Winning Space Race with Data Science

AMAL SATHEESAN

08/11/2023

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
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Maps and Dashboard
  - Predictive Results

Introduction

- Project background and context
  - SpaceX has a unique advantage in the rocket launch industry. They can reuse their Falcon 9 first stage, significantly reducing launch costs compared to other providers. This cost-saving strategy has sparked interest in understanding the likelihood of a successful landing for the Falcon 9. This information is valuable not just for SpaceX but for any company considering rocket launches as it impacts overall launch costs.
- Problems you want to find answers
  - What are the key characteristics that distinguish successful landings from failed ones?
  - How do different rocket variables influence the likelihood of a successful landing?
  - What conditions can SpaceX optimize to achieve the highest landing success rate?

SECTION I

Methodology

Methodology

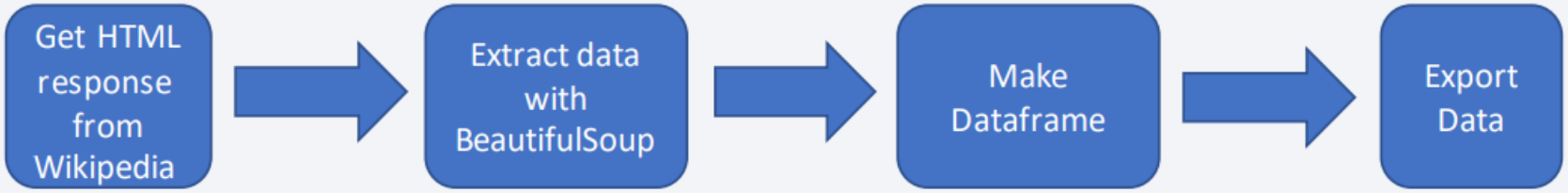
Executive Summary

- 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

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
  - The information obtained by the API are rocket, launches, payload information.
    - The Space X REST API URL 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 the 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 the 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 the 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 the Code

EDA with SQL

We performed SQL queries to gather and understand data from dataset:

Link to the Code

- Displaying the names of the unique launch 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_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

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 the 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 the Code

Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset

Link to the Code

- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen (see Notebook for result)

