IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Neila Chettaoui

08/03/2024

# Outline

# Executive Summary

## Summary of Methodologies

The project aims to identify the key factors contributing to a successful rocket landing. To achieve this goal, various methodologies were employed:

- **Data collection** via the SpaceX REST API and web scraping techniques.
- **Data wrangling** to establish a success/failure outcome variable.
- **Exploratory data analysis** via data visualization techniques, focusing on factors such as payload, launch site, flight number, and yearly trends.
- **SQL analysis** to calculate statistics, including total payload, payload range for successful launches, and the overall count of successful and failed outcomes.
- **Investigation** into launch site success rates considered proximity to geographical markers.
- **Visualization** techniques to highlight launch sites with the highest success rates and successful payload ranges.
- **Predictive models** (i.e., logistic regression, support vector machine (SVM), decision tree, and K-nearest neighbor) to forecast landing outcomes.

# Executive Summary

## Summary of Results

**Exploratory Data Analysis** revealed the following key findings:

- Success rates for rocket launches have shown improvement over time.
- KSC LC-39A emerged with the highest success rate among all landing sites.
- Orbits ES-L1, GEO, HEO, and SSO displayed a 100% success rate.

**Visualization/Analytics** pointed out that:

- Most launch sites are near the equator, and all are close to the coast.

**Predictive Analytics** revealed that:

- All models performed similarly on the test set. The decision tree model slightly outperformed.

# Introduction

## Project background and context

SpaceX, a prominent player in the space exploration sector, is dedicated to democratizing space travel by making it economically accessible to a broader audience. The company has achieved significant milestones, such as deploying spacecraft to the international space station, establishing a satellite constellation for global internet coverage, and executing manned missions into space. A key factor enabling SpaceX's cost-effectiveness is the innovative reuse of the first stage of its Falcon 9 rocket, resulting in a relatively low launch cost of $62 million. In contrast, competitors lacking this reusable capability incur substantially higher costs, reaching upwards of $165 million per launch. The viability of reusing the first stage is pivotal in determining the overall launch cost, and this can be assessed by predicting the likelihood of a successful first-stage landing using machine learning models and publicly available data.

**Exploration Objectives**

- Investigate the impact of payload mass, launch site, number of flights, and orbital considerations on the success of the first-stage landing.
- Analyze the temporal trends in the rate of successful landings over time.
- Identify the most effective predictive model for assessing the success of the first-stage landing using binary classification.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data was collected using SapceX API and web-scraping from Wikipedia.
- Perform data wrangling
  - Data was processed via one-hot encoding on categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Four classification models were used: logistic regression, SVM, decision tree, and KNN,
  - GridSearch and cross-validation were adopted to tune hyper parameters,
  - The accuracy was calculated via score method and the confusion matrix was plotted.

# Data Collection

The process of gathering data comprised the following steps:

## SpaceX API

- Retrieval of data involved the submission of a GET request to the SpaceX API.
- The obtained data was decoded as JSON using .json() and subsequently transformed into a Data Frame using .json normalize().
- A cleaning procedure was applied, identifying and addressing missing values as required.

## Web Scraping

- Data related to Falcon 9 launch records was acquired through web scraping from Wikipedia.
- An HTTP GET request was dispatched to the HTML page dedicated to Falcon 9 launches.
- Utilizing BeautifulSoup, the data was parsed and organized into a Data Frame for further analysis.

# Data Collection – SpaceX API

- A GET request was sent to the SpaceX API.
- The data was decoded using .json() and converted to Data Frame using json_normalize()

```
[ ] static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[ ] response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[ ] # Use json_normalize meethod to convert the json result into a dataframe
    data = pd.json_normalize(response.json())
```



GitHub URL

# Data Collection - Scraping

- An HTTP GET request was dispatched to the Falcon 9 Launch HTML page.
- The retrieved data was then parsed and compiled into a Data Frame with the help of BeautifulSoup.

[GitHub URL]

## TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[ ] # use requests.get() method with the provided static_url
    # assign the response to a object
    html_data = requests.get(static_url)
    html_data.status_code

    200
```

Create a `BeautifulSoup` object from the HTML `response`

```
[ ] # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
    soup = BeautifulSoup(html_data.text)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[ ] # Use soup.title attribute
    soup.title

    <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

## TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

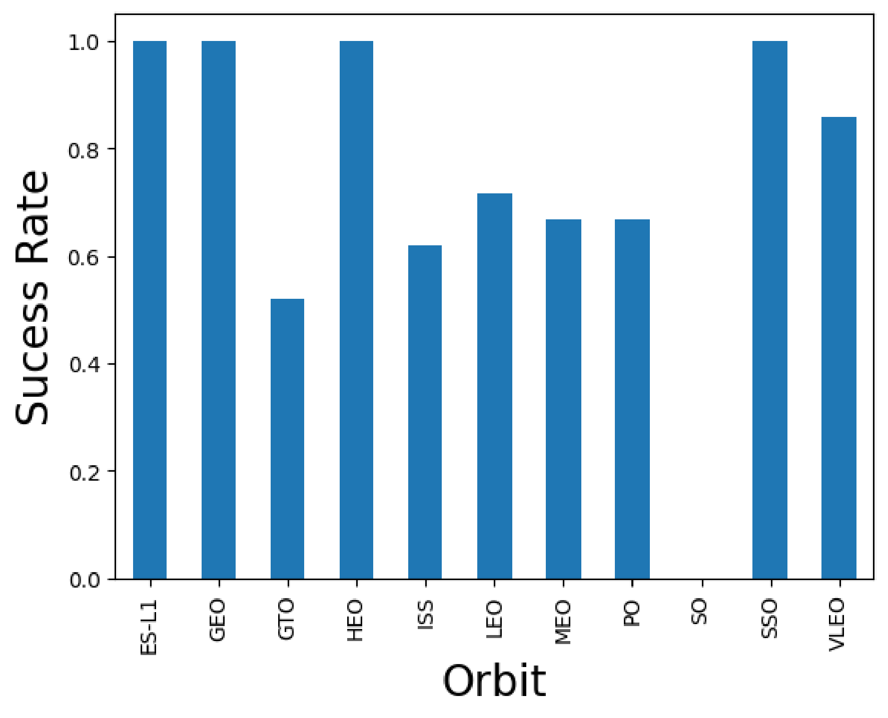
```
[ ] # Use the find_all function in the BeautifulSoup object, with element type `table`
    # Assign the result to a list called `html_tables`
    html_tables = soup.find_all('table')
```

