Winning Space Race with Data Science

Radhika
Choudhary
07/08/2023

Outline

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

Executive Summary

- Overview
  - Data collection involved utilizing API and web scraping methods to gather the necessary data. This raw data was subjected to Exploratory Data Analysis (EDA), initially with Data Visualization, and further exploration with SQL queries. The visualization of geographical data was achieved through interactive maps using Folium library. Additionally, the results were presented through interactive dashboards created with Plotly Dash, and predictive analysis was conducted to draw meaningful insights.
- Main findings
  - The Exploratory Data Analysis (EDA) yielded valuable insights, and the results were presented through interactive maps and dashboards. Moreover, the predictive analysis provided further valuable outcomes, creating a comprehensive summary of the study.

Table of Contents

Introduction

- Project background and context
  - The primary objective of this project is to develop a predictive model for determining the likelihood of a successful landing of the Falcon 9 first stage. SpaceX, a leading space exploration company, claims that their Falcon 9 rocket launch costs approximately 62 million dollars, while other providers charge upwards of 165 million dollars for each launch. The cost difference arises from SpaceX's ability to reuse the first stage of the rocket, significantly reducing expenses. By accurately predicting whether the first stage will land successfully, we can estimate the overall cost of a rocket launch. This information is of great interest to other companies looking to compete with SpaceX in the rocket launch industry.
- Research Questions: In pursuit of the project's objectives, we aim to address the following key questions:
- What are the key characteristics that differentiate a successful landing from a failed landing of the Falcon 9 first stage?
- How do various rocket variables and their relationships impact the success or failure of the landing?
- What specific conditions enable SpaceX to achieve the highest success rate for landing the Falcon 9 first stage?

Section 1

Methodology

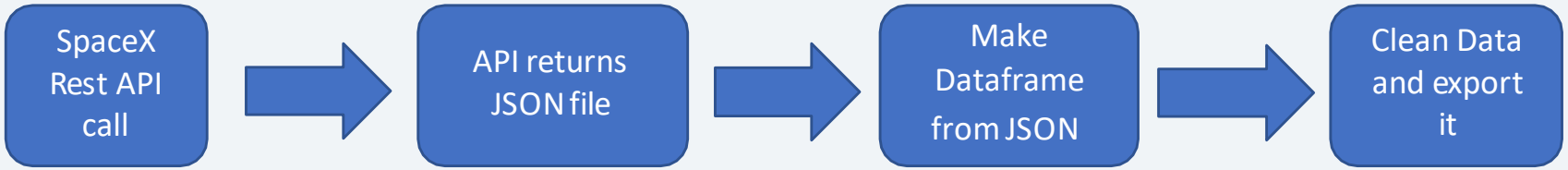
Methodology

Executive Summary

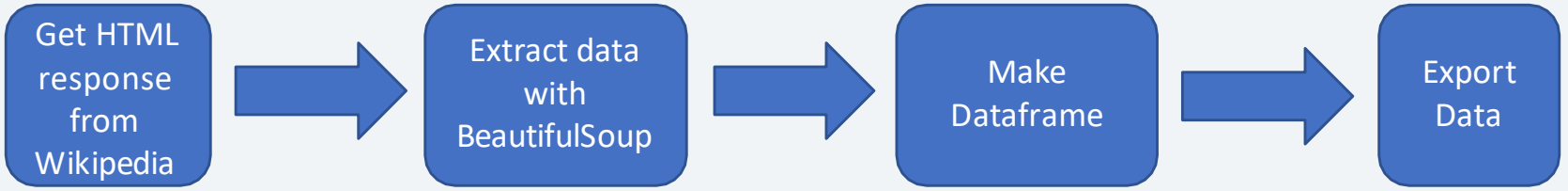
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using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

Data Collection

- The data for this project is sourced from two main channels: the SpaceX REST API and web scraping Wikipedia
- The API provides valuable information on rockets, launches, and payload details.
- The specific URL for accessing the SpaceX REST API is api.spacexdata.com/v4/.



- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
  - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection - SpaceX API

1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON File

```
data = response.json()
data = pd.json_normalize(data)
```

3. Transform data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

4. Create dictionary with data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Export to file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Link to code

Data Collection - Scraping

1. Getting Response from HTML

```
response = requests.get(static_url)
```

2. Create BeautifulSoup Object

```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables

```
html_tables = soup.findAll('table')
```

4. Get column names

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

See notebook for the rest of code

7. Create dataframe from dictionary

```
df=pd.DataFrame(launch_dict)
```

8. Export to file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Link to code

Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure.

1. Calculate launches number for each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Calculate the number and occurence of each orbit

```
df['Orbit'].value_counts()
```

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

3. Calculate number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

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

4. Create landing outcome label from Outcome column

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

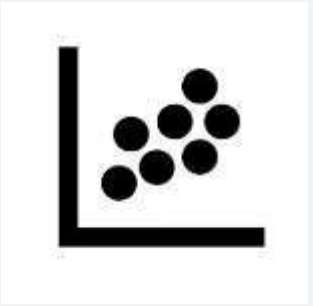
5. Export to file

