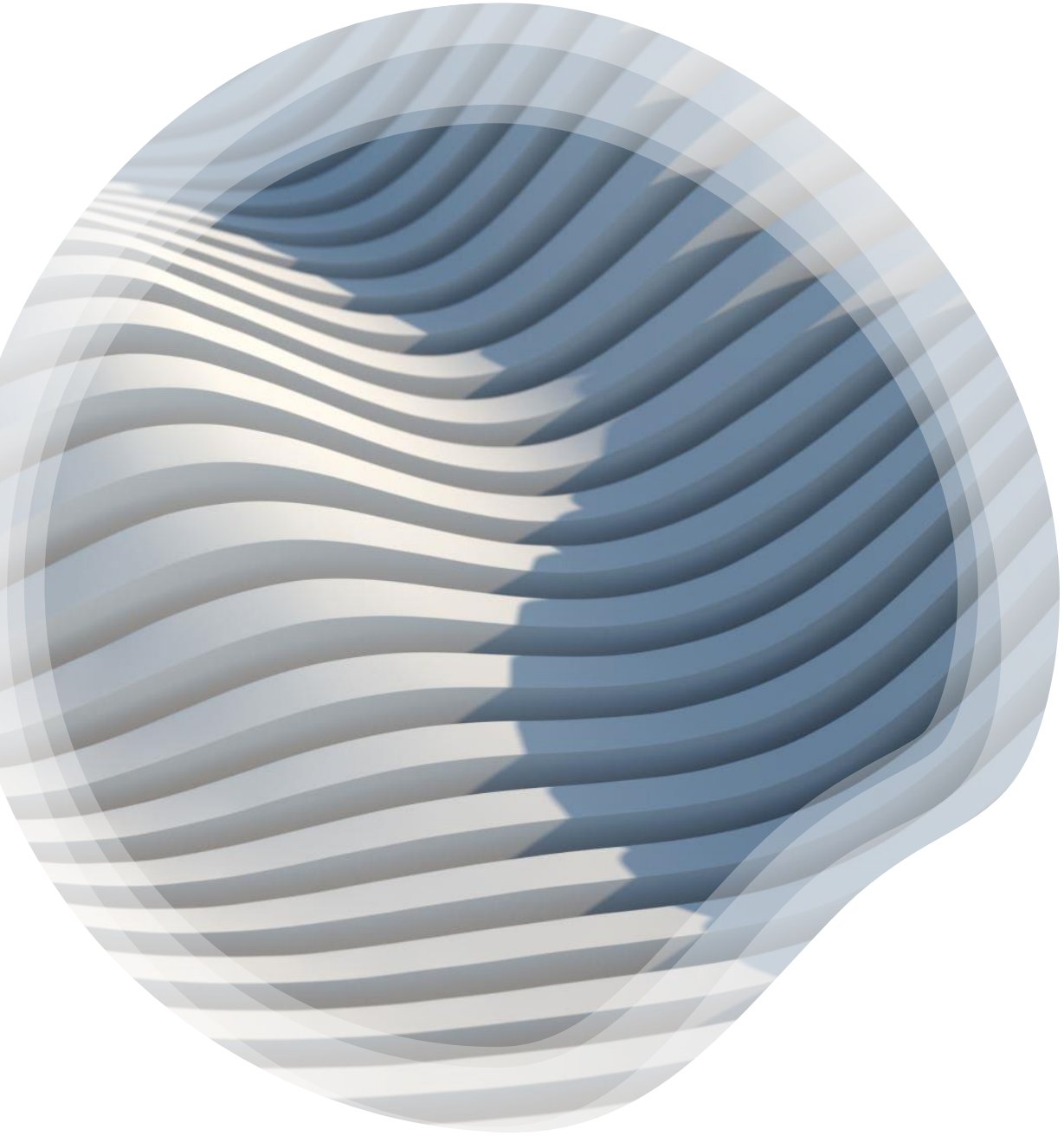
IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Kenroy Taylor
January 3, 2025.



# Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

# Executive Summary

## Summary of methodologies

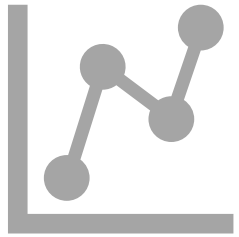
Data Collection

Data Wrangling

Exploratory Data Analysis

Interactive visual Analytics

Predictive Analysis



## Summary of all results

Exploratory Data Analysis

Interactive visual Analytics

Predictive Analysis

# Introduction

Space Y is embarking on a mission to revolutionize the commercial space industry by offering innovative and cost-effective launch solutions. Inspired by SpaceX's achievements, including reusable rocket technology and competitively priced Falcon 9 launches at $62 million, we aim to exceed the industry standard where competitors charge upwards of $165 million per launch. By leveraging data science and machine learning, Space Y will analyze publicly available data to gain insights into launch performance, predict rocket reusability, and optimize our cost structures. Our goal is to establish Space Y as a formidable competitor, making space exploration more accessible and affordable for all.

To address Space Y's competitive strategy, we aim to answer: What factors influence the likelihood of the SpaceX Falcon 9 first stage landing successfully, and how can predictive models be utilized to forecast the probability of a successful landing for future launches?

