IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Leilane Aragão
29/12/2025

Outline

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

Executive Summary

- Launch Success Patterns
- Payload Insights
- Booster & Landing Performance
- Machine Learning Outcomes
- Final Takeaways

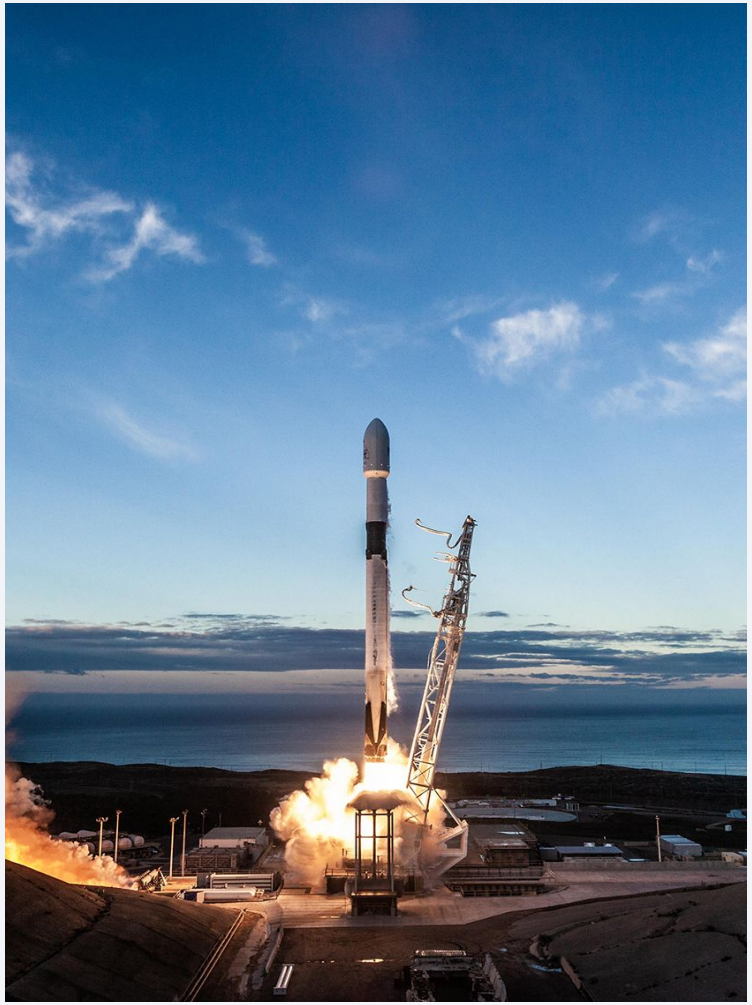
Falcon 9 launch success is influenced by a combination of:

- Orbit type
- Launch site
- Payload mass
- Booster version and reuse

Machine learning models can reliably predict mission success using engineered features

Introduction

- The commercial space age is advancing, with companies like Virgin Galactic, Rocket Lab, and Blue Origin making space travel more accessible.
- SpaceX stands out for its achievements, including sending spacecraft to the International Space Station and launching the Starlink satellite internet constellation.
- The Falcon 9 rocket's first stage is crucial for launch success and can be reused, significantly reducing costs.
- This project goal is predicting the likelihood of the Falcon 9's first stage landing successfully and creating dashboards to present your findings.



Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The SpaceX REST API provides data on launches, including rocket details, payloads, and landing outcomes.
- Perform data wrangling
  - The JSON response from the API is converted into a Pandas DataFrame using the json_normalize function for easier analysis.
- 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

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas dataframe

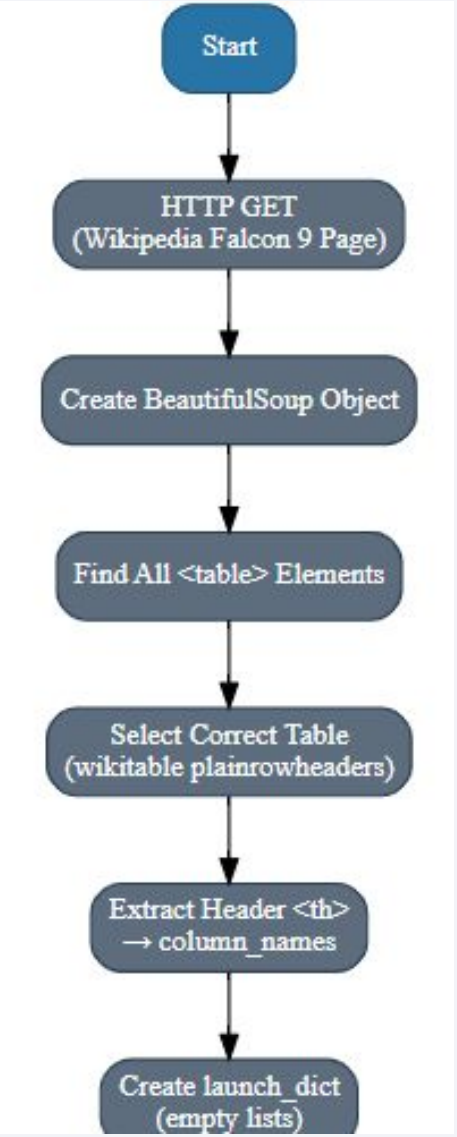
```
# use requests.get() method with the provided static_url and headers
# assign the response to a object
# Faz a requisição GET
response = requests.get(static_url, headers=headers)

# Verifica o status
print("Status code:", response.status_code)

# Mostra parte do conteúdo retornado
print(response.text[:500])
```

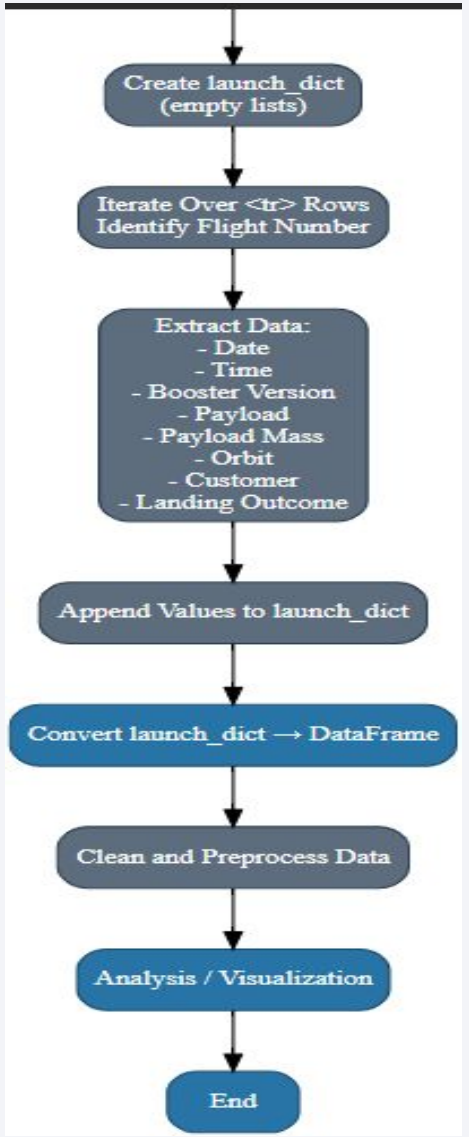
Data Collection – SpaceX API

- Performed REST API calls using the SpaceX v4 endpoints
- Retrieved past launch data from /launches/past
- Queried additional endpoints: /rockets, /payloads, /launchpads
- Merged launch data with rocket, payload, and launchpad metadata
- Built a structured dataset for analysis (JSON → Python → DataFrame)
- Ensured reproducibility through clean, modular code.
- GITHUB



