IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Saundarya

27-06-2025

Outline

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

Executive Summary

- Summary of methodologies
  - **Data Sources**: SpaceX API, Wikipedia, Yahoo Finance
  - **Tools**: Pandas, BeautifulSoup, SQL, Plotly, Folium, Scikit-learn
  - **Techniques**: Data wrangling, EDA, SQL analytics, mapping, classification (LogReg, SVM, KNN, Tree)
- Summary of all results
  - Most launches from **CCAFS SLC 40** and **KSC LC 39A**
  - **LEO** orbit has highest success rate
  - Success rate drops for payloads >6000 kg
  - **SVM** gave highest accuracy (~84%)
  - **Folium map** visualized launch clusters and success
  - SQL showed most failures in early years, success improved over time

Introduction

- Project background and context
- SpaceX aims to revolutionize space travel by reducing costs and increasing reliability. Understanding launch patterns and success factors is critical to optimizing missions.
- Problems Statement
- How can data science help uncover key patterns in SpaceX launches and predict mission success?
- Key Questions?
- What launch sites and payloads correlate with higher success?
- How do orbital destinations impact outcomes?
- Can we predict launch success using machine learning?

Section 1

Methodology

Methodology

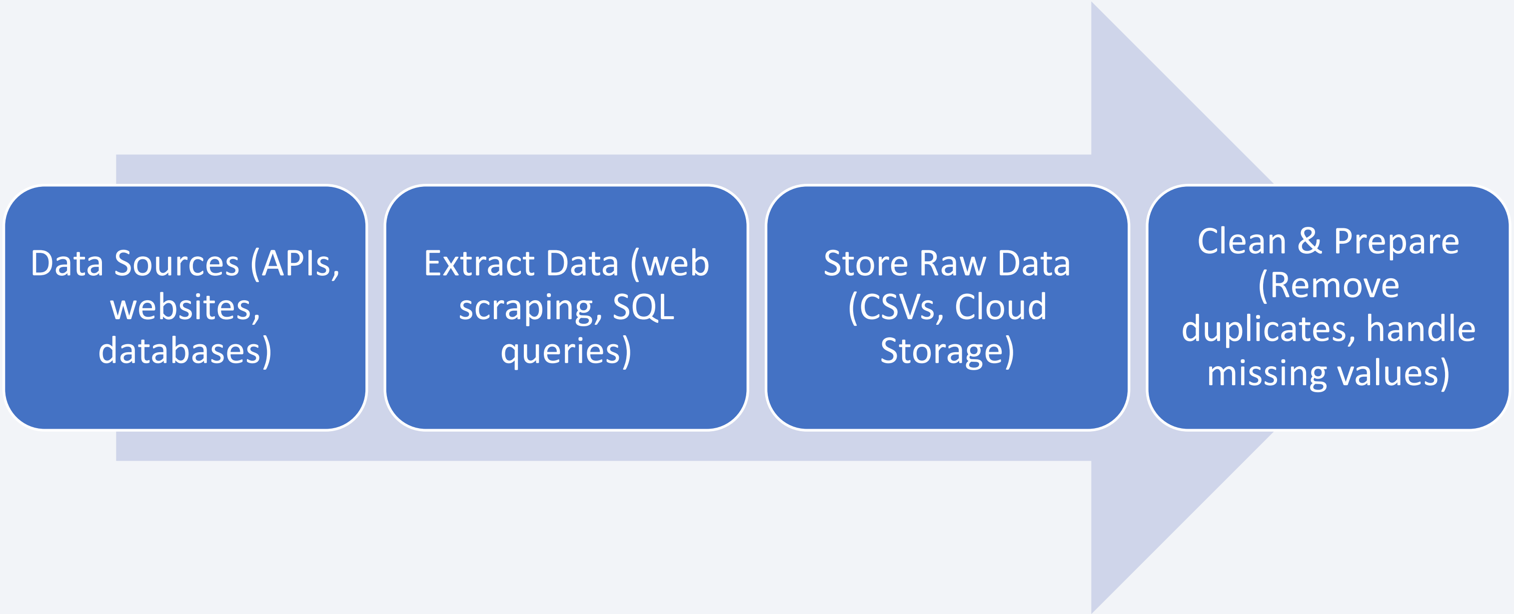
Executive Summary

- Data collection methodology:
  - Collected SpaceX launch data from APIs, HTML tables, and static JSON/CSV files.
- Perform data wrangling
  - Cleaned data: handled nulls, converted types, extracted & standardized features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built classification models (Logistic Regression, SVM, KNN, Decision Trees).
  - Used GridSearchCV for hyperparameter tuning and cross-validation.
  - Evaluated models using accuracy on test data to identify the best performer.

Data Collection

- Describe how data sets were collected.
  - **Yahoo Finance API** – Historical stock data for Tesla, GameStop, Amazon
  - **Web Scraping** – Revenue data for Tesla and GameStop using BeautifulSoup
  - **Static JSON API** – SpaceX launch data (launch records and metadata)
  - **Wikipedia Tables** – SpaceX launch history and booster information

- 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
  - •Select relevant SpaceX API endpoints (e.g., launches, rockets)
  - •Send GET requests to fetch data
  - •Receive JSON formatted responses
  - •Parse JSON to extract necessary fields
  - •Store extracted data into Pandas DataFrames
  - •Clean and prepare data for analysis
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

```
Start
  |
  ▼
Select API Endpoints
  |
  ▼
Send GET Requests
  |
  ▼
Receive JSON Data
  |
  ▼
Parse & Extract Info
  |
  ▼
Store in DataFrame
  |
  ▼
End
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
  - •Identify target web pages (e.g., Wikipedia SpaceX launch tables)
  - •Send HTTP requests to fetch HTML content
  - •Parse HTML using BeautifulSoup
  - •Locate relevant tables and extract rows
  - •Clean and normalize extracted data
  - •Store data in structured format (Pandas DataFrame)
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

