IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Name: Bhavya Mathur
Date: 21st May, 2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies:

1. Data Collection: Accessed SpaceX launch data via API and web scraped records from Wikipedia.

2. Data Cleaning & Preparation:

- Cleaned and formatted the data.
- Stored data in Db2 database and performed SQL queries.
- Conducted exploratory data analysis.

3. Feature Engineering: Created new features and standardized the data.

4. Interactive Visualizations:

- Mapped launch sites and success rates using Folium.
- Built an interactive dashboard with Plotly Dash.

5. **Model Building & Evaluation:**

- Implemented SVM, Decision Trees, and K-Nearest Neighbors.
- Tuned hyperparameters with GridSearchCV.
- Evaluated models using test data accuracy.

- Summary of all results:

1. Exploratory Data Analysis result
2. Interactive analytics in screenshots
3. Predictive Analytics result

Introduction

- Project background and context

In this capstone project, our goal is to predict the successful landing of the Falcon 9 first stage. SpaceX offers rocket launches at significantly lower prices than other providers, primarily because they can reuse the rocket's first stage. By accurately forecasting landing outcomes, we can better estimate launch costs and offer valuable insights for companies competing with SpaceX.

- Problems you want to find answers

➢ What are the key factors that impact the successful landing of the Falcon 9 first stage?
➢ How can machine learning models be used to reliably predict the landing outcome?
➢ Which machine learning algorithm provides the most accurate predictions for landing success?

Section 1

Methodology

Methodology

- Executive Summary: This project employs a comprehensive approach to predict the successful landing of the Falcon 9 first stage, incorporating data collection, processing, exploratory analysis, interactive visualizations, and predictive modeling.

Data Collection Methodology: Data was sourced from the SpaceX API, which provided detailed records of Falcon 9 launches, including launch dates, sites, payloads, and outcomes.

Perform Data Wrangling: Data cleaning involved handling missing values, standardizing formats, and ensuring consistency. Key features were extracted and new features engineered to enrich the dataset.

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

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/Bhavz9/IBM-Applied-Data-Science-Capstone-Project

Data Collection

- Step 1: SpaceX API Request

- Step 2: Web Scraping Wikipedia

- Step 3: Data Integration

Data Collection – SpaceX API

Step 1: Initiate API Request

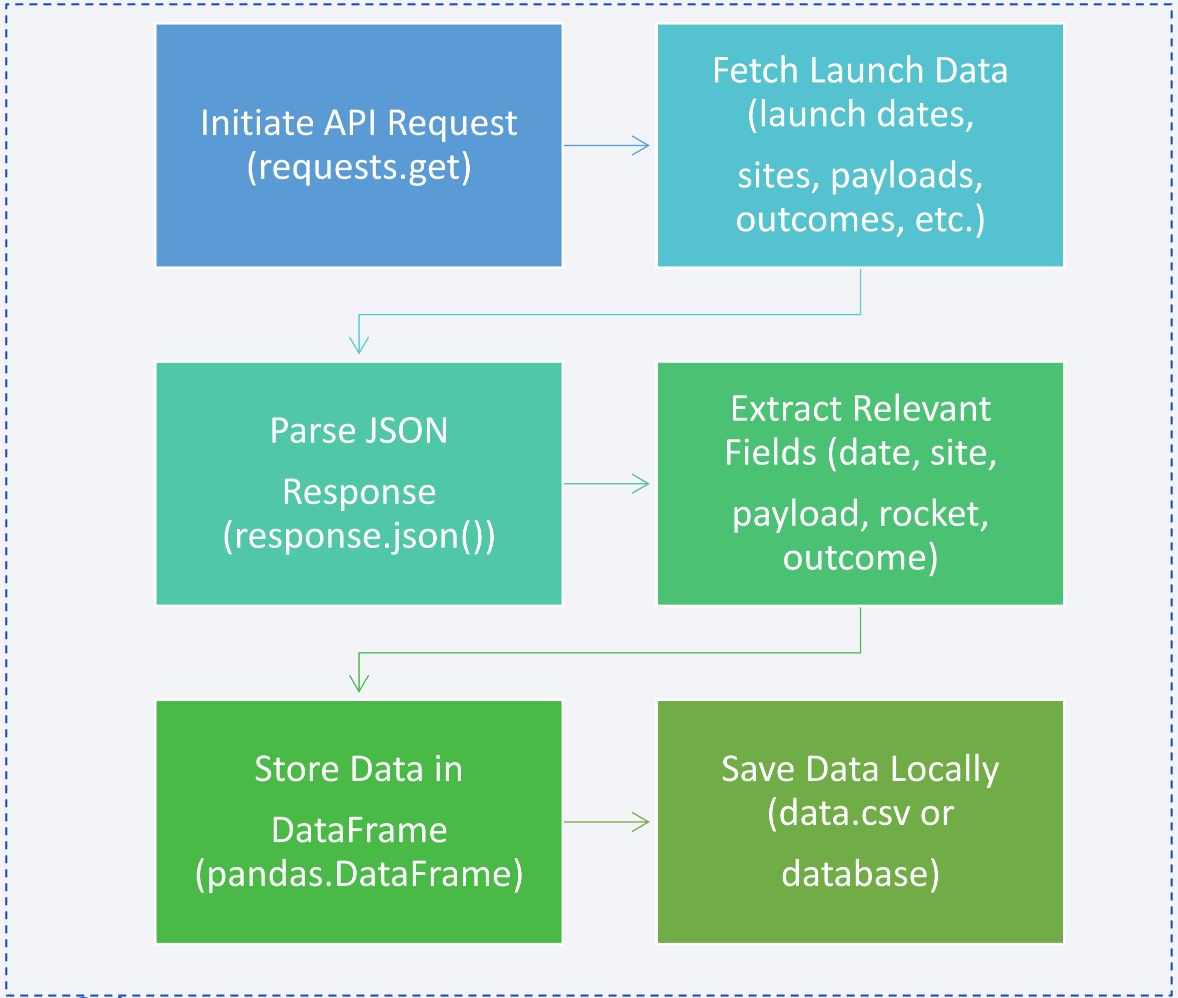
- Use Python's `requests` library to connect to the SpaceX API.
- Endpoint: `https://api.spacexdata.com/v4/launches`

Step 2: Parse API Response

- Convert API response from JSON to a Python dictionary.
- Extract relevant fields: launch date, launch site, payload mass, rocket type, outcome.

Step 3: Store Data Locally

- Save extracted data into a pandas DataFrame.
- Store the DataFrame locally for further processing.



GitHub URL: https://github.com/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api%20(1).ipynb

Data Collection - Scraping

Step 1: Initiate Web Scraping

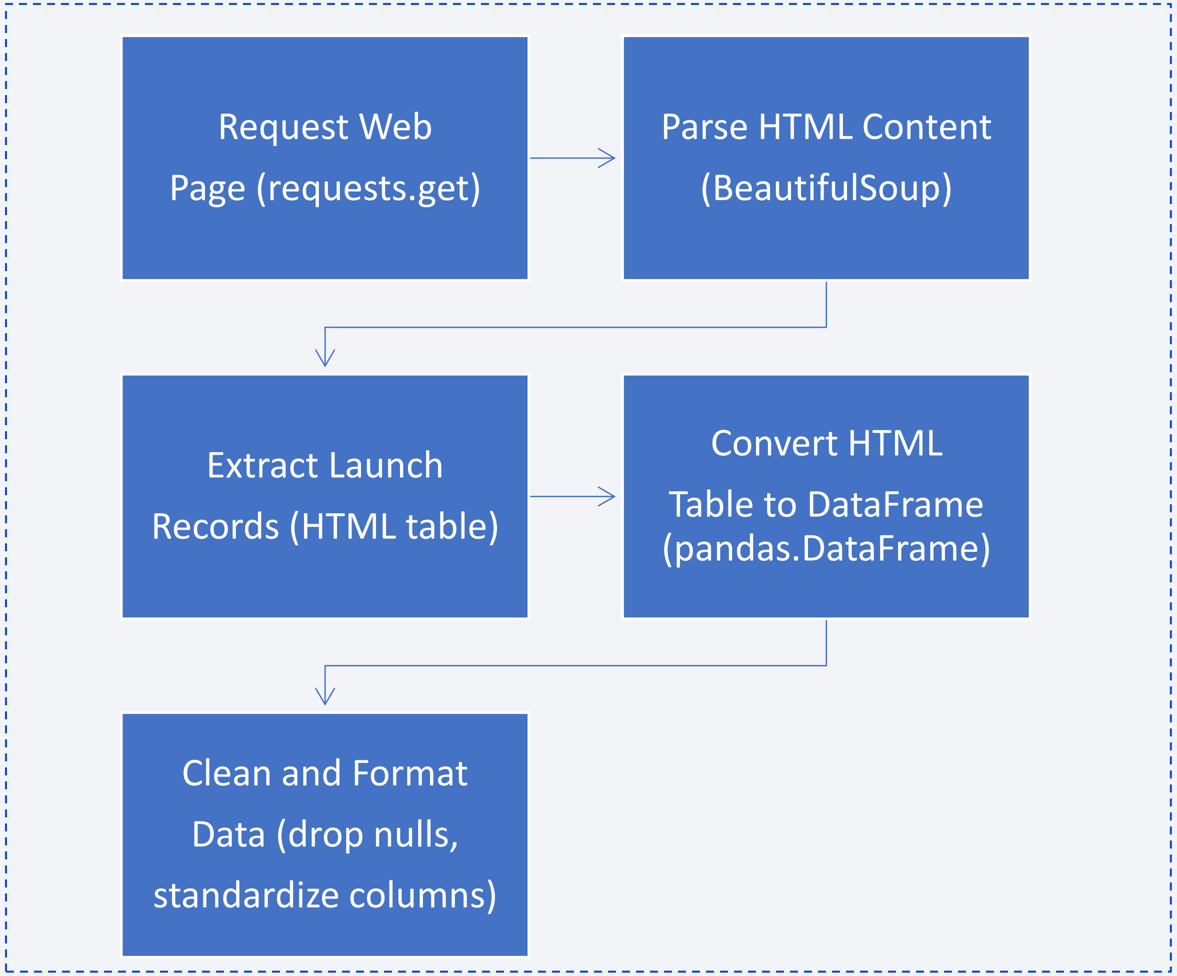
- Use Python's `requests` library to to fetch the HTML content of the Wikipedia page.
- Target Url: ``https://en.wikipedia.org/wiki/List_of_Falcon_9_ and_Falcon_Heavy_launches

