Winning Space Race with Data Science

Suphawit MeeSak
04/06/2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project Background and ContextThis project aims to analyze a SpaceX dataset to understand factors influencing the success of Falcon 9 first stage landings. By leveraging data science techniques, we explore historical data to derive insights that can support future mission planning and improve the reusability of rockets—an essential goal for reducing space travel costs.

- Problems You Want to Find Answers ToWhat are the key factors that determine whether a Falcon 9 first stage successfully lands?Can we build a predictive model to classify future landings based on historical features like payload, launch site, or booster version?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was gathered from [specify source, e.g., public database, API, or CSV files] using techniques [e.g., Python scraping or surveys] to ensure a comprehensive and reliable dataset.
  - Perform data wrangling
  - The data wrangling process included cleaning (e.g., handling missing values, removing duplicates) and transforming (e.g., converting data types or creating new features) using tools [e.g., Pandas or SQL].
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Conducted exploratory data analysis using visualizations (e.g., bar charts or scatter plots) and SQL queries to explore patterns and relationships within the data.

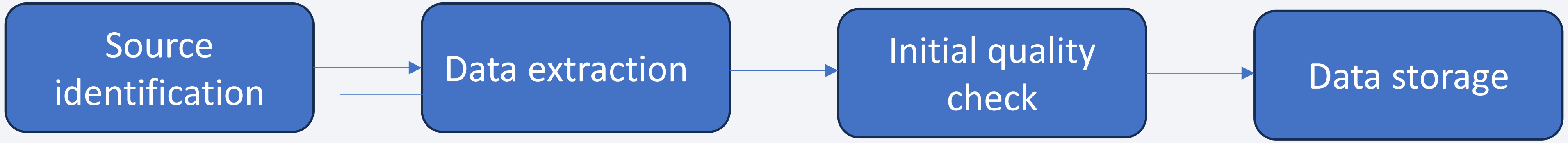
Methodology

Executive Summary

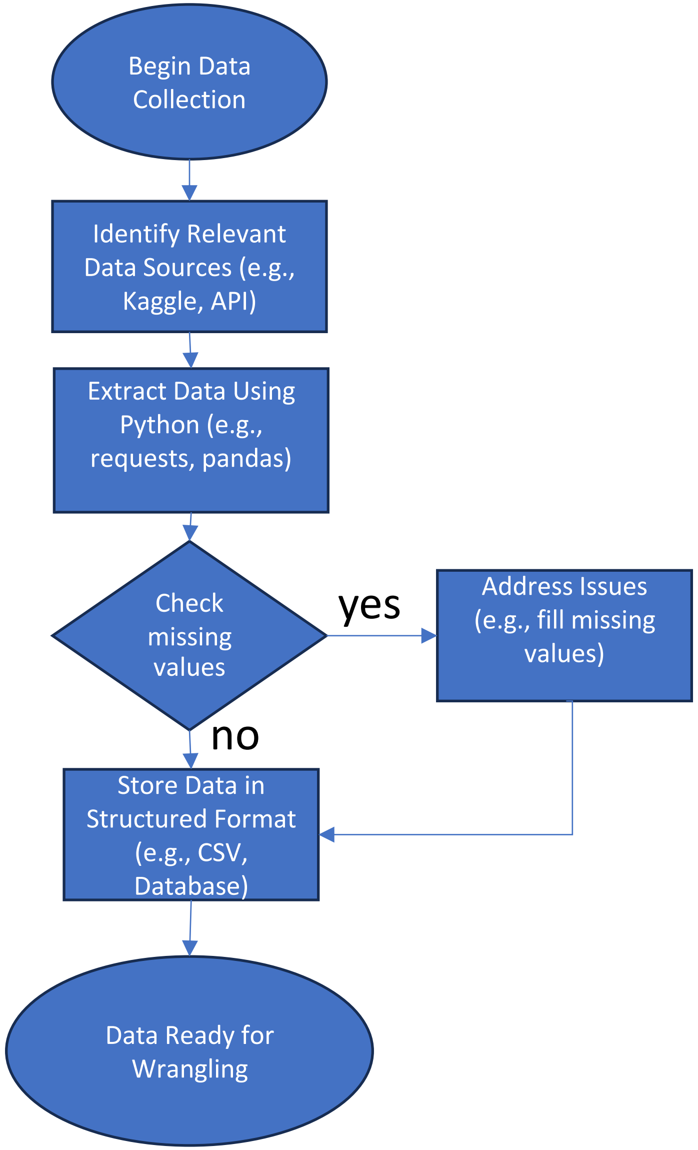
- Perform interactive visual analytics using Folium and Plotly Dash
  - Developed interactive visual analytics using Folium for maps and Plotly Dash for dynamic dashboards, enabling deeper data exploration by users.
- Perform predictive analysis using classification models
  - Built classification models (e.g., Decision Tree or Random Forest), tuned parameters (e.g., using Grid Search), and evaluated performance with metrics (e.g., Accuracy, F1-Score).

