IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yong Gil Hwang
Jan 8,2025

Outline

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

Executive Summary

- Summary of methodologies: The project focuses on leveraging SpaceX launch data to predict the likelihood of first-stage rocket landings. The methodologies include:
- **Data Collection**: SpaceX REST API is employed to fetch data on launches. Web scraping techniques, using the BeautifulSoup library, extract historical Falcon 9 launch data from Wikipedia.
- **Data Wrangling**: JSON responses from the API are normalized into Pandas DataFrames. Filtering to exclude Falcon 1 launches ensures data relevance. Handling missing values by calculating the mean for numeric columns (e.g., PayloadMass) and replacing nulls with the calculated mean.
- **Feature Engineering**: Extracting specific attributes such as booster versions, payload details, launch sites, and landing outcomes. One-hot encoding categorical variables for compatibility with machine learning algorithms.
- **Exploratory Data Analysis (EDA):**Relationships between variables like PayloadMass, LaunchSite, and OrbitType are visualized using bar charts, scatter plots, and line graphs. Success rates are analyzed across orbit types and over time
- **Landing Prediction models**: A pipeline evaluates predictive models (Logistic Regression, SVM, Decision Trees, KNN) to determine the optimal approach for predicting first-stage landings. Model tuning and validation are performed using grid search and cross-validation.
- **Dashboard Implementation**: An interactive dashboard built with Plotly Dash provides real-time visual analytics, including payload mass vs. orbit success rate and yearly success trends.

Executive Summary

- **Summary of all results**

Based **on** the comprehensive data analysis and modeling:

- Landing Predictions: Logistic regression and decision trees are likely to yield the highest accuracy in predicting landing outcomes, given their interpretability and performance in similar scenarios.
- Insights: High success rates are observed for Polar, LEO and ISS with heavy payloads. The sucess rate since 2013 kept increasing till 2020. Launch sites with robust infrastructure exhibit better landing success.
- Dashboard Insights: Users gain actionable insights into key metrics, aiding decision-making for future missions.

Conclusions: This approach ensures a robust framework for understanding SpaceX's operations and evaluating the feasibility of first-stage landings, paving the way for cost optimization in aerospace missions.

Introduction

- **Project background and context**

- The project focuses on analyzing SpaceX launch data to understand and predict the likelihood of successful first-stage rocket landings, utilizing data from APIs, web scraping, and machine learning. This initiative aims to optimize the cost-effectiveness of space missions, leveraging SpaceX's reusable rocket technology.

- **Problems you want to find answers**

- The primary questions include: What factors influence the success of landings? How can landing outcomes be predicted accurately? Additionally, how do payload characteristics, launch sites, and orbit types affect success rates?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Describe how data sets were collected.
- The data was collected using two main approaches:

-SpaceX REST API:Launch data, including rocket details, payloads, launch specifications, and landing outcomes, was retrieved from the SpaceX API endpoints.JSON responses were normalized into Pandas DataFrames for further analysis.

-Web Scraping:Historical Falcon 9 launch records were extracted from a Wikipedia page using the BeautifulSoup library.HTML tables were parsed and converted into structured Pandas DataFrames.

These methods ensured comprehensive and reliable data acquisition for the analysis

Data Collection – SpaceX API

Data collection process

Identify Data Sources:

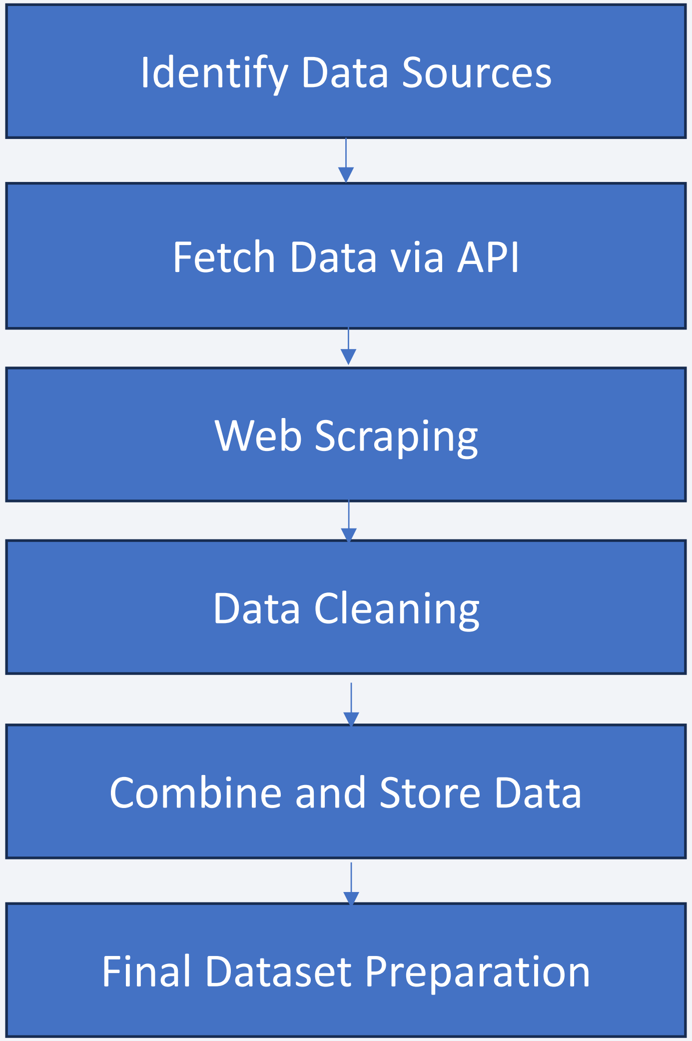
- SpaceX REST API for structured launch data.
- Wikipedia page for historical Falcon 9 and Falcon Heavy launch records.

1. Fetch Data via API:
   - Use the SpaceX API endpoint: `https://api.spacexdata.com/v4/launches/past`.
   - Perform GET requests using Python's `requests` library.
   - Parse JSON responses and normalize them into Pandas DataFrames.

2. API Call Process:
   - Target endpoints for additional details:
   - Use IDs from the initial response to query these endpoints.

GitHub:https://github.com/plaisirs30/AppliedDataScience/blob/main/jupyter-labs-spacex-data-collection-api.ipynb)

Flow Chart



Data Collection - Scraping

- Web scraping process

- Identify Target Page: Wikipedia page: "List of Falcon 9 and Falcon Heavy launches".

- Request Page Content: Use Python’s requests library to fetch the HTML content.

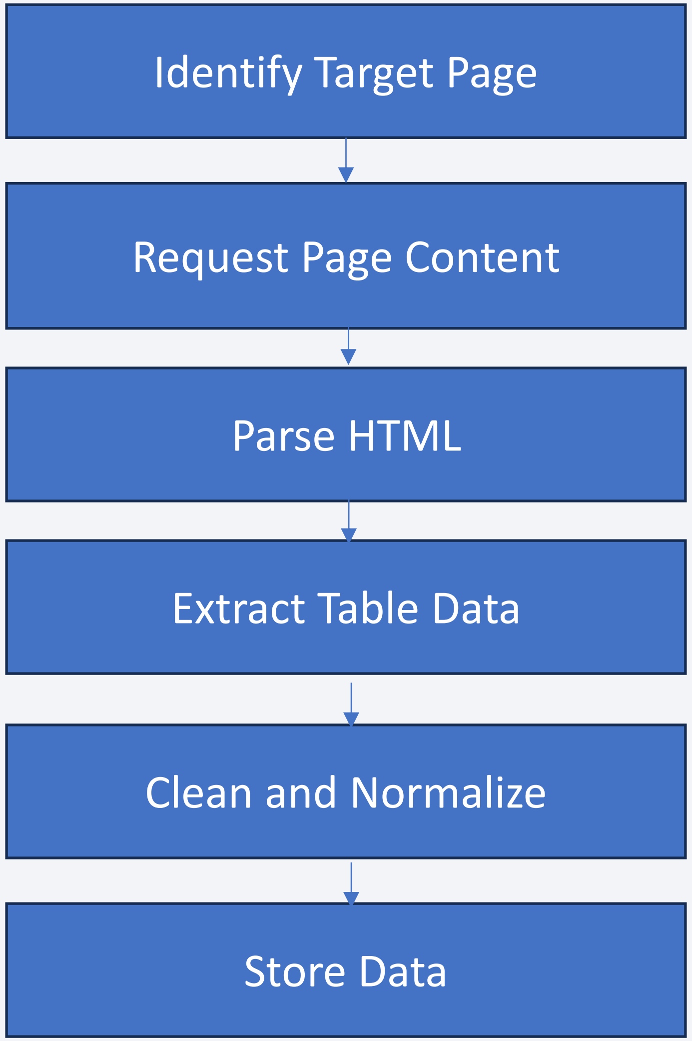
- Parse HTML: Utilize BeautifulSoup to parse HTML tables from the page.

