IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Danijel Petrović
14.03.2024

Outline

- ExecutiveSummary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

ExecutiveSummary

- Summaryof methodologies
  - Data Collectionvia API,Web Scraping
  - ExploratoryData Analysis(EDA) with Data Visualization
  - EDA with SQL
  - InteractiveMap with Folium
  - Dashboards with Plotly Dash
  - PredictiveAnalysis
- Summaryof all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictiveresults

Introduction

- Project background and context
- In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
- What are the main characteristics of a successful or failed landing ?
- What are the effects of each relationship of the rocket variables on the successor failure of a landing ?

- What are the conditions which will allow SpaceXto achieve the best landing successrate ?

Section 1

Methodology

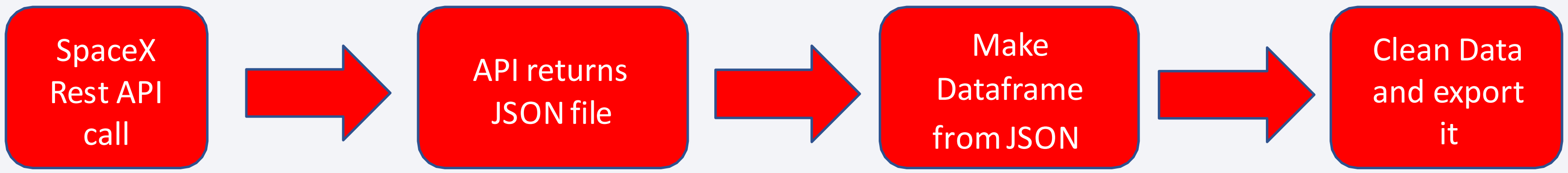
Methodology

Executive Summary

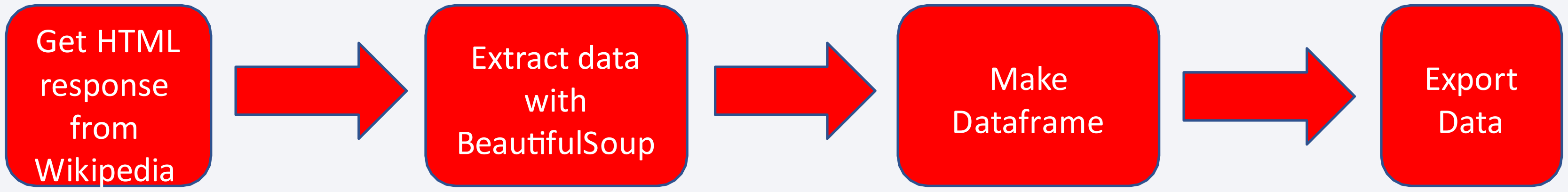
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping
- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly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

Data Collection

- Datasets are collected from Rest SpaceXAPI and webscrapping Wikipedia
  - The information obtained by the API are rocket, launches,payload information.



  - Theinformation obtained by the webscrapping of Wikipedia are launches,landing, payload information.



Data Collection - SpaceXAPI

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

3. Transform data

```
# Call getLaunchSite
getLaunchSite(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

```
# Create a data from launch_dict
df_launch = pd.DataFrame(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = df_launch[df_launch['BoosterVersion']!='Falcon 1']
# Hint 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

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

2. Create BeautifulSoup Object

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

3. Find all tables

```python
html_tables = soup.find_all("table")
```

4. Get column names

```python
for row in first_launch_table.find_all('th'):
    cols = row.find_all('td')
    name = extract_column_from_header(row)

    if name is not None and len(name) > 0:
        column_names.append(name)
```

[Link to code]

5. Create dictionary

```python
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

```python
for table_number,table in enumerate(soup.find_all('tab
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as num
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

See notebook for the rest of code

7. Create dataframe from dictionary

```python
df = pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

8. Export to file

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

Data Wrangling

- Within the dataset, numerous instances exist where the booster failed to land successfully. When denoted as True Ocean, True RTLS, and True ASDS, it signifies a successful mission. Conversely, False Ocean, False RTLS, and False ASDS denote mission failure. Our objective is to convert string variables into categorical ones, where 1 indicates a successful mission and 0 indicates failure.

1. Calculate launches number for each site

```
# Apply value_counts() on column LaunchSite
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

```
# 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
```

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

```
# 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
```

[Link to code]

4. Create landing outcome label from Outcome column

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

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

5. Export to file

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

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
- Bar Graph
  - Successrate vs. Orbit
- Line Graph
  - Successrate vs. Year

[Link to code]()

EDA with SQL

We conducted SQL queries to retrieve and analyze data from the dataset:

1. Displaying the unique launch site names in the space mission.
2. Displaying 5 records where launch sites begin with the string 'CCA'.
3. Displaying the total payload mass carried by boosters launched by NASA (CRS).
4. Displaying the average payload mass carried by booster version F9 v1.1.
5. Listing the date when the first successful landing outcome on a ground pad was achieved.
6. Listing the names of the boosters which have successfully landed on a drone ship and have a payload mass greater than 4000 but less than 6000.
7. Listing the total number of successful and failed mission outcomes.
8. Listing the names of the booster versions which have carried the maximum payload mass.
9. Listing the records that display the month names, failure landing outcomes on a drone ship, booster versions, and launch sites for the months in the year 2015.
10. Ranking the count of successful landing outcomes between the dates 04-06-2010 and 20-03-2017 in descending order.

