IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Subhashini Ananth
25-Jan-2026

Table of Contents

Executive Summary

This project focuses on predicting the successful landing of the Falcon 9 rocket's first stage using data science and machine learning techniques. Since SpaceX's reduced launch cost is largely driven by first-stage reusability, accurately predicting landing success is crucial for estimating launch costs and enabling competing companies to develop effective bidding strategies.

Summary of methodologies

- Historical launch data was collected from the **SpaceX REST API** in JSON format and transformed into a structured dataset using data normalization techniques.
- Data cleaning and wrangling were performed to handle missing values and convert complex landing outcomes into a binary classification variable representing successful and unsuccessful landings.
- Exploratory Data Analysis (EDA) was conducted using **SQL queries** and **Python-based visualizations** to identify patterns related to launch sites, payload mass, orbit types, and flight experience.
- Feature engineering techniques such as one-hot encoding were applied to categorical variables, and numerical features were standardized to prepare the dataset for machine learning.
- Multiple supervised classification models—**Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree Classifier**—were trained and optimized using **GridSearchCV**.
- Model performance was evaluated using test accuracy and confusion matrices.

Summary of all results

- EDA revealed that landing success rates improved significantly over time, with later missions showing higher success probabilities due to increased operational experience.
- Payload mass and orbit type demonstrated strong non-linear relationships with landing outcomes, while launch sites near coastlines exhibited higher safety and success rates.
- All machine learning models achieved similar baseline accuracy of approximately **83.33%** before hyperparameter tuning. After optimization, all models showed improved performance; however, the **Decision Tree classifier outperformed the others**, achieving the highest test accuracy of **87.68%**.
- The results indicate that landing success is influenced by a combination of flight number, payload mass, orbit type, and launch site, and that rule-based models such as Decision

Introduction

Project background and context

The commercial space sector has been transformed by the adoption of reusable launch vehicles, significantly reducing the cost of space access. SpaceX has achieved a major cost advantage through the successful recovery and reuse of the Falcon 9 first-stage booster, enabling launches at substantially lower costs compared to traditional expendable rockets. Predicting first-stage landing success is therefore essential for estimating launch costs and supporting competitive decision-making in the aerospace industry

Problems require answers

The primary objective of this project is to determine whether the successful landing of the Falcon 9 first stage can be predicted using historical launch data and machine learning techniques. To achieve this objective, the project seeks to answer the following key questions:

This project aims to determine

1. Whether the successful landing of the Falcon 9 first stage can be predicted using historical launch data and machine learning techniques.

2. Specifically, it seeks to identify the key factors influencing landing outcomes. Which launch-related factors (such as payload mass, orbit type, launch site, and flight experience) have the greatest influence on landing outcomes? How does operational experience, represented by flight number, impact landing success rates over time?

3. Which machine learning algorithm provides the most accurate and reliable predictions for landing success? Evaluate the impact of operational experience on success rates, and compare multiple machine learning models to determine the most effective approach for accurate landing success prediction.

4. How can these predictive insights be used to estimate launch costs and support competitive bidding strategies for alternative launch providers?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology

Historical Falcon 9 launch data was collected using the **SpaceX REST API**. The API provides launch-level information such as rocket configuration, payload characteristics, launch site details, orbit type, and landing outcomes.The API responses were returned in **JSON format**, where each JSON object represented a single launch event.

- Perform data wrangling

Once the data was loaded into a DataFrame: Nested JSON fields (rocket, payloads, cores) were flattened, Relevant attributes related to landing success were extracted, Columns were renamed and standardized for clarity, Inconsistent formats and redundant attributes were removed.This ensured that the dataset was suitable for both exploratory analysis and machine learning modeling. Data wrangling was performed to handle: Missing payload mass values, Incomplete landing outcome records, Boolean feature inconsistencies

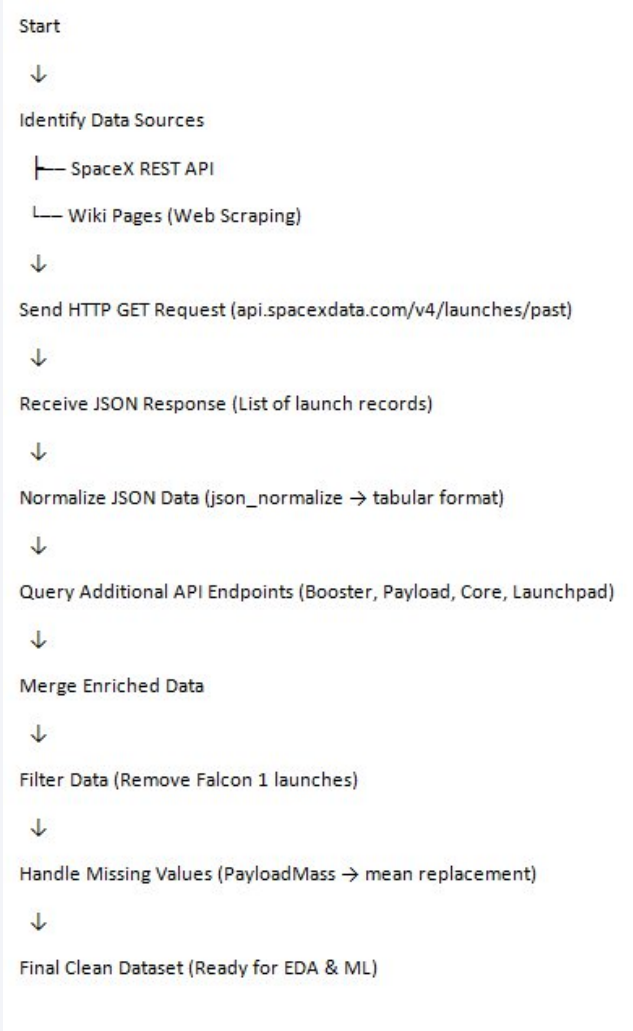
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Standardized data for effective model training. Dataset is divided into training and testing sets using the Train_test_split method. Various algorithms, including Logistic Regression, Support Vector Machines, Decision Tree Classifier, and K-nearest neighbors, have been tested. A Grid Search has been performed to identify the best hyperparameters for optimal model performance The model's accuracy assessed using the test data A confusion matrix was generated to evaluate the model's predictive performance

Data Collection

Dataset Collection Process:

- **SpaceX REST API**, a publicly available data source that provides detailed information on SpaceX missions. The API endpoint **api.spacexdata.com/v4/launches/past** was used to retrieve historical Falcon 9 launch records.
- Responses were obtained in **JSON format**, where each JSON object represented a single launch event.
- json_normalize() function was used to convert the structured JSON responses into a flat, tabular format suitable for analysis.
- To enrich the dataset, additional API endpoints were queried to retrieve detailed information associated with identifier fields such as **boosters, launchpads, payloads, and cores**. These API calls allowed the replacement of ID references with meaningful launch attributes.
- The dataset was then filtered to include **only Falcon 9 launches**, excluding Falcon 1 missions to maintain project relevance.
- In addition to API-based data collection, **web scraping** was performed using **BeautifulSoup** to extract supplementary Falcon 9 launch records from structured HTML tables available on public Wiki pages.
- The scraped data was parsed and converted into Pandas DataFrames for consistency and further analysis



Data Collection – SpaceX API

Data collection with SpaceX REST calls:

- SpaceX REST API call: use the API to extract information using identification numbers in the launch data.
- Requesting rocket launch data from SpaceX API with the following URL API endpoint (https://api.spacexdata.com/v4/launches/past)
- Request and parse the SpaceX launch data using the GET request
- Use json_normalize meethod to convert the json result into a dataframe
- Filter the dataframe to only include Falcon 9 launches

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1060	-80.603956	28.608058
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	13	B1058	-80.603956	28.608058
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1051	-80.603956	28.608058
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	12	B1060	-80.577366	28.561857
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	8	B1062	-80.577366	28.561857

GitHub Repository URL:

https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

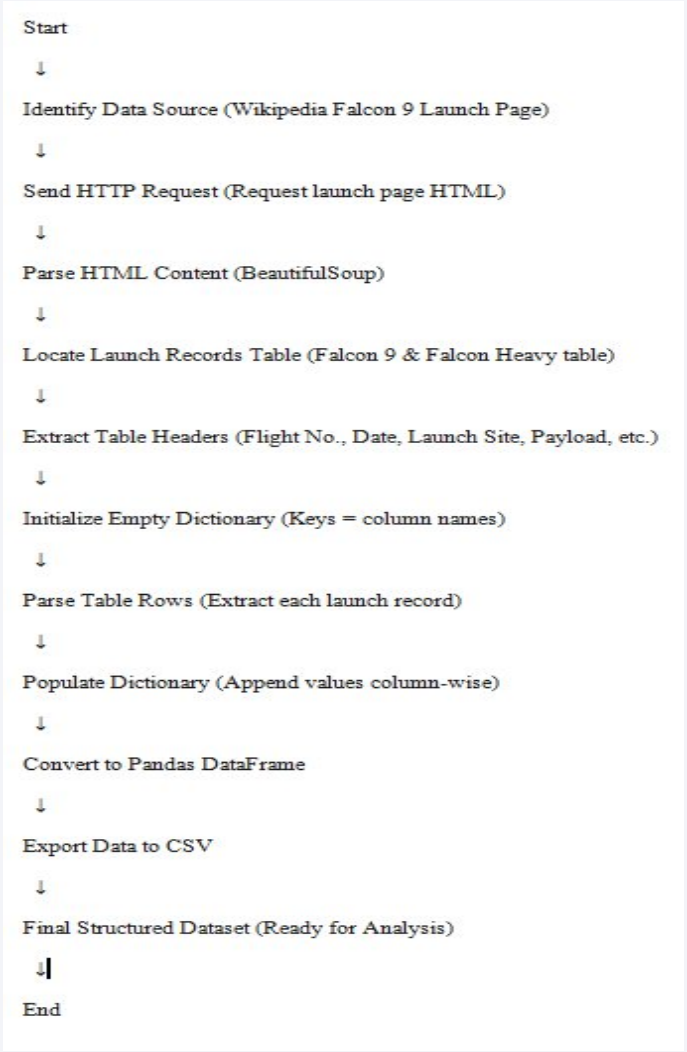
Data Collection - Scraping

Data collection – Webscraping process

- Wikipedia provides **structured historical launch data** in HTML table format.
- **BeautifulSoup** enables efficient extraction and parsing of HTML elements.
- Table headers are extracted first to ensure **schema consistency**.
- Parsing HTML tables into a **Pandas DataFrame** allows seamless analysis.
- **Column Names Extracted:** Flight No., Date and time (UTC), Launch site, Payload, Payload mass, Orbit, Customer, Launch outcome
- Web scraping complements API data by **filling data gaps and cross-validating records**.
- Exporting data to CSV ensures **reproducibility and independence of analysis steps**