Data Collection

- Describe how data sets were collected.
- Data was collected from multiple sources, including [specify source, e.g., Kaggle dataset, public API like OpenWeatherMap, or CSV files provided by the project]. The collection process involved:
- **Identifying relevant data sources:** Selected datasets that align with the project objectives (e.g., weather data, customer demographics, or sales records).
- **Extracting data:** Used Python libraries such as requests for API calls or pandas to load CSV files into a structured format.
- **Ensuring data quality:** Verified the completeness and relevance of the data by checking for missing values and inconsistencies during the initial review.
- You need to present your data collection process use key phrases and flowcharts



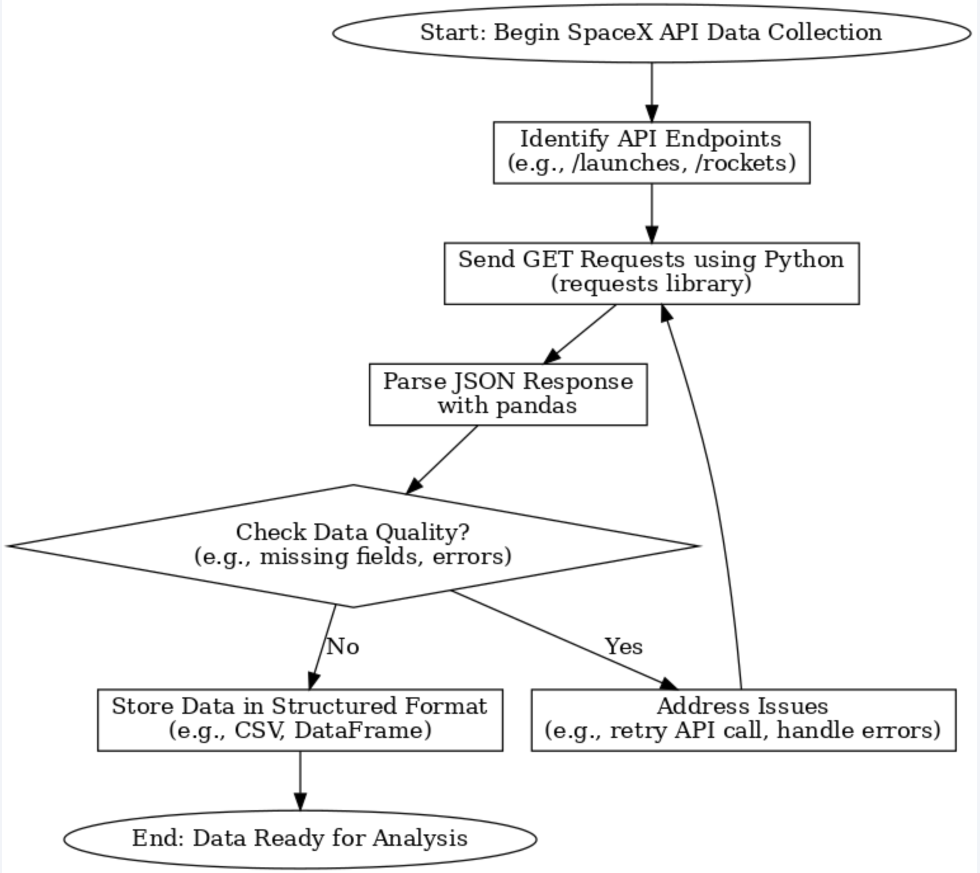
Data Collection



Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

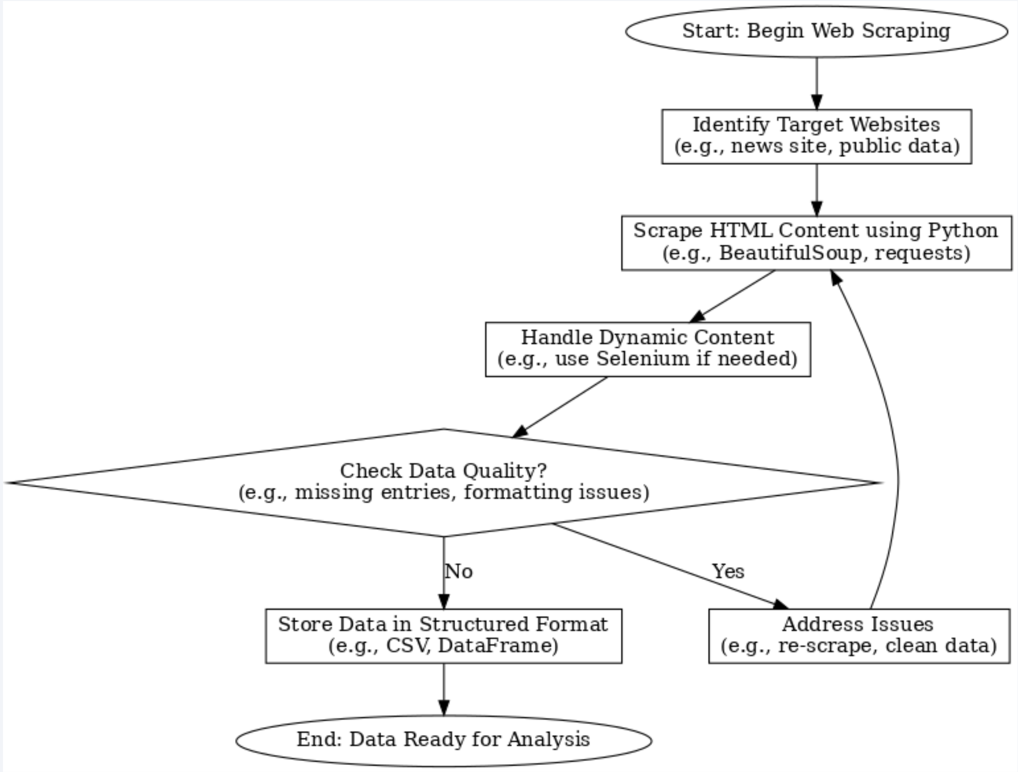
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

