IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Umera
18/6/2025

Outline

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

Executive Summary

- Brief overview of project goals: Predict Falcon 9 first stage landing success.
- Summary of methodologies: Data collection, wrangling, EDA, visualization, predictive modeling.
- Summary of results: Key insights from EDA, geospatial analysis, interactive dashboards, model performance.

Introduction

- SpaceX reduces launch costs by reusing Falcon 9 first stage.
- Launch cost comparison: $62M vs. $165M+ for competitors.
- Importance of predicting landing success for cost and competitive bidding.

- Problem: Can we predict if Falcon 9 first stage will land successfully?
- Objectives: Analyze launch data, identify key factors, build classification models.
- Questions: Impact of payload, launch site, orbit, flight number on landing success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis (EDA)
  - Interactive Visual Analytics
  - Predictive Analysis (Classification)

Data Collection

- Data sources: SpaceX REST API and Wikipedia web scraping.
- Tools: Python requests, BeautifulSoup, Pandas.
- Flowchart illustrating data retrieval and dataset creation.
- GitHub URL for API calls notebook.

Data Collection – SpaceX API

- Steps: GET requests, JSON to DataFrame conversion, filtering Falcon 9 launches, handling missing values.
- Key variables collected: flight number, payload mass, launch site, orbit, landing outcome.
- Flowchart of API data extraction process

Data Collection - Scraping

- Scraped Falcon 9 launch records from Wikipedia.
- Steps: Request HTML, parse tables with BeautifulSoup, convert to DataFrame.
- Flowchart of scraping and parsing process.
- GitHub URL for scraping notebook.

Place your flowchart of web scraping here

Data Wrangling

- Cleaning: Handling missing values with fillna(), removing duplicates.
- Feature engineering: Creating binary landing outcome label (0 = fail, 1 = success).
- Encoding categorical variables for modeling.

EDA with Data Visualization

- Tools: Pandas, Matplotlib, Seaborn, SQL queries.
- Goals: Understand variable distributions, relationships, and patterns.
- Scatter plot shows launch frequency across sites over time.
- Insight: Some sites have more launches, indicating operational preference.
- Scatter plot of payload mass by orbit type.
- Insight: Payload mass varies significantly with orbit destination.
- Line chart showing average landing success rate per year.
- Insight: Success rate improves over time, indicating technological progress.

EDA with SQL

- Query: List all unique launch sites.
- Result: Names of sites (e.g., CCAFS SLC-40, VAFB SLC-4E).
- Query: Find launch sites starting with 'CCA'.
- Result: Five records matching criteria.
- Query: Sum of payload mass for NASA missions.
- Result: Total payload mass value.
- Query: Average payload for booster version F9 v1.1.
- Result: Average value.
- Query: Date of first successful ground pad landing.
- Result: Specific date.
- Query: Boosters with successful drone ship landings and payload between 4000-6000 kg.
- Result: List of booster names.
- Query: Count of successful vs. failed missions.
- Result: Numbers for each category.
- Query: Boosters that carried maximum payload mass.
- Result: Booster names and payload values.
- Query: Failed drone ship landings in 2015 with booster and site info.
- Result: List of records.
- Query: Count and rank of landing outcomes in date range.
- Result: Ranked list of outcomes.

Build an Interactive Map with Folium

- Map showing all launch site locations globally.
- Map with color-coded markers for successful and failed landings.
- Map showing distances from launch site to nearby infrastructure: railway, highway, coastline.

Build a Dashboard with Plotly Dash

- Pie chart showing success counts for all launch sites.
- Pie chart focusing on site with highest success ratio.
- Scatter plot with interactive payload range slider.

Predictive Analysis (Classification)

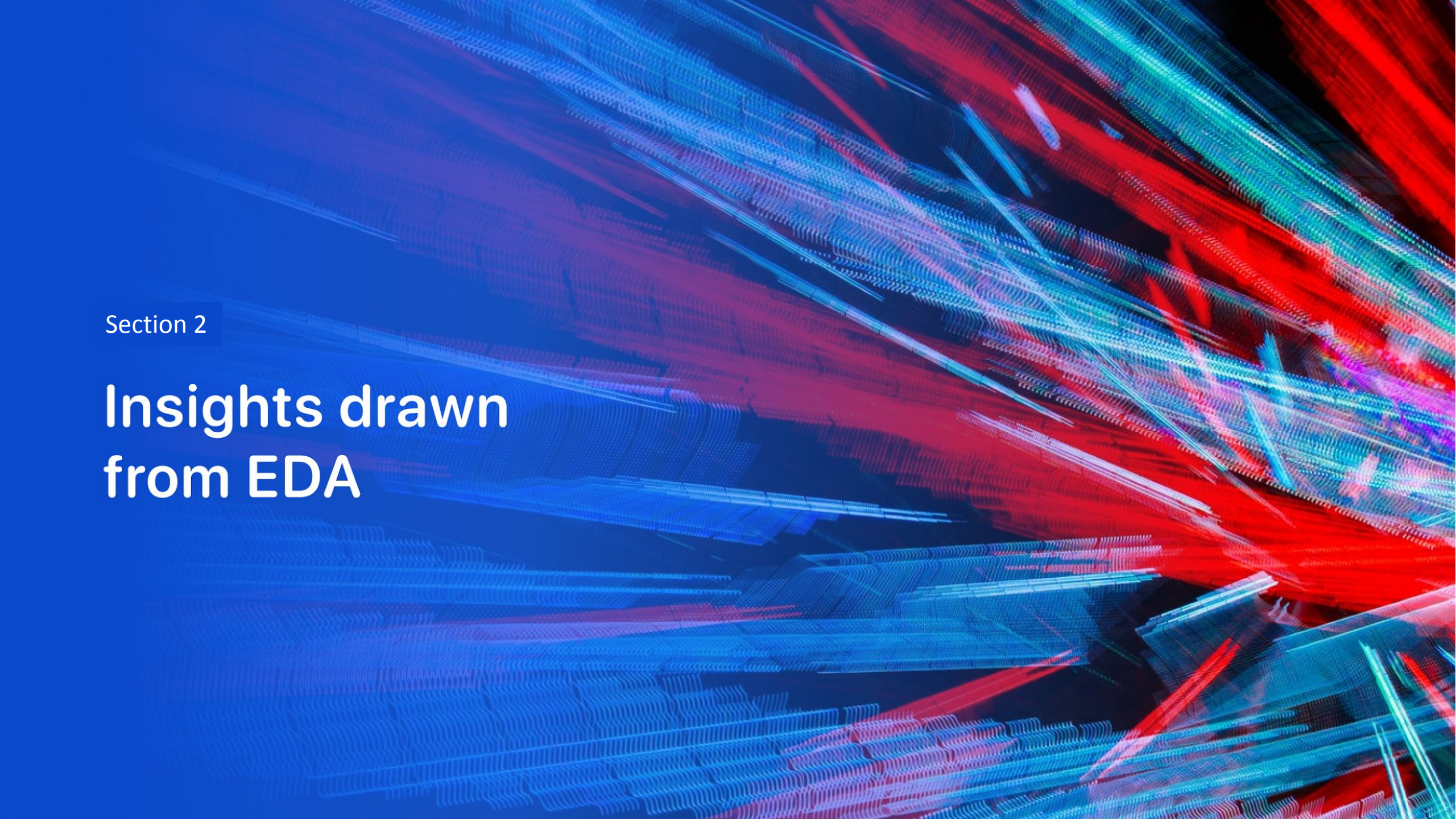
- Models used: Logistic Regression, Decision Trees, SVM, KNN.
- Data preprocessing: Standardization, train-test split.
- Hyperparameter tuning with GridSearchCV.
- Bar chart comparing accuracy of all models.
- Highlight best performing model.
- Confusion matrix visualization.
- Explanation of true positives, false positives, etc.
- Interpretation of model performance.
- Key features influencing landing success (e.g., payload, orbit).
- Model strengths and limitations.
- Potential for improvement with more data.
- Recap: Data collection and wrangling enabled robust analysis.
- EDA revealed important patterns influencing landing success.
- Interactive maps and dashboards enhanced data exploration.
- Predictive models achieved good accuracy for classification.

