IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies

  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction

- Summary of all results

  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

SpaceX has revolutionized the space industry by significantly reducing the cost of rocket launches. Through the reuse of the Falcon 9's first stage, SpaceX has brought launch prices down to as low as $62 million, compared to traditional providers charging upwards of $165 million per launch. This innovative approach to reusability has disrupted the market, making space travel more accessible and cost-effective.

As a data scientist working for a startup competing with SpaceX, this project aims to develop a machine learning pipeline to predict the landing outcome of a rocket's first stage. By accurately forecasting landing success, the company can strategically price its launch services and optimize operational efficiency.

Key Problems to Address:

- Identifying the key factors that influence the success or failure of a rocket landing.
- Understanding the relationship between different variables and their impact on landing outcomes.
- Determining the optimal conditions required to maximize the probability of a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling
  - Data was processed using one-hot encoding for categorical features
  - Cleaned and structured the dataset to remove inconsistencies and missing values.
- 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** to predict first-stage landing success ,
  - Evaluated multiple models
  - Optimized model performance through **hyperparameter tuning using GridSearch**

Data Collection

Data collection involves gathering and measuring relevant information on targeted variables to enable analysis, answer key questions, and evaluate outcomes. For this project, data was obtained using two primary methods:

1.REST API (SpaceX API)

- Data was retrieved by sending **GET requests** to the SpaceX API.
- The API response was in **JSON format**, which was then parsed and converted into a **Pandas DataFrame** using json_normalize().
- The dataset was cleaned by handling missing values and ensuring consistency.

2.Web Scraping (Wikipedia)

- Launch records were extracted from Wikipedia using **BeautifulSoup**.
- The relevant **HTML table** containing launch details was identified and parsed.
- The data was structured into a **Pandas DataFrame** for further analysis.

By combining API data with web-scraped data, a comprehensive dataset was built for machine learning and predictive analysis.

Data Collection – SpaceX API

Get request for rocket launch data using API

Use json_normalize method to convert json result to dataframe

Performed data cleaning and filling the missing value

From:
https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex-data-collection-api.ipynb

