IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nelio Junior
05/07/2025

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

EXECUTIVE SUMMARY

Methodologies

Data collection with API

Data Wrangling

EDA using SQL and visualization techniques

Dashboard with Plotly Dash

Predictive Analysis (Classification)

Results

We examined the dataset and uncovered trends and associations among variables linked to landing outcomes. Leveraging these findings, we developed a predictive model using logistic regression that could reliably estimate the likelihood of a successful landing, achieving an accuracy rate of 83%.

INTRODUCTION

SpaceX's goal of reusable rockets has reduced space travel costs. Retrieving the first rocket phase is critical to reuse costly components. Analyzing the success rate provides insights into efficiency and cost-effectiveness. This project predicts the success of the first phase retrieval event to help improve space industry decision-making.

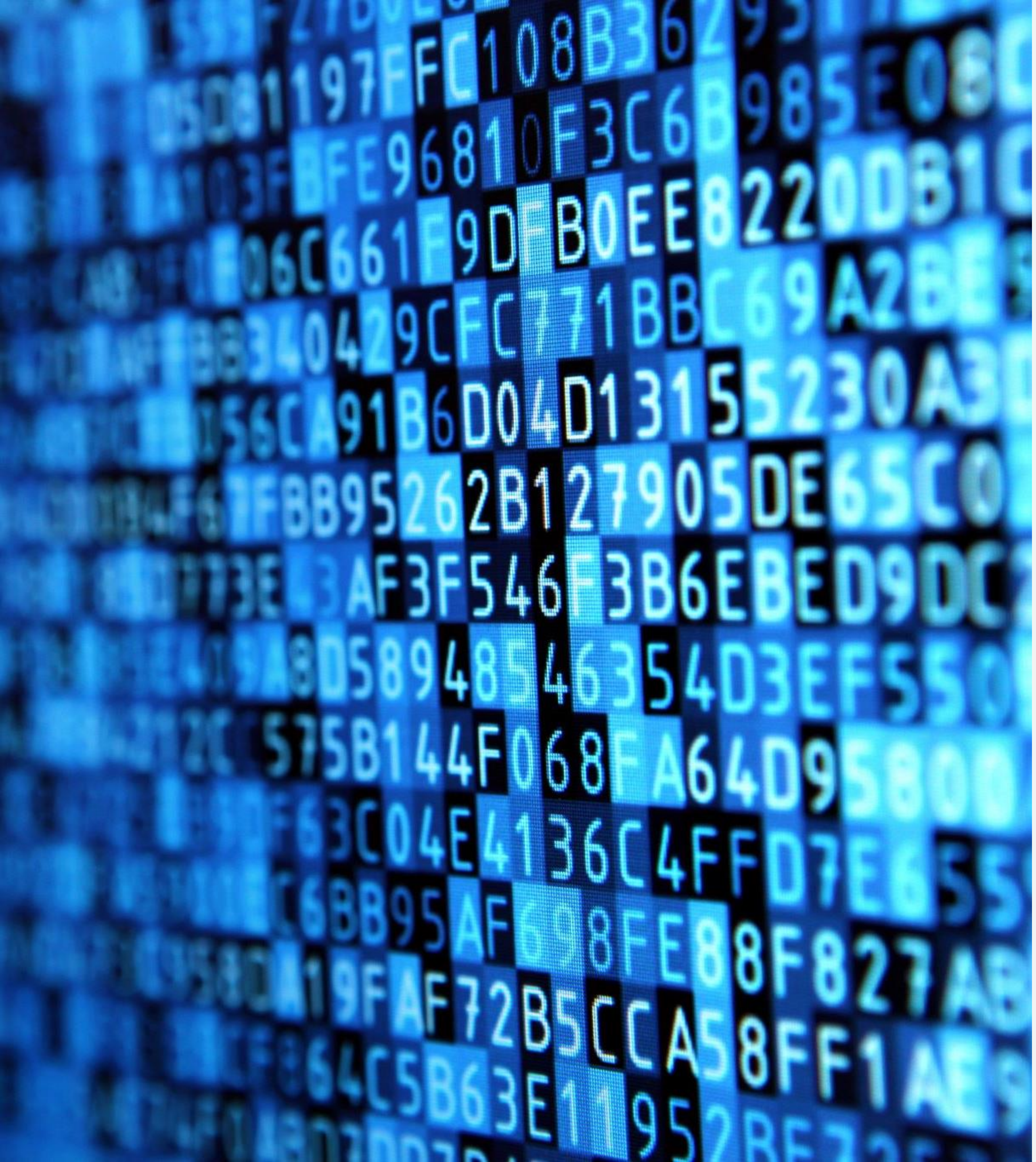
Our goal is to predict first phase rocket retrieval success to optimize resource allocation, enhancing mission success rates and cost savings.