```
df.to_csv("dataset_part_2.csv", index=False)
```

Link to code

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 behavior and make prediction for unseen data.

Link to code

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
  - Displaying the names of the unique lauunch sites in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landiing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Link to code

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name *(folium.Circle, folium.map.Marker).*
  - Red circles at each launch site coordinates with label showing launch site name *(folium.Circle, folium.map.Marker, folium.features.DivIcon).*
  - The grouping of points in a cluster to display multiple and different information for the same coordinates *(folium.plugins.MarkerCluster).*
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. *(folium.map.Marker, folium.Icon).*
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. *(folium.map.Marker, folium.PolyLine, folium.features.DivIcon)*
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Link to code

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites *(dash_core_components.Dropdown)*.
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component *(plotly.express.pie)*.
  - Rangeslider allows a user to select a payload mass in a fixed range *(dash_core_components.RangeSlider)*.
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass *(plotly.express.scatter)*.

Link to code

Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalization of data
  - Do Split data into training and test sets.
- Model preparation
  - Selection of ML algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Do Get the best hyperparameters for each type of ml model
  - Compute accuracy for each ML model with test dataset itself
  - Plot Confusion Matrix endly
- Model comparison
  - Comparison of ML models according to their accuracy of results.
  - The ML model with the best accuracy will be chosen (see Notebook for result)

Link to code

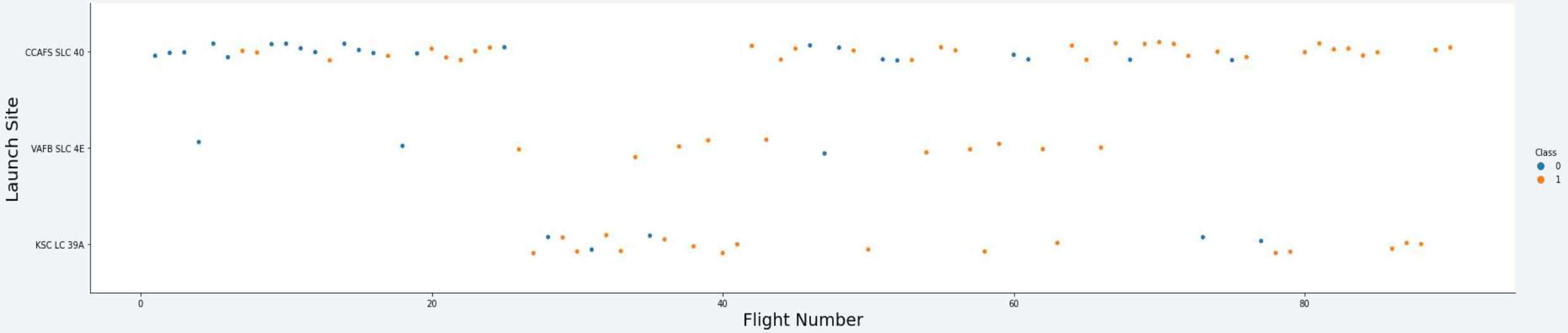
Results

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

Section 2

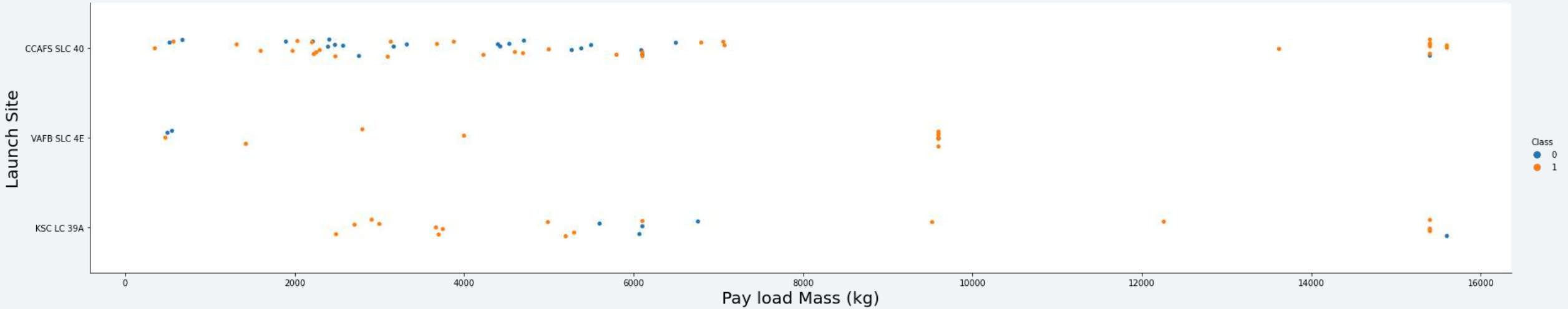
Insights drawn from EDA

Flight Number vs. Launch Site



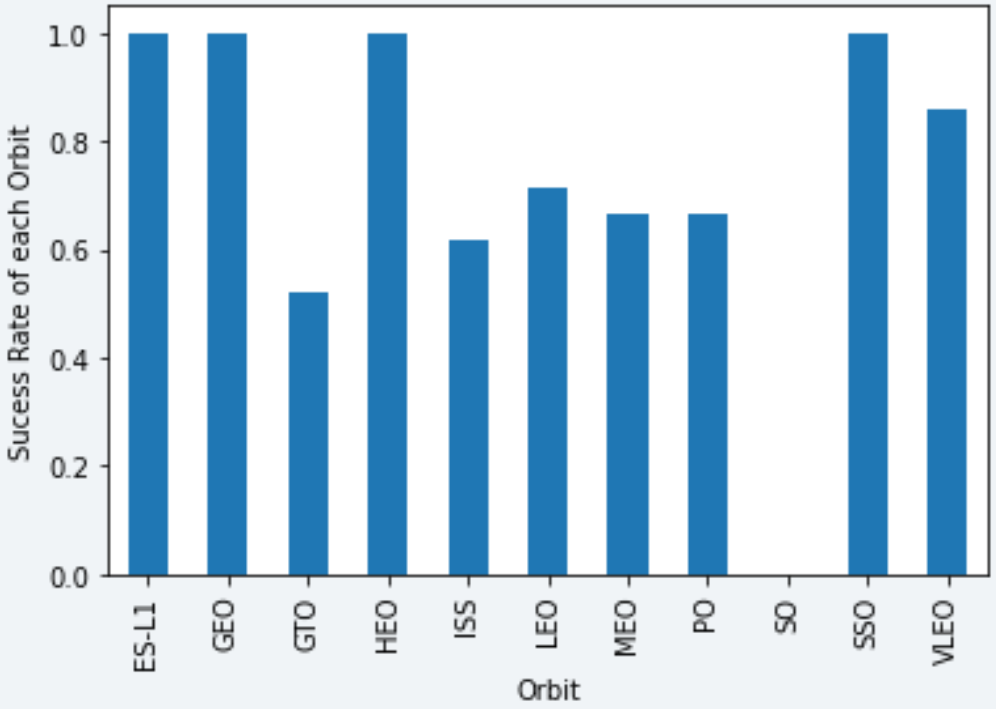
We are viewing that, for each site of flights , the success rate is perfectly increasing.

Payload vs. Launch Site



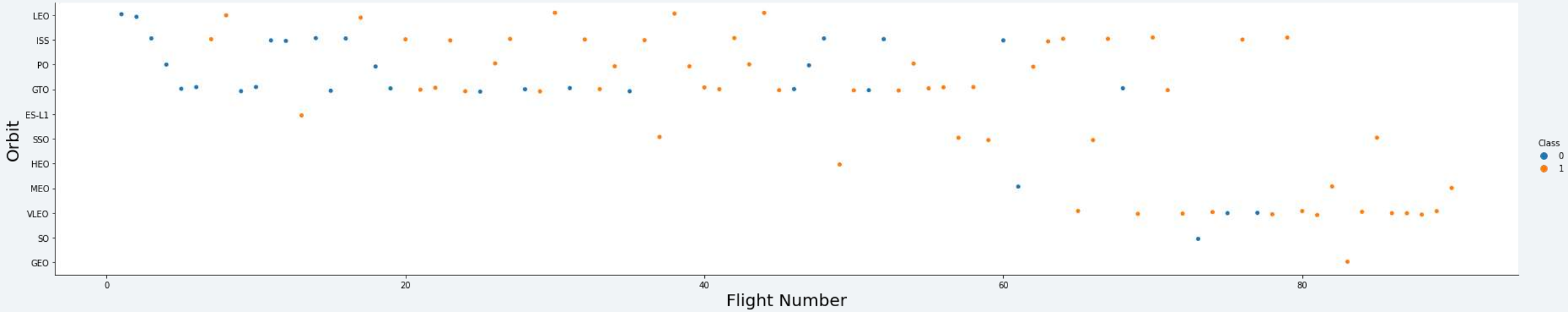
