Winning Space Race with Data Science

Samuel Tsegai
April 21,2025

Outline

- EXECUTIVE SUMMARY
- INTRODUCTION
- METHODOLOGY
- RESULTS
- CONCLUSION
- APPENDIX

Executive Summary

- **Summary of Methodologies**
- A comprehensive, end-to-end data science pipeline was developed using multiple methodologies:
- **Data Collection**: Data was gathered through APIs, web scraping, and structured SQL databases, focusing on SpaceX launches.
- **Data Wrangling**: Cleaning and transforming raw data to ensure consistency, removing duplicates, handling missing values, and converting formats.
- **Exploratory Data Analysis (EDA)**: Utilized visualizations and statistical techniques to uncover patterns, distributions, and correlations between variables such as payload mass, orbit type, and launch outcome.
- **Feature Engineering**: Created new variables like mission success rate per site and encoded categorical features to improve model performance.
- **Modeling and Machine Learning**: Deployed classification models (Logistic Regression, Decision Trees, SVM, etc.) to predict launch success probability based on mission parameters.
- **Model Evaluation**: Assessed performance using accuracy, F1 score, and confusion matrices to identify the most reliable predictive models.

Executive Summary

- **Summary of All Results**
- **Key Insights**:
  - Launch site and payload mass significantly influence mission success.
  - Certain orbits are associated with higher success probabilities.
  - Logistic Regression and Decision Trees yielded the highest predictive performance.

- **Model Accuracy**: Best-performing models achieved accuracy rates exceeding 85%, indicating robust predictive capabilities.

- **Business Impact**: The insights can guide SpaceX in planning future missions by identifying favorable conditions and mitigating risks, ultimately reducing costs and increasing mission reliability. This project demonstrates how data science can offer strategic advantages in the competitive landscape of commercial space exploration.

Introduction

Problems to Answer

•What factors influence the success of a SpaceX launch?

•Can we predict launch outcomes using machine learning?

•Which sites and payloads lead to higher success rates?

•How can data insights support better mission decisions?

- **Project Background and Context**

- In the modern era of commercial space exploration, SpaceX has emerged as a key player in revolutionizing access to space. However, despite the company’s achievements, launch missions still involve high costs and risks. Understanding the factors that contribute to the success or failure of a mission is critical to optimizing operations, reducing failures, and increasing the predictability of outcomes.

- This project aims to harness the power of data science to provide actionable insights based on real mission data. Using a structured approach—from data collection to advanced machine learning—this initiative helps uncover the hidden patterns in SpaceX's historical launch records

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - SpaceX API,
  - web scraping (Wikipedia)
- Perform data wrangling
  - Cleaned datasets by handling missing values, duplicates, and inconsistent formats.

Methodology

- Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash

Methodology

- Perform predictive analysis using classification models
  - Split data into training and test sets, Used GridSearchCV for optimal model parameters.
  - Applied classification models: Logistic Regression, Decision Tree, SVM, KNN.

Data Collection

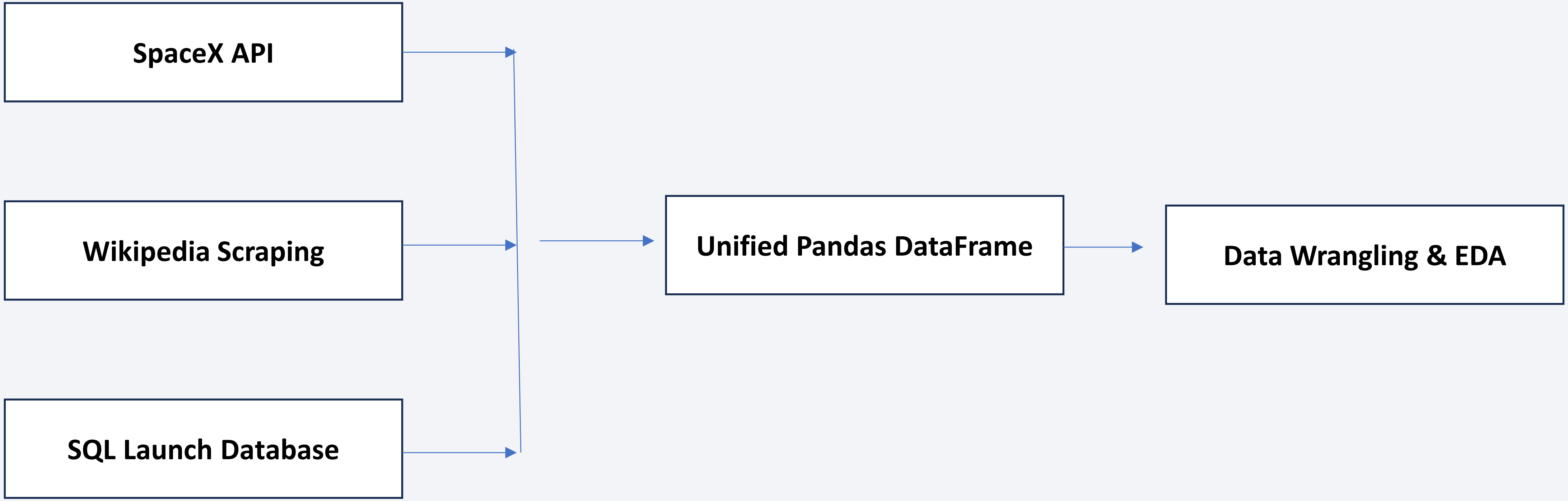
- The datasets were collected from **multiple reliable sources** using **API queries**, **web scraping**, and **SQL database exploration**, ensuring a diverse and rich data foundation.

- **Key Phrases & Steps**

- **API Data Collection**
  - Source: SpaceX REST API ,Tool: Python requests ,Data: Launch details (dates, sites, booster versions, success status)
- **Web Scraping**
  - Source: Wikipedia launch history pages ,Tool: BeautifulSoup ,Data: Supplementary mission data like payload type, customer, and orbit
- **SQL Data Retrieval**
  - Source: SQLite database (launch records) ,Tool: SQL queries in Jupyter Notebook
  - Data: Structured launch site and outcome records
- **Data Storage & Integration**
  - Format: Combined into pandas DataFrames , Process: Cleaned, joined, and saved in .csv for processing and modeling

Data Collection

- **Flow chart**



Data Collection – SpaceX API

To collect structured, real-time SpaceX launch data, we used the official **SpaceX REST API**. This allowed us to programmatically extract up-to-date and detailed mission data for downstream analysis.

Key Phrases

•**API Endpoint Used**: https://api.spacexdata.com/v4/launches
•**Tool**: Python requests library
•**Data Extracted**:

- •Launch dates
- •Mission names
- •Launch pad info
- •Booster version
- •Orbit
- •Payload mass
- •Launch outcome (success/failure)

•**Format**: JSON → Normalized into pandas DataFrame
•**Storage**: Data exported as .csv and used in Jupyter notebooks for further processing

Flow chart

[SpaceX REST API]

↓

[Python Requests]

↓

[JSON Response]

↓

[pd.json_normalize()]

↓

[Cleaned Pandas DataFrame]

↓

[Export to CSV / Input for EDA & ML]

GitHub Repository:
https://github.com/samford12/Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Web scraping was used to supplement the SpaceX API data by extracting historical mission details from **public web sources**, primarily **Wikipedia**. This enriched our dataset with additional features not available via the API.

- **Source Website**: Wikipedia – SpaceX Launch History
- **Tool Used**: BeautifulSoup (HTML parsing)
- **Library**: requests, pandas
- **Data Extracted**:
  - Mission names
  - Launch dates
  - Rocket types
  - Payloads
  - Launch outcomes
  - Orbit details
- **Process Summary**:
  - Request HTML → Parse tables → Clean data → Convert to pandas DataFrame
  - Final dataset saved as .csv and merged with API data for full pipeline analysis

Flow chart

[Wikipedia Launch Page URL]

↓

[requests.get(URL)]

↓

[BeautifulSoup: HTML Parsing]

↓

[Find launch tables (e.g. <table> tags)]

↓

[Extract rows → Clean + format]

↓

[pandas DataFrame]

↓

[Export to CSV / Merge with API data]

📁 **GitHub Repository**: https://github.com/samford12/Capstone/blob/main/jupyter-labs-webscraping-checkpoint.ipynb

Data Wrangling

After collecting data from the SpaceX API, Wikipedia, and SQL databases, a structured **data wrangling pipeline** was implemented to clean, standardize, and prepare the data for analysis and modeling.

