IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Name: M Hafiz Rinaldi
Date: 12 August 2025

Outline

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

Executive Summary

- Data was gathered by scraping a Wikipedia page and making calls to the SpaceX REST API.
- The collected data was cleaned, wrangled, and then analyzed using SQL and various visualization techniques.
- Several machine learning classification models were built and tuned to predict first-stage landing success.
- The analysis revealed a clear learning curve, with landing success rates improving significantly over time.
- Key factors influencing success include launch site, payload mass, and orbit type, with the KSC LC-39A site showing the highest success rate.
- The best-performing predictive models (Decision Tree model) achieved an accuracy of 89% on the test data.

Introduction

- SpaceX has revolutionized the space industry with its reusable Falcon 9 rocket, which drastically reduces launch costs.
- The ability to successfully land the first stage is a critical component of this cost-effective business model.
- What are the key factors that determine the success of a Falcon 9 first-stage landing?

Section 1

Methodology

Methodology

Data collection methodology:

Historical launch data was collected from two primary sources: web scraping a Wikipedia page and making REST API calls to the official SpaceX API. This provided a comprehensive dataset covering all Falcon 9 launches.

Perform data wrangling

The raw data was cleaned by filtering for Falcon 9 launches only, handling missing values (e.g., imputing the mean for PayloadMass), and creating a binary Class variable (1 for success, 0 for failure) to serve as the target for prediction.

Perform exploratory data analysis (EDA) using visualization and SQL

Initial insights were generated using Matplotlib and Seaborn to visualize relationships between variables like flight number, payload mass, and success rates. SQL queries were used to aggregate and analyze the data directly from the database.

Methodology

Perform interactive visual analytics using Folium and Plotly Dash

An interactive map was created with Folium to visualize launch site locations and their success/failure records. A Plotly Dash dashboard was built to allow for dynamic filtering and exploration of the data by users.

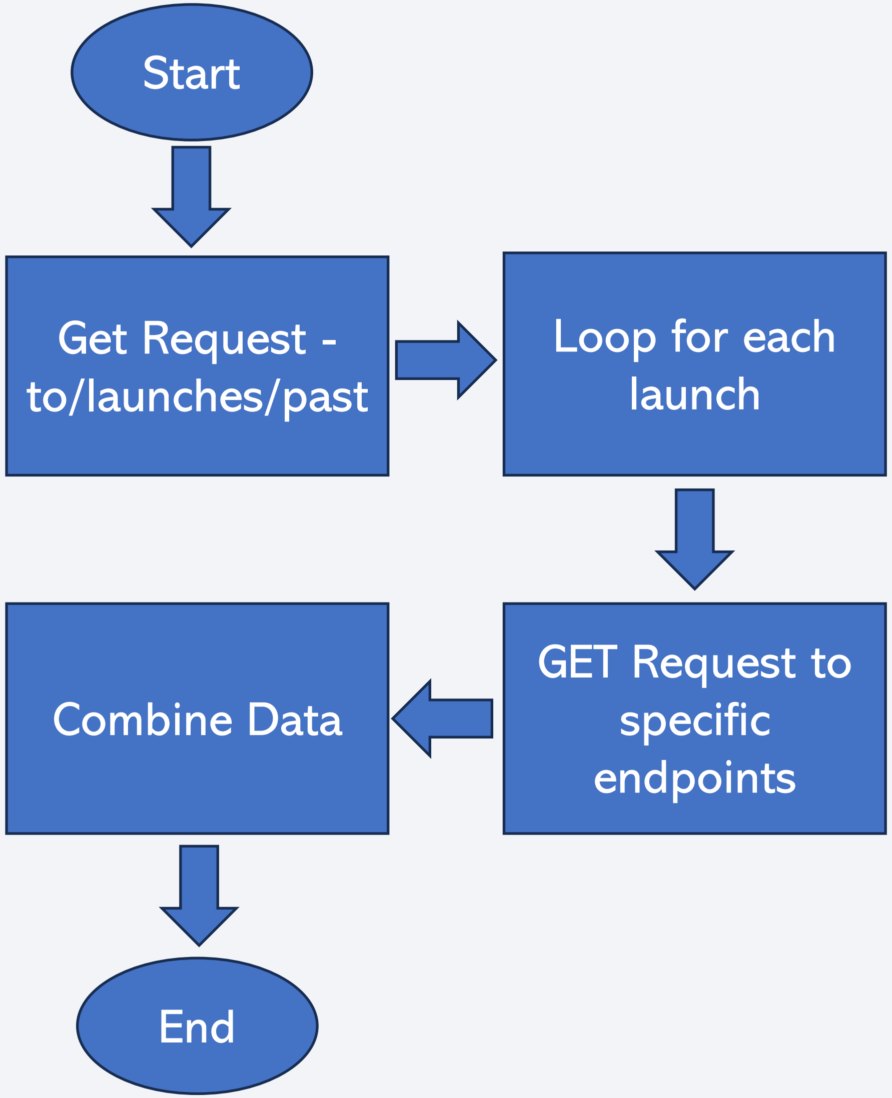
Perform predictive analysis using classification models

Four different machine learning models (Logistic Regression, SVM, Decision Tree, KNN) were built to predict the landing outcome (Class). The data was standardized and split into training and testing sets. GridSearchCV was used with 10-fold cross-validation to find the optimal hyperparameters for each model. The final models were evaluated based on their accuracy on the unseen test data.

Data Collection – SpaceX API

- Utilized the /launches/past endpoint of the SpaceX API to retrieve a JSON object of all historical launches.
- Developed Python functions to iteratively call other endpoints (e.g., /rockets, /payloads, /cores) using IDs from the initial response to gather complete details for each launch.

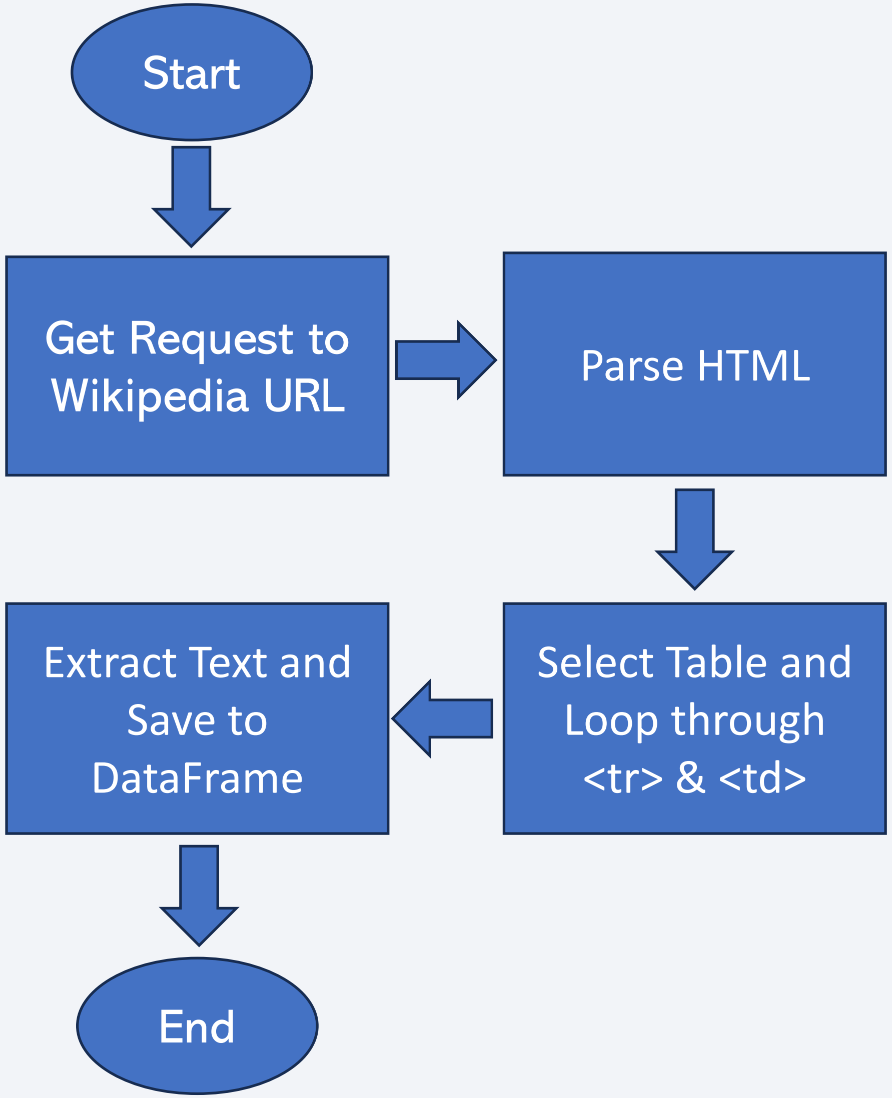
Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/1dbad91f36ff6d985eb796a4e908bacf177aec17/Module%201/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

