IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fernando Rioseco Rodriguez
Sep 20, 2025.

Outline

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

Executive Summary

Summary of Methodologies:

Data was collected from two main sources: the SpaceX API for past launch records filtered for Falcon 9 launches, and web scraping from Wikipedia to supplement historical data. Datasets were cleaned, merged, and structured. Data wrangling handled missing values, identified data types, and aggregated key features. EDA visualized relationships between payload mass, orbit type, and launch site using charts, SQL queries, and interactive Folium and Plotly Dash dashboards. Predictive analysis evaluated four models (Logistic Regression, SVM, Decision Tree, KNN), with the Decision Tree achieving the highest validation (88.9%) and ~83% test accuracy.

Summary of All Results:

EDA showed that success rates vary by orbit, with heavier payloads performing better in some cases. Florida launch sites are near the coast and infrastructure but far from cities for safety. Folium maps and Dash dashboards highlighted success patterns by site, payload, and booster. Predictive analysis confirmed the Decision Tree as the most reliable model, and key insights include improved SpaceX reliability over time and identification of booster and payload combinations with higher success rates.

Introduction

Project Background and Context

The commercial space era is rapidly growing, with companies like Virgin Galactic, Rocket Lab, and Blue Origin making space travel more accessible. SpaceX stands out for achievements like sending spacecraft to the ISS, launching Starlink satellites, and conducting crewed missions. Its cost advantage comes from reusing the Falcon 9 first stage, which greatly reduces launch expenses. Understanding first-stage success is crucial for estimating launch costs. This project simulates a data scientist at Space Y, a new company aiming to compete with SpaceX, analyzing public SpaceX data, building dashboards, and developing predictive models to determine whether a Falcon 9 first stage will be recovered.

Problems / Questions to Address

- Can we predict whether SpaceX will successfully recover the Falcon 9 first stage for a given launch?
- Which factors—such as payload mass, orbit, or booster version—affect first-stage recovery success?
- How can this information be used to estimate the cost of a launch for a new space company competing with SpaceX?
- What insights from past SpaceX launches can guide strategic decisions for future missions?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data for this project was collected from two main sources to ensure a comprehensive dataset of SpaceX launches:

Public API: past SpaceX launch records were obtained programmatically and transformed into a structured tabular format suitable for analysis. This included extracting key fields such as launch site, payload mass, booster version, orbit type, and launch outcome.

Web Scraping: additional launch information was extracted directly from Wikipedia pages to supplement the dataset with historical records that might not be available through the API. The web content was cleaned, normalized, and structured into a consistent tabular format.

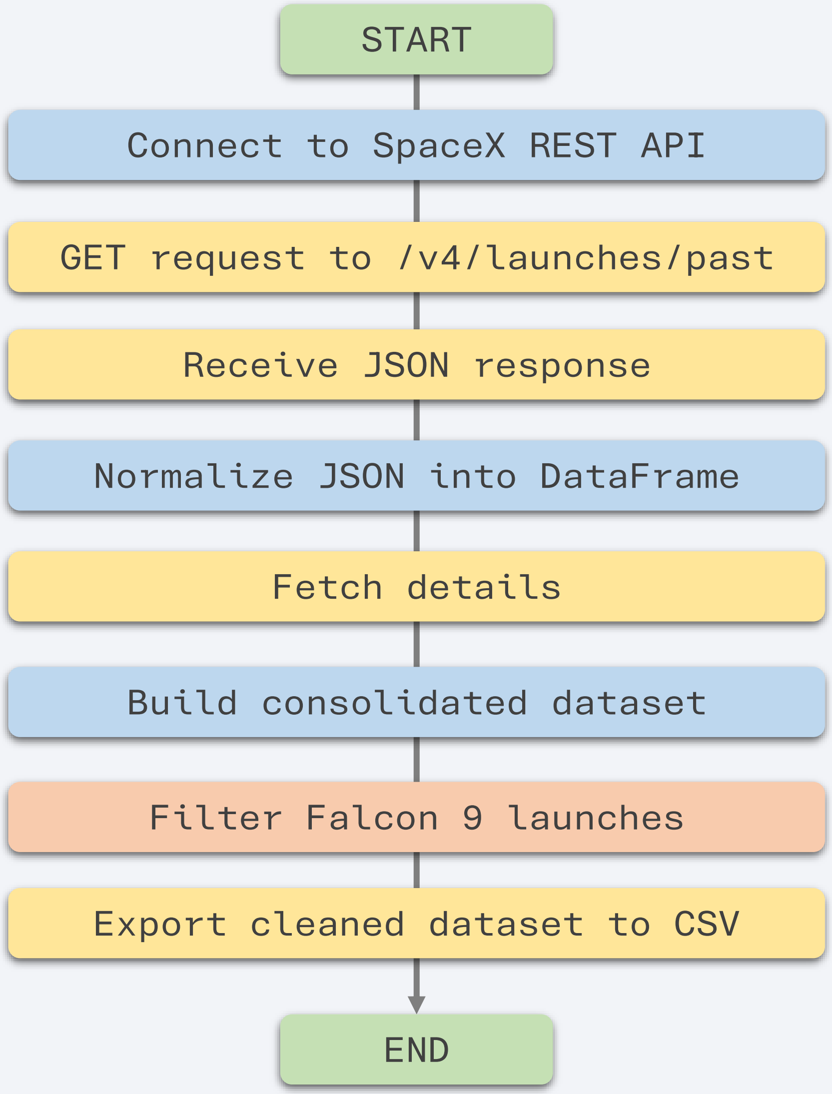
Both datasets were cleaned, filtered (to include only Falcon 9 and Falcon Heavy launches), checked for consistency, and merged to create a comprehensive dataset ready for further exploratory, interactive, and predictive analyses.

Data Collection – SpaceX API

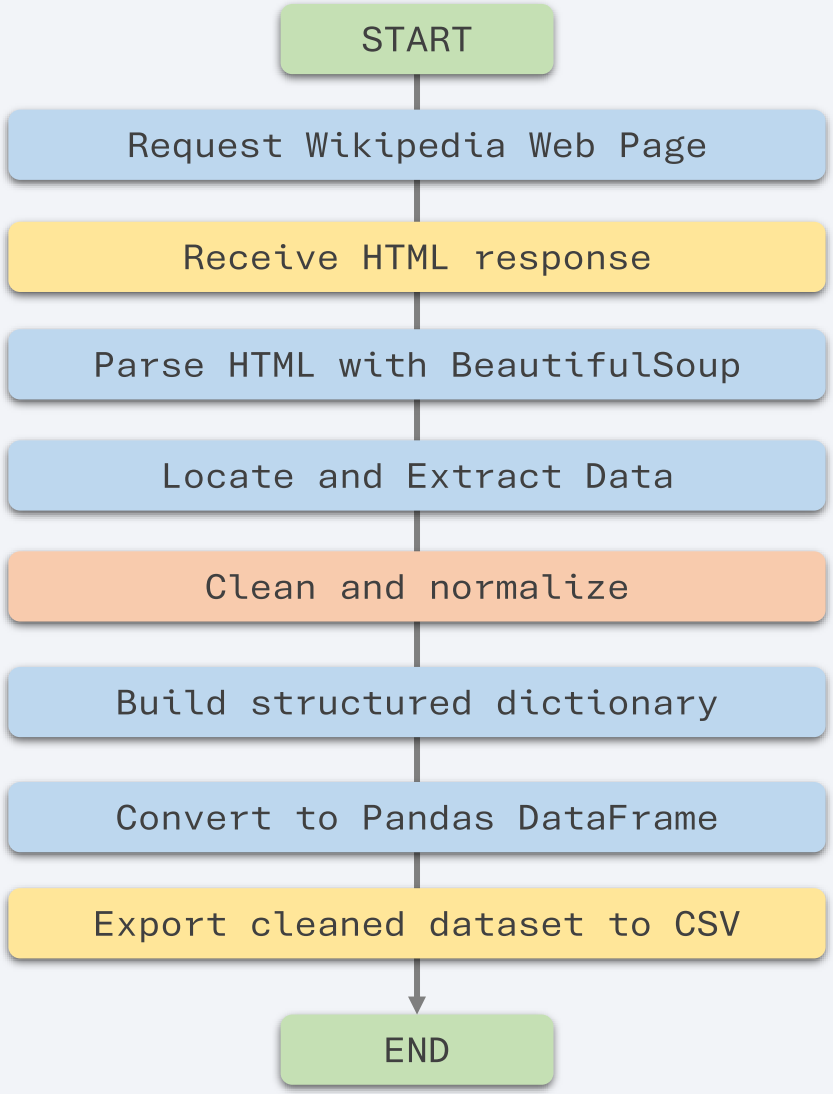
The flowchart for data collection is shown on the left.

In this phase, data on past SpaceX launches was collected through a public API. The raw data was transformed into a tabular format for analysis. Additional details were extracted and added to the dataset. Afterwards, the data was filtered to include only Falcon 9 launches. Finally, the cleaned dataset was saved in a text format.

GitHub: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/01-Spacex-data-collection-api.ipynb



Data Collection – Web Scraping



The flowchart for collecting data with Scraping is shown on the right.

In this phase, data on past Falcon 9 and Falcon Heavy launches was collected directly from a public Wikipedia page. The web content was requested through a web scraping technique. The information was then cleaned. Finally, the structured data was organized into a tabular format and saved as a CSV file for later analysis.