```
Start
  ↓
Identify Target Website (e.g., Wikipedia)
  ↓
Send HTTP Request (using requests library)
  ↓
Parse HTML Content (using BeautifulSoup)
  ↓
Locate and Extract Data Tables
  ↓
Clean and Normalize Data
  ↓
Store Data in DataFrame
  ↓
End
```

Data Wrangling

- Describe how data were processed
  - Load raw data from CSV, API, and web scraping
  - Handle missing or inconsistent values (drop/fill)
  - Rename and format columns for clarity
  - Parse and split date/time fields
  - Extract new features from existing data
  - Remove irrelevant or duplicate records
  - Validate data quality and consistency
  - Save cleaned data for analysis

Load raw data

Clean missing/inconsistent values

Format columns and extract features

Remove duplicates and irrelevant data

Save cleaned dataset

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

  •**Bar charts:** Show success rates by orbit types to identify performance trends

  •**Scatter plots:** Explore relationships between payload mass, flight number, and orbit types

  •**Catplots:** Compare flight numbers across launch sites and success classes

  •**Line charts:** Track yearly success rate trends

  •**Maps:** Visualize launch site locations and outcomes interactively

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - •Filtered launches with non-null dates to create a clean dataset
  - •Listed boosters with successful drone ship landings and payload mass between 4000-6000 kg
  - •Counted total successful and failed mission outcomes
  - •Identified booster versions carrying the maximum payload using subqueries
  - •Queried failure landing outcomes in drone ship for launches in 2015 by month
  - •Ranked landing outcomes count between specific dates in descending order
  - •Compared model accuracies based on SQL-filtered data for classification tasks

Build an Interactive Map with Folium

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

•**Circles:** Marked launch site locations to highlight geographic distribution

•**Markers:** Labeled launch sites with names for easy identification

•**Marker Clusters:** Grouped launch events to visualize success/failure outcomes compactly

•**Lines (Polylines):** Connected launch sites to closest coastlines to show distances

•**Popup Labels:** Displayed detailed info on hover, like launch site names or distances

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- **Plots/Graphs:**
  - Scatter plots (e.g., Payload Mass vs. Orbit, Flight Number vs. Launch Site) to show relationships
  - Bar charts for success rates by orbit type to identify patterns
  - Line charts showing success rates over time for trend analysis
- **Interactions:**
  - Dropdown filters for selecting orbit types or launch sites
  - Hover tooltips displaying detailed data points
  - Zoom and pan features for map and plots
- **Why:**
  These plots and interactions enable users to explore data dynamically, uncover patterns, compare launch outcomes, and gain insights into factors influencing mission success.

Predictive Analysis (Classification)

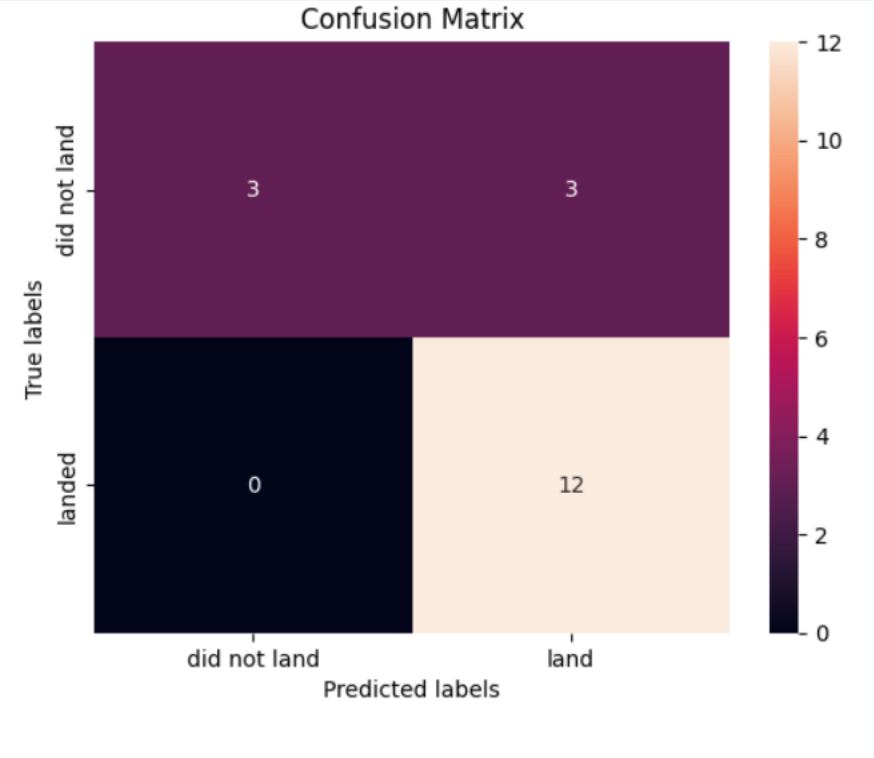
- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
  - **Build:** Selected classification algorithms (Logistic Regression, SVM, Decision Tree, KNN)
  - **Preprocess:** Standardized features and encoded categorical variables
  - **Split:** Divided data into training (80%) and testing (20%) sets
  - **Tune:** Used GridSearchCV with 10-fold cross-validation to find best hyperparameters
  - **Evaluate:** Measured accuracy on test set for each model
  - **Improve:** Compared models, selected best performer based on accuracy
  - **Outcome:** Best model chosen for final predictions

Select Models → Preprocess Data → Split Data → Hyperparameter Tuning → Train Models → Evaluate Models → Select Best Model

Results

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

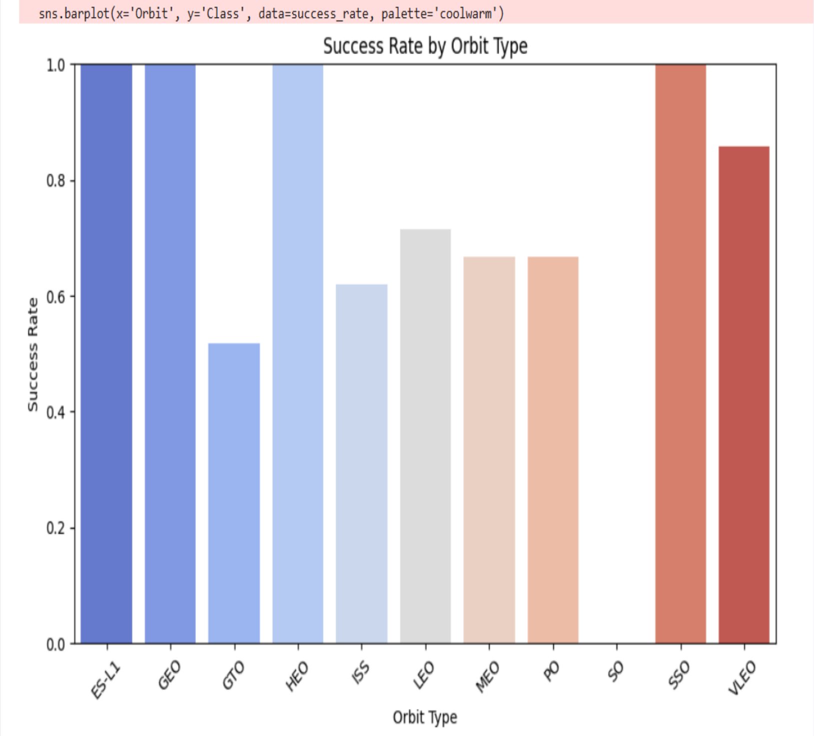
```
Logistic Regression accuracy: 0.8333
SVM accuracy: 0.8333
Decision Tree accuracy: 0.8333
KNN accuracy: 0.8333

