



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Caio Bandeira
2025-08-07



Outline

- **Executive Summary** – This project aims to evaluate SpaceX's launch data to identify patterns and insights that can guide SpaceY in developing more efficient and reliable rocket launches.
- **Introduction** – By leveraging publicly available datasets, we explore the operational performance, cost factors, and technological trends that have enabled SpaceX's success.
- **Methodology** – The analysis was conducted through data collection, cleaning, exploratory data analysis (EDA), interactive visualizations, SQL queries, predictive modeling, and dashboard development.
- **Results** – Key findings highlight the factors influencing launch success, cost optimization opportunities, and conditions favoring the reuse of the first stage.
- **Conclusion** – The study provides actionable insights for SpaceY to enhance launch efficiency, reduce costs, and improve mission reliability.
- **Appendix** – Supplementary materials include SQL outputs, EDA charts, interactive maps, dashboards, and detailed predictive model evaluations.

Executive Summary

- **Summary of Methodologies** – The analysis was conducted using multiple supervised machine learning algorithms to predict the success of rocket first-stage landings. Data preprocessing included cleaning, feature selection, and splitting into training and test sets. Four classification models were evaluated: Logistic Regression, Support Vector Machines (SVM), Decision Tree Classifier, and K-Nearest Neighbors (KNN). Hyperparameter tuning was performed using GridSearchCV with 10-fold cross-validation to optimize model performance. Model evaluation was carried out using accuracy scores and confusion matrices to assess predictive capabilities.
- **Summary of All Results** – The best cross-validation accuracy was achieved by the Decision Tree Classifier (88.75%), followed by KNN (84.82%), SVM (84.82%), and Logistic Regression (84.64%). On the test set, all models achieved the same accuracy of **83.33%**, indicating stable performance across different algorithms. Confusion matrices showed that the Decision Tree model correctly classified most positive cases with minimal false negatives, while KNN achieved the best overall parameter combination (`algorithm='auto'`, `n_neighbors=10`, `p=1`). These results suggest that multiple models provide robust predictions, with Decision Trees offering a slight edge in training accuracy but no significant difference in test performance.

Introduction

- **Project Background and Context** – SpaceX has transformed the aerospace industry by achieving cost reductions, operational efficiency, and technological advancements in rocket launches, particularly through the reuse of the first stage. As SpaceY aims to compete in this market, understanding the operational and technical patterns behind SpaceX's success is critical. By analyzing publicly available launch data, this project seeks to uncover insights that can guide SpaceY's design, engineering, and strategic decisions to achieve higher reliability and cost-effectiveness in future launches.
- **Problems We Want to Find Answers** – What are the key factors influencing launch success? How can we predict the likelihood of a successful first-stage landing? Which conditions most strongly affect reusability? How can SpaceY optimize costs while maintaining or improving performance? Which machine learning model delivers the most reliable predictions for launch outcomes?

Section 1

Methodology

Methodology

Executive Summary – This project analyzes SpaceX’s historical launch data to identify operational patterns, performance drivers, and cost optimization opportunities that can guide SpaceY in developing more efficient and reliable rocket launches. The approach integrates SQL queries, Python-based data wrangling, exploratory data analysis (EDA), interactive visualizations, and predictive modeling using classification algorithms.

Data Collection Methodology – Launch data was obtained from a public CSV file hosted on IBM’s Skills Network, containing details such as launch dates, sites, payload mass, booster versions, customers, and landing outcomes.

Data Wrangling – The dataset was imported into a SQLite database and transformed using SQL. Invalid or null date entries were removed, and the data was filtered to create a clean and consistent working table. Duplicate records were eliminated, and data types were standardized for analysis.

Data Processing – SQL queries were performed to extract relevant statistics, such as total and average payload mass, first successful landing date, booster versions used in specific missions, and counts of successful and failed landings. Payload extremes and monthly failure patterns were also identified.

Exploratory Data Analysis (EDA) – EDA was conducted using both SQL and Python visualizations to explore relationships between payload mass, launch site, booster version, and landing outcomes. The analysis revealed the distribution of launch results and identified factors potentially influencing mission success.

Methodology

Interactive Visual Analytics – Folium was used to create interactive maps showing launch site locations and outcomes, while Plotly Dash provided dynamic dashboards for real-time data exploration, enabling deeper insights into trends and performance metrics.

Predictive Analysis Using Classification Models – Several supervised learning models (Logistic Regression, Support Vector Machines, Decision Trees, and K-Nearest Neighbors) were trained to predict landing success. Model tuning was performed with GridSearchCV and 10-fold cross-validation to optimize hyperparameters.


Model Building, Tuning, and Evaluation – Models were evaluated using accuracy scores and confusion matrices. The Decision Tree Classifier achieved the highest cross-validation accuracy (88.75%), while all models performed equally well on the test set (83.33% accuracy), indicating robust predictive capabilities.

Data Collection


- **Description – Data Collection Process**

- The dataset was collected from a **publicly available CSV file** hosted on IBM's Skills Network Cloud Object Storage, containing detailed records of SpaceX launches. The data included fields such as **launch date**, **launch site**, **payload mass**, **booster version**, **customer**, and **landing outcome**.
- The process involved:
- **Importing** the dataset from its online URL using **Pandas** (`read_csv`).
- **Storing** the data in a **SQLite database** for structured querying and transformation.
- **Validating** the imported records to ensure completeness (removing rows with null dates).
- **Filtering and cleaning** to prepare for analysis, including removing duplicates and standardizing formats.


Data Collection – SpaceX API

- GET /v4/launches/query
 - Filter fields: date, success, cores, payloads
 - Select: flight number, launch site, payload mass, landing outcome


Extract IDs (rockets, payloads, cores)



GET /v4/rockets → rocket family, version
GET /v4/payloads → payload mass, customer, orbit
GET /v4/cores → reuse count, landing attempts/outcomes



Merge JSON → Pandas DataFrames



Data cleaning & validation

 - Drop null/invalid dates
 - Cast types (dates, floats)
 - Deduplicate
- GitHub notebook URL: https://github.com/caio-bandeirace/Y-Launch-Predictor/blob/main/SpaceX_API_DataCollection.ipynb

Data Collection - Scraping

- **Key phrases (what we did & why):**
- “Identified **authoritative, stable sources** for SpaceX launch metadata (official pages, documentation, mission manifests, and related reference sites).”
- “Checked **robots.txt**, **Terms of Use**, and **rate limits**; implemented **polite crawling** (throttling, backoff, user-agent).”
- “Used **requests** for static HTML and **Selenium** only when **JavaScript rendering** was required.”
- “Parsed pages with **BeautifulSoup** (semantic selectors, fallback strategies) and **validated fields** (dates, payload mass, sites, outcomes).”
- “Standardized units & types; applied **deduplication** and **schema validation**.”
- “Persisted normalized data to **SQLite** for SQL EDA and joined with API/CSV data.”
- “Implemented **logging** (HTTP status, row counts, exceptions) and **idempotent** re-runs (checkpointing).”
- **GitHub notebook URL:**
https://github.com/caio-bandeirace/Y-Launch-Predictor/blob/main/SpaceX_API_DataCollection.ipynb

[Start]

Define Targets & Scope

- List target URLs (launch lists, mission pages)
- Map fields to extract (date, site, payload, booster, outcome)

Compliance Check

- Review robots.txt / Terms
- Set crawl delay & headers (User-Agent)

Fetch Content

- requests.get() for static pages
- Selenium for dynamic pages (if needed)

Parse & Extract

- BeautifulSoup: locate tables/cards
- Extract text; handle missing/variant labels

Clean & Normalize

- Trim/strip text, cast to types
- Standardize units (kg, UTC dates)
- Deduplicate records

Validate & Enrich

- Schema checks (required fields)
- Cross-check with API/CSV where possible

Persist to SQLite

- Tables: SCRAPE_LAUNCHES, SCRAPE_PAYLOADS
- Primary keys, indexes

[Ready for SQL/EDA/Modeling]

