IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marina Costa Lima

04/04/2023

Outline

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

Executive Summary

- **Data collection methodology:**

Methodologies used:

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

- **Summary of all results:**

- EDA results
- Interactive analytics
- Predictive analysis

Introduction

- **Project background and context:**

SpaceX publishes on its website the launches of the Falcon 9 rocket, with a cost of 62 million dollars; other suppliers charge more than $165 million each, and much of the savings is due to SpaceX's ability to reuse the first stage.

- **Problems you want to find answers:**

The project's objective is to predict whether the first stage of SpaceX's Falcon 9 rocket will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**

- SpaceX Rest API

- Web Scrapping from Wikipedia

- **The methodology was take SpaceX launch data that is gathered from an API, specifically the SpaceX REST API.**
- **Perform data wrangling**

- One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant columns

- **The data will be stored in lists and will be used to create our dataset.**
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
- - LR, KNN, SVM and DT models have been built and evaluated for the best classifier

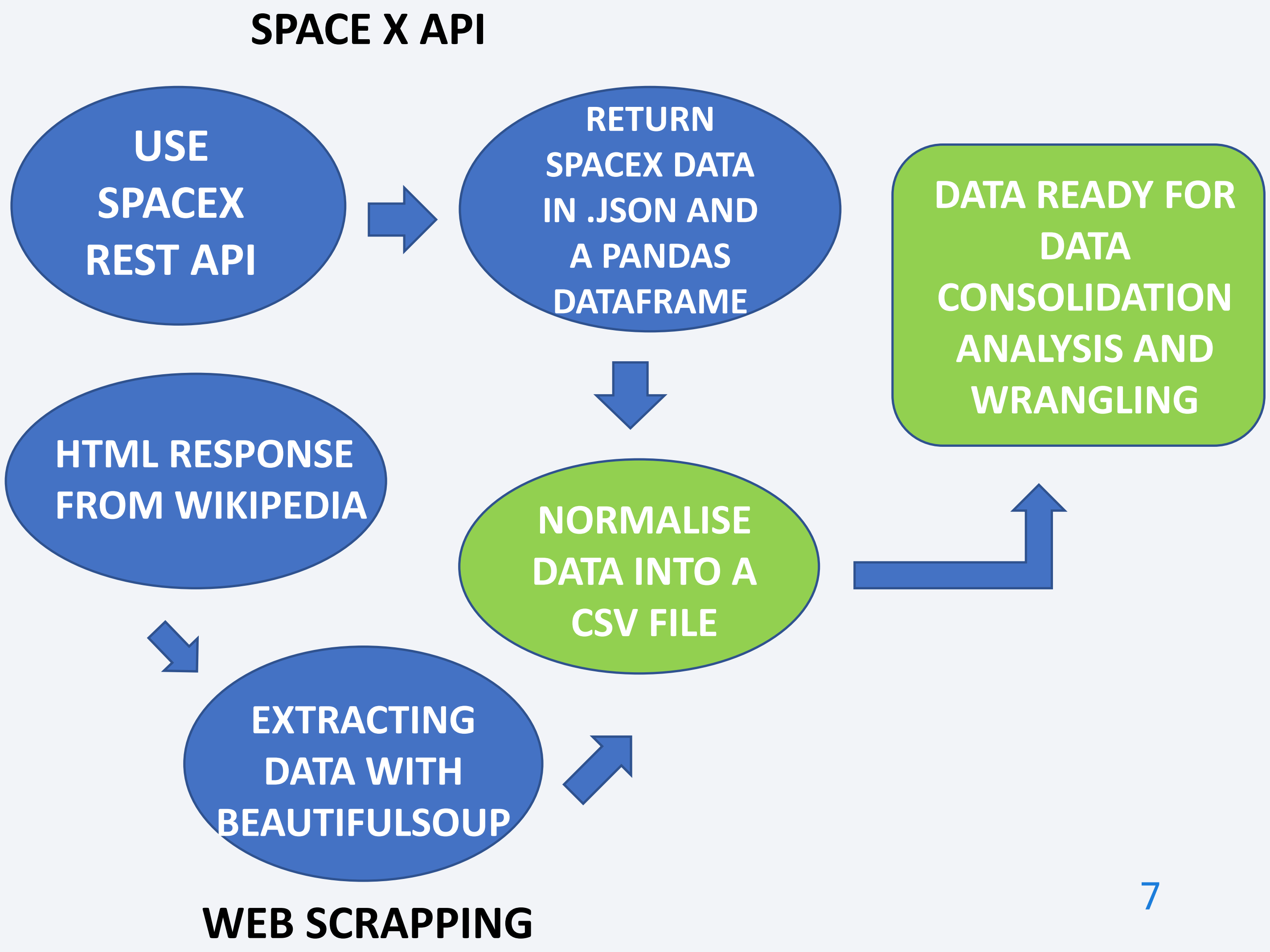
Data Collection

- The datasets was collected that way:

- Collection of SpaceX launch data through the REST API provided by the company. The API provides us with information about launches, including details about the rocket used, payload carried, launch specifications, landing specifications and landing results.

- SpaceX REST API endpoints have a URL beginning with "api.spacexdata.com/v4/".

- Another popular data source for information about the Falcon 9 launch is through data scraping from Wikipedia using the BeautifulSoup library.



Data Collection - SpaceX API

- Data Collection with SpaceX REST calls
- GITHUB:

https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/spacex-data-collection-api_TASK1_MARINALIMA.ipynb

1. Getting Data & response from the API

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

2. Converting response to a .json file and turn into a Pandas Dataframe

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
In [12]: # Get the head of the dataframe
         data.head()
```

Out[12]:

static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules

3. Apply custom functions to clean data

Data Collection - SpaceX API

- Data Collection with SpaceX REST calls

GITHUB:

https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/spacex-data-collection-api_TASK1_MARINALIMA.ipynb

3. Apply custom functions to clean data

```
In [16]: # Call getBoosterVersion
         getBoosterVersion(data)
```

the list has now been update

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4. Assign list to dictonary and dataframe

```
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}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

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

5. Filter dataframe and export to .csv

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

Data Collection - Scraping

- **WEBSCRAPING FROM WIKIPEDIA**

1. GETTING RESPONSE FROM HTML

```
# use requests.get() method with the provided static_url
response = requests.get(static_url).text
```

2. CREATING BEAUTIFULSOUP OBJECT

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

3. FINDING TABLES

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

4. GETTING COLUMN NAMES

```
column_names = []

# Apply find_all() function with `th` element on
# Iterate each th element and apply the provided
# Append the Non-empty column name (`if name is r
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. CREATION OF 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
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. APPENDING TO KEYS

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(sou
   # get table row
    for rows in table.find_all("tr"):
```

7. CONVERTING DICTIONARY TO DATAFRAME

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

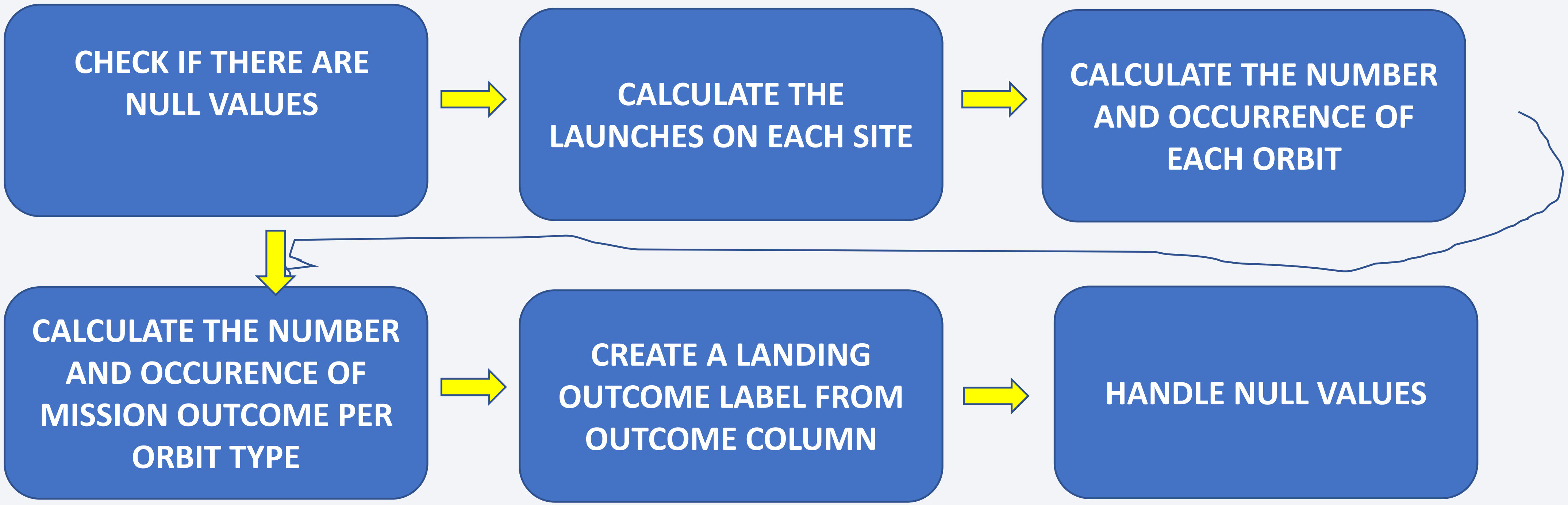
8. DATAFRAME TO CSV

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

GITHUB: https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/webscraping__MARINALIMA.ipynb