Results

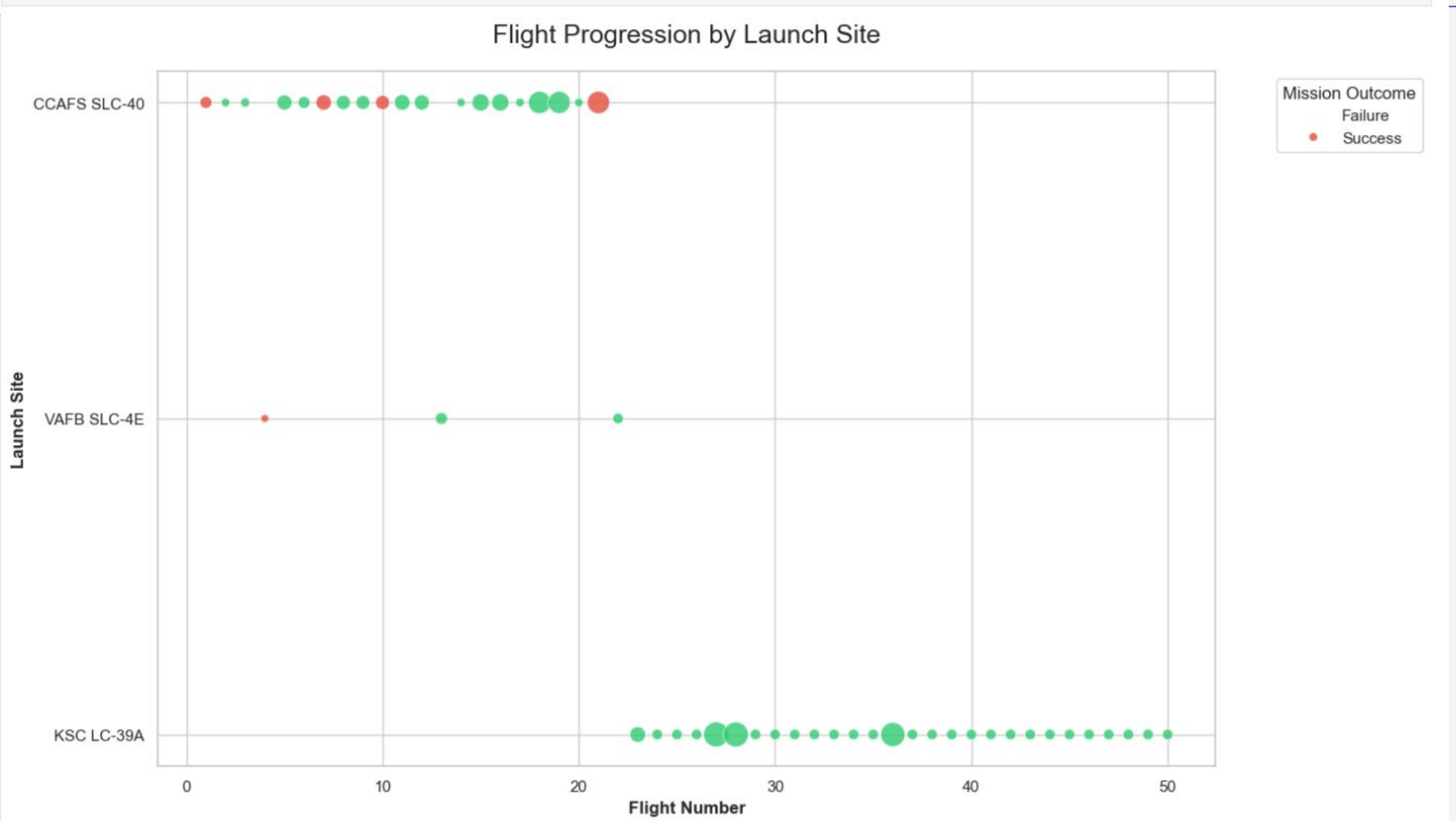
- Predicting landing success aids cost estimation and competitive bidding.
- Insights can guide launch planning and risk management.

