IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

José Ángel Marrero Domínguez
01/09/2025

Outline

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

Executive Summary

The project followed a structured data science pipeline to ensure a systematic and reproducible analysis:

1. **Data Collection**
   Data was gathered from reliable sources and imported into a structured format suitable for analysis.
2. **Data Wrangling**
   The dataset was cleaned by handling missing values, correcting inconsistencies, and ensuring appropriate data types for subsequent analysis.
3. **Data Processing**
   Relevant features were engineered and transformed. Standardization and encoding techniques (such as one-hot encoding) were applied to prepare the dataset for modeling.
4. **Exploratory Data Analysis (EDA)**
   Both visualization tools and SQL queries were used to uncover trends, correlations, and patterns in the data.
5. **Interactive Visual Analytics**
   Geographic and interactive insights were created using Folium (for maps) and Plotly Dash (for interactive dashboards), enhancing interpretability of results.
6. **Predictive Analysis**
   Various classification models were trained and tested to predict outcomes. Performance was evaluated based on accuracy, precision, recall, and other relevant metrics.
7. **Model Building, Tuning, and Evaluation**
   Classification models were fine-tuned using techniques such as GridSearchCV. The best-performing model was selected based on cross-validation scores and final test accuracy.

Summary of results

The best-performing model was the Decision Tree Classifier, which achieved an accuracy of 86% on the test data. This shows that the model is able to correctly classify the majority of the cases. Additionally, the recall values were similar across most of the models, meaning that their ability to correctly identify positive cases did not differ significantly. Therefore, accuracy became the main differentiating factor. While the Decision Tree performed best in terms of accuracy, it is still important to evaluate other metrics such as precision, recall, and the balance of false positives and false negatives to ensure robust performance. Overall, the Decision Tree stood out as the most effective method among those tested.

Introduction

The project is set in the context of the new commercial space age, where launch costs have dropped and competition has intensified with companies like Virgin Galactic, Rocket Lab, and Blue Origin entering the field. SpaceX currently dominates this market, largely due to the ability to reuse its first stage. This innovation has drastically reduced launch prices, with Falcon 9 missions listed at around $62 million compared to over $165 million from other providers. For a new entrant like Space Y, being able to predict whether the first stage will land successfully is critical. This capability allows a better understanding of the effective cost per launch and provides insight into the economic drivers that shape mission pricing. The task is to use public data to design dashboards and develop machine learning models that estimate landing success and its implications for pricing strategies. The main questions revolve around understanding the probability of a Falcon 9 first stage landing successfully when considering mission parameters such as payload mass, orbit type, launch site, booster version, and reuse history. From there, the focus extends to cost implications: how successful landings translate into reusability, savings, and lower estimated prices per launch. Another central concern is uncovering which factors, such as payload, orbit, site, booster upgrades, or operational history, most strongly influence outcomes. There is also an operational dimension. Examining site-specific or orbit-specific patterns can reveal where landings tend to succeed or fail, while analyzing outcomes over time highlights the impact of hardware upgrades and accumulated experience.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The dataset was gathered from the SpaceX API and from Wikipedia via Web Scrapping
- Perform data wrangling
  - Categorical columns such as Launch Site and Orbit were standardized into consistent labels. The Outcome variable, which originally had eight possible values describing different landing statuses, was recoded into a binary classification variable Y, where 0 indicated an unsuccessful landing and 1 indicated a successful landing.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

The dataset was collected from two primary sources: the **SpaceX API** and the **Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches"**. The process followed these main steps:

Key Steps:

- **Identify data sources** – Publicly available SpaceX API and Wikipedia launch records.
- **Extract data from API** – Used HTTP requests to pull structured JSON data containing launch attributes such as flight number, payload, orbit, booster version, and landing outcomes.
- **Web scraping** – Employed Python's BeautifulSoup library to scrape the Wikipedia table, capturing additional details not available via API.
- **Data integration** – Combined API and web-scraped datasets, aligning common fields and standardizing formats.
- **Data cleaning** – Removed duplicates, handled missing values, and converted categorical attributes into standardized labels.
- **Feature engineering** – Created a binary classification variable *Y* for landing success and encoded categorical variables for modeling.
- **Validation** – Checked the dataset for consistency and completeness before analysis.

Data Collection – SpaceX API

Retrieve Launch Records: Access the SpaceX API endpoint /v4/launches to get main launch data

Booster Version: Query /v4/rockets/<rocket_id> to extract the rocket name for each launch.Launch

Site Details: Query /v4/launchpads/<launchpad_id> to obtain name, longitude, and latitude for each launch site.

Payload Data: Query /v4/payloads/<payload_id> to collect mass_kg and orbit for each payload.

Core Information: Query /v4/cores/<core_id> to extract block, reuse_count, and serial. Track landing_success, landing_type, number of flights, grid fins, reused status, legs, and landing pad.

[Notebook in Github]

```
Start
  |
  ▼
Fetch main launch data (API /v4/launches)
  |
  ├─> Get Booster Version (API /v4/rockets/<rocket_id>)
  |        |
  |        ▼
  |     Extract 'name'
  |
  ├─> Get Launch Site (API /v4/launchpads/<launchpad_id>)
  |        |
  |        ▼
  |   Extract 'name', 'longitude', 'latitude'
  |
  ├─> Get Payload Data (API /v4/payloads/<payload_id>)
  |        |
  |        ▼
  |   Extract 'mass_kg', 'orbit'
  |
  └─> Get Core Data (API /v4/cores/<core_id>)
           |
           ▼
    Extract 'block', 'reuse_count', 'serial'
    Track 'landing_success', 'landing_type', flights, gridfins, reused, legs, landing pad
  |
  ▼
Aggregate all extracted features into lists
  |
  ▼
Convert lists to DataFrame
  |
  ▼
Validate data (missing values, duplicates, types)
  |
  ▼
End
```

Data Collection - Scraping

The Falcon 9 and Falcon Heavy launch data was collected from the Wikipedia page titled “List of Falcon 9 and Falcon Heavy launches”. Using Python’s BeautifulSoup library, all tables with the class "wikitable plainrowheaders collapsible" were identified. Each table row was parsed to extract relevant launch information. The Flight Number was used as a primary identifier to filter valid rows. For each row, attributes including launch date and time, booster version, launch site, payload, payload mass, orbit, customer, launch outcome, and booster landing were extracted.

```
Start → Request Wikipedia Page → Parse HTML with BeautifulSoup
       → Identify All Tables with 'wikitable plainrowheaders collapsible'
       → Loop Through Table Rows → Check if Flight Number is Numeric
           → Extract Date and Time → Extract Booster Version → Extract Launch Site
           → Extract Payload → Extract Payload Mass → Extract Orbit
           → Extract Customer → Extract Launch Outcome → Extract Booster Landing
           → Append to launch_dict
       → Convert launch_dict to Pandas DataFrame → Data Ready for Analysis
End
```

Notebook in Github

Data Wrangling

- Raw SpaceX dataset contained flight details, booster specifications, payload information, and landing outcomes.
- Cleaned and transformed variables to ensure consistency and usability for analysis.
- Standardized categorical columns such as Launch Site and Orbit into uniform labels.
- Recoded the Outcome variable into a binary classification variable Y:
  - 0 for unsuccessful landings
  - 1 for successful landings
- Preserved geographical coordinates, booster characteristics, and other relevant attributes as potential predictive features.
- Handled missing values and formatted columns for compatibility with machine learning models.
- Final dataset structured with clearly defined input features and output target for classification modeling.