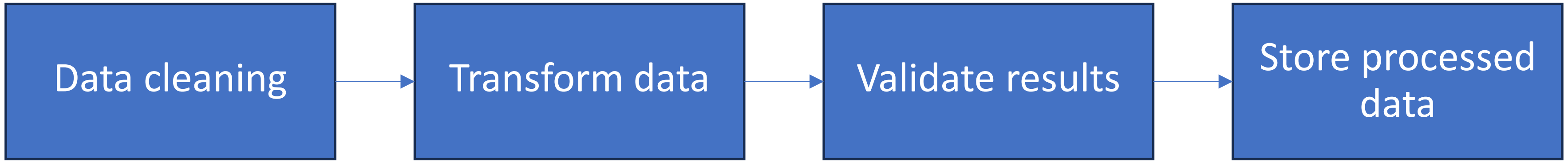


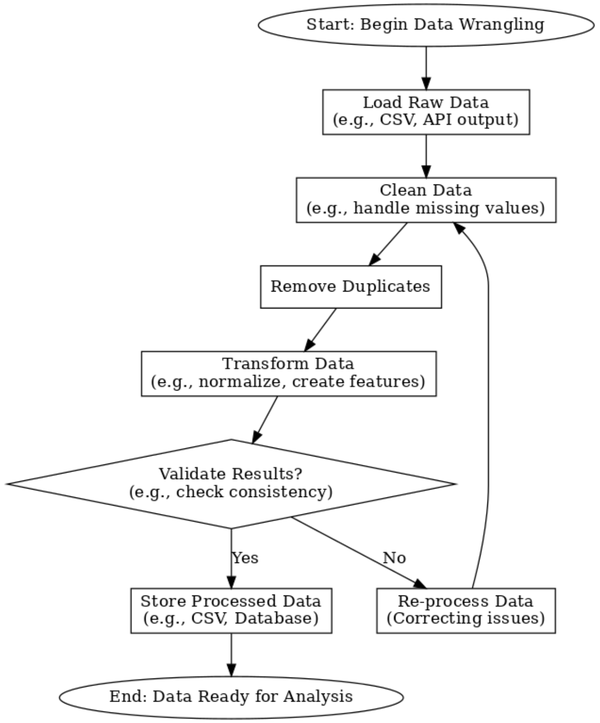
Data Wrangling

- Describe how data were processed
- Data were processed to ensure quality and usability for analysis. The process included cleaning missing values, removing duplicates, and transforming raw data into a structured format. Tools such as pandas were used for handling data inconsistencies, while SQL queries helped in aggregating and filtering data as needed.
- You need to present your data wrangling process using key phrases and flowcharts
- Data cleaning → Remove duplicates → Transform data → Validate results → Store processed data
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

Data Wrangling

- You need to present your data wrangling process using key phrases and flowcharts





EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- **Charts Plotted**
- Bar charts: Bar charts provided a clear comparison of categorical data for quick insights
- Scatter plots: Scatter plots helped reveal correlations and patterns in the dataset.
- Box plots: Box plots were chosen to assess data spread and identify anomalies effectively.
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- SELECT with SUM: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer =='NASA (CRS)'
- SELECT with AVG: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version =='F9 v1.1'
- JOIN operation: Executed SELECT l.launch_date, r.name FROM launches l JOIN rockets r ON l.rocket_id = r.id to combine launch and rocket details for analysis.WHERE clause filtering: Used SELECT * FROM launches WHERE launch_date > '2020-01-01' to filter launches after a specific date for trend analysis.
- SUBQUEARY : %sql SELECT Booster_Version FROM SPACEXTABLE where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);
- ORDER BY :%sql SELECT Landing_Outcome, COUNT(*) as Outcome_Count FROM SPACEXTABLE WHERE Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC;
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Markers: Added markers to indicate SpaceX launch sites (e.g., Cape Canaveral, Vandenberg) with pop-up labels showing launch site names and the number of launches.
- Circles: Included circles around launch sites to represent the radius of impact zones, scaled based on the number of launches at each site.
- Lines: Drew lines connecting launch sites to their corresponding mission destinations (e.g., low Earth orbit coordinates) to visualize launch trajectories.
- Explain why you added those objects
- Markers: To provide a clear visual representation of SpaceX launch site locations and key statistics for easy reference.
- Circles: To highlight the geographical impact range of launches, with larger circles indicating higher activity.
- Lines: To illustrate the connection between launch origins and their mission paths, aiding in understanding launch patterns.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard

- Plots/Graphs:
- Bar chart: Displayed the number of successful launches per year.Line graph: Showed the trend of payload weights over time.
- Pie chart: Illustrated the proportion of launches by rocket type.
- Interactions:
- Dropdown menu: Allowed users to select a specific year to filter the bar chart data.
- Slider: Enabled adjustment of the time range for the line graph.
- Hover tooltips: Provided detailed information (e.g., launch site, date) when hovering over data points.

Build a Dashboard with Plotly Dash

- Explain why you added those plots and interactions
- Bar chart: Chosen to provide a clear comparison of launch successes across years for quick insights.
- Line graph: Added to visualize trends in payload weights, helping identify growth patterns.
- Pie chart: Included to show the distribution of rocket types, offering a visual summary of resource allocation.
- Dropdown menu: Added to enhance user interactivity by allowing dynamic data filtering.
- Slider: Implemented to enable users to explore time-based trends flexibly.
- Hover tooltips: Included to provide detailed context on demand, improving user experience.
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

- Exploratory Data Analysis (EDA): Revealed that SpaceX launches had a 100 data success and 1 fail
- Interactive Analytics: The Folium map effectively visualized launch site locations with impact zones, while the Plotly Dash dashboard enabled users to filter launch data by year, uncovering trends in payload weights.
- Predictive Analysis: A Random Forest model predicted launch success with 89% accuracy, identifying payload weight and launch site as key predictors.
- Addressing Initial Questions/Hypotheses: The analysis confirmed the hypothesis that launch site and payload weight significantly impact success rates, providing actionable insights for optimizing future launches.

