IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raiya Said AlRashdi
20-11-2023

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- **Project background and context**

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.

- **Problems you want to find answers**

a) How the lunch cost In SpaceX is lower than other companies

b) How can any alternate company bid against SpaceX for a rocket launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

The data is collected from a SpaceX API

- Perform data wrangling

The data collected from HTML tables and converted to into a dataframe for vitualization using **json_normalize** method

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Using classification algorithms to predict if the first stage of Falcon 9 land successfully

Data Collection

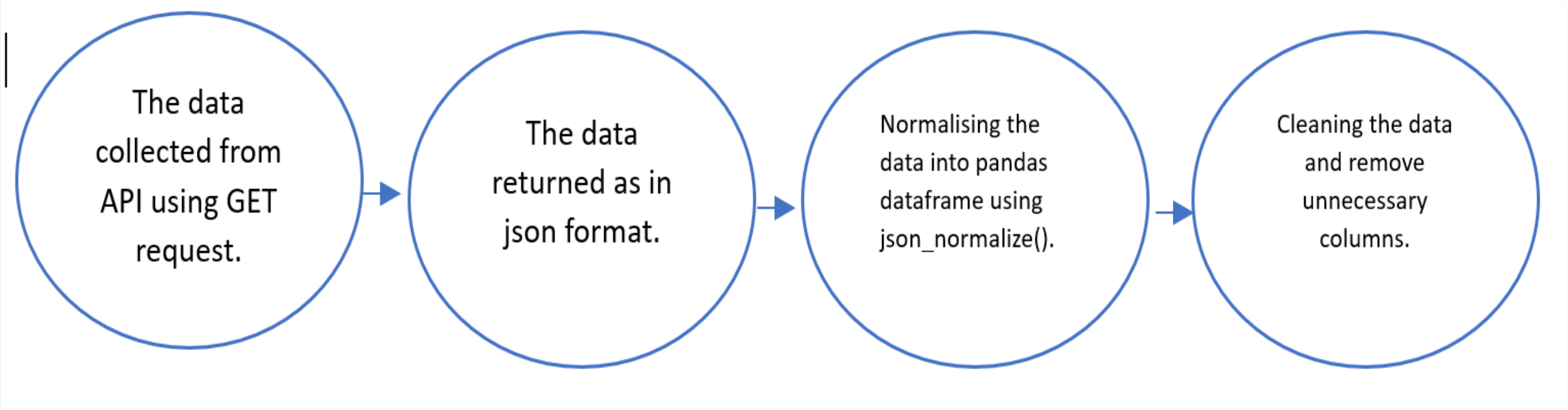
- Describe how data sets were collected.

1. The data collected from API using GET request.
2. The data returned as in json format
3. Normalizing the data into pandas dataframe using json_normalize()

- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

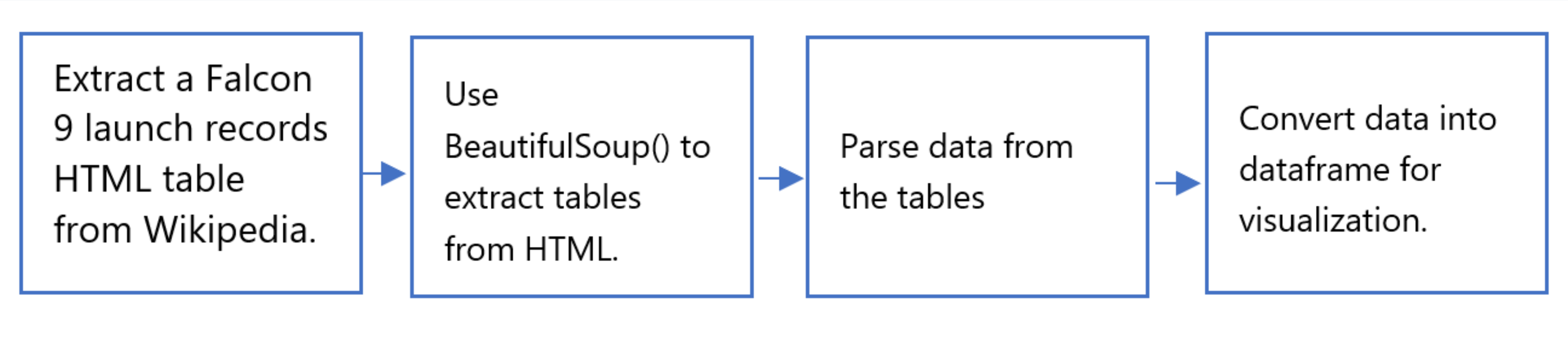
- Present your data collection with SpaceX REST calls using key phrases and flowcharts



- Add the GitHub URL of the completed SpaceX API calls notebook (https://github.com/raiyasaid91/raiyarepo/blob/master/Raiya_dataCollectionAPI.ipynb), as an external reference and peer-review purpose

Data Collection - Scraping

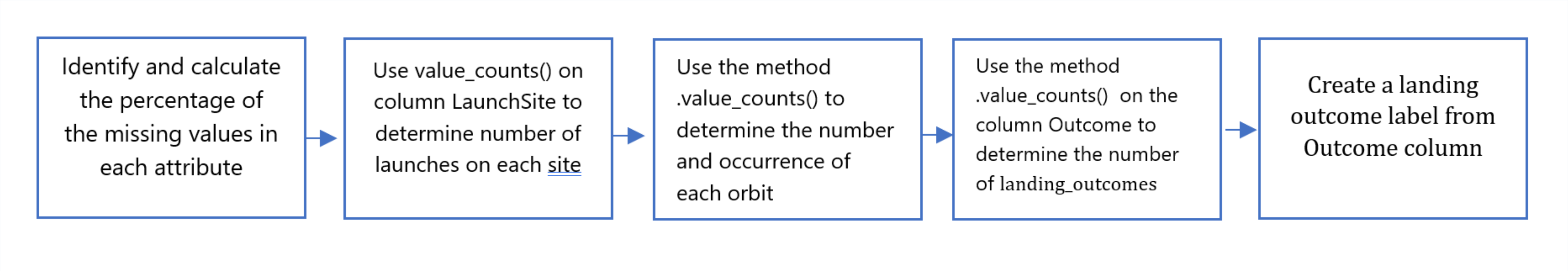
- Present your web scraping process using key phrases and flowcharts



- Add the GitHub URL of the completed web scraping notebook, (https://github.com/raiyasaid91/raiyarepo/blob/master/jupyter-labs-webscraping%20(6).ipynb) as an external reference and peer-review purpose

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts



- https://github.com/raiyasaid91/raiyarepo/blob/master/Raiya_labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

➢ Catplot: to visualize the relationship between the FightNumber & Payload

➢ Catplot: to visualize the relationship between the FightNumber & LaunchSite

➢ Catplot: to visualize the relationship between the Payload & LaunchSite

➢ Catplot: to visualize the relationship between the Payload & LaunchSite

➢ Bar Chart: to visualize the relationship between success rate of each OrbitType

➢ Catplot: to visualize the relationship between the FightNumber & OrbitType

➢ Catplot: to visualize the relationship between the Payload & OrbitType

➢ Line Chart: to vitualize the launch success yearly trend.

