IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hamza Husein
4th April, 2025

Outline

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

Executive Summary

Summary of methodologies

To predict whether SpaceX's Falcon 9 first stage will land successfully, the following steps were followed:

1. **Data Collection** – Gathered public data using APIs and web scraping.
2. **Data Wrangling** – Cleaned and structured the raw data for analysis.
3. **Exploratory Analysis** – Used **SQL queries, statistics, and visualizations** to explore trends.
4. **Key Findings** – Discovered that **flight number, payload mass, launch site, and orbit type** influence launch success.
    1. Success rates **increased steadily from 2013 to 2020**.
    2. Launch sites are typically **near the equator and coastlines** for efficiency.
5. **Predictive Modeling** – Built and tested machine learning models, evaluating accuracy with a **confusion matrix**.
    1. The **decision tree model** performed best with the highest accuracy.

Summary of all results

With a **highly accurate predictive model**, we can confidently determine if the Falcon 9's first stage will land successfully—helping estimate launch costs and supporting competitive decision-making (e.g., for companies like **SPACE Y**).

Introduction

The commercial space age has arrived, making space travel more affordable, with SpaceX leading the way due to cost-efficient rocket launches. SpaceX's Falcon 9 launches cost **62million; significantly cheaper than competitors’ 165 million**—largely because SpaceX reuses the first stage.

Key Insight: Predicting whether the Falcon 9's first stage will land successfully helps estimate launch costs. This is useful for competitors like **SPACE Y** when bidding against SpaceX.

Project Goal: Predict the success of Falcon 9's first-stage landing to determine launch cost implications.

Section 1

Methodology

Methodology

Executive Summary

1. Data Collection Methodology:

- **SpaceX REST API** – Provides detailed launch records, including:
  - Rocket specifications
  - Payload details
  - Launch & landing data (success/failure)
- **Web Scraping** – Used to extract Falcon 9 launch history from Wikipedia using Python's **BeautifulSoup**.
  - HTML tables were parsed and converted into a structured **Pandas DataFrame** for analysis.

2. Data Wrangling

To ensure data quality, we:

- Cleaned and transformed raw API/web-scraped data.
- Handled missing values (**Nulls**) and inconsistencies.
- Sampled data to maintain relevance for predicting **first-stage landing success**.
- This structured dataset enables accurate **visualization, analysis, and predictive modeling**.

Data Collection

We gathered SpaceX launch data from two key sources:

1. **SpaceX REST API** – Provides detailed launch records, including:
    1. Rocket specifications
    2. Payload details
    3. Launch and landing outcomes
2. **Web Scraping with BeautifulSoup** – Extracted additional Falcon 9 launch history from Wikipedia by parsing HTML tables into structured data.

- This combined approach ensures comprehensive and reliable datasets for analysis.

Data Collection – SpaceX API

To retrieve past launch records, we:

1. **Targeted a specific API endpoint** using its URL.
2. **Sent a GET request** (Python's requests library) to fetch the data.
3. **Converted the response** to structured JSON using .json(), yielding a list of launch records in JSON format.

Refer to the Github link below for more details:
https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/jupyter-labs-spacex-data-collection-api.ipynb

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [20]:
```
response = requests.get(spacex_url)
response.json()
```

Out[20]:

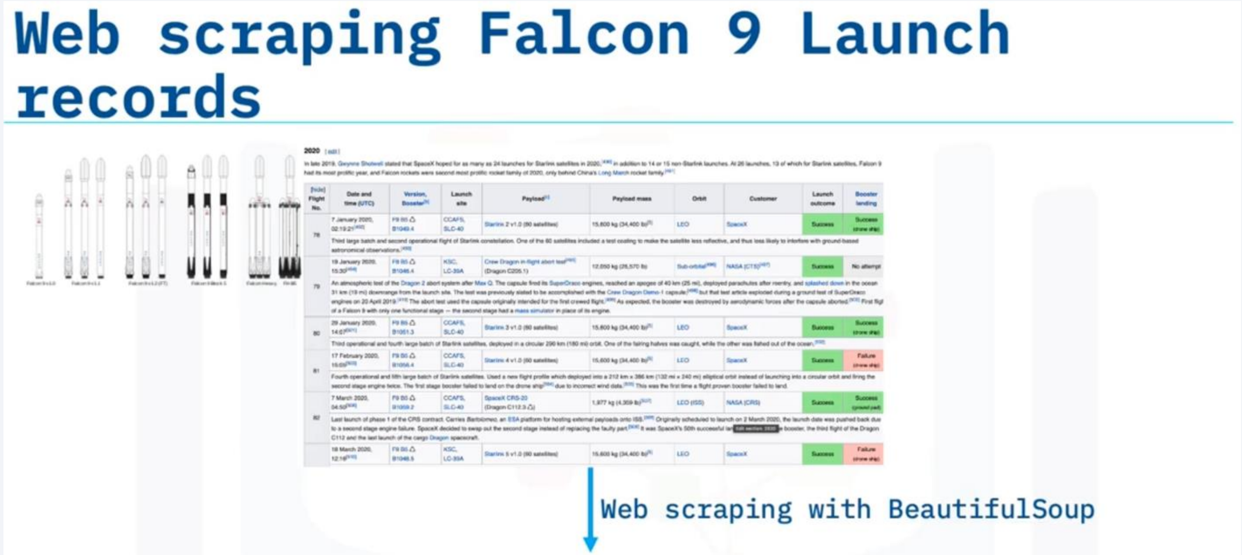
```
[{'fairings': {'reused': False,
   'recovery_attempt': False,
   'recovered': False,
   'ships': []},
  'links': {'patch': {'small': 'https://images2.imgbox.com/94/f2/NN6Ph45r_o.png',
    'large': 'https://images2.imgbox.com/5b/02/QcxHUb5V_o.png'},
   'reddit': {'campaign': None,
    'launch': None,
    'media': None,
    'recovery': None},
   'flickr': {'small': [], 'original': []},
   'presskit': None,
   'webcast': 'https://www.youtube.com/watch?v=0a_00nJ_Y88',
   'youtube_id': '0a_00nJ_Y88',
   'article': 'https://www.space.com/2196-spacex-inaugural-falcon-1-rocket-lost-launch.html',
   'wikipedia': 'https://en.wikipedia.org/wiki/DemoSat'},
  'static_fire_date_utc': '2006-03-17T00:00:00.000Z',
  'static_fire_date_unix': 1142553600,
  'net': False,
  'window': 0,
  'rocket': '5e9d0d95eda69955f709d1eb',
  'success': False,
  'failures': [{'time': 33,
    'altitude': None,
    'reason': 'merlin engine failure'}],
  'details': 'Engine failure at 33 seconds and loss of vehicle',
  'crew': [],
  'ships': [],
```

Data Collection - Scraping

We extracted key launch records by:

1. **Scraping HTML tables** (using Python's BeautifulSoup) containing Falcon 9 historical data
2. **Parsing and cleaning** the extracted table data
3. **Converting to a Pandas DataFrame** for structured analysis and visualization

This method supplements API data with additional launch records for comprehensive insights.



	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.047721
4	6	2010-06-04	Falcon 9	NaN	LEO	CCAFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857

Refer to the Github link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/jupyter-labs-webscraping.ipynb

Data Wrangling

To transform SpaceX's API response into usable data, we did the following:

