IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<EL ABBASSI Mohammed>
<September 3th 2025>

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Why this project—and what we aim to answer

Introduction

➔Project background & context

- Commercial launch prices vary widely; SpaceX's cost advantage comes largely from **reusing the Falcon 9 first stage**.
- A successful first-stage **landing** is the linchpin for reusability—so estimating landing likelihood informs cost/risk scenarios.
- This capstone builds a **reproducible pipeline** (API data → wrangling → EDA/visuals → geospatial view → classification model) to estimate that likelihood.

Introduction

➔Problems we want to find answers to

- P1 — **Drivers**: Which factors (e.g., payload mass, orbit, launch site, booster/flight history) most influence landing success?
- P2 — **Prediction**: Given mission parameters, what is the probability that the first stage lands successfully?
- P3 — **Reliability**: How stable are predictions across sites/boosters and over time (validation, calibration)?
- P4 — **Decisioning**: What threshold best balances false alarms vs. missed landings for stakeholder objectives (bidding, scheduling, risk)?

Section 1

Methodology

Methodology

Executive Summary

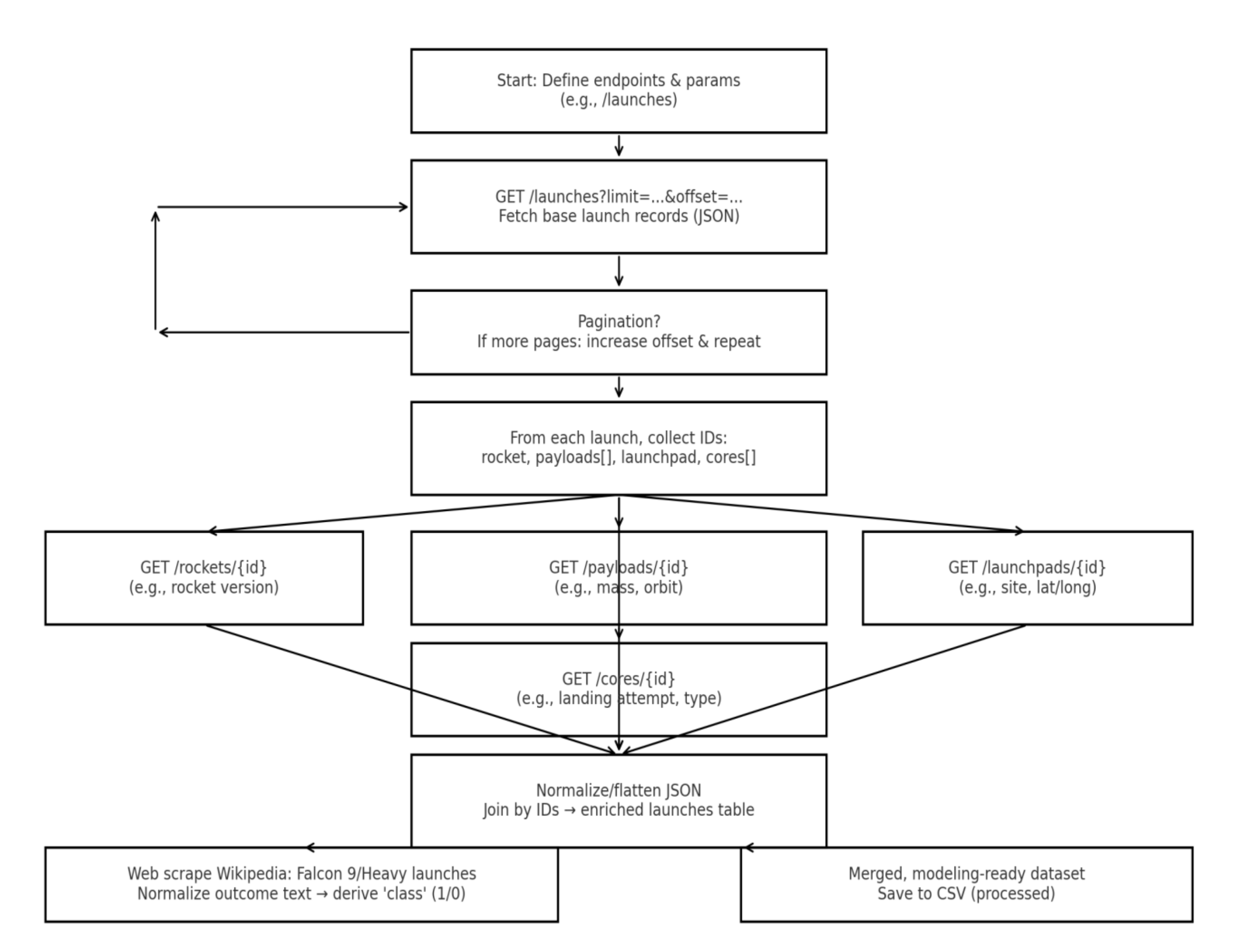
- **Data collection:** SpaceX REST API + targeted web scraping;
- **Wrangling & feature prep:** Typed columns, missing-value strategy, categorical encoding, leakage guards; export to modeling-ready tables.
- **EDA (SQL + visuals):** Rapid questions in SQL; confirm patterns with charts (distribution, correlation, categorical contrasts).
- **Interactive analytics:** Folium (maps) for site context; Plotly Dash for filterable trends.
- **Predictive analysis:** Baselines → tuned models; validation, calibration, and threshold setting aligned to stakeholder goals.

Data Collection

- **Primary source (API):** Collected launch records via the SpaceX REST API.
- **Challenge:** Key fields were returned as IDs only (rocket, payloads, launchpad, cores)—no descriptive attributes.
- **Solution:** Per-ID expansion calls to the API, then joins to enrich launches with rocket version, payload mass/orbit, launch site, and core/landing attempt metadata.
- **Complementary source (web):** Scraped the Wikipedia page “List of Falcon 9 and Falcon Heavy launches” to validate outcomes and fill gaps.
- **Quality controls:** Deduplication, type casting, schema checks, and basic provenance logs.

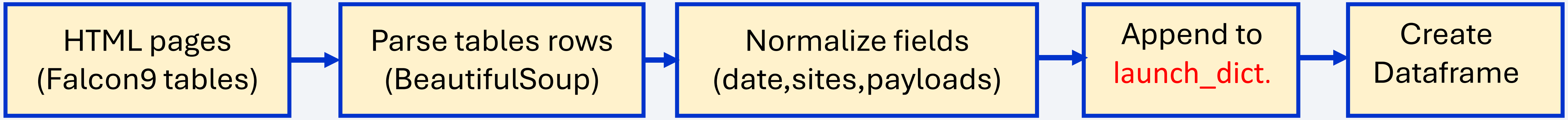
Data Collection – SpaceX API

- Base launches from /launches, expand linked resources by ID (/rockets, /payloads, /launchpads, /cores), join/clean and save the modeling-ready CSV."
- *GitHub URL* *(https://github.com/IBOSS24/Cap-stone/blob/main/Data_collecting/jupyter-labs-spacex-data-collection-api.ipynb)*



Data Collection - Scraping

- **Purpose :** Extract **structured launch records** from the Falcon 9/Heavy Wikipedia tables into a clean, tabular format (no outcome verification here).
- **Method:** Fetch HTML → parse table rows → **normalize fields** (date, site, payload, orbit, booster/serial, etc.) → append to a Python launch_dict.
- *GitHub URL (https://github.com/IBOSS24/Cap-stone/blob/main/Data_collecting/jupyter-labs-webscraping.ipynb)*



Data Wrangling

➔From raw outcomes to training labels

- **Goal:** Convert messy landing outcomes into a clean binary label

class (1 = successful landing, 0 = not).

- **Outcome normalization:** Standardize strings (trim/case) and map to labels:
  - **1:** True ASDS, True RTLS (landed on droneship or returned to launch site)
  - **0:** False ASDS, False RTLS, True Ocean, False Ocean, None ASDS *(no attempt/expended)*
- **Result:** Modeling-ready columns: class (target), LandingPad (clean categorical), plus consistent types across all features.
- *GitHub URL (https://github.com/IBOSS24/Cap-stone/blob/main/EDA/labs-jupyter-spacex-Data%20wrangling.ipynb)*

EDA with Data Visualization

➔Explore relationships & trends before modeling

- Build intuition about **which features move with landing success** and where data may be sparse or noisy.
- Surface **outliers, drift, and site/orbit differences** to guide feature engineering and model design.
- Keep this exploratory—**no causal claims**, just patterns to test later.
- *GitHub URL (https://github.com/IBOSS24/Cap-stone/blob/main/Data_Visualization/EDA_Visualization.ipynb)*

EDA with SQL

➔Fast, auditable exploration SQLite

- **Purpose:** Answer key questions quickly and **cross-check** pandas results using SQL in the notebook.
- **Method:** Load the processed launch table into **SQLite**, run compact SELECT…GROUP BY… queries, and export small result tables for charts.
- **Why it helps:** SQL is **transparent and reproducible** (great for reviews), catches data-quality issues early (missing pads, odd IDs), and validates assumptions before modeling.
- **Scope in this step:** counts, rates, and trends by **site, orbit, payload band, booster version**, and **year**—no modeling yet.

EDA with SQL (2)

Example questions we answer with SQL:

- How many **distinct launch sites**? Which sites have the most launches?
- What's the **success rate by LaunchSite** and by **Orbit**?
- Do **heavier payloads** (e.g., ≥ 8,000 kg) correlate with lower success?
- Which **booster versions** flew most often?
- Year-over-year **launch volume** and **success rate**.
- How many rows have **missing LandingPad** or **no landing attempt**?

