IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nick

18th July 2025

Outline

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

Executive Summary

- Analyzed launch data with visualizations and SQL
- Developed and optimized predictive models
- Success rates improved over time
- Performance varies by launch site, payload, and orbit
- Decision Tree Model achieved 80% precision and 100% recall
- Insights enable SpaceY to better predict Falcon 9 landing outcomes to know where to compete with SpaceX

Introduction

Background

SpaceX has revolutionized space travel by successfully reusing booster rockets, significantly reducing the cost of space missions. Their Falcon 9 rocket's ability to land and be reused gives them a competitive edge in the commercial launch industry

Problem Statement

As a data scientist at *SpaceY*, our goal is to predict whether the Falcon 9's first stage will successfully land. Accurate predictions will help estimate launch costs and develop a cost-effective strategy to compete with SpaceX in the commercial launch market.

Section 1

Methodology

Methodology

Executive Summary

🔍 **Data Collection**

- **SpaceX API**:
  - Rockets, Launch Pads, Payloads, Cores, Past Launches
- **Web Scraping**:
  - Wikipedia page of Falcon 9 launches

🧹 **Data Wrangling & Preparation**

- Converted landing outcome to binary classes (Success / Failure)
- One-hot encoded categorical variables
- Imputed missing values
- Normalized data types

📊 **Exploratory Data Analysis (EDA)**

- Used SQL queries and visualization libraries
- Created interactive visual dashboards using Plotly Dash
- Geospatial mapping with Folium

🤖 **Predictive Modeling**

- Applied classification models:
  - Logistic Regression, SVM, Decision Tree, KNN
- Used GridSearchCV for hyperparameter tuning
- Evaluated models with Confusion Matrix to assess Type I and II errors

Data Collection

SpaceX API

- Pulled structured data on:
    - Past launches
    - Launchpads
    - Rocket Cores
    - Payload details
- Accessed via public REST API using Python requests module

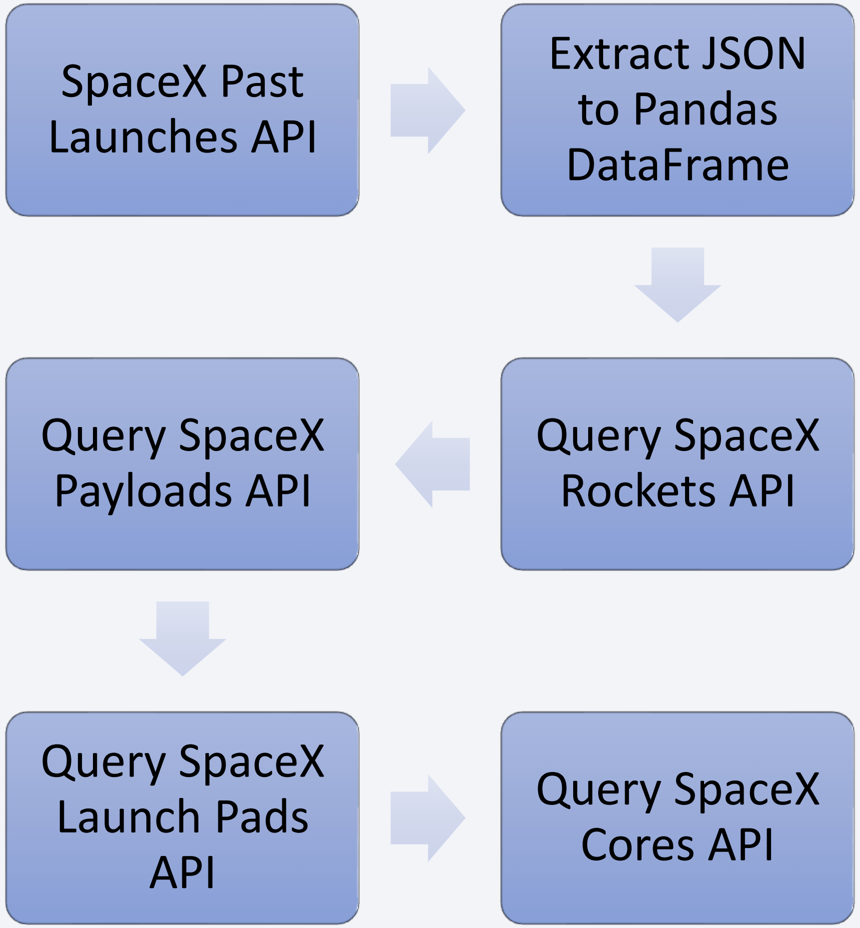
Web Scraping Wikipedia

- Scraped Falcon 9 launch data table
- Extracted additional features like mission names and launch outcomes
- Used BeautifulSoup for HTML parsing

Data Collection – SpaceX API

SpaceX API – Data Extraction

- Used Python's requests package to issue GET requests
- Queried the following endpoints:
  - /launches/past – Base dataset (static URL for consistency)
  - /rockets, /payloads, /launchpads, /cores – Supplemental details
- Parsed JSON responses and converted to Pandas DataFrame
- Merged datasets using common IDs (e.g., rocket_id, payload_id)