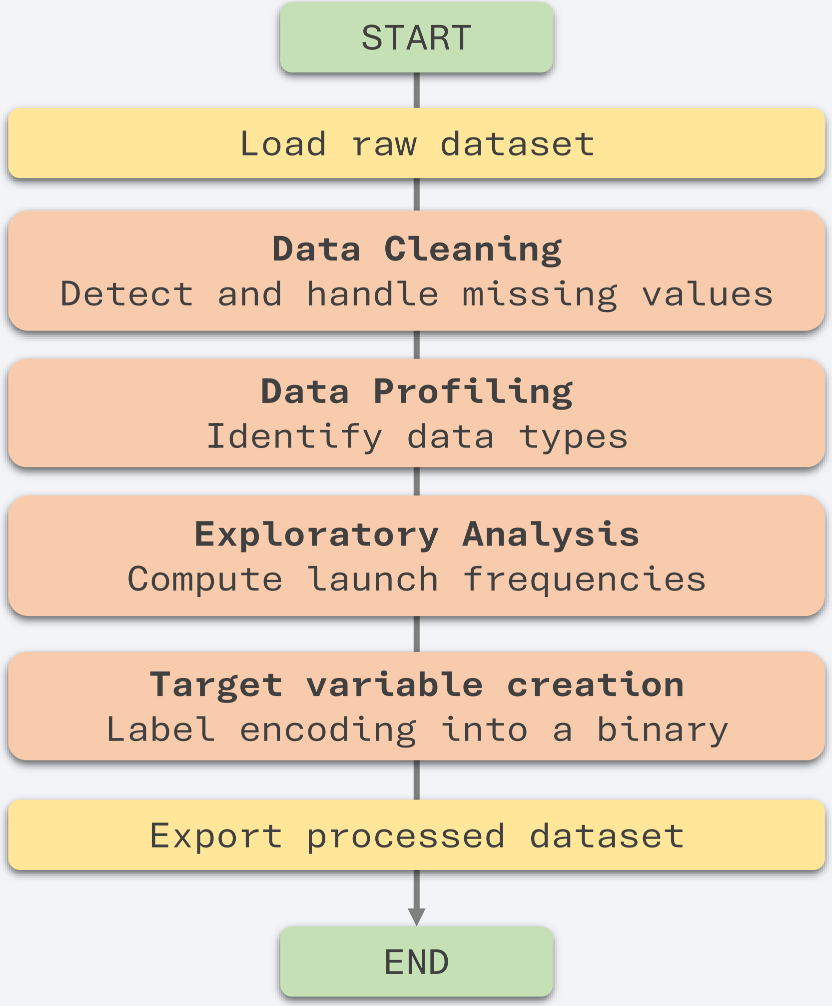
GitHub: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/02-Spacex-wiki-webscraping.ipynb

Data Wrangling

The flowchart for data Wrangling is shown on the left.

In this phase, the collected data was processed to prepare it for modeling. Missing values were detected and handled, and data types were identified (numerical vs categorical). Feature aggregation computed launch frequencies by site and orbit. Mission outcomes were then label-encoded into a binary target variable for supervised modeling. Finally, the processed dataset was exported.

GitHub: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/03-Spacex-Data-wrangling.ipynb



EDA with Data Visualization

The SpaceX Falcon 9 dataset was explored to understand relationships between key variables and the landing success.

- **Flight Number vs Payload MassType:** visualize how flight number and payload mass affect the success of the first stage landing. It helps identify trends in success with higher flight numbers or lighter payloads.
- **Flight Number vs Launch Site:** compare success rates across different launch sites and observe how success varies with flight number.
- **Payload Mass vs Launch Site:** examine the relationship between payload mass and launch site, detecting whether certain sites handle heavier payloads more successfully.
- **Success Rate by Orbit Type:** show the average success rate by orbit type, highlighting which orbits have higher reliability.
- **Flight Number vs Orbit Type:** explore whether success by orbit is related to the number of flights, revealing potential learning curves.
- **Payload Mass vs Orbit Type:** analyze how payload mass influences success for each orbit type, identifying orbits more sensitive to heavy payloads.
- **Yearly Launch Success Trend:** visualize the annual trend in launch success rate, showing improvements in reliability over time.

GitHub URL: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/04-EDA-data-visualization.ipynb

EDA with SQL

Summary of SQL Queries Performed for EDA

- Queried the unique launch sites using **DISTINCT**.
- Listed records of specific sites filtering with **LIKE** and **regular expressions**.
- Calculated the total payload mass by customer using the **aggregation function SUM**.
- Calculated the average payload mass by booster version using **the aggregation function AVG**.
- Identified the date of the first successful ground pad landing using **MIN**.
- Selected boosters with successful drone-ship landings filtering with **BETWEEN** and **combined conditions.**
- Counted the total number of successful and failed missions using **COUNT**.
- Retrieved boosters that carried the maximum payload using a **subquery** with **MAX**.
- Listed failed drone-ship landing results by month and year using **SUBSTR** to extract parts of the date.
- Ranked landing outcomes by count using **GROUP BY** and **ORDER BY** in **descending order**.

GitHub: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/05-EDA-with-sql.ipynb

Build an Interactive Map with Folium

- **Circles:** Added to mark the locations of each launch site to clearly highlight the launch site locations.
- **Markers & MarkerCluster:** Added to show individual launch outcomes (success/failure) for each launch while reducing clutter on the map.
- **Distance Markers:** Added to indicate the distances from the launch site to the closest coastline, highway, railway, and city, making proximity relationships clear.
- **Lines (PolyLine):** Added to draw connections between the launch sites and nearby points of interest to visualize geographic relationships.
- **MousePosition:** Added to allow interactive inspection of coordinates for any point on the map to aid exploration and further analysis.

GitHub: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/06-Spacex-launch-site-map-visualization.ipynb

Build a Dashboard with Plotly Dash

Plots/Interactions and Their Purpose:

- **Dropdown (Launch Site selection):** allows choosing all sites or a specific site to compare launches across sites or focus on one site.
- **Pie chart:** shows total successful launches by site, or success vs. failure for a selected site, helping to quickly visualize success distribution and identify top-performing sites.
- **Range Slider (Payload Mass):** filters launches by payload range, enabling users to explore how payload mass affects launch outcomes.
- **Scatter plot:** shows correlation between payload mass and launch outcome, color-coded by Booster Version, which helps to identify patterns in payload vs. success and evaluate booster performance.

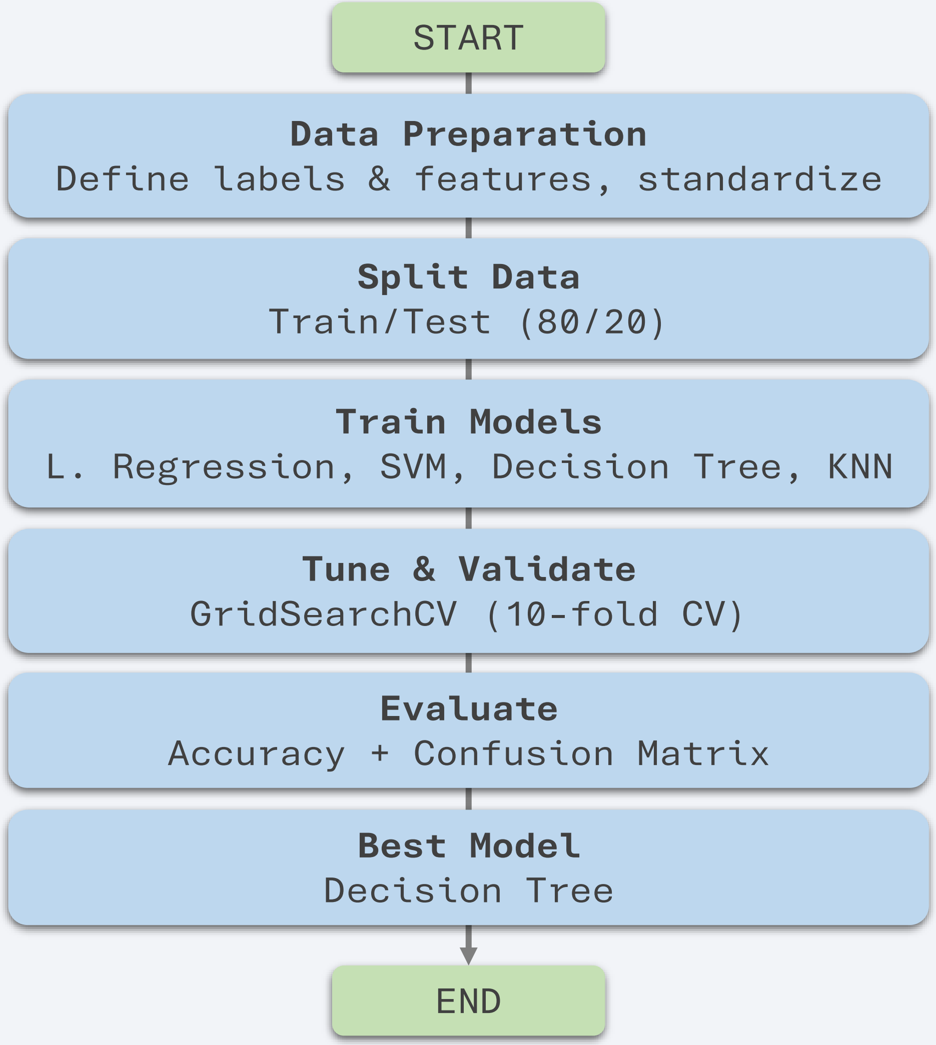
GitHub URL: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/07-Interactive-dashboard-with-Dash.py

Predictive Analysis (Classification)

The flowchart for predictive analysis is shown on the left.

We built and evaluated four classification models (Logistic Regression, SVM, Decision Tree, KNN) using standardized data. Hyperparameter tuning with GridSearchCV improved performance, and the Decision Tree was identified as the best model with 88.9% validation accuracy and ~83% test accuracy.