GitHub Repository URL:

https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/jupyter-labs-webscraping.ipynb



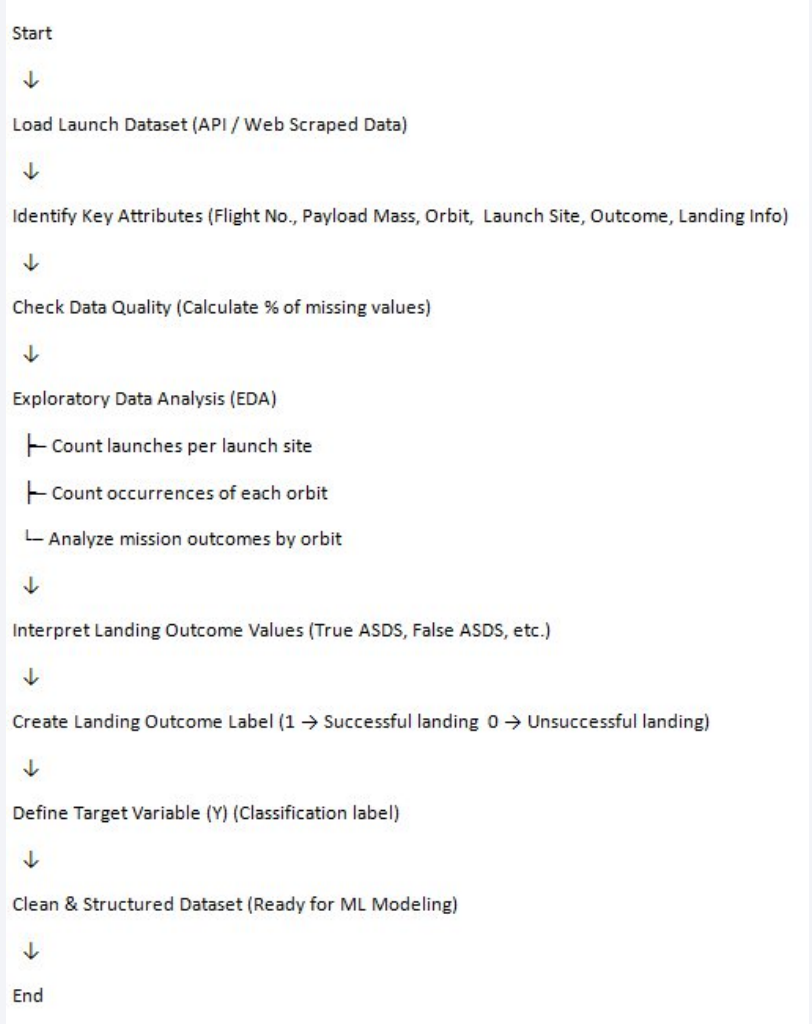
Data Wrangling

Data Wrangling Process

- Data wrangling transforms raw launch data into a **clean, structured, and ML-ready dataset**.
- Critical launch attributes such as **flight number, payload mass, orbit, launch site, and landing outcome** are identified and validated.
- Launch sites and orbit types provide **important categorical information** affecting landing success.
- Landing outcomes are converted into **binary class labels (0/1)** to support supervised machine learning.
- Missing value analysis ensures **data quality and reliability** before modeling.
- Exploratory calculations reveal **patterns across launch sites, orbits, and mission outcomes**.
- Creating a labeled target variable (**Y**) is a key step for training classification models.

GitHub Repository URL:

https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

Charts plotted and their usage

- **Scatter Plot (Flight Number vs. Payload Mass)**
  Purpose: To examine how launch experience (Flight Number) and payload mass influence landing success, showing improved outcomes with increasing flight numbers and tolerance to heavier payloads.
- **Scatter Plot (Flight Number vs. Launch Site)**
  Purpose: To analyze the relationship between launch experience and landing success across different launch sites, highlighting the correlation between higher flight numbers and successful landings.
- **Scatter Plot (Payload Mass vs. Launch Site)**
  Purpose: To assess whether payload mass impacts landing success at various launch sites and to observe changes in success probability with increasing payload weight.
- **Bar Chart (Success Rate by Orbit Type)**
  Purpose: To compare landing success rates across different orbit types and identify orbits with the highest success probabilities.
- **Scatter Plot (Flight Number vs. Orbit Type)**
  Purpose: To determine whether orbit type, along with launch experience, is a strong predictor of landing success.
- **Line Chart (Year vs. Average Success Rate)**
  Purpose: To visualize the yearly trend in launch success rates and evaluate improvements in mission reliability over time.

GitHub Repository URL:

https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/edadataviz.ipynb

EDA with SQL

- **Identification of Launch Sites**
  SQL was used to retrieve all *distinct launch sites* from the dataset, helping to understand the geographical distribution of SpaceX missions.
- **Filtering Launch Records by Site Name**
  Pattern matching (LIKE) was applied to extract launch records from sites beginning with *"CCA"*, enabling focused analysis on specific launch complexes.
- **Aggregate Payload Analysis for NASA Missions**
  The SUM function calculated the *total payload mass* delivered for *NASA (CRS)* missions, providing insight into mission scale and customer contribution.
- **Average Payload by Booster Version**
  The AVG function was used to determine the *mean payload mass* carried by the *F9 v1.1* booster, supporting performance comparison across booster versions.
- **Earliest Successful Ground Landing**
  The MIN function identified the *first date of a successful ground pad landing*, marking a key technological milestone.
- **Conditional Booster Performance Analysis**
  Queries with multiple conditions (AND, BETWEEN) were used to list *booster versions* that successfully landed on drone ships while carrying medium-range payloads (4000–6000 kg).
- **Mission Outcome Distribution**
  Grouping and counting (GROUP BY, COUNT) summarized the *number of successful and failed missions*, enabling outcome frequency analysis.
- **Maximum Payload Capability Identification**
  A *subquery with an aggregate function* identified booster versions that carried the *maximum payload mass*, highlighting peak performance missions.
- **Temporal and Conditional Failure Analysis**
  Date functions and filtering were applied to extract *drone ship landing failures in 2015*, along with associated booster versions and launch sites.
- **Ranking of Landing Outcomes Over Time**
  Aggregation, date filtering, and sorting (ORDER BY DESC) ranked landing outcomes between *2010 and 2017*, revealing dominant mission results over time.

GitHub Repository URL:

Notebook File save error in Coursera platform, hence uploading the screen captures

https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/2-Hands-on%20Lab%20Complete%20the%20EDA%20with%20SQL.docx

Build an Interactive Map with Folium

1. Folium Map Object: folium.Map

- To create an interactive geographic visualization centered at NASA Johnson Space Center, Houston, Texas.
- Provided a spatial reference frame for visualizing all SpaceX launch sites and related geographic features.

2. Launch Site Location Markers: folium.Circle, folium.Marker

- folium.Circle was used to highlight the physical area of each launch site and improve visual visibility at different zoom levels. Allowed intuitive visualization of launch site locations.
- folium.Marker with text labels was added to clearly identify each launch site by name. Revealed that all launch sites are located near coastlines and not close to the equator.

3. Color-Coded Launch Outcome Markers: folium.Marker with color encoding (green = success, red = failure), MarkerCluster

- Color-coded markers helped visually distinguish between successful and failed launches. Made it easy to identify launch sites with higher success rates.
- MarkerCluster reduced map clutter by grouping multiple launch records at the same geographic coordinates. Enabled quick visual comparison of performance across different launch locations.

4. Mouse Position Coordinate Tracker: MousePosition

- To display real-time latitude and longitude values when hovering over the map.
- Assisted in identifying exact coordinates of nearby geographic features such as coastlines, railways, highways, and cities. Enabled accurate distance measurements between launch sites and nearby infrastructures.

5. Proximity Markers: folium.Marker (for coastline, city, railway, highway)

- To mark the nearest points of interest around each launch site.
- Provided reference points for distance calculations. Helped analyze how launch sites are spatially separated from cities and critical infrastructure.

6. Distance Lines Between Launch Sites and Nearby Features: folium.PolyLine

- To visually represent the distance between launch sites and nearby geographic features (coastline, highways, cities, railways).
- Demonstrated that launch sites are: Very close to coastlines, Relatively close to highways, Not very close to railways, Strategically far from cities for public safety

Overall Findings from Map Visualization:

1. Launch sites are strategically located near coastlines to allow safe disposal of rocket stages.

2. Adequate distance from cities, railways, and highways minimizes risk to human life and infrastructure.

3. Interactive mapping significantly improves spatial understanding compared to tabular latitude–longitude data.

GitHub Repository URL:
https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

1. **Launch Site Drop-down Input Component**

- Select **All Sites** or Choose a **specific launch site** (e.g., CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E)
- The dropdown enables **site-wise comparison.** It allows users to **focus analysis on one launch site** instead of viewing aggregated data

2. **Payload Range Slider**: A **range slider** allowing users to select payload mass ranges from **0 to 10,000 kg**.

- Payload mass is a **critical factor** influencing landing success
- The slider allows exploration of: Light vs heavy payload missions, Success patterns across payload ranges

3. Success Pie Chart:

- A **pie chart** that updates dynamically based on the selected launch site. *Which site has the most successful launches? Which site has the highest success rate?*
- **All Sites selected** → shows total successful launches across all sites
- **Specific Site selected** → shows success vs failure counts for that site

4. Success-Payload Scatter Plot:

- A **scatter plot** that updates based on: **X-axis:** Payload Mass (kg), **Y-axis:** Launch outcome (Class), **Color:** Booster Version Category.
- Scatter plots reveal **relationships and patterns** between payload mass and mission outcome
- Selected launch site (dropdown)
- Selected payload range (slider)

GitHub Repository URL

https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

Model Development & Evaluation Summary

1. Data standardization:

The model development process began with **data standardization** to ensure all features were on a uniform scale

2. Train–test split

Split train and test data: **train–test splitting** to enable unbiased performance evaluation.

3. Multiple model training

Four supervised classification models—**Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree**—were trained as candidate models.

4. Baseline performance evaluation

Each model was first evaluated using baseline accuracy

5. Hyperparameter tuning (GridSearchCV)

Next **hyperparameter tuning with GridSearchCV** was applied to improve performance and reduce overfitting.

6. Model comparison

Post-tuning results showed performance gains across all models. A **comparative analysis** revealed that the **Decision Tree classifier achieved the highest accuracy (87.68%)**, outperforming other models.

