Winning Space Race with Data Science

Brett C. Daffron
4/28/2025

Outline

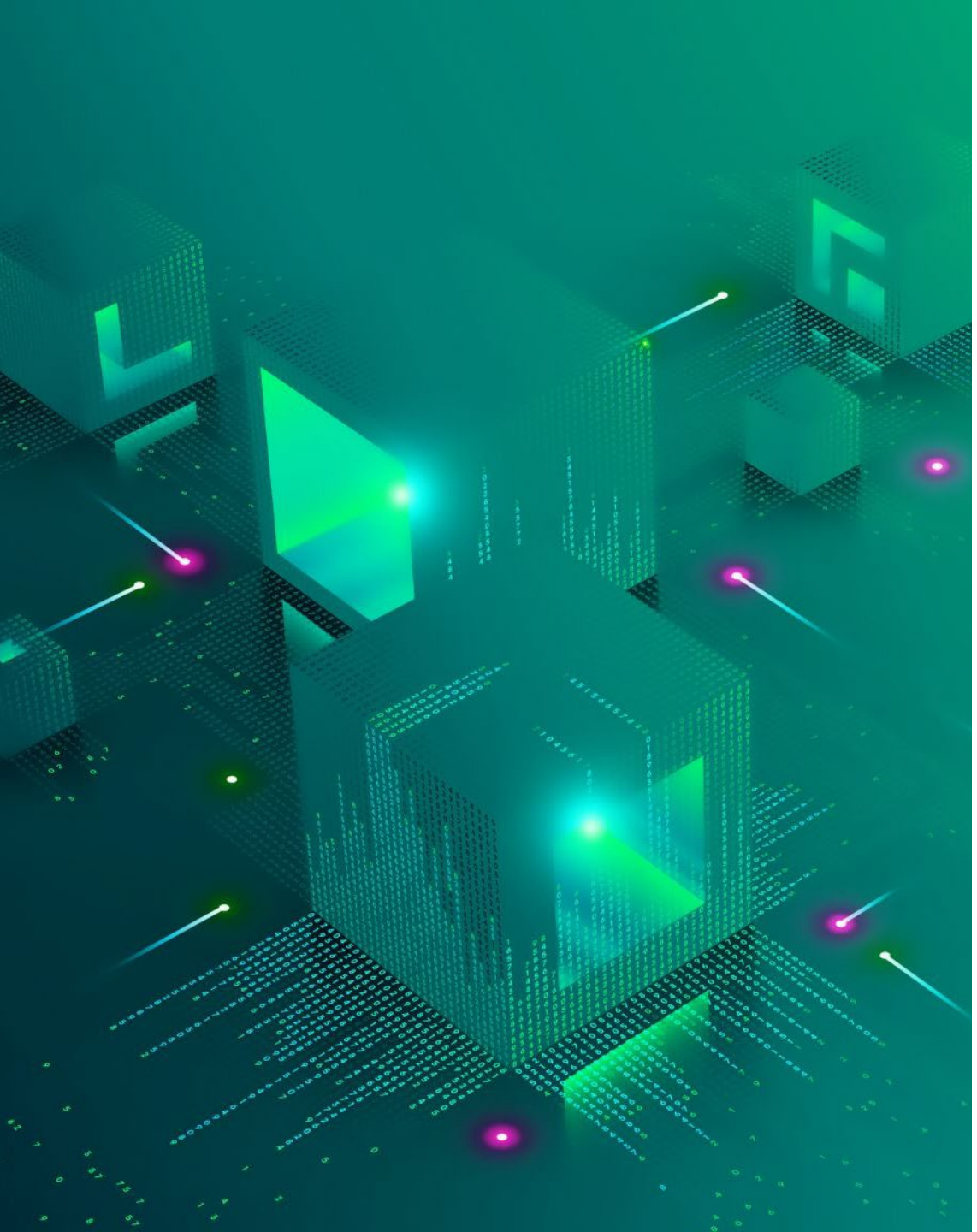
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

This a structured data science methodology, beginning with data collection through web scraping and public datasets, focusing on SpaceX launch data provided by the IBM course. First the data is collected, then is used in exploratory data analysis (EDA) using tools like SQL, Pandas, and visualization libraries (Matplotlib, Seaborn, Folium) to uncover patterns, trends, and correlations. These insights are used to plan the next steps in the analysis.

For predictive modeling various machine learning algorithms (e.g., logistic regression, SVM, decision trees) are used to forecast the success of SpaceX launches. this process includes feature engineering, data normalization, model training, and evaluation using metrics like accuracy and confusion matrices. Folium and Plotly Dash are used to create interactive visualizations of the data that allow for easier interpretation.

Each of these steps provides a path to better interpreting structured data and making predictions for future use. This can save resources and time in future endeavors leading to a more streamlined and profitable use of both. In this case a company can decide what features of a rocket launch leads to the safe return of the payload. Saving more money and resources necessary for future launches, per each safe return.



Introduction

This project, was developed as part of the IBM Applied Data Science Capstone, it explores historical SpaceX launch data to uncover patterns that influence launch success. Through evaluating variables such as launch site, payload mass, and booster version, the notion is to determine the ideal conditions for completing a successful launch and safe booster landing. Through data collection, exploratory analysis, and predictive modeling, the intent is to support decision-making for future missions by identifying factors that maximize launch reliability and safety.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- The data was collected using a combination of web scraping and publicly available datasets. Web scraping techniques were used to gather SpaceX launch data from online sources, and additional structured data was retrieved from a SpaceX API and CSV files provided by the course to support analysis.

Perform data wrangling

- Data wrangling involved cleaning and transforming the raw data to prepare it for analysis. This included handling missing values, converting data types, extracting relevant features and merging datasets to create a cohesive and structured format suitable for exploratory analysis and modeling.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

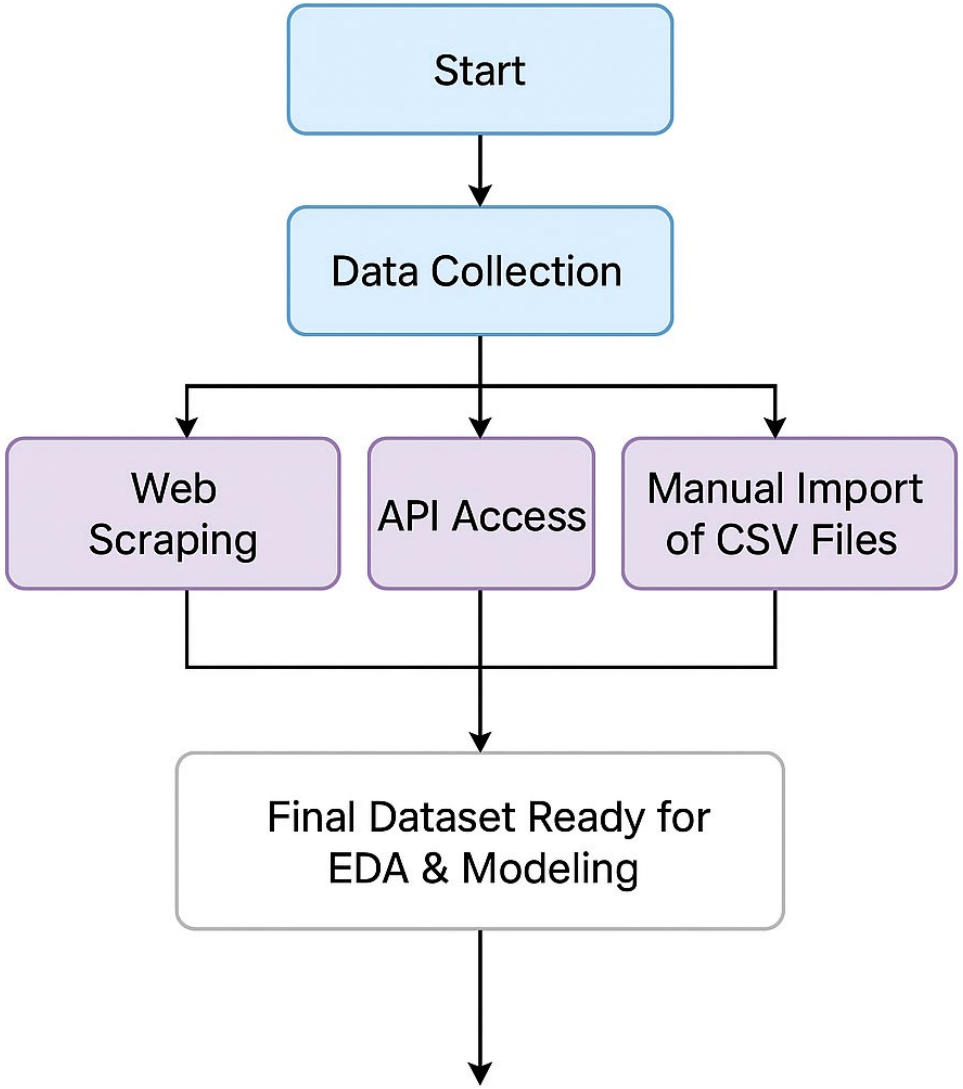
Perform predictive analysis using classification models

- Classification models were built by selecting key features from the cleaned dataset and training algorithms such as logistic regression, decision trees, and support vector machines. These models were then tuned using techniques like GridSearchCV, and evaluated with metrics such as accuracy, precision, recall, and confusion matrices to assess their performance.