Link to code

Build an Interactive Map with Folium

The Folium map object is configured to center on NASA Johnson Space Center in Houston, Texas. It includes several features:

1. A red circle marker at the coordinates of NASA Johnson Space Center, labeled with its name.
2. Red circle markers at each launch site's coordinates, labeled with the launch site name. Clustered markers are utilized to display multiple points with different information at the same coordinates.
3. Markers indicating successful and unsuccessful landings, colored green and red respectively.
4. Additional markers to denote distances between launch sites and key locations such as railways, highways, coastways, and cities, with lines drawn to represent these distances.

These features are designed to aid in understanding the problem and the data, facilitating the visualization of launch sites, their surroundings, and the outcomes of missions.

Link to code

Build a Dashboard with Plotly Dash

The dashboard comprises several components:
1. Dropdown: Users can select individual launch sites or view data for all launch sites using the dropdown feature (dash_core_components.Dropdown).
2. Pie chart: This chart displays the total success and failure outcomes for the selected launch site from the dropdown menu (plotly.express.pie).
3. Range slider: Users can specify a payload mass within a predefined range using the rangeslider component (dash_core_components.RangeSlider).
4. Scatter plot: This chart visualizes the relationship between two variables, specifically Success versus Payload Mass (plotly.express.scatter).

Predictive Analysis (Classification)

Data Preparation:
1.Loading the dataset.
2.Normalizing the data.
3.Splitting the data into training and test sets.

Model Preparation:
1.Selection of machine learning algorithms.
2.Setting parameters for each algorithm using GridSearchCV.
3.Training GridSearchModel models with the training dataset.

Model Evaluation:
1.Obtaining the best hyperparameters for each type of model.
2.Computing accuracy for each model using the test dataset.
3.Plotting the Confusion Matrix.

Model Comparison:
1.Comparing models based on their accuracy.
2.Choosing the model with the best accuracy