7. Best model selection

The Decision Tree performed best because it effectively captured **non-linear relationships and feature interactions** among payload mass, orbit type, launch site, and operational experience. Based on accuracy, interpretability, and robustness, it was selected as the **best-performing classification model** for predicting Falcon 9 landing success.

GitHub Repository URL

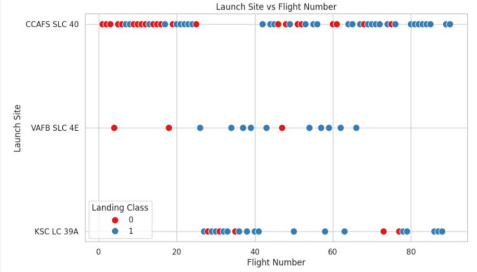
https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results (EDA Results)

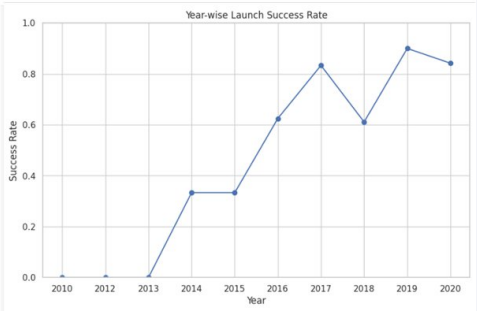
Exploratory data analysis results:

Exploratory Data Analysis was conducted to understand the structure, distribution, and relationships within the SpaceX launch dataset. Key attributes such as **flight number, payload mass, orbit type, launch site, and landing outcome** were analyzed using statistical summaries and visualizations. Missing values were identified and appropriately handled to ensure data quality and reliability for downstream modeling and prepared data for Feature Engineering

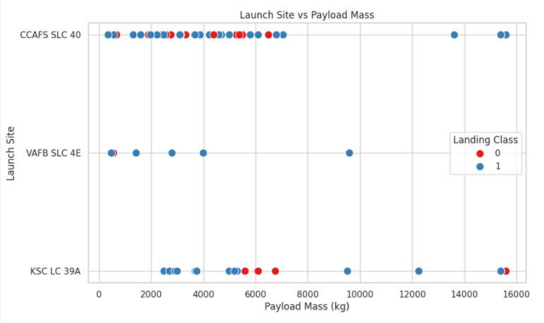
1. An increase in landing success rates as flight numbers increased,

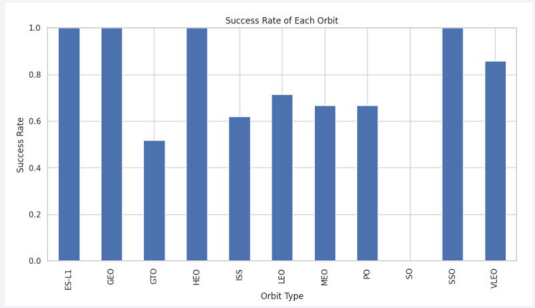


2. Indicating learning and operational maturity over time.

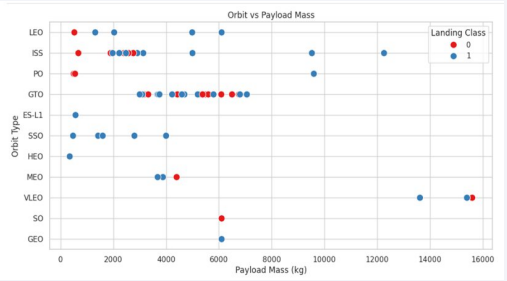


3. Payload mass and orbit type were also found to influence landing outcomes, with certain orbits(**LEO, ISS, GTO**) exhibiting higher success probabilities





4. Heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS

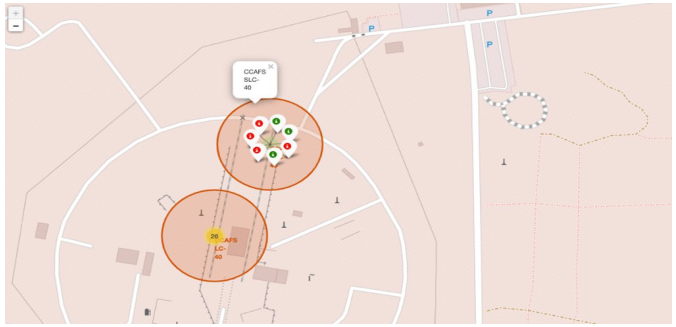


Results (Map Visualization Results)

Findings from Map Visualization

- Launch sites are strategically located near coastlines to allow safe disposal of rocket stages.
- Adequate distance from cities, railways, and highways minimizes risk to human life and infrastructure.

Marked the success/failed launches for each site on the map, Calculated the distances between a launch site to its proximities



Findings from Interactive Dashboard with Plotly

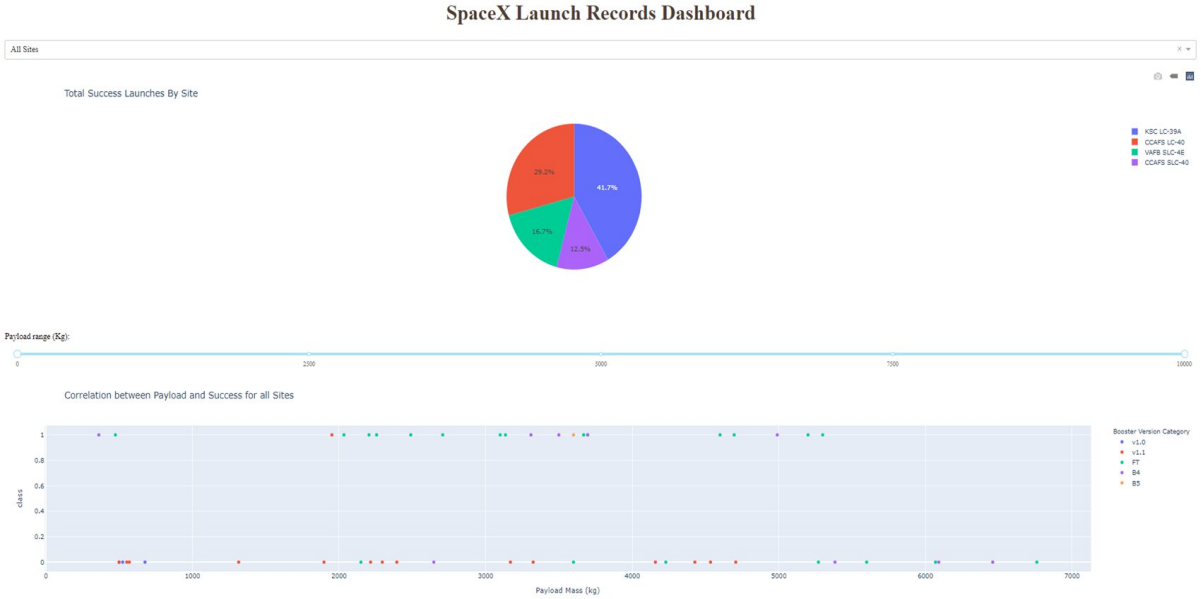
1. **KSC LC-39 A** launch site has the largest successful launches say 41.7%

2. **KSC LC-39 A** – has the highest launch success rate of 76.9%

3. Payload range(s) between **2000 to 5000** has the highest launch success rate

4. Payload range(s) between **0 to 1000** has the lowest launch success rate

5. **Block 5 (B5)** - Nearly all B5 launches appear successful. Performs well even at **higher payload masses**, indicating superior reliability. **Falcon 9 Full Thrust (FT) -** Also shows a high success rate, but with slightly more failures than B5. **v1.0 and v1.1 -** Display noticeably higher failure rates, particularly at lower payloads.



Results (Predictive Analysis Results)

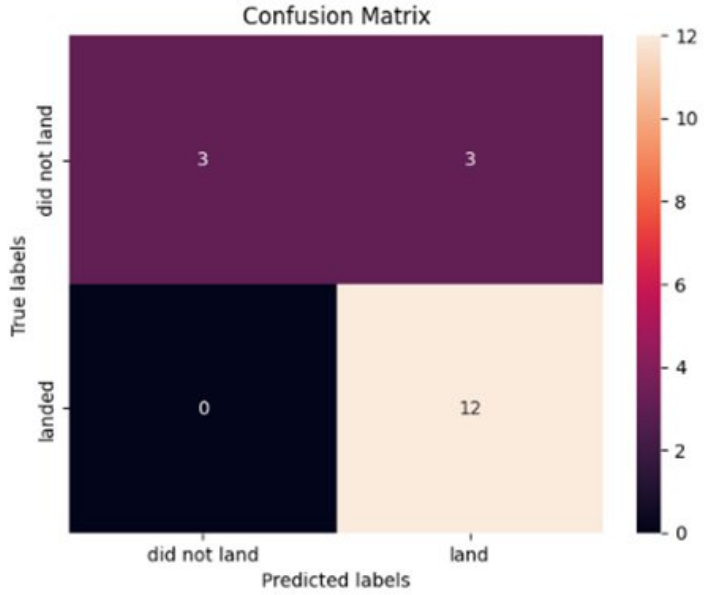
Predictive analysis results

Comparison before hyperparameter tuning

Model	Test Accuracy
Logistic Regression	0.8333
SVM	0.8333
Decision Tree	0.8333
KNN	0.8333

Comparison after hyperparameter tuning (GridSearchCV)

Model	Tuned Test Accura
Logistic Regression	0.8464
SVM	0.8482
Decision Tree	**0.8768**
KNN	0.8482



