IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mahendra M Joshi
31-Dec-2025

Table of Contents

Executive Summary

- **Objective & Scope**: Developed a predictive framework to determine the commercial viability of Falcon 9 booster recovery.

  The goal is to provide data-driven insights for competitive bidding in the commercial space launch market.

- **Data Strategy (Multi-Source Integration)**: Synthesized data from the SpaceX REST API and Web Scraping (Wikipedia) to build a comprehensive dataset of 90+ launches.

  Performed rigorous data cleaning and One-Hot Encoding to handle 80+ categorical features.

- **Exploratory Insights (EDA)**:

  SQL Analysis: Identified that success rates vary significantly by Launch Site (KSC LC-39A being the most reliable) and Payload Mass.

  Geospatial Analysis: Leveraged Folium to visualize launch site proximity to coastlines and railways, highlighting logistical dependencies for booster recovery.

- **Predictive Performance:**

  Evaluated 4 Machine Learning models (LogReg, SVM, Tree, KNN) using GridSearchCV.

  All models achieved a Test Accuracy of 83.33%, successfully identifying a consistent "Success Profile" for the Falcon 9 booster.

Introduction

The Problem:

Traditional space launches are prohibitively expensive because rockets are discarded after one use.

SpaceX has disrupted the industry by successfully landing and reusing the Falcon 9 first-stage booster.

The Context:

SpaceX advertises Falcon 9 launches for $62 million, while competitors charge upwards of $165 million.

The primary source of this cost savings is the ability to recover and reuse the first stage, which accounts for a significant portion of the total vehicle cost.

The Goal:

To compete with SpaceX, other companies need to determine the price of a launch.

If one can predict whether the first stage will land successfully, then we can calculate the actual cost of a launch.

The Technical Challenge:

Analyze historical launch data (orbits, payload mass, sites) to identify the specific conditions that lead to a successful landing.

Develop a robust predictive model that estimates landing probability for future mission profiles.

Section 1

Methodology

Methodology – 1 of 4

Sequence of Major Tasks

- Perform Data collection
- Perform Data wrangling
- Perform Exploratory data analysis (EDA) using visualization(Matplotlib, Seaborn) and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using four different classification models
- Draw conclusions based on the computed output.

Methodology - 2 of 4

1. Data Collection

Integrated multi-source data by extracting launch records from the SpaceX REST API and scraping historical mission details from Wikipedia using BeautifulSoup. Details of data collection steps are given in separate slides titled “Data Collection”

2. Data Wrangling

Cleaned and structured the dataset by handling missing values, creating binary success labels, and performing one-hot encoding on categorical features to prepare for machine learning. Details of data wrangling process are given in separate slides titled “Data Wrangling”.

3. EDA with SQL

Executed complex SQL queries to identify launch site trends, payload distributions, and mission success frequencies within specific historical timeframes.

4. EDA with Visualization

Utilized Matplotlib and Seaborn to visualize the relationship between payload mass, orbit types, and flight numbers to uncover patterns in booster recovery success. More details of EDA activities are provided in Section 2 slides.

Methodology – 3 of 4

5. Interactive Visual Analytics

Developed interactive maps in Folium to analyze launch site proximity to logistics and coastlines, and built a Plotly Dash dashboard for real-time success rate filtering

6. Predictive Analysis (ML)

Data Preparation

- Created a Feature Matrix (X) and a Target Vector (Y).
- Applied StandardScaler to ensure all numerical features have a mean of 0 and a variance of 1, preventing high-magnitude features (like Payload Mass) from biasing the models.

Model Training Strategy

- Performed an 80/20 Train/Test Split to evaluate model performance on unseen data.
- Employed GridSearchCV for hyperparameter tuning, ensuring the "best" version of each algorithm was used.

Validation Technique

- Used 10-Fold Cross-Validation during the training phase. This minimizes the risk of "overfitting" by testing the model on different subsets of the data

Methodology – 4 of 4

Algorithms Evaluated

- Logistic Regression: For baseline linear classification.
- Support Vector Machine (SVM): To find the optimal hyperplane for separation.
- Decision Tree: To capture non-linear relationships.
- K-Nearest Neighbors (KNN): To classify based on data proximity.

7. Interpretation of the Computed Results - Conclusions.

Data Collection

Process: Utilized the requests library to fetch launch data from the SpaceX REST API.

Data Extraction:

Custom functions used to pull nested data as listed in the table .

Technical Detail: Filtered out Falcon 1 launches to maintain focus on the reusable Falcon 9 fleet.

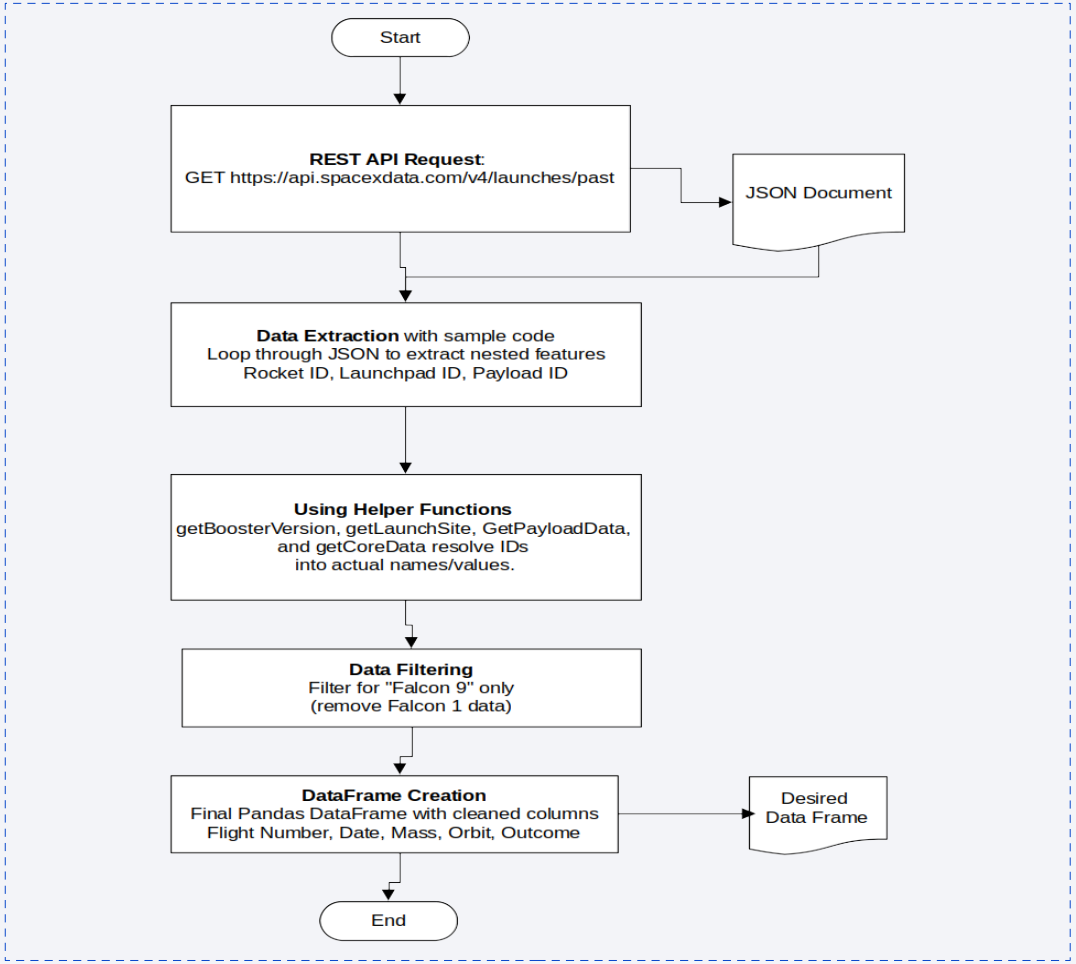
Result: Converted the raw JSON response into a Pandas DataFrame for initial analysis.

Specific Data	Custom Function
Booster Version	getBoosterVersion(data)
Payload (Mass and Orbit)	getPayloadData(data)
Launch Site names	getLaunchSite(data)
Landing Outcomes (Success/Failure and Landing Pad details)	getCoreData(data)

Data Collection – SpaceX API

- The flowchart shows data collection steps undertaken using the SpaceX REST calls.

- Access Full Notebook on GitHub: https://github.com/mmj009/Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- **Tooling**

  Leveraged BeautifulSoup4 for DOM navigation and requests for page fetching.