EDA with SQL (SQL- snippets)

```sql
SELECT DISTINCT "Launch_Site" from SPACEXTABLE
SELECT *  from SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%'LIMIT 5
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)'
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
SELECT min(Date )FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)'
SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4000 and 6000
SELECT count(*) as "Total_Number","Mission_Outcome" from SPACEXTABLE group by "Mission_Outcome"
SELECT Booster_Version FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_= (select max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
SELECT substr(Date, 6,2) as month ,Booster_Version,Launch_Site FROM SPACEXTABLE
    WHERE Landing_Outcome LIKE 'Failure (drone ship)' and substr(Date,0,5)='2015'
SELECT count(*) as Total_Outcomes,Landing_Outcome,Date FROM SPACEXTABLE
    where Date between '2010-06-04' and '2017-03-20'
    group by Landing_Outcome order by Total_Outcomes desc
```

- *GitHub URL (https://github.com/IBOSS24/Cap-stone/blob/main/EDA/jupyter-labs-eda-sql-coursera_sqllite.ipynb)*

Build an Interactive Map with Folium

➔Does location matter? Mapping launch sites & outcomes

- **Why map it:** Geography can shape operations . We visualize **where** launches occur and **how** success clusters by site.
- **How we analyze:** An interactive **Folium** map with **MarkerCluster**—each point is a launch; **green** = **success**, **red** = **failure**. Popups summarize site, payload band, orbit, and outcome.
- **What we look for:**
  - Site-level baseline success rates and **mix effects** (payload/orbit by site).
  - Spatial patterns (e.g., coastal approach paths, downrange recovery) that may correlate with outcomes.
- *GitHub URL (https://github.com/IBOSS24/Capstone/blob/main/Data_Visualization/lab_jupyter_launch_site_location.ipynb)*

Build a Dashboard with Plotly Dash

➔Self-serve exploration of landing drivers

- **What it is:** A **Plotly Dash** web app that lets stakeholders explore the SpaceX dataset without code.
- **Key interactions:**
  - **Launch Site selector** (“All Sites” or a specific pad)
  - **Payload mass slider** to focus by weight bands
  - (Optional) **Orbit / Booster version** filters
  - **Live charts:** success **pie** (by selection) + **scatter** (Payload vs. success), hover tooltips, click-to-filter
- **Why it helps:** Quickly compare sites, payload bands, and booster categories; spot patterns and outliers before we lock the model features.
- *GitHub URL (https://github.com/IBOSS24/Cap-stone/blob/main/Data_Visualization/SpaceX_DashBoard.py)*

Predictive Analysis (Classification)

- ➔**What we do in this step**
- **Define features & target:** Use the cleaned table (e.g., dataset_part_3.csv). Target is **class** (1=landed; 0=not).
  - **Numeric:** PayloadMass, Flights, etc.
  - **Categorical:** LaunchSite, Orbit, Booster/Serial (one-hot/target-encoded as needed).
- **Build pipelines:** ColumnTransformer + Pipeline so all preprocessing happens **inside CV** (prevents leakage).
- **Baselines:** Logistic Regression ,SVC, DecisonTree and KNN.
- **Candidate model:** Decision Tree has the highest accuracy on test data
- *GitHub URL (https://github.com/IBOSS24/Cap-stone/blob/main/Model_development/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb)*

Results

1. Exploratory data analysis results

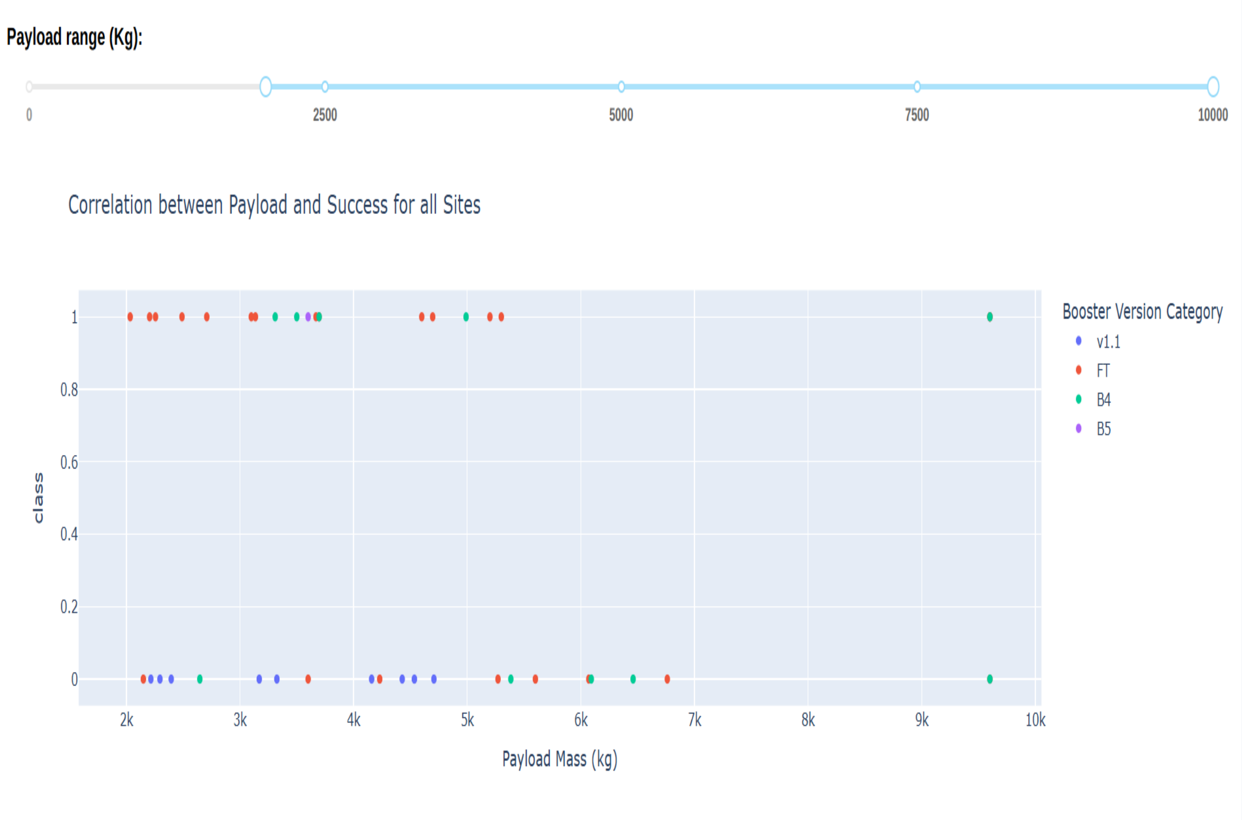
•**Site & orbit effects:** Success rates differ by **Launch Site** and **Orbit**
•**Payload impact:** Heavier **PayloadMass** bands correlate with lower success; effect size varies by site/orbit.
Trend over time: Yearly success is improving while launch volume rises—good sign of learning/operations maturity.

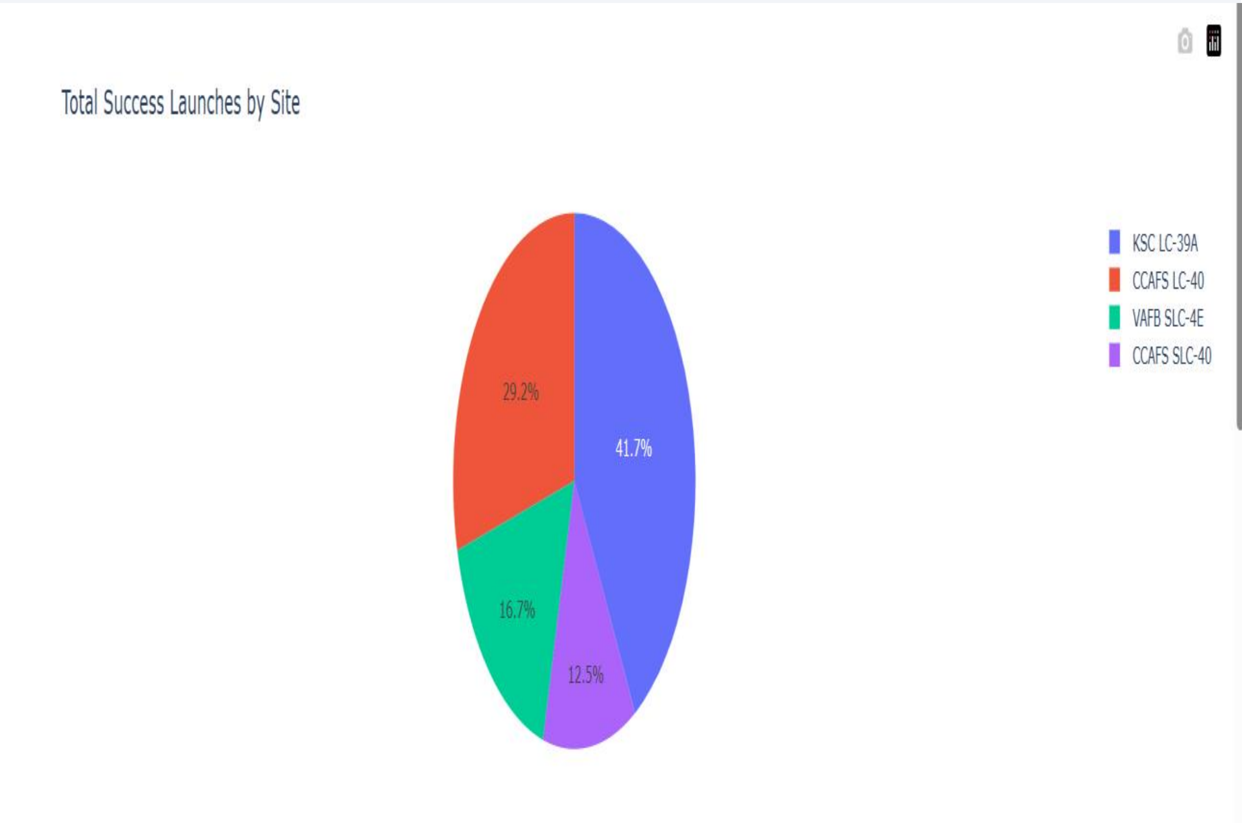
2. Predictive analysis results

Best model: `{DecisionTreeClassifier}` with pipelines (scalers/Transfomer) **inside CV** to prevent leakage.

