IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Igor Andrade
June 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis result

Introduction

- Project background and context

  SpaceX revolutionized commercial space travel by significantly reducing launch costs, mainly through reusing Falcon 9's first stage. While traditional providers charge over $165M per launch, SpaceX offers the same for just $62M. Predicting whether the first stage will successfully land directly impacts cost estimation.

- Problems you want to find answers
  - Analyze how payload mass, launch site, flight count, and orbit type influence landing success.
  - Examine trends in landing success over time.
  - Identify the most effective machine learning algorithm for predicting first stage landing outcomes.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Spacex API
  - Web Scrapping
- Perform data wrangling
  - Clear data with Pandas
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Avaliate Logistic Regression, SVM, Decision Tree and K-Means

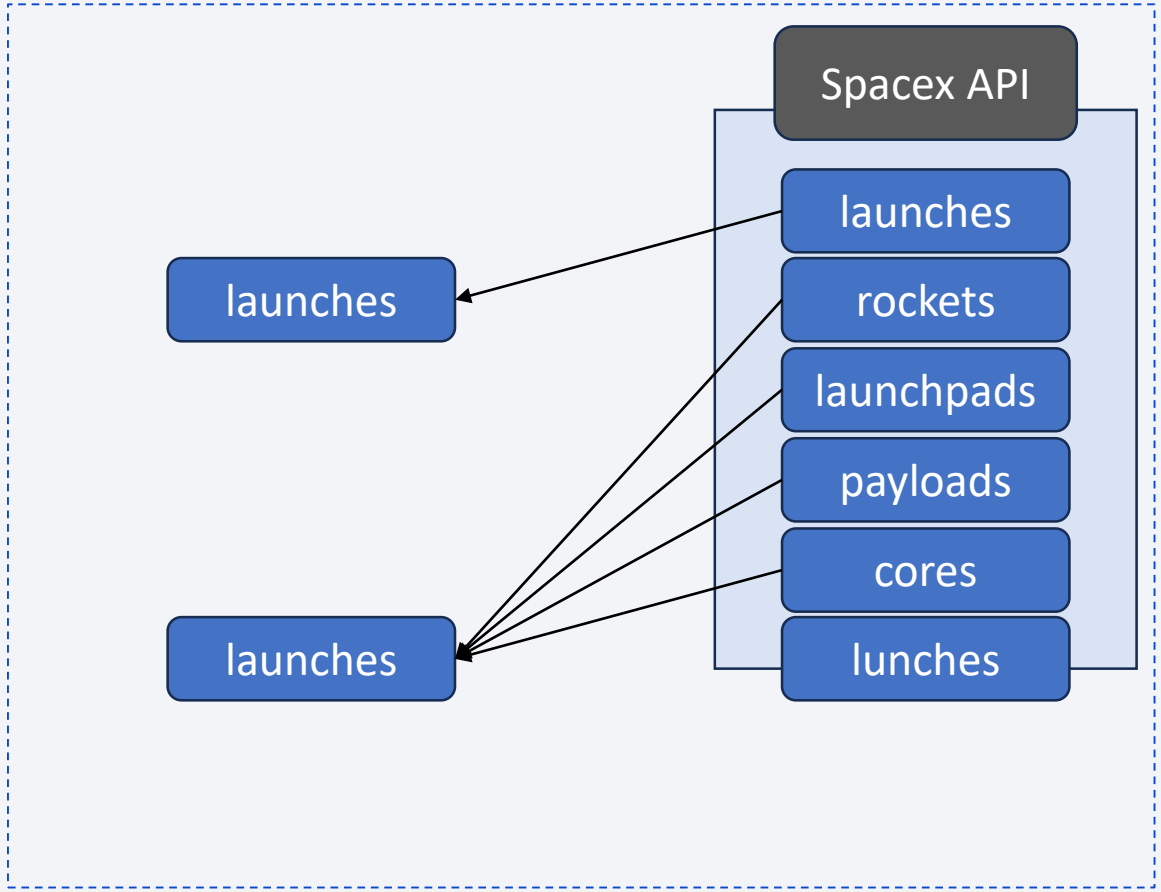
Data Collection

- The data was collected via SpaceX API calls, with the Python requests library and Web Scrapping, using requests and BeaufiulSoup libraries.
- For the SpaceX API provided the history of launches and its information;
- The Web Scrapping was used to access the Falcon 9 Wiki page and get complementary information about the rockets.

Data Collection – SpaceX API

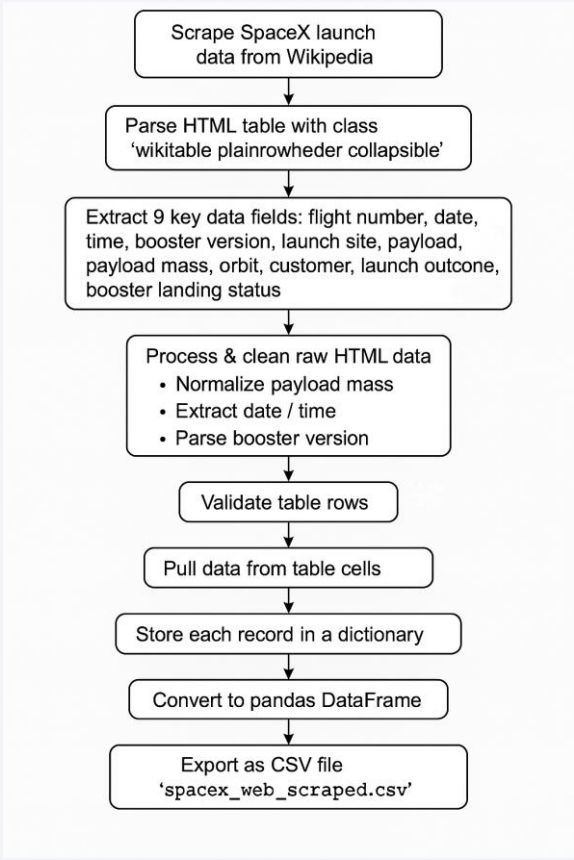
- First, the rocket launch history was gathered by **launches** endpoint;
- The rockets, launchpads, payloads, and cores data foreign keys from other endpoints, so helper functions were created to make these API calls and complete the data with meaningful content.