- **Precision**

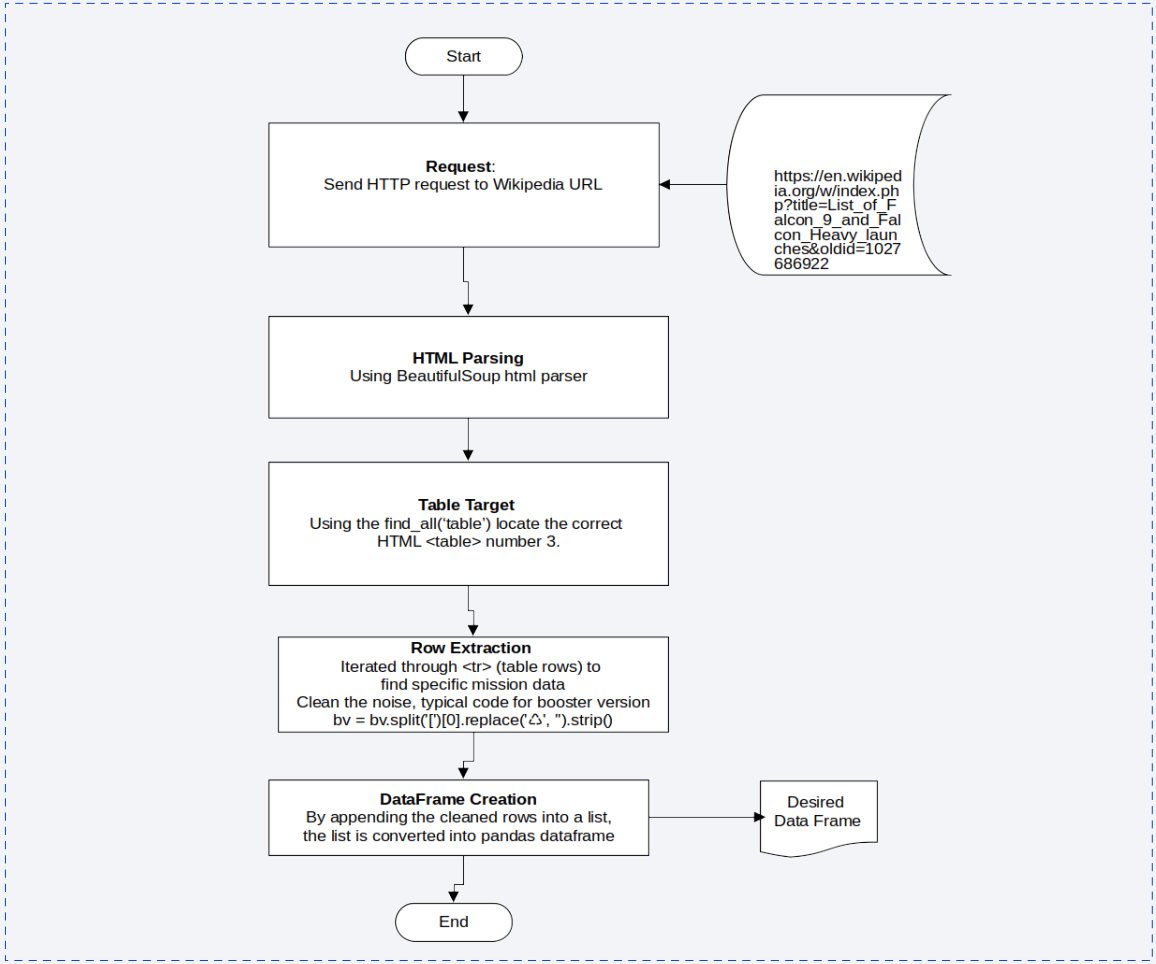
  Specific attributes targeted

- **Data Validation**

  Cross-referenced scraping results with the SpaceX API to ensure date and flight number consistency

- **Access Full Notebook on GitHub:**

  https://github.com/mmj009/Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- **EDA**

  Initial Exploratory Data Analysis to figure out the data types of each attribute and to find out the missing values in these attributes.

- **Number of launches for each site**

- **Identify number and occurrence of each orbit**

  Except GTO as it is a transfer orbit

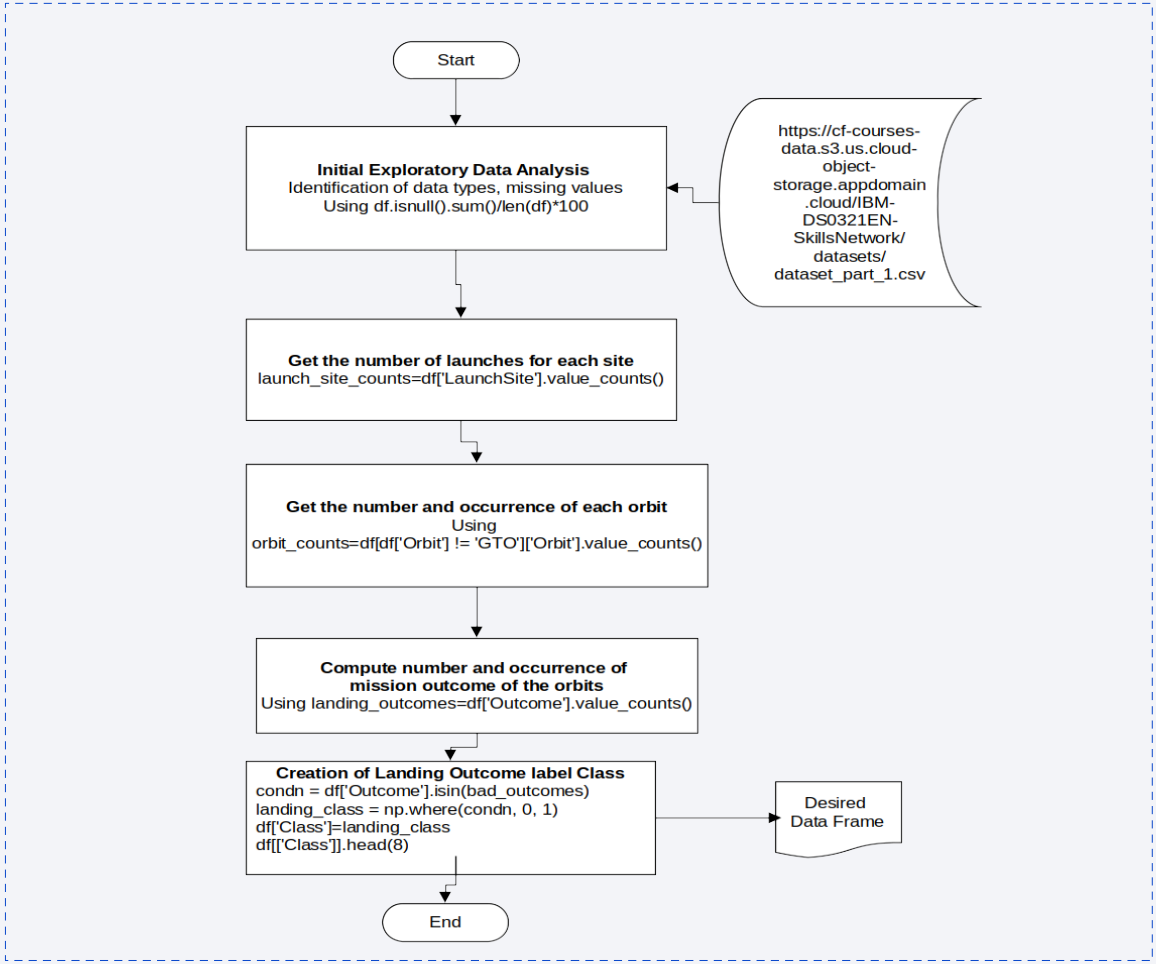
- **Computing the number and occurrence of mission outcomes of the orbits**

- **Creating Landing Outcome label**

  Binary classification (Class 1/0) to simplify the objective for the predictive models

- **Access Full Notebook on GitHub:**

  https://github.com/mmj009/Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling(1).ipynb



Section 2

Insights drawn from EDA

EDA with Data Visualization – 1 of 6

Following table provides summary of plots created using Matplotlib and Seaborn libraries and the associated reasoning behind these plots. It also provides URL for the associated Jupyter notebook hosted on github.

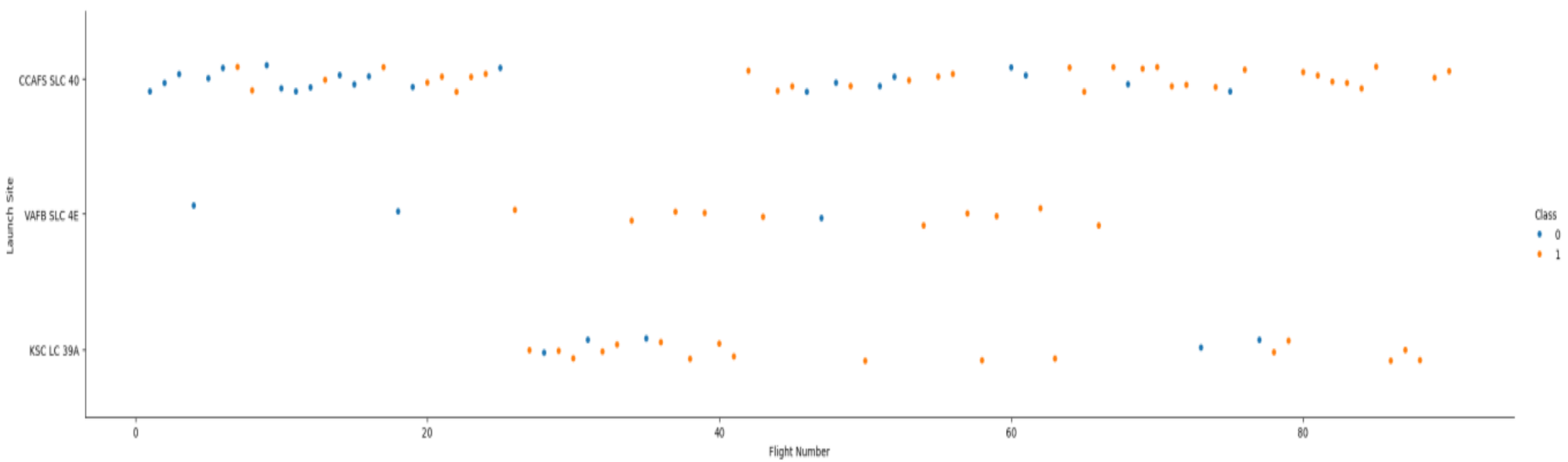
Plots and Graphs Utilized	Reasons for using these Plots and Graphs
Scatter Plots Employed to visualize the relationship between Flight Number vs. Launch Site and Payload Mass vs. Launch Site, with points color-coded by success/failure class.	**Pattern Recognition**: Scatter plots were chosen to identify "clusters" of failure or success. They helped determine if higher payload masses or specific launch sites were statistically more riskier or not. **Performance Benchmarking**: Bar charts provided a clear, high-level comparison of which orbits are most "booster-friendly," allowing us to rank mission types by their recovery probability. **Progress Tracking**: The line chart was essential to demonstrate the SpaceX Learning Curve, visually proving how landing technology matured from experimental stages to operational reliability over a decade. **Decision Support**: These visualizations transformed raw data into a helpful narrative for stakeholders to understand the variables that influence a $62M launch decision.
Bar Charts Created to compare the Success Rates across different Orbit types (e.g., LEO, GTO, SSO).	
Line Charts Developed to track the Launch Success Yearly Trend from 2010 to 2020.	
Categorical Plots Used to analyze the distribution of mission outcomes across various landing pads.	
Access Full Notebook on GitHub: https://github.com/mmj009/Data-Science-Capstone/blob/main/edadataviz.ipynb	

