Race to Launch: Powered by Data

Name : Brandi Dennis
Date: June 6, 2025

Executive Summary

- This capstone project analyzes SpaceX launch data to understand success drivers and predict future launch outcomes.
- We collected data via API and web scraping, performed EDA using SQL and visualization, and built classification models.
- Key outcomes: Launch site, payload, and booster version significantly affect success rates.

Introduction

SpaceX aims to make space travel **reliable and reusable**.

Our objective: **analyze historical launch data** to identify success patterns and **predict future launch outcomes**.

In this capstone, we predict whether the **Falcon 9 first stage will land successfully**.

A successful landing reduces launch costs:

SpaceX: **$62 million**
Other providers: **$165+ million**
Reusability is key to SpaceX's lower costs.

If we can predict landing success, we can estimate **launch cost**—useful for **competitive bids** from other companies.

Methodology

Methodology - Executive Summary

1. **Data Collection, Wrangling, and Formatting**
   - Collected data via the SpaceX REST API and web scraping from Wikipedia.
   - Cleaned, merged, and formatted data using Pandas and custom preprocessing logic.
2. **Exploratory Data Analysis (EDA)**
   - Performed EDA using Visualisation.
   - Used SQL queries to extract insights directly from structured data.
3. **Data Visualization**
   - Created static visualizations using Matplotlib and Seaborn.
   - Built interactive maps with Folium to display launch site locations and outcomes.
   - Developed a dynamic dashboard using Plotly Dash for user-driven exploration.
4. **Machine Learning Prediction**
   - Built classification models using Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
   - Tuned models with GridSearchCV and evaluated performance using accuracy metrics and confusion matrices.

Data Collection, Wrangling, and Formatting

1. SpaceX API

- The SpaceX REST API was used to collect structured JSON data about Falcon 9 launches.
- Key data points extracted included:
  - Launch site
  - Rocket type
  - Payload mass
  - Launch outcome
  - Landing type and success
- API requests were made using the `requests` library, and data was transformed into DataFrames for analysis.

. Wikipedia Scraping

- Historical launch records were scraped from the Wikipedia page:
  List of Falcon 9 and Falcon Heavy launches
- Used **HTTP GET requests** to retrieve the webpage content.
- Parsed the HTML using **BeautifulSoup** to locate the relevant launch tables.
- Extracted the tables and converted them into **Pandas DataFrames** for structured analysis.
  -

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...

Screenshot of the structured Falcon launch dataset extracted from the SpaceX AP.

```
1
4 June 2010
18:45
F9 v1.07B0003.18
CCAFS
Dragon Spacecraft Qualification Unit
Dragon Spacecraft Qualification Unit
LEO
SpaceX
Success
```

Sample console output showing extracted Falcon 9 launch details — including flight number, date, time, booster version, launch site, payload, orbit, customer, launch outcome, and booster landing status — as parsed row-by-row from the Wikipedia launch tables.

Data Collection, Wrangling, and Formatting

Handling Missing Values & Feature Engineering

- Calculated the mean of **PayloadMass** to handle missing values.
- Replaced missing **PayloadMass** entries with the computed mean.
- Created a **landing outcome** label derived from the **Outcome** column.

Exploratory Data Analysis (EDA) s

The data is explored and analyzed using SQL queries to uncover key insights. The analysis includes:

- Retrieving **unique launch sites** used by SpaceX
- Filtering launch records with sites beginning with `'CCA%'`
- Calculating **total payload mass** for **NASA (CRS)** missions
- Computing the **average payload mass** for **booster version F9 v1.1**
- Ranking **landing outcomes** (e.g., success, failure) between 2010–2017

Data Visualization

The following charts were plotted to analyze Falcon 9 launch data:

- **Categorical Scatter Plot**
  - **X:** Flight Number | **Y:** Payload Mass | **Hue:** Launch Success (Class)
  - Showed how payload size correlated with mission success over time
- **Categorical Scatter Plot**
  - **X:** Flight Number | **Y:** Launch Site | **Hue:** Launch Success
  - Helped identify launch sites with higher success rates
- **Scatter Plot**
  - **X:** Payload Mass | **Y:** Launch Site | **Hue:** Class
  - Explored variation in payloads across launch locations and outcomes

These visualizations provided intuitive, high-level insights into mission patterns and outcomes.