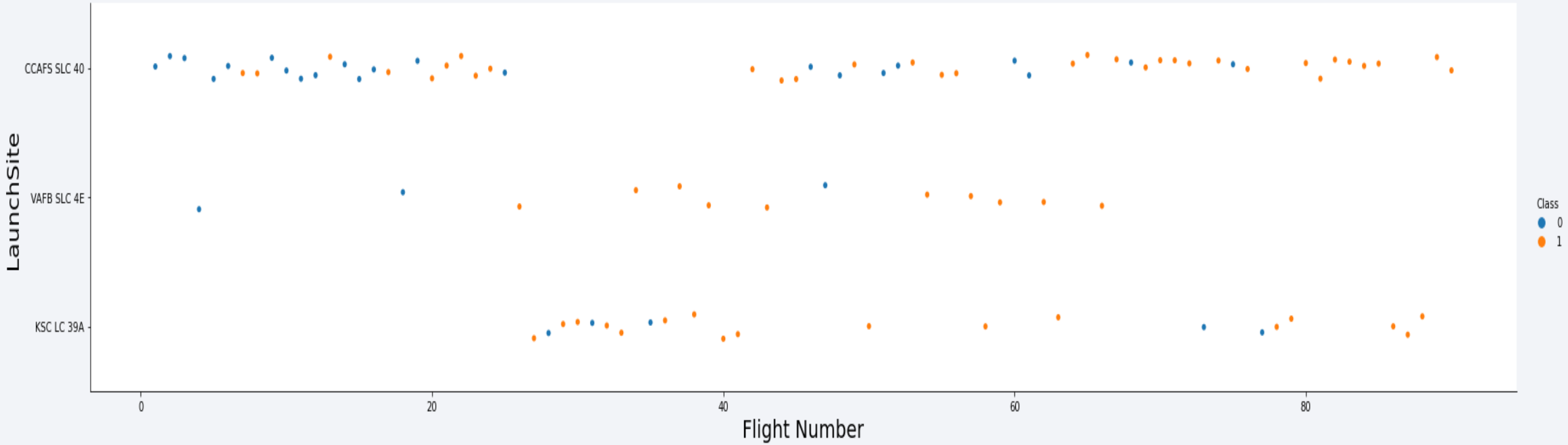
Results

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

SECTION II

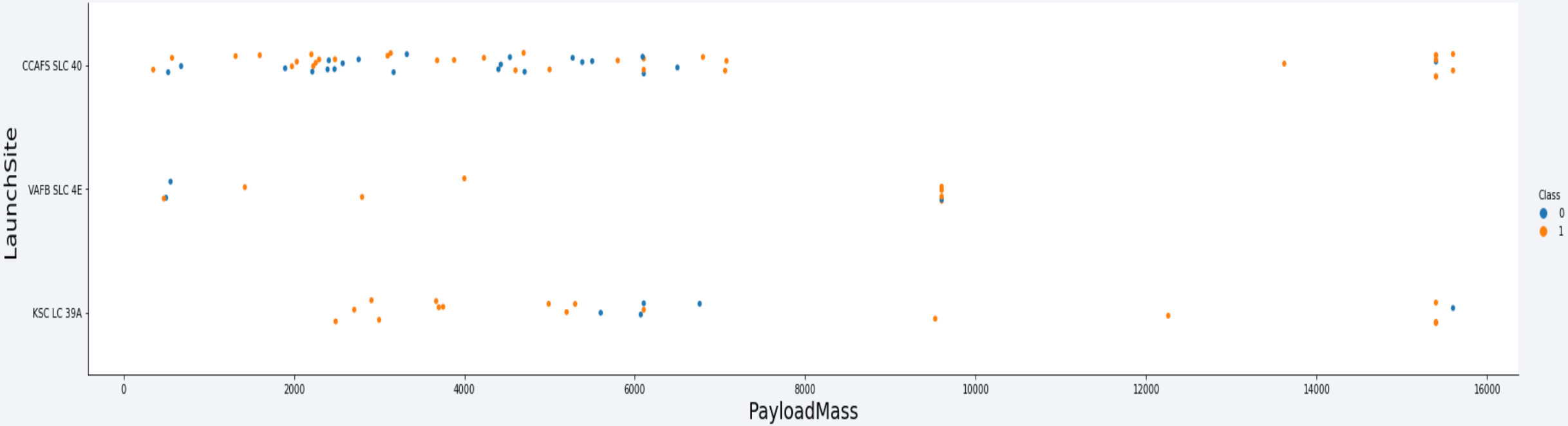
Insights Drawn From EDA

Flight Number vs. Launch Site



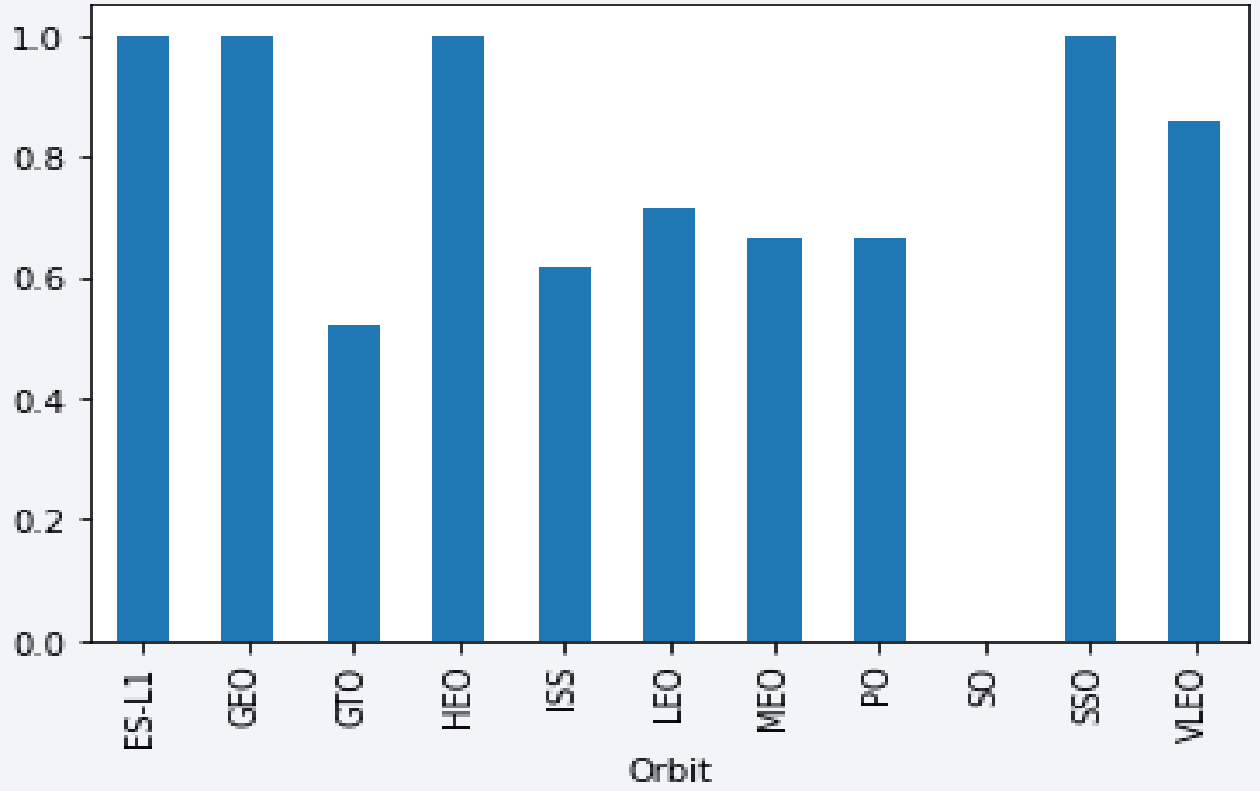
We Observe that for each site,the success rate

