IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

George Patsias

28-09-2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Objective

Estimate landing probability to support pricing / bid risk.

Top Findings

Success improved post-2017; payload / orbit matter; site differences narrowed.

Headline Metric

Model achieves ~83% accuracy on held-out data.

Business Impact

Enables sharper pricing while reducing downside risk on challenging profiles.

So What?

Supports risk-aware, competitive proposals.

Introduction

Why: First-stage landing underpins SpaceX's cost advantage; forecasting success supports better pricing and risk.

Questions:

1. How has success changed over time/sites?
2. Which factors matter most?
3. Can we predict success reliably?

Scope: Falcon 9 orbital launches, 2010–2020; focus on pre-launch factors (orbit, payload, site, reuse).

Deliverables: Clear EDA, interactive map/dashboard, and a predictive baseline for landing probability.

Data Collection Methodology

Sources

SpaceX REST API (launches/rockets/payloads) + Wikipedia pages (launch sites/orbit context).

Scope

Falcon 9 orbital launches, 2010–2020; exclude Falcon 1 and tests without usable metadata.

Access Pattern

requests → JSON → pandas.json_normalize → tabular; Wikipedia via requests + BeautifulSoup.

Keys Used to Join

flight_number, rocket.id, payloads[].id, launchpad.id.

Quality Checks

Record counts by year/site, null-rate audit per field, cross-check site/orbit text vs API.

Reproducibility

Parameterized notebooks + cached raw JSON; deterministic seeds and version-pinned libs.

Data Wrangling & Feature Engineering Methodology

Cleaning: Drop Falcon 1; coerce types (dates → datetime, masses → numeric); trim/standardize categorical text.

Target: Class ∈ {0,1} (landing outcome consolidated from outcome strings).

Core Features: Orbit, LaunchSite, PayloadMass, FlightNumber, Flights, GridFins, Legs, Reused, ReusedCount, LandingPad, Block, Year.

Derivations: Orbit grouping (e.g., LEO/GTO/ISS), payload bands, year/month, reuse indicators from booster history.

Missing & Outliers: Impute rare categorical levels as “Other”; keep payload outliers (true missions) but cap for EDA visuals.

Model Matrix: One-hot encode (Orbit, LaunchSite, LandingPad, Serial); scale numerics for LR/SVM; export dataset_part_3.csv.

EDA Methodology

- Questions framed (trends, drivers, site/orbit differences)
- Groupby/pivot/value_counts + SQL cross-checks
- Key visuals (scatter, bar, yearly trend)
- Minimal imputation; outliers kept (cap only for display)
- Reproducibility notes (notebook sections, saved figures)

Interactive Visual Analytics Methodology

Validation: interactive counts align with static EDA

Tools: Folium (MarkerCluster) + Plotly Dash

Inputs: Launch Site dropdown, Payload Mass slider

Outputs: Success pie (by site), Payload vs Success scatter (orbit hue)

Callbacks: inputs → filtered data → figures; handles empty states

UX: clear titles/legends, hover, responsive layout

Predictive Analysis Methodology

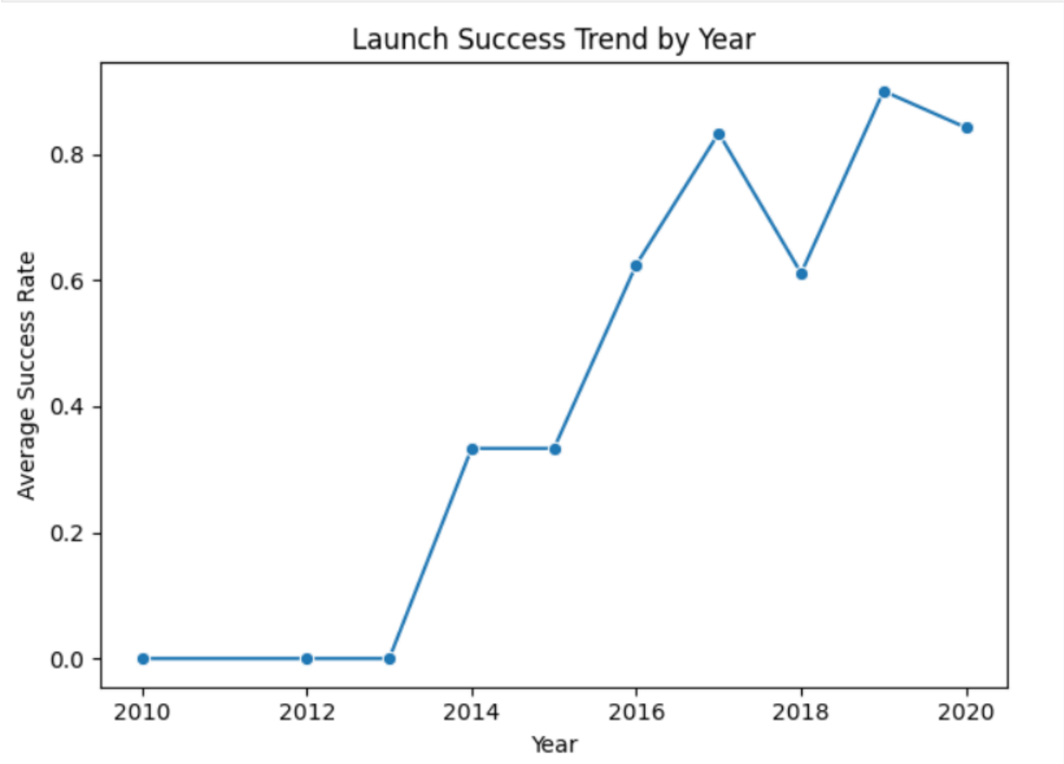
- **Target and Features** (landing success; payload, orbit, site, reuse, etc.)
- **Train/Test Split** (80/20, stratified, fixed seed)
- **Preprocessing** (standardize numerics for LR/SVM; one-hot categoricals)
- **Models Compared** (LogReg, SVM, Decision Tree, k-NN)
- **Hyperparameter Tuning** with GridSearchCV (cv=10)
- **Evaluation** (CV accuracy, test accuracy, confusion matrix)
- **Sanity Checks** (CV vs test consistency; feature effects)