Step 2: Parse HTML Content

- Use `BeautifulSoup` to parse the HTML content.
- Extract the HTML table containing Falcon 9 launch records.

Step 3: 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/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-webscraping%20(1).ipynb

Data Wrangling

Overview: Data wrangling involves cleaning, transforming, and organizing raw data into a structured format suitable for analysis.

- Step 1: Data Cleaning
  - ➢ Identify and fill or remove missing values in the dataset.
  - ➢ Use appropriate imputation techniques or drop rows/columns with excessive missing data.

- Step 2: Data Transformation
  - ➢ Convert data types to appropriate formats (e.g., date-time, numerical).
  - ➢ Standardize text (e.g., lowercase, remove whitespace).
  - ➢ Create new features from existing data (e.g., extract year from date).
  - ➢ Normalize/scale numerical features to ensure consistency.

Data Wrangling

- Step 3: 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.

- Step 4: Data Validation
  - ➢ Check for duplicate records and remove them.
  - ➢ Verify the accuracy and consistency of data entries.

Github URL: https://github.com/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/labs-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 Plotted:

1. **Histograms:**
    - **Purpose:** Used to visualize the distribution of numerical variables such as launch success rates, payload mass, and flight number.
    - **Why:** Helps in understanding the spread and central tendency of the data, identifying outliers, and assessing data skewness.
2. **Bar Charts:**
    - **Purpose**: Used to compare categorical variables such as launch outcomes (success/failure) across different categories like launch sites or rocket types.
    - Why: Provides a clear comparison of frequencies or proportions within categorical data, highlighting patterns or trends.
3. **Line Charts:**
    - **Purpose:** Used to track trends over time, such as the success rate of Falcon 9 launches across different years.
    - **Why**: Reveals temporal patterns and helps in understanding performance trends or changes over specific periods.

EDA with Data Visualization

4. **Scatter Plots:**

- **Purpose:** Used to explore relationships between two numerical variables, such as payload mass vs. launch success.
- Why: Identifies correlations or dependencies between variables, visualizing how one variable changes concerning another.

5. **Heatmaps:**

- Purpose: Used to visualize correlation matrices between multiple numerical variables.
- Why: Helps in identifying strong correlations (positive or negative) between variables, aiding feature selection or understanding multicollinearity.

6. **Box Plots:**

- Purpose: Used to display the distribution of numerical data through their quartiles.
- Why: Visualizes the spread and skewness of data, highlighting outliers and comparing distributions across different categories.

Github URL: https://github.com/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/edadataviz.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/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map Objects Created:

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.

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.

GitHub URL: https://github.com/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Plots/Graphs Added:

Success Pie Chart:

- Displays the distribution of successful and failed launches.
- Helps visualize the overall success rate and performance trends.

Success-Payload Scatter Plot:

- Shows the relationship between payload mass and launch success.
- Allows users to explore how payload mass influences mission outcomes.

Interactions Added:

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.

GitHub URL: https://github.com/srinibas-masanta/IBM-Applied-Data-Science-Capstone/blob/main/spacex_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 Tuning:

- 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:

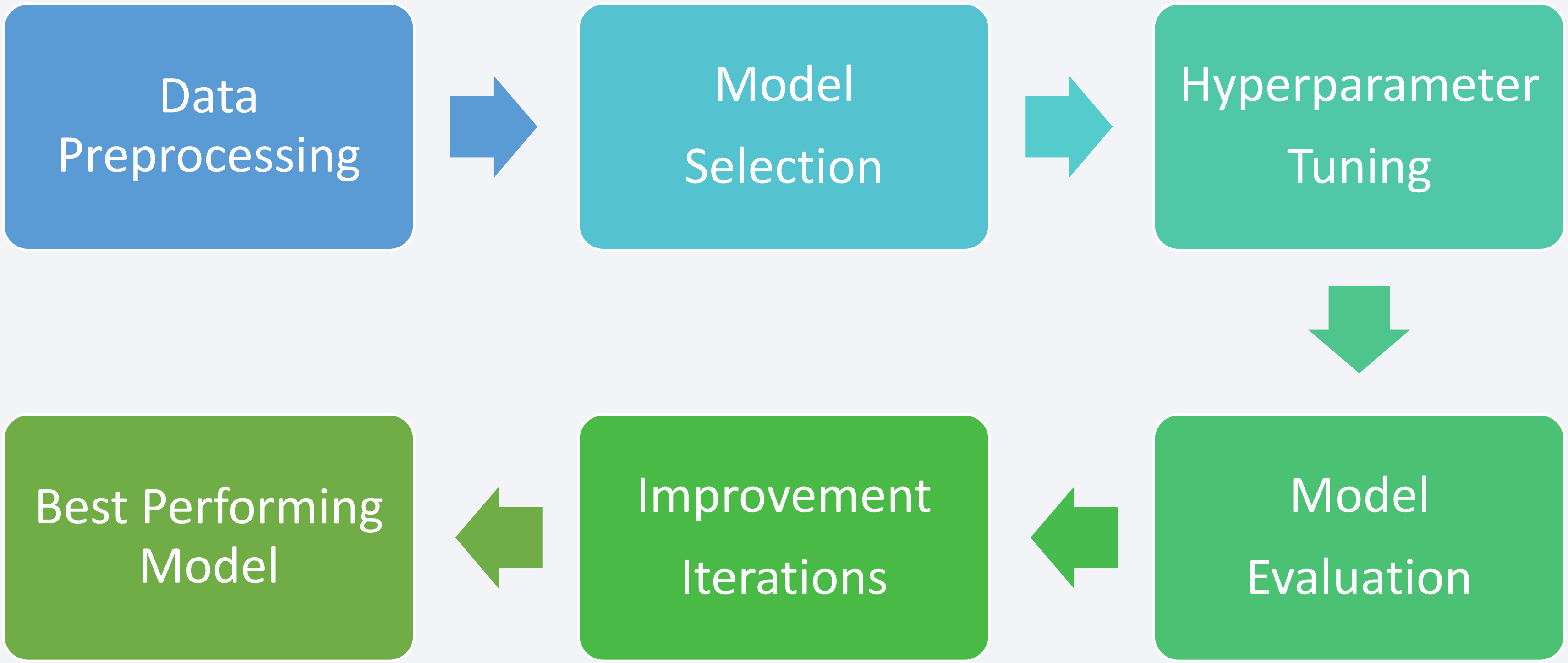
- Iteratively adjusted models based on insights from validation results.
- Fine-tuned hyperparameters to maximize predictive accuracy and reliability.

