Predicting SpaceX Launch Prices and First-Stage Reuse: A Data-Driven Approach for SpaceY

Capstone Project – SpaceX Falcon 9 Launch Analysis

Author: Luis Modesto

Date: 26/12/2025

TABLE OF CONTENTS


01. EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

Goal: Use historical SpaceX Falcon 9 launch data to (1) estimate launch prices and (2) predict first-stage reuse outcomes, to inform SpaceY's competitive strategy.

Approach:

- Collected data from SpaceX API and Wikipedia (Falcon 9 launches)
- Performed data wrangling, SQL analysis, and exploratory data analysis (EDA)
- Built interactive visualizations (Folium, Plotly Dash)
- Trained classification models to predict first-stage landing success

Key results:

- Launch price strongly driven by payload mass and orbit type
- First-stage landing success is predictable with high accuracy using mission features
- Clear patterns in launch sites, orbit success rates, and mission outcomes

Takeaway: SpaceX's launch and reuse strategy can be modeled from public data, giving SpaceY a data-driven baseline for pricing, operations, and infrastructure planning.

02. INTRODUCTION

INTRODUCTION

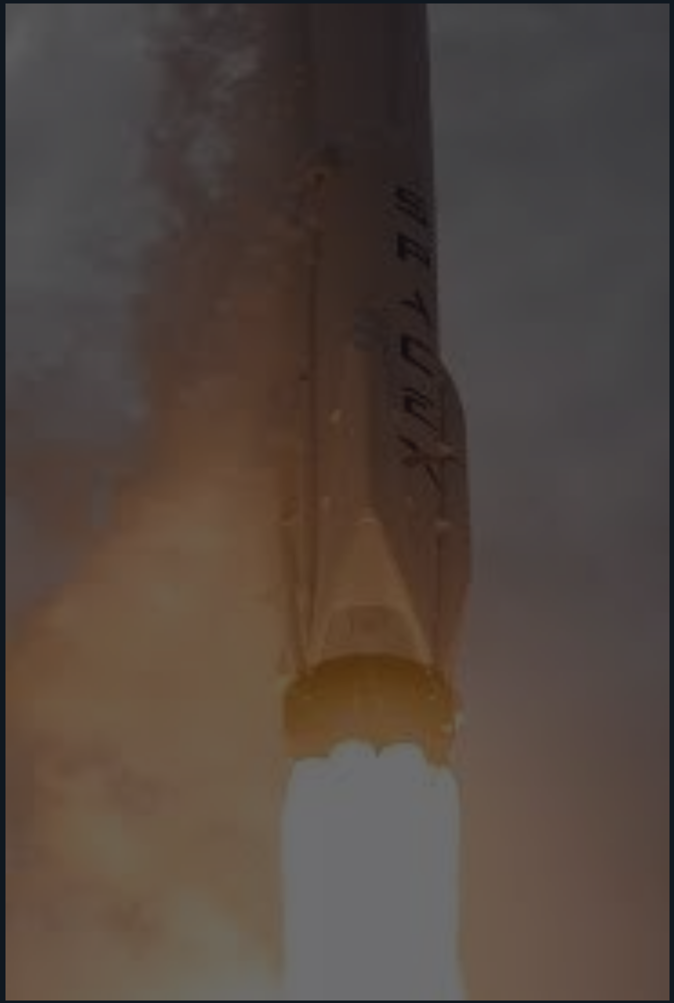
Context: Space Y plans to enter the commercial launch market and compete with SpaceX's Falcon 9. Understanding SpaceX's pricing and reuse patterns is essential to designing a viable business model.

Research questions:

1. What factors influence the price of a SpaceX launch, and how can we estimate that price?
2. Can we predict whether the first stage will land successfully and be reusable using public data?

Scope of work:

- Data collection from APIs and web scraping
- Data wrangling & normalization
- SQL-based analysis
- EDA & interactive visual analytics
- Machine learning models for landing success prediction



03. DATA COLLECTION & WRANGLING

DATA COLLECTION – SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

```
AverageValue = data_falcon9['PayloadMass'].astype(float).mean(axis=0)
data_falcon9['PayloadMass'].replace(np.nan, AverageValue, inplace=True)
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Getting Response from API

Converting Response to a .json file

Replace Missing Values

Assign list to dictionary to create dataframe

Export data to .csv file

```
data = pd.json_normalize(response.json())
data.head()
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False

DATA COLLECTION – WEB SCRAPPING

Getting Response from HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(url, headers=headers)
```

Creating BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html.parser")
soup.title
```

Finding tables

```
html_tables = soup.find_all("table")
first_launch_table = html_tables[2]
```

Getting column names

```
headers = first_launch_table.find_all("th")
for th in headers:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Creation of dictionary and appending data to keys

```
launch_dict= dict.fromkeys(column_names)
```

Converting dictionary to dataframe

Dataframe to .csv

	Flight No.	Version, Booster	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Booster landing
0	1	NaN	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	Failure
1	2	NaN	CCAFS	Dragon	0	LEO	NASA	Success	Failure
2	3	NaN	CCAFS	Dragon	525 kg	LEO	NASA	Success	No attempt\n
3	4	NaN	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	No attempt
4	5	NaN	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	No attempt\n

DATA WRANGLING

Calculate Number of launches at each site

```
df["LaunchSite"].value_counts()
```

```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

Calculate Number and occurrence of each orbit

```
filtered = df[df["Orbit"] != "GTO"]
filtered["Orbit"].value_counts()
```

```
Orbit
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
```

Calculate Number and occurrence of mission outcome per orbit type

```
landing_outcome = df["Outcome"].value_counts()
landing_outcome
```

```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
```

Create landing outcome label, converting 1 to success, and 0 to failure

```
landing_class = [
    0 if outcome in bad_outcomes else 1
    for outcome in df["Outcome"]
]
df['Class']=landing_class
df[['Class']].head(8)
```

