IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aisyah Abdullah
10th April 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through APIData
  - Collection with Web ScrapingData
  - WranglingExploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Background:
  - To explore commercial space exploration
  - Space X has best pricing ($62 million vs. $165 million USD)
  - Largely due to ability to recover part of rocket (Stage 1)
  - Space Y becoming a new competitor to Space X
  - To create a machine learning pipeline to predict if the first stage will land successfully.
- Problem:
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions need to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

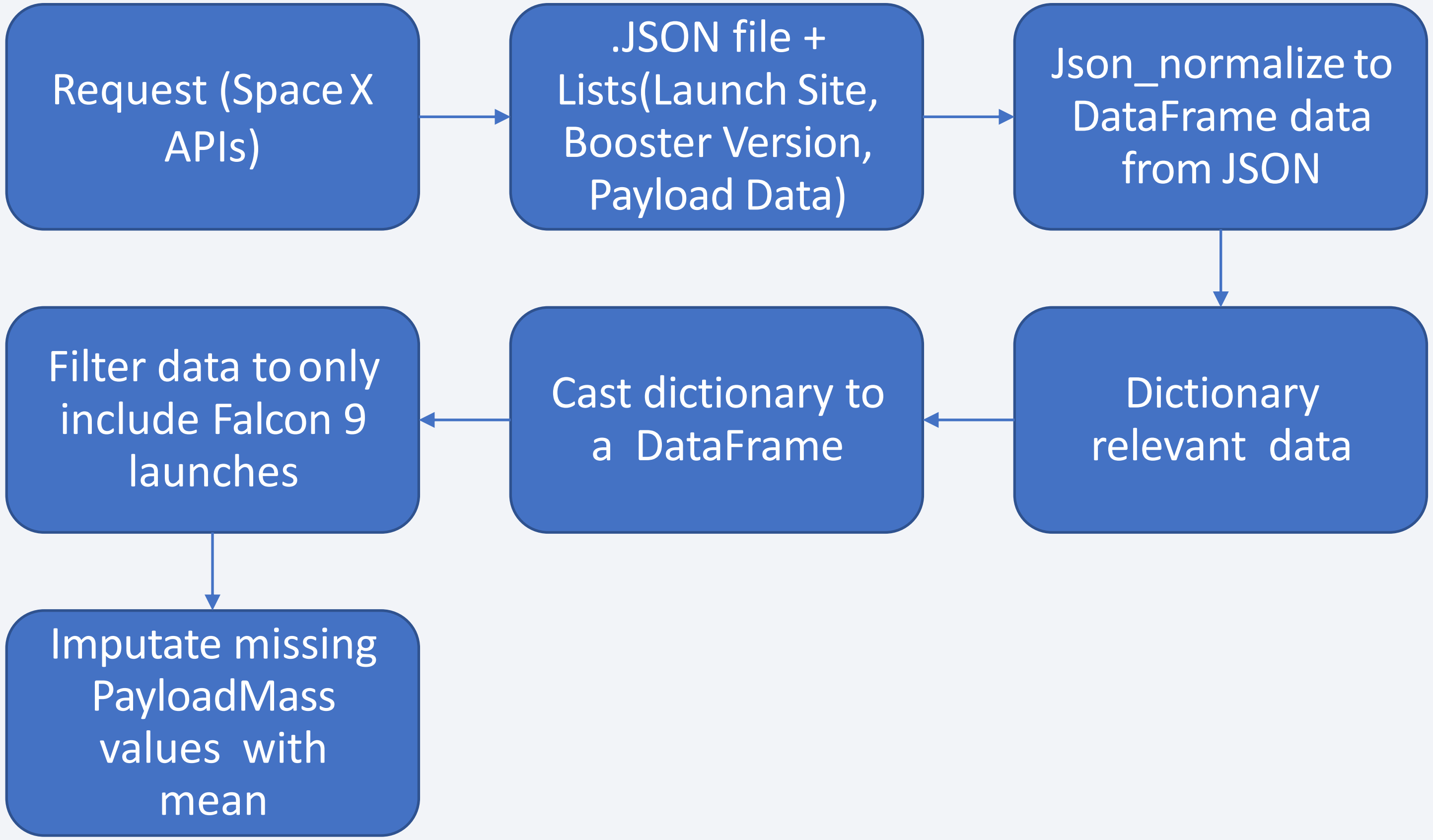
- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- 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

- GitHub URL: https://gist.github.com/cekelathersheys/b549ec2f1cae0e433ab98a77538c9abd



Request (Space X APIs)

.JSON file + Lists(Launch Site, Booster Version, Payload Data)

