IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Leen
24/10/2022

Outline

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

Executive Summary

- Summary of methodologies

1. Data Collection
2. Data wrangling
3. EDA with SQL
4. EDA with data visualization
5. Interactive maps with Folium
6. Predictive Analysis

- Summary of all results

EDA results, predictive, and interactive analysis

Introduction

- Project background and context

SpaceX is the most successful of the spaceflight providers, allowing spacecraft to be sent to the international Space Station unmanned, achieving relatively inexpensive launches.

SpaceX advertises Falcon 9 rocket launches and are able to reuse the first stage. Unlike other rocket providers, Falcon 9 can be recovered at the first stage. However, sometimes stage one crashes, or does not land, and in that case, SpaceX will sacrifice it due to payload, orbit, or customer.

- Problems you want to find answers

Determining whether SpaceX should reuse the first stage instead of using rocket science to determine whether the first stage will successfully land. We will train the machine learning model and use public data to predict whether or not SpaceX is capable of reusing stage 1.

Section 1

Methodology

Methodology

Executive Summary

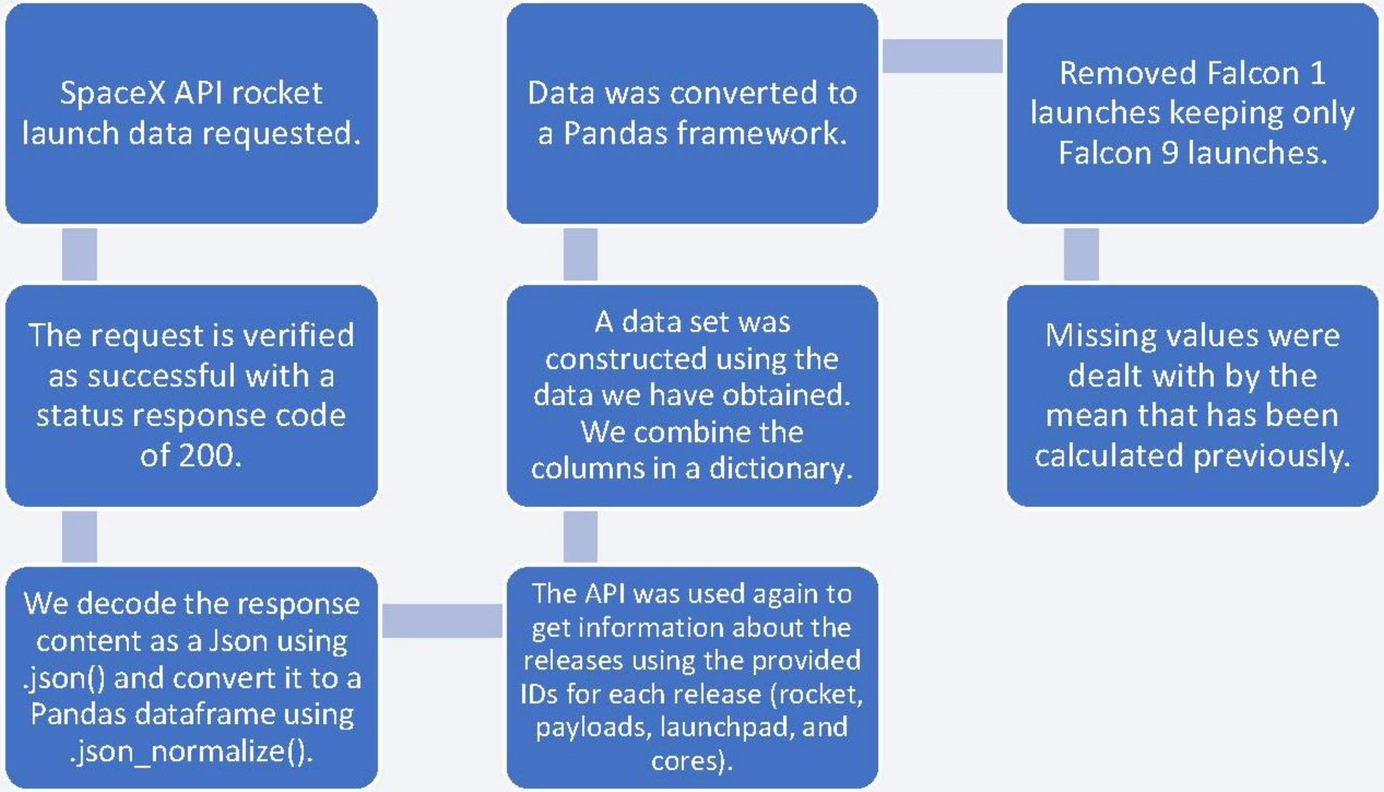
- Data collection methodology:
  - Web scraping and wrangling from the SpaceX API
- Perform data wrangling
  - Pattern recognition with EDA in order to place labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - A machine learning model was created to predict the outcome of what was determined as our goal in order to find the best accuracy.

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- SpaceX API data collected, converted into pandas framework, and filtering to only Falcon 9 launches



SpaceX API rocket launch data requested.

The request is verified as successful with a status response code of 200.

We decode the response content as a Json using .json() and convert it to a Pandas dataframe using .json_normalize().

The API was used again to get information about the releases using the provided IDs for each release (rocket, payloads, launchpad, and cores).

A data set was constructed using the data we have obtained. We combine the columns in a dictionary.

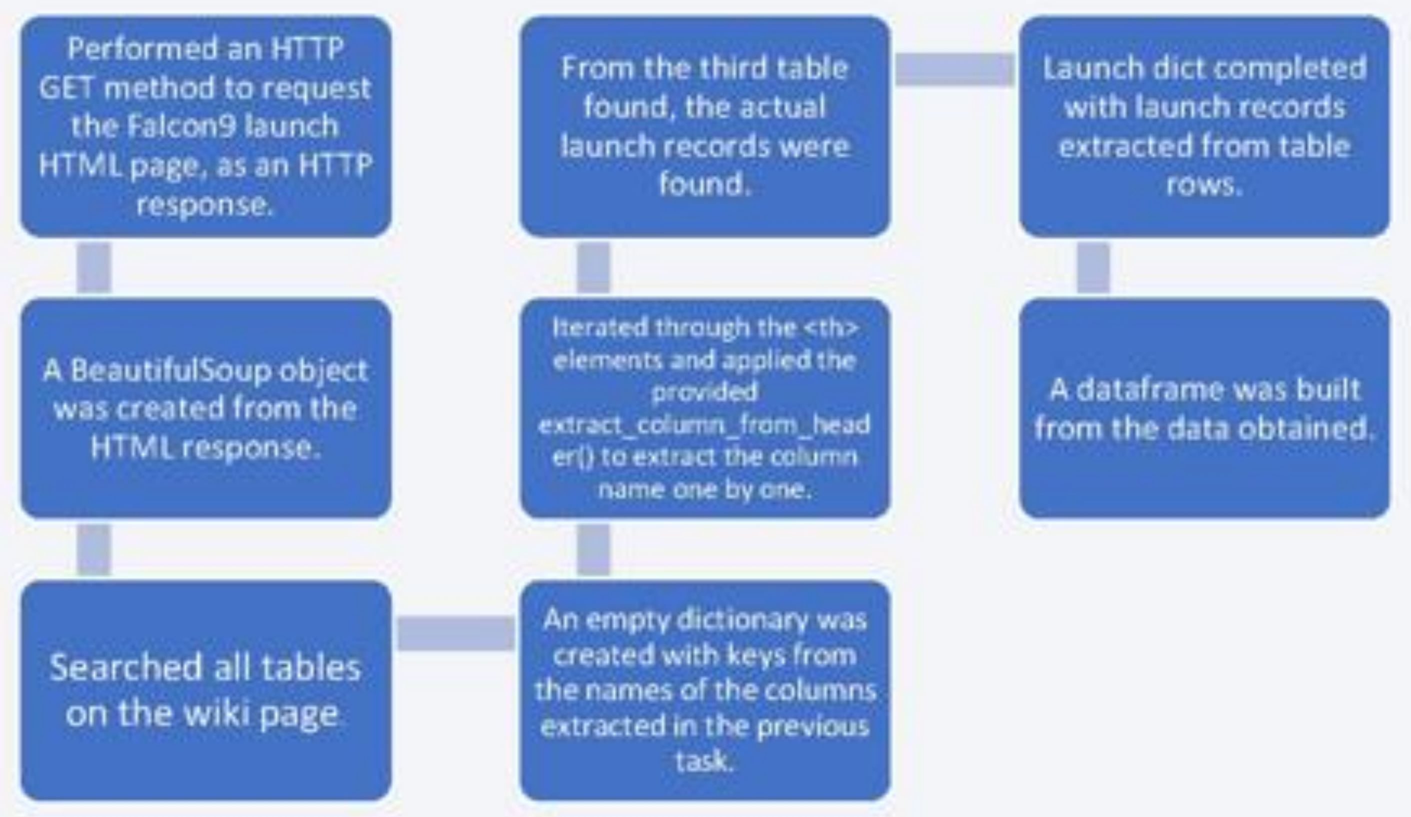
Data was converted to a Pandas framework.

Removed Falcon 1 launches keeping only Falcon 9 launches.

Missing values were dealt with by the mean that has been calculated previously.

Data Collection - Scraping

- We applied web scrapping to webscrape Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Performed an HTTP GET method to request the Falcon9 launch HTML page, as an HTTP response.

A BeautifulSoup object was created from the HTML response.

Searched all tables on the wiki page

From the third table found, the actual launch records were found.

Iterated through the <th> elements and applied the provided extract_column_from_header() to extract the column name one by one.

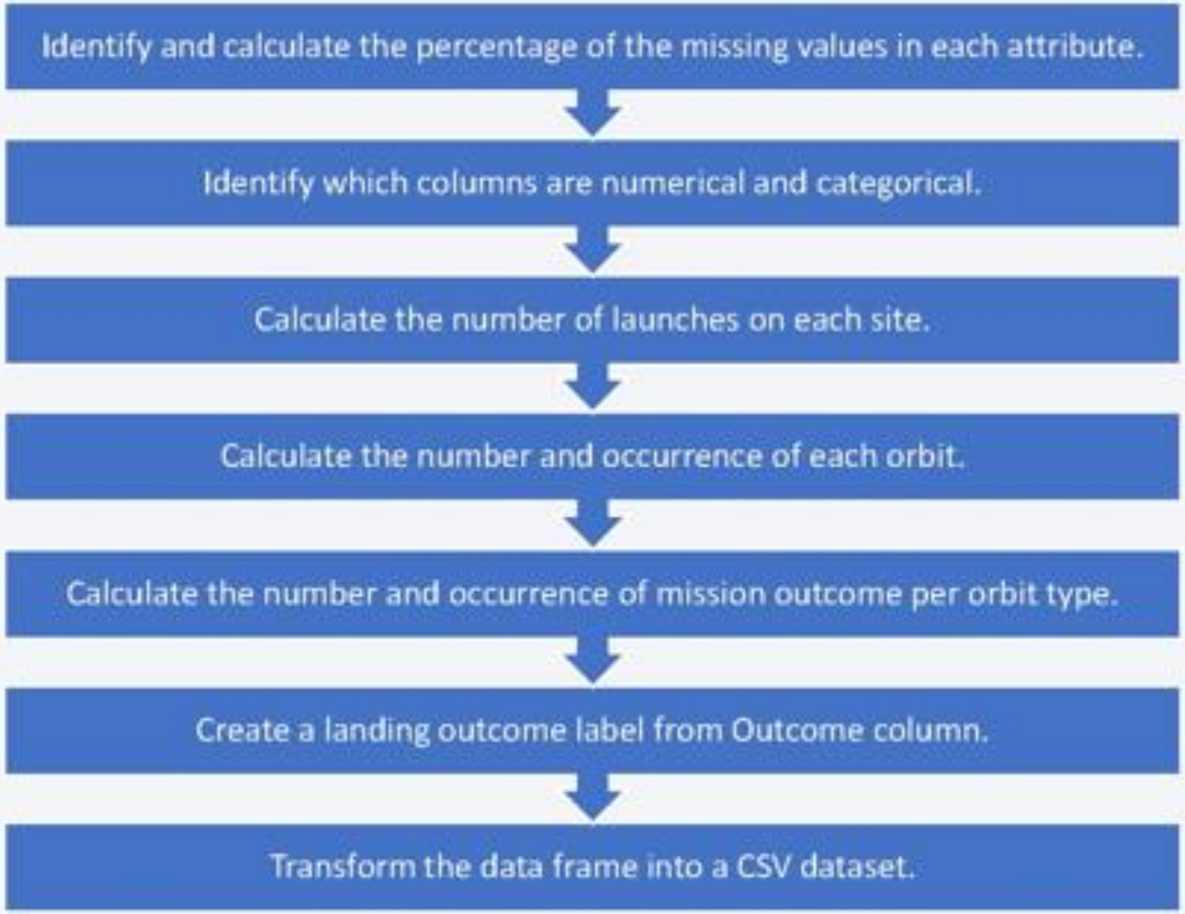
An empty dictionary was created with keys from the names of the columns extracted in the previous task.

