IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Virginia Levy Abulafia
07 Jun 2025

Outline

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

Executive Summary

- **Goal**: Predict the outcome (success/failure) of Falcon 9 first-stage landings to estimate launch cost-effectiveness.
- **Business Relevance**: SpaceX reduces costs by reusing the first stage; forecasting landing success is critical for new entrants like SpaceX.
- **Data Sources**:
  - API (https://api.spacexdata.com/v4/launches/past) for structured launch records.
  - Web scraping from Wikipedia for additional historical data.

Executive Summary

- **Key Steps**:
  - Data wrangling and feature engineering using Panda.
  - Exploratory Data Analysis with SQL and Python.
  - Predictive modeling using Scikit-learn Pipeline and GridSearchCV.
- **Tools**: Python, Pandas, BeautifulSoup, SQLite, Plotly, Dash, Scikit-learn
- **Result**:The Final classification model accurately predicts first-stage landing success and helps support operational planning.

Introduction

Commercial Space Context

- The new space race has begun: players like Virgin Galactic, Rocket Lab, and Blue Origin are revolutionizing the industry.
- Among them, **SpaceX** stands out by drastically lowering launch costs thanks to the reuse of its Falcon 9 first stage.
- A Falcon 9 launch costs $62 million vs. $165 million for traditional providers: **reusability makes the difference.**

Project Objective

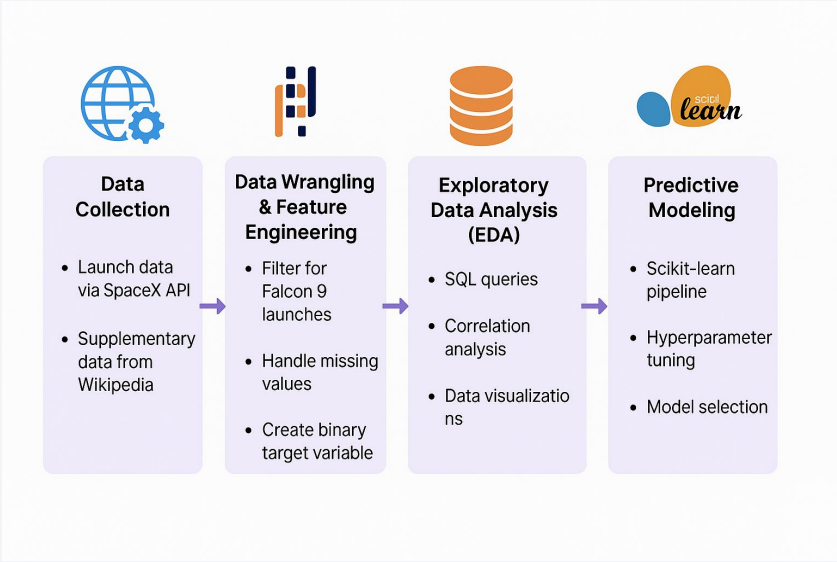
- My mission: predict whether the **Falcon 9 first stage will successfully land** after launch.
- Instead of rocket science, use **public data and machine learning** to make predictions.
- The outcome supports cost planning and strategic decisions for SpaceY, a startup aiming to compete with SpaceX.

Section 1

Methodology

Methodology

Executive Summary



1. Data Collection

- Launch data retrieved via SpaceX REST API: /v4/launches/past.
- Supplementary data scraped from Wikipedia using *BeautifulSoup* for older launches.

2. Data Wrangling & Feature Engineering

- Filtered for **Falcon 9** launches only (excluded Falcon 1).
- Used additional API endpoints (Booster, Payload, Launchpad) to enrich features.
- Replaced missing values in *PayloadMass* with column mean.
- Created binary target variable: **landing success (1)** vs **failure (0)**.

3. Exploratory Data Analysis (EDA)

- SQL queries on cleaned dataset using SQLite.
- Correlation heatmaps and pivot tables to identify key features.
- Visualized success by site, orbit, and booster version.

4. Predictive Modeling

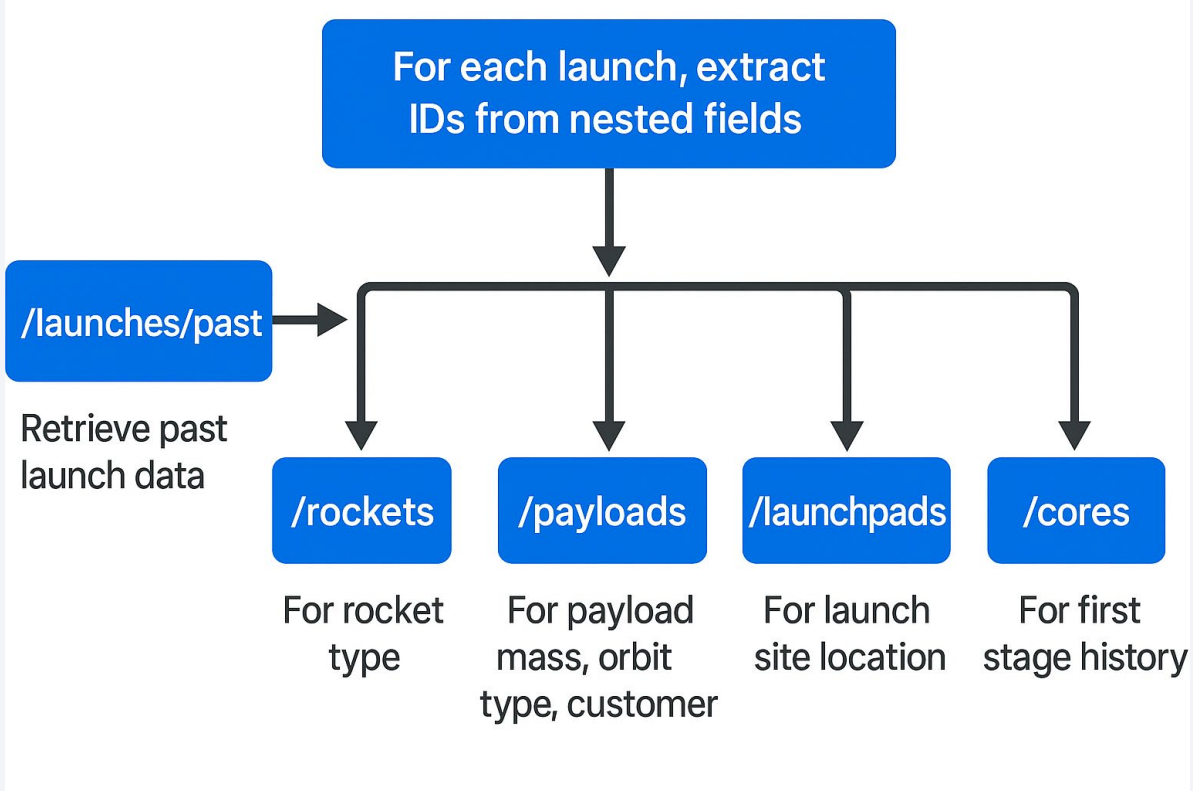
- Built a *Scikit-learn Pipeline* including preprocessing and classification steps.
- Performed hyperparameter tuning with **GridSearchCV.**
- Compared Logistic Regression, SVM, KNN, and Decision Trees.
- Selected best-performing model based on accuracy and interpretability.