- https://github.com/raiyasaid91/raiyarepo/blob/master/Raiya_Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
    - **Display the names of the unique launch sites in the space mission**
        - SELECT DISTINCT LaunchSite FROM SpaceX
    - **Display 5 records where launch sites begin with the string 'CCA'**
        - SELECT * FROM SpaceX WHERE LaunchSite LIKE 'CCA%' LIMIT 5
    - **Display the total payload mass carried by boosters launched by NASA (CRS)**
        - SELECT SUM(PayloadMassKG) AS Total_PayloadMass FROM SpaceX WHERE Customer LIKE 'NASA (CRS)'
    - **Display average payload mass carried by booster version F9 v1.1**
        - SELECT AVG(PayloadMassKG) AS Avg_PayloadMass FROM SpaceX WHERE BoosterVersion = 'F9 v1.1'
    - **List the date when the first successful landing outcome in ground pad was achieved**
        - SELECT MIN(Date) AS FirstSuccessfull_landing_date FROM SpaceX WHERE LandingOutcome LIKE 'Success (ground pad)'
- https://github.com/raiyasaid91/raiyarepo/blob/master/Raiya-EDA%20with%20SQL.ipynb

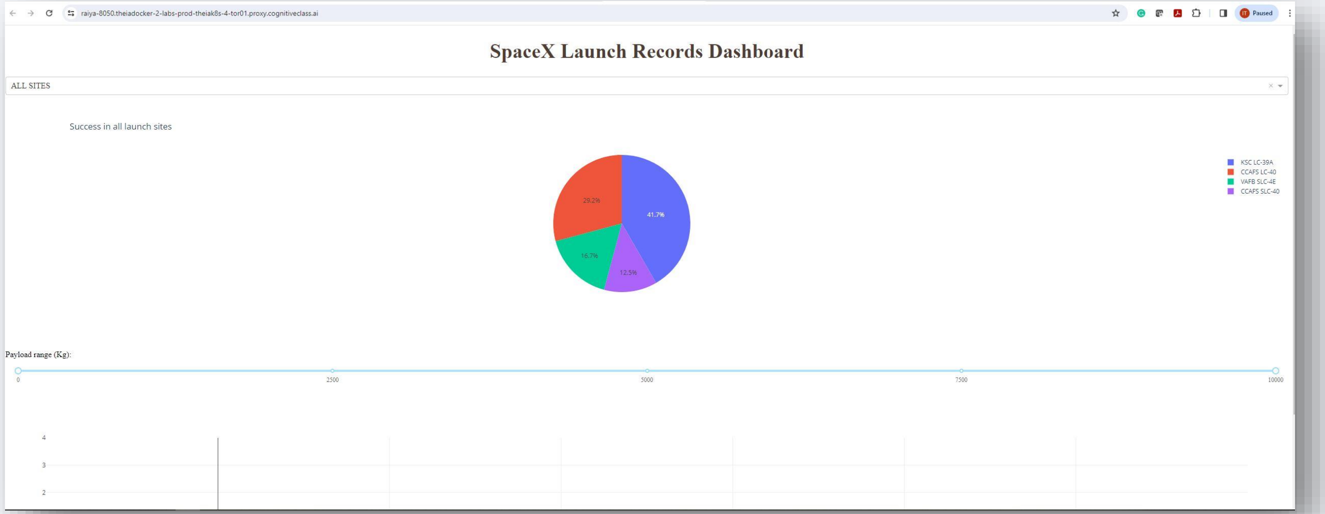
Build an Interactive Map with Folium

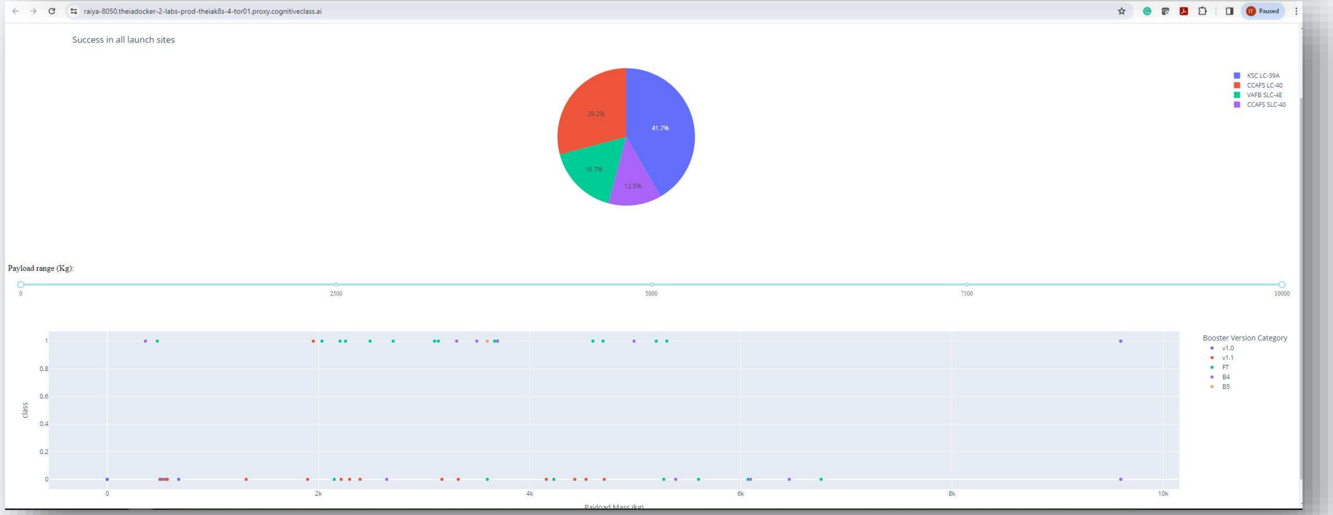
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- We first create a **folium Map** object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
- We used **folium.Circle** to add a highlighted circle area with a text label on a specific coordinate
- Create and add folium.Circle and folium.Marker for each launch site on the site map
- Add a folium.Marker to marker_cluster for each launch result in spacex_df data frame,
- Explain why you added those objects
- To add and highlight circle in the location area with a text label
- https://github.com/raiyasaid91/raiyarepo/blob/master/Raiya_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard

- We create an interactive dashboard with Plotly dash
- We add pie charts to show the total launches for each sites
- We Add scatter plot graph showing the relationship with Outcome and Payload Mass (Kg) for each version.
- https://github.com/raiyasaid91/raiyarepo/blob/master/spacex_dash_app.py





Predictive Analysis (Classification)

- We imported and processed data with numpy and pandas, conducted data transformation, and partitioned it into training and testing sets.
- Various machine learning models were constructed, and diverse hyperparameters were fine-tuned using GridSearchCV.
- Our model's performance was evaluated based on accuracy, and enhancements were made through feature engineering and algorithm tuning.