GitHub: https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning/blob/main/08-Spacex-machine-learning-prediction.ipynb



Results

- **Exploratory data analysis results** revealed key patterns in SpaceX launches: certain launch sites handle different payload mass ranges, and success rates vary by orbit, being higher for Polar, LEO, and ISS. Heavier payloads tend to show higher landing success in some orbits, and repeated flights improve outcomes. Some sites, like VAFB SLC 4E, do not handle very heavy payloads, whereas CCAFS SLC-40 operates across a wide mass range.
- **Interactive analytics demo in screenshots:** Interactive Folium maps showed that all launch sites are near the coastline for safety and close to highways and railways for logistics, with Florida sites at lower latitudes to take advantage of Earth's rotation and VAFB further north. Sites are also strategically distant from cities to minimize risk, and success/failure markers reveal consistently high success rates at some sites, with clusters highlighting clear patterns.
- **Predictive analysis results** using Logistic Regression, SVM, Decision Tree, and KNN models achieved around 83% test accuracy, with the Decision Tree performing best on validation (88.9%). Confusion matrices indicate false positives are the main challenge, meaning some failed launches were incorrectly predicted as successful.

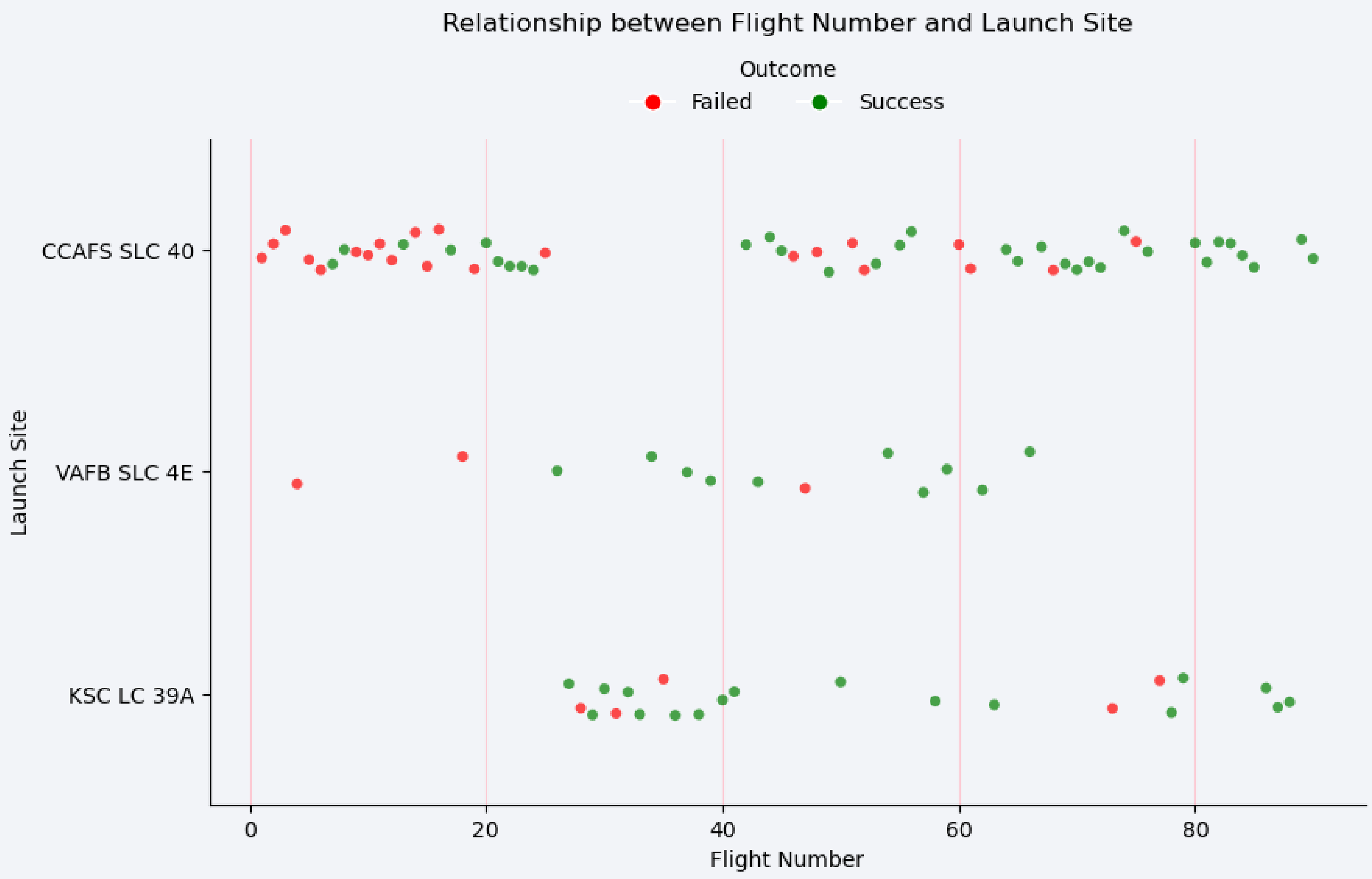
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

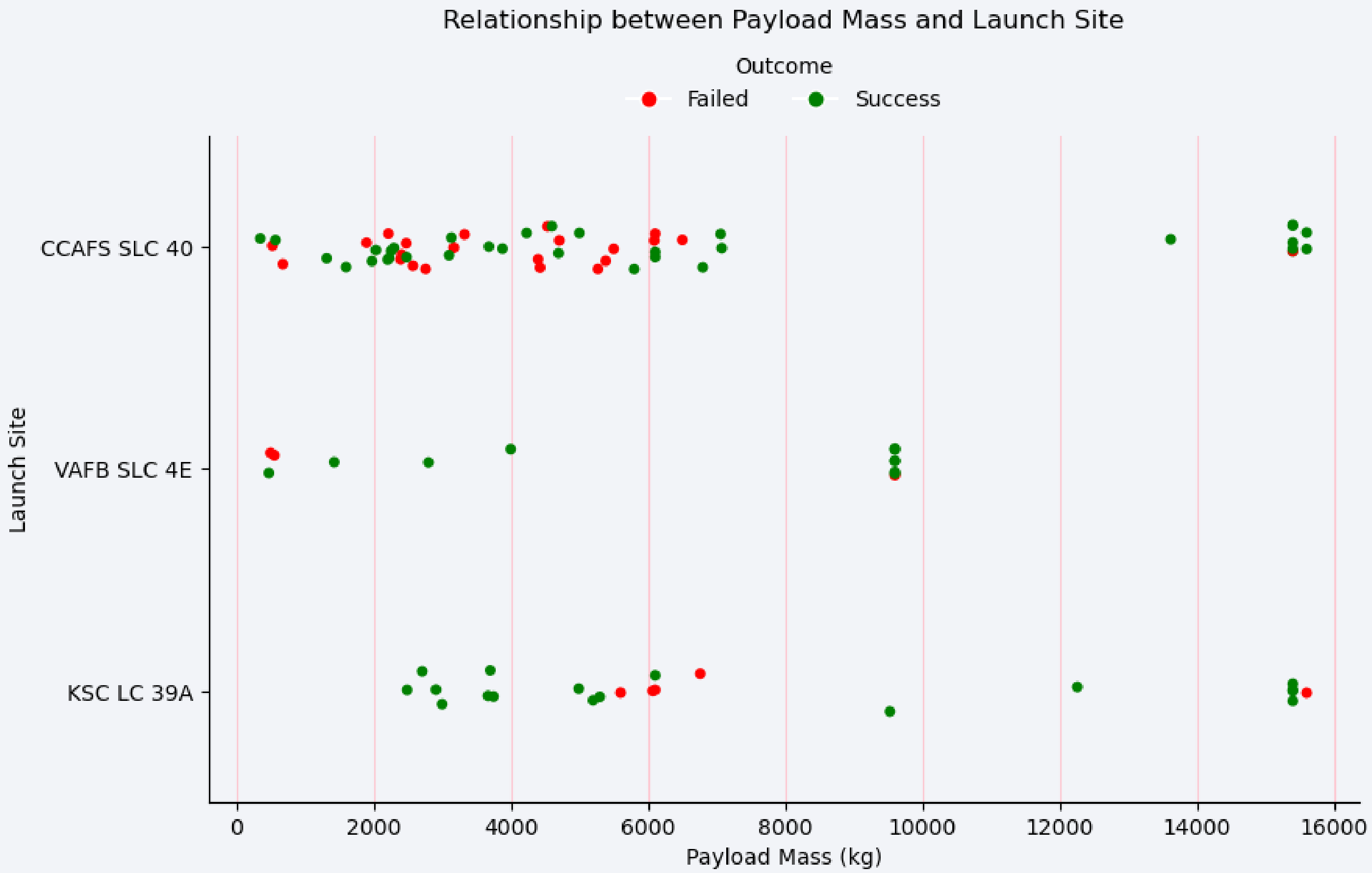
The plot shows the progression of Falcon 9 operations across multiple sites.

- The plot shows the progression of Falcon 9 operations across multiple sites.
- Early flights (low Flight Number) were concentrated at CCAFS SLC-40.
- Each dot represents a launch attempt at a specific site.
- Over time, launches expanded to VAFB SLC-4E and KSC LC-39A.