Payload vs Launch Site



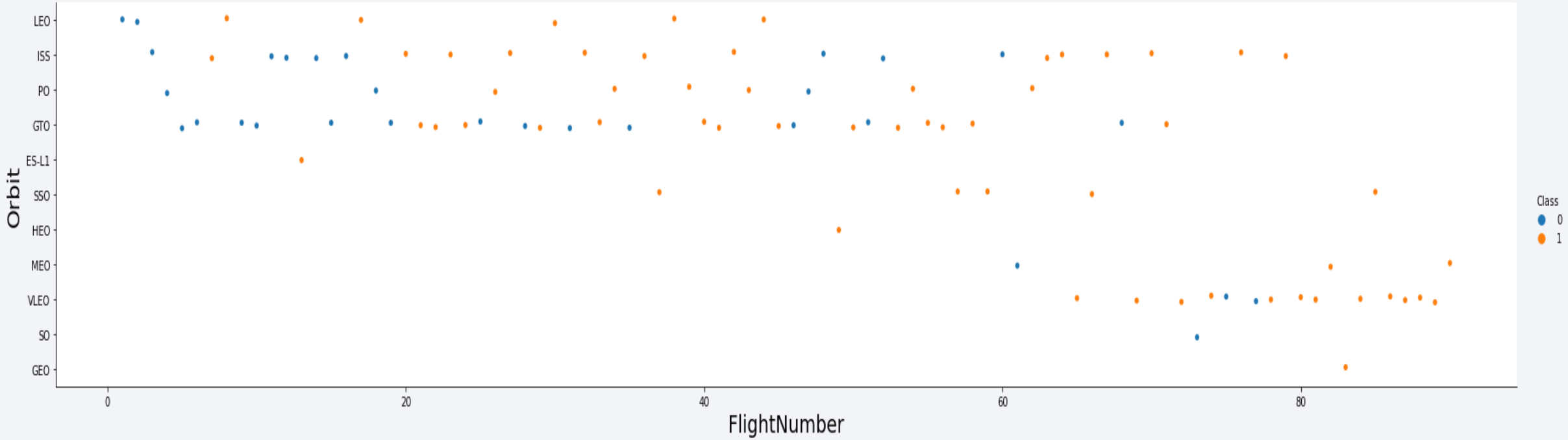
The choice of launch site plays a crucial role in determining whether a heavier payload will lead to a successful landing. Conversely, an excessively heavy payload can increase the risk of a failed landing.

Success Rate vs. Orbit Type



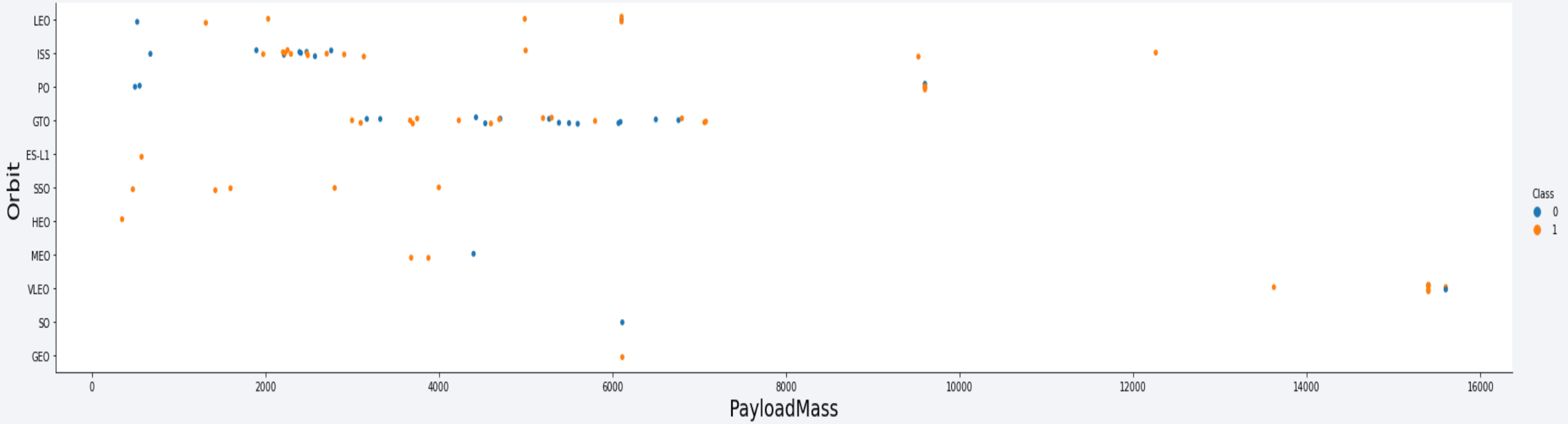
With this plot, we can see success rate for different orbit types.We note that ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