EDA with Data Visualization – 2 of 6

EDA - Launch Site & Payload Trends – 1 of 2

Flight Number vs. Launch Site

Observation: As the flight number increases, the success rate improves across all sites.
Insight: This visually represents the launch technology matured significantly after the first 20 odd launches.



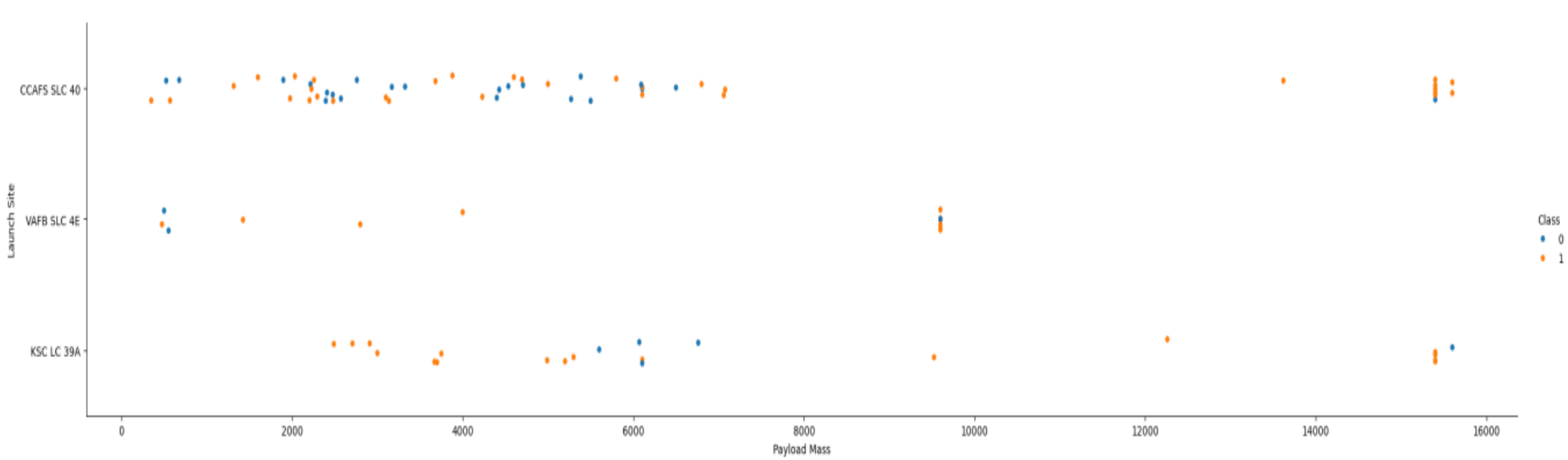
EDA with Data Visualization – 3 of 6

EDA - Launch Site & Payload Trends 2 of 2

Payload Mass vs. Launch Site

Observation: Most heavy payloads (>10,000Kg) are launched from KSC LC-39A and CCAFS SLC-40.

Insight: There is no clearly visible failure cluster for heavy payloads. Indicating the Falcon 9 is reliable for heavy payloads.



EDA with Data Visualization – 4 of 6

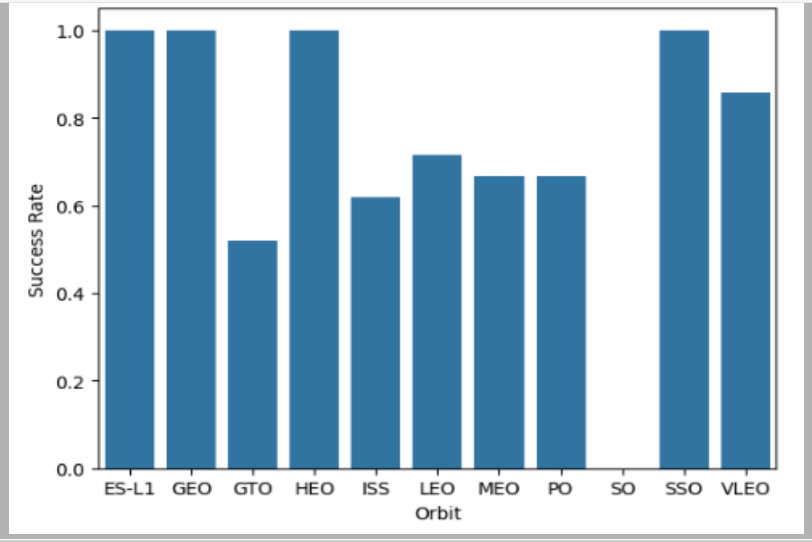
EDA - Success Rates by Orbit & Mission Profile

Success Rate by Orbit Type

Highest Success: Orbits like ES-L1, GEO, HEO, and SSO show a 100% success rate.

Lowest Success: SO (Sun-synchronous Orbit) shows 0% success (based on limited data points).

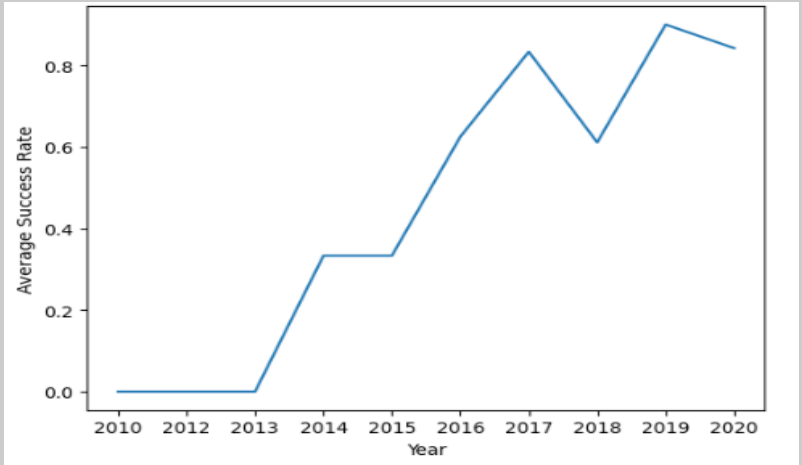
Insight: Mission profile is a strong predictor of landing success; certain orbits require more fuel for the primary mission, leaving less for the landing burn.



Success Rate Trend (Line Chart)

Trend: A clear upward trajectory in success rate from 2013 to 2020.

Insight: By 2020, successful landings became the norm rather than an exception.

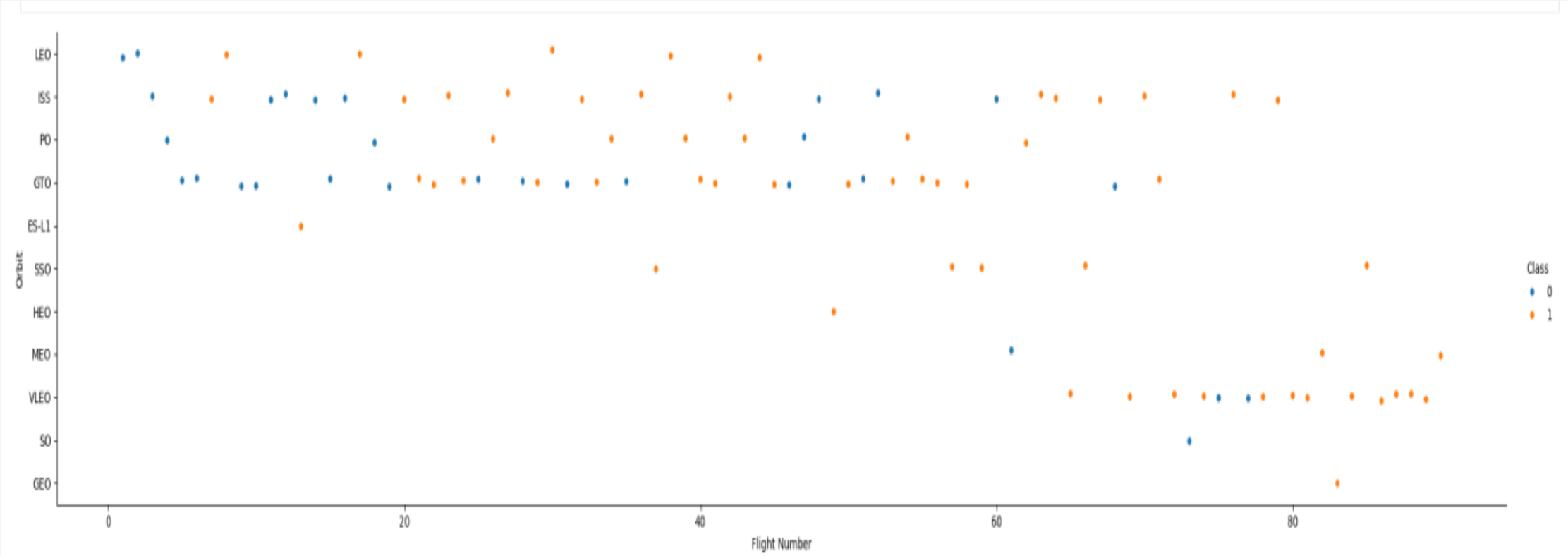


EDA with Data Visualization – 5 of 6

EDA - Landing Outcomes & Flight Consistency – 1 of 2

Flight Number vs. Orbit Type

Observation: For LEO Orbit the success has relationship with the flight number. For GTO orbit there is no such relationship visible.
Insight: Increased missions over time did not negatively impact landing success.