Export dataset as .csv

```
df.to_csv("dataset_part_2.csv", index=False)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004

04. METHODOLOGY

EDA METHODOLOGY

Exploratory Data Analysis Approach

To understand patterns in launch behavior, performance, and mission characteristics, performed a Structured EDA workflow:

1
Univariate analysis

- Distribution of launch sites
- Orbit frequency
- Landing outcomes
- Payload mass ranges

2
Bivariate analysis

- Flight Number vs Payload Mass
- Payload Mass vs Launch Site
- Flight Number vs Orbit Type
- Orbit Type vs Success Rate

3
Temporal analysis

- Yearly launch success trend
- Evolution of payload mass over time

4
Visual tools used

- Matplotlib & Seaborn for static plots
- Plotly for interactive scatter plots
- Folium for geospatial mapping

Purpose

Identify relationships, anomalies, and operational patterns that inform both SQL analysis and predictive modeling

SQL METHODOLOGY

1
Database Setup
- Loaded cleaned dataset into a relational database
- Removed nulls and standardized column types
- Created indexes on launch site, booster version, and orbit for faster queries

2
SQL Analysis Goals
- Validate dataset integrity
- Extract operational insights
- Identify patterns not easily visible in raw tables
- Support EDA findings with precise counts and groupings

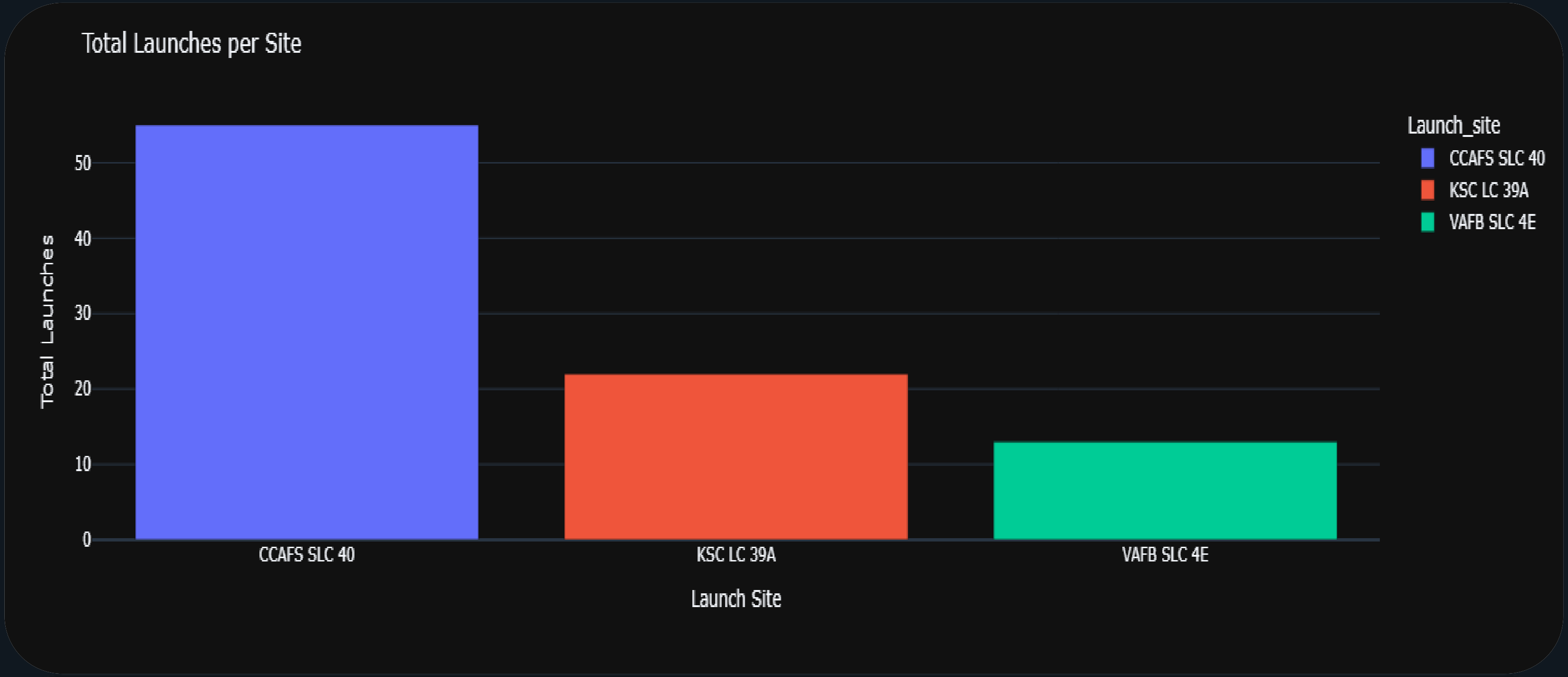
3
Key SQL Techniques Used
- GROUP BY for launch site, orbit, mission outcome
- JOIN operations to combine booster and payload data
- WHERE filters for date ranges and payload thresholds
- ORDER BY and LIMIT for ranking landing outcomes

Output
A set of validated metrics used in later slides (payload totals, mission outcomes, booster performance).

05. EDA & SQL RESULTS

EDA RESULTS – LAUNCH SITES

Launch Site Distribution

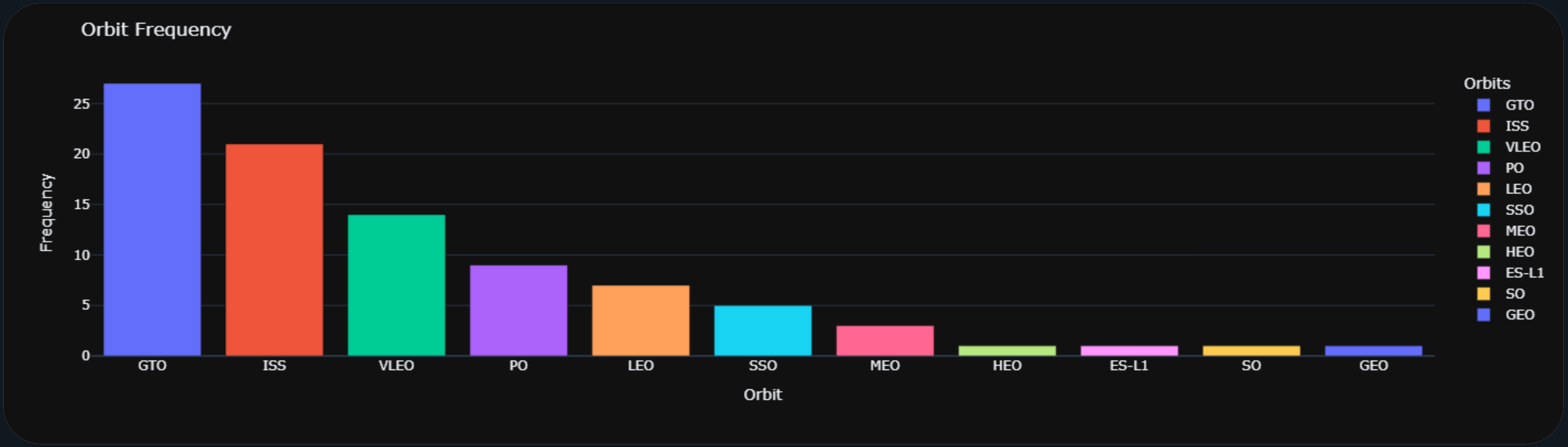


Launch_site	Total
CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

