IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kaan Durmuş
15 July 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- **Summary of methodologies**
  - Data Collection
  - Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Data Visualization
  - Launch Sites Analysis with Folium-Interactive Visual Analytics and Ploty Dash
- **Summary of all results**
  - Exploratory Data Analysis (EDA) Results
  - Interactive Visual Analytics and Dashboards
  - Predictive Modeling Results (Classification)

Introduction

- **Project background and context**

SpaceX offers Falcon 9 rocket launches at a cost of 62 million dollars, which is significantly lower than the 165 million dollars charged by other providers. This cost efficiency is primarily due to the reusability of the Falcon 9's first stage. Accurately predicting the first stage landing can provide a better understanding of the launch costs and help companies competing with SpaceX.

- **Problems you want to find answers**

Can we develop a reliable model to predict the success of the Falcon 9 first stage landing?

Which factors are most critical in determining the success of the first stage landing?

How does the accuracy of predictions improve with the use of historical launch data?

How can these predictions be leveraged to optimize launch strategies and costs?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using the SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Data was cleaned and processed using Pandas to handle missing values and prepare it for analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used GridSearchCV to tune hyperparameters and evaluate model performance.

Data Collection

- Data was collected using the SpaceX API and web scraping from Wikipedia.
- SpaceX API:

  Fetch Data: Used requests to make GET requests to SpaceX API endpoint.

  Example: https://api.spacexdata.com/v4/launches

  Parse JSON Response: Parsed the JSON response using the json module.

  Convert to DataFrame: Imported JSON data into a Pandas DataFrame.
- Web Scraping:

  Fetch HTML Content: Used requests to get HTML content from Wikipedia.

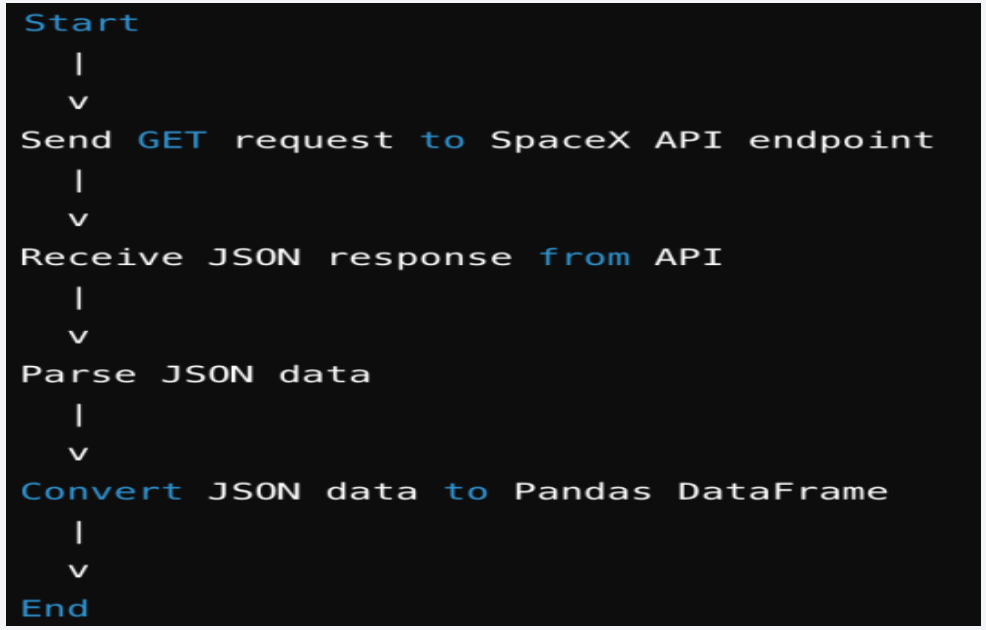
  Example: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

  Parse HTML with BeautifulSoup: Parsed HTML to find tables with BeautifulSoup.

  Extract Data from HTML Tables: Extracted data from table rows and cells.

  Convert to DataFrame: Imported the extracted data into a Pandas DataFrame.

Data Collection – SpaceX API



- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/1.CollectingData.ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url)
json_data = response.json()
data = pd.json_normalize(json_data)
```

Data Collection - Scraping

- Executed web scraping to gather historical Falcon 9 launch records from Wikipedia. Utilized BeautifulSoup and requests libraries to extract the Falcon 9 launch data from HTML tables on the Wikipedia page, then transformed the extracted data into a DataFrame by parsing the HTML content.
- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/2.WebScraping.ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url)
json_data = response.json()
data = pd.json_normalize(json_data)
```

Data Wrangling

- After collecting the data and creating a Pandas DataFrame, the data was filtered by the BoosterVersion column to retain only the Falcon 9 launches. Missing values in the LandingPad and PayloadMass columns were addressed, with missing PayloadMass values being filled with the column's mean value.

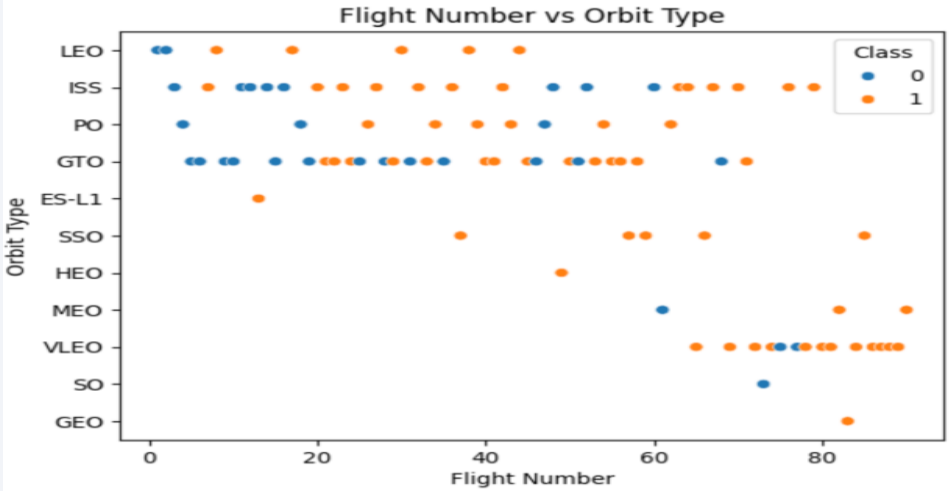
- Additionally, exploratory data analysis (EDA) was conducted to identify patterns within the data and to determine the appropriate labels for training supervised models.