[Notebook in Github]

EDA with Data Visualization

- Flight Number vs. Launch Site (scatter plot with hue = Class)
  - Purpose: To visualize how the success of first-stage landings changes over time (as flight numbers increase) across different launch sites.
  - Insight: Higher flight numbers tend to correlate with higher success rates.
- Payload Mass vs. Launch Site (scatter plot with hue = Class)
  - Purpose: To examine the relationship between payload mass and launch site, and how it affects landing success.
  - Insight: Even with heavier payloads, some launch sites show consistently successful first-stage landings.
- Orbit Type vs. Success Rate (bar chart)
  - Purpose: To visually assess how the likelihood of first-stage landing success varies across different orbit types.
  - Insight: Orbits with higher success rates indicate missions where first-stage recovery is more reliable, while lower rates highlight more challenging orbit profiles.
- Flight Number vs. Orbit type (scatter plot with hue = Class)
  - Purpose: To assess whether mission success varies with orbit type as more launches are conducted.
  - Insight: Helps detect orbit-specific trends in landing outcomes over time.
- Payload Mass vs. Orbit type (scatter plot with hue = Class)
  - Purpose: To explore the influence of payload mass on success for different orbit types.
  - Insight: Shows which orbit types are more sensitive to heavier payloads for first-stage recovery.
- Year vs. Average Launch Success Rate (line chart)
  - Purpose: To visualize trends in launch success over time and detect improvements or learning curves.
  - Insight: Displays how SpaceX's overall success rate evolves across years, highlighting technological or operational improvements.

Notebook in Github

EDA with SQL

- Retrieved the unique launch site names from the SpaceX mission dataset.
- Selected 5 records where launch sites begin with the string 'CCA’.
- Calculated the total payload mass carried by boosters launched by NASA (CRS).
- Computed the average payload mass for the booster version F9 v1.1.Identified the date of the first successful landing on a ground pad.
- Listed the names of boosters that successfully landed on a drone ship and carried payloads between 4000 and 6000 kg.
- Counted the total number of successful and failed mission outcomes.
- Found all booster versions that carried the maximum payload mass using a subquery with an aggregate function.
- Selected records displaying the month names, failed landing outcomes on drone ships, booster versions, and launch sites for missions in 2015.
- Ranked the count of landing outcomes (e.g., Failure (drone ship), Success (ground pad)) between 2010-06-04 and 2017-03-20 in descending order.

Notebook in Github

Build an Interactive Map with Folium

Launch Site Markers and Circles

- **Circles:** Highlighted each launch site's geographic location using a colored circle with a popup label.
- **Markers:** Added textual markers at the same coordinates to clearly identify the launch site by name.
- **Purpose:** These provide an intuitive visualization of where SpaceX launch sites are located, making it easy to see their distribution and proximity to each other.

Launch Outcome Markers with MarkerCluster

- **Markers:** Color-coded markers based on launch outcome—green for successful and red for failed launches.
- **MarkerCluster:** Clustered overlapping markers to reduce clutter where multiple launches occur at the same site.
- **Purpose:** This visualizes the success/failure record at each site and highlights sites with higher success rates, allowing quick assessment of reliability.

Distance Lines and Additional Markers

- **Polyline Lines:** Drawn between launch sites and nearby points of interest such as coastlines, cities, highways, and railways.
- **Markers with Distance Labels:** Added markers at these points showing the distance from the launch site.
- **Purpose:** Provides spatial context, showing how close launch sites are to infrastructure or natural features. This helps assess site safety, accessibility, and logistical considerations.

MousePosition Plugin

- **Purpose:** Enabled interactive exploration of map coordinates, allowing precise measurement of distances and locations on the map.

Overall, these map objects together create a rich, interactive visualization that supports spatial analysis of launch site distribution, mission outcomes, and proximities to important geographical features.

Notebook in Github

Build a Dashboard with Plotly Dash

Dashboard Components and Plots

- **Launch Site Dropdown**
  - Interactive dropdown allowing users to select a specific launch site or view all sites.
  - **Purpose:** Enables filtering of data to focus on a particular site, improving insight into site-specific launch outcomes.
- **Success Pie Chart**
  - Displays the total successful launches for all sites when “All Sites” is selected.
  - Shows success vs. failure counts for a specific site when a single launch site is selected.
  - **Purpose:** Provides a quick, visual summary of launch performance and highlights the proportion of successful vs. failed launches, making it easy to compare sites.
- **Payload Range Slider**
  - Allows users to filter launches based on payload mass.
  - **Purpose:** Enables exploration of how payload size affects launch outcomes, supporting scenario analysis for different mission sizes.
- **Payload vs. Outcome Scatter Chart**
  - Scatter plot of payload mass against launch outcome, colored by booster version category.
  - Updates interactively based on selected launch site and payload range.
  - **Purpose:** Visualizes correlations between payload size, booster type, and launch success. Helps identify patterns or trends that could influence first-stage landing success.

Interactions

- Selecting a launch site updates both the pie chart and the scatter plot to reflect data only for that site.
- Adjusting the payload slider dynamically filters the scatter plot to show only launches within the selected payload range

Code in Github

Predictive Analysis (Classification)

Model Development Process:

1. Data Splitting
•Split dataset into features (X) and target (Y)
•Use train_test_split with 80% training, 20% test, random_state=2
•Output: X_train, X_test, Y_train, Y_test
2. Model Selection
•Tested multiple classification algorithms:
- •Logistic Regression (LR)
- •Support Vector Machine (SVM)
- •Decision Tree (DT)
- •K-Nearest Neighbors (KNN)

3. Hyperparameter Tuning
•Use GridSearchCV with 10-fold cross-validation for each model
•Defined parameter grids for each algorithm:
- •**LR:** C, penalty, solver
- •**SVM:** C, kernel, gamma
- •**DT:** criterion, splitter, max_depth, min_samples_split, min_samples_leaf, max_features
- •**KNN:** n_neighbors, algorithm, p

4. Model Fitting
•Fit GridSearchCV on training data
•Find best hyperparameters (best_params_)
•Evaluate cross-validated training accuracy (best_score_)
5. Model Evaluation
•Compute accuracy on test set using .score(X_test, Y_test)
•Generate confusion matrix for detailed error analysis
•Identify false positives and false negatives
6. Comparison of Models
•Compare test accuracy and confusion matrices for all models
•Determine best performing model based on test accuracy and reliability
- •**Example:** Logistic Regression, SVM, and KNN achieved ~0.83 test accuracy
- •Decision Tree performed best on test set (~0.86 accuracy)

[Notebook in Github]