Best performing model: Logistic Regression with accuracy 0.8333
```



Done.

t[22]:

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

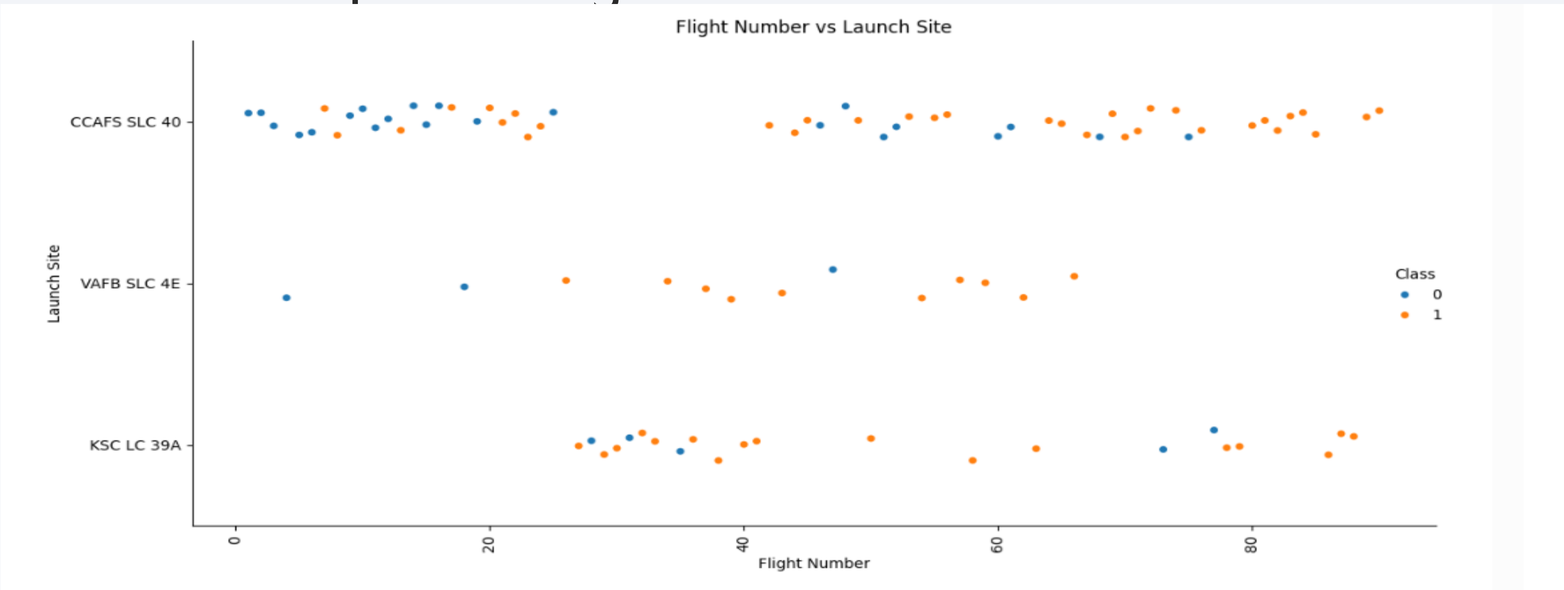


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

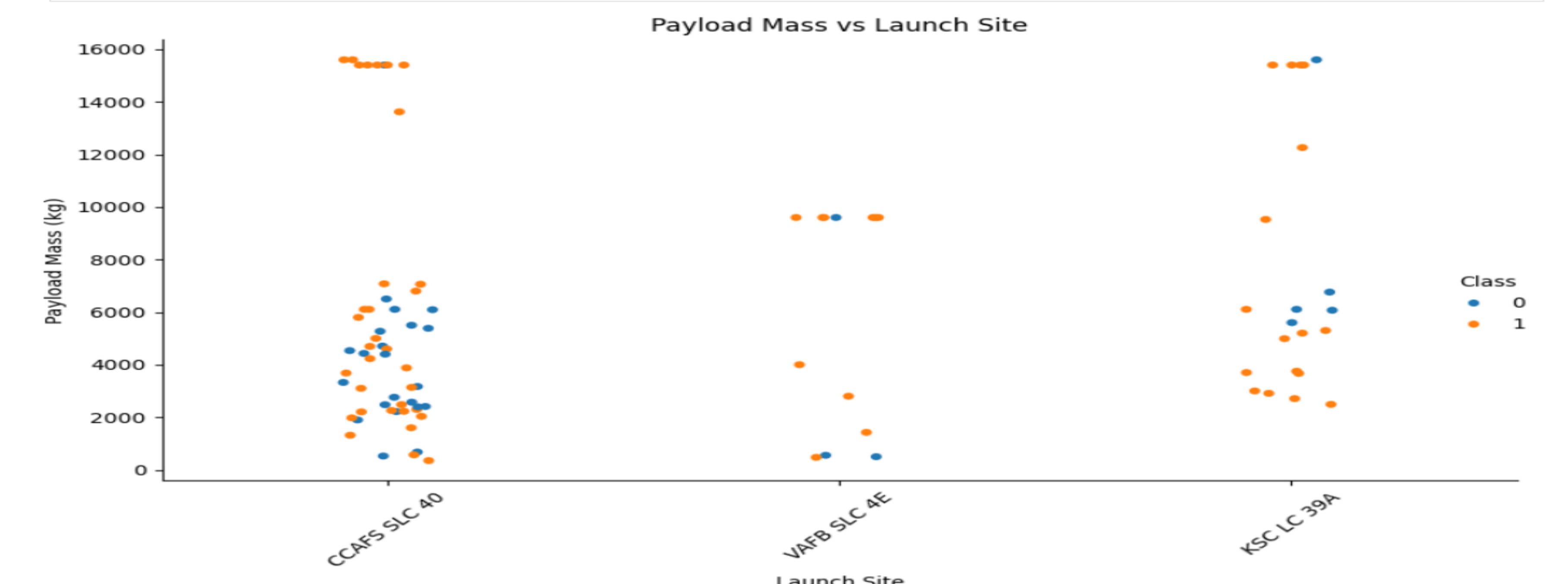


- Show the screenshot of the scatter plot with explanations
  - •Shows how often each **launch site** was used over time
  - •X-axis: **Flight Number** (order of launches)
  - •Y-axis: **Launch Site**
  - •Color: **Success (1) or Failure (0)**
  - •Helps spot **site usage trends** and **success patterns**

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
