Winning Space Race with Data Science

Marc THOMAS

November 3rd 2024

https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS

Outline

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

Executive Summary

Summary of Methodologies

1.Data Collection:

- **APIs and Web Scraping**: Used the SpaceX REST API to gather historical launch data, including details on rockets, payloads, launch sites, and landing outcomes. Additionally, used *BeautifulSoup* for web scraping Falcon 9 launch records from Wikipedia to complement the API data.
- **Data Normalization**: Employed *json_normalize* to convert JSON responses from the API into a flat table structure for easy manipulation.
- **Data Filtering and Sampling**: Filtered out unwanted data, such as launches of the Falcon 1, to focus solely on Falcon 9 data relevant to the predictive task.
- **Dealing with Missing Values**: Calculated the mean of PayloadMass and replaced NULL values, leaving LandingPad nulls as-is to be handled with one-hot encoding in later steps.

2.Data Processing and Feature Engineering:

- **Data Cleaning**: Cleaned and standardized data, handling inconsistencies and redundancies to ensure the dataset was suitable for modeling.
- **One-Hot Encoding**: Transformed categorical variables, such as LandingPad, into binary columns to make the dataset compatible with machine learning algorithms.
- **Feature Selection**: Selected relevant columns (features) for predicting the successful landing of Falcon 9's first stage, creating a target column, class, to label successful and unsuccessful landings.

3.Model Development:

- **Machine Learning Models**: Trained multiple models, including Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K-Nearest Neighbors, to classify the landing success of Falcon 9.
- **Hyperparameter Optimization**: Used GridSearchCV to identify optimal parameters for each model, aiming to enhance predictive accuracy.

4.Evaluation and Visualization:

- **Model Accuracy Visualization**: Compared accuracy scores across all models to evaluate performance.
- **Analysis of Predictions**: Assessed each model's performance, noting tendencies to overpredict successful landings, which suggested areas for improvement or the need for additional data.

Executive Summary

Summary of Results

- **Model Performance**:
  - All four machine learning models achieved comparable accuracy scores, approximately **83.33%**. This result indicates moderate predictive success but also highlights areas for improvement.
  - **Overprediction of Successful Landings**: Models displayed a trend toward overpredicting landing success, potentially due to imbalanced data or the limited scope of certain features.
- **Data Quality and Insights**:
  - The process of filtering and cleaning data, particularly handling null values and converting categorical variables, was crucial for developing viable models. The choice to handle PayloadMass nulls with mean values improved dataset integrity.
  - **Impact of Data Quantity and Quality**: Results suggested that more extensive or diverse data may enhance model accuracy and reduce bias in predictions, particularly in balancing predictions between successful and unsuccessful landings.
- **Business Application**:
  - The resulting model predictions and accuracy visualizations provide Space Y with a foundational tool to assess and predict Falcon 9 first-stage landings.
  - **Cost Implications**: By understanding the likelihood of a successful landing, Space Y can better estimate launch costs and potentially gain competitive insights against SpaceX.

This project sets a framework for continuous improvement in predicting reusable rocket stage landings, aligning closely with the strategic cost-reduction goals in commercial spaceflight.

Introduction

Project Background and Context

- The commercial space industry is expanding rapidly, with companies making space travel more affordable through advancements in reusable rocket technology.
- **SpaceX** stands out for its cost-effective Falcon 9 rocket, reducing launch expenses to about $62 million due to the reuse of its first stage, compared to $165 million for traditional, non-reusable rockets.
- Not every mission enables Falcon 9's first stage recovery; factors like payload weight and orbital requirements can lead to its sacrifice.
- This project positions us as a data scientist for **Space Y**, a fictional competitor aiming to develop cost-effective, reusable rockets.
- Using SpaceX’s historical launch data, the goal is to build a machine learning model that predicts first-stage landing success, giving Space Y insights into launch costs and planning.

Introduction

Problems to be Addressed

1. **Can we predict the success of a Falcon 9 first-stage landing?**
   - Given the significant cost advantage provided by reusable rockets, accurately predicting the success of a first-stage landing is critical. By determining the likelihood of a successful landing based on launch characteristics, Space Y can assess and strategize its own reusability efforts.
2. **What are the main factors influencing the landing success of the Falcon 9 first stage?**
   - Understanding which variables (e.g., payload weight, launch site, orbit type) correlate most strongly with landing success can inform Space Y's launch planning and design priorities for reusability. This insight can help prioritize data collection and refinement in future models.
3. **How accurate are machine learning models in predicting first-stage landing outcomes?**
   - Using various machine learning models (Logistic Regression, SVM, Decision Tree, and K-Nearest Neighbors), we aim to evaluate which model provides the most reliable predictions. Optimizing these models with GridSearchCV allows for an in-depth comparison to determine which methodology works best for this application.
4. **To what extent does data quality and completeness affect predictive accuracy?**
   - Given the trend of overpredicting successful landings, the project will explore how data gaps, imbalanced classes, or other quality issues may influence model accuracy. This analysis will provide insights into how additional or higher-quality data might improve predictions.
5. **How can the insights gained from this analysis support Space Y's competitive strategy?**
   - By creating interactive dashboards and visualizing model outcomes, we aim to present data-driven insights that Space Y can use for strategic decision-making in competing with SpaceX. Accurate predictions can guide Space Y in launch cost estimations, bidding for contracts, and optimizing its operational strategy for reusable rockets.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We used the SpaceX REST API to gather historical launch data, including details on rockets, payloads, launch sites, and landing outcomes.
  - We used the BeautifulSoup for web scraping Falcon 9 launch records from Wikipedia to complement the API data.
- Perform data wrangling
  - We selected relevant columns (features) for predicting the successful landing of Falcon 9's first stage, creating a target column, class, to label successful and unsuccessful landings.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Hyperparameter Optimization: Used GridSearchCV to identify optimal parameters for each model, aiming to enhance predictive accuracy.

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia.

For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas `dataframe` using `json_normalize()`. We then cleaned the data, checked for missing values and fill with whatever needed.

For web scrapping, we will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas `dataframe` for further analysis.

Data Collection

To initiate our analysis, we employed a multi-faceted approach to data collection, leveraging both REST API and web scraping techniques.

REST API Data Acquisition:

1. Data Fetching: We utilized HTTP GET requests to extract relevant data from the specified API endpoint.

2. Data Parsing: The raw JSON response was parsed and transformed into a structured Pandas DataFrame using the json_normalize() function.

3. Data Cleaning and Imputation: The DataFrame underwent rigorous cleaning, including handling missing values through appropriate imputation strategies.

Web Scraping Data Acquisition:

1. HTML Parsing: The BeautifulSoup library was employed to parse HTML tables containing launch records from Wikipedia.

2. Data Extraction: Table data was extracted and converted into a Pandas DataFrame for subsequent analysis.

By combining these methods, we successfully acquired and prepared a comprehensive dataset ready for in-depth exploration and modeling.

Data Collection – SpaceX API

This workflow ensures that the SpaceX data is cleaned, filtered, and structured for machine learning model training.

Request Data: Access SpaceX launch data through their APIs, which provide detailed information on each launch.

Combine JSON with Lists: Retrieve data in JSON format, supplemented by lists for key details like launch site, booster version, and payload.

Normalize JSON Data: Use json_normalize to transform the hierarchical JSON data into a flat format suitable for analysis.

Select Relevant Data: Extract key attributes necessary for analysis, such as launch details and landing outcomes, and organize them in a dictionary.

Convert to DataFrame: Convert the dictionary into a structured Pandas DataFrame for further data manipulation.

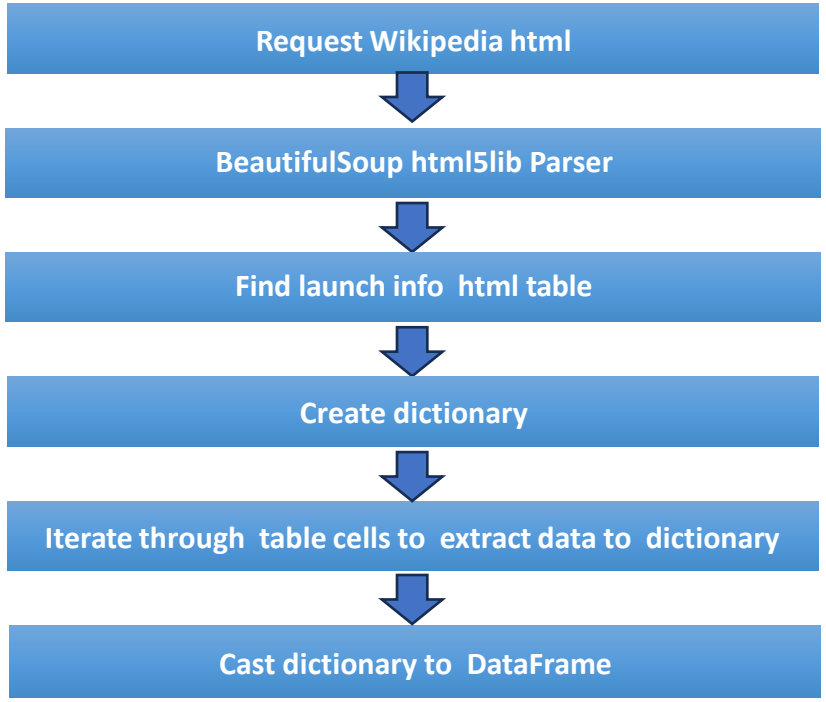
Filter for Falcon 9: Limit the dataset to only Falcon 9 launches by excluding records from other rocket types (e.g., Falcon 1).

Handle Missing Values: For columns like PayloadMass, filling missing values by calculating and replacing with the mean, ensuring data completeness for analysis.

GitHub URL of the completed SpaceX API calls notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_1/jupyter-labs-spacex-data-collection-api_EXECUTED.ipynb

Data Collection - Scraping

This process describes how web scraping is used to transform raw HTML data from Wikipedia into a structured format suitable for data analysis.



Request Wikipedia HTML: Initiate a request to retrieve the HTML content of the relevant Wikipedia page containing SpaceX launch data.

BeautifulSoup HTML5lib Parser: Use BeautifulSoup with the html5lib parser to process the retrieved HTML content, allowing structured extraction of data from the page.

Find Launch Info HTML Table: Locate the specific HTML table in the Wikipedia page that contains the desired launch information.

Create Dictionary: Initialize an empty dictionary to store the extracted data in a structured format, with keys representing different data fields.

