Winning Space Race with Data Science

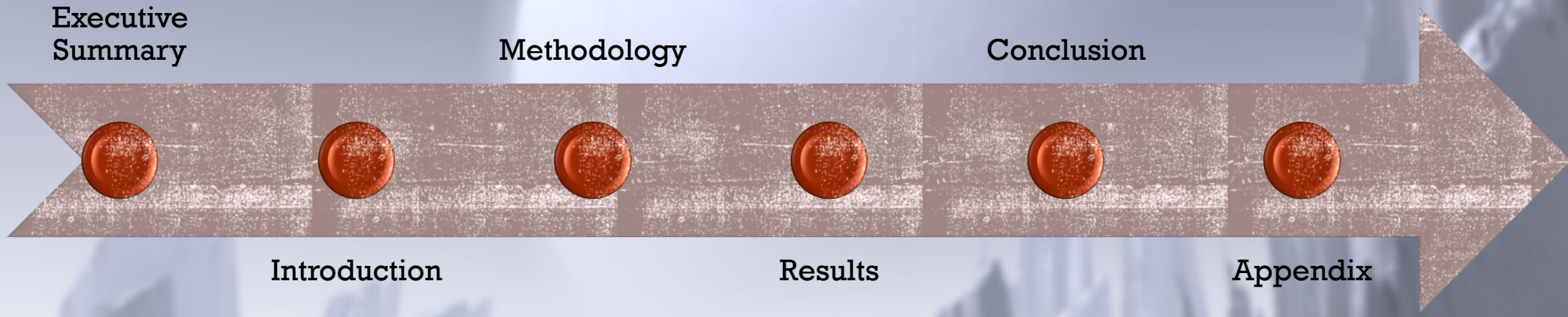
Sarmad Mehrabian

9/12/2023

GitHub Link for This Presentation

Note: For better readability, please use "Full Screen"

Outline


Executive Summary
Introduction
Methodology
Results
Conclusion
Appendix

Executive Summary

Summary of methodologies

Introduction

Project background and context:

- SpaceX promotes Falcon 9 rocket launches on its website, pricing them at $62 million, while competitors charge over $165 million for similar services. The primary reason for these cost savings is that SpaceX's Falcon 9 can recycle its first stage, unlike other rocket providers.

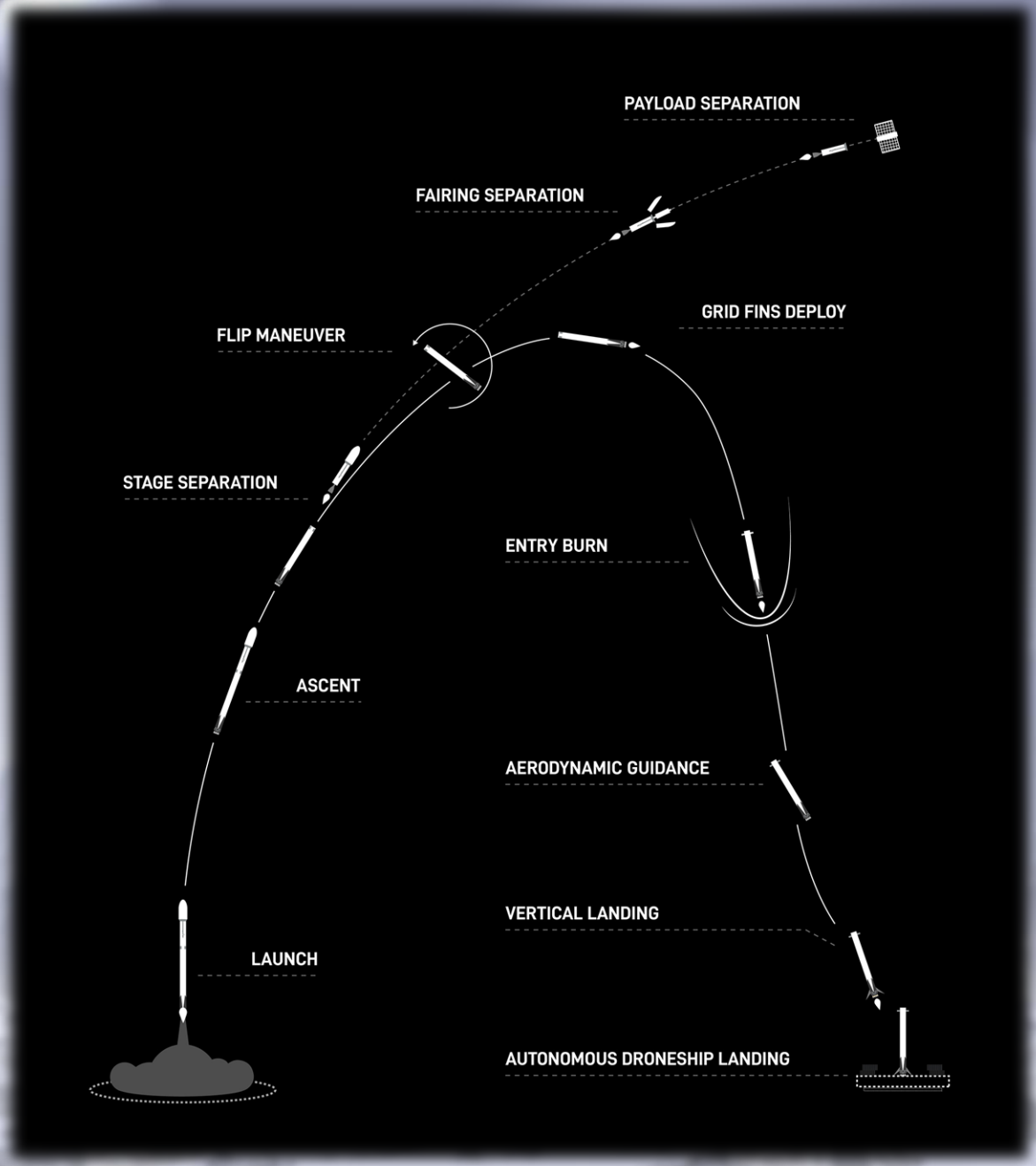
Problem to Solve:

- Our goal is to figure out whether the first stage of the Falcon 9 rocket will successfully land. This determination is crucial for estimating the overall launch cost. To achieve this, we plan to develop a machine learning model using historical data from Falcon 9 rocket launches to predict the likelihood of the SpaceX first-stage landing successfully.

On the next pages, you can observe the process of launching Falcon 9 Rockets and their landing on a Droneship or a Landing Zone

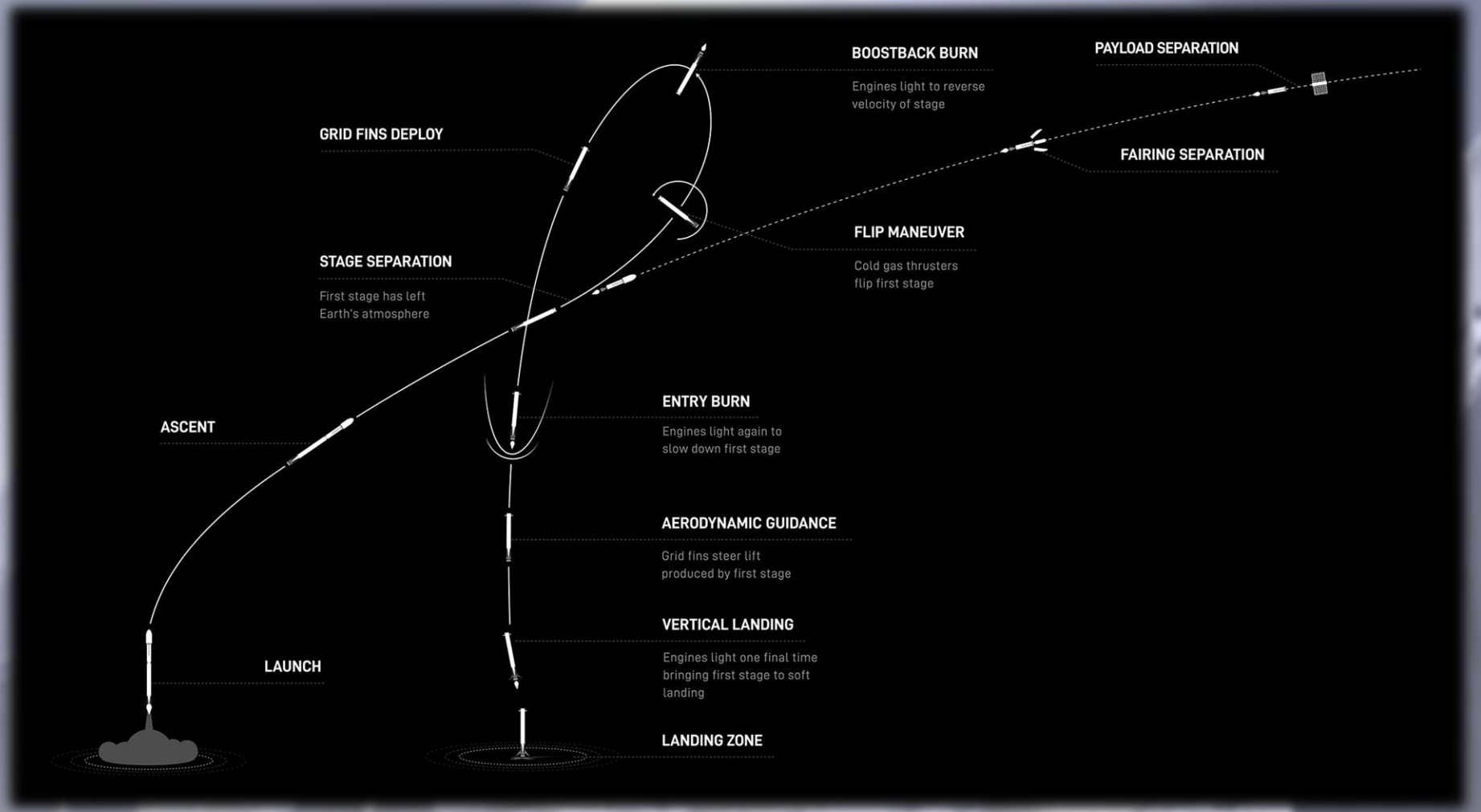
DRONESHIP

Source page



LANDING ZONE

Source page



Section 1

Methodology

Methodology

Executive Summary

A) Data collection methodology:

B) Data wrangling:

C) Exploratory Data Analysis (EDA) using Visualization and SQL

D) Interactive Visual Analytics using Folium and Plotly Dash

E) Predictive Analysis using Classification Models:

We've constructed and assessed Linear Regression, K Nearest Neighbors, Support Vector Machine, and Decision Tree models to identify the most effective algorithm.