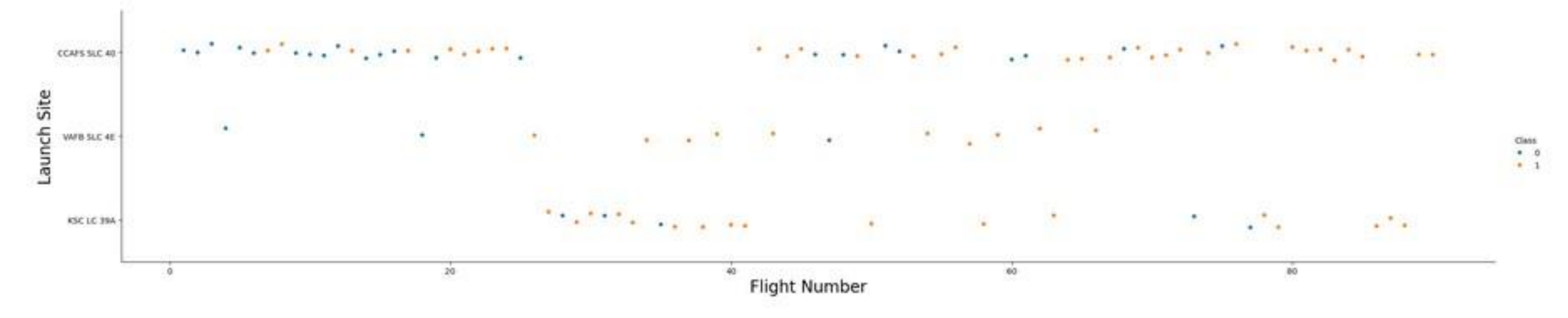
Results

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

Section 2

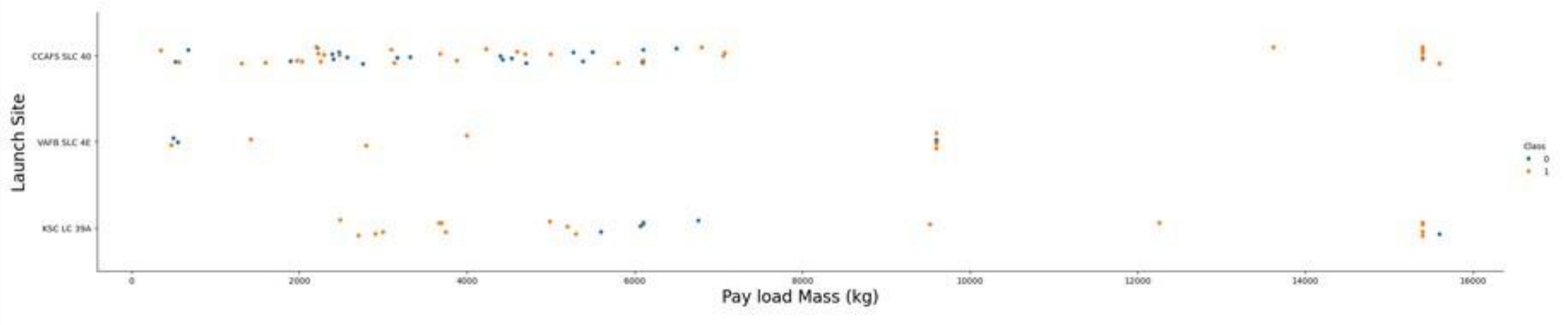
Insights drawn from EDA

Flight Number vs. Launch Site



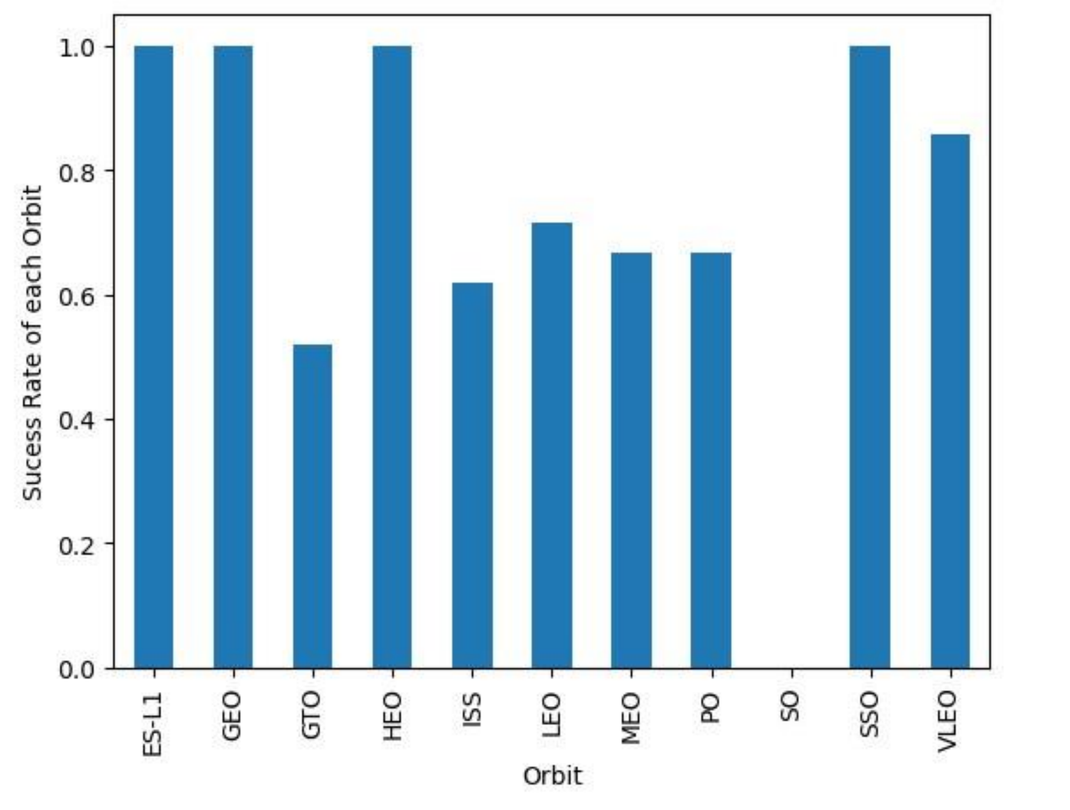
Successrate is increasing.

Payload vs. Launch Site



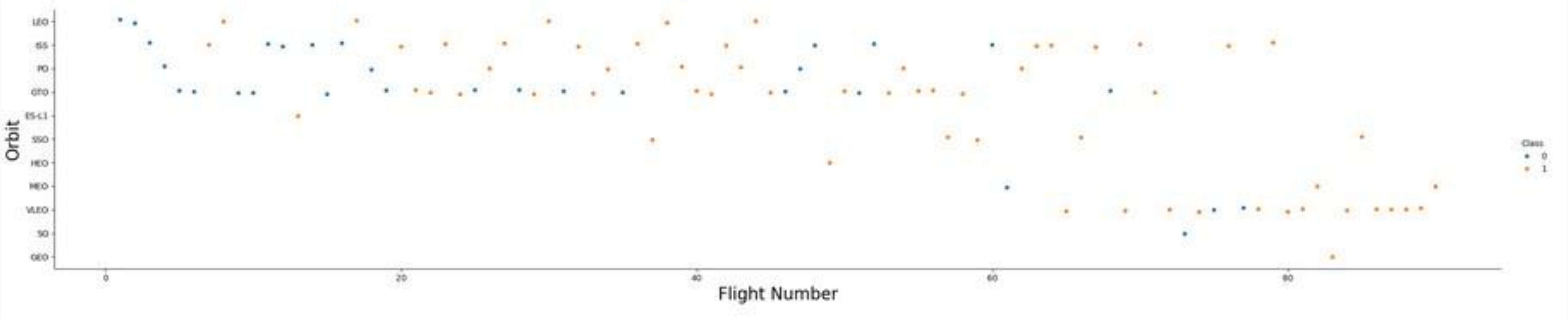
The success of a landing may depend on the specific launch site, where a heavier payload could potentially facilitate a successful landing. However, it's important to note that an excessively heavy payload might lead to a failed landing.

