IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

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
  - Plotly Dash dashboard
  - 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 space exploration with its focus on reusable rockets and cost-effective space missions. Analyzing historical mission data provides insights into the factors influencing success rates, payload efficiency, and landing outcomes. This project explores SpaceX's performance, focusing on key aspects like launch sites, booster versions, payloads, and mission outcomes over time.

In this presentation we will be answering the questions

- Which launch sites and booster versions have the highest success rates?
- How do payload mass and orbit type affect mission success?
- What factors contribute to failures in drone ship and ground pad landings?
- Which machine learning model best predicts landing success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - We collected using SpaceX REST API and by doing web scrapping from Wikipedia
- Perform data wrangling
    - We processed the data using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - How to build, tune, evaluate classification models

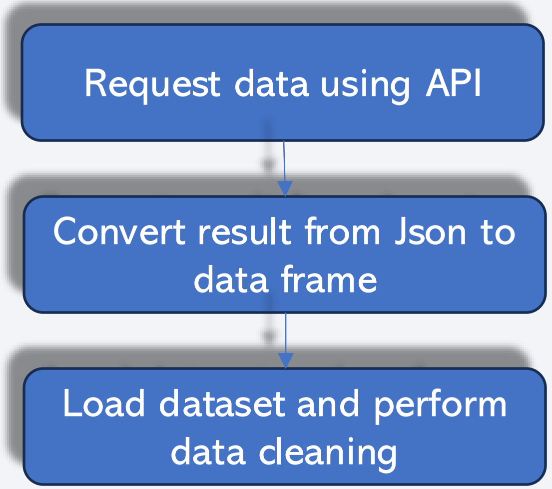
Data Collection

Data collection involves gathering and measuring information on targeted variables to answer relevant questions and evaluate outcomes.

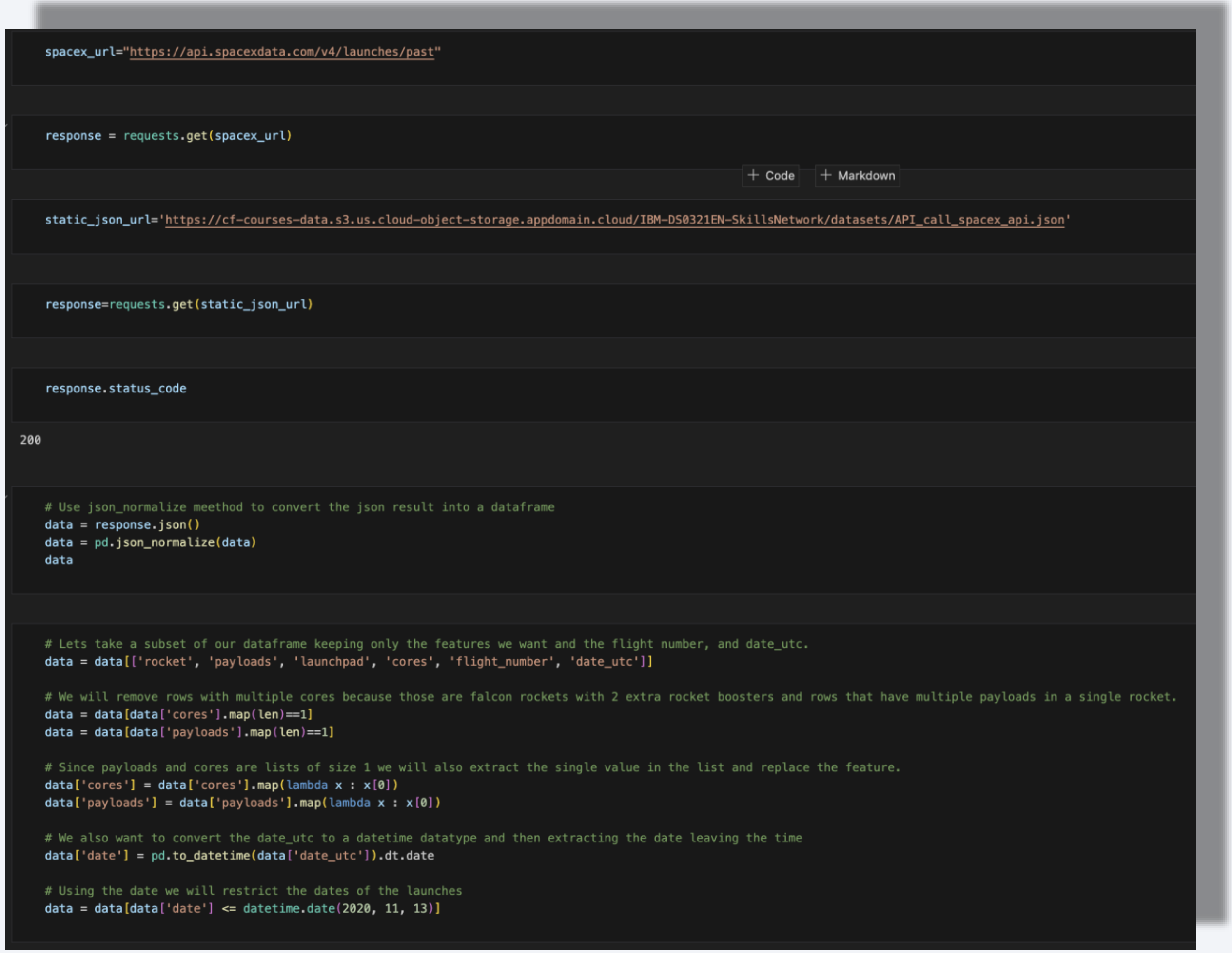
Sources:

- REST API:
  - Data was retrieved using GET requests.
  - The response was decoded as JSON and converted into a Pandas DataFrame using json_normalize().
  - The dataset was cleaned, missing values were identified, and necessary adjustments were made.
- Web Scraping:
  - BeautifulSoup was used to extract launch records from HTML tables on Wikipedia.
  - Parsed data was converted into a Pandas DataFrame for analysis.
- Outcome:
  - A clean, structured dataset was created, ready for analysis to address the project's key objectives.

Data Collection – SpaceX API



Source:
https://github.com/ROM117/Final-IBM-Applied-Data-Science-Capstone/blob/main/1.1_jupyter-labs-spacex-data-collection-api.ipynb



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

response = requests.get(spacex_url)

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

response=requests.get(static_json_url)

response.status_code
```

```
200
```

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

```python
# 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

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

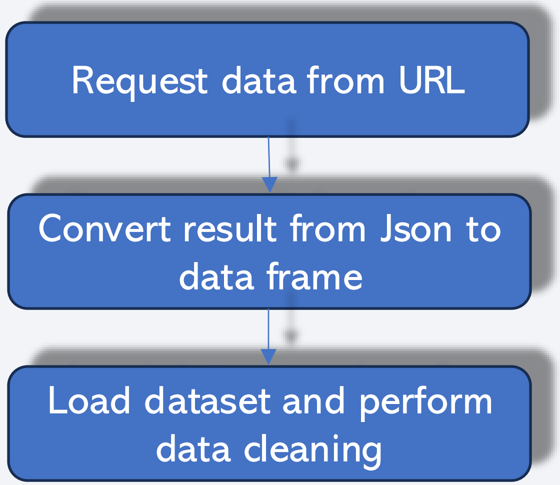
```python
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

