IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pablo Sánchez Arribas
10/15/2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of Methodologies

- This Machine Learning project aims to understand the main features behind SpaceX's successful launches.
- Followed methodology in order to achieve higher success rate in the initial design phase:
  - **Data collection** from calls to SpaceX's API, as well as web scrapping from Wikipedia.
  - **Data wrangling** into a desirable format and uploaded to database
  - **Exploratory data analysis** has allowed to understand the behavior of main features
    - Interactive graphs and maps helped gaining better insights
  - **Feature engineering & LM models** were used to predict launch success

Summary of Results

- Future rockets will be designed for optimum payload range (2.000 Kg – 6.000 Kg), to achieve maximum launch success probability.
- Selected launch site is KSC LC-39A, where maximum launch success has been achieved
- Early missions won't carry useful payload to avoid incurring into penalties with clients in case of unsuccessful launches.
- Rocket will be designed for optimal success, for which the following will be considered:
  - Booster design will resemble the latest versions examined (FT, B4, B5), which provide higher success.
  - Mission requirements will be planned to satisfy orbits with perfect success rate: ES-L1, GEO, HEO, SSO

Introduction

Project Background and Context:

- We aim to predict the successful landing of the Falcon 9 first stage. SpaceX advertises rocket launches at a significantly lower cost compared to other providers, largely due to their ability to reuse the first stage of the rocket.
- By accurately predicting landing success, we can estimate launch costs and provide valuable insights to study the viability of a new player in the market: SpaceY.

Main problems analyzed:

- What factors influence the successful landing of the Falcon 9 first stage?
- How can we accurately predict the landing outcome using machine learning models?
- Which machine learning model performs best in predicting the landing success?

Section 1

Methodology

Methodology 1/2

Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX's API, as well as web scrapping the launches from Wikipedia
- Perform data wrangling
  - Data was processed in order to filter by Falcon 9 launches and include mean PayloadMass values in missing registers
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Visualized launch success rates, payloads, and launch sites using Matplotlib and Seaborn.
  - Executed SQL queries to derive insights and answer specific questions regarding the dataset.

Methodology 2/2

- Perform interactive visual analytics using Folium and Plotly Dash
  - Used Folium to create interactive maps displaying launch sites and outcomes.
  - Developed a Plotly Dash application with interactive components like dropdowns and sliders to analyze launch success rates and payload ranges.
- Perform predictive analysis using classification models
  - Built and evaluated various classification models including Logistic Regression, SVM, KNN, and Decision Trees.
  - Employed GridSearchCV for hyperparameter tuning.
  - Evaluated models based on accuracy, and identified the best performing model for predicting landing success.

Github URL:

https://github.com/PabloSArribas/IBM-Data-Science/tree/d5c5c85ae7d3f52042eec21c2f8bea9efdc99dc0/10.%20Applied%20Data%20Science%20Capstone

Data Collection

1. SpaceX API request

Initiate API request → Fetch Launch Data → Store Data Locally

2. Web Scrapping Wikipedia

Extract HTML Table → Parse with BeautifulSoup → Convert it to DataFrame

3. Data Integration

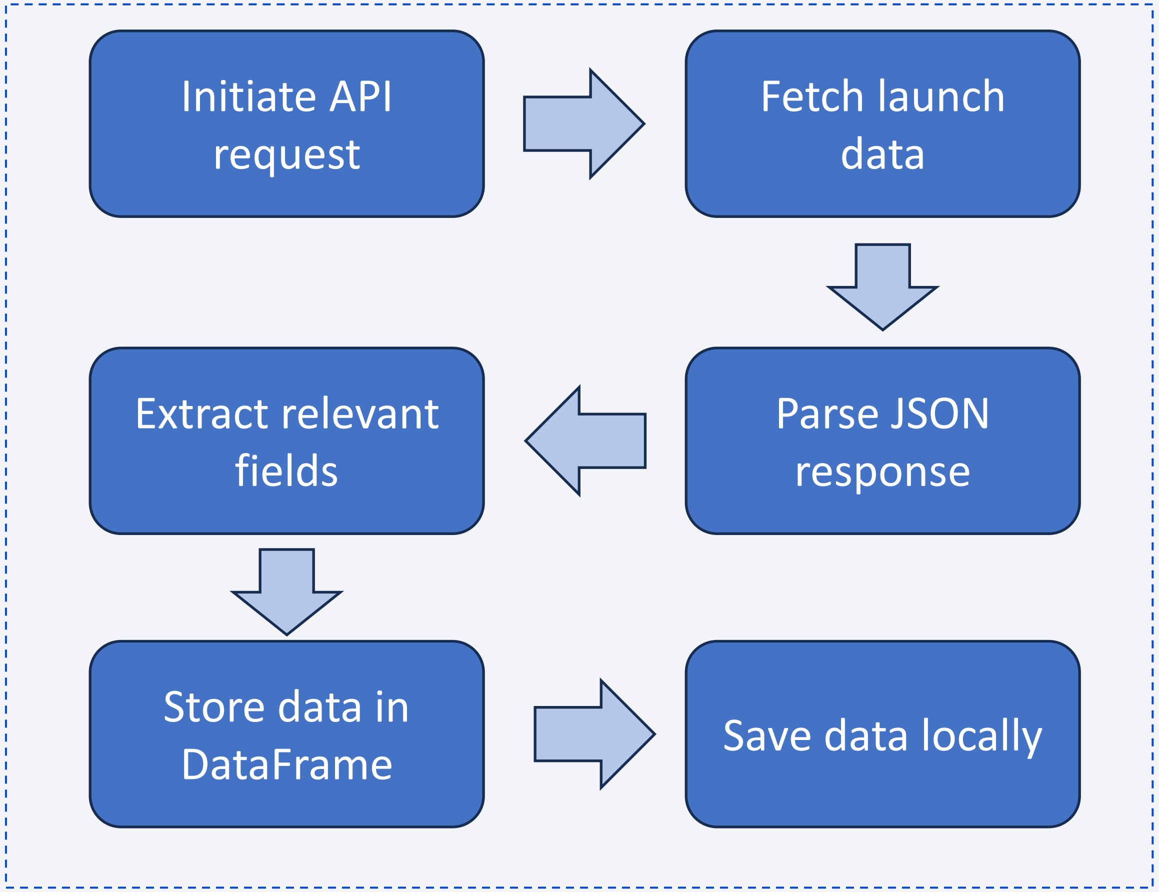
SpaceX API Data & Wikipedia Data → Merge Datasets → Final Integrated Data

Data Collection – SpaceX API

- **Initiate API request:**
  - Use Python's `requests` library to connect to the SpaceX API.
  - Endpoint: *https://api.spacexdata.com/v4/launches*

- **Parse API Response**
  - Convert API response from JSON to a Pandas' dataframe.
  - Extract relevant fields: launch date, launch site, payload mass, rocket type, outcome.

- **Store Data Locally**
  - Store the DataFrame locally in a .csv for further processing.

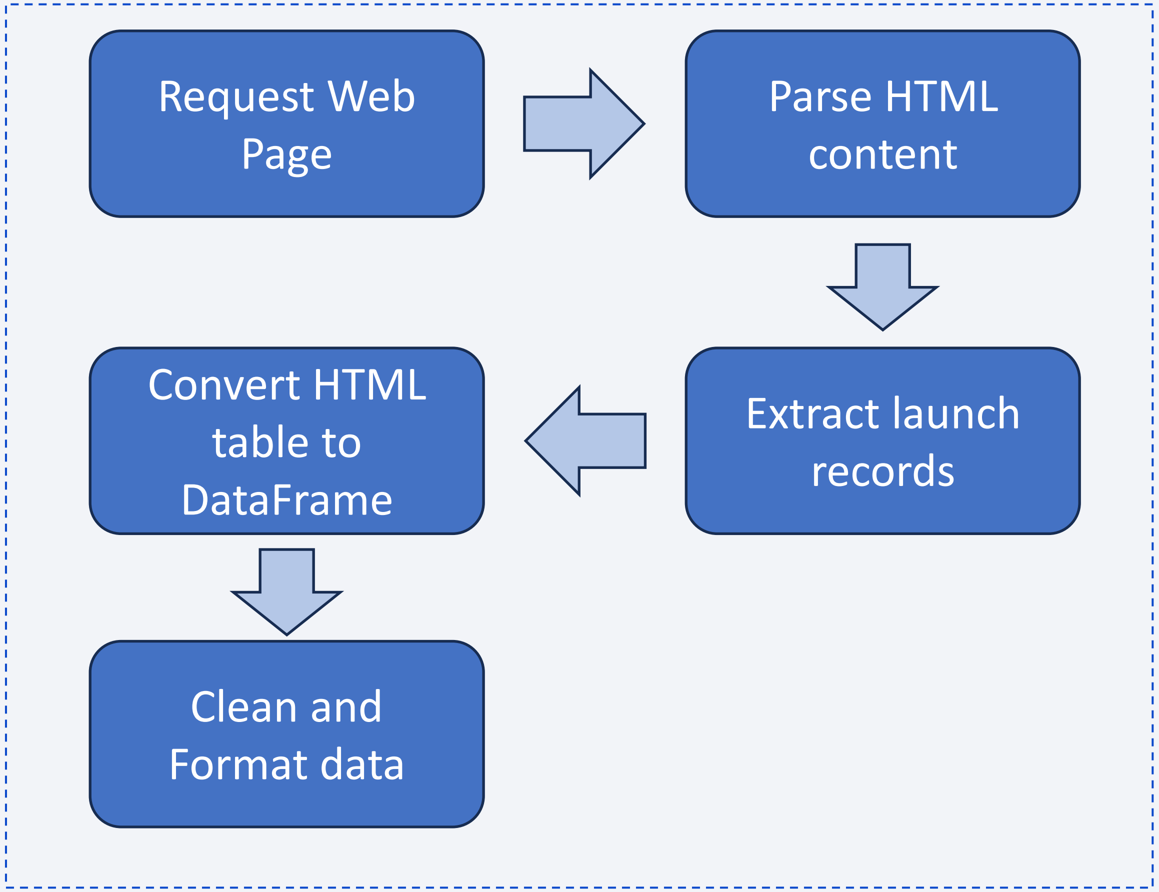


GitHub URL:

https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.1%20-%20jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