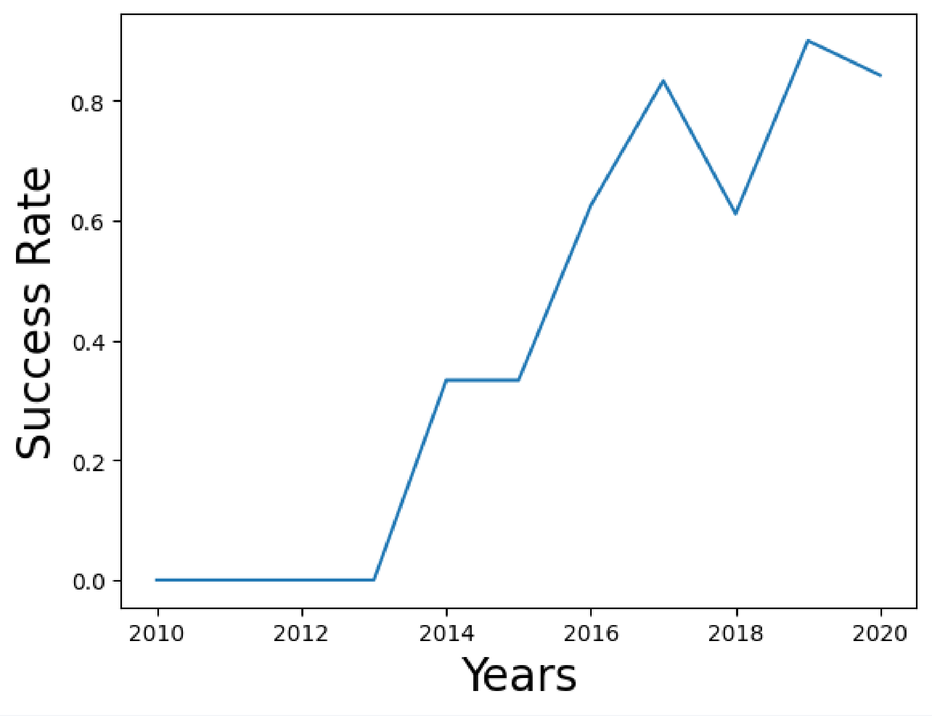
# Data Wrangling

- Determined the frequency of launches at each site, cataloged the number and types of orbits, and identified the variety and quantity of landing outcomes.
- Developed a label for landing outcomes based on the data from the Outcome column.
- GitHub URL

# EDA with Data Visualization

- The association between various factors was graphically represented, including flight number and launch site, payload versus launch site, success rate by orbit, total flights per orbit, the relationship between payload mass and orbit, and the yearly trend of launch successes.




[GitHub URL]

# EDA with SQL

The SQL table was imported into Jupyter Notebook, where various SQL queries were executed to extract insights:

- Identification of distinct launch sites involved in the space mission.
- Calculation of the cumulative payload mass transported by boosters under NASA (CRS) missions.
- Determination of the mean payload mass transported by the booster version F9 v1.1.
- Enumeration of mission outcomes, categorizing them into successes and failures.
- Analysis of unsuccessful drone ship landing attempts, including details on the booster version and names of the launch sites.



GitHub URL

# Build an Interactive Map with Folium

- Marked all launch sites on the Folium map, incorporating map objects such as markers, circles, and lines to visually represent the success or failure of each launch.
- Assigned launch outcomes, categorized as failure (class 0) and success (class 1), to facilitate visual identification.
- Leveraged color-labeled marker clusters to discern launch sites with notably high success rates.
- Conducted distance calculations between launch sites and nearby features, including railways, highways, and cities, enhancing the spatial analysis of the launch infrastructure.



GitHub URL

# Build a Dashboard with Plotly Dash

- Development of a dynamic and engaging interactive dashboard using Plotly Dash.
- Creation of visually informative pie charts, illustrating the cumulative launches from specific sites.
- Generation of a comprehensive scatter plot depicting the correlation between Outcome and Payload Mass (Kg) across various booster versions

GitHub URL

# Predictive Analysis (Classification)

- The dataset was divided into distinct training and testing sets to facilitate model evaluation.
- Various machine-learning models were constructed, and diverse hyperparameters were fine-tuned employing GridSearchCV.
- Accuracy served as the primary metric for model assessment, and enhancements were implemented through both feature engineering and hyperparameter tuning.
- The identification of the optimal-performing classification model was achieved through systematic evaluation and comparison.

GitHub URL

# Results

## Exploratory data analysis results

- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate

## Interactive analytics demo in screenshots

- Most launch sites are near the equator, and all are close to the coast
- Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities.

## Predictive analysis results

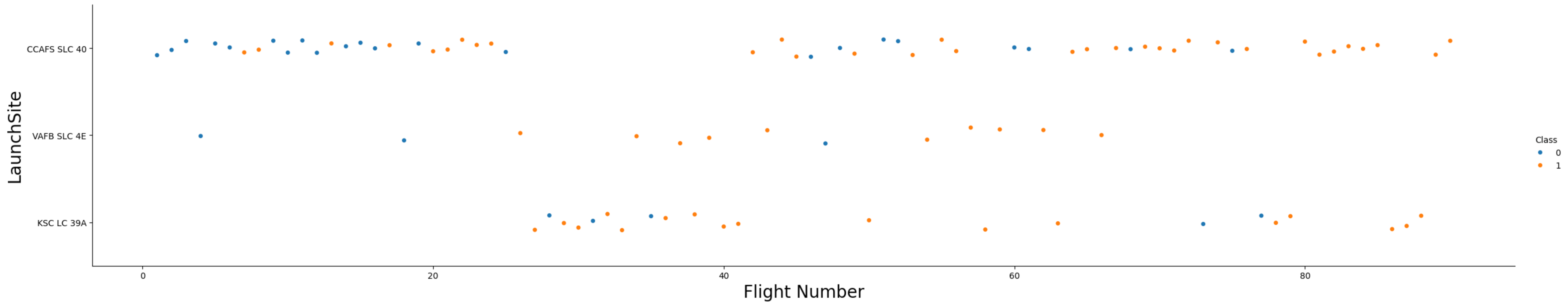
- Decision Tree model is the best predictive model for the dataset

Section 2

# Insights drawn from EDA

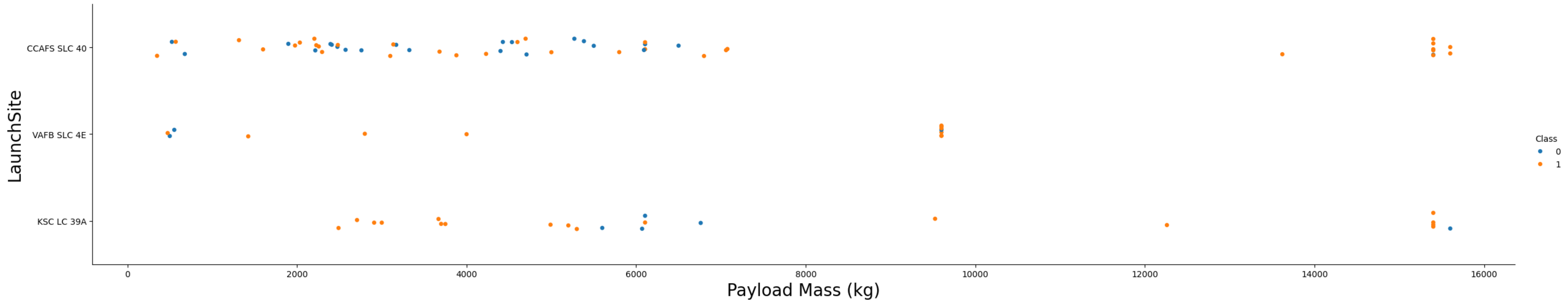
# Flight Number vs. Launch Site

- Initial flights demonstrated a comparatively lower success rate, marked by blue indicating failures.
- Subsequent flights exhibited an improved success rate, denoted by the prevalence of orange indicating successful missions.
- Approximately 50% of launches originated from the CCAFS SLC 40 launch site.
- VAFB SLC 4E and KSC LC 39A emerged with notably higher success rates compared to other launch sites.
- An inference can be drawn, suggesting that recent launches tend to boast a heightened success rate.