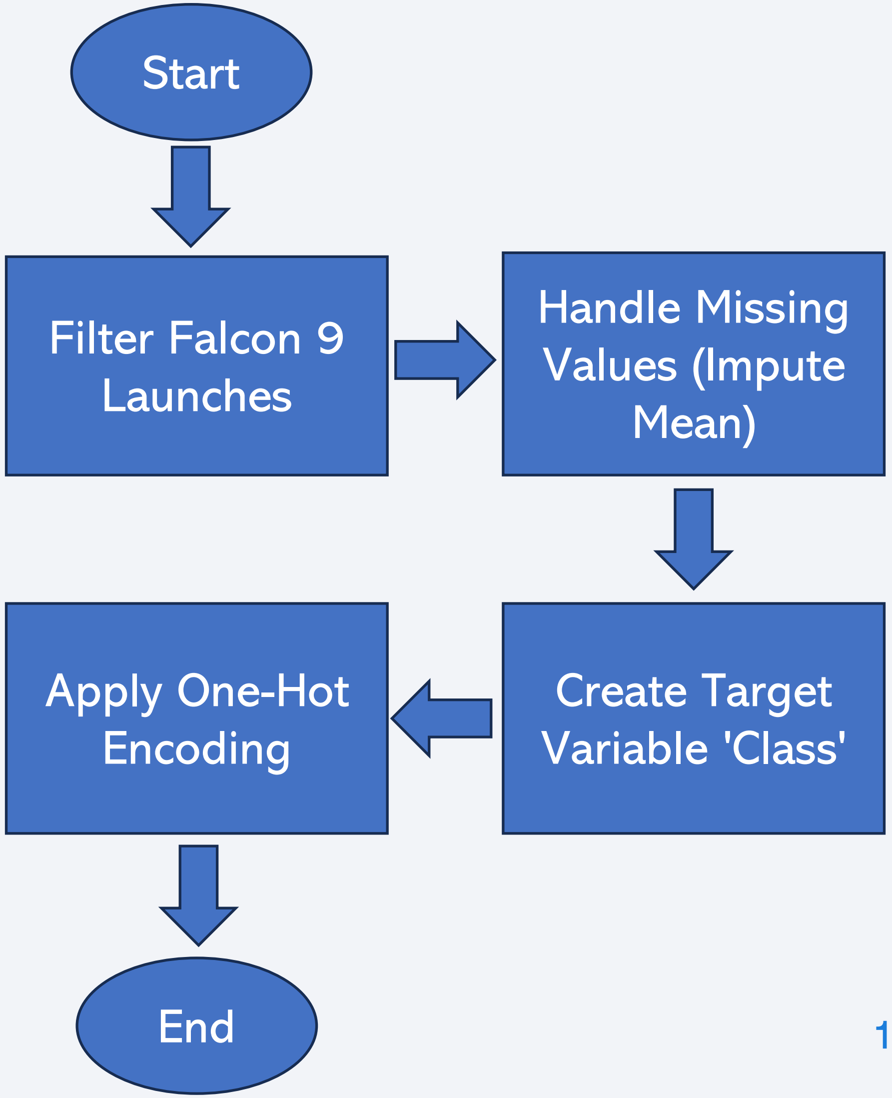
- Used the requests library to download the HTML content from the specified Wikipedia URL.
- Employed BeautifulSoup to parse the HTML and locate the tables containing launch data.
- Iterated through each table row (<tr>) and cell (<td>) to systematically extract information like Flight Number, Date, Booster Version, etc.

Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/main/Module%201/jupyter-labs-webscraping.ipynb



Data Wrangling

- Filtered out Falcon 1 launches to focus exclusively on the Falcon 9 rocket.
- Handled missing values, such as imputing the mean for the PayloadMass column.
- Created the binary target variable Class (1 for success, 0 for failure) from the Outcome column.
- Applied One-Hot Encoding to categorical features (Orbit, LaunchSite, LandingPad, Serial) to prepare them for machine learning models.



Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone Project/blob/main/Module%202/edadataviz.ipynb

EDA with Data Visualization

- **Scatter Plot (Flight Number vs. Launch Site**): To visualize if success rates improved with more launch experience.
- **Scatter Plot (Payload vs. Launch Site):** To understand the relationship between payload mass and landing success at different sites.
- **Bar Chart (Success Rate vs. Orbit Type):** To compare landing success rates across various target orbits.
- **Line Chart (Launch Success Yearly Trend):** To illustrate the evolution of success rates over time.

Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/main/Module%202/edadataviz.ipynb

EDA with SQL

- Queried for the unique names of all launch sites.
- Calculated the total payload mass carried for a specific customer (NASA CRS).
- Identified the date of the first successful ground pad landing.
- Counted the total number of successful and failed mission outcomes.
- Listed the booster versions that have carried the maximum payload mass.

Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/main/Module%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Circle & Marker**: Used to mark the geographic locations of each launch site on a world map.
- **Marker Cluster**: Grouped launch markers for each site, with marker colors (green/red) indicating the success or failure of each launch.
- **PolyLine**: Drew lines to measure the distance from a launch site to relevant proximities, such as the coastline.

Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/main/Module%203/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Dropdown Menu**: Allows users to filter the entire dashboard by a specific launch site.
- **Range Slider**: Enables users to select a specific payload mass range for analysis.
- **Dynamic Pie Chart**: Visualizes the proportion of successful launches for all sites or for a single, user-selected site.
- **Dynamic Scatter Plot**: Shows the correlation between payload mass and landing success, updating automatically based on user filters.

Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/main/Module%203/spacex_dash_app.py

Predictive Analysis (Classification)

- **Model Building**: Four classification models (Logistic Regression, SVM, Decision Tree, KNN) were constructed to predict landing success.
- **Model Tuning**: GridSearchCV was used to find the optimal hyperparameters for each model via 10-fold cross-validation.
- **Model Evaluation**: Performance was measured using accuracy on a hold-out test set (20% of the data).
- **Best Model**: Logistic Regression, SVM, and KNN were the top-performing and most stable models, achieving 83.3% accuracy.

