IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pavel Danilov
27 Oct 2024

Outline

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

Executive Summary

This project aimed to analyze SpaceX launch data to identify trends, understand factors impacting success, and gain insights from historical data.

Key Findings:

- The first successful ground landing occurred on June 4, 2010.
- NASA's CRS missions carried an estimated total payload of 45.6 tons.
- The average payload mass for F9 v1.1 booster missions is approximately 2.9 tons.
- The average success rate of landing outcomes is about 66%, showing improvement over time.

Approach:

- Data collection via API and web scraping
- Data wrangling, exploratory data analysis (EDA), and predictive modeling to optimize hyperparameters for SVM, Classification Trees, and Logistic Regression models.

Introduction

- **Project Context**: Could be more concise:
  - SpaceX’s reusable rockets drive down space travel costs, making launch success analysis critical for advancing reliability and technology for missions.

- **Data Source**:
  - Historical SpaceX launch data from Wikipedia and recent data via SpaceX API.

- **Problem Statement**:
  - "Which factors contribute to launch and landing success?"
  - "How have payload trends evolved over time?"
  - "Which model and hyperparameters best predict landing success?"

Section 1

Methodology

Methodology

Executive Summary

- Data collection:
  - I used BeautifulSoup to scrape Wikipedia tables, extracting launch details into a structured DataFrame.
  - Using requests, pandas, and numpy, I built helper functions to collect and organize SpaceX API data.
- Data wrangling:
  - Deleted rows with information about "BoosterVersion" = "Falcon 1"
  - In the 'PayloadMass' column, I replaced all zeros and 'NaN' values with the column's mean.
  - Determine the training labels

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Using the library sqlie3, I could save the dataframe into a ".db" file, and then quering it with SQL syntax to perform EDA
- Perform interactive visual analytics using Folium and Plotly Dash
  - Using folium library, and the geographic coordinates of the launch sites I could label them on the map, and then add the markers for all launch records
  - Using Plotly Dash I've built an dashboard application for users to perform interactive visual analytics on SpaceX launch data in real-time
- Perform predictive analysis using classification models
  - Using sklearn as base library, the base goal is to find best hyperparameters for SVM, Classification Trees and Logistic Regression, and find which method performs best using test data

Data Collection

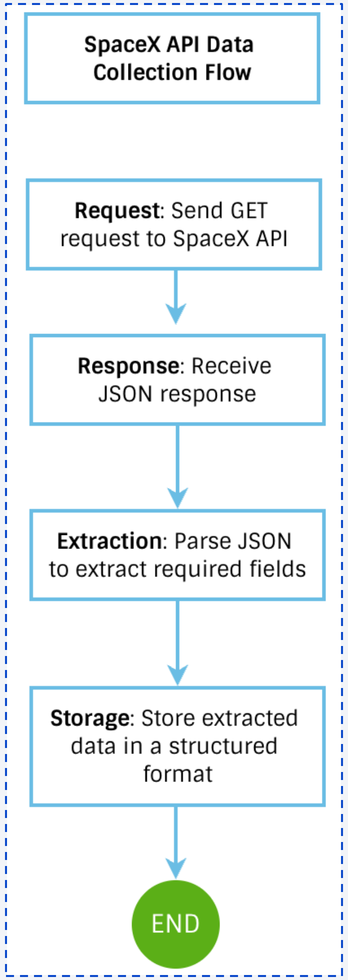
- **Web Scraping**: Used to gather historical launch data from Wikipedia.
    - **Tool**: BeautifulSoup (Python package) for parsing HTML.
    - **Source**: Wikipedia's page on Falcon 9 and Falcon Heavy launches.
    - **Output**: Structured data on each launch, stored in a DataFrame for further analysis.
- **API Requests**: Utilized SpaceX's public API for recent and detailed launch information.
    - **Tool**: requests library to send HTTP GET requests.
    - **Source**: SpaceX REST API for dynamic data collection.
    - **Output**: JSON data parsed into a structured DataFrame.

Data Collection – SpaceX API

Key Steps:

- **API Call**: HTTP GET request to the SpaceX API endpoint.
- **Response Handling**: Parse JSON response to access relevant fields.
- **Data Extraction**: Extract key data points (e.g., launch site, rocket type, launch date).
- **Storage**: Organize data into a DataFrame.

- GitHub URL of the completed SpaceX API calls notebook
  https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/2.jupyter-labs-spacex-data-collection-api.ipynb

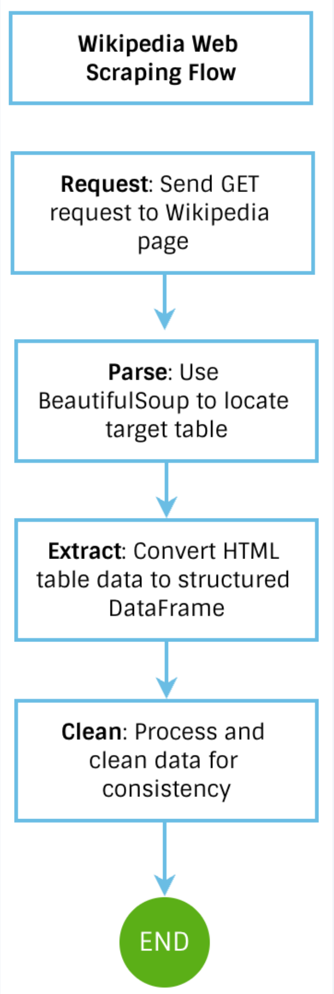


Data Collection - Scraping

Key Steps:

- **Page Request**: Send HTTP request to Wikipedia URL.
- **HTML Parsing**: Use BeautifulSoup to locate the HTML table with launch records.
- **Data Extraction**: Retrieve data, clean unnecessary HTML tags, and convert it to a DataFrame.
- **Data Cleaning**: Standardize data types and handle any missing or malformed entries.

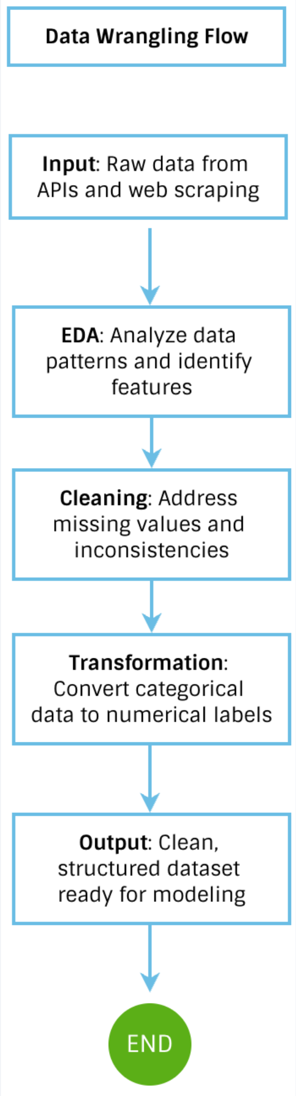
- GitHub URL of the completed web scraping notebook https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/1.jupyter-labs-webscraping.ipynb



Data Wrangling

- Exploratory Data Analysis (EDA):
  - Conducted initial analysis to understand data patterns and structure.
  - The different outcomes for Falcon 9 landings were mapped to binary labels (1 for successful landings, 0 for unsuccessful).
- Data Cleaning:
  - Removed rows with 'Falcon 1' booster versions and replaced NaNs/zeros in 'PayloadMass' with the column mean.
  - Missing Values: Calculated the percentage of missing values in each attribute to assess data quality.
- Label Creation:
  - Converted landing outcomes to binary labels to prepare for classification tasks.
- Data Transformation:
  - Ensured that the DataFrame was structured for subsequent analyses and model training.

- GitHub URL of the completed data wrangling notebook: https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/3.labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- **Flight Number vs. Payload Mass**:
  - **Purpose**: To investigate the relationship between the launch attempt sequence and payload mass on landing success.
  - **Insight**: Showed that higher flight numbers generally correlate with higher landing success, suggesting improvements in reliability over time.
- **Launch Success by Outcome**:
  - **Purpose**: Visualize successful vs. unsuccessful landing outcomes based on various factors.
  - **Insight**: Provided a clear distribution of successful landings, assisting in understanding potential influencing factors.
- **Correlation Heatmap**:
  - **Purpose**: Highlight relationships between multiple numeric features (e.g., payload mass, flight number).
  - **Insight**: Identified which features might be more predictive of landing outcomes.
- **Libraries Used**: Pandas for data manipulation, Matplotlib and Seaborn for visualizations.

- GitHub URL of your completed EDA with data visualization notebook: https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/5.edadataviz.ipynb

EDA with SQL

- **Initial Data Preview**:
  - Queried the first 10 rows of SPACEXTBL to understand the structure and sample values.
- **Count of Total Launches**:
  - Used COUNT to determine the total number of launches in the dataset.
- **Launch Success Rate by Site**:
  - Grouped data by launch site and calculated the number of successful landings for each, providing insights into site-specific success rates.
- **Payload and Orbit Analysis**:
  - Queried payload types across different orbits, helping to understand SpaceX's launch capabilities in various orbits.
- **Average Payload Mass per Orbit**:
  - Calculated the average payload mass for each orbit, useful for assessing payload limitations and preferences per mission type.
- **Yearly Launch Trends**:
  - Extracted the launch count per year to visualize trends and growth in SpaceX's launch frequency.

These SQL queries allowed us to gain critical insights into the dataset, such as launch patterns, payload distributions, and site-based success rates.

GitHub URL of your completed EDA with SQL notebook:
https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/4.jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

To visualize launch site locations and their surroundings, I used Folium to create an interactive map with various objects:

- **Markers:** Placed at each launch site to provide a visual reference for exact locations.
- **Circle Markers:** Used to denote areas of interest around each site, indicating proximity zones and possible impact areas.
- **Lines:** Drawn to connect related points, such as routes between launch sites, showing distances and connections.