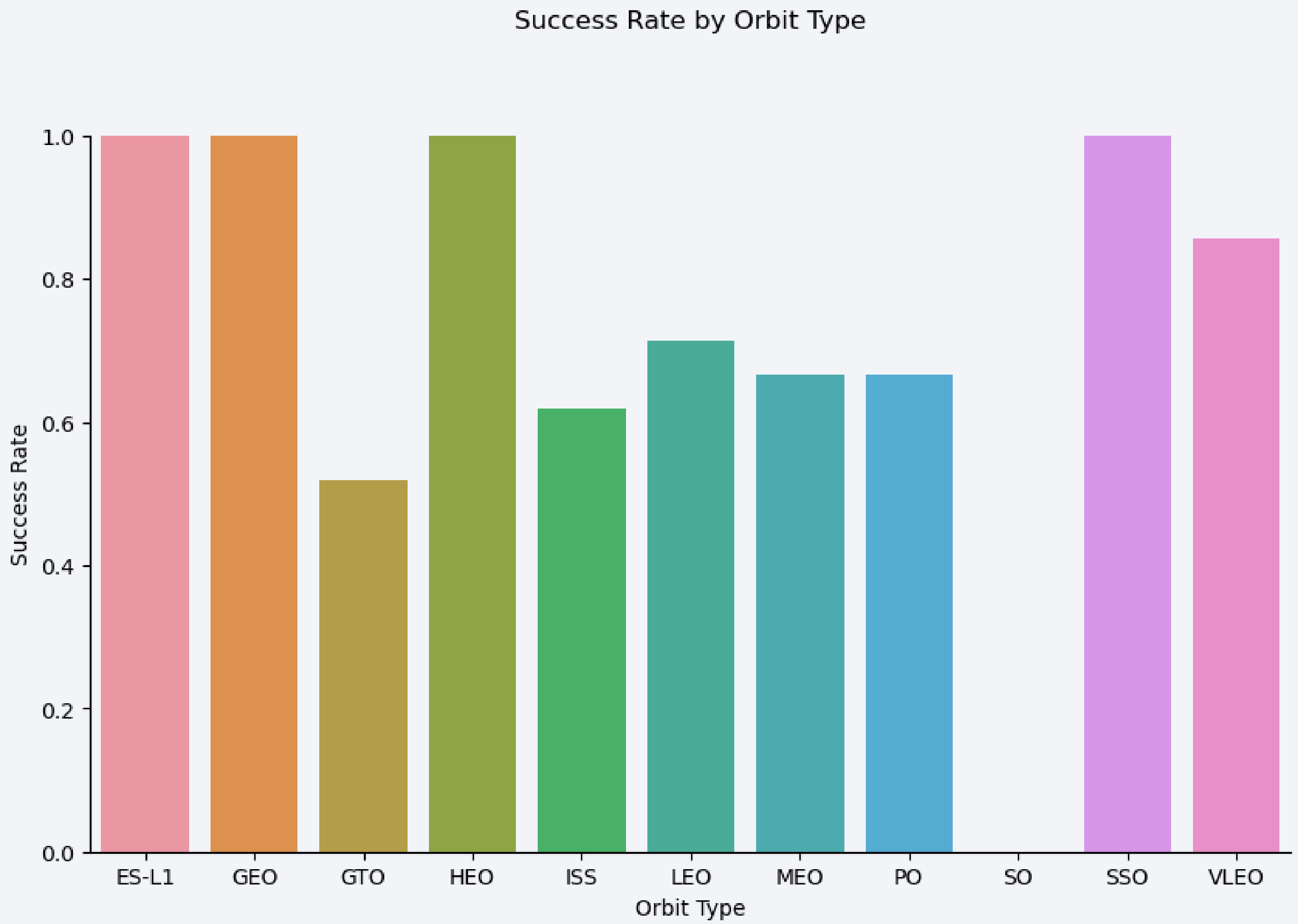
Payload vs. Launch Site

- Each point represents a mission with its corresponding payload mass at a launch site.
- CCAFS SLC-40 shows the highest concentration of missions with varied payloads.
- KSC LC-39A and VAFB SLC-4E appear mainly in later launches.
- Heavier payloads tend to be concentrated at certain sites, reflecting the capacity of each launch platform.



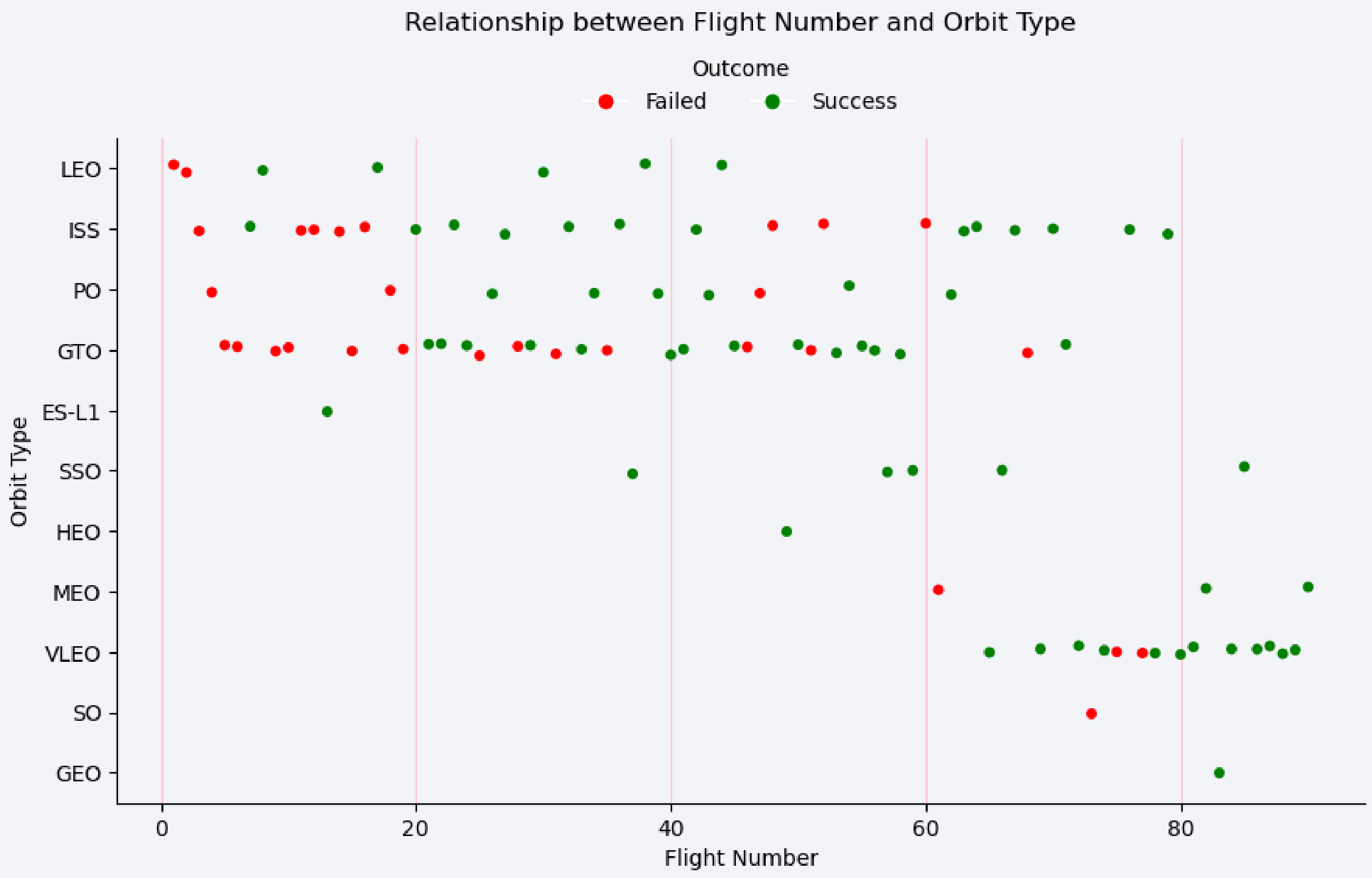
Success Rate vs. Orbit Type

- Each bar represents the success rate of missions for a specific orbit type.
- LEO and ISS missions show the highest success rates.
- Orbits like GTO and PO have more variability in outcomes.
- The chart helps identify which orbit types have historically higher or lower mission success.