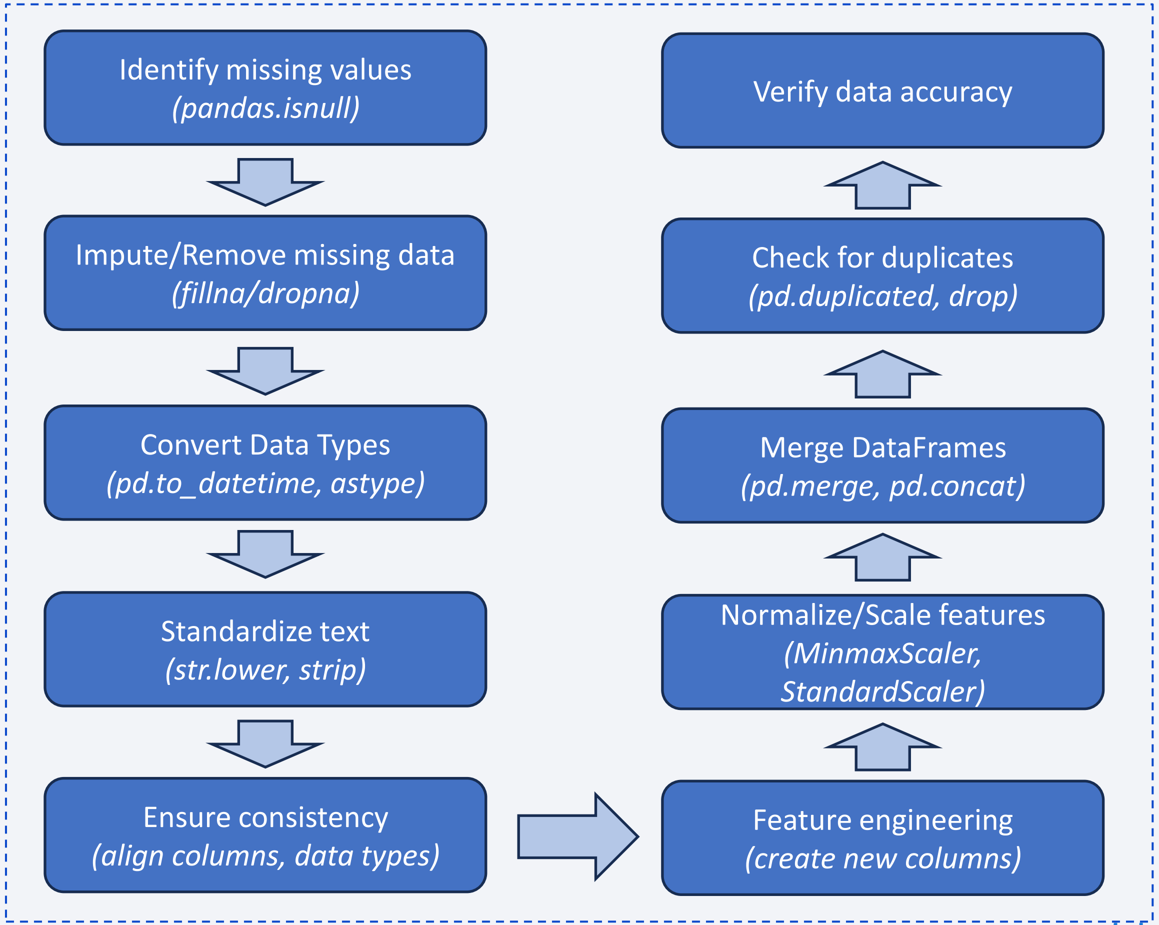
- **Initiate Web Scraping**
  - Use Python's `requests` library to fetch the HTML content of the Wikipedia page.
  - Target URL: `https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches`
- **Parse HTML Content**
  - Use `BeautifulSoup` to parse the HTML content.
  - Extract the HTML table containing Falcon 9 launch records.
- **Convert to DataFrame**
  - Convert the extracted HTML table into a pandas DataFrame.
  - Clean and format the DataFrame, ensuring data consistency.



GitHub URL:
https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.2%20-%20jupyter-labs-webscraping.ipynb

Data Wrangling

- **Data integration**
  - Merge datasets collected from different sources (API, web scraping) into a single cohesive dataset.
  - Ensure consistent column names and data formats across datasets.
- **Data validation**
  - Check for duplicate records and remove them.
  - Verify the accuracy and consistency of data entries.



GitHub URL:

https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.3%20-%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- **Overview**

Exploratory Data Analysis (EDA) involves visually exploring and summarizing the main characteristics of a dataset. The goal is to understand the data's distribution, identify patterns, and uncover relationships between variables.

- **Charts**

1. Histograms

- Used to visualize the distribution of numerical variables like payload mass. Helps understanding the spread and central tendency of the data, identifying outliers, and assessing data skewness.

2. Bar charts

- Used to compare categorical variables like rocket types, to provide a clear comparison of frequencies or proportions within categorical data, highlighting patterns or trends.

3. Line charts

- Used to track trends over time. Like the success rate along time.

4. Scatter plots

- Used to explore relationships between two numerical variables, as it helps identifying correlations or dependencies.

5. Heatmaps

- Used to visualize correlation matrices between numerical variables to identify correlations.

6. Box plots

- Used to display distribution of numerical data through quartiles, helping viasualize the spread or skewness, outliers and distributions.

GitHub URL:
https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.5%20-%20edadataviz.ipynb

EDA with SQL

- **Aggregate queries**
  - Calculated total number of launches.
  - Counted successful and failed launches.
  - Calculated success rates by launch site and rocket type.
- **Join queries**
  - Joined tables to link launch records with additional data (e.g., rocket details).
  - Combined datasets for comprehensive analysis.
- **Filtering queries**
  - Filtered data to focus on specific launch outcomes (success/failure).
  - Applied conditions to extract launches based on criteria like launch date or rocket configuration.
- **Sorting queries**
  - Sorted data to identify trends or outliers.
  - Ordered launches by date or success rate for analysis.
- **Subqueries**
  - Nested queries to calculate derived metrics (e.g., average payload mass per launch site).
  - Subqueries used to perform detailed analysis within larger datasets.

GitHub URL:

https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.4%20-%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map Objects

- **Markers**
  - Placed markers to indicate launch sites on the map.
  - Each marker represents a specific geographical location where SpaceX launches have occurred.
- **Circles**
  - Added circles around launch sites to visually represent proximity zones.
  - Circles help visualize the areas around launch sites that might influence operational decisions.
- **Lines**
  - Drew lines to connect launch sites with their proximities or other relevant locations.
  - Lines provide spatial context and connections between different points of interest related to launches.

Reasons for adding Objects

- **Markers**
  - To pinpoint exact launch locations for spatial reference.
  - Helps users identify where SpaceX has conducted launches geographically.
    - Created and added a *folium.Circle* and *folium.Marker* for each launch site
    - For each launch result in *spacex_df* data frame, added a folium.Marker to marker_cluster
- **Circles**
  - Illustrates the potential impact zones around launch sites.
  - Provides a visual representation of safety perimeters or operational boundaries.
- **Lines**
  - Shows connections or relationships between launch sites and relevant features.
  - Enhances understanding of spatial relationships and dependencies.
    - Created a line with distance to a closest city to VA
    - Created a line with distance to a closest highway to VA

GitHub URL:

https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.6%20-%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Plots/Graphs

- **Success Pie Chart:**
  - Displays the distribution of successful and failed launches.
  - Helps visualize the overall success rate and performance trends.
    - Provides metrics at a glance for stakeholders
- **Success-Payload Scatter Plot:**
  - Shows the relationship between payload mass and launch success.
  - Allows users to explore how payload
    - Helps identify correlations between payload characteristics and launch outcomes.