Launch dict completed with launch records extracted from table rows.

A dataframe was built from the data obtained.

Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose



Identify and calculate the percentage of the missing values in each attribute.

Identify which columns are numerical and categorical.

Calculate the number of launches on each site.

Calculate the number and occurrence of each orbit.

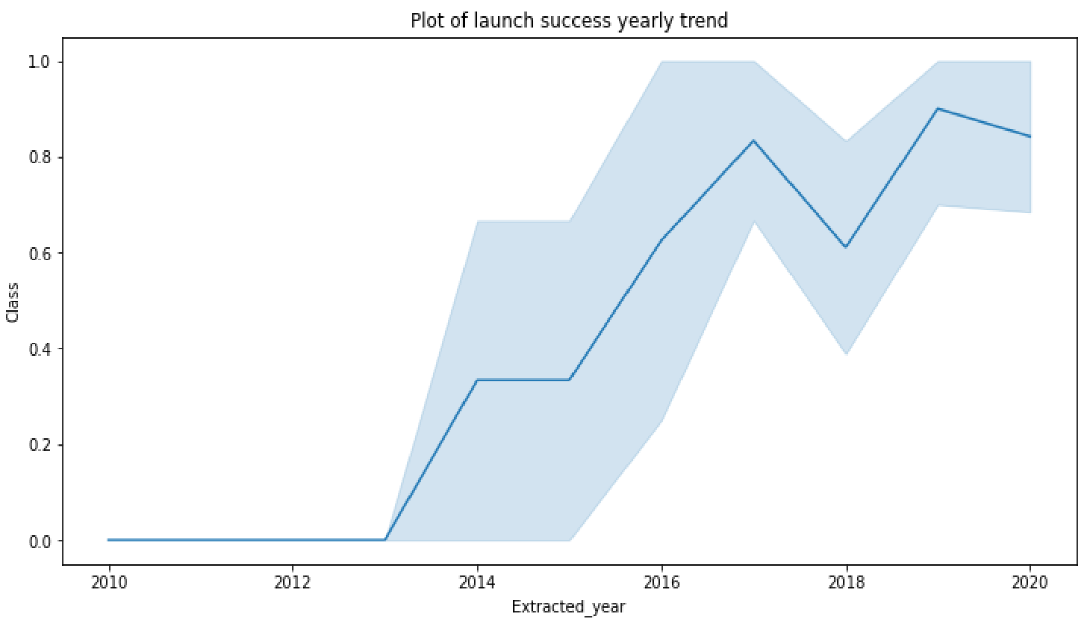
Calculate the number and occurrence of mission outcome per orbit type.

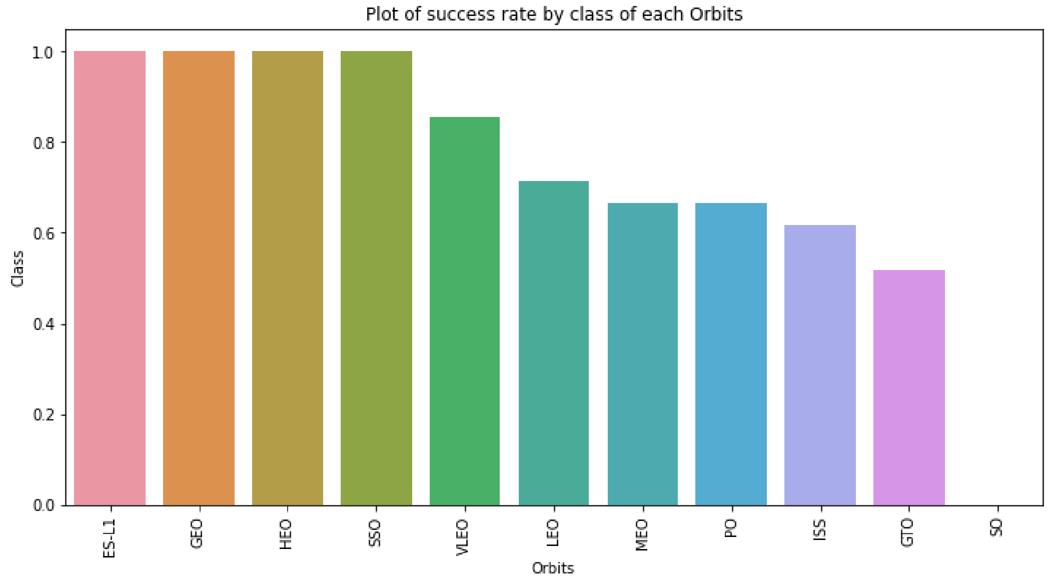
Create a landing outcome label from Outcome column.

Transform the data frame into a CSV dataset.

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

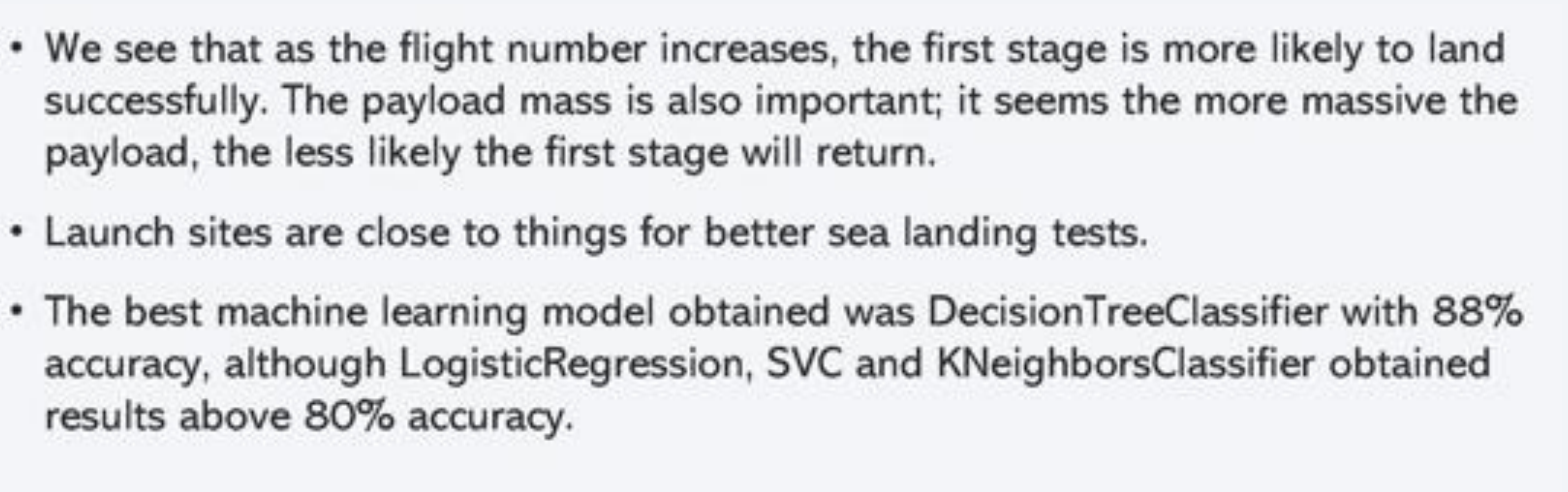
Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results



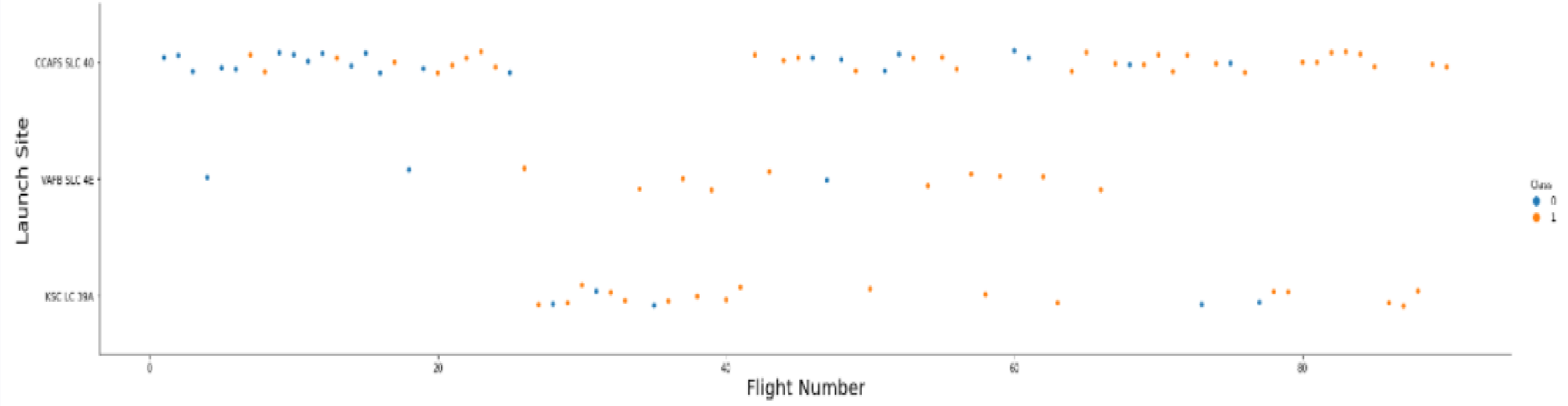
- We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.
- Launch sites are close to things for better sea landing tests.
- The best machine learning model obtained was DecisionTreeClassifier with 88% accuracy, although LogisticRegression, SVC and KNeighborsClassifier obtained results above 80% accuracy.