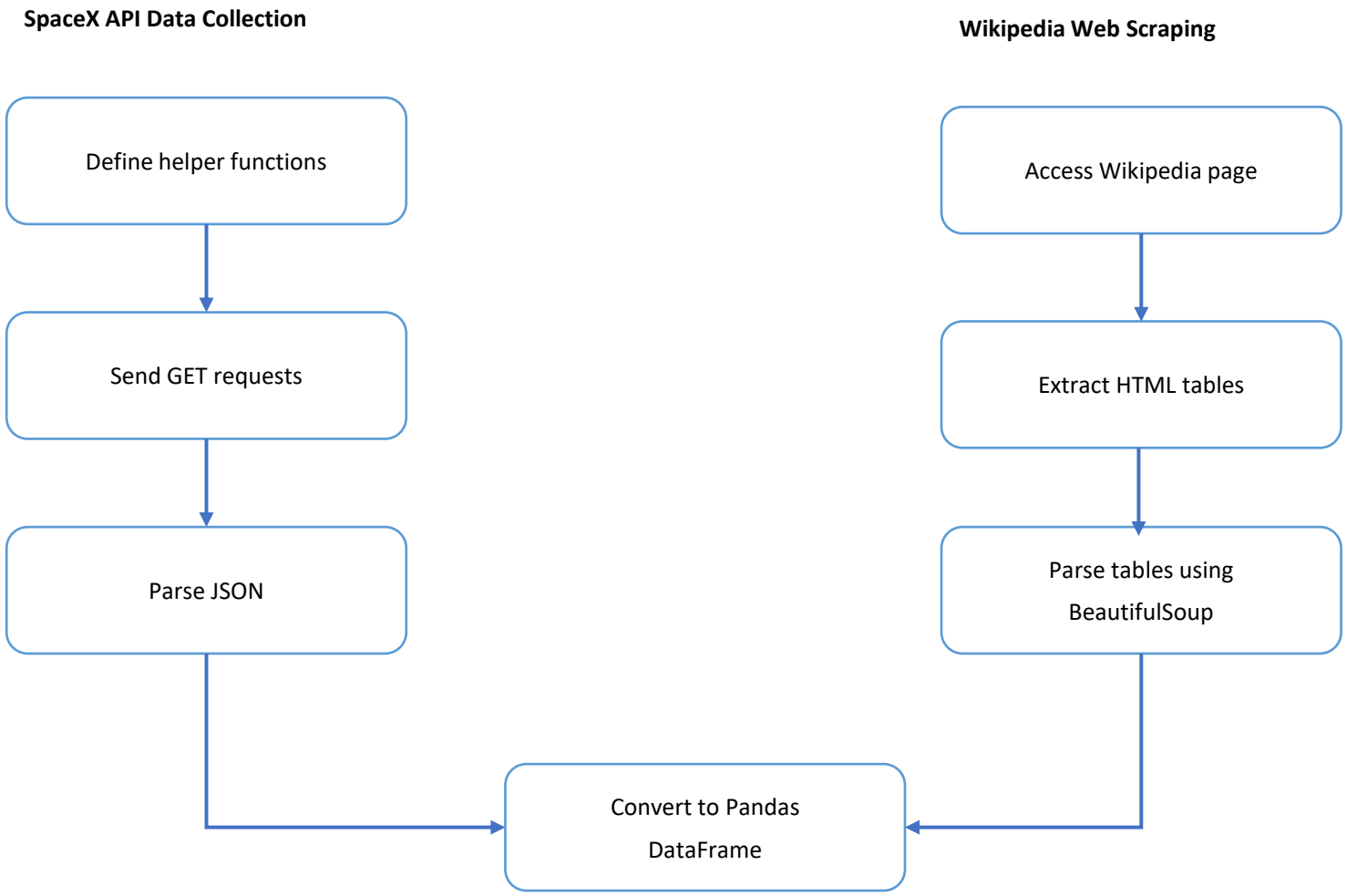
Section 1

# 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



# Data Collection – SpaceX API



Utilizing the SpaceX API to obtain comprehensive information about launches, details regarding the rockets employed, payloads delivered, launch parameters, landing specifications, and the outcomes of the landing events.

See Link:
https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/jupyter-labs-spacex-data-collection-api.ipynb

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

We should see that the request was successfull with the 200 status response code

```
response=requests.get(static_json_url)
```

```
response.status_code
```

```
200
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(requests.get(static_json_url).json())
# print(response.content))
```

```
# Create a data from launch_dict
df = pd.DataFrame(launch_dict)
```

# Data Collection – Scraping



Conducting web scraping to gather historical launch records of Falcon 9 from the Wikipedia page.

See Link:
https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/jupyter-labs-webscraping.ipynb

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
# Use soup.title attribute
print("Webpage Title:", soup.title.string)
```

```
Webpage Title: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

```
# Display the extracted data

df = pd.DataFrame(launch_dict)
df.head()
```

# Data Wrangling

To predict whether a booster will successfully land, a binary column is introduced with values 1 or 0, indicating a successful or unsuccessful landing, respectively. The process includes defining a set of failure outcomes (bad outcome), creating a landing class list to categorize each row, adding a Class column to store these classifications, and exporting the updated DataFrame as a .csv file.

See Link: https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

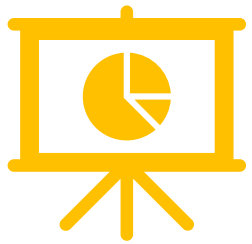
```
# landing_class = 0 if bad_outcome
bad_outcome = {"False Ocean", "None None", "False RTLS", "False ASDS"}
# landing_class = 1 otherwise
landing_class = [0 if outcome in bad_outcome else 1 for outcome in df['Outcome']]
# Assign the list to a new column in the dataframe
df['LandingClass'] = landing_class

# Display the first few rows to verify
df[['Outcome', 'LandingClass']].head()
```

```
df["Class"].mean()
```

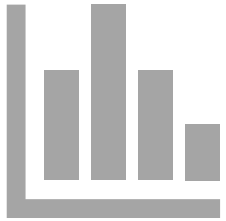
```
np.float64(0.6888888888888889)
```

# EDA with Data Visualization



SCATTER PLOT: Scatter plots are used to explore the relationships or correlations between two numerical variables. The following relationships were visualized:

Flight Number vs. Launch SitePayload vs. Launch SiteOrbit Type vs. Flight NumberPayload vs. Orbit Type



BAR GRAPH: Bar graphs are employed to compare numerical data with categorical variables. Both horizontal and vertical bar graphs can be utilized, depending on the dataset size. The relationship analyzed was Success Rate vs. Orbit Type

LINE GRAPH: Line graphs display numerical data on both axes and are typically used to show changes in a variable over time. The relationship visualized wa Success Rate vs. Year (annual trend in launch success)

See link: https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/edadataviz.ipynb

# EDA with SQL

- Perform SQL queries on the dataset to analyze space missions.
- Gather insights into:
  - Unique launch site names.
  - Launch site records starting with 'CCA’.
  - Total payload mass for boosters launched by NASA (CRS).
  - Average payload mass carried by booster version F9 v1.1.
  - Key dates, such as the first successful landing on a ground pad.
- Identify boosters with:
  - Specific payload mass ranges (4000–6000 kg).
  - Landing outcomes on drone ships.
- Determine:
  - Total mission successes and failures.
  - Boosters with the maximum payload capacity.
- Rank the count of landing outcomes (e.g., Failure or Success) between specific dates (2010-06-04 to 2017-03-20) in descending order.

- See link: https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium



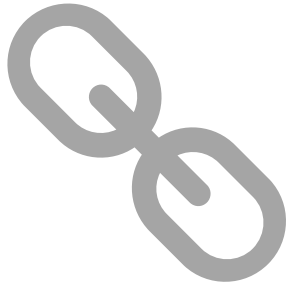
**Below are the features used to generate the interactive Map**

- Markers and circles were used to highlight the locations of all launch sites on the map for clear geographic representation. To visualize launch outcomes.
- Color-coded markers (green for successes and red for failures) and clustered overlapping launches for better organization.
- Distance markers and lines (Polylines) were used to show the proximity of each launch site to nearby points, providing insights into spatial relationships.

See link: https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

The dashboard includes two primary plots: a pie chart and a scatter plot. The pie chart visually represents the total successful launches per site, making it easy to identify the most successful sites and providing the option to filter by site to examine success/failure ratios. The scatter plot explores the correlation between launch outcomes (success or failure) and payload mass, with additional filtering options for payload mass ranges and booster versions to allow for more detailed analysis.



See link: https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/spacex_dash_app.py

# Predictive Analysis (Classification)



To summarize the process:

1. Model Development: Load the dataset, preprocess (including standardization), and split it into training and test sets using train_test_split(). Select appropriate machine learning algorithms for the task.

2. Model Evaluation and Improvement: For each algorithm, create a GridSearchCV object with hyperparameters, train the model on the training data, and evaluate using accuracy scores, confusion matrices, and tuned hyperparameters (best_params_).

3. Finding the Best Model: Compare accuracy scores across models and select the model with the highest score as the best-performing classification model.