Data Collection - Scraping

- Performed an HTTP GET request to retrieve the Falcon 9 launch list from Wikipedia
- Parsed the HTML using BeautifulSoup to navigate the document structure
- Identified and extracted the correct launch table (wikitable plainrowheaders)
- Extracted column headers dynamically from <th> elements
- Built a structured dictionary (launch_dict) to store cleaned data
- Iterated through table rows to extract:
  - Flight number
  - Date and time
  - Booster version
  - Launch site
  - Payload and mass
  - Orbit
  - Customer
  - Launch and landing outcomes
  - Converted the dictionary into a Pandas DataFrame
  - Cleaned and standardized fields for analysis (dates, masses, text normalization)
- GITHUB



Data Wrangling

- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
- Loaded raw scraped data into a Pandas DataFrame Standardized column names and removed irrelevant fields
- Cleaned inconsistent text values (e.g., whitespace, special characters, HTML artifacts)
- Converted payload mass values from strings to numeric (kg)
- Split combined fields (e.g., “Date and Time”) into separate Date and Time columns
- Normalized booster version names and landing outcomes
- Handled missing values using conditional parsing and fallback logic
- Converted date strings into Python datetime objects
- Ensured consistent data types across all columns
- Validated final dataset for completeness and correctness before analysis.
- GITHUB

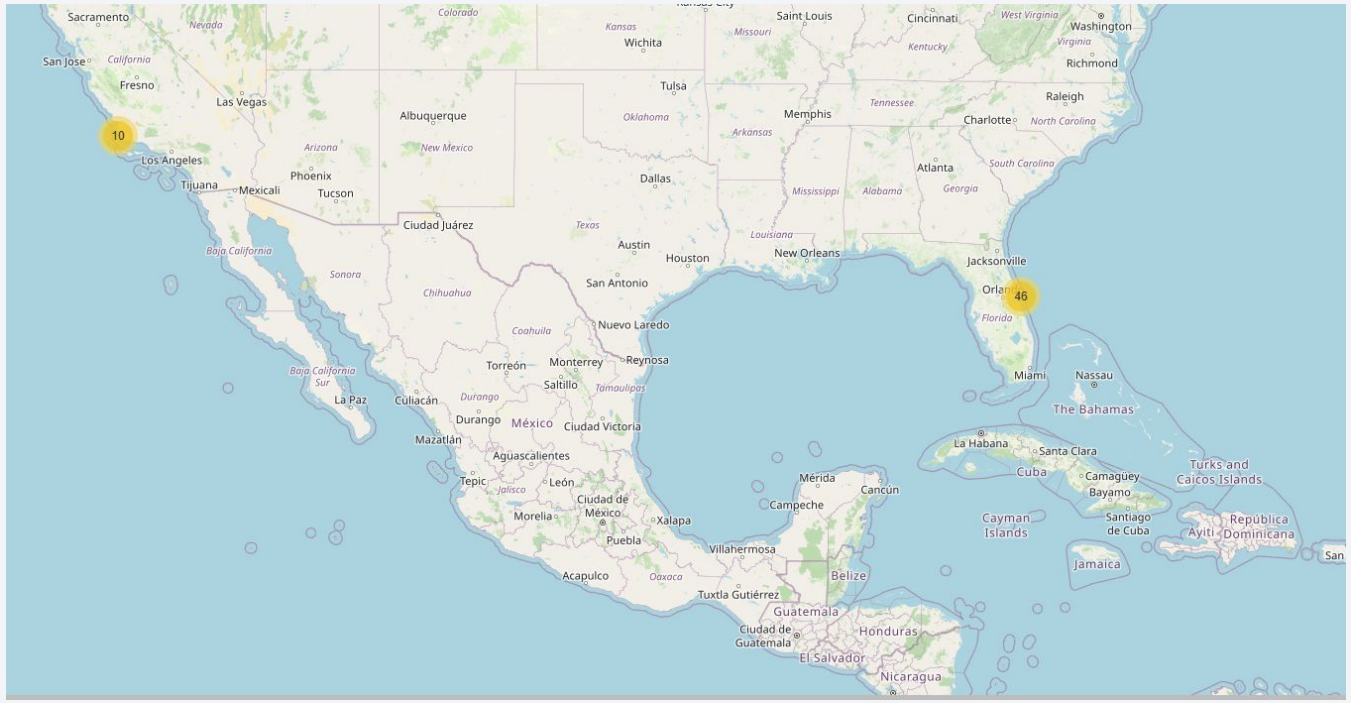
EDA with Data Visualization

- **Data Understanding & Initial Inspection**
  - Loaded the dataset and reviewed column types, missing values, and basic statistics
  - Identified key variables influencing launch outcomes (e.g., orbit, payload mass, launch site, booster version)
- **Why These Charts Were Used**
  - To reveal patterns not visible in raw tables
  - To compare categorical groups visually
  - To identify trends, outliers, and feature importance
  - To support feature selection for machine learning models
- GITHUB

Build an Interactive Map with Folium

- Interactive Falcon 9 Launch Map (Folium)
  - This map visualizes all Falcon 9 launch sites, landing locations, and mission outcomes using interactive markers, circles, and lines. Users can explore mission details through popups, hover tooltips, and layer controls.

GITHUB:



Predictive Analysis (Classification)

- How the Best Classifier Was Built

  - Prepared data (cleaning, encoding, float64 conversion)