Results

3. Interactive analytics demo in screenshots



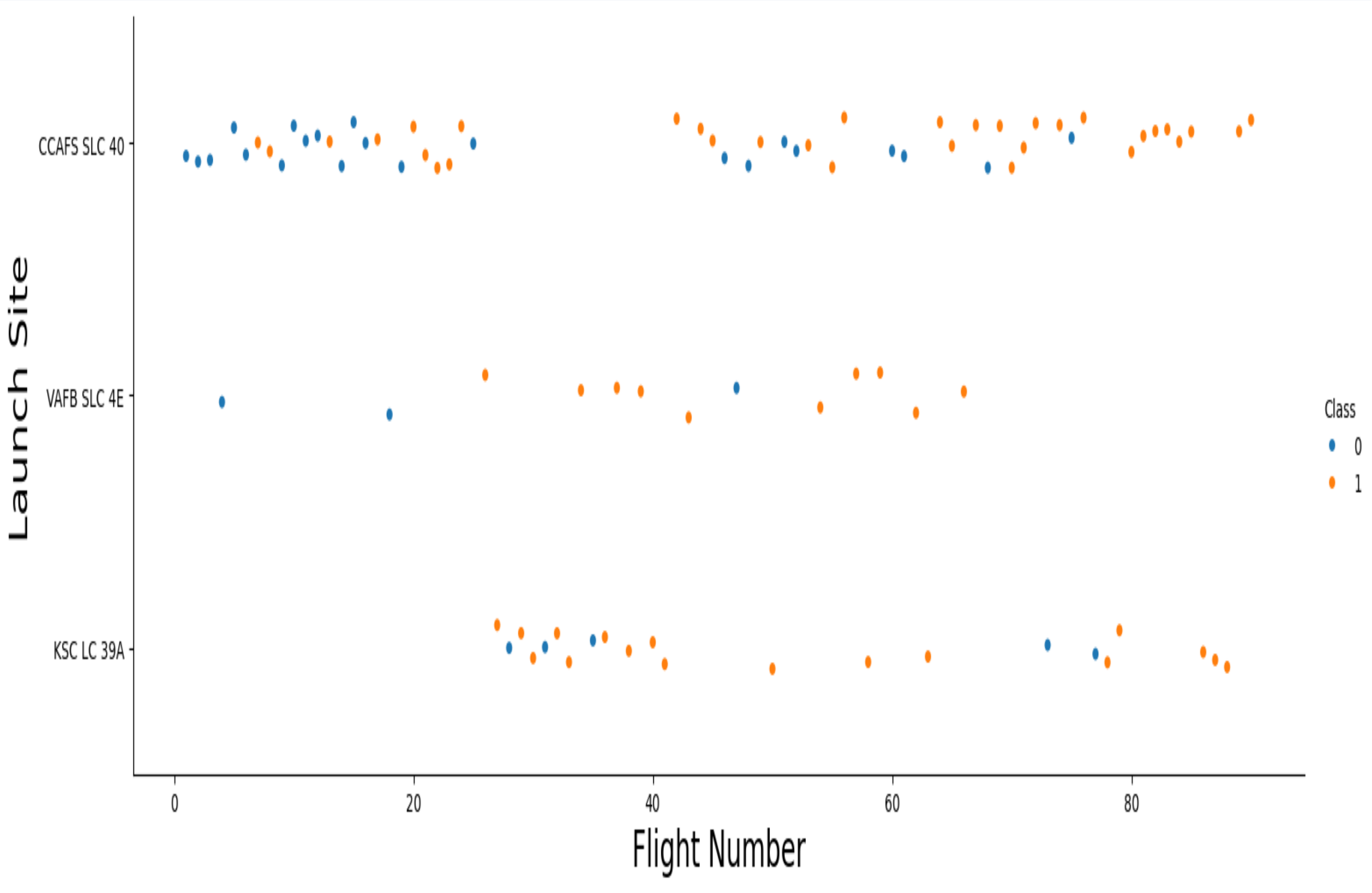


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

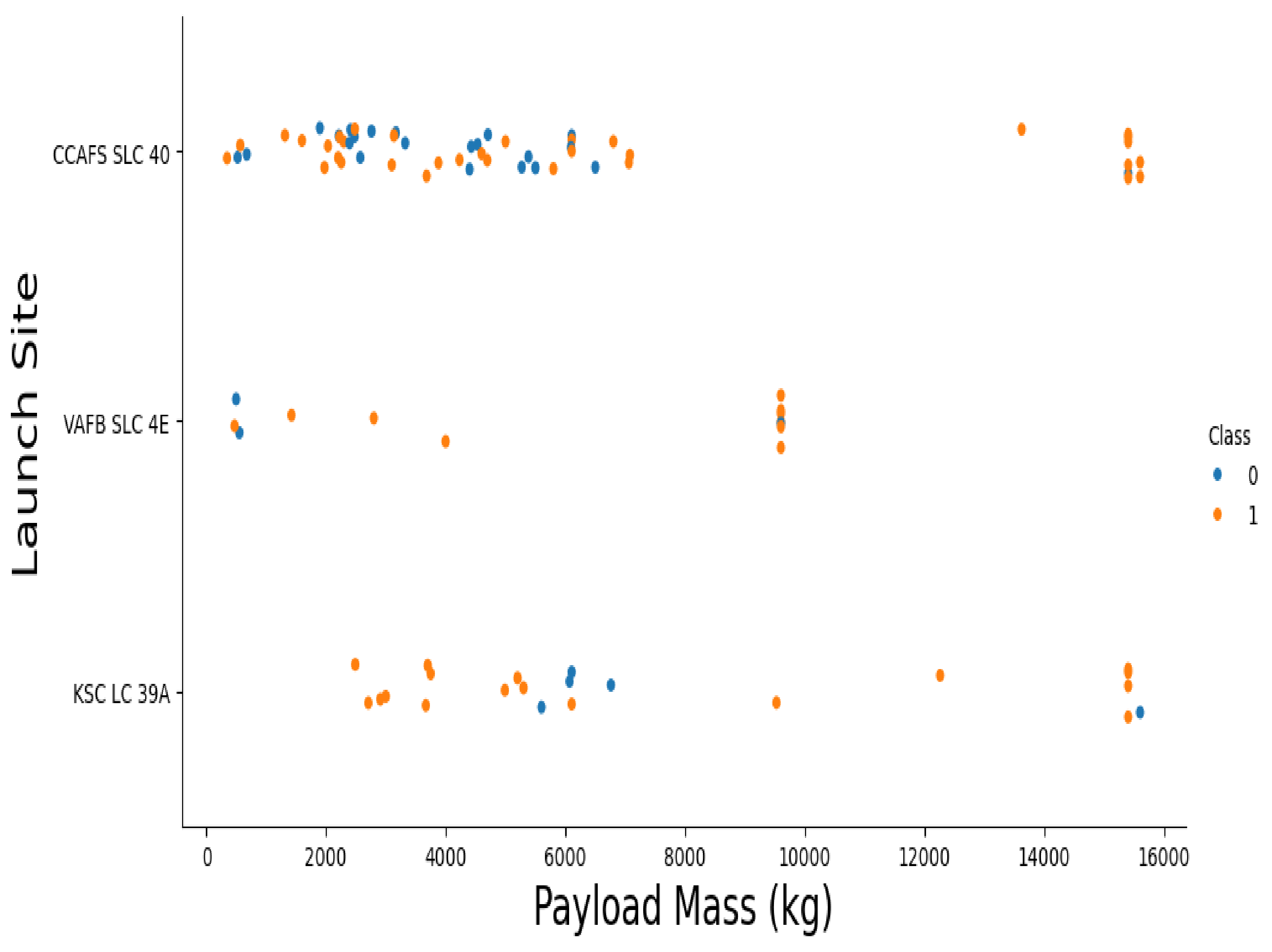
- **Learning curve visible:** Early flights (low Flight Number) show more **failures (blue)**; later flights trend to **success (orange)**—consistent with operational learning.
- **By site:**
  - **CCAFS LC-40** spans most of the program; success rate improves in later flight numbers.
  - **KSC LC-39A** appears mostly in later flights with **high success density**.
  - **VAFB SLC-4E** shows mid-program activity with mixed outcomes (mission mix likely differs).
- **Implication:** Apparent "site effect" is partly **confounded by time and mission mix** (payload/orbit).



Payload vs. Launch Site

Payload mass by site: heavy missions cluster at KSC/CCAFS; success remains broadly robust

- **Distribution:**
  - KSC LC-39A carries the **heaviest payloads (≈12–16 t)** and a wide mid-range.
  - CCAFS LC-40 spans a **broad mass range (≈0.5–7 t, plus a few >14 t)**.
  - VAFB SLC-4E clusters around **~9–10 t** with fewer total launches (small-N caution).
- **Outcome pattern:** **Success (orange)** dominates across masses and sites; **failures (blue)** appear sporadically, slightly more frequent in mid-mass bands at CCAFS/KSC.
- **Interpretation:** The mass effect alone looks **modest**; outcomes likely depend on **mission mix** (orbit type), **booster/flight history**, and **time**. Expect **site × payload × orbit** interactions.