Results

Exploratory Data Analysis (EDA) Results

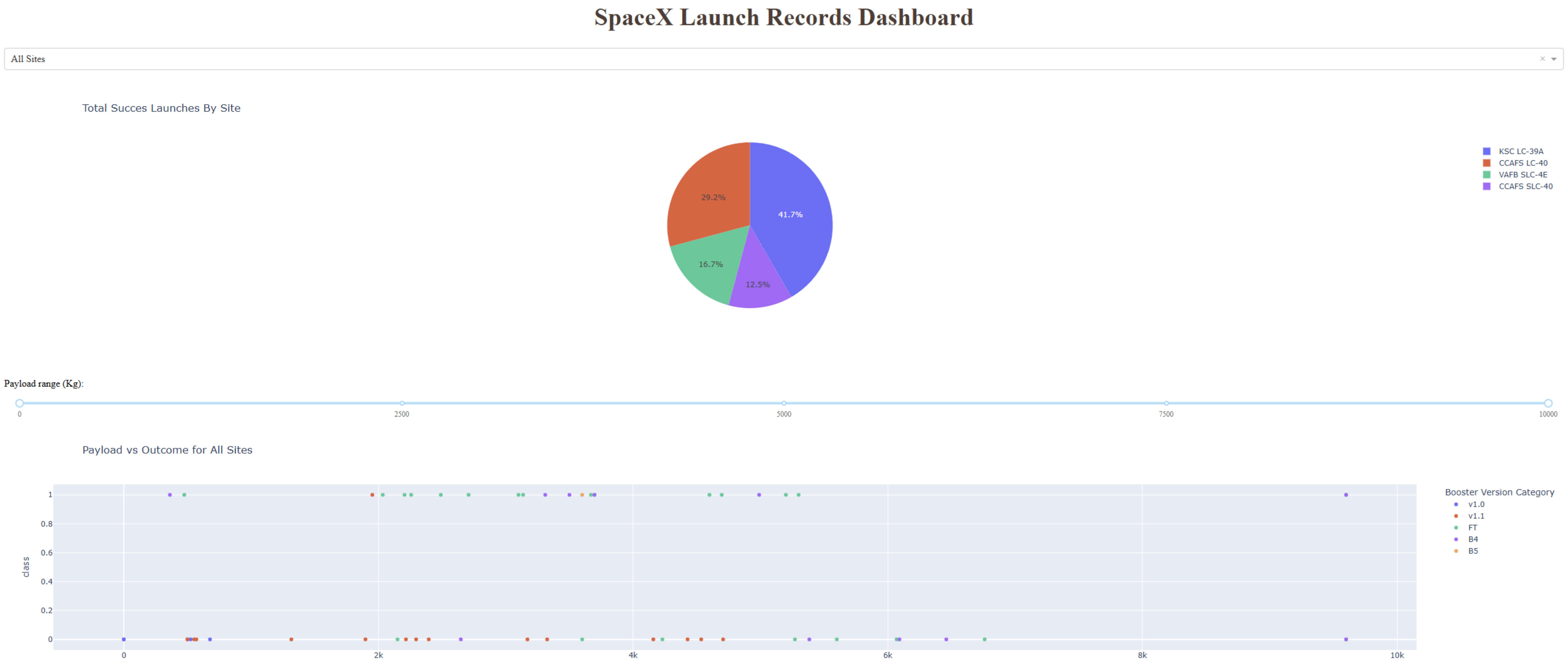
- **Launch Sites Analysis**
  - Unique launch sites identified: *CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40.*
  - Flight success appears to increase with FlightNumber.
  - Certain launch sites have higher payloads and success rates.
- **Payload vs Launch Outcome**
  - Scatter plots show heavier payloads do not necessarily reduce landing success.
  - Booster version and orbit type affect success probability.
- **Orbit Analysis**
  - Bar chart of success rate by orbit type highlights which orbits are easier for successful launches.
- **Trend Analysis**
  - Line chart of average success rate per year shows SpaceX's success rate improving over time.

Predictive Analysis Results

- **Model Training**
  - **Logistic Regression:**
    - Best hyperparameters: C=0.01, penalty='l2', solver='lbfgs'
    - CV Accuracy: 0.8464
    - Test Accuracy: 0.8333
  - **Support Vector Machine (SVM):**
    - Best hyperparameters: C=1.0, gamma=0.0316, kernel='sigmoid'
    - CV Accuracy: 0.8482
    - Test Accuracy: 0.8333
  - **Decision Tree:**
    - Best hyperparameters: criterion='entropy', max_depth=2, max_features='sqrt', min_samples_leaf=2, min_samples_split=5, splitter='best'
    - CV Accuracy: 0.8625 (highest among models)
    - Test Accuracy: 0.6667 (lower, indicating slight overfitting)
  - **K-Nearest Neighbors (KNN):**
    - Best hyperparameters: n_neighbors=10, algorithm='auto', p=1
    - CV Accuracy: 0.8482
    - Test Accuracy: 0.8333
- **Confusion Matrices**
  - Show model performance in terms of true positives, false positives, true negatives, and false negatives.
  - Logistic Regression, SVM, and KNN show consistent test performance.
  - Decision Tree overfits slightly despite highest CV score.

