Winning Space Race with Data Science

Tariq Aziz Rao
24th July, 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)

Executive Summary

- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

Project background and context

- SpaceX has significantly lowered the cost of orbital launches, primarily through the reusability of its **Falcon 9** first-stage boosters. While a typical launch by other providers costs upwards of **$165 million**, SpaceX offers launches for **$62 million,** a reduction largely due to booster recovery and reuse.
- Predicting whether a first-stage booster will successfully land is critical to estimating launch cost and operational efficiency. This project applies machine learning to publicly available SpaceX launch data to **predict the success of first-stage landings**.

Problem Statement

This project aims to answer the following key questions:

- What is the impact of variables such as payload mass, launch site, number of previous flights, and orbit type on first-stage landing success?
- Has the success rate of first-stage landings improved over time?
- Which binary classification algorithm (e.g., Logistic Regression, SVM, Decision Tree, KNN) provides the highest predictive accuracy for landing outcomes?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using SpaceX Rest API
  - Using Web Scrapping from Wikipedia
- Perform data wrangling
  - Filtering the data - Dealing with missing values
  - Using One Hot Encoding to prepare the data to a binary classification
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluation of classification models to ensure the best results

Data Collection

Data Collection Process:

To obtain a complete dataset for analysis, we combined data from two sources:

- **SpaceX REST API** – Provided structured technical data.
- **Wikipedia Web Scraping** – Supplied additional launch context not available via the API.
- This hybrid approach ensured full coverage of all relevant launch parameters.

Collected via the SpaceX REST API:

- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Scraped Wikipedia Fields

- Extracted from the launch history table on Wikipedia:
- Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

Request Launch Data
- Access SpaceX API endpoint
- Retrieve launch data using HTTP GET

Decode & Normalize JSON
- Parse response using .json()
- Convert to structured DataFrame using pandas.json_normalize()

Extract Relevant Launch Info
- Apply custom functions to select key fields
- Focus on mission name, payload, orbit, etc.

Structure as Dictionary
- Organize extracted data into a Python dict
- Prepare for DataFrame conversion

Convert to DataFrame
- Transform dictionary into a pandas DataFrame
- Validate schema and data types

Filter for Falcon 9 Launches
- Apply conditional filter: rocket name = Falcon 9

Handle Missing Values
- Replace NaNs in Payload Mass with mean value
- Use DataFrame.mean() for imputation

Decode & Normalize JSON
- Save cleaned DataFrame to CSV
- Ensure encoding and filename standards

GitHub URL: Data Collection API

Data Collection - Scraping

Send Request to Wikipedia
- Retrieve Falcon 9 launch data using HTTP GET

Parse HTML Content
- Create a BeautifulSoup object from the HTML response
- avigate the document tree

Extract Table Headers
- Locate the launch table
- Extract column names from the <thead> section

Parse Table Data
- Iterate over <tr> elements
- Clean and organize row-level data

Structure as Dictionary
- Store parsed content in a structured Python dict
- Align rows with column headers

Convert to DataFrame
- Create a pandas DataFrame from the dictionary
- Ensure correct formatting and data types

Export to CSV
- Save the final DataFrame to a .csv file
- Specify delimiter, encoding, and file name

GitHub URL: Data Collection with Web Scraping

Data Wrangling

The dataset contains various booster landing outcomes, categorized by method and success:

- **True/False Ocean** – Landing in the ocean
- **True/False RTLS** – Ground pad landing (Return to Launch Site)
- **True/False ASDS** – Drone ship landing (Autonomous Spaceport Drone Ship)

For model training, these outcomes were converted into binary labels:

- **1** → Successful landing
- **0** → Unsuccessful landing

GitHub URL: Data Wrangling

Perform exploratory Data Analysis and determine Training Labels

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column and finally export the data in CSV

EDA with Data Visualization

- Visualizations created include:
- **Scatter plots:** Examined relationships between variables such as Flight Number, Payload Mass, Launch Site, Orbit Type, and Success Rate to identify potential predictive features.
- **Bar charts:** Compared success rates across categorical variables like Launch Site and Orbit Type.
- **Line charts:** Tracked trends in success rates over time.
- These analyses guide feature selection and model development.

GitHub URL: EDA with Data Visualization

EDA with SQL

Performed SQL queries:

- Listed unique launch sites.
- Retrieved 5 launch sites starting with "CCA".
- Calculated total payload by NASA (CRS) launches.
- Found average payload for booster F9 v1.1.
- Identified first successful ground pad landing date.
- Listed boosters with successful drone ship landings and payload 4000–6000.
- Counted successful and failed missions.
- Found booster versions with max payload.
- Retrieved failed drone ship landings in 2015 with booster and launch site.
- Ranked landing outcome counts between 2010-06-04 and 2017-03-20

GitHub URL: EDA with SQL

Build an Interactive Map with Folium

Launch Site Mapping and Visualization

- Added a marker with a circle, popup, and text label at NASA Johnson Space Center using its latitude and longitude as the initial map location.
- Plotted markers with circles, popups, and text labels for all launch sites, illustrating their geographic locations relative to the Equator and nearby coastlines.
- Used colored markers clustered by launch out come green for successful and red for failed launches to highlight launch site success rates.
- Drew colored lines from KSC LC-39A to nearby landmarks such as railway, highway, coastline, and closest city to visualize proximity distances.

GitHub URL: Interactive Map with Folium

Build a Dashboard with Plotly Dash

Launch Sites Dropdown: Enables selection of a specific launch site for focused analysis.

Success Launches Pie Chart: Displays total successful launches for all sites or success vs. failure counts for the selected site.

Payload Mass Slider: Allows filtering data by a customizable payload mass range.

Payload vs. Success Scatter Chart: Visualizes the correlation between payload mass and launch success across different booster versions.