These elements enhance spatial understanding, allowing users to explore geographic relations and accessibility visually.

- GitHub URL of your completed interactive map with Folium map: https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/6.lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

An interactive dashboard created with Plotly Dash allows users to filter by launch site, analyze outcomes via a pie chart, and adjust payload range dynamically:

- **Dropdown Menu for Launch Sites:** Allows users to filter data by specific launch sites, enabling a detailed view of each location.
- **Pie Chart for Launch Outcomes:** Visualizes the success rate of launches for selected sites or all sites, providing insights into overall mission success.
- **Payload Range Slider:** Adjusts the range of payload masses, dynamically updating visuals to explore correlations between payload weight and mission outcomes.

These interactive features allow users to explore key performance metrics and trends within SpaceX launch data effectively.

- GitHub URL of the completed Plotly Dash lab:
  https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/7.spacex_dash_app.py
  https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/spacex_dash_app_screenshot.png

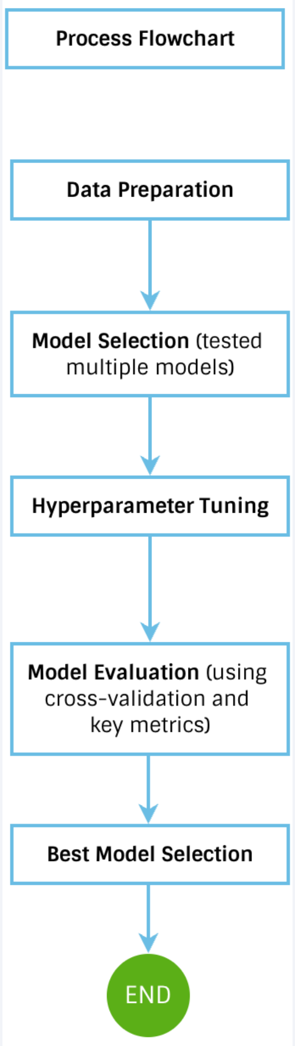
Predictive Analysis (Classification)

In this project, we aimed to build a high-performing classification model by following a structured approach to model development, evaluation, and refinement.

- **Data Preparation**: Data cleaning, feature selection, and normalization.
- **Model Selection**: Initial exploration with multiple classifiers (e.g., Logistic Regression, Random Forest, and SVM).
- **Hyperparameter Tuning**: Grid Search for optimized parameter settings.
- **Evaluation Metrics**: Measured accuracy, precision, recall, F1-score, and AUC for robust model assessment.
- **Best Model Selection**: Based on evaluation results, identifying the most effective model for predictive accuracy.

Following data preparation and model selection, I used grid search to tune hyperparameters.

- GitHub URL of the completed predictive analysis lab: https://github.com/danilovpavel1996/applied_data_science_capstone/blob/main/8.SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Results

Through this project, I analyzed SpaceX's launch data to uncover trends, assess landing success rates, and evaluate predictive model performance. Key results include:

- **Launch Success Trends**: Over time, SpaceX's launch success rate has shown significant improvement, highlighting the advancements in their reusable rocket technology.
- **Site-Based Insights**: Analysis by launch site revealed that KSC LC-39A has the highest success rate, reflecting operational efficiencies at this location.
- **Orbit-Specific Patterns**: Different orbits displayed distinct success rates, with high reliability for GEO and SSO, while GTO remained more challenging.
- **Modeling Performance**: The Decision Tree model emerged as the most accurate for predicting landing outcomes, outperforming SVM and Logistic Regression.

These insights provide a clearer picture of the factors contributing to SpaceX’s success and demonstrate the potential for predictive analytics in space operations.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

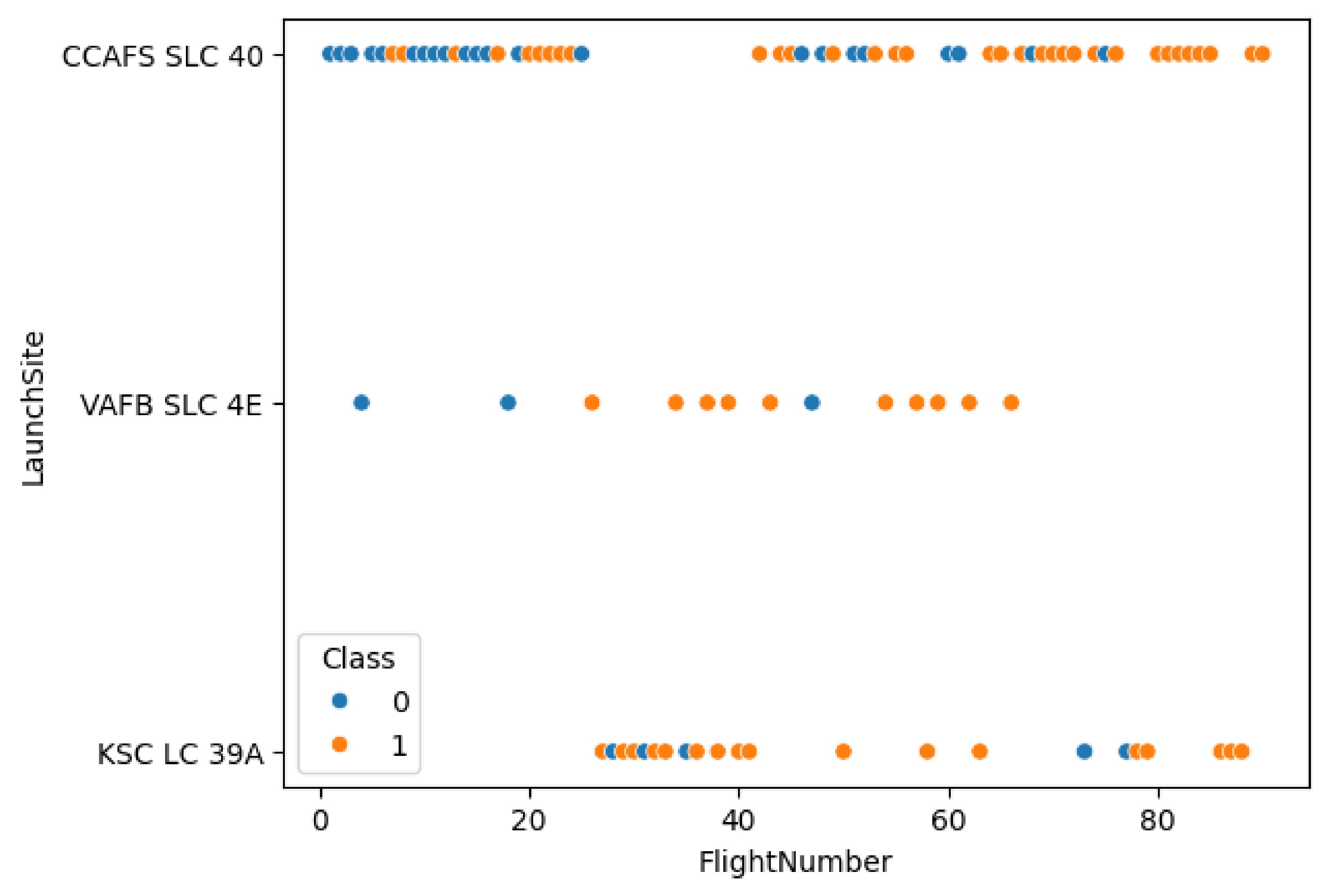
```
sns.scatterplot(y="LaunchSite",
x="FlightNumber", hue="Class", data=df)
plt.show()
```

To visualize the relationship between Flight Numbers and Launch Sites, we'll use seaborn's `catplot` function, which is perfect for categorical data visualization.

The plot will show:

- Flight Numbers on the x-axis
- Launch Sites on the y-axis
- Different colors for each launch outcome (success/failure) using the 'class' column

This visualization will help us identify patterns in launch frequency across different sites and see if there are any correlations between launch success and specific launch sites.



Payload Mass vs. Launch Site

```
sns.scatterplot(y="LaunchSite", x="PayloadMass",
hue="Class", data=df)
plt.show()
```

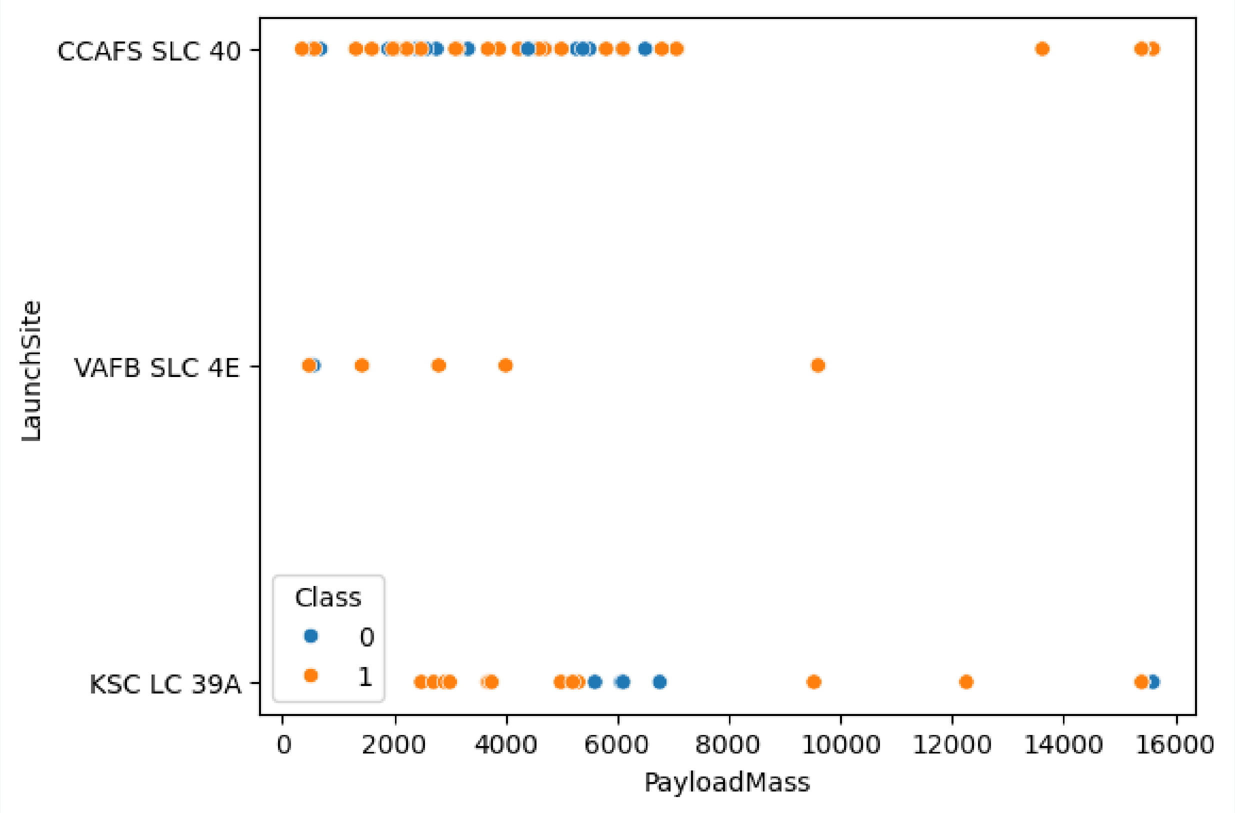
The scatter plot visualizes the relationship between Payload Mass and Launch Sites, with points color-coded by launch success (Class 1) or failure (Class 0). Key observations from this visualization:

- CCAFS SLC 40 (Cape Canaveral):
  - Handles a wide range of payload masses
  - Shows successful launches across the entire payload mass spectrum
  - Capable of launching both light and heavy payloads (up to ~15,000 kg)
- VAFB SLC 4E (Vandenberg):
  - Only handles lighter payload masses
  - Notable limitation: No launches with payload mass greater than 10,000 kg
  - Fewer launches overall compared to other sites
- KSC LC 39A (Kennedy Space Center):
  - Similar to CCAFS, handles a diverse range of payload masses
  - Shows both successful and failed launches
  - Capable of handling heavy payloads

The plot was created using seaborn's scatterplot function with:

- x-axis: PayloadMass (showing the mass distribution)
- y-axis: LaunchSite (showing the three launch locations)
- hue: Class (orange for successful launches, blue for failures)

This visualization effectively shows how different launch sites are utilized for various payload masses and their success rates.

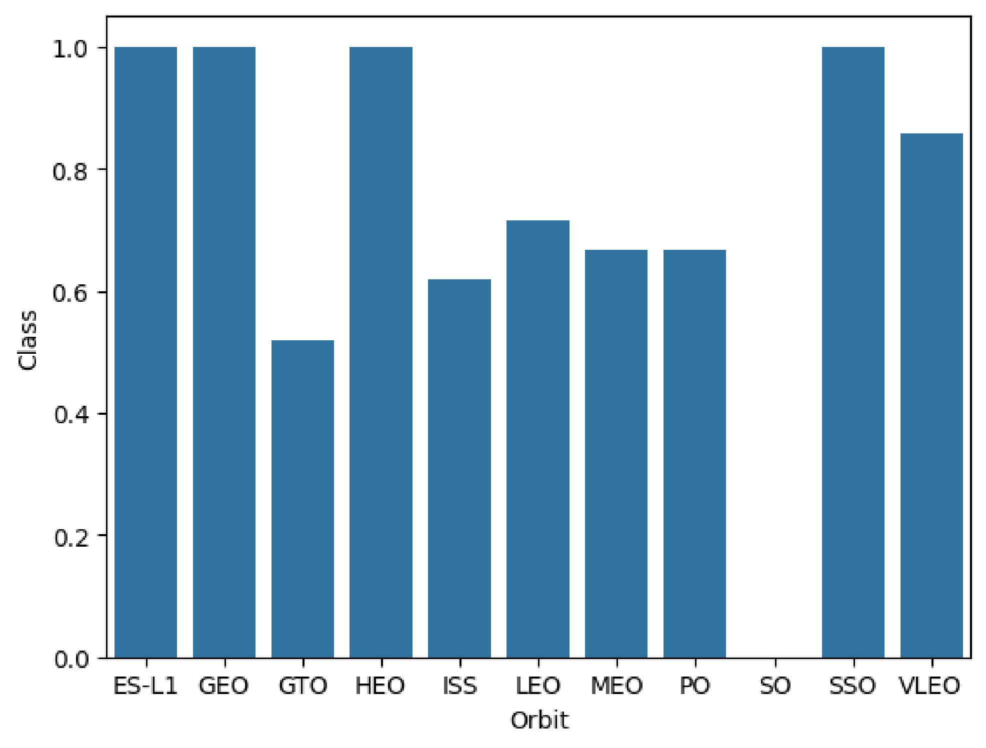


Success Rate vs. Orbit Type

```
mean_class_by_orbit =
df.groupby('Orbit')['Class'].mean().reset_index()
sns.barplot(x='Orbit', y='Class', data=mean_class_by_orbit)
```

The bar chart visualizes the success rate of SpaceX launches across different orbit types. The y-axis shows the success rate (from 0 to 1, where 1 represents 100% success), and the x-axis shows the different orbit types. Key observations:

- Perfect Success Rate (100%):
    - ES-L1 (Earth-Sun Lagrange Point 1)
    - GEO (Geostationary Earth Orbit)
    - SSO (Sun-Synchronous Orbit)
    - HEO (Highly Elliptical Orbit)
- High Success Rate (80-90%):
    - VLEO (Very Low Earth Orbit): ~85% success rate
- Moderate Success Rate (60-75%):
    - ISS (International Space Station orbit)
    - LEO (Low Earth Orbit)
    - MEO (Medium Earth Orbit)
    - PO (Polar Orbit)
- Lower Success Rate (50-60%):
    - GTO (Geosynchronous Transfer Orbit) shows the lowest success rate at around 50%



The visualization was created by:

- Grouping the data by orbit type
- Calculating the mean success rate (Class) for each orbit
- Using seaborn's barplot function to display the results

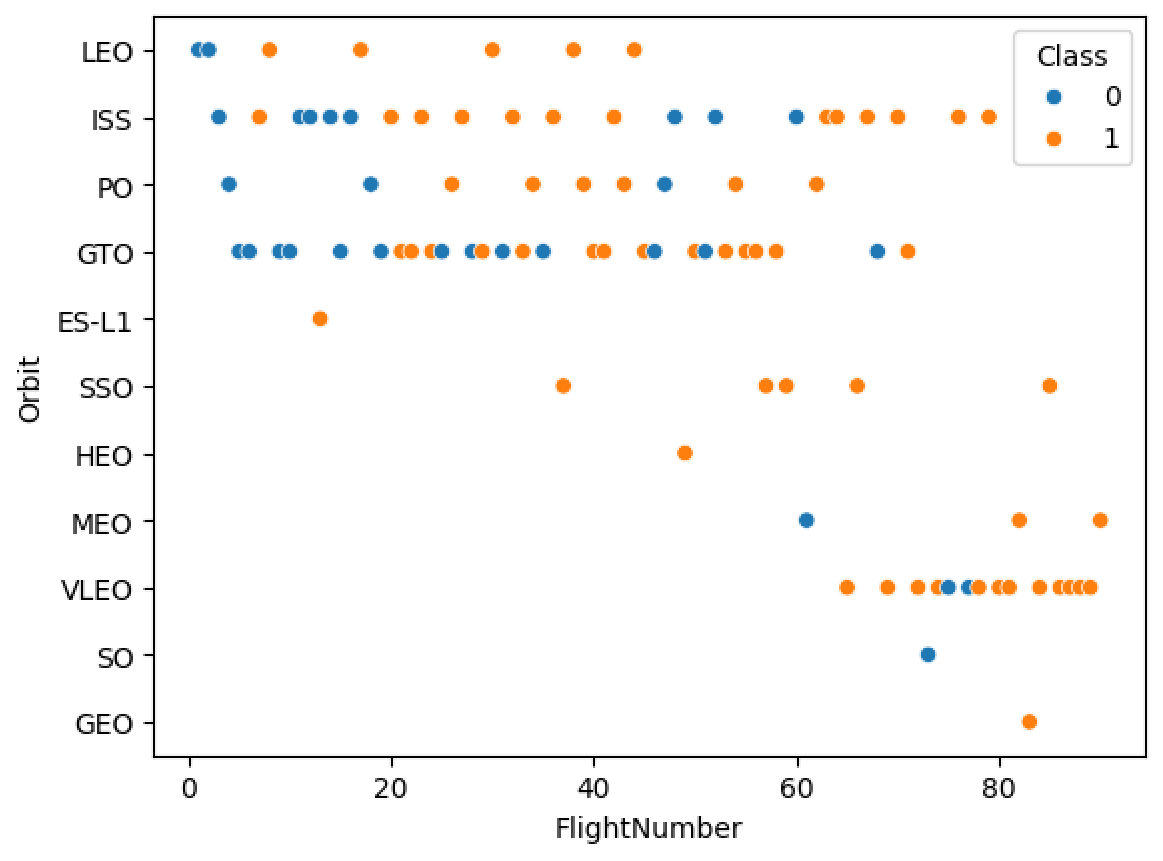
This chart helps identify which orbit types have been more challenging for SpaceX launches and which have been consistently successful.