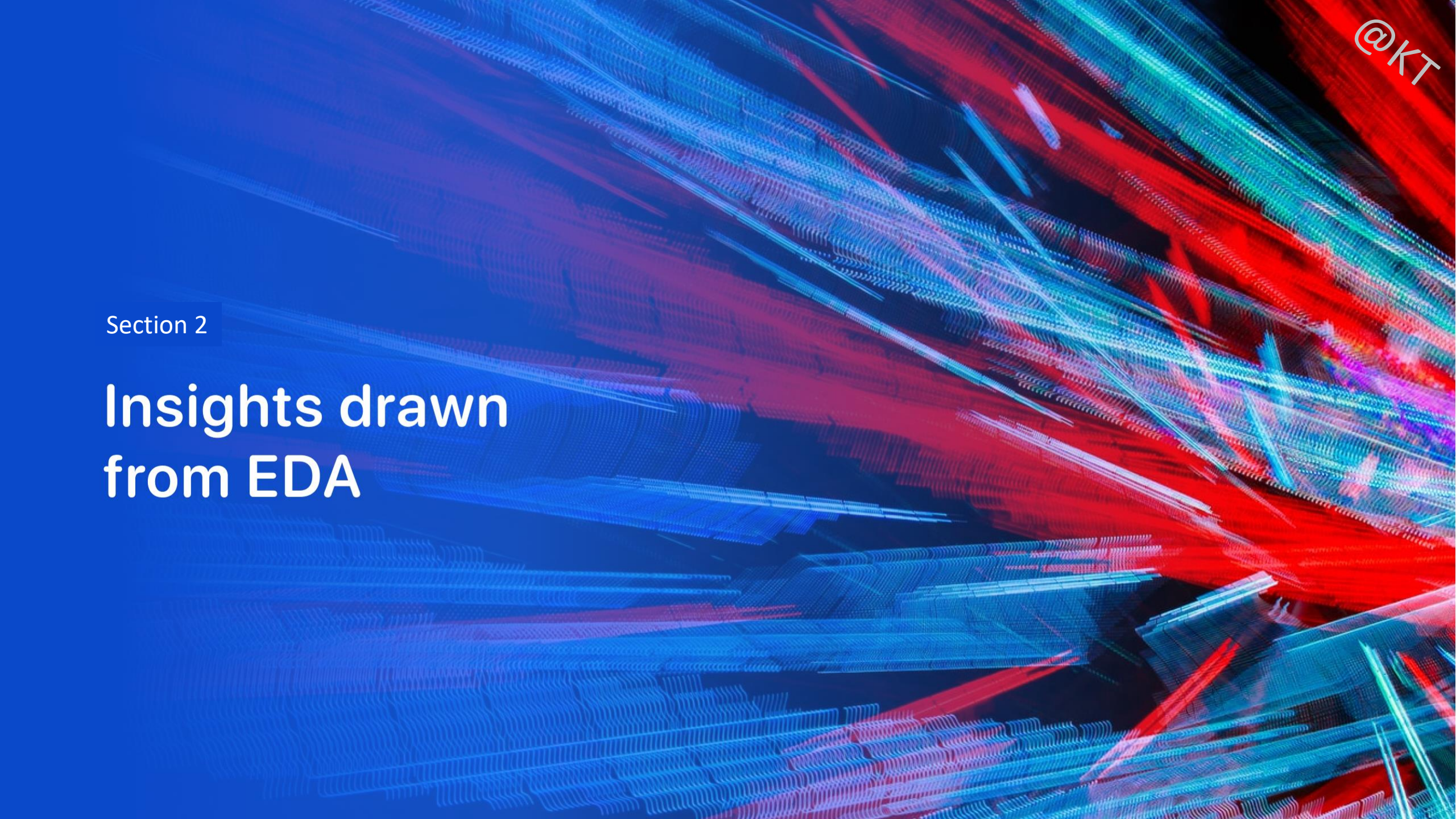
See link: https://github.com/kennyt1-hub/testrepo/blob/main/Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results

EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

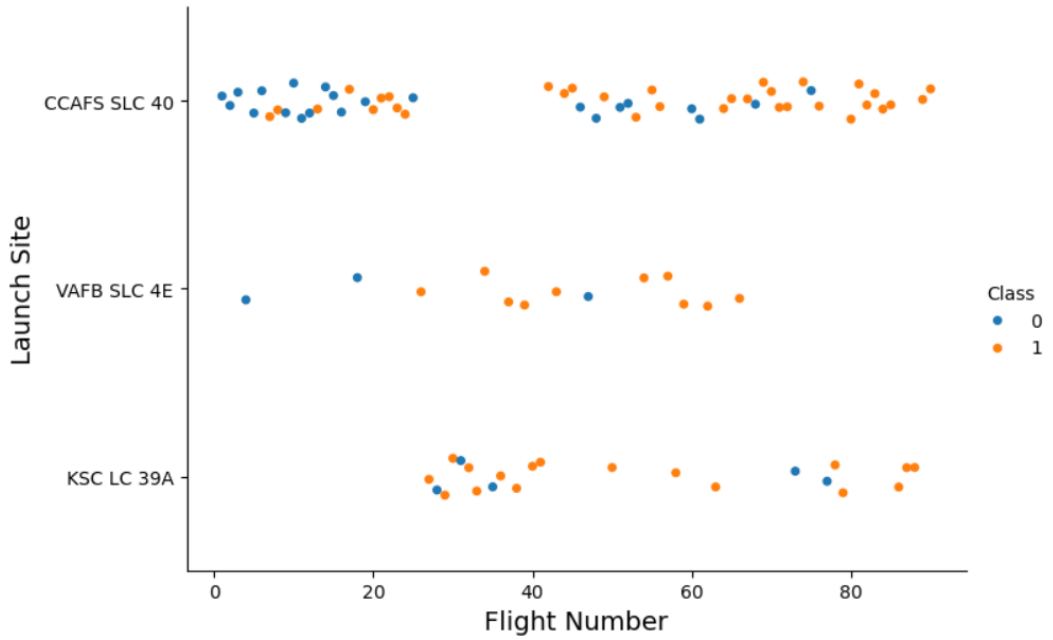



Section 2

# Insights drawn from EDA

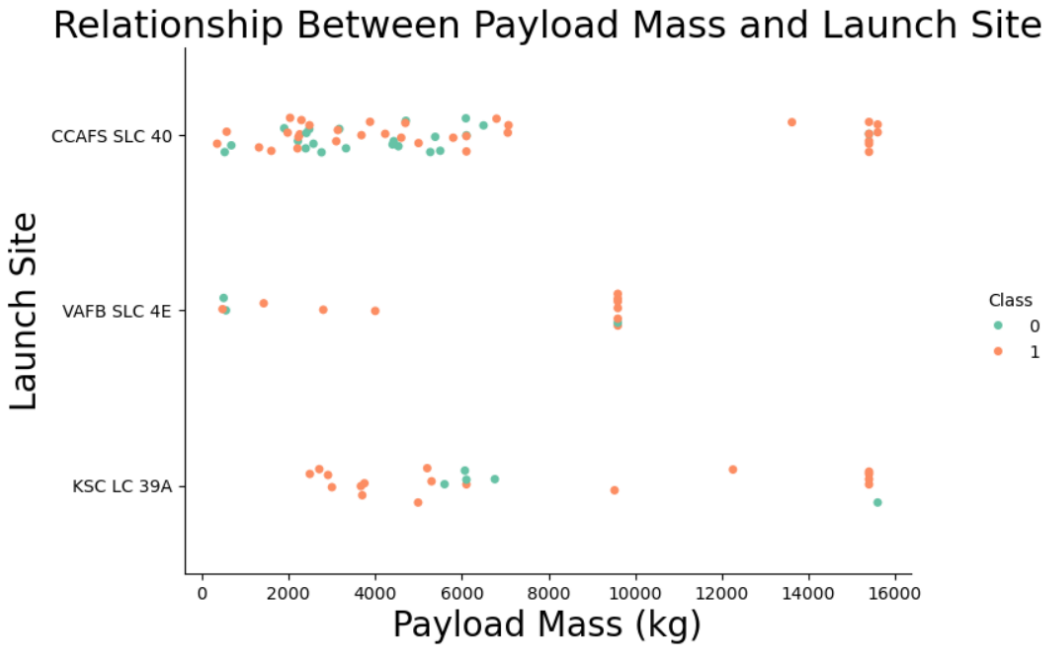
# Flight Number vs. Launch Site

- The graph shows the distribution of SpaceX flight outcomes (success denoted as Class 1 and failure as Class 0) across different launch sites (CCAFS SLC 40, VAFB SLC 4E, and KSC LC 39A) against flight numbers. Each dot represents a flight, and the success rate varies by launch site, with CCAFS SLC 40 and KSC LC 39A showing higher frequencies of successful launches compared to VAFB SLC 4E.