Best performing method

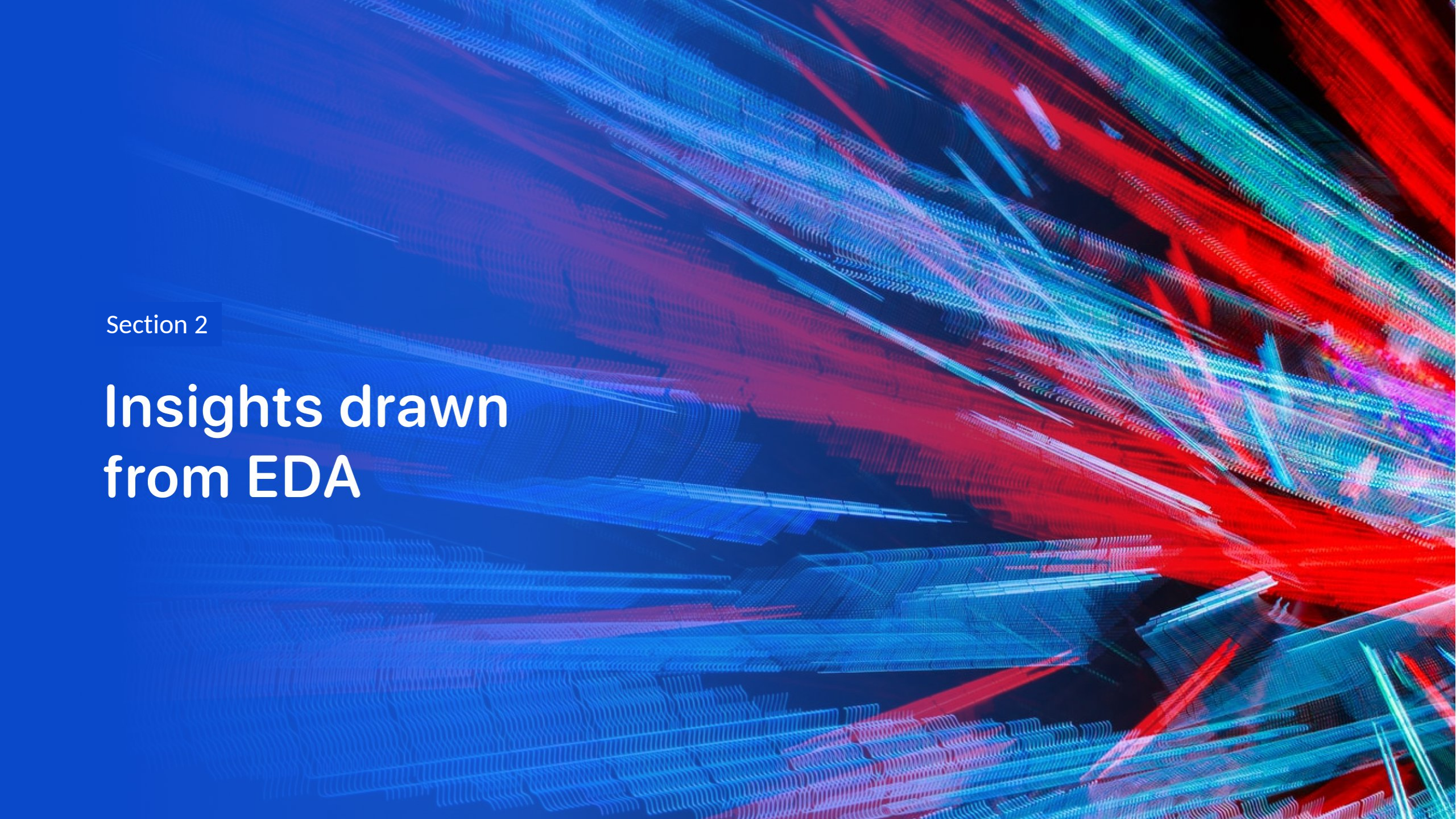
Highest tuned test accuracy**: 87.68%**
Outperforms: Logistic Regression, SVM, KNN
Best Model**: Decision Tree Classifier**

Conclusion

Decision Trees:
- Capture **non-linear relationships** effectively
- Do not assume linear boundaries (unlike Logistic Regression)
- Work well with **feature interactions**

Hyperparameter tuning (depth, splits, leaf size) helped:
- Reduce overfitting
- Improve generalization

The dataset structure favors **rule-based splits**, which suits Decision Trees


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

Improvement over time

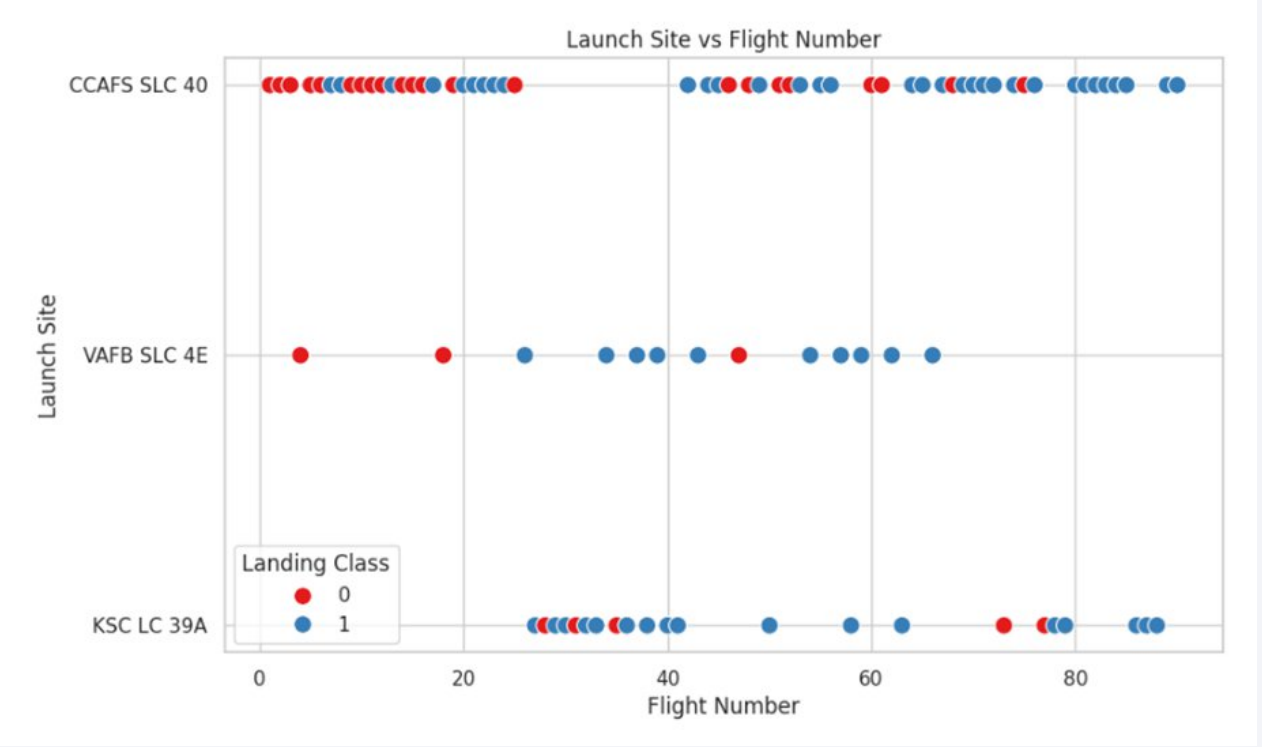
- Early flight numbers (left side) have **more red dots** → more failures.
- As flight numbers increase, **blue dots dominate** → landing success improves.
- This clearly shows **SpaceX's technological learning and reliability growth** over time.

No clear failures in later stages

- After ~Flight Number 60:
  - Failures become **rare**
  - Success is consistent across **all sites**
- This suggests **process stabilization and booster reusability success**.

Overall conclusion

- Landing success strongly correlates with flight number
- Later missions have significantly higher success rates
- All launch sites show improvement over time
- KSC LC-39A has the highest success density



Payload vs. Launch Site

Payload mass strongly affects landing success

- **Lower payload masses (0–4000 kg) : High landing success rate**
- **Medium payload masses (4000–8000 kg) :** Mixed outcomes
- **Very heavy payloads (>10,000 kg) :** Fewer data points, **Higher failure risk**, especially early missions

Insight: Heavier payloads reduce available fuel for landing, increasing risk.

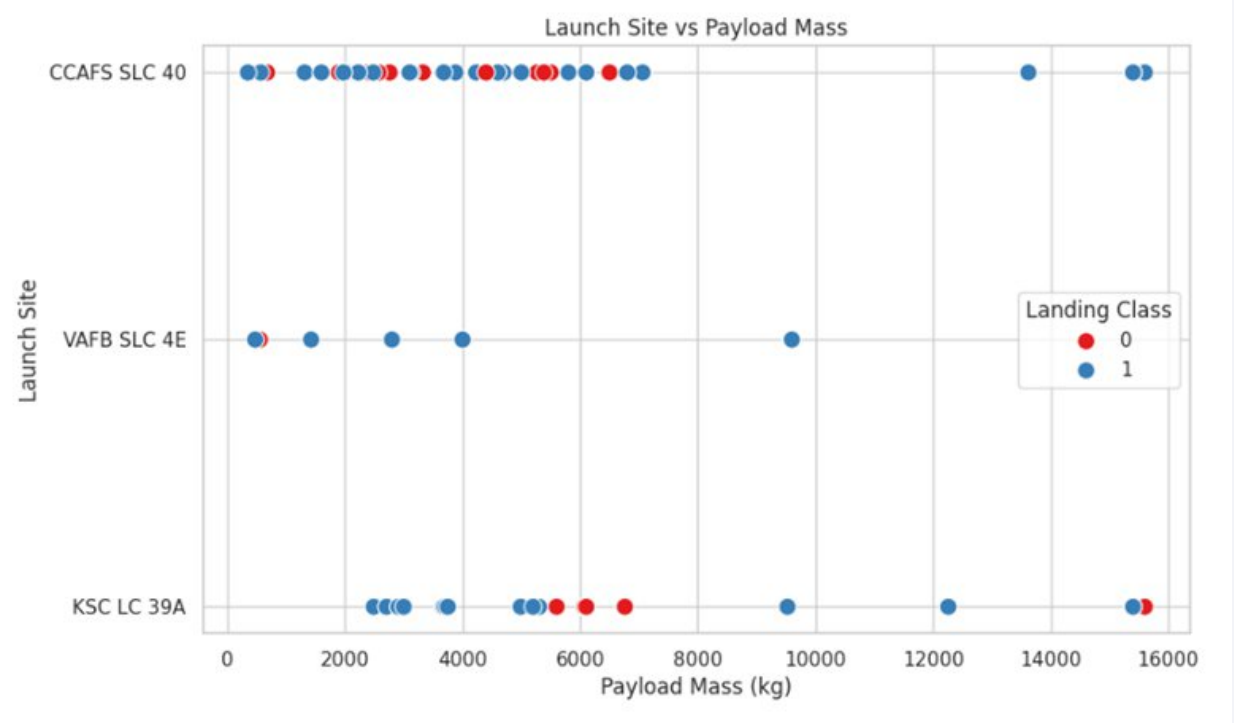
Success becomes possible even for heavy payloads

- Blue dots appear even at **>12,000 kg**
- Shows **technological advancement** (booster reusability, better landing control)

Payload mass is important, but **not the only factor**.