GitHub URL: Dashboard with Plotly Dash

Predictive Analysis (Classification)

Prepare Labels
- Extract target variable ('Class') as NumPy array

Standardize Features
- Use StandardScaler to fit and transform feature data

Split Dataset
- Use train_test_split to divide into training & test sets

Define Hyperparameter Search
- Create GridSearchCV object (cv=10) for cross-validation

Train Multiple Models
- Apply GridSearchCV on:
  - Logistic Regression
  - SVM
  - Decision Tree
  - KNN

Evaluate Accuracy
- Use .score() method to assess model performance on test set

Analyze Confusion Matrix
- Generate confusion matrix for all models

Compare Metrics
- Use Jaccard Index and F1 Score to identify best model

GitHub URL:
Predictive Analysis (Classification)

Results

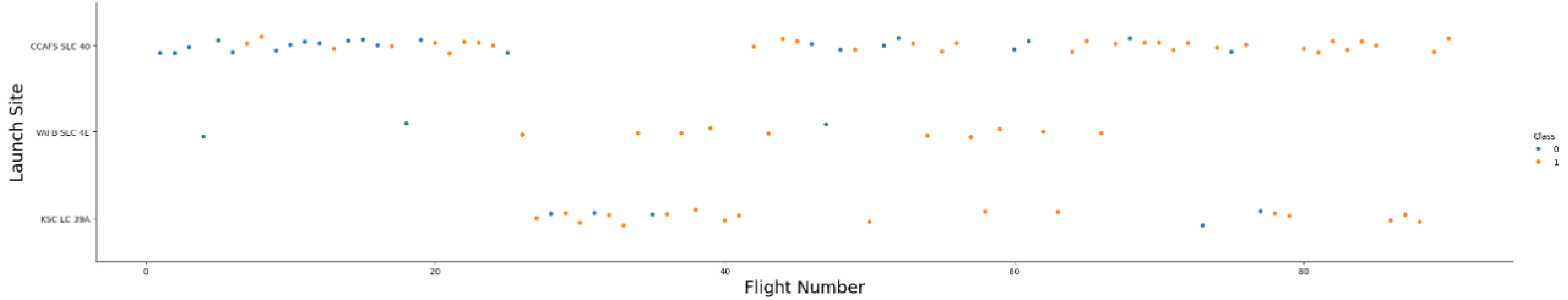
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

```
[7]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x='FlightNumber', y='LaunchSite', hue='Class', data=df, aspect=5)
plt.xlabel('Flight Number', fontsize=20)
plt.ylabel('Launch Site', fontsize=20)
plt.show()
```

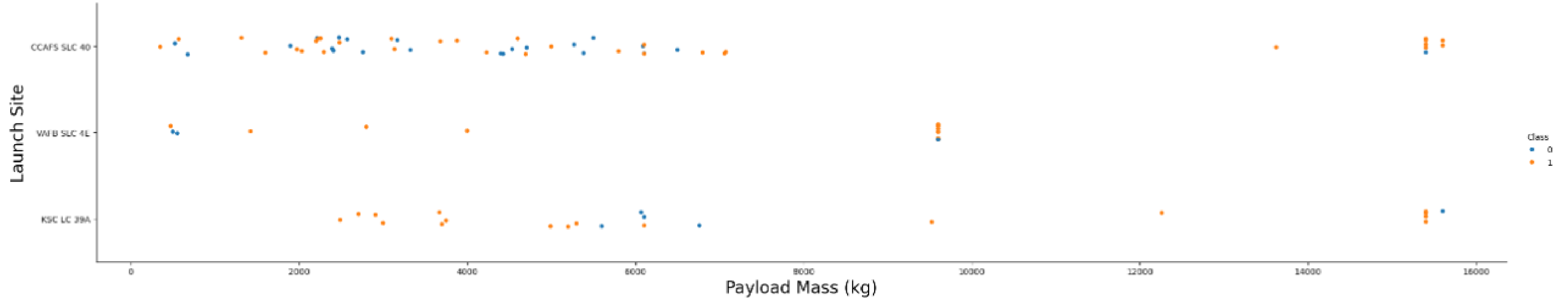


Explanation:

- Early Falcon 9 flights experienced failures, while recent launches have achieved consistent success.
- The CCAFS SLC-40 launch site accounts for approximately 50% of all launches.
- Launch sites VAFB SLC-4E and KSC LC-39A demonstrate higher success rates.
- There is a clear trend indicating increasing success rates with each subsequent launch.

Payload vs. Launch Site

```
[8]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x='PayloadMass', y='LaunchSite', hue='Class', data=df, aspect = 5)
plt.xlabel('Payload Mass (kg)',fontsize=20)
plt.ylabel('Launch Site',fontsize=20)
plt.show()
```



Explanation:

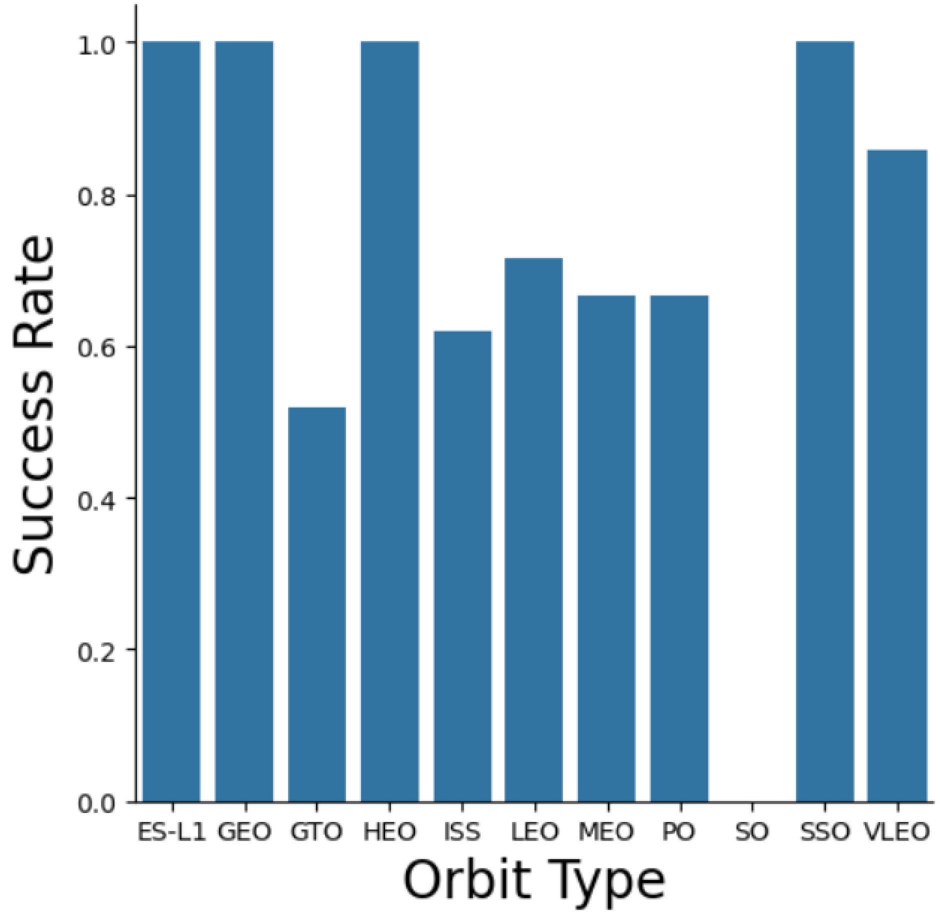
- Across all launch sites, higher payload mass is generally associated with higher success rates.
- Most launches carrying payloads over **7000 kg** were successful.
- **KSC LC-39A** achieved a **100% success rate** for launches with payloads under **5500 kg**.