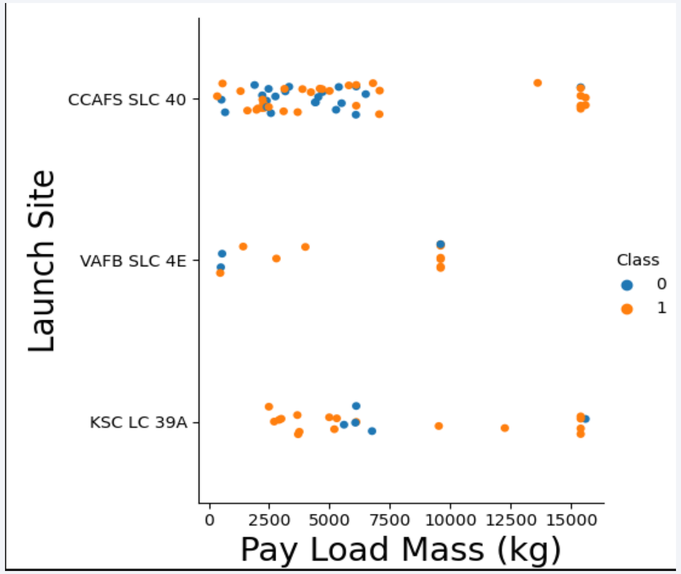
Data Wrangling

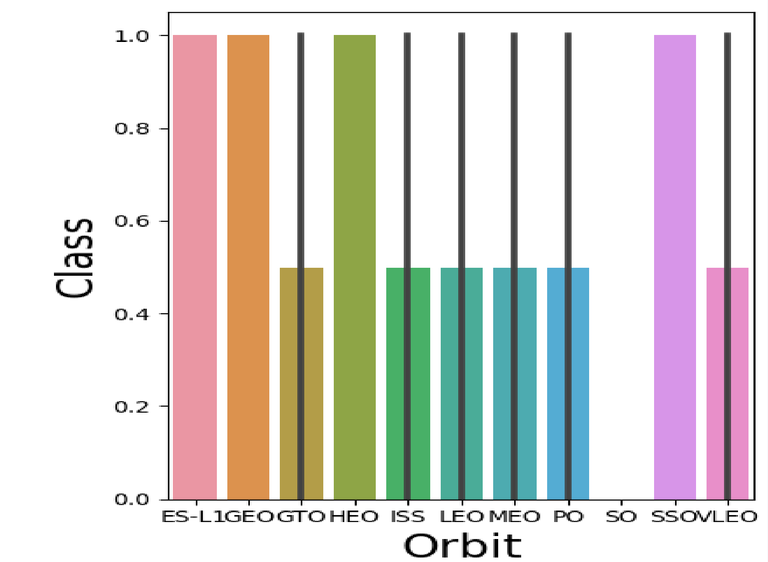
EDA ANALYSIS

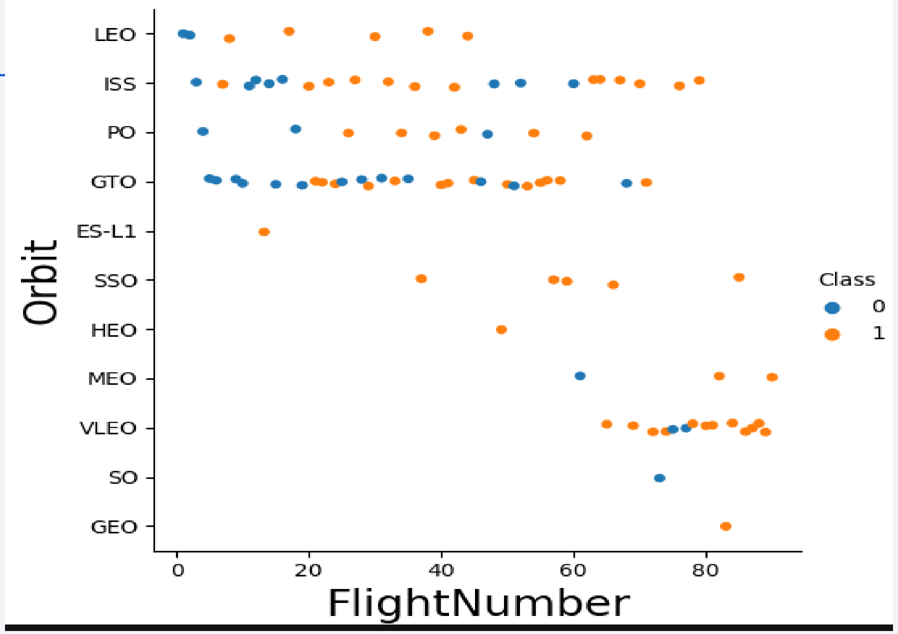


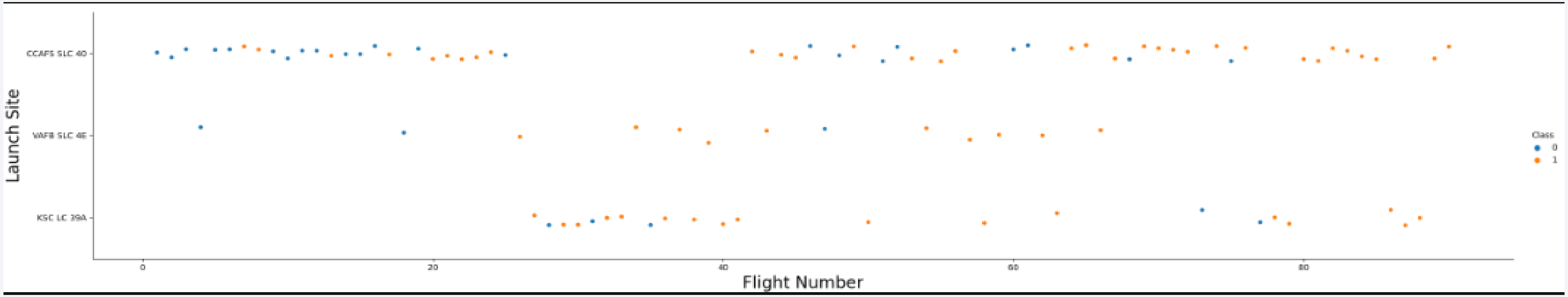
GITHUB: https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/EDA_data_wrangling_MARINALIMA.ipynb

EDA with Data Visualization



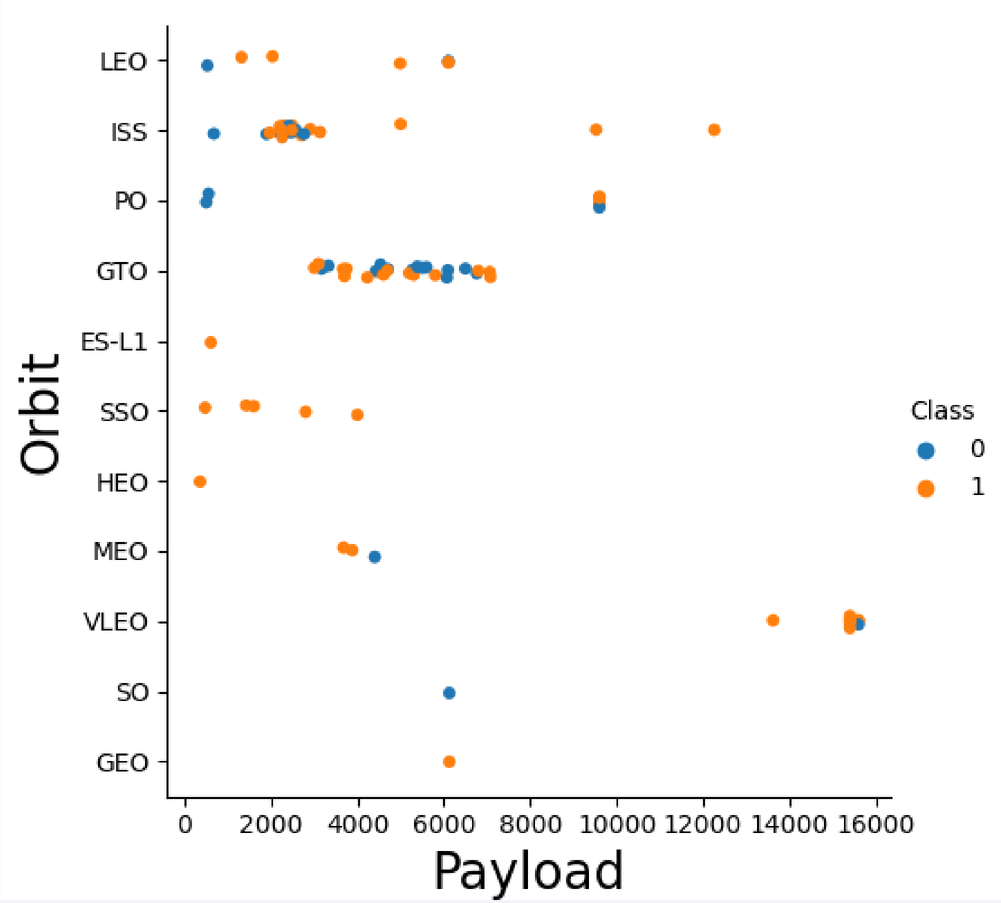


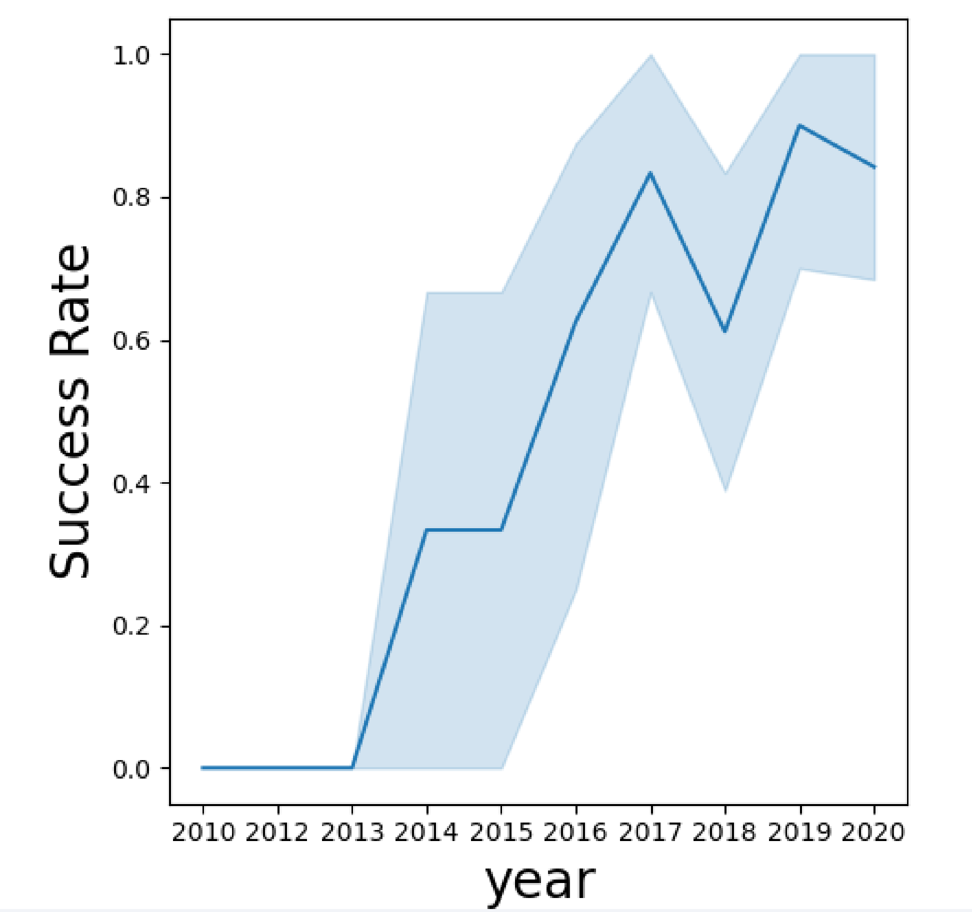




https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/eda-dataviz_MARINALIMA.ipynb

EDA with Data Visualization





https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/eda-dataviz_MARINALIMA.ipynb

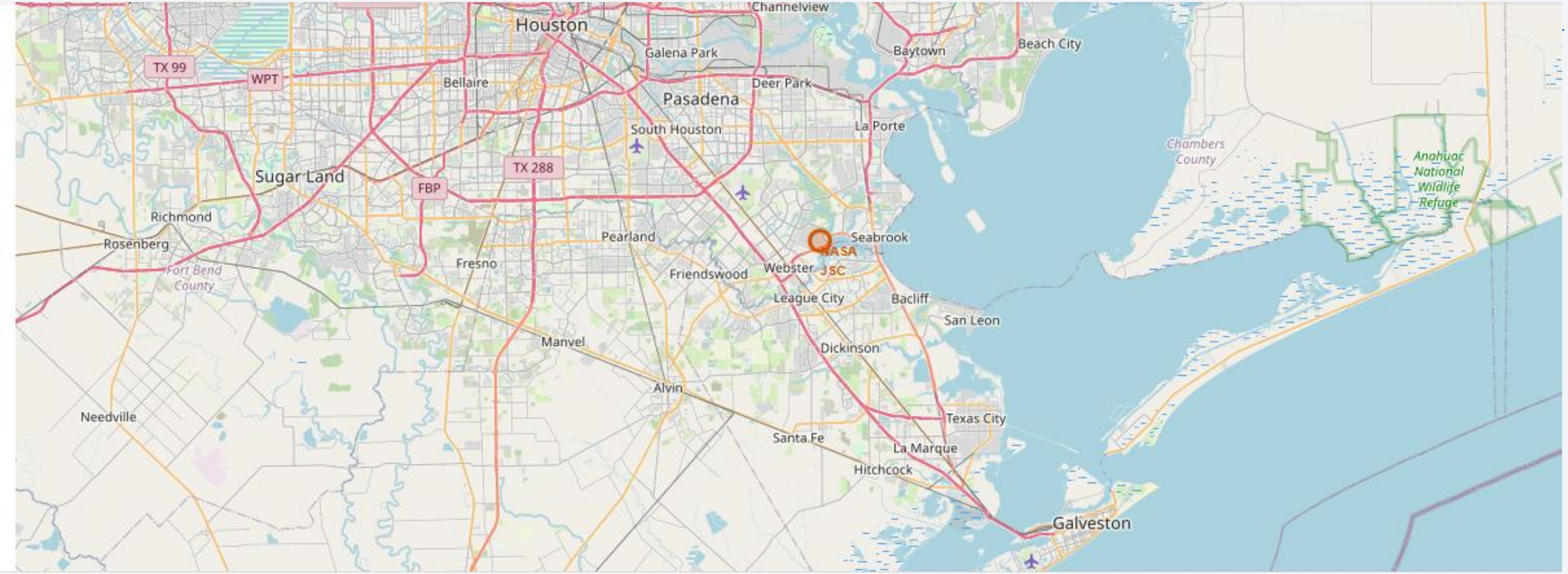
EDA with SQL

- **SQL queries performed:**

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster
- versions, launch_site for the months in year 2017
- Ranking the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 in descending

https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/EDA%20WITH%20SQL_marinalima.ipynb

