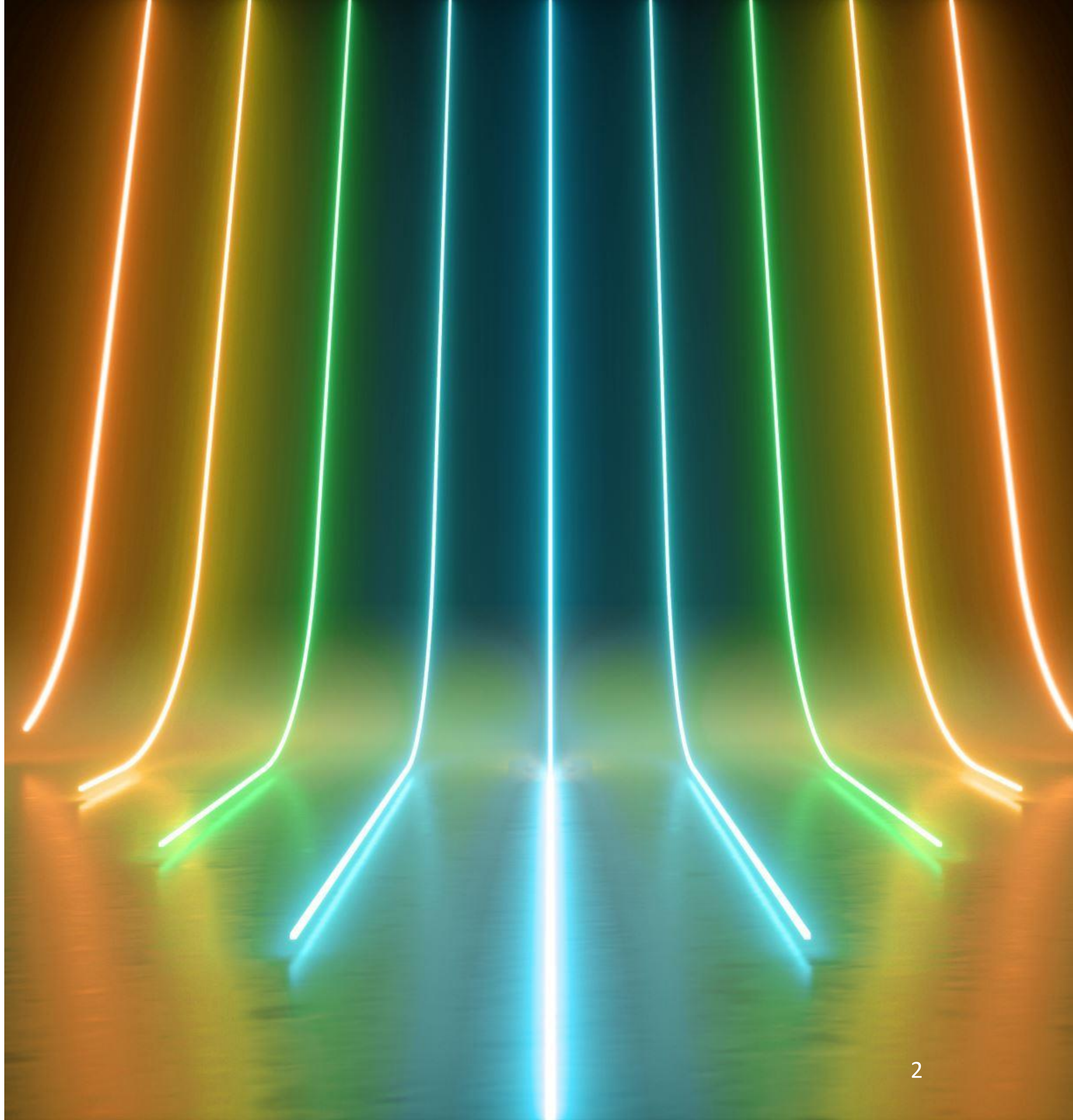
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jeremy Harkness
21 September 2023

Outline



Executive Summary

We analyze Falcon 9 first-stage landings data to determine the likelihood of a successful landing.

Methodologies:

- Data collection involved SpaceX REST API and web scraping for past launches.
- Data wrangling focused on improving data quality and filtering Falcon 1 launches.
- EDA revealed correlations between launch site, payload mass, and successful landings.
- Interactive visual analytics using Folium and Plotly Dash showcased launch site proximity.
- Predictive analysis utilized SVM, Classification Trees, and Logistic Regression.

Key Findings:

1. Launch site selection, payload mass, and mission success rates impact successful landings.
2. Interactive visualizations provide valuable understanding of optimal launch sites.
3. Predictive models assist in predicting successful landings accurately.

Implications:

1. Informed decision-making regarding cost-effectiveness and reusability of Falcon 9.
2. Identified crucial factors for successful first-stage landings .
3. Potential for increased competition in the space launch market.

Introduction

- Background:
  Rapid expansion of commercial space travel, and the emergence of SpaceY, as new competitor, requires understanding of costs and reliability of reusable rockets.

- Determine Falcon 9 Launch Cost Estimation
  - Gather SpaceX cost data.
  - Develop cost estimation models.
  - Create informative dashboards.

- First Stage Reusability Prediction
  - Utilize machine learning.
  - Analyze public information.
  - Forecast Falcon 9 first stage reusability.

Section 1

Methodology

Methodology

Data collection methodology:

- Utilized SpaceX REST API to gather past launch data
- Implemented web scraping techniques to extract Falcon 9 launch records from HTML tables

Data wrangling:

- Handled null values and missing data
- Filtered out Falcon 1 launches to focus on Falcon 9 data
- Transformed the data into a clean and structured format

Methodology

- Data processing:
  - Cleaned and organized the data to ensure quality
  - Filled in missing values and removed irrelevant data

- Exploratory data analysis (EDA) using visualization and SQL:
  - Used visualization techniques to uncover patterns and trends in the data
  - Employed SQL queries for deeper analysis and extract specific information

- Interactive visual analytics using Folium and Plotly Dash:
  - Created interactive Folium maps for launch site locations and proximity
  - Used Plotly Dash to build interactive dashboards for dynamic exploration

Methodology

- Predictive analysis using classification models:
    - Built classification models (e.g. Support Vector Machines, Classification Trees, Logistic Regression)
    - Split the data into training and test sets
    - Tuned hyperparameters to optimize model performance using techniques like grid search or randomized search

- How to build, tune, evaluate classification models:
    - Constructed classification models by training them on the training set
    - Tuned the models' hyperparameters to improve their performance
    - Evaluated the models using metrics such as accuracy, precision, recall, and F1 score
    - Selected the best-performing model for predicting the success of Falcon 9 first-stage landings

Data Collection

Method 1: SpaceX API Data Collection

- **Objective:** To collect and format data from an API to predict the success of Falcon 9 first stage landings.
- **Steps:**
    - Identifying the necessary API endpoints.
    - Making API calls to collect data.
    - Formatting and structuring the collected data for analysis.
- **Tools:** Python, SpaceX API - https://api.spacexdata.com/v4/launches/past.

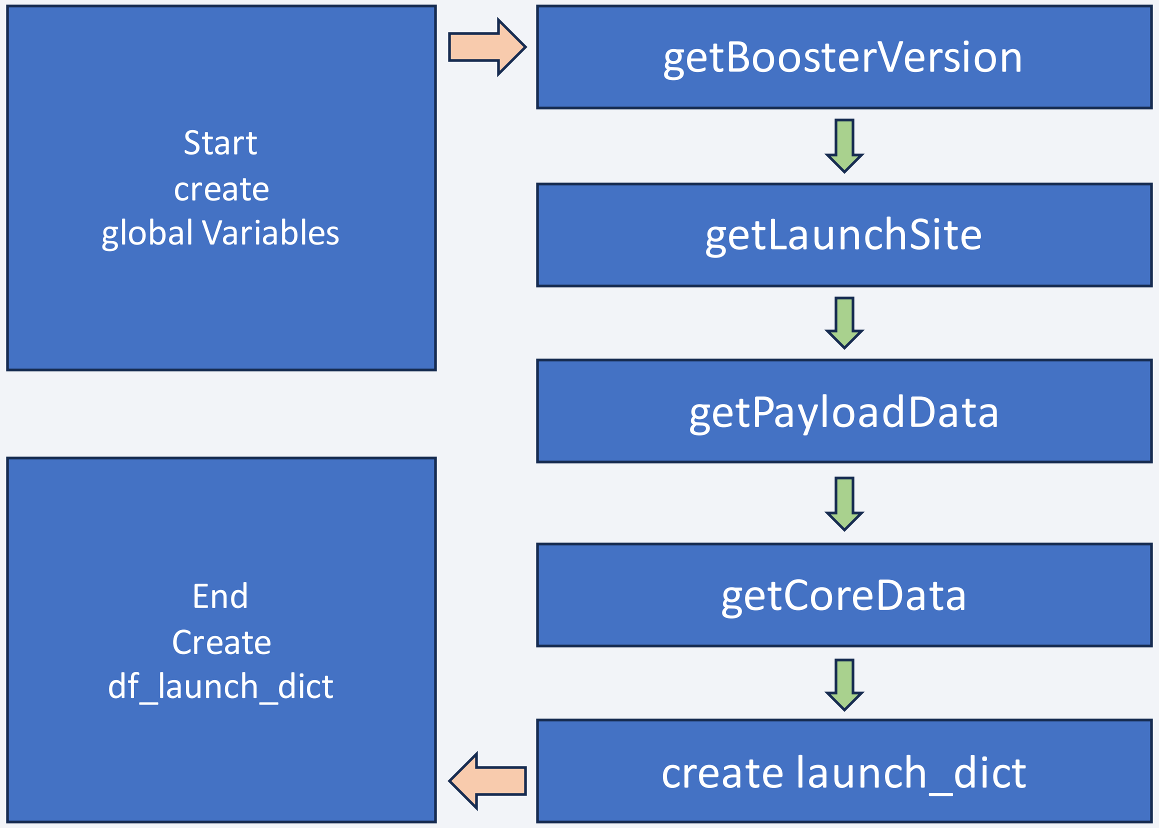
Method 2: Web Scraping

- **Objective:** To scrape Falcon 9 and Falcon Heavy launch records from a Wikipedia page.
- **Steps:**
    - Identifying the Wikipedia page with the necessary data.
    - Utilizing web scraping techniques to extract data.
    - Cleaning and formatting the scraped data for analysis.
- **Tools:** Python, BeautifulSoup library.
- **Source:** Wikipedia page https://api.spacexdata.com/v4/launches/past

Data Collection – SpaceX API

- Using the **'requests'** library with appropriate **function calls**, we populate the **global variables** with the data from the SpaceX **REST endpoints.**

- The GitHub URL of the completed SpaceX API calls notebook
- jupyter-labs-spacex-data-collection-api.ipynb



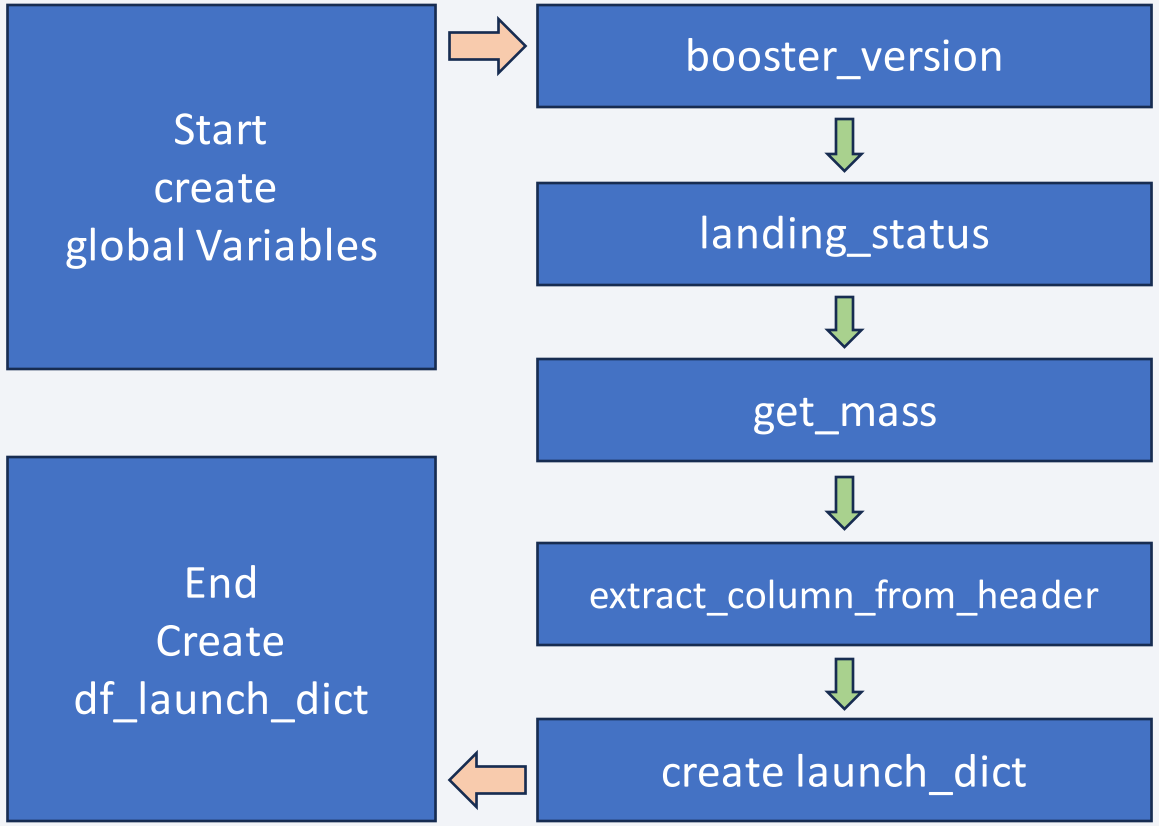
https://github.com/jeremyharkness/Applied-Data-Science-Capstone/blob/main/Week-1-Introduction/10%20-%20App%20Item%C2%A0-%20Hands-on%20Lab%20-%20Complete%20the%20Data%20Collection%20API%20Lab/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Using the **'BeautifulSoup'** library with appropriate **function calls**, we populate the **global variables** with the data from a snapshot of the SpaceX Wikipedia Wiki.

