IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohammed Babiker
January 21, 2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

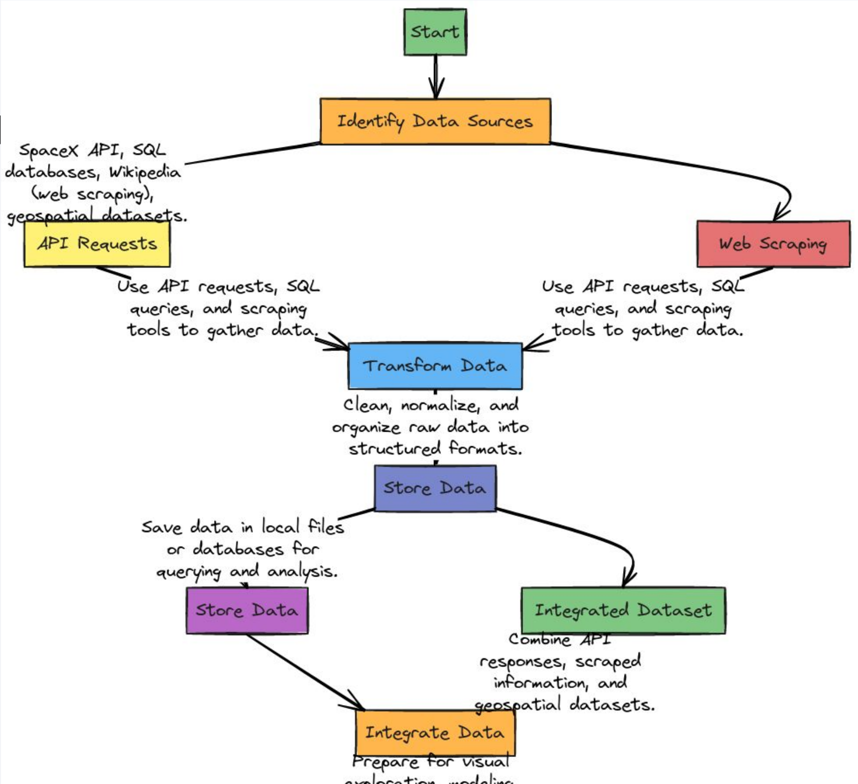
Executive Summary

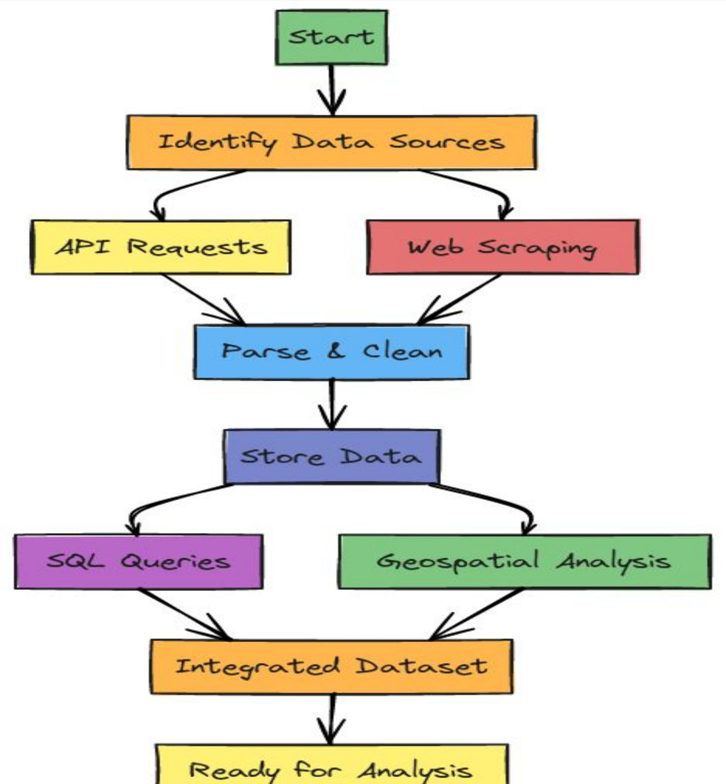
- Data collection methodology:
    - **Data for all stages was collected using multiple methods, including API requests, SQL queries, and web scraping. SpaceX's API provided detailed launch information, such as booster versions, payload details, and landing outcomes, while SQL databases enabled structured queries for aggregation and filtering. Additionally, geospatial data and interactive visual tools like Folium and Plotly Dash were used to integrate geographical and temporal patterns into the analysis**
- Perform data wrangling
    - The collected data was cleaned by removing irrelevant or missing values, standardizing formats, and filtering for Falcon 9 launches. Lists and JSON data from the API were transformed into structured DataFrames, and categorical variables were encoded to prepare the data for analysis and machine learning.
- Perform exploratory data analysis (EDA) using visualization and SQL
    - EDA was performed by visualizing key relationships between variables, such as payload mass, orbit type, and success rates, using libraries like Matplotlib and Seaborn. SQL queries were used to extract insights from the dataset, including launch site statistics, payload trends, and success rates for different mission outcomes, enabling deeper analysis of patterns in the data.
- Perform interactive visual analytics using Folium and Plotly Dash
    - Interactive visual analytics were conducted by creating dynamic maps with Folium to explore launch site proximities and success patterns, and using Plotly Dash to build interactive dashboards for analyzing launch data trends and performance metrics.
- Perform predictive analysis using classification models
    - Predictive analysis was conducted by building, tuning, and evaluating classification models such as Logistic Regression, Support Vector Machines, Decision Trees, and K-Nearest Neighbors using GridSearchCV for hyperparameter optimization. The models were assessed for accuracy and performance using confusion matrices and test data to identify the best-performing approach

Data Collection

- Describe how data sets were collected.
  - Data for this project was gathered from a combination of sources and methods to ensure comprehensive and accurate insights into SpaceX Falcon 9 launches. The primary dataset was collected using SpaceX's API, which provided detailed information about launch history, booster versions, payloads, orbit types, and landing outcomes. Web scraping was employed to extract supplementary data from publicly available sources, such as the Wikipedia page detailing Falcon 9 and Falcon Heavy launches. This allowed for the inclusion of historical and contextual information that was not directly available through the API.
  - In addition, SQL databases were utilized to store and query the data for structured exploration. Queries allowed for filtering, aggregating, and extracting specific data points such as payload trends, launch site usage, and mission outcomes. Geospatial data, including launch site coordinates, was also included to analyze geographical factors and their impact on mission success. These coordinates were combined with Folium and Plotly Dash to create dynamic visualizations, enabling deeper exploration of spatial and temporal patterns. Together, these diverse methods ensured the data was both robust and ready for advanced analysis.
  -

- You need to present your d charts

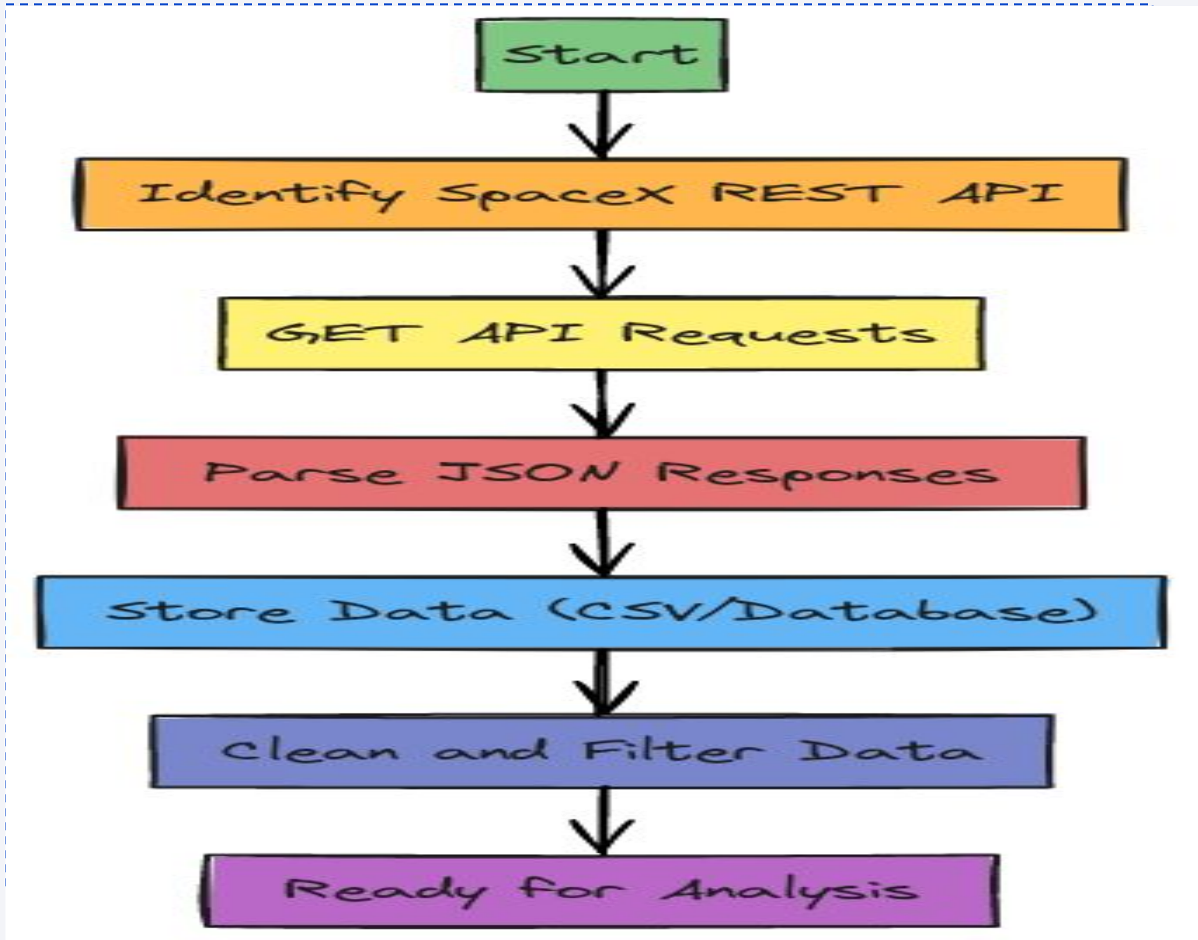




Data Collection – SpaceX API

- **Identify Source**: Utilize the **SpaceX REST API** as the main data source for rocket launches and associated information.
- **API Request**: Perform **GET requests** to retrieve data on booster versions, payload details, launch sites, and landing outcomes.
- **Parse and Store**: Extract and parse the **JSON responses**, saving the structured data into **Pandas DataFrames** or **CSV files** for further analysis.
- **Data Wrangling**: Apply data cleaning and filtering techniques to focus on **Falcon 9 launches** and specific metrics such as **launch sites**, **payload mass**, and **success outcomes**.

- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/jupyter-labs-spacex-data-collection-api%20(3).ipynb



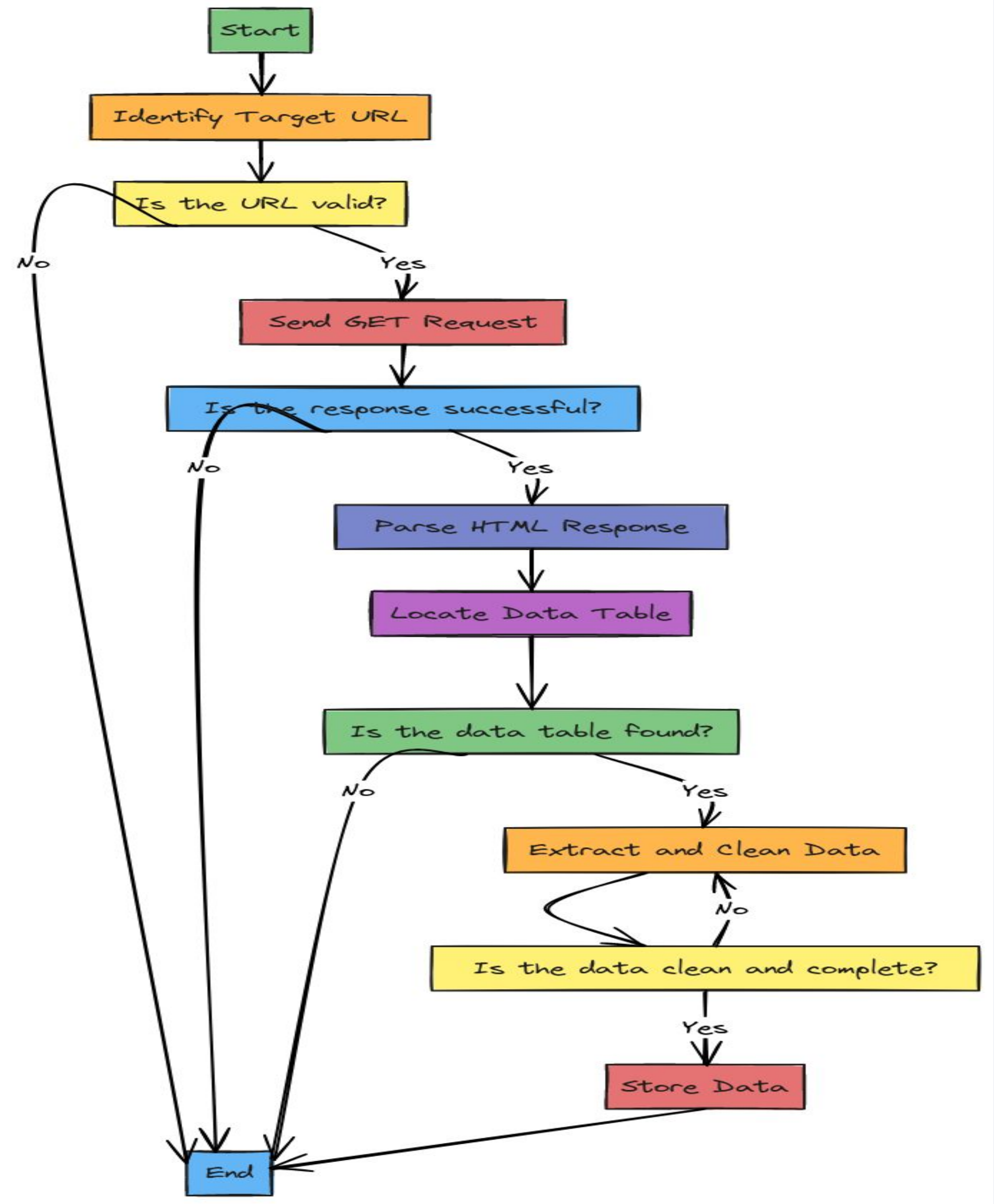
Data Collection - Scraping

Key Phrases for Web Scraping:

1. **Identify Target Page**: Target the SpaceX Wikipedia page containing Falcon 9 launch data.
2. **Send HTTP Request**: Use Python libraries like `requests` to send an HTTP GET request to retrieve the HTML content of the webpage.
3. **Parse HTML Content**: Use the `BeautifulSoup` library to parse the HTML and locate the table containing the relevant data (e.g., launch dates, sites, outcomes, payloads).
4. **Extract Data**: Iterate through the table rows to extract data fields, cleaning or transforming as needed.
5. **Store Data**: Save the extracted data into structured formats like Pandas DataFrames or CSV files for analysis.

- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/jupyter-labs-webscraping

Plac



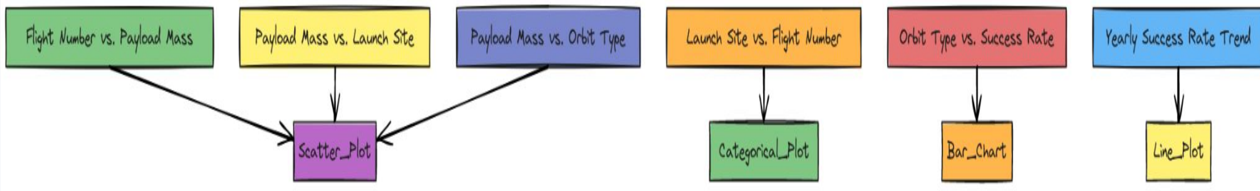
Data Wrangling

- **Load Raw Data**: Import raw datasets from SpaceX API responses, CSV files, and scraped HTML tables into Pandas DataFrames.
- **Clean Missing and Duplicate Values**: Identify and remove rows with missing or irrelevant values; ensure unique records by eliminating duplicates.
- **Standardize Formats**: Convert data types for consistency (e.g., dates, numeric fields) and normalize values like payload mass to a standard unit.
- **Filter Data**: Focus on Falcon 9 launches by selecting relevant rows based on specific criteria, such as launch type or booster versions.
- **Feature Engineering**: Encode categorical variables (e.g., launch site, orbit type) using one-hot encoding and create new features like success rates for analysis.
- **Store Structured Data**: Save the cleaned and processed data into CSV files or Pandas DataFrames for downstream analysis.

- You need to present your data wrangling process using key phrases and flowcharts
- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/labs-jupyter-spacex-Data%20wrangling%20(3).ipynb

EDA with Data Visualization

- **Flight Number vs. Payload Mass (Scatter Plot)**:
  - **Why**: To visualize how payload mass and flight experience (flight number) influence landing success.
  - **Insights**: Shows trends that higher flight numbers correlate with increased success, and payload mass impacts success rates.
- **Launch Site vs. Flight Number (Categorical Plot)**:
  - **Why**: To identify trends at different launch sites and their flight histories.
  - **Insights**: Highlights site-specific success patterns or operational frequency.
- **Payload Mass vs. Launch Site (Scatter Plot)**:
  - **Why**: To analyze payload distribution and success rates at various launch sites.
  - **Insights**: Identifies which sites handle heavy payloads and their associated success rates.
- **Orbit Type vs. Success Rate (Bar Chart)**:
  - **Why**: To evaluate orbit types with the highest and lowest success rates.
  - **Insights**: Provides actionable insights into which orbits are more likely to achieve successful landings.
- **Yearly Success Rate Trend (Line Plot)**:
  - **Why**: To examine the temporal trend in success rates.
  - **Insights**: Shows improvement in success rates over time, reflecting operational advancements.
- **Payload Mass vs. Orbit Type (Scatter Plot)**:
  - **Why**: To correlate payload mass with orbit type and landing success.
  - **Insights**: Highlights which payload sizes succeed in specific orbits.



- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/edadataviz%20(2).ipynb

EDA with SQL

- **Database and Table Setup**
  - Installed required libraries (`sqlalchemy`, `ipython-sql`, `prettytable`).
  - Loaded the `sqlite` database extension and connected to a SQLite database (`my_data1.db`).
  - Created a new table `SPACEXTBL` from a CSV file (`Spacex.csv`) with flight data.
  - Dropped a table named `SPACEXTABLE` if it already existed.
  - Created `SPACEXTABLE` with filtered data where `Date` is not null.
- **Basic Queries**
  - Retrieved distinct `Launch_Site` values from `SPACEXTABLE`.
  - Queried the first 5 records from `SPACEXTABLE` where `Launch_Site` starts with "CCA".
- **Aggregations**
  - Calculated the total payload mass for flights with customer `NASA (CRS)`.
  - Computed the average payload mass for flights with `Booster_Version` = "F9 v1.1".
  - Found the earliest date of a successful ground pad landing.
- **Advanced Queries**
  - Retrieved distinct `Booster_Version` values for successful drone ship landings with payload mass between 4000 and 6000 kg.
  - Counted the occurrences of each landing outcome (`Landing_Outcome`) grouped by the outcome.
- **Nested Queries**
  - Identified `Booster_Version` used for the flight with the maximum payload mass.
  - Retrieved all distinct landing outcomes associated with the heaviest payload flight.
- **Date and String Manipulations**
  - Extracted month information from the `Date` column and filtered failed drone ship landings in 2015.
  - Counted landing outcomes for flights between June 4, 2010, and March 20, 2017, and ordered results by outcome count.