Success Rate vs. Orbit Type

Explanation:

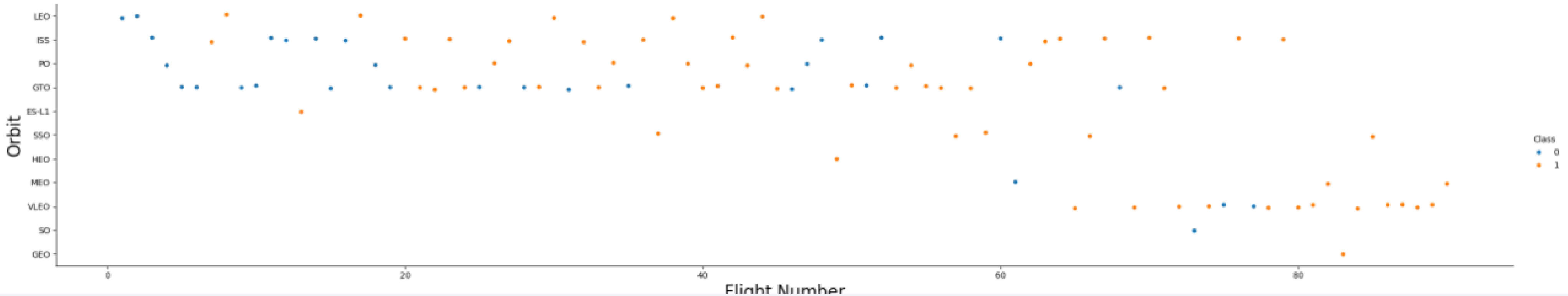
- 100% Success Rate:
  - ES-L1, GEO, HEO, SSO
- 0% Success Rate:
  - SO
- Moderate Success Rate (50%–85%):
  - GTO, ISS, LEO, MEO, PO

```
[9]: # HINT use groupby method on Orbit column and get the mean of Class column
sns.catplot(x= 'Orbit', y = 'Class', data = df.groupby('Orbit')['Class'].mean().reset_index(), kind = 'bar')
plt.xlabel('Orbit Type',fontsize=20)
plt.ylabel('Success Rate',fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

```
[10]: # Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x = 'FlightNumber', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Flight Number', fontsize = 20)
plt.ylabel('Orbit', fontsize = 20)
plt.show()
```



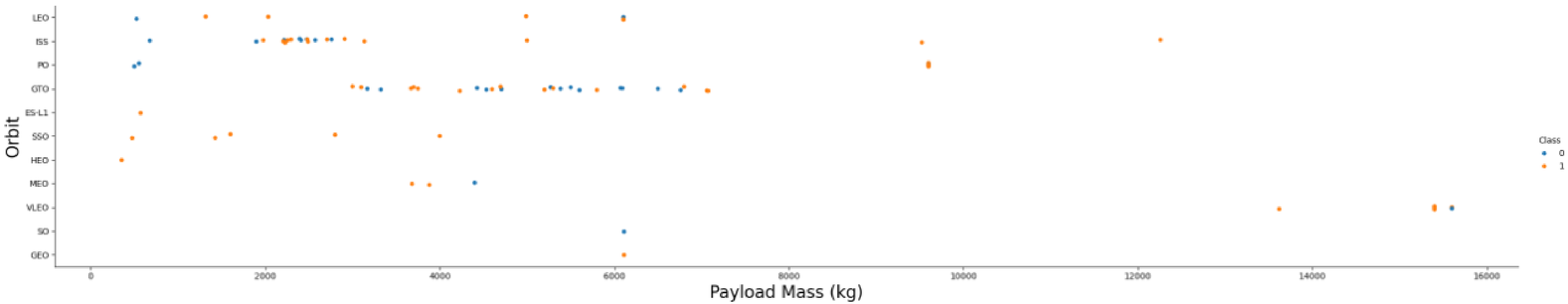
Explanation:

- In **LEO (Low Earth Orbit)**, launch success appears to improve with an increasing number of flights, indicating a positive correlation with experience.
- In contrast, for **GTO (Geostationary Transfer Orbit)**, there is no clear relationship between flight number and success rate.

Payload vs. Orbit Type

[11]:

```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(x = 'PayloadMass', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Payload Mass (kg)', fontsize = 20)
plt.ylabel('Orbit', fontsize = 20)
plt.show()
```



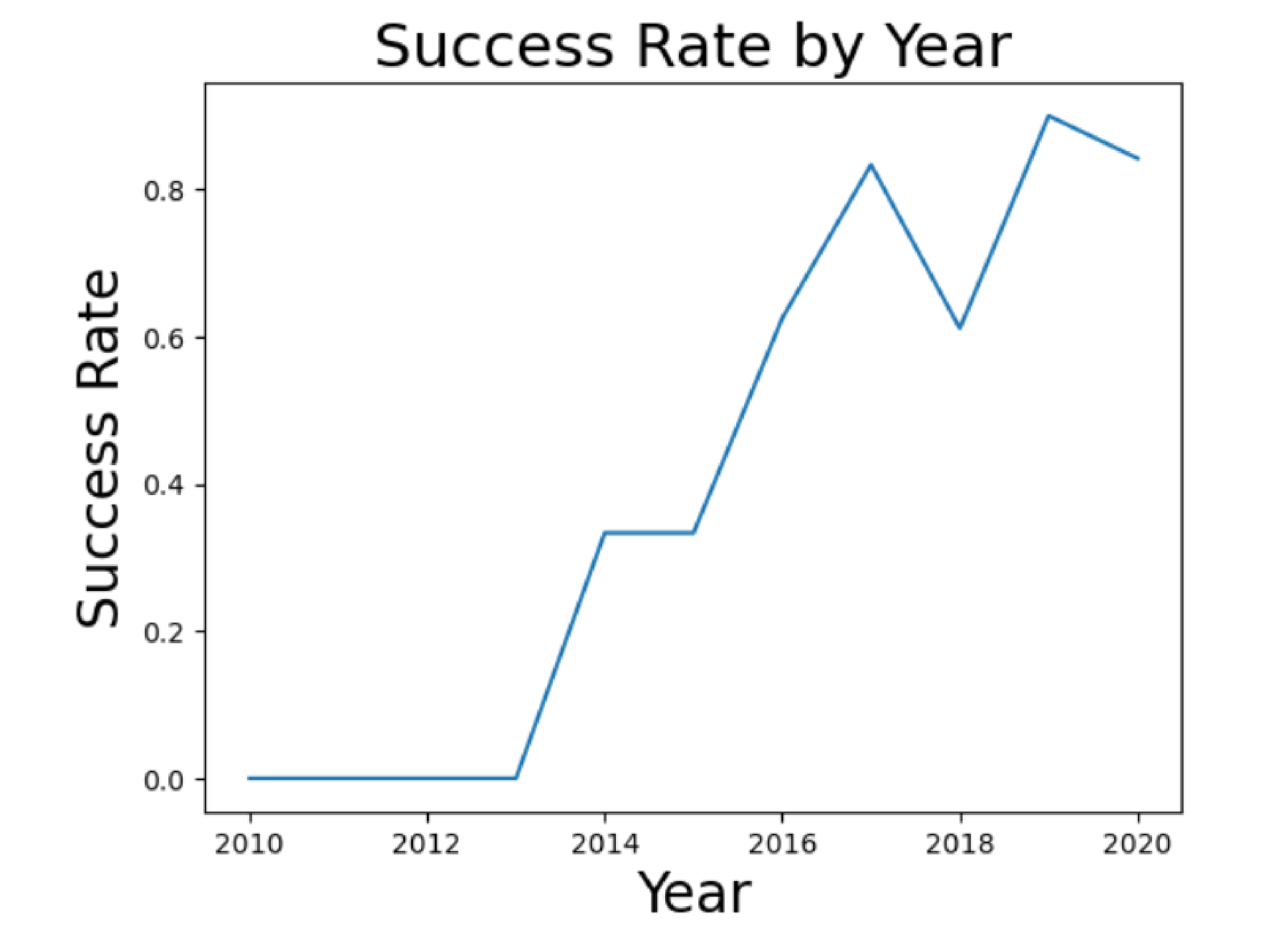
Explanation:

- Heavy payloads negatively impact success rates for **GTO** orbits.
- Conversely, heavy payloads have a positive effect on success in **Polar** and **LEO (ISS)** orbits.

Launch Success Yearly Trend

Explanation:

- The success rate since 2013 kept increasing till 2020



All Launch Site Names

Display the names of the unique launch sites in the space mission

```
[13]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;

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

[13]:

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

Explanation:

- Displaying the names of the unique launch sites in the space mission

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

[24]:
```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

[24]:

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

Explanation:

- Displaying 5 records where launch sites begin with the string 'CCA'.

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

[15]:
```
%sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

[15]:

Total_Payload_Mass
45596

Explanation:

- Displaying the total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
[16]: %sql SELECT AVG("Payload_Mass__kg_") AS Avg_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';

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

[16]:

Avg_Payload_Mass
2928.4

Explanation:

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

First Successful Ground Landing Date

```
[17]: %sql SELECT MIN("Date") AS First_Successful_Ground_Pad_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';

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

[17]:

First_Successful_Ground_Pad_Landing
2015-12-22

Explanation:

- Listing the date when the first successful landing outcome in ground pad was achieved

Successful Drone Ship Landing with Payload between 4000 and 6000

Explanation:

- List the names of boosters that have successfully landed on a drone ship and carried payloads between 4000 kg and 6000 kg.

```
[18]: %sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Mission_Outcome" = 'Success' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;

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

[18]:

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[26]:
```
%sql SELECT "Mission_Outcome", COUNT(*) AS total_number FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

[26]:

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

Explanation:

- Listing the total number of successful and failure mission outcomesa

Boosters Carried Maximum Payload

Explanation:

- Listing the names of the booster versions which have carried the maximum payload mass

```
[21]: %sql SELECT "Booster_Version", "Payload_Mass" FROM SPACEXTABLE WHERE "Payload_Mass" = (SELECT MAX("Payload_Mass") FROM SPACEXTABLE);
```

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

[21]:

Booster_Version	"Payload_Mass"
F9 v1.0 B0003	Payload_Mass
F9 v1.0 B0004	Payload_Mass
F9 v1.0 B0005	Payload_Mass
F9 v1.0 B0006	Payload_Mass
F9 v1.0 B0007	Payload_Mass
F9 v1.1 B1003	Payload_Mass
F9 v1.1	Payload_Mass
F9 v1.1	Payload_Mass
F9 v1.1	Payload_Mass
F9 v1.1	Payload_Mass
F9 v1.1	Payload_Mass
F9 v1.1 B1011	Payload_Mass
F9 v1.1 B1010	Payload_Mass
F9 v1.1 B1012	Payload_Mass
F9 v1.1 B1013	Payload_Mass
F9 v1.1 B1014	Payload_Mass
F9 v1.1 B1015	Payload_Mass

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[42]:
```
%sql SELECT strftime('%m', Date) AS month_number, strftime('%Y-%m-%d', Date) AS Date, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landin
```

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

[42]:

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

Explanation:

- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015

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.