Section 2

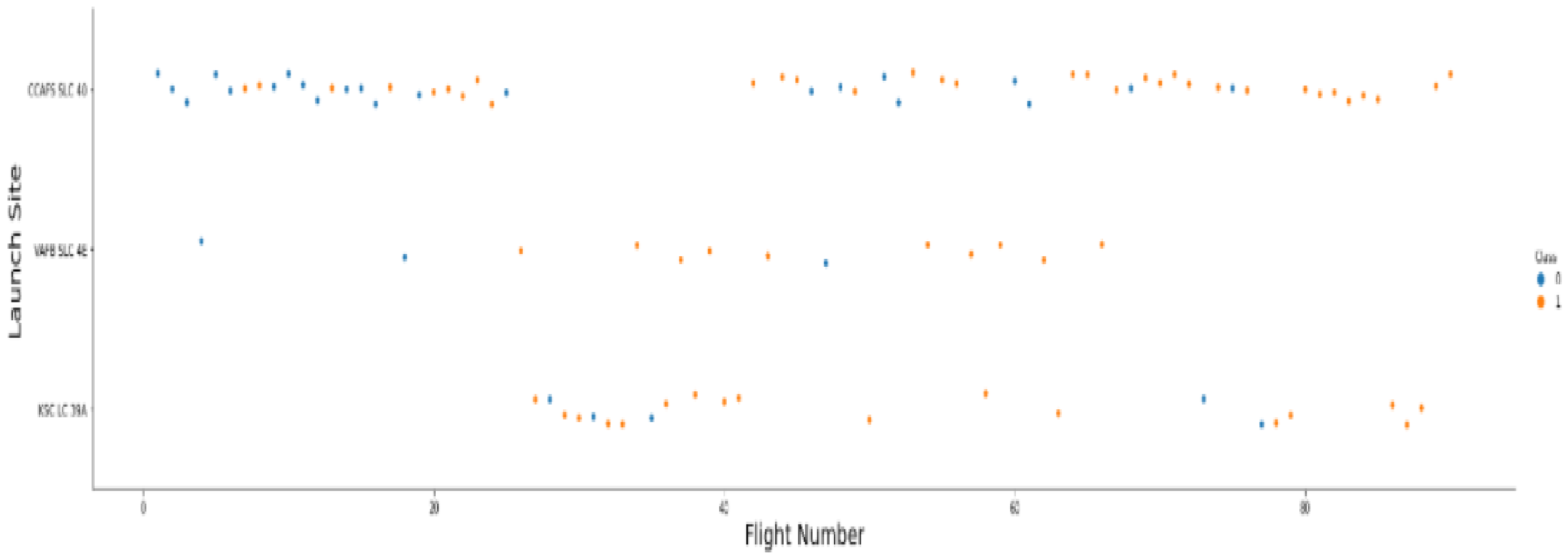
Insights drawn from EDA

Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



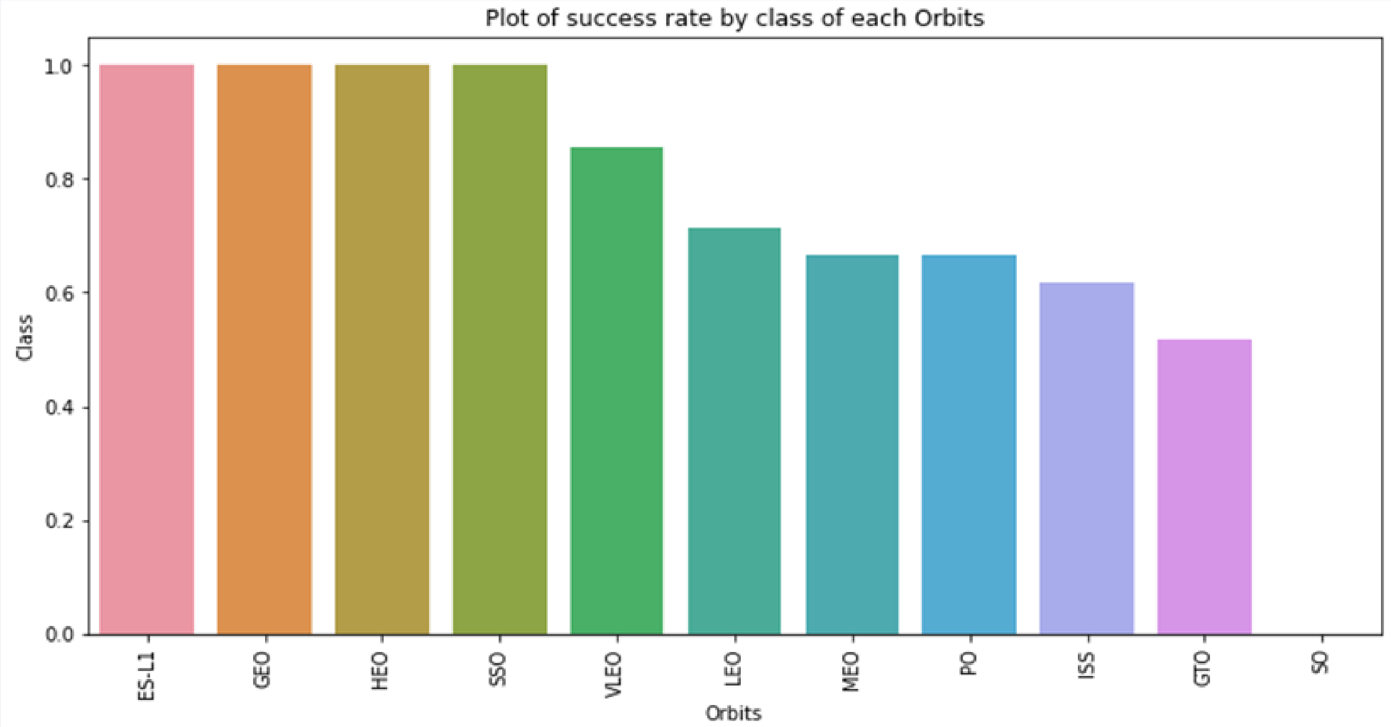
Payload vs. Launch Site



The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.

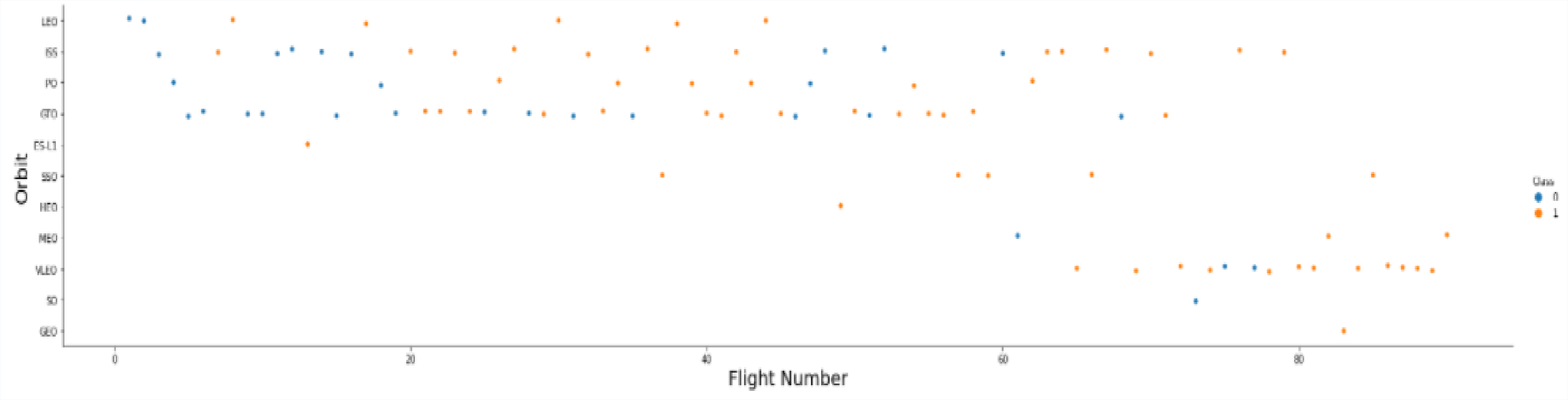
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO had the most success rate according to the chart



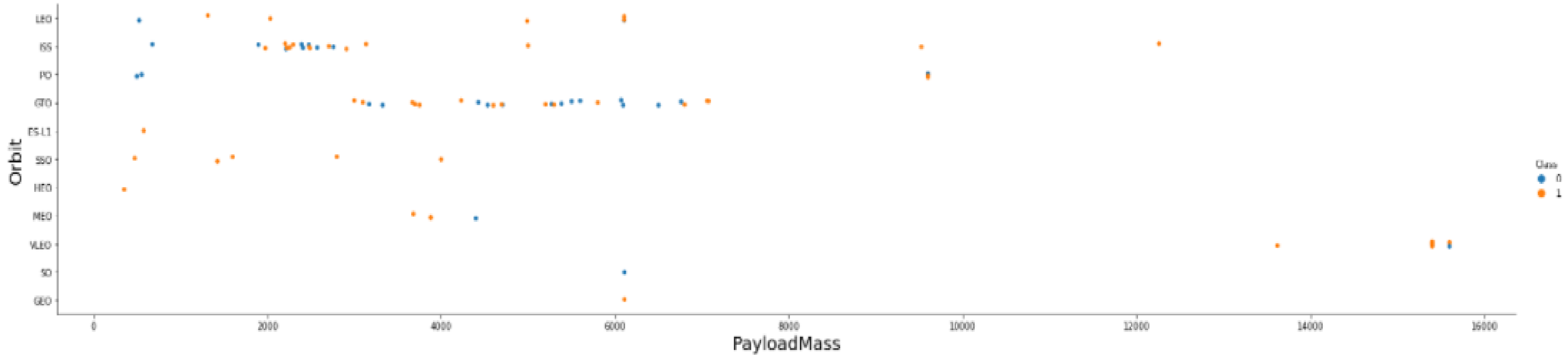
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



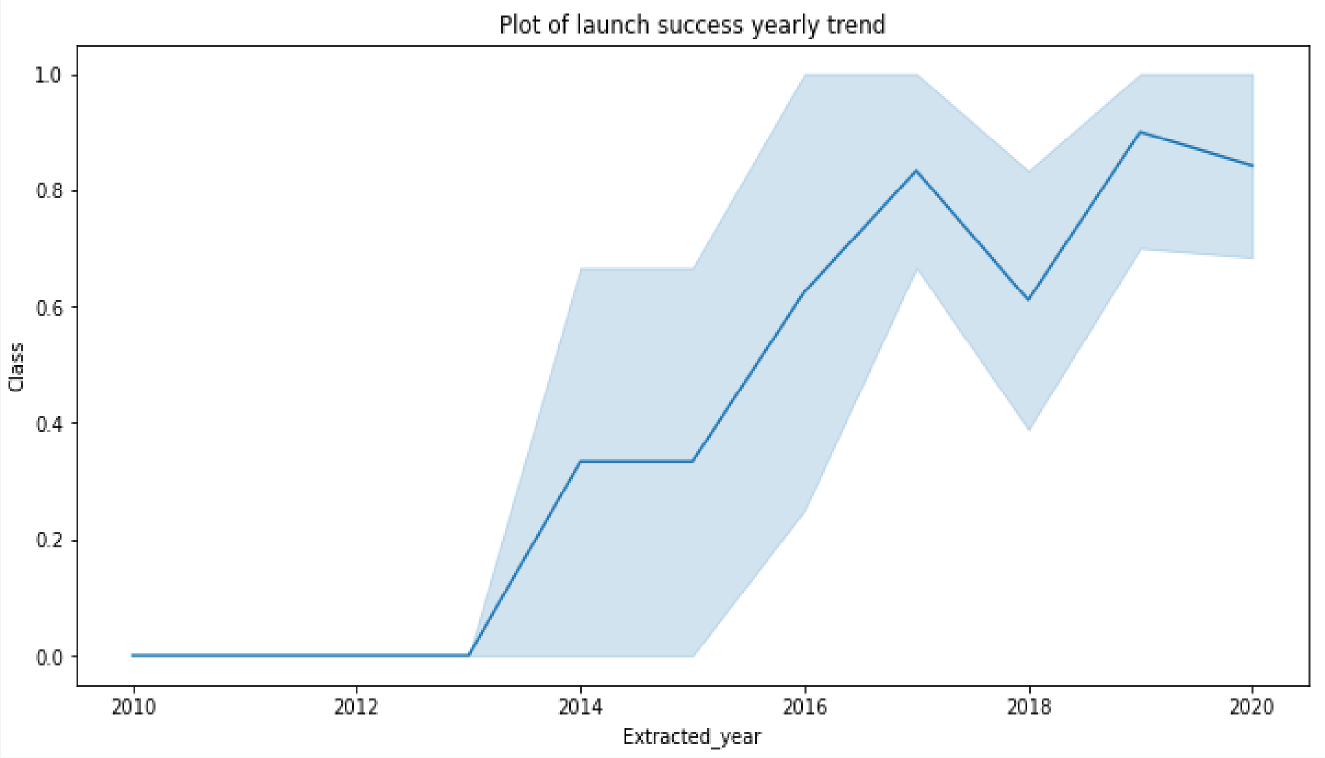
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