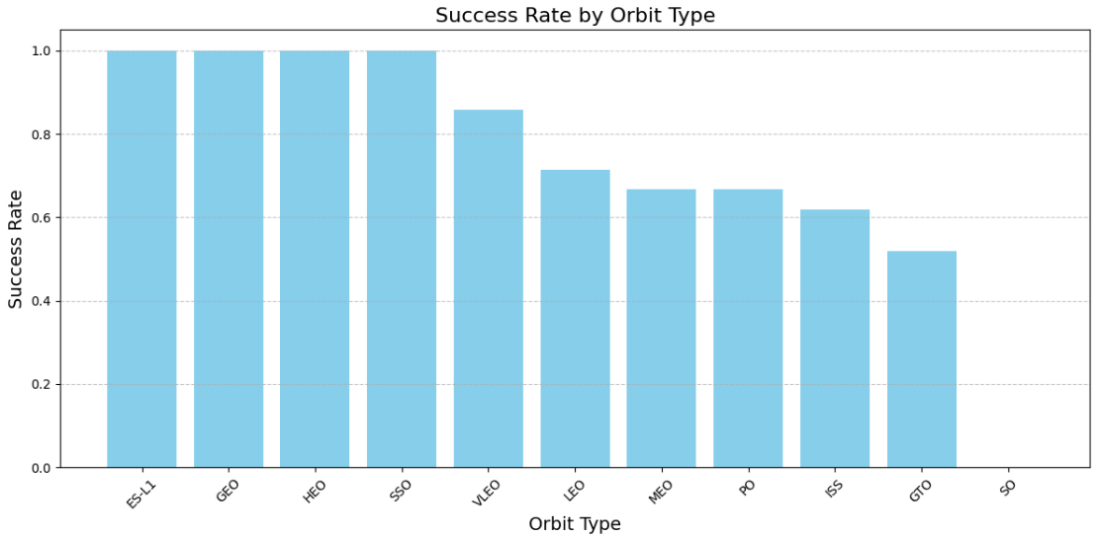
# Payload vs. Launch Site

- The graph depicts the relationship between payload mass (in kg) and the launch site, highlighting the success (Class 1) and failure (Class 0) of SpaceX launches. It shows that payload masses vary significantly across the sites, with heavier payloads (above 10,000 kg) primarily launched from CCAFS SLC 40 and achieving more successful outcomes.



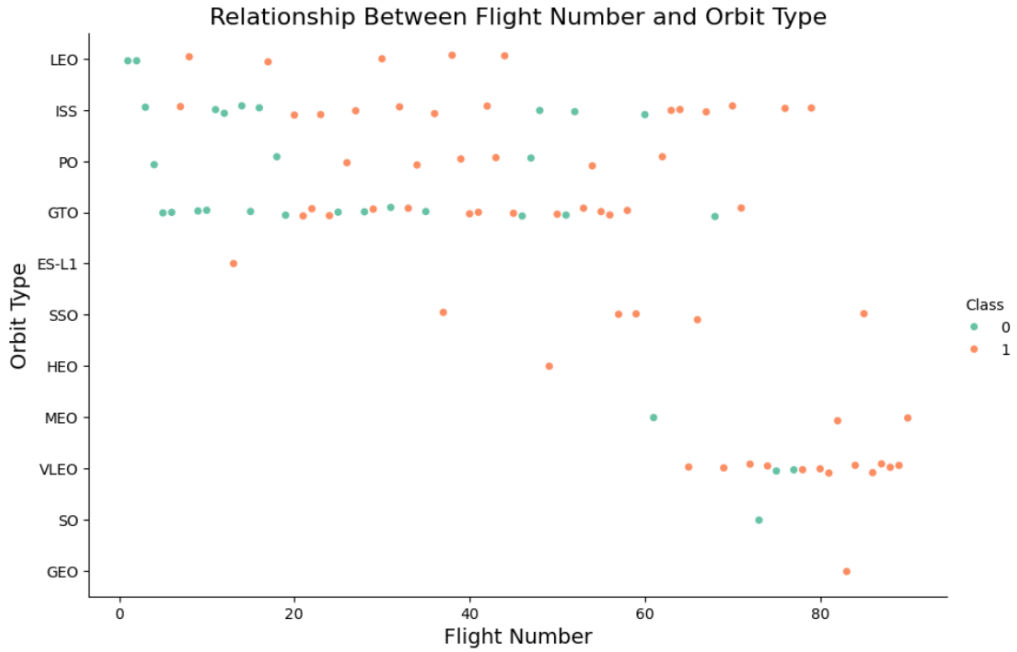
# Success Rate vs. Orbit Type

- The bar chart illustrates the success rate of SpaceX launches categorized by orbit type. It reveals that certain orbit types, such as ES-L1, GEO, HEO, SSO, and VLEO, have a perfect success rate (100%), while others like GTO and SO have comparatively lower success rates.



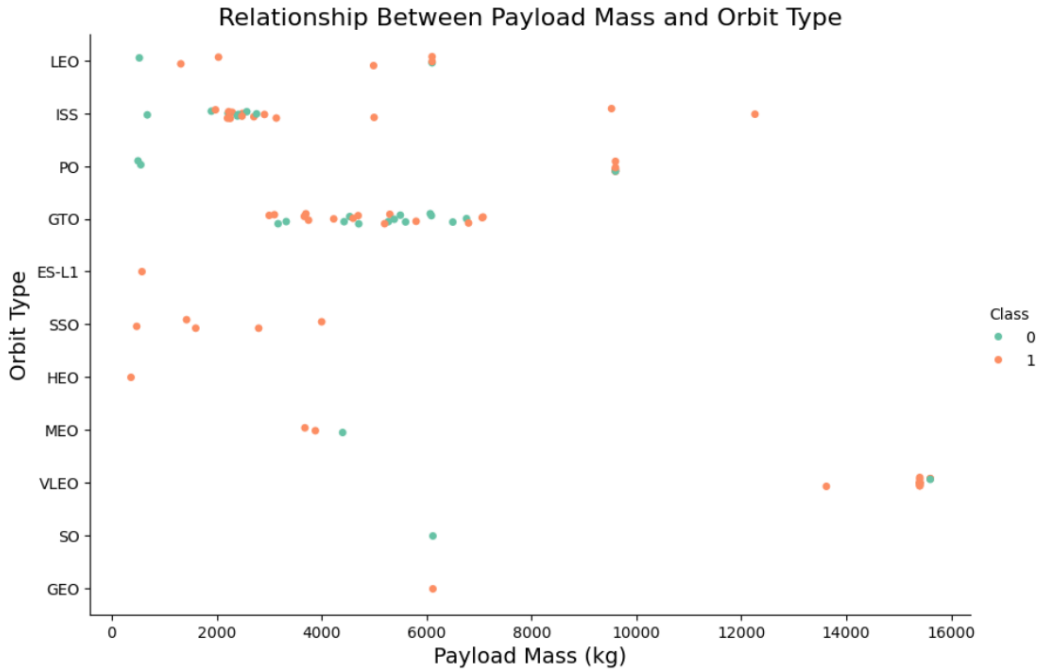
# Flight Number vs. Orbit Type

- The scatter plot illustrates the relationship between flight numbers and orbit types, showing the outcomes of SpaceX launches (Class 1 for success and Class 0 for failure). It indicates that higher flight numbers generally correspond to more successful launches across various orbit types, suggesting improved reliability over time.