- https://github.com/raiyasaid91/raiyarepo/blob/master/Raiya_Machine%20Learning%20Prediction.ipynb

Results

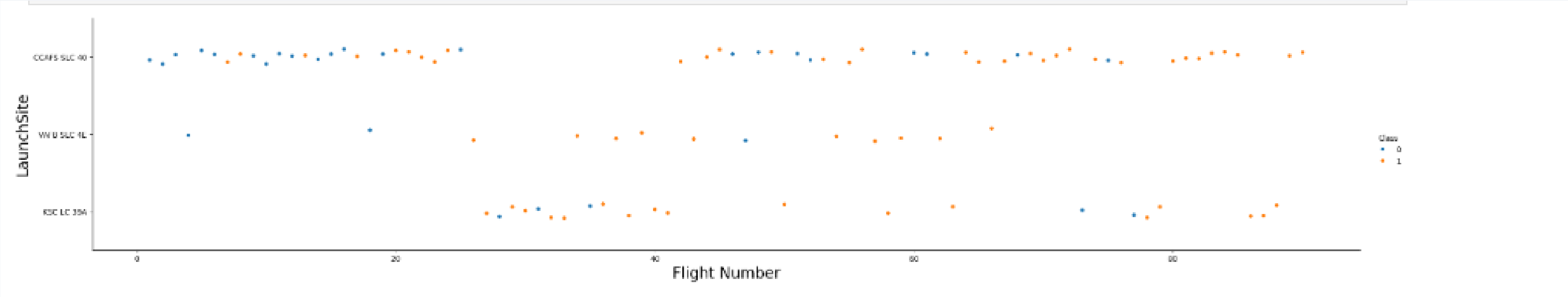
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

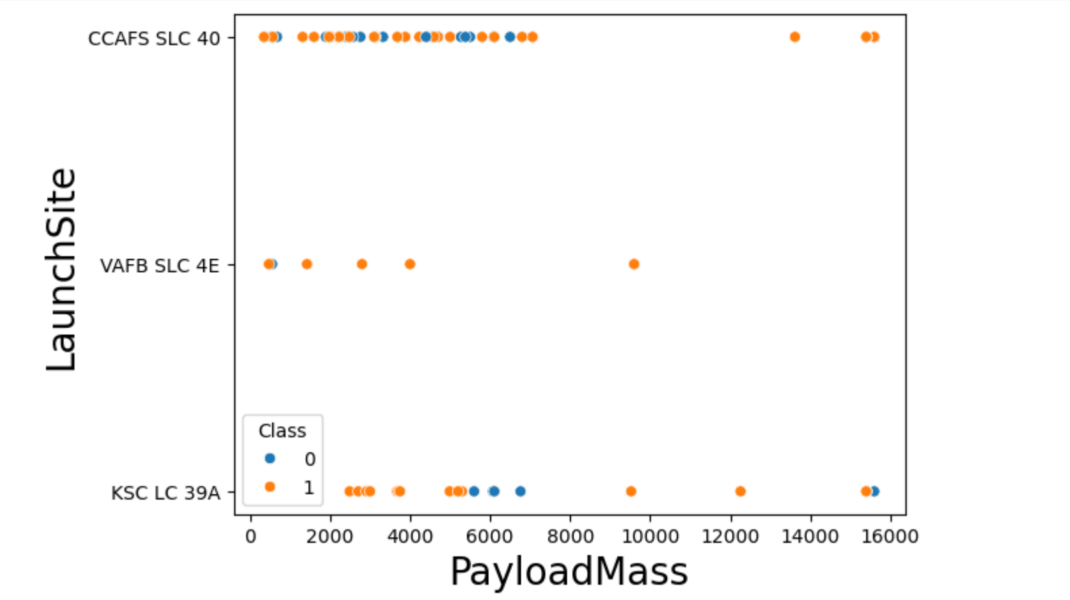
Flight Number vs. Launch Site

- the plot shows that VAFB SLC and KSC LC 39A has higher success rate than CCAFS SLL 40 but CCAFS SLL 40 has majority of the launches



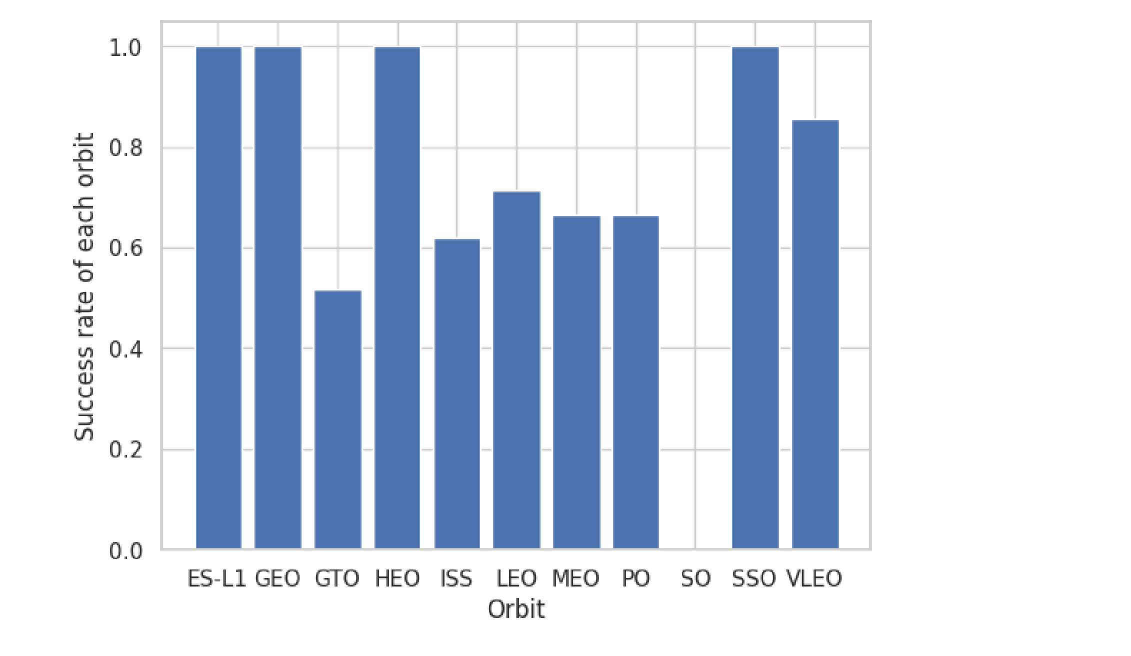
Payload vs. Launch Site

- The scatter plot of Payload vs. Launch Site shows for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- The most launches of CCAFS SLC 40 payload between 1000 – 6000
- Success rate for KSC LC 39A higher in less payloadmass



