Winning Space Race with Data Science

Hemant Kumar
6th Dec 2025

Outline

- **Executive Summary** – Concise overview of the key findings, recommendations and implications.
- **Introduction** – Context, objectives and the scope of the report.
- **Methodology** – Data source, analytical approaches and limitations.
- **Results** – Key findings supported by visuals and limitations.
  - EDA with Visualization
  - EDA with SQL
  - Interactive Maps with Folium
  - Plotly Dash Dashboard
  - Predictive Analysis
- **Conclusion** – Interpretation of results and strategic implications.
- **Appendix** – Supplement data, charts and technical details. – Links to GitHub

EXECUTIVE SUMMARY – PREDICTING ROCKET LANDING SUCCESS

SUMMARY OF METHODOLOGIES

- **Data Acquisition**
  - SpaceX REST API and Web Scraping
  - SQL queries for structured data retrieval
- **Data Wrangling & Analysis**
  - Creation of success/fail outcome variable
  - Payload, launch site, flight number, and orbit classification
- **Exploratory Data Analysis (EDA)**
  - SQL, Pandas, Matplotlib for statistical insights
  - Yearly trends and launch site performance
- **Interactive Visual Analytics**
  - Folium maps of geographic proximity
  - Plotly Dash dashboards for dynamic exploration.

EXECUTIVE SUMMARY – PREDICTING ROCKET LANDING SUCESS

- **Predictive Modeling**
  - Classification models: Logistic Regression, SVM, Decision Tree, KNN
  - Hyperparameter tuning and model comparison

SUMMARY OF KEY RESULTS

- **Exploratory Insights**
  - Rocket landing success has improved over time.
  - Launch site **KSC LC-39A** shows highest success rate.
  - Orbits **ES-L1, GEO, HEO, SSO** achieved 100% success.
- **Visual Aanlytics**
  - Most launch sites are near the equator and coastal area
  - Interactive maps reveal geographic clustering of successful launches
- **Predictive Performance**
  - All models performed comparably, except for the Decision Tree.
  - Confusion matrix confirms classification accuracy across models

INTRODUCTION

SPACEX

Project Background

- IBM Applied Data Science Capstone simulates working as a data scientist for a startup competing with SpaceX
- Focus: Predicting Falcon 9 first-stage rocket landing success
- Data from SpaceX API + Wikipedia (payloads, boosters, launch sites, outcomes)
- Applied full **data science lifecycle**: collection → wrangling → analysis → visualization → modeling

Problems to Answer

- **Cost Efficiency** → How much do reusable rockets reduce launch costs?
- **Landing Success Factors** → Which variables (payload, site, booster) drive success?
- **Predictive Modeling** → Can ML models accurately forecast landing outcomes?
- **Business Insights** → How can predictions guide investment, planning, and risk management?

Section 1

Methodology

Methodology – Executive Summary

1. **Data Acquisition**
   - Retrieved launch records from the **SpaceX Open Source REST API**
   - Scraped supplementary data from the Wikipedia page *List of Falcon 9 and Falcon Heavy Launches*

2. **Data Wrangling & Preparation**
   - Filtered and cleaned raw data to remove missing or irrelevant entries
   - Applied **One Hot Encoding** to transform categorical variables for machine learning compatibility

3. **Exploratory Data Analysis (EDA)**
   - Conducted statistical analysis using **SQL queries**
   - Visualized trends across payload mass, launch sites, flight numbers, and orbital destinations

4. **Interactive Visual Analytics**
   - Built dynamic dashboards using **Folium** for geospatial mapping and **Plotly Dash** for interactive charts

5. **Predictive Modeling & Evaluation**
   - Developed classification models to forecast landing outcomes
   - Algorithms used: **Logistic Regression**, **K-Nearest Neighbors**, **Support Vector Machines**, and **Decision Tree**
   - Tuned hyperparameters and evaluated model performance to identify the most effective approach

Data Collection

Primary Data Source:

SpaceX REST API – Queried structured launch data (e.g., rocket, payloads, launchpad, cores)

Ensured **reproducibility** via documented endpoints

Supplementary Data Source:

Static JSON snapshot – Used for consistent results across learners

URL: https://cf-courses-data...API_call_spacex_api.json

Initial Data Wrangling Steps:

Imported libraries: requests, pandas, numpy, datetime

Normalized JSON into a Pandas DataFrame

Subset columns: rocket, payloads, launchpad, cores, flight_number, date_utc

Filtered rows with single core and payload

Converted date to datetime.date and restricted to pre–Nov 13, 2020

GitHub Link to: 1-jupyter-labs-spacex-data-collection-api.ipynb

API Enrichment Functions:

getBoosterVersion() → Extracts **booster name** from rocket ID

getLaunchSite() → Extracts **site name**, **longitude**, **latitude**

getPayloadData() → Extracts **mass** and **orbit**

getCoreData() → Extracts **landing outcome**, **reuse info**, **serial**, **block**, etc.

Final Output (Part 1):

Created enriched DataFrame data_enriched with 17 columns

Filtered to **Falcon 9 only** → data_falcon9

Imputed missing PayloadMass with **mean value**

Retained None for LandingPad when unavailable

GitHub Link to: 2-jupyter-labs-webscraping.ipynb

SpaceX Data Collection Pipeline

API Source → Scraped Source → Unified Data Set

API Source

- SpaceX REST API
- Request SpaceX launch data
- As JASON
- Use API endpoint

Scraped Source

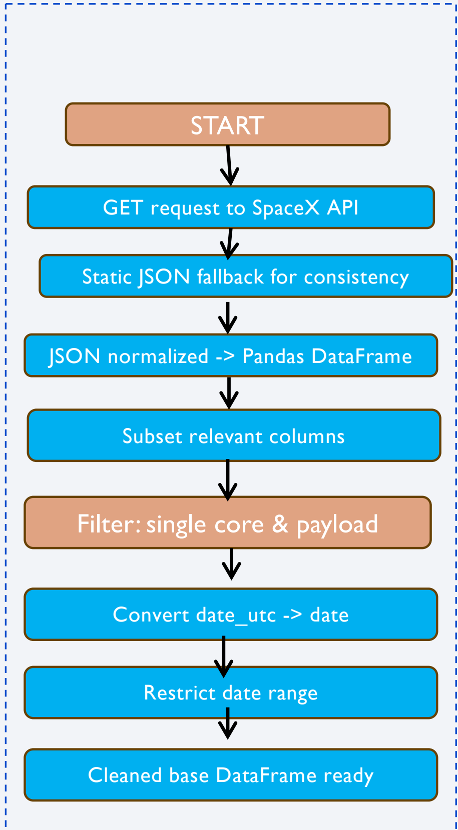
- Wikipedia launch table
- Use BeautifulSoup HTML Parser
- Request URL and Parse Tables

Unified Data Set

- Merge API and scaped data
- Clean and transform DataFrame
- Run and validated.

Data Collection – SpaceX API

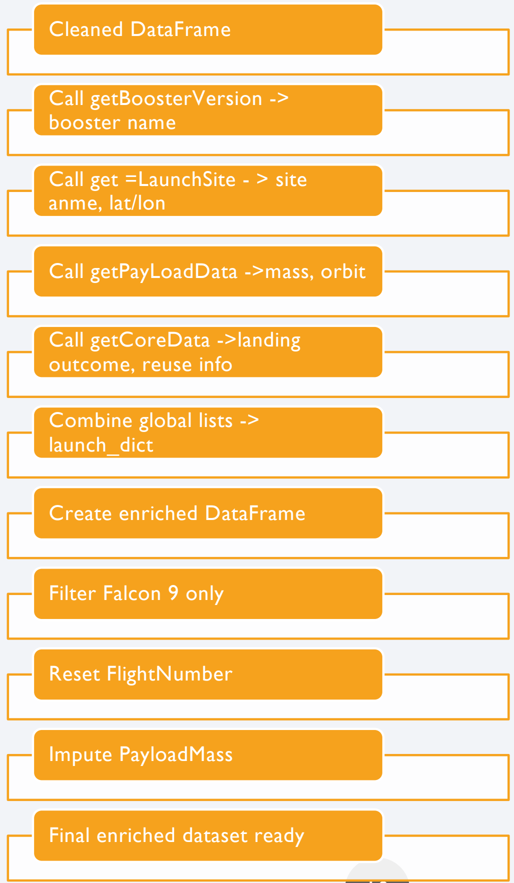
- **Part 1: Raw Data Acquisition and Normalization**
- **Key Phrases:**
- GET request to SpaceX REST API
- Static JSON snapshot for reproducibility
- json_normalize → Pandas DataFrame
- Subset columns: rocket, payloads, launchpad, cores
- Filter: single-core, single-payload launches
- Convert date_utc → datetime.date
- Restrict to launches before 2020-11-13



Part 2: Enrichment via ID-Based REST Calls
Key Phrases:

- rocket → getBoosterVersion() → booster name
- launchpad → getLaunchSite() → site name, lat/lon
- payloads → getPayloadData() → mass, orbit
- cores → getCoreData() → landing outcome, reuse info
- Global lists → launch_dict → enriched DataFrame
- Filter Falcon 9 only → data_falcon9
- Reset FlightNumber sequence
- Impute PayloadMass with mean
- Retain None for LandingPad

GitHub - [GitHub Link to: 1-jupyter-labs-spacex-data-collection-api.ipynb]



Data Collection - Scraping

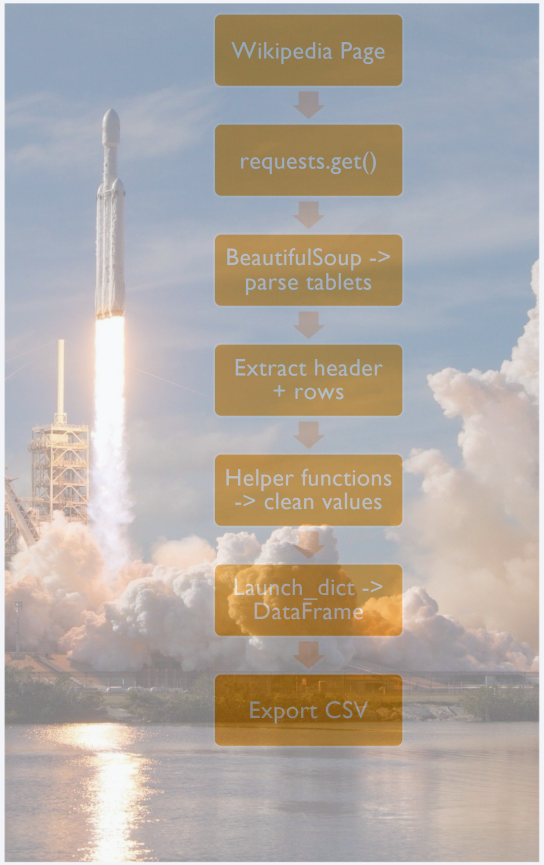
SpaceX Falcon 9 – Web Scraping Process

Part 1: Request & Parse Page

- **requests.get()** → Wikipedia snapshot (June 2021)
- **BeautifulSoup** → parse HTML content
- **find_all('table')** → locate launch records table
- Extract **column names** from <th>

Part 2: Build DataFrame

- Initialize **launch_dict** with empty lists
- Loop through table rows → parse with helper functions
  - date_time() → Date, Time
  - booster_version() → Booster type
  - get_mass() → Payload mass
  - landing_status() → Landing outcome
- Append values → **launch_dict**
- Convert to **Pandas DataFrame**
- Export → spacex_web_scraped.csv

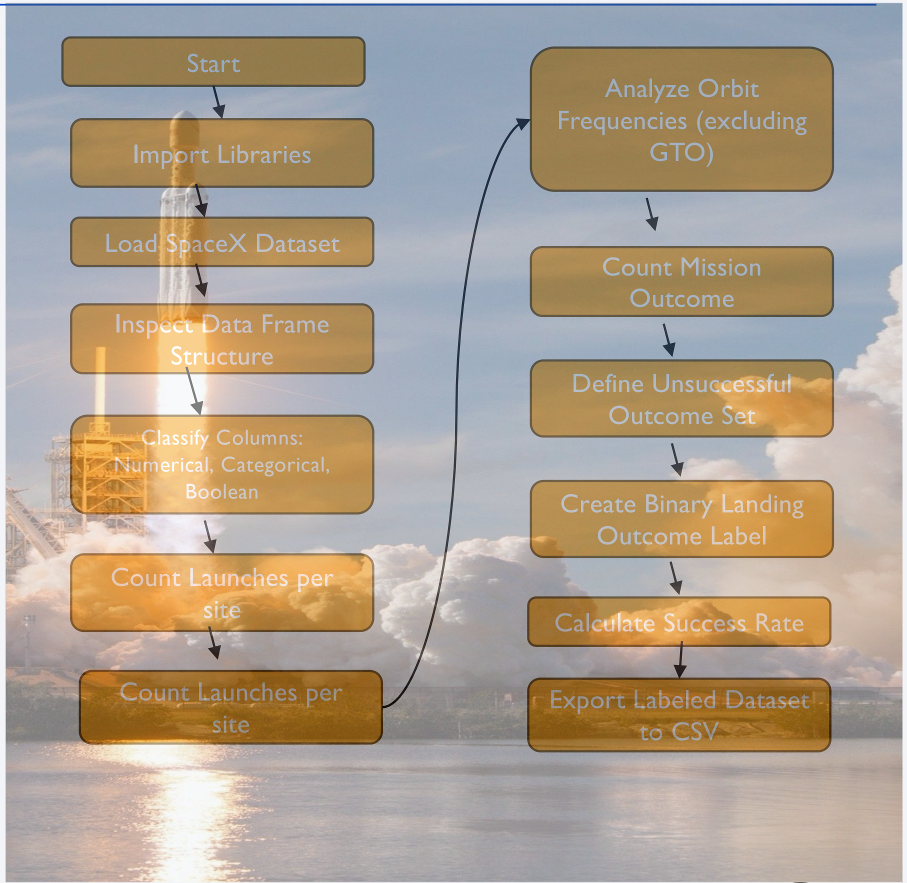


- GitHub URL - https://github.com/KumMelb/Data-Science-Capstone---xSpace/blob/7483eb42d7d36690fa83dfbef8ca133344d81fb0/2-jupyter-labs-webscraping.ipynb

Data Wrangling

Data Processing in Brief

- **Import libraries**: pandas and numpy for analysis.
- **Load dataset**: read CSV from IBM cloud storage.
- **Check missing values**: identify incomplete columns (e.g., LandingPad ~29% missing).
- **Classify columns**: numerical, categorical, and boolean features.
- **Aggregate launches**: count launches per site.
- **Analyze orbits**: tabulate orbit frequencies (excluding GTO).
- **Map outcomes**: list mission results (True/False/None for ASDS, RTLS, Ocean).
- **Create labels**: assign 1 for successful landings, 0 for failures.
- **Calculate success rate**: ~66.7% overall success, 41 drone ship landings.
- **Export dataset**: save processed data with labels to CSV for modeling.
- You need to present your data wrangling process using key phrases and flowcharts
- GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose as follows -
- [Data wrangling.ipynb]



Data Wrangling

Lab 2: Data Wrangling – Falcon 9 Launch Analysis

Part 1: Orbit Analysis

- **Orbit counts** → df['Orbit'].value_counts()
- Exclude **GTO** (transfer orbit, not geostationary)
- Identify frequency of orbits (LEO, ISS, SSO, GEO, etc.)