Iterate Through Table Cells to Extract Data to Dictionary: Loop through each cell in the table, pulling out relevant information and populating the dictionary with key-value pairs.

Cast Dictionary to DataFrame: Convert the populated dictionary into a DataFrame for easier analysis and manipulation, making it compatible with data processing and machine learning tasks.

GitHub URL of the completed web scraping notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_1/jupyter-labs-webscraping_EXECUTED.ipynb

Data Wrangling

To prepare our data for in-depth analysis and modeling, we embarked on a comprehensive data wrangling process:

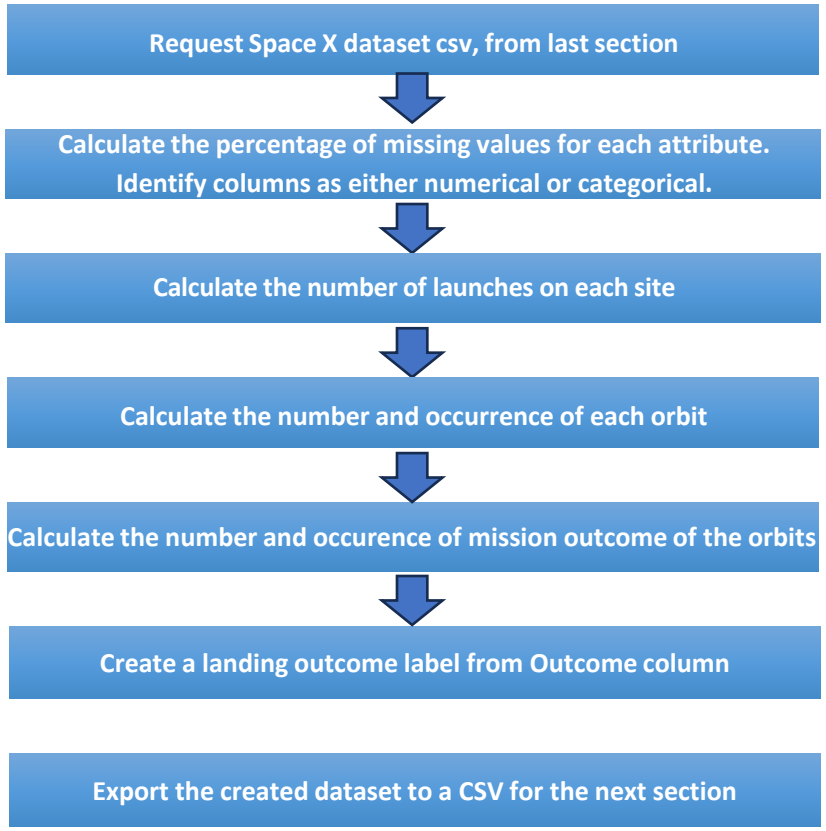
1. **Data Cleaning and Unification:** We meticulously cleaned and standardized the dataset, addressing inconsistencies and ensuring data integrity.
2. **Exploratory Data Analysis (EDA):**
   - **Launch Site Analysis:** We calculated the number of launches per launch site to identify trends and patterns.
   - **Mission Outcome Analysis:** We analyzed the frequency and types of mission outcomes across different orbit types.
   - **Landing Outcome Labeling:** We created a new categorical variable, "landing outcome," derived from the "outcome" column. This simplified subsequent analysis, visualization, and machine learning tasks.

The final, cleaned dataset was exported to a CSV file for further exploration and modeling.

GitHub URL of the completed data wrangling notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_1/labs-jupyter-spacex-Data%20wrangling_EXECUTED.ipynb

Data Wrangling

This process describes how Data Wrangling is used to transform Space X dataset, from last section into a structured format suitable for data analysis.



Request SpaceX Dataset CSV: Load the SpaceX dataset from a previously saved section.

Calculate Missing Values and Identify Column Types: Determine the percentage of missing values in each attribute and classify columns as numerical or categorical.

Calculate Launches per Site: Count the number of launches at each launch site.

Calculate Orbits: Count the occurrences of each orbit type.

Calculate Mission Outcomes per Orbit: Count the occurrences of each mission outcome for each orbit type.

Create Landing Outcome Label: This variable will represent the classification variable that represents the outcome of each launch

Export Dataset: Save the processed dataset as a CSV file for the next stage.

GitHub URL of the completed data wrangling notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_1/labs-jupyter-spacex-Data%20wrangling_EXECUTED.ipynb

EDA with Data Visualization

To gain insights into the underlying relationships between key variables, different visualization techniques to compare relationships between variables to decide if a relationship exists so that they could be used in training the machine learning model :

Visualization:

- **Scatter plots:** Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes.
- **Bar Charts:** We employed bar charts to visualize the distribution of mission outcomes across different orbit types. This allowed us to identify orbits with higher success rates.
- **Line Plots:** Line plots were used to track trends in launch success over time. By visualizing yearly launch success rates, we can identify potential patterns or anomalies.

Feature Engineering:

- To prepare the data for predictive modeling, we performed feature engineering:
- **Categorical Variable Encoding:** We converted categorical variables (such as launch site, orbit type, and landing outcome) into numerical representations using one-hot encoding. This step is crucial for machine learning algorithms, which typically work with numerical data.

By combining these techniques, we aimed to extract meaningful insights and create a robust dataset for future predictive modeling.

GitHub URL of the completed EDA with Data Visualization notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_2/edadataviz_EXECUTED_V2.ipynb

EDA with Data Visualization

To gain insights into the underlying relationships between key variables we did this set of data analysis:

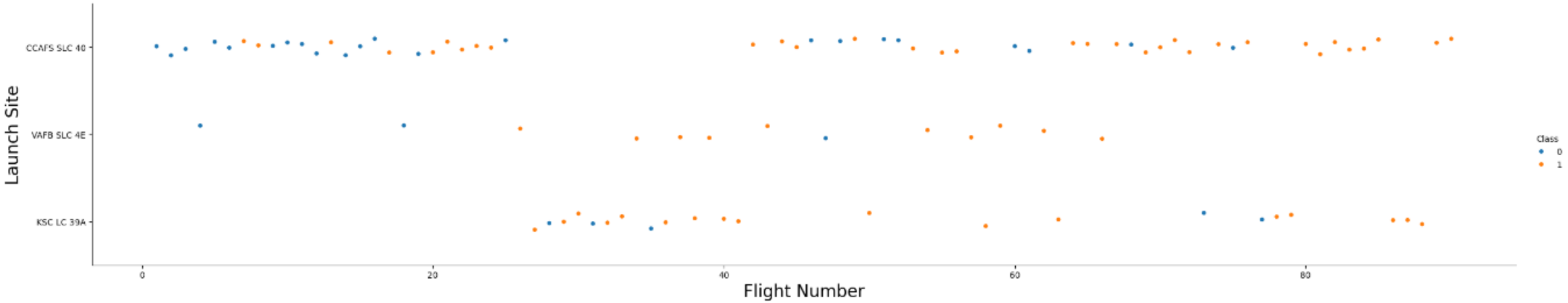
- **Flight Number vs. Launch Site (scatter plots):** We analyzed the distribution of launches across different launch sites over time.
- **Payload Mass vs. Launch Site (scatter plots):** We investigated the relationship between payload mass and launch site.
- **Success rate vs. Orbit type (bar chart):** We studied the relationship between success rate and the target orbit.
- **Payload Mass vs. Flight Number (scatter plots):** We explored the correlation between the mass of the payload and the mission sequence number.
- **Flight Number vs. Orbit Type (scatter plots):** We examined the evolution of orbit types over the course of the missions.
- **Payload vs. Orbit Type (scatter plots):** We studied the relationship between payload mass and the target orbit.
- **The launch success yearly trend**

By visualizing these relationships, we were able to identify potential factors influencing landing outcomes. These insights will be crucial for feature engineering and model development in subsequent steps.

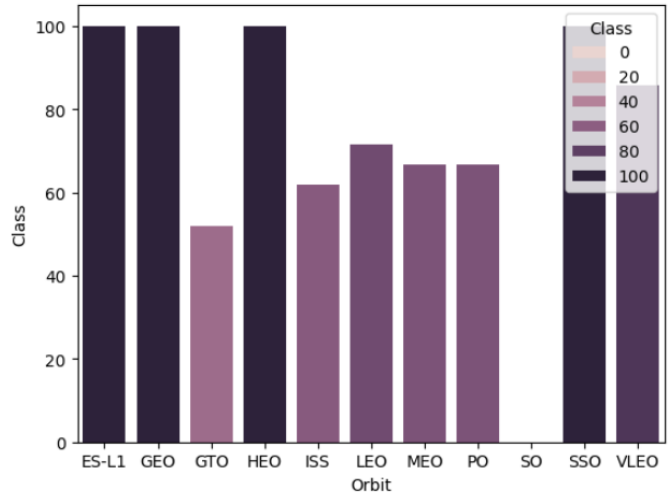
GitHub URL of the completed EDA with Data Visualization notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_2/edadataviz_EXECUTED_V2.ipynb

EDA with Data Visualization (Examples)

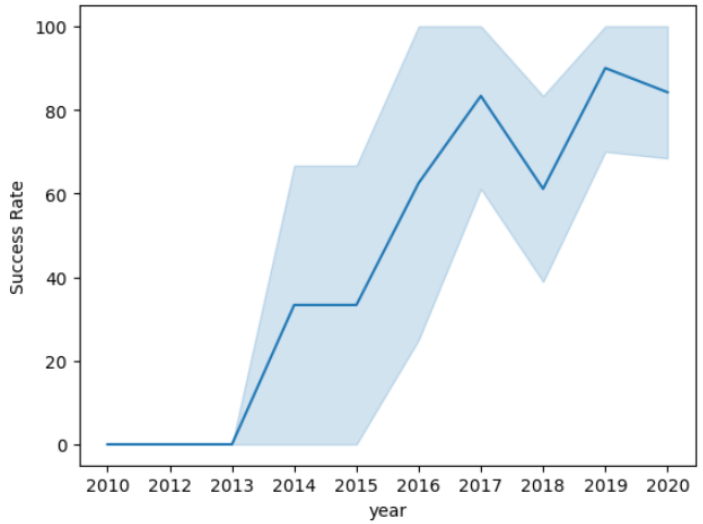
The relationship between Flight Number and Launch Site



The relationship between success rate of each orbit type



Visualization of the launch success yearly trend



GitHub URL of the completed EDA with Data Visualization notebook:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_2/edadataviz_EXECUTED_V2.ipynb

EDA with SQL

To facilitate data exploration and analysis, we integrated the dataset into an IBM DB2 database. This centralized repository allowed us to efficiently query and extract relevant information using SQL.