Data Collection

Data Collection Process Description

The dataset for this project was collected through a multi-step pipeline combining web scraping, API access, and manual data import. Web scraping techniques using Python libraries like Beautiful Soup were used to retrieve launch data from SpaceX's official launch records. Structured data was obtained using the SpaceX REST API, providing detailed technical attributes such as payload mass, booster versions, and launch success. Additionally, CSV files provided by the IBM course were imported to supplement the scraped data and ensure completeness.

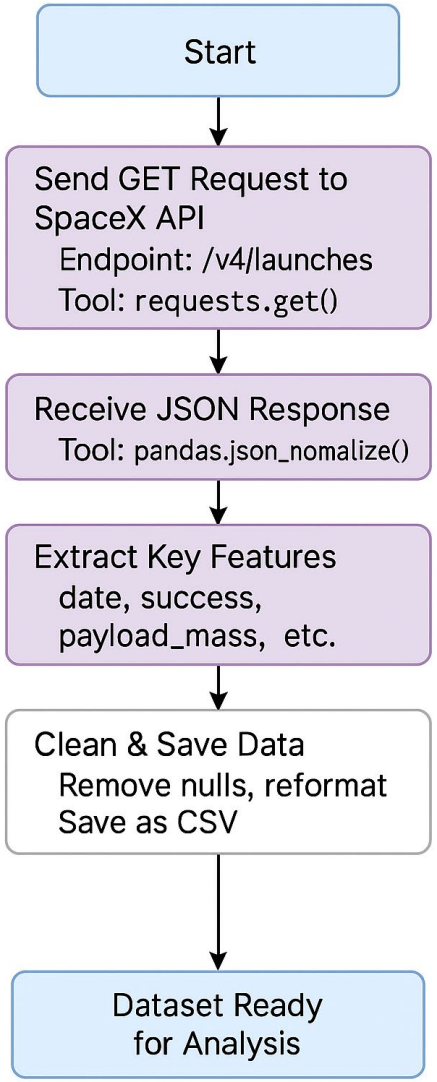


Data Collection – SpaceX API

- **API Endpoint**: Used https://api.spacexdata.com/v4/launches to retrieve structured launch data.
- **Request Handling**: Sent GET requests using the requests library in Python.
- **JSON Parsing**: Converted JSON response into Python dictionaries and dataframes using json and pandas.
- **Data Normalization**: Applied pd.json_normalize() to flatten nested structures (e.g., payload and rocket info).
- **Filtering**: Extracted key features such as date_utc, launchpad, payload_mass_kg, success.
- **Storage**: Saved the clean dataset locally as a .csv file for reuse in modeling and analysis.

Git URL-

https://github.com/BrettDaff/Capstone/blob/main/spacex-data-collection-api.ipynb

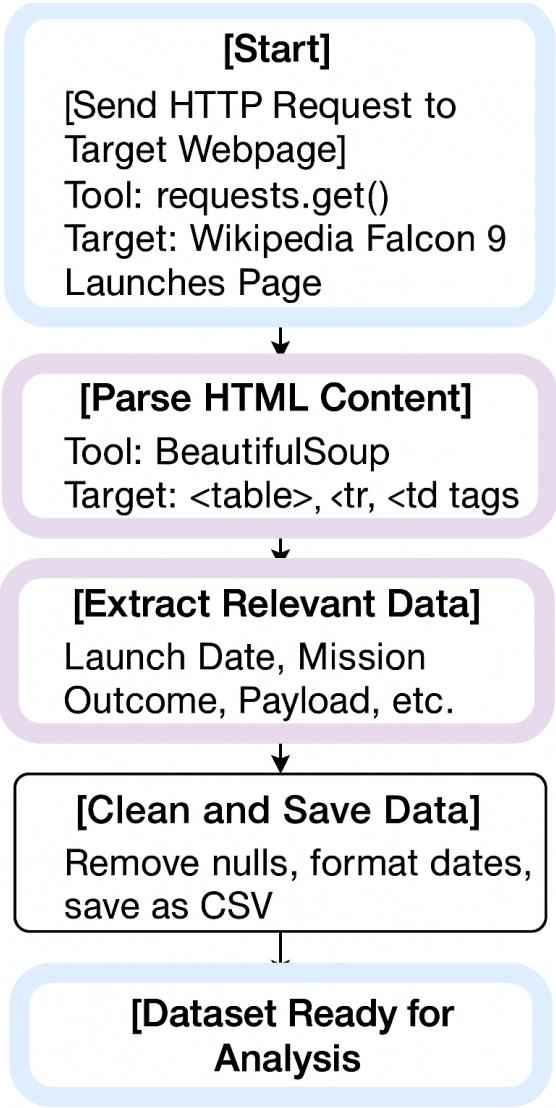


Data Collection - Scraping

- **Target Source**: Scraped SpaceX launch data from a Wikipedia page listing Falcon 9 launches.
- **Libraries Used**: Utilized requests to retrieve the HTML content and BeautifulSoup to parse and navigate the webpage structure.
- **Data Extraction**: Identified HTML table elements using tags like <table>, <tr>, and <td>, and extracted rows containing launch details.
- **Data Structuring**: Organized the extracted data into a structured format (list of dictionaries), then converted it into a pandas DataFrame.
- **Data Cleaning**: Removed empty rows, handled inconsistent formats (e.g., date and payload strings), and saved the cleaned dataset for analysis.

Git URL-

https://github.com/BrettDaff/Capstone/blob/main/jupyter-labs-webscraping.ipynb

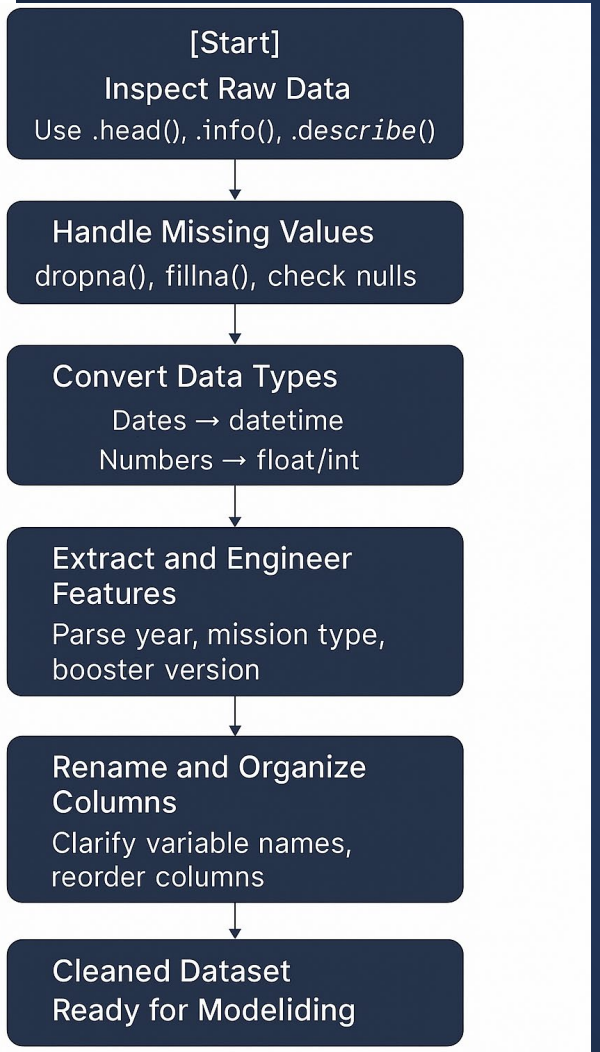


Data Wrangling

- **Initial Inspection**: Used .info(), .head(), and .describe() to understand data types and structure.
- **Handling Missing Values**: Identified and removed or imputed missing/null values using dropna() or fillna().
- **Data Type Conversion**: Converted string/object columns to datetime or numeric types as needed (e.g., pd.to_datetime()).
- **Feature Extraction**: Parsed relevant features from complex columns (e.g., extracting year from launch date).
- **Column Renaming & Reordering**: Used df.rename() and reordering techniques for better clarity and analysis.
- **Merging Datasets**: Joined scraped, API, and CSV data sources using pd.merge() to create a unified dataset.
- **Final Output**: Cleaned dataset saved to CSV for downstream analysis and modeling.