Success Rate vs. Orbit Type



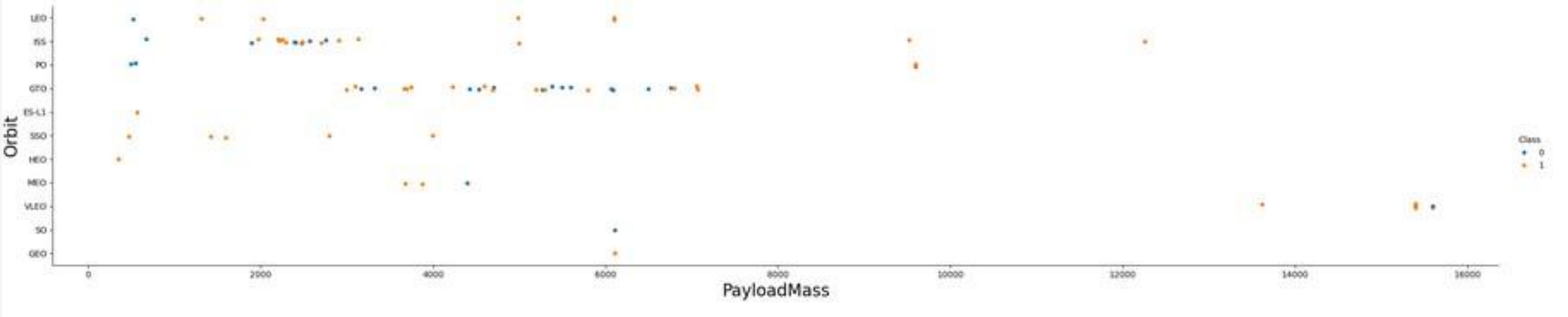
By examining this plot, we can observe the success rates across various orbit types. It's evident that orbits such as ES-L1, GEO, HEO, and SSO demonstrate the highest success rates.

Flight Number vs. Orbit Type



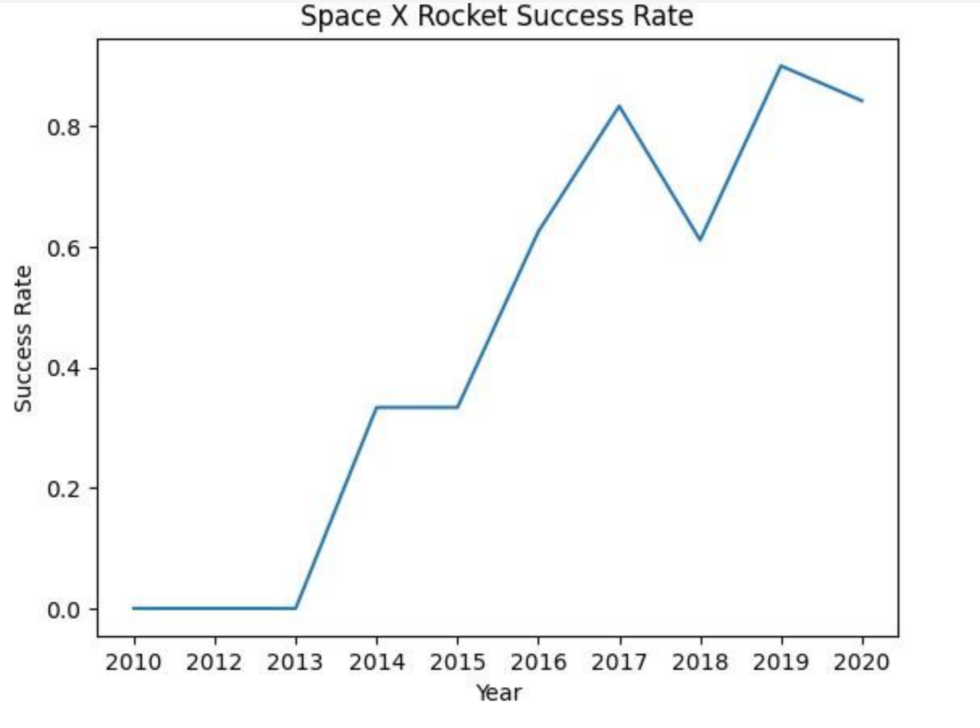
We observe a correlation between the success rate and the number of flights for the LEO orbit, indicating an increase in success rate with more flights. However, for orbits like GTO, there seems to be no discernible relationship between the success rate and the number of flights. It's plausible to suggest that the high success rates observed in orbits such as SSO or HEO are attributed to the knowledge gained from previous launches across different orbits.

Payloadvs. Orbit Type



The payload weight significantly impacts the success rate of launches in specific orbits. For instance, in the LEO orbit, heavier payloads tend to enhance the success rate. Conversely, decreasing the payload weight for a GTO orbit has been found to improve the likelihood of a successful launch.