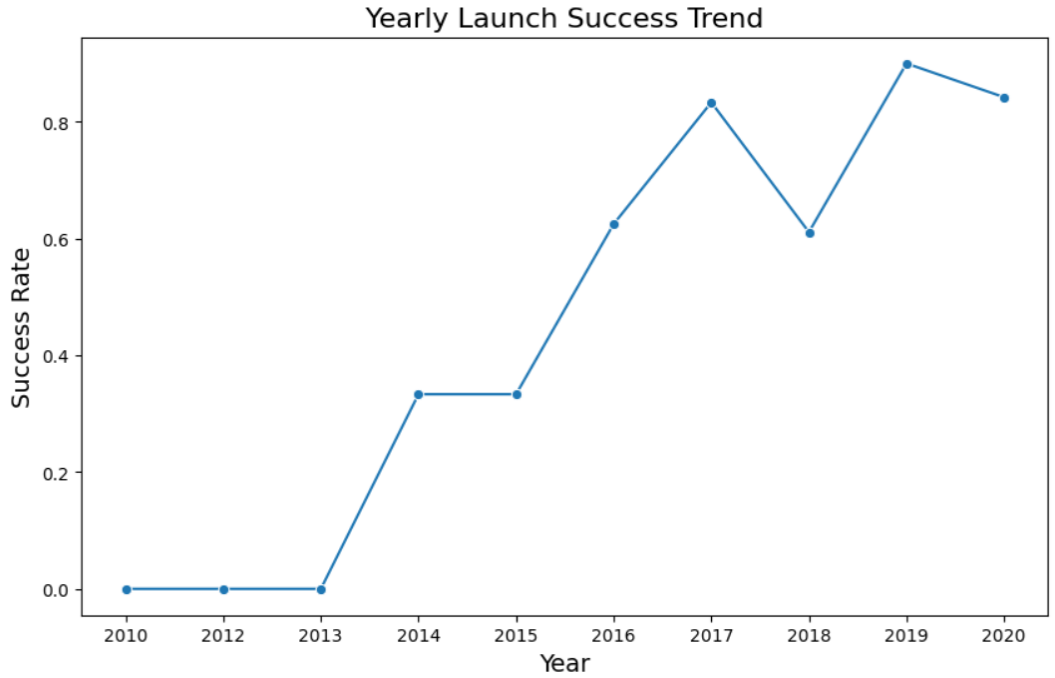
# Payload vs. Orbit Type

- The scatter plot shows the relationship between payload mass (in kg) and orbit types, with the outcomes of SpaceX launches indicated as successful (Class 1) or failed (Class 0). It highlights that successful launches (Class 1) occur across a range of payload masses and orbit types, particularly for ISS and GTO orbits, while higher payloads (above 10,000 kg) tend to have consistent success.



# Launch Success Yearly Trend

- The line chart depicts the yearly trend in SpaceX's launch success rate from 2010 to 2020. It shows a consistent improvement in success rates over the years, with significant growth after 2013 and a peak in 2019, reflecting advancements in reliability and technology.



# All Launch Site Names

- The SQL query SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE; retrieves the unique launch sites from the SPACEXTABLE database. The result shows three distinct launch sites: CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A, with a duplicate entry for CCAFS LC-40 likely due to formatting or data inconsistencies.

```
%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'

- The SQL query SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5; retrieves the first five records from the SPACEXTABLE where the launch site name starts with "CCA". The result includes detailed information about launches from the CCAFS LC-40 site, such as date, payload, payload mass, orbit, customer, mission outcome, and landing outcome.

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

```
 * 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 |

# Total Payload Mass

The SQL query SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)'; calculates the total payload mass carried for NASA (CRS) missions from the SPACEXTABLE. The result shows that the total payload mass for these missions is 45,596 kg.

```
%sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

| Total_Payload_Mass |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

- The SQL query SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1'; calculates the average payload mass for launches using the booster version 'F9 v1.1'. The result shows that the average payload mass for these launches is 2,928.4 kg.

```
%sql SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

| Average_Payload_Mass |
|---|
| 2928.4 |

# First Successful Ground Landing Date

- The SQL query SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)'; retrieves the earliest date of a successful ground pad landing from the SPACEXTABLE. The result indicates that the first successful ground pad landing occurred on December 22, 2015.

```
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

| First_Successful_Landing |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL query SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass__kg_" > 4000; retrieves the unique booster versions that successfully landed on a drone ship while carrying a payload mass greater than 4000 kg. The result lists specific booster versions, such as F9 FT B1022, F9 FT B1026, F9 FT B1021.2, and F9 FT B1031.2, meeting the criteria.

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "Payload_Mass_
```

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

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

# Total Number of Successful and Failure Mission Outcomes

- The SQL query SELECT "Mission_Outcome", COUNT(*) AS Count FROM SPACEXTABLE GROUP BY "Mission_Outcome"; groups all records in the SPACEXTABLE by their mission outcome and counts the occurrences of each outcome. The result shows the distribution of mission outcomes, including 98 successes, 1 in-flight failure, and 2 other success-related outcomes with additional details.

```
%sql SELECT "Mission_Outcome", COUNT(*) AS Count FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

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

# Boosters Carried Maximum Payload

- The SQL query SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE); retrieves the booster versions associated with the maximum payload mass recorded in the SPACEXTABLE. The result lists multiple booster versions capable of carrying the heaviest payload, indicating their significant payload capacity.

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = ( SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABL
```

```
 * 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 |

```
%sql SELECT substr("Date", 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE substr
```

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

| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

# 2015 Launch Records

- The SQL query SELECT substr("Date", 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE substr("Date", 1, 4) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'; extracts the month from the date and filters records from the year 2015 where the landing outcome was a drone ship failure. The query returns details of such failures, including the month, booster version, and launch site.

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '20
```

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

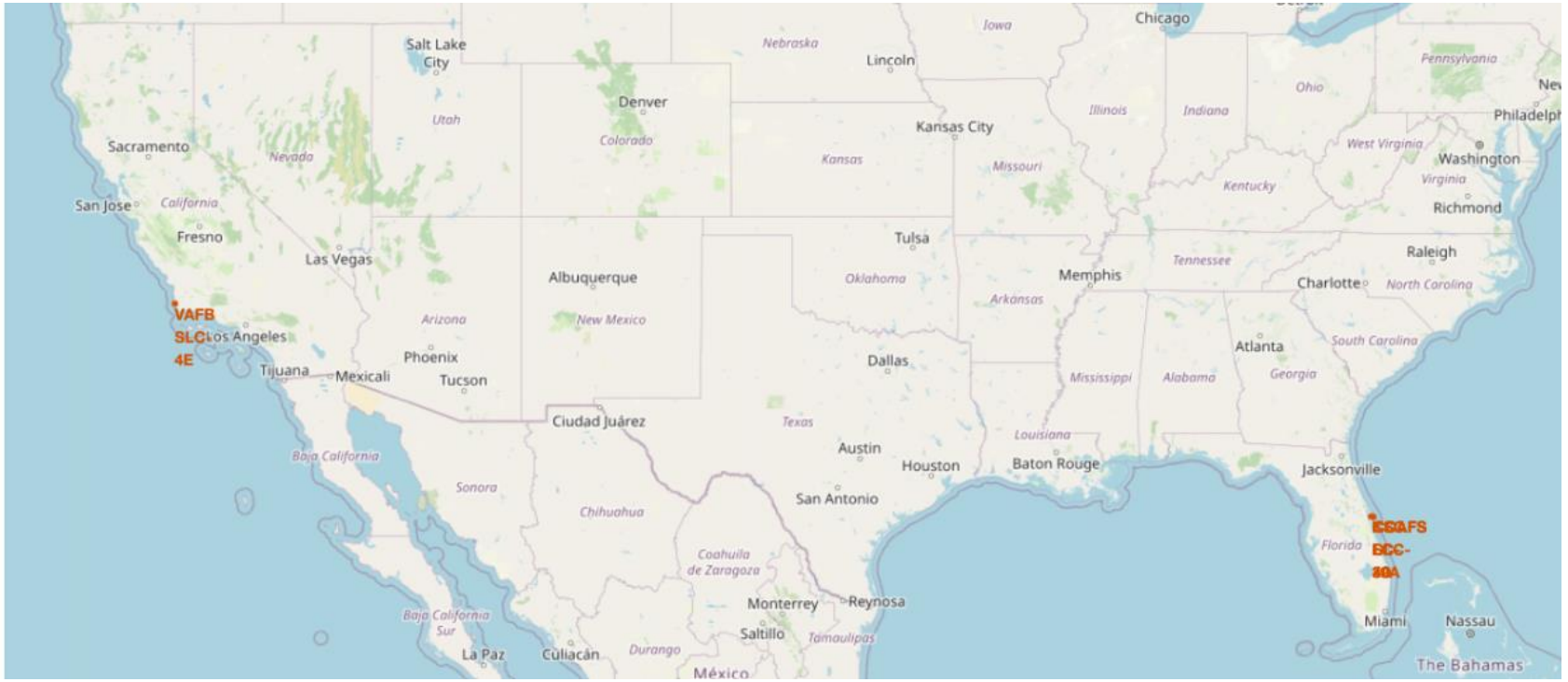
| Landing_Outcome | Count |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