- Extract Table Data: Identify relevant tables containing launch records and extract rows.

- Clean and Normalize: Remove annotations, handle missing values, and format data.

- Store Data: Convert parsed data into a Pandas DataFrame for analysis

[GitHub](https://github.com/plaisirs30/AppliedDataScience/blob/main/jupyter-labs-webscraping.ipynb)
https://github.com/plaisirs30/AppliedDataScience/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Data Cleaning:

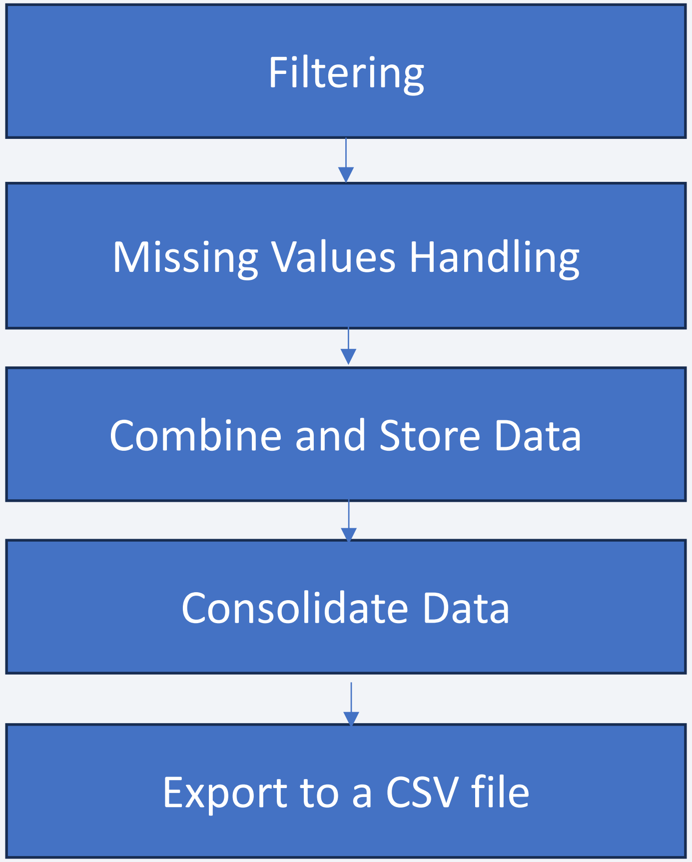
 - Filter Falcon 1 launches.

 - Handle missing values, e.g., calculate and replace null values in PayloadMass.

 - Combine and Store Data

 - Consolidate API and scraped data into a single DataFrame

 - Export the final dataset to a CSV file for further analysis.



EDA with Data Visualization

Plotted Charts and Their Purpose:

- **Flight Number vs Launch Site**: A categorical plot to examine the distribution of flight numbers across different launch sites, revealing site utilization trends.
- **Payload Mass vs Launch Site**: A scatter plot to identify relationships between payload mass and launch sites, highlighting site-specific payload capacities.
- **Success Rate by Orbit Type**: A bar chart to compare success rates across orbit types, helping identify orbits with higher landing success.
- **Flight Number vs Orbit Type**: A scatter plot to study the relationship between flight number and orbit type, offering insights into mission trends over time.
- **Payload Mass vs Orbit Type**: A scatter plot to explore how payload mass affects success rates across different orbit types, identifying favorable conditions.
- **Yearly Success Rate Trend**: A line chart to visualize annual trends in launch success, providing a temporal performance overview

GitHub: https://github.com/plaisirs30/AppliedDataScience/blob/main/edadataviz%20(1).ipynb

EDA with SQL

SQL Queries Performed:

- Unique Launch Sites: Query to list the names of unique launch sites.
- Launch Records: Retrieve five records where launch sites begin with 'CCA'.
- Total Payload Mass: Calculate the total payload mass carried by NASA missions.
- Average Payload Mass: Compute the average payload mass for booster version F9 v1.1.
- First Successful Landing: Identify the date of the first successful ground pad landing.
- Successful Drone Ship Missions: List booster names with successful drone ship landings and payload mass between 4000 and 6000.
- Mission Outcomes: Count total successful and failed mission outcomes.
- Maximum Payload Booster: Determine the booster version carrying the maximum payload mass.
- 2015 Failure Records: List failure records for drone ship landings in 2015 by month, booster version, and launch site.
- Landing Outcome Rankings: Rank landing outcomes between 2010-06-04 and 2017-03-20 in descending order

GitHub https://github.com/plaisirs30/AppliedDataScience/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map Objects Added to Folium Map:

- Markers: Placed at each launch site to visually identify the geographical location of SpaceX launch pads.
  - Reason: To provide a clear visualization of where launches occurred.
- Circles: Added around launch sites to indicate the area of activity.
  - Reason: To represent the impact or vicinity of launch operations.
- Lines: Drawn to connect launch sites to their corresponding landing locations (if applicable).
  - Reason: To depict the trajectory or relationship between launch and landing sites visually.
- Popups: Included with markers to display detailed information, such as the name of the site and its coordinates.
  - Reason: To provide additional context and interactive details for the user.

GitHub https://github.com/plaisirs30/AppliedDataScience/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

Dashboard Plots/Graphs and Interactions:

- Success-Pie Chart: Displays the proportion of successful vs. failed landings.
  - Reason: To provide an overview of the mission success rate.
- Payload Scatter Chart: Visualizes the relationship between payload mass and success for specific orbits.
  - Reason: To identify trends in payload performance and landing success.
- Interactive Dropdowns: Allows users to filter data by launch site and payload range.
  - Reason: To customize the analysis based on specific user interests.
- Line Chart for Success Trends: Plots the annual success rate over time.
  - Reason: To track the improvement in launch performance year by year.
- Orbit Success Bar Chart: Compares success rates across different orbit types.
  - Reason: To identify which orbits are more conducive to successful landings.