- SpaceX API calls Notebook



Data Collection - Scraping

- The script scrapes SpaceX launch data from Wikipedia tables, validates rows by numeric flight numbers, and extracts 9 fields per launch using helper functions. It stores data in a dictionary structure and exports to CSV.
- Scrapping Notebook



Data Wrangling

- Data Loading: Load SpaceX dataset and display first rows.
- Data Quality: Check missing values percentage and data types.
- Exploratory Analysis: Examine launch sites, orbits, and landing outcomes.
- Binary Classification: Map bad outcomes to 0 (failed) and others to 1 (success).
- Results: Calculate success rate and export cleaned dataset.
- Outcome: Transform raw data into binary classification format for ML.
- Data Wranglign Notebook

EDA with Data Visualization

- Setup: Import pandas, numpy, matplotlib, seaborn for analysis and plotting.
- Data Loading: Download SpaceX dataset from URL and load into dataframe.
- Flight Plots: Visualize payload vs flight number and launch sites vs flight number by success.
- Site Analysis: Plot launch sites vs payload mass to identify success patterns.
- Orbit Analysis: Create success rate bar chart by orbit and scatter plots of orbit relationships.
- Time Analysis: Extract plot success rate trends over time.
- Feature Engineering: Select features and apply one-hot encoding to categorical variables

EDA with SQL

- Setup: Load SQL extension and connect to SQLite database.
- Data Import: Load SpaceX CSV from URL into SQL table.
- Basic Queries: Find distinct launch sites and filter by site patterns.
- Aggregations: Sum NASA CRS payload and average F9 v1.1 payload.
- Analysis: Find first successful landing and specific payload ranges.
- Outcomes: Count mission outcomes and find maximum payload booster.
- Temporal: Extract failure details by date and analyze landing outcomes.
- Result: SQL exploration of SpaceX launch data patterns.
- EDA with SQL Notebook

Build an Interactive Map with Folium

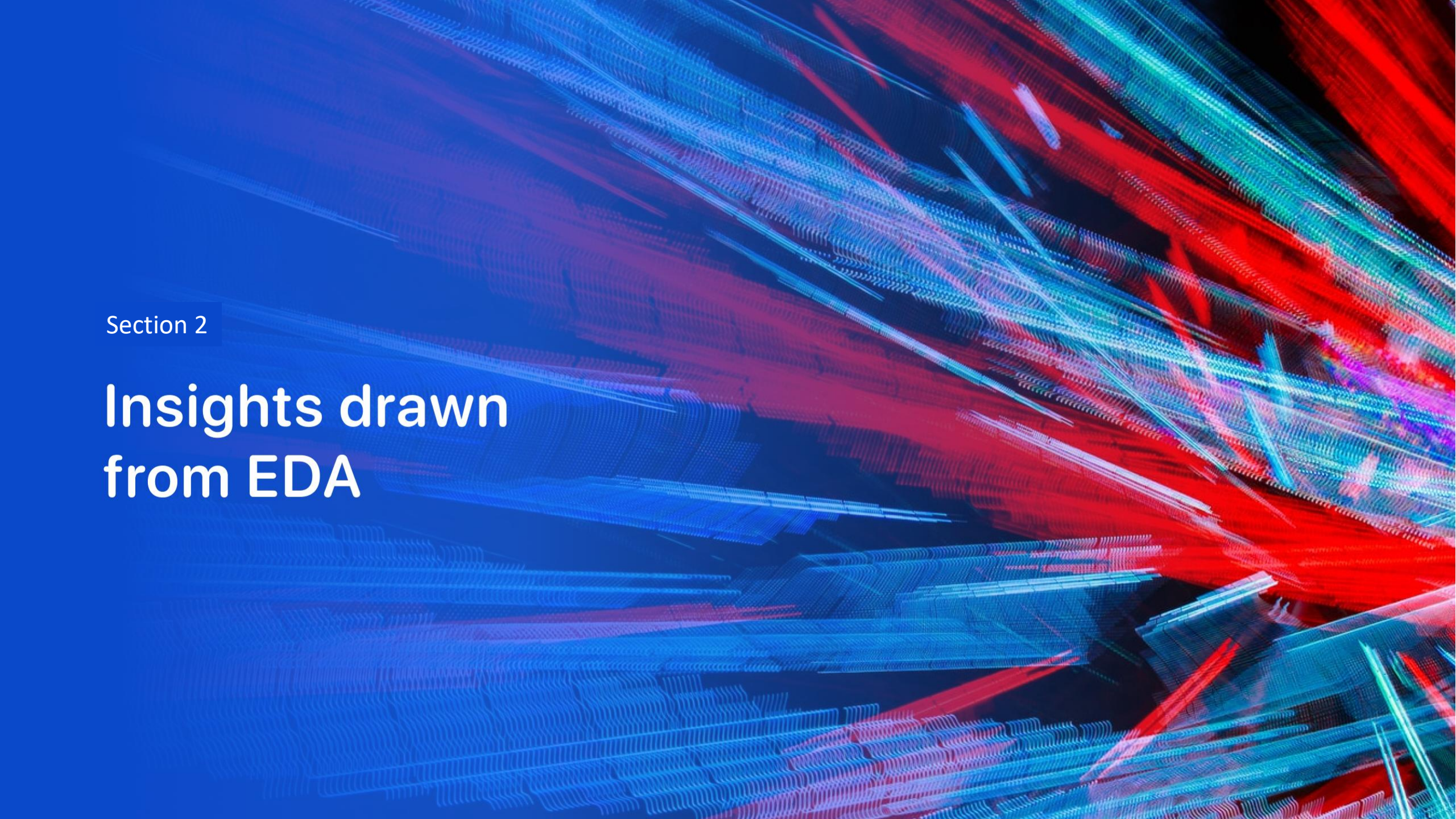
- **Map Objects Created:**
  - Orange Circles (1500m radius) around each launch site
  - Orange Text Markers showing launch site names
  - Clustered Markers (green=success, red=failure) for all launches
  - Distance Markers showing KM to coastline, city, highways, railway
  - Polylines connecting launch sites to nearby landmarks
- **Purpose:**
  - Circles & Site Markers: Identify launch locations
  - Color-coded Clusters: Visualize success/failure patterns
  - Distance Markers & Lines: Analyze proximity to infrastructure
  - Mouse Position: Enable precise coordinate analysis
  - Mouse Position Widget for coordinate trackingAdd the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- Map Notebook