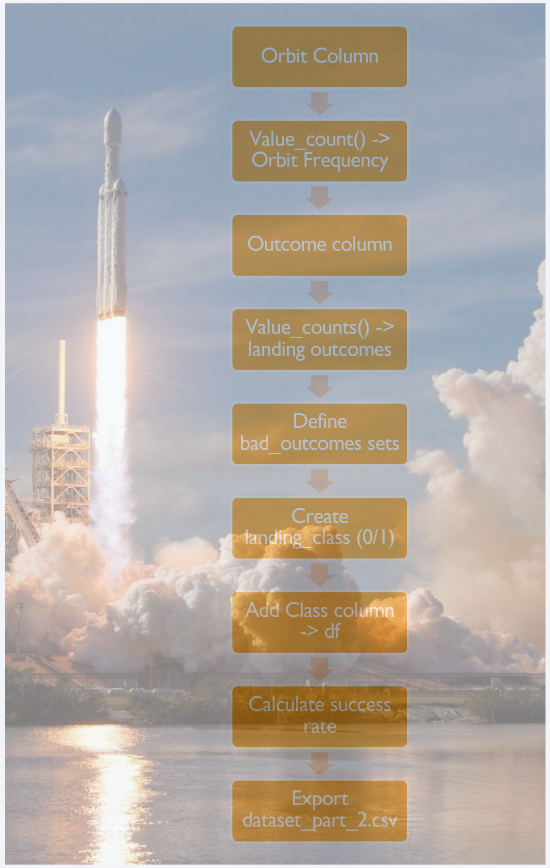
Part 2: Mission Outcomes

- **Outcome counts** → df['Outcome'].value_counts()
- Categories:
  - **True Ocean / False Ocean**
  - **True RTLS / False RTLS**
  - **True ASDS / False ASDS**
  - **None ASDS / None None** → failure to land

Part 3: Create Training Labels

- Define **bad_outcomes set** → unsuccessful landings
- Generate landing_class:
  - 0 = unsuccessful
  - 1 = successful
- Add new column → df['Class']
- Calculate **success rate** → df['Class'].mean()
- Export cleaned dataset → dataset_part_2.csv

[Orbit column]
↓
[value_counts() → orbit frequencies]
↓
[Outcome column]
↓
[value_counts() → landing outcomes]
↓
[Define bad_outcomes set]
↓
[Create landing_class (0/1)]
↓
[Add Class column → df]
↓
[Calculate success rate]
↓
[Export dataset_part_2.csv]



EDA with Data Visualization

Summary of Plotted Charts and Their Purpose

- yearly_success_trend.png: Shows the trend of successful launches over time to assess improvement or consistency in mission outcomes. GitHub link -
- payloadmass_vs_orbit.png: Explores how payload mass varies across different orbits, helping identify orbit-specific payload constraints.
- flightnumber_vs_orbit.png: Examines the relationship between flight sequence and orbit type, potentially revealing strategic shifts over time.
- success_rate_by_orbit.png: Compares success rates across orbit categories to evaluate reliability and risk.
- payloadmass_vs_launchsite.png: Investigates payload mass distribution across launch sites, useful for site capability analysis.
- flightnumber_vs_launchsite.png: Tracks launch activity over time per site, indicating operational frequency and growth.
- flight_payload_plot.png: Likely a composite or scatter plot showing flight number versus payload mass, useful for trend detection and anomaly spotting.
- Link of the plots mentioned above, on the following slide. Download the file and open in browser.

Link to the plots on GitHub

Yearly Success Trends

Payloadmass_vs_orbit.png

Flightnumber_vs_orbit.png

Success_rate_by_orbit.png

Payloadmass_vs_launchsite.png

Flightnumber_vs_launchsite.png

Flight_payload_plot.png

EDA with Data Visualization

- GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- [5-edadataviz.ipynb]

EDA with SQL

Summarize the SQL queries performed

- **Displayed unique launch site names** using SELECT DISTINCT Launch_Site FROM SPACEXTABLE.
- **Filtered 5 records** where launch sites begin with 'CCA' using LIKE 'CCA%'.
- **Calculated total payload mass** for boosters launched by 'NASA (CRS)' using SUM(PAYLOAD_MASS__KG_).
- **Computed average payload mass** for booster version 'F9 v1.1' using AVG(PAYLOAD_MASS__KG_).
- **Identified earliest successful ground pad landing date** using MIN(Date) with a filter on Landing_Outcome.
- **Listed booster versions** with successful drone ship landings and payload mass between 4000–6000 using BETWEEN.
- **Counted mission outcomes** grouped by Mission Outcome using GROUP BY and COUNT(*).
- **Selected booster versions** with maximum payload mass using a subquery with MAX(PAYLOAD_MASS__KG_).
- **Extracted month, landing outcome, booster version, and launch site** for failed drone ship landings in 2015 using substr(Date, 6, 2) and substr(Date, 0, 5).
- **Ranked landing outcomes by count** between two dates using BETWEEN, GROUP BY, and ORDER BY DESC.
- GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose –
- 4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

- The table summarize the SQL queries performed
- GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose is as in the link below –
- 4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Task	Objective	SQL Operation
1	Display unique launch sites	SELECT DISTINCT Launch_Site
2	Show 5 records from launch sites starting with "CCA"	WHERE Launch_Site LIKE 'CCA%' LIMIT 5
3	Total payload mass by NASA (CRS)	SELECT SUM(PAYLOAD_MASS__KG_) WHERE Customer = 'NASA (CRS)'
4	Average payload mass for booster version F9 v1.1	SELECT AVG(PAYLOAD_MASS__KG_) WHERE Booster_Version = 'F9 v1.1'
5	First successful ground pad landing date	SELECT MIN(Date) WHERE Landing_Outcome = 'Success (ground pad)'
6	Boosters with successful drone ship landings and payloads 4000–6000 kg	WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
7	Count of mission outcomes (success/failure)	GROUP BY Mission_Outcome
8	Boosters that carried the maximum payload mass	WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(...))
9	Failed drone ship landings in 2015 with month, booster, and site	WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome = 'Failure (drone ship)'
10	Rank landing outcomes between 2010–2017	GROUP BY Landing_Outcome ORDER BY COUNT(*) DESC

Build an Interactive Map with Folium

Task 1: Mark all launch sites

We created a **folium map** centered at NASA Johnson Space Center.
Added **circles** at each launch site coordinate (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E).
Each circle included a **popup label** with the site name.
Added **text markers** using DivIcon to display site names directly on the map.
Finding: All launch sites are near the coastline and relatively close to the equator compared to other U.S. regions, which is advantageous for orbital launches.

Task 2: Mark success/failed launches

We enhanced the map by visualizing launch outcomes.
Created a **MarkerCluster** to group overlapping markers.
For each launch record, added a **rocket icon marker**:

- **Green** → Successful launch (class=1)
- **Red** → Failed launch (class=0)

Each marker displayed the launch site name in a popup.
Finding: From the clustered markers, it is clear that some sites (e.g., KSC LC-39A) show higher success rates, while others have more mixed outcomes.

Task 3: Calculate distances to proximities

We analyzed proximities of launch sites to nearby features.
Added the **MousePosition plugin** to capture coordinates interactively.
Implemented the **Haversine formula** to calculate distances in kilometers.
Plotted **markers with distance labels** (e.g., "2.50 KM") for coastline, highway, and railway points.
Connected launch sites to these proximities using **colored polylines** (red for highway, green for railway, blue for coastline).
Findings:
Launch sites are within a few kilometers of **railways** and **highways**, supporting logistics and transport.
They are very close to the **coastline**, allowing rockets to launch safely over the ocean.
Sites maintain a **buffer distance from cities**, balancing accessibility with public safety.

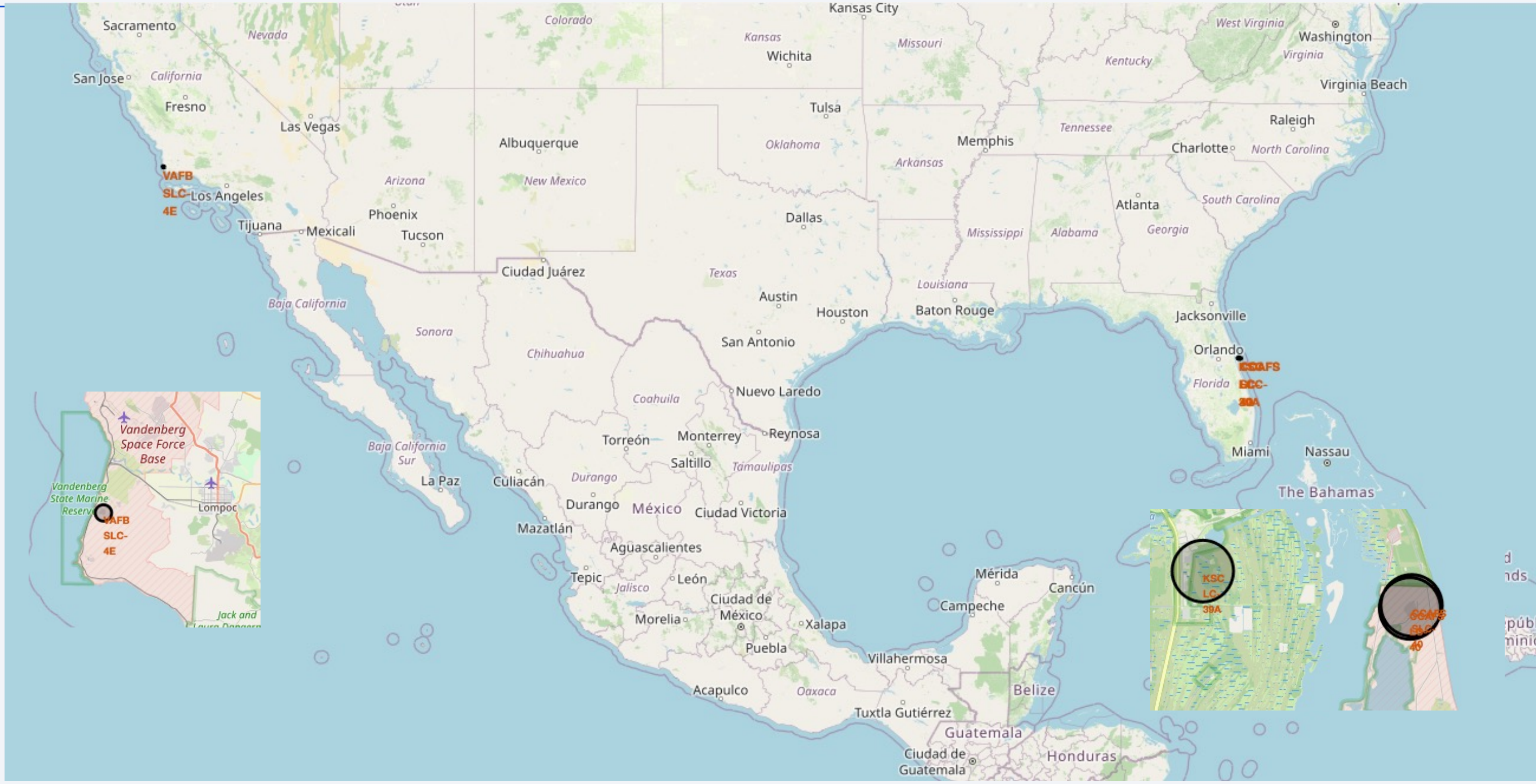
Overall Conclusion

Across all tasks, the maps reveal that **SpaceX launch sites are strategically located**:
Near coastlines for safety,
Near railways and highways for logistics,
Away from dense urban centers to reduce risk,
Positioned closer to the equator to maximize launch efficiency.
This demonstrates how geography plays a critical role in launch site selection, ensuring both **operational efficiency** and **public safety**.

GitHub URL of the completed interactive Folium as follows:

[Lunch Sites](#) (right click -> hyperlink -> open. Download the file and view the interactive map on your browser.

Build an Interactive Map with Folium



Build an Interactive Map with Folium

Launch Site Circles: Each site (e.g., CCAFS LC-40, KSC LC-39A) was marked with a folium.Circle to highlight its location.

Reason why the sites are near coast and at equatorial regions –

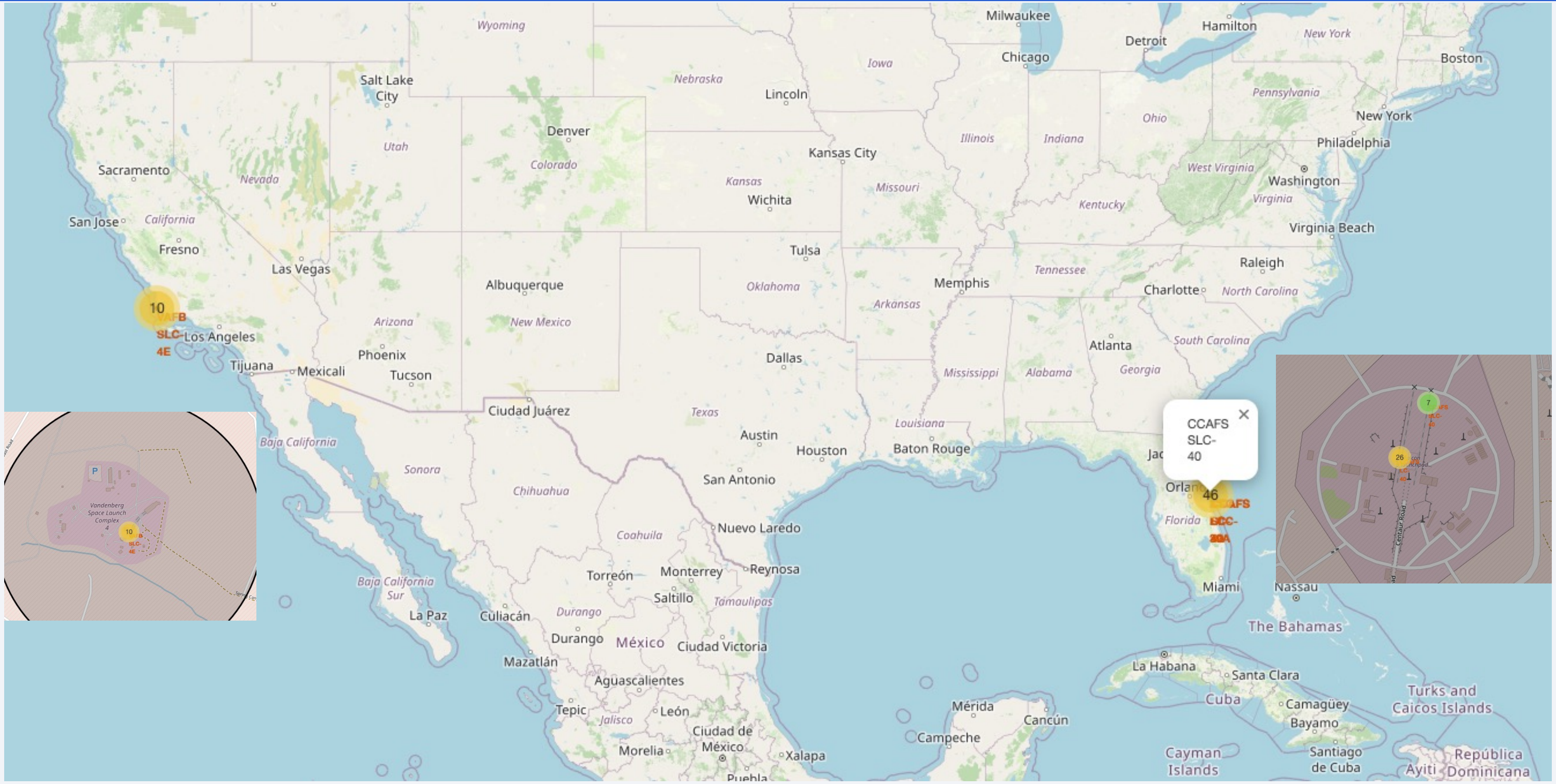
Safety Considerations

- **Debris risk mitigation**: If a rocket fails during launch or ascent, falling debris is more likely to land in the ocean rather than populated areas, minimizing harm to people and infrastructure.
- **Controlled recovery zones**: Coastal launches allow for designated splashdown areas for reusable stages or capsules, which are easier to manage in open water.

Trajectory and Performance Benefits

- **Eastward launches over water**: Most rockets launch eastward to take advantage of Earth's rotation. Coastal sites on eastern shores (like Florida's Cape Canaveral) allow rockets to fly over the ocean, avoiding land masses while gaining extra velocity.
- **Equatorial proximity**: Sites closer to the equator benefit from higher rotational speed, which provides a natural boost to payloads heading into orbit. Coastal equatorial sites like Kourou (French Guiana) and Sriharikota (India) are strategically chosen for this reason
- GitHub URL of the completed interactive Folium as follows:
- Lunch Sites (right click -> hyperlink -> open. Download the file and view the interactive map on your browser.