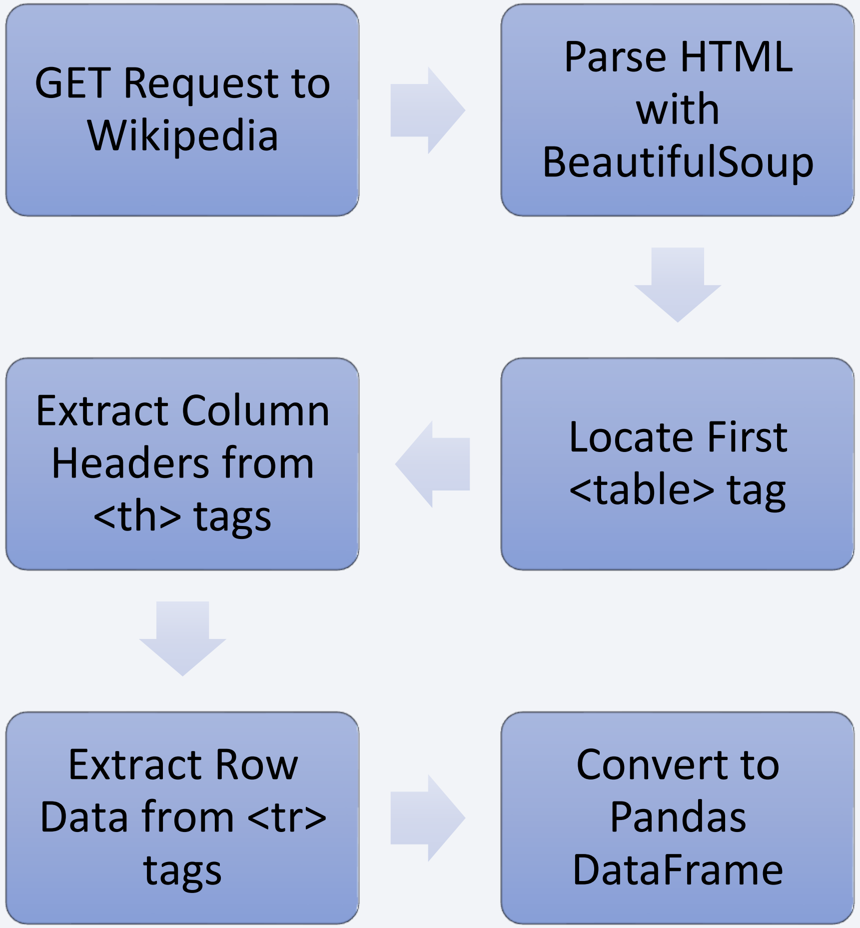
Github link

Data Collection - Scraping

◆ Wikipedia Web Scraping

- Issued a GET request to the Wikipedia page on Falcon 9 launches
- Parsed the HTML content using BeautifulSoup
- Located and extracted the first launch table using the <table> tag
- Extracted column headers by looping through all <th> tags
- Extracted row data by iterating through <tr> tags and capturing each cell’s content

[GitHub link]



Data Wrangling

◆ Merging & Structuring Data

- Merged launch data with payload, rocket, core, and launchpad data using unique IDs
- Flattened nested JSON structures from the API

◆ Cleaning & Normalization

- Converted landing outcome to binary labels (1 = Success, 0 = Failure)
- Standardized column datatypes

◆ Handling Missing Values

- Imputed missing payload masses with the mean

◆ Feature Engineering

- One Hot Encoded categorical variables

[GitHub link]

EDA with Data Visualization

- 🔍 **Exploratory Data Analysis (EDA)**
- Investigated how key variables relate to launch success
- Analyzed trends in:
  - **Payload mass**
  - **Flight number**
  - **Orbit type**
  - **Launch site**
  - **Year of launch**
- Identified patterns and correlations with landing outcomes
- Found improvement in success rate over time
- Guided feature selection for predictive modeling

🔗 GitHub link

EDA with SQL

🗄 SQL-Based Data Exploration

- Queried unique launch sites and filtered for those starting with 'CCA'
- Summed payloads for NASA CRS missions
- Calculated average payload for Falcon 9 v1.1
- Identified first successful landing on a ground pad
- Filtered for drone ship successes with payloads between 4000–6000 kg
- Counted total mission outcomes: Success vs. Failure
- Found boosters that carried the maximum payload
- Retrieved drone ship failures in 2015
- Ranked landing outcomes between 2010-06-14 and 2017-03-20

🔗 [GitHub link]

🗺️Interactive Map with Folium

- 🟢 **Circles for Launch Sites**
  Represented all known SpaceX launch sites to provide geographic context
- 🔴🟢 **Color-Coded Markers for Each Mission**
  Plotted individual launches, colored by success (green) or failure (red), to visualize performance across locations
- 📏 **Polylines to Shorelines & Railroads**
  Drew lines from launch sites to nearest shorelines and railroads to analyze proximity to infrastructure and its potential impact on launch logistics

🔗[GitHub link]

Dashboard with Plotly & Dash

- Visualized distribution of successful launches to identify top-performing sites
- Highlighted each sites success rate when filtered
- Analyzed payload size and booster version influence on mission success
- Enabled customization of scatter plot using Payload Mass slider

GitHub link

Predictive Analysis (Classification)

- Converted target to NumPy array and standardized features using StandardScaler
- Split data 80/20 into training and test sets
- Tuned Logistic Regression, SVM, Decision Tree, and KNN using Grid Search
- Used random_seed = 0 for reproducibility
- Evaluated each model’s accuracy and confusion matrix
- Selected best model through systematic tuning and evaluation

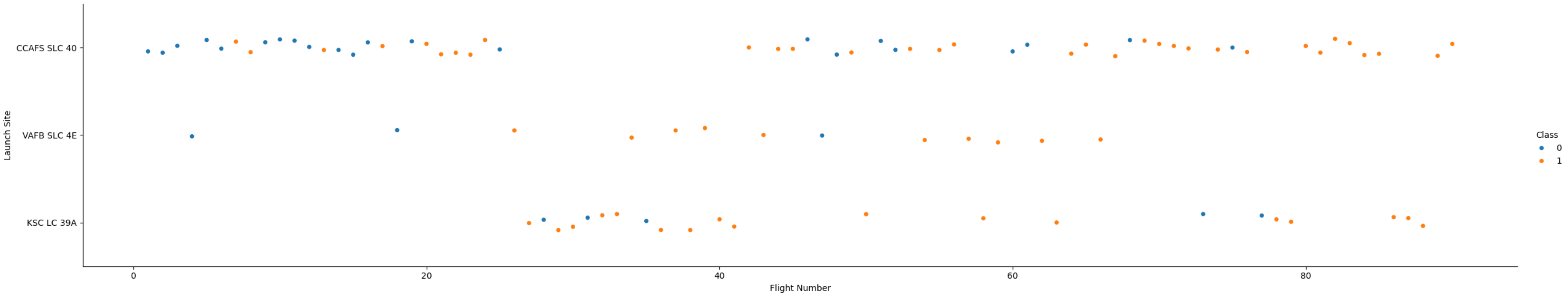
[GitHub link]

