IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pravinkumar Gahlod
24-09-2025

Outline

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

Executive Summary

- Summary of methodologies

Data Collection & Preparation

- Loaded the SpaceX dataset from CSV.
- Converted `Date` to datetime format and extracted `Year`.
- Checked for missing or inconsistent data and cleaned as needed.
- **Exploratory Data Analysis (EDA)**
- **Categorical Analysis:** Used `catplot` to visualize relationships between `FlightNumber`, `LaunchSite`, `Orbit`, `PayloadMass`, and `Class`.
- **Success Rate Analysis:**
    - Grouped by `Orbit`, `LaunchSite`, and `Year` to calculate mean `Class` values (success rates).
    - Visualized success trends with bar charts and line plots.
    - **Visualization Techniques**
- **Bar Charts:** Success rate by orbit and launch site.
- **Scatter/Strip Plots:** FlightNumber vs LaunchSite, PayloadMass vs LaunchSite, FlightNumber vs Orbit, PayloadMass vs Orbit.
- **Line Plot:** Yearly success rate trend.

- Summary of all results

Executive Summary

- Summary of all results
- **Overall Success Trends**
- SpaceX launch success has generally increased over the years.
- Early launches had higher failure rates, but recent launches show near-perfect success.
- **Launch Site Analysis**
- Certain launch sites consistently show higher success rates.
- Variability exists depending on the site and flight history.
- **Orbit Analysis**
- Success rates vary across orbit types.
- Some orbits have consistently higher success due to simpler mission requirements.
- **Payload Mass and Booster Analysis**
- Payload mass does not strictly determine success but extremely heavy or light payloads may influence outcomes.
- Reused boosters show comparable success rates to new boosters, indicating reliability in reuse strategies.
- **Flight Number Trends**
- Higher flight numbers (more experienced boosters) tend to show higher success, indicating learning and optimization over time.

Introduction

- ## Project background and context

SpaceX, a leading private aerospace company, conducts regular rocket launches for satellite deployment, cargo delivery, and space exploration. Each launch involves various parameters such as payload mass, orbit type, launch site, booster version, and flight outcomes. Analyzing historical launch data can provide insights into factors influencing launch success, improve future mission planning, and support decision-making for payload assignments and launch strategies.

The dataset contains information about **flight numbers, dates, booster versions, payloads, launch sites, orbits, and launch outcomes** (success/failure), along with additional technical details like booster reuse and landing outcomes.

- ## Problems you want to find answers

The goal of this analysis is to answer key questions such as:

- **Launch Success Analysis**
    - What is the overall success rate of SpaceX launches?
    - How does launch success vary over time (yearly trends)?
- **Launch Site and Orbit Performance**
    - Which launch sites have the highest success rates?
    - How do different orbit types affect launch success?
- **Payload and Booster Insights**
    - Does payload mass influence launch success?
    - Are certain booster versions or reused boosters associated with higher success rates?
- **Flight Number Trends**
    - Are there patterns in success related to flight numbers (experience of boosters over time)?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

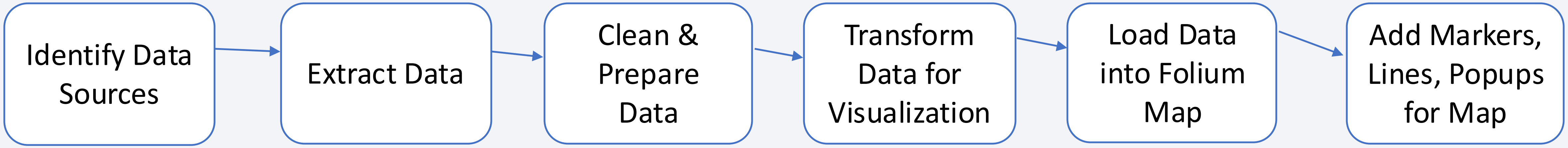
- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection Process (Key Phrases)

- **Identify Data Sources**
  - SpaceX launch data → official SpaceX API / Kaggle dataset
  - Geographic coordinates → launch site latitudes and longitudes
  - Proximity data → cities, highways, railways from OpenStreetMap / Google Maps API
- **Extract Data**
  - Download CSV/JSON datasets for launches and payloads
  - Query APIs for geolocation and proximity data
- **Clean and Prepare Data**
  - Remove duplicates and null values
  - Standardize column names (`Lat`, `Long`, `Launch Site`)
  - Create calculated columns (e.g., `marker_color` based on `class`)

Data Collection

- **Transform Data for Visualization**
  - Compute distances from launch sites to nearby cities/railways/highways
  - Categorize outcomes (Success/Failure)
  - Merge datasets to form a unified dataframe for mapping
- **Load Data into Mapping Tool**
  - Feed prepared data into Folium for map visualization
  - Add markers, lines, and popups