Overall conclusions

- Landing success probability decreases as payload mass increases
- Lower payload missions are significantly safer
- CCAFS SLC 40 is the most flexible launch site
- KSC LC-39A supports heavier missions but with higher risk
- Improvements over time enable success even at high payload masses



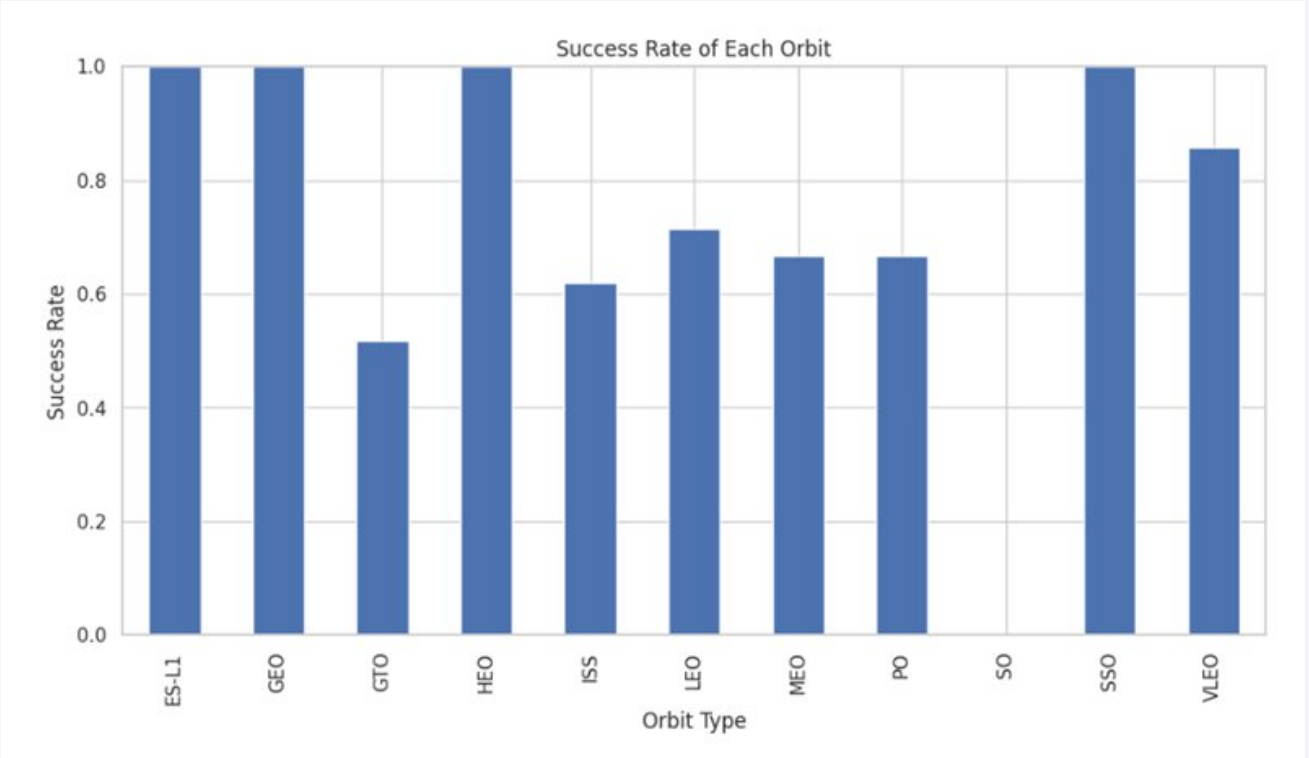
Success Rate vs. Orbit Type

Orbits with highest success rates:

1. Higher Success Rate orbits:

- ES-L1
- GEO
- HEO
- SSO

2. GTO has lower success rate



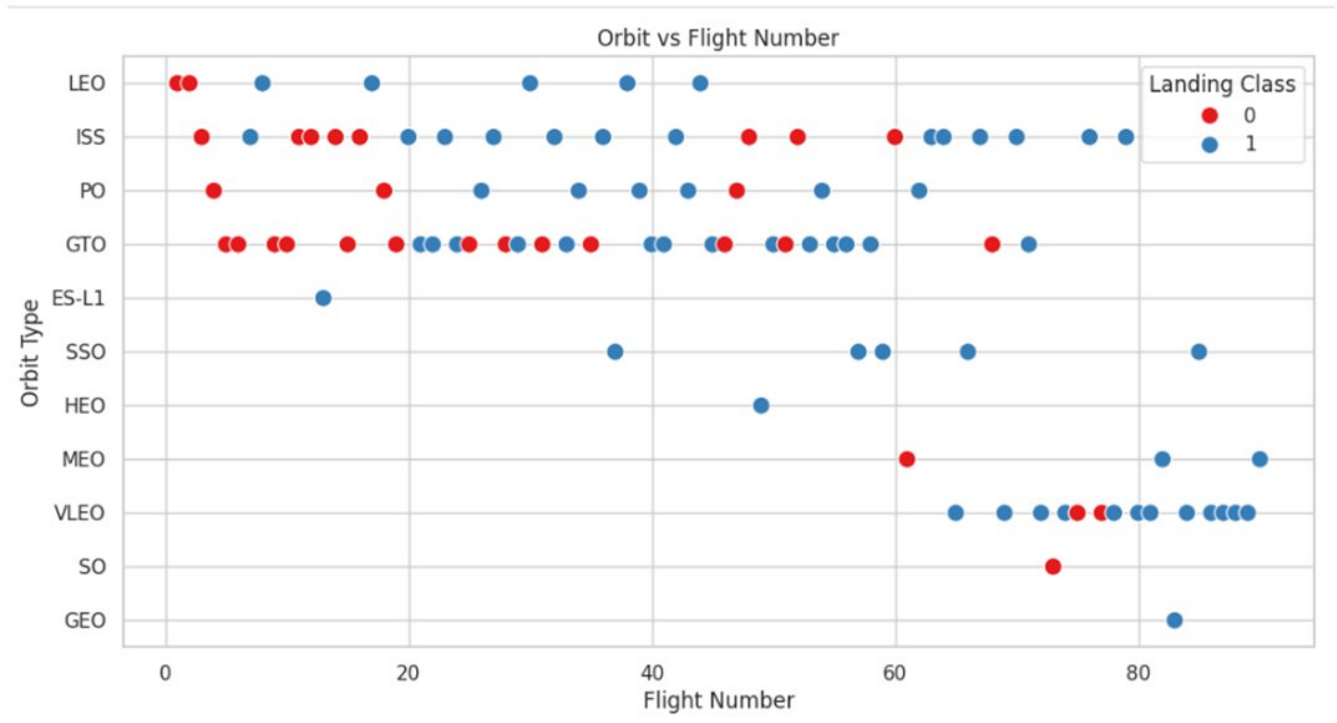
Flight Number vs. Orbit Type

Results:

- **Early flight numbers** → more failures across orbits
- **Later flight numbers** → more successes
- Orbits like **LEO, ISS, GTO** show **higher success density**
- Rare orbits show **scattered and inconsistent outcomes**

Key insights:

- Flight experience matters across **all orbit types**
- Orbit is a strong predictor of landing success
- Certain orbits are inherently safer and better optimized



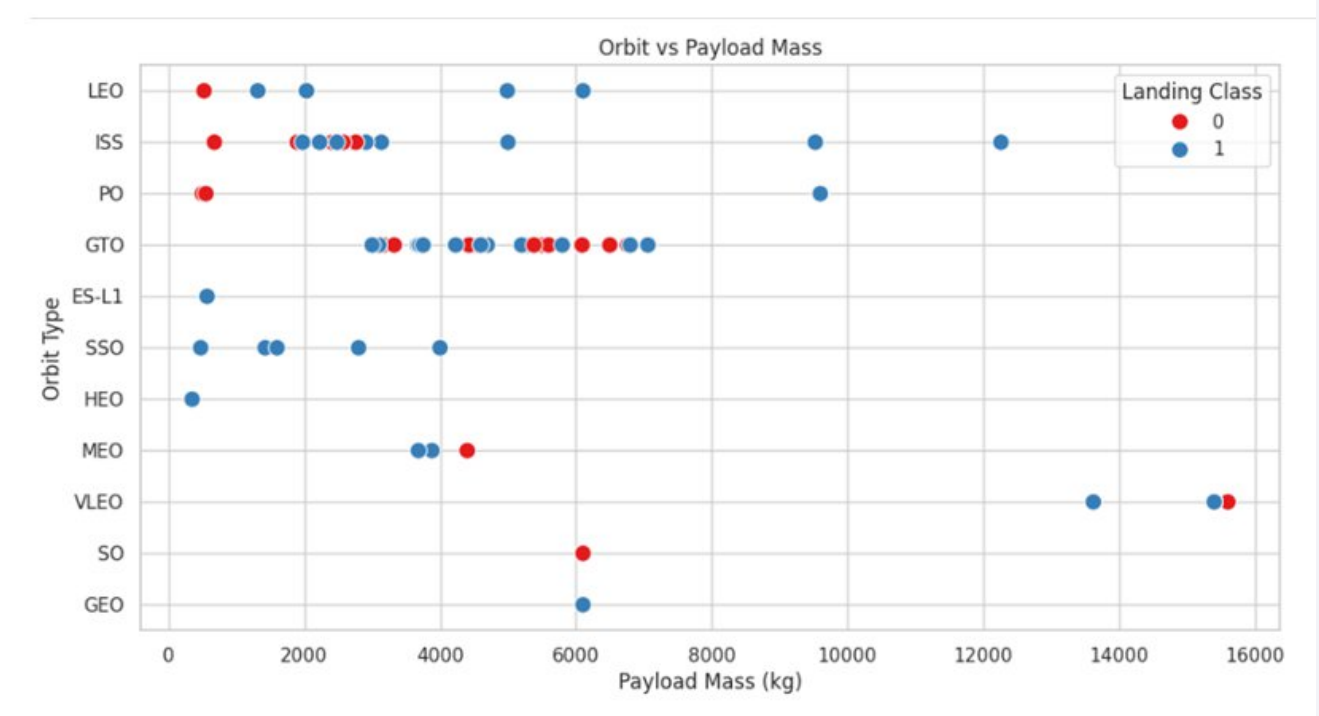
Payload vs. Orbit Type

Results

- **Lower payload masses** → mostly successes across orbits
- **Heavier payloads** → more failures, especially in **GTO**
- **LEO / ISS** orbits show **higher success rates**
- Rare orbits have fewer data points → higher uncertainty