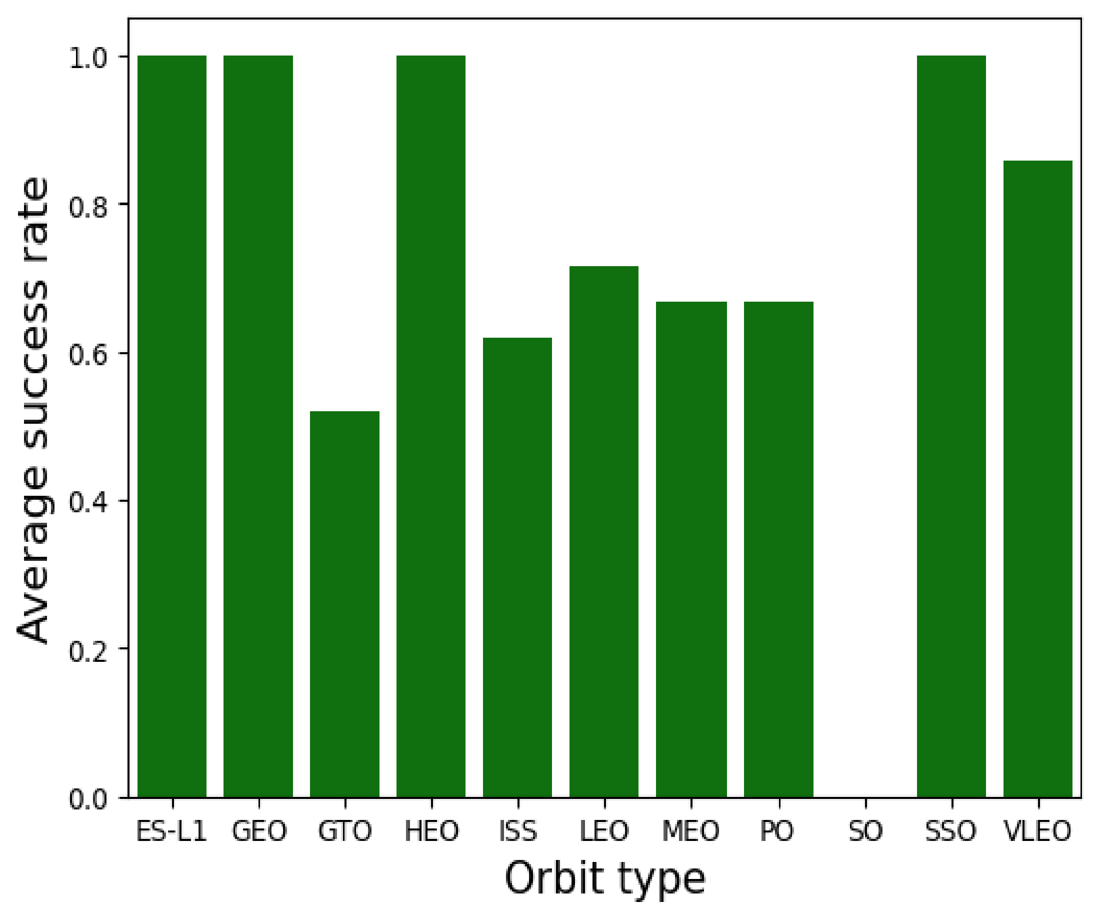


Success Rate vs. Orbit Type

Average landing success rate by orbit type. Bars reflect sample means ;

- **Standouts: GEO, SSO, ES-L1, HEO ~100%** average success in this sample; **VLEO ~0.85–0.90**.
- **Lower tier: GTO ~0.5** on average—consistent with **higher-energy missions** and longer downrange recovery, which raise landing difficulty.
- **Middle group: LEO/MEO/PO ~0.65–0.75**.
- **For modeling:**

We will Keep **Orbit** as a key categorical feature and test **interactions** with **PayloadMass** and **LaunchSite**.



Flight Number vs. Orbit Type

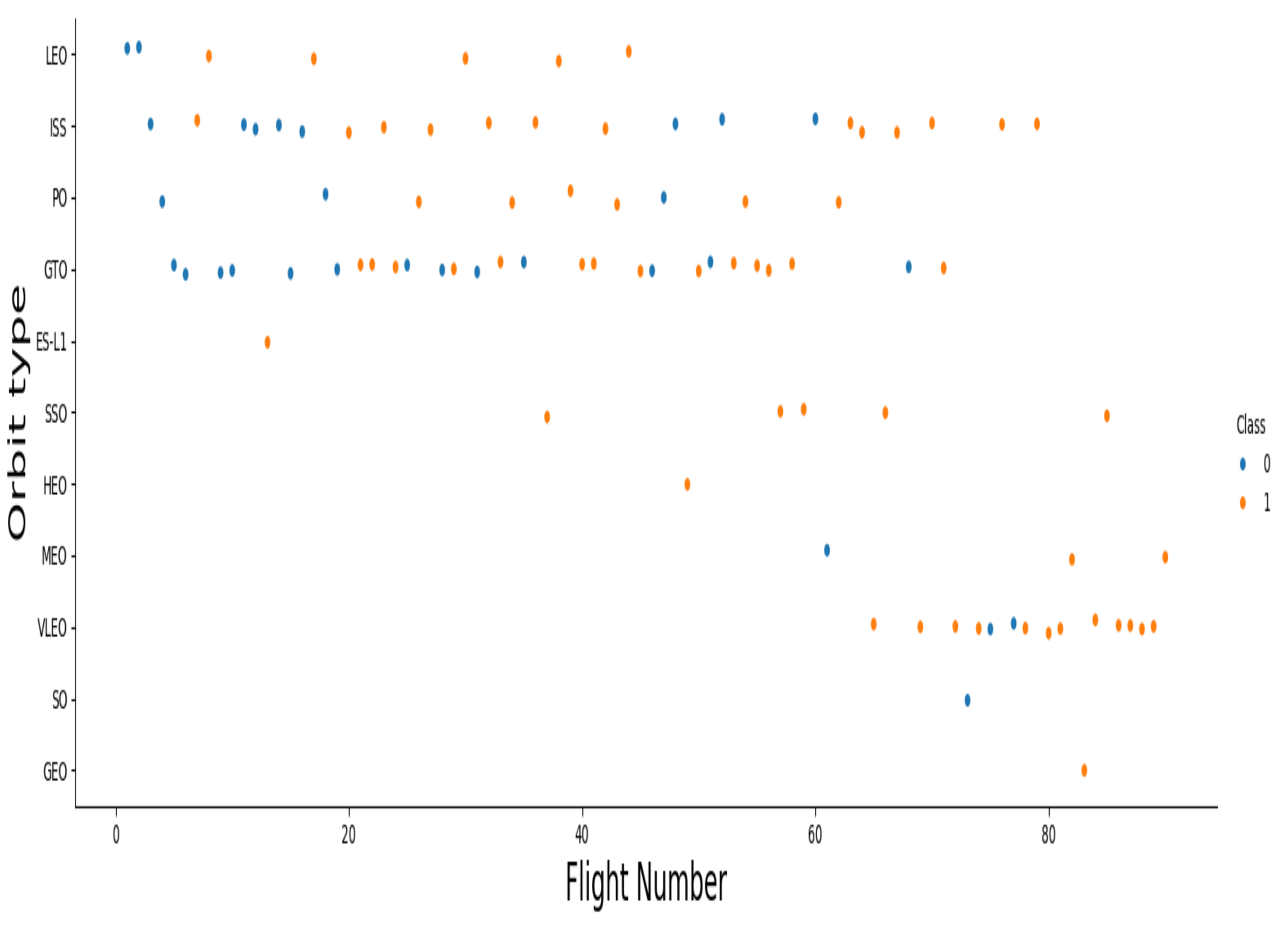
Orbit mix shifts over time; success rises within most orbits

•**Orbit mix changes with program maturity:** Early flights are mostly **LEO/ISS/PO**; later flights add many **VLEO** missions and some **GTO/SSO**.

•**Within-orbit learning:** In **GTO** and **VLEO**, early points include more **failures (blue)**, while later points are predominantly **successes (orange)** → operational learning/upgrades.

•**Small-N orbits:** **ES-L1/HEO/MEO/GEO** appear rarely; treat any ***"100%" bars*** cautiously in summaries.

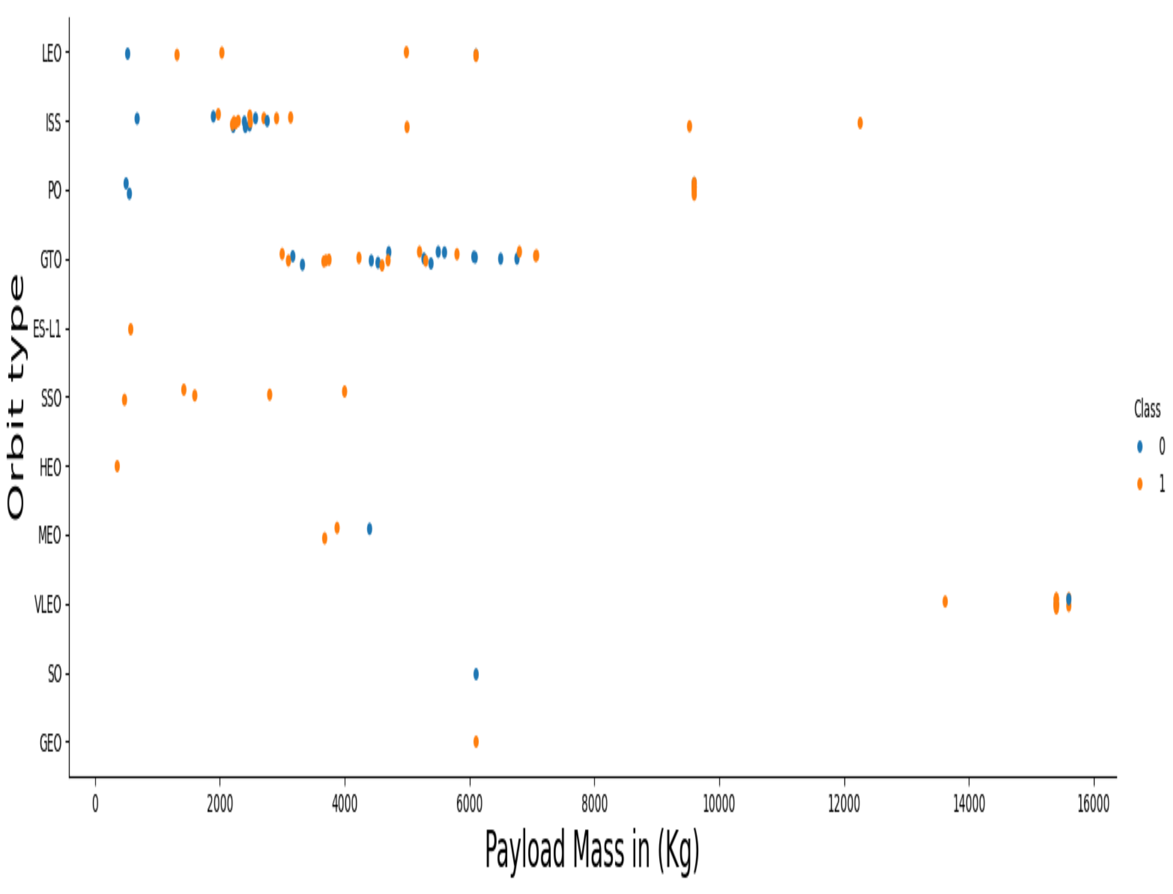
•**Modeling implication:** Keep **Orbit** as a primary categorical and include **Orbit × time (FlightNumber/Date)** and **Orbit × payload/site** interactions to separate orbit difficulty from program learning.