EDA with Data Visualisation Results 1

Launch Success Trend by Year

Key Takeaway: Success rates climbed sharply after 2017, stabilizing around 80–90% by 2019–2020.

Risk dropped after 2017; newer missions priced confidently.



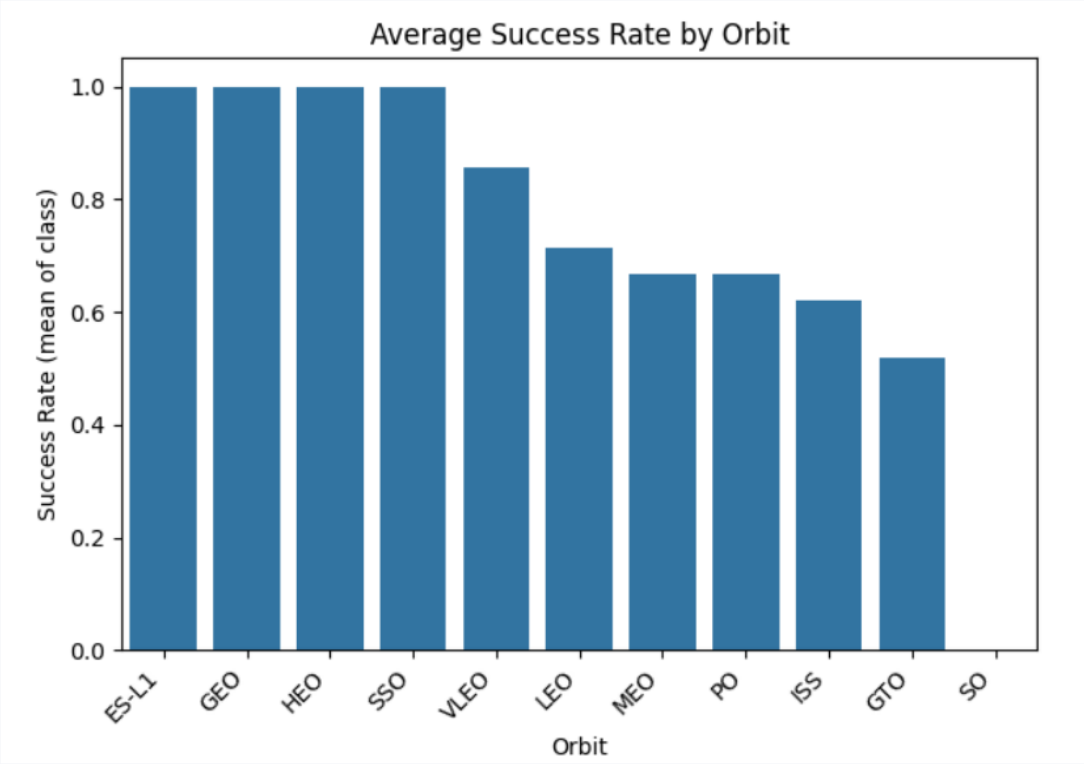
Average success rate by year—clear post-2017 inflection.

EDA with Data Visualisation Results 2

Success Rate by Orbit (mission profile risk)

Key Takeaway: Risk varies by orbit: GTO and ISS historically lower; ES-L1/GEO/HEO near perfect in this sample.

Risk dropped after 2017; newer missions priced confidently.