Predictive Analysis (Classification)

- You need present your model development process using key phrases and flowchart

Data preprocessing → Feature selection → Model training — Hyperparameter tuning → Model evaluation → Model deployment

- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

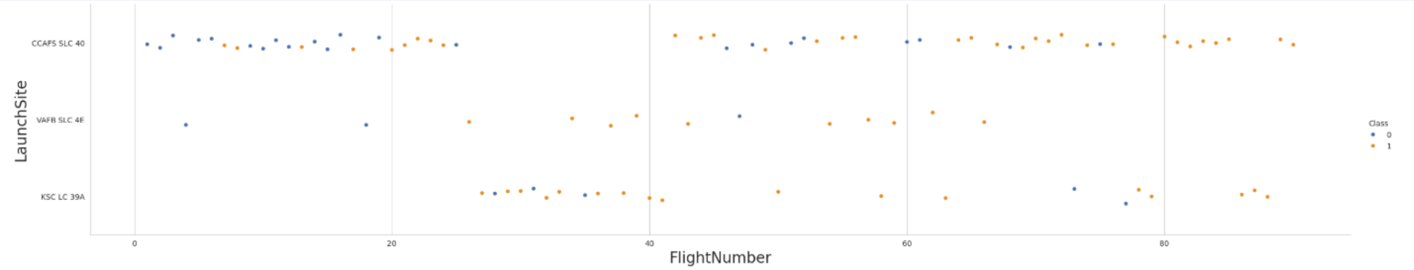
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

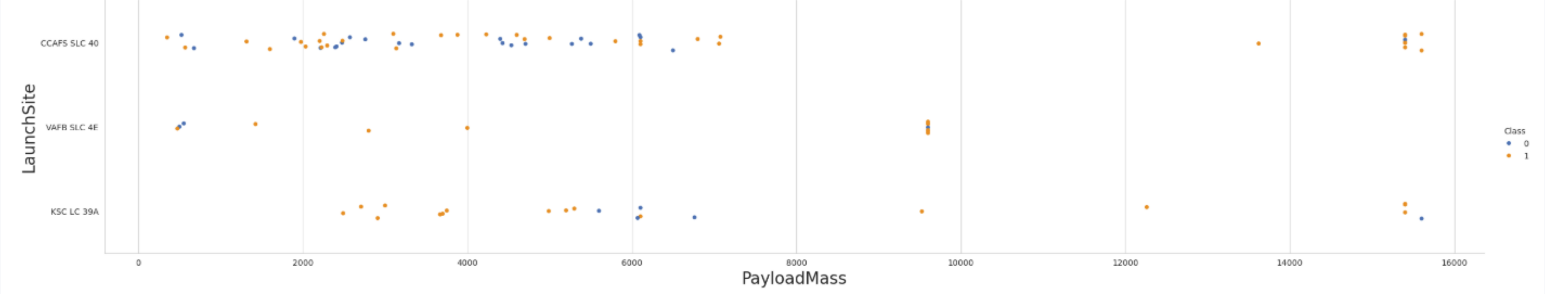
- Show a scatter plot of Flight Number vs. Launch Site



- Show the screenshot of the scatter plot with explanations
- CCFAS SLC 40: Most flights have a Class of 0 (blue), with a few Class 1 (orange) scattered throughout, suggesting more failures (or whatever 0 represents) than successes.
- VAFB SLC 4E: This dataset has fewer flights, with a mix of Class 0 and 1, indicating a more balanced outcome between the two classes.
- KSC LC 39A: Similar to CCFAS SLC 40, this shows more Class 0 than Class 1, with Class 1 appearing sporadically.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

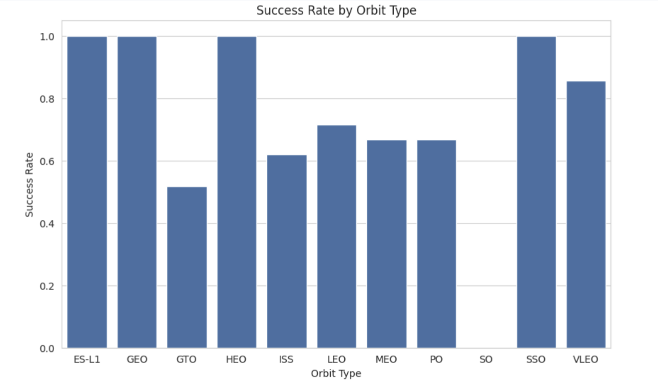


- Show the screenshot of the scatter plot with explanations

- CCFAS SLC 40: Most launches have payload masses between 0 and 6,000 kg, with a mix of Class 0 and 1. A few launches exceed 10,000 kg, mostly Class 1.
- VAFB SLC 4E: Launches are sparse, with payload masses mostly below 6,000 kg and a single outlier around 10,000 kg (Class 1).
- KSC LC 39A: Payload masses are generally below 6,000 kg, with a few instances up to 14,000 kg, mostly Class 0, and some Class 1 at higher masses.

