IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Suprit Patil
21/09/2024

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection via API & Web Scraping
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Created Interactive Map with Folium
  - Dashboards with Ploty Dash
  - Predictive Analysis
- Summary of all results
  - Identified top launch sites and key factors affecting landing success.
  - Logistic Regression gave the highest accuracy.
  - Predicted first stage landing with over 80% accuracy.

# Introduction

- Project background and context
  - The goal of this project is to predict whether the Falcon 9 first stage will land successfully. According to SpaceX's website, the launch of the Falcon 9 rocket cost $62 million. Other providers cost up to 165 million dollars each. SpaceX can reuse the first stage, which explains the price difference. The cost of a launch can be estimated by determining whether the stage will land. This information is valuable for companies looking to compete with SpaceX for rocket launches.
- Problems you want to find answers
  - What are the key characteristics of a successful or unsuccessful landing?
  - How do the relationships between rocket variables impact landing success?
  - What conditions will enable SpaceX to achieve the highest landing success rate?

Section 1

# Methodology

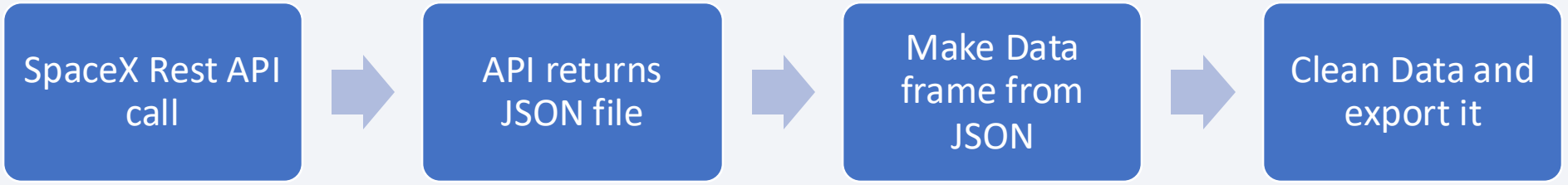
# Methodology

## Executive Summary

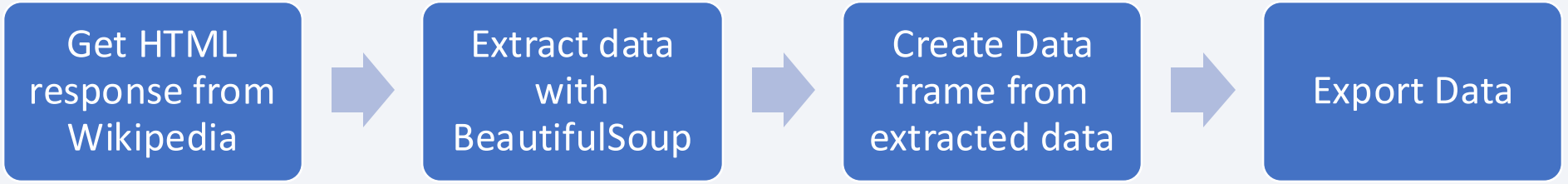
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropped unnecessary columns and handles null values
  - One Hot encoding for ML models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

- Data for rocket, launches was collected by pulling data from API and web scrapping of Wikipedia.
- Rocket, launches, payload information from Space X REST API




- Launches, landing & payload information from web scrapping Wikipedia



# Data Collection – SpaceX API

- Used the SpaceX API to retrieve Falcon 9 launch data, such as dates, locations, and landing results.
- Parsed the JSON response and converted it to a pandas data frame.
- Key metrics collected include launch site, date, booster version, landing outcome, payload mass, and orbit.
- Data was saved in CSV format for further processing and analysis.


Data Collection using API - Link

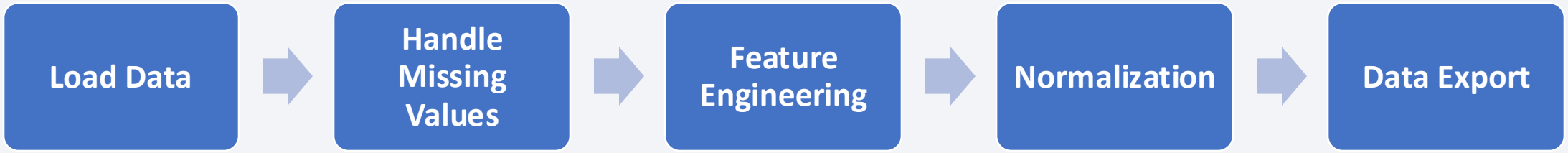
# Data Collection - Scraping

- Used Python's BeautifulSoup library to extract SpaceX launch data from HTML pages.
- Data extraction involved parsing specific launch data, such as mission names, dates, and outcomes.
- Data cleaning involved removing duplicates and irrelevant information to ensure that the data was clean for analysis.
- The scraped data was exported to a structured CSV format for future use.


Data Collection Web Scrapping - Link

# Data Wrangling

- Imported raw SpaceX data from CSV files.
- Identified and addressed missing values by filling or dropping rows as needed.
- Created new features from existing columns and found cases where the booster failed to land stated below:
  - Successful Landings: True Ocean, True RTLS, True ASDS.
  - Failed Landings: False Ocean, False RTLS, False ASDS.
  - Label Transformation: Converted outcomes into binary categories—1 for success and 0 for failure.
- Scaled numerical features enhance model performance.
- The cleaned and processed data was saved for future modelling and analysis.






Data Wrangling - Link

# EDA with Data Visualization

Scatter Graphs:

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass

Scatter plots show relationship between variables. This relationship is called the correlation.

- Bar Graph: Success rate vs. Orbit
- Bar graphs show the relationship between numeric and categoric variables.

- Line Graph: Success rate vs. Year
- Line graphs show data variables and their trends.
- Line graphs can help to show global behaviour.
- and make prediction for unseen data.

EDA with Data Visualization - Link

# EDA with SQL

We performed several SQL queries to analyse the dataset:

- Retrieved unique launch sites and listed five sites starting with 'CCA'.
- Calculated total and average payload mass for NASA (CRS) and F9 v1.1 boosters, respectively.
- Identified the first successful ground pad landing and boosters with drone ship success carrying 4000-6000 kg payloads.
- Counted successful and failed missions.
- Found the boosters with the maximum payload and ranked successful landings between June 2010 and March 2017.

EDA with SQL - Link

# Build an Interactive Map with Folium

- The Folium map is centered on NASA Johnson Space Center, Houston, Texas, with key markers:
- Red circles with labels at NASA and launch sites.
- Clustered points for multiple data at the same location.
- Markers indicating successful (green) and unsuccessful (red) landings.
- Markers and lines showing distances from launch sites to key locations (railways, highways, coastlines, cities). These map objects help visualize launch sites, surroundings, and landing success rates, making it easier to understand the data and problem at hand.

Interactive Map with Folium- Link

# Build a Dashboard with Plotly Dash

- The dashboard includes dropdown, pie chart, range slider, and scatter plot components.
- Dash_core_components.Dropdown enables users to select either a specific launch site or all launch sites.
- The pie chart displays the success and failure rates for the launch site selected using the dropdown component (plotly.express.pie).
- The Rangeslider (dash_core_components.RangeSlider) allows users to select a payload mass within a fixed range.
- The scatter chart (plotly.express.scatter) visualizes the relationship between two variables, specifically Success and Payload Mass.

