IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sina Bolouri

20th Feb 2025

https://github.com/cnabolouri/cnabolouri.github.io.git

Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

https://github.com/cnabolouri/cnabolouri.github.io.git

Executive Summary

Summary of Methodologies

- Data Collection & Processing

- Extracted SpaceX launch data from online sources.
- Cleaned missing values and standardized data formats.
- Applied one-hot encoding for categorical variables.

- Exploratory Data Analysis (EDA)

- Analyzed trends in launch sites, payload masses, and success rates.
- Visualized distributions using scatter plots, bar charts, and pie charts.
- Mapped launch sites and their proximities to coastlines and infrastructures.

- Feature Engineering

- Created dummy variables for categorical columns.
- Normalized numerical variables (e.g., Payload Mass).

https://github.com/cnabolouri/cnabolouri.github.io.git

Executive Summary

- Machine Learning Model Training

- Used **Logistic Regression, SVM, Decision Trees, and KNN** for classification.
- Applied **GridSearchCV** to optimize hyperparameters.
- Evaluated models using accuracy, confusion matrix, and best hyperparameters.

- Geospatial Analysis

- Mapped launch sites using **Folium**.
- Plotted distances between launch sites and key landmarks (cities, railways, highways).
- Analyzed launch site success rates using marker clusters.

https://github.com/cnabolouri/cnabolouri.github.io.git

Executive Summary

Summary of All Results

- Launch Success Rates

- Overall success rate: **~83%** across all launch sites.
- Highest success rate observed at **KSC LC-39A** launch site.

- Payload vs. Success

- Higher payload masses had a higher success rate.
- Geosynchronous (GTO) orbits showed **lower** success rates than Low Earth Orbits (LEO).

https://github.com/cnabolouri/cnabolouri.github.io.git

Executive Summary

- Best Machine Learning Model

- **Support Vector Machines (SVM) with RBF Kernel** provided the highest validation accuracy (~**83%**).
- Logistic Regression, Decision Tree, and KNN had similar accuracy.

- Launch Site Proximities

- All launch sites are near coastlines for safety reasons.
- Most launch sites are **far from major cities** but **close to infrastructure like railways and highways**.

Introduction

Project Background and Context

- Space exploration has significantly advanced in recent years.
- Private companies, particularly SpaceX, are at the forefront of innovations in reusable rocket technology.

The Falcon 9 rocket, developed by SpaceX, has:

- - Drastically lowered the cost of space travel.
- - Allowed for the reuse of first-stage boosters.

A thorough understanding of the factors that influence successful rocket landings is essential for:

- - Enhancing efficiency.
- - Lowering costs.
- - Ensuring the reliability of missions.

Introduction

This project aims to:

- - Analyze historical SpaceX launch data.
- - Identify the key factors affecting mission success.

The methodologies employed include:

- - Data analytics.
- - Machine learning.
- - Geospatial visualization.

The analysis will focus on:

- - Launch patterns.
- - The impact of payloads.
- - The influence of infrastructure on launch success.

https://github.com/cnabolouri/cnabolouri.github.io.git

Introduction

Problems We Want to Find Answers To

- What factors influence the success or failure of Falcon 9 launches?

- Examining the role of **payload mass, orbit type, and booster version** on landing success.
- Analyzing how **launch sites** impact mission outcomes.

- Which launch sites have the highest success rates?

- Identifying the best-performing launch sites for booster landings.
- Evaluating the proximity of launch sites to coastlines, cities, and infrastructure.

https://github.com/cnabolouri/cnabolouri.github.io.git

Introduction

- Can we predict launch success using machine learning?

- Training classification models (**Logistic Regression, SVM, Decision Trees, KNN**) to predict successful launches.
- Comparing model performance and selecting the most accurate predictor.

- What insights can geospatial analysis provide?

- Mapping launch site locations and their distances to nearby cities, highways, and coastlines.
- Exploring patterns in launch site selection and infrastructure accessibility.

Section 1

Methodology

https://github.com/cnabolouri/cnabolouri.github.io.git

https://github.com/cnabolouri/cnabolouri.github.io.git

Methodology

Methodology

This project follows a structured data-driven approach to analyze SpaceX launch success factors. The methodology consists of the following key steps:

1. Data Collection Methodology

Data was sourced from publicly available SpaceX launch records, including **historical mission data, payload specifications, and launch outcomes**.

Web scraping techniques were employed to extract relevant data from Wikipedia and SpaceX API.

The dataset was stored in CSV format and loaded into **Pandas** for further analysis.

2. Data Wrangling

The dataset underwent preprocessing to handle missing values and ensure consistency.

Data was **filtered and cleaned** to retain only relevant Falcon 9 launches.

Columns with missing values, such as **Payload Mass and Landing Pad**, were either imputed with mean values or retained as missing indicators.

https://github.com/cnabolouri/cnabolouri.github.io.git

3. Exploratory Data Analysis (EDA)

SQL queries were used to **extract insights on launch frequencies, payload distributions, and landing success rates**.

Visualizations using Matplotlib and Seaborn helped identify patterns in launch outcomes based on payload, launch sites, and orbit types.

4. Interactive Visual Analytics

Folium was used to create an **interactive map** of SpaceX launch sites, highlighting their proximity to coastlines, highways, and urban areas.

Plotly Dash was implemented to build a dynamic dashboard for real-time exploration of mission success rates and launch trends.

5. Predictive Analysis Using Classification Models

Machine learning models (**Logistic Regression, SVM, Decision Tree, and KNN**) were trained to predict **launch success**.

GridSearchCV was used for hyperparameter tuning to optimize model performance.

Model evaluation was conducted using accuracy scores, confusion matrices, and validation datasets.

Data Collection

The data collection process for this project involved extracting, filtering, and structuring SpaceX launch data from various sources. The key steps are outlined below:

1. Data Sources

- **Wikipedia Web Scraping**: Used BeautifulSoup to extract launch records such as **flight numbers, dates, payload mass, orbit types, and mission outcomes**.
- **SpaceX API**: Retrieved **real-time launch data**, including landing outcomes and booster versions.
- **CSV Files**: Preprocessed datasets from **IBM Skills Network** were used to ensure structured and clean data for analysis.

https://github.com/cnabolouri/cnabolouri.github.io.git

Data Collection

2. Data Extraction & Processing

- Extracted relevant **columns from web-scraped tables** and **formatted raw JSON data** from the API.
- Used **Pandas** to clean missing values and standardize data types.
- Removed Falcon 1 launches to focus only on **Falcon 9 missions**.