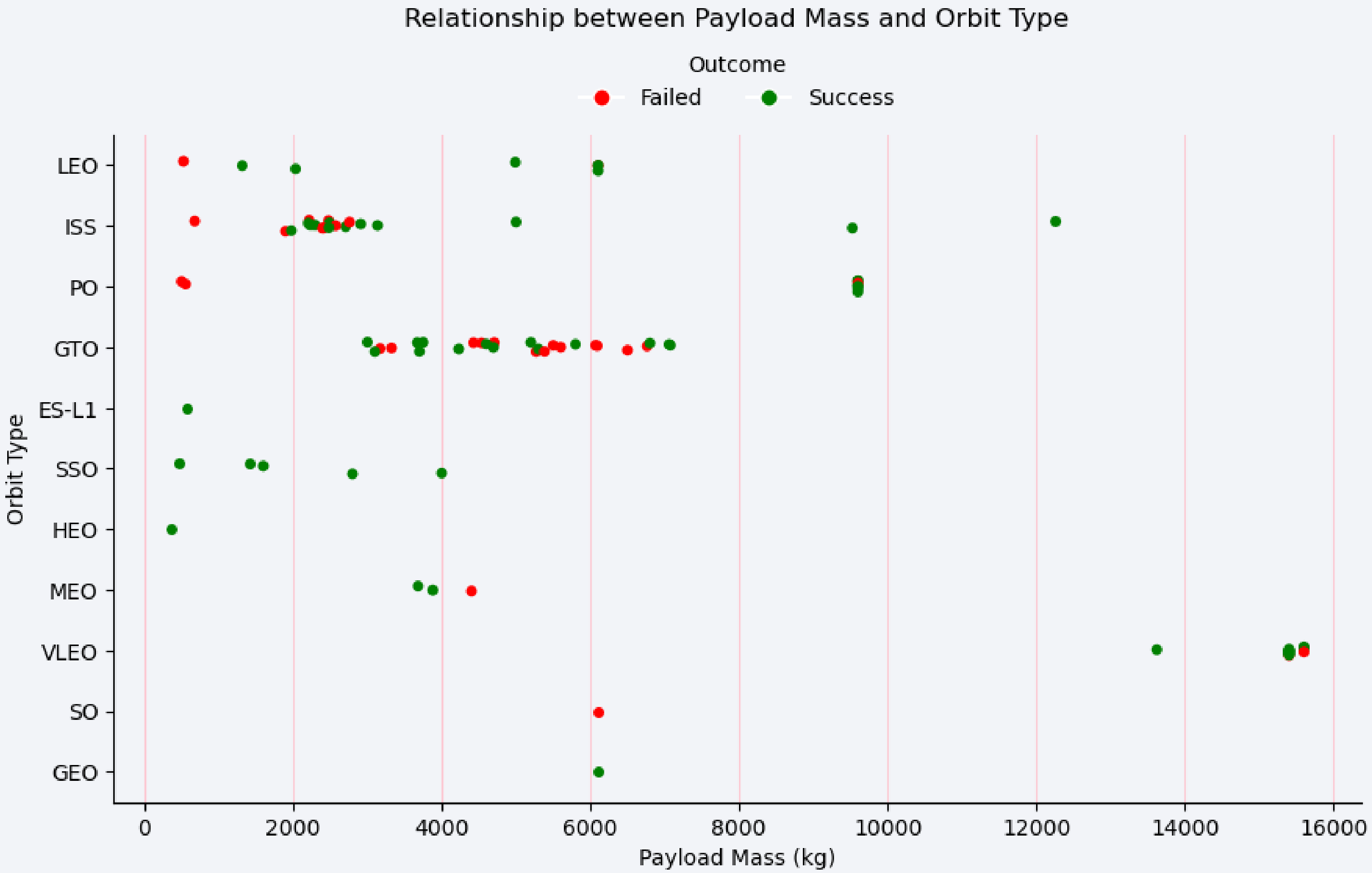
Flight Number vs. Orbit Type

- Each point represents a mission, with the flight number on the x-axis and the orbit type on the y-axis.
- The plot shows which orbit types were targeted at different stages of SpaceX's launch history.
- Clusters of points indicate frequent missions to specific orbits (e.g., LEO and ISS).
- The scatter plot helps visualize trends in mission frequency by orbit type over time.



Payload vs. Orbit Type

- Each point represents a mission, with payload mass on the x-axis and orbit type on the y-axis.
- The plot shows the distribution of payloads for different orbit types.
- Clusters reveal common payload ranges for certain orbits (e.g., lower mass for LEO, higher mass for GTO).
- The scatter plot helps understand how payload size varies depending on the mission's target orbit.



Launch Success Yearly Trend

- Each point/line represents the number of successful launches per year.
- Shows the evolution of Falcon 9 mission success over time.
- A gradual increase in success rate is observed after the initial test years.
- Helps identify improvements in launch reliability and technological maturity.

All Launch Site Names

SQL Code: `SELECT DISTINCT(Launch_Site) FROM SPACEXTBL;`

Output:

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

Explanation:

This query shows all the unique launch sites used in SpaceX missions. Using the DISTINCT keyword in SQL, we were able to filter out duplicate entries and display only one instance of each launch site. This helps us quickly identify all the different locations from which SpaceX has launched rockets.

Launch Site Names Begin with 'CCA'

SQL Code: `SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;`

Output:

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

Explanation:

This code displays 5 records from the SPACEXTBL table where the launch site name starts with 'CCA'. The LIKE 'CCA%' condition filters the sites, and LIMIT 5 shows only the first results as representative examples.

Total Payload Mass

SQL Code: `SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';`

Output:

sum(PAYLOAD_MASS__KG_)
45596

Explanation:

This query calculates the total payload mass carried by all boosters for missions where the customer is 'NASA (CRS)'. The SUM(PAYLOAD_MASS__KG_) function adds up the payloads for all matching records, giving the combined weight of all NASA payloads in the dataset.

Average Payload Mass by F9 v1.1

SQL Code:

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

Output:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

Explanation:

This query calculates the average payload mass for all missions using booster versions that start with 'F9 v1.1'. The AVG(PAYLOAD_MASS__KG_) function computes the mean of the PAYLOAD_MASS__KG_ values for all matching records, giving a single value representing the typical payload carried by F9 v1.1 boosters.

First Successful Ground Landing Date

SQL Code:

```
SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

Output:

MIN(Date)
2015-12-22

Explanation:

This query finds the earliest date when a booster successfully landed on the ground pad. The MIN(Date) function returns the smallest (i.e., first) date from all records where Landing_Outcome is 'Success (ground pad)', showing when SpaceX achieved its first ground landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Code:

```
SELECT Booster_Version FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (drone ship)’
AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

Output:

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

Explanation:

This query lists all booster versions that successfully landed on a drone ship and carried a payload between 4000 and 6000 kg. It filters the table using Landing_Outcome = 'Success (drone ship)' and the BETWEEN operator on PAYLOAD_MASS__KG_ to select only launches within that payload range.

Total Number of Successful and Failure Mission Outcomes

SQL Code:

```
SELECT
(SELECT COUNT(*) FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Success%') AS Total_Success,
(SELECT COUNT(*) FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Failure%') AS Total_Failure;
```

Output:

Total_Success	Total_Failure
100	1

Explanation:

This query summarizes the total number of successful and failed missions. It groups all outcomes that start with “Success” and all outcomes that start with “Failure”. The result shows a single row with two columns—Total_Success and Total_Failure—representing the total number of successful and failed missions, respectively.

Boosters Carried Maximum Payload

SQL Code:

```
SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

Output:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Explanation:

This query retrieves all Booster_Version entries that have carried the maximum payload mass recorded in the table. First, the subquery (select max(PAYLOAD_MASS__KG_) from SPACEXTBL) calculates the highest value of PAYLOAD_MASS__KG_ in the dataset. Then, the outer query selects all records where PAYLOAD_MASS__KG_ matches this maximum value, thus showing all boosters that achieved this maximum payload. This ensures that all boosters sharing the same maximum payload are included.

2015 Launch Records

SQL Code:

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

Output:

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

Explanation:

This query extracts the month, landing outcome, booster version, and launch site for all missions in 2015 that ended in a drone ship failure. Since SQLite does not directly support month names, we use substr(Date, 6, 2) to extract the month number from the Date column. The condition substr(Date, 1, 4) = '2015' ensures we only consider records from the year 2015, and Landing_Outcome = 'Failure (drone ship)' filters for drone ship failures specifically. The result shows which boosters failed on a drone ship and in which month of 2015.

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

SQL Code:

```
SELECT Landing_Outcome, COUNT(*) as Count FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20’
GROUP BY Landing_Outcome
ORDER BY count desc;
```

Output:

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

Explanation:

This query counts how many times each type of landing outcome occurred between June 4, 2010, and March 20, 2017. The WHERE clause filters the records within the specified date range. The GROUP BY Landing_Outcome groups the data by each unique landing outcome so that COUNT(*) can calculate the total for each group. Finally, ORDER BY Count DESC sorts the results from the most frequent landing outcomes to the least frequent.

Section 3

Launch Sites Proximities Analysis

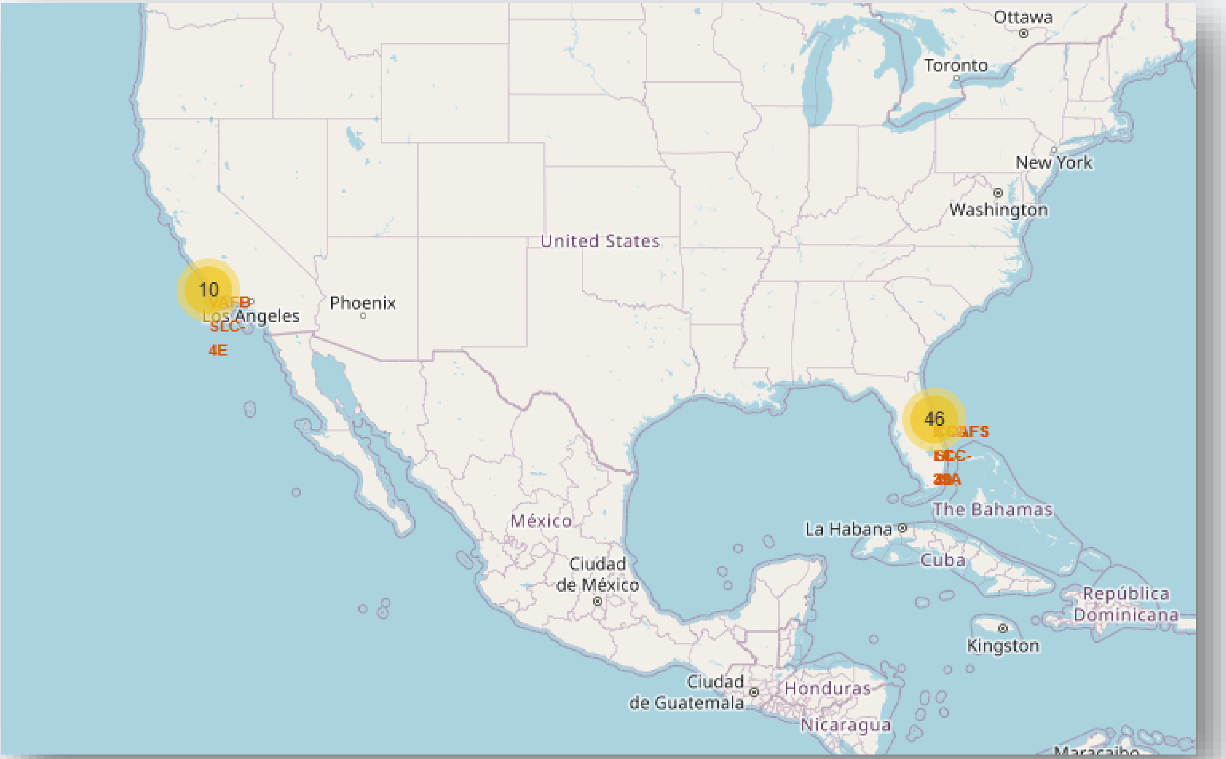
Marking SpaceX Launch Sites

Key Elements on the Map

- **Circles**: show the exact coordinates of each launch site.
- **Text labels**: indicate the launch site names (e.g., CCAFS LC-40).
- **Map centered view**: adjusted to display all launch sites within the U.S.

Main Findings

- SpaceX has four main launch sites in the U.S.
- Three in Florida (Cape Canaveral and Kennedy Space Center).
- One in California (Vandenberg).
- All sites are located close to the coast, which is important for safety during launches.
- None of the sites are located near the Equator; the lowest latitude is around 28.5°N.



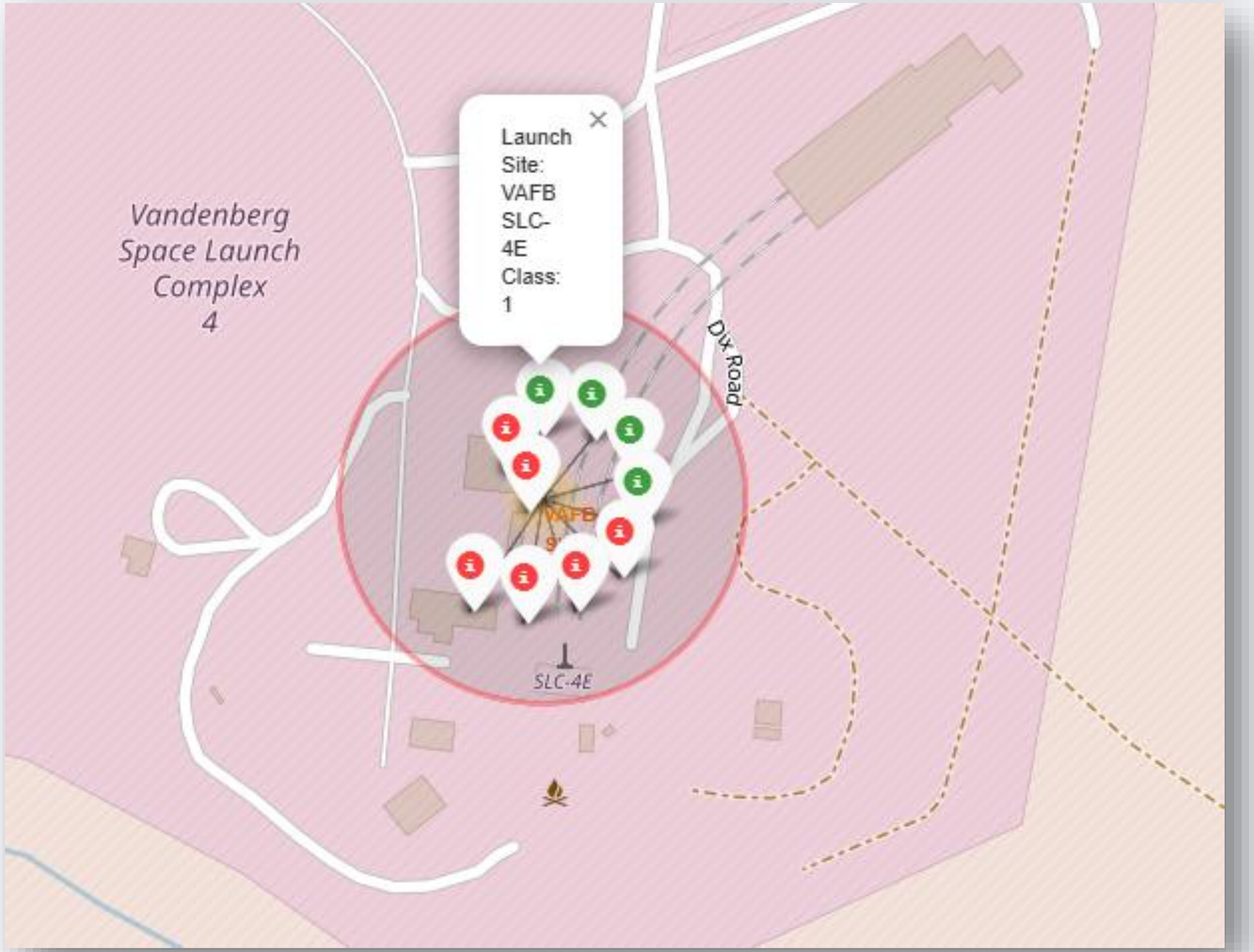
Color-Labeled Launch Outcomes

Key Elements on the Map

- **Green markers:** represent successful launches.
- **Red markers:** represent failed launches.
- **Marker clusters:** group overlapping launches at the same site for easier visualization.
- **Popup info:** clicking a marker shows the site name and launch outcome.

Main Findings

- The map clearly shows success and failure distribution by site.
- Florida sites (Cape Canaveral and Kennedy Space Center) have many launches, with a high number of green markers (successes).
- Vandenberg (California) shows fewer launches but still with visible outcomes.
- Marker clusters help to quickly identify which sites have higher success rates.



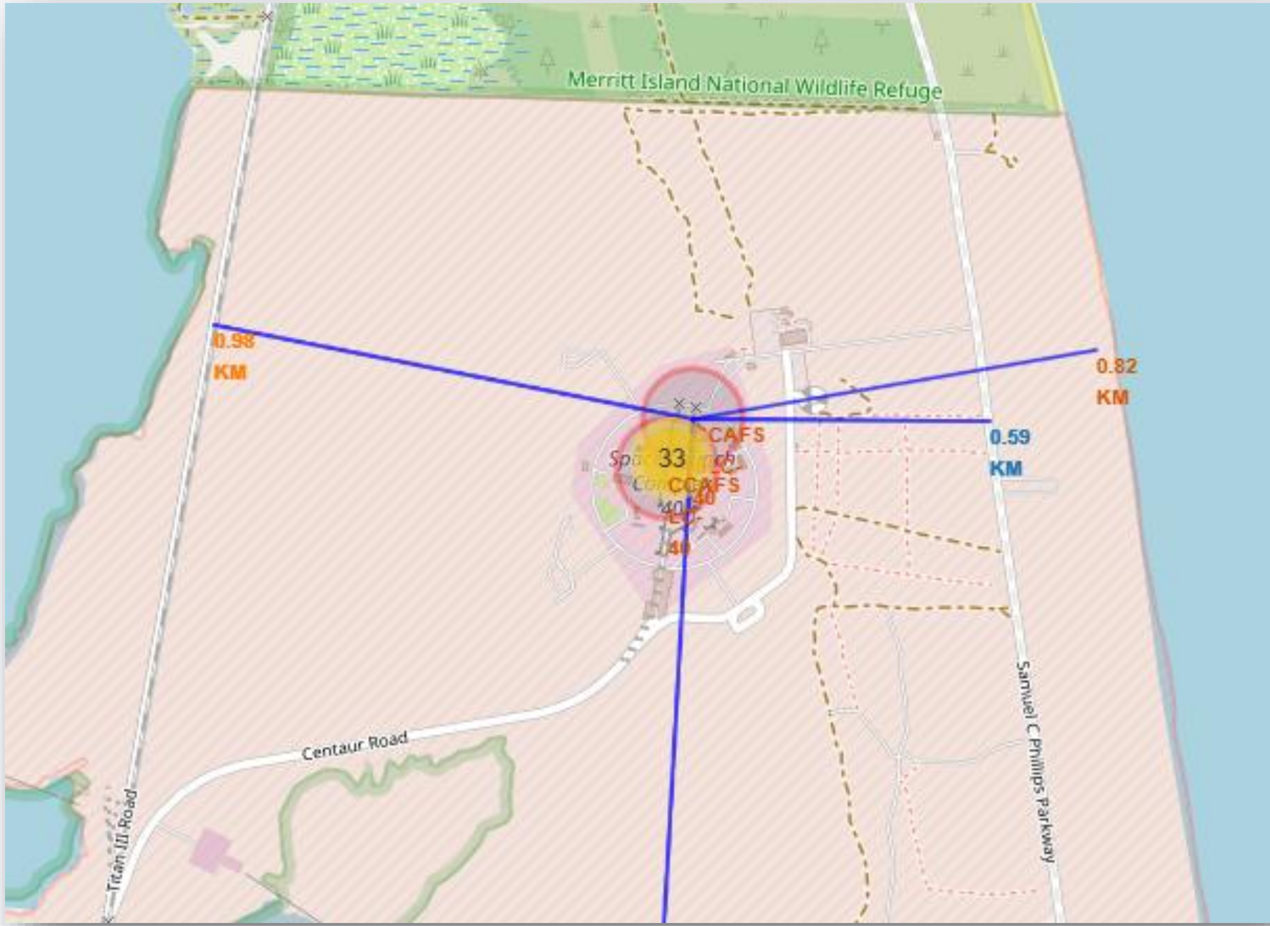
Distances from Launch Sites to Nearby Infrastructure

Key Elements on the Map

- **Blue lines (Polylines):** connect the launch site with selected points (coastline, highway, railway, city).
- **Distance labels:** markers display the calculated distance in kilometers.