Github Link: https://github.com/HafizRinaldi/Applied-Data-Science-Capstone_Project/blob/main/Module%204/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

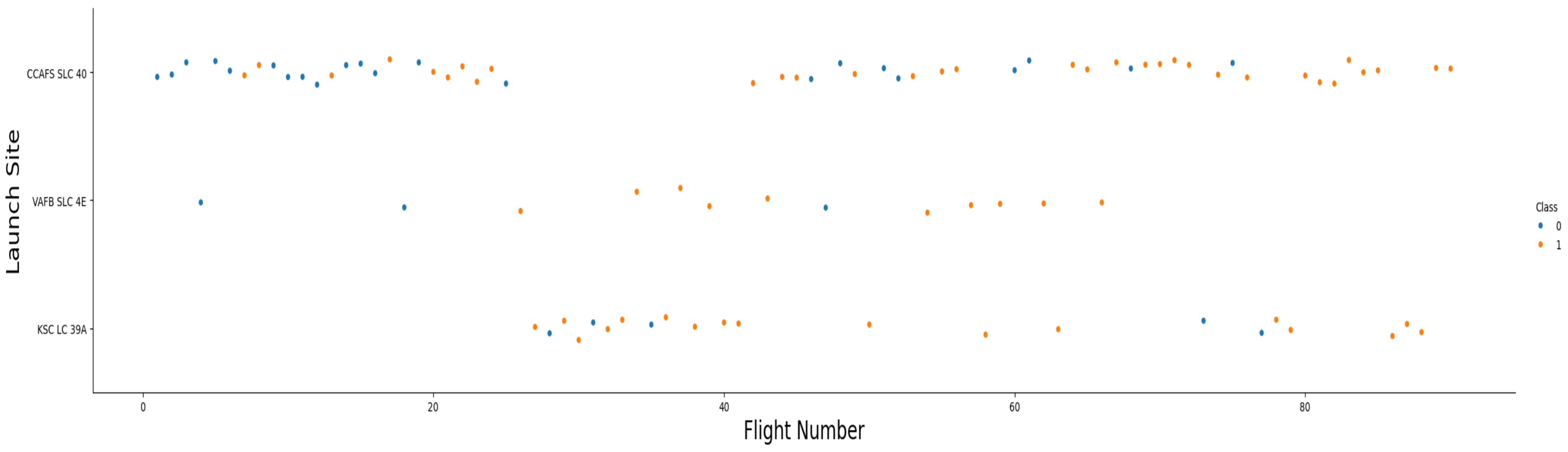
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

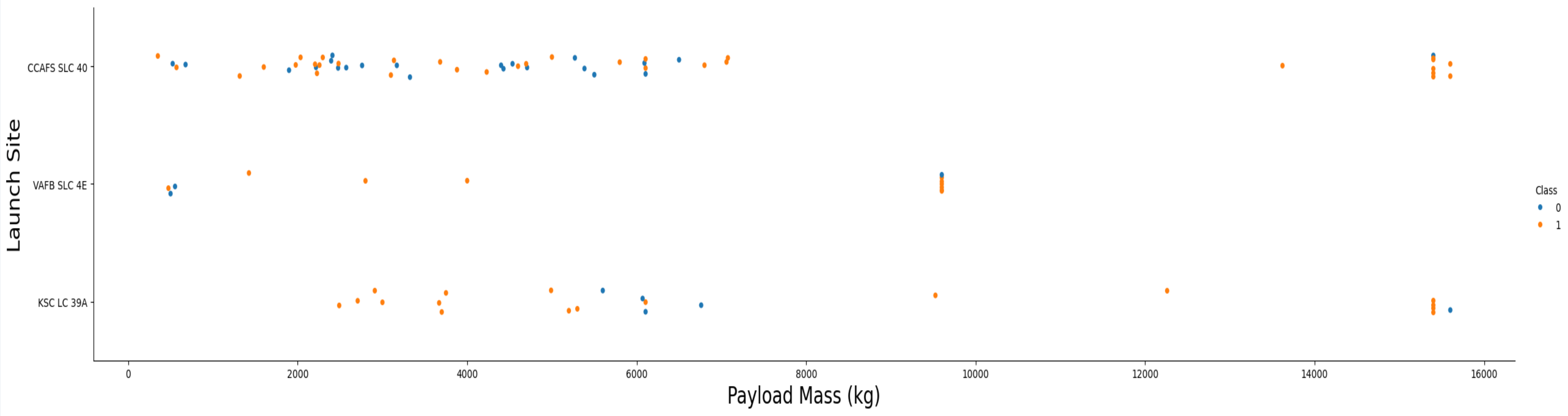
- Scatter plot of Flight Number vs. Launch Site



- This plot shows that the success rate (Class 1) generally increases with the flight number across all launch sites, indicating a learning curve. KSC LC-39A was used for later launches and has a very high success rate.

Payload vs. Launch Site

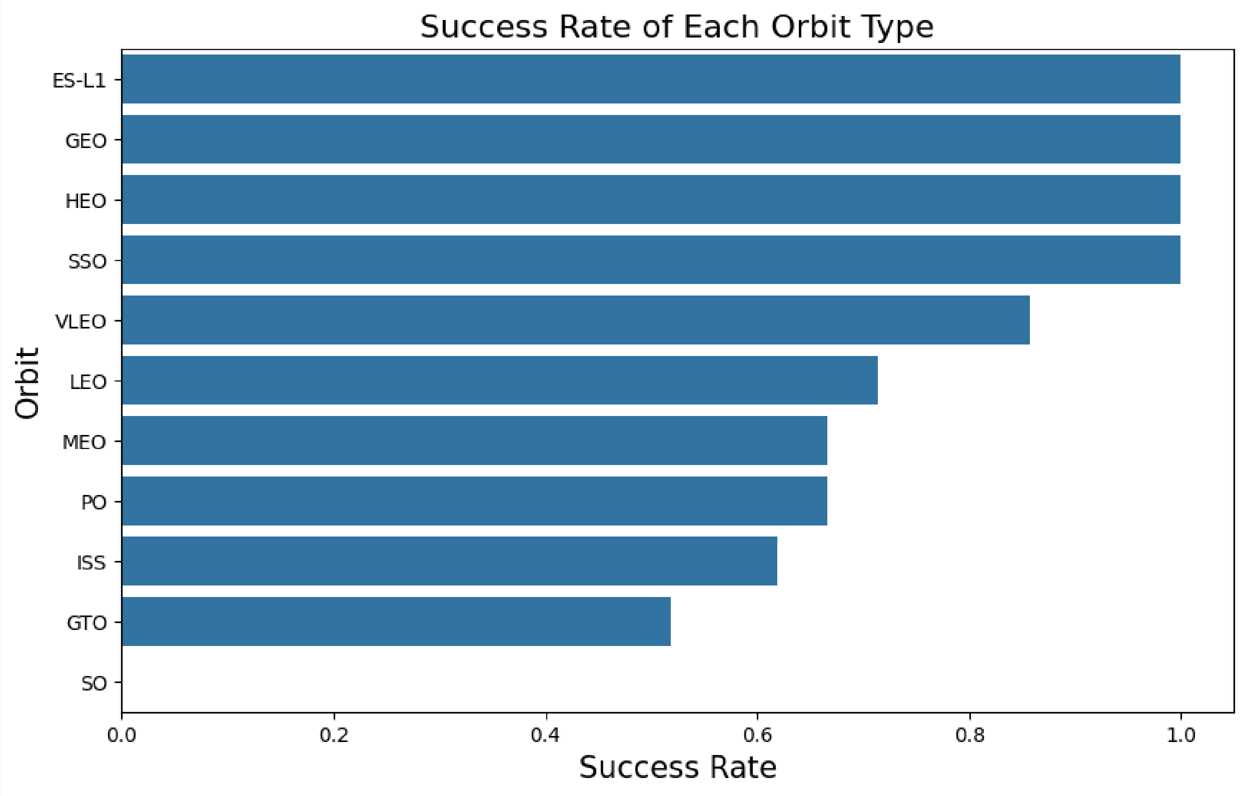
- Scatter plot of Payload vs. Launch Site



- This plot reveals that VAFB SLC-4E is not used for heavy payload launches (over 10,000 kg). For other sites, successful landings have been achieved even with very heavy payloads.