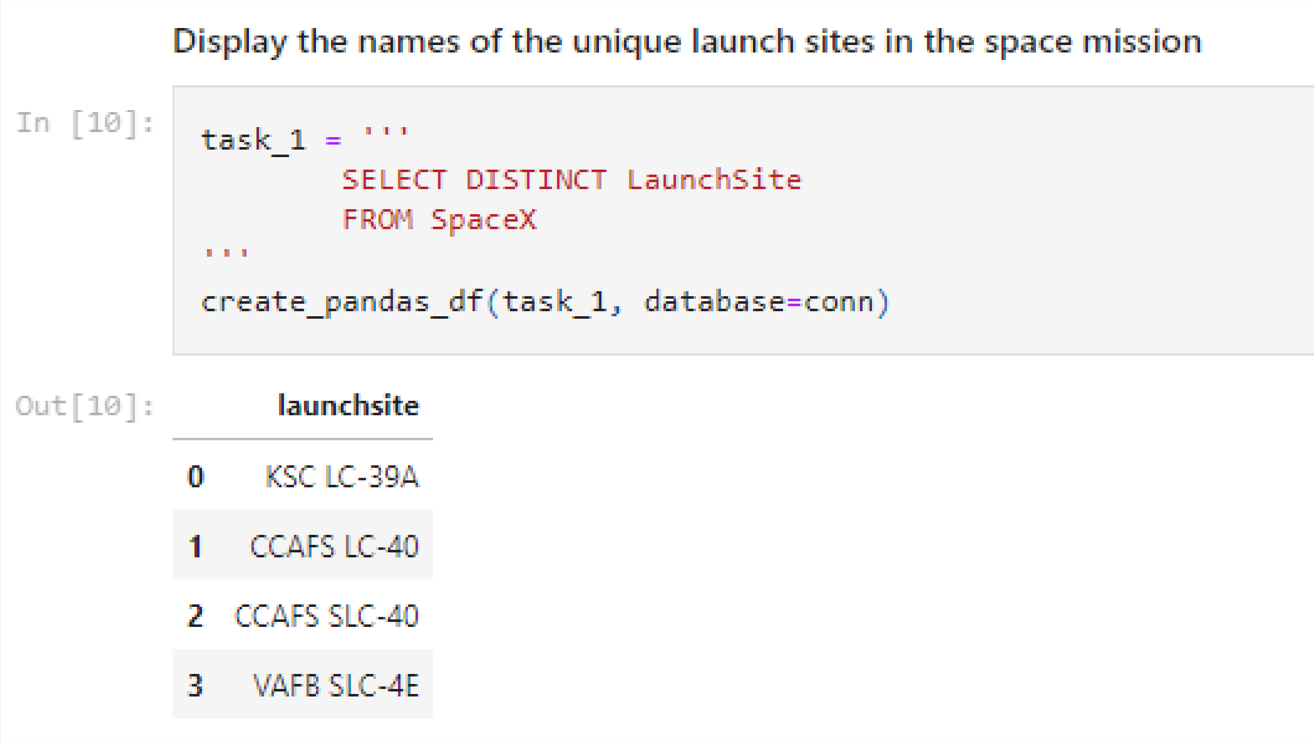
Launch Success Yearly Trend

- Success rate has been increasing from 2013 until 2020 – the end of the timeline



All Launch Site Names

- Using DISTINCT, only the unique launch sites from SpaceX were shown



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

In [10]:

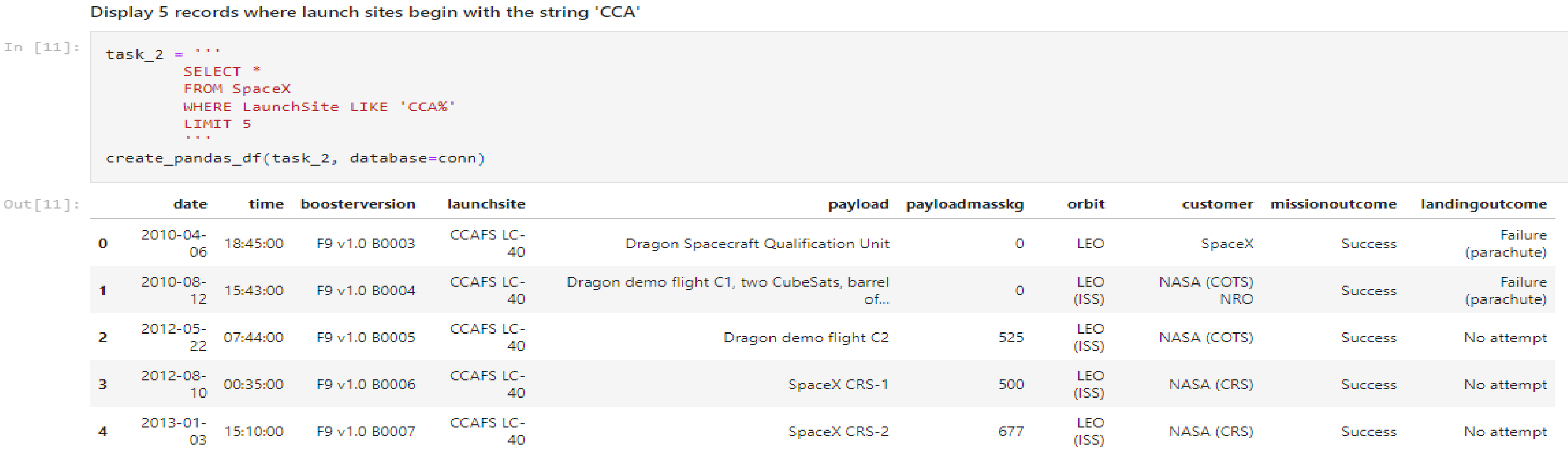
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

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'

- Displaying first 5 CCA launch sites



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

In [11]:

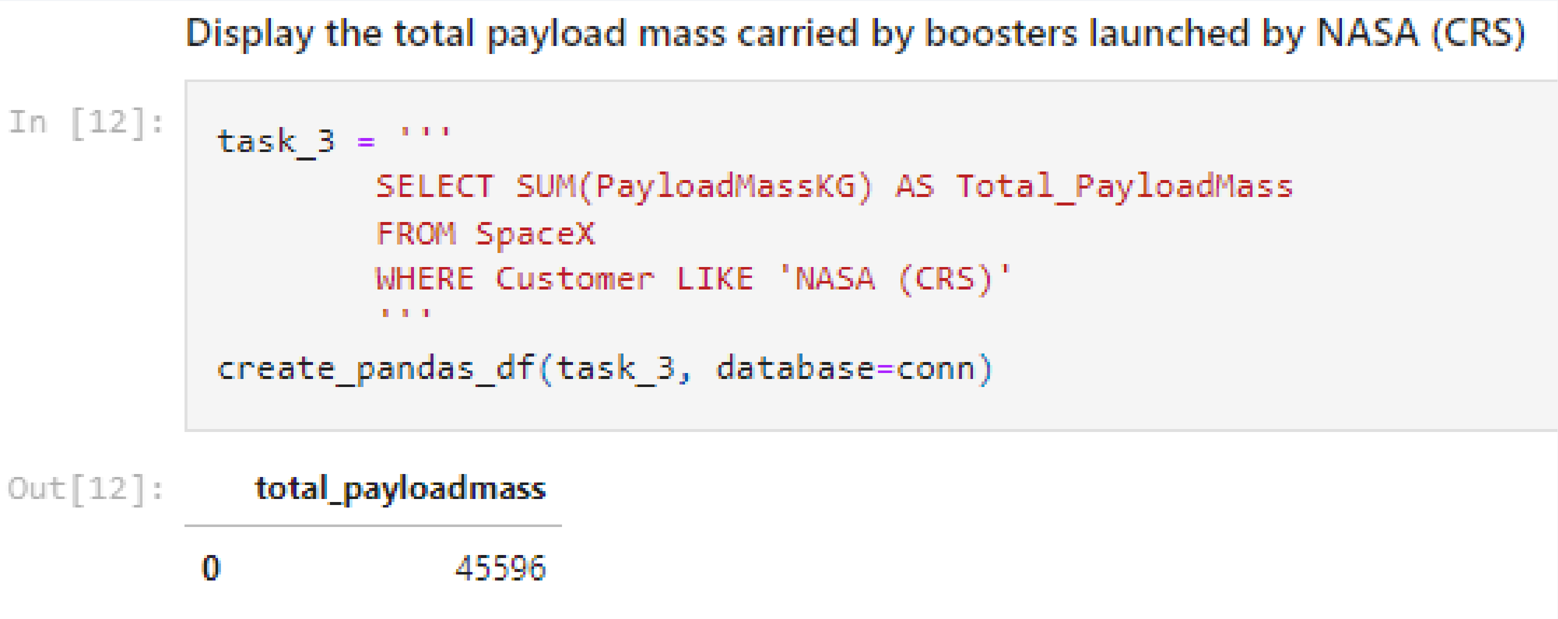
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