The viability of a successful landing can vary depending on the launch site, where a heavier payload might play a crucial role. However, it is essential to note that an excessively heavy payload could lead to a failed landing attempt.

Success Rate vs. Orbit Type



By examining this plot, we can observe the success rates for various orbit types. Notably, ES-L1, GEO, HEO, and SSO exhibit the highest success rates among the listed orbits.

Flight Number vs. Orbit Type

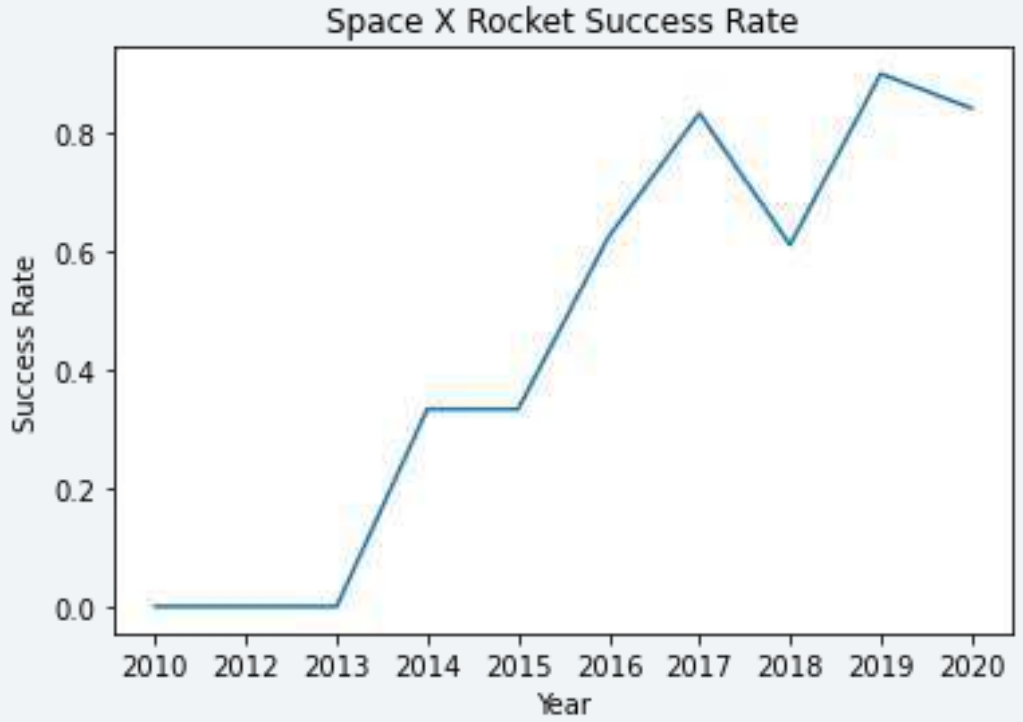


Upon observation, we find that the success rate tends to rise with the increasing number of flights for the LEO orbit. However, for orbits like GTO, there seems to be no discernible correlation between the success rate and the number of flights. Nevertheless, it is plausible to assume that the high success rates observed in orbits such as SSO or HEO may be attributed to the knowledge gained from previous launches in different orbits, contributing to their successful outcomes.

Payload vs. Orbit Type

The success rate of launches in specific orbits is significantly impacted by the weight of the payloads. In the case of the LEO orbit, heavier payloads tend to enhance the success rate. Conversely, for GTO orbits, reducing the payload weight has been observed to improve the chances of a successful launch.

Launch Success Yearly Trend



From the observation since 2013, we can see an increase in the Space X Rocket success rate gradually.

All Launch Site Names

SQL Query

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Results

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

Explanation

By utilizing the DISTINCT keyword in the query, duplicate LAUNCH_SITE entries can be effectively eliminated.

Launch Site Names Begin with 'CCA'

SQL Query

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

Explanation

The combination of the WHERE clause followed by the LIKE clause allows for filtering launch sites that contain the substring "CCA." Applying LIMIT 5 displays the first 5 records resulting from this filtering process.

Results

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Total Payload Mass

SQL Query

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Results

SUM("PAYLOAD_MASS__KG_")
45596

Explanation

This query returns the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

SQL Query

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

Results

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

Executing this query provides the average payload mass for all records where the booster version includes the substring "F9 v1.1."

First Successful Ground Landing Date

SQL Query

