Winning Space Race With Data Science

Aaron Goldman

10/18/24

Outline

Executive Summary

- **Summary of methodologies**
- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

- **Summary of all results**
- EDA results
- Interactive analytics
- Predictive analysis

Introduction

Project background and context

SpaceX promotes its Falcon 9 rocket launches on its website at a price of $62 million. In comparison, other providers charge over $165 million for similar launches. A significant portion of SpaceX's cost savings comes from its ability to reuse the first stage of the rocket. If we can accurately predict whether the first stage will land successfully, we can better estimate the overall cost of a launch.

The key problems we want to address in analyzing Falcon 9 rocket launches include:

Predicting landing success by identifying the factors that affect the Falcon 9 first stage landing and creating a model for accurate predictions. Also, we aim to understand how first stage landing success influences the overall launch cost and the savings achieved through reusability. A competitive analysis will explore how SpaceX′s pricing compares to other providers and what strategies alternative companies might adopt to compete effectively. We will also examine how landing predictions impact launch pricing strategies and identify trends in pricing between SpaceX and its competitors. Finally, we want to figure out effective methods for collecting data on launch outcomes and how to apply machine learning techniques to analyze this data.

Section 1

Methodology

Data Collection API

The objective was to collect historical launch data for Falcon 9 and Falcon Heavy rockets using web scraping techniques.
Key steps included:

Web Scraping:

Used BeautifulSoup to extract data from the Wikipedia page, "List of Falcon 9 and Falcon Heavy launches."
Parsed the HTML table containing launch records, including details such as launch dates, payload, and landing outcomes.

Extracting Column Names:

Identified the relevant table and extracted column headers from the HTML table using a helper function extract_column_from_header().

Data Parsing and Cleaning:

Created a dictionary launch_dict to store the scraped data and populated it by iterating through table rows.
Handled common issues such as missing values, inconsistent formatting, and annotations.

Data Conversion:

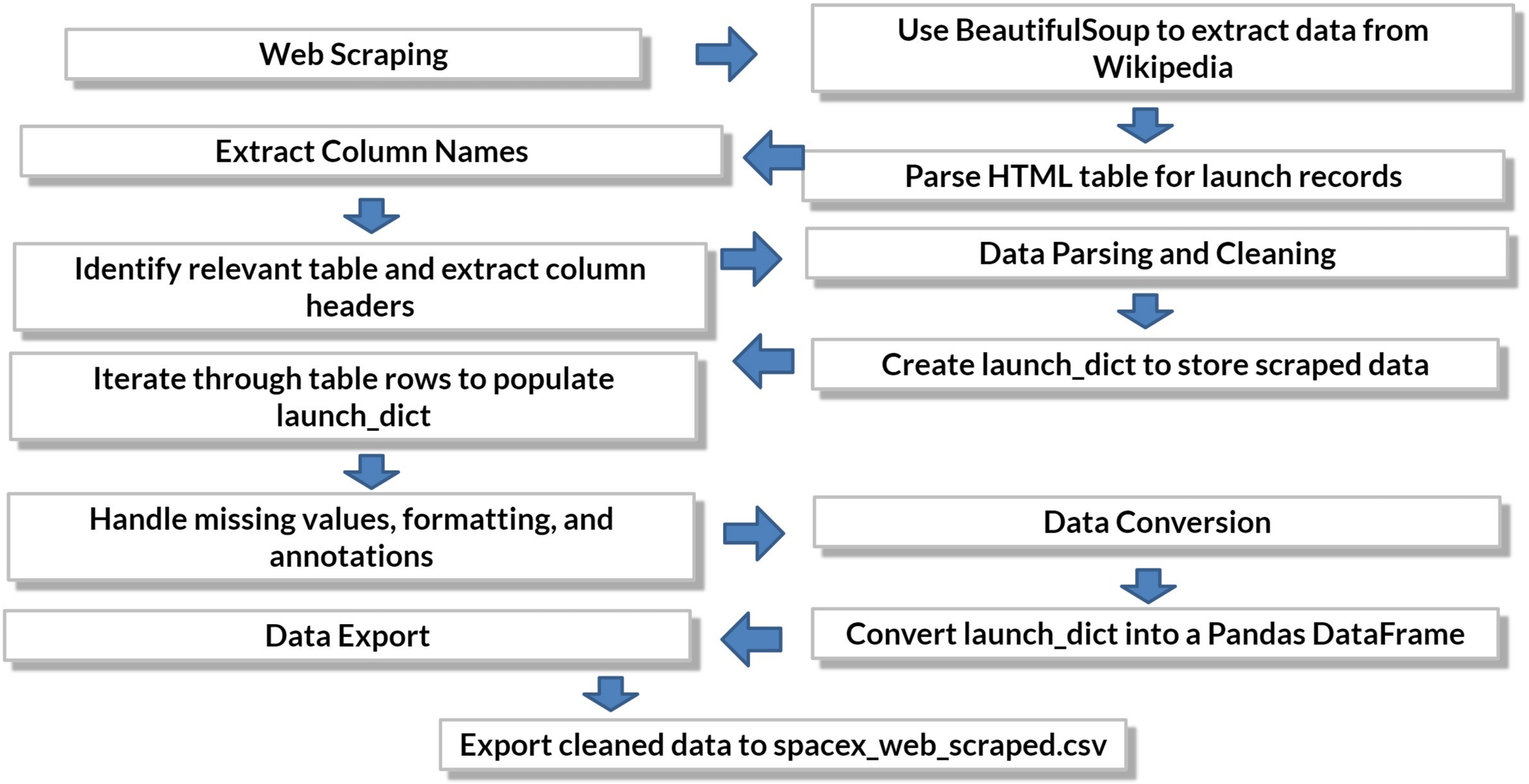
Converted the parsed data into a Pandas DataFrame for easy manipulation and analysis.

Data Export:

Exported the cleaned data to a CSV file, spacex_web_scraped.csv, to be used in future labs for further analysis and machine learning tasks.
This methodology ensured that the Falcon 9 launch data was collected, cleaned, and prepared for further analysis.

GITHUB LINK
https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Data Collection-API

Data Collection API Flowchart



Data Collection - Scraping

Key Steps

Web Scraping:

Objective: Scrape Falcon 9 and Falcon Heavy launch records from a Wikipedia page using BeautifulSoup and Requests libraries.

Process:

Request the HTML content from the Wikipedia page using requests.get().

Parse the HTML table containing launch data using BeautifulSoup.

Extracting Data:

Column Extraction: Use helper functions like extract_column_from_header() to retrieve the column names from the table headers.

Data Parsing: Develop functions (date_time(), booster_version(), landing_status(), etc.) to extract specific information from the table rows such as launch date, booster version, landing outcome, and payload mass.

Data Cleaning:

Handle missing values, inconsistent formatting, and references (e.g., annotations in the table such as B0004.1[8]).

Normalize strings and numeric data (like mass in kg) to ensure uniform formatting.

Data Conversion:

Store the parsed launch data in a dictionary launch_dict.

Convert the dictionary into a Pandas DataFrame for further manipulation and analysis.

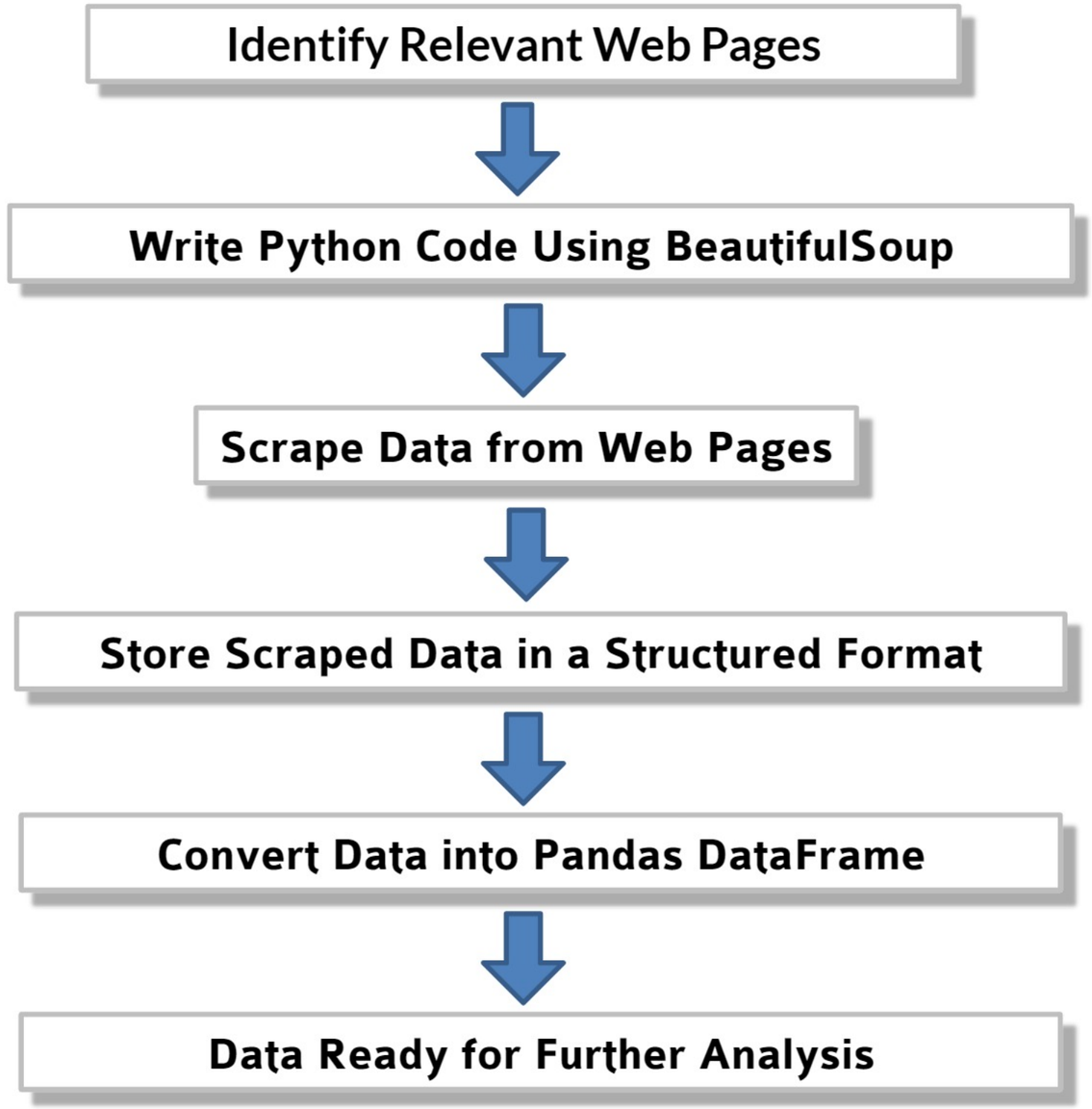
Data Export:

Once the data is parsed and cleaned, export it to a CSV file (spacex_web_scraped.csv) for use in subsequent tasks, such as machine learning analysis.

GITHUB LINK

https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Web-Scraping

Data Collection with Web Scraping Flowchart



Data Wrangling

The primary goal was to perform Exploratory Data Analysis (EDA) on SpaceX's Falcon 9 launch data and generate binary labels for supervised machine learning, where 1 indicated a successful landing and 0 represented a failure.