Insight: CCAFS SLC-40 is SpaceX’s primary site, while VAFB is used for polar orbits.

EDA RESULTS – ORBITS

Orbit Frequency

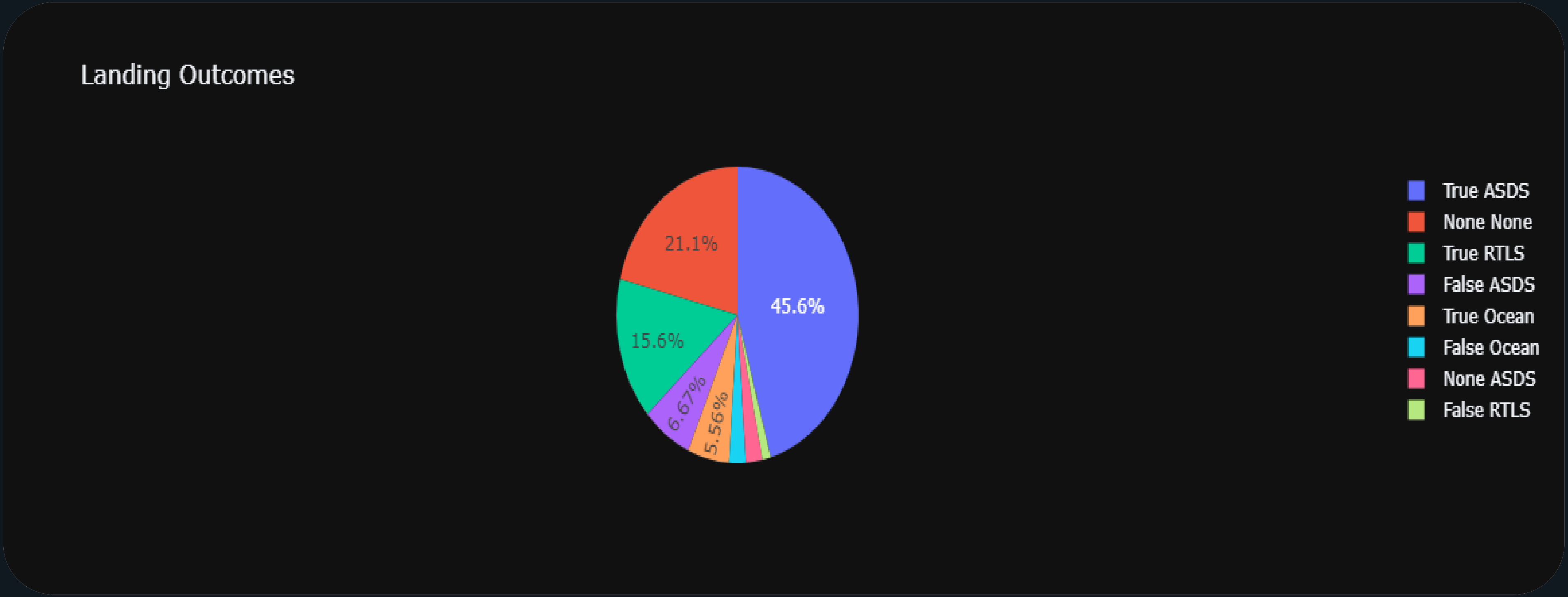


Orbits	Total
GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
HEO	1
ES-L1	1
SO	1
GEO	1

Insight: ISS and VLEO dominate, reflecting SpaceX’s strong presence in resupply and low-Earth-orbit missions.

EDA RESULTS – OUTCOMES

Landing Outcomes

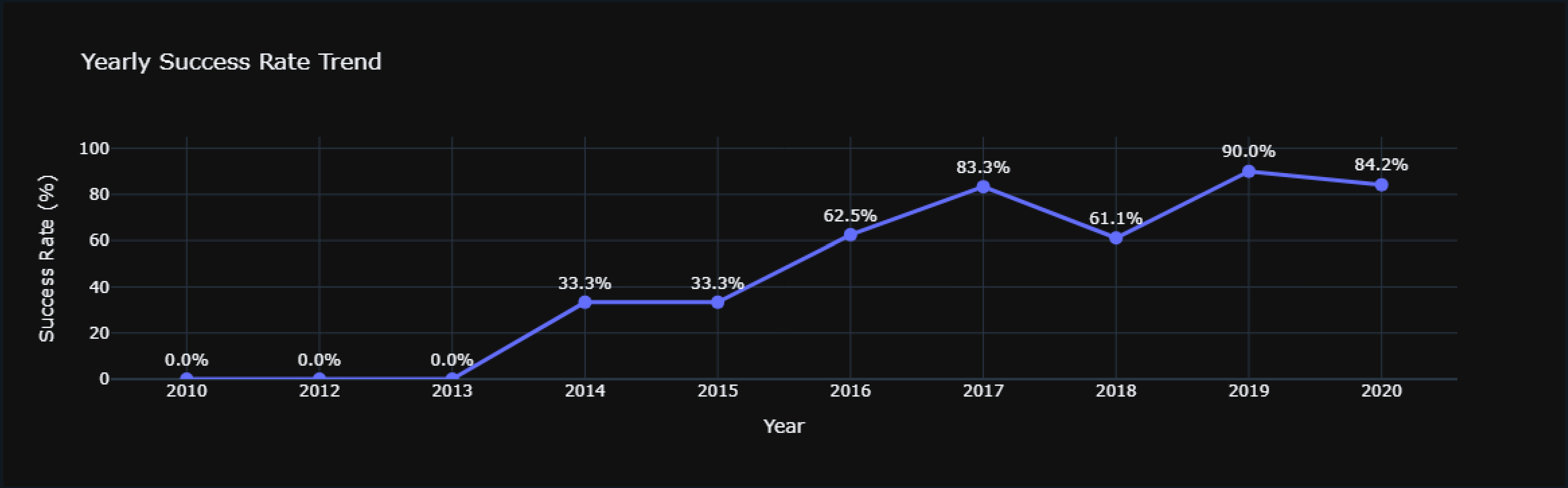


	Outcome	Total
0	True ASDS	41
1	None None	19
2	True RTLS	14
3	False ASDS	6
4	True Ocean	5
5	False Ocean	2
6	None ASDS	2
7	False RTLS	1

Insight: ASDS landings dominate, confirming SpaceX's reliance on drone ships for recovery.

EDA RESULTS – TRENDS

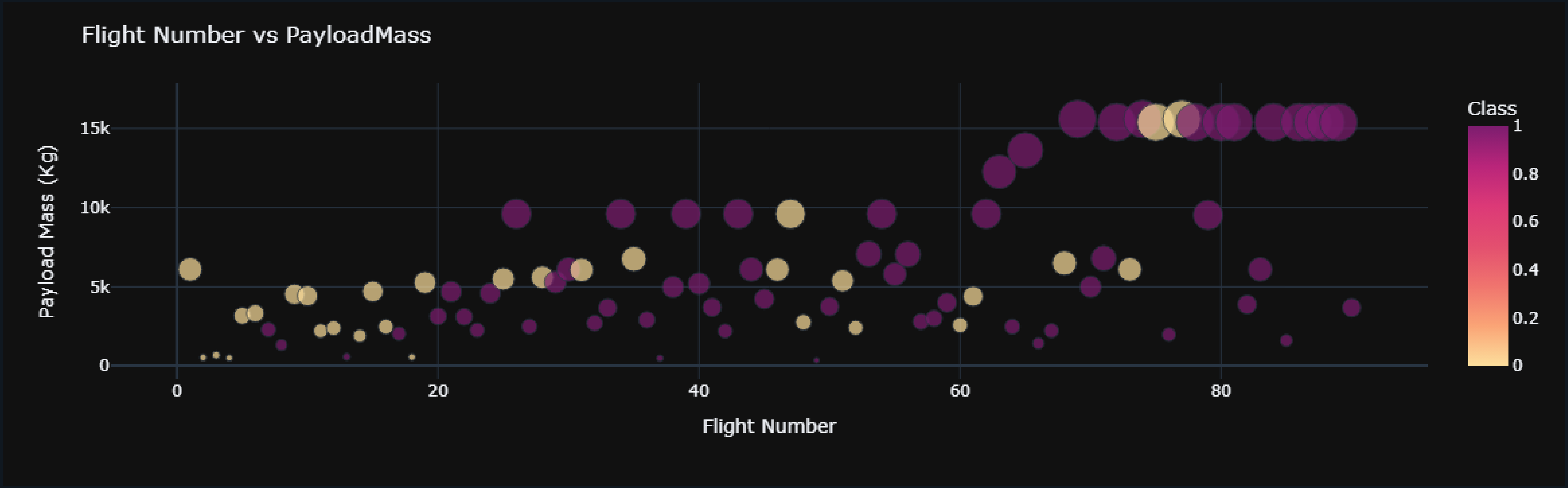
Yearly Success Trend



Insight: Success rate increases sharply after 2015, aligning with the introduction of Block 4/5 boosters and improved landing algorithms.

EDA RESULTS – FLIGHT NUMBER VS PAYLOAD MASS

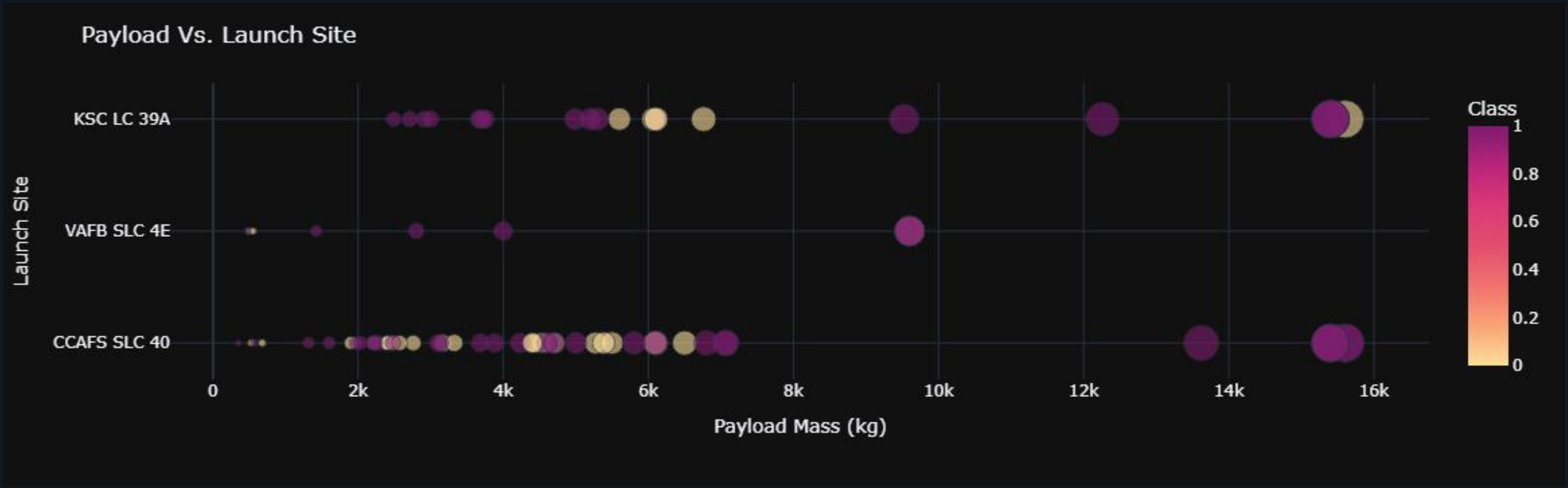
Flight Number vs Payload Mass



Insight: Later missions carry heavier payloads and show higher success rates — evidence of iterative engineering improvements.

EDA RESULTS – PAYLOAD MASS VS LAUNCH SITE

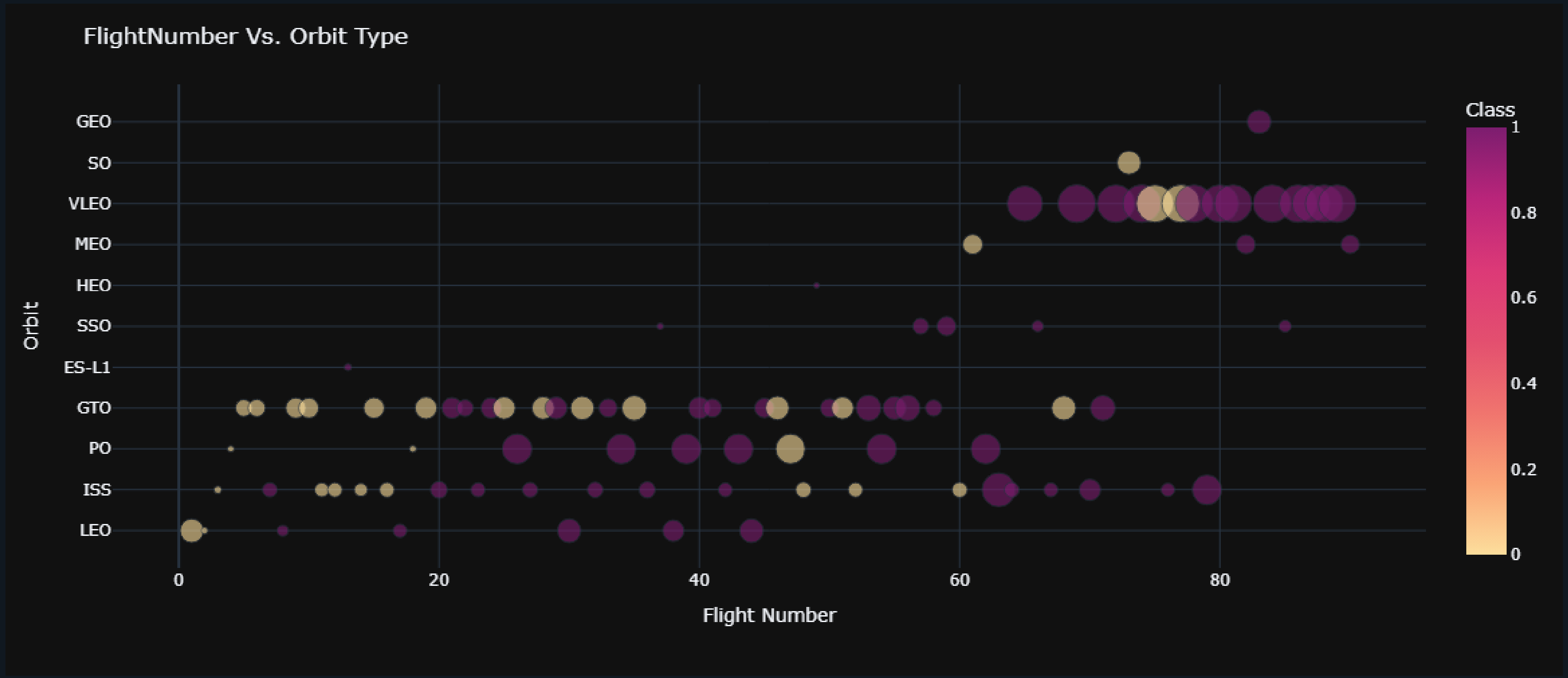
Payload Mass vs Launch Site



Insight: KSC LC-39A handles the heaviest missions, consistent with its infrastructure and historical use for high-energy orbits.

EDA RESULTS – FLIGHT NUMBER VS ORBIT TYPE

Flight Number vs Orbit Type



Insight: Certain orbits (e.g., GTO, ISS) cluster in specific mission eras, reflecting SpaceX's evolving customer base.

SQL RESULTS – MISSION OUTCOMES & PAYLOADS

Total Payload Mass for NASA

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

45,596 kg

NASA is a major contributor to SpaceX's early manifest, especially ISS resupply missions.

Average Payload Mass for F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass FROM SPACEXTBL WHERE "Booster_Version" = 'F9 v1.1'
```

