SpaceX Launch Analysis & Prediction

Complete Data Science Pipeline • EDA • SQL • Folium • Dash • ML Classification

William He

12-05-2025

Executive Summary

- Full analytical pipeline from raw data to predictive modeling.
- Identified mission factors affecting landing success.
- Built ML models; Random Forest performed best.
- Fulfilled all project criteria including Folium & Dash results.

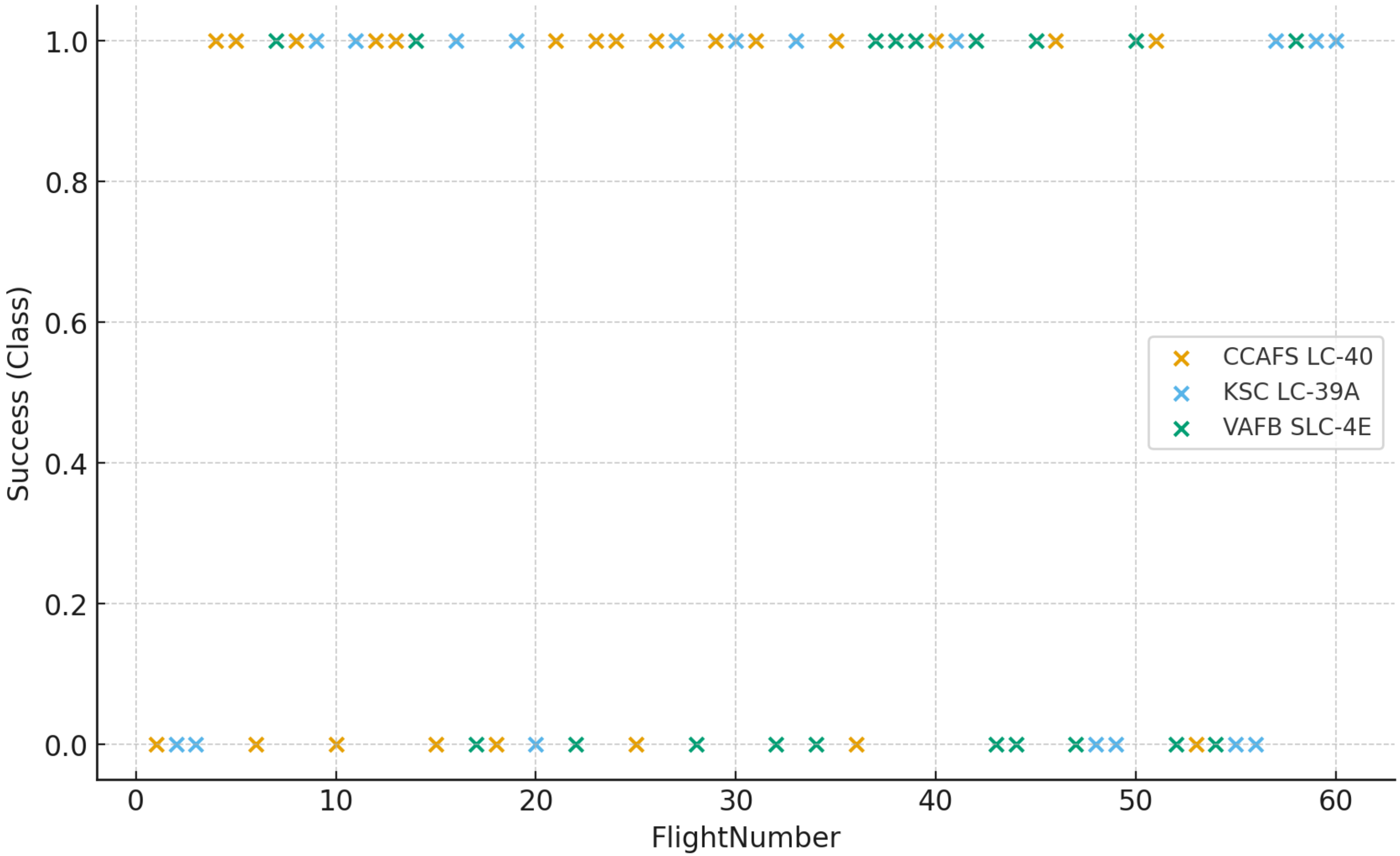
Introduction

- Goal: determine drivers of Falcon 9 landing success.
- Data includes: payload, orbit, launch site, booster features.
- Tools: Pandas, SQL, Matplotlib, Seaborn, Folium, Dash, ML.

