IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

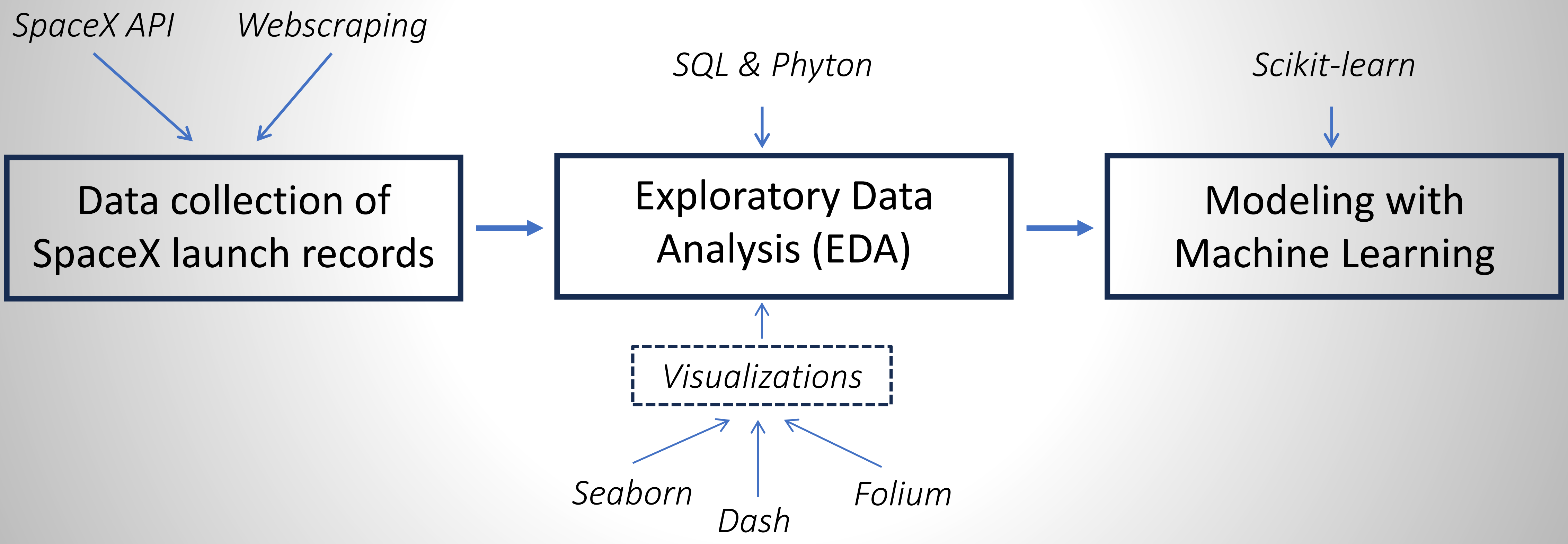
Marcos Birchner
13/11/2025

Outline

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

Executive Summary

- Methodologies



Executive Summary

❑ Summary of results

- ✓ The analysis revealed key relationships among features such as launch site, payload mass, orbit type, flight number and booster version, all of which influence the success of a landing mission. Using Folium, launch sites were mapped to identify geographic trends in landing success.

- ✓ Multiple classification models were tested, and the Decision Tree Classifier achieved the best accuracy. This model can predict whether the first stage will land successfully, enabling more informed pricing decisions for future launches.

Introduction

SpaceX advertises Falcon 9 rocket launches at a cost of 62 million dollars, while other providers charge upwards of 165 million dollars per launch. A major reason for this price advantage is that SpaceX can reuse the first stage of its rockets.

The goal of this project is to use data from past launches to build a model that, based on specific features of a planned mission, can predict whether the first stage of a Falcon 9 rocket will land successfully.

Therefore, if we can predict whether the first stage will successfully land, we can estimate the cost of a launch. This information could be valuable for competing companies that want to bid against SpaceX on future launch contracts.

Section 1

Methodology

Methodology

❑ Executive Summary

I. Data collection methodology:

- Data was retrieved from the SpaceX API using HTTP requests and JSON responses.

II. Perform data wrangling.

- Extracted and combined features from multiple endpoints; filtered dataset to include only Falcon 9 launches.

III. Perform exploratory data analysis (EDA) using visualization and SQL.

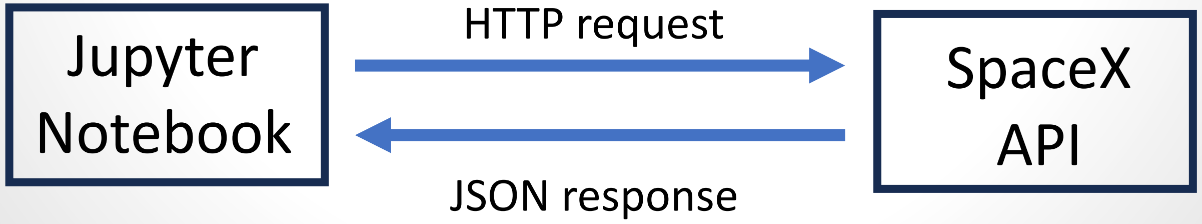
IV. Perform interactive visual analytics using Folium and Plotly Dash.

V. Perform predictive analysis using classification models.

- Tested Logistic Regression, SVM, K-Nearest Neighbors, and Decision Tree to predict first-stage landing success.
- Tuned models using GridSearchCV and evaluated them with accuracy scores.

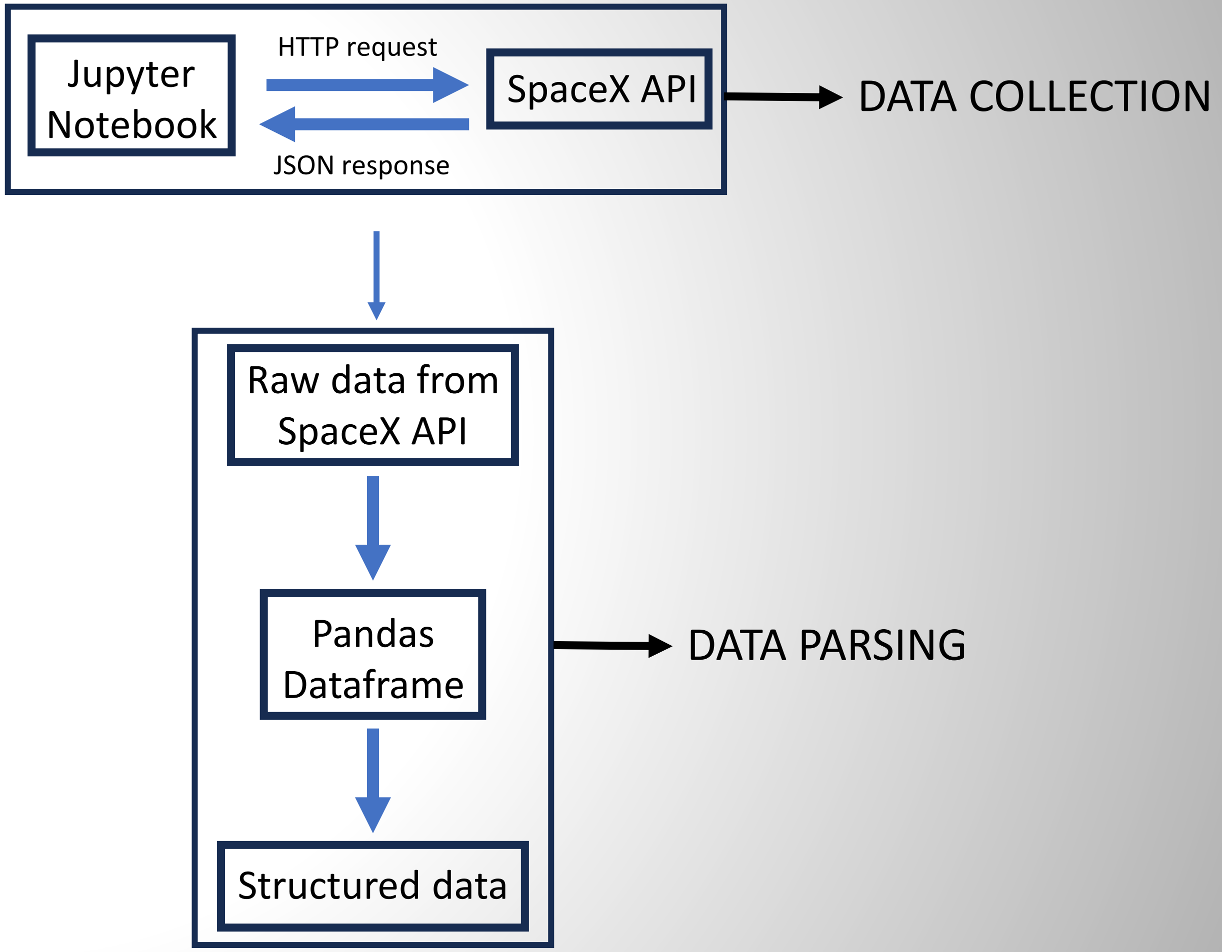
Data Collection

Data was collected by sending **HTTP requests** to the **SpaceX API**. The responses were returned in **JSON format** and parsed to extract relevant data. Additional API calls were made to access detailed information from related endpoints using launch IDs — specifically for rockets, payloads, launchpads, and cores.



Data Collection – SpaceX API

- The API response is received as raw data in JSON format, which is then parsed and organized for analysis. This raw, unstructured data is transformed into a structured format by loading it into a Pandas DataFrame, enabling efficient processing, exploration, and visualization.

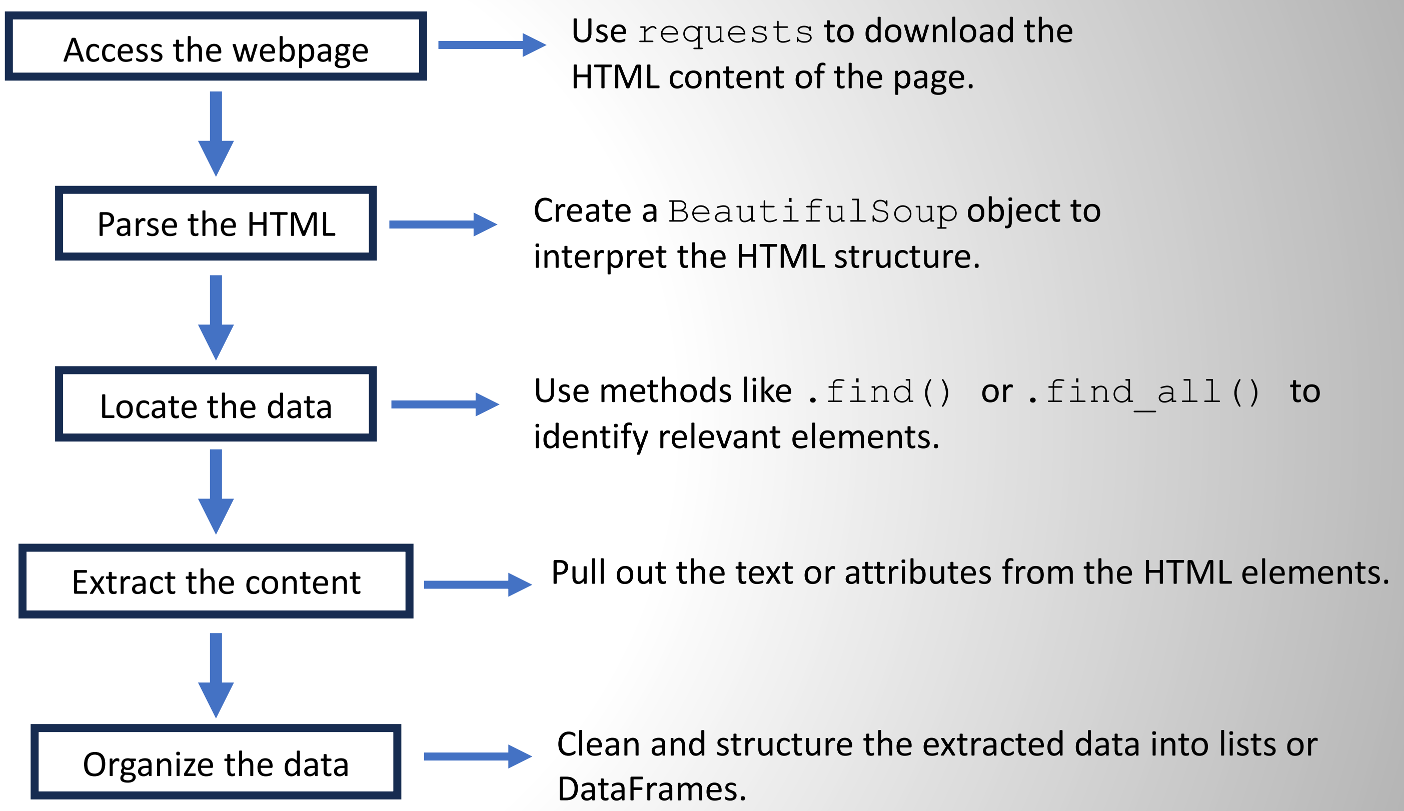


➢ The completed Data Collection - SpaceX API notebook is available on GitHub for external reference and peer-review purposes:

https://github.com/MarcosDS12/Capstone/blob/350282d12cb0359e4be900cc3cf714319279976f/Data%20Collection%20SpaceX%20API.ipynb

Data Collection - Scraping

- This workflow outlines the essential steps of web scraping using BeautifulSoup. Starting with accessing the webpage and parsing its HTML content, we identify and extract the desired data. The information is then cleaned and structured, making it ready for analysis or storage.