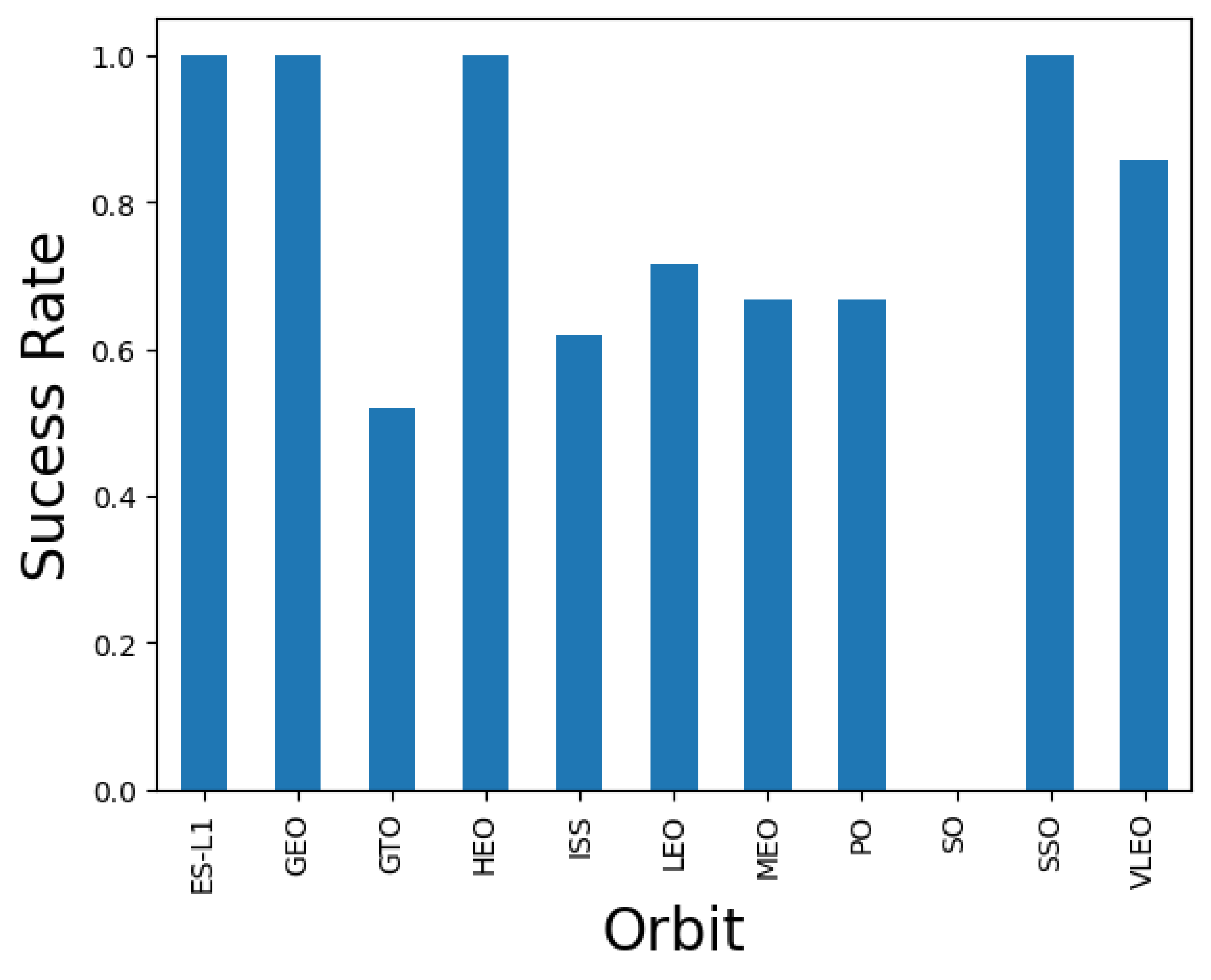
# Payload vs. Launch Site

- Generally, a positive correlation exists between higher payload masses (kg) and elevated success rates.
- Launches featuring payloads surpassing 7,000 kg predominantly resulted in success.
- KSC LC 39A boasts a flawless 100% success rate for launches with payloads less than 5,500 kg.
- VAFB SLC 4E, on the other hand, has not conducted any launches exceedingly approximately 10,000 kg, indicating a specific payload capacity trend for this launch site.



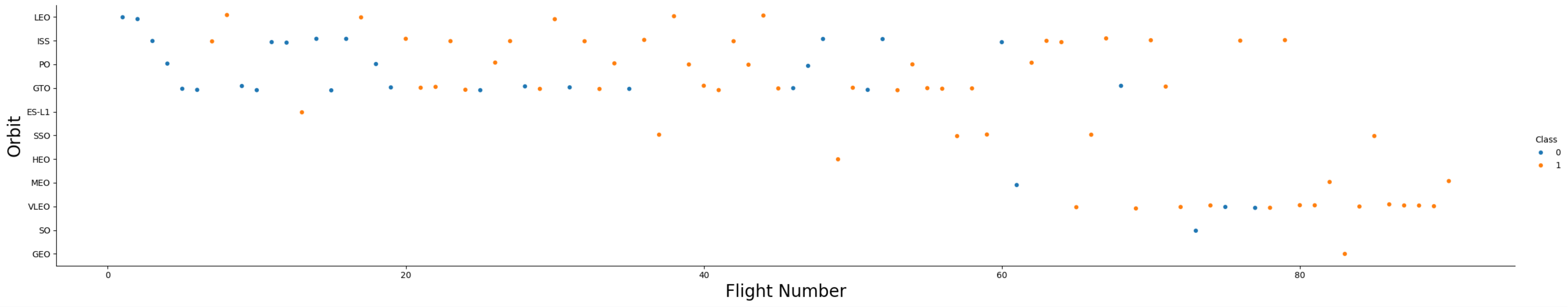
# Success Rate vs. Orbit Type

- **100% Success Rate**: ES-L1, GEO, HEO and SSO
- **50%-80% Success Rate**: GTO, ISS, LEO, MEO, PO
- **0% Success Rate**: SO