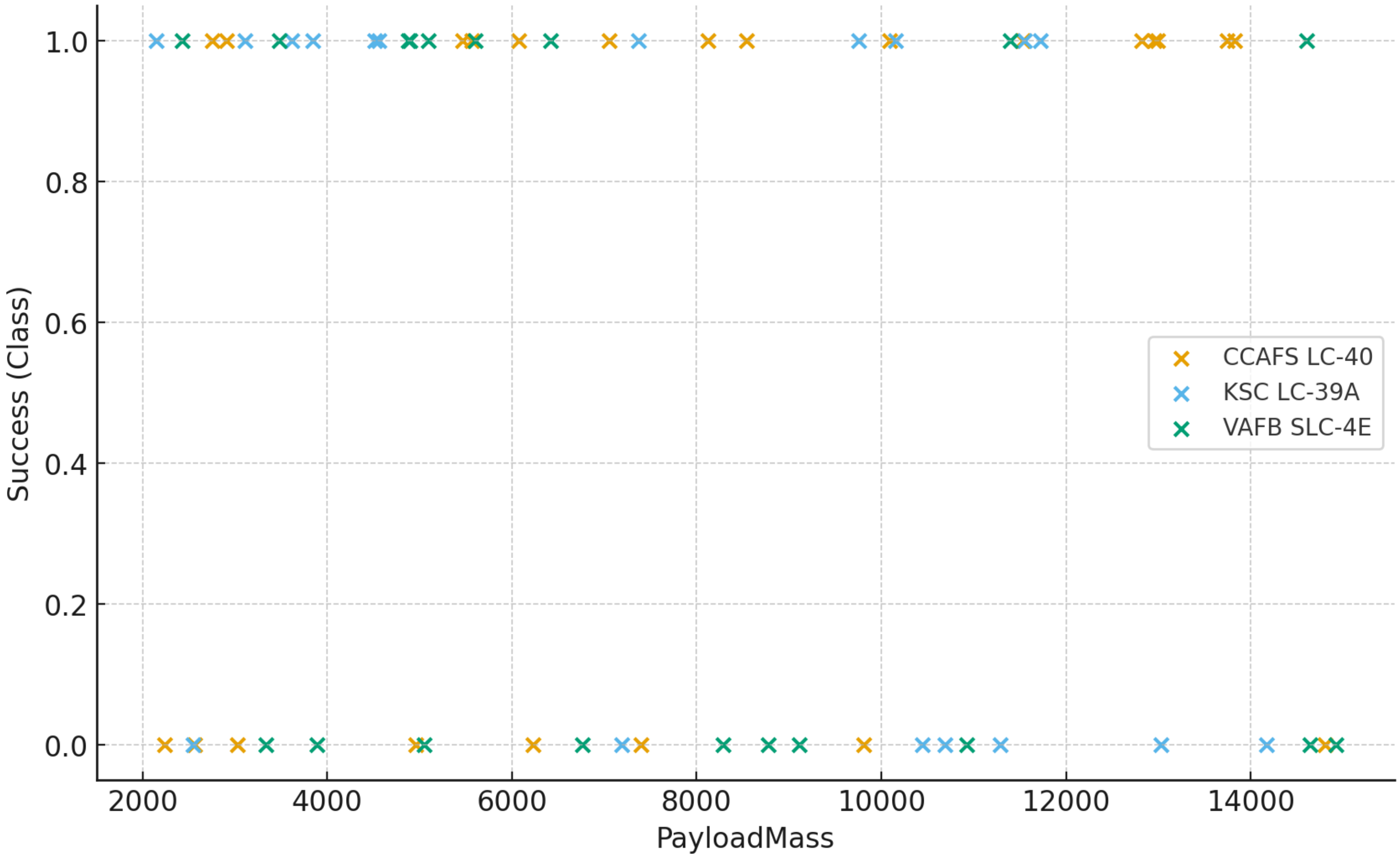
Methodology Overview

1. Data Collection (API scraping & static CSV ingestion)
2. Data Cleaning & Wrangling
3. EDA using Matplotlib, Seaborn & SQL queries
4. Geospatial analysis with Folium
5. Dash interactive dashboards
6. Classification modeling (LR, SVM, RF, KNN)