Json_normalize to DataFrame data from JSON

Dictionary relevant data

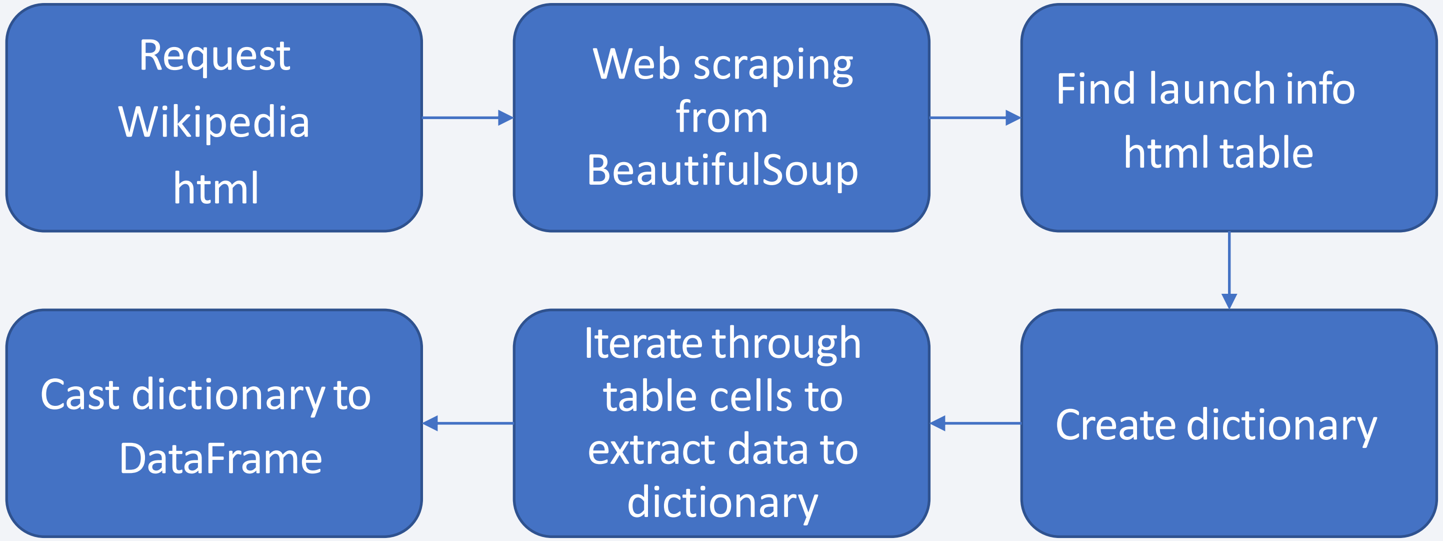
Cast dictionary to a DataFrame

Filter data to only include Falcon 9 launches

Imputate missing PayloadMass values with mean

Data Collection - Scraping

- GitHub URL: https://gist.github.com/cekelathersheys/b33d5b8e492f35b21f69c93c679c273b



Request Wikipedia html

Web scraping from BeautifulSoup

Find launch info html table

Create dictionary

Iterate through table cells to extract data to dictionary

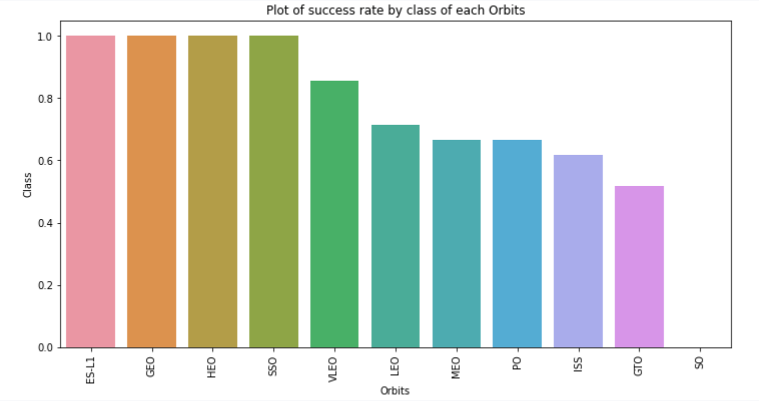
Cast dictionary to DataFrame

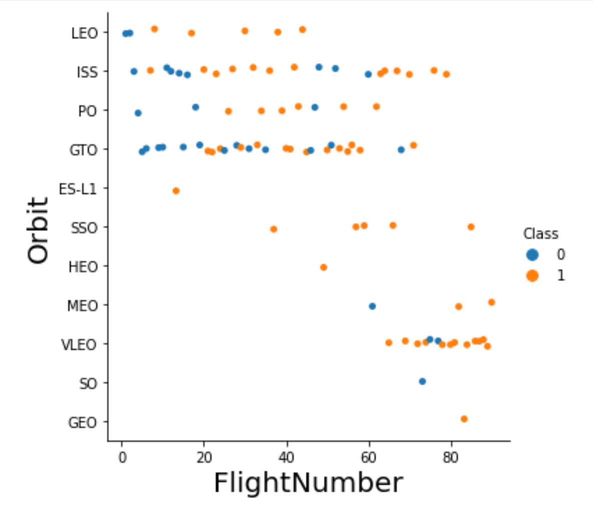
Data Wrangling

- Performed exploratory data analysis and determined the training labels.
- The number of launches are calculated at each site, and the number and occurrence of each orbit
- Landing outcome label created from outcome column and exported the results to csv.
- GitHub Link: https://gist.github.com/cekelathersheys/c24d67315394ca4cc0fca4d7033e2e1e

