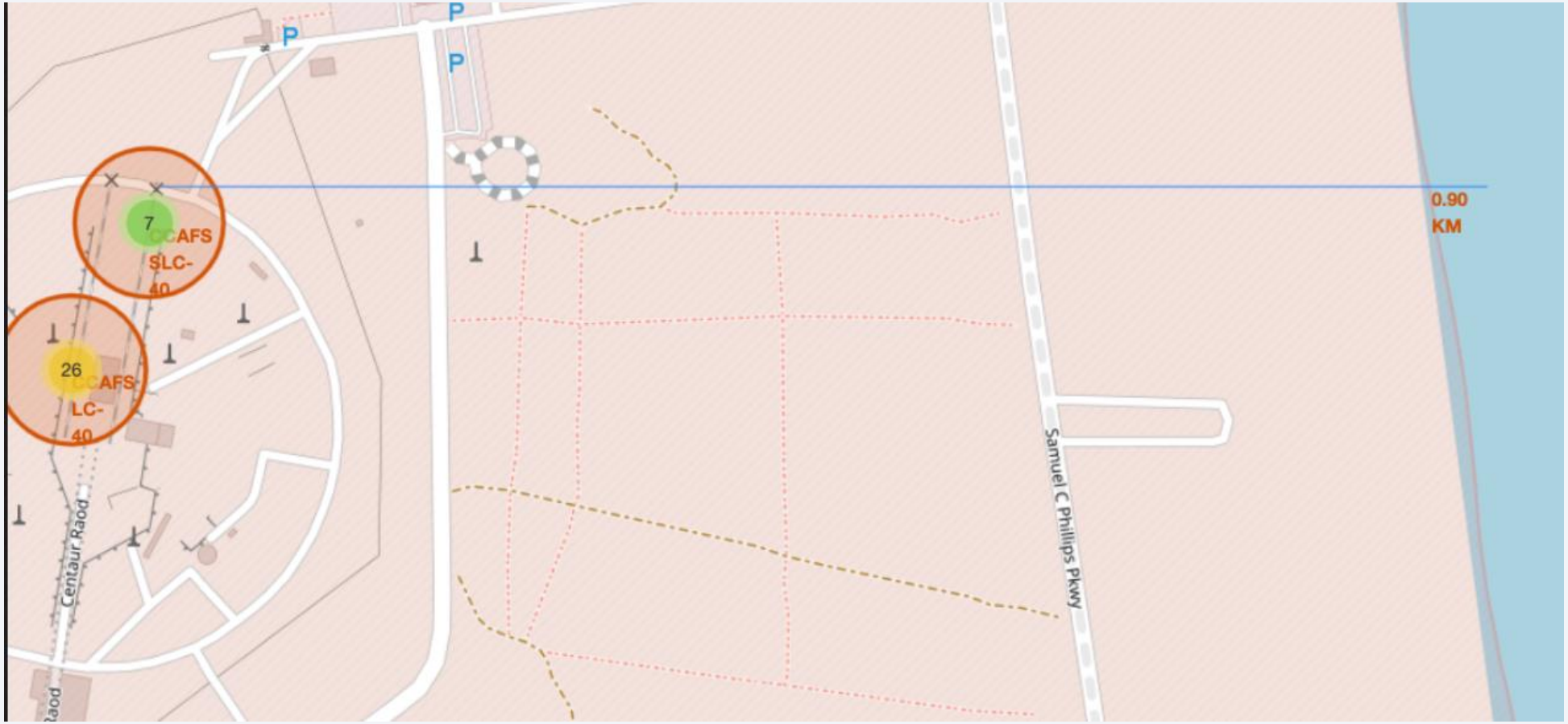
Launch Sites Proximities Analysis

