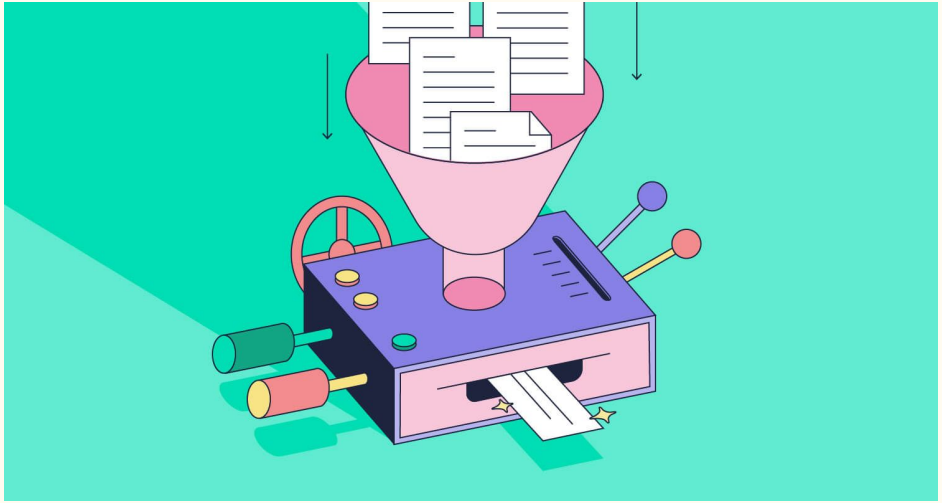
Precision Descent:

Using Data Science to Forecast Successful Rocket Landings

Christopher Wood
June 28th, 2024

Project Overview

Executive Summary



Summary of Methodologies:

- Collect data using SpaceX REST API and Webscraping
- Wrangle data to create outcome variable and put feature data into format suitable for analysis
- Explore data with visualization techniques, considering the features payload, launch site, flight number, and orbit
- Analyze the data with SQL, calculating useful statistics from the data
- Use Folium to find patterns in launch site success rates and distance to geographical markers
- Build models to predict landing outcomes using logistic regression, support vector machine (SVM), decision tree analysis, and K -nearest neighbor

Summary of Results:

- Launch success rates have improved over time
- Kennedy Space Center's LC-39A shows the highest success rate among landing sites
- Features like payload size were identified as potential factors influencing launch success
- Specific orbits such as ES-L1, GEO, HEO, and SSO achieved a 100% success rate
- Launch sites are all located near the equator and coastlines, likely contributing to higher success rates
- Multiple models were evaluated, with the decision tree model demonstrating slightly superior performance in predicting rocket landing success

Table of Contents

Overview - Slide 2

- Executive Summary
- Introduction

Methodology - Slide 6

- Summary
- Data Collection
- Data Wrangling
- Exploratory Data Analysis
- Interactive Map
- Interactive Dashboard
- Predictive Analysis
- Results

Results - Slide 17

- Summary
- Insights from Exploratory Data Analysis
- Insights from Mapping
- Insights from Interactive Dashboard
- Insights from Predictive Analysis

Conclusion - Slide 46

- Summary of Conclusions

Appendix - Slide 48

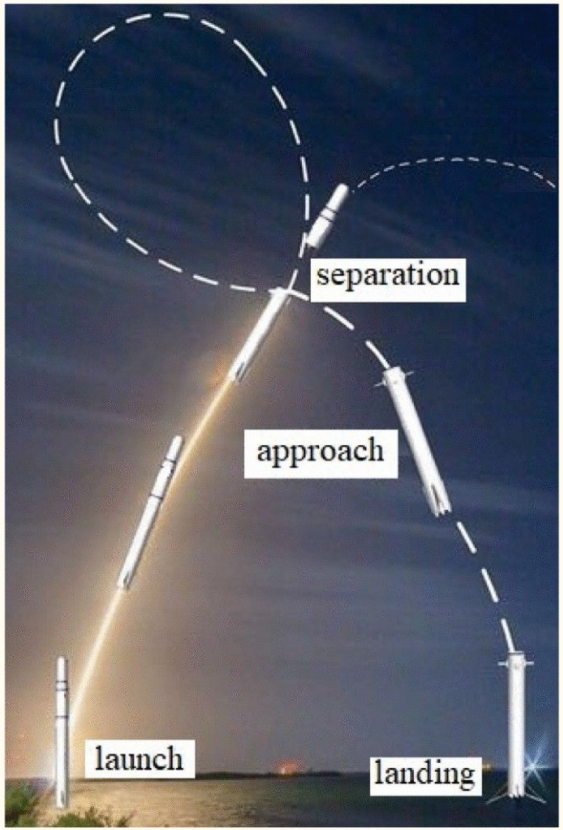
- Future Considerations

Introduction

Background: Private industry in space is a rapidly growing field, with SpaceX leading the charge. SpaceX's Mission is to build the next generation of rockets that are powerful, reliable, and affordable enough to make colonizing other planetary bodies possible. So far it has achieved sending a spacecraft to the international space station, launching a satellite internet constellation, and sending crewer missions into space.

This is all possible because the rocket launches are relatively inexpensive (less than have the cost of their competitors). The launches are so inexpensive because SpaceX reuses the first stage of its Falcon 9 rocket. We can use historical data and machine learning algorithms to predict whether the first stage will land for any future launches, and thus determine the price of launch.

Questions: How features of historical data affect the success of a landing (features to explore include payload mass, launch site, number of flights, and type of orbits). How the rate of successful landings has changed over time. Which predictive model for classification of successful landings best fits the data.



Methodology

Methodology - Executive Summary

In this project I aimed to predict the likelihood of a rocket landing successfully by employing a comprehensive methodology. I began with data collection, sourcing relevant datasets from the SpaceX API and webscraping data from the SpaceX Wikipedia. This initial step ensured that I had a rich set of data points to work with. Next, I performed data wrangling to clean and prepare the data, handling missing values, removing duplicates, and transforming the data into a suitable format for analysis. This step was crucial to ensure the integrity and accuracy of the subsequent analyses.

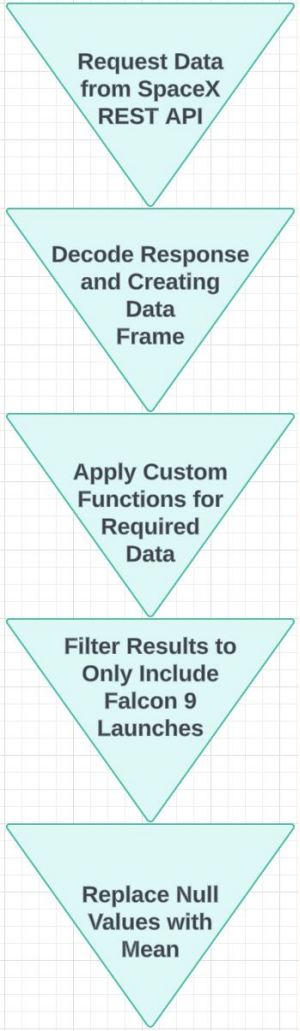
Next, I conducted exploratory data analysis (EDA) to uncover patterns and relationships within the data. I used data visualization techniques to create multiple graphs and charts, providing insights into trends and distributions. Additionally, I utilized Folium interactive maps to visualize geographical data, such as launch and landing sites, enhancing spatial understanding. To consolidate these visualizations I built a dashboard using Plotly, which allowed for dynamic data interaction.

Finally, I applied machine learning techniques for predictive analysis, focusing on classification algorithms to predict the success of rocket landings. This predictive model leveraged historical data to classify future landing attempts, providing a robust tool for forecasting and decision-making in space missions. All of the predictive models I created were evaluated to decide which best predicted the outcome of a launch.

Data Collection - SpaceX API

The data collection process began with requesting rocket launch data from the SpaceX REST API. The API responses were decoded using the .json() method and then converted into dataframes via .json_normalize(). Custom functions were employed to gather detailed launch information from the SpaceX API, which was then organized into dictionaries and converted into dataframes. The dataset was filtered to include only Falcon 9 launches, and missing values in the Payload Mass column were replaced with the calculated mean. The final dataset was exported to a CSV file for further analysis.