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.
- GitHub Link: https://gist.github.com/cekelathershеys/f3a723f8d29a47f9e7baf25efdfa33e1





EDA with SQL

- The SpaceX dataset is loaded into a PostgreSQL database without leaving the jupyter notebook.
- The EDA with SQL is then applied 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 failed mission outcomes
    - The failed landing outcomes in drone ship, their booster version and launch site names.

Build an Interactive Map with Folium

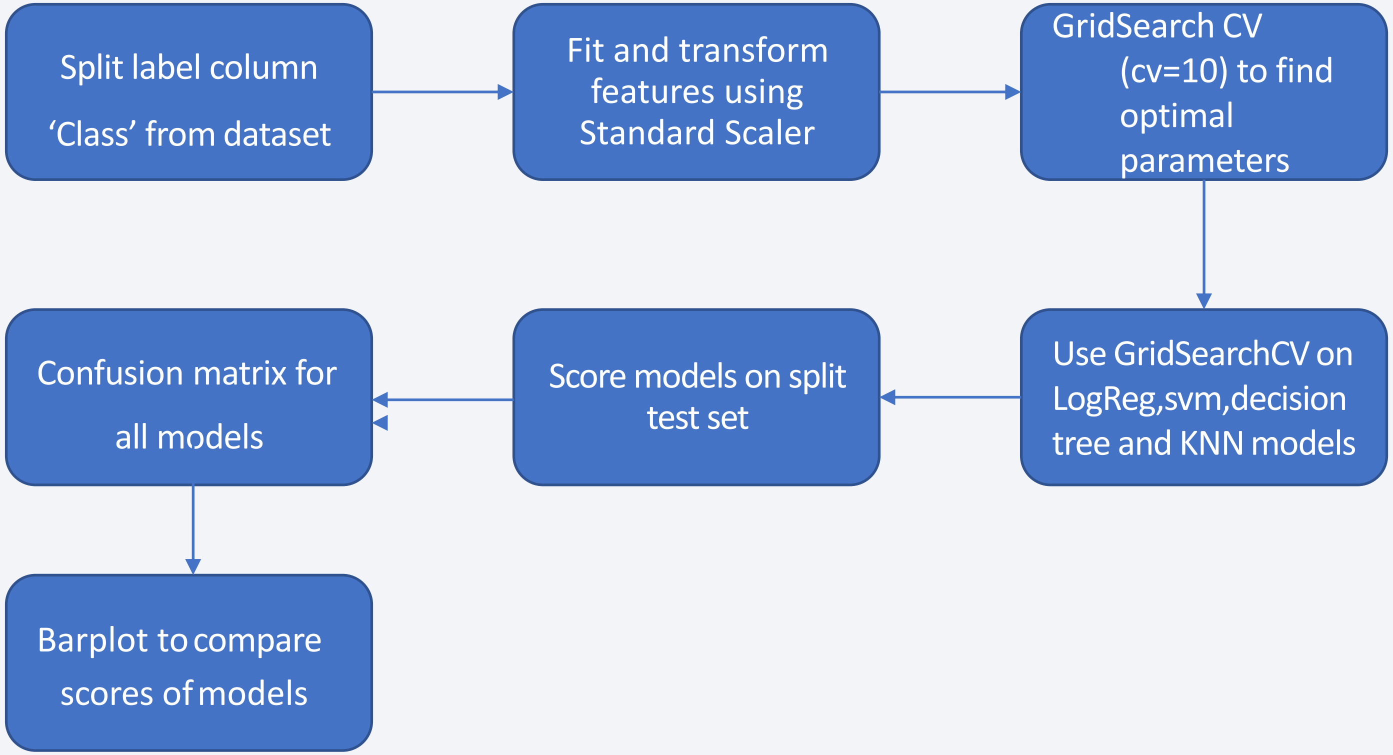
- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled 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.
- GitHub Link: https://gist.github.com/cekelathersheys/ade36d41fc21c788b28355ae0d530813

Build a Dashboard with Plotly Dash

- Dashboard includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.
- The pie chart is used to visualize launch site success rate.
- The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