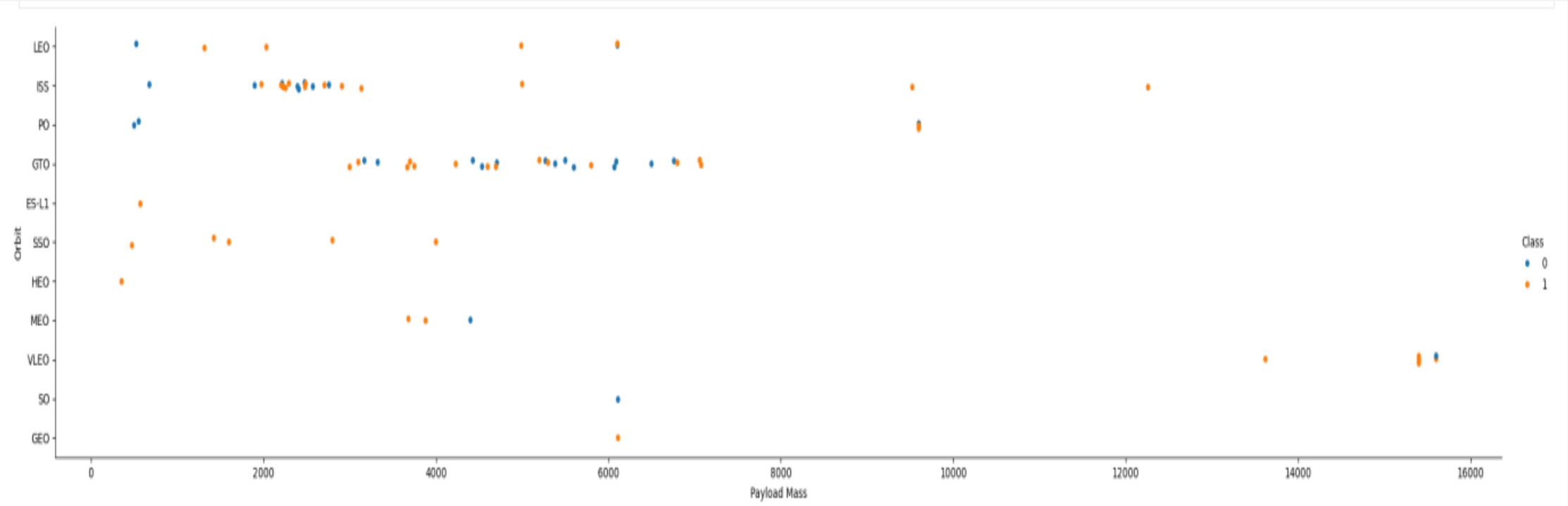


EDA with Data Visualization – 6 of 6

EDA - Landing Outcomes & Flight Consistency 2 of 2

Payload Mass and Orbit Type

Trend: Successful landing rate for Polar, LEO and ISS orbits is higher for heavy payloads but the same is not applicable for GTO orbit.
Insight:Orbit type is also one of the risk factor as seen from the GTO orbit's successful/unsuccessful class distribution.



EDA with SQL – 1 of 5

Following table provides summary of SQL queries used and the reasoning behind these queries. It also provides URL for the associated Jupyter notebook hosted on github.

SQL Queries Used	Reasons for using these SQL Queries
Filtering & Selection Retreived unique launch sites and filtered records for specific missions (e.g., the first successful landing at CCAFS).	**Data Integrity**: SQL allowed for precise extraction of data subsets directly from the database, ensuring that the foundational numbers used for later visualizations were 100% accurate. **Complex Filtering:** SQL was used to answer specific business questions like "Which booster version carried the maximum payload?". **Efficiency:** Using SQL we can handle large datasets more efficiently than manual inspection allowing us to quickly summarize many data rows into actionable insights like "Average Payload per Orbit." **Foundation for Analysis:** The results of these queries served as the "sanity check" for the entire project, ensuring the Machine Learning models were built on a verified historical record.
Aggregate Functions Used COUNT, SUM, and AVG to calculate total payload masses and determine the frequency of landing outcomes.	
Conditional Logic Applied CASE statements and WHERE clauses to isolate specific timeframes (e.g., success rates between 2010 and 2017).	
Subqueries & Joins Nested queries were used to identify the "top" payloads and correlate boosters with their specific mission success records.	
Access Full Notebook on GitHub: https://github.com/mmj009/Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite(1).ipynb	

EDA with SQL – 2 of 5

Launch Site and Payload Insights – 1 of 2

- **List of Unique Launch Sites**
  %sql SELECT DISTINCT Launch_Site from SPACEXTABLE
  This provides a clear insight on the geographical scope of analysis.

Out[54]:

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

- **List of few records where the site name begins with 'CCA'**
  %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5

Out[55]:

Date	Time (UTC)	Booster_Version	Launch_Site
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40

- **Total payload mass carried by boosters launched by NASA (CRS)**
  %sql SELECT SUM(PAYLOAD_MASS__KG_) AS 'Total Payload' FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'

  NASA (CRS) sent 45596Kg of payload!

Out[56]:

Total Payload
45596

- **Average payload mass carried by booster version F9 v1.1**
  %sql SELECT AVG(PAYLOAD_MASS__KG_) AS 'Average Payload Mass' FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'

  Average payload is about 2535Kg

Out[57]:

Average Payload Mass
2534.6666666666665

EDA with SQL – 3 of 5

Launch Site and Payload Insights – 2 of 2

- **Boosters which have success in drone ship and have payload mass between 4000Kg and 6000Kg**
  %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)

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

- **Booster versions that have carried the maximum payload mass**
  %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)

  Part of the output is shown.

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

EDA with SQL – 4 of 5

Mission Outcomes and History – 1 of 2

- **First successful landing outcome in ground pad**
  %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)

MIN(Date)
2015-12-22

- **Month-wise landing failure in the year 2015 with booster version and launch site name**
  %sql SELECT CASE substr(Date, 6, 2) \
  WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'Marc'\
  WHEN '04' THEN 'April' WHEN '05' THEN 'May' WHEN '06' THEN 'June'\
  WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September'\
  WHEN '10' THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December'\
  END AS "Month Name",\
  Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015'

Out[63]:

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

- **Total number of successful and failure mission outcomes**
  %sql SELECT SUM(CASE WHEN Landing_Outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS 'Total Number of Successful Missions', \
  SUM(CASE WHEN Landing_Outcome LIKE 'Failure%' THEN 1 ELSE 0 END) AS 'Total Number of Failure Missions'\
  FROM SPACEXTABLE\
  WHERE Landing_Outcome NOT LIKE '%No attempt%'

Total Number of Successful Missions	Total Number of Failure Missions
61	10

EDA with SQL – 5 of 5

Mission Outcomes and History – 2 of 2

- **Count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.**
  %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

- **Insights** :: This is the era [2010 – 2017] during which "**No attempt**" was having maximum value, which shows the transition from early "Failures" and "Uncontrolled" ocean landings to the emergence of "Success (ground pad)" and "Success (drone ship).
  So, re-usability wasn't an overnight success.

- **Note**: SQL sorting in the Lab notebook was limited by string-literal interpretation; data manually verified and sorted for this visualization in the adjoining table. The corrected SQL code shown above wherein Outcome_Count is the column title instead of 'Outcome Count' as seen in the lab code marked with a smiley.

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

```
GROUP BY Landing_Outcome \
ORDER BY 'Outcome Count' DESC
```

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

Out[64]:

Landing_Outcome	Outcome Count
Uncontrolled (ocean)	2
Success (ground pad)	3
Success (drone ship)	5
Precluded (drone ship)	1
No attempt	10
Failure (parachute)	2
Failure (drone ship)	5
Controlled (ocean)	3

Section 3

Launch Sites Proximities Analysis

Build an Interactive Map with Folium

The table summarizes which map objects were created and added to a Folium map and rational behind these actions.