Data Collection – SpaceX API

GitHub Repository

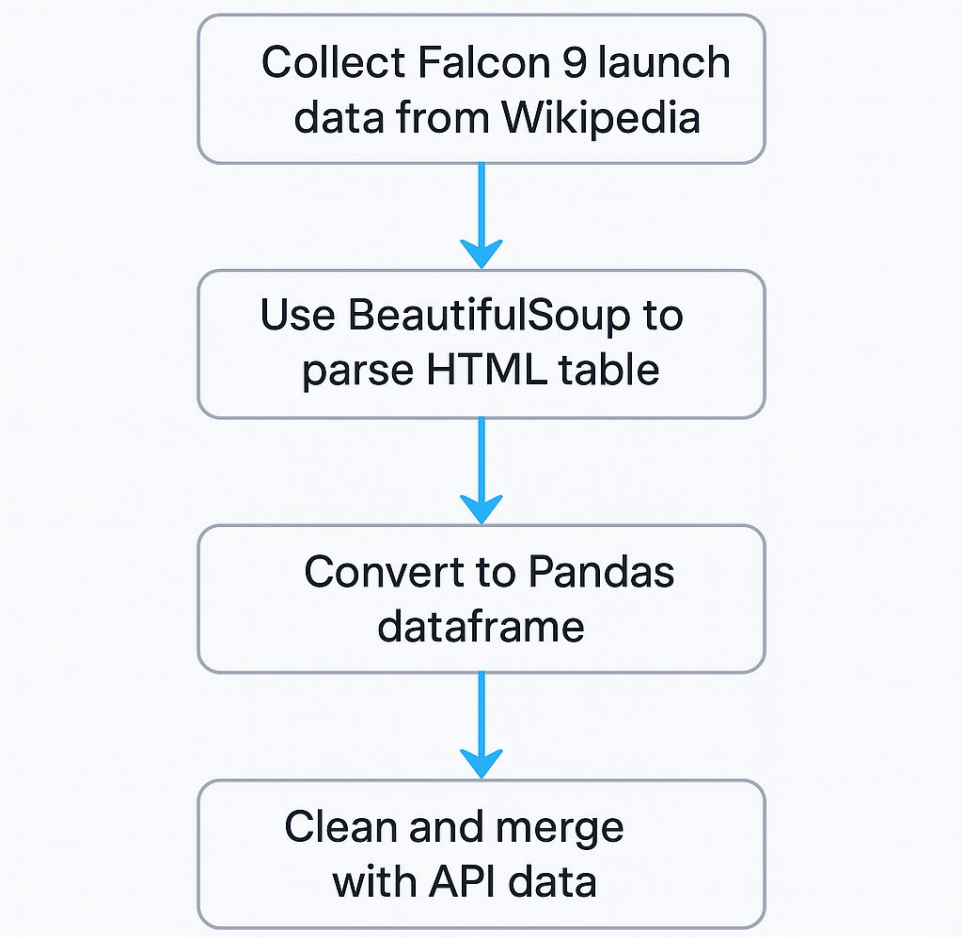
The full implementation of the SpaceX API data collection, including the completed code and output cells, is available in the notebooks folder of the repository:

🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction



Data Collection - Scraping

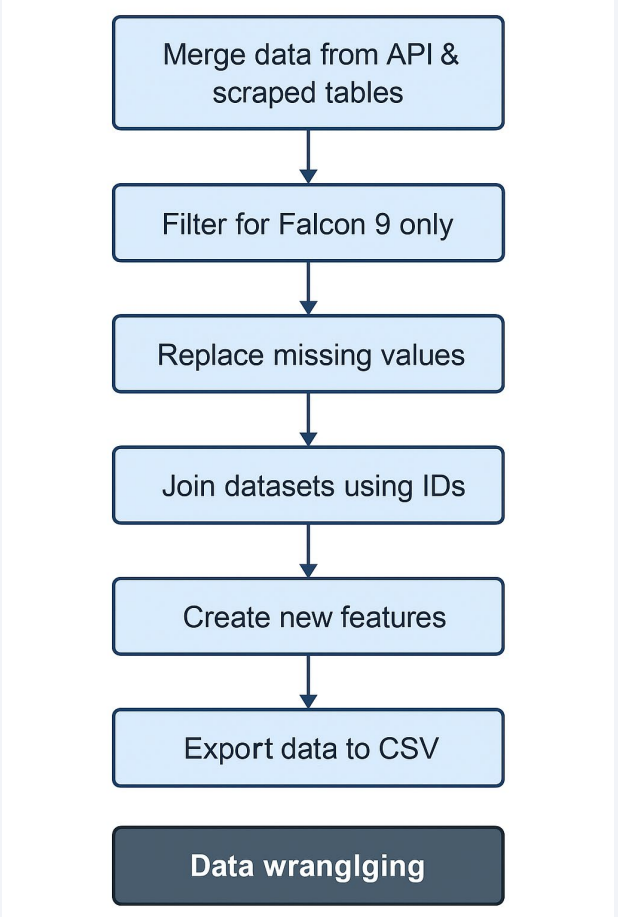
- Collected supplementary Falcon 9 launch data from Wikipedia.
- Parsed HTML tables using **BeautifulSoup.**
- Transformed scraped data into a Pandas dataframe for wrangling.
- Cleaned data and merged with API records to enrich the dataset.



GitHub Repository
The complete notebook used for web scraping is available in the notebooks folder at:
🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

Data Wrangling

- Filtered for Falcon 9 launches only
- Merged data from multiple sources (API + scraped tables)
- Replaced missing values (e.g. Payload Mass) using column mean
- Joined datasets using rocket, payloads, launchpad, and cores IDs
- Created new features (e.g. landing success class)
- Stored final dataset as CSV for EDA and modeling



GitHub Repository
All data wrangling steps are documented in the notebooks folder:
🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

EDA with Data Visualization

Charts and Why They Were Used:

- **Bar Plot: Landing Success per Orbit**
  To analyze if different mission orbits influence the success of first-stage landings.
- **Bar Plot: Landing Success per Launch Site**
  To evaluate geographic performance and identify more reliable locations.
- **Histogram: Payload Mass Distribution**
  To check for skewness and outliers that might affect model inputs.
- **Scatter Plot: Payload Mass vs. Landing Outcome**
  To explore possible correlation between payload weight and landing success.
- **Correlation Heatmap**
  To understand relationships among numerical features and guide feature selection.

GitHub Repository
The complete EDA with data visualization notebook is available at:
🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

EDA with SQL

SQL Queries Performed

- Queried landing outcomes by **booster version category** to analyze success distribution
- Counted number of **successful landings per orbit**
- Grouped data by **launch site** and **outcome** to identify patterns across locations
- Calculated **average payload mass per orbit** to examine mission profiles
- Filtered for **non-successful landings** to explore failure conditions
- Used JOIN operations to combine launch and payload tables for more context
- Created views and CASE WHEN clauses to label success/failure for binary classification

GitHub Repository
The SQL analysis is documented in the EDA with SQL notebook, available at:
🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

Build an Interactive Map with Folium

Map Objects Added:

- **Launch Site Markers**
  - Each marker represents a unique Falcon 9 launch site.
  - The pop-up tooltip displays the launch site name for quick reference.
- **Circle Markers**
  - Visualize the geographical location of the launch sites with enhanced visibility.
  - Used to emphasize site density and location relevance.