- The GitHub URL of the completed SpaceX API calls notebook

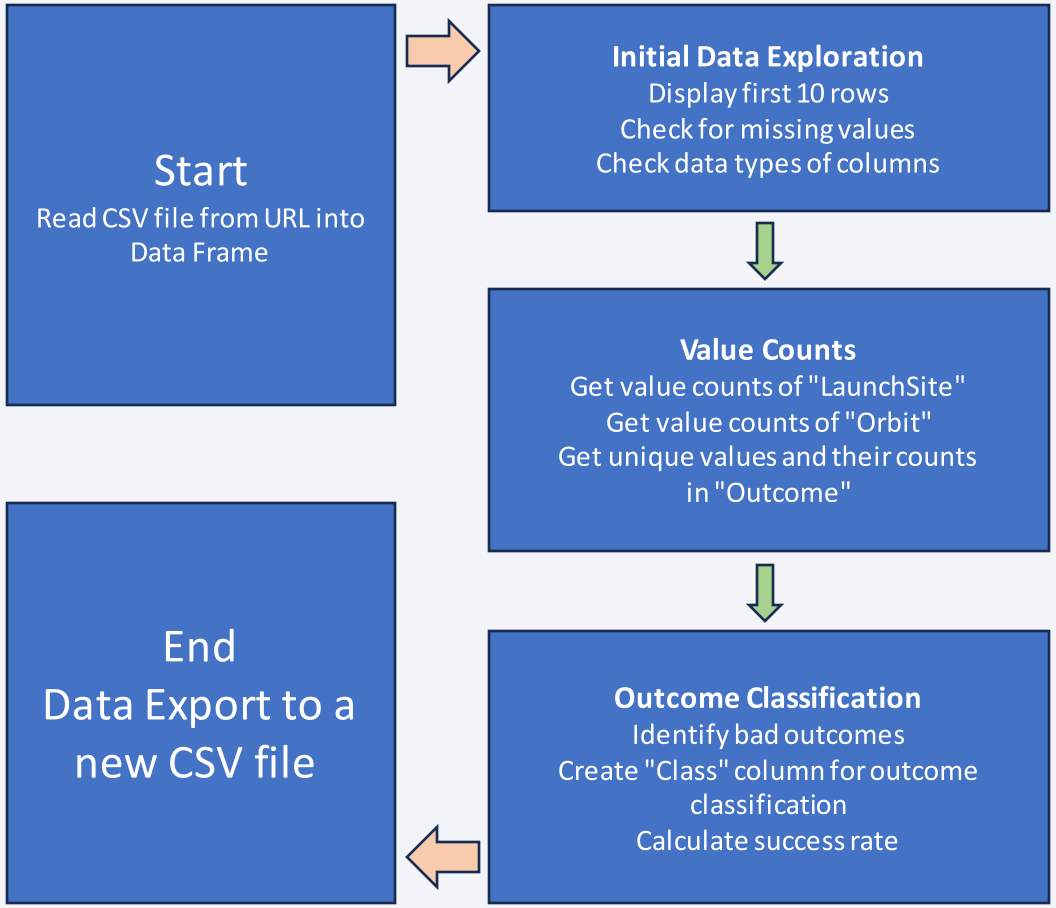
- jupyter-labs-webscraping.ipynb



https://github.com/jeremyharkness/Applied-Data-Science-Capstone/blob/main/Week-1-Introduction/12%20-%20App%20Item%C2%A0-%20Hands-on%20Lab%20-%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab/jupyter-labs-webscraping.ipynb

Data Wrangling

- Check for **missing** values & view data types
  - df.isnull().sum() & df.dtypes
- Get the **value counts** of the columns
  - df['**LaunchSite**'].value_counts()
  - df['**Orbit**'].value_counts()
  - df['**Outcome**'].value_counts()
- **Identify** & **group** bad outcomes into a set
- Create column "Class" to **classify** outcomes.
  0 for **bad** , & **1** for **good** outcomes.
- labs-jupyter-spacex-data_wrangling.ipynb



Start
Read CSV file from URL into Data Frame
Initial Data Exploration
Display first 10 rows
Check for missing values
Check data types of columns
Value Counts
Get value counts of "LaunchSite"
Get value counts of "Orbit"
Get unique values and their counts in "Outcome"
Outcome Classification
Identify bad outcomes
Create "Class" column for outcome classification
Calculate success rate
End
Data Export to a new CSV file

https://github.com/jeremyharkness/Applied-Data-Science-Capstone/blob/main/Week-1-Introduction/16%20-%20App%20Item%C2%A0-%20Hands-on%20Lab%20-%20Data%20Wrangling/labs-jupyter-spacex-data_wrangling.ipynb

EDA with Data Visualization

- **Task 1:** Scatter Plot - Visualize the relationship between 'flight number' and 'launch site'.
- **Task 2:** Scatter Plot - Analyze the relationship between the 'payload mass' (in kg) and the 'launch site'.
- **Task 3:** Bar Chart - Visualize the relationship between success rate of each 'orbit type'.
- **Task 4:** Scatter Plot - Visualizes the distribution of different 'orbit types'' over the flight numbers'.
- **Task 5:** Scatter Plot - Visualize the relationship between the payload mass and the orbit type.
- **Task 6:** Line Plot - Visualize the trend of launch success over the years.

- Final notebook jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Task 1: Display the names of the unique launch sites in the space mission.
  - Selecting distinct values from the "Launch_Site" column in the SPACEXTABLE.
  - **SQL Query:** %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
- Task 2: Display 5 records where launch sites begin with the string 'CCA'.
  - Selecting all columns from SPACEXTABLE.
  - Filtering rows to include only those where the "Launch_Site" column starts with 'CCA'.
  - Limiting the output to the first 5 rows.
  - **SQL Query:** %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5

EDA with SQL

- Task 3: Display the total payload mass carried by boosters launched by NASA (CRS).
  - Calculating the sum of the values in the "PAYLOAD_MASS__KG_" column.
  - Filtering the rows where the "Customer" column starts with 'NASA (CRS)'.
  - **SQL Query:** %sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" LIKE 'NASA (CRS)%'
- Task 4: Display the average payload mass carried by booster version F9 v1.1.
  - Calculating the average of the values in the "PAYLOAD_MASS__KG_" column.
  - Filtering the rows where the "Booster_Version" column starts with 'F9 v1.1'.
  - **SQL Query:** %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" like 'F9 v1.1%'

EDA with SQL

- Task 5: List the date when the first successful landing on the ground pad was achieved.
  - Finding the minimum date in the "Date" column.
  - Filtering the rows where the "Landing_Outcome" column starts with 'Success (ground pad)'.

  - **SQL Query:** %sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (ground pad)%'

EDA with SQL

- Task 6: List the names of the boosters which have success landing on a drone ship and have a payload mass greater than 4000 but less than 6000.
  - Selecting the "Booster_Version" column.
  - Filtering the rows where:
  - The "Landing_Outcome" column starts with 'Success (drone ship)'.
  - The "PAYLOAD_MASS__KG_" column has values between 4000 and 6000.

  - **SQL Query:** %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (drone ship)%' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;

EDA with SQL

- Task 7: List the total number of successful and failure mission outcomes.
  - Creating a new column "Mission_Result" based on the "Mission_Outcome" column values.
  - Counting the number of occurrences of each "Mission_Result".
  - Grouping the results by the "Mission_Result" column.

  - **SQL Query:** %sql SELECT CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 'Success' ELSE 'Failure' END AS "Mission_Result", COUNT("Mission_Outcome") FROM SPACEXTABLE GROUP BY "Mission_Result";

EDA with SQL

- Task 8: List the names of the booster versions which have carried the maximum payload mass (using a subquery).
  - Selecting the "Booster_Version" column.
  - Filtering the rows where the "PAYLOAD_MASS__KG_" column equals the maximum value in the "PAYLOAD_MASS__KG_" column in the SPACEXTABLE.
  - **SQL Query:** %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG _" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);

EDA with SQL

- Task 9: List the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in the year 2015. Creating a new column "Month" to represent the month name.
  - Selecting the "Landing_Outcome", "Booster_Version", and "Launch_Site" columns.
  - Filtering the rows where:
    - The "Landing_Outcome" column starts with 'Failure (drone ship)'.
    - The year extracted from the "Date" column equals '2015'.
  - **SQL Query:**

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

EDA with SQL

- Task 10: Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order.
  - Selecting the "Landing_Outcome" column and counting the number of occurrences of each "Landing_Outcome".
  - Filtering the rows where the "Date" column falls between '2010-06-04' and '2017-03-20'.
  - Grouping the results by the "Landing_Outcome" column.
  - Ordering the results in descending order based on the count of "Landing_Outcome".
  - **SQL Query:** %sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS Count FROM SP ACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Count DESC;

EDA with SQL

- Add the GitHub URL of your completed EDA with SQL notebook.
- jupyter-labs-eda-sql-coursera_sqllite.ipynb

- https://github.com/jeremyharkness/Applied-Data-Science-Capstone/blob/main/Week-2%20-%20Exploratory%20Data%20Analysis%20-%20EDA/02%20-%20App%20Item%C2%A0-%20Hands-on%20Lab%20-%20Complete%20the%20EDA%20with%20SQL/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Task 1: Mark all launch sites on a map

  - Objective: - Visualize SpaceX launch sites on a map using coordinates.
  - Map Objects and Purpose:
    - Folium Map - Base map centered at NASA Johnson Space Center.
    - Folium Circle and Marker - Highlight NASA Johnson Space Center location.
    - Folium Circle and Marker (for each launch site)
      Mark and label launch sites to facilitate geographical analysis.

Build an Interactive Map with Folium

- Task 2: Mark the success/failed launches for each site on the map
  - Objective: - Visualize launch outcomes (success/failure) at each SpaceX launch site.
  - Map Objects and Purpose:
    - Folium CircleMarker - Mark launch sites with popup labels.
    - Folium Circle (Green/Red) - Represent success/failure of launches, with radius indicating count.
    - Marker Cluster - Cluster markers for individual launches, color-coded for success (green) or failure (red).

Build an Interactive Map with Folium

- Task 3: Calculate the distances between a launch site to its proximities
  - Objective: - Analyze proximities (like coastlines, cities) to launch sites by calculating and visualizing distances.
  - Map Objects and Purpose:
    - MousePosition Plugin - Help find coordinates of points of interest on the map.
    - Folium Marker - Mark points of interest and display distances from the launch site.
    - Folium PolyLine - Draw lines between the launch site and points to visually represent distances.

- Add the GitHub URL of your completed interactive map with Folium lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Task 1:
  - A dropdown for selecting a launch site was added allowing users to interact with the dashboard by selecting a launch site to analyze.
  - A pie chart was added as a visual representation of the success rate of the launches from the selected site, helping users quickly grasp the performance of each site.

Build a Dashboard with Plotly Dash

- Task 1:
    - A dropdown for selecting a launch site was added allowing users to interact with the dashboard by selecting a launch site to analyze.
    - A pie chart was added as a visual representation of the success rate of the launches from the selected site, helping users quickly grasp the performance of each site.
- Task 2:
    - A slider to filter the payload mass was added, allowing users to focus on launches within a specific payload mass range, facilitating more detailed analysis.
    - A scatter plot was introduced to depict the correlation between payload and launch success for the selected launch site, offering insights into the optimal payload mass for successful launches.

Build a Dashboard with Plotly Dash

- Task 3:
  - A scatter plot was integrated to visualize the success rate of each booster version, helping users to understand how different booster versions perform and identify the most and least successful versions.
- Task 4:
  - A dropdown to select a range of dates for the launches was added, letting users analyze launches within a specific time frame, thus facilitating the focus on more recent launches or exploring historical performance.
  - The functionality to filter the scatter plot (from task 3) based on the selected date range was added, empowering users to identify potential trends or changes in performance over time, aiding in a more nuanced analysis.