Interactions

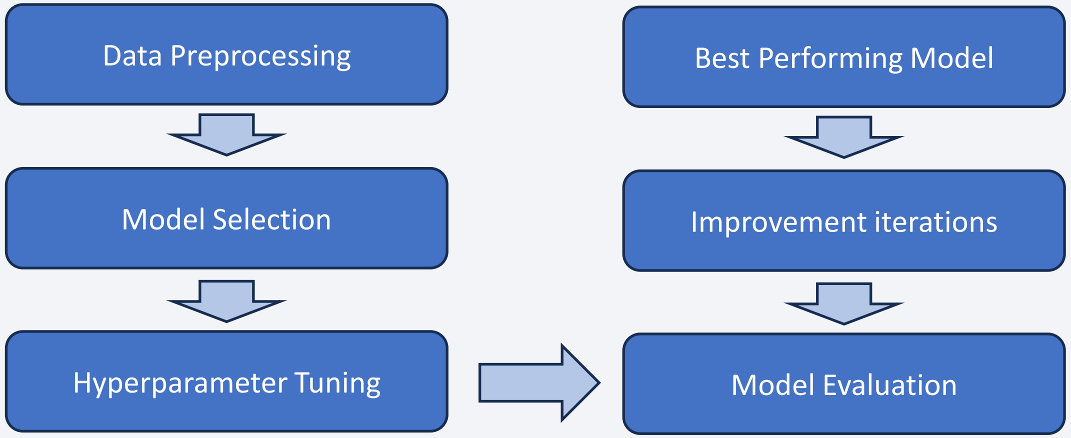
- **Launch Site Dropdown:**
  - Enables users to select specific launch sites for analysis.
  - Facilitates filtering and focused exploration based on geographical locations.
- **Range Slider for Payload:**
  - Allows users to adjust payload mass ranges dynamically.
  - Offers flexibility in examining launch success concerning payload mass variations.
    - Offers interactive exploration of how payload mass affects mission success.
    - Enables detailed analysis and insights into payload-related performance factors.

GitHub URL:

https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.7%20-%20spacex-dash-app.py

Predictive Analysis (Classification)

1. **Data preprocessing:**
   - Standardized features to ensure all variables contribute equally.
   - Split data into training and test sets for model validation.
2. **Model selection:**
   - Explored multiple classification algorithms: SVM, Decision Trees, and K-Nearest Neighbors (KNN).
   - Chose algorithms suitable for binary classification tasks based on project requirements.
3. **Hyperparameter tunning:**
   - Used *GridSearchCV* to systematically search for optimal hyperparameters.
   - Tuned parameters such as *C* (SVM), *max_depth* (Decision Trees), and *n_neighbors* (KNN).
4. **Model evaluation:**
   - Evaluated models using cross-validation techniques to ensure robustness and generalizability.
   - Utilized metrics like accuracy, precision, recall, and F1-score to assess model performance.
5. **Improvement iterations:**
   - Iteratively adjusted models based on insights from validation results.
   - Fine-tuned hyperparameters to maximize predictive accuracy and reliability.
6. **Selection of Best Performing Model:**
   - Identified the model with the highest accuracy on the test set as the best performer.
   - Considered both training and test set performance to avoid overfitting and ensure real-world applicability.

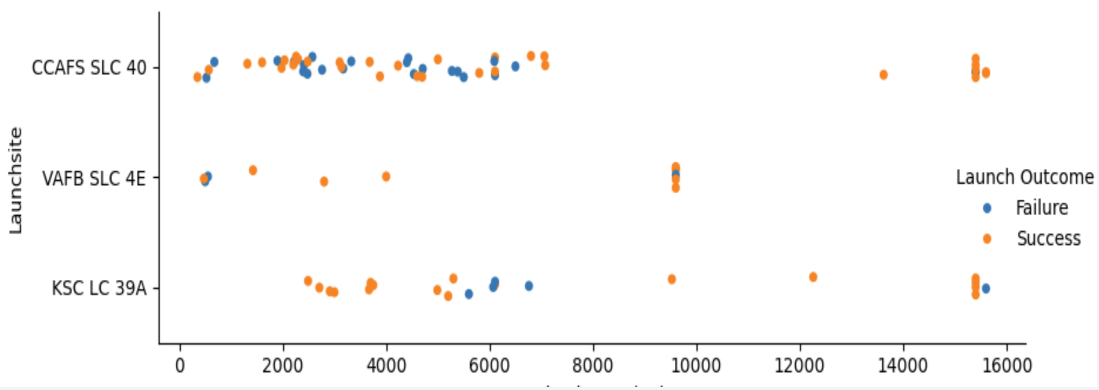


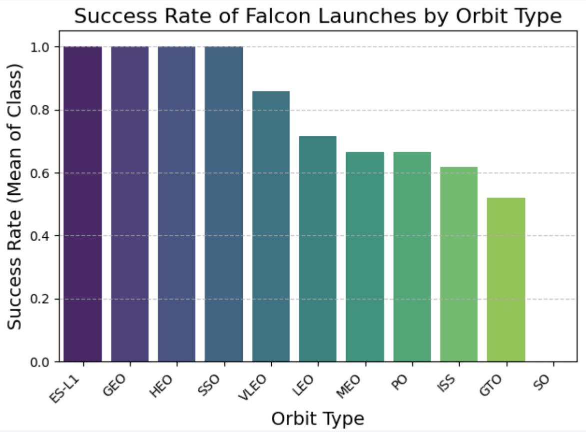
GitHub URL:

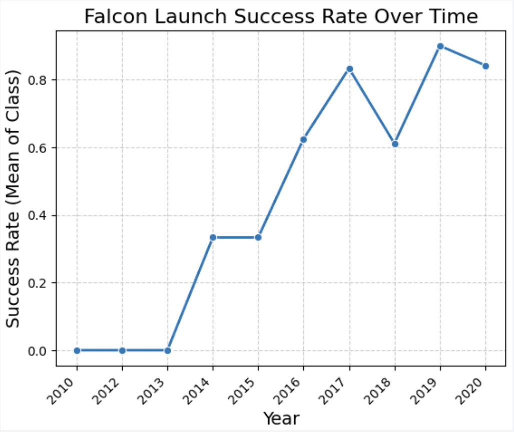
https://github.com/PabloSArribas/IBM-Data-Science/blob/main/10.%20Applied%20Data%20Science%20Capstone/Module%201.8%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