```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

Results

MIN("DATE")
01-05-2017

Explanation

This query aims to identify the oldest successful landing. The WHERE clause filters the dataset to retain only records where the landing was successful. By using the MIN function, we select the record with the earliest date, representing the oldest successful landing.

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

Explanation

The query retrieves the booster version when the landing was successful, and the payload mass falls between 4000 and 6000 kg. The WHERE and AND clauses are used to filter the dataset accordingly.

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

Explanation

In the initial SELECT statement, we present the subqueries and their corresponding results. The first subquery counts the number of successful missions, while the second subquery counts the number of unsuccessful missions. The WHERE clause, followed by the LIKE clause, filters the mission outcomes. The COUNT function is then utilized to count the records that meet the specified filtering criteria.

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

Explanation

To filter the data and retrieve only the heaviest payload mass, we employed a subquery using the MAX function. The main query then utilizes the results from the subquery to identify unique booster versions (SELECT DISTINCT) associated with the heaviest payload mass.

2015 Launch Records

SQL Query

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

Results

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

Explanation

The query provides the month, booster version, and launch site where the landing was unsuccessful and took place in the year 2015. The Substr function is used to extract the month or year from the landing date. Substr(DATE, 4, 2) retrieves the month, while Substr(DATE, 7, 4) extracts the year from the date.

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

SQL Query

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

Results

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

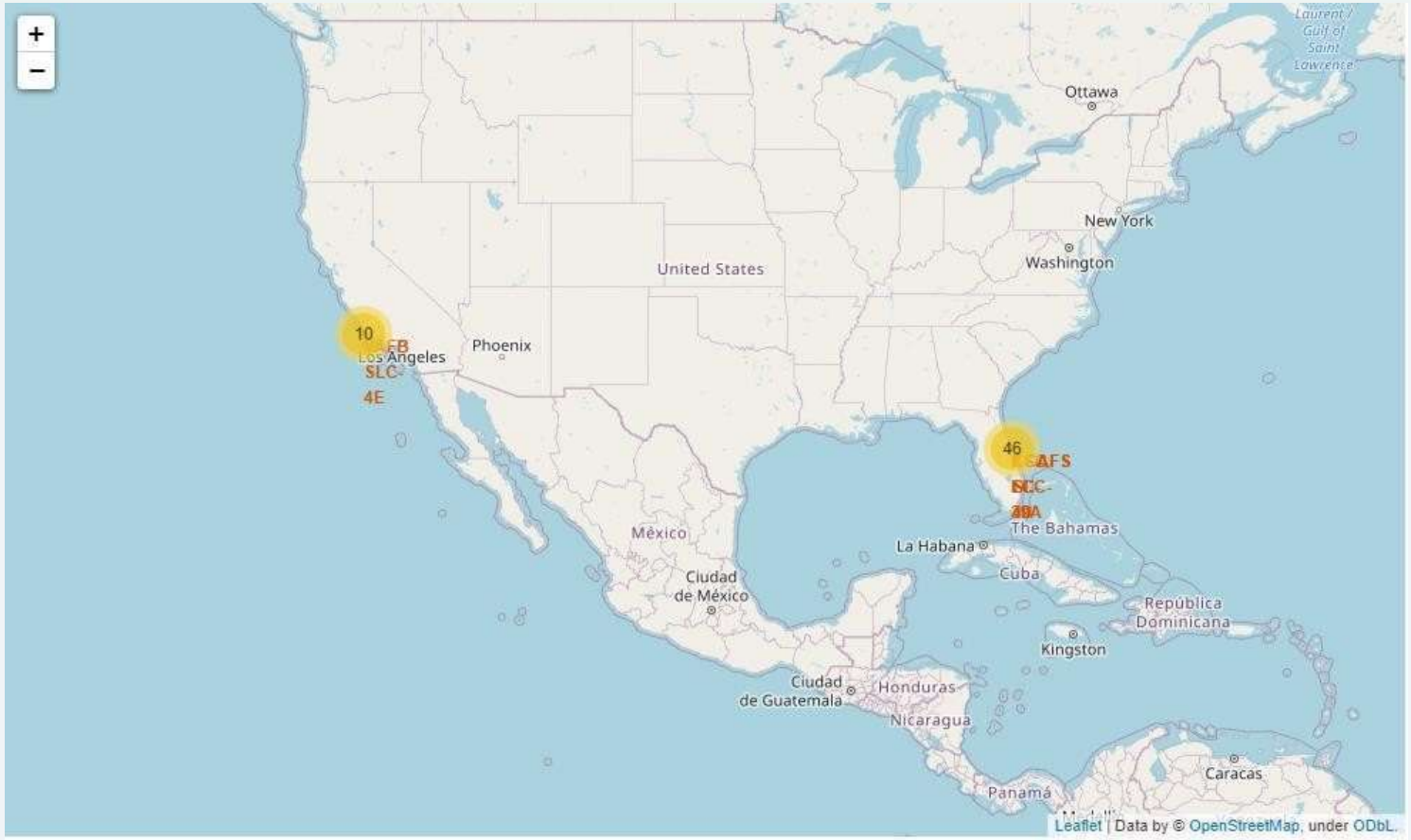
Explanation

The query presents the landing outcomes and their respective counts when the mission was successful, and the date falls between 04/06/2010 and 20/03/2017. The GROUP BY clause groups the results based on the landing outcome, and the ORDER BY COUNT DESC displays the results in decreasing order of counts.