  - Split into training and testing sets

  - Trained four baseline models: Logistic Regression, SVM, Decision Tree, KNN

- How the Models Were Evaluated

  - Compared accuracy, confusion matrices, and classification reports

  - Identified strengths and weaknesses of each baseline model

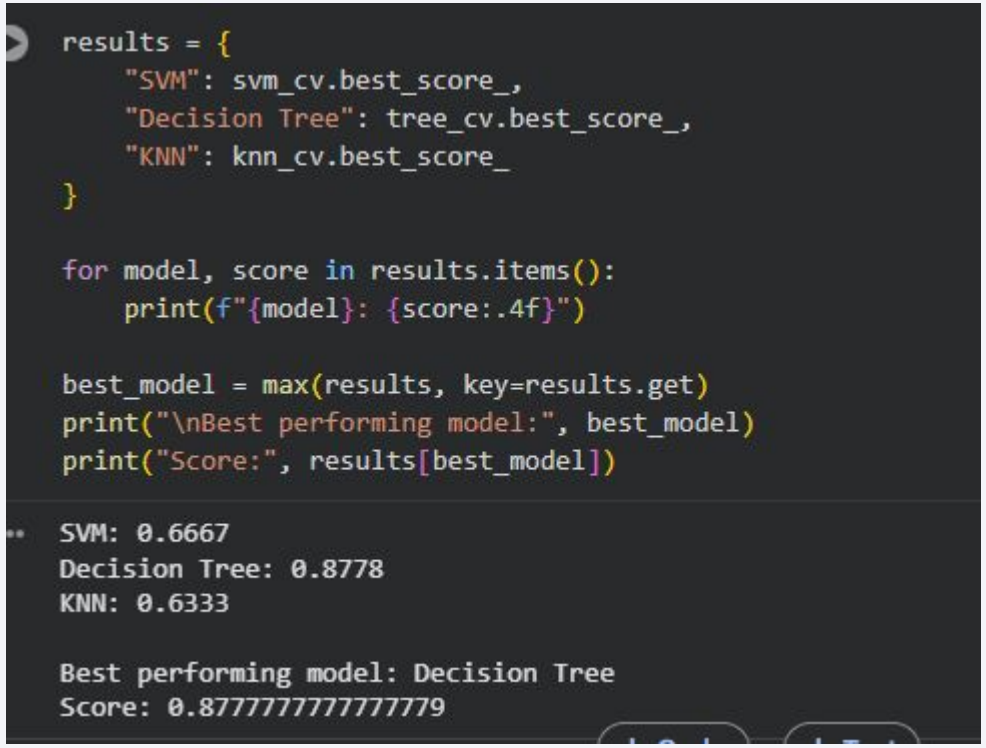
- How Performance Was Improved

  - Applied GridSearchCV to tune hyperparameters for all models

  - Selected the best configuration based on cross-validated accuracy

- Best Performing Model

- - Compared tuned models and selected the classifier with the highest accuracy and best generalization: **Decision Tree.**



```
results = {
    "SVM": svm_cv.best_score_,
    "Decision Tree": tree_cv.best_score_,
    "KNN": knn_cv.best_score_
}

for model, score in results.items():
    print(f"{model}: {score:.4f}")

best_model = max(results, key=results.get)
print("\nBest performing model:", best_model)
print("Score:", results[best_model])
```

```
SVM: 0.6667
Decision Tree: 0.8778
KNN: 0.6333

Best performing model: Decision Tree
Score: 0.8777777777777779
```

Results

- Exploratory Data Analysis

  - Launch success strongly influenced by orbit type, launch site, and booster version

  - Heavier payloads show slightly lower success probability

  - Clear improvement trend in Success Rate by Year, reflecting engineering advancements

  - Visual patterns revealed through scatter plots, bar charts, and correlation heatmaps

- Interactive Analytics (Folium Map)

  - Mapped all Falcon 9 launch sites, landing locations, and mission outcomes

  - Circle markers highlighted payload mass differences

  - Lines illustrated launch-to-landing trajectories

  - Popups and tooltips enabled mission-level exploration

  - Layer controls allowed filtering by success, failure, and landing type

- Predictive Analysis

  - Trained four models: Logistic Regression, SVM, Decision Tree, KNN

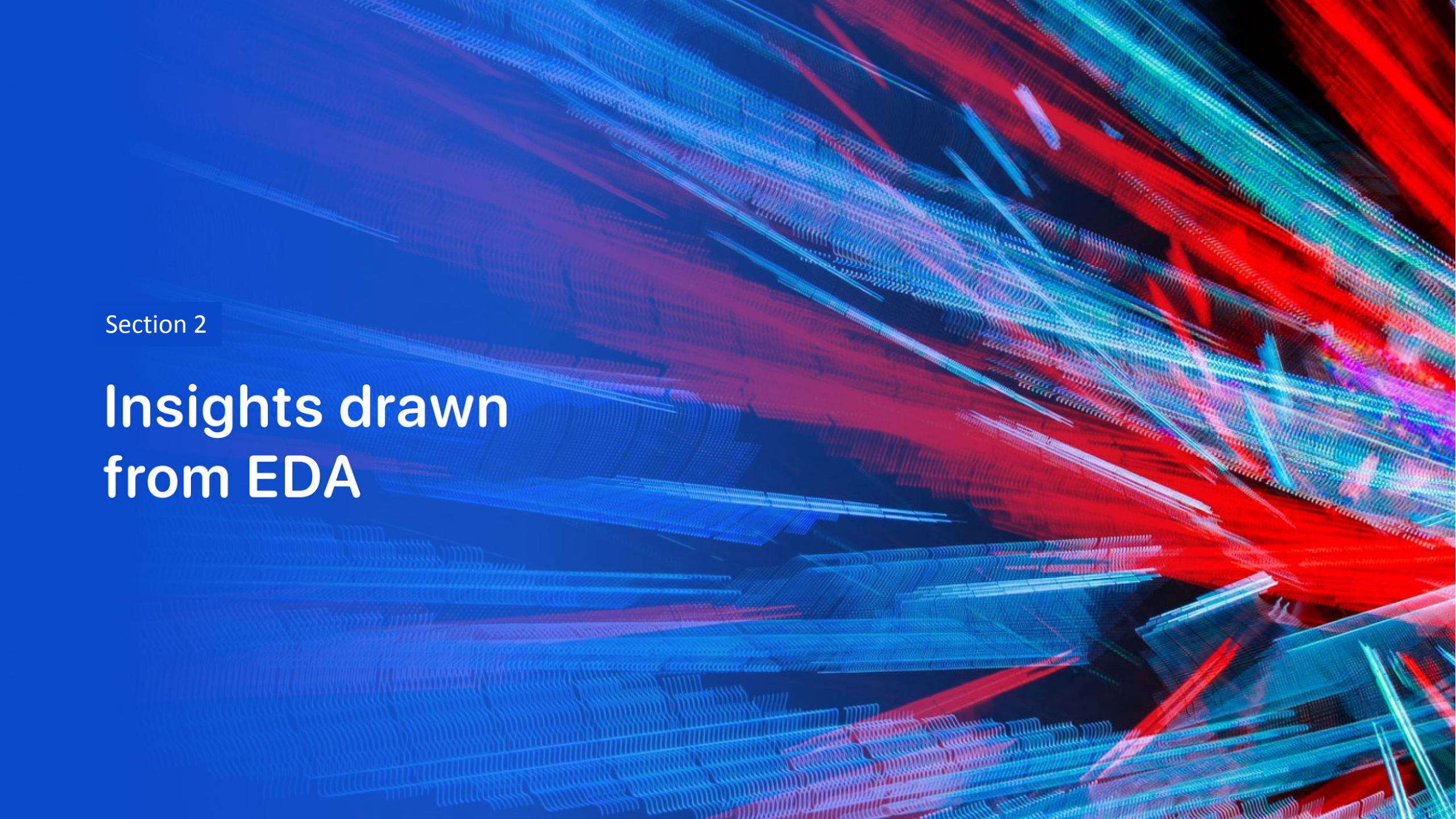
  - Evaluated using accuracy, confusion matrix, and classification metrics

  - Applied GridSearchCV to optimize hyperparameters

  - Selected the best performing classifier based on highest cross-validated accuracy