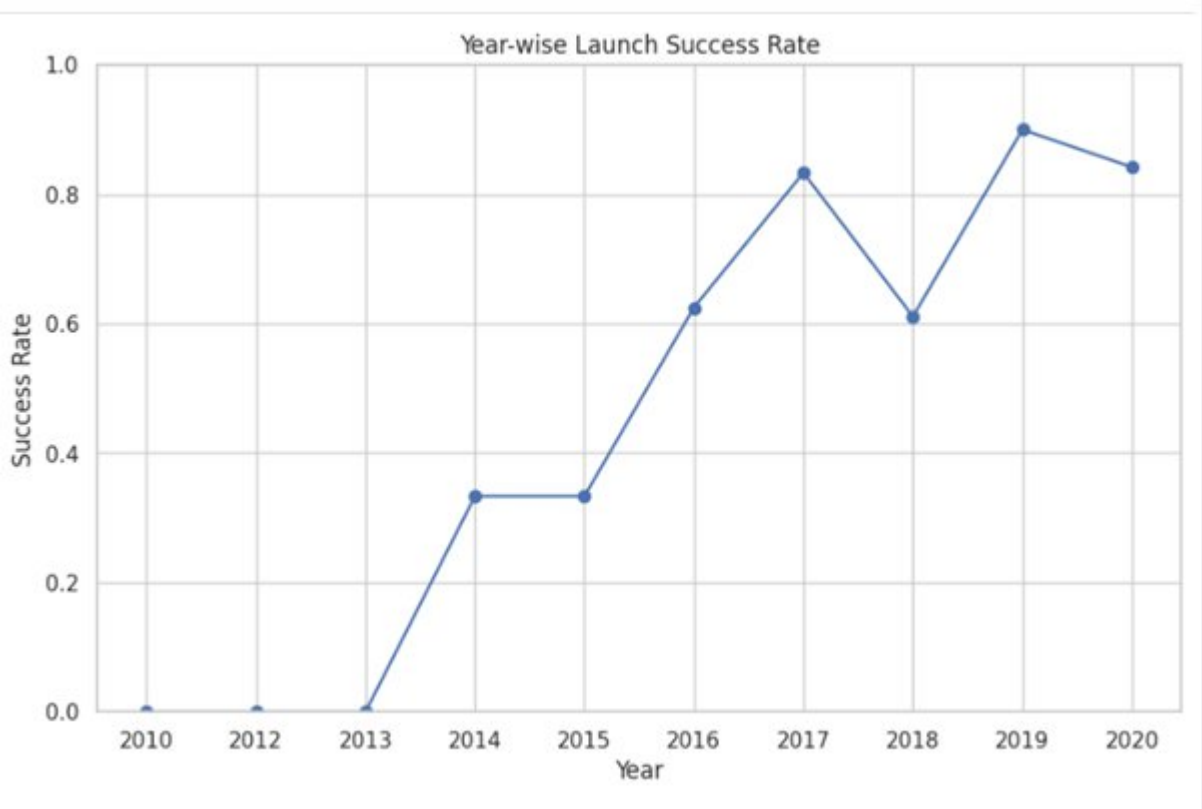
Key insights

- Payload mass impacts landing success differently per orbit
- Orbit type and payload mass interact (non-linear relationship)
- PayloadMass + Orbit are strong ML features



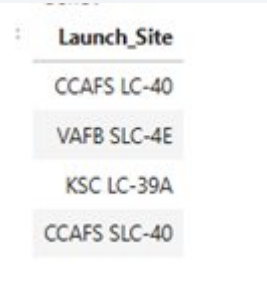
Launch Success Yearly Trend

- Shows the average launch success trend
- Success rate since 2013 kept increasing till 2020



All Launch Site Names

%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;



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

Identification of Launch Sites

SQL was used to retrieve all *distinct launch sites* from the dataset, helping to understand the geographical distribution of SpaceX missions.

Launch Site Names Begin with 'CCA'

%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

Filtering Launch Records by Site Name

Pattern matching (LIKE) was applied to extract launch records from sites beginning with “*CCA*”, enabling focused analysis on specific launch complexes

Total Payload Mass

%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Customer = 'NASA (CRS)'

SUM(PAYLOAD_MASS__KG_)
45596

Aggregate Payload Analysis for NASA Missions

The SUM function calculated the *total payload mass* delivered for *NASA (CRS)* missions, providing insight into mission scale and customer contribution

Average Payload Mass by F9 v1.1

%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Booster_Version = 'F9 v1.1'

AVG(PAYLOAD_MASS__KG_)
2928.4

Average Payload by Booster Version
The AVG function was used to determine the *mean payload mass* carried by the *F9 v1.1* booster, supporting performance comparison across booster versions.

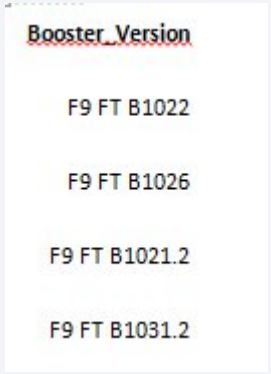
First Successful Ground Landing Date

%sql SELECT min(Date) FROM SPACEXTABLE where Landing_Outcome = 'Success (ground pad)‘

Earliest Successful Ground Landing
The MIN function identified the *first date of a successful ground pad landing*, marking a key technological milestone

Successful Drone Ship Landing with Payload between 4000 and 6000

%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

Conditional Booster Performance Analysis

Queries with multiple conditions (AND, BETWEEN) were used to list *booster versions* that successfully landed on drone ships while carrying medium-range payloads (4000–6000 kg).

Total Number of Successful and Failure Mission Outcomes

%sql SELECT trim(Mission_Outcome), Count(1) from SPACEXTABLE group by trim(Mission_Outcome)

trim(Mission_Outcome)	Count(1)
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Mission Outcome Distribution

Grouping and counting (GROUP BY, COUNT) summarized the *number of successful and failed missions*, enabling outcome frequency analysis.

Boosters Carried Maximum Payload

%sql SELECT Booster_Version FROM SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

Maximum Payload Capability Identification

A *subquery with an aggregate function* identified booster versions that carried the *maximum payload mass*, highlighting peak performance missions.

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

%sql 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)‘

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

Temporal and Conditional Failure Analysis

Date functions and filtering were applied to extract *drone ship landing failures in 2015*, along with associated booster versions and launch sites.

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

%sql SELECT Landing_Outcome, count(1) as Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' group by Landing_Outcome order by Outcome_Count desc

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

Ranking of Landing Outcomes Over Time

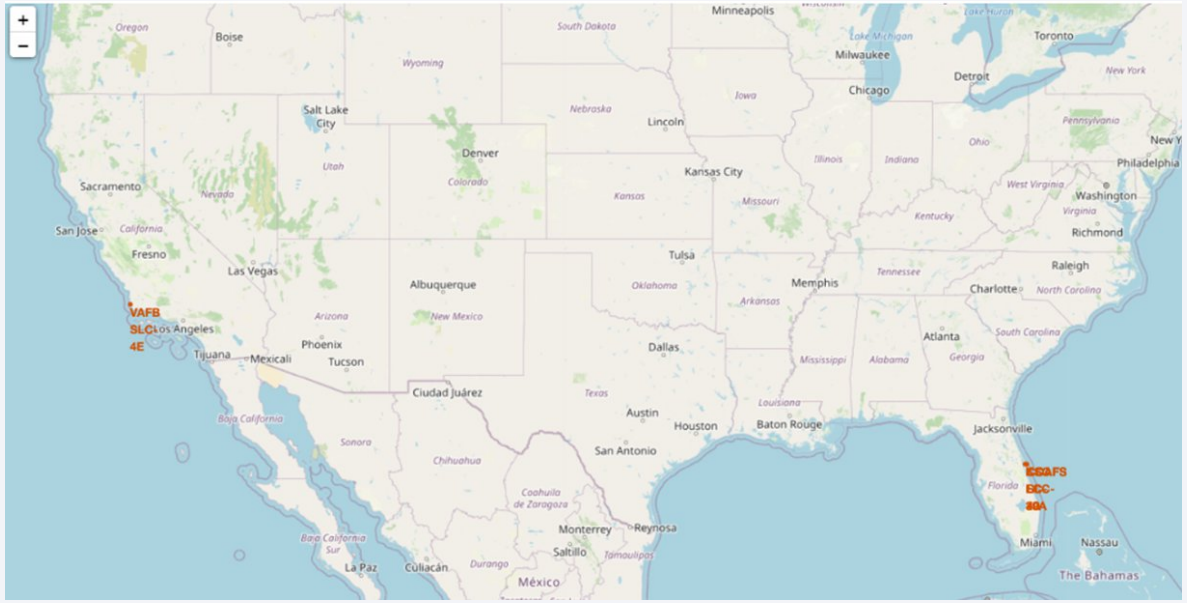
Aggregation, date filtering, and sorting (ORDER BY DESC) ranked landing outcomes between *2010 and 2017*, revealing dominant mission results over time.

Section 3

Launch Sites Proximities Analysis

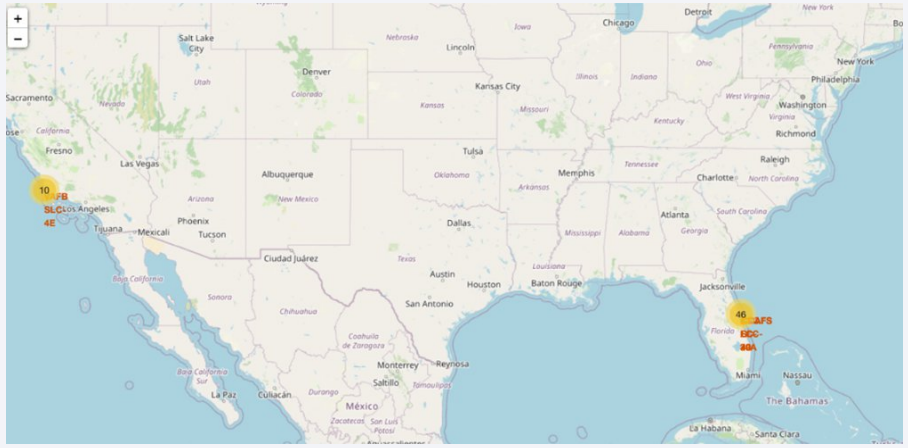
Folium Map – All launch sites location

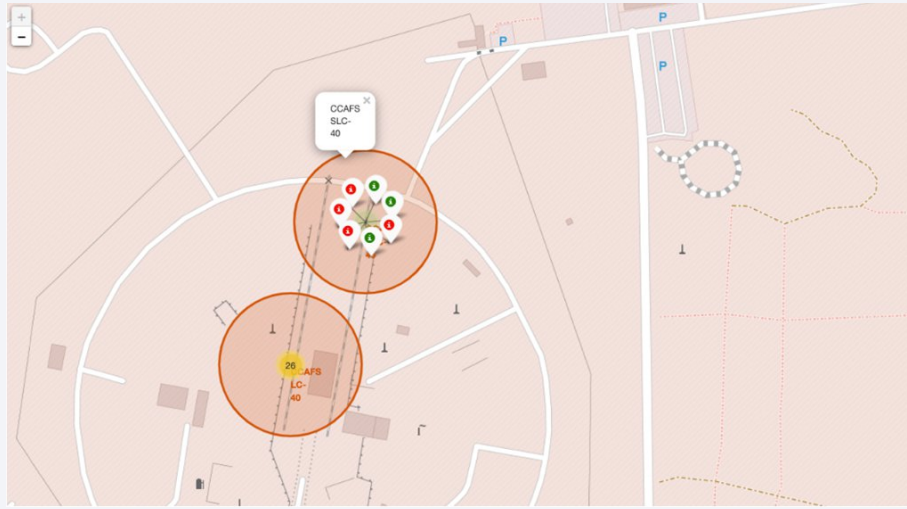
- Map highlights the physical area of each launch site and improve visual visibility at different zoom levels.
- Allowed intuitive visualization of launch site locations.
- Text labels were added to clearly identify each launch site by name. Here four launch site locations are marked in the map with appropriate text labels
- Revealed that all launch sites are located near coastlines and not close to the equator.