Purpose of the Map

- To provide **spatial context** for Falcon 9 launch activities
- To support **geographic pattern analysis** in conjunction with EDA
- To offer a more engaging and **interactive view** of where missions take place

GitHub Repository

The complete notebook with the Folium map is available at:

🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

Build a Dashboard with Plotly Dash

Plots and Interactions Added

- **Launch Site Dropdown Menu**
  - Enables users to select a specific launch site or view all sites combined.
  - Drives both pie chart and scatter plot interactivity.
- **Pie Chart**
  - Displays the number of successful landings per site.
  - Helps identify which sites have the highest success rate.
- **Payload Mass Range Slider**
  - Allows users to dynamically filter the scatter plot by payload weight (kg).
  - Facilitates exploration of how payload mass impacts landing success.
- **Scatter Plot**
  - Visualizes the relationship between payload mass and landing outcome.
  - Booster version color-coded for deeper analysis.

Build a Dashboard with Plotly Dash

Purpose of These Elements

- To provide an **interactive exploration tool** for launch outcomes.
- To allow **cross-filtering** by site and payload characteristics.
- To help identify trends and outliers in Falcon 9 mission data.

GitHub Repository

The completed Plotly Dash app is available in the project folder:
🔗 github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

Predictive Analysis (Classification)

Machine Learning Approach

- **Goal**: Predict binary outcome, whether the Falcon 9 first stage lands successfully (1) or not (0)

- **Pipeline Setup**:
  - Preprocessing with StandardScaler()
  - Feature selection with SelectKBest()
  - Classification model (tuned via GridSearchCV)

Predictive Analysis (Classification)

Models Evaluated:

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

Why These Models?

- Provide a range of interpretability vs. complexity Allow robust testing of both linear and non-linear decision boundaries
- Easy to tune using GridSearchCV inside a unified pipeline

Best Performing Model:

SVM with RBF kernel

- Achieved the highest validation accuracy among all tested models
- Balanced performance and generalization

Results

Model Comparison

Model	Accuracy
Logistic Regression	83.3%
SVM (RBF Kernel)	83.3%
KNN	83.3%
Decision Tree	77.8%

Validation Accuracy

- All models performed well; **SVM and Logistic Regression** showed the best generalization.
- Models were evaluated using **GridSearchCV** within a unified pipeline.
- KNN matched the performance but was less interpretable.

Confusion Matrix & Feature Insights

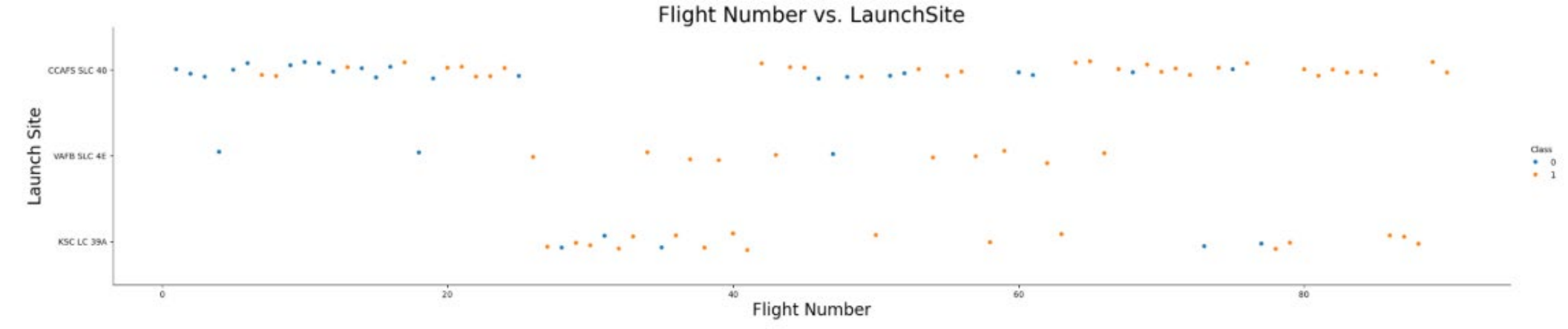
- **Confusion Matrix (Best Model: SVM)**
  - Balanced prediction between successful and failed landings
  - Low false positives, indicating reliability in success classification
- **Key Features**
  - Payload Mass (kg)
  - Orbit
  - Flight Number

These features most strongly influenced the outcome, confirming insights from EDA.

Section 2

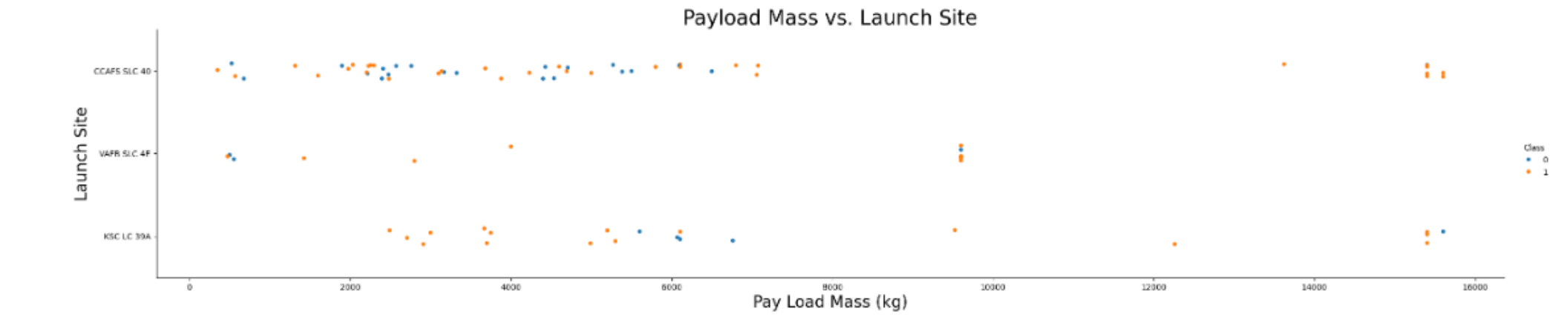
Insights drawn from EDA

Flight Number vs. Launch Site



This scatter plot shows how each launch site was used over time. Sites like **CCAFS SLC 40** were active throughout many flights, suggesting operational consistency.
Differences in landing outcomes across sites may reflect variations in infrastructure, mission profile, or strategic shifts, for example, increased use of **KSC LC 39A** may align with improved landing capabilities.
Overlaying success/failure outcomes helps highlight whether certain sites are more favorable for first-stage recovery.

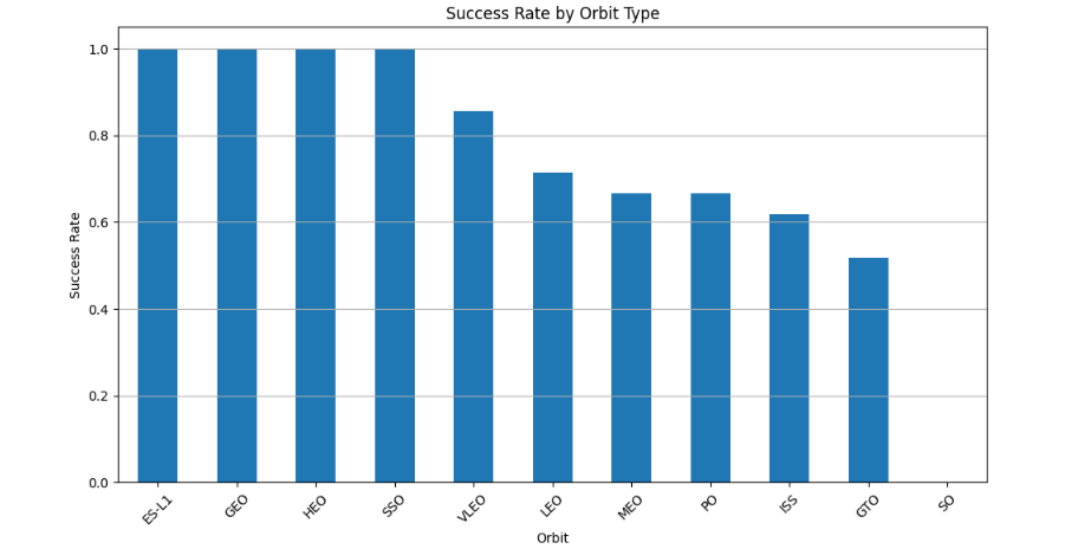
Payload vs. Launch Site



This scatter plot displays the distribution of payload masses (kg) across different launch sites. We observe that **CCAFS SLC 40** handled a broad range of payloads, while **KSC LC 39A** launched heavier missions.