Flight Number vs. Orbit Type

```
sns.scatterplot(x='FlightNumber', y='Orbit', hue='Class',
data=df)
```

This scatter plot visualizes the relationship between Flight Numbers and Orbit Types, with launches color-coded by success (orange, Class 1) and failure (blue, Class 0). Several interesting patterns emerge:

- Low Earth Orbit (LEO):
  - Earlier flights show more failures
  - Success rate improves with later flight numbers
  - Relatively fewer launches compared to other orbits
- International Space Station (ISS):
  - Consistent presence throughout SpaceX's flight history
  - Mixed success early on, but shows improvement over time
  - Regular launches across different flight numbers
- Geosynchronous Transfer Orbit (GTO):
  - High frequency of launches
  - Success/failure pattern appears random
  - No clear correlation between flight number and success rate
- Very Low Earth Orbit (VLEO):
  - Appears in later flight numbers (after flight 60)
  - High concentration of successful launches
  - Shows SpaceX's recent focus on this orbit type
- Other Orbits (SSO, HEO, MEO, GEO, ES-L1):
  - More scattered, occasional launches
  - Generally successful in later flights
  - Lower frequency compared to main orbit types



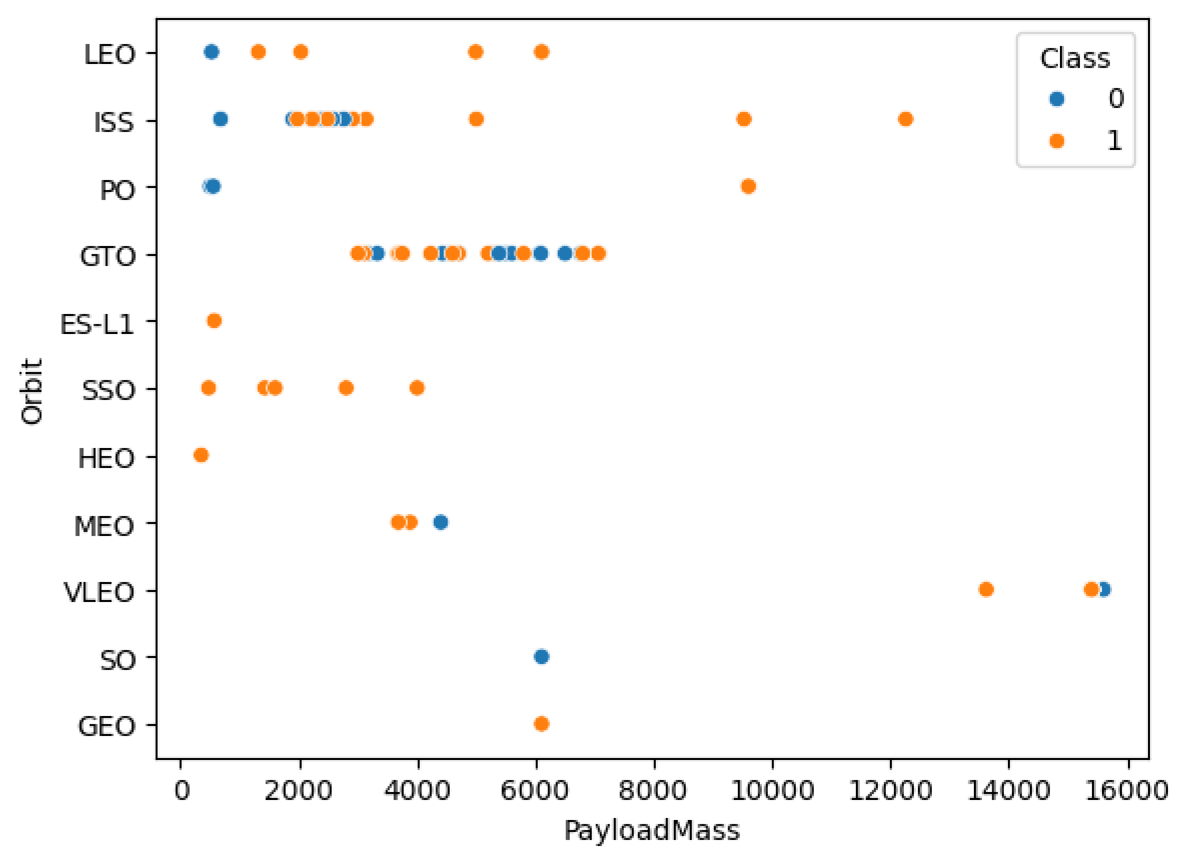
The visualization effectively shows SpaceX's learning curve and evolving launch patterns across different orbit types over time. There's a notable trend of improved success rates in later flights for most orbit types, suggesting technological maturation and learning from experience.

Payload vs. Orbit Type

```
sns.scatterplot(x='PayloadMass', y='Orbit', hue='Class', data=df)
```

This scatter plot reveals the relationship between Payload Mass and Orbit Types, with launches color-coded by success - orange and failure - blue. Key observations:

- Very Low Earth Orbit (VLEO):
  - Concentrated in the heavy payload range (around 15,000 kg)
  - Generally successful launches
  - Limited number of launches
- Geosynchronous Transfer Orbit (GTO):
  - Moderate payload mass range (2,000-6,000 kg)
  - Mixed success rate
  - Shows both successful and failed launches across its payload range
- International Space Station (ISS):
  - Clustered around 2,000-3,000 kg
  - High concentration of successful launches
  - Consistent payload mass range, suggesting standardized cargo missions
- Low Earth Orbit (LEO):
  - Wide range of payload masses
  - Better success rate with heavier payloads
  - Scattered distribution across mass ranges
- Other Orbits (SSO, HEO, MEO, GEO, ES-L1):
  - Generally lighter payloads
  - Mostly successful launches
  - Less frequent launches



Notable Trends:

- Heavier payloads (>10,000 kg) show better success rates in LEO, VLEO, and ISS orbits
- GTO missions show mixed success regardless of payload mass
- Different orbits have distinct preferred payload mass ranges
- Orbit SSO tend to handle lighter payloads with good success rate

Launch Success Yearly Trend

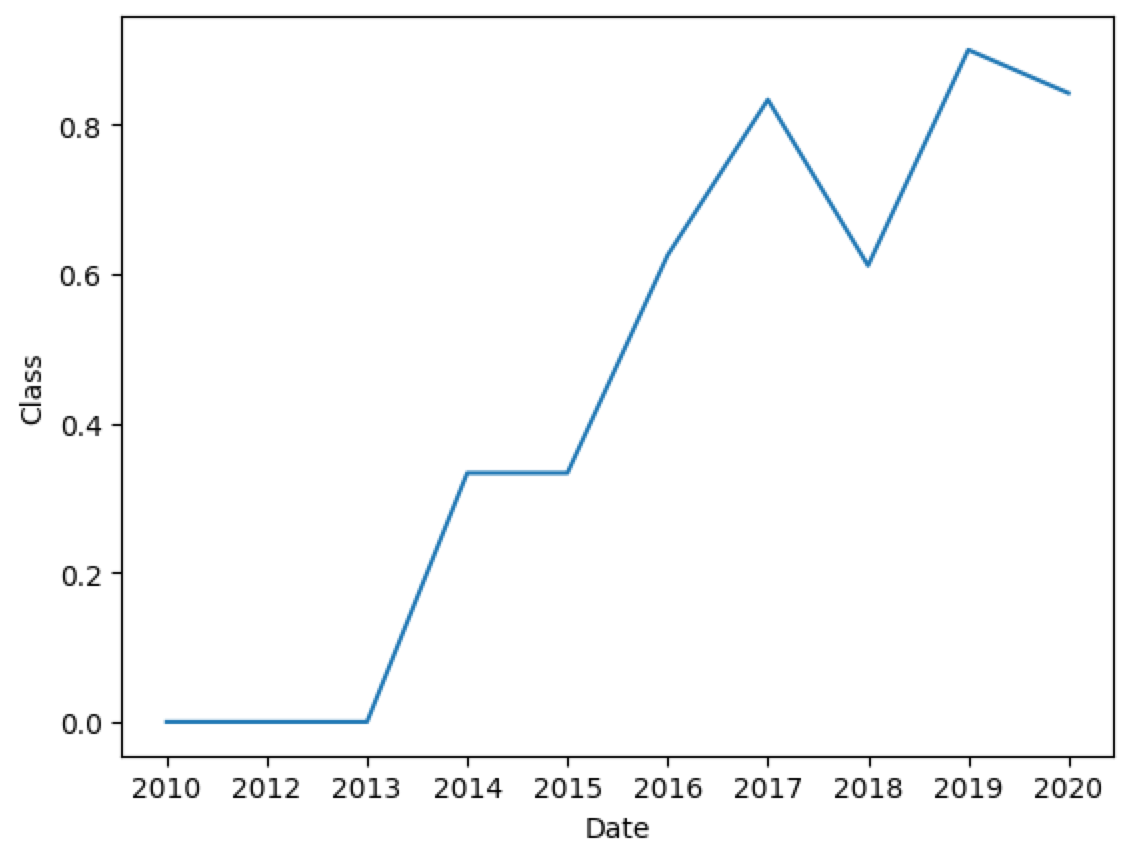
```
mean_success_rate_by_date = df.groupby('Date')['Class'].mean().reset_index()
sns.lineplot(x='Date', y='Class', data=mean_success_rate_by_date)
```

The line chart shows the average launch success rate over the years, with the x-axis representing the year and the y-axis representing the success rate (ranging from 0 to 1, where 1 is a 100% success rate).

Key observations from the chart:

- Early Years (2010-2012):
  - The success rate starts off low
  - This indicates that SpaceX was still in its early stages and facing significant challenges in achieving consistent launch success.
- Steady Improvement (2013-2017):
  - The success rate steadily climbs to nearly 60% by 2017.
  - This demonstrates SpaceX's learning curve and technological advancements during this period.
- Dramatic Increase (2018-2020):
  - The success rate experiences a sharp jump, reaching around 80% by 2018.
  - This substantial improvement suggests that SpaceX had overcome major technical hurdles and refined its launch operations.
- Near-Perfect Success (2020):
  - The success rate reaches a high of around 0.9 (90%) in 2020.
  - This indicates that SpaceX has become highly reliable and consistent in its launch operations.