Section 4

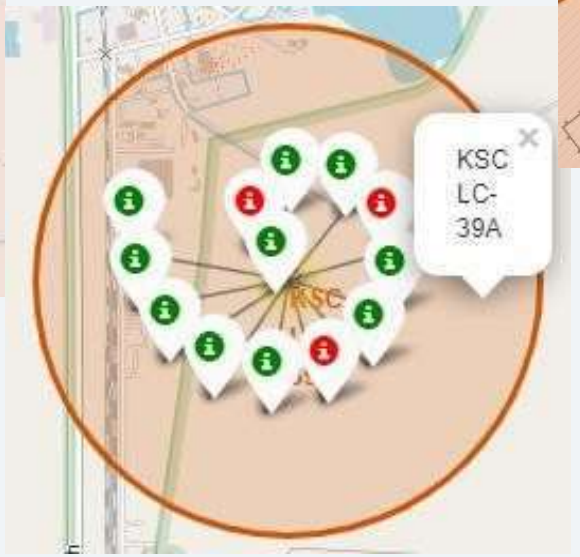
Launch Sites Proximities Analysis

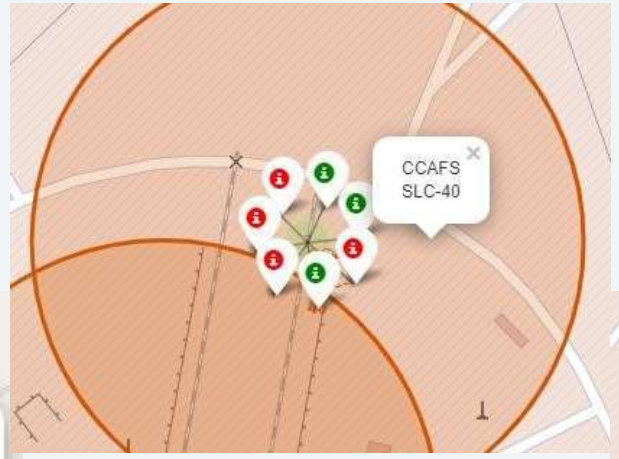
Folium map - Ground stations

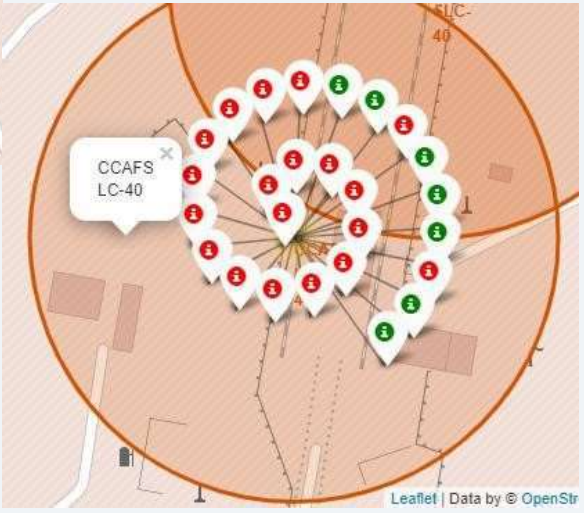


We observe that Space X launch sites are located on the coast of the United States

Folium map - Color Labeled Markers

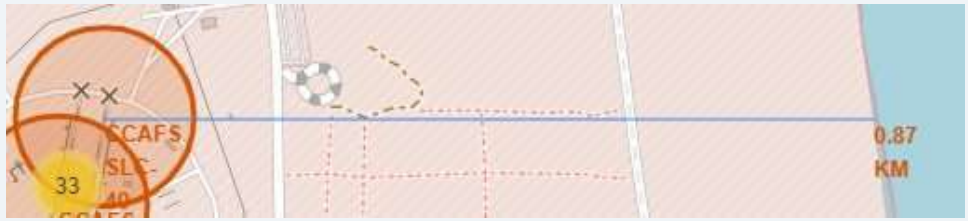


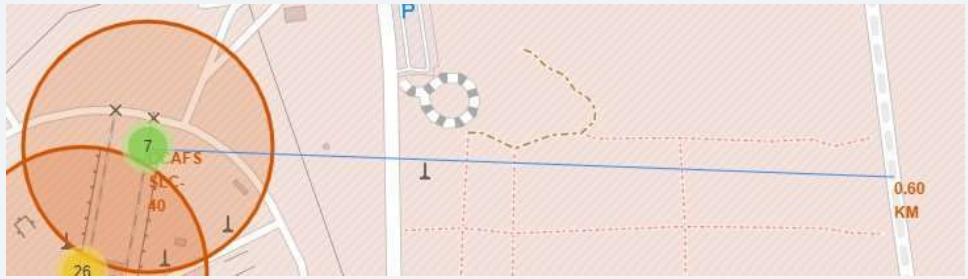


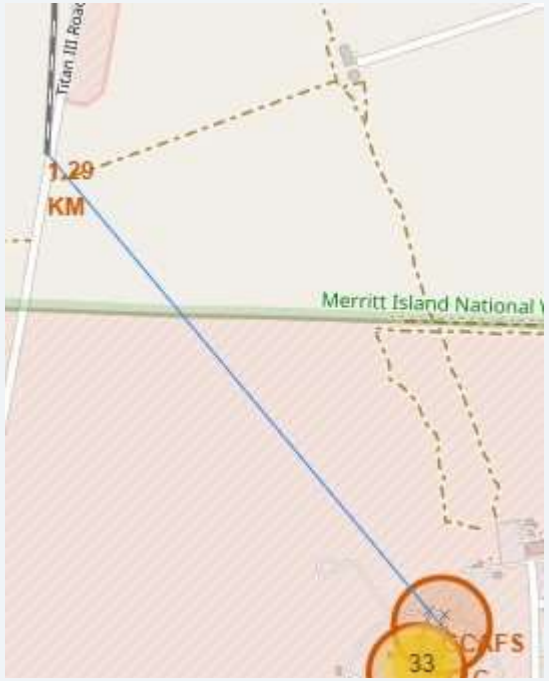


The green marker indicates successful launches, while the red marker represents unsuccessful launches. Observing the data, it is evident that KSC LC-39A exhibits a higher launch success rate.