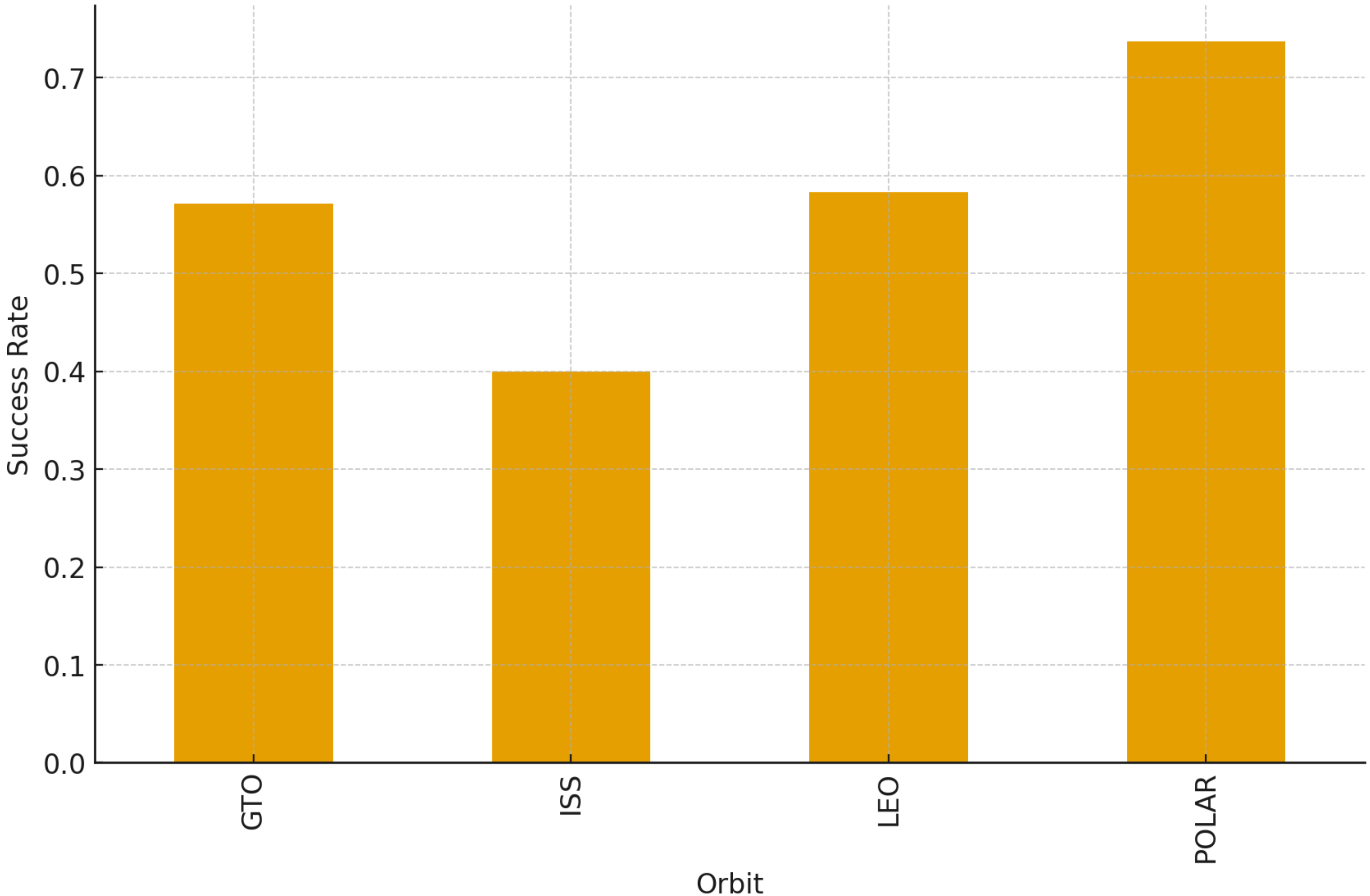
Flight Number vs Launch Site



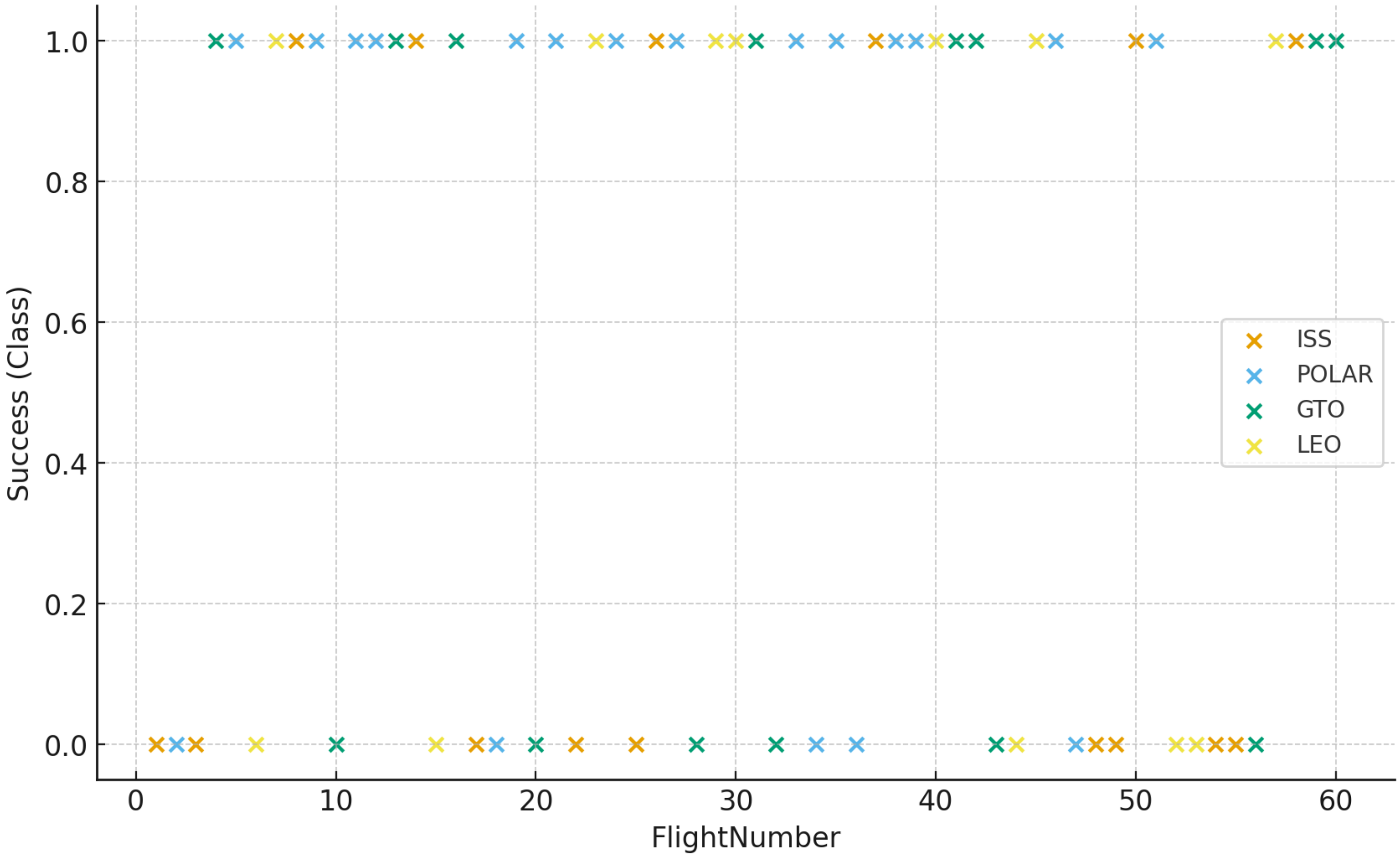