```python
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
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
            time = datatimelist[1]
            #print(time)
```

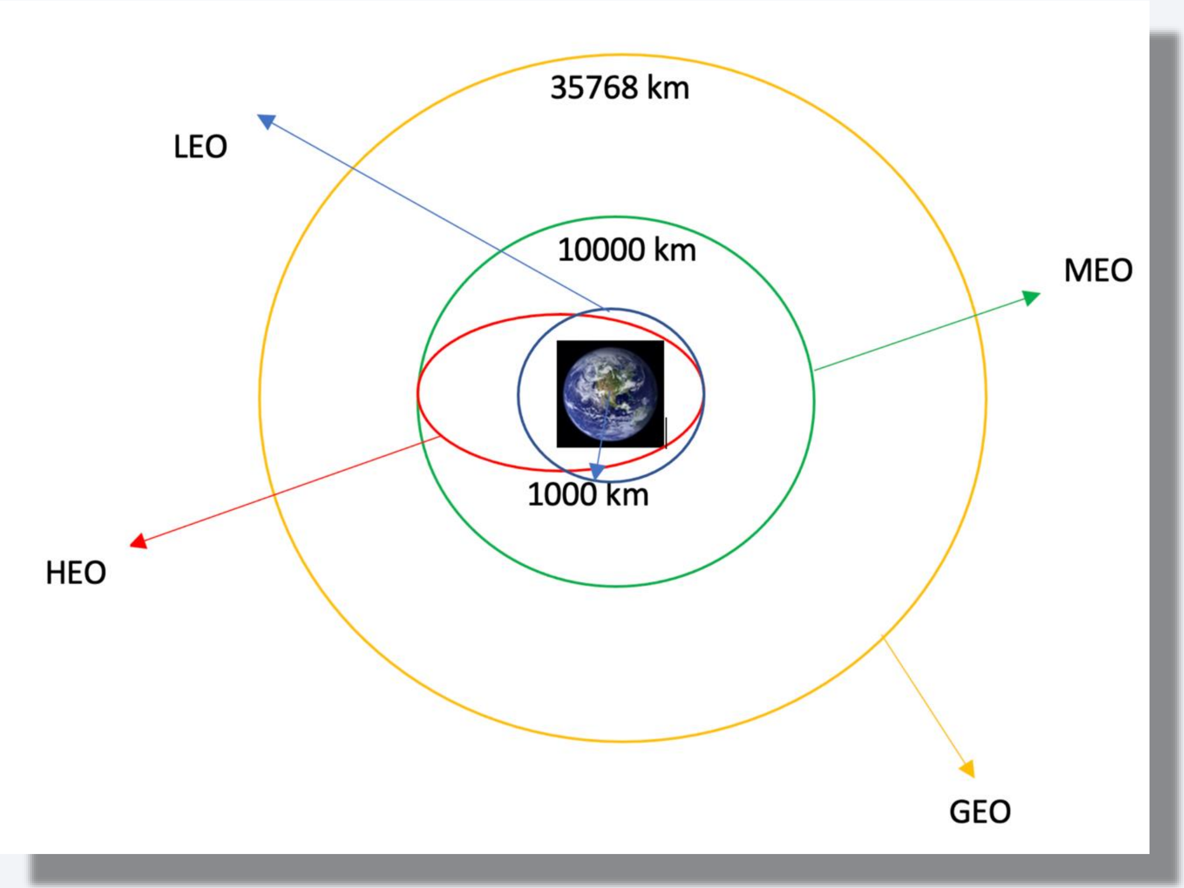


Source:
https://github.com/R0M117/Final-IBM-Applied-Data-Science-Capstone/blob/main/1.2_jupyter-labs-webscraping.ipynb

Data Wrangling

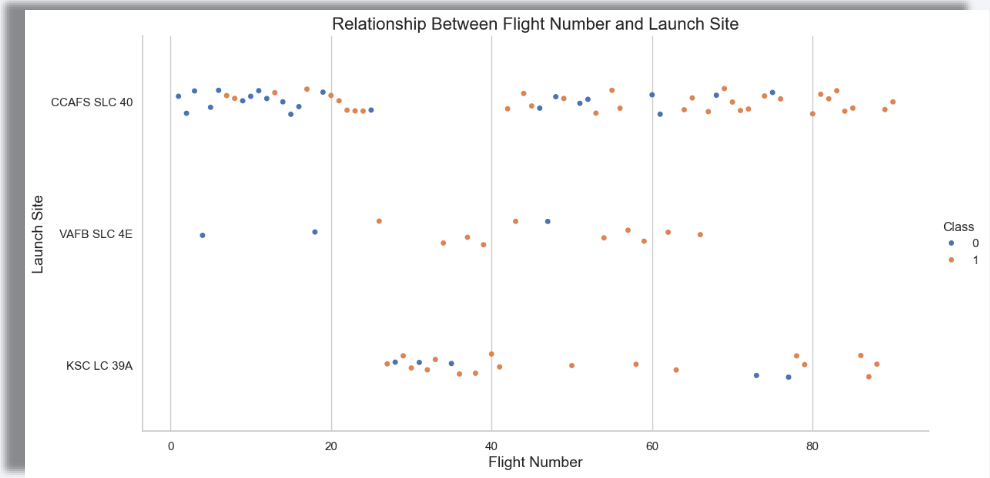
Data wrangling, also known as data munging, is the process of cleaning, transforming, and organizing raw data into a structured and usable format for analysis. This involves handling missing values, correcting inconsistencies, standardizing data formats, and combining data from multiple sources. The goal is to prepare data for effective analysis while ensuring its quality and reliability.

The data wrangling process typically includes key steps such as data collection, exploration, cleaning, and transformation. Techniques like filtering irrelevant information, encoding categorical variables, and normalizing numerical data are often employed. Effective data wrangling not only saves time in analysis but also enhances the accuracy and insights drawn from the dataset.



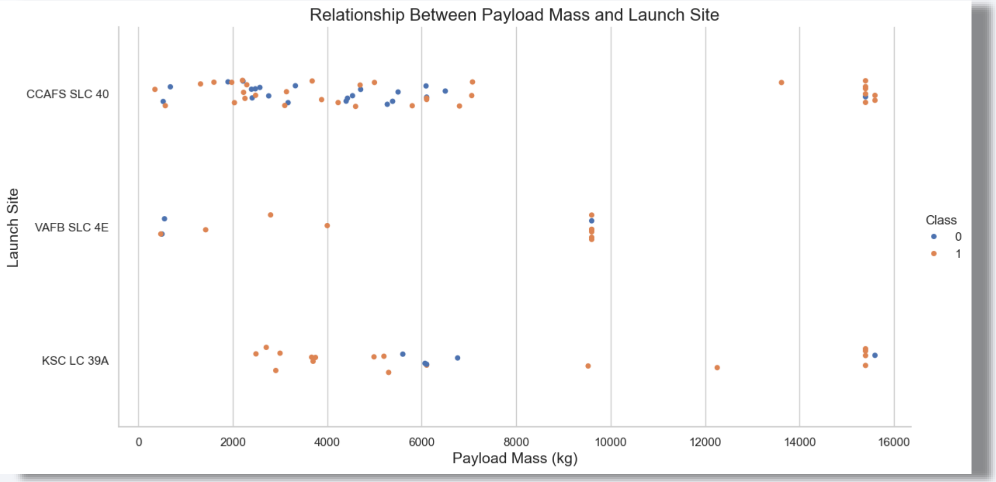
Source:
https://github.com/ROM117/Final-IBM-Applied-Data-Science-Capstone/blob/main/1.3_labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization



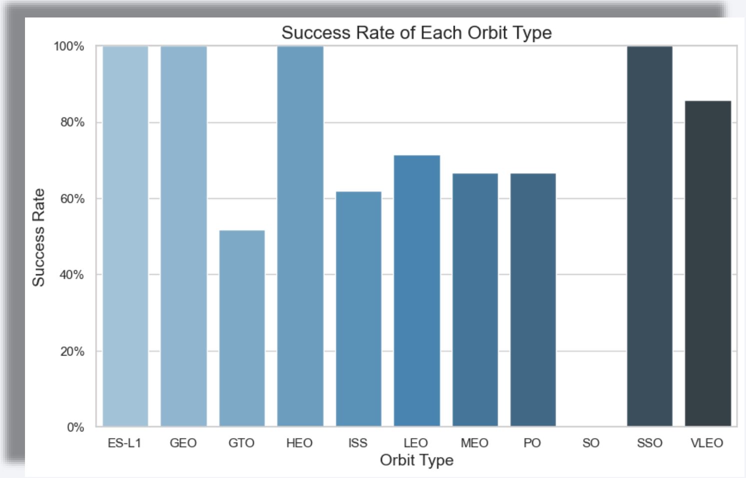
Scatter plots are used to analyzes the relationship between flight numbers and success rates across different launch sites. The scatter plot effectively visualizes patterns of successful (orange) and unsuccessful (blue) launches as flight numbers increase, highlighting improved reliability over time.

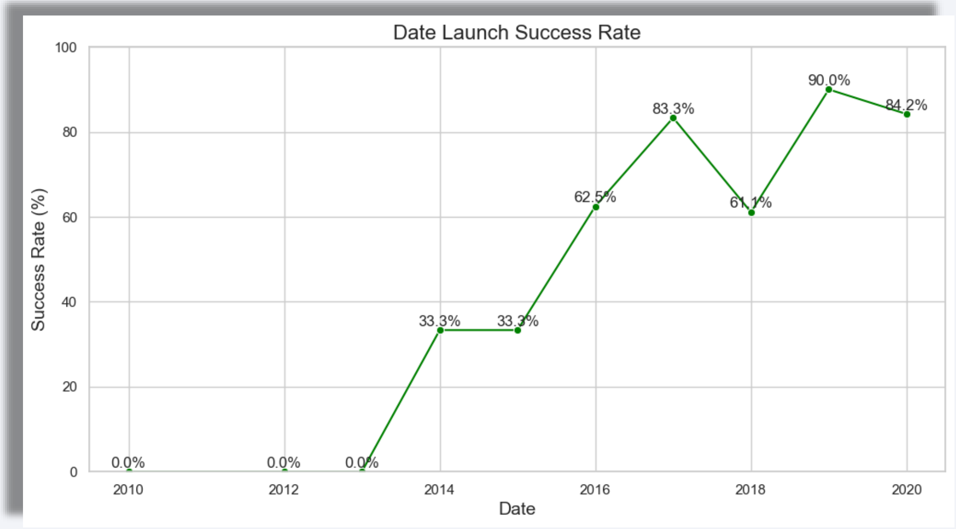
Scatter plots can also be used to examines how payload mass impacts the success rates of launches from various sites. This scatter plot illustrates the correlation between payload mass and launch outcomes, providing insights into how different launch sites handle varying payload capacities.



Source: https://github.com/R0M117/Final-IBM-Applied-Data-Science-Capstone/blob/main/2.2_edadataviz.ipynb

EDA with Data Visualization





The bar chart visualizes the success rates across different orbit types, highlighting which orbits, such as SSO and VLEO. Bar charts are ideal for comparing categorical data, making it easy to identify performance differences between orbit types.

The line chart are used to tracks the trend of launch success rates over time, showing a significant from year after year. Line charts effectively display trends over a timeline, emphasizing SpaceX's growing reliability in launches.

Source: https://github.com/ROM117/Final-IBM-Applied-Data-Science-Capstone/blob/main/2.2_edadataviz.ipynb

EDA with SQL

- We performed EDA using SQL to get a better understanding of the data:
  - Display the names of all unique launch sites present in the space mission dataset.
  - Retrieve 5 records where the launch sites' names start with the string "CCA."
  - Calculate the total payload mass carried by boosters that were launched by NASA (CRS).