2,928.4 kg

Early Falcon 9 versions carried significantly lighter payloads compared to Block 5.

First Successful Ground Landing

```
%sql SELECT MIN(Date) AS First_Ground_Pad_Landing FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%RTLS%' OR "Landing_Outcome" LIKE '%ground pad%'
```

2015-12-22

A major milestone marking the beginning of reliable reusability.

Booster Versions with Maximum Payload

```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

Block 5 is optimized for reuse and heavy payloads — a key competitive advantage.

SQL RESULTS – BOOSTERS, MAX PAYLOADS, 2015 FAILURES

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTBL
WHERE Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP BY "Landing_Outcome" ORDER BY Outcome_Count DESC;
```

Ranking of Landing Outcomes (2010-06-04 → 2017-03-20)

Landing 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

Insights

•Early missions had **many “no attempt” outcomes**, reflecting pre-reusability era.

•Drone ship landings show **equal success and failure counts** in early years — consistent with experimental phases.

•Ground pad landings were fewer but more reliable.

•Ocean landings (controlled/uncontrolled) indicate fallback strategies.

Why this matters

Understanding early landing behavior helps contextualize SpaceX’s rapid improvement and informs Space Y’s expectations for early-stage reusability.

SQL RESULTS – LANDING OUTCOME RANKING

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

Failures on Drone Ship Landings in 2015

Month	Booster Version	Launch Site	Landing Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Insights

→ Both failures occurred early in 2015, before the first successful RTLS landing in December 2015.

→ Both failures used **F9 v1.1**, a pre-Block-5 booster with lower landing reliability.

