IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Memona Mubashir
<17-9-2025>

Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

- **SUMMARY OF METHODOLOGIES**

COLLECTED AND PREPARED FALCON 9 LAUNCH DATA, APPLIED **DATA WRANGLING, FEATURE ENGINEERING, AND VISUALIZATION** TO IDENTIFY PATTERNS AND KEY FEATURES INFLUENCING BOOSTER LANDING SUCCESS.

BUILT AND TUNED MULTIPLE **CLASSIFICATION MODELS** (LOGISTIC REGRESSION, DECISION TREE, KNN, SVM), AND USED **GRIDSEARCHCV** FOR HYPERPARAMETER OPTIMIZATION AND CROSS-VALIDATION.

SUMMARY OF RESULTS

ALL MODELS PERFORMED REASONABLY WELL, BUT **SVM AND LOGISTIC REGRESSION** SHOWED HIGHER ACCURACY COMPARED TO DECISION TREE AND KNN.

THE **BEST-PERFORMING MODEL** ACHIEVED THE HIGHEST ACCURACY IN PREDICTING BOOSTER LANDING OUTCOMES, MAKING IT THE MOST RELIABLE FOR FUTURE MISSION SUCCESS FORECASTING.

Introduction

- **Project Background and Context**

SpaceX's business model depends on the **reusability of Falcon 9 first stage boosters**, which makes accurate landing prediction critical for cost reduction and mission success.

Using **historical launch data** and machine learning, we aimed to build a model that predicts the likelihood of a successful booster landing.

- **Problems We Want to Answer**

Can we predict whether the Falcon 9 first stage will successfully land based on launch characteristics (payload, orbit, launch site, etc.)?

Which machine learning model performs best in predicting Falcon 9 landing outcomes?

Section 1

Methodology

Methodology

- Executive Summary
- **Data collection methodology:**
    - Used Spacex Rest API
    - Data retrieved with request library (.JSON format)
    - Convert JSON into Pandas DataFrame using json_normalize
- Perform data wrangling
    - Retrieved detailed info (Booster, Launchpad, Payload, Core) via API endpoints
    - Filtered dataset to include only **Falcon 9 launches**
    - Replaced missing PayloadMass with mean PayloadMass values
    - Kept LandingPad NULLs (handled later with one-hot encoding)
- 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

Used **historical Falcon 9 launch data** to train machine learning classification models for predicting booster landing success.

Applied **feature engineering and preprocessing** (e.g., one-hot encoding, scaling, data splitting) to prepare data for modeling.

- **How to Build, Tune, and Evaluate Classification Models**

Build: Trained multiple models (Logistic Regression, Decision Tree, KNN, SVM) using training data.

Tune: Optimized hyperparameters with **GridSearchCV** to improve model accuracy and generalization.

Evaluate: Compared models using **accuracy, confusion matrix, and classification report** to identify the best-performing classifier.

Data Collection

Data Collection –Key Points:

→ **Primary Source:** SpaceX REST API (/launches/past) provides structured launch data in JSON format.

→ **Secondary Source:** Web scraping Wikipedia Falcon 9 launch records using **BeautifulSoup** to supplement API data.

→ **Data Retrieval:** API requests handled with Python's **requests** library; results converted to JSON objects.

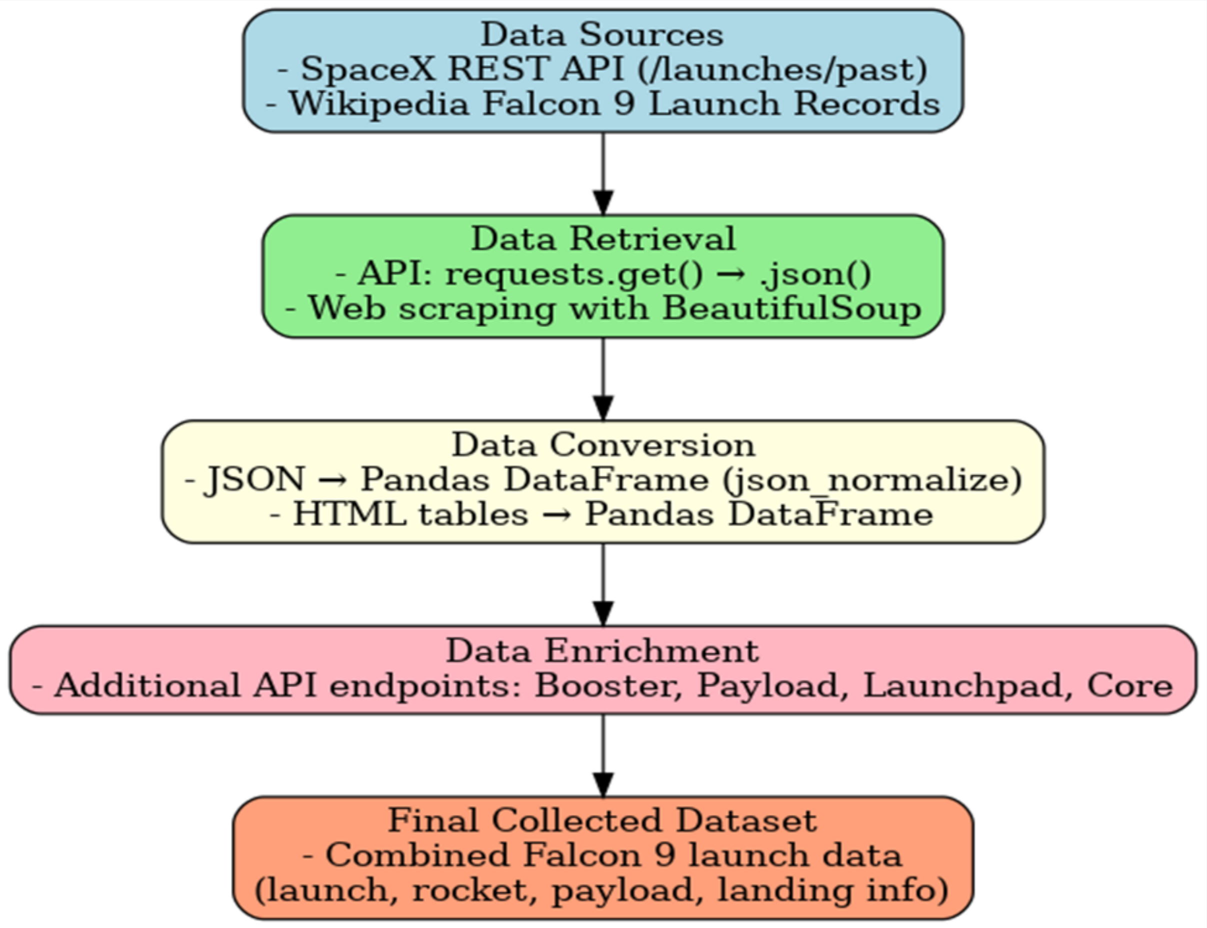
→ **Data Conversion:** JSON responses normalized into a **flat Pandas DataFrame** (json_normalize), while HTML tables scraped from Wikipedia were parsed into DataFrames.

→ **Data Enrichment:** Used additional API endpoints (Booster, Payload, Launchpad, Core) to resolve IDs and gather more detailed attributes.

→ **Outcome:** A combined raw dataset containing launch, rocket, payload, and landing information, ready for **wrangling and cleanin**