6. Selection of Best Performing Model:

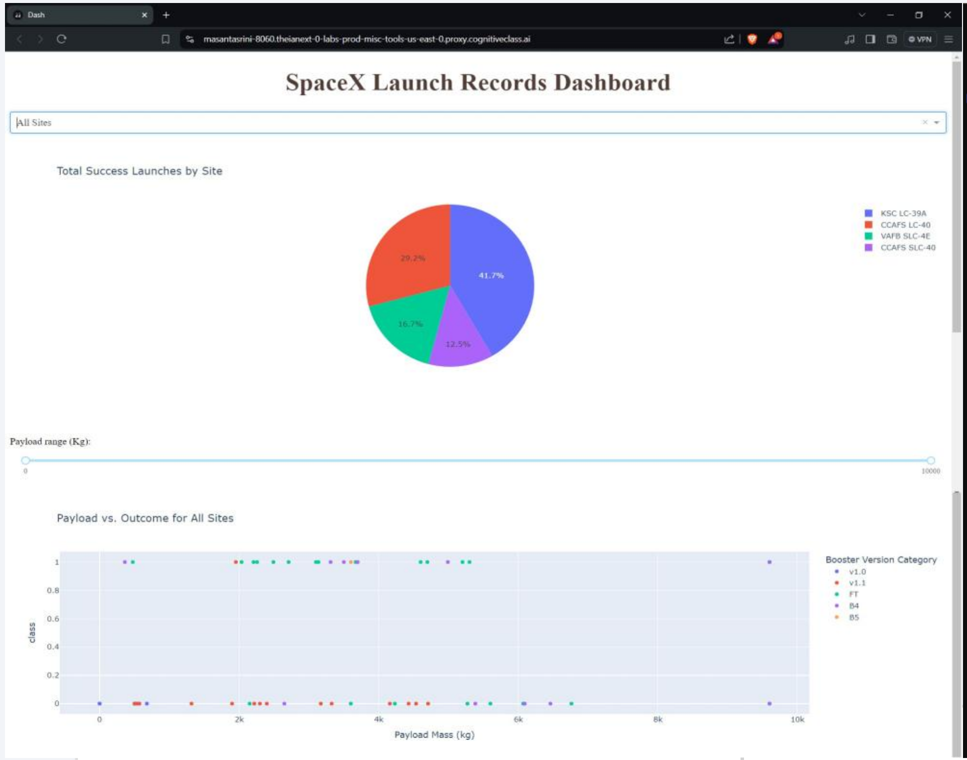
- Identified the model with the highest accuracy on the test set as the best performer.
- Considered both training and test set performance to avoid overfitting and ensure real-world applicability.

GitHub URL: https://github.com/Bhavz9/IBM-Applied-Data-Science-Capstone-Project/blob/main/7.%20SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Predictive Analysis (Classification) Flow Chart



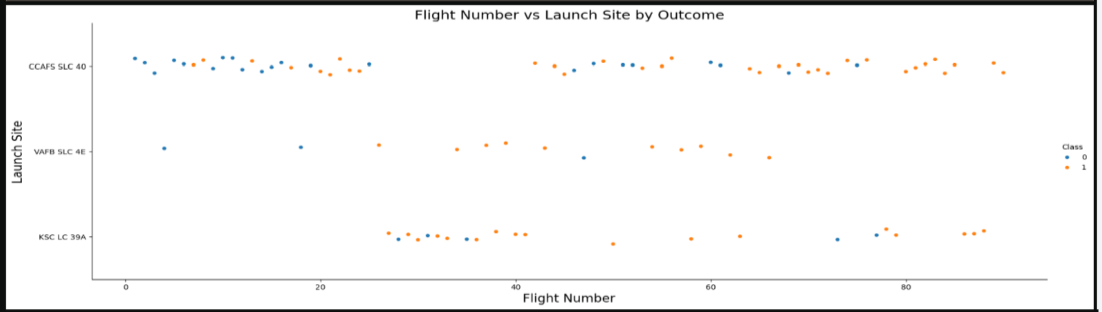
Results



TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

```
[10]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect=3, height=6, kind="strip")
plt.xlabel("Flight Number", fontsize=16)
plt.ylabel("Launch Site", fontsize=16)
plt.title("Flight Number vs Launch Site by Outcome", fontsize=18)
plt.show()
```



TASK 12

Find the method performs best:

Based on the results I got:

1. **Logistic Regression**: Accuracy on test data: 0.8333
2. **Support Vector Machine (SVM)**: Accuracy on test data: 0.8333
3. **Decision Tree**: Accuracy on test data: 0.9444
4. **K Nearest Neighbors (KNN)**: Accuracy on test data: 0.8333

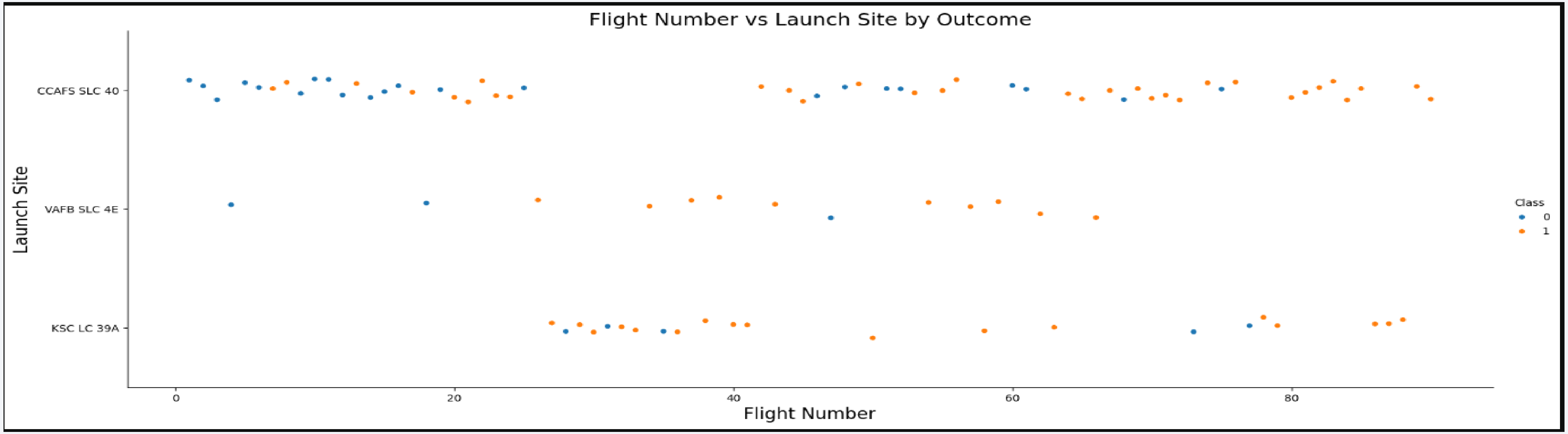
It seems there was a significant increase in accuracy for the Decision Tree model, which shows the highest accuracy of 0.9444 on the test set. Therefore, in this comparison, the Decision Tree model performs the best among the classifiers you've trained.