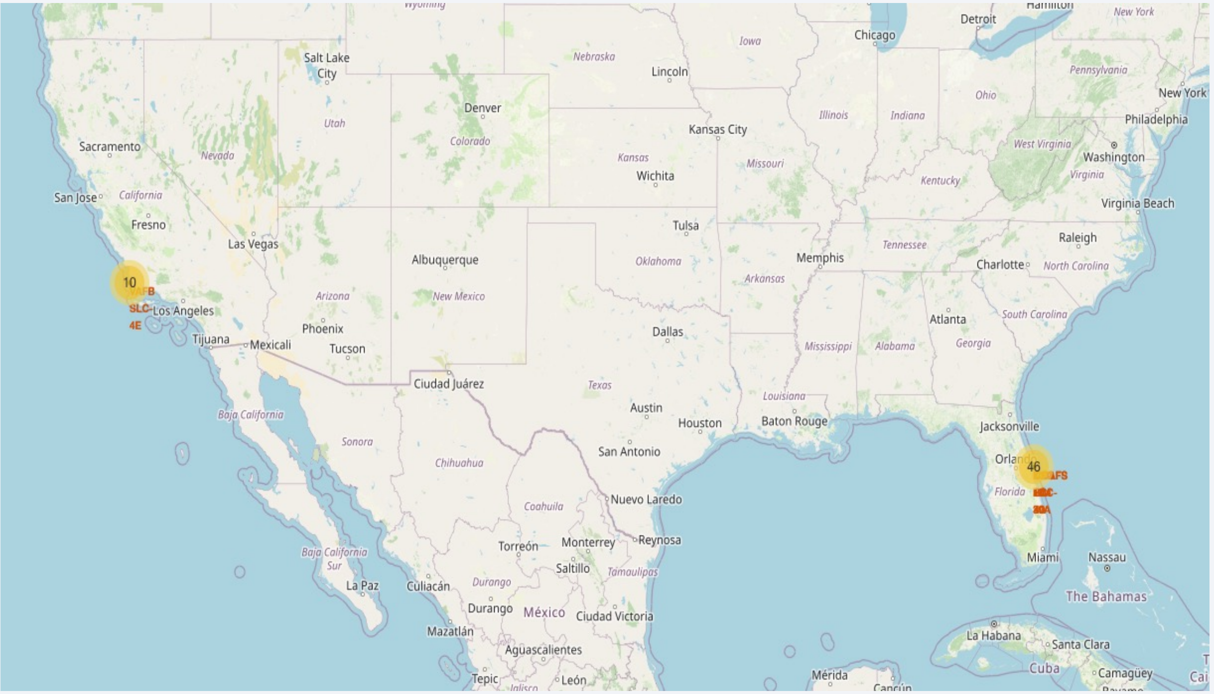
Success / Failed Launches For Each Site

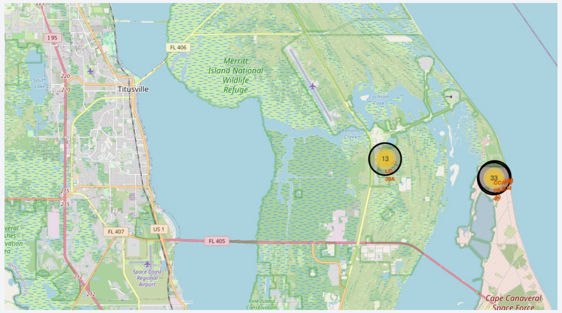


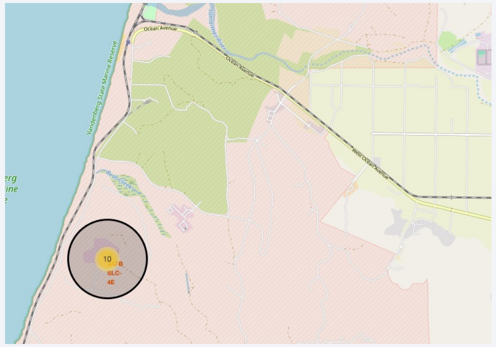
Success / Failed Launches For Each Site

- This interactive Folium map shows the statistics on the successful launch from the site. From the map we can see that the Florida site "KSC LC 39A" as the most successful lunch site.
- The GitHub URL of the completed interactive Success / Failed Launches of the Folium map is as follows:
- [Success / Failed Lauches for each site]

Launch Site and its Proximities



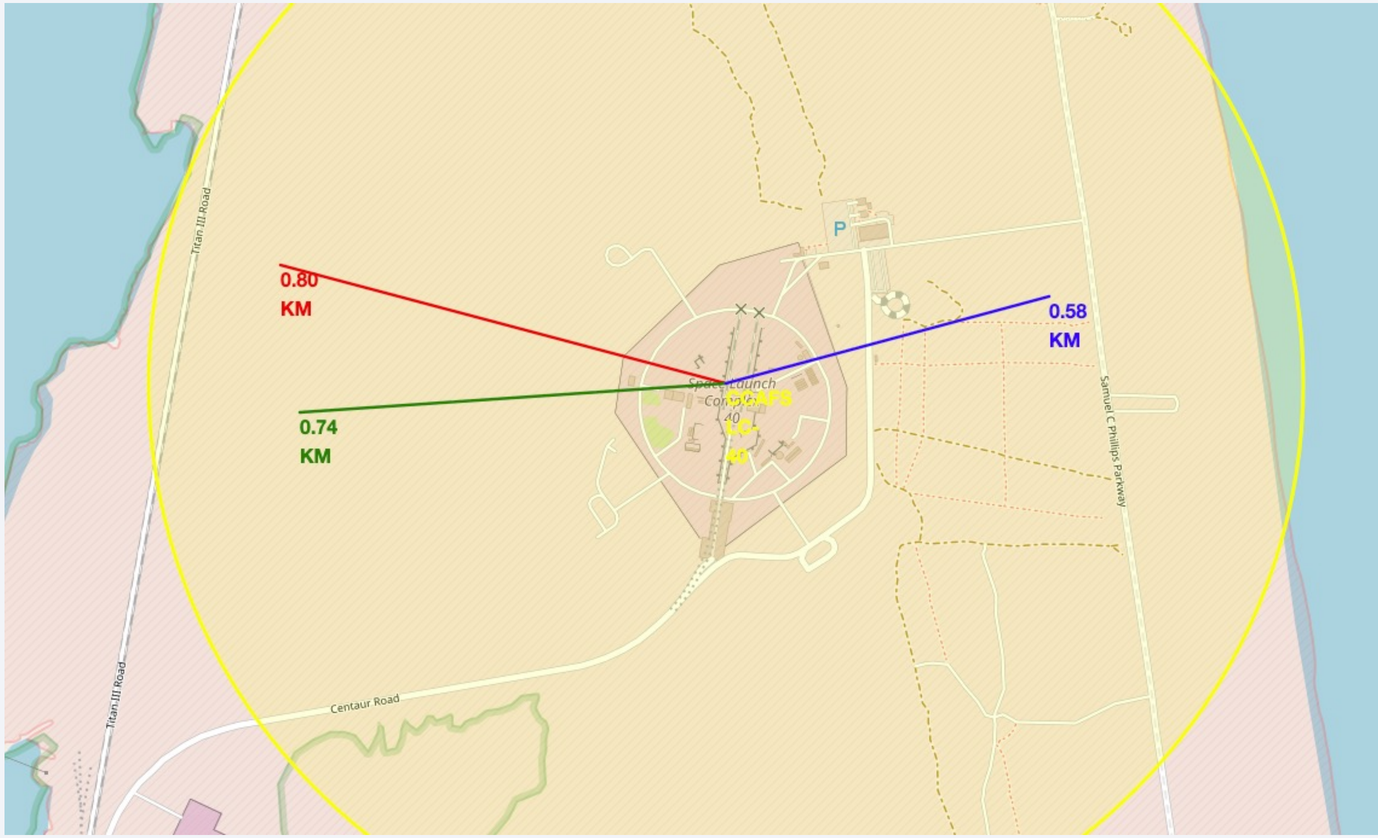




Launch Site and its Proximities

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Space rocket launch sites in the USA are often located near the sea and connected to roads, railways, and other infrastructure because it maximizes safety, efficiency, and logistics. Oceans provide safe zones for falling debris, while transport links allow heavy rocket components and fuel to be delivered smoothly.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- Distance Map – To view the interactive map, download the file from GitHub and open in a browser.

Launch Site and its Proximities



Build an Interactive Map with Folium

Distance Markers (0.58 km, 0.74 km, 0.80 km)

- These lines likely represent **safety perimeters or exclusion zones**.
- They help define how far critical infrastructure or personnel must be from the launch pad during liftoff to avoid damage or injury from heat, shockwaves, or debris.

Roads: Titan III Road, Centaur Road, Samuel C Phillips Parkway

- These roads are named after historic rocket programs and provide **essential access routes** for:
  - Transporting rocket components and fuel.
  - Moving personnel and emergency vehicles.
  - Connecting to broader logistics networks (rail, highway).

GitHub URL of the completed Infrastructure proximity interactive Folium map.

Proximity map – download the file from GitHub and open on your browser to view the Proximity Interactive Map.

Build a Dashboard with Plotly Dash

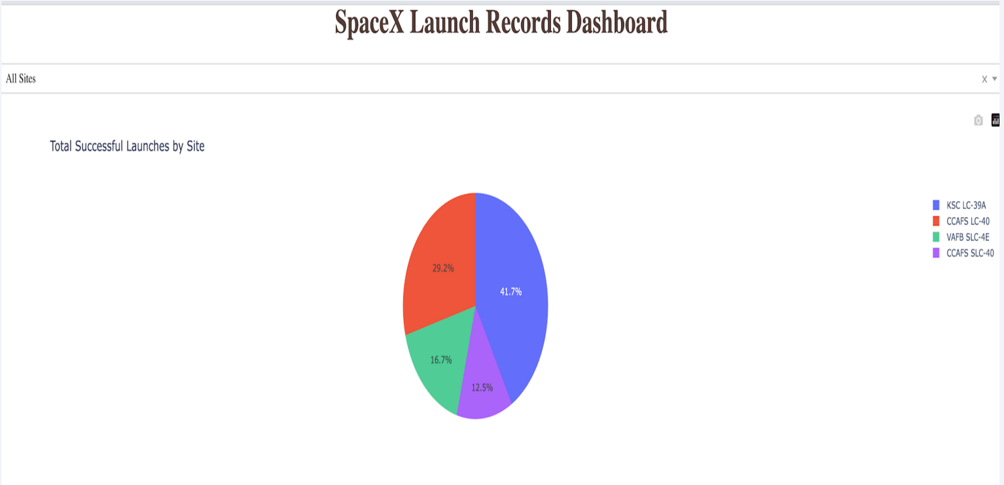
Interaction controls on the Plotly dashboard

- Dropdown List with Lauch Sites – allow the user to select all launch or single launch sites
- Pie Chart Showing Successful Launches – allow user to see the the relative success of the different launch site.
- Slider of the Payload Mass Range – all the user the payload mass range.
- Scatter Chart Showing Payload Mass vs Success Rate by Booster Version – allow the the observe the correlation between the payload and Launh results.

The GitHub URL of the completed Plotly Dash lab, as an external reference and peer-review purpose is as below –

[Spacex-dash-app]

Build a Dashboard with Plotly Dash

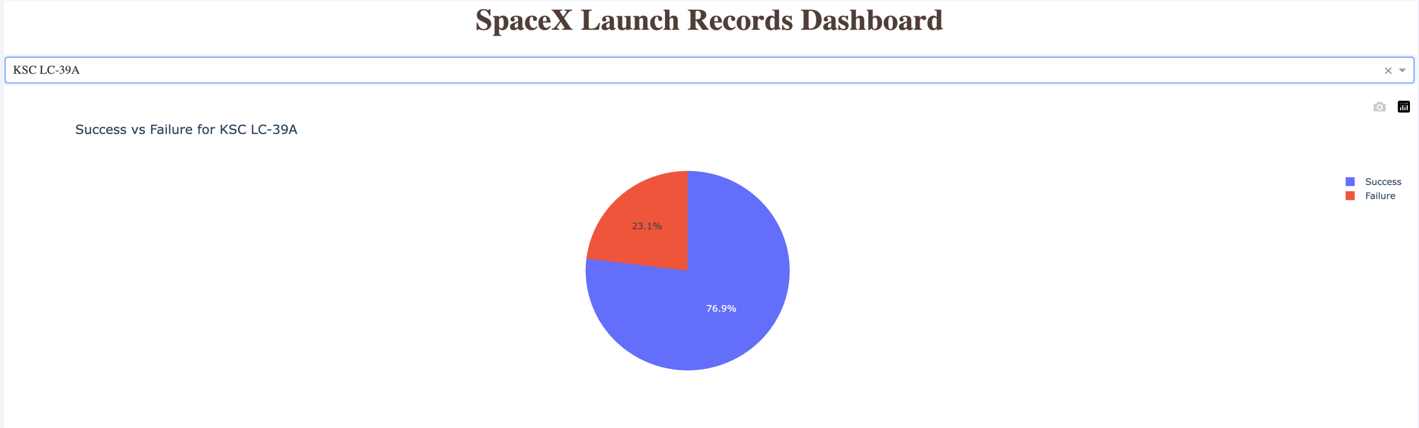


Launch Site	Percentage	Color	Interpretation
KSC LC-39A	41.7%	Blue	Most frequently used site for successful launches
CCAFS LC-40	29.2%	Red	Second most used site, also located in Florida
VAFB SLC-4E	16.7%	Green	West Coast site, used for polar orbit missions
CCAFS SLC-40	12.5%	Purple	Least used among the four shown

This pie chart visualizes the proportion of successful SpaceX launches from four different launch sites. Each slice represents a percentage of the total successful launches

- **KSC LC-39A** (Kennedy Space Center) is the dominant launch site, likely due to its capability to support heavy-lift missions and crewed flights.
- **CCAFS LC-40** (Cape Canaveral) also plays a major role, especially for commercial and satellite launches.
- **VAFB SLC-4E** (Vandenberg Air Force Base) is used for missions requiring polar or sun-synchronous orbits.
- GitHub link - [Plotly Dashboard - All Sites Successful.png]

Build a Dashboard _ SpaceX success



The pie chart filtered from the drop list for “KSC LC-39A” , which is the site used by SpaceX abd Falcon 9 Heavy missions.

The success rate at 76.9%, indicates that the majority pf SpaceX launches were successful.

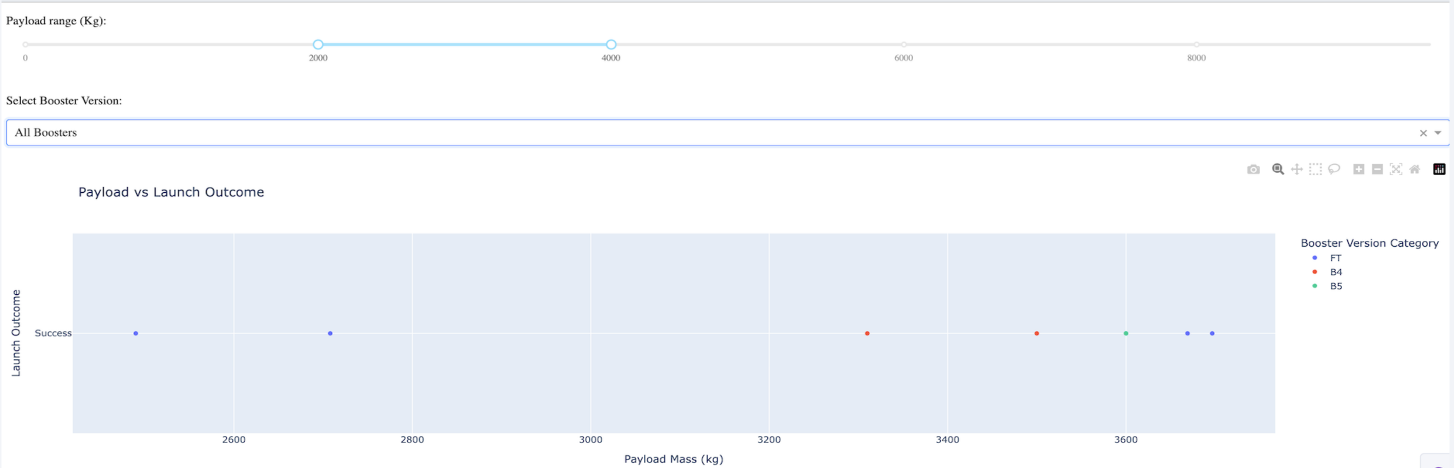
The Failure rate in Red segment of chart, shows the SpaceX failure rate of 23.1%.