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

- The SQL query SELECT "Landing_Outcome", COUNT(*) AS Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Count DESC; counts the occurrences of each landing outcome for launches conducted between June 4, 2010, and March 20, 2017. The results are grouped by landing outcome and sorted in descending order of count, showing the most frequent outcomes during the specified time period.


Section 3

# Launch Sites Proximities Analysis

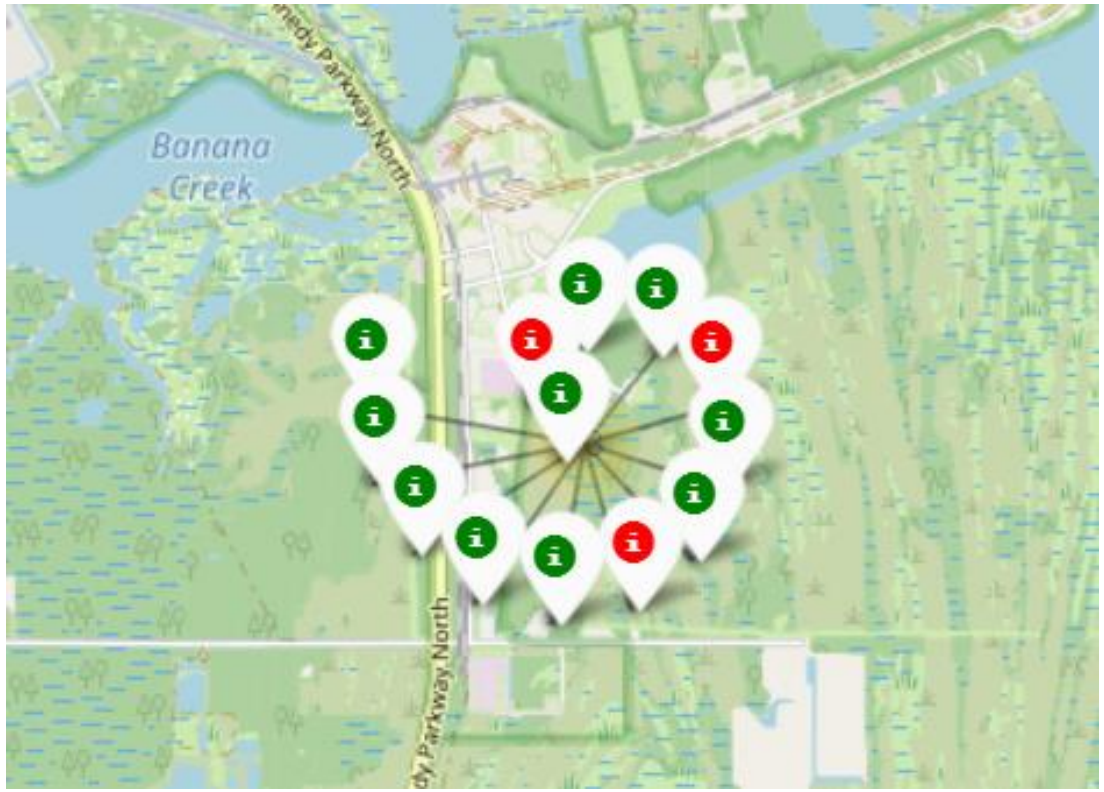



# All launch sites

Launch sites are predominantly located near coastal areas, with Vandenberg Space Force Base being the sole facility on the U.S. Pacific West Coast, situated in California. All other launch sites are concentrated along the U.S. Atlantic East Coast, primarily in the state of Florida.