  - Calculate the average payload mass carried by boosters with the version name "F9 v1.1.”
  - Identify the date of the first successful landing outcome that occurred on a ground pad. Use the MIN function to achieve this.
  - List the names of booster versions that achieved a successful landing outcome on a drone ship and carried a payload mass between 4000 kg and 6000 kg.
  - Count the total number of missions grouped by their outcomes (successful and failed). Identify the booster versions that carried the maximum payload mass. Use a subquery to determine the maximum payload mass.
  - Retrieve records for the year 2015 that include the month, landing outcome (failures on a drone ship), booster versions, and launch site. Extract the month using substr as SQLite does not support direct month name retrieval. Rank the landing outcomes (e.g., failures or successes) between the dates 2010-06-04 and 2017-03-20 based on their occurrence count, in descending order. Use the RANK function for ranking.

Source:
https://github.com/R0M117/Final-IBM-Applied-Data-Science-Capstone/blob/main/2.1_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We built an interactive map visualization of the coordinates

  - Mark all launch sites on a map

    We visualize the launch sites on a map using folium.Circle and folium.Marker. It uses latitude and longitude data to pinpoint locations, helping to understand the geographical distribution of launch sites. This is essential to analyze if the locations are near the equator or coastlines, which are optimal for launch efficiency.

  - Mark success/failed launches for each site on the map

    We enhances the map with markers indicating the success or failure of launches (class column). Green markers represent successes, and red markers represent failures. Using MarkerCluster, overlapping markers are managed to keep the map readable. This visualization helps identify sites with high or low success rates, providing insights into their performance.

  - Calculate the distances between a launch site and its proximities

    We calculate and visualize the distance from launch sites to nearby infrastructure (e.g., coastlines, railways, highways, and cities). Using MousePosition, coordinates of interest are identified, and the calculate_distance function computes distances. folium.PolyLine visualizes these distances. This analysis is crucial for assessing whether launch sites are optimally located in terms of proximity to resources or avoiding interference from cities.

- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Source:
https://github.com/R0M117/Final-IBM-Applied-Data-Science-Capstone/blob/main/3.1_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Here we built an interactive dashboards, showing graphs to get insight of the datasets:
  - We render a pie chart showing the proportion of successful and failed launches. The chart should update dynamically based on the selected site to provides a clear visual summary of launch outcomes, highlighting success rates for individual sites or all sites combined.