Data Collection

Data was gathered through a variety of techniques:

Space X REST API

- - To initiate data collection, we utilized a GET request to access the SpaceX API.
- - Subsequently, we decoded the response content as JSON using the .json() function and transformed it into a pandas data frame using .json_normalize().
- - Following that step, we conducted data cleansing, examined for any absent values, and filled in missing data when required.

Web Scraping

- - Additionally, we engaged in web scraping from Wikipedia to obtain Falcon 9 launch records utilizing BeautifulSoup.
- - Our primary aim was to extract the launch records presented in HTML tables, parse this tabular data, and convert it into a pandas data frame to facilitate future analysis.

On the next page, you can observe
a scheme for these steps

Data Collection

1. Use Space X REST API
2. Returns Space X data in Json format
3. Convert to a data frame using Normalize method

Ready for data wrangling and preparation

3. Convert to a dataframe
2. Extract data using beautiful soup
1. Get HTML response from Wikipeida

Data Collection – SpaceX API

We employed a GET request to access the SpaceX API for data retrieval, where we subsequently performed data cleaning, basic data manipulation, and formatting tasks.

1. GET request for rocket launch data using API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"


response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json results to a data frame

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

3. Then, we performed data cleaning and filling the missing values

```
# Calculate the mean value of PayloadMass column
payloadmass_mean = data_falcon9["PayloadMass"].mean()
print("Mean PayloadMass:", payloadmass_mean)

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payloadmass_mean, inplace=True)
```

Here's the link to the API Data Collection notebook

Data Collection - Scraping

We utilized web scraping techniques using BeautifulSoup to extract Falcon 9 launch records. Subsequently, we parsed the acquired table data and transformed it into a pandas data frame.

Here's the link to the Web Scraping notebook

1. Applying GET method to request the Falcon 9 rocket launch page

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

✓ 0.0s

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

✓ 3.8s

```
#Checking the status code
response.status_code
```

✓ 0.0s

```
200
```

Go to the next page to see the rest of this page

2. Create a BeautifulSoup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

✓ 0.6s

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

```
# Use soup.title attribute
title = soup.title.string
print(f"Page Title: {title}")
```

✓ 0.0s

```
Page Title: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

Here's the link to the Web Scraping notebook

3. Extract all column names from the HTML table header

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

for i in first_launch_table.find_all('th'):
    if extract_column_from_header(i)!=None:
        if len(extract_column_from_header(i))>0:
            column_names.append(extract_column_from_header(i))
```

✓ 0.0s

Data Wrangling

1. Exploratory data analysis was performed, and the training labels were determined.
2. The number of launches at each site was calculated, along with the number and occurrence of each orbit.
3. The landing outcome label was created from the outcome column, and the results were exported to a CSV file.

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

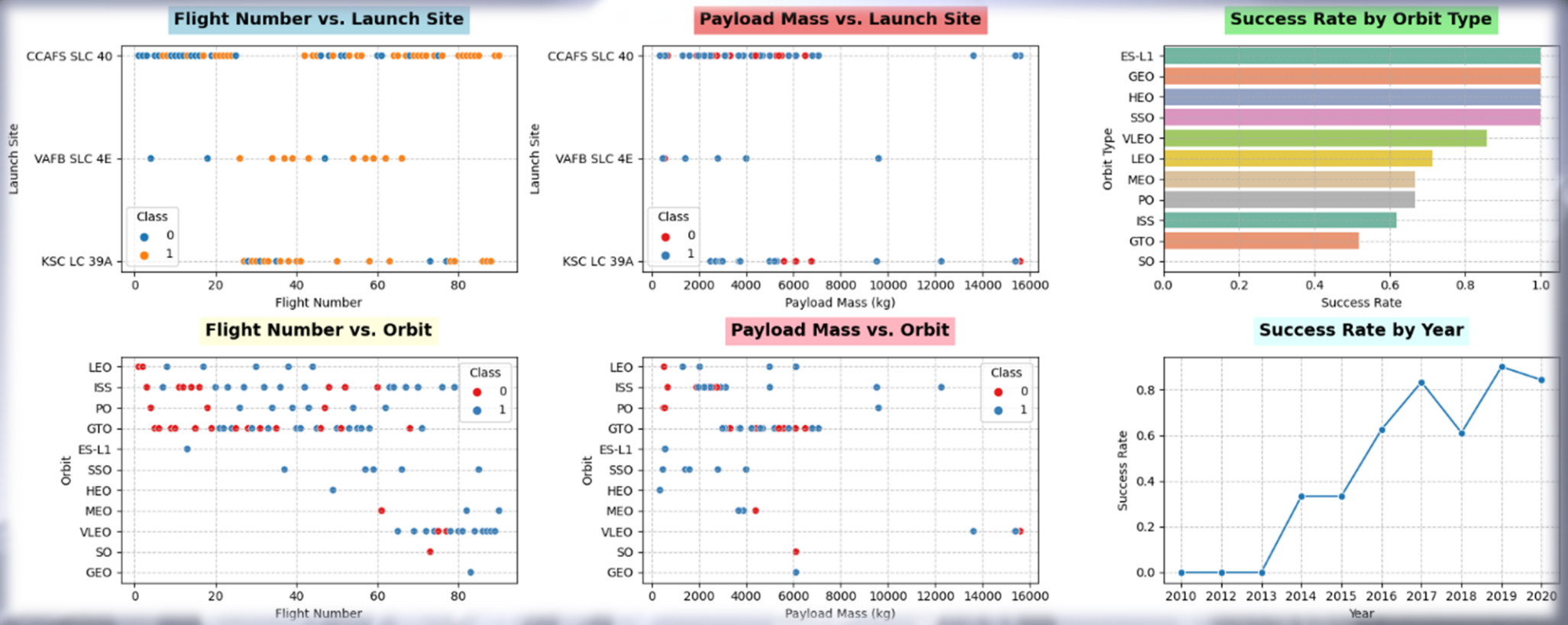
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)