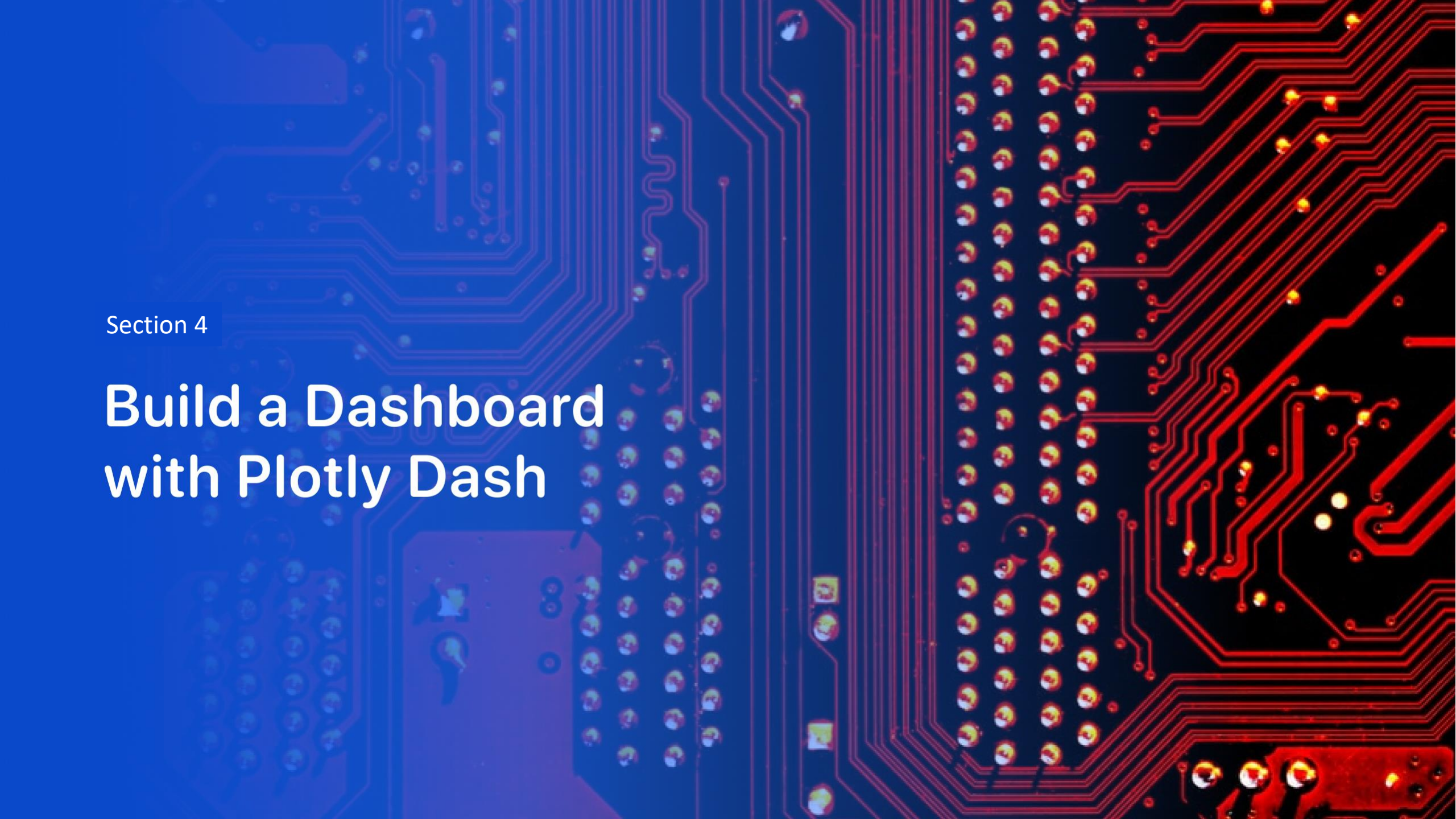
Mark all launch sites on a map