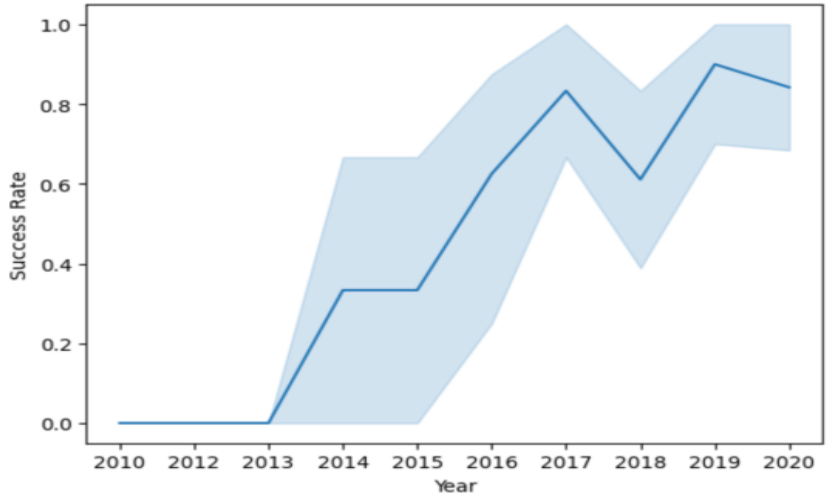
- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/3.DataWrangling.ipynb

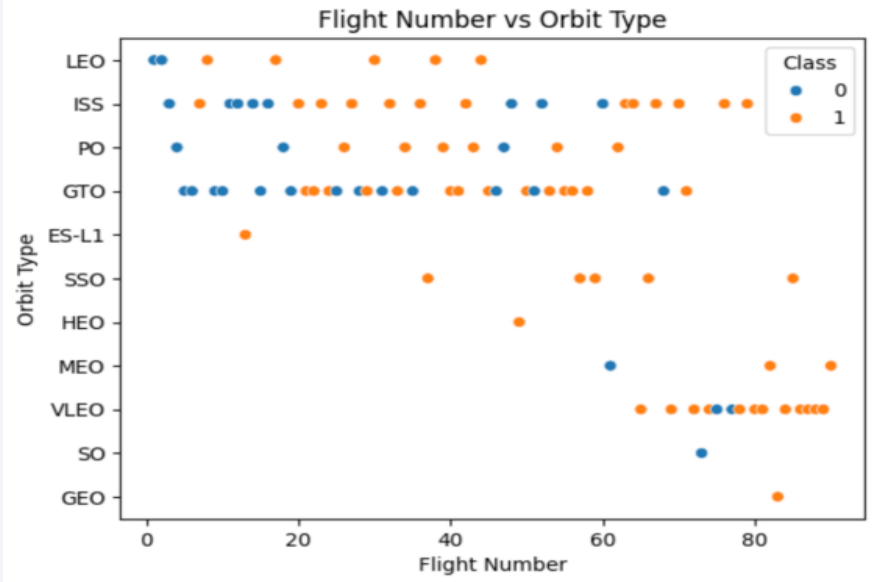
EDA with Data Visualization

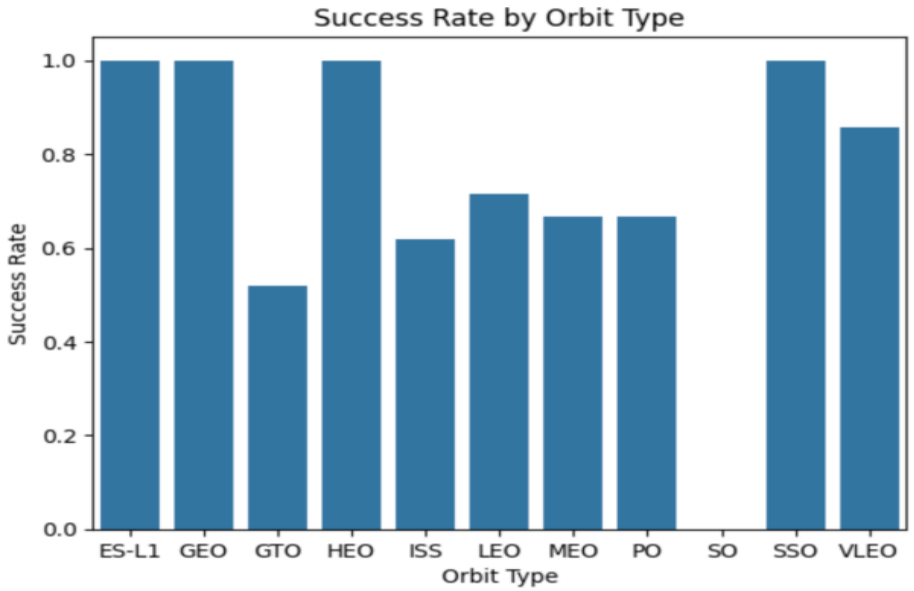
- Conducted data analysis and feature engineering using Pandas and Matplotlib.
- Performed Exploratory Data Analysis (EDA).
- Prepared data for feature engineering.
- Utilized scatter plots to visualize relationships between:
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit type
- Payload and Orbit type
- Employed bar charts to visualize the success rate for each orbit type.
- Created line plots to visualize the yearly trend of launch success.
- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/4.EDA-SQL.ipynb

EDA with SQL


Flight Number vs Orbit Type
Class
0
1
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
Orbit Type
0
20
40
60
80
Flight Number


1.0
0.8
0.6
0.4
0.2
0.0
Success Rate
2010
2012
2013
2014
2015
2016
2017
2018
2019
2020
Year


Flight Number vs Orbit Type
Class
0
1
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
Orbit Type
0
20
40
60
80
Flight Number


Success Rate by Orbit Type
1.0
0.8
0.6
0.4
0.2
0.0
Success Rate
ES-L1
GEO
GTO
HEO
ISS
LEO
MEO
PO
SO
SSO
VLEO
Orbit Type

Build an Interactive Map with Folium

- Developed a Folium map to display all the launch sites, including map objects such as markers, circles, and lines to indicate the success or failure of launches at each site.
- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/5.EDA-Dataviz.ipynb