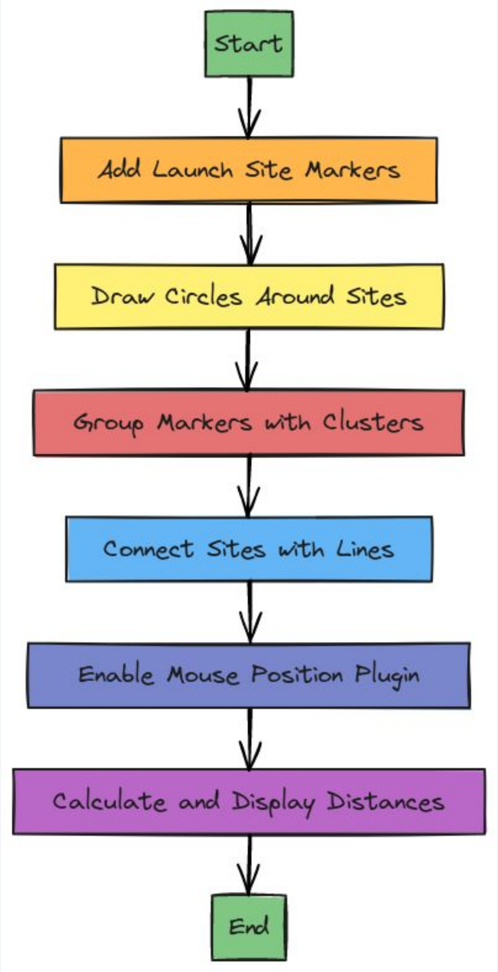
https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

Build an Interactive Map with Folium

- **Markers**
  - Added markers to represent launch sites with popups displaying their names.
  - Used different colors (`green`, `red`, `blue`) to indicate different categories (e.g., success, failure, specific sites).
  - Added markers with custom HTML labels using `DivIcon` to display text on the map.
- **Circles**
  - Placed blue circles around launch sites with a defined radius and semi-transparent fill.
  - Included popup labels inside circles to display the site name.
- **Clustered Markers**
  - Used the `MarkerCluster` plugin to group launch site markers into clusters for better map readability.
- **Lines**
  - Drew `PolyLine` objects to connect launch sites to their closest points of interest (e.g., coastline).
  - Used these lines to visually demonstrate distances between points.
- **Mouse Position Plugin**
  - Enabled the `MousePosition` plugin to display real-time latitude and longitude coordinates as the user moves the mouse over the map.
- **Distance Calculations**
  - Used the Haversine formula to calculate the distance between a launch site and its closest coastline.
  - Displayed calculated distances on the map as custom markers.
-
- **Markers**: Pinpointed specific locations (e.g., launch sites) and provided details through popups for quick identification.
- **Circles**: Highlighted areas around locations to emphasize regions of interest and spatial context.
- **Clustered Markers**: Reduced clutter by grouping nearby markers, improving readability in dense areas.
- **Lines (PolyLine)**: Connected related points (e.g., launch sites and coastlines) to illustrate geographic relationships.
- **Mouse Position Plugin**: Allowed real-time display of coordinates for precise location identification.
- **Distance Calculations**: Quantified spatial relationships (e.g., site-to-coastline distance) for actionable insights.
- **Purpose**: Enhanced visual context, improved interactivity, clarified outcomes (e.g., success/failure), and facilitated data analysis.

- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/lab_jupyter_launch_site_location%20(1).ipynb



Build a Dashboard with Plotly Dash

Plots/Graphs

1. **Scatter Plot (Payload vs. Launch Success)**
   - **Purpose**: Visualize the relationship between payload mass and launch success for each launch site.
   - **Customization**: Allows users to filter by launch site to focus on specific data.
2. **Pie Chart (Success Count by Launch Site)**
   - **Purpose**: Show the proportion of successful launches for each site.
   - **Customization**: Dynamically updates based on the user's launch site selection.

Interactions

1. **Dropdown Menu (Launch Site Selection)**
   - **Functionality**: Allows users to select a specific launch site or view data for all sites.
   - **Impact**: Filters data displayed in the scatter plot and pie chart.
2. **Payload Range Slider**
   - **Functionality**: Enables users to adjust the range of payload mass for filtering data.
   - **Impact**: Updates the scatter plot to display launches within the selected payload range.
3. **Dynamic Updates**
   - Both the scatter plot and pie chart update in real-time based on the user's selections in the dropdown menu and slider.

Purpose

- These elements provide a user-friendly and interactive way to explore SpaceX launch data, enabling insights into launch performance, payload impacts, and site-specific metrics.

- Explain why you added those plots and interactions

- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/spacex_dash_app.py%20(1).1

Plots

1. **Scatter Plot (Payload vs. Launch Success)**
   - **Purpose**: To explore how payload mass relates to launch success.
   - **Why It Was Added**:
     - It helps users identify trends because it shows whether heavier or lighter payloads impact success.
     - It provides a clear view of variability because users can see different outcomes based on payload size.
2. **Pie Chart (Success Count by Launch Site)**
   - **Purpose**: To show the proportion of successful launches for each site.
   - **Why It Was Added**:
     - It gives a quick summary of site performance because users can compare success rates visually.
     - It helps stakeholders find the most reliable sites to make decisions efficiently.

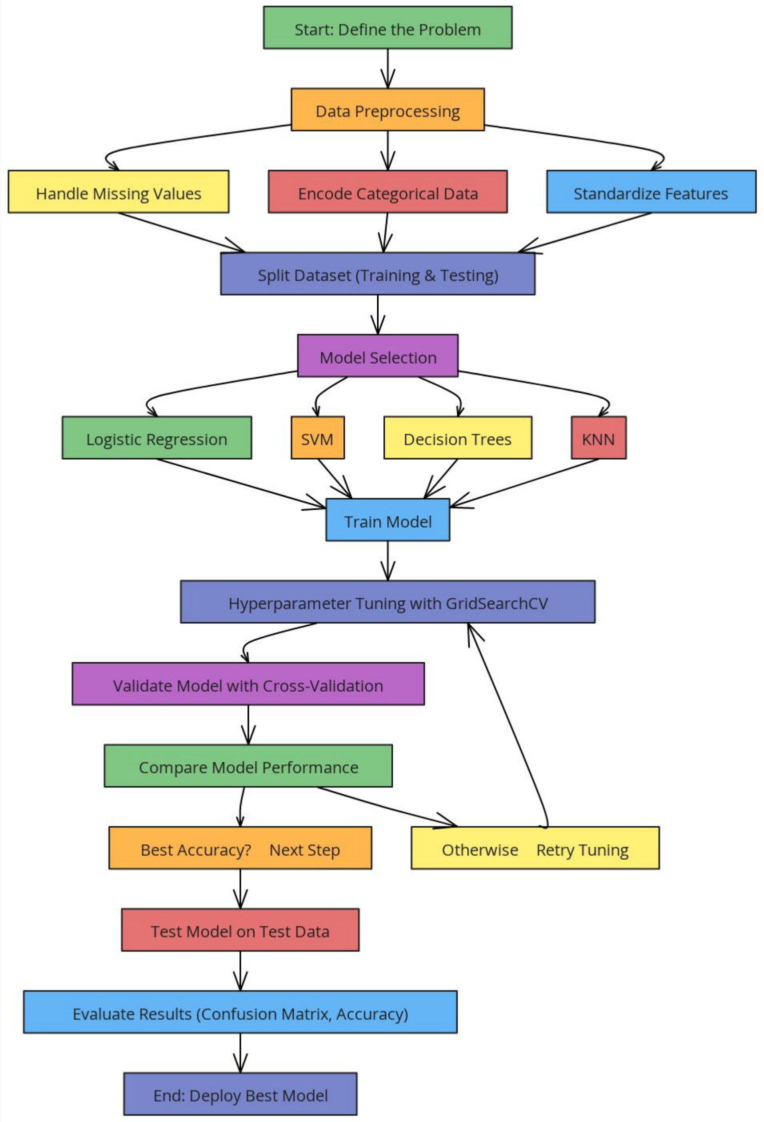
Interactions

1. **Dropdown Menu (Launch Site Selection)**
   - **Purpose**: To allow users to focus on data from specific launch sites or view all sites together.
   - **Why It Was Added**:
     - It supports detailed analysis because users can filter by individual sites.
     - It provides flexibility to explore specific trends or overall performance.
2. **Payload Range Slider**
   - **Purpose**: To filter launches by payload mass.
   - **Why It Was Added**:
     - It enables focused analysis because users can examine trends in specific payload ranges.
     - It helps analyze lighter or heavier payloads to find patterns in their performance.
3. **Dynamic Updates**
   - **Purpose**: To make the dashboard responsive to user interactions.
   - **Why It Was Added**:
     - It ensures real-time updates because users need instant feedback for their selections.
     - It enhances interactivity to make exploring the data more engaging and intuitive.

Predictive Analysis (Classification)

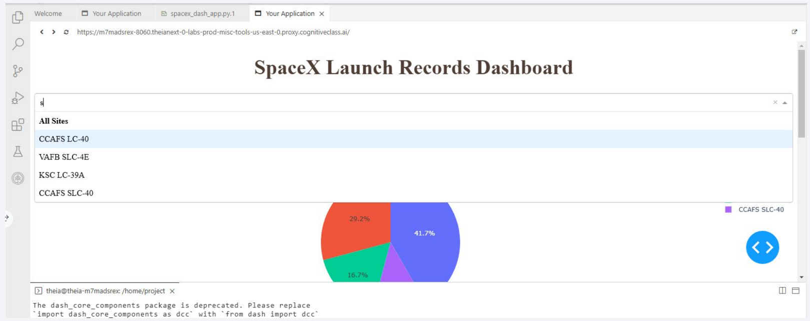
- **Data Preparation**:
  - Preprocessed the dataset by standardizing features, creating training and testing splits, and generating a target variable (`Class`).
- **Model Building and Hyperparameter Tuning**:
  - Built models using Logistic Regression, Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN).
  - Applied `GridSearchCV` to optimize hyperparameters for each model, such as `C`, `kernel`, and `solver`.
- **Evaluation**:
  - Used cross-validation (`cv=10`) to ensure consistency across training folds and computed metrics such as accuracy.
  - Visualized results with confusion matrices to assess predictions, focusing on true positives and false positives.
- **Testing**:
  - Evaluated models on test data, with Logistic Regression, SVM, and KNN achieving the highest accuracy of **83.3%**.
- **Selection**:
  - Identified Logistic Regression, SVM, and KNN as the best-performing models with comparable results, while Decision Trees performed slightly lower at **77.8%** accuracy.