# Convert the 'landing_class' list to a Pandas Series
landing_class = pd.Series(landing_class, name='landing_class')
```

Here's the link to the Data Wrangling notebook

EDA with Data Visualization



Here's the link to the Data Visualization notebook

EDA with SQL

- ✓ Showing the names of the unique places where space missions launch from.
- ✓ Displaying 5 entries where launch sites start with 'KSC'.
- ✓ Revealing the total weight of cargo carried by NASA's launched rockets (CRS).
- ✓ Showing the average weight of cargo carried by rocket version F9 v1.1.
- ✓ Listing the dates when drones successfully landed on a ship.
- ✓ Listing the names of rockets that landed successfully on the ground and had cargo weights between 4000 and 6000.
- ✓ Counting the total number of missions that succeeded and failed.
- ✓ Listing the names of rocket versions that carried the heaviest cargo.
- ✓ Displaying the records that include month names, successful landings on ground pads by rocket versions, and launch sites for the months in the year 2017.
- ✓ Ranking the number of successful landings between June 4, 2010, and March 20, 2017, from highest to lowest.

Here's the link to the SQL notebook

Build an Interactive Map with Folium

All launch sites were marked on the Folium map, and map objects such as markers, circles, and lines were added to indicate the success or failure of launches at each site.

The launch outcomes were assigned the features of class 0 and 1, specifically 0 for failure and 1 for success.

With the use of color-coded marker clusters, the launch sites with relatively high success rates were identified.

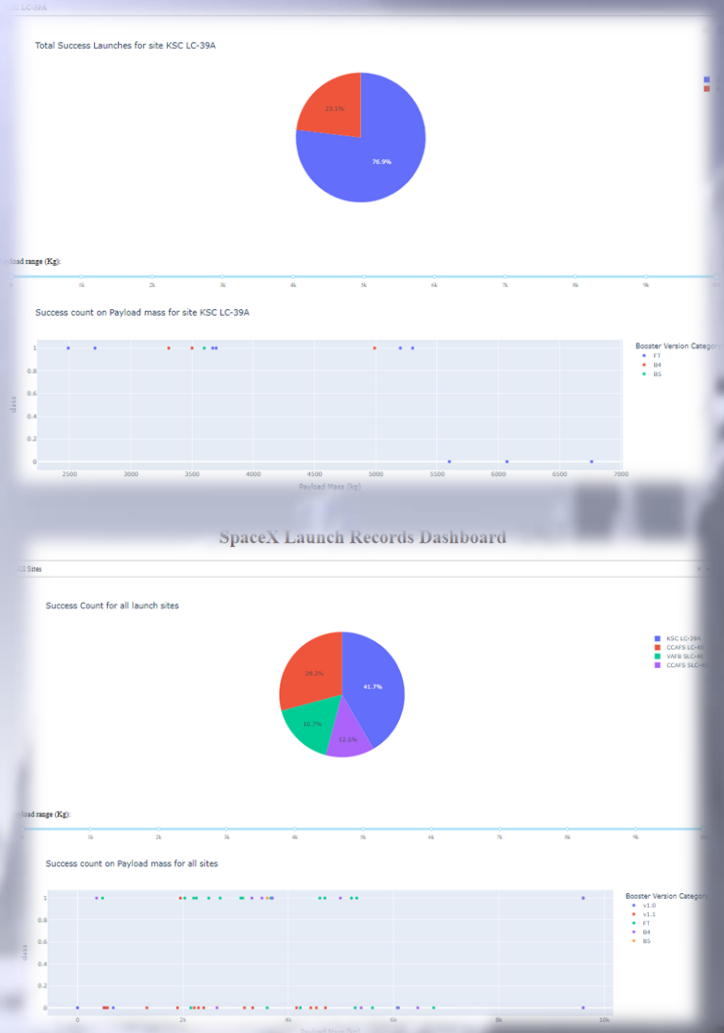
Distances between a launch site and its proximities were calculated.

[Here's the link to the Interactive Map notebook]

Build a Dashboard with Plotly Dash

The dashboard has these charts:

- A dropdown menu to pick a launch site.
- A pie chart showing how often launches succeed at the chosen site.
- A slider to choose the payload weight range.
- A scatter plot that shows how payload weight affects launch success at the chosen sites.



Here's the link to the Plotly Dash notebook

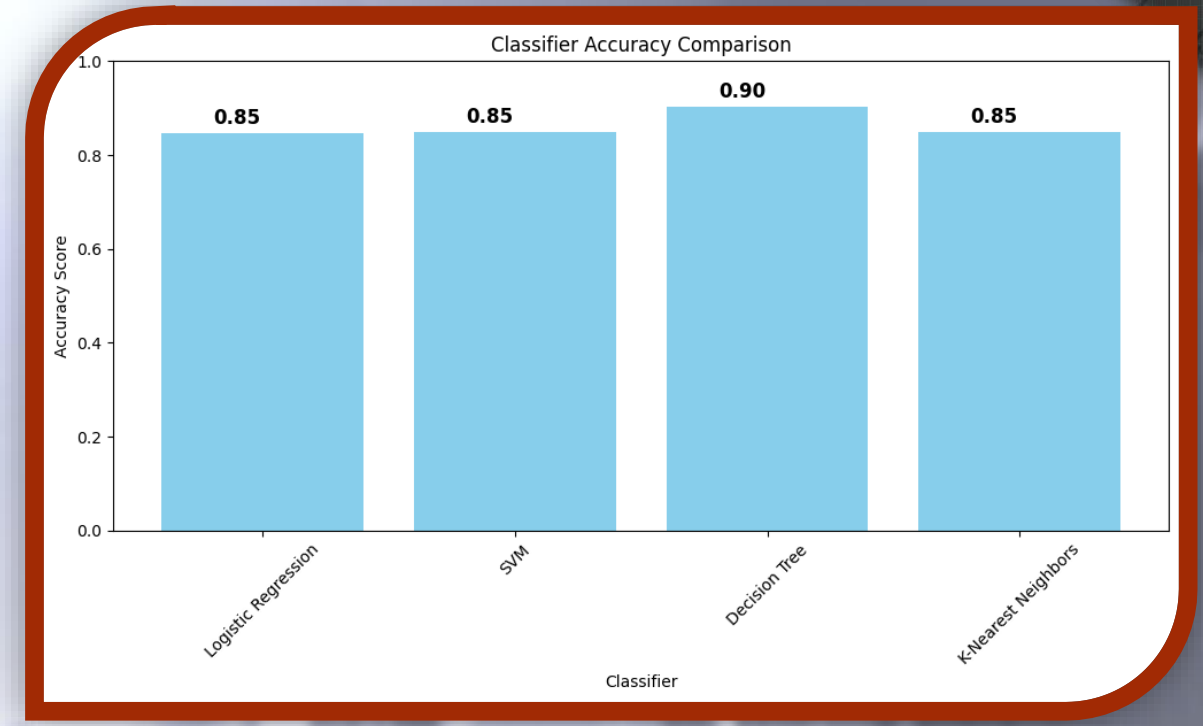
Predictive Analysis (Classification)

1. The data was loaded using numpy and pandas, followed by data transformation, and then a split into training and testing sets was performed.

2. Different machine learning models were built, and various hyperparameters were tuned using GridSearchCV.

3. Accuracy was utilized as the metric for our model, and improvements were made to the model through feature engineering and algorithm tuning.

4. The best-performing classification model was identified.



Here's the link to the Classification notebook

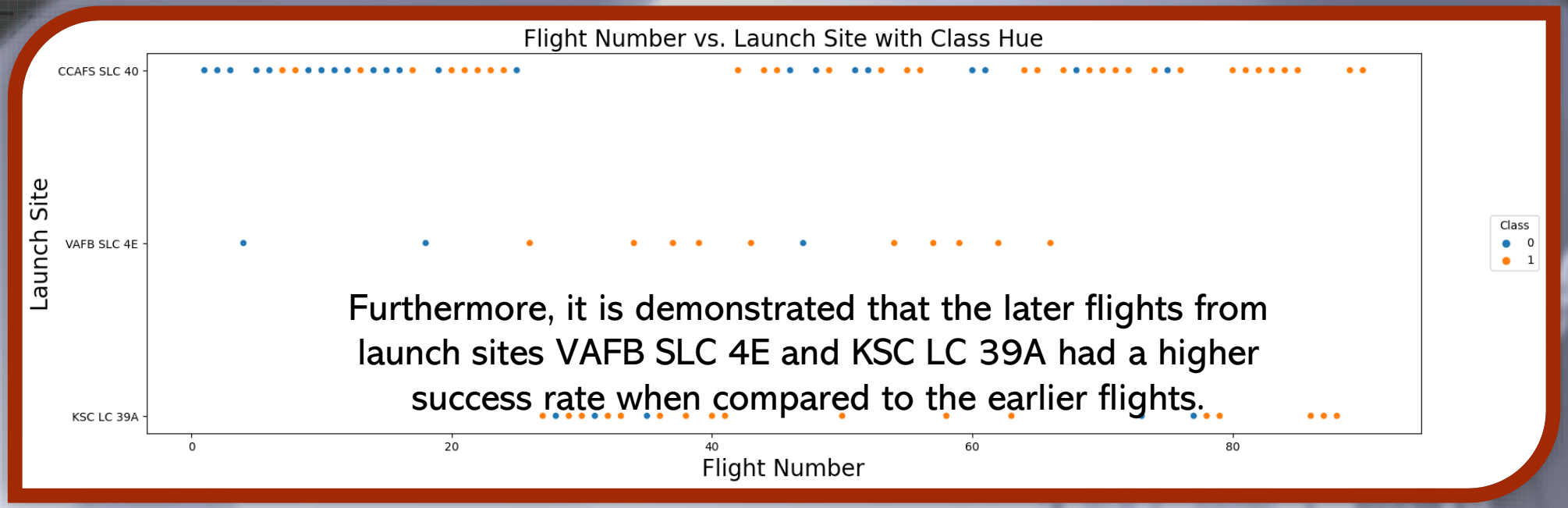
Results

1. A successful landing on a ground pad was achieved for the first time on December 22, 2015.