GitHub link - Plotly Dashboard - SpaceX Launch Result Pie Chart.png

Build a Dashboard – All Payloads and Success

- This dashboard visualizes the relationship between **payload mass** and **launch outcome** for rocket launches, with an interactive interface that allows filtering by payload and booster version.
- **Success Clustering**: Most successful launches are concentrated across a wide payload range, especially between **2000 kg and 6000 kg**, indicating reliable performance in this mass window.
- **Failure Distribution**: Failures appear sporadically across the payload spectrum, but are more frequent at **lower payload masses** (<2000 kg), suggesting early-stage or less optimized launches.
- **Booster Version Trends**:
  - **FT (Red)**: Shows a mix of success and failure, with more failures at lower payloads.
  - **B4 (Blue)**: Appears less frequently; outcomes are mixed but lean toward success.
  - **B5 (Green)**: Dominates the successful launches, especially at higher payloads (>4000 kg), indicating **strong reliability and performance**.
- GitHub link - [Plotly Dashboard - All Payloads and Boosters.png]

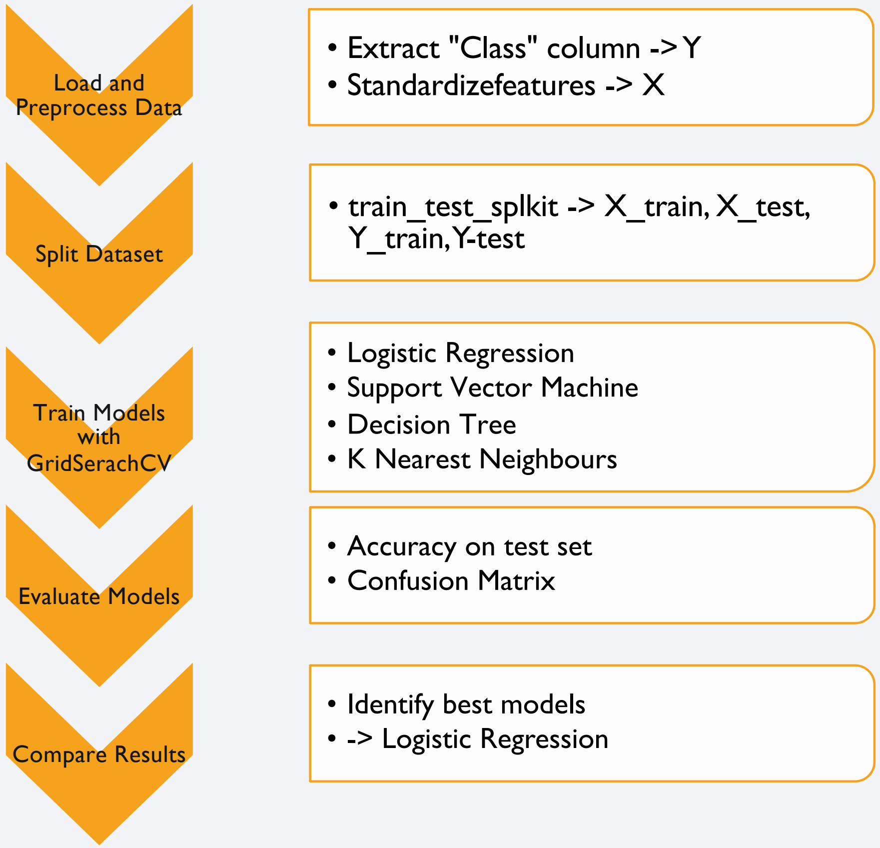
Build a Dashboard with Plotly Dash – Smaller Payload



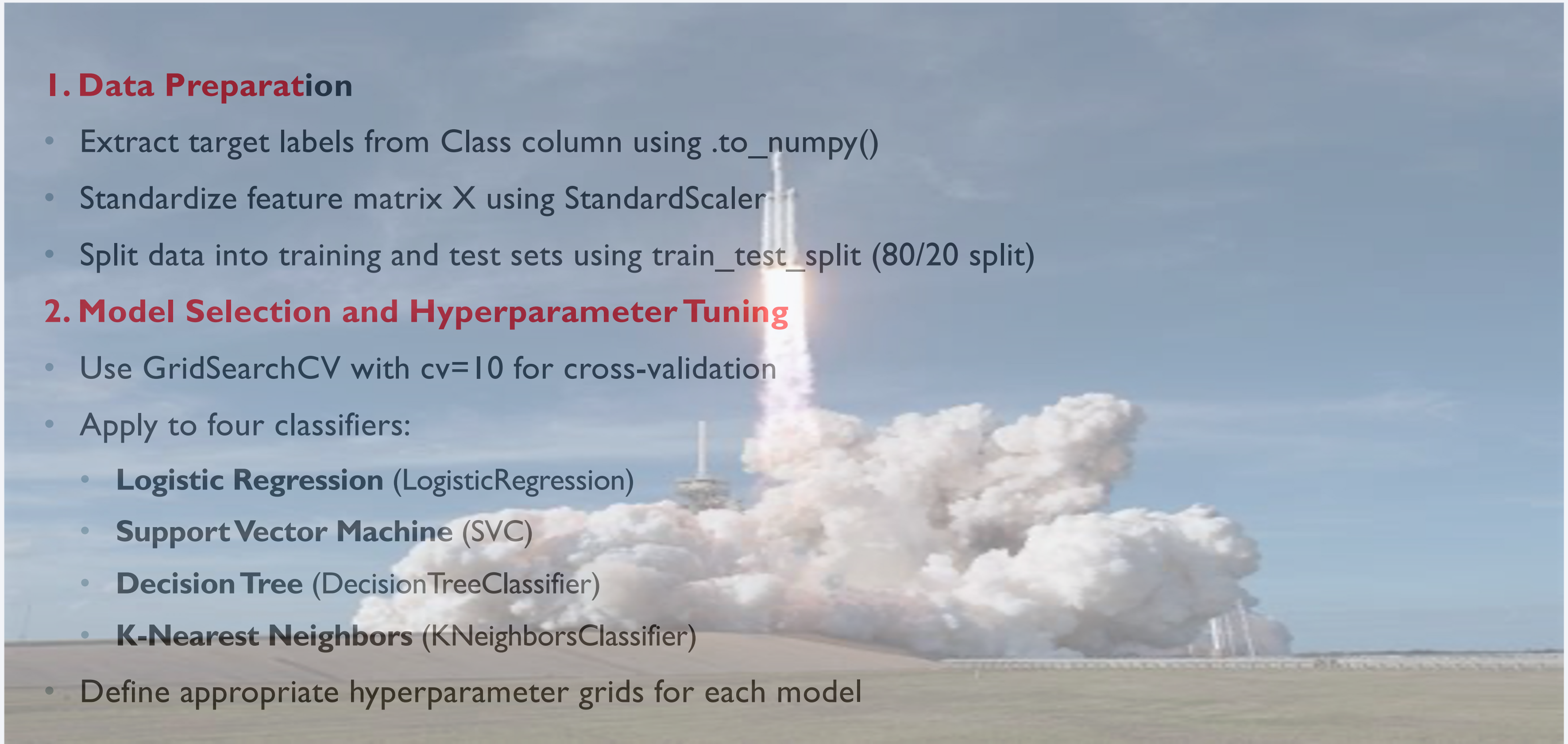
- Using the slider the payload is restricted to the 2000 Kg to 4000 kg range. In this chart "All Boosters" was selected from the "Selected Booster Version".
- Lower payloads show consistent success Most points at the lower end of the payload range cluster around "Success," suggesting lighter payloads are reliably launched.
- Higher payloads introduce variability As payload mass increases, outcomes become more mixed, indicating heavier payloads pose greater risk
- Overall conclusion Success rates are strongly tied to both payload mass and booster version. Later versions (B5) demonstrate improved performance at heavier payloads compared to earlier versions (B4)
- GitHub - [Plotly Dashboard - Selected Payload for all boasters.png]

Predictive Analysis (Classification)

- The summarize on how the classification was built, evaluated, improved, and the best performing classification model selected is on the 2 following slides and flowchart.
- Flowchart of the predictive analysis is as attached.
- The GitHub URL of completed predictive analysis lab, as an external reference and peer-review purpose is as follows –
- [8-SpaceX_Machine Learning Prediction_Part_5.ipynb]



Predictive Analysis (Classification)



1. Data Preparation

- Extract target labels from Class column using .to_numpy()
- Standardize feature matrix X using StandardScaler
- Split data into training and test sets using train_test_split (80/20 split)

2. Model Selection and Hyperparameter Tuning

- Use GridSearchCV with cv=10 for cross-validation
- Apply to four classifiers:
  - **Logistic Regression** (LogisticRegression)
  - **Support Vector Machine** (SVC)
  - **Decision Tree** (DecisionTreeClassifier)
  - **K-Nearest Neighbors** (KNeighborsClassifier)
- Define appropriate hyperparameter grids for each model

Predictive Analysis (Classification)

3. Model Evaluation

- Evaluate each model on test data using .score()
- Plot confusion matrices to visualize prediction errors
- Calculate:
  - **Accuracy**
  - **Jaccard Score**
  - **F1 Score**

4. Model Comparison and Selection

- Compare test set performance across all models
- Identify best-performing model based on accuracy and diagnostic metrics
- Visualize comparison using a bar chart

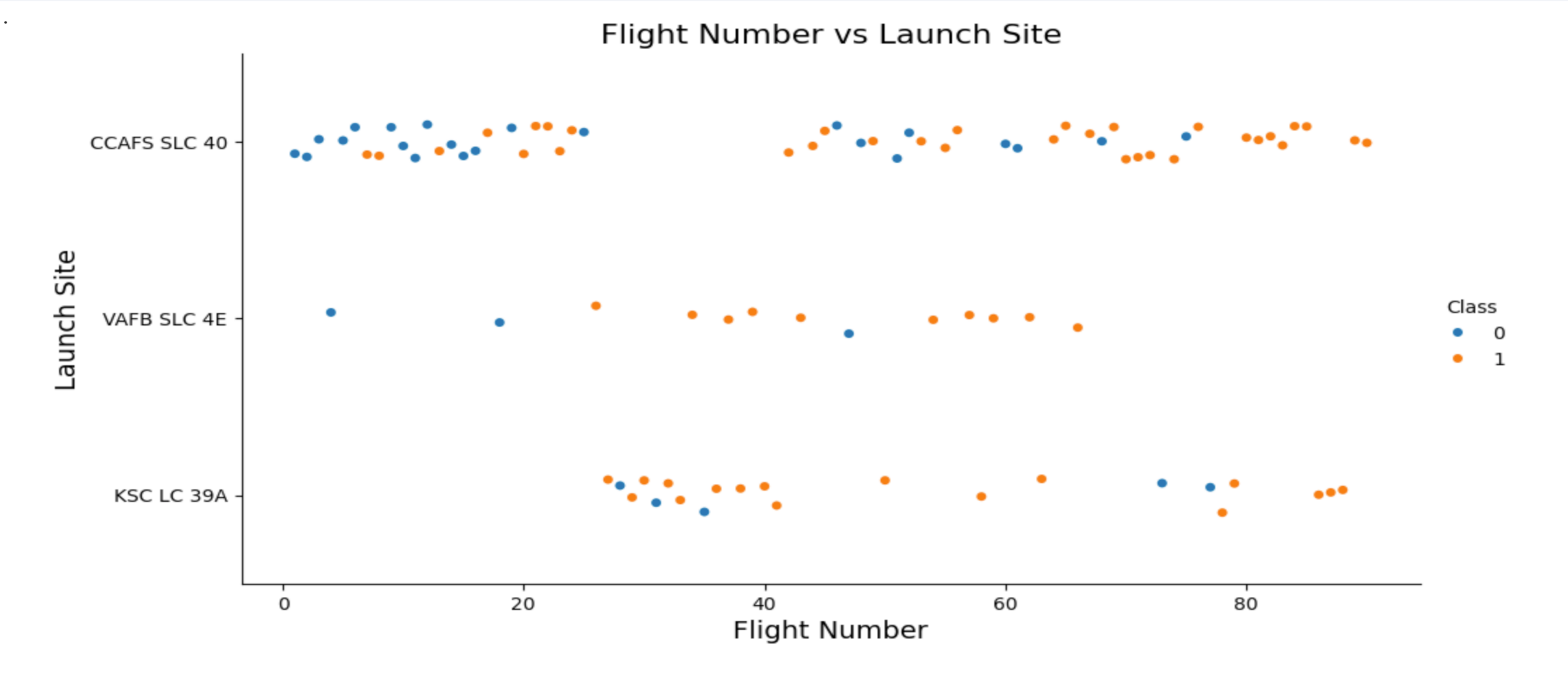
Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site Scatterplot



Flight Number vs. Launch Site

🚀 Key Takeaways

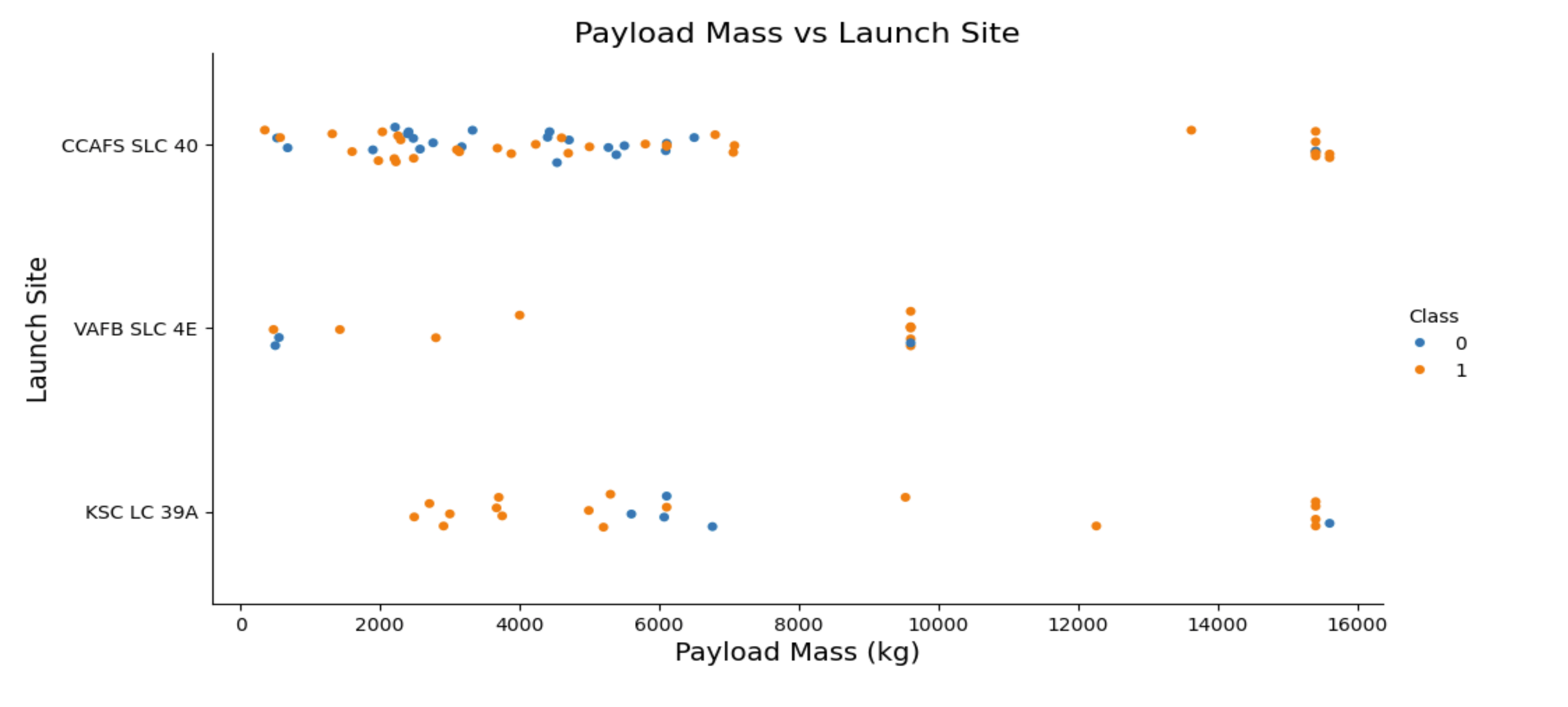
KSC LC 39A: Later flights, almost all **successful (orange)** → strong reliability.
CCAFS SLC 40: Used throughout, mixed **success and failure** → main test site with improving results.
VAFB SLC 4E: Few launches, mostly **unsuccessful (blue)** → limited use, lower success rate.