Section 1

Methodology

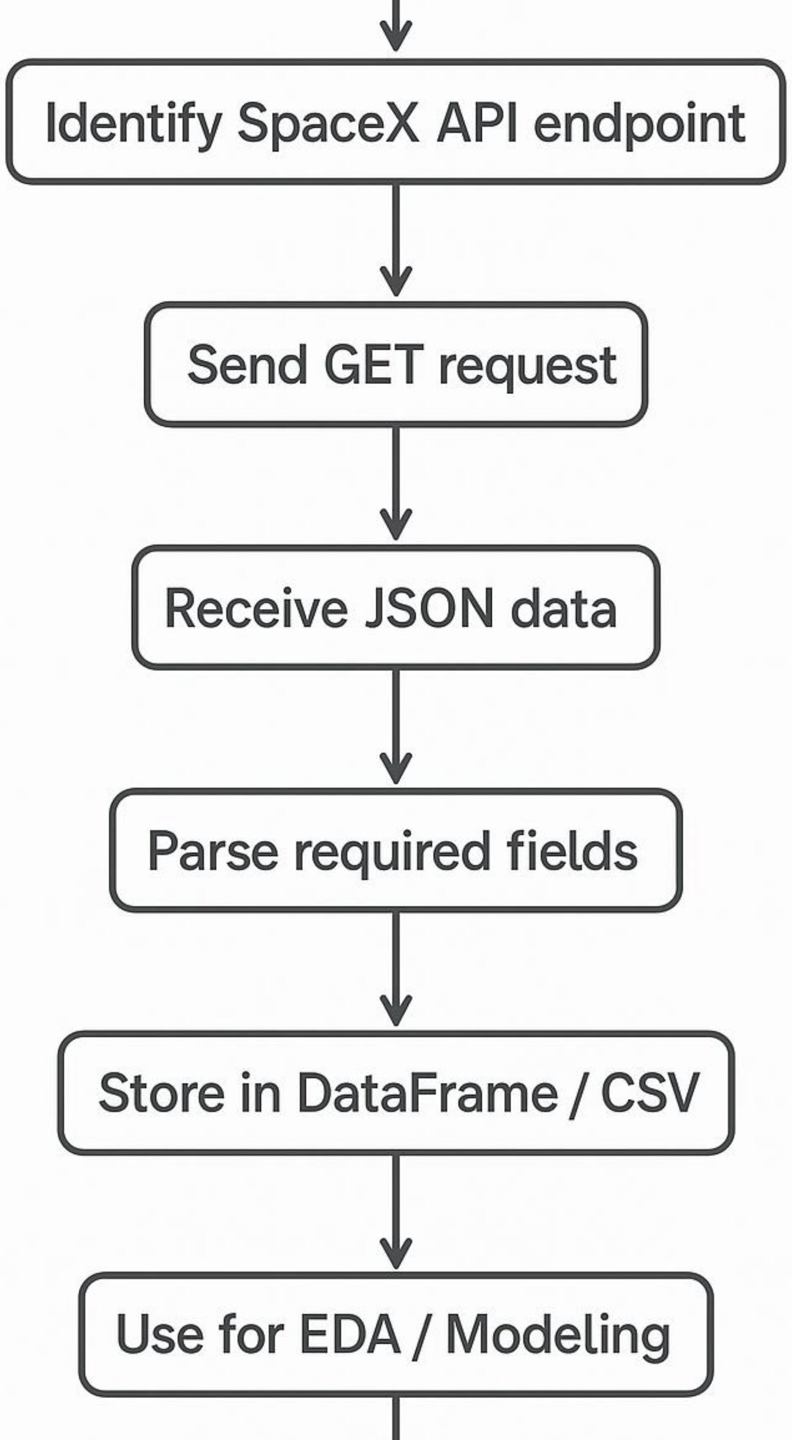
METHODOLOGY

- Executive Summary
- Data collection methodology
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models



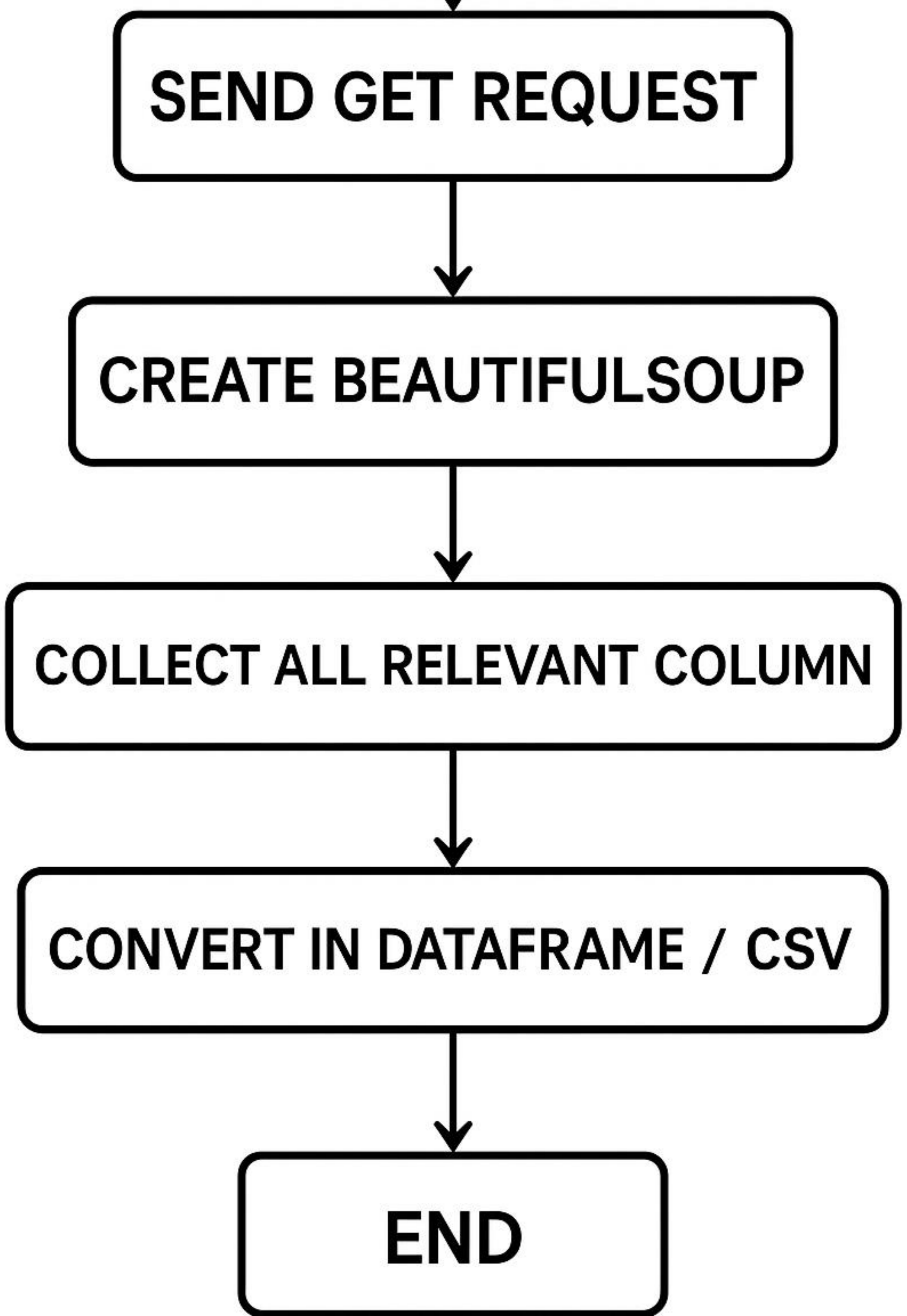
DATA COLLECTION

- Data was collected via SpaceX data API and scrapped from SpaceX related Wikipedia pages.



DATA COLLECTION – SPACEX API

- SpaceX REST API:
  - A public API to fetch data about launches, rockets, payloads, etc.
  - GET request: To retrieve data from the SpaceX API.
  - Endpoints: URLs like https://api.spacexdata.com/v4/launches to fetch data.
  - JSON: The data format returned by the API.
  - Filter/Parse: Selecting relevant fields (e.g., name, date_utc, rocket) from the JSON.
  - Store: Save the data in CSV, Pandas DataFrame, or database for analysis.
- https://github.com/NelioBJunior/public-projects/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



DATA COLLECTION - SCRAPING

- We used the requests.get method to download the page code.
- Created a BeaultifulSoup object to manipulate the html text.
- Collected all relevant column names from the HTML table header
- Converted the data from the HTML to pandas DataFrame format.

https://github.com/NelioBJunior/public-projects/blob/main/jupyter-labs-webscraping.ipynb

DATA WRANGLING

- Load the data and explore the first rolls to understand their structure.
- Identify and calculate the percentage of the missing values in each attribute
- Check the values and counts for the categorical and numerical columns.
- Perform data analysis such as:
  - Calculate the number of launches on each site;
  - Calculate the number and occurrence of each orbit;
  - Calculate the number and occurences of mission outcome of the orbits;
  - Create a landing outcome label from Outcome column.