We executed a series of SQL queries to gain a comprehensive understanding of the dataset:

- **Launch Site Analysis:** We identified unique launch sites and their corresponding mission outcomes.
- **Payload Analysis:** We explored the distribution of payload sizes across different customers and missions.
- **Booster Analysis:** We analyzed the various booster versions used in the launches.
- **Landing Outcome Analysis:** We investigated the frequency and types of landing outcomes.

By leveraging SQL's powerful querying capabilities, we were able to uncover valuable insights from the data.

GitHub URL of the completed EDA with SQL notebook,
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_2/jupyter-labs-eda-sql-coursera_sqllite_EXECUTED.ipynb

EDA with SQL

To delve deeper into the dataset and extract valuable insights, we employed a series of SQL queries:

1. Launch Site Exploration:
   1. Identified unique launch site names.
   2. Filtered for launch sites beginning with "CCA".

2. Payload Analysis:
   1. Calculated the total payload mass carried by NASA CRS missions.
   2. Determined the average payload mass of F9 v1.1 boosters.

3. Landing Outcome Analysis:
   1. Identified the date of the first successful ground pad landing.
   2. Filtered for boosters with successful drone ship landings and specific payload mass ranges.
   3. Analyzed the total number of successful and failed missions.
   4. Identified booster versions carrying the maximum payload mass.
   5. Analyzed failed drone ship landings in 2015, including booster versions and launch sites.
   6. Ranked landing outcomes between specific dates in descending order (2010-06-04 and 2017-03-20)

These SQL queries provided a solid foundation for subsequent data visualization and modeling efforts.

GitHub URL of the completed EDA with SQL notebook,
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_2/jupyter-labs-eda-sql-coursera_sqllite_EXECUTED.ipynb

Build an Interactive Map with Folium

To provide a visual representation of the launch data and explore spatial relationships, we employed the following techniques:

Interactive Map Visualization:

1.Geocoding Launch Sites: We utilized latitude and longitude coordinates to plot circle markers on an interactive map, representing each launch site.

2.Visualizing Launch Outcomes: We categorized launch outcomes (success and failure) as 0 and 1, respectively. These categories were then mapped to different marker colors (**green for success**, **red for failure**) on the map, using MarkerCluster() for efficient visualization.

Spatial Analysis:

1.Distance Calculations: We employed the Haversine formula to calculate the distance between launch sites and various landmarks, including:

1.Railways 2.Highways 3.Coastlines 4.Nearby Cities

By analyzing these distances, we aimed to answer crucial questions such as:

- ✓ How proximate are launch sites to transportation infrastructure?
- ✓ What is the spatial relationship between launch sites and urban areas?

These spatial insights provide valuable context for understanding the operational constraints and potential environmental impacts of space launches.

GitHub URL of the completed an Interactive Map with Folium:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_3/lab_jupyter_launch_site_location_EXECUTED%20%26%20UPDATED_%2B%2B.ipynb

Build a Dashboard with Plotly Dash

To provide a dynamic and user-friendly exploration of the launch data, we developed an interactive dashboard using Plotly Dash. This dashboard empowers users to delve into the data and uncover insights:

Key Visualizations:

1. **Pie Charts:** We incorporated pie charts to visualize the distribution of launches across different launch sites. This provides a clear overview of the launch activity at each site.
2. **Scatter Plots:** Scatter plots were employed to examine the relationship between mission outcome and payload mass for various booster versions. This visualization helps identify trends and potential correlations between these factors. The scatter plot help us to see how success varies across launch sites, payload mass, and booster version category.

By providing interactive elements and customizable visualizations, this dashboard enables users to ask and answer questions about the data, fostering a deeper understanding of space launch trends and patterns.

GitHub URL of the completed Dashboard with Plotly Dash:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_3/Ploty%20Dash/spacex_dash_app.py

Predictive Analysis (Classification)

Data Preparation and Model Selection:

1.Data Loading: We imported the dataset into NumPy and Pandas for efficient data manipulation.

2.Data Transformation and Splitting: The data was preprocessed and split into training and test sets to prepare for model training and evaluation.

3.Algorithm Selection: We carefully considered the nature of the problem and the characteristics of the dataset to select appropriate machine learning algorithms.

4.Hyperparameter Tuning: We employed GridSearchCV to systematically explore different hyperparameter combinations for each algorithm, optimizing model performance.

Model Evaluation:

1.Performance Metrics: We evaluated the performance of each model using relevant metrics such as accuracy, precision, recall, and F1-score.

2.Hyperparameter Insights: We analyzed the tuned hyperparameters to gain insights into the optimal configuration for each algorithm.

3.Confusion Matrix Analysis: We visualized the confusion matrix to understand the model's classification performance in more detail.

Model Improvement:

1.Feature Engineering: We explored feature engineering techniques to create informative features that could enhance model performance.

2.Algorithm Tuning: We further refined the hyperparameters and explored alternative algorithms to improve model accuracy and generalization.

Model Selection

The final step involved selecting the model with the highest accuracy score and the best overall performance across various metrics. This model will be our preferred choice for making predictions on new, unseen data.

GitHub URL of the completed Predictive Analysis with Python:
https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS/blob/main/Module_4/SpaceX_Machine%20Learning%20Prediction_Part_5_EXECUTED_UPDATED.ipynb

Results

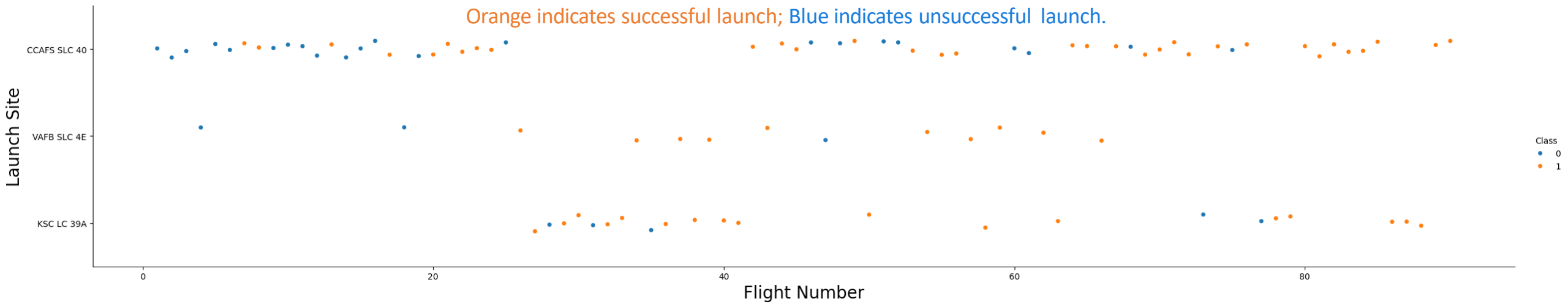
Our analysis will yield three primary outcomes:

- **Exploratory Data Analysis (EDA) Results:** A comprehensive report detailing the key findings from the exploratory analysis, including data quality assessments, statistical summaries, and visualizations.
- **Interactive Analytics Demo:** A series of screenshots showcasing an interactive dashboard, developed using tools like Plotly Dash or Tableau. This dashboard will enable users to explore the data dynamically and gain insights.
- **Predictive Analysis Results:** A detailed report presenting the outcomes of predictive modeling efforts, including model selection, feature engineering, model training, and evaluation. This report will also include performance metrics and insights into the model's predictive capabilities.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

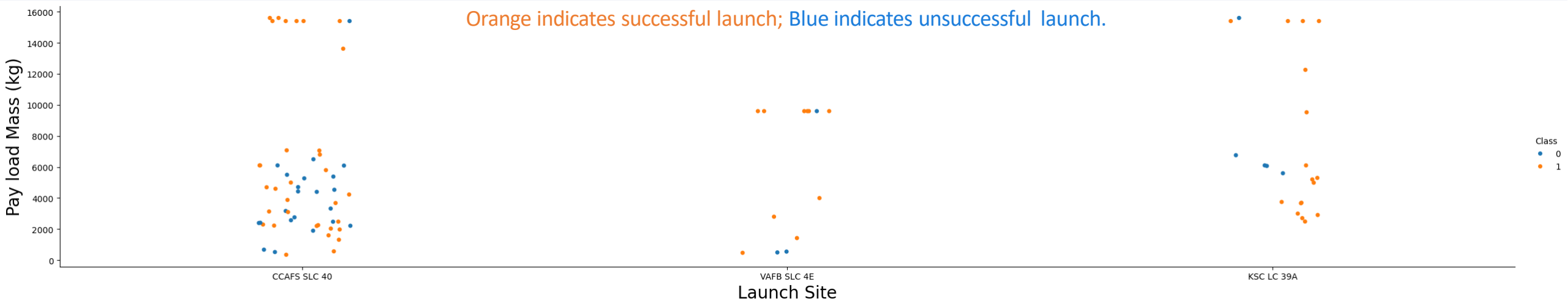


The scatter plot reveals several intriguing trends:

1. **Positive Correlation between Flight Number and Success Rate:** As the number of flights increases for a specific launch site, the overall success rate tends to improve. This suggests that experience and technological advancements play a significant role in mission success.
2. **Significant Breakthrough around Flight 20:** A notable shift in success rates is observed around flight number 20. This could be attributed to a major technological breakthrough or improved operational procedures.
3. **CCAFS SLC40, a Dominant Launch Site:** CCAFS SLC40 emerges as the primary launch site, characterized by the highest volume of launches. However, it exhibits a less pronounced correlation between flight number and success rate, indicating potential factors beyond experience influencing its performance.

Further analysis is required to delve deeper into the specific factors contributing to the varying success rates across different launch sites and flight numbers.