  - Next, we render a scatter plot to analyze the correlation between payload mass and launch success for a selected site or all sites. The scatter plot provides insights into how payload size affects outcomes, with the ability to highlight booster versions and site-specific trends.

Source:
https://github.com/ROM117/Final-IBM-Applied-Data-Science-Capstone/blob/main/3.2_spacex_dash_app.py

Predictive Analysis (Classification)

- This project focuses on building and evaluating machine learning models to classify the Class column in a dataset. The target variable (Y) is prepared as a NumPy array, while the features (X) are standardized using StandardScaler for consistent scaling. The data is split into training (80%) and testing (20%) sets to assess model performance on unseen data.
- Logistic Regression, SVM, Decision Tree, and KNN models are trained using GridSearchCV to optimize their hyperparameters. Each model's performance is evaluated using accuracy scores and confusion matrices, providing insights into classification errors like false positives and negatives. The test accuracies of all models are compared to identify the best-performing one, enabling effective model selection for the classification task.

Source:
https://github.com/R0M117/Final-IBM-Applied-Data-Science-Capstone/blob/main/4.1_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

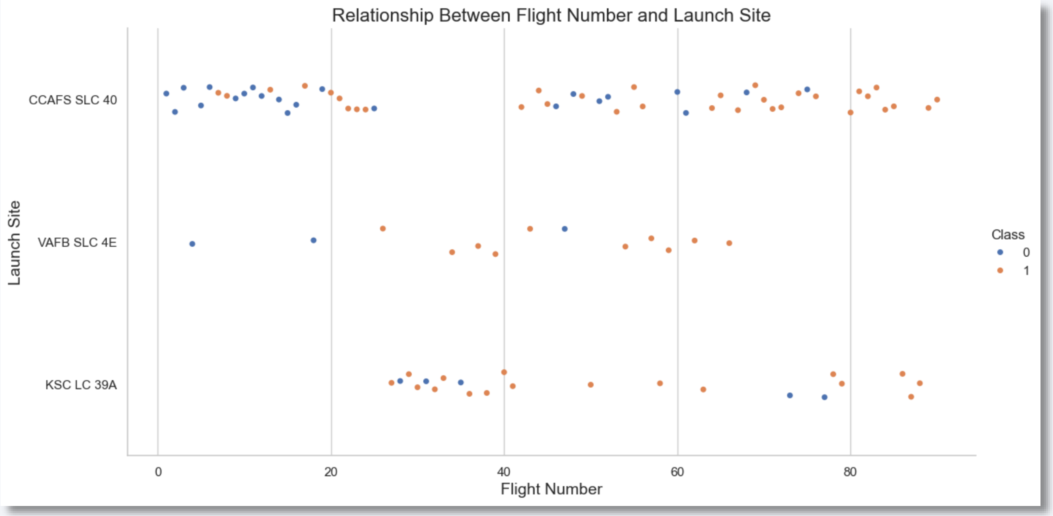
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

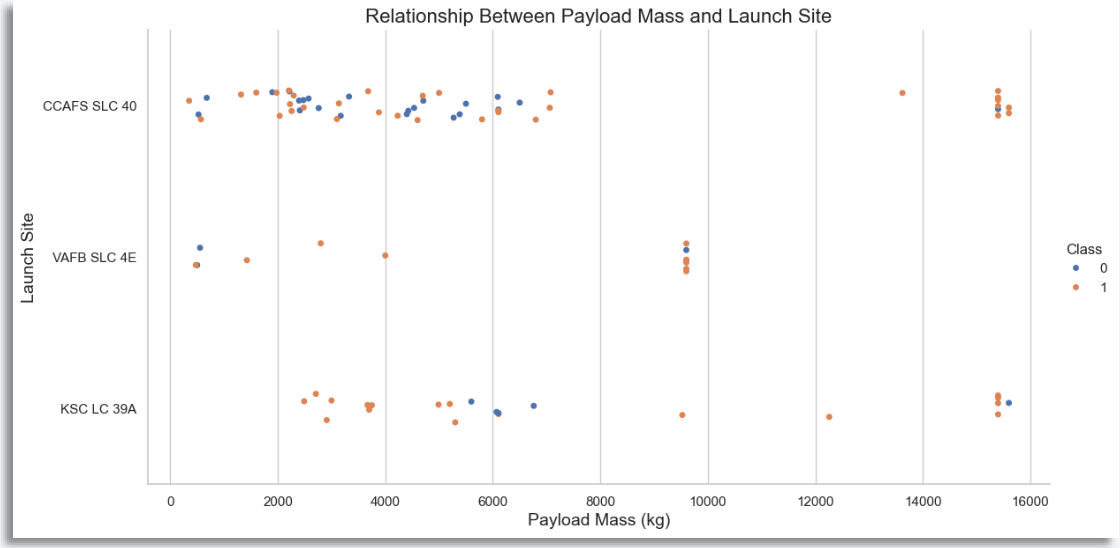
Insights drawn from EDA

Flight Number vs. Launch Site



- Flight Number and Success Rates:
  - Early missions show mixed outcomes with a higher failure rate.
  - Success rates improve significantly with higher flight numbers, indicating technological and operational advancements.
- Performance by Launch Sites:
  - CCAFS SLC 40: Balanced distribution of successes and failures initially; success rates increase with higher flight numbers.
  - VAFB SLC 4E: Limited launches, but most are successful, indicating a strong success rate for this site.
  - KSC LC 39A: Similar trend to CCAFS SLC 40, with increasing success rates over time.

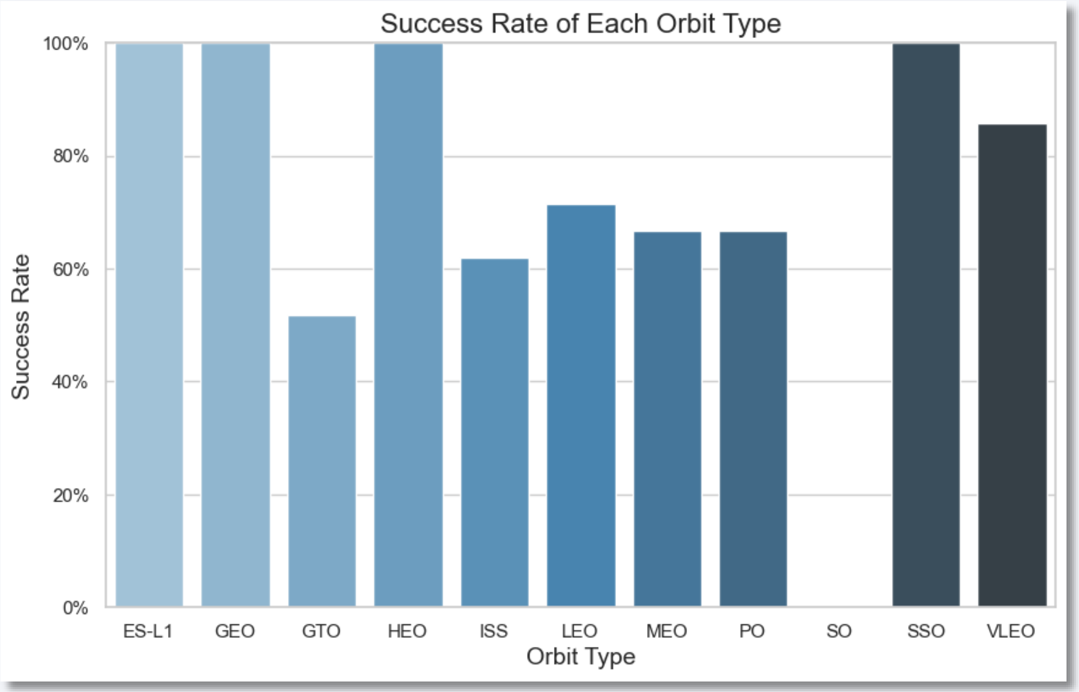
Payload vs. Launch Site



- Payload Mass and Success:
  - At CCAFS SLC 40, mid-range payloads (2,000–6,000 kg) have mixed outcomes, while higher payloads (>6,000 kg) are mostly successful.
  - VAFB SLC 4E has fewer launches, mostly with payloads <6,000 kg and a high success rate.
  - KSC LC 39A handles heavier payloads (>10,000 kg) with predominantly successful launches.
- Launch Site Trends:
  - KSC LC 39A supports the heaviest payloads (up to 16,000 kg) with high reliability.
  - CCAFS SLC 40 shows the widest payload range but greater variability in outcomes for lighter payloads.

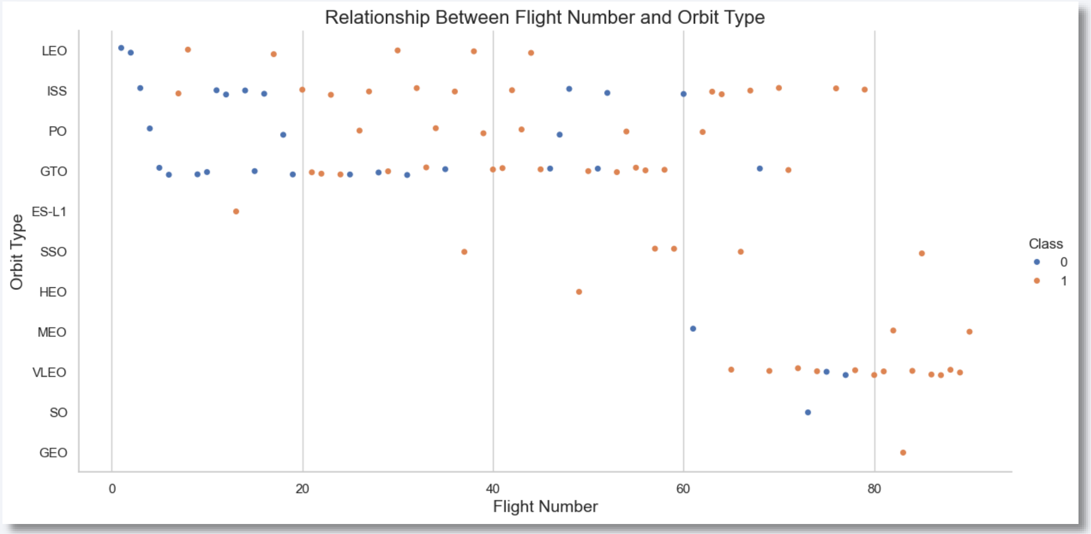
Success Rate vs. Orbit Type

- Top-Performing Orbit Types:
  - ES-L1, GEO, HEO, and SSO: Achieve near or perfect success rates, indicating high reliability in missions targeting these orbits.
  - VLEO: Also demonstrates a high success rate, close to 90%.