- https://github.com/NelioBJunior/public-projects/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA WITH DATA VISUALIZATION

Scatterplot - LaunchSite vs FlightNumber

Scatterplot – LaunchSite vs PayloadMass

Barplot – Success rate vs Orbit

Scatterplot – Orbit vs FlightNumber

Scatterplot – Orbit vs PayloadMass

Linechart – Success rate vs Years

https://github.com/NelioBJunior/public-projects/blob/main/edadataviz.ipynb

EDA WITH SQL

Unique launch site names

CCA launch site records

Total NASA launched payload

Average F9 1.1 launched payload

First successful ground pad landing

Booster versions that carried the heaviest payload

https://github.com/NelioBJunior/public-projects/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

BUILD AN INTERACTIVE MAP WITH FOLIUM

Summary of Folium Map Objects and Their Purpose

Markers: Show exact locations on the map, highlighting important points like launch sites or landmarks.

Circles/CircleMarkers: Represent areas around points, such as impact zones or coverage ranges.

Polylines: Connect locations with lines to illustrate routes or relationships, like paths between launch sites and coastlines.

Popups/Tooltips: Provide extra information interactively without cluttering the map.

Custom Icons (DivIcon): Display styled labels or dynamic info (e.g., distances) to enhance clarity.

Why add these objects?

They improve map clarity, add interactivity, visualize spatial relationships, and help tell a geographic story, making data easier to understand and analyze.

https://github.com/NelioBJunior/public-projects/blob/main/lab_jupyter_launch_site_location.ipynb

PREDICTIVE ANALYSIS (CLASSIFICATION)

Model Development Summary

Prepare data: clean, select features, split train/test

Build models: train classifiers (Logistic Regression, Random Forest, SVM, XGBoost)

Evaluate: use accuracy, precision, recall, F1 with cross-validation

Improve: tune hyperparameters, handle imbalance, select features

Select best: choose model with highest validated performance, test on final data

Prepare data → Build models → Evaluate → Improve → Select best

https://github.com/NelioBJunior/public-projects/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

RESULTS

The CCAFS is the most frequent launching site.

The SO, GTO, ISS, PO, MEO and LEO are the most unsuccessful orbits to launch rockets and retrieve their first phase.

ES-L1, GEO, HEO and SSO orbits have always had successful retrievements.

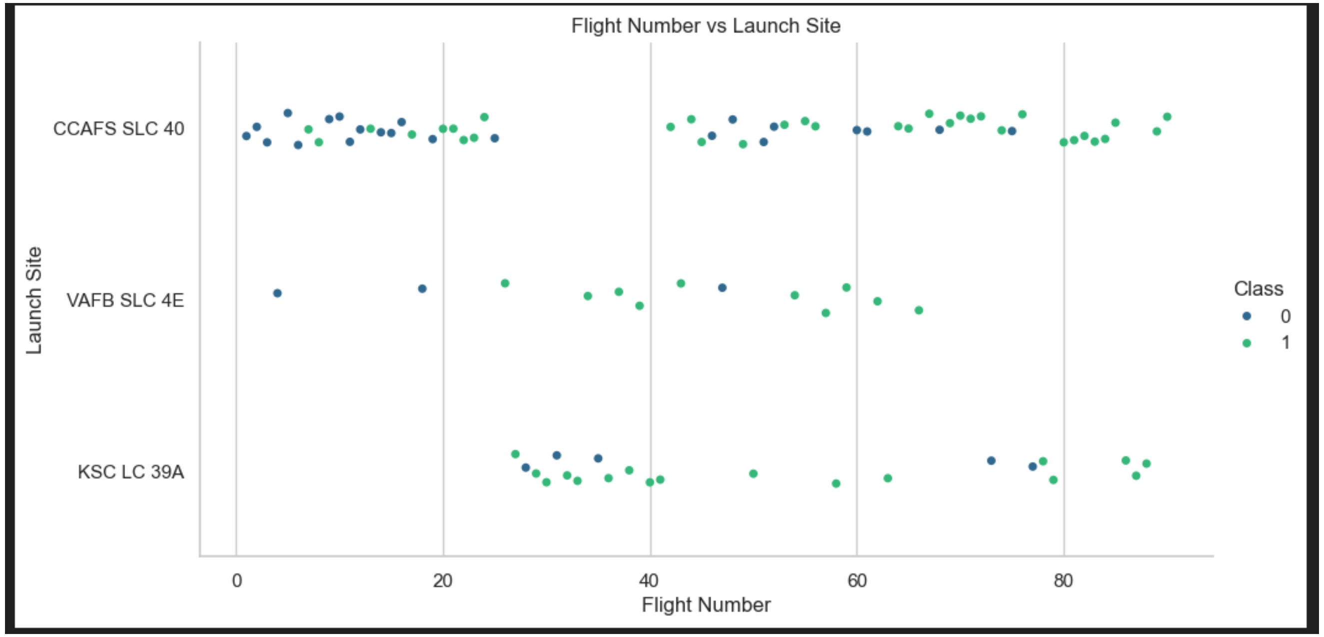
The success kept increasing since 2013.

All the different classification models (Logistic regression, KNN, SVM, Decision Tree) had a similar test set performance, the logistic regressionpresented a superior performance in the test set.