Key Steps:

Data Loading and Exploration:

Loaded the Falcon 9 dataset and identified both numerical and categorical columns.

Examined the percentage of missing values across all attributes.

EDA and Labeling:

Converted various landing outcomes (e.g., True/False Ocean, ASDS, RTLS) into binary labels (success/failure).

Created the landing_class variable to serve as the classification label.

Launch Site and Orbit Analysis:

Used the value_counts() method to calculate the number of launches per site and by orbit type (e.g., LEO, GTO, SSO).

Calculated the occurrence of mission outcomes based on orbit types.

Mission Outcome Labeling:

Created a landing outcome label by setting 0 for unsuccessful landings and 1 for successful landings.

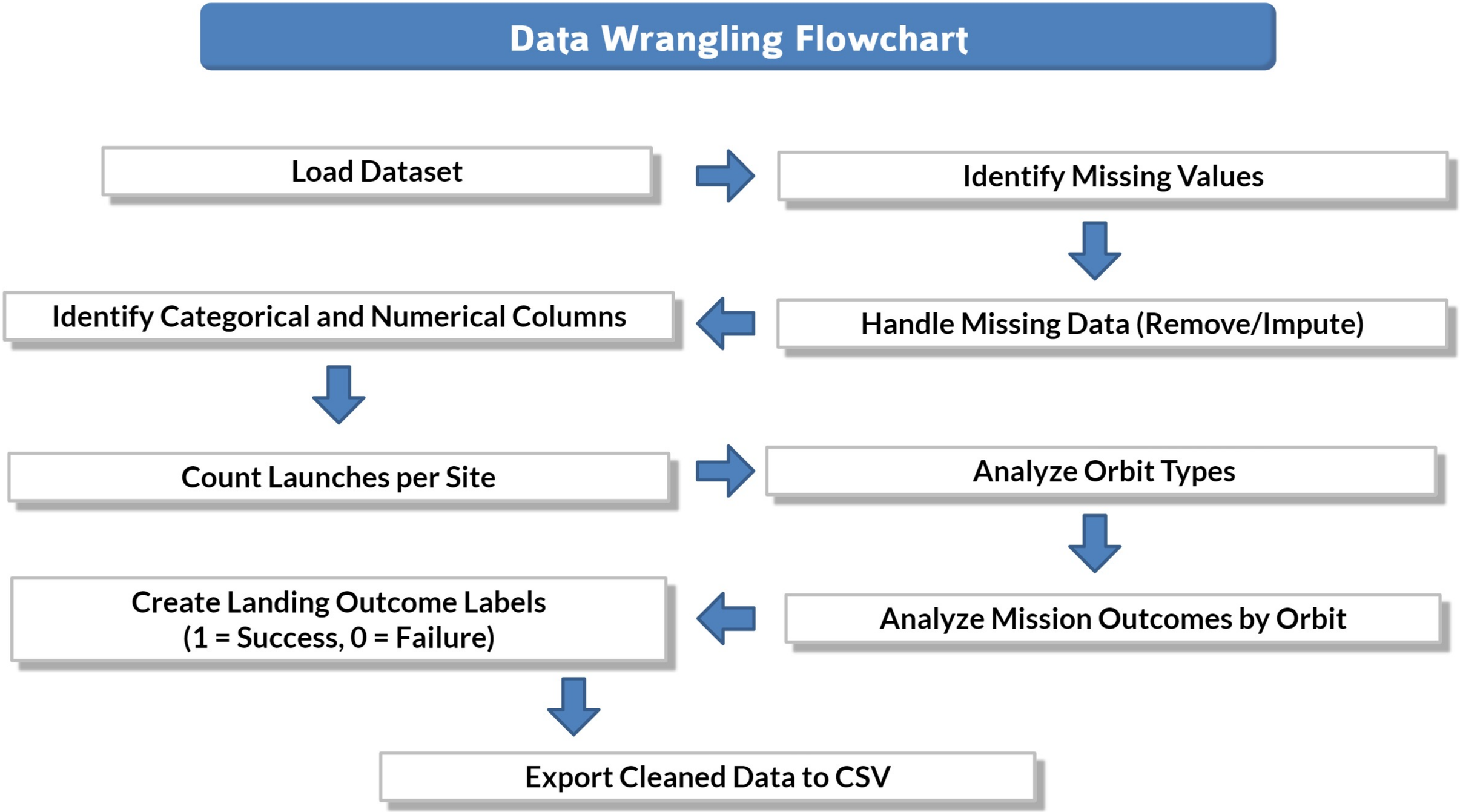
Data Export:

The cleaned and labeled dataset was exported as dataset_part_2.csv for use in future analysis and model training.

This process ensured the data was ready for machine learning model development.

GITHUB LINK
https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Data-Wrangling

Data Wrangling Flowchart



Exploratory Data Analysis (EDA) with data visualization.

The goal was to Analyze SpaceX's Falcon 9 launch data to predict the success of first stage landings through Exploratory Data Analysis (EDA) and feature engineering.

Dataset Overview: The dataset includes historical records of Falcon 9 launches, emphasizing the cost-effectiveness of SpaceX's reusable rockets.

Data Acquisition:

Load the SpaceX dataset into a Pandas DataFrame for analysis.

Exploratory Data Analysis (EDA):

Flight Number vs. Payload Mass: Create scatter plots to visualize trends in landing success related to flight numbers and payload weight.

Task 1: Use catplot to visualize Flight Number against Launch Site with success classifications.

Task 2: Scatter plot of Payload Mass vs. Launch Site shows that some sites have not launched heavy payloads.

Task 3: Bar chart of success rates by orbit type identifies the highest success rates.

Task 4: Analyze Flight Number vs. Orbit type to find correlations in LEO and GTO.

Task 5: Scatter plot of Payload Mass vs. Orbit Type assesses landing success rates across various orbits.

Task 6: Line chart shows average success rates by year.

Feature Engineering:

Task 7: Convert categorical columns into dummy variables for better model training.

Task 8: Cast numeric columns to float64 for consistency.

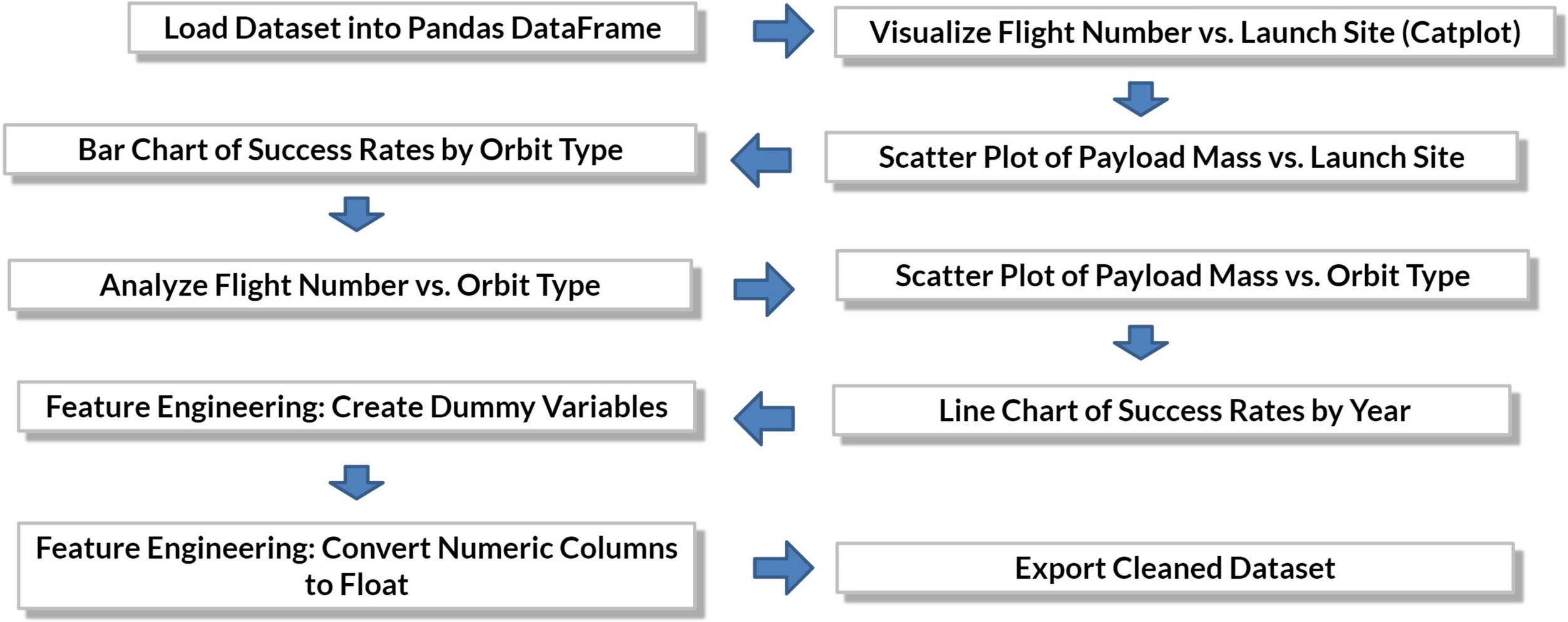
Data Export:

Export the cleaned dataset as dataset_part_3.csv for future analysis and model training.

This methodology prepares the dataset for predictive modeling, facilitating accurate forecasts of Falcon 9 first stage landings.

GITHUB LINK
https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Data-Analysis-EDA