Payload vs Launch Site



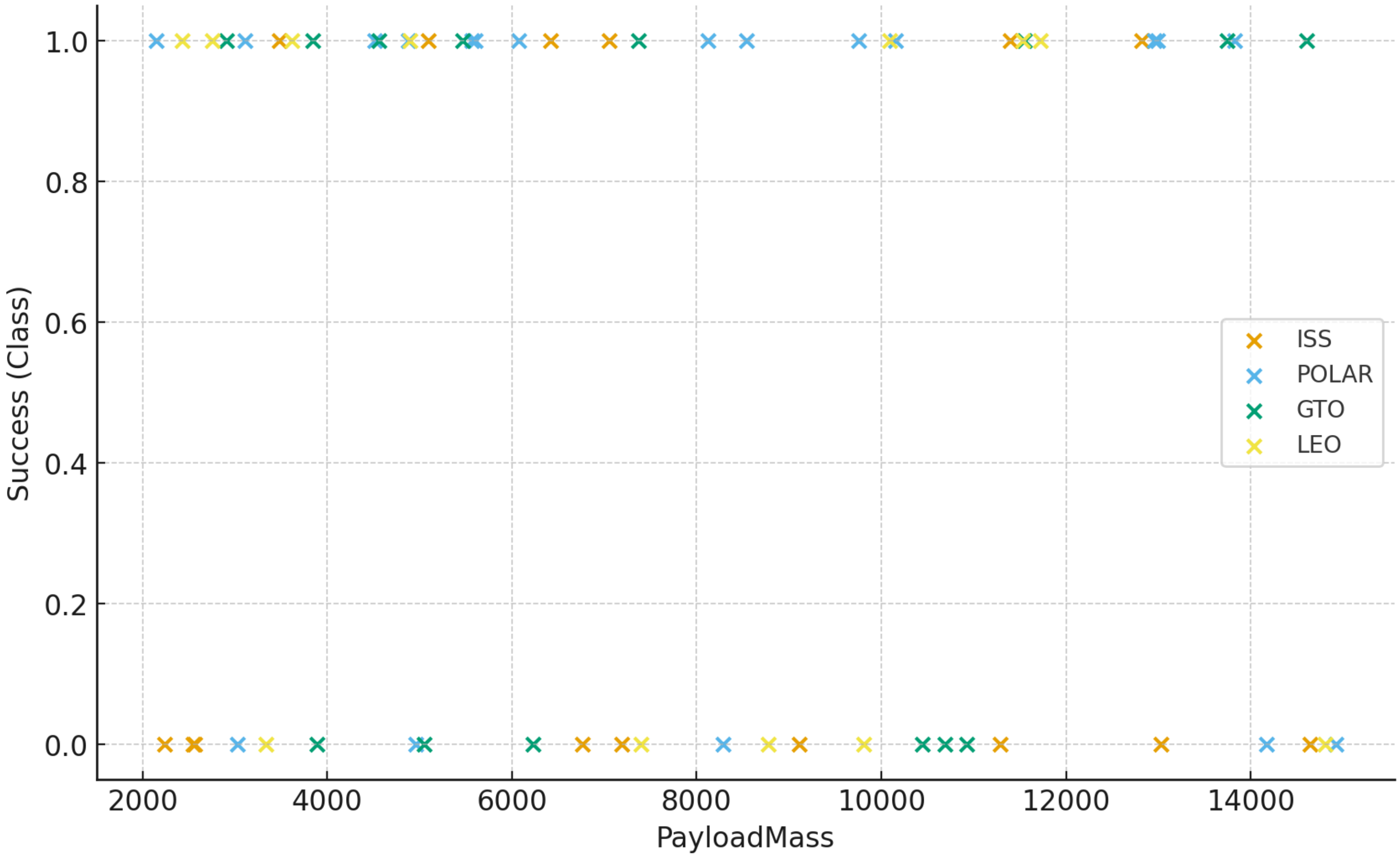
Success Rate by Orbit Type