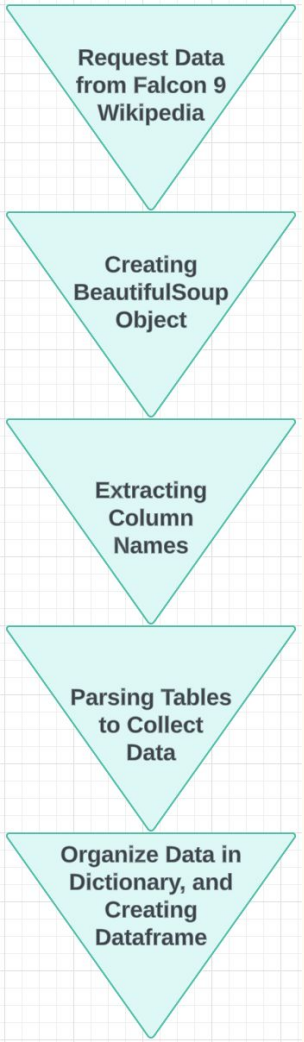
API Data Collection GitHub Link



Data Collection - Webscraping

To ensure comprehensive coverage, API requests from the SpaceX REST API was combined with web scraping data from a table in SpaceX's Wikipedia entry. This dual approach provided complete information about the launches, enhancing the analysis. Wikipedia web scraping involved requesting Falcon 9 launch data, creating a BeautifulSoup object from the HTML response, extracting column names from the HTML table header, and parsing HTML tables to collect the data. This data was then organized into dictionaries, converted into dataframes, and exported to a CSV file.

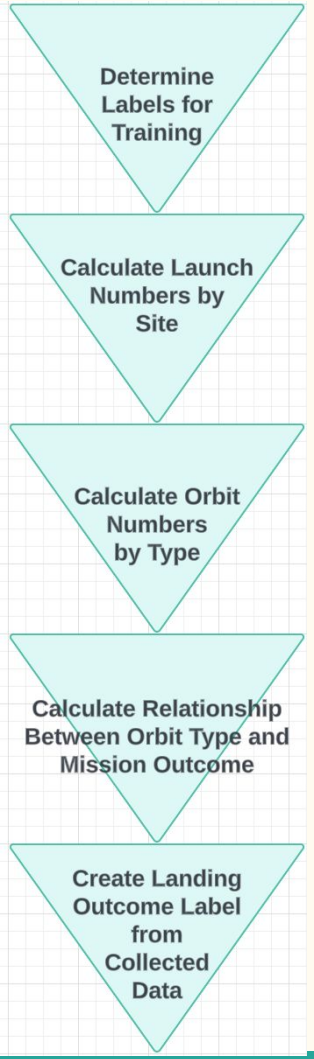
Webscraping Data Colletion GitHub Link



Data Wrangling

Exploratory data analysis was performed to determine the data labels. This involved calculating the number of launches for each site, the number of each orbit type occurrence, and the number of different mission outcomes per orbit type. A binary landing outcome column was created as the dependent variable. Landing outcomes were originally defined as follows: False Ocean for an unsuccessful landing in the ocean, True RTLS for a successful landing on a ground pad, False RTLS for an unsuccessful landing on a ground pad, True ASDS for a successful landing on a drone ship, False ASDS for an unsuccessful landing on a drone ship, and True Ocean for a successful landing in the ocean. These outcomes were converted into binary values, with 1 representing a successful landing and 0 representing an unsuccessful landing.

Data Wrangling GitHub Link



Exploratory Data Analysis - SQL

The queries involved in exploratory data analysis were:

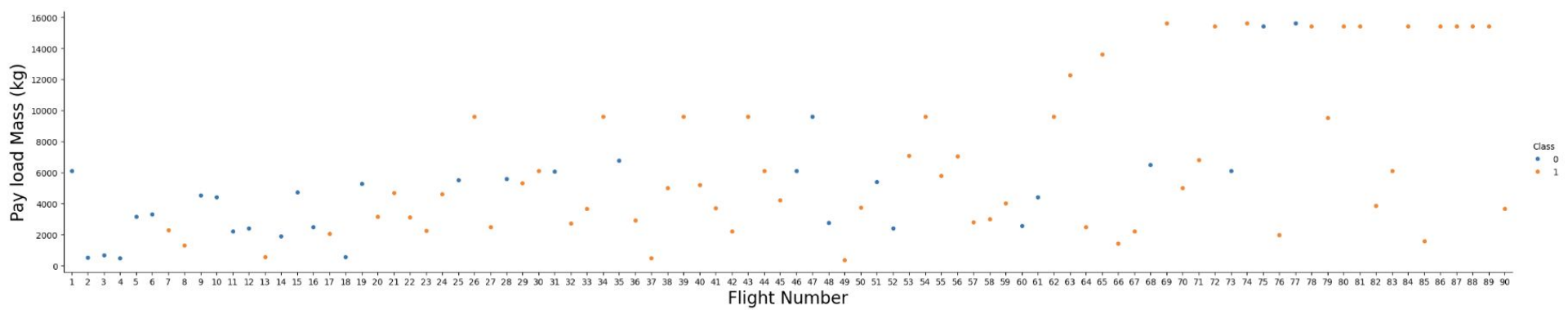
- Unique launch site names
- Records for each launch site
- Total payload mass carried for NASA
- Average payload mass for Falcon 9's current version
- Date of first successful ground landing
- Names of boosters with successful drone ship landing with payload mass between 4000kg and 6000kg
- Total number of successful and failed missions
- Names of boosters that have carried maximum payload
- Booster versions and launch sites for failed landings in 2015
- Landing outcome counts between June 2010 and March 2017

Exploratory Data Analysis with SQL GitHub Link

Exploratory Data Analysis - Visualizations

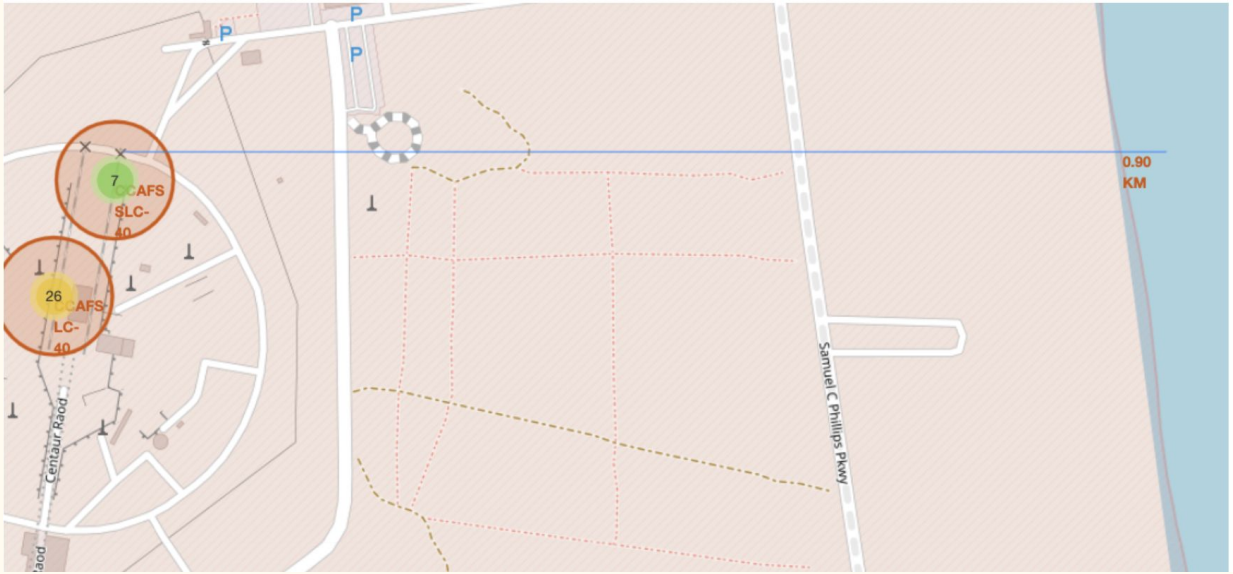
The analysis involved creating various charts, including Flight Number vs. Payload, Flight Number vs. Launch Site, Payload Mass (kg) vs. Launch Site, and Payload Mass (kg) vs. Orbit Type. Using exploratory data analysis (EDA) with visualization, scatter plots were employed to examine potential relationships between variables, which could be useful for machine learning. Bar charts were used to compare discrete categories, illustrating the relationships between these categories and measured values.