```
[46]: %sql SELECT Landing_Outcome, COUNT(*) AS Landing_Outcomes FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Lan
```

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

[46]:

Landing_Outcome	Landing_Outcomes
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

Explanation:

- Rank landing outcomes (e.g., Failure on drone ship, Success on ground pad) by their count between **2010-06-04** and **2017-03-20**, sorted in descending order.

Section 3

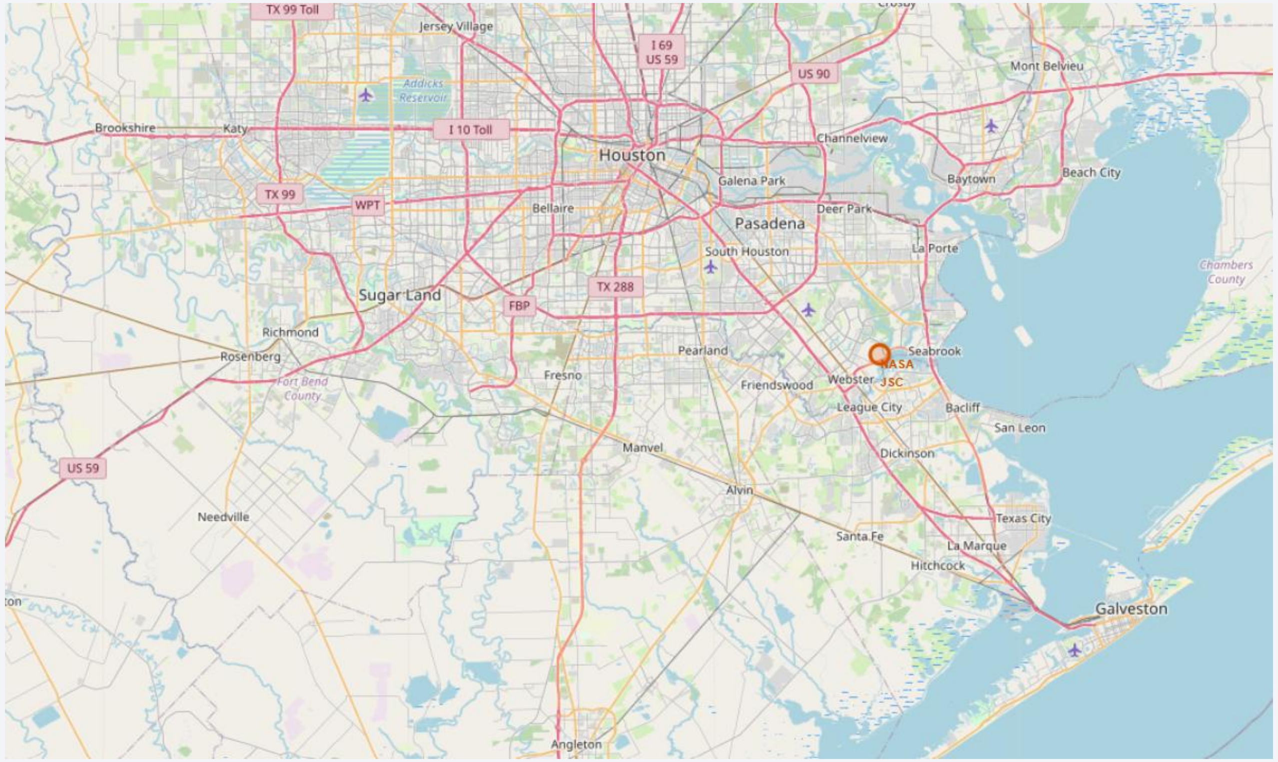
Launch Sites Proximities Analysis

Map Screenshot All launch sites' location markers on global map

Explanation:

Most launch sites are located near the **Equator**, where the Earth's surface rotates at approximately **1670 km/h**—the fastest rotational speed on the planet. Launching from the equator provides rockets with an initial velocity boost due to inertia, aiding in achieving and maintaining orbital speed efficiently.

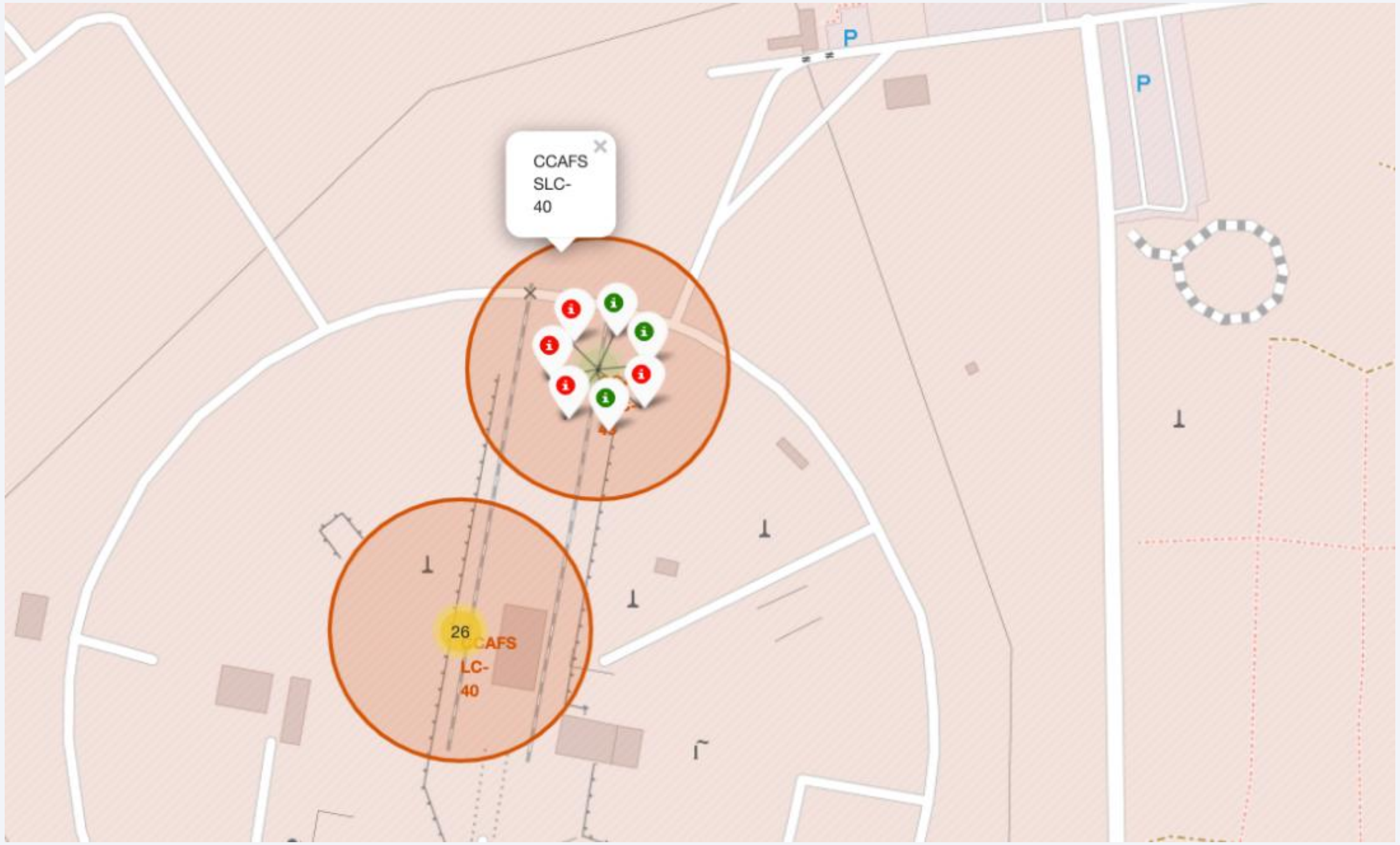