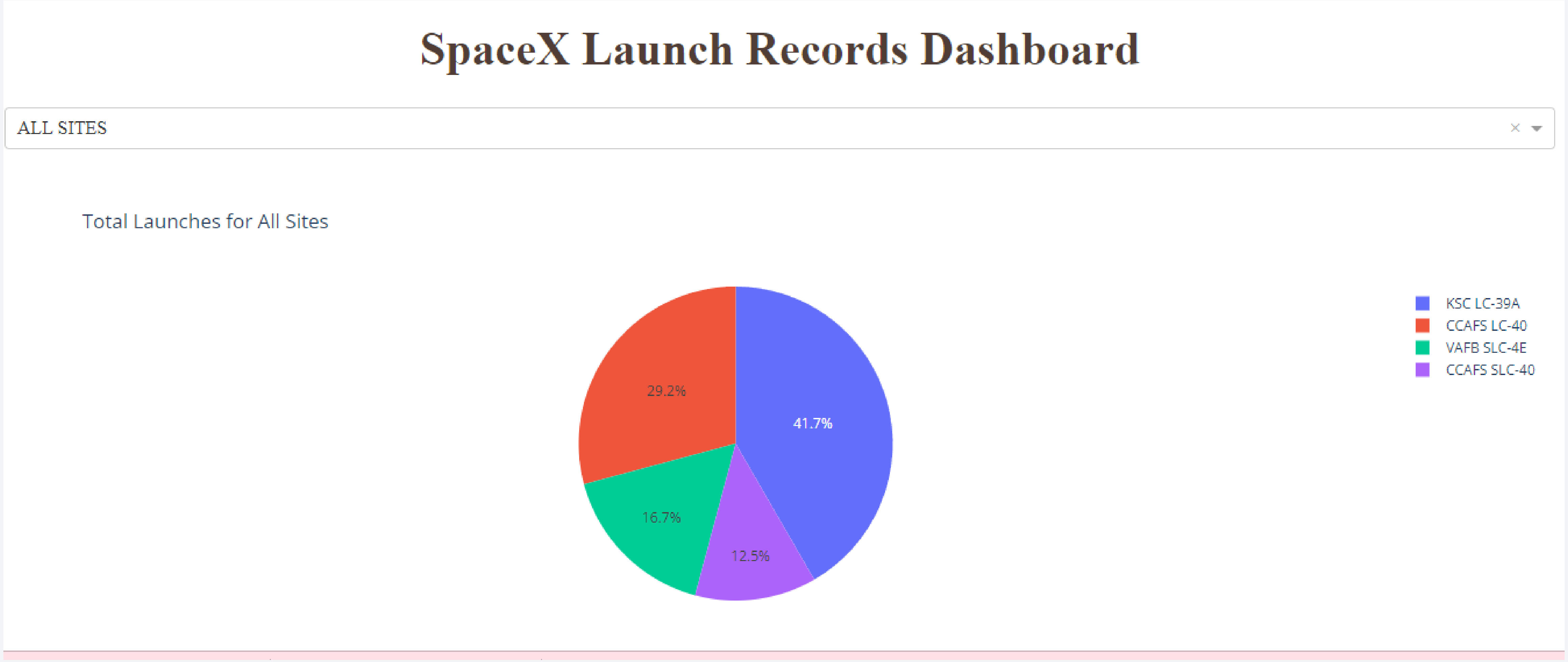
Build an Interactive Map with Folium



Map markers have been added to the map with aim to finding an optimal location for building a launch site

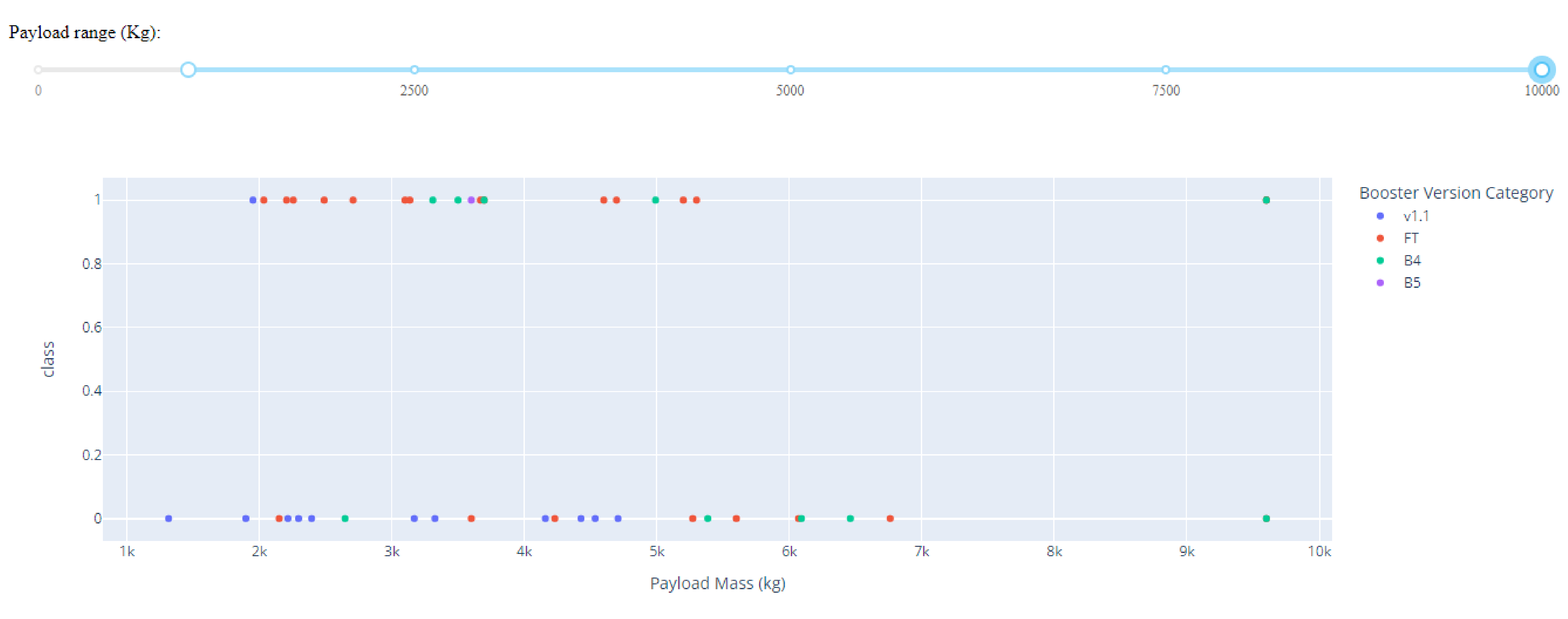
https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/Building_Interactive_Maps_with_Folium_IBM_Final_PROJECT_Developed_by_Marina_Lima%20(2).ipynb

Build a Dashboard with Plotly Dash



https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/Dashboard_Plotly_Dash_IBM_MARINALIMA.ipynb

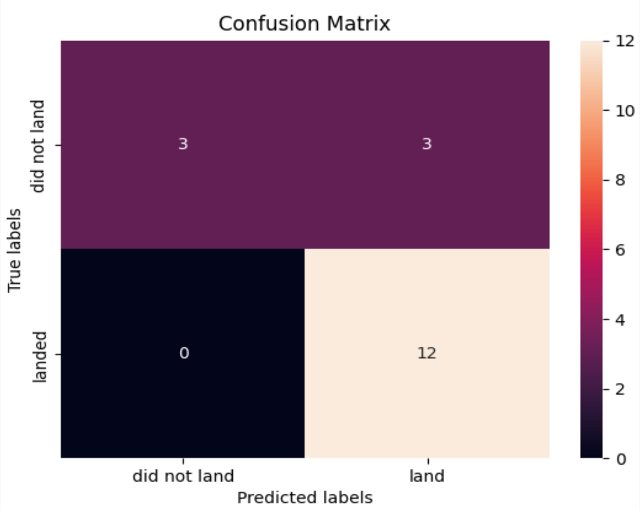
Build a Dashboard with Plotly Dash



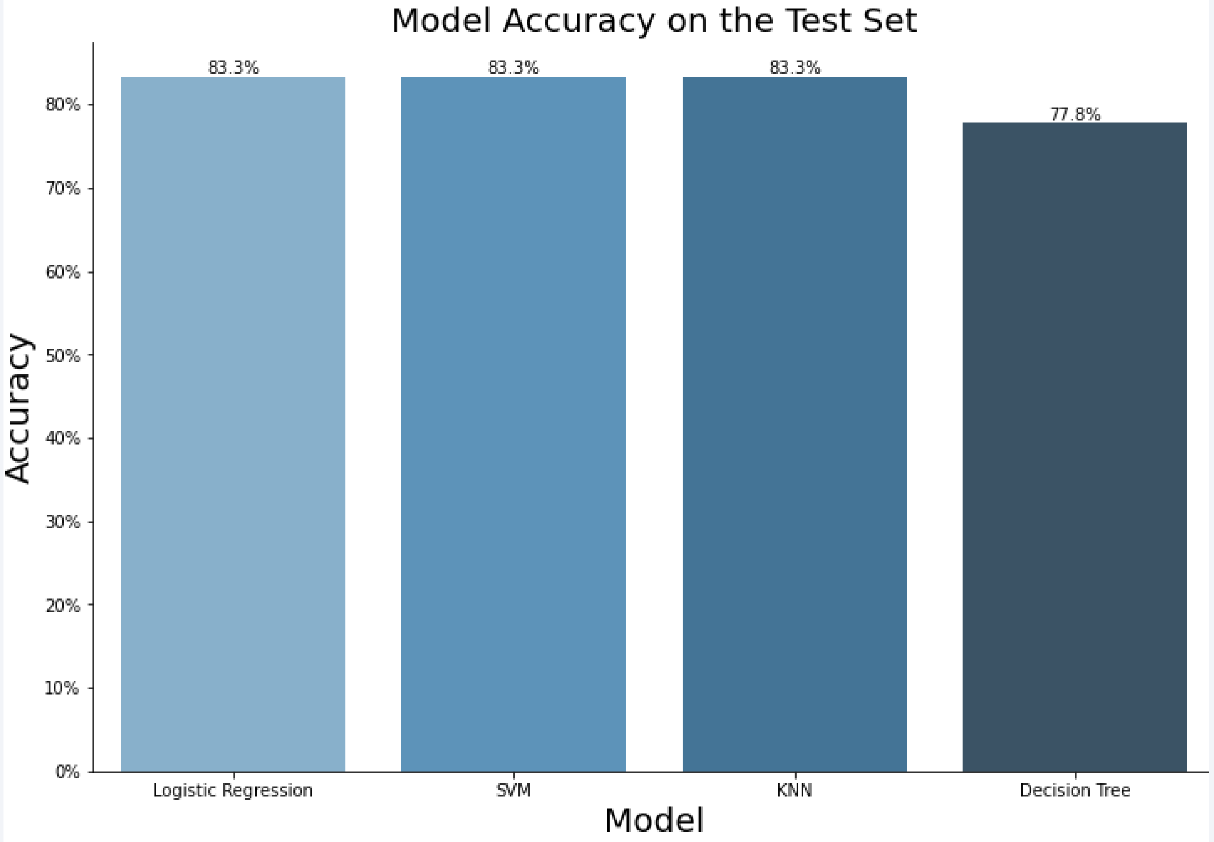
https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/Dashboard_Plotly_Dash_IBM_MARINALIMA.ipynb

Predictive Analysis (Classification)

- The SVM, KNN, and Logistic Regression model achieved the highest accuracy at 83.3%, while the SVM performs the best in terms of Area Under the Curve at 0.958



	F1-Score	Precision	Recall	Accuracy
Decision Tree	0.917	0.917	0.917	0.889
Logistic Regression	0.889	0.800	1.000	0.833
SVM	0.889	0.800	1.000	0.833
KNN	0.889	0.800	1.000	0.833



https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/SpaceX_Machine_Learning_Prediction_MARINALIMA.ipynb

Results

Based on the analysis we can conclude that:

1.The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.