Build a Dashboard with Plotly Dash

- Dashboard Summary
  - Plots/Graphs:
  - Pie Chart - Success rates by site or aggregated across all sites
  - Scatter Plot - Payload mass vs success, colored by booster version
- Interactive Components:
  - Site Dropdown - Filter by specific launch site or view all sites
  - Payload Range Slider - Filter data by payload mass (0-10,000 kg)
- Purpose:
  - Pie Chart: Compare success rates between sites and overall performance
  - Scatter Plot: Analyze payload weight impact on success with booster technology context
  - Dropdown: Focus analysis on specific sites or view aggregate trends
  - Slider: Filter by payload ranges to identify weight-related success patterns
- Dash Code

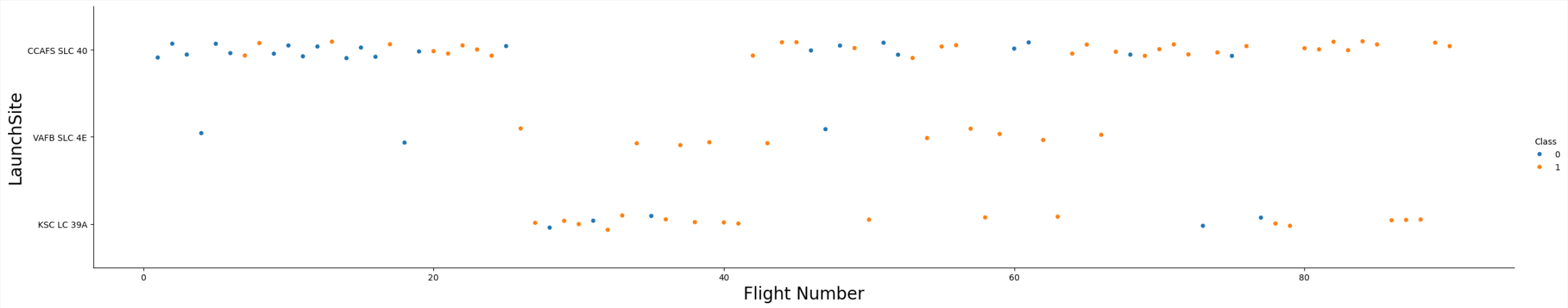
Predictive Analysis (Classification)

- **Data Prep:**
- Load features & target, apply StandardScaler, train/test split
- **Models & Hyperparameters:**
- Logistic Regression: C values, L2 penalty
- SVM: Kernels, C, gamma parameters
- Decision Tree: Criterion, depth, samples
- KNN: Neighbors, algorithms, distance
- **Process:**
- GridSearchCV with 10-fold CV for each model
- Evaluate with accuracy scores and confusion matrices
- **Result**: Compare all models to select best performer for SpaceX landing predictionAdd the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- Analysis Notebook


Section 2
Insights drawn from EDA

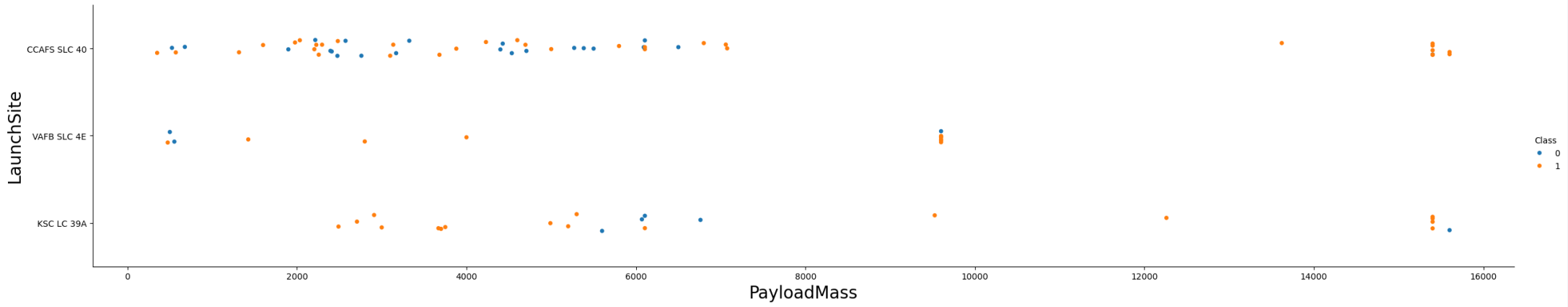
Flight Number vs. Launch Site

- Chart demonstrates SpaceX's technological maturation from experimental failures to reliable operations, with newer facilities benefiting from accumulated knowledge and advanced systems.