→ Both occurred at **CCAFS LC-40**, indicating early drone ship landing challenges.

Why this matters

This SQL slice highlights the transition period before SpaceX achieved consistent reusability — valuable context for Space Y's early operational planning.

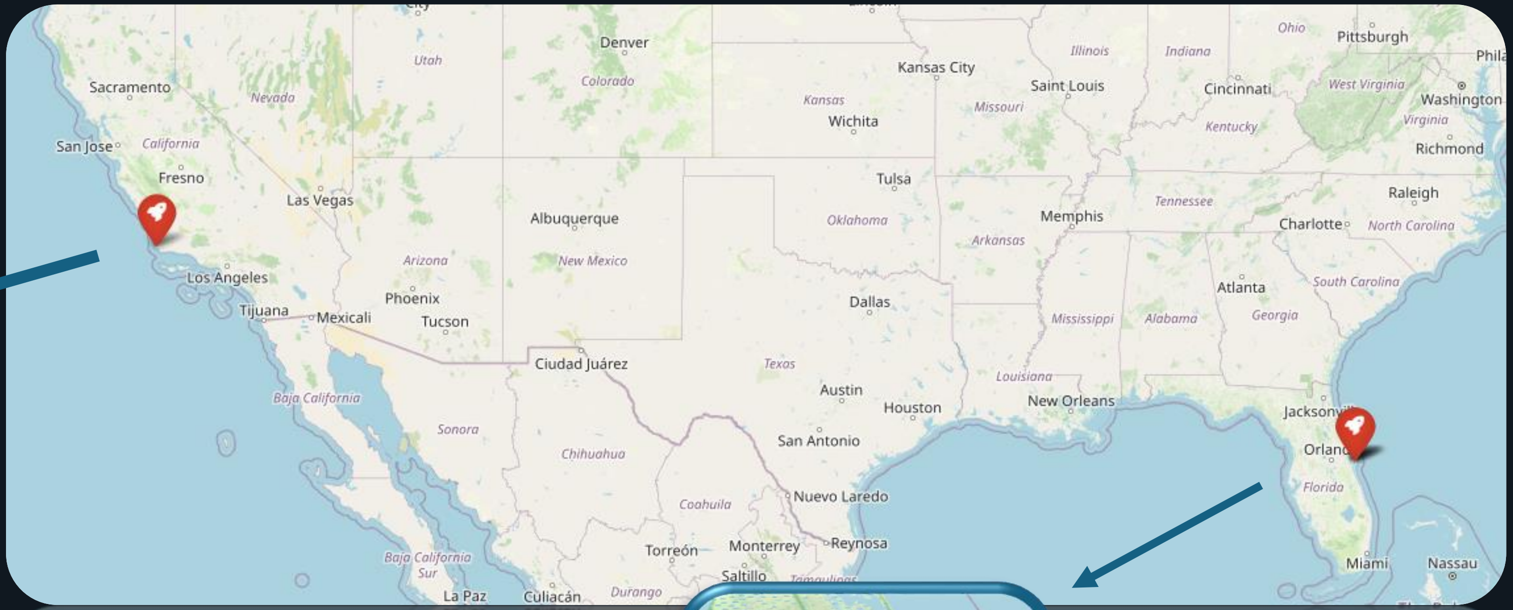
06. INTERACTIVE VISUALIZATIONS - FOLIUM

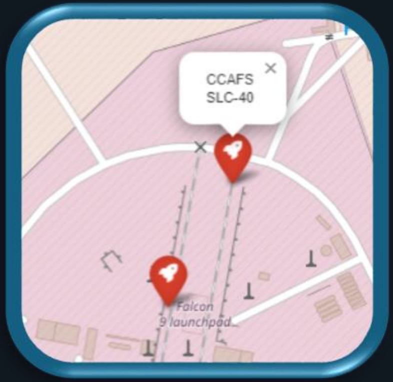
INTERACTIVE VISUAL ANALYTICS – FOLIUM (MAPS)

Geospatial Analysis with Folium
Using Folium, I created interactive maps to visualize:

1. Launch Site Locations

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



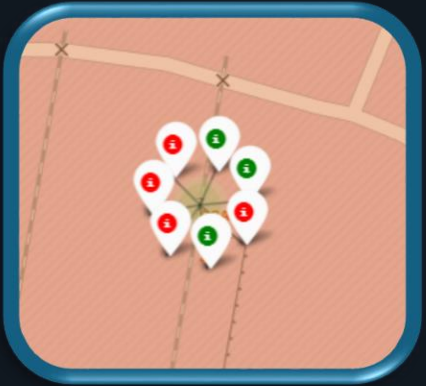


All launch sites are located near coastlines → essential for ASDS landings.

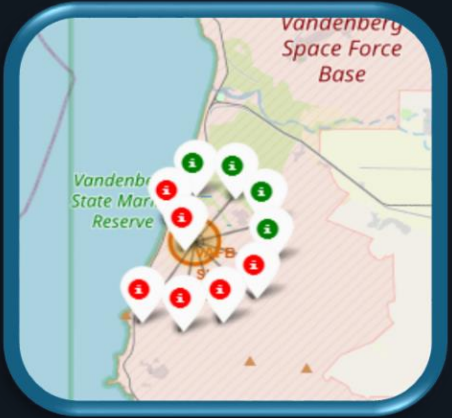
INTERACTIVE VISUAL ANALYTICS – FOLIUM (MAPS)

2. Success vs Failure Markers

CCAFS LC-40	26
KSC LC-39A	13
VAFB SLC-4E	10
CCAFS SLC-40	7



CCAFS SLC-40



VAFB SLC-4E

KSC LC-39A

CCAFS LC-40

Failure markers cluster in early years, confirming learning curve.

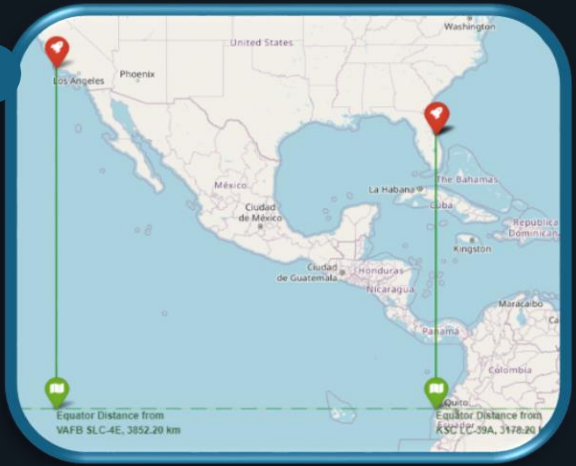
INTERACTIVE VISUAL ANALYTICS – FOLIUM (DISTANCES)

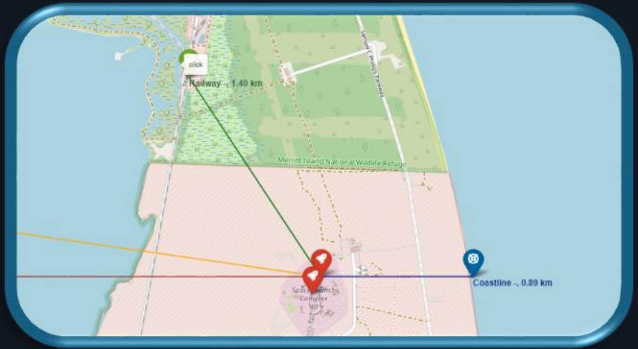
3. Proximity Calculations

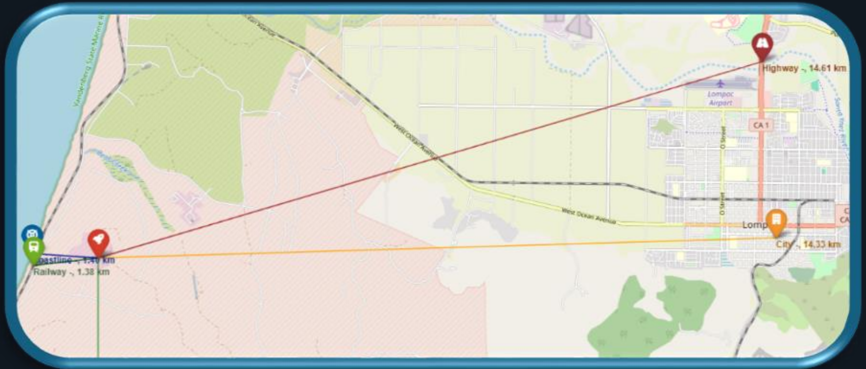
For each launch site, calculated distance to:

1. Equator
2. Coastline
3. Railways
4. Highways
5. Nearest City

1





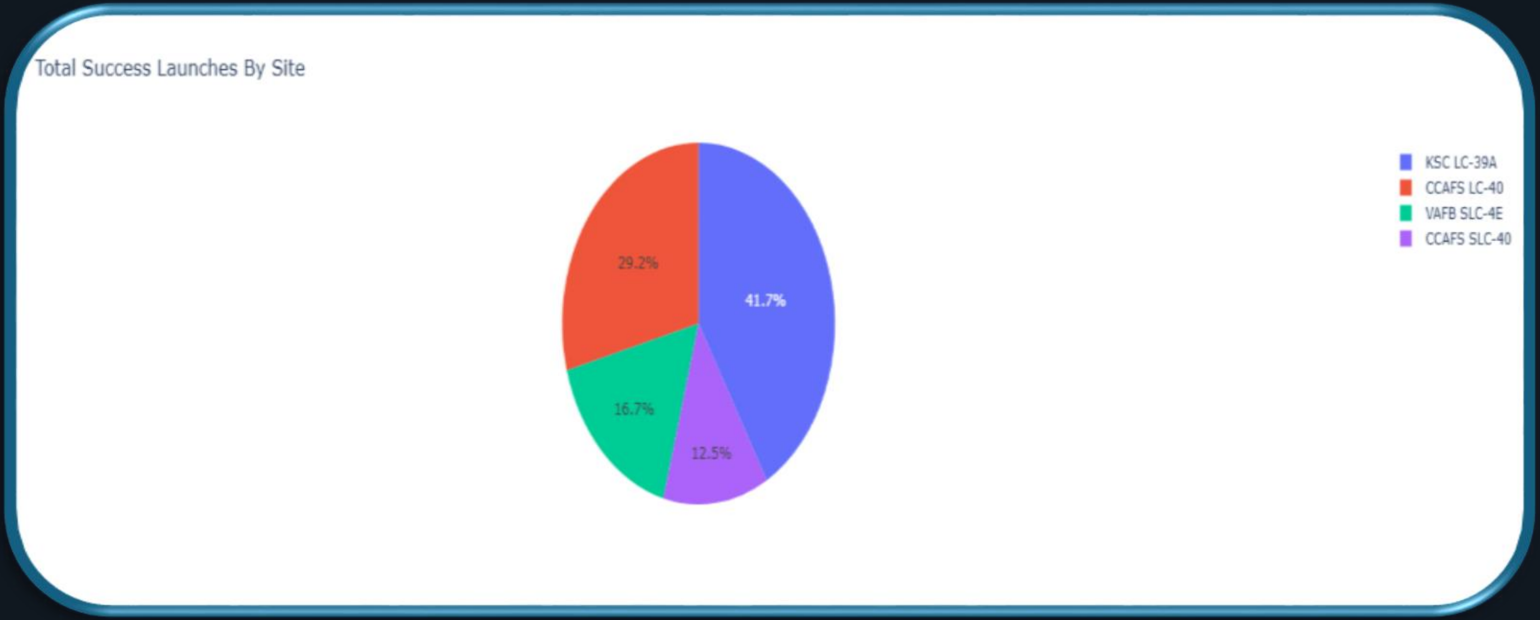


Proximity to infrastructure (roads, rail) supports logistics and booster transport.

06. INTERACTIVE VISUALIZATIONS – PLOTLY DASH

INTERACTIVE VISUAL ANALYTICS – PLOTLY DASH

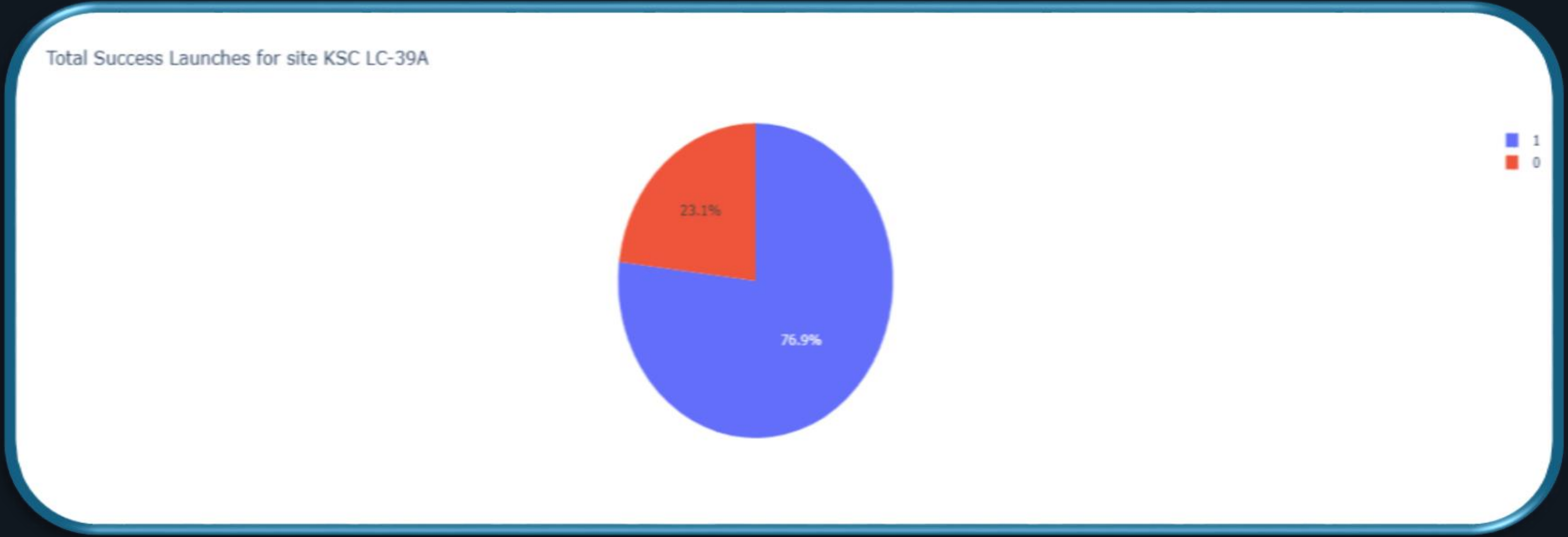
Launch Success Count for All Sites



The launch site **KSC LC-39 A** had the most successful launches, with 41.7% of the total successful launches.