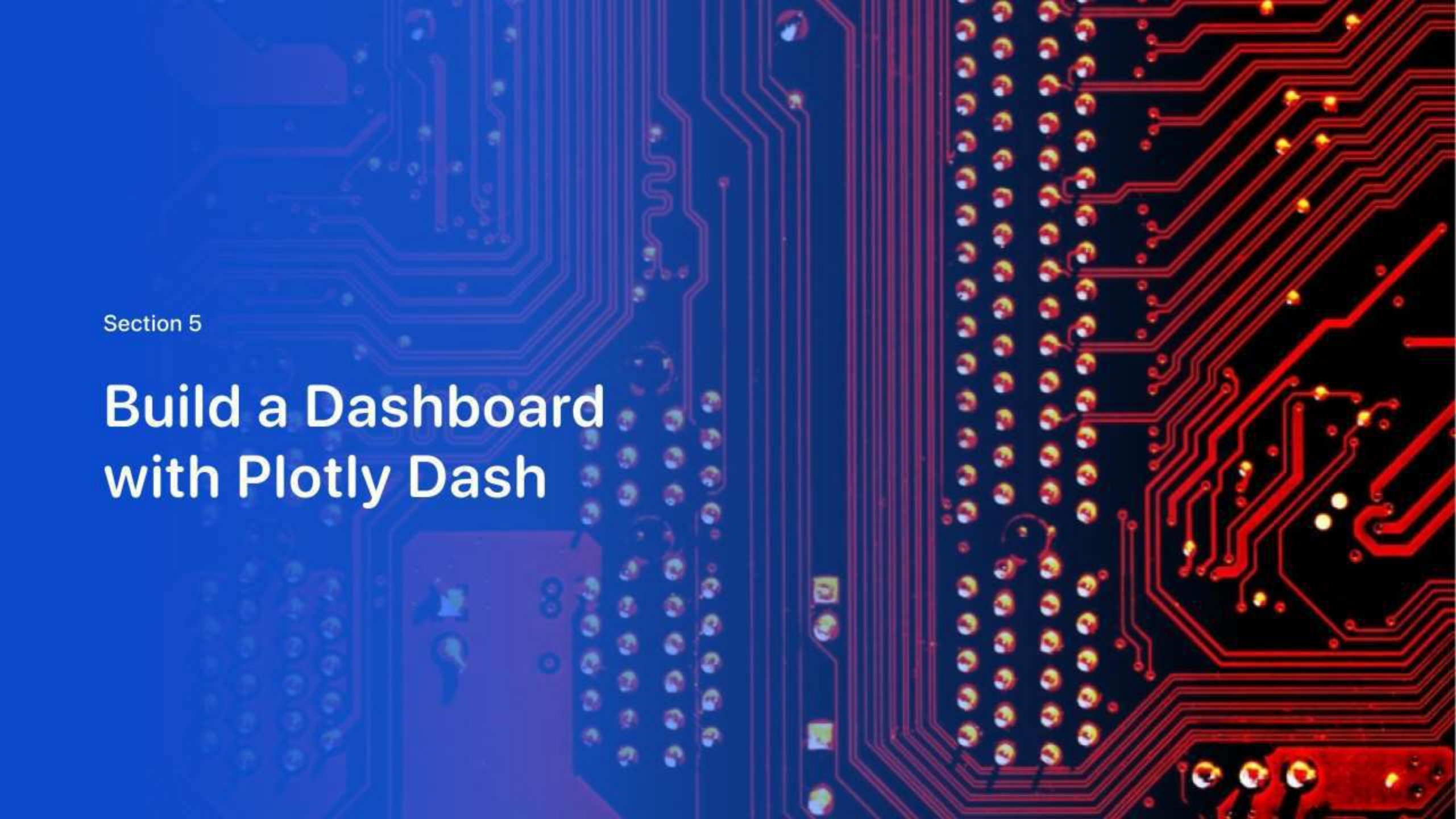
Folium Map - Distances between CCAFS SLC-40 and its proximities







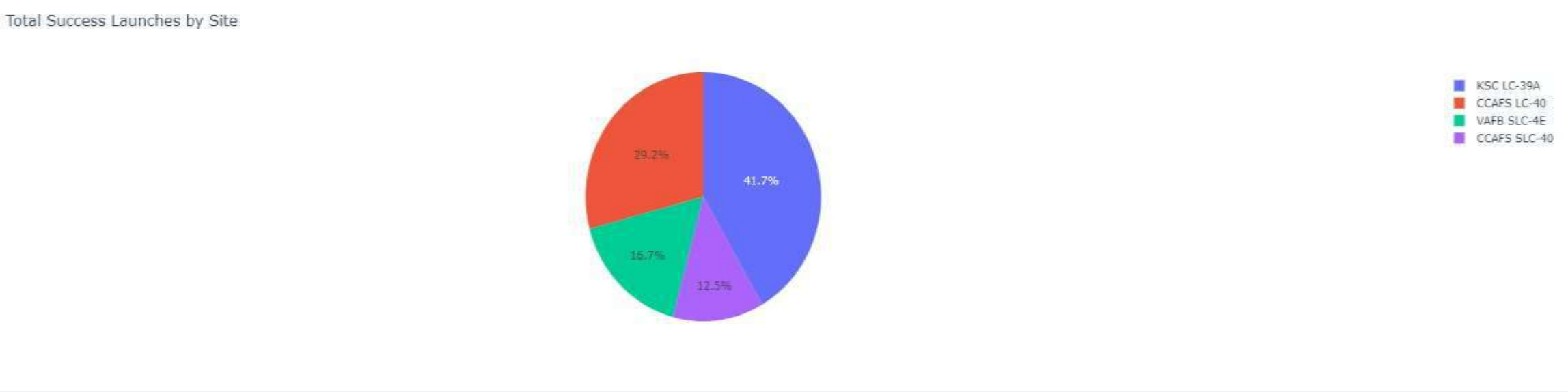
CCAFS SLC-40 is located in close proximity to railways, highways, and the coastline. However, it does not maintain a significant distance from cities.