3. Data Storage & Retrieval

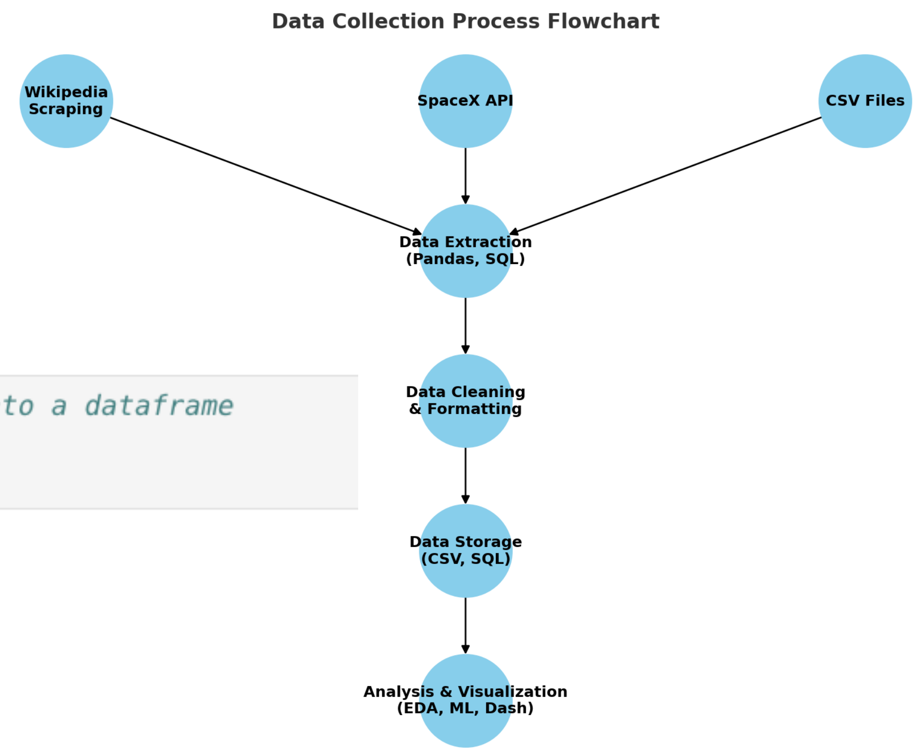
- Data was stored in **CSV files** for structured processing.
- SQL database (SQLite) was used for **query-based data retrieval**.
- Interactive visual analytics tools such as **Folium and Plotly Dash** used this structured data for real-time insights.

https://github.com/cnabolouri/cnabolouri.github.io.git

Data Collection – SpaceX API

```
response=requests.get(static_json_url)
```

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



https://github.com/cnabolouri/cnabolouri.github.io.git

Data Collection - Scraping

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
# Create a BeautifulSoup object from the HTML response
soup = BeautifulSoup(response.text, 'html.parser')
```

```python
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
# Extract all tables in the Wikipedia page
html_tables = soup.find_all("table")
```


https://github.com/cnabolouri/cnabolouri.github.io.git

Data Wrangling

- Data analysis and exploration.
- Establishing training labels.

```
# Apply value_counts() on column LaunchSite
# Count the number of launches for each launch site
launch_counts = df['LaunchSite'].value_counts()
```

```
# Apply value_counts on Orbit column
# Count the number of launches for each orbit type
orbit_counts = df['Orbit'].value_counts()
```