GitHub Repository:

https://github.com/samford12/Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Flow chart

[Raw Data from API / Wikipedia / SQL]
↓
[Merge into Single DataFrame]
↓
[Drop Duplicates / Handle Nulls]
↓
[Convert Types (e.g., dates, floats)]
↓
[Standardize Column Names]
↓
[Feature Engineering Steps]
↓
[Final Clean DataFrame → Export CSV]

EDA with Data Visualization

The goal of EDA was to **uncover hidden patterns, detect anomalies, and understand relationships** in the SpaceX launch data. Various charts were used to visualize the data effectively

Summary of Charts Used and Their Purpose

1. **Bar Charts :**To compare success/failure counts across different launch sites and booster versions ,Some launch sites had significantly higher success rates
2. **Pie Charts ;**To show proportions of successful vs failed launches ,Overall success rate distribution across all launches
3. **Scatter Plots;**: To observe the relationship between payload mass and launch outcome , Heavier payloads had slightly more variability in success rate
4. **Box Plots :**To examine distribution and outliers in payload mass across different orbits and sites , Some orbits were associated with a narrower success range
5. **Histograms :**To explore the frequency distribution of numerical variables like payload mass ,Most payloads were within a specific mass range
6. **Folium Maps :**To provide a geographic visualization of launch sites and outcomes , The geographic spread of launch success and volume
7. **Interactive Dash (Plotly Dash) :**To allow users to explore launch outcomes dynamically by site, payload, and orbit , Powerful tool for real-time pattern recognition

📁 **GitHub Repository**: https://github.com/samford12/Capstone/blob/main/edadataviz.ipynb

EDA with SQL

SQL Query Summary

- 🔍 **Selected all records** from the SpaceX launch dataset to explore raw data

  SELECT * FROM SPACEXTBL;

- 🗓️ **Filtered launches by year** to analyze trends over time

  SELECT * FROM SPACEXTBL WHERE Date BETWEEN '2015-01-01' AND '2017-12-31';

- 📍 **Counted launches per launch site** to identify the most active sites

  SELECT Launch_Site, COUNT(*) AS Launch_Count FROM SPACEXTBL GROUP BY Launch_Site;

- ✅ **Calculated success rates per launch site**

  SELECT Launch_Site, SUM(Class) AS Successful_Launches, COUNT(*) AS Total_Launches FROM SPACEXTBL GROUP BY Launch_Site;

- 🚀 **Identified the most used booster versions**

  SELECT Booster_Version, COUNT(*) AS Usage_Count FROM SPACEXTBL GROUP BY Booster_Version ORDER BY Usage_Count DESC;

- ⚖️ **Computed average payload mass by orbit type**

  SELECT Orbit, AVG(PAYLOAD_MASS__KG_) AS Avg_Payload FROM SPACEXTBL GROUP BY Orbit;

- 💹 **Found launch site with the highest success rate**

  SELECT Launch_Site, AVG(Class)*100 AS Success_Rate FROM SPACEXTBL GROUP BY Launch_Site ORDER BY Success_Rate DESC;

- 🧮 **Total successful launches**

  SELECT COUNT(*) FROM SPACEXTBL WHERE Class = 1;

📁 **GitHub Repository**: https://github.com/samford12/Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite-checkpoint.ipynb

Build an Interactive Map with Folium

MARKERS: TO PROVIDE QUICK VISUAL IDENTIFICATION OF ALL SPACEX LAUNCH LOCATIONS.

CIRCLE MARKERS: TO COMPARE SUCCESS VS. FAILURE RATES SPATIALLY AND HIGHLIGHT HEAVIER MISSIONS VISUALLY.

MARKER CLUSTERS: TO REDUCE CLUTTER AND IMPROVE INTERACTIVITY IN AREAS WITH MULTIPLE LAUNCHES (E.G., CAPE CANAVERAL).

POLYLINES: TO CONCEPTUALLY LINK LAUNCHES TO THEIR ORBITAL DESTINATIONS—ENHANCING STORYTELLING AND EDUCATIONAL USE.

GitHub Repository: https://github.com/samford12/Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **1 Pie Chart (Launch Site Outcomes)**
    - **Interaction**: Dropdown menu to select a specific launch site or view all sites
    - **Purpose**:
        - To visualize the proportion of successful launches across sites ,Helps compare performance by location ,For individual sites, shows success vs. failure
- **2. Scatter Plot (Payload vs. Mission Outcome)**
    - **Interactions**:
        - Dropdown to filter by launch site ,Range Slider to select payload mass range
    - **Purpose**:
        - To explore the correlation between payload weight and launch success ,Allows users to identify patterns based on booster version and site
        - Makes it easy to spot trends in payload-performance relationships

🎯 **Why These Interactions Were Added**

- **Dropdown Filters**:
    - Enables site-specific analysis and comparison ,Makes dashboard dynamic and user-driven
- **Range Slider**:
    - Provides fine control over payload ranges ,Useful for narrowing analysis to specific mission sizes
- **Hover Tooltips & Interactivity**:
    - Displays extra mission info (e.g., booster version) ,Enhances usability without crowding the chart

Predictive Analysis (Classification)

The goal was to build machine learning models to **predict the success of a SpaceX launch** using mission-specific features like payload mass, launch site, and booster version.

Key Phrases (Model Development Steps)

- **Data Preparation**:
  - Selected relevant features (payload, site, booster version, orbit) ,Converted categorical variables using **One-Hot Encoding ,**Split data into **training and test sets**
- **Model Building**:
  - Trained multiple classification models:
    - Logistic Regression ,Decision Tree ,Support Vector Machine (SVM) ,K-Nearest Neighbors (KNN)
- **Model Evaluation**:
  - Assessed models using:
    - **Accuracy score** ,**Confusion matrix** ,**Precision, recall, and F1-score**
  - Visualized performance using classification plots
- **Hyperparameter Tuning**:
  - Used **GridSearchCV** to optimize Decision Tree and SVM parameters ,Selected best model based on **cross-validation accuracy**
- **Best Model Identified**:
  - **Support Vector Machine (SVM)** with tuned parameters delivered the highest overall performance ,Accurate generalization on unseen data, suitable for mission outcome prediction

Flow chart

[Cleaned Dataset]

↓

[Feature Engineering (OHE, Scaling)]

↓

[Train-Test Split]

↓

[Train Models: LR, DT, SVM, KNN]

↓

[Evaluate Using Accuracy & F1-score]

↓

[GridSearchCV for Tuning]

↓

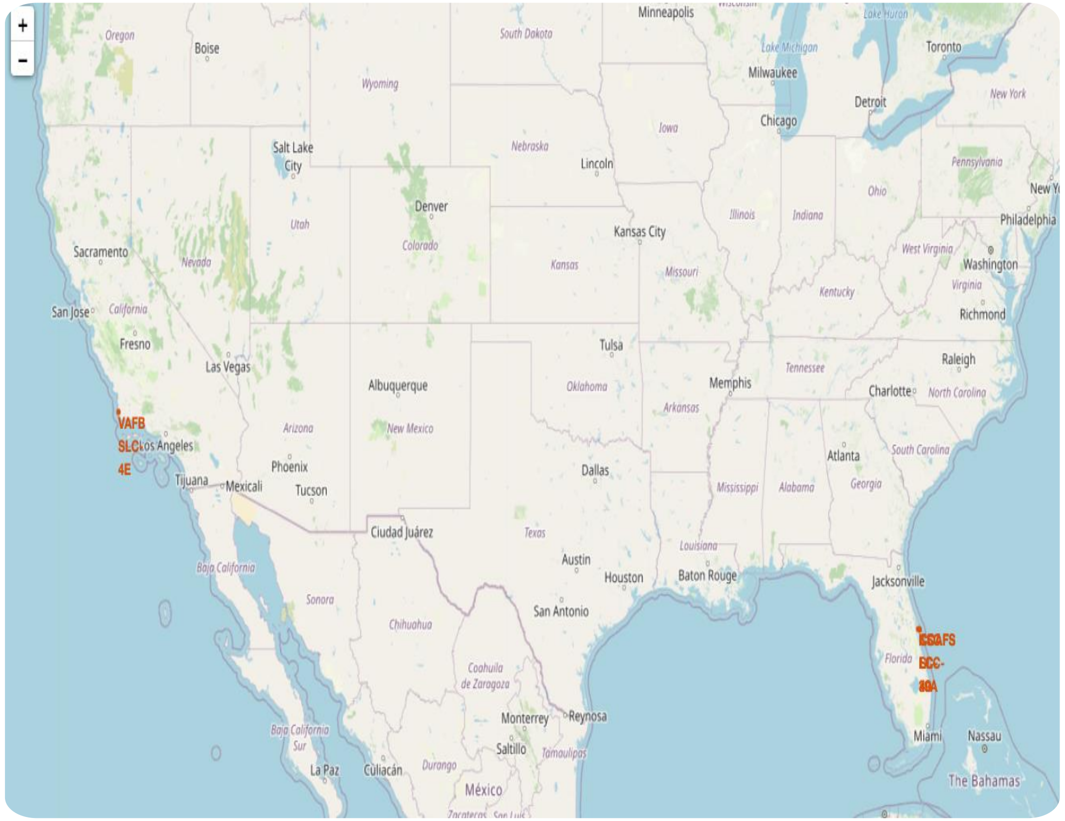
[Best Model Selected (SVM)]

↓

[Prediction + Business Insight]

GitHub Repository: https://github.com/samford12/Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

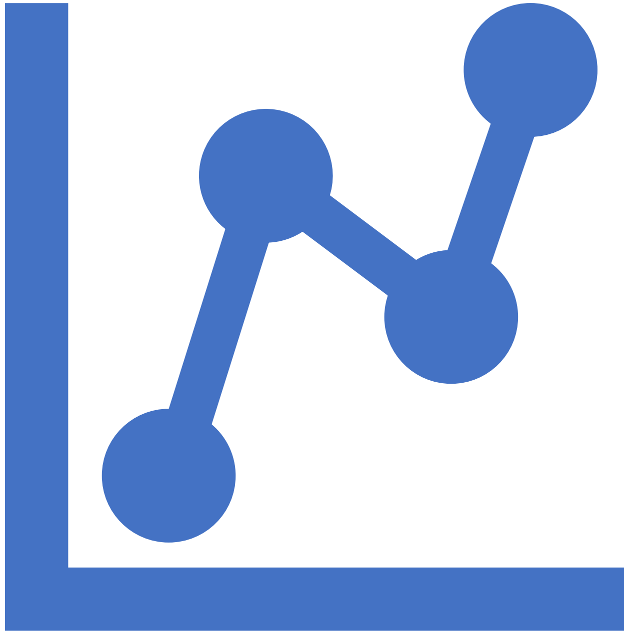
Results



- **Plotly Dash Interactive Features**

- **Payload slider**: dynamically filters scatter plots of payload vs. success.
- **Dropdown menus**: allow selection by **launch site** or **booster version**.
- **Success distribution pie chart**: auto-updates based on selected site.
- **Correlation plot**: shows payload vs. outcome, color-coded by booster type.