Predictive Analysis (Classification)

- The data is loaded using numpy and pandas, transformed the data, and split our data into training and testing.
- Different machine learning models are built and tune for different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model and improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- GitHub Link: https://gist.github.com/cekelathersheys/3e9c398e3d23237591edd6b81fb0d765



Split label column ‘Class’ from dataset

Fit and transform features using Standard Scaler

GridSearch CV (cv=10) to find optimal parameters

Use GridSearchCV on LogReg,svm,decision tree and KNN models

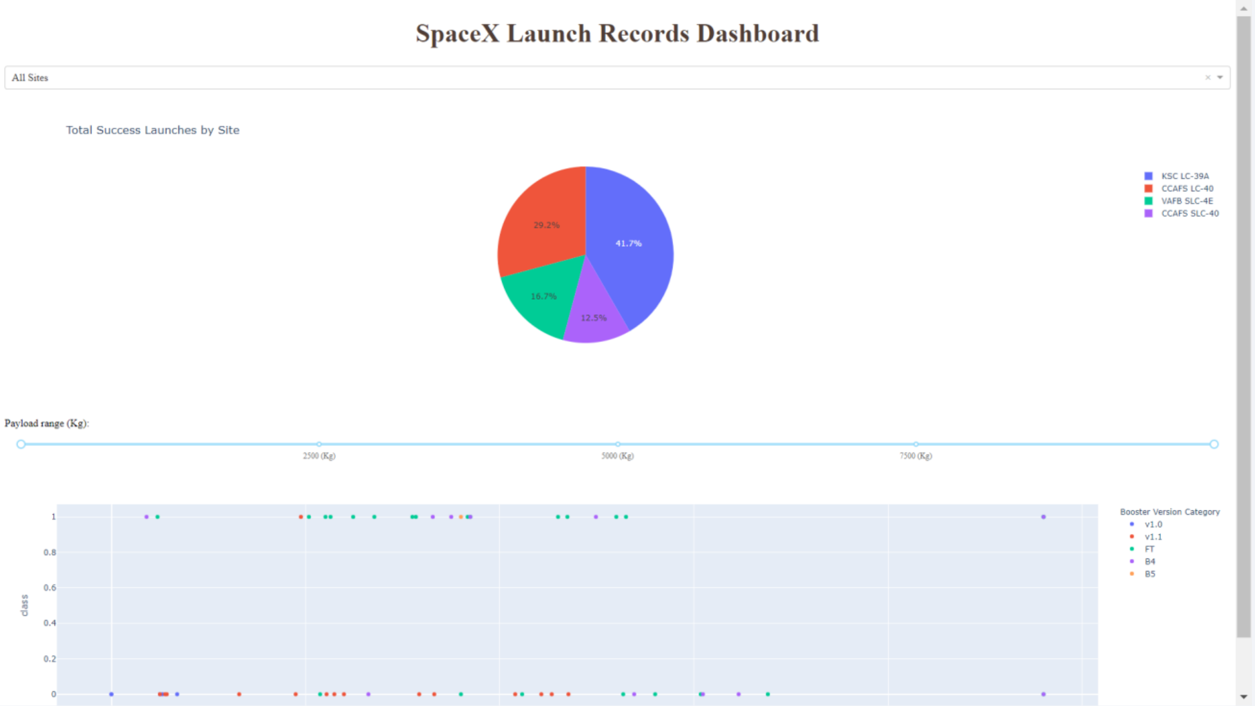
Score models on split test set

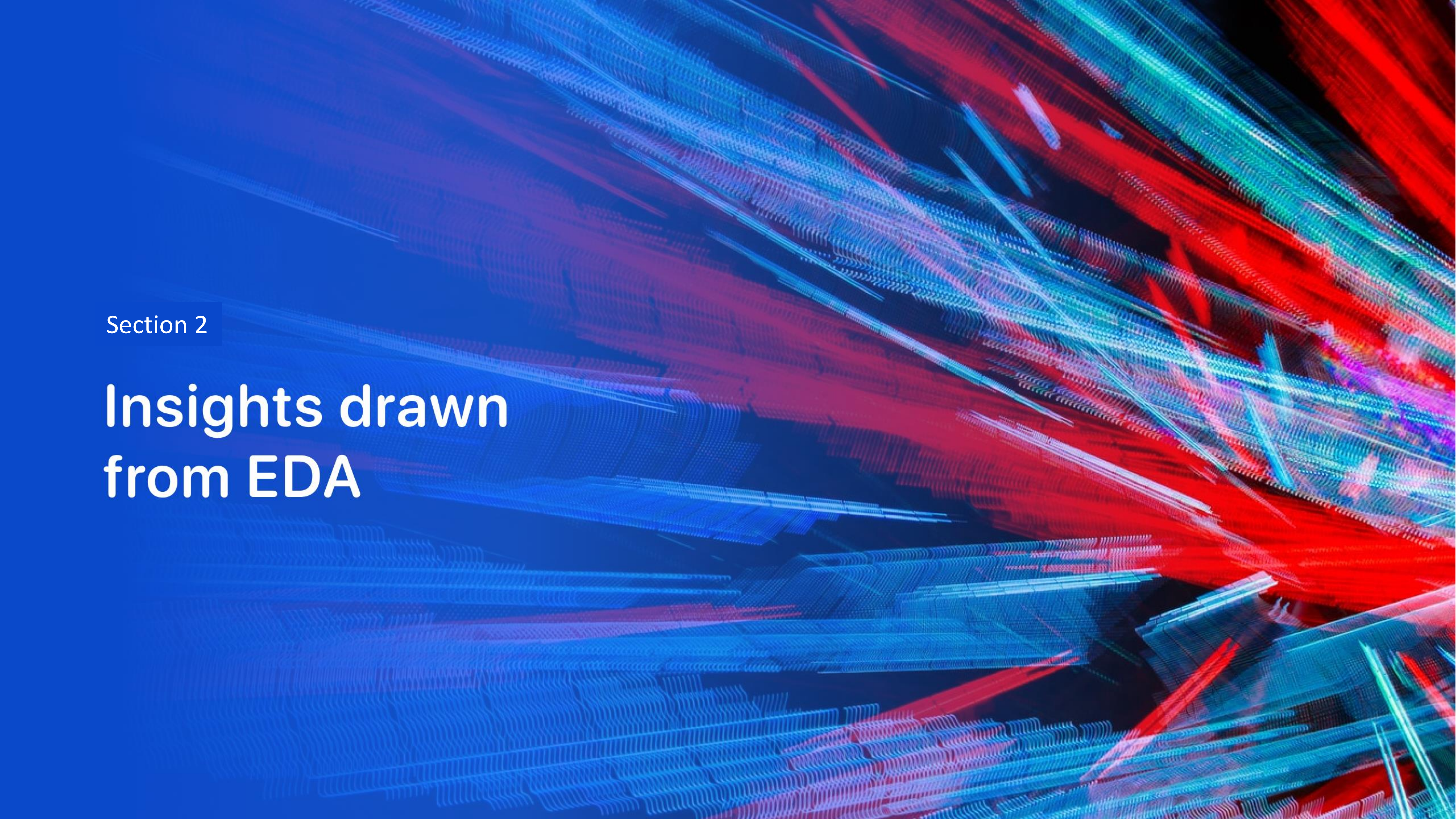
Confusion matrix for all models

Barplot to compare scores of models

Results

Based on the Plotly dashboard, the following sides will show the results of EDA with visualization, EDA with SQL, Interactive Map with Folium, and finally the results of our model with about 83% accuracy.