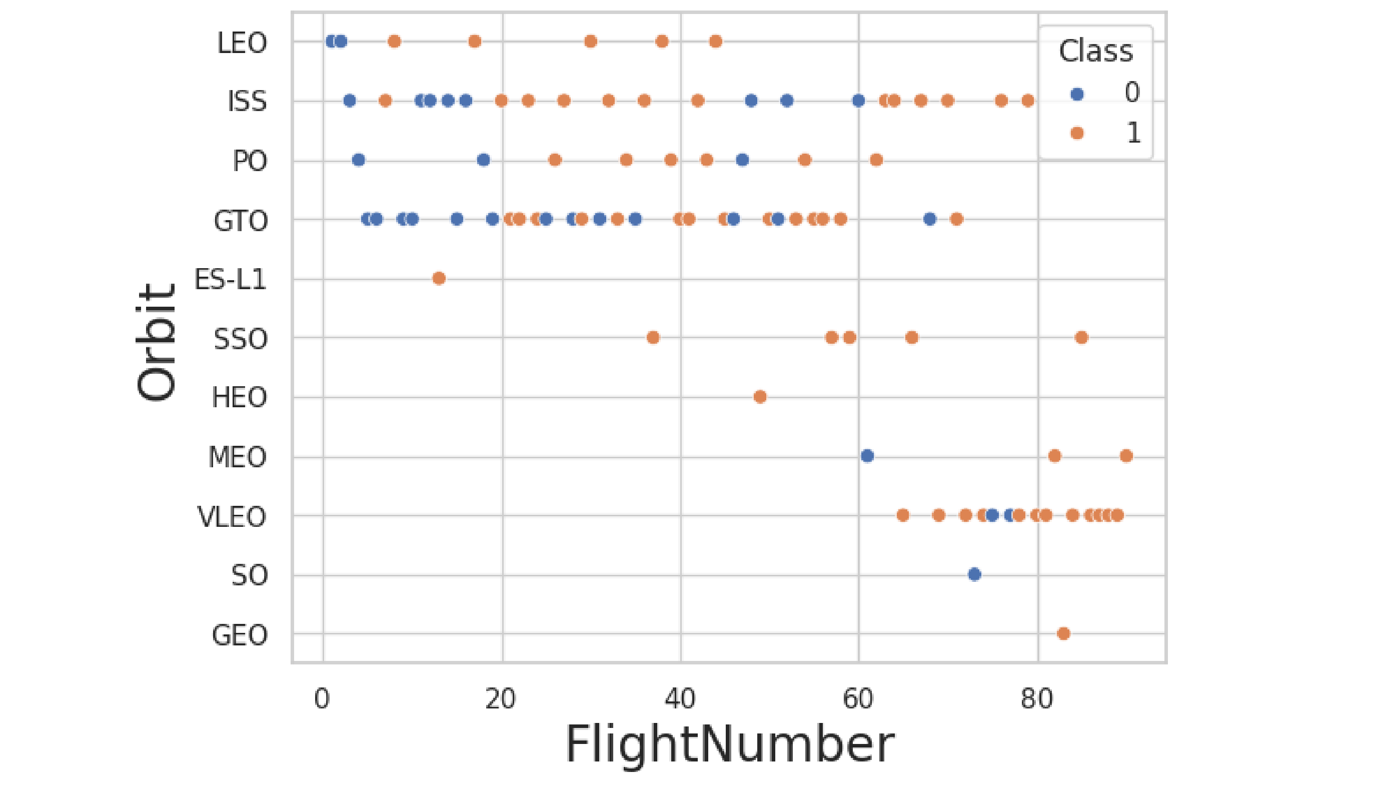
Success Rate vs. Orbit Type

- From the bar chart, we can find that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