- Additionally, all launch sites are situated close to the **coastline**. Launching rockets over the ocean minimizes the risk of falling debris or accidents impacting populated areas, enhancing safety for surrounding communities



Color-labeled launch record on the map

Explanation:

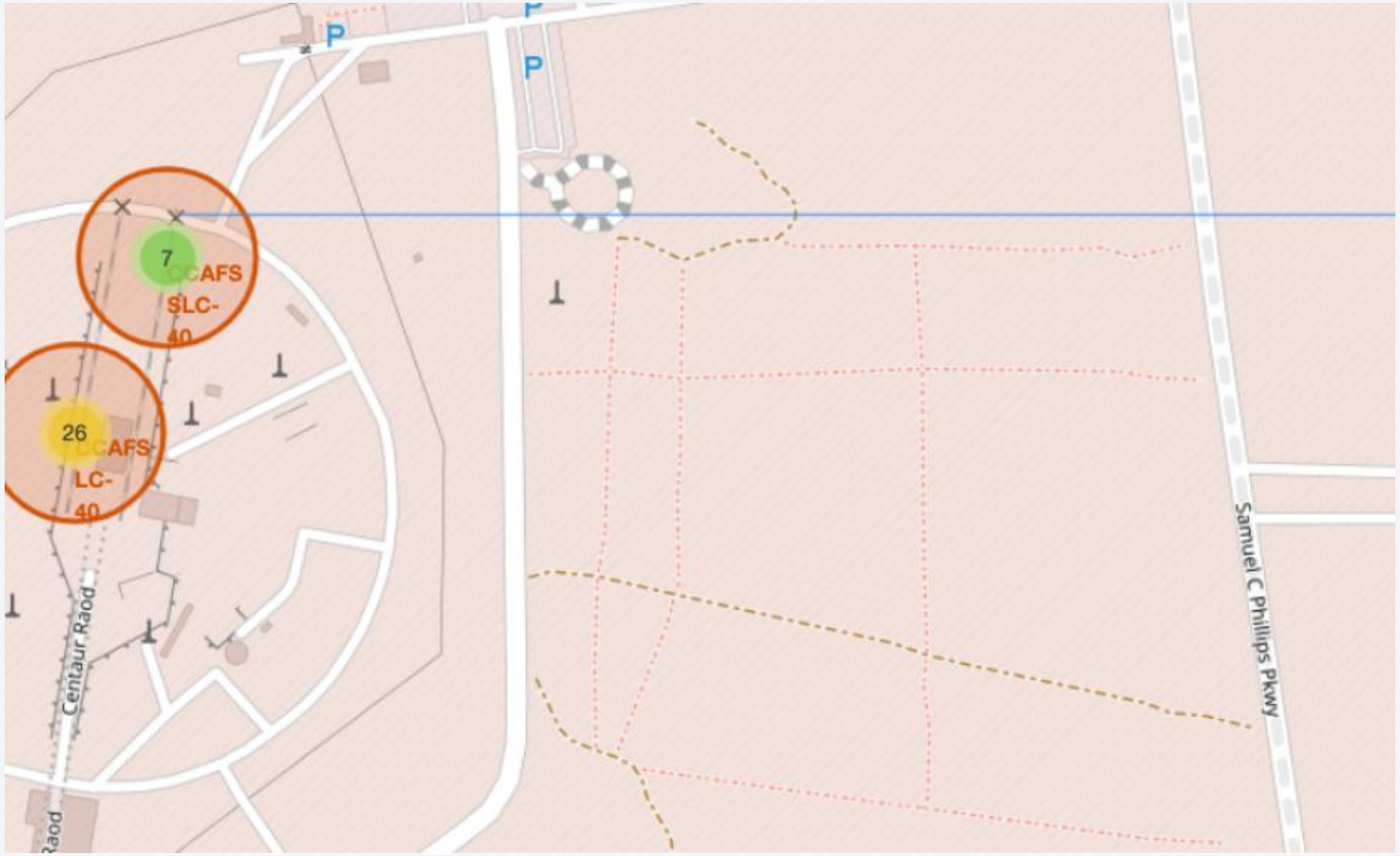
- Color-coded markers clearly indicate launch site success rates:
  - **Green:** Successful launch
  - **Red:** Failed launch
- The launch site **KSC LC-39A** notably demonstrates a very high success rate.



Distance from the launch site KSC LC-39A to its proximities

Explanation:

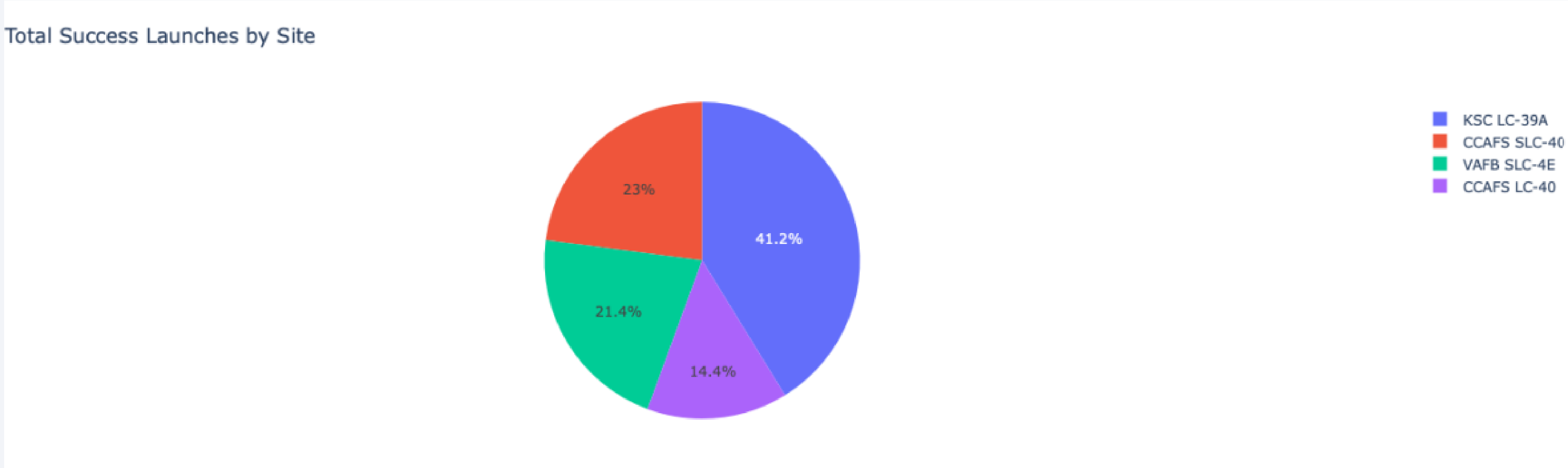
- Visual analysis shows that KSC LC-39A is located relatively close to key infrastructures and populated areas:
  - Railway: **15.23 km**
  - Highway: **20.28 km**
  - Coastline: **14.99 km**
  - Closest city (Titusville): **16.32 km**
- Given that a failed rocket traveling at high speeds can cover 15–20 km within seconds, proximity to these areas presents potential safety risks to nearby populations and infrastructure.