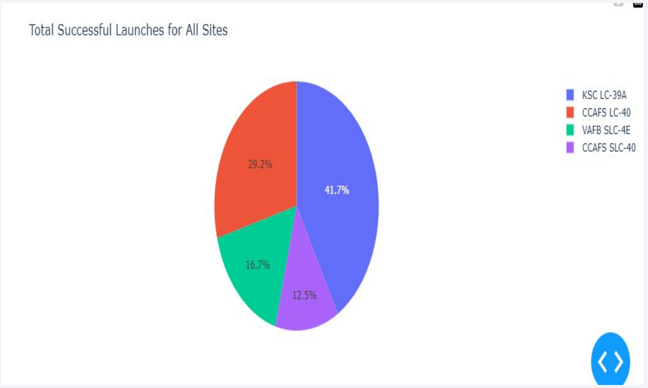
- https://github.com/MoSkarrar/SpaceX-Falcon-9-landing-prediction-project/blob/codes/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb



Results

- Exploratory data analysis results
  - This analysis revealed key factors influencing SpaceX's mission success, including the impact of launch sites, orbit-specific trends, and payload optimization, with Cape Canaveral Launch Complex 40 and payloads in the 2,000–5,000 kg range showing the highest success rates. Yearly improvements in success rates, combined with clear landing outcome labels, provide a foundation for machine learning models and operational enhancements.
- Interactive analytics demo in screenshots
- Predictive analysis results



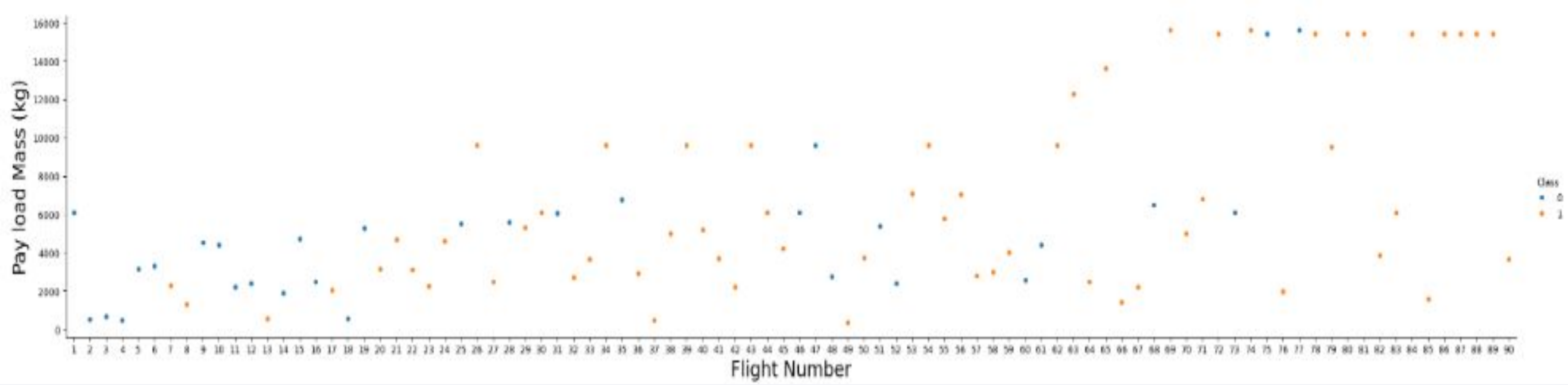


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site



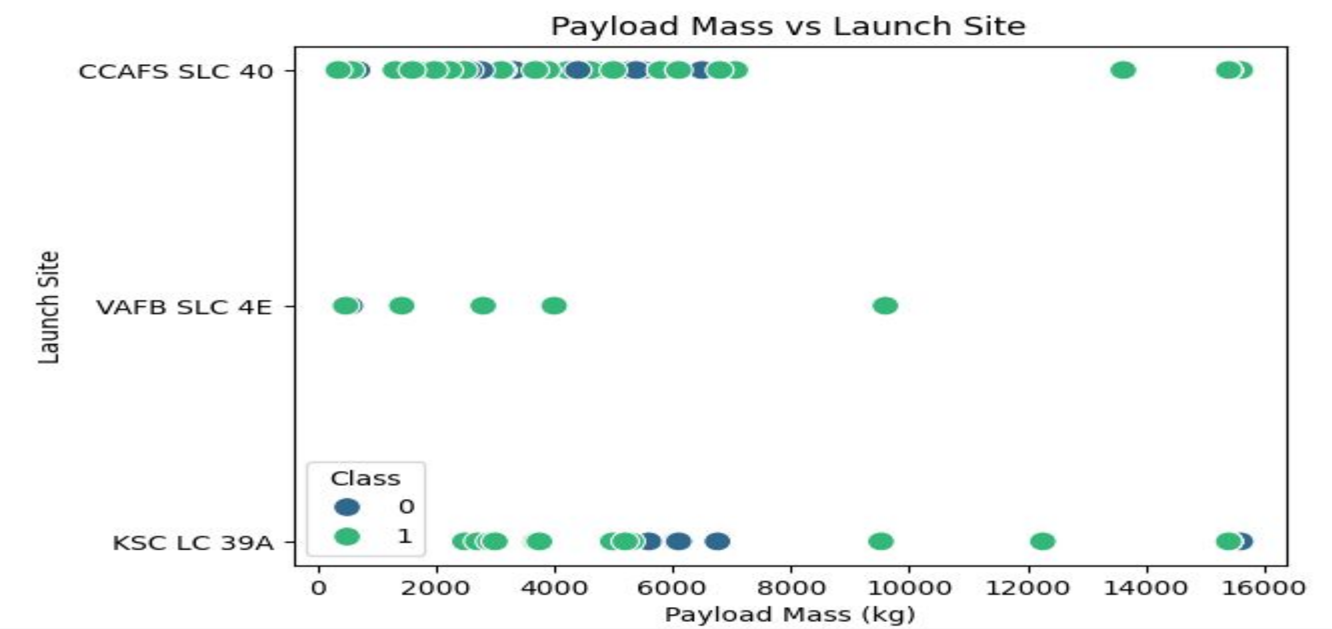
- Show the screenshot of the scatter plot with explanations

The scatter plot shows that SpaceX's success rates (orange points) have significantly improved over time, as indicated by a higher proportion of successful launches in later flights. Payloads between 2,000 and 6,000 kg saw both successes and failures, while heavier payloads above 10,000 kg were predominantly successful in more recent missions. This demonstrates SpaceX's advancements in reliability and technology, handling larger and more complex payloads with higher success rates over time.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

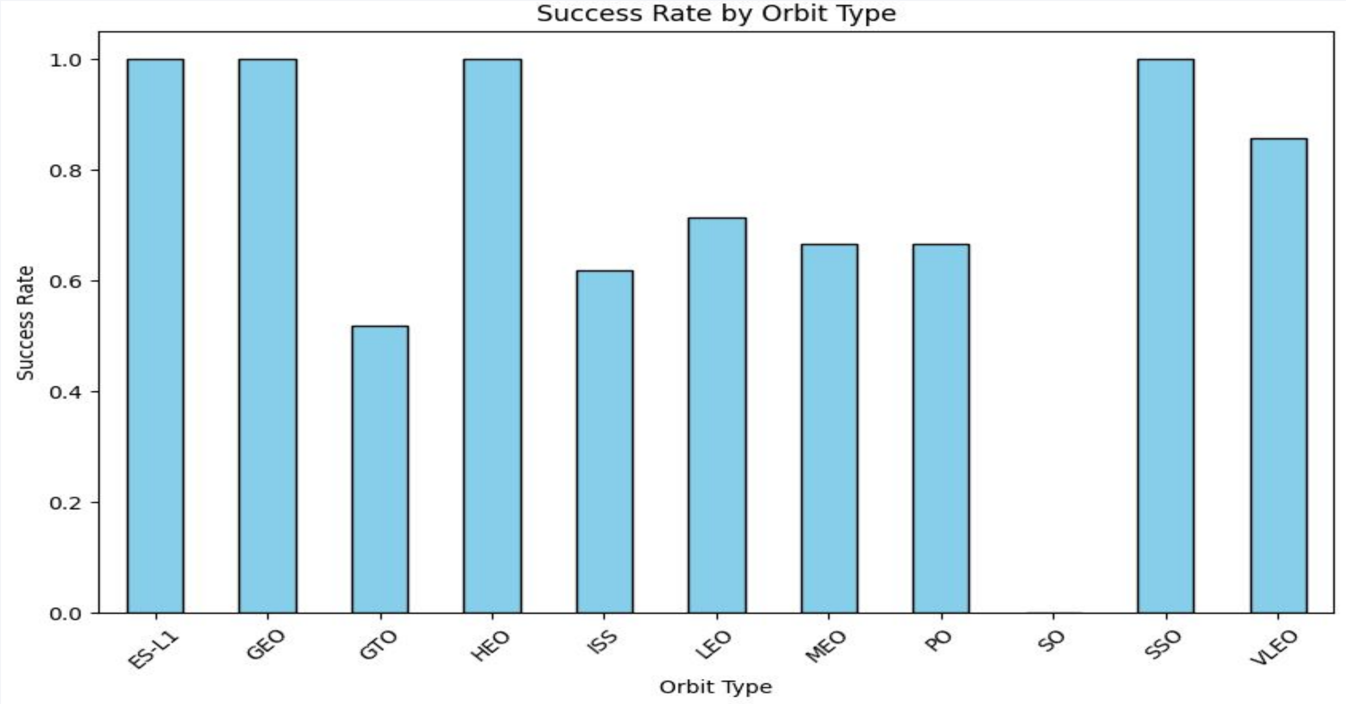
- Show the screenshot of the scatter plot with explanations



The scatter plot shows that **CCAFS SLC 40** handles the highest number of launches across a wide range of payloads, achieving consistent success (lighter points) even for larger payloads. **KSC LC 39A**, although hosting fewer launches, focuses on heavier payloads (over 6,000 kg) with a high success rate, while **VAFB SLC 4E** supports lighter payloads with mixed results. This demonstrates SpaceX's strategic use of launch sites, where **CCAFS SLC 40** serves as a versatile hub, and **KSC LC 39A** specializes in missions requiring advanced heavy-lift capabilities.

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations

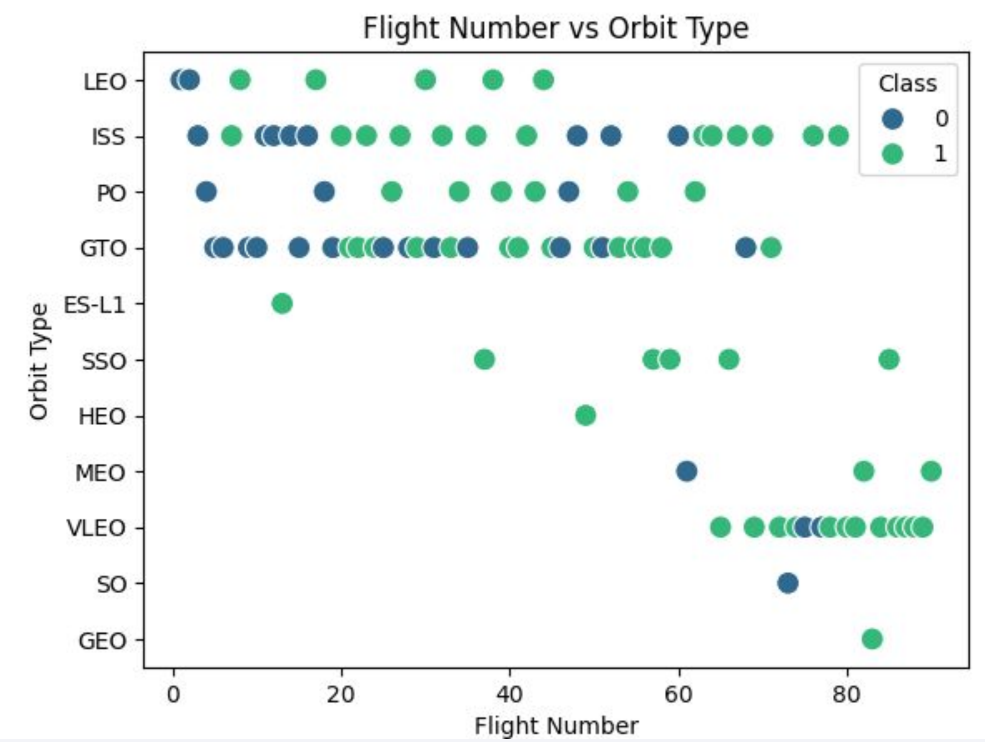


This chart demonstrates the variability in success rates across different orbit types, highlighting significant differences in mission outcomes. Orbits like **ES-L1**, **GEO**, and **SSO**, which show a 100% success rate, are typically used for highly specialized and high-priority missions, such as Earth observation, geosynchronous communications, or solar observation, where meticulous planning and precise execution minimize the risk of failure. Conversely, the **GTO** orbit, with the lowest success rate, is often associated with heavy payloads and complex orbital transfer maneuvers, which increase technical and operational challenges, including higher risks of failures during deployment or landing phases. These insights emphasize the need for tailored strategies and optimized resource allocation to improve success rates, particularly for challenging orbits like GTO, while maintaining the high reliability of specialized mission-critical orbits.