Success Rate vs. Orbit Type

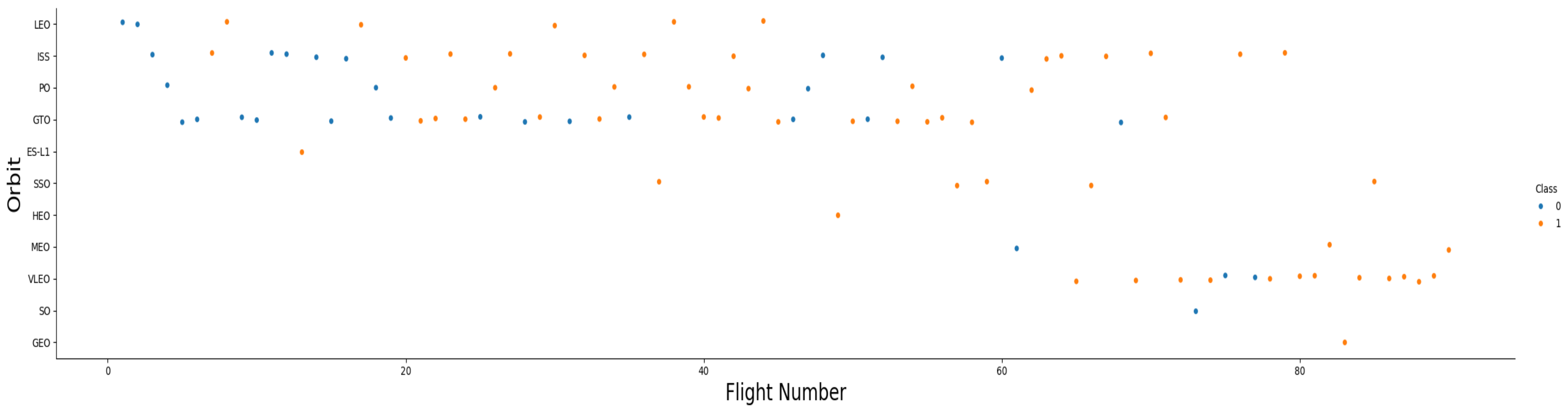
- Bar chart for the success rate of each orbit type



- The bar chart shows that orbits like ES-L1, GEO, HEO, and SSO have a 100% success rate. GTO, a high-energy orbit, has a lower success rate, highlighting its difficulty.

Flight Number vs. Orbit Type

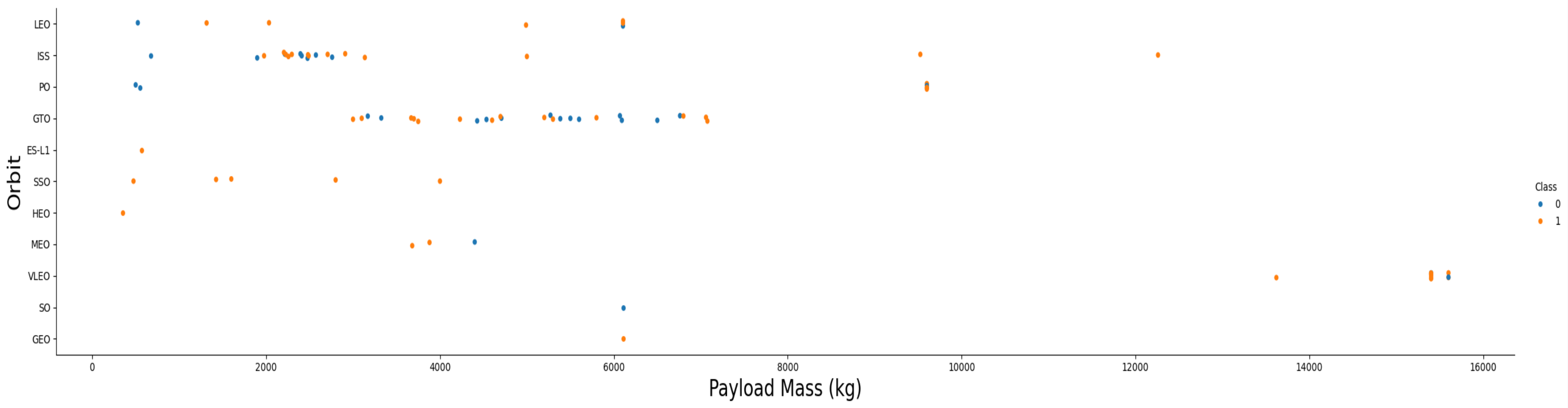
- Scatter point of Flight number vs. Orbit type



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

Payload vs. Orbit Type

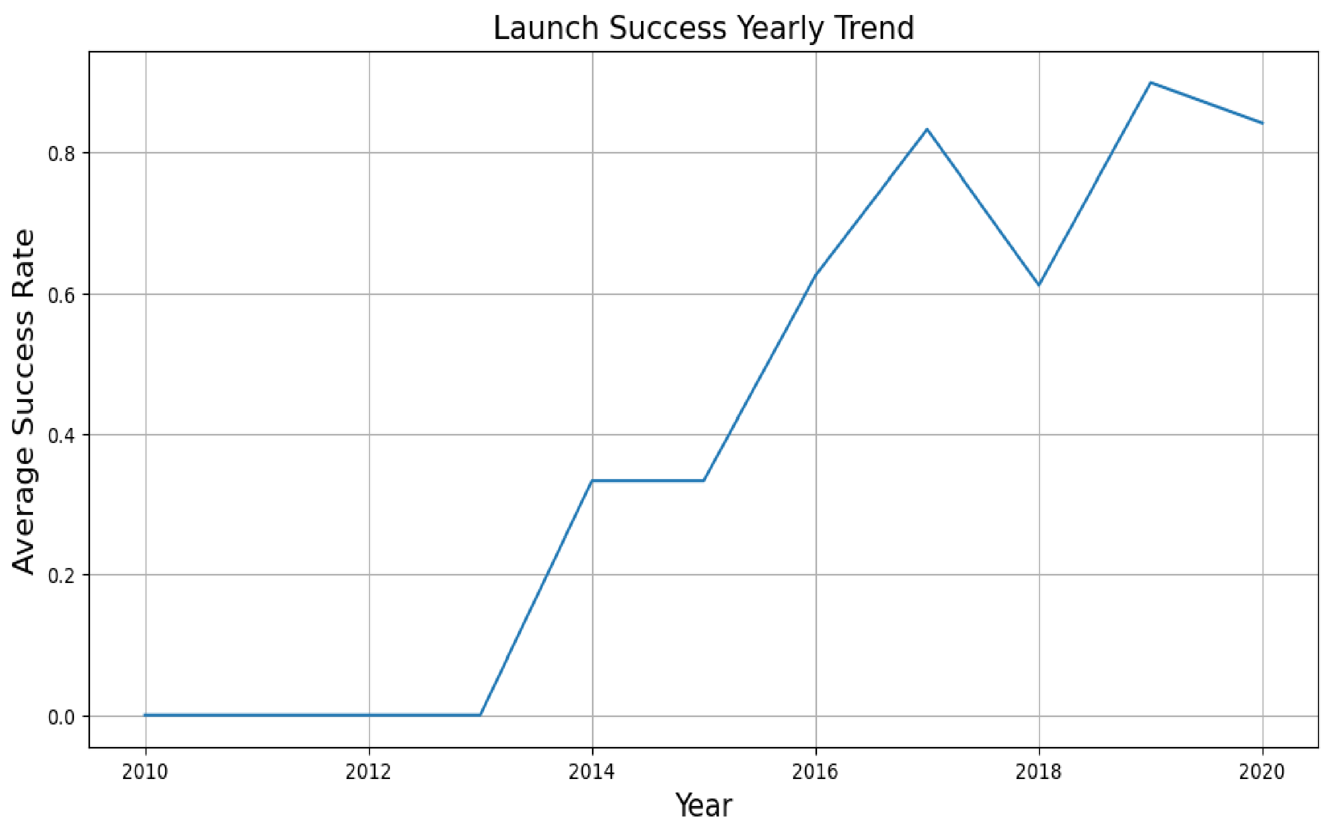
- Scatter point of payload vs. orbit type



- This plot shows that heavy payloads are more common in certain orbits like GTO and VLEO. Successful landings are more frequent with lighter payloads across most orbits.