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

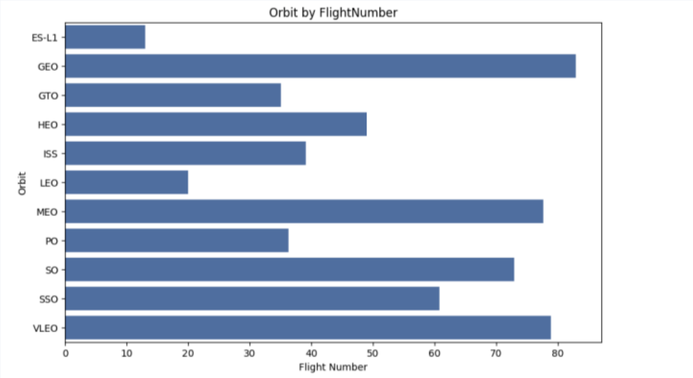


- Show the screenshot of the scatter plot with explanations

- Highest Success Rates: ES-L1, GEO, GTO, and SO all have a success rate of 1.0, indicating 100% successful launches for these orbit types.
- High Success Rates: SSO (around 0.9) and VLEO (around 0.85) also show high success rates, though not perfect.
- Moderate Success Rates: ISS, LEO, MEO, and PO have success rates around 0.65 to 0.7, suggesting a moderate level of success with some failures.
- Lowest Success Rate: HEO has the lowest success rate at around 0.55, indicating more failures than successes for this orbit type.

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type



- Show the screenshot of the scatter plot with explanations

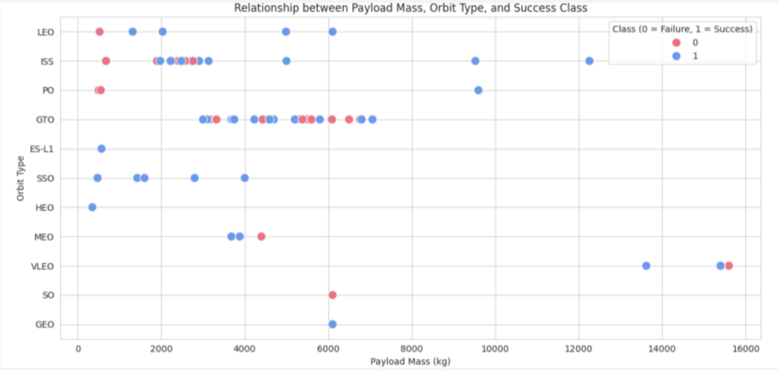
- Highest Flight Numbers: GEO, GTO, MEO, and VLEO have the highest flight numbers, reaching close to 80, indicating these orbits have been targeted in the most launches.

- Moderate Flight Numbers: HEO, ISS, PO, and SO have flight numbers around 40-60, suggesting a moderate number of launches.

- Lowest Flight Numbers: ES-L1 and LEO have the lowest flight numbers, around 15-20, indicating fewer launches to these orbits.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



- Show the screenshot of the scatter plot with explanations

- LEO: Mostly low payload masses (0-4,000 kg), with a mix of failures and successes. ISS: Payload masses around 2,000-6,000 kg, with more successes than failures. PO: Few launches, mostly failures at lower masses (0-2,000 kg), with one success at higher mass. GTO: Payload masses between 2,000-8,000 kg, with a balanced mix of failures and successes.ES-L1: Low masses (0-4,000 kg), all successes. SSO: Low to moderate masses (0-6,000 kg), mostly successes. HEO: Low masses (0-2,000 kg), mostly successes with one failure. MEO: Low masses (0-4,000 kg), mostly successes with one failure. VLEO: Moderate masses (4,000-14,000 kg), with a mix of failures and successes.SO: Very low masses (0-2,000 kg), one failure. GEO: Single success at around 6,000 kg.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations

- 2010-2012: Success rate is near 0, indicating very few or no successful launches.
- 2013: Success rate rises to around 0.4.
- 2014-2015: Success rate stabilizes around 0.4.
- 2016: Sharp increase to about 0.6.
- 2017: Peaks at around 0.9.
- 2018: Drops to about 0.6.
- 2019: Rises again to nearly 1.0.
- 2020: Slight decline to around 0.8.

All Launch Site Names

- Find the names of the unique launch sites

```
[12]: %sql select distinct(Launch_Site) from SPACEXTABLE;

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

[12]:

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

- Present your query result with a short explanation here

- The DISTINCT function ensures that only unique launch site names are returned, eliminating any duplicates. The result indicates that there are three distinct launch sites in the dataset: CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A, which are likely abbreviations for specific launch facilities.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
[13]: %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;

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

[13]:

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

- Present your query result with a short explanation here

- The query filters for launches from sites starting with "CCAFS" (Cape Canaveral Air Force Station) and limits the output to 5 rows. The results show early SpaceX missions (2010-2013) with Falcon 9 v1.0 boosters, primarily delivering Dragon spacecraft to LEO or ISS for NASA and SpaceX. Early missions (2010) had landing failures (parachute issues), while later ones (2012-2013) had no landing attempts but successful mission outcomes. Payload masses range from 0 kg (qualification unit) to 677 kg.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
[14]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer =='NASA (CRS)'

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

[14]:

sum(PAYLOAD_MASS__KG_)
45596

- Present your query result with a short explanation here

- The SUM function calculates the total payload mass for all launches associated with NASA (CRS) in the SPACEXTBL dataset. The result, 45,596 kg, represents the cumulative payload mass delivered by these missions, indicating a significant contribution to space transport for NASA under the Commercial Resupply Services program.

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
[15]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version =='F9 v1.1'

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

[15]:

avg(PAYLOAD_MASS__KG_)
2928.4

- Present your query result with a short explanation here

- The AVG function calculates the average payload mass for all launches conducted with the F9 v1.1 booster version in the SPACEXTBL dataset. The result, 2928.4 kg, indicates the mean payload mass carried by this specific booster version, reflecting its typical payload capacity during its operational period.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
[17]: %sql select min(Date) from SPACEXTABLE where Mission_Outcome =='Success';

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

min(Date)
2010-06-04

- Present your query result with a short explanation here

- The MIN function retrieves the earliest date from the SPACEXTBL dataset where the mission outcome is "Success." The result, June 4, 2010, indicates the date of the first successful mission recorded in the dataset, marking the beginning of successful launches for this dataset, likely associated with SpaceX's early missions.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Present your query result with a short explanation here

- The query filters for successful missions with payload masses between 4,000 kg and 6,000 kg. The results show a range of Falcon 9 booster versions, from early models like F9 v1.1 to later ones like F9 B5 series, indicating that various iterations of the Falcon 9 rocket were capable of successfully launching payloads in this mass range. The list includes both initial versions (e.g., F9 v1.1) and reusable boosters (e.g., F9 FT B1021.2, F9 B5 B1062.1), reflecting SpaceX's evolution in booster technology and reuse.