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

```
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want
#  and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters
# and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list
#  and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype
# and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from URL

Create a BeautifulSoup from the HTML response

Extract all column/variable names from the HTML header

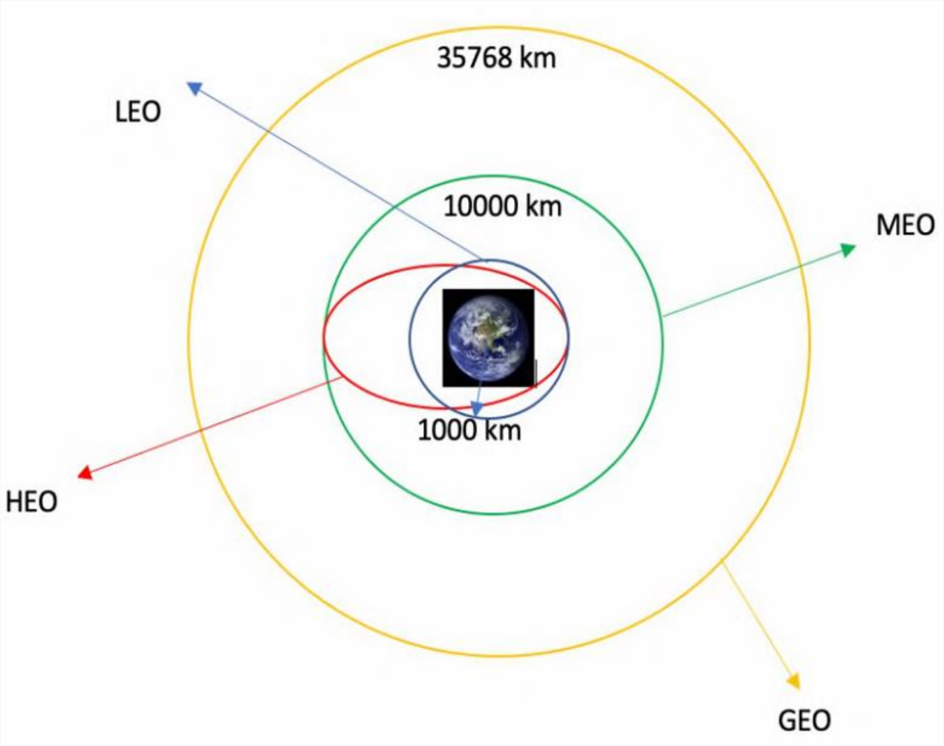
- FROM: https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/Data-Collection%20using%20web_scraping.ipynb

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text)
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

Data Wrangling



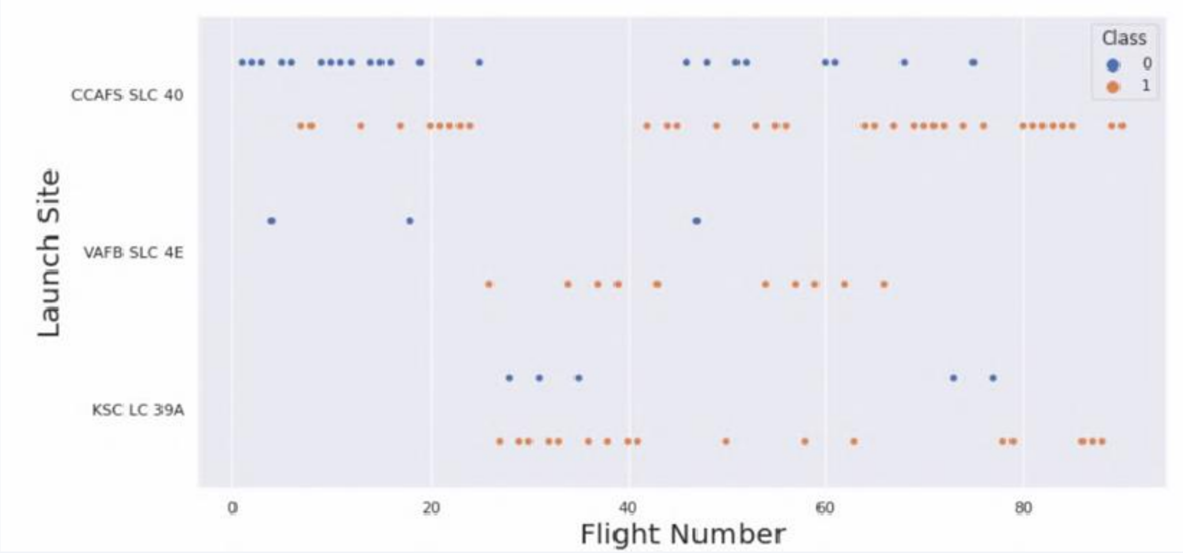
From: https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex-Data%20wrangling.ipynb

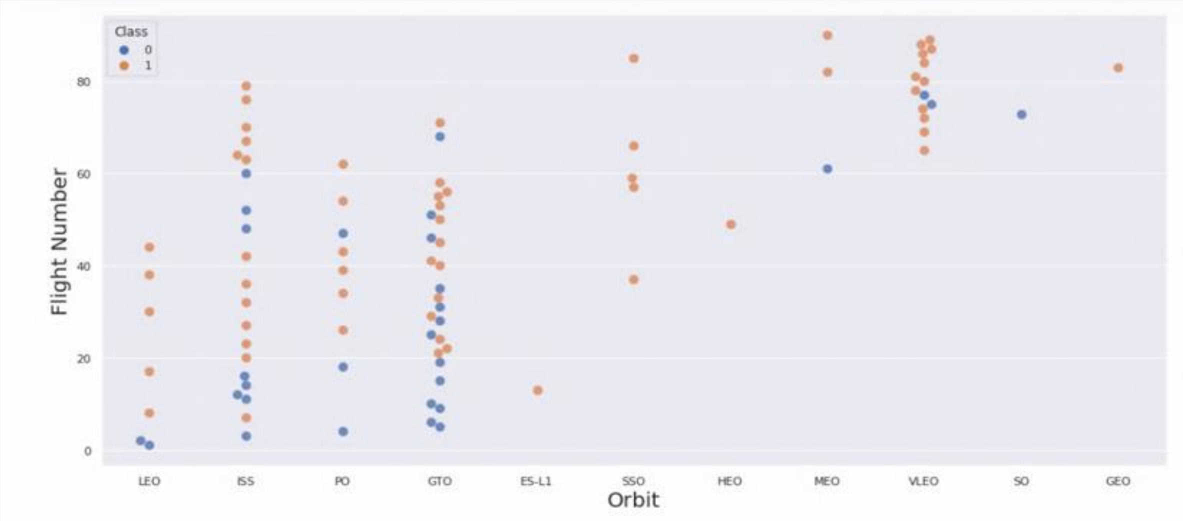
Data wrangling is the process of cleaning, transforming, and structuring raw data to make it more accessible for analysis and visualization. In this project, the following steps were performed:

1. **Launch Site Analysis**
   - Calculated the **number of launches per site** to identify frequently used locations.
2. **Mission Outcome by Orbit Type**
   - Analyzed the **occurrence of mission outcomes** across different orbit types to understand success rates in various conditions.
3. **Creating a Landing Outcome Label**
   - Extracted and transformed the **landing outcome** column into a **binary classification label** (successful vs. unsuccessful landings) to facilitate machine learning modeling.
4. **Exporting Data**
   - The processed dataset was **exported to a CSV file** for further analysis and modeling.

This structured approach ensures data consistency and prepares it for **Exploratory Data Analysis (EDA)** and **machine learning applications**.

EDA with Data Visualization





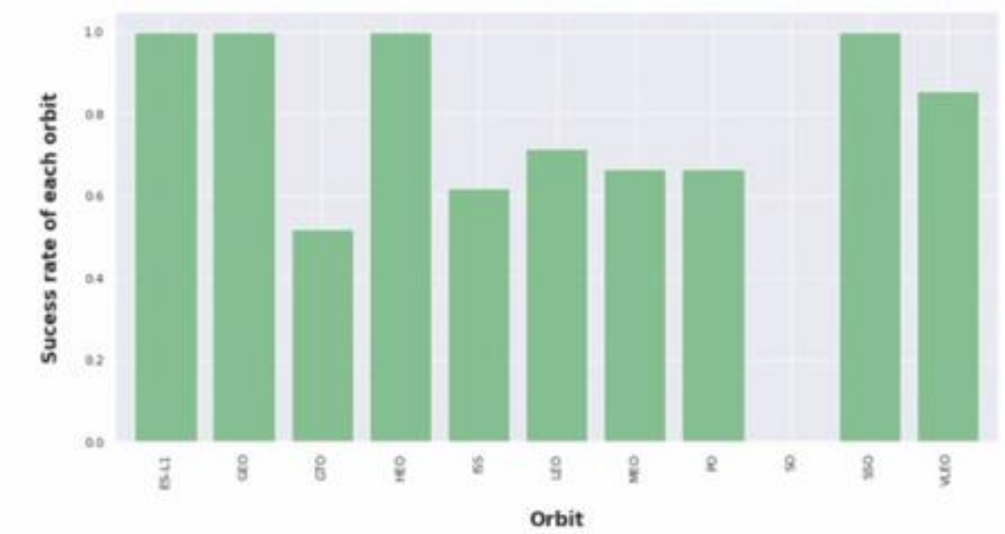
To understand the relationships between key attributes and their impact on landing success, we utilized **scatter plots** to visualize dependencies. The following attribute pairs were analyzed:

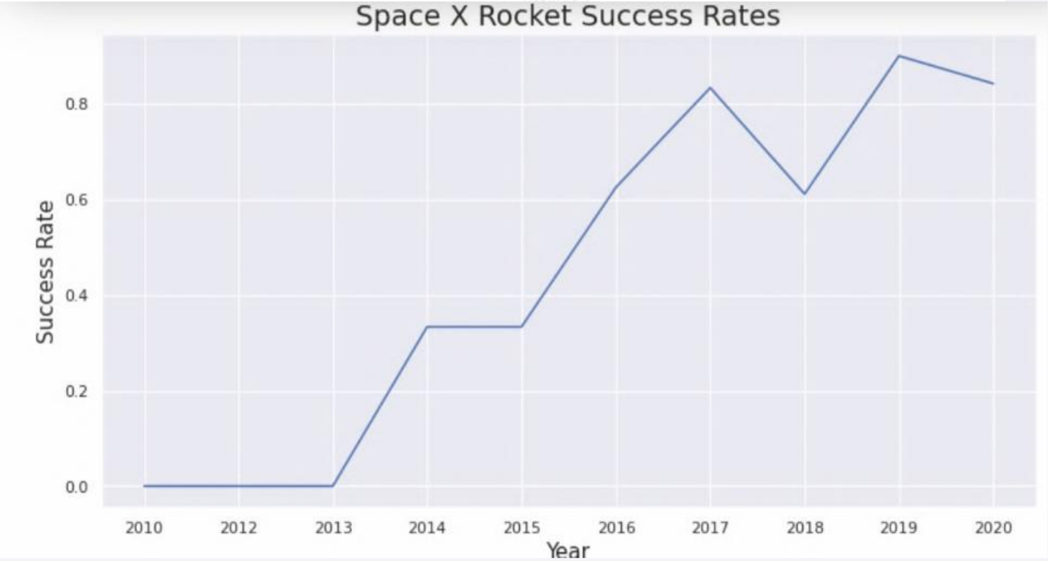
- **Payload vs. Flight Number** – To assess how payload size influences mission frequency.
- **Flight Number vs. Launch Site** – To identify patterns in launch site usage.
- **Payload vs. Launch Site** – To determine if certain sites handle heavier payloads more frequently.
- **Flight Number vs. Orbit Type** – To examine if mission frequency varies across different orbit types.
- **Payload vs. Orbit Type** – To explore how payload mass influences the choice of orbit.

Scatter plots provided insights into attribute dependencies and helped in identifying key factors affecting **landing success rates**. By recognizing patterns in the data, we gained a clearer understanding of the most influential variables for successful rocket landings.

From: https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20visualizations.ipynb

EDA with Data Visualization





After gaining initial insights from scatter plots, we used additional visualization techniques to further analyze trends and relationships:

1. **Bar Graphs**
   1. Bar graphs were used to **compare the success rates across different orbit types**.