Data Visualization

objective:
Demonstrate how SpaceX launch data varies across geographical locations.

Tools Used:
Folium (for maps)
Pandas (for data handling)

Key Visuals:

- **Launch Site Locations:** Map showing all launch site coordinates.
- **Outcome Markers:** Colored icons indicating launch success or failure.
- **Payload Radius:** Scaled circles showing payload mass per site.

Insights:

- Majority of launches occur from coastal sites, likely due to safety protocols.
- Specific launch pads show higher success rates.
- Larger payloads tend to launch from particular high-capacity sites.

Data Visualization

Built an interactive web dashboard using **Dash** and **Plotly Express** for exploratory data analysis:

a. Dropdown Menu (Launch Site Filter)

- Enabled users to filter results by **individual sites** or view all sites combined.

b. Pie Chart Visualization

- Displayed **success ratios**:
    - For all sites: distribution of successful launches across different locations.
    - For a specific site: success vs failure proportions.

c. Payload Range Slider

- Implemented a **range slider** to dynamically filter launch records based on **payload mass (kg)**.

d. Scatter Plot Visualization

- Correlated **payload mass** with **launch success**.
- Color-coded by **Booster Version Category** to identify trends across hardware types.

Machine Learning Prediction

1. Data Preparation

- Target Variable:
  Y = pd.Series(data['Class'].to_numpy())

- Feature Scaling:
  Standardized X using StandardScaler

2. Data Splitting

- Train-Test Split:
  train_test_split(X, Y, test_size=0.2, random_state=42

Machine Learning Prediction

3. Model Training & Tuning

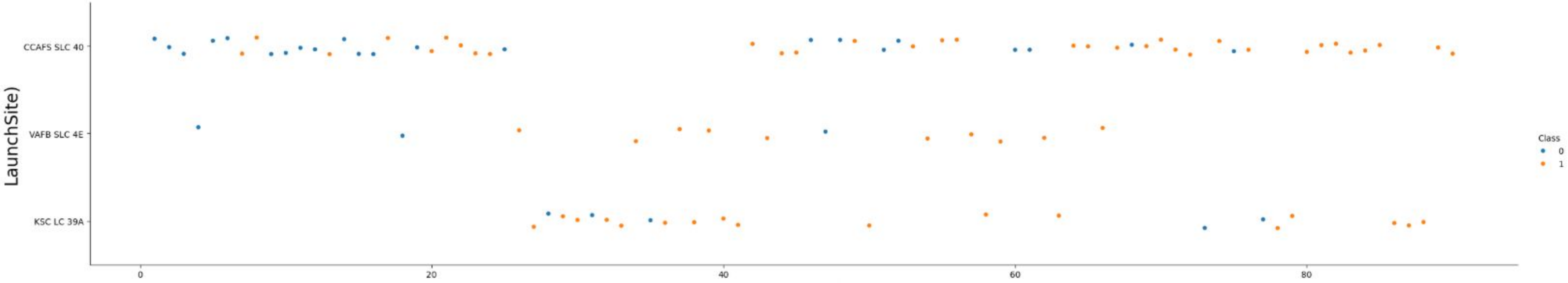
- Used **GridSearchCV** with **10-fold CV**
- Models & Hyperparameters Tuned:
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Decision Tree
    - K-Nearest Neighbors (KNN)