Section 4

Build a Dashboard with Plotly Dash

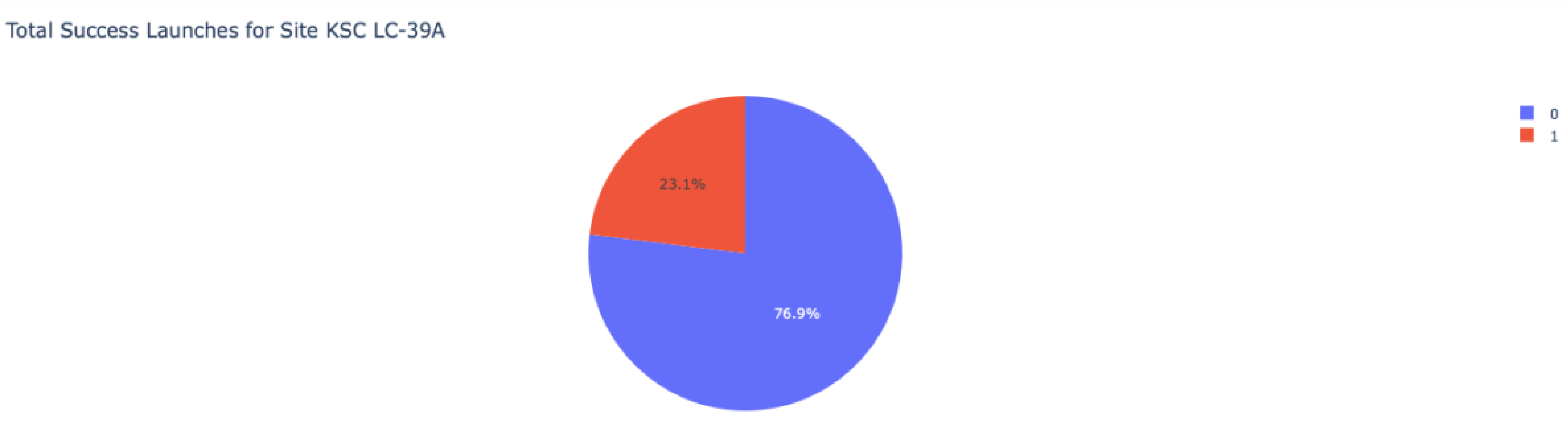
Launch success count for all sites



Explanation:

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

Launch site with highest launch success ratio

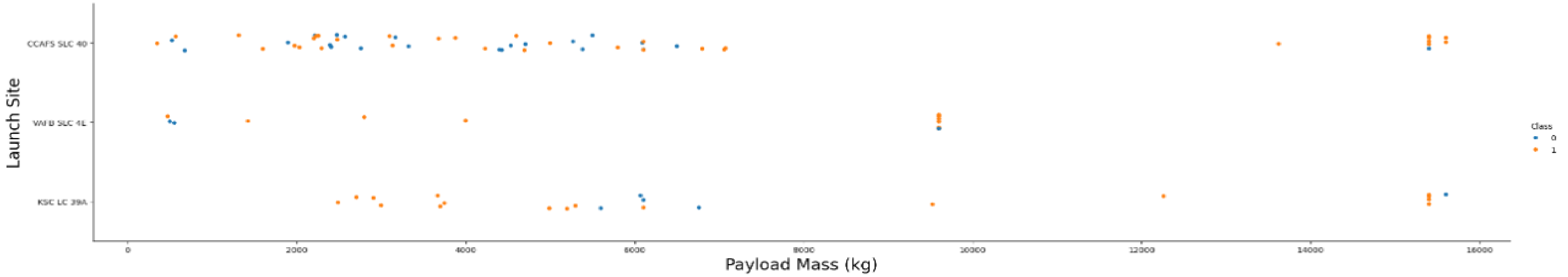


Explanation:

- KSC LC-39A exhibits the highest launch success rate at **76.9%**, with **10 successful** landings and only **3 failures**.

Payload Mass Vs. Launch Outcome for all sites

[8]:
```python
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x='PayloadMass', y='LaunchSite', hue='Class', data=df, aspect = 5)
plt.xlabel('Payload Mass (kg)',fontsize=20)
plt.ylabel('Launch Site',fontsize=20)
plt.show()
```



Explanation:

- The charts show that payloads between 2000 and 5500 kg have the highest success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy

[58]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.833333	0.819444
F1_Score	0.909091	0.916031	0.909091	0.900763