Launch Success Yearly Trend



the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

The names of the unique launch sites in the space mission

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

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with ‘CCA’

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

The total payload mass carried by boosters launched by NASA (CRS)

Total payload mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

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

Average payload mass by Booster Version F9 v1.1
2928

First SuccessfulGround Landing Date

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

Date of first successful landing outcome in ground pad
2015-12-22

SuccessfulDrone Ship Landing with Payload between 4000 and 6000

The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

Total Number of Successfuland Failure Mission Outcomes

The total number of successful and failure mission outcomes

number_of_success_outcomes	number_of_failure_outcomes
100	1

Boosters Carried Maximum Payload

The names of the booster versions which have carried the maximum payload mass

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 LaunchRecords

The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

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

The count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

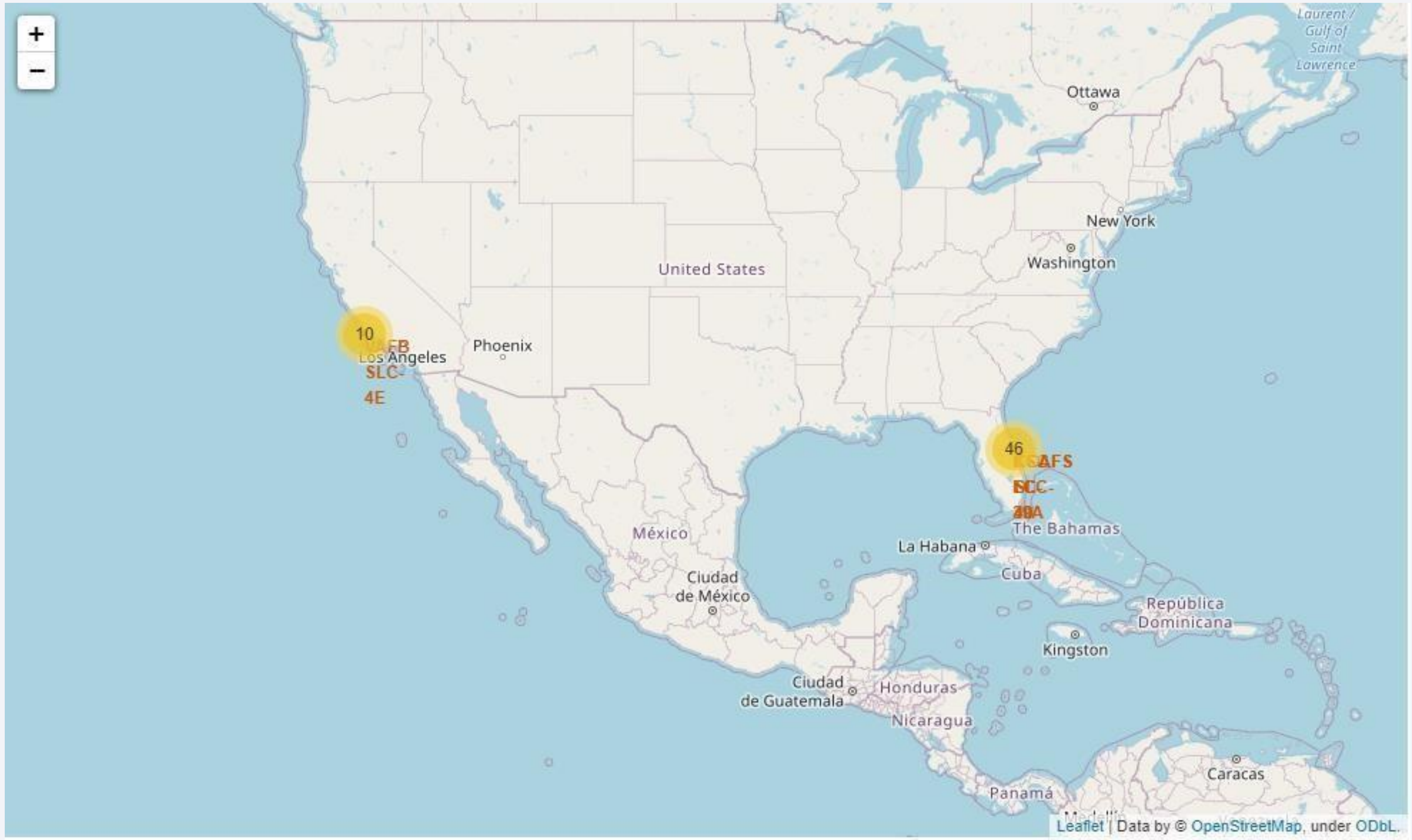
landing__outcome	landing_count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