Payload vs. Orbit Type

Payload mass varies by orbit; outcomes stay strong across most bands

- **Orbit–mass fingerprints:**
  - **ISS/LEO/PO** cluster in **2–6 t**;
  - **GTO** concentrates around **4–6 t**;
  - **VLEO** carries the **heaviest payloads (~15–16 t)**;
  - **ES-L1/HEO/MEO/GEO** appear rarely (small-N).
- **Outcomes: Success (orange)** is common across orbits and masses; **failures (blue)** are sporadic—slightly denser in the **~5–6 t** band for **GTO/LEO**.
- **Interpretation:** Payload mass **alone** doesn't explain outcomes; effects likely depend on **orbit** and **recovery mode** (ASDS vs RTLS), plus **program maturity**.
- **Modeling implication:** Include **mass × orbit** (and **mass × site**) interactions;



Launch Success Yearly Trend

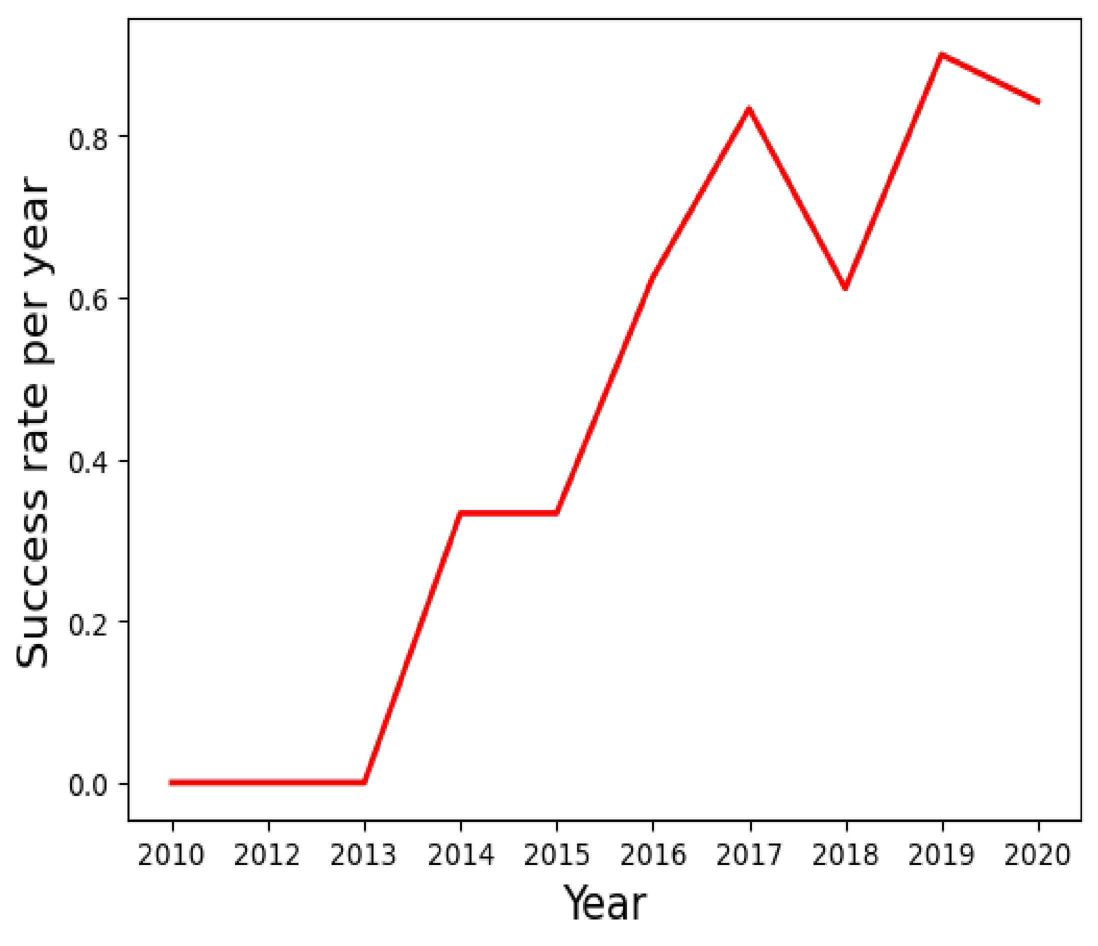
Reliability improves over time—clear learning curve with a brief 2018 dip

•**Upward trend:** Success rate rises from **0% (2010–2013)** → **~0.33 (2014–2015)** → **~0.63 (2016)** → **~0.84 (2017)**.

•**Momentary setback:** **2018** dips to **~0.61**, followed by a rebound to **~0.90 (2019)** and **~0.85 (2020)**.

•**Takeaway:** The program shows **systematic reliability gains** over the decade—consistent with operational learning and hardware/process upgrades.

•**Caution:** Early years have **few launches**, so rates are volatile; confirm trends with **counts** and rolling averages.



All Launch Site Names

➔What this query does

- Get the list of unique launch sites (for filters & mapping)
- **DISTINCT** returns **unique** values of "Launch_Site" from the table.
- Quotes are needed because the column name contains a **space**.
- Purpose is **schema sanity**: ensure we have the expected sites

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	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 attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Why it’s useful

- Quick **sanity check** and preview of records for a specific site family (Cape Canaveral).
- Helps verify column formatting before building visuals/filters.

Total Payload Mass

How much payload mass did NASA (CRS) fly?

- Sums "PAYLOAD_MASS__KG_" only where Customer is exactly NASA (CRS). SUM ignores NULLs; result is in kilograms.

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

What’s the average payload mass for F9 v1.1 missions?

LIKE 'F9 v1.1%' captures F9 v1.1 and any suffix AVG ignores NULLs automatically.

AVG("PAYLOAD_MASS__KG_")
2534.666666666665

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad was in

‘2015-12-22’

➔*Finds the **earliest return-to-launch-site (RTLS)** success in our dataset—useful as a program milestone and to anchor the timeline in later analyses.*

Successful Drone Ship Landing with Payload between 4000 and 6000

What this query asks:

Return the **booster version** and **payload mass** for launches that **successfully landed on the droneship (ASDS)** and carried **4,000–6,000 kg**.

- 4–6 t is a **mid-weight band**—useful to compare **booster variants** and **sites** under similar load.

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

How often does each mission outcome occur?

- Groups rows by **Mission_Outcome** and returns the **count per outcome**.

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

- **Total outcomes :** Success : 100 / Failure :1

Boosters Carried Maximum Payload

Which booster version flew the **maximum payload mass** ?

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

2015 Launch Records

When and where did ASDS (drone-ship) failures occur in 2015?

month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

We’re isolating **2015** and **ASDS failures** to understand early recovery challenges.

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

This summarizes outcome frequencies in the **2010-2017** window; use it to benchmark later model results against historical rates.

Total_Outcomes	Landing_Outcome	Date
10	No attempt	2012-05-22
5	Success (drone ship)	2016-04-08
5	Failure (drone ship)	2015-01-10
3	Success (ground pad)	2015-12-22
3	Controlled (ocean)	2014-04-18
2	Uncontrolled (ocean)	2013-09-29
2	Failure (parachute)	2010-06-04
1	Precluded (drone ship)	2015-06-28