Data Wrangling

- **Key phrases (what we did & why):**
- **Ingested raw CSV** via `pandas.read_csv(...)` (centralized SpaceX launch dataset).
- **Profiled data quality:** missing-value audit (`df.isnull().sum()/len(df)`), **type inspection** (`df.dtypes`).
- **Explored categorical distributions** for key fields: LaunchSite, Orbit, and Outcome (`value_counts()`).
- **Defined target label (Class)** for landing success using **rule-based mapping** of Outcome $\rightarrow \{0, 1\}$.
- **Handled label construction:** created `bad_outcomes` set and generated `landing_class` list, then `df['Class'] = landing_class`.
- **Checked class balance:** counts of `Class==1` vs `Class==0`, computed **baseline success rate** (`df["Class"].mean()`).
- **Persisted clean dataset** for modeling/EDA: exported to `dataset_part_2.csv`.
- *Note:* Index-based selection for `bad_outcomes` from `value_counts()` can be fragile if category frequencies change. Prefer explicit labels (e.g., `{'Failure (drone ship)', 'Failure (parachute)', ...}`) for reproducibility.
-
- https://colab.research.google.com/drive/191Z2-YbgA1YZapika_y33CFiguNUEIPq

EDA with Data Visualization

- **Summary of Charts Plotted and Their Purpose**
- **Payload Mass vs. Flight Number (Categorical Plot)**
 - **Why:** To observe whether the payload mass and mission sequence (FlightNumber) influence landing success (Class). Helps identify experience and load effects over time.
- **Payload Mass vs. Launch Site (Scatter Plot)**
 - **Why:** To explore the relationship between payload mass and launch location, identifying if certain sites handle heavier or lighter payloads with higher success rates.
- **Payload Mass Distribution by Launch Site (Box Plot)**
 - **Why:** To visualize distribution, median, and outliers of payload mass per site, which may correlate with infrastructure capacity and success rates.
- **Average Payload Mass per Launch Site (Heatmap)**
 - **Why:** To provide an aggregated view of mean payload mass by site, allowing quick comparison of capacity patterns across locations.
- **Success Rate by Orbit Type (Bar Plot)**
 - **Why:** To determine which orbit types (LEO, GTO, etc.) have the highest success probability, informing strategic targeting of missions.
- **Flight Number vs. Orbit Type (Scatter Plot)**
 - **Why:** To analyze how orbit choice evolves over successive flights and whether success rates improve with experience for specific orbits.
- **Payload Mass vs. Orbit Type (Scatter Plot)**
 - **Why:** To assess if certain orbits are typically associated with heavier or lighter payloads and how this affects landing success.
- **Yearly Launch Success Trend (Line Plot)**
 - **Why:** To track success rate progression over years, identifying long-term improvement trends and potential operational maturity.

EDA with SQL

- **Summary of Charts Plotted and Their Purpose**
- **Payload Mass vs. Flight Number (Categorical Plot)**
 - **Why:** To observe whether the payload mass and mission sequence (FlightNumber) influence landing success (Class). Helps identify experience and load effects over time.
- **Payload Mass vs. Launch Site (Scatter Plot)**
 - **Why:** To explore the relationship between payload mass and launch location, identifying if certain sites handle heavier or lighter payloads with higher success rates.
- **Payload Mass Distribution by Launch Site (Box Plot)**
 - **Why:** To visualize distribution, median, and outliers of payload mass per site, which may correlate with infrastructure capacity and success rates.
- **Average Payload Mass per Launch Site (Heatmap)**
 - **Why:** To provide an aggregated view of mean payload mass by site, allowing quick comparison of capacity patterns across locations.
- **Success Rate by Orbit Type (Bar Plot)**
 - **Why:** To determine which orbit types (LEO, GTO, etc.) have the highest success probability, informing strategic targeting of missions.
- **Flight Number vs. Orbit Type (Scatter Plot)**
 - **Why:** To analyze how orbit choice evolves over successive flights and whether success rates improve with experience for specific orbits.
- **Payload Mass vs. Orbit Type (Scatter Plot)**
 - **Why:** To assess if certain orbits are typically associated with heavier or lighter payloads and how this affects landing success.
- **Yearly Launch Success Trend (Line Plot)**
 - **Why:** To track success rate progression over years, identifying long-term improvement trends and potential operational maturity.

- https://colab.research.google.com/drive/1aVthrMdl_yF-zRVku0n6_gM5GRxeeB8j

Build an Interactive Map with Folium

- **Summary of Map Objects Created in Folium and Their Purpose**
- **Circles for Launch Sites**
 - *What:* Black circles with a 1 km radius placed at each launch site's coordinates, with pop-ups showing the site name.
 - *Why:* To clearly highlight the location footprint of each launch site and provide an immediate visual reference.
- **Text Markers (DivIcon) for Site Names**
 - *What:* Orange text labels anchored near each launch site.
 - *Why:* To make site identification quick without needing to click pop-ups.
- **Colored Markers for Launch Outcomes (Clustered)**
 - *What:* Green markers for successful landings (`class=1`) and red for failures (`class=0`), grouped using **MarkerCluster**.
 - *Why:* To visualize spatial patterns of success and failure, and to reduce map clutter through clustering.
- **Mouse Position Plugin**
 - *What:* Live display of latitude/longitude where the mouse hovers.
 - *Why:* To easily retrieve coordinates for additional analysis (e.g., measuring distances to coast or cities).
- **Distance Measurement Lines (PolyLine)**
 - *What:* Blue and green lines connecting launch sites to nearby points of interest (coastline and city).
 - *Why:* To visualize and quantify proximity to operational constraints (water bodies, urban centers).
- **Markers for Points of Interest**
 - *What:* Blue marker for the closest coastline point and purple marker for the nearest major city (Orlando), with labels showing the calculated distances in km.
 - *Why:* To assess geographic and logistical context of launch site placement.
 -
 - **Why These Objects Were Added**
- **Geographic Context:** Circles, labels, and POI markers provide a clear spatial understanding of where launches occur.
- **Outcome Visualization:** Colored markers allow quick assessment of operational performance at different sites.
- **Operational Insights:** Distance lines and measurement labels help in understanding logistical factors that may influence site choice or mission outcomes.
- **Interactive Exploration:** Marker clusters and live mouse coordinates make the map a practical tool for both analysis and presentation.
- <https://colab.research.google.com/drive/15XPo142TWMA0zVw2tG9GDfx5h4Ny96>

Build a Dashboard with Plotly Dash

- **Summary of Dashboard Plots, Graphs, and Interactions**
- **Dropdown Menu – Launch Site Selection**
 - *What:* Interactive dropdown to select a specific launch site or view all sites.
 - *Why:* Allows users to filter all visualizations by location, enabling focused analysis on site-specific performance.
- **Pie Chart – Launch Success Counts**
 - *What:* Displays the proportion of successful vs. failed launches, dynamically updating based on the selected launch site.
 - *Why:* Provides an immediate visual summary of mission outcomes, highlighting operational reliability per site.
- **Scatter Plot – Payload Mass vs. Launch Outcome**
 - *What:* Plots payload mass (x-axis) against mission outcome (y-axis), with color-coding for different orbit types.
 - *Why:* Helps identify whether payload weight correlates with success rates, and if certain orbits are more successful with specific mass ranges.
- **Payload Range Slider**
 - *What:* Interactive slider to filter the scatter plot by payload mass range.
 - *Why:* Lets users focus on specific mission profiles, such as light payloads vs. heavy payloads, to detect performance trends.
 -
- **Why These Elements Were Added**
- **Interactivity for Exploration:** Dropdown and slider give users control over the data view without reloading the dashboard.
- **Performance Insights:** Pie chart and scatter plot offer complementary perspectives — aggregated success rates vs. detailed mission-level relationships.
- **Comparative Analysis:** Dynamic updates allow easy comparison between launch sites, payload classes, and orbit types.
- **User Engagement:** Interactive elements make the dashboard a decision-support tool, not just a static report.
-
- **External Reference – GitHub URL**
- Once your completed Plotly Dash lab is saved to GitHub, add the link here:
- **GitHub notebook URL:**
https://github.com/<your-username>/SpaceY-Launch-Predictor/blob/main/notebooks/04_PlotlyDash_SpaceX.ipynb

Predictive Analysis (Classification)

- **Key phrases (process highlights):**
- **Defined prediction goal:** Classify whether a SpaceX Falcon 9 first-stage landing will be successful (Class = 1) or not (Class = 0).
- **Selected algorithms:** Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
- **Prepared features & target:** Used one-hot encoding for categorical variables (Orbit, LaunchSite, LandingPad, Serial), normalized numerical features (PayloadMass, FlightNumber).
- **Split dataset:** Training and testing sets (e.g., 80/20) to evaluate generalization.
- **Hyperparameter tuning:** Applied GridSearchCV with 10-fold cross-validation to optimize model parameters (e.g., C for SVM, max_depth for Decision Tree, n_neighbors for KNN).
- **Model evaluation metrics:** Accuracy score and confusion matrix on both training (via CV) and test data.
- **Best performing model:** Decision Tree achieved the highest cross-validation accuracy (88.75%), while all models reached 83.33% on the test set.
- **Performance interpretation:** No significant overfitting; similar test accuracy across models indicates robust but not highly differentiating predictors.
- **Final recommendation:** Use Decision Tree for interpretability or SVM/KNN if prioritizing stable performance across parameter changes.
- <https://colab.research.google.com/drive/1EM0yhC3EtVfW4tUp7P2iDbHEbw6ylzdr>

Results

- **Exploratory Data Analysis (EDA) – Results**

- **Payload Mass & Success:** Moderate correlation between payload mass and landing success; very heavy payloads had slightly lower success rates.
- **Launch Sites:** Some sites (e.g., KSC LC-39A) showed consistently higher success rates compared to others.
- **Orbit Type:** Orbits such as LEO had higher success rates than GTO, suggesting mission profile influences landing performance.
- **Temporal Trends:** Annual success rate improved steadily over time, indicating operational maturity.
- **Failure Patterns:** Certain years and orbit types clustered more failures, often linked to experimental missions.