```
# landing_outcomes = values on Outcome column
# Count the number of each unique mission outcome
landing_outcomes = df['Outcome'].value_counts()
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
# Assign 0 if Outcome is in bad_outcomes, else 1
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
```


https://github.com/cnabolouri/cnabolouri.github.io.git

EDA with Data Visualization

Bar Chart: Launch Success Rate by Site

- This chart was used to compare the number of successful and failed launches across different launch sites.
- It helped identify which launch sites had the highest success rates.

Pie Chart: Proportion of Successful Launches

- Displayed the overall success rate of SpaceX launches, showing the proportion of successful vs. failed launches.
- Provided a quick understanding of SpaceX's performance in landing rocket boosters.

Scatter Plot: Payload Mass vs. Success Rate

- Plotted payload mass against mission success to examine whether heavier payloads affected launch outcomes.
- Different colors were used to distinguish successful and unsuccessful launches.

Scatter Plot: Flight Number vs. Launch Site

- Used to analyze how experience (number of launches) correlated with success rates at different launch sites.
- Helped determine if launch sites improved their success rate over time.

Bar Chart: Success Rate by Orbit Type

- Showed the number of successful vs. failed missions for different orbit types (LEO, GTO, ISS, etc.).
- Helped assess whether certain orbits were more challenging than others.

Geographical Map with Folium

- Plotted the launch sites on a world map to visualize their locations.
- Used different markers to indicate success and failure at each site.

Line Chart: Yearly Launch Trends

- Illustrated the trend of launch attempts and successes over time.
- Helped analyze how SpaceX's performance evolved over the years.

https://github.com/cnabolouri/cnabolouri.github.io.git

EDA with SQL

Retrieve a Sample of Data

- SELECT * FROM SPACEXTBL LIMIT 20;
- Fetched the first 20 rows from the dataset to get an overview of the data structure.

Identify Unique Launch Sites

- SELECT DISTINCT Launch_Site FROM SPACEXTBL;
- Retrieved unique launch site names to understand the different locations used by SpaceX.

Count the Number of Launches per Site

- SELECT Launch_Site, COUNT(*) AS Launch_Count FROM SPACEXTBL GROUP BY Launch_Site;
- Determined how frequently each launch site was used.

Analyze Launch Outcomes

- SELECT Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTBL GROUP BY Outcome;
- Summarized the number of successful and failed launches.

Find the Earliest Successful Ground Landing

- SELECT MIN(Date) FROM SPACEXTBL WHERE Outcome LIKE '%Success%' AND LandingPad IS NOT NULL;
- Identified the first successful landing of a rocket on a ground pad.

Payload Mass Analysis by Booster Version

- SELECT BoosterVersion, AVG(PayloadMass) AS Avg_Payload FROM SPACEXTBL GROUP BY BoosterVersion;
- Calculated the average payload mass carried by each booster version.

Filter Launches for a Specific Year

- SELECT * FROM SPACEXTBL WHERE SUBSTR(Date, 1, 4) = '2015';
- Extracted all launch records from the year 2015.

Identify the Most Common Orbit Type

- SELECT Orbit, COUNT(*) AS Orbit_Count FROM SPACEXTBL GROUP BY Orbit ORDER BY Orbit_Count DESC LIMIT 1;
- Determined which orbit type was most frequently used for launches.

Determine the Maximum Payload Mass for Each Launch Site

- SELECT Launch_Site, MAX(PayloadMass) AS Max_Payload FROM SPACEXTBL GROUP BY Launch_Site;
- Found the heaviest payload launched from each site.

https://github.com/cnabolouri/cnabolouri.github.io.git

Build an Interactive Map with Folium

Map Objects Created and Added

- **Markers**
  - Placed markers at each SpaceX launch site using folium.Marker().
  - Used **green markers** for successful launches (class=1) and **red markers** for failed launches (class=0).
  - **Purpose**: Allows easy identification of launch sites and their success rates.
- **Circles**
  - Added **circle markers** around launch sites using folium.Circle().
  - The radius of each circle represented the launch site's area.
  - **Purpose**: Highlights the launch site location and visually emphasizes its significance.
- **Lines (Polylines)**
  - Used folium.PolyLine() to draw lines connecting launch sites to their nearest geographical features:
    - **Coastline**
    - **Highway**
    - **Railway**
  - **Purpose**: Helps analyze the proximity of launch sites to infrastructure, which can impact logistics and rocket recovery.
- **Mouse Position Plugin**
  - Used MousePosition() to enable real-time display of latitude and longitude coordinates as the user moves the cursor.
  - **Purpose**: Facilitates exploration of different locations on the map.
- **Marker Clusters**
  - Used MarkerCluster() to group multiple markers with similar coordinates to prevent overlapping.
  - **Purpose**: Simplifies visualization by clustering overlapping points.

Why These Objects Were Added

Understanding Launch Site Distribution: The map allows for an interactive exploration of where SpaceX launch sites are located.

Analyzing Infrastructure Proximity: By drawing lines to coastlines, highways, and railways, we gain insights into how well-connected the launch sites are.

Visualizing Launch Success Rates: Using color-coded markers provides an intuitive representation of launch outcomes.

Enhancing Usability: The interactive features like **marker clusters** and **mouse position tracking** improve map navigation and data analysis.

https://github.com/cnabolouri/cnabolouri.github.io.git

Build a Dashboard with Plotly Dash

Plots and Graphs Added

Pie Chart - Success Count of Launches

- Displays the proportion of successful and failed launches.
- **Interaction**: Users can select a specific launch site from a dropdown to filter the success rate for that site.
- **Purpose**: Helps in assessing the performance of each launch site.

Scatter Plot - Payload vs. Launch Outcome

- Plots **Payload Mass (kg)** on the x-axis and **Launch Outcome (Success/Failure)** on the y-axis.
- Color-coded by **Booster Version** to compare the impact of different boosters.
- **Interaction**: Users can filter data based on payload range and launch site using a **Range Slider** and **Dropdown Menu**.
- **Purpose**: Allows exploration of whether payload mass affects launch success.

Dropdown Menu - Launch Site Selection

- Allows users to filter data for a specific **launch site** or view **all sites** together.
- **Purpose**: Enables targeted analysis of individual launch site performance.

Range Slider - Payload Mass Filter

- Enables users to select a specific payload mass range to update the scatter plot dynamically.
- **Purpose**: Helps in understanding how payload weight influences mission success.

Why These Plots and Interactions Were Added?

Interactivity: Users can dynamically adjust filters to focus on specific aspects of the data.

Comparative Analysis: The dashboard allows easy comparison of launch success rates across different sites and payload ranges.

Data Exploration: The interactive visualizations help uncover patterns between payload, booster versions, and mission success.

Decision Making: Stakeholders can use the insights to optimize launch strategies and improve mission success rates.

https://github.com/cnabolouri/cnabolouri.github.io.git

Predictive Analysis (Classification)

- **Model Development Process**
  - **Data Preprocessing**
    - Features were selected from the dataset, including **Flight Number, Payload Mass, Orbit, Launch Site, Reusability Factors, Booster Version, and more**.
    - Categorical variables (**Orbit, Launch Site, etc.**) were converted to numerical representations using **One-Hot Encoding**.
    - The dataset was split into **training and testing sets** (80% training, 20% testing).
    - **Standardization** was applied to numerical features to ensure consistent scaling.
  - **Model Selection**
    - Several machine learning classification models were tested, including:
      - **Logistic Regression**
      - **Support Vector Machines (SVM)**
      - **Decision Tree**
      - **K-Nearest Neighbors (KNN)**
    - Hyperparameter tuning was performed using **GridSearchCV** to find the best model configurations.
  - **Model Evaluation**
    - Each model was evaluated on the **test dataset** using:
      - **Accuracy Score**: Measures the percentage of correct predictions.
      - **Confusion Matrix**: Analyzes True Positives, True Negatives, False Positives, and False Negatives.
      - **Precision & Recall**: Evaluates the model's ability to predict successful launches accurately.
  - **Best Performing Model**
    - The **Support Vector Machine (SVM) with an optimized kernel** achieved the highest accuracy.
    - The **Decision Tree and Logistic Regression** also performed well but had slightly lower accuracy.
    - **KNN** was less effective due to sensitivity to feature scaling and data distribution.
  - **Model Optimization**
    - The **best hyperparameters** were determined for each model using **cross-validation**.
    - **Feature importance analysis** was conducted to understand which factors contributed most to launch success.

https://github.com/cnabolouri/cnabolouri.github.io.git

Results

Exploratory Data Analysis (EDA) Results:

Mapped launch sites and analyzed their proximity to infrastructure (coastlines, railways, highways).

Visualized payload vs. success rate trends to identify optimal launch conditions.

Interactive Analytics (Dashboard):

Developed a **Plotly Dash application** to dynamically explore launch success rates.

Implemented interactive visualizations such as pie charts and scatter plots for deeper insights.

Predictive Analysis Results:

Trained multiple machine learning models to predict booster landing success.

Evaluated model accuracy using **confusion matrix and classification metrics**.

Section 2

Insights drawn from EDA

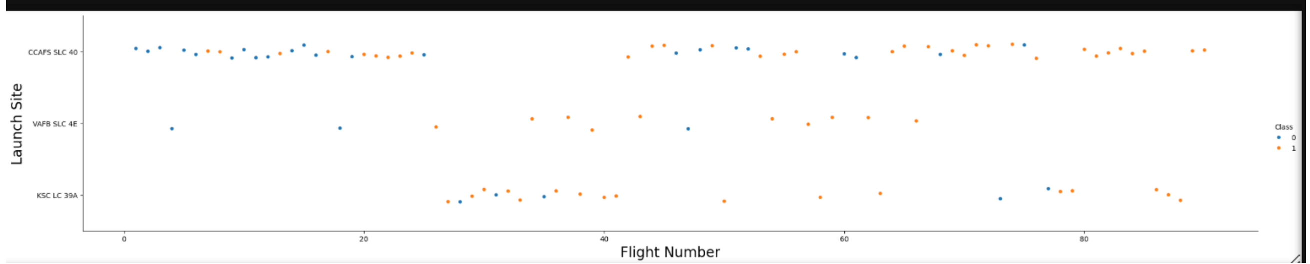
https://github.com/cnabolouri/cnabolouri.github.io.git

Flight Number vs. Launch Site

```python
# Plot Flight Number vs Launch Site
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect=5)