Results

- Conducted exploratory data analysis to understand data characteristics
- Developed interactive analytics dashboard
- Built and evaluated predictive models to assess performance

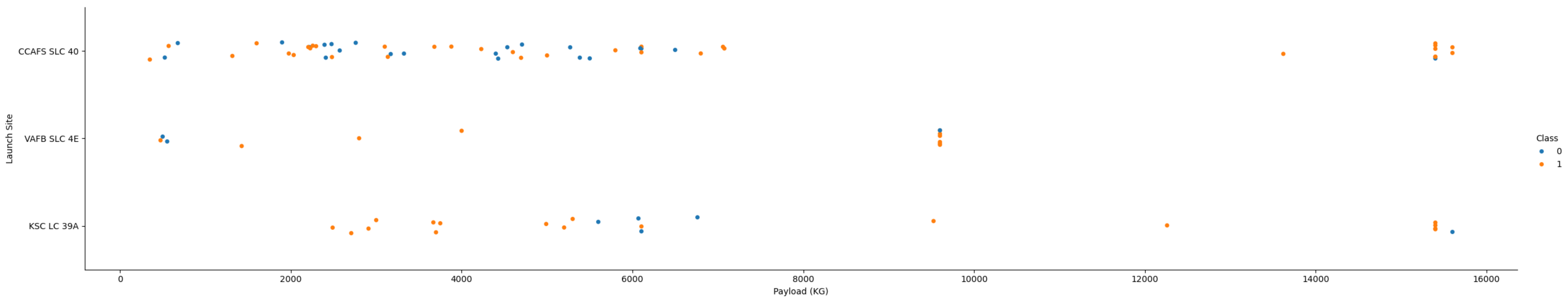
Section 2

Insights drawn from EDA



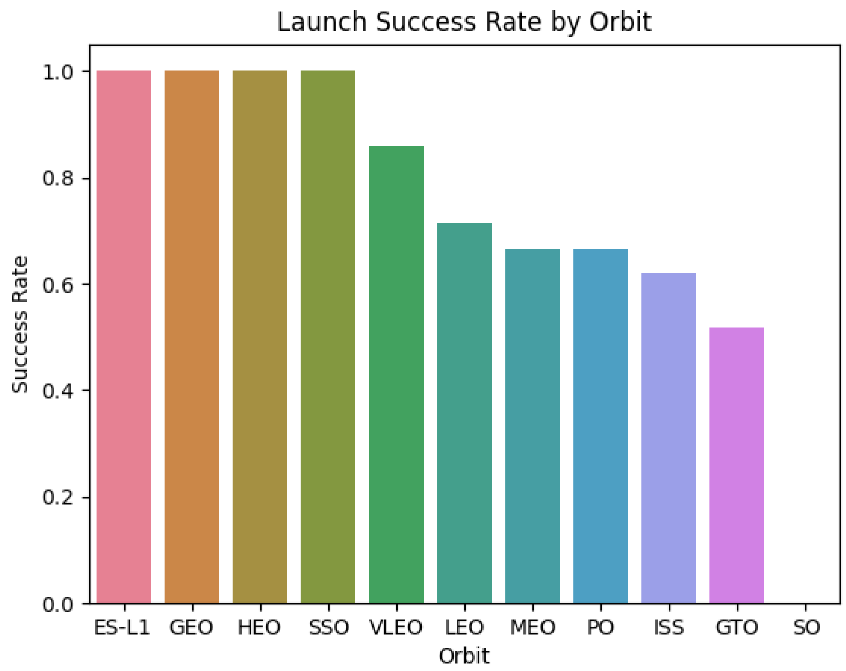
Flight Number vs. Launch Site

- Early SpaceX launches had high failure rates
- Success rate improved steadily with more launches
- Some variation in success by launch site



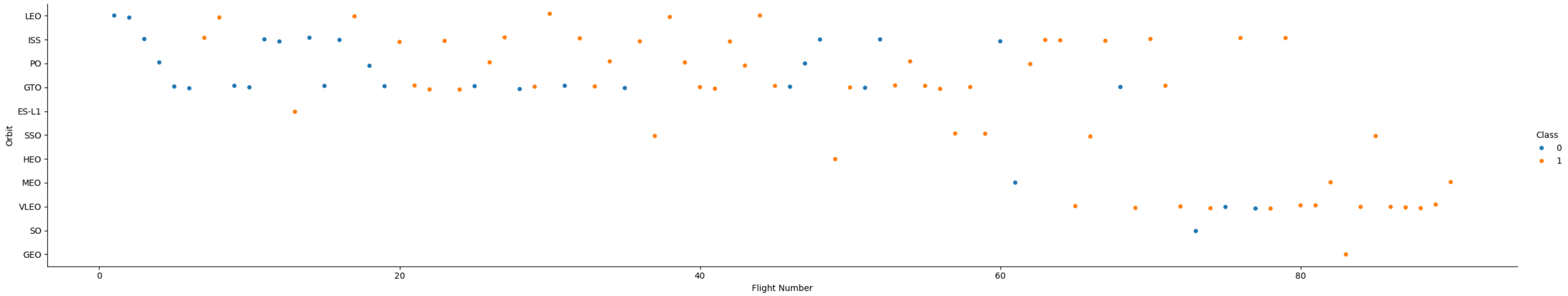
Payload vs. Launch Site

- Launch site performance varies by payload mass
- CCAFS SLC 40 has more successes with heavier payloads
- KSC LC 39A had most of its failures with payloads around 6000 KG