- Add the GitHub URL of your completed Plotly Dash lab
  spacex_dash_app_task4_completed.py

Predictive Analysis (Classification)

- Data Preparation
  - Loaded two datasets from specified URLs using asynchronous fetch requests
  - Tasks 1 & 2 - Created a NumPy array using the "Class" column
  - Standardized the feature set using StandardScaler
  - Data Splitting
  - Task 3 - Split the data into training (80%)
    testing (20%) sets using a random state of 2

Predictive Analysis (Classification)

- Model Evaluation
  - Tasks - 5, 7, 9, and 11
  - For each of the models built in tasks 4, 6, 8, and 10, the accuracy on the test data was calculated using the score method
  - Visualized the performance using confusion matrices to understand the true positives, true negatives, false positives, and false negatives

Predictive Analysis (Classification)

- Model Building and Hyperparameter Tuning
  - **Task 4 -** Built a logistic regression model using GridSearchCV with a 10-fold cross-validation to find the best hyperparameters from a specified grid
  - **Task 6 -** Applied a similar process for SVM, exploring various kernels and other parameters
  - **Task 8 -** Built a decision tree classifier exploring different criteria and depth levels
  - **Task 10 -** Applied k-Nearest Neighbors (KNN) classification, tuning the number of neighbors and the distance metric

Predictive Analysis (Classification)

- Model Selection
  - **Task 12 -** Compared the accuracies of all the models on the test data
  - Identified the best performing model, which was found to be **Logistic Regression** with an accuracy of **83.33%**

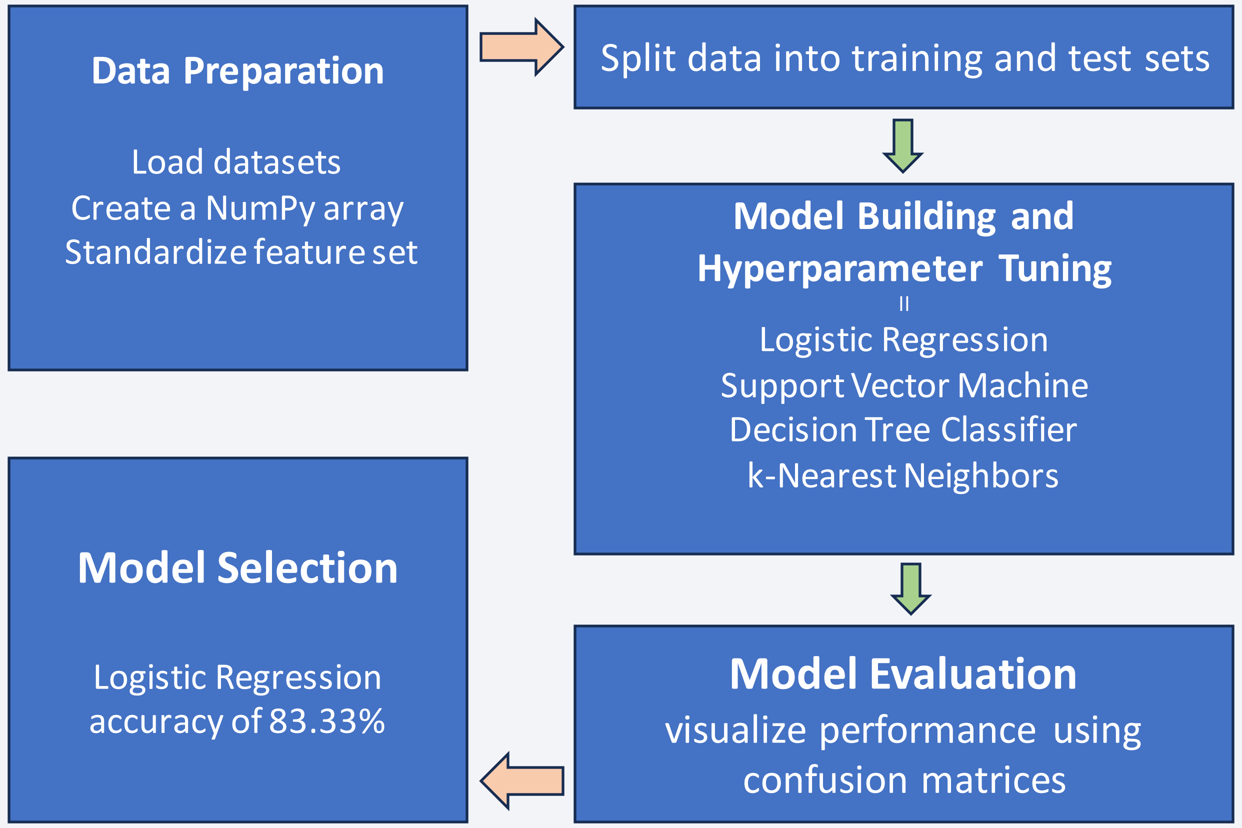
Predictive Analysis (Classification)

- Key Phrases

  Exploratory Data Analysis
  Data Standardization
  Training and Testing Split
  Hyperparameter Tuning
  Model Evaluation
  Confusion Matrix Analysis
  Best Model Selection

- The GitHub URL of the completed SpaceX API calls notebook

  - SpaceX_Machine_Learning_Completed.ipynb



https://github.com/jeremyharkness/Applied-Data-Science-Capstone/blob/main/Week-4%20-%20Predictive%20Analysis%20-%20Classification/02%20-%20App%20Item%C2%A0-%20Hands-on%20Lab%20-%20Complete%20the%20Machine%20Learning%20Prediction%20lab/SpaceX_Machine_Learning_Completed.ipynb

Results

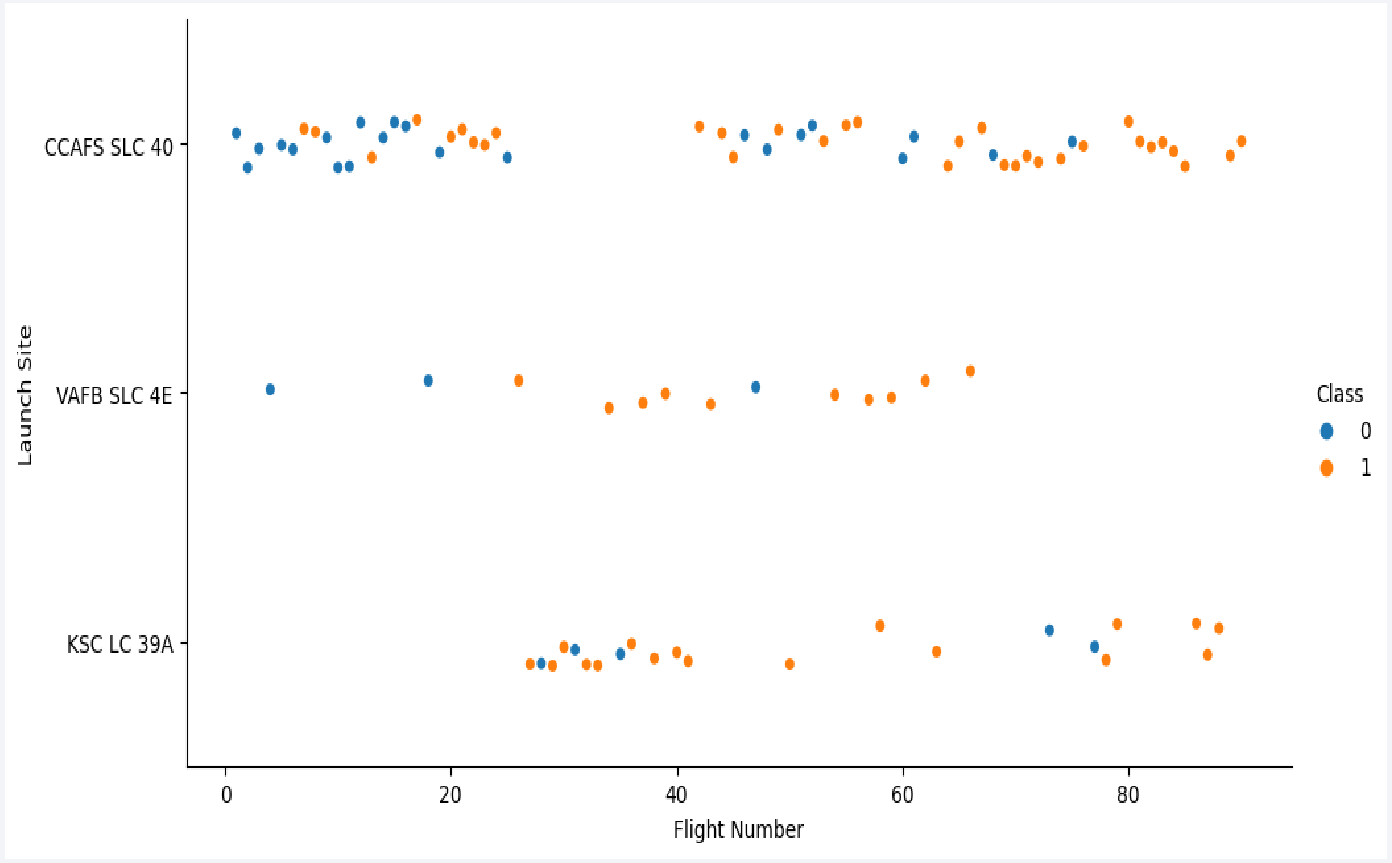
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

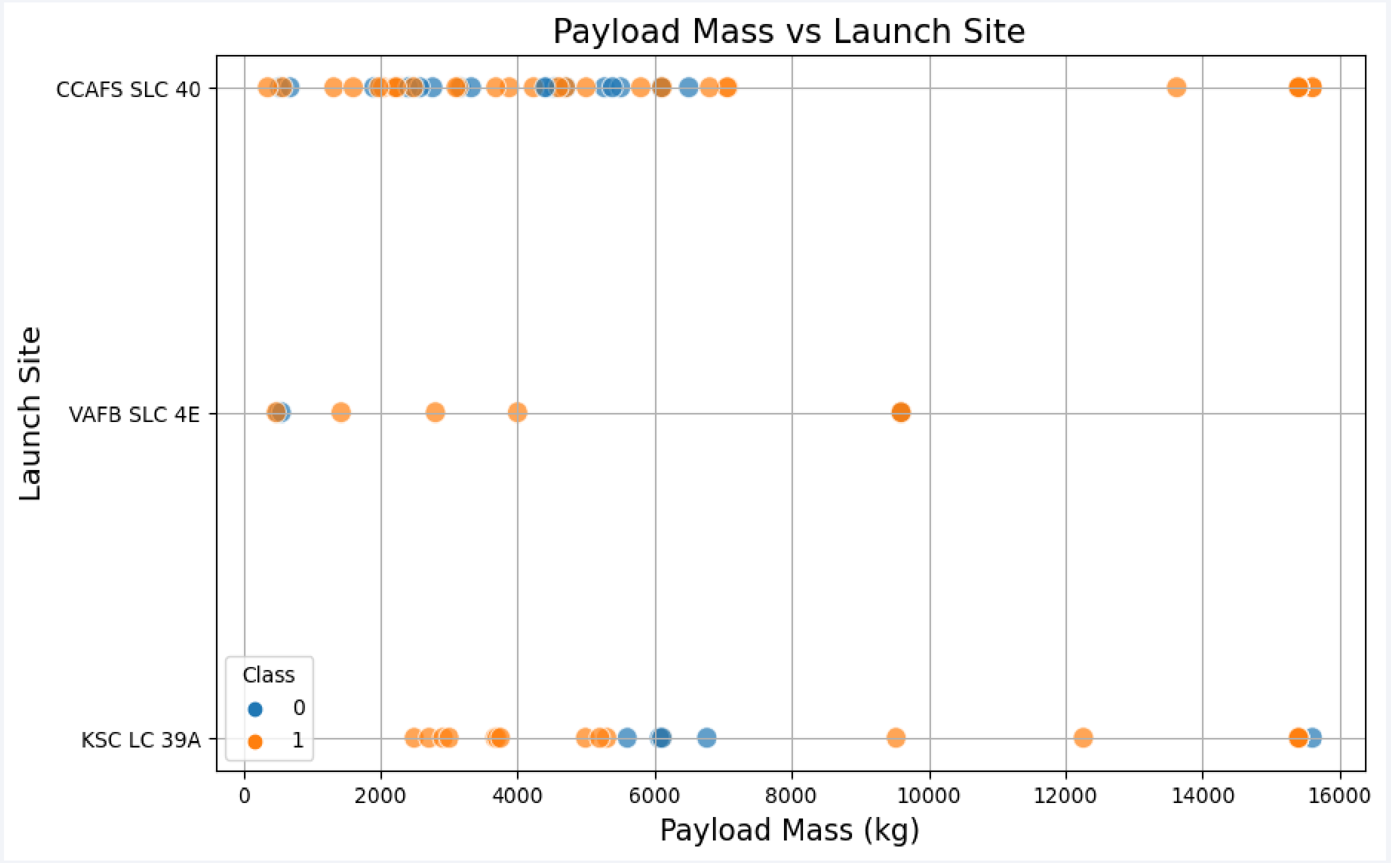
Flight Number vs. Launch Site

- The plot allows for a deeper dive into each launch site to visualize its detailed launch records.
- By observing the scatter points, we can identify patterns in flight numbers and their success rates at different launch sites



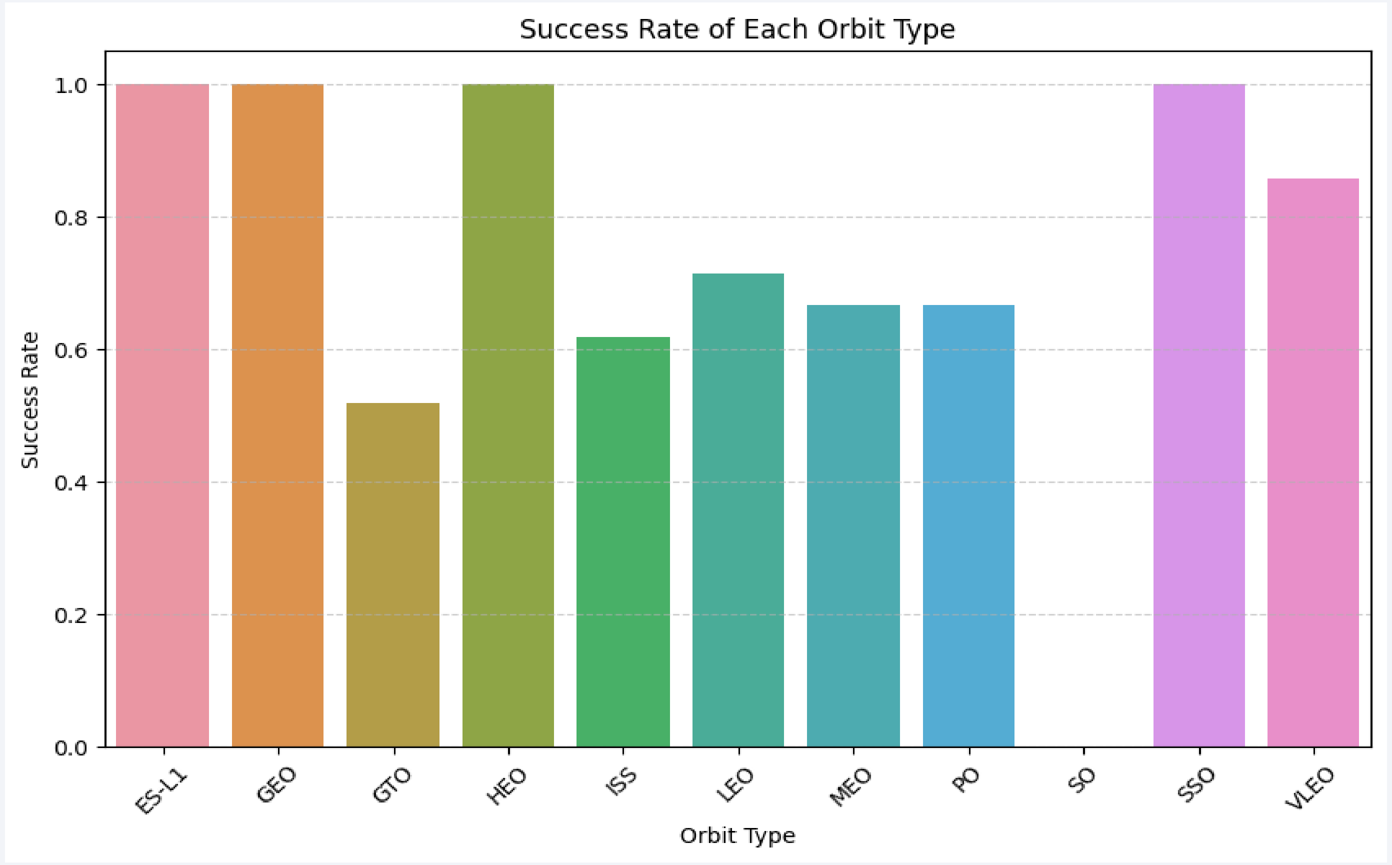
Payload vs. Launch Site

- This plot visualizes the relationship between different launch sites and their payload mass.
- We can observe that for the VAFB-SLC launch site, there are no rockets launched for heavy payload mass (greater than 10,000).
- This suggests that certain launch sites might be preferred for specific payload mass ranges, potentially due to infrastructure or location constraints.