Section 3

Launch Sites Proximities Analysis

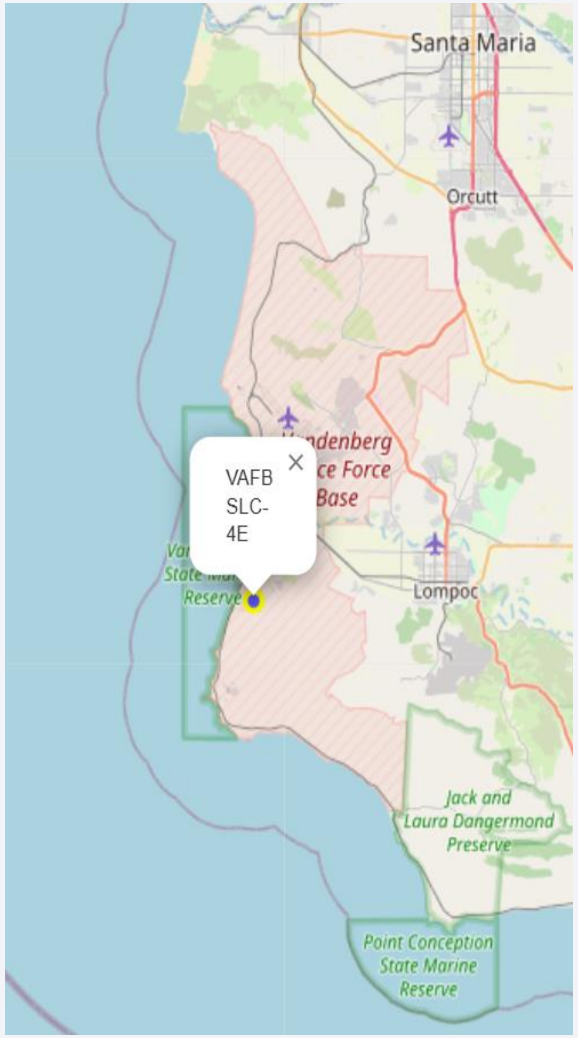
<Map launch sites>

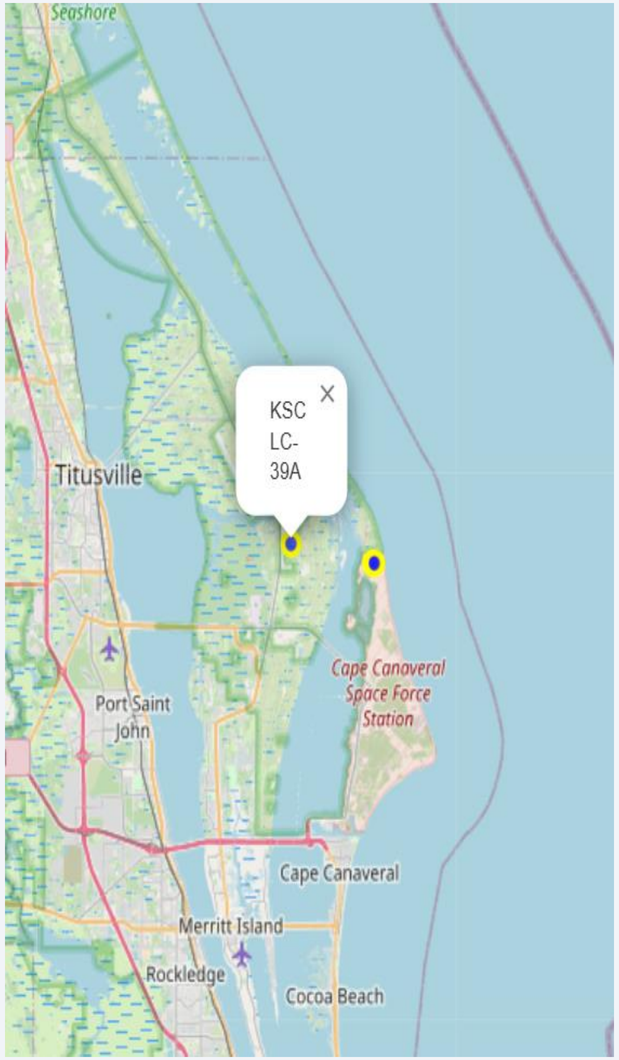
•**Coastal siting enables recovery:** all site are **immediately coastal**, which supports both **RTLS** (return-to-launch-site) and **ASDS** (droneship) recoveries downrange.

•**Operational proximity:** The pad sits near **population centers (Titusville/Merritt Island)** and major roads (I-95/FL-528), highlighting practical launch & recovery logistics.

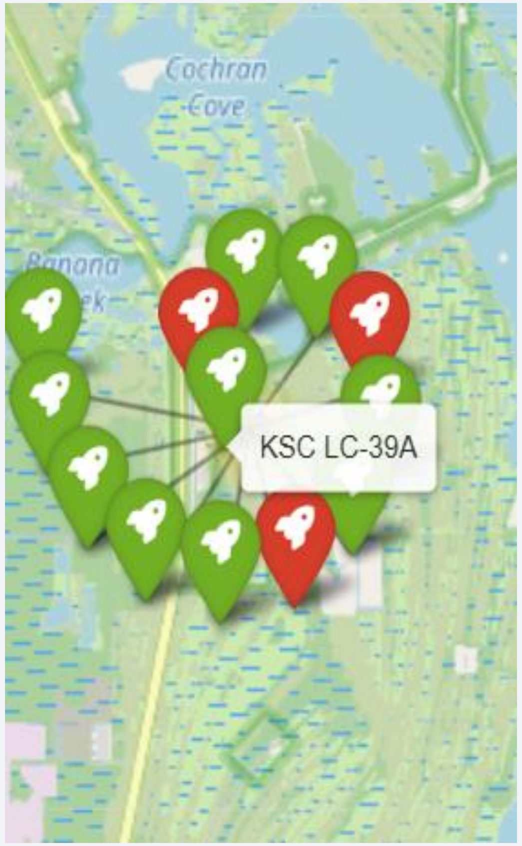
•**Feature engineering hook**: From this view you motivate distance features such as `dist_to_coast`, `dist_to_city`, and `dist_to_highway` that can be added to the dataset.

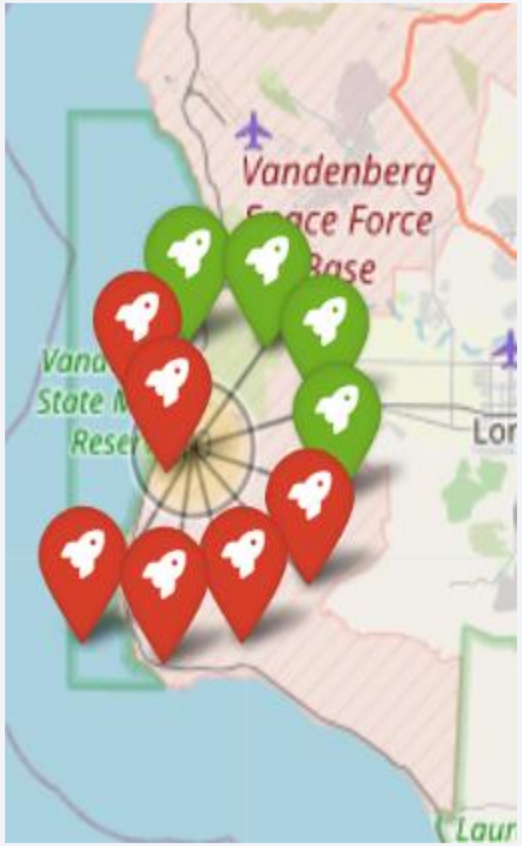
•**Caveat**: The map is **descriptive**, not causal—any site effect must be disentangled from **mission mix** (payload/orbit) and **program maturity** in the modeling step.

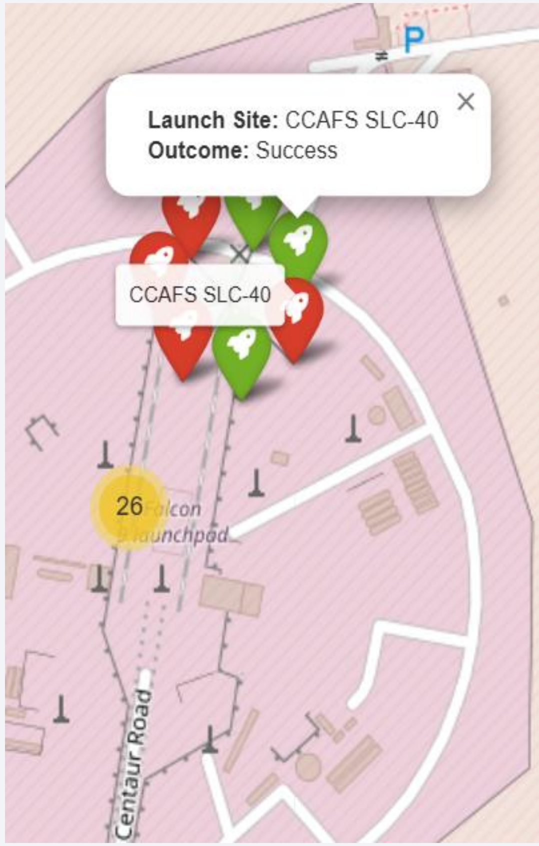


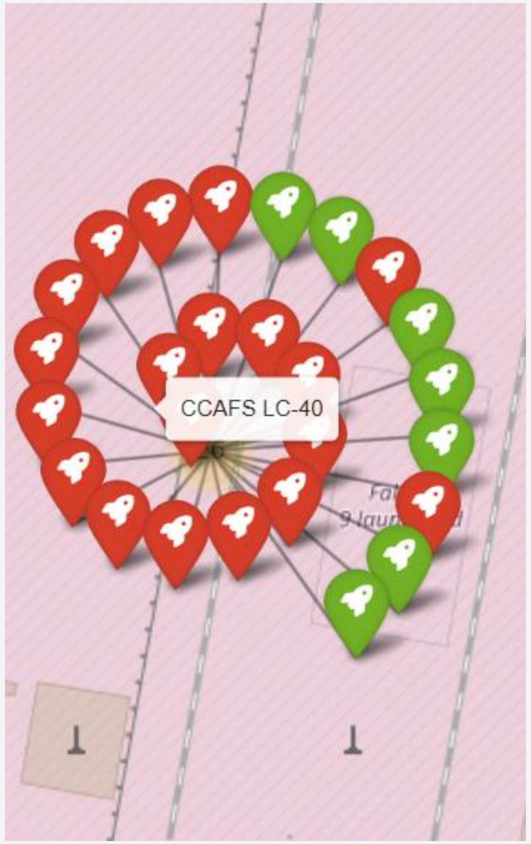


<Interactive map (per-site outcomes)> 1/2









<Interactive map (per-site outcomes)>2/2

What the screenshots show

•**Markers:** **Green** = successful landing (Class=1), **red** = not successful (Class=0). Rocket icon = individual launch.

•**Clusters & "spiderfy":** The **yellow bubble with a number** (e.g., *26*) is a cluster count. Clicking/zooming **fans markers out in a circle** around the pad so overlapping launches are visible.

•**Popups:** Each pin shows **Launch Site** and **Outcome** (sample: *"CCAFS SLC-40 → Success"*).

•**Background context:** Base boundaries, roads, and coastline help frame recovery logistics (RTLS vs downrange ASDS).