Results

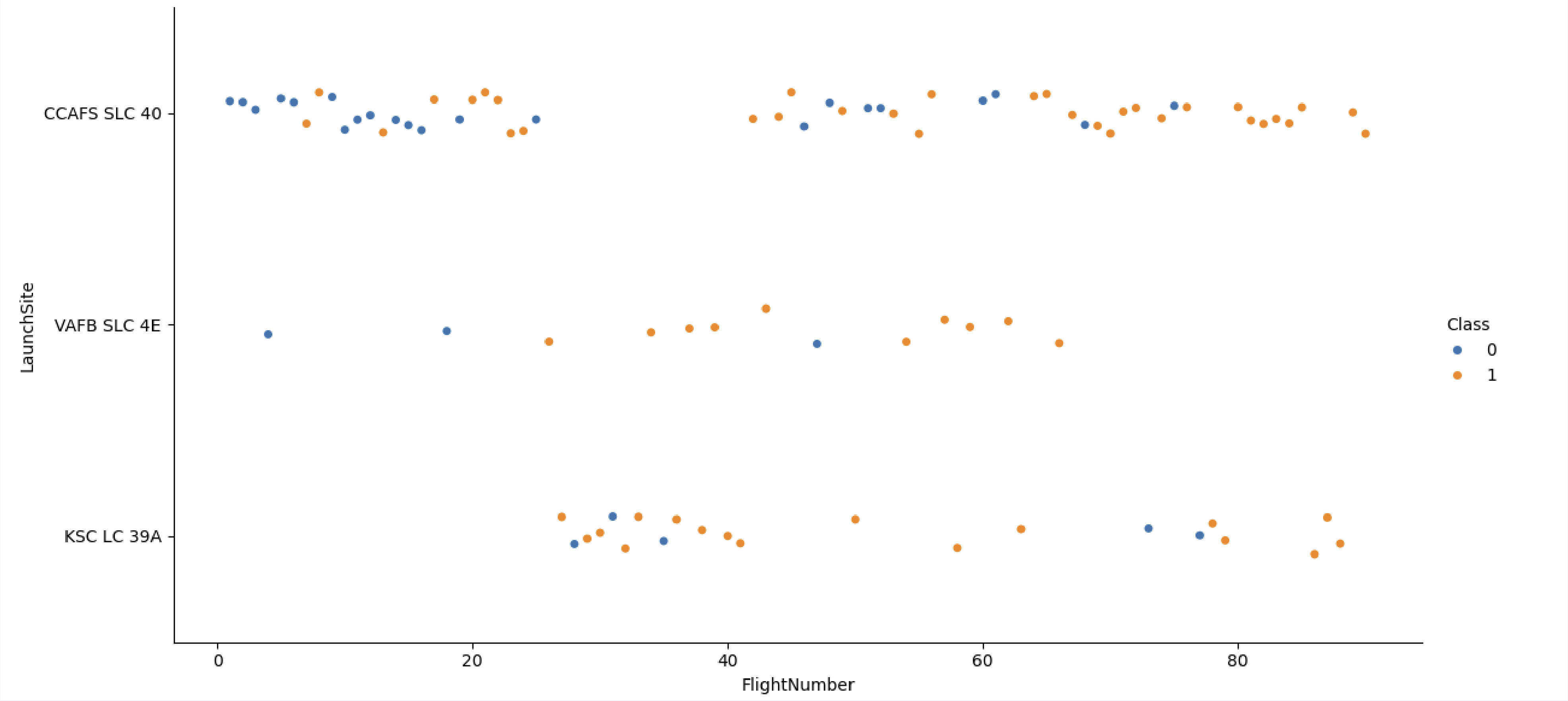
Interactive analytics demo



Section 2

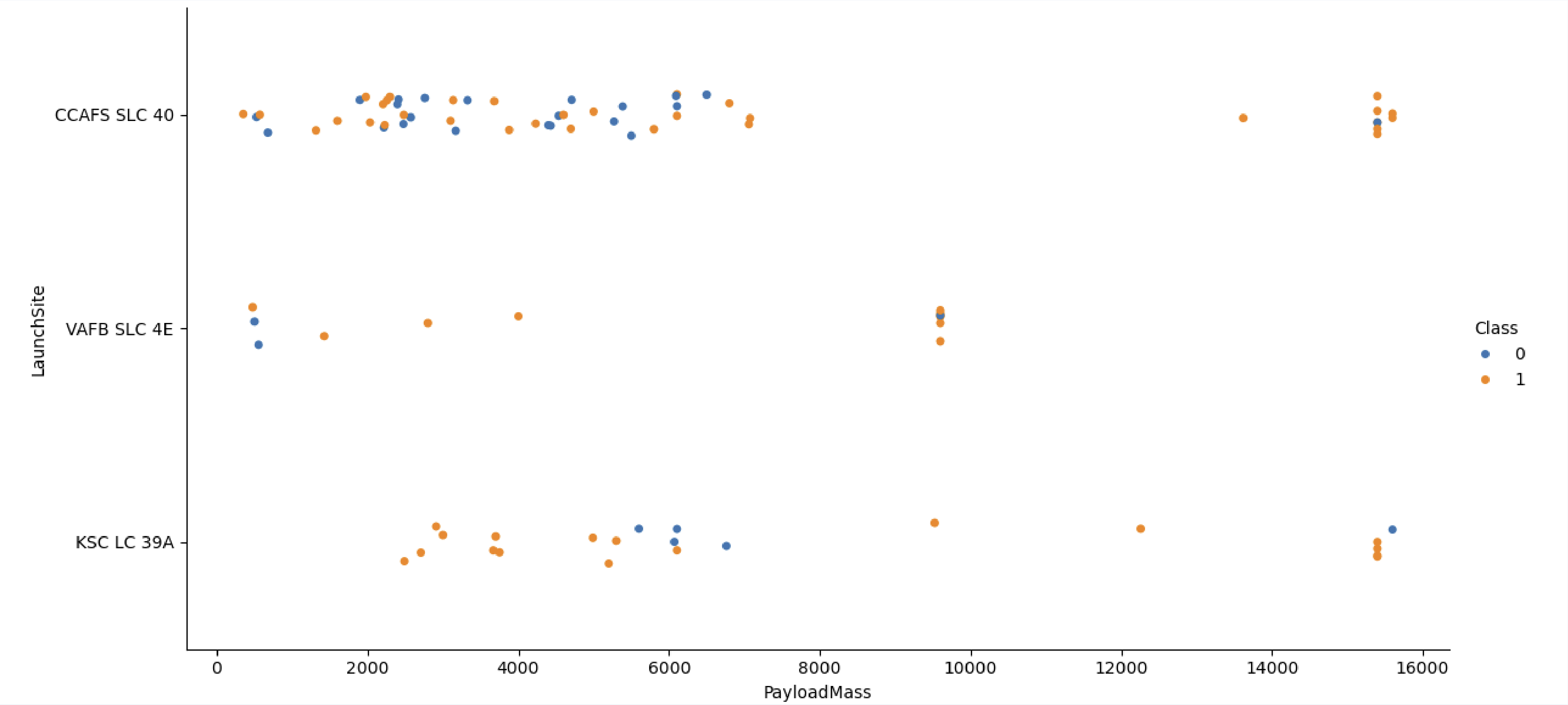
Insights drawn from EDA

Flight Number vs. Launch Site



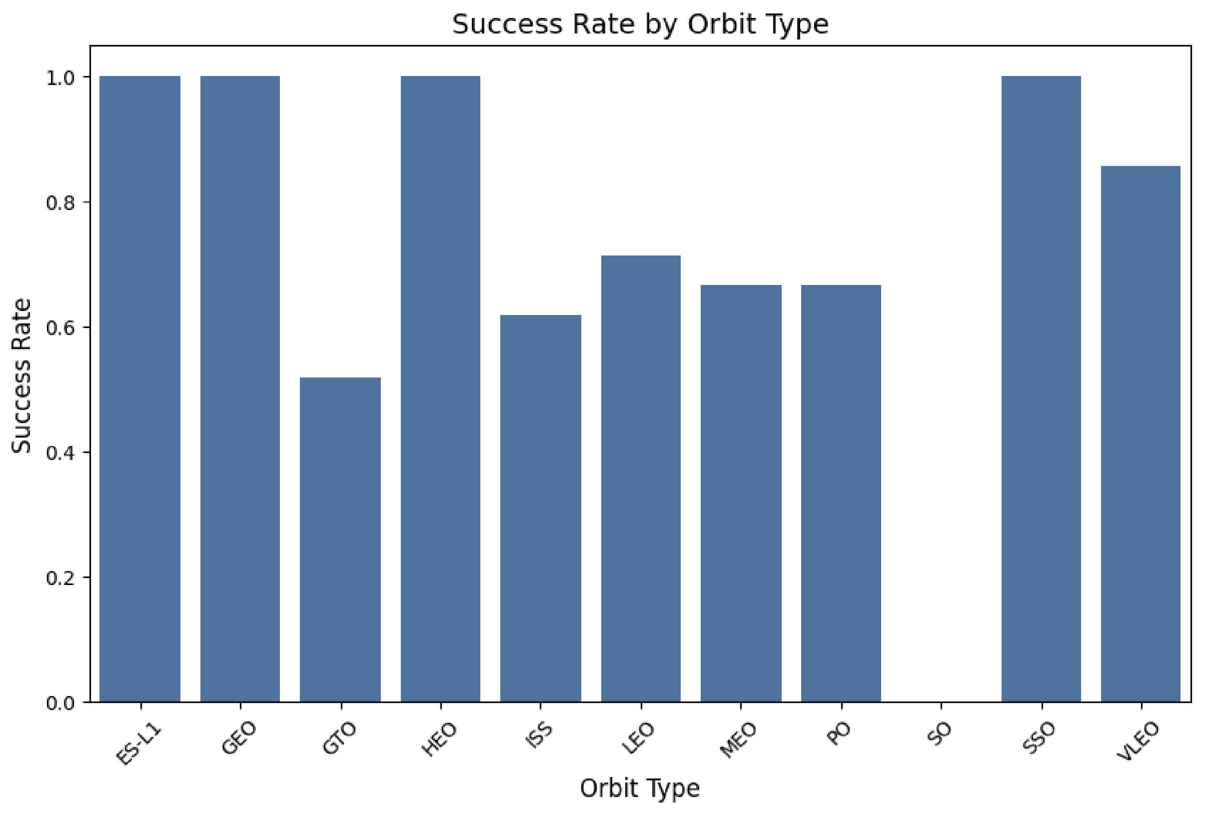
For all launch sites, there are improvements with the flight number