Exploratory Data Analysis with Visualization GitHub Link



Interactive Mapping

A map was created using Folium with various markers indicating launch sites. A blue circle was added at NASA Johnson Space Center's coordinates with a popup label showing its name, using its latitude and longitude coordinates. Red circles were added at all other launch site coordinates, also with popup labels showing their names. Colored markers were used to indicate launch outcomes, with green for successful launches and red for unsuccessful ones, highlighting the success rates at each site. Additionally, colored lines were added to show the distance between the launch site CCAFS SLC-40 and its proximity to the nearest coastline, railway, highway, and city, helping to determine the impact of geographic and manmade features on launch and landing success.



[Folium Interactive Mapping GitHub Link]

Interactive Dashboard

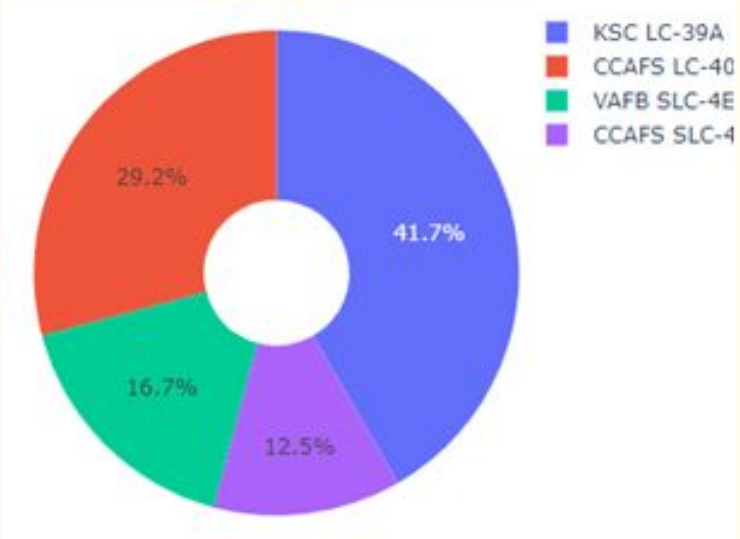
Created Launch Site Dropdown List
• Allows the user to choose a specific launch site or data from all launch sites

Created Slider for Range of Payload Mass
• Allows the user to choose a specific range of payload masses

Created Successful Launches Pie Chart
• Allows the user to see how successful and unsuccessful launches relate to total launches

Created Payload Mass and Success Rate Scatter Chart by Booster Version
• Allows the user to see the relationship Payload has with Launch Success

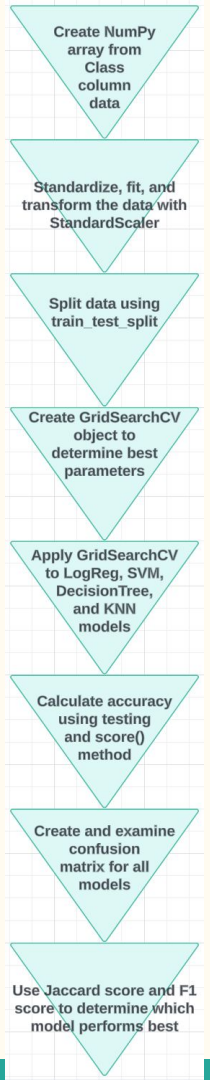
Plotly Dashboard GitHub Link



Predictive Analysis

The goal was to build, evaluate, and improve the classification model. First a NumPy array was created from the Class column. Then the data was standardized using StandardScaler, fitting and transforming it. Next, the data was split using train_test_split and a GridSearchCV object was created with cv=10 for parameter optimization. The GridSearchCV was applied to various algorithms, including logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree (DecisionTreeClassifier()), and K-Nearest Neighbor (KNeighborsClassifier()). For each model, the accuracy on the test data was calculated using the .score() method and assessed the confusion matrix. The best model was identified using Jaccard_Score, F1_Score, and Accuracy.

Predictive Analysis GitHub Link



Results

Results Summary

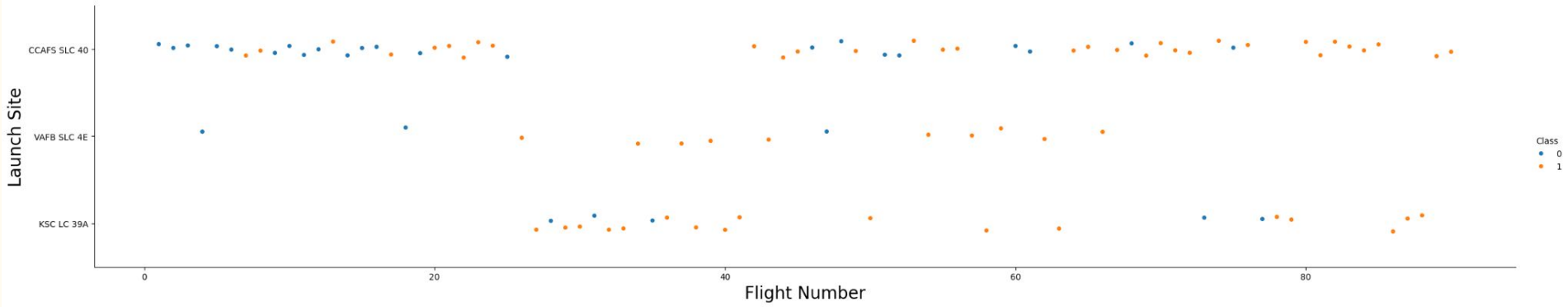
During the exploratory data analysis phase, I discovered that launch success rates have improved over time, with the Kennedy Space Center's LC-39A emerging as the site with the highest success rate among landing locations. There were several features in the data that could affect the success of a launch, for example, payload size. Additionally, specific orbits, including ES-L1, GEO, HEO, and SSO, have achieved a perfect 100% success rate. Through data visualization and analytics, I noted that most launch sites are situated near the equator and close to coastlines, which likely aids in successful launches and landings. In the predictive analytics phase, various models were tested, with the decision tree model slightly outperforming others, demonstrating its effectiveness in predicting rocket landing success.

Insights from Exploratory Data Analysis

Exploratory Data Analysis

Flight Number vs. Launch Site

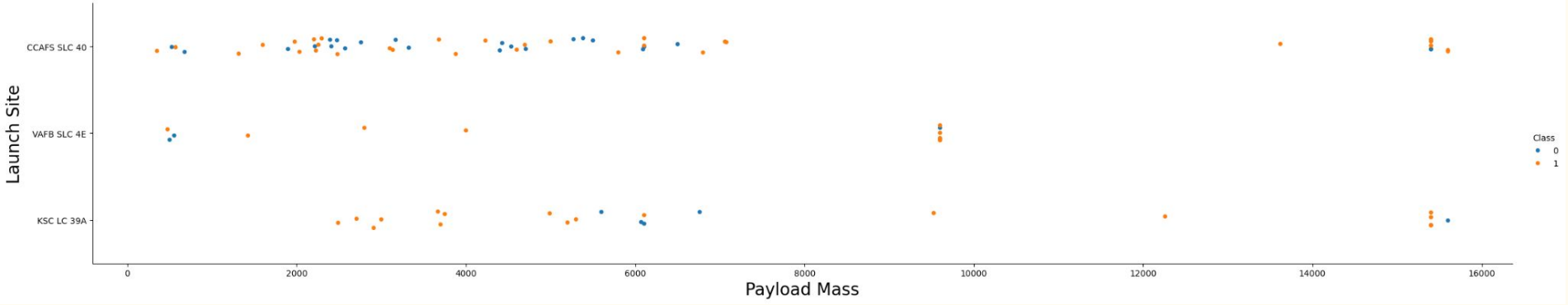
- Flights had a lower success rate in the beginning, and a higher success rate as flight number increases. (orange = success, blue = fail)
- Around half of all launches were from CCAFS SLC-40
- VAFB SLC-4E and KSC LC-39A have higher success rates