2. Success rates for SpaceX launches showed improvement from 2013 to 2020.
3. Mission success appears to be significantly influenced by the launch location.
4. More successful launches were recorded at KSC LC-39A compared to other sites.
5. Lighter payloads outperform heavier ones.
6. The highest success rates were observed in ES-L1, GEO, HEO, and SSO orbits.
7. When it comes to heavy payloads, VLO and ISS orbits demonstrated higher success rates.
8. Among the different models used, the Decision Tree model was determined to be the most accurate for predicting outcomes with this dataset.

Section 2

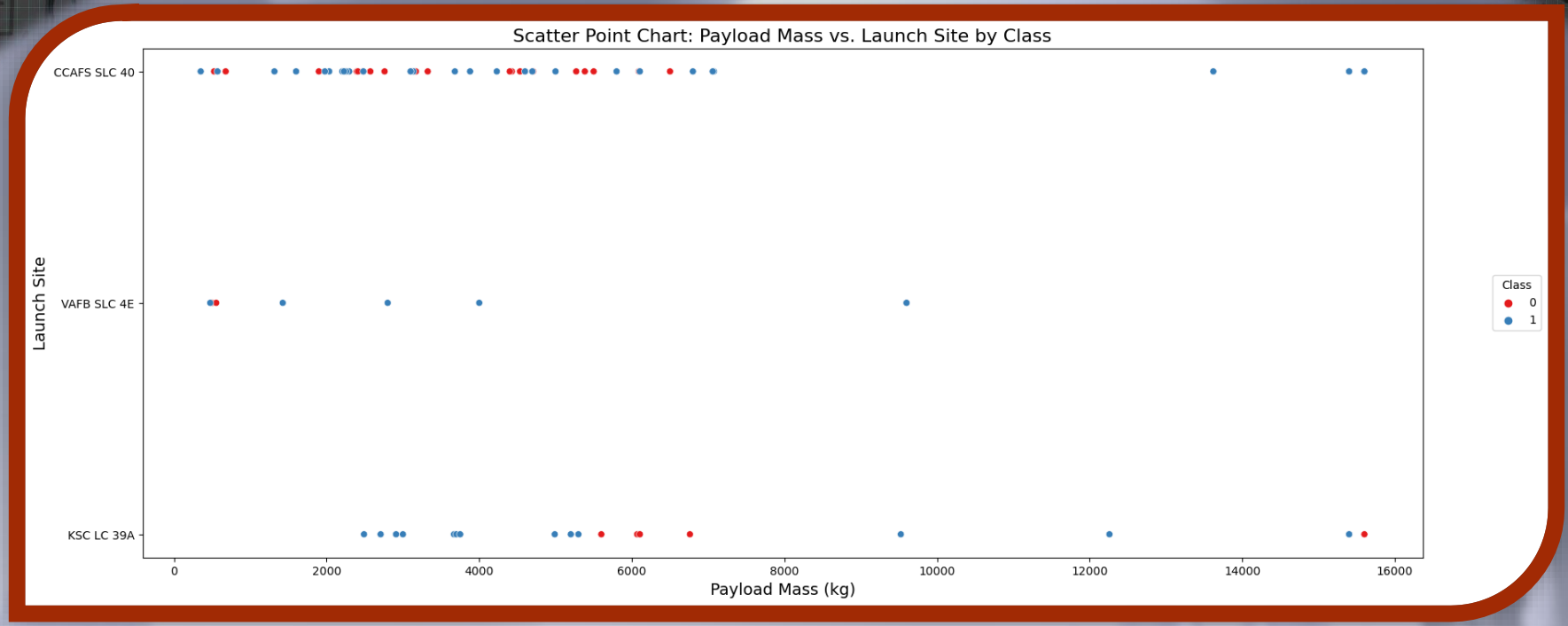
Insights drawn from EDA

Flight Number vs. Launch Site



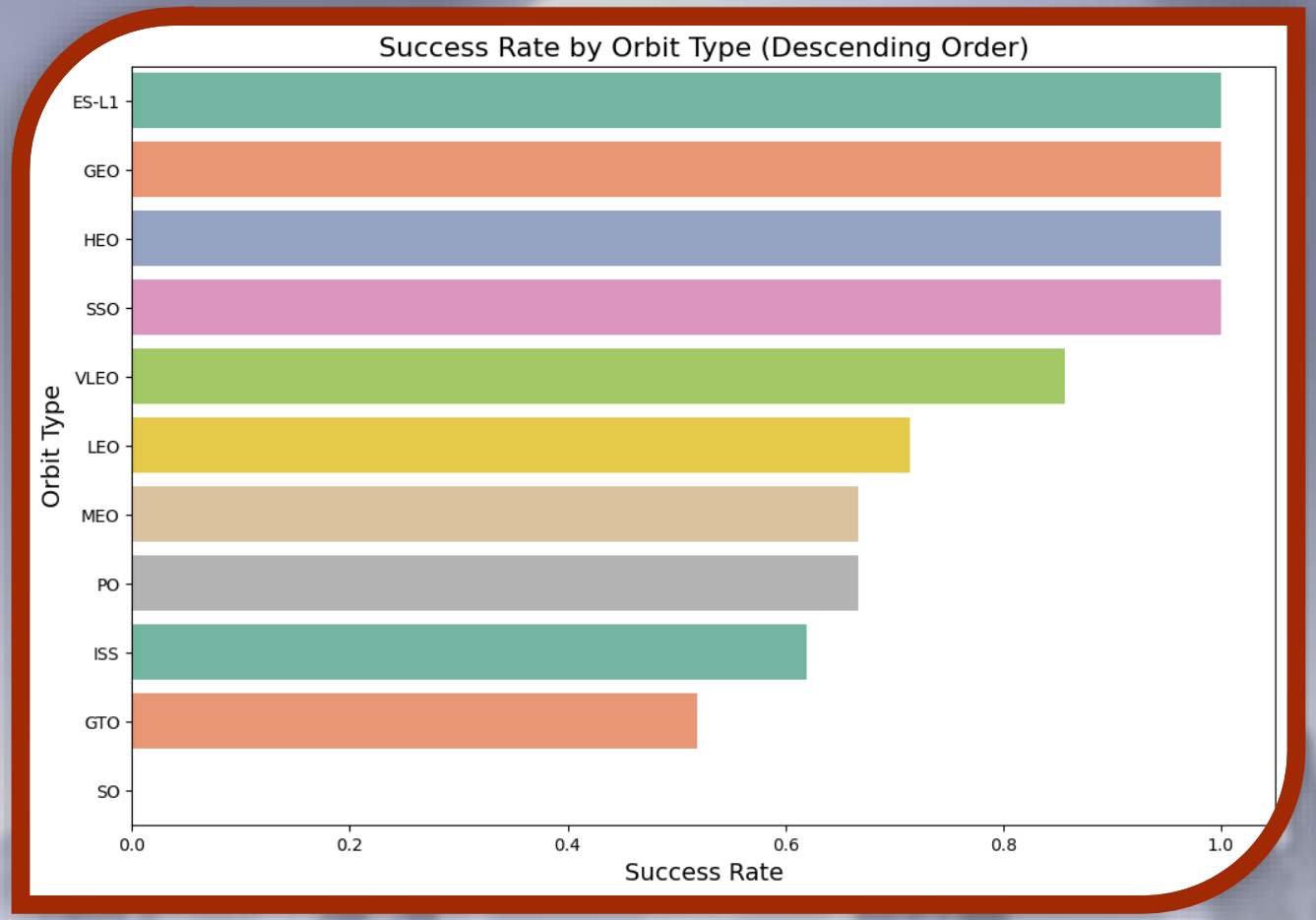
- The figure below indicates that the launch site CCAFS SLC 40 has seen the highest number of rocket launches in comparison to the other sites.
- Furthermore, it is demonstrated that the later flights from launch sites VAFB SLC 4E and KSC LC 39A had a higher success rate when compared to the earlier flights.

Payload vs. Launch Site



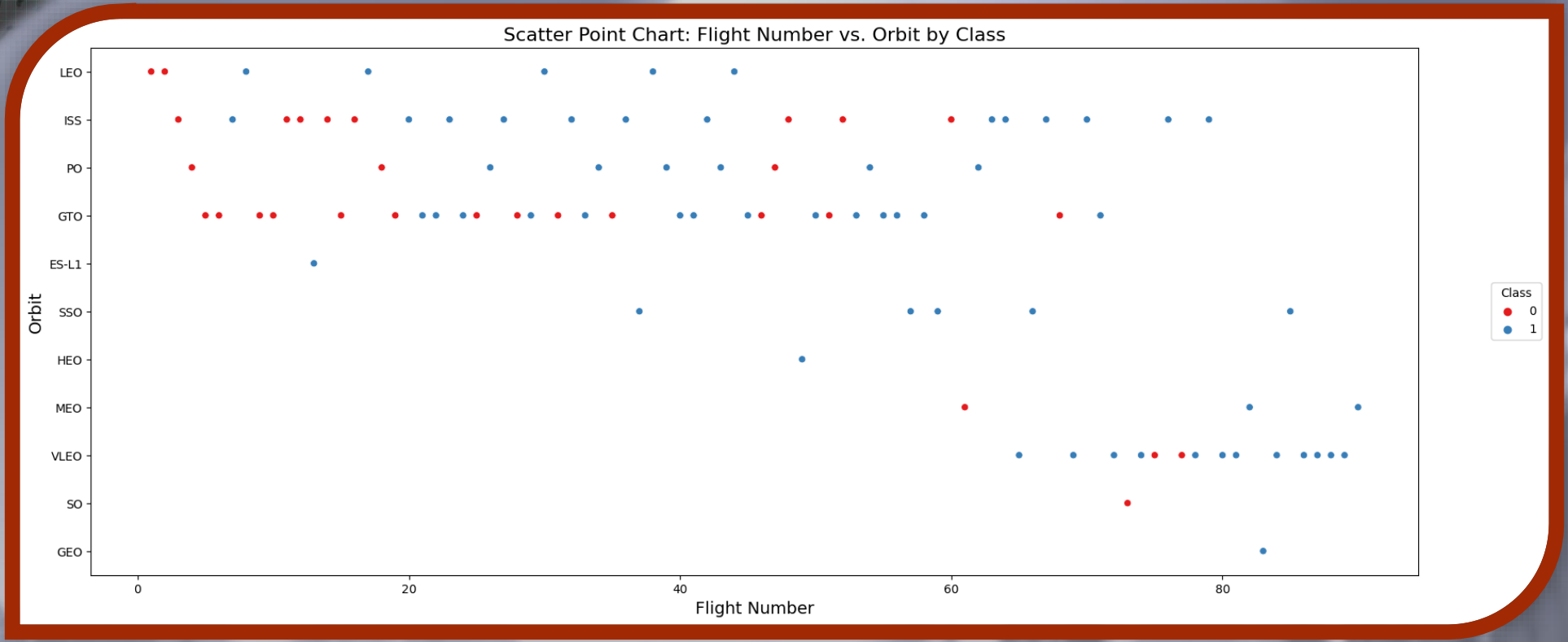
- Looking at the figure below, it is clear that at the VAFB-SLC 4E launch site, no rockets have been launched for heavy payloads exceeding 10,000kg.
- Additionally, it is apparent that the majority of rockets launched at all the launch sites carry payloads weighing less than 9,000kg.
- In comparison to VAFB-SLC 4E and KSC LC 39A, CCAFS SLC 40 shows a higher success rate for rockets launched with heavy payload masses of 14,000kg and 16,000kg.

Success Rate vs. Orbit Type



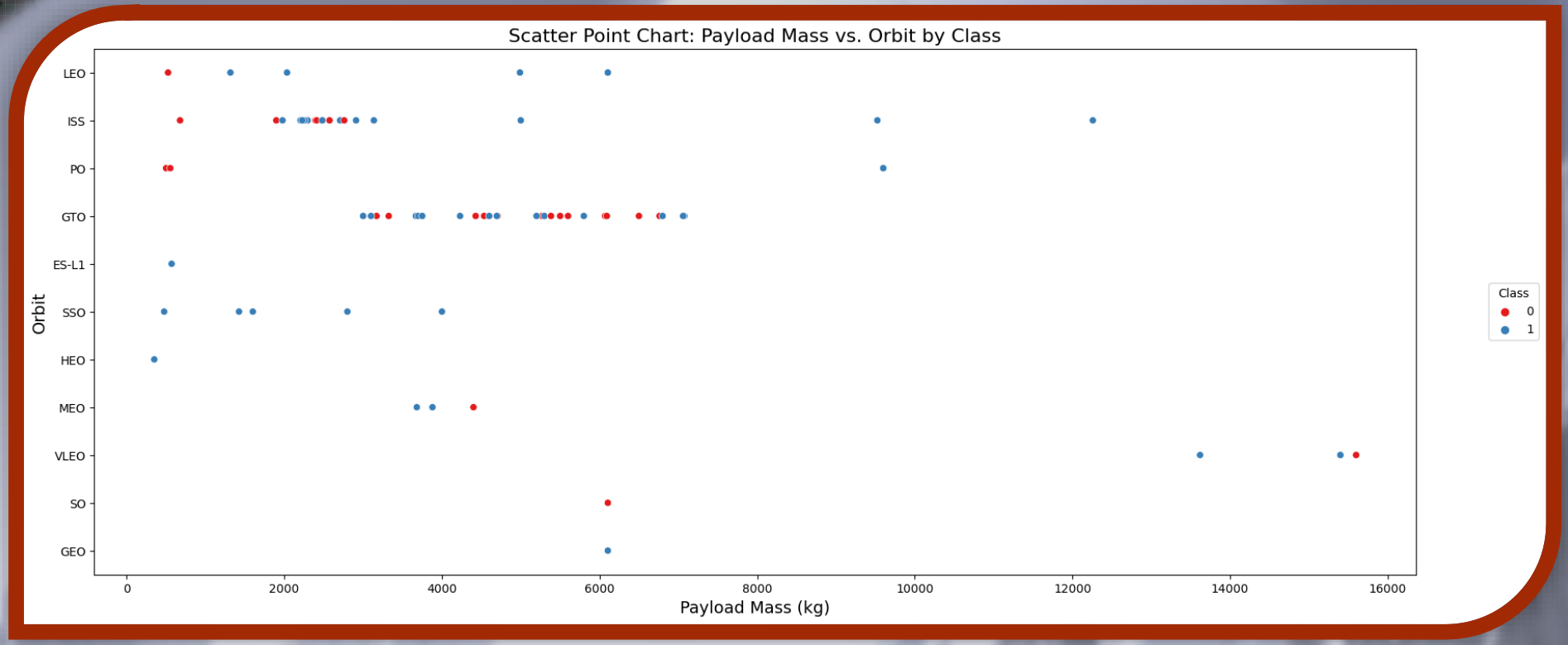
ES-L1, GEO, HEO, and SSO orbits had a perfect success rate of 100%, while VLEO did really well with over 80% success, and LEO also did great with over 70% success.

Flight Number vs. Orbit Type



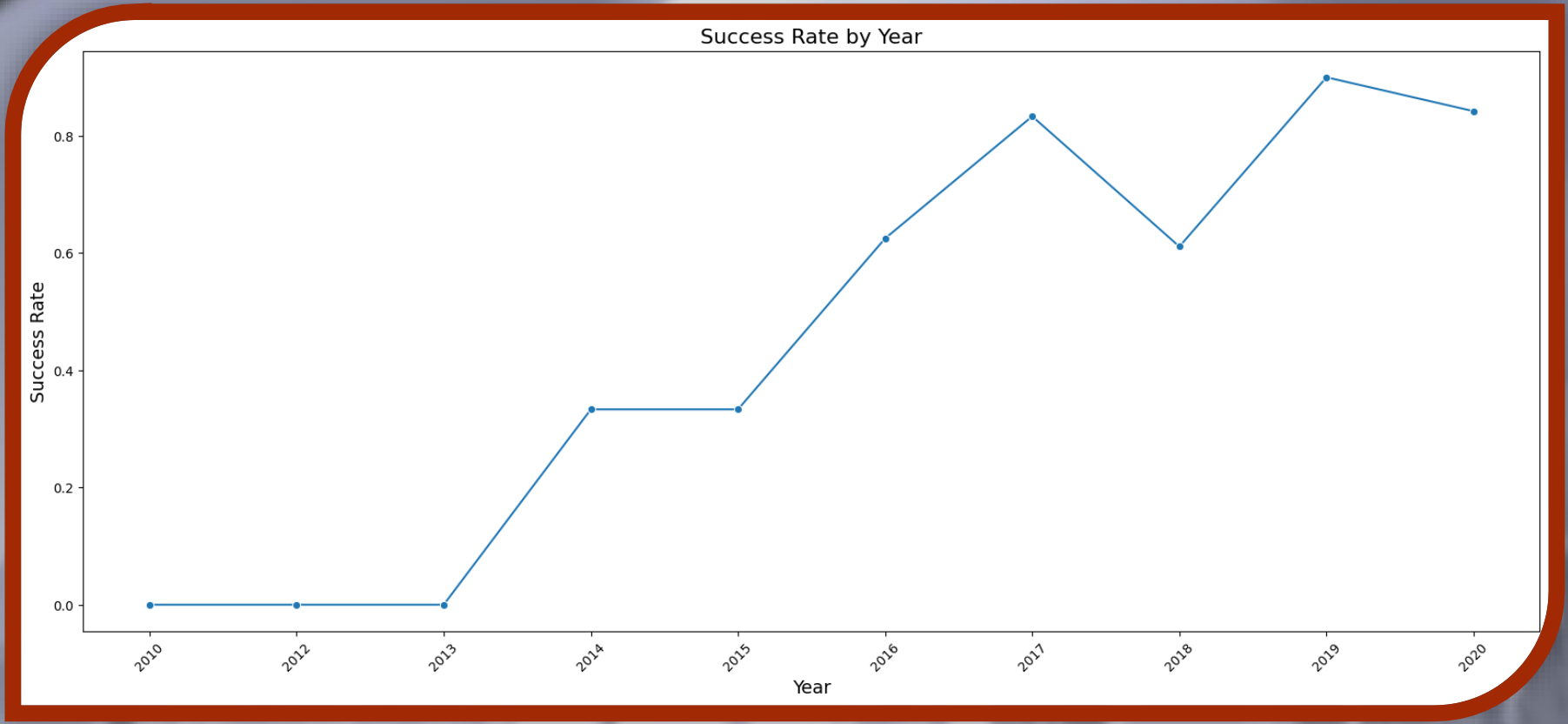
Looking at the chart provided, we can see that there were more rocket launches in LES ISS PO GTO and VLEO orbits. Additionally, it's clear that in the LEO orbit, success seems to be connected to the number of flights, while for other orbits, there doesn't appear to be a correlation between the number of flights and success.

Payload vs. Orbit Type



Rockets with big payloads do well when they're sent into PO, LEO, and ISS orbits. But if you send rockets into SSO and MEO orbits with smaller payloads, they also do pretty well. Now, GTO launches are a bit different because whether the payload is big or small, some rockets land successfully while others don't.

Launch Success Yearly Trend



The success rate has shown a significant increase since 2013, continuing to rise until 2020, possibly due to technological advancements, while the first three years (2010-2013) appear to have been focused on fine-tuning and technological enhancement.

All Launch Site Names