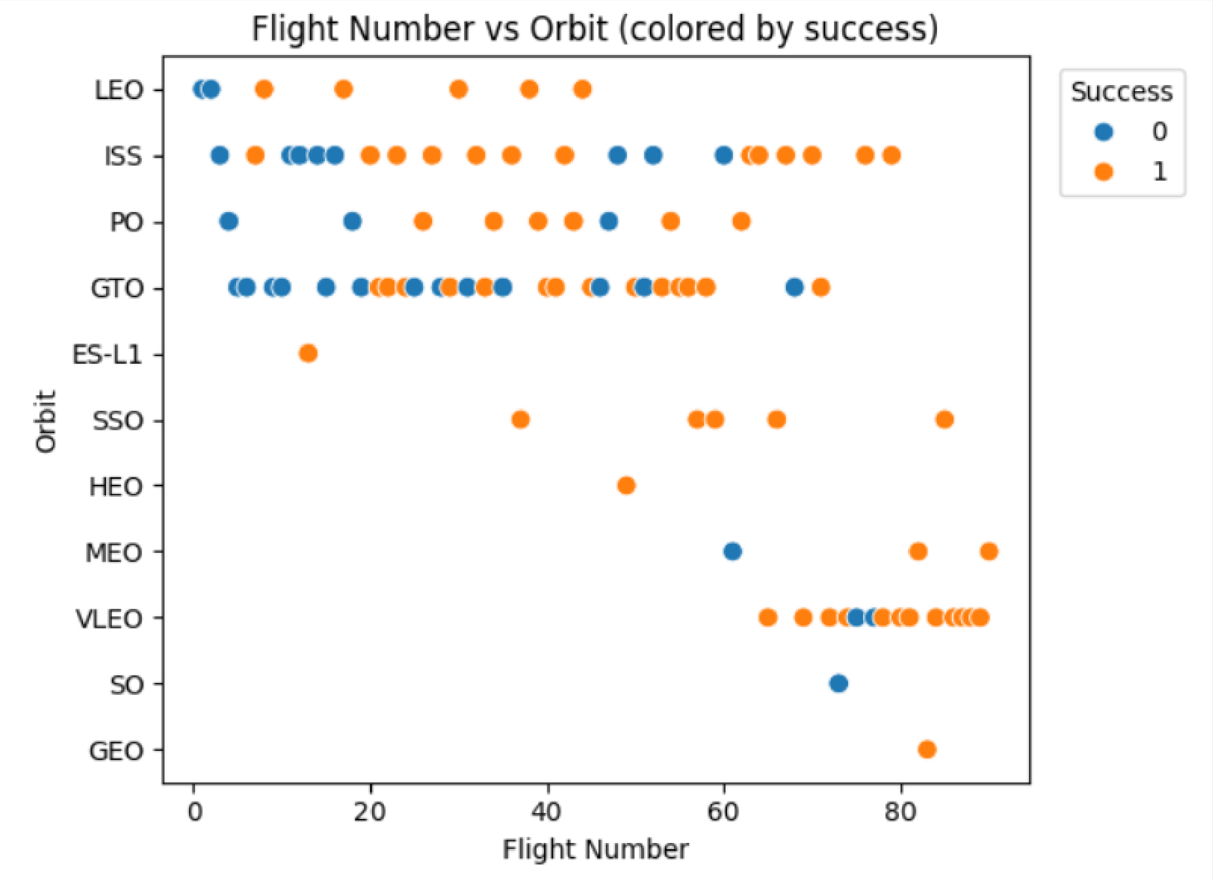
Mean success rate by orbit - mission profile matters

EDA with Data Visualisation Results 3

Payload Mass vs Orbit (colored by outcome)

Key Takeaway: Heavier payloads in tougher orbits (e.g., GTO/VLEO bands here) correlate with lower success.

Risk dropped after 2017; newer missions priced confidently.



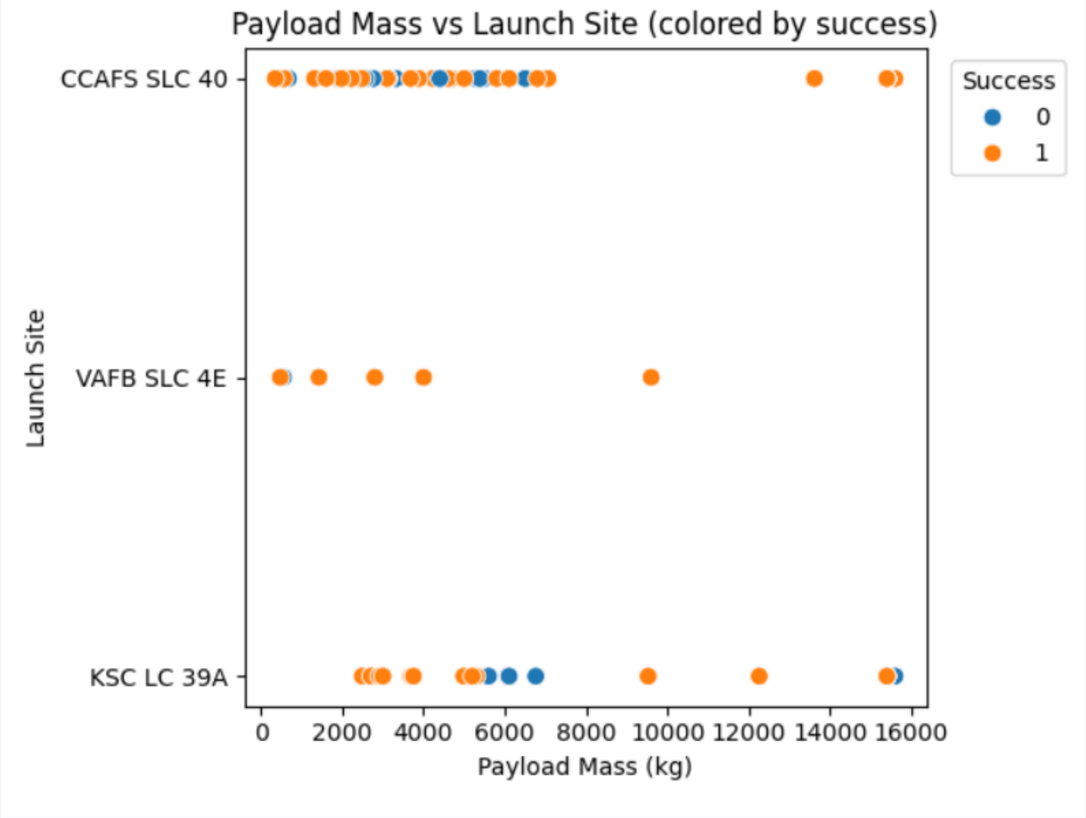
Within-orbit weight pockets show outcome shifts.