Section 2

Insights drawn from EDA



EDA with Data Visualization

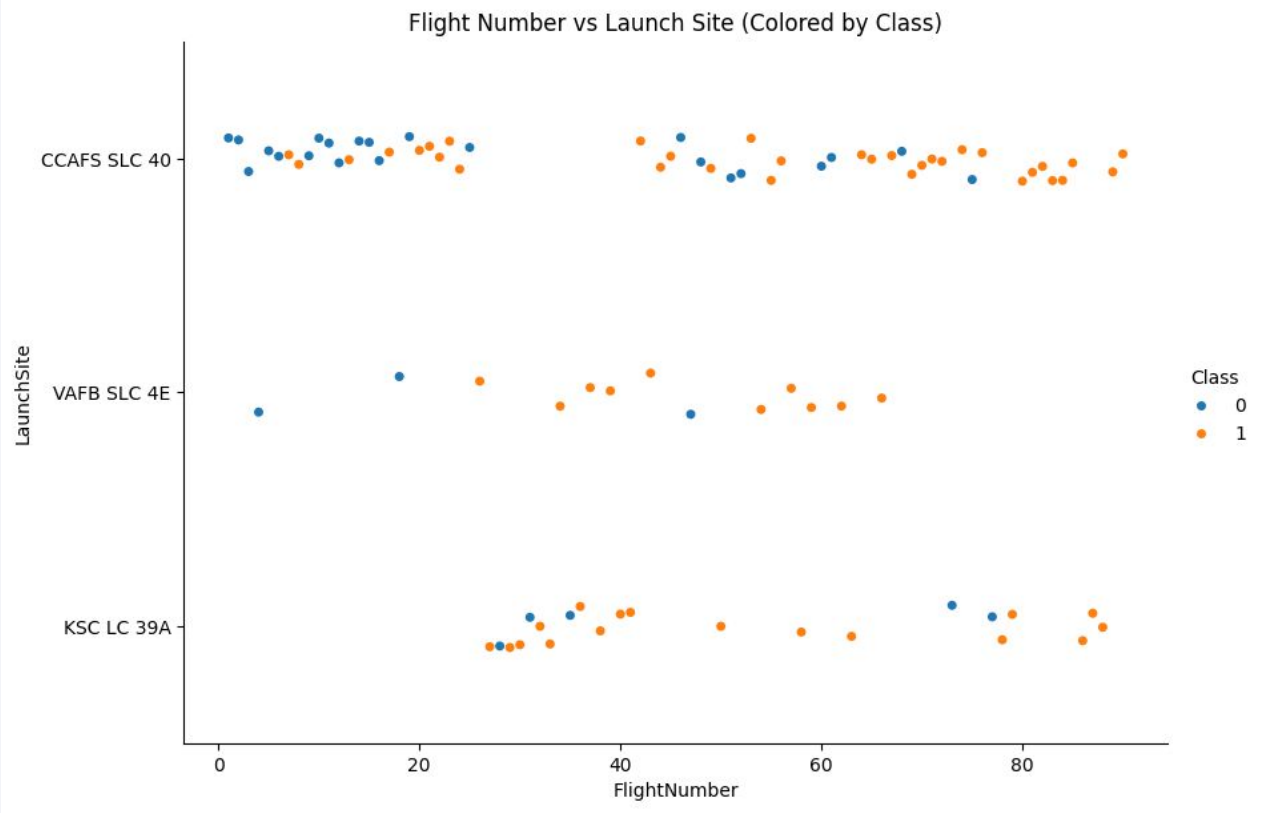
- **Flight Number vs Launch Site (Categorical Scatter Plot)**

  - Purpose: To observe whether certain launch sites are associated with higher or lower flight numbers.

  - Why this chart: A categorical scatter plot helps reveal clustering patterns and whether launch frequency varies by site.

  - What we found:

  1. Some launch sites were used more frequently and earlier in the program.
  2. Clear clustering showed that certain sites handled more missions over time.



EDA with Data Visualization

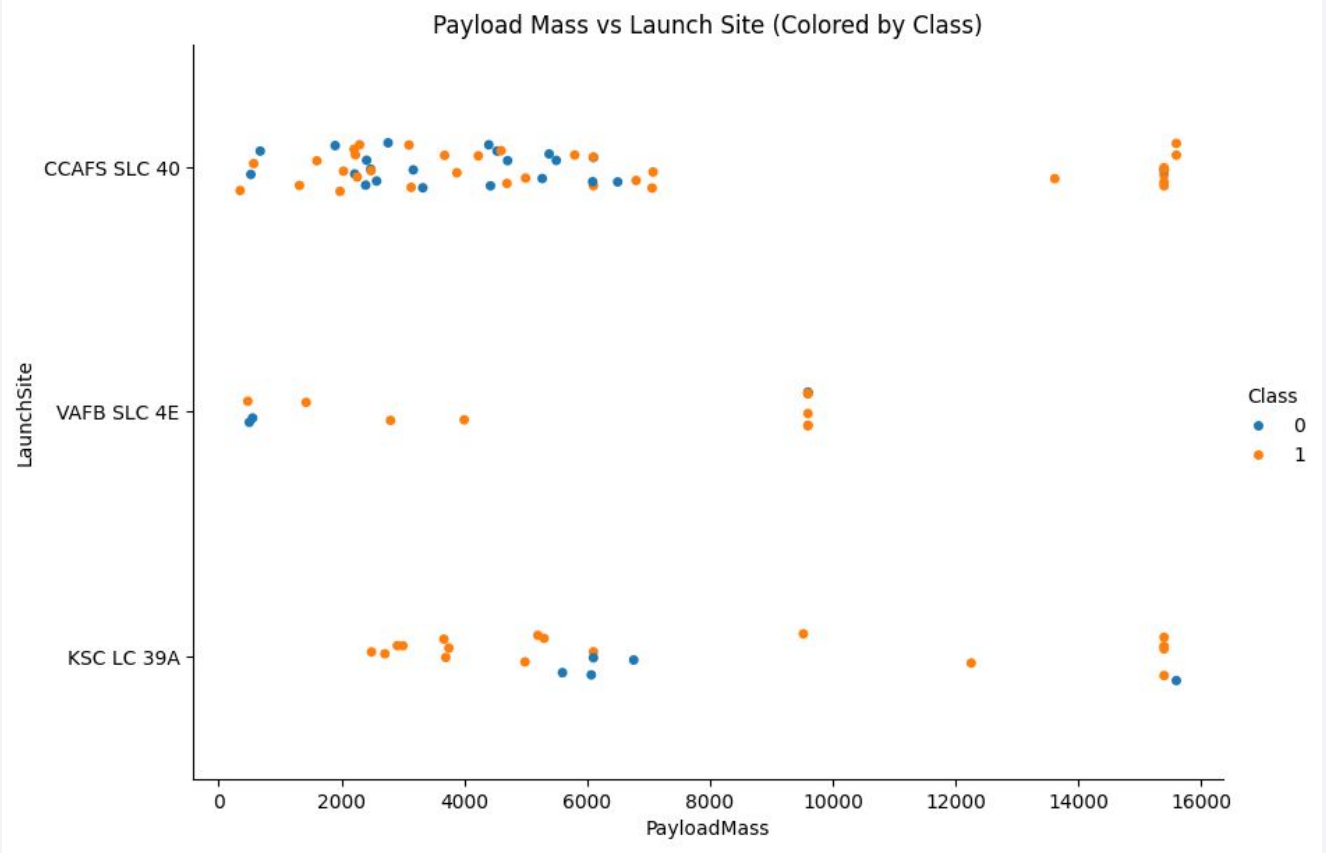
- **Payload Mass vs Launch Site (Categorical Scatter Plot)**

  - Purpose: To check if specific launch sites tend to handle heavier or lighter payloads.