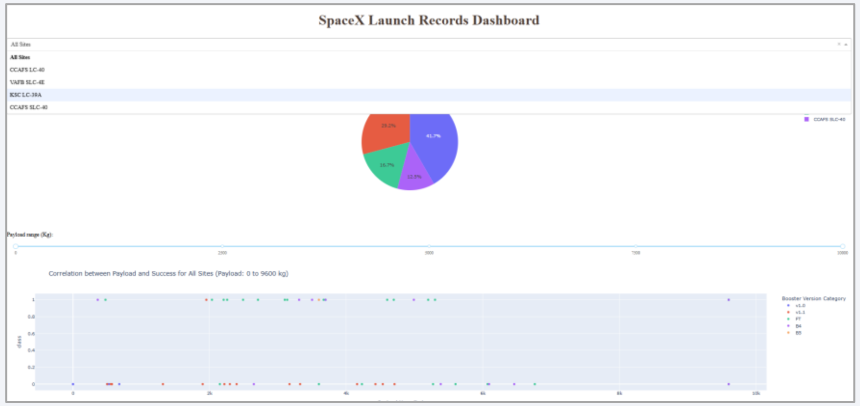
- Exploratory data analysis results



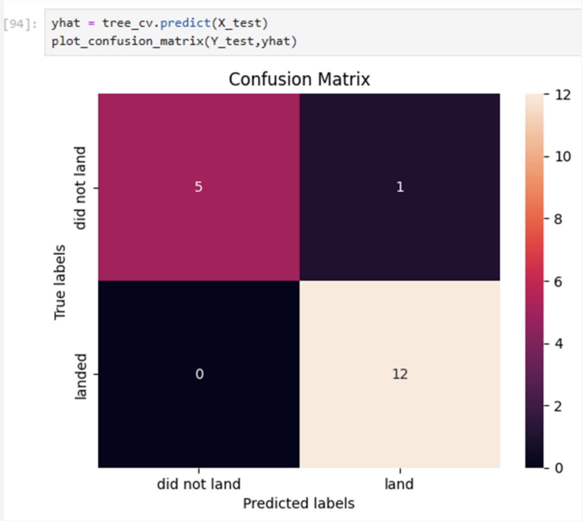




- Interactive analytics demo in screenshots



- Predictive analysis results

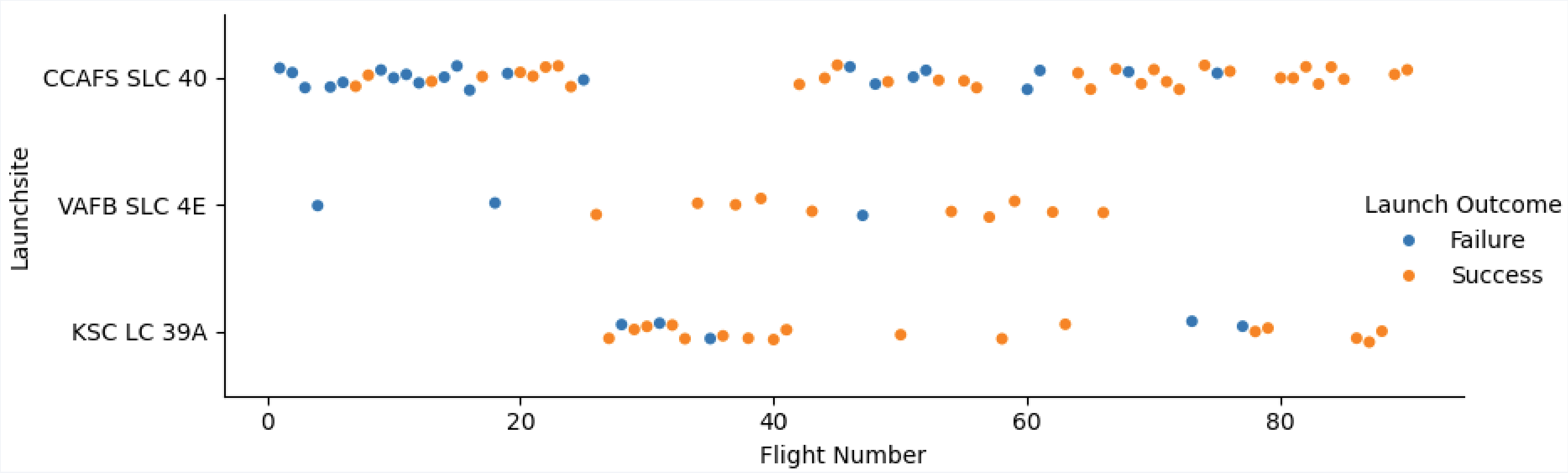


Section 2

Insights drawn from EDA

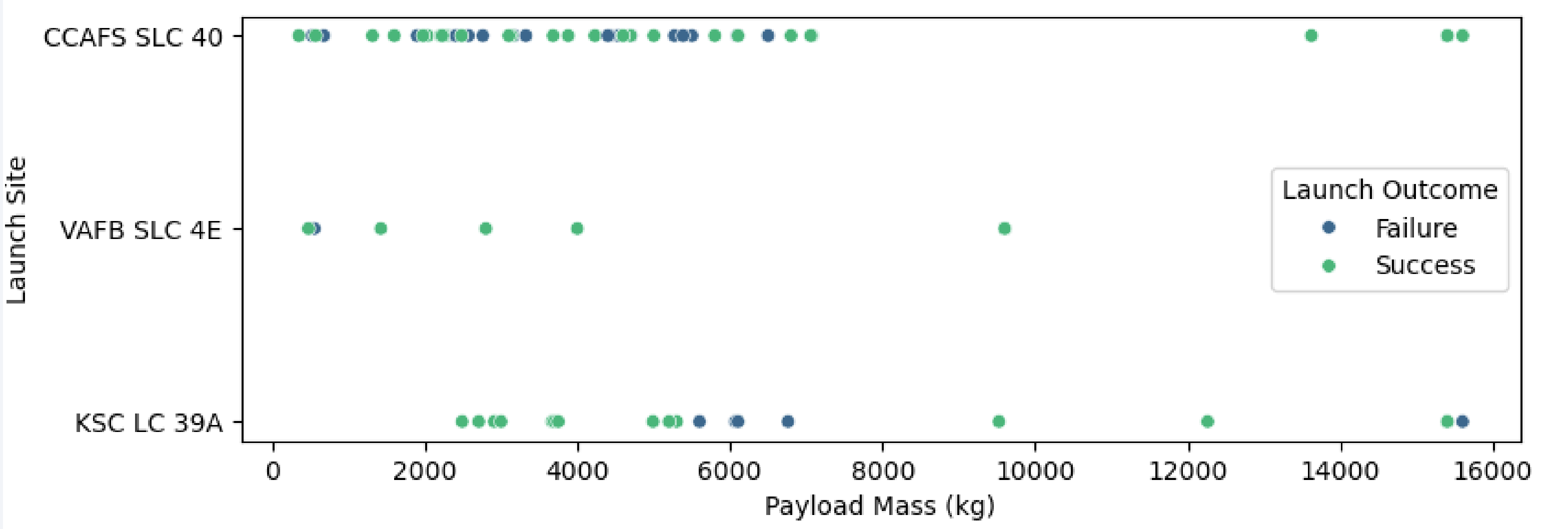
Flight Number vs. Launch Site

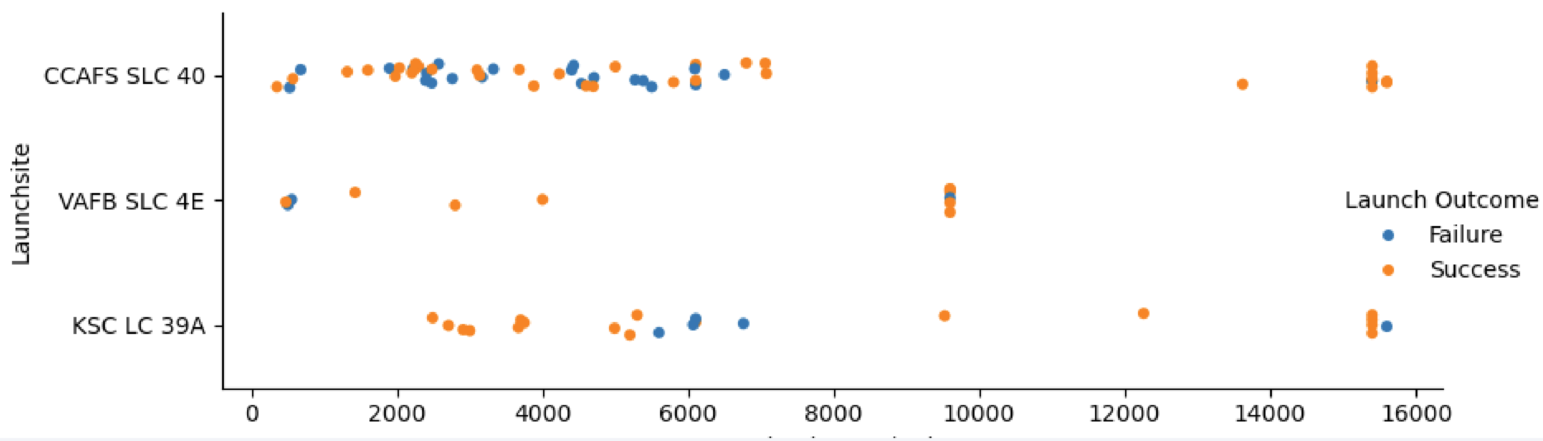
- **Variable activity across launch sites**:
  - Initial activity was mostly carried at CCAFS SLC-40, except for 2 failed launches at VAFB.
  - For flights ~25-45 activity was primarily held at KSC, and for flights ~45 onwards activity was mostly held at all sites.
  - Activity in VAFB ceased from flight ~68.
- **Increased success along flight numbers**, as more experience was gained in time.



Payload vs. Launch Site

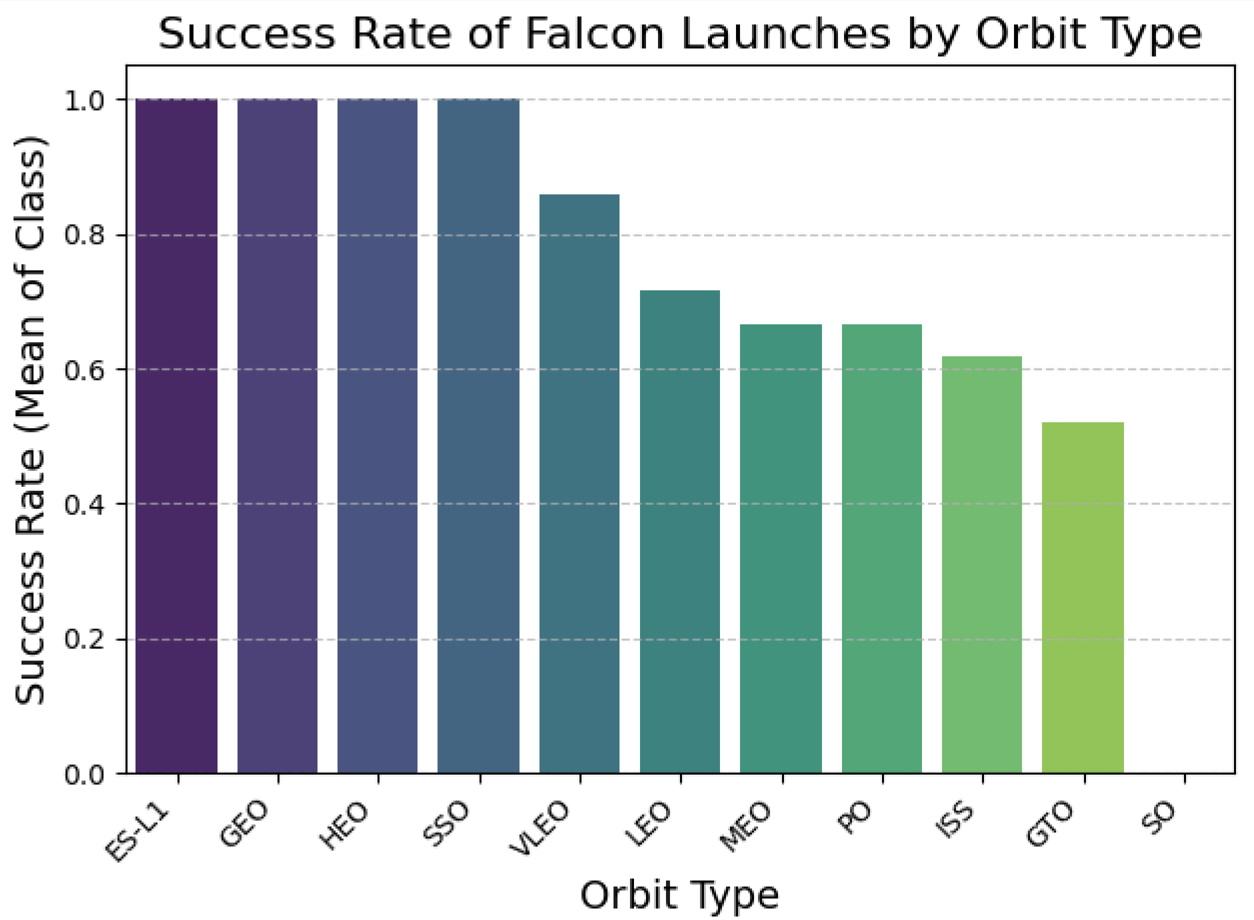
- **Payload distribution**:
  - Most launches have payloads in between the following two ranges: ~500-7.500Kg and ~15.000-16.000Kg
  - VAFB SLC 4E site only held launches which payload doesn't exceed 10.000Kg.
- **Success distribution:**
  - Launch success is uniformly distributed along payloads, except for the heaviest segment, suggesting they were launched further in time after greater experience was gained.