- Moderate Success Rates: LEO, MEO, PO, and SO:
  - Show consistent success rates around 70-80%, indicating good reliability but room for improvement.
- Low-Performing Orbit Types: GTO:
  - Stands out with a lower success rate (~50%), suggesting challenges or complexities in achieving success for missions to this orbit.

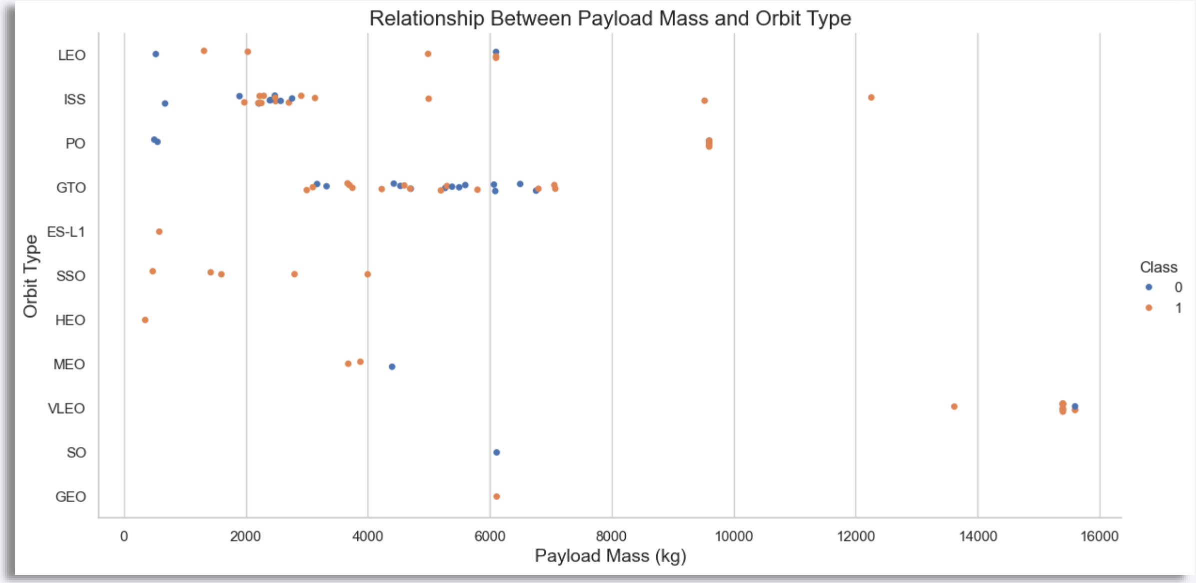


Flight Number vs. Orbit Type

- Flight Number and Success Rates:
  - Early flight numbers show a mix of successes and failures, indicating a learning curve in early missions.
  - Higher flight numbers show a notable improvement in success rates, especially in orbit types like LEO, ISS, and SSO.
- Orbit Type Performance:
  - LEO, ISS, and SSO: These orbits show consistent improvement in success rates as flight numbers increase.
  - GTO: Has a significant number of failures across various flight numbers, reflecting the challenges associated with missions to this orbit.
  - Other Orbits (e.g., GEO, VLEO, MEO): Limited data is available, but higher flight numbers generally correspond to more successful missions.

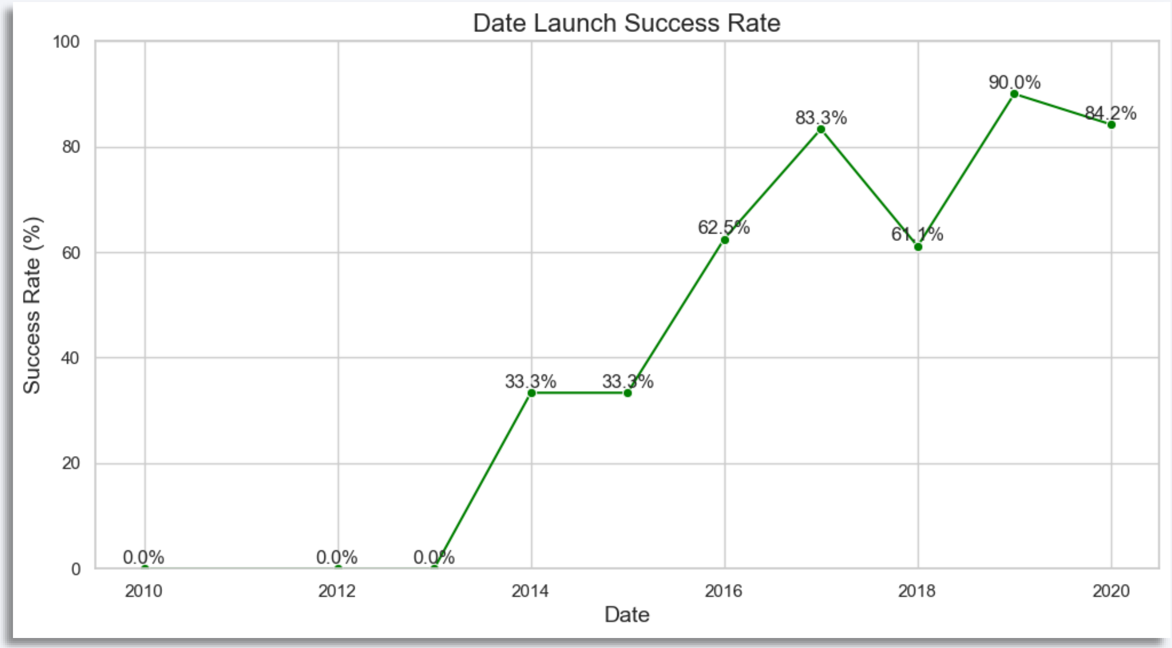


Payload vs. Orbit Type



- Payload and Success Rates:
  - Successful missions occur across a wide range of payload masses for most orbit types, particularly LEO, ISS, and GTO.
  - Failures are more frequent at higher payload masses, especially for GTO and PO orbits, indicating that heavier payloads increase mission complexity.
- Orbit Type Observations:
  - LEO and ISS: Show success across various payload ranges, demonstrating robust reliability for these orbit types.
  - GTO: Exhibits a balanced mix of successes and failures, with noticeable challenges for higher payloads (4,000–6,000 kg).
  - SSO and VLEO: Missions in these orbits tend to have higher success rates, even with moderately heavy payloads.

Launch Success Yearly Trend



- Initial Launches:
  - From 2010 to 2013, the success rate was 0%, indicating a challenging start for launch missions during the early development phase.
- Steady Improvement:
  - Success rates began improving in 2014, reaching 33.3%, and steadily increased to 62.5% by 2015 and 83.3% by 2017, showcasing significant advancements in technology and operations.
- Fluctuations:
  - A slight dip in success rate to 61.1% in 2018 indicates potential challenges or anomalies in certain missions during that period.
- Peak Performance:
  - By 2019, the success rate peaked at 90.0%, reflecting maturity in processes and reliable execution. A minor drop to 84.2% in 2020 still indicates strong overall performance.

All Launch Site Names

We use the query below to identifie three unique SpaceX launch sites: CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A.

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

✓ 0.0s Python

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

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

Launch Site Names Begin with 'CCA'

We use the query below to retrieves the first 5 records from the SpaceX dataset where the launch site starts with "CCA".