- **Git URL-**
- https://github.com/BrettDaff/Capstone/blob/main/spacex-data-collection-api.ipynb

- https://github.com/BrettDaff/Capstone/blob/main/jupyter-labs-webscraping.ipynb



Section 2

Insights drawn from EDA

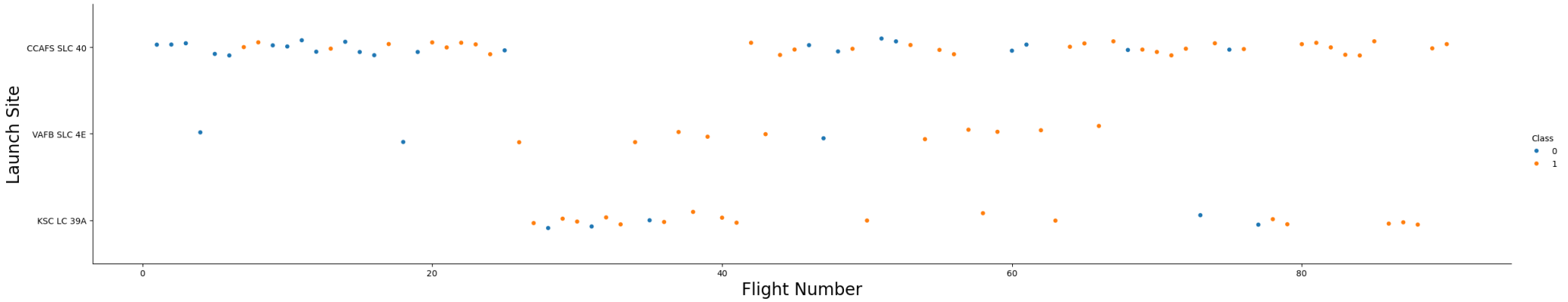
EDA with Data Visualization

- **Catplots**
  - Helped determine the distribution of launches and the payload mass of those launches across the various launch sites, as well as the success of those launches.
- **Barchart**
  - Showed the relationship between the type of orbit(Low Earth, Geostationary, Etc.) and the success rate of landing
- **Scatterplots**
  - Determined the relationship of success between number of flights and a particular orbit type.
  - Determined the relationship of success between the payload mass and the orbit type.
- **Lineplot**
  - Showed the yearly success rate.

After determining various important predictive features using visualizations we go through the process of feature engineering. This includes separating the determined important predictive features into a separate data frame for further processing and use. Processing includes using one hot encoding which transforms categorical data into binarily classified units, allowing future predictive models to better interpret a specific categories effect on the data. Essentially this turns the entire data frame into numeric values where each category is represented to be on or off, on being 1 and off being 0. The last step is casting the data frame to data type float64 so that the predictive models can read the data as numeric values.

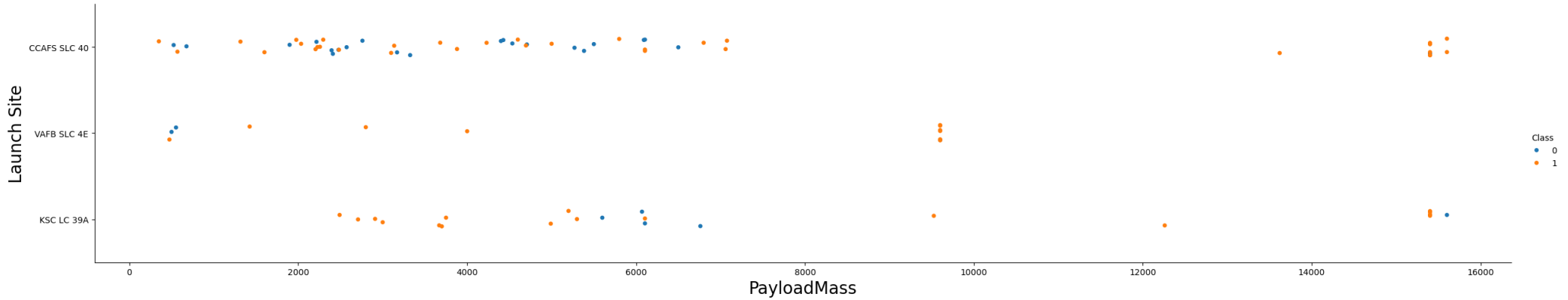
Git URL-

https://github.com/BrettDaff/Capstone/blob/main/edadataviz.ipynb



Flight Number vs. Launch Site

- The scatter plot indicates that as a launch site participates in more launches the success rate increases. This is visualized by the dots being more consistently orange, indicating success, as the flight number increases.

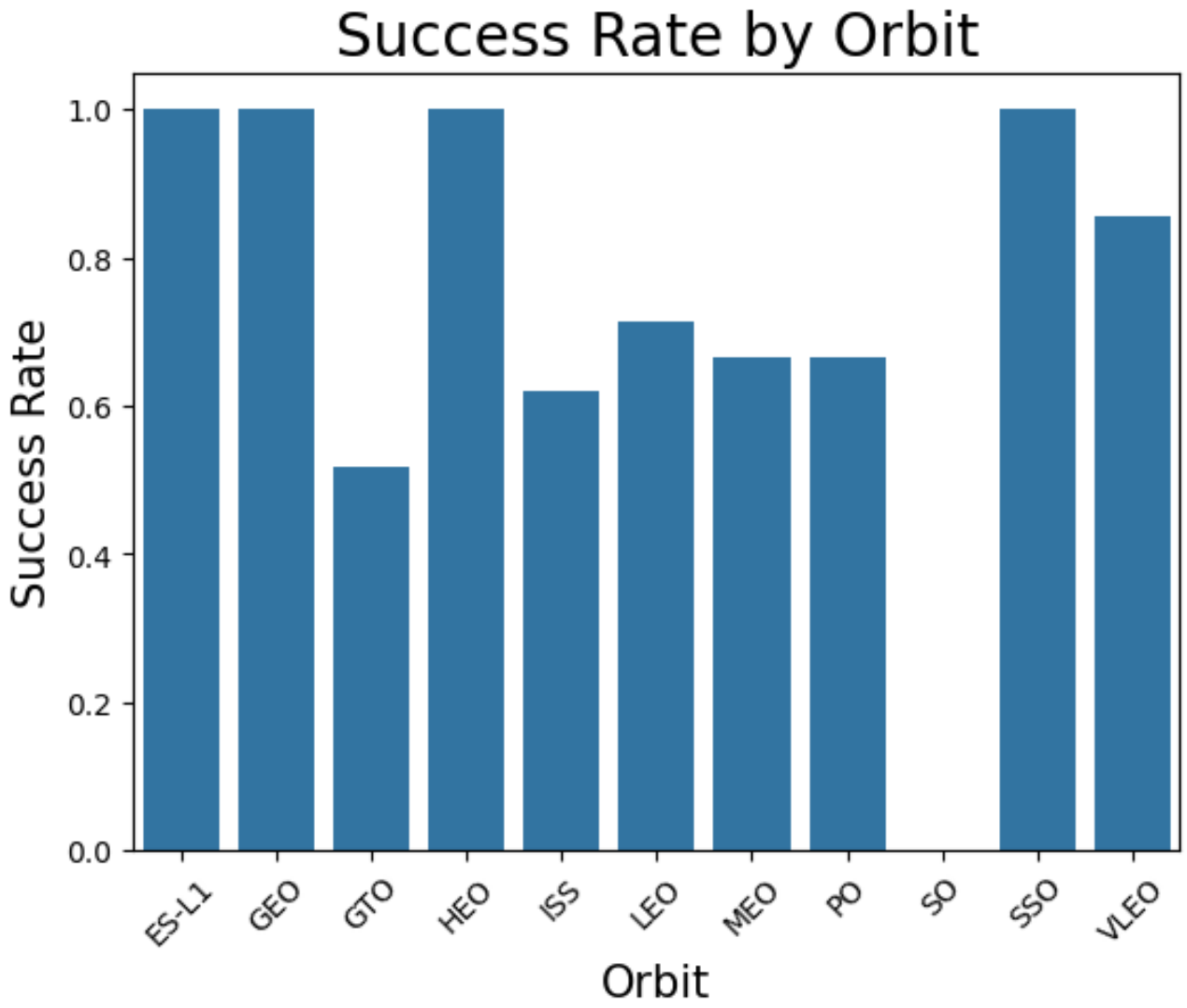


Payload vs. Launch Site

- Site CCAFS SLC 40 seems to have a varied success rate between 0-8000kg but a great success rate above the 12000kg range
- Site VAFB SLC 4E has a great overall success rate and based on the data has yet to exceed 10000kg
- Site KSC LC 39A seems to struggle in the 6000kg range