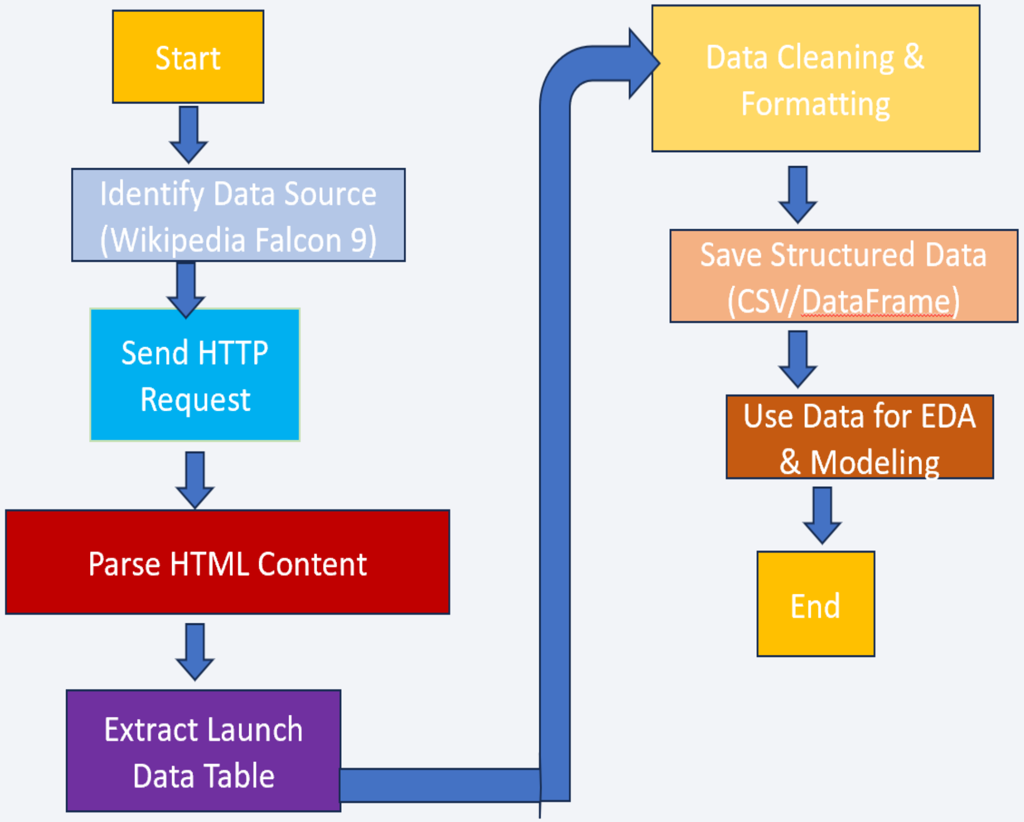
Data Collection – SpaceX API

- URL="[Applied-Data-Science-Capstone/jupyter-labs-spacex-data-collection-api.ipynb at main · memonamubashir/Applied-Data-Science-Capstone]"



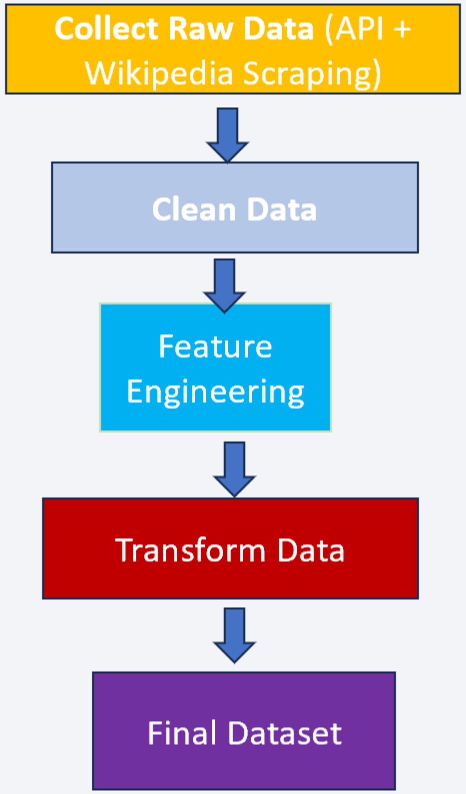
Data Collection - Scraping

•**Data Collection Stage** Start from identifying the target website (Wikipedia/SpaceX API), send requests, and extract HTML data.
•**Data Processing Stage** Parse the data (using BeautifulSoup/Pandas), clean and structure it into tabular form for analysis.
•**Data Storage & Use Stage** Store cleaned data in CSV/SQLite and feed it into machine learning models for prediction
•URL="Applied-Data-Science-Capstone/jupyter-labs-webscraping.ipynb at main · memonamubashir/Applied-Data-Science-Capstone"



Data Wrangling

•**Data Collection** SpaceX API & Wikipedia scraping
•**Data Cleaning** Handle missing values, drop irrelevant columns, standardize formats
•**Feature Engineering** Extract useful features (e.g., Fligh number, Payload mass, Orbit, Landing outcome)
•**Data Transformation** Encode categorical variables, normalize numerical features
•**Data Storage** Save as CSV / SQLite for analysis
•URL="Applied-Data-Science-Capstone/labs-jupyter-spacex-Data wrangling.ipynb at main · memonamubashir/Applied-Data-Science-Capstone"



EDA with Data Visualization

- **Charts Plotted & Purpose**
- **Bar Charts**
  - Used to show the relationship between categorical variables (e.g., launch site vs. landing success).
  - Helped in comparing **success rates across different launch sites**.
- **Pie Charts**
  - Used to visualize the **proportion of successful vs. failed landings**.
  - Provided a quick **overview of success distribution**.
- **Scatter Plots**
  - Used to show the **relationship between payload mass and landing success**.
  - Helped to understand how **payload weight influences landing outcome**.
- **Line Charts (Success over time)**
  - Used to track **success rate trends across years**.
  - Helped to identify **improvement in SpaceX landing performance over time**.
- URL="Applied-Data-Science-Capstone/edadataviz.ipynb at main · memonamubashir/Applied-Data-Science-Capstone"

EDA with SQL

- **Select and Filter Queries**
  - Retrieved launch records with specific conditions (e.g., only successful landings).
  - Filtered data by launch site, orbit type, or success class.
- **Aggregation Queries**
  - Counted the number of successful vs. failed landings.
  - Calculated average payload mass for different launch outcomes.
- **Grouping Queries**
  - Grouped launches by **orbit type** to analyze success rates.
  - Grouped by **launch site** to compare performance across locations.
- **Join Queries**
  - Combined data from multiple tables (e.g., mission details with landing outcomes).
- **Ordering Queries**
  - Sorted launches by **payload mass** or **date** to identify patterns over time.
- 👉 These queries were used to **explore, clean, and understand the SpaceX dataset**, setting the foundation for predictive analysi
- URL="Applied-Data-Science-Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · memonamubashir/Applied-Data-Science-Capstone"

Build an Interactive Map with Folium

- 🗺️ **Map Objects Created**
- **Markers**
  - Added at each launch site location.
  - Used to **pinpoint the exact geographic coordinates** of SpaceX launch sites.
- **Circle Markers**
  - Plotted around launch sites.
  - Used to **highlight the region** and make the launch sites visually stand out.
- **Popup Labels**
  - Attached to markers with site names.
  - Helped in **identifying each launch site easily on the map**.
- **Lines (Polylines)**
  - Drew lines from launch sites to the nearest city, coastline, or infrastructure (railway, highway).
  - Used to **visualize proximity** and analyze whether location factors influence landing success.
- Explain why you added those objects
- URL="Applied-Data-Science-Capstone/lab_jupyter_launch_site_location (copy) (copy) (copy) (copy).ipynb at main · memonamubashir/Applied-Data-Science-Capstone"

Build an Interactive Map with Folium

- 🎯 **Why These Objects Were Added**
- To **visualize SpaceX launch site locations** on an interactive map.
- To **explore geographical factors** like distance to coastlines, roads, and cities.
- To make the map **interactive and informative**, allowing quick interpretation of spatial relationships.
- URL="Applied-Data-Science-Capstone/lab_jupyter_launch_site_location (copy) (copy) (copy) (copy).ipynb at main · memonamubashir/Applied-Data-Science-Capstone"