Build a Dashboard with Plotly Dash

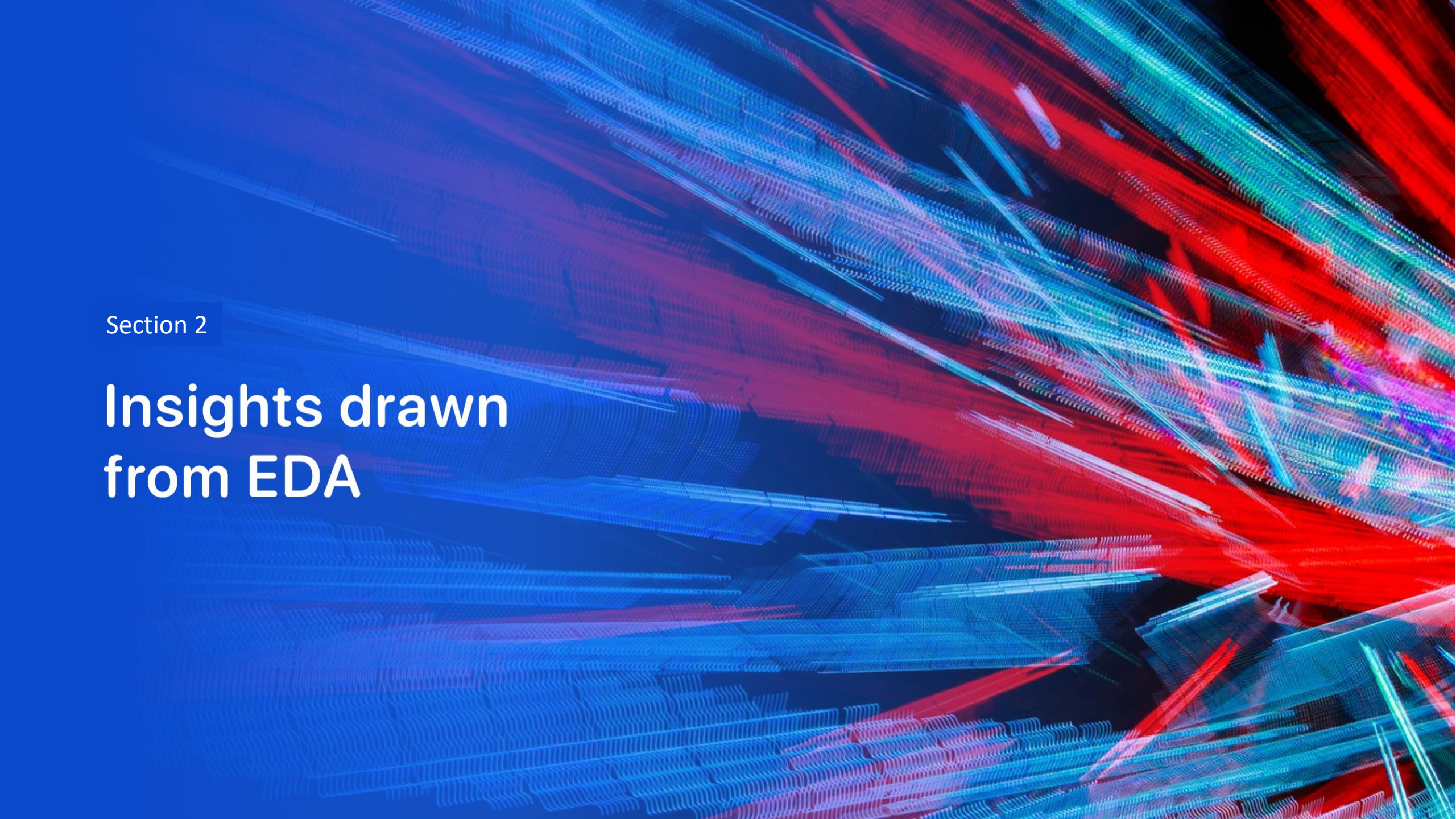
- Created an interactive dashboard application with Plotly Dash by:
  - Incorporating a Launch Site drop-down input component
  - Adding a callback function to render a success pie chart based on the selected site from the drop-down
  - Including a range slider to select payload
- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/7.spacex_dash_app.py

Predictive Analysis (Classification)

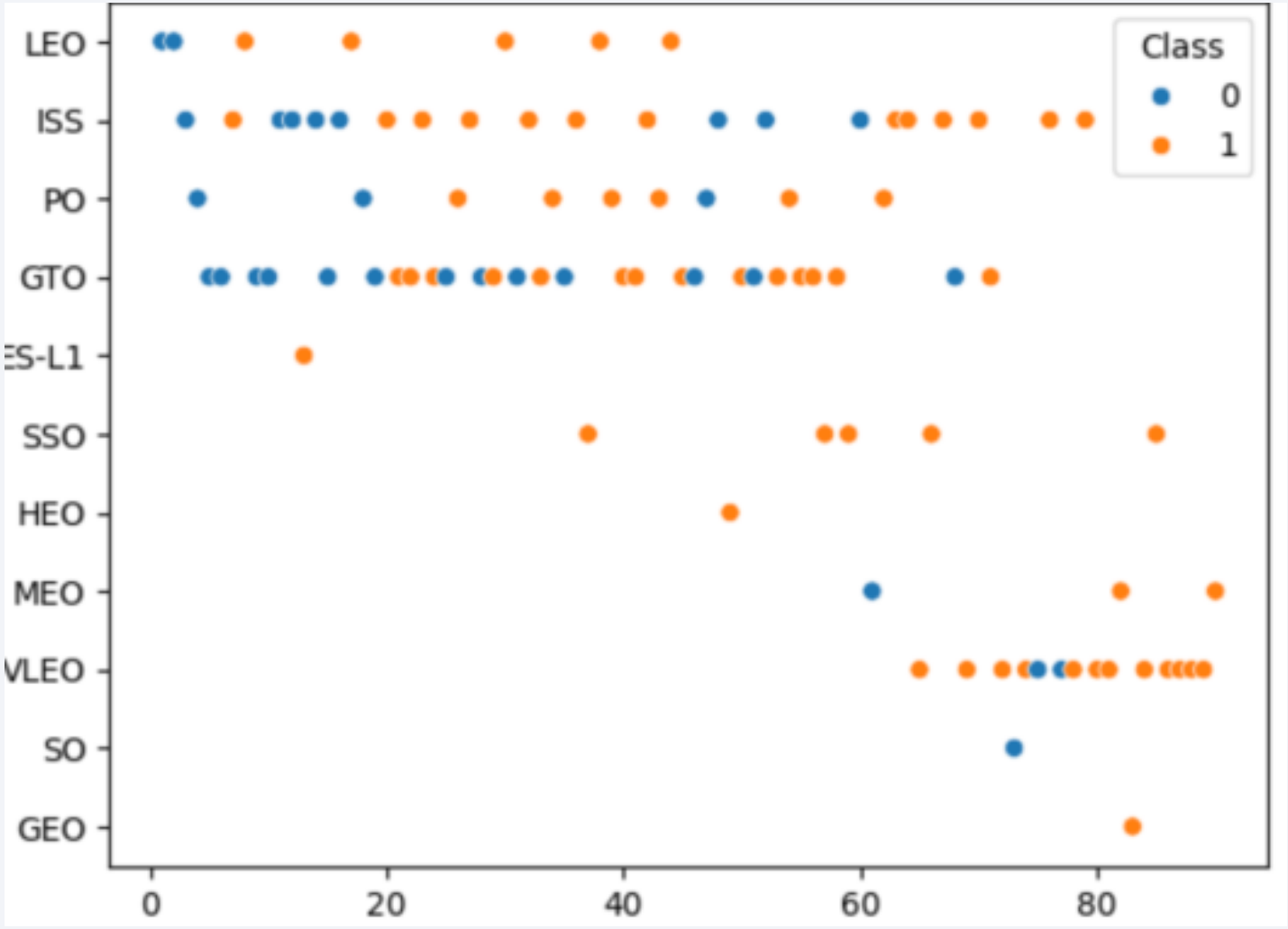
- To identify the best-performing ML model/method among SVM, Classification Trees, k-Nearest Neighbors, and Logistic Regression:
- Created an object for each algorithm, then initialized a GridSearchCV object with a set of parameters for each model.
- For each model under evaluation, configured the GridSearchCV object with cv=10 and fit the training data into the GridSearch object to find the best hyperparameters.
- After training, output the GridSearchCV object for each model and displayed the best parameters using the bestparams attribute and the accuracy on the validation data using the bestscore attribute.
- Finally, calculated the accuracy on the test data for each model using the score method and plotted a confusion matrix for each using the test and predicted outcomes.
- https://github.com/kaandurmus23/Space_X_Ibm/blob/main/8.SpaceX_Machine_Learning.ipynb