➢ The completed web scraping notebook is available on GitHub for external reference and peer-review purposes.

https://github.com/MarcosDS12/Capstone/blob/e125432b062f872c63993fd98662b7ead263f6ce/Data%20Collection%20Webscraping%20Space%20X.ipynb

Data Wrangling

- The data wrangling process involved taking the raw data collected from APIs and web scraping, and preparing it for analysis. This included cleaning missing values, converting columns to appropriate data types, extracting relevant fields, and organizing the information into a structured DataFrame format. These transformations ensured consistency and accuracy, enabling effective analysis in subsequent steps.

1. Load raw or semi-structured data → Data from previous API or scraping steps is imported (e.g., CSV, JSON, DataFrame).
2. Clean missing or inconsistent values → Empty fields, incorrect types, and anomalies are identified and fixed.
3. Normalize and standardize formats → Dates, text values, and numeric formats are aligned to a consistent structure.
4. Transform columns as needed → Extract or combine columns, convert data types (e.g., `str → float`), etc.
5. Reorganize the dataset → Reset indices, rename columns, or sort data for downstream use.

➢ The completed Data Wrangling notebook is available on GitHub for external reference and peer-review purposes.
https://github.com/MarcosDS12/Capstone/blob/f1bd3087d01447e6e5ad6d2d57bfe12d1aabe855/Data%20wrangling%20Space%20X.ipynb

EDA with Data Visualization

Summary of Visualization Techniques

- ***Scatter Plots***

  Used to explore relationships between variables, such as payload mass vs. flight number or launch site vs. success rate.

  Example: Visualizing how payload mass affects mission outcome across launch sites.

- ***Bar Chart***

  Helpful to compare categorical data like success rates across orbit types.

  Example: Comparing average launch success by orbit category.

- **Line Plot**

  Effective for showing trends over time or sequential launches.

  Example: Observing how success rate improves over the sequence of flight numbers.

- The completed EDA with Data Visualization notebook is available on GitHub for external reference and peer-review purposes.
  https://github.com/MarcosDS12/Capstone/blob/757751f84ba5690e5bf01c78b91c117430a7623b/EDA%20with%20Data%20Visualization%20Space%20X.ipynb

EDA with SQL

Summary of SQL Queries Performed

- ❑ Created a new table excluding records with null dates for cleaner analysis.
- ❑ Retrieved all unique launch sites from the dataset.
- ❑ Filtered launches starting with “CCA” to inspect specific launch site data.
- ❑ Calculated total payload mass for NASA (CRS) missions.
- ❑ Computed average payload mass for the “F9 v1.1” booster version.
- ❑ Found the date of the first successful landing on a ground pad.
- ❑ Listed booster versions that landed successfully on drone ships with payloads between 4000–6000 kg.
- ❑ Counted mission outcomes by classifying them as success or failure.
- ❑ Identified booster versions that carried the maximum payload mass using a subquery.
- ❑ Displayed records showing month names, failed drone ship landings, booster versions, and launch sites from 2015.
- ❑ Ranked landing outcomes (e.g., failure/success) between two specific dates in descending order.

➢ The completed EDA with SQL notebook is available on GitHub for external reference and peer-review purposes.
https://github.com/MarcosDS12/Capstone/blob/6b8604a49a5675e9c73f5083df34412de0fa8e96/EDA%20with%20SQL%20Space%20X.ipynb

Build an Interactive Map with Folium

Map Elements Overview

To explore and visualize the geographic and operational context of SpaceX launch sites, various map objects were added using Folium:

- ✅ Markers were placed at each launch site to pinpoint locations
- ✅ Success rates were displayed next to each site using labeled icons
- ✅ Circle markers highlighted the launch zones visually
- ✅ Polylines connected launch sites to nearby cities, coastlines, or highways to show proximity
- ✅ Distance labels were added to indicate how far each feature was from the site

These elements made the map interactive and informative, helping to analyze how location factors might influence launch outcomes.

- The completed Interactive Map with Folium notebook is available on GitHub for external reference and peer-review purposes.
  https://github.com/MarcosDS12/Capstone/blob/6b8604a49a5675e9c73f5083df34412de0fa8e96/Interactive%20Map%20with%20Folium%20Space%20X.ipynb

Build a Dashboard with Plotly Dash

Plotly Dash Dashboard Overview

This dashboard was built to explore SpaceX launch outcomes interactively using **dropdowns**, **sliders**, and dynamic **charts**.

✅ **Plots and Interactions Included:**

•**Pie Chart:**

- Shows total successful launches by site
- Updates to show success vs. failure when a specific site is selected

•**Scatter Plot:**

- Displays the correlation between payload mass and launch outcome
- Color-coded by booster version
- Dynamically updates based on selected site and payload range

•**Dropdown Menu:**

- Allows selection of launch site (or all sites) to filter the data

•**Payload Range Slider:**

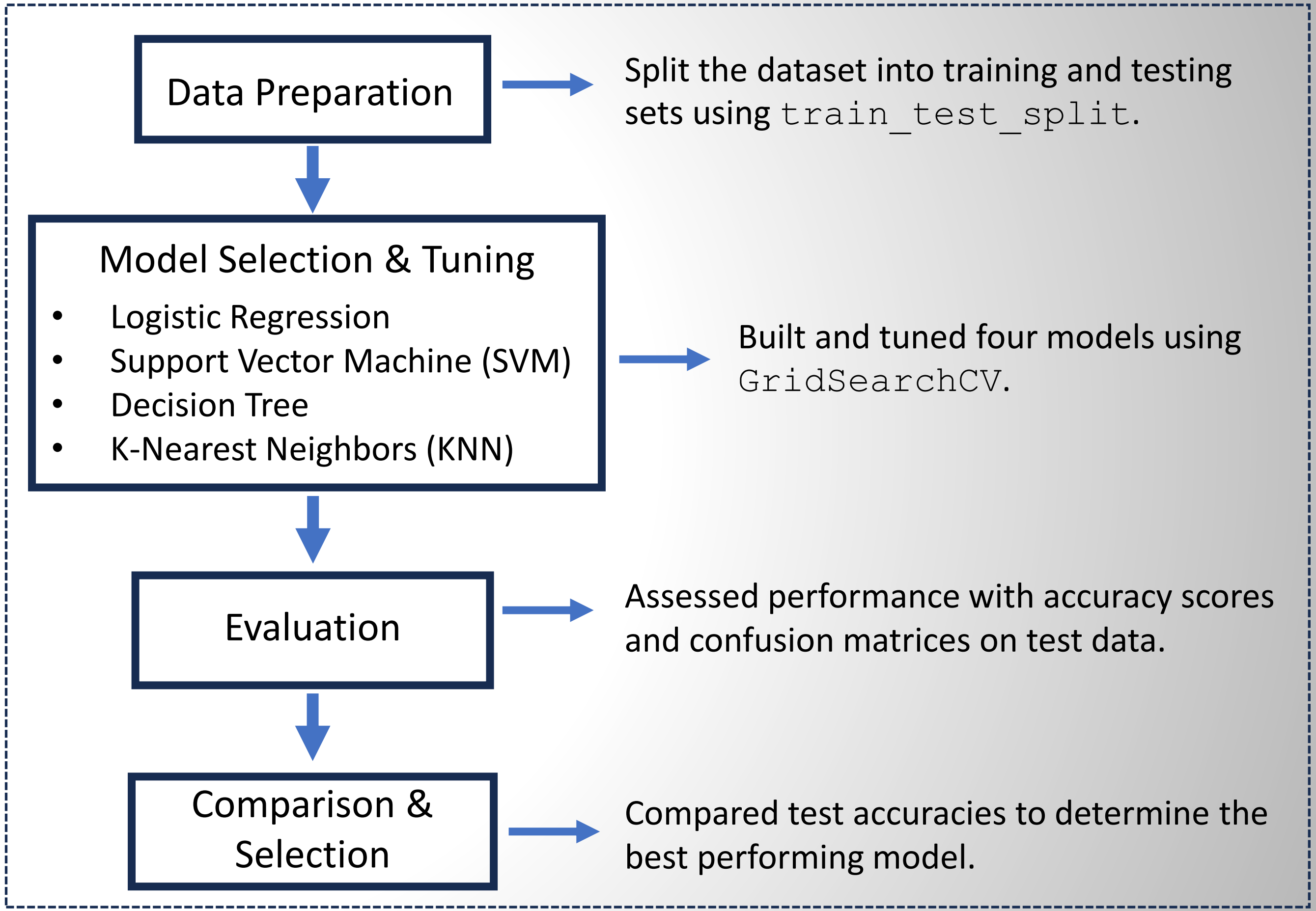
- Filters the scatter plot to show only launches within a selected payload range

➢ The completed Dashboard with Plotly Dash notebook is available on GitHub for external reference and peer-review purposes.

https://github.com/MarcosDS12/Capstone/blob/6d07b02acf7ecfa9a744f46f69966457aedea444/Ploty%20Dashboard%20Space%20X%20app.py

Predictive Analysis (Classification)

- We applied a systematic machine learning workflow to predict launch success. Starting with data preparation, we split the dataset into training and testing sets. Then, we trained and tuned four classification models — Logistic Regression, SVM, Decision Tree, and KNN — using GridSearchCV to find the best hyperparameters. Each model was evaluated using test accuracy and confusion matrices. Finally, we compared their performance and selected the model with the highest test accuracy as the best performer.



- The completed Predictive Analysis notebook is available on GitHub for external reference and peer-review purposes.
  https://github.com/MarcosDS12/Capstone/blob/a40c87c1154177ed248c1d9c56268096051209e9/Machine%20Learning%20Prediction%20Space%20X.ipynb

Results

Exploratory Data Analysis (EDA)

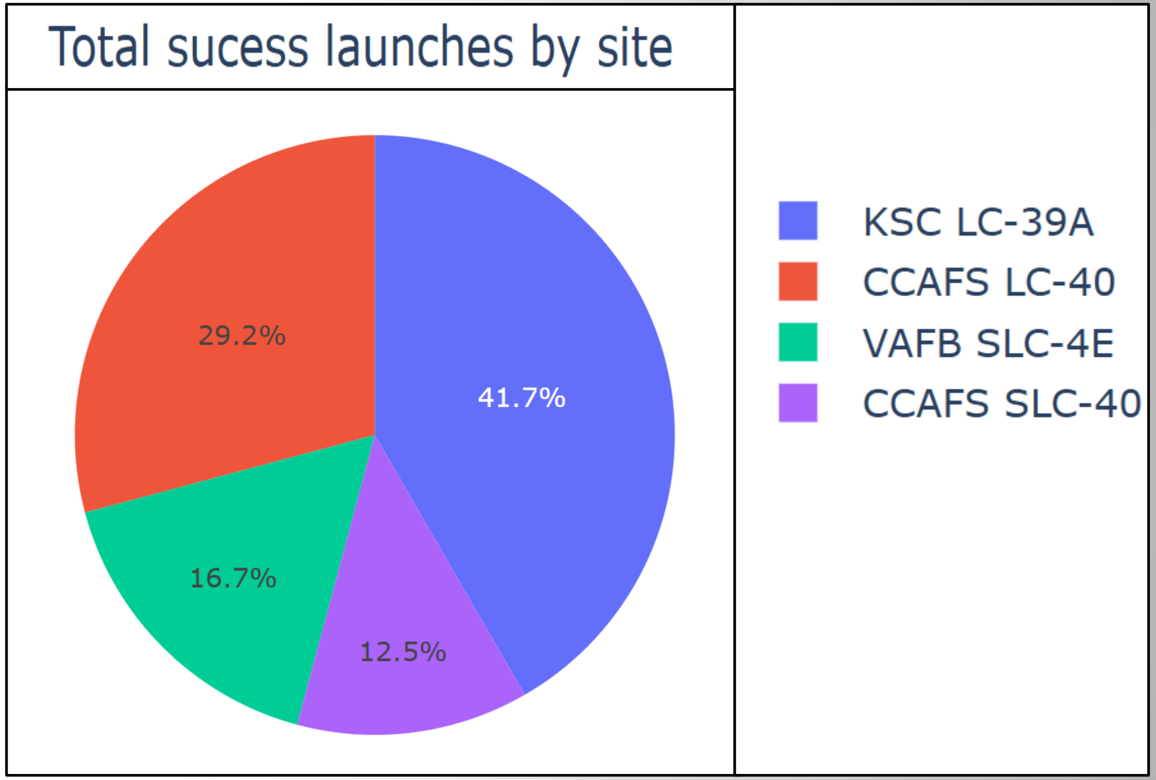
- Launch success rates varied by site and orbit
- Higher payload mass slightly decreased success probability
- Most launches were concentrated in CCAFS LC-40 and KSC LC-39A
- GTO orbits had lower success rates compared to LEO

Results

Interactive Analytics (Dash)

Success Distribution by Site (Pie Chart)

- The pie chart shows the distribution of successful launches across sites.
- KSC LC-39A accounts for the highest share of successful missions (~42%).
- Sites like VAFB SLC-4E and CCAFS LC-40 contribute a smaller share.