Section 2

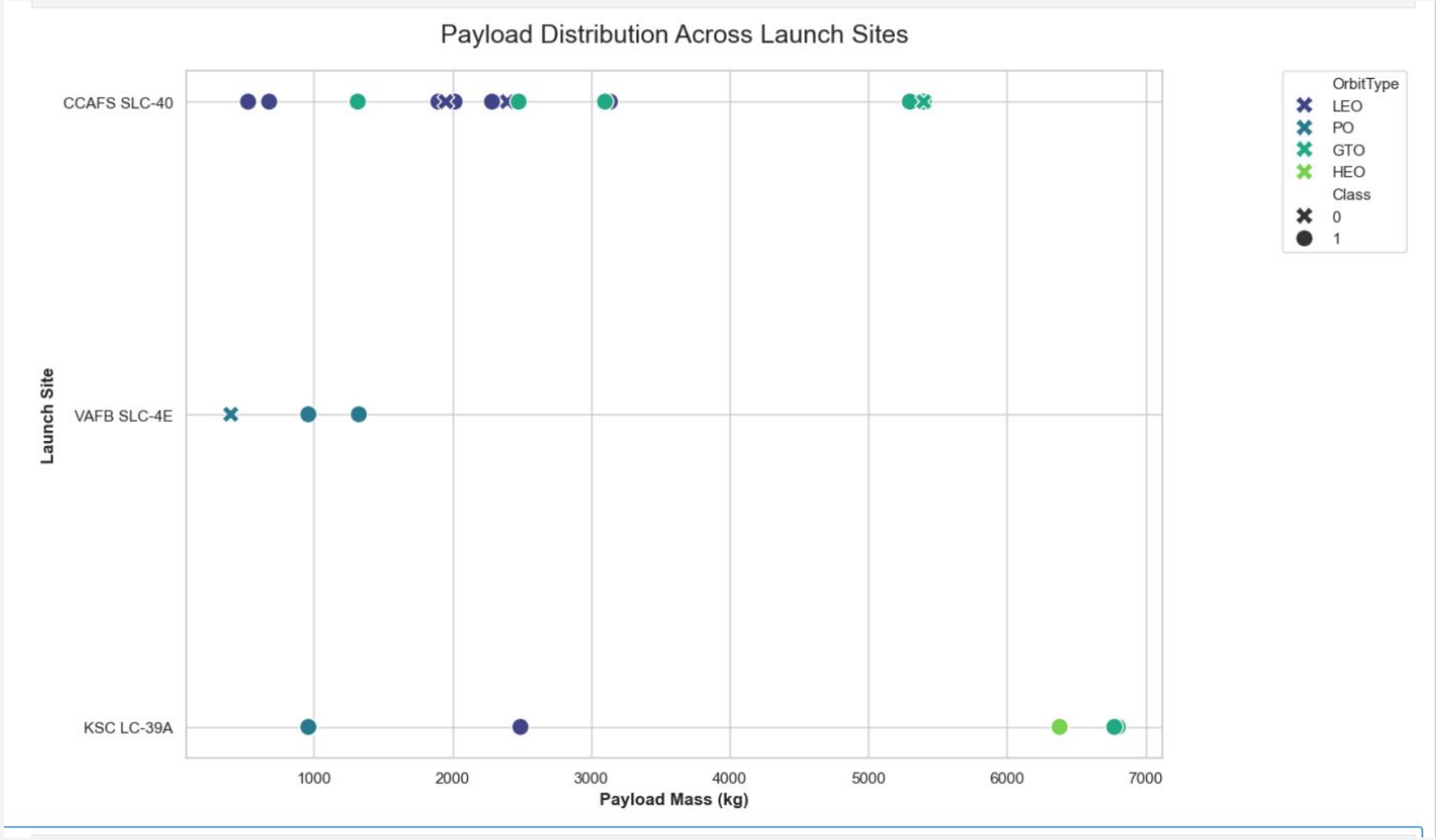
Insights drawn from EDA



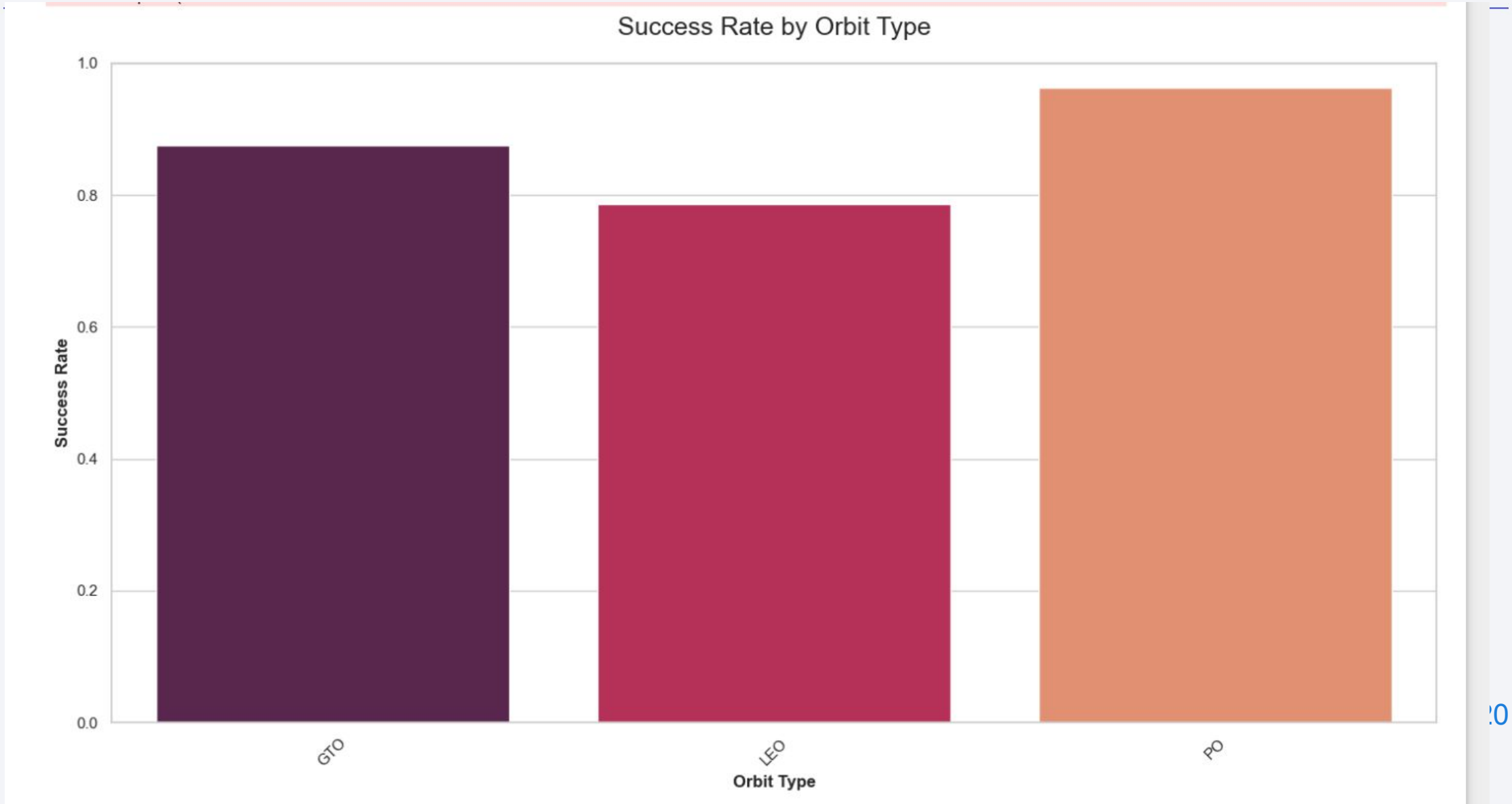
Flight Number vs. Launch Site



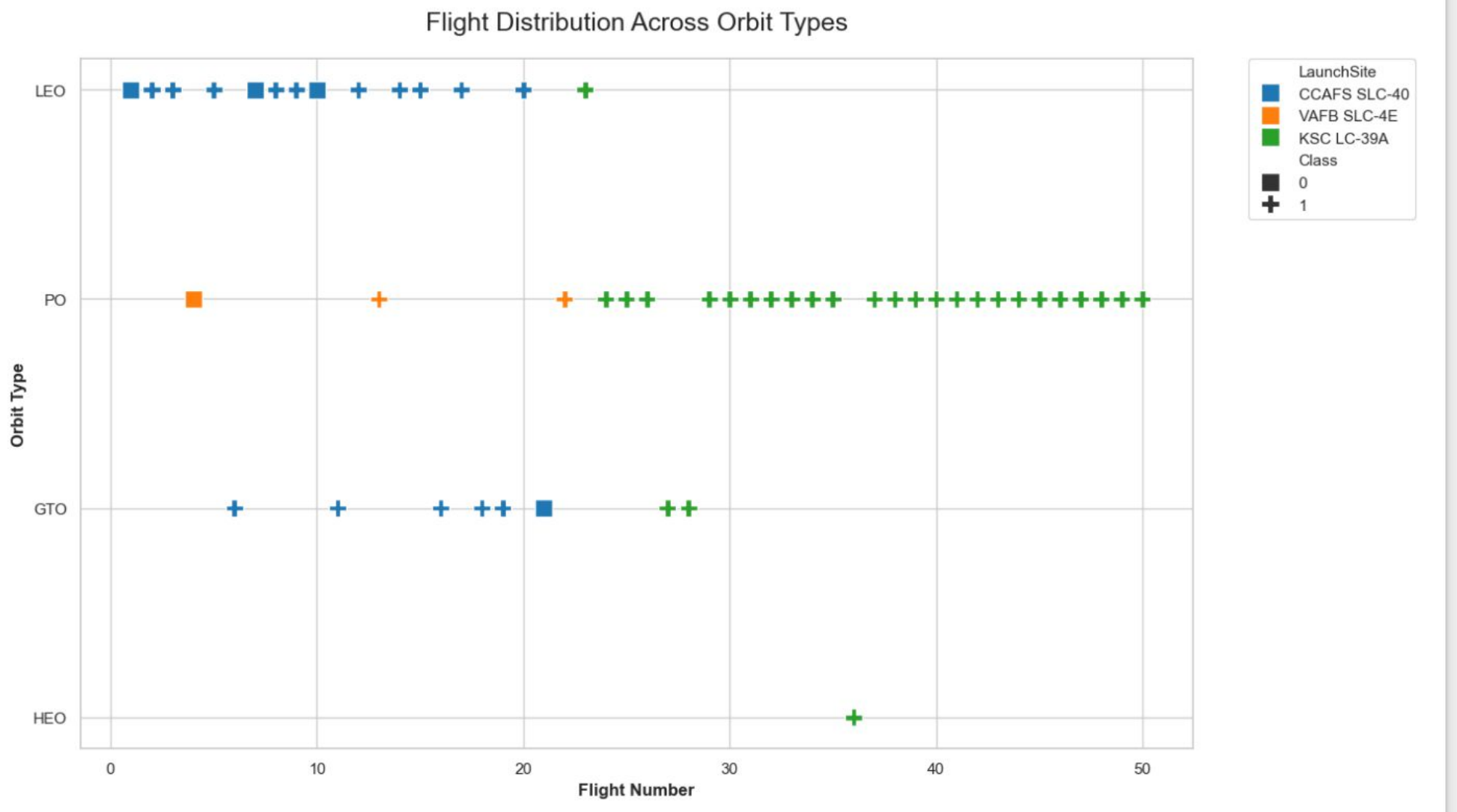
Payload vs. Launch Site



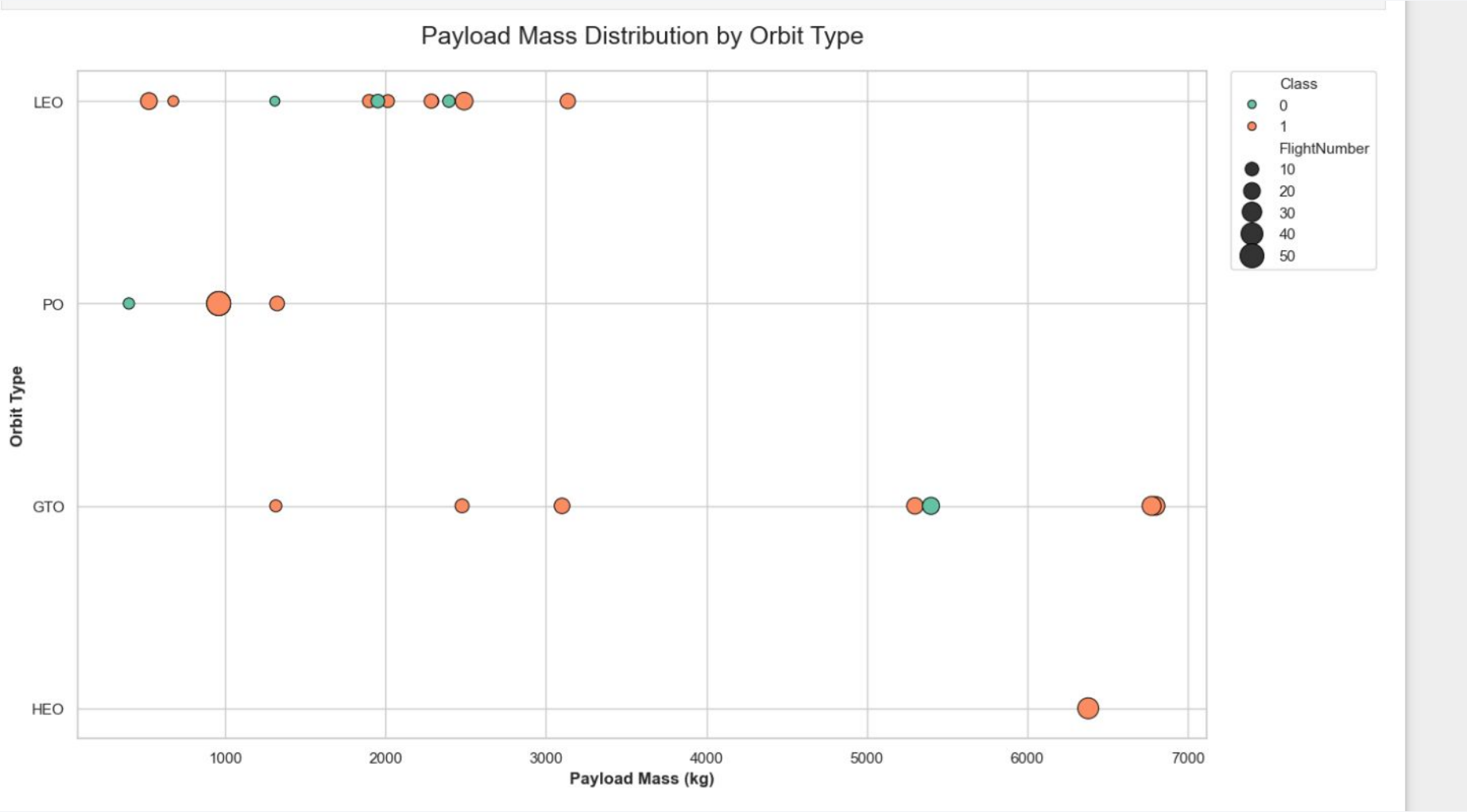
Success Rate vs. Orbit Type



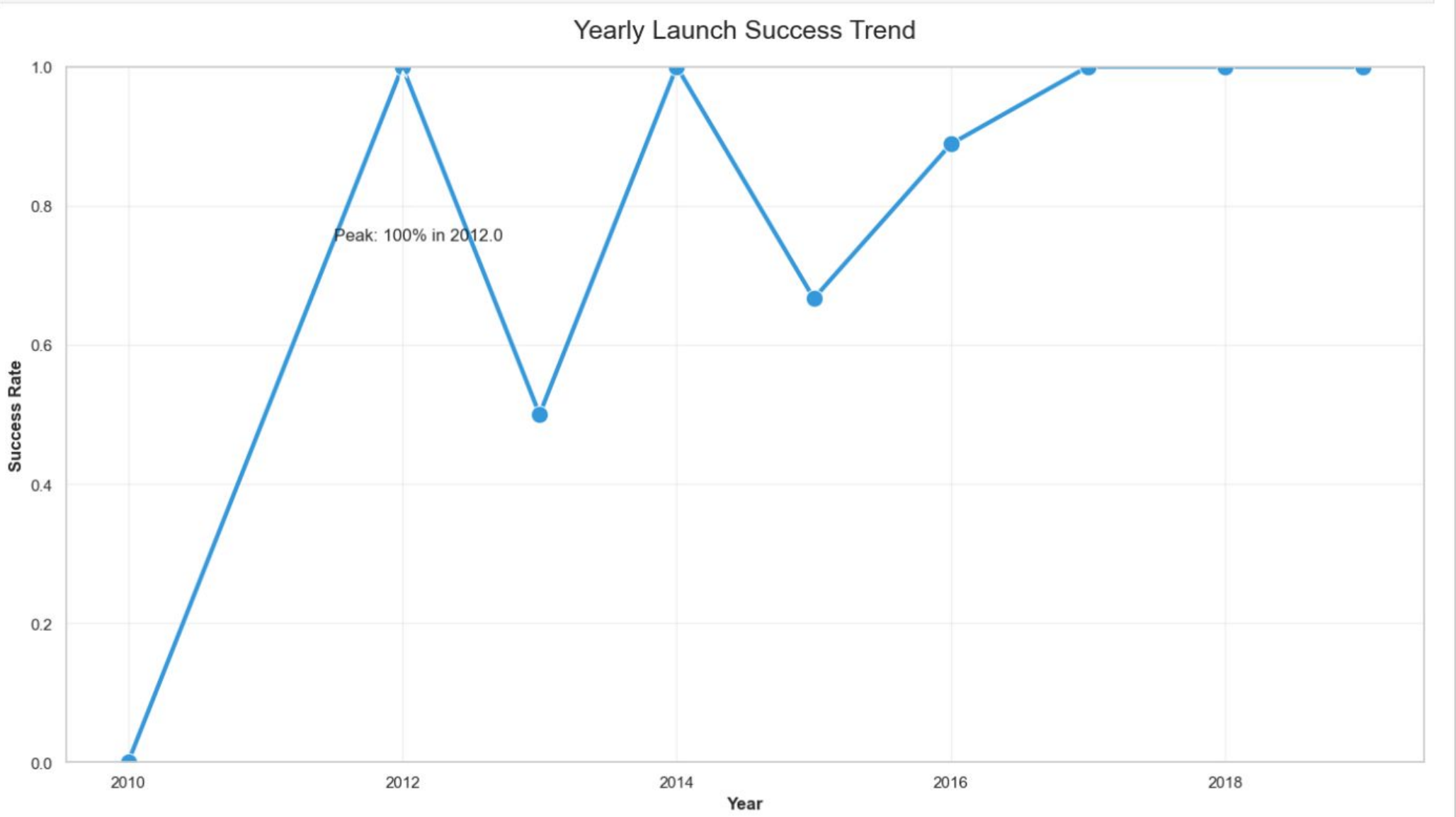
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

1. All Launch Site Names

Query:

```sql
SELECT DISTINCT LaunchSite FROM SpaceX;
```

Sample Result:

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

Launch Site Names Begin with 'CCA'

2. Launch Site Names Begin with `CCA`

Query:

```sql
SELECT LaunchSite FROM SpaceX WHERE LaunchSite LIKE 'CCA%';
```

Sample Result:

LaunchSite
CCAFS SLC-40

Explanation:

This query finds all launch site names that begin with 'CCA', which refers to Cape

Total Payload Mass

3. Total Payload Mass by NASA Boosters

Query:

```sql
SELECT SUM(PayloadMass) AS TotalPayload FROM SpaceX WHERE Customer = 'NASA (CRS)';
```

Sample Result:

TotalPayload
45596

Explanation:

This query calculates the total payload mass carried by boosters for NASA (CRS) missions [6] [7].

Average Payload Mass by F9 v1.1

4. Average Payload Mass by F9 v1.1

Query:

```sql
SELECT AVG(PayloadMass) AS AvgPayload FROM SpaceX WHERE BoosterVersion
LIKE 'F9 v1.1%';
```

Sample Result:

AvgPayload
2928

Explanation:

This query finds the average payload mass carried by the F9 v1.1 booster version [6] [7].

First Successful Ground Landing Date

5. First Successful Ground Landing Date

Query:

```sql
SELECT MIN(Date) AS FirstGroundLanding FROM SpaceX WHERE LandingOutcome
= 'Success (ground pad)';
```

Sample Result:

FirstGroundLanding
2015-12-22

Explanation:

This query returns the date of the first successful Falcon 9 ground pad landing [6] [7].

Successful Drone Ship Landing with Payload between 4000 and 6000

6. Successful Drone Ship Landing with Payload between 4000 and 6000

Query:

```sql