Results

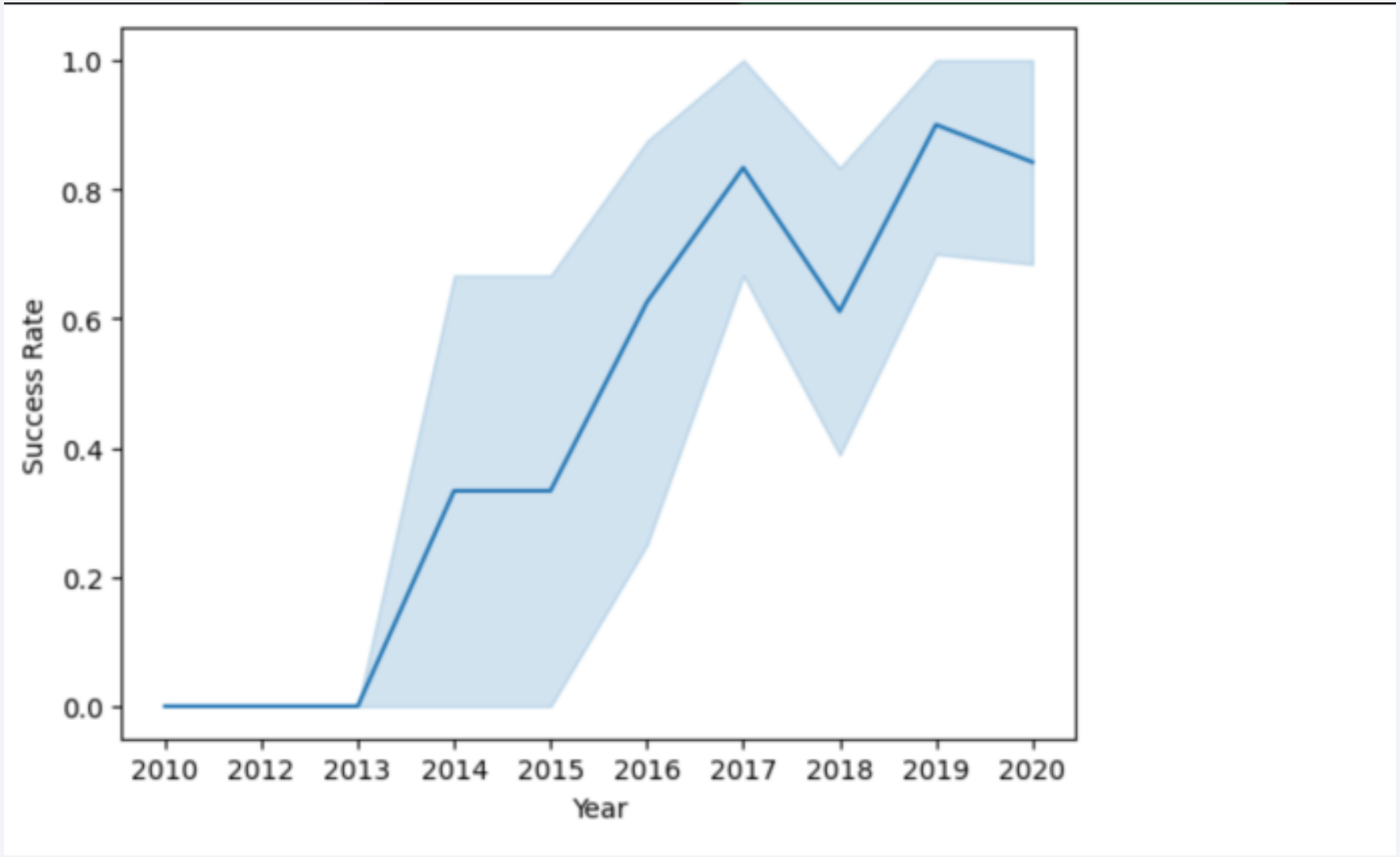
- Exploratory Data Analysis (EDA) Results:
- Conducted in-depth analysis to identify patterns and insights within the data.
- Visualized relationships between various features using scatter plots, bar charts, and line plots.
- Interactive Analytics Demo:
- Provided screenshots showcasing the interactive dashboard and visualizations created using Folium and Plotly Dash.
- Demonstrated the use of drop-down menus, sliders, and interactive plots to explore launch data.
- Predictive Analysis Results:
- Evaluated the performance of various machine learning models (SVM, Classification Trees, k-Nearest Neighbors, Logistic Regression).
- Identified the best hyperparameters and assessed model accuracy using GridSearchCV.
- Presented confusion matrices and accuracy scores for each model based on test data.