Exploratory Data Analysis

Payload Mass vs. Launch Site

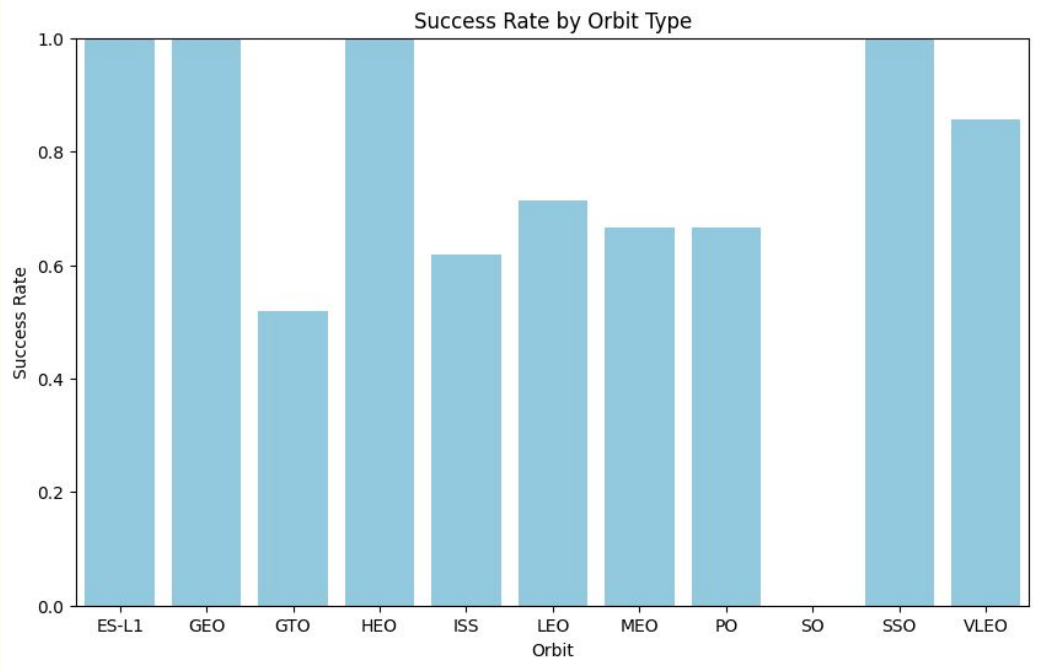
- The larger the payload mass, the better the success rate
- Launches with a payload greater than 7,000 kg were most often a success
- KSC LC-39A has a 100% success rate for launches below 5,500 kg
- VAFB SKC-4E has not launched anything larger than 10,000 kg



Exploratory Data Analysis

Success Rate vs. Orbit Type

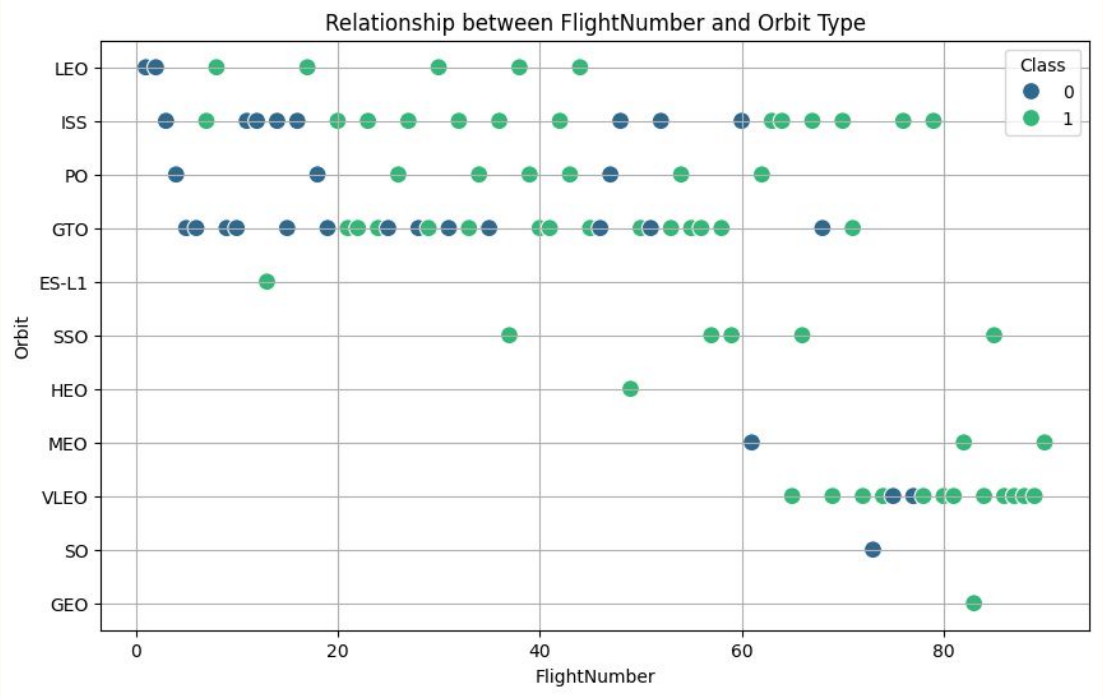
- ES-L1, GEO, HEO, and SSO have a 100% success rate
- GTO, ISS, LEO, MEO, and PO have a success rate between 50% and 80%
- SO has a success rate of 0%



Exploratory Data Analysis

Flight Number vs. Orbit Type

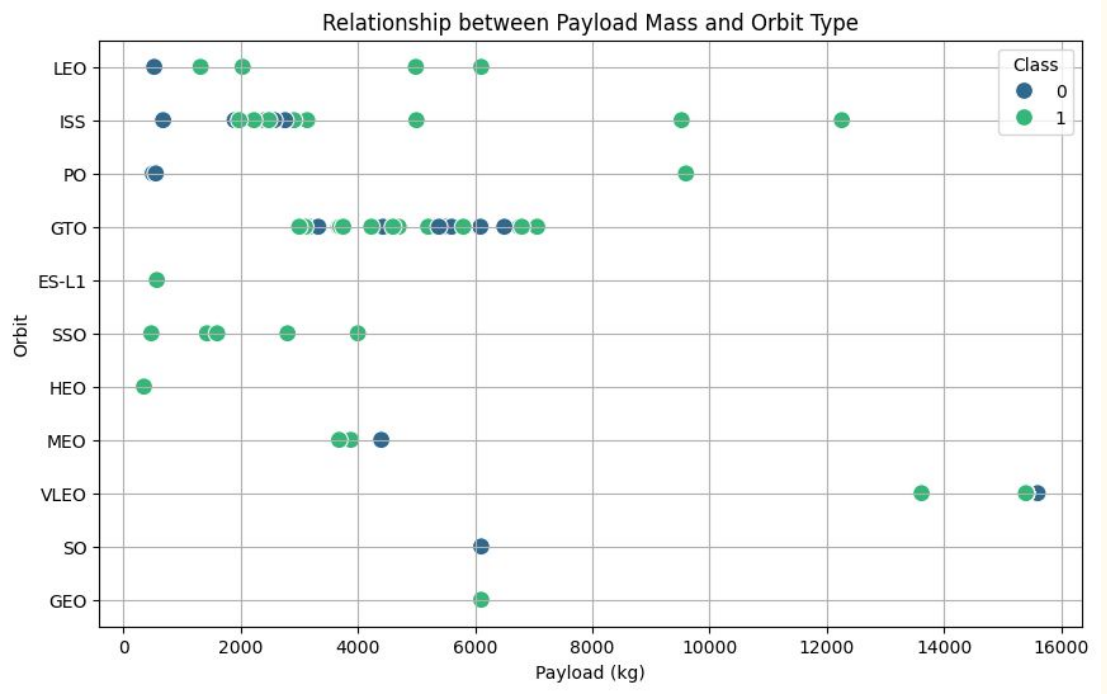
- Success rate generally increases with the number of flights for each orbit
- The above relationship is strongest for the LEO orbit
- The above relationship does not tend to hold true for the GTO orbit
- The ES-L1, HEO, SO, and GEO orbits have only been attempted once each