The overall trend showcases SpaceX's remarkable progress over the years, transforming from a company with relatively low success rates in the early years to one with near-perfect launch success by 2020. This visualization effectively highlights the company's technological maturation and engineering prowess in the space industry.



All Launch Site Names

To find the unique launch site names and to get the number of launches for each site, we can use the `value_counts()` method on the 'LaunchSite' column:

```
df['LaunchSite'].value_counts()
```

This will output the following counts:

```
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
```

Name: count, dtype: int64

The data contains launches from three primary SpaceX launch facilities:

- **CCAFS SLC 40** (*Cape Canaveral Air Force Station Space Launch Complex 40*) - **has the most launches with 55**
- **VAFB SLC 4E** (*Vandenberg Air Force Base Space Launch Complex 4E*) - **has 22 launches**
- **KSC LC 39A** (*Kennedy Space Center Launch Complex 39A*) - **has 13 launches**

As well displaying the list of the unique Launch sites using SQL query

```
[28]  %%sql
      select distinct Launch_Site from SPACEXTBL

 * 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'

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

```
%%sql
select *
from SPACEXTBL
where Launch_Site like "CCA%"
limit 5;
```

Python

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

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

This query selects all columns from the `SPACEXTBL` table where the `Launch_Site` column starts with the string 'CCA' (using the `like` operator with the wildcard '%').

The `limit 5` clause ensures that only the first 5 matching records are returned.

Total Payload Mass

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

```
%%sql
select sum(PAYLOAD_MASS__KG_) as "Sum of payload mass, NASA"
from SPACEXTBL
where Customer = "NASA (CRS)";
```

[38] Python

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

Sum of payload mass, NASA
45596

This query selects the sum of the `PAYLOAD_MASS_KG_` column for all records where the `Customer` is "NASA (CRS)".

The result is assigned the alias "Sum of payload mass, NASA".

The output of this query is: 45596

Average Payload Mass by F9 v1.1

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

```
%%sql
select avg(PAYLOAD_MASS__KG_) as "Average of payload mass, F9 v1.1"
from SPACEXTBL
where Booster_Version like "F9 v1.1";
```

[39]

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

Average of payload mass, F9 v1.1
2928.4

This query selects the average value of the PAYLOAD_MASS_KG_ column for all records where the `Booster_Version` matches the pattern "F9 v1.1".

The result is assigned the alias "Average of payload mass, F9 v1.1".

The output of this query is: 2928.4

First Successful Ground Landing Date

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

```
%%sql
select min(Date) as "First succeseful landing achieved"
from SPACEXTBL;
```

[40]

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

First succeseful landing achieved
2010-06-04

The SQL query selects the minimum date from the SPACEXTBL table, which represents the "First succeseful landing achieved" date.

The query result shows that the first successful landing outcome on a ground pad was achieved on 2010-06-04.

Successful Drone Ship Landing with Payload between 4000 and 6000

[45]

```
%%sql
select distinct Booster_Version
from SPACEXTBL
where Mission_Outcome = "Success" and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

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

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 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

The query result shows the distinct Booster_Version values from the SPACEXTBL table where the Mission_Outcome is "Success" and the PAYLOAD_MASS_KG is between 4000 and 6000.

This query provides the list of booster versions that have successfully landed on a drone ship and had a payload mass in the range of 4000 to 6000 kg.

This information can be useful to highlight the capabilities and achievements of the space program in terms of successful landings and payload delivery within a specific mass range.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
select distinct Mission_Outcome, count(*) as Count
from SPACEXTBL
group by Mission_Outcome
order by 2 desc;
```

[49]

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

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

The query result shows the distinct Mission_Outcome values from the SPACEXTBL table, along with the count of each outcome.

The key results are:

- Success: 98
- Success (payload status unclear): 1
- Success: 1
- Failure (in flight): 1

This information can be used to give an overview of the overall mission success rate and highlight the breakdown between successful and failed missions in the space program's operations.

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql
select distinct Booster_Version
from SPACEXTBL
where PAYLOAD_MASS__KG_ = (
    select max(PAYLOAD_MASS__KG_)
    from SPACEXTBL
);
```

[54]

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

The query result shows the distinct Booster_Version values from the SPACEXTBL table, where the PAYLOAD_MASS_KG is the maximum value.

The booster versions that have carried the maximum payload mass are:

- 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

This information can be used to highlight the boosters that have demonstrated the highest payload carrying capabilities in the space program's operations.

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

```
%%sql
select
    case strftime('%m', Date)
        when '01' then 'January'
        when '02' then 'February'
        when '03' then 'March'
        when '04' then 'April'
        when '05' then 'May'
        when '06' then 'June'
        when '07' then 'July'
        when '08' then 'August'
        when '09' then 'September'
        when '10' then 'October'
        when '11' then 'November'
        when '12' then 'December'
    end as Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
from SPACEXTBL
where Landing_Outcome LIKE 'Failure%'
  and strftime('%Y', Date) = '2015';
```

[26] ✓ 0.0s

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

Month	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

The SQL query retrieves the records from the SPACEXTBL table that meet the following criteria:

- The year is 2015
- The mission outcome indicates a failure

The result represents two cases that occurred in January and in April months.

The SQLite does not have a direct way to convert month numbers into month names, so I used the "case" statement.

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.

```
%%sql
select Landing_Outcome, count(*) as Count
from SPACEXTBL
where Date between "2010-06-04" and "2017-03-20"
group by Landing_Outcome
order by Count desc;
```

[19] ✓ 0.0s

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

Landing_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

The SQL query retrieves the count of each distinct landing outcome from the SPACEXTBL table for the date range between 2010-06-04 and 2017-03-20.

The results are ordered in descending order by the count.

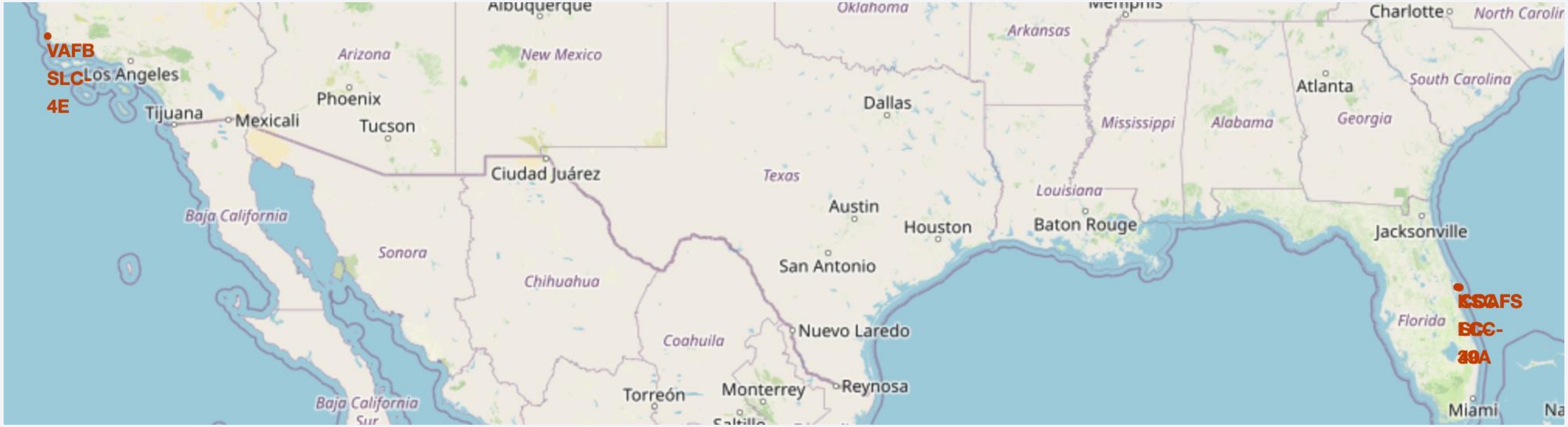
The key insights from the query output are:

- The most common landing outcome is "No attempt" with 10 occurrences.
- There are 5 occurrences each of "Success (drone ship)" and "Failure (drone ship)".
- There are 3 occurrences each of "Success (ground pad)", "Controlled (ocean)", and "Uncontrolled (ocean)".
- There are 2 occurrences each of "Failure (parachute)" and "Precluded (drone ship)".

Section 3

Launch Sites Proximities Analysis

Launch Site Locations



This map visualization shows the locations of different launch sites around the world.

The key elements are:

- The map covers a global view, allowing us to see the distribution of launch sites across different regions.
- Each launch site is represented by a colored circle, indicating its precise geographic coordinates.
- The size of the circles is set to a radius of 1000 meters, providing a visual representation of the approximate coverage area for each site.
- The launch site names are displayed as text labels on the map, making it easy to identify the specific locations.

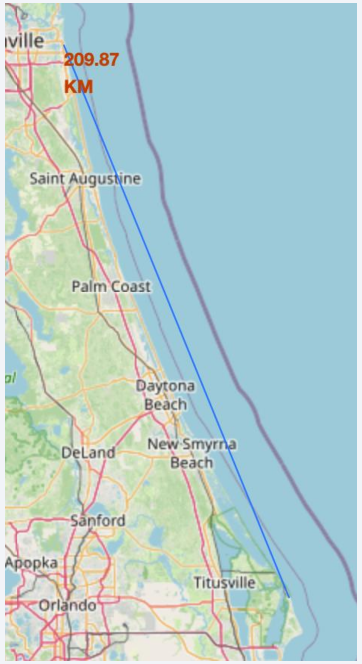
SpaceX Launch Sites and Outcomes

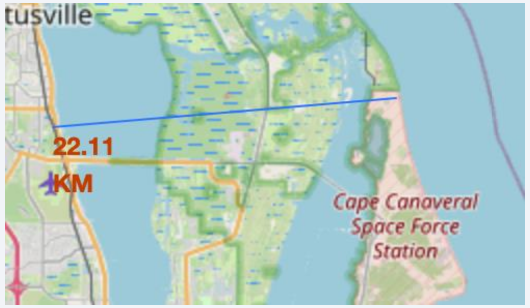
The map highlights the SpaceX launch sites with color-coded markers representing the outcomes of each launch.