Section 2
Insights drawn from EDA

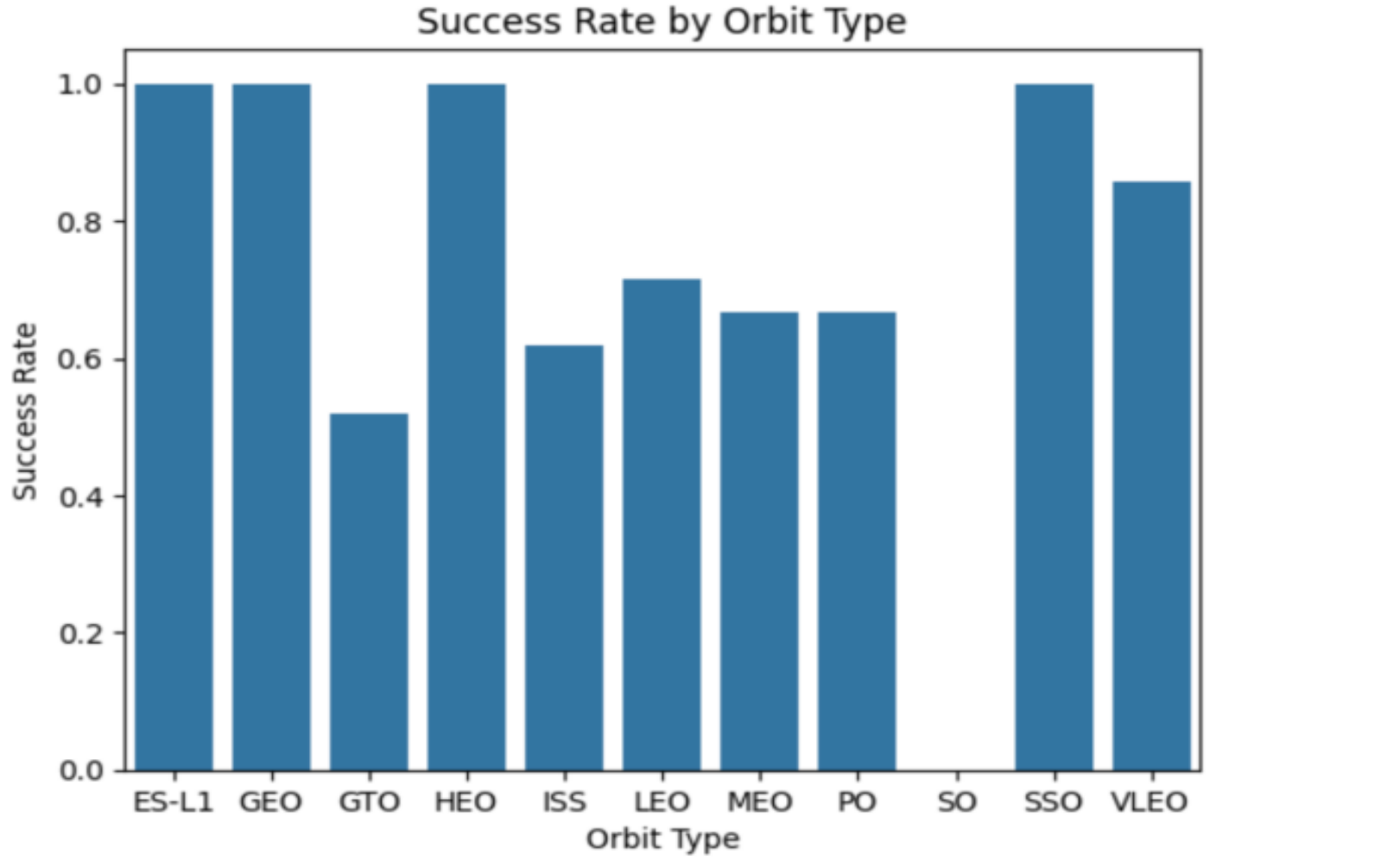
Flight Number vs. Launch Site



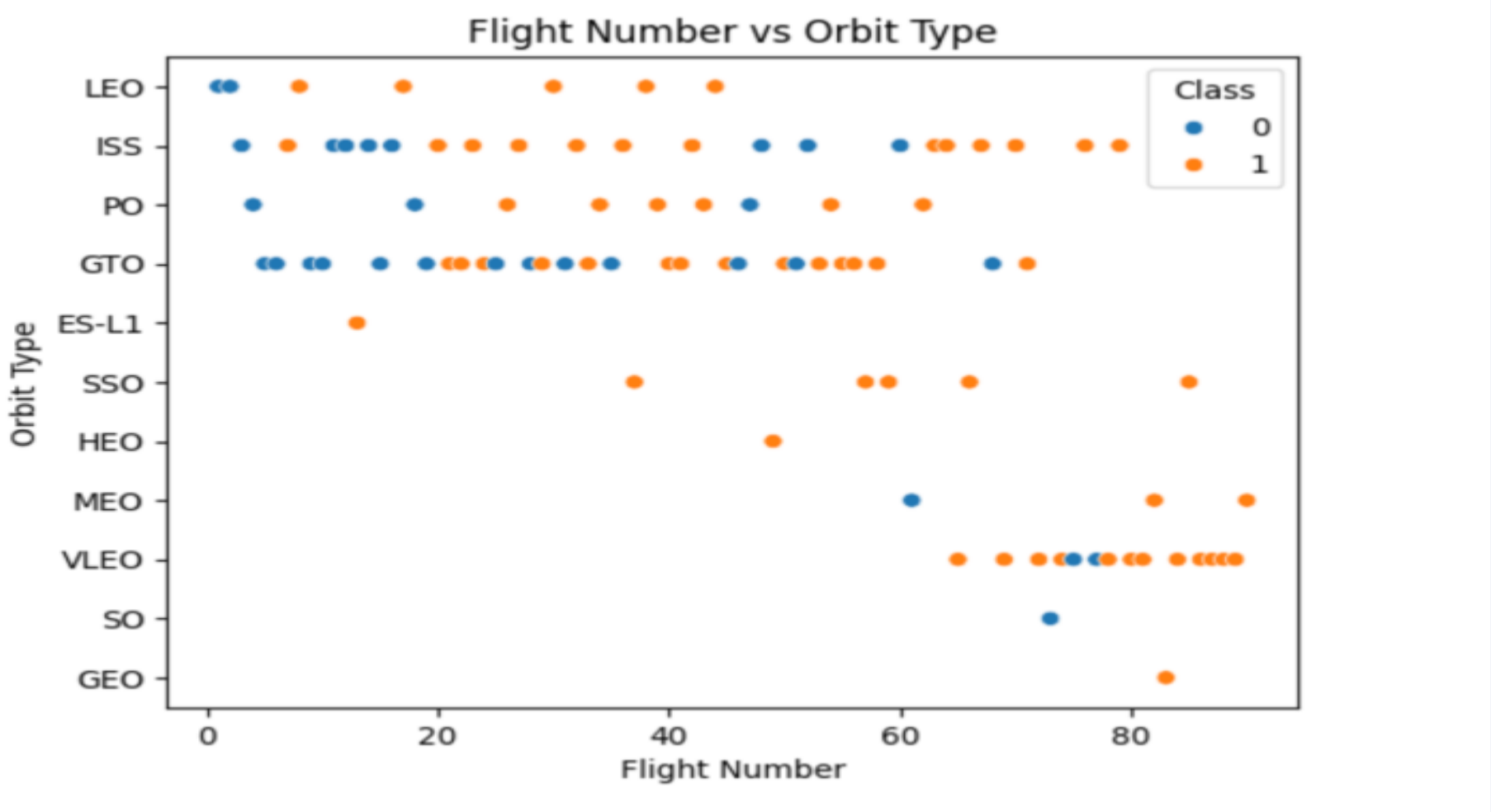
Payload vs. Launch Site



Success Rate vs. Orbit Type



Flight Number vs. Orbit Type



All Launch Site Names

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

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

```
 * 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;
```

```
 * 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)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTBL WHERE Customer LIKE '%NASA (CRS)%';
```

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

total_payload_mass
48213

Average Payload Mass by F9 v1.1

Task 4

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';
```

```
 * 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)';