Section 2

Insights drawn from EDA

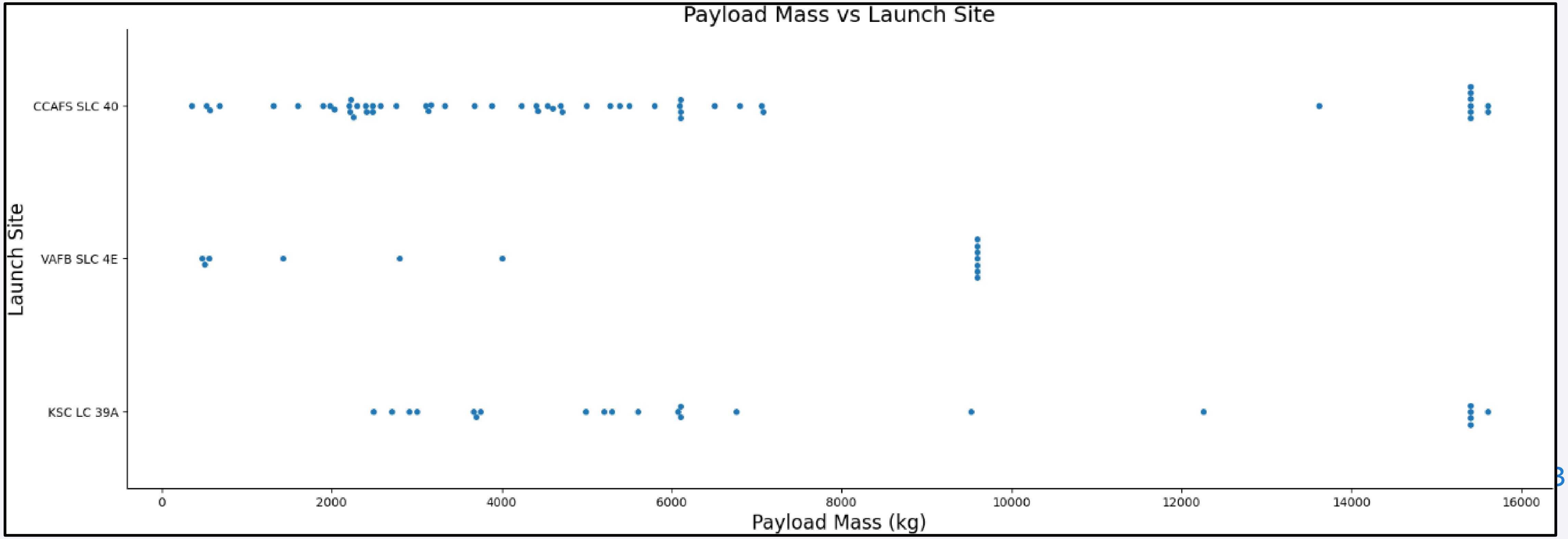
Flight Number vs. Launch Site

- **Mixed Outcomes at Major Launch Sites**: Both CCAFS SLC 40 and KSC LC 39A have a mix of successful (orange) and unsuccessful (blue) landings, indicating that factors other than the launch site itself may influence the landing success.
- **Consistent Activity Across Flight Numbers: Flight numbers are widely distributed**, showing consistent launch activity with no clear trend in landing success over time.



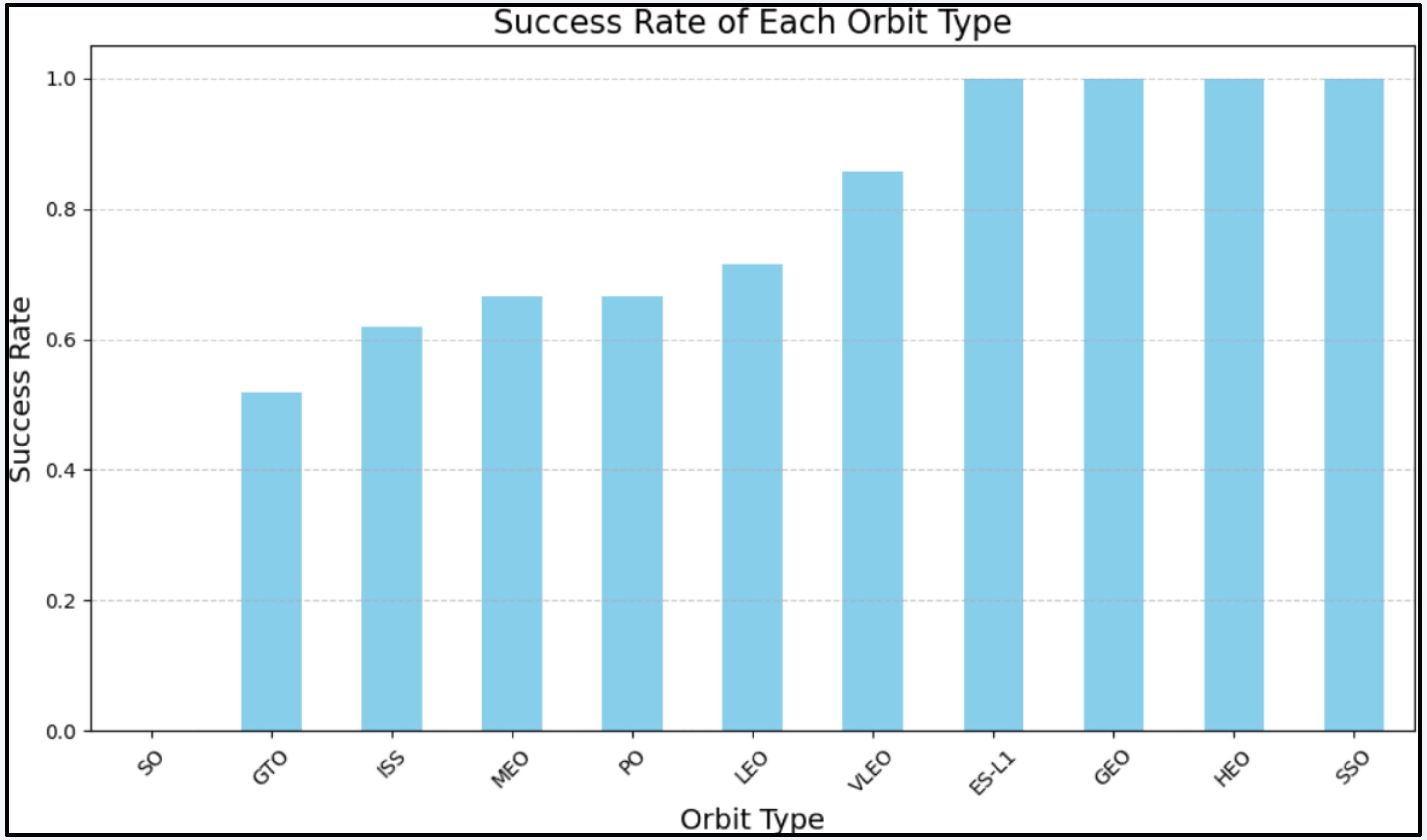
Payload vs. Launch Site

- **Payload Range by Site:** CCAFS SLC 40 mostly handles payloads under 10,000 kg, while VAFB SLC 4E and KSC LC 39A support a broader range, reflecting diverse missions.
- **Heavy Payload Launches:** KSC LC 39A often launches payloads over 15,000 kg, highlighting its role in high-capacity missions.

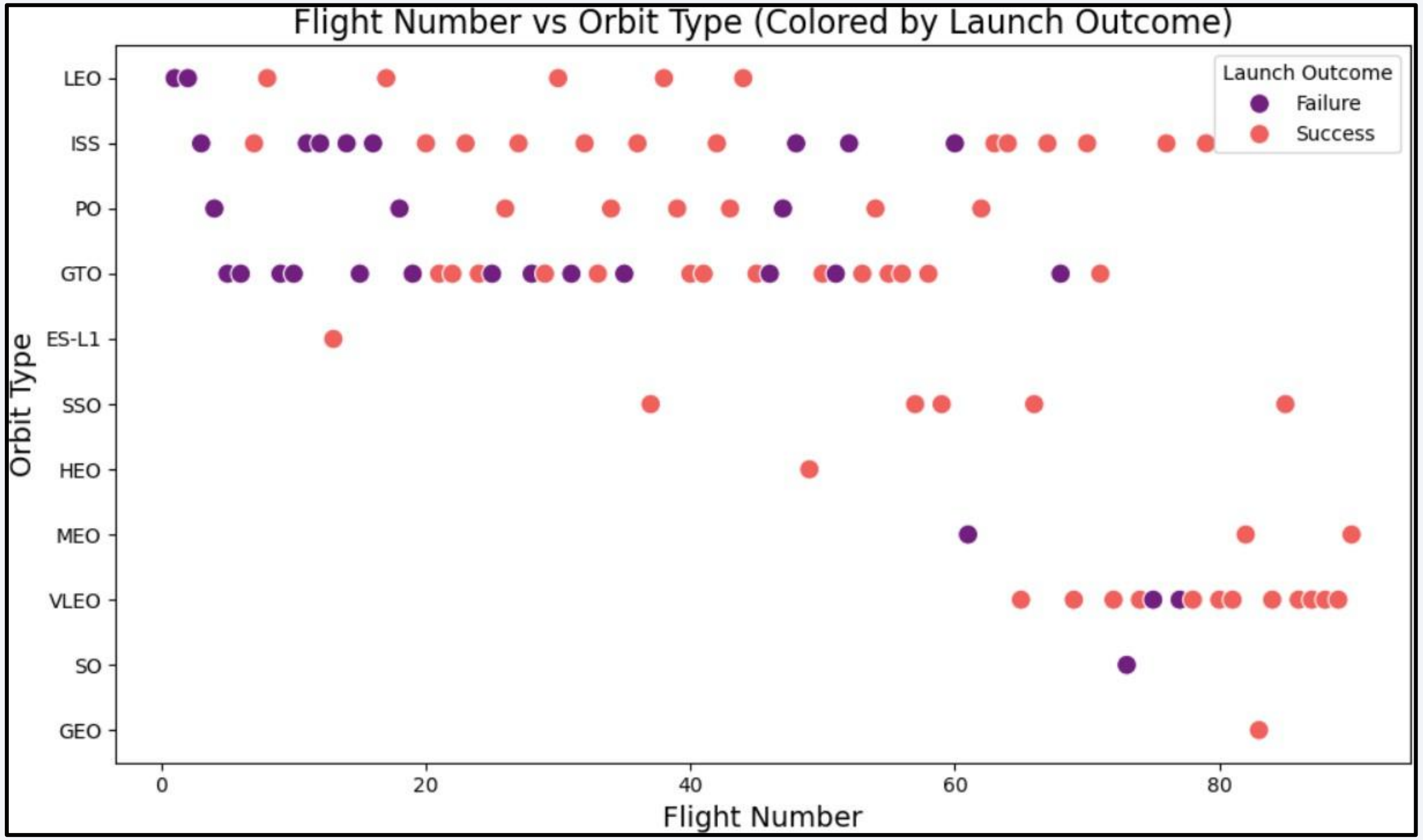


Success Rate vs. Orbit Type

- **High Success Rates:** Missions to VLEO, ES-L1, GEO, HEO, and SSO orbits have achieved a perfect success rate, indicating these orbits are highly reliable for successful first stage landings.
- **GTO Challenges:** GTO missions have lower success rates, indicating higher complexity.