GitHub https://github.com/plaisirs30/AppliedDataScience/blob/main/CapstoneLab5_DASH.py

Predictive Analysis (Classification)

Model development process

- **Data preprocessing**

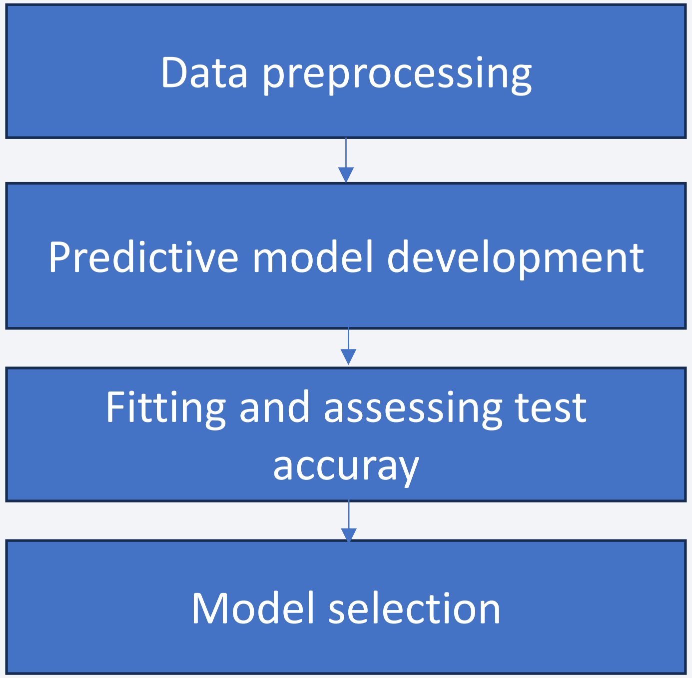
- create a numpy array from the column <Class>,
- Standardize the data in X
- assigning into training and test data set using train_test_split funciotn

- **Model development**

- Create objects for various models (logistic regression, SVM, decision tree, KNN)

- Create GridSearchCV objects
- Fitting for best parameters and best accuracy
- Test accuracy on the test data and plotting confusion matrix
- the accuracy on the test data

* **Model selection:** compare test accuracy



GitHub https://github.com/plaisirs30/AppliedDataScience/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

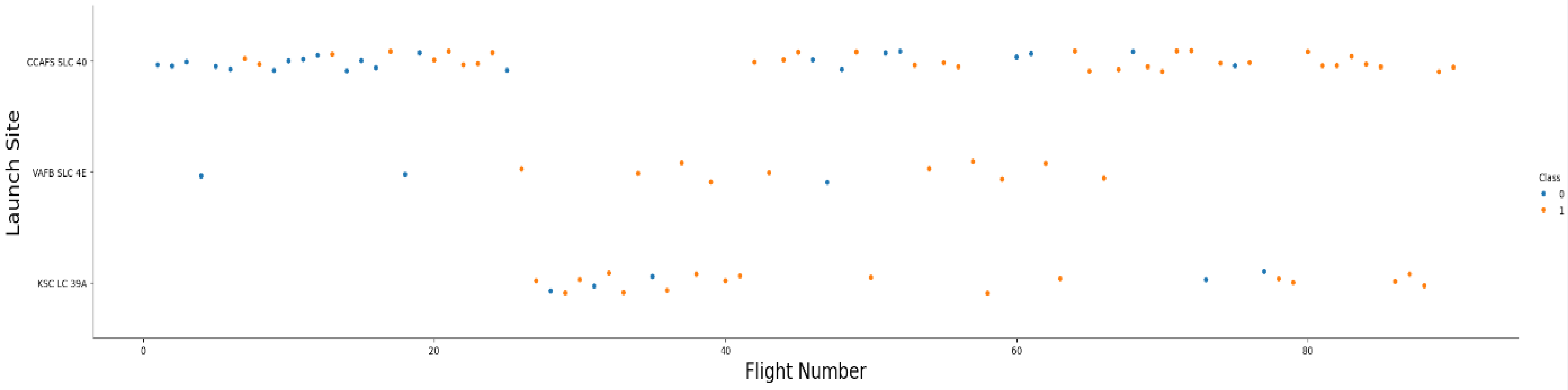
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

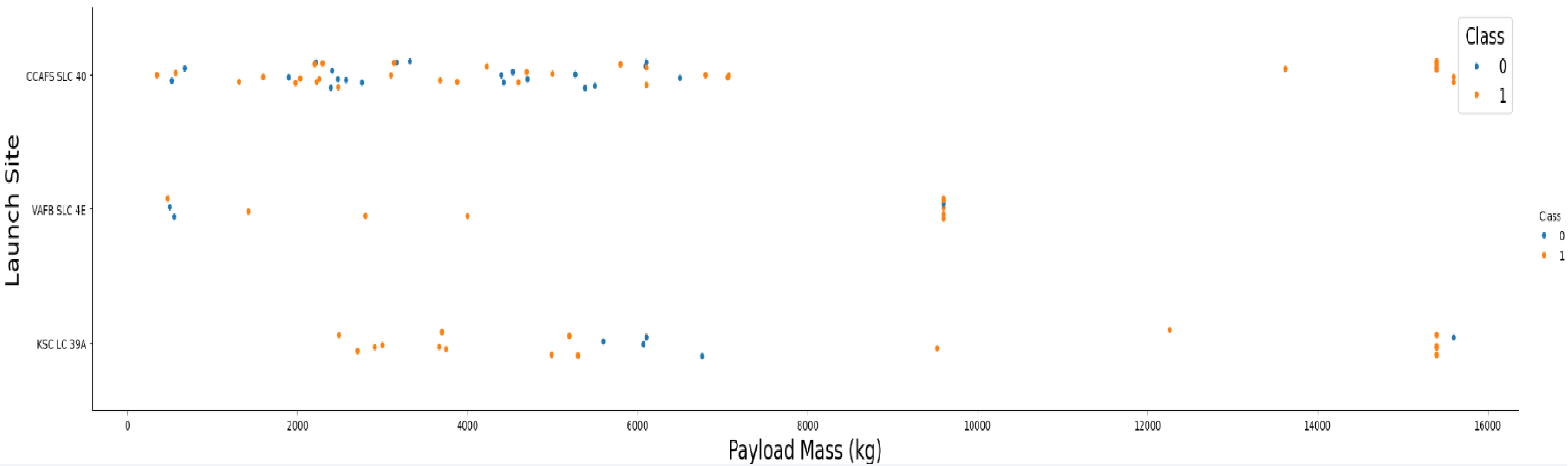
scatter plot of Flight Number vs. Launch Site



In general, success rate has increased with increased flight number, especially at the CCAPS