Map objects created and added	Reasons for addition of these Map objects
Coordinate Markers (Marker): Placed at the exact Latitude and Longitude of each launch site (CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E).	**Safety & Risk Mitigation**: The circles and coastline lines prove that launch pads are strategically placed near water to ensure that any "off-nominal" flight path results in an ocean splashdown rather than an inland catastrophe. **Logistics Efficiency**: Measuring the distance to railways and highways explains how SpaceX manages the massive logistical challenge of transporting 70-meter rocket stages. **Trend Visualization**: MarkerClusters help identify which sites matured the fastest; for example, seeing a cluster turn from mostly red to mostly green at KSC LC-39A over time.
Safety Buffer Zones (Circle): Added 1km circles around launch pads to visualize the immediate operational safety perimeter.	
Success/Failure Clusters (MarkerCluster): Grouped launch outcomes into clusters that expand upon clicking, using Green for success and Red for failure.	
Proximity Measurement Lines (PolyLine): Drew lines between launch pads and the nearest Coastline, Highway, and Railway.	
Distance Markers (DivIcon): Added text labels on the map to display the calculated Haversine Distance (in kilometers) for each PolyLine.	
Access Full Notebook on GitHub: https://github.com/mmj009/Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb	

All the Launch Sites – 1 of 2

Overview of the Launch Sites on the global map where we can see the site on LHS of the map titled as VAFB-SLC-4E and a group of 3 sites on the RHS marked with red dot. The RHS is zoomed and shown further in the next slide.

Insights

Coastal Proximity: All the four launch sites are located within 1 km of the coastline. This is a strategic safety requirement to ensure that in the event of a launch failure, debris falls into the ocean rather than on the populated cities.

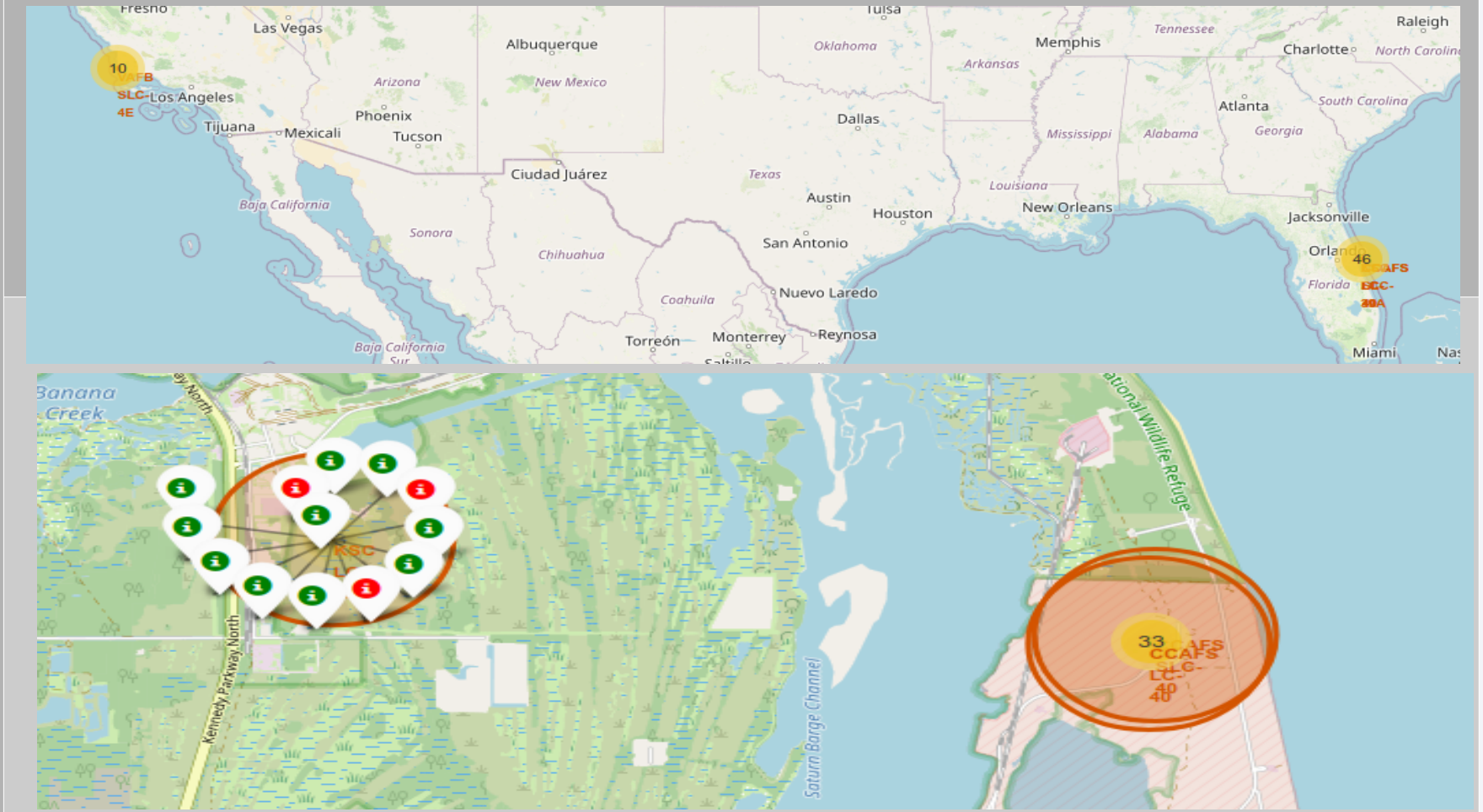
All the Launch Sites – 2 of 2

RHS of the map of the previous slide is zoomed in to show the other 3 sites therein marked with red circles.
Insights:
Transport Logistics: Launch sites are situated in close proximity to Railways and Highways. This infrastructure is vital for transporting large booster stages and fuel components from manufacturing facilities to the pad.
Urban Safety: Sites maintain a significant buffer distance from Cities to minimize acoustic impact and risks associated with the launch operations.

Launch Sites with Success & Failures

Insights:

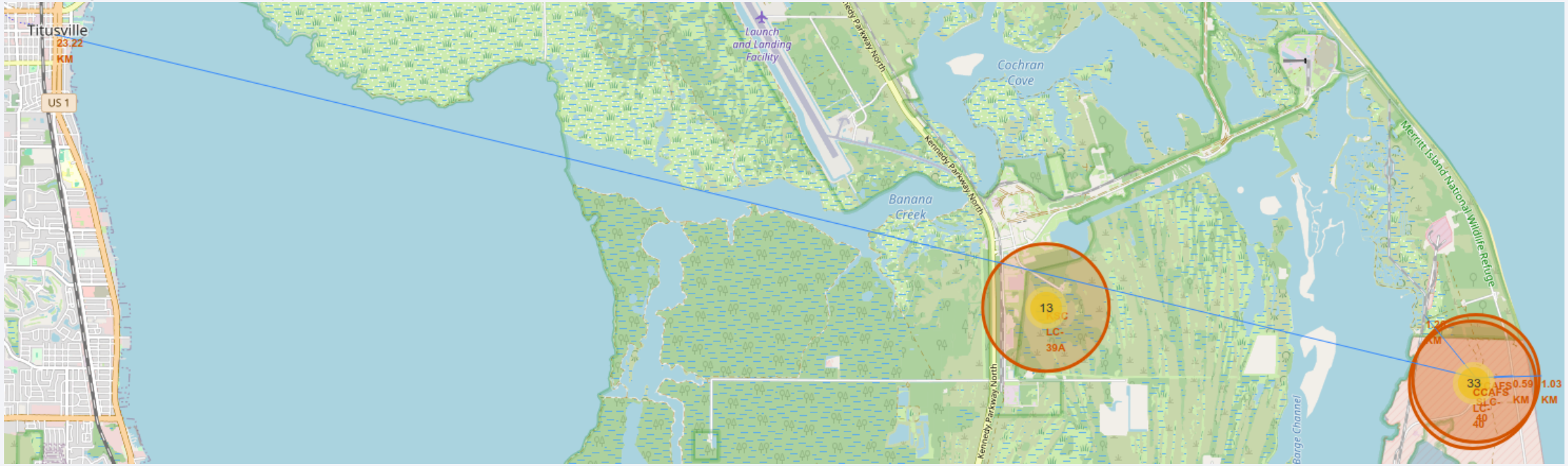
On the RHS of the plot, we have the number of launches (33) from the specific geographic area, drilling it down leads to the site specific successful launches using a green marker and failures with red marker as shown on the LHS of the plot for the site KSC-LC-39A which indicates 10 successful and 3 failed launches.