Payload vs. Launch Site



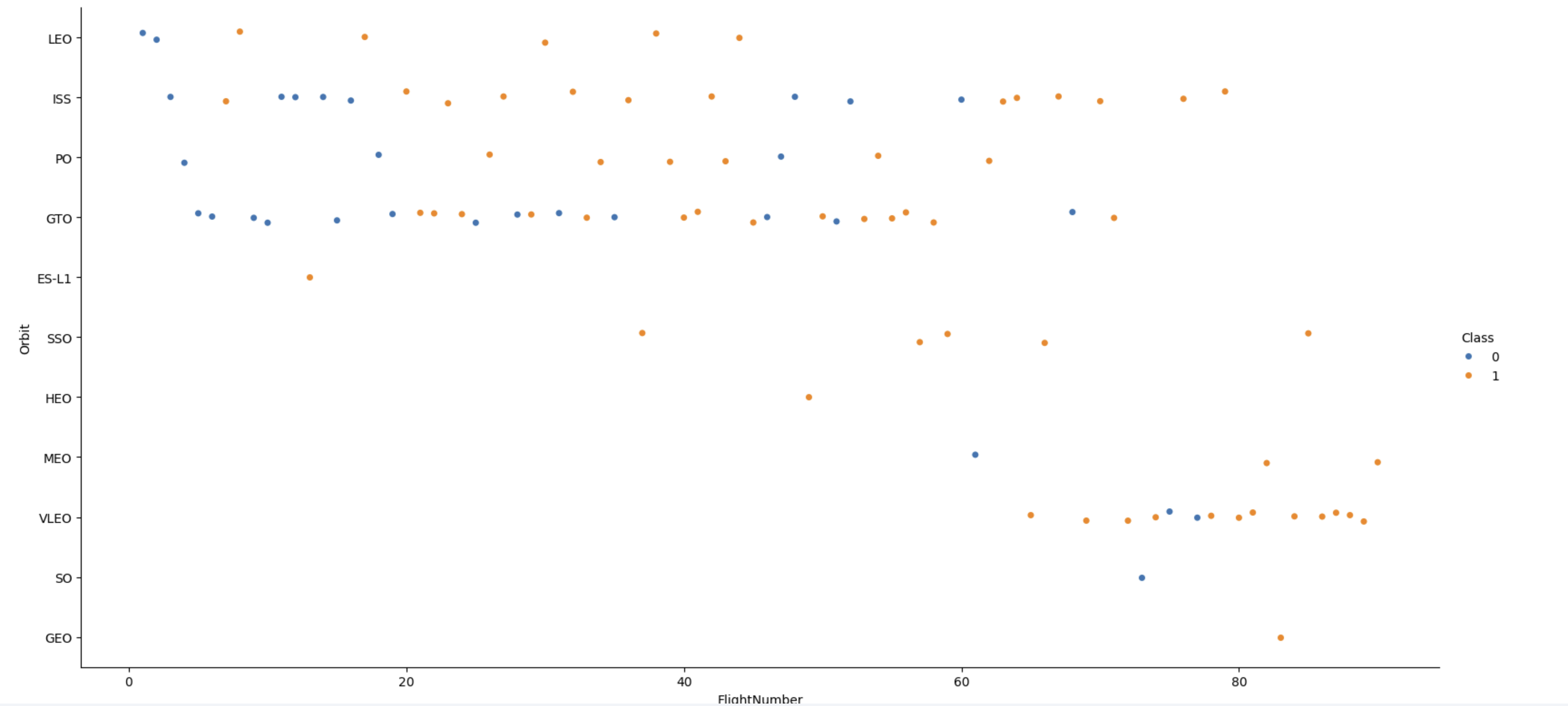
For the VAFB SLC-4E launch site, there are no rockets launched with a heavy payload mass exceeding 10,000 kg. This indicates that this site has been primarily used for lighter payload missions.

Success Rate vs. Orbit Type



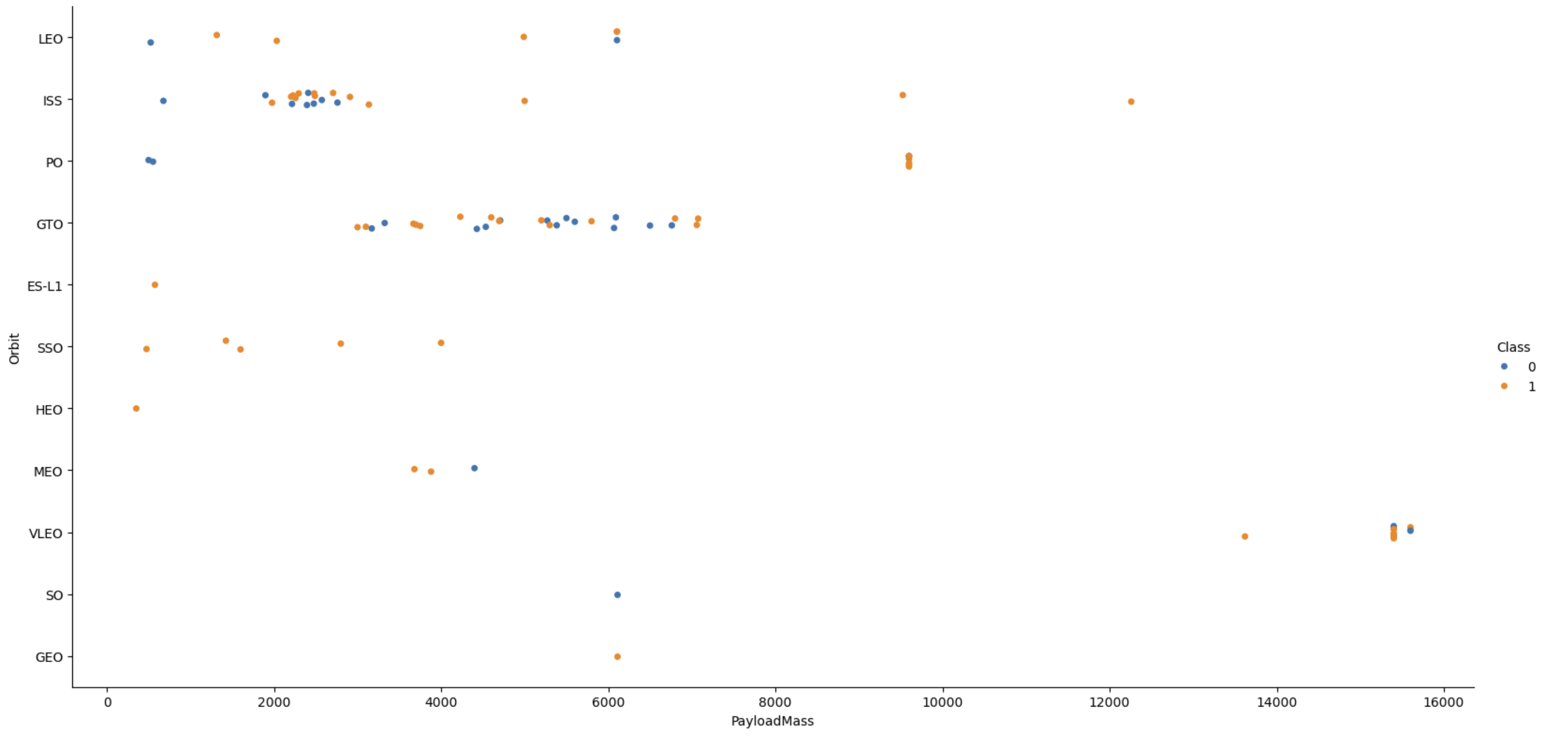
Orbits ES-L1, GEO, HEO, and SSO show a perfect success rate of 100%, whereas the SO orbit has a success rate of 0%.