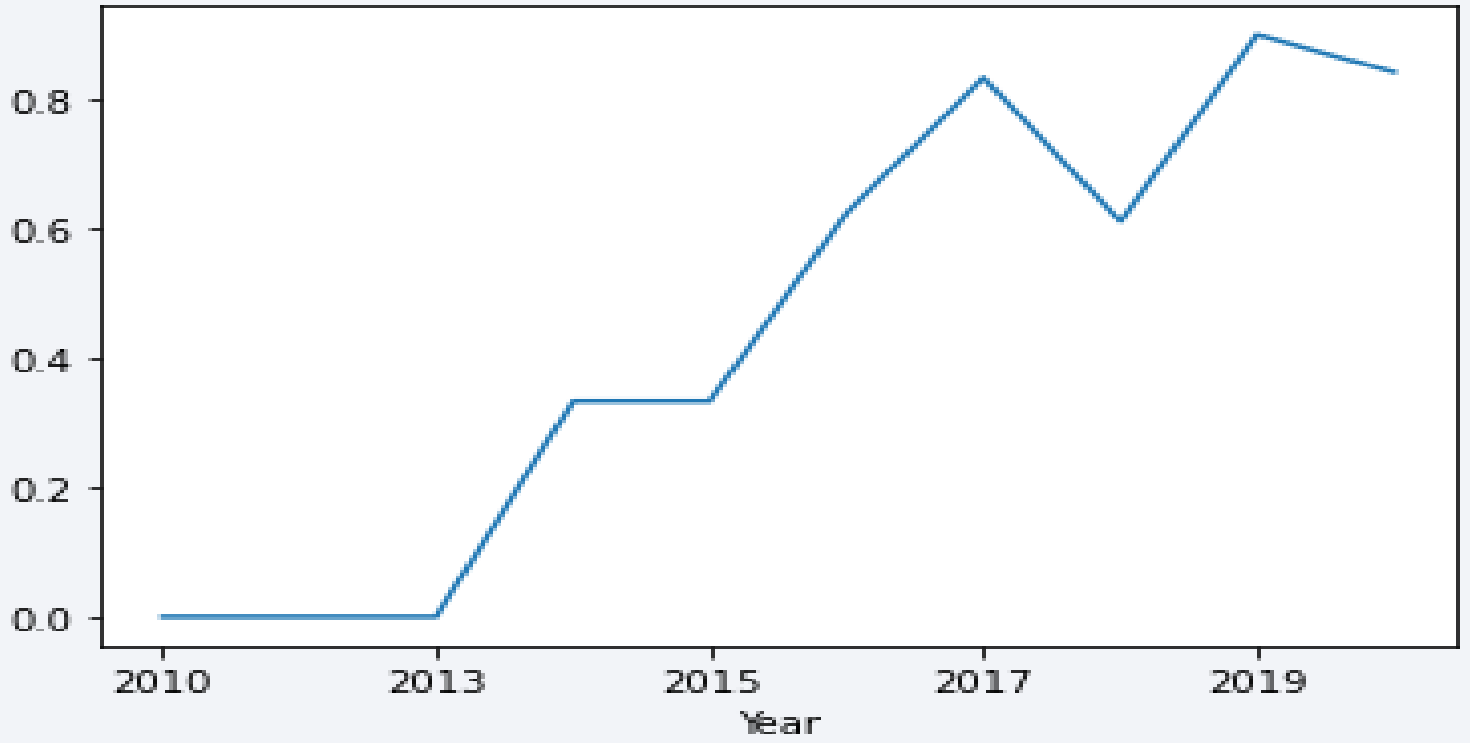
While the success rate tends to increase with the number of flights for Low Earth Orbit (LEO) missions, there appears to be no clear correlation between the success rate and the number of flights for missions to Geostationary Transfer Orbit (GTO). However, it's plausible that the high success rates observed in orbits like Sun-Synchronous Orbit (SSO) or Highly Elliptical Orbit (HEO) are influenced by knowledge gained from prior launches in different orbit types

Payload vs Orbit Type



The success rate of launches in specific orbits is significantly affected by the payload weight. Notably, heavier payloads tend to enhance the success rate for Low Earth Orbit (LEO) missions. Conversely, reducing the payload weight for launches to Geostationary Transfer Orbit (GTO) has a positive impact on launch success

Launch Success Yearly Trend



From 2013 onwards we can see an increase in the Space X Rocket Success rate

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

The DISTINCT Keyword in the query provides the unique values of the attribute “LAUNCH_SITE”

Launch Site Names Begin with 'CCA'

SQL Query

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

Explanation

The WHERE clause filters out the LAUNCH_SITES which have the substring “CCA” and we LIMIT’s to show the first 5 records only

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 total payload carried by boosters from NASA

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

This query returns the average payload mass carried by booster version F9 v1.

First Successful Ground Landing Date

SQL Query

```
SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

Results

FIRST_SUCCESS_GP
2015-12-22

Explanation

This query returns the dates of the first successful landing outcome on ground pad.

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

Above query returns the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

Results

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

Explanation

This query returns the total number of successful and failure mission outcomes

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

We employed a subquery to filter data, utilizing the MAX function to retrieve the highest payload mass. The primary query then utilizes the results from the subquery to identify distinct 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

This query retrieves the month, booster version, and launch site for unsuccessful landings that occurred in 2015. It processes the date using the Substr function to extract either the month or year. Substr(DATE, 4, 2) is used to extract the month, and Substr(DATE, 7, 4) extracts the year.

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

SQL Query