   2. This helped in identifying which orbits had the highest probability of a successful landing.
2. **Line Graphs**
   1. Line plots were employed to **analyze launch success trends over time**.
   2. This visualization provided insights into how launch success rates have evolved over the years.

Feature Engineering

To prepare the data for predictive modeling, we applied **feature engineering**, which included:

- **Creating dummy variables** for categorical columns (e.g., launch sites, orbit types).
- Transforming categorical data into a numerical format to be used in **machine learning models**.

This step ensured that the dataset was optimized for training classification models to predict landing success.

From: https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20visualizations.ipynb

EDA with SQL

To gain deeper insights into the dataset and better understand the relationships between various attributes, several **SQL queries** were executed. These queries helped in extracting key information about launch sites, payloads, boosters, and mission outcomes. Here are the queries performed:

- **Displaying Launch Sites:** Query: Displaying the names of all **launch sites** in the dataset.
- **Filtering Launch Sites**: Displaying 5 records where **launch sites** begin with the string "CCA".
- **Payload Mass by NASA (CRS):** Displaying the **total payload mass** carried by boosters launched by **NASA (CRS)**.
- **Average Payload Mass for Booster Version F9 v1.1**: Displaying the **average payload mass** carried by **booster version F9 v1.1**.
- **First Successful Landing Date on Ground Pad**: Listing the **date when the first successful landing outcome** was achieved on a **ground pad**.
- **Booster Success on Drone Ship with Payload Mass Range**: Listing the names of **boosters** that succeeded in landing on a **drone ship** and had a **payload mass between 4000 and 6000**.
- **Counting Mission Outcomes**: Displaying the **total number of successful** and **failed mission outcomes**.
- **Booster Versions with Maximum Payload Mass**: Listing the **booster versions** that carried the **maximum payload mass**.
- **Failed Landings on Drone Ship (2015)**: Listing the **failed landing outcomes** on the **drone ship** in **2015**, including their **booster versions** and **launch sites**.
- **Landing Outcomes Ranking (2010-2017):** Ranking the count of **landing outcomes (success/failure)** between **2010-06-04** and **2017-03-20**, ordered in **descending order**.

These SQL queries allowed for a detailed examination of the dataset and helped uncover trends, anomalies, and valuable insights that informed the analysis and modeling stages.

https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

To visualize launch data on an interactive map, we used **Folium**, a Python library that allows for easy creation of interactive maps. The following steps were taken:

1.Launch Site Markers

- We extracted the **latitude and longitude coordinates** for each launch site.
- Each launch site was marked with a **circle marker**, and a **label** displaying the launch site name was added to the map.

2.Launch Outcome Classification

- The dataset was assigned **launch outcomes** (success or failure) to classes 0 (failure) and 1 (success).
- Different **color markers** (red for failure, green for success) were used to represent these outcomes.