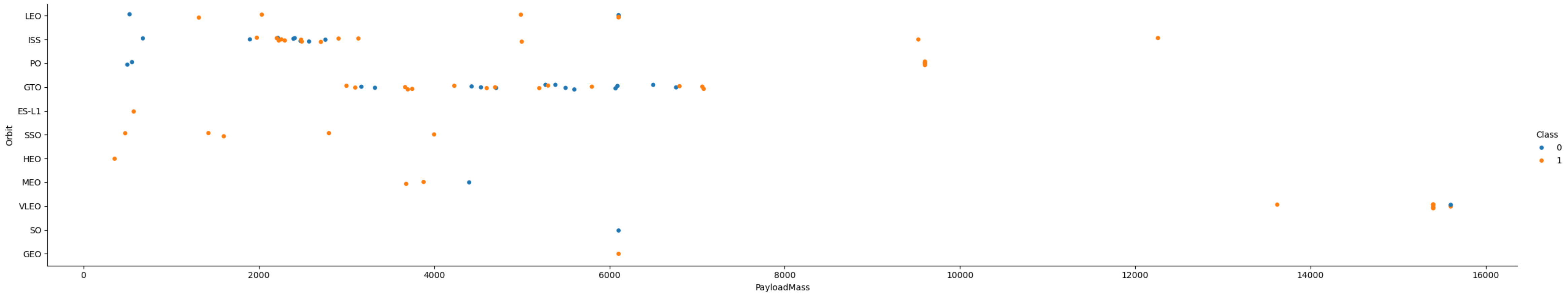
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, & SSO are the most successful orbits for launches
- SO has not had a successful launch



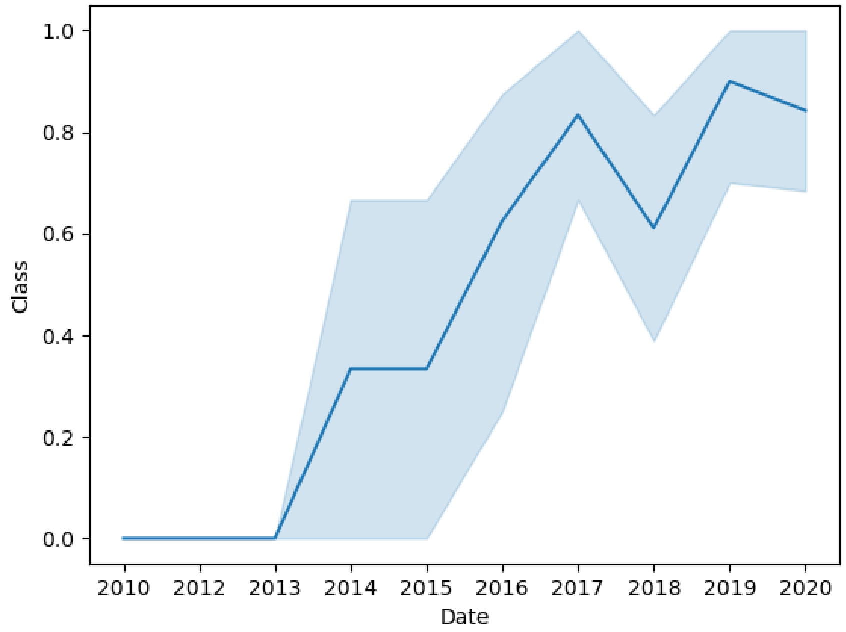
Flight Number vs. Orbit Type

- GEO, ES-L1, HEO & SO have only one recorded flight each
- The first flight was LEO which could attribute to its lower success rate
- Most successful orbits have limited flight data, reducing reliability of conclusions



Payload vs. Orbit Type

- Orbit success varies by payload weight
- GTO shows many successes around 4000 KG & 7000 KG
- VLEO & PO have higher success rates with payloads above 10,000 KG



Launch Success Yearly Trend

- Successful launches have increased over time

```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE
```

```
 * sqlite:///my_data1.db
Done.
```

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

All Launch Site Names

- Applied DISTINCT to show unique Launch Sites

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (paracl
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 (paracl
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No att
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No att
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No att

Launch Site Names Begin with 'CCA'

- Used the LIKE operator to find all Launch Sites that contain CCA

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

SUM("PAYLOAD_MASS__KG_")
45596

Total Payload Mass

- Aggregated Payload weights to find the total Payload Mass for NASA

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%';
```

```
 * sqlite:///my_data1.db
Done.
```

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Average Payload Mass by F9 v1.1

- Calculated average payload using AVG()
- Filtered results to the Booster Version matching F9 v1.1

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
2015-12-22

First Successful Ground Landing Date

- Filtered results to successful ground pad launches
- Used MIN to identify the first successful launch date

```
%sql SELECT DISTINCT(Booster_Version) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000
```

```
 * sqlite:///my_data1.db
Done.
```

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- Retrieved unique booster version names with DISTINCT
- Filtered results to Successful Drone Ship landing outcomes
- Filtered the payload to between 4000 and 6000 KG

Total Number of Successful and Failure Mission Outcomes

- Used a CASE statement to categorize landings as successful or failed
- Calculated totals using COUNT()
- Used GROUP BY to group the counts by the successes and failures

```
%sql SELECT CASE WHEN Landing_Outcome LIKE 'Success%' THEN 'Success' WHEN Landing_Outcome LIKE 'Failure%' THEN 'Failure'
```

```
 * sqlite:///my_data1.db
Done.
```

Outcome	Total
None	30
Failure	10
Success	61

Boosters Carried Maximum Payload

- Used a subquery to find the max payload
- Filtered the table to where the payloads equaled the max
- Used DISTINCT to find all the Boosters that had the max payload

```
%sql SELECT DISTINCT(booster_version) FROM SPACEXTABLE WHERE "Payload_Mass__KG_" = (SELECT MAX("Payload_Mass__KG_") FROM SP
```

```
 * sqlite:///my_data1.db
Done.
```

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

- Used the SUBSTR function with a CASE statement to convert month numbers to names
- Filtered data to year 2015 and launch failures

```
ql
 SELECT CASE
     WHEN SUBSTR(Date, 6, 2) = '01' THEN 'January'
     WHEN SUBSTR(Date, 6, 2) = '02' THEN 'February'
     WHEN SUBSTR(Date, 6, 2) = '03' THEN 'March'
     WHEN SUBSTR(Date, 6, 2) = '04' THEN 'April'
     WHEN SUBSTR(Date, 6, 2) = '05' THEN 'May'
     WHEN SUBSTR(Date, 6, 2) = '06' THEN 'June'
     WHEN SUBSTR(Date, 6, 2) = '07' THEN 'July'
     WHEN SUBSTR(Date, 6, 2) = '08' THEN 'August'
     WHEN SUBSTR(Date, 6, 2) = '09' THEN 'September'
     WHEN SUBSTR(Date, 6, 2) = '10' THEN 'October'
     WHEN SUBSTR(Date, 6, 2) = '11' THEN 'November'
     WHEN SUBSTR(Date, 6, 2) = '12' THEN 'December'
 END AS month, Landing_Outcome, booster_version, launch_site
 FROM SPACEXTABLE
 WHERE SUBSTR(Date, 0, 5) = '2015' AND Landing_Outcome = 'Failure
```