```
SELECT LANDING_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING_OUTCOME ORDER BY QTY DESC;
```

Results

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

Explanation

This query ranks the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

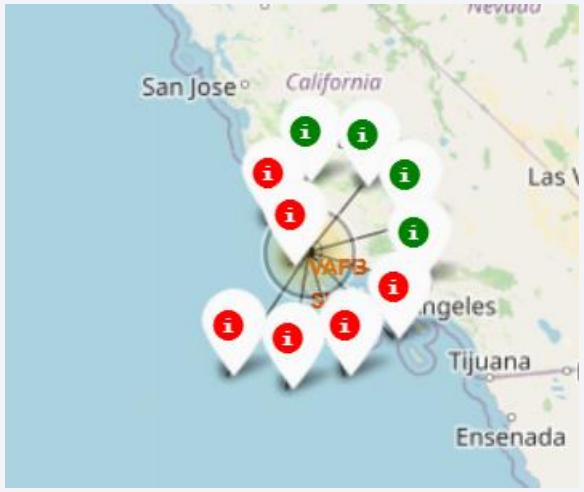
SECTION III

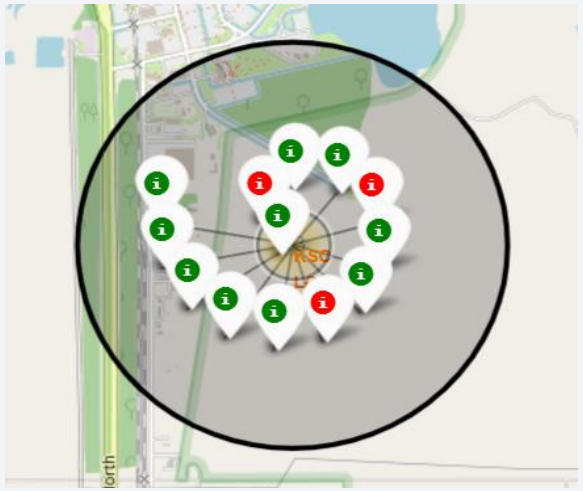
Launch Sites Proximities Analysis

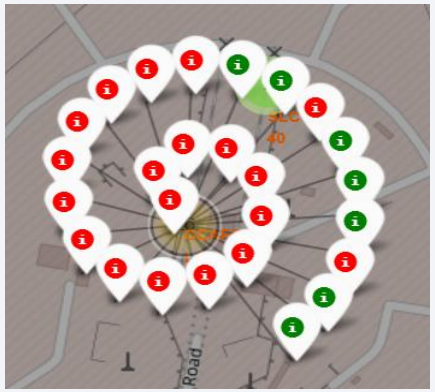
Launch Sites Location

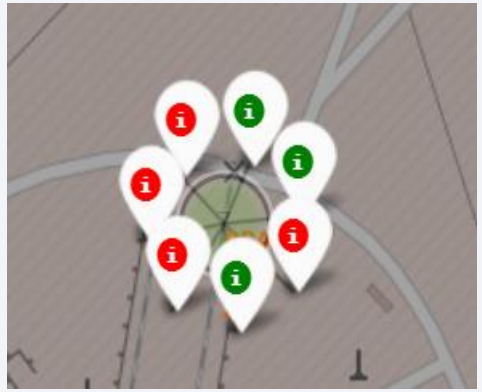
We see that Space X launch sites are located on the coast of the United States

Launch Sites And their Success rate



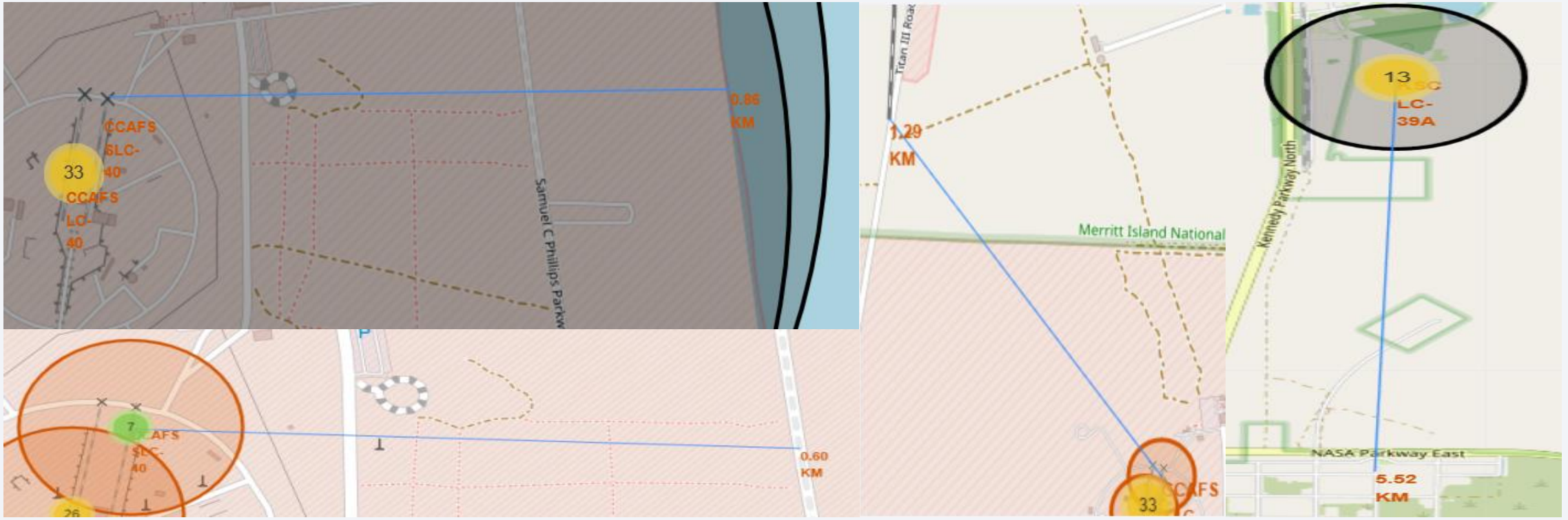






Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

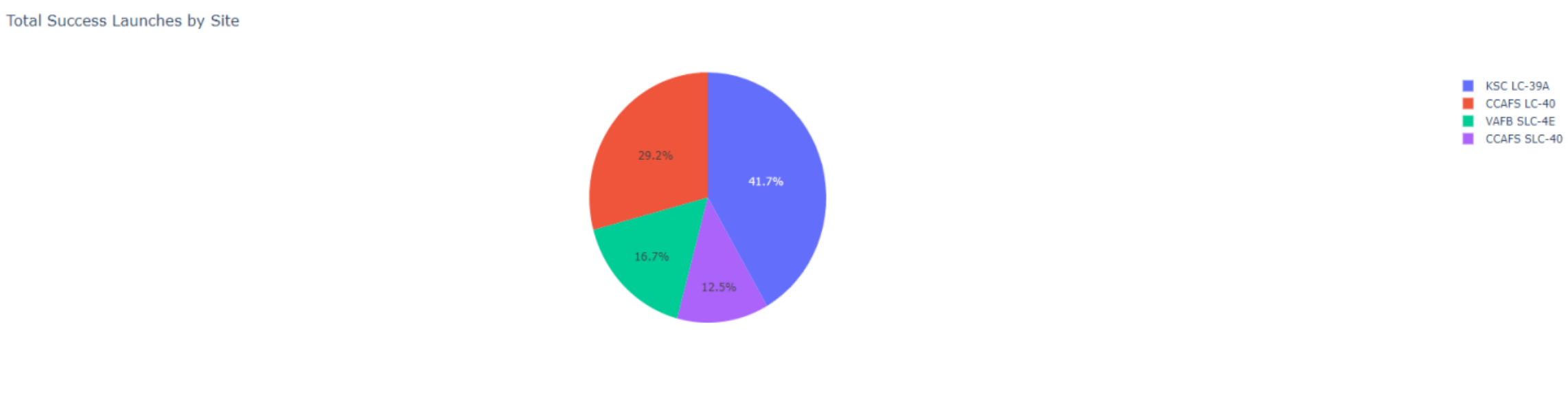
Launch sites and its Proximities



- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes except CCAFS SLC-40

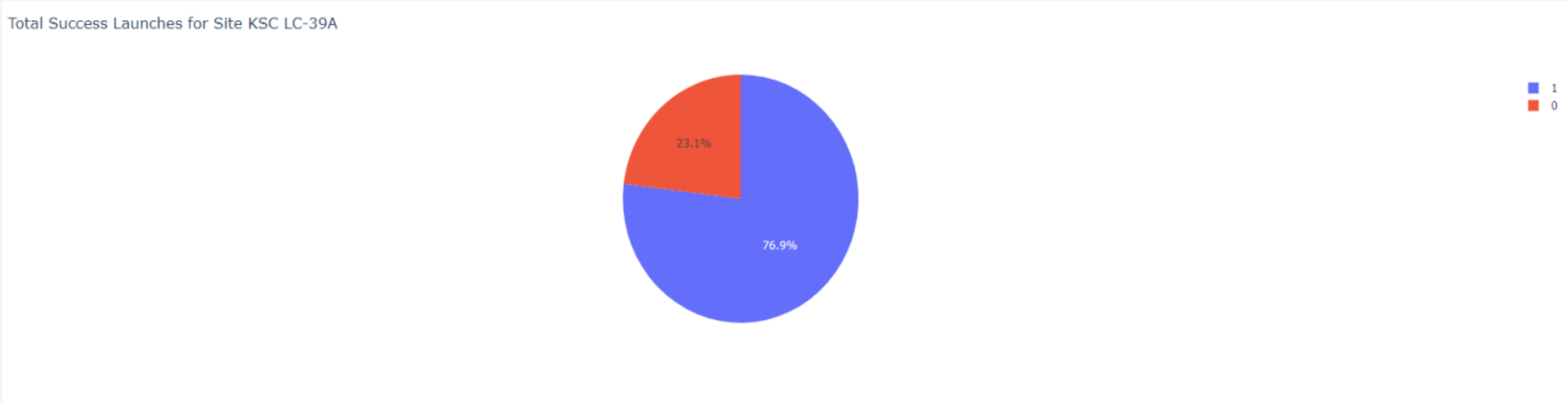

SECTION IV
Build a Dashboard with Plotly Dash