Orange dots (class 1) suggest a higher success rate even for heavy payloads, especially at **KSC LC 39A**, possibly indicating advanced landing support or mission planning.

Success Rate vs. Orbit Type



This bar chart shows the landing success rate by orbit type.

Missions to **ES-L1, GEO, HEO, and SSO** had a perfect success rate, likely due to optimized mission profiles and consistent conditions.
Lower success rates for **GTO** and **SO** orbits suggest greater complexity or energy demands, potentially impacting first-stage recovery.
This insight supports the inclusion of **orbit type** as a key feature in the predictive model.

Flight Number vs. Orbit Type

This scatter plot shows how orbit types vary across flight history and how they relate to landing outcomes.

Red dots (class 0) indicate failed landings, particularly concentrated in early flights and in orbits like **GTO** and **ISS**.

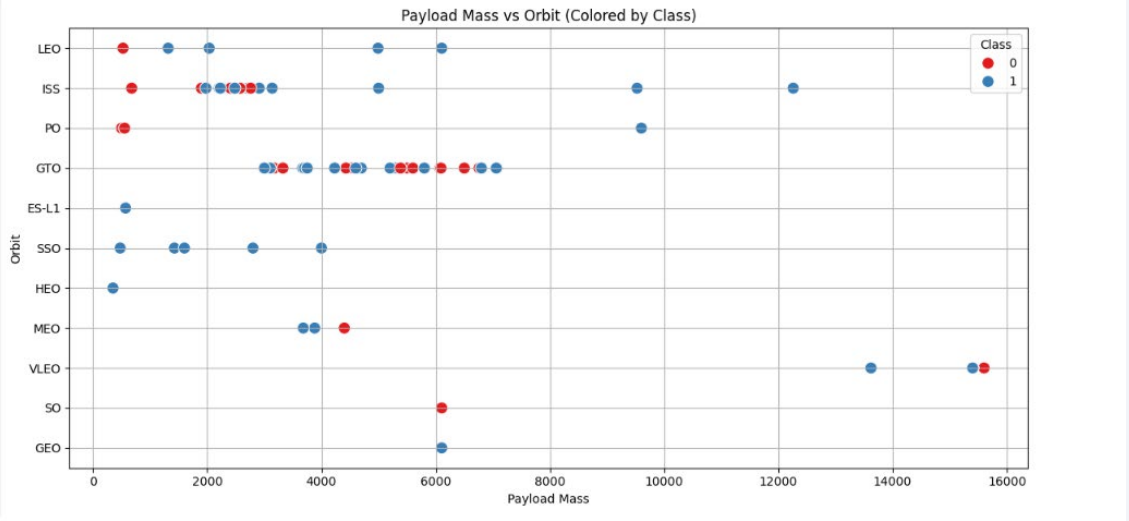
Blue dots (class 1) dominate later missions, suggesting that **technological improvements and mission selection** over time have increased success rates.

Together with the bar chart, this supports the orbit type's importance as a predictive feature, not only for outcome but also for **temporal evolution**.

Payload vs. Orbit Type

This scatter plot explores the relationship between payload mass and orbit type, colored by landing outcome.

Lower payloads (under 6,000 kg) are spread across several orbits, with mixed results.



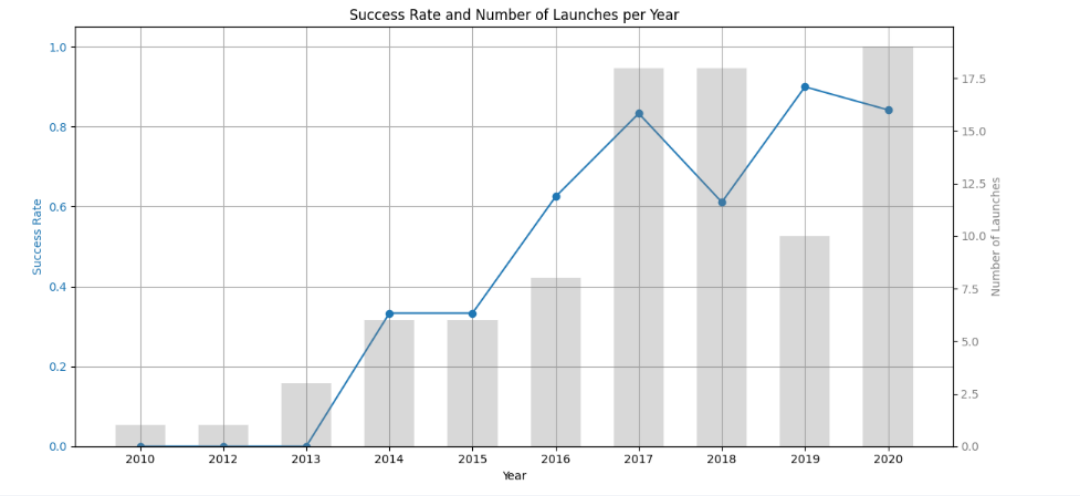
Notably, heavier payloads (above 10,000 kg) are mostly associated with **successful landings** (blue), despite their complexity, suggesting that **payload alone doesn't determine recovery failure**.

Certain orbits, like **LEO and SSO**, show consistently successful landings across various payload ranges, reinforcing their stability for recovery missions.

Launch Success Yearly Trend

This combined chart shows the **average landing success rate (line)** and **number of Falcon 9 launches (bars)** per year.

The trend highlights a clear **increase in reliability over time**, with success rates improving significantly from 2014 onwards.



Peaks in launch activity (2017, 2018, 2020) coincide with high success rates, reflecting **operational maturity** and **technological refinement**.

The data confirms that **experience and scale correlate with better landing outcomes**, supporting the time-dependent evolution of model features.

All Launch Site Names

SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;

This SQL query returns the **distinct launch site names** from the dataset.
The result identifies three unique sites used by Falcon 9:

- CCAFS LC-40
- VAFB SLC-4E
- KSC LC-39°

Note: CCAFS LC-40 appeared more than once in the raw data, but the DISTINCT clause ensured uniqueness.

Launch Site Names Begin with 'CCA'

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

This query retrieves the **first five launches** from sites whose names begin with "CCA" (i.e., **Cape Canaveral**).

The result shows that:

- All five records are from **CCAFS LC-40**
- These early missions (2010–2013) primarily targeted **LEO (ISS)**
- Although the **mission outcome** was “Success”, the **landing outcome** was either “Failure (parachute)” or “No attempt”, indicating early development stages in recovery capability

Total Payload Mass

```
SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass
FROM SPACEXTABLE
WHERE "Customer" = 'NASA (CRS)';
```

This query calculates the **total payload mass** (in kilograms) launched by Falcon 9 boosters for **NASA's Commercial Resupply Services (CRS)** missions.

- The result shows a total of **45,596 kg**, confirming NASA as one of the major contributors to Falcon 9 launch mass.
- These missions typically target the **International Space Station**, reflecting recurring cargo supply objectives.