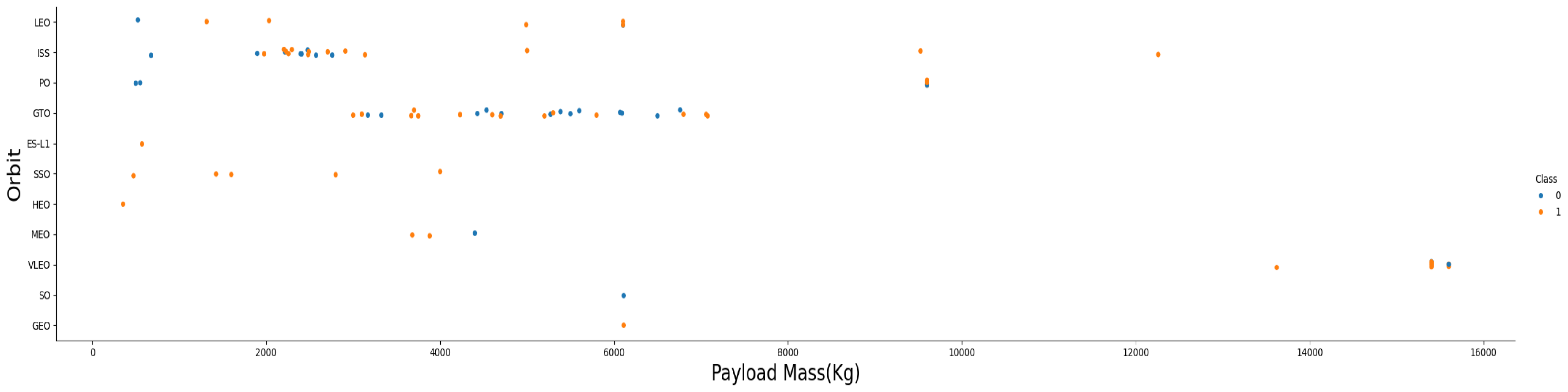
# Flight Number vs. Orbit Type

- The success rate demonstrates a consistent upward trend as the number of flights per orbit increases.
- This correlation is particularly pronounced in the case of the LEO Orbit.
- Conversely, the GTO Orbit exhibits a deviation from this observed trend, displaying a distinct pattern in its success rates.



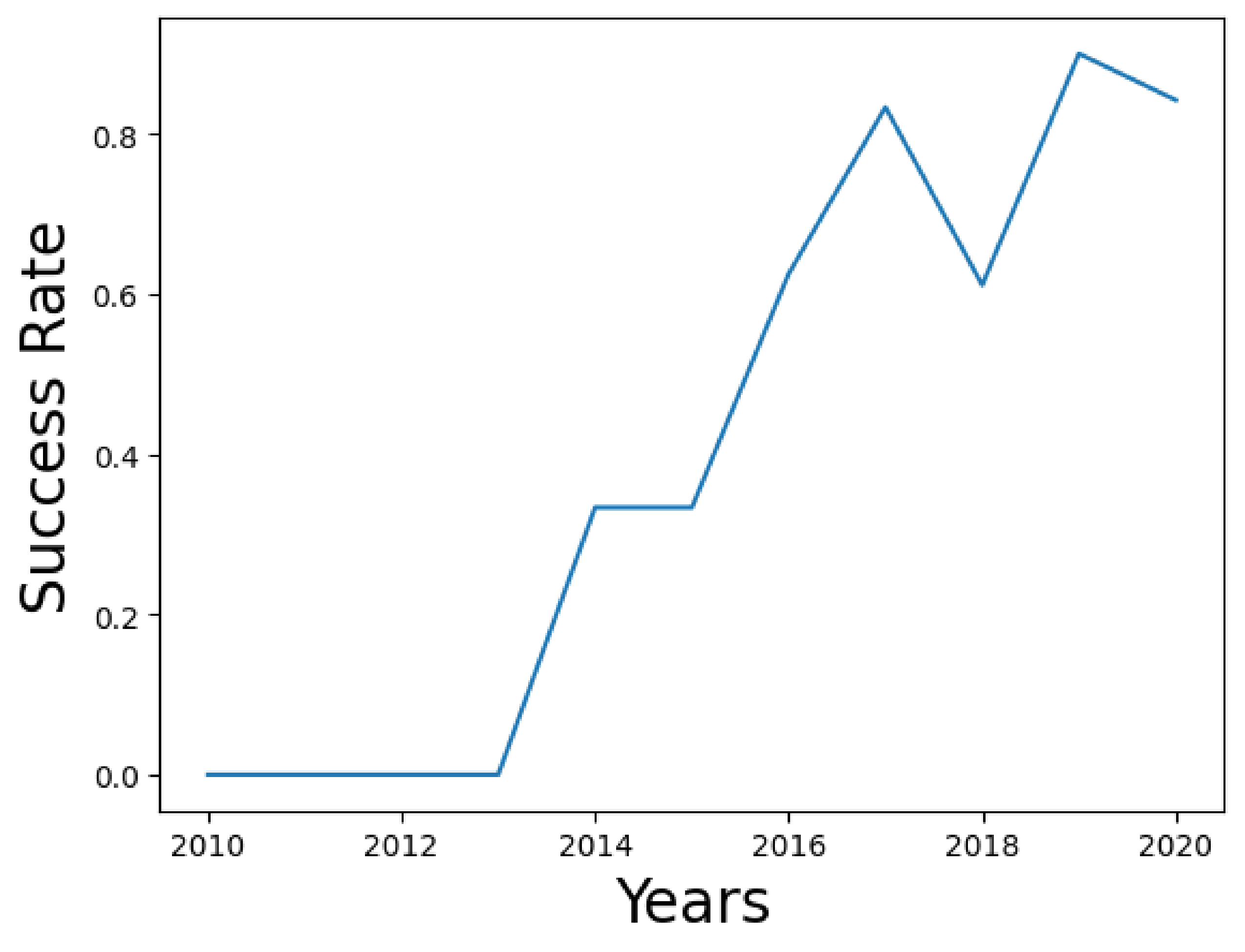
# Payload vs. Orbit Type

- Optimal performance is observed for heavy payloads in LEO, ISS, and PO orbits.
- The Geostationary Transfer Orbit (GTO) exhibits varying success rates, particularly with heavier payloads, showcasing a nuanced performance in this orbit



# Launch Success Yearly Trend

- Notable enhancement in the success rate was observed during the periods 2013-2017 and 2018-2019.
- A decline in the success rate was noted in the intervals 2017-2018 and from 2019-2020.
- There has been a positive trajectory in the success rate since the initiation of the program in 2013.



# All Launch Site Names

- The keyword DISTINCT was used to select unique launch sites from the table.

```
[ ]  %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;

 * 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'

- The keyword LIKE is used to specify that the launch site name begins with CCA.
- The limit keyword is used to query only 5 records.