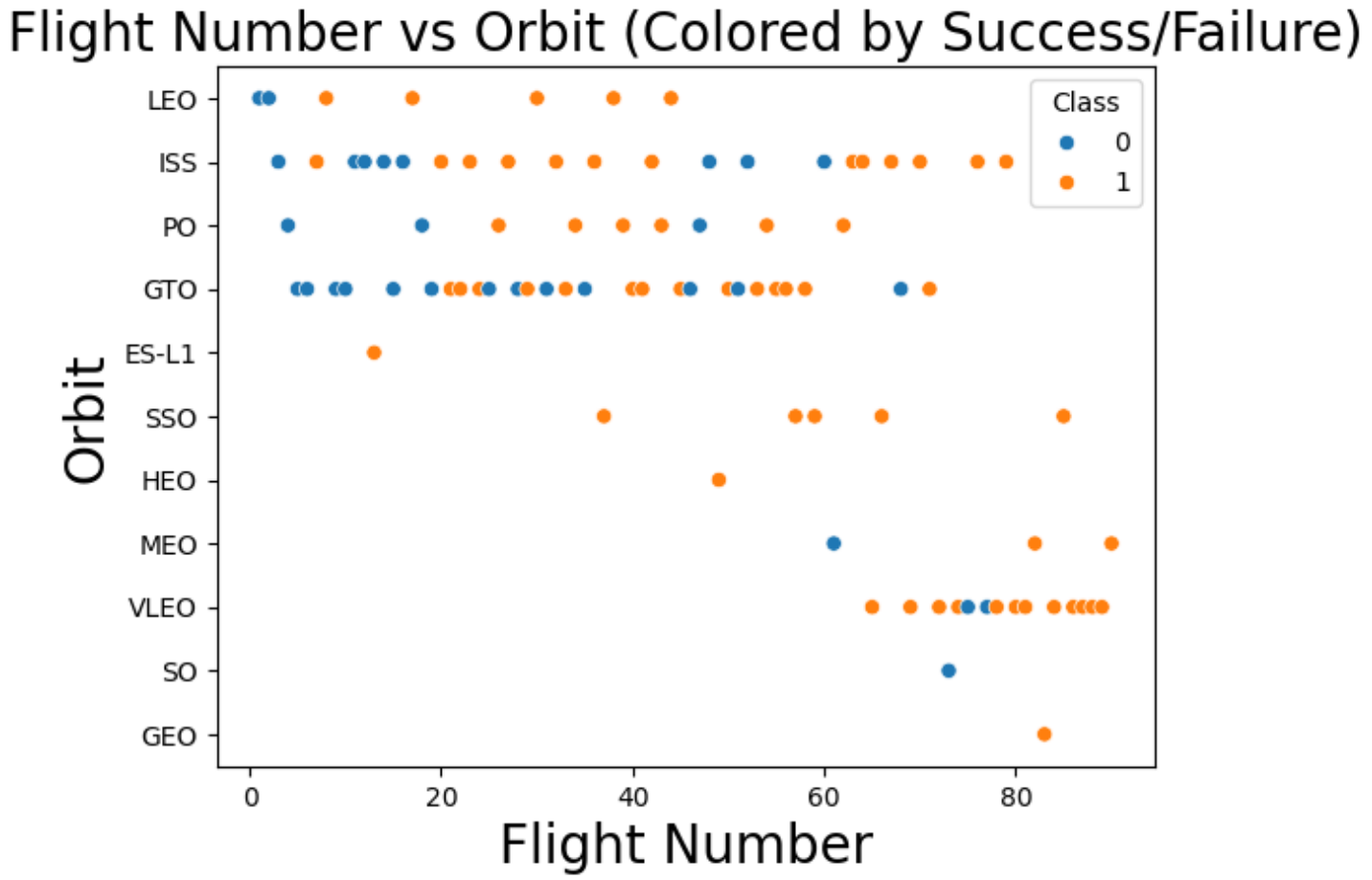
Success Rate vs. Orbit Type

- Orbit types with the best success rate are ES-L1, GEO, HEO, SSO
- GEO, and ISS seem to have the lowest success rate while still being near or above 50% success
- SO appears to have no successful launches either this means there isn’t data on SO, SO does not have enough successful launches to appear in this chart, or SO has never completed a successful launch and retrieval



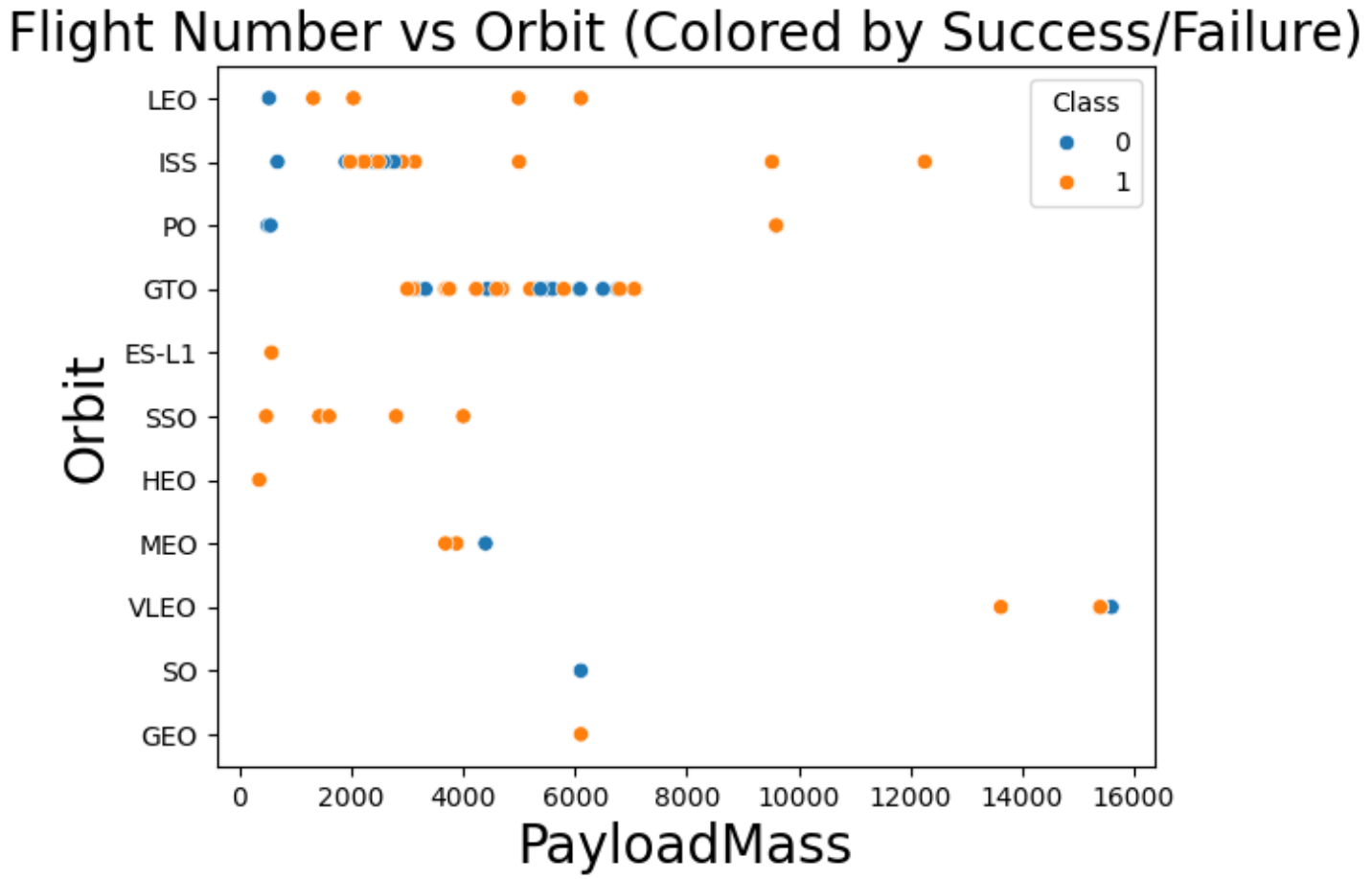
Flight Number vs. Orbit Type

- Based on this data SO has a 1 for 1 failure rate and GEO has a 1 for 1 success rate this means the orbit vs. success chart weighs these orbit types heavier due to less comparable data
- LEO seems to have a stronger correlation of success with more flights, ISS and PO seem to be similarly indicated but GTO has little to no indication of greater success with more flights