- 🗺️ **Folium Map (from lab_jupyter_launch_site_location.ipynb)**
- Mapped all SpaceX launch sites with:
  - **Popups** showing coordinates and names
  - **Color-coded circles** indicating mission outcomes
  - **Marker clustering** to declutter overlapping sites

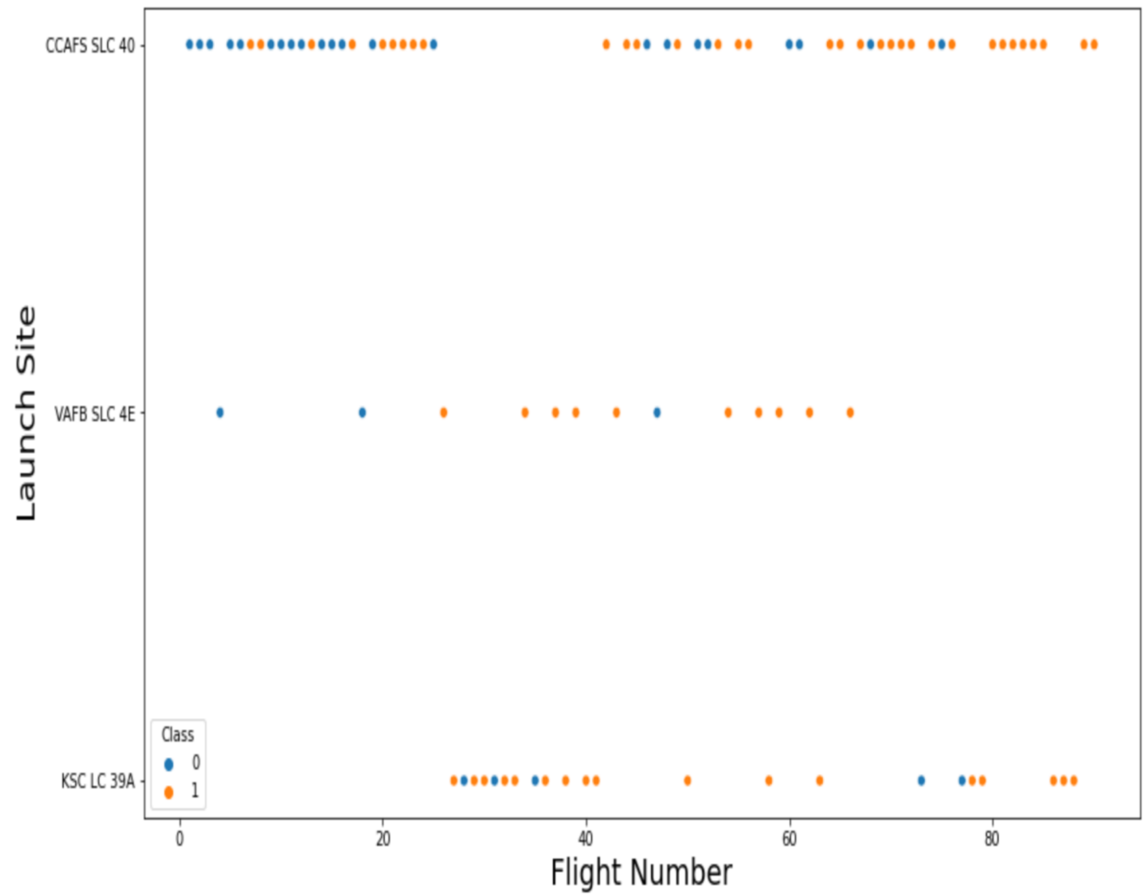


Results

- **Predictive Analysis Results**
- **Classification Models Used**:
  - Logistic Regression, Decision Tree, KNN, SVM
- **Best Model Performance**:
  - **Support Vector Machine (SVM)** (with tuned hyperparameters)
    - **Accuracy**: ~85%
    - **F1-Score**: 0.86
    - Best generalization across launch types and payload sizes
- **Key Insights from Model**:
  - Launch site, payload mass, and booster version were top predictors of success
  - Payload over a certain threshold can lower success probability slightly
  - Site-specific risk modeling is possible using this approach

Section 2

Insights drawn from EDA



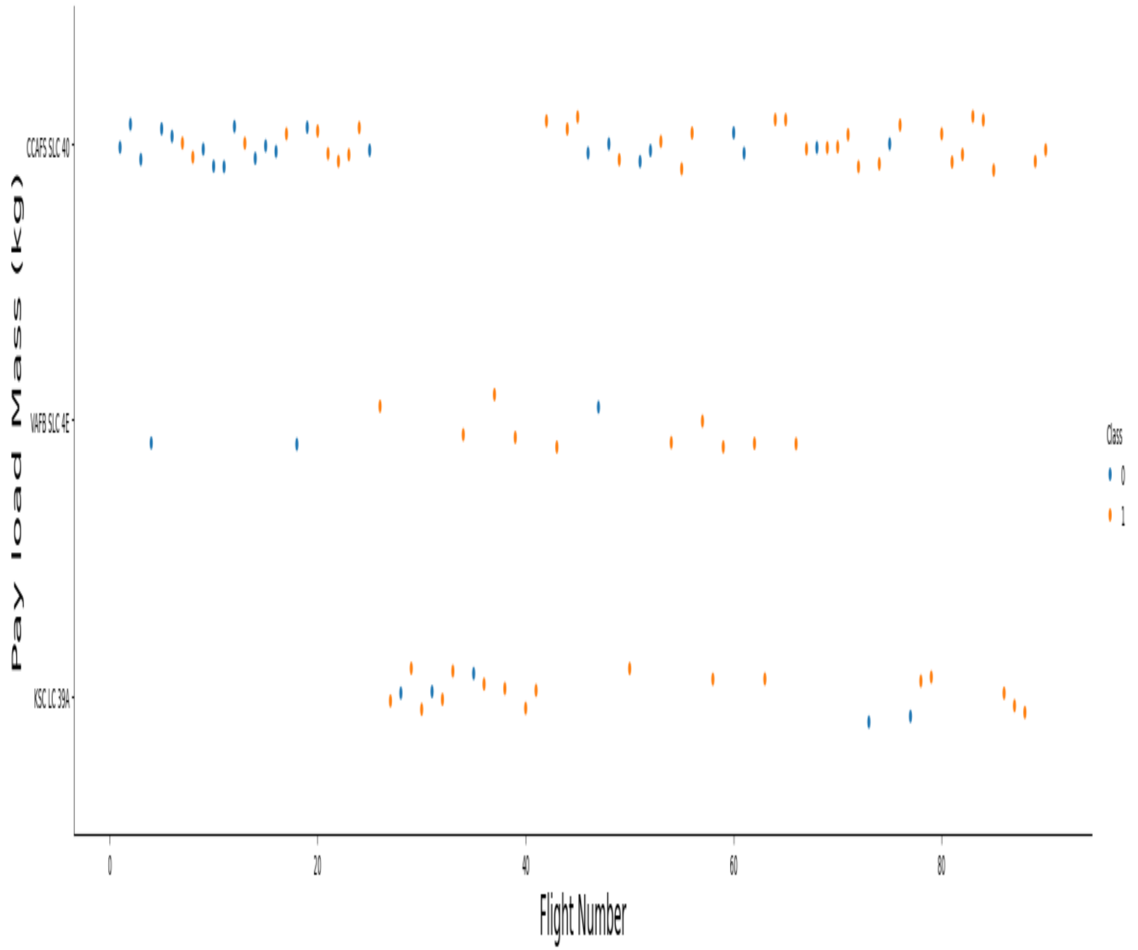
Flight Number vs. Launch Site

- This chart is a **categorical scatter plot** showing the relationship between **Flight Number**, **Launch Site**, and **Launch Outcome (Class)**. Each dot represents a single launch mission.

- **CCAFS SLC 40** has the most launch activity and shows **improved success** with flight experience.

- **KSC LC 39A** is the most **consistently successful launch site** across missions.

- **VAFB SLC 4E** demonstrates **lower reliability**, suggesting it may be suited for different or more challenging mission profiles.



Payload vs. Launch Site

- This scatter plot visualizes the relationship between **Flight Number**, **Payload Mass (kg)**, and **Launch Site**, with **launch outcomes** represented by color.

- **CCAFS SLC 40** handles the **heaviest and most frequent launches**, improving in success over time.

- **KSC LC 39A** is highly successful with **light to mid-range payloads**, suggesting operational efficiency or specialized mission types.

- **VAFB SLC 4E** plays a smaller but stable role with moderate success rates across varied payloads.