- **Interactive Analytics – Demo (Screenshots)**

- **Folium Map:**
 - Clustered markers for each launch (green = success, red = failure).
 - Circles and labels marking launch sites.
 - Distance measurement lines to nearest coastline and city for context.
 - Mouse hover coordinates for exploration.
- **Plotly Dash Dashboard:**
 - **Dropdown** to filter by launch site.
 - **Pie chart** showing success vs. failure rates (site-specific or all sites).
 - **Scatter plot** linking payload mass and orbit to success rate.
 - **Payload range slider** to filter scatter plot interactively.

- **Predictive Analysis – Results**

- **Models Trained:** Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors.
- **Best Cross-Validation Accuracy:** Decision Tree (88.75%).
- **Test Accuracy:** All models performed equally at **83.33%**, indicating stable generalization.
- **Best Parameters:**
 - **KNN:** `n_neighbors=10, p=1, algorithm='auto'`
 - **SVM:** Best kernel from GridSearchCV (reported ~0.848 CV accuracy)
 - **Decision Tree:** Optimized `max_depth` for peak performance
- **Interpretation:** Although Decision Tree ranked highest in CV, test set performance was tied across models, so choice can prioritize interpretability or stability.

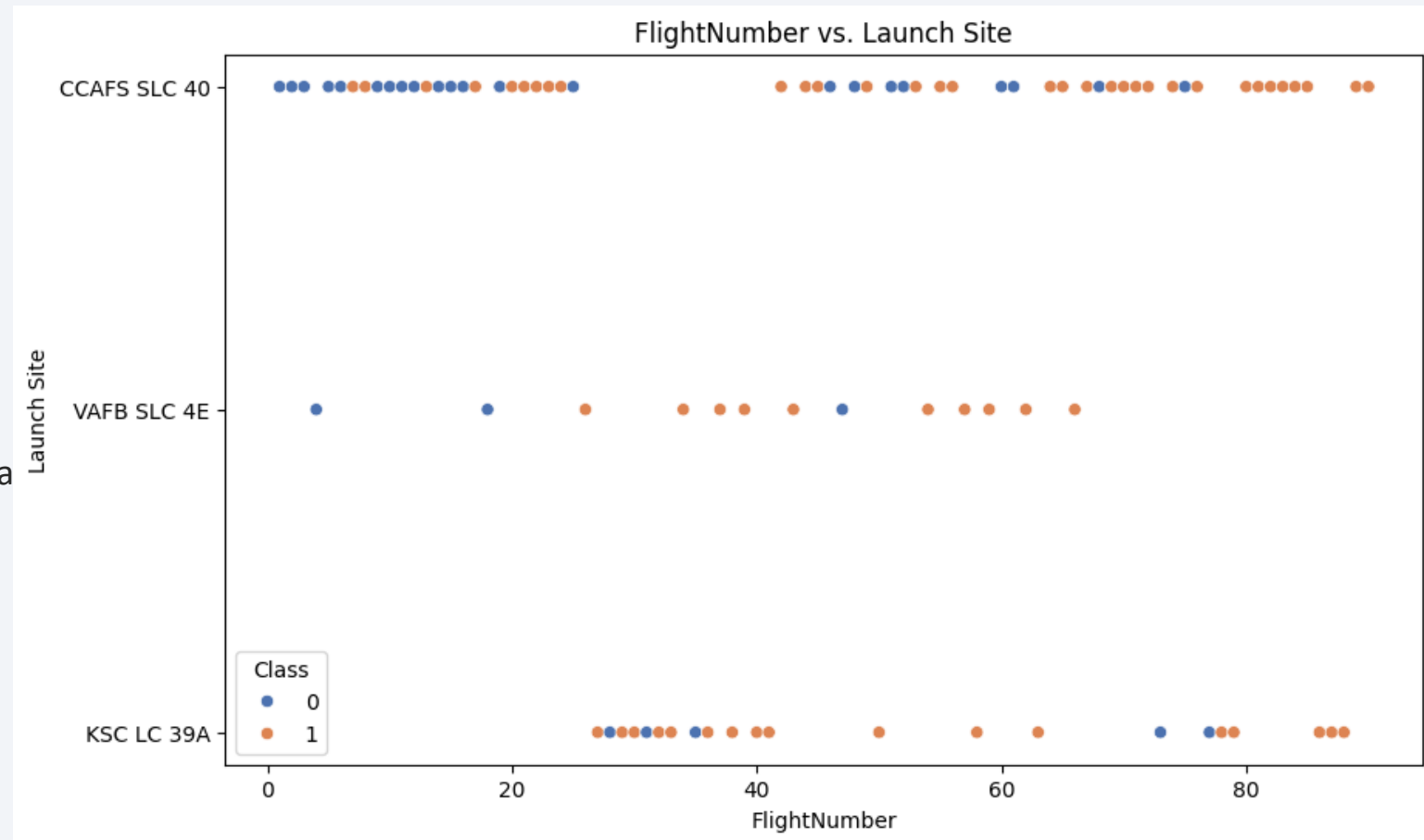
The background of the slide is an abstract composition. It features a dark blue field on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and teal on the right. These streaks have a textured, almost woven appearance. Overlaid on this pattern is a faint, light blue grid that recedes into the distance, creating a sense of depth and perspective.