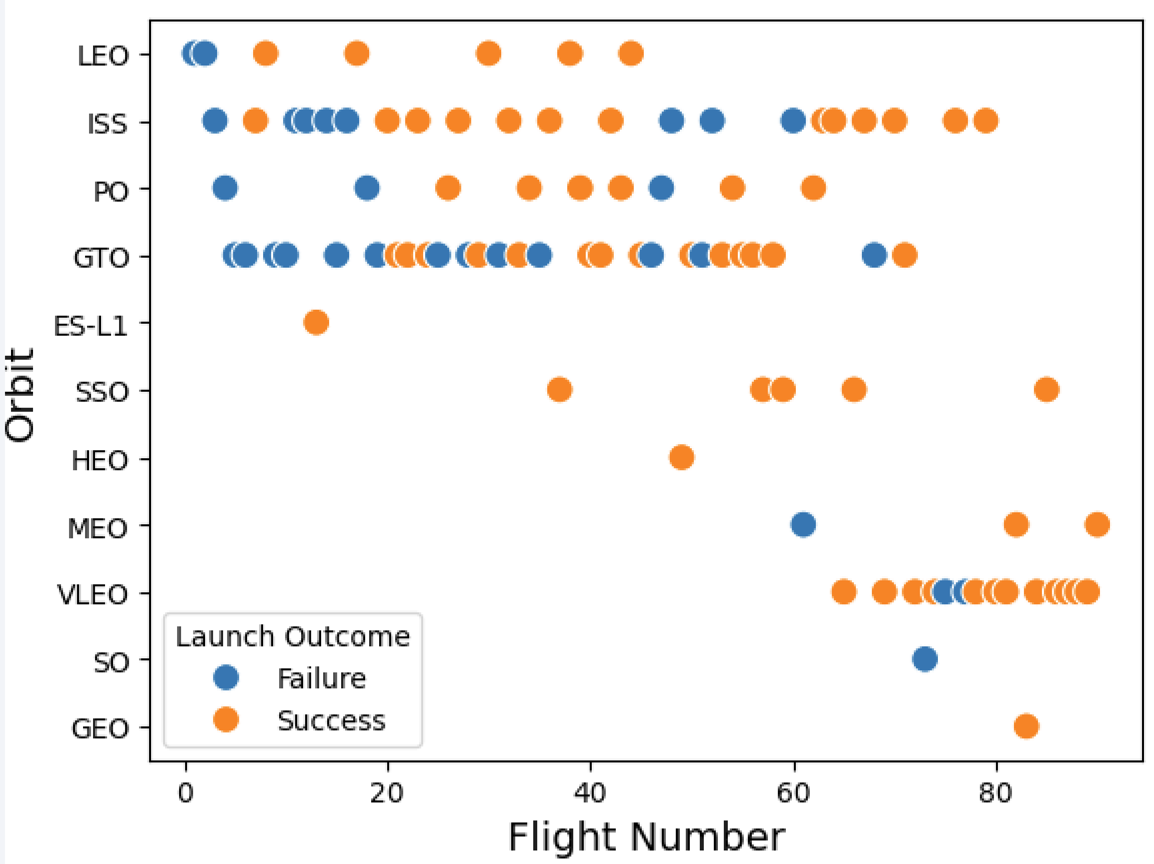
Success Rate vs. Orbit Type

- **Success distribution over Orbit**:
  - VLEO, ES-L1, GEO, HEO, and SSO orbits show 100% success rate
  - The rest of the orbits show a success rate between ~50%-85%
  - SO orbit stands for it 0% success rate, although only 1 mission was launched.



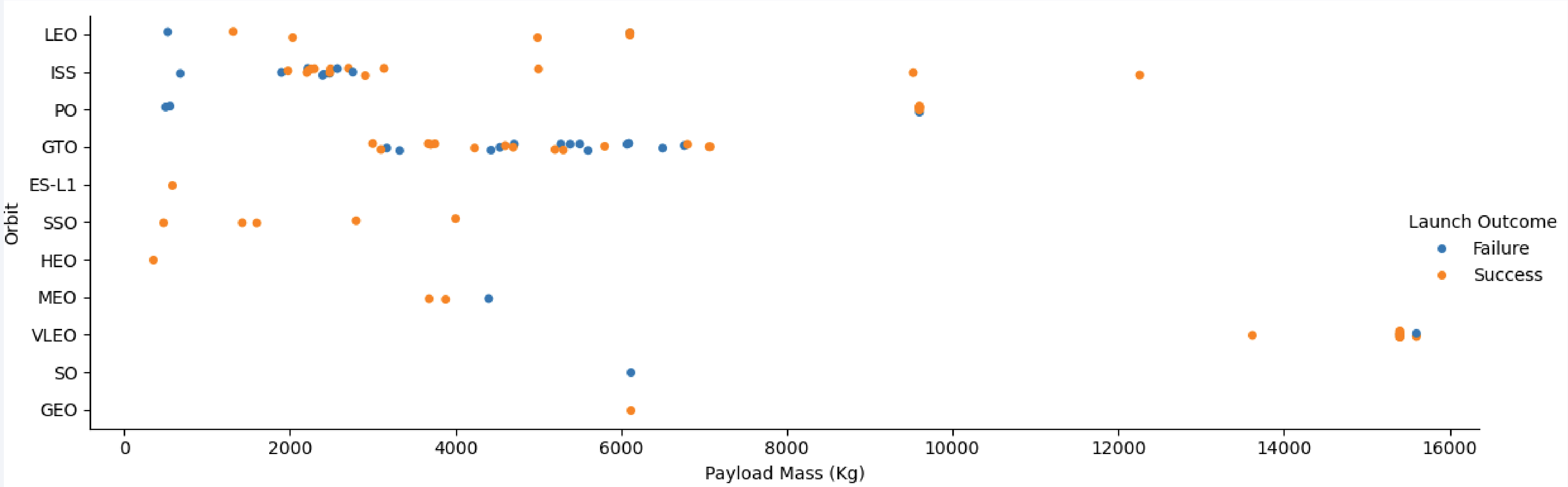
Flight Number vs. Orbit Type

- **Increased success rate over time**:
  - As seen in previous graphs, an increased experience is translated in better success rates.
- **Orbit-specific performance**:
  - Launch success of LEO, ISS, GTO show high number of failures initially and an improved performance over time.
  - Most orbits with 100% launch success started their activity after flight ~40 (except for ES-L1), favoured by a higher experience.