Payload vs. Launch Site

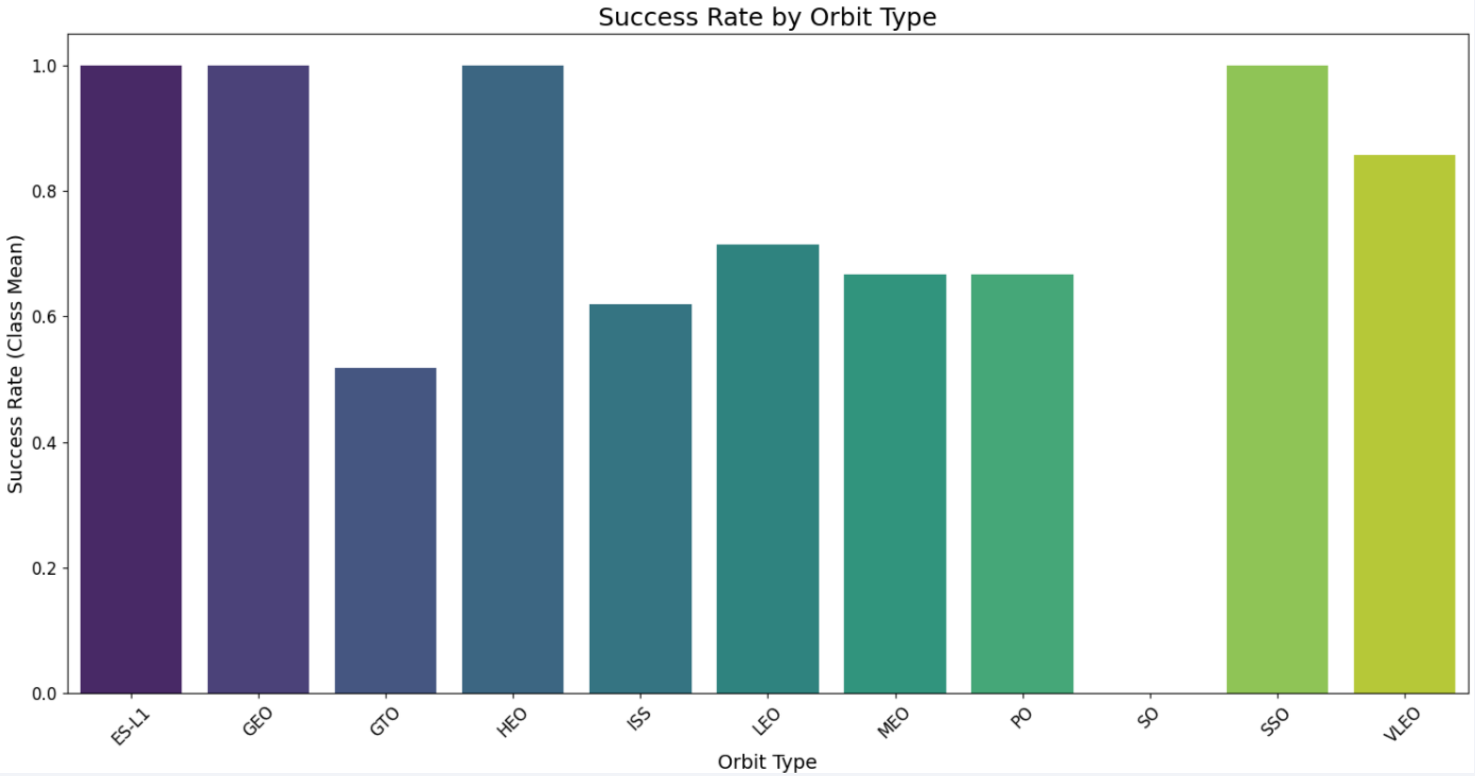
scatter plot of Payload vs. Launch Site



The more massive payloads, the first stage often returns successfully. In VAFB-SLC lanchsite, no rockets launched for heavypaylod mass > 10000.

Success Rate vs. Orbit Type

bar chart for the success rate of each orbit type

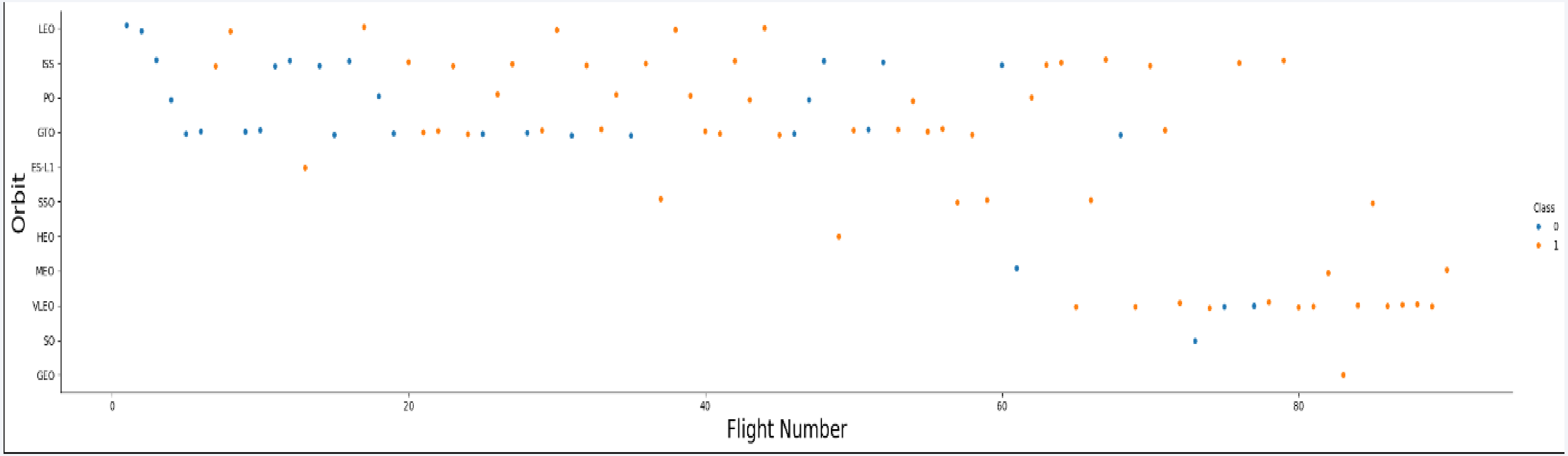


	Orbit	orbit_class
0	ES-L1	1.000000
1	GEO	1.000000
2	GTO	0.518519
3	HEO	1.000000
4	ISS	0.619048
5	LEO	0.714286
6	MEO	0.666667
7	PO	0.666667
8	SO	0.000000
9	SSO	1.000000
10	VLEO	0.857143

Orbits (ES-L1,GEO,HEO,SSO) have the highest success rates.

Flight Number vs. Orbit Type

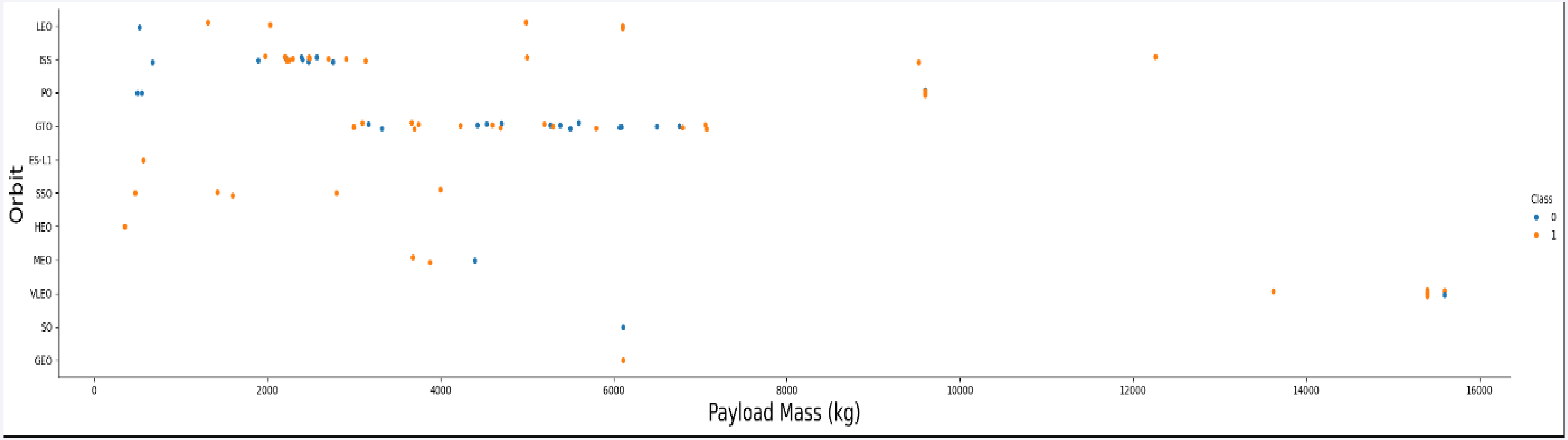
Scatter plot of Flight number vs. Orbit type



Success seems to be related to the number of flights in the LEO orbit. Conversely, in the GTO orbit. there appears to be no relationship between flight number and success.