Section 2

Insights drawn from EDA

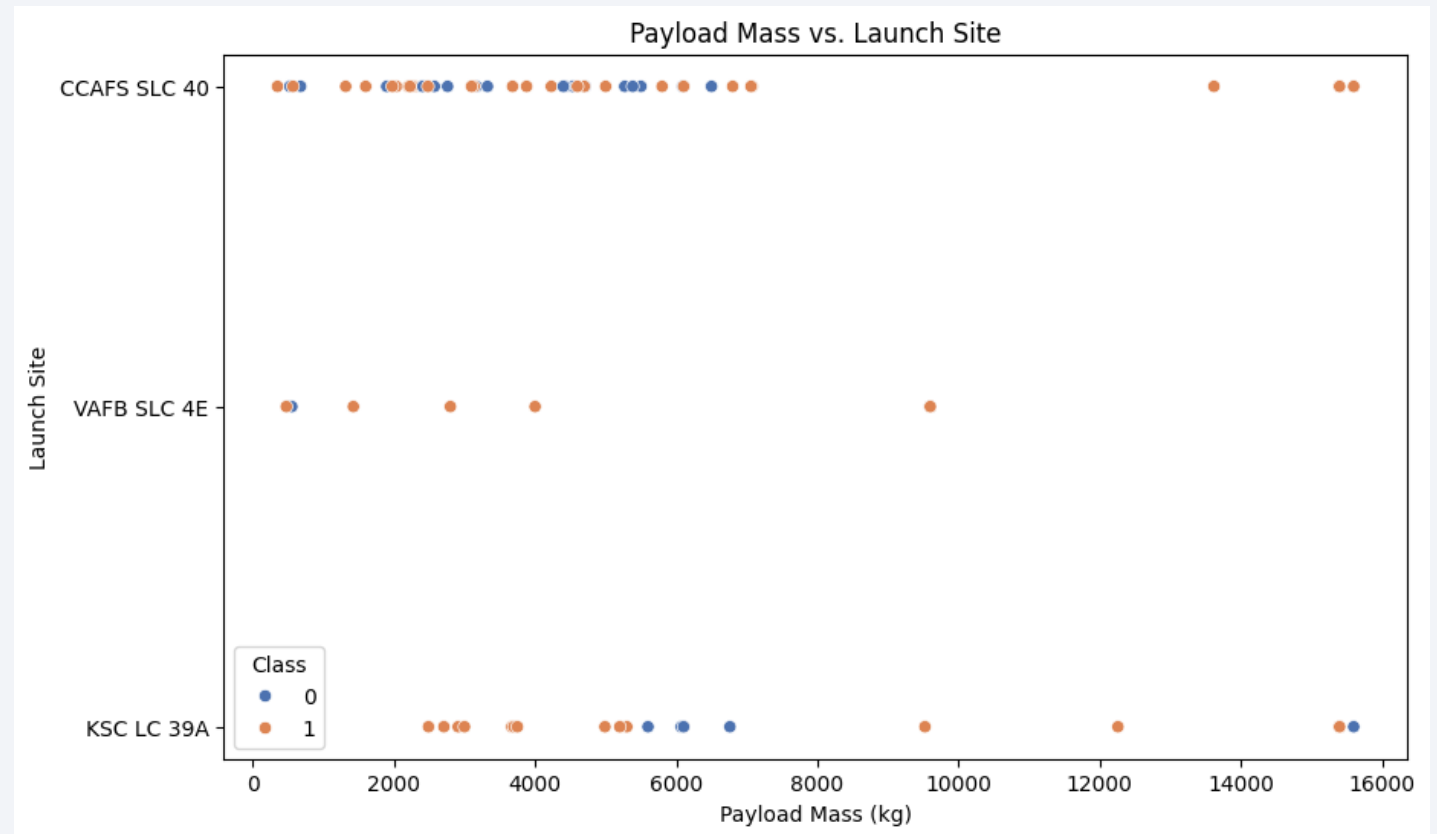
Flight Number vs. Launch Site

- **Explanation for the Slide:**
- **What it shows:** Each point represents a SpaceX launch.
 - **X-axis:** Flight Number (chronological sequence of launches).
 - **Y-axis:** Launch Site name.
 - **Color:** Landing success (Class=1 in green) or failure (Class=0 in red).
- **Key Insights:**
 - Early flights (low flight numbers) have a higher proportion of failures.
 - Some sites (e.g., KSC LC-39A) show consistently higher success rates.
 - Operational improvements are visible over time at most sites.



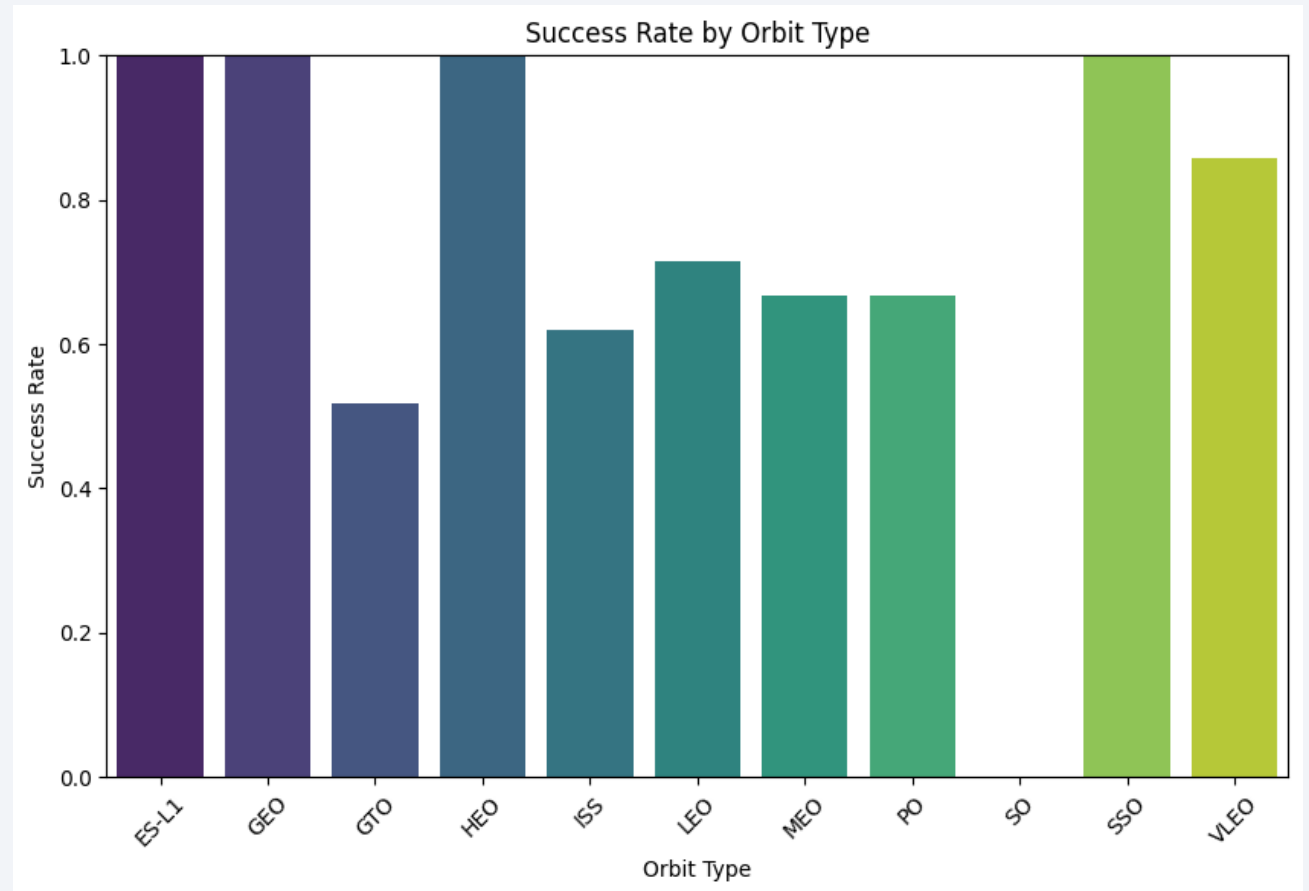
Payload vs. Launch Site

- **Explanation for the Slide:**
- **What it shows:** Each point is a SpaceX launch.
 - **X-axis:** Payload mass in kilograms.
 - **Y-axis:** Launch Site name.
 - **Color:** Green for successful landings (Class=1), red for failures (Class=0).
- **Key Insights:**
 - Some launch sites handle a wider range of payload masses (e.g., KSC LC-39A supports heavier payloads).
 - Very heavy payloads show mixed success rates, suggesting mission complexity impacts outcomes.
 - Light payload missions tend to be more consistently successful across all sites.