Average Payload Mass by F9 v1.1

```
SELECT AVG("PAYLOAD_MASS__KG_") AS average_payload_mass
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

This query calculates the **average payload mass** (in kilograms) launched using the **F9 v1.1** booster version.

- The result is an average of **2,928.4 kg**, indicating that this early Falcon 9 variant was typically used for **medium-weight missions**.
- This helps contextualize the payload evolution across different booster generations.

First Successful Ground Landing Date

```
SELECT MIN("Date") AS first_successful_ground_pad_landing
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)'
```

	first_successful_ground_pad_landing
0	2015-12-22

This query finds the earliest recorded instance of a **successful ground pad landing** by applying the MIN() funciotn to the launch dates.
It filters the dataset for entries where the landing outcome was 'Success (ground pad)' returning **the first successful ground recovery date** in the dataset.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
SELECT "Booster_Version“
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
        AND PAYLOAD_MASS__KG_" > 4000
        AND "PAYLOAD_MASS__KG_" < 6000
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

This query retrieves the names of the booster versions that successfully landed on a drone ship while carrying a payload mass between 4,000 and 6,000 kg.
The result helps us understand which boosters are capable of executing successful landings under medium payload stress conditions.

Total Number of Successful and Failure Mission Outcomes

```
SELECT "Mission_Outcome", COUNT(*) AS total_missions
FROM SPACEXTABLE
GROUP BY "Mission_Outcome"
```

This query groups the dataset by Mission_Outcome and counts how many launches resulted in each outcome.

It provides a **global summary of mission success and failure**, useful to assess SpaceX's overall performance.

	Outcome	total_missions
0	Failure (in flight)	1
1	Success	99
2	Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
SELECT "Booster_Version", "PAYLOAD_MASS__KG_"

FROM SPACEXTABLE

WHERE "PAYLOAD_MASS__KG_" = (

    SELECT MAX("PAYLOAD_MASS__KG_") FROM
SPACEXTABLE
)
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

This query identifies the **booster versions** that carried the **maximum payload mass** recorded in the dataset.
By nesting a MAX() function in the WHERE clause, it filters all records to return only those matching the highest payload value.
The result shows that **multiple Falcon 9 Block 5 boosters** successfully delivered a payload of **15,600 kg**, highlighting their high performance and reliability.

2015 Launch Records

```
SELECT    substr("Date", 6, 2) AS Month,    "Booster_Version",
"Launch_Site",    "Landing_Outcome“

FROM SPACEXTABLE

WHERE "Landing_Outcome" = 'Failure (drone ship)'  AND substr("Date",
0, 5) = '2015'
```

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

This query filters the dataset to display all **drone ship landing failures** that occurred in **2015**.
It selects the **month**, **booster version**, **launch site**, and **landing outcome** for each failed attempt.
The result provides insight into the **early challenges** SpaceX faced during drone ship recoveries, offering useful context for understanding the evolution of landing success over time.

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