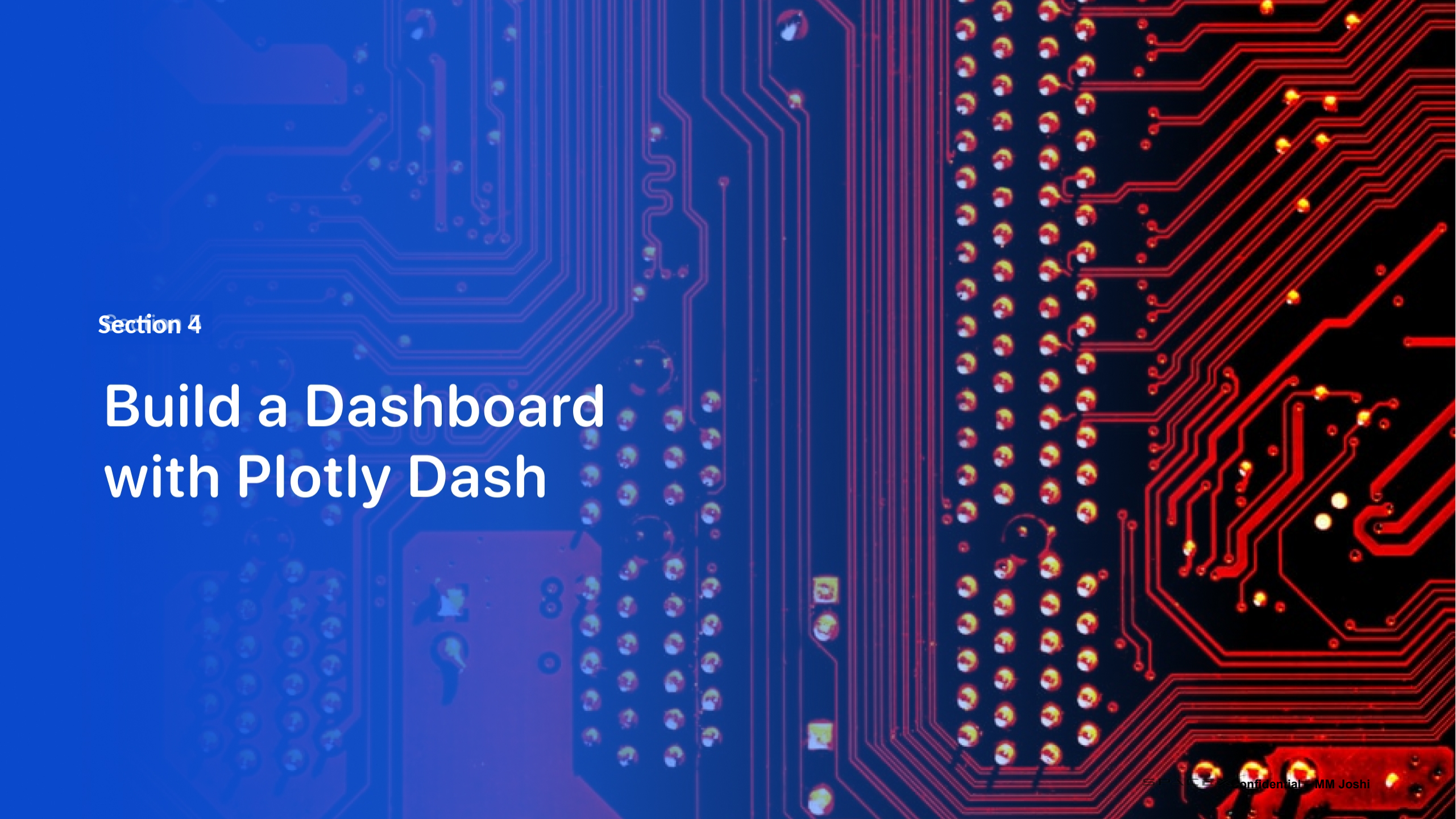


Launch Site - Proximity Analysis

Below plot shows proximity analysis of one of the launch site CCAFS-SLC-40.

Insights:
It can be seen that the Seashore, Highway and RailLine in just about 1Km away, Titusville city is 23.22Km away.
These geospatial factors confirm that the launch site is not only the launch pad but also requires logistical ecosystem like Transportation, Safety, Reusability.





Section 4

Build a Dashboard with Plotly Dash

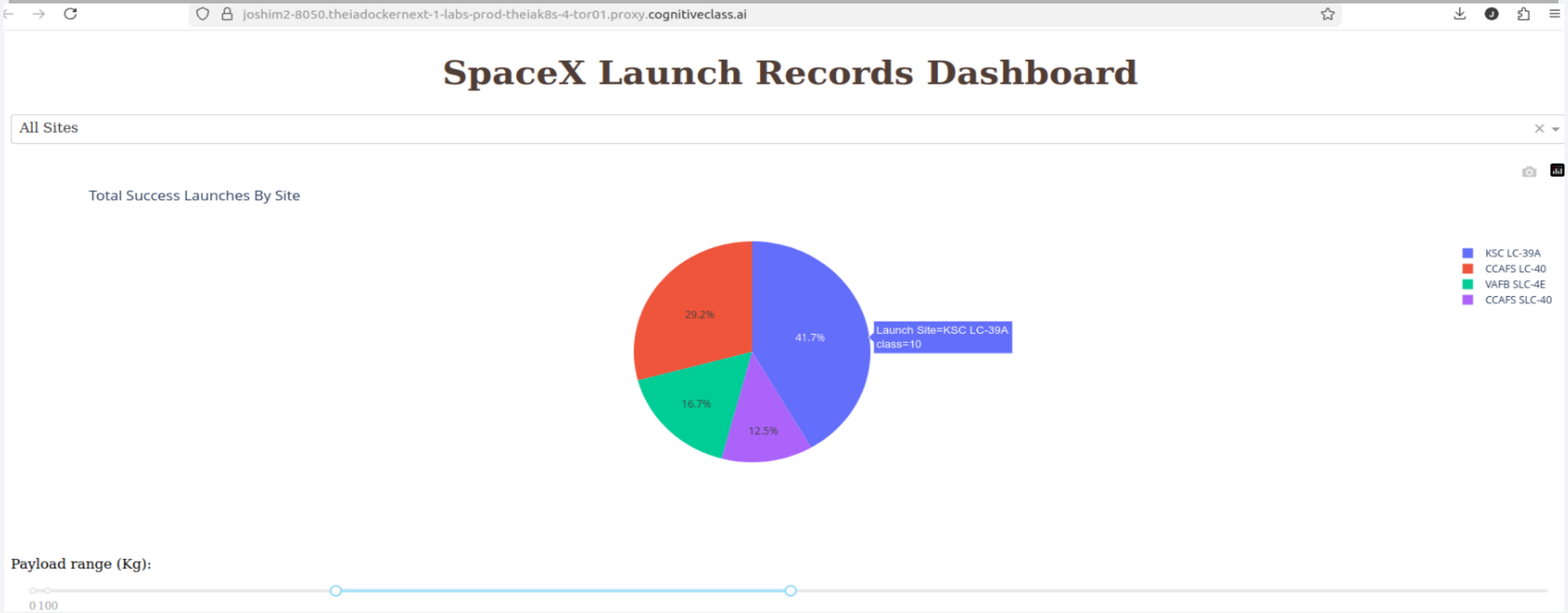
Build a Dashboard with Plotly Dash

The following table lists components used in buiding the project dashboard using Plotly Dash. It also lists the reasoning behind the selection of the various components.

Dashboard Components & Interactions	Reasons for addition of these Components
Launch Site Dropdown Menu: An interactive selector that allows users to toggle between "All Sites" or specific locations (e.g., KSC LC-39A).	**Site-Specific Deep Dives**: The Dropdown and Pie Chart were added to identify which launch pads have the highest reliability. This helps in understanding if certain sites (like KSC LC-39A) provide any advantages for landings. **Correlation Identification:** The Scatter Plot was added to determine if heavier payloads negatively impact landing success. By color-coding by Booster Version, one can visually track how newer hardware (like Block 5) handle varying weights more successfully than older versions. **Data Granularity:** The Range Slider allows data analyst to "zoom in" on specific mission profiles (e.g., small satellites vs. heavy Starlink batches) to see if there is any spot/range for recovery success.
Payload Range Slider: A dual-point slider used to filter launch data by payload mass (from 0 to 10,000 kg).	
Success Rate Pie Chart: A dynamic chart that updates based on the selected site to show the ratio of successful vs. failed missions.	
Success vs. Payload Scatter Plot: A multi-dimensional plot that displays the relationship between payload mass and landing outcome, color-coded by the Booster Version Category.	
Access Full Notebook on GitHub: https://github.com/mmj009/Data-Science-Capstone/blob/main/spacex-dash-app.py	