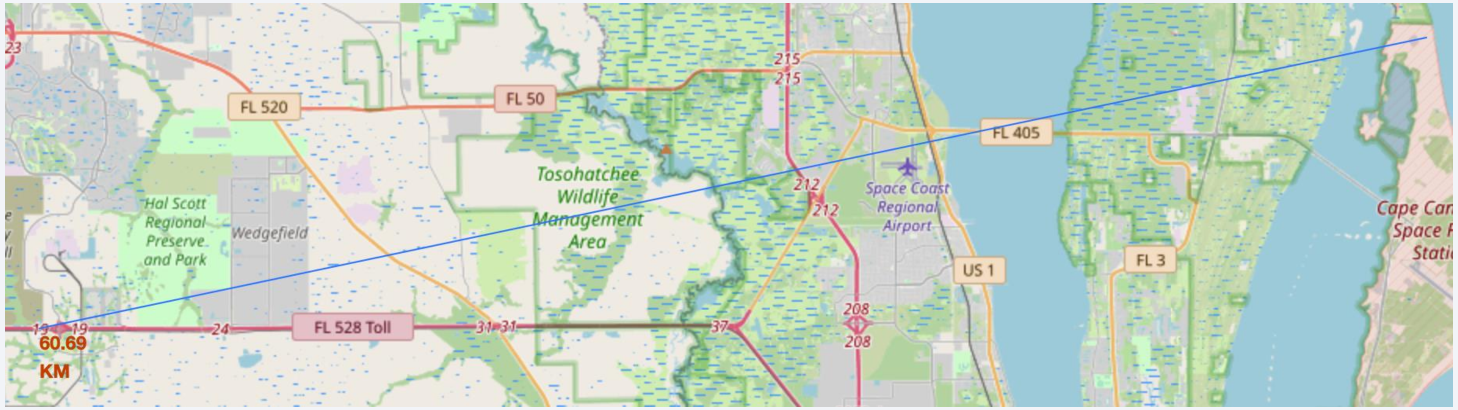
- **Green Markers**: Successful launches (`class=1`).
- **Red Markers**: Failed launches (`class=0`).

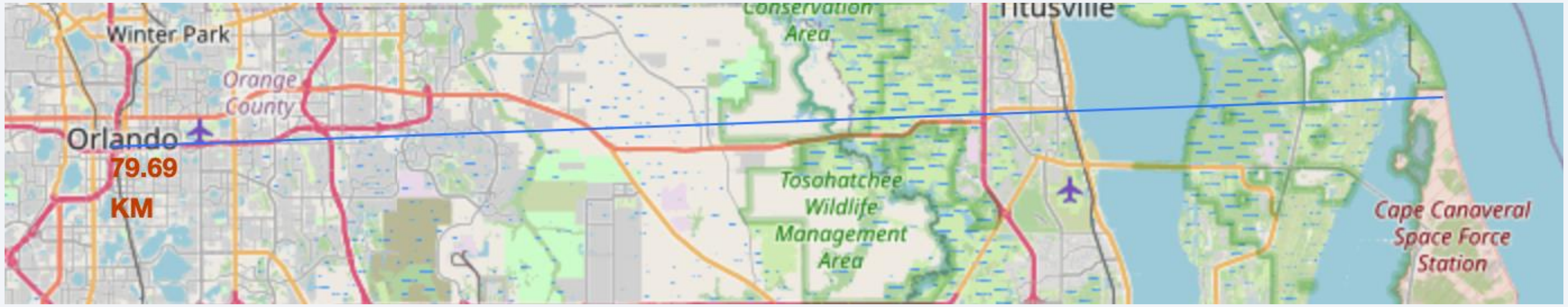
Using `MarkerCluster` simplifies the map, enabling clear visualization of launch outcomes even when multiple launches occurred from the same location.

Proximity Analysis of SpaceX Launch Sites









This map shows the proximity of a selected SpaceX launch site to nearby key infrastructures such as coastlines(209.87km), railways(22.11km), highways(60.69km) and cities.

Calculated distances (in kilometers) from the launch site to each nearby point of interest are displayed.

Each line connects the launch site to a specific nearby feature, making it easy to visualize distances and directions.

Section 4

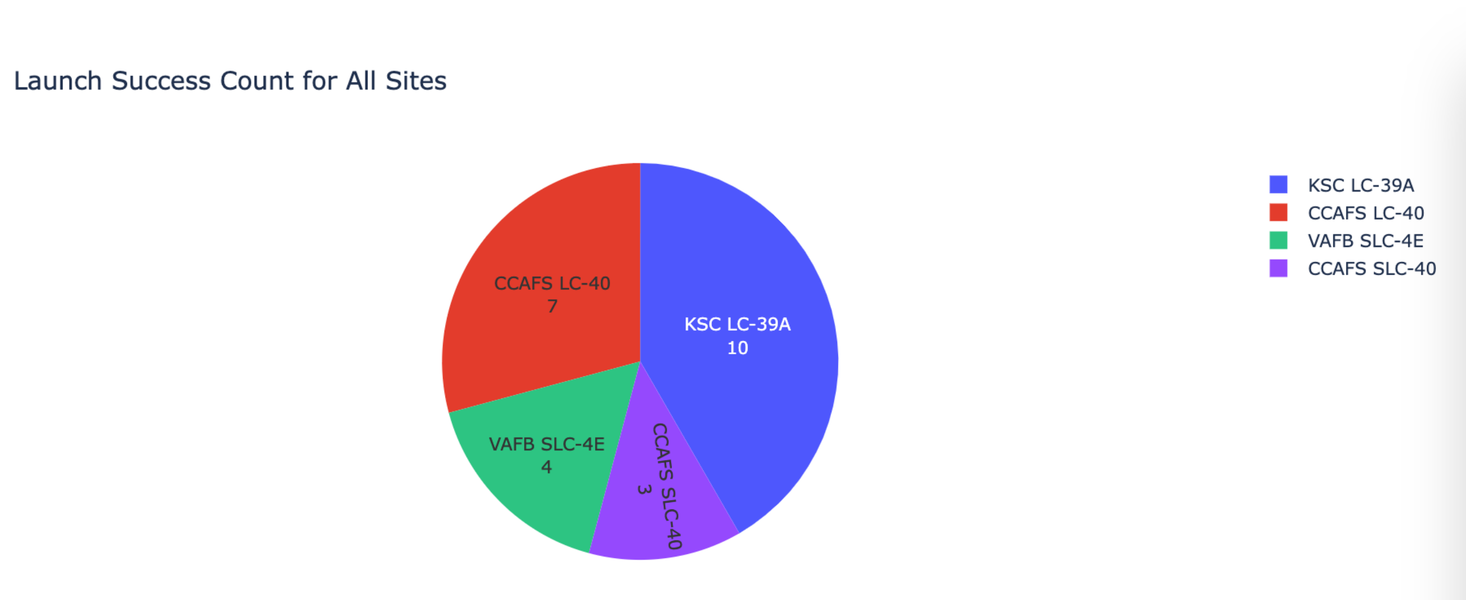
Build a Dashboard with Plotly Dash

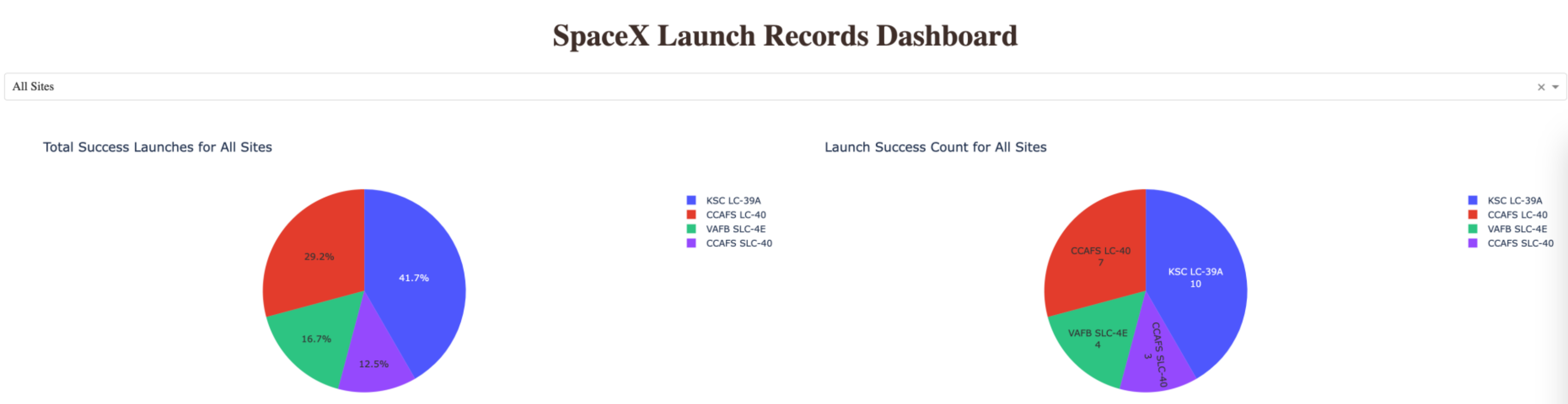
SpaceX Launch Success Count by Launch Sites

The pie chart visualizes the successful launch distribution across SpaceX's four main launch facilities.