```
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
```

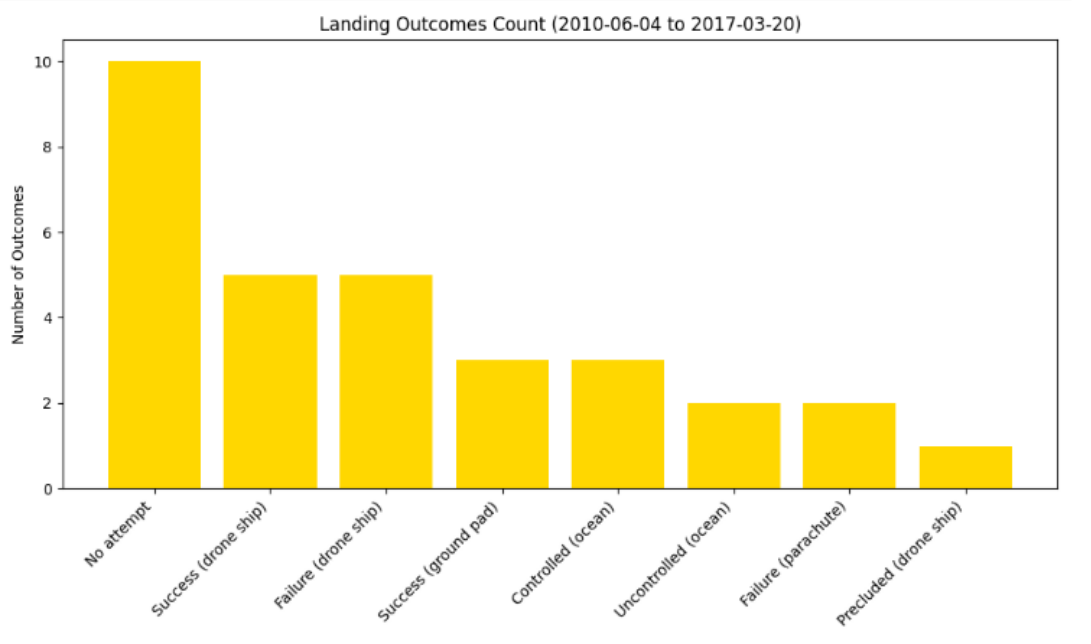
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

This query provides a clear view of how **landing strategies evolved** during SpaceX’s early years:

- **"No attempt"** was the most common outcome, especially in early launches.
- Equal counts of **successful** and **failed drone ship landings** show experimental stages.
- The presence of **ground pad successes** and **controlled ocean landings** reflects progressive refinement of recovery techniques.

This ranking helps explain the transition from experimental recovery to consistent reuse.

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



This bar chart visualizes the frequency of landing outcomes during SpaceX's early missions.
Most launches during this phase did not attempt recovery, while drone ship landings show an equal count of successes and failures.
Ground pad landings and controlled ocean outcomes appear less frequently, highlighting the **experimental nature of recovery efforts** during this period.

Section 3

Launch Sites Proximities Analysis

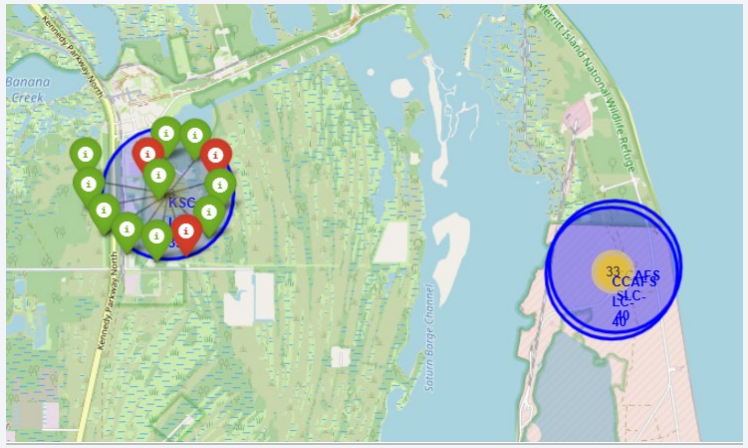
Falcon 9 Launch Sites Mapped with Folium

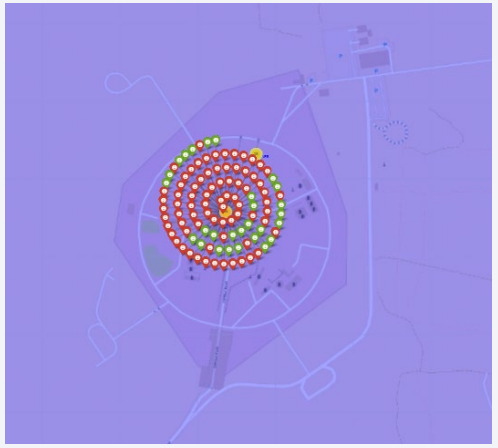
This Folium map shows the **geographical locations** of SpaceX's Falcon 9 launch sites across the U.S. Key elements include:

- **Markers** for each launch site, labeled with site names (e.g., CCAFS, KSC, VAFB)
- Sites are concentrated along the **U.S. coastlines**, reflecting proximity to safe launch corridors over the ocean
- **Florida** hosts both **KSC LC-39A** and **CCAFS SLC-40**, which are the most frequently used sites
- **VAFB SLC-4E**, located in California, supports west coast polar and sun-synchronous launches

The map helps contextualize how geography influences launch direction, orbit type, and recovery strategies.

Launch Outcomes Visualized by Site with Folium





These Folium map visualizations display **individual launch outcomes** clustered around each major SpaceX launch site.

- **Colored markers** represent mission outcomes:
  Green = Success
  Red = Failure

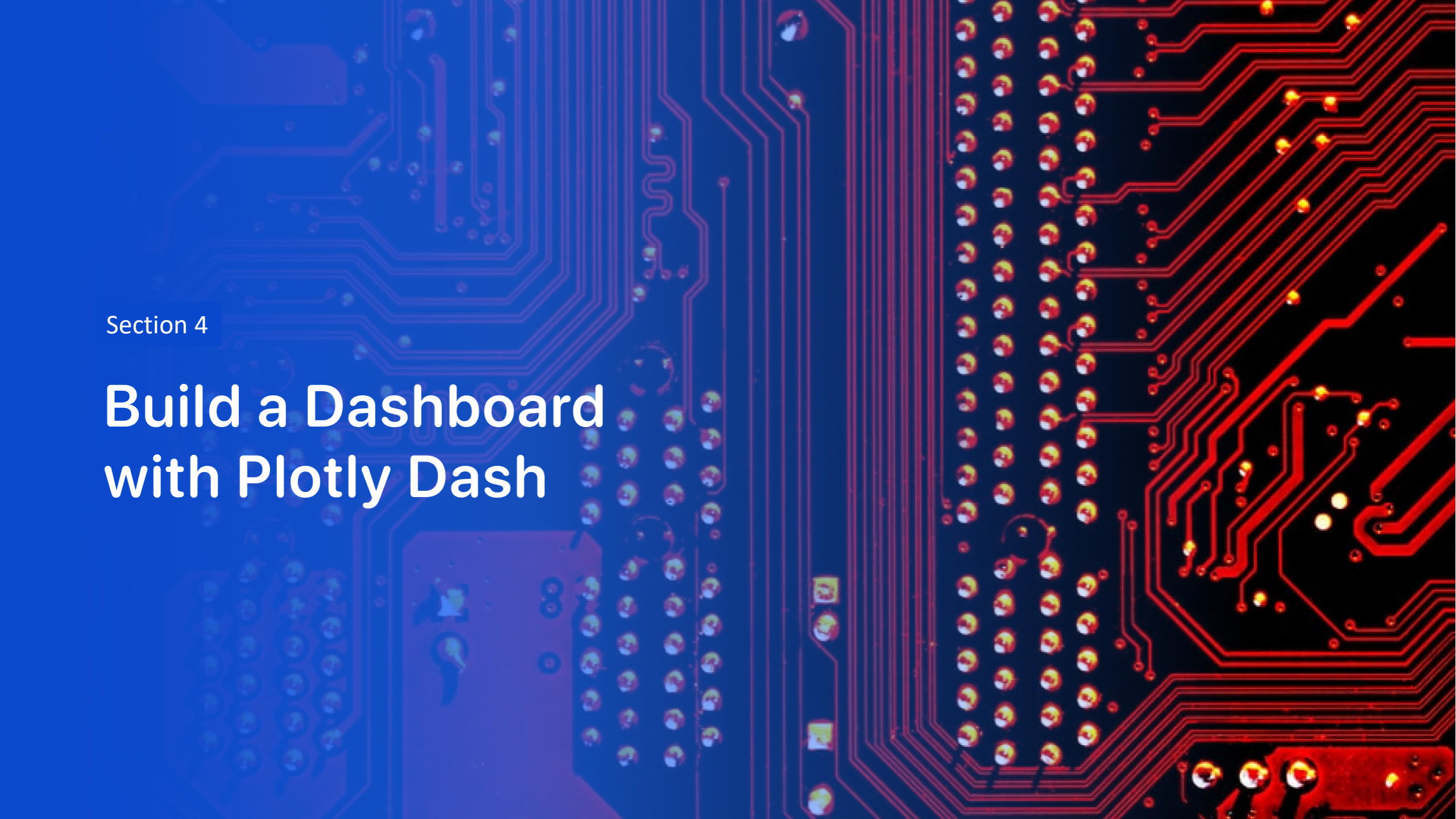
- Clusters help identify **success density** at each site:
  - **CCAFS SLC-40** and **KSC LC-39A** show a high concentration of successful landings
  - The presence of **failures in early launches** can be observed, particularly offshore
  - **Circle overlays** around markers highlight proximity and geographic context

Launch Site Proximity to Coastline – CCAFS SLC-40

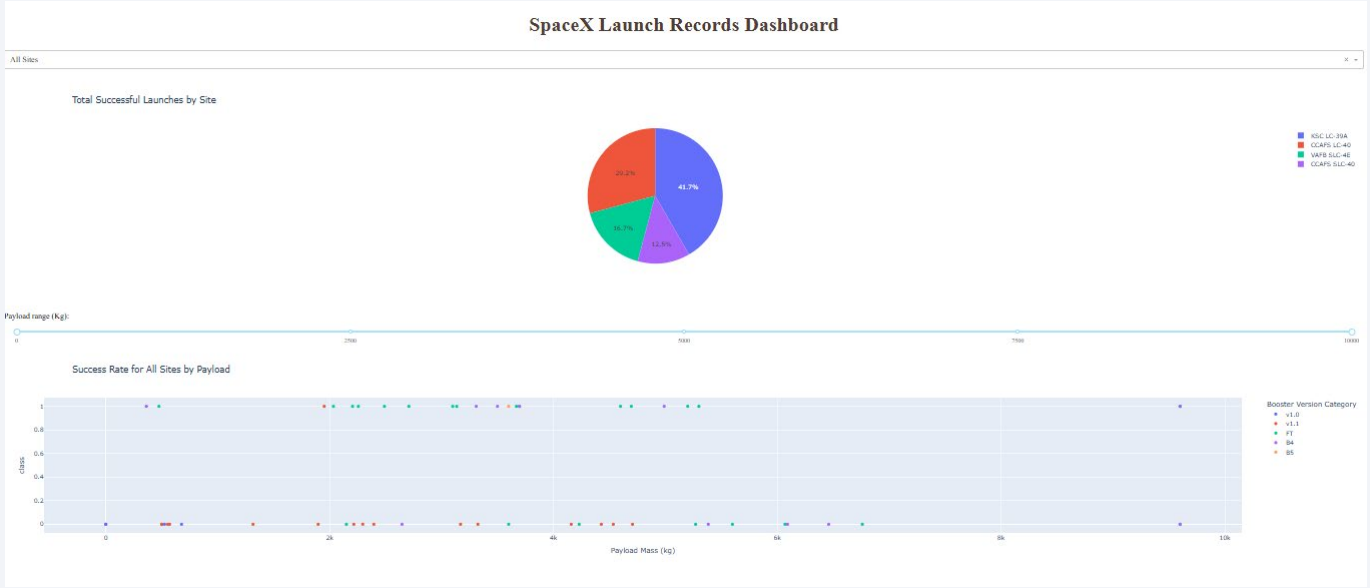
This Folium map displays the **proximity of the CCAFS SLC-40 launch pad** to the **Atlantic coastline**.

- The **blue line** indicates the shortest path between the launch pad and the coast
- The **measured distance** is approximately **0.90 km**, displayed on the line
- Marker clusters show the **number of launches from this pad**, color-coded by outcome