Exploratory Data Analysis (EDA) with data visualization Flowchart

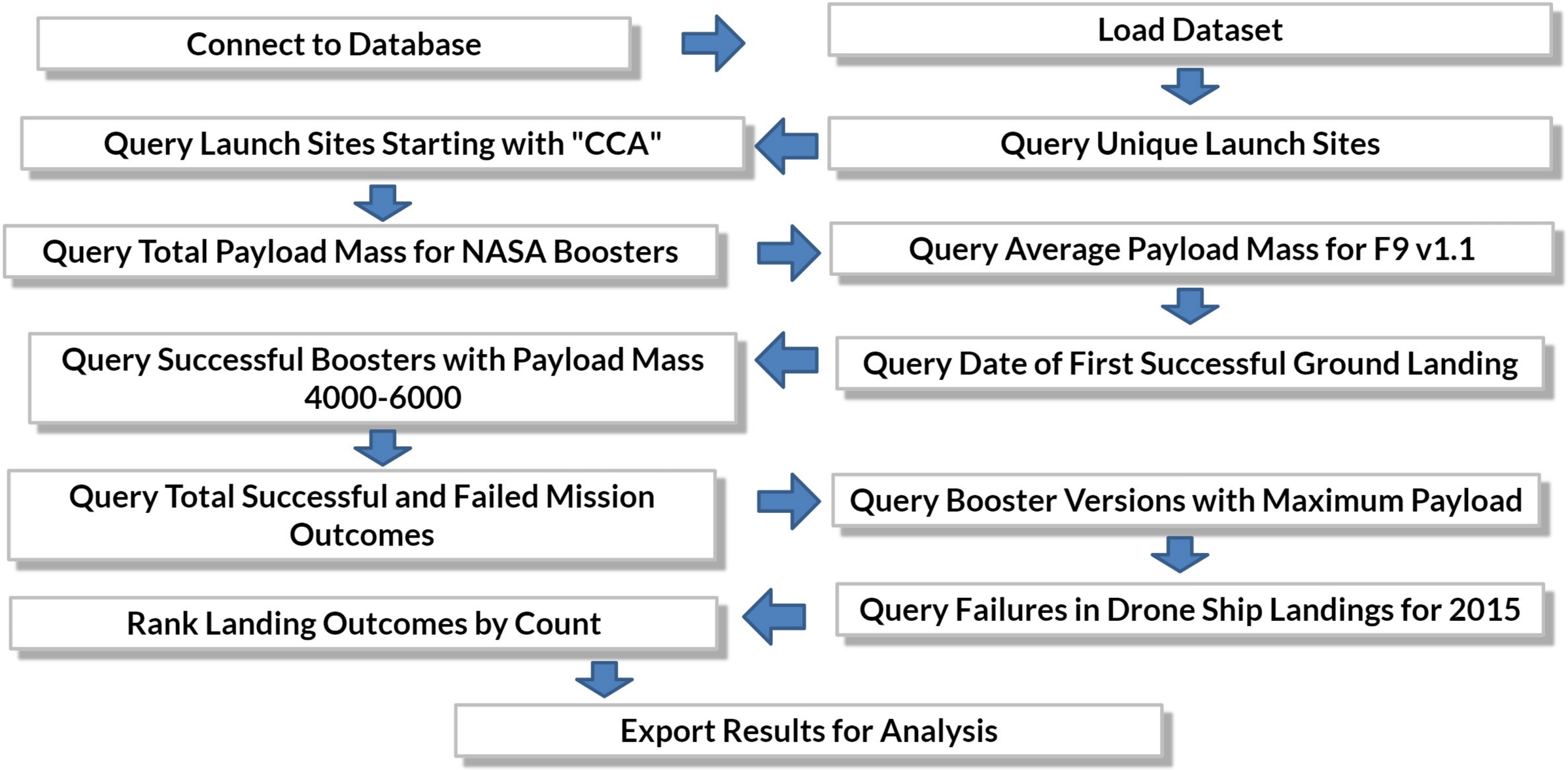


Exploratory Data Analysis (EDA) process using SQL

The goal was to Analyze SpaceX's Falcon 9 dataset to evaluate landing success and its influence on launch costs.
Dataset Overview: The dataset includes records of payloads from SpaceX missions, highlighting milestones in private space exploration.
Data Acquisition: The .CSV file was downloaded for analysis.
Data Connection: A SQL extension was used to establish a database connection for query execution.
Data Cleaning: Blank rows were removed to ensure data integrity.
Tasks and SQL Queries:
Extract unique launch sites.
Identify launch sites starting with "CCA."
Calculate total payload mass for NASA boosters.
Determine average payload mass for booster version F9 v1.1.
List the date of the first successful ground landing.
Identify boosters with successful drone ship landings and payloads between 4000 and 6000.
Count total successful and failed mission outcomes.
Find booster versions with maximum payload mass using a subquery.
Extract records of landing outcomes on drone ships for 2015.
Rank landing outcomes by count for missions from June 4, 2010, to March 20, 2017.
Data Export: Results were compiled for further analysis and decision-making.

GITHUB LINK
https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Data-Analysis-SQL

Exploratory Data Analysis (EDA) process using SQL Flowchart



Building an interactive map with Folium

The goal was to Analyze geographical factors affecting SpaceX Falcon 9 launch success rates using Folium for interactive visualization.

Key Tasks:

1.**Map Creation**: Load the dataset with latitude and longitude for launch sites, centering the Folium map on the NASA Johnson Space Center.

2.**Marking Launch Sites**:

Highlight each launch site using folium.Circle and add detailed markers with folium.Marker.

Visualizing Launch Outcomes:

Color-code markers based on launch success (green for success, red for failure) and utilize MarkerCluster to manage overlapping markers.

Proximity Analysis:

Use MousePosition to find coordinates of key locations (e.g., coastlines, cities) and draw distance lines using folium.PolyLine.

Findings:

Assess the proximity of launch sites to infrastructure (railways, highways) and urban areas, analyzing how these factors influence launch operations.

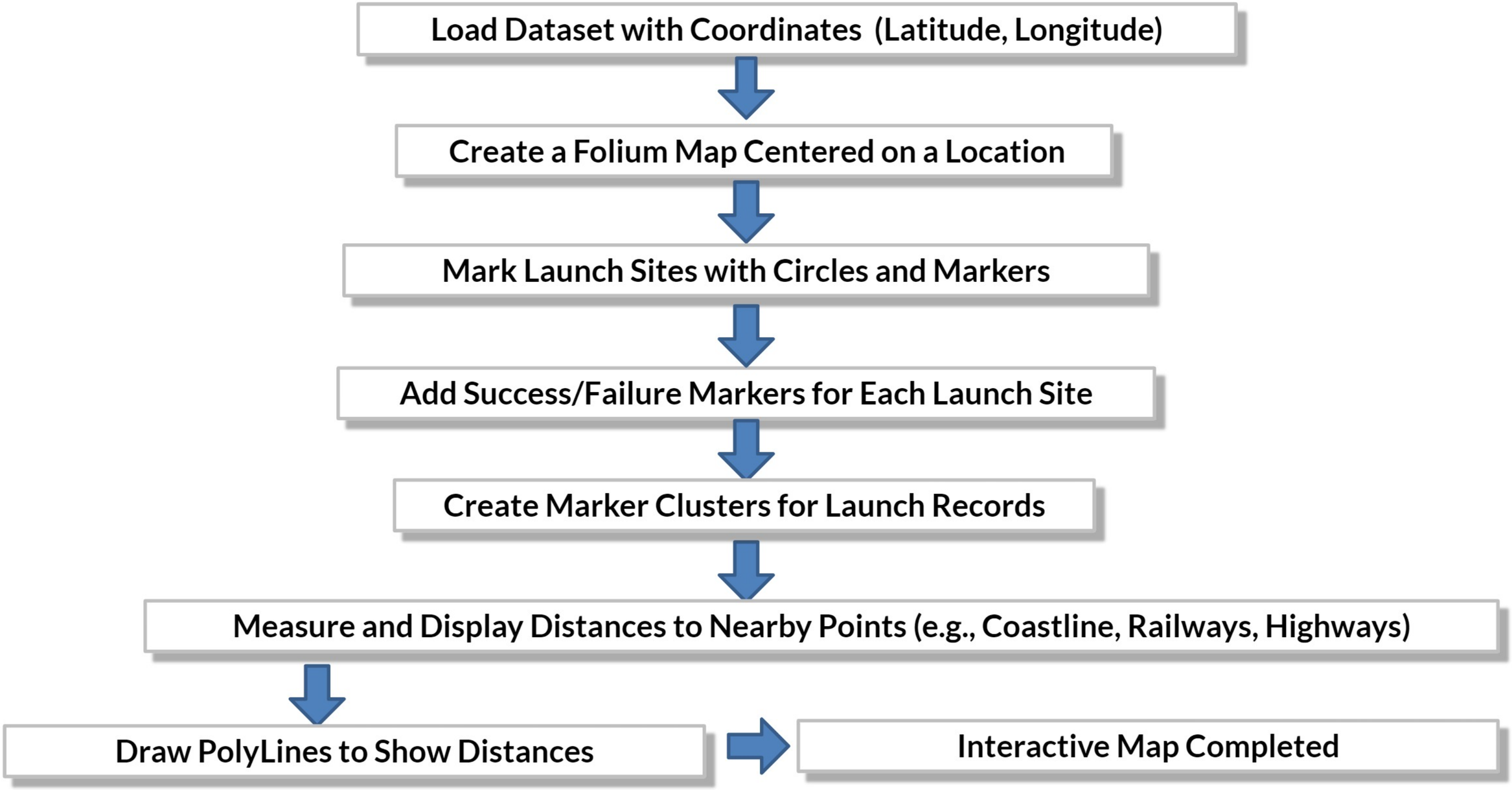
Data Export:

Document insights and export the annotated map for further analysis.

This methodology provided a framework for exploring spatial relationships related to launch success rates, aiding in the evaluation of optimal launch site locations.

GITHUB LINK
https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Folium

Building an interactive map with Folium flowchart



Building a Dashboard with Plotly Dash

The goal was to Create an interactive Plotly Dash application to visualize SpaceX launch data, allowing to analyze success rates based on launch site and payload mass in real-time.

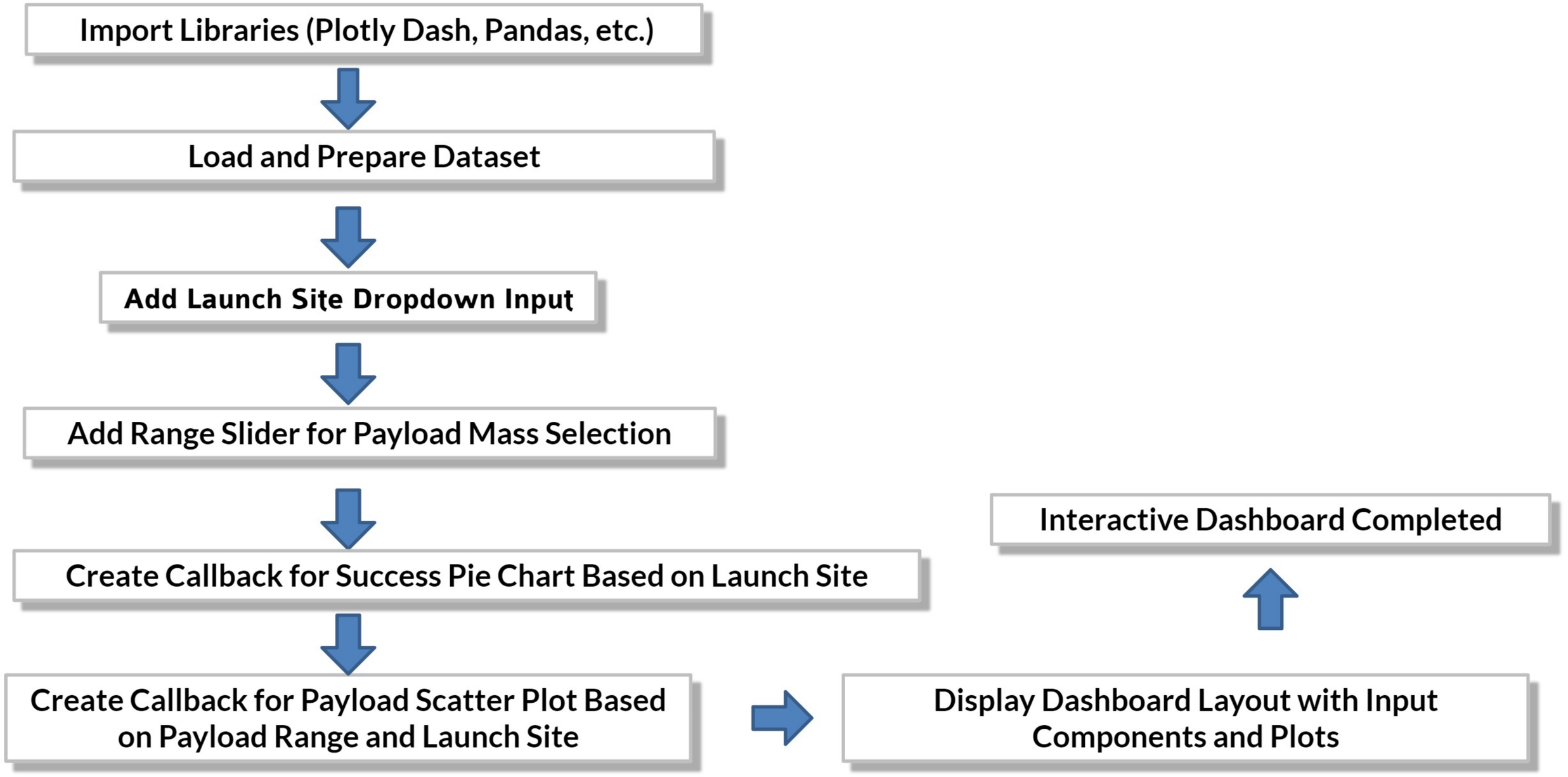
Key Tasks:

Set Up Environment:
Import necessary libraries, including Dash and Plotly, to create the web application.
Create Input Components:
Launch Site Dropdown: Implement a dropdown list for user to select a launch site.
Range Slider: Add a slider to allow user to select a range of payload masses for analysis.
Callback Functions:
Pie Chart Rendering: Develop a callback function that updates a pie chart showing launch success rates based on the selected launch site from the dropdown.
Scatter Plot Rendering: Create another callback function to render a scatter plot that displays the relationship between payload mass and launch success, adjusted according to the selected payload range from the slider.

This methodology provided a structured approach to building a dynamic data visualization dashboard, enhancing users' ability to analyze SpaceX launch data interactively.

GITHUB LINK
https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Plotly-Dash

Building a Dashboard with Plotly Dash Flowchart



Predictive analysis (Classification)

The goal is to develop a machine learning pipeline that predicts whether the Falcon 9 first stage will successfully land, using historical launch data.

Key Steps:

Data Preparation:

Load the SpaceX dataset and create a target variable, Y, by extracting the class labels.

Standardize feature data, X, for improved model performance.

Data Splitting:

Use train_test_split to divide the dataset into training and testing sets, allocating 80% for training and 20% for testing.

Model Training and Hyperparameter Tuning:

Logistic Regression: Create and tune a logistic regression model using GridSearchCV to find optimal parameters from a predefined parameter grid.

Support Vector Machine (SVM): Implement and optimize an SVM model similarly.

Decision Tree Classifier: Construct and fine-tune a decision tree classifier using grid search.

K-Nearest Neighbors (KNN): Develop a KNN model and adjust hyperparameters.

Model Evaluation:

Assess each model's accuracy on the test dataset using the score method.

Generate and examine confusion matrices for each model to analyze prediction performance.

Performance Comparison:

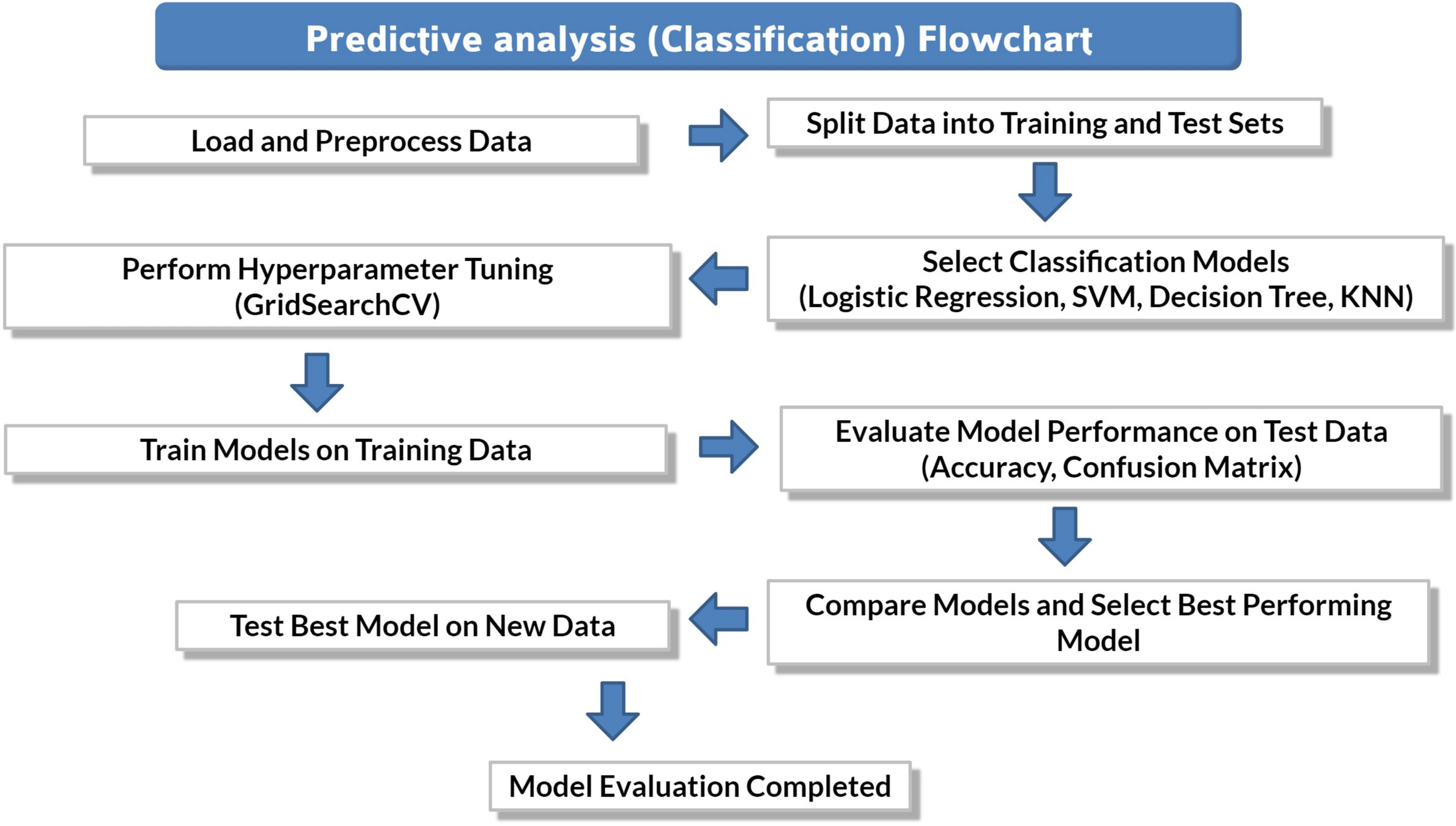
Compare the accuracy and confusion matrices of all models to determine the most effective algorithm for predicting landing success.

This structured methodology guided the process of preparing data, training various machine learning models, and evaluating their predictive capabilities for SpaceX Falcon 9 landing outcome.

GITHUB LINK

https://github.com/AaronKGoldman/Data-Science-Capstone-IBM-Predictive-Analysys

Predictive analysis (Classification) Flowchart



Results

Exploratory data analysis results

Flight Experience: Success rates improve with more flight attempts, showing that experience boosts performance.

Flight Number vs. Success Rate: Success improves with flight numbers at all launch sites, with VAFB SLC 4E reaching 100% after 50 flights, and KSC LC 39A and CCAFS SLC 40 after 80 flights.
Payload vs. Launch Site: No heavy payloads (over 10,000 kg) have been launched from VAFB-SLC.
Orbit Success Rates: ES-L1, GEO, HEO, and SSO have 100% success; GTO, ISS, LEO, MEO, and PO range from 50% to 80%, while SO has 0%.
Flight Number vs. Orbit: Success rises with flight numbers for most orbits, especially LEO, but not for GTO.
Yearly Trends: Success rates improved from 2013–2017 and 2018–2019, with slight declines in 2017–2018 and 2019–2020. Overall, success has risen since 2013.

These results show factors that influence landing success, helping future mission planning and predictive modeling.

Results

Interactive Analytics Demo Results in Screenshots

Launch Sites Near Equator & Coastlines

Map: Launch sites are shown near the equator and coastlines.
Result: Sites benefit from Earth's rotation for efficient launches and are positioned near oceans for safety.
Flight Path & Coast Proximity

Map: A line connects launch sites to the coast.
Result: Demonstrates the strategic positioning to ensure safe ocean-bound flight paths.

Fuel Efficiency
Graph: Shows fuel savings from launching near the equator.
Result: Launches from the equator save fuel and boost success rates.

Safety Zones
Map: Circular zones around launch sites.
Result: Highlights safe ocean zones to minimize risks to populated areas.

Results

Predictive analysis results

Model Performance Comparison

Logistic Regression: Accuracy of 83.3% on test data.
SVM: Accuracy of 83.3%, matching Logistic Regression.
Decision Tree: Similar accuracy of 86.7%.
KNN: Also achieved 83.3% accuracy.

Best Model: Decision Tree

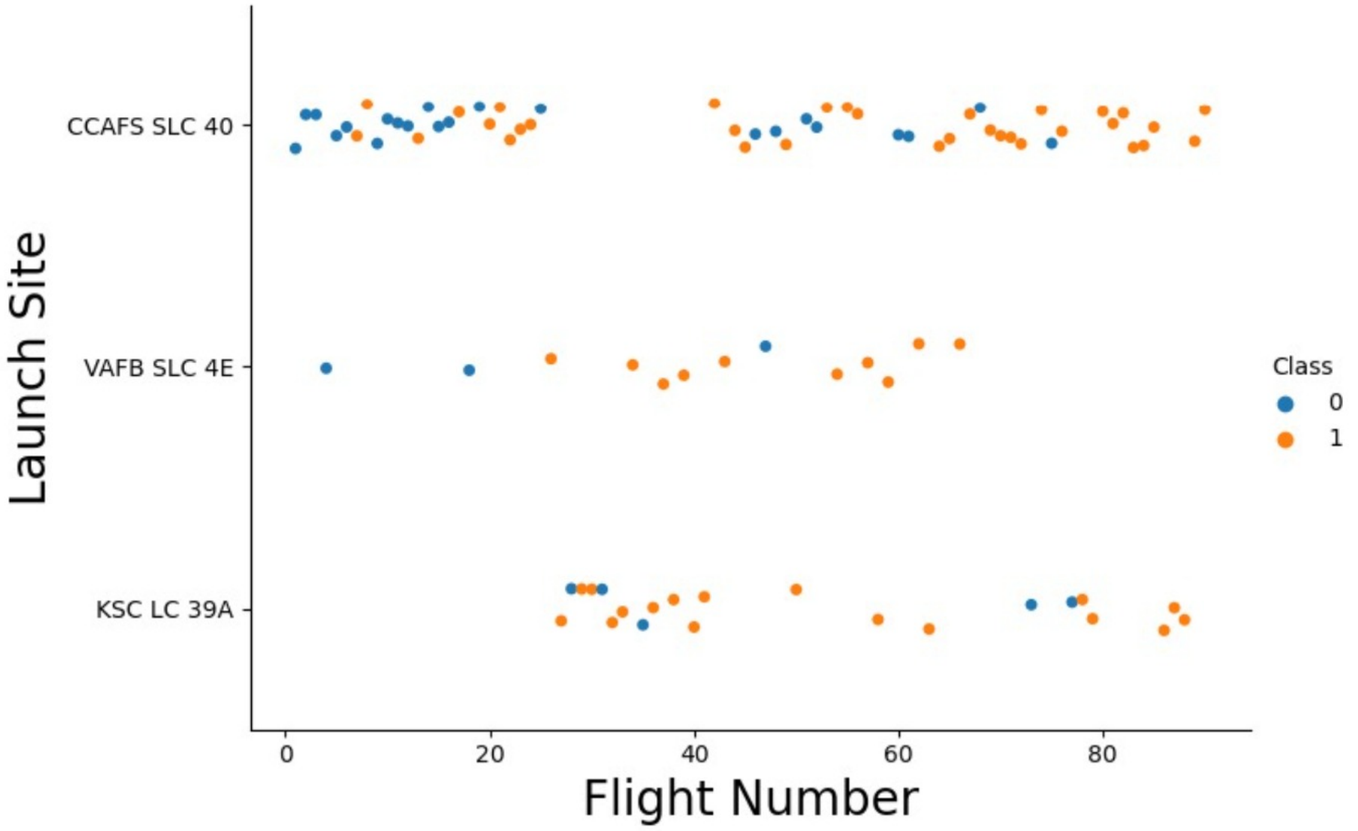
The **Decision Tree model** achieved the highest accuracy, 86.7%, outperforming the other models (Logistic Regression, SVM, and KNN) which each reached 83.3%.