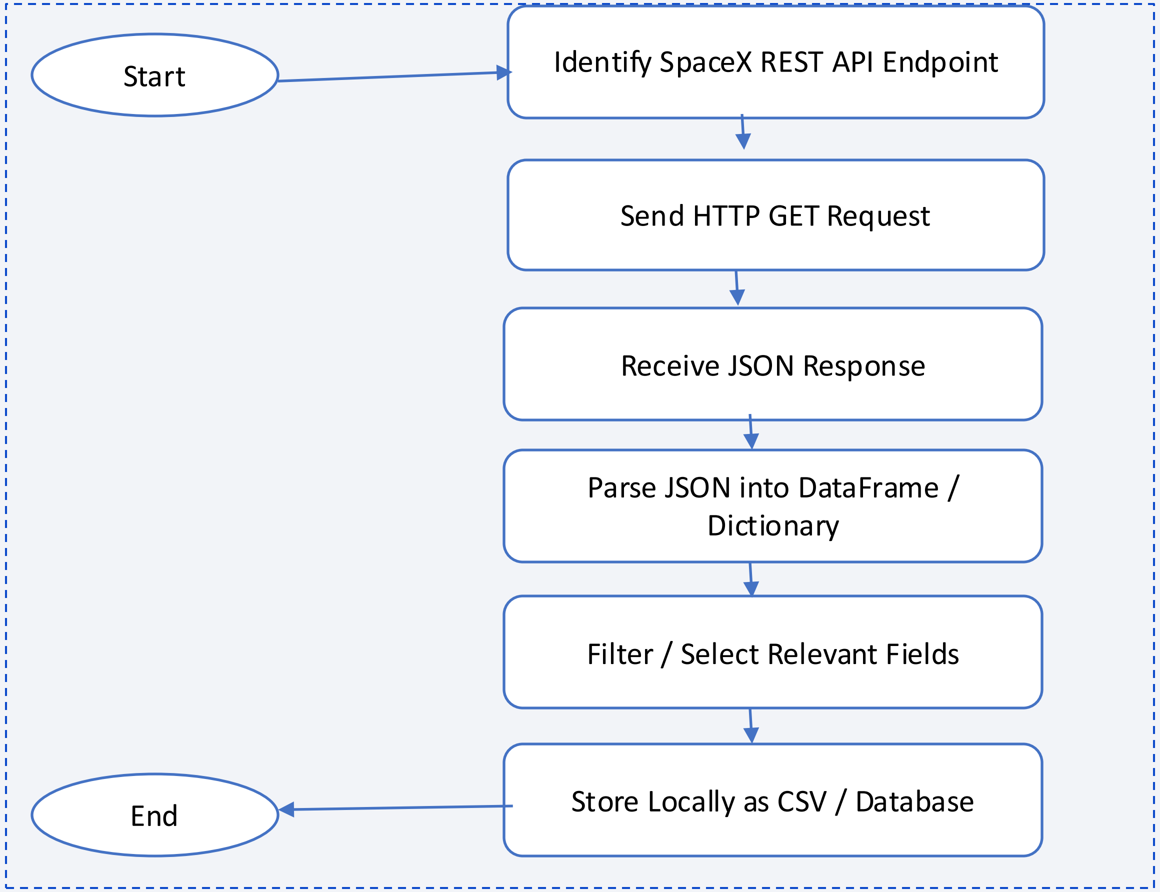


Data Collection – SpaceX API

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

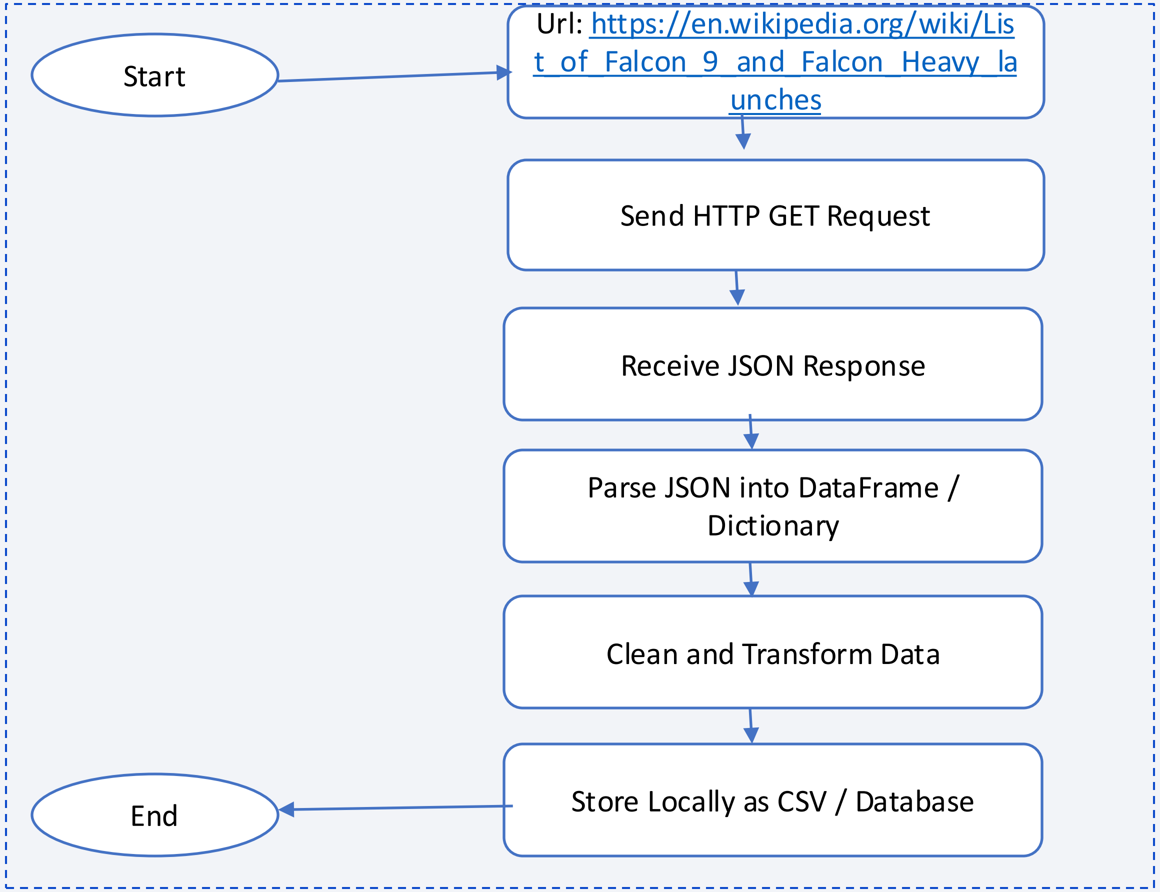
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

- https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/SpaceXAPI.ipynb



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose
- https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/jupyter-labs-webscraping.ipynb



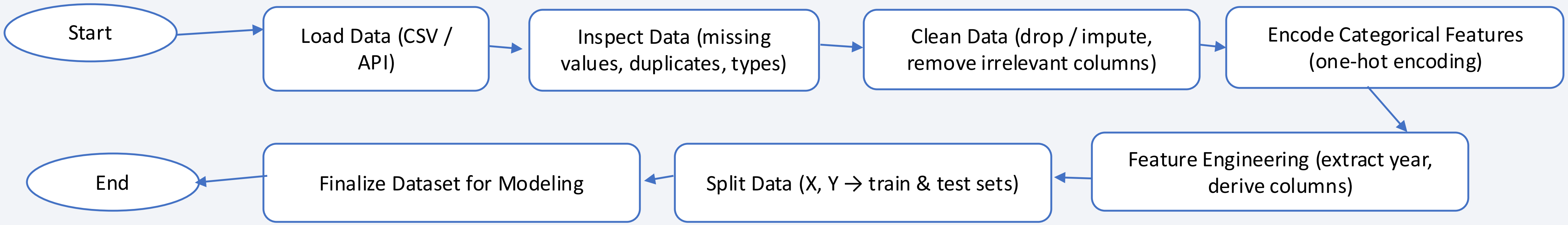
Data Wrangling

- Describe how data were processed

- **Load Data:** Imported CSV or API JSON into pandas DataFrame.
- **Inspect Data:** Checked for missing values, duplicates, and column types.
- **Handle Missing/Invalid Values:** Dropped or imputed missing values; removed irrelevant columns.
- **Encode Categorical Features:** Converted categorical columns to numeric using one-hot encoding.
- **Feature Extraction:** Derived new features like year from `Date`.
- **Split Features and Target:** Defined X (features) and Y (`Class`).
- **Train-Test Split:** Split dataset into training and testing sets (e.g., 80-20).
- **Final Dataset:** Cleaned, numeric, and ready for modeling.

Data Wrangling

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



- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- **FlightNumber vs LaunchSite (hue=Class):** Show success patterns across launch sites.
- **PayloadMass vs LaunchSite (hue=Class):** Assess effect of payload on success at each site.
- **Bar Chart: Success Rate by Orbit:** Compare success rates across orbit types.
- **FlightNumber vs Orbit / PayloadMass vs Orbit (hue=Class):** Identify success patterns for different orbits.
- **Line Chart: Success Rate by Year:** Track success trends over time.
- **Confusion Matrix:** Evaluate model predictions (true/false positives & negatives).
- **Bar Chart: Model Accuracy Comparison:** Identify the best-performing classification model.
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/EDA%20with%20Data%20Visualization%E2%80%8B.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

- https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite_EDAwithSQL.ipynb
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes
- List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
- 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.
- 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.

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

  https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

Map objects added:

- **Markers** – for launch sites (blue) and proximities like cities, highways, railways (orange)
- **Lines (PolyLine)** – connecting each proximity to its nearest launch site
- **Popups** – showing details such as site name, proximity type, and distance

Reason for adding:

- **Markers** highlight locations on the map for easy identification.
- **Lines** visualize the spatial relationship and distance between launch sites and nearby points.
- **Popups** provide contextual information without cluttering the map.