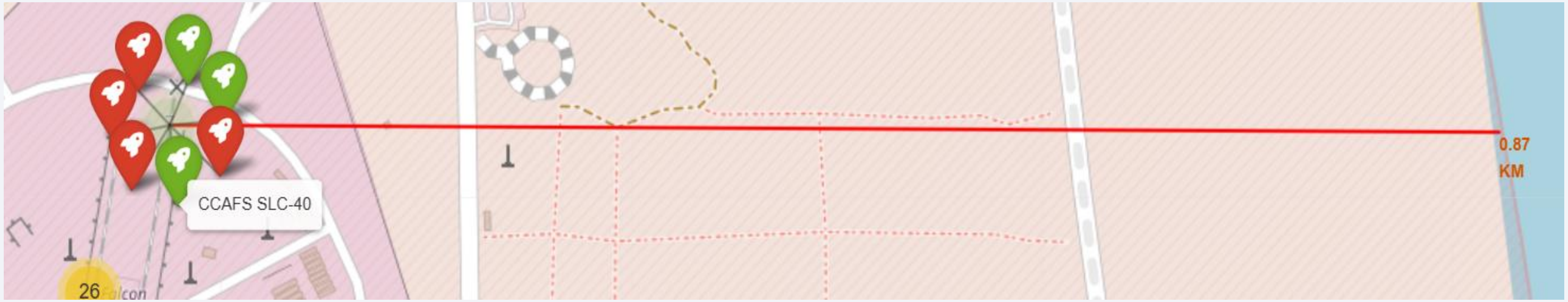
Site-level takeaways (descriptive)

•**CCAFS LC-40 (Cape Canaveral):** Large history with a **mix** of results—early program **red** markers are visible alongside later **green** ones.

•**KSC LC-39A (Kennedy):** Cluster **skews green**, consistent with **later-program** operations.

•**VAFB SLC-4E (Vandenberg):** **Fewer total launches**; outcomes are **mixed**

<Proximity feature (distance to coast)>



What the screenshot shows

- **Red line & label:** A straight line from the pad to the shoreline with a measured distance of **~0.87 KM**.
- **Context:** Coastal range infrastructure immediately east of the pad.

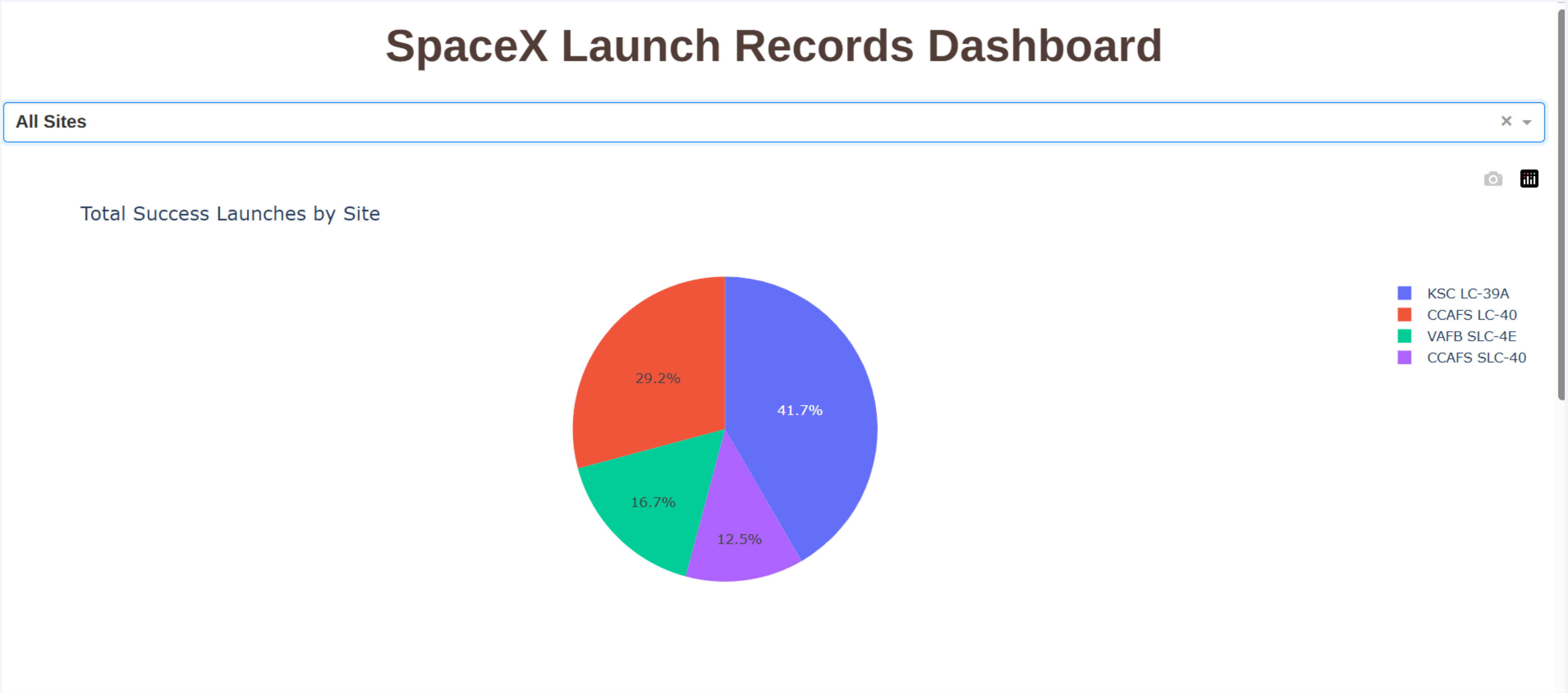
Why this matters

- **Operational intuition:** Shorter coast distance is consistent with **ASDS/RTLS logistics** (downrange recovery corridors, range safety).
- **Model input:** Proximity features can interact with **orbit** and **payload** to capture site-ops effects beyond the site ID alone.

Section 4

Build a Dashboard with Plotly Dash

<Dash screenshot (Total Success Launches by Site)>1/2



<Dash screenshot (Total Success Launches by Site)> 2/2

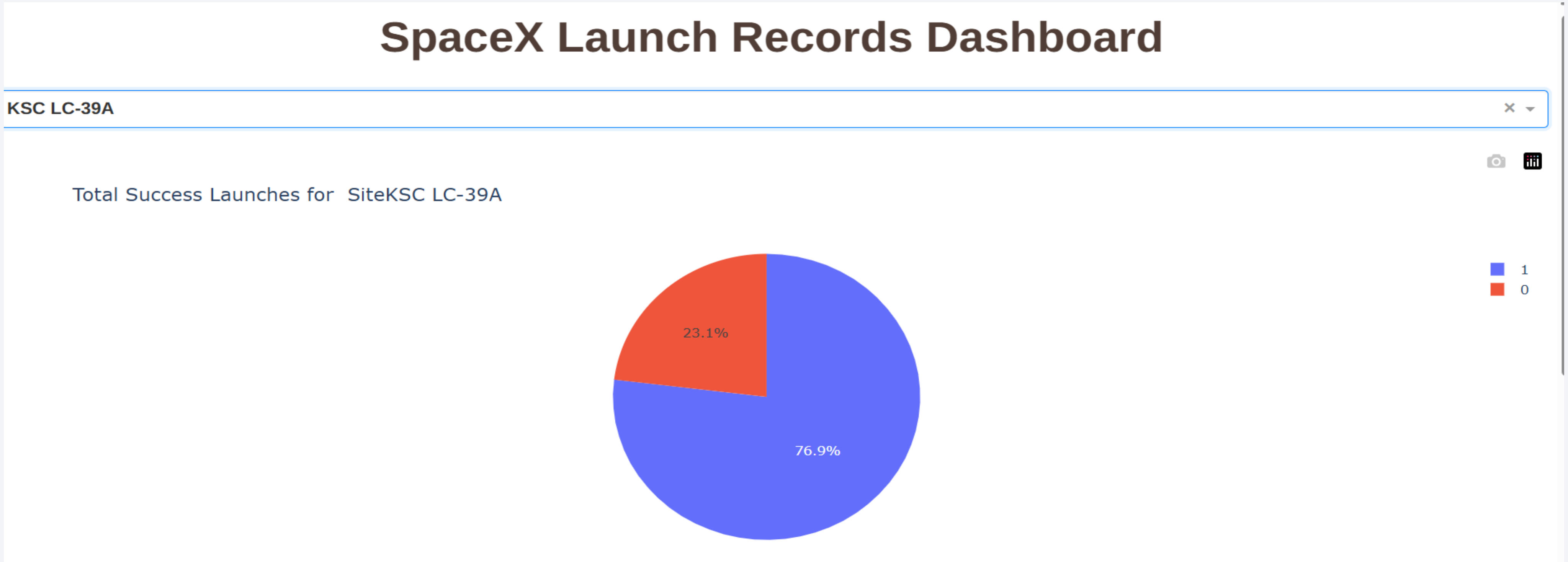
What the chart shows

- Pie segments represent the **share of all successful launches** by site (with *All Sites* selected).
- In this snapshot: **KSC LC-39A ≈ 41.7%, CCAFS LC-40 ≈ 29.2%, VAFB SLC-4E ≈ 16.7%,** and **“CCAFS SLC-40 ≈ 12.5%”**.

Key takeaways

- **Most recorded successes come from KSC LC-39A**, followed by **CCAFS LC-40**.
- **VAFB SLC-4E** contributes a smaller share, reflecting **fewer total launches** in the dataset.
- **Data quality note:** “**CCAFS LC-40**” and “**CCAFS SLC-40**” appear as **two labels for the same pad**. Normalize site names to avoid splitting one site into two slices.

<Dash view for the site with highest success rate



With the site set to **KSC LC-39A**, **~77%** of launches result in successful landings (data in this dashboard).

<Dash scatter (Payload × Outcome,colored by booster version)>1/2

<Dash scatter (Payload × Outcome,colored by booster version)> 2/2

What the three views show

•**All sites (full range):** Both **success (class=1)** and **failure (class=0)** appear across **~2–10 t**. No strong monotonic "heavier = failure" pattern.