```
# Display the names of the unique launch sites
unique_launch_sites = %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
unique_launch_sites
```

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

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

The SQL table "SPACEXTBL" was created by utilizing the existing data frame, and to identify the unique launch sites, the "DISTINCT" keyword was applied to the corresponding column.

Launch Site Names Begin with 'CCA'

```
# Display 5 records where launch sites begin with 'CCA'
launch_sites_starting_with_CCA = %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
launch_sites_starting_with_CCA
```

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

The "CCA%" keyword was employed to retrieve launch site names starting with "CCA," and the "LIMIT 5" keyword was utilized to display just five records.

Total Payload Mass

```
# Display the total payload mass carried by boosters launched by NASA (CRS)
total_payload_mass_nasa_crs = %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
total_payload_mass_nasa_crs
```

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

TotalPayloadMass
45596

The total payload mass for customers with the name 'NASA (CRS)' was computed using the SUM function.

Average Payload Mass by F9 v1.1

```
# Display the average payload mass carried by booster version F9 v1.1
average_payload_mass_f9_v1_1 = %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
average_payload_mass_f9_v1_1
```

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

AveragePayloadMass
2928.4

The average payload mass carried by booster version F9 v1.1 was calculated using the AVG function.

First Successful Ground Landing Date

```
# List the date when the first successful landing outcome on a ground pad was achieved
first_successful_landing_date = %sql SELECT MIN(Date) AS FirstSuccessfulLandingDate FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
first_successful_landing_date
```

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

FirstSuccessfulLandingDate
2015-12-22

An SQL query was executed to identify the date of the first successful ground pad landing, which revealed that December 22, 2015, marked the initial successful ground landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
# List the names of boosters with success in drone ship and payload mass within the specified range
successful_drone_ship_boosters = %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 600
successful_drone_ship_boosters
```

Python

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

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

Using the keywords BETWEEN and AND, the names of boosters that had payload masses greater than 4000kg but less than 6000kg and had successfully landed on a drone ship were displayed, resulting in a total of 4 rockets being shown in the results.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

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

The total number of successful and failed missions was counted using the COUNT function, and the results indicate that there were 100 successful missions and 1 failed mission.