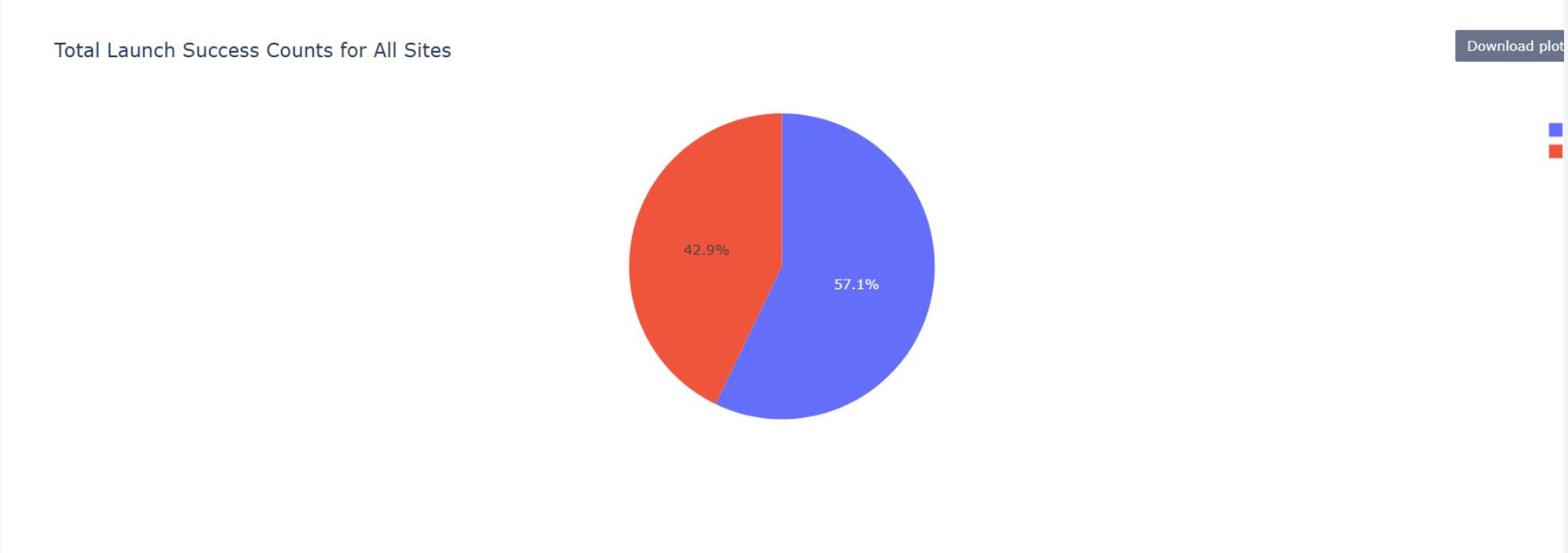
Mark the success/failed launches for each site on the map