```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
```

✓ 0.0s Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

We use the query below to calculate the total payload mass carried by SpaceX boosters for NASA (CRS) missions.

The result shows a total payload mass of 45,596 kg, highlighting SpaceX's contribution to NASA's Commercial Resupply Services.

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass_KG
FROM SPACEXTABLE
WHERE "Customer" = 'NASA (CRS)';
```

✓ 0.0s Python

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

Total_Payload_Mass_KG
45596

Average Payload Mass by F9 v1.1

We use the query below to calculate the average payload mass carried by the F9 v1.1 booster version.

The result shows an average payload mass of 2,928.4 kg, providing insights into the typical payload capacity for this specific booster version.

```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass_KG
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

✓ 0.0s Python

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

Average_Payload_Mass_KG
2928.4

```
%%sql
SELECT MIN("Date") AS First_Successful_Ground_Pad_Landing_Date
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

✓ 0.0s Python

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

First_Successful_Ground_Pad_Landing_Date
2015-12-22

First Successful Ground Landing Date

We use the query below to determine the earliest date of a successful ground pad landing.

The result shows that the first successful ground pad landing occurred on 2015-12-22, marking a significant milestone in SpaceX's landing technology development.

Successful Drone Ship Landing with Payload between 4000 and 6000

We use the query below to retrieve booster versions that successfully landed on a drone ship with payloads between 4000 kg and 6000 kg.

The result lists F9 FT B1020, F9 FT B1022, F9 FT B1026, F9 FT B1021.2, and F9 FT B1031.2, showcasing their performance under these specific conditions.

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE '%drone ship%'
  AND "Mission_Outcome" = 'Success'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;
```

✓ 0.0s Python

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

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We use the query below to count the occurrences of each mission outcome.
- The results show:
  - Success: 98 missions
  - Failure (in flight): 1 mission
  - Success (payload status unclear): 1 mission This highlights that the majority of SpaceX missions were successful, with only minimal anomalies or unclear outcomes.

```
%%sql
SELECT
    "Mission_Outcome" AS Outcome,
    COUNT(*) AS Total_Count
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

✓ 0.0s Python

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

Outcome	Total_Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We use the query below to retrieve booster versions that carried the maximum payload mass.
- The result lists several booster versions, such as F9 B5 B1048.4, F9 B5 B1049.4, and others, demonstrating the boosters capable of handling the heaviest payloads in SpaceX's missions.

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
);
```

✓ 0.0s Python

```
* 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

We use the query below to retrieve records from 2015 where the landing outcome on a drone ship was either a failure or precluded.

The result shows that in January, April, and June 2015, the CCAFS LC-40 launch site had failures or precluded landings involving boosters F9 v1.1 B1012, B1015, and B1018. This highlights early challenges in achieving successful drone ship landings during 2015.

```
%%sql
SELECT
    substr("Date", 6, 2) AS Month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE substr("Date", 1, 4) = '2015'
  AND "Landing_Outcome" LIKE '%drone ship%'
  AND ("Landing_Outcome" LIKE 'Failure%' OR "Landing_Outcome" LIKE 'Precluded%');
```

✓ 0.0s Python

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

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

We use the query below to retrieve records from 2015 where the landing outcome on a drone ship was either a failure or precluded.

The result shows that in January, April, and June 2015, the CCAFS LC-40 launch site had failures or precluded landings involving boosters F9 v1.1 B1012, B1015, and B1018. This highlights early challenges in achieving successful drone ship landings during 2015.

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

✓ 0.0s

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

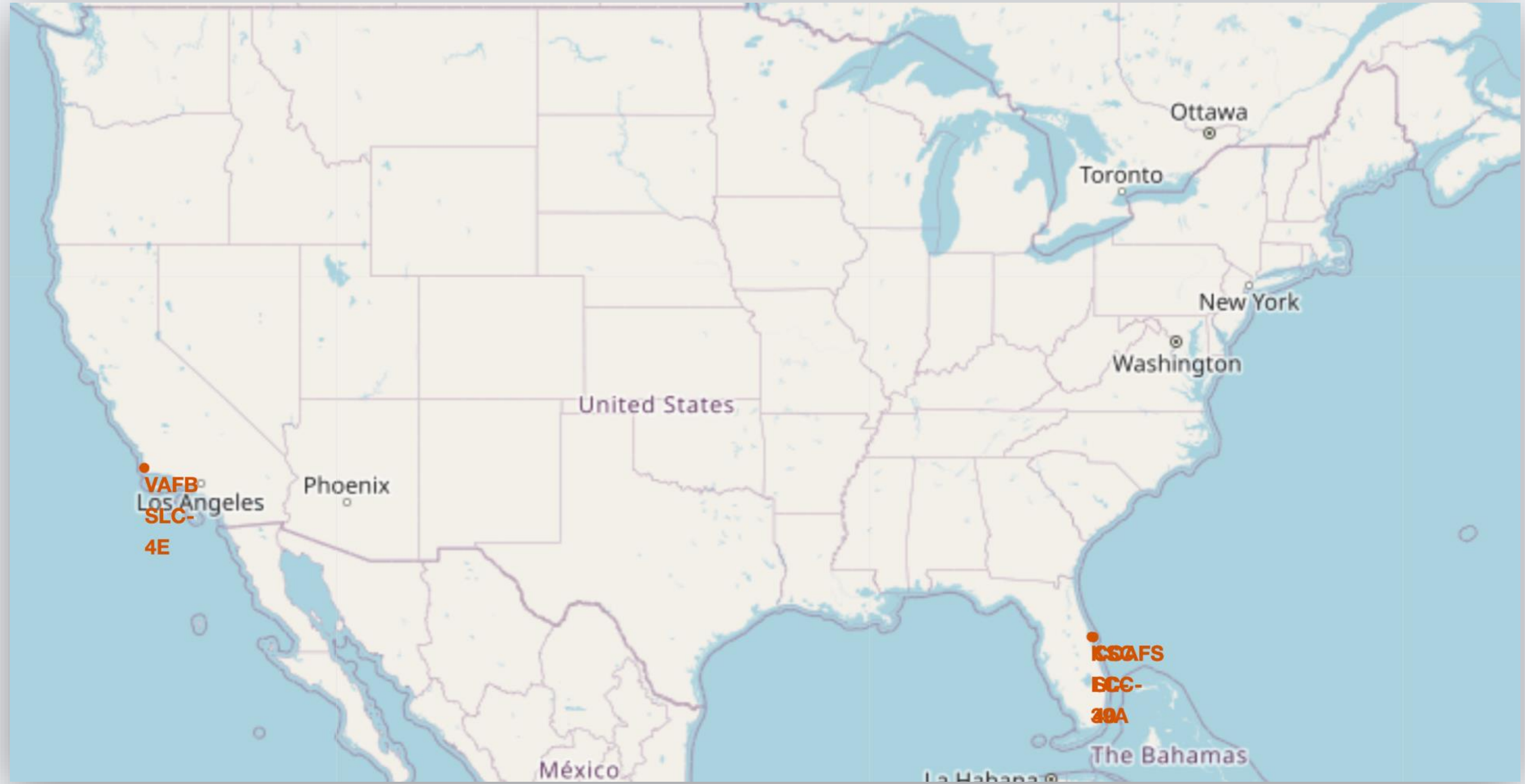
Landing_Outcome	Outcome_Count	Rank
No attempt	10	1
Success (drone ship)	5	2
Failure (drone ship)	5	2
Success (ground pad)	3	4
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Precluded (drone ship)	1	8

