Winning Space Race with Data Science

Beatriz Angelica Torres
16/11/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive Analytics in screenshots
  - Predictive Analytics

Introduction

- Project background and context
  - The objective of this project is to forecast the successful landing of the Falcon 9 first stage. According to SpaceX's official website, the cost of a Falcon 9 rocket launch is stated at 62 million dollars, a notable contrast to other providers whose charges exceed 165 million dollars per launch. The substantial price disparity is attributed to SpaceX's innovative approach of reusing the first stage. By ascertaining the landing outcome, we can derive the cost associated with each launch. This information holds significance for other companies intending to compete with SpaceX in the rocket launch market.
- Problems you want to find answers
  - What are the primary attributes distinguishing a successful from a failed landing?
  - How do the interrelations among rocket variables influence the outcome of a landing?
  - What conditions need to be in place for SpaceX to attain the highest possible landing success rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Information was gathered through the SpaceX REST API and web scraping from Wikipedia.
- Perform data wrangling
  - Eliminating redundant columns
  - Employing one-hot encoding for categorical feature representation, data underwent processing 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
  - Building, fine-tuning, and assessing classification models

Data Collection

Data collection involves the systematic gathering and measurement of information related to targeted variables within a defined system, facilitating the ability to address pertinent questions and assess outcomes. In this context, the dataset was acquired through both REST API and web scraping from Wikipedia.

For the REST API, the process begins with a GET request. Subsequently, the response content is decoded as JSON, and utilizing json_normalize(), it is transformed into a pandas dataframe. Following this, data cleaning procedures are implemented, including the identification and handling of missing values.

Regarding web scraping, BeautifulSoup is employed to extract launch records from HTML tables. The extracted data is then parsed and converted into a pandas dataframe, facilitating further analysis.

Data Collection – SpaceX API


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

```python
response = requests.get(spacex_url)
```

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

```python
# Lets take a subset of our dataframe keeping only the features we want a
nd the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number',
'date_utc']]

# We will remove rows with multiple cores because those are falcon rocket
s with 2 extra rocket boosters and rows that have multiple payloads in a
single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the s
ingle value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then ex
tracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Link Here

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from url

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

Create a BeautifulSoup from the HTML response

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

Extract all column/variable names from the HTML header

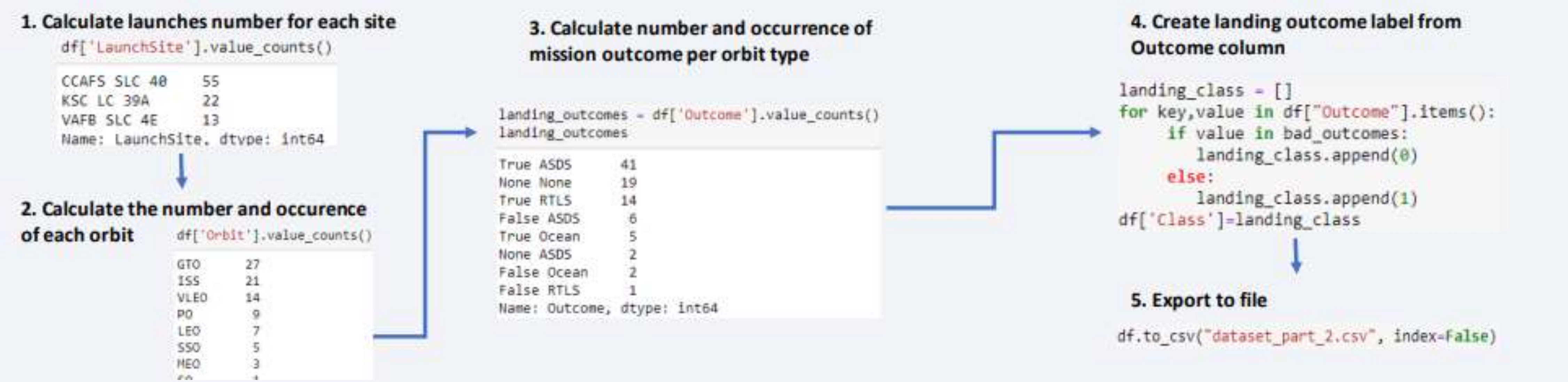
```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plai
nrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding t
o launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

[Link Here]()

Data Wrangling

Within the dataset, instances exist where the booster did not achieve a successful landing.

- "True Ocean," "True RTLS," and "True ASDS" collectively signify a successful mission.
- Conversely, "False Ocean," "False RTLS," and "False ASDS" indicate a mission failure.
- It is imperative to convert string variables into categorical variables, assigning the value of 1 to denote mission success and 0 to signify mission failure.



Link Here

EDA with Data Visualization

Our initial approach involved employing scatter graphs to examine relationships between various attributes, including:

- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

Scatter plots effectively illustrate the interdependencies among these attributes. By discerning patterns from the graphs, it becomes straightforward to identify the factors that exert the greatest influence on the success of landing outcomes.

Link Here

EDA with SQL

We executed SQL queries to collect and comprehend data from the dataset.

- SQL queries were executed to retrieve and comprehend data from the dataset:
- Displaying the names of unique launch sites in space missions.
- Displaying 5 records where launch sites start with the string 'CCA.'
- Displaying the total payload mass carried by boosters launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date of the first successful landing outcome on a ground pad.
- Listing the names of boosters with success on a drone ship and a payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failed mission outcomes.
- Listing the names of booster versions that have carried the maximum payload mass.
- Listing records displaying month names, failure landing outcomes on a drone ship, booster versions, and launch sites for the months in the year 2015.
- Ranking the count of successful landing outcomes between the dates 04-06-2010 and 20-03-2017 in descending order.