📈 **Overall Trend**

Early flights = more **failures**.
Later flights = mostly **successes**, especially at **KSC LC 39A**.
Clear progression in reliability as the program matured.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

🧭 **Launch Site Trends**

KSC LC 39A:

Handles the **heaviest payloads**, often exceeding 10,000 kg.

Strong concentration of successful missions, indicating high reliability for large-scale launches.

CCAFS SLC 40:

Supports a **wide range of payload masses**, from light to moderately heavy.

Mixed outcomes suggest it's a versatile site used for varied mission profiles.

VAFB SLC 4E:

Primarily used for **lighter payloads**.

Fewer launches overall, possibly indicating a more specialized or limited role.

📐 **Payload Mass Patterns**

Higher payloads are almost exclusively launched from **KSC LC 39A**, suggesting it's optimized for heavy-lift missions.

Lighter payloads are distributed across all sites, but especially common at **VAFB SLC 4E**.

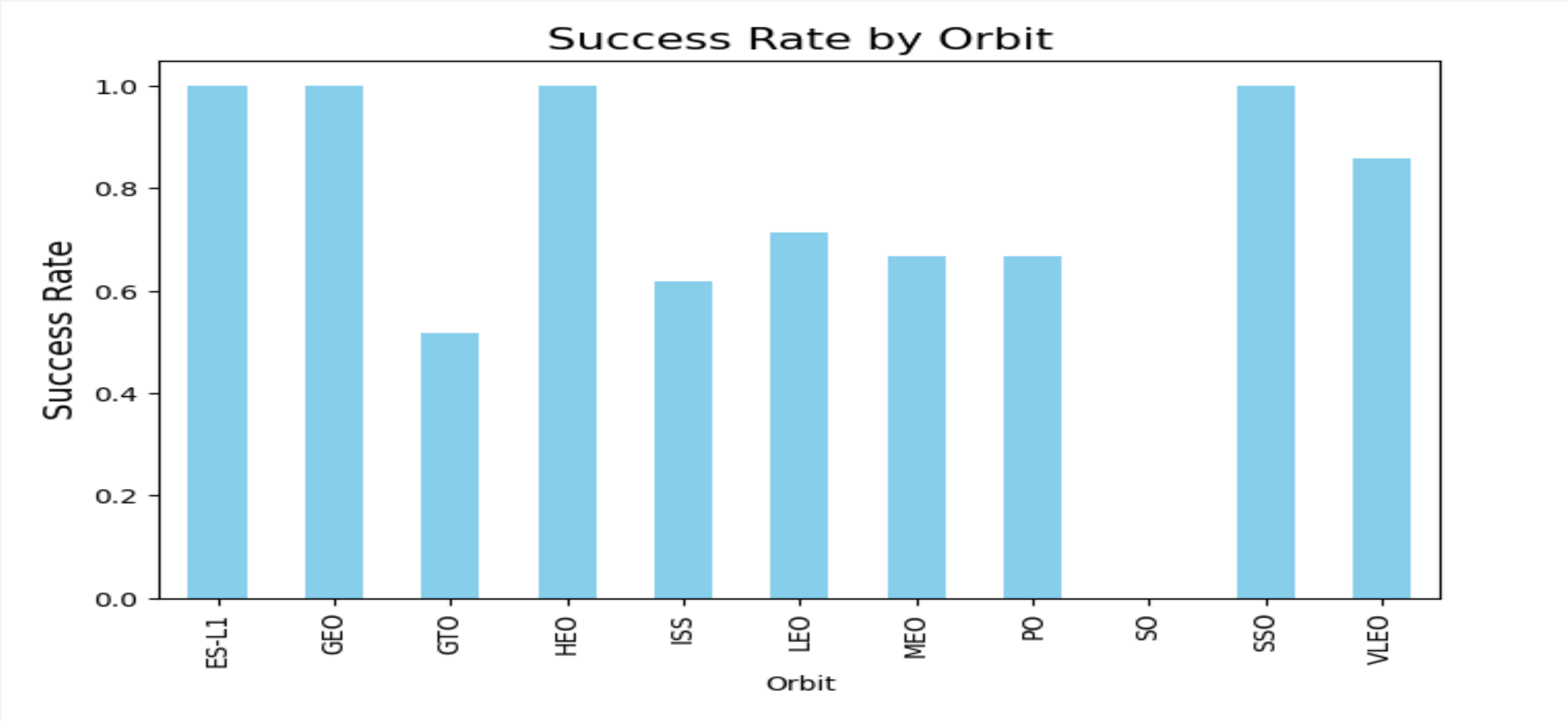
📊 **Distribution Insights**

The vertical spread of points at each site shows the **diversity of missions**.

The horizontal spread (mass) shows how **mission scale varies** by site.

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type



Success Rate vs. Orbit Type

Show the screenshot of the scatter plot with explanations

🚀 **Key Observations**

🏆 **Highest Success Rates**

ES-L1 (Earth–Sun Lagrange Point 1) and **SSO (Sun-Synchronous Orbit)** show **near-perfect success rates**, close to 1.0.

These orbits are often used for scientific and Earth observation missions, which may benefit from mature technology and stable launch conditions.

✅ **Strong Performers**

LEO (Low Earth Orbit), **ISS (International Space Station)**, and **PO (Polar Orbit)** also have **high success rates**, generally above 0.8.

LEO and ISS are common destinations for crewed and cargo missions, suggesting reliable infrastructure and frequent launches.

⚠️ **Moderate Success**

GEO (Geostationary Orbit) and **SO (Solar Orbit)** fall in the mid-range, with success rates around 0.6–0.7.

These missions may involve more complex navigation or longer durations, increasing risk.

❌ **Lowest Success Rate**

GTO (Geostationary Transfer Orbit) has the **lowest success rate**, below 0.5.

GTO is a transitional orbit requiring precise maneuvers to reach GEO, which can introduce more failure points.

📉 **Sparse or Specialized Orbits**

HEO (Highly Elliptical Orbit), **MEO (Medium Earth Orbit)**, and **VLEO (Very Low Earth Orbit)** have fewer data points but moderate success rates.

These orbits are less commonly used, possibly for niche applications like navigation or experimental missions.

📊 **Overall Insight**

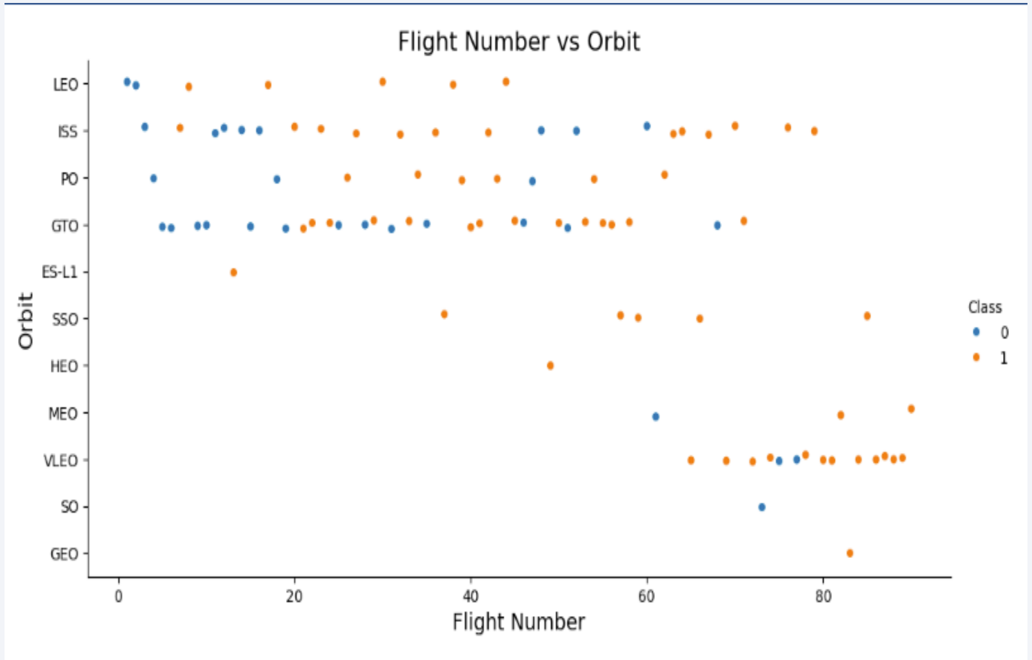
Success rates vary significantly by orbit type, reflecting differences in mission complexity, frequency, and technological maturity. Orbits like ES-L1 and SSO are consistently reliable, while transitional orbits like GTO pose greater challenges.

Craig Bobchin

Flight Number vs. Orbit Type

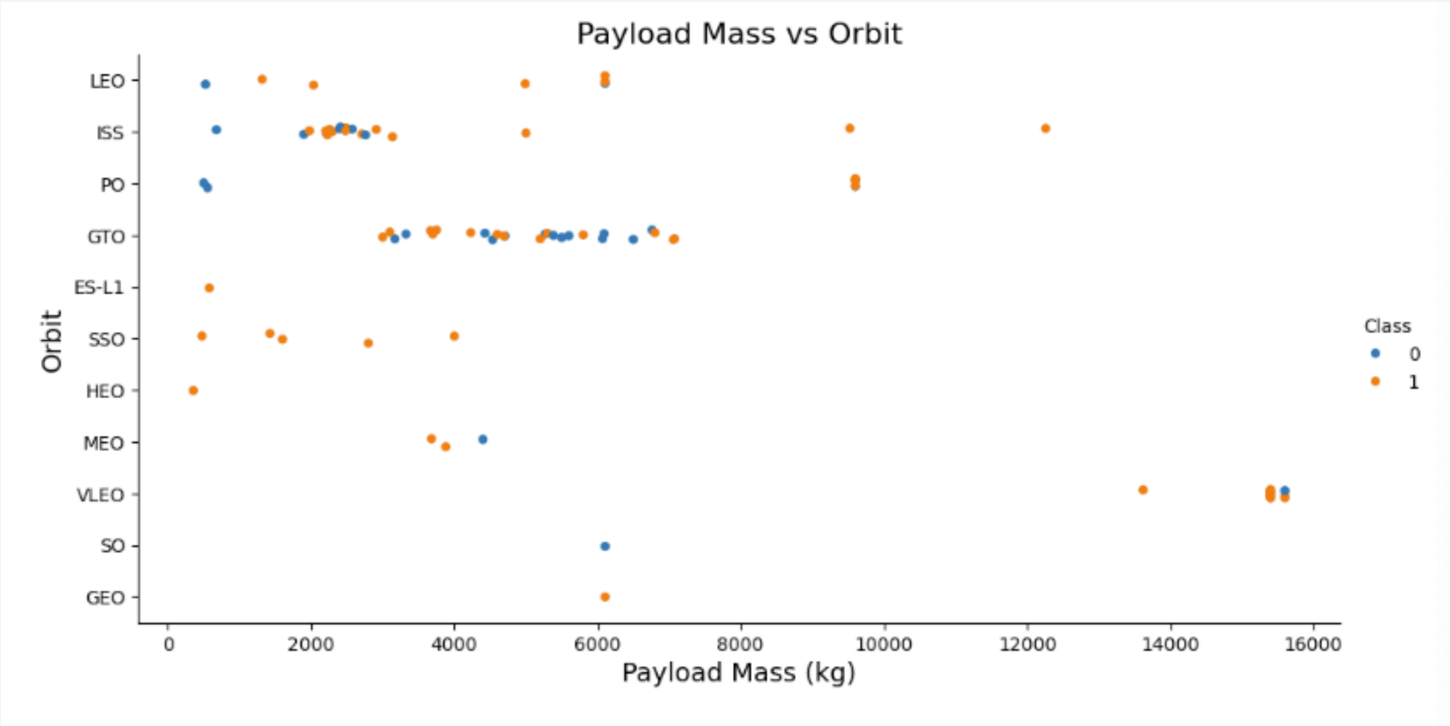
- Show the screenshot of the scatter plot with explanations

Flight Distribution Across Orbits
LEO: Most frequently targeted orbit with a wide spread of flight numbers.
ISS: Consistently targeted with flights appearing across the timeline.
PO (Polar Orbit): Appears sporadically, indicating specialized missions.
GTO: Concentrated in mid-to-late flight numbers, suggesting growing capability.
ES-L1: Rarely used, with only a few missions reaching this deep-space point.
SSO (Sun-Synchronous Orbit): Moderately used, often for Earth observation.
HEO (Highly Elliptical Orbit): Very few missions, likely experimental or niche.
MEO (Medium Earth Orbit): Limited use, possibly for navigation satellites.
VLEO (Very Low Earth Orbit): Rare, possibly for short-duration tech demos.
SO (Solar Orbit): Extremely rare, representing ambitious interplanetary goals.
GEO (Geostationary Orbit): Appears in later flights, showing advanced deployment.



Payload vs. Orbit Type

Show the screenshot of the scatter plot with explanations



Payload vs. Orbit Type

Show the screenshot of the scatter plot with explanations

🚀 **What the Plot Shows**
X-axis (Flight Number): Represents the sequence of launches over time.
Y-axis (Orbit): Lists various orbital destinations such as LEO, ISS, GTO, SSO, etc.
Color Coding (Class): You've confirmed that Class 0 (blue) and Class 1 (orange) represent mission outcomes.
📊 **Key Patterns and Insights**
🛰 **Orbit Usage Over Time**
LEO and ISS appear frequently across a wide range of flight numbers, indicating consistent use throughout the launch history.
GTO is also common but shows a mix of outcomes, aligning with its lower success rate seen in the previous chart.
SSO and ES-L1 are used in later flight numbers, suggesting these orbits became more prominent as the launch program matured.
📈 **Evolution of Mission Complexity**
Early flights are concentrated in **LEO and ISS**, which are simpler and closer orbits.
More diverse orbits like **SSO, ES-L1, and SO** appear in higher flight numbers, implying technological progression and mission expansion.
🎯 **Success Trends by Orbit**
SSO and ES-L1 show mostly orange points (Class 1), reinforcing their high success rates.
GTO has a noticeable number of blue points (Class 0), consistent with its lower reliability.
🧠 **Strategic Implications**
The scatter plot suggests a shift from routine missions (LEO, ISS) to more ambitious orbital targets over time.
It also highlights which orbits are riskier and which are more dependable, useful for future mission planning.
I can also correlate flight number with payload mass or launch site if you'd like to explore multi-dimensional trends.

Craig Bobchin

Launch Success Yearly Trend

- **2010–2013:** Success rate stayed at **0.0**, showing no progress in the early years.
- **2014–2015:** A breakthrough occurred, climbing to **0.35** and holding steady — likely the result of a new strategy or change.
- **2016–2017:** Rapid growth, reaching **0.65** and then peaking at **0.85**, marking the strongest performance period.
- **2018:** A noticeable dip to **0.6**, suggesting challenges or setbacks.
- **2019–2020:** Recovery and stabilization at **0.85** and **0.8**, showing resilience and sustained high success.
- 👉 Overall, the trend highlights a **turning point in 2014**, strong growth through 2017, a temporary setback in 2018, and then a stable high-performance phase.