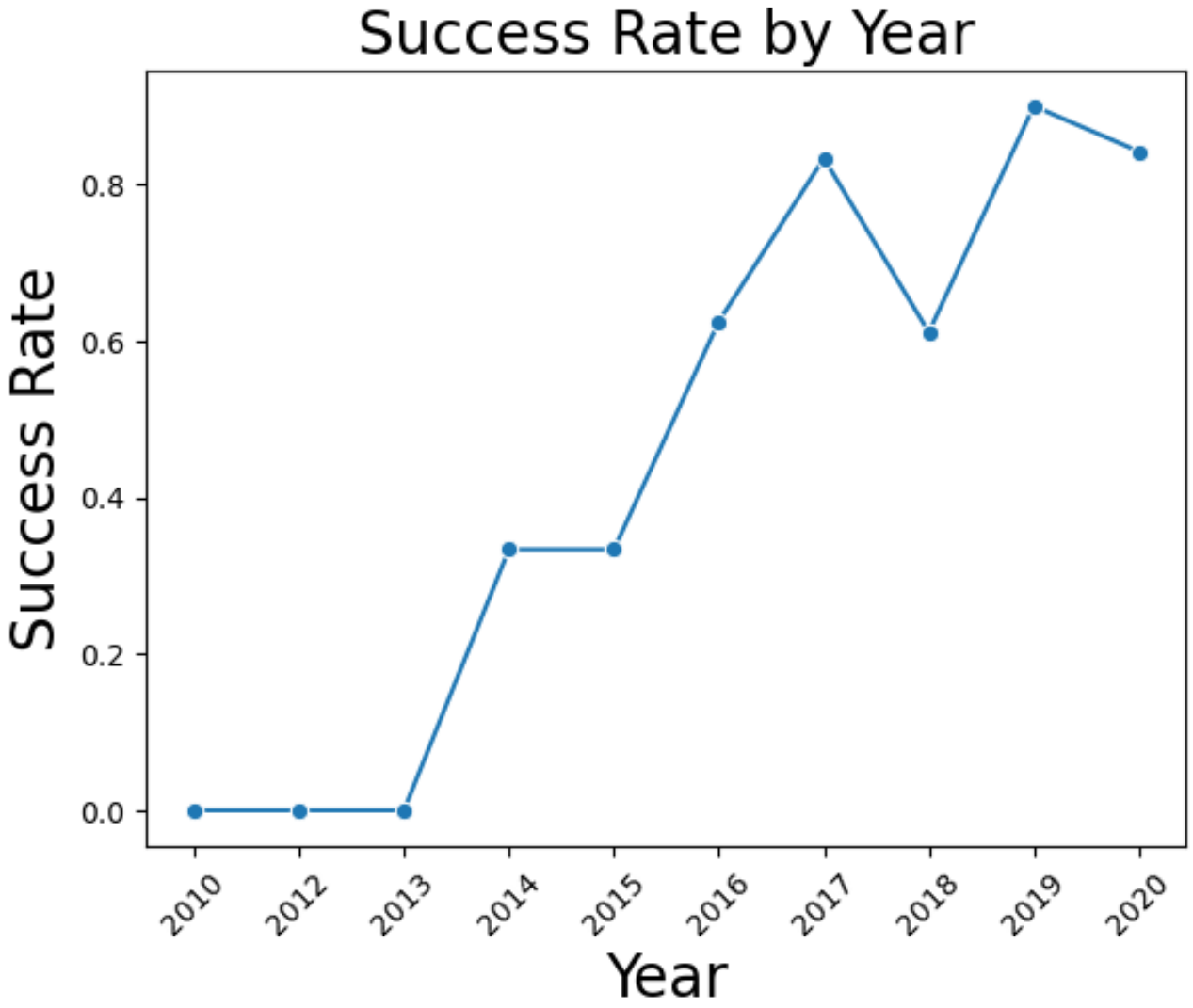
Payload vs. Orbit Type

- LEO and ISS seem to have a greater success with higher payloads
- GTO seems to have little difference in success depending on payload size
- SSO does not have much data to go on but seems to remain stable in success as payload increases.



Launch Success Yearly Trend

- This plot shows a direct correlation with success overtime
- Ther is a fairly dramatic downturn in success between 2017 and 2018
- Between 2018 and 2019 the rate of success returns but then begins a slight downward trend into 2020
- These downturns could possibly be explained by economic factors or changes in the companies testing methods



EDA with SQL

Displayed the names of the unique launch sites in the space mission:

- **%sql** SELECT DISTINCT Launch_Site FROM SPACEXTABLE;

Displayed 5 records where launch sites begin with the string 'CCA':

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

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

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

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

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

Listed the total number of successful and failure mission outcomes

Listed all the booster_versions that have carried the maximum payload mass. Useing a subquery.

Listed the records which displayed the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Ranked 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.

Git URL-

https://github.com/BrettDaff/Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

All Launch Site Names

Launch Sites:

- 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 SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT %sql 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
- %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE WHERE Customer LIKE '%NASA (CRS)%';

Total_Payload_Mass
48213

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';

Avg_Payload
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- %sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)';

First_Successful_Landing_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
- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 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
- %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTABLE 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
- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

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

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

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
- %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;

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

Results

Exploratory Data Analysis & Visualization

Most frequent launch sites identified were CCAFS SLC 40 and KSC LC 39A.

Launch outcomes showed a mix of success and failure, with overall high success rates.

Payload mass distributions indicated most launches carried 2000–6000 kg payloads.

Correlation plots showed limited direct linear correlation between features and landing success.

Visualizations helped guide feature selection for machine learning.


Section 3
Launch Sites
Proximities Analysis

Build an Interactive Map with Folium

- **Markers**
  - Added for each SpaceX launch site (e.g., CCAFS, VAFB, KSC).
  - Purpose: To visually identify and label each launch site on the map.
- **Circles**
  - Drawn around each launch site using a fixed radius.
  - Purpose: To highlight the region around each site, improving spatial visibility and scale perception.
- **Lines (Polylines)**
  - Created to draw distance paths between launch sites and nearby facilities (e.g., coastlines, airports, railroads).
  - Purpose: To evaluate proximity and potential safety/accessibility factors for launch logistics.
- **Popups & Tooltips**
  - Added to markers to display site names and additional info interactively.
  - Purpose: To provide users with immediate contextual data when hovering or clicking.

These visual map elements were added to:

- Analyze the geographic distribution of launch sites,
- Evaluate environmental and logistical factors (like distance to infrastructure),
- And to make interactive exploration of the launch environment easy and intuitive.

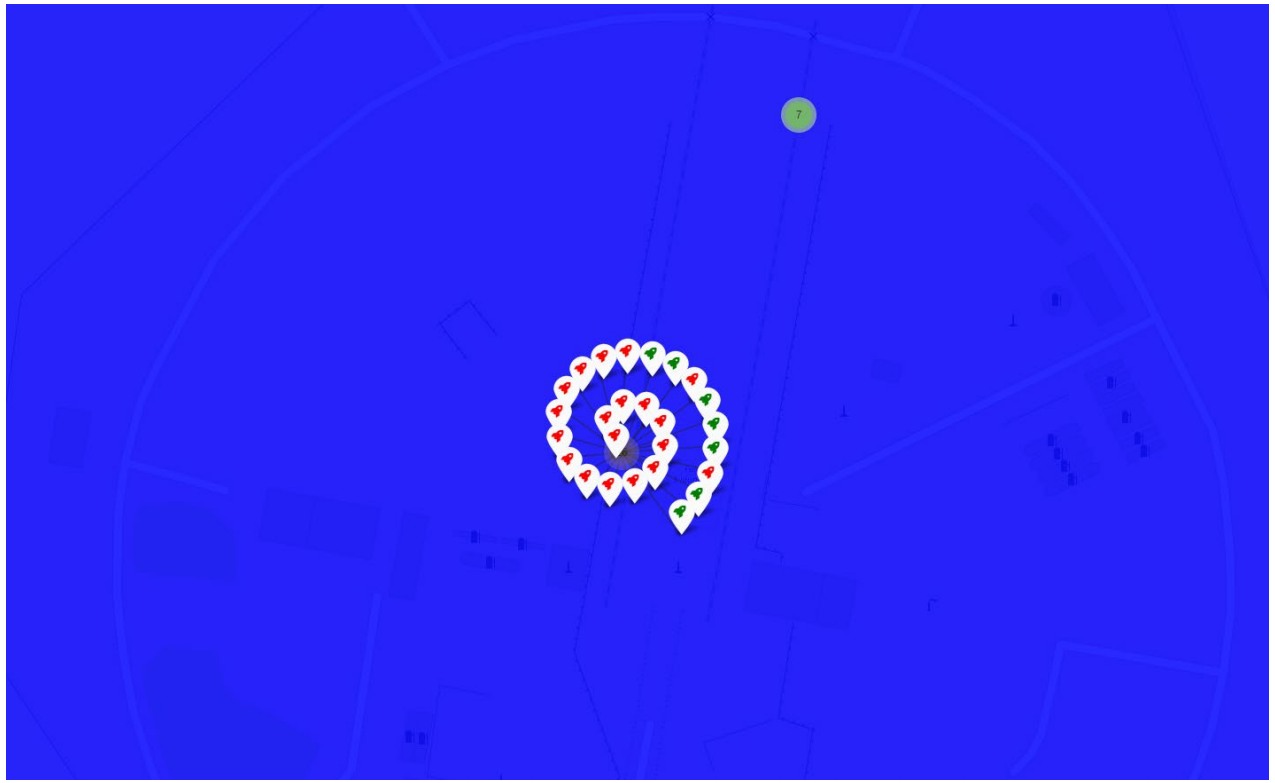
Git URL-

https://github.com/BrettDaff/Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Launch Sites