2. Low weighted payloads perform better than the heavier payloads.

3. KSC LC 39A had the most successful launches from all the sites.

4. Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

5. The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches

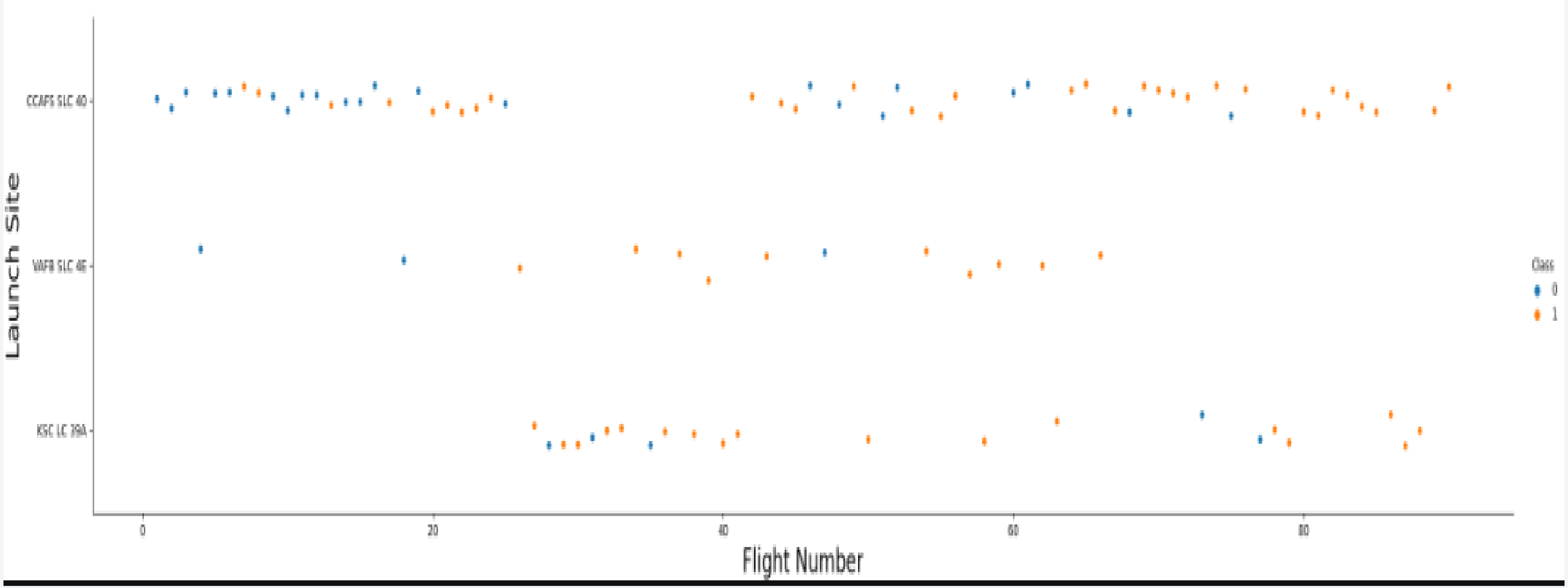
	F1-Score	Precision	Recall	Accuracy
Decision Tree	0.917	0.917	0.917	0.889
Logistic Regression	0.889	0.800	1.000	0.833
SVM	0.889	0.800	1.000	0.833
KNN	0.889	0.800	1.000	0.833

https://github.com/marinallima/IBM_Data_Science_Professional_Certificate/blob/main/SpaceX_Machine_Learning_Prediction_MARINALIMA.ipynb

Section 2

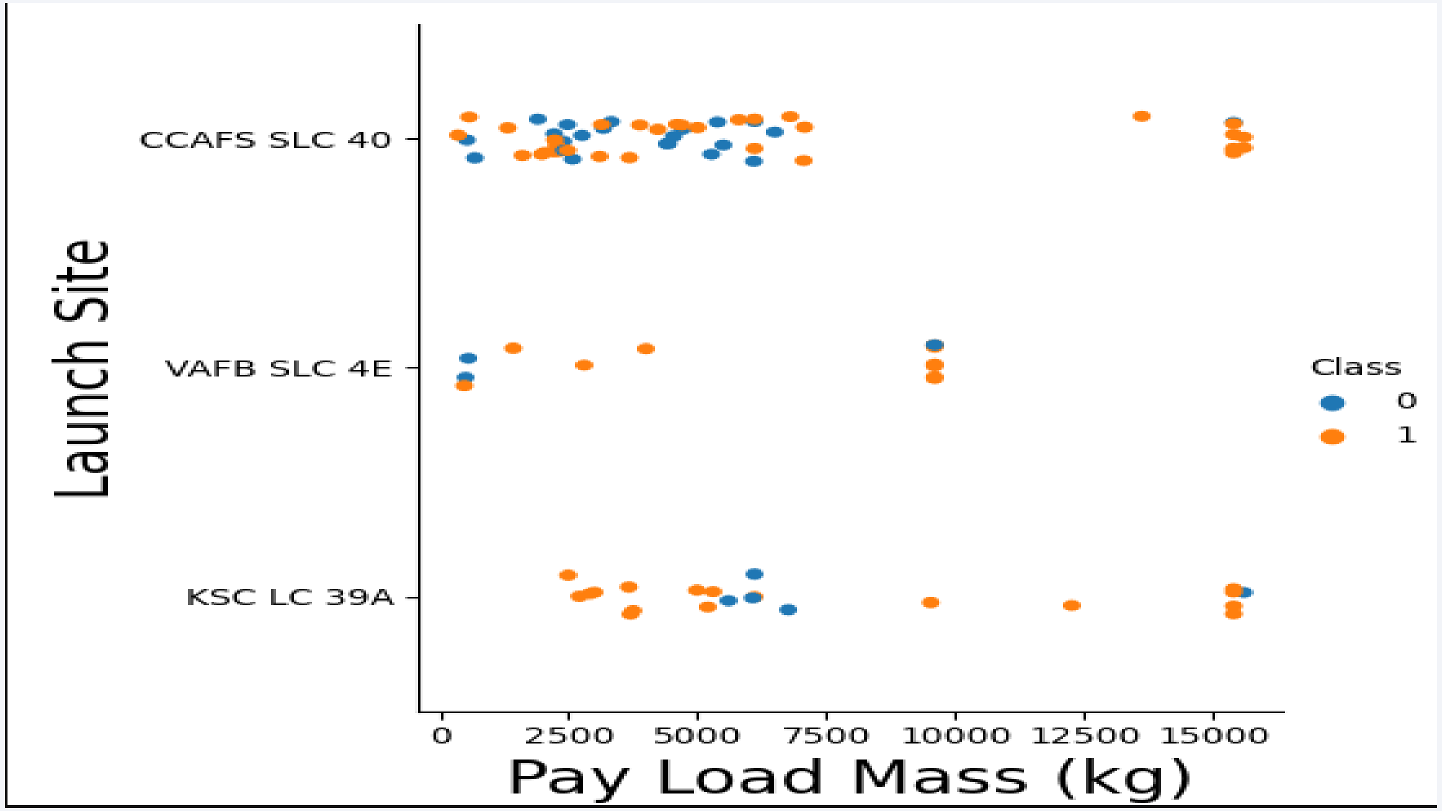
Insights drawn from EDA

Flight Number vs. Launch Site



- We can conclude that launches from the site of CCAFS SLC 40 are higher than launches form other sites.

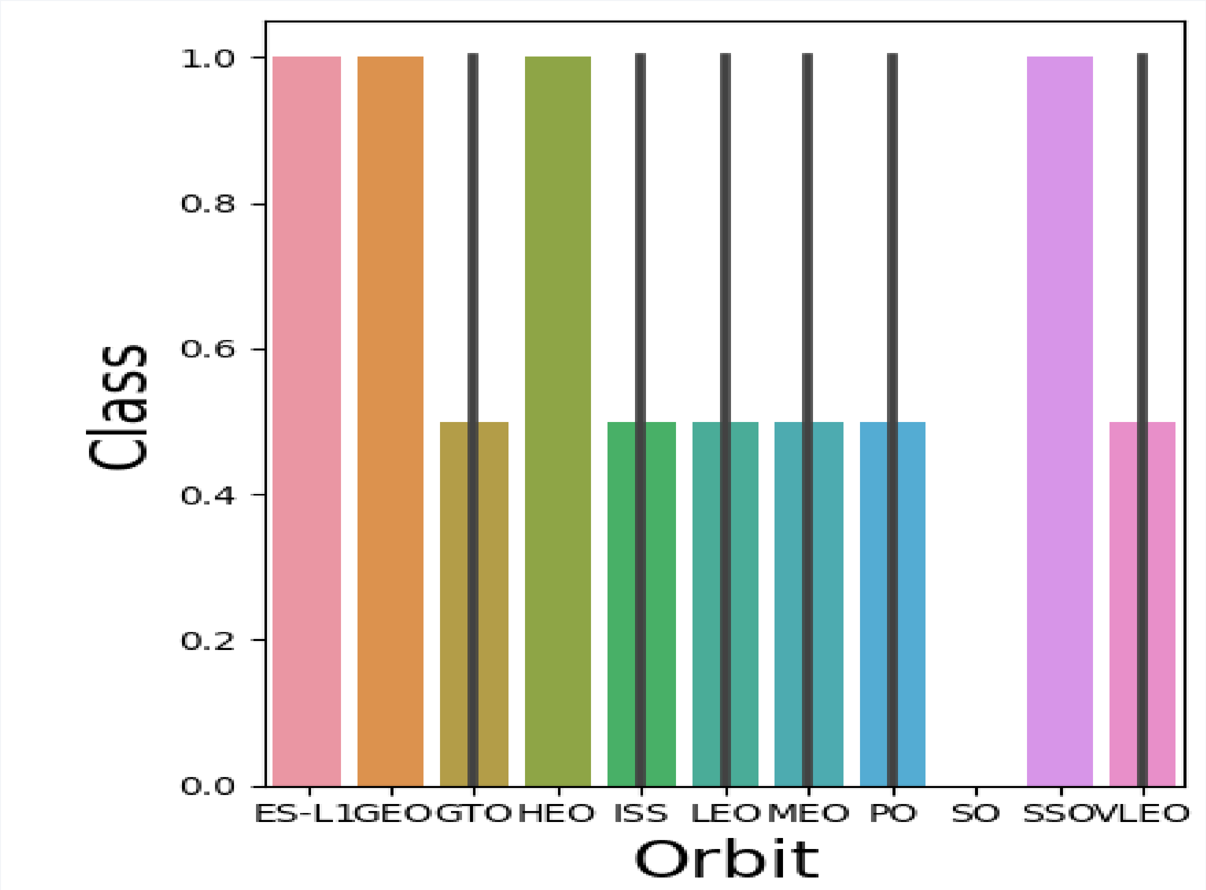
Payload vs. Launch Site



- We can conclude that the majority of lPay Loads with lower Mass have been launched from CCAFS SLC 40.