All Launch Site Names

```
# Task 1
# Display the names of the unique launch sites in the space mission
%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

Explanation:

SELECT DISTINCT ensures that only unique values are returned.

launch_site is the column assumed to contain the names of the launch sites.

launches is the assumed name of the table holding the data.

This query will gives a list of all different launch site names without any duplicates.

Link to the lab as follow -

4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
# Task 2
# Display 5 records where launch sites begin with the string 'CCA'

import pandas as pd

query = "SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;"
df_task2 = pd.read_sql_query(query, con)   # con is your sqlite3 connection
df_task2
```

	Date	Time (UTC)	Booster_Version	Launch_Site	
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft (
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two Cub
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Drago
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	

This query filters the SPACEXTABLE to show only rows where the Launch_Site starts with "CCA", using the LIKE 'CCA%' pattern.
The LIMIT 5 clause ensures that only five matching records are returned.

From your earlier result, we know two launch sites match this pattern:
CCAFS LC-40
CCAFS SLC-40
So the query will return launches from these two sites.

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
# Task 3
# Display the total payload mass carried by boosters launched by NASA (CRS)
query3 = "SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';"
df_task3 = pd.read_sql_query(query3, con)
df_task3
```

	Total_Payload_Mass
0	45596

SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';

🔍 **Explanation**

SUM(PAYLOAD_MASS__KG_) adds up all payload masses for the filtered rows.
WHERE Customer = 'NASA (CRS)' ensures only launches by NASA under the CRS program are included.
The result, 45596, represents the **total payload mass in kilograms** carried by NASA (CRS) boosters across all relevant launches.

Average Payload Mass by F9 v1.1

SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_Mass
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';

Explanation
AVG(PAYLOAD_MASS__KG_) computes the mean payload mass across all matching records.
WHERE Booster_Version = 'F9 v1.1' filters the dataset to include only launches using that specific booster version.
The result, 2928.4, represents the **average payload mass in kilograms** carried by F9 v1.1 boosters.

```
# Task 4
# Display average payload mass carried by booster version F9 v1.1
query4 = "SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';"
df_task4 = pd.read_sql_query(query4, con)
df_task4
```

	Avg_Payload_Mass
0	2928.4

First Successful Ground Landing Date

SELECT MIN(Date) AS First_Successful_Ground_Pad_Landing
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)';

Explanation
MIN(Date) finds the earliest date in the filtered results.
WHERE Landing_Outcome = 'Success (ground pad)' ensures only successful landings on ground pads are considered.
The result, 2015-12-22, marks the **first recorded successful ground pad landing** in the dataset.

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
# Task 5
# List the date when the first successful landing outcome in ground pad was achieved
query5 = "SELECT MIN(Date) AS First_Successful_Ground_Pad_Landing FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';"
df_task5 = pd.read_sql_query(query5, con)
df_task5
```

	First_Successful_Ground_Pad_Landing
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SELECT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome =
'Success (drone ship)'
AND PAYLOAD_MASS__KG_
BETWEEN 4000 AND 6000;

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
# Task 6
# List the names of the boosters which have success in drone ship and payload mass between 4000 and 6000
query6 = "SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;"
df_task6 = pd.read_sql_query(query6, con)
df_task6
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Explanation

Landing_Outcome = 'Success (drone ship)' filters for boosters that successfully landed on a drone ship.
PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 restricts results to missions with payloads in that range.
The result lists four boosters:

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

This query helps identify mid-weight missions with successful drone ship recoveries, useful for analyzing booster reuse and performance. I can also help you build a comparison matrix of booster versions by landing type and payload range.

Total Number of Successful and Failure Mission Outcomes

```
SELECT Mission_Outcome, COUNT(*)
AS Outcome_Count
FROM SPACEXTABLE
GROUP BY Mission_Outcome;
```

Task 7

List the total number of successful and failure mission outcomes

```
# Task 7
# List the total number of successful and failure mission outcomes
query7 = "SELECT Mission_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE GROUP BY Mission_Outcome;"
df_task7 = pd.read_sql_query(query7, con)
df_task7
```

	Mission_Outcome	Outcome_Count
0	Failure (in flight)	1
1	Success	98
2	Success	1
3	Success (payload status unclear)	1

Explanation

COUNT(*) tallies the number of records for each unique Mission_Outcome.

GROUP BY Mission_Outcome aggregates the results by outcome type.

The output shows:

- **Success**: 99 total (though split into two rows—likely due to whitespace or formatting inconsistencies)
- **Success (payload status unclear)**: 1
- **Failure (in flight)**: 1

Boosters Carried Maximum Payload

```
# Task 8
# List all booster_versions that have carried the maximum payload mass
query8 = "SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);"
df_task8 = pd.read_sql_query(query8, con)
df_task8
```

	Booster_Version	PAYLOAD_MASS__KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1049.4	15600
2	F9 B5 B1051.3	15600
3	F9 B5 B1056.4	15600
4	F9 B5 B1048.5	15600
5	F9 B5 B1051.4	15600
6	F9 B5 B1049.5	15600
7	F9 B5 B1060.2	15600
8	F9 B5 B1058.3	15600
9	F9 B5 B1051.6	15600
10	F9 B5 B1060.3	15600
11	F9 B5 B1049.7	15600

Explanation
The subquery SELECT MAX(PAYLOAD_MASS__KG_) finds the highest payload mass in the dataset.
The outer query retrieves all booster versions that match this maximum value.
The result shows **12 boosters**, all of type **F9 B5**, each carrying **15600 kg**.
This highlights the top-performing boosters in terms of payload capacity.

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
# Task 9
# List records with month names, failure landing outcomes in drone ship,
# booster versions, and launch site for the months in year 2015
query9 = """
SELECT substr(Date, 6, 2) AS Month,
       Landing_Outcome,
       Booster_Version,
       Launch_Site
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015'
  AND Landing_Outcome = 'Failure (drone ship)';
"""
df_task9 = pd.read_sql_query(query9, con)
df_task9
```

	Month	Landing_Outcome	Booster_Version	Launch_Site
0	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Explanation

substr(Date, 0, 5) = '2015' filters records from the year 2015.

Landing_Outcome = 'Failure (drone ship)' selects only failed drone ship landings.

substr(Date, 6, 2) extracts the month portion from the date string.

The result shows two such failures:

- **January**: F9 v1.1 B1012 at CCAFS LC-40
- **April**: F9 v1.1 B1015 at CCAFS LC-40

This query helps pinpoint when and where early drone ship landing failures occurred. I can also help you build a timeline of landing outcomes or a failure rate chart by year and landing type.

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