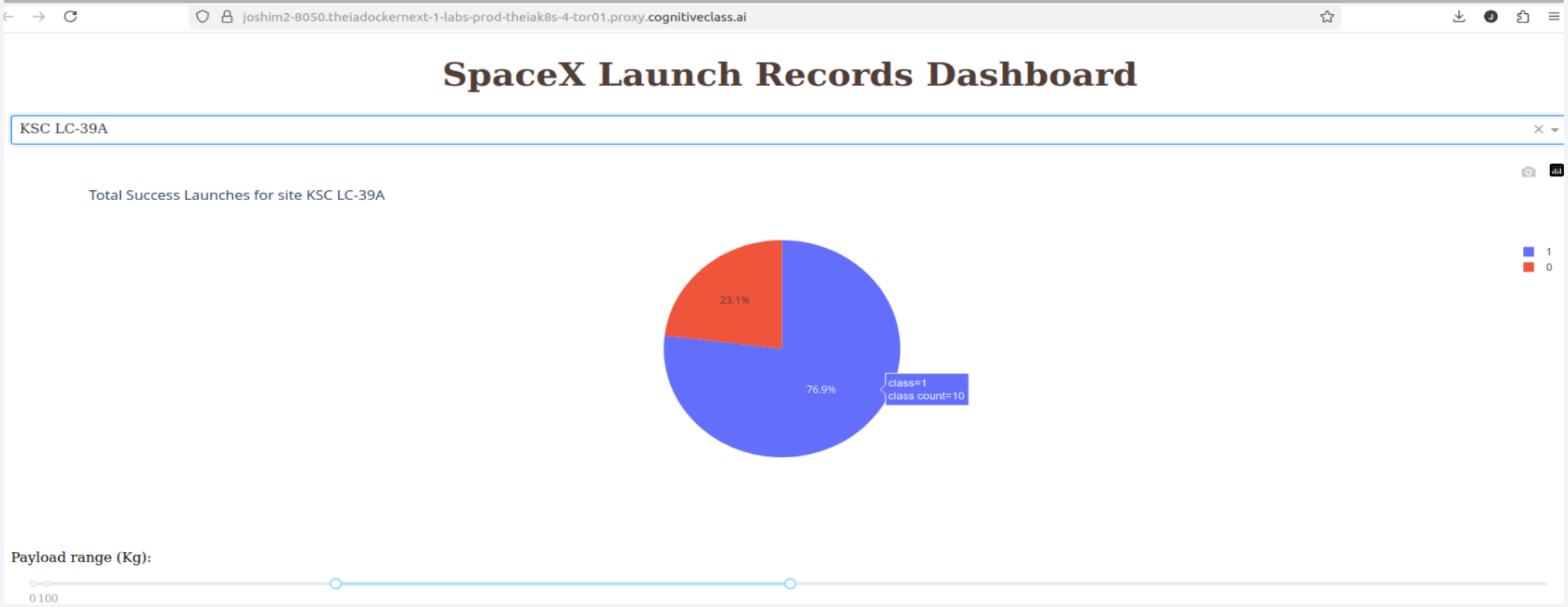
Total Success Launches By Sights

The pie chart screenshot shows Total Success Launches Site-Wise. A specific site or ‘All Sites’ can be selected from the pull down list. The Payload range slider allows one to select desired payload range. **Insights**:The site ‘KSC LC-39A’ has the highest successful launches amongst all the launch sites.



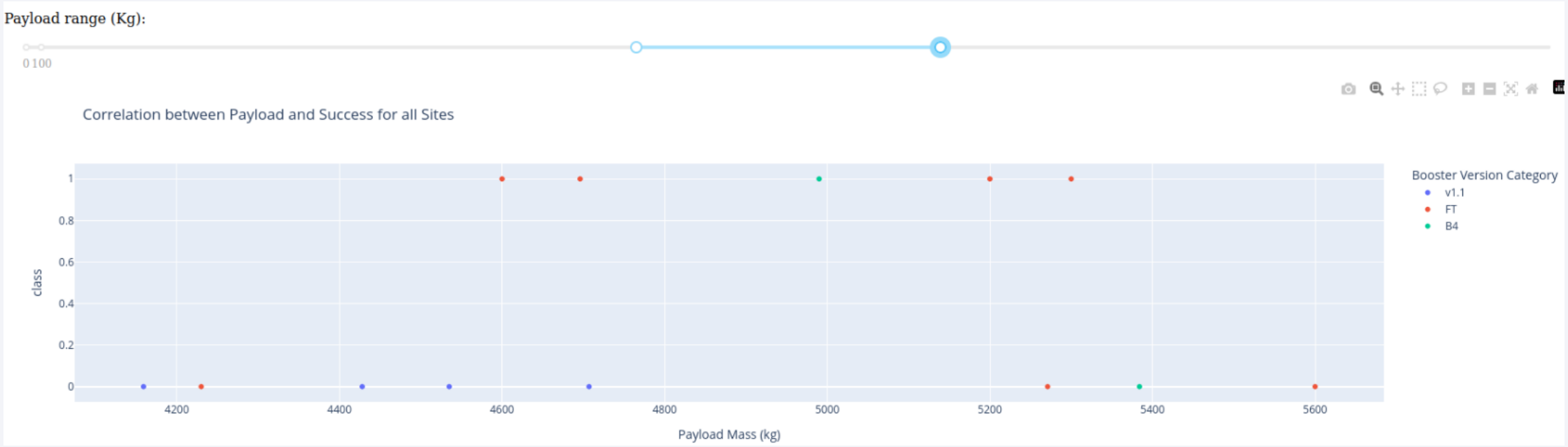
Total Success Launch Rate for KSC LC 39-A

The adjacent pie chart screenshot shows the launch site ‘KSC LC 39-A’ has success rate of 76.9%. The legend ‘1’ indicates success while ‘0’ indicates failure.
Insights: The site ‘KSC LC-39A’ is the most succesful site amongst all the launch sites in terms of successful launches.



Payload vs. Launch Outcome scatter plot for all sites

The adjacent pie chart shows ‘All Sites’ were selected and the scatter plot for all the sites screenshot shows the launch success rate for payload range between 4000Kg and 5600Kg.
Insights:The scatter plot shows lowest launch success rate for the selected payload range 4000Kg and 5600Kg.
This is helpful in further Risk Assessments study compared to looking only at the Highest Success Rate.



Section 5

Section 6

Predictive Analysis (Classification)

Predictive Analysis Methodology (Chart)

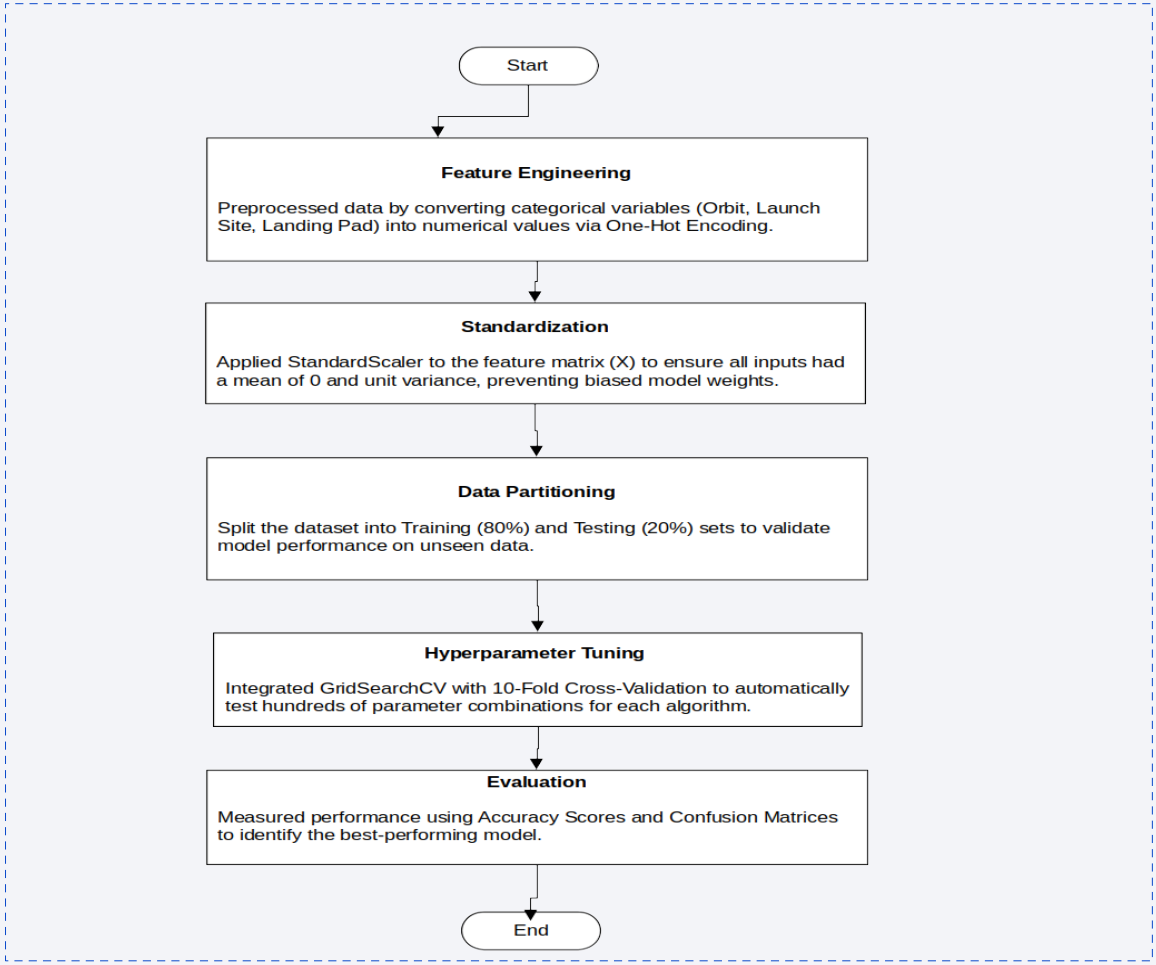
- **Reasoning Behind the methodology used**
  - **Efficiency**

    Used automated search (GridSearchCV) to find the better parameters like C (Regularization) in Logistic Regression and n_neighbors in KNN.
  - **Robustness**

    Cross-validation ensured that the model's accuracy was not guess work on a specific slice of data, but a consistent result across the entire training set
  - **Optimization**

    Focused on minimizing False Positives to ensure landing predictions were as reliable as possible for cost-estimation purposes.
- **Access Full Notebook on GitHub:**

  https://github.com/mmj009/Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Predictive Analysis (Classification)

- **Uniform Performance**

  All four classification models, Logistic Regression, SVM, KNN, and Decision Tree achieved an identical accuracy score of 83.33% on the **test** dataset.