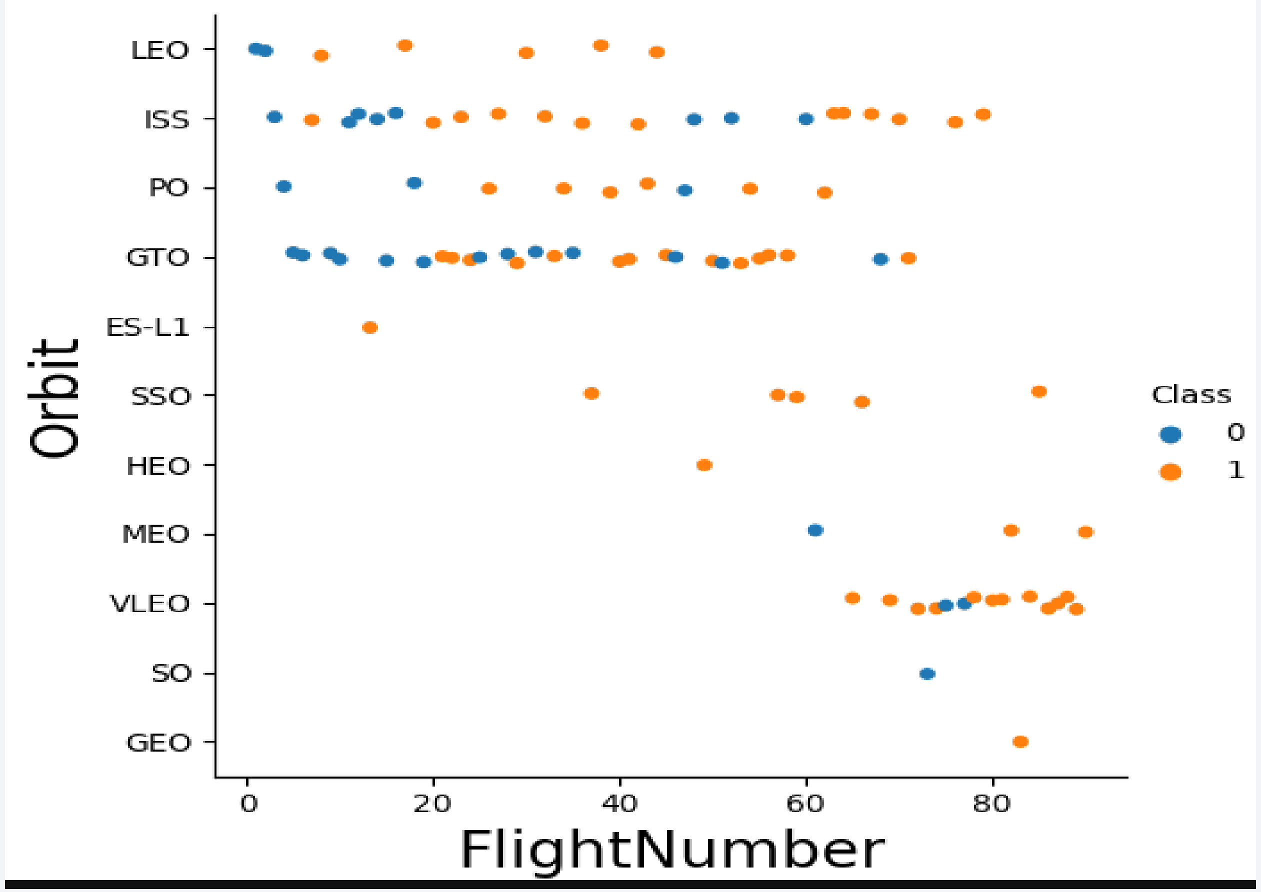
Success Rate vs. Orbit Type

- The orbit types of ES-L1, GEO, HEO, SSO are among the highest success rate.

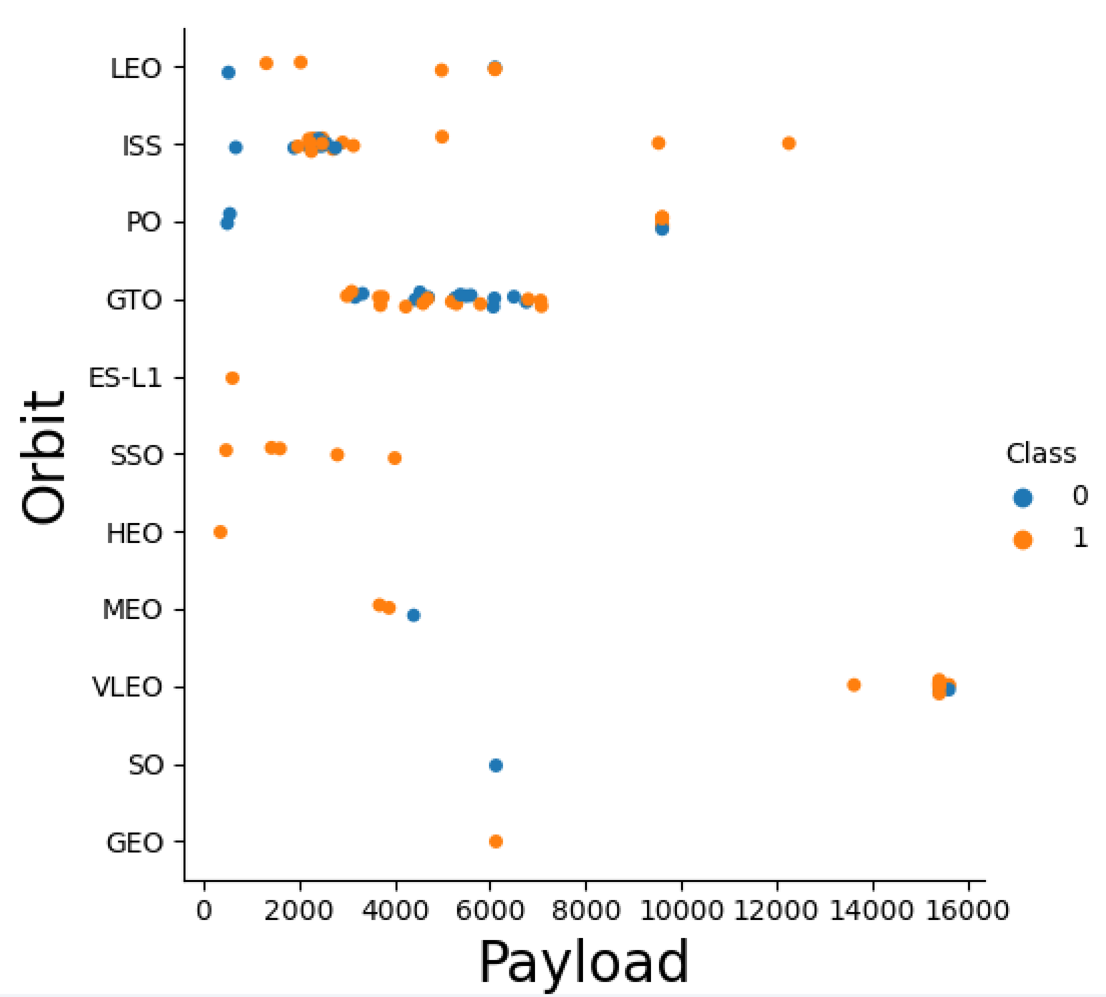


Flight Number vs. Orbit Type

- A trend can be observed of shifting to VLEO launches in recent years.



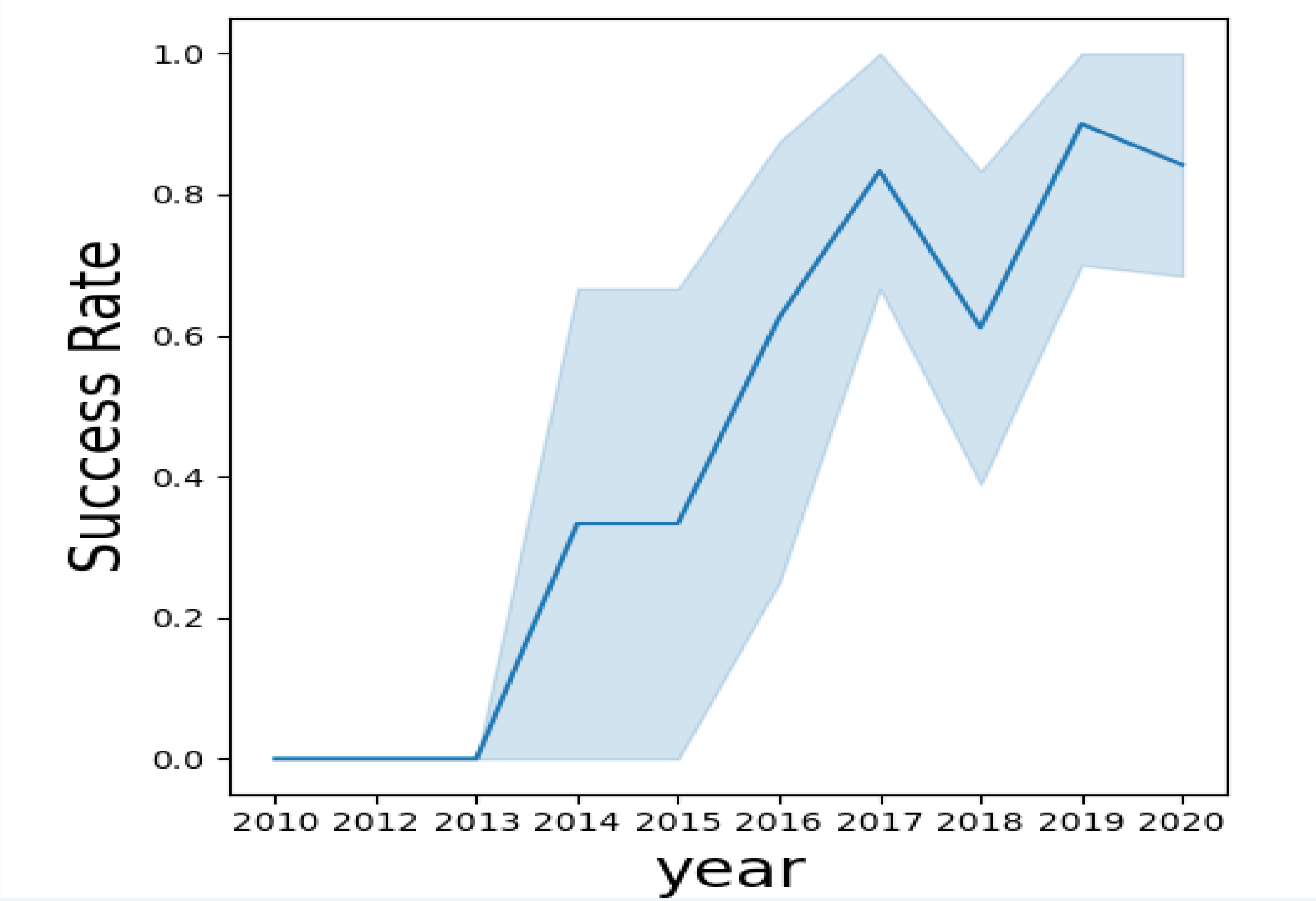
Payload vs. Orbit Type



There are strong correlation between ISS and Payload at the range around 2000, as well as between GTO and the range of 4000-8000.

Launch Success Yearly Trend

- Launch success rate has increased significantly since 2013 and has stablised since 2019, potentially due to advance in technology and lessons learned.