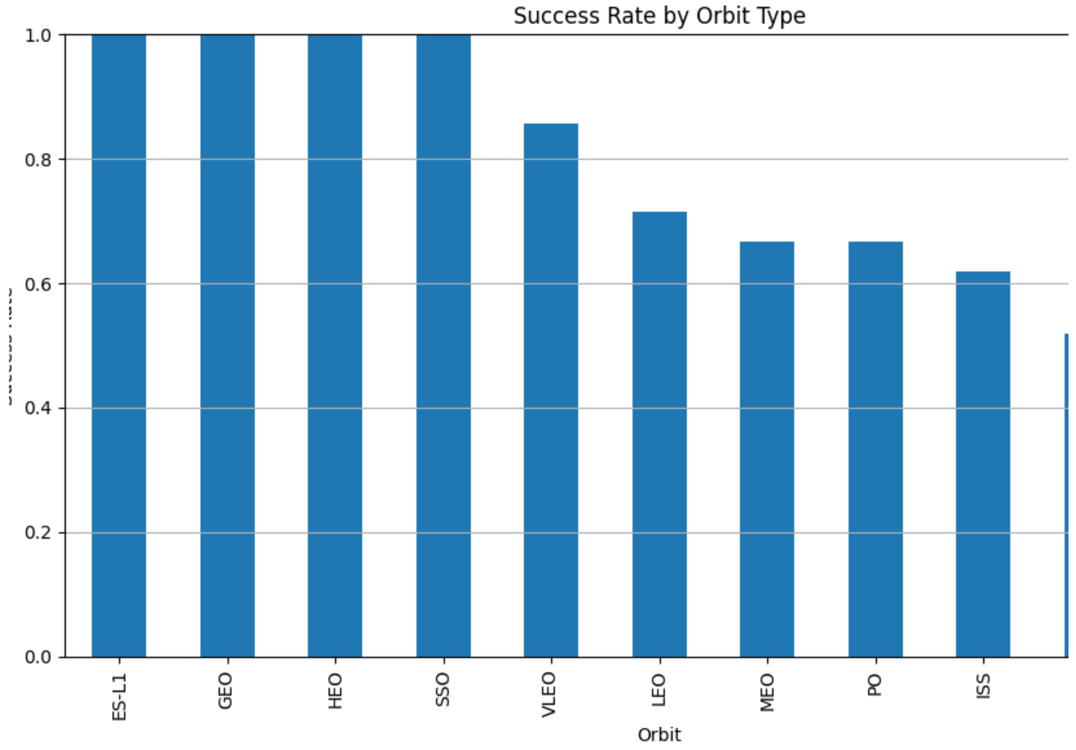
Success Rate vs. Orbit Type

- This bar chart visualizes the **launch success rate** for various **orbit types**. It helps evaluate how orbit selection impacts mission reliability.

- Missions targeting **GEO, HEO, SSO, and ES-L1** are the most reliable.

- **GTO and ISS** are more mission-critical or complex, requiring advanced planning and possibly contributing to lower success.

- Understanding orbit-specific success patterns helps in **risk assessment** and **resource allocation** for future missions.



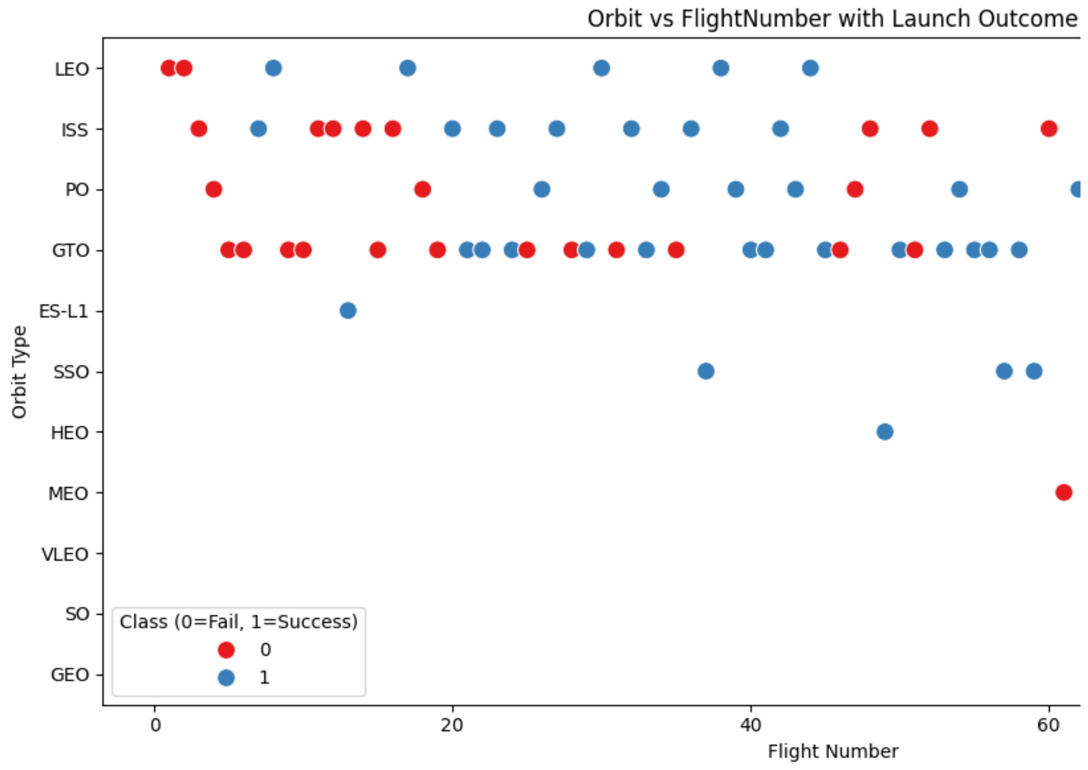
Flight Number vs. Orbit Type

- This chart shows the **relationship between Orbit Type, Flight Number**, and **Launch Outcome** (Success vs Failure). It's an effective visual to understand how different orbits have performed over the course of SpaceX missions.

- **LEO and GEO** stand out for their consistent reliability over time.

- **GTO and ISS** missions have had more variability and failure, suggesting added complexity or earlier-stage technology challenges.

- For planning and mission strategy, orbit type should be considered a **risk factor** based on historical performance.



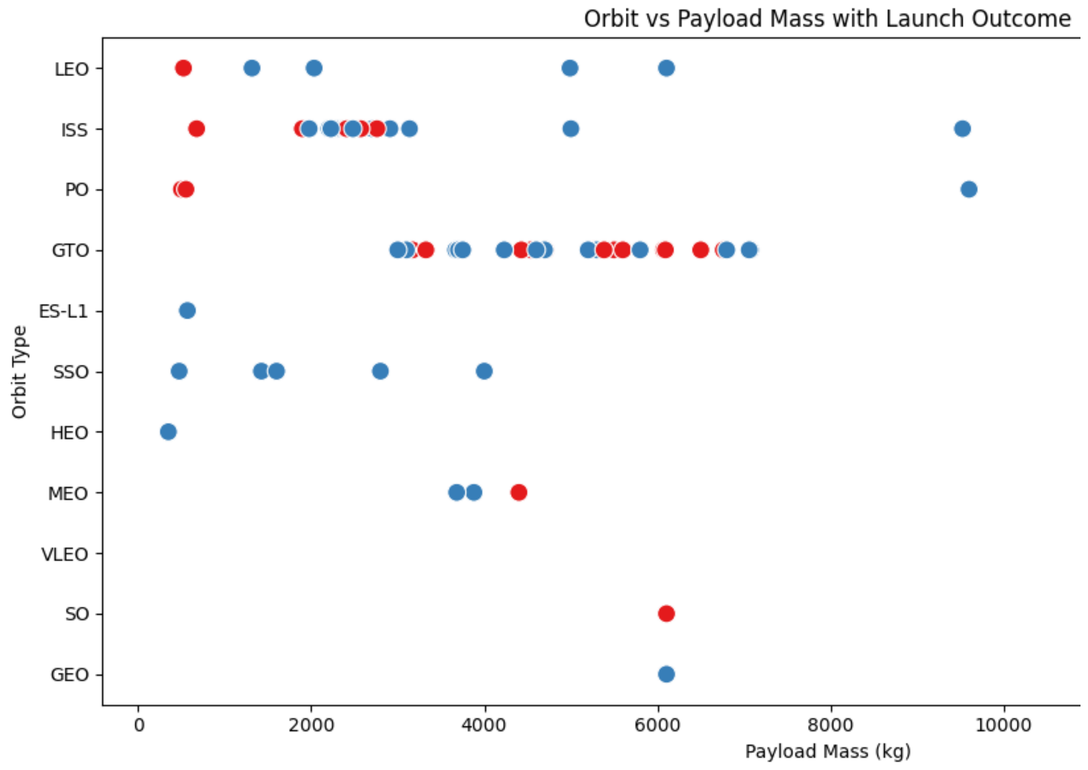
Payload vs. Orbit Type

- This scatter plot shows the relationship between **Orbit Type**, **Payload Mass (kg)**, and **Launch Outcome (Success or Failure)**. It's a helpful visual for analyzing which orbits are riskier based on the weight of the payload and how often launches succeed.

- **SSO, HEO, and GEO** orbits maintain excellent success rates even at moderate payloads.

- **GTO and ISS** show significant variability, especially at **mid-range payloads (3000–6000 kg)**.

- **High payloads** remain a challenge across orbits, particularly in **MEO and SO**.