Link Here

Build an Interactive Map with Folium

To visualize the launch data on an interactive map, we extracted latitude and longitude coordinates for each launch site and incorporated a circle marker around each location, labeled with the launch site's name. Subsequently, we assigned the dataframe launch_outcomes (categorized as failure and success) to classes 0 and 1, represented by Red and Green markers on the map, organized using MarkerCluster.

- Next, we utilized Haversine's formula to calculate the distance between launch sites and various landmarks to address questions such as:
- Proximity of launch sites to railways, highways, and coastlines.
- Proximity of launch sites to nearby cities.

Link Here

Build a Dashboard with Plotly Dash

The dashboard comprises dropdown, pie chart, rangeslider, and scatter plot components.

- The dropdown enables users to choose either a specific launch site or all launch sites (dash_core_components.Dropdown).
- The pie chart illustrates the total success and failure for the selected launch site using the dropdown component (plotly.express.pie).
- The rangeslider enables users to select a payload mass within a predefined range (dash_core_components.RangeSlider).
- The scatter chart visually represents the relationship between two variables, specifically Success vs. Payload Mass (plotly.express.scatter).

Link Here

Predictive Analysis (Classification)

Data Preparation:

- Import the dataset.
- Normalize the data.
- Divide the data into training and test sets.

Model Preparation:

- Choose machine learning algorithms.
- Define parameters for each algorithm using GridSearchCV.
- Train GridSearchModel models with the training dataset.

Model Evaluation:

- Obtain the best hyperparameters for each model type.
- Calculate accuracy for each model using the test dataset.
- Generate a Confusion Matrix plot.

[Link Here]

Model Comparison:

- Compare models based on their accuracy.
- Select the model with the highest accuracy (refer to the notebook for detailed results).