Section 5

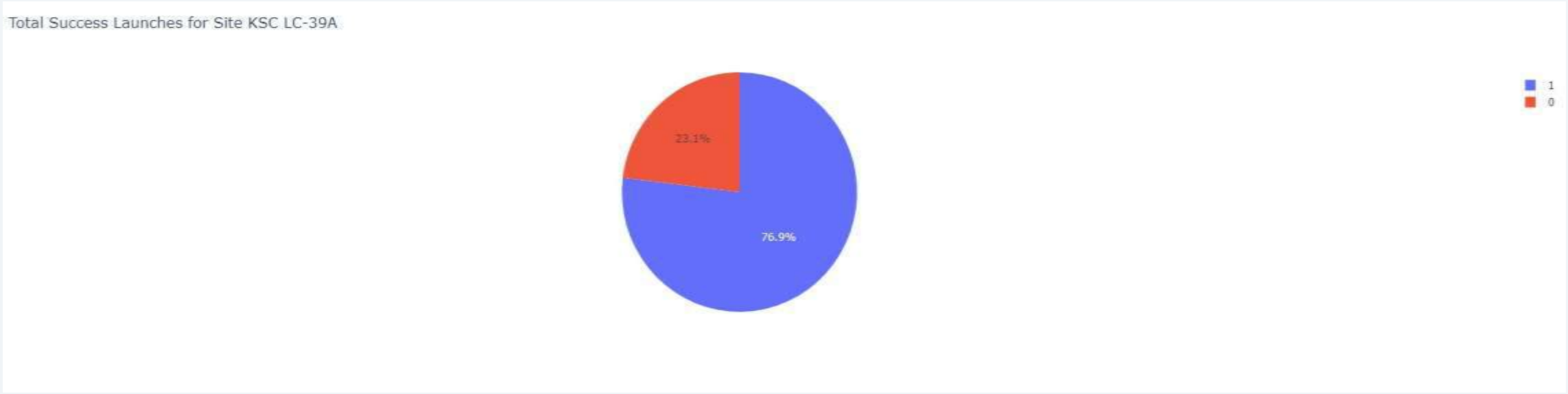
Build a Dashboard with Plotly Dash

Dashboard - Total success by Site



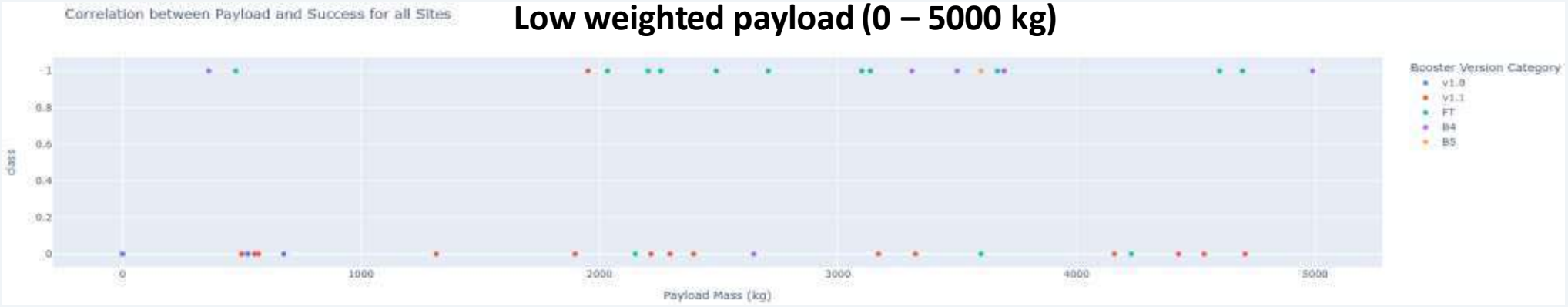
We observe that KSC LC-39A has the best success rate of launches in the figure above.

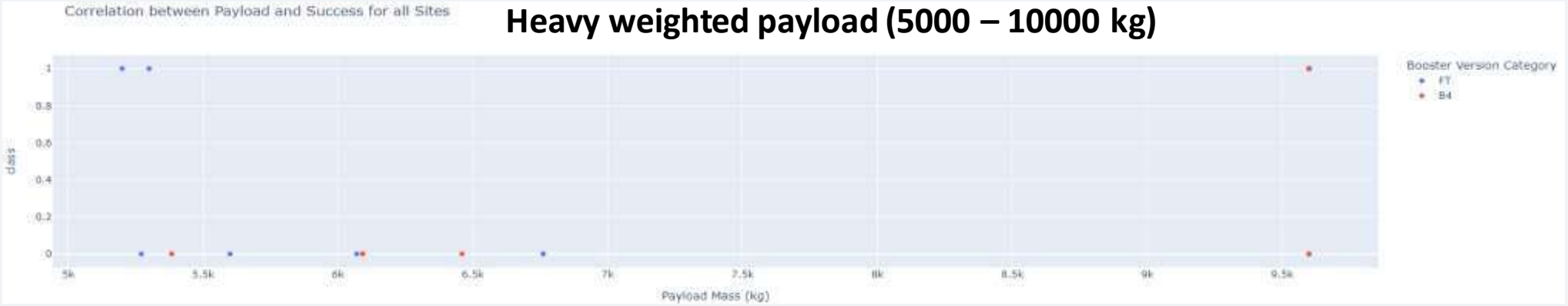
Dashboard - Total success launches for Site KSC LC-39A



Upon examination, KSC LC-39A has attained a success rate of 76.9% and a failure rate of 23.1%.