- **Reference points:** coastline, roads, railways, and cities marked as target locations.

Main Findings

- Launch sites are located very close to the coastline, typically less than 1 km.
- They are also near highways and railways, enabling logistics and transport.
- Sites are placed at a safe distance from cities, reducing risks in case of launch failures.
- The locations show a strategic balance: near infrastructure for operations, but away from populated areas for safety.



Section 4

Build a Dashboard with Plotly Dash

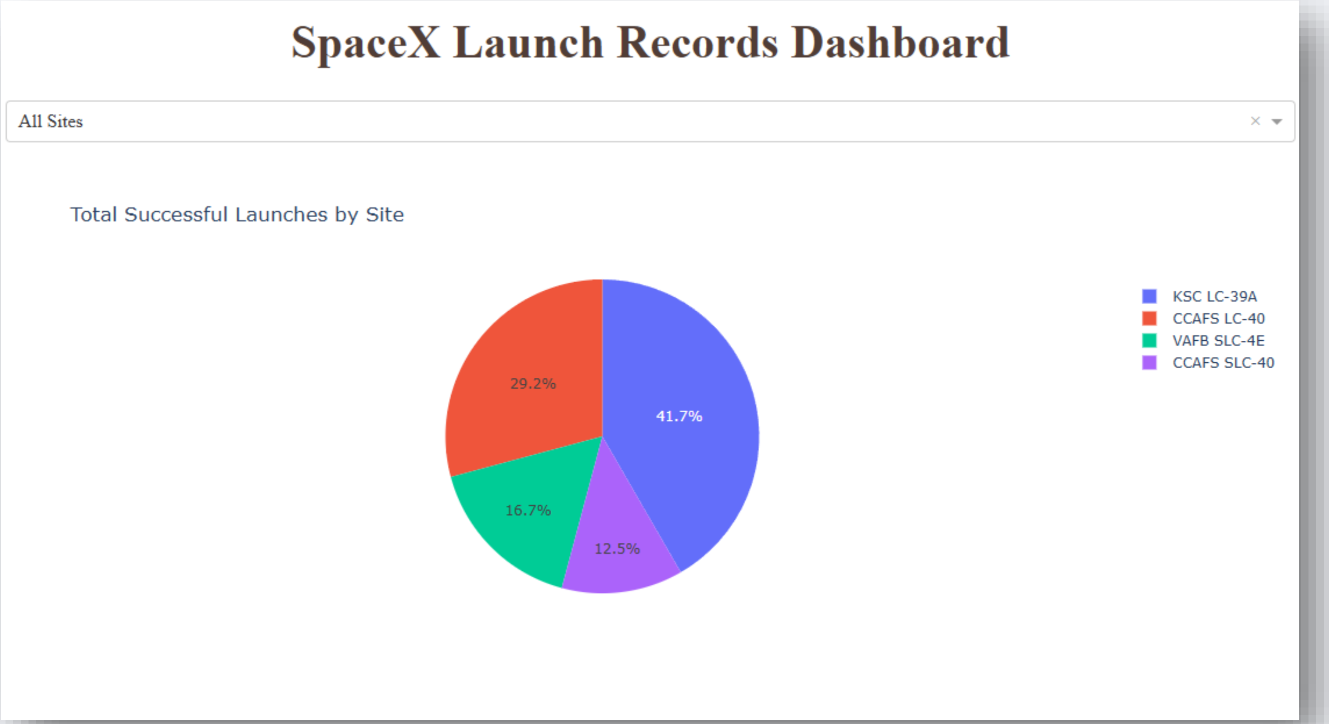
Total Successful Launches by SpaceX Launch Site

Pie chart:

- El gráfico muestra la proporción de lanzamientos exitosos por cada sitio de lanzamiento de SpaceX.
- Cada sector del pastel representa un sitio de lanzamiento distinto.
- El tamaño de cada sector indica el número total de lanzamientos exitosos en ese sitio.

Hallazgos importantes:

- El sitio con el sector más grande tiene el mayor número de lanzamientos exitosos.
- Los sitios con sectores más pequeños han tenido menos éxitos totales, pero puede reflejar que han tenido menos lanzamientos en general.



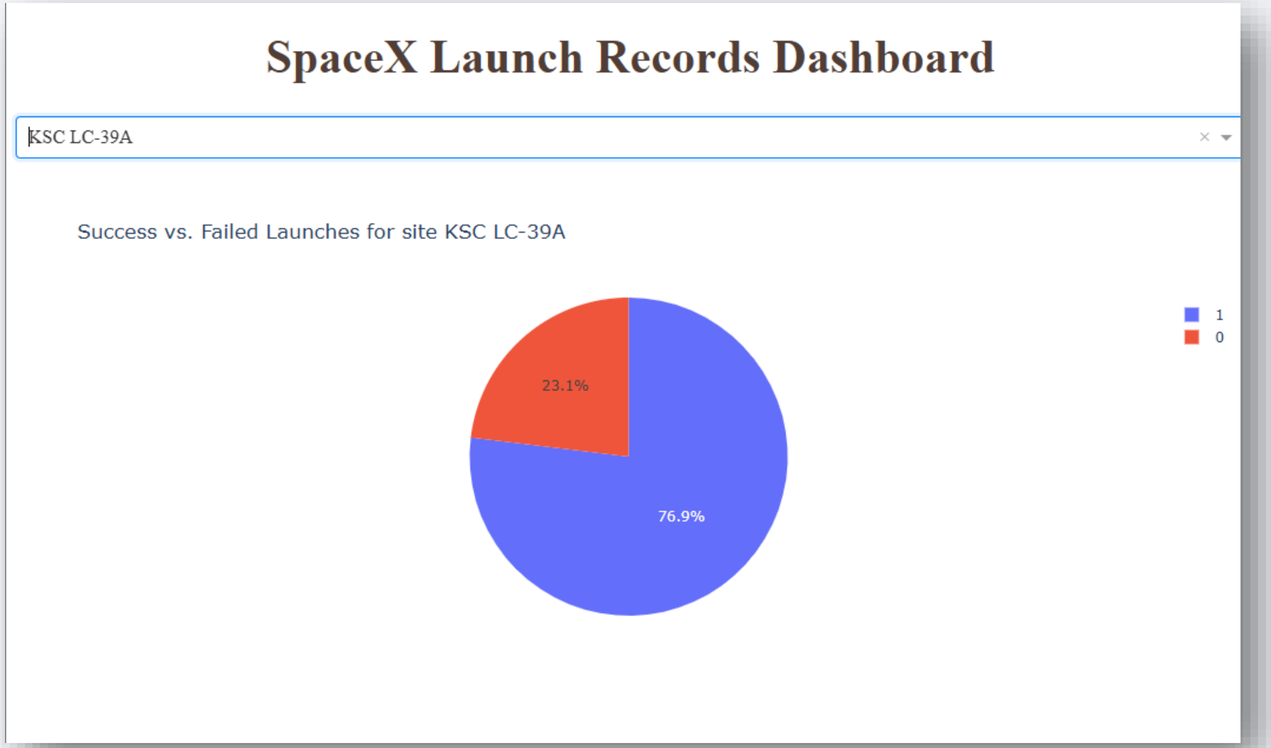
Total Successful Launches by SpaceX Launch Site

Pie chart:

- The chart shows the proportion of successful vs. failed launches specifically for the site KSC LC-39A, which has the highest success rate.
- Each sector of the pie represents a launch outcome: success: 76.9%, failure: 23.1%.
- This allows us to visualize the relative efficiency of the site, showing that most launches have been successful.

Hallazgos importantes:

- 76.9% of launches at KSC LC-39A have been successful, confirming it as the most reliable site in terms of relative success.
- 23.1% of launches failed, highlighting that although less frequent, failures do occur and should be considered in risk analysis and planning.



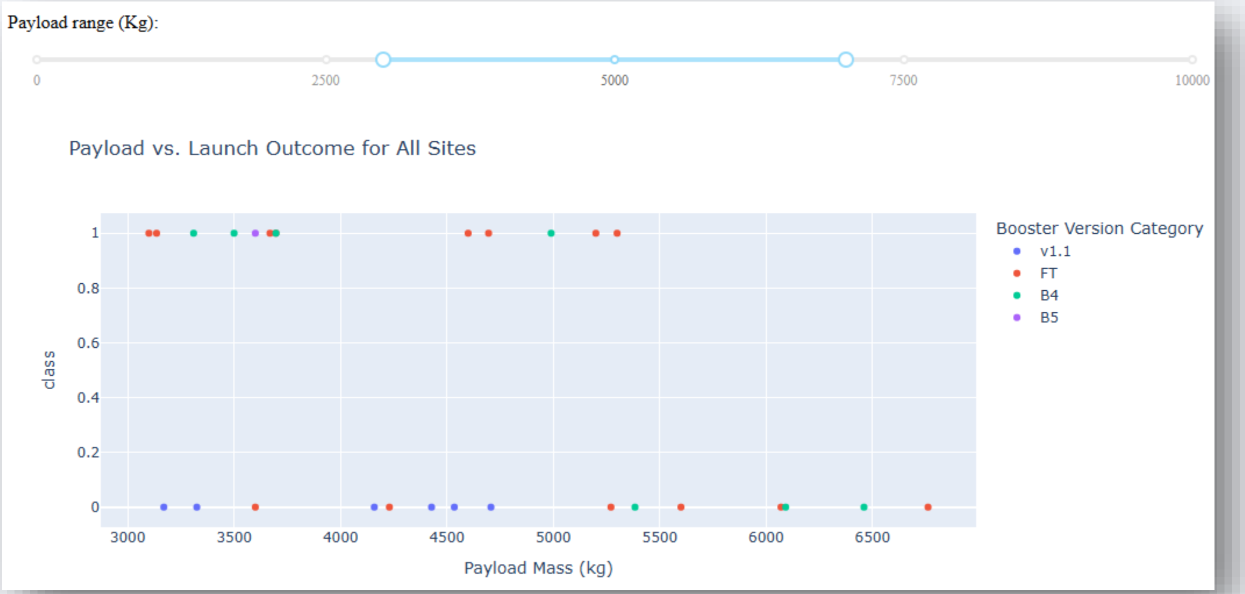
Payload vs. Launch Outcome for (3000–7000 kg)

Pie chart:

- The plot shows the relationship between payload mass and launch outcome for the selected site.
- Each point is color-coded by Booster Version, allowing comparison of booster performance.
- Payload is filtered to 3000–7000 kg to focus on this range.

Hallazgos importantes:

- This range highlights the most successful launches and the boosters with the highest reliability.
- Shows how payload mass impacts success, and which booster-payload combinations performed best.