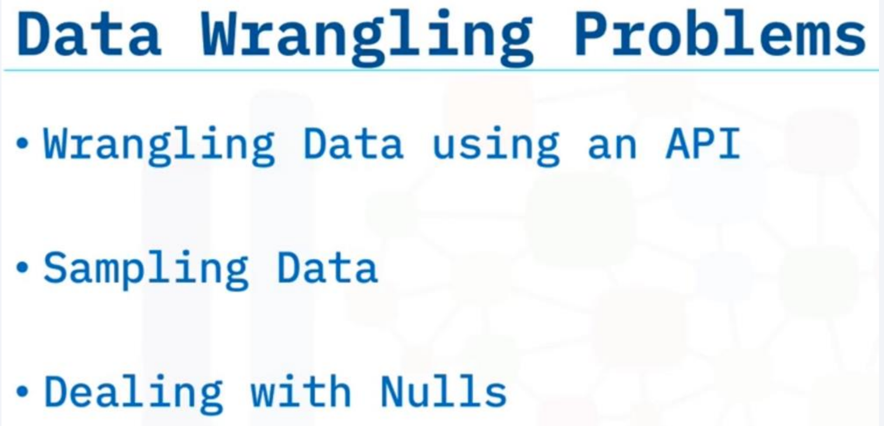
1. JSON Conversion

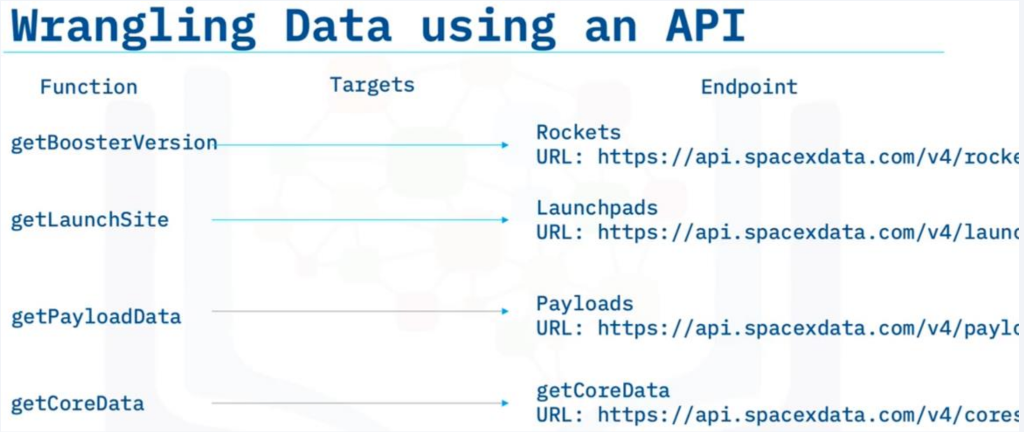
1. API responses come as a list of JSON objects (one per launch)
2. Used json_normalize() to convert nested JSON into a flat Pandas DataFrame

2. Data Enrichment

1. Some columns (like rocket/launchpad IDs) only contained reference numbers
2. Made additional API calls to:
    1. Fetch booster details
    2. Launchpad locations
    3. Payload specifications
    4. Core stage data
3. Stored enriched data in lists for final dataset assembly

Result: A complete, analysis-ready dataset combining normalized JSON with detailed launch information.





Data Wrangling

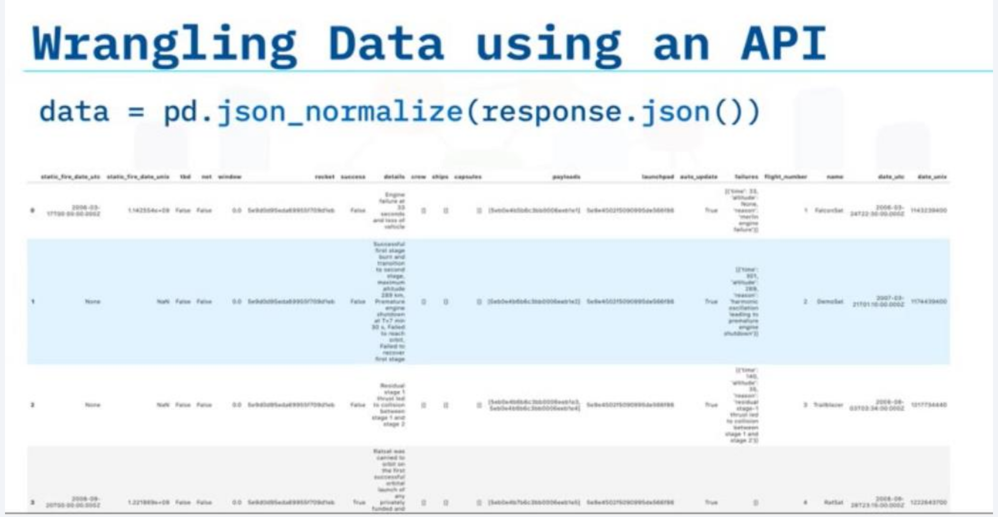
- To refine our dataset for Falcon 9 analysis:

1. **Filtering Falcon 1 Launches**
    1. Removed Falcon 1 records to focus exclusively on Falcon 9 launches.
2. **Handling Missing Data**
    1. Addressed NULL values by:
        1. Calculating the **mean** for numerical columns
        2. Replacing NULLs with the computed mean
    2. Ensures dataset completeness for accurate analysis.

- **Result:** A clean, Falcon 9-specific dataset ready for exploration and modeling.

Refer to the GitHub link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/labs-jupyter-spacex-Data%20wrangling.ipynb



	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.047721
4	6	2010-06-04	Falcon 9	NaN	LEO	CCAFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857

EDA with Data Visualization

To analyze how different factors influence launch success, we created these visualizations:

1. **Flight & Payload Analysis**
    1. *Flight Number vs. Payload Mass* (Scatter) – Trends in payload capacity over time
    2. *Flight Number vs. Launch Site* (Scatter) – Launch site preferences by mission count
2. **Payload & Location Impact**
    1. *Payload Mass vs. Launch Site* (Scatter) – How payload size varies by location
    2. *Payload Mass vs. Orbit Type* (Scatter) – Orbit-specific payload requirements
3. **Success Patterns**
    1. *Orbit Type vs. Success Rate* (Bar) – Which orbits have highest success rates
    2. *Yearly Launch Success* (Line) – Improvement in success rates over time

Purpose: Identify which variables (payload, orbit, launch site) most strongly correlate with successful landings.

Refer to the GitHub link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/edadataviz.ipynb

EDA with SQL

- **1. Launch Site Information**

  -- Get all unique launch sites

  %sql select distinct "Launch_Site" from SPACEXTABLE

  -- Show 5 launches from sites starting with 'CCA'

  %sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5

- **2. Payload Analysis**

  -- Total payload mass for NASA (CRS) missions

  %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer"='NASA (CRS)’

  -- Average payload for F9 v1.1 booster

  %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" = 'F9 v1.1’

  -- Boosters with max payload capacity

  %sql select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_=(select MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)

EDA with SQL

- **3. Landing Outcomes**

  -- First successful ground pad landing date

  %sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)’

  -- Successful drone ship landings (payload 4000-6000 kg)

  %sql select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000

  -- Mission success/failure counts

  %sql select count(substr("Mission_Outcome",0,8)) as outcome_count, substr("Mission_Outcome",0,8) as outcome from SPACEXTABLE group by outcome

- **4. Historical Data**

  -- 2015 drone ship failures with month breakdown

  %sql select substr("Date",0,5) as year, substr("Date", 6,2) as month, "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where "Landing_Outcome"='Failure (drone ship)' and substr("Date",0,5)='2015’

  -- Landing outcome ranking (2010-2017)

  %sql select "Landing_Outcome", count("Landing_Outcome") as outcome_count from SPACEXTABLE where "Date" between '2010-06-04' and '2017-03-20' group by "Landing_Outcome" order by outcome_count desc

Refer to the GitHub link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

We conducted geospatial analysis of SpaceX launch sites using Folium to create interactive maps with these key features:

1. Launch Site Mapping

- Plotted all launch locations on a Folium map with:
  - Highlighted circular areas
  - Text labels at each coordinate
- Key observations:
  - All sites are clustered **near the equator**
  - Located in **close proximity to coastlines**

2. Success/Failure Visualization

- Enhanced markers show launch outcomes:
  - **Green markers** = Successful launches (class=1)
  - **Red markers** = Failed launches (class=0)
- Implemented **marker clustering** to handle overlapping coordinates from multiple launches at the same site

3. Proximity Analysis

- Added tools to measure distances to key landmarks:
  - Enabled **MousePosition** to get real-time coordinates when hovering
  - Drew **PolyLines** between launch sites and nearby features:
    - Coastlines
    - Highways
    - Railroads
- Allows precise measurement of logistical relationships

Key Insights: This interactive analysis reveals how geographical factors like coastal access and equatorial position correlate with launch success rates.

Refer to the GitHub link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

We developed a real-time analytics dashboard to explore SpaceX launch data through interactive visualizations. The application features:

1. Input Components

- **Launch Site Dropdown**: Allows selection of specific launch locations
- **Payload Range Slider**: Enables filtering by payload mass (kg)

2. Interactive Visualizations

- **Success Rate Pie Chart**:
  - Dynamically updates based on selected launch site
  - Displays proportion of successful vs failed launches
- **Payload vs Outcome Scatter Plot**:
  - Adjusts based on payload range selection
  - Shows correlation between payload mass and launch success

3. Technical Implementation

- Implemented callback functions to:
  - Generate the pie chart when users select a launch site
  - Update the scatter plot when adjusting the payload range
- Designed for real-time data exploration and pattern discovery

Key Functionality: Users can investigate how launch sites and payload characteristics relate to mission outcomes through immediate visual feedback.