4. Model Evaluation

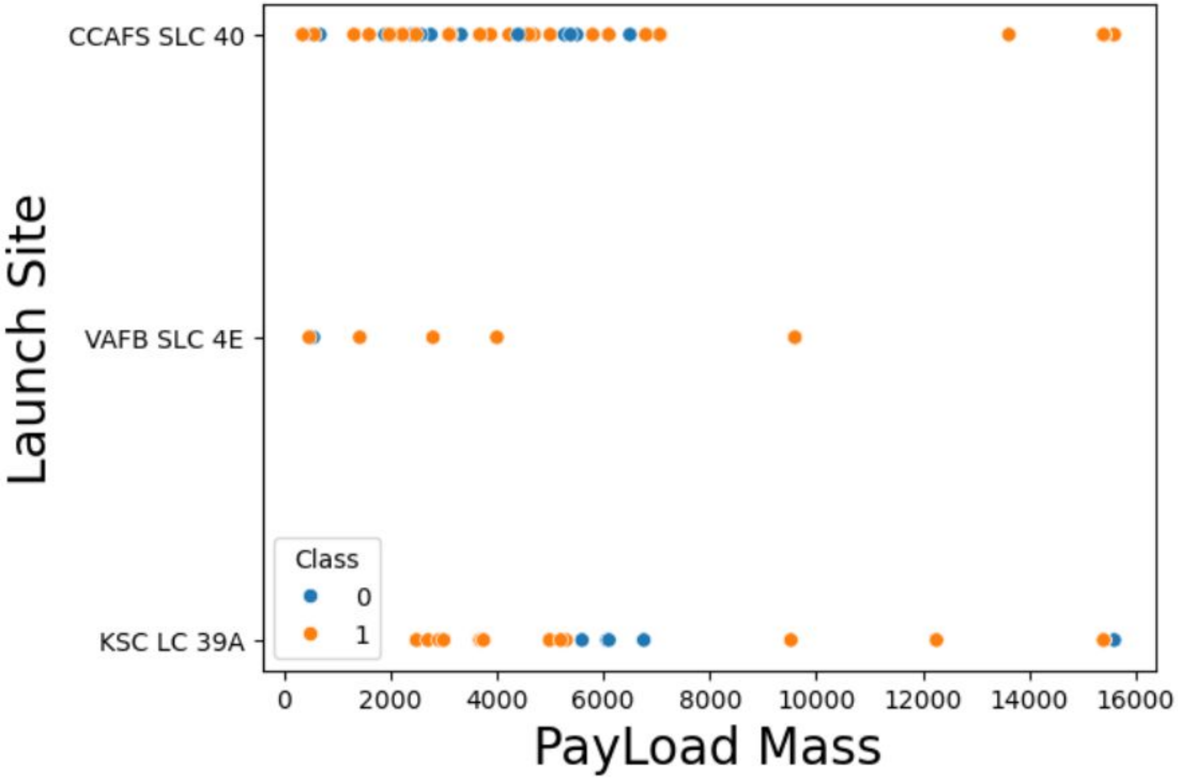
- Metrics Evaluated:
    - **Accuracy**, **F1 Score**, **Recall**
    - Tested on both: **Test Set** & **Full Dataset**
- Confusion matrices plotted for all models