# Labels
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)

# Show the plot
plt.show()
```



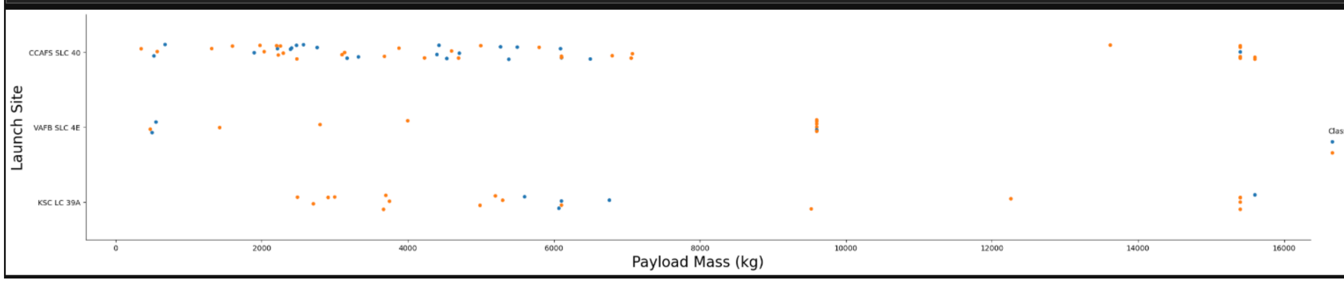
https://github.com/cnabolouri/cnabolouri.github.io.git

Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
# Plot Payload Mass vs Launch Site
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect=5)

# Labels
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)

# Show the plot
plt.show()
```



https://github.com/cnabolouri/cnabolouri.github.io.git

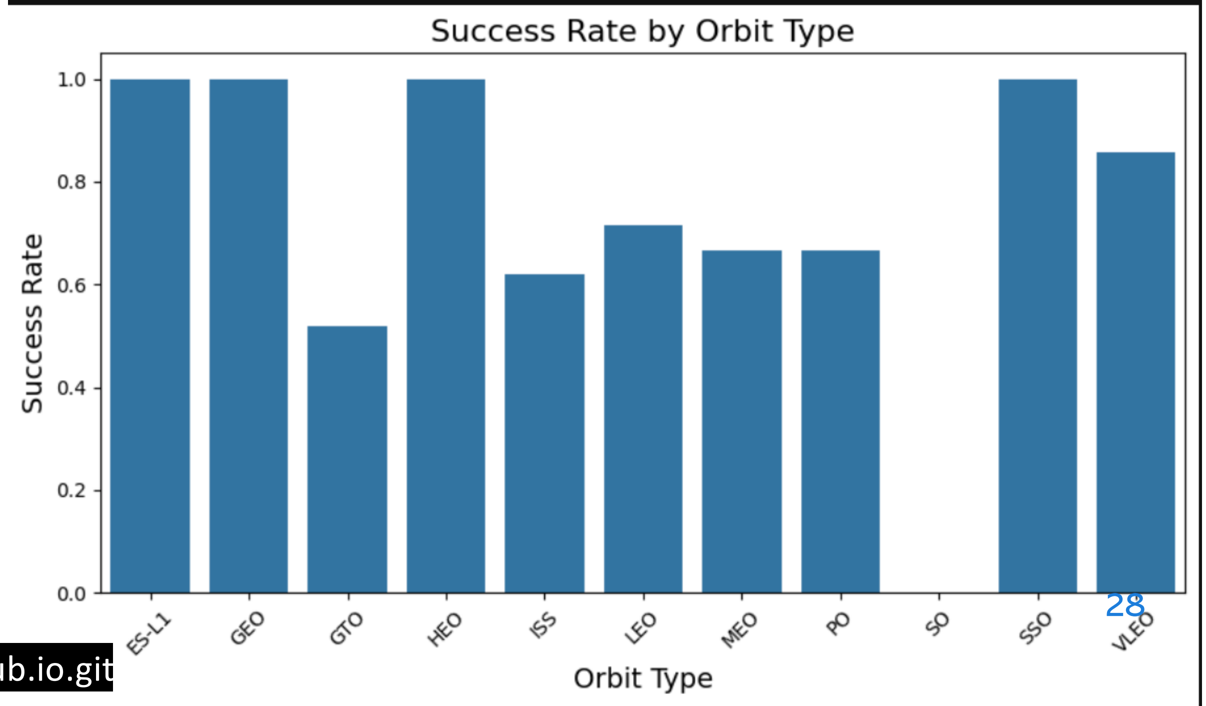
Success Rate vs. Orbit Type

```
# Group data by Orbit and compute the mean success rate
orbit_success = df.groupby("Orbit")["Class"].mean().reset_index()

# Create a bar chart for success rate of each orbit type
plt.figure(figsize=(10,5))
sns.barplot(x="Orbit", y="Class", data=orbit_success)

# Labels and title
plt.xlabel("Orbit Type", fontsize=14)
plt.ylabel("Success Rate", fontsize=14)
plt.title("Success Rate by Orbit Type", fontsize=16)
plt.xticks(rotation=45)

# Show plot
plt.show()
```



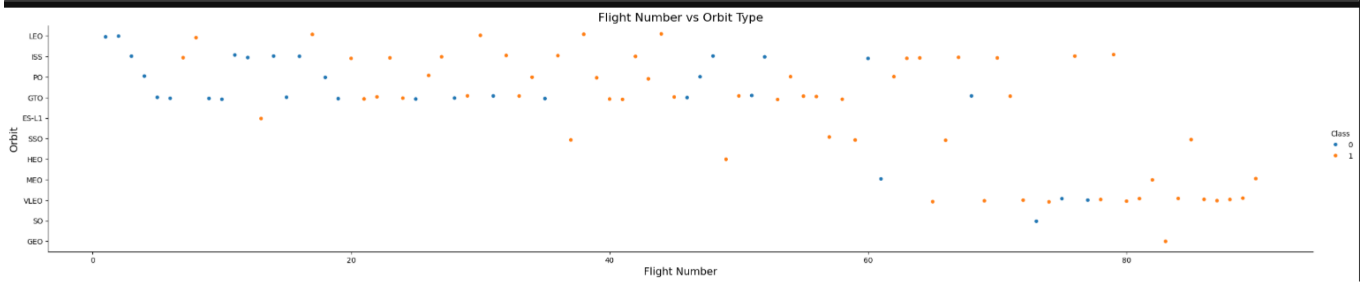
https://github.com/cnabolouri/cnabolouri.github.io.git

Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
# Scatter plot: Flight Number vs Orbit Type
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect=5)

# Labels
plt.xlabel("Flight Number", fontsize=14)
plt.ylabel("Orbit", fontsize=14)
plt.title("Flight Number vs Orbit Type", fontsize=16)

# Show plot
plt.show()
```



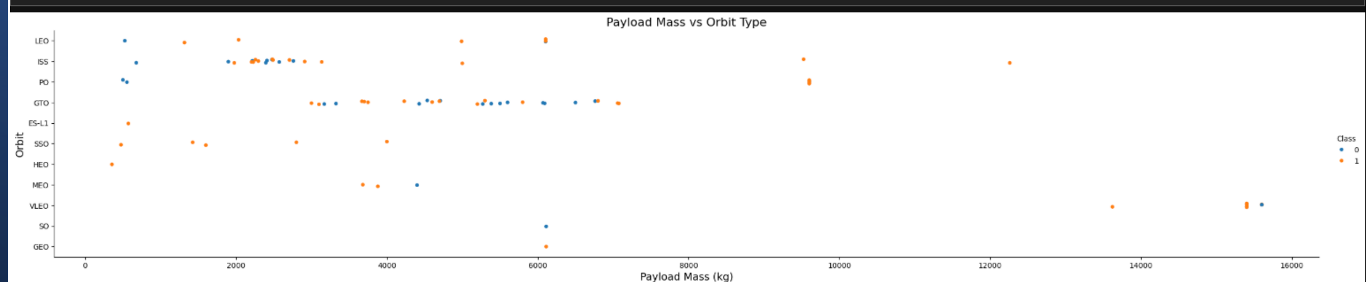
https://github.com/cnabolouri/cnabolouri.github.io.git

Payload vs. Orbit Type