Value: Provides actionable insights through an intuitive, user-controlled interface without requiring technical expertise.

Refer to the GitHub link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/spacex-dash-app.py

Predictive Analysis (Classification)

1. Data Preparation

- **Preprocessing**: Standardized data to ensure consistent scales for all features
- **Train/Test Split**: Divided dataset into training (model development) and testing (final evaluation) subsets

2. Model Training & Optimization

- Implemented four classification algorithms:
  - Logistic Regression
  - Support Vector Machines (SVM)
  - Decision Tree Classifier
  - K-Nearest Neighbors (KNN)
- Conducted **Grid Search** to identify optimal hyperparameters for each model
- Selected best-performing model based on training accuracy

3. Evaluation

- Generated **confusion matrices** to assess model performance
- Compared results across algorithms to determine most reliable predictor

Objective: Create an accurate system for forecasting Falcon 9 first-stage landing success to inform cost estimates and mission planning.

Key Benefit: Methodical approach ensures robust, data-driven predictions through systematic testing of multiple algorithms.

Refer to the GitHub link below for more details:

https://github.com/husham35/ibm_data_science/blob/main/applied_data-science_capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

Section 2

Insights drawn from EDA

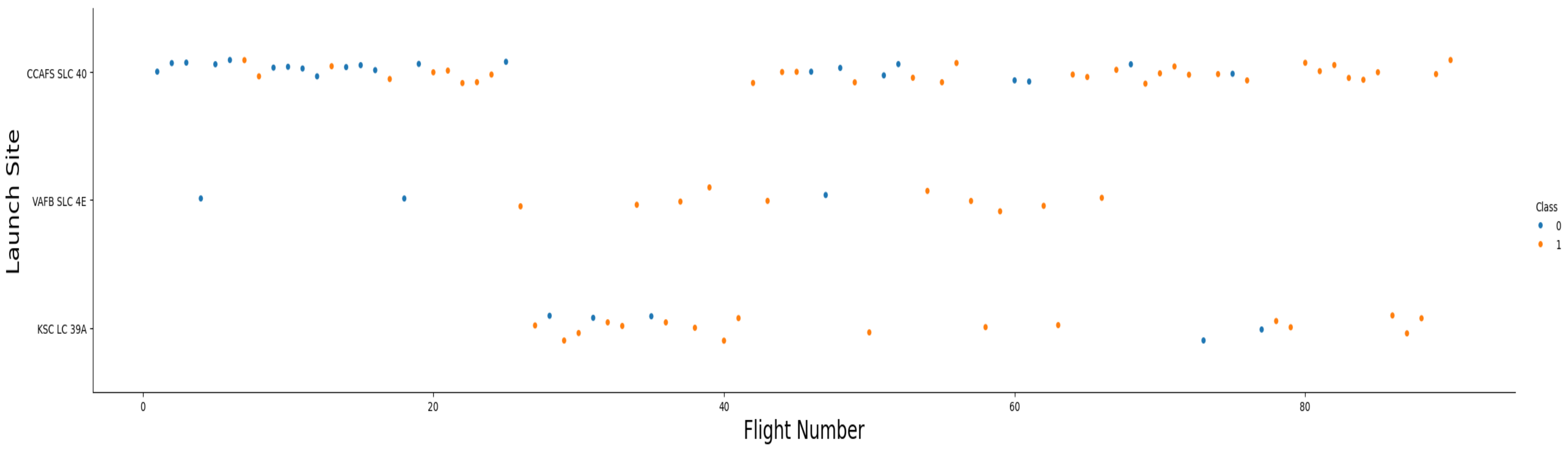
Flight Number vs. Launch Site

Key observations:

- **KSC LC-39A** and **VAFB SLC 4E** show higher success rates (**77%**)
- **CCAFS LC-40** has a lower success rate (**60%**)

Implications:

- Site-specific factors (location, infrastructure, or mission profiles) may influence outcomes
- Mission planning could prioritize higher-success sites when possible



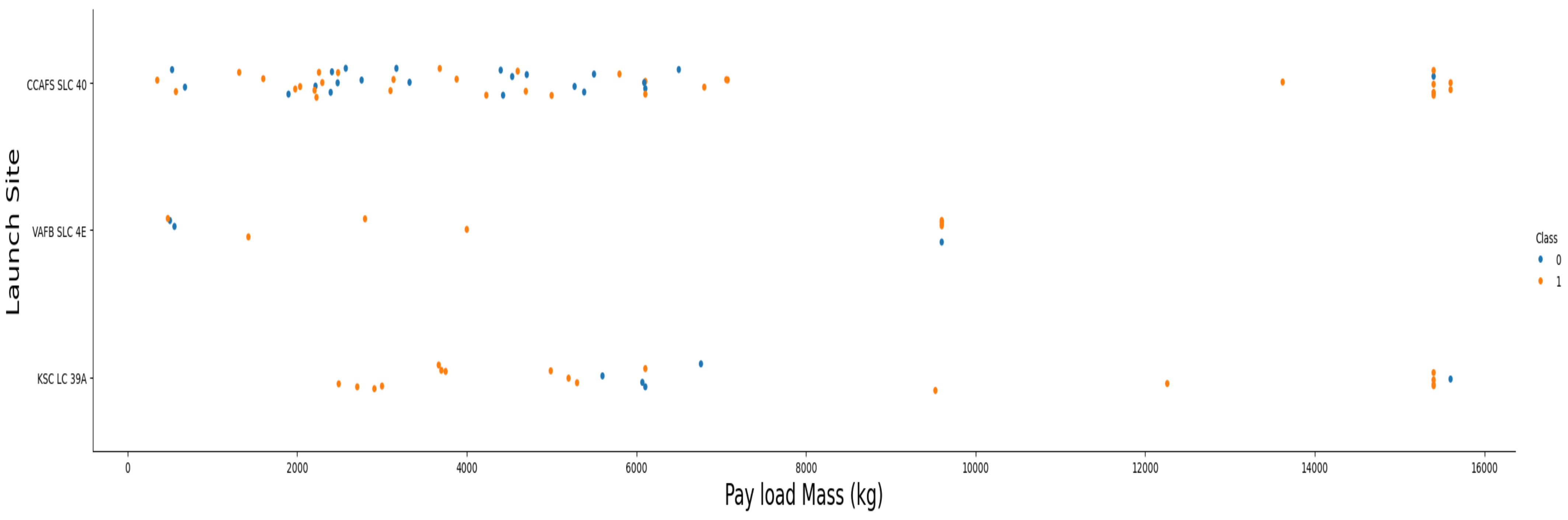
Payload vs. Launch Site

Key Finding:

- **VAFB SLC-4E** has **no recorded launches** with heavy payloads (>10,000 kg), based on the scatter plot analysis.