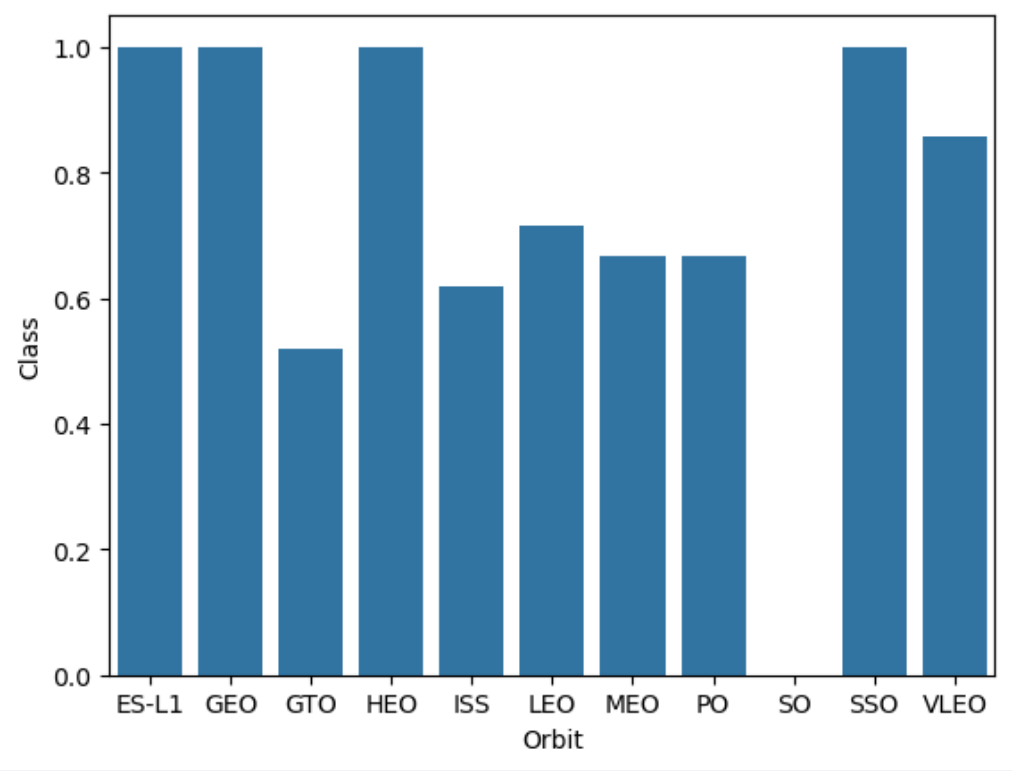
Payload vs. Launch Site

- Different launch sites handle varying payload ranges, with KSC LC-39A typically managing heavier payloads and showing higher success rates, suggesting site-specific optimization.



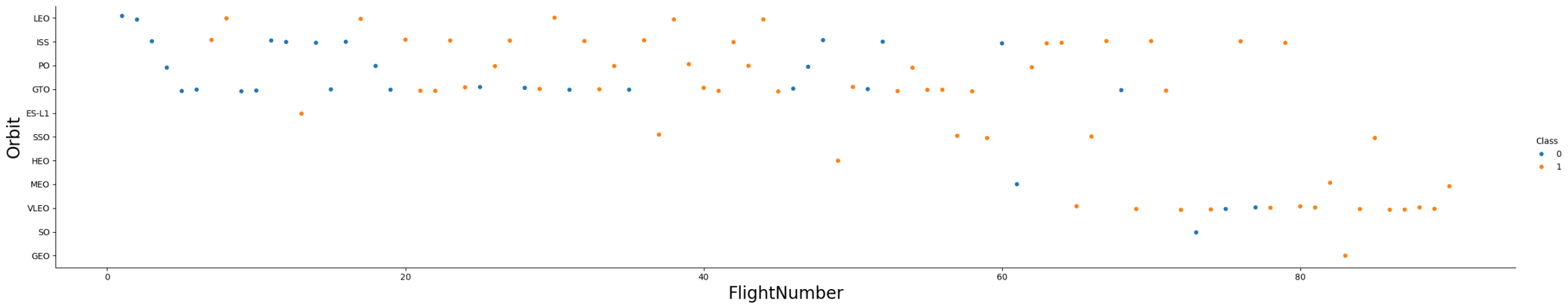
Success Rate vs. Orbit Type

- Certain orbit types demonstrate significantly higher landing success rates, indicating that mission profile and orbital mechanics directly impact recovery feasibility.