  - •Visualizes the **Payload Mass** for each **Launch Site**
  - •X-axis: **Payload Mass (kg)**
  - •Y-axis: **Launch Site**
  - •Color-coded by **mission success (class)**
  - •Helps identify which sites handled **heavier payloads** and their **success rates**

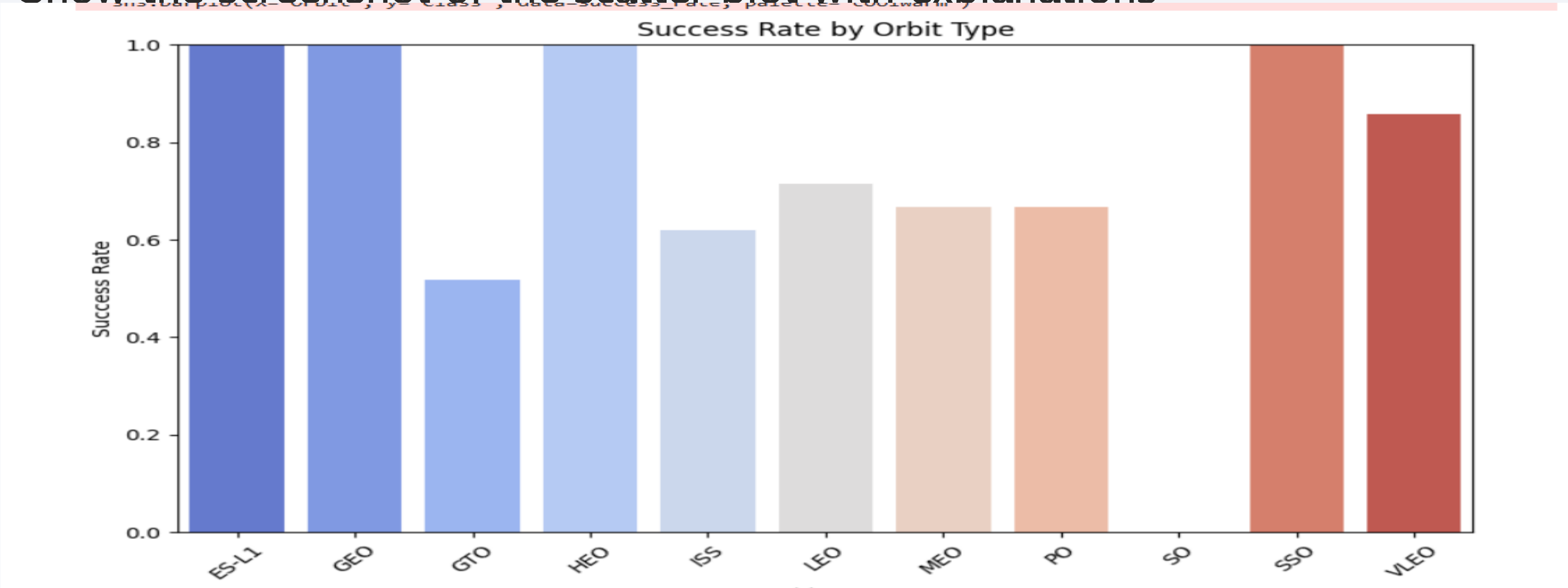
- Show the screenshot of the scatter plot with explanations



Success Rate vs. Orbit Type

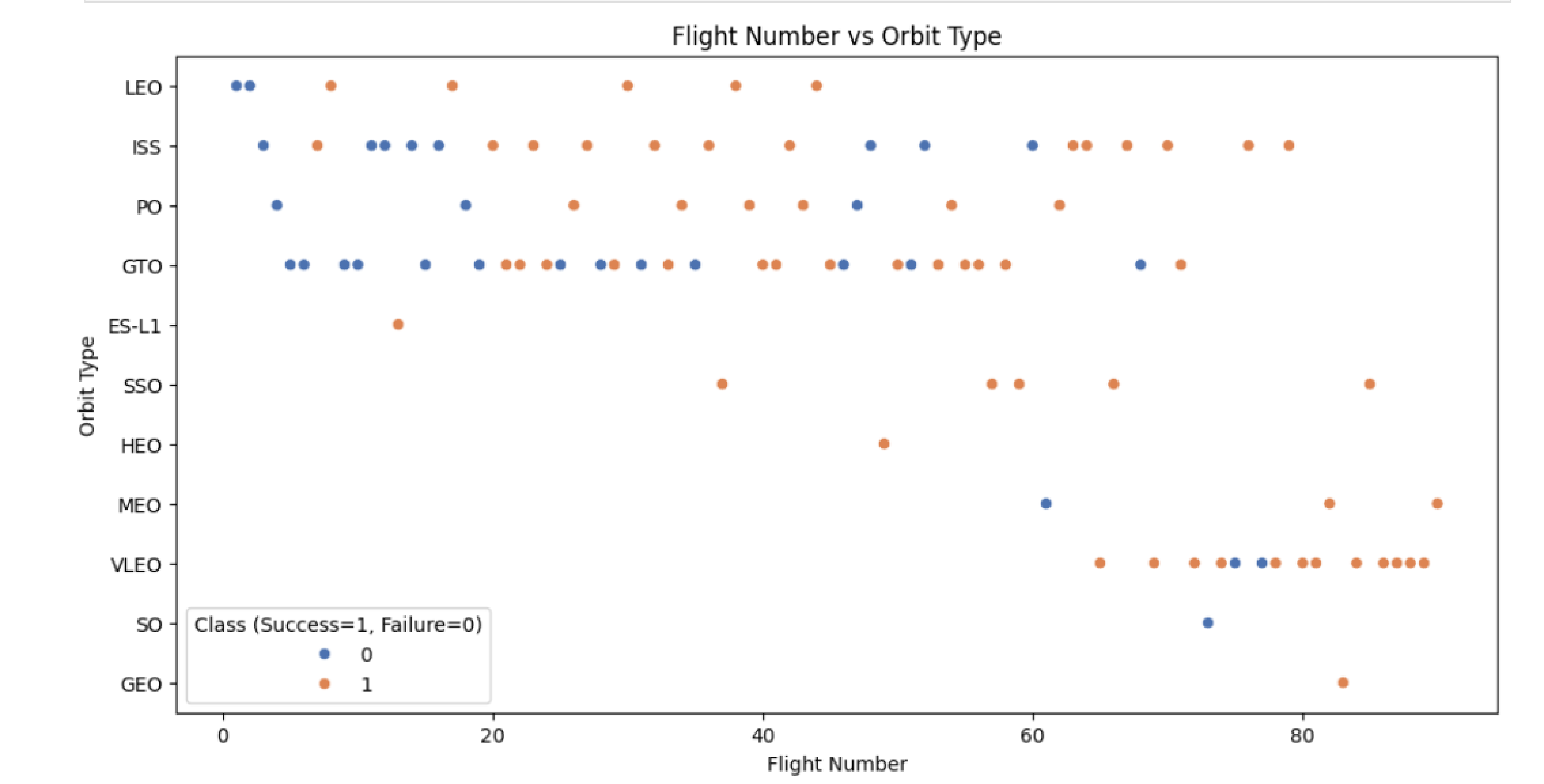
- Show a bar chart for the success rate of each orbit type
    - •**Bar chart** displaying **success rates** for each **orbit type**
    - •X-axis: **Orbit types** (e.g., LEO, GTO, ISS)