Results

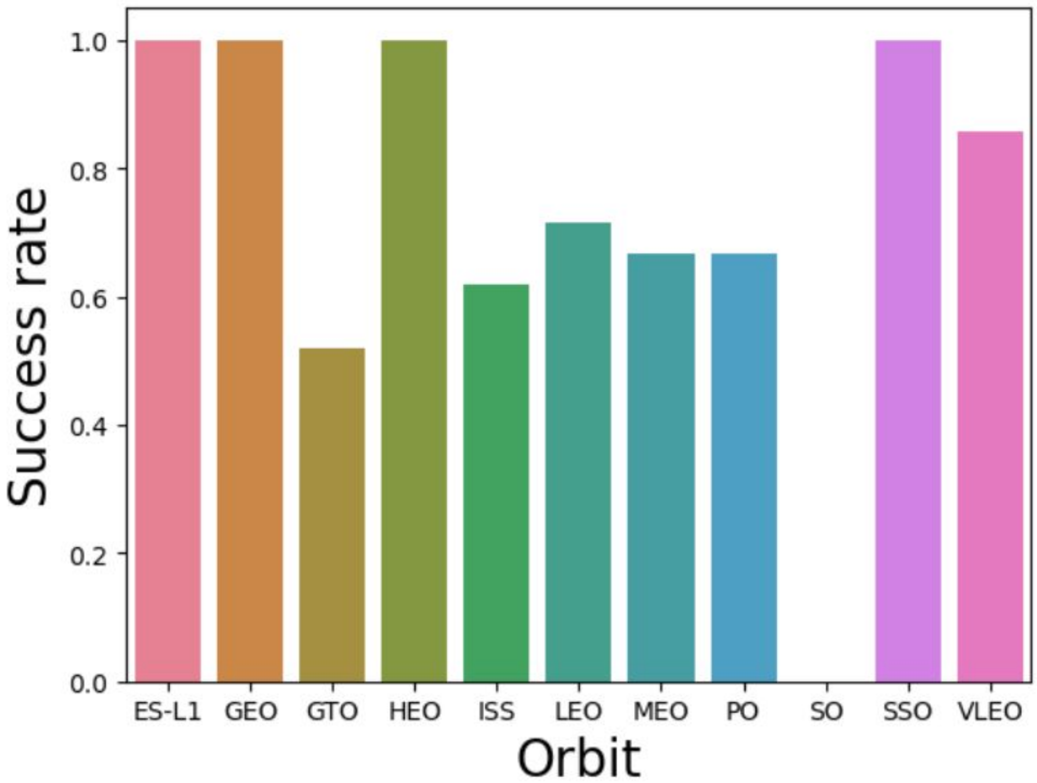
The following slides are the insights uncovered during this project. They present key results from data queries, visualizations, geospatial maps, and analytical findings.