INTERACTIVE VISUAL ANALYTICS – PLOTLY DASH

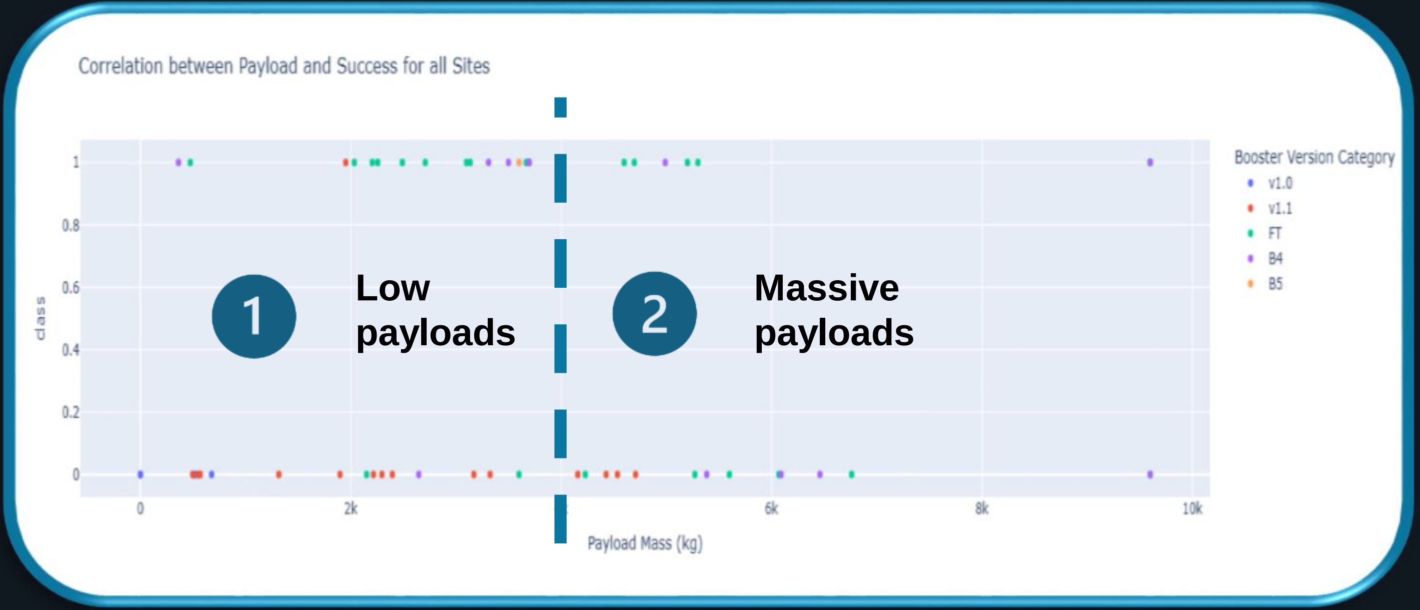
Launch Site with Highest Launch Success Ratio

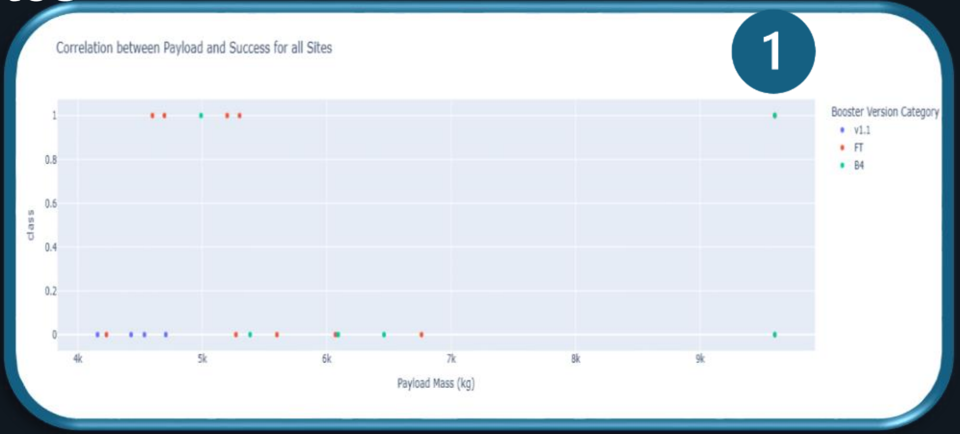


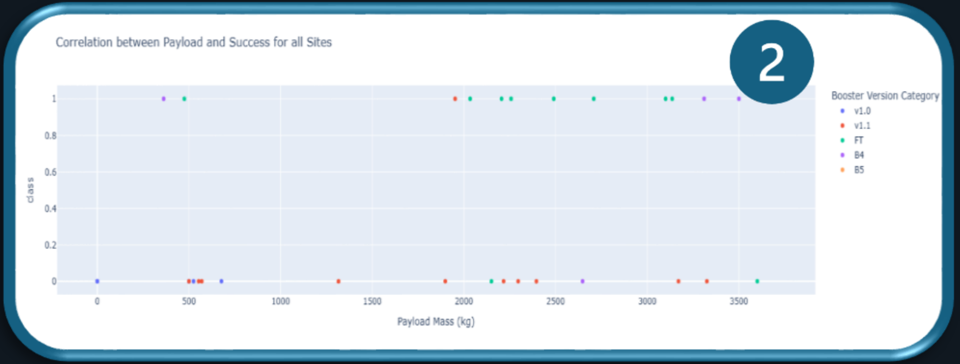
The launch site **KSC LC-39 A** also had the highest rate of successful launches, with 76.9% success rate.

INTERACTIVE VISUAL ANALYTICS – PLOTLY DASH

Launch Outcome vs Payload Scatter Plot for all Sites







Plotting the launch outcomes vs payload for all sites shows a gap around 4000kg, so it makes sense to Split the data into 2 ranges:

- 0 – 4000 kg (low payloads)
- 4000 – 10000 kg (massive payloads)

From these 2 plots, we can observe that **the success for massive payloads is lower than that for low payloads.**

07. PREDICTIVE ANALYSIS

PREDICTIVE ANALYSIS METHODOLOGY

ML Pipeline Overview

1. Data Preparation

- Used dataset_part3.csv with dummy variables
- Normalized numeric features
- Split into train/test sets (80/20)

2. Models Tested

- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree Classifier
- Random Forest Classifier

3. Hyperparameter Tuning

- Logistic Regression: C, penalty, solver
- SVM: C, gamma, kernel
- Decision Tree: criterion, max_depth, min_samples_split, min_samples_leaf, max_features, splitter

4. Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

PREDICTIVE ANALYSIS RESULTS – MODELS & METRICS

Model Comparison (GridSearchCV-tuned)

Insights

- All models generalize similarly on test data (0.833 accuracy).
- Decision Tree achieves the **highest training accuracy**, indicating it captures more complex patterns.
- Logistic Regression provides a strong baseline with minimal complexity.
- SVM performs well despite using a non-linear kernel (sigmoid), showing non-linear relationships in the data.

Why this matters

Space Y can rely on multiple model types to predict landing success — the signal is strong and consistent across algorithms.

```
Tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'}
Accuracy for decision tree classifier: 0.9017857142857144
Accuracy for decision tree classifier on the test data using the method score: 0.8888888888888888
```

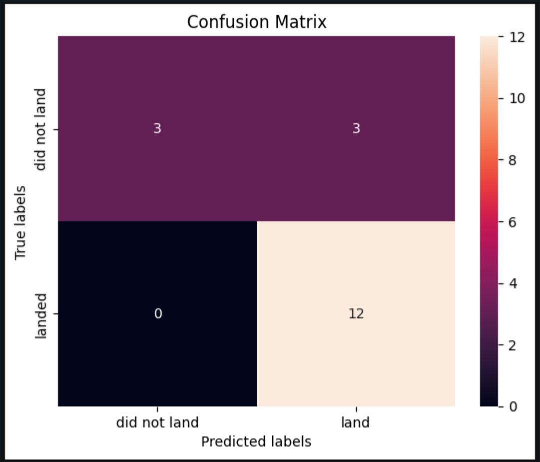
```
Tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
Accuracy for logistic regression: 0.8464285714285713
Accuracy for logistic regression using the method score: 0.8333333333333334
```

```
Tuned hpyerparameters :(best parameters)  {'C': np.float64(1.0), 'gamma': np.float64(0.03162277660168379), 'kernel': 'sigmoid'}
Accuracy for support vector machine: 0.8482142857142856
Accuracy for support vector machine on the test data using the method score: 0.8333333333333334
```