- The map also reveals **road access** (e.g., Samuel C Phillips Pkwy), highlighting logistical infrastructure

> This spatial insight reinforces how **close proximity to the ocean** supports safe launch and recovery operations, minimizing risk in case of failure.


Section 4
Build a Dashboard with Plotly Dash

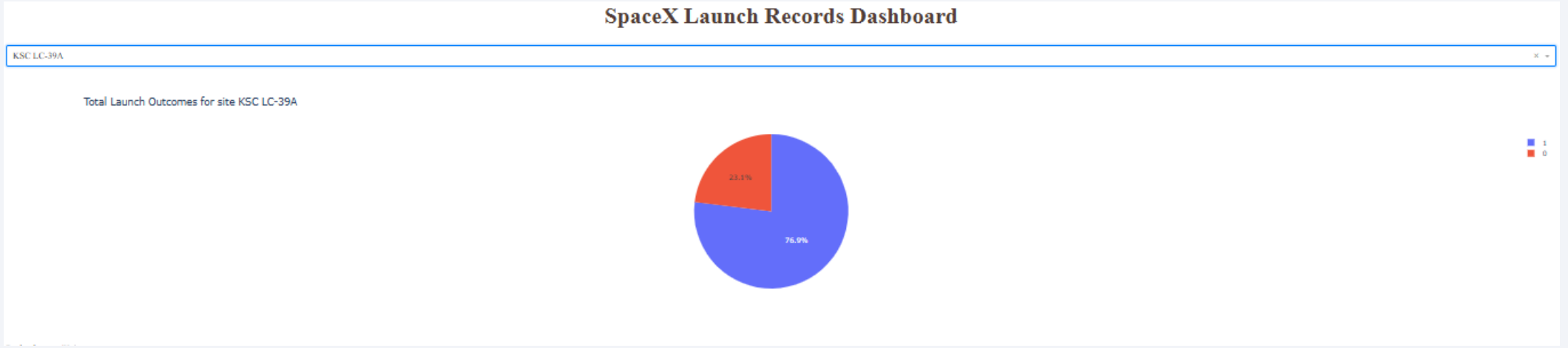
Dashboard Overview – All Launch Sites Selected



- The **pie chart** displays the **total number of successful launches per site**, helping identify the most frequently successful locations
- The **scatter plot** shows the relationship between **payload mass and landing outcome** across all sites, color-coded by booster version
- The **payload range slider** allows users to filter the scatter plot dynamically

> This view offers a global perspective, enabling high-level pattern analysis across SpaceX's entire launch history.

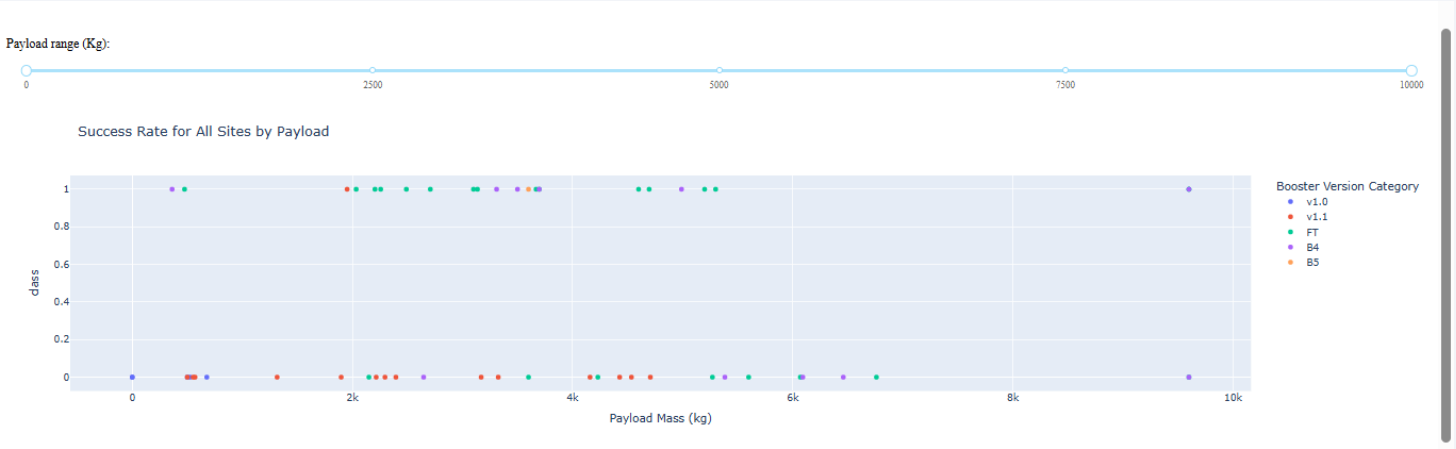
Launch Outcomes – KSC LC-39A



- This dashboard view shows the **pie chart of launch outcomes** for the selected site: **KSC LC-39A**.
- The **blue segment (76.9%)** represents **successful landings**
- The **red segment (23.1%)** corresponds to **failed landings**

> The site dropdown allows users to isolate KSC LC-39A and assess its **performance independently**: KSC LC-39A shows a **high success rate**, confirming its role as a **reliable and advanced launch site**, an insight useful for mission planning and feature engineering.

Payload vs. Landing Outcome – All Sites



This dashboard section visualizes the relationship between **payload mass (kg)** and **landing outcome** (class: 1 = success, 0 = failure), with points color-coded by **booster version**.

> This view helps identify **payload thresholds and booster effectiveness**, making it a powerful tool for feature inspection in modeling and mission optimization.

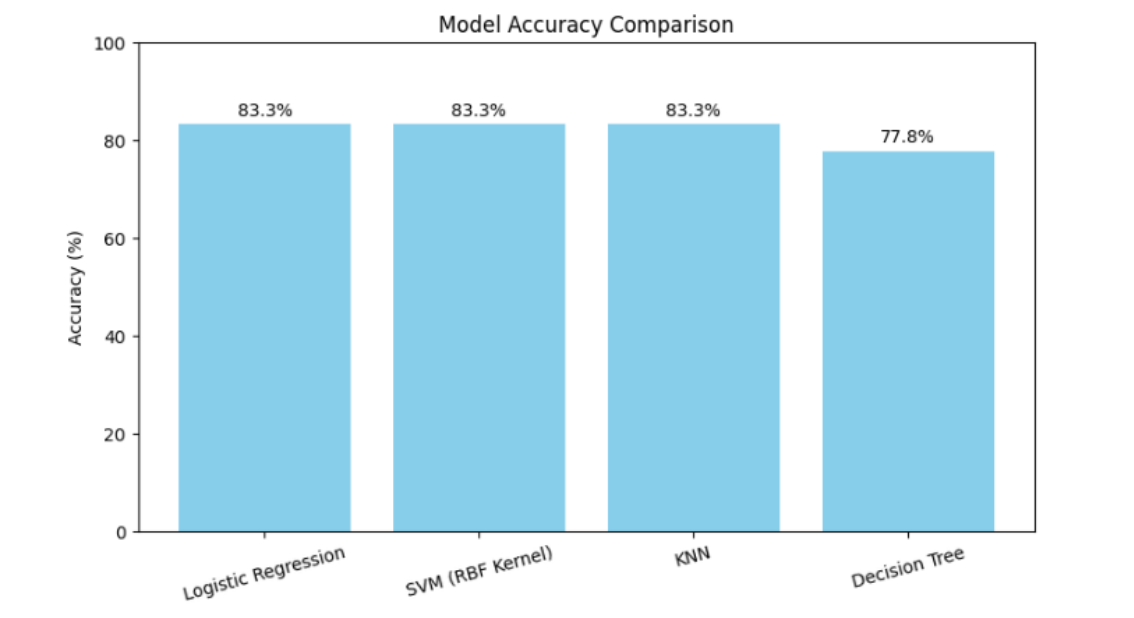
The **slider above** allows users to interactively filter payloads by mass.
The scatter plot reveals:

- Successful landings occur across all payload ranges, even beyond 6,000 kg
- Newer boosters (e.g., **B5**) are more consistently associated with successes, especially at higher payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

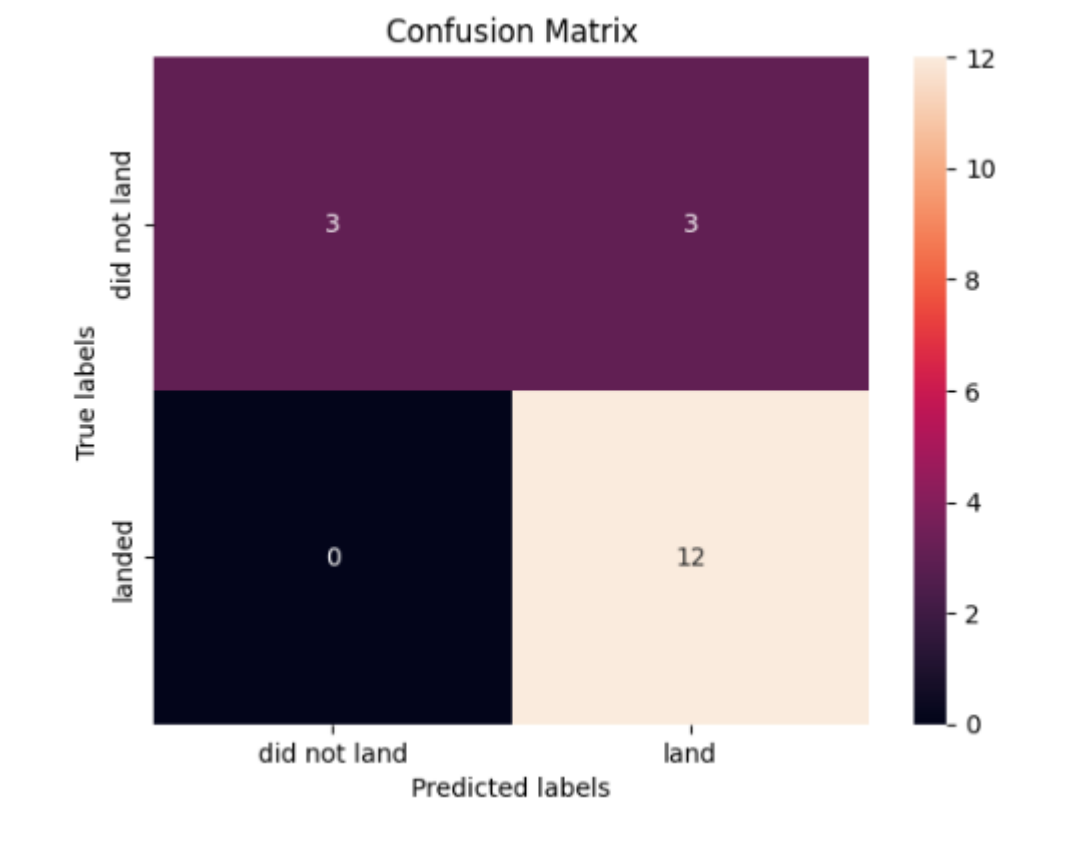


Model Accuracy Comparison

This bar chart presents the **validation accuracy** of the four classification models tested during the project.

- **Logistic Regression**, **SVM with RBF kernel**, and **K-Nearest Neighbors** all achieved an accuracy of **83.3%**, indicating strong performance and consistency on this dataset.
- The **Decision Tree** model underperformed slightly, with an accuracy of **77.8%**, possibly due to overfitting.
- These results suggest that **linear and kernel-based models** are better suited to the structure of the data.

Confusion Matrix



Confusion Matrix & SVM Choice