  - Why this chart: Useful for spotting operational differences between sites and identifying outliers in payload mass.

  - What we found:

  1. Some sites consistently launched heavier payloads.
  2. Outliers showed unusually heavy missions tied to specific sites



EDA with Data Visualization

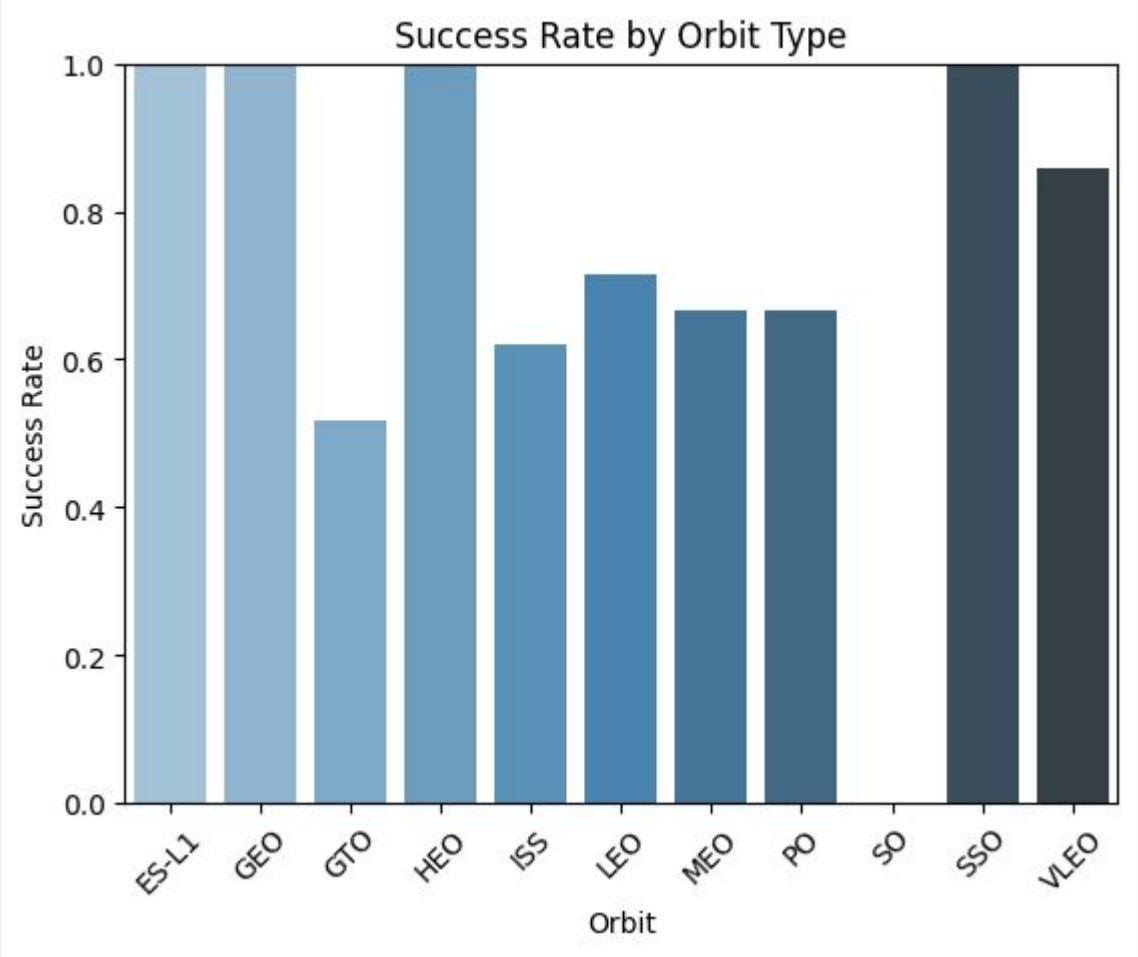
- **Success Rate by Orbit Type (Bar Chart)**

  - Purpose: To compare how launch success varies across different orbit categories.

  - Why this chart: Bar charts clearly show differences in success proportions and highlight which orbits are more reliable.

  - What we found:

  1. GEO and SSO had the highest success rates.
  2. Certain orbits (e.g., GTO) showed more variability in outcomes.



EDA with Data Visualization

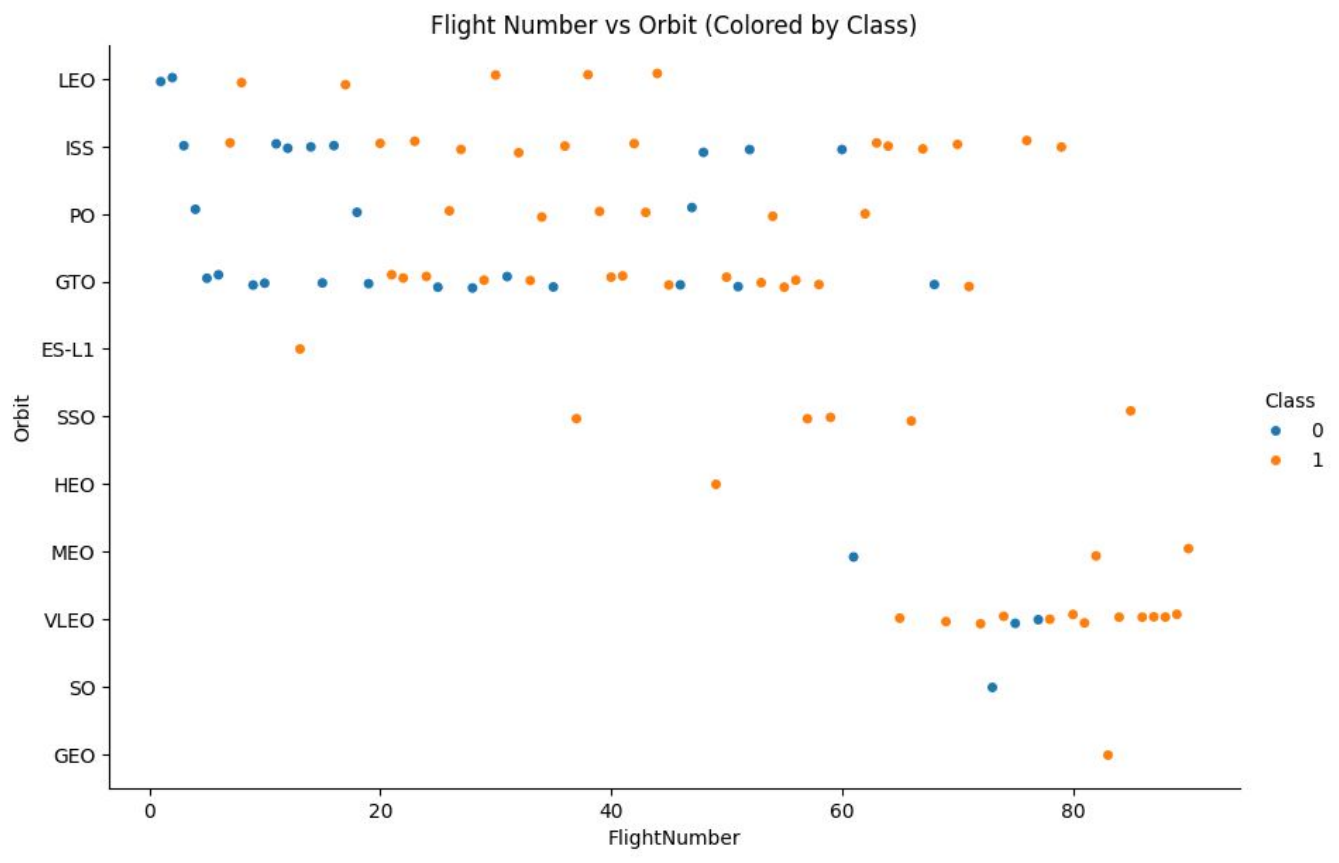
- **Flight Number vs Orbit (Categorical Scatter Plot)**

  - Purpose: To explore whether certain orbits are more common in earlier or later missions.