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations

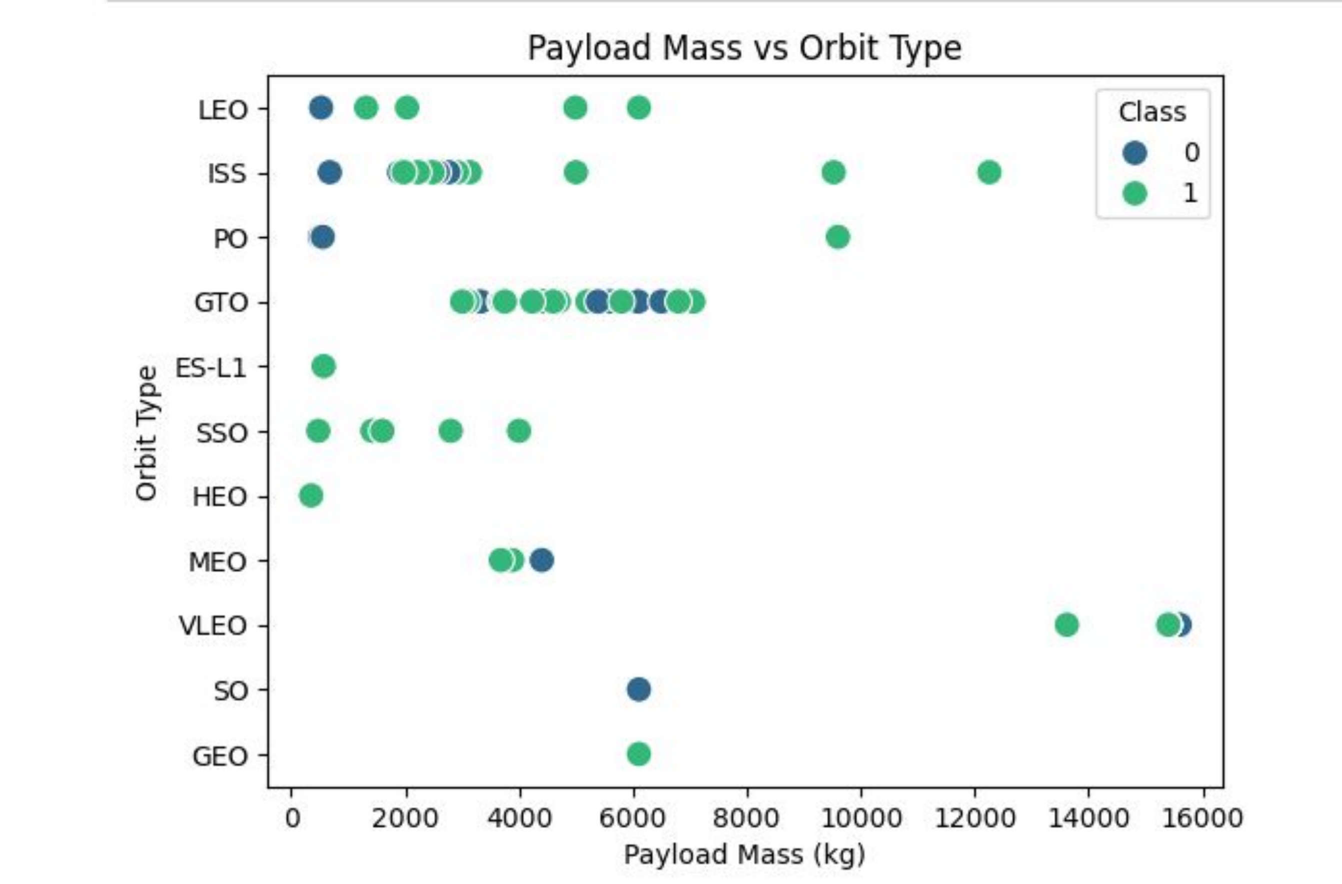


This plot highlights the evolution of SpaceX's mission reliability over time, as seen through the increasing success rates across various orbit types. Early failures across multiple orbits, such as **GTO** and **LEO**, reflect SpaceX's initial learning curve in achieving consistent launch and landing capabilities, likely due to technical limitations and operational adjustments during its developmental phase. Over time, orbits like **LEO** and **ISS** show a notable improvement in success rates, with later flights exhibiting far fewer failures, demonstrating SpaceX's growing operational maturity and mastery of routine missions. The scattered failures in **GTO** highlight the ongoing challenges of high-complexity missions involving heavy payloads and intricate orbital maneuvers. Meanwhile, the near-perfect success in specialized orbits such as **SSO** and **ES-L1** showcases SpaceX's ability to execute precision-critical missions with high reliability, often for advanced scientific or observational purposes.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

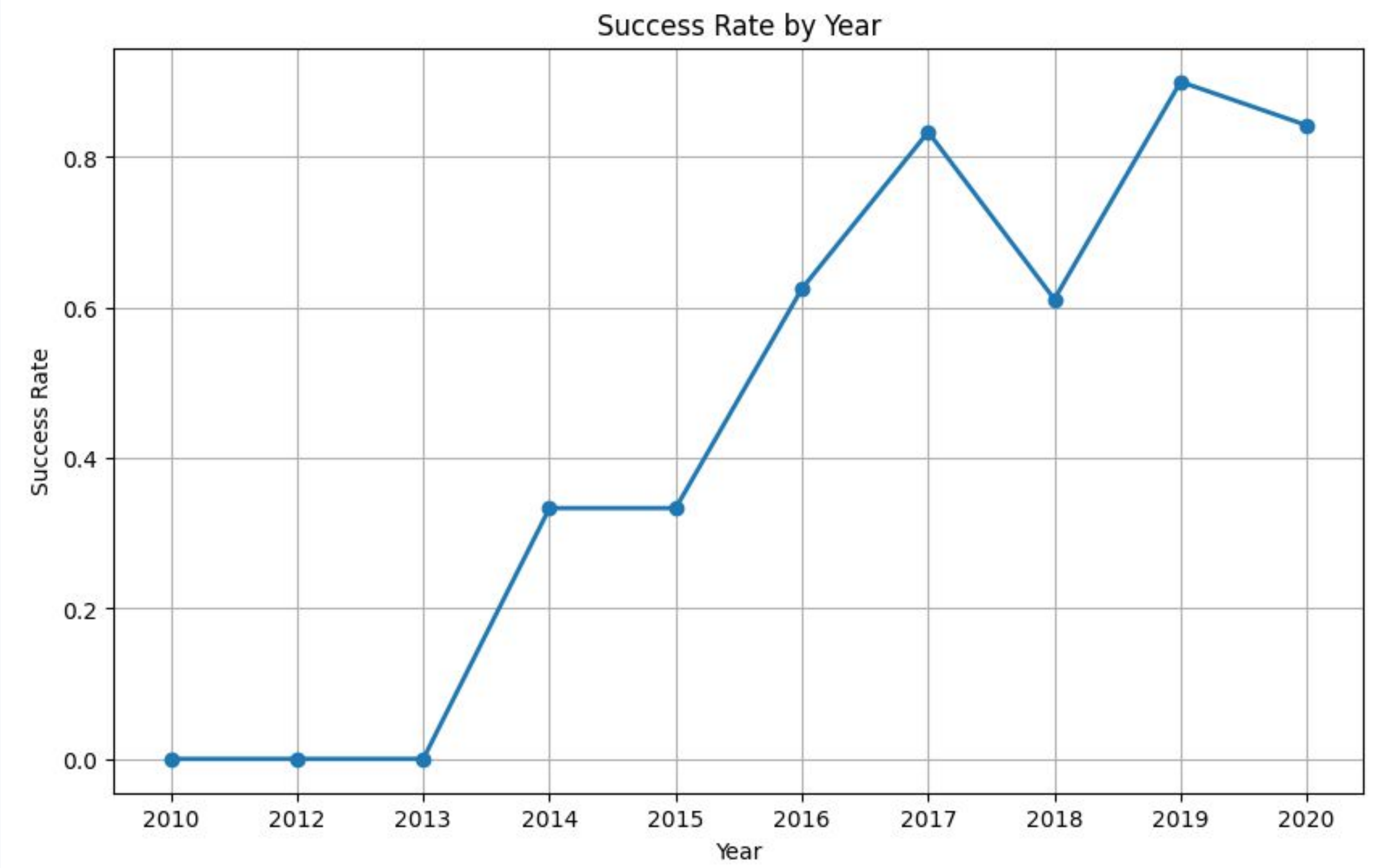
This scatter plot highlights the relationship between payload mass and mission success across different orbit types. Routine orbits like **LEO** and **ISS** achieve high success rates for lower payloads due to extensive operational experience and lower technical complexity. Specialized orbits like **SSO** and **GEO** are associated with larger payloads, showcasing SpaceX's ability to execute precision-critical and high-stakes missions successfully. The mixed performance in **GTO** for mid-range payloads underlines the challenges of handling orbital transfers and complex maneuvers, requiring further optimization. Overall, the data reflects SpaceX's expertise in managing payload-specific challenges tailored to different orbital requirements.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate.

This line chart shows SpaceX's remarkable progression in launch success rates over time, reflecting significant milestones in its technological and operational development. From 2010 to 2013, success rates remained at 0%, indicating the early experimental phase where the company focused on overcoming foundational challenges in rocket design and landing techniques. The sharp rise in 2015, reaching a 50% success rate, highlights a pivotal moment driven by advancements in reusable rocket technology and improvements in operational efficiency. By 2017, SpaceX achieved a peak success rate of nearly 90%, signifying the maturity of its systems, with successful missions becoming the norm even for complex launches. Although a slight dip occurred in 2018, success rates rebounded and stabilized above 80%, showcasing SpaceX's ability to maintain high reliability while scaling its operations and supporting diverse, high-stakes missions globally.