Payload vs. Launch Site

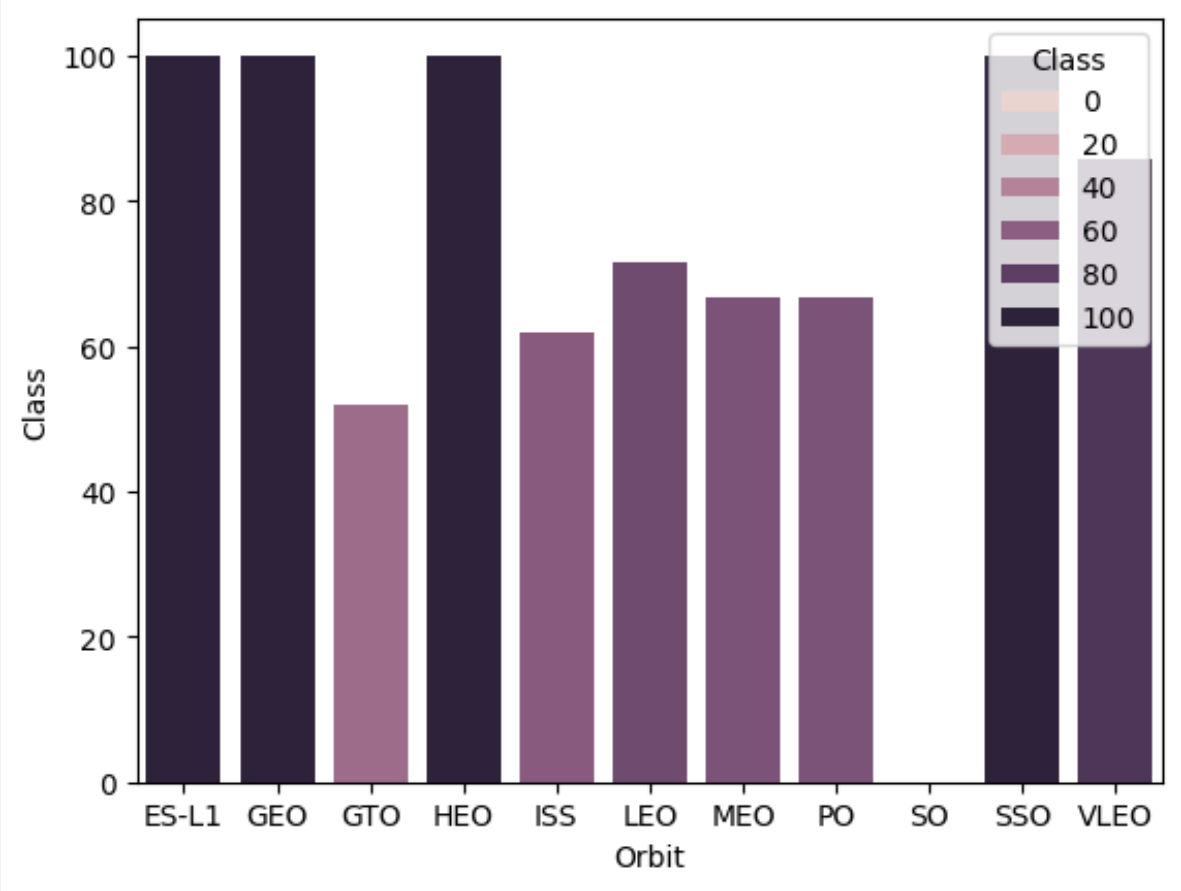


The scatter plot provides valuable insights into the relationship between payload mass and mission success:

1. **Payload Mass Threshold:** A distinct trend emerges: missions with a payload mass exceeding 7000 kg exhibit a significantly higher probability of success. This suggests that technological advancements and operational improvements have enabled the successful launch of heavier payloads.
2. **Launch Site-Specific Payload Preferences:** Different launch sites appear to cater to specific payload mass ranges. This variation could be attributed to factors such as launch vehicle capabilities, infrastructure limitations, and mission objectives.
3. **No Clear Site-Payload-Success Correlation:** While payload mass plays a crucial role in mission success, the scatter plot does not reveal a clear relationship between launch site and payload mass in terms of success rate. This indicates that other factors, such as launch vehicle performance, weather conditions, and mission complexity, likely influence the outcome.

Further analysis, potentially involving statistical tests or machine learning models, is necessary to uncover more nuanced relationships between these variables and their impact on mission success.

Success Rate vs. Orbit Type



The figure suggests a potential correlation between orbit type and landing outcome:

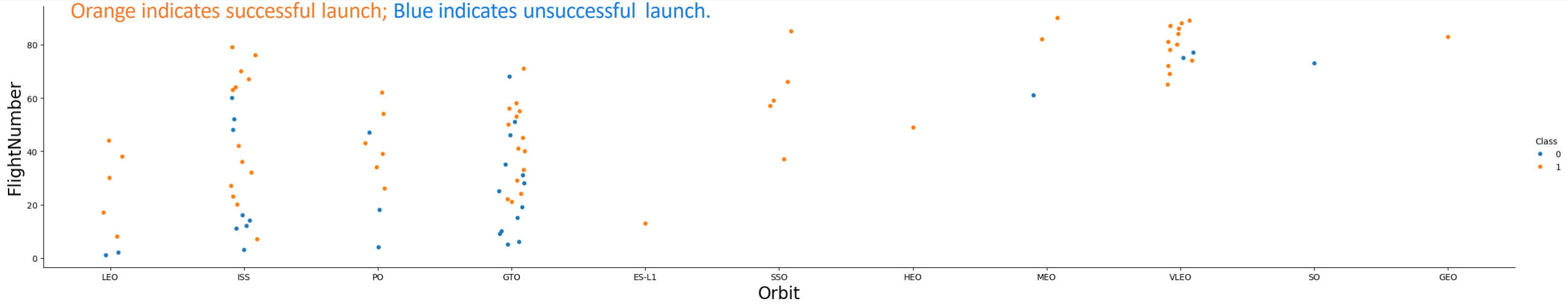
- **High Success Rate Orbits:** Orbits such as SSO, HEO, GEO, and ES-L1 appear to have a 100% success rate in landing.
- **Low Success Rate Orbit:** The SO orbit, on the other hand, seems to have a 0% success rate.
- **Caveats and Further Analysis:**
- It's important to note that the limited sample size for certain orbits, particularly GEO, SO, HEO, and ES-L1, may skew the results. With only one occurrence for each of these orbits, it's difficult to draw definitive conclusions about their impact on landing outcomes.

To gain a more accurate understanding of the relationship between orbit type and landing success, a larger dataset with more diverse and representative samples is necessary. This will enable us to conduct more robust statistical analysis and identify any underlying patterns or trends.

ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis)
SSO (5) has 100% success rate
VLEO (14) has decent success rate and attempts
SO (1) has 0% success rate
GTO (27) has the around 50% success rate but largest sample

Flight Number vs. Orbit Type

Orange indicates successful launch; Blue indicates unsuccessful launch.



The scatter plot provides insights into the relationship between flight number and success rate for different orbits:

1. **Positive Correlation for LEO Orbits:** A clear trend emerges for LEO orbits: as the number of launches increases, the success rate generally improves. This suggests that SpaceX has gained significant experience and refined its launch capabilities for LEO missions.
2. **GTO Orbit Variability:** In contrast, GTO orbits exhibit a more varied pattern, with no clear correlation between flight number and success rate. This could be attributed to the inherent challenges of GTO missions, such as higher energy requirements and more complex trajectory maneuvers.
3. **SpaceX's Focus on Lower Orbits:** SpaceX's initial focus on LEO missions, followed by a recent shift towards VLEO, aligns with the observed success trends. Lower orbits, including LEO and VLEO, appear to be SpaceX's areas of expertise, where they have achieved consistent success.
4. **Potential for Sun-Synchronous Orbits:** The scatter plot also hints at the potential for high success rates in Sun-Synchronous orbits. However, further analysis with a larger dataset is needed to confirm this trend.

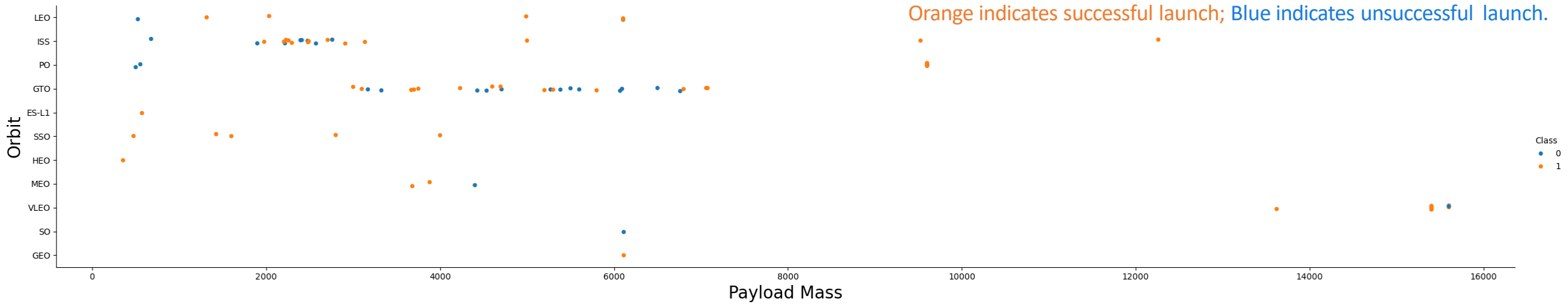
By understanding these orbit-specific trends and SpaceX's strategic focus, we can gain valuable insights into the factors driving mission success and the company's future trajectory.

Cautionary Note:

It's crucial to consider the limited sample size for certain orbits, particularly those with only one occurrence. Such limited data may not accurately represent the true relationship between flight number and success rate for these orbits.

To draw more definitive conclusions, a larger dataset with more diverse and representative samples is essential.

Payload vs. Orbit Type



The analysis of payload mass and orbit type reveals several key findings:

Payload Mass Impact on Orbit Success:

- **Positive Impact:** Heavier payloads have a positive correlation with mission success for LEO, ISS, and P0 orbits. This suggests that SpaceX has optimized its launch vehicles and technologies to handle heavier payloads effectively for these orbits.
- **Negative Impact:** Conversely, heavier payloads seem to negatively impact mission success for MEO and VLEO orbits. This could be due to factors such as the specific requirements of these orbits, the limitations of launch vehicles, or the complexity of the missions.
- **No Clear Correlation:** GTO orbits exhibit no discernible relationship between payload mass and success rate. This could be attributed to the diverse nature of GTO missions, which involve a wide range of payload types and launch vehicle configurations.