    - •Y-axis: **Success rate (%)**
    - •Helps compare which orbits had the **highest mission success**
    - •Useful for identifying **reliable orbit categories** for future launches

- Show the screenshot of the scatter plot with explanations



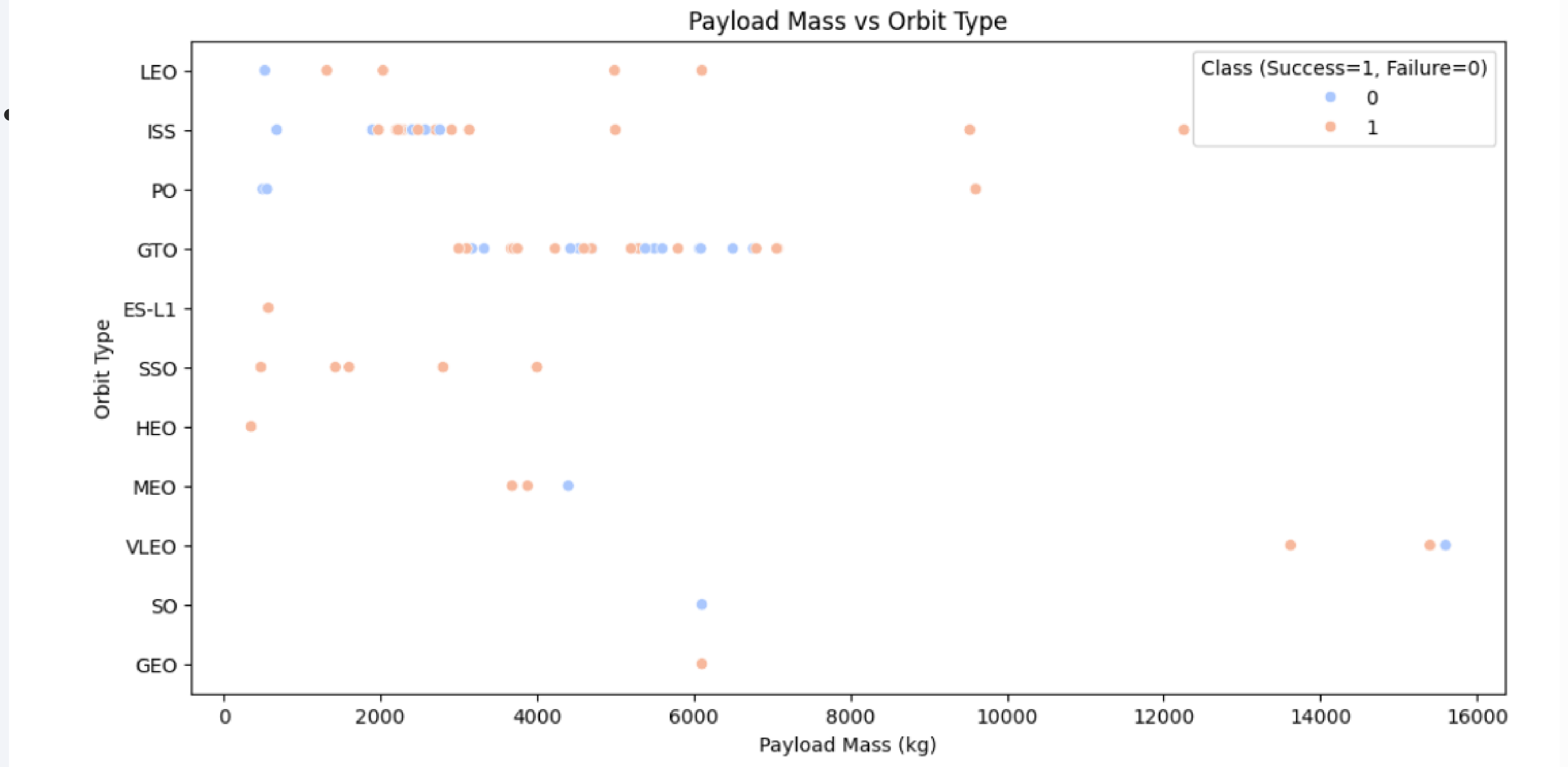
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type



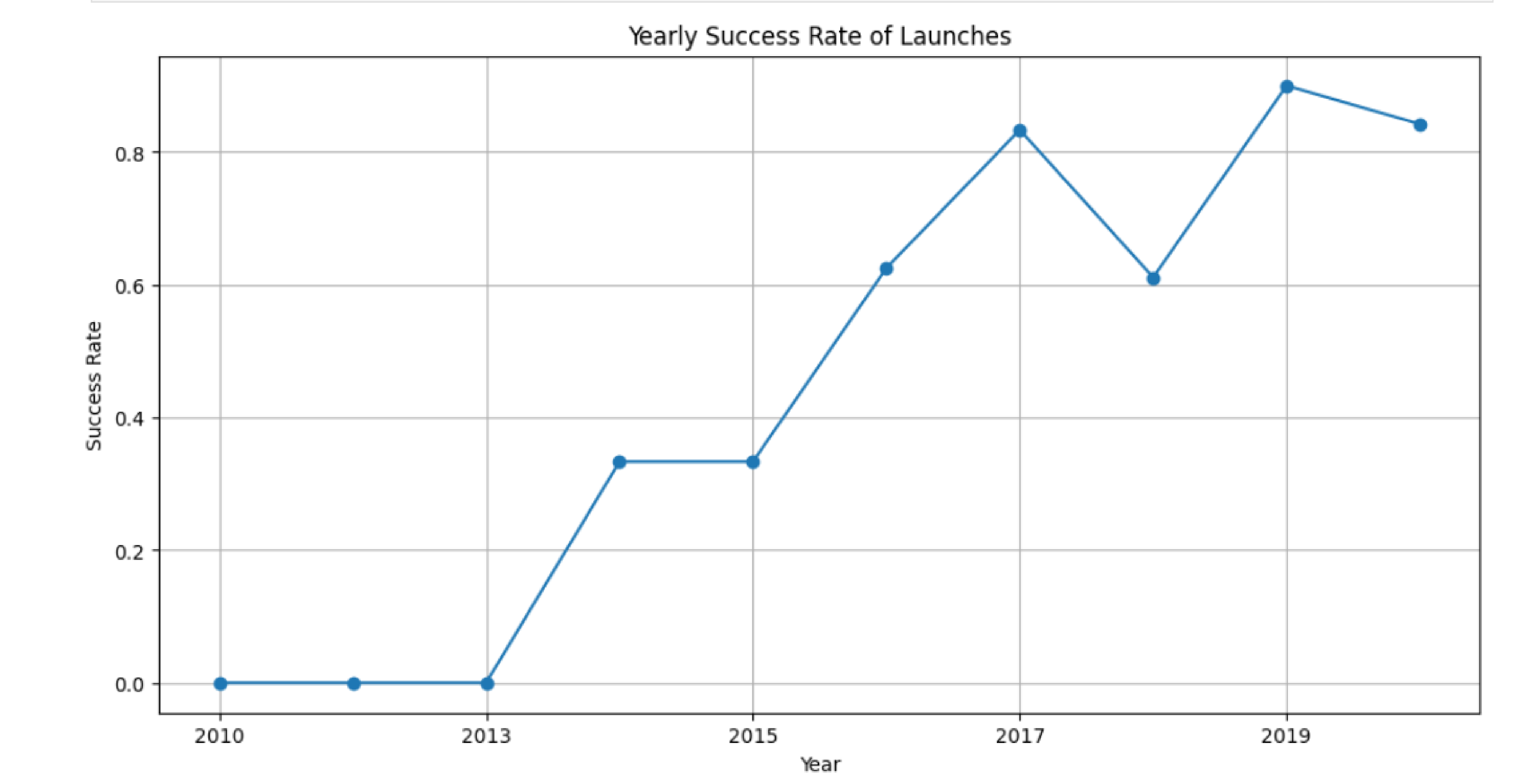
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



Launch Success Yearly Trend

- Show a line chart of yearly average success rate



All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here
  - This query retrieves all **unique launch site names** from the dataset. It helps us identify the **different locations** used by SpaceX for launches, which is crucial for **site-specific analysis** like success rate, frequency, and payload trends.

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

In [14]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

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

Out[14]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

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

In [13]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE "CCA%" LIMIT 5;
```