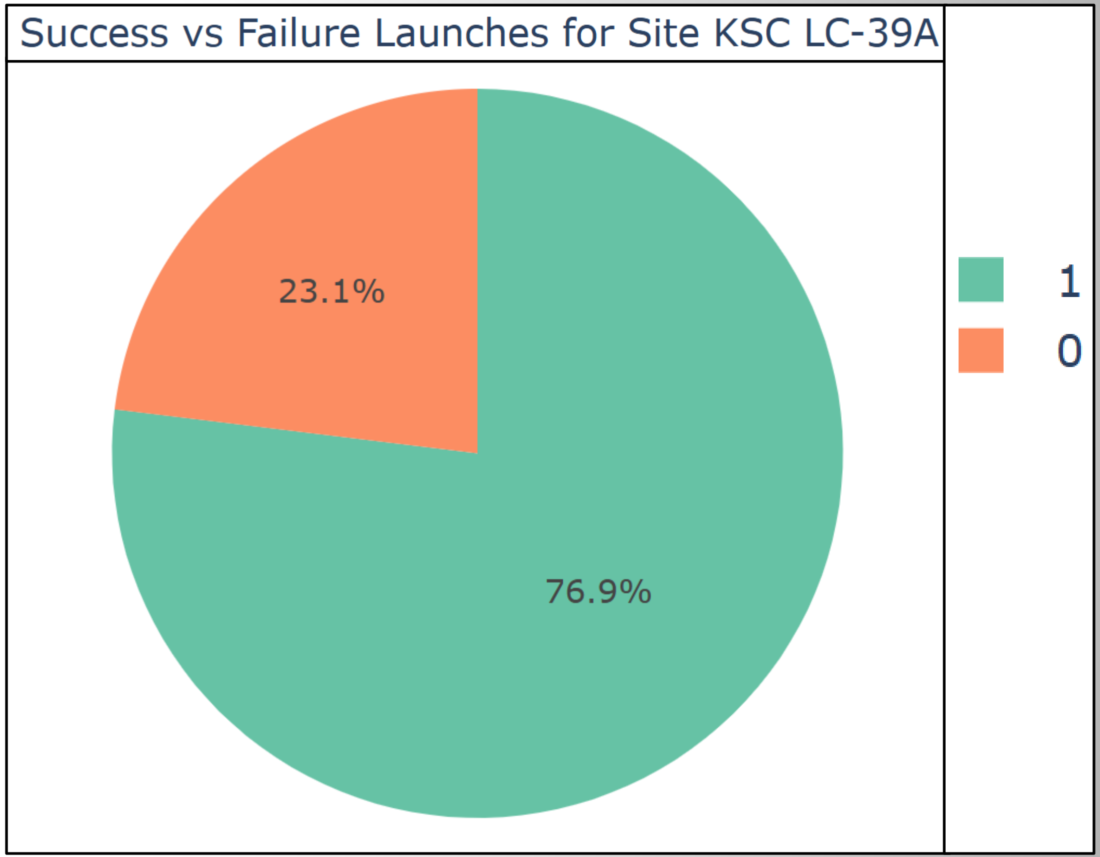


Results

Interactive Analytics (Dash)

Success Distribution by Site (Pie Chart)

- When filtering specifically for KSC LC-39A, the chart shows a 76.9% launch success rate.
- This demonstrates the effectiveness of filtering per site to assess performance individually.

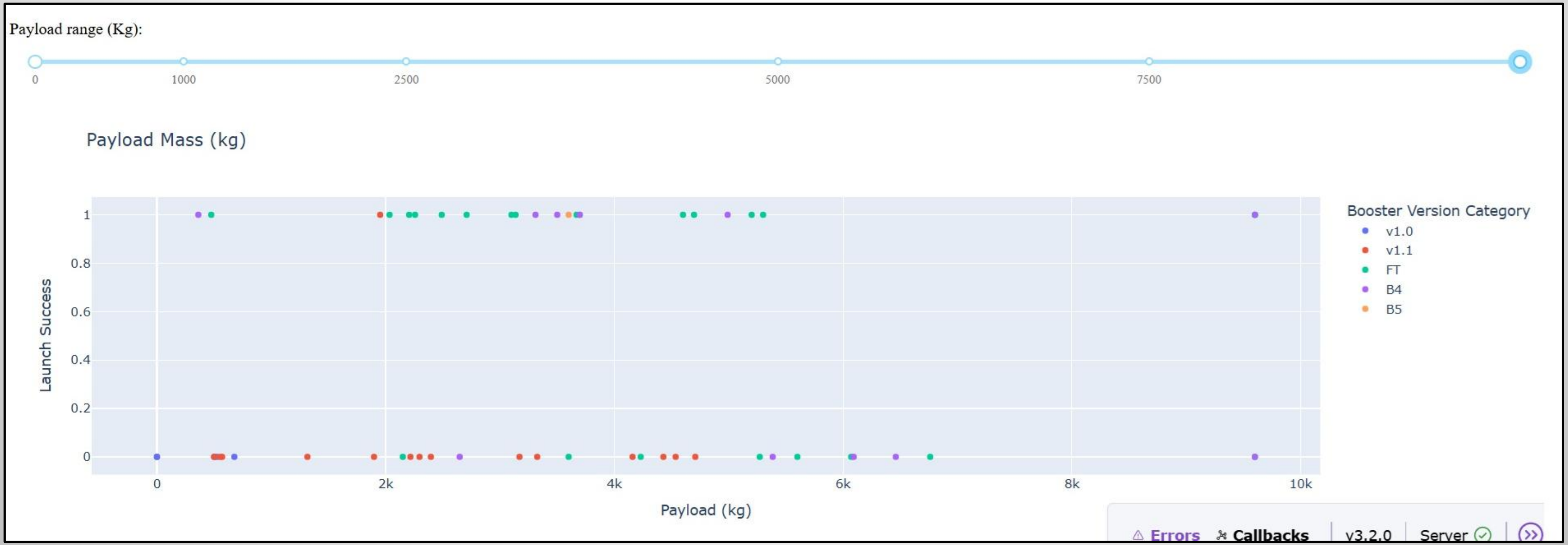


Results

Interactive Analytics (Dash)

Payload Mass vs. Launch Success (Scatter Plot)

- Launch success is not strongly correlated with payload mass.
- Most launches under 6000 kg had higher success frequencies.
- All booster versions (v1.0 to B5) are represented, with no clear booster dominating success.



Results

Predictive Model Results

We evaluated four classification models using test set accuracy:

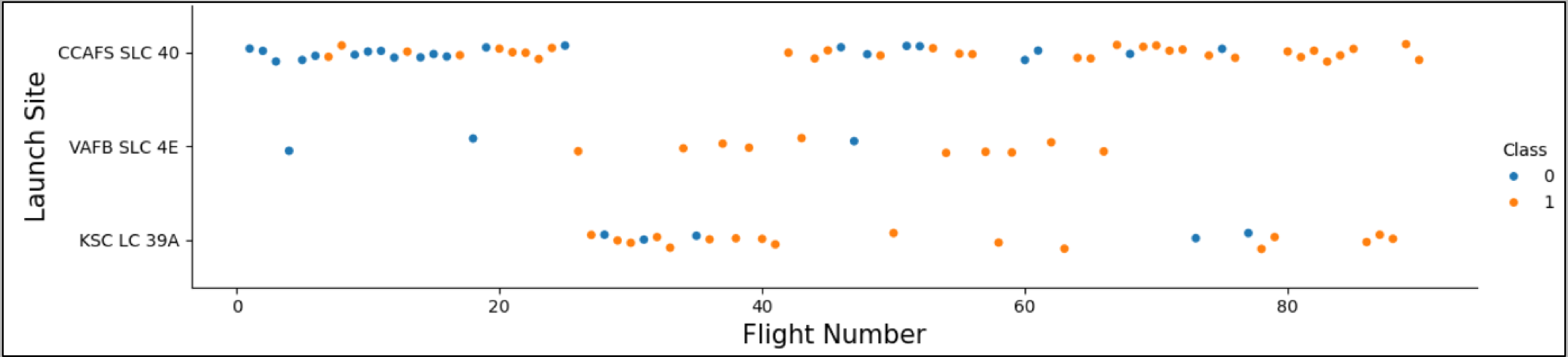
- **Logistic Regression**: 0.833
- **Support Vector Machine (SVM)**: 0.833
- **Decision Tree**: *0.667*
- **K-Nearest Neighbors (KNN)**: *0.778*

- Both Logistic Regression and SVM achieved the same highest accuracy (83.3%), resulting in a tie for best performance.
- Final model selection may depend on model interpretability (favoring Logistic Regression) or generalization capability (favoring SVM). Prediction model shows strong performance in classifying launch success.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

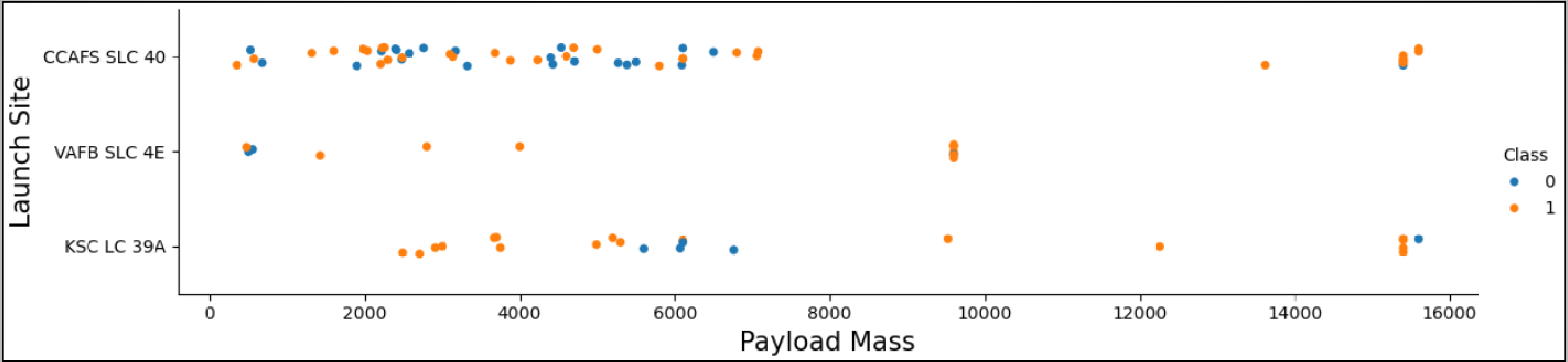


Graph 1: Scatter plot of Flight Number vs. Launch Site.