Dashboard with Ploty Dash - Link

# Predictive Analysis (Classification)

**Data Preparation:**

- Load and normalize dataset.
- Split data into training and test sets.

**Model Preparation:**

- Select algorithms and set GridSearchCV parameters.
- Train models with training data.

**Model Evaluation:**

- Retrieve optimal hyperparameters.
- Compute accuracy on test data and plot Confusion Matrix.

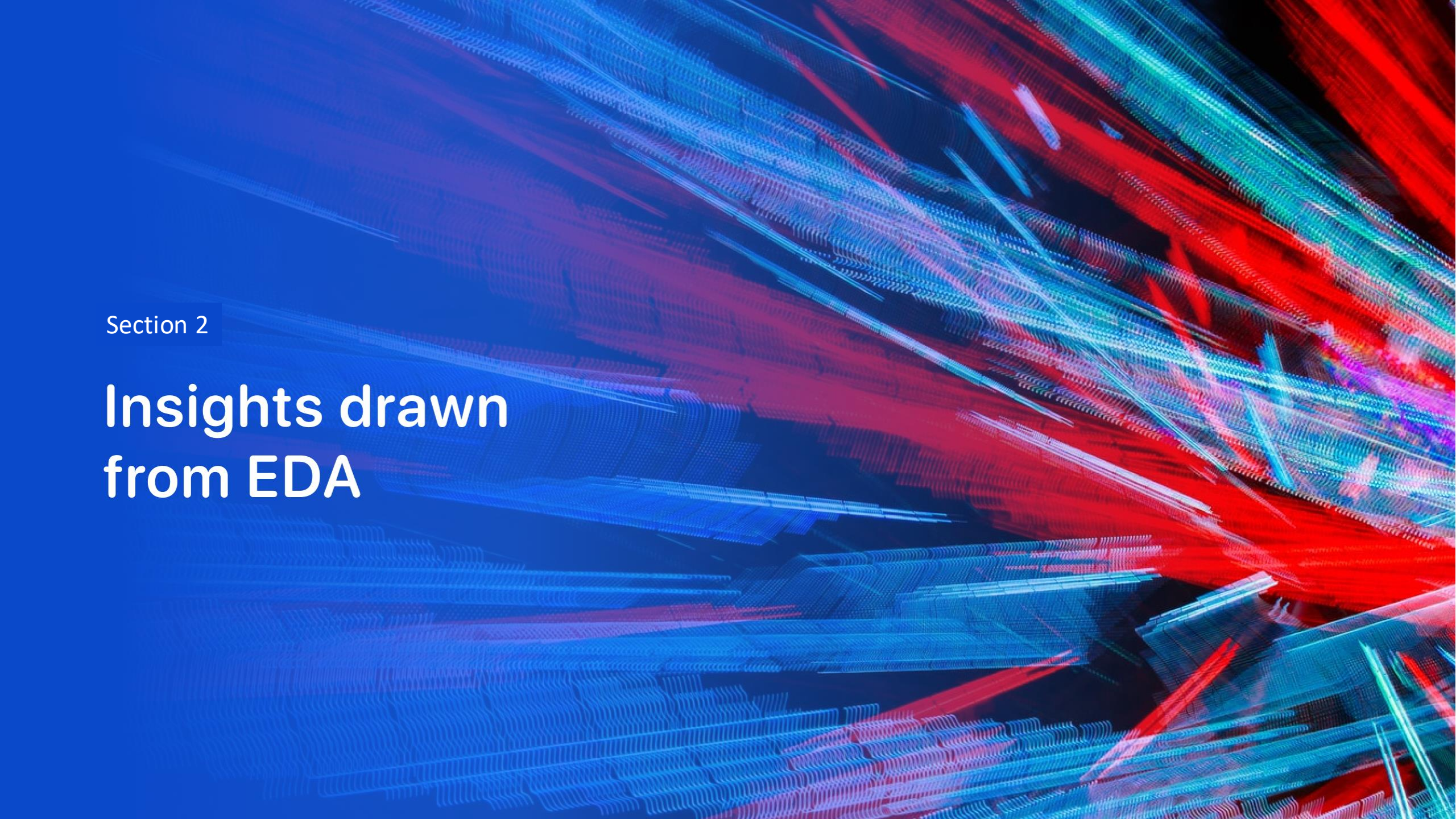
**Model Comparison:**

- Compare models by accuracy and select the best-performing one (details in Notebook).