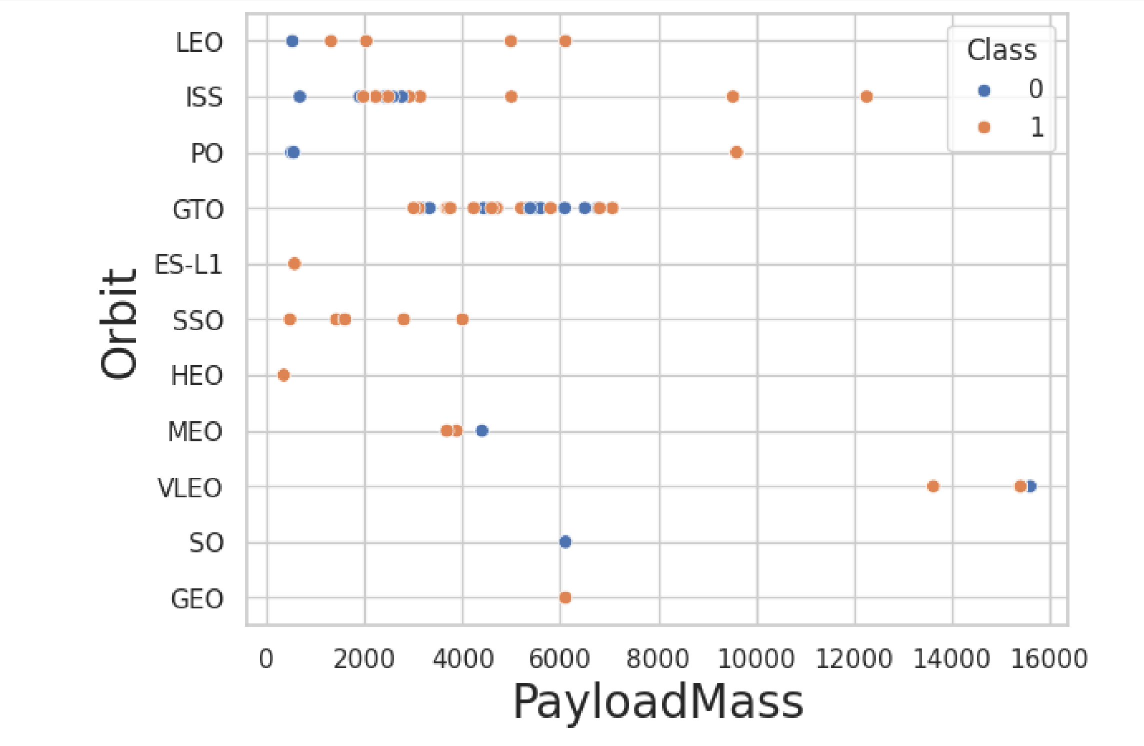
Flight Number vs. Orbit Type

- In LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



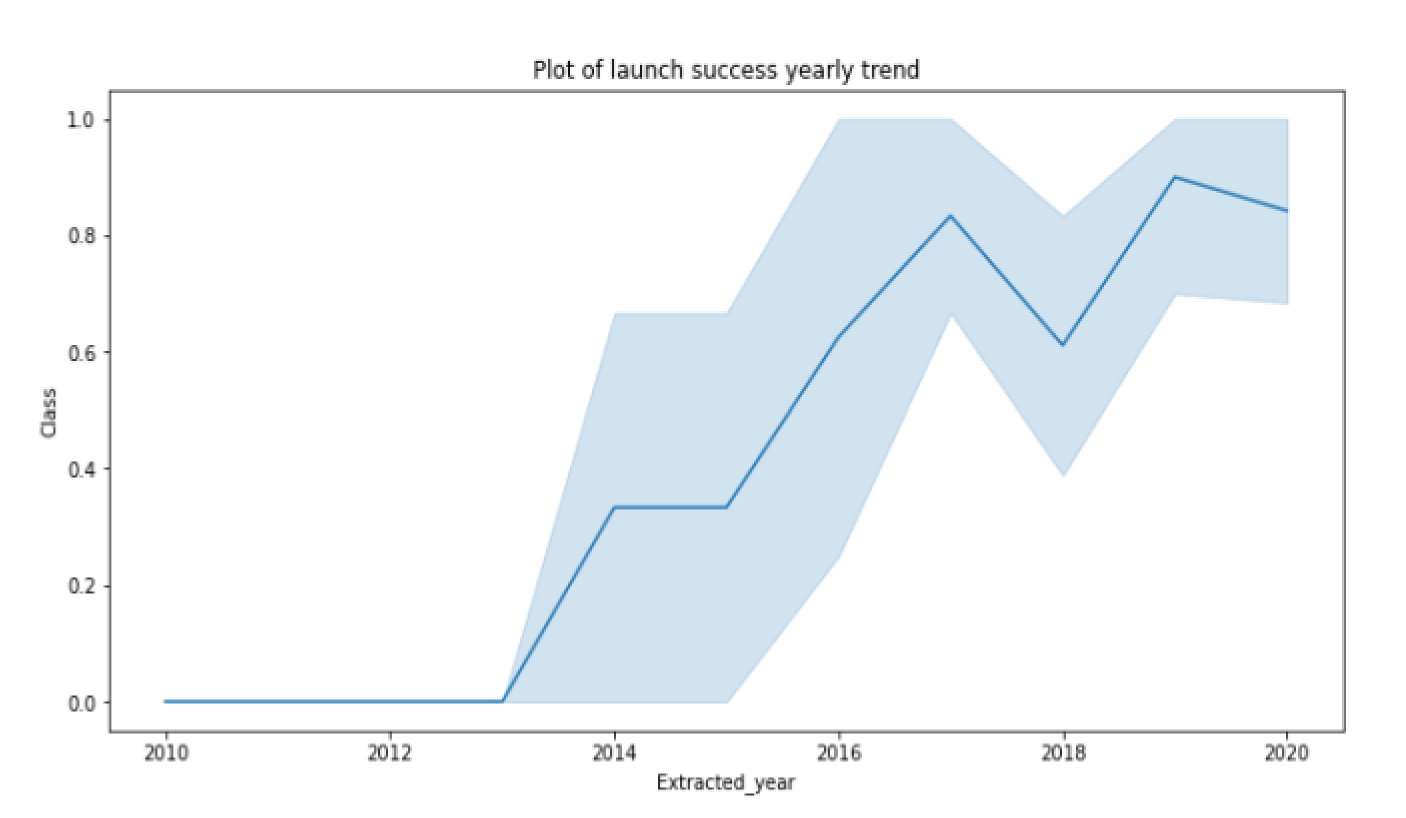
Payload vs. Orbit Type

- In the chart we can find with heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- From the plot, we can find the success rate since 2013 kept on increasing till 2020



All Launch Site Names

- By using DISTINCT in SQL we got result that shows the different lunches site name

Display the names of the unique launch sites in the space mission

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Using the SELECT SQL query and using (where LIKE) to find 5 records launch site 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
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Using SUM function, we can calculate the total

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

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Using AVG() function we calculated the average payload.

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

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- Using MIN() function in date column, we got the first date + where the outcome is Success (ground pad)

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using > < = we are able to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

The total number of success outcome is 100

The total number of failed outcome is 1

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

Boosters Carried Maximum Payload

- Using nested SELECT query + ORDER BY we can list the names of the booster which have carried the maximum payload mass

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- Using BETWEEN to specify the date and LIKE to specify the landingoutcome name to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Rank 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
- We used COUNT() + GROUP BY + ORDER BY to retrieve the count for each outcome

Rank 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

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

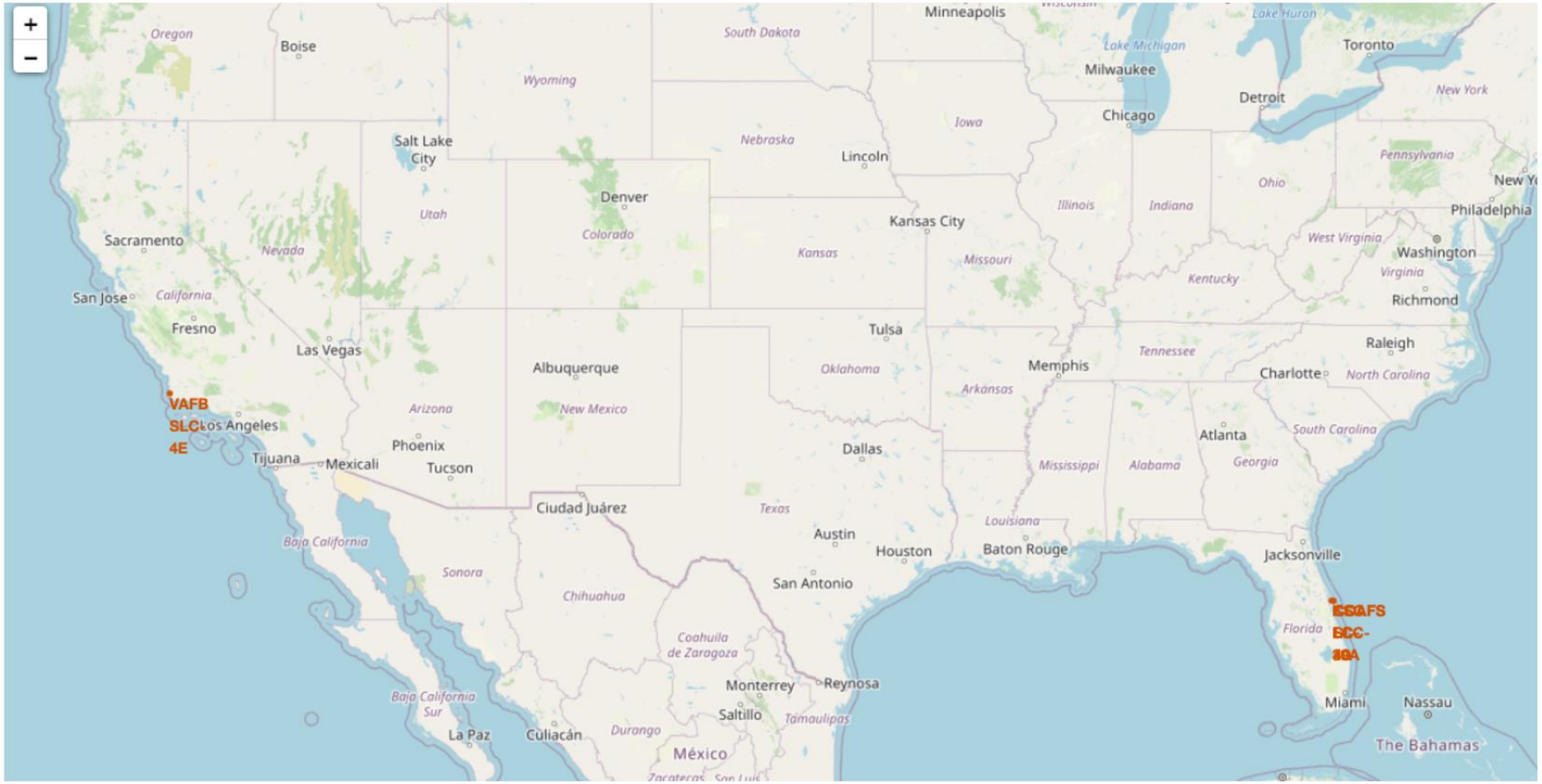
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