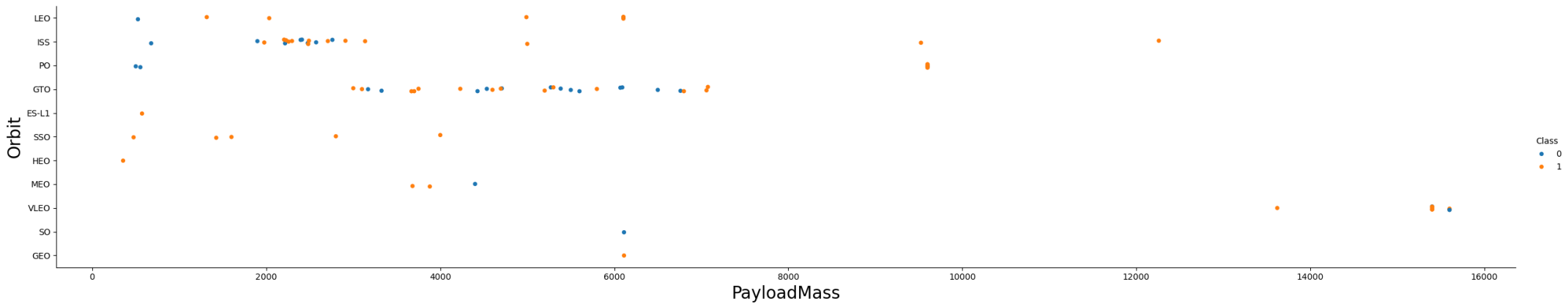
Flight Number vs. Orbit Type

- Mission complexity and orbit selection evolved over time, with later flights targeting more diverse orbits while maintaining improved success rates through technological advancement.



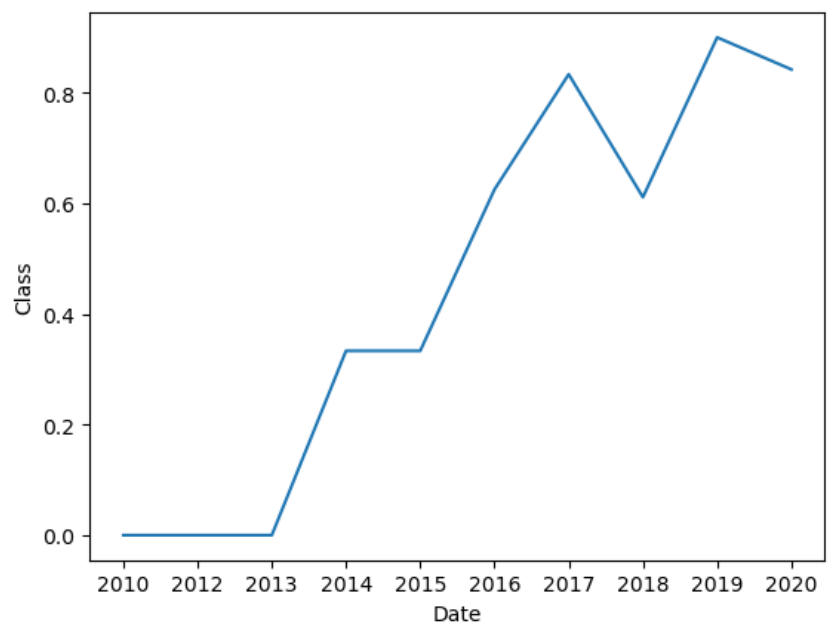
Payload vs. Orbit Type

- Specific orbit-payload combinations show distinct success patterns, revealing optimal operational parameters for successful first-stage recovery based on mission requirements



Launch Success Yearly Trend

- Clear upward trend demonstrates SpaceX's rapid learning curve and systematic improvement, with success rates dramatically increasing from early experimental phases to current operational excellence.



All Launch Site Names

- The unique Launch Sites:
- CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E, CCAFS LC-40

```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

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

Sum_Payload
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS 'Mean_Payload' FROM SPACEXTABLE WHERE "Booster_Version" LIKE 'F9 v1.1 %'
```

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

Mean_Payload
2337.8

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%sql SELECT * FROM SPACEXTABLE WHERE DATE = (SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success %')
```

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

Python

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

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

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT SUBSTR("Date", 0, 5) AS year, CASE SUBSTR("Date", 6, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN
```

Python

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

year	month_name	Booster_Version	Launch_Site
2015	January	F9 v1.1 B1012	CCAFS LC-40
2015	April	F9 v1.1 B1015	CCAFS LC-40