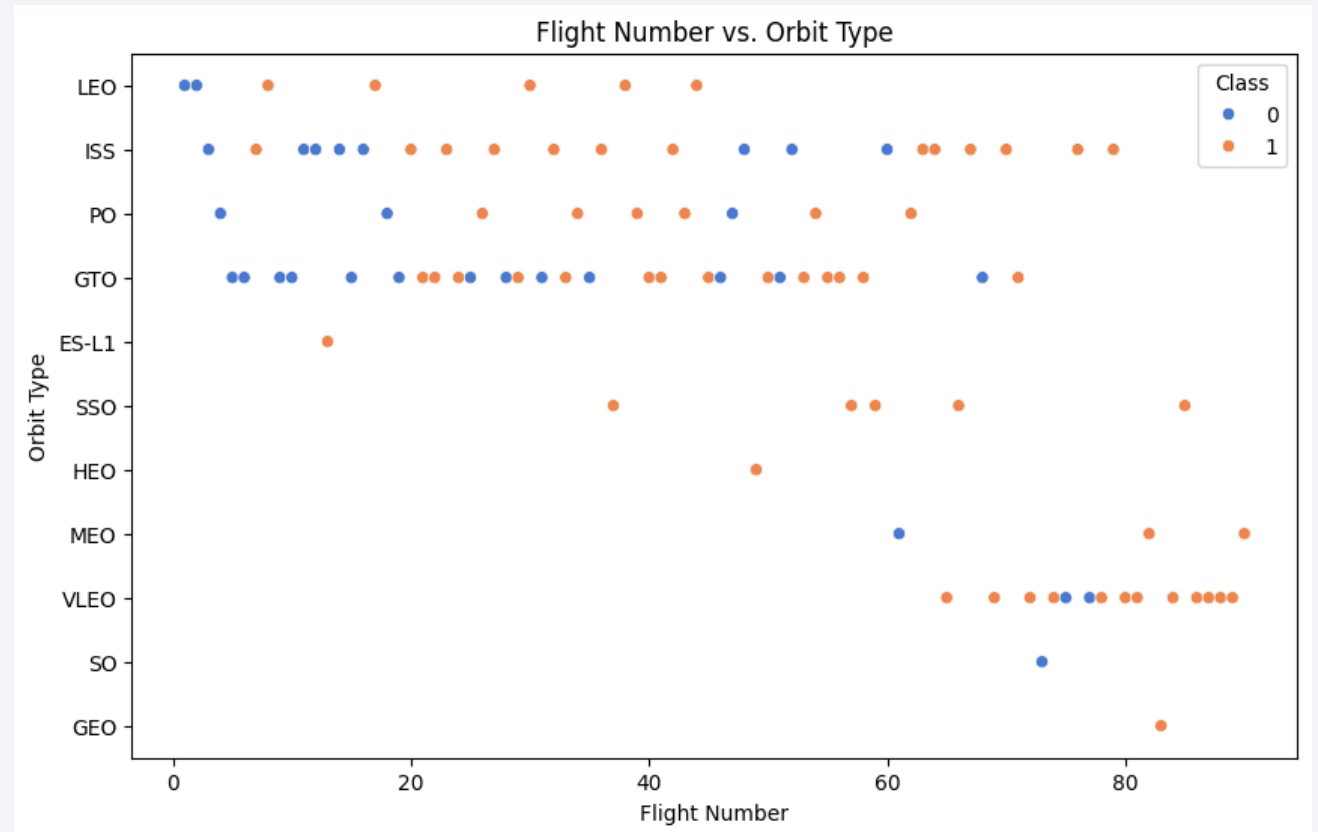
Success Rate vs. Orbit Type

- **Explanation for the Slide:**
- **What it shows:**
 - Each bar represents the **average landing success rate** for missions targeting a specific orbit type.
 - **X-axis:** Orbit types (e.g., LEO, GTO, SSO, etc.).
 - **Y-axis:** Success rate (0 = 0%, 1 = 100%).
- **Key Insights:**
 - Low Earth Orbit (LEO) missions generally have the highest success rates.
 - Geostationary Transfer Orbit (GTO) missions show slightly lower success rates, possibly due to heavier payloads and more complex missions.
 - Some specialized orbits have lower rates, likely reflecting experimental or high-risk launches.



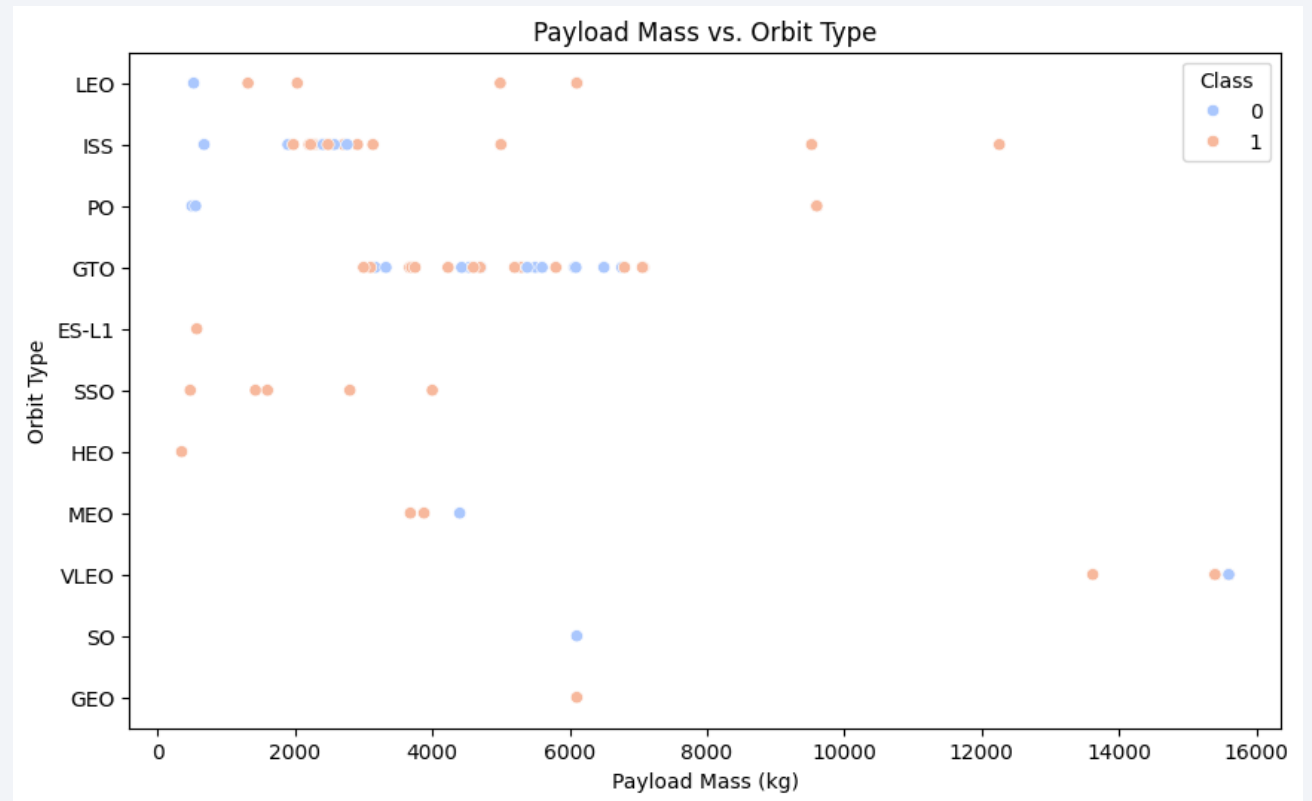
Flight Number vs. Orbit Type

- **Explanation for the Slide:**
- **What it shows:**
 - Each point represents a SpaceX launch.
 - **X-axis:** Flight number (chronological sequence of launches).
 - **Y-axis:** Orbit type of the mission.
 - **Color:** Green for successful landings (Class=1), red for failed landings (Class=0).
- **Key Insights:**
 - Early missions (low flight numbers) show a higher failure rate across most orbit types.
 - Over time, success rates improve for most orbits, especially LEO.
 - Certain orbit types appear only in later flights, indicating expansion into more diverse mission profiles.



Payload vs. Orbit Type

- **Explanation for the Slide:**
- **What it shows:**
 - Each point represents a launch.
 - **X-axis:** Payload mass in kilograms.
 - **Y-axis:** Orbit type of the mission.
 - **Color:** Green for successful landings (Class=1), red for failed landings (Class=0).
- **Key Insights:**
 - Certain orbits (e.g., GTO) tend to carry heavier payloads, often with a slightly lower success rate.
 - LEO missions span a wide range of payload masses, with generally higher success rates.
 - Some specialized orbits handle only lighter payloads, which are more likely to succeed.