```

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

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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 PAYLO
```

```
 * 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 Landing_Outcome, COUNT(*) AS total_count FROM SPACEXTBL GROUP BY Landing_Outcome;
```

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

Landing_Outcome	total_count
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

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);
```

```
 * 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.

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

```
 * 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.

```
%%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;
```

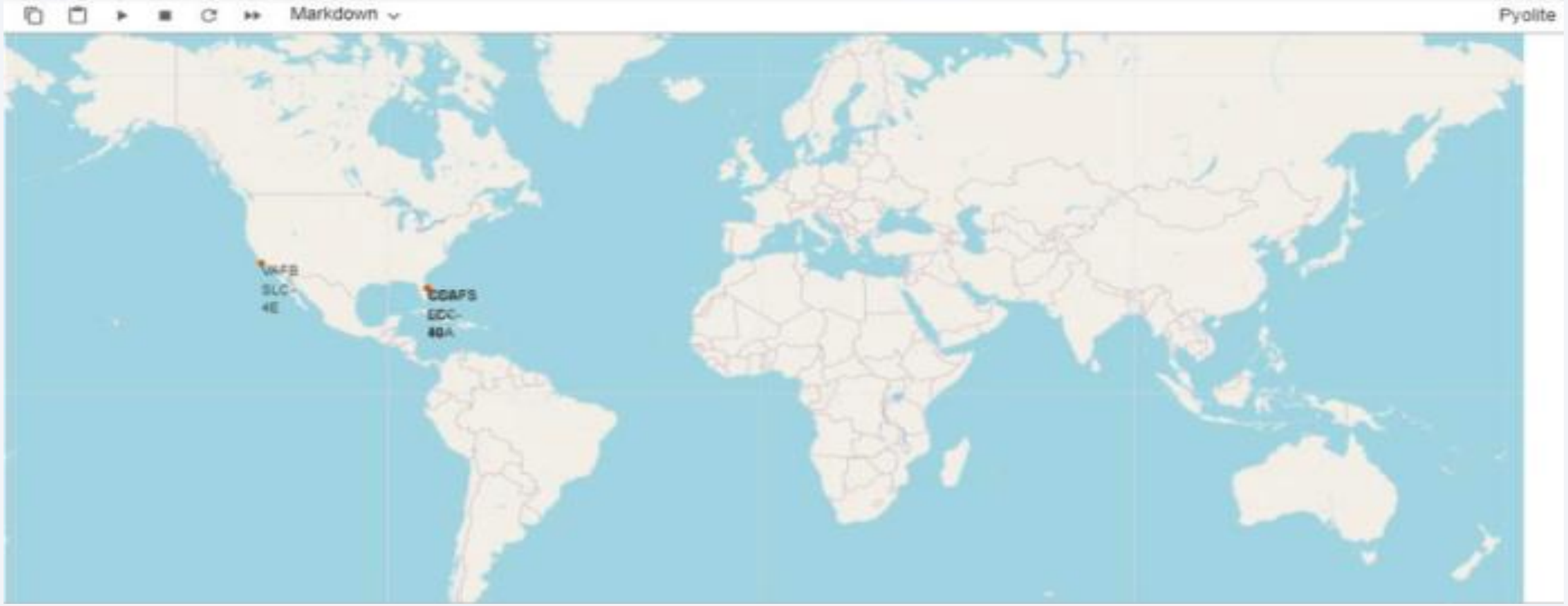
```
 * sqlite:///my_data1.db