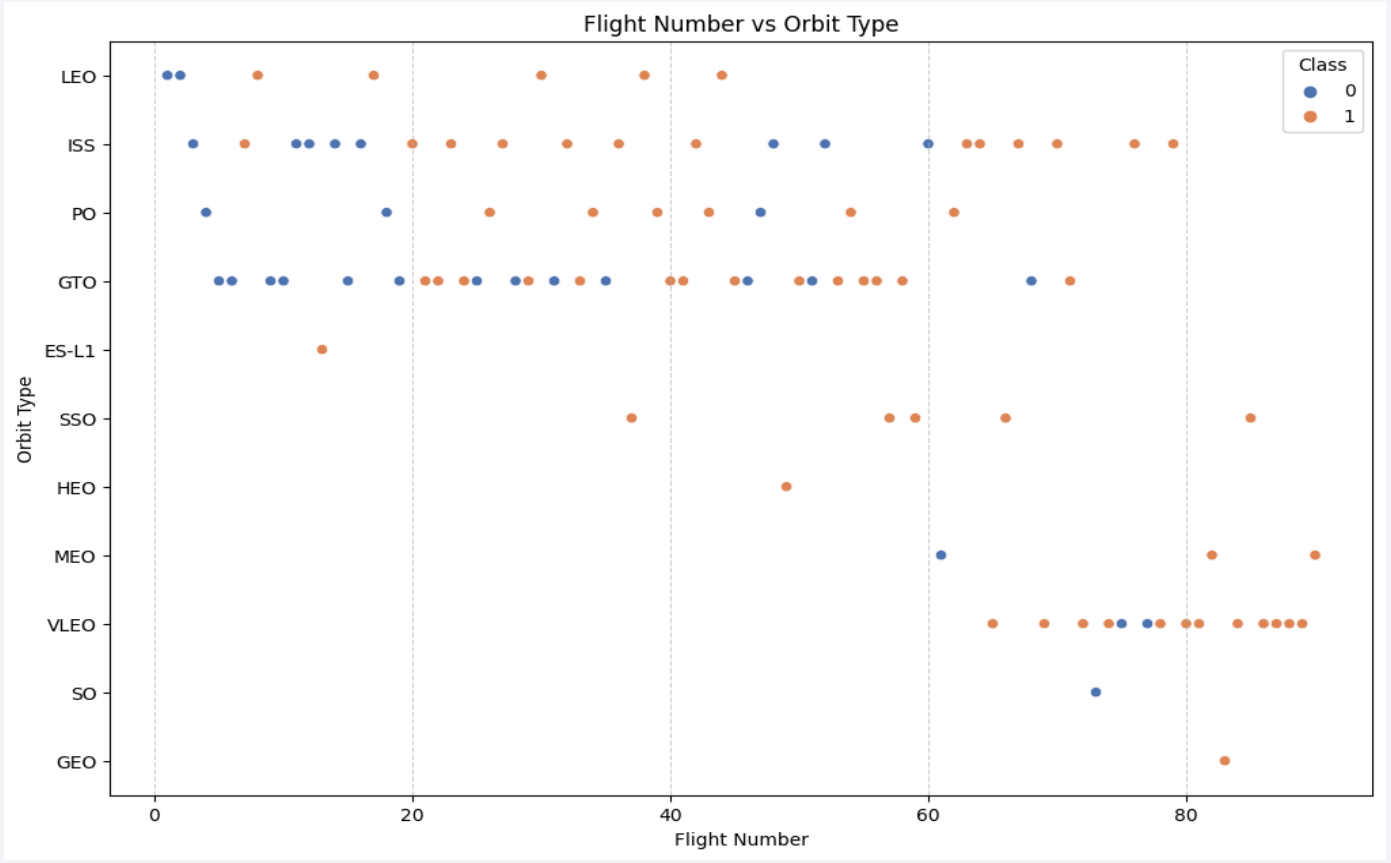
Success Rate vs. Orbit Type

- This bar chart displays the success rate for each orbit.
- It allows us to determine which orbits have a higher success rate.
- By analyzing the bar chart, we can derive insights about the relationship between success rates and orbit types, which can be crucial for planning future launches.



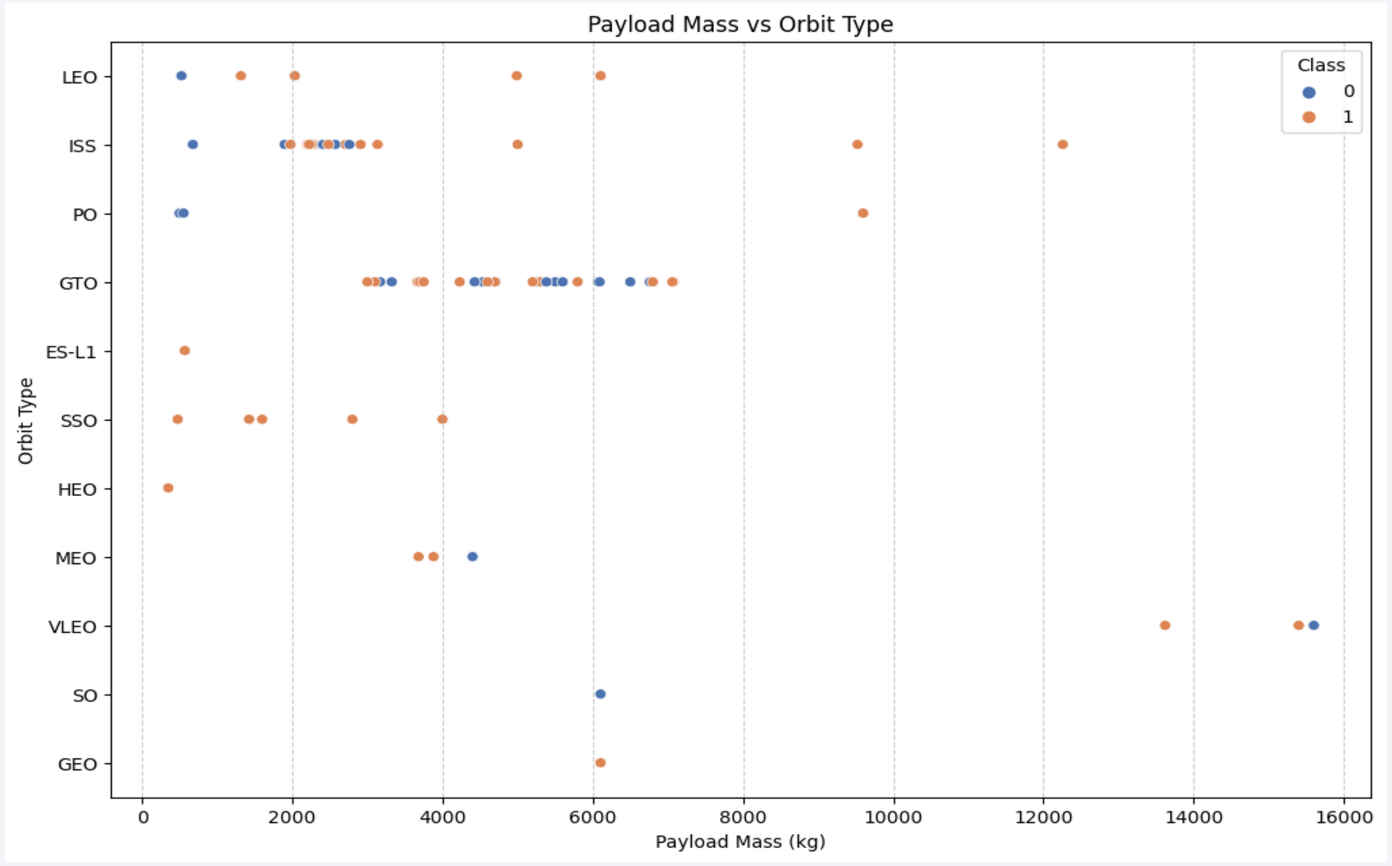
Flight Number vs. Orbit Type

- This plot shows the relationship between FlightNumber and Orbit type.
- In the LEO orbit, success appears to be related to the number of flights.
- Conversely, for the GTO orbit, there seems to be no relationship between the flight number and success.
- This plot helps in understanding how the success rate varies with the number of flights for different orbit types, which can be essential for risk assessment and planning.



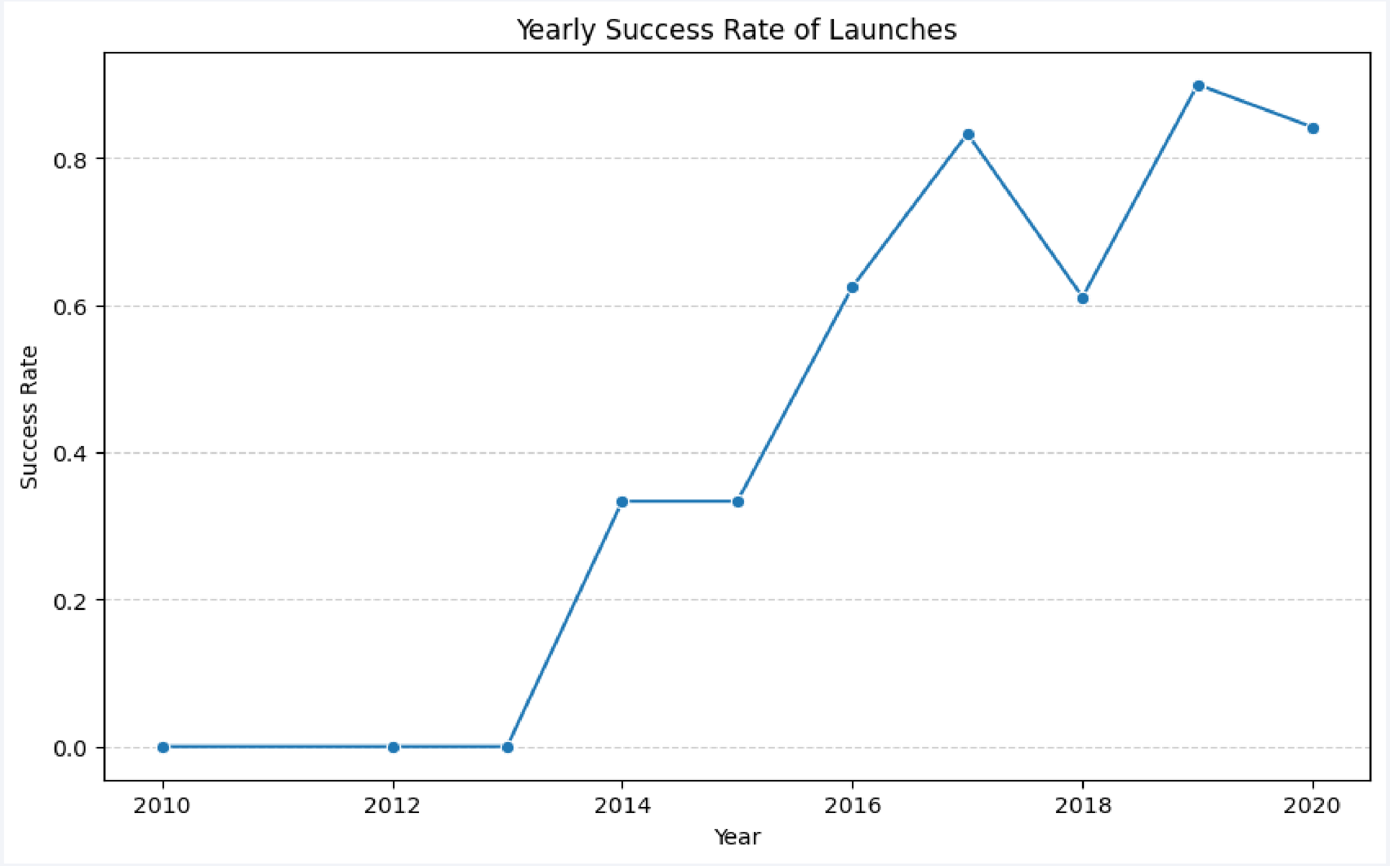
Payload vs. Orbit Type

- This scatter plot visualizes the relationship between Payload and Orbit type.
- We can observe that with heavier payloads, the successful landing rates are higher for Polar, LEO, and ISS orbits.
- For the GTO orbit, the distinction is less clear as both successful and unsuccessful landings are observed for various payload masses.
- This indicates that the success of a landing might be influenced by both the payload's weight and the intended orbit.