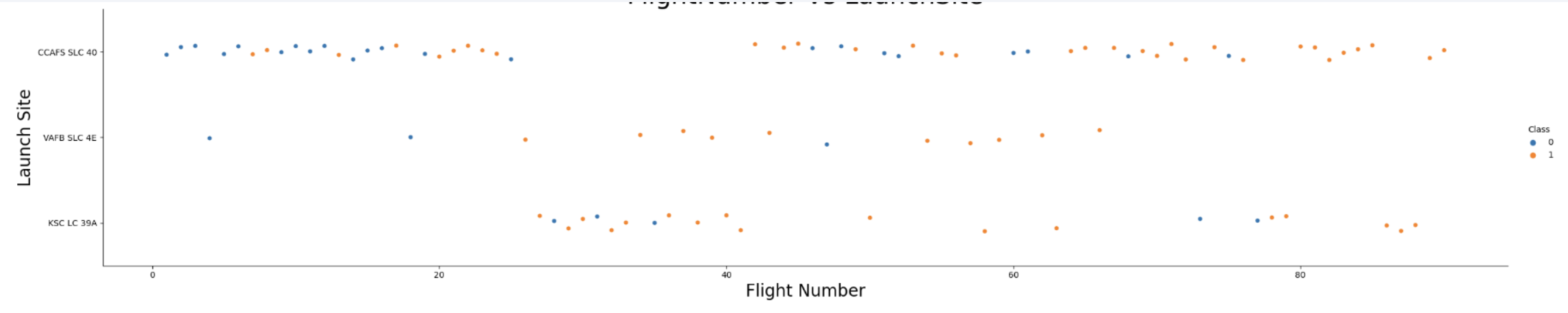
Predictive Analysis - Link

# Results

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

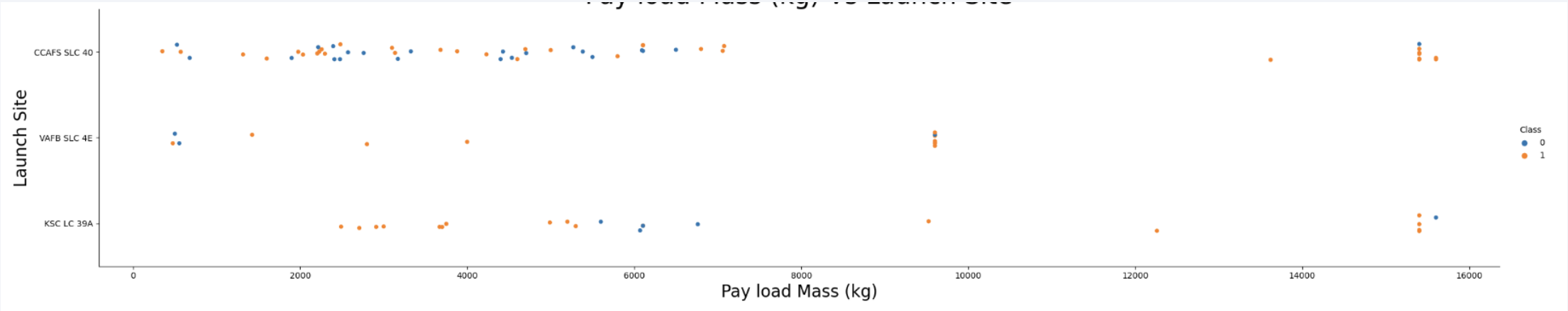

Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site



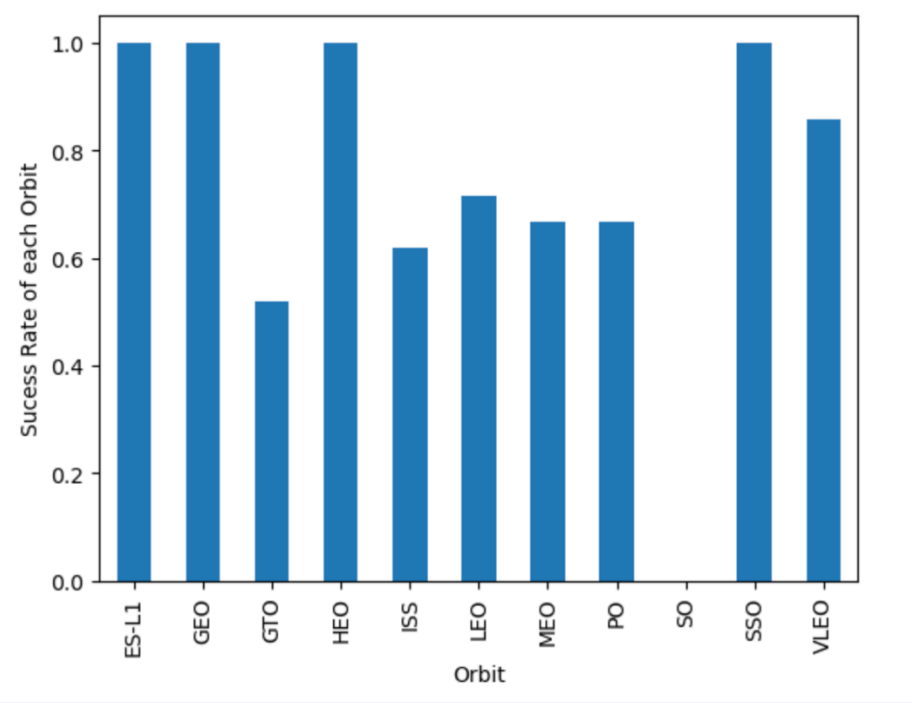

- We observe that, for each site, the success rate is increasing.

# Payload vs. Launch Site



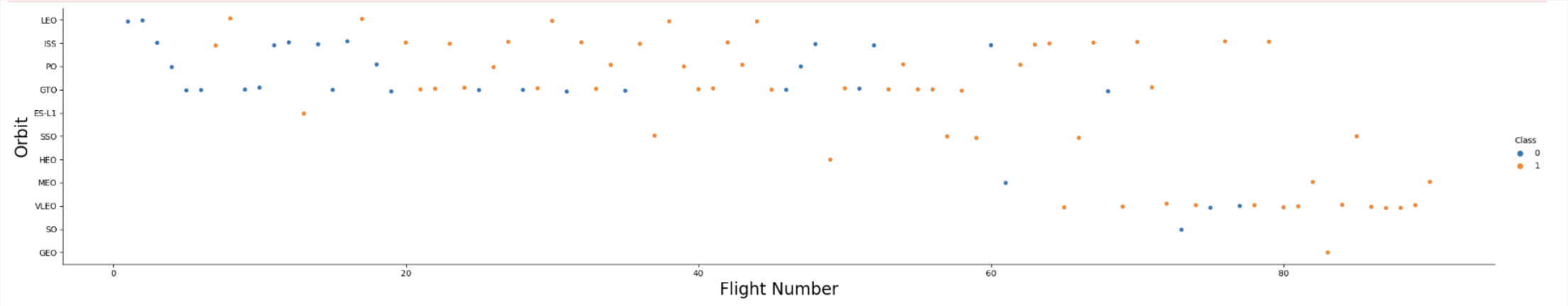

- Depending on the launch site, a heavier payload may be required for a successful landing. However, a heavy payload can cause landing failures.

# Success Rate vs. Orbit Type



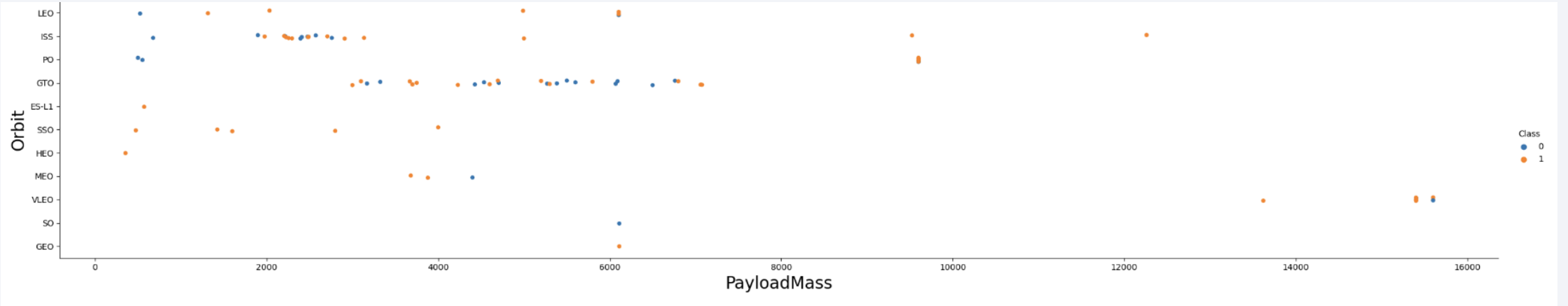

- This plot shows the success rates for various orbit types. ES-L1, GEO, HEO, and SSO exhibit the highest success rate.