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

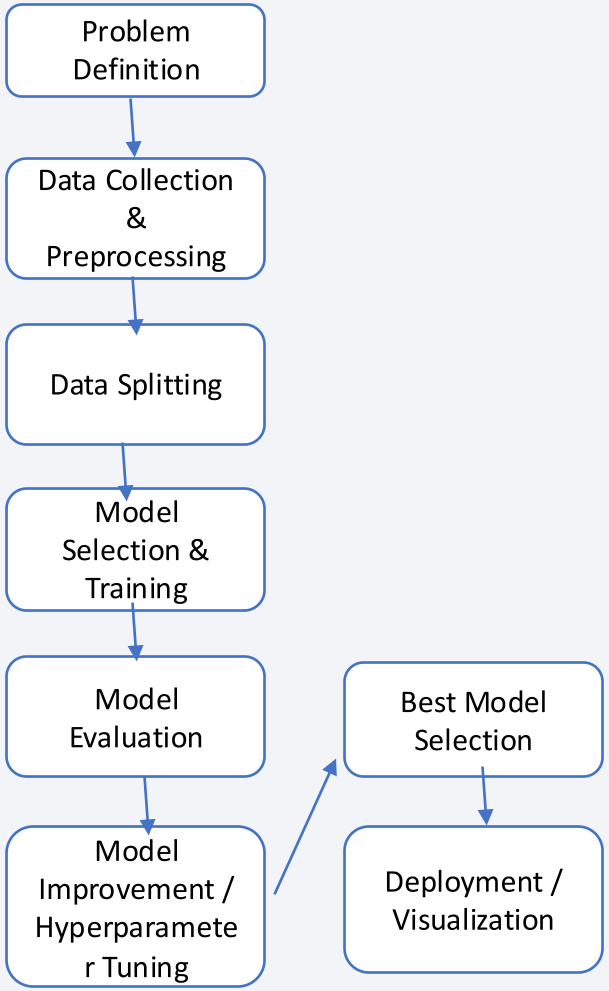
- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

https://github.com/prgahlod/CourseraCapstoneSpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

- Model Development Summary (Key Phrases)
- Problem Definition
  - Predict launch success (class = 0 or 1) based on payload, booster version, and other features.
- Data Collection & Preprocessing
  - Collect SpaceX launch data from CSV/API
  - Handle missing values and duplicates
  - Encode categorical variables (e.g., Booster Version, Launch Site)
  - Scale numeric features (e.g., Payload Mass)
- Data Splitting
  - Split data into training set and test set (e.g., 80% train, 20% test)
  - Optional: further split train set into validation set for hyperparameter tuning

Predictive Analysis (Classification)

- **Model Selection & Training**
  - Evaluate multiple classification algorithms (Logistic Regression, Random Forest, XGBoost, SVM)
  - Train models on the training data
- **Model Evaluation**
  - Metrics: Accuracy, Precision, Recall, F1-score, ROC-AUC
  - Confusion matrix to analyze misclassifications
  - Cross-validation to assess robustness
- **Model Improvement**
  - Feature engineering (add relevant features, remove irrelevant ones)
  - Hyperparameter tuning (Grid Search, Random Search)
  - Ensemble methods or boosting for better performance
- **Best Model Selection**
  - Select model with highest validation/test performance
  - Check for overfitting by comparing train vs test metrics
- **Deployment/Visualization**
  - Use model for predictions (e.g., success probability)
  - Visualize results using dashboards or charts (e.g., scatter plot, pie chart)