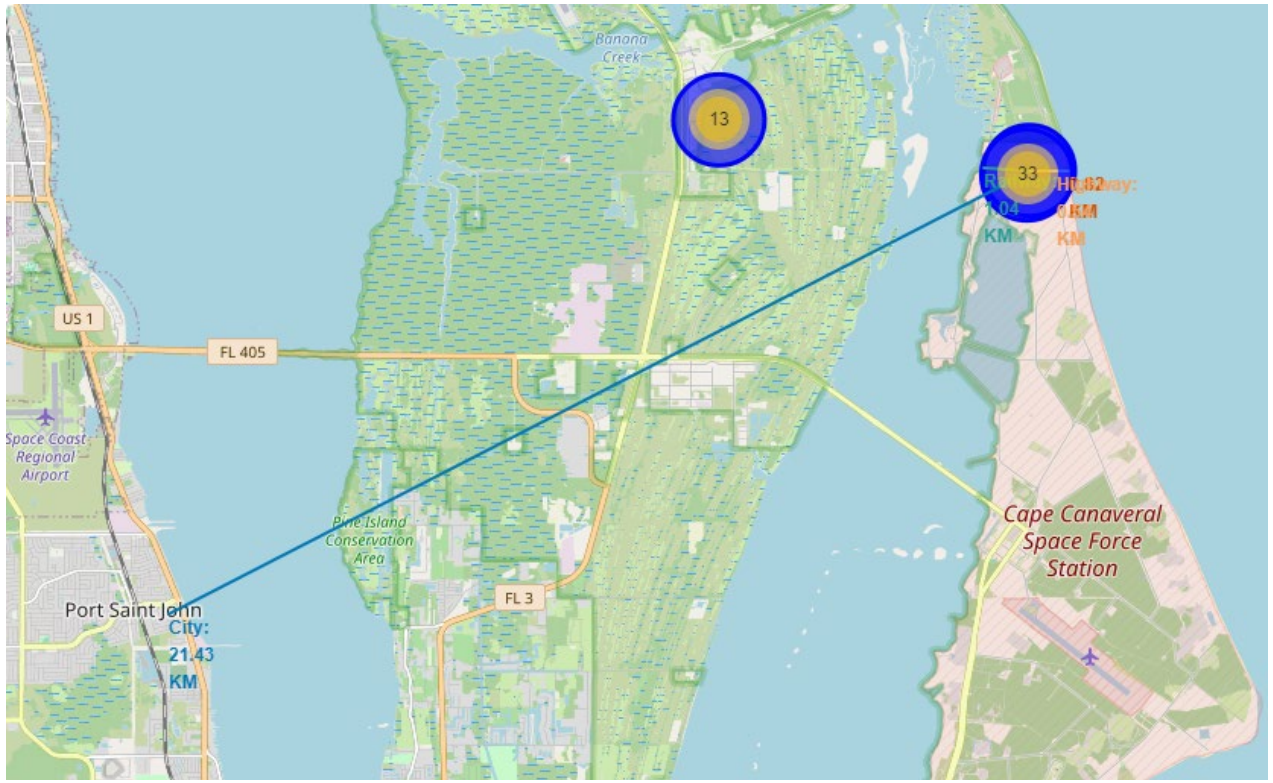
- This folium map shows the location and count of launches

Launch Success

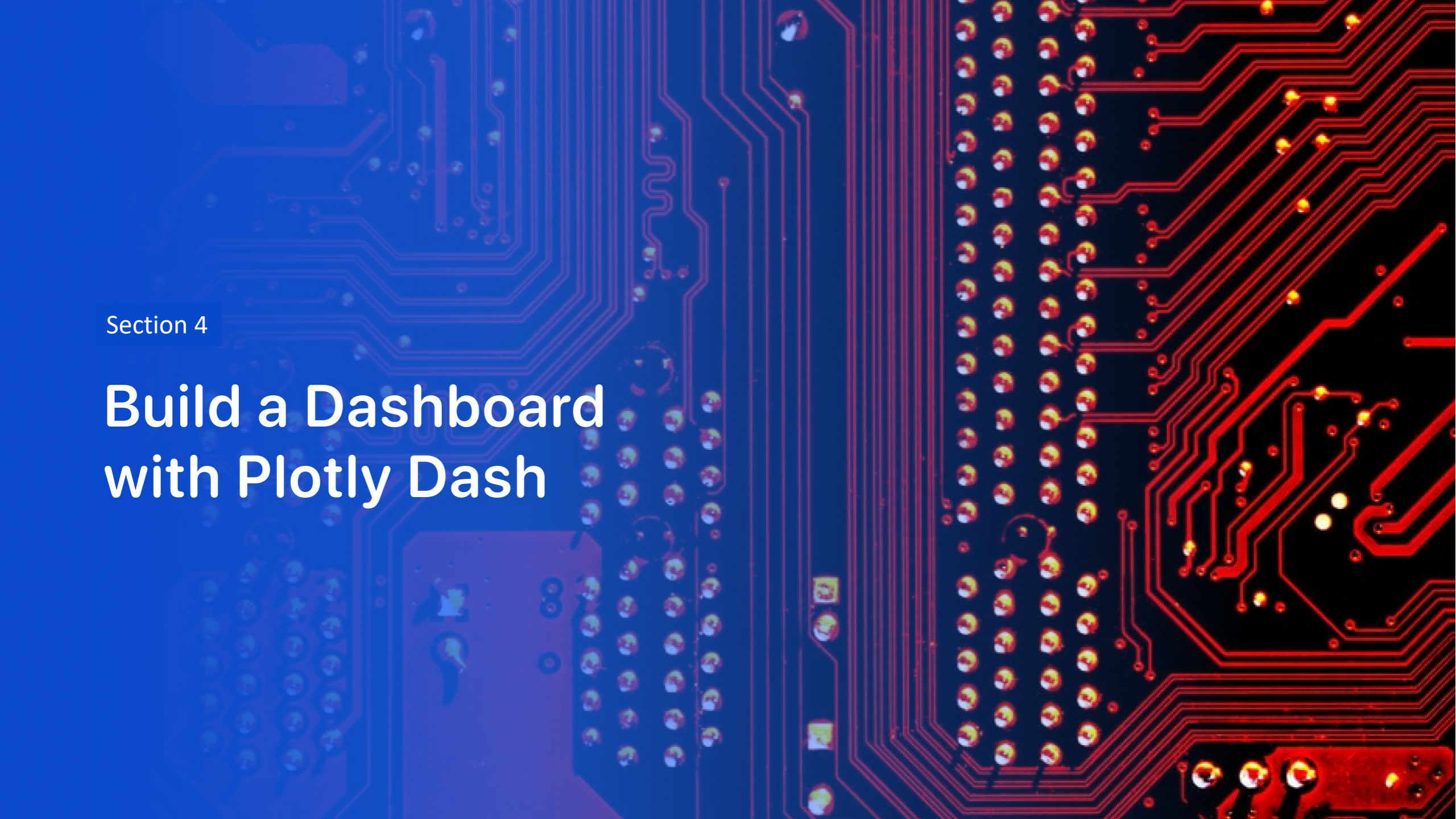


- This zoomed in portion of the map displays the number of successful and failed launches at a given site

Launch Distance



- This shows the distance between a given launch site and the nearest city, rail way, highway, and coastline
- This data gives us the indication that launch sites need to be a certain distance from cities and it seems that close proximity to coastlines and railways is ideal


Section 4
Build a Dashboard with Plotly Dash

Build a Dashboard with Plotly Dash

Plots/Graphs in the Dashboard

1.Pie Chart – Total Successful Launches by Site

- •Displays the count of successful launches grouped by launch site.
- • Purpose: To provide a quick visual comparison of how frequently each site achieves a successful mission.

2.Scatter Plot – Correlation Between Payload Mass and Success

- •Plots each launch as a point, with payload mass on the x-axis and mission outcome on the y-axis.
- • Purpose: To explore if there's a relationship between payload size and landing success.

Interactive Features

1.Dropdown Menu (Launch Site Selector)

- •Allows users to filter the pie and scatter plots by a specific launch site or view all sites.
- • Purpose: Helps users analyze launch performance site-by-site.

2.Payload Mass Range Slider

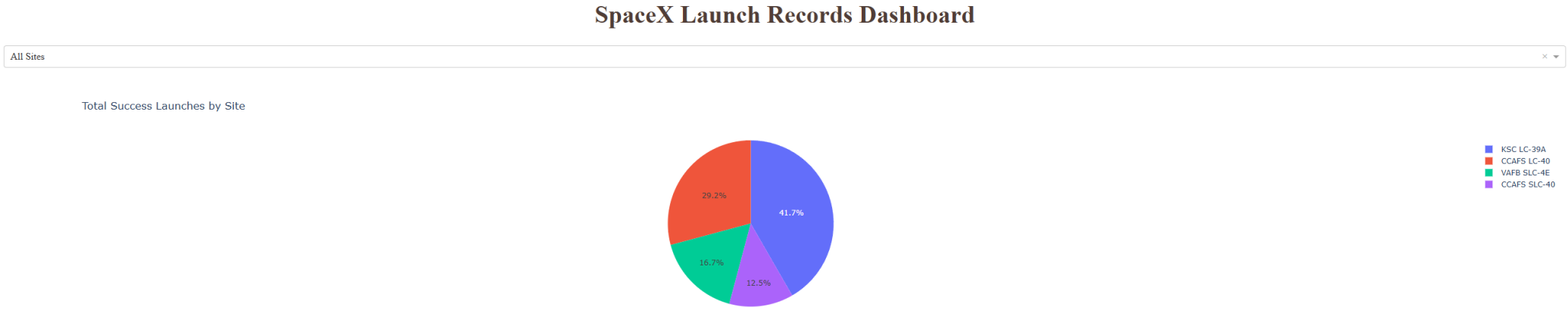
- •Lets users filter launches shown in the scatter plot by payload mass.
- • Purpose: Enables exploration of how launch outcomes vary across different payload weights.