Flight Number vs. Orbit Type



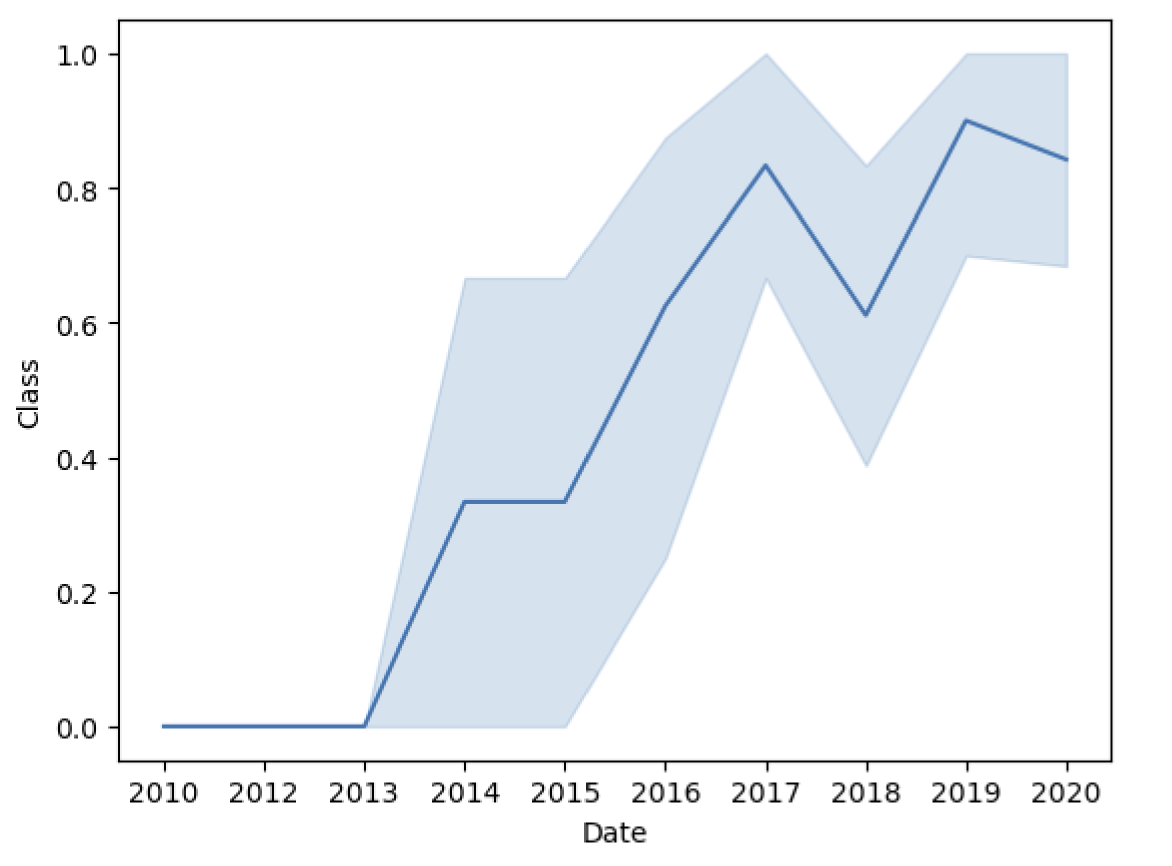
In the LEO orbit, launch success appears to correlate with the number of flights. Conversely, for the GTO orbit, there seems to be no observable relationship between flight number and success.

Payload vs. Orbit Type



For heavy payloads, the rate of successful landings is higher for Polar, LEO, and ISS orbits. In contrast, for GTO orbits, distinguishing between successful and unsuccessful landings is challenging, as both outcomes occur.

Launch Success Yearly Trend



The launch success rate has steadily increased from 2013 through 2020.

All Launch Site Names

- Find the names of the unique launch sites
- SELECT DISTINCT Launch_Site FROM SPACEXTBL

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5

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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer is "NASA (CRS)"

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE "F9 v1.1"

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome is "Success (ground pad)"

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome is "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTBL GROUP BY Mission_Outcome;

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IS (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date, 1, 4) = '2015' AND Landing_Outcome LIKE '%Failure%' AND Landing_Outcome LIKE '%drone ship%'

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

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY OutcomeCount DESC

Landing_Outcome	OutcomeCount
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