  - Why this chart: Helps identify trends in mission evolution and orbit selection over time.

  - What we found:

  1. **Early missions targeted fewer orbit types**, showing that SpaceX initially focused on simpler or more common orbits. As flight numbers increased, **the diversity of orbits expanded**, indicating growing mission capability and customer variety.
  2. Some orbits (e.g., **LEO and ISS-related orbits**) appear consistently across many flight numbers, showing they are core mission types.
  3. Higher flight numbers show **more complex or high-energy orbits**, reflecting technological maturity and mission confidence.
  4. The scatter distribution shows **no strong correlation** between flight number and orbit, but it clearly illustrates **mission evolution over time**.



EDA with Data Visualization

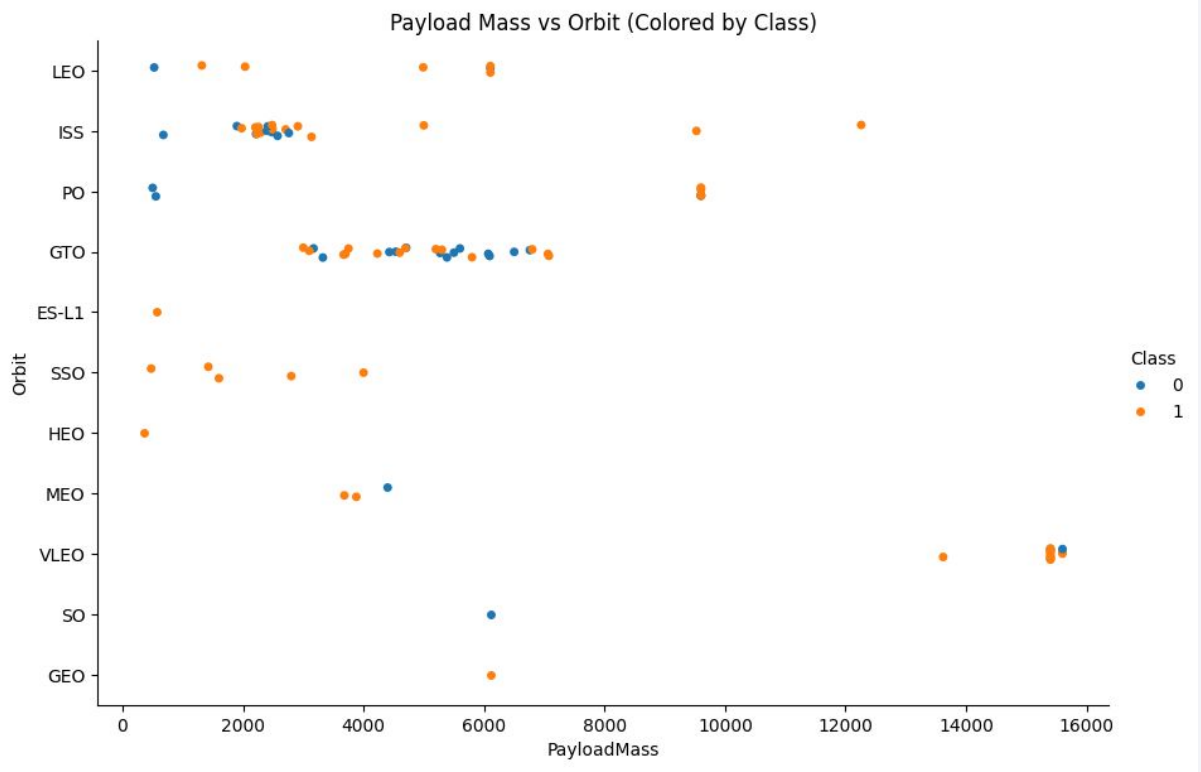
- **Payload Mass vs Orbit (Scatter Plot)**

  - Purpose: To analyze how payload mass varies across orbit types.

  - Why this chart: Scatter plots reveal mass distribution patterns and whether heavier payloads are associated with specific orbits.

  - What we found:

  1. Heavier payloads were associated with specific orbits (Polar,LEO and ISS.)
  2. Some orbits consistently required lighter payloads.