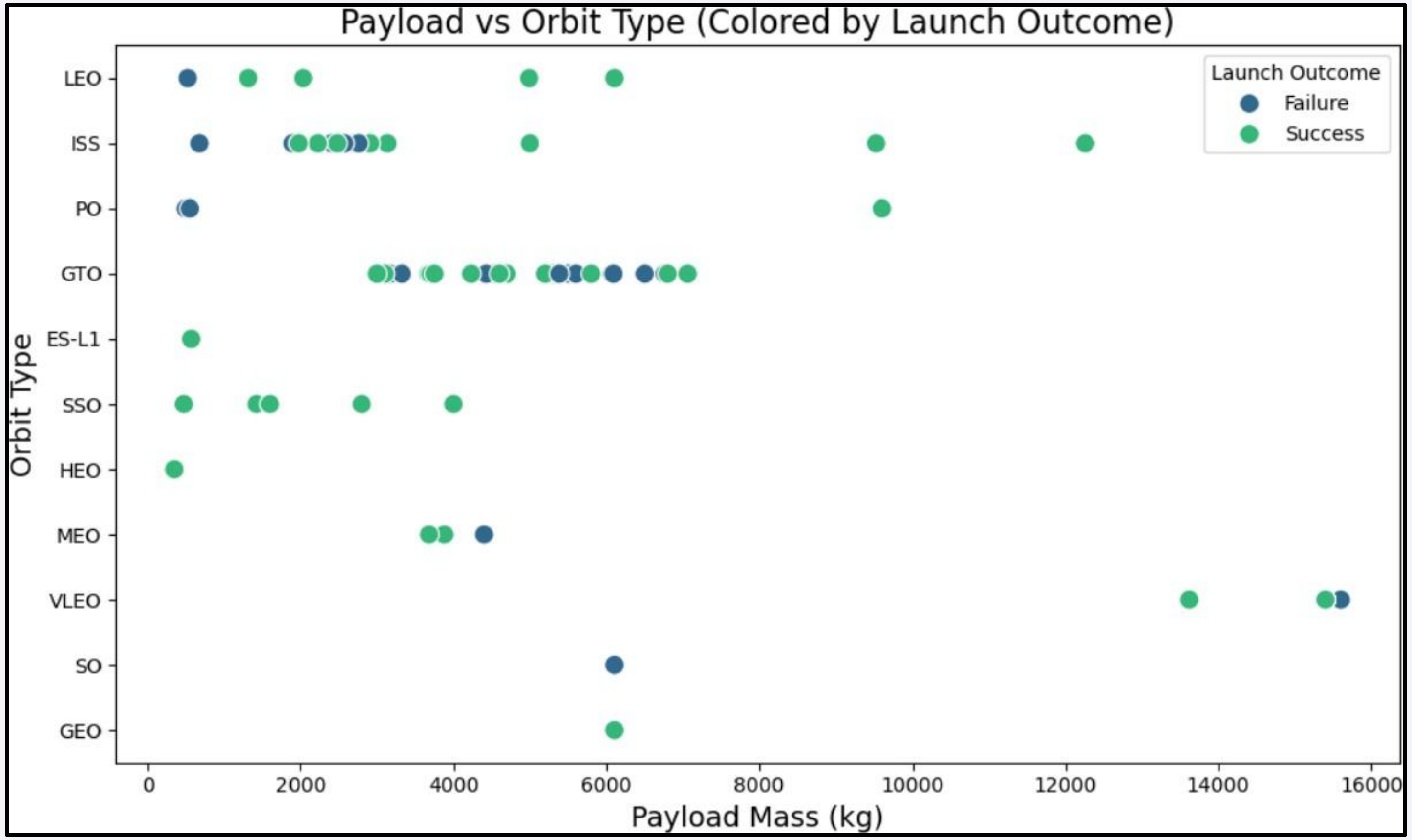


Flight Number vs. Orbit Type



- **Increased Success Over Time:** The success rate of Falcon 9 launches improves significantly with higher flight numbers, indicating that experience and iterative improvements contribute to better outcomes.
- **Better Recent Performance:** GTO and ISS missions have become more successful over time, showing progress in planning and execution.

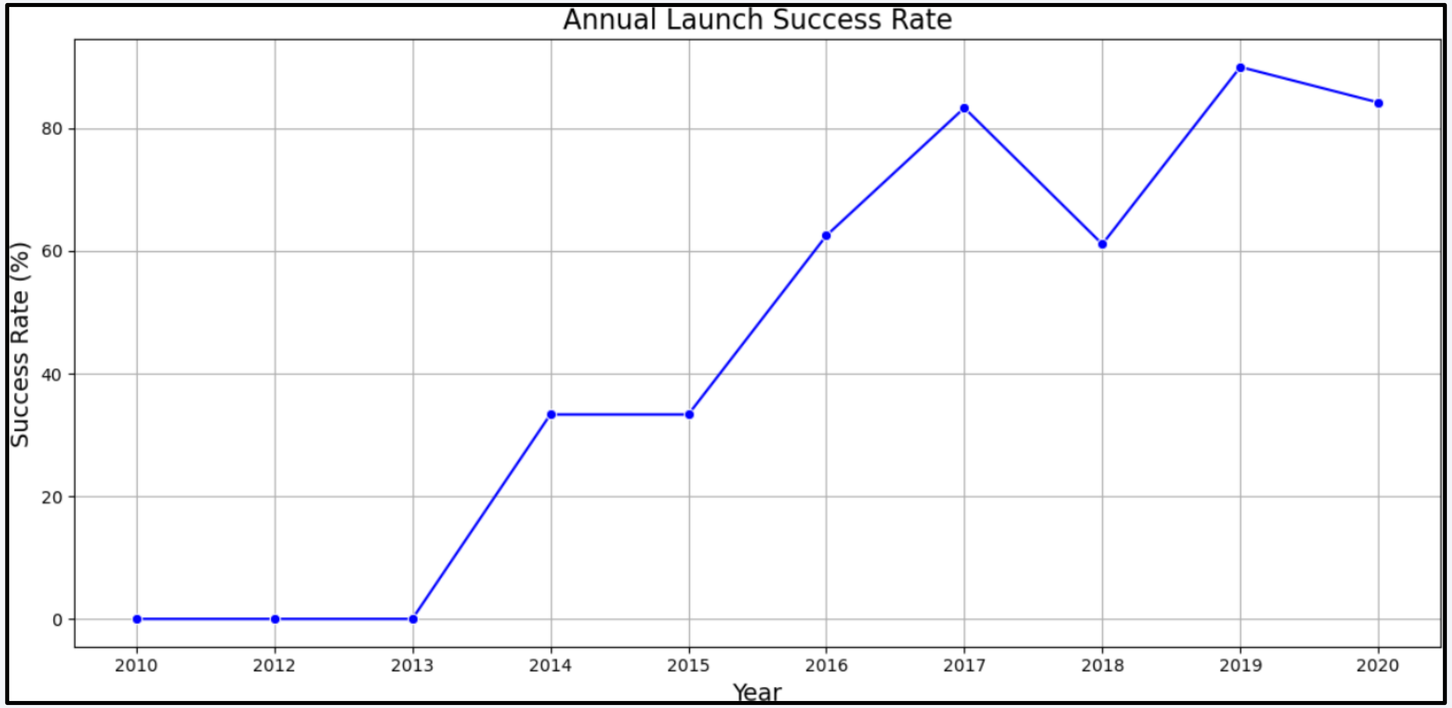
Payload vs. Orbit Type



- Successful landings are more frequent across all orbit types, especially for payloads less than 6000 kg.
- Higher payload masses (above 10,000 kg) show a mix of successes and failures, indicating increased difficulty with heavier payloads.