Boosters Carried Maximum Payload

```
# List the names of booster versions with the maximum payload mass
max_payload_booster_versions = %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
max_payload_booster_versions
```

```
 * 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

The boosters that carried the maximum payload were retrieved using a sub-query with the MAX function, and a total of 12 were found in the results.

2015 Launch Records

```
dates = []
for date in df["Date"]:
    if date[6:10] == "2015":
        dates.append(date)
df_2015 = df[df["Date"].isin(dates)]
df_2015 = df[(df["Landing_Outcome"] == 'Failure (drone ship)')]
failed_boosters = df_2015["Booster_Version"].to_list()
print(f"the boosters \n {failed_boosters} \n have failed to land with drone ships in 2015")
```

```
the boosters
 ['F9 v1.1 B1012', 'F9 v1.1 B1015', 'F9 v1.1 B1017', 'F9 FT B1020', 'F9 FT B1024']
 have failed to land with drone ships in 2015
```

Dates from the year "2015" are gathered and stored in a list called `dates`. The DataFrame `df` is then filtered to create `df_2015`, in which rows marked with "Failure (drone ship)" as the "Landing_Outcome" are retained.

Subsequently, "Booster_Version" values are extracted from `df_2015` and compiled into a list named `failed_boosters." Finally, a message is generated, indicating that drone ship landing failures were experienced by boosters in `failed_boosters` during the year 2015.

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

```
lower_date = df[df["Date"] == "04-06-2010"]
higher_date = df[df["Date"] == "16-03-2017"]
df_timed = df.iloc[0:31,]
df_timed['Landing_Outcome'].value_counts(ascending=False)
```