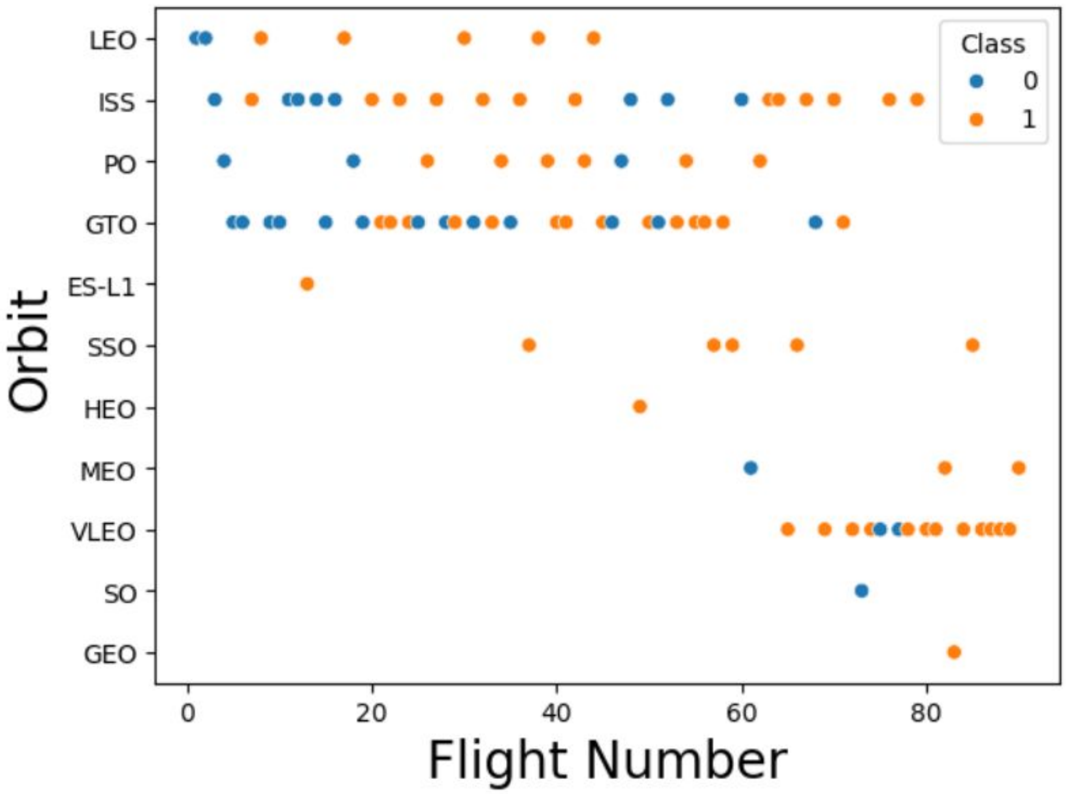
Flight Number vs. Launch Site



Payload vs. Launch Site



Success Rate vs. Orbit Type



Flight Number vs. Orbit Type

All Launch Site Names

To identify the different launch sites used by SpaceX, we executed the following SQL query:

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

The resulting table lists all unique launch sites from the dataset:

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

Launch Site Names Begin with 'CCA

To isolate launch sites located at Cape Canaveral, we used a pattern match on the Launch_Site column:

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

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

To calculate the total payload mass carried for NASA’s Commercial Resupply Services (CRS) missions this query was run

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

The SUM function adds up all the payload masses in kilograms.

The FROM clause tells the database which table to use — in this case,

The WHERE clause filters the data to include only the rows where the customer is NASA (CRS), so the total represents payloads from those specific missions.

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

To analyze the performance of the F9 v1.1 booster version, we used the following SQL query:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as 'F9 v1.1 Average Payload Mass'FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

Result: This query calculates the average payload mass (in kg) launched by SpaceX using the F9 v1.1 booster.

This helps assess the typical load this version carried, giving insight into its operational capacity and evolution compared to newer boosters.

F9 v1.1 Average Payload Mass
2928.4

First Successful Ground Landing Date

```
%%sql

SELECT Min(Date) AS 'First Success in Ground Pad'

FROM SPACEXTABLE

  WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

First Success in Ground Pad
2015-12-22

The MIN function returns the earliest date from the selected rows.

The FROM clause tells the database to use the SPACEXTABLE.

The WHERE clause filters records where the Landing_Outcome is a successful ground pad landing.

This query shows the date when the first successful ground pad landing happened.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

SELECT Booster_Version AS 'Success in Drone Ship where 4000 < plm < 6000'

FROM SPACEXTABLE

WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000

  AND Landing_Outcome LIKE 'Success (drone ship)';
```

Success in Drone Ship where 4000 < plm < 6000
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

The SELECT statement chooses the booster versions that meet the criteria.

The FROM clause uses the SPACEXTABLE.

The WHERE clause filters rows where the payload mass is between 4000 and 6000 kg and the landing outcome is a successful drone ship landing.

This lists the booster versions with successful drone ship landings carrying payloads in that weight range.

Total Number of Successful and Failure Mission Outcomes

The CASE statement categorizes mission outcomes into Success, Failure, or Other.

The FROM clause specifies SPACEXTABLE.

The WHERE clause filters rows to include only those with success or failure outcomes.

The GROUP BY groups the results by outcome type.

This query counts the number of successful and failed missions.

Outcome_Type	Mission_Count
Failure	1
Success	100

Boosters Carried Maximum Payload

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

The subquery (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) finds the maximum payload mass in the table.

The main query selects booster versions where the payload mass equals this maximum value.

The FROM clause uses SPACEXTABLE.

This returns the booster versions that carried the heaviest payloads.

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 CASE statement converts the month number in the date into the month name.

The FROM clause uses SPACEXTABLE.

The WHERE clause filters for drone ship landing failures in the year 2015.

This lists details of failed drone ship landings in 2015, showing the month, date, booster, and launch site.

Month	Date	Booster_Version	Launch_Site	Landing_Outcome
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

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

Landing_Outcome	Landing_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

Launch Sites Marked

Filtered SpaceX data to extract unique launch site names and coordinates.

Created a folium.Map() centered on the average coordinates.

Plotted launch site locations using folium.Marker().

All launch sites are near the coast.