# Flight Number vs. Orbit Type



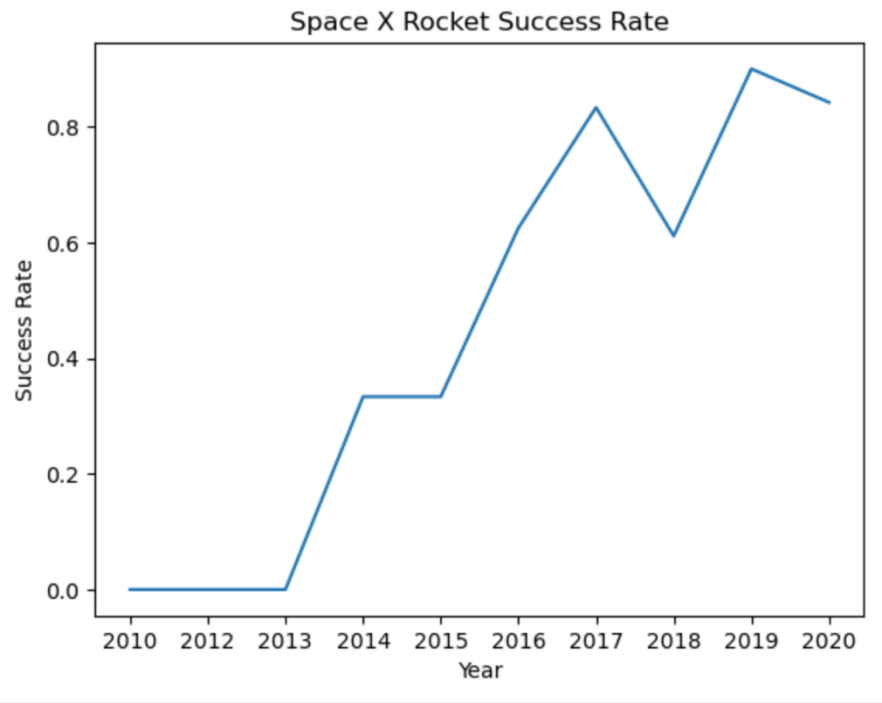

- We can see that the success rate for the LEO orbit increases with the number of flights. For certain orbits, such as GTO, there is no correlation between success rate and number of flights. The high success rate of certain orbits, such as SSO or HEO, may be attributed to knowledge gained from previous launches in other orbits.

# Payload vs. Orbit Type




- The weight of payloads significantly impacts launch success rates in specific orbits. For example, heavier payloads increase the success rate of the LEO orbit. Reducing the payload weight for a GTO orbit improves launch success rates.

# Launch Success Yearly Trend




- Since 2013, the success rate of Space X rockets has been increasing.

# All Launch Site Names

SQL query:

```
In [18]: %sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

Results:

Out[18]:

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

- The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

# Launch Site Names Begin with 'CCA'

SQL query:

```
%sql SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Results:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_ |
|---|---|---|---|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (p |
| 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 (p |
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | N |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | N |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | N |

- The WHERE clause is followed by a LIKE clause, which filters launch sites that contain the substring CCA. LIMIT 5 displays 5 results from filtering.

# Total Payload Mass

SQL query:

```
In [20]: %sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
```

Results:

Out[20]:

| SUM("PAYLOAD_MASS__KG_") |
|---|
| 45596 |

- The query returns the total payload mass for all NASA (CRS) customers.b

# Average Payload Mass by F9 v1.1

SQL query:

```
In [21]: %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
 * sqlite:///my_data1.db
Done.
```

Results:

Out[21]:

| AVG("PAYLOAD_MASS__KG_") |
|---|
| 2534.6666666666665 |

- This query calculates the average of all payload masses with the booster version containing the substring F9 v1.1.

# First Successful Ground Landing Date

SQL query:

```
In [31]: %sql SELECT MIN("Date") FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%Success%'
 * sqlite:///my_data1.db
Done.
```

Results:

Out[31]:

| MIN("Date") |
| --- |
| 2015-12-22 |

- This query selects the oldest successful landings.
- The WHERE clause filters the dataset to only include records where landing was successful. Using the MIN function, we find the record with the oldest date.

# Successful Drone Ship Landing with Payload between 4000 and 6000

SQL query:

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

Results:

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

- This query returns booster versions with successful landings and payload masses ranging from 4000 to 6000 kg. The WHERE and AND clauses filter the data.

# Total Number of Successful and Failure Mission Outcomes

SQL query:

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE
"MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, (SELECT
COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE
'%Failure%') AS FAILURE
```

Results:

| SUCCESS | FAILURE |
|---|---|
| 100 | 1 |

- With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

# Boosters Carried Maximum Payload

SQL query:

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (SELECT
max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Results:

| 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 |

- To filter data, we used a subquery and the MAX function to return only the heaviest payload mass. The main query uses subquery results to return a unique booster version (SELECT DISTINCT) with the highest payload mass.

# 2015 Launch Records

SQL query:

**%sql** SELECT substr("DATE", 6, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' and substr("DATE",0,5) = '2015'

Results:

| MONTH | Booster_Version | Launch_Site |
|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 |

- This query returns the month, booster version, launch site, and landing date for unsuccessful landings in 2015. Use the substr function to get the month or year of a given date. Substr(DATE, 6, 2) displays month. Substr(DATE,0,5) displays the year.

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

SQL query:

```
In [36]:
%sql SELECT "LANDING_OUTCOME", COUNT("LANDING_OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '2010-06-04' and "DATE" <= '2017-03-20' and "LANDING_OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING_OUTCOME" \
ORDER BY COUNT("LANDING_OUTCOME") DESC

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

Results:

Out[36]:

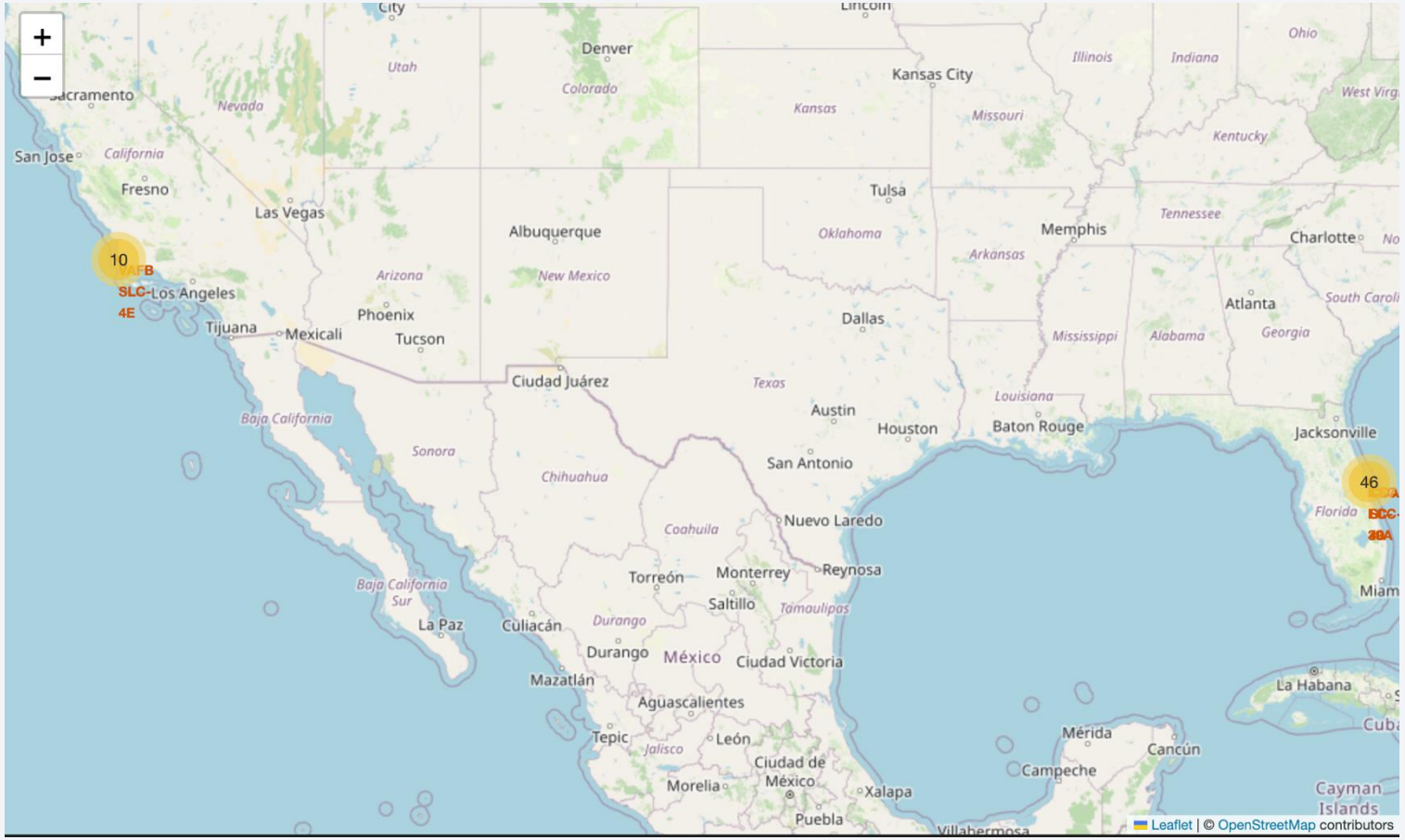
| Landing_Outcome | COUNT("LANDING_OUTCOME") |
|---|---|
| Success (drone ship) | 5 |
| Success (ground pad) | 3 |