Payload vs. Orbit Type

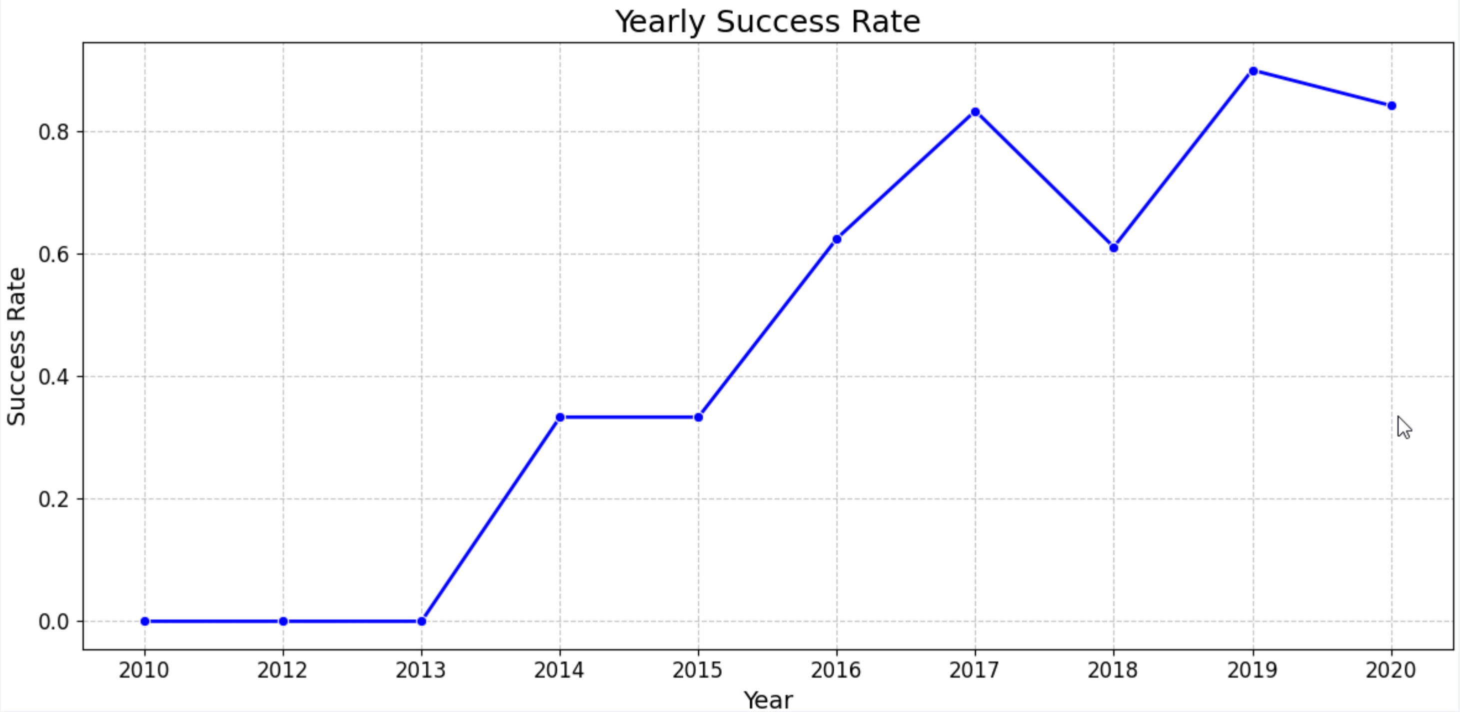
scatter plot of payload vs. orbit type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend

Line chart of yearly average success rate



The overall success rate since 2013 kept increasing till 2020.

All Launch Site Names

Names of the unique launch sites

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

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

DISCINCT query functions like unique(). Returns an array of unique values from the Launch_Site column.

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`

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

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

The above query functions like the following:
records_cca = df[df['Launch_Site'].str.startswith('CCA')].head(5)
print(records_cca)

Total Payload Mass

The total payload carried by boosters from NASA

Total_Payload_Mass
14642

%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1’;

The above query functions like the following:
total_payload_f9 = df[df['Booster_Version'] == 'F9 v1.1']['PAYLOAD_MASS__KG_'].sum()
print("Total Payload Mass for F9 v1.1:", total_payload_f9, "kg")

Average Payload Mass by F9 v1.1

The average payload mass carried by booster version F9 v1.1

AVG_Payload_Mass
2928.4

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version
= 'F9 v1.1’
```

The above query functions like the following:

```
avg_payload_f9 = df[df['Booster_Version'] == 'F9 v1.1']['PAYLOAD_MASS__KG_'].mean()
print("Average Payload Mass for F9 v1.1:", avg_payload_f9, "kg")
```

First Successful Ground Landing Date

The dates of the first successful landing outcome on ground pad

First_Successful_Ground_Pad
2015-12-22

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

The above query functions like the following:

```
first_success_ground_pad = df[df['Landing_Outcome'] == 'Success (ground pad)']['Date'].min()
print("First Successful Ground Pad Landing Date:", first_success_ground_pad)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

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

The above query functions like the following:

```
boosters_success_drone_ship = df[ (df['Landing_Outcome'] == 'Success (drone ship)') &
(df['PAYLOAD_MASS__KG_'] > 4000) &(df['PAYLOAD_MASS__KG_'] < 6000)]['Booster_Version'].unique()
print("Boosters with Success on Drone Ship and Payload Mass 4000-6000:", boosters_success_drone_ship)
```

Total Number of Successful and Failure Mission Outcomes

The total number of successful and failure mission outcomes

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

```
%sql SELECT Mission_Outcome, COUNT(*) AS Count FROM SPACEXTABLE GROUP BY Mission_Outcome
```

The above query functions like the following:

```
mission_outcomes_count = df['Mission_Outcome'].value_counts()
print("Mission Outcomes Count:\n", mission_outcomes_count)
```

Boosters Carried Maximum Payload

The names of the booster which have carried the maximum payload mass

%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

The above query functions like the following:
max_payload_mass = df['PAYLOAD_MASS__KG_'].max()#boosters_max_payload = df[df['PAYLOAD_MASS__KG_'] == max_payload_mass]['Booster_Version'].unique()
print("Boosters with Maximum Payload Mass:", boosters_max_payload)

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

The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

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

The above query functions like the following:

```
df_2015 = df[(df['Landing_Outcome'] == 'Failure (drone ship)') &  (df['Date'].str[0:4] == '2015')]
df_2015['Month'] = df_2015['Date'].str[5:7]  # extract month
print(df_2015[['Month', 'Landing_Outcome', 'Booster_Version', 'Launch_Site']])
```