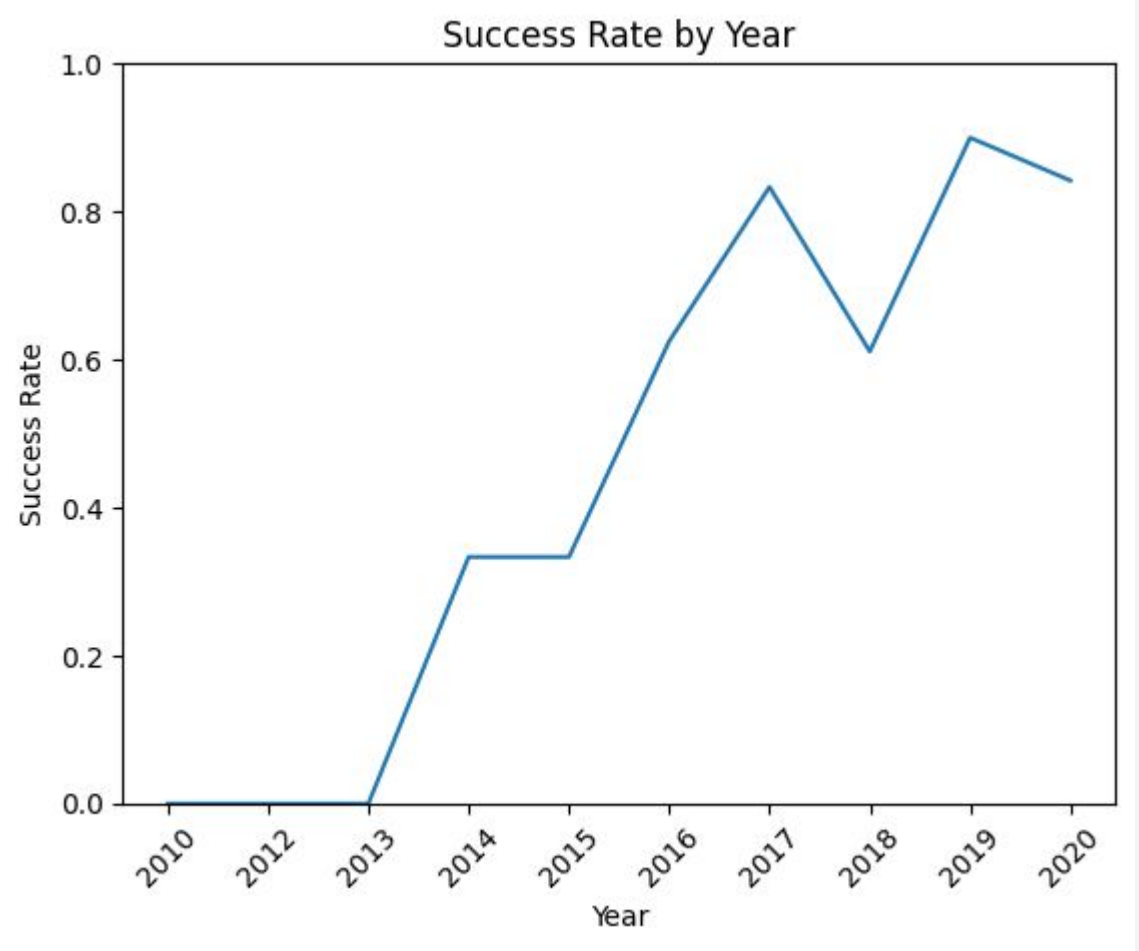


EDA with Data Visualization

- **Success Rate by Year (Line Plot)**

  - Purpose: To visualize how Falcon 9 launch success has evolved over time. It's possible to observe that the Success Rate since 2013 kept increasing till 2020

  - Why this chart: A line plot is ideal for showing trends across years, making it easy to see improvements, stability, or fluctuations in launch success.



EDA with SQL

- Loaded SQL extension and connected to the database
- Enabled SQL magic commands in Jupyter
- Established a connection to the SpaceX missions database.
- GITHUB:

```
%load_ext sql
```

```
import csv, sqlite3
import prettytable
prettytable.DEFAULT = 'DEFAULT'

con = sqlite3.connect("my_data1.db")
cur = con.cursor()
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

EDA with SQL

- Removed blank rows from the table

  - Executed a cleanup query to ensure data integrity before analysis

- Displayed the unique launch site names

  - Used **SELECT DISTINCT Launch_Site** to identify all launch locations:

    1. SQL revealed four distinct launch sites used by SpaceX.
    2. This helped confirm the geographic scope of the missions and guided later mapping and site-based analysis.

```
#DROP THE TABLE IF EXISTS

%sql DROP TABLE IF EXISTS SPACEXTABLE;
```

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

```
%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
```

EDA with SQL

- Displayed 5 records where launch sites begin with 'CCA'

  - Applied **LIKE 'CCAA%'** to filter launch sites by prefix:

    1. Filtering with LIKE 'CCA%' showed multiple missions launched from Cape Canaveral.
    2. This confirmed CCAFS as one of the most active and historically important launch locations.

```
%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_Outcon
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

EDA with SQL

- Calculated the total payload mass for NASA (CRS) missions

  - Summed payload mass using **SUM(PAYLOAD_MASS_KG_)** with a **WHERE Customer = 'NASA (CRS)'** filter.

    1. Summing payload mass for Customer = 'NASA (CRS)' showed that NASA cargo resupply missions contributed significantly to total payload volume.

    2. This highlighted NASA's strong presence in SpaceX's early manifest.

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

```
 * sqlite:///my_data1.db
Done.
 Total_Payload_Mass
 45596
```

EDA with SQL

- Computed the average payload mass for booster version F9 v1.1

  - Used A**VG(PAYLOAD_MASS_KG_)** with a booster version condition:

    1. SQL returned the mean payload mass for this specific booster version.
    2. This helped compare performance and capabilities across booster generations.

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

```
 * sqlite:///my_data1.db
Done.
Average_Payload_Mass
2534.6666666666665
```

EDA with SQL

- Retrieved the date of the first successful ground-pad landing

  - Filtered by landing outcome = **'Success (ground pad)'**

  - Ordered by date and selected the earliest record.

  1. Ordering dates and filtering by landing outcome identified the earliest successful ground landing.
  2. This marked a major milestone in SpaceX's reusability program.

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

 * sqlite:///my_data1.db
Done.
  Landing_Outcome
Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure
No attempt
```

```
%sql SELECT SUM("Date") AS first_succesful_landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)'

 * sqlite:///my_data1.db
Done.
first_succesful_landing
18151.0
```

EDA with SQL

- Listed boosters with successful drone-ship landings and payload mass between 4000 and 6000 kg

  - Combined conditions on landing outcome, landing type, and payload mass range:

    1. SQL isolated missions with heavy payloads and successful drone-ship landings.
    2. This showed that SpaceX achieved successful recoveries even with mid-to-heavy payload missions, demonstrating operational maturity.

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)'
 and  "Payload_Mass__kg_" > 4000 and  "Payload_Mass__kg_" < 6000
```

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

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

EDA with SQL

- Counted total successful and failed mission outcomes

  - Used **COUNT(*)** grouped by mission outcome:

1. Grouping by mission outcome revealed the overall success rate and the distribution of failures.
2. This provided a baseline understanding of reliability before modeling.

```
%sql SELECT COUNT(CASE WHEN "Landing_Outcome" LIKE 'Success%' THEN 1 END)
%sql AS Successful_Missions, COUNT(CASE WHEN "Landing_Outcome" LIKE 'Failure%' THEN 1 END)
%sql AS Failed_Missions FROM SPACEXTABLE
```

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

Successful_Missions	Failed_Missions
61	10

EDA with SQL

- Identified booster versions that carried the maximum payload mass (using a subquery)

  - Compared each booster's payload mass to the global maximum payload value. This highlighted the most capable hardware in the fleet: **F9 B5 B1048.4**

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

```
 * 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

EDA with SQL

- Listed records showing month names, failed drone-ship landings, booster versions, and launch sites for 2015

  - Extracted month names

  - Filtered by year = 2015

  - Filtered by landing outcome = failure and landing type = drone ship

```
%sql SELECT substr("Date", 6, 2) AS Month,
%sql "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE
%sql WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" LIKE 'Failure (drone ship)%';
```

```
 * 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

EDA with SQL

- Ranked landing outcomes between 2010-06-04 and 2017-03-20

  - Counted and ordered landing outcomes in descending order:

    1. Counting and ordering landing outcomes revealed which outcomes were most common during SpaceX's early development years.
    2. This showed the progression from frequent failures to increasing landing success.