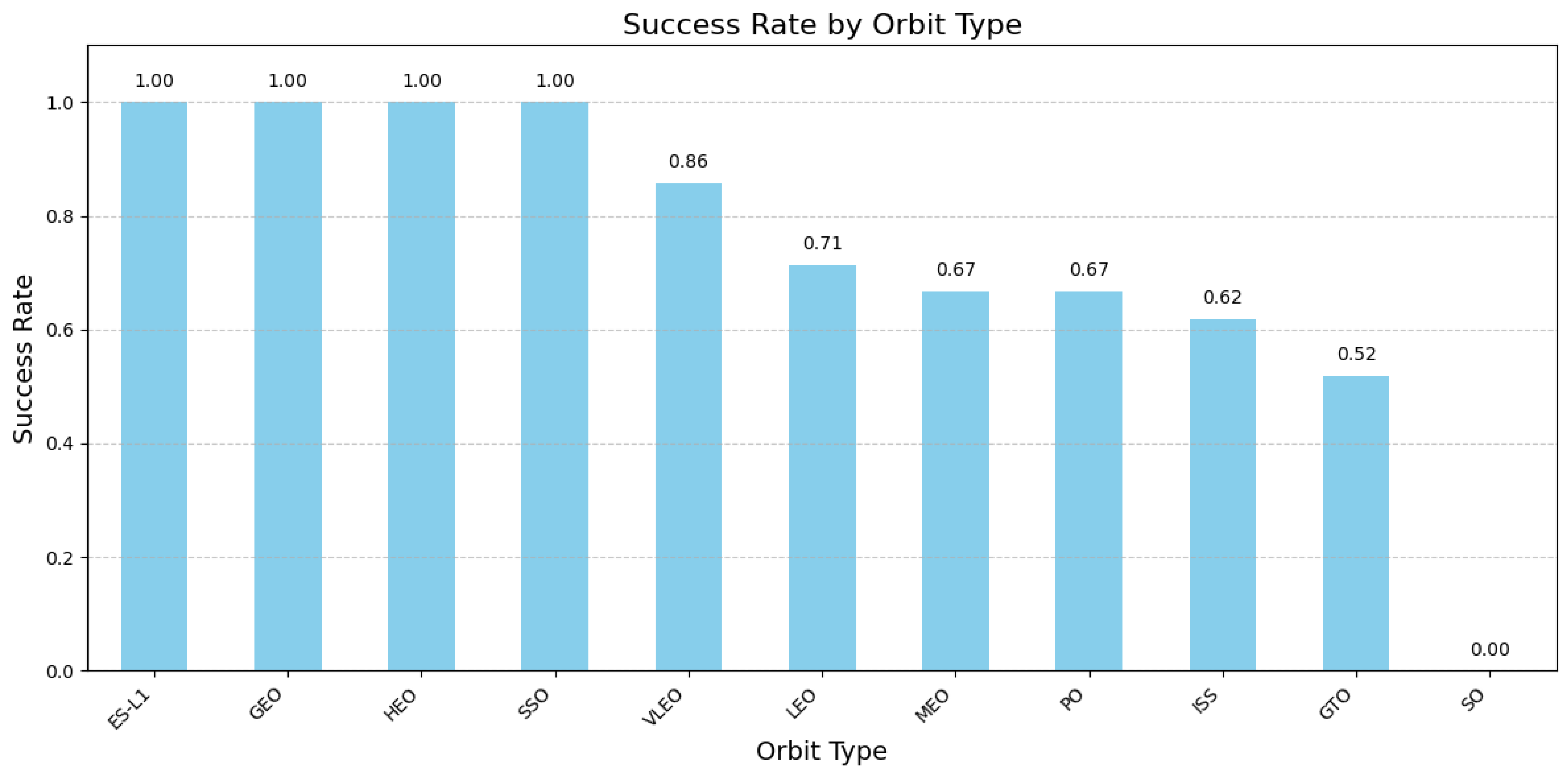
Implications:

- Payload mass requirements directly influence launch site selection
- Heavy missions (>10,000 kg) must use alternative sites like **KSC LC-39A** or **CCAFS LC-40**



Success Rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO, and SSO have 100% success rate.



Flight Number vs. Orbit Type

LEO (Low Earth Orbit) Missions

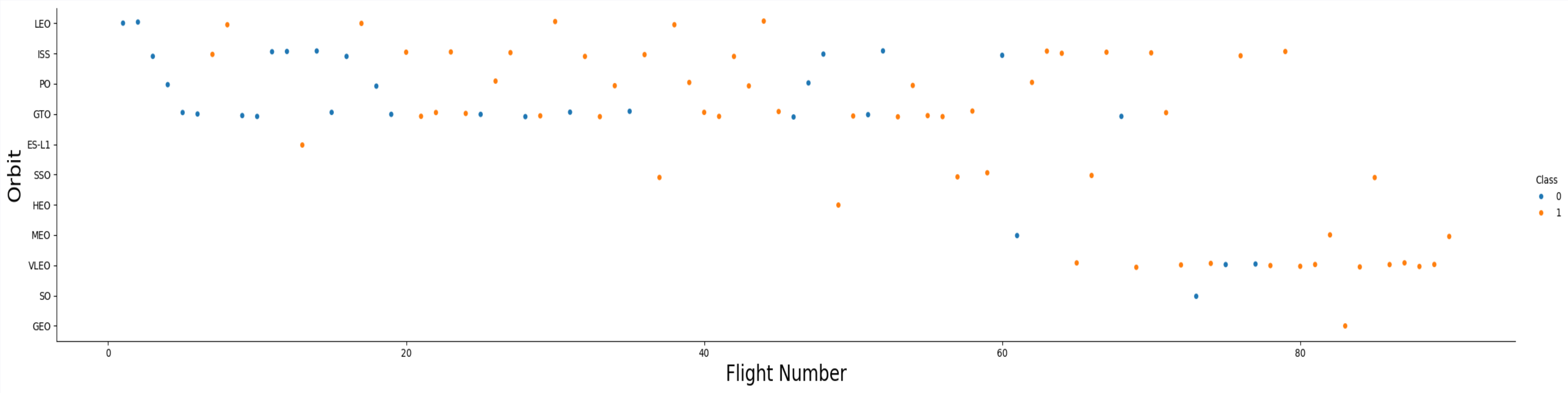
- **Clear Correlation**: Success rates improve with increasing flight numbers
- **Reasoning**: Likely due to SpaceX's iterative refinements in LEO launch/recovery operations

GTO (Geostationary Transfer Orbit) Missions

- **No Apparent Pattern**: Flight number shows no consistent impact on success rates
- **Potential Factors**:
  - Higher energy demands of GTO missions complicate recovery
  - Limited sample size of GTO launches

Key Insight:

Flight experience boosts reliability for LEO but not GTO, suggesting orbit type significantly influences landing outcomes.



Payload vs. Orbit Type

High Success Orbits (with heavy payloads)

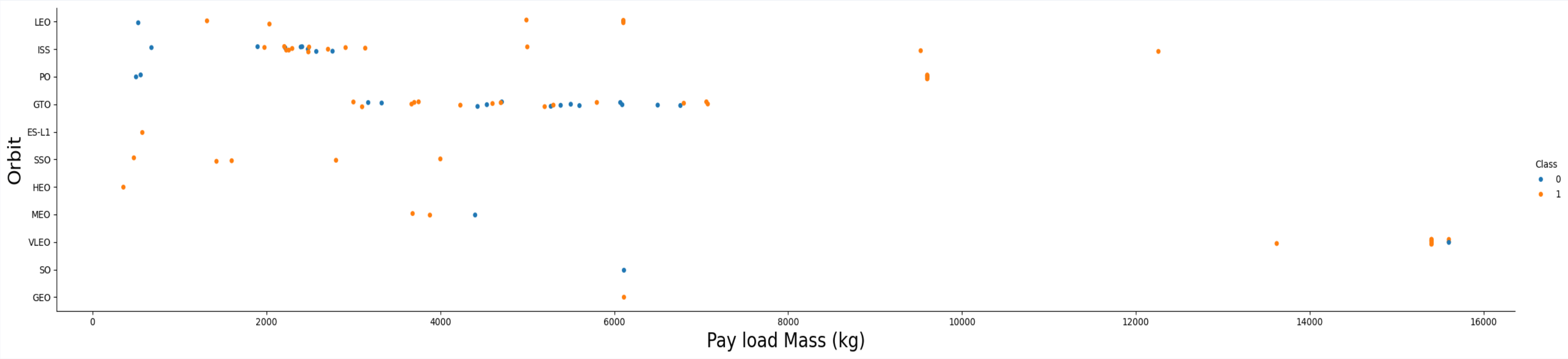
- **Polar, LEO, and ISS missions** show strong landing success rates even with heavy payloads
- Indicates reliable first-stage recovery for these flight profiles

GTO Orbit Challenges

- Displays **mixed results** for heavy payload missions
- Both successful and unsuccessful landings observed
- Suggests greater variability in GTO mission outcomes

Key Insight: Orbit type significantly impacts landing success, with GTO missions presenting unique challenges for booster recovery compared to more predictable LEO/Polar/ISS trajectories.

Operational Implication: Mission planners should consider orbit-specific success trends when evaluating recovery feasibility for heavy payload launches.



Launch Success Yearly Trend

1. Improvement Over Time:

1. Success rates **increased significantly** from **33% (2010)** to **84% (2018)**.
2. Peaked at **90% in 2017**, reflecting SpaceX's advancements in reusable rocket technology.