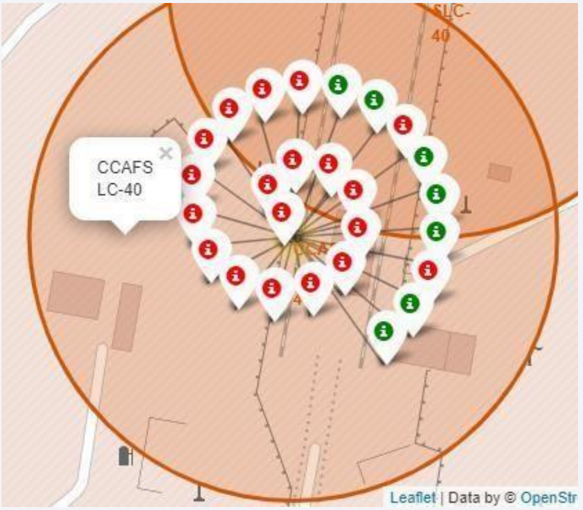
Folium map - Ground stations

SpaceX launch sites are located on the coast of the United States

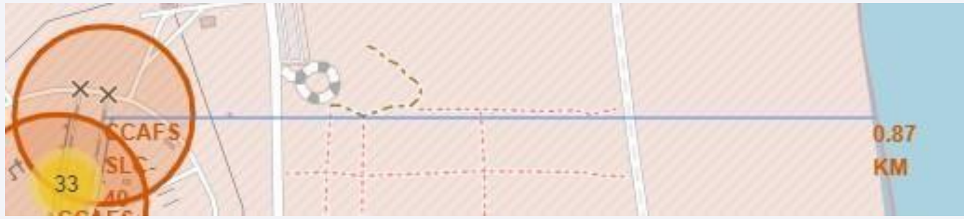


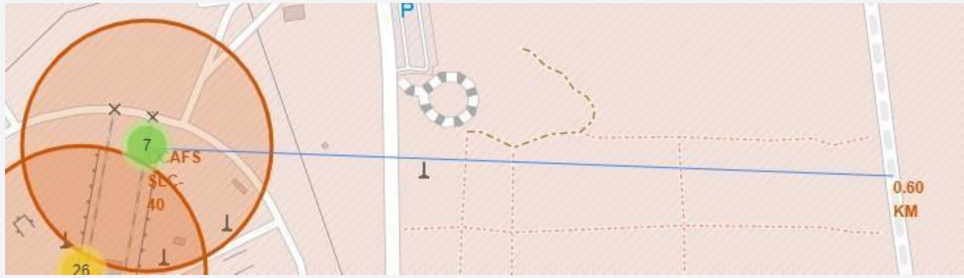
Folium map - Color Labeled Markers

Successful launches are indicated by green markers, while unsuccessful launches are represented by red markers. It is notable that KSC LC-39A exhibits a higher launch success rate.



Folium Map - Distancesbetween CCAFSSLC-40 and its proximities



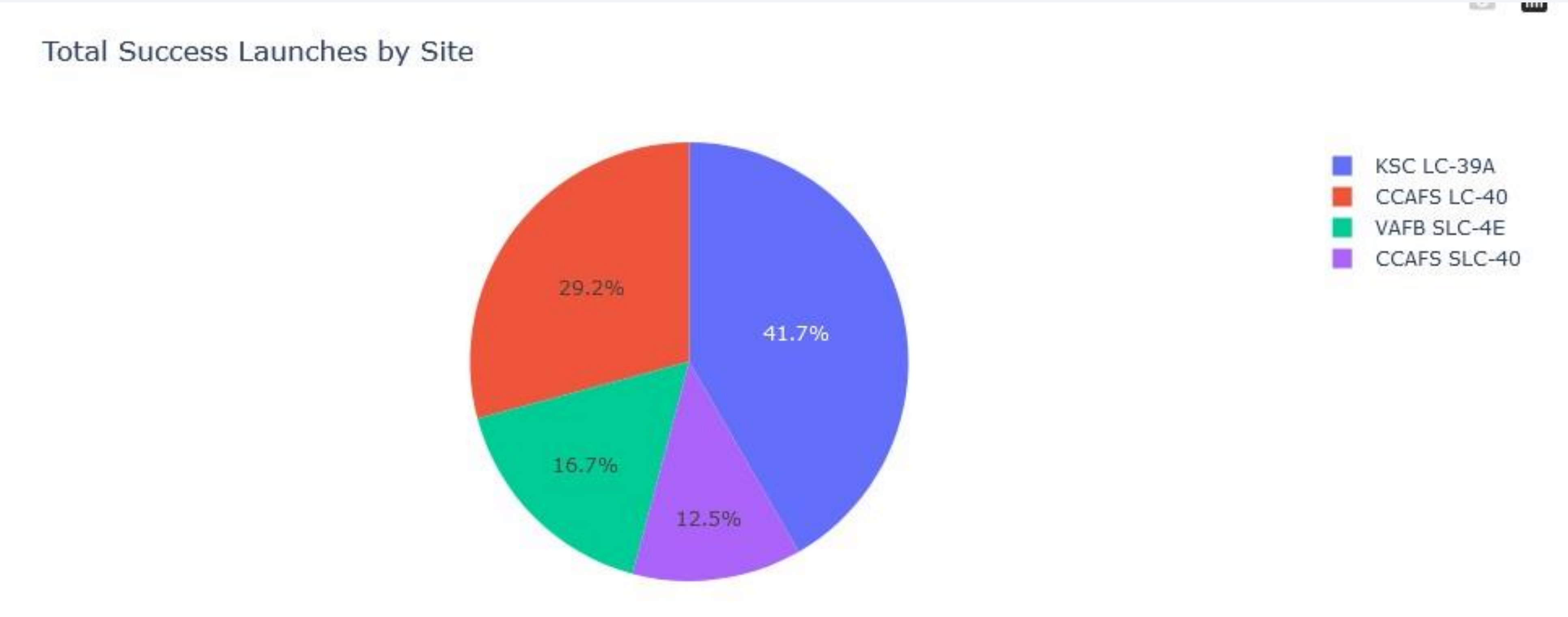


Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFSSLC-40 keeps certain distance away from cities ? No