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

Total Payload Mass

- Total payload ended up being 45596



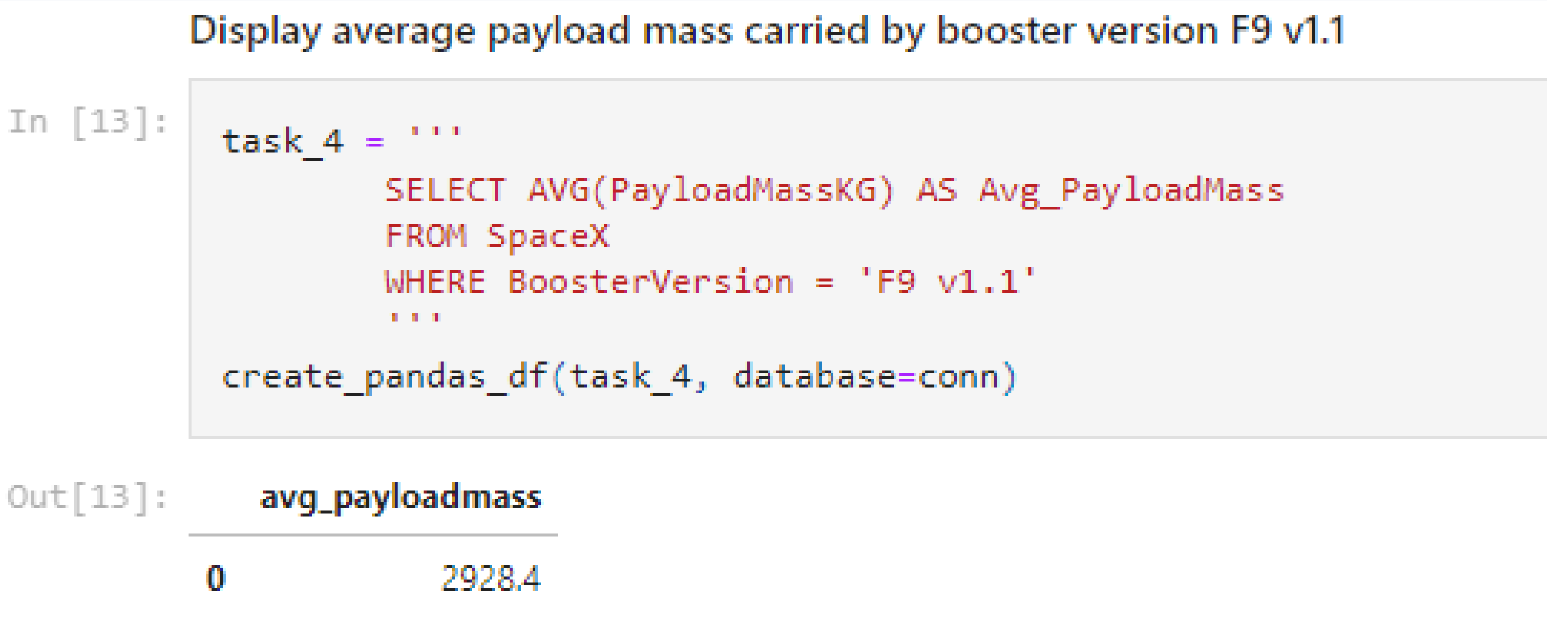
Display the total payload mass carried by boosters launched by NASA (CRS)

```
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

- The average payload ended up being 2928.4



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

In [13]:

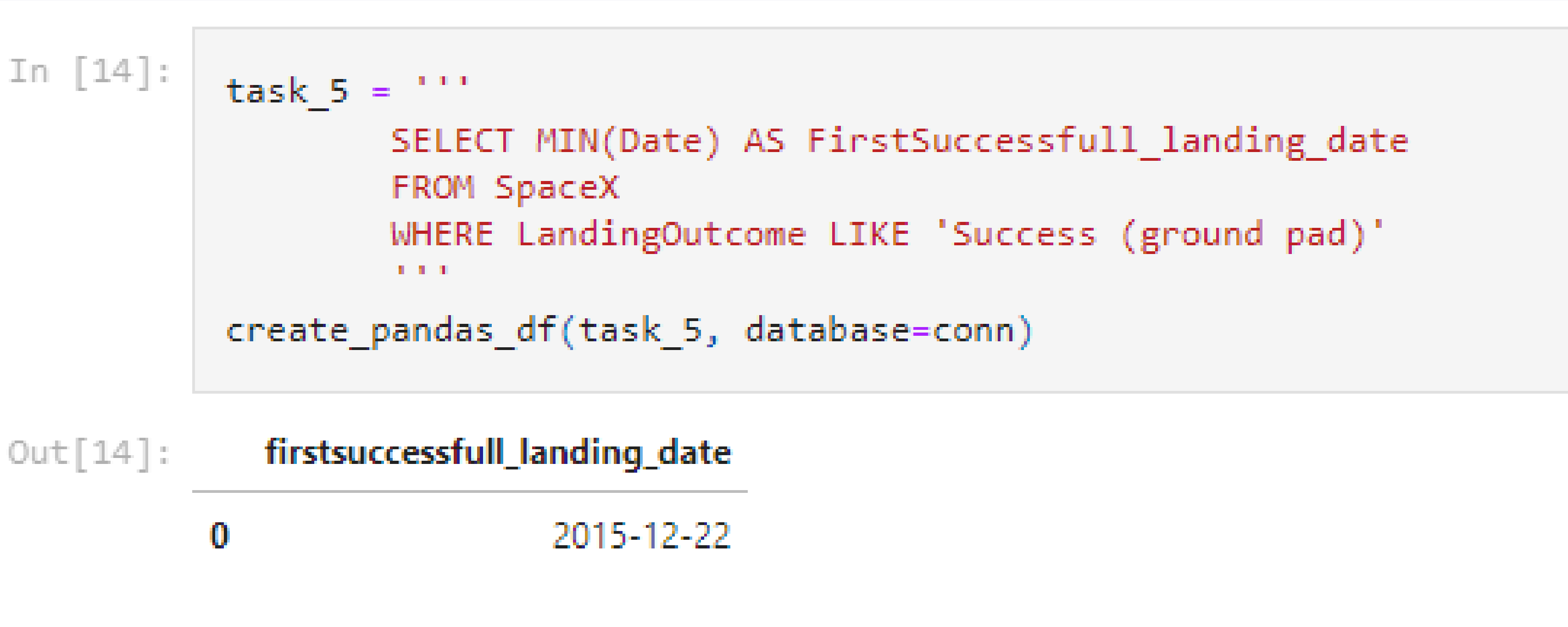
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- The first successful landing happened in 22nd of Dec 2015

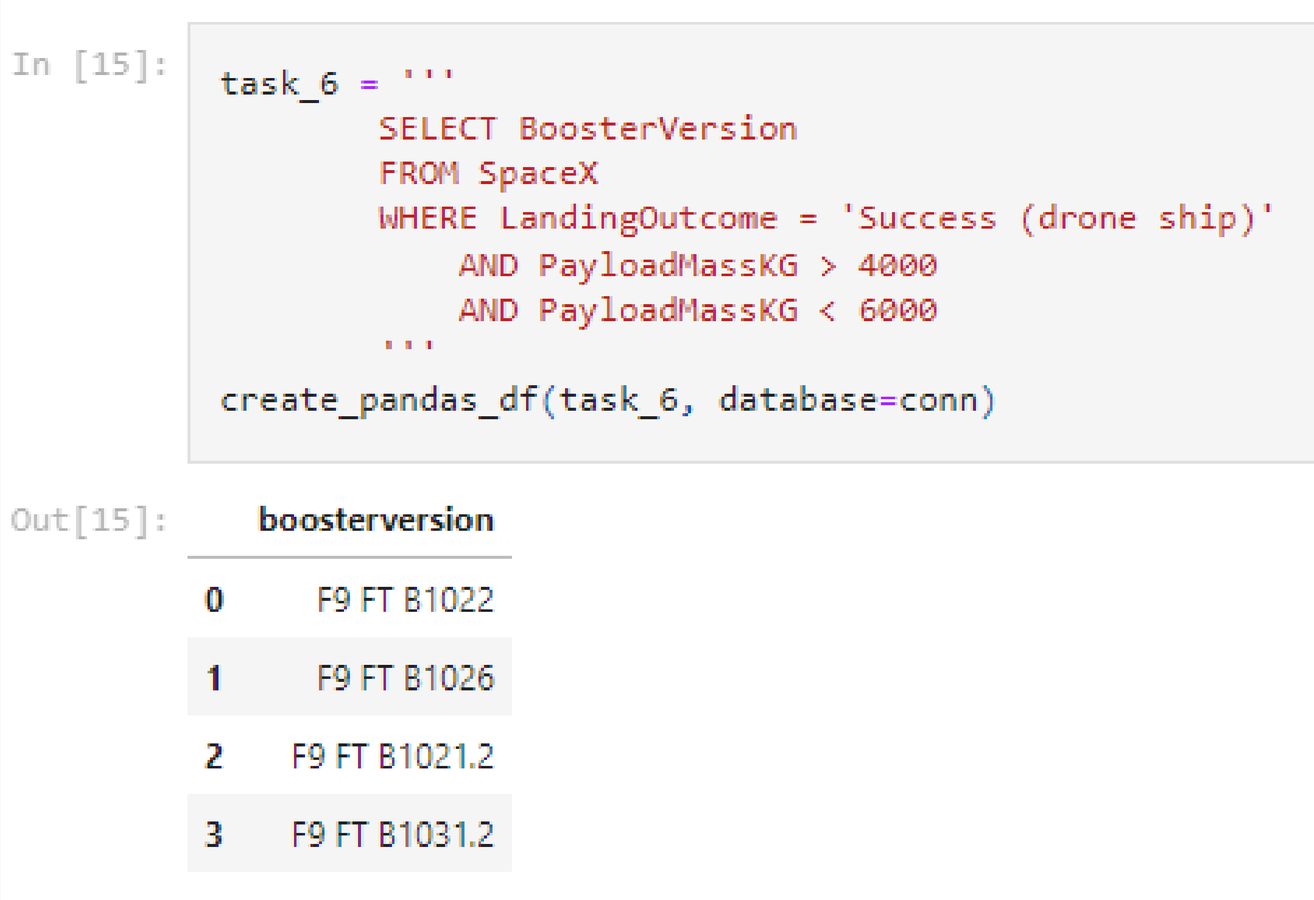


```
In [14]:  task_5 = '''
                  SELECT MIN(Date) AS FirstSuccessfull_landing_date
                  FROM SpaceX
                  WHERE LandingOutcome LIKE 'Success (ground pad)'
                  '''
          create_pandas_df(task_5, database=conn)
```

```
Out[14]:     firstsuccessfull_landing_date
          0                     2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000



In [15]:

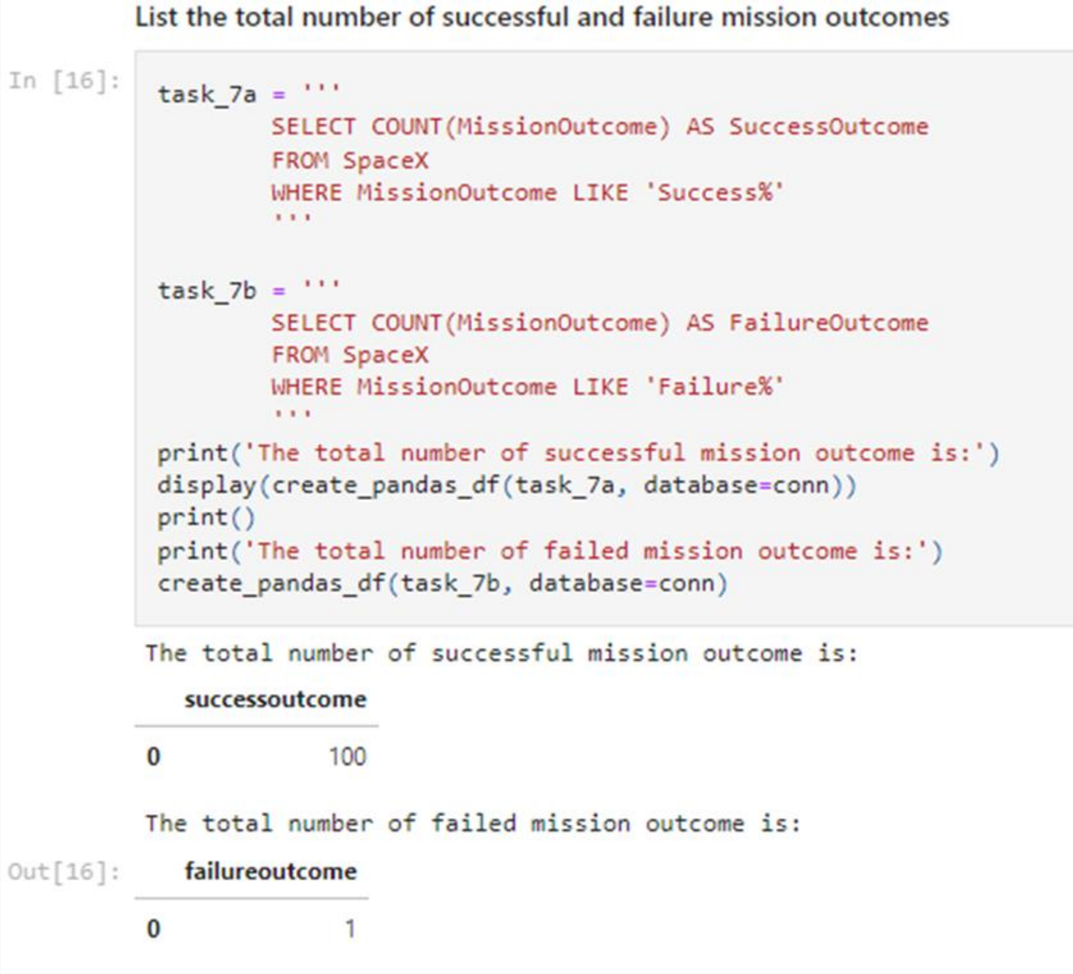
```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	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

- We used wildcard like ‘%’ to filter for **WHERE** MissionOutcome was a success or a failure.
- 120 successful and 1 failure



List the total number of successful and failure mission outcomes

In [16]:

```
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

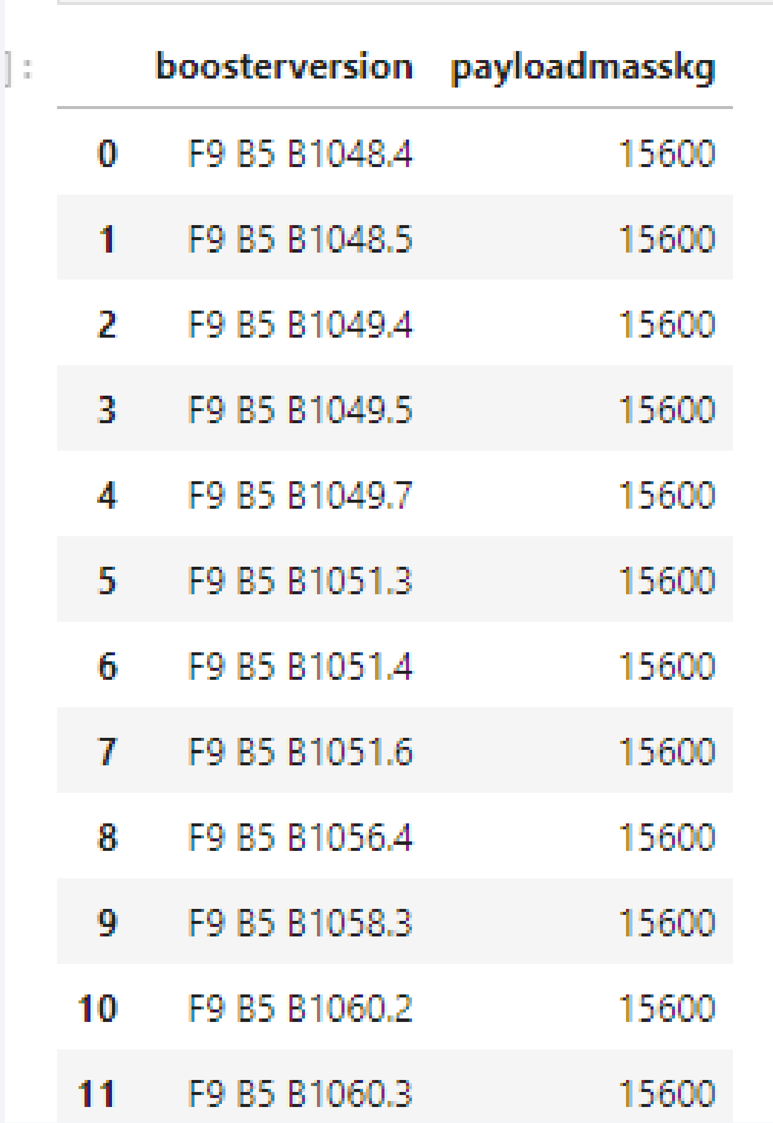
```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Boosters Carried Maximum Payload

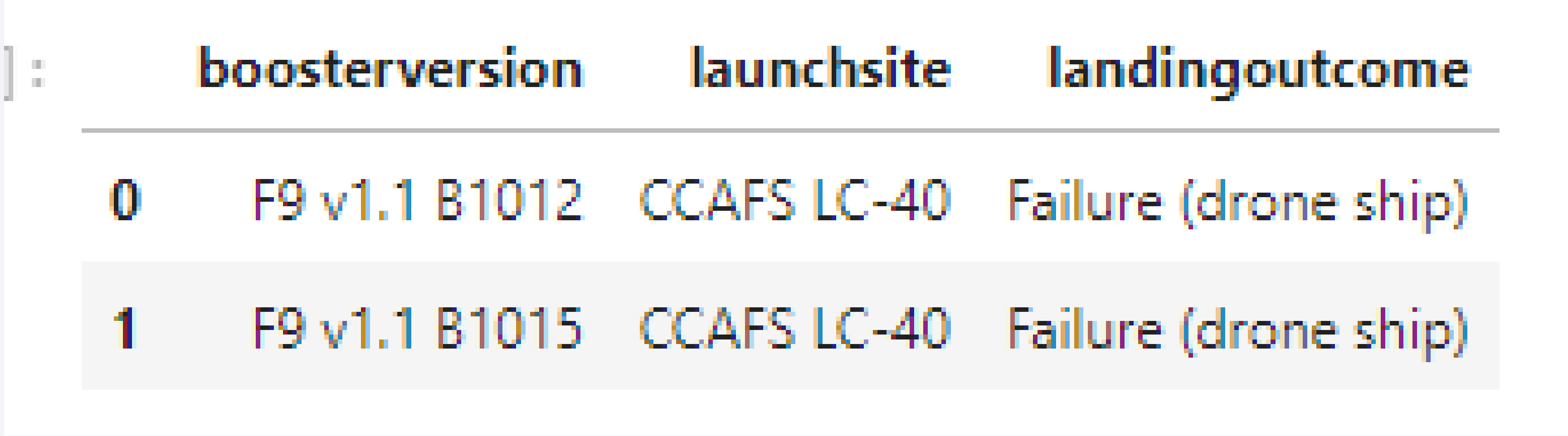
- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.
- All these boosters carried the maximum payload



	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

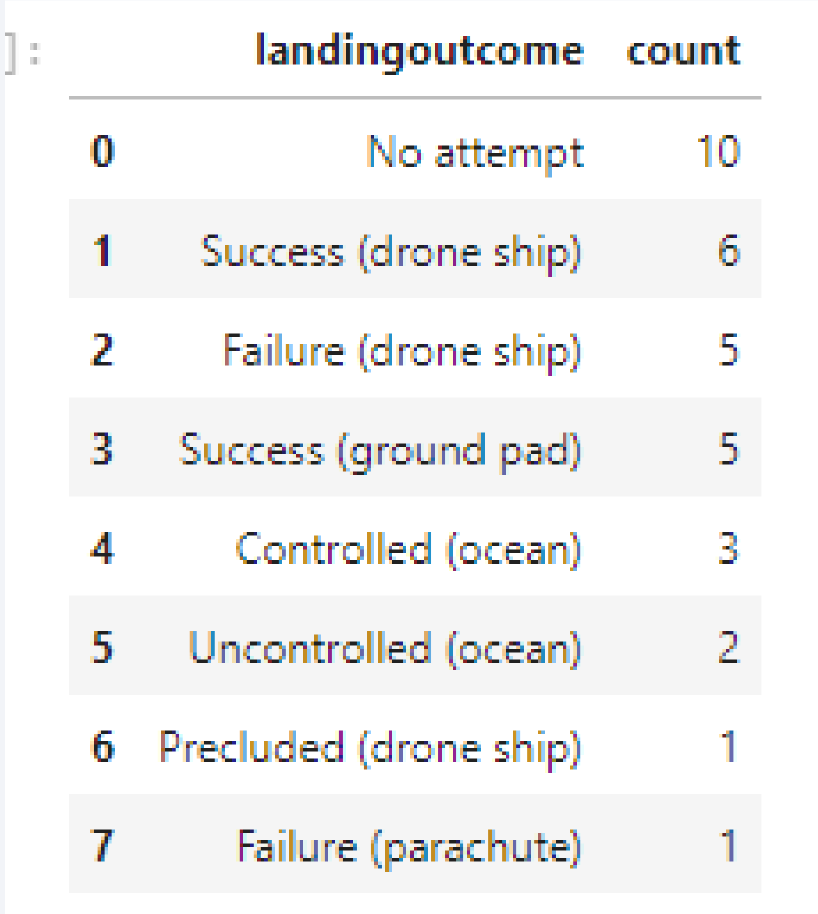
- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015



	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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.



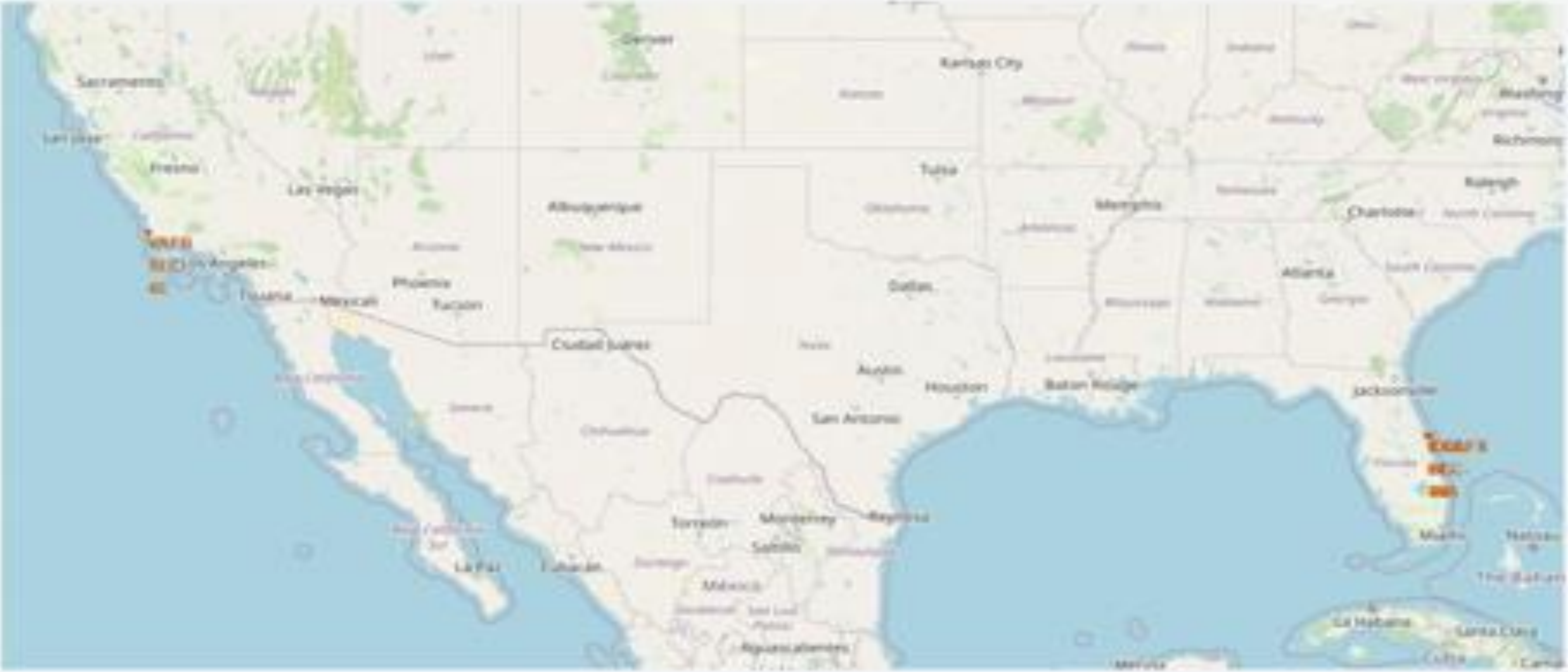
]:

	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

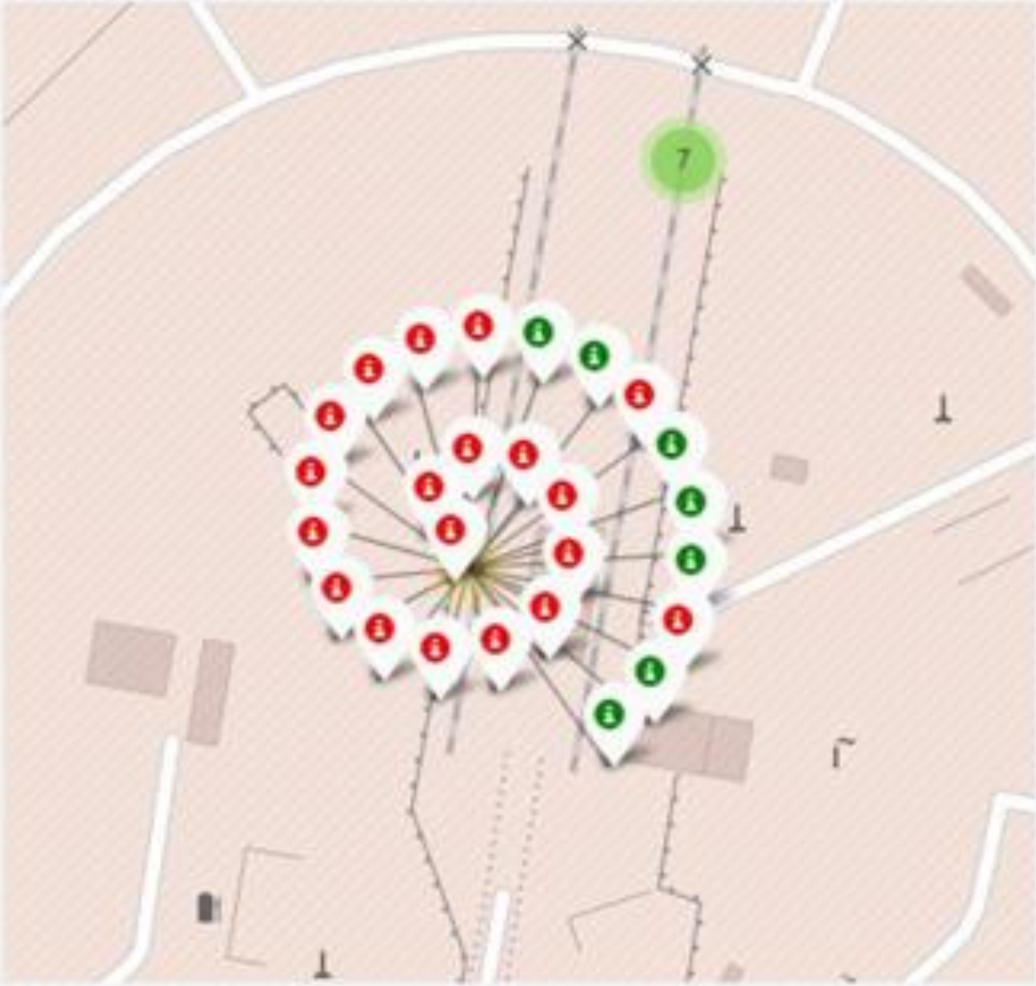
Launch Sites Proximities Analysis

Launch Sites That are Close to the Coast



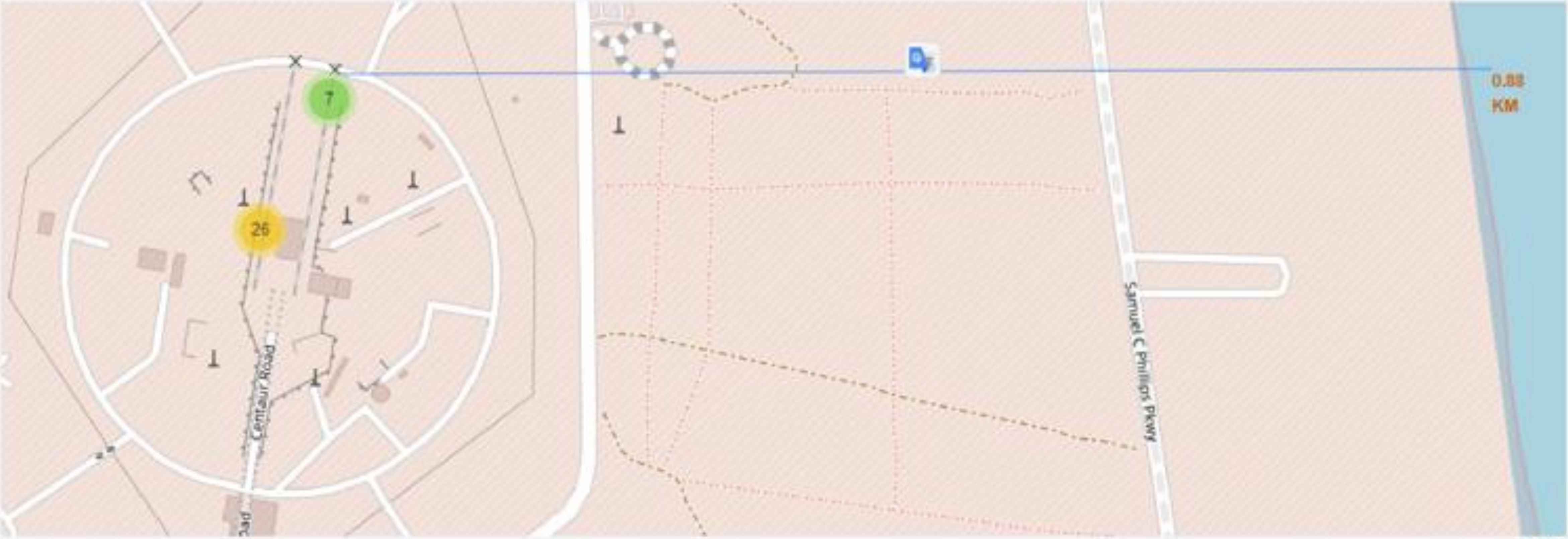
Launch Fails and Successes Per Site

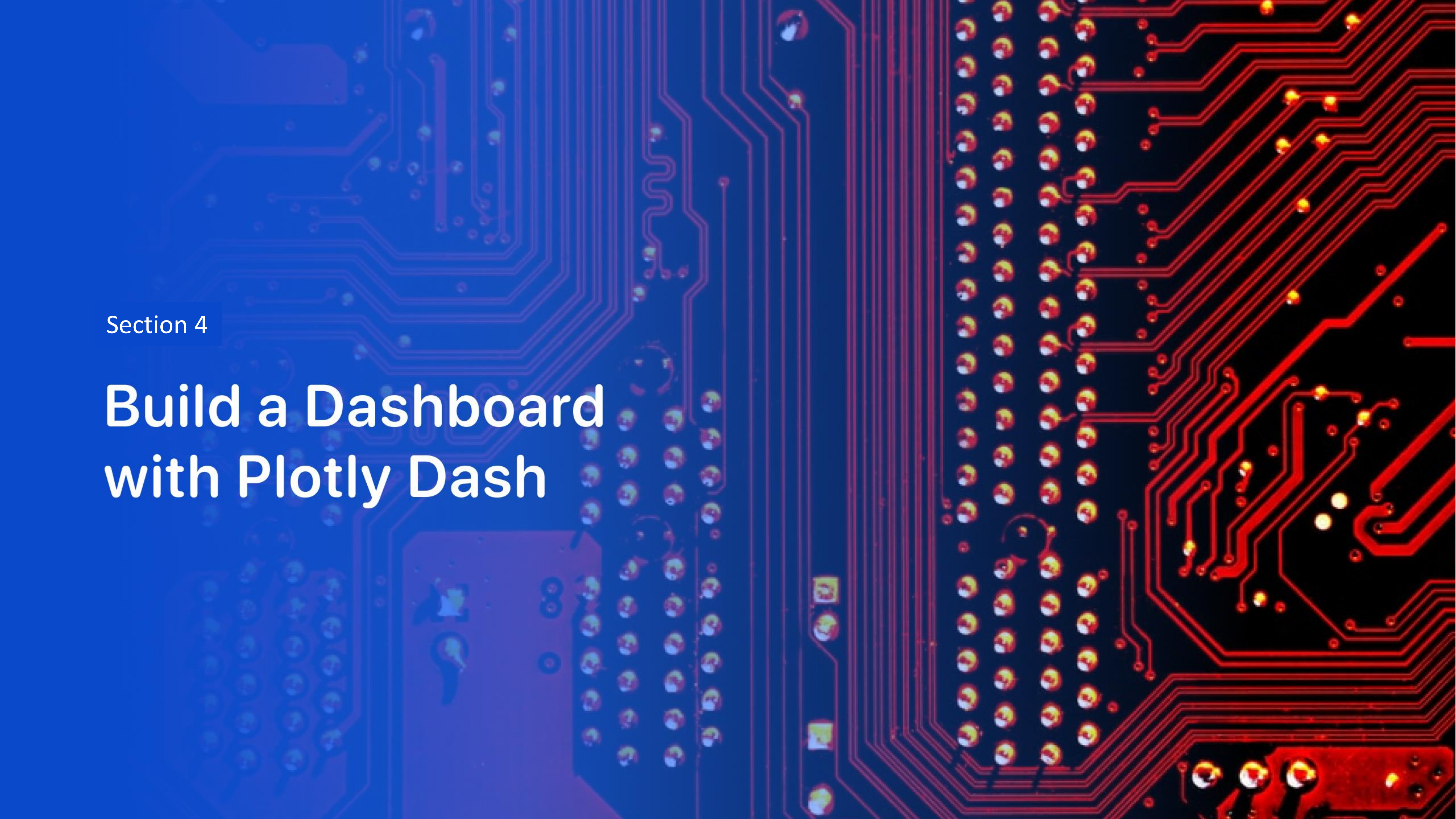
- 26 releases in this location, the red representing failed and the green representing successful



Distance Between Launch Sites and its Proximities

- Distance between this launch site and the coast is about 0.88 km

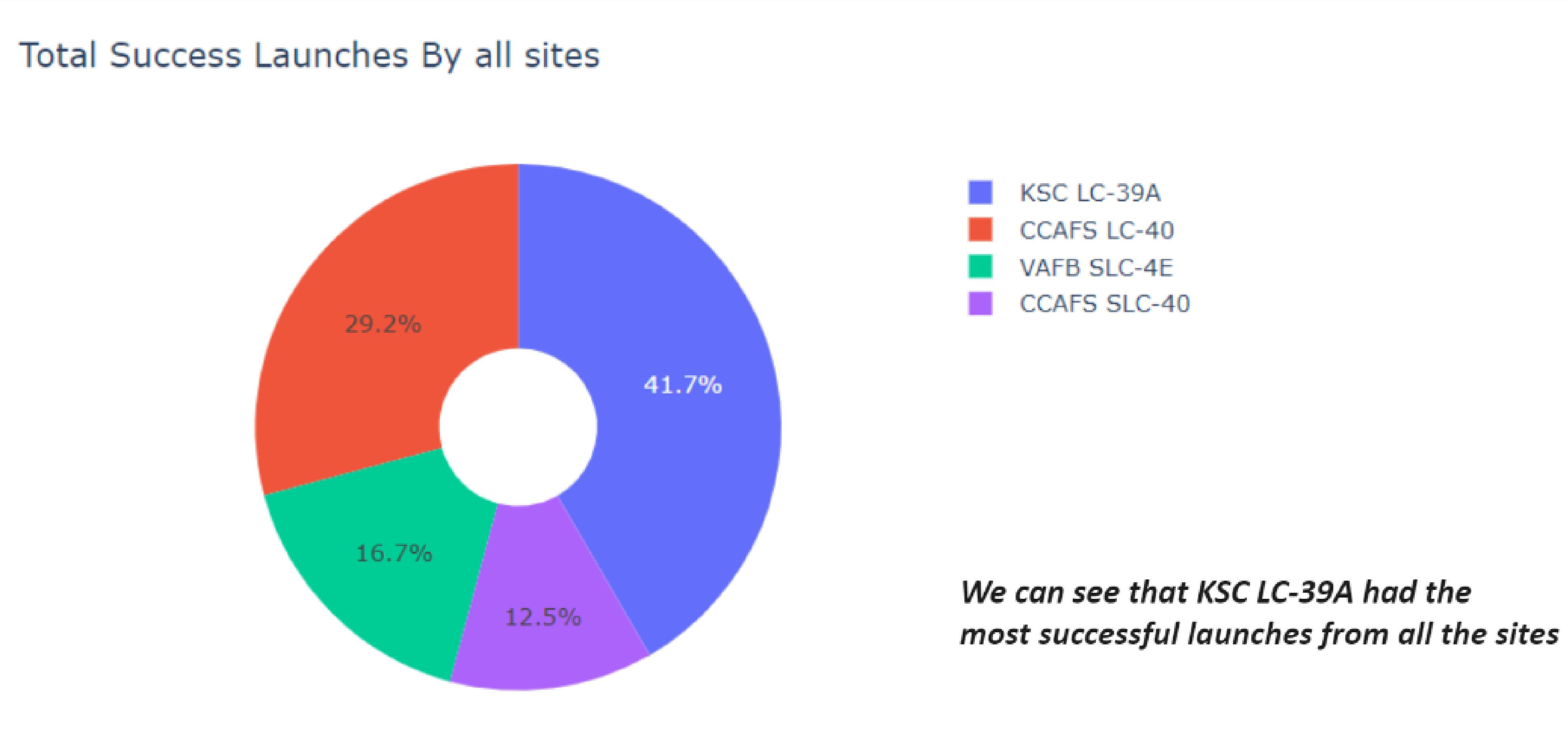




Section 4

Build a Dashboard with Plotly Dash

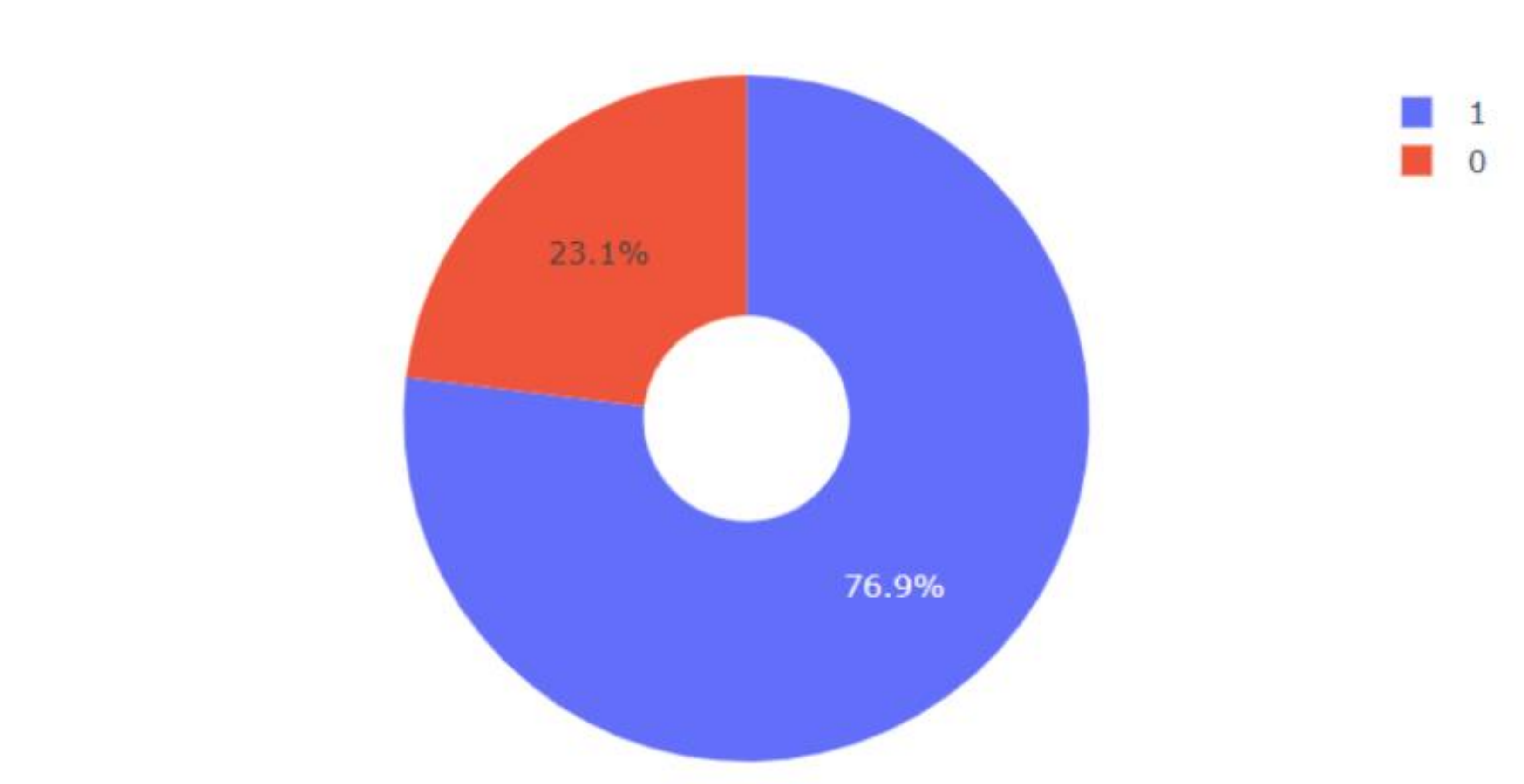
Success Rate per Launch Site