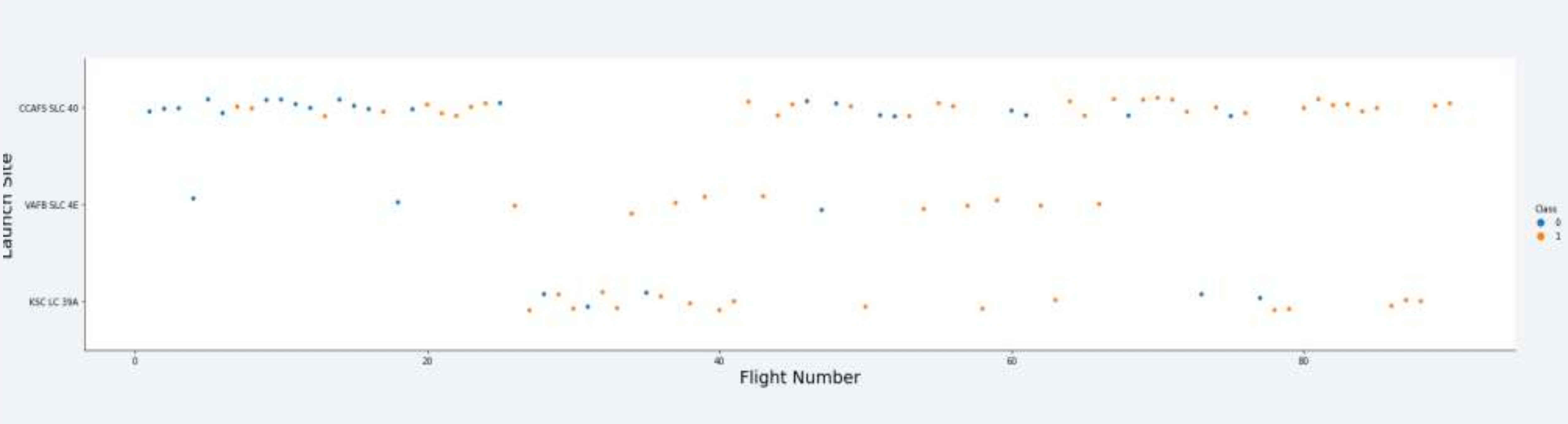
Results

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

Section 2

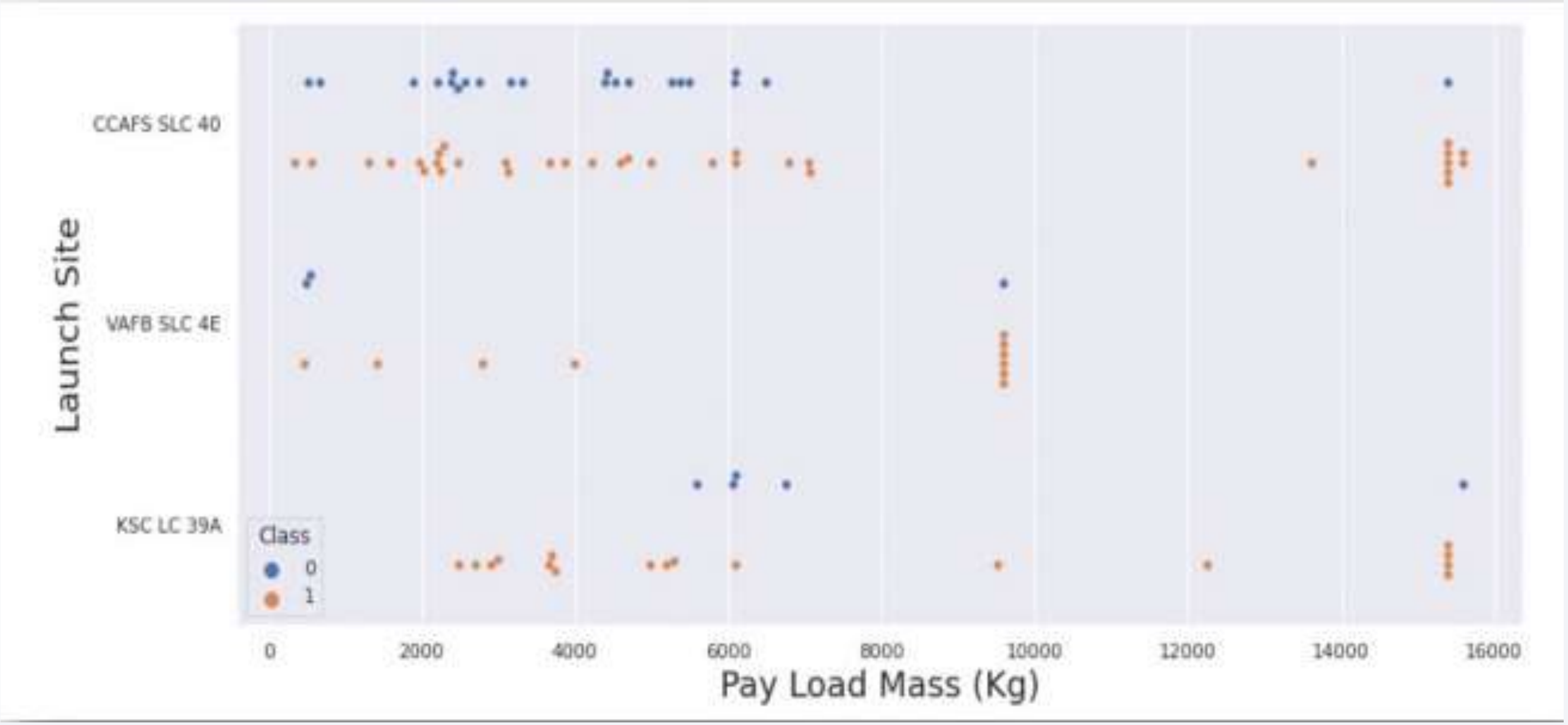
Insights drawn from EDA

Flight Number vs. Launch Site



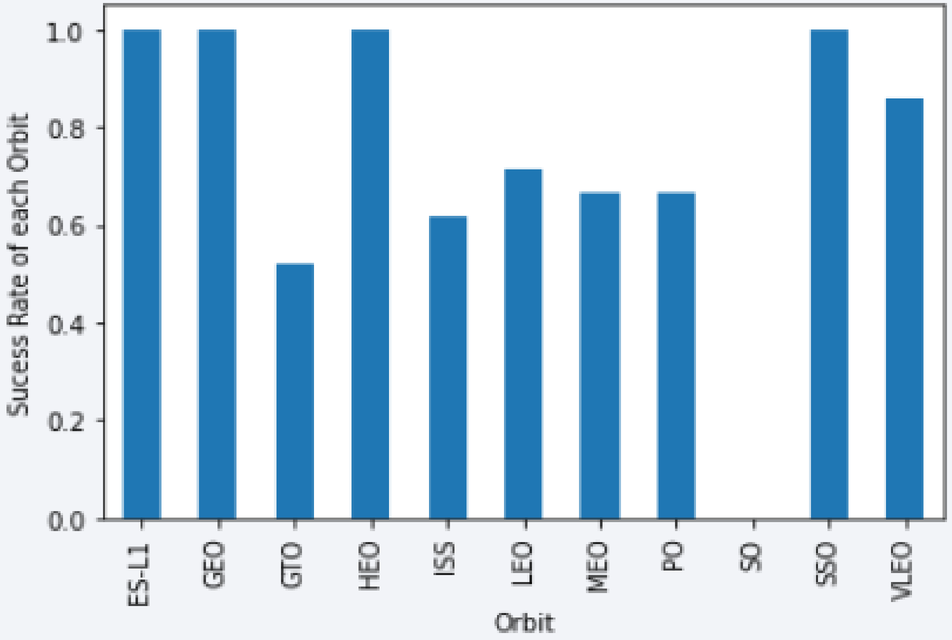
- This scatter plot indicates a positive correlation between the number of flights from a launch site and the corresponding success rate. Generally, an increase in the volume of flights is associated with a higher success rate. Notably, site CCAFS SLC40 deviates from this trend, displaying a less pronounced pattern in this regard.

Payload vs. Launch Site



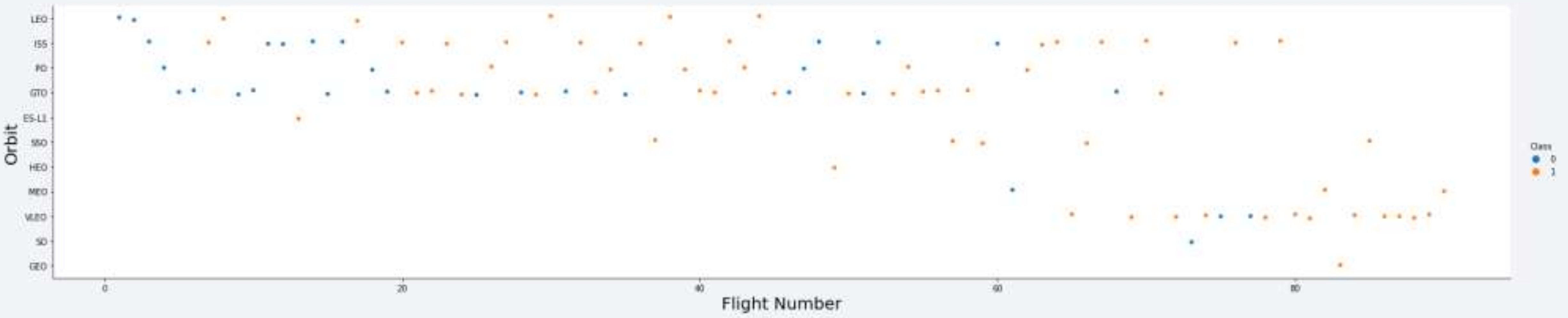
- The success of a landing may be contingent on the launch site, where a heavier payload could be a contributing factor. Conversely, an excessively heavy payload has the potential to result in a failed landing.