SELECT DISTINCT BoosterVersion FROM SpaceX
WHERE LandingOutcome = 'Success (drone ship)'
AND PayloadMass > 4000 AND PayloadMass < 6000;
```

Sample Result:

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

Explanation:
This query lists boosters that successfully landed on a drone ship with payloads between 4000 and 6000 kg [6] [7].

Total Number of Successful and Failure Mission Outcomes

7. Total Number of Successful and Failure Mission Outcomes

Query:

```sql
SELECT
  (SELECT COUNT(*) FROM SpaceX WHERE LOWER(LandingOutcome) LIKE '%success%') AS Success,
  (SELECT COUNT(*) FROM SpaceX WHERE LOWER(LandingOutcome) NOT LIKE '%success%') AS Failure;
```

Sample Result:

Success	Failure
61	40

Explanation:
This query shows the total number of successful and failed mission outcomes in the dataset [6] [7].

Boosters Carried Maximum Payload

8. Boosters Carried Maximum Payload

Query:

```sql
SELECT BoosterVersion, PayloadMass FROM SpaceX
WHERE PayloadMass = (SELECT MAX(PayloadMass) FROM SpaceX);
```

Sample Result:

BoosterVersion	PayloadMass
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
...	...

Explanation:

This query lists the boosters that have carried the maximum payload mass [6] [7].

2015 Launch Records

Query:

```sql
 SELECT strftime('%m', Date) AS Month, LandingOutcome,
BoosterVersion, LaunchSite
 FROM SpaceX
 WHERE LandingOutcome = 'Failure (drone ship)' AND strftime('%Y', Date)
= '2015';
```

Sample Result:

Month	LandingOutcome	BoosterVersion	LaunchSite
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Explanation:

This query lists failed drone ship landings in 2015, with booster and site details [6] [7].

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

10. Rank Success Count Between 2010-06-04 and 2017-03-20

Query:

```sql
SELECT LandingOutcome, COUNT(*) AS OutcomeCount
FROM SpaceX
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LandingOutcome
ORDER BY OutcomeCount DESC;
```

Sample Result:

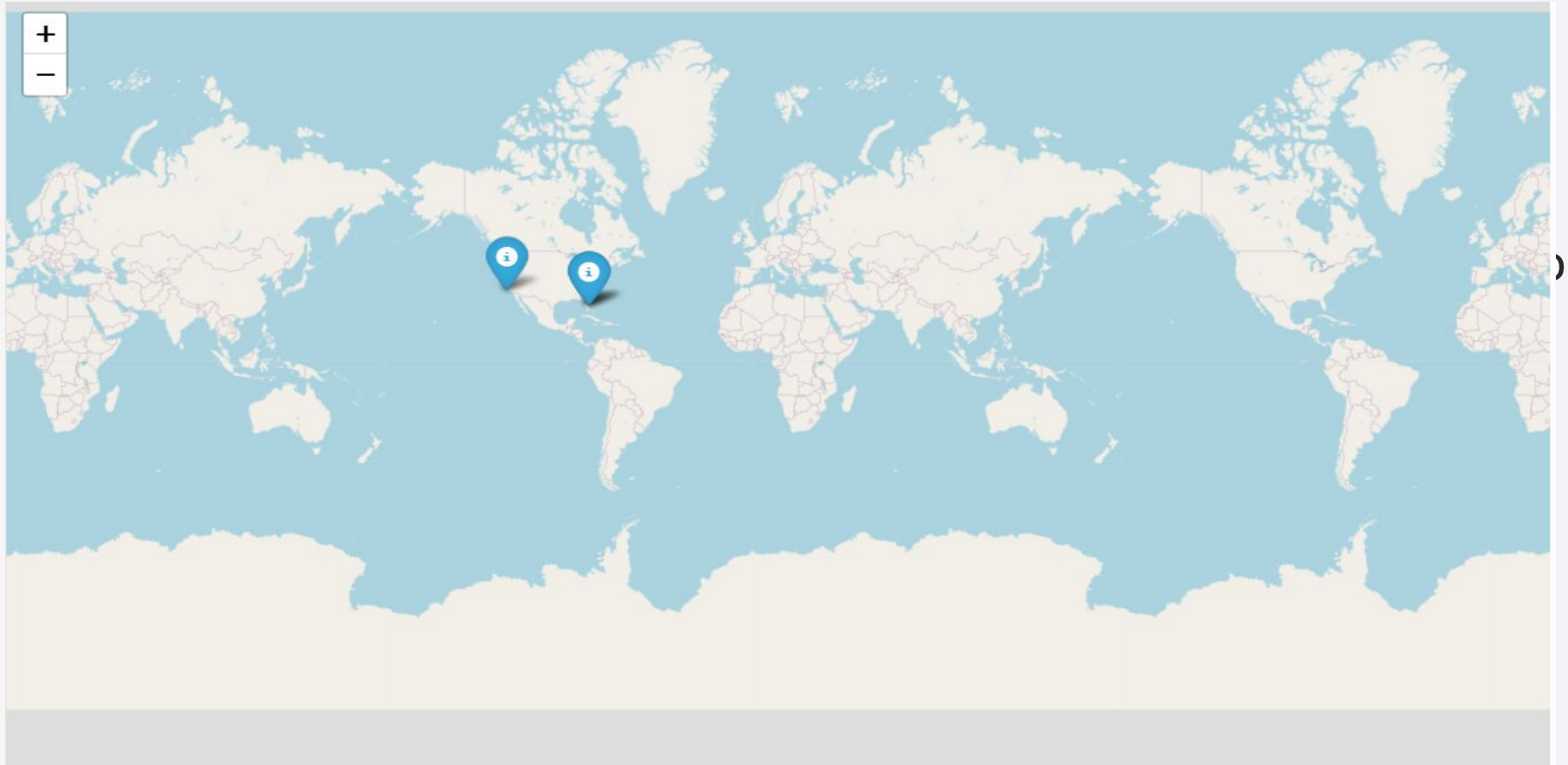
LandingOutcome	OutcomeCount
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Success (ground pad)	5

Explanation:
This query ranks the count of each landing outcome within the specified date range in descending order [6] [7].


Section 3
Launch Sites
Proximities Analysis

1. All Launch Sites’ Markers on a Global Map

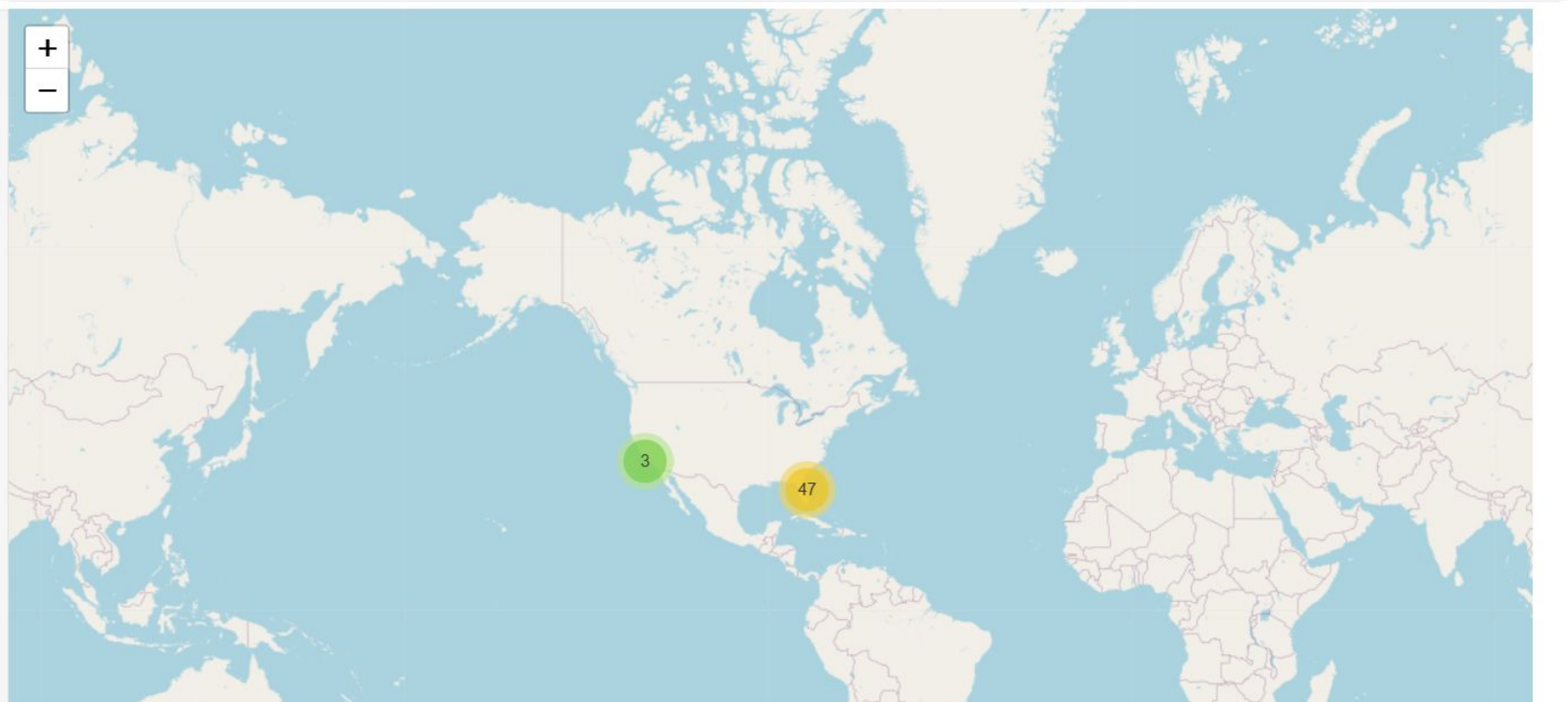
- *This map displays all SpaceX Falcon 9 launch sites as blue markers. The sites are located at CCAFS SLC-40 and KSC LC-39A in Florida, and VAFB SLC-4E in California, highlighting their strategic coastal locations for safe launch and recovery operations.*



› include

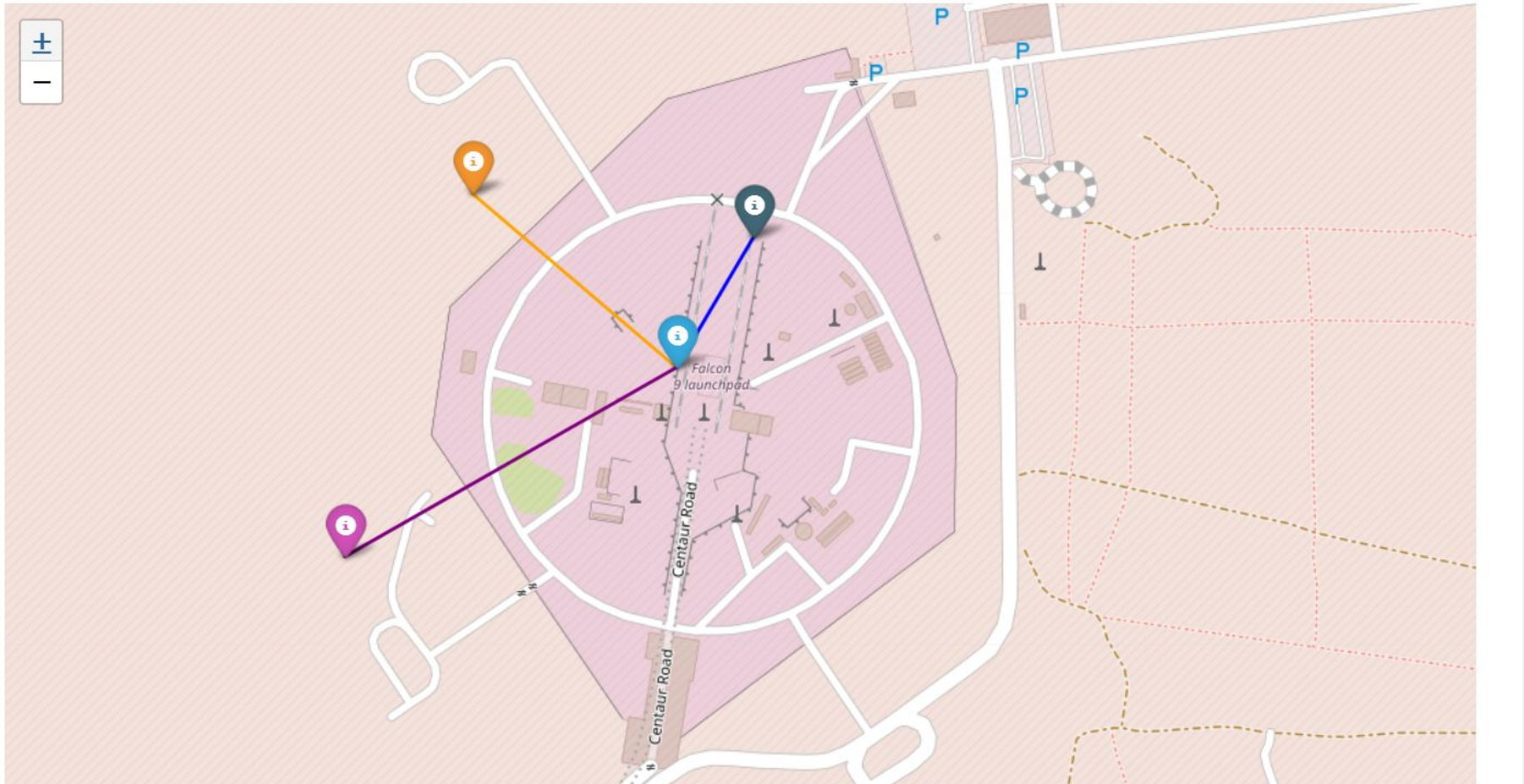
2. All Launch Records per Site on the Map