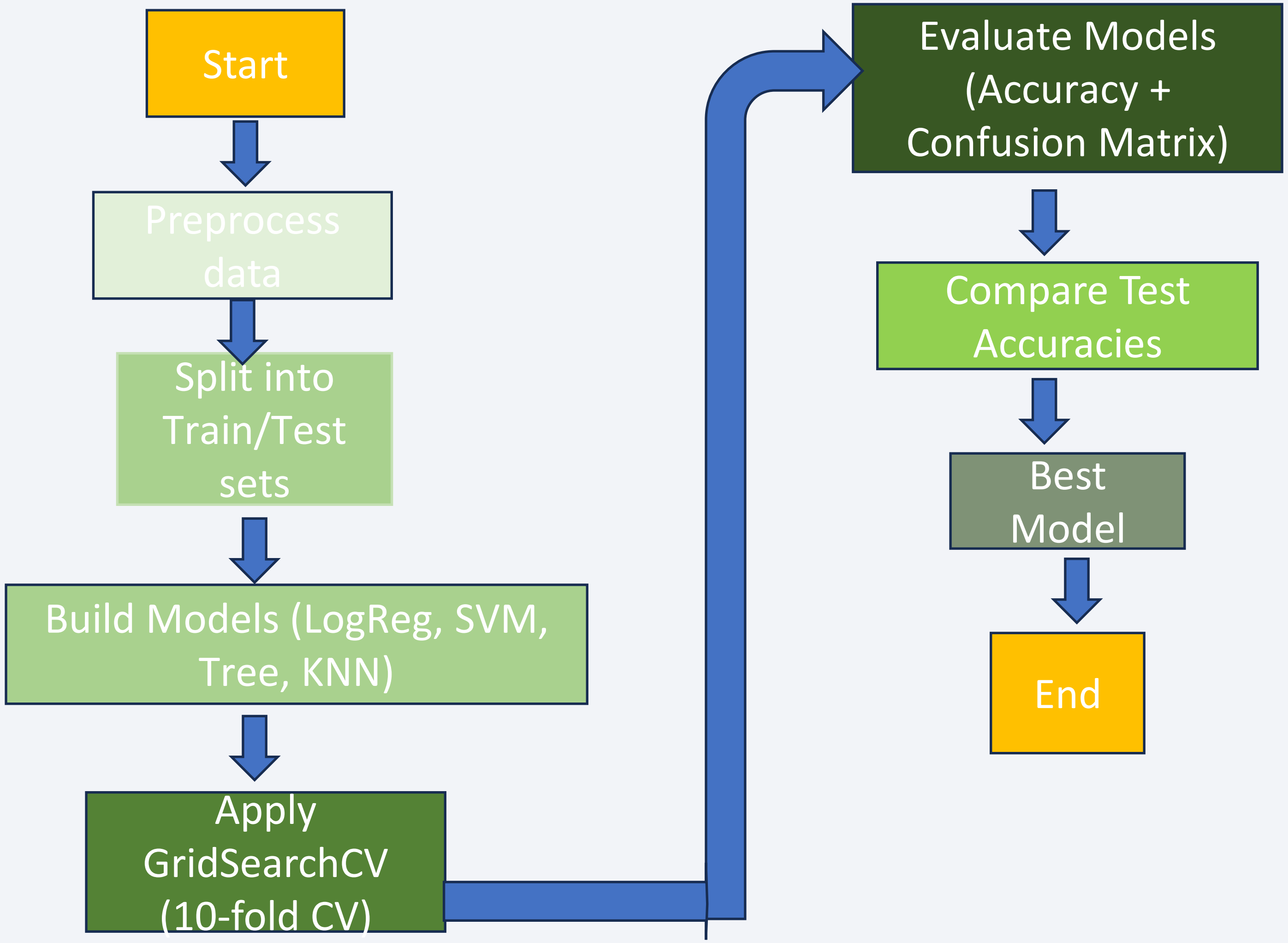
Build a Dashboard with Plotly Dash

- **Summary of Plots, Graphs, and Interactions Added:**
  - Launch Site Dropdown (Task 1)
    - **Functionality:** Lets users select either "All Sites" or a specific launch site to filter the visualizations.
  - Success Pie Chart (Task 2)
    - If "All Sites" is selected → shows total successful launches per site.
    - If a specific site is selected → shows success vs failure (class 1 vs 0) for that site.
  - Payload Range Slider (Task 3)
    - **Functionality:** Allows users to select a payload mass range (0–10000 kg)
  - Success-Payload Scatter Plot (Task 4)
    - **Functionality:**
    - X-axis → Payload Mass (kg)
    - Y-axis → Mission outcome (class: 0 = failure, 1 = success)
    - Color → Booster Version Category
    - Supports both "All Sites" and specific site selection.

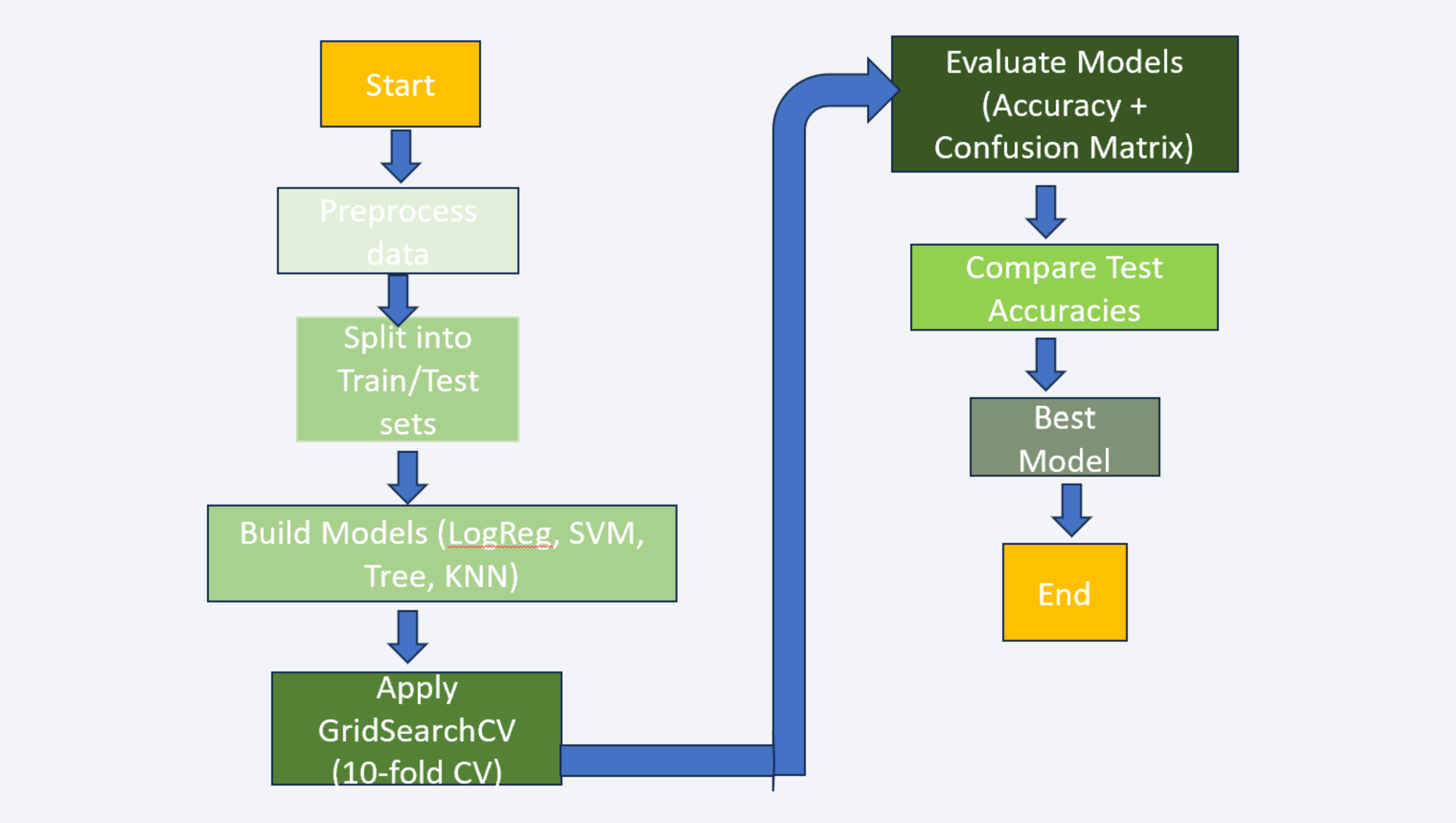
Build a Dashboard with Plotly Dash

- **Why These Plots and Interactions Were Added:**
- **Dropdown for launch site:** Enables users to focus on a particular site or view overall performance, making comparisons easy.
- **Pie chart:** Provides a quick visual summary of success counts, which helps answer questions like "Which site has the largest successful launches?" and "Which site has the highest success rate?"
- **Payload range slider:** Adds interactivity for payload-based analysis, helping to answer "Which payload range(s) has the highest or lowest launch success rate?"
- **Scatter plot:** Shows detailed correlations between payload and mission outcome, and helps visualize the impact of different booster versions on success rates. This plot gives insights into patterns not obvious from aggregated data.
- URL="Applied-Data-Science-Capstone/spacex-dash-app.py at main · memonamubashir/Applied-Data-Science-Capstone"

Predictive Analysis (Classification)



Predictive Analysis (Classification)



Predictive Analysis (Classification)

- **Model Development Process Summary**
- **1. Data Preparation**
- Imported datasets (dataset_part_2.csv, dataset_part_3.csv).
- Extracted target variable **Y** (Class) and features **X**.
- Standardized features using StandardScaler.
- Split dataset into **training** and **test sets**.
- **2. Model Building**
- Built four classification models:
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Decision Tree
    - K-Nearest Neighbors (KNN)
- **3. Hyperparameter Tuning**
- Used **GridSearchCV** with **10-fold cross-validation** for each model.
- Tuned hyperparameters (e.g., C, gamma, kernel, max_depth, n_neighbors).