Success Rate vs. Orbit Type



- This illustration highlights the potential impact of different orbits on landing outcomes. Certain orbits, such as SSO, HEO, GEO, and ES-L1, exhibit a 100% success rate, whereas the SO orbit registers a 0% success rate. However, upon closer examination, it becomes apparent that some of these orbits have only one occurrence, including GEO, SO, HEO, and ES-L1. This limited dataset suggests the need for additional data to discern patterns or trends before drawing any conclusive

Flight Number vs. Orbit Type

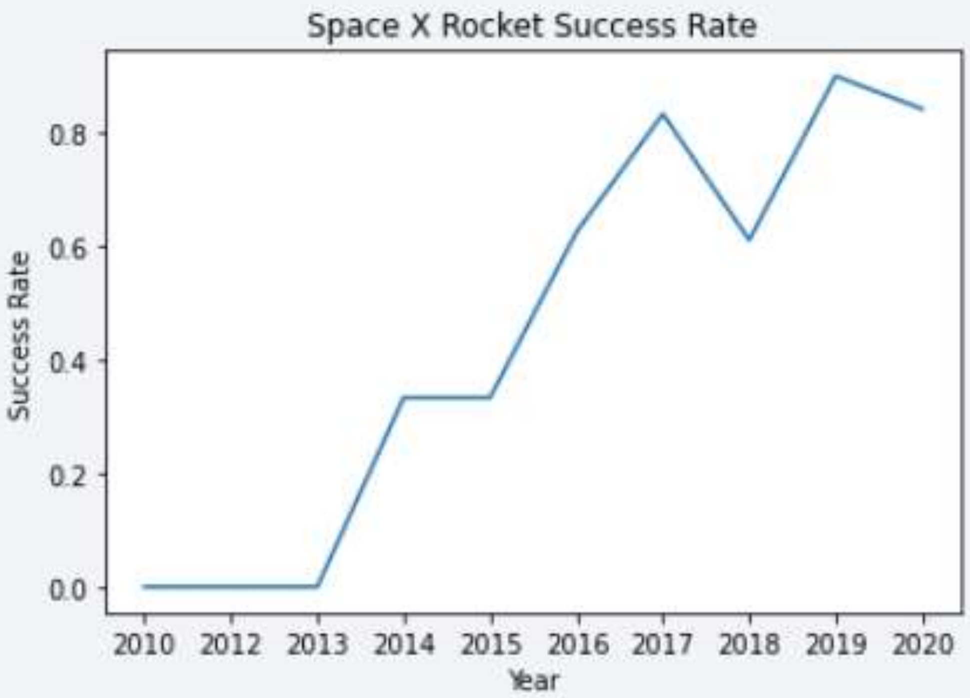


- This scatter plot indicates a positive correlation between the flight number and success rate in most orbits, particularly in LEO. However, GTO orbit shows no clear relationship between the two attributes. Orbits with only one occurrence should be excluded from the above statement, as they require additional data for meaningful analysis.

Payload vs. Orbit Type

- Payload weight significantly impacts launch success rates in specific orbits. Heavier payloads enhance success rates in the LEO orbit, while decreasing payload weight in a GTO orbit improves the likelihood of a successful launch.

Launch Success Yearly Trend



- These figures clearly show a rising trend from 2013 to 2020. If this trend continues in the coming years, the success rate is expected to steadily increase, potentially reaching a 100% success rate.

All Launch Site Names

```
In [5]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;

 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[5]:

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

- We employed the keyword DISTINCT to display only unique launch sites within the SpaceX data.

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [11]:

```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	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 WHERE clause, coupled with the LIKE clause, filters launch sites containing the substring "CCA." The LIMIT 5 command displays the first five records from this filtered result.

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Total Payload Mass by NASA (CRS)
45596

- We computed the total payload carried by NASA boosters as 45596 using the following query.

Average Payload Mass by F9 v1.1

- We determined the average payload mass carried by the booster version F9 v1.1 to be 2928.4.

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

- We employed the min() function to obtain the result. It was observed that the date of the first successful landing outcome on the ground pad was December 22, 2015.

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query retrieves the booster version for which the landing was successful, and the payload mass falls between 4000 and 6000 kg. The WHERE and AND clauses are utilized to filter the dataset accordingly.

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Total Number of Successful and Failure Mission Outcomes

- In the initial SELECT statement, we present subqueries that yield results. The first subquery calculates the count of successful missions, while the second subquery counts the unsuccessful missions. The WHERE clause, coupled with the LIKE clause, filters the mission outcomes, and the COUNT function tallies the records that meet the specified criteria.

```
%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
```

SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

- We employed a subquery to filter the data by retrieving only the heaviest payload mass using the MAX function. The main query utilizes the results from the subquery, returning unique booster versions (SELECT DISTINC along with the heaviest payload mass.

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

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

em título]

2015 Launch Records

We applied a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes on drone ships, along with their corresponding booster versions and launch site names for the year 2015.