- **MarkerCluster()** was employed to group the markers based on proximity for better visualization, especially for sites with numerous launch records.

3.Distance Calculation Using Haversine Formula

- We utilized the **Haversine formula** to calculate the **distance** between launch sites and various **landmarks** (e.g., railways, highways, coastlines, nearby cities).
- This allowed us to answer questions like:
  - **How close are the launch sites to railways, highways, and coastlines?**
  - **How close are the launch sites to nearby cities?**

By combining **interactive mapping** and **distance analysis**, we gained valuable insights into the spatial relationships between launch sites and their surrounding infrastructure.

https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/Geospatial-Data-Visualization-with-Folium.ipynb

Build a Dashboard with Plotly Dash

To provide a more dynamic and user-interactive experience with the launch data, we developed an interactive **dashboard using Plotly Dash**. This allowed users to explore the data, visualize trends, and customize their analysis. Here's what was included in the dashboard:

1. Total Launches by Site

- We created **pie charts** displaying the **total number of launches** at each launch site.
- The pie chart provides a quick and intuitive overview of how launch activity is distributed across different sites.

2. Outcome vs. Payload Mass

- We plotted **scatter graphs** to visualize the relationship between **launch outcomes** (success or failure) and **payload mass (Kg)** for different **booster versions**.
- This scatter plot helps identify if payload mass has any significant influence on the outcome of the launch, and how different booster versions perform.

The dashboard provided an interactive way for users to explore the data, offering features like filtering, zooming, and drilling down into specific information.

https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Building the Model

Load the Dataset

- Import the dataset into **NumPy** and **Pandas** for efficient data manipulation.

Data Transformation and Splitting

- Clean and preprocess the data (e.g., encoding categorical features).
- Split the data into **training** and **test** datasets.

Model Selection

- Decide on the type of machine learning model (e.g., **Logistic Regression**, **Random Forest**, etc.) based on the problem.

Hyperparameter Tuning

- Use **GridSearchCV** to find the optimal hyperparameters for each model and fit the data accordingly.

Evaluating the Model

Model Accuracy

- Evaluate the accuracy of each model using performance metrics (e.g., **accuracy score**, **precision**, **recall**).

Tuned Hyperparameters

- Extract the best hyperparameters identified by **GridSearchCV** for each algorithm.

Confusion Matrix

- Plot the **confusion matrix** to visualize the performance of the model (i.e., false positives, false negatives, true positives, and true negatives).

Improving the Model

Feature Engineering

- Enhance the model's performance by creating new features or refining existing ones.

Algorithm Tuning

- Adjust model parameters and test various algorithms to improve the predictive power.

Find the Best Model

- The **best-performing model** will be the one with the highest **accuracy score** and optimal hyperparameters after tuning.