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%sql SELECT "Landing_Outcome", COUNT(*) AS 'Total_Outcome' FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Ou
```

Python

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

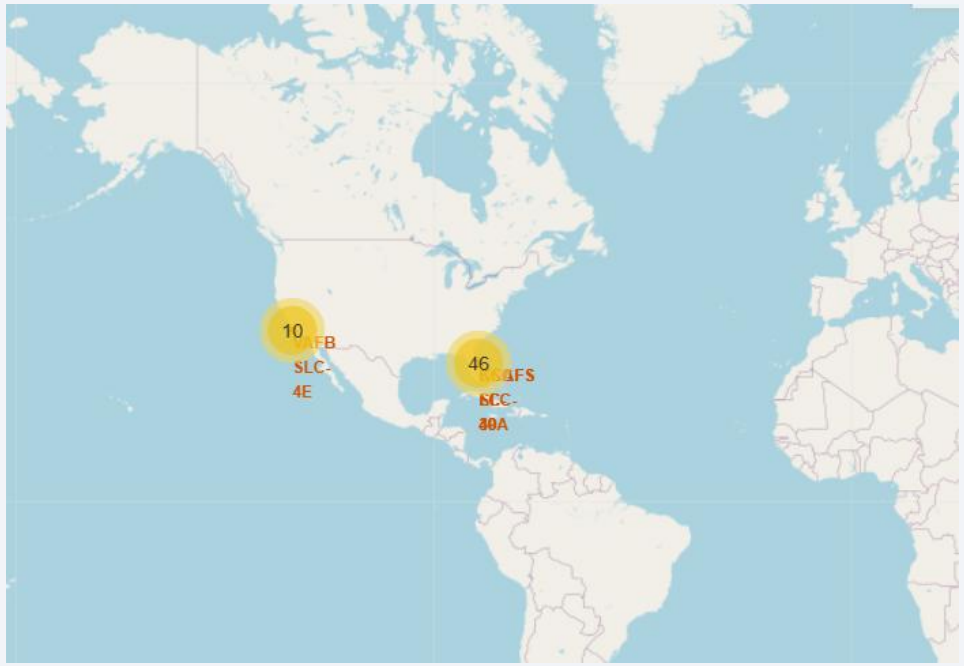
Landing_Outcome	Total_Outcome
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All launch sites’ location markers on a global map

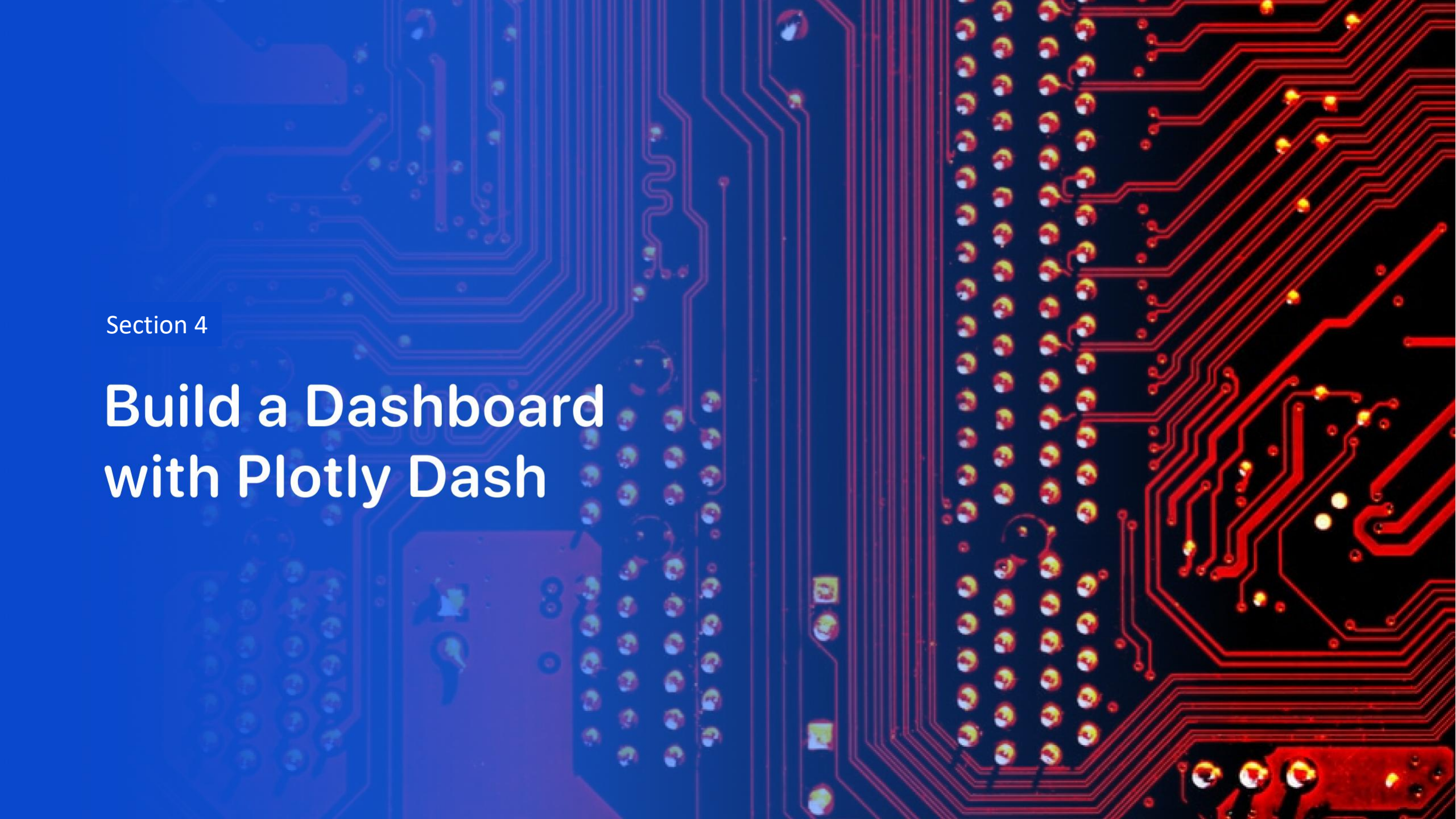
- Launch sites are near the equator, where Earth's rotation boosts rocket speed by 1670 km/h due to inertia.
- Equatorial speed helps spacecraft maintain necessary orbital velocity after launch, making launches more energy-efficient.
- Coastal launch sites reduce risk by ensuring rocket debris falls safely over the ocean, away from populated areas.



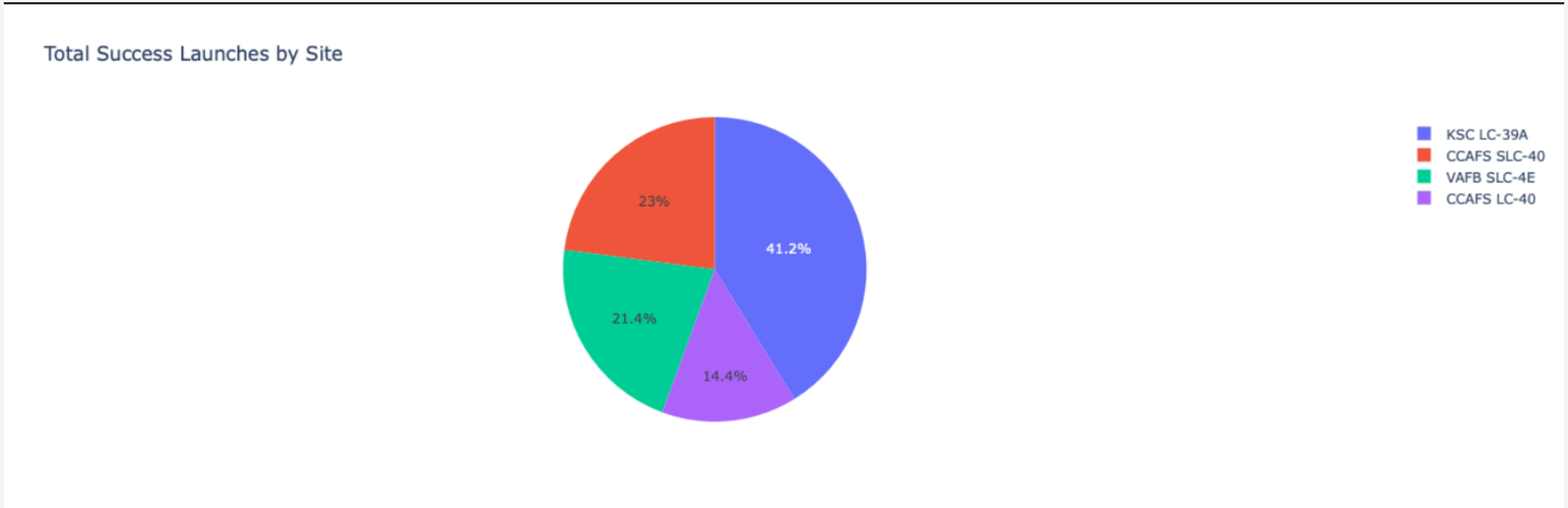
Colour-labeled launch records

- Green markers show success; red markers indicate launch failure for quick visual comparison.