- Useful for strategic decisions on booster selection and payload planning.

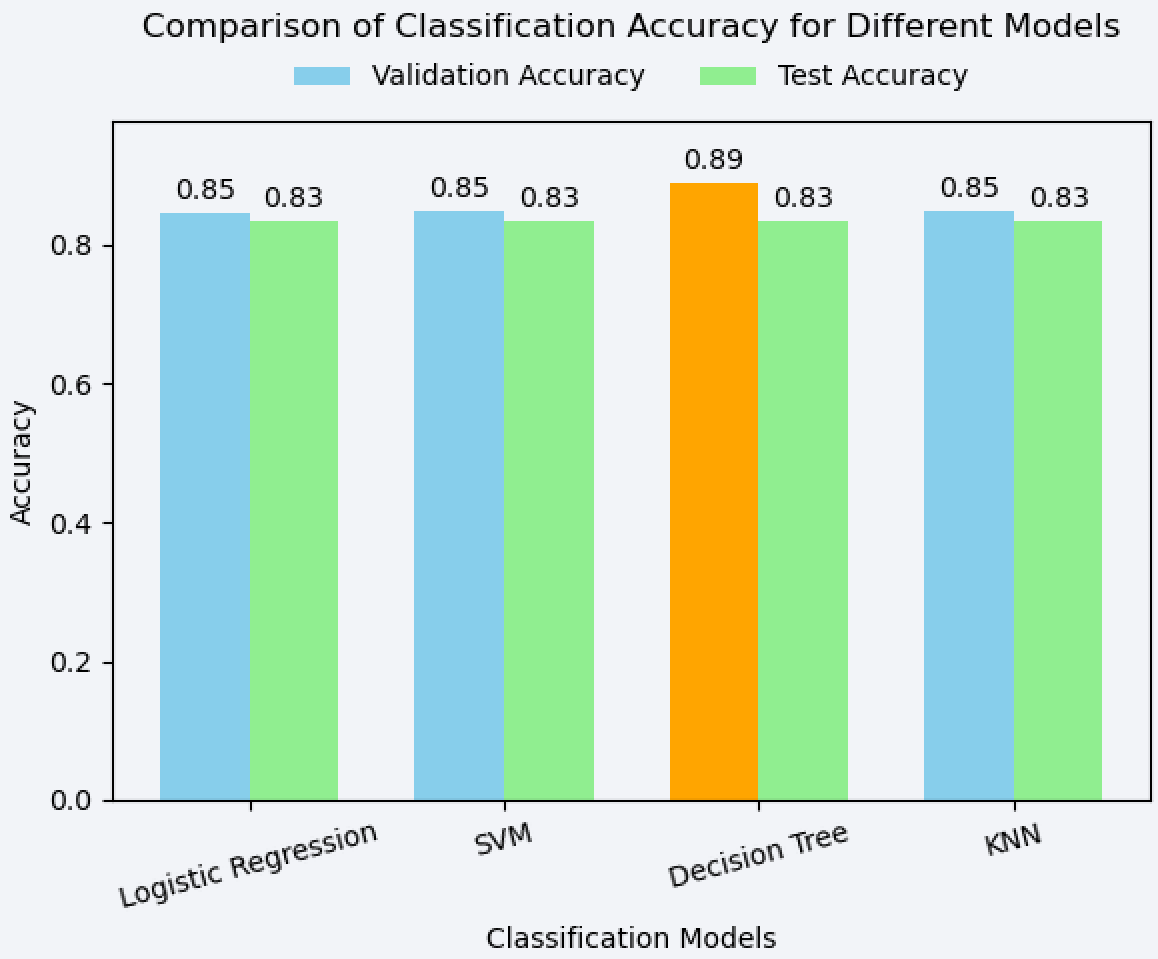


Section 5

Predictive Analysis (Classification)

Classification Accuracy

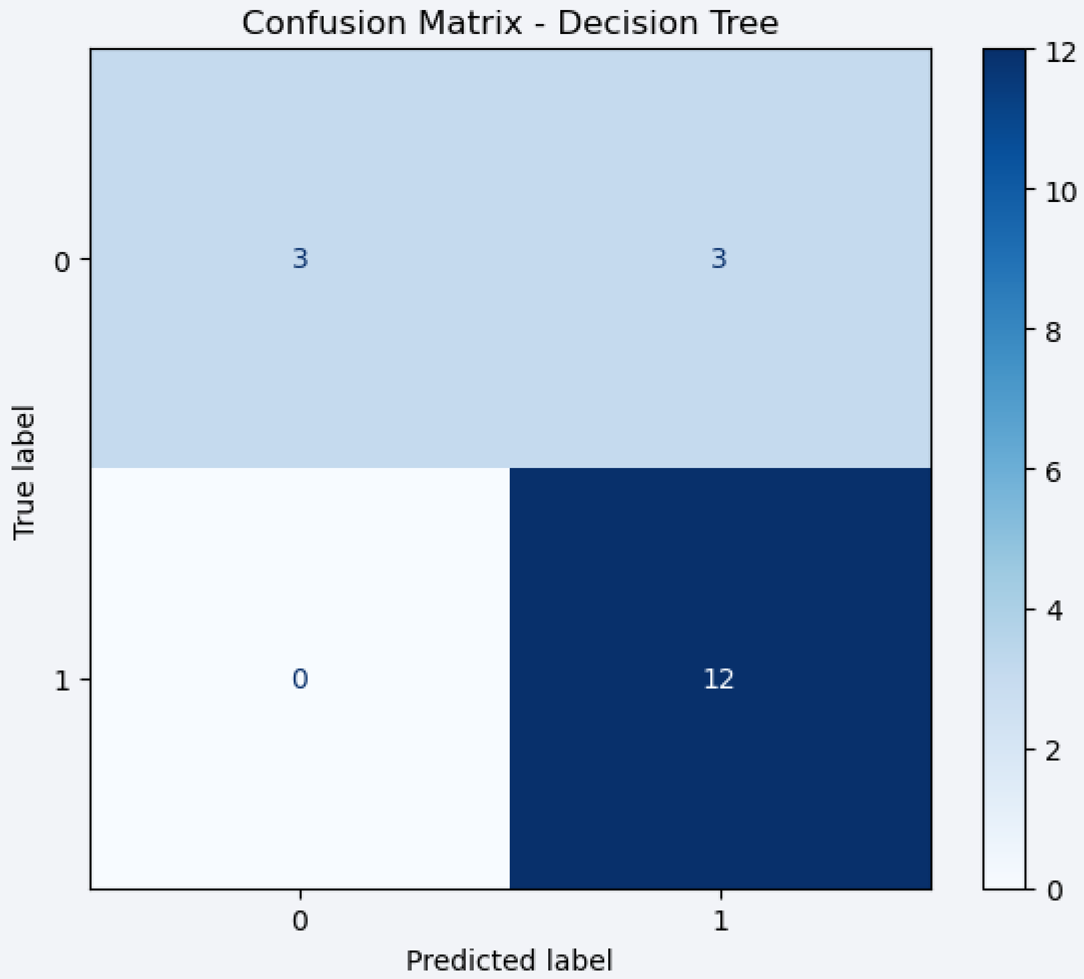
We compared the performance of four classification models—Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN)—using validation and test accuracy. All models achieved similar test accuracy of 83.3%, while the Decision Tree had the highest validation accuracy (88.9%), suggesting better fitting on the training data. The comparison is visualized in a bar chart, highlighting Decision Tree as the model with the highest validation accuracy.



Confusion Matrix

The confusion matrix of the Decision Tree model shows the distribution of correct and incorrect predictions for each class. The values along the diagonal represent correct predictions, while the off-diagonal values indicate misclassifications.

Although the matrix reflects accuracy on the test set, the model's validation accuracy was 0.89, meaning the model correctly predicted approximately 89% of examples during cross-validation. This confirms that the Decision Tree is the best-performing model among those evaluated, demonstrating good fit without significant overfitting.



Conclusions

Decision Tree as Best Model: The Decision Tree achieved the highest validation accuracy (0.89), outperforming Logistic Regression, SVM, and KNN. This indicates it captures the underlying patterns in the training data better than the other models.

Consistent Test Performance: All models had similar test accuracy (~0.83), which shows that despite differences in training performance, the models generalize reasonably well to unseen data. This reinforces the robustness of the Decision Tree.

Confusion Matrix Insights: The confusion matrix of the Decision Tree reveals that the majority of predictions were correct, with relatively few misclassifications. This confirms the model's ability to accurately differentiate between classes.

Validation and Cross-Validation Reliability: Using cross-validation ensured the model was evaluated on multiple subsets of data, reducing the risk of overfitting. The Decision Tree's strong validation performance indicates a reliable balance between fitting the training data and generalizing to new data.

Appendix

GitHub URL for this project:

- https://github.com/fer78/Space-Launch-Optimization-with-Machine-Learning

Thank you!