Section 3

Launch Sites Proximities Analysis

Launch sites located

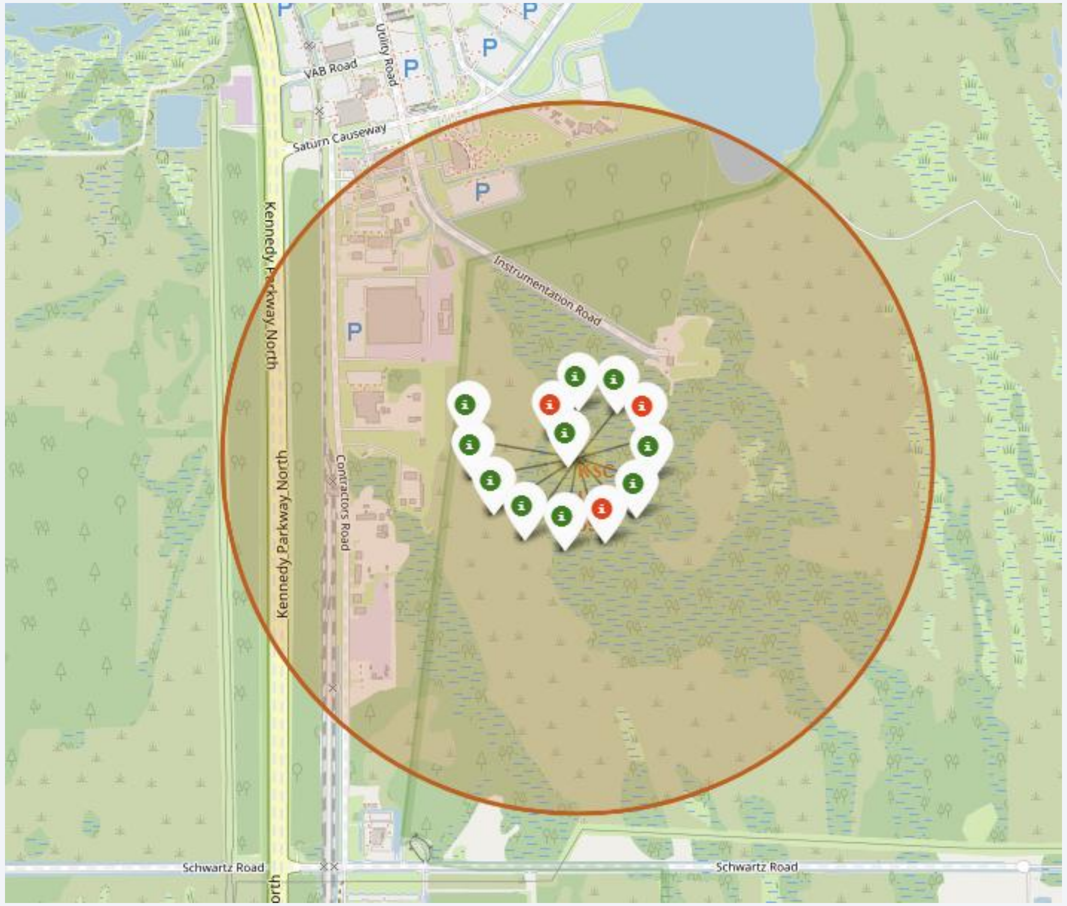
- All launch sites are located in proximity to the Equator because provides rotational advantages for certain orbits
- All launch sites are proximity to the coast to ensure rockets can be launched over open water

Success/failed launches for each site

- Color-coded markers in the clusters clearly show which launch sites have higher success rates.

You can also make it slightly more visual:

- **Green markers:** Successful launches
- **Red markers:** Failed launches
- **Insight:** Sites with more green markers indicate consistently higher success rates.



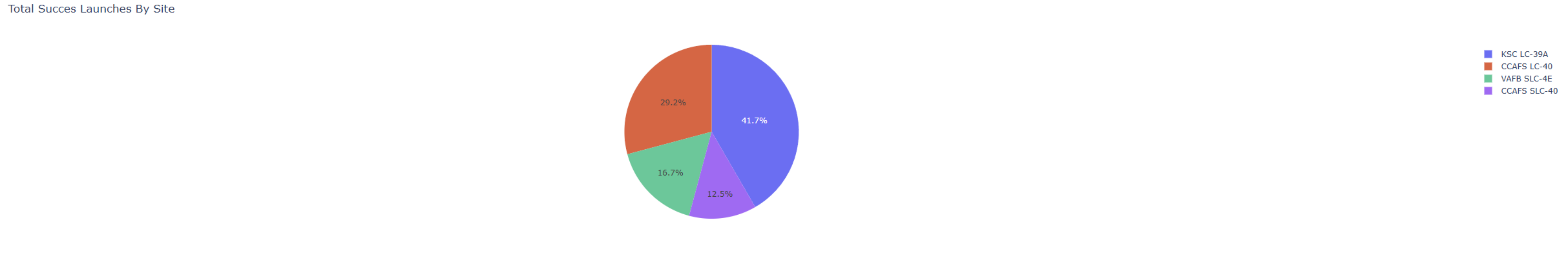
Distances between a launch site to its proximities

- **Railways:** Some launch sites are located near railway lines for logistical support.
- **Highways:** Launch sites are generally close to highways for easy transport access.
- **Coastline:** Most launch sites are near the coast to facilitate over-water flight safety.
- **Cities:** Launch sites maintain a safe distance from major cities to minimize risk in case of launch failures.

Section 4

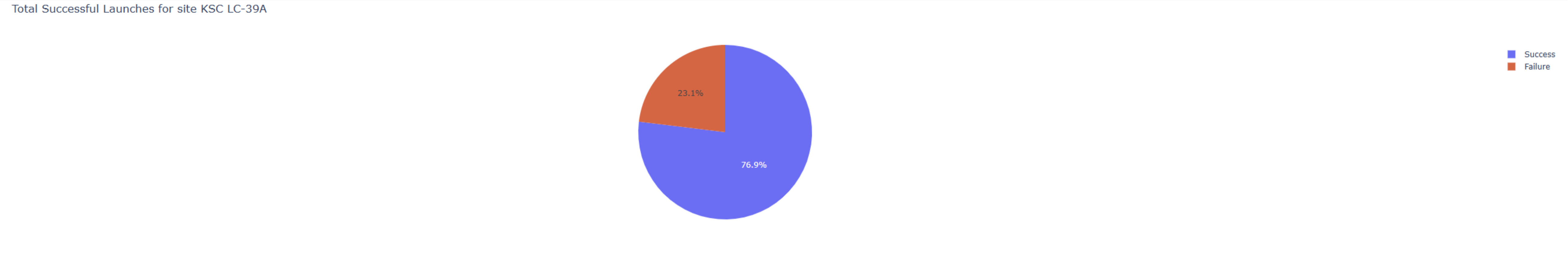
Build a Dashboard with Plotly Dash

Total success launches by site



- KSC LC-39A has the largest successful launches
- CCAFS SLC-40 has the lowest number of successful launches