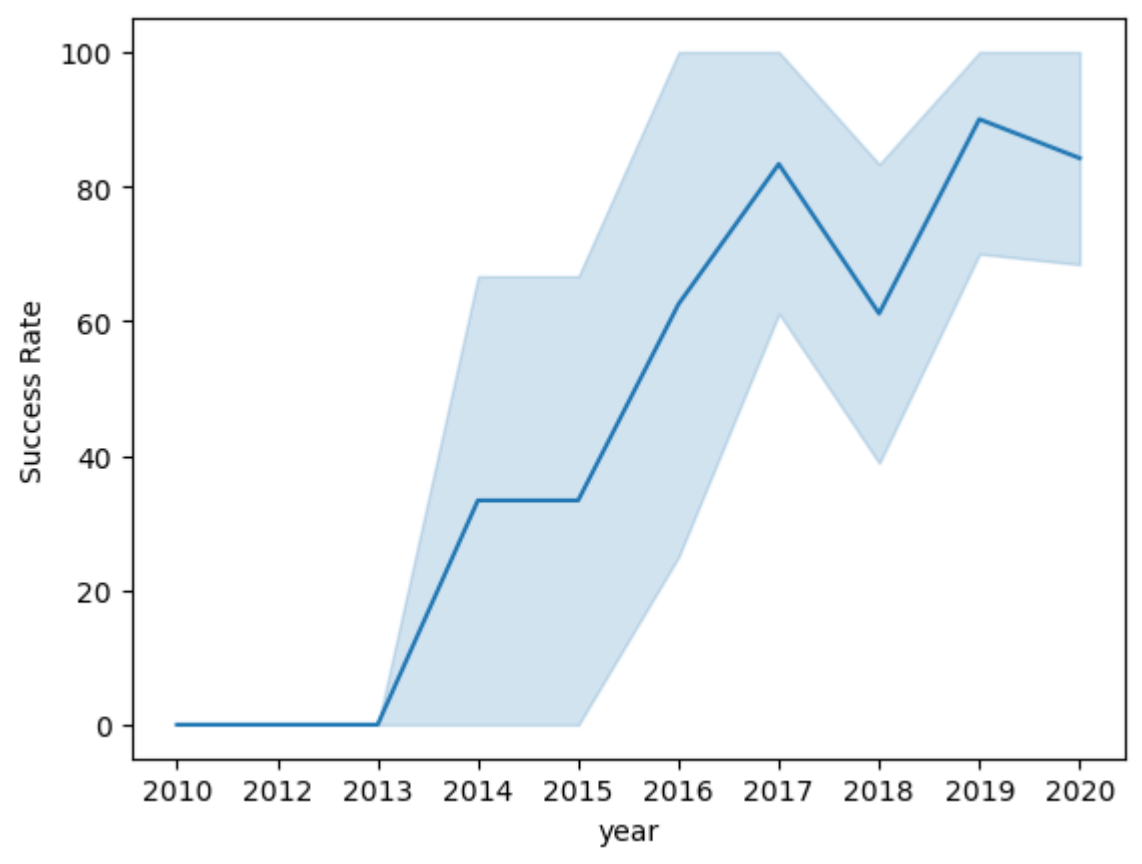
Orbit-Specific Payload Trends:

- **LEO and SSO:** These orbits typically involve lighter payloads, aligning with their lower altitude and mission objectives.
- **VLEO:** The successful VLEO missions tend to involve heavier payloads, indicating SpaceX's capability to launch substantial payloads into these orbits.

Data Limitations and Future Analysis:

It's important to note that the limited dataset for orbits like SO, GEO, and HEO hinders a definitive analysis of payload mass impact.
Further analysis with a larger dataset is necessary to draw more conclusive insights.
By understanding these orbit-specific payload trends, we can gain valuable insights into SpaceX's launch capabilities, mission priorities, and the factors influencing mission success.

Launch Success Yearly Trend



Key Insights from the Launch Success Trend:

- **Positive Trend:** A consistent increase in launch success rates from 2013 to 2020.
- **Potential for Near-Perfect Success:** Continued improvement suggests a future of near-perfect launch success.
- **Temporary Setback in 2018:** A slight dip in success rates around 2018, possibly due to specific challenges.
- **Strong Recent Performance:** A consistently high success rate of approximately 80% in recent years.

All Launch Site Names

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
q = pd.read_sql('select distinct Launch_Site from spacextbl', con)
q
```

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with `CCA`

```
q = pd.read_sql("select * from spacextbl where Launch_Site like 'CCA%' limit 5", con)
q
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcor
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachu
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachu
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attem
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attem
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attem

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below:

```
q = pd.read_sql("select sum(PAYLOAD_MASS__KG_) from spacextbl where Customer='NASA (CRS)'", con)
q
```

	sum(PAYLOAD_MASS__KG_)
0	45596

CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

Average Payload Mass by F9 v1.1

This query calculates the average payload mass or launches which used booster version F9 v1.1

Average payload mass of F9 1.1 is on the low end of our payload mass range

```
q = pd.read_sql("select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version='F9 v1.1'", con)
q
```

	avg(PAYLOAD_MASS__KG_)
0	2928.4

First Successful Ground Landing Date

We use the min() function to find the result.

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
q = pd.read_sql("SELECT min(Date) FROM spacextbl WHERE `Landing_Outcome` = 'Success (ground pad)'", con)
q
```

	min(Date)
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
q = pd.read_sql("select distinct Booster_Version from spacextbl where `Landing_Outcome`='Success (drone ship)' and `PAYLOAD_MASS__KG_` between 4000 and 6000", con)
q
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

This query returns a count of each mission outcome.

SpaceX appears to achieve its mission outcome nearly 99% of the time.

This means that most of the landing failures are intended.

Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

```
q = pd.read_sql("select substr(Mission_Outcome,1,7) as Mission_Outcome, count(*) from spacextbl  group by 1", con)
q
```

	Mission_Outcome	count(*)
0	Failure	1
1	Success	100

Boosters Carried Maximum Payload

This query returns the booster versions that carried the highest payload mass of 15600 kg.

These booster versions are very similar and all are of the F9 B5 B10xx.x variety.

This likely indicates payload mass correlates with the booster version that is used.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
q = pd.read_sql("select distinct Booster_Version from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)", con)
q
```

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

2015 Launch Records

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

We used a combinations of the WHERE SUBSTR clause, CASE, and AND.

There were two such occurrences.

```
q = pd.read_sql("""
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_Name,
    Booster_Version,
    Launch_Site,
    `Landing_Outcome`
FROM spacextbl
WHERE SUBSTR(Date, 0, 5) = '2015'
AND `Landing_Outcome` = 'Failure (drone ship)';
""", con)

q
```

	Month_Name	Booster_Version	Launch_Site	Landing_Outcome
0	January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

We selected Landing outcomes and the COUNT of landing outcomes from the data and

used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.

We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

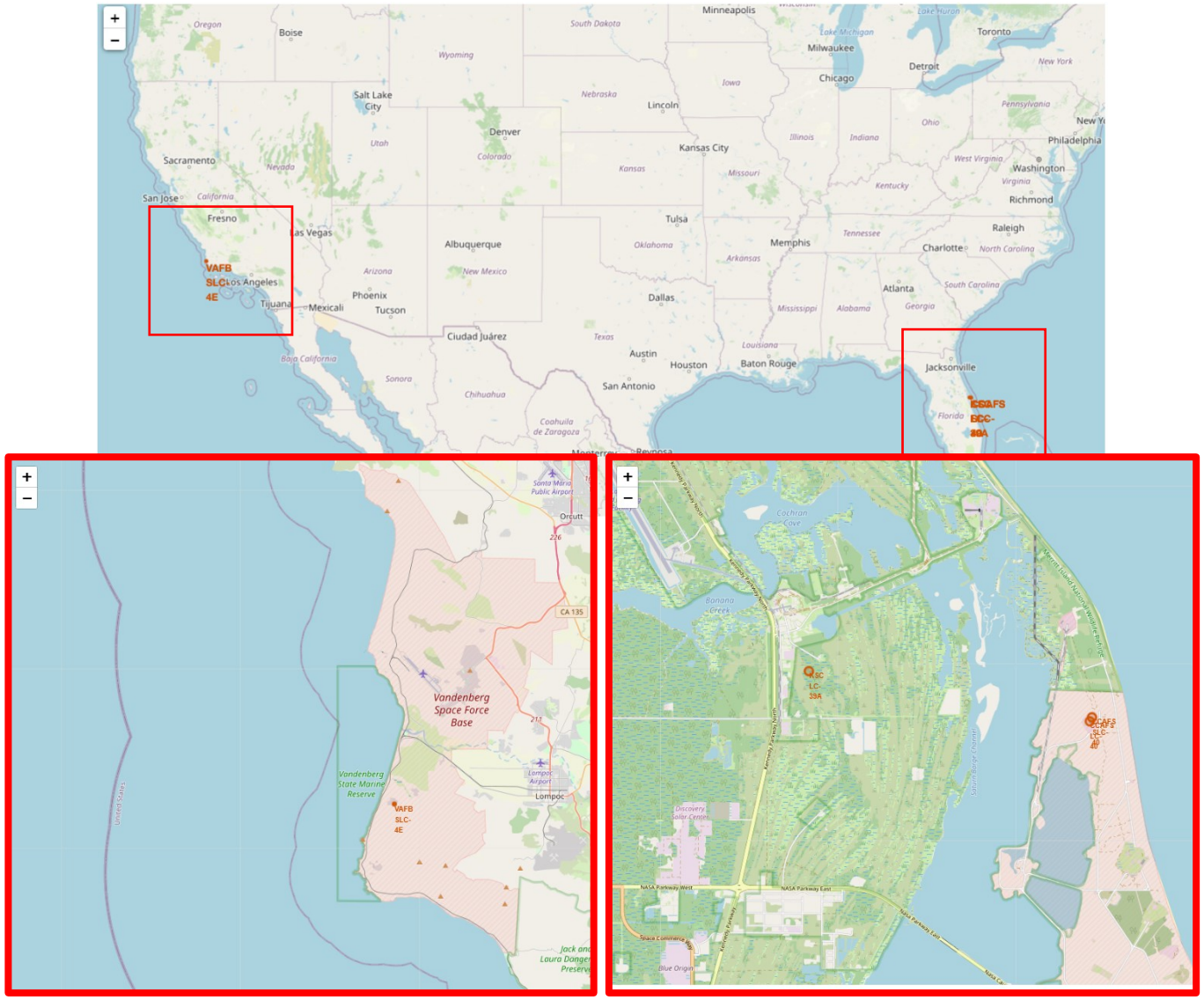
```
q = pd.read_sql("""
    SELECT
        Landing_Outcome,
        COUNT(*) AS Outcome_Count
    FROM spacextbl
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY Outcome_Count DESC;
""", con)
q
```

	Landing_Outcome	Outcome_Count
0	No attempt	10
1	Success (drone ship)	5
2	Failure (drone ship)	5
3	Success (ground pad)	3
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Failure (parachute)	2
7	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Location of all the Launch Sites



We can see that all the SpaceX launch sites are located inside the United States.