https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

Results

Exploratory Data Analysis (EDA) Results

- Insights into relationships between attributes like **payload and flight number,** and **launch outcomes** across **orbit types** and **launch sites** using scatter plots and bar graphs.

Interactive Analytics Demo (Screenshots)

- **Interactive map** showing **launch sites** and their proximity to infrastructure.
- **Dashboard** with **pie charts** and **scatter plots** to explore launches by site and the relationship between **outcomes and payload mass**.

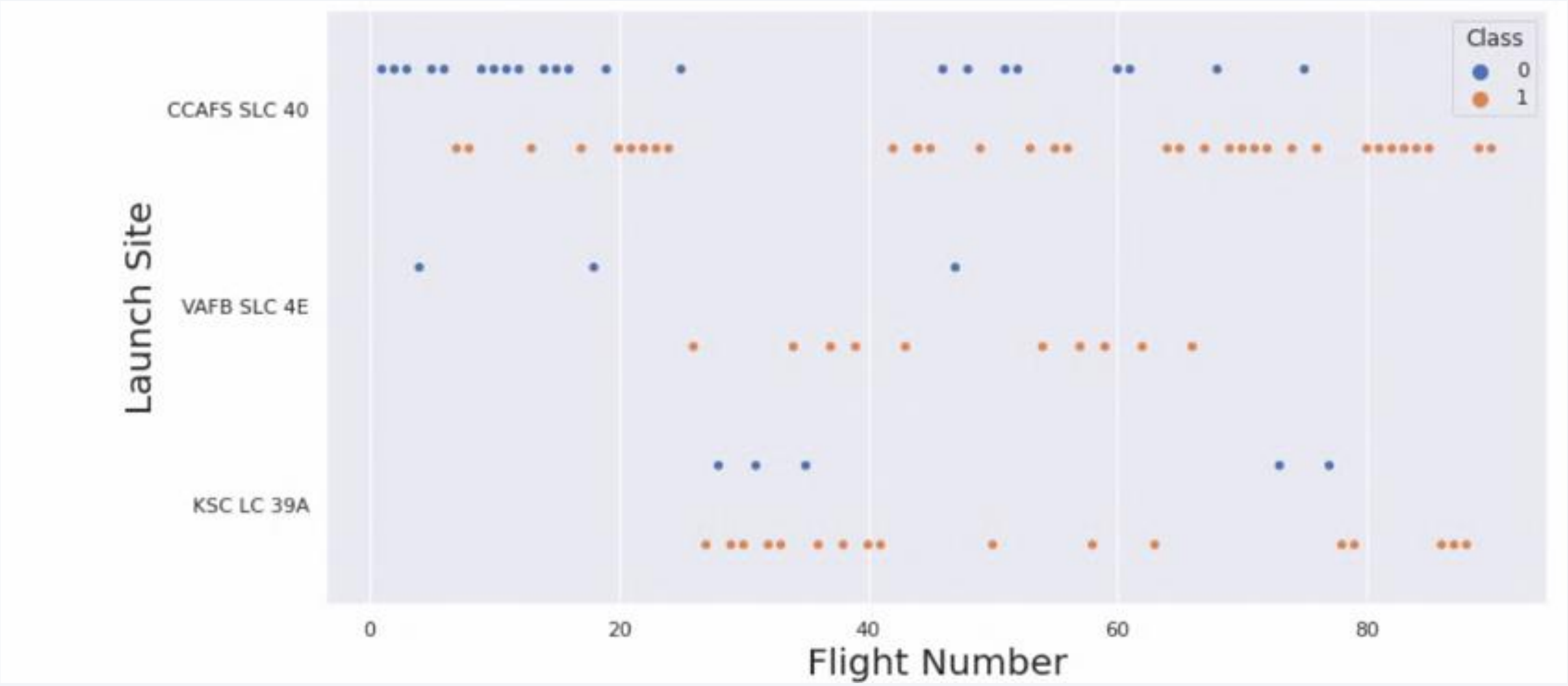
Predictive Analysis Results

- Performance of **classification models** (e.g., **Logistic Regression**, **Random Forest**) in predicting **landing success**, with evaluation metrics like **accuracy** and **precision**.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



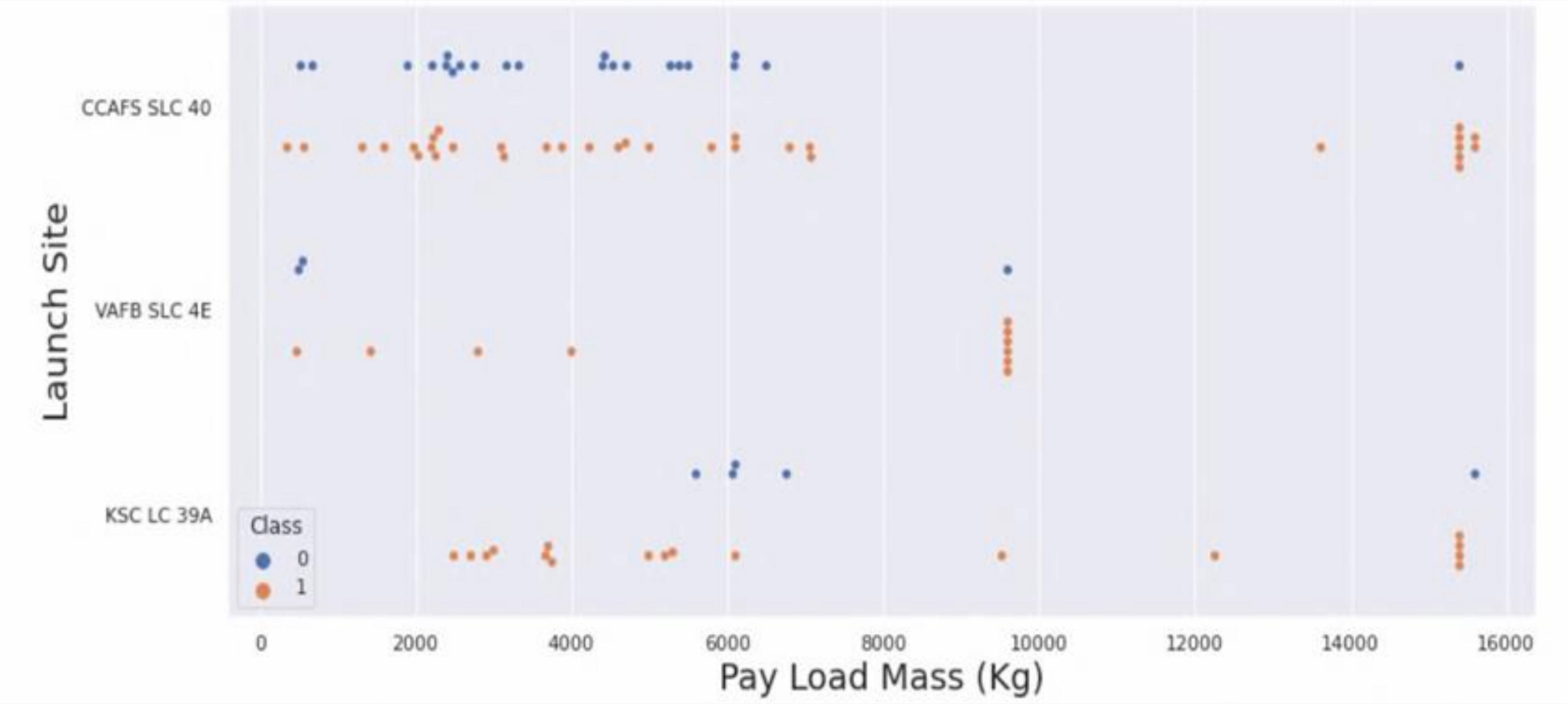
This scatter plot suggests that launch sites with a higher number of flights tend to have a greater success rate.

However, the CCAFS SLC-40 site does not follow this pattern as strongly.

Payload vs. Launch Site

This scatter plot indicates that when the payload mass exceeds 7000kg, the probability of a successful launch significantly increases.

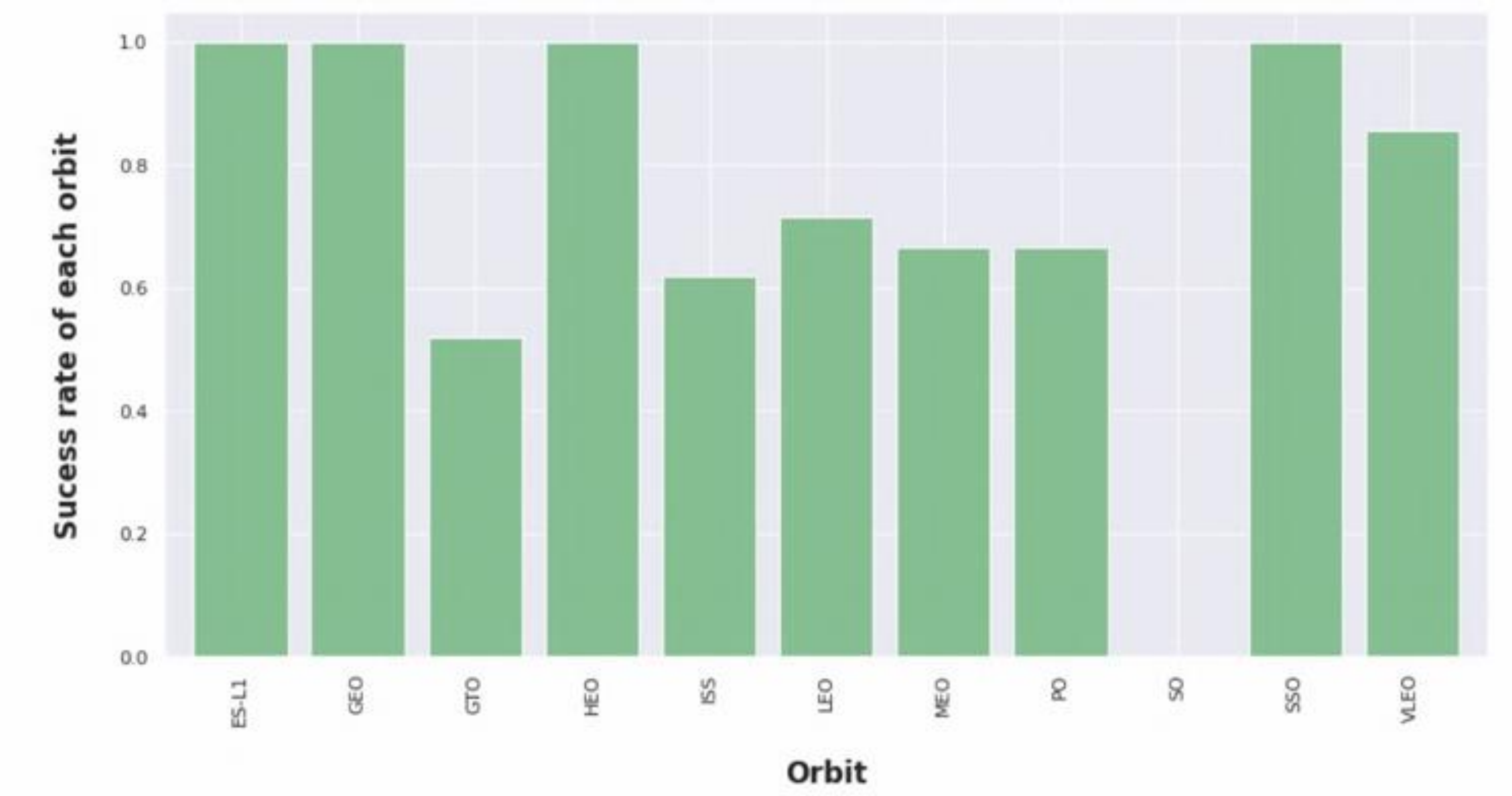
However, there is no clear pattern suggesting that the launch site influences the success rate based on payload mass.



Success Rate vs. Orbit Type

This figure illustrates the potential influence of orbit types on landing outcomes, as some orbits, such as SSO, HEO, GEO, and ES-L1, show a 100% success rate, while the SO orbit has a 0% success rate.

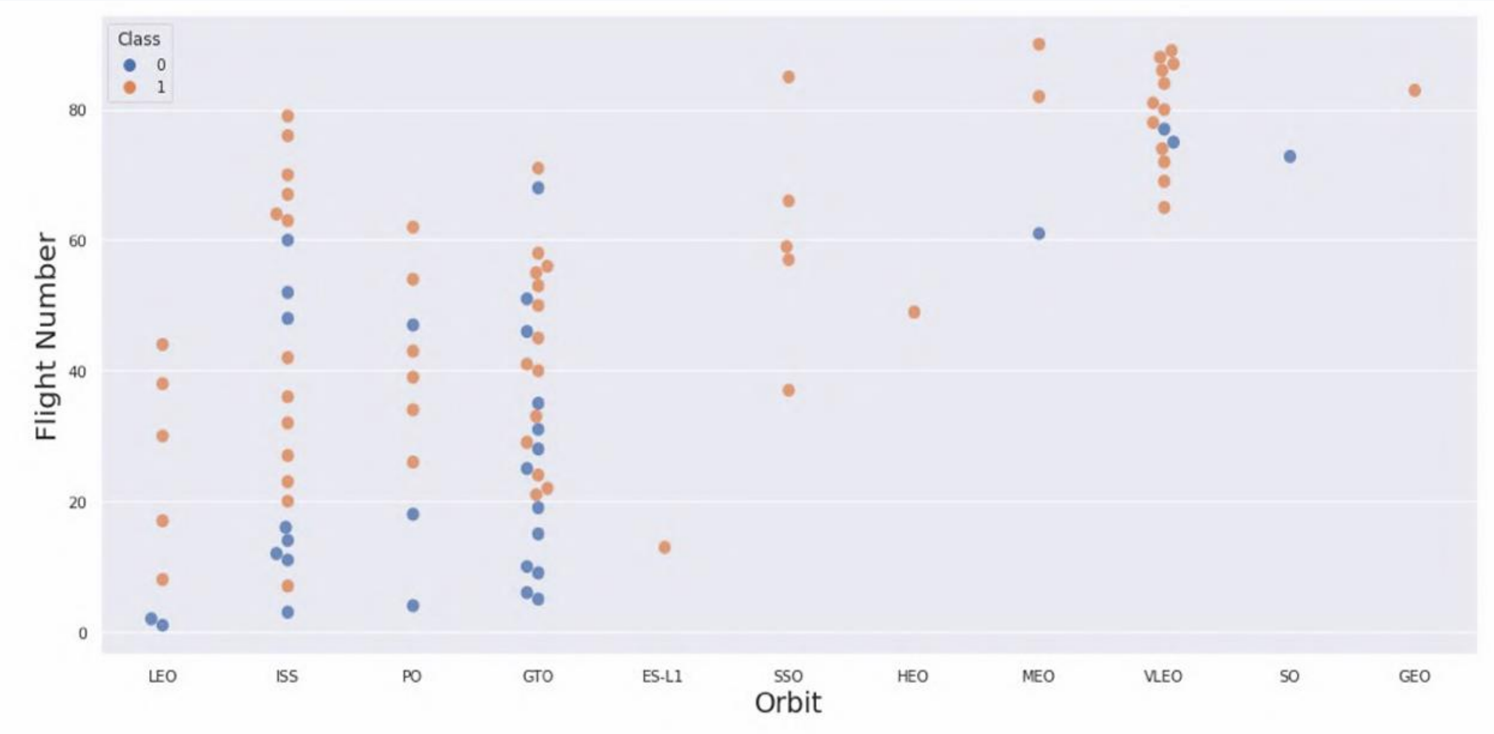
However, a deeper analysis reveals that some of these orbits, including GEO, SO, HEO, and ES-L1, have only a single occurrence. This suggests that a larger dataset is needed to identify patterns or trends before drawing definitive conclusions.



Flight Number vs. Orbit Type

This scatter plot indicates that, in general, a higher number of flights per orbit corresponds to a greater success rate, particularly for LEO.

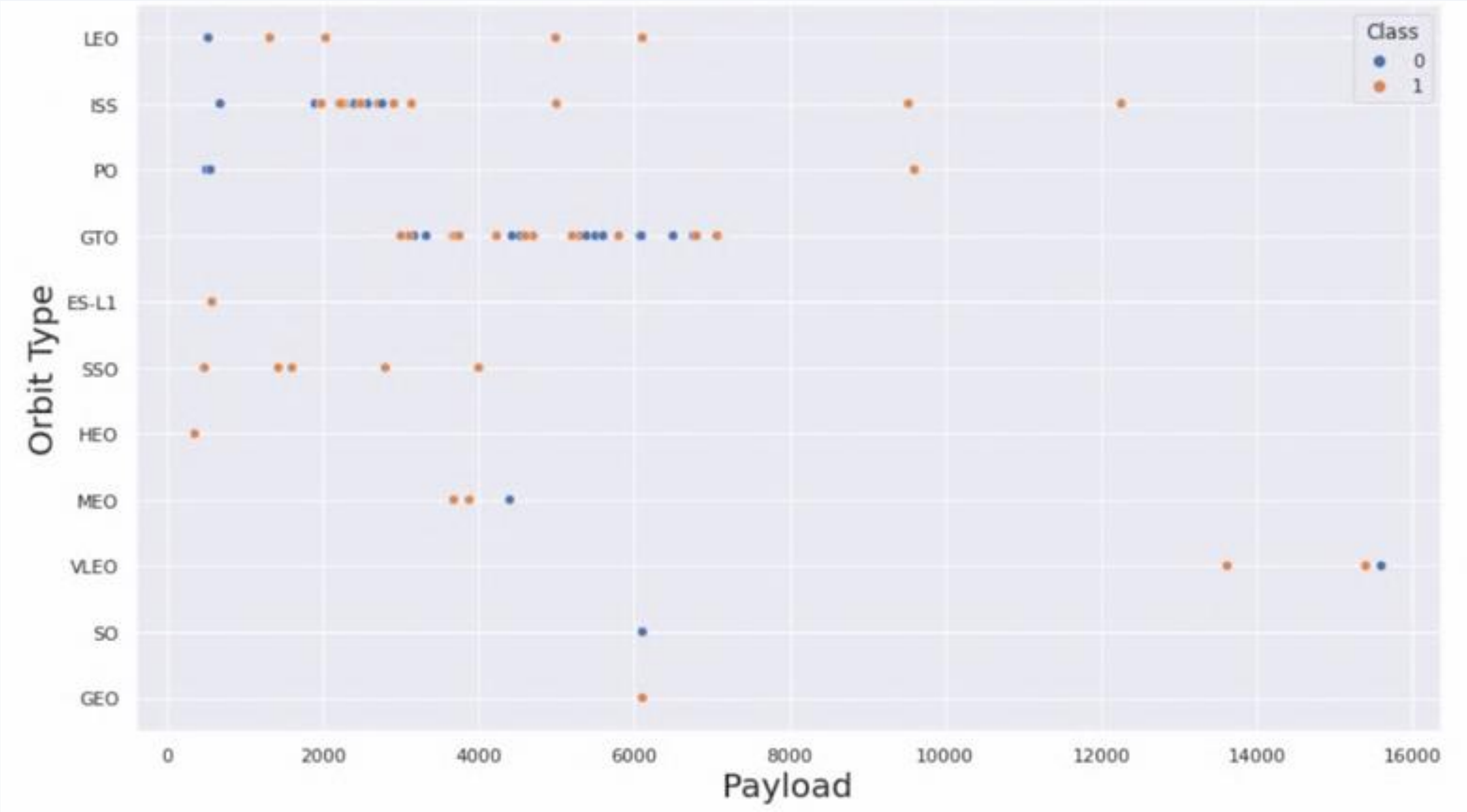
However, GTO does not exhibit a clear relationship between flight count and success rate. Additionally, orbits with only a single occurrence should be excluded from this analysis, as more data is needed to establish a reliable pattern.



Payload vs. Orbit Type

Heavier payloads have a positive impact on LEO, ISS, and PO orbits but a negative impact on MEO and VLEO orbits. GTO orbit appears to show no clear relationship between payload mass and success rate.

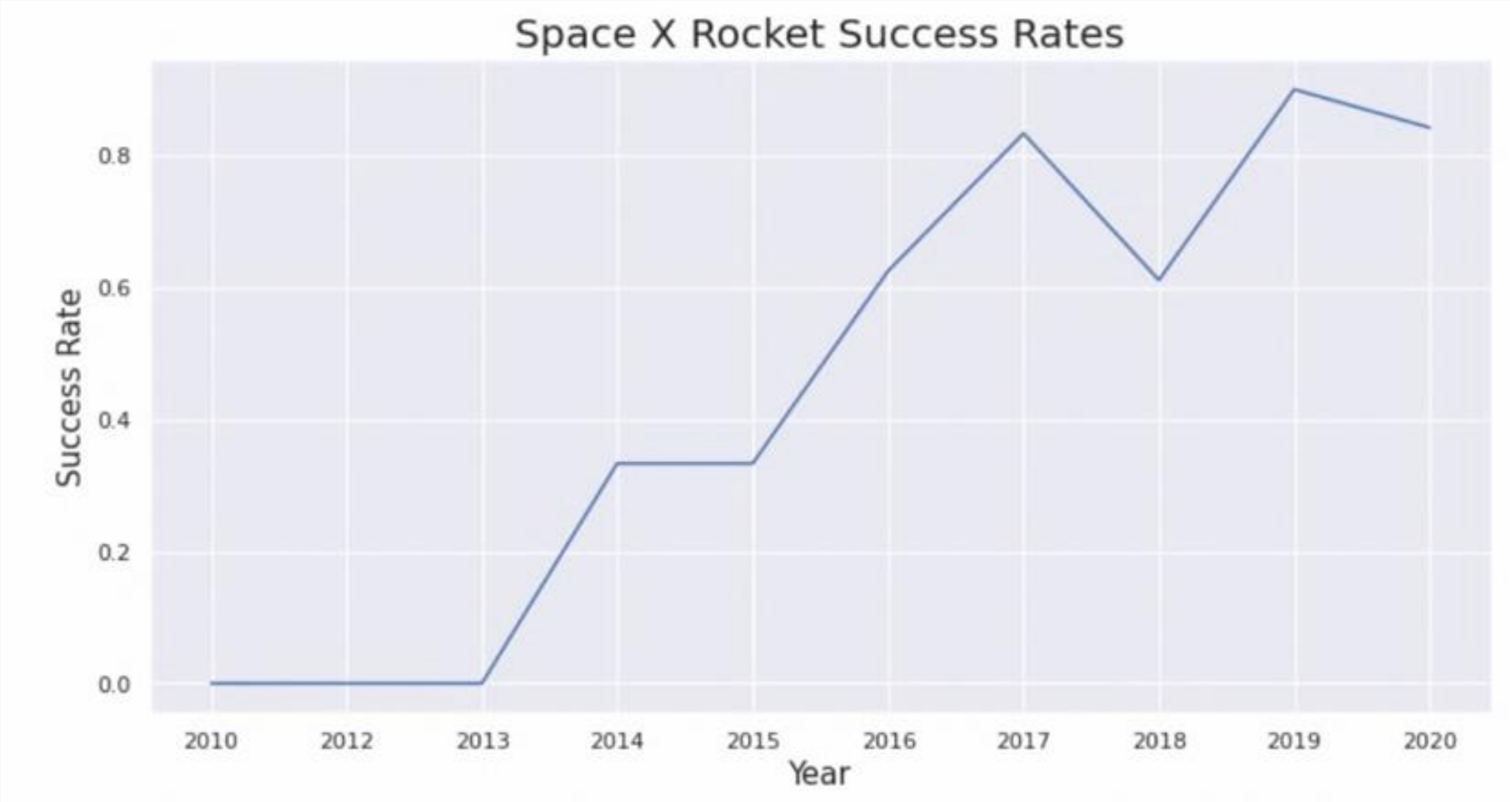
Meanwhile, SO, GEO, and HEO orbits require more data for a conclusive analysis.



Launch Success Yearly Trend

These figures clearly show an increasing trend in success rates from 2013 to 2020.

If this trend continues in the coming years, the success rate is expected to steadily rise, potentially reaching 100%.



All Launch Site Names

We applied the keyword DISTINCT to retrieve only unique launch sites from the SpaceX dataset

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

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

Launch Site Names Begin with 'CCA'

We used the query above to display five records where the launch site names begin with 'CCA'.