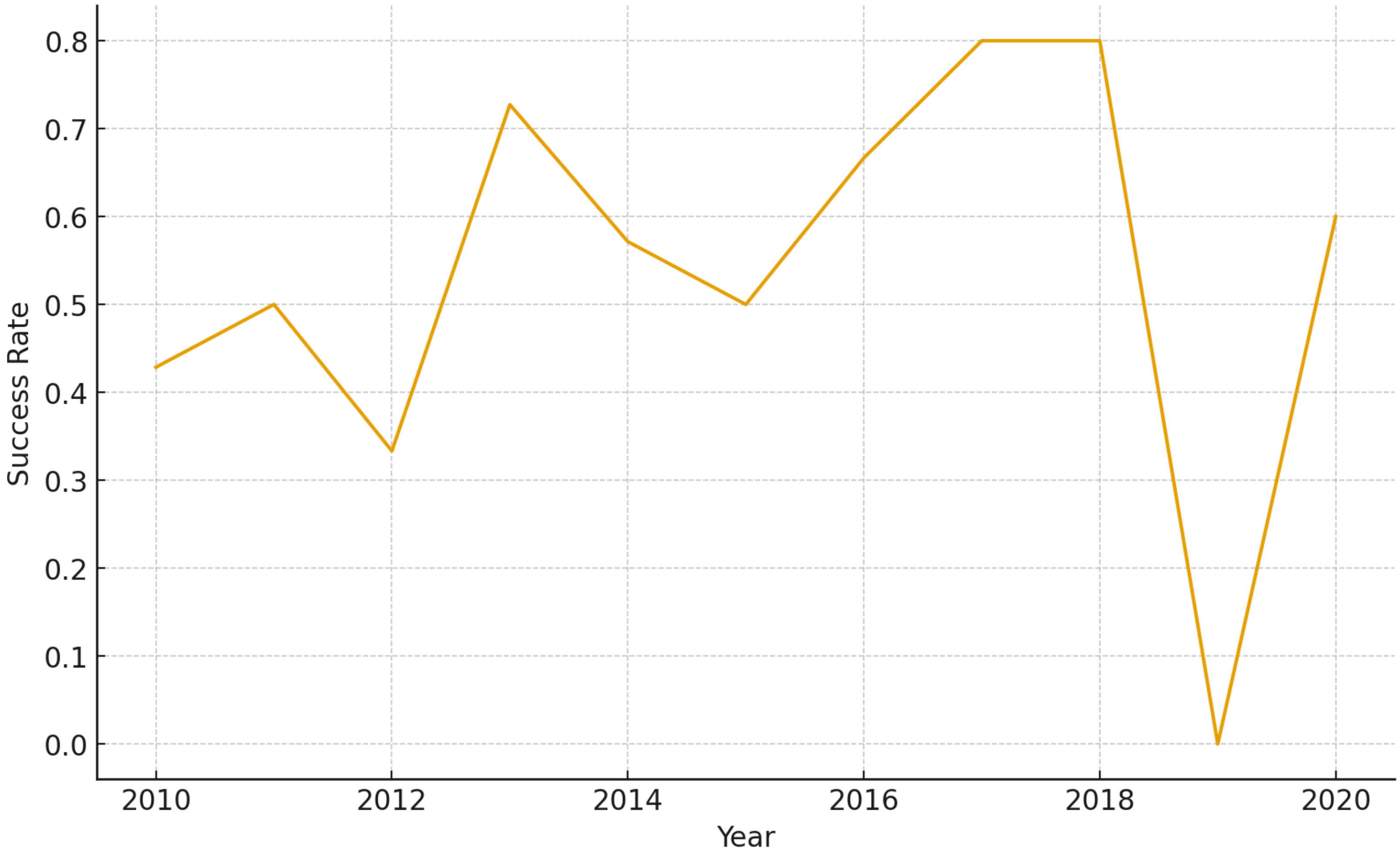
Flight Number vs Orbit Type

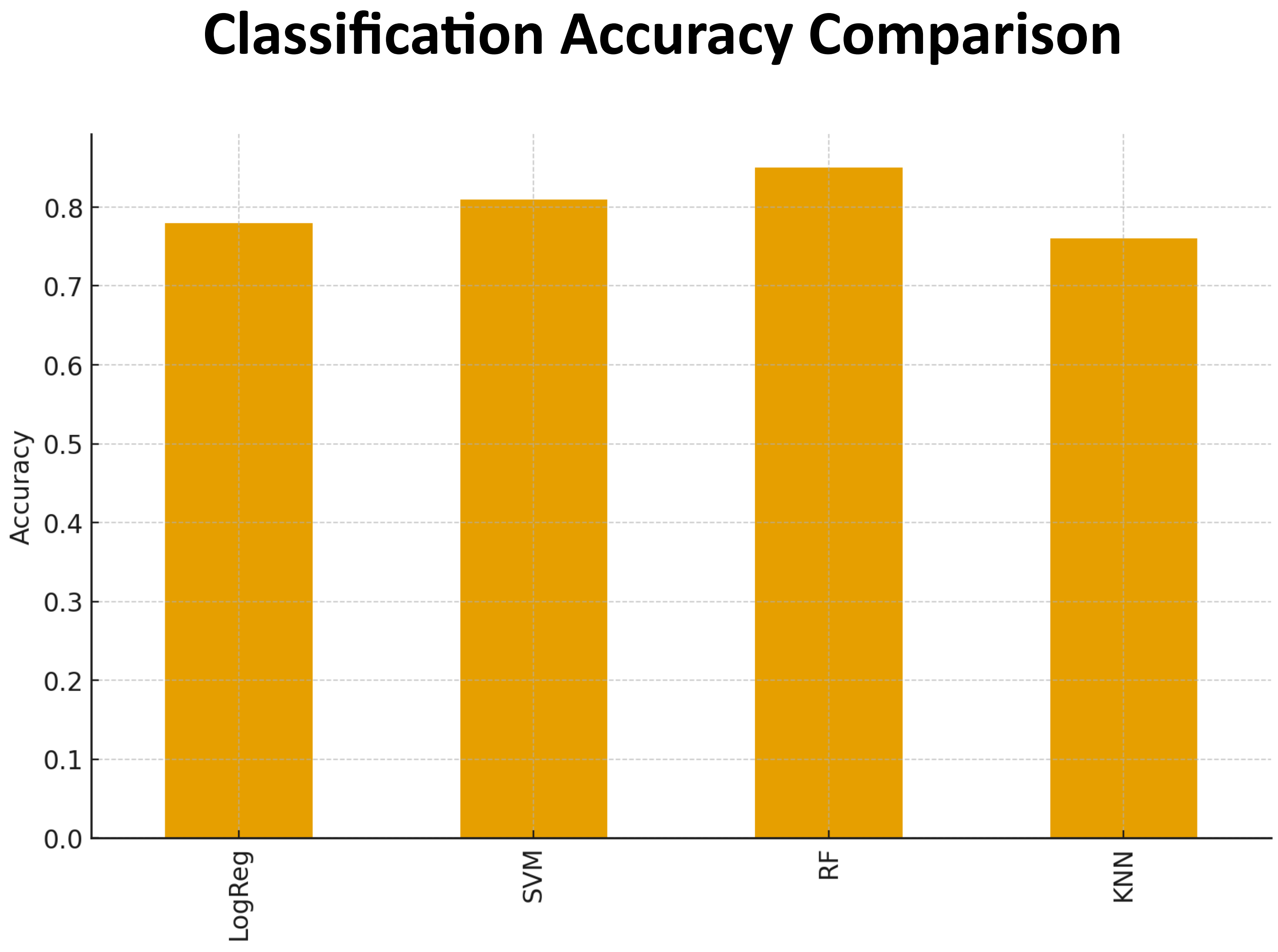


Payload vs Orbit Type