- This query returns landing outcomes and counts for missions that were successful between 2010-06-04 and 2017-03-20. The GROUP BY clause organizes results by landing outcome, while ORDER BY COUNT DESC displays results in decreasing order.

Section 3

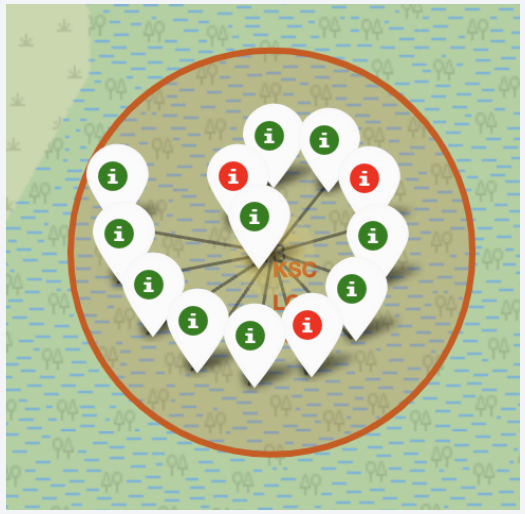
# Launch Sites Proximities Analysis

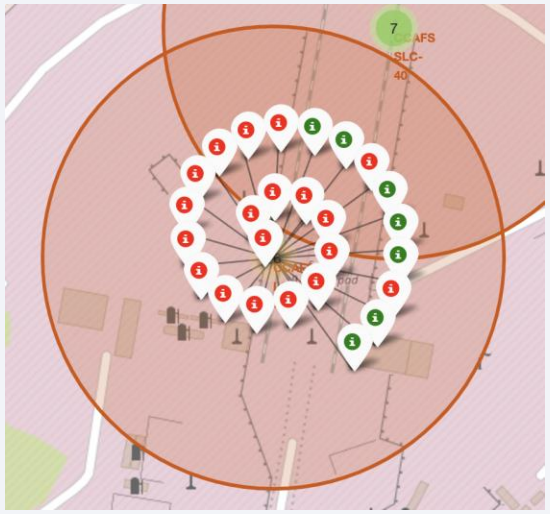
# Folium map – Ground stations

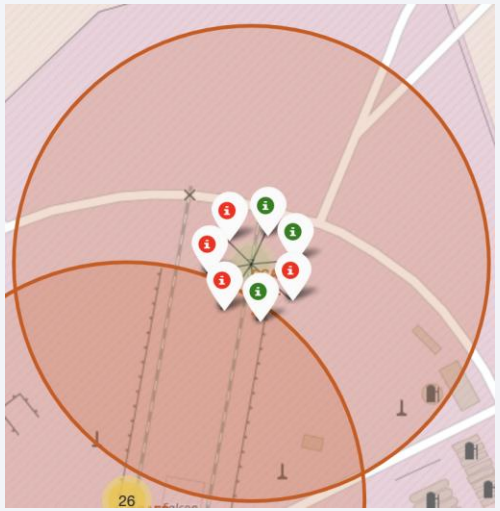



- We see that Space X launch sites are located on the coast of the United States