❖ Insights:

- At CCAFS SLC 40, higher Flight Numbers are associated with a greater share of successful landings.
- At VAFB SLC 4E, the sample size is small, so we cannot draw firm conclusions.
- At KSC LC-39A, the pattern appears similar to CCAFS SLC 40.
- Overall, across sites, the likelihood of a successful landing tends to increase with Flight Number, suggesting improvement over time.

Payload vs. Launch Site

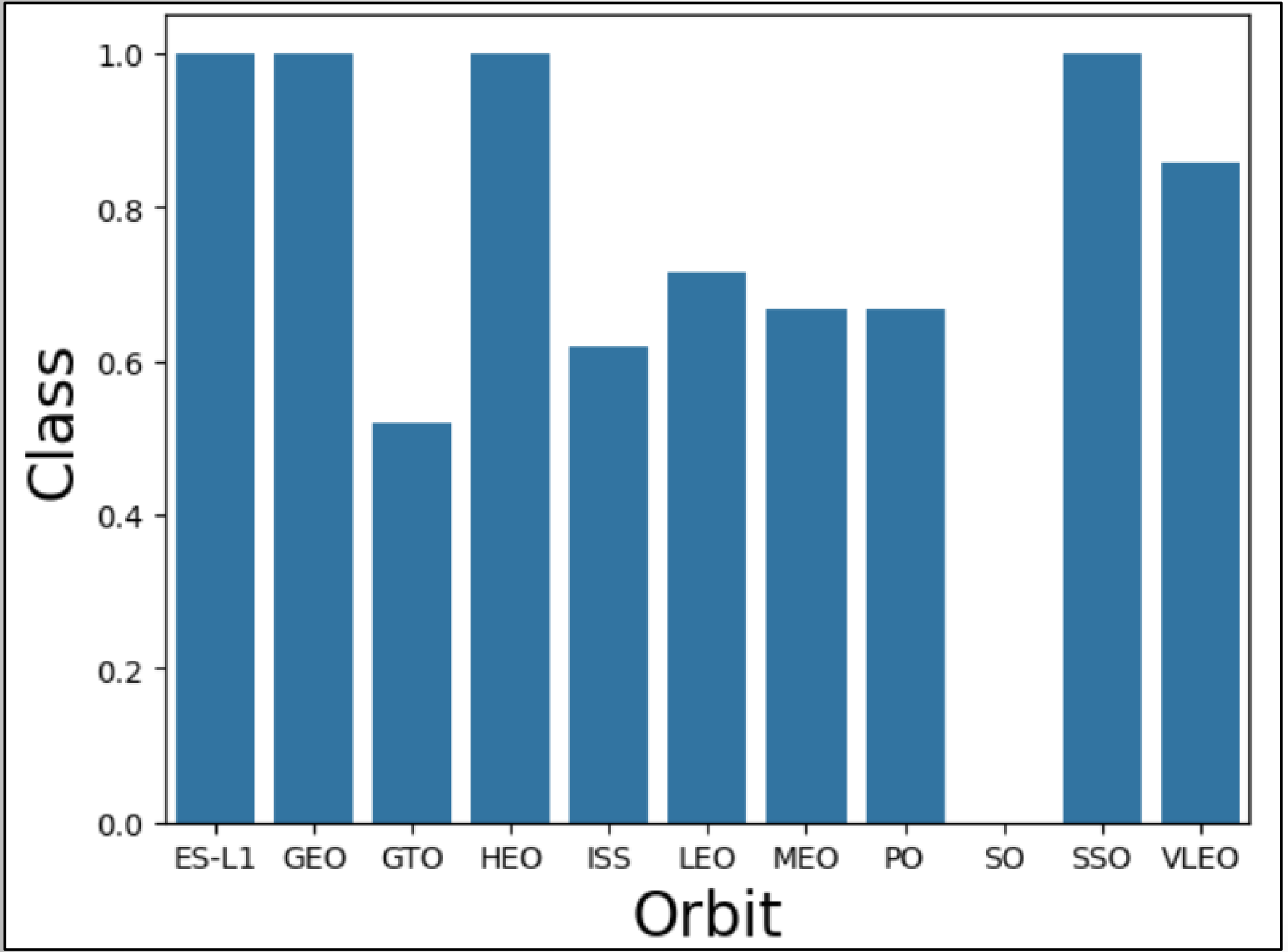


Graph 2: Scatter plot of Payload vs. Launch Site.

❖ Insights:

- At CCAFS SLC 40, successful and unsuccessful landings occur across a broad range of payload masses, with no clear decline in success as mass increases.
- At VAFB SLC-4E shows no launches above 10,000 kg payload mass; however, the sample size for this site is small.
- At KSC LC-39A, the heaviest missions (≈12–16 t) are mostly successful, with a few exceptions, suggesting heavy-payload capability without a systematic drop in landing success.

Success Rate vs. Orbit Type

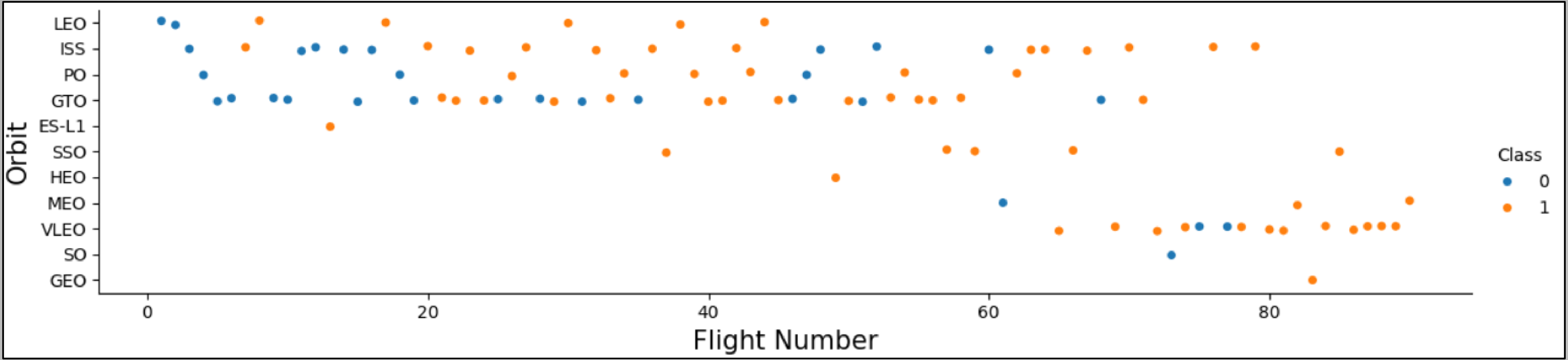


Graph 3: Bar plot of Success Rate vs. Orbit Type.

❖ Insights:

- Landing success varies by orbit: it's highest for ES-L1/GEO/HEO/SSO, moderate for LEO/MEO/PO/ISS, lowest for GTO, and zero for SO in this dataset.

Flight Number vs. Orbit Type

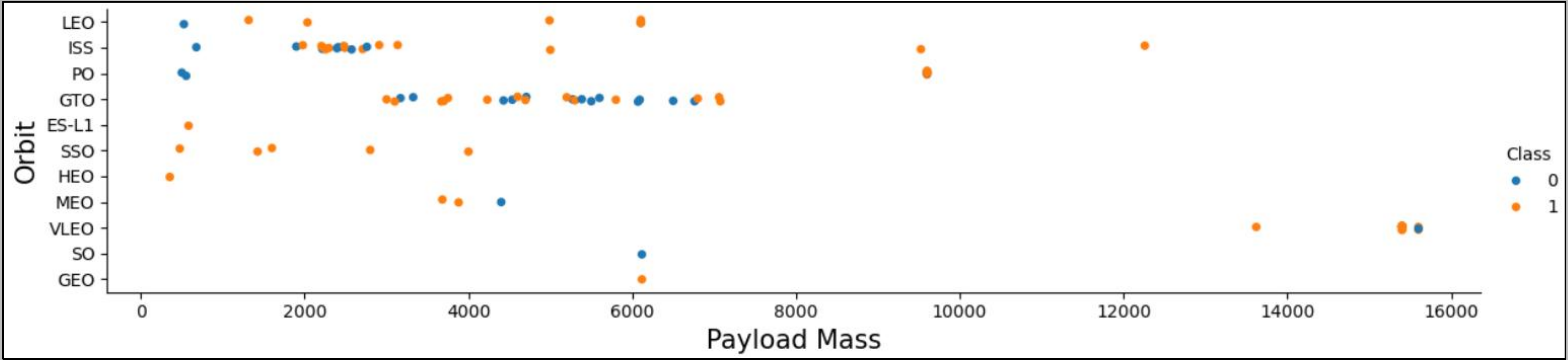


Graph 4: Scatter plot of Flight Number vs. Orbit Type.

❖ Insights:

- Across orbits, landing success increases with Flight Number, suggesting operational learning over time; later VLEO/SSO missions show consistently high success, while early LEO/GTO/ISS flights had more failures.

Payload vs. Orbit Type

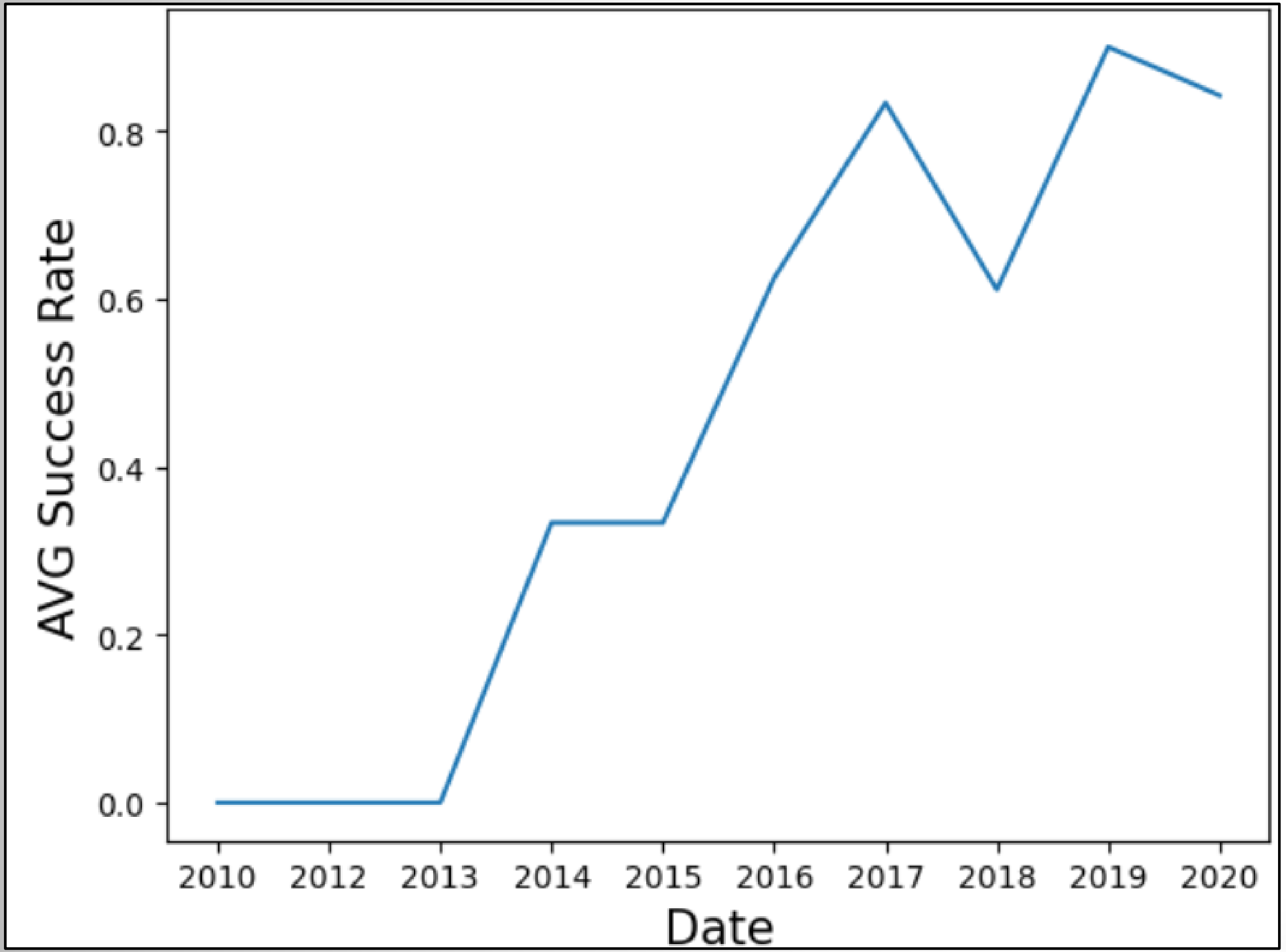


Graph 5: Scatter plot of Payload vs. Orbit Type.