Results

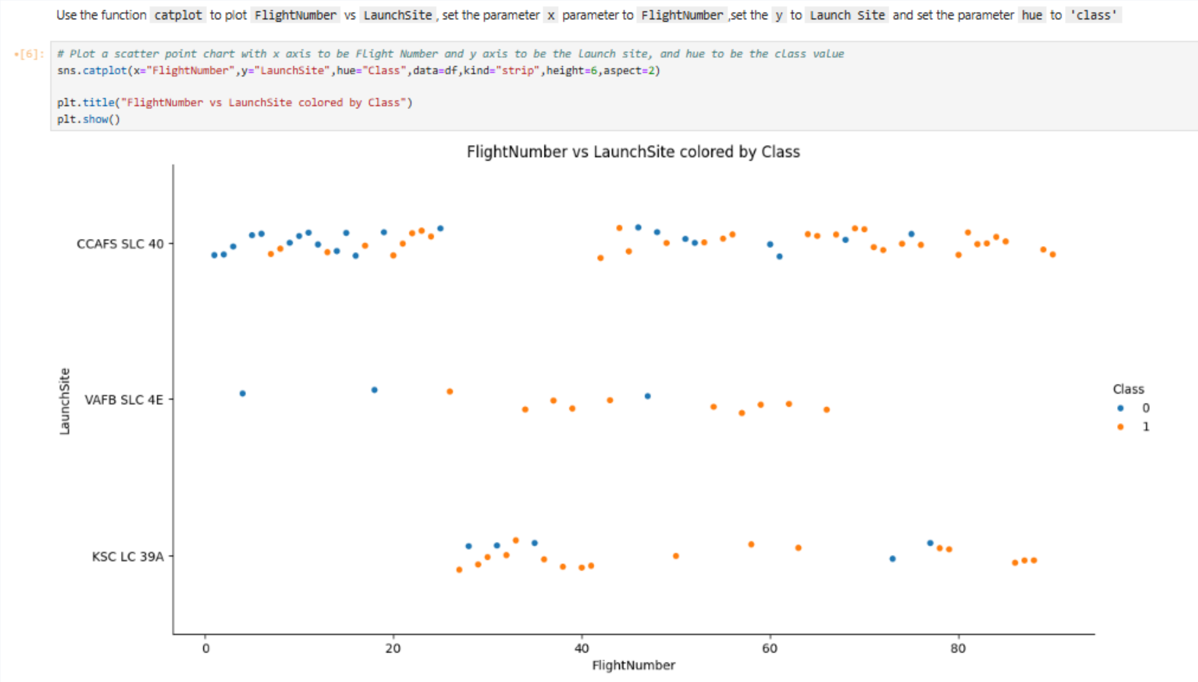
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations



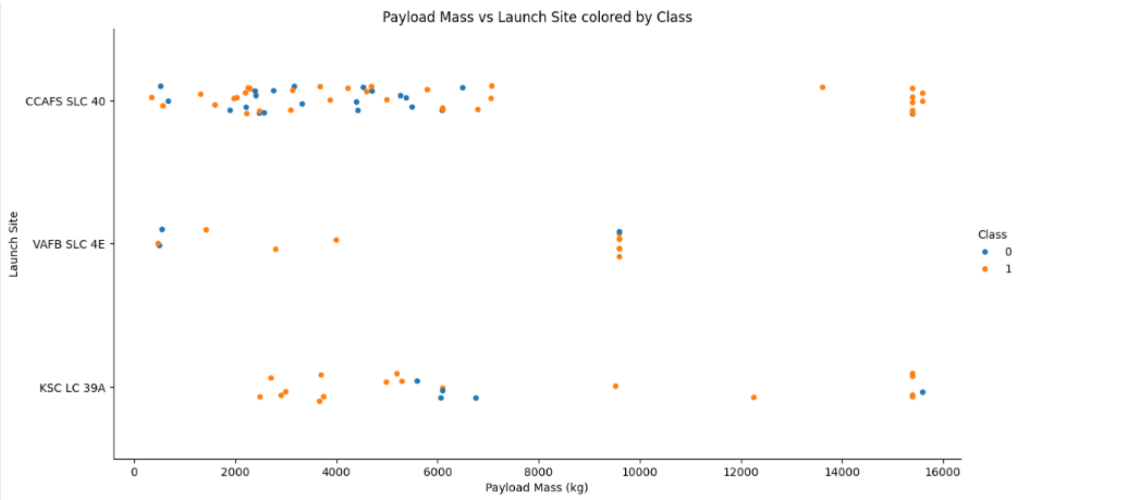
- **Seaborn's `catplot`** to visualize the relationship between `FlightNumber` (x-axis) and `LaunchSite` (y-axis), coloring each point by `Class` (hue). Each point represents a flight, and the hue shows whether it was successful or failed. Using `kind="strip"` spreads points along the y-axis for clarity

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations

We also want to observe if there is any relationship between launch sites and their payload mass.