EDA with Data Visualisation Results 4

Site Patterns: Volume vs. Payload Mix

Key Takeaway: SSite differences narrow over time; CCAFS SLC-40 carries the most volume; KSC handles heavier payloads with mixed outcomes.

Mix > site: 40 volume, 39A heavy, 4E mid/few



Payload mix by site explains outcome patterns

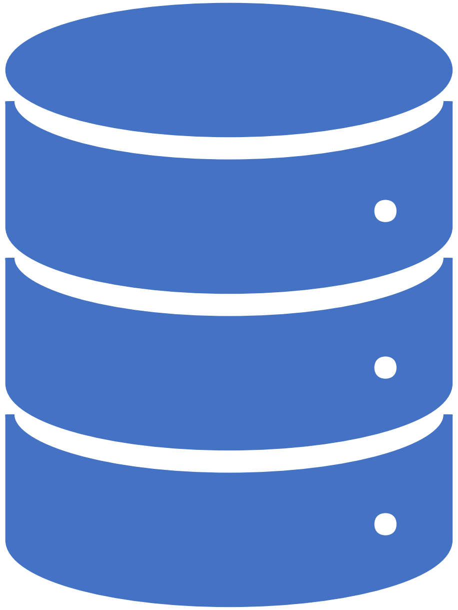
EDA with SQL Results 1

SQL Data Model & Staging

How many records did we analyse?

```
SELECT COUNT(*) AS n_rows,
       MIN(date) AS min_date,
       MAX(date) AS max_date
FROM launches;
```

We analyse N flights, from min_date to max_date.



EDA with SQL Results 2

Data Quality Checks

Basic Data Quality (SQL)

```
SELECT
    SUM(CASE WHEN orbit IS NULL THEN 1 ELSE 0 END) AS null_orbit,
    SUM(CASE WHEN launch_site IS NULL THEN 1 ELSE 0 END) AS null_site,
    SUM(CASE WHEN payload_mass_kg IS NULL THEN 1 ELSE 0 END) AS null_payload
FROM launches;

SELECT flight_number, COUNT(*)
FROM launches
GROUP BY flight_number
HAVING COUNT(*) > 1;
```

No/limited nulls; no duplicate flight numbers found (or removed X duplicates).

EDA with SQL Results 3

Yearly Launches & Success Rate

Yearly Activity & Success (SQL aggregation)

```
SELECT year,
    COUNT(*) AS launches,
    SUM(class) * 1.0 / COUNT(*) AS success_rate
FROM launches
GROUP BY year
ORDER BY year;
```

Success rose sharply after 2017, reaching ~0.8–0.9

EDA with SQL Results 4

Impact of Reuse

Reuse & Learning Effects

```
SELECT reused,
    reusedcount,
    COUNT(*) AS launches,
    ROUND(AVG(class), 3) AS success_rate
FROM launches
GROUP BY reused, reusedcount
ORDER BY reused, reusedcount;
```

	Reused	ReusedCount	launches	success_rate
0	0	0	30	0.267
1	0	1	14	1.000
2	0	2	3	1.000
3	0	3	3	1.000
4	0	4	1	1.000
5	0	5	2	1.000
6	1	1	10	0.600
7	1	2	6	0.833
8	1	3	9	0.778
9	1	4	3	0.667
10	1	5	9	1.000

Higher reuse counts tend to track with higher success - evidence of learning & hardware maturity.

EDA with SQL Results 5

Yearly Launches & Success Rate (SQL)

Falcon 9 Landings: Scale & Reliability (2010–2020)

```
SELECT year,
        COUNT(*)  AS launches,
        ROUND(SUM(class) * 1.0 / COUNT(*), 3) AS success_rate
FROM launches
GROUP BY year
ORDER BY year;
```

- **Scope: 101 Falcon 9 launches analyzed (2010–2020).**
- **Cadence: Annual launches grew 1 → 19 (~19× in a decade).**
- **Reliability: Landing success rose 0.00 → 0.84, peaking at 0.90 in 2019.**

	year	launches	success_rate
0	2010	1	0.000
1	2012	1	0.000
2	2013	3	0.000
3	2014	6	0.333
4	2015	6	0.333
5	2016	8	0.625
6	2017	18	0.833
7	2018	18	0.611
8	2019	10	0.900
9	2020	19	0.842

Year vs. Success Rate)

EDA with SQL Results 6

Heaviest Failures (Top 10)

Where failures were most costly (mass proxy)

```
SELECT date, launch_site, orbit, payload_mass_kg
FROM launches
WHERE class = 0
ORDER BY payload_mass_kg DESC
LIMIT 10;
```

	year	launches	success_rate	avg_payload_kg
0	2010	1	0.000	6104.959412
1	2012	1	0.000	525.000000
2	2013	3	0.000	1449.000000
3	2014	6	0.333	3019.333333
4	2015	6	0.333	2346.833333
5	2016	8	0.625	3639.125000
6	2017	18	0.833	5365.719967
7	2018	18	0.611	4832.767190
8	2019	10	0.900	7551.370000
9	2020	19	0.842	11477.522043

A few heavy-payload failures cluster in GTO/upper bands.