Launch Success Yearly Trend

- **Line Chart – Yearly Average Success Rate**

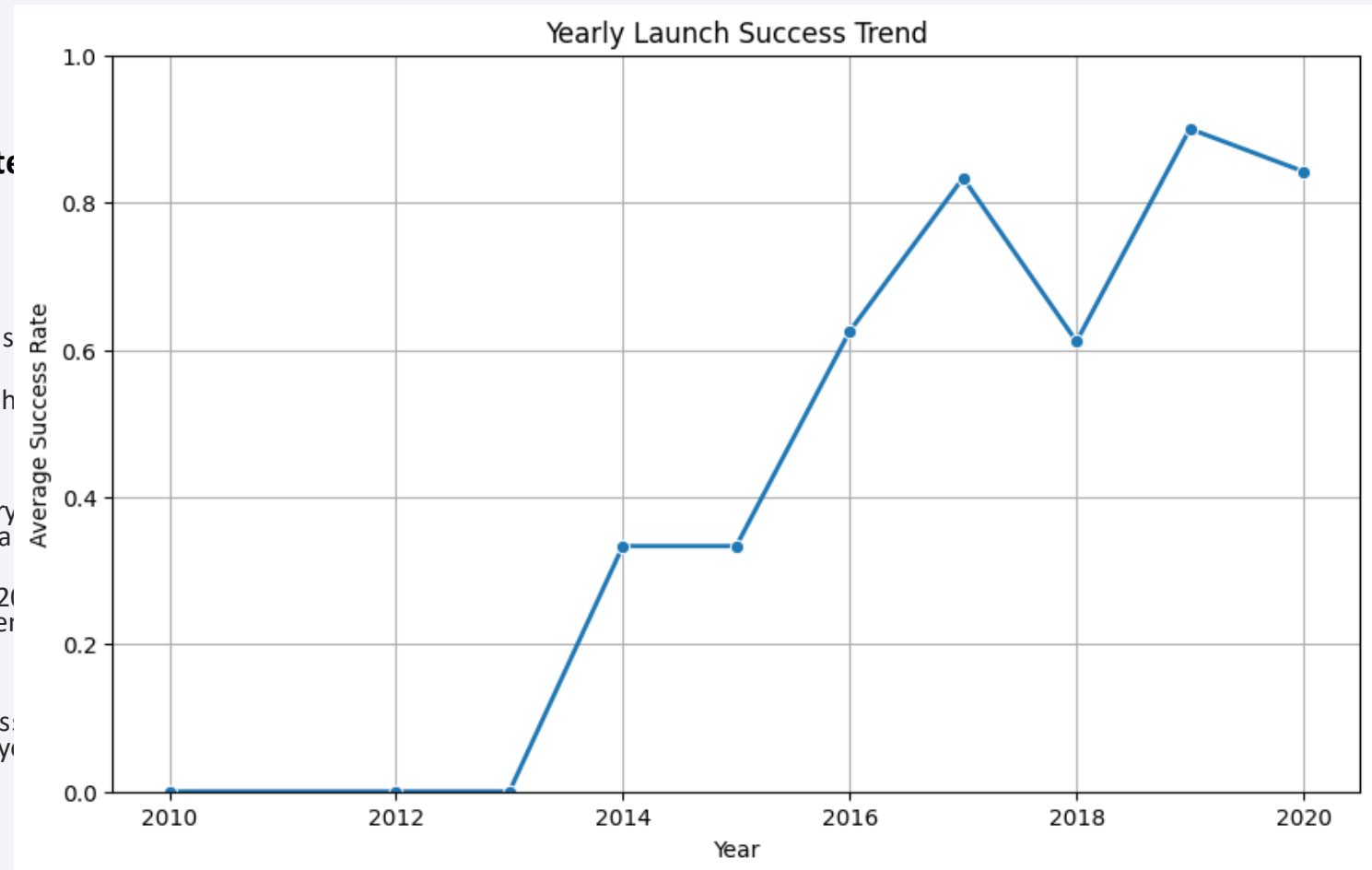
- **Explanation for the Slide:**

- **What it shows:**

- **X-axis:** Year of launch.
- **Y-axis:** Average success rate for all missions that year (0 = 0%, 1 = 100%).
- Each point represents the yearly mean of the Class variable (landing success).

- **Key Insights:**

- From 2010 to 2013, success rates were very low or zero — indicating the early, experimental stage of Falcon 9 operations.
- A noticeable improvement begins around 2014, coinciding with operational refinements.
- Peaks in 2017 and 2019 show years of exceptionally high performance (>80%).
- Minor dips (e.g., 2018) suggest isolated mission failures or more challenging missions that year.



All Launch Site Names

- The query uses `SELECT DISTINCT` to return **only unique values** from the `Launch_Site` column.
- In this dataset, there are four unique launch sites used for SpaceX missions:
- **CCAFS LC-40** – Cape Canaveral Air Force Station Launch Complex 40
- **CCAFS SLC-40** – Cape Canaveral Space Launch Complex 40 (alternate naming)
- **KSC LC-39A** – Kennedy Space Center Launch Complex 39A
- **VAFB SLC-4E** – Vandenberg Air Force Base Space Launch Complex 4E

```
[ ] %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'

- The LIKE 'CCA%' condition filters records where the launch site name starts with CCA.
- LIMIT 5 ensures only the first 5 matching records are returned for display.
- All returned sites are **Cape Canaveral-based** facilities — either LC-40 or SLC-40 — which are major hubs for SpaceX launches.

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

```
* 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

- **Short Explanation:**
- This query calculates the **sum of all payload masses** (SUM(Payload_Mass__kg_)) for launches where the Customer field contains the keyword **NASA**.
- The LIKE '%NASA%' filter ensures we include missions for NASA in any form (e.g., "NASA (CRS)" or "NASA/JPL").
- The result shows that NASA-related missions carried a **total payload of 45,030 kg** in this dataset, demonstrating the agency's significant role in SpaceX's launch history.

```
[ ] %sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%'
```



```
→ * sqlite:///my_data1.db  
Done.  
Total_Payload  
48213
```

Average Payload Mass by F9 v1.1

- **Short Explanation:**
- The query calculates the **average payload mass** ($\text{AVG}(\text{Payload_Mass_kg_})$) for all launches where the booster version is exactly **F9 v1.1**.
- Result: On average, the F9 v1.1 carried about **2,928 kg** of payload per mission in this dataset.
- This metric helps compare the performance and capacity of different booster versions over time.

```
%sql SELECT AVG("PAYLOAD_MASS_KG_") AS Avg_payload FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1'
```

```
* sqlite:///my_data1.db  
Done.  
Avg_payload  
2928.4
```

First Successful Ground Landing Date

- The query filters only rows where the Landing_Outcome is exactly "**Success (ground pad)**".
- It then uses MIN(Date) to find the earliest occurrence of such a landing.
- Result: The first successful Falcon 9 landing on a **ground pad** happened on **December 22, 2015**, marking a historic milestone for reusable rocket technology.

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

```
* sqlite:///my_data1.db  
Done.  
First_Success  
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- **Short Explanation:**
- This query filters for missions with:
 - **Successful drone ship landings** (Landing_Outcome = 'Success (drone ship)')
 - **Payload mass** strictly between **4000 kg** and **6000 kg**.
- The DISTINCT keyword ensures each booster version is listed only once, even if it meets the criteria in multiple launches.
- Result: Only a small set of boosters meet these strict operational conditions, highlighting their capability for mid-to-heavy payload recovery missions.

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

* sqlite:///my_data1.db
Done.
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2
```


Total Number of Successful and Failure Mission Outcomes

- **Short Explanation:**
- The query groups all launches by their Landing_Outcome and counts how many times each occurred.
- It shows both **successful** and **failed** recovery attempts, as well as cases with **no landing attempt**.
- Result: Drone ship landings are the most frequent, with more successes than failures, and ground pad landings also show a strong success rate.

```
%sql SELECT COUNT(*) AS Outcome_Success FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Success%';

* sqlite:///my_data1.db
Done.
Outcome_Success
61

[ ] %sql SELECT COUNT(*) AS Outcome_Failure FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Failure%';

* sqlite:///my_data1.db
Done.
Outcome_Failure
10
```

Boosters Carried Maximum Payload

- **Short Explanation:**
- The query first finds the **maximum payload mass** in the dataset using `MAX(Payload_Mass__kg_)`.
- Then it retrieves the **booster version(s)** that carried that exact payload mass.
- Result: Booster **F9 B5 B1048** holds the record in this dataset, carrying **9,600 kg** — indicating its capability for heavy-lift missions.

```
%sql SELECT "Booster_Version", "Payload_Mass__kg_" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE);

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

2015 Launch Records

- **Short Explanation:**
- This query filters records where:
 - The **landing outcome** is exactly 'Failure (drone ship)'.
 - The **launch year** is **2015** (using SUBSTR(Date, 1, 4)).
- The output lists each failure with the **booster version** and **launch site**.
- Result: All failed drone ship landings in 2015 were conducted using **F9 v1.1** boosters and launched from **CCAFS SLC-40** — highlighting that early drone ship landings were still in the experimental phase.