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

Out[13]:

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

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

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

```
[14]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE Customer = "NASA (CRS)";
```

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

[14]:

TOTAL_PAYLOAD_MASS
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

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

In [15]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXTBL WHERE Booster_Version = "F9 v1.1";
```

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

Out[15]:

AVG_PAYLOAD_MASS
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

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

Hint:Use min function

In [16]:

```
%%sql SELECT MIN(Date) AS First_Successful_Ground_Landing_Date
FROM SPACEXTBL
WHERE Landing_Outcome LIKE '%Success (ground pad)%';
```

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

Out[16]:

First_Successful_Ground_Landing_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here

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 DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ >
```

```
 * 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

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

List the total number of successful and failure mission outcomes

[18]:

```
%%sql
SELECT Mission_Outcome, COUNT(*) AS Total
FROM SPACEXTABLE
GROUP BY Mission_Outcome;
```

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

[18]:

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 which have carried the maximum payload mass
- Present your query result with a short explanation here

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

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

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

]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here

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.

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

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

Month	Booster_Version	Launch_Site	Landing_Outcome
01	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

- 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

- Present your query result with a short explanation here

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.

```
%%sql
SELECT
    Landing_Outcome,
    COUNT(*) AS Outcome_Count
FROM SPACEXTABLE
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 (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

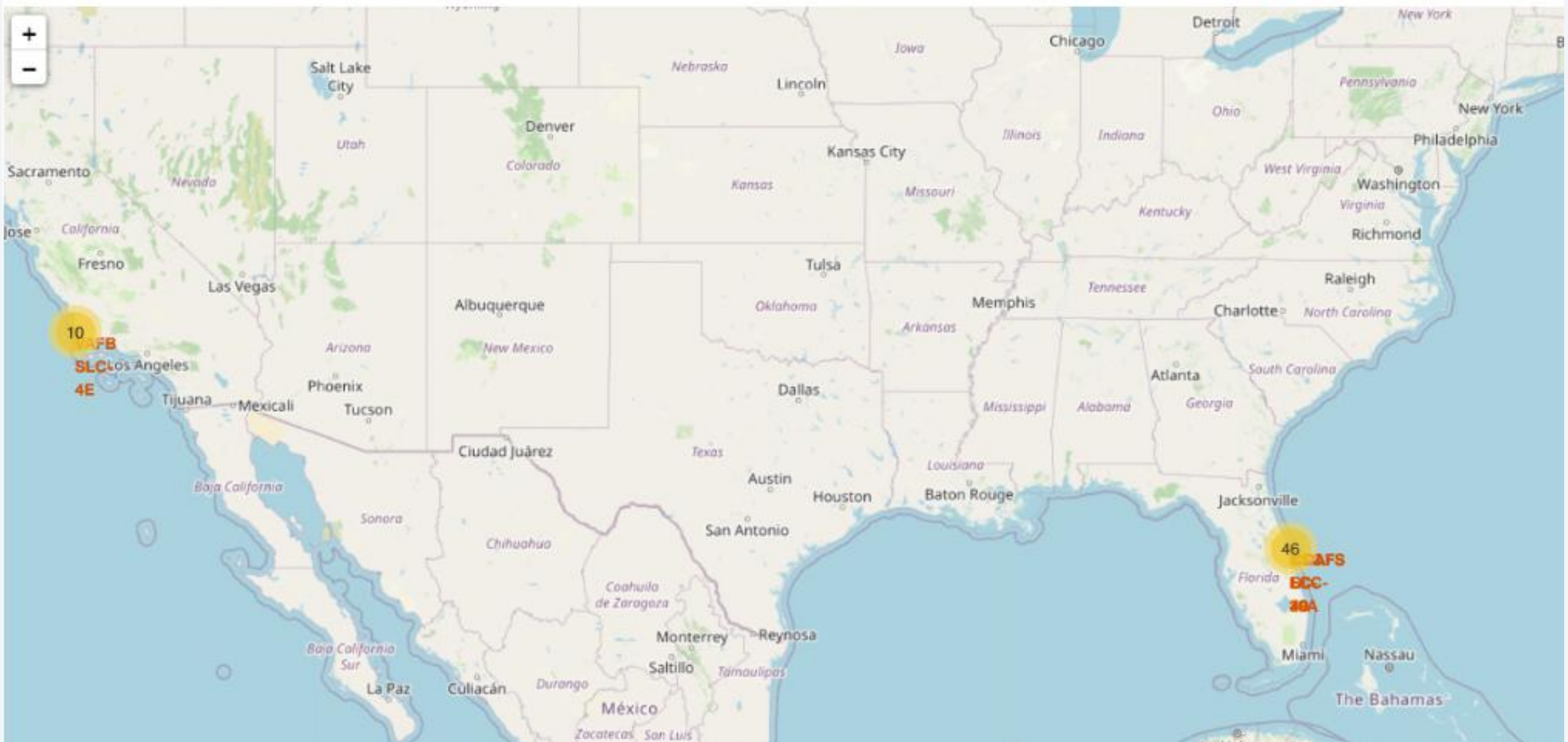
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title