All Launch Site Names

- Find the names of the unique launch sites

The SQL query `SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;` retrieves all unique launch site names from the `SPACEXTABLE`. This query helps identify the specific locations SpaceX has used for its rocket launches.

Insights:

1. The dataset includes three primary launch sites: **CCAFS LC-40**, **VAFB SLC-4E**, and **KSC LC-39A**.
2. **CCAFS LC-40** (Cape Canaveral Air Force Station) appears more frequently in SpaceX's operations, likely due to its proximity to key infrastructure and its role in handling a wide range of missions.
3. **VAFB SLC-4E** (Vandenberg Air Force Base) is specialized for polar and sun-synchronous orbit missions, supporting launches that require a southern trajectory.
4. **KSC LC-39A** (Kennedy Space Center) is often used for high-priority missions, including Falcon Heavy launches and crewed flights, highlighting its advanced infrastructure.

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

Launch Site Names Begin with 'CCA'

Explanation:

The SQL query `SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;` retrieves the first 5 records from the `SPACEXTABLE` where the launch site starts with "CCA" (indicating Cape Canaveral locations). This query filters the dataset to focus on launches originating from Cape Canaveral.

Insights:

1. **Launch Site**:
   - All five launches took place at **CCAFS LC-40**, one of SpaceX's key launch facilities, indicating its central role in early SpaceX missions.
2. **Payload and Mission**:
   - Early payloads included experimental and demo missions like the **Dragon Spacecraft Qualification Unit** and **Dragon Demo Flight C1**, reflecting SpaceX's initial efforts to establish reliability and capability.
   - Customers include SpaceX itself and NASA (COTS and CRS programs), highlighting early partnerships that supported SpaceX's development.
3. **Payload Mass**:
   - Payloads were relatively light in these early missions (0–677 kg), which aligns with the experimental nature of these launches.
4. **Orbit Type**:
   - All missions targeted **LEO (Low Earth Orbit)**, demonstrating a focus on simpler and shorter missions during SpaceX's early stages of operations.
5. **Landing Outcome**:
   - Landing attempts were either marked as **failure (parachute)** or **no attempt**, reflecting the learning phase in perfecting landing technologies for rocket reusability

```
* 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

Total_Payload_Mass
45596

Explanation:

The SQL query `SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';` calculates the total payload mass (in kilograms) carried by SpaceX on behalf of NASA's **Commercial Resupply Services (CRS)** missions.

Insights:

1. The total payload mass delivered by SpaceX for NASA (CRS) missions is **45,596 kg**, reflecting the significant role SpaceX plays in resupplying the International Space Station (ISS).
2. These missions demonstrate SpaceX's reliability in supporting NASA's logistics needs for scientific equipment, supplies, and technology delivery to the ISS.
3. The high payload mass highlights the efficiency of SpaceX's Falcon 9 rockets in handling consistent and repeated missions for NASA.
4. This success underscores SpaceX's long-term partnership with NASA and its capability to meet demanding payload requirements for critical orbital missions.

Average Payload Mass by F9 v1.1

Explanation:

The SQL query `SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';` calculates the average payload mass (in kilograms) carried by the **Falcon 9 v1.1** booster version.

Average_Payload_Mass
2928.4

Insights:

1. The average payload mass for missions using the **F9 v1.1** booster is **2,928.4 kg**, showcasing its capability for medium-range payloads.
2. This payload capacity highlights the Falcon 9 v1.1's role in missions targeting **LEO** and **GTO**, which require moderate payload flexibility.
3. The results reflect SpaceX's incremental progress in developing booster versions with increasing payload capacities, paving the way for larger and more demanding missions.
4. The Falcon 9 v1.1 booster served as a critical stepping stone in SpaceX's evolution, bridging earlier versions and the more powerful Falcon 9 Full Thrust and Block 5 variants.

First Successful Ground Landing Date

- The query identifies the date of SpaceX's first successful landing on a ground pad by selecting the earliest date where the "Landing_O was marked as "Success (ground pad)." This query helps pinpoi significant milestone in SpaceX's development of reusable rocket technology.

First_Successful_Landing
2015-12-22

- This is the query “The SQL query `SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';`”

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query identifies specific Falcon 9 Full Thrust (FT) booster versions that successfully landed on a drone ship while carrying payloads within the 4,000–6,000 kg range. These boosters demonstrate precision in handling mid-range payloads and performing controlled landings, showcasing their role in SpaceX’s reusable rocket operations. The query filters the data to find distinct booster versions with a **"Landing_Outcome"** of 'Success (drone ship)' and payload masses between **4,000 and 6,000 kg**. It uses the `DISTINCT` clause to ensure each booster version is listed only once.