Total successful launches for KSC LC-39A



- 76,9% Success
- 23,1% Failure

- KSC LC-39A has largest successful launches and the best ratio

Payload vs Outcome for All sites

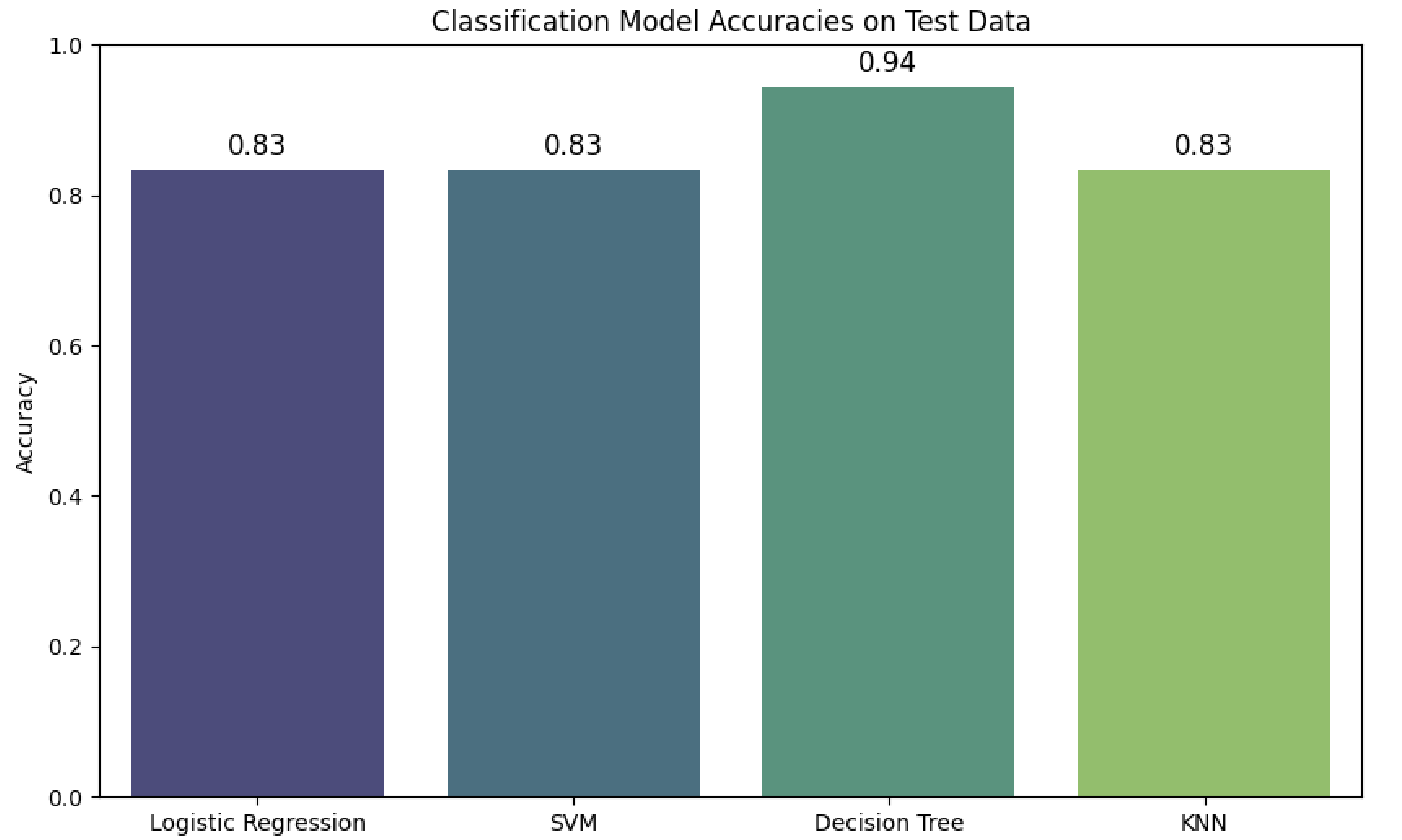
- Boosters B4 and FT: Only ones with payload mass above 5,000 kg.
- FT Booster: Achieves a higher success rate compared to B4.

Section 5

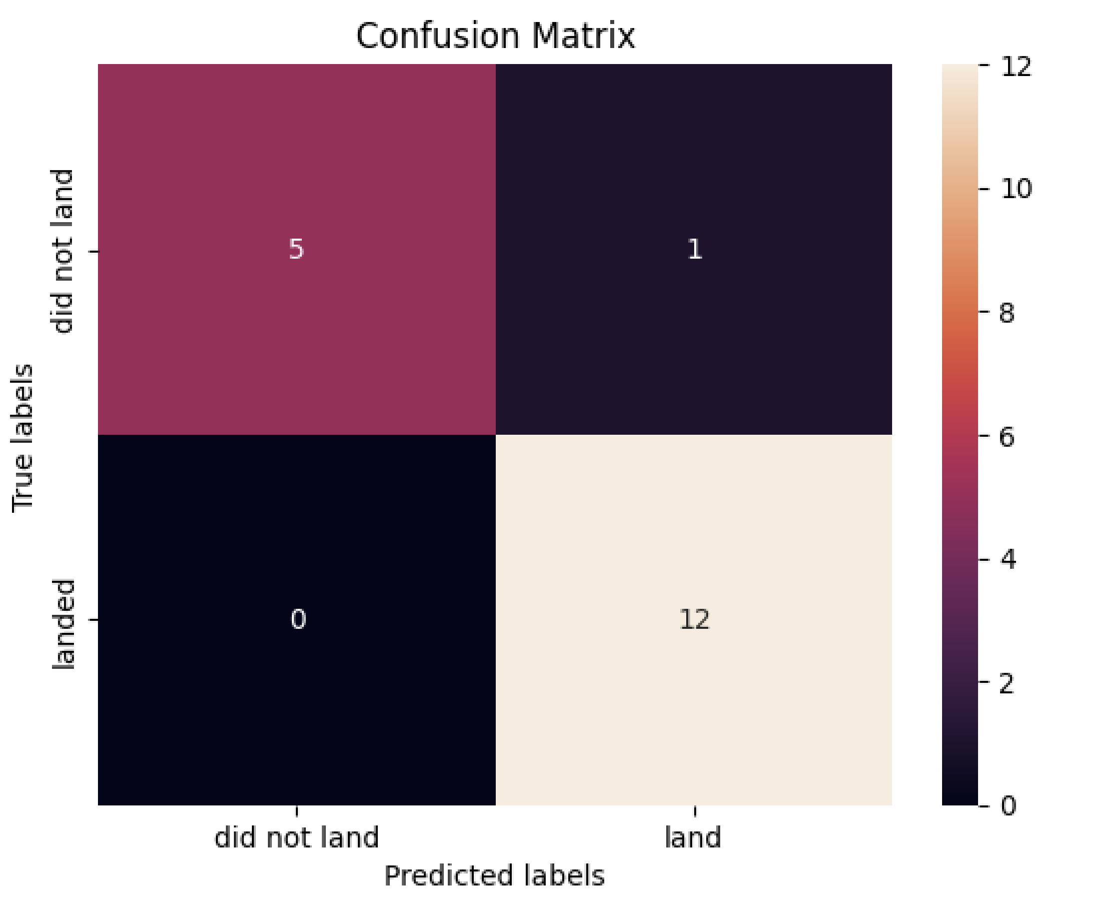
Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree has the best classification accuracy on Test Data



Confusion Matrix



- The classification model demonstrates high recall, effectively identifying most positive cases.
- On the test dataset, there was only one misclassification, which was a false positive.
- This indicates the model is reliable for predicting successful landings with minimal errors.

Conclusions

- **Launch Site Insights:** KSC LC-39A recorded the highest number of successful launches, while CCAFS SLC-40 had the lowest. Launch sites are generally close to the coast and maintain safe distances from cities, highways, and railways.
- **Payload and Booster Analysis:** Heavier payloads (above 5000 kg) were mostly handled by boosters B4 and FT, with FT showing a higher success rate. LEO, Polar, and ISS missions exhibit higher successful landing rates for heavy payloads.
- **Orbit and Success Trends:** Success rates vary by orbit type; orbits such as ES-L1, GEO, HEO, and SSO had 100% success, while some orbits like SO showed no successful landings. LEO success tends to increase with the number of flights, unlike GTO.
- **Temporal Trends:** The overall launch success rate has steadily increased from 2013 to 2020, reflecting SpaceX's operational and technological improvements.
- **Predictive Model Performance:** Classification model (Decision Tree) achieved high accuracy, showing minimal false positives and strong recall, indicating reliability in predicting successful landings.

Appendix

- Repository with all the code and notebooks: https://github.com/JoseMarrero-Dom/DataScience_Capstone.git

Thank you!