Section 5

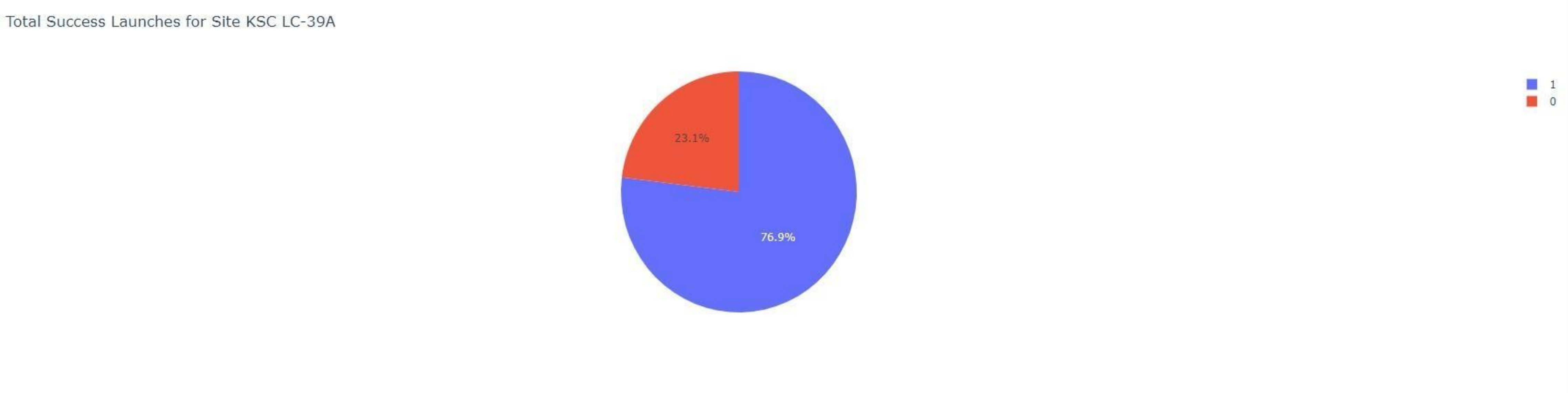
Build a Dashboard with Plotly Dash

Dashboard - Total successfor all sites



We see that KSCLC-39A is mostsuccessful

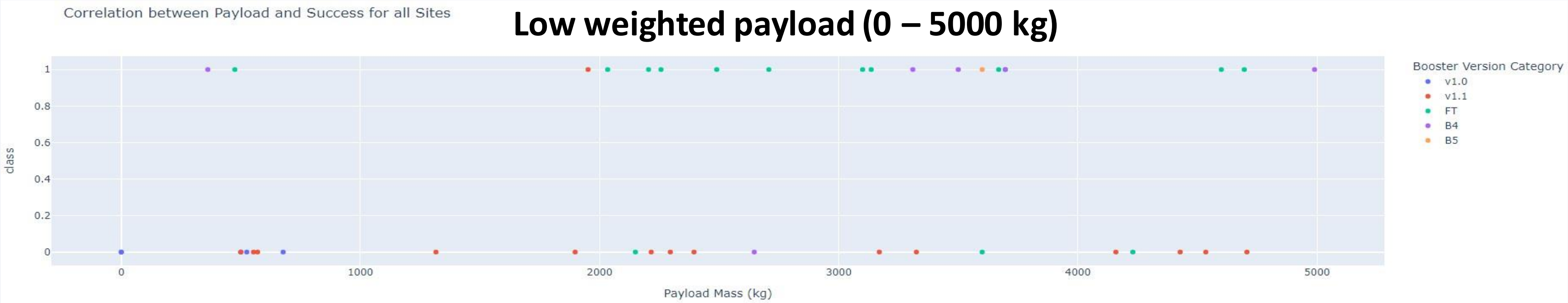
Dashboard - Total success launches for Site KSCLC-39A



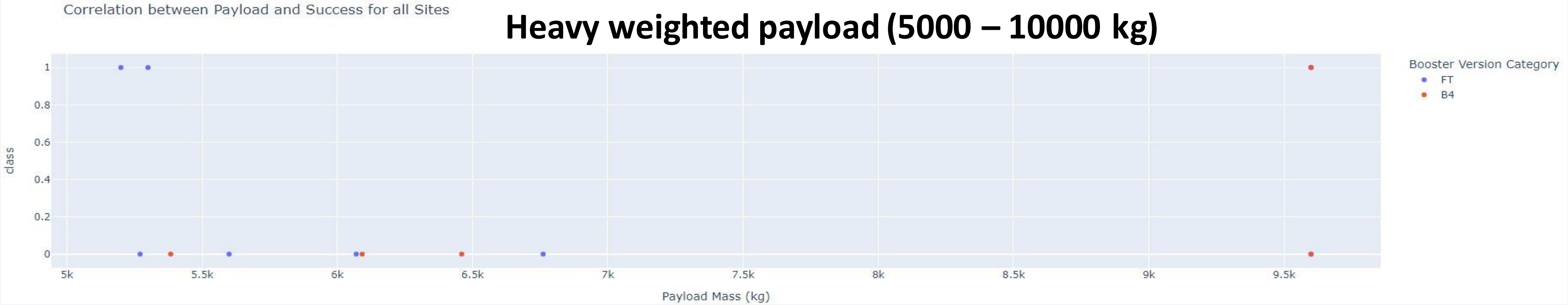
We see that KSCLC-39A has achieved a 76.9 % success rate while getting a 23.1 % failure rate.