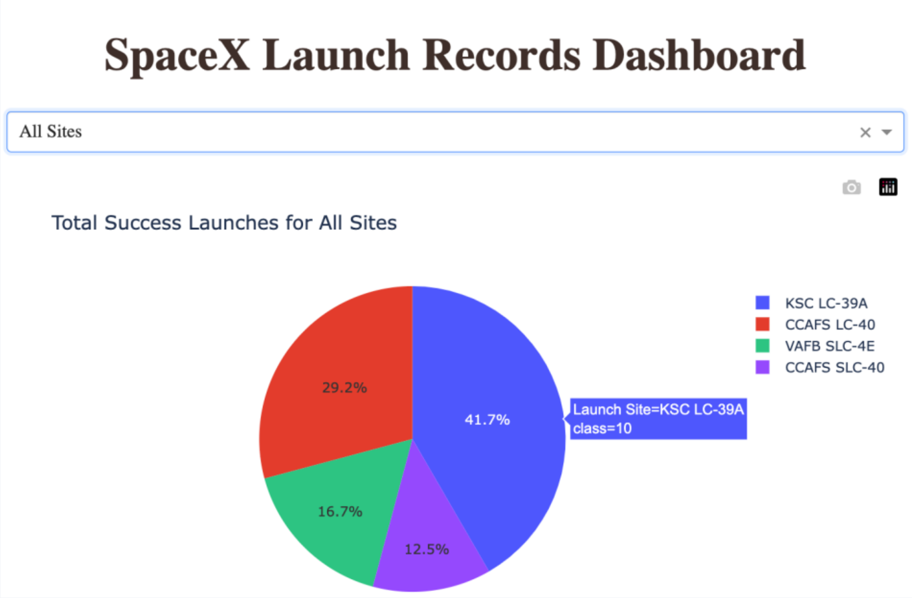
Key findings:

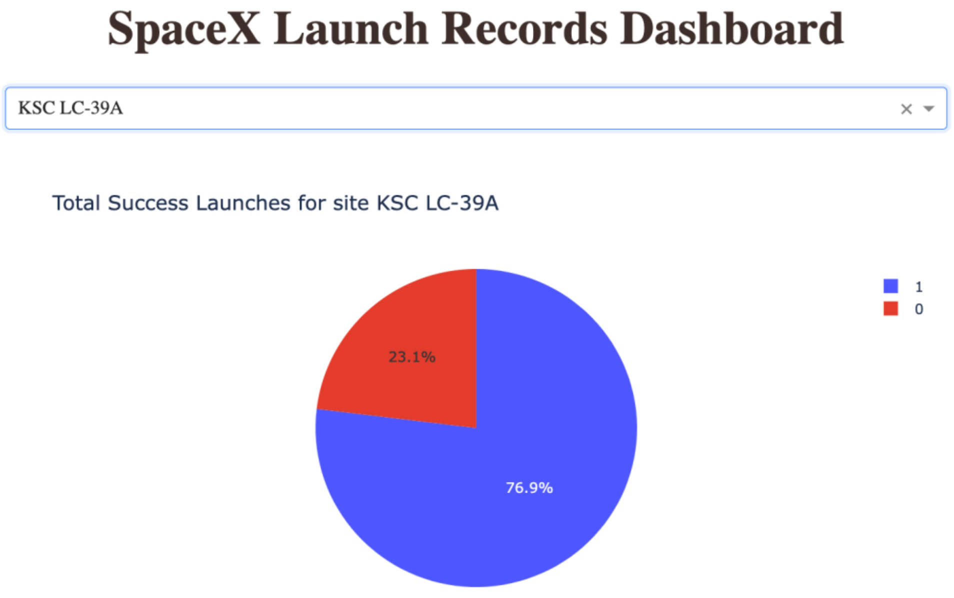
- KSC LC-39A (Kennedy Space Center) leads with 10 successful launches, representing the highest success rate
- CCAFS LC-40 (Cape Canaveral Air Force Station) follows with 7 successful launches
- VAFB SLC-4E (Vandenberg Air Force Base) recorded 4 successful launches
- CCAFS SLC-40 had 3 successful launches





SpaceX Launch Success by Site

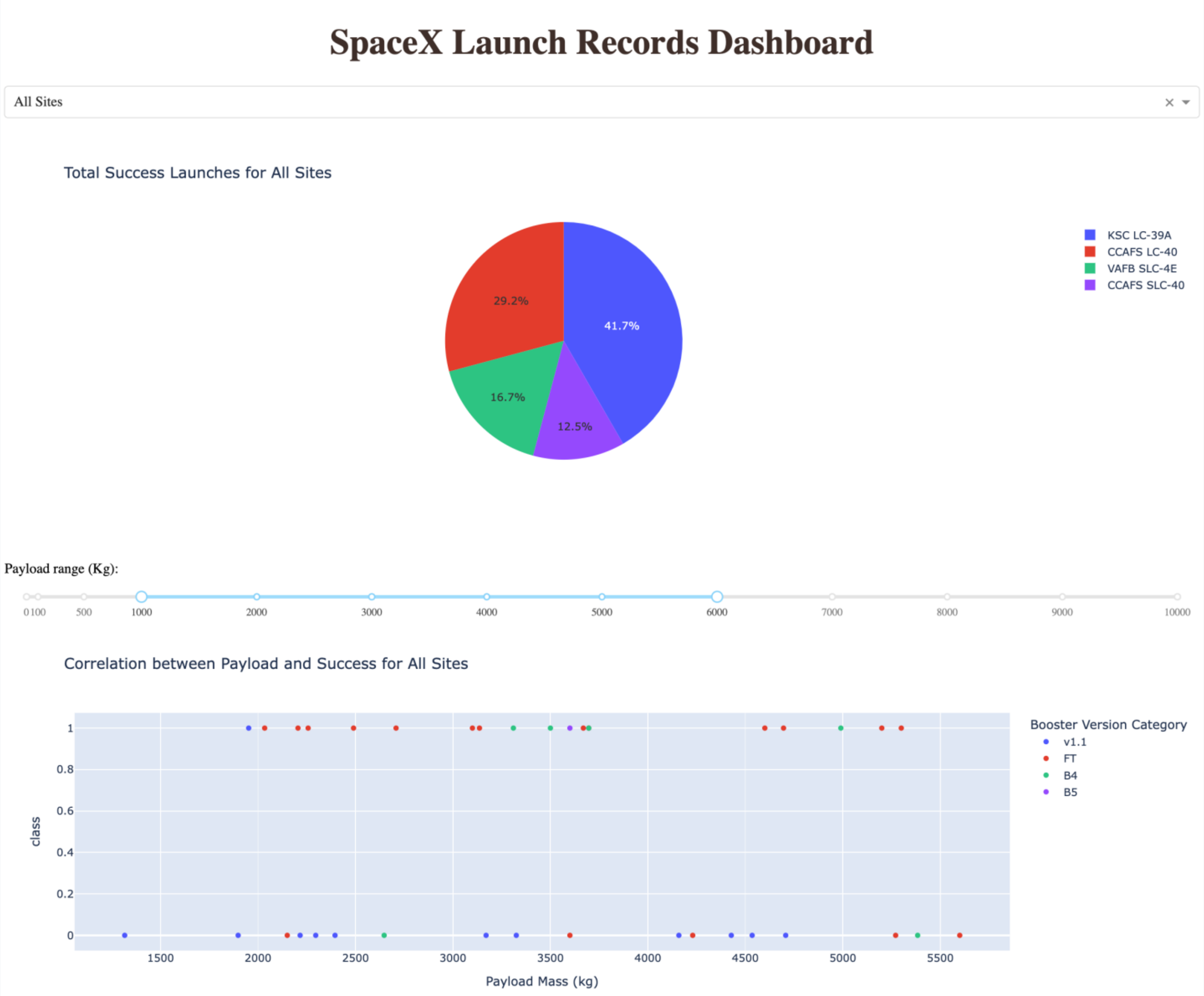




Key Findings:

- KSC LC-39A consistently achieved over 70% success, significantly higher than other sites.
- This site's success rate is significantly higher than the overall average success rate across all SpaceX launch sites.
- This highlights the reliability and efficiency of operations at KSC LC-39A.

SpaceX Launch Success by Payload Range and Booster Version



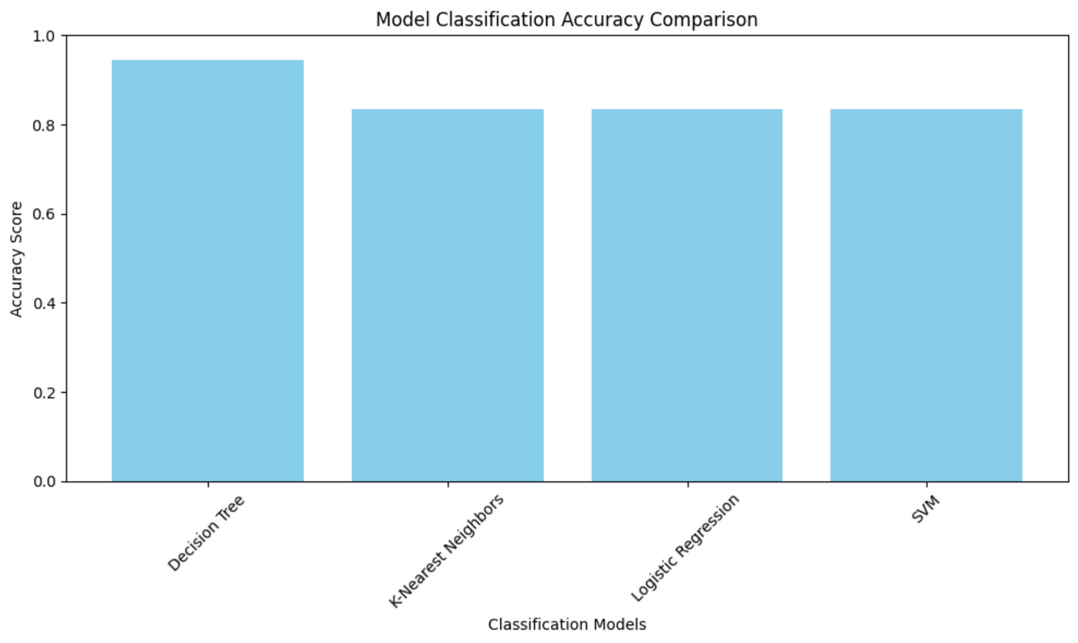
- We've set the payload rage to 1000 – 6000 kg. Also selected all sites from the drop down.
- **Payload Range:** The scatter plot reveals a trend where higher payload masses tend to have a higher success rate. This is likely due to SpaceX's continuous improvements in rocket technology and payload integration.
- **Booster Version:** The color-coded points indicate different booster versions. By observing the distribution of colors within the scatter plot, you can identify which booster versions have the highest success rates across various payload ranges.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The bar chart compares the accuracy (calculated using the method **score**) of four classification models: Decision Tree, K-Nearest Neighbors, Logistic Regression, and SVM(Support Vector Machine).
- The y-axis shows the accuracy scores, ranging from 0 to 1, while each model is represented along the x-axis.
- **Best Model**: Decision Tree has the highest accuracy score, suggesting it may be the best option among these models for this task.