```python
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
# Scatter plot: Payload Mass vs Orbit Type
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect=5)

# Labels
plt.xlabel("Payload Mass (kg)", fontsize=14)
plt.ylabel("Orbit", fontsize=14)
plt.title("Payload Mass vs Orbit Type", fontsize=16)

# Show plot
plt.show()
```



https://github.com/cnabolouri/cnabolouri.github.io.git

Launch Success Yearly Trend

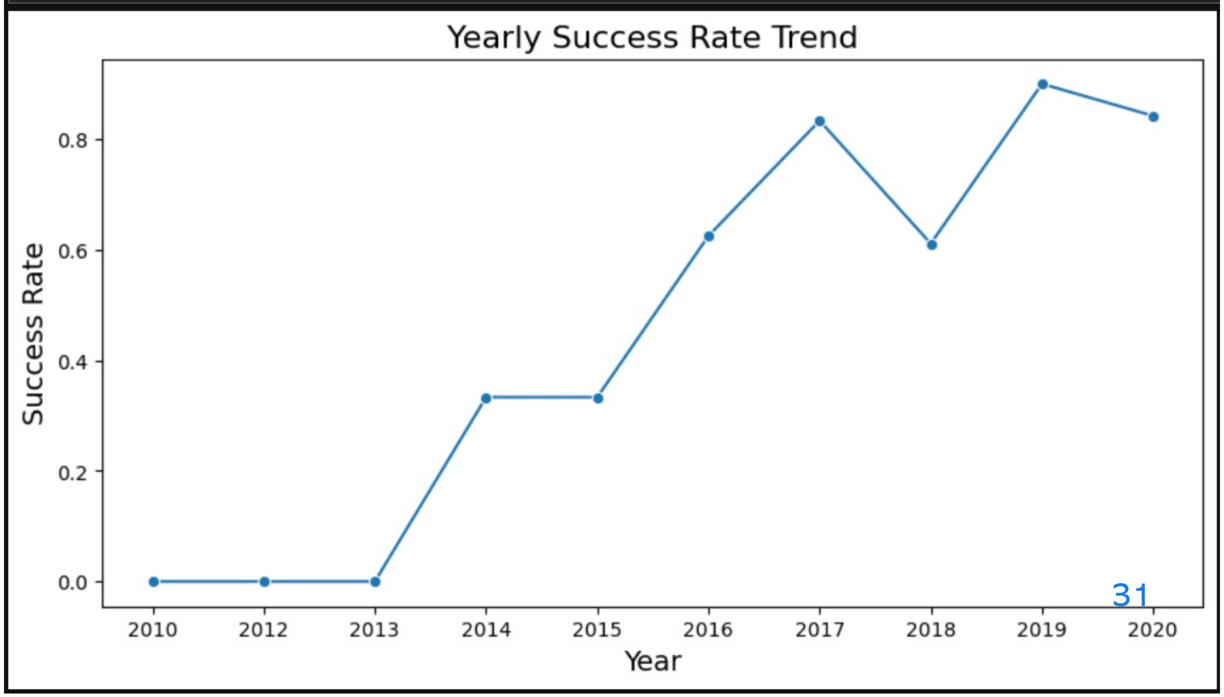
https://github.com/cnabolouri/cnabolouri.github.io.git

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
# Group by year and compute the success rate
yearly_success = df.groupby("Date")["Class"].mean().reset_index()

# Plot success rate trend
plt.figure(figsize=(10,5))
sns.lineplot(x="Date", y="Class", data=yearly_success, marker="o")

# Labels and title
plt.xlabel("Year", fontsize=14)
plt.ylabel("Success Rate", fontsize=14)
plt.title("Yearly Success Rate Trend", fontsize=16)

# Show plot
plt.show()
```



All Launch Site Names

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

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

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

https://github.com/cnabolouri/cnabolouri.github.io.git

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	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

https://github.com/cnabolouri/cnabolouri.github.io.git

Total Payload Mass

```
%%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

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_Mass
FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Avg_Payload_Mass
2534.6666666666665

https://github.com/cnabolouri/cnabolouri.github.io.git

First Successful Ground Landing Date

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

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

First_Successful_Landing
2015-12-22

https://github.com/cnabolouri/cnabolouri.github.io.git

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT DISTINCT Booster_Version
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

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

https://github.com/cnabolouri/cnabolouri.github.io.git

Total Number of Successful and Failure Mission Outcomes

https://github.com/cnabolouri/cnabolouri.github.io.git

```
%%sql
SELECT Mission_Outcome, COUNT(*) AS Count
FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

https://github.com/cnabolouri/cnabolouri.github.io.git

```
%%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

```
%%sql
SELECT SUBSTR(Date, 6, 2) AS Month, Booster_Version, Launch_Site, Landing_Outcome
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Failure (drone ship)'
AND SUBSTR(Date, 1, 4) = '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)

https://github.com/cnabolouri/cnabolouri.github.io.git

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

https://github.com/cnabolouri/cnabolouri.github.io.git

```
%%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.
```