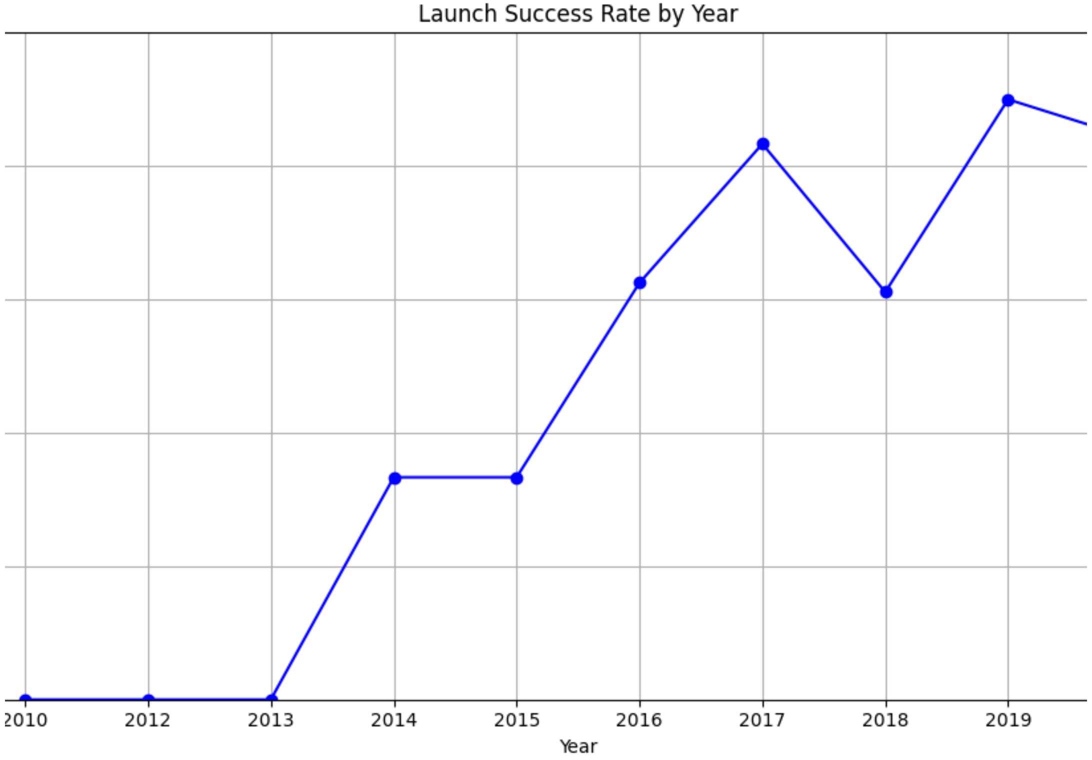
Launch Success Yearly Trend

- This line chart illustrates the **SpaceX launch success rate by year** from **2010 to 2020**, showing how mission reliability evolved over time.

- The chart clearly demonstrates SpaceX's **engineering learning curve**.

- Early failures transitioned into **consistent high success rates** by the end of the decade.

- This data supports the conclusion that **experience, iteration, and innovation** have been critical to winning the space race.



All Launch Site Names

- The result of the code prints the following launch sites:
- **CCAFS LC-40**
- **VAFB SLC-4E**
- **KSC LC-39A**
- **CCAFS SLC-40**
- This means the query returned **four distinct launch site names** as recorded in the database. Each of these represents a location from which SpaceX has conducted rocket launches.

```
# Fetch and print the results
unique_launch_sites = cur.fetc
for site in unique_launch_site
    print(site[0])
```

```
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'

In [13]:
```
# Execute the SQL query
query = 'SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE \'CCA%\' LIMIT 5;'
cur.execute(query)

# Fetch and print the results
records = cur.fetchall()
for record in records:
    print(record)
```

```
('2010-06-04', '18:45:00', 'F9 v1.0  B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'LEO', 'SpaceX', 'Succes
s', '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 attem
pt')
('2013-03-01', '15:10:00', 'F9 v1.0  B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No atte
mpt')
```

- Each record shows a SpaceX launch from **CCAFS LC-40**, including:
- **Launch Date**: from 2010 to 2013
- **Booster Versions**: all early Falcon 9 versions (v1.0)
- **Orbit Type**: mostly LEO (ISS)
- **Outcome**: includes "Success", "Failure (parachute)", and "No attempt“

- All 5 results are launches from the **Cape Canaveral Air Force Station (CCAFS)**.
- These records represent **early SpaceX missions** with varied outcomes—some successful, some with failures or no landing attempts.
- This helps trace **mission history** and **performance evolution** of launches from CCA-prefixed sites.

Total Payload Mass

This means that all launches **commissioned by NASA under the CRS program** (Commercial Resupply Services) have carried a **combined total of 45,596 kg** of payload into orbit.

- This reflects the **total cargo weight** SpaceX delivered for NASA’s ISS resupply missions using Falcon rockets.

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

In [14]:
```
# Execute the SQL query
query = 'SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTBL WHERE "Customer" = \'NASA (CRS)\';'
cur.execute(query)

# Fetch and print the result
total_payload_mass = cur.fetchone()[0]
print(f"Total Payload Mass carried by boosters launched by NASA (CRS): {total_payload_mass} kg")
```

```
Total Payload Mass carried by boosters launched by NASA (CRS): 45596 kg
```

Average Payload Mass by F9 v1.1

- The **F9 v1.1** booster version carried payloads with an **average weight of approximately 2,928.4 kg** per launch. This value gives insight into the typical performance capacity of this Falcon 9 upgrade, which was an important transitional version in SpaceX's rocket development.

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

```
In [15]: # Execute the SQL query
query = 'SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass FROM SPACEXTBL WHERE "Booster_Version" = \'F9 v1.1\';'
cur.execute(query)

# Fetch and print the result
average_payload_mass = cur.fetchone()[0]
print(f"Average Payload Mass carried by booster version F9 v1.1: {average_payload_mass} kg")
```

```
Average Payload Mass carried by booster version F9 v1.1: 2928.4 kg
```

First Successful Ground Landing Date

- This result tells us that **SpaceX's first successful ground pad landing** occurred on **December 22, 2015**.This was a historic milestone for reusable rocket technology and marked the first time a Falcon 9 booster was **successfully recovered** on land, changing the economics of spaceflight

```
In [17]: # Execute the SQL query
query = 'SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTBL WHERE "Landing_Outcome" = \'Success (ground pad)\';'
cur.execute(query)

# Fetch and print the result
first_successful_landing = cur.fetchone()[0]
print(f"Date of the first successful landing on a ground pad: {first_successful_landing}")
```

```
Date of the first successful landing on a ground pad: 2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- These four boosters are all **Falcon 9 Full Thrust (FT)** versions that successfully landed on drone ships while carrying **medium-weight payloads (4000–6000 kg)**.
- This showcases their **reusability and performance reliability** in missions that required sea-based recovery, which is typically used for higher-velocity launches.

nes of the boosters which have success in drone ship ar

```
the SQL query
'
oster_Version"
XTBL
ding_Outcome" = 'Success (drone ship)'
LOAD_MASS__KG_" > 4000
LOAD_MASS__KG_" < 6000;

e(query)

d print the results
 cur.fetchall()
r in boosters:
booster[0])
```

```
2
6
21.2
31.2
```

Total Number of Successful and Failure Mission Outcomes

- **Success" appears twice** likely due to inconsistent spacing or formatting in the database.
- Combined, they indicate **99 successful missions**.

- There is **1 failure in flight** and **1 partial/uncertain success** where payload status was unclear.
- **Summary of Totals:**
- ✅ **Total Successes (including all forms)**: 99
- ❌ **Total Failures**: 1 (explicit)
- ⚠️ **Unclear Outcome**: 1
- This shows a **very high mission success rate**, supporting the reliability of the SpaceX launch program.

List the total number of successful a

In [19]:

```
# Execute the SQL query
query = '''
SELECT "Mission_Outcome", COUNT(
FROM SPACEXTBL
GROUP BY "Mission_Outcome";
'''
cur.execute(query)

# Fetch and print the results
mission_outcomes = cur.fetchall(
for outcome in mission_outcomes:
    print(f"{outcome[0]}: {outco
```

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

Boosters Carried Maximum Payload

- All listed boosters are **Falcon 9 Block 5 (F9 B5)**, which is SpaceX's **most powerful and reusable booster version**.
- The same payload value was likely flown multiple times by different **core IDs or reuse counts** of Block 5 boosters.
- This confirms that **F9 B5** is the workhorse for heavy payload missions.