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

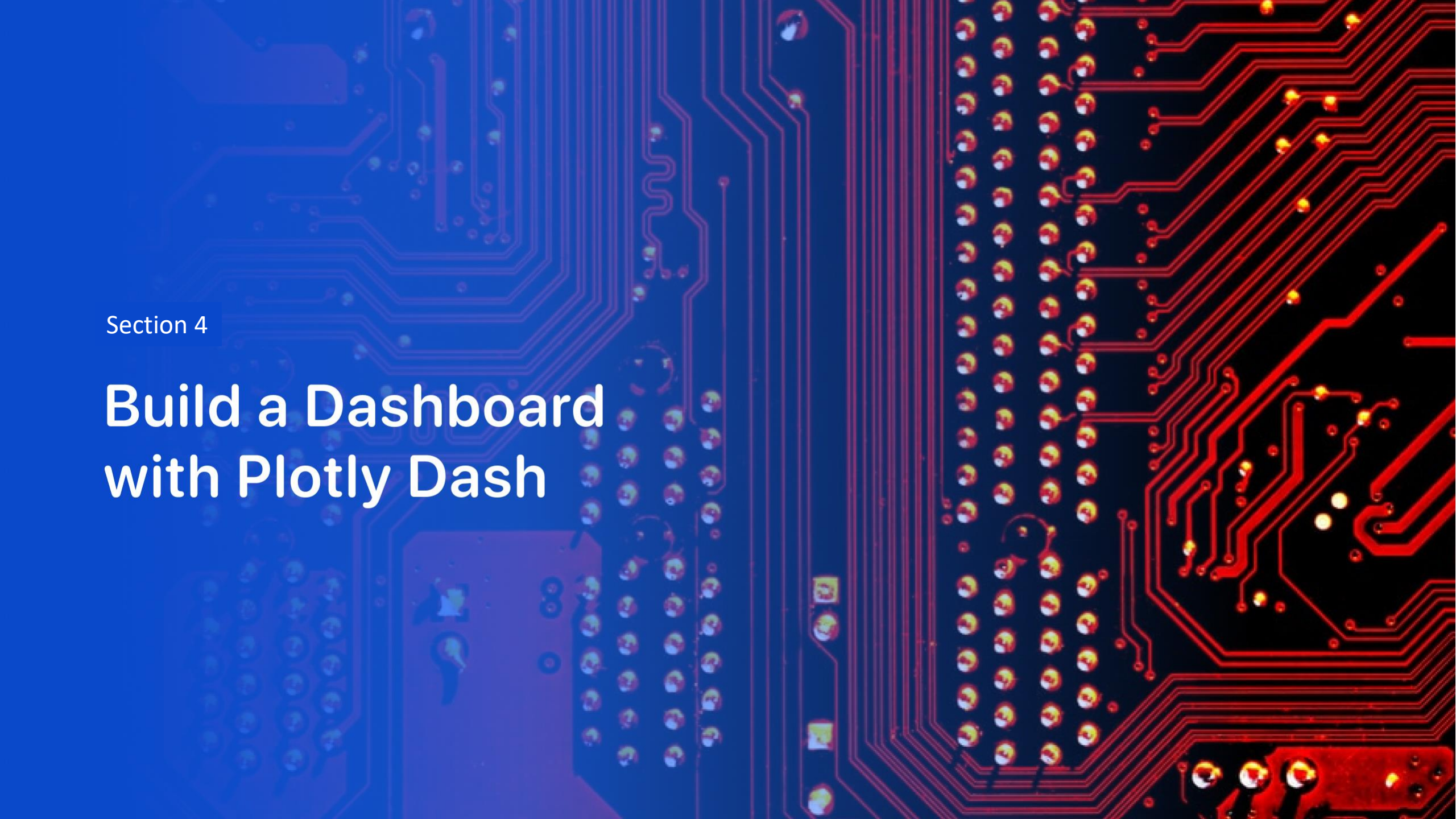
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