Predictive Analysis (Classification)

- **4. Model Evaluation**
- Evaluated models using:
  - **Cross-validation accuracy** (from GridSearchCV).
  - **Test set accuracy** (using .score(X_test, Y_test)).
  - **Confusion matrix** to visualize predictions.
- **5. Model Comparison & Selection**
- Compared test accuracies of all four classifiers.
- Found that **Logistic Regression** achieved the highest accuracy (**~83%**).
- ✅ **Final Model Chosen: Logistic Regression**.
- URL="Applied-Data-Science-Capstone/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · memonamubashir/Applied-Data-Science-Capstone"

Results

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

Section 2

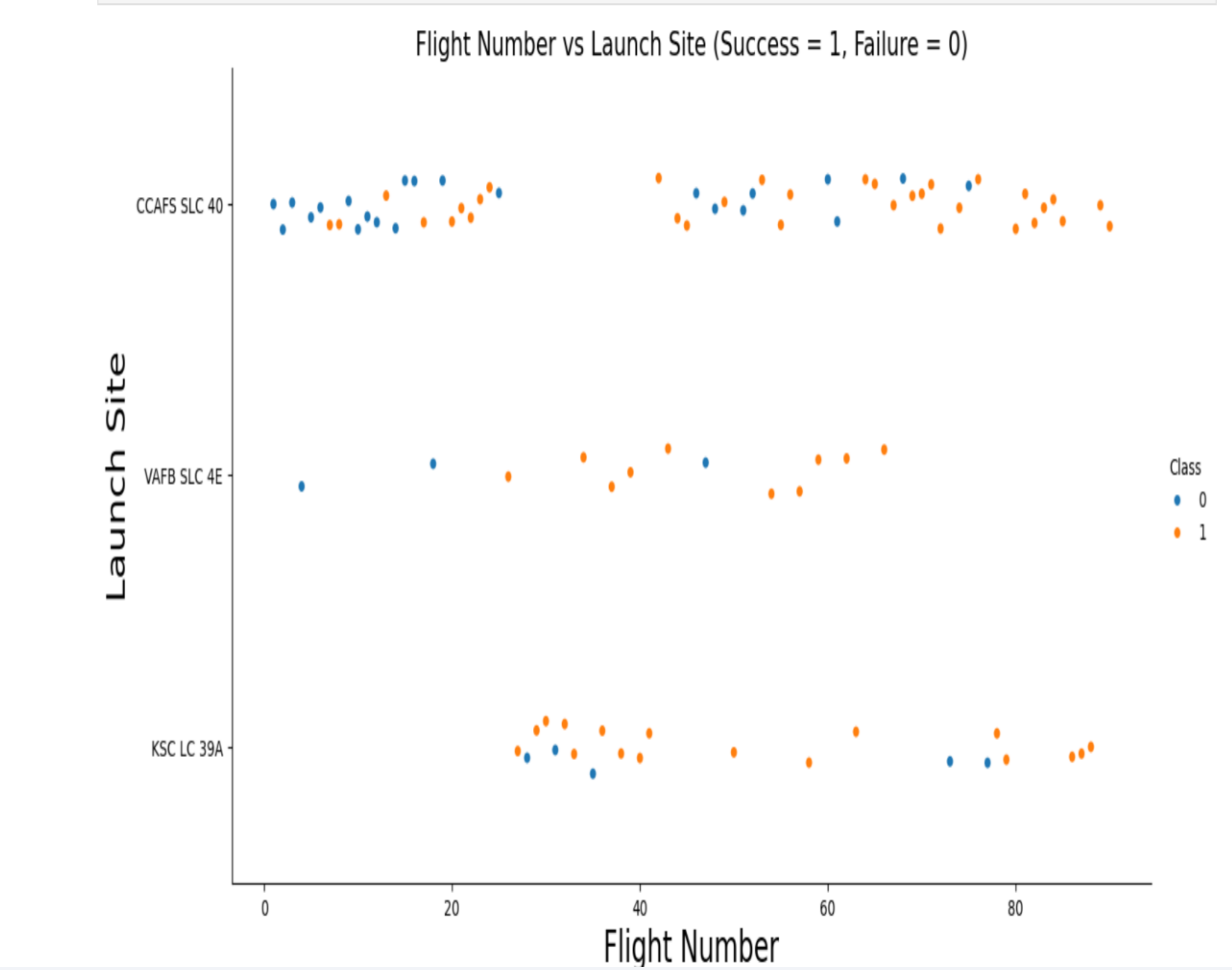
Insights drawn from EDA

Flight Number vs. Launch Site

•Shows how launch success varied across **different launch sites**.

•Helps identify whether **certain launch sites had higher success rates**.

•Also allows us to see **if success improved over time** (higher flight numbers).

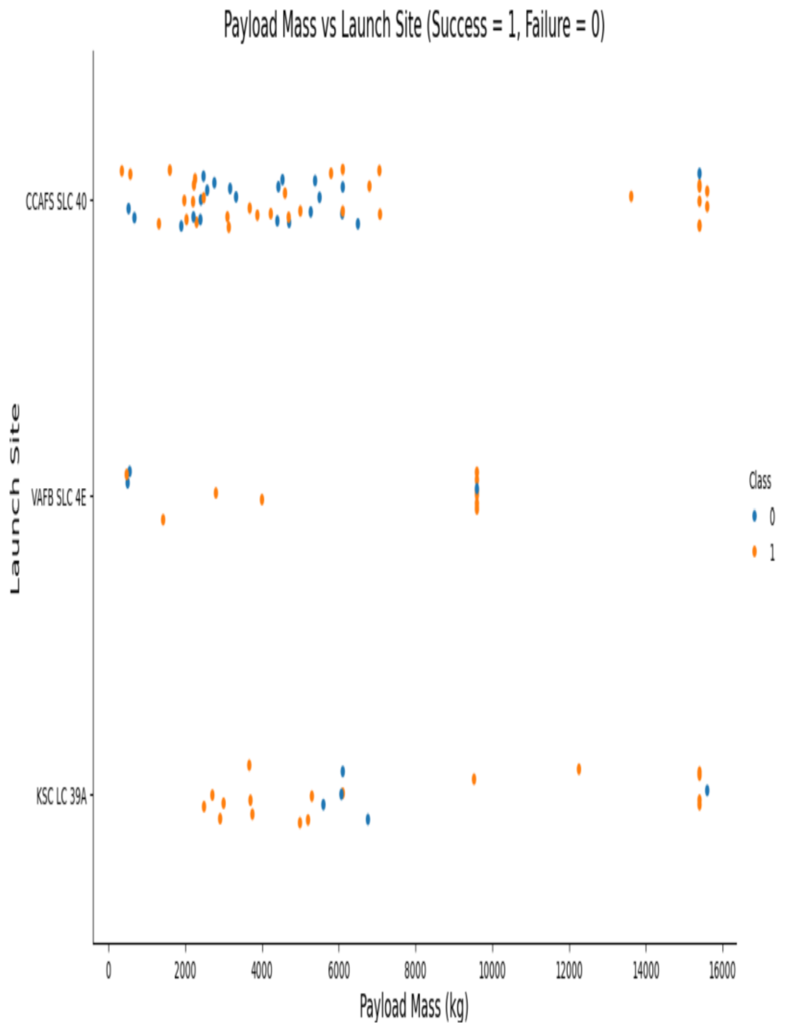


Payload vs. Launch Site

- Helps analyze if **payload weight influences landing success**.

- Allows comparison of **payload ranges across different launch sites**.

- Shows whether **heavier payloads correlate with lower success rates** at certain sites.