Total Success Launches By all sites

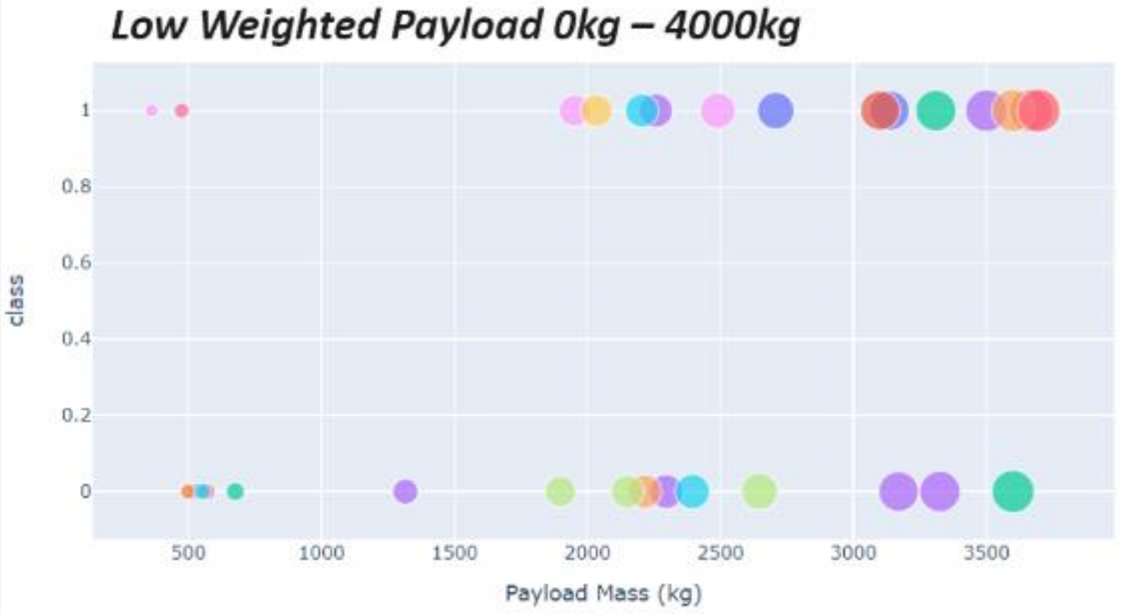
We can see that KSC LC-39A had the most successful launches from all the sites

The Launch Site with the Highest Success/Fail Ratio

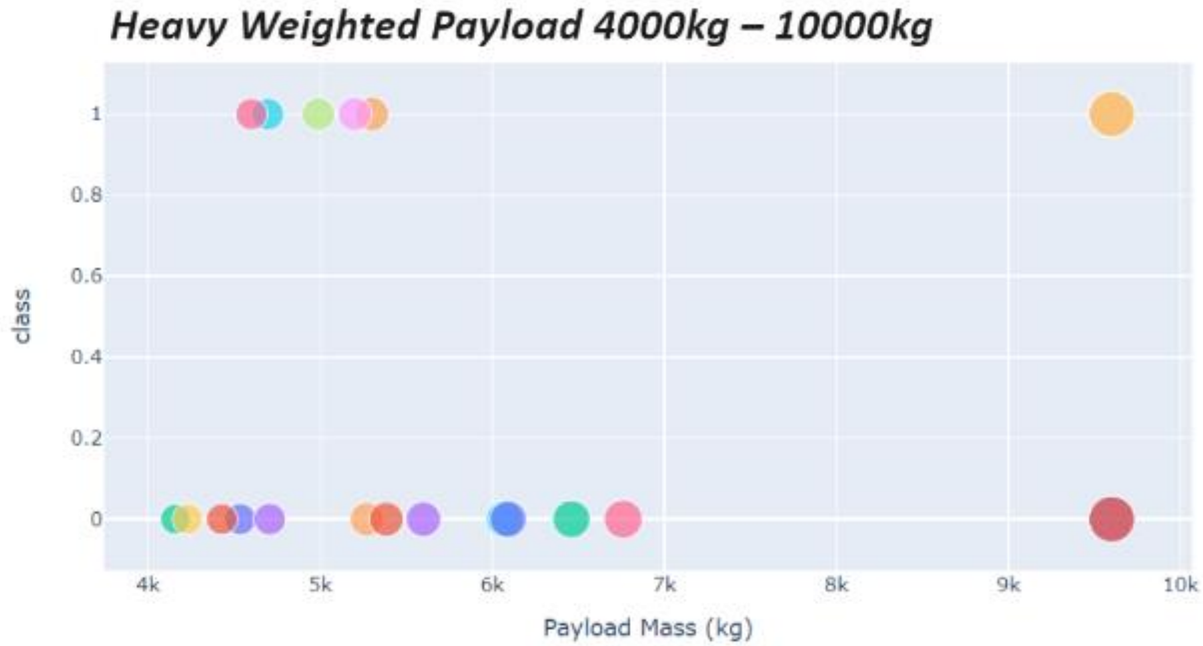


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome of sites



Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg

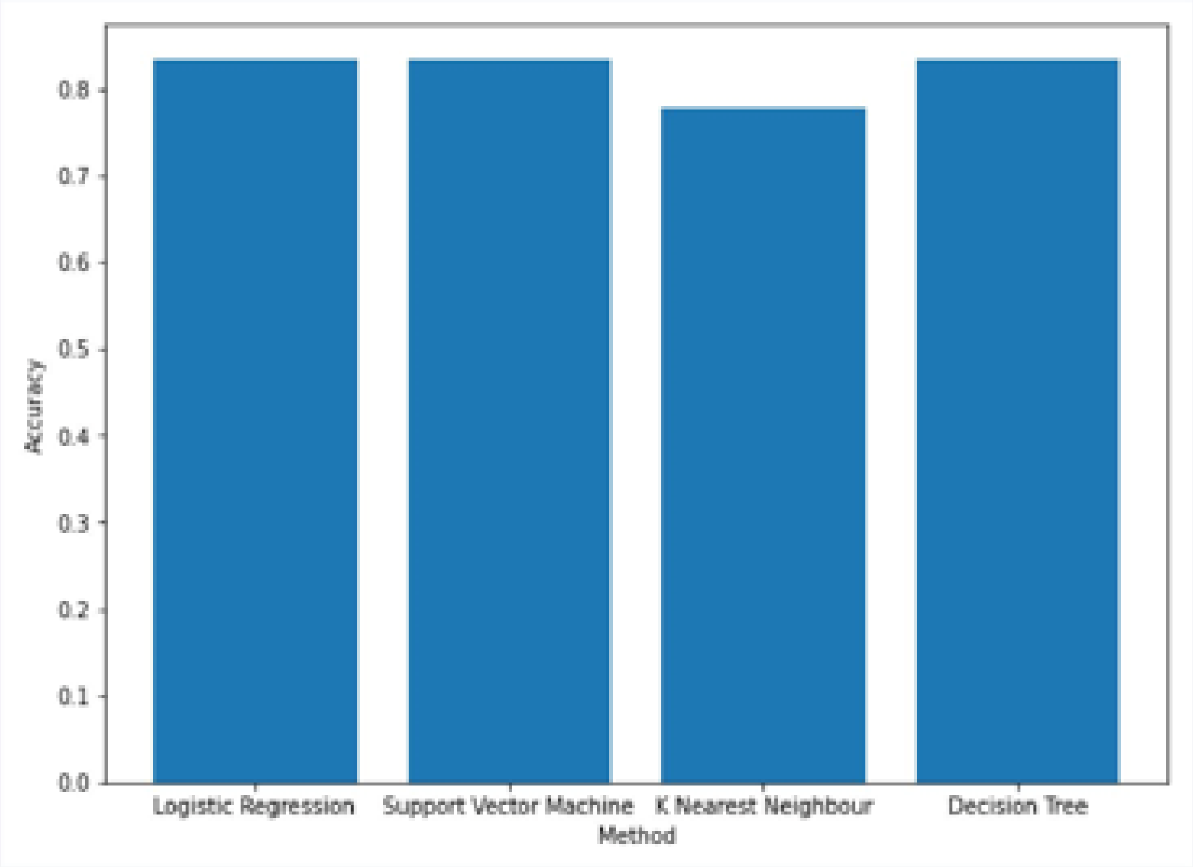
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

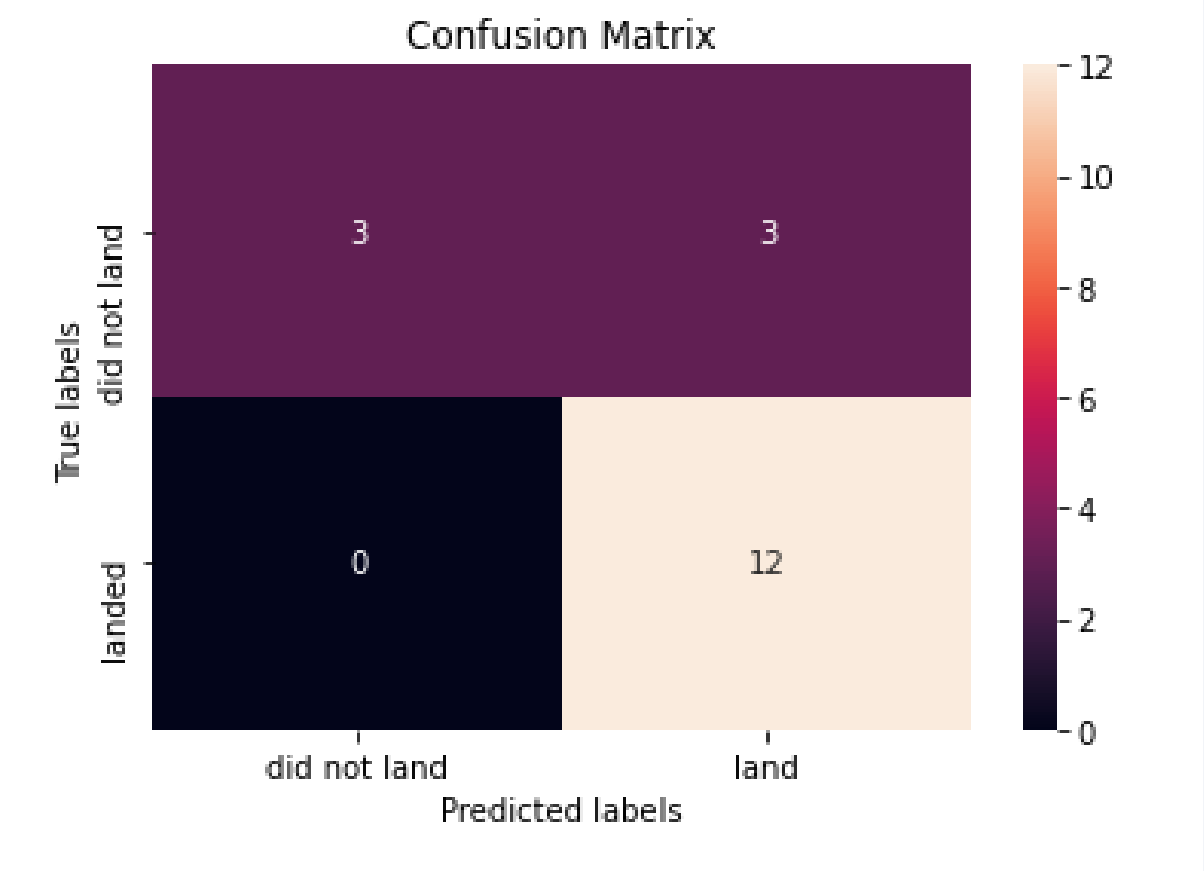
Classification Accuracy

- LR, SVM, and DT all have 83%



Confusion Matrix

- Only 3 failed attempts, high accuracy



Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!