```
lite:///my_data1.db
```

ıth	Landing_Outcome	Booster_Version	Launch_Site
ary	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
pril	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Used COUNT() to get the counts for each landing outcome
- Filtered data between 2010-06-04 and 2017-03-20
- Grouped results by landing outcome
- Ordered by counts descending to rank outcomes

```
%%sql
SELECT landing_outcome, COUNT(*) AS totals
FROM SPACEXTABLE
WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY 2 DESC;
```

```
* sqlite:///my_data1.db
one.
```

Landing_Outcome	totals
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2

Section 3

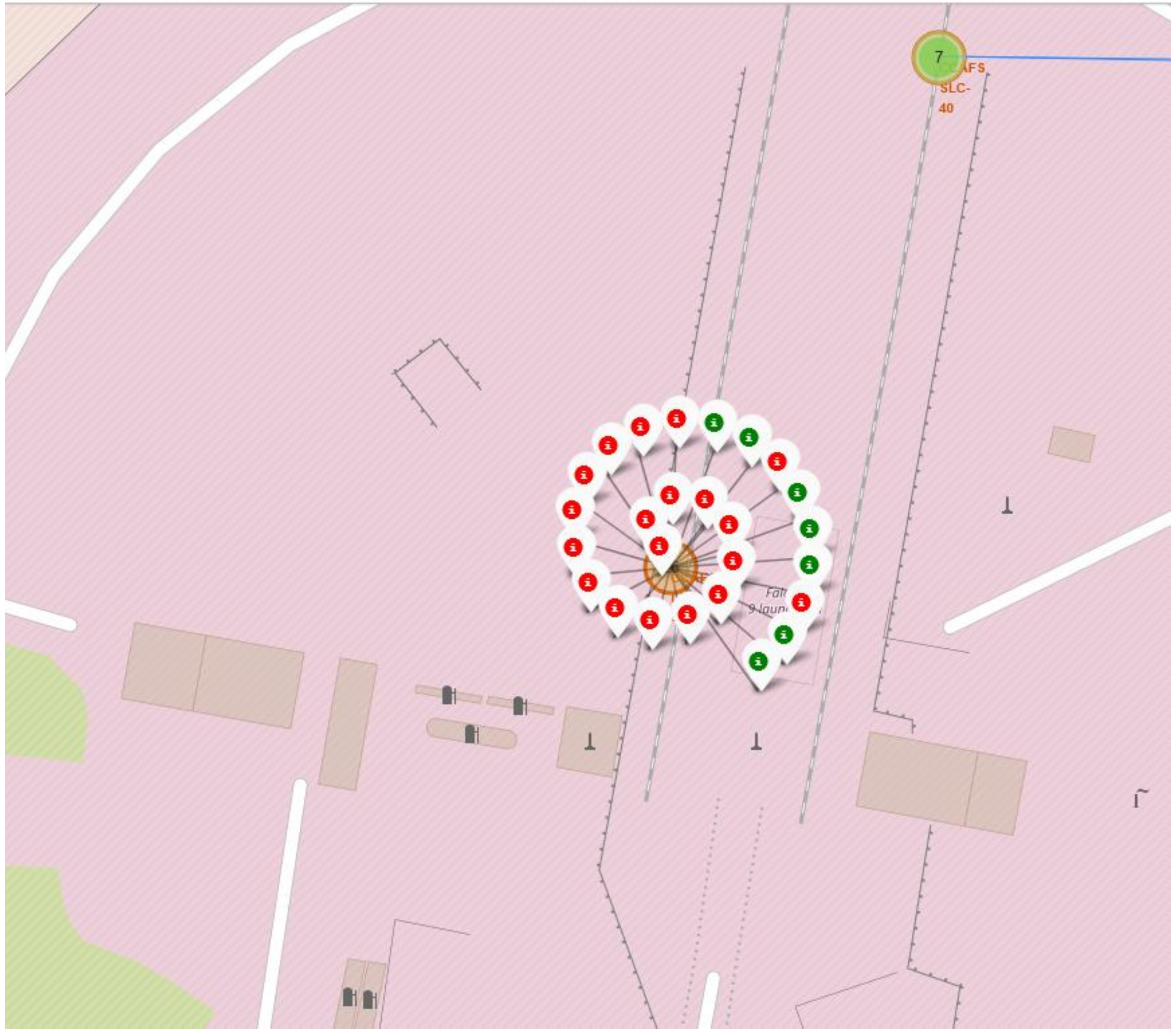