```
No attempt               10
Failure (drone ship)      5
Success (drone ship)      5
Controlled (ocean)        3
Success (ground pad)      3
Failure (parachute)       2
Uncontrolled (ocean)      2
Precluded (drone ship)    1
Name: Landing_Outcome, dtype: int64
```

Dates from the year "2015" are gathered and stored in a list called `dates`. The DataFrame `df` is then filtered to create `df_2015`, in which rows marked with "Failure (drone ship)" as the "Landing_Outcome" are retained.

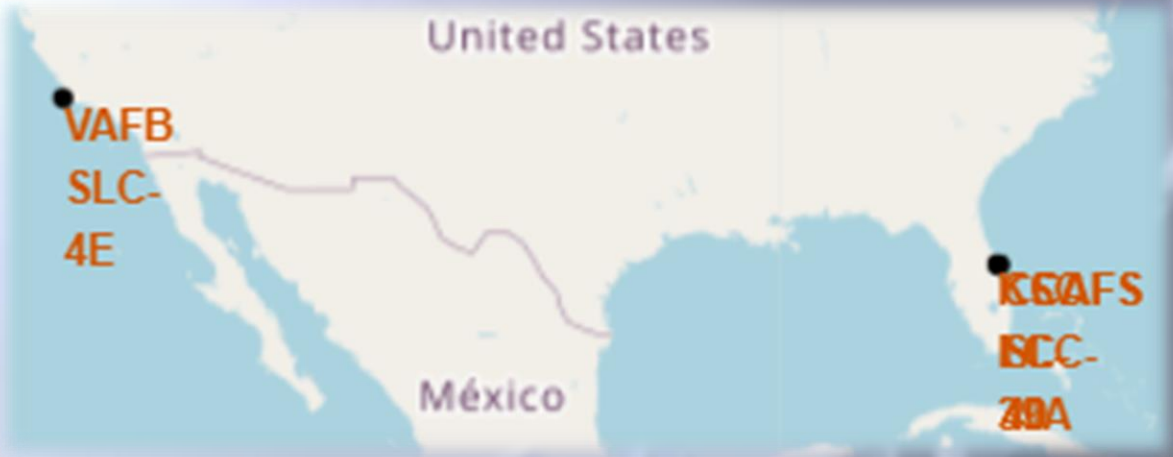
Subsequently, "Booster_Version" values are extracted from `df_2015` and compiled into a list named `failed_boosters." Finally, a message is generated, indicating that drone ship landing failures were experienced by boosters in `failed_boosters` during the year 2015.

Section 3

Launch Sites Proximities Analysis

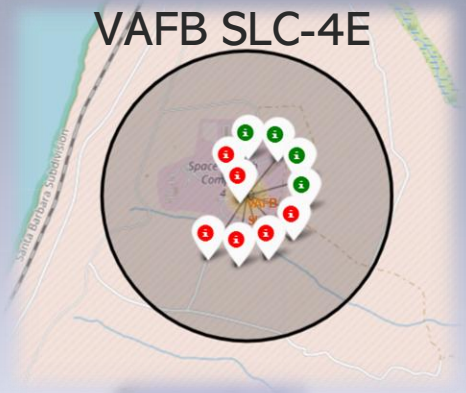
Launch Site Locations for Space X Falcon 9

The figures illustrate that all launch sites are situated within coastal cities of the United States of America.

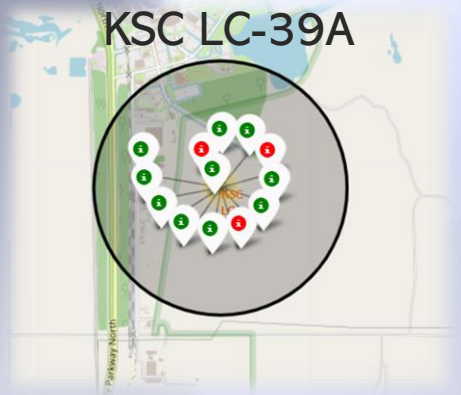


Launch Outcomes for Space X Falcon 9

VAFB SLC-4E

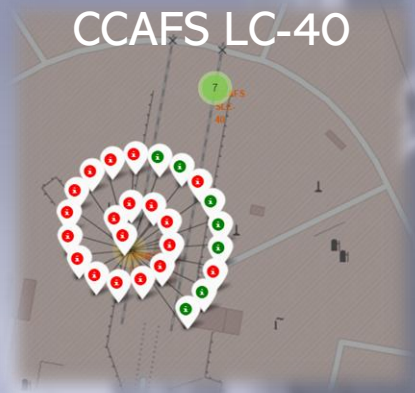


KSC LC-39A



CCAFS SLC-40

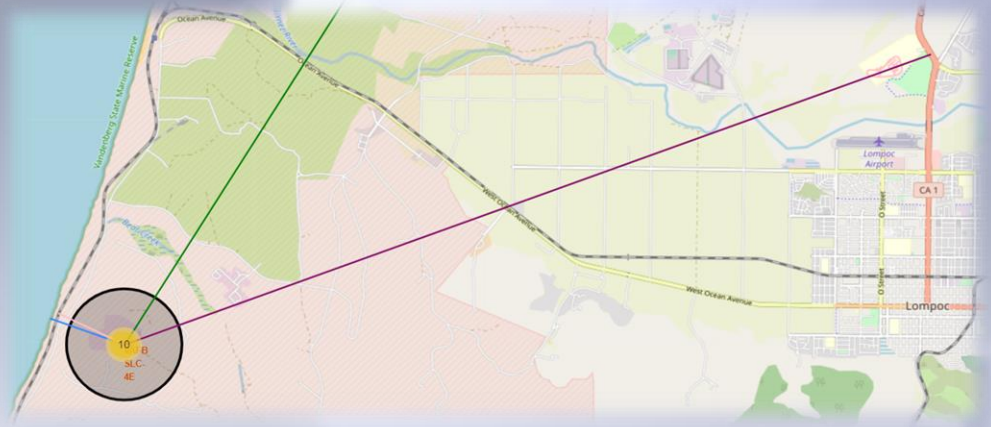
CCAFS LC-40



The figures display the results of launches from different sites. Red icons represent unsuccessful attempts, while green icons signify successful ones.

Launch site distance from coastline, cities, railways and highways

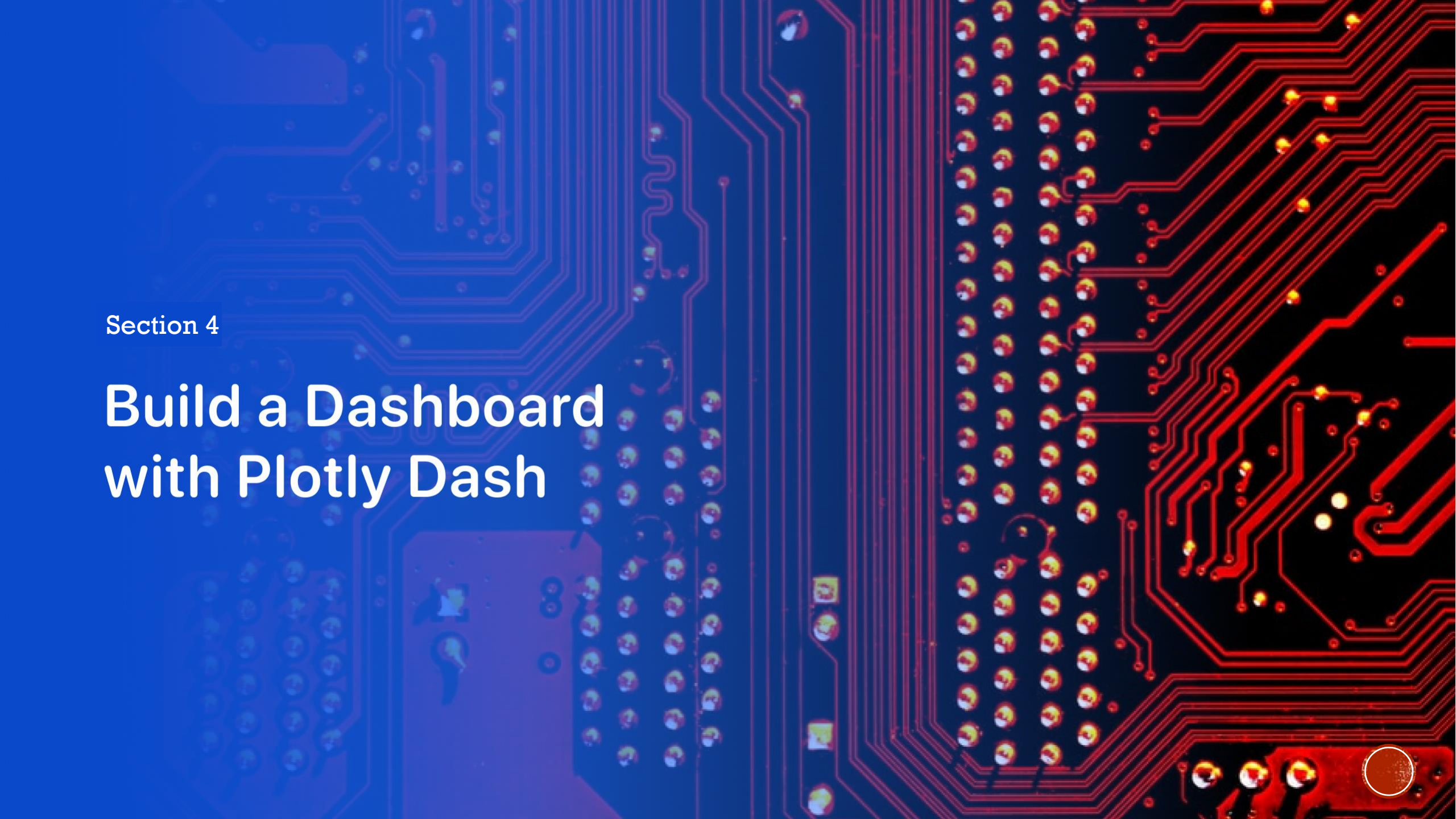
Distance to the nearest city
(Green line)



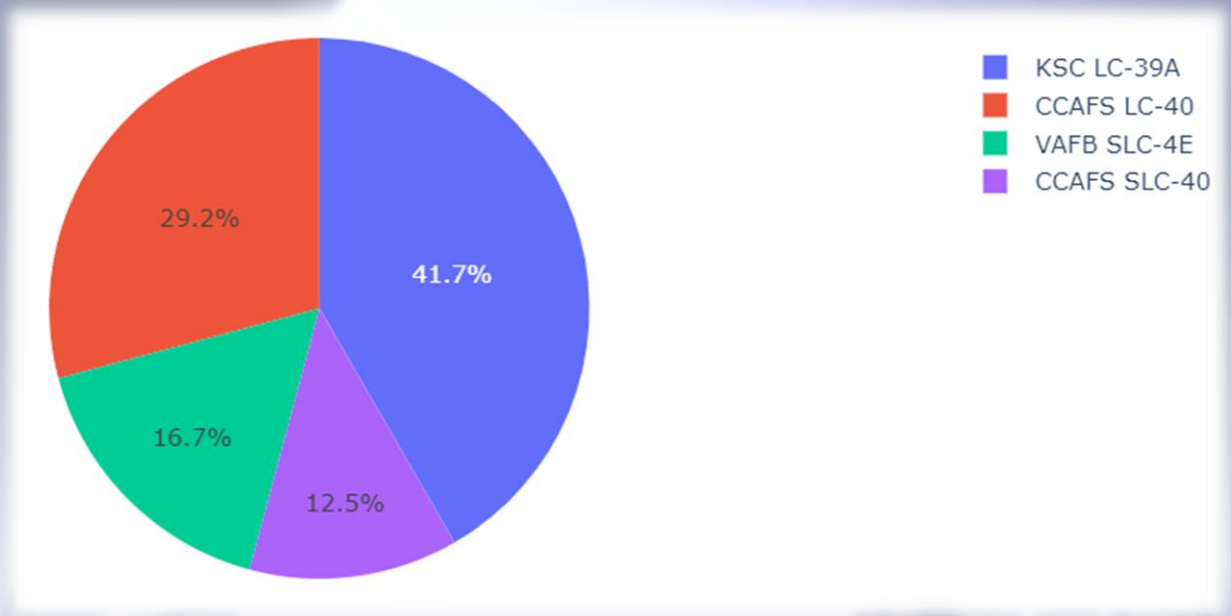
Distance to the nearest highway
(Purple line)

Distance to the nearest railway
(Pink line)

The figure shows that the launch site VAFB SLC 4E is close to the coast and a railway. In contrast, some highways don't have this proximity. Additionally, launch sites are far from major cities.


Section 4
Build a Dashboard with Plotly Dash

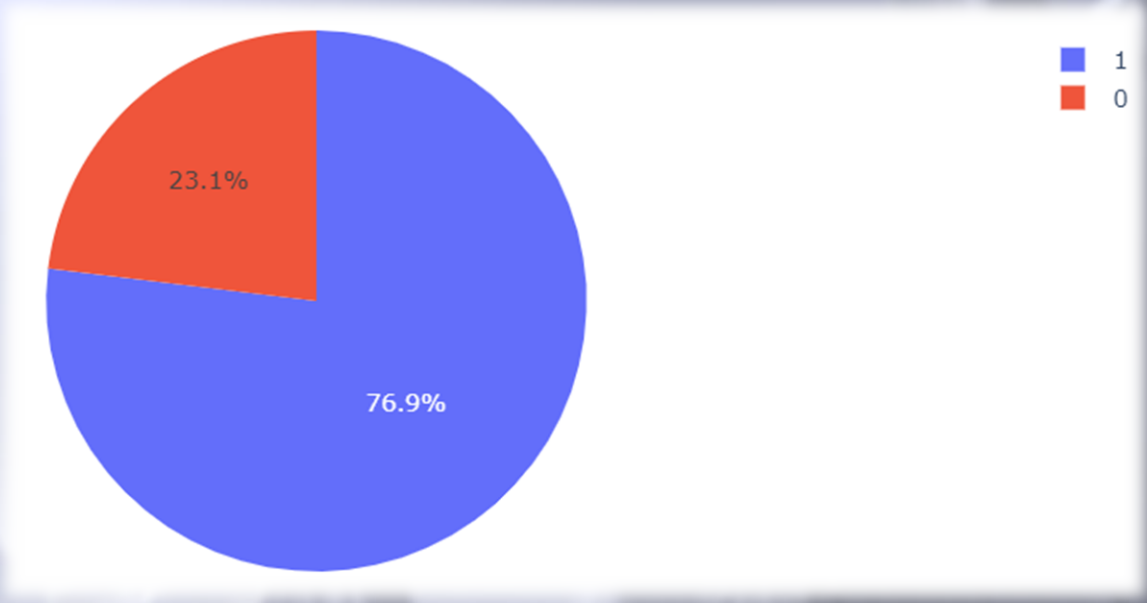
Pie Chart of Success Count for all Sites



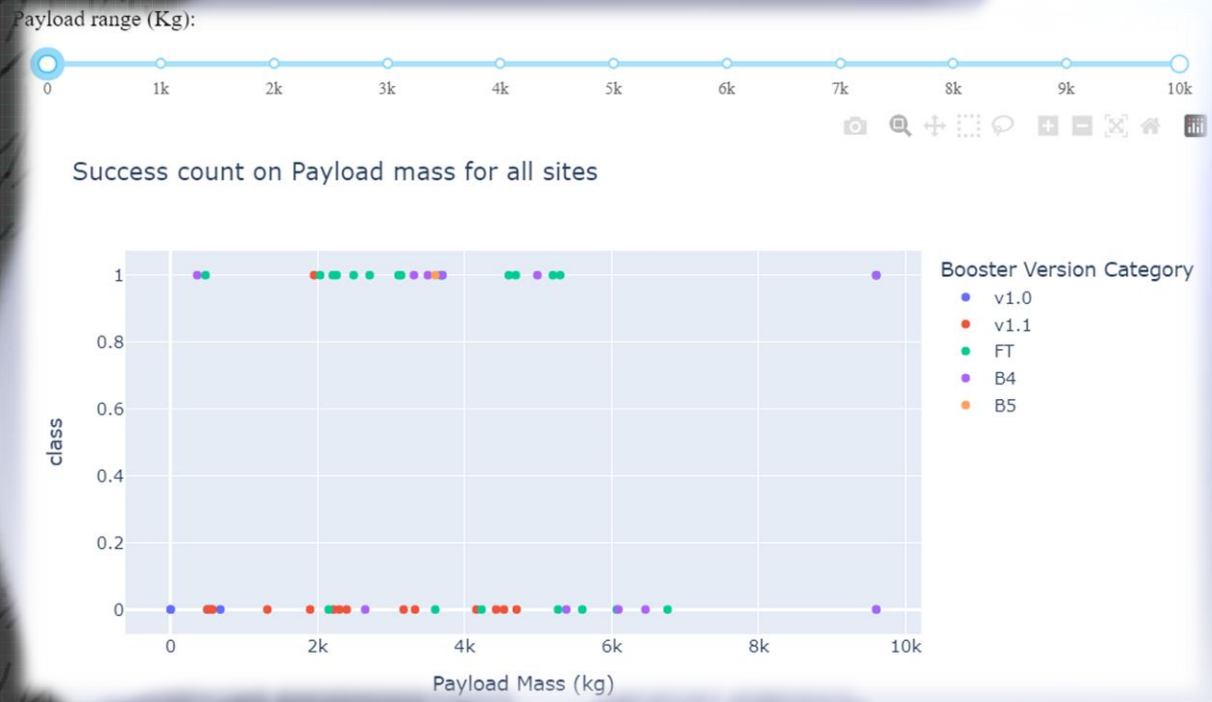
This figure demonstrates that the highest success rate, accounting for approximately 41.7% of the overall success ratio among other sites, is achieved by KSC LC-39A.