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

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

Total Number of Successful and Failure Mission Outcomes

- The query groups rows by `Landing_Outcome` and counts the occurrences of each unique outcome using the `COUNT(*)` function. It returns a summary table with distinct landing outcomes and their respective counts.
- The data shows **"No attempt"** as the most frequent outcome, reflecting early missions without recovery goals, while **"Success"** (drone ship and ground pad combined) dominates, showcasing SpaceX's advancements in reusable rockets. Failures like **"Failure (drone ship)"** and rare cases like **"Uncontrolled (ocean)"** highlight operational challenges during the developmental phase.
- %sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE GROUP BY "Landing_Outcome";

Landing_Outcome	Outcome_Count
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- The query finds distinct booster versions used for the mission with the maximum payload mass by nesting a subquery that calculates the highest payload mass in the dataset. The main query filters rows where the payload matches this maximum value and returns unique booster versions.
- The query identifies booster versions capable of handling the heaviest payload, showcasing the operational capacity and reliability of these specific Falcon 9 Block 5 boosters for high-stakes missions.

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

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

- The query extracts the month from the `Date` column and filters rows where the `Landing_Outcome` is "Failure (drone ship)" in the year 2015. It selects columns for `Month`, `Landing_Outcome`, `Booster_Version`, and `Launch_Site` for targeted analysis.
- The image shows that in 2015, there were two failures involving drone ship landings, occurring in **January** and **April**, both using the **F9 v1.1** booster version at the **CCAFS LC-40** launch site. These results reflect early challenges in achieving successful drone ship recoveries during SpaceX's development of reusable rockets.
- %sql SELECT substr("Date", 6, 2) AS Month,"Landing_Outcome","Booster_Version","Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr("Date", 1, 4) = '2015';

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The query filters data for landing outcomes between **2010-06-04** and **2017-03-20**, groups the rows by `Landing_Outcome`, and counts their occurrences using `COUNT(*)`. The results are ordered by `Outcome_Count` in descending order for easy interpretation.
- During the specified period, **"No attempt"** was the most frequent outcome (10 instances), reflecting SpaceX's initial focus on payload delivery without recovery efforts. Failures like **"Failure (drone ship)"** and **"Failure (parachute)"** highlight early challenges in achieving controlled landings. Successful outcomes, including **"Success (drone ship)" (5)** and **"Success (ground pad)" (3)**, showcase SpaceX's progress in mastering reusable rocket technology. Controlled landings on the ocean (3) indicate occasional operational adjustments when specific recovery methods were not feasible.
- %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;

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

Section 3

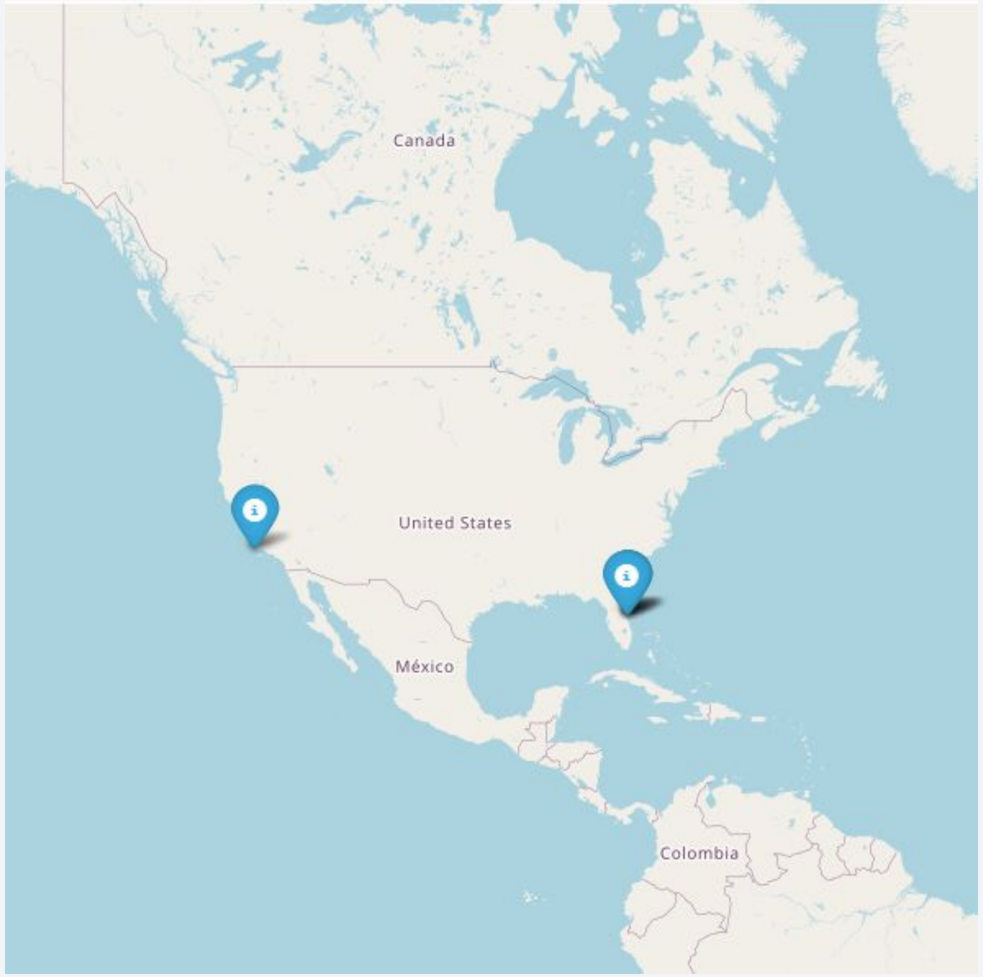
Launch Sites Proximities Analysis

Global View of SpaceX Launch Sites

The code initializes a map centered near Cape Canaveral with a zoom level of 5. It iterates over a list of launch sites and adds markers for each site using their latitude and longitude, displaying the site name in a popup.

Explanation:

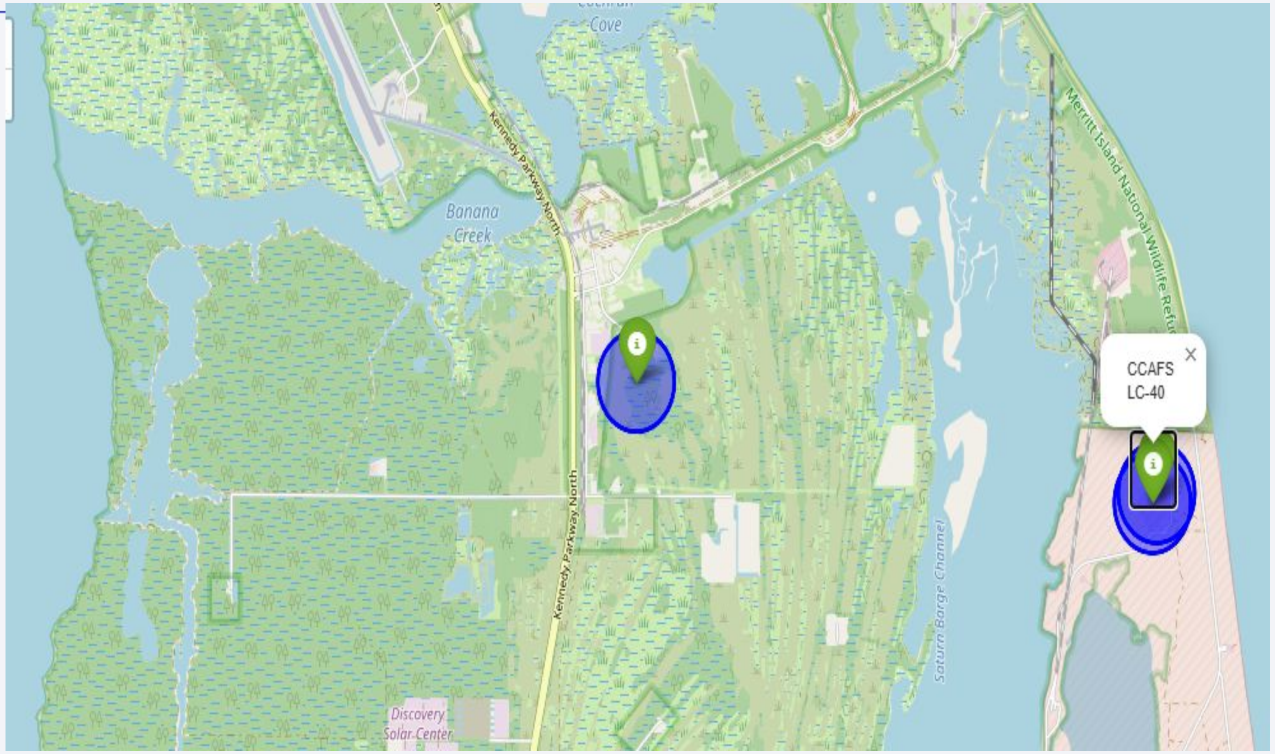
The map shows the geographic distribution of SpaceX's key launch sites in the United States. **CCAFS LC-40**, **CCAFS SLC-40**, and **KSC LC-39A** are clustered in Florida near Cape Canaveral, reflecting its significance as a hub for launches. **VAFB SLC-4E** in California supports polar and sun-synchronous orbits, showcasing SpaceX's strategic use of diverse locations for mission-specific needs.



Color-Labeled Launch Outcomes Map

The code initializes a map centered near Kennedy Space Center and iterates over launch sites to add circles with a 500-meter radius and markers with popup labels for each site. The circles visualize the immediate vicinity of each launch site, while markers highlight their specific locations.

The image highlights **CCAFS SLC-40**,**CCAFS LC-40, and KSC LC-39A** showing a marker pinpointing its precise location and a 500-meter radius circle visualizing its operational zone. This provides a spatial understanding of the site's proximity to infrastructure and geographical constraints, crucial for safe rocket launches and recoveries.



Proximity Analysis of Launch Site

Summary of Key Elements and Findings:

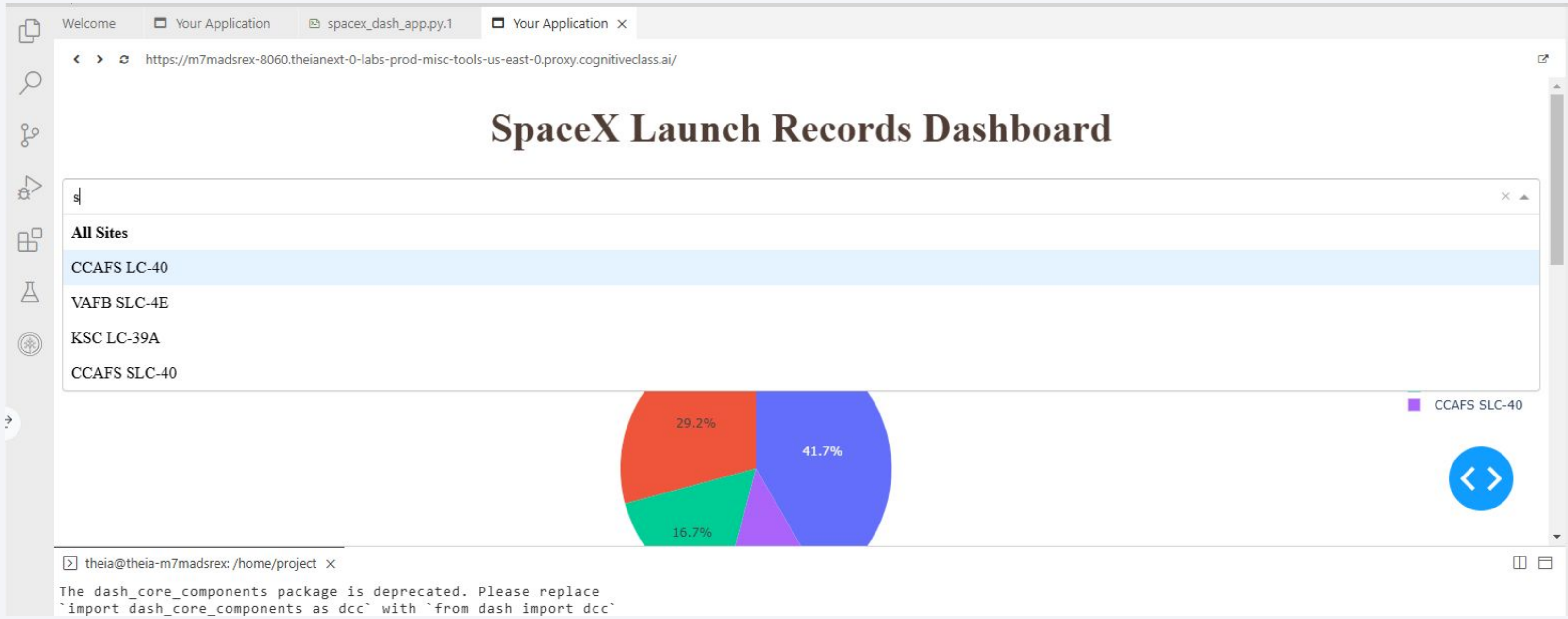
1. **Launch Site (Cape Canaveral)**:
   - Marked with a **red icon** to indicate its central role in the visualization.
2. **Coastline (Blue)**:
   - A **blue PolyLine** connects the launch site to the coastline.