```
# Task 10
# Rank the count of landing outcomes between 2010-06-04 and 2017-03-20
query10 = """
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;
"""
df_task10 = pd.read_sql_query(query10, con)
df_task10
```

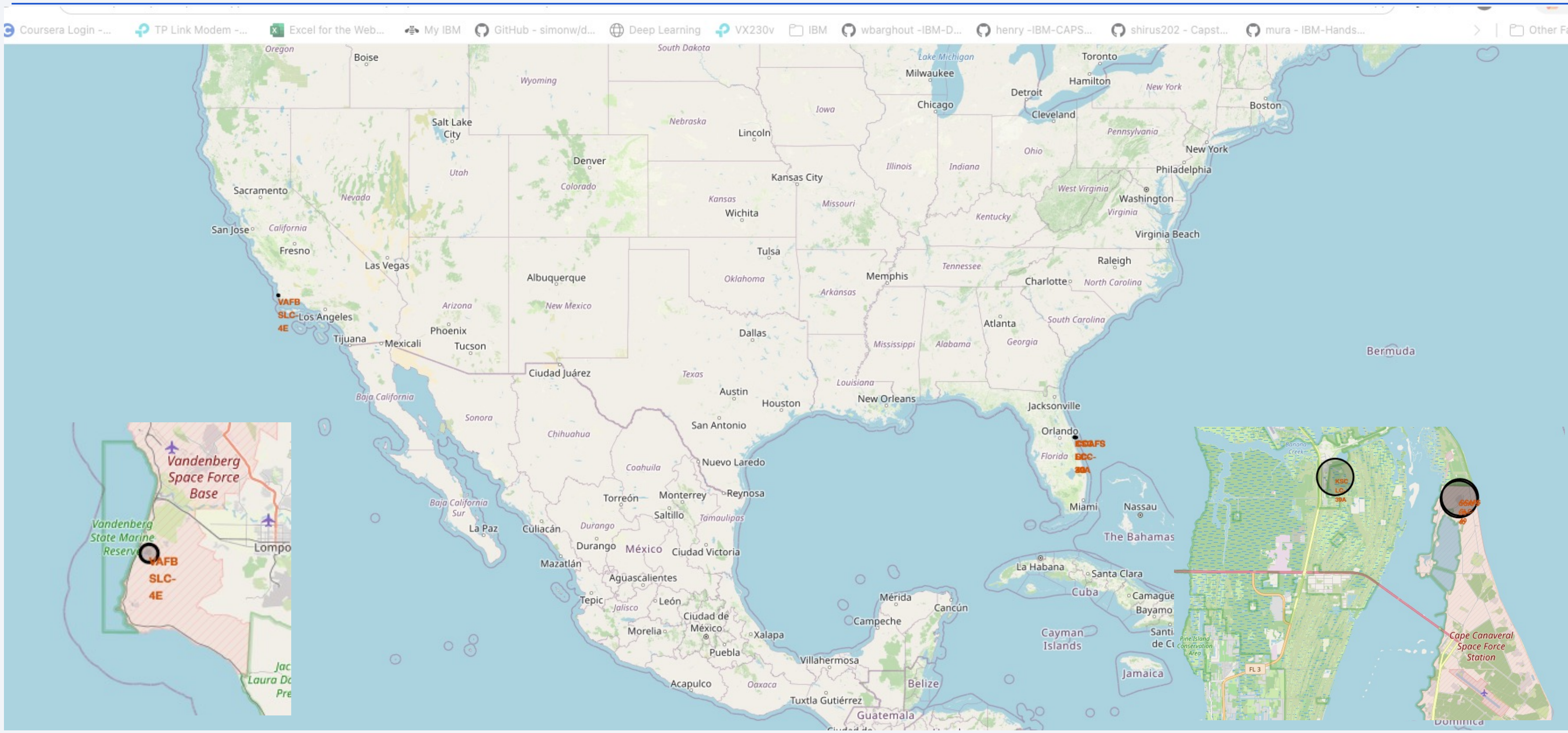
	Landing_Outcome	Outcome_Count
0	No attempt	10
1	Success (drone ship)	5
2	Failure (drone ship)	5
3	Success (ground pad)	3
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Failure (parachute)	2
7	Precluded (drone ship)	1

- **"No attempt"** dominates, suggesting many missions didn't aim for recovery.
- **Drone ship landings** had a 50% success rate (5 out of 10).
- **Ground pad landings** were less frequent but more reliable.
- **Ocean landings** show a mix of control and failure, hinting at fallback strategies.

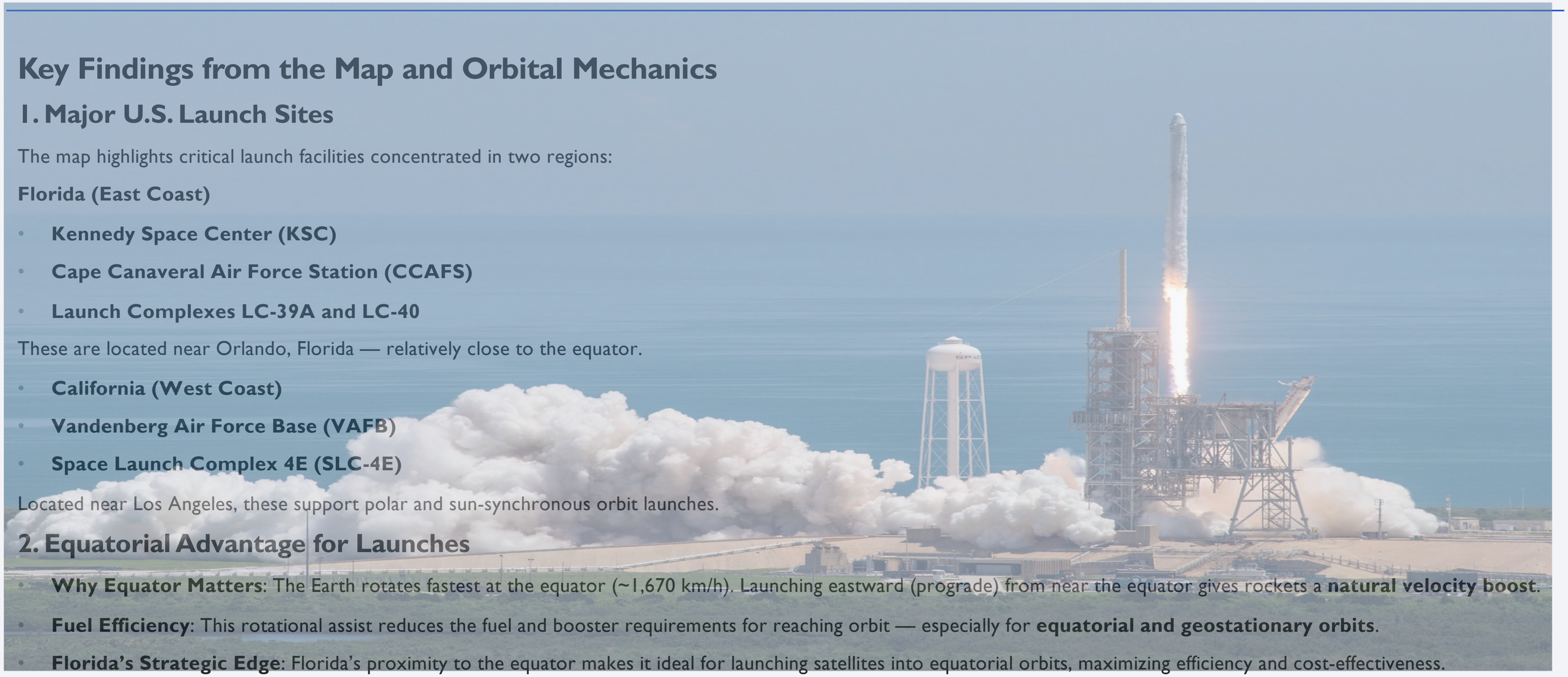
Section 3

Launch Sites Proximities Analysis

Folium Map - Launch Sites



Launch Sites



Key Findings from the Map and Orbital Mechanics

1. Major U.S. Launch Sites

The map highlights critical launch facilities concentrated in two regions:

Florida (East Coast)

- **Kennedy Space Center (KSC)**
- **Cape Canaveral Air Force Station (CCAFS)**
- **Launch Complexes LC-39A and LC-40**

These are located near Orlando, Florida — relatively close to the equator.

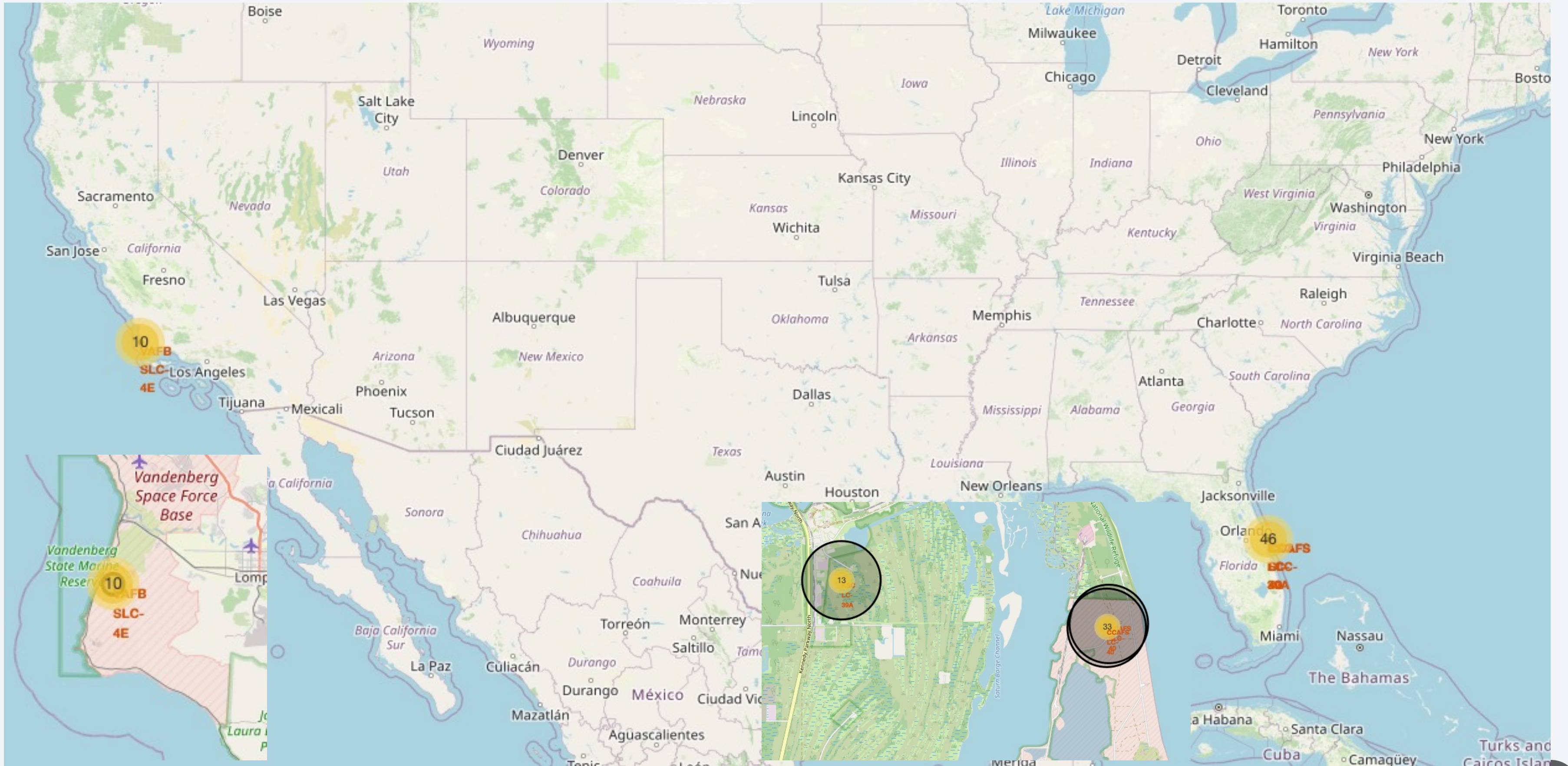
- **California (West Coast)**
- **Vandenberg Air Force Base (VAFB)**
- **Space Launch Complex 4E (SLC-4E)**

Located near Los Angeles, these support polar and sun-synchronous orbit launches.

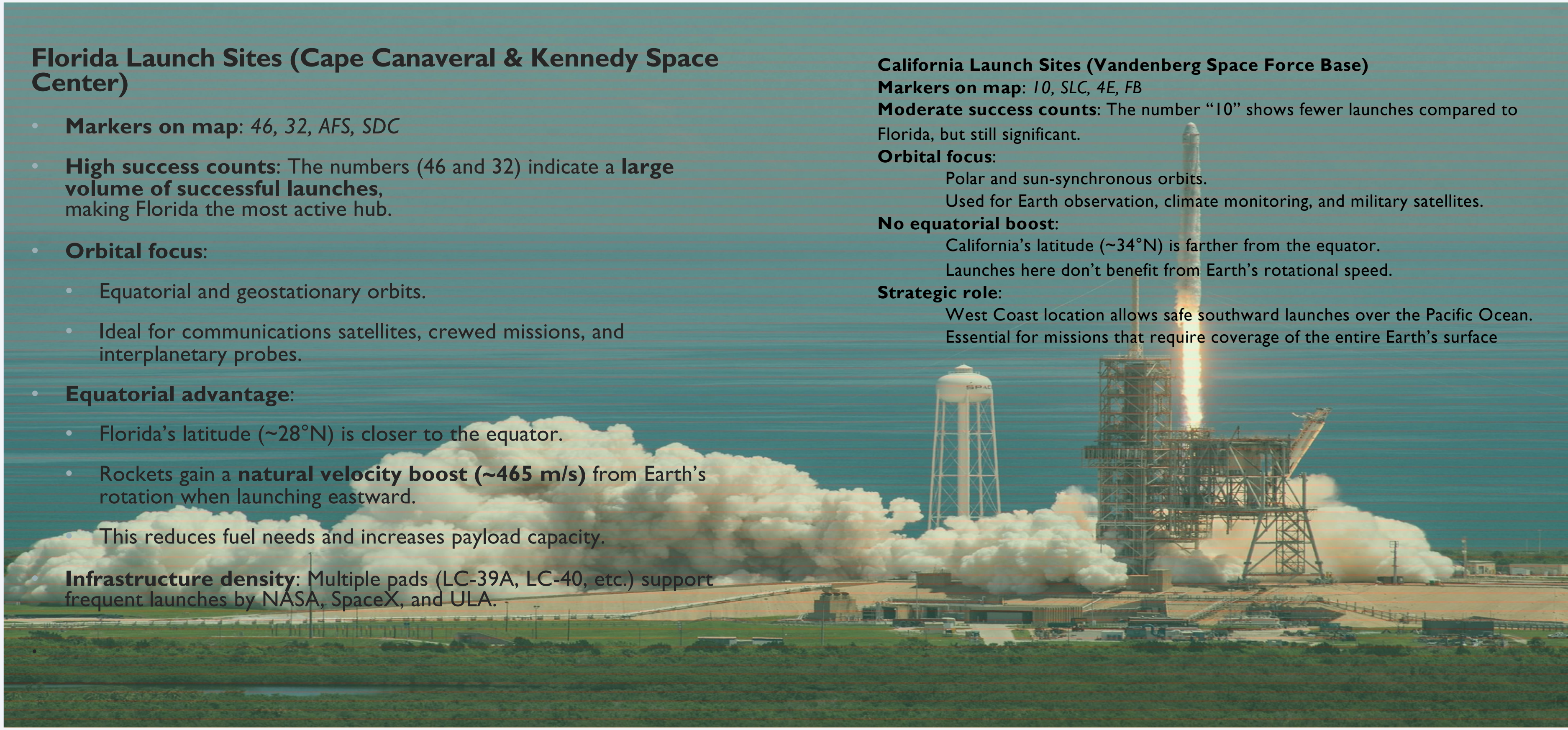
2. Equatorial Advantage for Launches

- **Why Equator Matters**: The Earth rotates fastest at the equator (~1,670 km/h). Launching eastward (prograde) from near the equator gives rockets a **natural velocity boost**.
- **Fuel Efficiency**: This rotational assist reduces the fuel and booster requirements for reaching orbit — especially for **equatorial and geostationary orbits**.
- **Florida's Strategic Edge**: Florida's proximity to the equator makes it ideal for launching satellites into equatorial orbits, maximizing efficiency and cost-effectiveness.

Folium Map - Launch Success



Launch Success

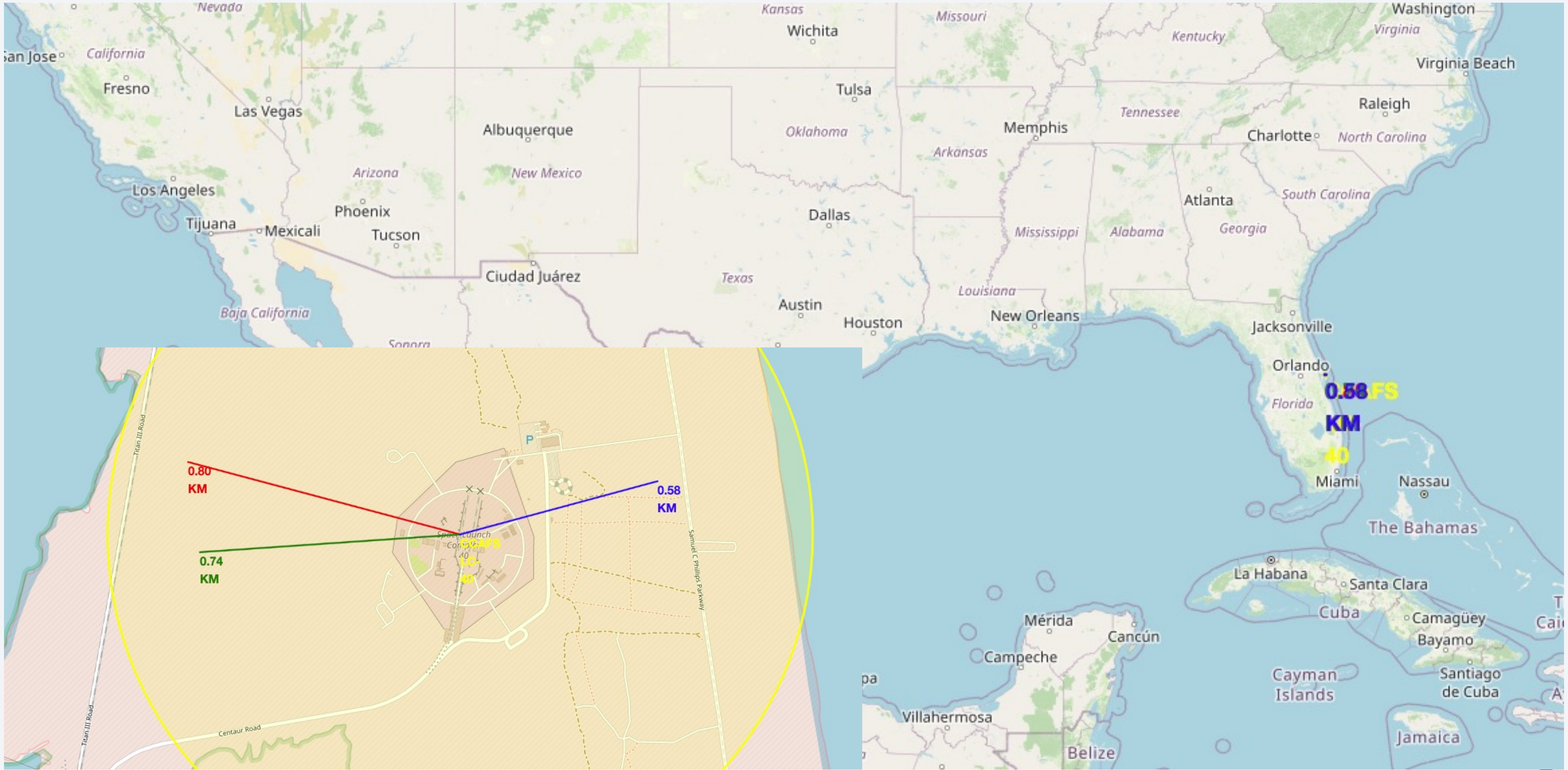


Florida Launch Sites (Cape Canaveral & Kennedy Space Center)

- **Markers on map**: *46, 32, AFS, SDC*
- **High success counts**: The numbers (46 and 32) indicate a **large volume of successful launches**,
  making Florida the most active hub.
- **Orbital focus**:
  - Equatorial and geostationary orbits.
  - Ideal for communications satellites, crewed missions, and interplanetary probes.
- **Equatorial advantage**:
  - Florida's latitude (~28°N) is closer to the equator.
  - Rockets gain a **natural velocity boost (~465 m/s)** from Earth's rotation when launching eastward.
  - This reduces fuel needs and increases payload capacity.
- **Infrastructure density**: Multiple pads (LC-39A, LC-40, etc.) support frequent launches by NASA, SpaceX, and ULA.

California Launch Sites (Vandenberg Space Force Base)
Markers on map: *10, SLC, 4E, FB*
Moderate success counts: The number "10" shows fewer launches compared to Florida, but still significant.
Orbital focus:
Polar and sun-synchronous orbits.
Used for Earth observation, climate monitoring, and military satellites.
No equatorial boost:
California's latitude (~34°N) is farther from the equator.
Launches here don't benefit from Earth's rotational speed.
Strategic role:
West Coast location allows safe southward launches over the Pacific Ocean.
Essential for missions that require coverage of the entire Earth's surface

Distance to Proximities



Folium - Distance to Proximities

Central Focus and Radial Distances

- **Yellow circular boundary**: Represents a defined radius or zone of interest around the central point — possibly a buffer zone or area under study.
- **Three colored lines** radiate outward from the center to nearby landmarks or measurement points:
  - 🔴 **Red line**: 0.87 km
  - 🔵 **Blue line**: 0.58 km
  - 🟢 **Green line**: 0.74 km
- These lines likely indicate **straight-line (Euclidean) distances** from the central location to key reference points such as buildings, infrastructure, or environmental features.

Geographic Context

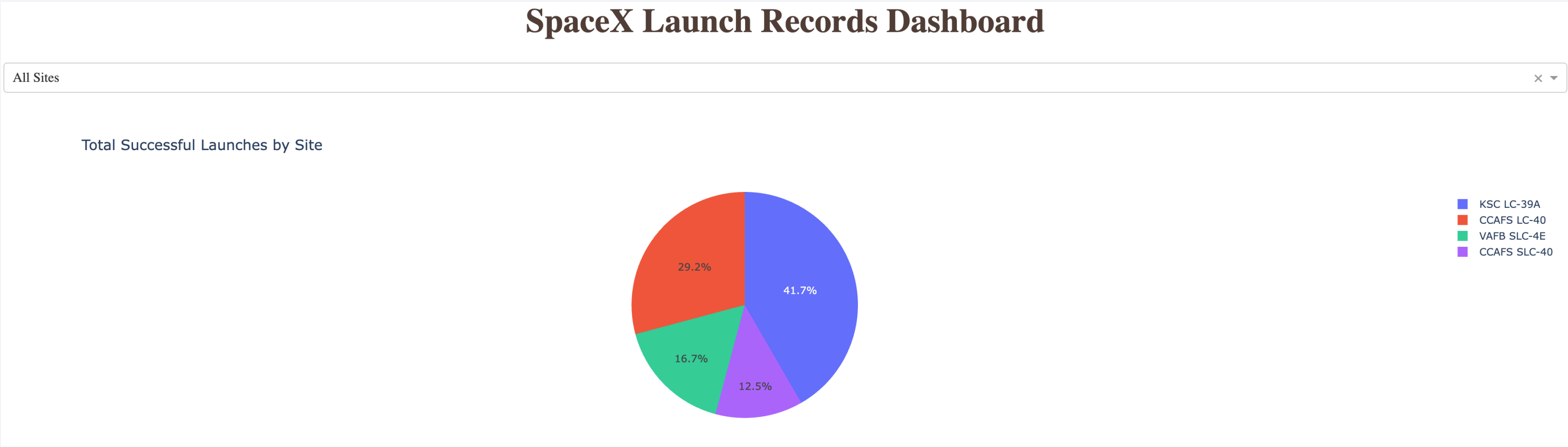
- **Coastline on the left**: Suggests the site is near a water body, which may influence planning decisions (e.g., flood risk, access to waterfront).
- **Roads and buildings**: Visible urban infrastructure implies this is a developed or semi-developed area.
- **Purpose**: This layout is often used for:
  - Site suitability analysis (e.g., how close is the site to schools, hospitals, or hazards?)
  - Emergency planning (e.g., evacuation zones, service coverage)
  - Environmental impact studies (e.g., proximity to sensitive areas)

Section 4

Build a Dashboard with Plotly Dash

Plotly - Launch Success Distribution by Site

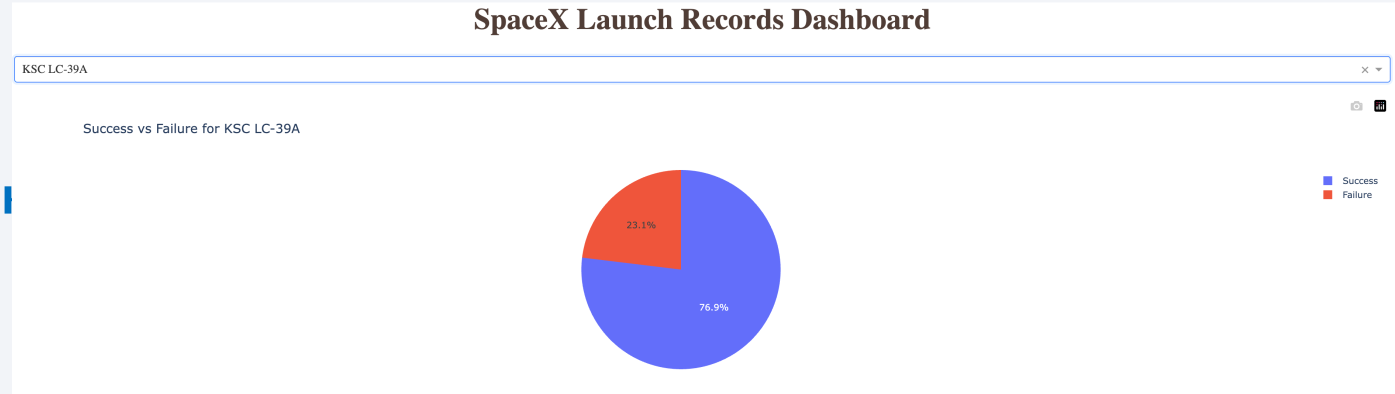
- Show the screenshot of launch success count for all sites, in a pie chart



Key Elements and Insights

Dominant Site: *KSC LC-39A* leads with 41.7% of total successful launches, indicating it's the most utilized or reliable site.
Secondary Contributor: *CCAFS LC-40* follows with 29.2%, showing strong operational activity.
Smaller Shares: *VAFB SLC-4E* and *CCAFS SLC-40* contribute less, possibly due to site-specific constraints or mission types.