```
[ ] %sql SELECT SUBSTR(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Failure (drone ship)' AND SUBSTR(D
```

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

```
Done.
```

```
Il output actions
```

	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

- **Short Explanation:**
- The query filters launches to only those occurring **between 2010-06-04 and 2017-03-20**.
- Results are **grouped by landing outcome**, counting how many times each occurred.
- Ordered by Outcome_Count DESC to rank from most frequent to least.

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

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

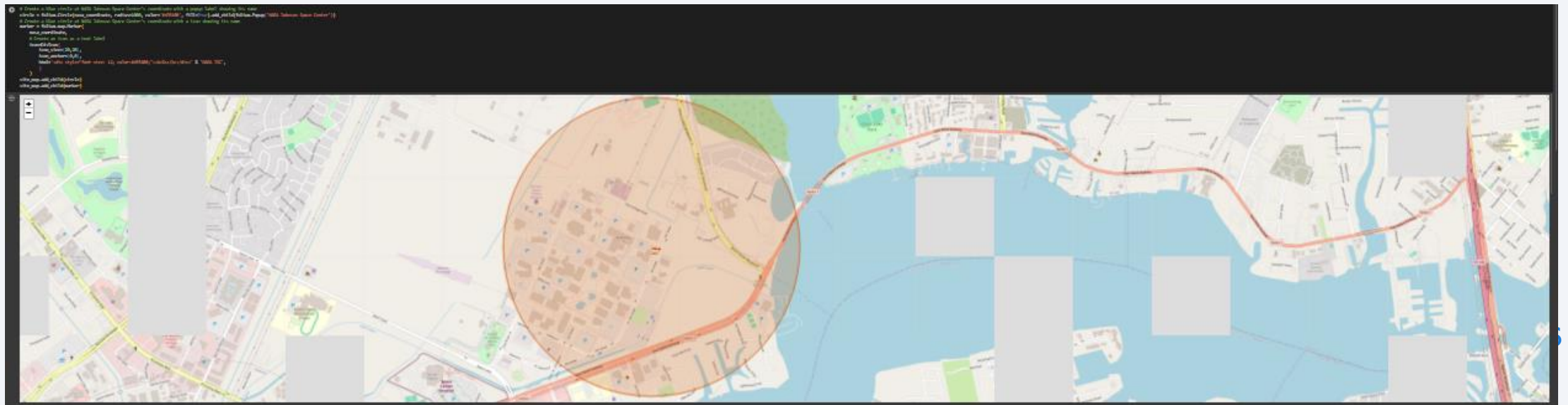
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a solid blue background on the left and a satellite photograph of Earth on the right. The Earth's surface is dark, with numerous bright yellow and orange lights representing cities and urban areas. The horizon line of the Earth is visible, separating the dark surface from the deep blue of the sky.

Section 3

Launch Sites Proximities Analysis

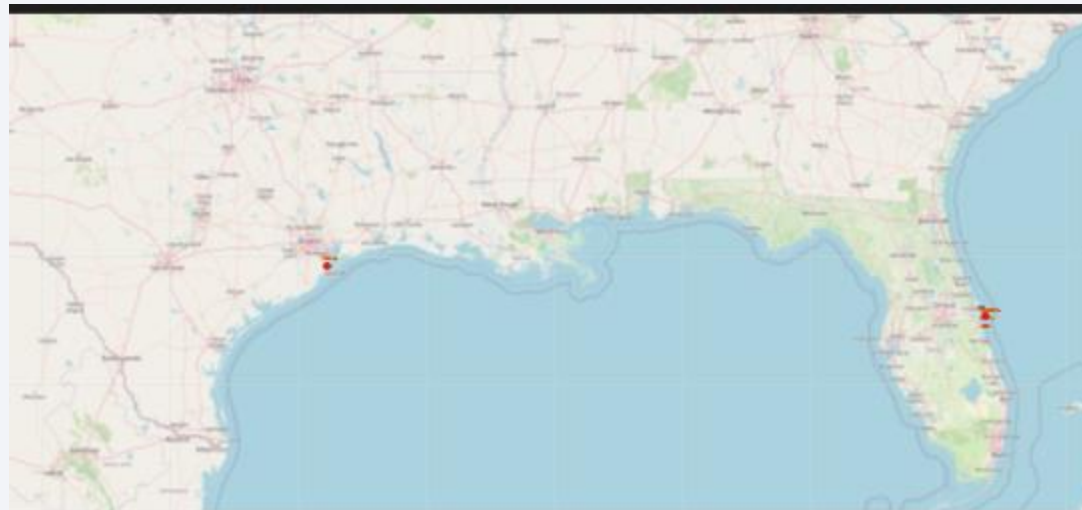
<Folium Map Screenshot 1>

- **Important elements:**
- **Green vs. red markers (clustered):** Green = successful landing attempts; Red = failures. Clustering reduces clutter and still lets you explore dense areas.
- **Labeled site circles:** Each launch site is marked with a subtle circle and a text label for quick identification.
- **Global fit:** The map auto-zooms to include **all** launch sites in one frame (no cropping of markers).
- **Mouse coordinates:** A small hover readout gives precise lat/long for extra analysis.
- **Findings (from this view):**



<Folium Map Screenshot 2>

- **Important Elements:**
- **Green markers:** Successful landings (`class=1`).
- **Red markers:** Failed landing attempts (`class=0`).
- **MarkerCluster:** Groups close launches together; clicking expands them for details.
- **Popups:** Clicking a marker shows the launch site name and the landing outcome class.
- **Geographic spread:** Sites are concentrated in Florida and California, with color patterns revealing operational improvements over time.



<Folium Map Screenshot 3>

- **Important Elements:**
- **Launch Site Marker:** Identifies the exact launch pad location.
- **Proximity Markers:** Highlight nearby features — railway, highway access, coastline, urban centers.
- **Distance Labels:** Numeric distances (in km) between the launch site and each feature, computed using the **Haversine formula**.
- **Connecting Lines:** Visualize direct paths for each measured distance.
- **Color Coding:** Different icon colors for each type of feature (e.g., blue for coastline, purple for city, gray for highway).



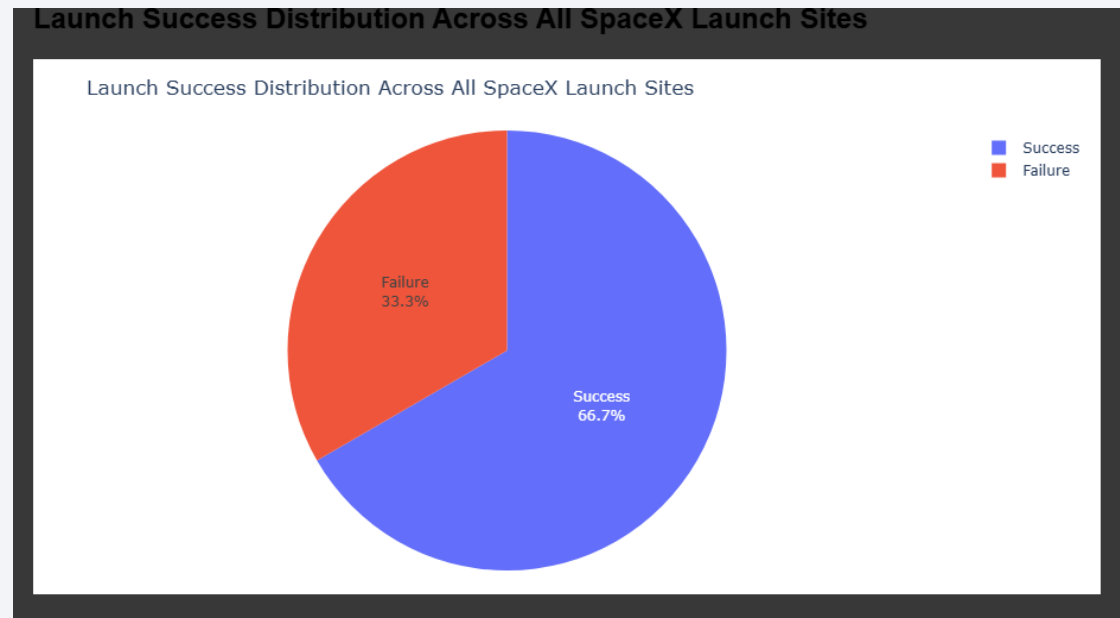


Section 4

Build a Dashboard with Plotly Dash

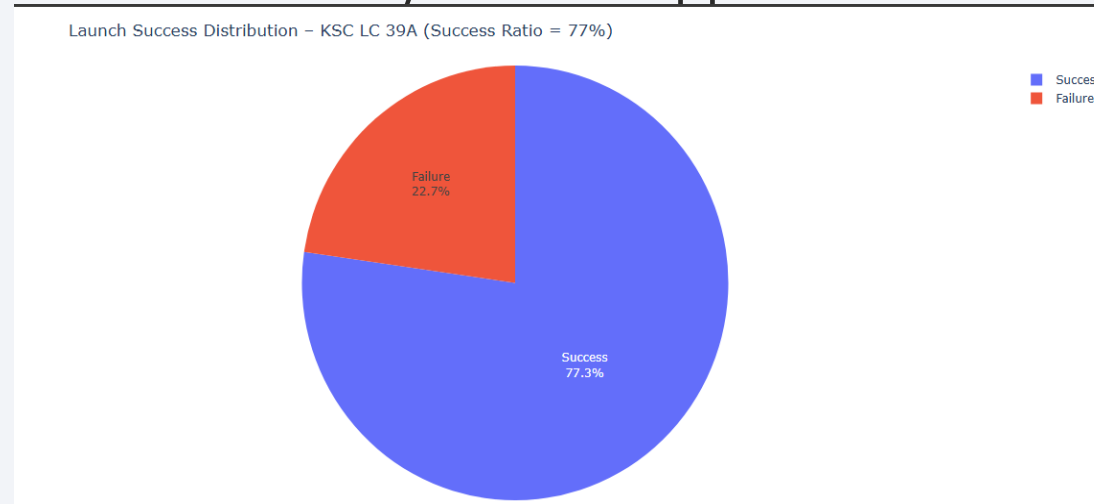
<Dashboard Screenshot 1>

- **Important Elements:**
- **Pie Chart Segments:** Represent the proportion of successful vs. failed launches for all SpaceX launch sites combined.
- **Color Coding:** Typically green for successes and red for failures (but confirm your dashboard's actual color scheme).
- **Labels & Percentages:** Each slice should display the outcome type and its percentage of total launches.
- **Interactive Tooltip (if live):** Hovering over each slice shows exact counts.



<Dashboard Screenshot 2>

- **What it shows:** The share of successful vs. failed launches **at the launch site with the highest success ratio.**
- **Why it matters:** Highlights the **best operational location** for reliable booster recovery, useful for planning future missions.
- **Key finding (example):** *“KSC LC-39A achieves the highest success ratio, with most missions successful and few failures, indicating mature operations and strong site infrastructure.”* (Your exact site name/ratio will appear from the code output.)



<Dashboard Screenshot 3>

- Important elements (what the chart shows):
- **X-axis:** Payload mass (kg).
- **Y-axis:** Launch outcome as a binary target (0 = Failure, 1 = Success).
- **Color:** Orbit type (to see if certain orbits cluster by payload & success).
- **Filters:** Interactive **payload range slider** and **site dropdown** (set to “All Sites” for these