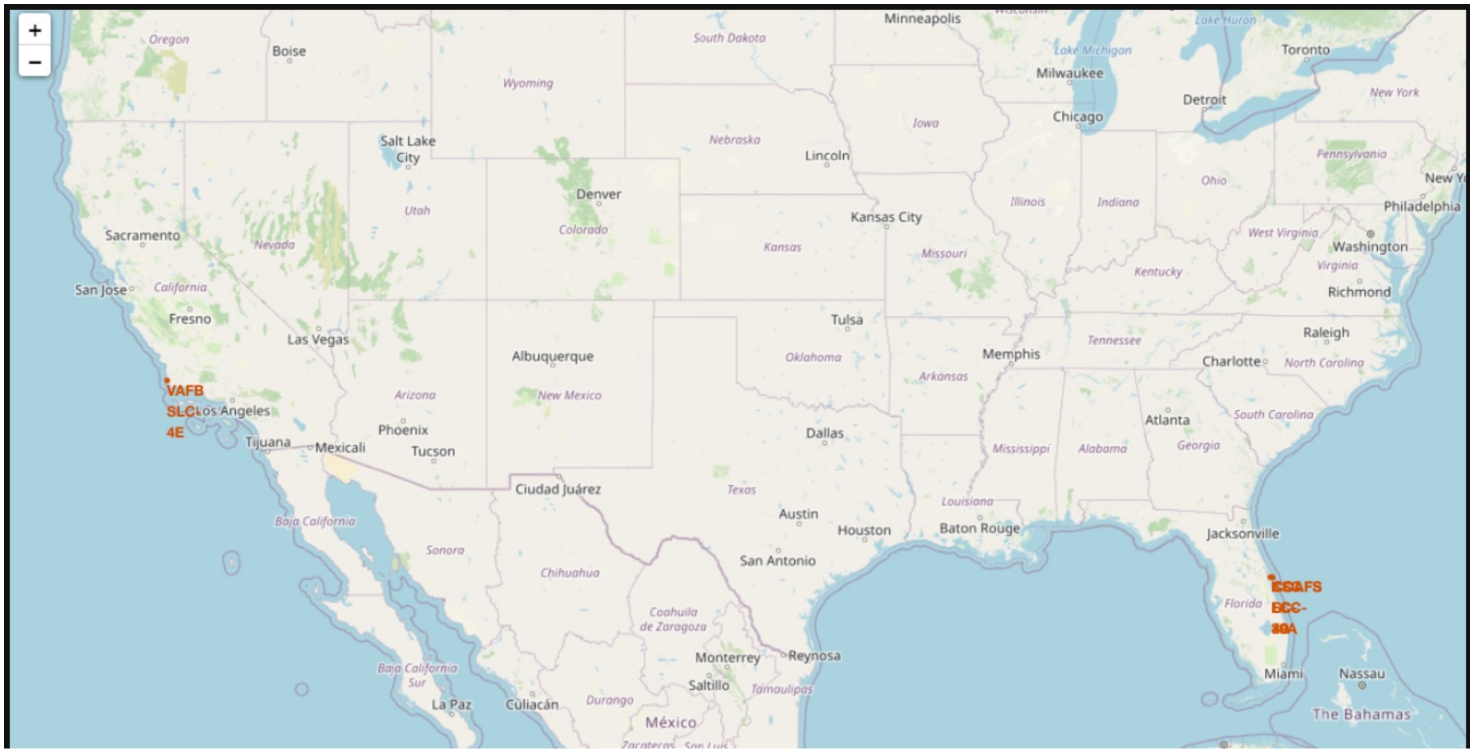
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

https://github.com/cnabolouri/cnabolouri.github.io.git

Visualizing SpaceX Launch Sites on an Interactive Map

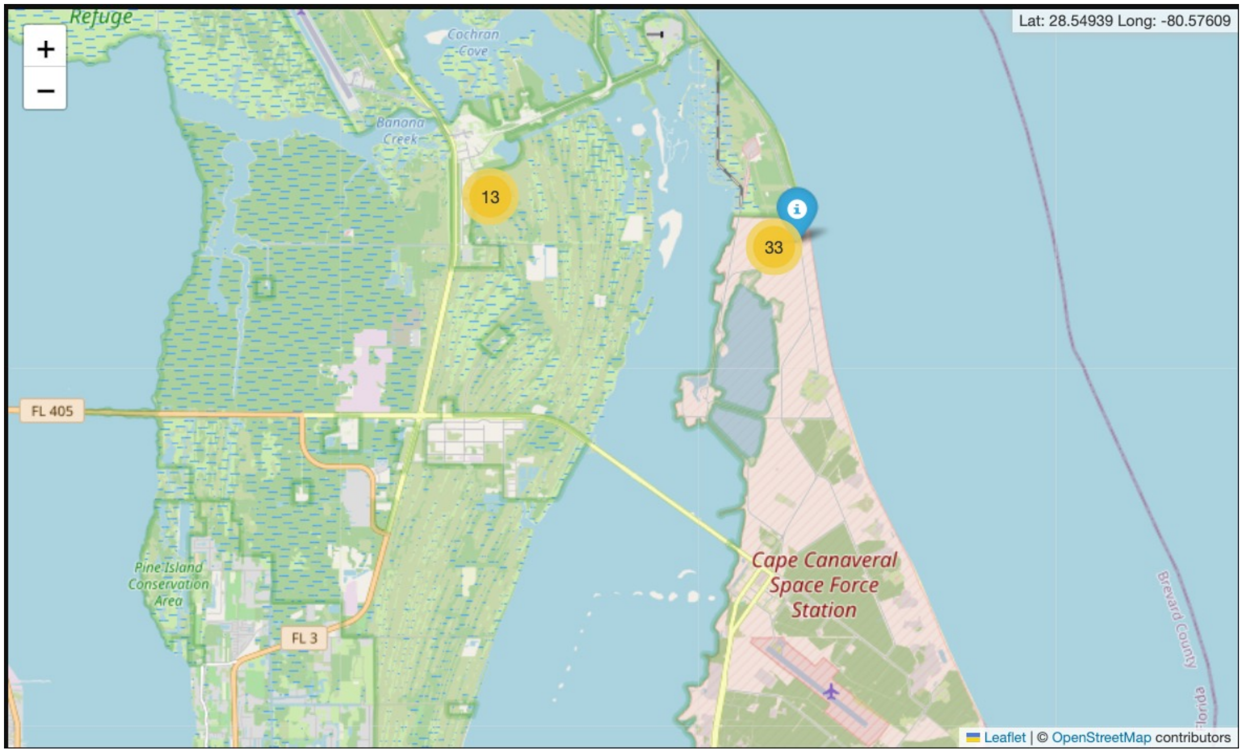


https://github.com/cnabolouri/cnabolouri.github.io.git

Mapping SpaceX Launch Outcomes with Color-Coded Markers

https://github.com/cnabolouri/cnabolouri.github.io.git

Analyzing Launch Site Proximities: Distance to Railways, Highways, and Coastlines



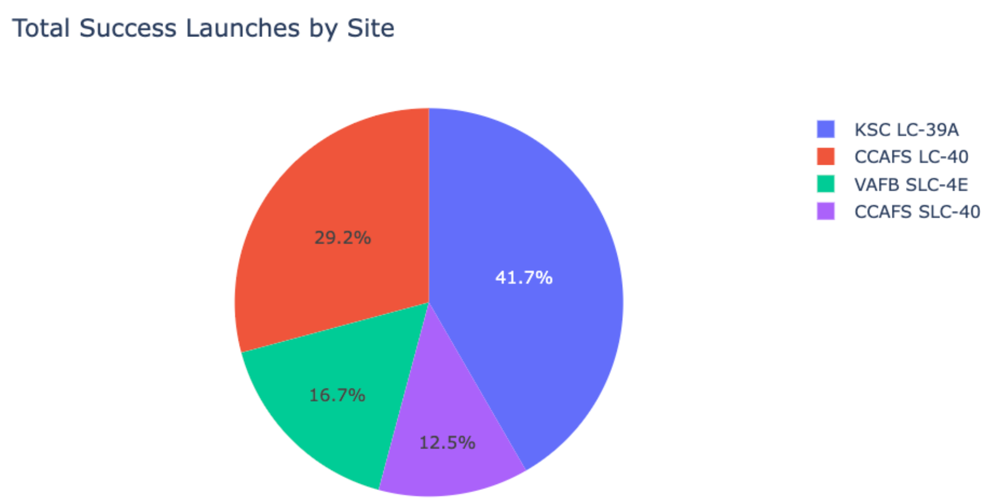
https://github.com/cnabolouri/cnabolouri.github.io.git