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





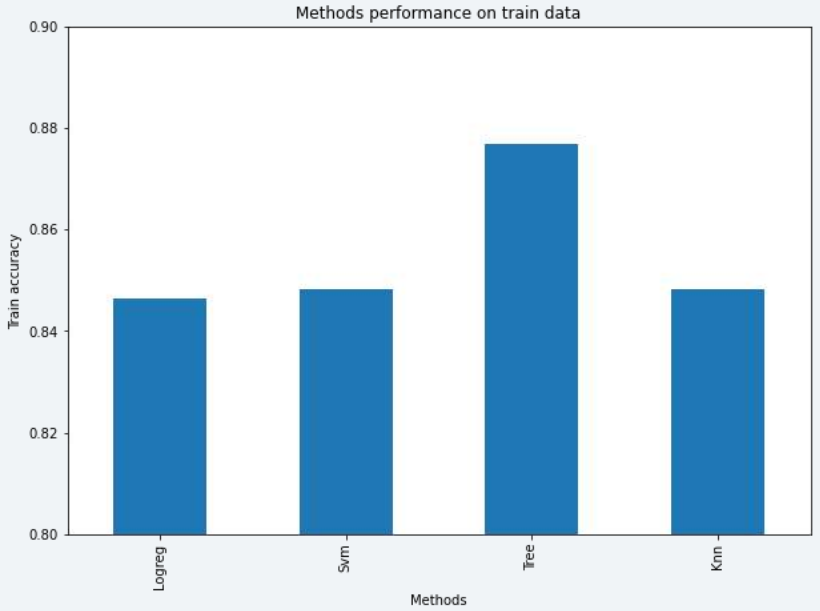
Payloads with lower weight demonstrate a higher success rate compared to payloads with heavier weight..

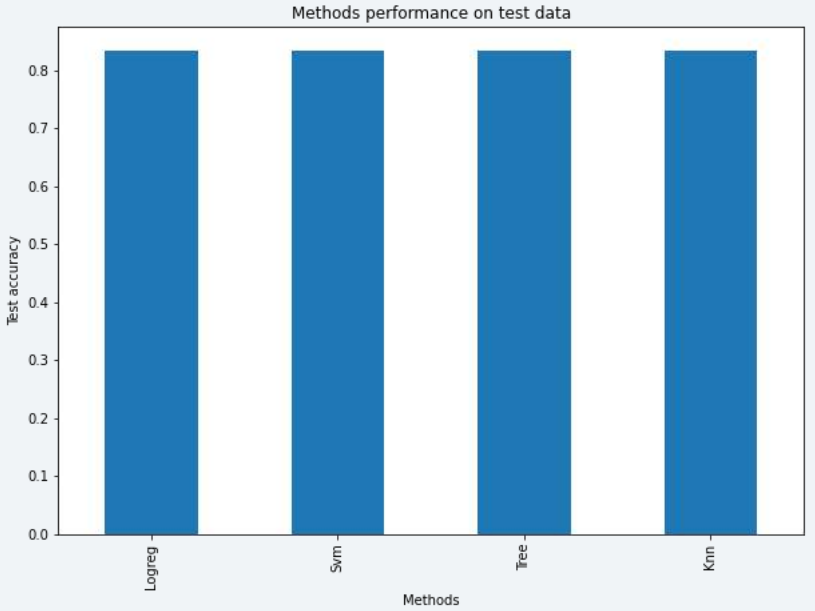
Section 6

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333





During the accuracy test, all methods showed similar performance. To make a final decision, obtaining more test data could be beneficial. However, if an immediate choice is necessary, the decision tree method would be preferred. **Decision tree best parameters**

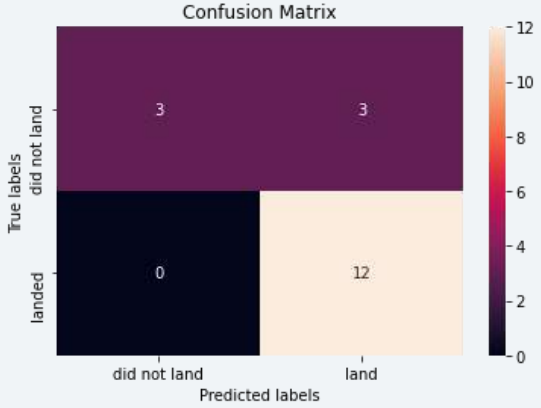
```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix

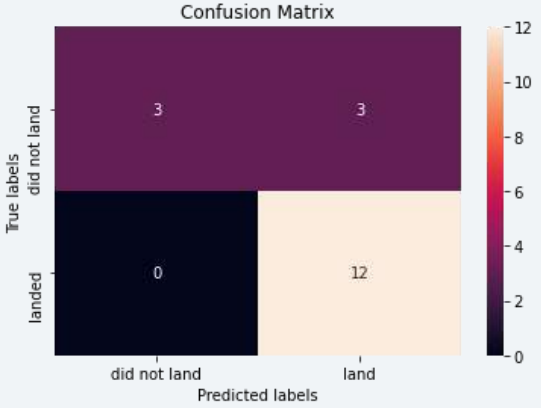
Logistic regression

Decision Tree

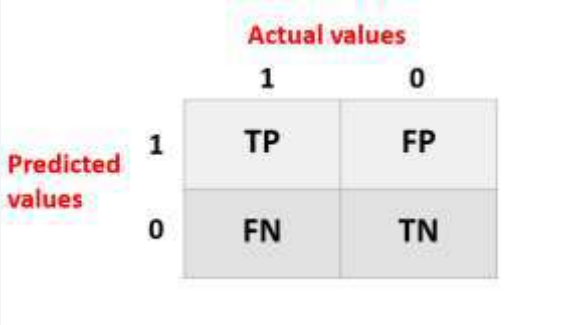
kNN



SVM



Since the test accuracies of all models are identical, the confusion matrices are also the same. The primary issue encountered with these models is the occurrence of false positives.



Conclusions

- The success of a mission can be attributed to various factors, including the launch site, the orbit, and particularly the number of previous launches. It is reasonable to assume that the accumulation of knowledge from prior launches contributed to the transition from launch failures to successful missions.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- The success of a mission is influenced by the orbits, where the payload mass becomes a critical factor. Different orbits demand specific light or heavy payload masses. However, as a general trend, lower weighted payloads tend to achieve better mission success compared to heavier payloads.
- Based on the available data, the reasons for the performance variations among launch sites, with KSC LC-39A being the most successful, cannot be explicitly explained. To address this issue, obtaining additional atmospheric or relevant data might be necessary to gain deeper insights into the disparities observed among launch sites.
- Among the models used for this dataset, the Decision Tree Algorithm is selected as the preferred choice, even though the test accuracy is the same for all models. The decision is based on the fact that the Decision Tree Algorithm exhibits a higher training accuracy compared to the others.

Thank you!