Pie Chart of Success Count for all Sites

This figure reveals that KSC LC-39A boasts the highest success ratio of approximately 76.9%, surpassing the other sites.
- 73.1% for CCAFS LC-40
- 60% for VAFB SLC-4E
- 57.1% for CCAFS SLC-40



Payload vs Launch outcome for all sites

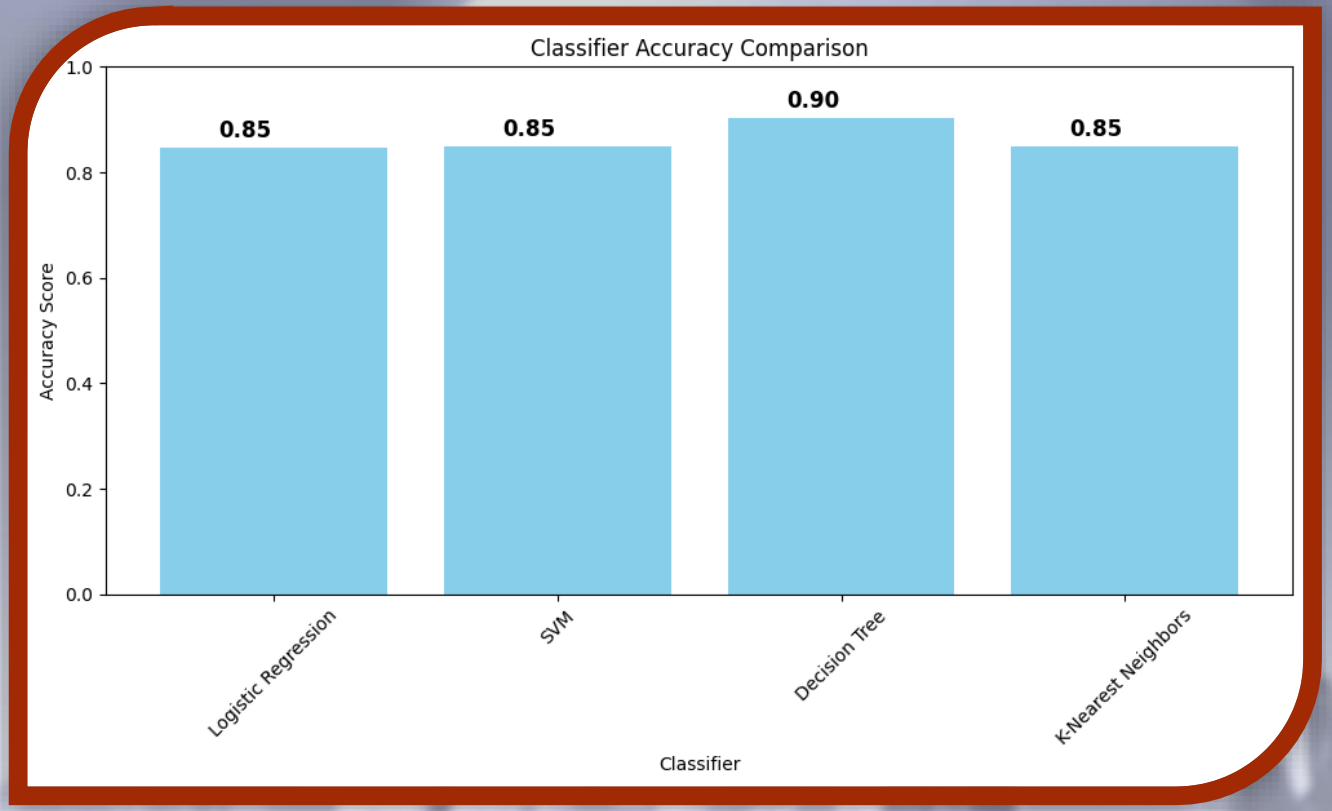


These figures indicate that Booster version FT achieves the highest success rate when its payload mass falls within the range of approximately 700kg to 5,500kg, while rockets carrying payload masses exceeding 5,500kg exhibit lower success rates, highlighting that the heavier the payload, the lower the likelihood of a successful outcome.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

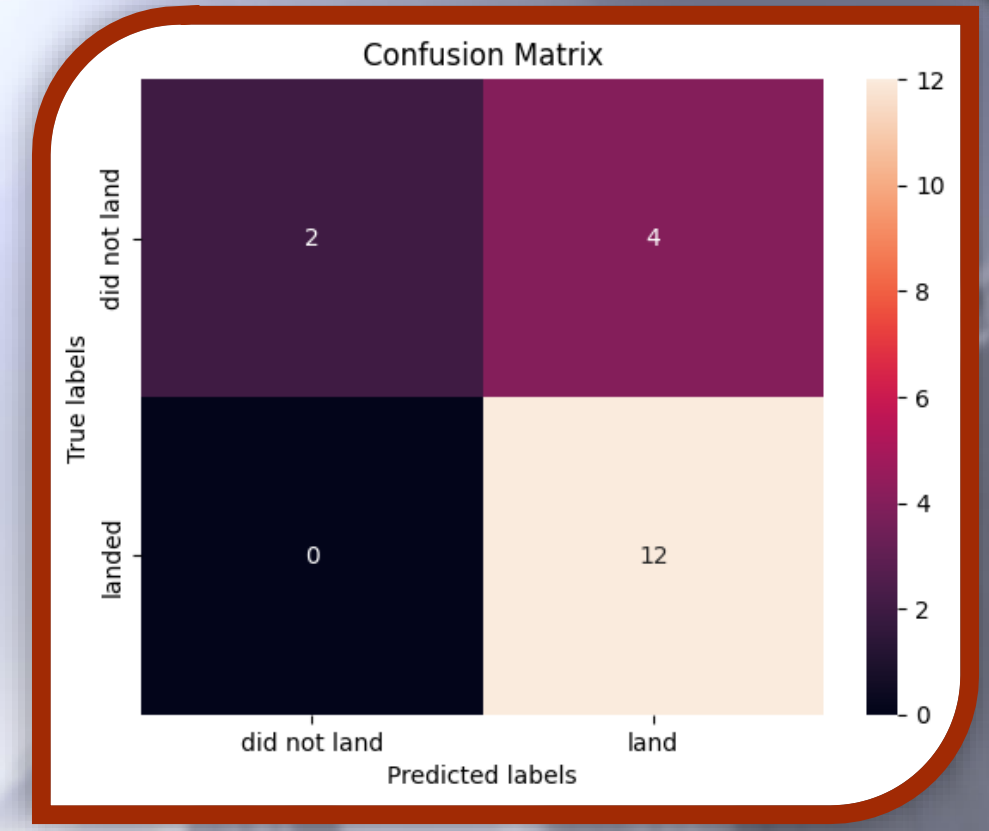


In the figure's bar chart, the best performance was achieved by the Decision tree classifier, which recorded an accuracy score of 90%.

Confusion Matrix

Upon the dataset being divided into training and test sets,18 test samples were generated. Within this limited test set, accurate predictions were made by the decision tree classifier for 12 instances of successful landings (12 True Positives) and 2 cases of non-landings (2 True Negatives).

The classifier registered zero false negatives, signifying that it did not make any incorrect predictions regarding successful landings. Nonetheless, it did exhibit four false positives by inaccurately predicting successful outcomes for four observations.



Conclusions

- To ensure mission success, consider the payload's weight, as rockets with lighter payloads have a better chance of succeeding.
- Also, think about the type of orbit when planning launches. Rockets sent into orbits like VLEO, ES-L1, GEO, HEO, and SSO tend to have higher success rates compared to others.
- Launch sites are chosen carefully; they're near the coast for easy retrieval and far from busy areas like highways and cities to reduce harm in case of a failure.
- In recent years, missions have been more successful, especially with later launches.
- Among different methods, decision tree classifiers performed the best, with an impressive 90% success rate, making them a good choice for predicting landing outcomes.

Thank you!