Visual Design: The use of distinct colors and percentage labels enhances readability and quick comparison.
Dropdown Context: The "All Sites" dropdown confirms that the chart aggregates data across all launch locations.

Plotly - Success vs Failure for KSC LC-39A



- **High Success Rate:** With nearly 77% of launches succeeding, KSC LC-39A demonstrates strong operational reliability.
- **Visual Clarity:** The pie chart offers an immediate visual cue—dominant blue indicates a favorable launch history.
- **Site Performance Focus:** This chart helps compare launch site effectiveness, especially useful for identifying the most reliable site.

Plotly - Payload vs. Launch Outcome scatter plot for all sites

🧠 **Key Findings**

Success Dominance: Most launches are clustered around the "Success" category, indicating high reliability across payloads.

Booster Performance:

FT (blue) appears most frequently, suggesting it's widely used and generally successful.

B4 and B5 have fewer data points, but still show strong success rates.

Payload Impact: No clear failure trend tied to heavier payloads—success spans across the payload range.

Plotly - Payload vs. Launch Outcome scatter plot for all sites

Key Findings

Success Dominates: Most data points are aligned with the "Success" category, indicating high reliability across booster versions and payloads.

FT Booster (Blue): Appears most frequently and shows consistent success across a wide payload range — suggesting it's the most robust and widely used.

B4 and B5 Boosters (Red and Green): Less frequent but still show strong success rates, with minimal failures.

Payload Impact: No clear correlation between heavier payloads and failure — successful launches span from ~2000 kg to ~7000 kg.

This graph helps assess how different booster versions perform across varying payload masses. The dominance of successful launches, especially with FT boosters, suggests strong engineering reliability. It also shows that payload mass alone doesn't predict failure, which is valuable for planning future missions.

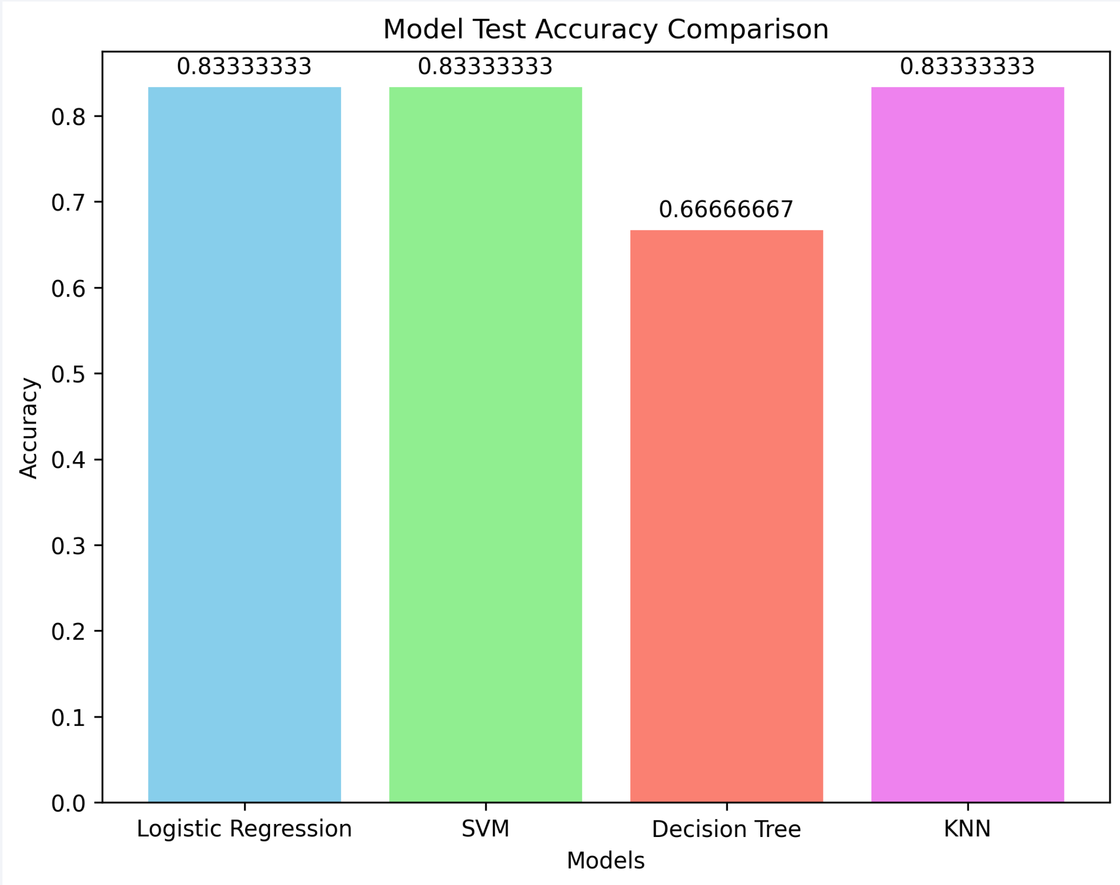
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Visualize the built model accuracy for all built classification models, in a bar chart

- 📊 **Model Accuracies**
- Logistic Regression: 0.83333333
- SVM: 0.83333333
- Decision Tree: 0.66666667
- KNN: 0.83333333
- 🔎 **Interpretation**
- **Logistic Regression, SVM, and KNN** all achieved the same accuracy (~83.3%). This suggests that these models are equally effective at capturing the patterns in your dataset.
- **Decision Tree** performed noticeably worse (~66.7%), which indicates it may be overfitting or failing to generalize well on the test data. Decision Trees can be sensitive to noise and small variations in the dataset.
- **Logistic Regression** as the "best performing model" because it was the first maximum encountered in the dictionary. Technically, Logistic Regression, SVM, and KNN are tied.



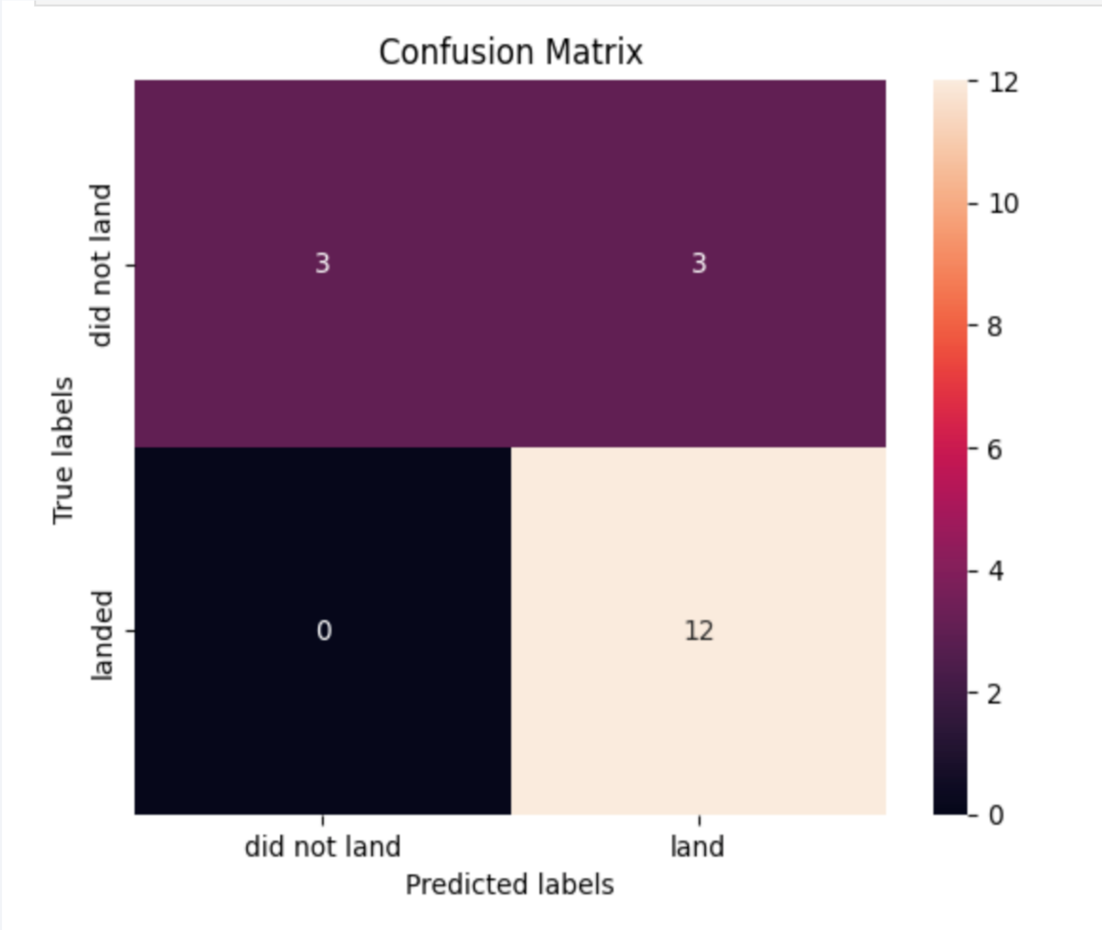
Confusion Matrix

The confusion matrix is identical across **Logistic Regression**, **SVM**, and **Decision Tree**, we can interpret it once and apply that understanding to all three models.

- **True Positives (TP = 12)**: The model correctly predicted 12 landings.
- **True Negatives (TN = 3)**: The model correctly predicted 3 non-landings.
- **False Positives (FP = 3)**: The model incorrectly predicted 3 landings that didn't happen.
- **False Negatives (FN = 0)**: The model never missed a landing — excellent!

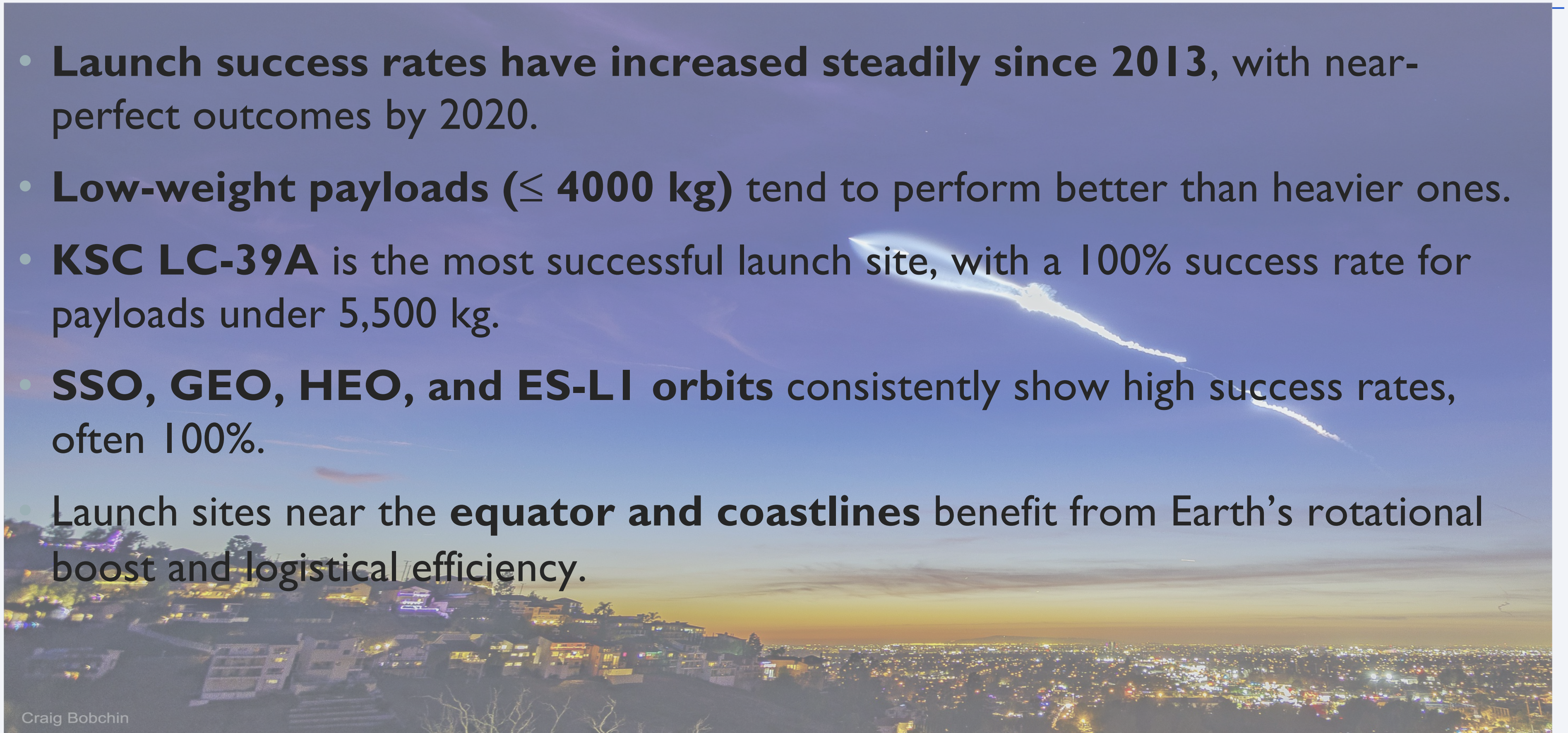
Interpretation

- The model is **very strong at detecting actual landings** (no false negatives).
- It makes **some mistakes by predicting land when it didn't happen** (false positives).
- This trade-off might be acceptable if missing a landing is worse than falsely predicting one.



Conclusions

- **Launch success rates have increased steadily since 2013**, with near-perfect outcomes by 2020.
- **Low-weight payloads (≤ 4000 kg)** tend to perform better than heavier ones.
- **KSC LC-39A** is the most successful launch site, with a 100% success rate for payloads under 5,500 kg.
- **SSO, GEO, HEO, and ES-L1 orbits** consistently show high success rates, often 100%.
- Launch sites near the **equator and coastlines** benefit from Earth's rotational boost and logistical efficiency.



APPENDIX

GitHub Link – link to notebooks and python files

- 1-jupyter-labs-spacex-data-collection-api.ipynb
- 2-jupyter-labs-webscraping.ipynb
- 3-jupyter-spacex-Data wrangling.ipynb
- 4-jupyter-labs-eda-sql-coursera_sqllite.ipynb
- 5-edadataviz.ipynb
- 6-lab_jupyter_launch_site_location.ipynb
- 7-spacex-dash-app.py
- 8-SpaceX_Machine Learning Prediction_Part_5.ipynb

- task1_launch_sites_map.html
- task2_launch_success_map.html
- task3_distance_map.html

- Plotly Dashboard - All Payloads and Boosters.png
- Plotly Dashboard - All Site Success - Pie Chart.png
- Plotly Dashboard - SpaceX Launch Result Pie Chart.png
- flight_payload_plot.png
- flightnumber_vs_launchsite.png

APPENDIX

- Plotly Dashboard - All Payloads and Boosters.png
- Plotly Dashboard - All Site Success - Pie Chart.png
- Plotly Dashboard - SpaceX Launch Result Pie Chart.png

- flight_payload_plot.png
- flightnumber_vs_launchsite.png
- flightnumber_vs_orbit.png
- payloadmass_vs_launchsite.png
- payloadmass_vs_orbit.png
- success_rate_by_orbit.png
- yearly_success_trends.png


Thank you!