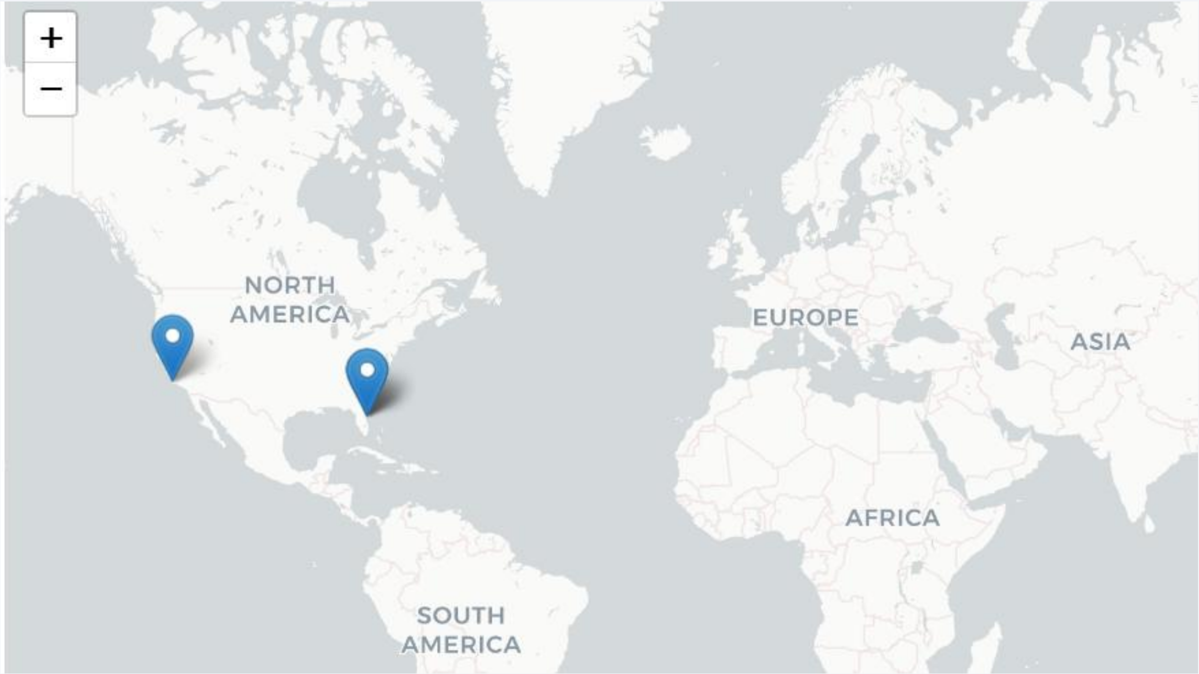
Interactive Map (Folium) - Code & Output

```
import folium, pandas as pd

sites = [
    ("SLC-40", 28.561857, -80.577366),
    ("LC-39A", 28.608389, -80.604333),
    ("SLC-4E", 34.632834, -120.610746),
]
df = pd.DataFrame(sites,
columns=["site","lat","lon"])

m = folium.Map(location=[30,-95],
zoom_start=4, tiles="CartoDB positron")
for _, r in df.iterrows():
    folium.Marker([r.lat, r.lon],
tooltip=r.site).add_to(m)

m.save("interactive_launch_map.html")
```



Launch pad map: SLC-40/LC-39A and SLC-4E plotted; baseline for orbit/payload layers.

Predictive Analysis Results Overview

Objective: predict booster landing; compare Logistic Regression, SVM, Decision Tree, and KNN.

Test set size: 18

Predictive Analysis Summary

Model Comparison - Key Takeaways

Top Accuracy: Logistic Regression & SVM = 0.83 (15/18 correct).

Both prioritize recall for 'landed' (Recall=1.00) but low specificity (0.50) → more false alarms.
Decision Tree & KNN are more balanced (Specificity=0.67; Recall=0.83) but lower accuracy (0.78).