Dashboard – Total Success By Site



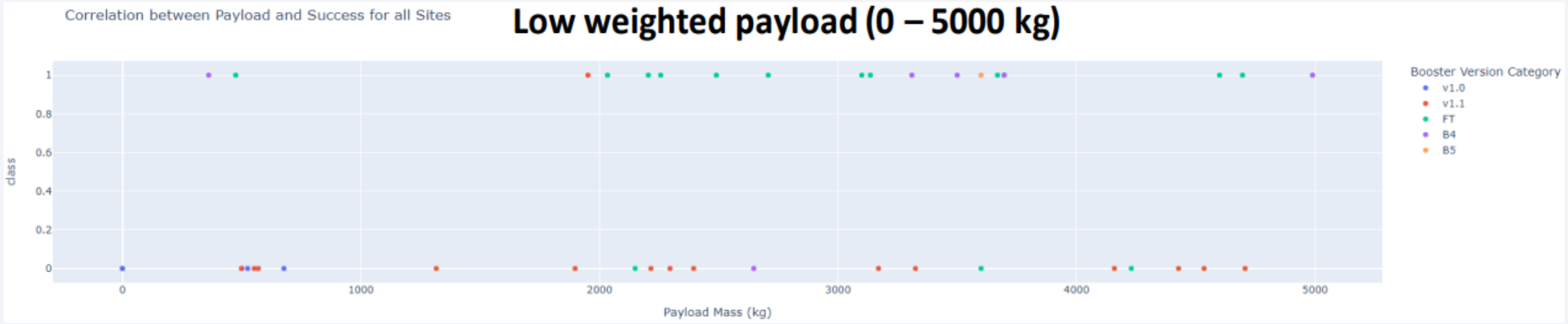
We see that KSC LC-39A has the best success rate of launches.

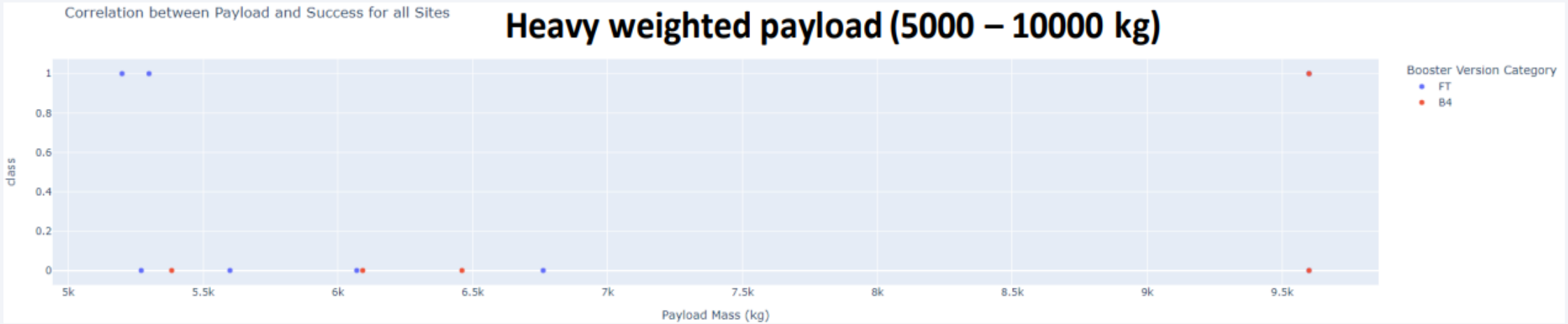
Dashboard – Total Success Launches For KSC LC-39A



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

Dashboard – Payload vs Outcome with different Mass

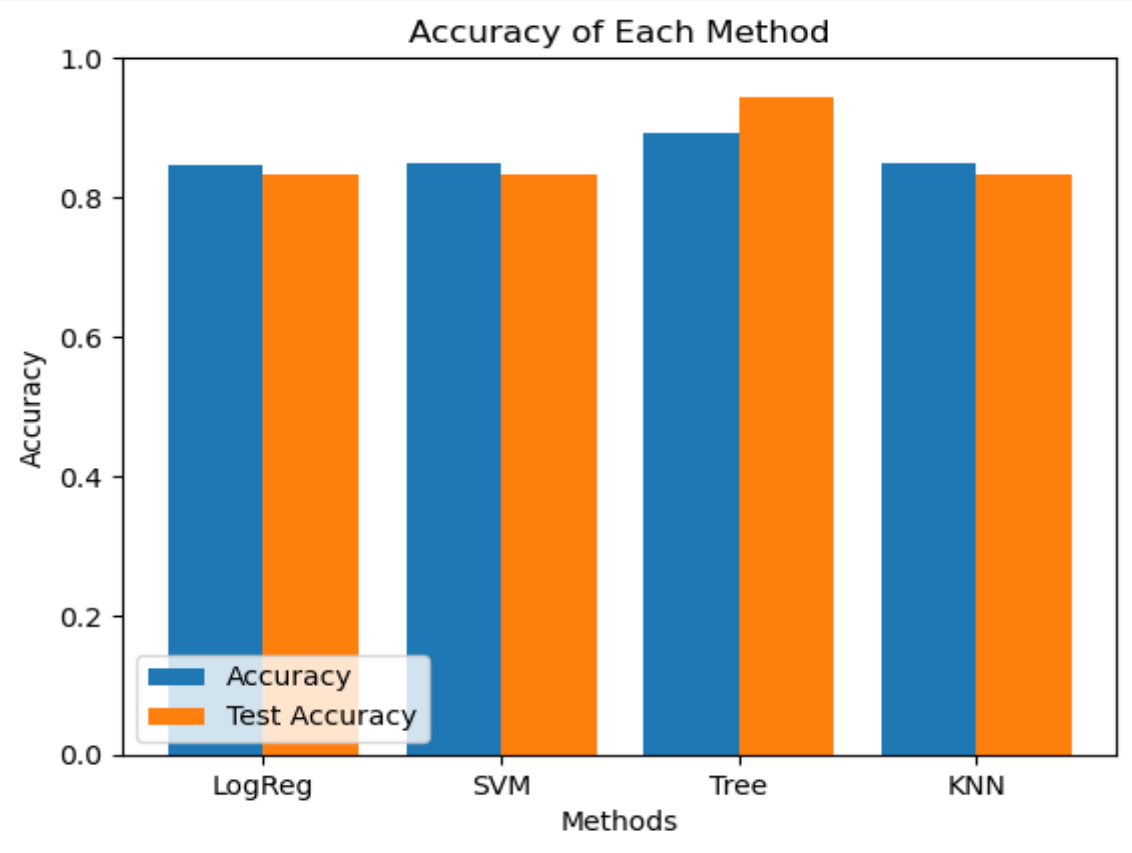




SECTION V

Predictive Analysis (Classification)

Classification Accuracy

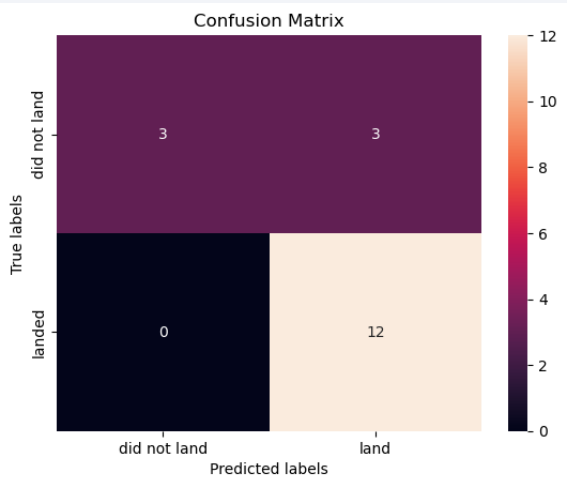


Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.89107	0.94444
KNN	0.84821	0.83333

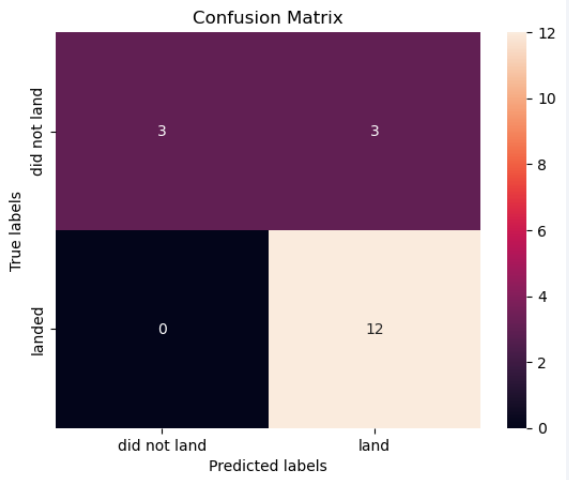
For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the Decision Tree.

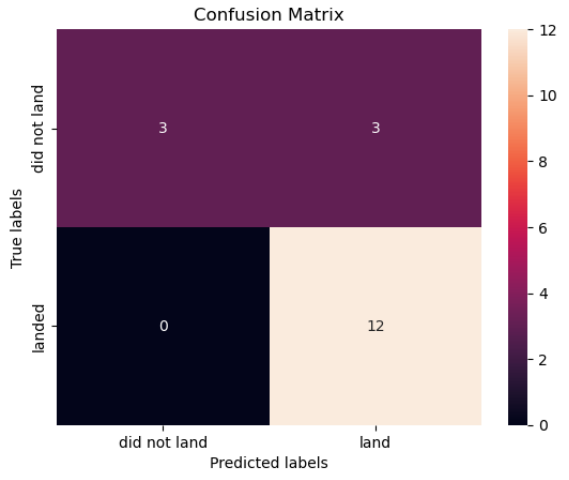
Confusion Matrix

Logistic regression

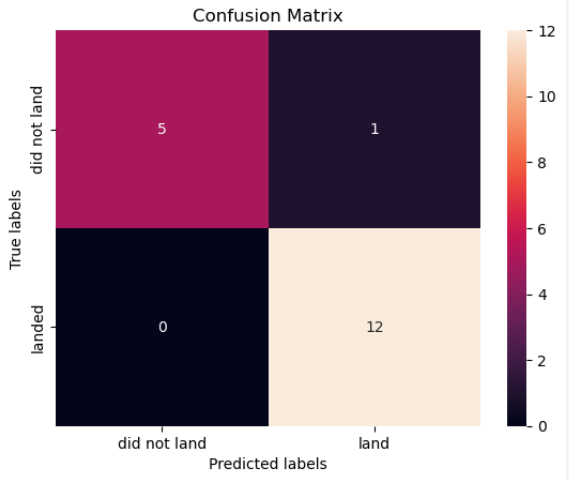


SVM





kNN



Decision Tree

As the test accuracy are almost equal except Decision Tree. The confusion matrices are also identical for others. The main problem of these models are false positives.