Folium Map – Launch outcomes

- Success/failed launches for each site on the map
- Color-coded markers helped visually distinguish between successful and failed launches. Made it easy to identify launch sites with higher success rates. From the screenshot, CCAFS SLC-40 has 3 successful and 4 failed launches
- MarkerCluster reduced map clutter by grouping multiple launch records at the same geographic coordinates.
- Enabled quick visual comparison of performance across different launch locations.

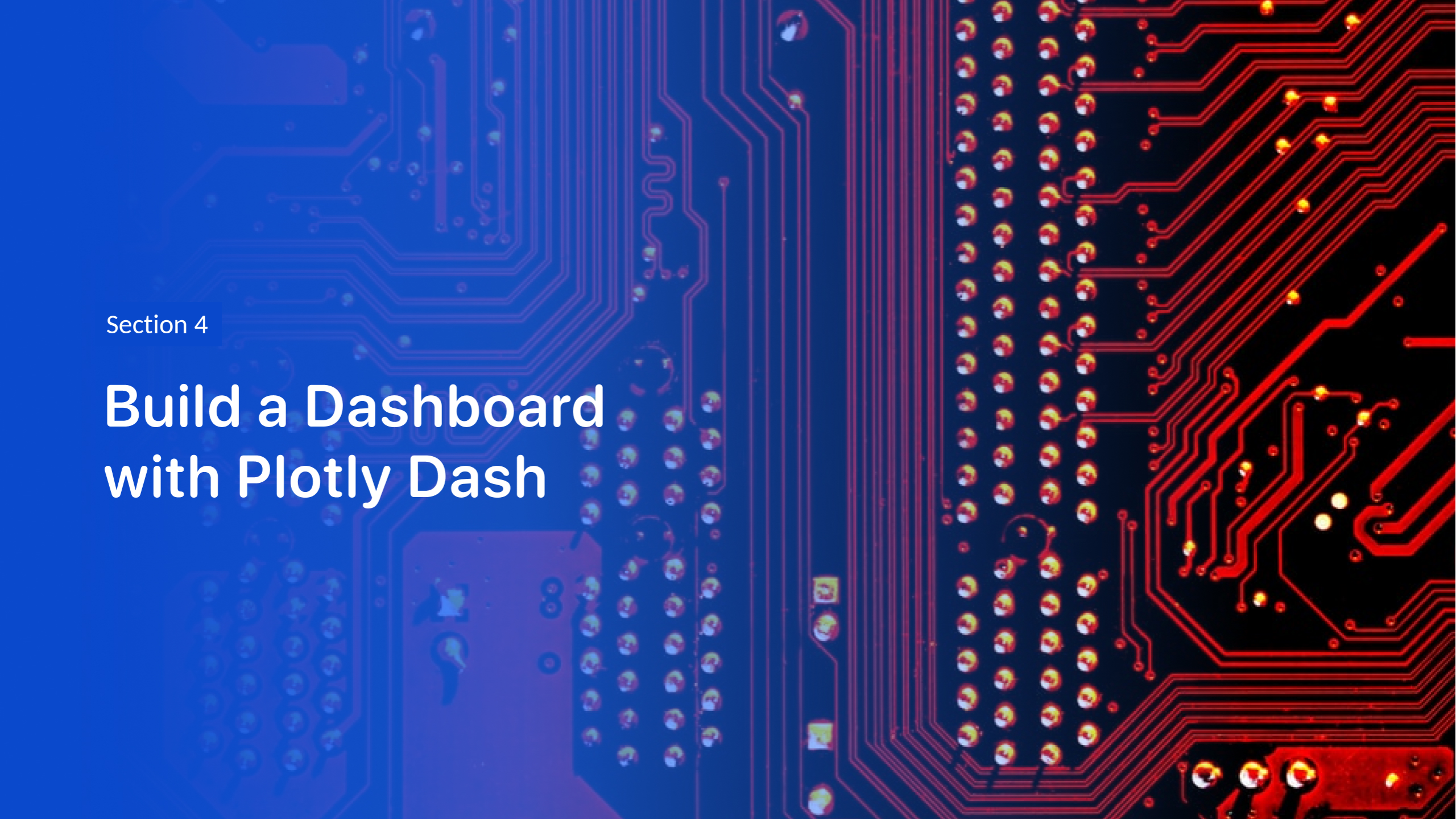




Folium Map – Distance between launch site to its proximities

- Displays real-time latitude and longitude values when hovering over the map.
- Assisted in identifying exact coordinates of nearby geographic features such as coastlines, railways, highways, and cities. Enabled accurate distance measurements between launch sites and nearby infrastructures.
- Helped analyze how launch sites are spatially separated from cities and critical infrastructure.
- To visually represent the distance between launch sites and nearby geographic features (coastline, highways, cities, railways).

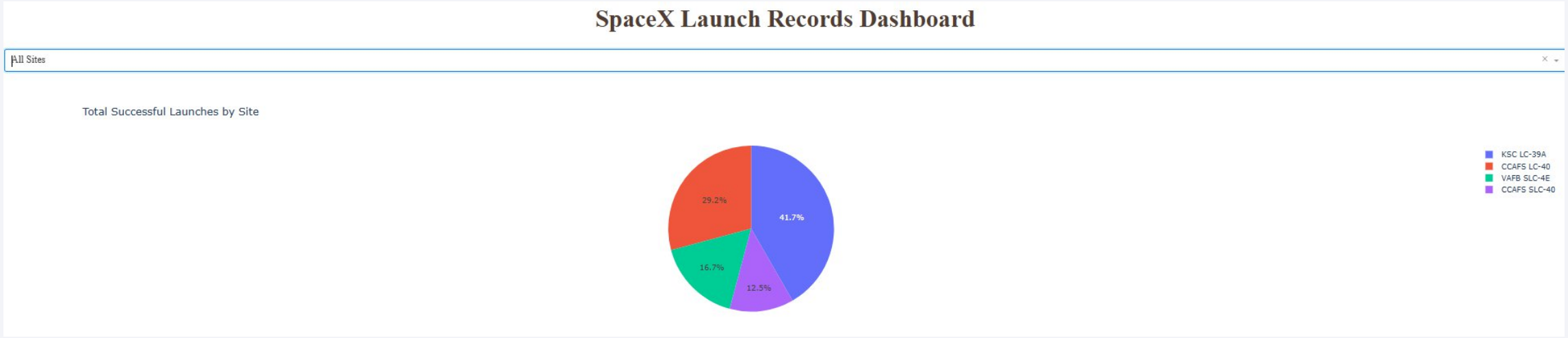
Findings: Very close to coastlines, Relatively close to highways, Not very close to railways, Strategically far from cities for public safety


Section 4
Build a Dashboard with Plotly Dash

Dashboard –Highest successful launches

Pie chart updates dynamically based on the selected launch site to identify which site has the most successful launches

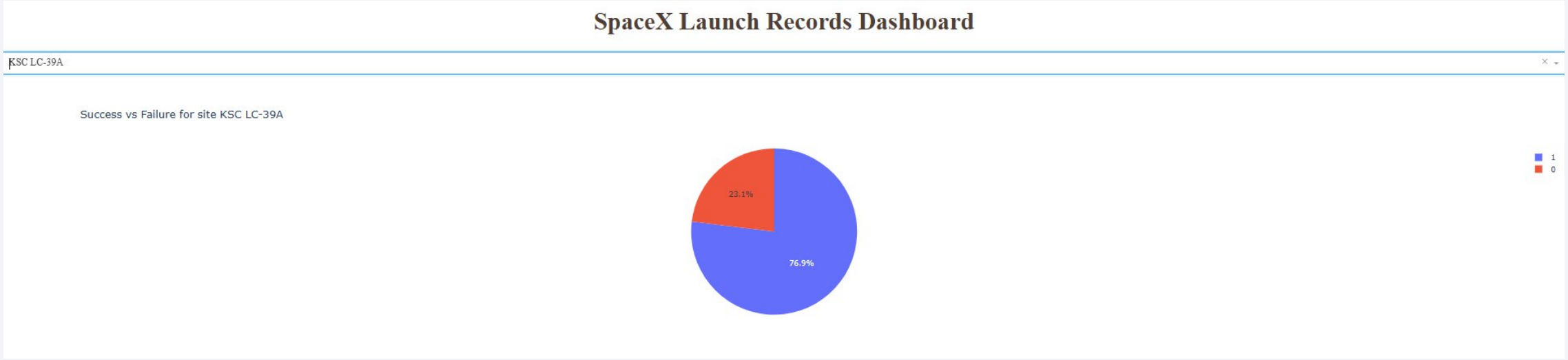
KSC LC-39 A – it has largest successfully launches say 41.7%



Dashboard –Highest launch success ratio

Pie chart updates dynamically based on the selected launch site to identify which site has the highest success ratio.