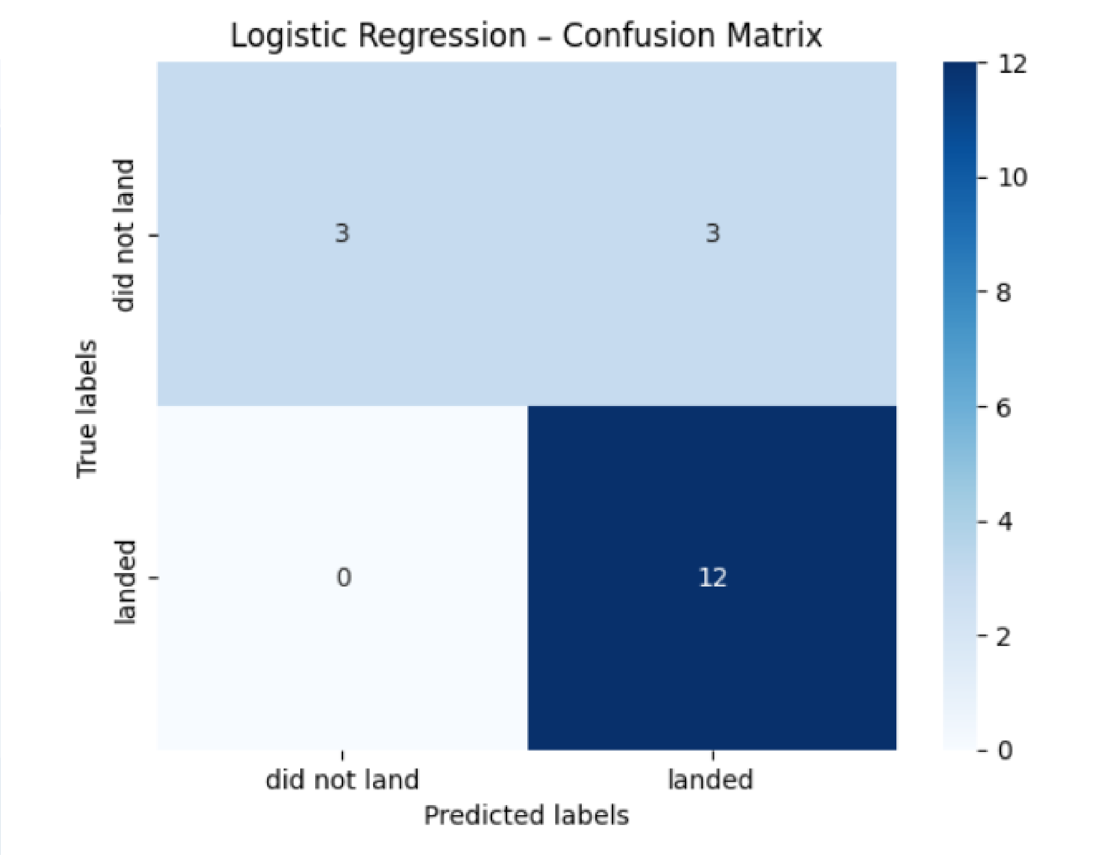
Business Fit: if missing a landing is costly, prefer LR/SVM; if false alarms are costly, prefer Tree/KNN or tune threshold.

Next: threshold tuning, class weights, calibration, and feature enrichment (orbit × payload, site, year).

Predictive Analysis Results 1

Logistic Regression - Confusion Matrix

- **Key Metrics**
  - Accuracy: 0.83
  - Precision (landed): 0.80
  - Recall (landed): 1.00
  - Specificity (did not land): 0.50
  - F1 (landed): 0.89