```
[ ]  %sql select LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

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

| Launch_Site |
|---|
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |

# Total Payload Mass

- The total payload mass for customer NASA (CSR) is 45,596 kg.
- The function SUM is adopted to gather the sum of payload mass.
- The WHERE clause is used to specify that the customer should be NASA (CRS)

```
[ ]  %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

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

| payloadmass |
|---|
| 619967 |

# Average Payload Mass by F9 v1.1

- The average payload mass for rockets with booster version F9 v1.1 is 2928.4 kg.
- The function AVG is used to calculate the average of payload mass.
- The WHERE clause is used to specify that the booster version should be F9 v1.1

```
[ ]  %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

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

| payloadmass |
| --- |
| 6138.287128712871 |

# First Successful Ground Landing Date

- The first successful ground landing occurred on 22 December 2015.
- The function MIN is used to find the earliest date.
- The WHERE clause is used to specify that the landing outcome is a successful ground pad landing.

```
%sql SELECT strftime('%m', DATE) as Month, MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL \
WHERE strftime('%Y', DATE) = '2015';
```

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

| Month | Mission_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Success | F9 v1.1 B1012 | CCAFS LC-40 |
| 02 | Success | F9 v1.1 B1013 | CCAFS LC-40 |
| 03 | Success | F9 v1.1 B1014 | CCAFS LC-40 |
| 04 | Success | F9 v1.1 B1015 | CCAFS LC-40 |
| 04 | Success | F9 v1.1 B1016 | CCAFS LC-40 |
| 06 | Failure (in flight) | F9 v1.1 B1018 | CCAFS LC-40 |
| 12 | Success | F9 FT B1019 | CCAFS LC-40 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- Identification of successfully landed boosters on a drone ship with a payload mass exceeding 4000 but less than 6000, including: F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2.
- Implementation of the WHERE clause to filter cases where the landing is classified as a successful drone ship landing.
- Introduction of the AND clause to incorporate an additional condition specifying that the payload mass falls within the range of 4000 to 6000.

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * 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

- Successful Mission Outcomes: 61
- Failed Mission Outcomes: 10
- The COUNT function was employed to determine the total number of landings.
- The LIKE clause played a crucial role in categorizing outcomes as either successes or failures, contributing to a more detailed analysis.

```
[ ]  %sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
     FROM SPACEXTBL \
     GROUP BY MISSION_OUTCOME;

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

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

# Boosters Carried Maximum Payload

- Leveraging the WHERE clause, the selection process is refined to focus specifically on the booster version with the highest payload mass.
- Utilizing the MAX function, the system identifies and retrieves the maximum payload mass associated with the booster versions under consideration

```
[ ]  %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);

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

| boosterversion |
| --- |
| F9 B5 B1048.4 |
| F9 B5 B1049.4 |
| F9 B5 B1051.3 |
| F9 B5 B1056.4 |
| F9 B5 B1048.5 |
| F9 B5 B1051.4 |
| F9 B5 B1049.5 |
| F9 B5 B1060.2 |
| F9 B5 B1058.3 |
| F9 B5 B1051.6 |
| F9 B5 B1060.3 |
| F9 B5 B1049.7 |

# 2015 Launch Records

- The strftime() function serves to format a date-time value according to a specified format, providing flexibility in date representation.
- To refine queries, the WHERE clause is employed, allowing for conditional filtering of data.
- In a specific instance, the WHERE clause is utilized to narrow down results, specifying that the date should align with the year 2015.

```
[ ] %sql SELECT strftime('%m', DATE) as Month, MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL \
    WHERE strftime('%Y', DATE) = '2015';
```

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

| Month | Mission_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Success | F9 v1.1 B1012 | CCAFS LC-40 |
| 02 | Success | F9 v1.1 B1013 | CCAFS LC-40 |
| 03 | Success | F9 v1.1 B1014 | CCAFS LC-40 |
| 04 | Success | F9 v1.1 B1015 | CCAFS LC-40 |
| 04 | Success | F9 v1.1 B1016 | CCAFS LC-40 |
| 06 | Failure (in flight) | F9 v1.1 B1018 | CCAFS LC-40 |
| 12 | Success | F9 FT B1019 | CCAFS LC-40 |