This makes the Decision Tree model the best choice for predicting the success of Falcon 9 first stage landings based on the available data. It effectively balances complexity and predictive power, making it a strong candidate for further refinement and deployment.

Section 2

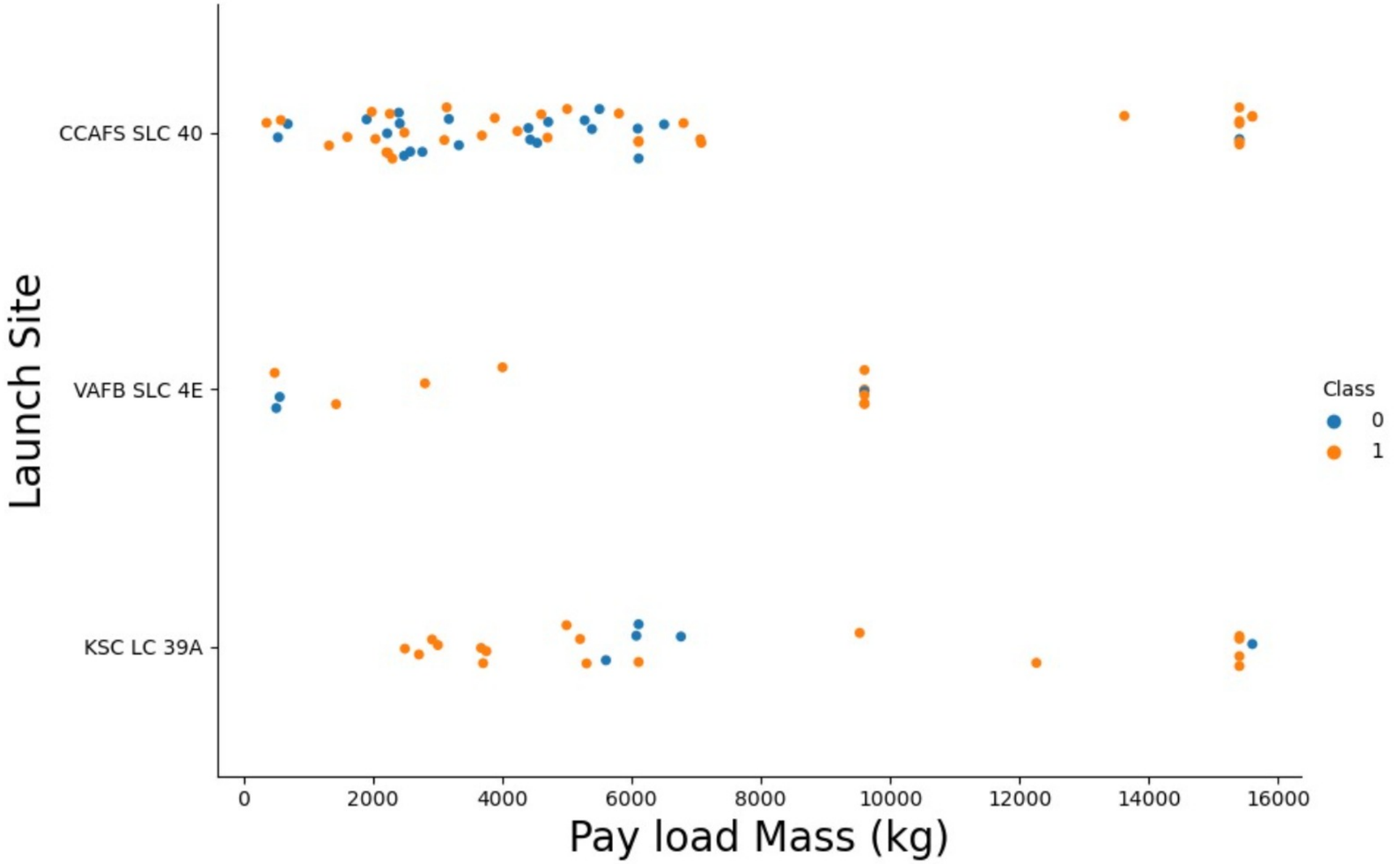
Insights drawn from EDA

Flight Number vs. Launch Site



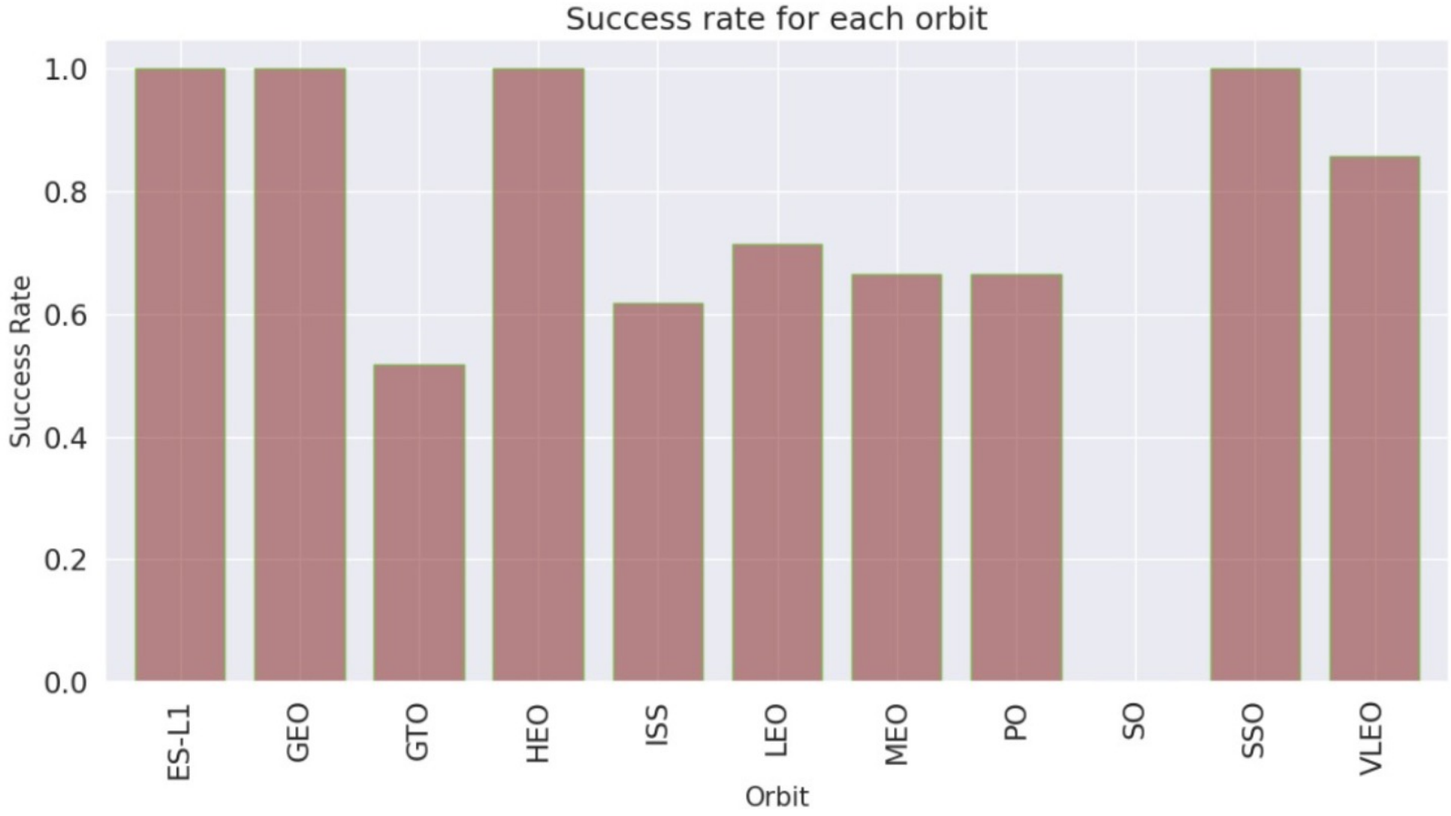
It can be concluded that as the flight number increases at each of the three launch sites, the success rate also improves. The VAFB SLC 4E launch site achieved a 100% success rate after the 50th flight, while both KSC LC 39A and CCAFS SLC 40 reached 100% success after the 80th flight.

Payload vs. Launch Site



Looking at the scatter plot of Payload vs. Launch Site, it is evident that no rockets with heavy payloads (greater than 10,000) have been launched from the VAFB-SLC site.