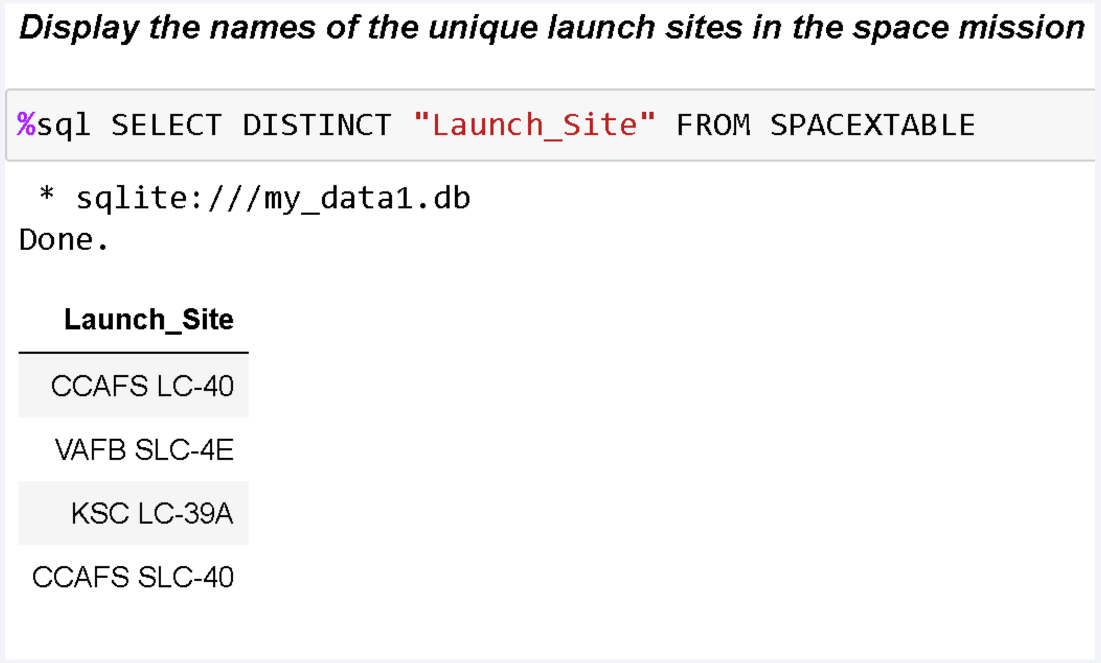
Launch Success Yearly Trend

- This plot represents the trend in the average launch success rate over the years.
- It allows us to understand how SpaceX's launch success rate has evolved over time.
- By analyzing the trend, we can gain insights into the improvements or challenges faced by SpaceX in ensuring successful launches.



All Launch Site Names

- %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
- Launch_Site
  CCAFS LC-40
  VAFB SLC-4E
  KSC LC-39A
  CCAFS SLC-40
- These launch sites represent the different locations from which SpaceX has conducted its launches, as recorded in the "SPACEXTABLE" table.



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

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

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

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

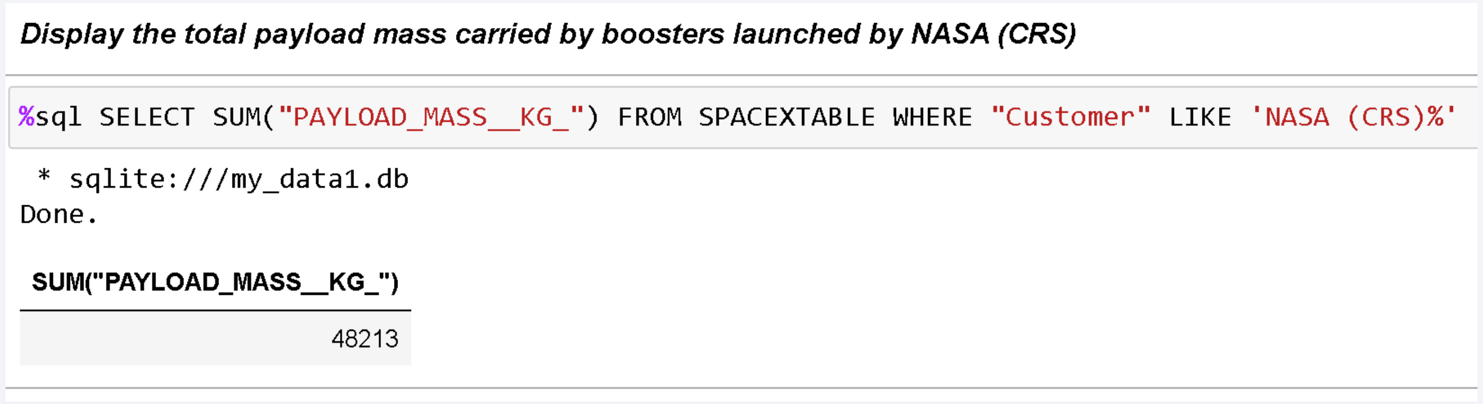
Launch Site Names Begin with 'CCA'

- SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
- Result

```
Launch_Site
CCAFS LC-40
CCAFS SLC-40
```

- When using the LIMIT 5 clause in the query, it means the query will return at most 5 records that match the criteria. If there are fewer than 5 records that match as in this case, where only 2 records match, then only those matching records will be returned

Total Payload Mass



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

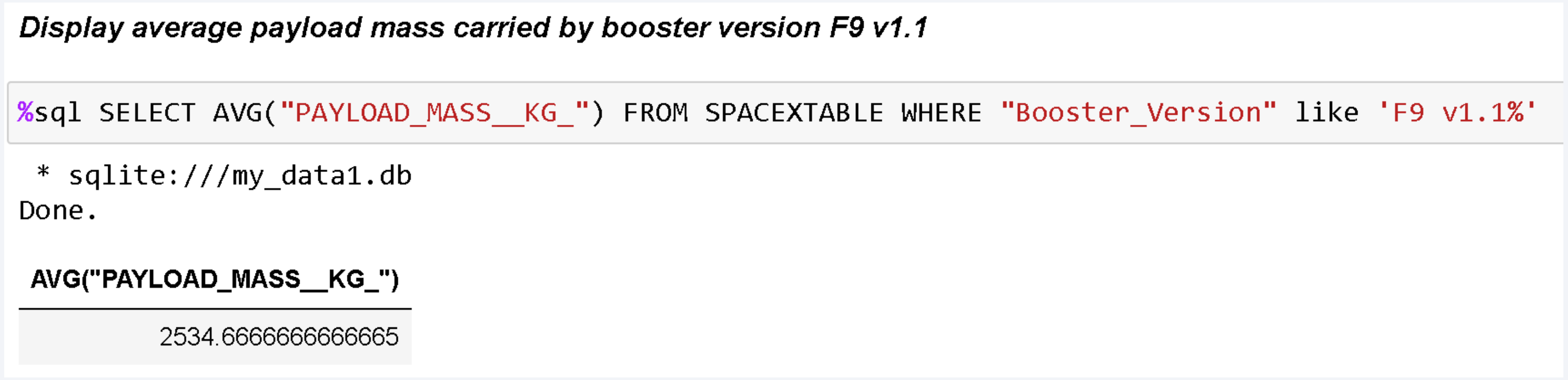
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" LIKE 'NASA (CRS)%'
```

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

SUM("PAYLOAD_MASS__KG_")
48213

- The query filters the records in the SPACEXTABLE where the "Customer" column value starts with "NASA (CRS)". It then sums up the values in the "PAYLOAD_MASS__KG_" column

Average Payload Mass by F9 v1.1



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

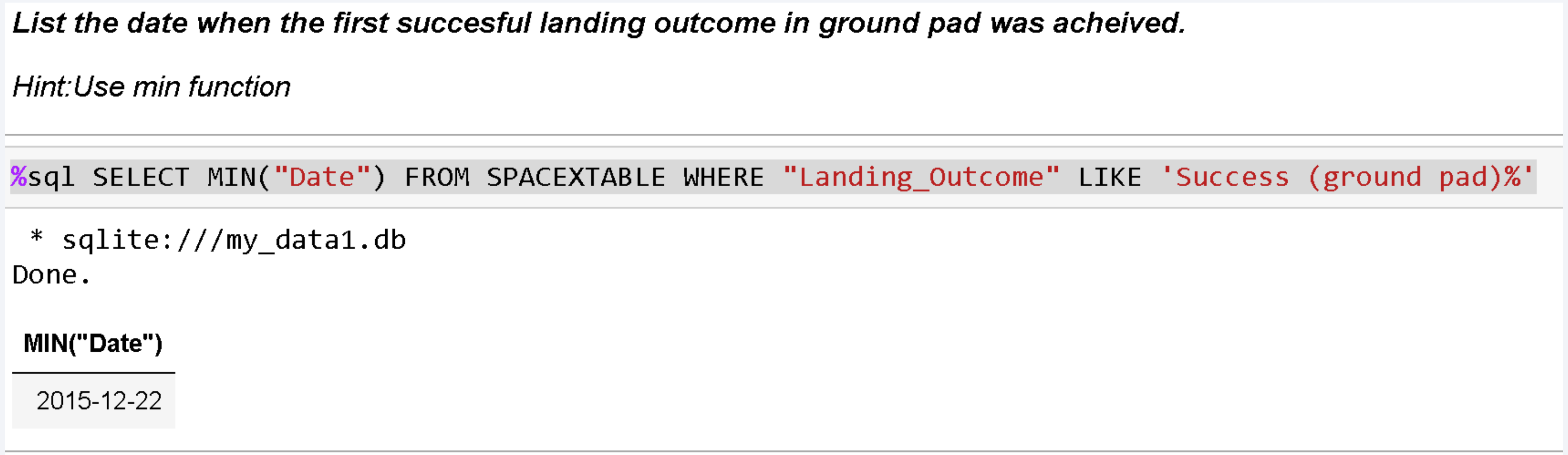
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" like 'F9 v1.1%'
```

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

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

- %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" like 'F9 v1.1%'
- The query shows that the booster version "F9 v1.1" has carried payloads with an average mass of approximately 2,534.67 kg.

First Successful Ground Landing Date



List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

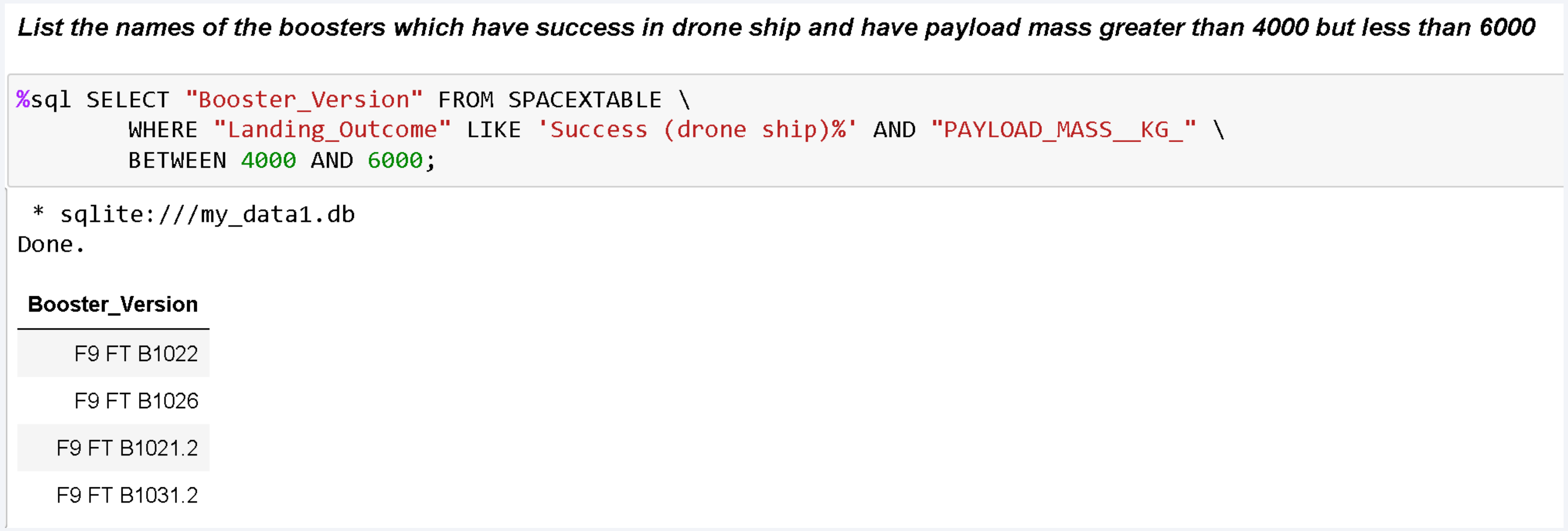
```
%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (ground pad)%'
```