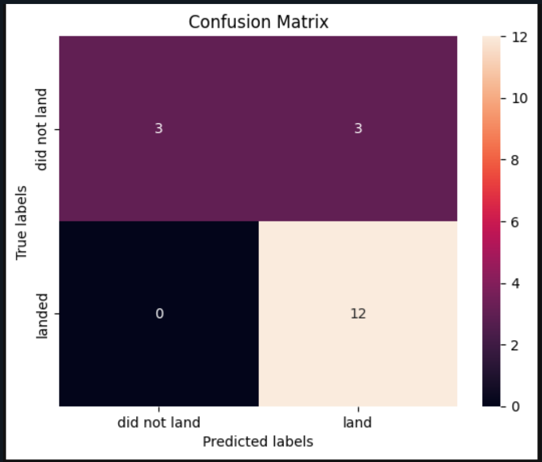
	Algorithm	Accuracy
0	Logistic Regression	0.846429
1	SVM	0.848214
2	KNN	0.848214
3	Decision Tree	0.901786

PREDICTIVE ANALYSIS RESULTS – CONFUSION MATRICES & INTERPRETATION

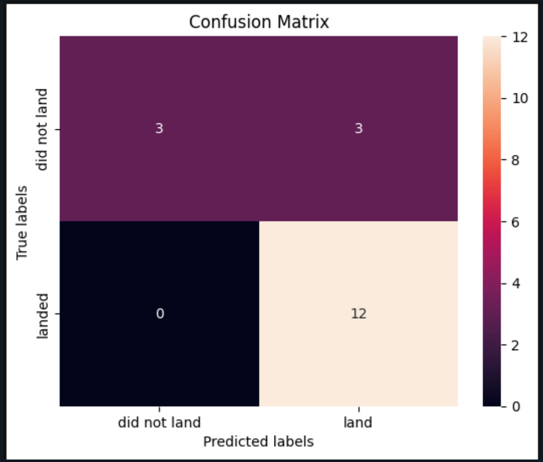
Confusion Matrix — Logistic Regression



Confusion Matrix — SVM



Confusion Matrix — Decision Tree



- Models correctly classify most “landed” outcomes.
- False negatives (predicting failure when it landed) are low — important for operational planning.
- False positives (predicting landing when it fails) are slightly higher, reflecting the difficulty of predicting borderline missions.

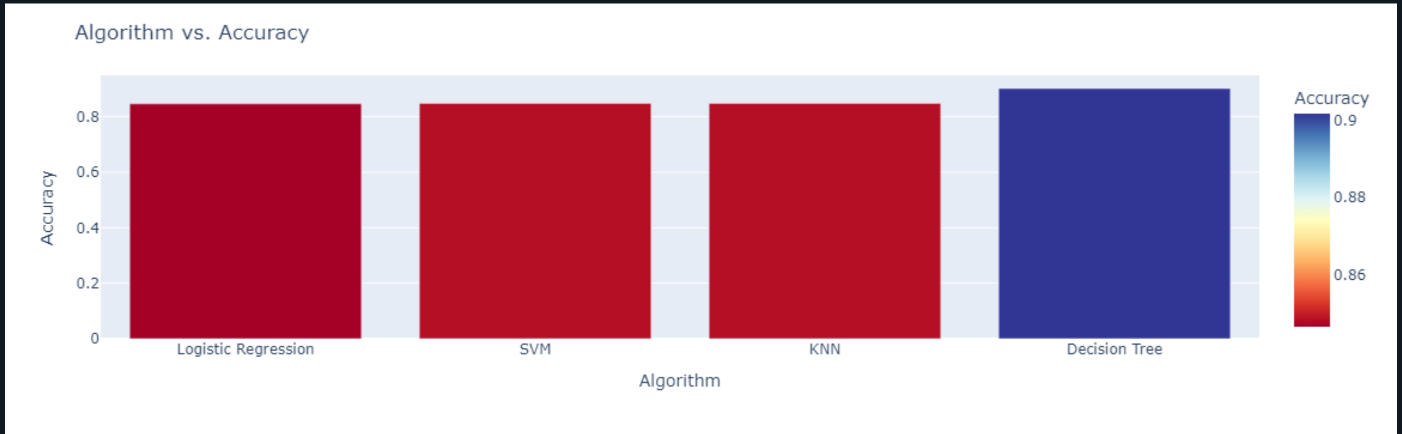
Why this matters

Confusion matrices reveal *how* the model makes mistakes — critical for risk-sensitive decisions like booster recovery.

PREDICTIVE ANALYSIS RESULTS – FEATURE IMPORTANCE

Top Predictive Features (Decision Tree / Random Forest)

1. Orbit Type
2. Payload Mass
3. Booster Version
4. Landing Pad Availability
5. Launch Site
6. Reused Count
7. Grid Fins/ Legs



Insights

- Orbit type is the strongest predictor → mission profile dictates landing feasibility.
- Payload mass influences fuel margins for landing burns.
- Booster version matters — Block 5 boosters are far more reliable.
- Landing pad availability is a major operational constraint.
- Launch site affects trajectory and landing options.

08. KEY INSIGHTS

KEY INSIGHTS FOR Space Y

1. Launch Pricing Is Predictable

Payload mass and orbit type explain most of the variance in launch price. → Space Y can benchmark pricing with confidence.

2. Reusability Is Systematic

Landing success is predictable with ~89% accuracy. → Space Y can design missions to maximize reuse probability.

3. Infrastructure Drives Reuse

Landing pad availability and launch site strongly influence outcomes. → Early investment in landing infrastructure is essential.

4. Block 5-style Boosters Are Critical

Booster version is a top predictor of success. → Space Y should prioritize a robust, reusable booster design.

5. Operational Maturity Matters

Success rates improve dramatically after 2015. → Space Y should expect a learning curve but rapid improvement.

LIMITATIONS & FUTURE WORK

Limitations

- No access to real-time telemetry (fuel margins, thrust, weather).
- Some scraped fields contain missing or inconsistent values.
- Landing outcome labels simplified into binary classes.
- Payload mass sometimes missing for classified missions.
- No cost data directly available — price estimation is inferred.

Future Work

- Integrate weather and wind-shear data for better landing predictions.
- Add booster age, refurbishment cycles, and flight history.
- Incorporate real-time telemetry if available.
- Expand dataset to include Falcon Heavy and Starship.
- Build a full pricing model using regression + cost modeling.
- Deploy the ML model as an API for Space Y's operations team.

09. CONCLUSIONS

CONCLUSION

What This Project Demonstrated

- SpaceX’s launch pricing and reuse behavior can be modeled using publicly available data.
- Launch price is strongly influenced by **payload mass**, **orbit type**, and **booster reuse**.
- First-stage landing success is **predictable with ~89% accuracy** using mission features.
- SQL, EDA, geospatial analysis, and ML together provide a complete operational picture.

Implications for Space Y

- **Pricing:** Space Y can benchmark competitive launch prices using regression insights.
- **Operations:** Mission planning can be optimized for reusability using ML predictions.
- **Infrastructure:** Landing pad availability is a major driver of reuse success.
- **Strategy:** Space Y can anticipate competitor behavior and design a more efficient launch system.

Final Takeaway

Data-driven analysis provides Space Y with a strong foundation for entering the commercial launch market and competing effectively with SpaceX.

10. APPENDIX

LINKS

GitHub – Project Capstone IBM SpaceX

DATA PIPELINE DIAGRAM