Success Rate vs. Orbit Type



This bar chart shows :

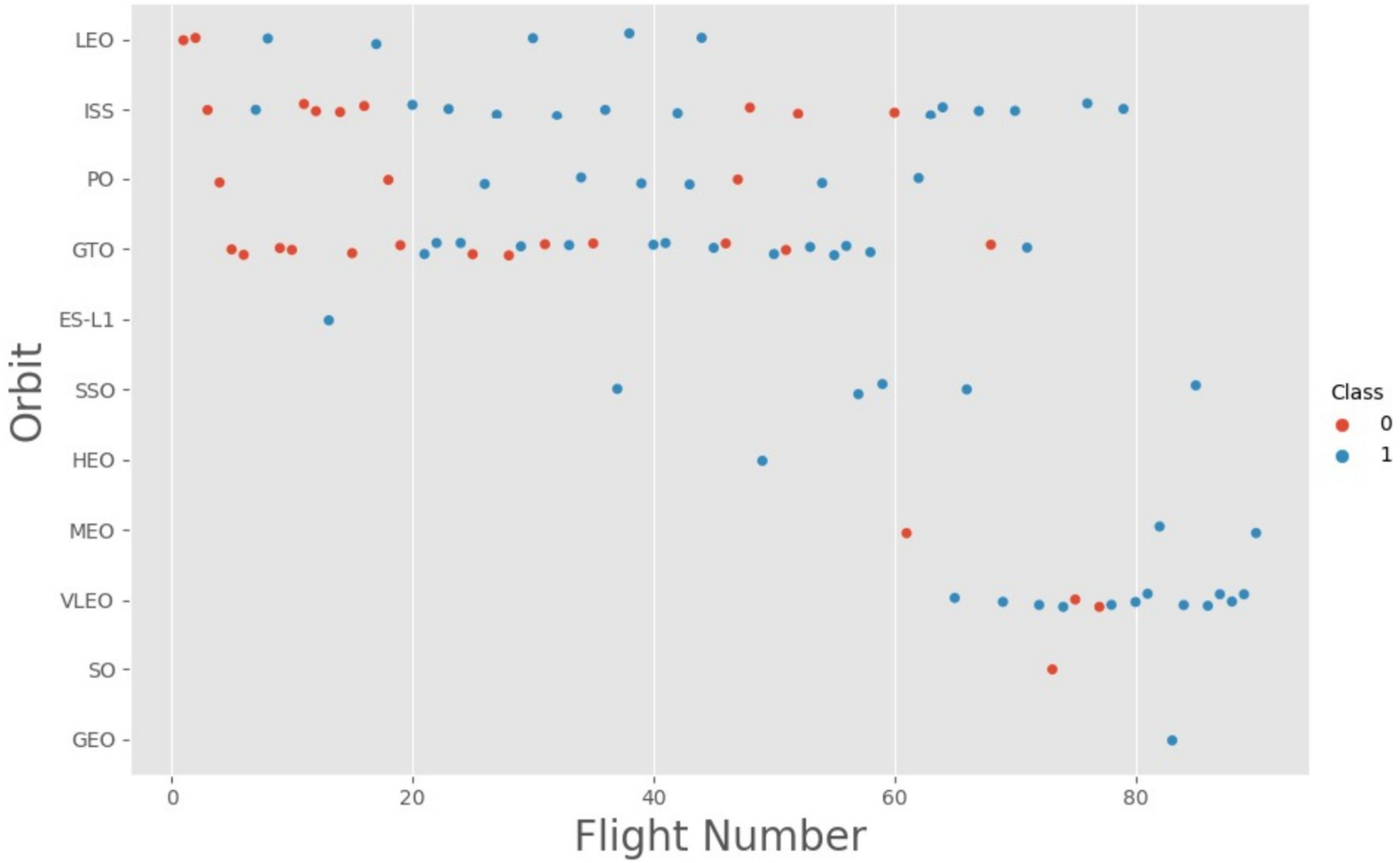
ES L1, GEO, HEO and SSO: **100% Success Rate**

GTO, ISS, LEO, MEO, PO : **50% 80% Success Rate**

SO: **0% Success Rate**

It illustrates the success rates of different orbits for Falcon 9 launches. ES-L1, GEO, HEO, and SSO orbits achieved a perfect 100% success rate. GTO, ISS, LEO, MEO, and PO orbits had success rates ranging between 50% and 80%. SO orbit had a 0% success rate.

Flight Number vs. Orbit Type

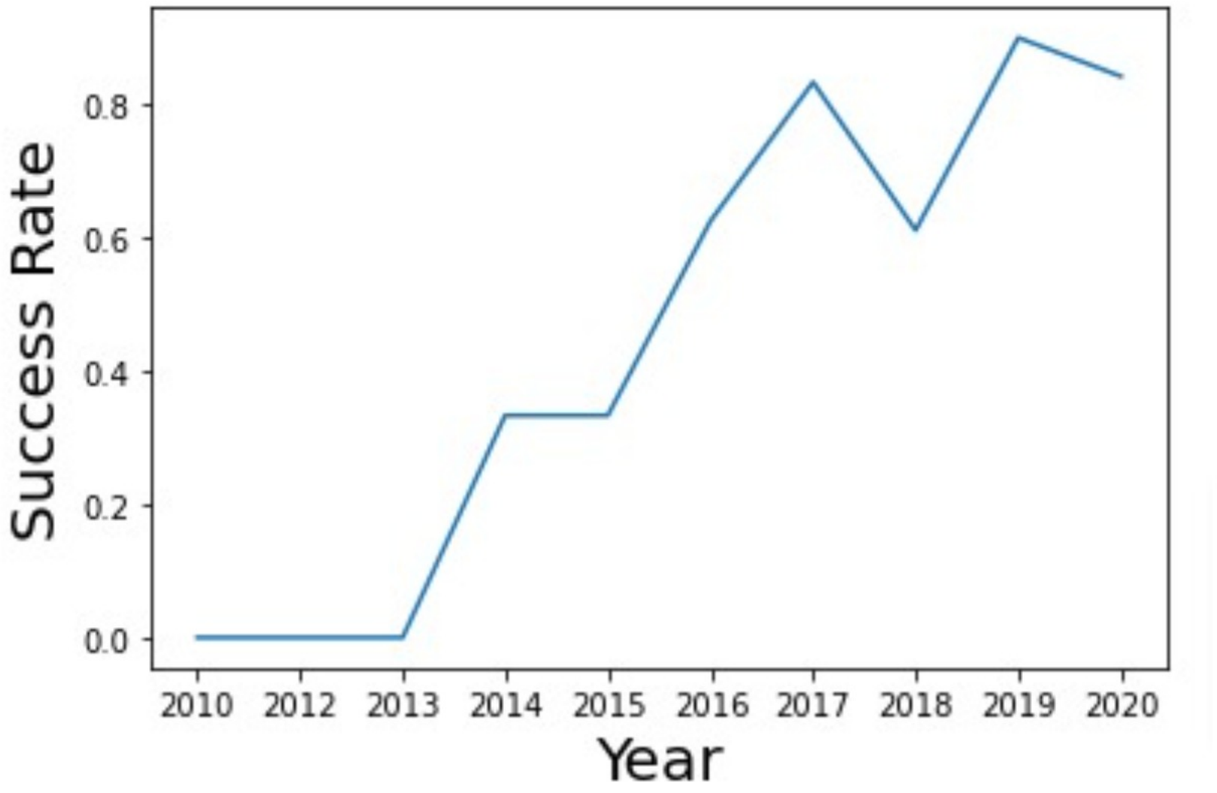


The success rate tends to improve with the increase in flight numbers across most orbits, particularly in the LEO orbit. However, this pattern does not hold for the GTO orbit.

Payload vs. Orbit Type

This scatter plot shows the relationship between payload mass and orbit type, with the hue indicating successful (positive) and unsuccessful (negative) landings.
Heavier payloads tend to have more successful landings in Polar, LEO, and ISS orbits.
For GTO orbits, the results are mixed, with both successful and unsuccessful landings, making it harder to distinguish a clear trend.

Launch Success Yearly Trend



The success rate increased from 2013 to 2017 and again from 2018 to 2019. However, there was a decline between 2017 and 2018, and another between 2019 and 2020. Despite these variations, the overall trend has been an upward improvement since 2013.

All Launch Site Names

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

Display the names of the unique launch sites in the space mission

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

Explanation:
The query ---> **%sql select distinct launch_site from SPACEXTBL;**
is used to retrieve all unique launch site names from the SPACEXTBL table.
By using the DISTINCT keyword, the query ensures that only unique launch site names are returned, eliminating any duplicate entries from the result set. This allows us to see a list of all different launch sites recorded in the SPACEXTBL table.

Launch Site Names Begin with 'CCA'

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

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

Explanation:
select *: Retrieves all columns from the table.
from SPACEXTBL: Specifies the table SPACEXTBL from which to retrieve data.
where LAUNCH_SITE like 'CCA%': Filters the results to only include rows where the LAUNCH_SITE column starts with 'CCA'. The % is a wildcard that represents any sequence of characters following 'CCA'.
limit 5: Restricts the result to the first 5 rows that match the criteria.
This query was used to get the first 5 rows from the SPACEXTBL table where the launch site names start with "CCA".

Total Payload Mass

sum(PAYLOAD_MASS__KG_)
45596

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

Explanation:
%sql: This command is used to run SQL queries in a Jupyter notebook.
SELECT SUM(PAYLOAD_MASS__KG_): This selects the sum (total) of the PAYLOAD_MASS__KG_ column, which likely represents the total weight of payloads in kilograms.
FROM SPACEXTBL: This specifies that the data is being retrieved from the SPACEXTBL table.
WHERE CUSTOMER = 'NASA (CRS)': This filters the results to include only those rows where the CUSTOMER column has the value 'NASA (CRS)', meaning only payloads carried for NASA under its Commercial Resupply Services (CRS) missions are included in the sum.
The query returned the total mass (in kilograms) of all payloads launched for NASA (CRS) missions.

Average Payload Mass by F9 v1.1

avg(PAYLOAD_MASS__KG_)
2928.4

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

Explanation:
%sql: This command is used to run SQL queries in a Jupyter notebook.
SELECT AVG(PAYLOAD_MASS__KG_): This selects the average value of the PAYLOAD_MASS__KG_ column, representing the average payload mass in kilograms.
FROM SPACEXTBL: Specifies the table SPACEXTBL where the data is being retrieved from.
WHERE BOOSTER_VERSION = 'F9 v1.1': Filters the rows to only include launches where the booster version is 'F9 v1.1', which refers to a specific version of the Falcon 9 rocket.
The query returned the average payload mass (in kilograms) for all launches that used the 'F9 v1.1' booster version.

First Successful Ground Landing Date

first_successful_landing
2015-12-22

```
%sql select min(date) as first_successful_landing from SPACEXTBL where landing__outcome = 'Success (ground pad)';
```

Explanation:
%sql: This command is used in Jupyter notebooks to indicate that the following line of code should be interpreted as SQL.
SELECT: Specifies the columns to retrieve from the database.
MIN(date): This aggregate function returns the earliest date from the date column in the SPACEXTBL table.
AS first_successful_landing: This gives an alias to the result of MIN(date), labeling it as first_successful_landing for clarity.
FROM SPACEXTBL: Specifies the table from which data is queried; SPACEXTBL contains records of SpaceX launches.
WHERE landing__outcome = 'Success (ground pad)': Filters records to include only those with a landing outcome classified as "Success (ground pad)."

Executing this query returned the earliest date where the landing outcome was successful on a ground pad, labeled as first_successful_landing, providing clear insight into the data point.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)"
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Explanation:
%sql: Command in Jupyter to execute SQL queries.
SELECT booster_version: Retrieves the booster_version column from the table.
FROM SPACEXDATASET: Specifies the table SPACEXTBL where the query is executed.
WHERE landing__outcome = 'Success (drone ship)': Filters rows where the landing outcome was a successful landing on a drone ship.
AND payload_mass__kg_ BETWEEN 4000 AND 6000: Further filters to include only those rows where the payload mass is between 4000 and 6000 kilograms.
This query returned the booster versions that successfully landed on a drone ship with a payload mass between 4000 and 6000 kilograms.