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

Landing_Outcome	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

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

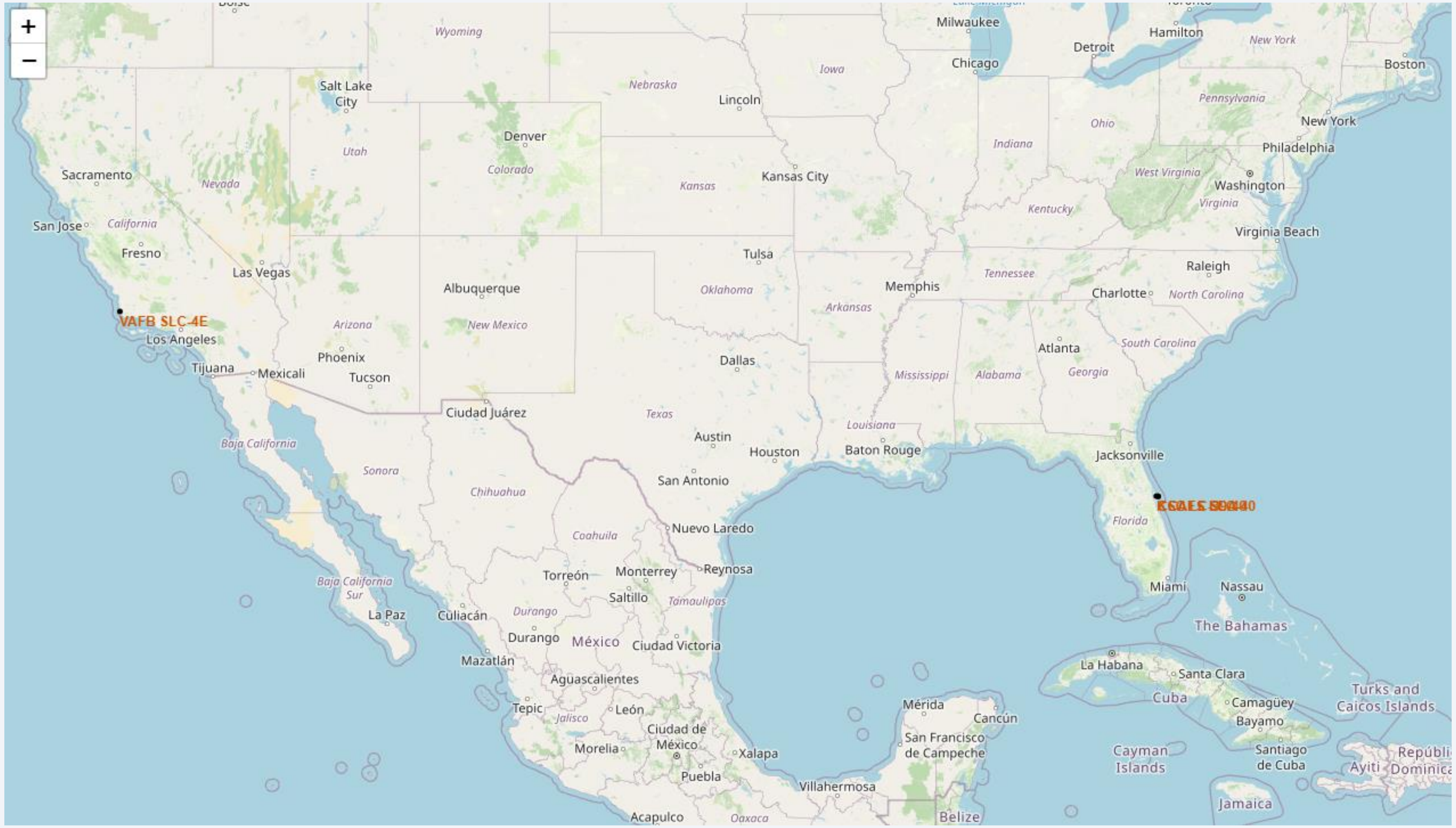
The above query functions like the following:

```
landing_outcomes_ranked = df[(df['Date'] >= '2010-06-04') & (df['Date'] <= '2017-03-20')]['Landing_Outcome'].value_counts().sort_values(ascending=False)
print("Ranked Landing Outcomes:\n", landing_outcomes_ranked)
```