KSC LC-39 A – it has highest success ratio of 76.9%



Dashboard – Payload vs. Launch outcome

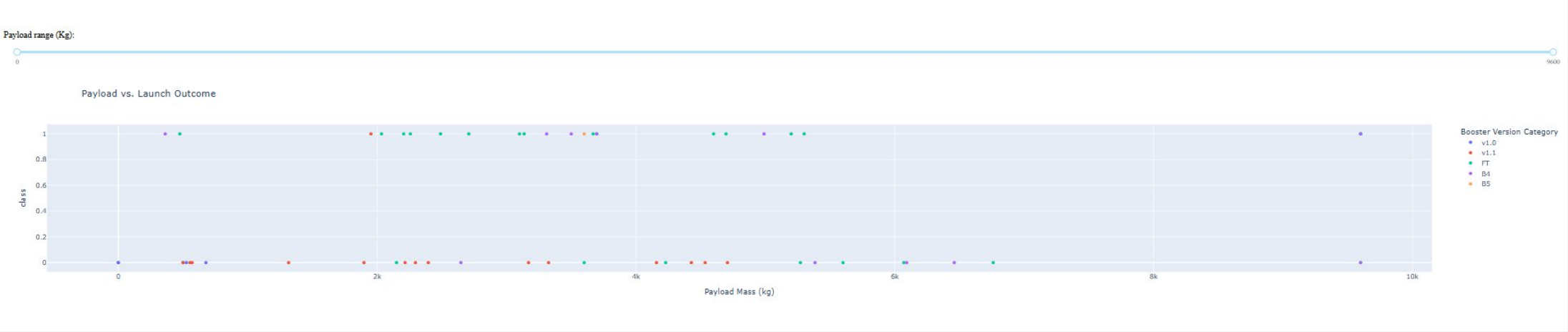
1. Payload range(s) has the highest launch success rate: **Between 2000 to 5000.**

2. Payload range(s) has the lowest launch success rate: **Between 0 to 2000.**

- **Low payloads (0–1,000 kg)** shows **more failed launches.** Many of these launches correspond to **early missions** and older booster versions, largely due to early-stage operational learning
- **Mid-range failures (~1,500–3,000 kg):** A noticeable cluster of failures appears here, especially for earlier booster designs

3. **Booster version with the highest launch success rate - Falcon 9 Block 5 (B5)** has the **highest launch success rate**, reflecting design maturity and operational optimization

- **Block 5 (B5)** - Nearly all B5 launches appear at **class = 1 (successful)**. Performs well even at **higher payload masses**, indicating superior reliability.
- **Falcon 9 Full Thrust (FT) -** Also shows a high success rate, but with slightly more failures than B5.
- **v1.0 and v1.1 -** Display noticeably higher failure rates, particularly at lower payloads.

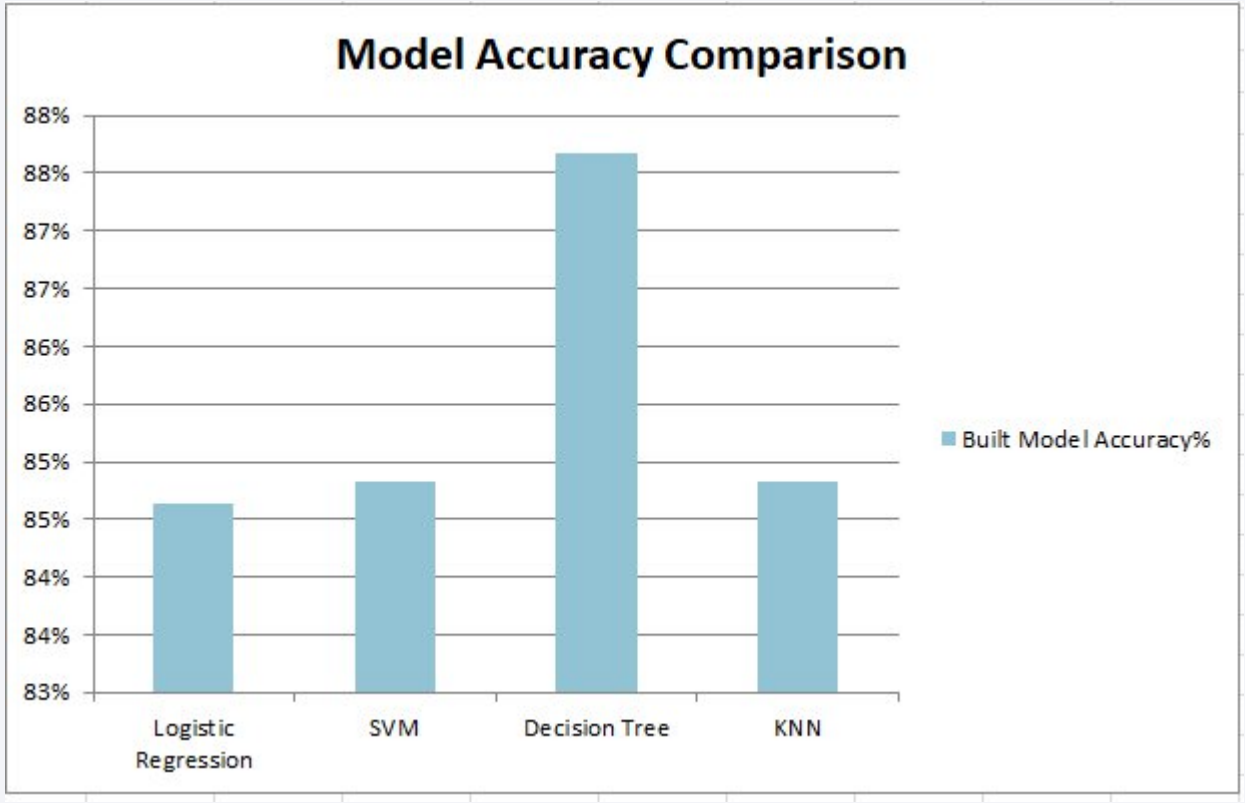


Section 5

Predictive Analysis (Classification)

Classification Accuracy

Decision Tree classification has the highest model accuracy of 88%



Confusion Matrix

Confusion matrix values:

	Predicted: Did Not Land	Predicted: Land
Actual: Did Not Land	3 (True Negatives)	3 (False Positives)
Actual: Landed	0 (False Negatives)	12 (True Positives)

Key Observations

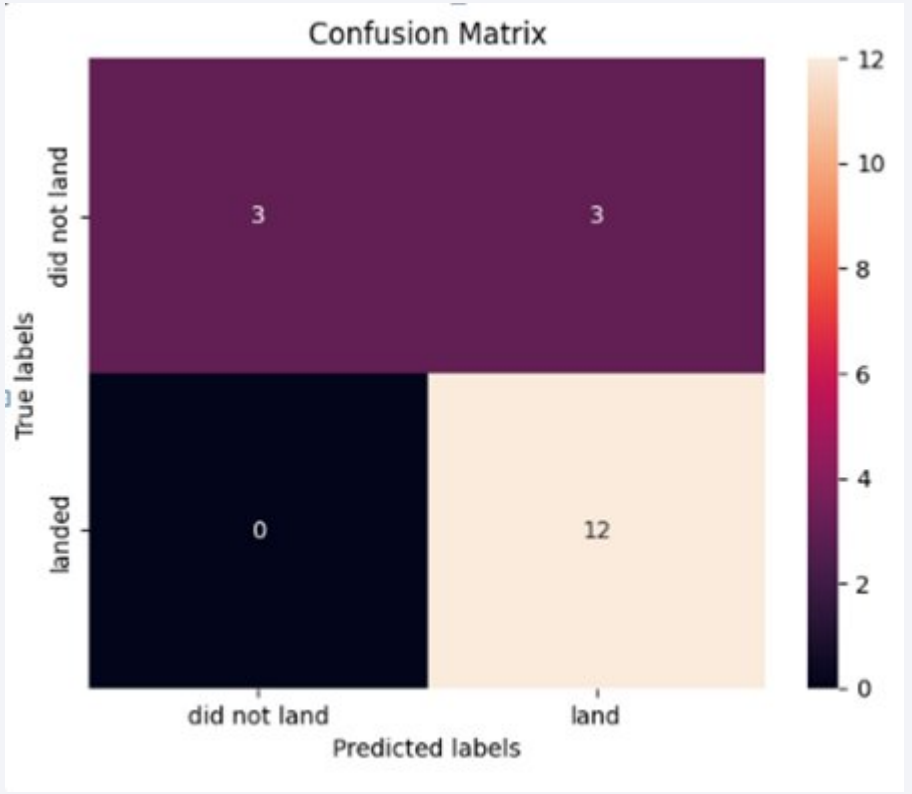
True Positives (12): The model correctly predicted 12 successful landings.

True Negatives (3): The model correctly identified 3 failed landings.

False Positives (3): 3 failed landings were incorrectly predicted as successful.

False Negatives (0): No successful landings were misclassified as failures.

Confusion matrix of Decision Tree Model:



Conclusions

- Successfully collected and integrated SpaceX launch data from **REST APIs and web-scraped sources**, ensuring a comprehensive and reliable dataset.
- Performed **effective data wrangling and exploratory data analysis**, uncovering key relationships between payload mass, orbit type, launch site, and landing success.
- Visual analytics revealed that **launch success increases with operational experience**, as indicated by higher success rates at larger flight numbers.
- Geographic analysis using **interactive Folium maps** showed that launch sites are strategically located **near coastlines and away from populated areas**, prioritizing safety and recovery efficiency.
- Multiple machine learning classification models were developed and evaluated, including **Logistic Regression, SVM, KNN, and Decision Tree**.
- **Hyperparameter tuning significantly improved model performance**, demonstrating the importance of model optimization.
- The **Decision Tree classifier emerged as the best-performing model**, achieving the highest accuracy and capturing complex non-linear relationships effectively.
- Confusion matrix analysis confirmed **perfect recall for successful landings**, making the model reliable for predicting landing success.
- Newer booster versions, particularly **Falcon 9 Block 5**, consistently showed **higher launch success rates**, highlighting technological maturity.
- Overall, the project demonstrates how **data science and machine learning can effectively predict rocket landing outcomes**, offering valuable insights for mission planning and aerospace operations.

Appendix

Data Sources:

SpaceX launch data: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json

SpaceX Rocket launch history API URL: https://api.spacexdata.com/v4/launches/past

Wiki page for Falcon 9 launches: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Reference Links:

Hands-on Lab : String Patterns, Sorting and Grouping

Hands-on Lab: Built-in functions

Hands-on Lab : Sub-queries and Nested SELECT Statements

Hands-on Tutorial: Accessing Databases with SQL magic

Hands-on Lab: Analyzing a real World Data Set

GitHub Repository URLs:

Data Collection: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Wrangling: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/edadataviz.ipynb

EDA with SQL: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/2-Hands-on%20Lab%20Complete%20the%20EDA%20with%20SQL.docx

Interactive Map with Folium: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/lab_jupyter_launch_site_location.ipynb

Dashboard with Plotly Dash: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/spacex-dash-app.py

Predictive Analysis: https://github.com/subhashinig81-beep/Coursera-Data-Science/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Thank you!