Accuracy	0.866667	0.877778	0.866667	0.855556

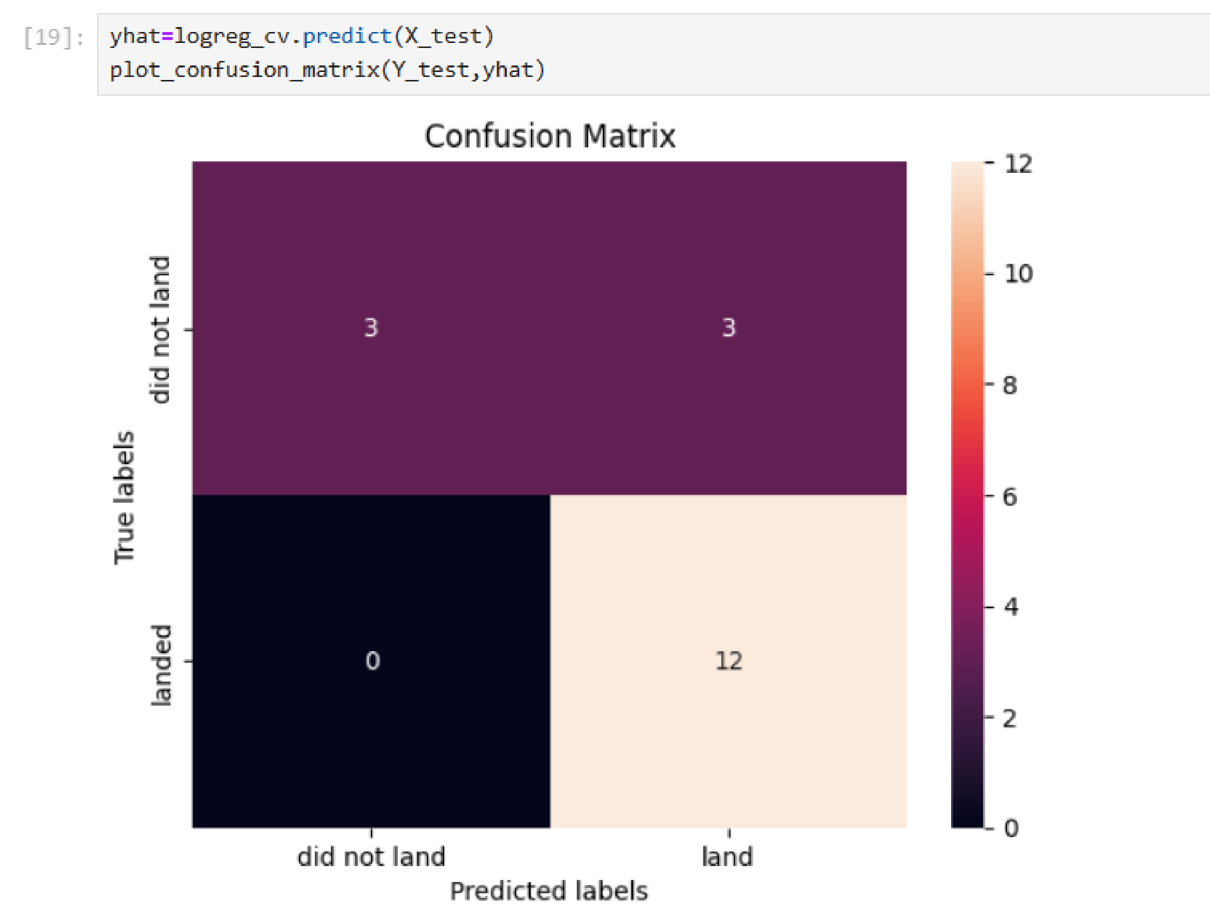
Explanation:

- Evaluation on the full dataset indicates that the **SVM model** outperforms others, achieving the highest accuracy and superior overall scores.

Confusion Matrix

Explanation:

- The confusion matrix shows that logistic regression effectively distinguishes between classes. However, the primary issue observed is a higher rate of **false positives**.



Conclusions

- The **SVM** is the best-performing algorithm for this dataset.
- Launches with lower payload mass tend to have higher success rates compared to heavier payloads.
- Most launch sites are located near the **Equator** and close to the **coastline**.
- Launch success rates have improved steadily over the years.
- **KSC LC-39A** records the highest success rate among all launch sites.
- Orbits **ES-L1**, **GEO**, **HEO**, and **SSO** show a **100% success rate**.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!