The right map shows the two Florida launch and the left map shows the California launch site.

Findings Summary

- **Equator Proximity:**

  SpaceX launch sites are not close to the Equator but are strategically located in the U.S. for logistical reasons.

- **Coastal Proximity:**

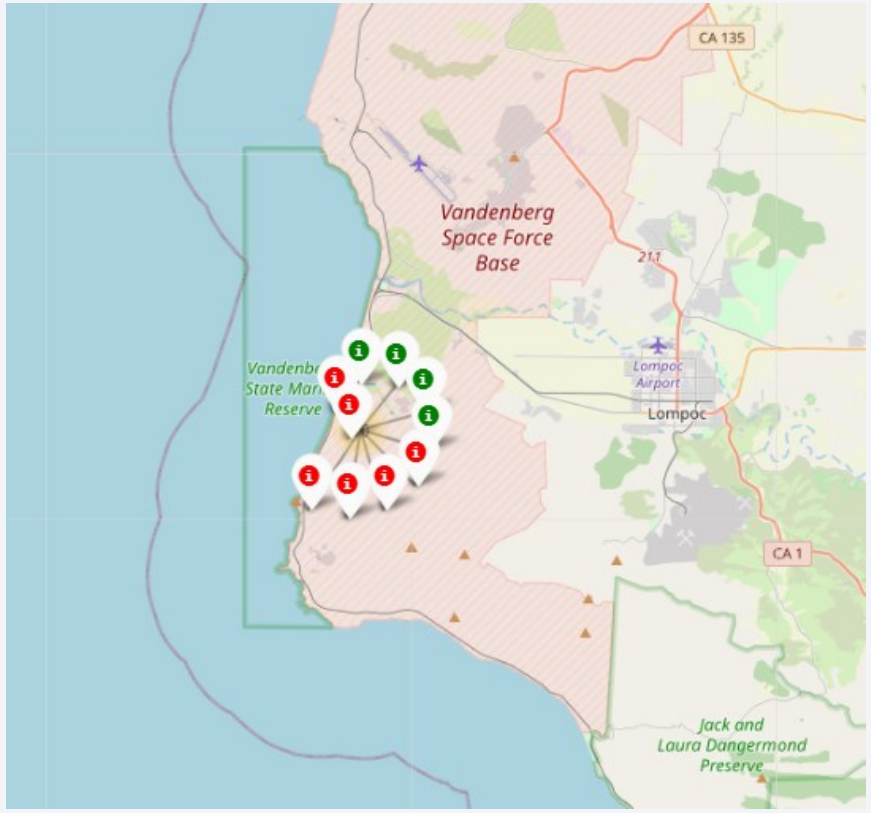
  We can observe that all launch sites are near the ocean.

  In fact all SpaceX launch sites are in very close proximity to the coast to ensure safety and to facilite efficient launch trajectories.

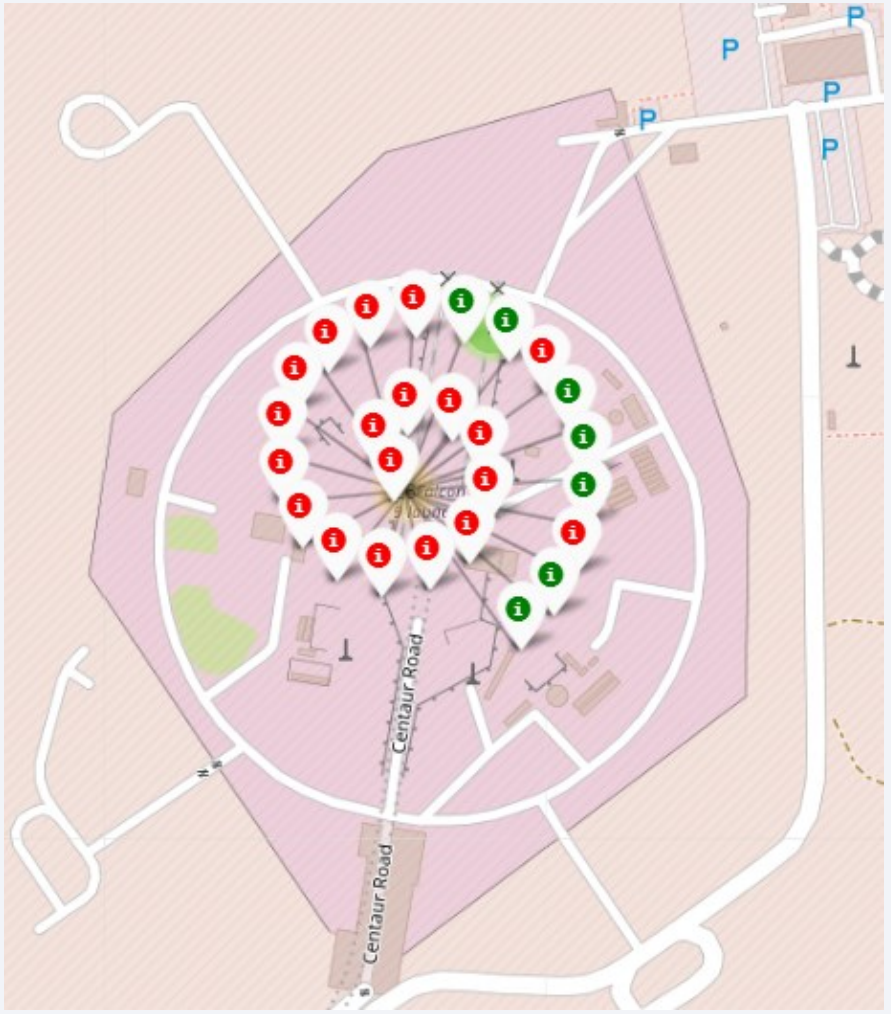
Color-Coded Launch Markers

Clusters on Folium map can be clicked on to display :

- successful landing (green icon)
- failed landing (red icon).



VAFB SLC-4E shows 4 successful landings and 6 failed landings.



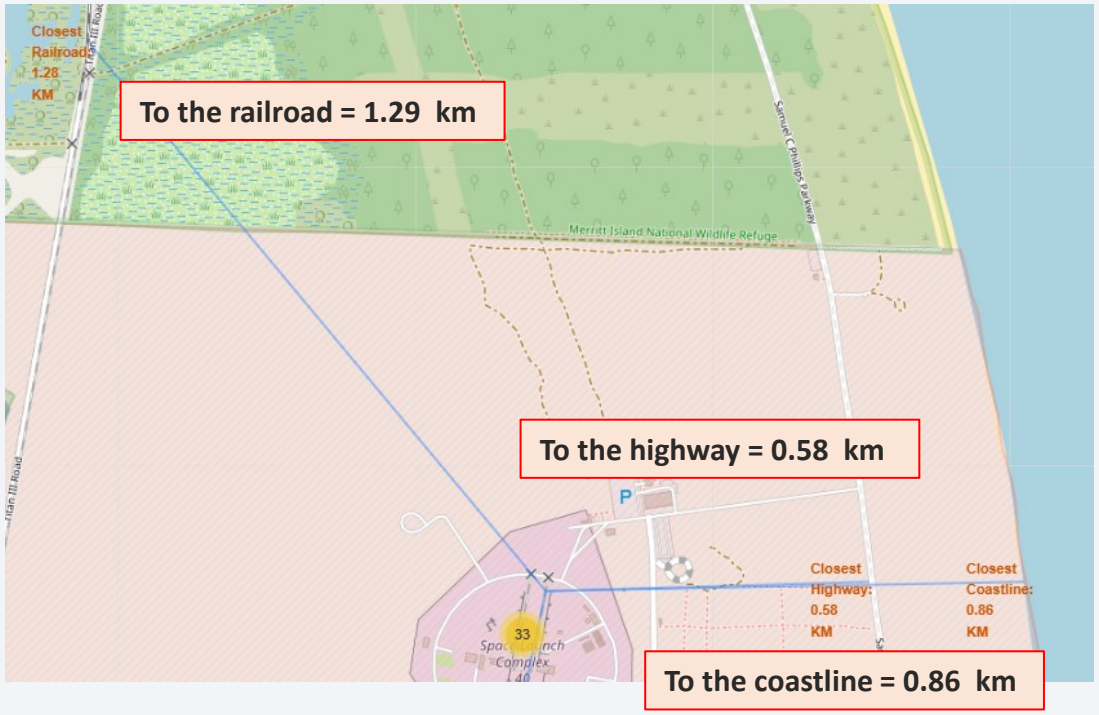
CCAFS LC-40 shows 7 successful landings and 19 failed landings.