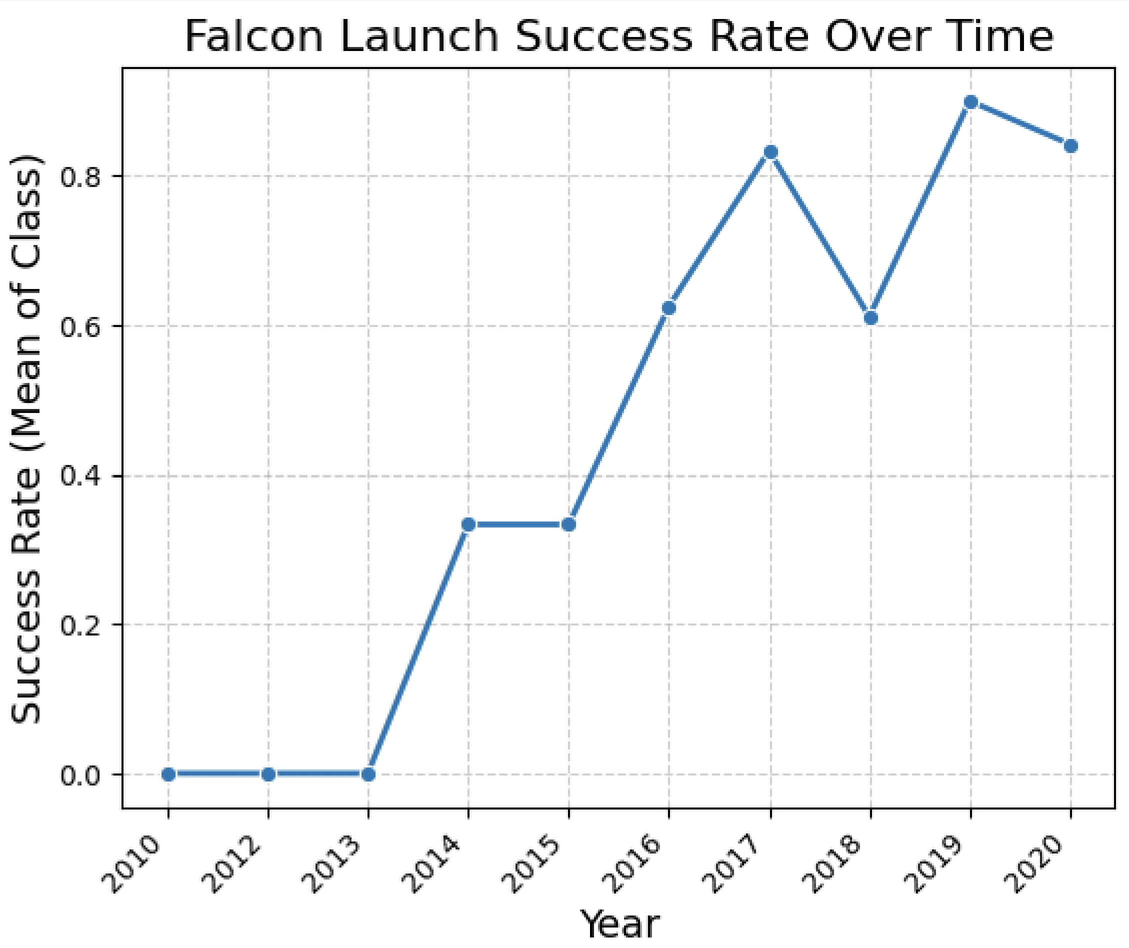
Payload vs. Orbit Type

- **Mixed success rate for payloads <10.000Kg**:
    - Mixed success launches are seen along this payload span.
- **Higher success rate for payloads >10.000Kg**:
    - It's worth noting that heavier payloads seem to experience higher success



Launch Success Yearly Trend

- **Increased success rate over time**:
  - Success rate has increased from 0% in early years to over 80% in the last few years.



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

[13]:
```
%sql SELECT Launch_Site FROM SPACEXTABLE GROUP BY Launch_Site
```

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

[13]:

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

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

[15]:
```
%sql SELECT Launch_Site FROM SPACEXTABLE WHERE SUBSTR(Launch_Site, 1, 3) = 'CCA' LIMIT 5
```

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

[15]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

[19]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE CUSTOMER = 'NASA (CRS)'
```

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

[19]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Task 4

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

```
[21]: %sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
```

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

[21]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

[26]:
```
%sql SELECT DATE FROM SPACEXTABLE WHERE Mission_Outcome = 'Success' ORDER BY DATE ASC LIMIT 1
```

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

[26]:

Date
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

[35]:
```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

[35]:

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[42]:
```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

[42]:

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

Boosters Carried Maximum Payload

Task 8

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

[44]:
```
%sql SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

[44]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[54]:
```
%sql SELECT SUBSTR(DATE, 6, 2) AS MONTH_NAME, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE SUBSTR(DATE,0,5)='2015' AND Landing_Outcome = 'Failure (drone ship)'
```

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

[54]:

MONTH_NAME	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

Task 10

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.

```
58]: %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Landing_Outcome
```

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

58]:

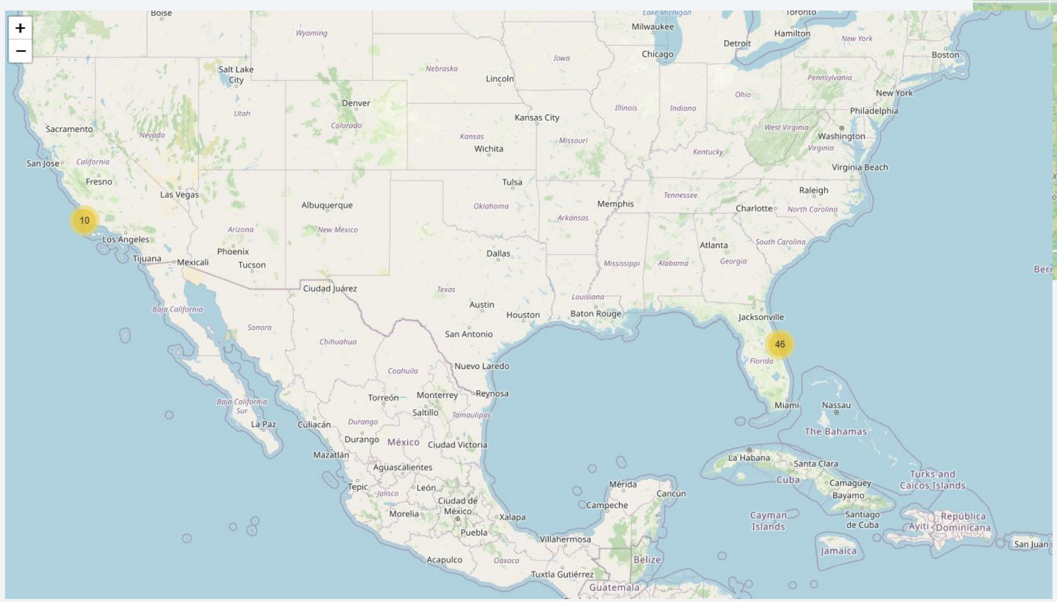
Landing_Outcome	COUNT(Landing_Outcome)
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	10
Precluded (drone ship)	1
Success (drone ship)	5
Success (ground pad)	3
Uncontrolled (ocean)	2

Section 3

Launch Sites Proximities Analysis

SpaceX's launch sites

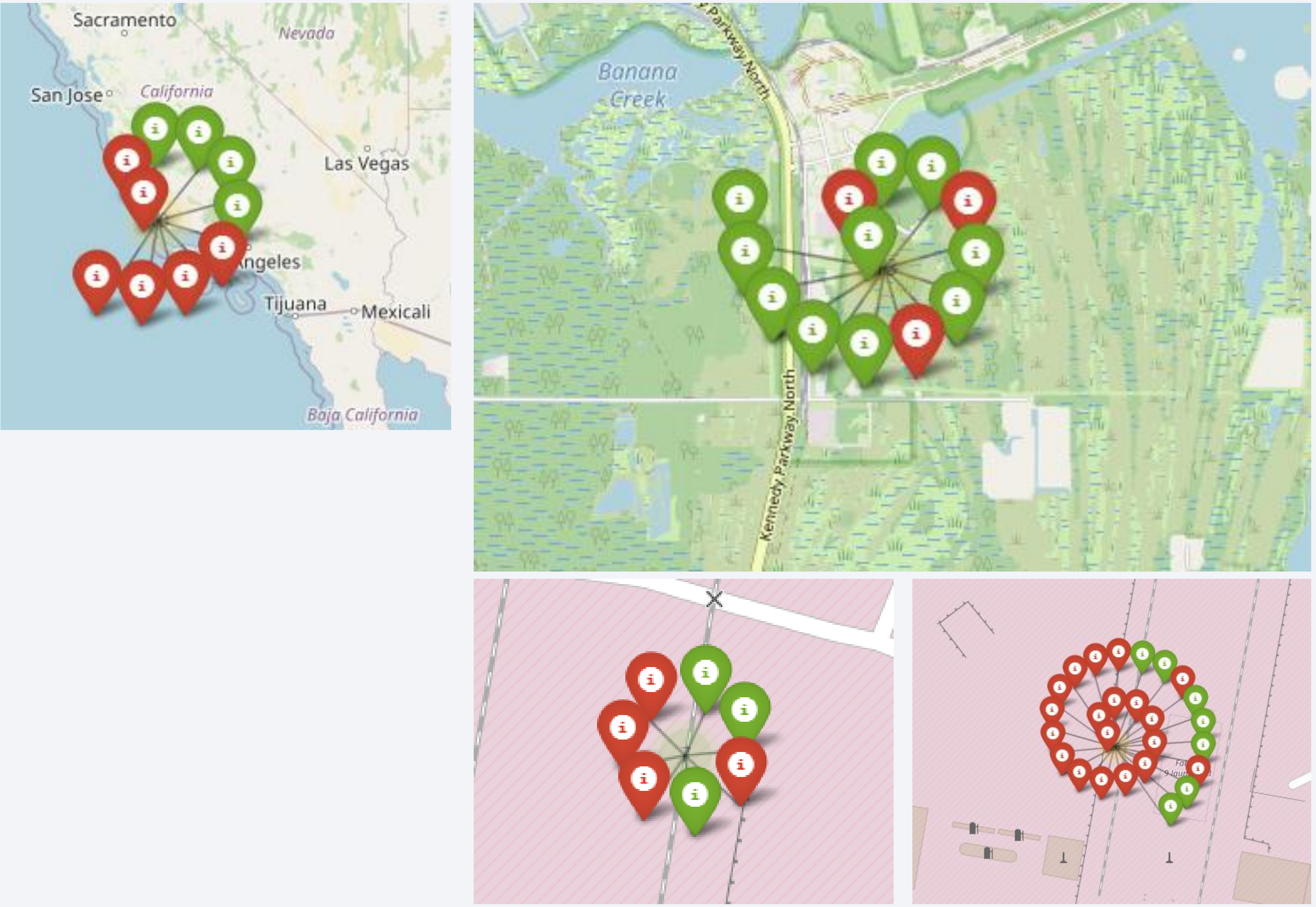
- **Are all launch sites in proximity to the Equator line?**
  - Not all launch sites are close to the Equator. The launch site at Vandenberg Air Force Base (VAFB SLC-4E) is located at a latitude of 34.63, which is further from the Equator compared to the other sites in Florida.