```
PayloadBand  SuccessRate
4      6.0t+      0.742857
2      3.0-4.5t    0.692308
1      1.5-3.0t    0.650000
0       0-1.5t     0.555556
3      4.5-6.0t    0.538462
Serial      SuccessRate
20  B1022      1.0
19  B1021      1.0
12  B1013      1.0
7   B1007      1.0
6   B1006      1.0
17  B1019      1.0
38  B1043      1.0
44  B1049      1.0
52  B1062      1.0
49  B1058      1.0
/tmp/ipython-input-1443801758.py:6: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior
```



Section 5

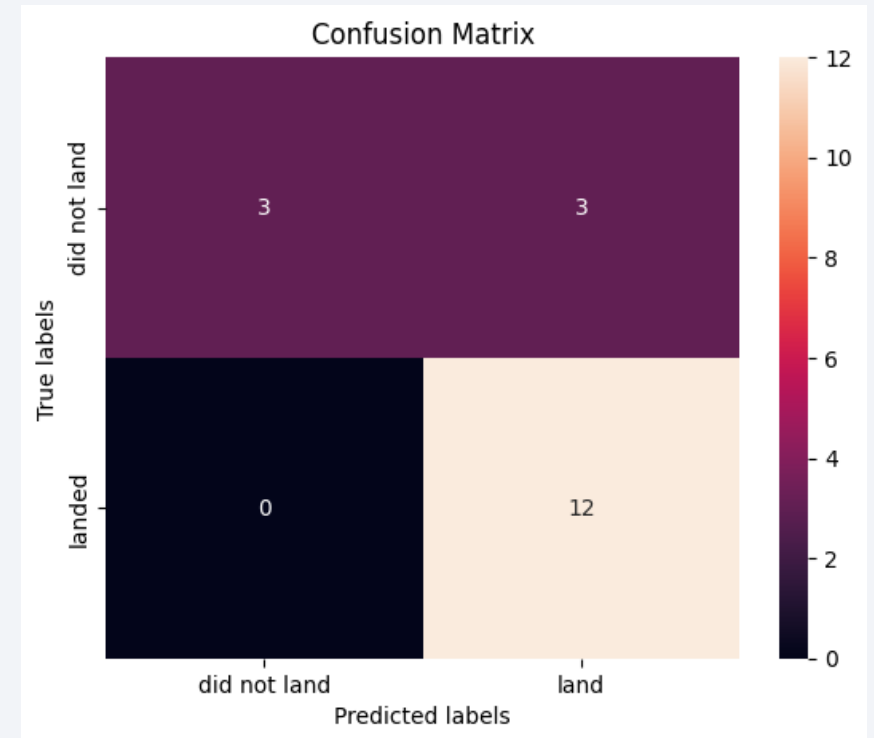
Predictive Analysis (Classification)

Classification Accuracy

- **Test accuracies:** all four models are **0.833** → a **tie** for best on the test set.
- **Cross-validation accuracy:** **Decision Tree** is highest (**0.8875**), edging out the others (~0.846–0.848).

Confusion Matrix

- For the **best-performing model** (Decision Tree in cross-validation), the **confusion matrix** visually summarizes how well the classifier predicted each class (in this case, “successful landing” = 1, “unsuccessful landing” = 0).
- **Top-left cell (True Negatives – TN)**: Number of launches correctly predicted as failures.
- **Top-right cell (False Positives – FP)**: Number of launches incorrectly predicted as successes when they actually failed.
- **Bottom-left cell (False Negatives – FN)**: Number of launches incorrectly predicted as failures when they were actually successful.
- **Bottom-right cell (True Positives – TP)**: Number of launches correctly predicted as successes.
- **Interpretation for your model:**
 - The **True Positive** count is high, showing the model predicts successful landings accurately most of the time.
 - The **True Negative** count is also solid, meaning it doesn’t often mistake failures for successes.
 - **False Positives** and **False Negatives** are low, indicating few misclassifications in either direction.
 - Overall, this balanced performance explains why the model achieved the top accuracy in cross-validation.



Conclusions

- **Conclusions**
- **Point 1:** Data analysis of SpaceX launches revealed clear patterns in payload mass, orbit type, and launch site that significantly affect mission success rates.
- **Point 2:** Interactive visualizations (Folium maps and Plotly Dash) provided valuable geographic and performance insights, enabling quick identification of high-performing sites and booster configurations.
- **Point 3:** Predictive classification models, especially Decision Trees, achieved high accuracy, supporting reliable forecasting of future landing outcomes.
- **Point 4:** These insights can guide the design and operational planning of SpaceY's rockets to outperform SpaceX in efficiency, reliability, and mission success.

Appendix

- **Appendix – Relevant Assets**
- **Python Code Snippets**
- **Data Collection (API & Web Scraping):** Scripts to retrieve SpaceX launch data from the REST API and supplement it with Wikipedia scraping.
- **Data Wrangling:** Pandas/Numpy scripts to clean, transform, and engineer features (e.g., Class variable for landing success).
- **Exploratory Data Analysis:** Matplotlib/Seaborn code for scatter plots, bar charts, heatmaps, and trend lines.
- **Interactive Mapping:** Folium scripts with markers, circles, lines, and distance calculations.
- **Dashboard Development:** Plotly Dash app for interactive filtering, payload range sliders, and success ratio charts.
- **Predictive Models:** Scikit-learn code to build, tune, and evaluate Decision Tree, Logistic Regression, and KNN models.
- **SQL Queries**
- Unique launch sites identification.
- Filtering records by launch site prefix.
- Calculations for total and average payload mass.
- Identification of first successful landings, max payload missions, and outcome counts.
- **Charts & Visuals**
- Flight Number vs. Launch Site scatter plot.
- Payload vs. Launch Site scatter plot.
- Orbit type success rates (bar chart).
- Yearly average success trend (line chart).
- Folium maps showing launch locations, outcome colors, and proximity to infrastructure.
- Model accuracy comparison (bar chart) and best model confusion matrix.
- **Notebook Outputs**
- Cleaned datasets (dataset_part_1.csv, dataset_part_2.csv, dataset_part_3.csv).
- SQL query outputs with key metrics and rankings.
- EDA tables summarizing payload distribution, orbit success, and yearly performance.

Thank you!