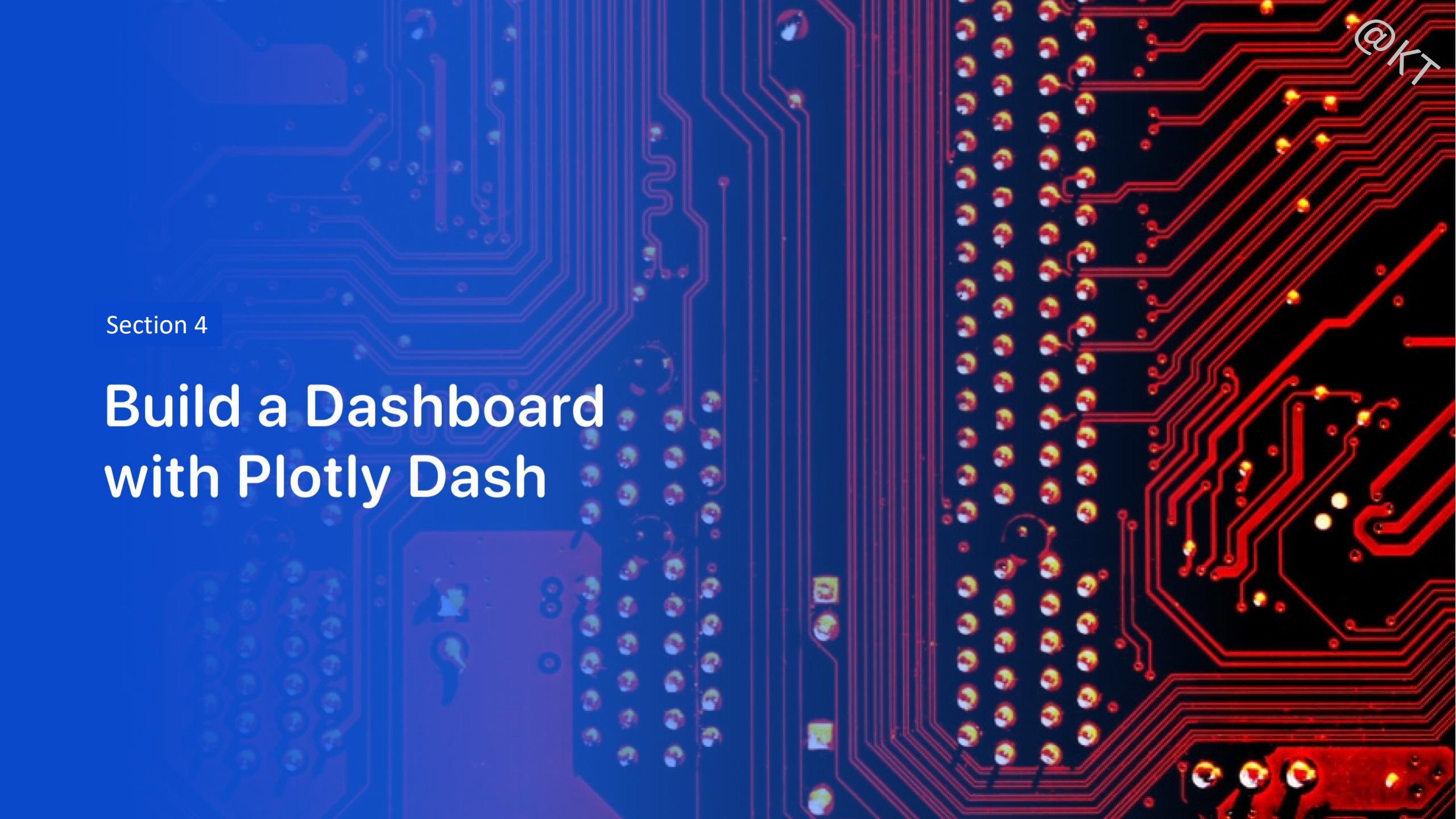
# Success and Failure Launches

- Color-coded markers make it easy to identify launch sites with higher success rates: green for successful launches and red for failures. The KSC LC-39A launch site stands out with a very high success rate.



# Approximate Launch Sites

- The map visualizes the proximity between two points: a launch site (right) and a coastline marker (left), connected by a purple line representing the measured distance. The numbers "33" and "13" indicate markers associated with specific attributes or distances at their respective locations.

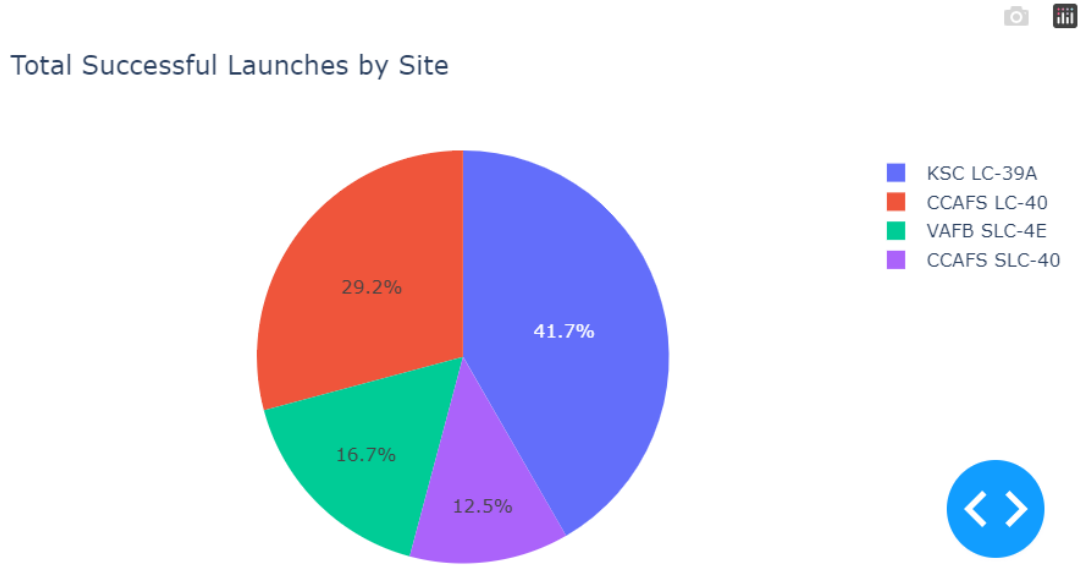



Section 4

# Build a Dashboard with Plotly Dash

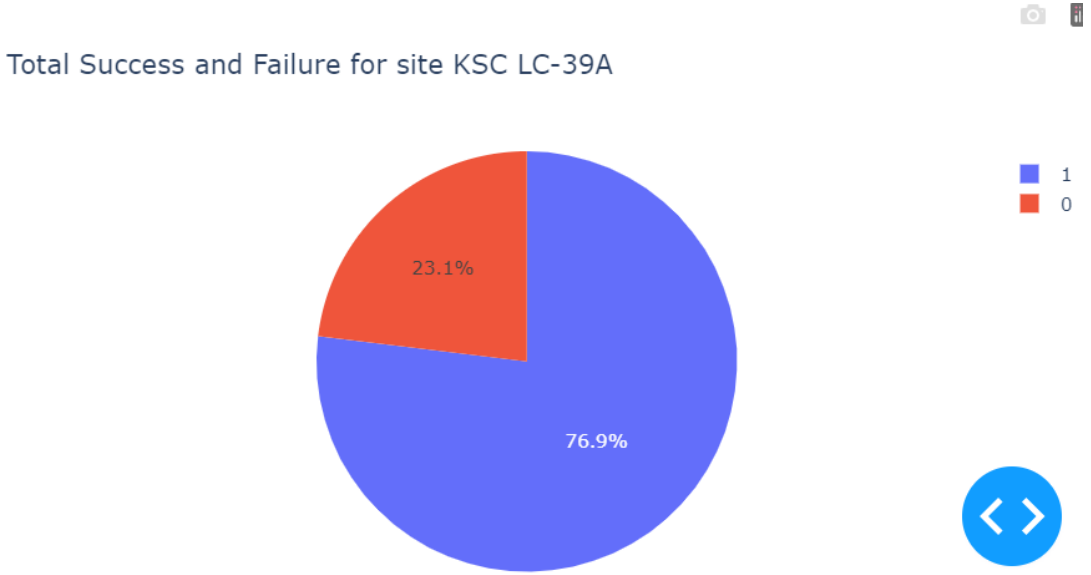
# Total Successful

This pie chart represents the percentage distribution of total successful launches by site. The majority of launches (41.7%) occurred at KSC LC-39A, followed by 29.2% at CCAFS LC-40, 16.7% at VAFB SLC-4E, and 12.5% at CCAFS SLC-40.



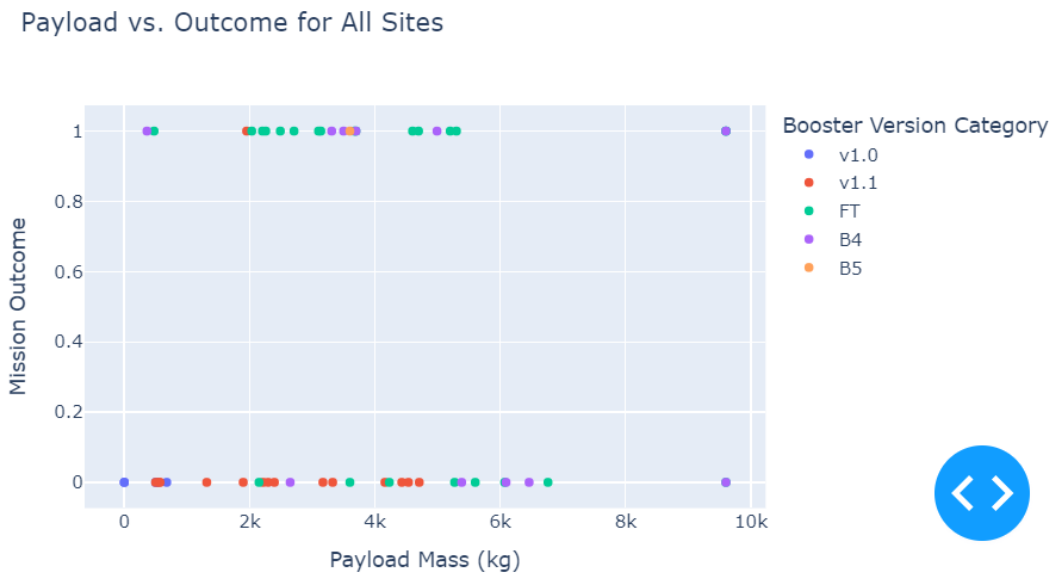
# Launches at the KSC LC-39A site

This pie chart represents the success (blue) and failure (red) rates for launches at the KSC LC-39A site. The majority of launches (76.9%) were successful, while 23.1% of launches resulted in failure.