Section 4

Build a Dashboard with Plotly Dash

https://github.com/cnabolouri/cnabolouri.github.io.git

Launch Success Distribution: Pie Chart Overview

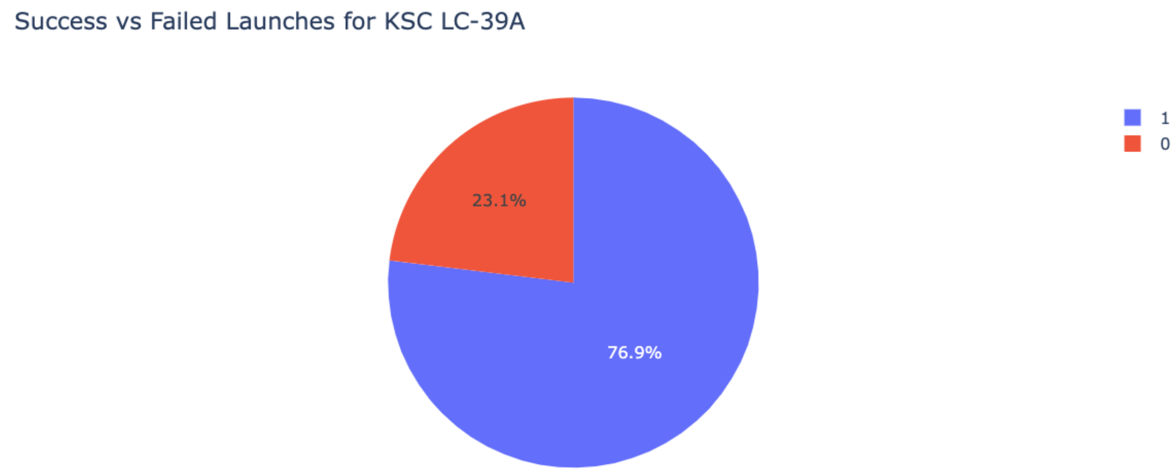


Insights and Key Takeaways:

- The **KSC LC-39A** site appears to be the most preferred and successful launch site.
- **CCAFS LC-40 and VAFB SLC-4E** also show significant contributions.
- **CCAFS SLC-40 has the lowest success share**, possibly due to fewer launches or technical challenges at the site.

https://github.com/cnabolouri/cnabolouri.github.io.git

Launch Success Ratio by Site – Highest Performing Site: Pie Chart Overview

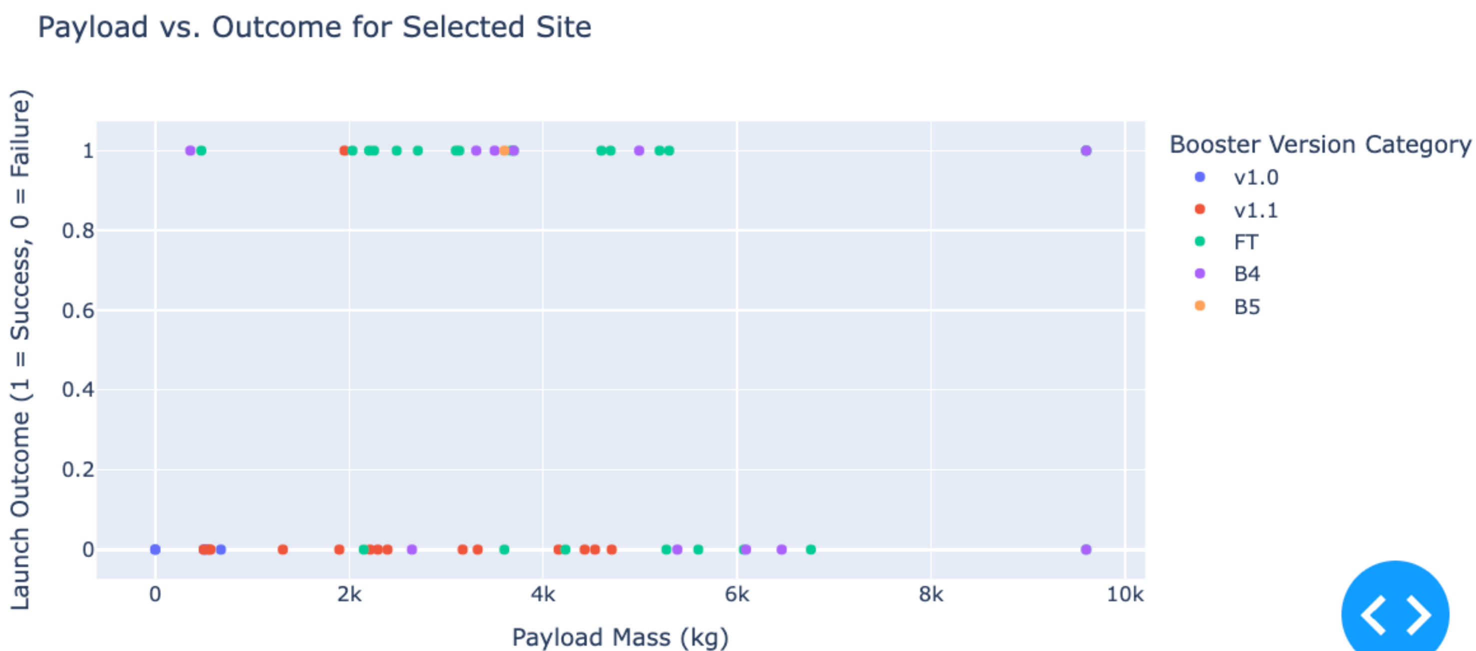


Key Findings

- Identify the site with the **highest success ratio**.
- If one site significantly outperforms others, it may indicate better launch conditions, improved booster versions, or superior technology.

https://github.com/cnabolouri/cnabolouri.github.io.git

Payload vs. Outcome for All Sites: Scatter Plot Overview



Key Findings:

•SpaceX's later booster versions (FT, B4, B5) tend to have higher success rates.

•Launches with lower payloads tend to have more failures, possibly due to early-stage technology.

•Heavy payload launches (closer to 10,000 kg) show more successful landings, suggesting advancements in booster design.