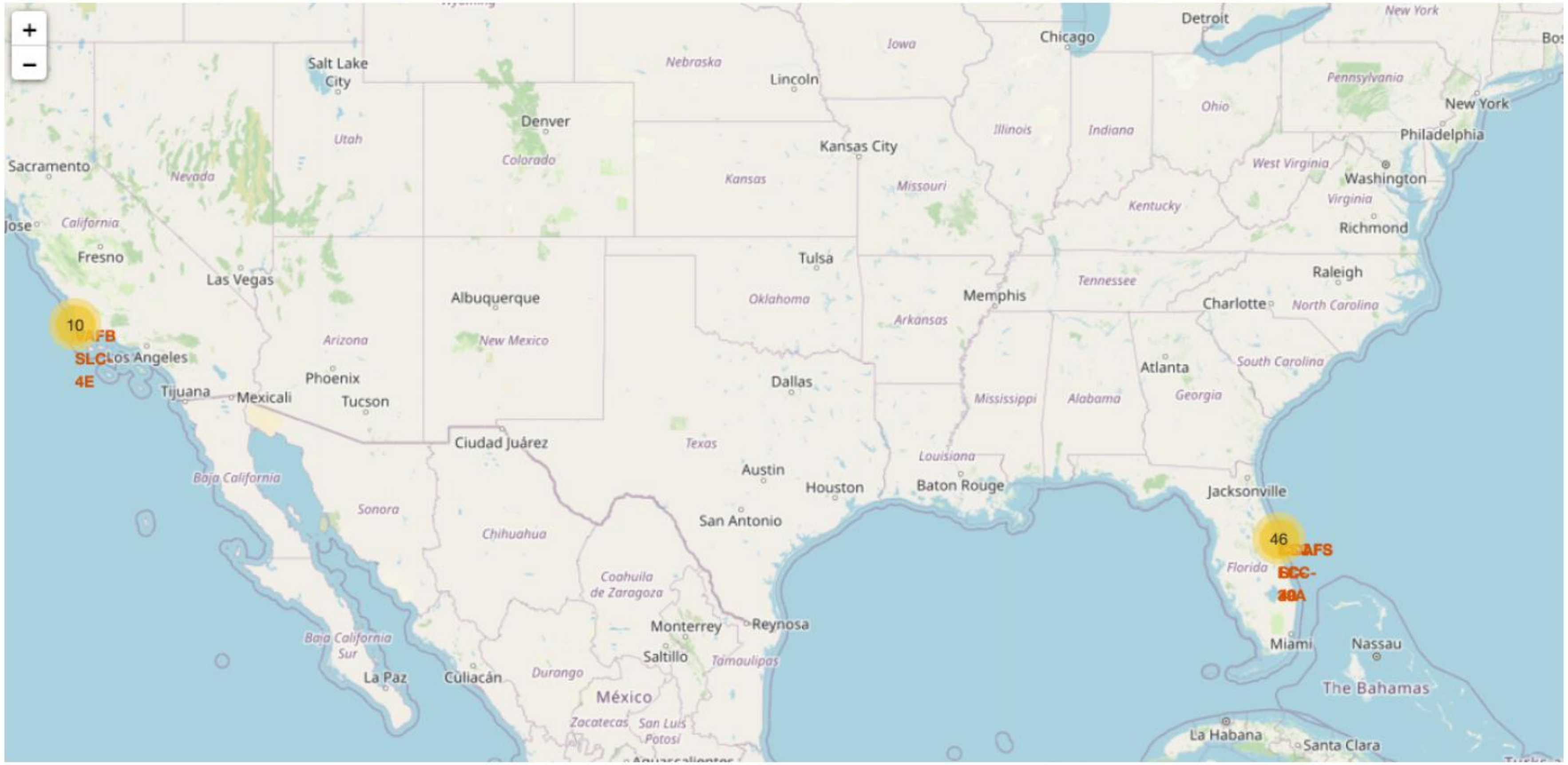
Launch Sites Proximities Analysis

launch sites map markers

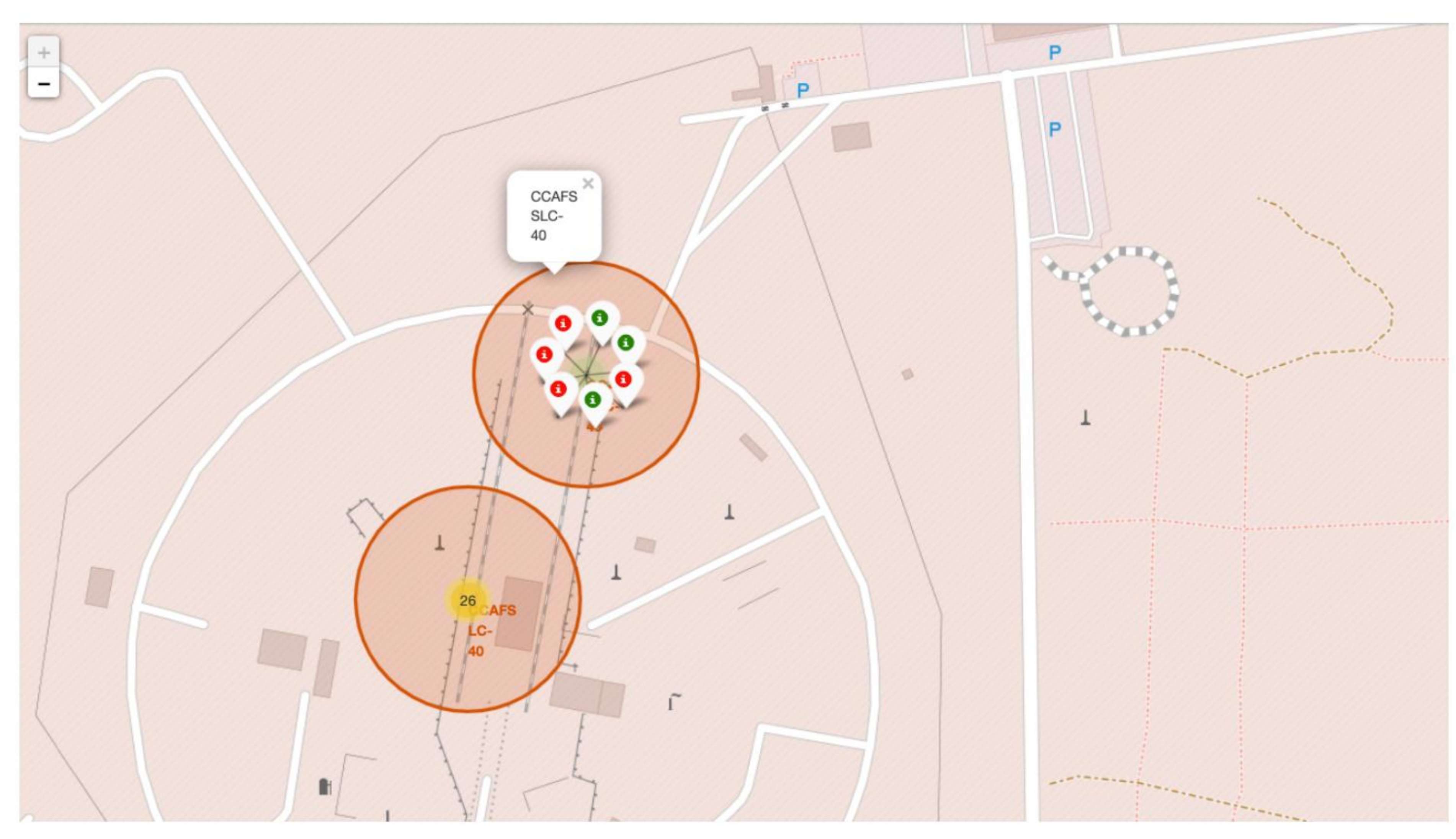
- All sites are in America



Lunch sites with yellow label