# Payload vs. Outcome

This scatter plot shows the relationship between payload mass (x-axis) and mission outcomes (y-axis) for various booster version categories. Successful missions (y=1) are distributed across different payload masses and booster versions, while failed missions (y=0) are less frequent and also vary in payload and booster type.

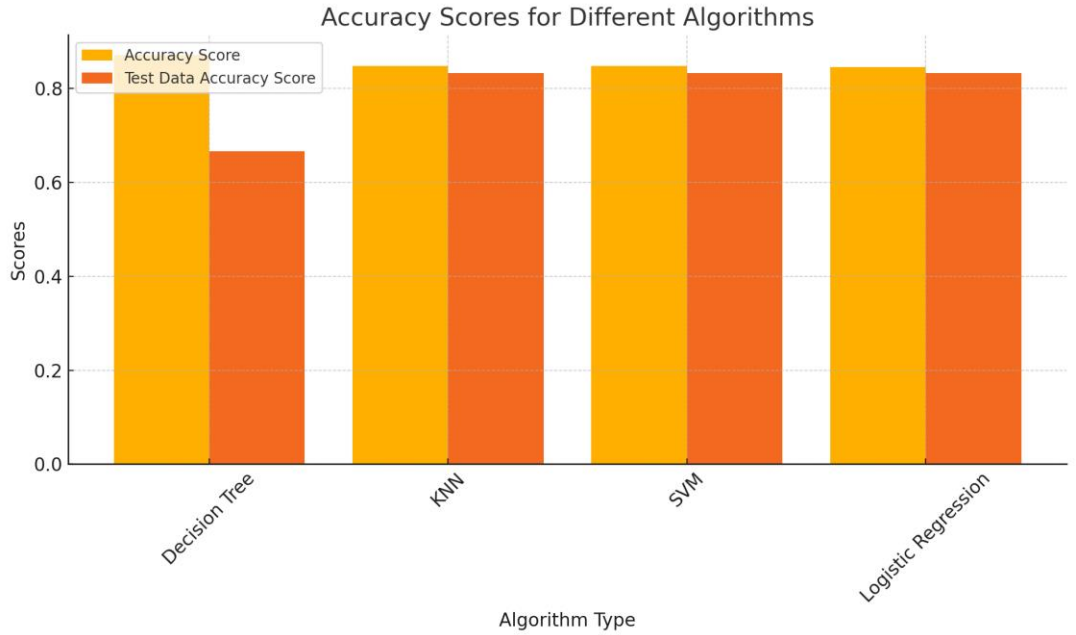



Section 5

# Predictive Analysis (Classification)

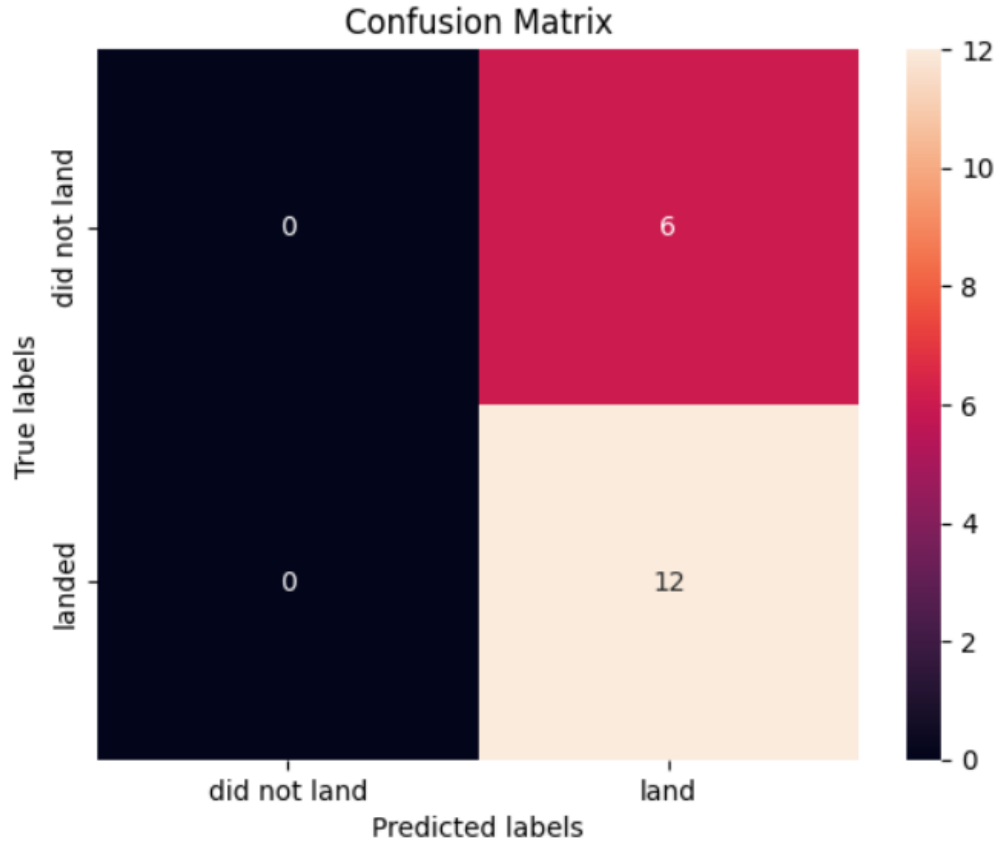
# Classification Accuracy

| | Algo Type | Accuracy Score | Test Data Accuracy Score |
|---|---|---|---|
| **2** | Decision Tree | 0.871429 | 0.666667 |
| **3** | KNN | 0.848214 | 0.833333 |
| **1** | SVM | 0.848214 | 0.833333 |
| **0** | Logistic Regression | 0.846429 | 0.833333 |



# Confusion Matrix

This confusion matrix shows the performance of a classification model where all "landed" instances (12) are correctly predicted, but all "did not land" instances (6) are misclassified as "landed." The model demonstrates a high bias towards predicting "landed," leading to no true negatives or false negatives.



# Conclusions

- Optimize Launch Sites: Prioritize launches from KSC LC-39A and CCAFS LC-40 due to their higher success rates and efficiency with heavier payloads, as identified through data analysis.
- Enhance Payload Strategies: Focus on maximizing payload capacities for specific orbit types, such as ISS and GTO, which demonstrate consistent success trends, ensuring mission reliability.
- Leverage Predictive Models: Implement the developed classification models to forecast mission outcomes and proactively address potential failure risks, improving overall operational planning.
- Invest in Orbit-Specific Technologies: Target advancements for lower-performing orbit types (e.g., GTO) to align their success rates with consistently successful orbits like GEO and SSO.

# Appendix

- Acknowledgments
  - Data sources: SpaceX API and web scraping tools.
  - Python libraries used: Pandas, Matplotlib, Seaborn, Scikit-learn, Plotly, Folium.

- Project WorkflowData

1. Collection: Gathered data using the SpaceX API and web scraping techniques.
2. Data Wrangling: Cleaned and structured data for analysis.
3. Exploratory Data Analysis (EDA): Visualized trends and key insights using SQL and Python.
4. Predictive Modeling: Developed and optimized classification models for launch success predictions.
5. Visualization: Created interactive dashboards and geographic maps for better data interpretation.

# Thank you!