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.c
loud:32731/bludb
Done.
```

Landing Outcome	Total 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

- We chose landing outcomes and their counts from the data, using the WHERE clause to filter for outcomes between June 4, 2010, and March 20, 2010. Applying the GROUP BY clause, we grouped the landing outcomes and used the ORDER BY clause to arrange them in descending order.

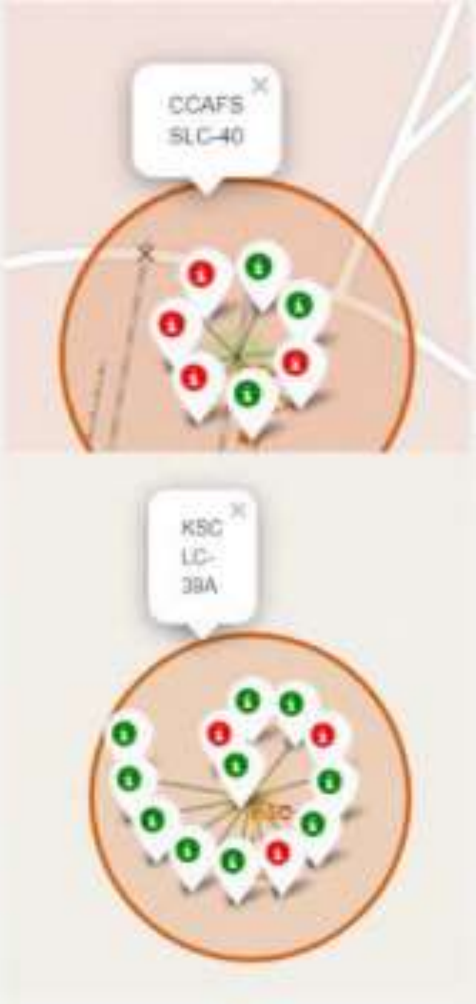
Section 3

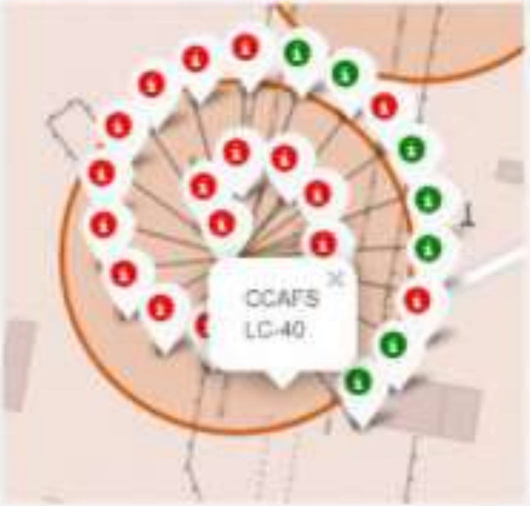
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

SpaceX launch sites are situated along the coastline of the United States.