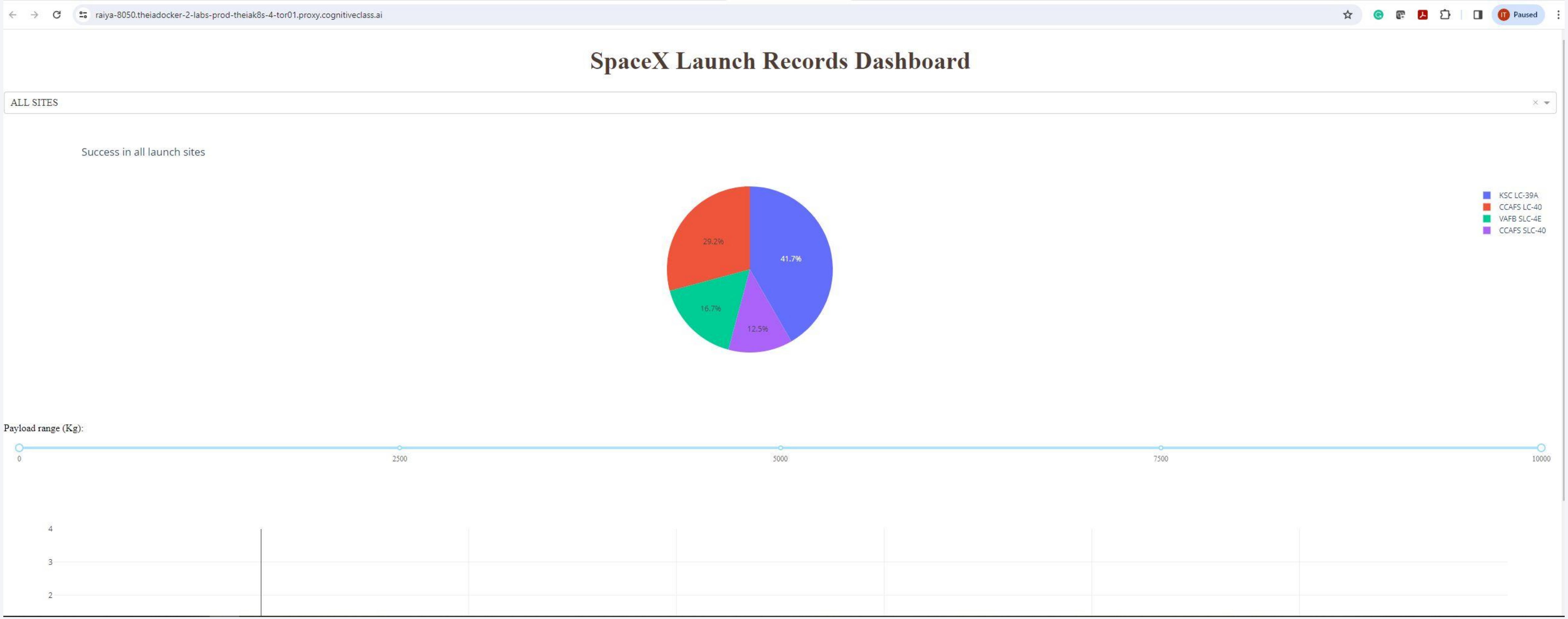
Detailed lunch Sites



Section 4

Build a Dashboard with Plotly Dash

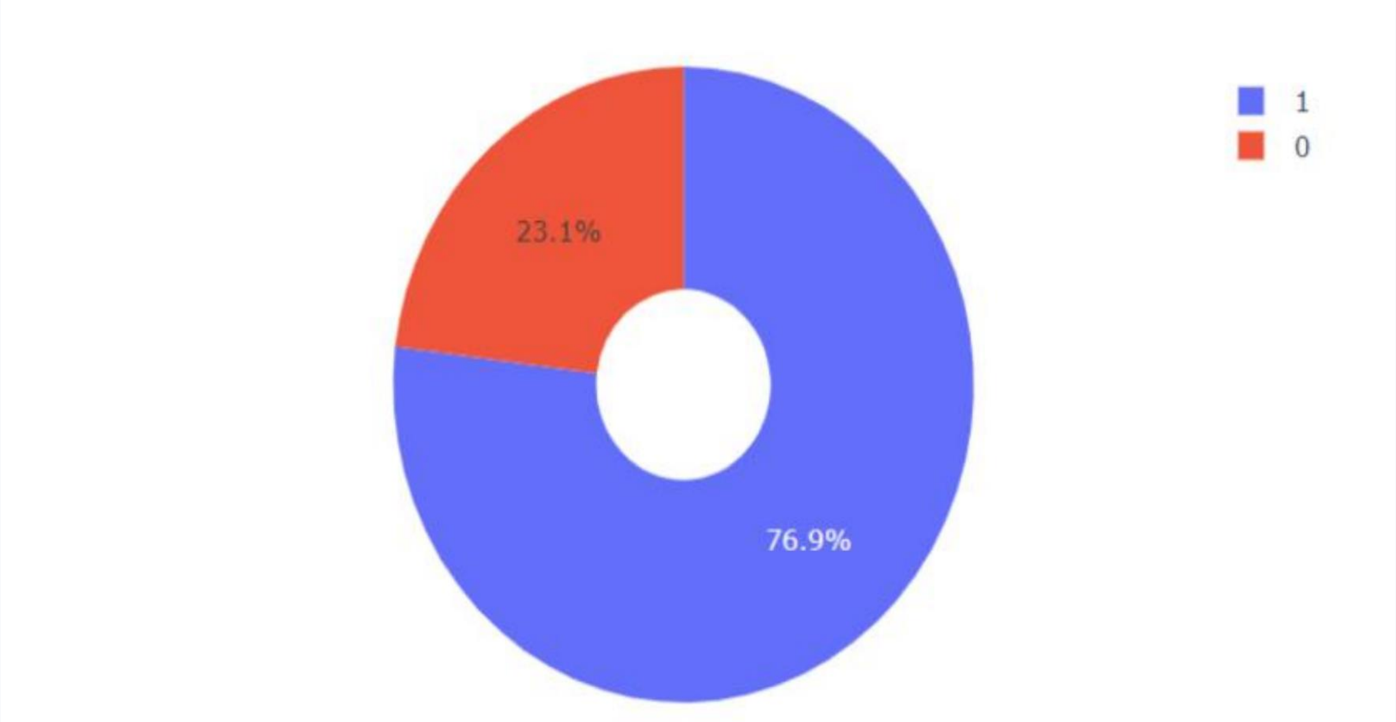
Pie chart Of success percentage achieved by each launch site