Key Location Proximities for CCAFS SLC-40

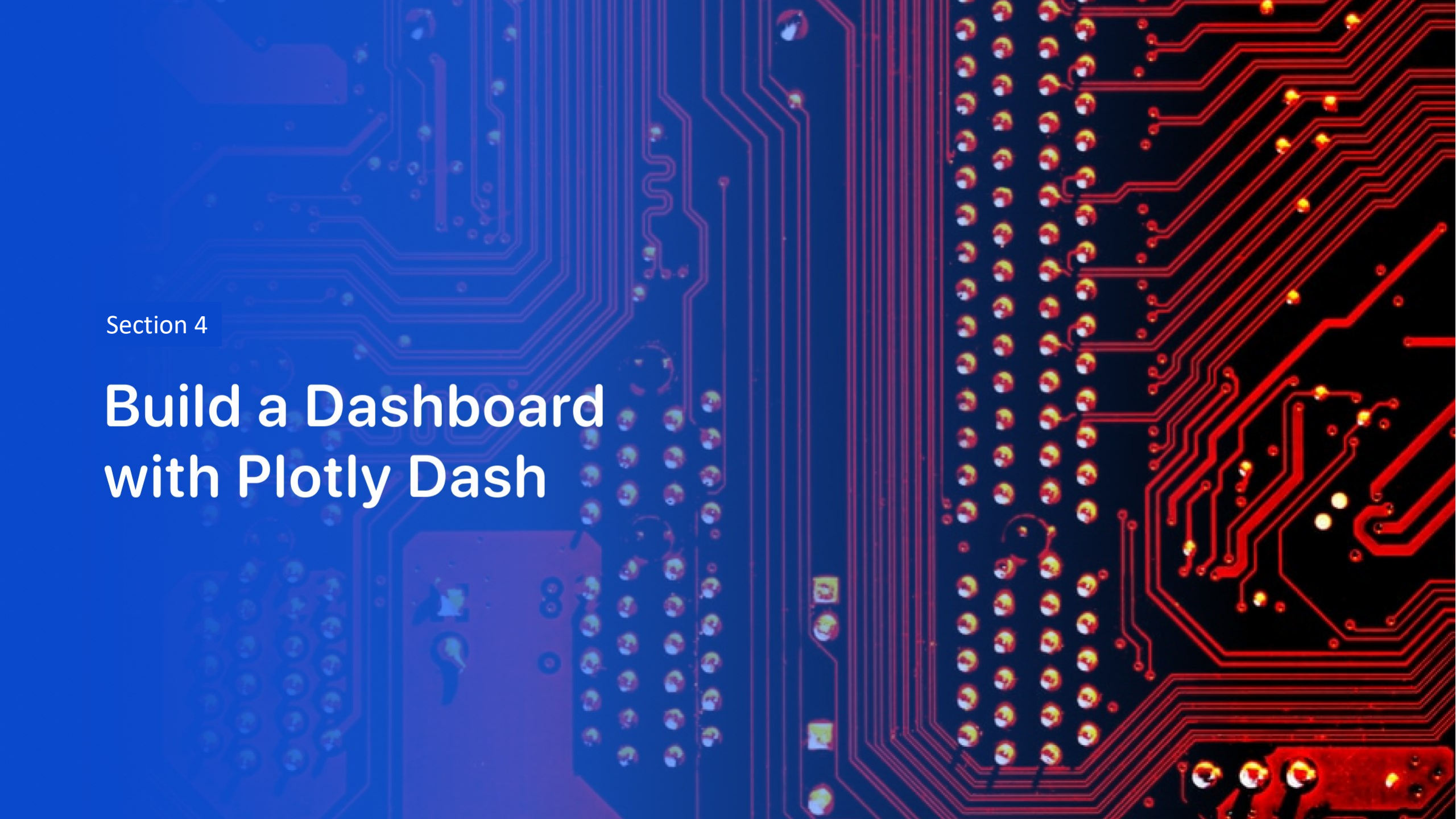
For the launch site CCAFS SLC-40 with coordinates Lat = 28.56 and Long =-80.57 we have these distances

To the coastline = 0.86 km; To the highway = 0.58 km; To the railroad = 1.29 km; To the city = 17.96 km

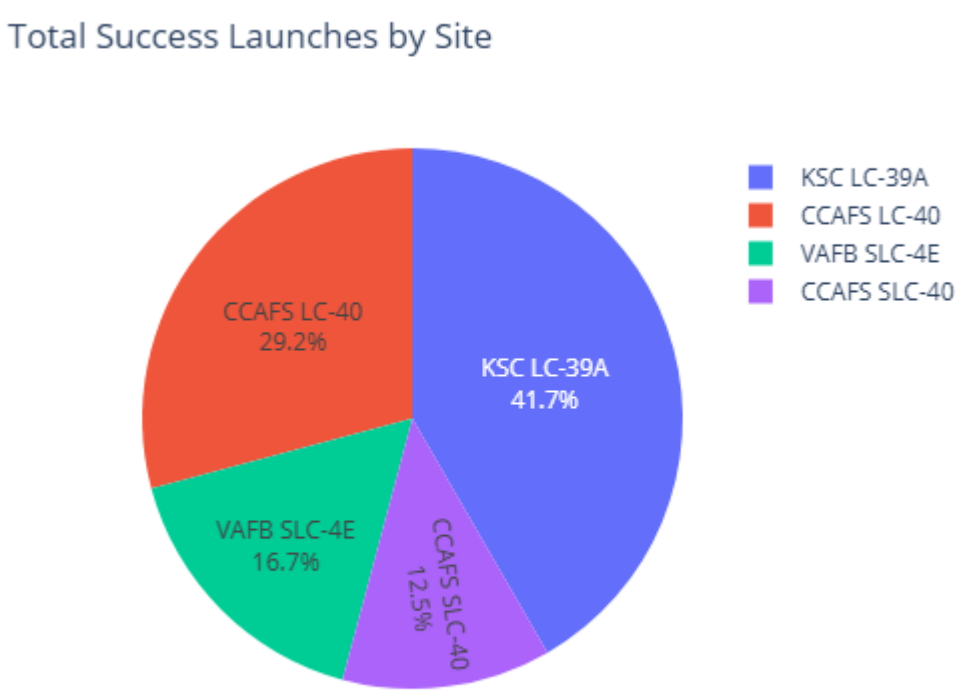
Launch sites are close to highways for human and supply transport.



Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.


Section 4
Build a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



This distribution reveals that launch sites CCAFS SLC-40, formerly known as CCAFS LC-40, and KSC share the highest number of successful landings. However, it's important to note that a significant portion of these successes occurred before the name change.

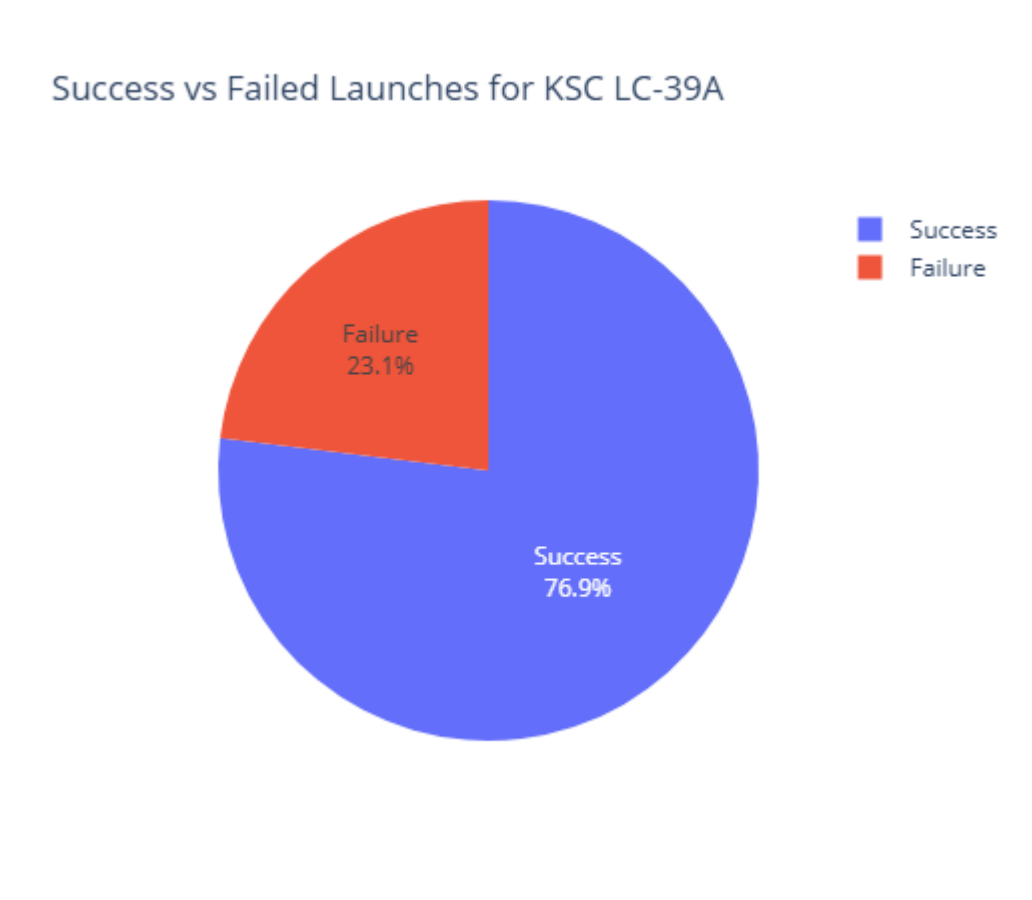
VAFB launch site exhibits the lowest number of successful landings, which may be attributed to a combination of factors:

- **Smaller Sample Size:** VAFB may have fewer total launches compared to the other sites.
- **Increased Launch Difficulty:** Launching from the West Coast can be more challenging due to factors like weather conditions and geographical constraints.

Further analysis is required to delve deeper into the specific reasons behind the lower success rate at VAFB.

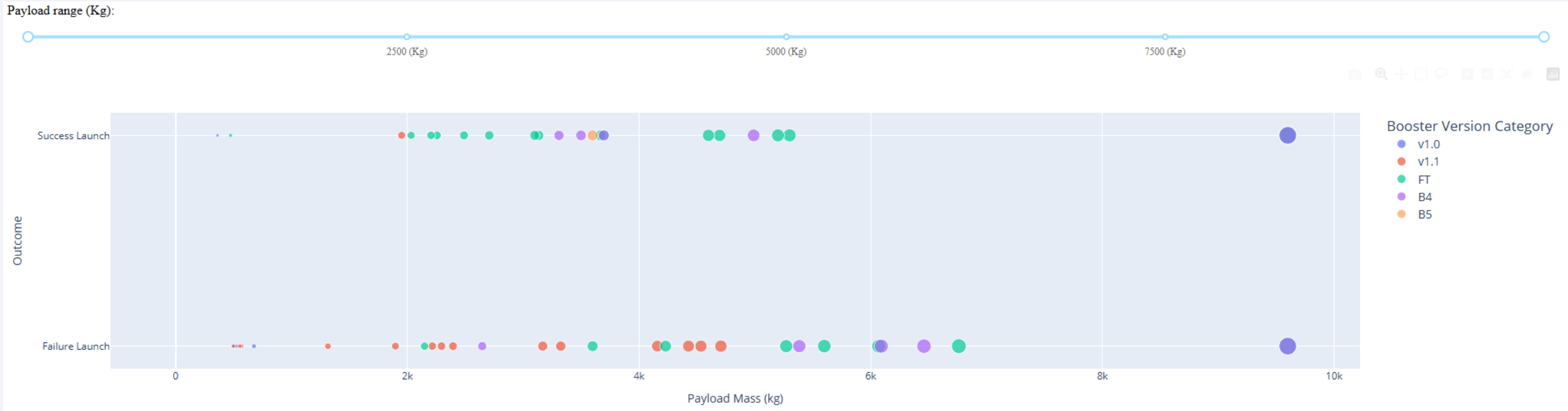
	Launch Site	Unsuccessful Landings	Successful Landings	Total Landings
0	CCAFS LC-40	19	7	26
1	CCAFS SLC-40	4	3	7
2	KSC LC-39A	3	10	13
3	VAFB SLC-4E	6	4	10

Highest Success Rate Launch Site



KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload Mass vs. Success vs. Booster Version Category



Interpreting the Payload Range Selector and Scatter Plot

The Plotly dashboard's Payload range selector is currently set between 0 and 10,000 kg, which doesn't fully capture the maximum payload capacity of 15,600 kg. The scatter plot visualizes the relationship between payload mass, landing success, booster version, and launch frequency:

- **Landing Success:** Successful and failed landings are indicated on y axis
- **Booster Version:** Different booster versions are represented by distinct colors.
- **Launch Frequency:** The size of each data point corresponds to the number of launches for that specific payload mass and booster version.