2. Yearly Fluctuations:

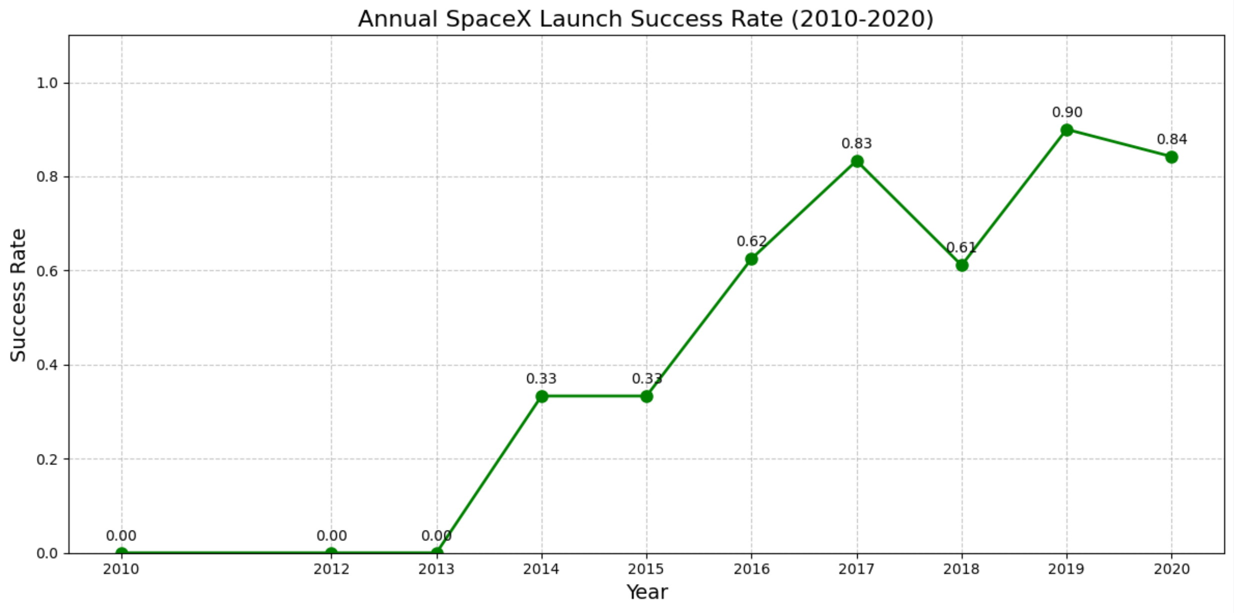
1. **2012–2014**: Rates stabilized near **60–65%**, suggesting iterative testing phases.
2. **2015–2017**: Rapid improvement (**83% → 90%**) as Falcon 9 refinements matured.
3. **2018–2020**: Slight dip but maintained **~84% success**, indicating consistent reliability.

Notable Observations

- **2010–2012**: Low initial rates (33–67%) align with early Falcon 9 development and testing.
- **2017**: **Highest success rate (90%)** correlates with SpaceX's mastery of booster landings.

Implications

- **Reusability Milestones**: Post-2015 surge matches SpaceX's focus on first-stage recovery.
- **Operational Maturity**: Post-2017 stability reflects refined launch processes.



All Launch Site Names

To find the names of the unique launch sites, we use the following magic SQL code below:

```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

Launch_Site:

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

Launch Site Names Begin with 'CCA'

To find 5 records where launch sites begin with `CCA` we use the following magic SQL code below:

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

```
* 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

+ Code + Markdown

Total Payload Mass

To calculate the total payload carried by boosters from NASA, we use the following magic SQL code below:

%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

To calculate the average payload mass carried by booster version F9 v1.1, we use the following magic SQL code below:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

To find the date of the first successful landing outcome on ground pad, we use the following magic SQL code below:

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

To list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, we use the following SQL code:

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND 4000 < PAYLOAD_MASS__KG_ < 6000
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

To calculate the total number of successful and failure mission outcomes, we use the following magic SQL code below:

```
%sql SELECT SUM(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS Successful_Missions,
SUM(CASE WHEN Mission_Outcome LIKE 'Failure%' THEN 1 ELSE 0 END) AS Failed_Missions, COUNT(*) AS
Total_Missions FROM SPACEXTBL;
```

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

Successful_Missions	Failed_Missions	Total_Missions
100	1	101

Boosters Carried Maximum Payload

To list the names of the booster which have carried the maximum payload mass, we use the following magic SQL code below:

%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

```
 * 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

To list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015, we use the following magic SQL code below:

```
%sql SELECT SUBSTR(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE SUBSTR(Date, 0, 5) = '2015' AND Landing_Outcome LIKE 'Failure (drone ship)%'
```

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

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

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

To 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, we use the following magic SQL code:

%sql SELECT Landing_Outcome, COUNT(*) AS Count FROM SPACEXTBL GROUP BY Landing_Outcome ORDER BY Count DESC;

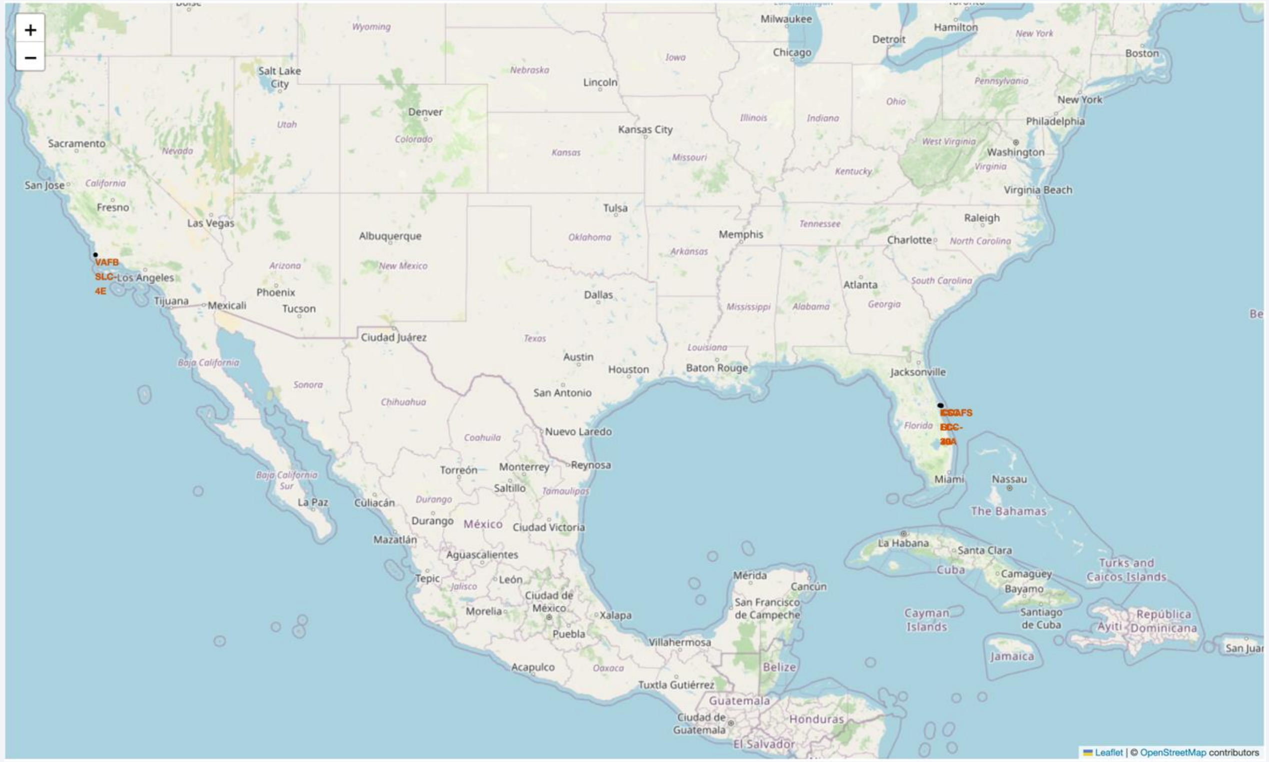
```
* sqlite:///my_data1.db
Done.
```

Landing_Outcome	Count
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

Launch Sites Proximities Analysis

Launch Sites



The map displays all launch site locations, highlighting two key geographical patterns:

- **Equatorial Positioning**: Every site is located near the Earth's equator.
- **Coastal Proximity**: All sites are situated immediately adjacent to coastlines.

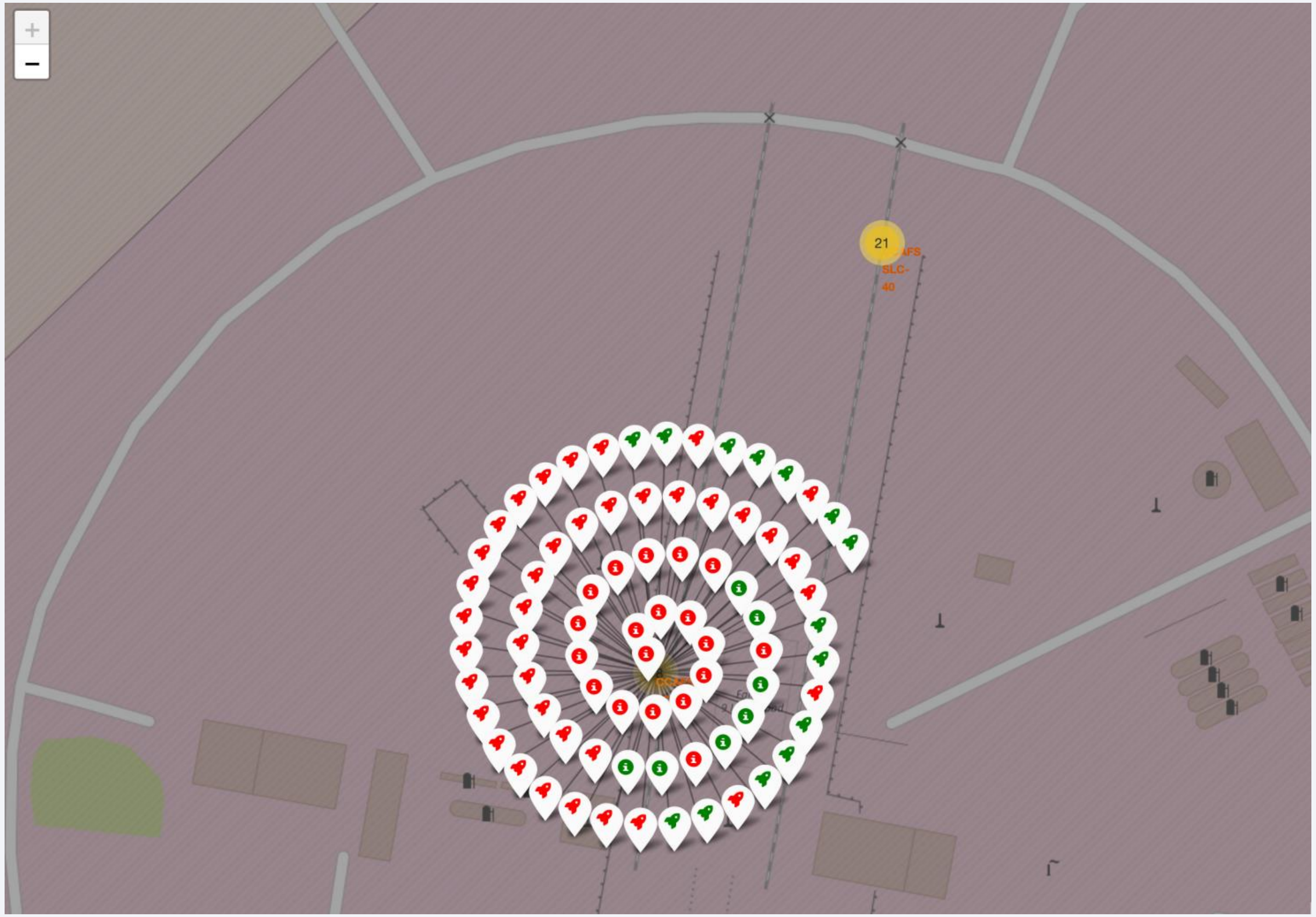
This strategic placement optimizes launch efficiency and mission safety.

Launch Result by Site

- **Color-Coded Markers**:
  - **Green** = Successful launch (class 1)
  - **Red** = Failed launch (class 0)
- **Marker Clustering**:
  - Groups overlapping markers at identical coordinates (since all launches occur at just 4 sites)
  - Prevents map clutter while preserving outcome trends

Purpose: Enables clear, interactive analysis of success/failure patterns by location.

Key Benefit: Maintains readability despite high launch density at shared coordinates.

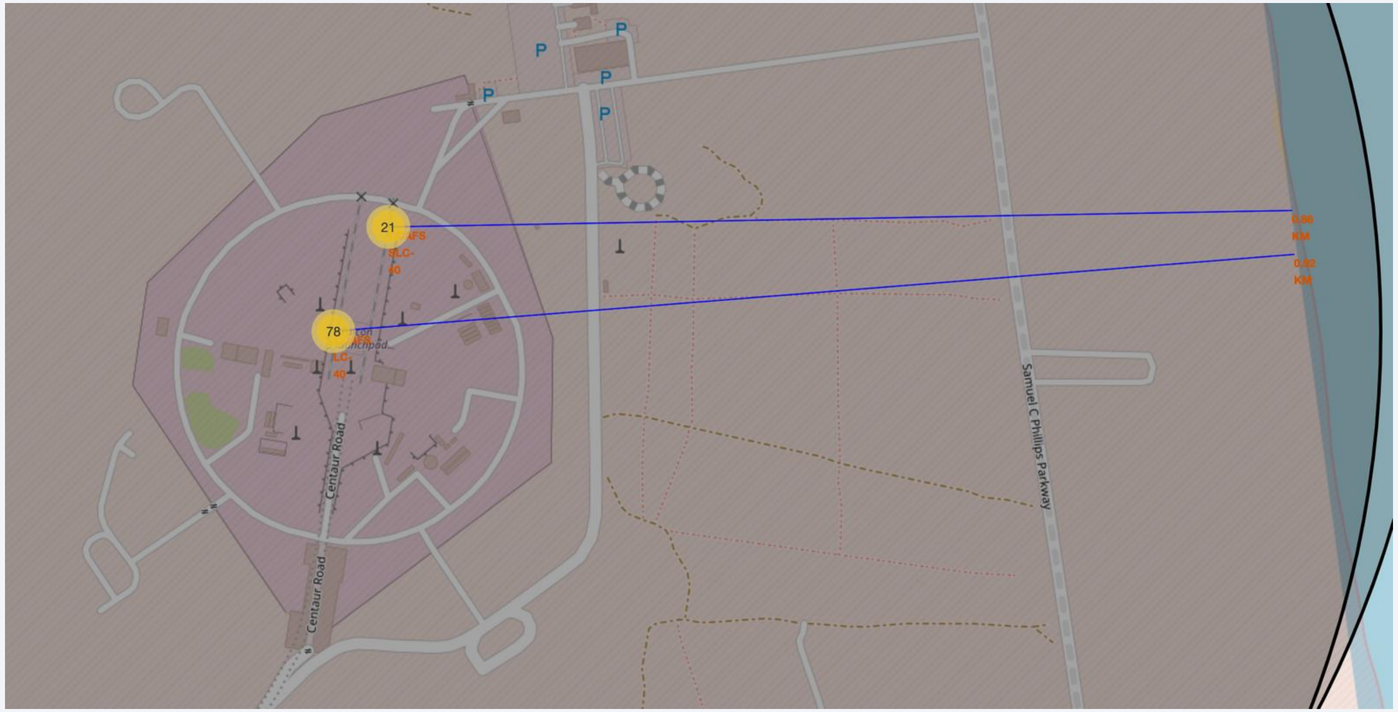


Launch Site Proximity to Railway, Highway, or Coastline

The interactive Folium map visualizes proximity measurements by displaying calculated distance lines connecting each launch site to nearby geographical features, including:

- Coastlines
- Highways
- Railroads