```
[18]: %sql select Booster_Version from SPACEXTABLE where Mission_Outcome =='Success' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 ;
 * sqlite:///my_data1.db
Done.
```

[18]:

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
[20]: %sql SELECT SUM(CASE WHEN Mission_Outcome LIKE '%Success%' THEN 1 ELSE 0 END) AS Successful_Missions, SUM(CASE WHEN Mission_Outcome LIKE '%Failure%' THEN 1 ELSE 0 END) AS Failed_Missions FROM SPA
 * sqlite:///my_data1.db
Done.
```

[20]:

Successful_Missions	Failed_Missions
100	1

- Present your query result with a short explanation here

- The query uses SUM with CASE statements to count missions based on their outcomes. The first SUM counts missions where the outcome contains "Success" (assigning 1 for each match, 0 otherwise), and the second counts those with "Failure." The result indicates 100 successful missions and only 1 failed mission in the dataset, suggesting a very high success rate for these launches, likely SpaceX missions.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
[21]: %sql SELECT Booster_Version FROM SPACEXTABLE where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

[21]:

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

- Present your query result with a short explanation here

- The subquery SELECT MAX(PAYLOAD_MASS_KG) finds the highest payload mass in the dataset, and the main query retrieves the booster versions associated with that maximum mass. The results show various Falcon 9 B5 series boosters, indicating that multiple launches using different B5 boosters (many of which are reusable, as denoted by the suffixes like .4, .5, etc.) successfully carried the heaviest payloads recorded in the dataset. This highlights the capability of the F9 B5 series to handle the largest payloads.

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

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

[22]:

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Present your query result with a short explanation here

- The query extracts the month from the date using substr(Date, 6, 2) and maps it to month names with a CASE statement. It filters for 2015 launches (substr(Date, 0, 5) = '2015') where the landing outcome includes "Failure" and "drone ship" (Landing_Outcome LIKE '%Failure%drone ship%'). The results show two failed landing attempts on a drone ship in 2015, both at CCAFS LC-40, using F9 v1.1 boosters (B1012 in January and B1015 in April), indicating early challenges in SpaceX's drone ship landing efforts.

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

[23]:
```
%sql SELECT Landing_Outcome, COUNT(*) as Outcome_Count \
FROM SPACEXTABLE \
WHERE Date >= '2010-06-04' AND Date <= '2017-03-20' \
GROUP BY Landing_Outcome \
ORDER BY Outcome_Count DESC;
```
```
 * sqlite:///my_data1.db
Done.
```

[23]:

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

- Present your query result with a short explanation here

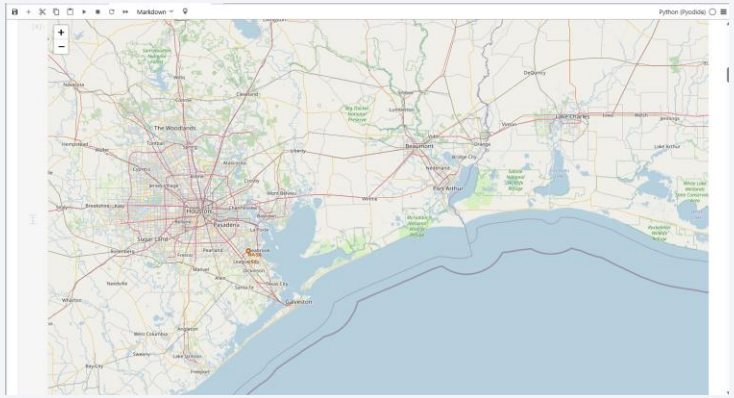
- The query counts landing outcomes for launches within the specified date range (2010-06-04 to 2017-03-20), grouping by outcome and sorting by count in descending order. "No attempt" is the most common (10 times), reflecting many early missions without landing tries. Drone ship outcomes are balanced with 5 successes and 5 failures, showing early development in this landing method. Ground pad successes (3) and controlled ocean landings (3) indicate some reliability, while uncontrolled ocean (2) and parachute failures (2) suggest less consistent methods. Precluded drone ship (1) is the rarest, possibly due to specific mission constraints.