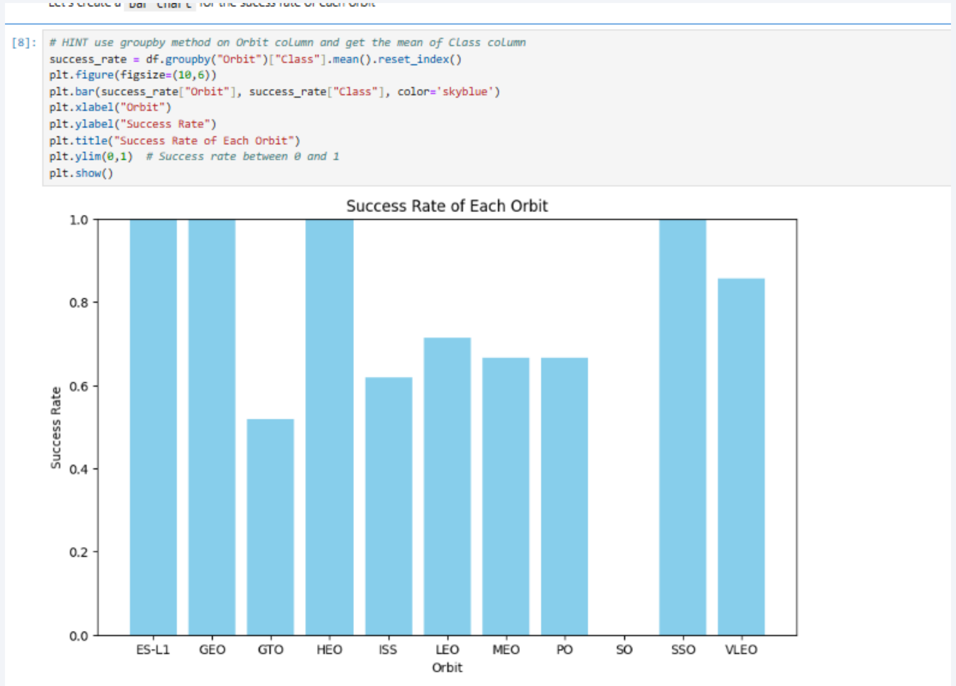
[7]:
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x="PayloadMass",y="LaunchSite",hue="Class",data=df,kind="strip",height=6,aspect=2)
plt.title("Payload Mass vs Launch Site colored by Class")
plt.xlabel("Payload Mass (kg)")
plt.ylabel("Launch Site")
plt.show()
```



- This plot shows **Payload Mass (x-axis) vs Launch Site (y-axis)**, with points colored by `Class`. Each point represents a flight, and the hue indicates success (1) or failure (0)

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations



- This bar chart shows the **success rate for each Orbit**, calculated as the mean of the `Class` column. Higher bars indicate orbits with more successful launches.

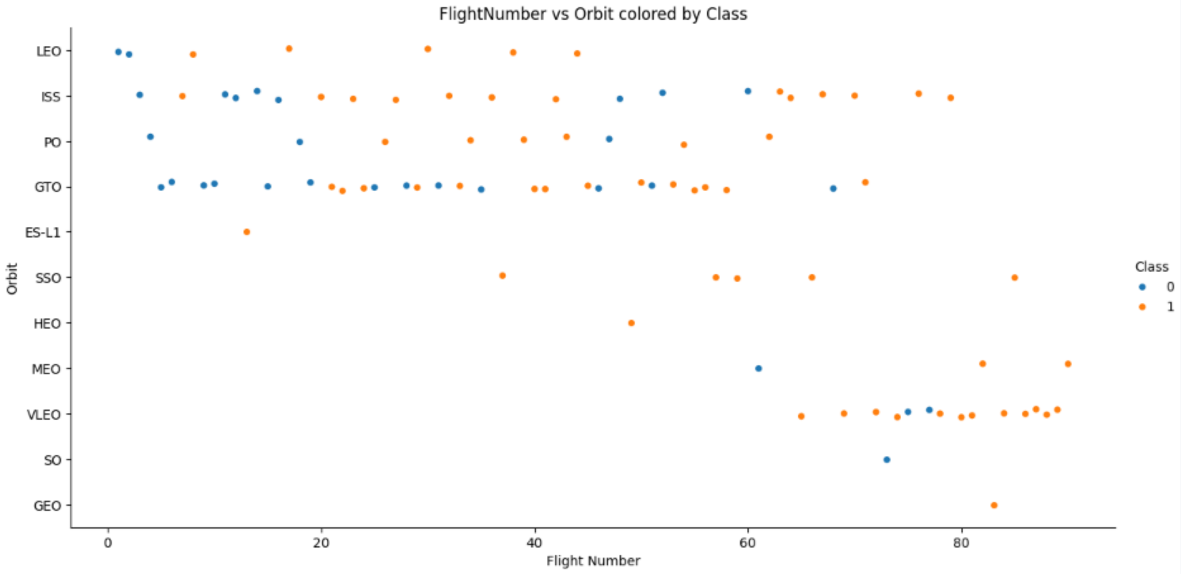
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

```
[9]: # Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="FlightNumber",y="Orbit",hue="Class",data=df,kind="strip",height=6,aspect=2)