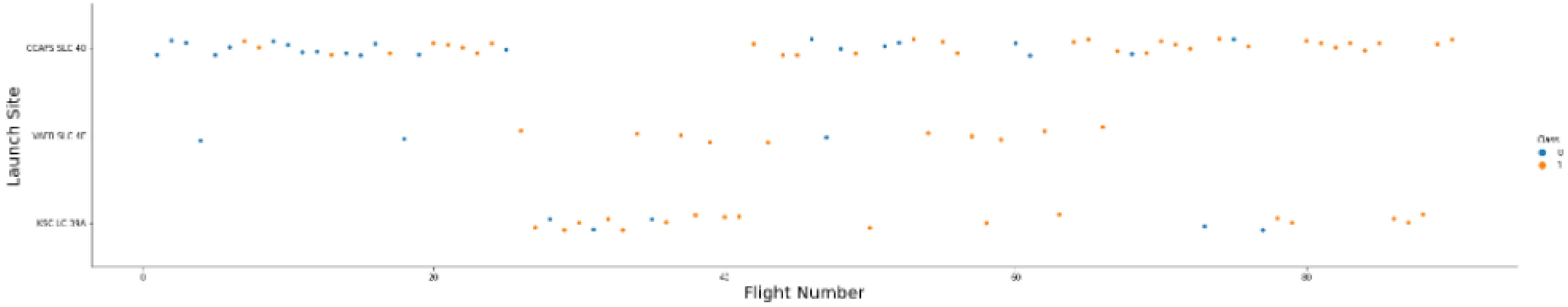


Section 2

Insights drawn from EDA

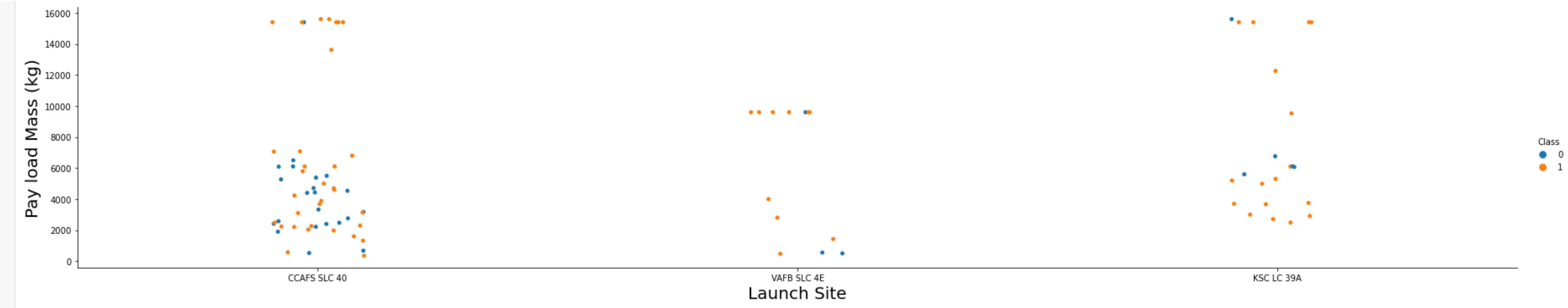
Flight Number vs. Launch Site

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



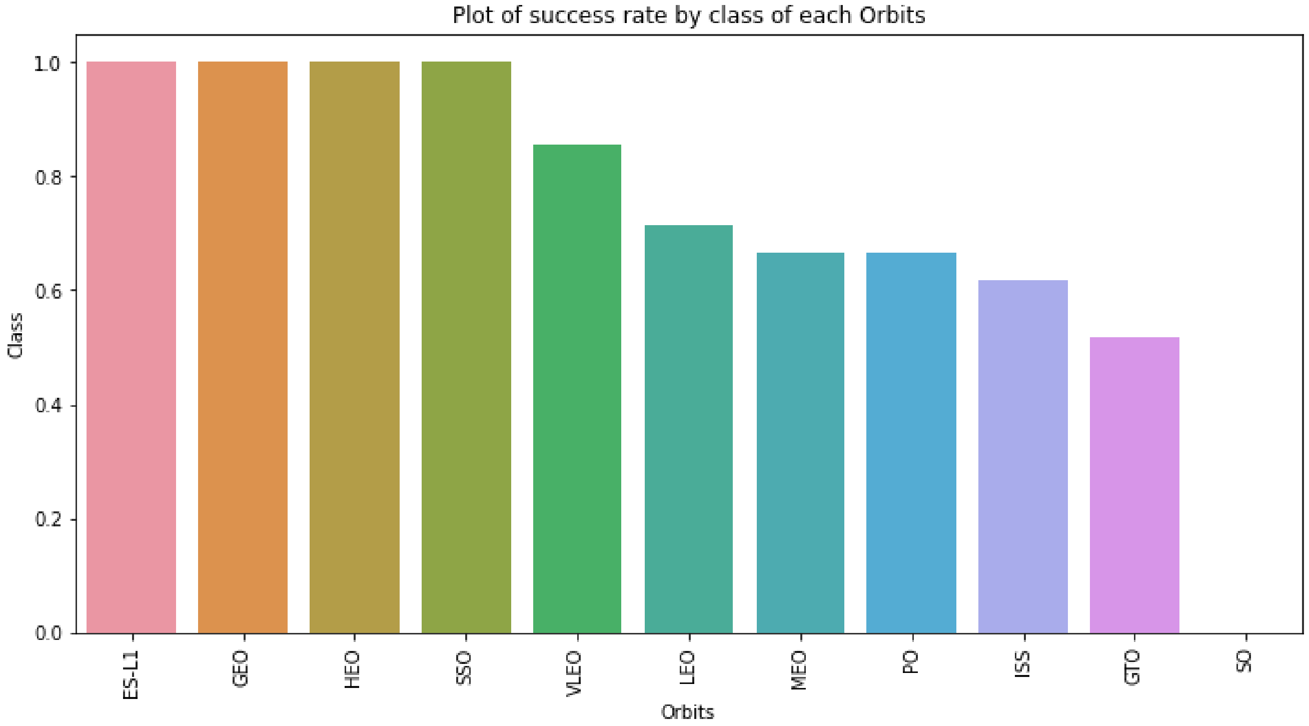
Payload vs. Launch Site

- Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.