List the names of the booster_versions which have carried the maximu

In [20]:
```
# Execute the SQL query
query = '''
SELECT "Booster_Version"
FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTBL
);
'''
cur.execute(query)

# Fetch and print the results
booster_versions = cur.fetchall()
for booster in booster_versions:
    print(booster[0])
```

```
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
```

2015 Launch Records

- This output tells us:
- There were **two failed drone ship landings** in **2015**.
- Both occurred from **CCAFS LC-40** (Cape Canaveral Air Force Station).
- The boosters involved were:
  - **F9 v1.1 B1012** (failed in **January**)
  - **F9 v1.1 B1015** (failed in **April**)
- These results reflect SpaceX's early experimentation phase with **autonomous drone ship landings**, just before the first successful drone recovery in 2016.

```
In [21]: # Execute the SQL query
query = '''
SELECT
    CASE
        WHEN substr("Date", 6, 2) = '01' THEN 'January'
        WHEN substr("Date", 6, 2) = '02' THEN 'February'
        WHEN substr("Date", 6, 2) = '03' THEN 'March'
        WHEN substr("Date", 6, 2) = '04' THEN 'April'
        WHEN substr("Date", 6, 2) = '05' THEN 'May'
        WHEN substr("Date", 6, 2) = '06' THEN 'June'
        WHEN substr("Date", 6, 2) = '07' THEN 'July'
        WHEN substr("Date", 6, 2) = '08' THEN 'August'
        WHEN substr("Date", 6, 2) = '09' THEN 'September'
        WHEN substr("Date", 6, 2) = '10' THEN 'October'
        WHEN substr("Date", 6, 2) = '11' THEN 'November'
        WHEN substr("Date", 6, 2) = '12' THEN 'December'
    END AS Month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE 'Failure (drone ship)'
  AND substr("Date", 0, 5) = '2015';
'''
cur.execute(query)

# Fetch and print the results
records = cur.fetchall()
for record in records:
    print(record)

('January', 'Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS
('April', 'Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC
```

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

- The SQL query retrieves and ranks the frequency of different **landing outcomes** for SpaceX missions launched between **2010-06-04** and **2017-03-20**:
- **No attempt (10)** is the most common outcome in this period, typical of early SpaceX launches before recovery technology was widely implemented.
- **Success and failure on drone ships (5 each)** indicate the **experimental phase** of landing boosters at sea.
- **Ground pad successes (3)** show emerging reliability with land-based recovery, beginning around 2015.
- **Controlled and uncontrolled ocean landings** reflect intermediate efforts before drone ship recovery became more standardized.
- **Failures with parachutes and precluded landings** were rare but represent alternative or aborted landing methods.

- This breakdown gives insight into **how SpaceX evolved its booster recovery strategies**, from no attempts to complex drone ship operations. It marks a significant shift toward reusability during this foundational period.