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

MIN("Date")
2015-12-22

- %sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (ground pad)%'
- The query uses MIN("Date") to identify that SpaceX achieved its first successful landing on a ground pad on December 22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE \
        WHERE "Landing_Outcome" LIKE 'Success (drone ship)%' AND "PAYLOAD_MASS__KG_" \
        BETWEEN 4000 AND 6000;
```

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

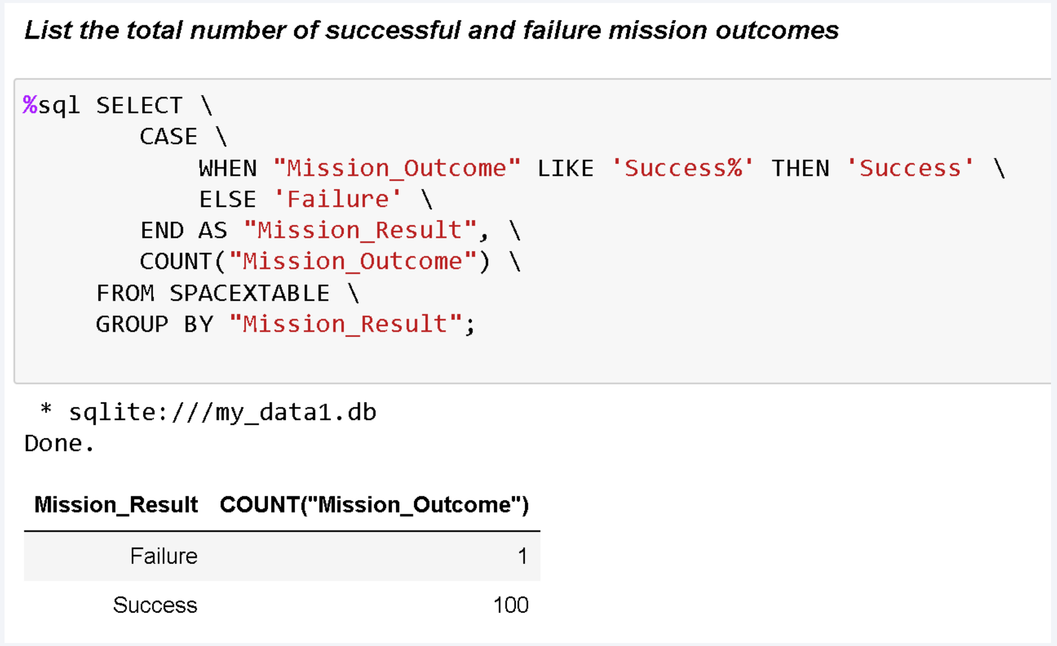
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- From the provided SpaceX dataset, it was identified that four booster versions (F9 FT B1022, F9 FT B1026, F9 FT B1021.2, and F9 FT B1031.2) successfully landed on a drone ship and carried payloads with a mass ranging between 4,000 kg and 6,000 kg. These booster versions highlight the capability of SpaceX's Falcon 9 Full Thrust boosters in safely landing on drone ships while carrying significant payload masses.

Total Number of Successful and Failure Mission Outcomes

-
  ```
  %sql SELECT \
      CASE \
        WHEN "Mission_Outcome" LIKE 'Success%' THEN 'Success' \
        ELSE 'Failure' \
      END AS "Mission_Result", \
      COUNT("Mission_Outcome") \
  FROM SPACEXTABLE \
  GROUP BY "Mission_Result";
  ```

- Out of the missions recorded in the "SPACEXTABLE", 100 were successful, and only 1 was a failure. This showcases a high success rate for SpaceX missions.



List the total number of successful and failure mission outcomes

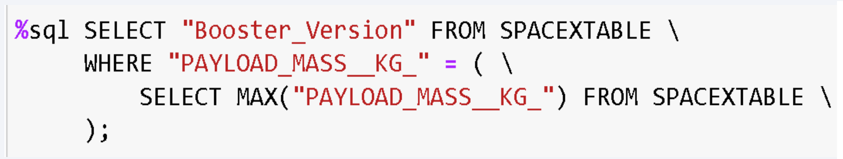
```
%sql SELECT \
        CASE \
            WHEN "Mission_Outcome" LIKE 'Success%' THEN 'Success' \
            ELSE 'Failure' \
        END AS "Mission_Result", \
        COUNT("Mission_Outcome") \
     FROM SPACEXTABLE \
     GROUP BY "Mission_Result";
```

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

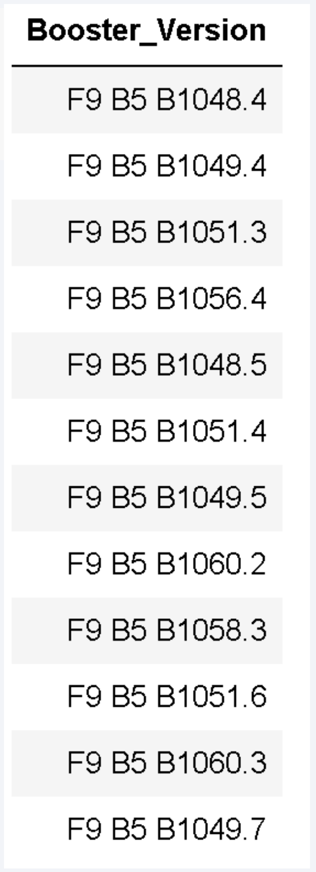
Mission_Result	COUNT("Mission_Outcome")
Failure	1
Success	100

Boosters Carried Maximum Payload

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



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



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