Success/Failed Launches for each site

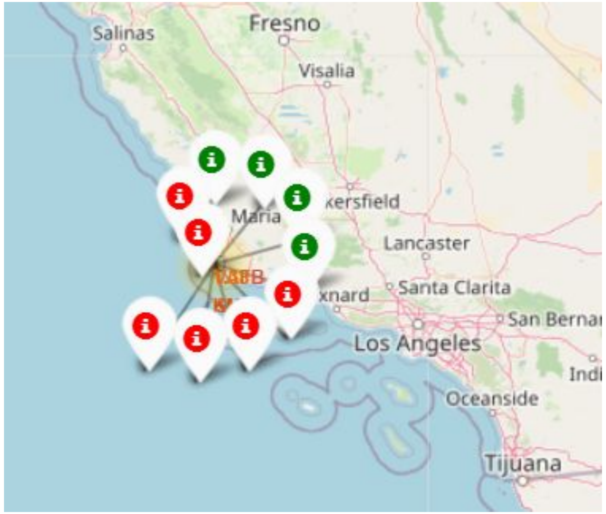
Used marker_cluster to group overlapping points.

Color-coded markers:

Green = Success (class = 1)

Red = Failure (class = 0)

KSC LC-39A shows high success rate (mostly green).



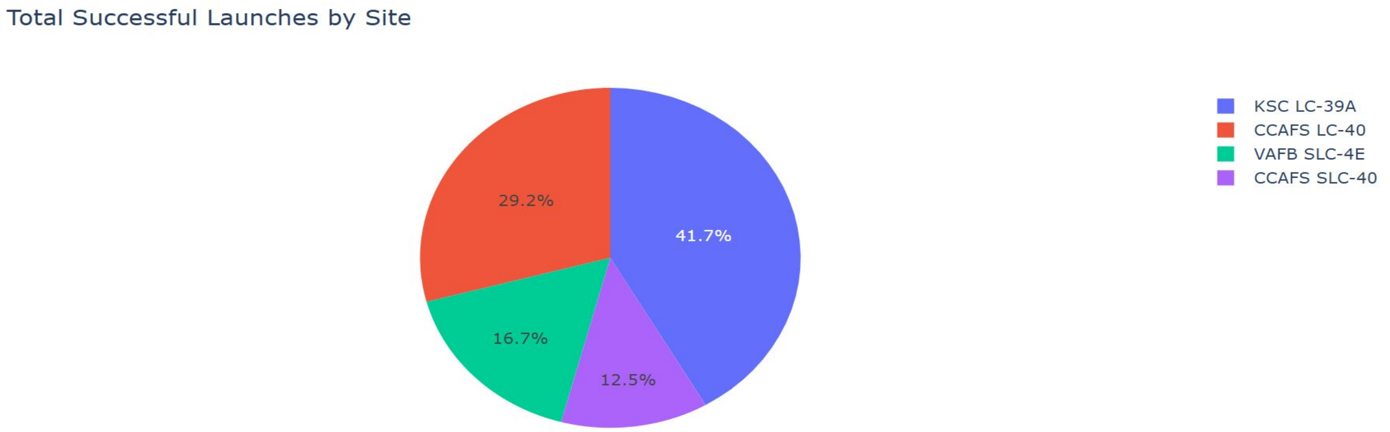
Distances marked

Used MousePosition to collect proximity coordinates.