plt.title("FlightNumber vs Orbit colored by Class")
plt.xlabel("Flight Number")
plt.ylabel("Orbit")
plt.show()
```



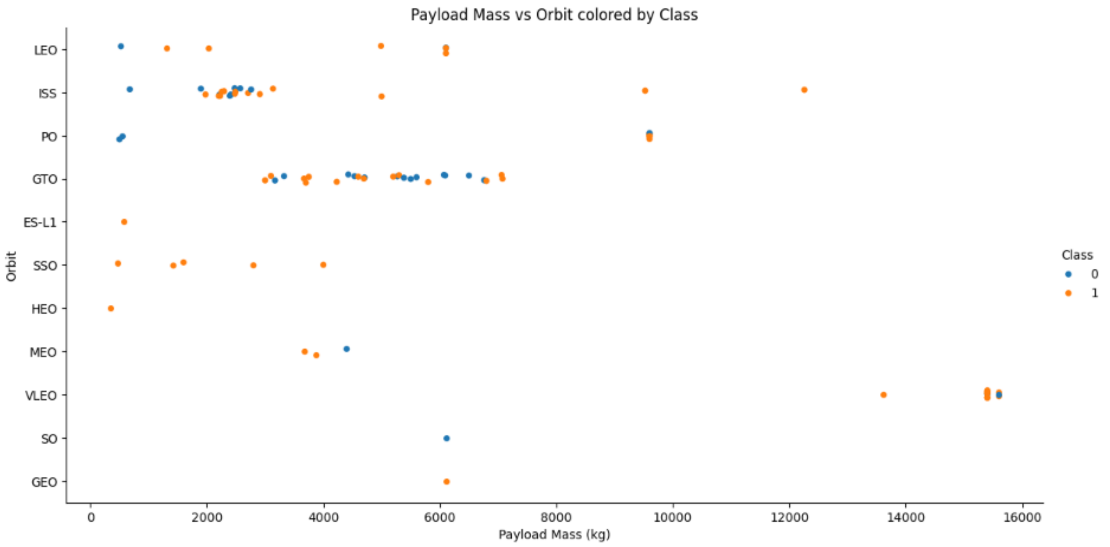
- This plot shows **FlightNumber vs Orbit**, with points colored by `Class` to indicate success (1) or failure (0).

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

Similarly, we can plot the Payload Mass vs. Orbit scatter point charts to reveal the relationship between Payload Mass and Orbit type

[10]:
```python
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="PayloadMass",y="Orbit",hue="Class",data=df,kind="strip",height=6,aspect=2)
plt.title("Payload Mass vs Orbit colored by Class")
plt.xlabel("Payload Mass (kg)")
plt.ylabel("Orbit")
plt.show()
```



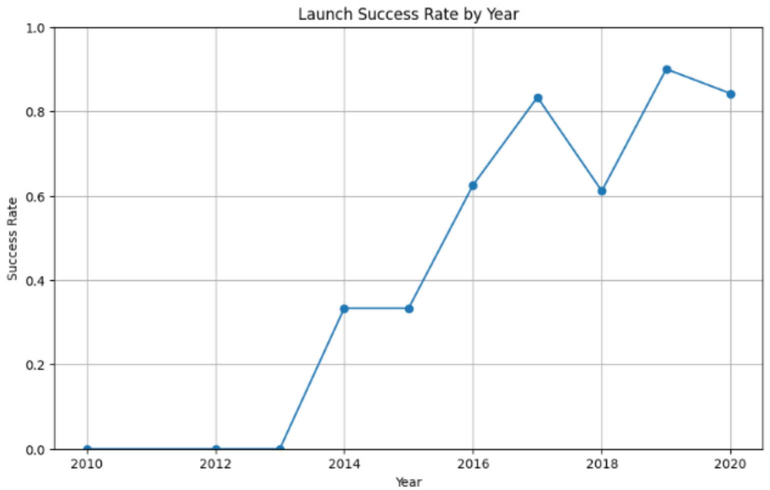
- This plot shows **Payload Mass vs Orbit**, with points colored by `Class` to indicate success (1) or failure (0)

Launch Success Yearly Trend

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

[12]:
```python
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
success_rate_year = df.groupby('Year')['Class'].mean().reset_index()

plt.figure(figsize=(10,6))
plt.plot(success_rate_year['Year'], success_rate_year['Class'], marker='o', linestyle='-')
plt.xlabel("Year")
plt.ylabel("Success Rate")
plt.title("Launch Success Rate by Year")
plt.ylim(0,1)
plt.grid(True)
plt.show()
```



- This line chart shows the **yearly launch success rate**, with each point representing the average `Class` value for that year, highlighting trends in launch success over time.

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

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

```
[21]: %sql select Distinct Launch_site from SPACEXTBL
```

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

[21]:

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

- Using Distinct we can query unique names

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

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

```
[23]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_site LIKE 'CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

[23]:

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

- Using limit only required rows display

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

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

```
[26]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

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

[26]:

TotalPayloadMass
45596

- Sum function can used to calculate the total of column values

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

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

```
[27]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AvgPayloadMass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

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

[27]:

AvgPayloadMass
2928.4

- Using Average function Payload mass average is calculated

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

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

Hint:Use min function

```
[30]: %sql SELECT Date FROM SPACEXTABLE WHERE Landing_Outcome = 'Success' ORDER BY Date ASC LIMIT 1;
```

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

[30]:

Date
2018-07-22

Get the result using Landing outcomes with ascending, the first record will be first date of landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here

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

```
[35]: %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

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

[35]:

Booster_Version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

- Display the booster version range in between 4000 to 6000

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

List the total number of successful and failure mission outcomes

```
[37]: %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

[37]:

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

- Using count and group by we can calculate the success and failures.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
[41]: %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