<Folium Map Screenshot 2>



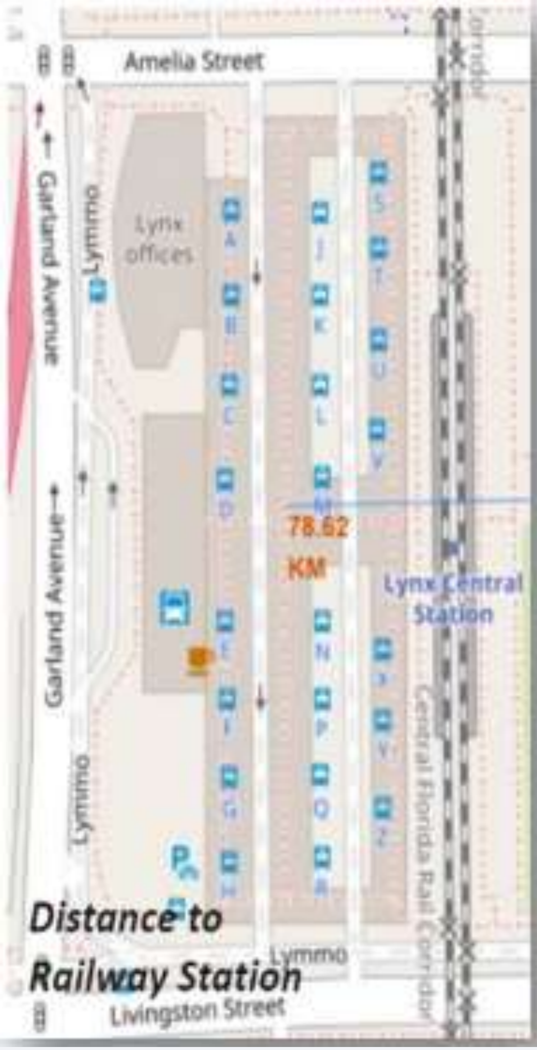


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

<Folium Map Screenshot 3>

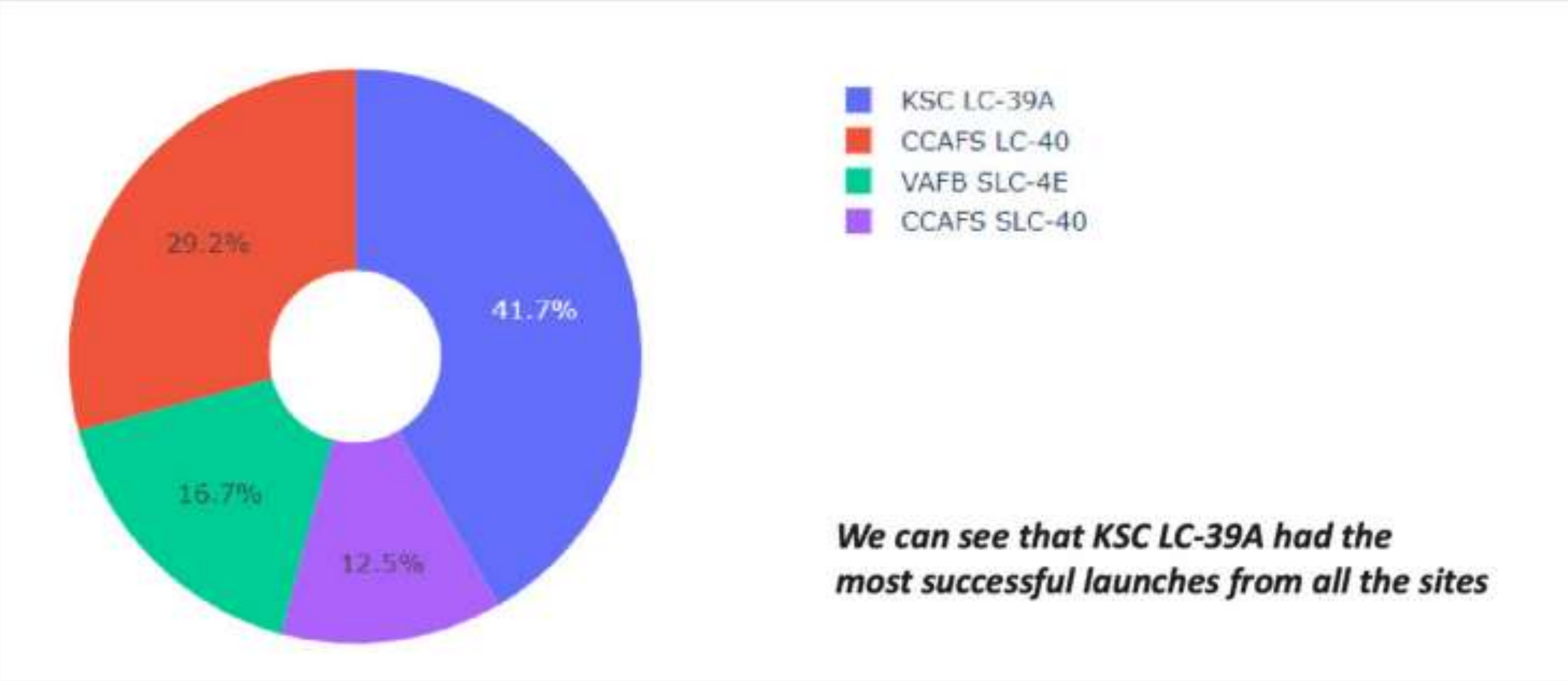


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

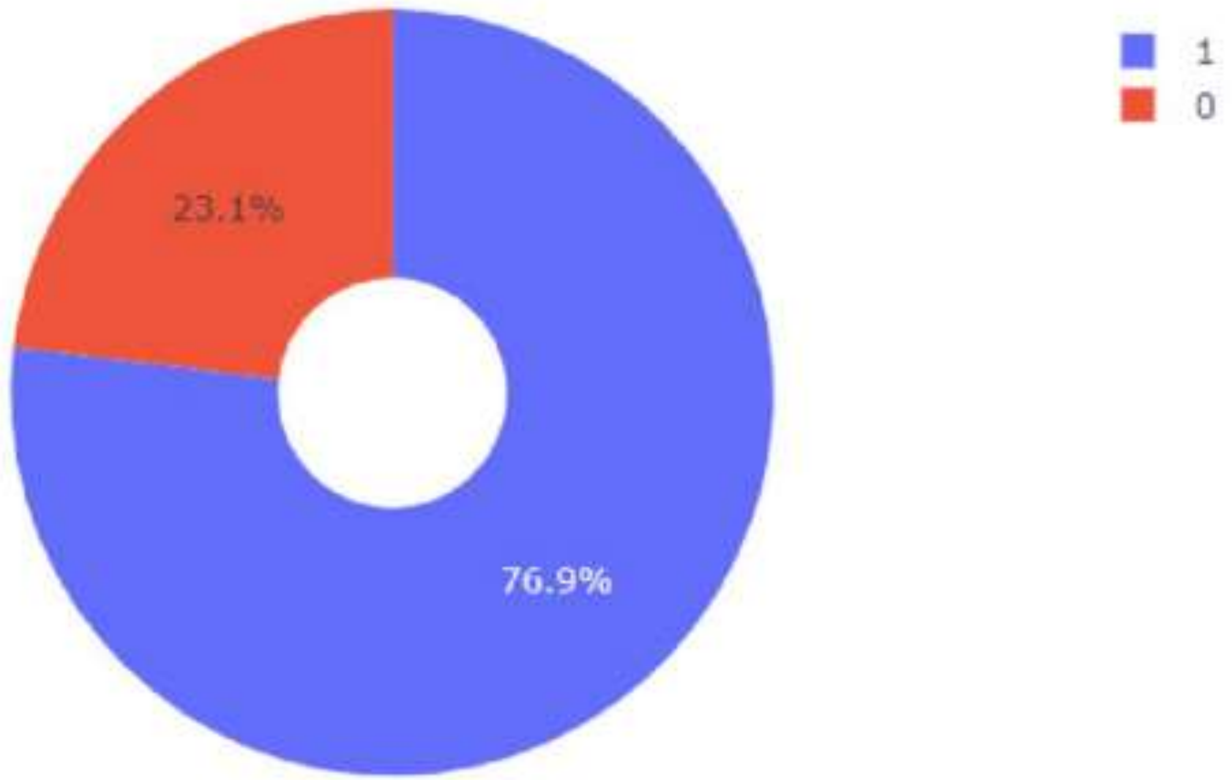
Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



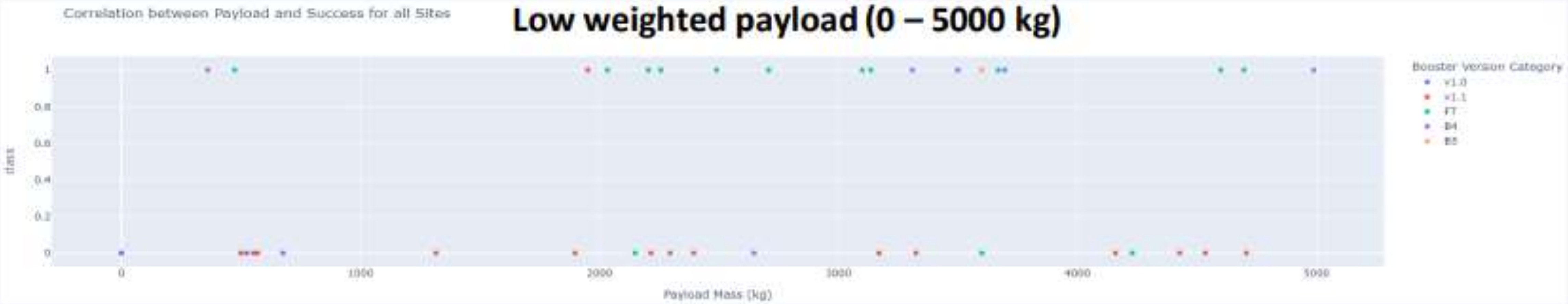
<Dashboard Screenshot 2>



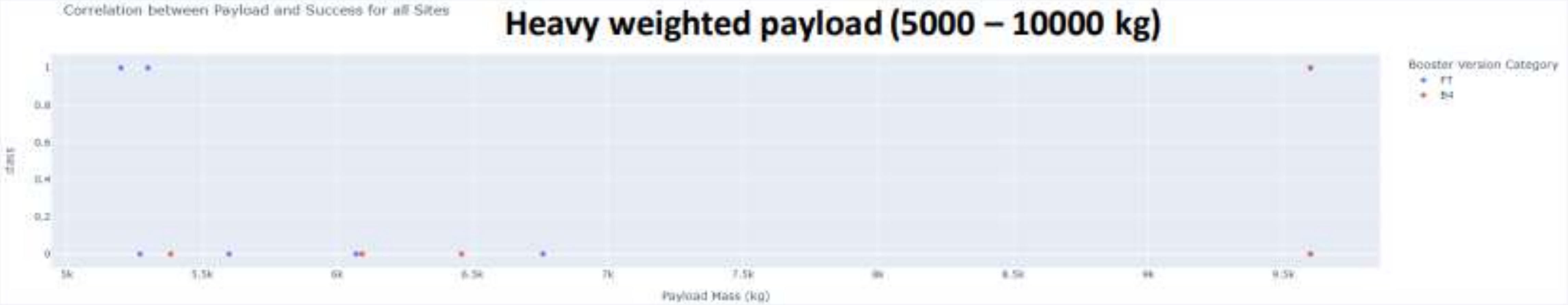
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Dashboard Screenshot 3>

Low weighted payload (0 – 5000 kg)



Heavy weighted payload (5000 – 10000 kg)

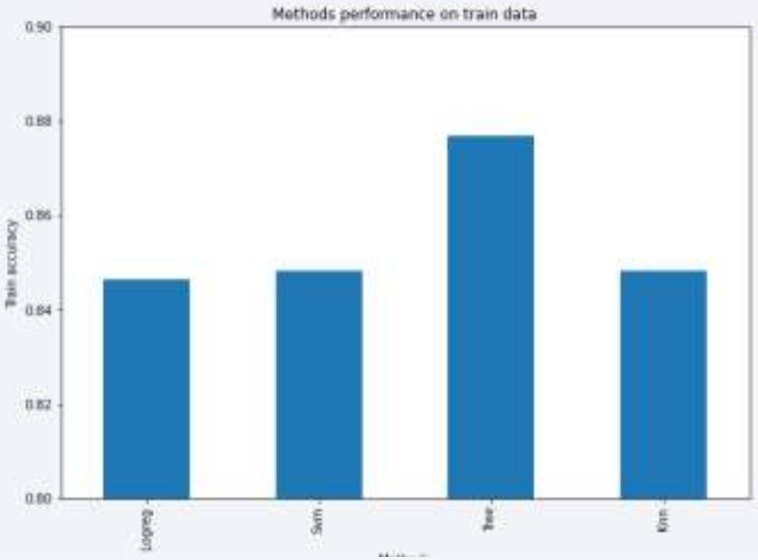