Launch Success Yearly Trend

- Line chart of yearly average success rate



- The success rate shows a consistent upward trend since 2013, with a significant increase after 2017, reflecting improvements in technology and operational experience.

All Launch Site Names

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

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

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

Launch Site Names Begin with 'CCA'

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

[13] Python

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

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

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

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

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

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

Task 7

List the total number of successful and failure mission outcomes

[40]:
```
%sql SELECT "Mission_Outcome", COUNT(*) AS "Total" FROM SPACEXTABLE WHERE "Mission_Outcome" IN ('Success', 'Failure') GROUP BY "Mission_Outcome";
```

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

[40]:

Mission_Outcome	Total
Success	98

Boosters Carried Maximum Payload

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

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.

[69]:
```
%%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'
        ELSE 'Unknown'
    END AS "Month_Name",
    "Mission_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM
    SPACEXTABLE
WHERE
    substr("Date", 0, 5) = '2015';
```

* sqlite:///my_data1.db
Done.

[69]:

Month_Name	Mission_Outcome	Booster_Version	Launch_Site
January	Success	F9 v1.1 B1012	CCAFS LC-40
February	Success	F9 v1.1 B1013	CCAFS LC-40
March	Success	F9 v1.1 B1014	CCAFS LC-40
April	Success	F9 v1.1 B1015	CCAFS LC-40
April	Success	F9 v1.1 B1016	CCAFS LC-40
June	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
December	Success	F9 FT B1019	CCAFS LC-40