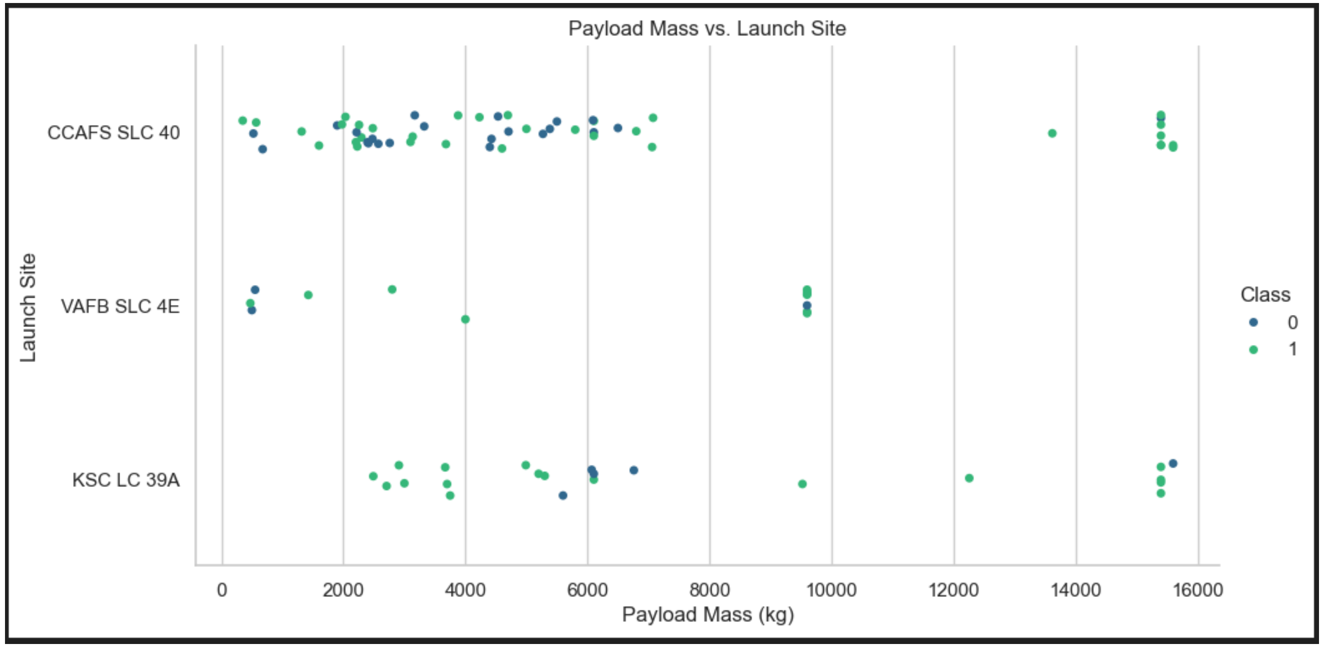
Section 2

Insights drawn from EDA

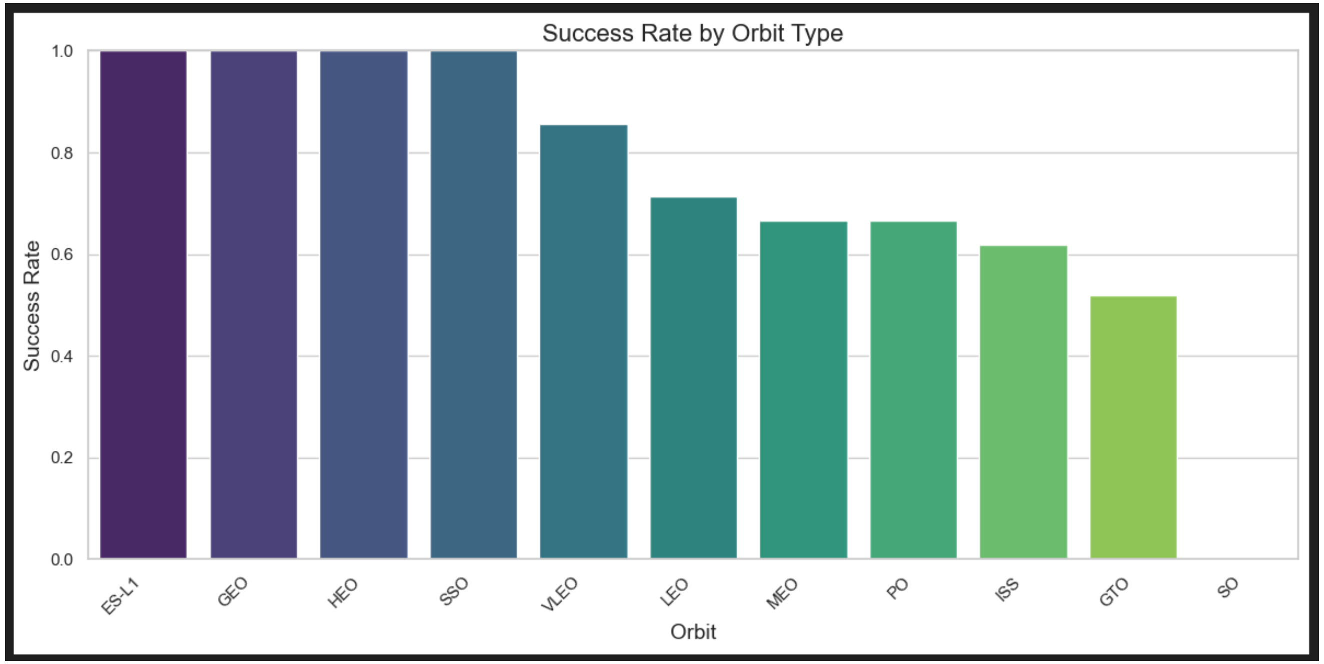
FLIGHT NUMBER VS. LAUNCH SITE



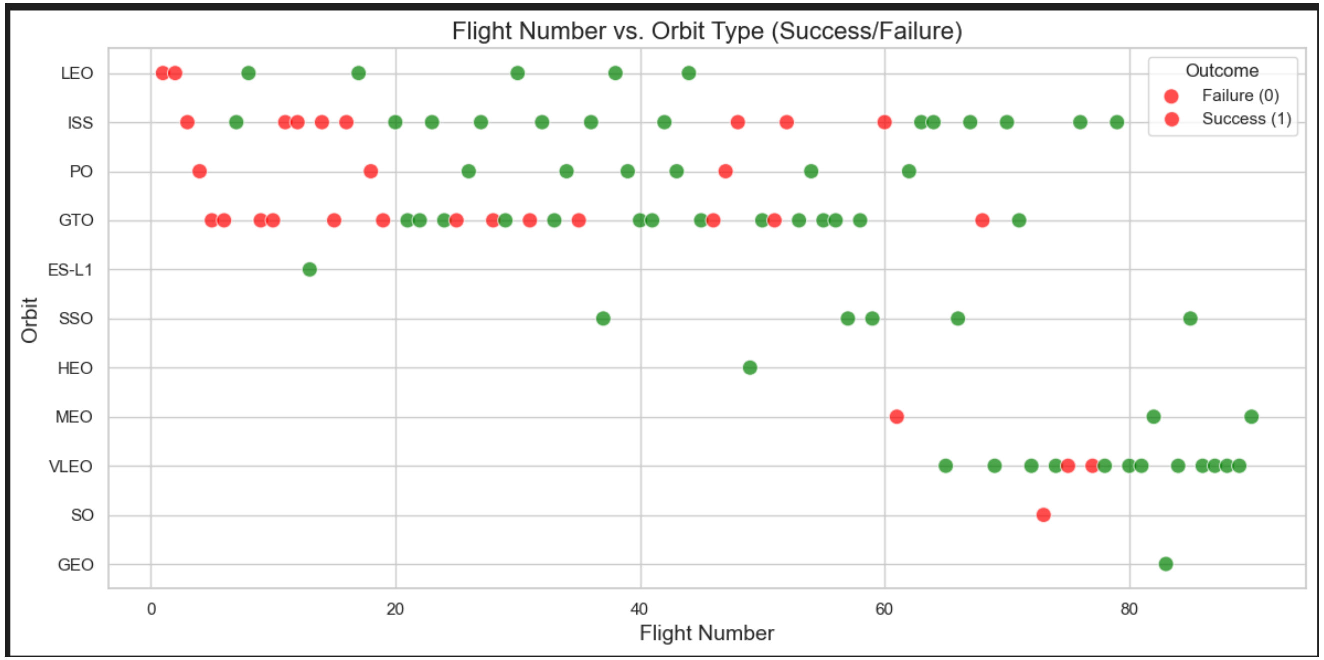
PAYLOAD VS. LAUNCH SITE



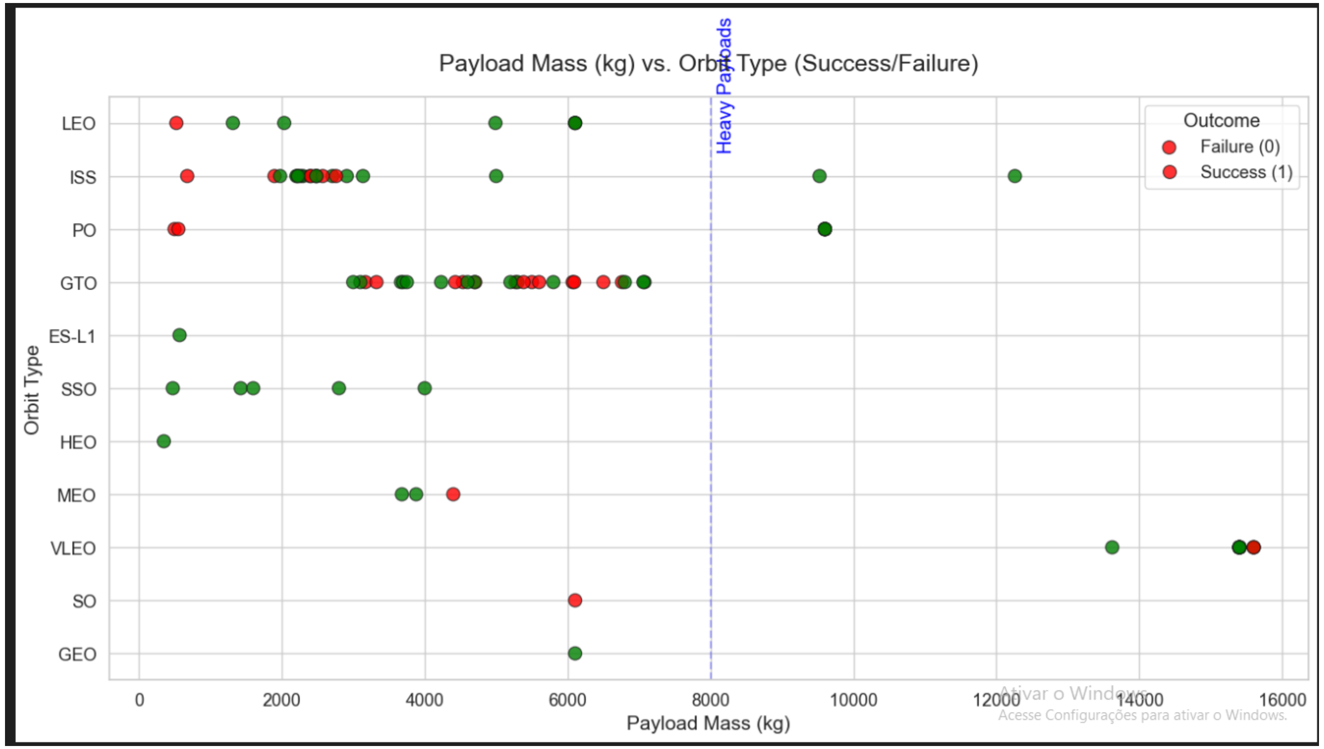
SUCCESS RATE VS. ORBIT TYPE