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

- Count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

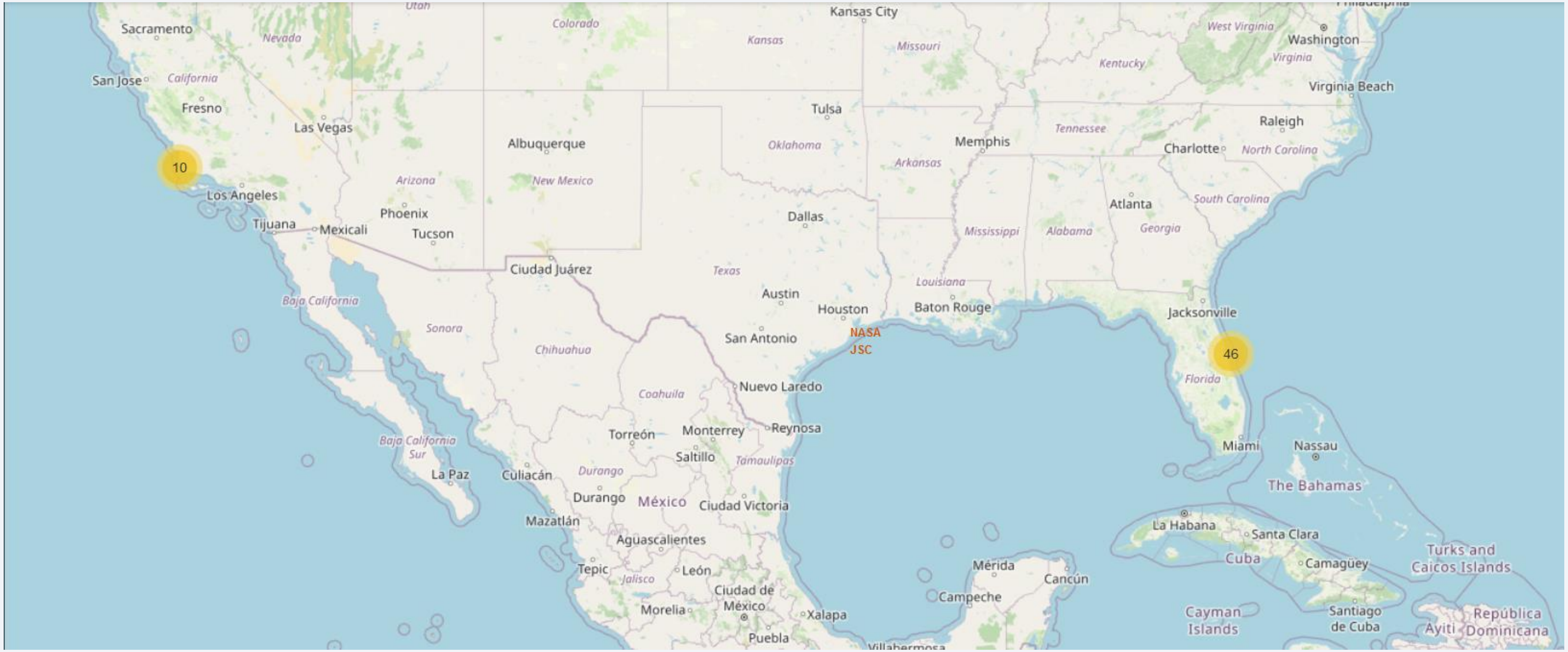
```
[ ] %sql SELECT "Landing_Outcome" FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

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

| Landing_Outcome |
|---|
| No attempt |
| Success (ground pad) |
| Success (drone ship) |
| Success (drone ship) |
| Success (ground pad) |
| Failure (drone ship) |
| Success (drone ship) |
| Success (drone ship) |
| Success (drone ship) |
| Failure (drone ship) |
| Failure (drone ship) |
| Success (ground pad) |
| Precluded (drone ship) |
| No attempt |
| Failure (drone ship) |
| No attempt |
| Controlled (ocean) |
| Failure (drone ship) |
| Uncontrolled (ocean) |
| No attempt |
| No attempt |

Section 3

# Launch Sites Proximities Analysis

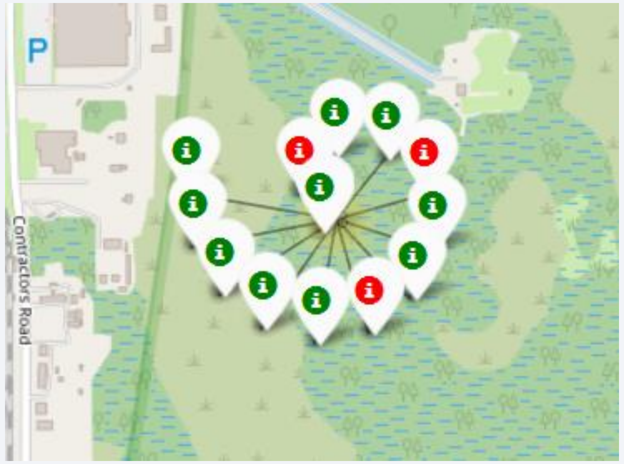
# SapceX Launch Sites

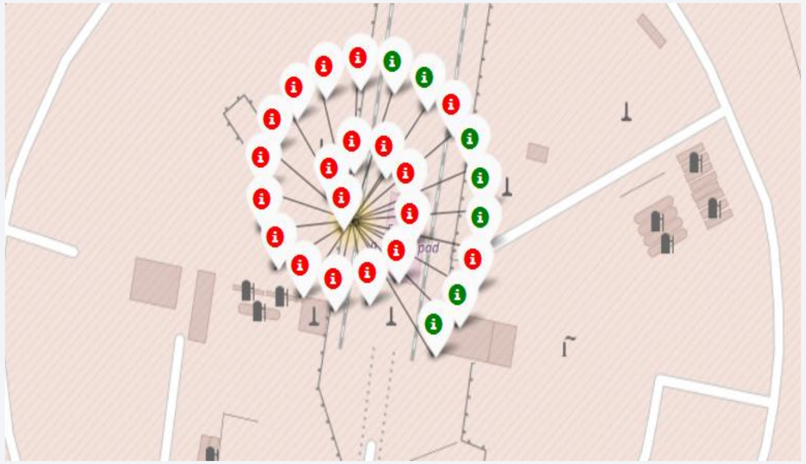



- SpaceX Launch Sites are strategically positioned along the coastlines of Florida and California in the United States.
- Among these launch sites, VAFB SLC-4E stands as the sole facility located in California, while the remaining sites are situated in Florida. This strategic distribution allows SpaceX to optimize launch capabilities from distinct geographical vantage points.