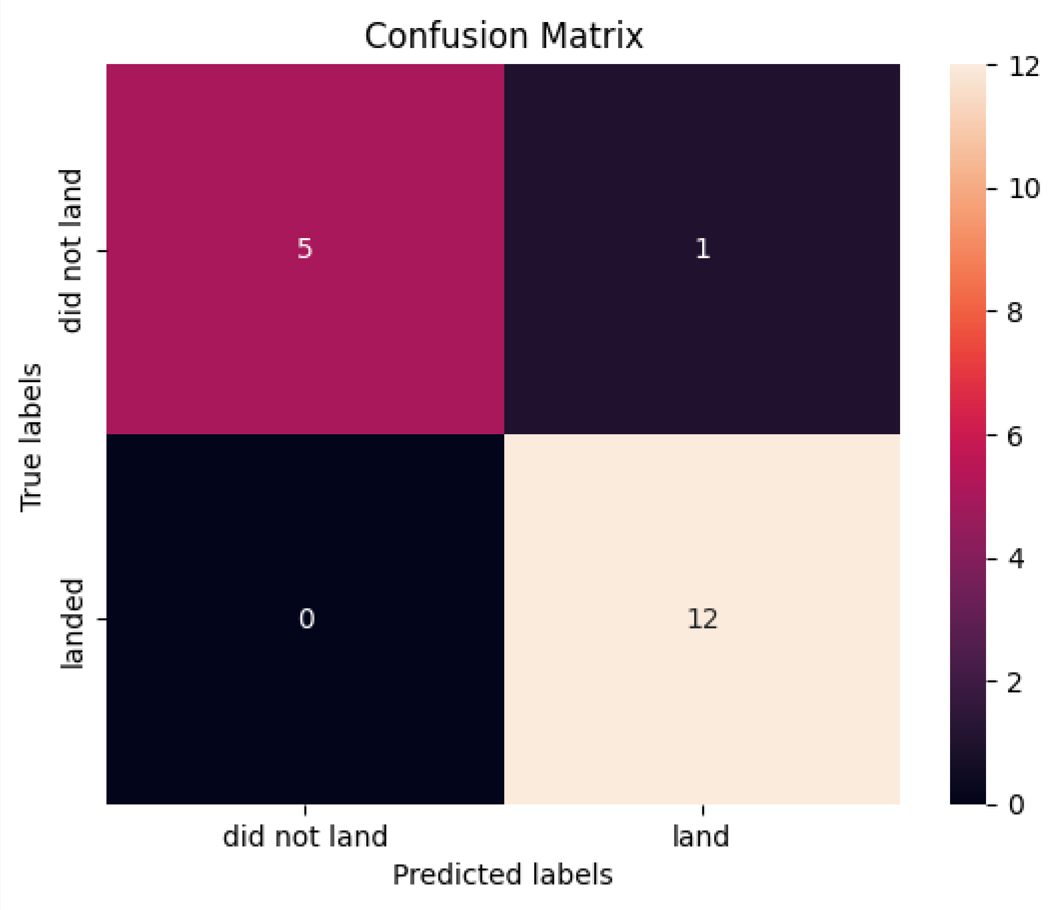
Confusion Matrix for Decision Tree Model

Confusion Matrix Description:

- The confusion matrix for the Decision Tree model illustrates the model's performance in classifying landings.
- **True Positives (Bottom Right)**: The model correctly predicted 12 landings.
- **True Negatives (Top Left)**: The model correctly identified 5 cases where the rocket did not land.
- **False Positives (Top Right)**: There was 1 instance where the model incorrectly predicted a landing.
- **False Negatives (Bottom Left)**: The model did not have any false negatives, meaning it captured all actual landings without missing any.

Interpretation:

- The model performed well overall, with only one misclassification.
- The absence of false negatives indicates strong predictive reliability in identifying successful landings, which is crucial for accurate forecasting in this context.
- The single false positive suggests minor room for improvement, but overall, the Decision Tree model is highly effective in distinguishing between landing outcomes.



Conclusions

1. **Decision Tree Model Effectiveness**:

   The Decision Tree model proved the most effective, with only one misclassification, offering strong sensitivity in identifying successful landings.

2. **Alternative Models**:

   While Logistic Regression and SVM showed competitive accuracy, they did not outperform the Decision Tree model. These models could be considered viable alternatives but may require parameter tuning or additional features for improved performance.

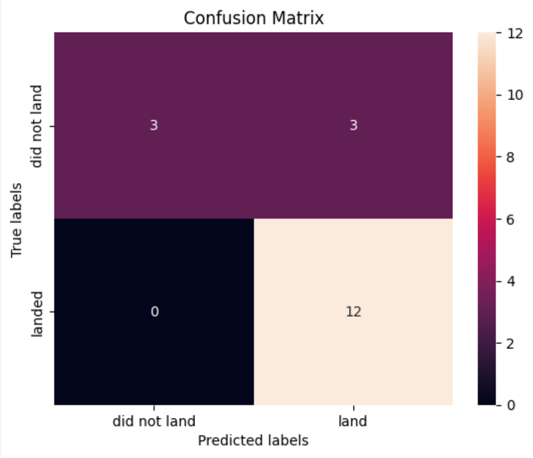
3. **Insights from the Confusion Matrix**:

   The confusion matrix revealed that the Decision Tree model effectively minimized false negatives, capturing all actual landings without missing any. This high sensitivity is valuable for applications where missed predictions of successful landings could lead to costly oversights.

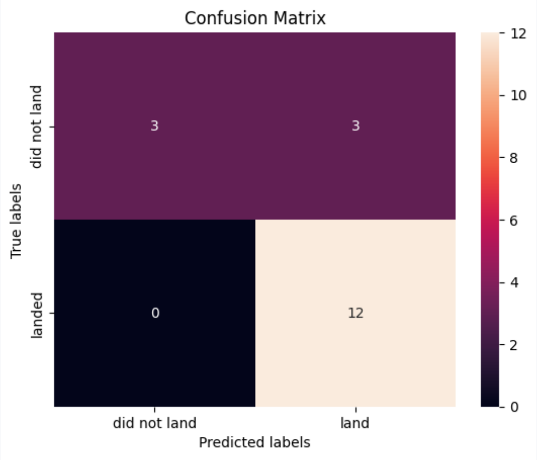
4. **Future Improvements**:

   Additional data or feature engineering could further enhance model performance. For example, incorporating more specific features about the rocket’s specifications, weather conditions, or launch parameters may increase classification accuracy across all models.

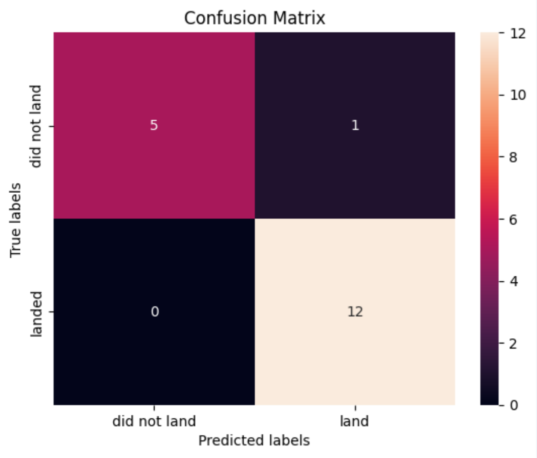
Appendix



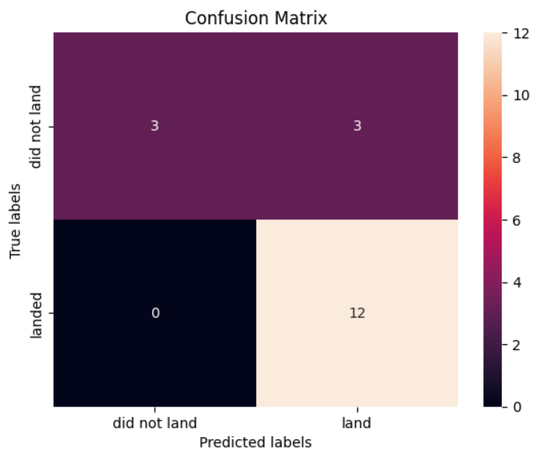
Logistic regression confusion matrix



SVM confusion matrix



Decision tree confusion matrix



K nearest neighbors confusion matrix

Here we have the confusion matrices for each classification model used in predicting rocket landing success.

Summary: All models performed well in identifying successful landings, but the Decision Tree model outperformed others with the fewest misclassifications, making it the most robust choice for this analysis. This comparison highlights each model's ability to distinguish between successful and unsuccessful landings.

Thank you!