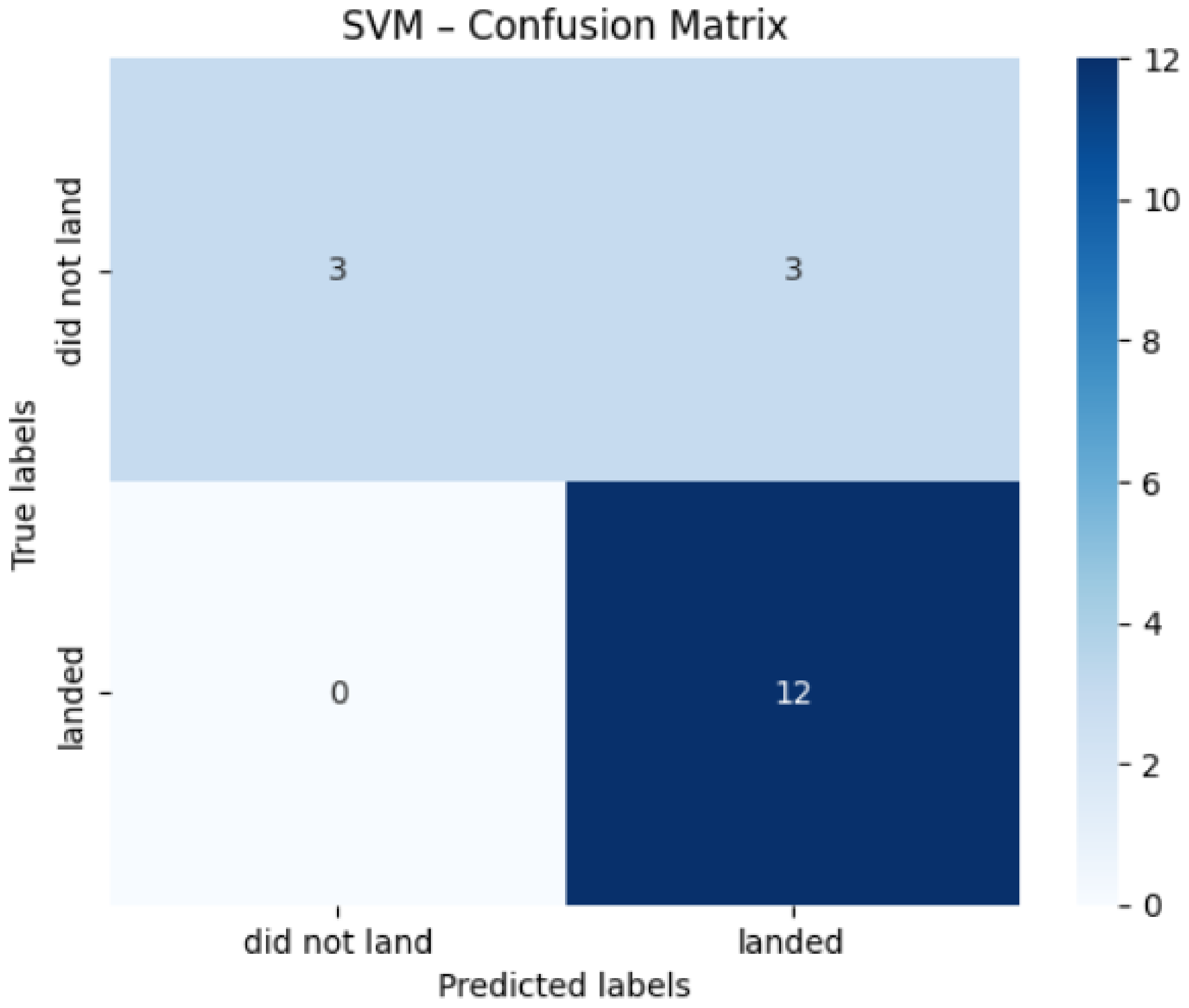


Notes: Zero missed landings (FN=0); 3 false positives on 'landed' (over-optimistic).

Predictive Analysis Results 2

SVM - Confusion Matrix

- **Key Metrics**
  - Accuracy: 0.83
  - Precision (landed): 0.80
  - Recall (landed): 1.00
  - Specificity: 0.50
  - F1 (landed): 0.89

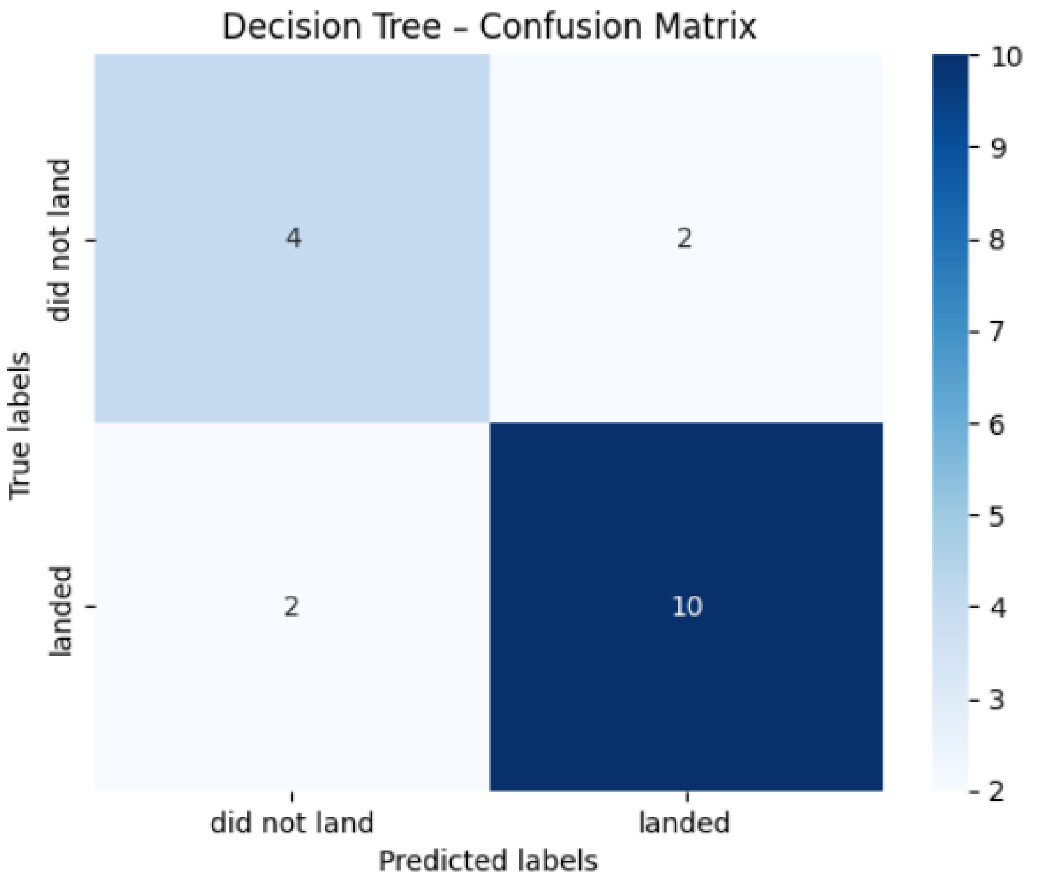


Notes: Pattern mirrors Logistic Regression; strong recall, weaker negative class discrimination.