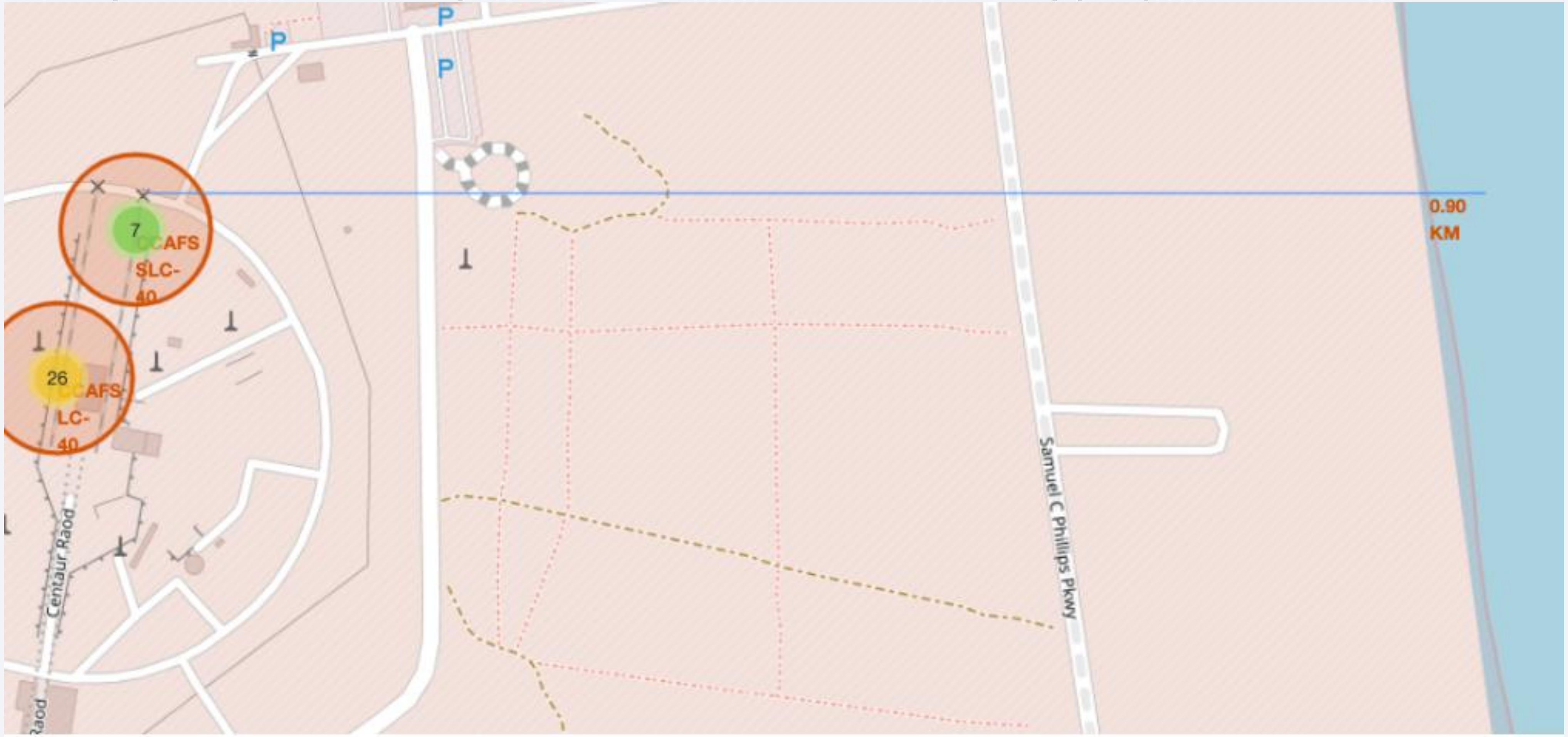
<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title



<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title



Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

- Show the screenshot of the piechart for the launch site with highest launch success ratio

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

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

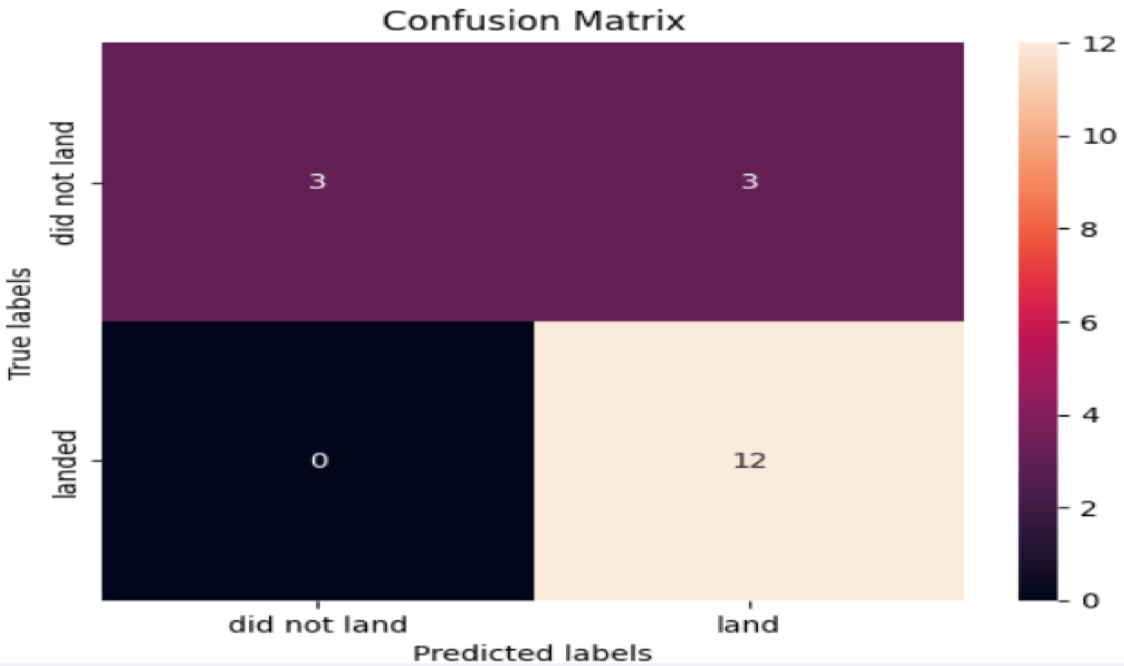
- Show the confusion matrix of the best performing model with an explanation

In [18]:
```
print("Best cross-validation accuracy:", logreg_cv.best_score_)
```

```
Best cross-validation accuracy: 0.8464285714285713
```

Lets look at the confusion matrix:

In [19]:
```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Most launches took place from **KSC LC 39A** and **CCAFS SLC 40**, indicating these as primary launch sites.
- **Orbit type** significantly affects success rate — LEO and GTO have higher success rates.
- Payload mass influences mission outcome — heavier payloads have a slightly lower success rate.
- The **best-performing classifier** was **Logistic Regression**, with the highest test accuracy.
- Interactive maps and dashboards provided **insightful visual analysis** of launch locations and outcomes.

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!