Launch Success Yearly Trend

- The annual launch success rate has shown a significant improvement from 2013 onwards, reaching over 80% by 2020.
- Despite a dip in 2018, the overall trend indicates increasing reliability and success in Falcon 9 launches over the years.



All Launch Site Names

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

Task 1

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

[21]:
```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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

[21]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

[26]:
```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

[26]:

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

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

Total Payload Mass

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

Task 3

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

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

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

[30]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

Task 4

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

```
[34]: %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';

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

[34]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

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

Task 5

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

Hint:Use min function

```
[36]: %sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

[36]:

MIN(Date)
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.

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

```
[38]: %sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS
```

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

[38]:

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

Total Number of Successful and Failure Mission Outcomes

- This SQL query counts the total number of successful and failed missions from the SPACEXTABLE, grouping the results by mission outcome.

Task 7

List the total number of successful and failure mission outcomes

```
[42]: %sql SELECT mission_outcome, count(*) as Count FROM SPACEXTABLE GROUP by Mission_Outcome     ORDER BY Mission_Outcome
```

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

[42]:

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

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function

Task 8

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

```
[42]: %sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);

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

[42]:

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

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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.

•[52]:
```
%sql SELECT substr(Date, 6, 2) AS Month, landing_outcome, booster_version, Launch_Site FROM SPACEXTABLE WHERE substr(Date, 1, 4) = '2015'
AND landing_outcome = 'Failure (drone ship)';
```

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

[52]:

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

- 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.

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.

```
•[57]: %sql select landing_outcome, count(*) as count from SPACEXTABLE where Date >= '2010-06-04' AND Date <= '2017-03-20'
         GROUP by landing_outcome ORDER BY count Desc
```

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

[57]:

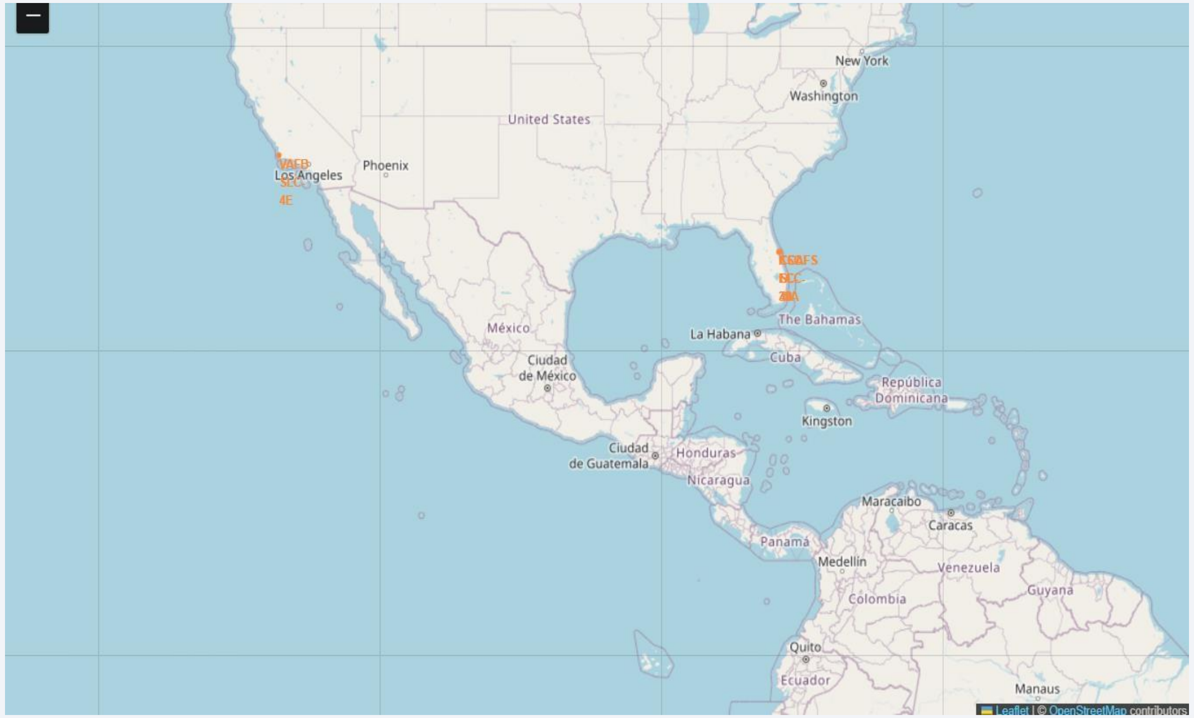
Landing_Outcome	count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Task 1: Mark all launch sites on a map

1. **Are all launch sites in proximity to the Equator line?**
   - No, not all launch sites are in close proximity 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.



2. **Are all launch sites in very close proximity to the coast?**
   - Yes, all launch sites are in close proximity to the coast.
   - The 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.

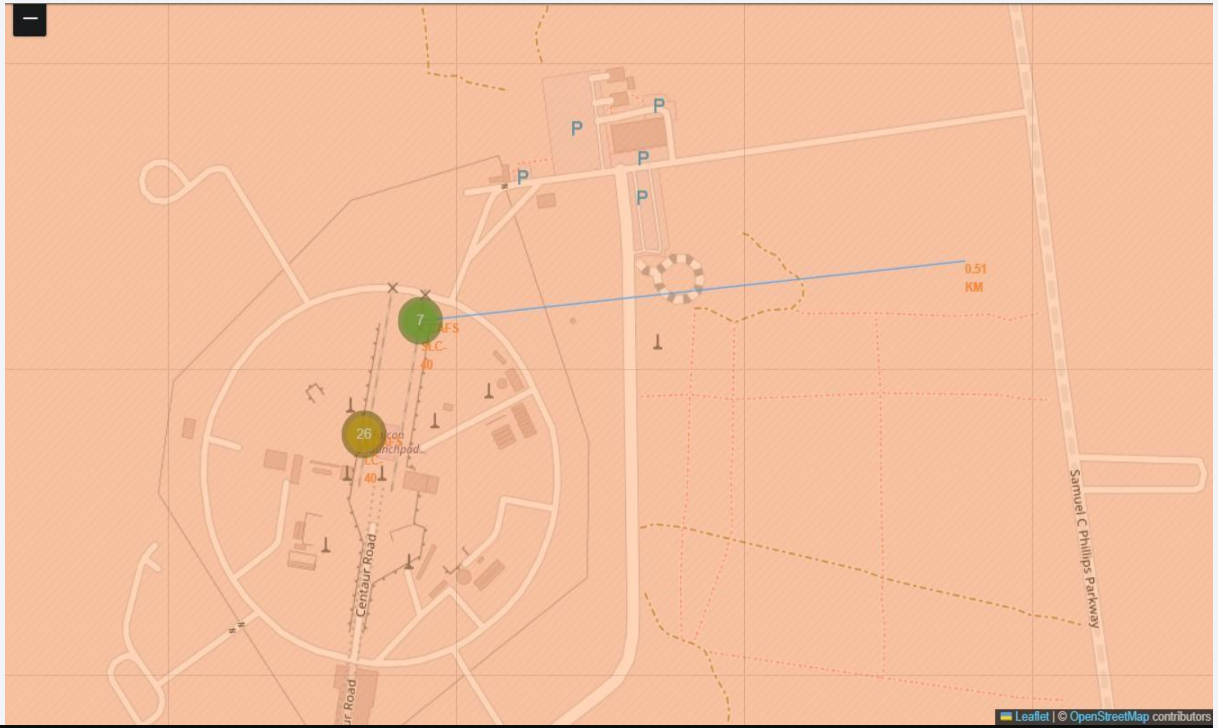
Task 2: Mark the success/failed launches for each site on the map

- The use of clustered markers in the visualization improves the analysis of SpaceX launch data by organizing numerous data points, making patterns more visible and easier to interpret. Marker colors and popups offer additional insight into launch details and outcomes.

- For instance, in the screenshot, CCAFS LC-40 has 26 launches—19 red markers and 7 green—clearly showing success and failure rates. Red likely represents failed launches, and green indicates successes, offering instant visual feedback on site performance.

Task 3: Calculate the distances between a launch site to its proximities

This plot shows the CCAFS SLC-40 launch site is about 0.51 km from the coast. The PolyLine highlights this direct distance, reflecting the common practice of placing launch sites near coastlines for safe over-water flight paths and recovery.

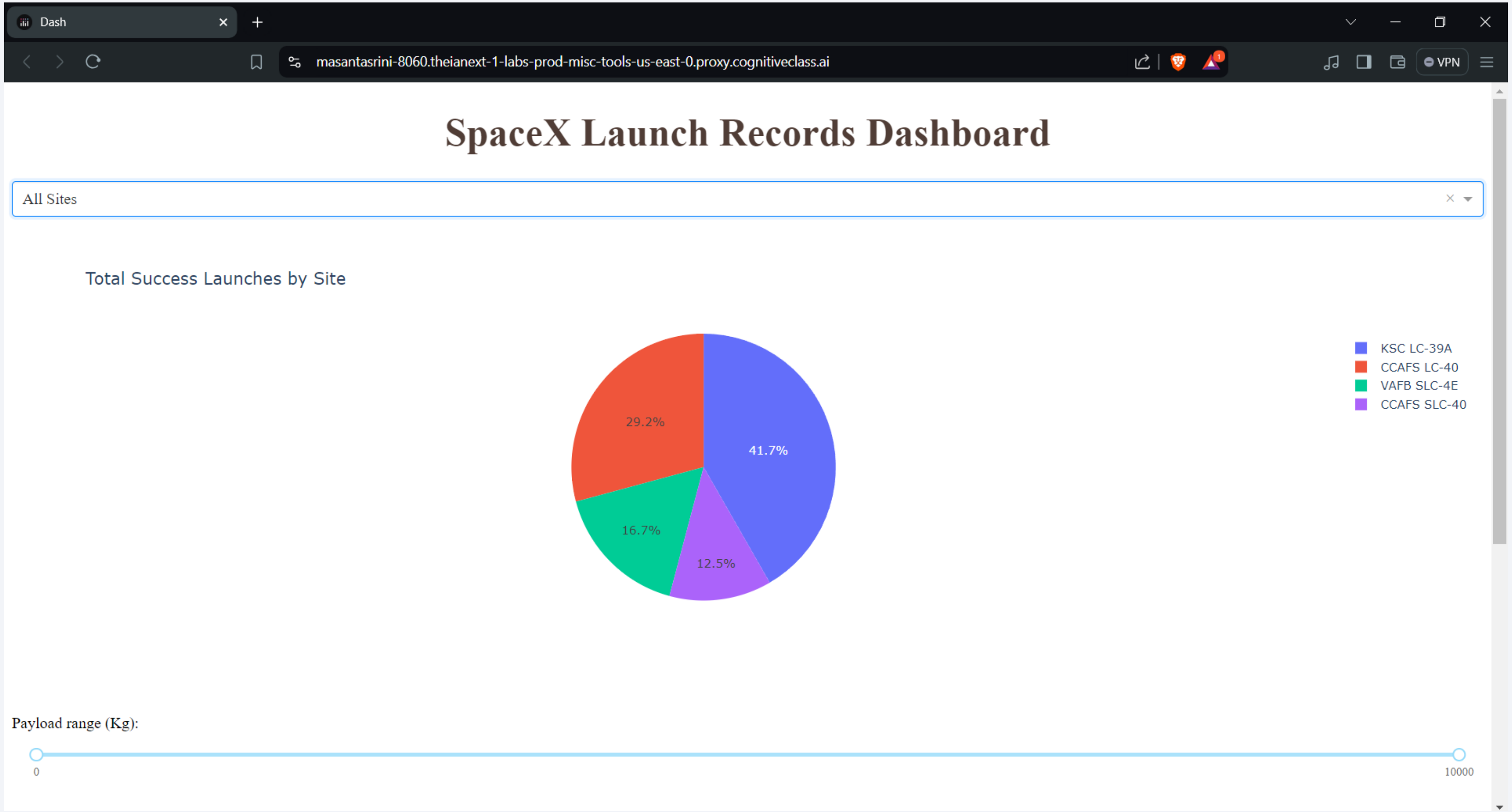