❖ Insights:

- For heavy payloads, missions to Polar, LEO, and ISS orbits show higher landing success rates. In contrast, GTO exhibits a mixed record, making it hard to separate successes from failures.

Launch Success Yearly Trend



Graph 6: Line plot of Launch Success Yearly Trend.

❖ <u>Insights:</u>

- Sucess rate since 2013 kept increasing till 2020.

All Launch Site Names

❖ Query:

```
%%sql
SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

- The query lists all unique launch sites recorded in the SpaceX dataset, using the DISTINCT keyword to avoid duplicates.

Launch Site Names Begin with 'CCA'

❖ Query:

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

- The query retrieves the first five launches from sites whose names start with “CCA”, such as CCAFS LC-40. The LIKE operator is used for partial text matching.

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

- <u>Query:</u>

```
%%sql
SELECT SUM (PAYLOAD_MASS__KG_) FROM SPACEXTABLE
WHERE Customer LIKE 'NASA (CRS)%';
```

SUM (PAYLOAD_MASS__KG_)
48213

- The query sums the total payload mass for NASA (CRS) missions, returning a combined value of 48,213 kg.

Average Payload Mass by F9 v1.1

❖ Query:

```
%%sql
SELECT AVG (PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE '%F9 v1.1%';
```

AVG (PAYLOAD_MASS__KG_)
2534.6666666666665

- The query finds the average payload mass for launches using the Falcon 9 v1.1 booster version.

First Successful Ground Landing Date

- <u>Query:</u>

```
%%sql
SELECT MIN (Date) FROM SPACEXTABLE
WHERE Landing_Outcome == 'Success (ground pad)';
```

MIN (Date)
2015-12-22

- The query identifies the first successful landing on a ground pad using the MIN() function.

Successful Drone Ship Landing with Payload between 4000 and 6000

❖ Query:

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
WHERE Landing_Outcome == 'Success (drone ship)’
AND PAYLOAD_MASS__KG_>4000
AND PAYLOAD_MASS__KG_<6000;
```

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

- The query lists boosters that landed successfully on a drone ship with payloads between 4,000 kg and 6,000 kg.

Total Number of Successful and Failure Mission Outcomes

❖ Query:

```
%%sql
SELECT
   CASE
      WHEN Landing_Outcome LIKE '%Success%' THEN 'Success'
      WHEN Landing_Outcome LIKE '%Failure%' THEN 'Failure'
      ELSE 'Other'
   END AS Outcome_Type,
   COUNT(*) AS Total
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%Success%' OR Landing_Outcome LIKE '%Failure%'
GROUP BY Outcome_Type;
```

Outcome_Type	Total
Failure	10
Success	61

- The query classifies landings as “Success” or “Failure” using a CASE statement and counts how many missions belong to each group.

Boosters Carried Maximum Payload

❖ Query:

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

- The query uses a subquery to find the booster version that carried the heaviest payload in the dataset.

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

❖ Query:

```
%%sql
SELECT
   CASE substr(Date, 6, 2)
      WHEN '01' THEN 'January’
      WHEN '02' THEN 'February’
      WHEN '03' THEN 'March’
      WHEN '04' THEN 'April’
      WHEN '05' THEN 'May’
      WHEN '06' THEN 'June’
      WHEN '07' THEN 'July’
      WHEN '08' THEN 'August’
      WHEN '09' THEN 'September’
      WHEN '10' THEN 'October’
      WHEN '11' THEN 'November’
      WHEN '12' THEN 'December’
  END AS Month_Name,  Landing_Outcome,  Booster_Version,  Launch_Site
FROM SPACEXTABLE
WHERE substr(Date, 1, 4) = '2015'                    -- año 2015
  AND Landing_Outcome = 'Failure (drone ship)’
ORDER BY substr(Date, 6, 2), Date;
```

- The query uses substr() and a CASE statement to display 2015 drone ship landing failures by month, along with booster version and launch site, sorted chronologically.

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

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

❖ Query:

```
%%sql
SELECT
   CASE
      WHEN Landing_Outcome LIKE '%Failure (drone ship)%' THEN 'Failure (drone ship)’
      WHEN Landing_Outcome LIKE '%Success (ground pad)%' THEN 'Success (ground pad)’
   END AS Outcome_Type,
   COUNT(*) AS Total
FROM SPACEXTABLE
WHERE
       Landing_Outcome LIKE '%Failure (drone ship)%’
       OR Landing_Outcome LIKE '%Success (ground pad)%’
       AND Date BETWEEN '2010-06-04' and '2017-03-20’
GROUP BY Outcome_Type;
```

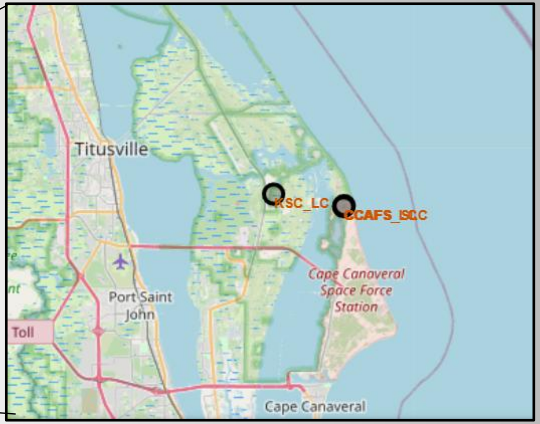
Outcome_Type	Total
Failure (drone ship)	5
Success (ground pad)	3

- The query counts drone-ship failures and ground-pad successes between 2010-06-04 and 2017-03-20, using a CASE statement for classification and grouping by outcome type.

Section 3

Launch Sites Proximities Analysis

Geographical location of the launch sites

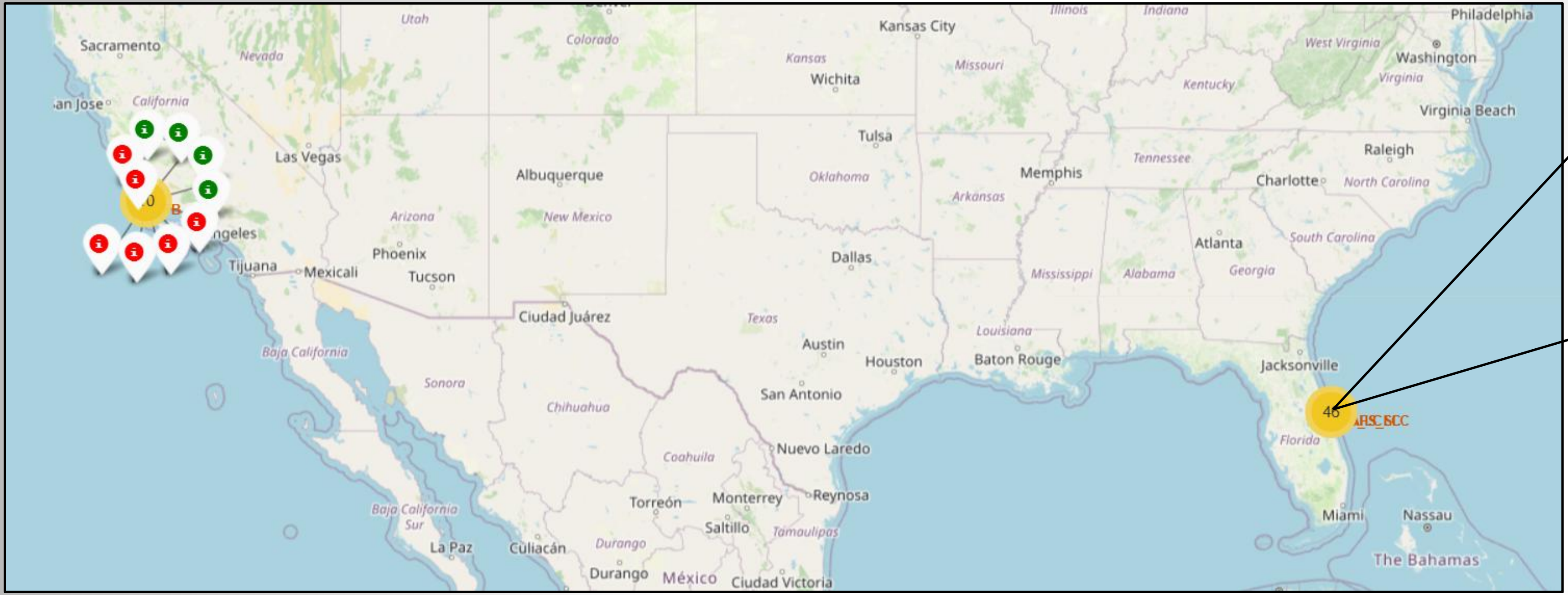


Map 1: Geographical location of the launch sites

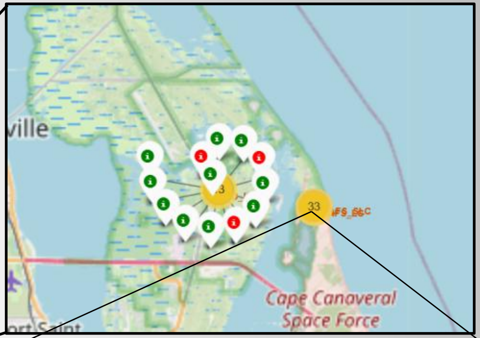
❖ <u>Insights:</u>

- US launch sites are **coastal** (Atlantic/Pacific) for range safety and mission profiles, but they are **not near the Equator** (≈28–35°N), so they don't benefit from the maximum equatorial rotational boost.

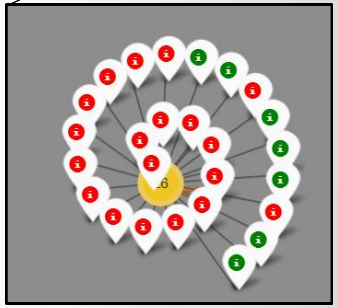
Launch Outcomes by Location



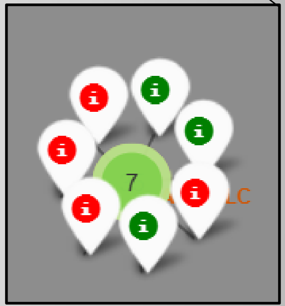
Map 2: Launch Outcomes by Location.



Map 2.1: KSC LC-39A.



Map 2.2: CCASF LC-40.

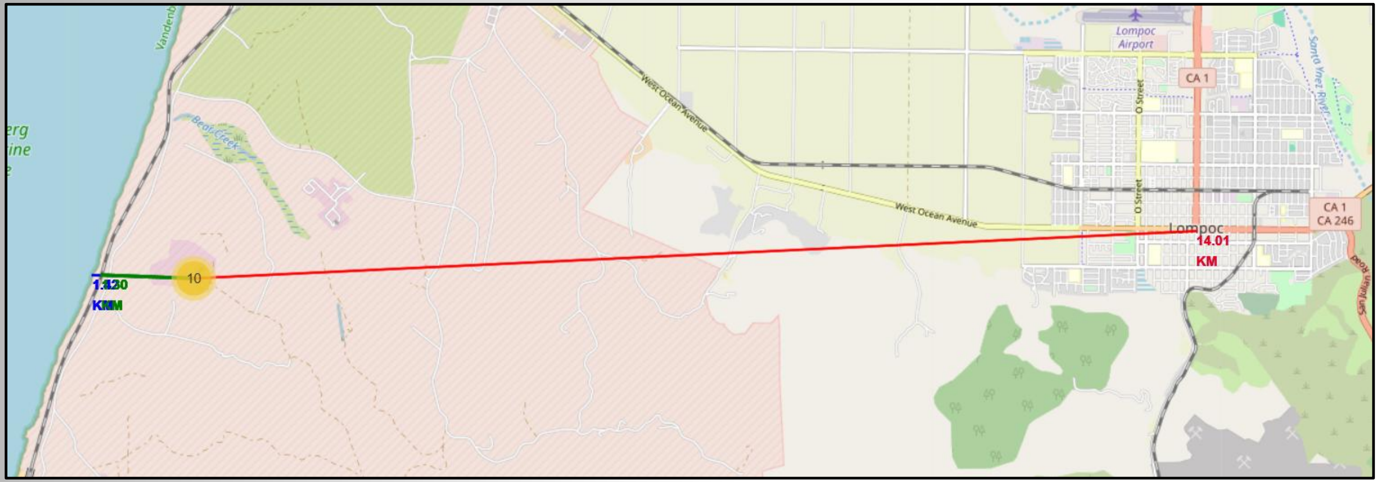


Map 2.3: CCASF SLC-40.