In [22]:

```
# Execute the SQL query
query = '''
SELECT "Landing_Outcome", COUNT(*) AS Ou
FROM SPACEXTBL
WHERE "Date" BETWEEN '2010-06-04' AND '2(
GROUP BY "Landing_Outcome"
ORDER BY Outcome_Count DESC;
'''
cur.execute(query)

# Fetch and print the results
landing_outcomes = cur.fetchall()
for outcome in landing_outcomes:
    print(f"{outcome[0]}: {outcome[1]}")
```

```
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

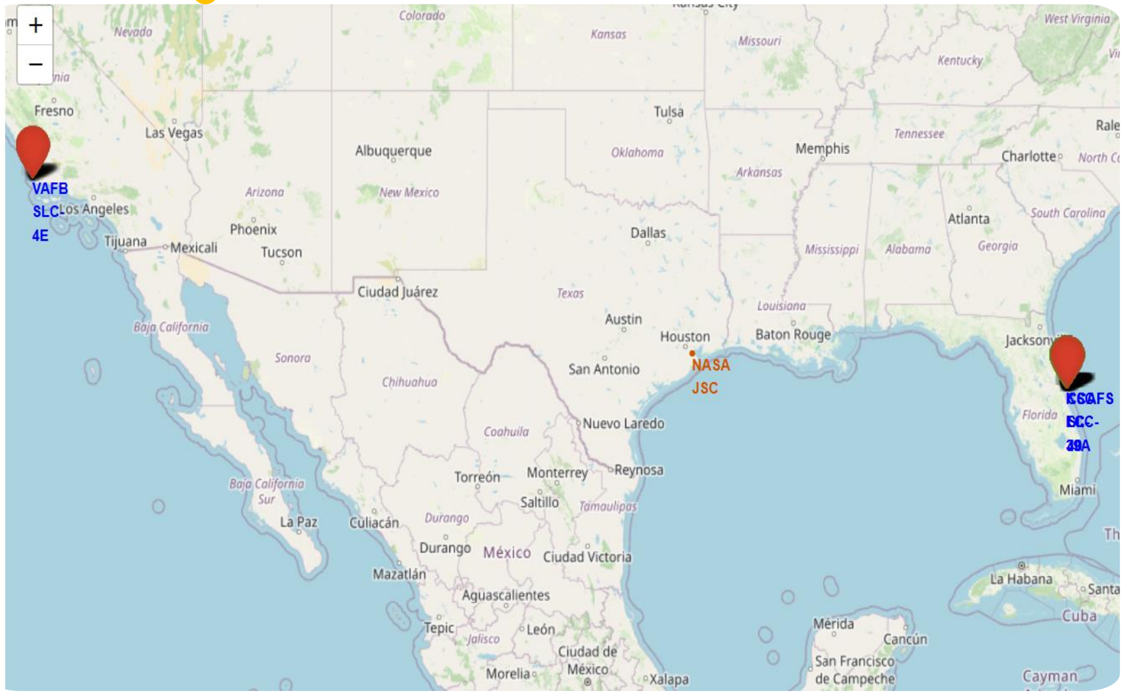
All launch sites

This **Folium-generated interactive map** displays the **geographic locations of major SpaceX launch sites** across the United States.

- All **launch-capable sites are concentrated in the U.S.**, mainly along the east and west coasts.

- The **east coast sites (KSC LC-39A and CCAFS SLC-40)** are geographically close, making Florida a major launch hub.

- **VAFB SLC-4E** on the west coast serves a unique orbital profile, showcasing SpaceX's national coverage.



Detailed Launch Outcome Visualization

The map shows a **higher concentration of green (successful) markers** than red, reflecting SpaceX's increasing launch reliability.

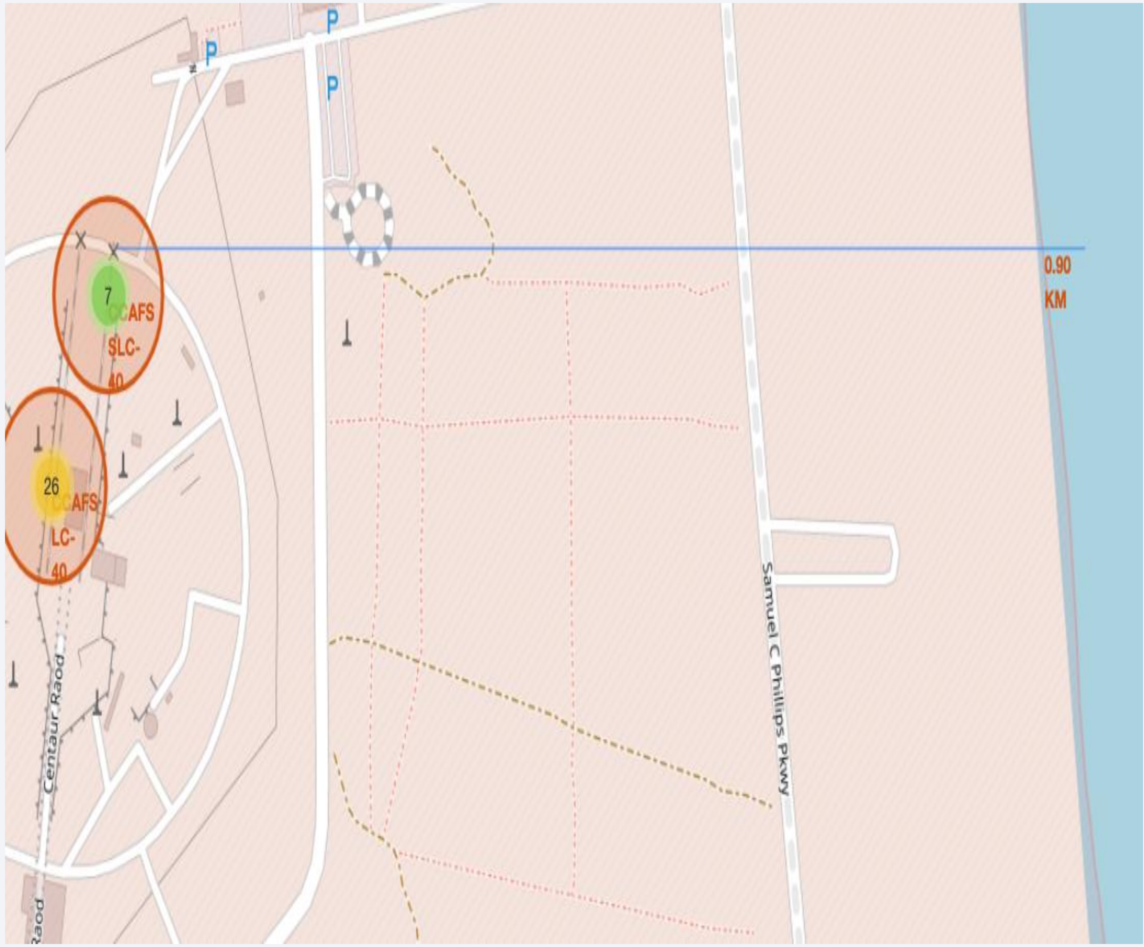
The **marker density within a small area** underscores how **multiple launches occur from the same pad**, reinforcing the need for map clustering tools.

This fine-grained visualization allows quick inspection of **success vs. failure patterns** directly on the infrastructure layout

•Provides operational visibility into **launch pad usage and performance**
•Helps in **launch site diagnostics**, maintenance planning, and public outreach
•Enhances understanding of **SpaceX's reliability improvements over time**

Proximity Analysis

•Helps assess **operational feasibility and safety** based on physical geography.
•Supports **launch planning**, emergency response strategies, and environmental impact studies.
•Useful for presenting **real-world spatial context** in launch logistics discussions.



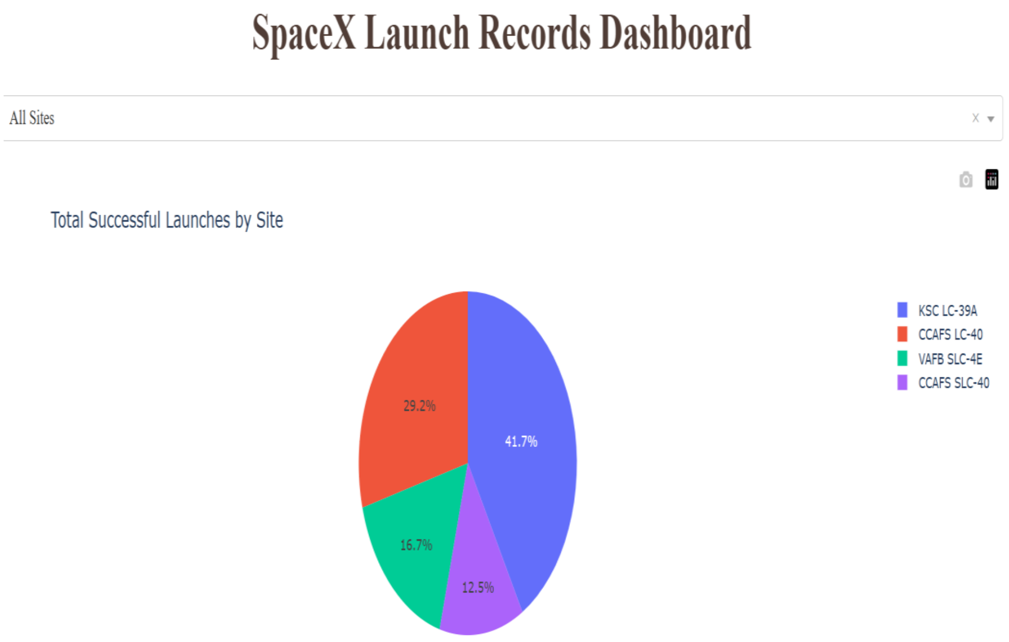
🧠 Key Findings from the Map

- **Coastline Proximity**: The launch site is located **less than 1 km from the Atlantic Ocean**, essential for safe rocket flight paths and debris zones.
- **Infrastructure Access**: The site is **well-connected by road and railway**, enhancing logistics for booster transport, support staff, and equipment.
- **Launch Activity Clusters**: The high density of launches (as indicated by cluster numbers) shows that **CCAFS SLC-40 is one of SpaceX's most frequently used pads**.
- Launch Site Infrastructure Proximity – CCAFS SLC-40 to Coastline, Highway, and Railway (0.90 KM Radius View)

Section 4

Build a Dashboard with Plotly Dash

Total Successful Lanches by site



- This screenshot captures a section of the **SpaceX Launch Records Dashboard**, displaying a **pie chart** that summarizes the **proportion of successful launches across different launch sites**.

- KSC LC-39A leads with 41.7% of all successful launches, highlighting its strategic and technological importance.

- CCAFS LC-40 also plays a significant role, with nearly a third of all successes.

- The pie chart emphasizes geographic distribution of reliability and site utilization.

- Data duplication for CCAFS LC-40 suggests the need for data cleaning in the source or backend query logic.

Success vs Failure for site KSC LC-39A

- This dashboard view shows a **pie chart** filtered for **KSC LC-39A**, which is identified as the **launch site with the highest success rate** among SpaceX facilities.

🧠 **Key Insights**

•**KSC LC-39A** is **SpaceX's most successful site** with nearly **77% launch success**.