Launch Sites Proximities Analysis

Launch Site Locations

- Mapped all SpaceX Launch Site Locations
- All sites are in the United States
- Launch sites located in Florida and California

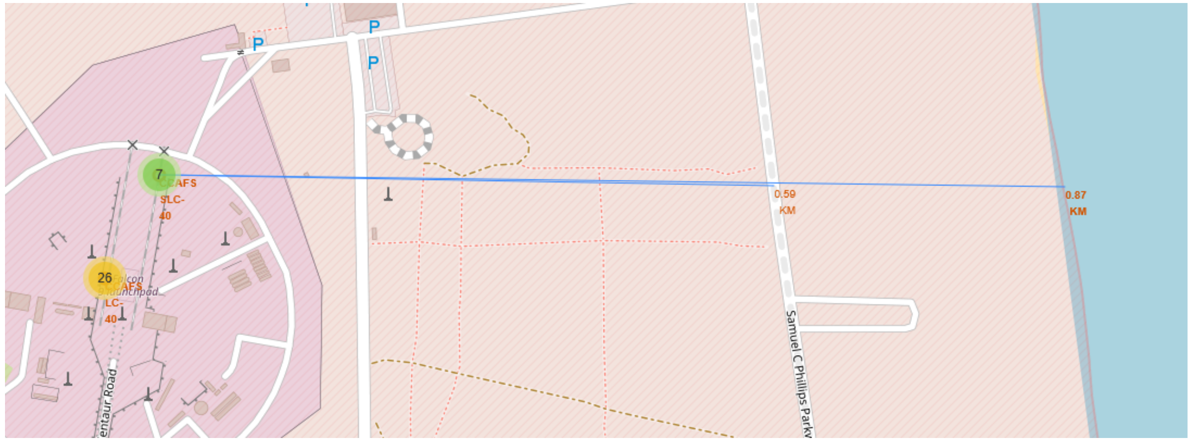
Launch Site Landing Outcomes

- Displayed individual launch site with all associated launches
- Launch outcomes are color coded
  - Green = Success
  - Red = Failure



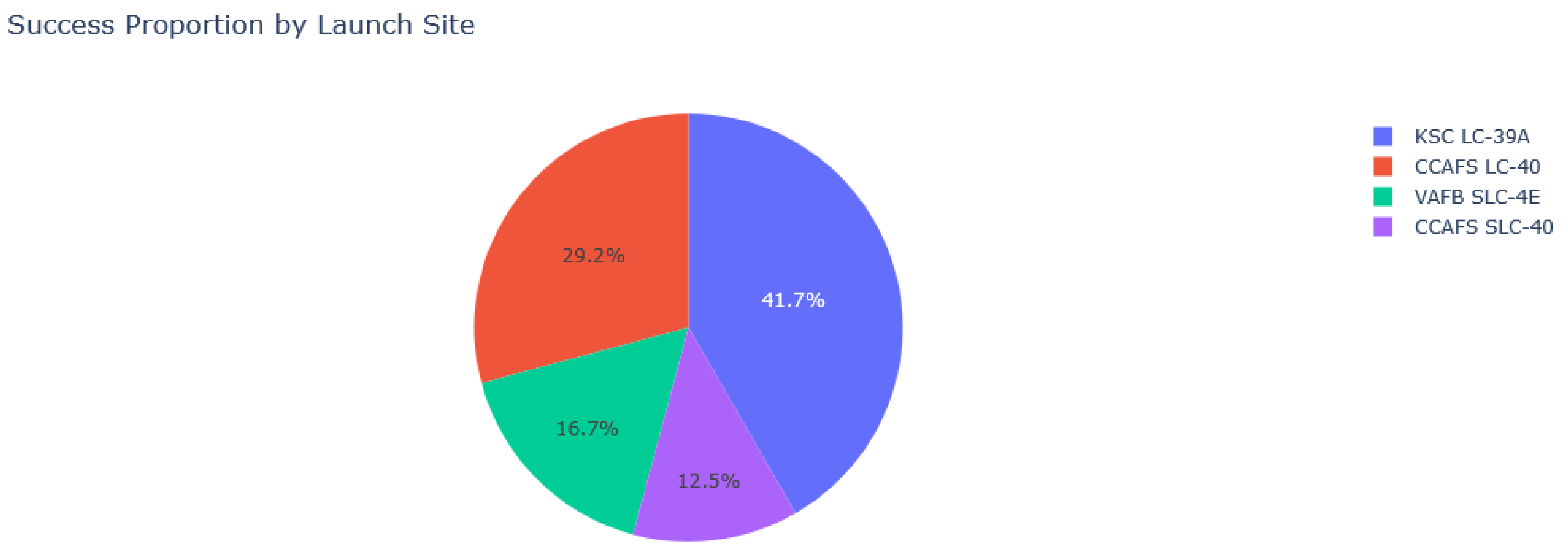
Launch Site Distance to Populace

- Visualized distance from launch site to nearby infrastructure
- Example Launch Site is 60KM from the closest railroad
- Example Launch Site is 90 KM from the shoreline



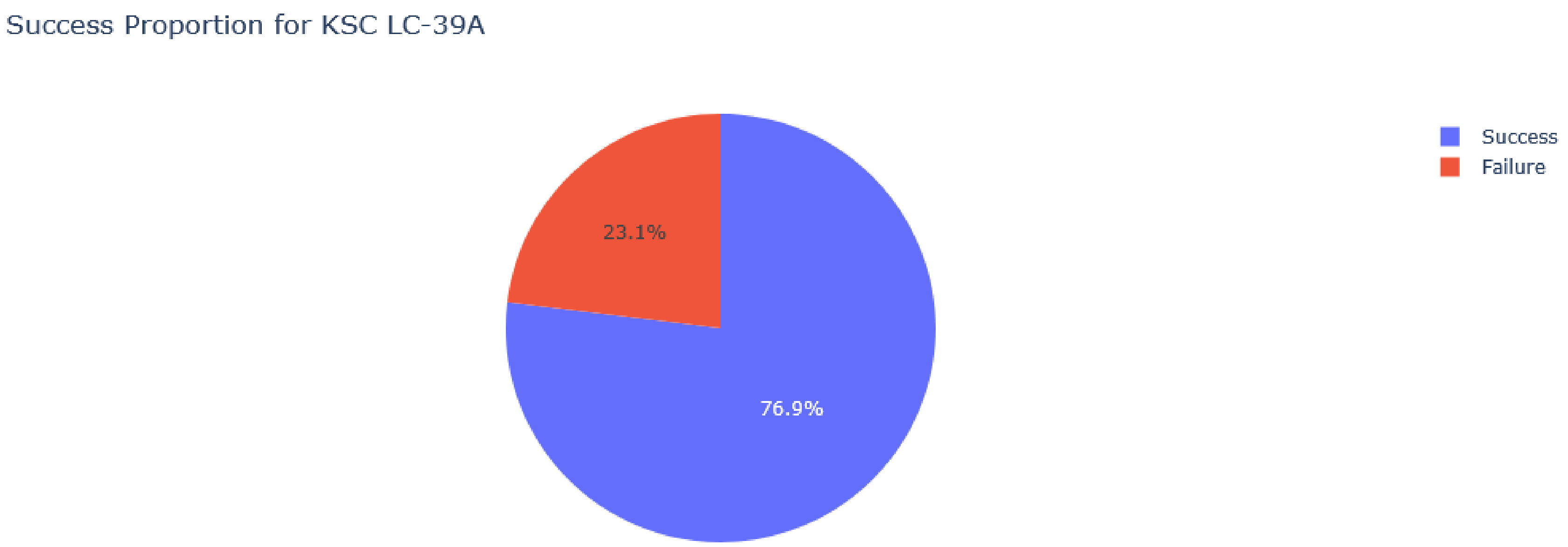