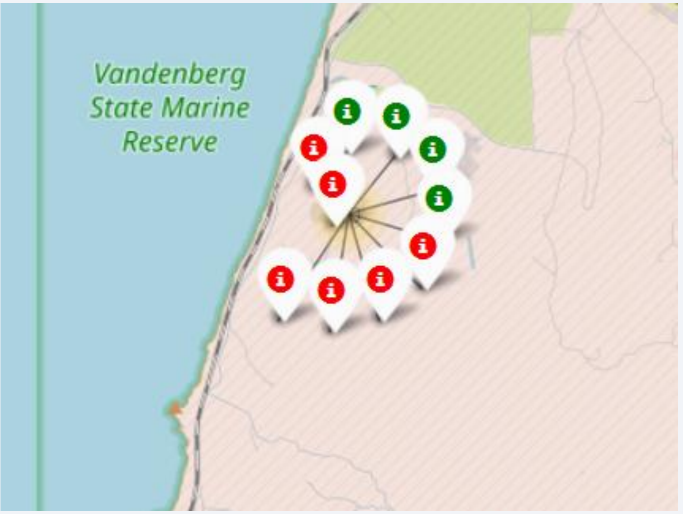
# Launches Outcomes (Success/Failure)

- Florida Launches



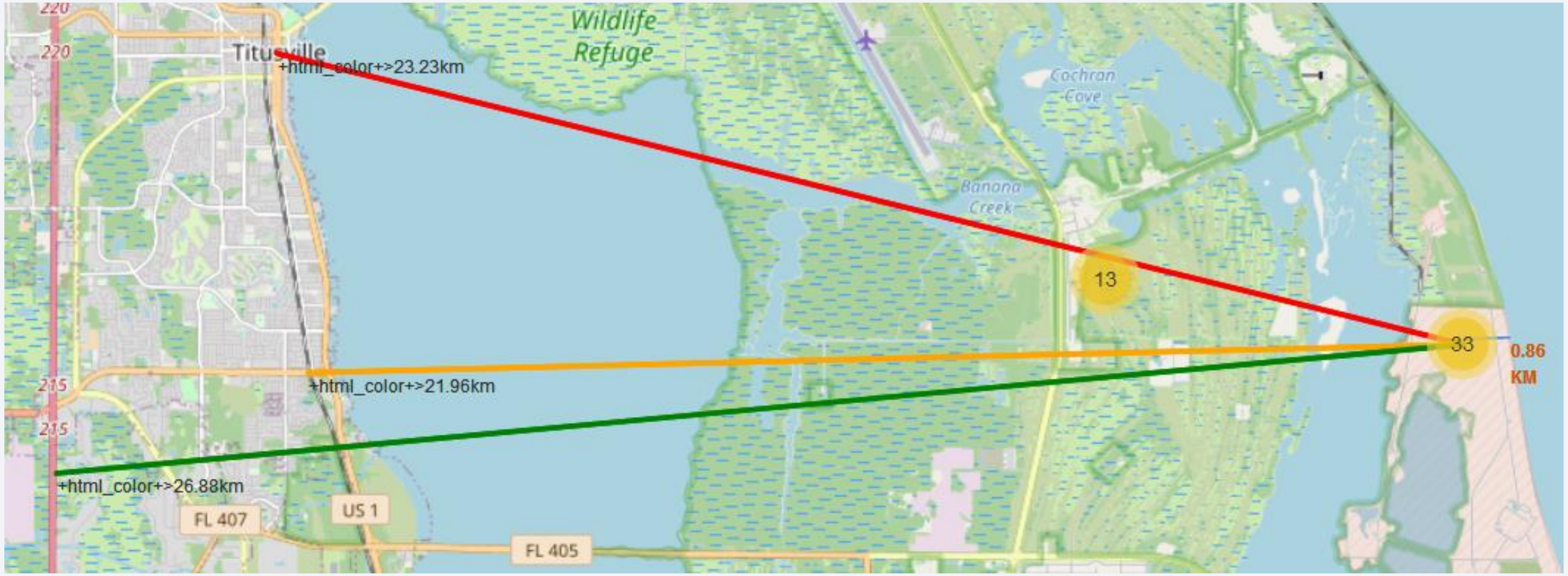



- California Launches



# Launches Sites Proximity From Landmarks

- City Distance 23.234
- Railway Distance 21.96
- Highway Distance 26.882
- Coastline Distance 0.862

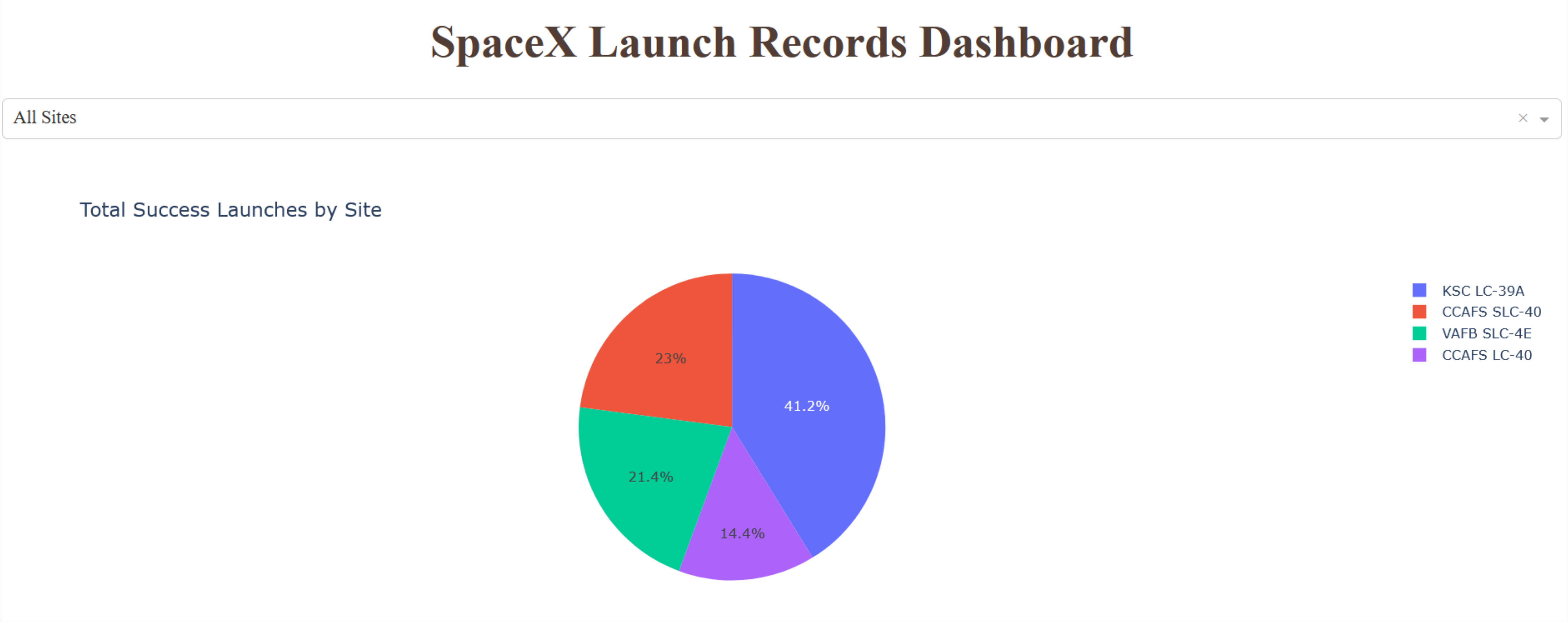


Section 4

# Build a Dashboard with Plotly Dash

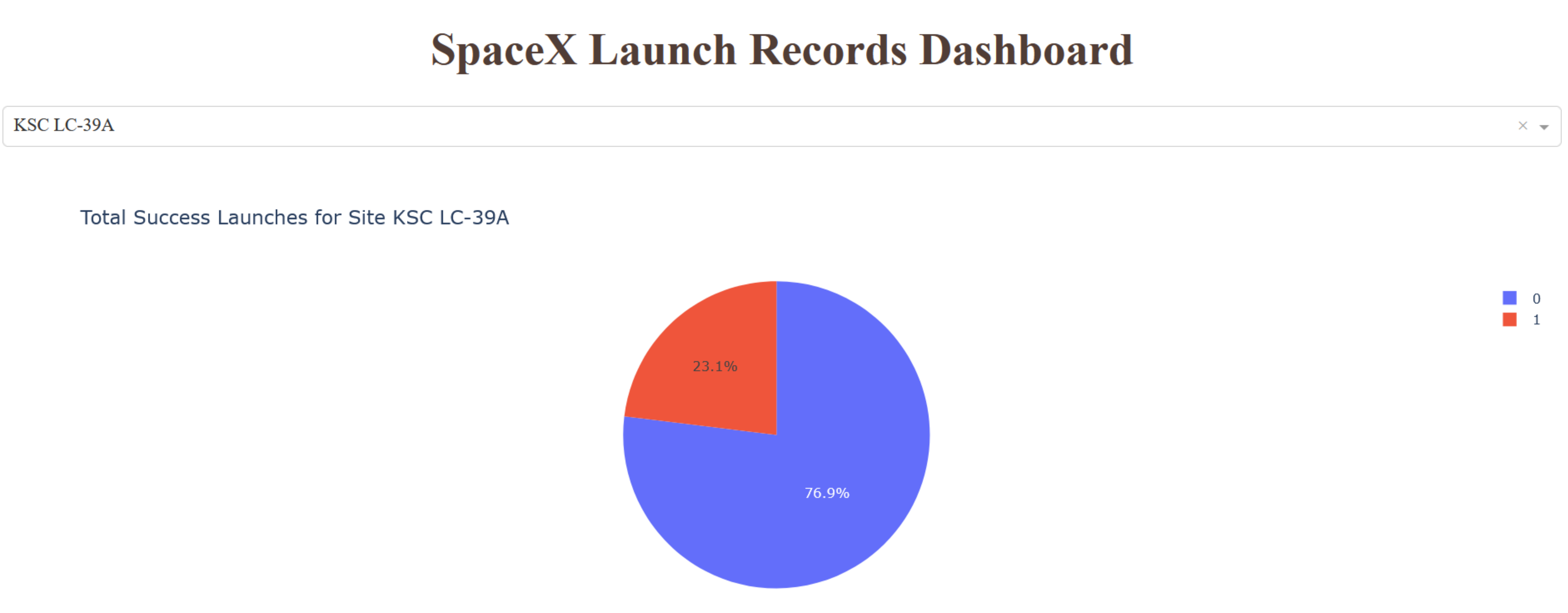
# Launch Success by Site

Launch Site KDC LC-39 A has the Greatest Launch Success Rate



# Launched for KSC LC-39A: Success/Failure Rates

Launch site KSC LC-39A achieved a launch success rate of 76.9%



# Payload Mass and Success

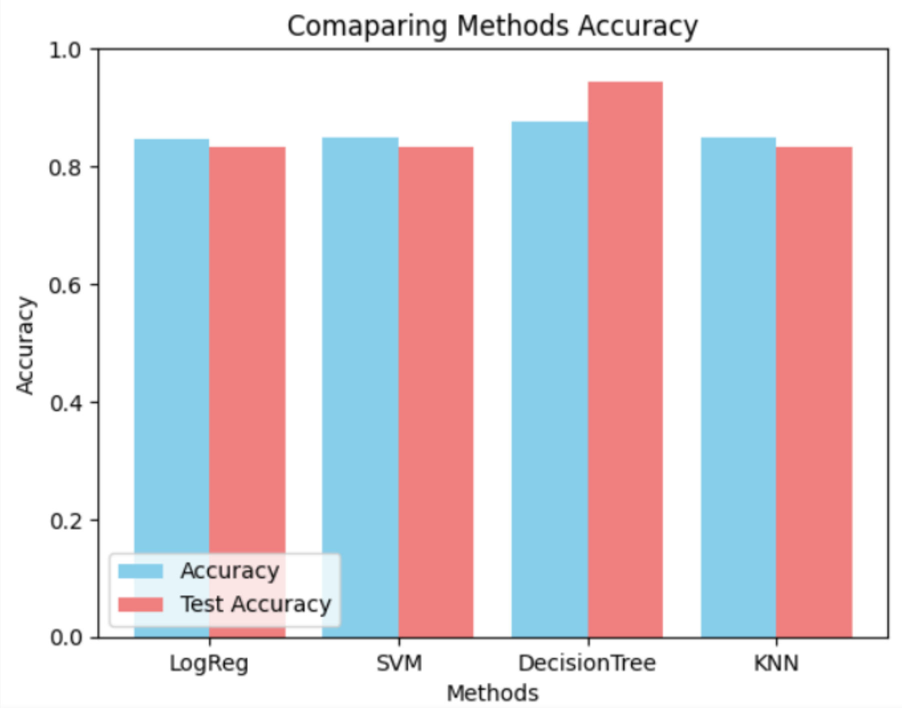
Payloads between 2,000 kg and 5,000 kg have the highest success rate

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- Visualization of the built model accuracy for all built classification models, in a bar chart

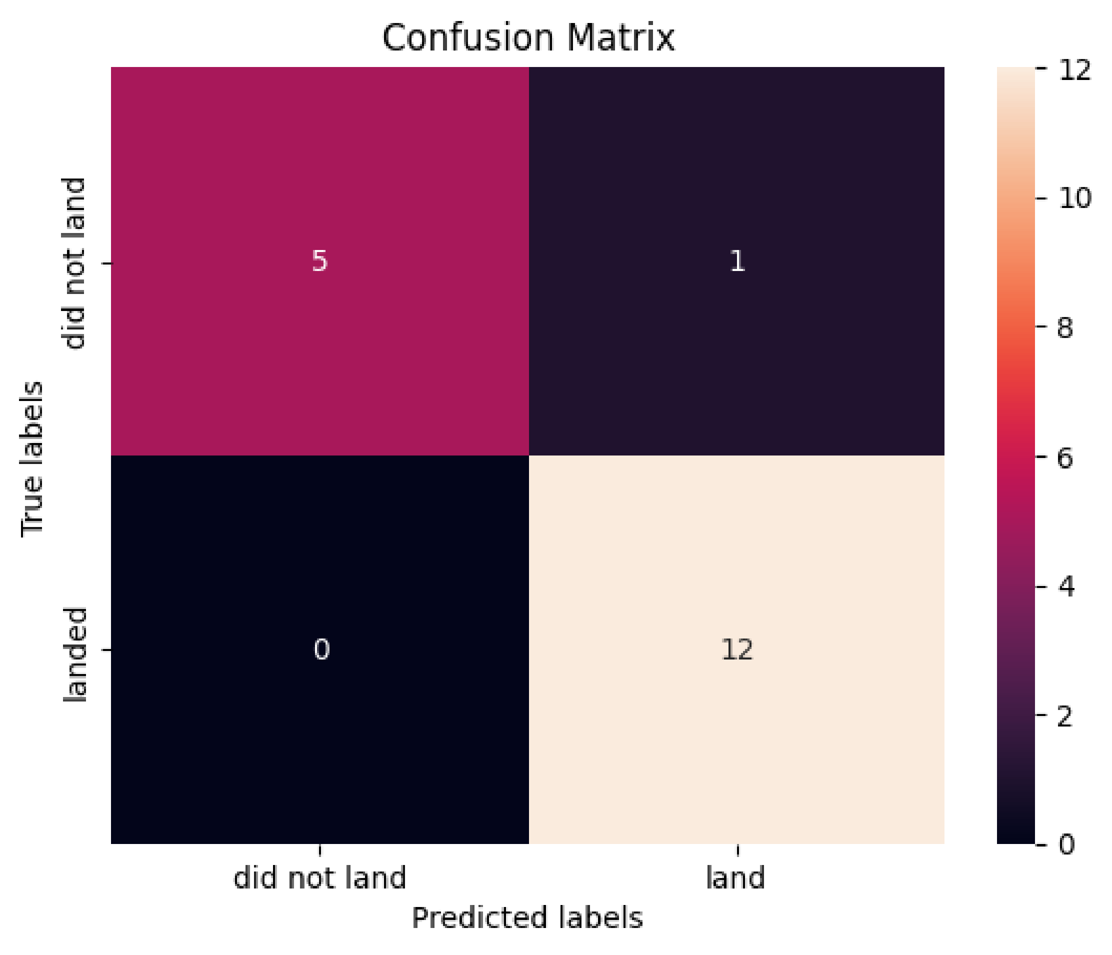



- The Decision Tree model has the highest classification accuracy

| Model | Accuracy | TestAccuracy |
|---|---|---|
| LogReg | 0.84643 | 0.83333 |
| SVM | 0.84821 | 0.83333 |
| DecisionTree | 0.875 | 0.94444 |
| KNN | 0.84821 | 0.83333 |

# Confusion Matrix

- The Decision Tree Classifier stands out as the most optimal model.
- Confusion matrix reveals a notable occurrence of false positives, indicating instances where the rocket did not experience a successful failure, yet the classifiers incorrectly predicted a lack of success.
- This insight emphasizes the need for further examination and potential adjustments to improve the model's predictive accuracy.



# Conclusions

- The models exhibited comparable performance on the test set, with the decision tree model demonstrating a slight high accuracy.
- Proximity to the Equator is strategically leveraged, capitalizing on the Earth's rotational speed to achieve additional natural boost, thereby minimizing the need for extra fuel and boosters.
- All launch sites are strategically located near coastlines, optimizing logistical and operational efficiencies.
- The success rate of launches has demonstrated a consistent upward trajectory over time.
- KSC LC-39A stands out with the highest success rate among launch sites. Remarkably, it boasts a 100% success rate for launches with a payload less than 5,500 kg.
- Launches targeting ES-L1, GEO, HEO, and SSO orbits have consistently achieved a 100% success rate.
- Across all launch sites, a positive correlation exists between higher payload mass (kg) and a heightened success rate.

# Thank you!