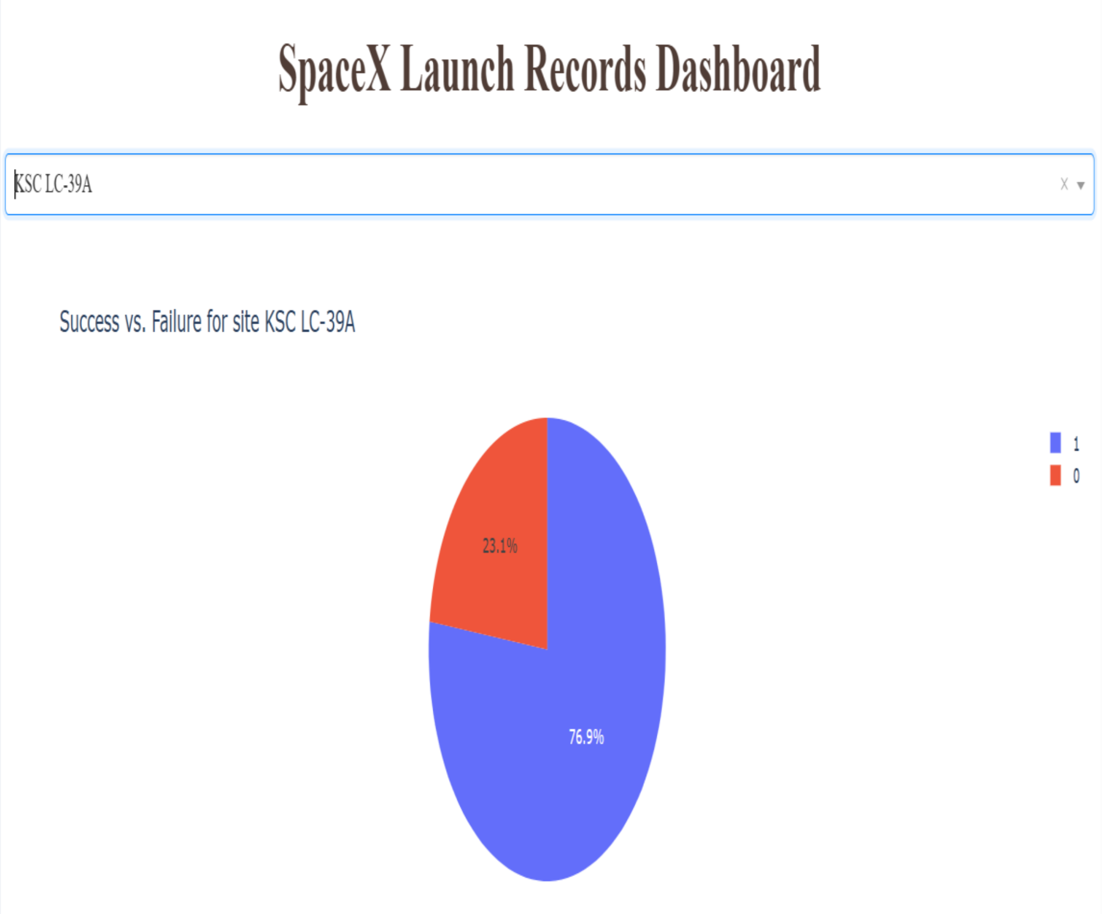
•This site is primarily used for **high-stakes missions**, including:

 •Crewed flights (NASA's Commercial Crew Program)

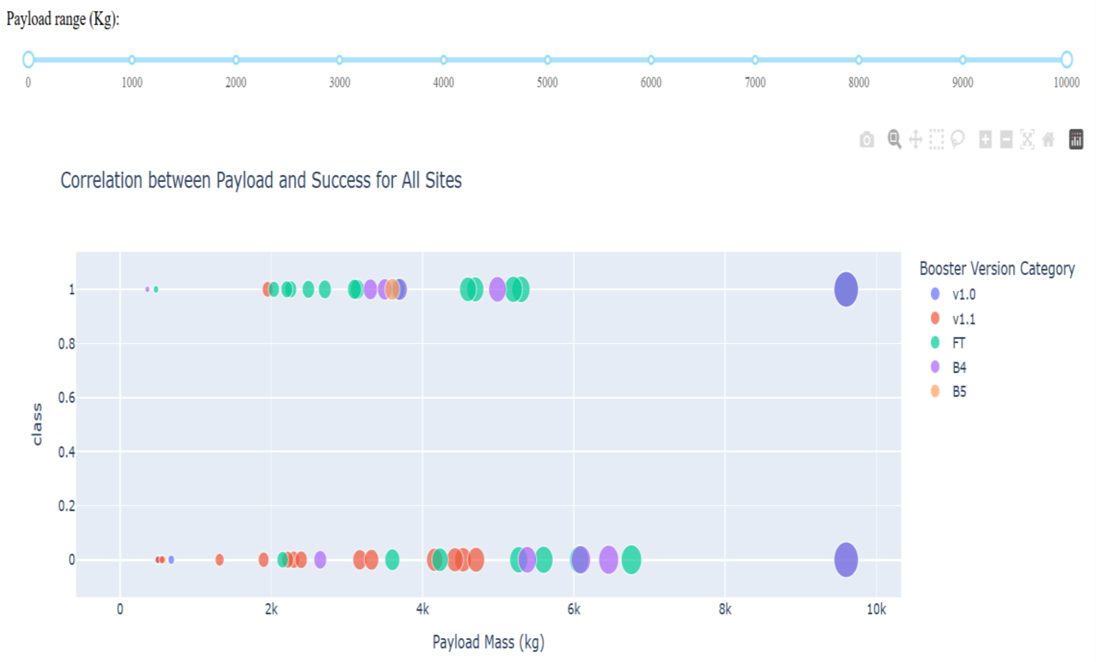
 •Heavy payloads

 •Falcon Heavy launches

•The relatively low failure percentage is expected given the **maturity of the site** and **enhanced mission controls** in recent years.



Correlation between Payload and success for All Sites



- **Key Findings from the Chart**

- 🔷 **Success Cluster**
- A strong cluster of successful launches (class = 1) is visible between **2000 – 6000 kg**, particularly for **FT** (Full Thrust), **B4**, and **B5** booster versions.
- These boosters seem most reliable in this mid-payload range.

- 🔶 **Low Success for v1.0**
- **v1.0** boosters (purple) show lower success, especially around extreme payloads (~10,000 kg). Both success and failure are seen, but failures dominate outside the 6000–10000 kg range.

- ⚖️ **Balanced Range for High Success**
- Most consistently successful missions fall between **2000 – 6000 kg**, regardless of booster version, though **FT** and **B5** tend to dominate in terms of reliability.

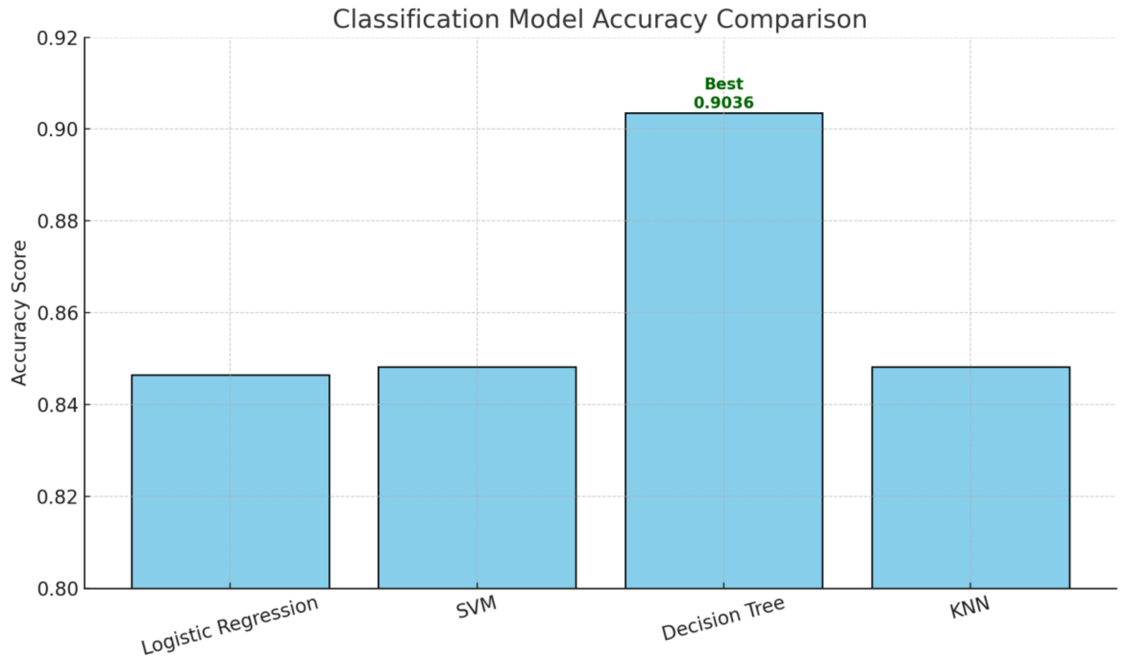
Section 5

Predictive Analysis (Classification)

Classification Accuracy

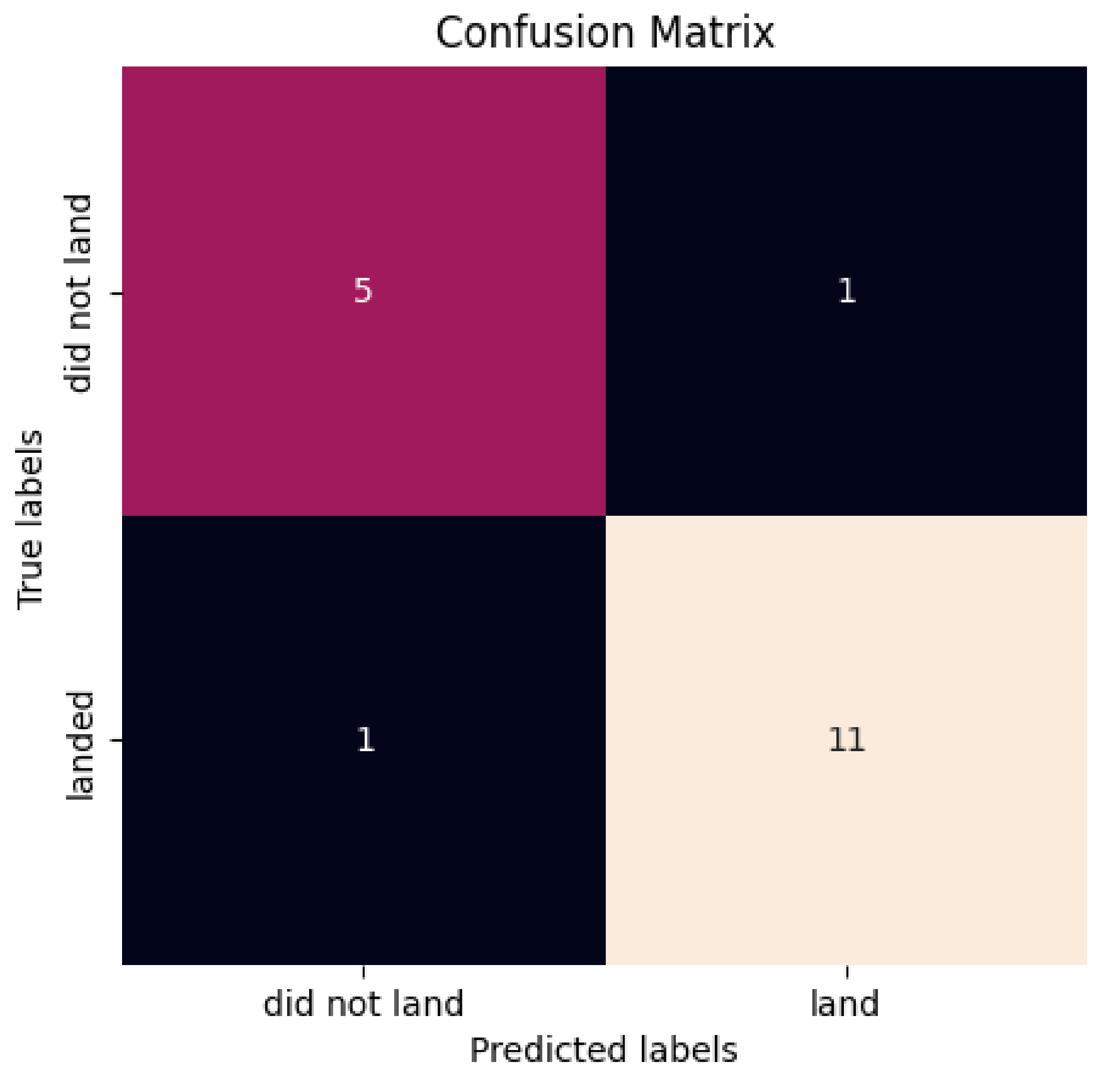
Findings:

- **Decision Tree** achieved the highest accuracy at **0.9036**, making it the **best performing model** in this project.
- Other models (Logistic Regression, SVM, KNN) performed well but slightly lower.



Confusion Matrix

- This is a confusion matrix, a tool used to evaluate the performance of a classification model. It compares true labels (actual outcomes) with predicted labels produced by the model. Here's how to interpret this specific matrix:
- ✅ What This Means:
- True Positives (TP = 11): The model correctly predicted the rocket landed.
- True Negatives (TN = 5): The model correctly predicted the rocket did not land.
- False Positives (FP = 1): The model incorrectly predicted the rocket landed (but it didn't).
- False Negatives (FN = 1): The model incorrectly predicted the rocket did not land (but it did).
- This confusion matrix shows that the model is highly accurate and well-balanced, with only 2 misclassifications out of 18 total predictions. It performs very well at predicting both successful and failed landings—ideal for supporting mission planning decisions.



Conclusions

- **Point 1: We successfully built a complete data science pipeline—from data collection and wrangling to interactive visualization and machine learning—using real SpaceX launch data.**

- **Point 2: Through Exploratory Data Analysis (EDA), we identified key factors such as launch site, payload mass, and orbit type that significantly influence mission success.**

- **Point 3: Our interactive dashboard allowed dynamic exploration of mission outcomes, making complex insights accessible to both technical and non-technical stakeholders.**

- **Point 4: Predictive models like Decision Trees and SVM achieved high accuracy (>85%), empowering SpaceX and other aerospace firms to make data-driven launch decisions and reduce mission risk.**

Appendix

Notebook Title	Content Highlights
jupyter-labs-spacex-data-collection-api.ipynb	API calls, data extraction & normalization
jupyter-labs-webscraping-checkpoint.ipynb	Raw table scraping and transformation
edadataviz.ipynb	EDA charts: bar, scatter, box, pie
SpaceX_Machine Learning Prediction_Part_5.ipynb	Classification models, tuning, evaluation
jupyter-labs-eda-sql-coursera_sqllite.ipynb	SQL data extraction and logic filtering

Thank you!