```
[43]: coastline_lat = 28.56367
      coastline_lon = -80.57163

      # Example launch site coordinates (replace with actual launch site coordinates)
      launch_site_lat = launch_sites_df.loc[launch_sites_df['Launch Site'] == 'CCAFS SLC-40', 'Lat'].values[0]
      launch_site_lon = launch_sites_df.loc[launch_sites_df['Launch Site'] == 'CCAFS SLC-40', 'Long'].values[0]

      # Calculate distance using the calculate_distance function
      distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)

      print(f"Distance from launch site to closest coastline: {distance_coastline} km")

      Distance from launch site to closest coastline: 0.5097439631188213 km
```

Section 4

Build a Dashboard with Plotly Dash

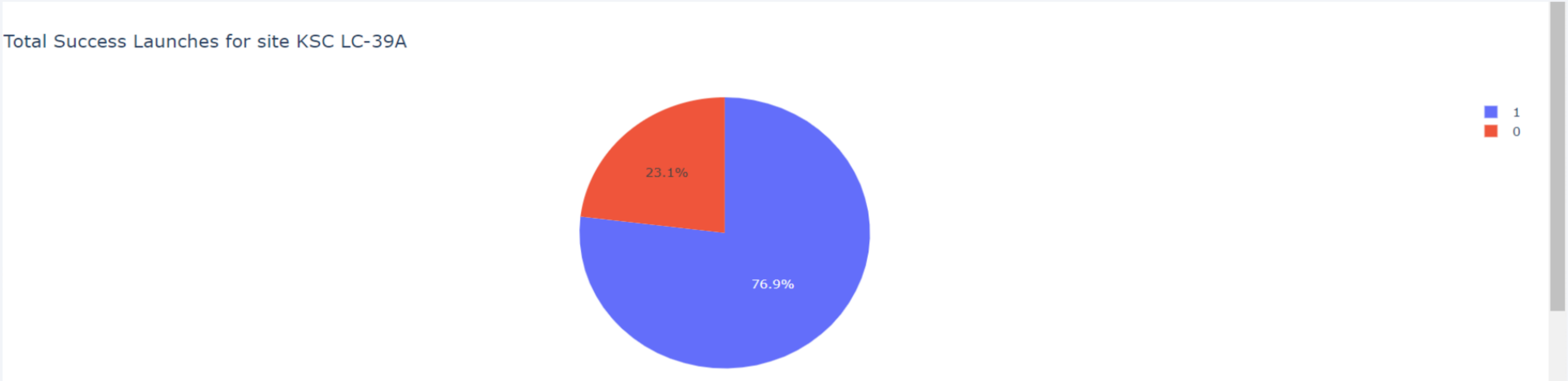
Launch Success Count for all sites (in a pie chart)



Key Findings:

- **CCAFS LC-40**: 29.2%
- **CCAFS SLC-40**: 12.5%
- **VAFB SLC-4E**: 16.7%
- **KSC LC-39A**: 41.7%
- The **KSC LC-39A** launch site has the highest number of successful launches, making up 41.7% of the total
- successes. This indicates that KSC LC-39A is a highly reliable site for SpaceX launches.

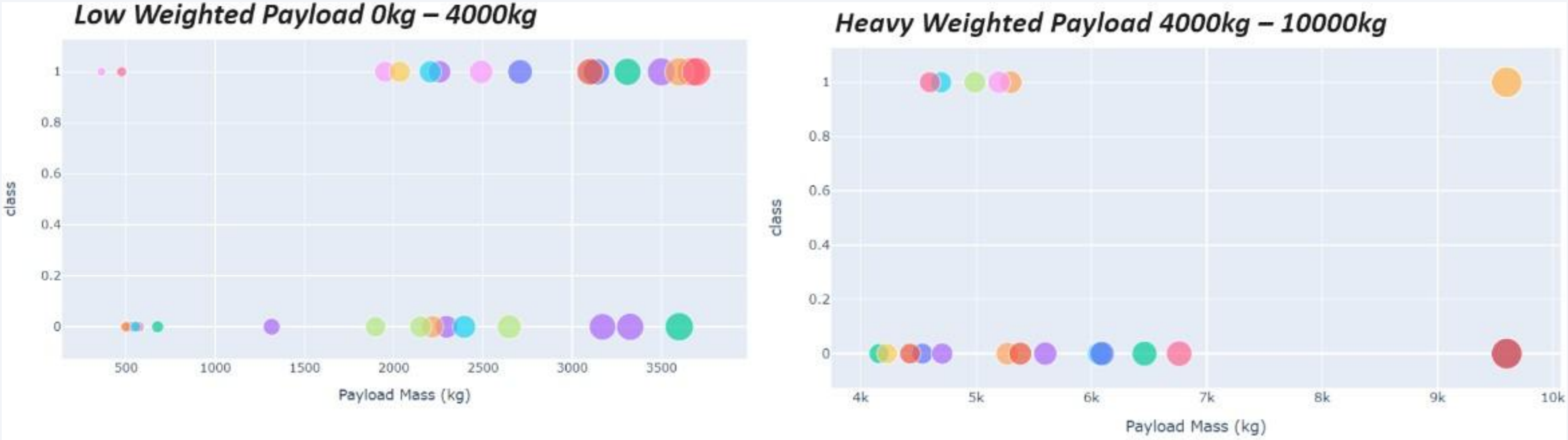
Pie chart for the launch site with highest launch success ratio



Key Findings:

- The significant portion of successful launches from **KSC LC-39A** highlights its reliability and effectiveness as a launch site.
- For **KSC LC-39A**:
  - **Class 1** (Successful Launches): 76.9%
  - **Class 0** (Unsuccessful Launches): 23.1%
- The high success rate (76.9%) for **Class 1** launches underscores the effectiveness and reliability of the KSC LC-39A site.

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



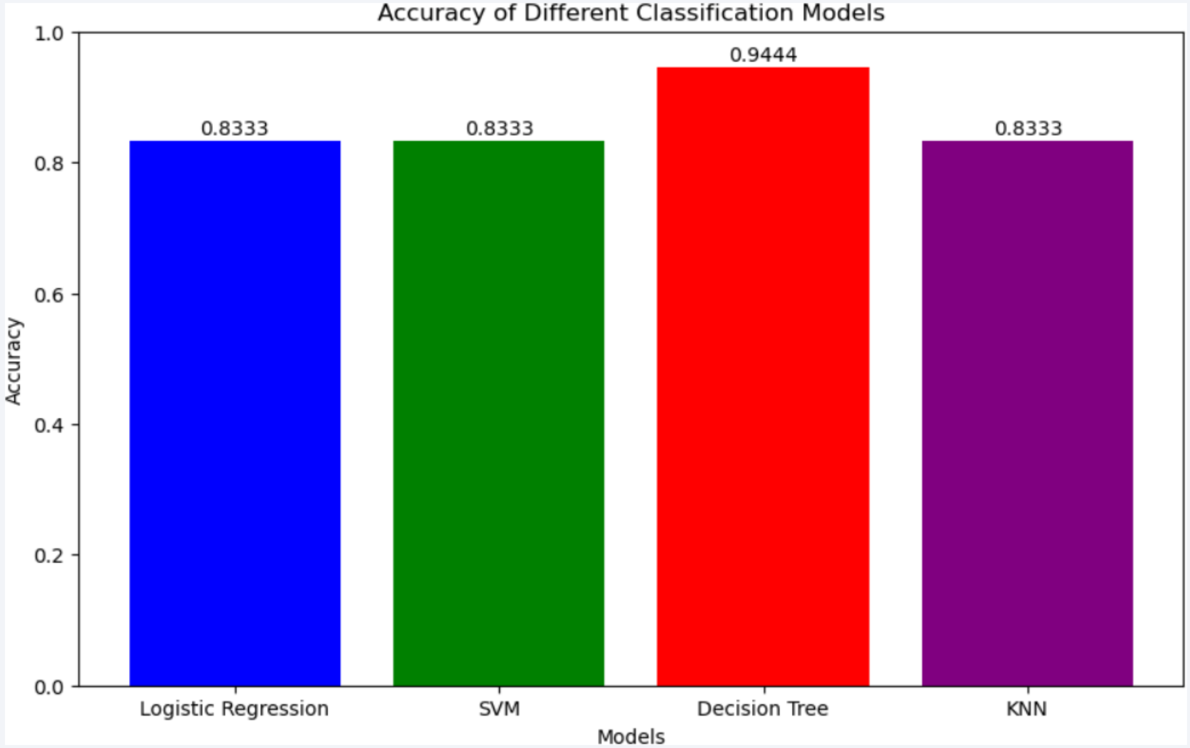
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

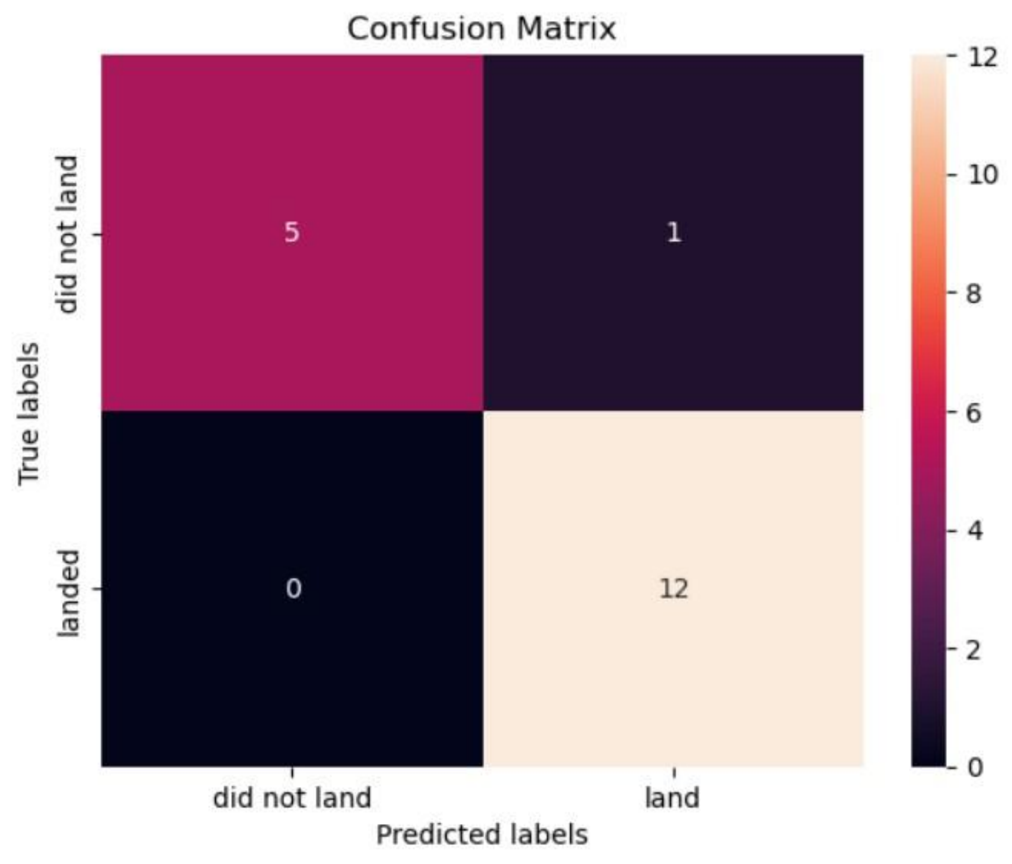
Predictive Analysis (Classification)

Classification Accuracy