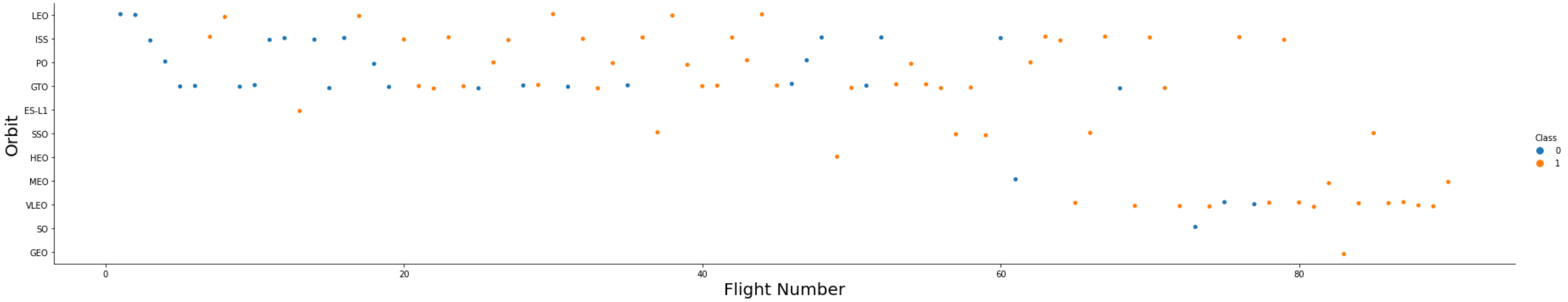
Success Rate vs. Orbit Type

- ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis) SSO (5) has 100% success rate
- VLEO (14) has decent success rate and attempts
- SO (1) has 0% success rate
- GTO (27) has the around 50% success rate but largest sample



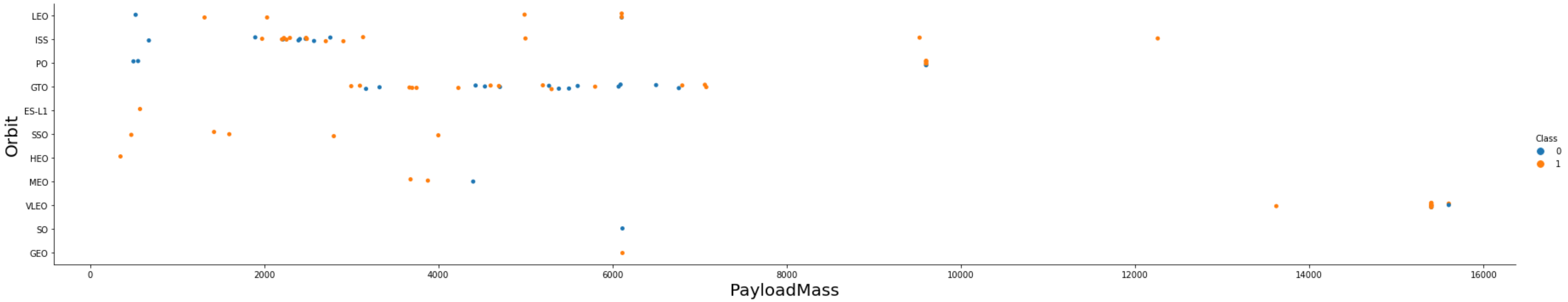
Flight Number vs. Orbit Type

- 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.