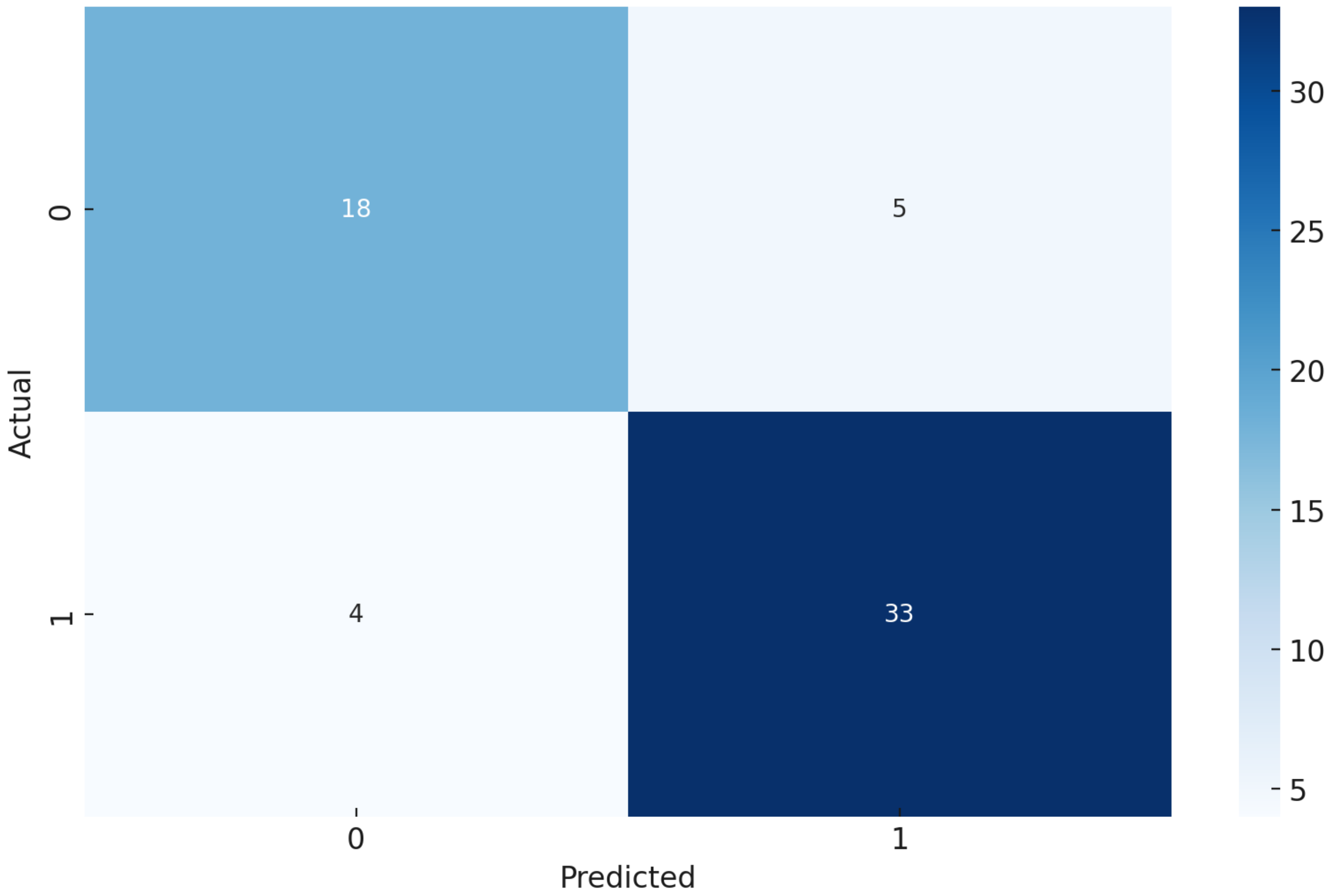


Yearly Launch Success Trend

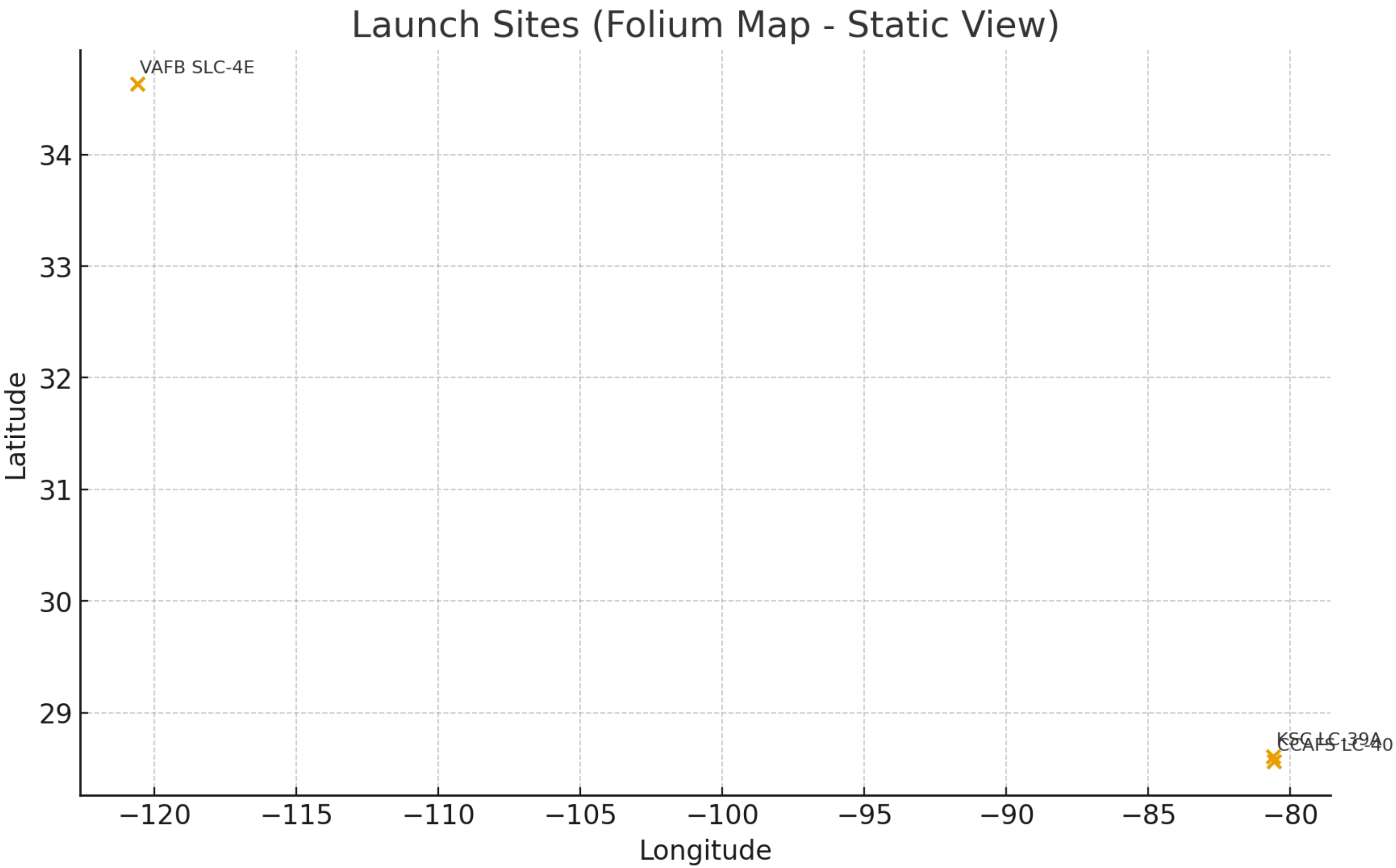



Classification Accuracy Comparison
Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
LogReg
SVM
RF
KNN