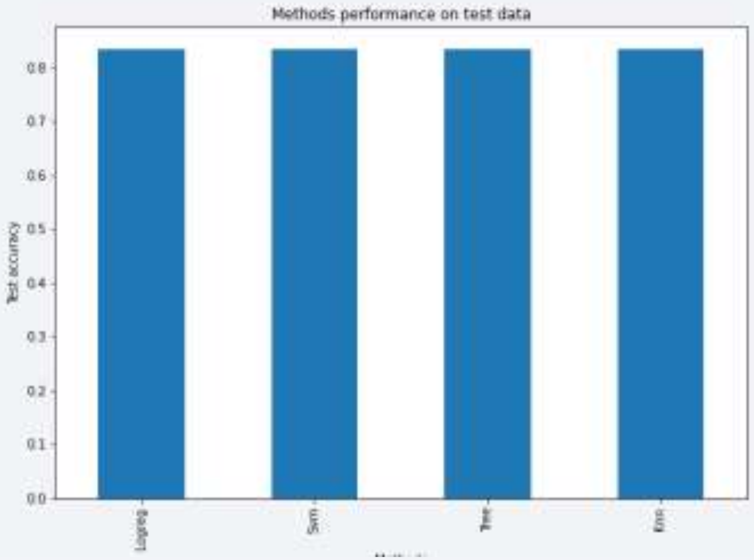
Section 5

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

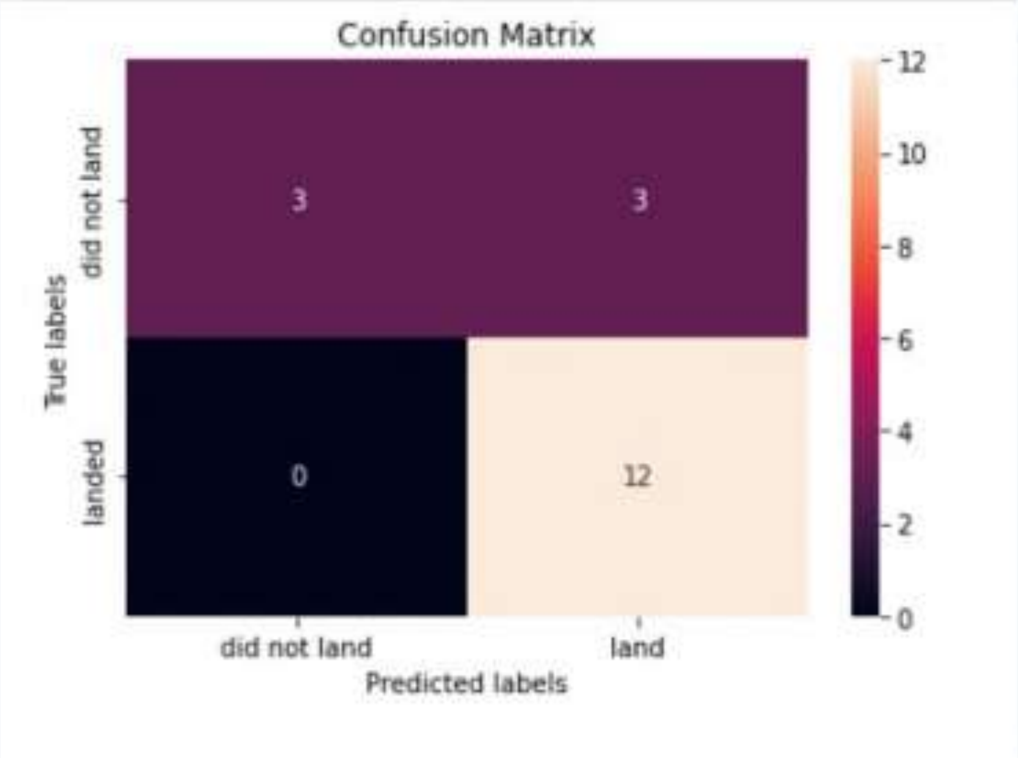


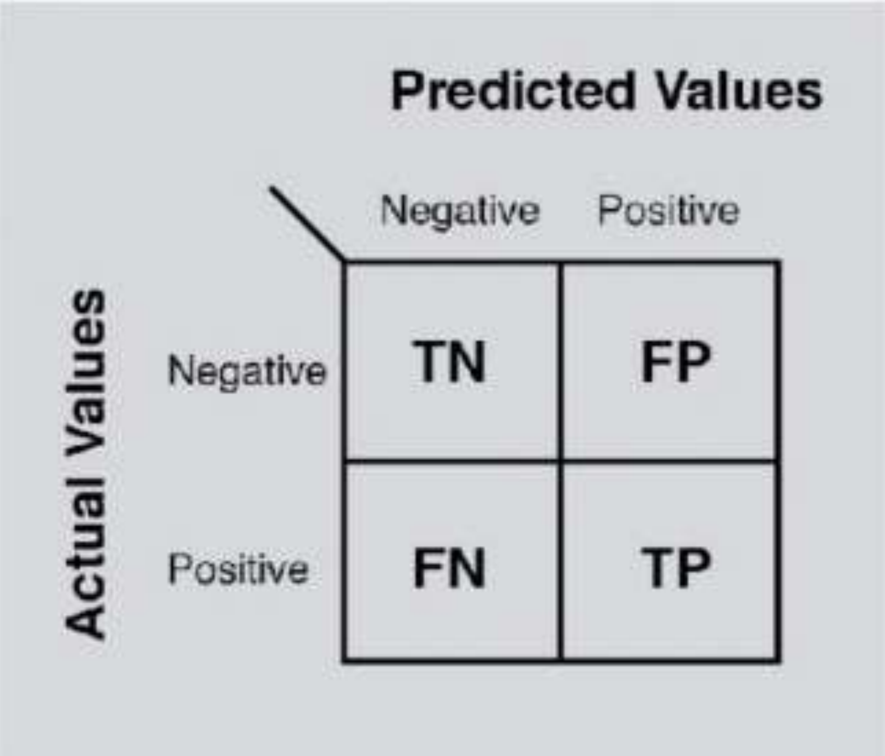


In the accuracy test, all methods demonstrated similar performance. Obtaining additional test data could help in making a more informed decision between them. However, if an immediate choice is necessary, the decision tree method would be preferred.

Confusion Matrix

The decision tree classifier's confusion matrix indicates effective class differentiation, with a notable issue of false positives—instances where unsuccessful landings are incorrectly identified as successful by the classifier.





Conclusions

Mission success is influenced by factors including the launch site, orbit, and notably, the number of previous launches. This suggests a potential knowledge gain over time, enabling the transition from launch failures to successful missions.

- Light payloads (defined as 4000kg and below) outperformed heavy payloads.
- The success rate of SpaceX launches has steadily increased since 2013, and this positive trend is expected to continue into the future.
- KSC LC-39A boasts the highest success rate among all launch sites, standing at 76.9%.
- The SSO orbit exhibits the highest success rate, reaching 100%, with more than one occurrence.

Thank you!