•**All sites :**When you focus on **~2–6 t**, differences by **booster generation** emerge:

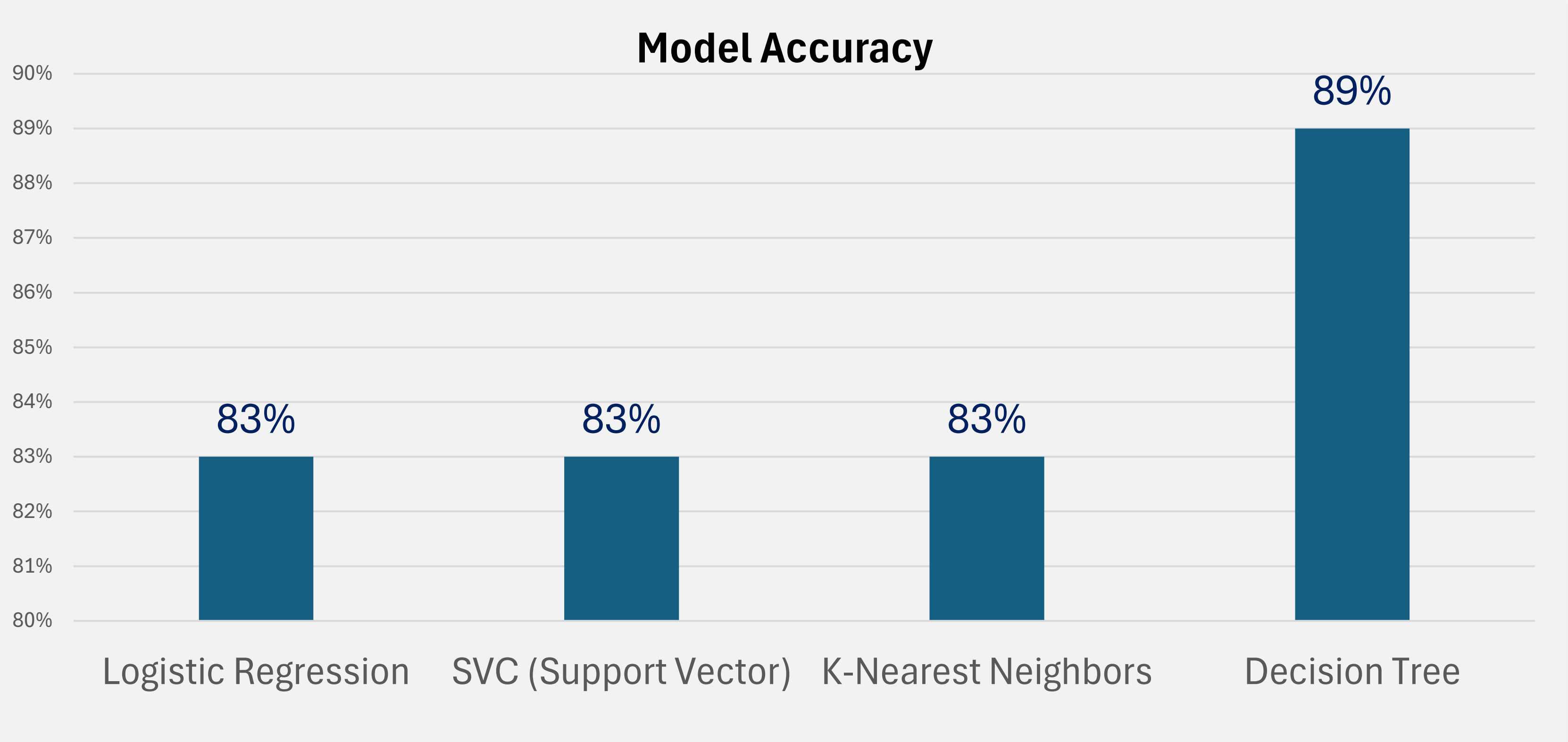
- Earlier **v1.0 / v1.1** points include more failures.
- Later **FT / B4 / B5** trend heavily to **success (y=1)** at the same masses.

•**KSC LC-39A subset:** Mostly **successes** across **~2.5–7 t** with a few failures around **~5.6–6.8 t** (FT). Site-time mix and vehicle generation likely explain more variance than payload mass by itself.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

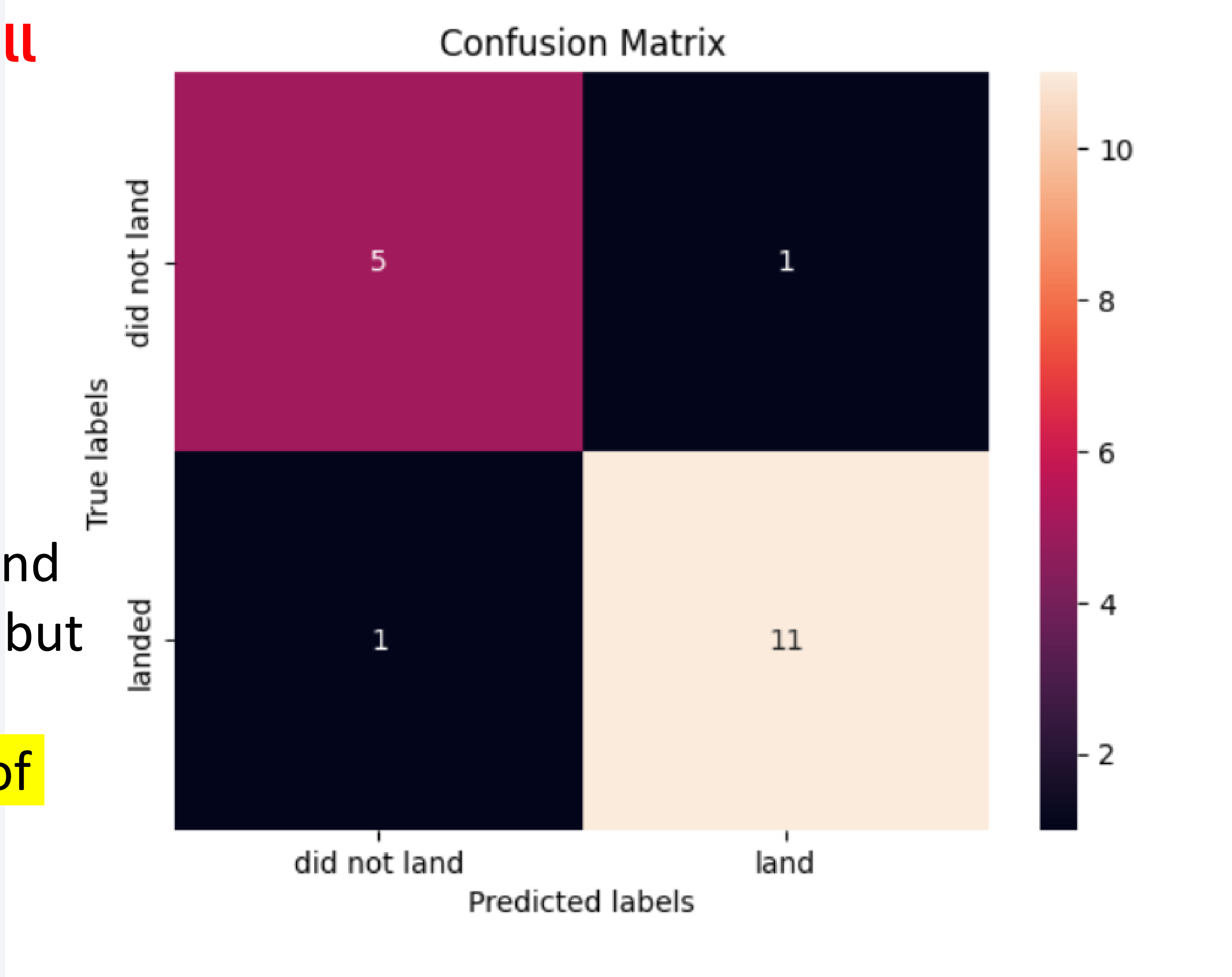


➔ **Decision Tree** shows the highest accuracy (**+6 pts** vs. others)

Confusion Matrix of the best Model

Few mistakes; strong precision/recall for “landed” predictions

- **TP (landed → landed): 11**
- **TN (did not land → did not land): 5**
- **FP (did not land → landed): 1**
- **FN (landed → did not land): 1**
- **Total N = 18**



- Only **2 errors**: **1 false alarm** (predicted land but didn’t) and **1 miss** (predicted no-land but actually landed).
- Current threshold gives a **good balance** of catching true landings and avoiding false promises.

Conclusions

1. What we accomplished

- Built a **reproducible, end-to-end pipeline**: API + scraping → wrangling/labeling → SQL & visual EDA → geospatial map → interactive Dash → trained classifier.
- Converted messy outcomes to a clean `class` label and engineered features (site, orbit, payload mass, booster version, proximity).
- **Model performance *(holdout):*** Accuracy **≈89%**,
  Precision/Recall (landed) **≈92%/92%** with only **2 errors** out of 18 (1 FP, 1 FN).

2. What the data says

- **Drivers are contextual:** orbit type, site and **booster generation** explain more than payload alone; reliability improves over time.
- **Geography informs features** (coastal sites, access) but isn't causal on its own—effects intertwine with mission mix and maturity.

Conclusions

3. Limitations to keep in mind

•**Imbalance & small-N buckets** (e.g., ES-L1/HEO/GEO) → show CIs and avoid over-interpreting perfect rates.
•**Label/metadata noise** from heterogeneous sources; site naming normalization is required.
•Metrics reflect the **analyzed window**; monitor for drift as new launches arrive.

➔What this enables

•**Pre-launch risk screens** using calibrated probabilities and a business-aligned threshold.
•**Clear explainability** (feature effects) and stakeholder-friendly views (map + dashboard).

Appendix

Recommended next steps (actionable)

1. **Operationalize** the best model with **probability calibration** and a documented **decision threshold** (optimize for the preferred trade-off).
2. **Pilot** on the **next cohort of launches** with shadow predictions; track PR-AUC/F1, precision/recall, and calibration over time.
3. **Enrich features:** weather/sea state, wind/ILS constraints, booster refurbishment/flight count recency; per-site/ per-orbit interaction terms.
4. **MLOps:** model card, monitoring (data drift & performance), retraining cadence, and alerting on threshold breaches.
5. **Governance:** finalize data quality checks (naming, units, dates) and retain an audit trail of queries/notebooks.

Thank you!