This spatial analysis highlights the logistical relationships between launch infrastructure and key transportation routes or natural boundaries.

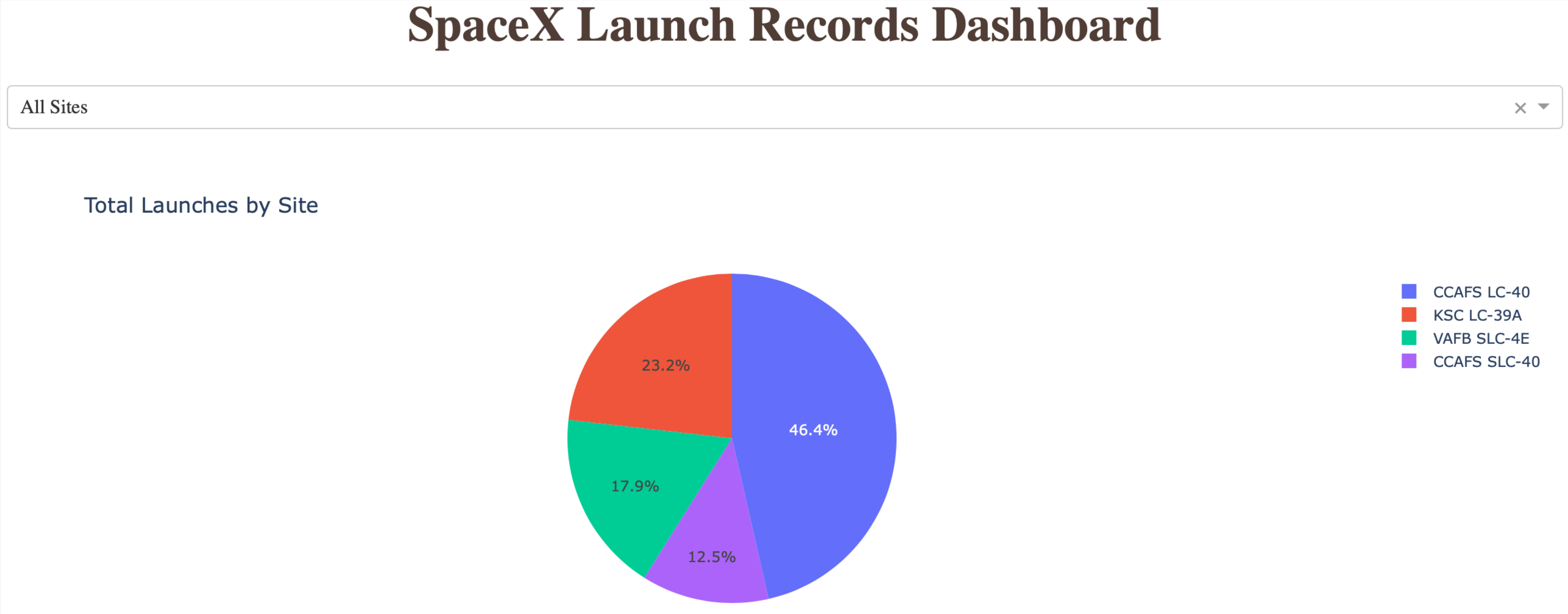


Section 4

Build a Dashboard with Plotly Dash

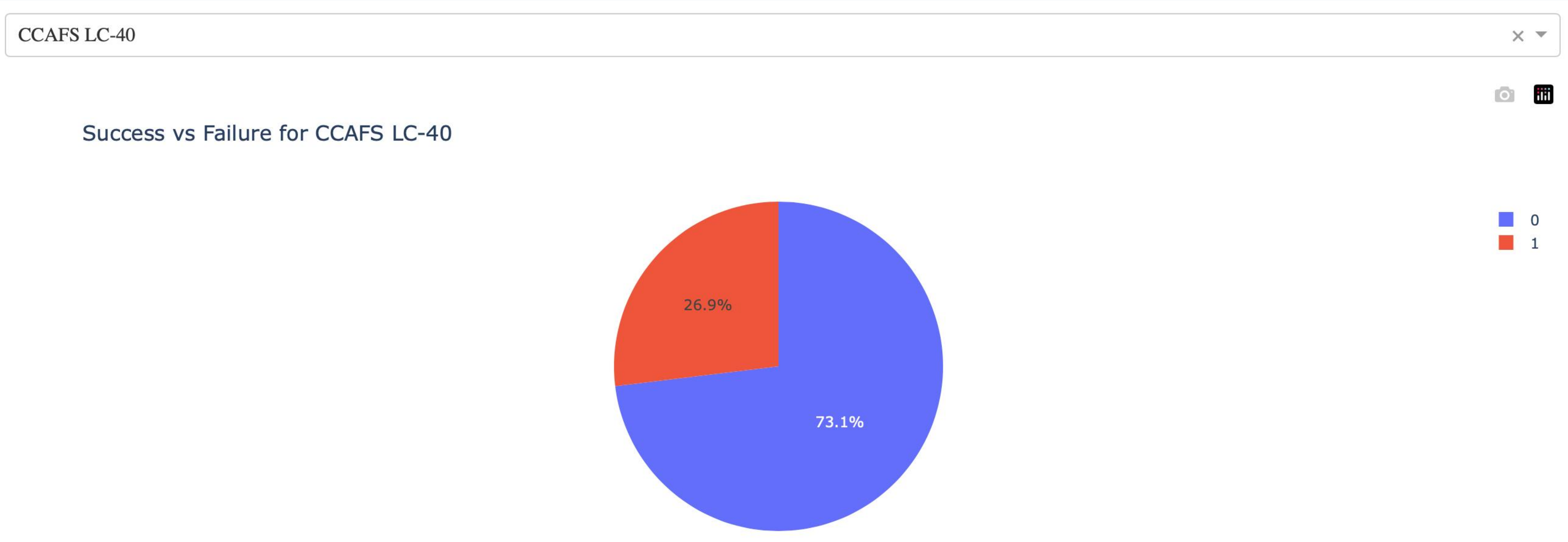
Successful Launch Records

The pie chart shows total successful launches for all sites. Site CCAFS LC-40 has the highest successful launches among the sites.



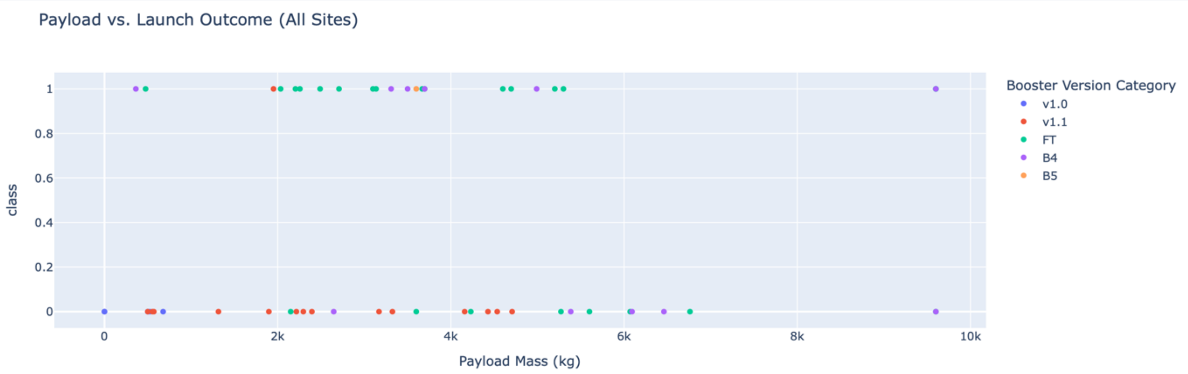
Launch Site With Highest Launch Success Ratio

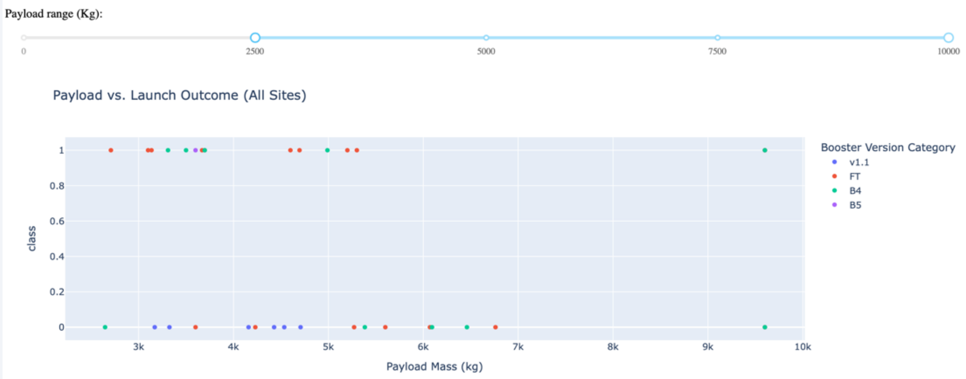
The pie chart shows the total launch outcome for site CCAFS LC-40. It has 73.1% successful, and 26.9% failed launches.

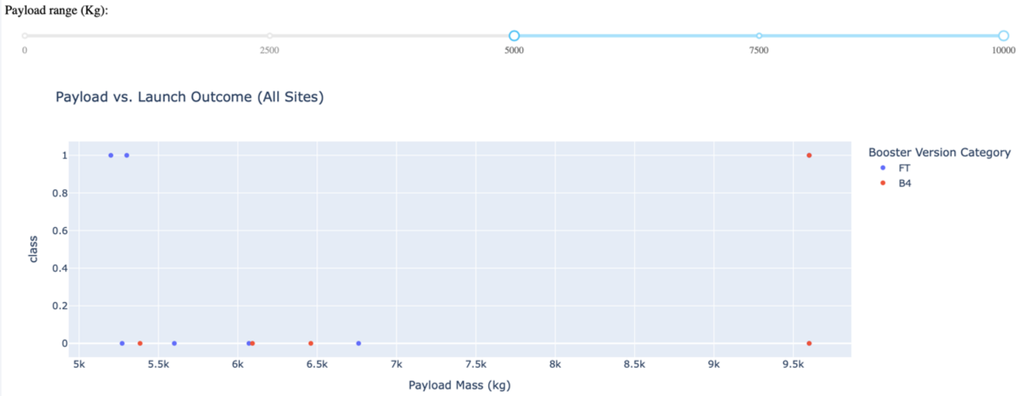


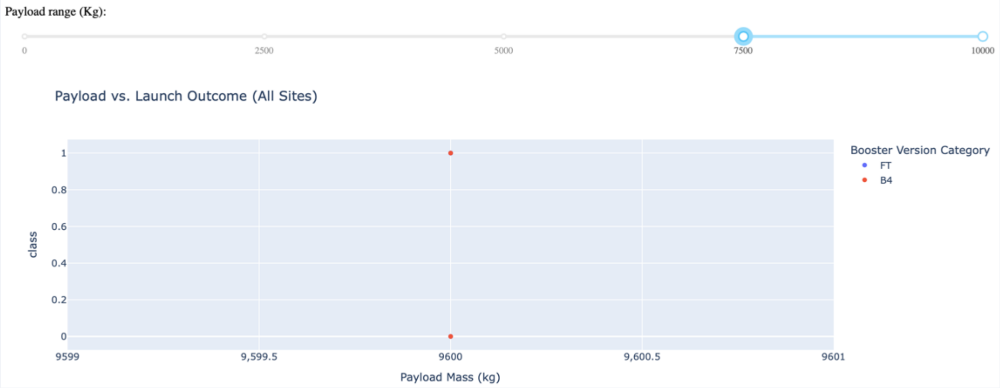
Payload VS Launch Outcome Scatter Plot

Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider. It can be seen that, Booster versions FT and B4 are the only ones that launched payloads within 7,500 and 10,000 kilograms









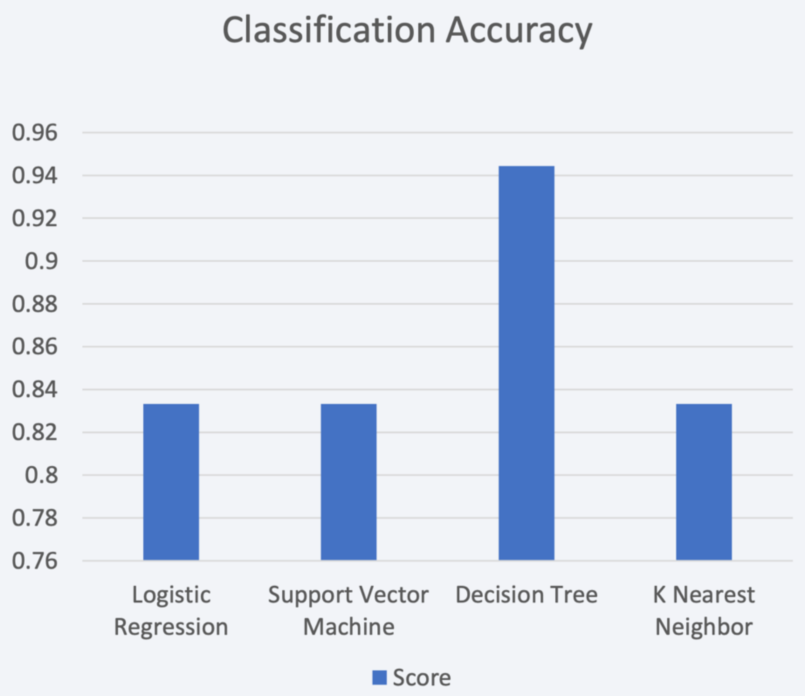
Section 5

Predictive Analysis (Classification)

Classification Accuracy

1. **Data Preparation**
   - Standardized features to ensure consistent scaling
   - Split data into training/testing sets for model validation
2. **Model Training & Optimization**
   - Tested four algorithms:
     - Logistic Regression
     - Support Vector Machines (SVM)
     - Decision Tree Classifier
     - K-Nearest Neighbors (KNN)
   - Performed **Grid Search** to identify optimal hyperparameters
3. **Performance Evaluation**
   - Used **confusion matrices** to assess accuracy
   - **Decision Tree outperformed others** with:
     - **Training accuracy: 94.4%**
     - **Testing accuracy: 82%**

	Algorithm	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.889286	0.944444
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333



Confusion Matrix

Optimal Hyperparameters