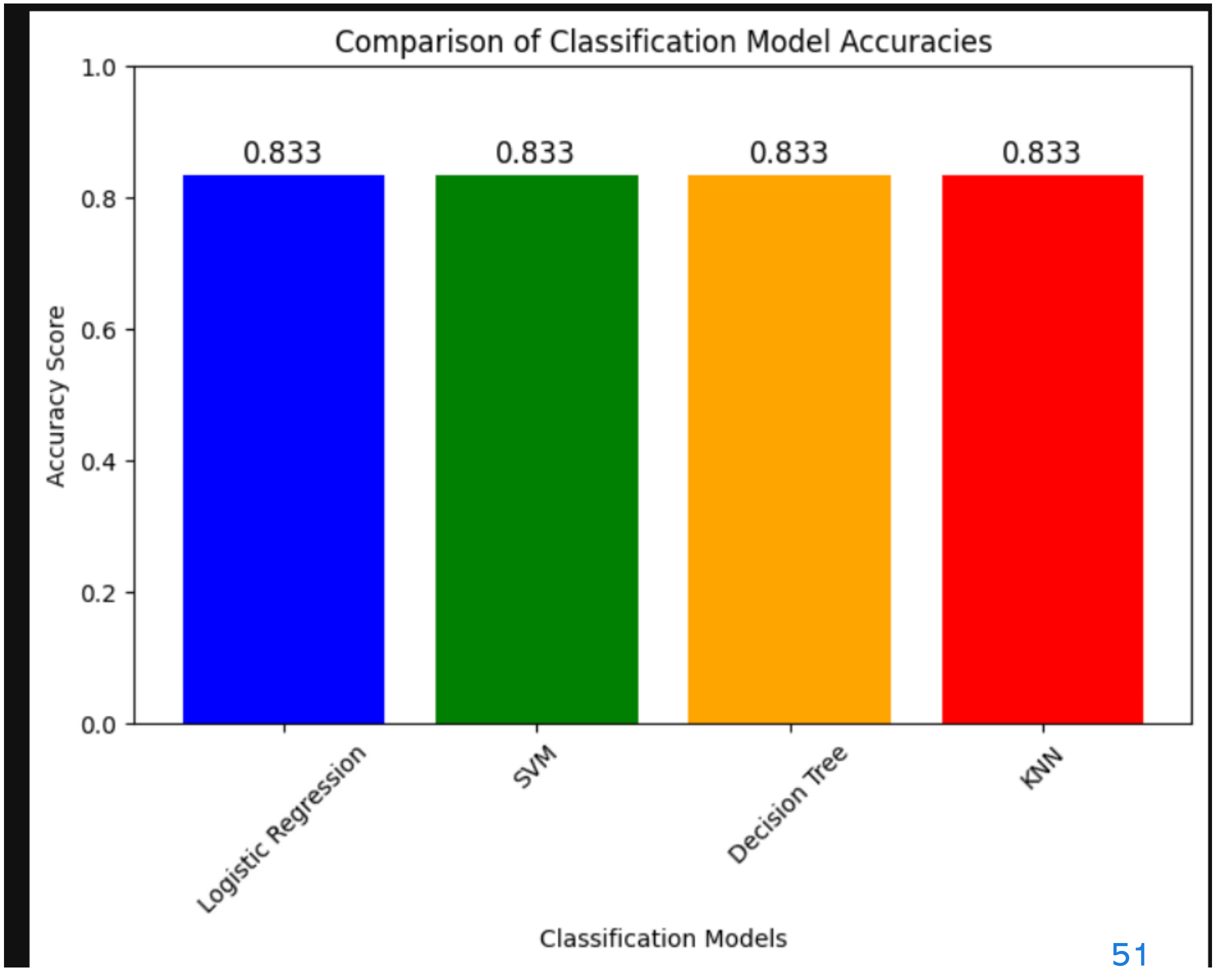
https://github.com/cnabolouri/cnabolouri.github.io.git

Section 5

Predictive Analysis (Classification)

https://github.com/cnabolouri/cnabolouri.github.io.git

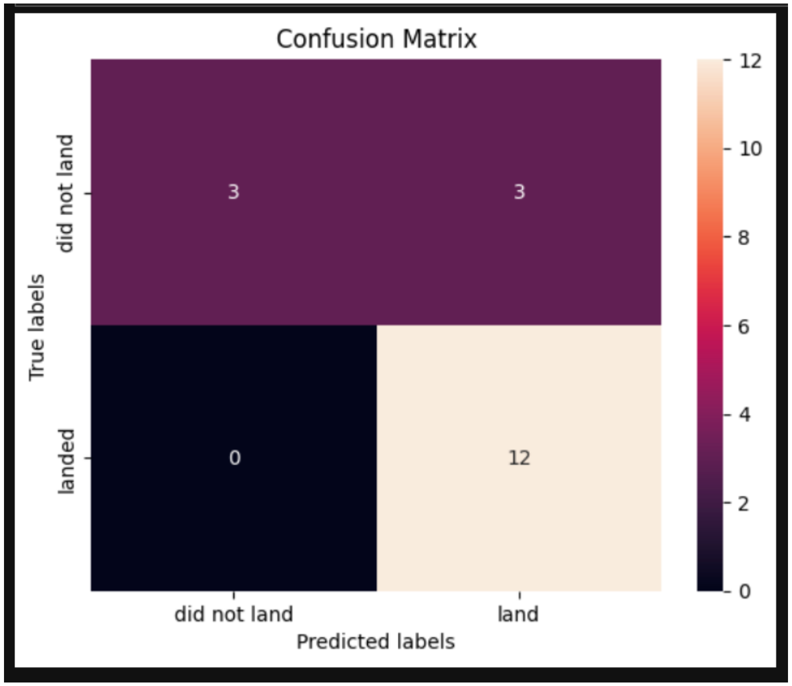
Classification Accuracy



https://github.com/cnabolouri/cnabolouri.github.io.git

Confusion Matrix

https://github.com/cnabolouri/cnabolouri.github.io.git



Key Findings:

- High Correct Classification Rate: The model correctly predicts all 12 successful landings (100% recall) and identifies 3 out of 6 failed landings.
- Zero False Negatives: It ensures successful landings are not misclassified as failures, which is crucial for decision-making.
- Trade-off between False Positives and False Negatives: The model has 3 false positives, but this is acceptable. Misclassifying a failed landing as successful is more serious than having false positives.
- Effective Handling of Non-Linearity: Decision trees capture complex patterns influenced by factors like weather and payload, unlike linear models.

Conclusions

Data-Driven Insights: Analyzed SpaceX launch data to identify key factors influencing booster landing success.

Geospatial Analysis: Mapped launch sites, examined proximity to coastlines, highways, and railways, and evaluated site efficiency.

Interactive Dashboard: Developed a Plotly Dash tool for visualizing launch success rates and payload relationships dynamically.

Machine Learning Predictions:

- Trained multiple models to predict booster landing success.
- **Decision Tree emerged as the best model**, minimizing false negatives and providing the most reliable predictions.

Key Takeaways:

- Machine learning can optimize **mission success rates** and **reduce costs**.
- Geospatial analysis helps in **strategic site selection** and **launch planning**.
- Predictive models can assist in **future mission planning and risk assessment**.

https://github.com/cnabolouri/cnabolouri.github.io.git


Thank you!
https://github.com/cnabolouri/cnabolouri.github.io.git