		Actual values 1	Actual values 0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

- The success of a mission is influenced by multiple factors, including the launch site, the orbit type, and, notably, the number of previous launches. It can be inferred that accumulated knowledge from previous launches contributes to improved success rates over time.
- The orbits with the highest success rates in this dataset are GEO (Geostationary Orbit), HEO (Highly Elliptical Orbit), SSO (Sun-Synchronous Orbit), and ES-L1 (Earth-Sun L1 Lagrange Point). These orbits exhibit consistent successful landings.
- Payload mass is a critical consideration for mission success, depending on the specific orbit. Some orbits require lighter payloads, while others may perform better with heavier payloads. Generally, missions with lower payload mass tend to have higher success rates.
- The dataset does not provide sufficient information to explain why certain launch sites are more successful than others. To address this, additional data such as atmospheric conditions or other relevant factors would be needed.
- After evaluating various machine learning models, the Decision Tree Algorithm was chosen as the best model for this dataset. While test accuracy was similar across different models, the Decision Tree Algorithm was selected for its slightly superior performance.

Appendix

- **GitHub Project Repository**
  - For complete project details, code, and notebooks, please visit our GitHub repository:
    - GitHub Repository
- **Dataset Source**
  - The dataset used for this project can be accessed from the following source:
    - Dataset Source
- **Python Code and Notebooks**
  - The Python code and Jupyter notebooks used for data analysis and modeling are available in our GitHub repository. You can find detailed code and explanations in the notebooks.


Thank You