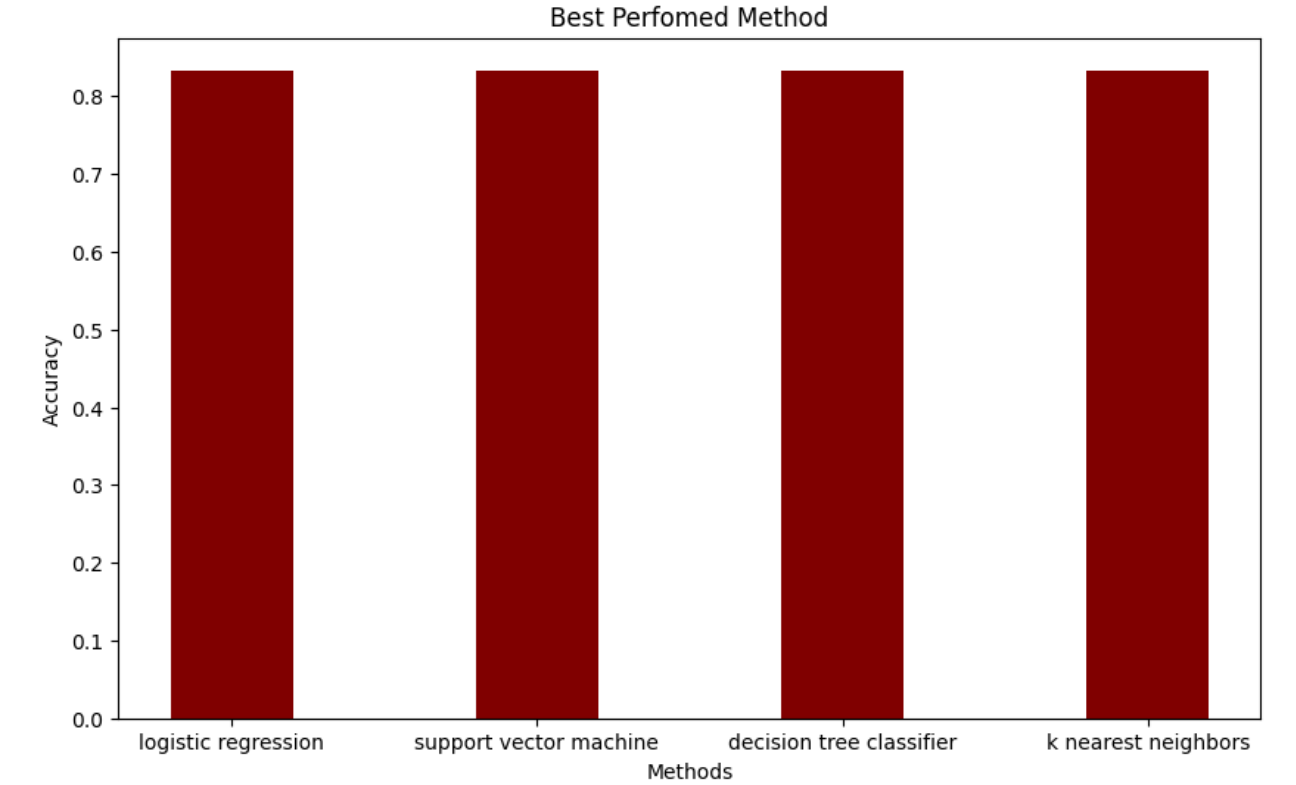
An interesting observation is the presence of two failed landings with zero payload mass within the 0-6000 kg range. This anomaly warrants further investigation to understand the underlying causes of these failures.

By exploring the interactive features of the dashboard, we can gain deeper insights into the factors influencing launch success and the performance of different booster versions across various payload ranges.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

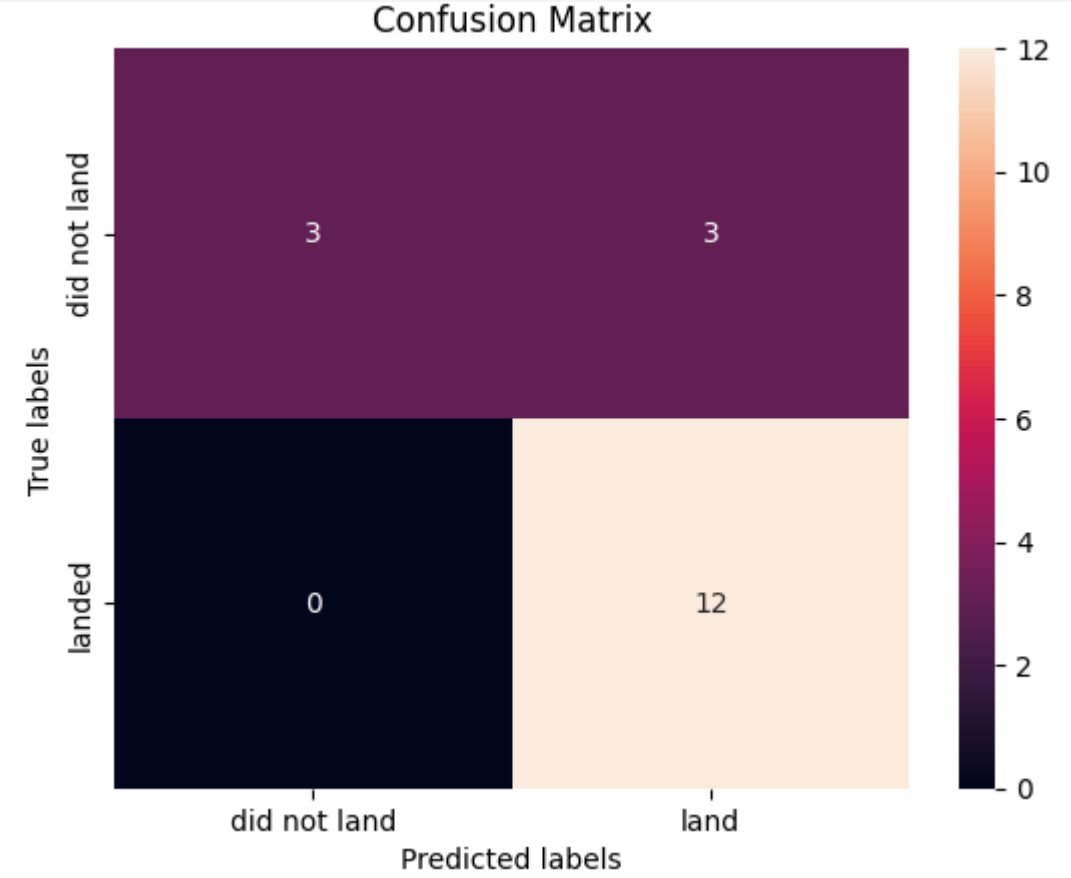


Methods	logistic regression	support vector machine	decision tree classifier	k nearest neighbors
Accuracy Scores	0.833333	0.833333	0.833333	0.833333

Model Performance and Data Limitations

- All models achieved an accuracy of 83.33% on the test set.
- The small sample size of 18 can lead to significant variance in model performance.
- A larger dataset is necessary to make a more definitive assessment of model performance.

Confusion Matrix



Model Performance Analysis: Confusion Matrix

- The confusion matrix provides valuable insights into the model's performance:
- **True Positives:** The models correctly predicted 12 successful landings.
- **True Negatives:** The models correctly predicted 3 unsuccessful landings.
- **False Positives:** The models incorrectly predicted 3 unsuccessful landings as successful.

A key observation is the tendency of the models to overpredict successful landings. This indicates a potential bias in the models, which may be due to imbalanced data or other factors.

To address this issue, we may need to explore techniques like class weighting or oversampling/undersampling to improve the model's ability to correctly classify unsuccessful landings.

Conclusions

Predicting SpaceX Stage 1 Landings: A Data-Driven Approach

- **Problem Statement:**

Our objective was to develop a machine learning model capable of accurately predicting the successful landing of SpaceX's Stage 1 boosters. This predictive capability could potentially save SpaceX significant costs, estimated at approximately $100 million USD per successful landing.

- **Data Acquisition and Preparation:**

1. **Data Sourcing:** We collected relevant data from a public SpaceX API and through web scraping of the SpaceX Wikipedia page.
2. **Data Labeling:** We carefully labeled the data to identify successful and unsuccessful landings.
3. **Data Storage:** The cleaned and labeled data was stored in an IBM DB2 SQL database for efficient querying and analysis.

- **Exploratory Data Analysis (EDA) and Visualization:**

To gain insights into the data, we created an interactive dashboard using Plotly Dash. This dashboard enabled us to visualize key trends and relationships between various factors, such as launch site, payload mass, and booster version.

- **Model Development and Evaluation:**

We developed a machine learning model that achieved an accuracy of 83% in predicting successful Stage 1 landings. This model can be a valuable tool for SpaceX to make informed decisions about launch attempts, potentially leading to significant cost savings.

- **Future Directions:**

To further enhance the model's accuracy and reliability, we recommend:

- ✓ **Collecting More Data:** A larger and more diverse dataset would allow for more robust model training and evaluation.
- ✓ **Exploring Advanced Techniques:** Implementing advanced machine learning techniques, such as deep learning or ensemble methods, could potentially improve performance.
- ✓ **Continuous Monitoring and Refinement:** Regularly monitoring the model's performance and retraining it with new data can help maintain its accuracy over time.

By leveraging data-driven insights and machine learning, we aim to contribute to SpaceX's mission of reducing launch costs and advancing space exploration.

Appendix

GitHub repository url:

https://github.com/marcthomas2710/Applied-Data-Science-Capstone_Marc_THOMAS

Instructors:

Instructors: Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

Special Thanks to All Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors

Thank you!