- **Are all launch sites in very close proximity to the coast?**
  - All launch sites are in close proximity to the coast. Cape Canaveral sites (CCAFS LC-40 and CCAFS SLC-40) and Kennedy Space Center (KSC LC-39A) are near the coast in Florida. Vandenberg Air Force Base (VAFB SLC-4E) is also near the coast in California.

Success/failed launches for each site

- **Enhanced visualization with clustered markers** allows for better exploration in the map.
- This **color-coding** helps to quickly identify the success rate and other categorical distinctions of the launches from this specific site. The red markers might represent unsuccessful launches, while the green markers indicate successful ones, providing immediate visual feedback on the performance of launches at each site.



Distances between a launch site to its proximities

- Considering VAFB launch site, the map shows the **distance to Highway 101 and to the closest town** (Lompoc). *37.84 km and 11.76 km respectively*

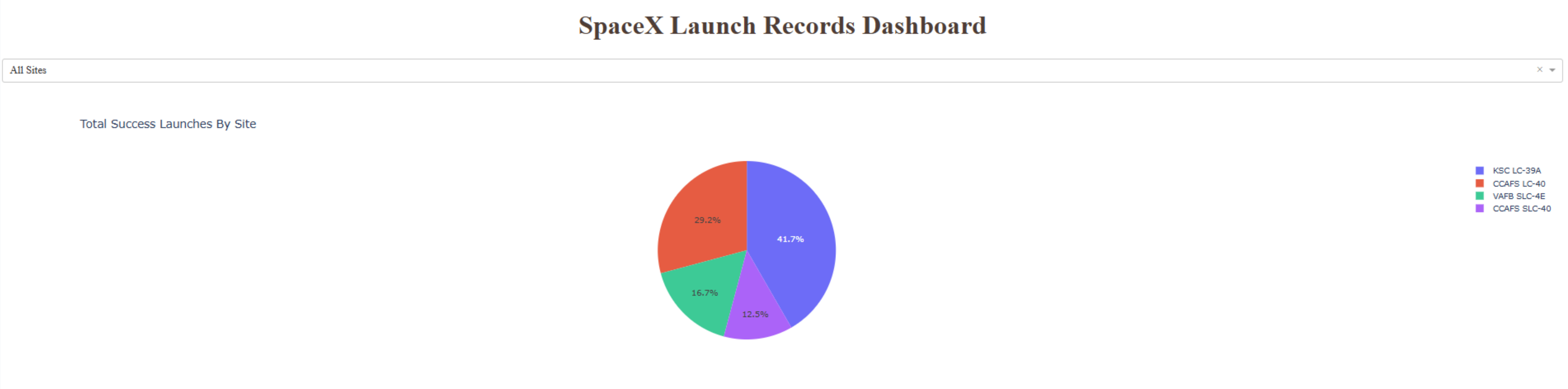

```
[22]: # Create a marker with distance to a closest city, railway, highway, etc.
# Draw a line between the marker to the launch site

#Closes city to VA
city_lat = 34.6323
city_lon = -120.4822
coord_city = [city_lat, city_lon]

distance_marker1 = folium.Marker(
   location=[coord_city[0], coord_city[1]],
   icon=DivIcon(
       icon_size=(20,20),
       icon_anchor=(0,0),
       html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f} KM".format(calculate_distance(VA_launch_site_lat, VA_launch_site_lon, coord_city[0], coord_city[1])),
       )
   )
site_map.add_child(distance_marker1)

lines1=folium.PolyLine(locations=[[coord_city[0], coord_city[1]], [VA_launch_site_lat, VA_launch_site_lon]], weight=1)
site_map.add_child(lines1)


#Closest highway to VA
hw_lat = 34.7202
hw_lon = -120.2110
coord_hw = [hw_lat, hw_lon]

distance_marker2 = folium.Marker(
   location=[coord_hw[0], coord_hw[1]],
   icon=DivIcon(
       icon_size=(20,20),
       icon_anchor=(0,0),
       html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f} KM".format(calculate_distance(VA_launch_site_lat, VA_launch_site_lon, coord_hw[0], coord_hw[1])),
       )
   )
site_map.add_child(distance_marker2)

lines2=folium.PolyLine(locations=[[coord_hw[0], coord_hw[1]], [VA_launch_site_lat, VA_launch_site_lon]], weight=1)
site_map.add_child(lines2)

site_map
```

Section 4

Build a Dashboard with Plotly Dash

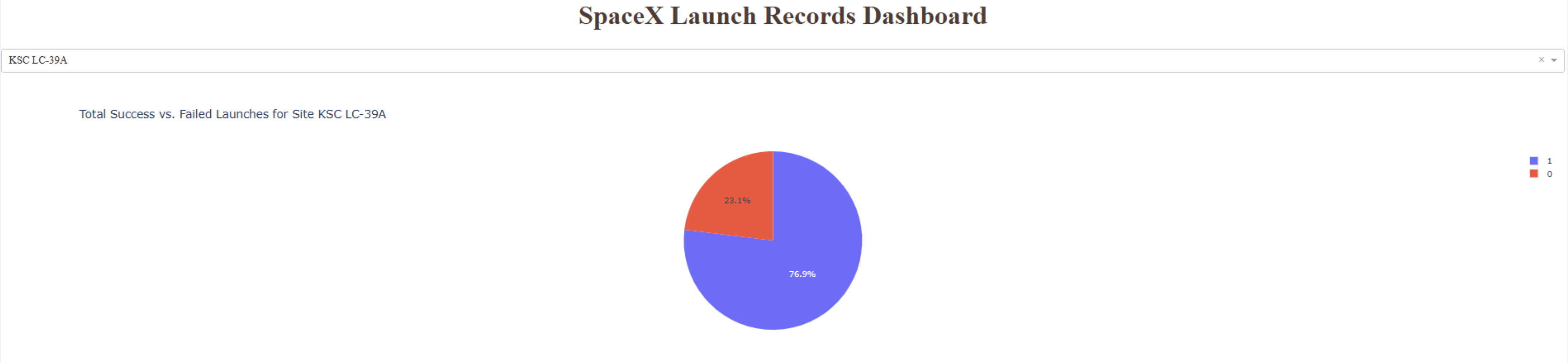
Launch success count for all sites, in a piechart

- KSC LC-39A is a highly reliable site, as it has the highest number of successful launches, making up 41.7% of all successful launches.