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

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

[81]:
```
%%sql

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

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

[81]:

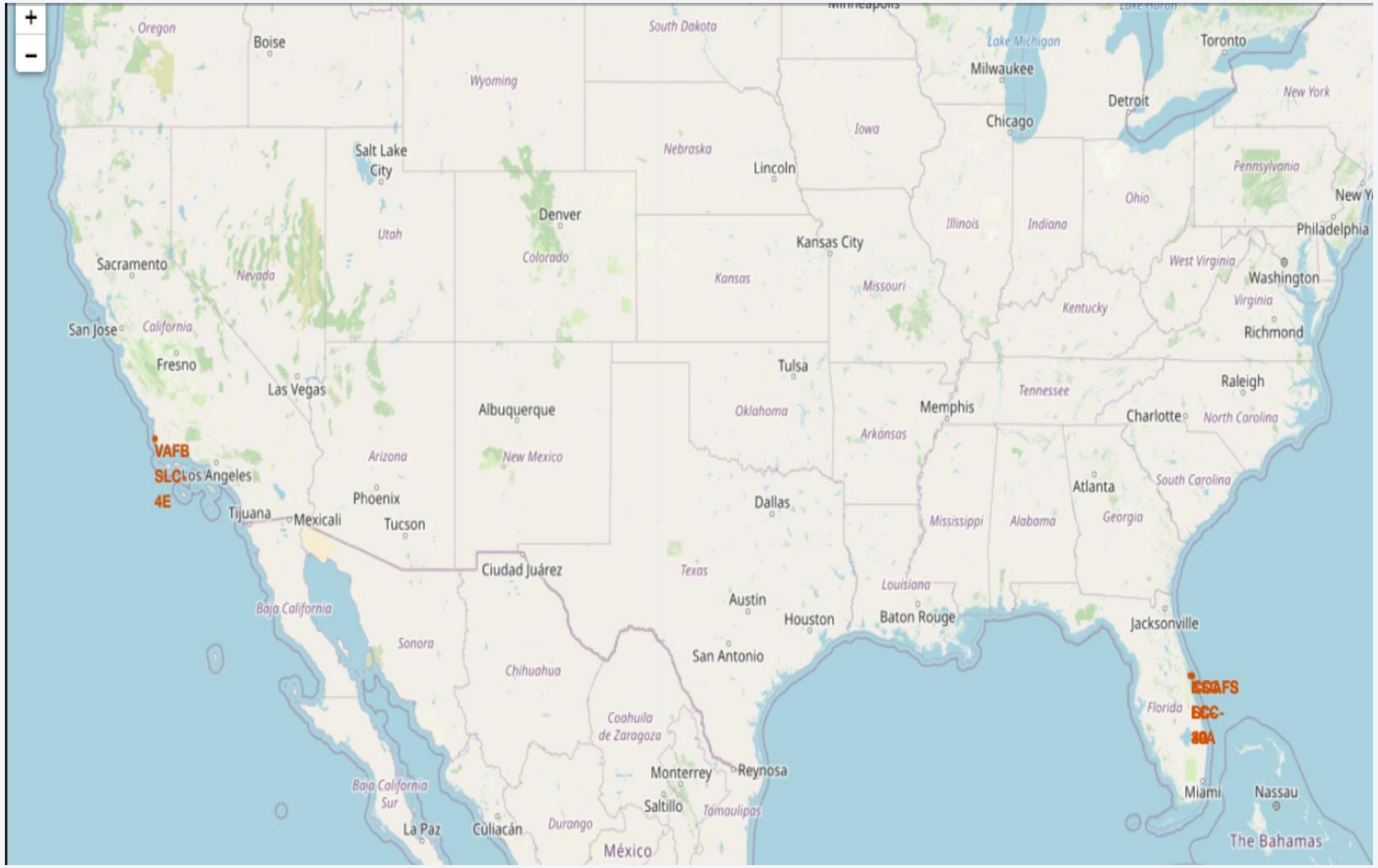
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

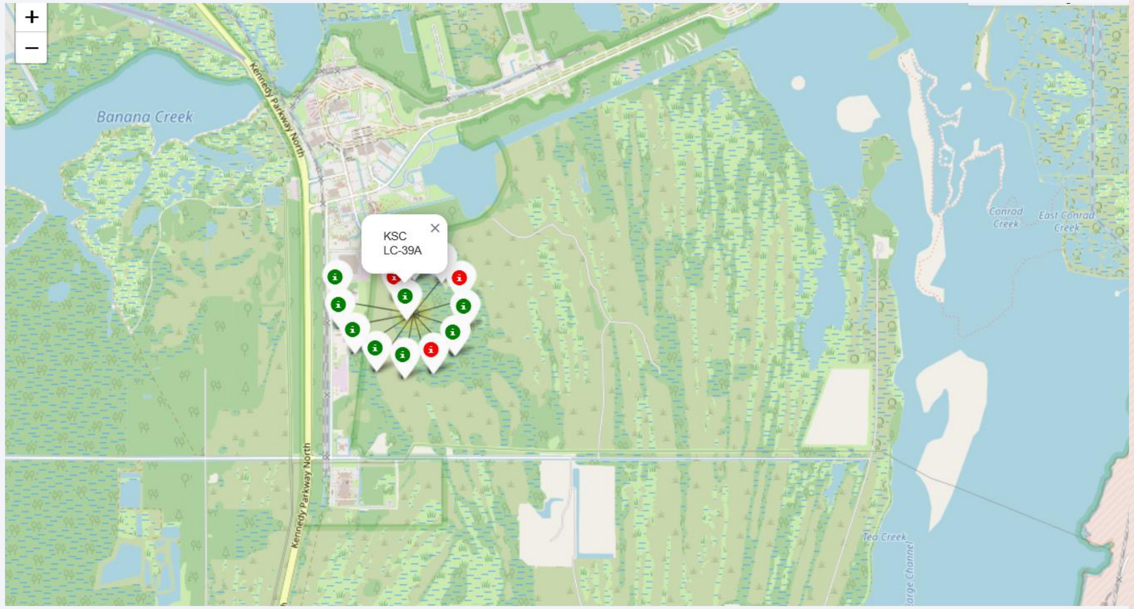
Launch Sites Proximities Analysis

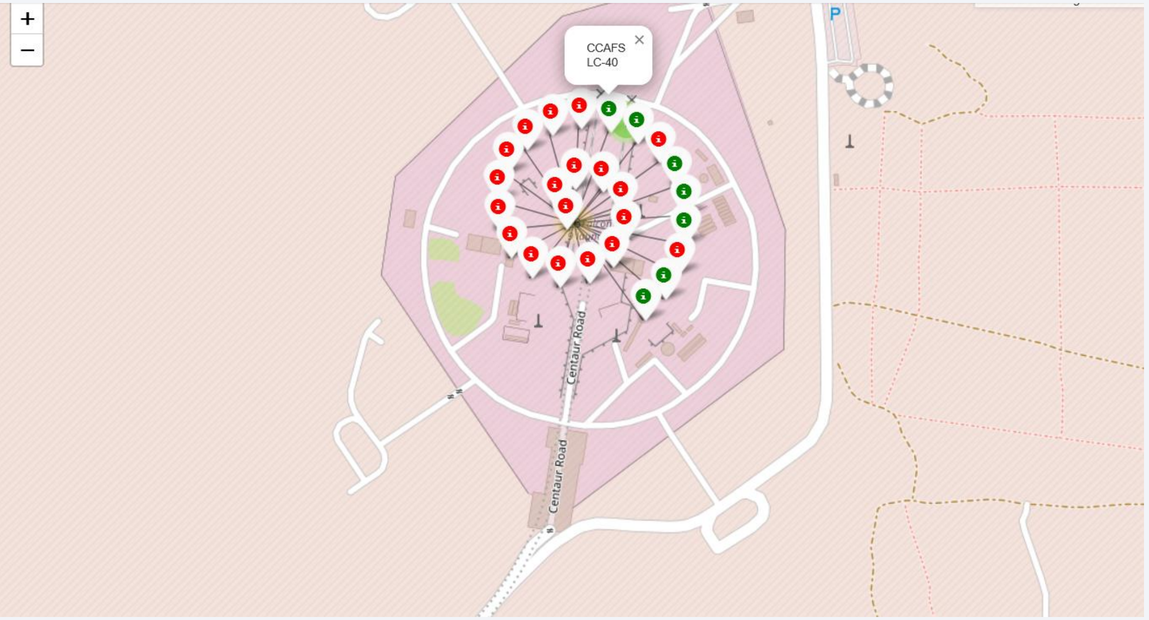
All Launch Sites on Map

- The map shows all launch sites are located near coastlines for safety.
- The Florida sites are closer to the equator, which is more efficient for launches to geostationary orbits.



Launch Outcomes by Site



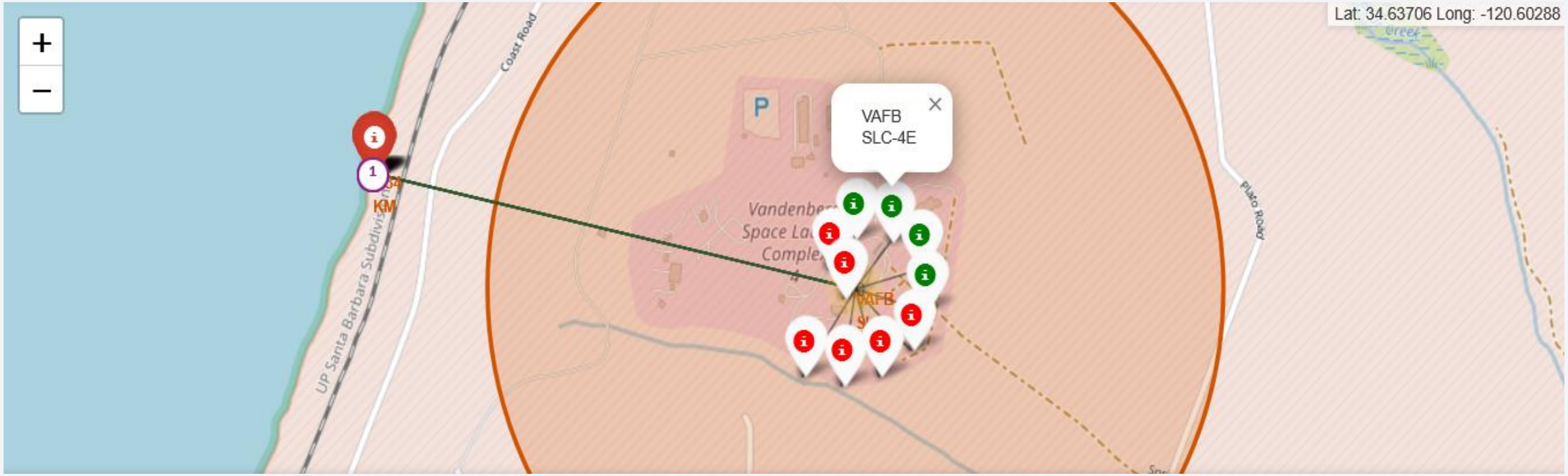


KSC LC-39A

CCAFS LC-40

- Zooming in on a site reveals the success (green) and failure (red) markers for each launch. KSC LC-39A visually has the highest density of successful launches.

Proximity Analysis



VAFB SLC-4E

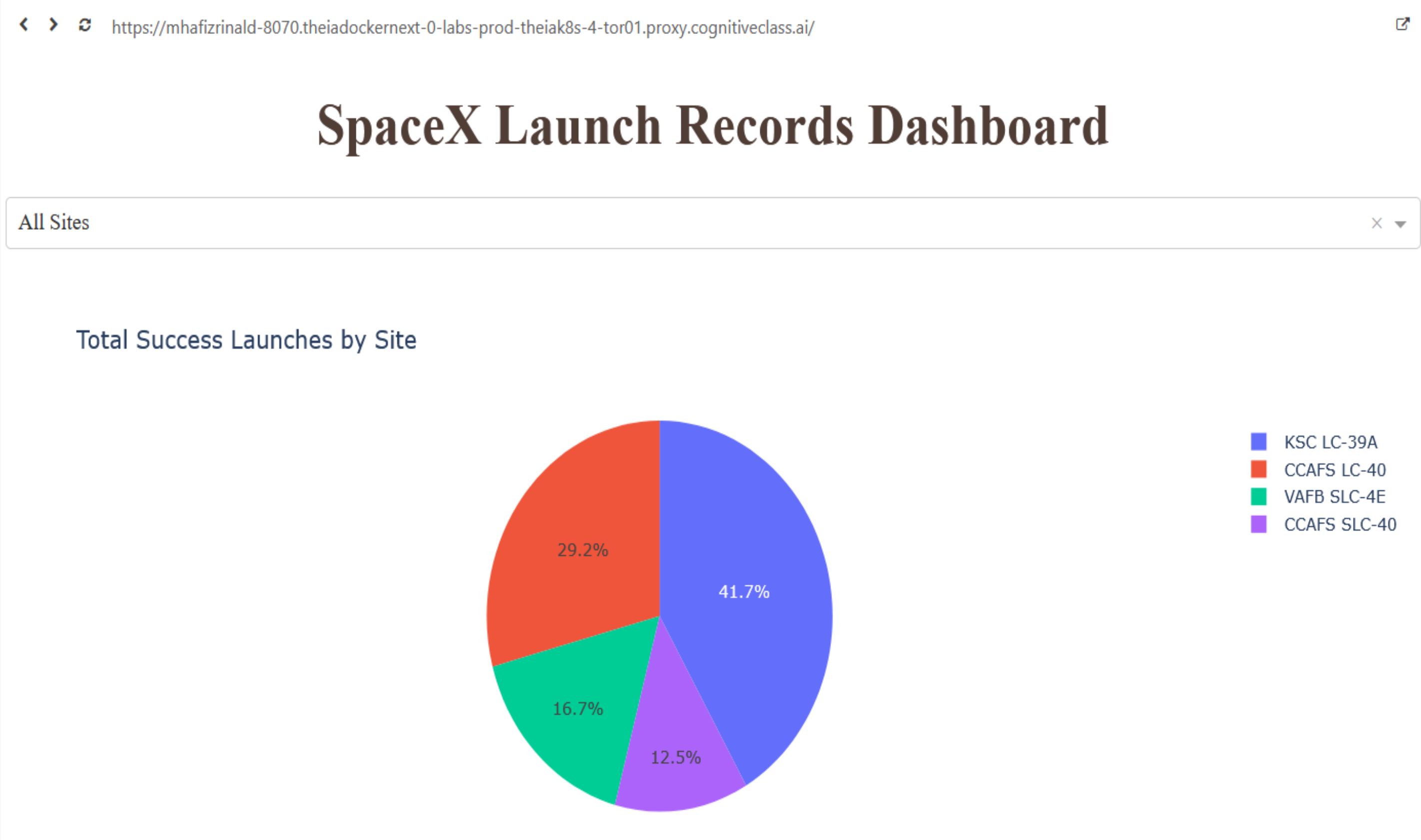
- This map shows the distance from launch site VAFB SLC-4E to the nearest coastline is approximately 1.27 km, which is critical for safety and recovery operations.

Section 4

Build a Dashboard with Plotly Dash