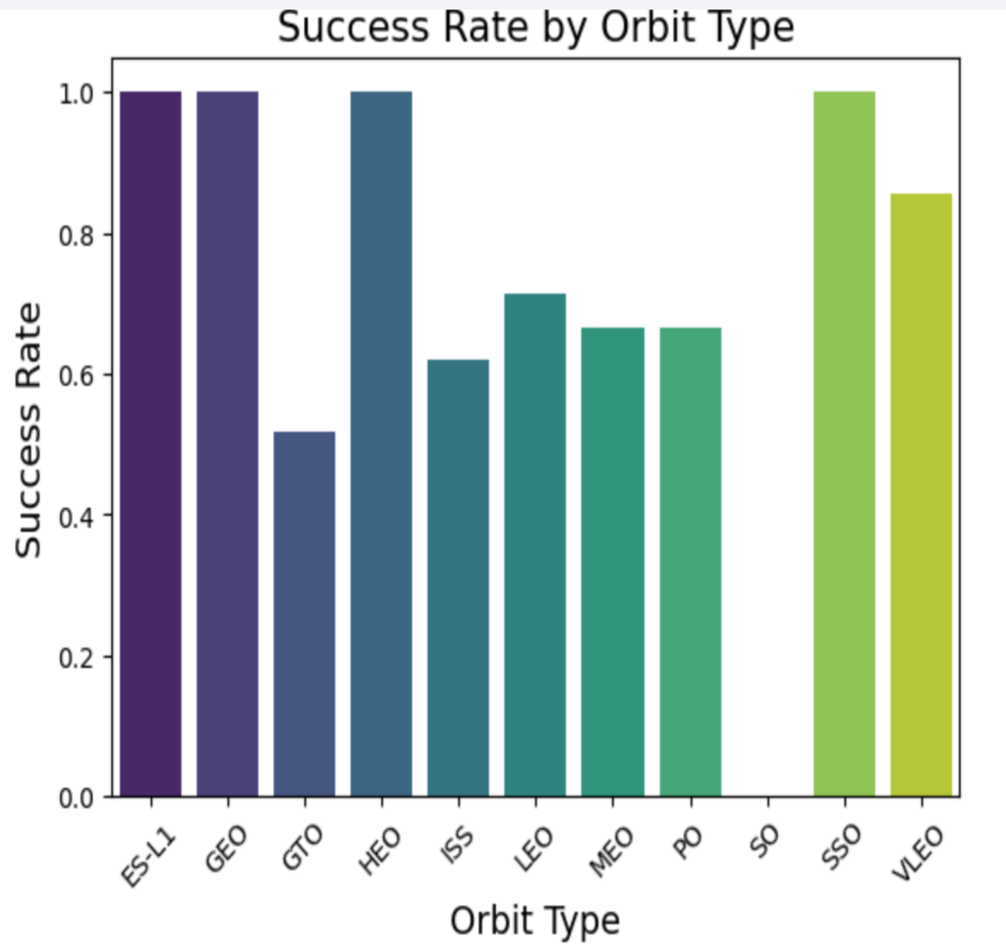
Success Rate vs. Orbit Type

•You can **compare landing success probabilities** across orbit types.

•For example:
•Some orbits (like LEO) may show **higher success rates**.

•Others (like GTO with heavy payloads) may show **lower success rates**.

•This helps understand how **mission type affects Falcon 9 landing performance**.

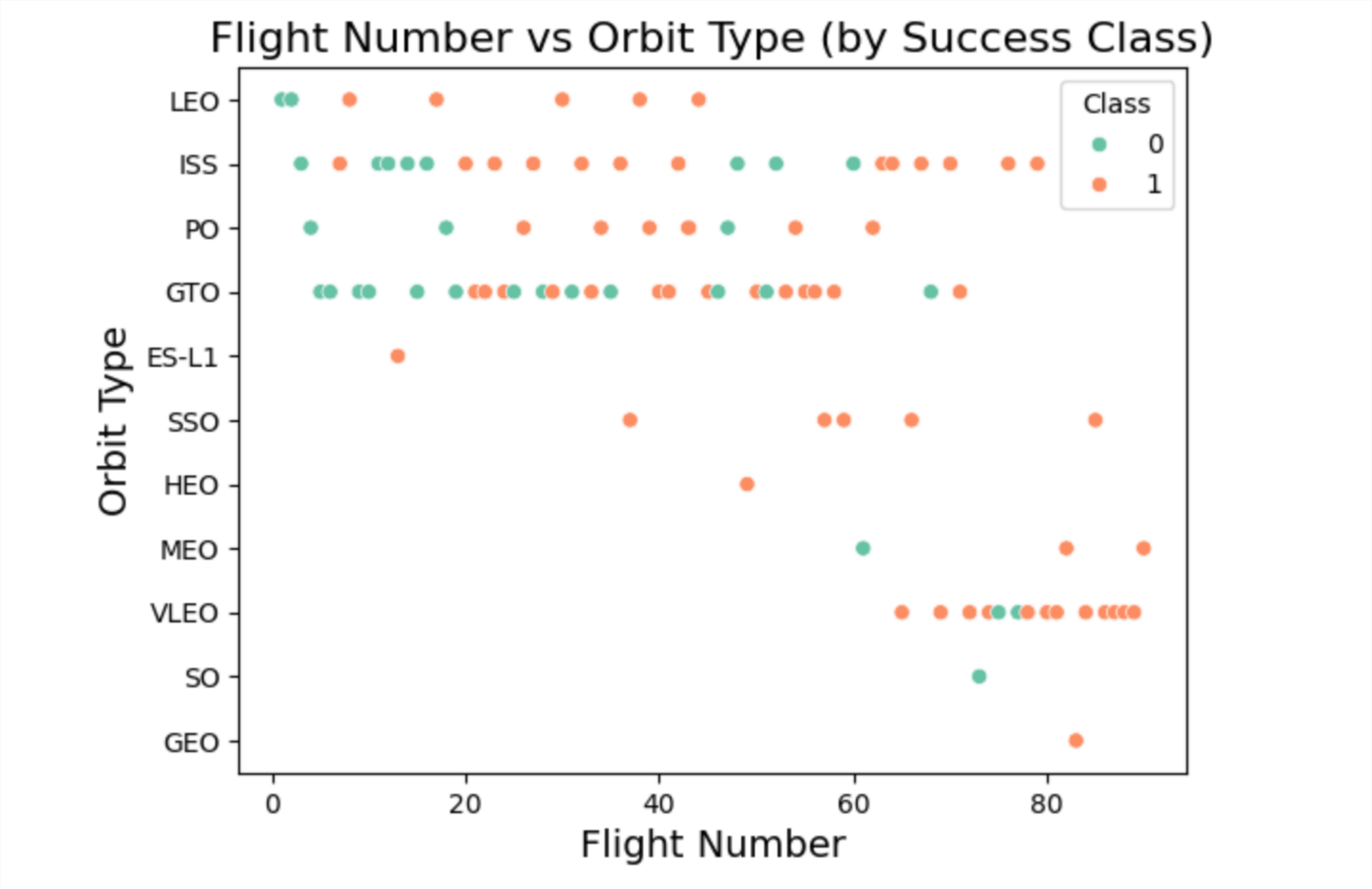


Flight Number vs. Orbit Type

•Shows the **distribution of orbits across Falcon 9 flight history**.

•Helps check whether **landing success rates improved** with higher flight numbers.

•Reveals whether some **orbit types are more challenging**, as failures may cluster in certain categories.

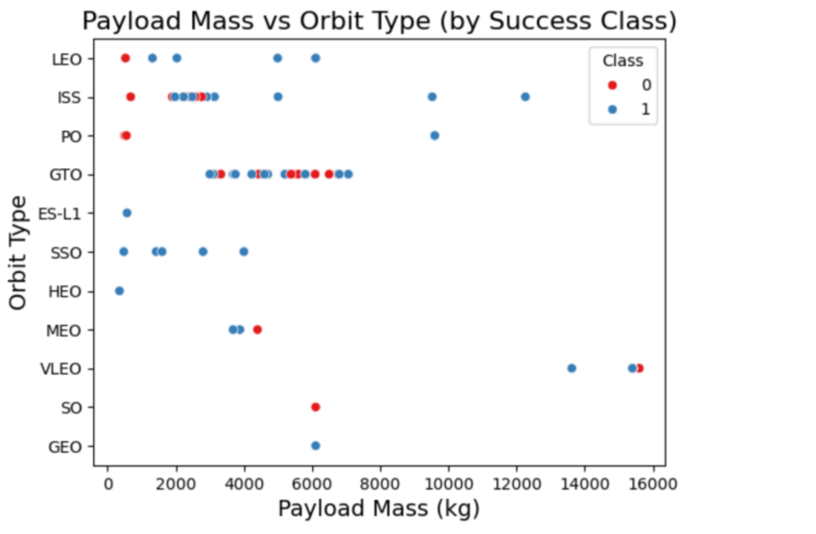


Payload vs. Orbit Type

- Shows which **orbit types usually involve heavier payloads** (e.g., GTO often has higher payload mass).

- Helps analyze whether **heavier payloads affect landing success rates**.

- Useful to spot trends, e.g., **lighter payloads → more successful landings** or **certain orbits with consistent success/fail patterns**.