# Folium map – Color Labeled Markers



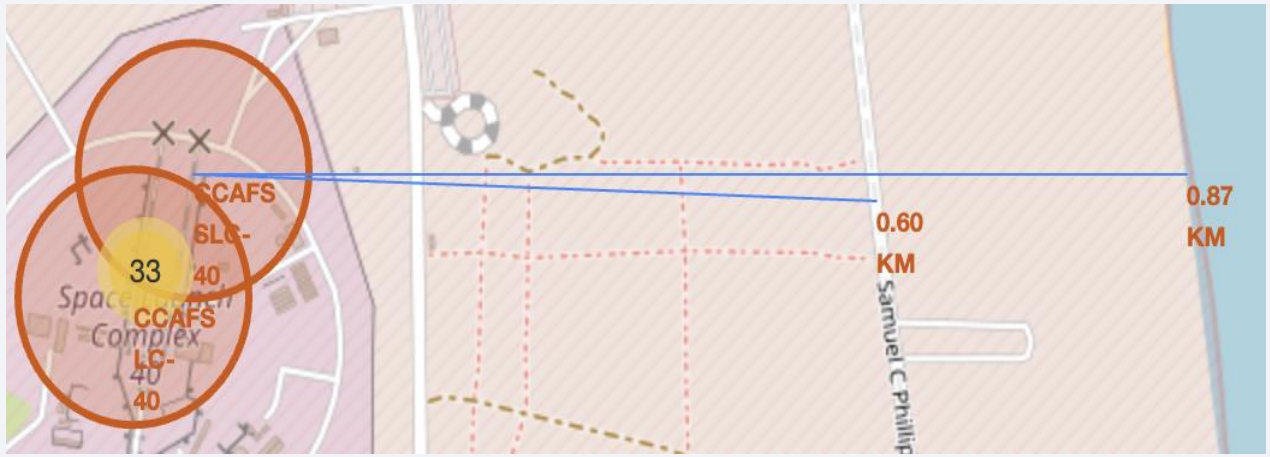





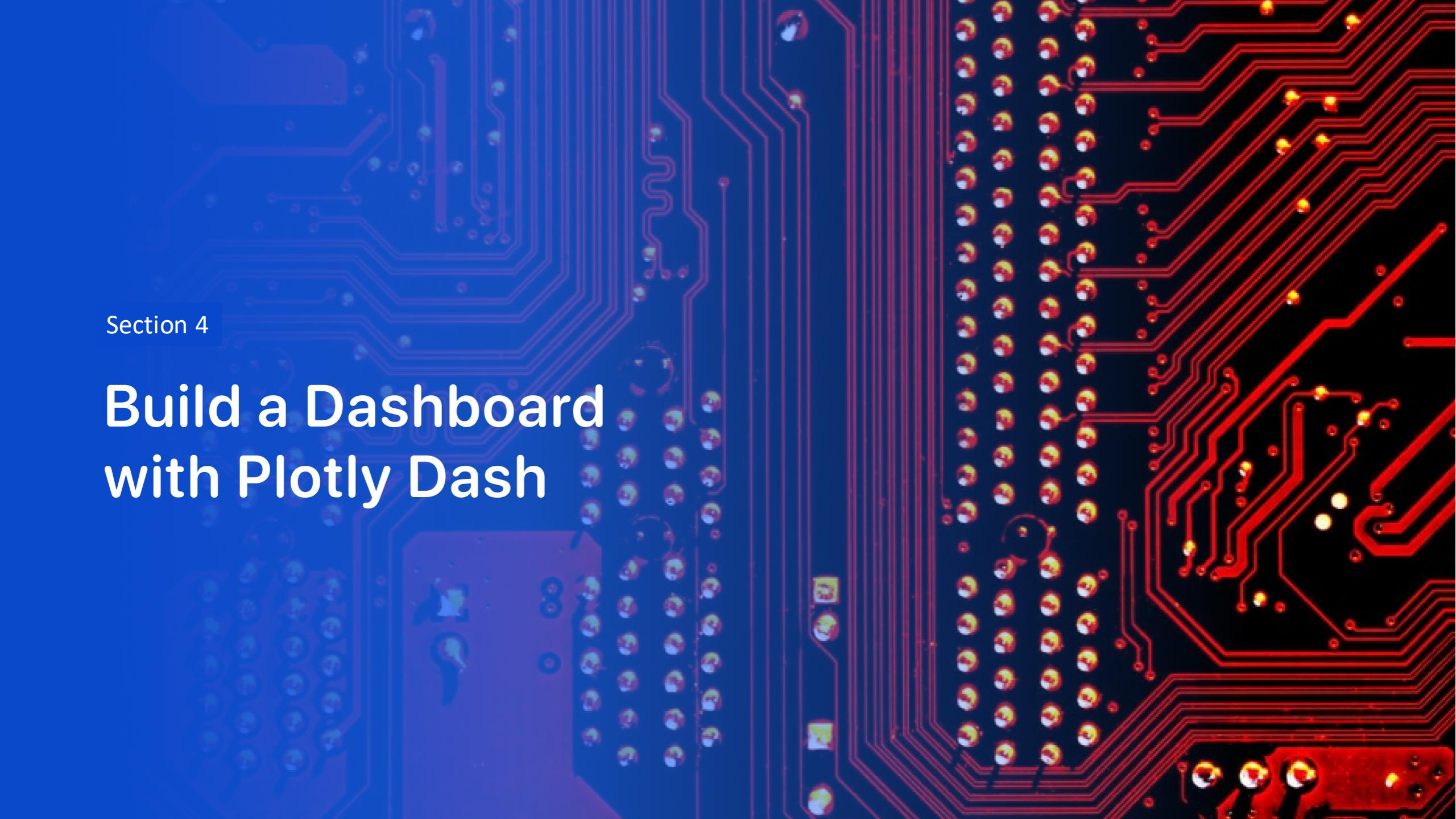
- The green marker represents successful launches. The red marker represents unsuccessful launches. KSC LC-39A has higher launch success rates.

# Folium Map – Distances between CCAFS SLC-40 and its proximities

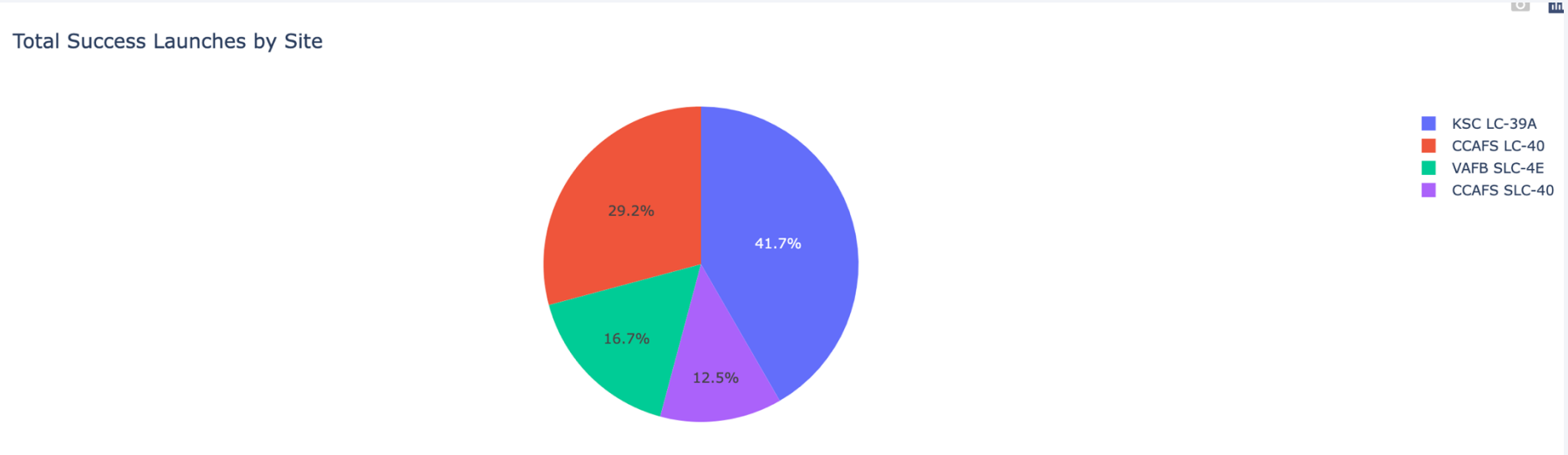





- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40 in close proximity to highways ? Yes
- Is CCAFS SLC-40 in close proximity to coastline ? Yes
- Do CCAFS SLC-40 keeps certain distance away from cities ? No