Predictive Analysis Results 3

Decision Tree - Confusion Matrix

- **Key Metrics**
  - Accuracy: 0.78
  - Precision (landed): 0.83
  - Recall (landed): 0.83
  - Specificity: 0.67
  - F1 (landed): 0.83

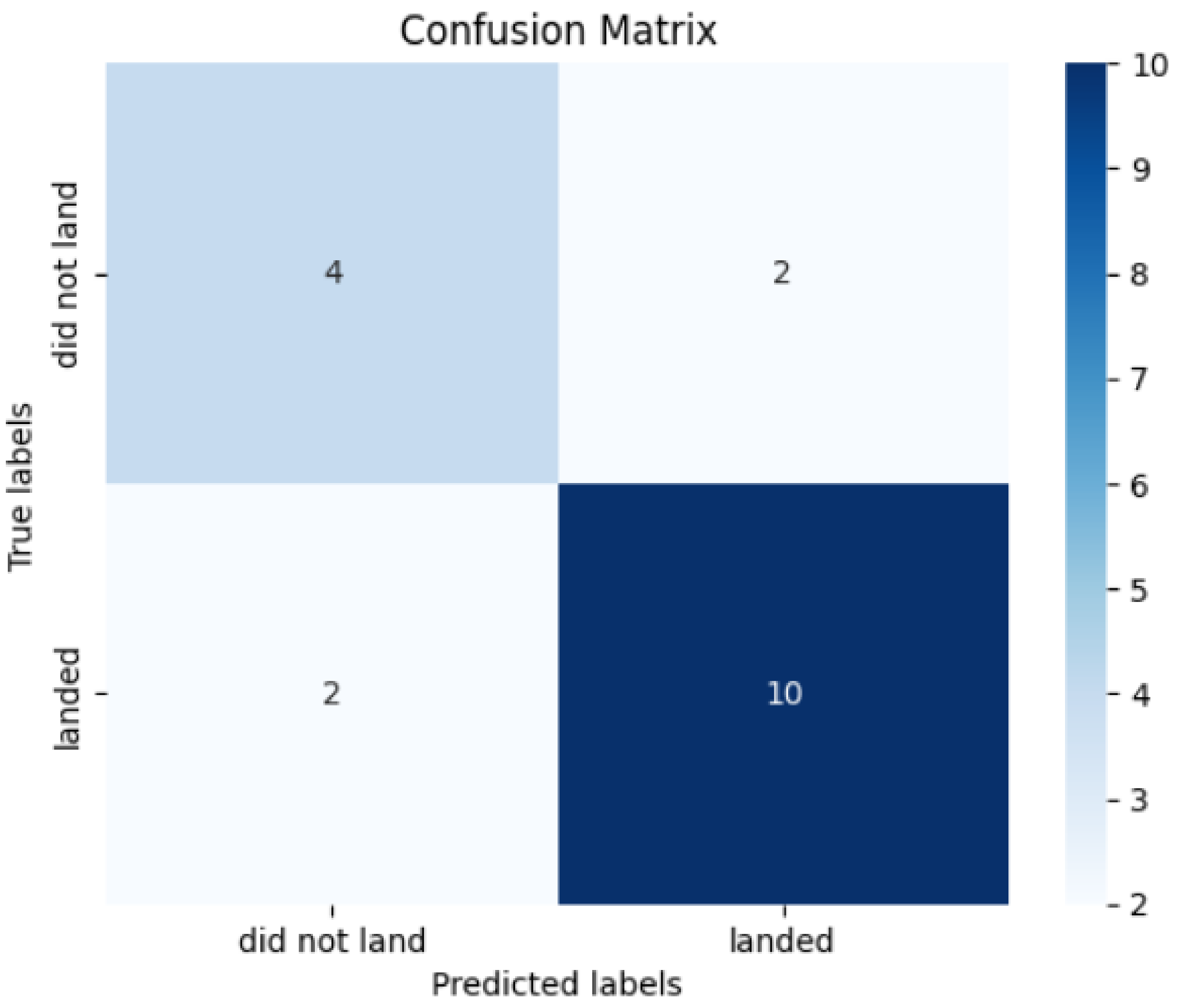


Notes: More balanced errors (FP=2, FN=2); slightly lower overall accuracy but fewer false alarms.

Predictive Analysis Results 4

KNN - Confusion Matrix

- **Key Metrics**
  - Accuracy: 0.78
  - Precision (landed): 0.83
  - Recall (landed): 0.83
  - Specificity: 0.67
  - F1 (landed): 0.83



Notes: Performance similar to Decision Tree; balanced but less accurate than LR/SVM on this split.

Conclusion

We can now quantify landing likelihood and its drivers - turning historical launch records into actionable pricing and risk insights.

Objectives Met

Built a reproducible pipeline to predict first-stage landing success and surface the drivers (orbit, payload, site, flight maturity).

Key Findings

Success climbed sharply after 2017; heavier/GTO missions are harder; site differences have narrowed; flight number (experience) matters.

Modelling Result

Simple classifiers (best = Linear SVM) achieve ~83–90% held-out accuracy with few false positives—useful for pricing & bid risk.

Business Impact

Enables evidence-based launch pricing, scenario analysis by mission type, and competitive bids for alternate providers.

Limitations

Modest sample size; public data noise; payload/mission details partially observed (e.g., exact landing method, weather).

What's Next

Add booster age/propellant margins, weather/sea-state, and landing method; calibrate probabilities; deploy a lightweight Dash app + scheduled data refresh.

Thank you!