Total Number of Successful and Failure Mission Outcomes

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

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

Explanation:
This query retrieved the different **mission outcomes** and the **total number** of occurrences for each outcome from the **SPACEXTBL** table.
SELECT MISSION_OUTCOME: Retrieves the column "MISSION_OUTCOME."
COUNT(*) as total_number: Counts how many records (or rows) exist for each mission outcome and labels the result as "total_number."
FROM SPACEXTBL: Specifies the table (SPACEXTBL) from which to get the data.
GROUP BY MISSION_OUTCOME: Groups the data by the "MISSION_OUTCOME" column, so the count is calculated for each unique outcome.
It showed each unique mission outcome and how many times it occurred.

Boosters Carried Maximum Payload

```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Explanation:
This query retrieved the **BOOSTER_VERSION** associated with the **maximum payload mass** from the **SPACEXTBL** table.
SELECT BOOSTER_VERSION: Retrieves the "BOOSTER_VERSION" for the row with the highest payload mass.
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL): Finds the booster version where the payload mass is equal to the maximum value of "PAYLOAD_MASS__KG_" in the table.
It returned the booster version that carried the heaviest payload.

2015 Launch Records

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

```
%%sql
SELECT Landing_Outcome, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
  AND strftime('%Y', DATE) = '2015';
```

Explanation:
This query retrieved specific details from the **SPACEXTBL** table for SpaceX launches in the year 2015 that had a **Landing_Outcome** of 'Failure (drone ship)'.
SELECT Landing_Outcome, BOOSTER_VERSION, LAUNCH_SITE: Retrieves the **Landing_Outcome**, **Booster_Version**, and **Launch_Site** for relevant records.
FROM SPACEXTBL: Specifies that the data is coming from the **SPACEXTBL** table.
WHERE Landing_Outcome = 'Failure (drone ship)': Filters results to only include records where the landing outcome was a failed drone ship landing.
AND strftime('%Y', DATE) = '2015': Ensures the results are limited to launches that occurred in the year 2015.
It showed the failed drone ship landings in 2015, along with their booster version and launch site.

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