Section 3

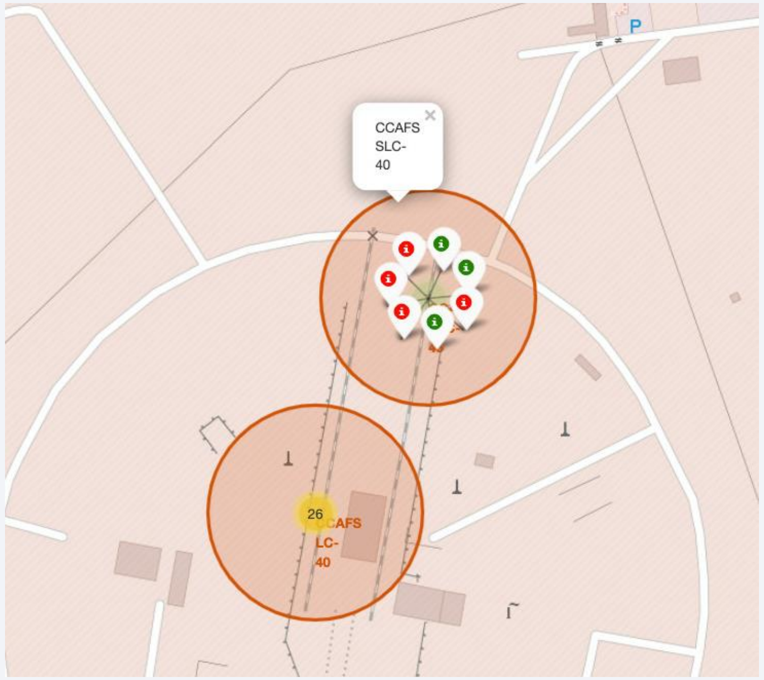
Launch Sites Proximities Analysis

All launch sites' location markers on a global map



Color-labeled launch outcomes on the map

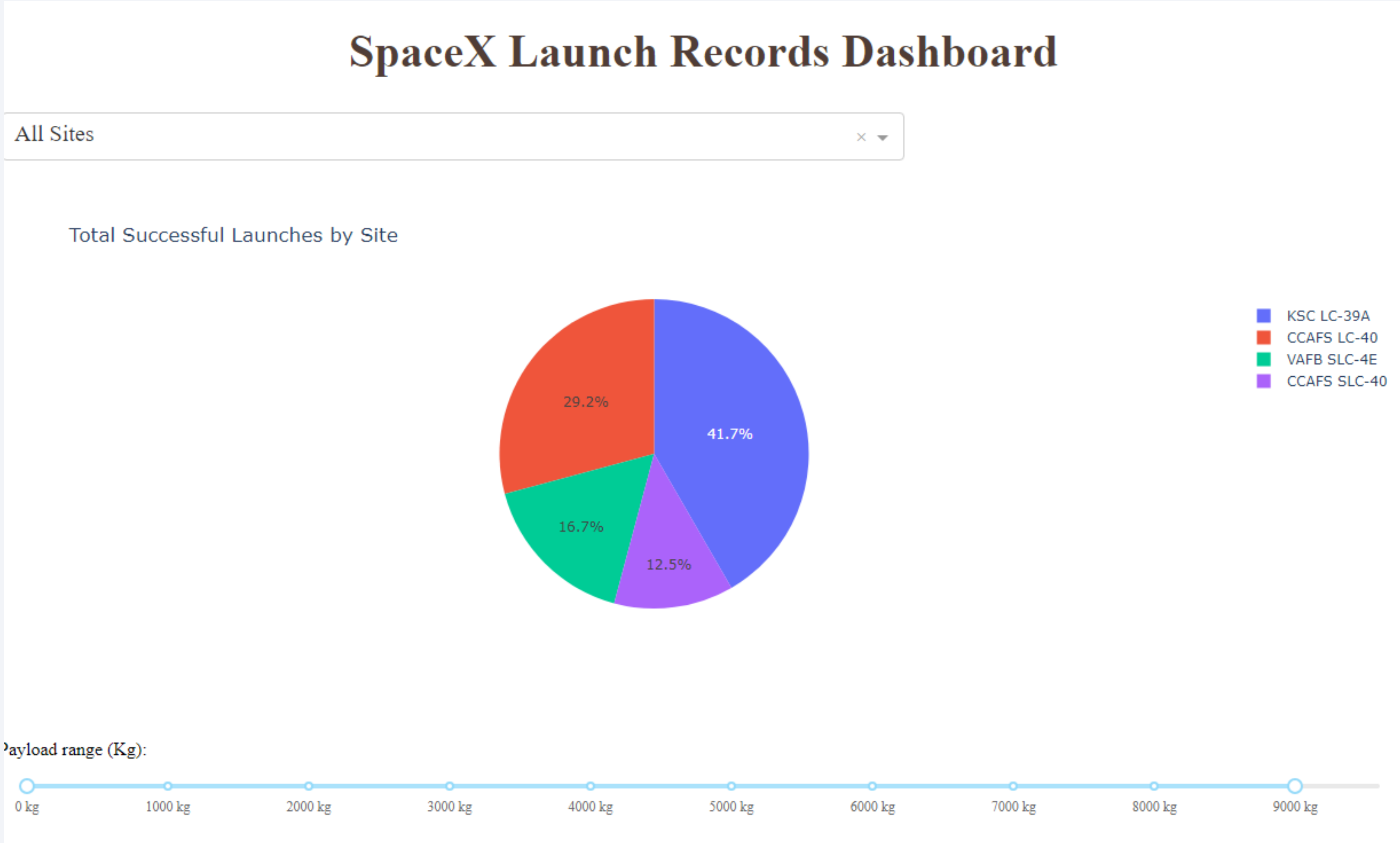
Selected launch site to its proximities



Section 4

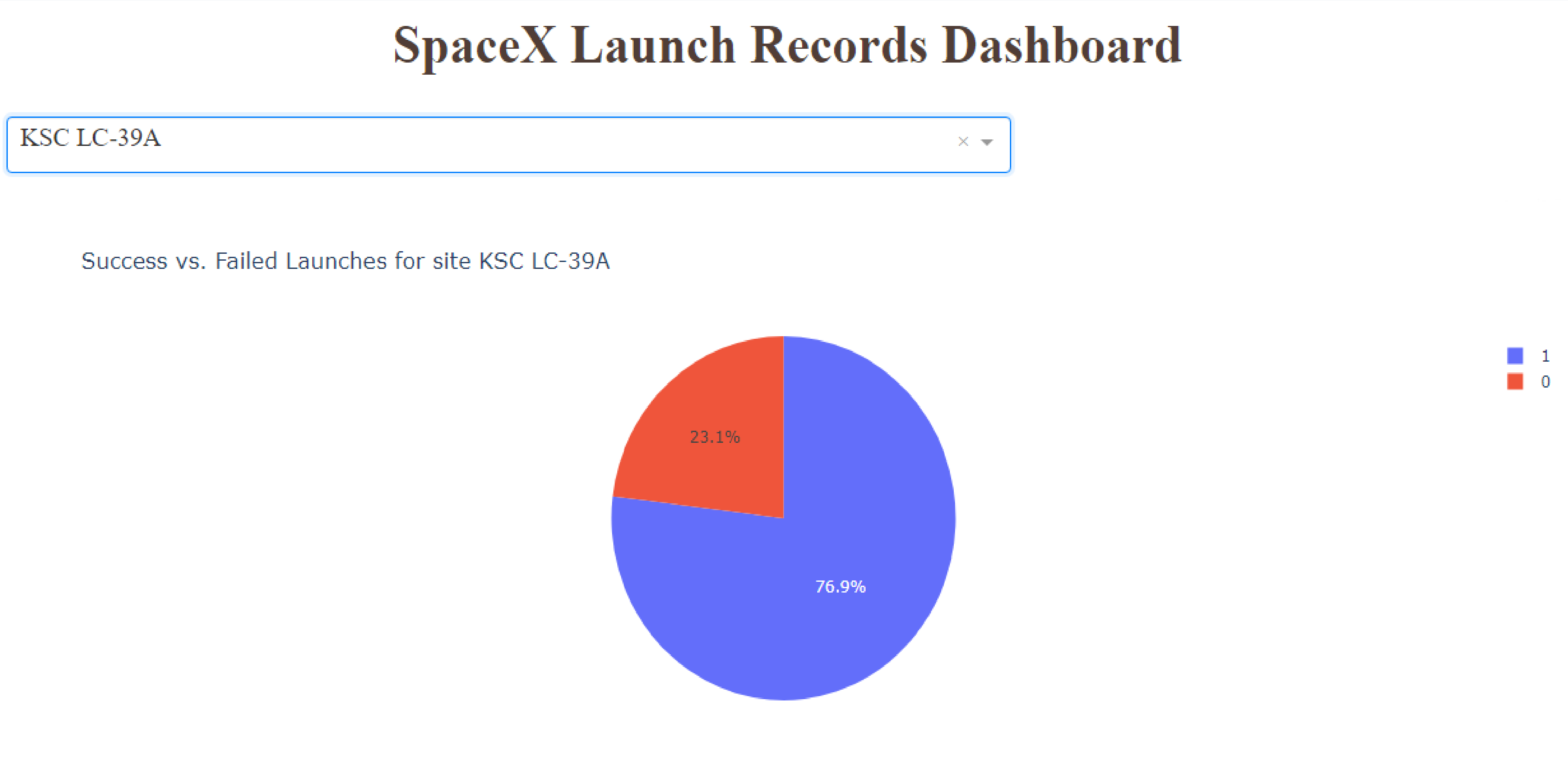
Build a Dashboard with Plotly Dash

Launch success count for all sites



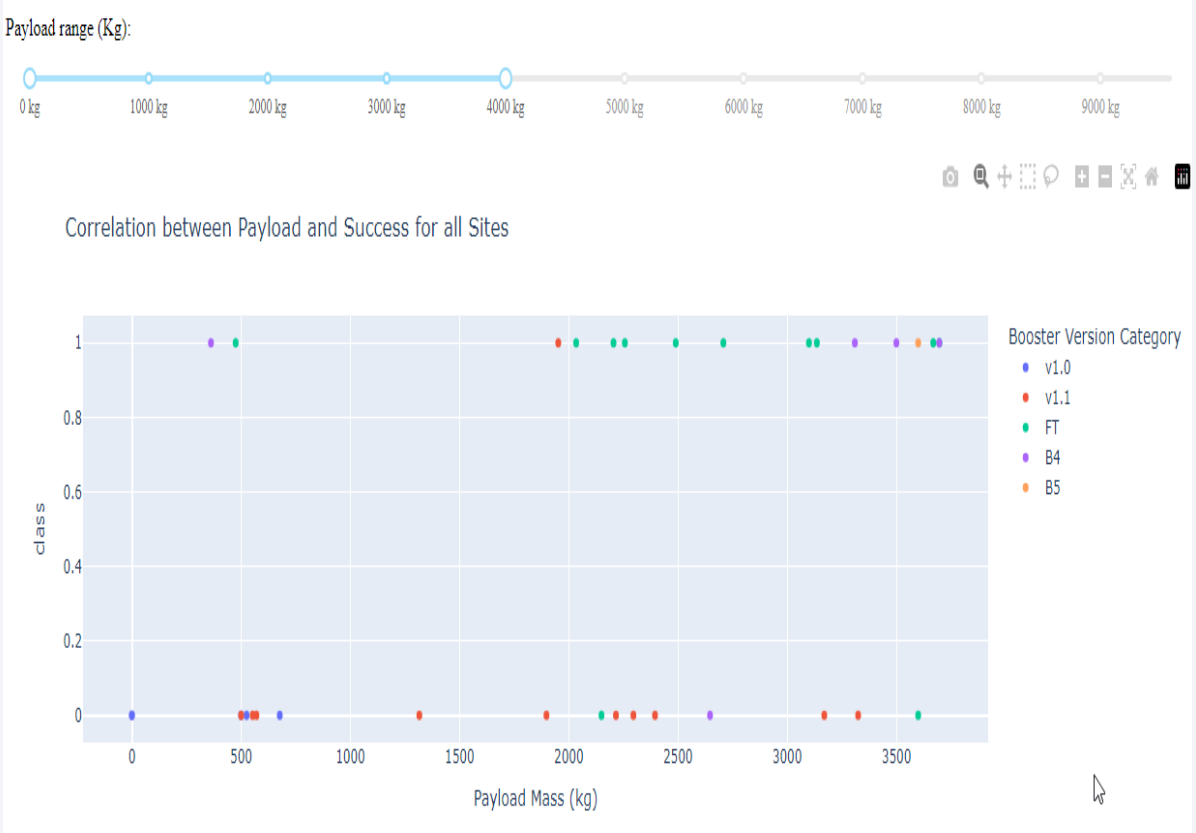
KSC LC has the highest total successful launches and CFAFS SLC has the lowest.

Launch site with highest launch success ratio



KSC LC has the highest launch success ration with 76.9%.