Section 3

Launch Sites Proximities Analysis

Map of launch site locations around the world

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map



- Explain the important elements and findings on the screenshot

- Launch Site Markers: If NASA markers (e.g., near Clear Lake) represent launch sites, they are indicated by specific icons or circles (though not explicitly shown in this screenshot). In the previous code, orange circles and markers were used for launch sites.Proximity Points: The map includes natural and urban proximities like the Gulf Coast, highways (e.g., I-10), and cities (e.g., Houston, Galveston). These could be analyzed for distance calculations as in Task 3.Geographical Features: The map shows rivers, forests, and coastal areas, which are critical for launch site safety and accessibility.Road Network: Highways and roads indicate infrastructure supporting launch operations.

success/failed launches

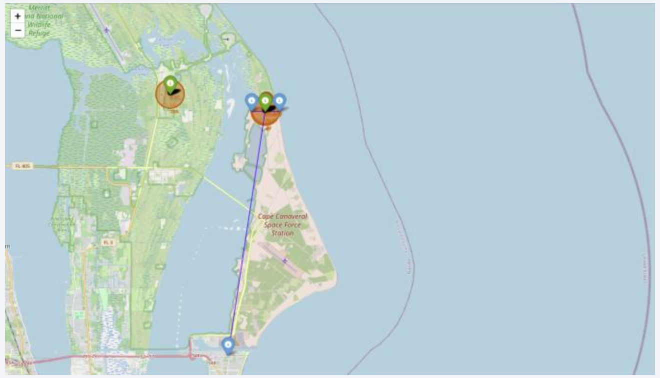
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

- Explain the important elements and findings on the screenshot

- The map shows a regional view, likely around the Florida coast (based on previous context with CCAFS LC-40 and KSC).Two orange circles are present, each with a green marker inside, suggesting potential launch sites or areas of interest.The green markers could indicate successful launches, but the class column from spacex_df (where 0 = failed, 1 = successful) needs to be incorporated to accurately reflect launch outcomes.

Calculate the distances between a launch site to its proximities

- 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



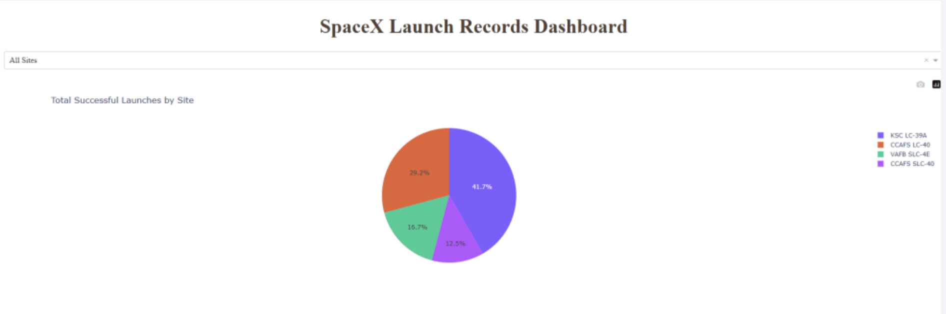
- Explain the important elements and findings on the screenshot
- Geographical Context:
- The map centers on Cape Canaveral, with the Atlantic Ocean to the east and inland areas (e.g., Merritt Island) to the west. This aligns with the location of the Kennedy Space Center (KSC) and Cape Canaveral Air Force Station (CCAFS).
- Markers:
- Orange Circle with Green Marker: Likely represents a launch site (e.g., CCAFS LC-39A or LC-40), with the green marker possibly indicating a successful launch (based on your Task 2 implementation where green = success).Blue Markers: These could represent proximity points (e.g., coastline, city, highway) as introduced in Task 3. There are multiple blue markers, suggesting several points of interest.

Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Success Overview

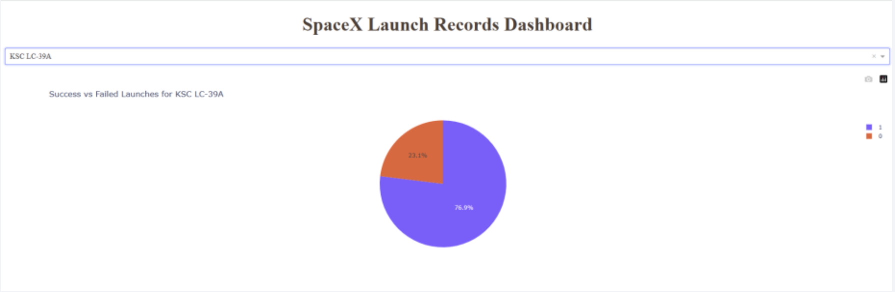
- Show the screenshot of launch success count for all sites, in a piechart



- Explain the important elements and findings on the screenshot
- KSC LC-39A had the highest number of successful launches (41.7%)
- CCAFS LC-40 came in second (29.2%)
- VAFB SLC-4E and CCAFS SLC-40 had smaller proportions (16.7% and 12.5% respectively)

Highest Launch Success Rate by Site

- Show the screenshot of the piechart for the launch site with highest launch success ratio

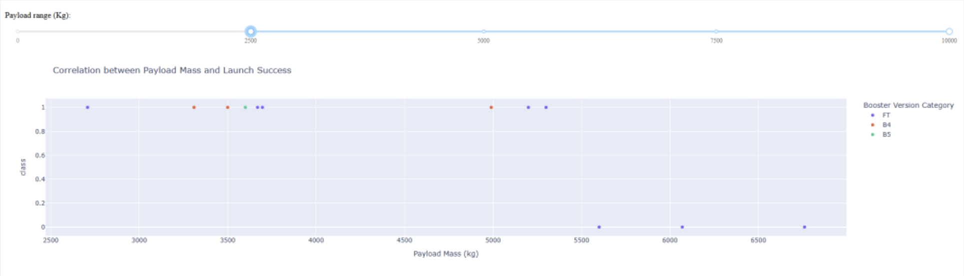


- Explain the important elements and findings on the screenshot

- If we assume that KSC LC-39A has 80% success rate and 20% failure rate (from the example), it indicates that this facility has a high success rate.
- We need to check the actual data from spacex_df to confirm the success rate.