```
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
```

- The subquery SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE determines the maximum payload mass from the entire "SPACEXTABLE".

- The main query then selects the "Booster_Version" from the "SPACEXTABLE" where the "PAYLOAD_MASS__KG_" matches this maximum value.

- Given that multiple booster versions have carried this maximum payload mass, the result lists all of these boosters. This indicates that several SpaceX booster versions have achieved the capability to carry the highest payload mass recorded in the table.

2015 Launch Records

- Since sqlLite has no function to convert month number to month name, I must add a column to the DB with the month names.
- query = "SELECT * FROM SPACEXTABLE"
  df = pd.read_sql(query, con)
  df['Date'] = pd.to_datetime(df['Date'])
- Create a new column with month names
  df['month_name'] = df['Date'].dt.strftime('%B')
- Upload the updated dataframe back into the SQL database then run the final query.
  df.to_sql('SPACEXTABLE', con, if_exists='replace', index=False)
- %sql SELECT "month_name" AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" \
  FROM SPACEXTABLE \
  WHERE "Landing_Outcome" LIKE 'Failure (drone ship)%' AND substr("Date", 1, 4) = '2015';

- The query identifies that in 2015, there were two SpaceX missions that attempted to land on a drone ship but unfortunately failed. Both of these missions launched from the CCAFS LC-40 site and used the F9 v1.1 booster versions.:

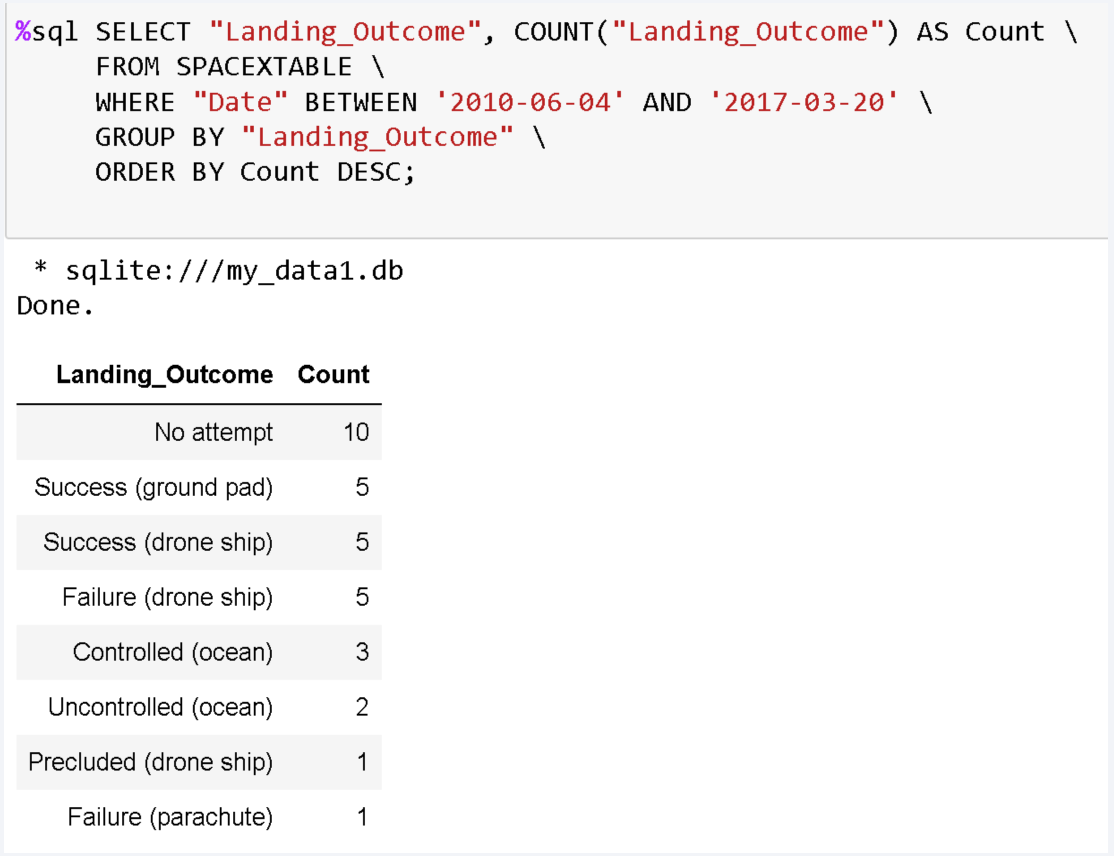
Month	Landing_Outcome	Booster_Version	Launch_Site
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- %sql SELECT "Landing_Outcome", \
  COUNT("Landing_Outcome") AS Count \
  FROM SPACEXTABLE \
  WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' \
  GROUP BY "Landing_Outcome" \
  ORDER BY Count DESC;

- The query provides a ranked overview of SpaceX's landing outcomes for missions between
  June 4, 2010, and
  March 20, 2017.
  The most common outcome was "No attempt" with 10 occurrences

- GitHub Folder - Hands-on Lab - Complete EDA with SQL



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

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

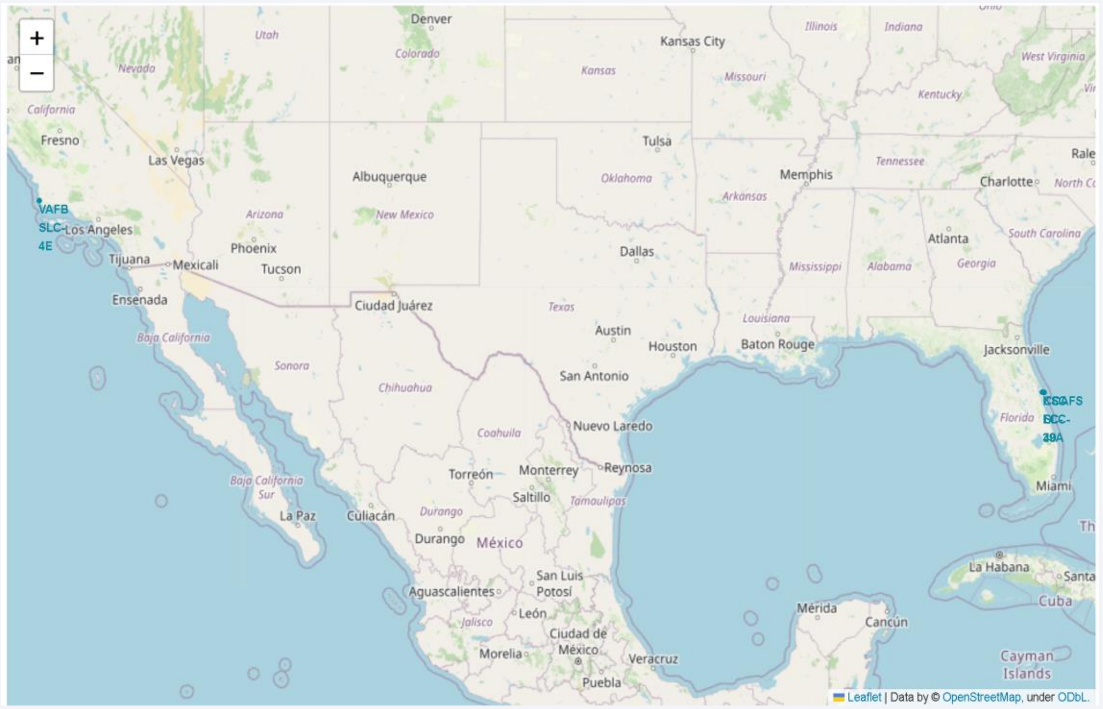
Landing_Outcome	Count
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

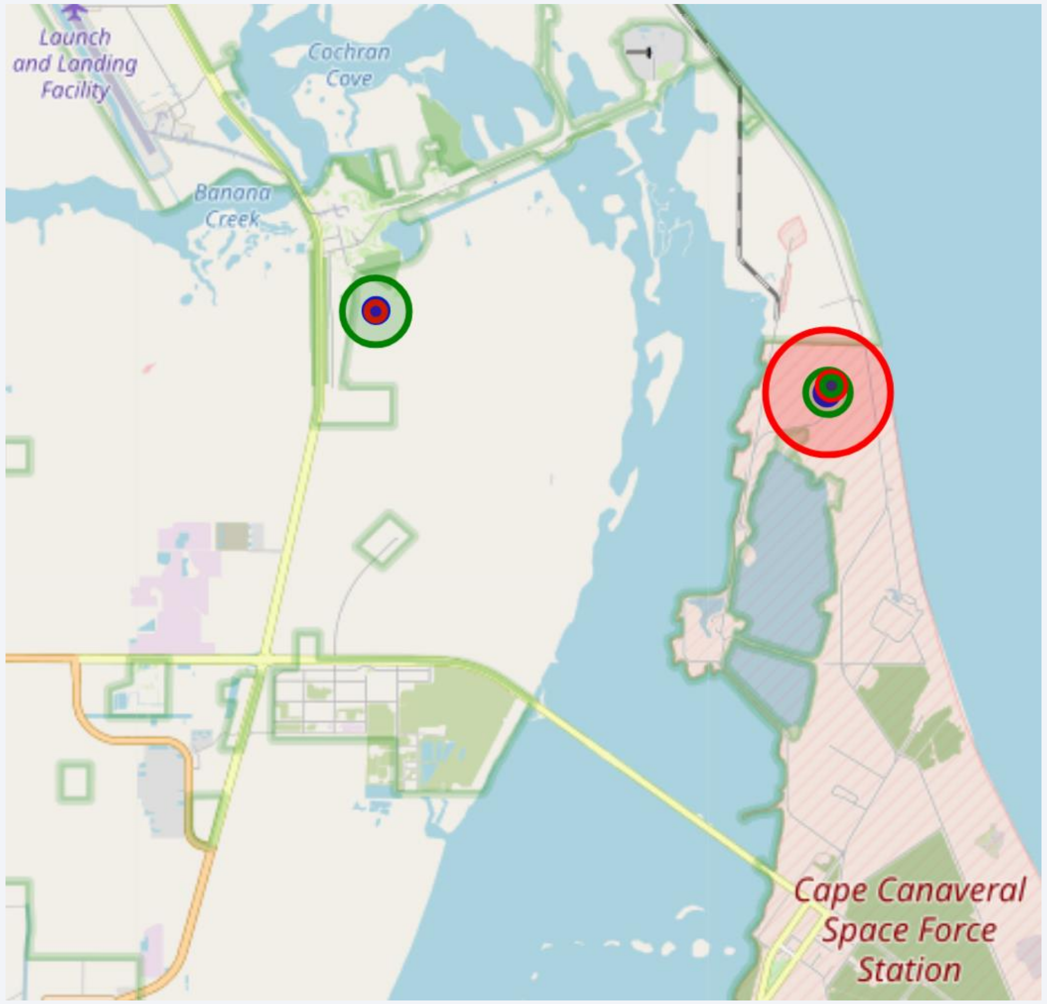
Launch Site Locations

- None of SpaceX's launch sites are close to the Equator. launch sites near the Equator benefit from Earth's rotation.
- Launch sites are located on the coasts to ensure rockets' trajectories are over open water, minimizing risks to populated areas in case of launch failures.
- Coastal sites also offer flexibility in choosing launch directions, essential for different satellite orbits.
- Being near coasts facilitates the transportation of large rocket components, by sea.



SpaceX Launch Outcomes by Site

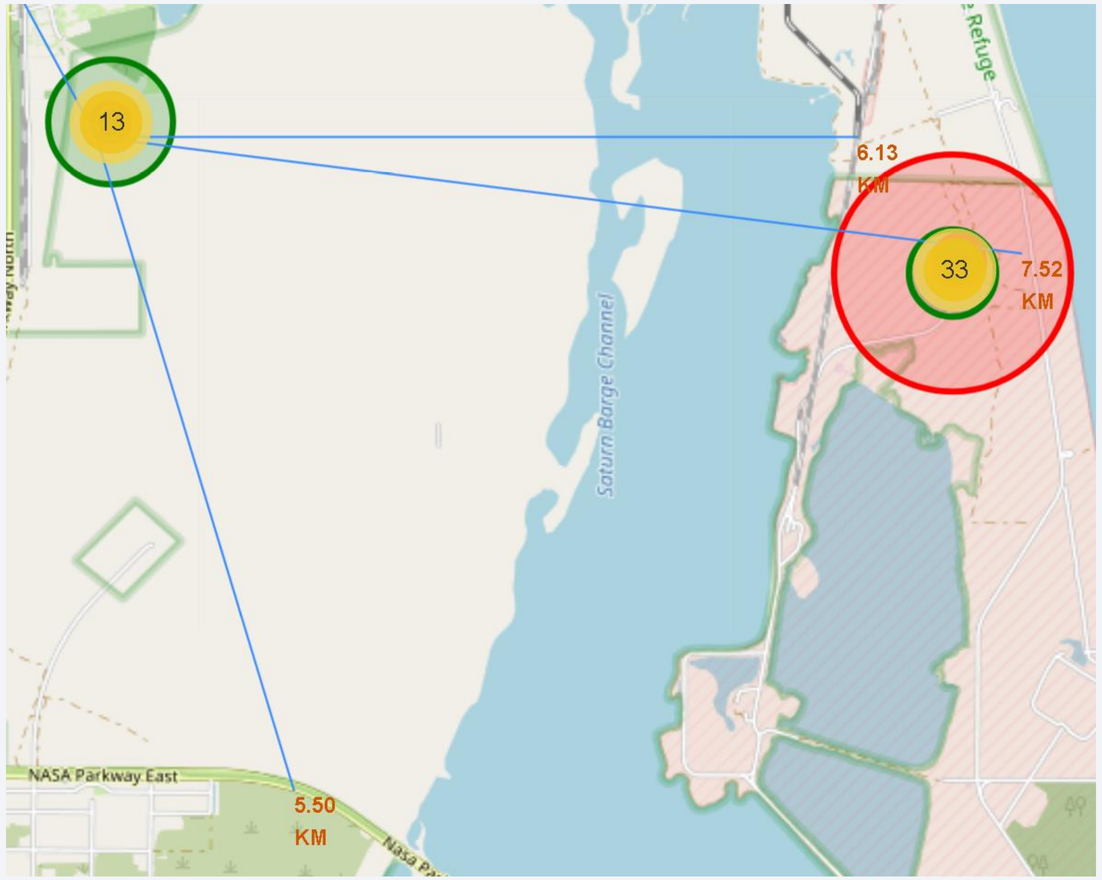
- Blue Markers: These represent the locations of SpaceX launch sites. The presence of a blue marker at 'Cape Canaveral Space Force Station' indicates it's one of the launch sites.
- Green Circles: These denote successful launches. The size of the circle corresponds to the number of successful launches; a larger circle indicates more successful launches.
- Red Circles: These denote launch failures. The size of the circle relates to the number of failed launches; a larger circle indicates more failures.
- By observing the size and color of the circles around 'Cape Canaveral Space Force Station', one can gauge the success rate and volume of launches, giving insights into the performance and reliability of launches from that specific site

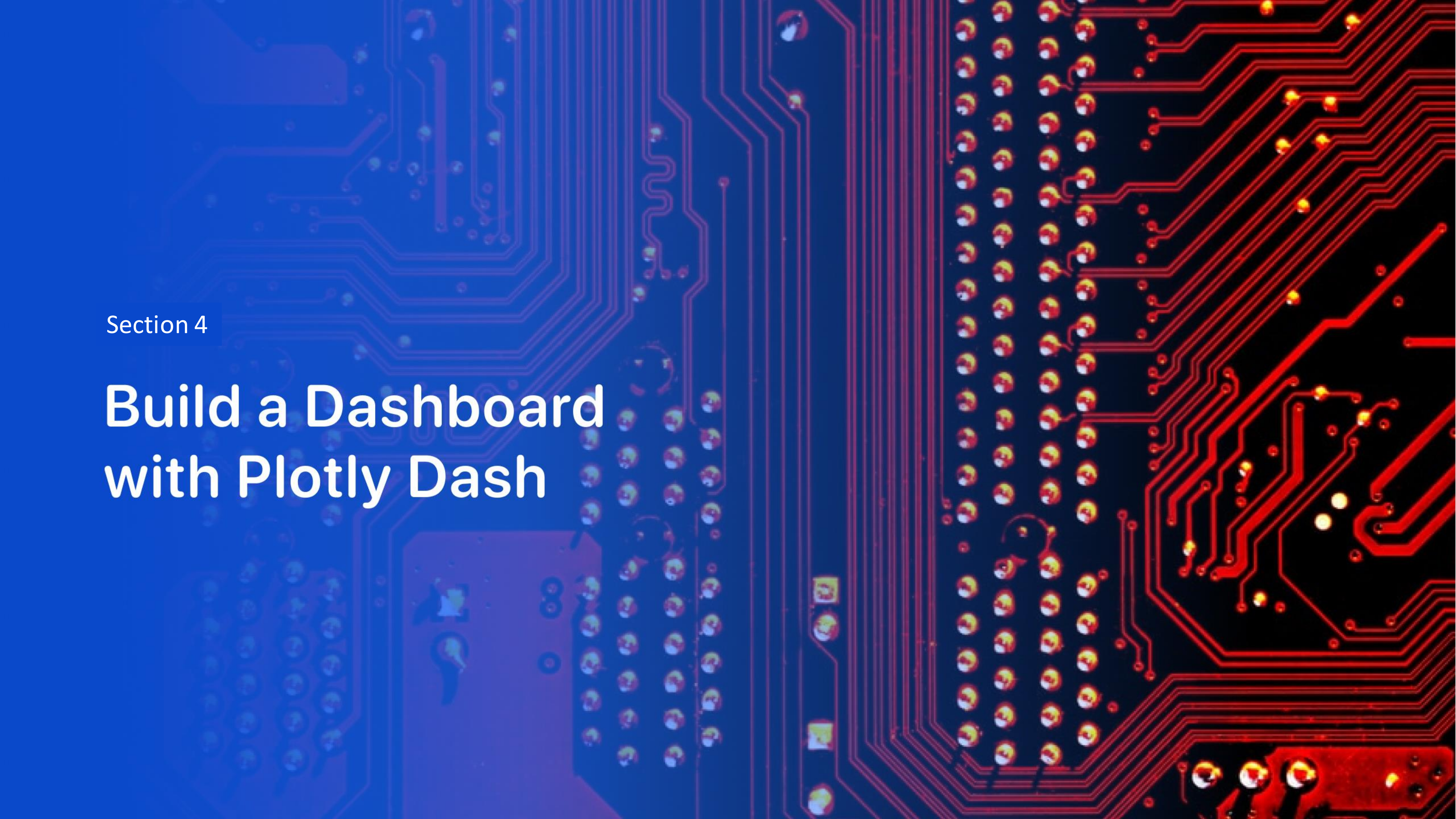

Launch and Landing Facility
Cochran Cove
Banana Creek
Cape Canaveral Space Force Station

Launch site Proximity to Highway, Rail and Coast