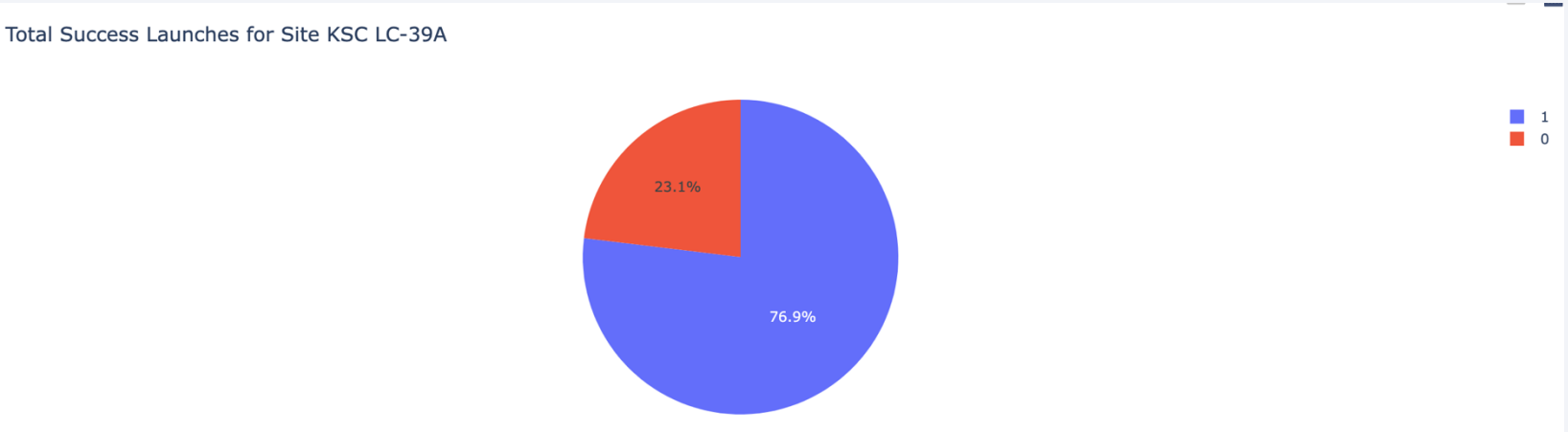

Section 4
Build a Dashboard with Plotly Dash

# Dashboard – Total success by Site



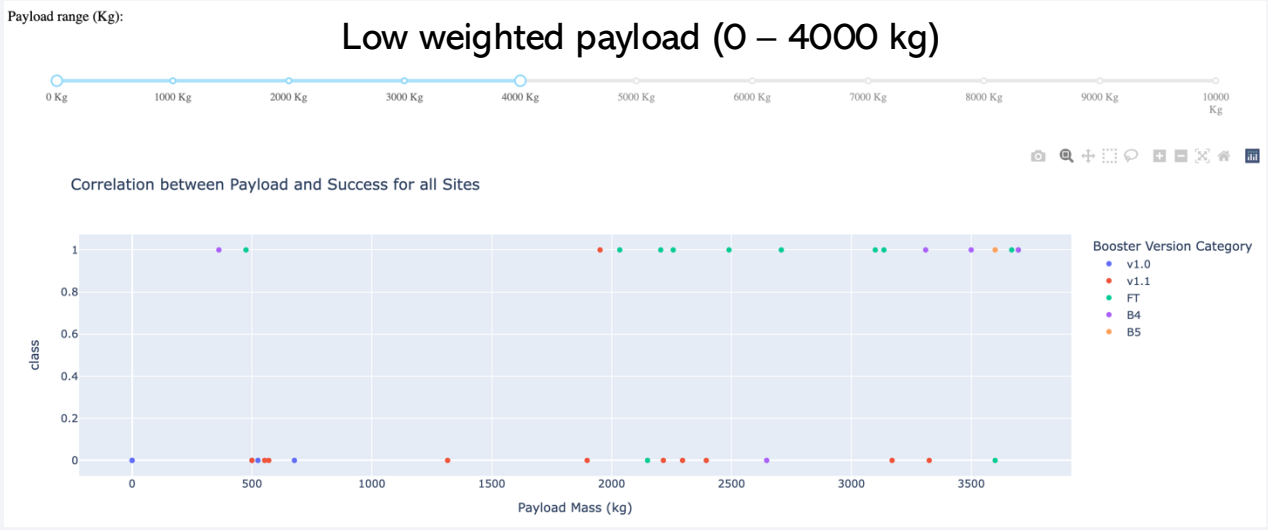

- We see that KSC LC-39A has the highest success rate in launches.

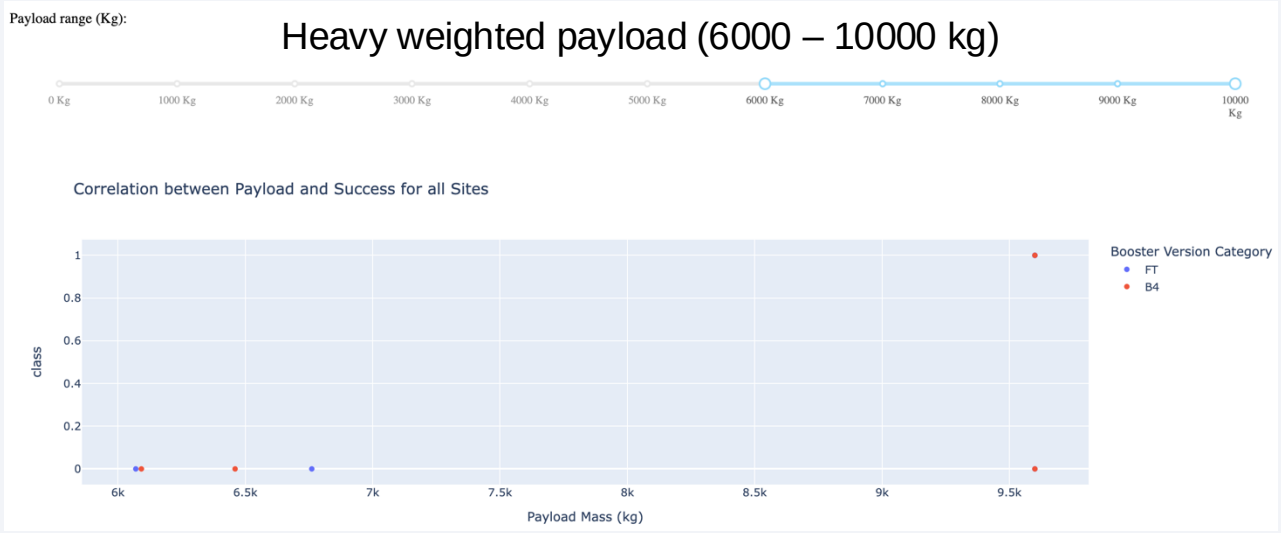
# Dashboard – Total success launches for Site KSC LC-39A




- We see that KSC LC-39A has a 76.9% success rate and a 23.1% failure rate.

# Dashboard – Payload mass vs Outcome for all sites with different payload mass selected




- Low weighted payloads have a better success rate than the heavy weighted payloads.