Payload vs. Launch Outcome scatter plot for all sites

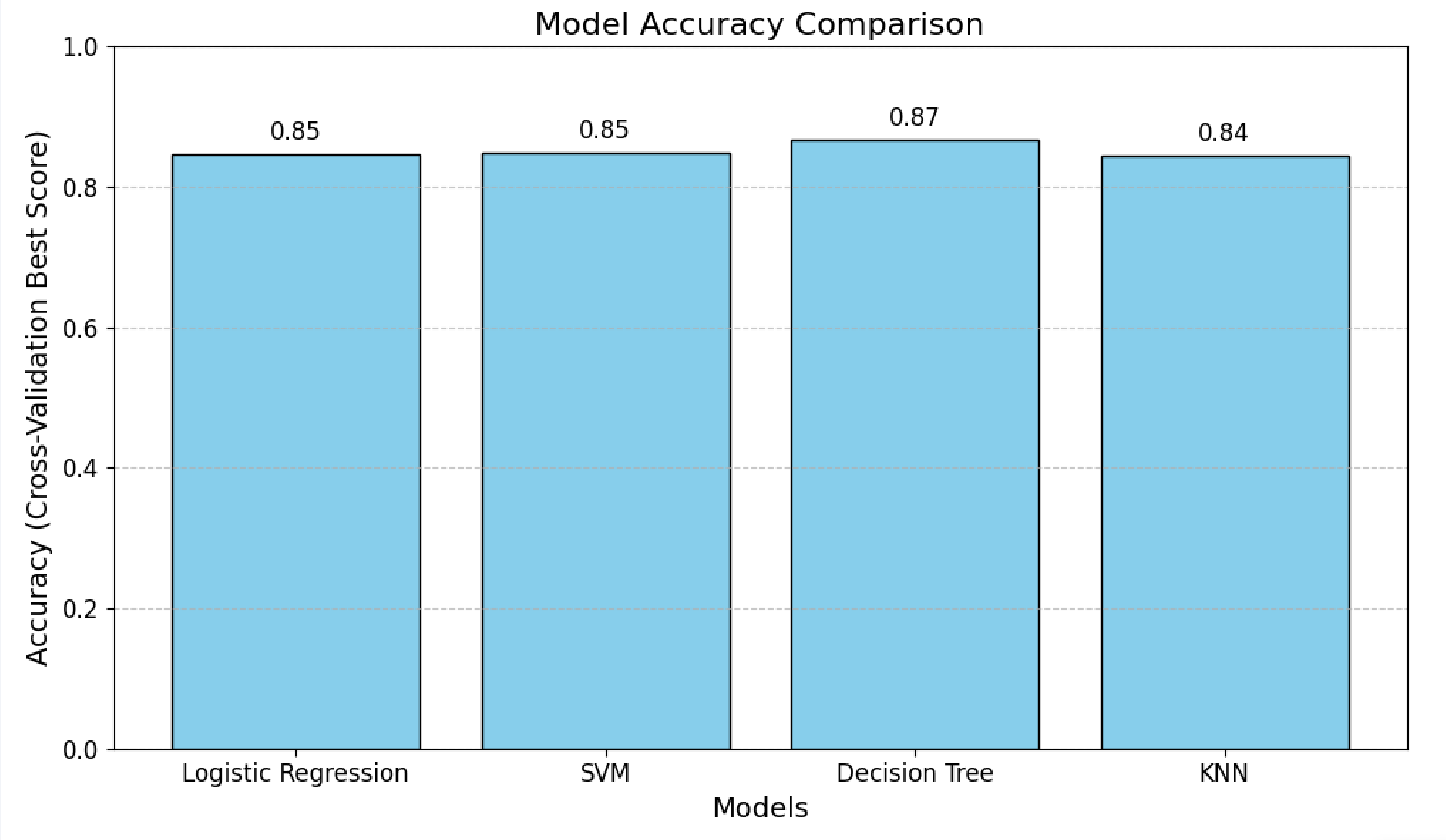


Section 5

Predictive Analysis (Classification)

Classification Accuracy

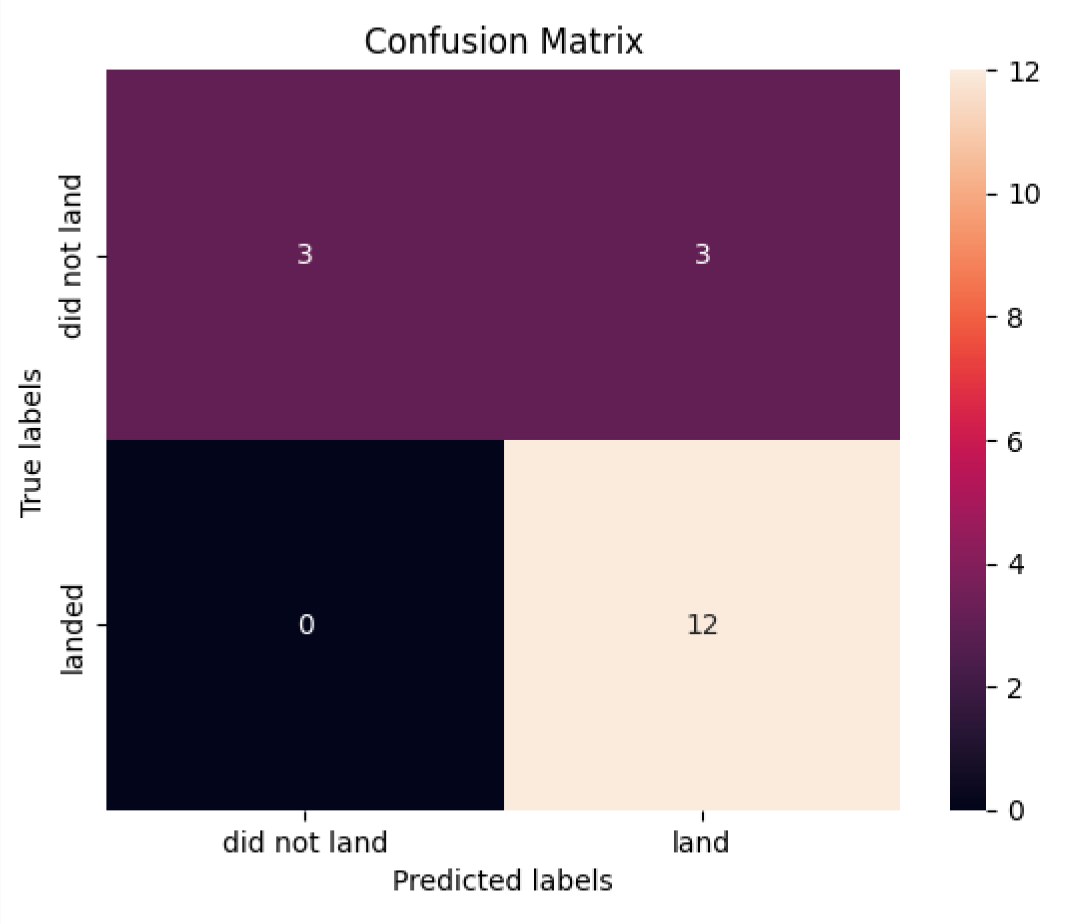
Model accuracy for all built classification models



Decision Tree Model has the highest accuracy

Confusion Matrix

The confusion matrix of the best performing model



True Positive: 12
False Positive: 3
Accuracy on the test data: 0.8888888888888888

Conclusions

Based **on** the comprehensive data analysis and modeling:

- Landing Predictions: Decision tree model likely to yield the highest accuracy in predicting landing outcomes, given their interpretability and performance in similar scenarios.
- Insights: High success rates are observed for Polar, LEO and ISS with heavy payloads. The success rate since 2013 kept increasing till 2020.
- Dashboard Insights: Users gain actionable insights into key metrics, aiding decision-making for future missions.

Conclusions: This approach ensures a robust framework for understanding SpaceX's operations and evaluating the feasibility of first-stage landings, paving the way for cost optimization in aerospace missions.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!