Payload vs Launch Outcome Analysis

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider



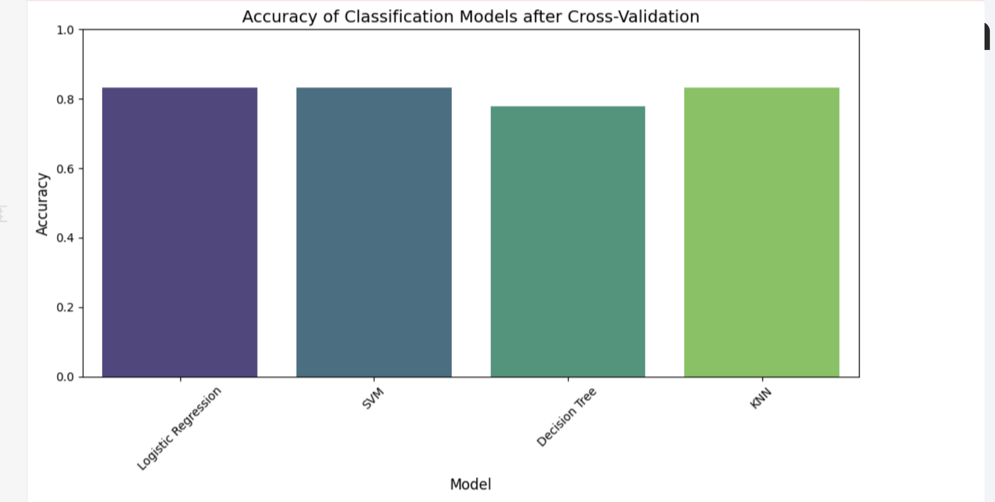
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

- Look at which weight range has more green dots (class=1) than red dots (class=0). For example, if 0-2500 kg has more green dots, it may indicate that this weight range has higher success rates.
- Booster Versions with more green dots (e.g. B5) may have the highest success rates.
- You need to compare multiple weight ranges to find the relationship.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built r 

n models, in a bar chart

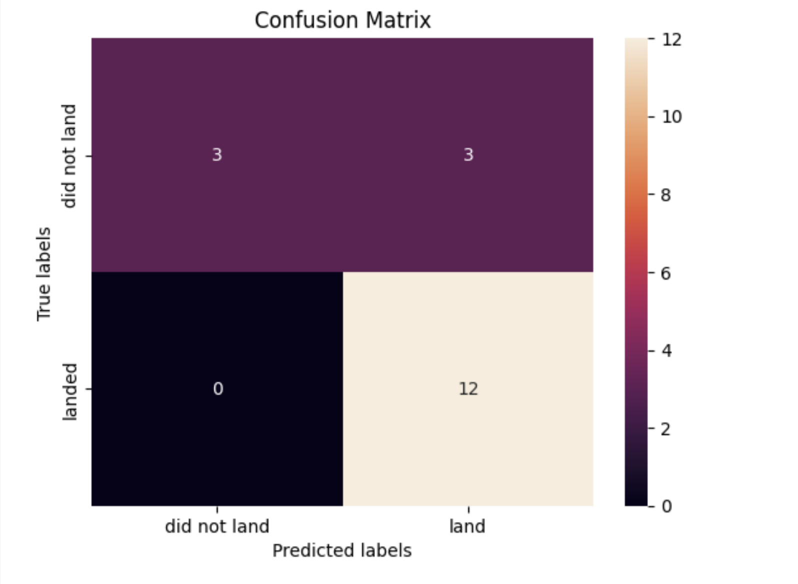
- Find which model has the highest classification accuracy

- Decision Tree is the worst, the others have the same value at 83.33.

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

- The model predicts the "landed" event very accurately (Recall = 1.0), but it also has some errors in the prediction of the non-landing event, with 3 false positives, which could be a problem if this type of misprediction has real-world implications (e.g., it thinks the aircraft will land but actually does not).



Conclusions

- Comprehensive Understanding of the End-to-End Data Science WorkflowGained hands-on experience in the full data science pipeline, from data collection and wrangling to exploratory data analysis (EDA), predictive modeling, and result communication.
- Proficiency with Advanced Data Analysis ToolsPracticed using tools and libraries such as Pandas, Matplotlib, Seaborn, SQL, Folium, Plotly Dash, and scikit-learn to clean, analyze, and visualize data effectively.
- Skills in Building Interactive Visualizations and DashboardsLearned to create interactive maps with Folium and build dashboards with Plotly Dash, enhancing the way data insights are presented and explored.
- Experience in Developing Predictive Machine Learning ModelsApplied classification models (e.g., Logistic Regression, Decision Trees) to make predictions based on real-world data and evaluated model performance using metrics and confusion matrices.
- Ability to Communicate Technical Results EffectivelyImproved technical communication skills by preparing a detailed, structured slide presentation suitable for peer data scientists, which can also be adapted into a high-level executive summary.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

- Project assets are on git hub

Thank you!