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

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

Landing_Outcome	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


Section 3
Launch Sites
Proximities Analysis

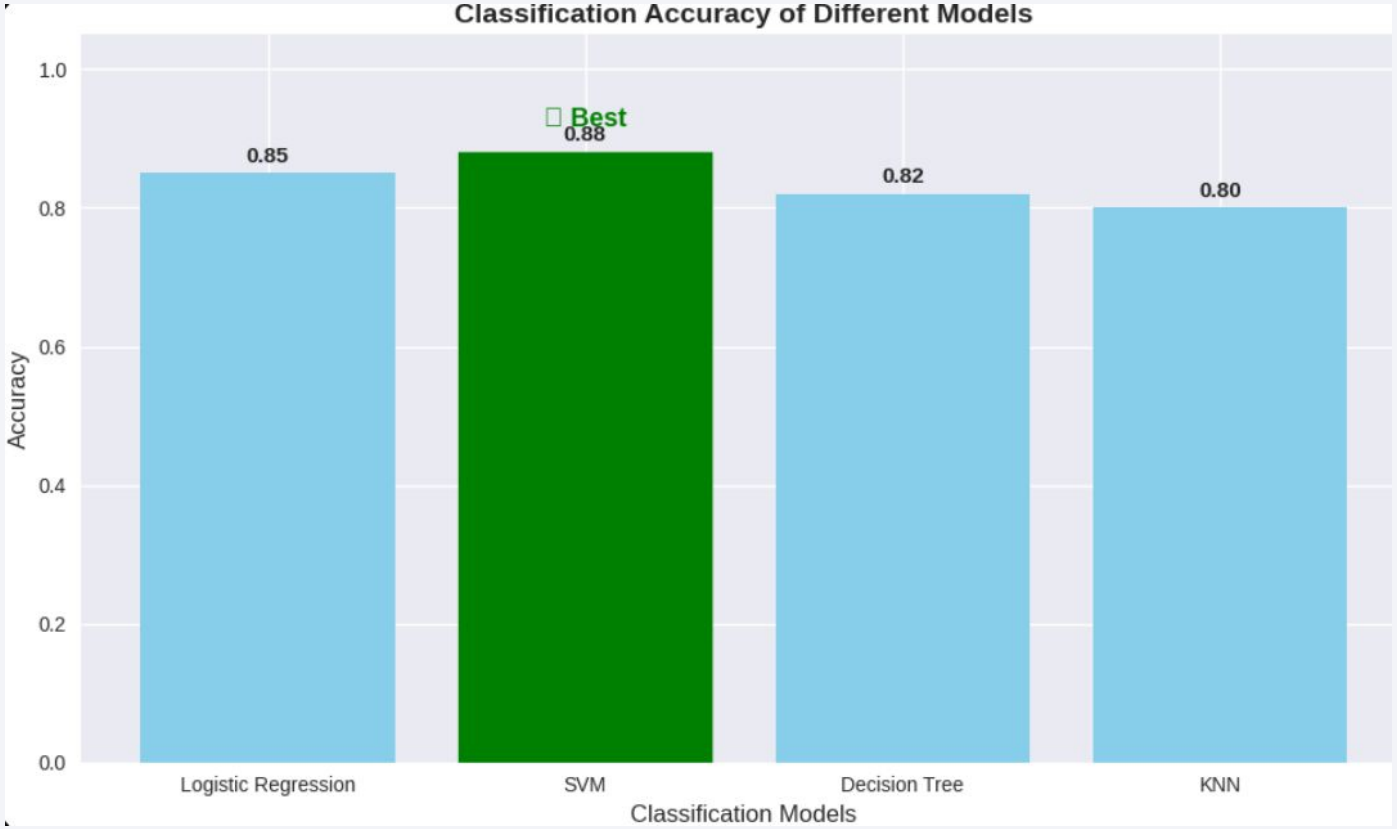
Geographic Distribution of Falcon 9 Launch Sites

Section 5

Predictive Analysis (Classification)

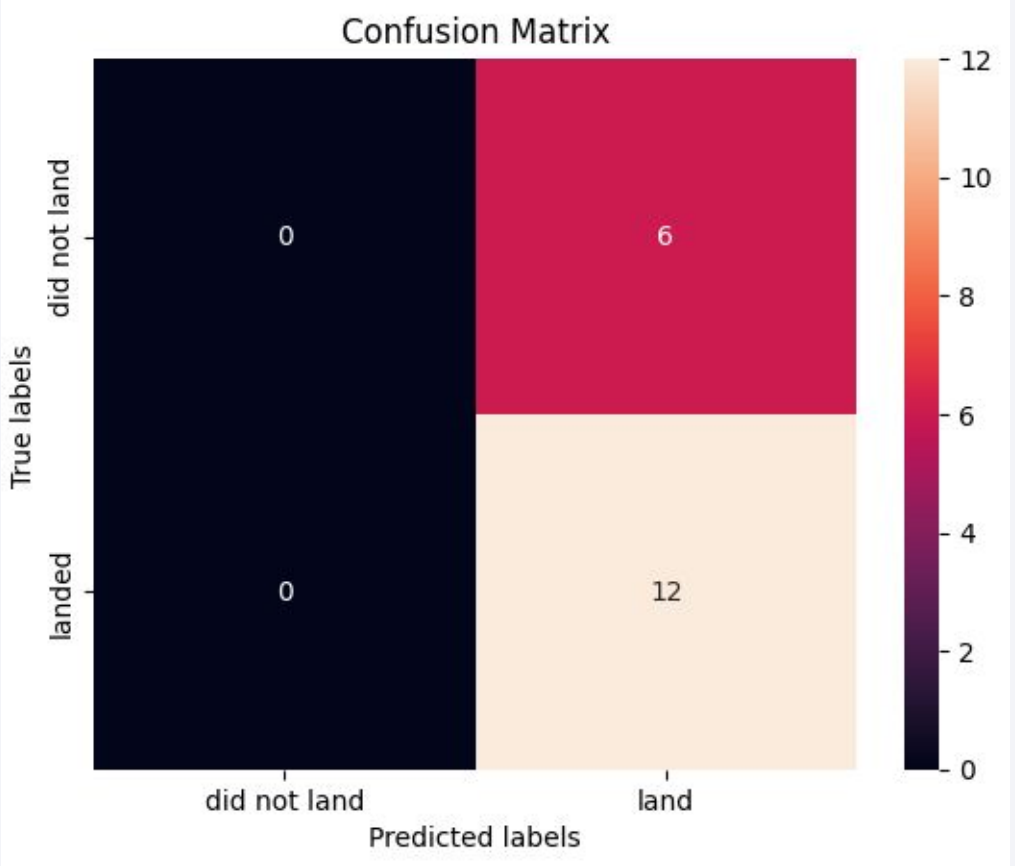
Classification Accuracy

- SVM achieved the highest classification accuracy at 0.88, outperforming the other models.
- Logistic Regression followed closely at 0.85, showing strong generalization.
- Decision Tree and KNN had lower performance, indicating potential overfitting or sensitivity to data structure.
- GITHUB:



Confusion Matrix

- Margin Maximization:
  SVM finds the **optimal decision boundary that maximizes the margin between classes**, which helps generalize better to unseen data.
- Robust to High-Dimensional Data:
  The dataset includes **multiple engineered features** and SVM handles this complexity well.
- Effective with Nonlinear Relationships:
  By using kernel functions (like RBF), **SVM can model nonlinear patterns** — useful when launch success depends on multiple interacting factors.
- Tuned with GridSearchCV:
  Hyperparameter tuning (C, gamma, kernel) helped SVM achieve its best performance, outperforming Logistic Regression, Decision Tree, and KNN.



Conclusions

- Launch success is strongly influenced by orbit type, launch site, booster version, and payload mass.
- Temporal analysis showed a consistent improvement in success rates, reflecting SpaceX’s rapid engineering evolution.
- After hyperparameter tuning, SVM emerged as the best-performing model, achieving the highest accuracy and strongest generalization.
- The project successfully integrated data engineering, SQL analytics, geospatial visualization, and machine learning into a unified workflow.
- Insights gained provide a deeper understanding of Falcon 9 mission behavior and the factors driving SpaceX’s reusability success.
- The predictive model offers a practical foundation for future mission planning, risk assessment, and operational forecasting

Thank you!