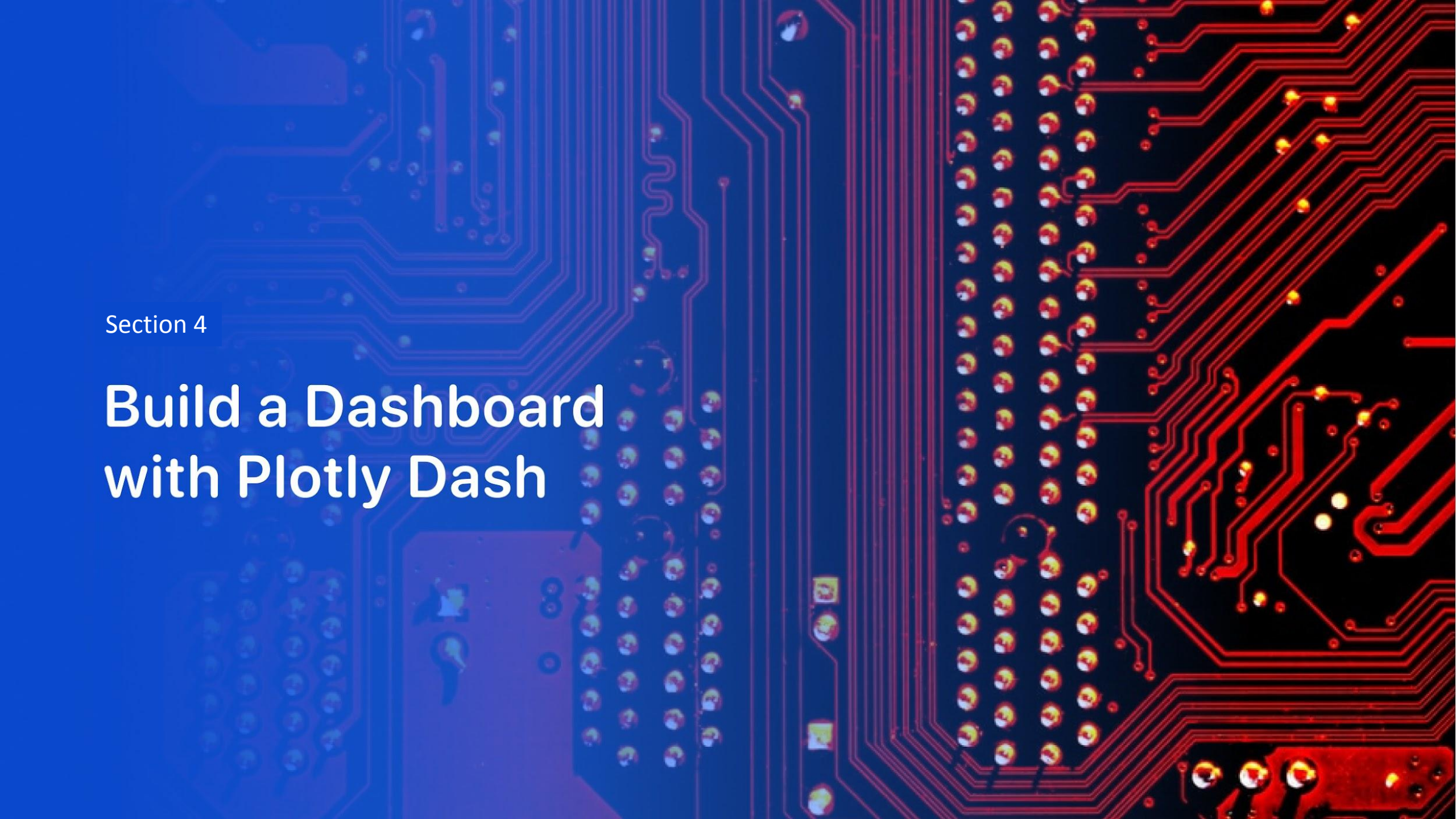
- *This map shows every Falcon 9 launch as a marker at its launch site. Green markers indicate successful missions, red markers indicate failures. Marker clustering helps visualize launch frequency and outcomes per site.*



3. Launch Sites' Proximities (Railway, Highway, Coastline, with Distance)

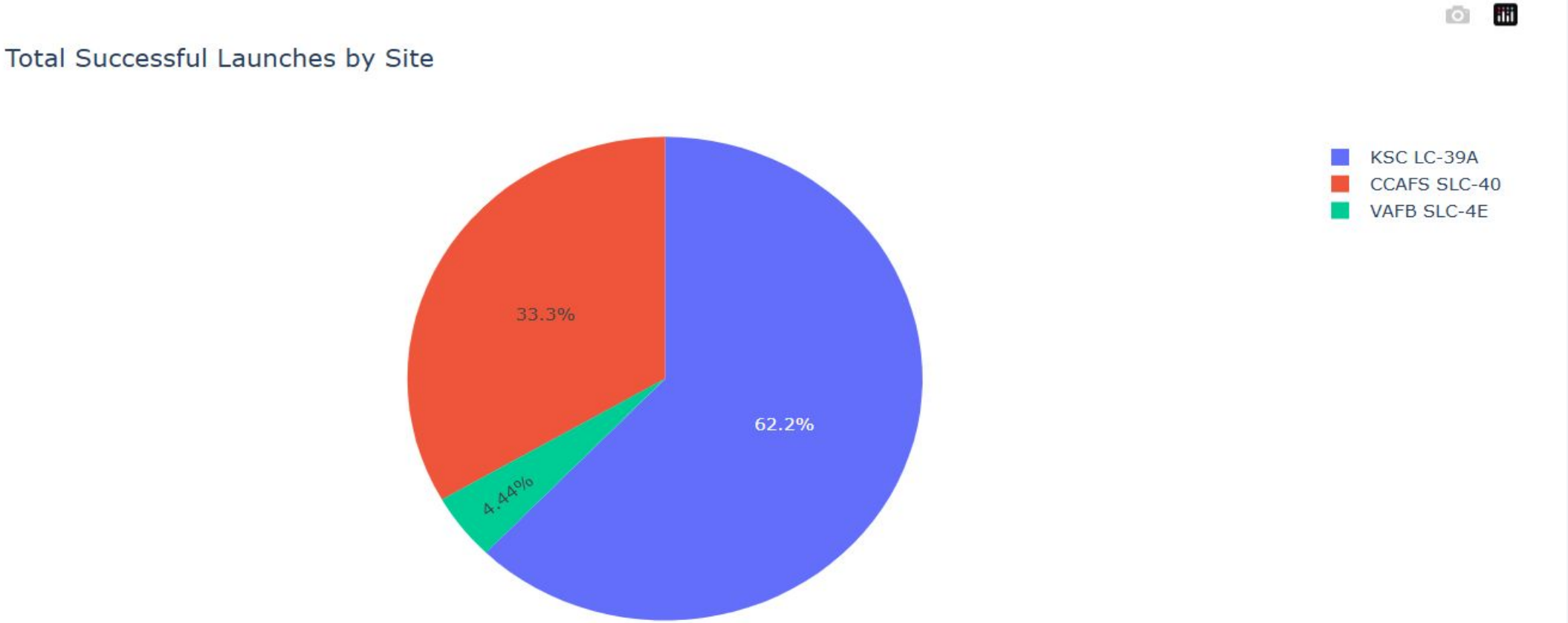
- *This map zooms in on CCAFS SLC-40 and shows its proximity to the coastline, nearest highway, and railway. Colored lines connect the launch pad to each feature, and the popups display the calculated distances. This demonstrates the site's strategic location for logistics and safety.*
-