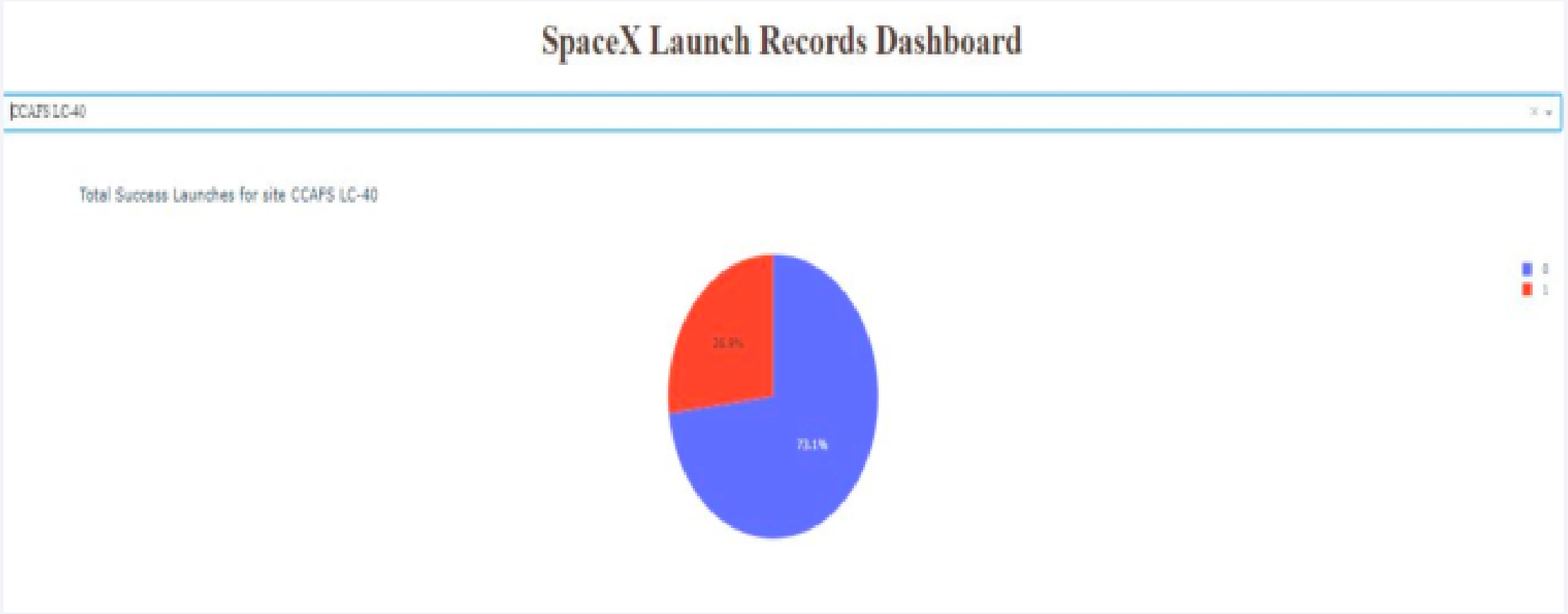


<Folium Map Screenshot 2>

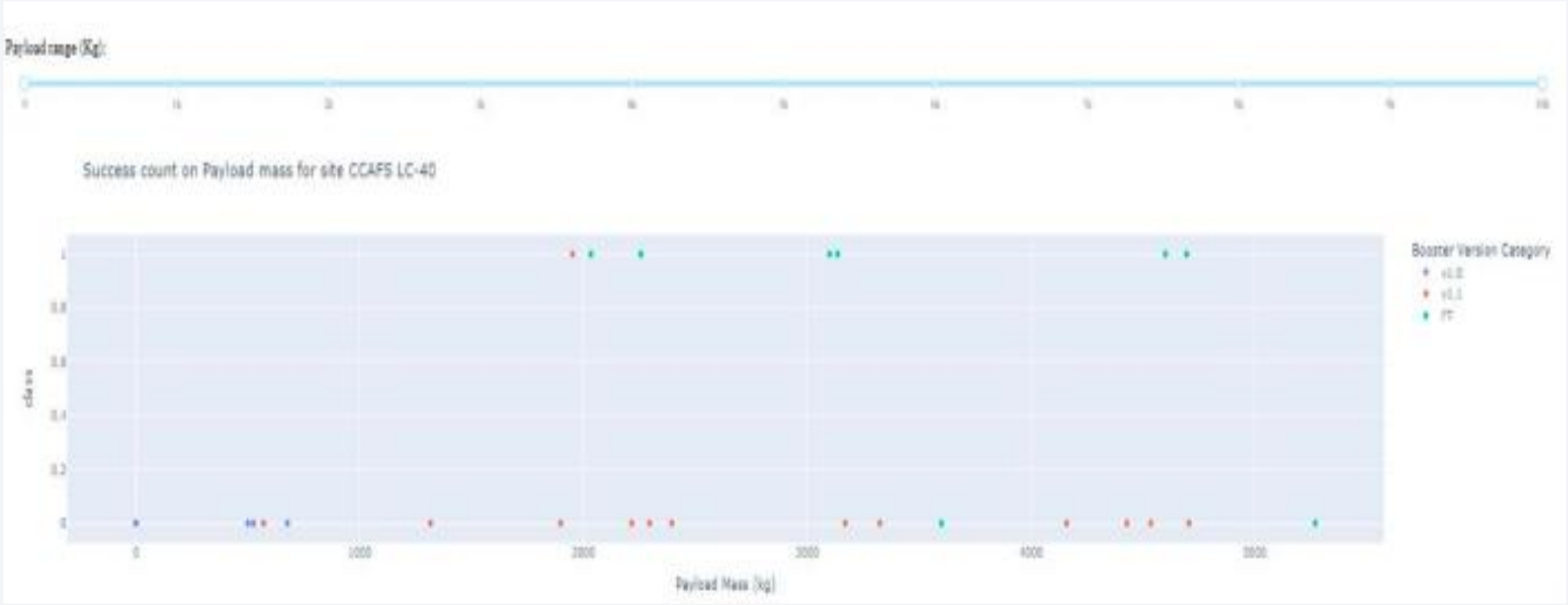
<Folium Map Screenshot 3>


Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



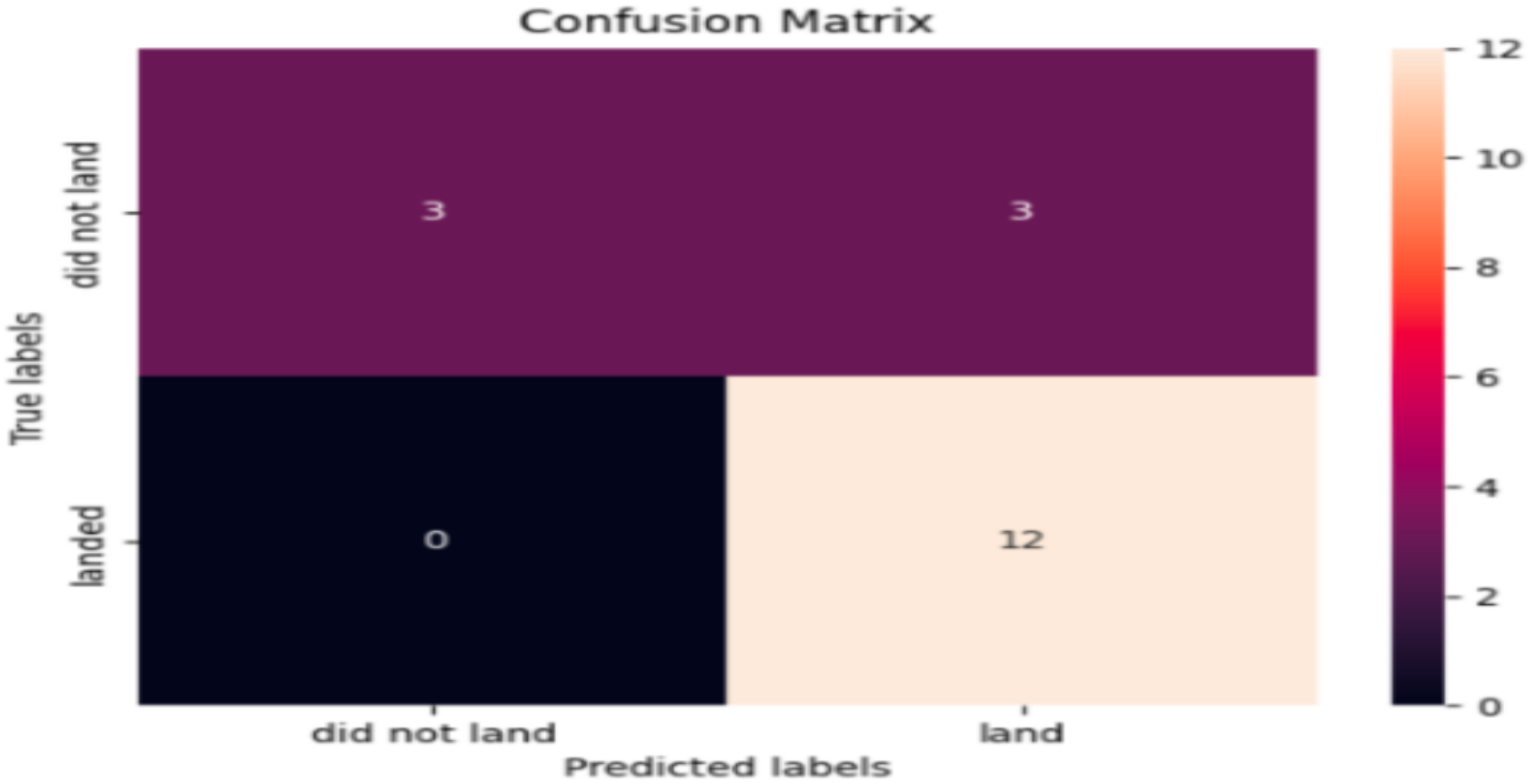
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix



Conclusions

- Different launch sites have varying success rates. CCAFS LC-40 has a success rate of 60%, while KSC LC-39A and VAFB SLC-4E have success rates of 77%.
- As the flight number increases at each of the three launch sites, so does the success rate. For example, the success rate at the VAFB SLC-4E site is 100% after the 50th flight. Both KSC LC-39A and CCAFS SLC-40 achieve 100% success rates after the 80th flight.
- Observing the Payload vs. Launch Site scatter plot reveals that no rockets with a payload mass greater than 10,000 kg have been launched from the VAFB SLC-4E site.
- Orbits ES-L1, GEO, HEO, and SSO have the highest success rates at 100%, whereas the SO orbit has the lowest success rate at approximately 50%, with SO orbit having a 0% success rate.
- For LEO orbit, the success rate appears to be related to the number of flights. However, there seems to be no relationship between flight number and success rate in the GTO orbit.

Thank you!