Exploratory Data Analysis

Payload Mass vs. Orbit Type

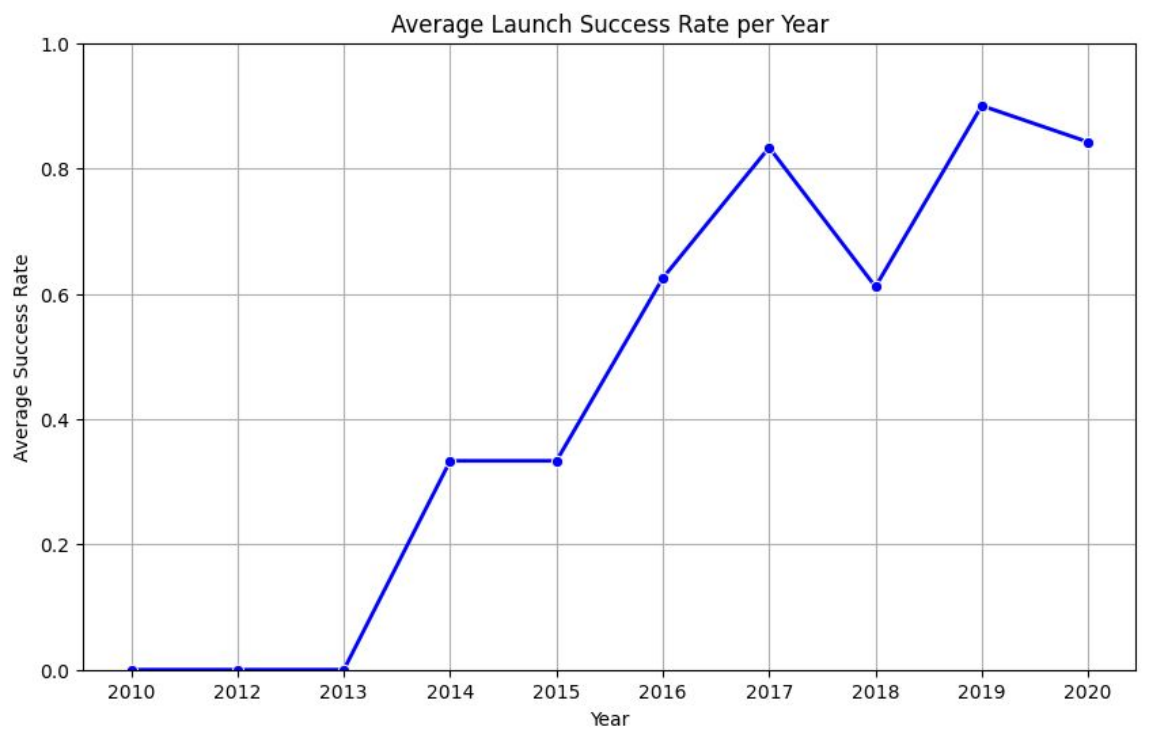
- Heavier payloads have a better success rate with LEO, ISS, and PO orbits
- The GTO orbit has mixed success with all payload masses
- The SSO orbit has success with all payload masses attempted



Exploratory Data Analysis

Launch Success Yearly Trend

- Success rate improved between 2013 and 2017, as well as between 2018 and 2019
- Success rate decreased between 2017 and 2018 and between 2019 and 2020
- Overall, the success rate has increased since 2013



Exploratory Data Analysis

All Launch Site Names

Task: Query data for all unique launch site names

Code:

```
# Execute the SQL query
query = 'SELECT DISTINCT "Launch_Site" FROM "SPACEXTBL";'
cur.execute(query)

# Fetch and print the results
launch_sites = cur.fetchall()
for site in launch_sites:
    print(site)
```

Results:

```
('CCAFS LC-40',)
('VAFB SLC-4E',)
('KSC LC-39A',)
('CCAFS SLC-40',)
```

Exploratory Data Analysis

Launch Site Names Beginning with “CCA”

Task: Query five results with the launch site name beginning with CCA

Code:

```
# Define the SQL query
query = '''
SELECT *
FROM "SPACEXTBL"
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the results
records = cur.fetchall()
for record in records:
    print(record)
```

Results:

```
('2010-06-04', '18:45:00', 'F9 v1.0  B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'LEO', 'Spa
ceX', 'Success', 'Failure (parachute)')
('2010-12-08', '15:43:00', 'F9 v1.0  B0004', 'CCAFS LC-40', 'Dragon demo flight C1, two CubeSats, barrel of Brouer
e 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 (COT
S)', 'Success', 'No attempt')
('2012-10-08', '0:35:00', 'F9 v1.0  B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500, 'LEO (ISS)', 'NASA (CRS)', 'Succes
s', 'No attempt')
('2013-03-01', '15:10:00', 'F9 v1.0  B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677, 'LEO (ISS)', 'NASA (CRS)', 'Succe
ss', 'No attempt')
```

Exploratory Data Analysis

Total Payload Mass

Task: Display total payload mass by boosters launched by NASA

Code:

```
# Define the SQL query
query = '''
SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass
FROM "SPACEXTBL"
WHERE "Customer" = 'NASA (CRS)';
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the result
total_payload_mass = cur.fetchone()[0]
print("Total Payload Mass carried by boosters launched by NASA (CRS):", total_payload_mass)
```

Results:

```
Total Payload Mass carried by boosters launched by NASA (CRS): 45596
```

Exploratory Data Analysis

Average Payload Mass for Falcon 9 Version 1.1

Task: Display the average payload mass for the Falcon 9 v1.1 booster.

Code:

```
# Define the SQL query
query = '''
SELECT AVG("PAYLOAD_MASS__KG_") AS average_payload_mass
FROM "SPACEXTBL"
WHERE "Booster_Version" = 'F9 v1.1';
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the result
average_payload_mass = cur.fetchone()[0]
print("Average Payload Mass carried by booster version F9 v1.1:", average_payload_mass)
```

Results:

```
Average Payload Mass carried by booster version F9 v1.1: 2928.4
```

Exploratory Data Analysis

First Successful Ground Landing Date

Task: Display the first date of a successful ground landing.

Code:

```
# Define the SQL query
query = '''
SELECT MIN("Date") AS first_successful_landing_date
FROM "SPACEXTBL"
WHERE "Landing_Outcome" = 'Success (ground pad)';
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the result
first_successful_landing_date = cur.fetchone()[0]
print("Date of the first successful landing outcome in ground pad:", first_successful_landing_date)
```

Results:

```
Date of the first successful landing outcome in ground pad: 2015-12-22
```

Exploratory Data Analysis

Successful Drone Ship Landings with Payload Mass Between 4,000kg and 6,000kg

Task: Create a list of boosters that have successfully completed a drone ship landing with a payload mass between 4,000kg and 6,000kg

Code:

```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING_OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

Results:

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Exploratory Data Analysis

Total Number of Success and Failure Mission Outcomes

Task: Display the total number of mission outcomes classified as successes or failures.

Code:

```
# Define the SQL query
query = '''
SELECT "Mission_Outcome", COUNT(*) AS mission_outcome_count
FROM "SPACEXTBL"
GROUP BY "Mission_Outcome";
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the results
results = cur.fetchall()
for row in results:
    print(f"Mission Outcome: {row[0]}, Count: {row[1]}")
```

Results:

```
Mission Outcome: Failure (in flight), Count: 1
Mission Outcome: Success, Count: 98
Mission Outcome: Success , Count: 1
Mission Outcome: Success (payload status unclear), Count: 1
```

Exploratory Data Analysis

Boosters that have Carried the Maximum Payload Mass

Task: Create a list of names of boosters that have carried the maximum payload mass.

Code:

```
query = '''
SELECT "Booster_Version"
FROM "SPACEXTBL"
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM "SPACEXTBL"
);
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the results
results = cur.fetchall()
for row in results:
    print("Booster Version with maximum payload mass:", row[0])
```

Results:

```
Booster Version with maximum payload mass: F9 B5 B1048.4
Booster Version with maximum payload mass: F9 B5 B1049.4
Booster Version with maximum payload mass: F9 B5 B1051.3
Booster Version with maximum payload mass: F9 B5 B1056.4
Booster Version with maximum payload mass: F9 B5 B1048.5
Booster Version with maximum payload mass: F9 B5 B1051.4
Booster Version with maximum payload mass: F9 B5 B1049.5
Booster Version with maximum payload mass: F9 B5 B1060.2
Booster Version with maximum payload mass: F9 B5 B1058.3
Booster Version with maximum payload mass: F9 B5 B1051.6
Booster Version with maximum payload mass: F9 B5 B1060.3
Booster Version with maximum payload mass: F9 B5 B1049.7
```

Exploratory Data Analysis

Launch Records from 2015

Task: List the records from 2015 that failed while attempting a landing on a drone ship, displaying month, anding outcome, booster version, and launch site.

Code:

```
# Define the SQL query
query = '''
SELECT
    CASE substr("Date", 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS Month_Name,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM "SPACEXTBL"
WHERE substr("Date", 0, 5) = '2015'
    AND "Landing_Outcome" LIKE '%Failure (drone ship)%';
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the results
results = cur.fetchall()
for row in results:
    print("Month:", row[0])
    print("Landing Outcome:", row[1])
    print("Booster Version:", row[2])
    print("Launch Site:", row[3])
    print("------------------------")
```

Results:

```
Month: January
Landing Outcome: Failure (drone ship)
Booster Version: F9 v1.1 B1012
Launch Site: CCAFS LC-40
------------------------
Month: April
Landing Outcome: Failure (drone ship)
Booster Version: F9 v1.1 B1015
Launch Site: CCAFS LC-40
------------------------
```

Exploratory Data Analysis

Count of all Landing Outcomes

Task:

Code:

```
# Define the SQL query with window function for ranking
query = '''
SELECT
    "Landing_Outcome",
    COUNT(*) AS outcome_count,
    ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS outcome_rank
FROM "SPACEXTBL"
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY outcome_count DESC;
'''

# Execute the SQL query
cur.execute(query)

# Fetch and print the results
results = cur.fetchall()
for row in results:
    print("Landing Outcome:", row[0])
    print("Outcome Count:", row[1])
    print("Outcome Rank:", row[2])
    print("------------------------")
```

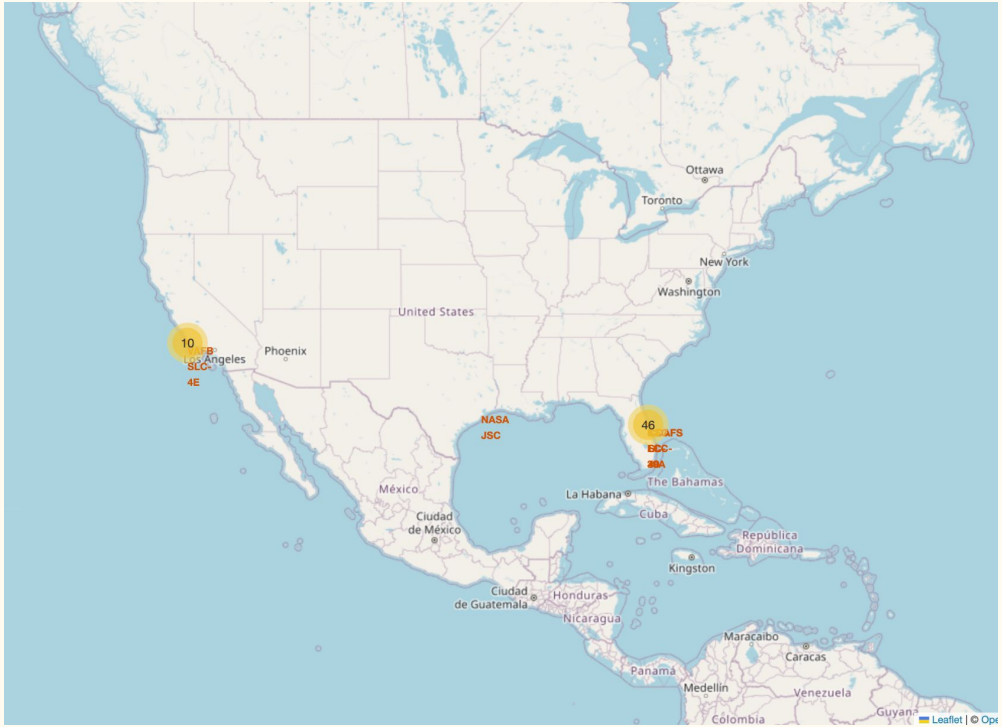
Results:

```
Landing Outcome: No attempt
Outcome Count: 10
Outcome Rank: 1
------------------------
Landing Outcome: Success (drone ship)
Outcome Count: 5
Outcome Rank: 2
------------------------
Landing Outcome: Failure (drone ship)
Outcome Count: 5
Outcome Rank: 3
------------------------
Landing Outcome: Success (ground pad)
Outcome Count: 3
Outcome Rank: 4
------------------------
Landing Outcome: Controlled (ocean)
Outcome Count: 3
Outcome Rank: 5
------------------------
Landing Outcome: Uncontrolled (ocean)
Outcome Count: 2
Outcome Rank: 6
------------------------
Landing Outcome: Failure (parachute)
Outcome Count: 2
Outcome Rank: 7
------------------------
Landing Outcome: Precluded (drone ship)
Outcome Count: 1
Outcome Rank: 8
------------------------
```

Insights from Mapping with Folium

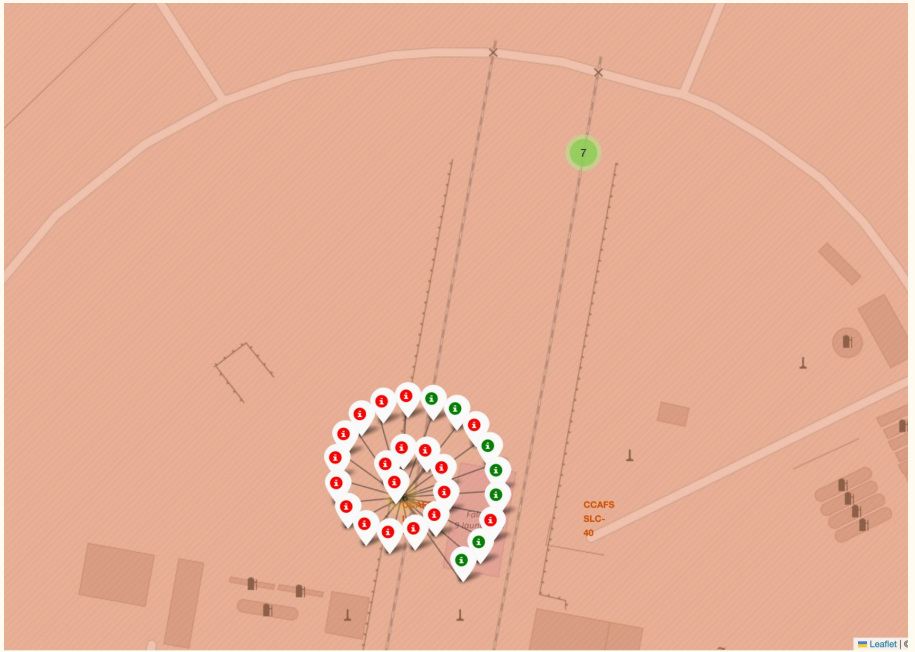
Mapping with Folium

Most of the launch sites are near to the equator. The closer a launch site is to the equator, the easier it is to launch into orbit. This helps lower the cost of a launch.



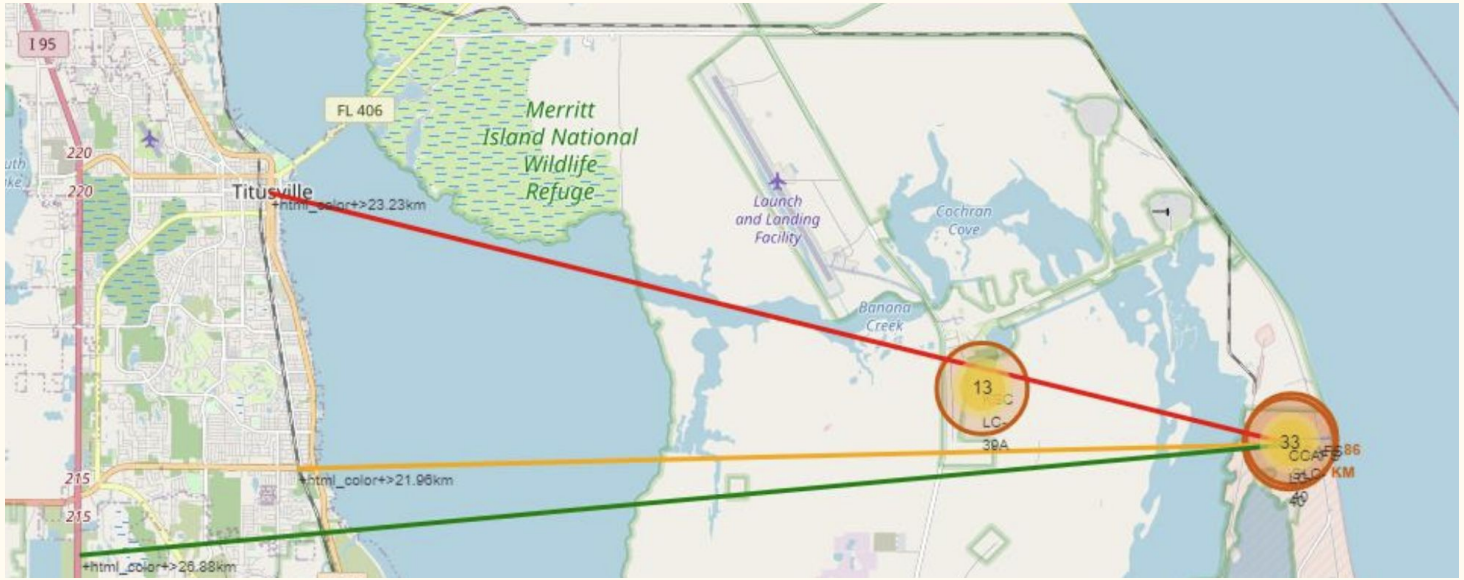
Mapping with Folium

Green circles represent successful launches. Red circles represent failed launches.



Mapping with Folium

Launch sites tend to be near coastlines, railways, cities, and highways. This helps with the logistics of delivering the boosters to the launch pads and landing/collecting boosters for reuse.

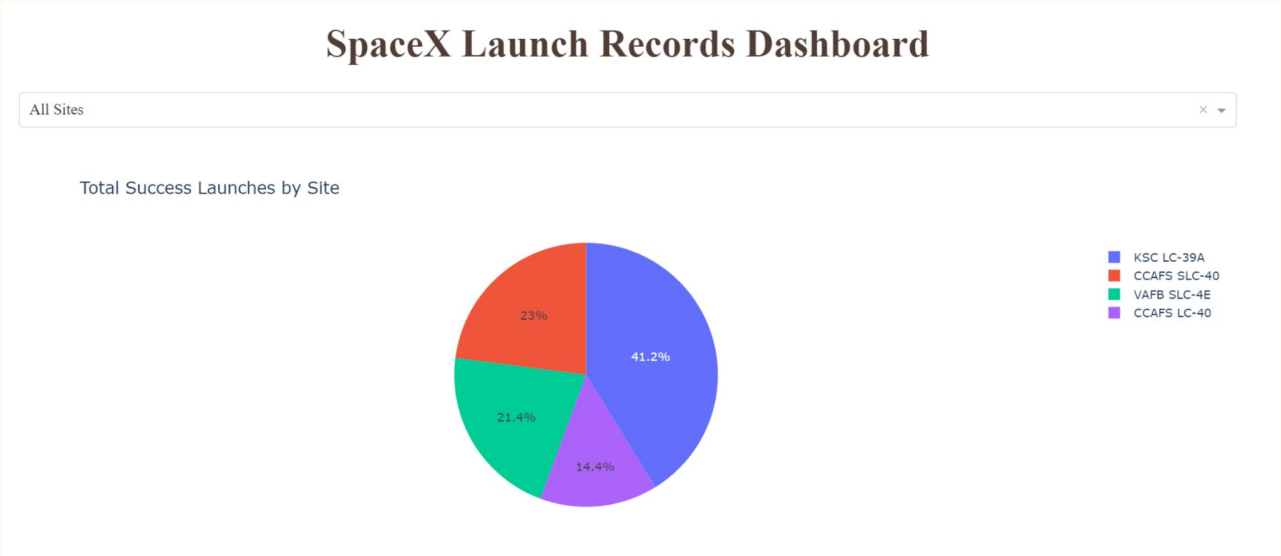


Insights from Interactive Dashboard

Interactive Dashboard

Success Rate of All Launch Sites

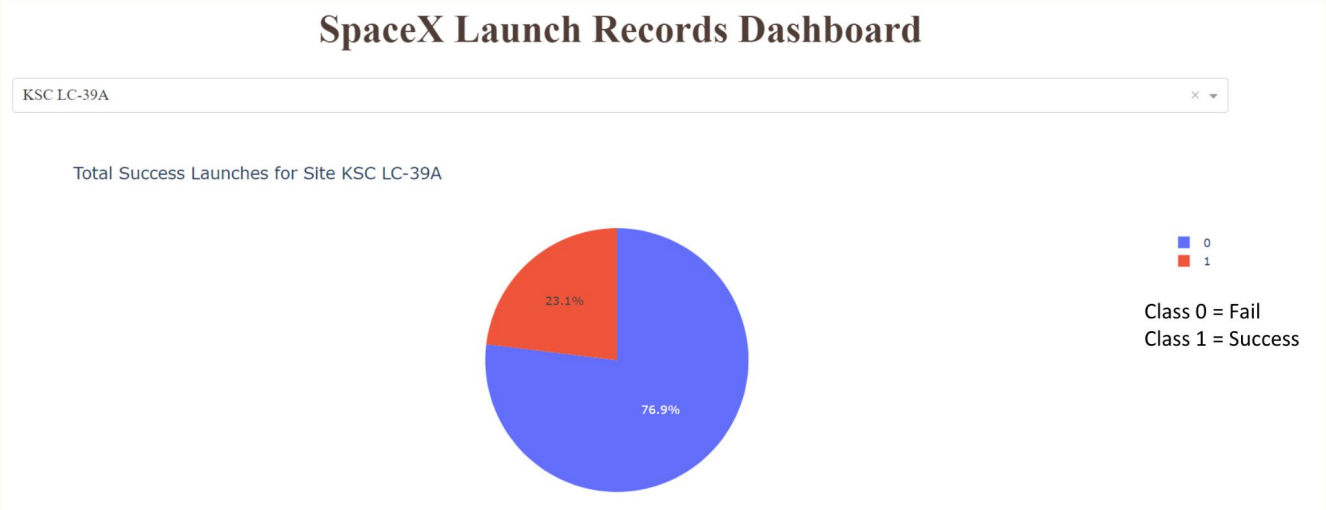
KSC LC-39A accounts for 41.2% of all successful launches, which is the highest of all sites.



Interactive Dashboard

Success Rate of KSC LC-39A

KSC LC-39A has a successful launch rate of 76.9%, which is the highest of all launch sites.