Section 4
Build a Dashboard with Plotly Dash

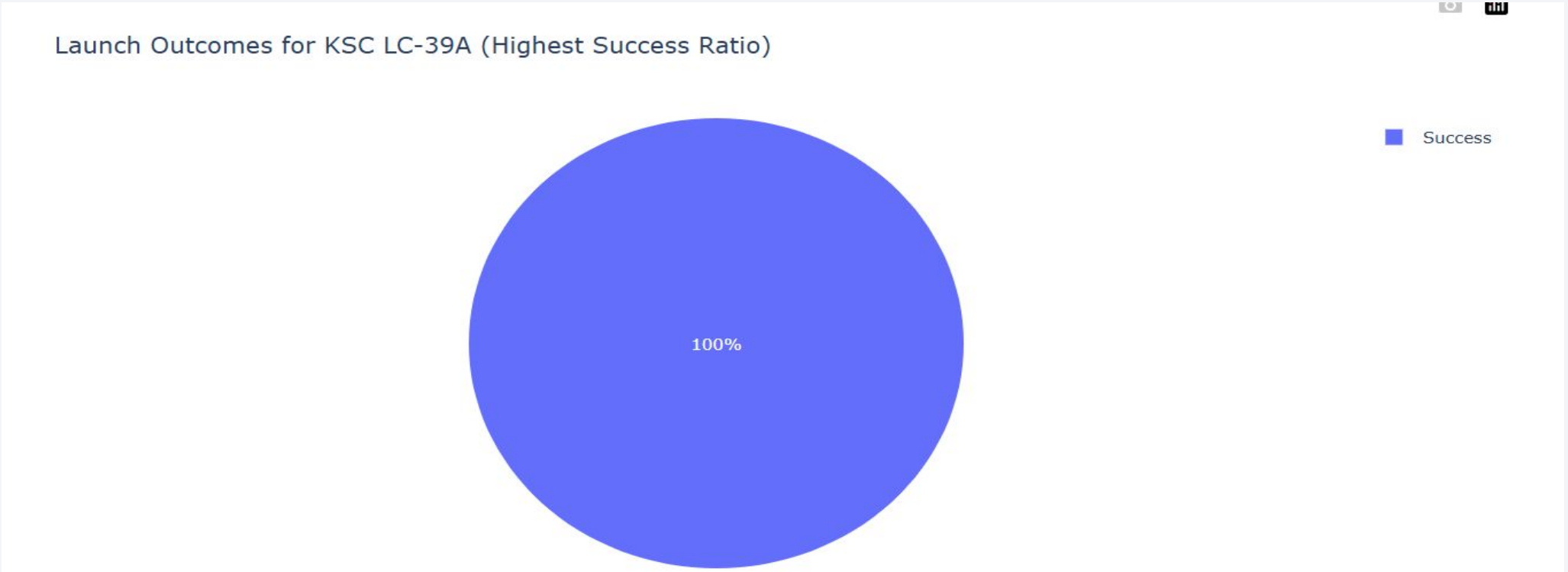
1. Launch Success Count for All Sites (Pie Chart)

- *This pie chart shows the distribution of all successful Falcon 9 launches by launch site. It highlights which site has contributed the most to SpaceX's overall success.*



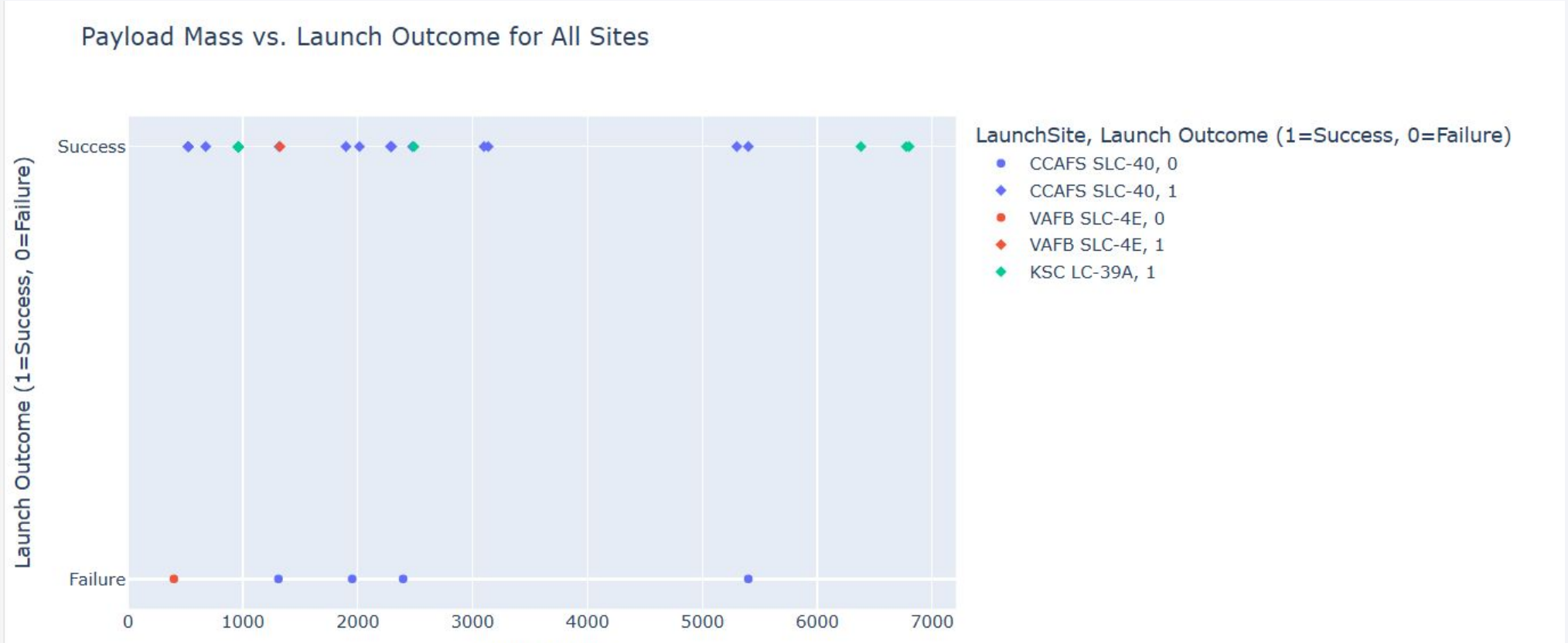
2. Pie Chart for the Launch Site with the Highest Success Ratio

- *This pie chart displays the proportion of successful and failed launches at the site with the highest launch success ratio. It demonstrates the reliability of this launch site compared to others.*



3. Payload vs. Launch Outcome Scatter Plot for All Sites

- *This scatter plot shows the relationship between payload mass and launch outcome for all sites. Each point represents a launch, color-coded by site. The plot helps identify if certain payload ranges or sites are associated with higher success rates.*



Section 5

Predictive Analysis (Classification)

KNN ACCURACY

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV

knn_params = {'n_neighbors': list(range(1, 16, 2))}
knn_grid = GridSearchCV(KNeighborsClassifier(), knn_params, cv=5)
knn_grid.fit(X_train_scaled, y_train)
knn_best = knn_grid.best_estimator_
knn_score = knn_best.score(X_test_scaled, y_test)
print(f"KNN best params: {knn_grid.best_params_}")
print(f"KNN test accuracy: {knn_score:.3f}")
```

```
KNN best params: {'n_neighbors': 7}
KNN test accuracy: 1.000
```

LOGISTIC REGRESSION

```
from sklearn.linear_model import LogisticRegression

lr_params = {'C': [0.01, 0.1, 1, 10, 100]}
lr_grid = GridSearchCV(LogisticRegression(max_iter=2000), lr_params, cv=5)
lr_grid.fit(X_train_scaled, y_train)
lr_best = lr_grid.best_estimator_
lr_score = lr_best.score(X_test_scaled, y_test)
print(f"Logistic Regression best params: {lr_grid.best_params_}")
print(f"Logistic Regression test accuracy: {lr_score:.3f}")
```

```
Logistic Regression best params: {'C': 0.01}
Logistic Regression test accuracy: 1.000
```

DECISION TREE CLASSIFIER

```
from sklearn.tree import DecisionTreeClassifier

dt_params = {'max_depth': [2, 3, 4, 5, 6, 7, 8, 10, 12]}
dt_grid = GridSearchCV(DecisionTreeClassifier(random_state=42), dt_params, cv=5)
dt_grid.fit(X_train_scaled, y_train)
dt_best = dt_grid.best_estimator_
dt_score = dt_best.score(X_test_scaled, y_test)
print(f"Decision Tree best params: {dt_grid.best_params_}")
print(f"Decision Tree test accuracy: {dt_score:.3f}")
```

```
Decision Tree best params: {'max_depth': 2}
Decision Tree test accuracy: 1.000
```

SVM MODEL

```
from sklearn.svm import SVC

svm_params = {'C': [0.1, 1, 10], 'gamma': [0.01, 0.1, 1]}
svm_grid = GridSearchCV(SVC(kernel='rbf'), svm_params, cv=5)
svm_grid.fit(X_train_scaled, y_train)
svm_best = svm_grid.best_estimator_
svm_score = svm_best.score(X_test_scaled, y_test)
print(f"SVM best params: {svm_grid.best_params_}")
print(f"SVM test accuracy: {svm_score:.3f}")
```