This confusion matrix visualizes the predictions of the **best-performing model (SVM with RBF kernel)** on the test set.

- The model correctly predicted **12 out of 12 successful landings**
- It correctly predicted **3 out of 6 failed landings**, with 3 false positives
- The overall **test accuracy is 83.3%**, matching other models like Logistic Regression and KNN

Conclusions

Despite similar accuracy scores, **SVM was chosen** as the final model because:

- **Better generalization**
  SVM with RBF kernel can capture **non-linear decision boundaries**, making it more robust to variations in data compared to simpler models like KNN or Decision Tree.
- **Lower risk of overfitting**
  Decision Trees can overfit on small datasets, while SVM finds a **max-margin hyperplane** that balances bias and variance.
- **Consistent performance**
  SVM performed consistently during **cross-validation (GridSearchCV)** and had no false negatives (it never missed a successful landing).
- **Interpretability in this context**
  While Logistic Regression is more interpretable, SVM is better suited for complex relationship and the confusion matrix confirms that its **errors are limited to false positives**, which may be more acceptable in this application.

Appendix

Data Assets - Sources and Processing Summary

The project leverages two main sources of data:

- **1. SpaceX REST API**
  Data was retrieved from api.spacexdata.com/v4/launches/past, returning a JSON structure containing:
  - Launch specifications
  - Rocket and booster IDs
  - Payload data
  - Landing outcomes

  Additional information was gathered using nested endpoints such as /rockets, /cores, /payloads, and /launchpads, using these IDs.
- **2. Web Scraping (Wikipedia)**
  Supplementary launch records were scraped using BeautifulSoup from Wikipedia launch tables, then parsed and converted into Pandas DataFrames.

Appendix

Data Preparation Steps Included:

- Normalizing JSON responses with json_normalize()
- Filtering to include only **Falcon 9** launches
- Handling **null values**, especially in PayloadMass, by imputing with the column mean
- One-hot encoding categorical features (e.g., orbit, launch pad)
- Merging API and scraped data into a single, cleaned dataset ready for modeling

SQL Queries

- SELECT DISTINCT Launch_Site ...
- SELECT SUM(PAYLOAD_MASS__KG_) WHERE Customer = 'NASA (CRS)'
- SELECT MIN(Date) WHERE Landing_Outcome = 'Success (ground pad)'
- All grouped aggregations and filters (Mission Outcomes, Yearly Success, etc.)

Appendix

Python Snippets

- **Data Wrangling**: payload cleaning, one-hot encoding, merging API + scraped data
- **ML Pipeline**: StandardScaler → SelectKBest → GridSearchCV
- **Plotting**: seaborn heatmaps, matplotlib bar charts, Plotly Dash callbacks

Notebook Outputs

- Confusion matrix (SVM)
- Classification accuracy chart
- Folium map screenshots

Repository

GitHub with full project:
github.com/VirginiaYonit/Falcon-9-First-Stage-Landing-Prediction

Thank you!