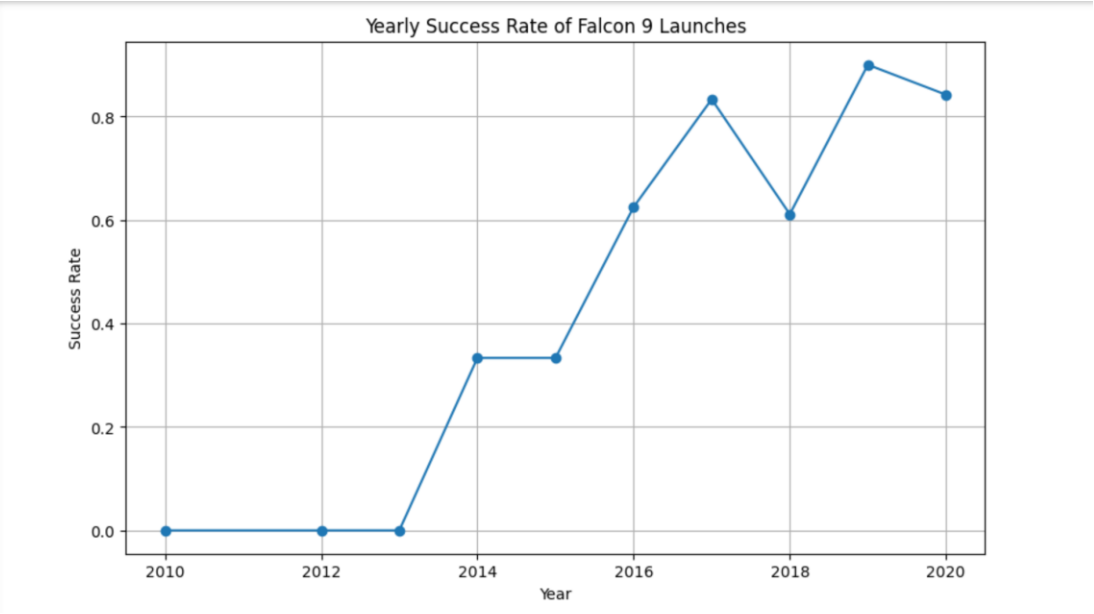


Launch Success Yearly Trend

- Early years (2010–2014) had **low or zero success rates**, as landings were still experimental.

- Around **2015–2016**, success rates **started climbing rapidly**.

- In later years (2017+), success rates **stabilized at high levels (~0.8–1.0)**, showing SpaceX mastered reusable rocket technology



Find the names of the unique launch sites

All Launch Site Names

```
[10]: %sql SELECT DISTINCT "Launch site" FROM SPACEXTABLE;
```

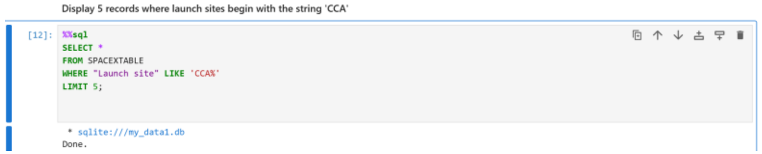
```
 * sqlite:///my_data1.db
Done.
```

[10]:

"Launch site"
Launch site

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with \`CCA\`



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

```
[13]: %%sql SELECT SUM("Payload mass") AS Total_PayloadMass
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

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

[13]:

Total_PayloadMass
0.0

Total Payload Mass

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

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

```
[14]: %%sql SELECT AVG("Payload mass") AS Average_PayloadMass
FROM SPACEXTABLE
WHERE "Booster Version" = 'F9 v1.1';
```

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

[14]:

Average_PayloadMass
None

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

Task 5

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

Hint:Use min function

[15]:
```
%%sql SELECT MIN(Date) AS First_Successful_Ground_Pad_Landing
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

[15]:

First_Successful_Ground_Pad_Landing
2015-12-22

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

Task 6

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

[16]:
```
%%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.
```

[16]:

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

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

Task 7

List the total number of successful and failure mission outcomes

[17]:
```
%%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Count
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

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

[17]:

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

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

Task 8

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

[18]:
```
%%sql SELECT DISTINCT "Booster_Version", "PayloadMass"
FROM SPACEXTABLE
WHERE "PayloadMass" = (
    SELECT MAX("PayloadMass")
    FROM SPACEXTABLE
);
```

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

[18]:

Booster_Version	"PayloadMass"
F9 v1.0 B0003	PayloadMass
F9 v1.0 B0004	PayloadMass
F9 v1.0 B0005	PayloadMass
F9 v1.0 B0006	PayloadMass

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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.

[19]:
```
%%sql SELECT
    substr("Date", 6, 2) AS Month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE '%Failure%'
  AND "Landing_Outcome" LIKE '%drone ship%'
  AND substr("Date", 1, 4) = '2015';
```

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

[19]:

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

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

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

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

[20]:

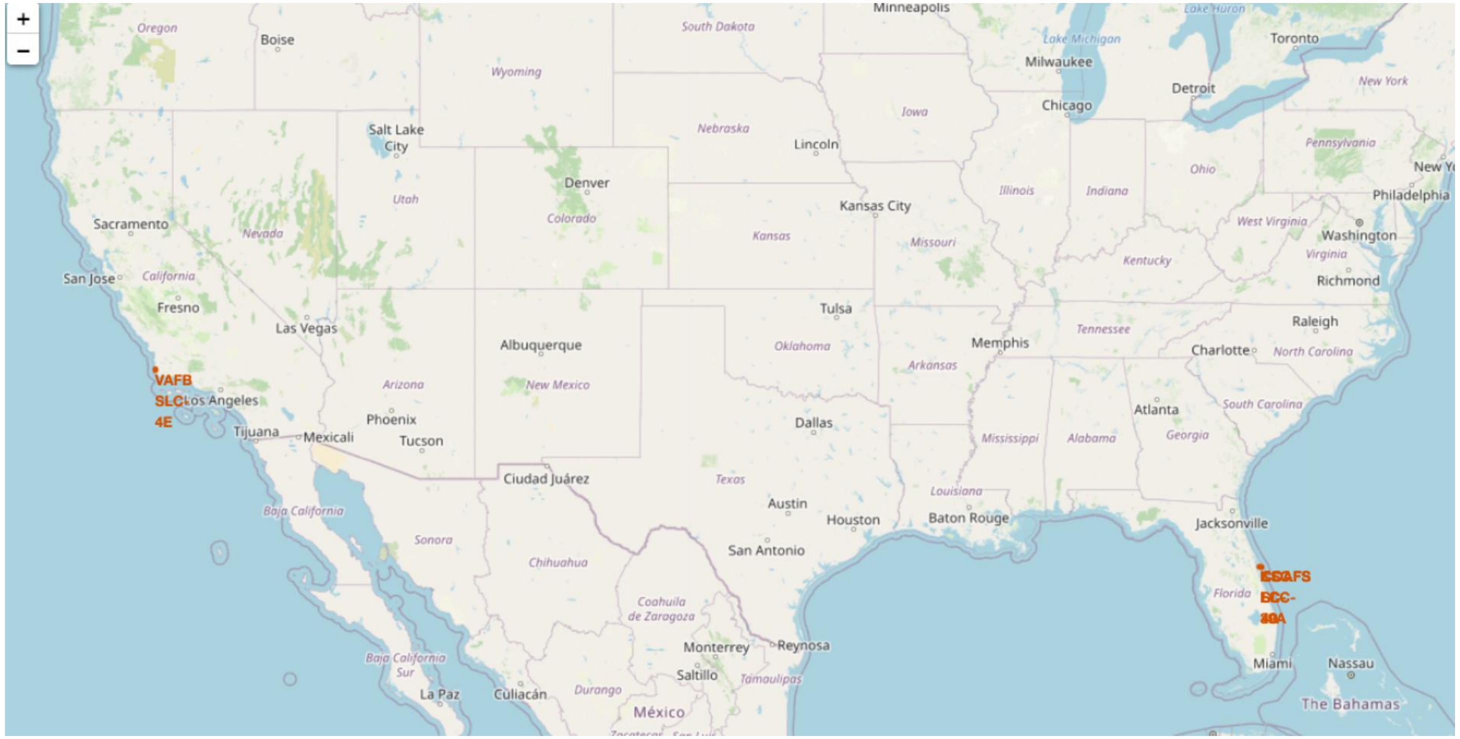
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

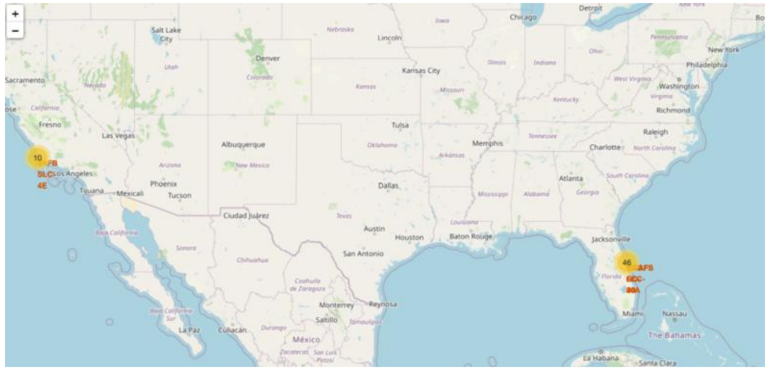
Generated map with marked launch sites

•**Markers (blue rocket icons):**
Each represents a major SpaceX launch site. Clicking a marker shows the site name.
•**CCAFS LC-40 & CCAFS SLC-40** → Cape Canaveral, Florida.
•**KSC LC-39A** → Kennedy Space Center, Florida.
•**VAFB SLC-4E** → Vandenberg Air Force Base, California.
•**Global map view:**
Provides context showing that **3 sites are clustered in Florida's east coast** and **1 site in California**, highlighting SpaceX's main U.S. launch infrastructure.
•**Findings:**
•Florida launch sites are close to the Atlantic Ocean, ideal for safe rocket trajectory paths.
•The California site enables launches into **polar orbits**.



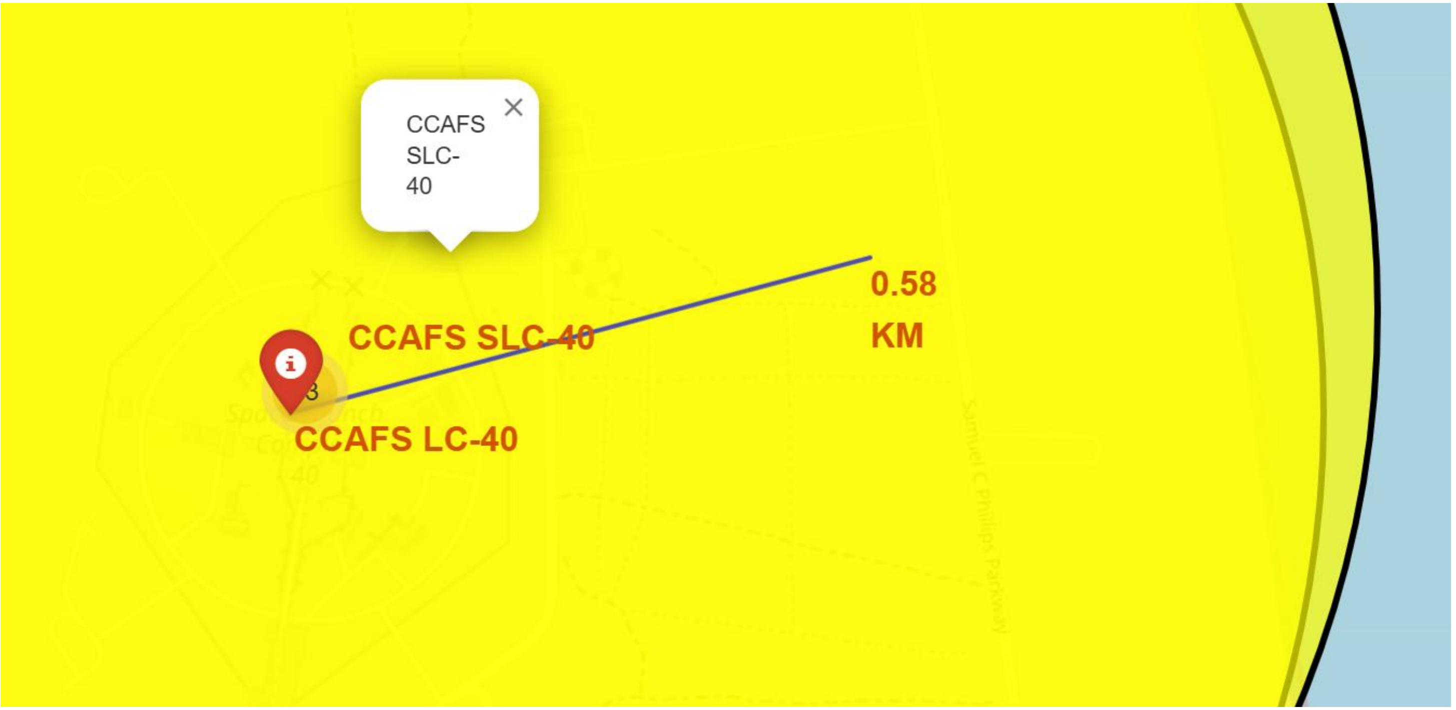
Lauch Outcomes of Each site

- **Geographical distribution**
- Markers are placed on the **actual lauch site locations** such as Kennedy Space Center (KSC), Cape Canaveral (CCAFS), and Vandenberg (VAFB).
- It highlights that most launches are clustered around Florida, with some from California.
- **Key findings**
- Success rates appear higher at certain launch sites compared to others.
- The map helps connect **geographic location** with **landing outcome patterns**, giving insight into operational performance.



Close Proximities Marking

- **Key findings**
  - Launch sites are positioned close to the **coastline**, ensuring safe rocket trajectories over the ocean.
  - Accessibility is supported by **nearby highways and railways**, which are important for transporting rocket parts and equipment.
  - The visual distances confirm that **location planning** balances logistics (roads/railways) with safety (coastline proximity).
- 👉 Overall, the map shows how **infrastructure and geography** support Falcon 9 launch operations.



Section 4

Build a Dashboard with Plotly Dash

Success Lauches for all Sites

ɔy Site

- **Key findings**
- The chart highlights which launch site contributed the **most successful launches**.
- Sites with fewer successes are also clearly visible, helping in site performance comparison.