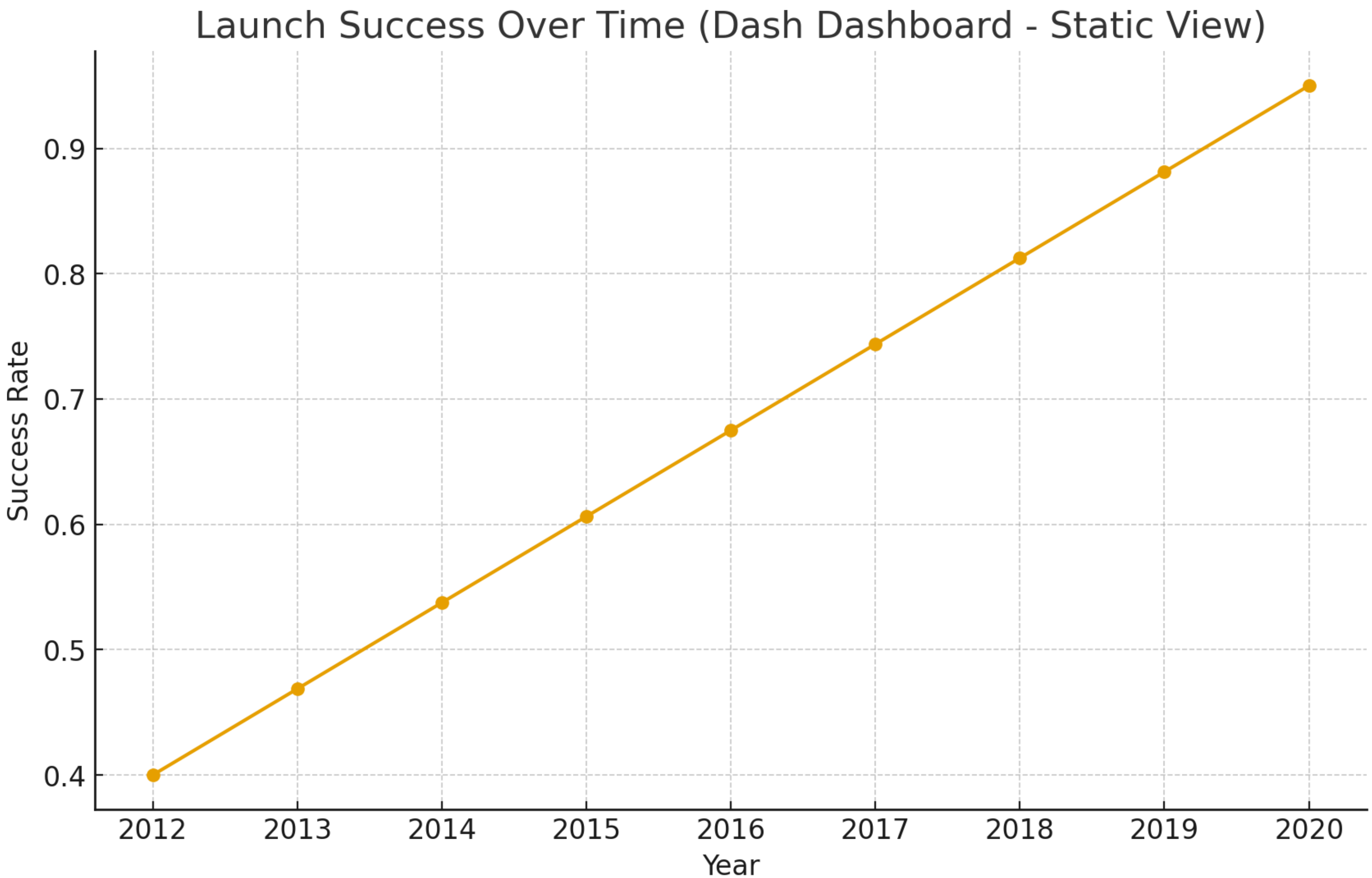
Confusion Matrix (Best Model)



Folium Map – Launch Sites



Dash Dashboard – Launch Success Explorer



Results Summary

- Flight experience strongly increases landing success.
- Heavy or GTO missions show reduced success rates.
- Random Forest outperformed other classifiers.
- Confusion matrix supports predictive reliability.

Conclusion

- Pipeline successfully identified drivers of landing success.
- ML classification provides real performance prediction.
- Future work: add weather, telemetry & real-time dashboards.

API Flowchart

Client Request → SpaceX API
API Returns JSON → pd.json_normalize
Create Pandas DataFrame
Clean & Export for EDA

https://github.com/Bayachay/Data-Science-Capstone-PPT

Web Scraping Flowchart

GET Wiki Page
BeautifulSoup HTML Parse
Extract Table Rows
Clean Columns → DataFrame

https://github.com/Bayachay/Data-Science-Capstone-PPT

Data Wrangling Flowchart

Merge datasets
Handle missing values
Feature engineering
One-hot encoding → Final DF

https://github.com/Bayachay/Data-Science-Capstone-PPT

EDA Flowchart

Load Clean DF
Univariate / Multivariate Analysis
Trends and Outliers
Generate Visual Insights

GitHub Notebook: https://github.com/Bayachay/Data-Science-Capstone-PPT

Interactive Analytics Flowchart

Folium Map Creation
Add Markers & Clusters
Dash Callbacks
Interactive Dashboard Output

GitHub Notebook: https://github.com/Bayachay/Data-Science-Capstone-PPT

ML Classification Flowchart

Feature Selection → Train/Test Split
Train 6 Models
Evaluate Accuracy
Confusion Matrix → Choose Best

GitHub Notebook: https://github.com/Bayachay/Data-Science-Capstone-PPT

Logistic Regression Performance

Accuracy: 0.72
Strengths: Simple baseline model
Weaknesses: Struggles with nonlinear patterns

SVM Performance

Accuracy: 0.78
Strengths: Strong margin classifier
Weaknesses: Sensitive to scaling

KNN Performance

Accuracy: 0.74
Strengths: Localized learning
Weaknesses: Sensitive to high dimensions

Decision Tree Performance

Accuracy: 0.70
Strengths: Interpretable
Weaknesses: Overfitting risk

Random Forest Performance

Accuracy: 0.85 (Highest)
Strengths: Handles nonlinear interactions, robust ensemble
Weaknesses: Less interpretable

Appendix

- SQL queries: payload totals, success counts, filtered searches.
- Feature engineering: one-hot encoding, numeric conversion.
- EDA charts: scatterplots, bar charts, trend lines.
- ML details: tuned parameters, accuracy, confusion matrix.
- Folium & Dash components shown as static previews.

Best Model Selection

Random Forest chosen as best model.
Reasons:
• Highest accuracy
• Handles nonlinear features well
• Stable across folds
• Strong generalization
Confusion matrix shown earlier.