❖ Insights:

- At the Florida sites, a clear contrast can be observed: KSC LC-39A displays mostly green markers, indicating a high landing success rate, whereas CCAFS SLC-40 and CCASF LC-40 shows more red markers, suggesting a higher number of failed landings.
- At Vandenberg (SLC-4E), launches are less frequent and the distribution of green and red markers is more mixed, suggesting a lower or more variable success rate.

Geographical Position of Vandenberg Launch Site (SLC-4E)



❖ Insights:

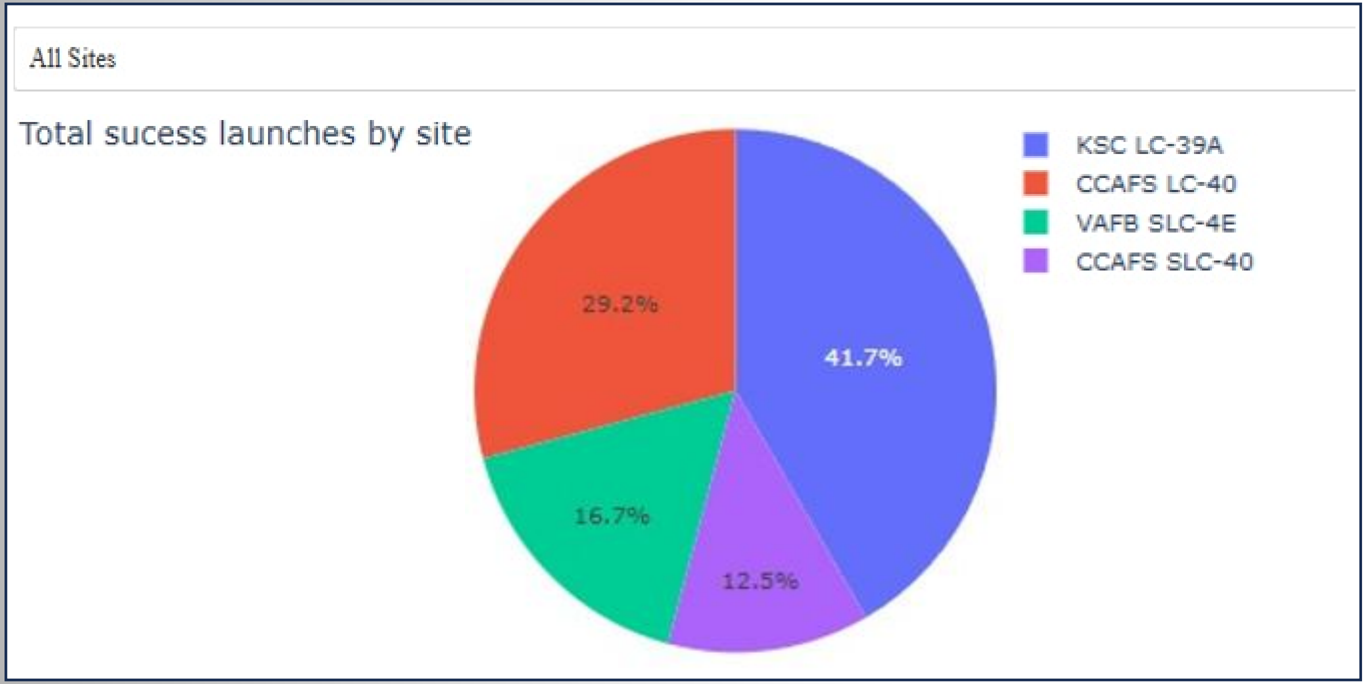
- The site lies approximately 1.42 km from the Pacific coastline, 1.30 km from the UP Santa Barbara Subdivision railway line, and about 14 km from the city of Lompoc — which is also roughly the same distance to the nearest highway (CA-1).

Map 3: Geographical Position of Vandenberg Launch Site (SLC-4E).

Section 4

Build a Dashboard with Plotly Dash

Distribution of successful launches by site

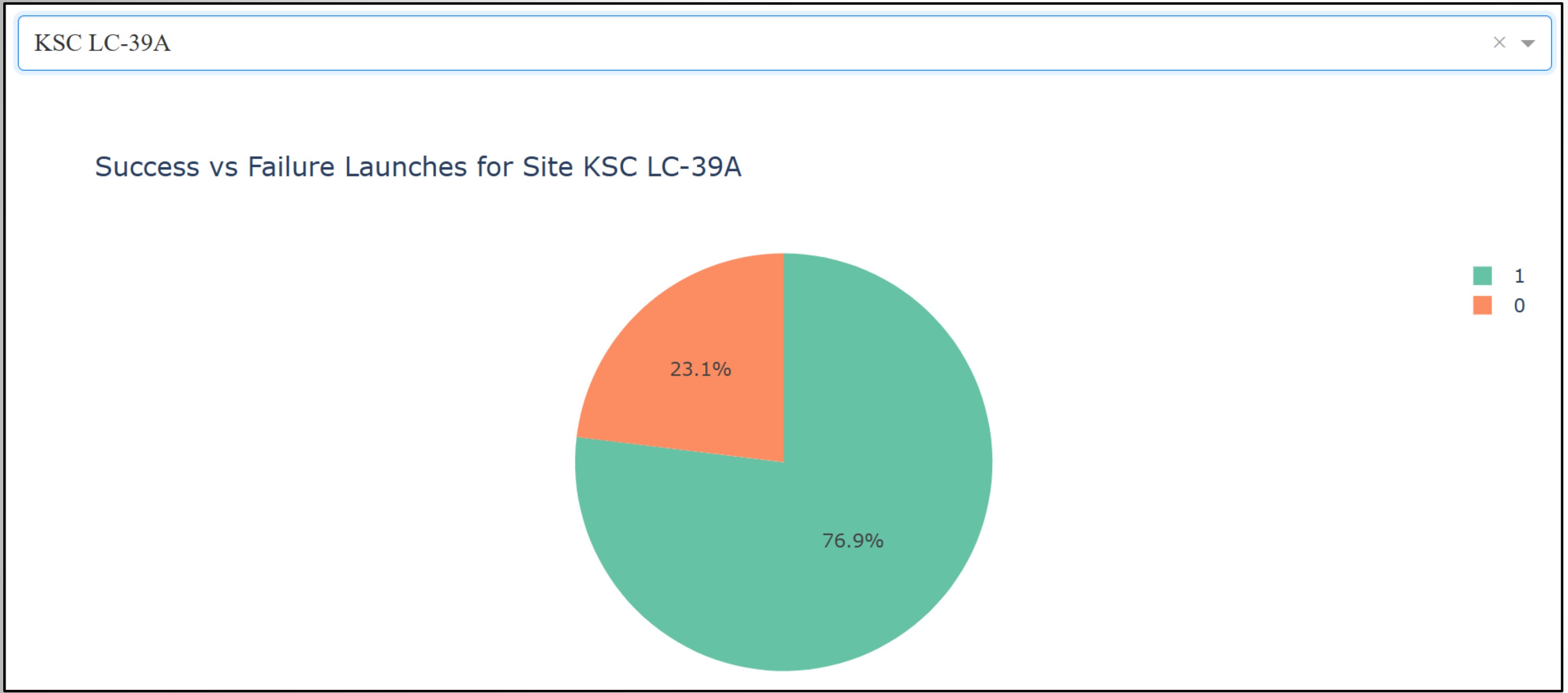


Graph 7: Pie chart distribution of successful launches by site.

❖ Insights:

- The **KSC LC-39A** site accounts for the **largest share of successful launches (41.7%)**, followed by **CCAFS LC-40** with **29.2%**, **VAFB SLC-4E** with **16.7%**, and **CCAFS SLC-40** with **12.5%**.
- This suggests that **KSC LC-39A** is SpaceX's **most reliable and frequently used site**, while **VAFB SLC-4E** and **CCAFS SLC-40** have played smaller but consistent roles in the company's successful missions.

Success vs. Failure Launches – KSC LC-39A (Highest Success Rate)



Graph 7: Pie chart distribution of Success vs. Failure Launches – KSC LC-39A .

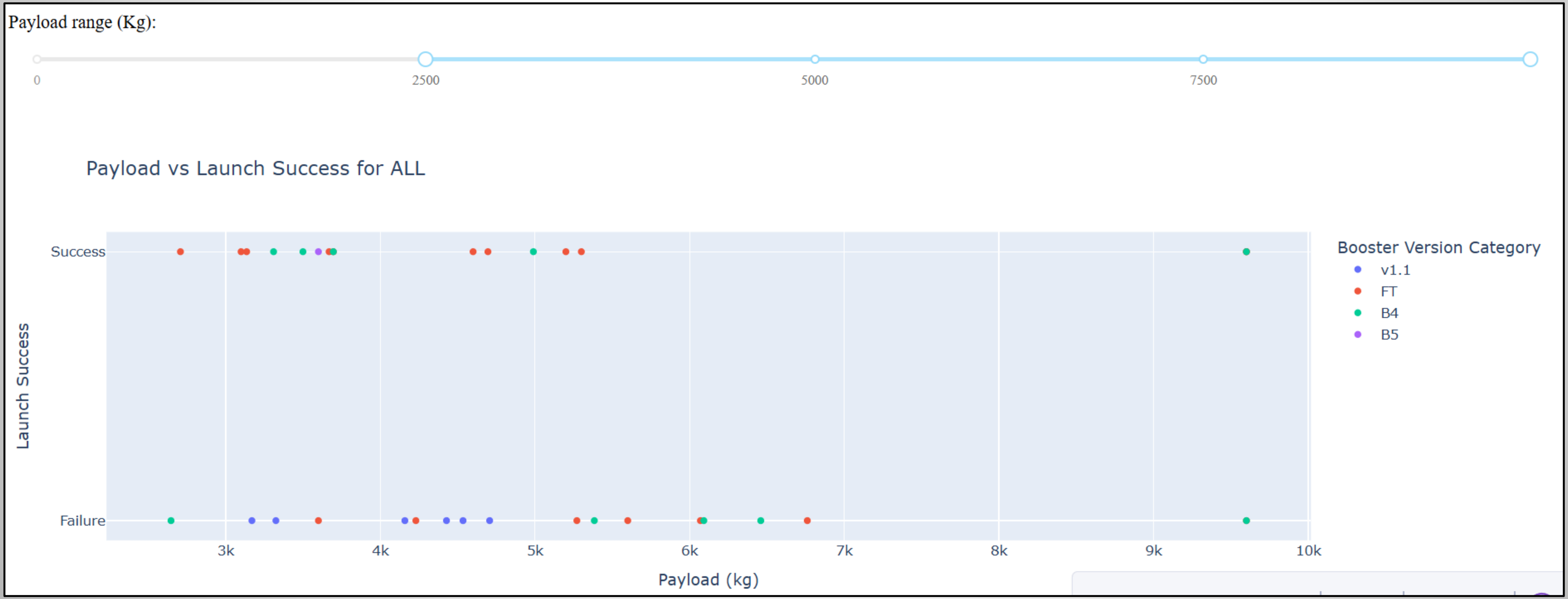
❖ Insights:

- This chart shows the performance of the **Kennedy Space Center Launch Complex 39A (KSC LC-39A)**, which holds the **highest success rate** among all SpaceX launch sites.