   - **Distance**: `0.95 km`, emphasizing the site's strategic proximity to the ocean for safe rocket launches.
3. **Railroad (Yellow)**:
   - A **yellow PolyLine** connects the launch site to the nearest railroad.
   - **Distance**: `0.66 km`, highlighting efficient logistical access for heavy equipment transportation.
4. **Highway (Purple)**:
   - A **purple PolyLine** connects the launch site to the nearest highway.
   - **Distance**: `0.59 km`, showing accessibility for personnel and smaller equipment.
5. **Nearest City (Red)**:
   - A **red PolyLine** connects the launch site to the nearest city.
   - **Distance**: `19 km`, ensuring a safe buffer from urban areas while maintaining reasonable accessibility.

Section 4

Build a Dashboard with Plotly Dash

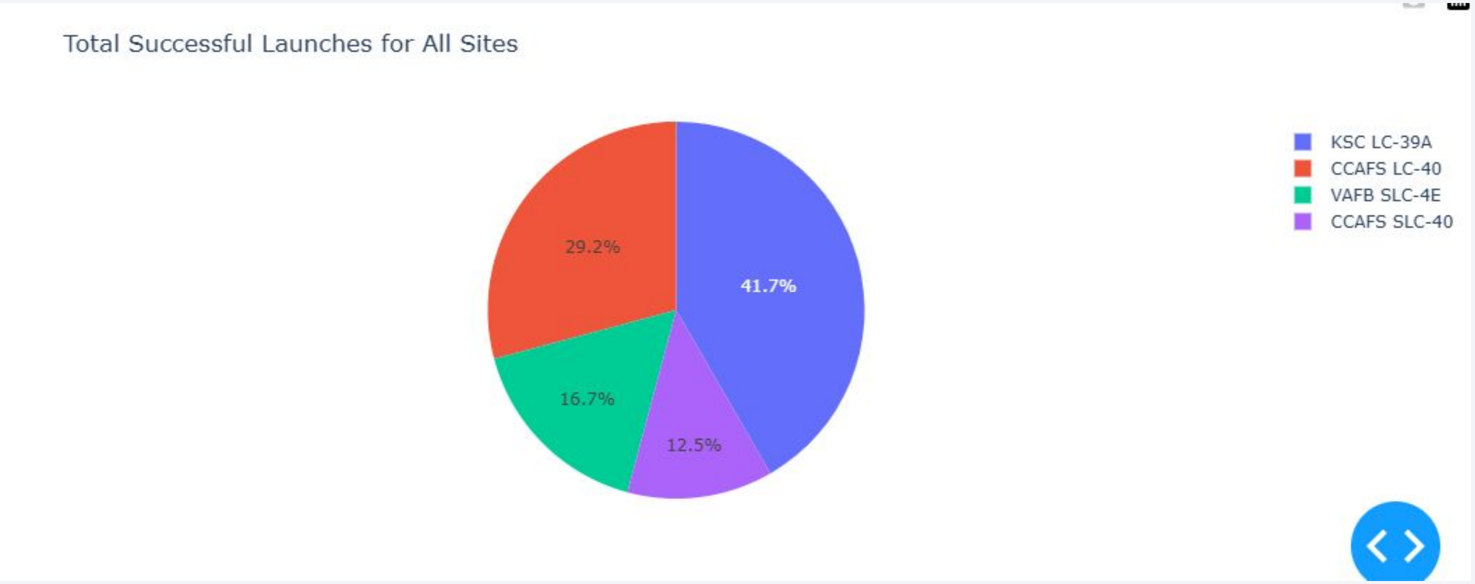
Launch Success Counts by Site - Dashboard View

- **Dropdown Menu**: The dropdown menu at the top allows users to select specific launch sites or view all sites collectively. In this example, "All Sites" is selected, providing an overview.
- **Pie Chart**: The pie chart visually represents the proportion of successful launches from various SpaceX launch sites. Each segment corresponds to a site, with percentages displayed to highlight their contributions.
- **Findings**:
  - **CCAFS SLC-40** holds the highest share at 41.7%, indicating its significant role in SpaceX operations.
  - Other sites like **VAFB SLC-4E** and **KSC LC-39A** contribute a smaller but meaningful percentage of successful launches.
  - This visual helps identify the operational importance of each site and their success distribution.



Total Successful Launches for All Sites - Pie Chart View

- **Pie Chart Overview**: This visualization represents the proportion of successful launches across all SpaceX launch sites.
- **Color-Coded Segments**:
  - **Blue (41.7%)**: Indicates the success rate of **KSC LC-39A**, the leading launch site in terms of successful launches.
  - **Red (29.2%)**: Represents the contributions from **CCAFS LC-40**, showcasing its significant role in SpaceX operations.
  - **Green (16.7%)**: Reflects successes at **VAFB SLC-4E**, which has a smaller but impactful percentage.
  - **Purple (12.5%)**: Highlights the additional contribution of **CCAFS SLC-40** to total successful launches.
- **Findings**:
  - **KSC LC-39A** emerges as the top-performing site.
  - The chart clearly demonstrates the operational distribution and success rates for SpaceX's launch facilities.



Payload vs. Success for All Sites

- **Scatter Plot Overview**: This plot visualizes the relationship between payload mass (in kilograms) and the success rate of launches for all SpaceX launch sites.
- **Key Elements**:
  - **X-Axis**: Represents the payload mass (in kilograms).
  - **Y-Axis**: Represents the launch outcome (0 for failure, 1 for success).
  - **Color-Coded Dots**:
    - **Blue (v1.1)**: Launches using booster version **v1.1**.
    - **Red (FT)**: Launches using booster version **FT**.
    - **Green (B4)**: Launches using booster version **B4**.
- **Findings**:
  - Successful launches (value 1) are clustered at various payload ranges, indicating the versatility of the boosters in handling different payload masses.
  - Certain payload ranges (e.g., near 5000 kg) appear to have consistently higher success rates across booster versions.

Section 5

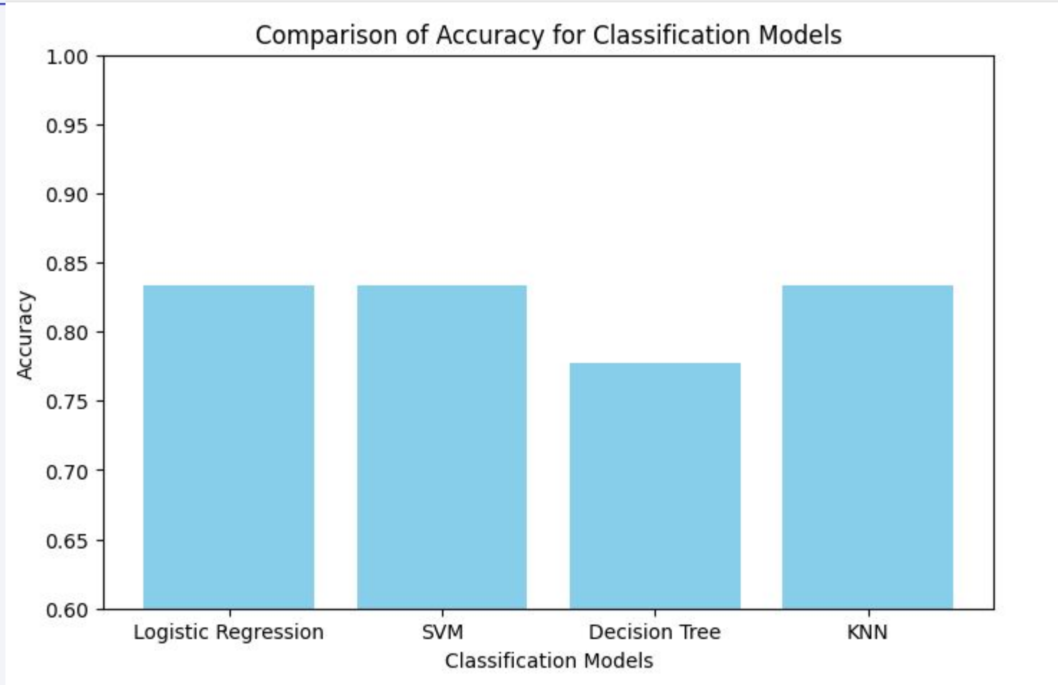
Predictive Analysis (Classification)

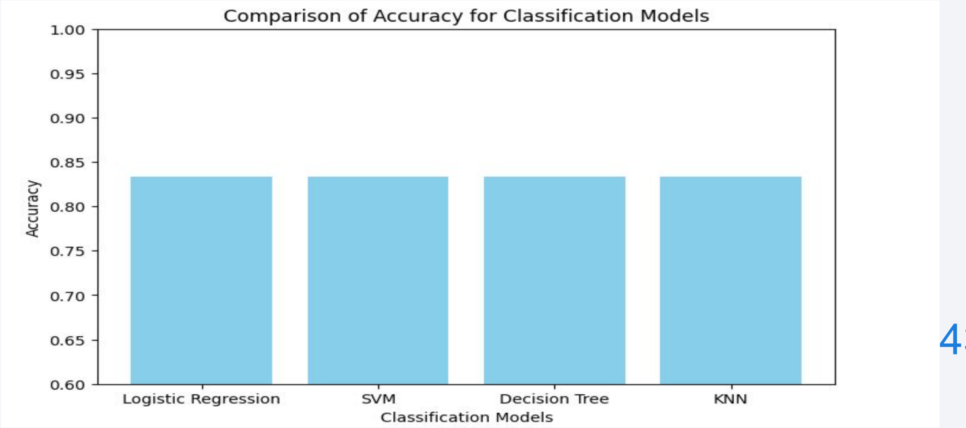
Classification Accuracy

Based on the bar chart and the comparison of accuracy scores, the **Support Vector Machine (SVM)** is the **best-performing classification model** with 0.83333334 accuracy.

Key Reasons:

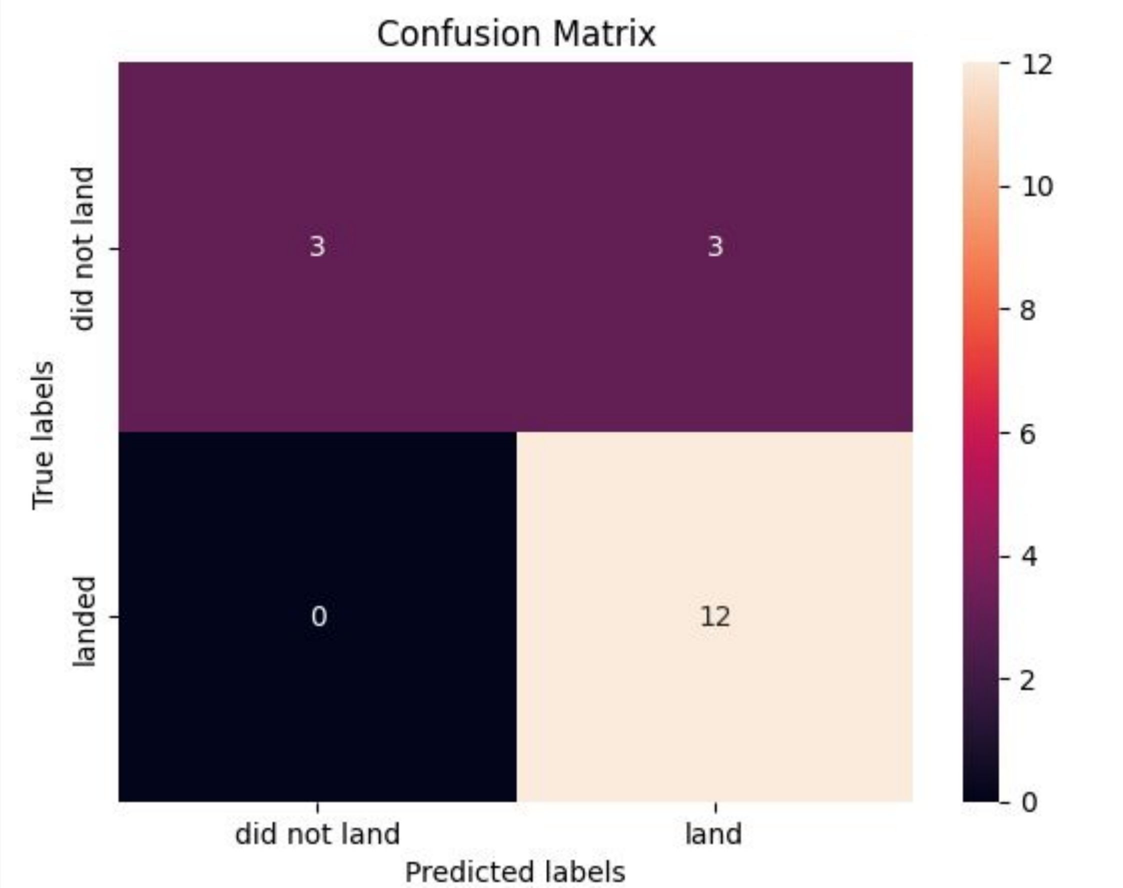
1. **Effective for High-Dimensional Data**:
   - SVM handles high-dimensional spaces well and performs robustly when the number of features is large relative to the number of samples.
2. **Flexibility**:
   - By using kernel tricks (e.g., RBF or linear), SVM can model both linear and nonlinear relationships effectively.





Confusion Matrix

- The best-performing model (SVM) excels in predicting successful landings, with 12 true positives and no false negatives, showcasing strong accuracy for the "landed" class. However, it struggles with predicting unsuccessful landings, resulting in 3 false positives, indicating room for improvement in distinguishing "did not land" cases.



Conclusions

- **SVM and Logistic Regression** perform similarly, showing high accuracy, as both are effective for linear or near-linear separable data when properly tuned.
- **KNN** achieves comparable performance to Logistic Regression and SVM, indicating it works well with well-distributed and scaled data, but it might be sensitive to the choice of neighbors.
- **Decision Tree** performs the worst due to its tendency to overfit, even with parameter tuning, highlighting its sensitivity to noise and lack of generalization.
- When tuned with the best parameters, **KNN, Logistic Regression, and SVM** show consistency and robustness, while **Decision Tree** struggles despite adjustments, emphasizing the need for ensemble methods like Random Forest for improvement.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!