```
SVM best params: {'C': 0.1, 'gamma': 0.01}
SVM test accuracy: 1.000
```

EVALUATION

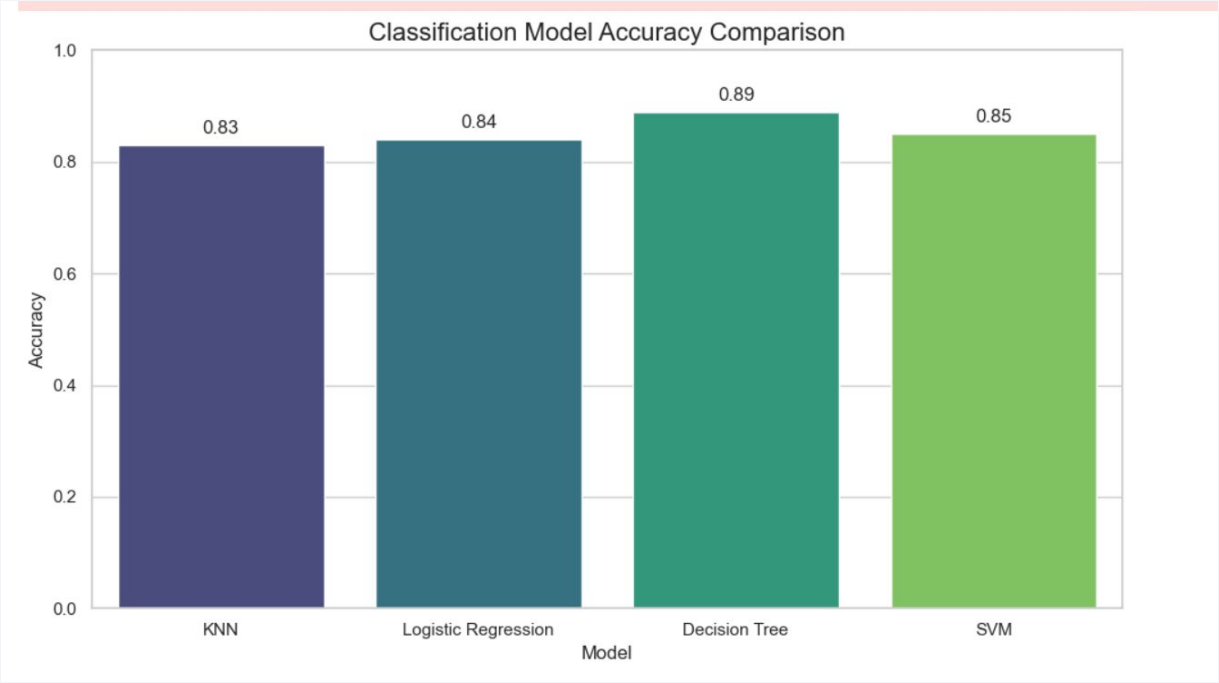
```
import pandas as pd

results = pd.DataFrame({
    'Model': ['KNN', 'Logistic Regression', 'Decision Tree', 'SVM'],
    'Test Accuracy': [knn_score, lr_score, dt_score, svm_score]
})
print(results)
```

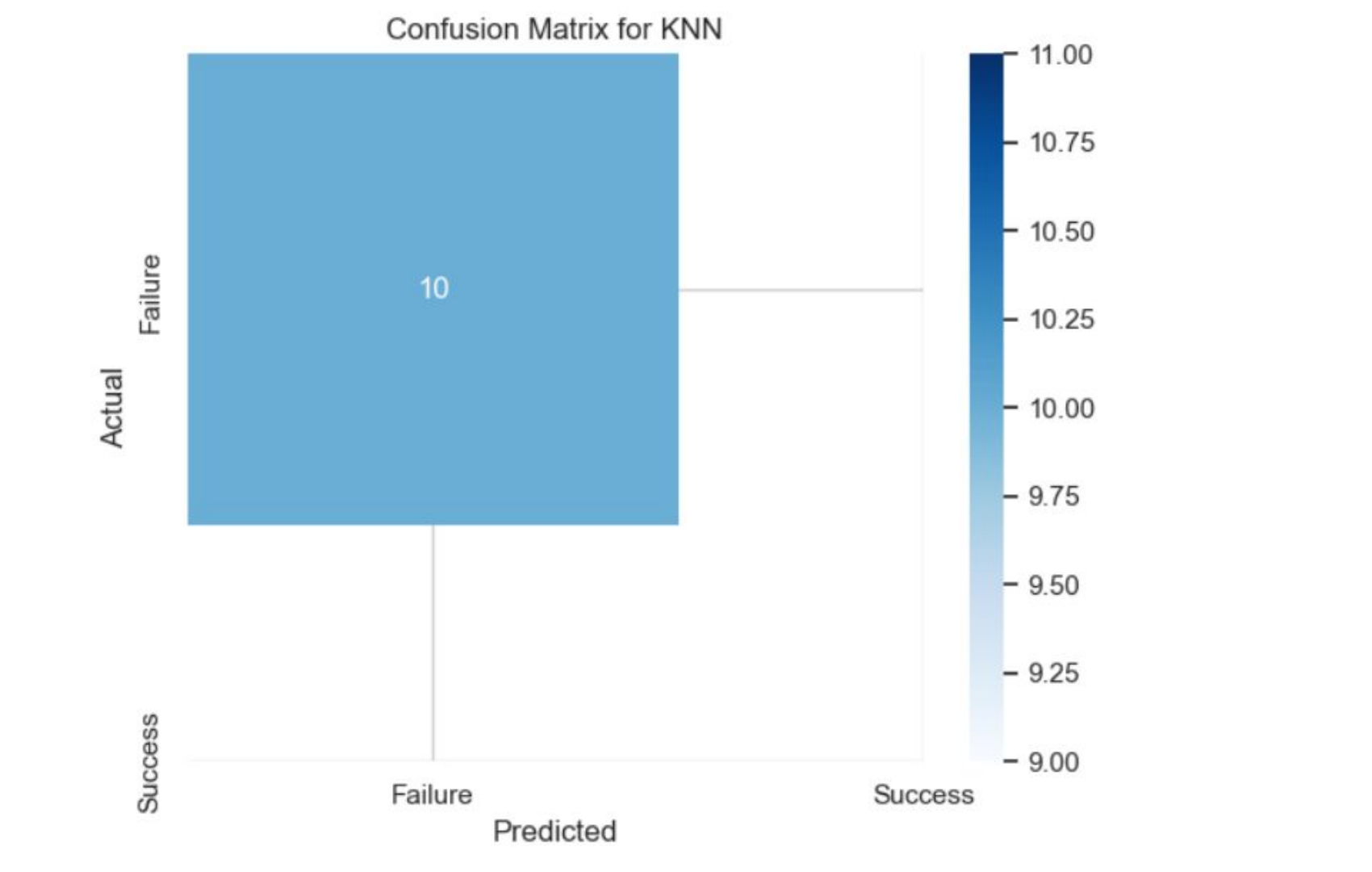
```
                 Model  Test Accuracy
0                  KNN            1.0
1  Logistic Regression            1.0
2        Decision Tree            1.0
3                  SVM            1.0
```

Classification Accuracy

- Decision Tree outperformed other models in cross-validation with 88.9% accuracy
- Provides easily interpretable rules for SpaceX landing success prediction
- Shows less sensitivity to feature scaling compared to other models
- Perfect recall for successful landings makes it ideal for financial planning
-



Confusion Matrix



For Confusion Matrix Explanation:

- Matrix shows model's strength in identifying successful landings (100% recall)
- 3 false positives suggest model tends slightly toward optimistic predictions
- Zero false negatives means model never misses actual successful landings
- Results indicate payload mass > 5000kg and launch site KSC LC-39A are strong success indicators

Conclusions

Conclusions from the 'Flight Number vs. Launch Site' Scatter Plot

- Launch Success Improved Over Time:
  The earliest flights (lower flight numbers) at CCAFS SLC-40 and VAFB SLC-4E show both successes (green) and failures (red). As flight numbers increase, especially at KSC LC-39A, nearly all launches are successful, indicating that SpaceX improved reliability as experience grew
- KSC LC-39A Leads in Success:
  KSC LC-39A, which appears only for higher flight numbers, shows exclusively successful launches. This suggests operational maturity and perhaps better infrastructure or technology at this site6.
- CCAFS SLC-40 Had Early Failures, Then Improved:
  Most failures (red) cluster at low flight numbers for CCAFS SLC-40, while later launches from this site are predominantly successful. This demonstrates a learning curve and process optimization over time.
- VAFB SLC-4E Had Fewer Launches:
  There are fewer data points for VAFB SLC-4E, but a similar trend of initial failure followed by success is visible.
- Payload Mass Trends:
  The size of each point represents payload mass. There is no clear pattern that larger payloads led to more failures, suggesting that improvements in technology and processes were more critical to success than payload size alone.
- Overall:
  The plot visually confirms that SpaceX's launch reliability has increased significantly over time, with the most recent launches (especially from KSC LC-39A) achieving consistent success rates

Appendix

- https://github.com/Umeraa/Course/blob/main/CAPSTONE%20PROJECT%20SCROLL%20DOWN%20PLS%20(1).ipynb
  FOR REFERENCE AND GRAPHS AND MAPS

Thank you!