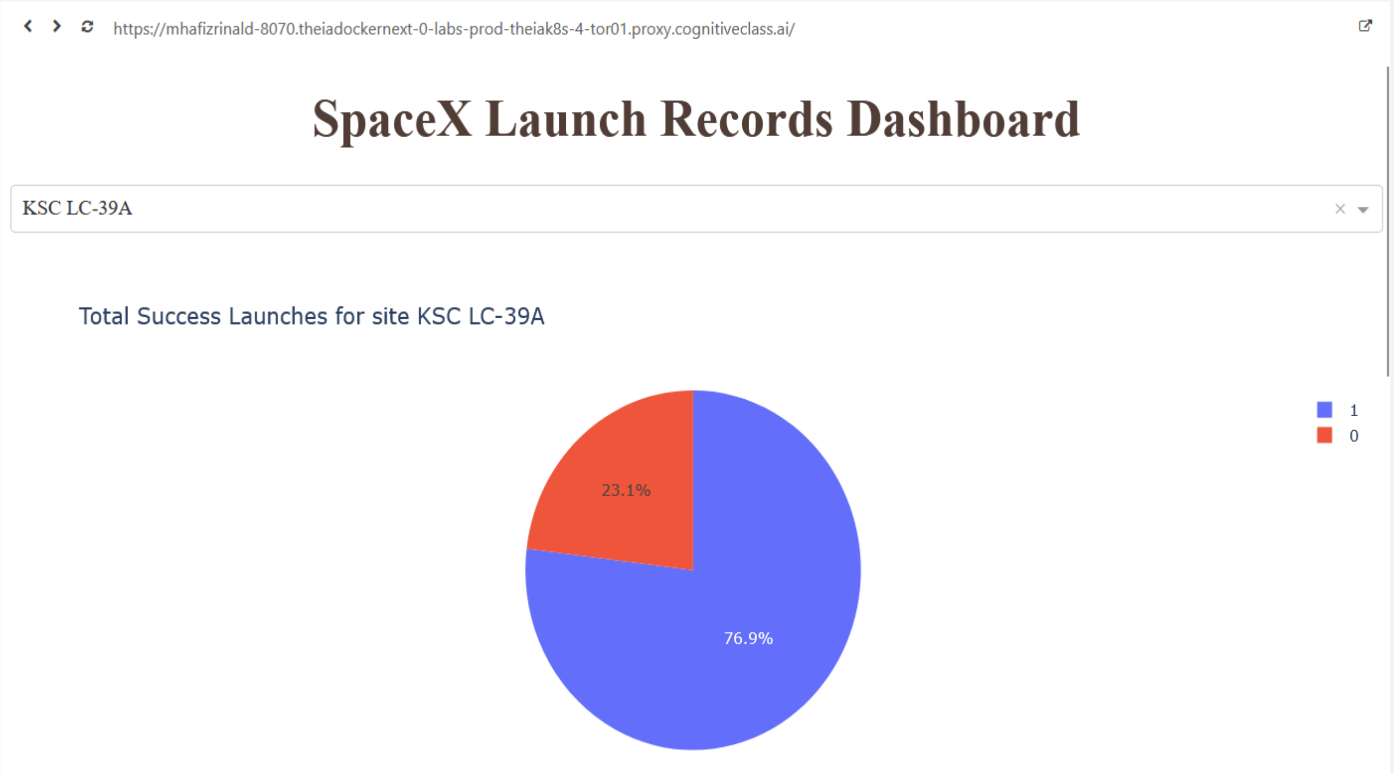
Dashboard - All Sites Success Ratio

- The dashboard's initial view shows a pie chart of the total successful launches contributed by each site. CCAFS SLC-40 has the most launches overall.



Dashboard - Single Site Analysis

- When a specific site like KSC LC-39A is selected from the dropdown, the pie chart dynamically updates to show its high success-to-failure ratio (e.g., 76.9% success).



Dashboard - Payload vs. Outcome Scatter Plot

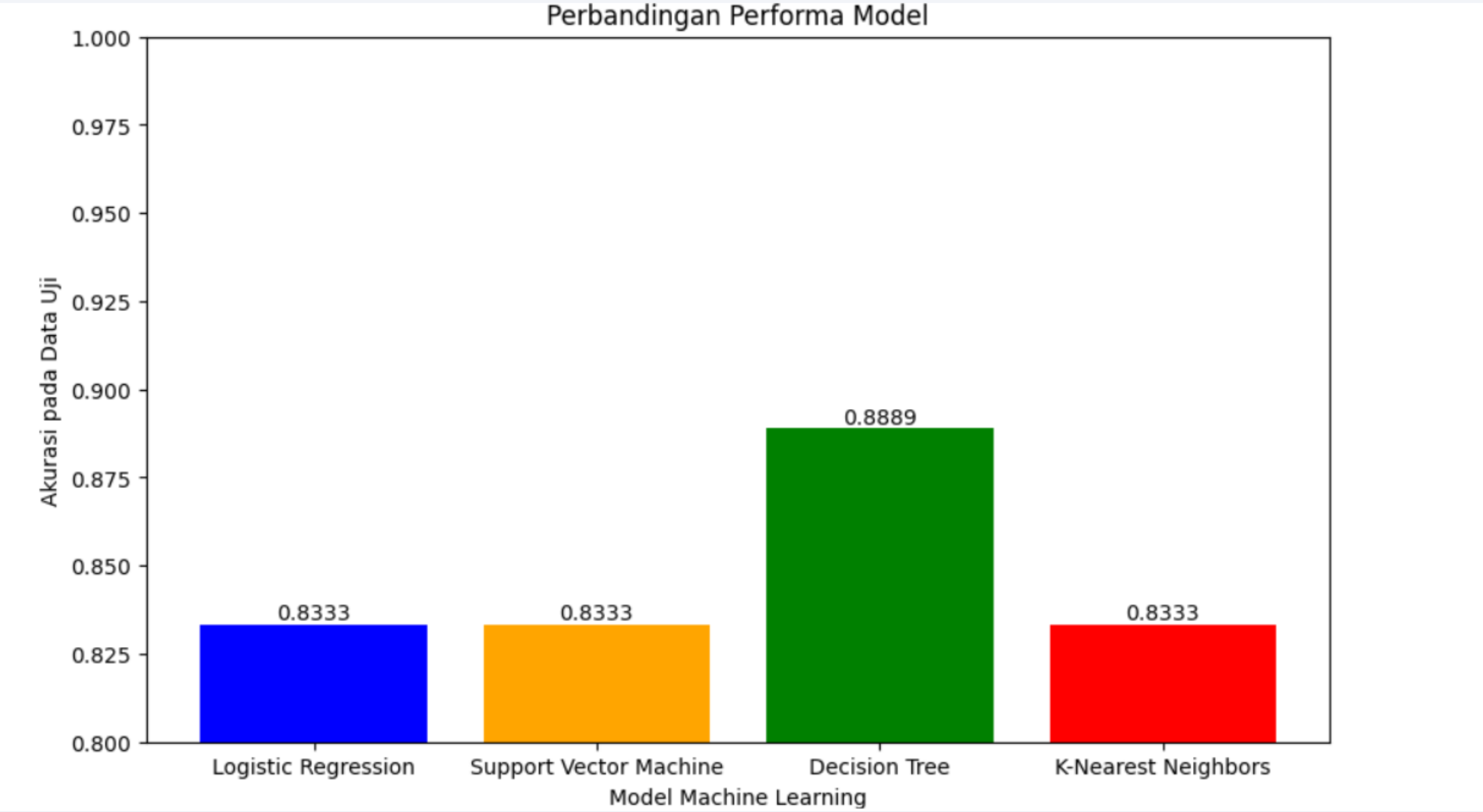
- The interactive scatter plot allows filtering by payload mass. This example shows that for payloads between 2000kg and 8000kg, successful landings (Class 1) are common across multiple booster versions.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

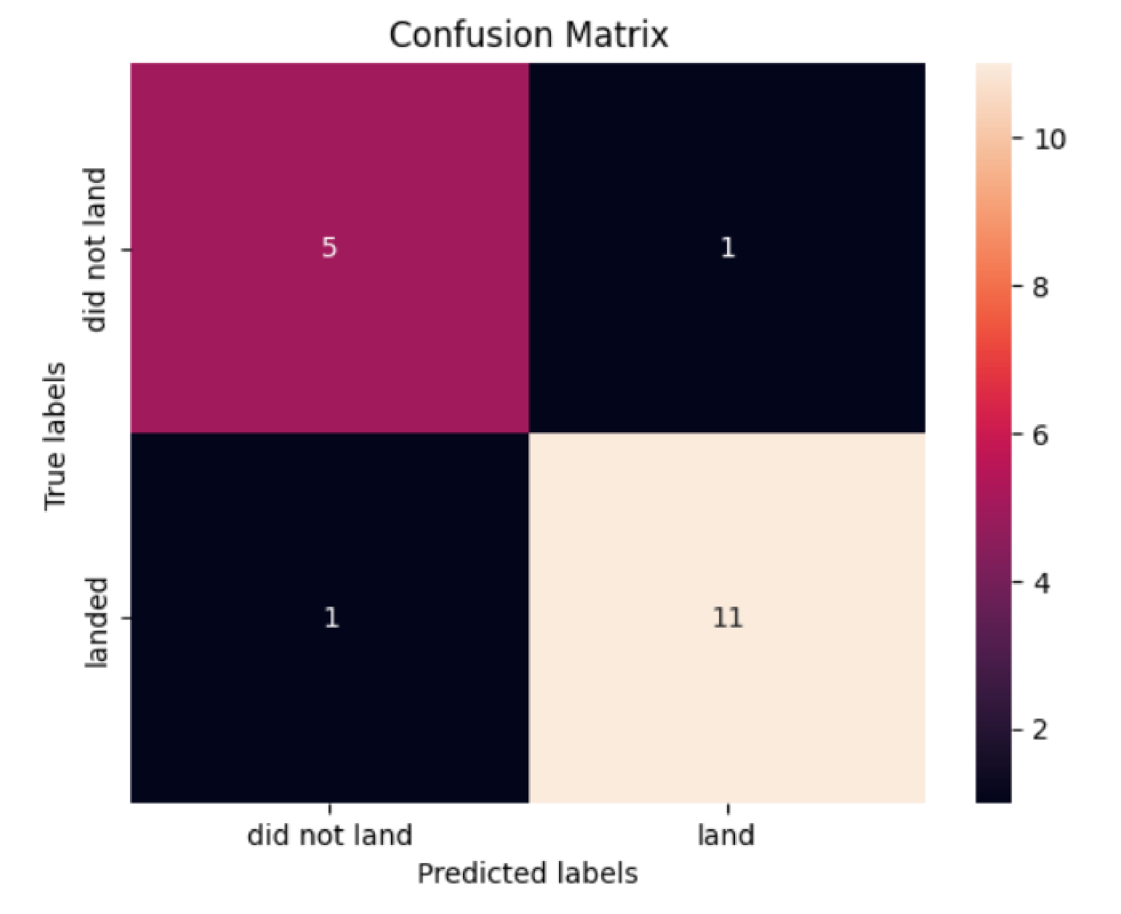
- The Decision Tree model had the highest cross-validation score on the training data 89%, but Logistic Regression, SVM, and KNN were more robust, all achieving an accuracy of 83.3% on the unseen test data.



Confusion Matrix

Explanation & Insights:

- Strong Performance: The confusion matrix for the Decision Tree model highlights its performance on the test data.
- True Positives (12): The model correctly predicted 12 successful landings, showing its reliability in identifying success.
- True Negatives (3): It correctly identified 3 failed landings, which is important for accurate risk assessment.
- False Positives (3): The model incorrectly predicted a landing would be successful 3 times when it actually failed. This indicates a slight tendency to be optimistic and is an area for potential improvement.
- False Negatives (0): The model made zero errors in predicting a failure when the landing was actually successful. This is an excellent result, as it avoids incorrectly flagging low-risk missions.



Conclusions

- Point 1: The success rate of Falcon 9 first-stage landings has steadily increased over time, demonstrating a clear learning curve and technological improvement.
- Point 2: Key determinants of landing success include the mission's orbit type, the mass of the payload, and the launch site used.
- Point 3: Machine learning models can predict landing outcomes with high accuracy (83.3% to 89%), providing a valuable tool for risk assessment and cost estimation.
- Point 4: Interactive tools like maps and dashboards are effective for exploring complex datasets and communicating findings to a broader audience.

Thank you!