```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select SUM(PAYLOAD_MASS__KG_) AS total_payload_mass from SPACEXTABLE where Customer = 'NASA (CRS)'
```

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

We use the min() function to find the result We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pac
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters that successfully landed on a drone ship. We then applied the AND condition to select those with a payload mass greater than 4000 but less than 6000.

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Total Number of Successful and Failure Mission Outcomes

We used the wildcard % to filter for records where the MissionOutcome contains either 'success' or 'failure'.

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

Failure Mission
1

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

We determined the booster that carried the maximum payload using a subquery in the WHERE clause along with the MAX() function.

2015 Launch Records

We used a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes on the drone ship, along with their booster versions and launch site names for the year 2015.

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

We selected landing outcomes and the count of landing outcomes from the data, using the WHERE clause to filter for landing outcomes between June 4, 2010, and March 20, 2010. We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to sort the grouped landing outcomes in descending order.“

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Note: There seems to be a date range issue (June 4, 2010, to March 20, 2010), as the end date is earlier than the start date. If that’s a mistake, you might want to adjust the range.

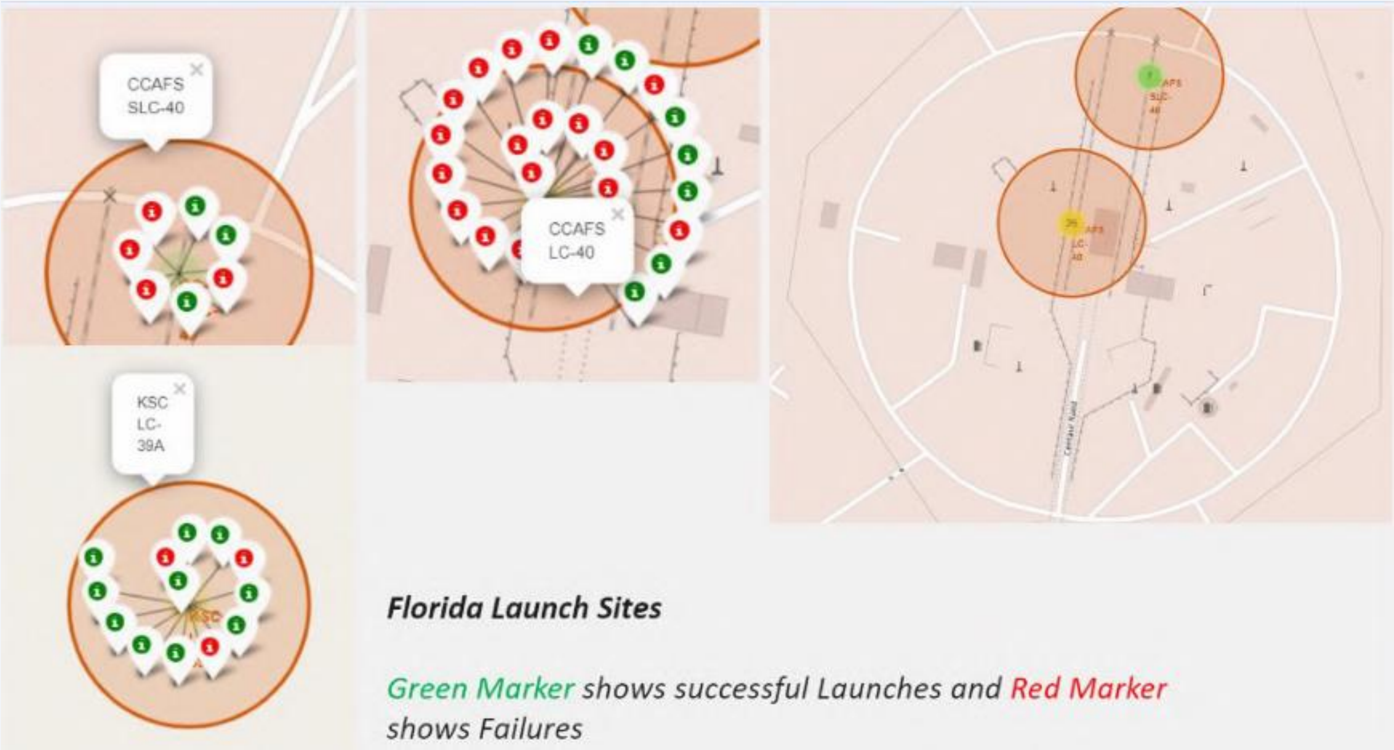
Section 3

Launch Sites Proximities Analysis

Launch Sites Locations

- We can see that all the SpaceX launch sites are located inside the United States

Visualizing Launch Sites with Labeled Color Markers



Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

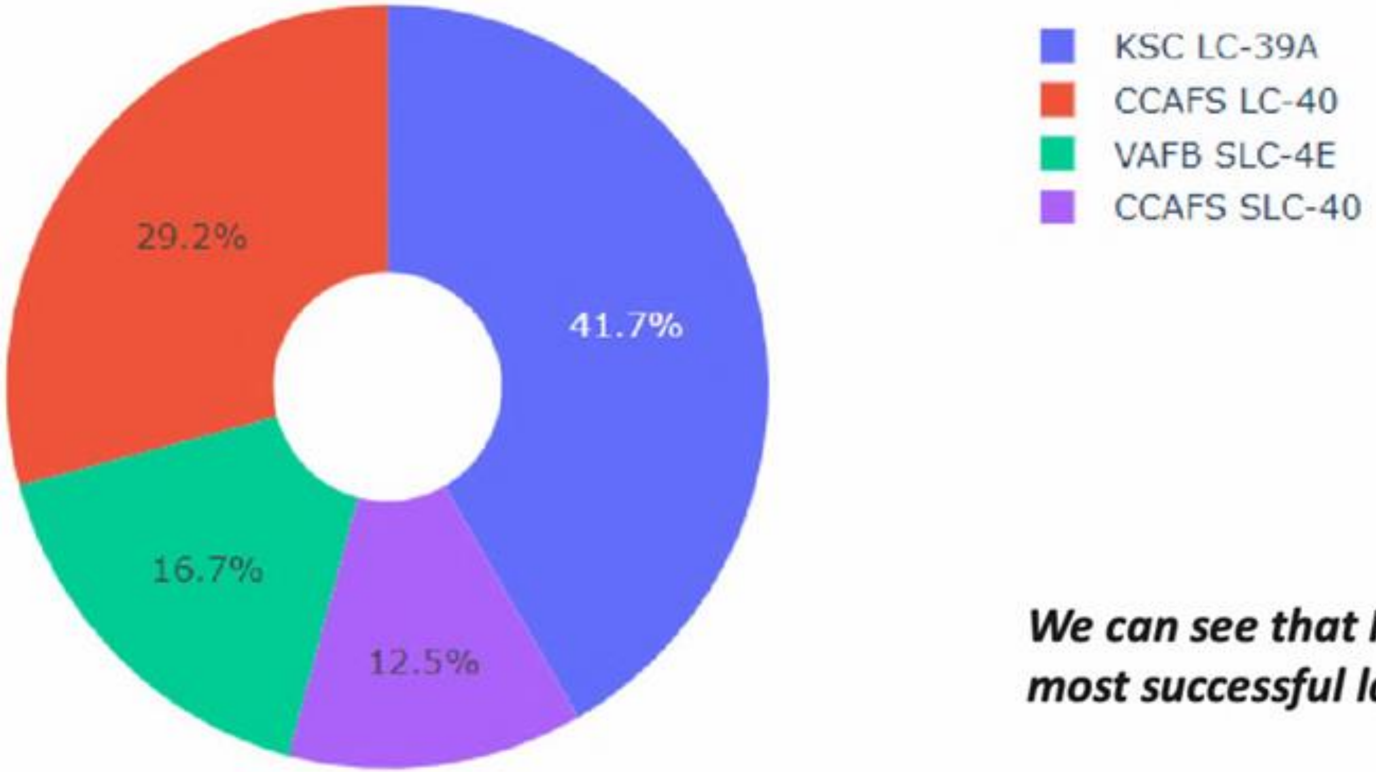
Analyzing Launch Site Proximity to Key Landmarks

- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

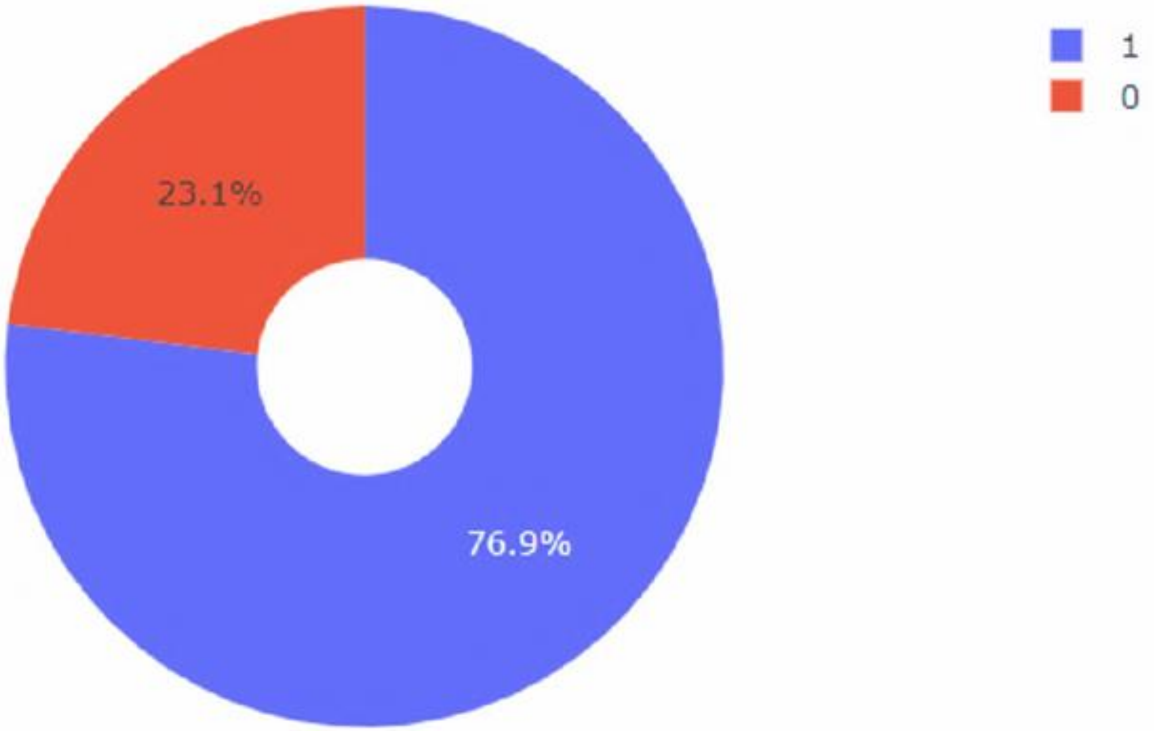
Build a Dashboard with Plotly Dash

Success Rate Analysis of Each Launch Site



We can see that KSC LC-39A had the most successful launches from all the sites

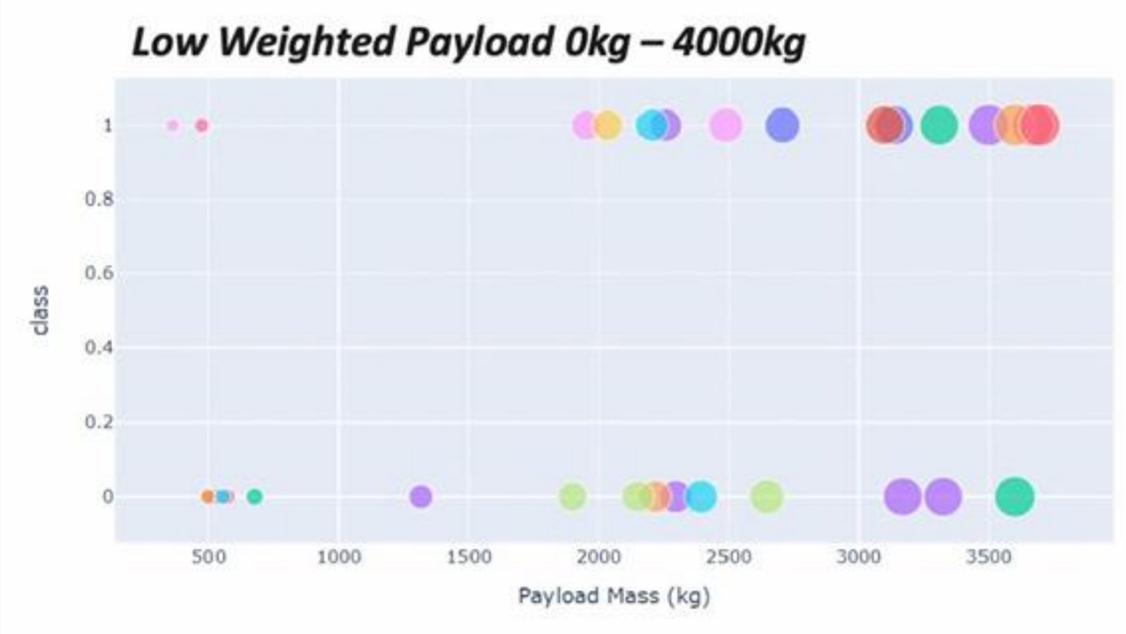
KSC LC-39A: Leading in Launch Success Ratio

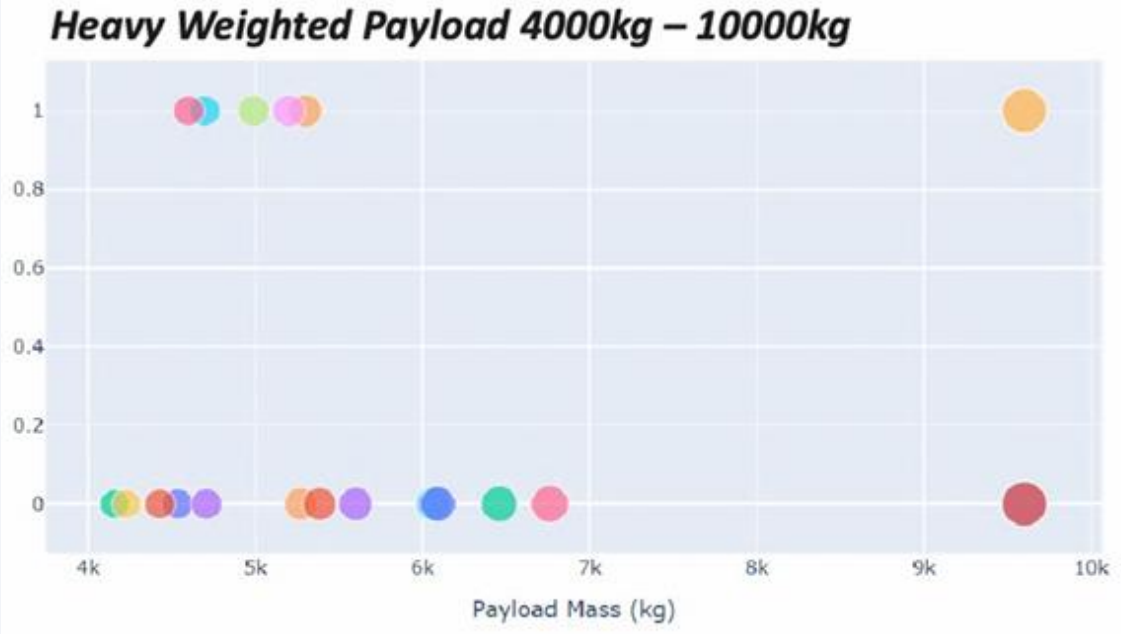


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter Plot Analysis: Payload Weight vs. Launch Outcome

- Impact of Payload Weight on Launch Success: Higher Success Rates for Lighter Payloads





Section 5

Predictive Analysis (Classification)

Classification Accuracy

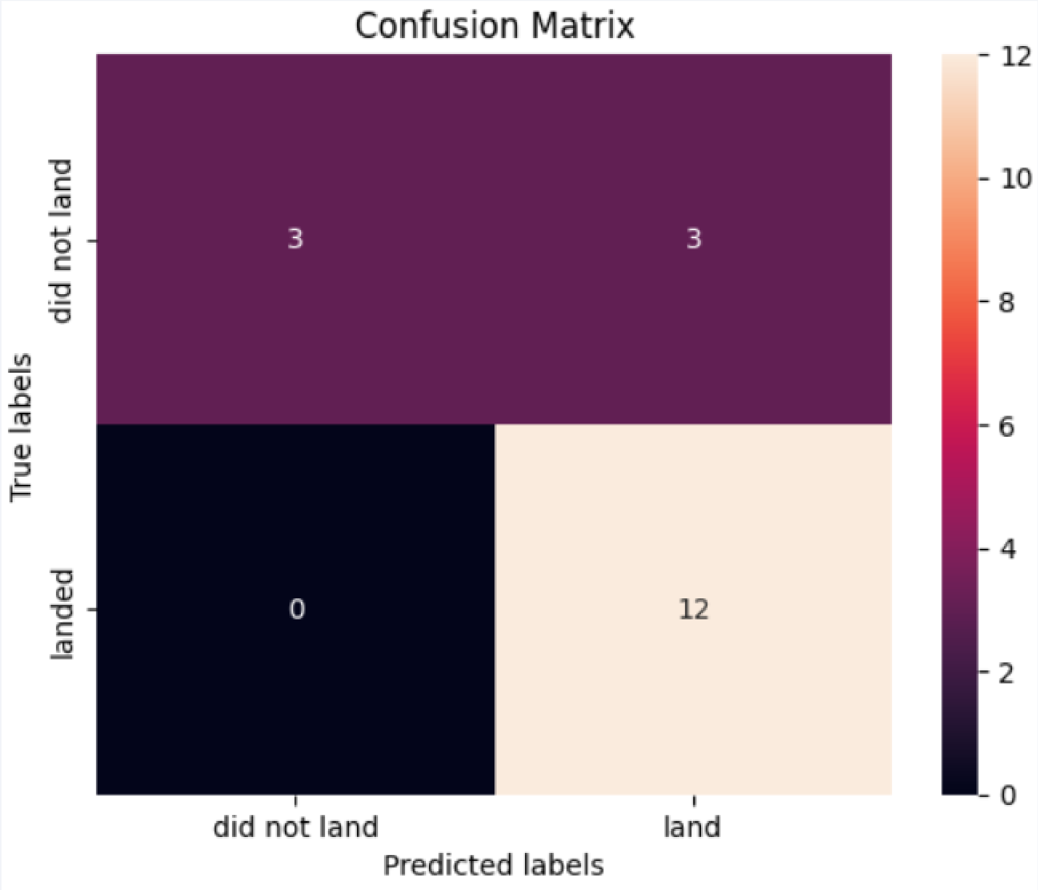
- As demonstrated in the following code, the **Tree Classifier Algorithm** was identified as the best-performing machine learning model, achieving the highest classification accuracy.

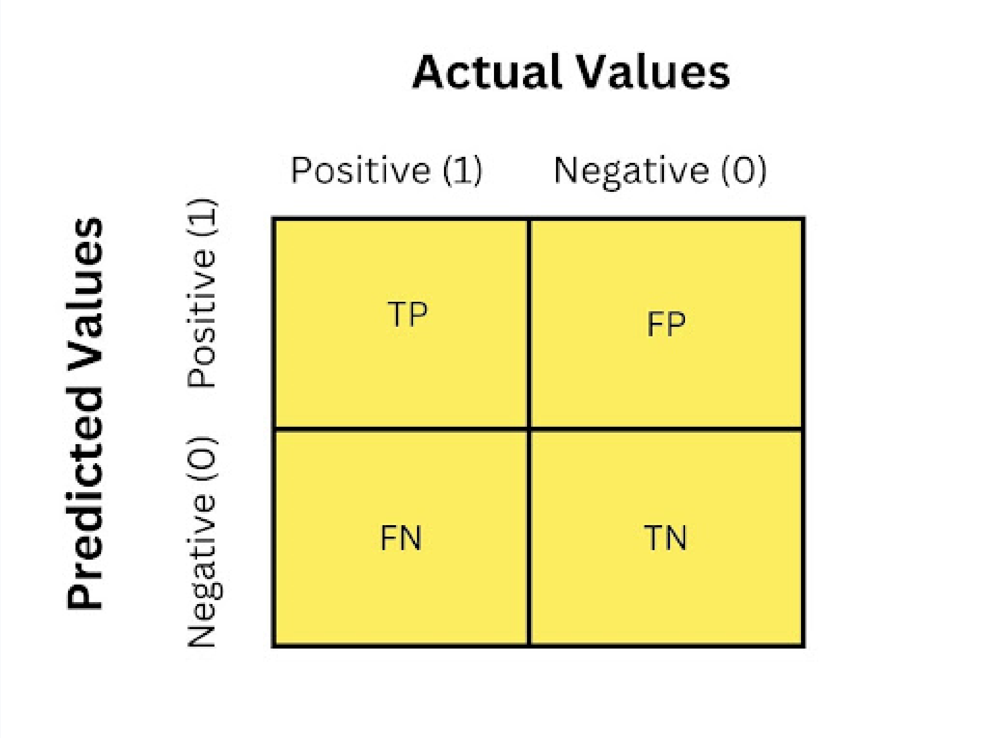
```
models_accuracies = {
    'Model': ['Logistic Regression', 'Support Vector Machine', 'Decision Tree', 'K-Nearest Neighbors'],
    'Test Accuracy': [
        logreg_cv.best_estimator_.score(X_test, Y_test),
        svm_cv.best_estimator_.score(X_test, Y_test),
        tree_cv.best_estimator_.score(X_test, Y_test),
        knn_cv.best_estimator_.score(X_test, Y_test)
    ]
}