Calculate the distances between a launch site to its proximities

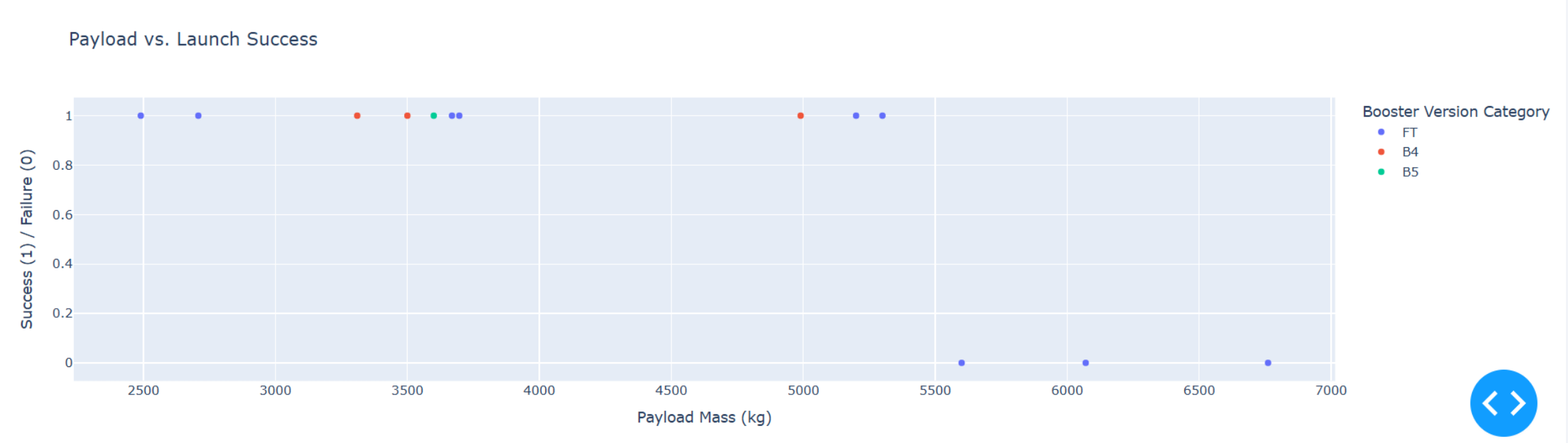



Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



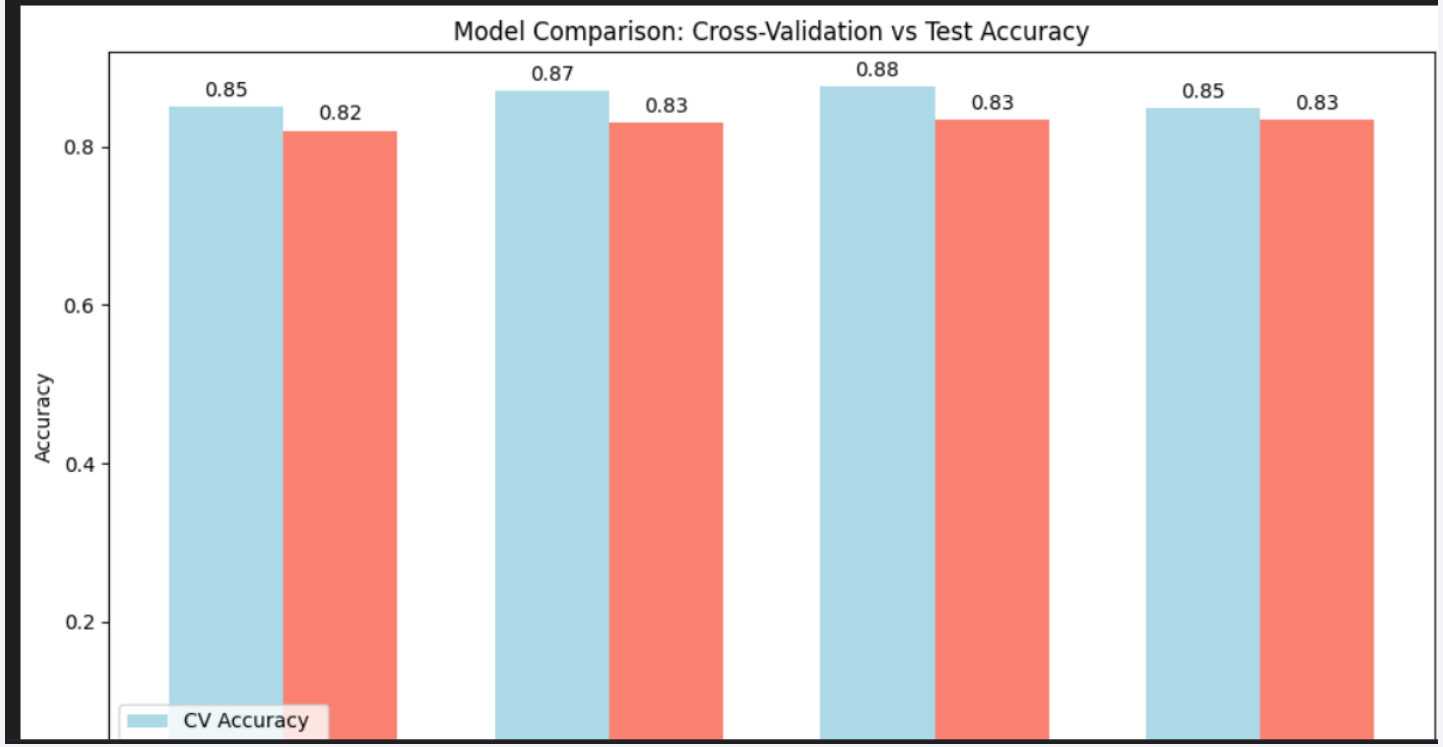
Section 5

Predictive Analysis (Classification)