```
%%sql
SELECT Landing_Outcome, COUNT(*) AS count_outcomes
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY count_outcomes DESC;
```

Landing_Outcome	count_outcomes
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

Explanation:

This query retrieved the count of different landing outcomes for SpaceX launches between specific dates.

SELECT Landing_Outcome, COUNT(*) AS count_outcomes: Retrieves the landing outcome and the count of how many times each outcome occurred. The result is labeled as **count_outcomes**.

FROM SPACEXTBL: Specifies the **SPACEXTBL** table as the source of the data.

WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20': Filters the records to include only those where the launch date is between June 4, 2010, and March 20, 2017.

GROUP BY Landing_Outcome: Groups the results by each unique landing outcome.

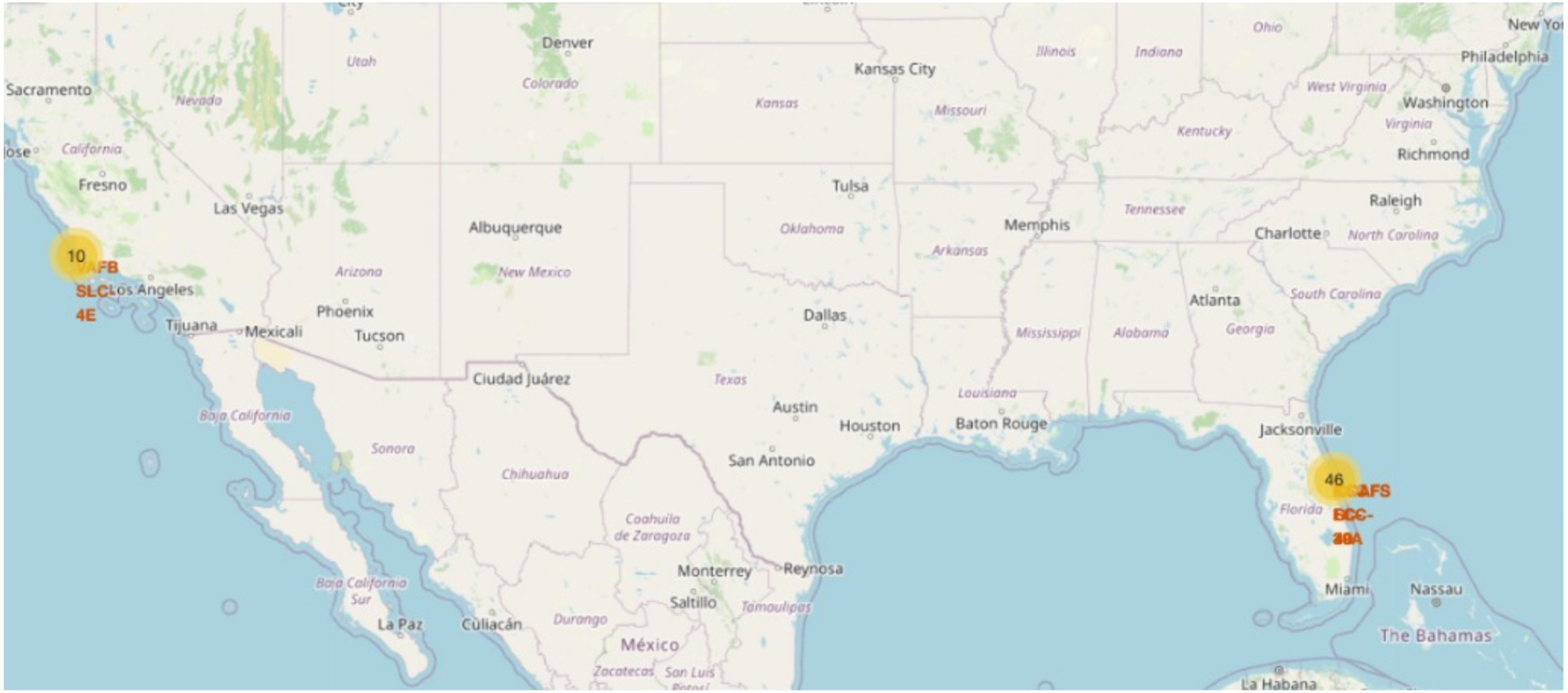
ORDER BY count_outcomes DESC: Sorts the results in descending order based on the count of outcomes.

It returned the number of occurrences for each landing outcome within the specified date range, with the most frequent outcomes listed first.

Section 3

Launch Sites Proximities Analysis

Launch Sites



Most launch sites are located near the equator, benefiting from the Earth's faster rotation at this latitude.
Launching near from the equator provides a natural speed boost due to inertia, helping spacecraft reach and maintain orbit more efficiently.
Additionally, all launch sites are positioned close to the coast, reducing the risk of debris falling near populated areas by directing launches over the ocean. This strategic placement near the equator and coastlines helps reduce fuel costs and enhances the safety and success of rocket launches.

Launch Results Overview: Successes and Failures

The **green** markers on the chart indicate successful rocket launches, while the **red** markers represent launches that were unsuccessful.
This visual distinction helps quickly identify the outcomes of the launches at a glance.

Visualizing Launch Site and Coastline Dynamics

The polyline serves as a visual connector, indicating the direct geographical relationship between the launch site and the coastline.
This helps understand how far the launch site is from the coast and can provide context for operations related to rocket launches, such as flight paths, safety zones, and environmental considerations.

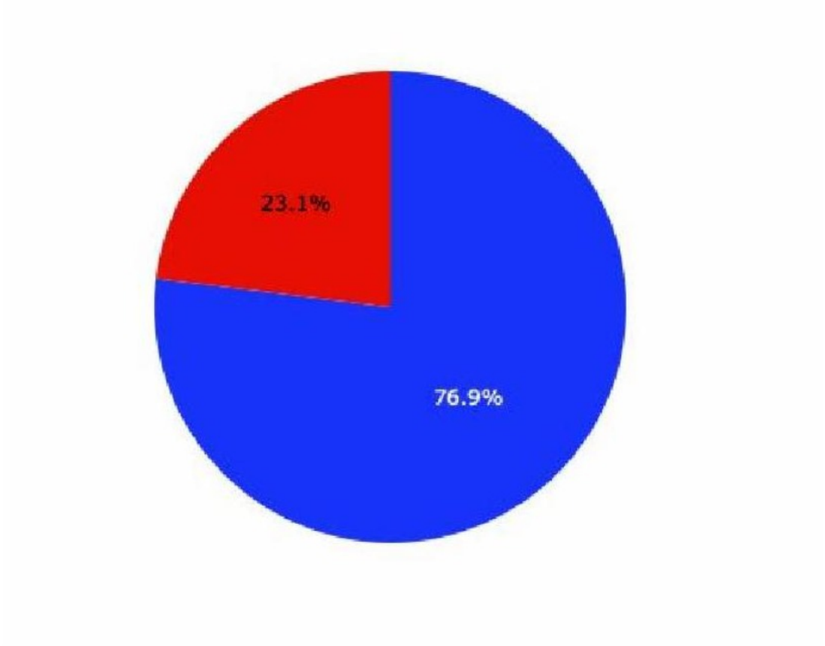
Section 4

Build a Dashboard with Plotly Dash

Total Success Launches by Site

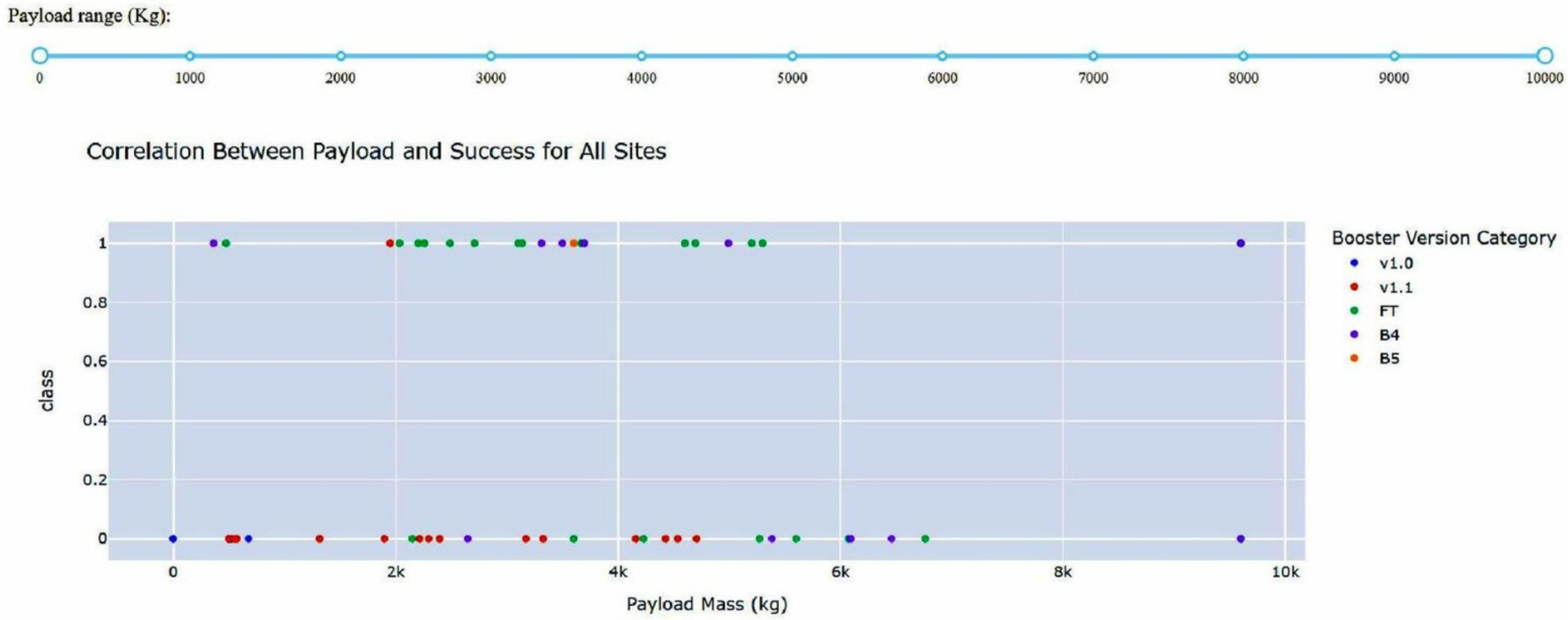
This pie chart illustrates the proportion of successful launches across all sites. It highlights that **KSC LC-39A** has the greatest number of successful launches, followed by **CCAFS SLC-40**, which ranks second in successful launches.

Launch Site with highest launch success ratio



KSC LC-39A, the launch site with the highest success rate, has successfully completed 76.9% of its launches, while 23.1% of its launches have resulted in failure.

Screenshot of Payload vs. Launch Outcome scatter plot for all sites

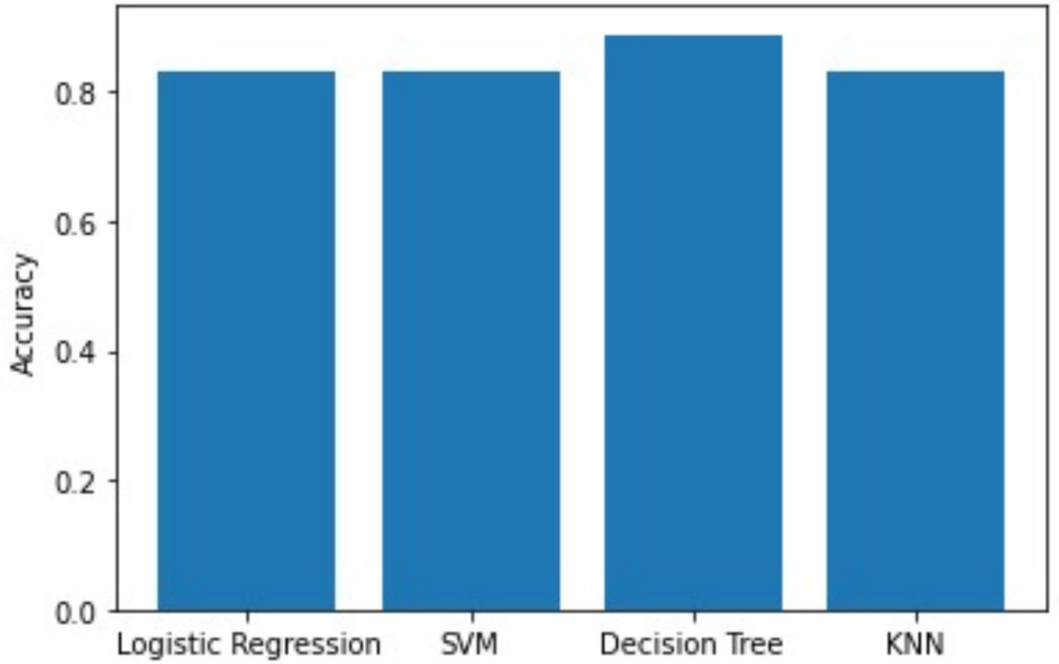


The screenshot highlights that the **2,000 to 4,000 kg** payload range has the highest success rate, followed by **4,000 to 6,000 kg**. Among booster versions, **FT** (green spots) has the most successful launches, with **B4** (purple spots) coming in second. These findings are crucial in understanding which payload sizes and booster versions have a higher likelihood of successful launches, aiding in decision-making for future launches.

Section 5

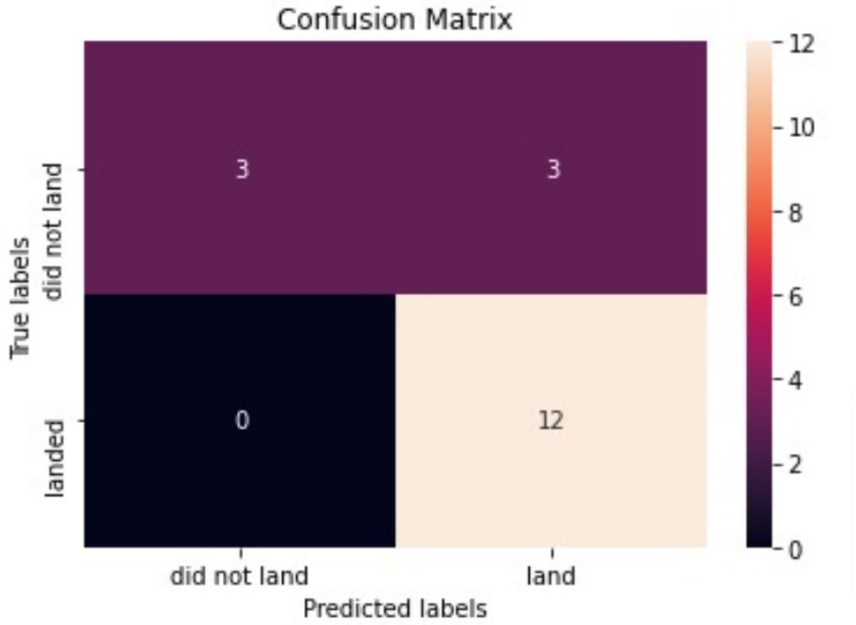
Predictive Analysis (Classification)

Classification Accuracy



The bar chart illustrates the accuracy of various classification models utilized in the project.
Among these, the **Decision Tree Classifier** stands out with the highest accuracy, indicating its effectiveness for the given dataset. In contrast, the **KNN classifier** shows the lowest accuracy score, suggesting that this particular configuration may not perform as well in accurately classifying the data compared to the other models.
This comparison provides valuable insights into the performance of each model, guiding future decisions on model selection and optimization

Confusion Matrix



The confusion matrix for the **Decision Tree Classifier** indicates that the model is capable of distinguishing between the different classes.
The main issue lies in the occurrence of **false positives**, where the classifier incorrectly predicts **unsuccessful landings** as **successful landings**.
This type of error can lead to misleading results and may impact the overall reliability of the model, especially when making decisions based on success rates.
By addressing the issue of false positives, the overall reliability of the classifier can be enhanced, leading to more accurate and trustworthy predictions.


Conclusions

SpaceX’s ability to reuse the first stage of the Falcon 9 rocket plays a critical role in reducing launch costs, bringing prices down to $62 million compared to over $165 million charged by competitors. This reusability is key to their competitive pricing strategy.
Predicting the success of the first stage landing is directly tied to estimating the overall cost of launches. Accurately forecasting the outcome of the landing helps SpaceX optimize costs and maintain their price advantage.

Key Factors Affecting Landing Success:

Through data analysis, the project aimed to identify the main factors that influence the success of the Falcon 9 first stage landing. Understanding these factors not only improves the accuracy of predictions but also provides valuable insights into improving operational efficiency.
The findings showed that payload size, booster version, and other launch-specific conditions significantly impact landing success rates, making them critical factors for future predictions.

Machine Learning for Predicting Landing Success:

By developing machine learning models, such as Decision Tree Classifiers, the project successfully created a predictive framework for estimating the likelihood of first stage landings. While the models showed promising accuracy, particularly the Decision Tree, false positives (incorrectly predicting successful landings) highlighted areas for further refinement.
These predictions are crucial for estimating the overall cost of launches since successful landings enable rocket reusability, driving down long-term costs.

Impact on Launch Pricing Strategies:

SpaceX’s competitive pricing is supported by its ability to consistently reuse rocket components. Accurate landing predictions help SpaceX optimize this strategy, allowing them to offer lower prices while maintaining high profit margins.
Understanding the trends in pricing and how competitors may respond allows for better forecasting of future market conditions. Competitors might adopt similar reusability strategies or focus on other innovations to lower their costs and stay competitive.

Data Collection and Analysis:

The project highlights the importance of continuous and effective data collection regarding launch outcomes, payload types, environmental conditions, and booster versions. This data is essential for improving machine learning models, predicting future outcomes, and optimizing SpaceX's operations.
Applying machine learning techniques to analyze this data can provide insights not only into landing success but also into broader trends in the space launch industry, helping companies make data-driven decisions.

Implications for the Broader Market:
The analysis highlighted how SpaceX’s pricing and reusability model are revolutionizing the space launch industry, pushing competitors to rethink their strategies. Companies may need to explore innovations in rocket design, reusability, or cost optimization to compete with SpaceX’s cost-effective approach.
Also, landing predictions play a vital role in shaping future pricing models and competitive positioning within the space launch market.

Overall Conclusion:

The project demonstrates that predicting first stage landing success is crucial to SpaceX's cost savings and competitive advantage.

By identifying key factors affecting landings and leveraging machine learning models to make accurate predictions, SpaceX can further optimize its operations, maintain its pricing advantage, and drive continued innovations in the space launch industry. These findings not only support SpaceX's strategy but also provide insights into how competitors might adapt to remain effective in the evolving market.

Appendix

Key Concepts in the Machine Learning Section

This appendix provides an overview of key concepts related to the machine learning models and techniques used in this project. Each concept is defined and explained in the context of predicting Falcon 9 rocket landing success.In this project, we applied four different machine learning models to predict the success of Falcon 9 rocket landings:

1.**Logistic Regression**: Is a statistical model that estimates the probability of a successful landing by analyzing the relationships between the input variables and the outcome.

2.**Support Vector Machines (SVM)**: Is a classification algorithm that works by identifying the optimal boundary (or hyperplane) between different classes (successful vs. unsuccessful landings).

3.**Decision Tree**: Is a model that splits the dataset into subsets based on conditions that lead to successful or unsuccessful landings. It progressively breaks the data into smaller decisions based on input features.

4.**K-Nearest Neighbors (KNN)**: Is a simple algorithm that classifies new data points by comparing them to the most similar data points in the dataset (the nearest neighbors).

Model Evaluation

The performance of the models was assessed using the following metrics:

Jaccard Score: Measures the similarity between the predicted outcomes and the actual outcomes by comparing how often they agree.

F1 Score: Is an average of precision (the proportion of correct positive predictions) and recall (the proportion of actual positives correctly identified), offering a balance between the two.

Accuracy: Reflects the percentage of correctly predicted outcomes out of all predictions made by the model.

Among these models, the **Decision Tree** model demonstrated the highest accuracy, slightly outperforming the other models.


Thank you!
SPACEX