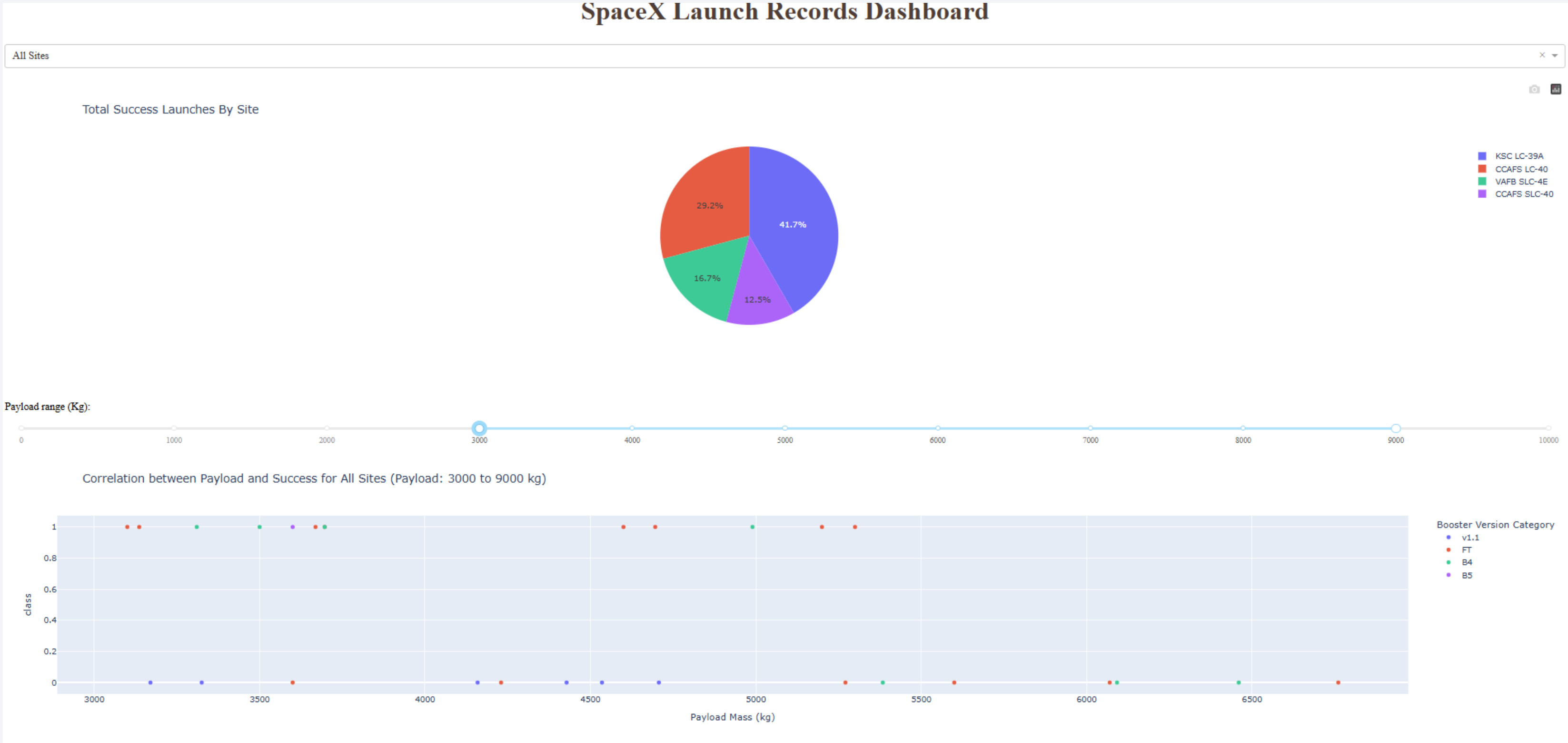
Launch site with highest success ratio

- The high success rate (76.9%) for Class 1 launches underscores the effectiveness and reliability of the KSC LC-39A site.



Correlation bw. payload and success for a range of payloads

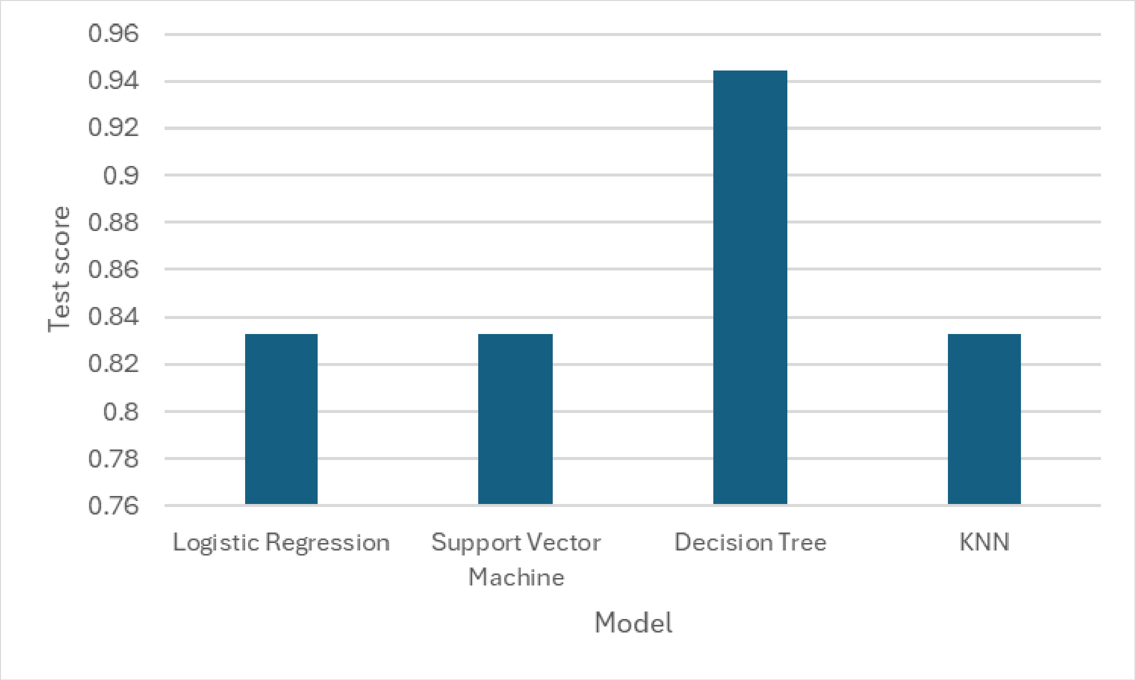
- **For the selected payload range (3.000Kg-9.000Kg):**
  - For payloads >5.000Kg, only FT and B4 boosters are used.
  - Booster B5 has been used only once, suggesting recent design.
  - All boosters experience mixed successful rates, except for boosters v1.1 which show null successful rate.



Section 5

Predictive Analysis (Classification)

Classification Accuracy



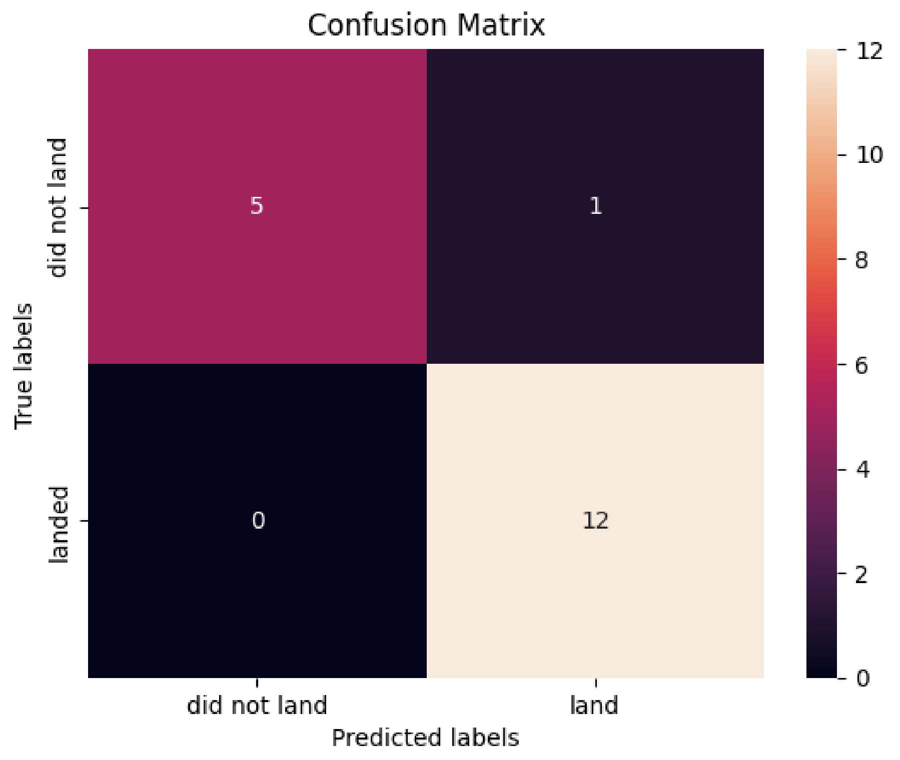
- **Highest accuracy of 0.94 is achieved with a Decision Tree Classifier**
- Models tested:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree Classifier
  - KNN

Confusion Matrix

Explanation and insights

- **Highest Accuracy** achieved, with a score of 94.4%
- **No false negatives**, suggesting the model reliably predicts successful landings
- **Limited false positives**, suggesting also an interesting reliability detecting failed launches.
- **Balanced performance overall.** Even the model is slightly biased towards successful landing, overall performance is better than the rest of models, with a high accuracy. This allows good estimates of the company's future planning.

```
[94]: yhat = tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Launch site KSC LC-39A has the highest success rate among all sites, accounting for 41.7% of successful launches. This indicates that this site might have optimal conditions or processes that contribute to a higher success rate.
- Booster v1.1 has the lowest success rate across various payload masses, showing its limited reliability and robustness compared to other booster versions. This suggests that future missions will avoid using v1.1 booster version to avoid reliability issues.
- No clear pattern was observed linking higher payload masses to lower success rates, indicating that factors other than payload mass, such as launch site conditions and booster versions, play a more significant role in determining the outcome of a launch.
- Interactive data visualizations using Folium provide valuable insights in a map. It allowed understand that launch sites are all located by the sea, separated from public infrastructure.
- Tree Decision Classifier is the selected ML model to reliably predict the outcome of future launches and provide proper planning.
- The insights gathered can help improve launch strategies and contribute to the ongoing success of reusable rocket technology to be designed for SpaceY.

Thank you!