- **Consistency**

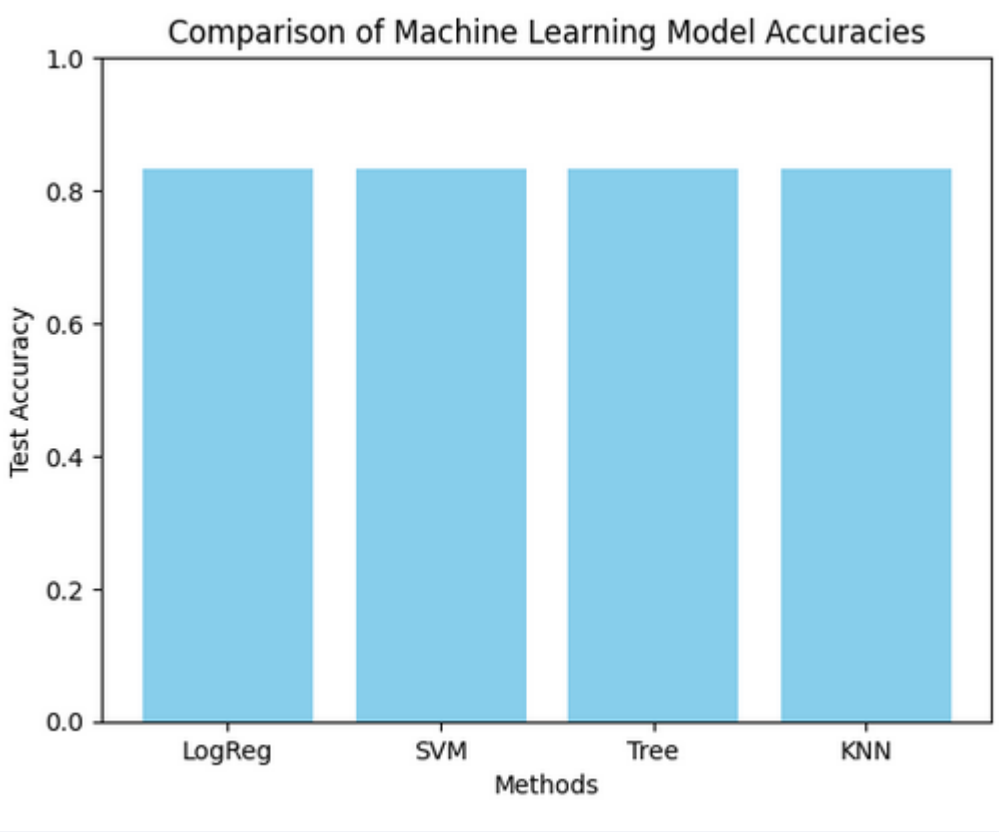
  The fact that different algorithms (linear, distance-based, and tree-based) converged on the same result indicates that the provided features Payload, Orbit, Site have a very strong and consistent predictive signal.

- **Accuracy Ceiling**

  An accuracy of 83.33% suggests that the models have extracted the maximum predictive value from the given dataset.

  The balance 16.67% of unpredictable component is likely due to factors like,

  - A small data set was used create and test all the four models,
  - Other data bits related with weather, hardware sensory data, software failures not being captured for success and failure cases.



Confusion Matrix

- **Confusion matrix is same for all the four classification models considered.**
- **Quantitative Breakdown**

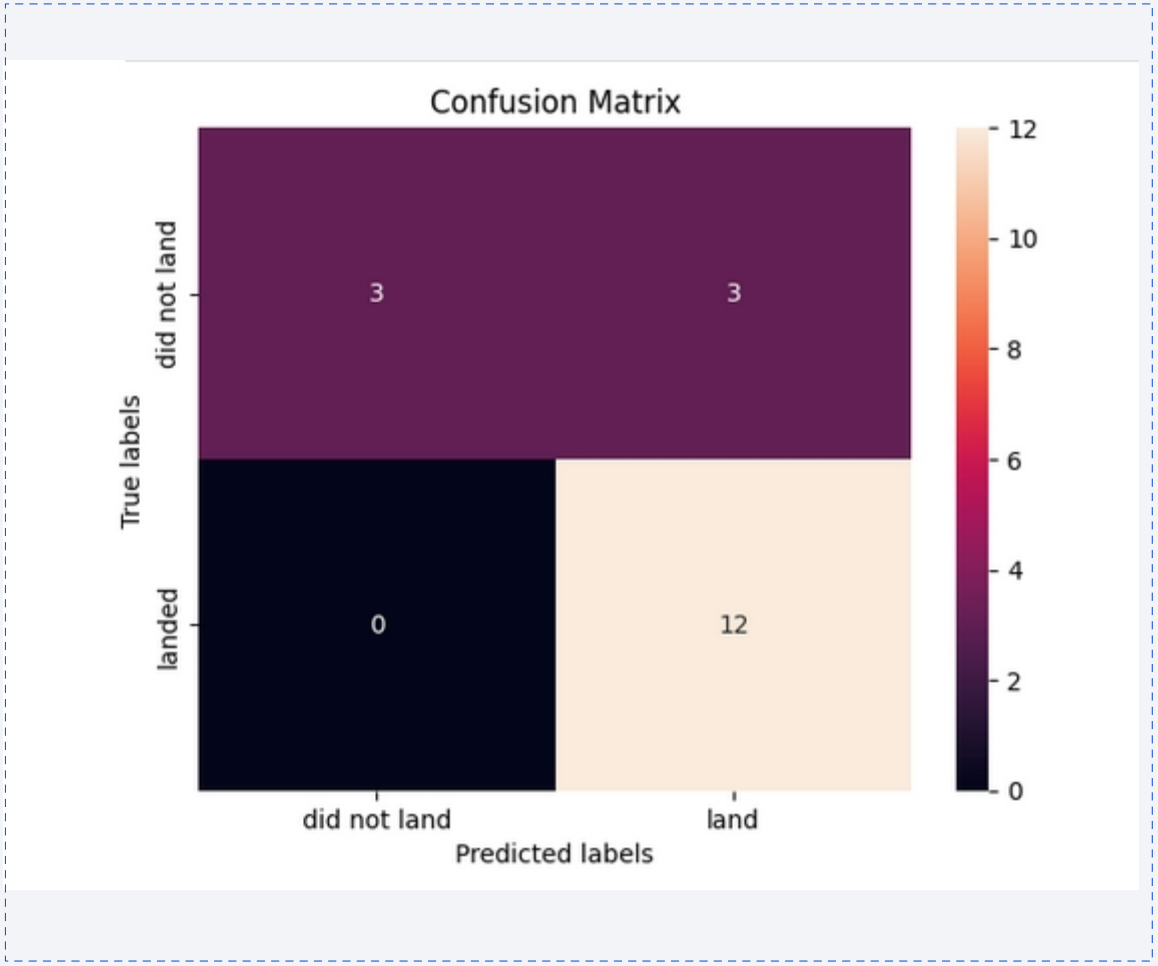
  The models correctly predicted 15 out of 18 landing outcomes in the test set.
- **True Positives (12)**

  The models were 100% accurate at predicting successful landings when the conditions were right.
- **False Positives (3)**

  The models predicted a "Success" for 3 missions that actually failed.
- **Insight**

  The model views the mission profile based on Orbit, Payload, Site as success maker, but it looks like the actual failures were likely caused by other variables which are either not captured or not provided in the given dataset.
- **Conclusion**

  The model is certainly optimistic and highly reliable for identifying the conditions required for a successful recovery.



Conclusions

- **Predictive Success**

  We successfully developed a machine learning pipeline capable of predicting SpaceX Falcon 9 landings with 83.33% accuracy.

- **Experience is the King**

  Landing success rates increased significantly over time (Organizational Learning).

- **Orbit & Site Correlation**

  Certain orbits (ES-L1, SSO) and specific launch pads (KSC LC-39A) are statistically safer for booster recovery.

- **Business Impact**

  This model allows a competitor or insurance firm to estimate the real cost of a SpaceX launch by calculating the probability of booster loss, a key component of SpaceX's $62M price point.

- **Future Improvements**

  Integrate weather and other sensory data (wind speed, visibility) to reduce False Positives.
  Include booster "age" (number of previous re-flights) to account for mechanical wear and tear.

Results

Exploratory Data Analysis (EDA)

- **Trend**
  Success rate increased from 0% (2010) to over 90% (2020).

  Orbit Impact
  100% success rates observed for ES-L1, GEO, HEO, and SSO orbits.

Interactive Analytics (Folium & Dash)

- **Proximity**
  All launch sites are strategically located within **1 KM** of the coastline for safety and near highways/railways for logistics.

- **Dashboard**
  Identified that the KSC LC-39A site and Payloads between 3,000–7,000 kg represent the "optimal" success profile.

Predictive Analysis

- **Top Performance**
  All models (LogReg, SVM, KNN, Tree) achieved a consistent **83.33%** accuracy.

- **Robustness**
  The models are highly reliable at predicting successful landings, with errors primarily occurring due to unpredictable external mission variables.

Appendix

Primary Data Sources

- **SpaceX REST API:** The primary source for historical launch, rocket, and core data. (Source: https://api.spacexdata.com/v4/)
- **Wikipedia (SpaceX Launch List):** Used for web-scraping historical mission details, payload weights, and specific landing outcomes to augment API data.
- **IBM Cloud / Skills Network:** Provided the SQL environment, DB2 database hosting, and Jupyter Notebook infrastructure for analysis.

Technical Reference

- **Programming Language:** Python 3.x
- **Primary Libraries:**
  Data Wrangling: Pandas, NumPy
  Visualization: Matplotlib, Seaborn, Folium, Plotly Dash
  Machine Learning: Scikit-Learn
  Database Interaction: SQL / SQLite
  Geospatial Logic: Haversine Formula for distance calculations between launch sites and coastline infrastructure.

External Documentation

- **Full Project Repository**: https://github.com/mmj009/Data-Science-Capstone

Thank you!