```
{
  'criterion': 'gini',
  'max_depth': 4,
  'max_features': 'sqrt',
  'min_samples_leaf': 2,
  'min_samples_split': 10,
  'splitter': 'best'
}
```

Accuracy Results

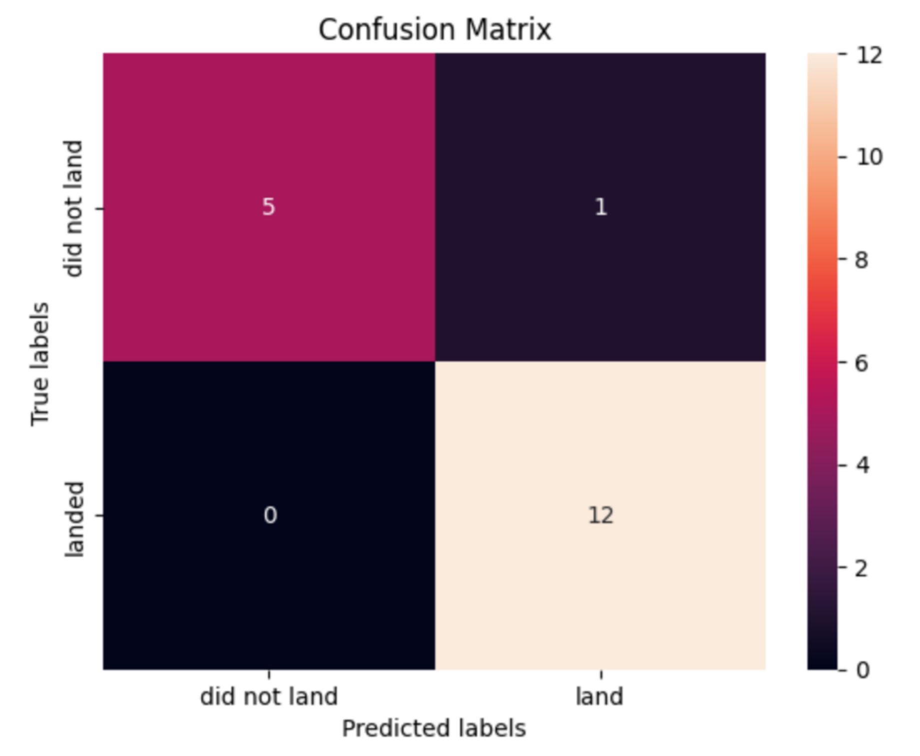
- **Validation Data**: 87.5%
- **Test Data**: **94.4%** (Highest among all tested models)
- **Correct Predictions**:
  - 5/6 "Did Not Land" (83.3% recall)
  - 12/12 "Landed" (100% precision)
- **Misclassifications**:
  - Only 1 false negative ("Did Not Land" → "Landed")

Key Takeaways

- **High Reliability**:
  - Near-perfect accuracy (94.4%) on unseen test data.
  - Zero false positives for landings (all "Landed" predictions correct).
- **Practical Utility**:
  - Model excels at identifying successful landings, critical for cost/reusability analysis.
- **Limitation**:
  - Slight under-detection of failures (1/6 missed).

Actionable Insight:

- Deploy this model for mission planning, with optional manual review for edge-case payloads/orbits.
- Consider cost-sensitive learning if false negatives (missed failures) are economically critical.



Conclusions

Data Collection & Preparation

- Gathered launch data via **APIs and web scraping**, followed by **cleaning and formatting** for analysis.

Key Insights from EDA

1. **Critical Factors Affecting Success**:
    - Flight number (experience)
    - Payload mass
    - Launch site location
    - Orbit type
2. **Success Rate Trends**:
    - **Steady increase from 2013 to 2020**, reflecting SpaceX's iterative improvements.
3. **Geographical Patterns**:
    - All launch sites are **near the equator** and **close to coastlines** for optimal efficiency.

Model Performance

- **Decision Tree** achieved the **highest accuracy (94.4%)** among tested models (Logistic Regression, SVM, KNN).
- **Confusion matrix** confirmed strong predictive power, especially for successful landings.

- **Conclusion**

The high-accuracy model enables **reliable predictions** of Falcon 9 first-stage landing success, supporting:

- Cost estimates for reusable launches
- Competitive bidding strategies (e.g., for SPACE Y)
- Mission planning and risk assessment

Final Takeaway: Data-driven insights and robust modeling validate SpaceX's reusability success while providing actionable intelligence for industry stakeholders.

Appendix

Kindly refer to the following GitHub repository below. It contains all the completed labs and related files.

https://github.com/husham35/ibm_data_science/tree/main/applied_data-science_capstone

Thank you!