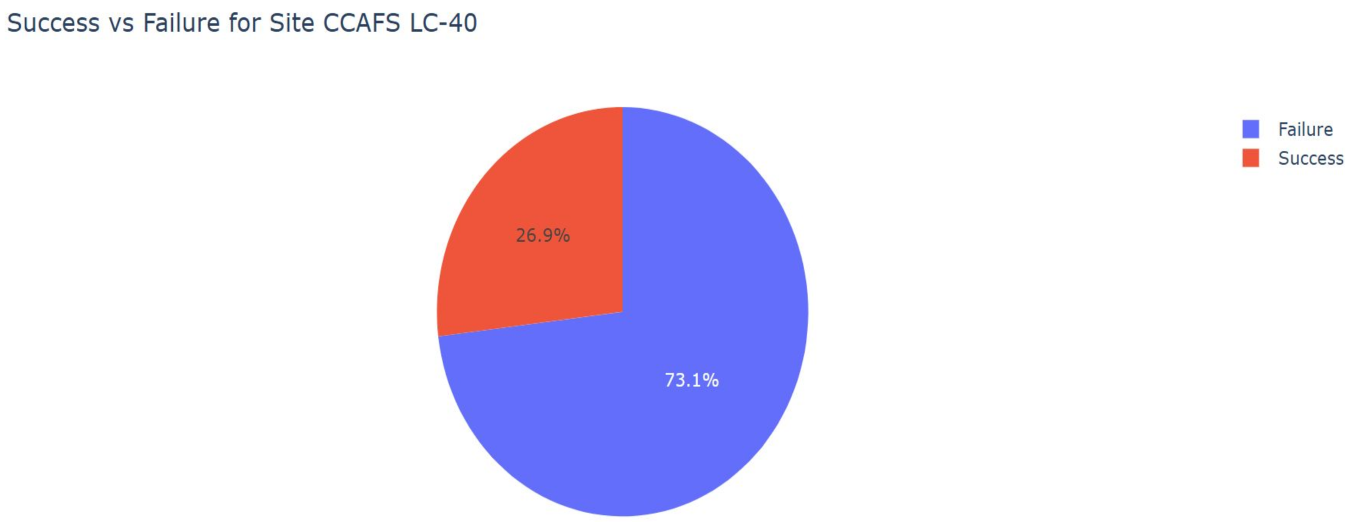
Coastline: Launch sites are very close to the ocean — about 1.4 kilometers away. Being near the coast is smart because rockets can safely fly over water instead of people.

Cities: They keep a good distance from cities — almost 40 kilometers away. This helps keep people safe from any accidents and reduces noise or other disruptions.