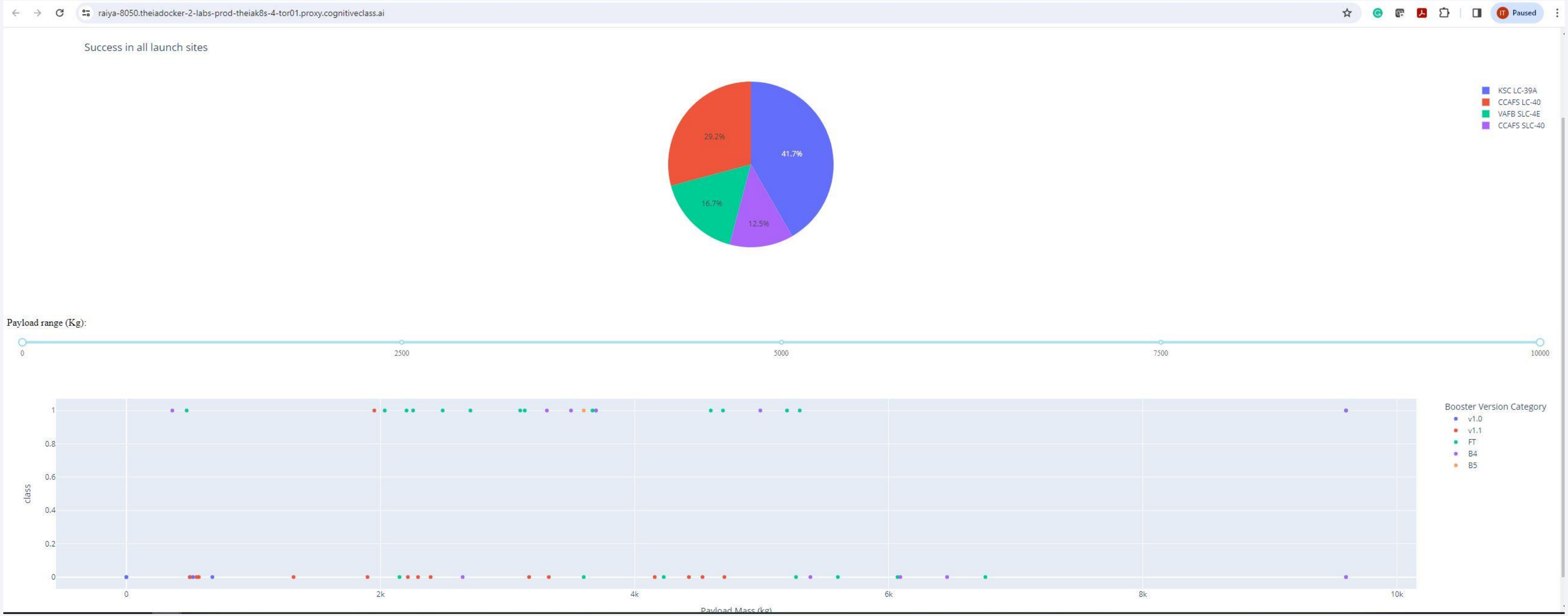
This shows the has KSC LC 39 A the most successful lunches

Pie chart for Launch site with the highest launch success ratio

- This chart shows the highest success ratio is for **KSC LC 39 A** site



Scatter plot of Payload vs Launch Outcome for all sites

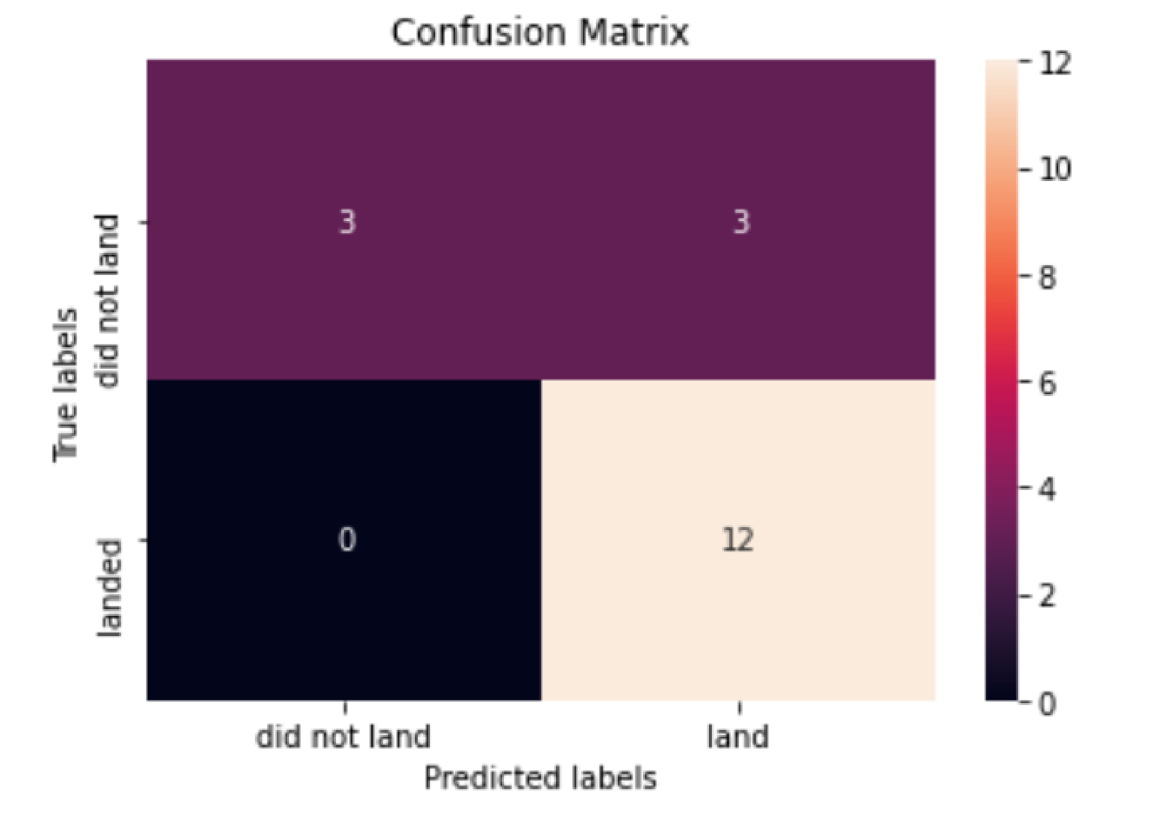


Section 5

Predictive Analysis (Classification)

Confusion Matrix

- The decision tree classifier can differentiate between different classes, but it has a significant number of false positives. Which make sometimes misclassifies unsuccessful landings as successful ones.



Conclusions

- The success rate of a launch site is positively correlated with the number of flights launched from that site.
- The orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rates.
- KSC LC-39A had the highest number of successful launches among all sites.
- The decision tree classifier is the most effective machine learning algorithm for this task
- The launch success rate has been increasing since 2013 and continued to do so until 2020.

Thank you!