- Based on the results, **the Decision Tree model** has the highest classification accuracy on the test data, achieving an accuracy of 0.9444. This suggests that the Decision Tree model is better suited for this dataset compared to Logistic Regression, Support Vector Machine, and K Nearest Neighbors, all of which achieved an accuracy of 0.8333.



Confusion Matrix



Explanation and Insights:

•**High Accuracy**: The model achieved 94.44% accuracy with strong true positive and true negative rates.
•**No False Negatives**: All actual landings were correctly predicted, ensuring operational safety.
•**Low False Positives**: Only one false positive, which is manageable and less critical than false negatives.
•**Balanced Performance**: Slightly biased toward predicting success—suitable for cost estimation and planning in aerospace.

Conclusions

- **Point 1**: Analysis shows that "CCAFS LC-40" has the highest success rate, contributing to 43.7% of all successful launches, suggesting favorable conditions or processes at this site.
- **Point 2**: The "FT" booster version demonstrated strong reliability across different payloads, making it a preferable choice for enhancing mission success.
- **Point 3**: There’s no strong link between payload mass and landing success, implying that other factors like launch site and booster type have a greater impact.

Conclusions

Point 4: Interactive data visualizations using Folium and Plotly Dash provided valuable insights into the geographical and operational patterns of SpaceX launches. These tools allowed for a deeper understanding of the data, enabling stakeholders to make informed decisions based on comprehensive visual analytics.

In conclusion, our predictive analysis and interactive visualizations have not only shed light on key factors influencing SpaceX's launch success but also provided a robust framework for future assessments and decision-making in the aerospace industry. The insights gathered can help improve launch strategies and contribute to the ongoing success of reusable rocket technology.

Thank you!