- Launch sites being near railways, roads and coastlines implies logistical convenience. Railways, major roads and Coastlines facilitate easier transportation of heavy and large rocket components. In addition Proximity to coastlines, offers safety (trajectory over water) and flexibility in launch azimuths.

- Being away from cities is a safety measure. In case of launch anomalies or failures, risks to populated areas are minimized.



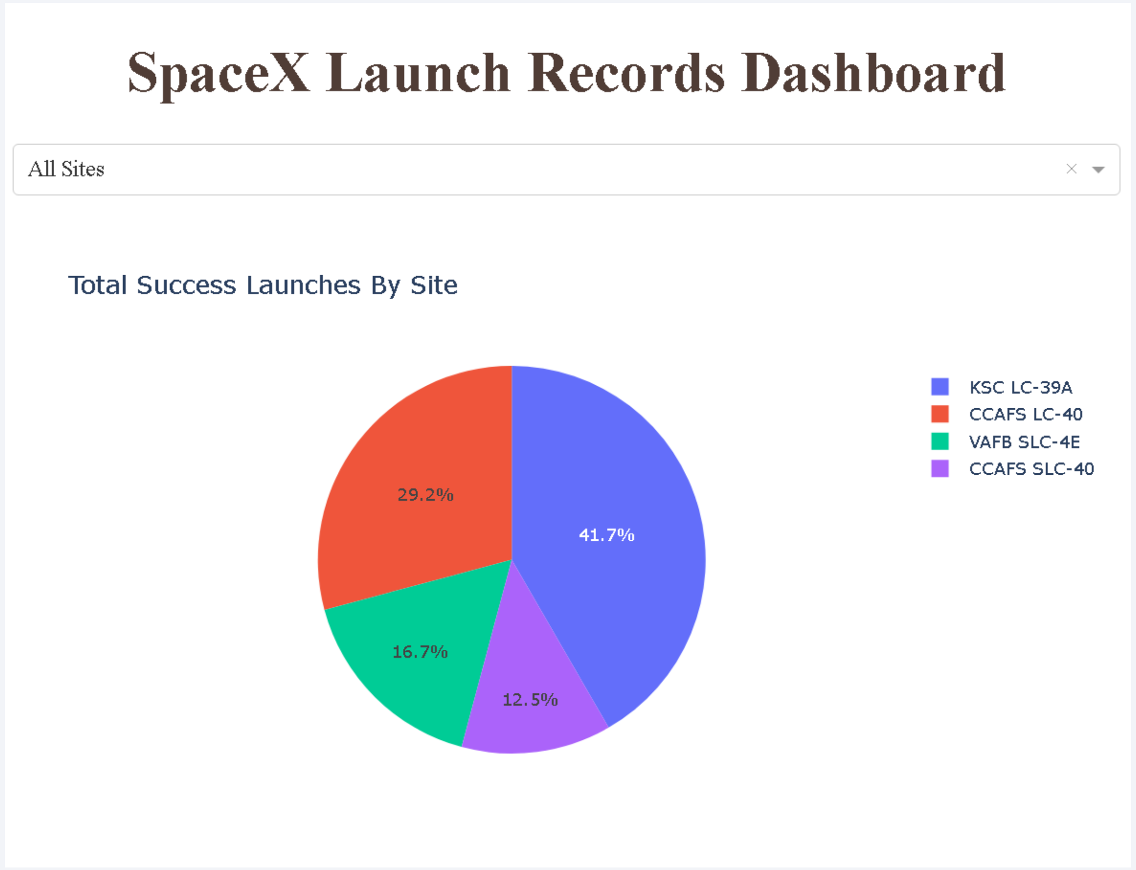


Section 4

Build a Dashboard with Plotly Dash

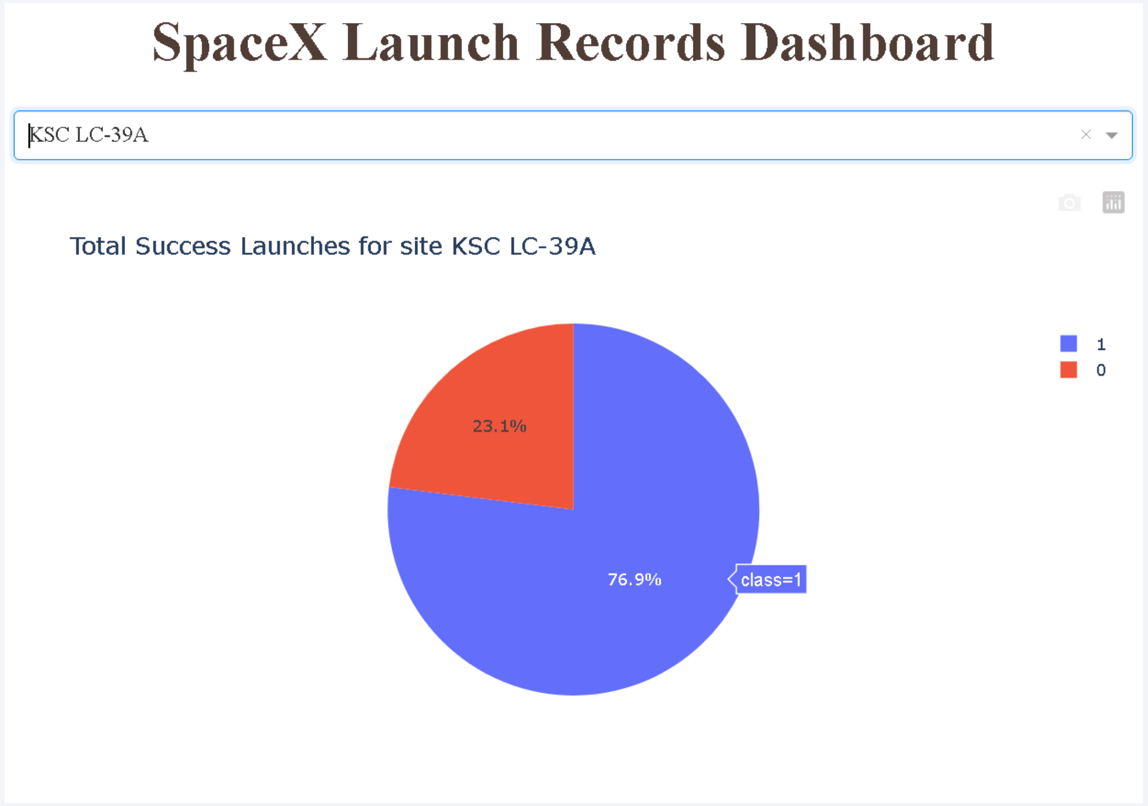
Total Successful Launches for All Sites

- The pie chart provides a breakdown of the total successful launches for each launch site.
- Each segment of the pie chart represents a different launch site, with the site's name and the count of successful launches displayed.
- Provides an overview of the success distribution across all launch sites.



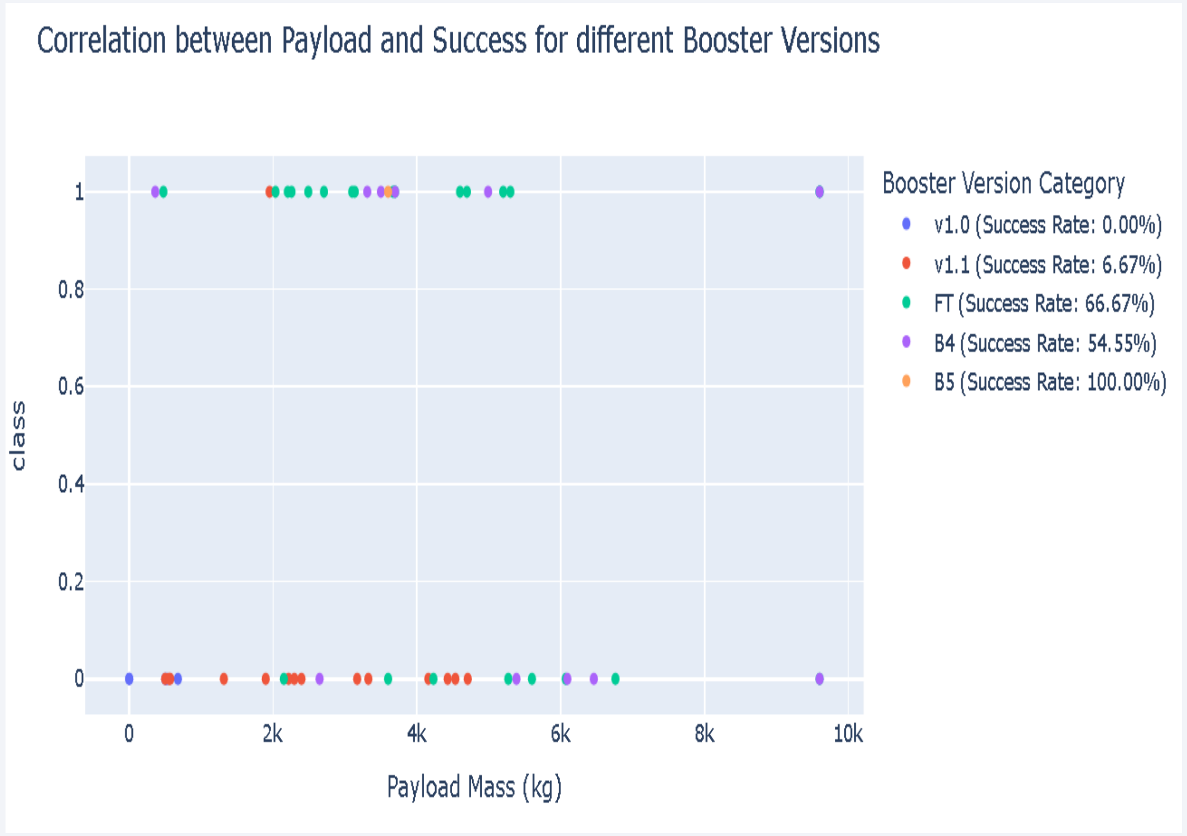
With Most Successful Launch Site

- High Success Rate: "KSC LC-39A" has a notable success rate, as indicated by its success ratio. This suggests that launches from this site have been largely successful, making it a reliable site for SpaceX.



Payload Success Vs Booster Version

- The Booster Version B5 is the most successful across all launch sites.
- Launches carrying payloads in the range of 7,500 kg to 10,000 kg tend to have higher success rates.
- KSC LC-39A: This launch site has the highest success rate at approximately 76.92%.

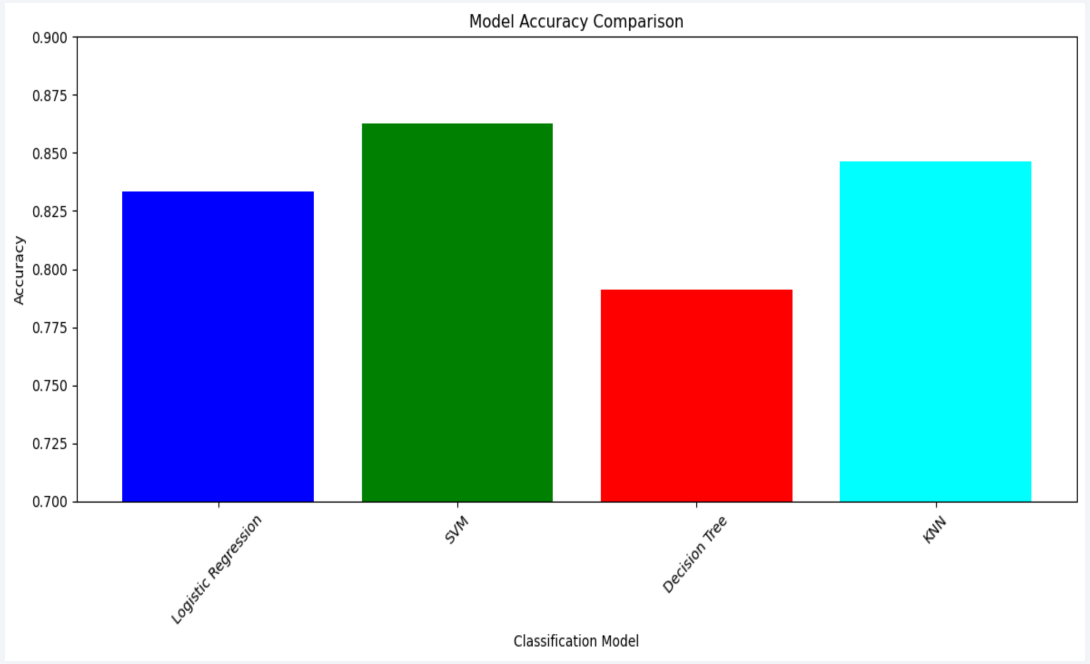


Section 5

Predictive Analysis (Classification)

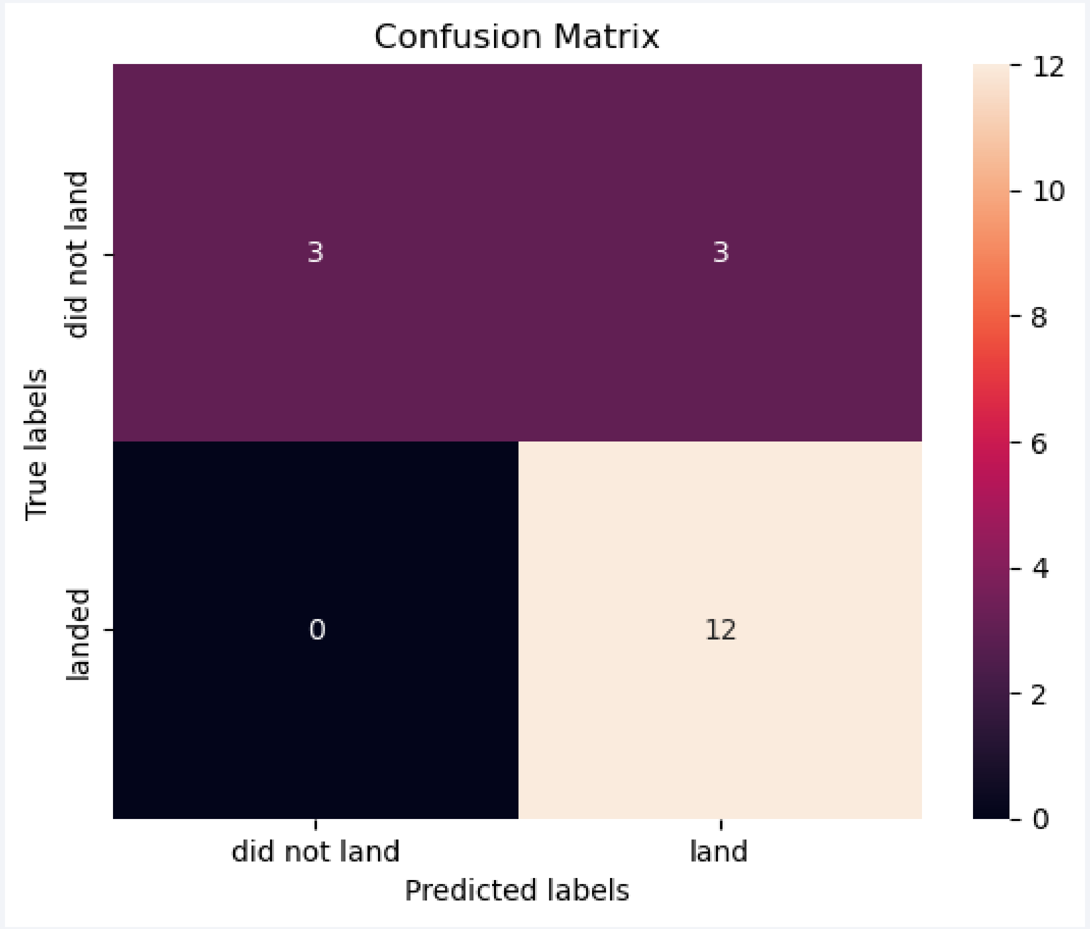
Classification Accuracy

- SVM has the highest accuracy at 86.25%.
- Logistic Regression follows with 83.33%.
- KNN is next with an accuracy of 84.64%.
- Decision Tree has the lowest accuracy at 78.93%.



Confusion Matrix for SVM

- **Confusion Matrix Results:**
  - **True Positives:** The model correctly predicted 3 rocket landings.
  - **True Negatives:** The model never correctly predicted when a rocket wouldn't land.
  - **False Positives:** The model incorrectly forecasted 3 landings that didn't happen.
  - **False Negatives:** The model missed 12 actual landings, predicting they wouldn't land.
- **Implications:**
  - The model is fairly accurate at predicting when rockets will land.
  - However, it struggles to correctly predict when rockets won't land.



Conclusions

- **Confusion Matrix Results:**
  - **True Positives (3):** This indicates that, the actual outcome (rocket landing) and the model's prediction aligned perfectly.
  - **True Negatives (0):** The absence of true negatives (a value of 0) suggests that the model has challenges accurately identifying situations where the rocket will not land.
  - **False Positives (3):** These are instances where the model predicted a rocket landing, but in reality, the rocket did not land. This implies that, in three cases, the model was overly optimistic, anticipating successful landings when they did not occur.
  - **False Negatives (12):** A high number of false negatives suggest that the model was quite conservative or cautious. In twelve instances, the model predicted that the rocket would not land successfully, when, in fact, they did. This could be viewed as the model erring on the side of caution. This is very desirable because rocket mistakes cost a lot of money
- **In summary**
  - Given the high cost and risks associated with rocket launches, the model's inclination towards false negatives (erring on the side of caution) can be seen as a safety measure, anticipating potential failures even when they don't occur.
  - The model's primary strength lies in its conservative approach, signaling alerts even when things go right.
  - In high-stakes space industry, caution can be crucial, as it ensures safety and prevents potential losses.

Appendix

- All relevant assets are included in the GitHub Repository link below including :
- SQL queries and CSV data set Files
- Charts and Images
- Notebooks and Python code
- Applied-Data-Science-Capstone
- **Applied-Data-Science-Capstone**
  - https://github.com/jeremyharkness/Applied-Data-Science-Capstone

Thank you!