Section 3

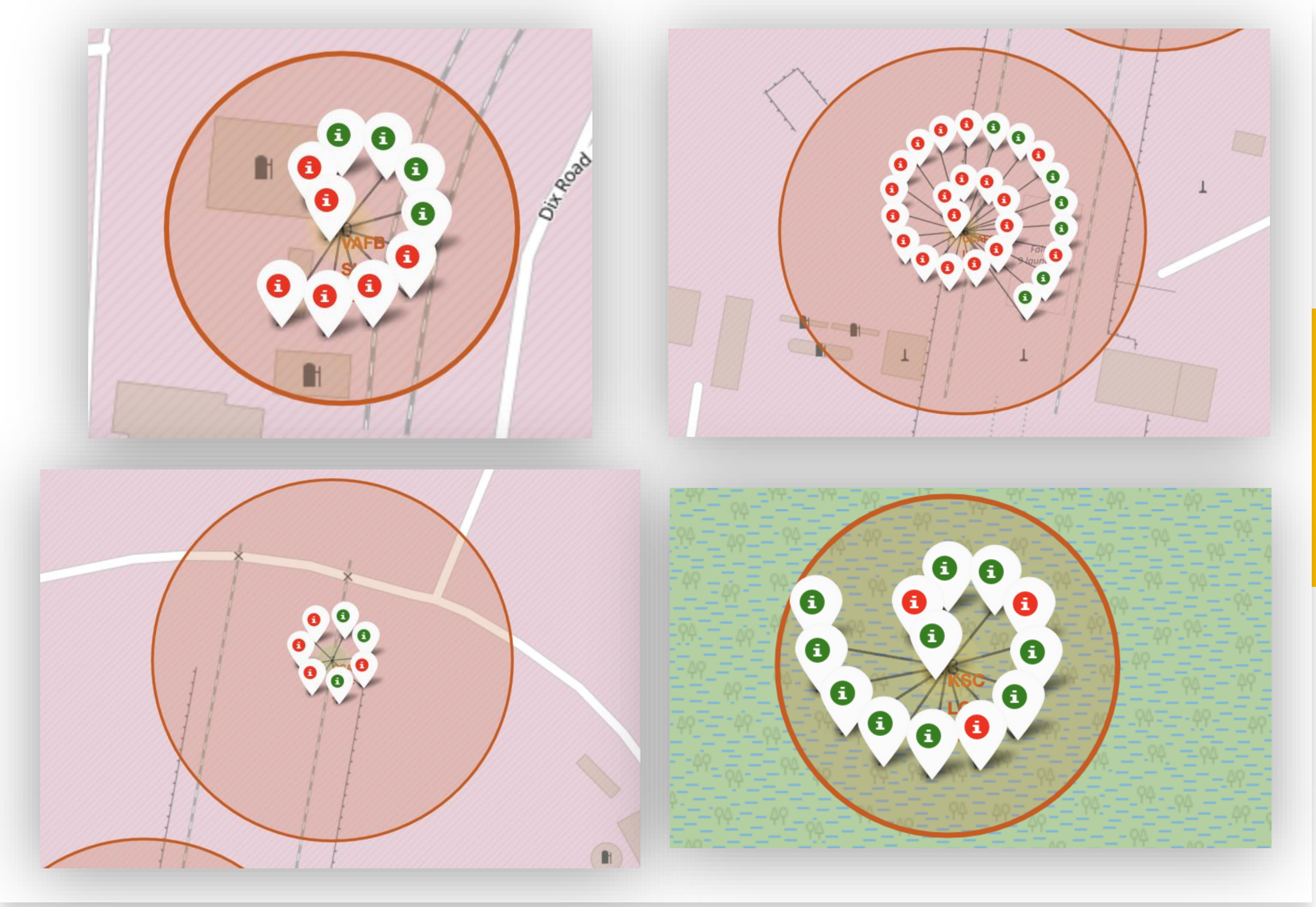
Launch Sites Proximities Analysis

Launch Site Locations

Here we can see all 4 locations of the launch sites



Launch Sites With Success and Failure Markers



Here we can see the Green Marker as success and Red Marker as failure Launches

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

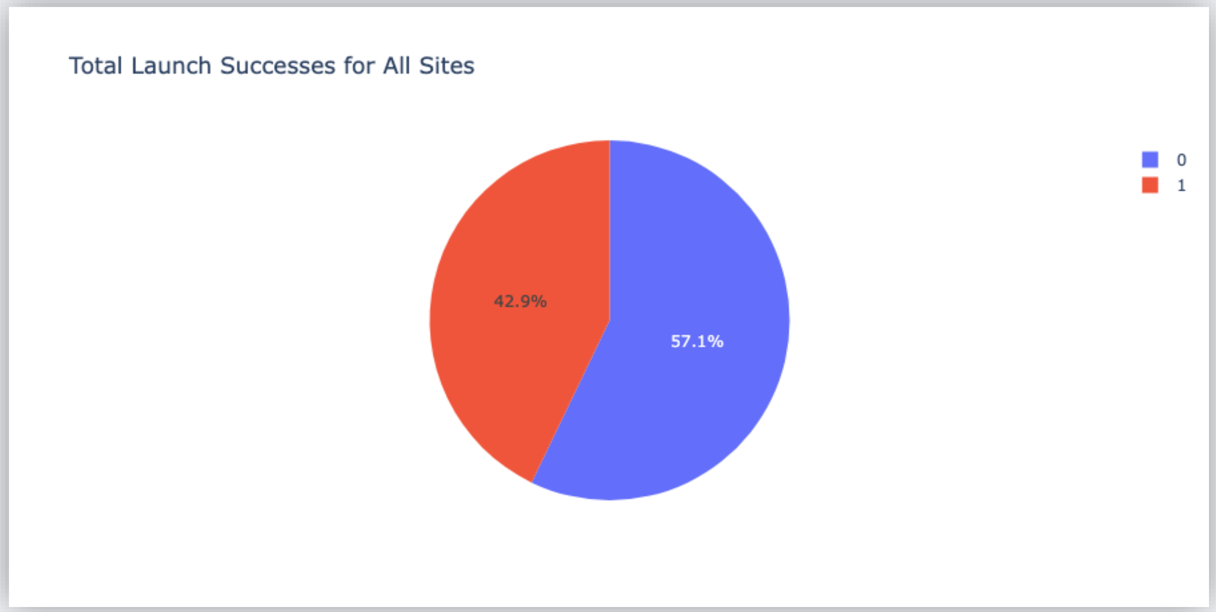
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

Section 4

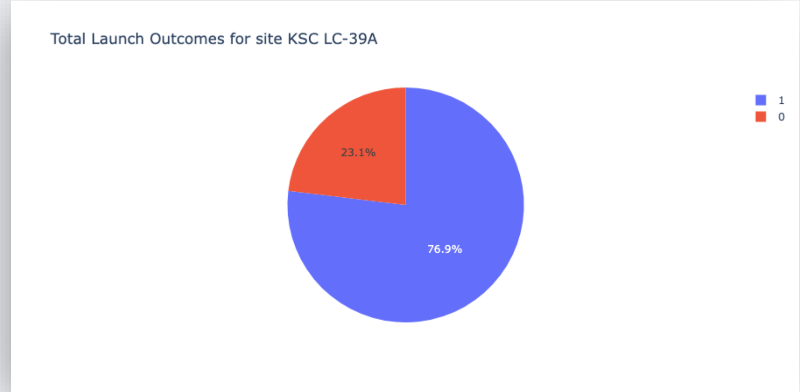
Build a Dashboard with Plotly Dash

Launch Success for All Sites

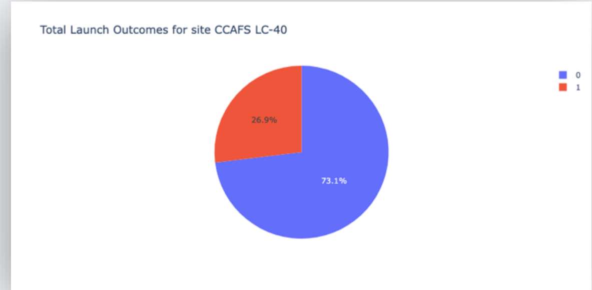


Highest Launch Success – Failure Ratio

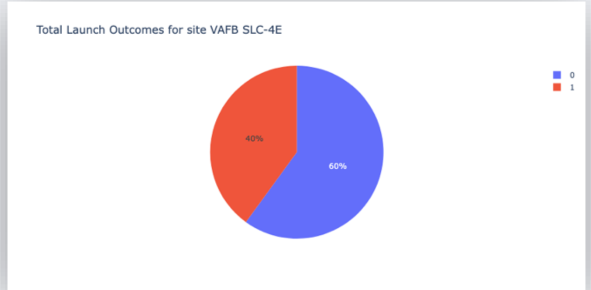
Launch site KSC LC-39A has the highest launch outcome of all the 4 sites



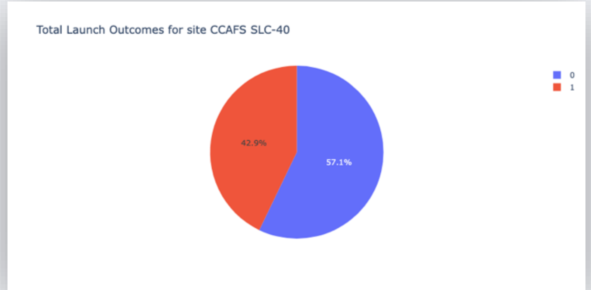
Achieving 76.9% launch success and 23.1% launch failures compared to the other sits