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

Low weighted payload (0 – 5000 kg)



Heavy weighted payload (5000 – 10000 kg)



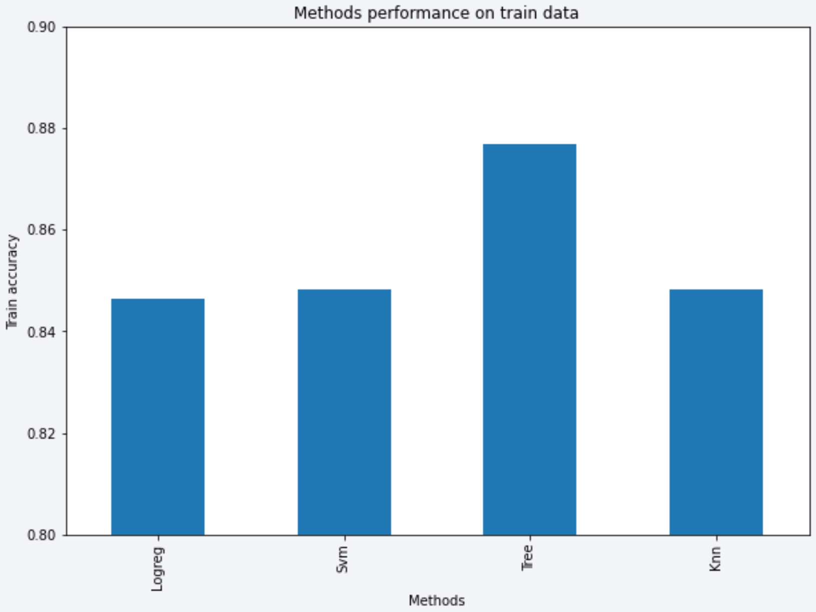
Low weighted payloads have a better success rate than the heavy weighted payloads.

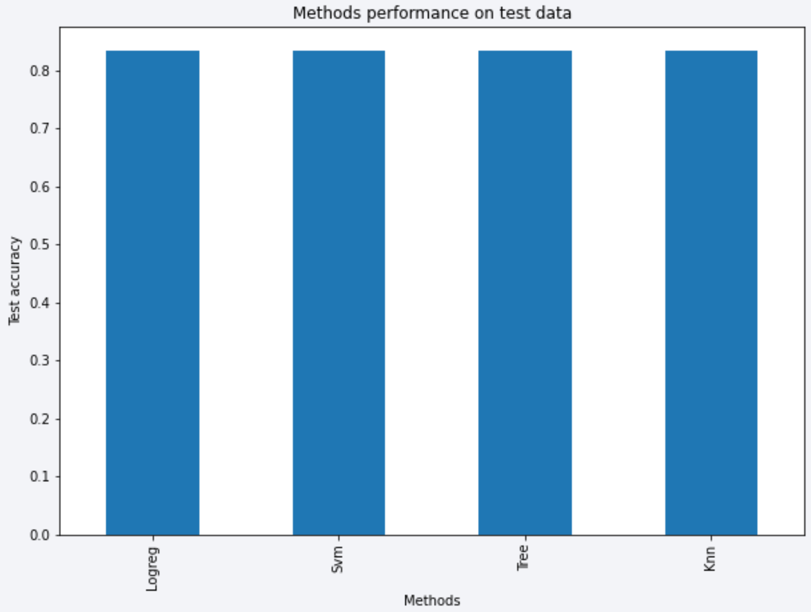
Section 6

Predictive Analysis (Classification)

ClassificationAccuracy

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





For accuracytest, all methods performed similar. Wecould get more test data to decide between them. But if we really need to chooseone right now, we would take the decision tree.

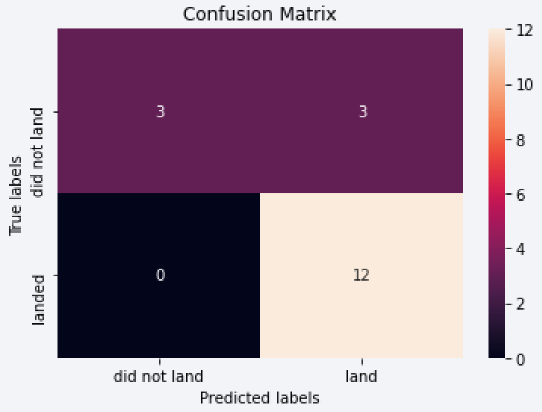
Decision tree best parameters

```
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

As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

Conclusions

The success of a mission can be attributed to various factors, including the launch site, orbit, and notably, the number of previous launches. It is reasonable to assume that accumulated knowledge between launches contributes to the transition from launch failures to successes. Orbits with the highest success rates include GEO, HEO, SSO, and ES-L1.

Payload mass can also significantly impact mission success depending on the orbit. Certain orbits necessitate either light or heavy payload masses. However, in general, missions with lower payload masses tend to perform better than those with heavier payloads.

At present, the dataset does not provide explanations for why certain launch sites outperform others, such as KSC LC-39A being the top-performing site. Obtaining additional atmospheric or relevant data could help shed light on this issue.

Despite identical test accuracies across all models used, we select the Decision Tree Algorithm as the preferred model for this dataset. This decision is based on its superior train accuracy.

Thank you!