- Approximately 76.9% of its launches have been successful, while 23.1% resulted in failures.
- This makes KSC LC-39A the most reliable and consistent launch site in SpaceX's operations, reflecting its central role in the company's most successful missions

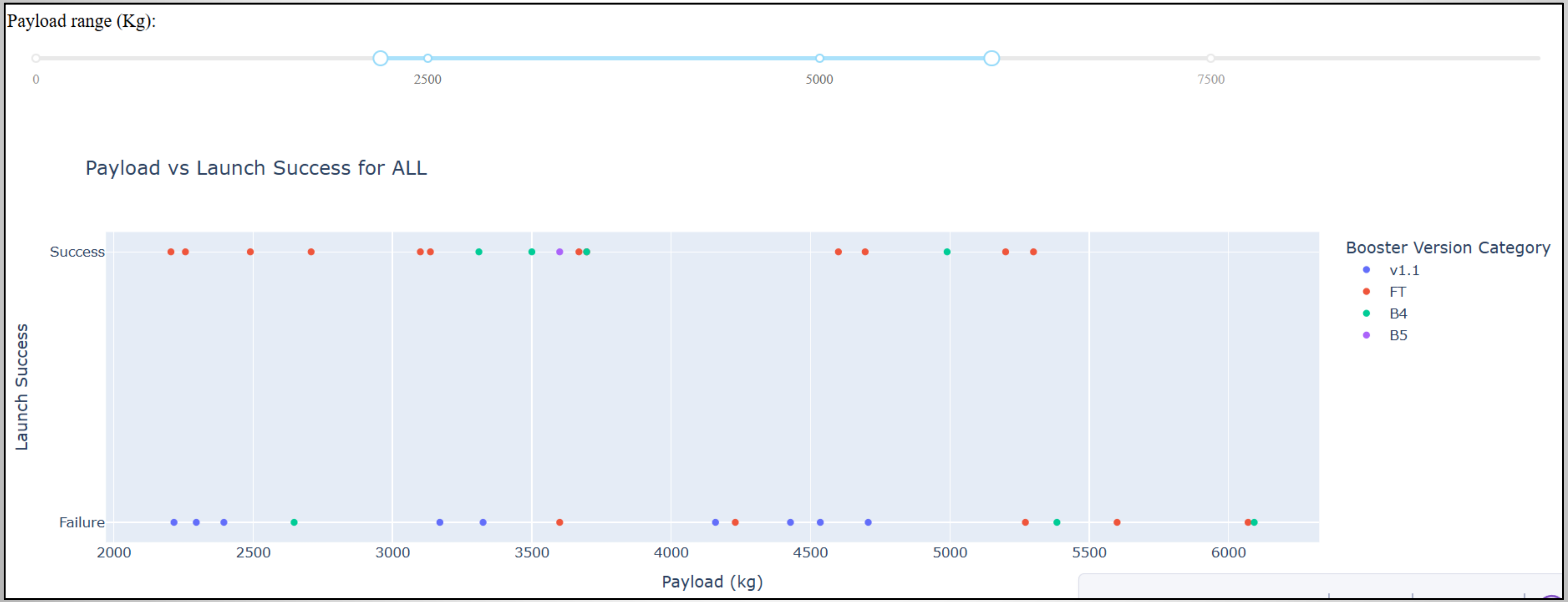
Payload Mass vs. Launch Outcome (All Launch Sites)

- The scatter plot shows the correlation between **payload mass** and **launch success** across all SpaceX launch sites. In the following image, the **launch outcomes are displayed for payloads between 2500 kg and 10000 kg**, highlighting how success varies among different launch sites within this payload range.



Payload Mass vs. Launch Outcome (All Launch Sites)

- Overall, most successful launches occur within this intermediate payload band, where **FT and B5 booster versions** maintain higher success consistency compared to earlier versions such as **v1.0** and **v1.1**..

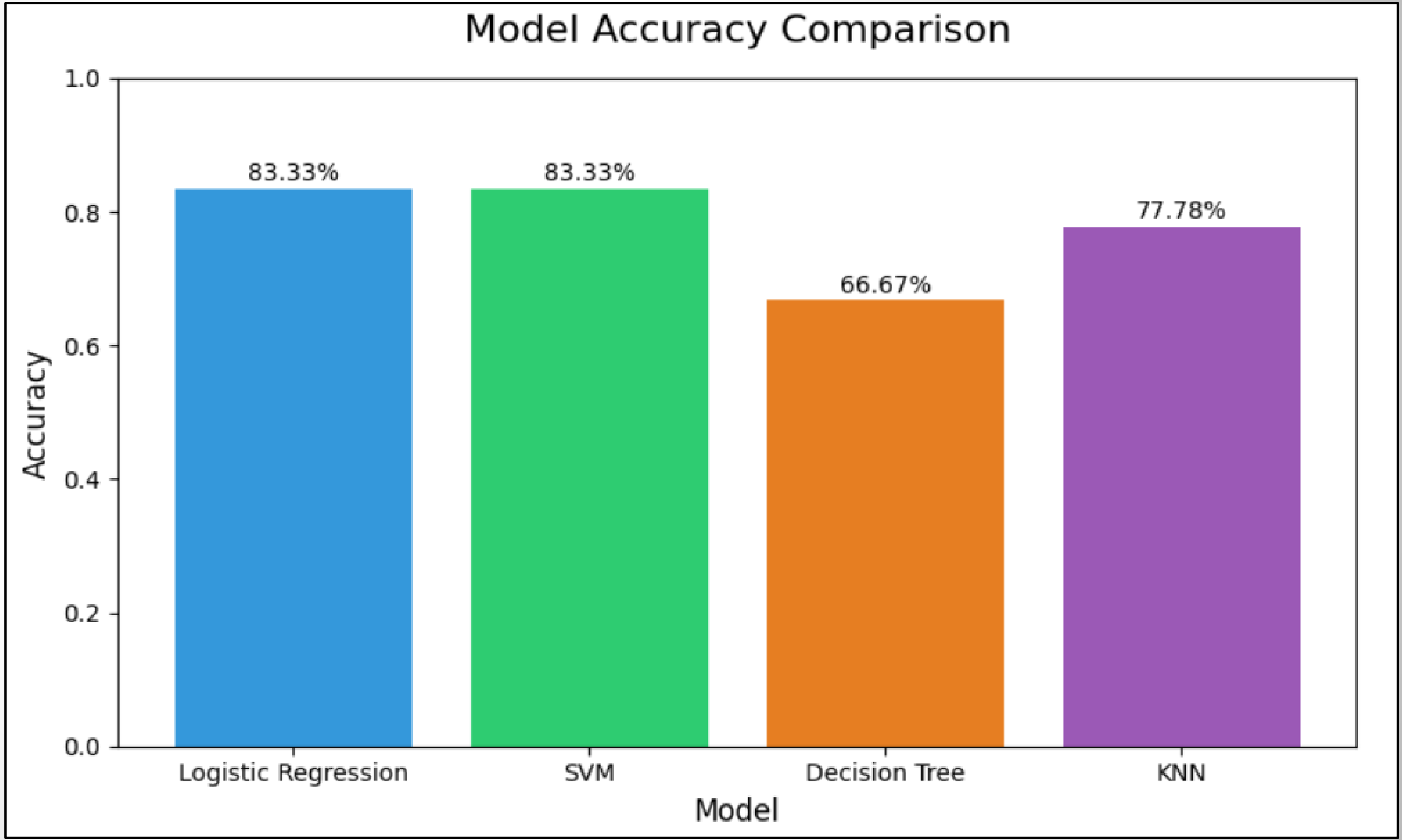


Section 5

Predictive Analysis (Classification)

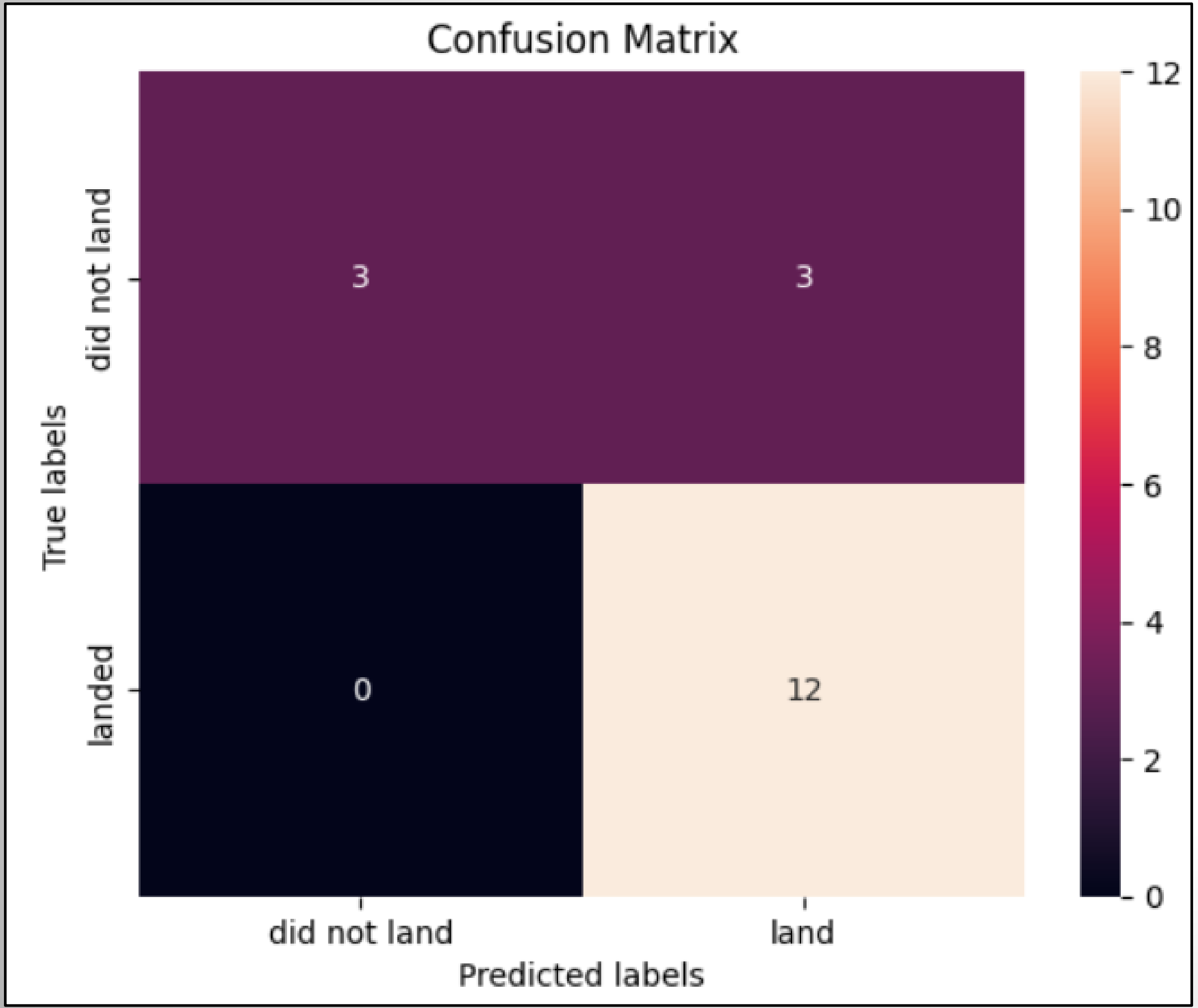
Classification Accuracy

- Both **Logistic Regression** and **SVM** models achieved the same accuracy, which indicates that they likely learned very similar decision boundaries on this dataset.