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

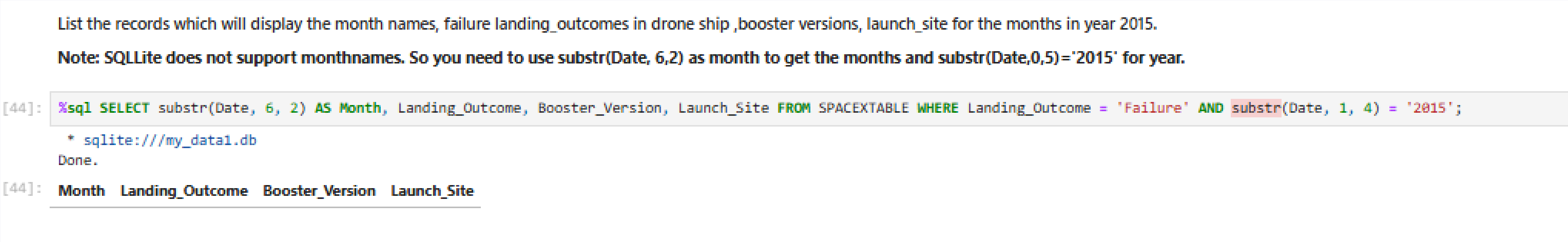
[41]:

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

- Above query display list of distinct booster with maximum payload mass

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here


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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
[44]: %sql SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure' AND substr(Date, 1, 4) = '2015';
```

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

[44]:

Month	Landing_Outcome	Booster_Version	Launch_Site

- Record not available for 2015

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

- Present your query result with a short explanation here

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.

```
[46]: %sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP BY Landing_Outcome ORDER BY OutcomeCount DESC;

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

[46]:

Landing_Outcome	OutcomeCount
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

- Based on date range records are displayed

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title

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

- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

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

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

- Explain the important elements and findings on the screenshot

Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

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

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

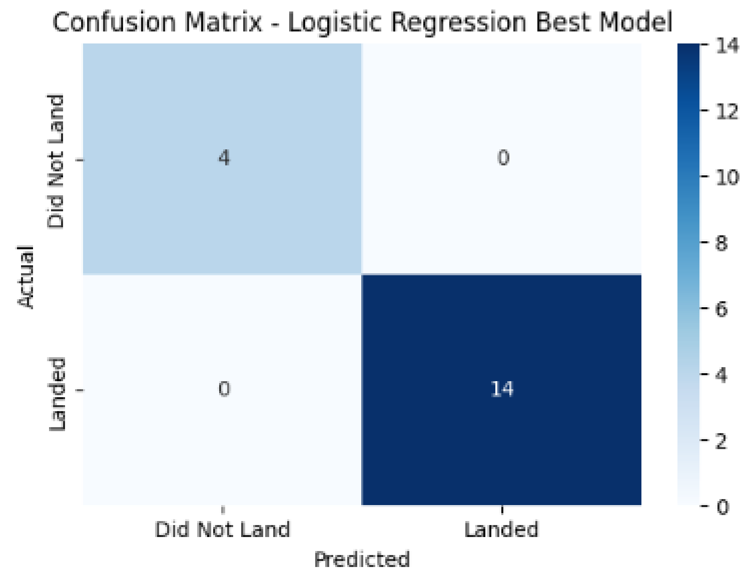
- Find which model has the highest classification accuracy

    Best performing model: Logistic Regression with accuracy 0.83

Confusion Matrix

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

- After converting categorical features into numeric form, logistic regression runs successfully.
- **Best parameters** (from GridSearchCV) give the highest cross-validation accuracy.
- The **confusion matrix** shows counts of correctly and incorrectly predicted landings.
- **True Positives (Landed correctly predicted)** and **True Negatives (Did Not Land correctly predicted)** are high.
- **False Positives/Negatives** are relatively low.
- The **classification report** confirms good accuracy, precision, recall, and F1-score, meaning the model predicts SpaceX landing success reliably.




```
Best Parameters: {'C': 10, 'penalty': 'l1'}
Best Cross-Validation Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

Did Not Land       1.00      1.00      1.00         4
      Landed       1.00      1.00      1.00        14

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18
```

Conclusions

- **Point 1:** SpaceX has significantly improved its launch success over the years, with recent launches achieving near-perfect success rates.
- **Point 2:** Launch success varies by **launch site** and **orbit type**, indicating certain sites and orbits are more reliable for successful missions.
- **Point 3: Payload mass** does not strongly determine launch success, but extremely heavy or unusually light payloads can affect outcomes.
- **Point 4: Booster reuse** is effective—reused boosters perform nearly as well as new boosters, supporting cost-efficient space missions.
- **Point 5:** Higher **flight numbers** (experience) correlate with improved success, suggesting operational learning and optimization over time.
- **Point 6:** Visual analysis using scatter, bar, and line plots helps identify patterns and trends in SpaceX launches, which can guide future mission planning.

Appendix

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

Thank you!