All Launch Site Names

%sql select distinct(LAUNCH_SITE) from SPACEXTBL

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

Launch Site Names Begin with 'CCA'

%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5

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

%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)‘

VALUE = 45596

Average Payload Mass by F9 v1.1

%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1

VALUE = 2928.400000

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)‘
```

RESPONSE: 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where
MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

100

Boosters Carried Maximum Payload

%sql select 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

2015 Launch Records

%sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between ‘2015-01-01' and ‘2015-12-31') order by date desc

time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)

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

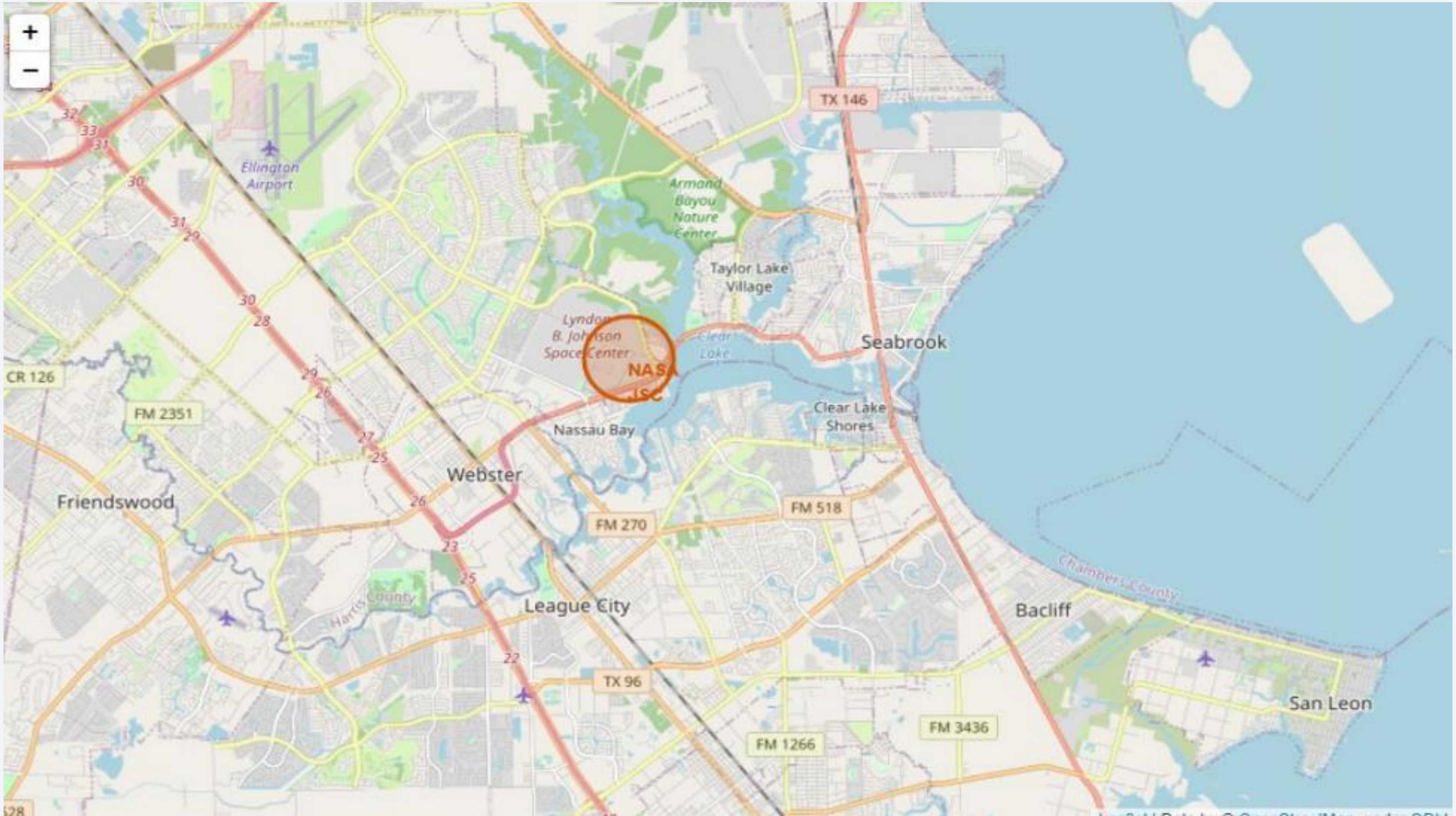
```
%sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE
between '2010-06-04' and '2017-03-20') order by date desc
```

2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

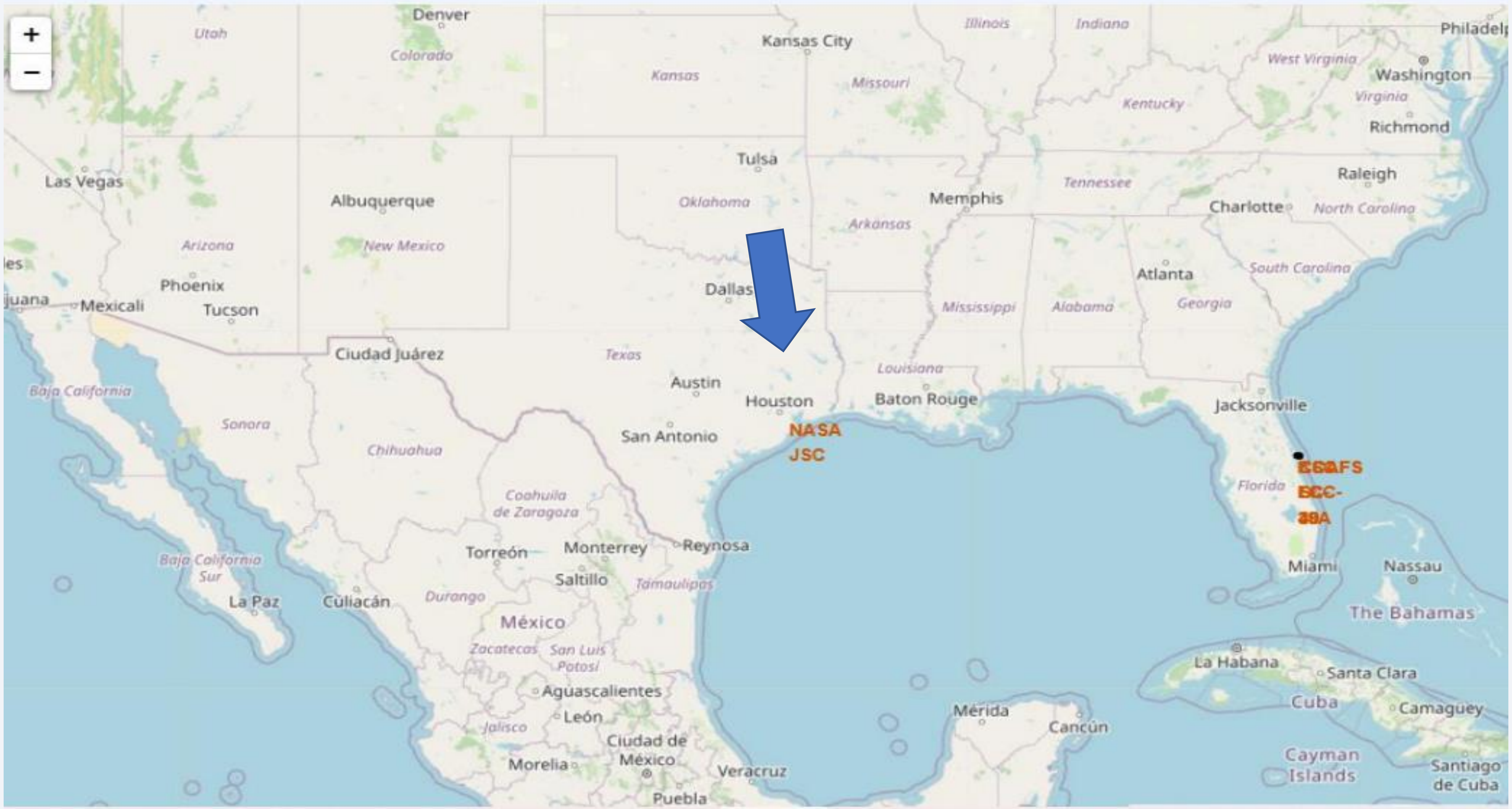
Section 3

Launch Sites Proximities Analysis

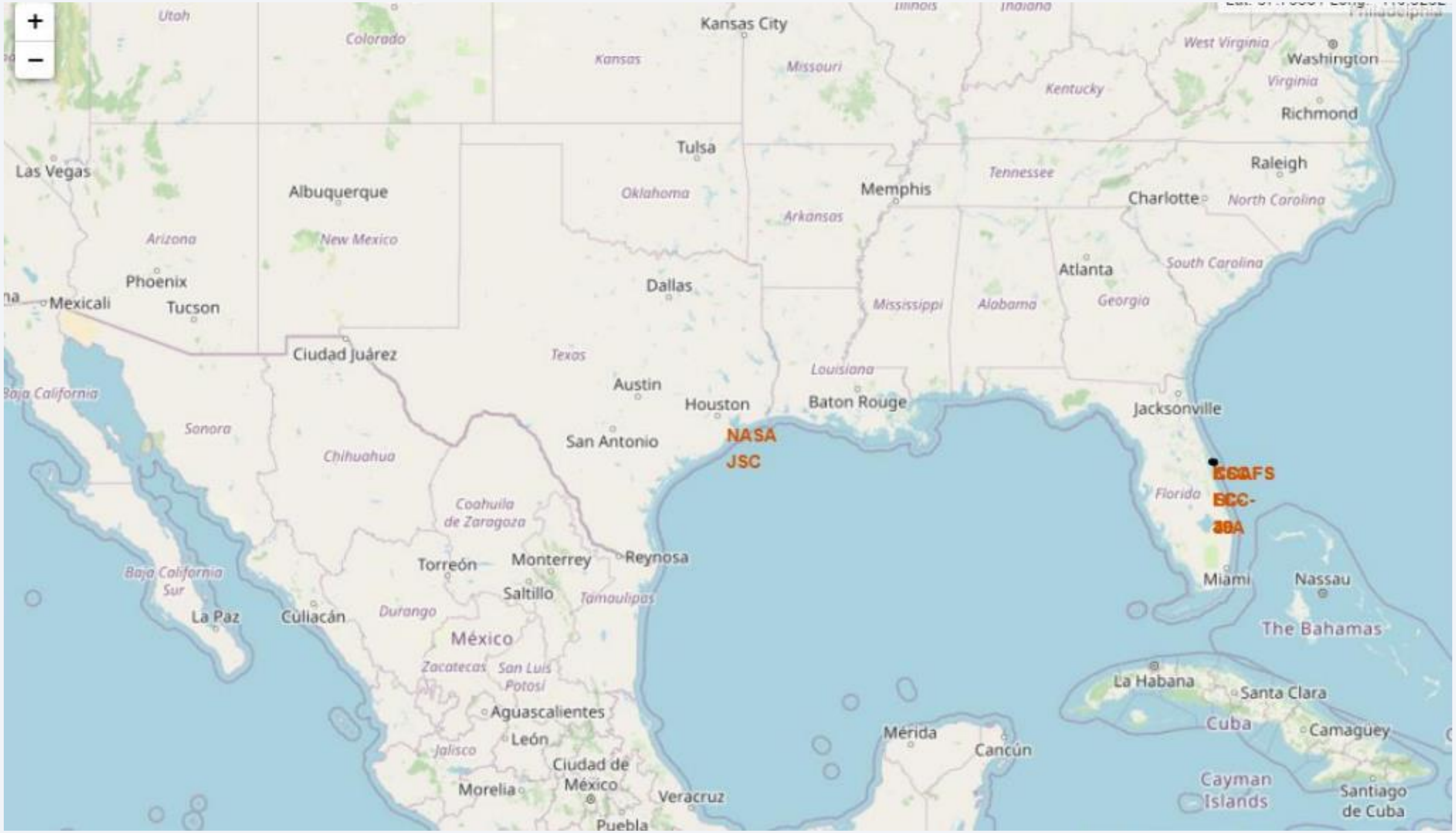
All launch sites



FAILED AND SUCCESS LAUNCHES



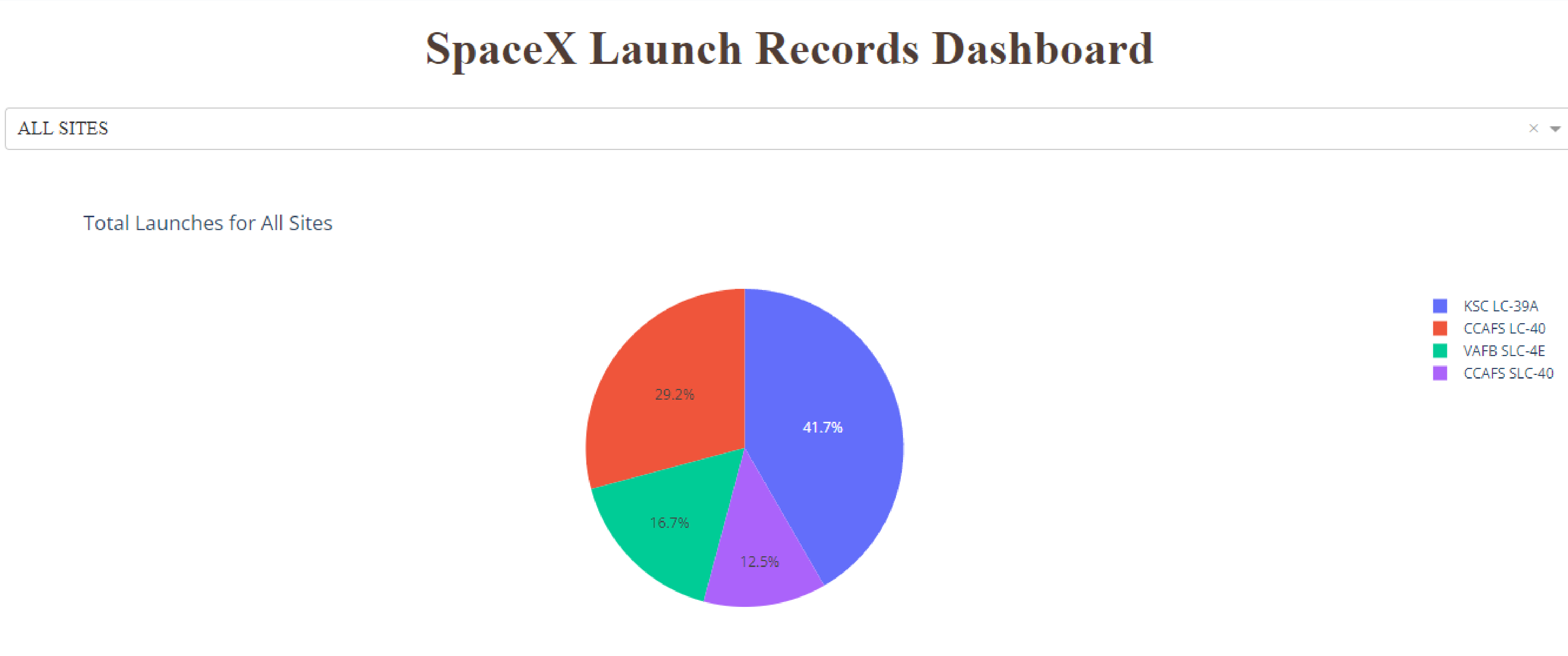
Selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



Section 4

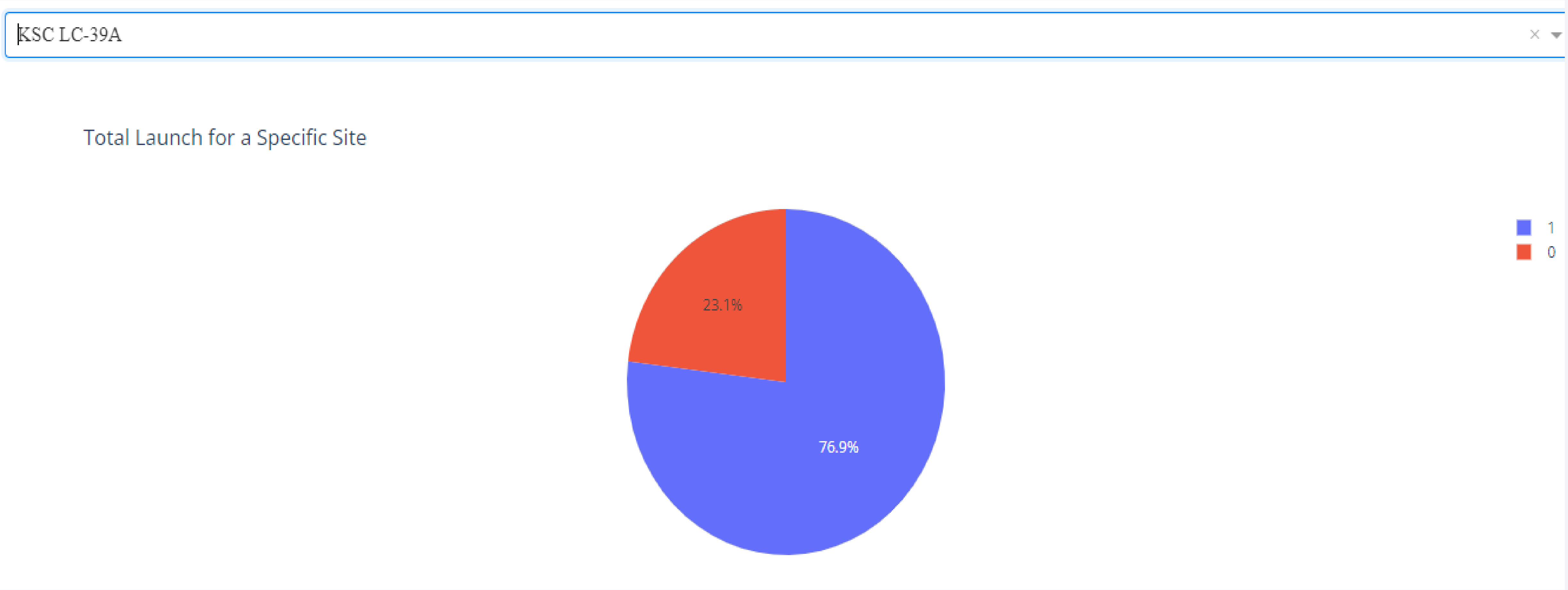
Build a Dashboard with Plotly Dash

Total success launches by all sites



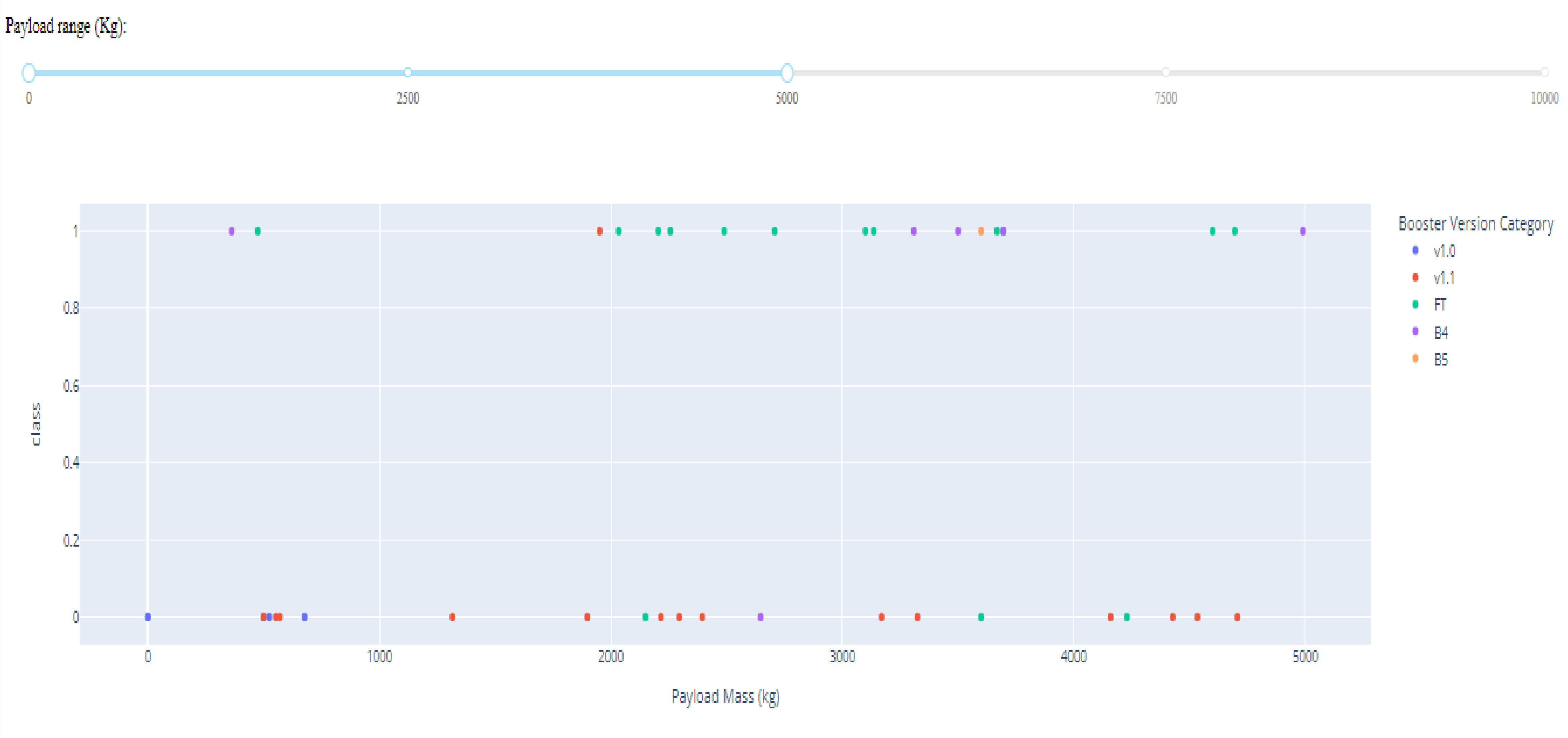
- Conclusion: KSC LC-39A had the most successful launches from all the sites

Success rate by site



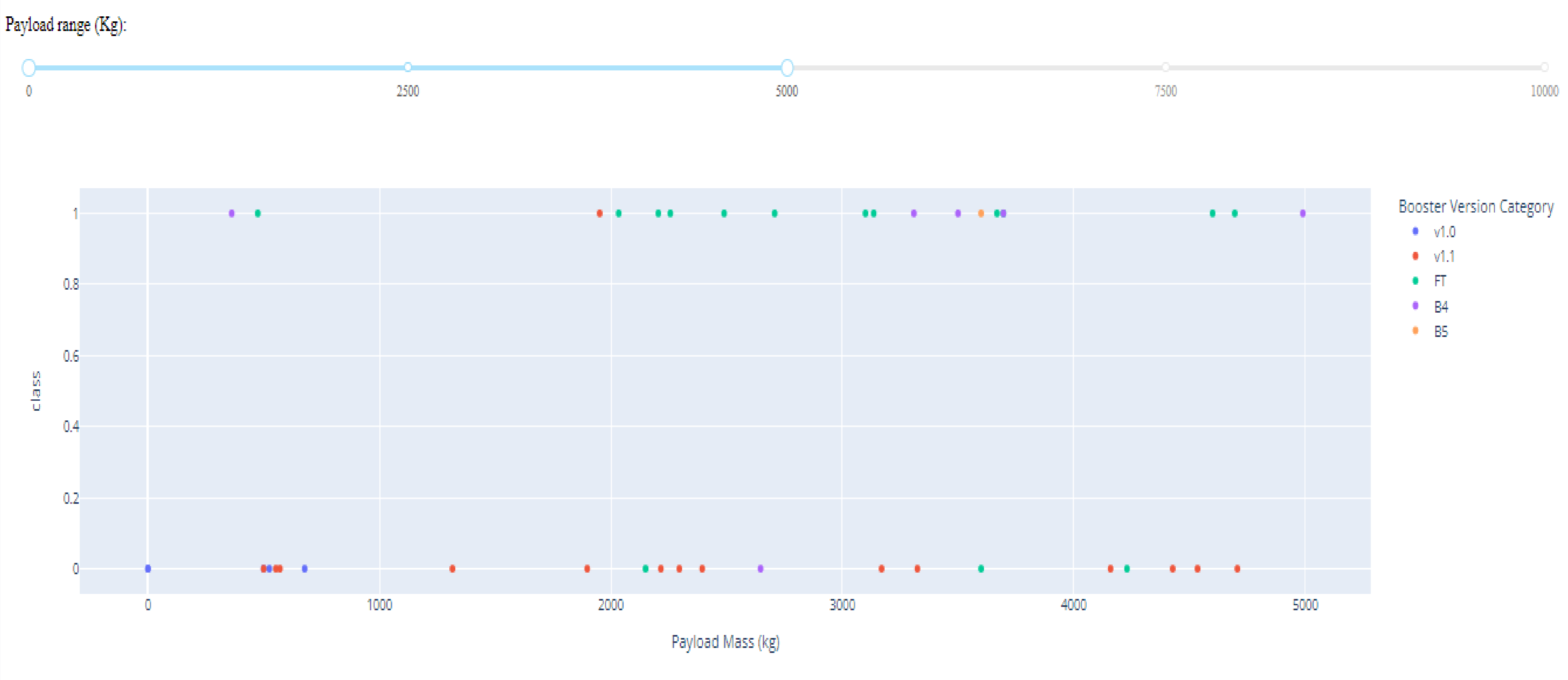
We can conclude that KSC LC-39 achieved 76.9% success rate while getting a 23.1% failure rate