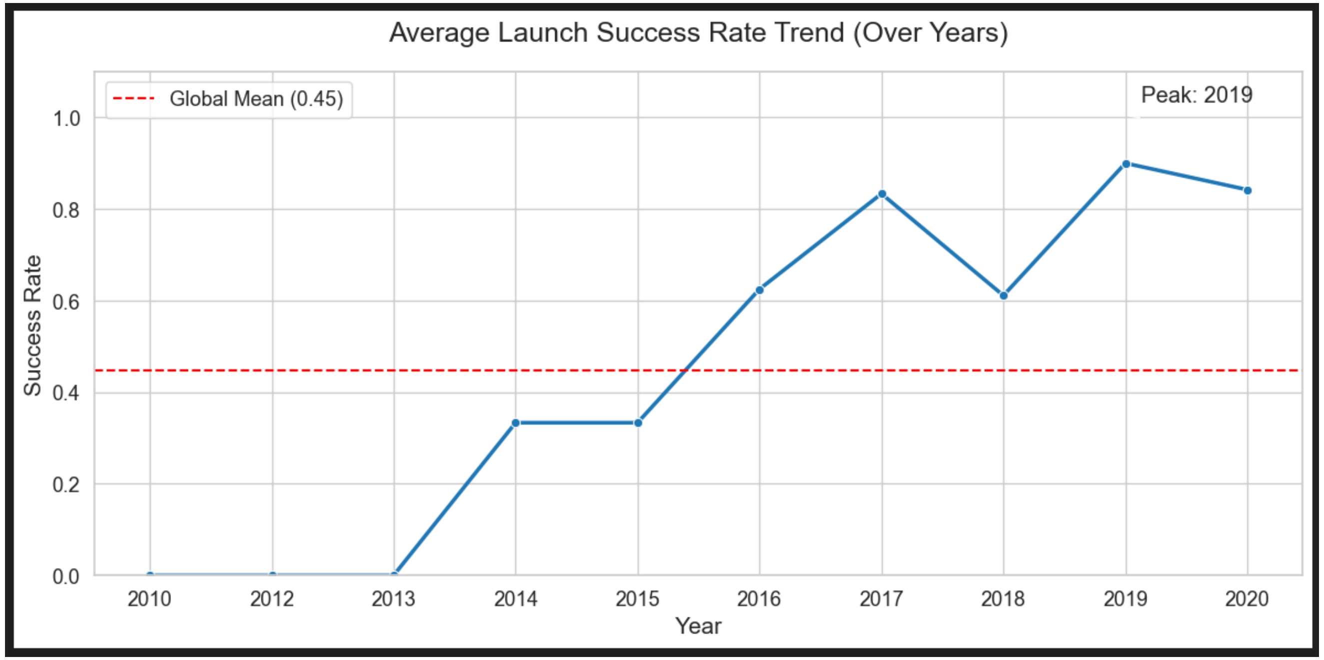
FLIGHT NUMBER VS. ORBIT TYPE



PAYLOAD VS. ORBIT TYPE



LAUNCH SUCCESS YEARLY TREND



ALL LAUNCH SITE NAMES

Select Distinct

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

LAUNCH SITE NAMES BEGIN WITH 'CCA'