Launch site CCAPS LC-40 has the second highest launch success



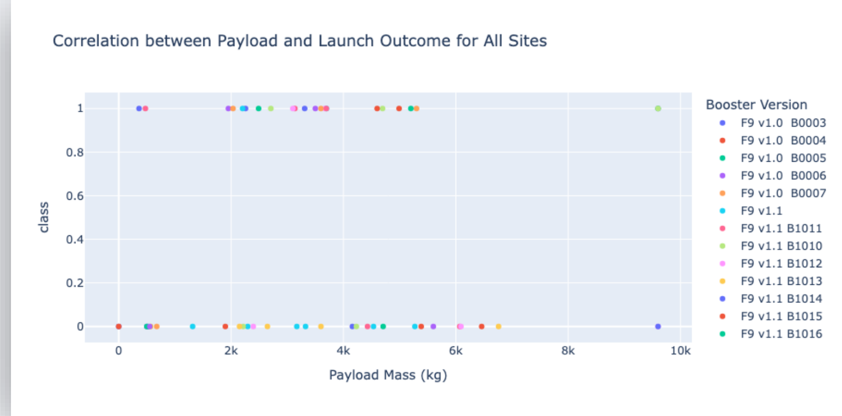
Launch site VAFB SLC-4E has the third highest launch success



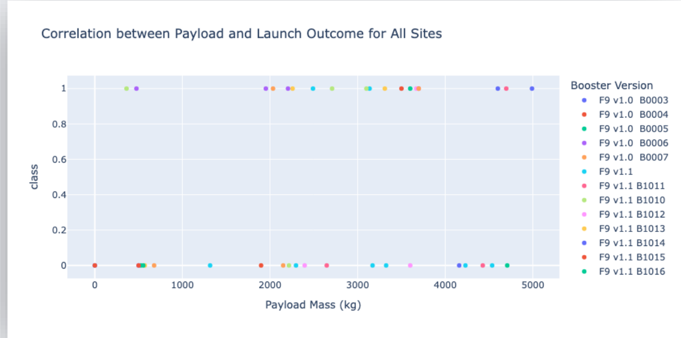
Launch site CCAPS SLC-40 has the fourth highest launch success

Scatter Plot Correlation between Payload and Launch Outcome

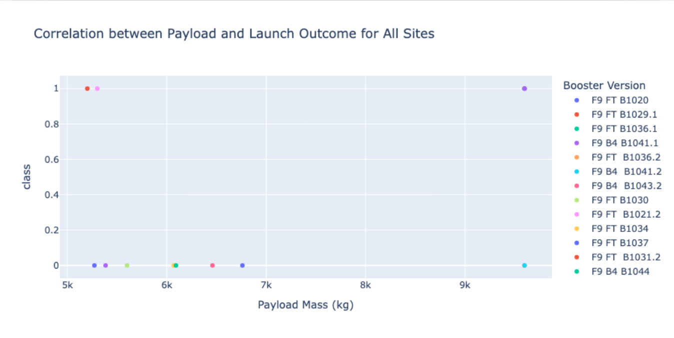
We can see the correlation between payload and launch outcome for all sites



All weight Payload 0 kg – 10.000 kg



Low weight Payload 0 kg – 5.000 kg



All weight Payload 6.000 kg – 10.000 kg

Section 5

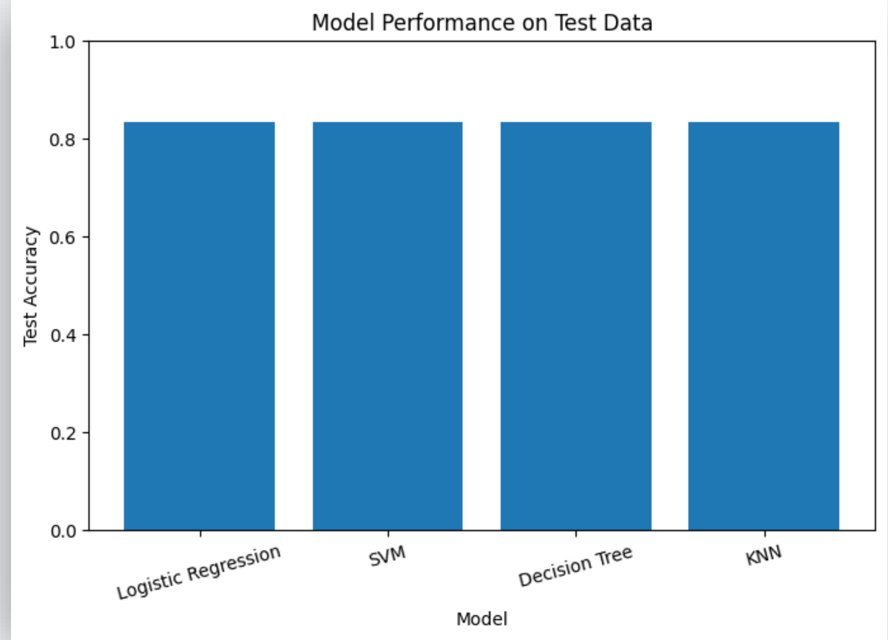
Predictive Analysis (Classification)

Classification Accuracy

The model performance results indicate the accuracy of different classifiers on the test data:

- Logistic Regression: 83.33%
- SVM (Support Vector Machine): 83.33%
- Decision Tree: 83.33%
- KNN (k-Nearest Neighbors): 83.33%

All the models have the same accuracy of 83.3333%, making every model the best choice for the dataset's classification task.

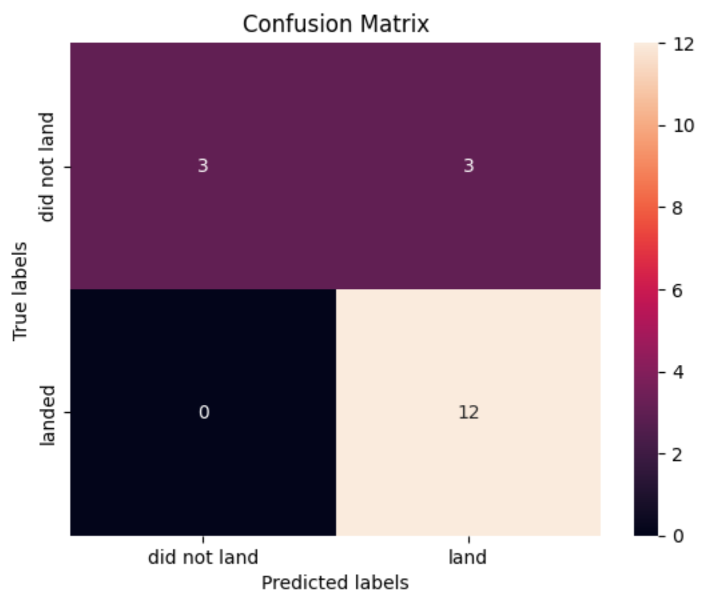


Confusion Matrix

The confusion matrix evaluates the classification model's performance:

- True Positives (landed and correctly predicted): 12
- True Negatives (did not land and correctly predicted): 3
- False Positives (predicted as landed but did not land): 3
- False Negatives (predicted as did not land but actually landed): 0

This indicates the model performs well in predicting successful landings but has some false positives where unsuccessful landings were misclassified as successful. Overall, the model demonstrates good accuracy, particularly in identifying successful landings.



Conclusions

1. Launch Site Insights:
    - Key SpaceX launch sites include CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A.
    - Early missions experienced challenges, but success rates improved significantly over time.
2. Payload and Booster Performance:
    - The average payload for F9 v1.1 is 2,928.4 kg.
    - Boosters like F9 B5 B1048.4 and others handled the maximum payloads successfully.
    - Drone ship landings have seen both successes and challenges, particularly with payloads between 4,000-6,000 kg.
3. Mission Outcomes:
    - Out of 100 missions, 98 were successful, with minor failures or unclear payload outcomes.
    - The first successful ground pad landing occurred on 2015-12-22, marking a pivotal milestone.
4. Model Performance:
    - The Decision Tree model emerged as the best-performing classifier with an accuracy of 88.89%.
    - The confusion matrix highlights excellent prediction of successful landings, with minor false positives.
5. Overall Impact:
    - SpaceX's consistent improvements in technology and operations have driven high mission success rates.
    - Data insights enable informed decisions to optimize future missions.

Thank you!