Why These Were Added

•Allow real-time exploration of success patterns by site and payload.
•Help identify optimal launch conditions based on historical performance.

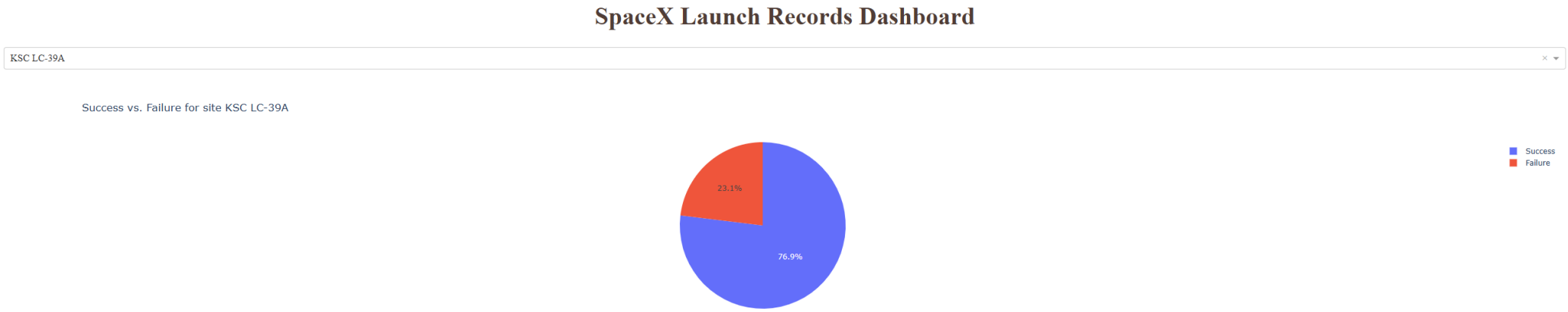
•Enable stakeholders to make data-driven decisions about future launch planning and resource allocation.

Git URL-
https://github.com/BrettDaff/Capstone/blob/main/Dash.py



All site success

- According to this data KSC LC 39A Is the most successful launch site
- CCAFS SLC-40 is the least successful Launch site

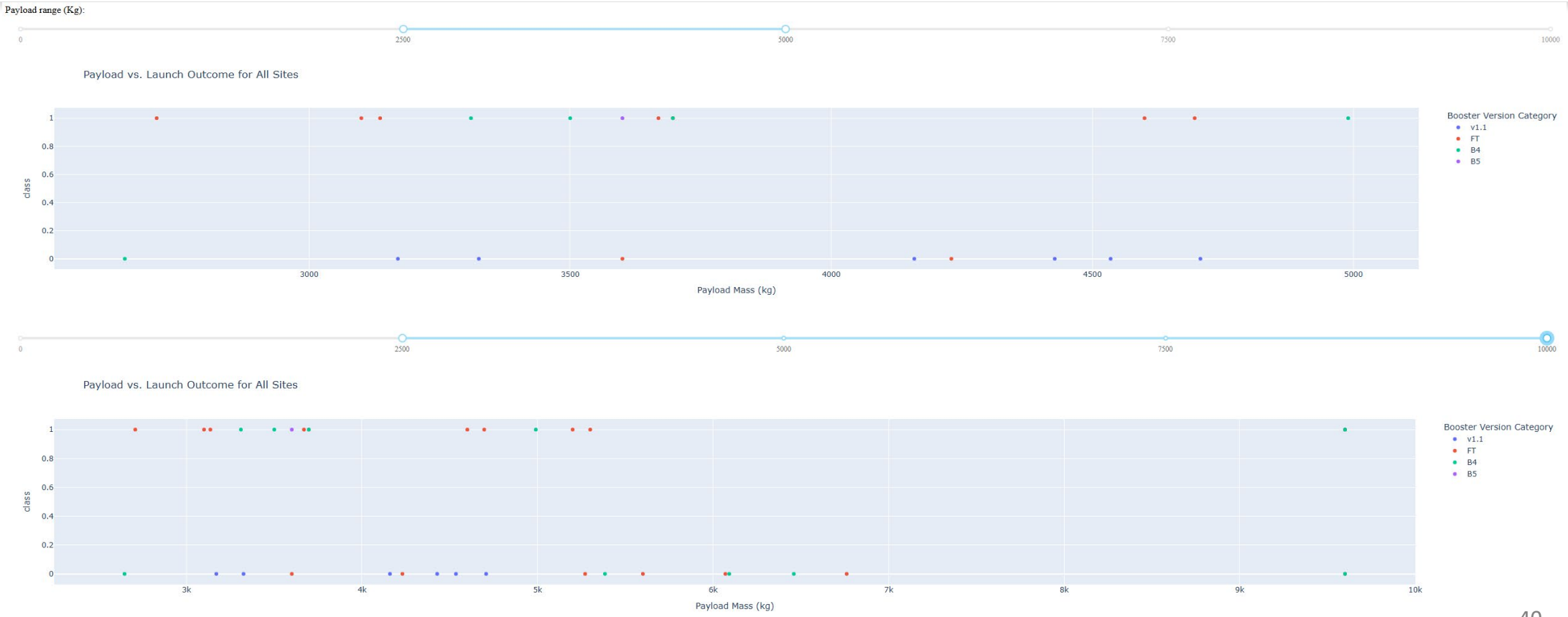


KSC LC-39A success rate

- KSC LC-39A Completes a successful launch nearly 80% of the time

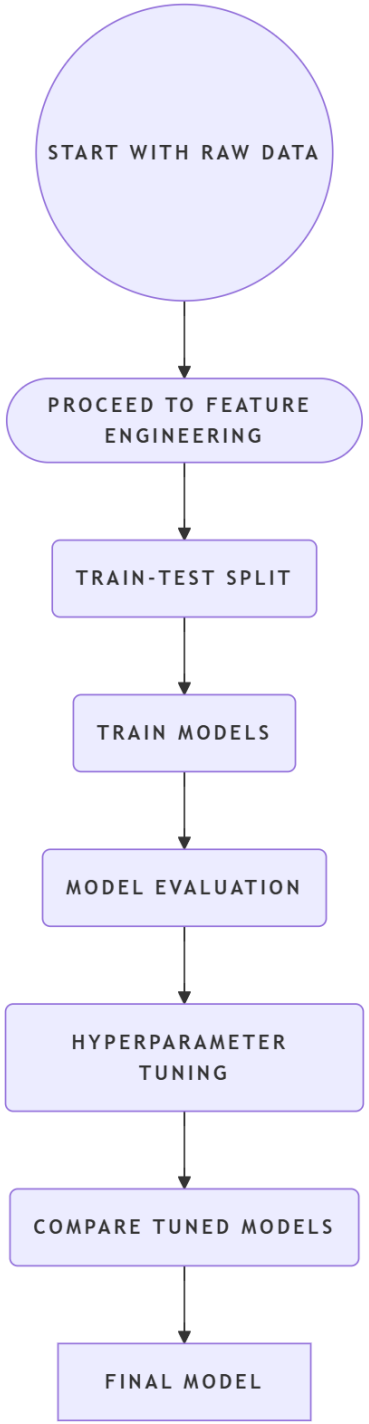
Launch success payload and booster type

- FT seems to do well between 2500 and 5500kg
- B4 does well between 3000 and 5000kg but any heavier or lighter and it begins to perform poorly
- V1.1 seems to preform poorly regardless of weight
- B5 seems to have one launch in the selected ranges and it was successful



Section 5

Predictive Analysis (Classification)



Predictive Analysis (Classification)