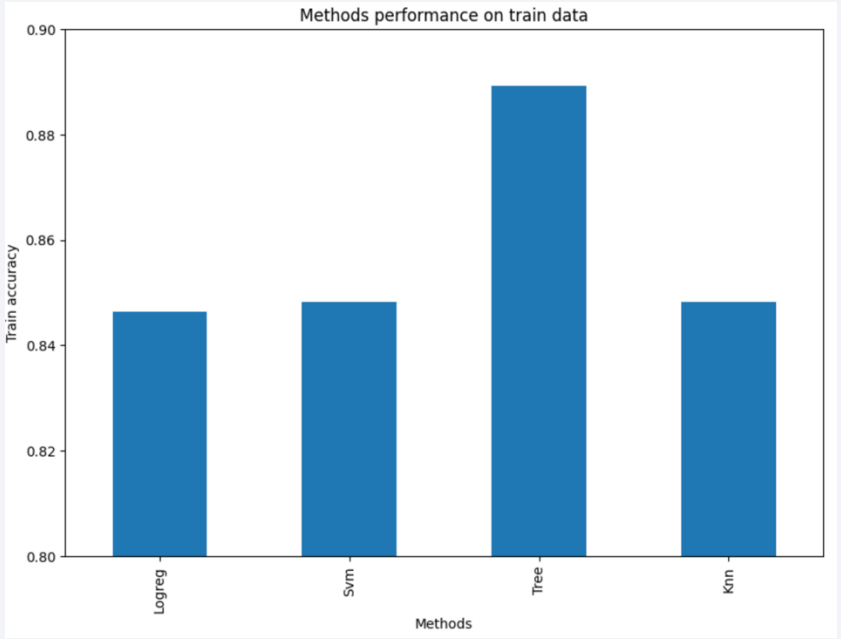
Section 5

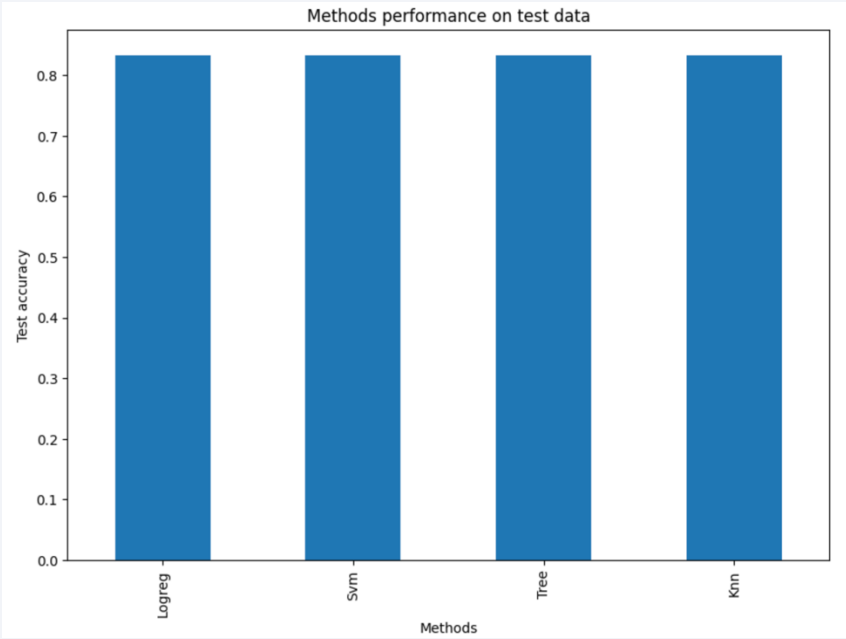
# Predictive Analysis (Classification)

# Classification Accuracy

Out[37]:

| | Accuracy Train | Accuracy Test |
|---|---|---|
| **Tree** | 0.889286 | 0.833333 |
| **Knn** | 0.848214 | 0.833333 |
| **Svm** | 0.848214 | 0.833333 |
| **Logreg** | 0.846429 | 0.833333 |

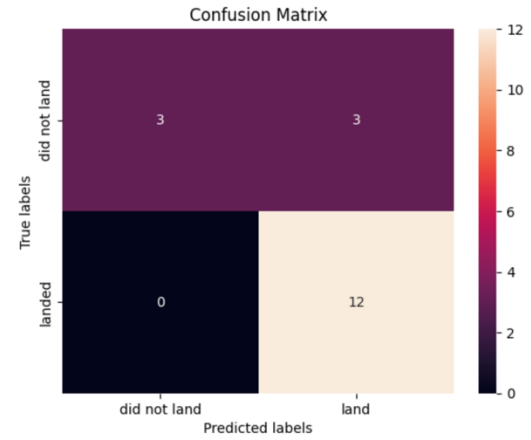


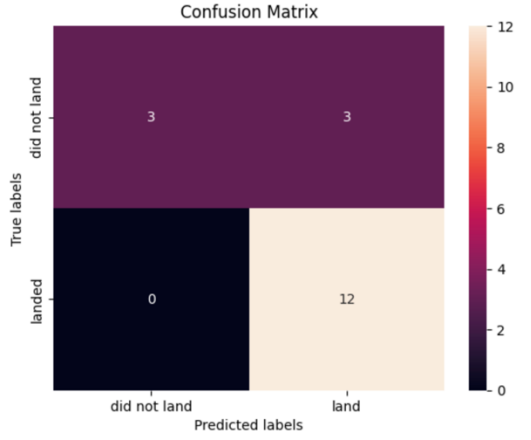

- For accuracy testing, all methods performed similarly. We could collect more test data to decide between them. But if we really had to choose one right now, we'd go with the decision tree.

# Confusion Matrix

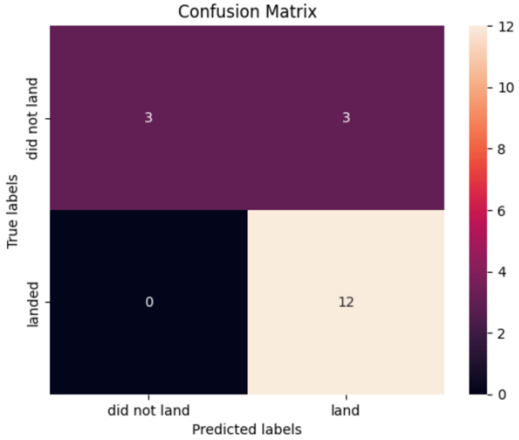
## Logistic Regression



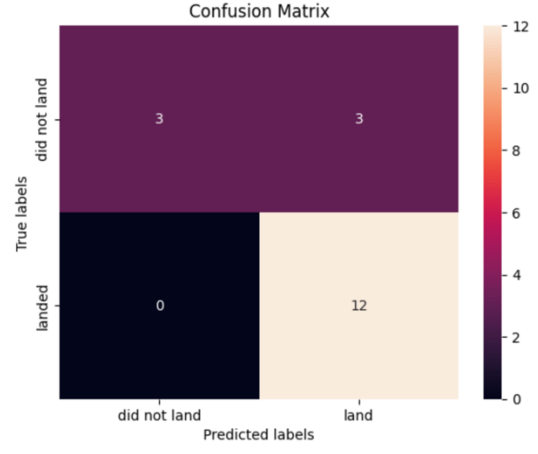

## Decision Tree




## KNN




## SVM




- Because all the models test accuracy is equal their confusion matrices are also the same, indicating that the models make similar types of errors. One major issue is the high number of false positives, in which the model predicts a successful landing when it fails.

# Conclusions

- Mission success is influenced by factors such as launch site, orbit, and number of previous launches, as experience grows with each launch.
- The orbits with the highest success rates are GEO, HEO, SSO, and ES-L1.
  Payload mass influences success; lighter payloads outperform heavier ones.
- The superiority of some launch sites (for example, KSC LC-39A) remains unknown in the absence of additional data such as atmospheric conditions.
- The Decision Tree Algorithm was chosen as the best model because of its higher train accuracy, despite identical test accuracies across models.

# Thank you!