accuracy_df = pd.DataFrame(models_accuracies)

print(accuracy_df)
print(accuracy_df['Test Accuracy'].idxmax())
print(accuracy_df.loc[])
```

```
                    Model  Test Accuracy
0     Logistic Regression       0.833333
1  Support Vector Machine       0.833333
2           Decision Tree       0.944444
3     K-Nearest Neighbors       0.833333
2
```

Confusion Matrix

- The confusion matrix for the **Decision Tree Classifier** illustrates that the model effectively distinguishes between the different classes. However, the major issue lies in the **false positives**, where unsuccessful landings are incorrectly predicted as successful landings by the classifier.





Conclusions

•**Tree Classifier Algorithm** proved to be the most effective Machine Learning approach for this dataset.

•**Low-weighted payloads (≤4000kg)** had a higher success rate compared to heavier payloads.

•**SpaceX's success rate** has shown a consistent upward trend since **2013**, indicating improvements over time, potentially leading to near-perfect launches in the future.

•**KSC LC-39A** recorded the highest success rate among all launch sites, with **76.9%** successful launches.

•**SSO orbit** had the highest success rate (**100%**) with multiple successful launches.

Appendix

- For additional resources, including Python code snippets, SQL queries, charts, notebook outputs, and datasets used in this project, visit the GitHub repository: https://github.com/OmarFawzyShehab/Applied-Data-Science-Capstone-SpaceX

Thank you!