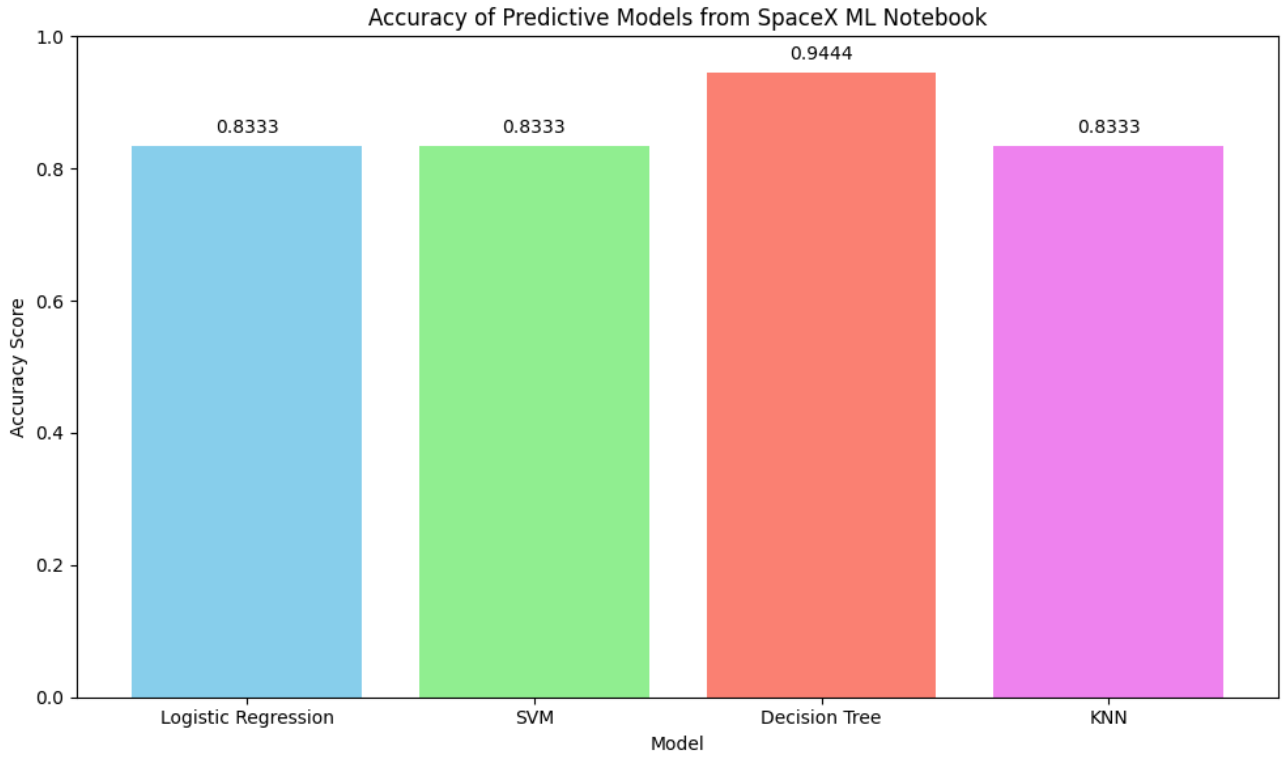
- **Model Development Summary Data Preparation**
  - Applied feature scaling and train-test split (80/20).
  - Encoded categorical features using One-Hot Encoding.
- **Model Building**
  - Built multiple classification models:
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Decision Tree
    - K-Nearest Neighbors (KNN)
- **Model Evaluation**
  - Used accuracy, confusion matrix, and classification report.
  - Evaluated each model's performance on the test set.
- **Model Tuning & Improvement**
  - Tuned hyperparameters using GridSearchCV (e.g., C, gamma, max_depth, k).
  - Compared default vs. optimized model scores.
- **Best Model Selection**
  - Chose the model with the highest accuracy after tuning.
  - In this notebook, the decision tree model with optimized parameters performed best.

Git URL-

https://github.com/BrettDaff/Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

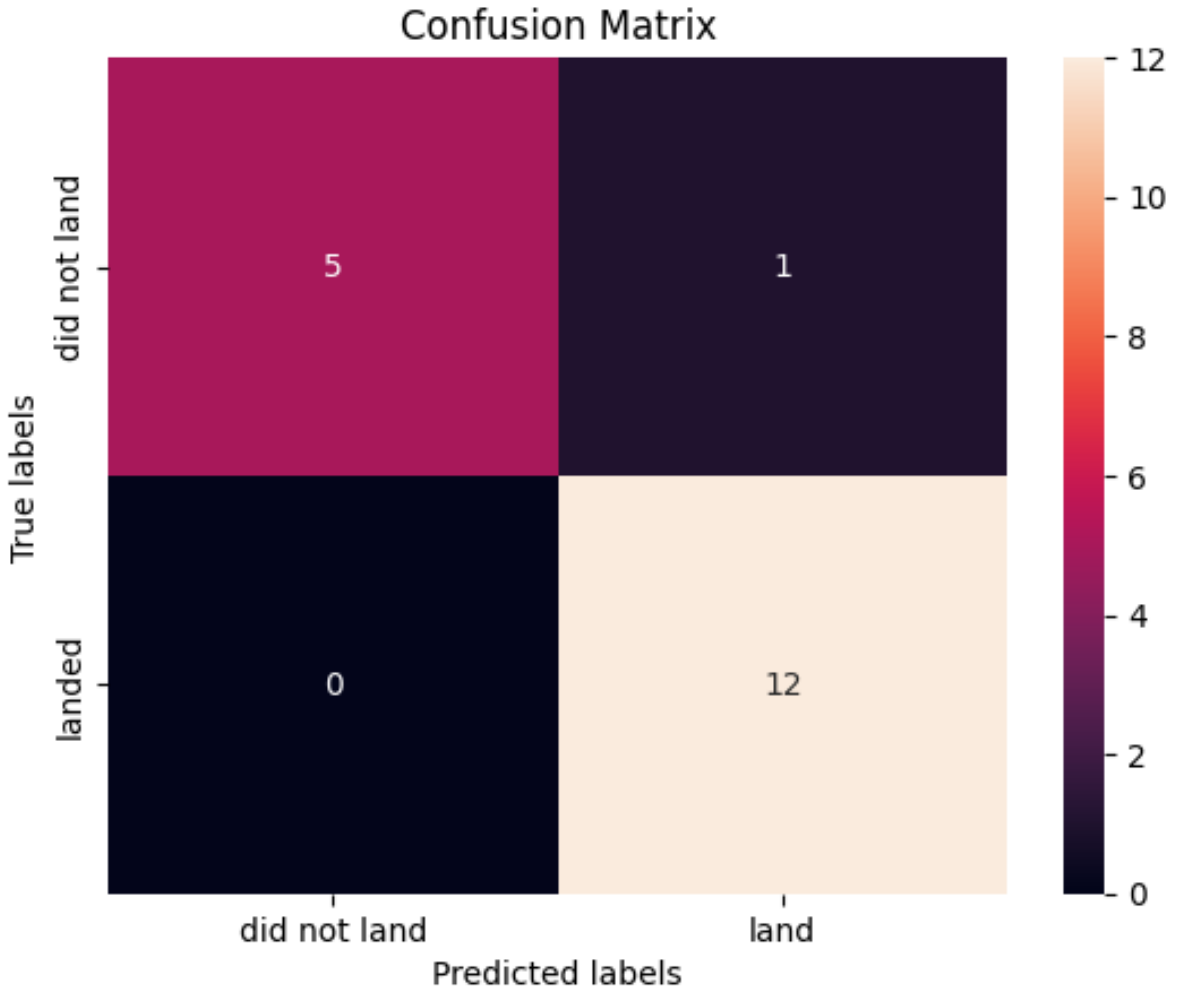
Classification Accuracy

- Each model has an accuracy rate above 80%
- The Logistic Regression, Support Vector Machine(SVM), and Knn models seem to have performed very well with an 83% accuracy across the board
- The Decision Tree model has performed the best with an 94% accuracy



Confusion Matrix

The confusion matrix of the decision tree model shows only one incorrect prediction out of the total 18. Accurately predicting 5 failures, and 12 successes. With one failed launch predicted as a success.



Results

Model Building & Evaluation

Trained multiple classifiers: Logistic Regression, SVM, Decision Tree, and KNN.

Hyperparameter tuning (GridSearchCV) was applied to improve model performance.

The Decision tree model with tuned parameters achieved the best accuracy on the test data.

Evaluation metrics (accuracy, precision, recall) were used to compare model results.

Final model can predict landing success based on features like orbit, payload, site, and booster type

Conclusions

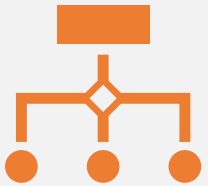
Booster type, payload mass, orbit type, and launch site all play an important role in the success or failure of a launch.

Different booster types are suited better for different payload masses

Different orbit types have a greater level of difficulty involved in a successful landing, based on this particular data some orbit types have not had enough launches to accurately determine their success rate

Using a decision tree model we can predict the success of a launch with 90% accuracy and based on the SpaceX trend of higher success rates coupled with an increase of launches over time, it is likely true that successful launches will become easier to predict and complete over time

Appendix



ALL RELEVANT CODE DATA AND CHARTS CAN BE FOUND IN THE GIT HUB URL PROVIDED:

HTTPS://GITHUB.COM/BRETTDAFF/CAPSTONE.GIT

Thank you!