- Use LIKE ‘CCA%’ and Limit to 5

	Launch_Site
0	CCAFS LC-40
1	CCAFS LC-40
2	CCAFS LC-40
3	CCAFS LC-40
4	CCAFS LC-40

TOTAL PAYLOAD MASS

48.213kg

```
# SQL query
query = """
SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass
FROM SPACEXTABLE
WHERE Customer LIKE '%NASA (CRS)%';
"""
```

AVERAGE PAYLOAD MASS BY F9 V1.1

2.928,4 kg

```
# SQL query
query = """
SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';
"""
```

FIRST SUCCESSFUL GROUND LANDING DATE

2015-12-22

```
# SQL query
query = """
SELECT MIN(Date) AS First_Successful_Landing_Date
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)';
"""
```

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

```
# SQL query
query = """
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000;
"""
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

```
# SQL query
query = """
SELECT Mission_Outcome, COUNT(*) AS Total_Count
FROM SPACEXTABLE
GROUP BY Mission_Outcome;
"""
```

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

BOOSTERS CARRIED MAXIMUM PAYLOAD

```
# SQL query
query = """
SELECT Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
);
"""
```

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

2015 LAUNCH RECORDS

```python
# SQL query
query = """
SELECT strftime('%m', Date) AS Month,
       Landing_Outcome,
       Booster_Version,
       Launch_Site
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE 'Failure (drone ship)%'
AND strftime('%Y', Date) = '2015';
"""
```

	Month	Landing_Outcome	Booster_Version	Launch_Site
0	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

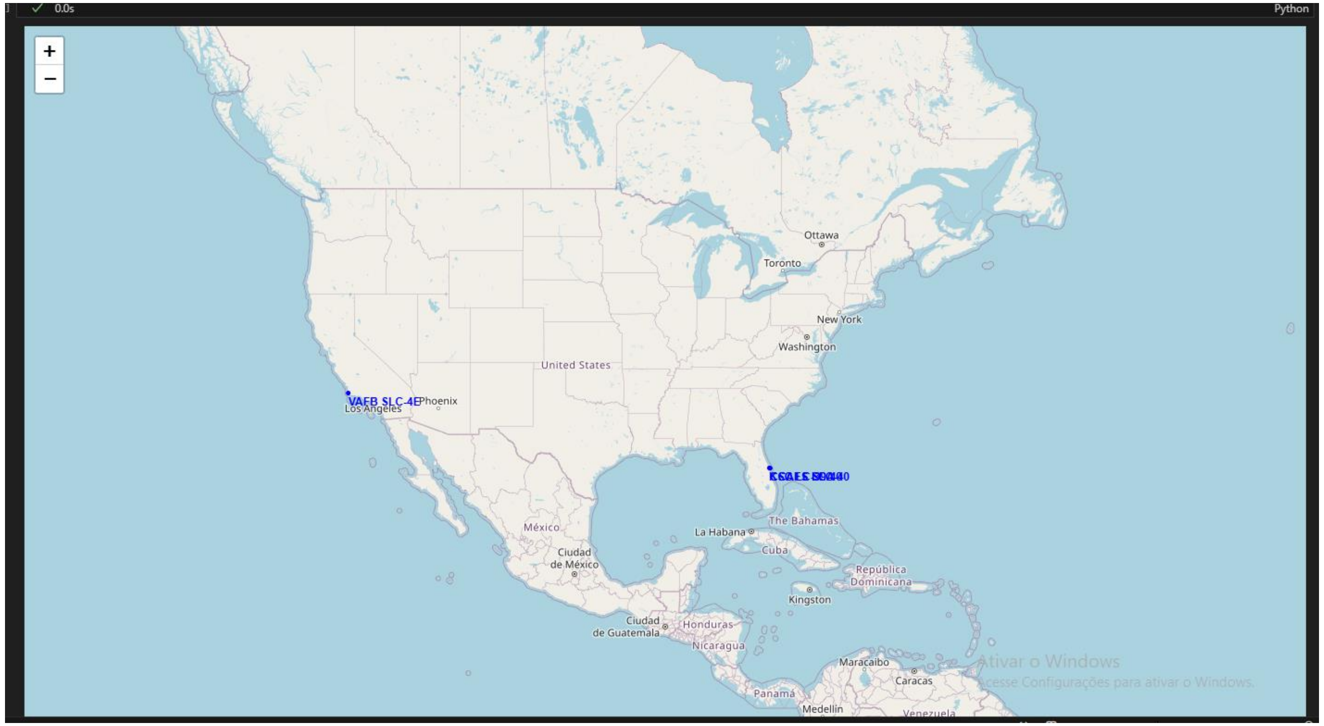
```
# SQL query
query = """
SELECT Landing_Outcome, COUNT(*) AS Outcome_Count,
          RANK() OVER (ORDER BY COUNT(*) DESC) AS Rank
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Rank;