Interactive Dashboard

Payload vs. Launch Outcome Slider

Payloads between 2,000kg and 4,000kg, and 4,500kg and 5,500kg show the highest success rate of any payload size.

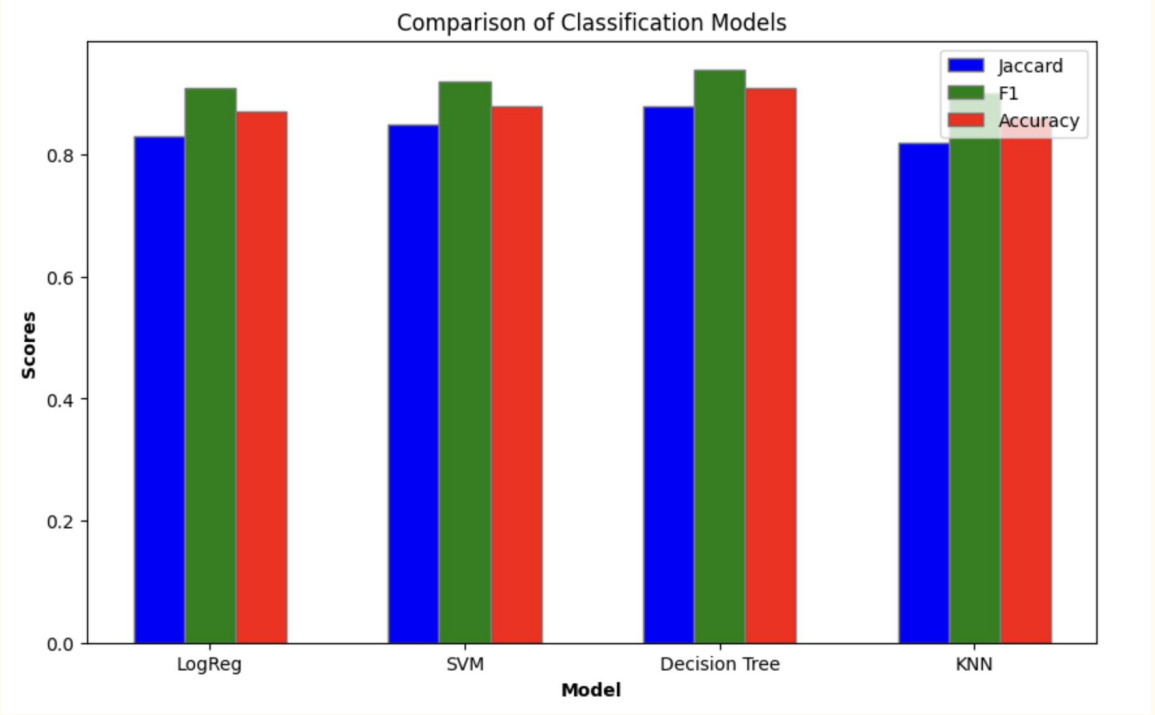
Insights from Predictive Analysis

Predictive Analysis

Classification Accuracy

LogReg, SVM, Decision Tree, and KNN performed very similarly. KNN was close overall but with slightly lower scores, while Decision Tree was close overall but with slightly higher scores.

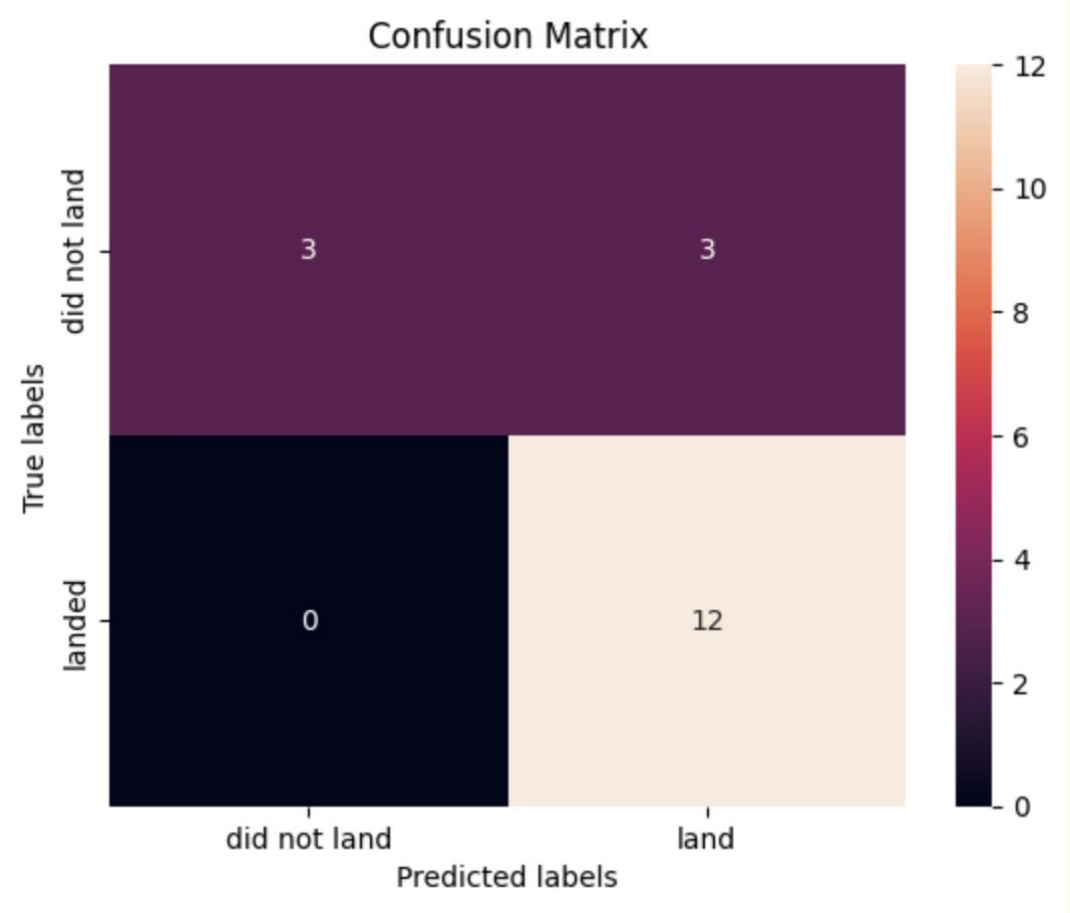
The similarity in performance is likely due to the small sample size of the data.



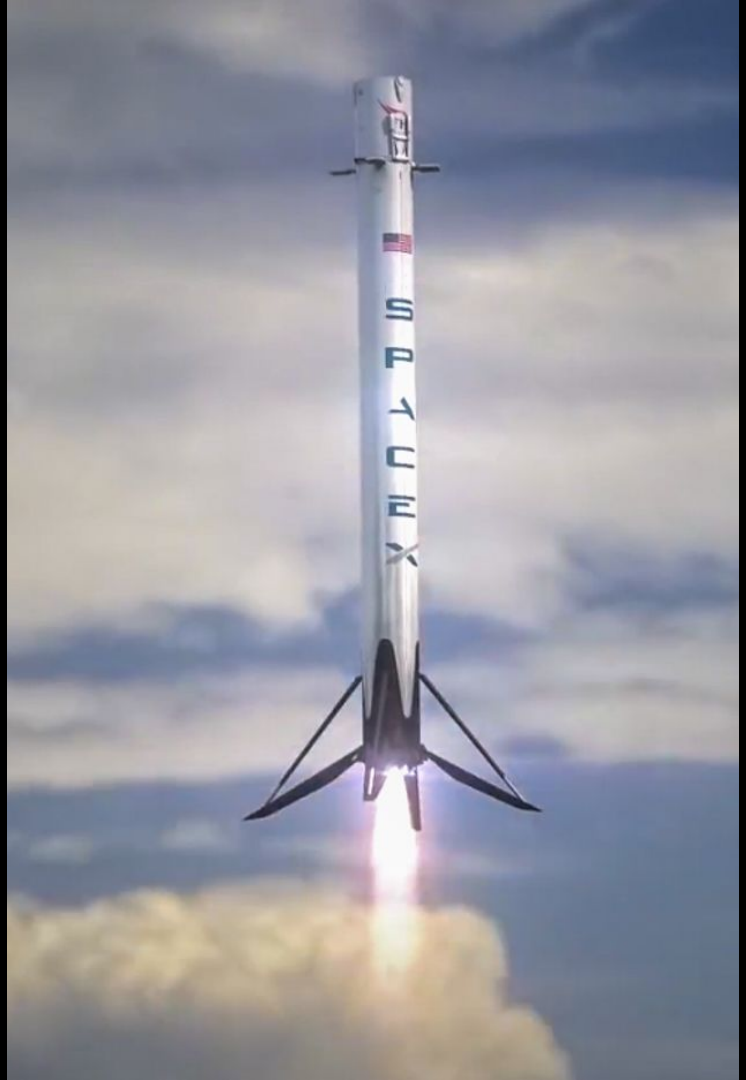
Predictive Analysis

Confusion Matrix

The Confusion Matrix for Decision Tree can be seen here. While the machine learning models can classify the landed category correctly, there is a risk for false positives even with the best performing model.



Conclusion



Summary of Conclusions

• Model Performance: Decision Tree slightly outperformed the other models, while KNN slight underperformed the other models. There is still a risk of false positives with our best model

• Launch Site Determination:

- Near Equator: Helps save the cost of extra fuel and boosters
- Near Coast: Aid in retrieval for ocean or drone ship landing, safety for aborted launches
- Near Infrastructure: Roads, railroads, and cities provide necessary means to transport/manufacture boosters and staff launch sites adequately
- KSC LC-39A: Has the highest success rate among launch sites

• Orbits: ES-L1, GEO, HEO, and SSO have 100% success rate

• Payload Mass: Larger payload masses have a higher success rate

• Overall Launch Success: Has increased over time

Appendix

Future Considerations

- Larger data set required to refine model and improve performance, especially to eliminate false positives