- Colors help identify reliable launch sites easily.

Distance from the launch site

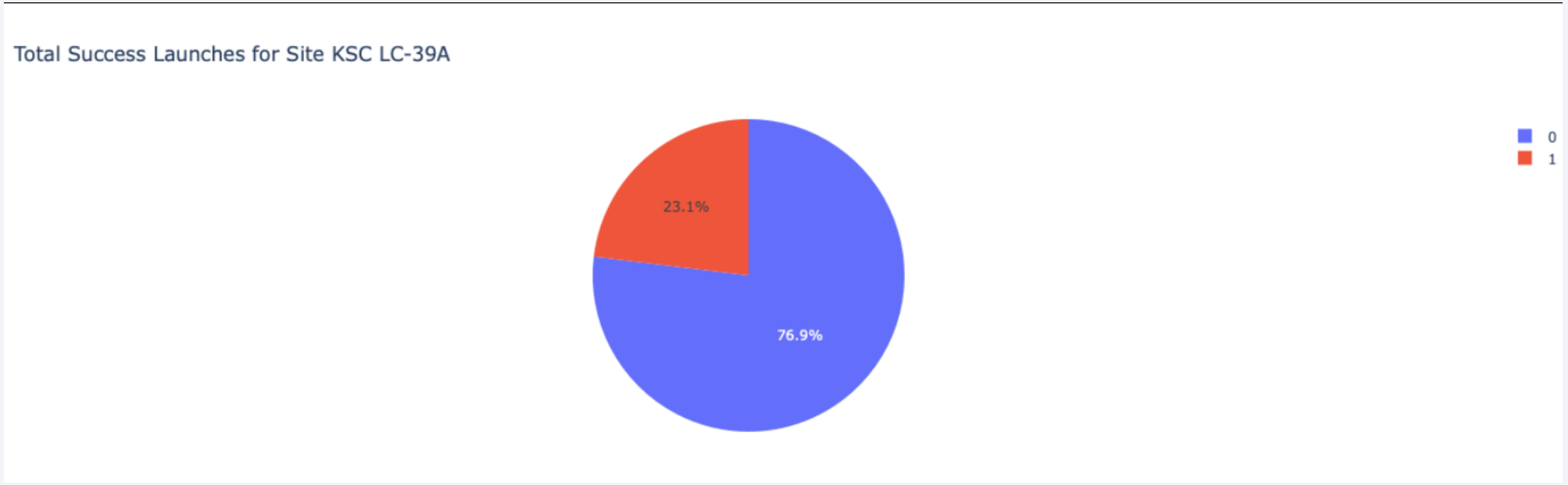

Section 4
Build a Dashboard with Plotly Dash

Launch Success



- The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches

Launch site with highest launch success



Payload Mass vs. Launch Outcome

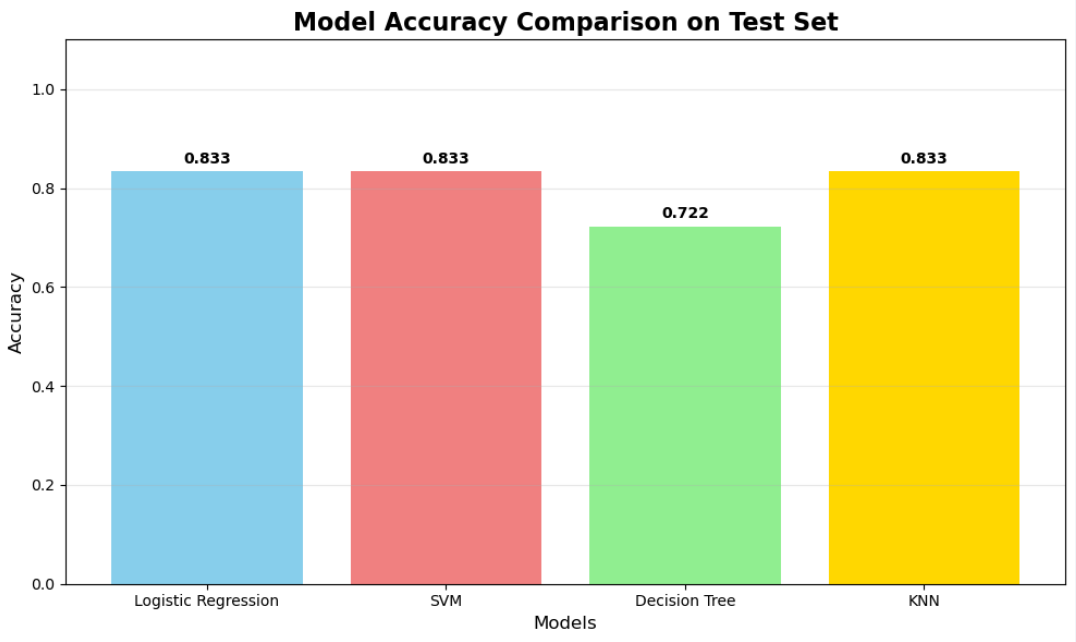
Section 5

Predictive Analysis (Classification)

Classification Accuracy

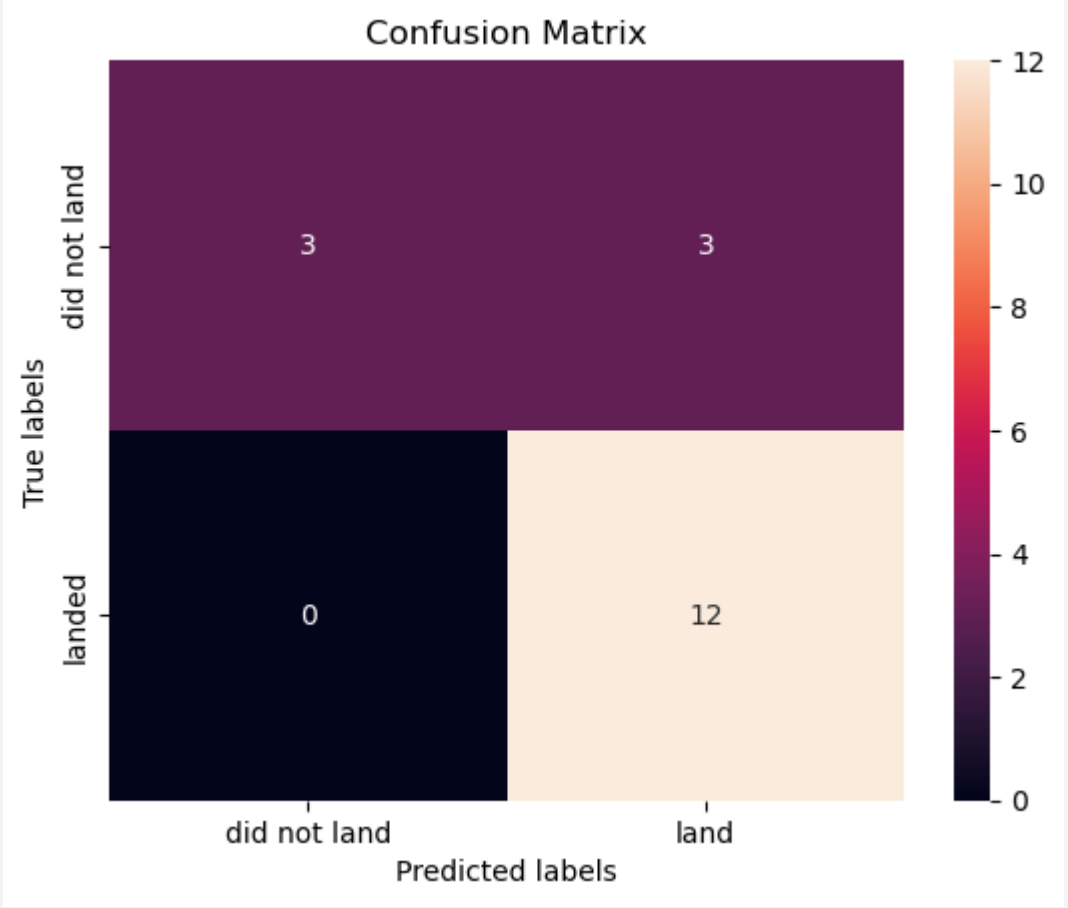
Best Model: Logistic Regression (83.33% Accuracy)

- • Interpretability: Clear coefficients show which factors influence landing success
- • Efficiency: Fastest training and prediction among all models tested
- • Probability Output: Provides confidence levels, not just yes/no predictions



Confusion Matrix

- What Actually Happened vs What We Predicted:
- Perfect Score on Successful Landings: Our model got it right every single time a rocket actually landed - it never missed one! (0 false negatives)
- Some False Alarms: The model got excited and predicted 3 rockets would land when they actually didn't. Think of it like being overly optimistic about landing success.
- Great Overall Performance: Out of 18 test cases, the model got 15 correct (83.3% accuracy)



Conclusions

- • **83.33% Accuracy**: Logistic Regression achieved highest prediction accuracy
- • **Perfect Landing Detection**: Never missed a successful landing (0 false negatives)
- • **4 Models Tested**: Logistic Regression, SVM, Decision Tree, KNN compared
- • **Best Choice**: Logistic Regression - simple, fast, interpretable
- • **Conservative Approach**: Better to prepare unnecessarily than miss landings
- • **Cost Savings**: Optimizes recovery operations and resource allocation
- • **Business Ready**: Model can be implemented in SpaceX mission planning

Thank you!