Payload vs. Launch Outcome scatter plot for all sites



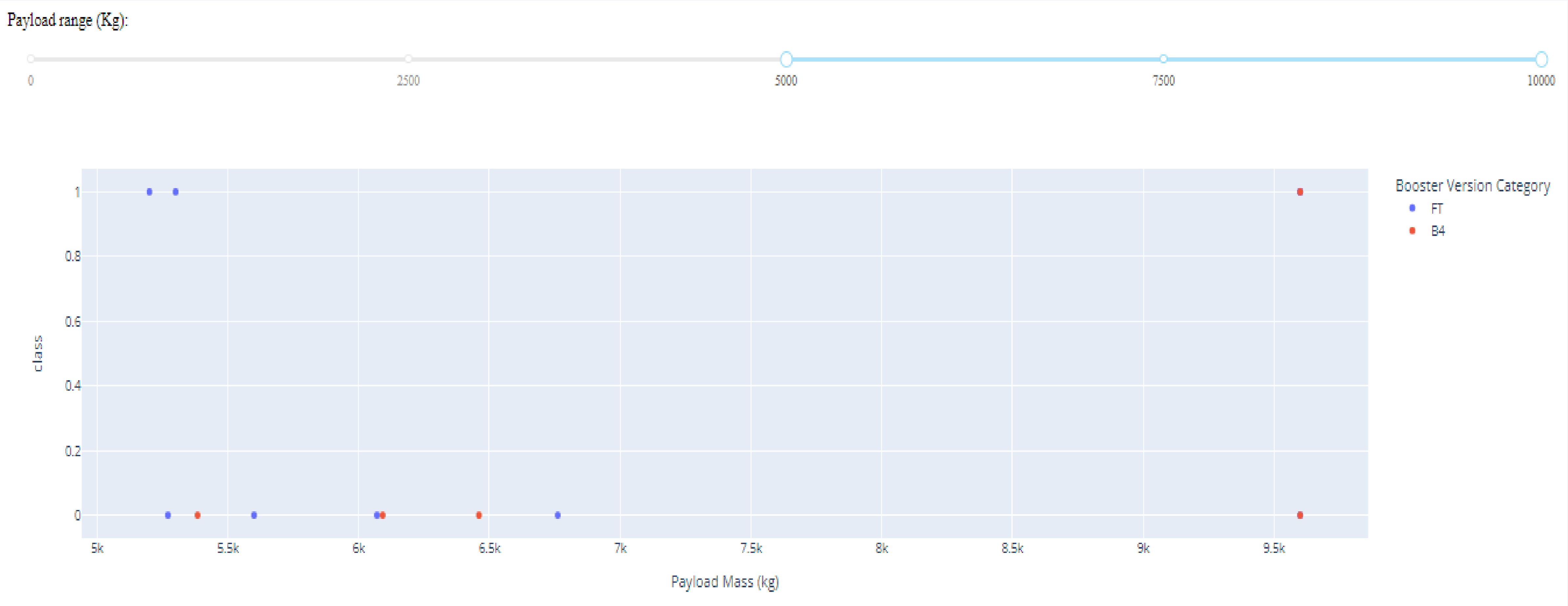
Payload vs. Launch Outcome scatter plot for all sites

LOW WEIGHT PAYLOAD 0KG – 5000KG



Payload vs. Launch Outcome scatter plot for all sites

HEAVY WEIGHT PAYLOAD 5000KG – 10000KG

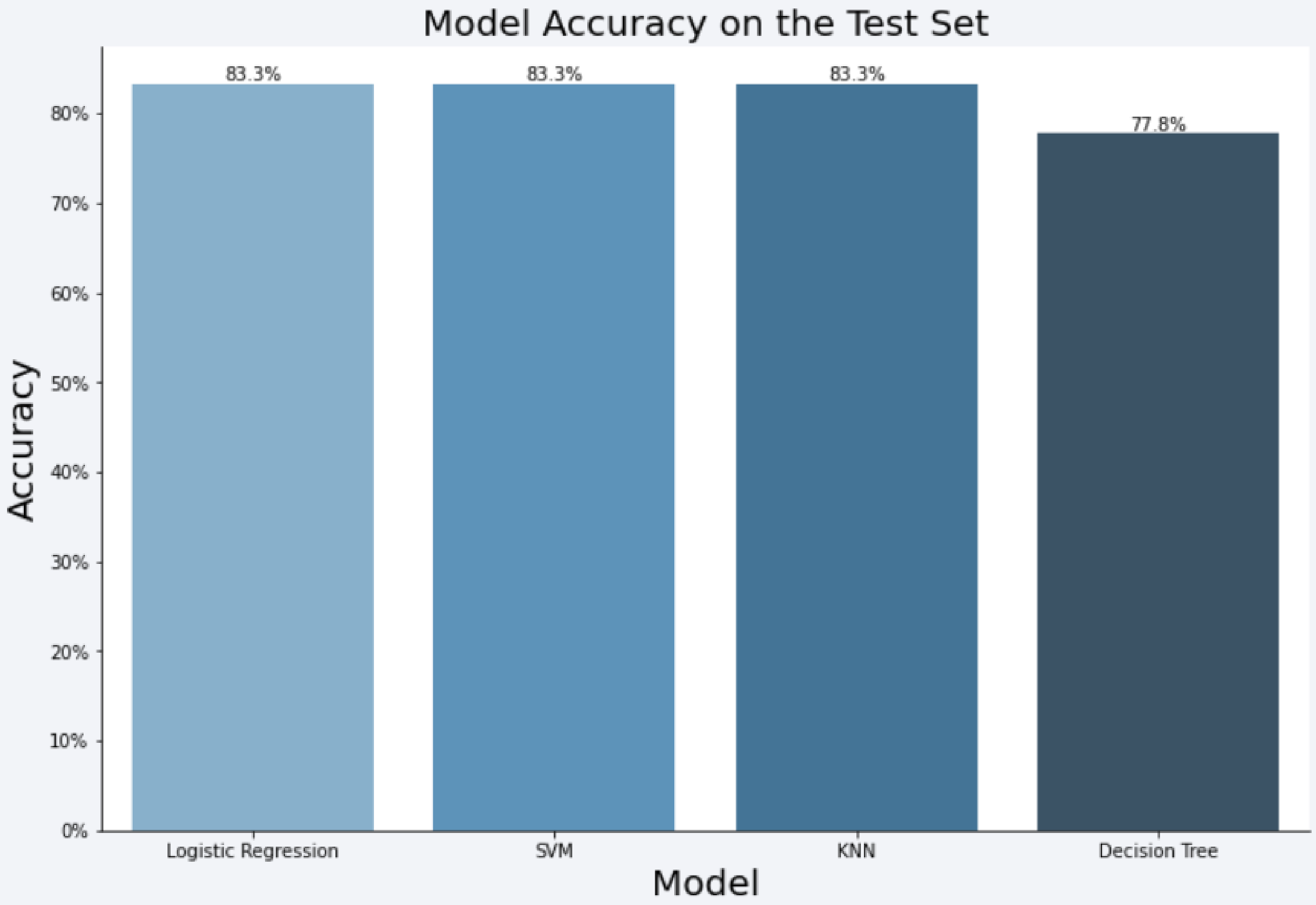


WE CAN CONCLUDE THAT SUCCESS RATES FOR LOW WEIGHTED PAYLOADS IS HIGHER THAN THE HEAVY WEIGHTS PAYLOADS

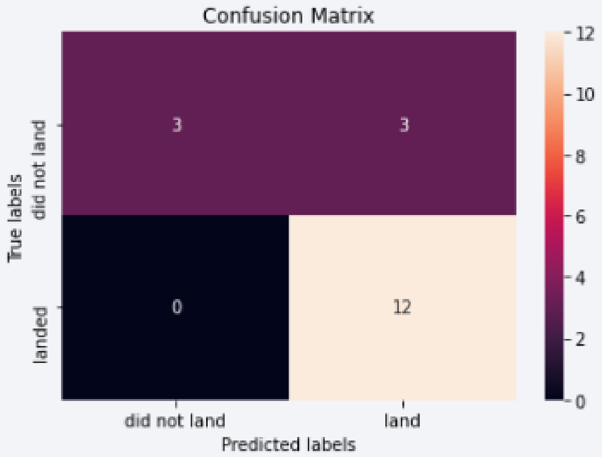
Section 5

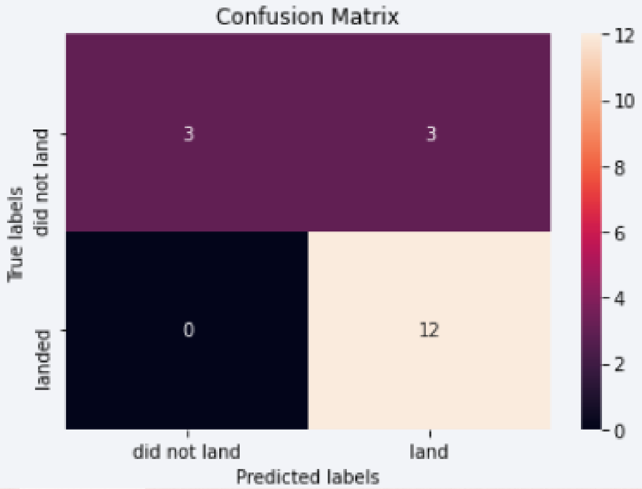
Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix





Conclusions

We can conclude that:

• The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.

• Low weighted payloads perform better than the heavier payloads.

• KSC LC 39A had the most successful launches from all the sites.

• Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

• The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.

Thank you!