Section 4

Build a Dashboard with Plotly Dash



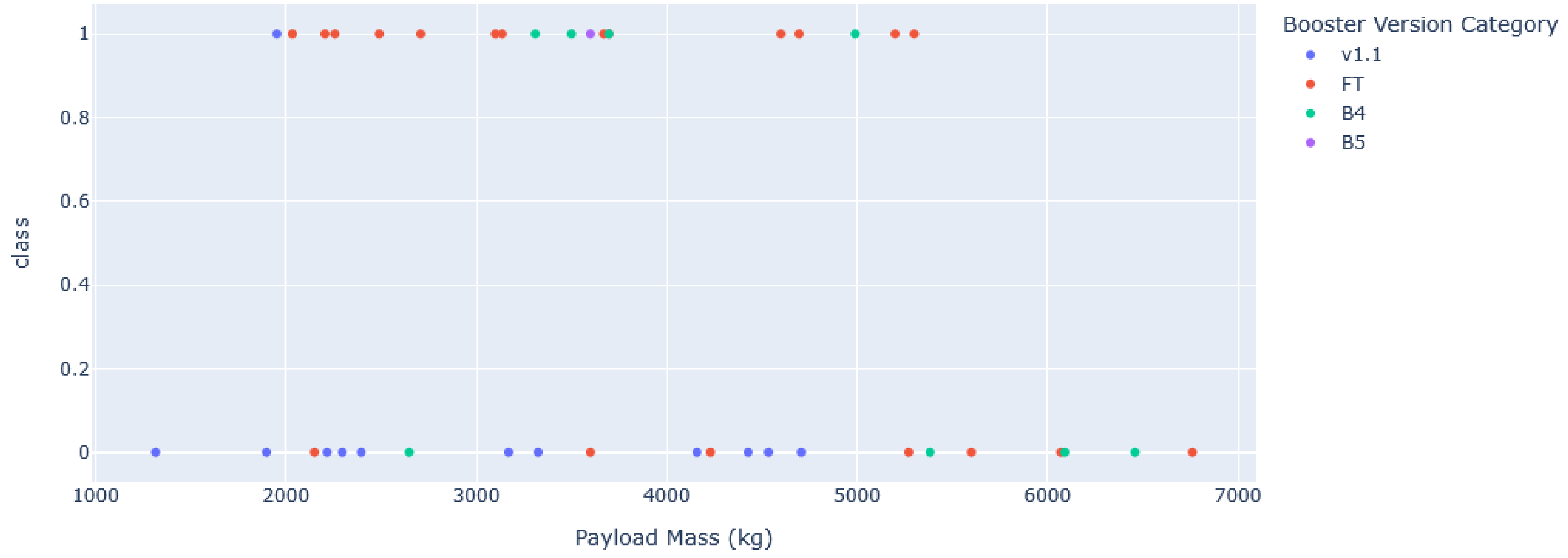
- KSC LC-39A has the most successful launches
- CCAFS SLC-40 has the least successful launches

Launch Sites Success Rates



- Success and Failure proportion of specific launch sites
- KSC LC-39A is successful in ~77% of its launches

Launch Site Success Rate



- Payloads < 4000 KG has more successes
- Booster FT has the most successes

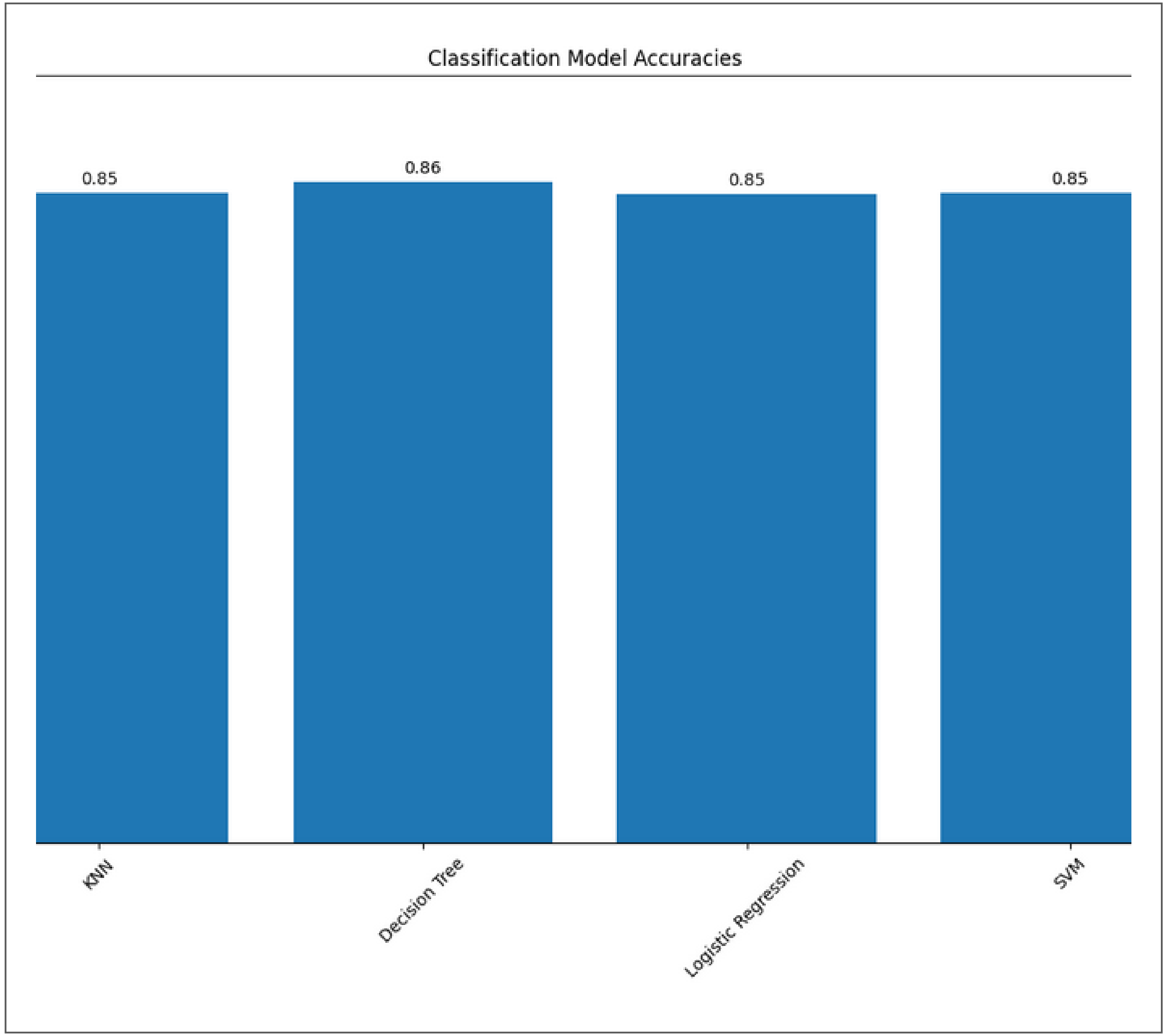
Booster Successes by Payload

Section 5

Predictive Analysis (Classification)

Classification Accuracy

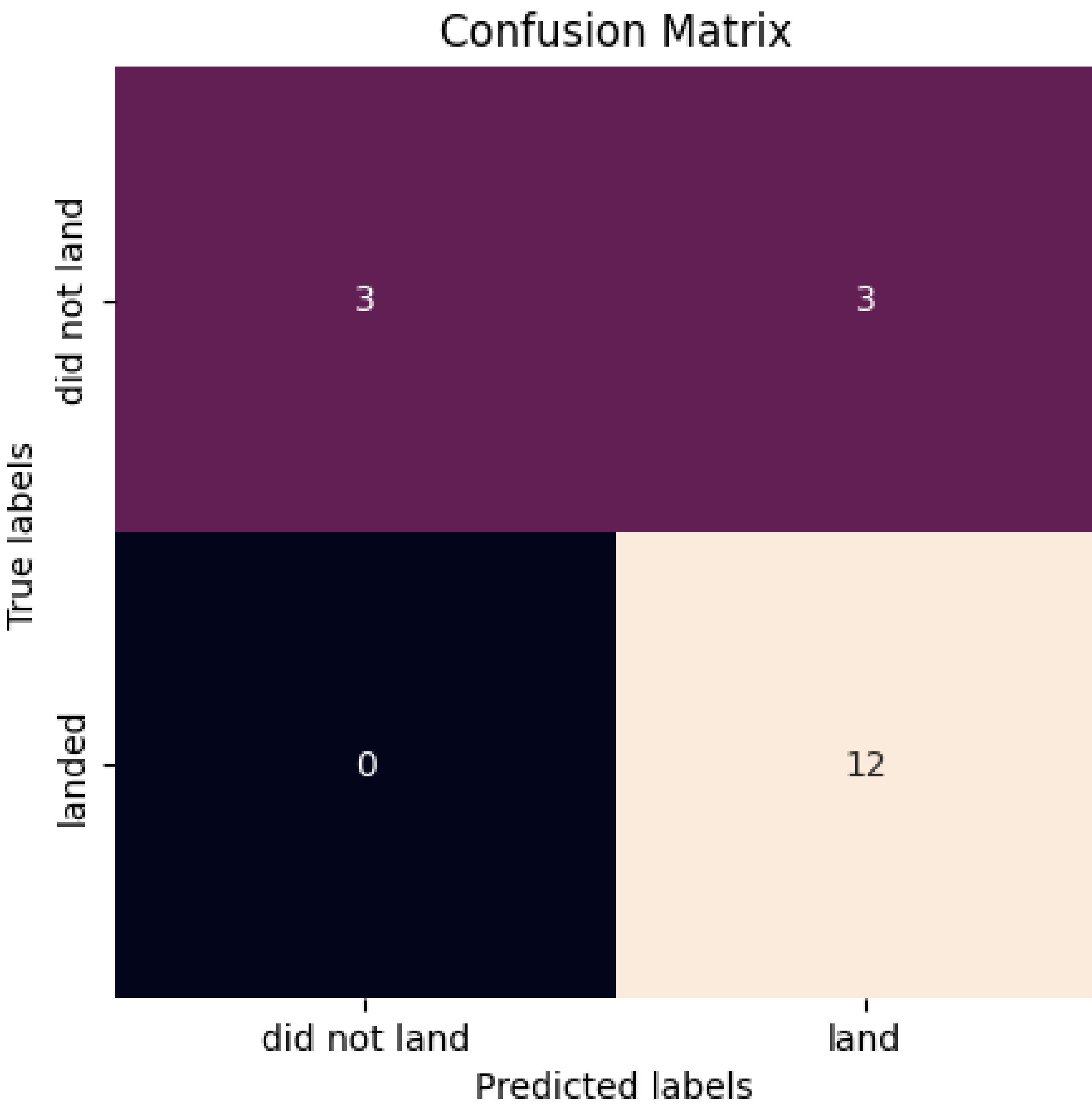
- Decision Tree has the highest Classification Accuracy on the training set
- Similar accuracies were found on the test set indicating good generalization
- Best hyperparameters
  - Criterion: gini
  - Max Depth: 6
  - Max Features: sqrt
  - Min Samples Left: 4
  - Min samples split: 10
  - Splitter: random



Confusion Matrix

Precision: 80% (correct positive predictions)

Recall: 100% (all actual positives identified)



Conclusions

- SpaceX launch success rates have improved each year
- Launch site performance varies notably by payload and orbit
- Decision Trees provided the most accurate classification of launch outcomes
  - Supported by strong precision and high recall
- Precision showing some false positives which could affect the pricing

Thank you!