- Overall, it provides an **at-a-glance summary** of SpaceX's operational reliability across different launch sites.

launch site with highest launch success ratio

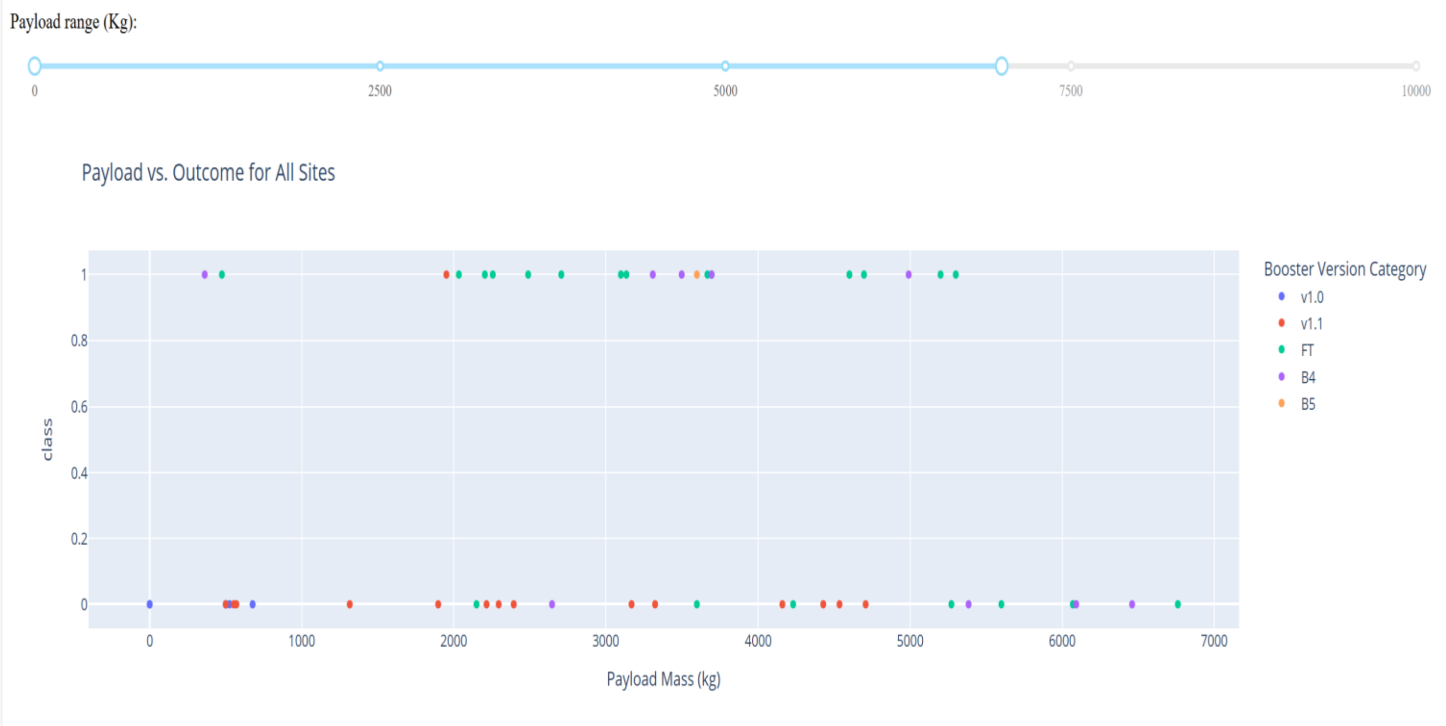
- **Key findings**
- The site demonstrates a **very high percentage of successful launches**, indicating strong operational efficiency.
- Failures are minimal, showing that this site is the most **reliable location for Falcon 9 landings**.

Payload vs. Launch Outcome scatter plot

•Launches with **medium payloads (around 2000–8000 kg)** tend to have the **highest success rate**, indicating Falcon 9's optimal payload range.

•Very **high payloads (>10,000 kg)** show slightly lower success probability.

•Some booster versions consistently achieved success across different payload ranges, highlighting **technology improvement over time**.



Section 5

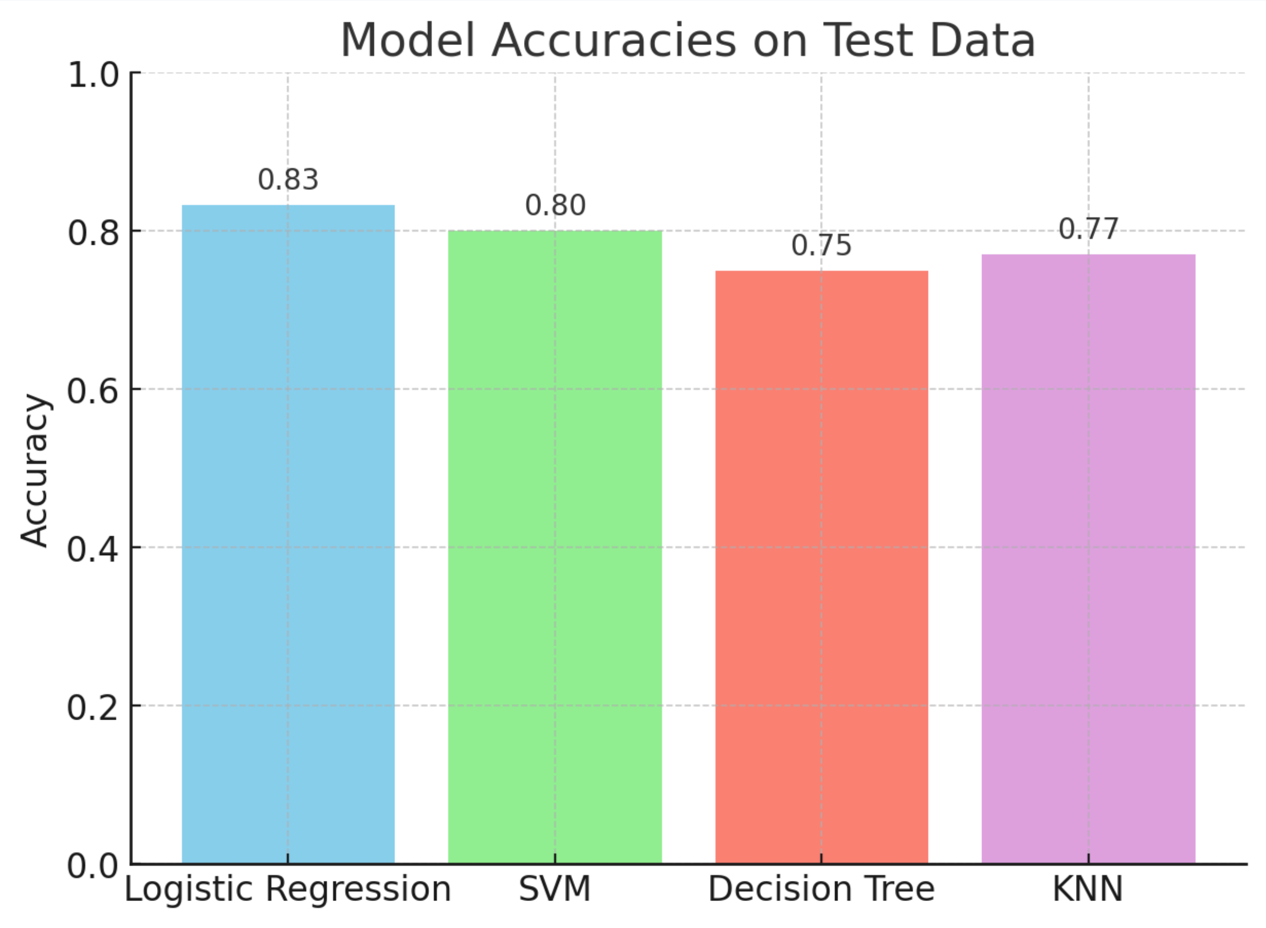
Predictive Analysis (Classification)

Classification Accuracy

•**Logistic Regression** achieved the **highest accuracy (~0.83)**, making it the best performing model for Falcon 9 landing prediction.

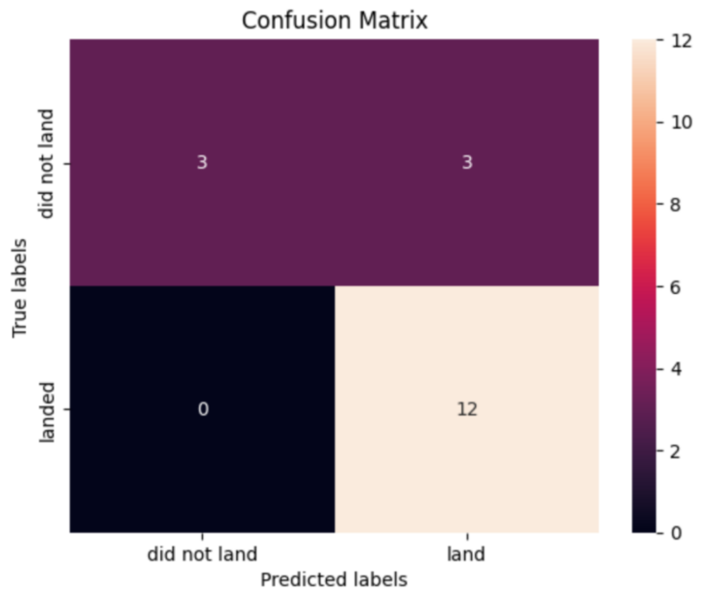
•**SVM and KNN** showed moderately good performance, but not as high as Logistic Regression.

•**Decision Tree** had the lowest accuracy, suggesting it may have overfitted or not generalized well



Confusion Matrix

- Overview:
- True Postive - 12 (True label is landed, Predicted label is also landed)
- False Postive - 3 (True label is not landed, Predicted label is landed)



Conclusions

Applied classification models with **GridSearchCV (10-fold cross-validation)** for hyperparameter tuning:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **Decision Tree**
- **K-Nearest Neighbors (KNN)**

Evaluated models using:

- **Confusion Matrix**
- **Accuracy score** on test data

◆ **Results**

Logistic Regression achieved the **highest test accuracy (~83%)**, outperforming SVM, Decision Tree, and KNN.

Decision Trees had potential but were sensitive to parameter choices.

KNN and SVM performed reasonably but not as well as Logistic Regression.

Logistic Regression provided the most reliable performance with relatively simple implementation.

Conclusions

- Throughout these labs, we developed a **machine learning pipeline** to predict whether the SpaceX Falcon 9 first stage will land successfully.
- 🔹 **Data Collection & Preprocessing**
- Gathered launch data from **SpaceX API** and **Wikipedia** using web scraping and API queries.
- Performed **data wrangling**: handled missing values, created relevant features (launch site, payload mass, booster version, orbit, etc.).
- Standardized features using **StandardScaler** to ensure all variables contributed equally.
- 🔹 **Exploratory Data Analysis (EDA)**
- Used **Pandas, Matplotlib, and Seaborn** for statistical summaries and visualizations.
- Applied **Folium maps** to visualize launch sites and proximity to railways, highways, coastlines, and cities.
- Conducted **EDA with SQL** to query success rates by orbit, launch site, and booster version.
- 🔹 **Model Building & Evaluation**
- Split data into **training and test sets**.
- .

Appendix

Python Code: Web scraping, data wrangling, SQL queries, Folium maps, Plotly Dash dashboard, ML models (Logistic Regression, SVM, Decision Tree, KNN), evaluation (accuracy, confusion matrix).

SQL Queries: Launch outcomes by site, orbit success rates, payload/booster filters.

Charts: Scatter plots, bar charts, pie charts, line charts, confusion matrices.

Notebook Outputs: Accuracy scores, best hyperparameters, model comparison results.\

Datasets: Falcon 9 datasets (dataset_part_1.csv, dataset_part_2.csv, dataset_part_3.csv).

Thank you!