"""
```

	Landing_Outcome	Outcome_Count	Rank
0	No attempt	10	1
1	Success (drone ship)	5	2
2	Failure (drone ship)	5	2
3	Success (ground pad)	3	4
4	Controlled (ocean)	3	4
5	Uncontrolled (ocean)	2	6
6	Failure (parachute)	2	6
7	Precluded (drone ship)	1	8

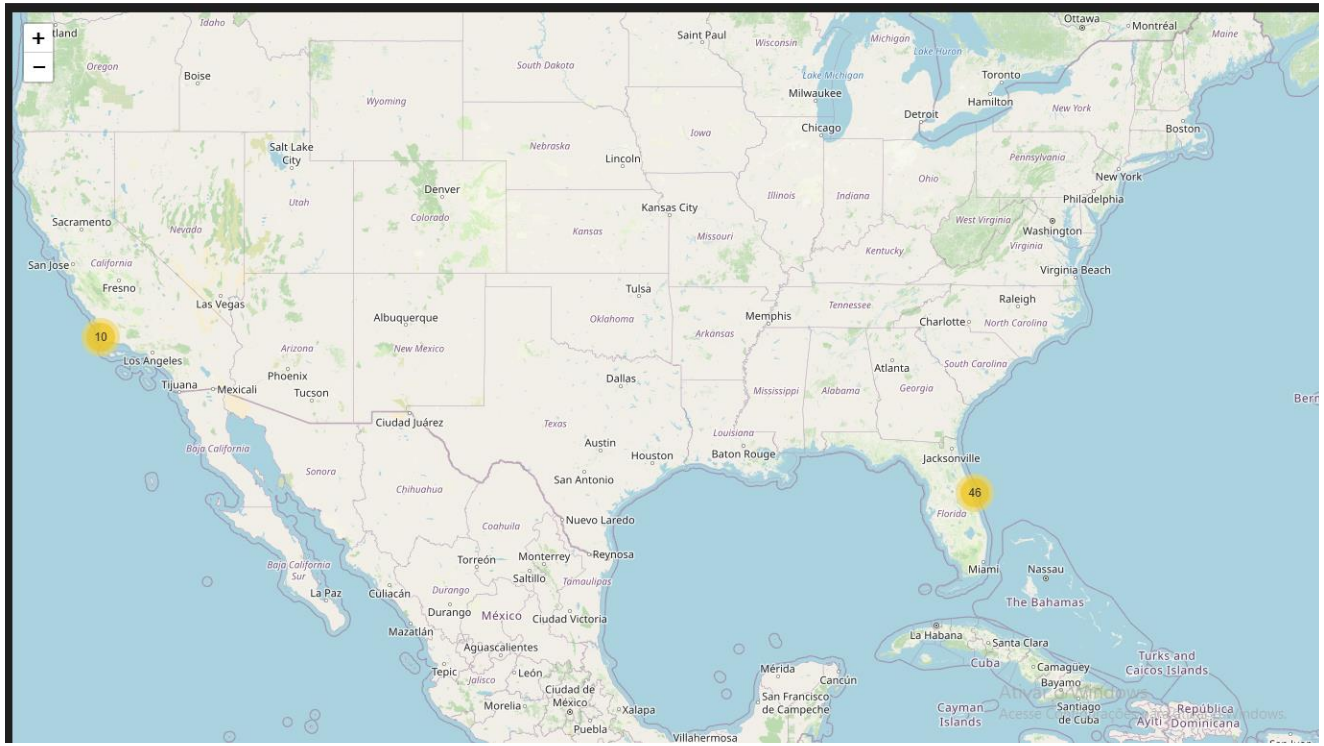
Section 3

Launch Sites Proximities Analysis

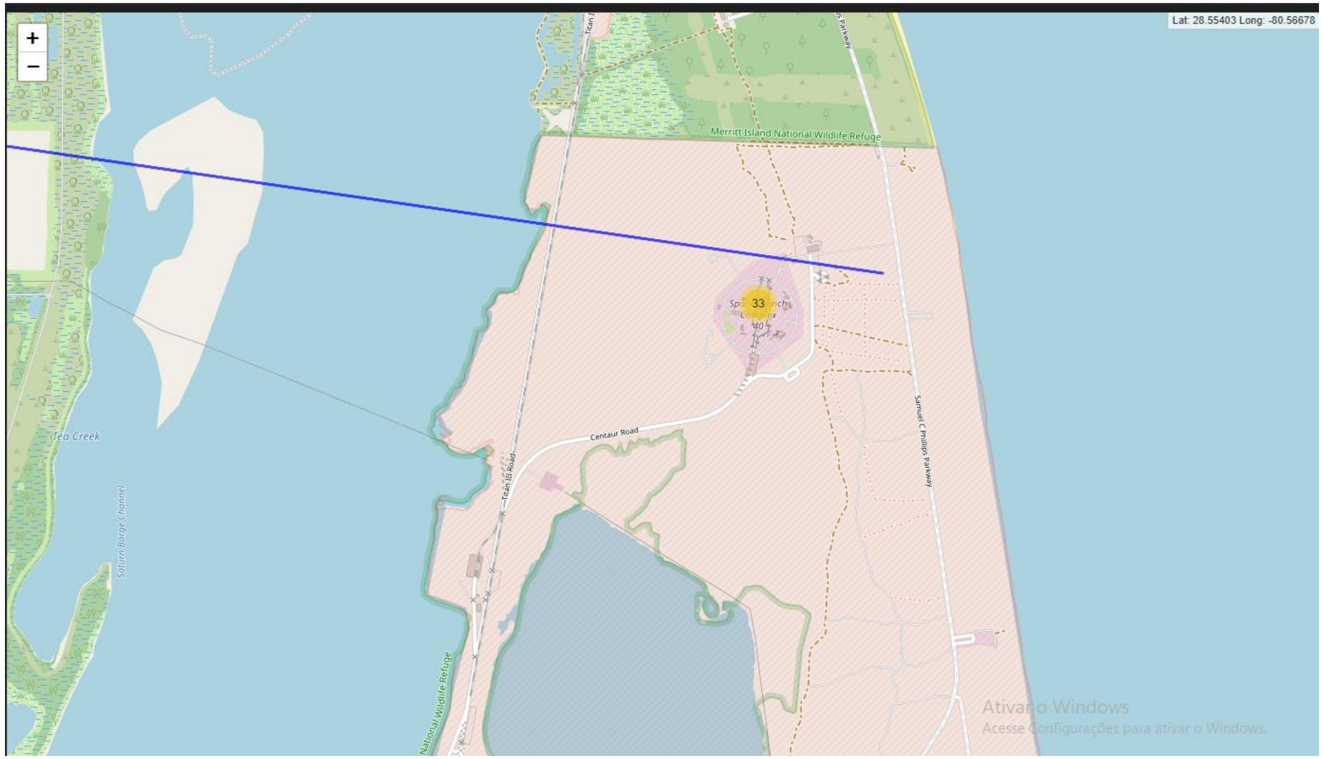
LAUNCH SITE LOCATIONS



LAUNCH LABELED



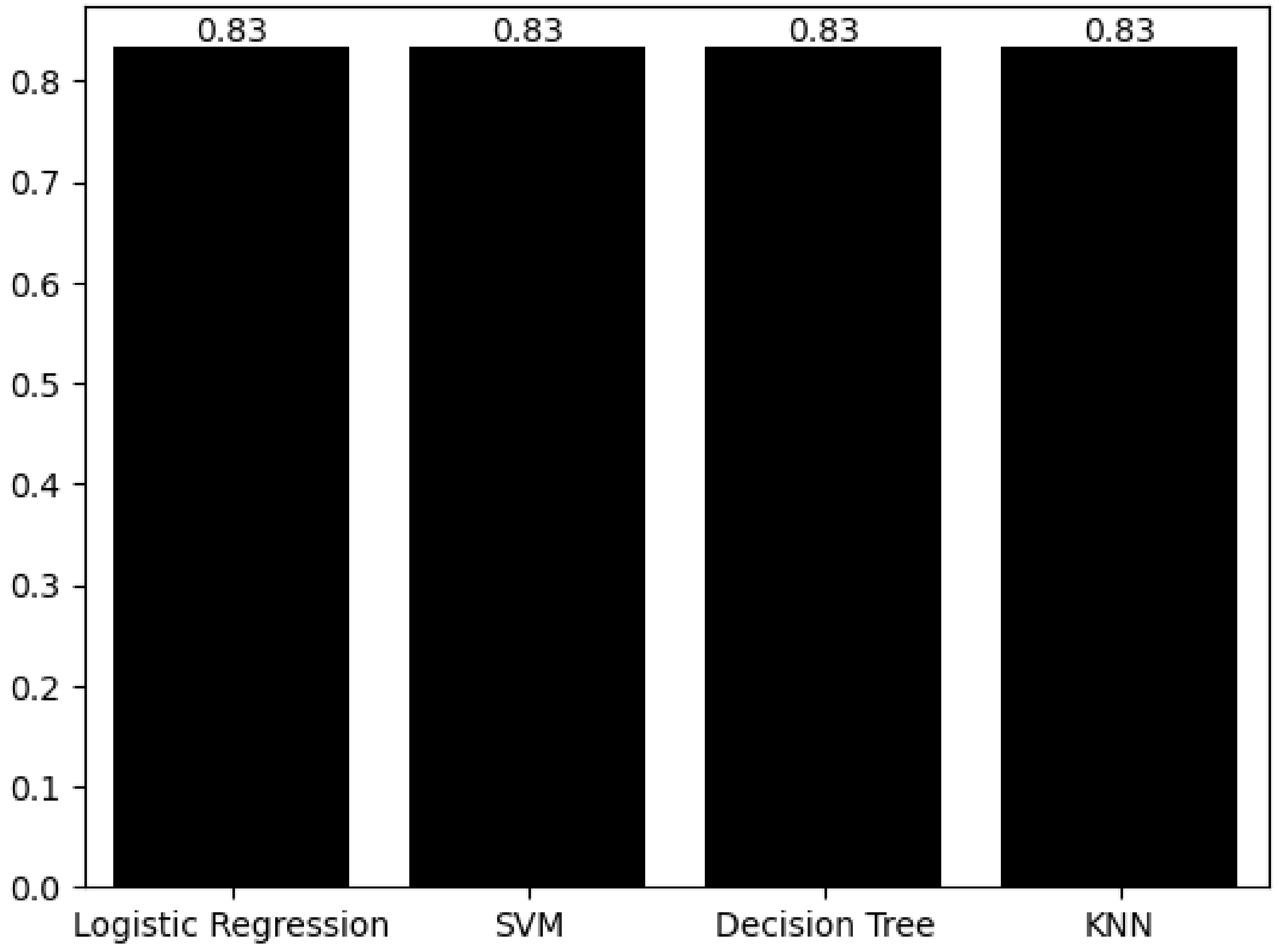
LAUNCH DISTANCE



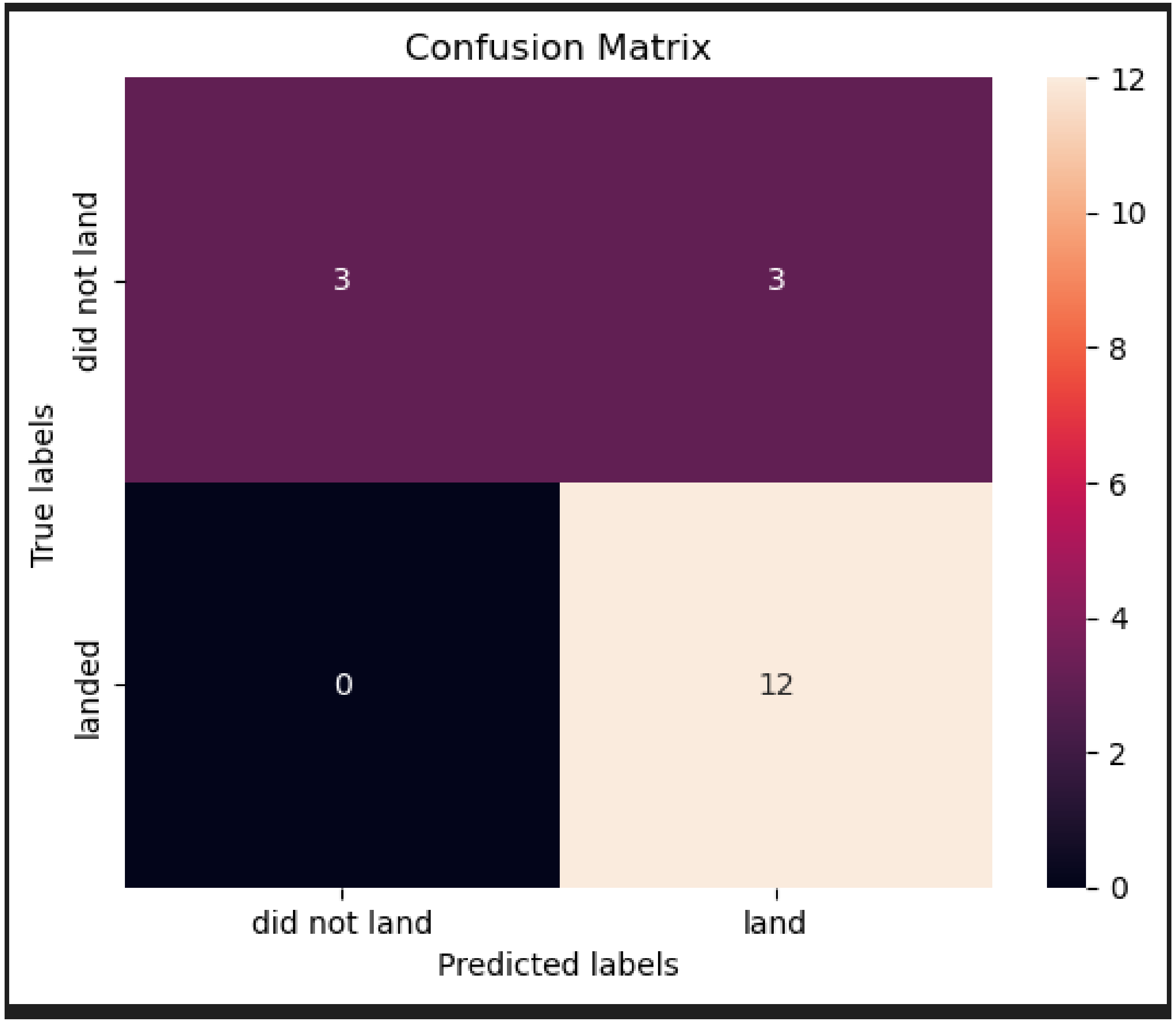
Section 5

Predictive Analysis (Classification)

CLASSIFICATION ACCURACY



CONFUSION MATRIX



CONCLUSIONS

- **Model Performance**:
  - The Machine Learning model achieves **83% accuracy** in predicting retrieval operations.
  - To evaluate its effectiveness, we compare this to the **baseline prediction rate** (i.e., the accuracy of always predicting the most frequent outcome).
- **Baseline Comparison**:
  - If the model's accuracy is **significantly higher** than the baseline, it demonstrates **true predictive value**.
  - If the improvement is **minimal**, the model may not offer much advantage over a simple rule-based approach.
- **Operational Impact**:
  - SpaceX's retrieval success rates have **consistently improved** over time.
  - This trend is expected to drive **further cost savings** and **enhance stakeholder confidence** in the business.

Thank you!