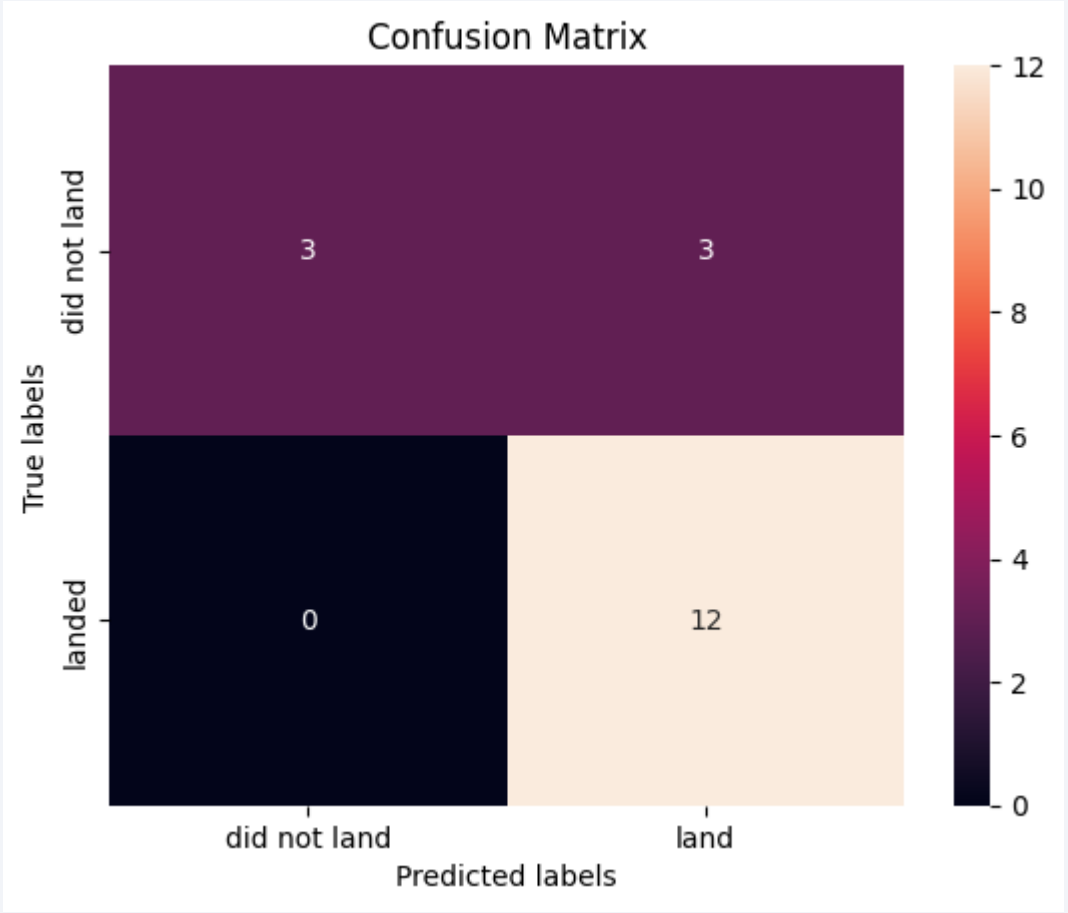
Classification Accuracy

	Model	CV Accuracy	Test Accuracy
0	Logistic Regression	0.850000	0.820000
1	Support Vector Machine	0.870000	0.830000
2	Decision Tree	0.875000	0.833333
3	K-Nearest Neighbors	0.848214	0.833333

The best performing model is: Decision Tree with a test set accuracy of 0.8333



Confusion Matrix



Conclusions

- We can conclude that:
  - The larger the flight amount at a launch site, the greater the success rate at a launch site.
  - Launch success rate started to increase in 2013 till 2020.
  - Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
  - KSC LC-39A had the most successful launches of any sites.
  - The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!