Confusion Matrix

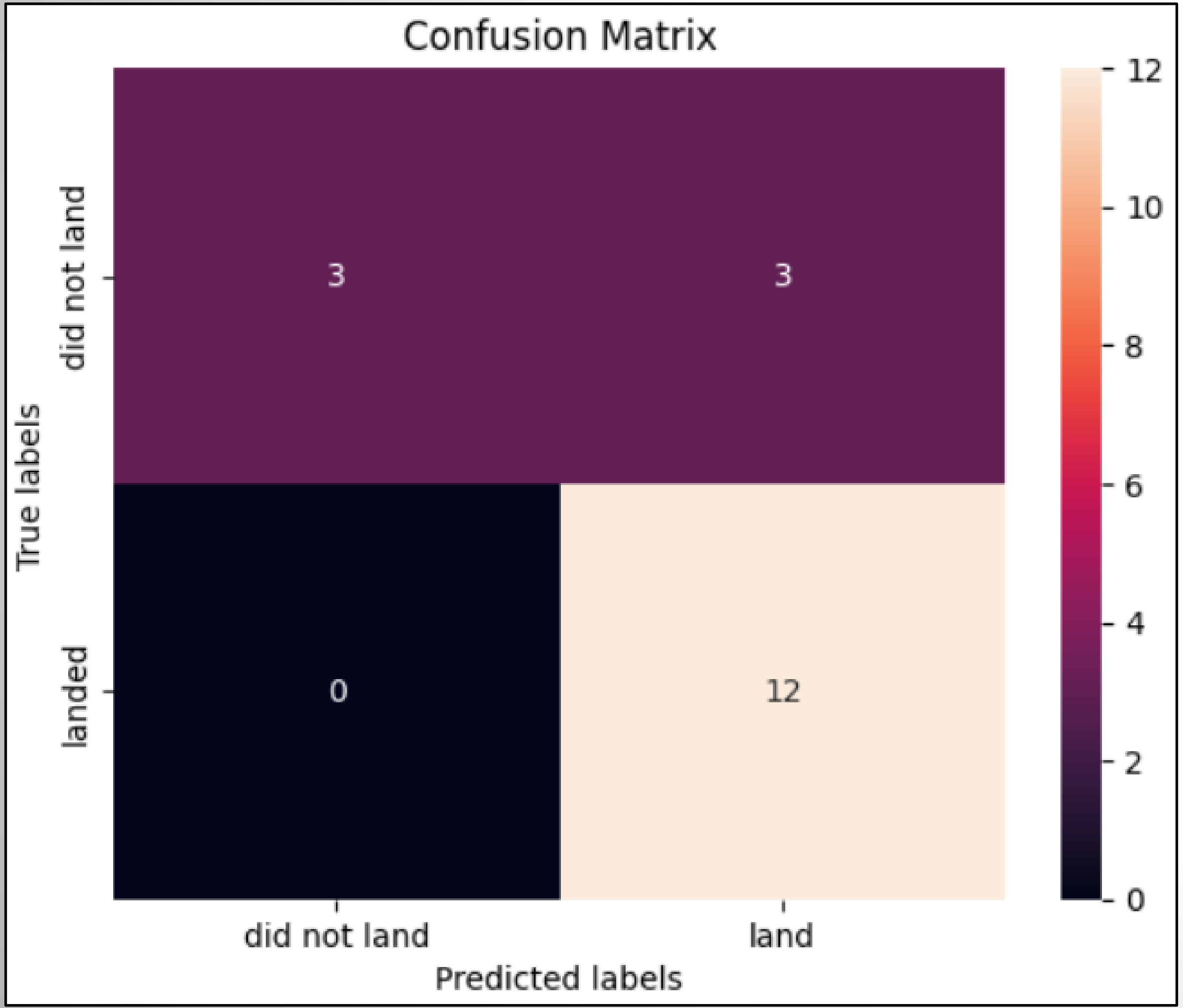
❖ <u>Logistic Regression</u>



- The confusion matrix shows that the Logistic Regression model correctly predicted 12 successful landings and 3 failed landings. However, it also misclassified 3 failed landings as successful. Overall, this indicates that the model performs well in identifying successful launches but is slightly less accurate when predicting failures.

Confusion Matrix

❖ SVM



- The confusion matrix for the SVM model shows that it correctly predicted 12 successful landings and 3 failed landings, while misclassifying 3 failed attempts as successful. This performance is identical to the Logistic Regression model, indicating that both classifiers behave similarly on this dataset.

Conclusions

1. ***Launch site matters:***

   KSC LC-39A consistently shows the highest launch success rate (~77%), confirming its reliability as SpaceX's top-performing site.

2. ***Payload mass has limited impact:***

   While most successes occur with payloads under 6000 kg, no strong correlation was found between payload mass and launch outcome.

3. ***Interactive analytics empower insight:***

   Tools like Folium and Dash enabled intuitive exploration of geospatial and mission data, revealing patterns that static analysis might miss.

4. ***Modeling can predict landing success:***

   Logistic Regression and SVM models both achieved 83.3% accuracy, demonstrating that mission success can be predicted from launch data with high reliability.

Appendix

- Machine Learning Prediction with GridSearchCV (Logistic Regression)

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr, parameters, cv=10)
logreg_cv.fit(X_train, Y_train)
```

```
▸ GridSearchCV
▸ estimator: LogisticRegression
  ▸ LogisticRegression
```

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.85
```

```
test_accuracy = logreg_cv.score(X_test, Y_test)

print("Test Accuracy :", test_accuracy)
```

```
Test Accuracy : 0.8333333333333334
```

Appendix

➢ Machine Learning Prediction with GridSearchCV (SVM)

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```

```
svm_cv = GridSearchCV(svm, parameters, cv=10)
svm_cv.fit(X_train, Y_train)
```

```
GridSearchCV
  estimator: SVC
    SVC
```

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8642857142857144
```

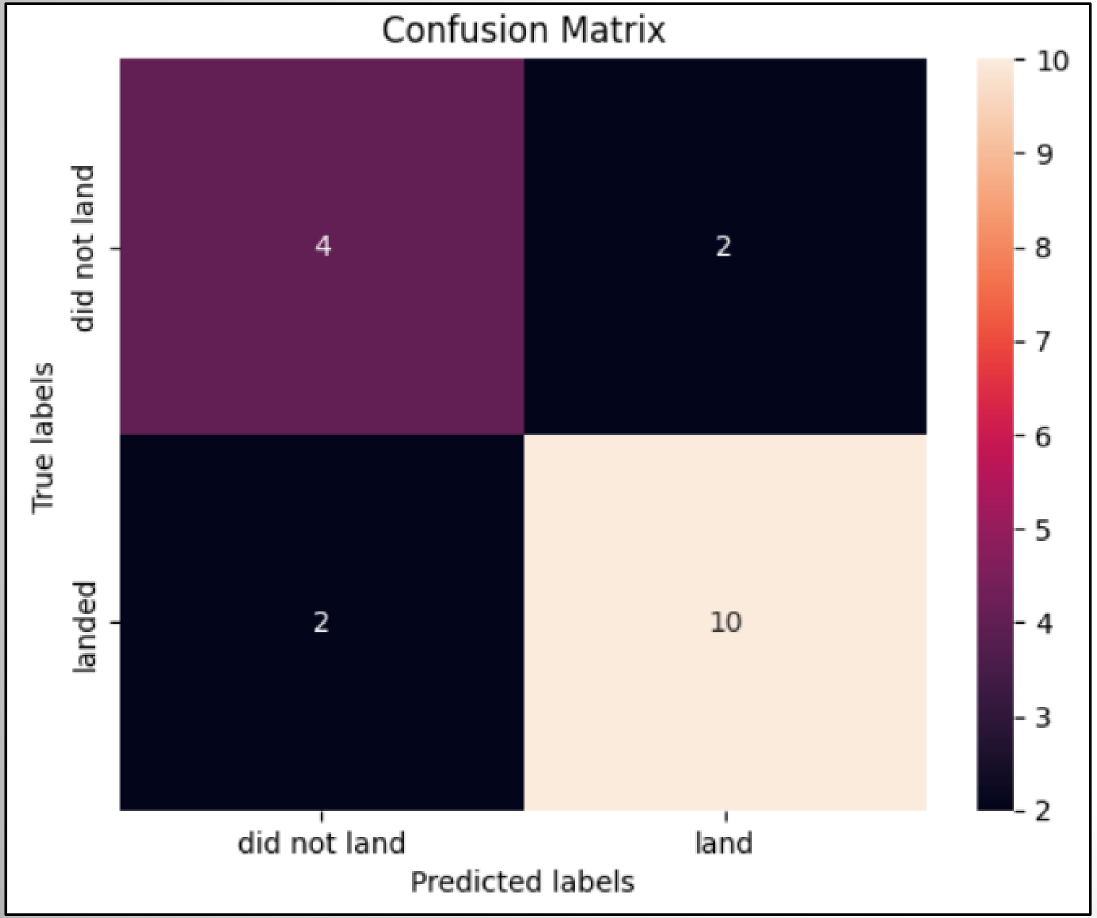
```
test_accuracy_svm = svm_cv.score(X_test, Y_test)

print("Test Accuracy :", test_accuracy_svm)
```

```
Test Accuracy : 0.8333333333333334
```

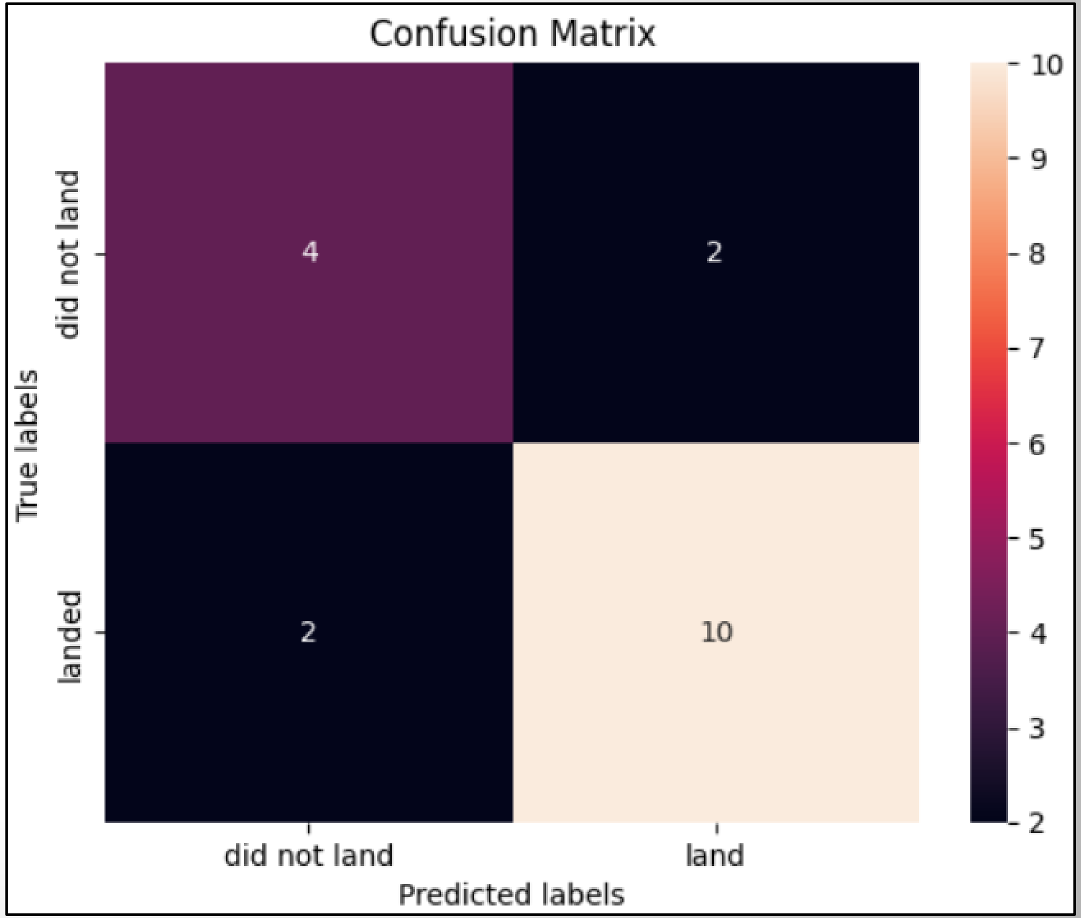
Appendix

➢ Machine Learning Prediction: Confusion matrix

DecisionTreeClassifier



KNeighborsClassifier



Appendix

- Links to GitHub Notebooks:

Data Collection - SpaceX API notebook:
https://github.com/MarcosDS12/Capstone/blob/350282d12cb0359e4be900cc3cf714319279976f/Data%20Collection%20SpaceX%20API.ipynb

Web Scraping notebook:
https://github.com/MarcosDS12/Capstone/blob/e125432b062f872c63993fd98662b7ead263f6ce/Data%20Collection%20Webscraping%20Space%20X.ipynb

Data Wrangling notebook:
https://github.com/MarcosDS12/Capstone/blob/f1bd3087d01447e6e5ad6d2d57bfe12d1aabe855/Data%20wrangling%20Space%20X.ipynb

EDA with Data Visualization notebook:
https://github.com/MarcosDS12/Capstone/blob/757751f84ba5690e5bf01c78b91c117430a7623b/EDA%20with%20Data%20Visualization%20Space%20X.ipynb

EDA with SQL notebook:
https://github.com/MarcosDS12/Capstone/blob/6b8604a49a5675e9c73f5083df34412de0fa8e96/EDA%20with%20SQL%20Space%20X.ipynb

Interactive Map with Folium notebook:
https://github.com/MarcosDS12/Capstone/blob/6b8604a49a5675e9c73f5083df34412de0fa8e96/Interactive%20Map%20with%20Folium%20Space%20X.ipynb

Dashboard with Plotly Dash notebook:
https://github.com/MarcosDS12/Capstone/blob/6d07b02acf7ecfa9a744f46f69966457aedea444/Ploty%20Dashboard%20Space%20X%20app.py

Predictive Analysis notebook:
https://github.com/MarcosDS12/Capstone/blob/a40c87c1154177ed248c1d9c56268096051209e9/Machine%20Learning%20Prediction%20Space%20X.ipynb

Thank you!