Successful Launches by Site



Worst performing site



Correlation between Payload and Success

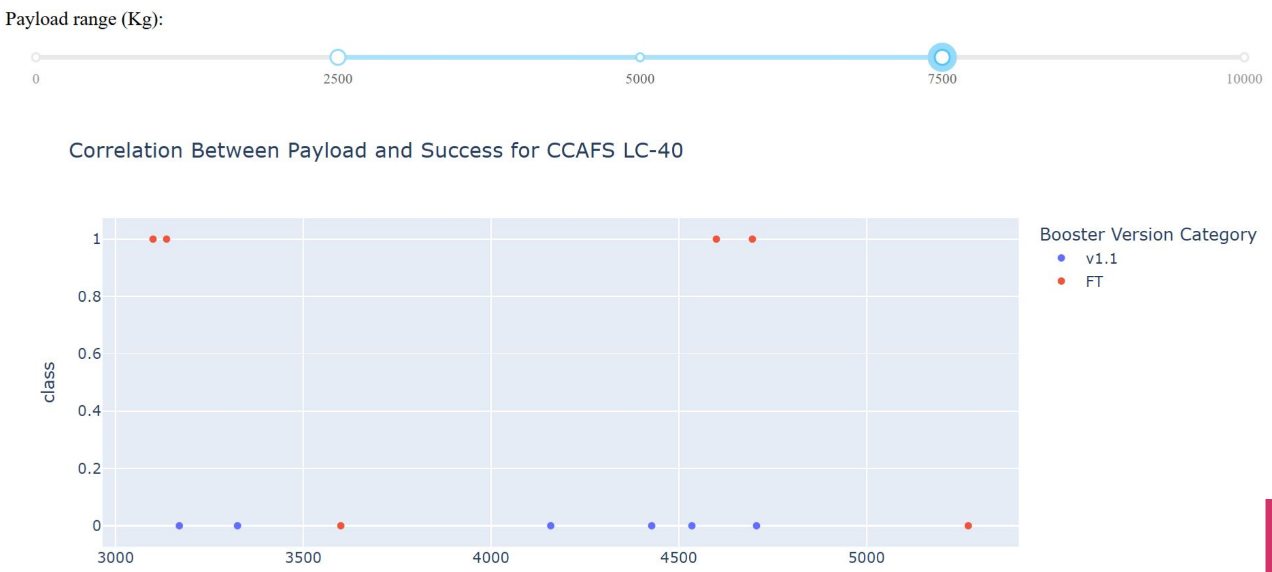


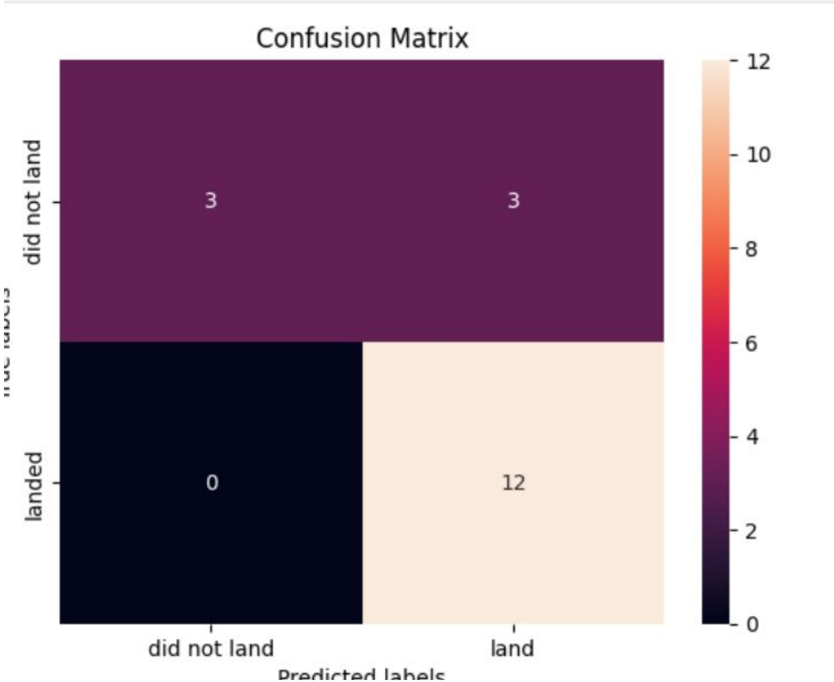
Table Comparing the Models

Model	Accuracy (Test)	F1 Score (Test)	Recall (Test)	Accuracy (Full)	F1 Score (Full)	Recall (Full)
Logistic Regression	0.8333	0.8889	1.0	0.8667	0.9091	1.0000
SVM	0.8333	0.8889	1.0	0.8778	0.9160	1.0000
Decision Tree	0.8333	0.8889	1.0	0.9000	0.9302	1.0000
KNN	0.8333	0.8889	1.0	0.8556	0.9008	0.9833

Based on the test and full dataset performance:

- All models performed equally well on the test set, with an accuracy of 83.33%, an F1 score of 0.8889, and a recall of 1.0, indicating they all correctly identified every positive case.
- On the full dataset:
  - Decision Tree performed the best overall, with the highest accuracy (90%) and F1 score (0.9302), and maintained a perfect recall (1.0).
  - SVM also showed strong performance with 87.78% accuracy and F1 score of 0.9160.
  - Logistic Regression followed closely behind with 86.67% accuracy.
  - KNN had the lowest full dataset scores, particularly in recall (0.9833), though still strong.

Decision Tree Confusion Model



Conclusion

Comprehensive analysis of SpaceX launch data using visual analytics and machine learning

Explored key factors affecting mission success with Matplotlib, Seaborn, Folium, and Plotly Dash

Interactive maps and visualizations revealed patterns in launch sites, payloads, and boosters

Developed and evaluated models: Logistic Regression, Decision Trees, SVM, KNN

GitHub Url : https://github.com/voilabrandi/IMB-Applied-Data-Science-Capstone