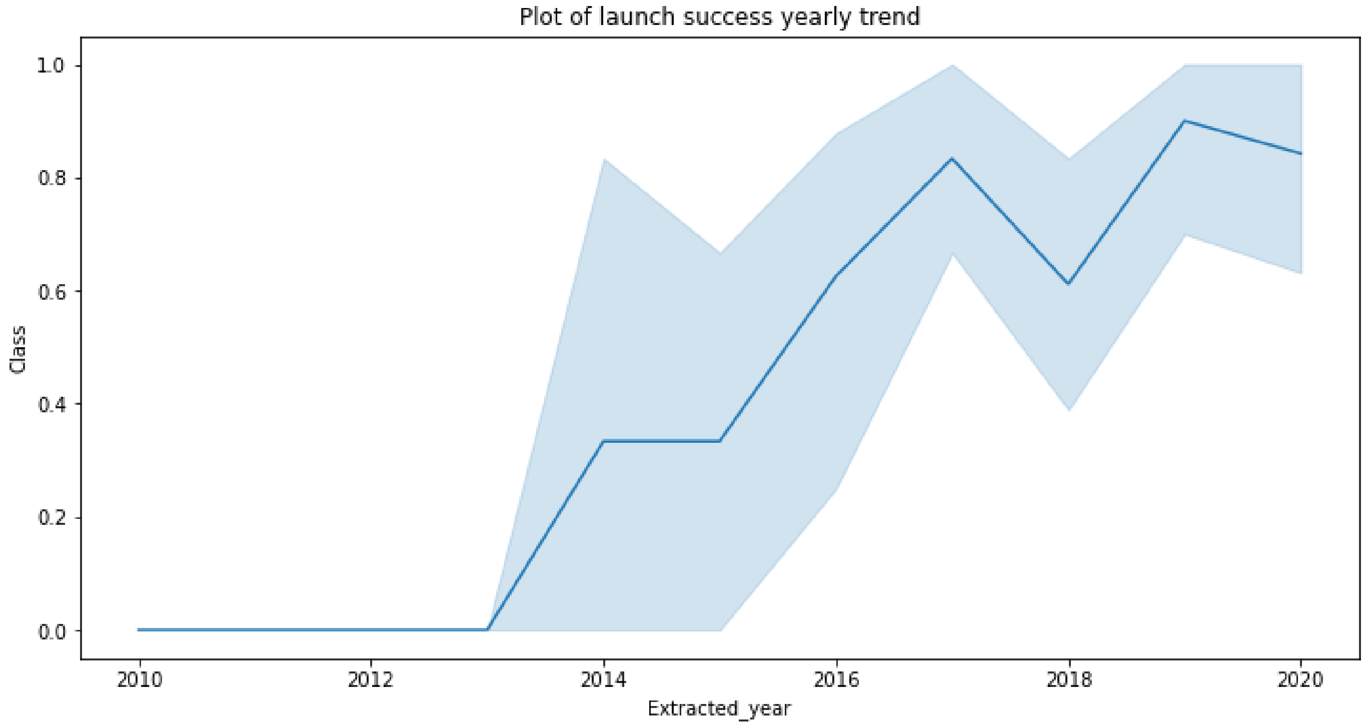
Payload vs. Orbit Type

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



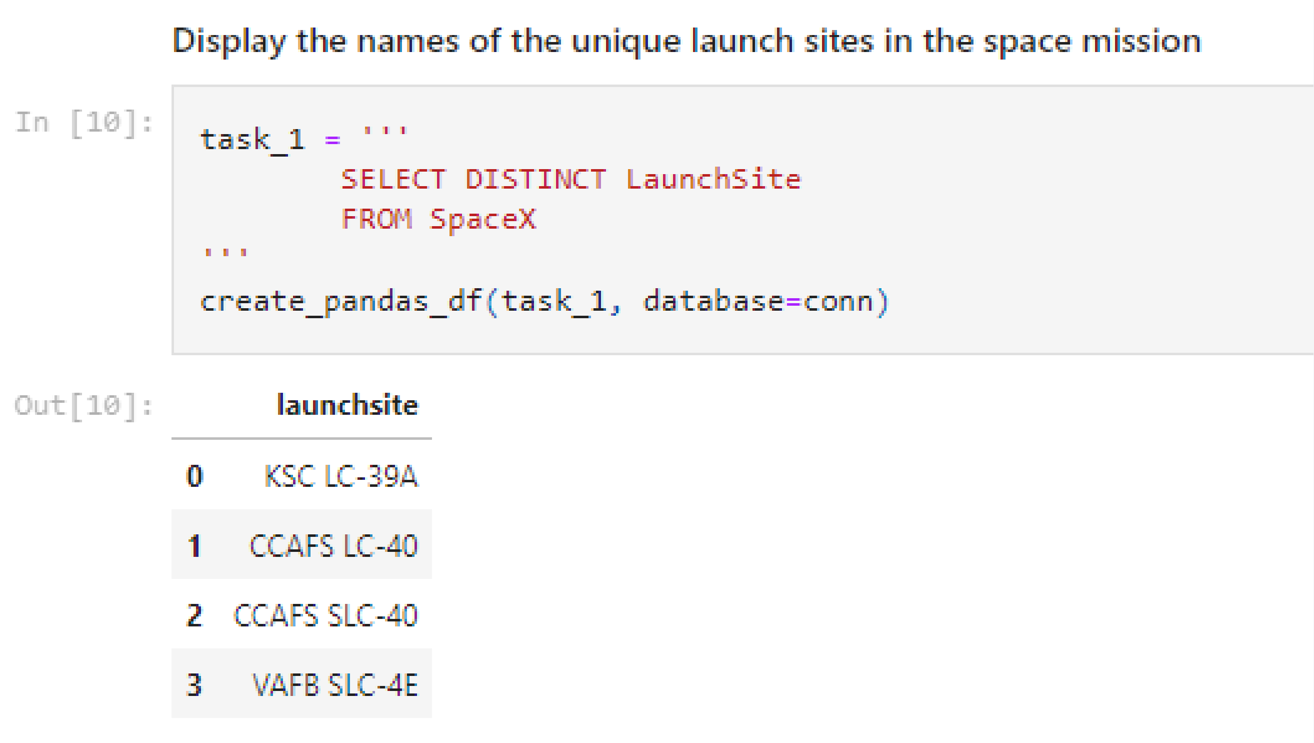
Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.



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

In [10]:

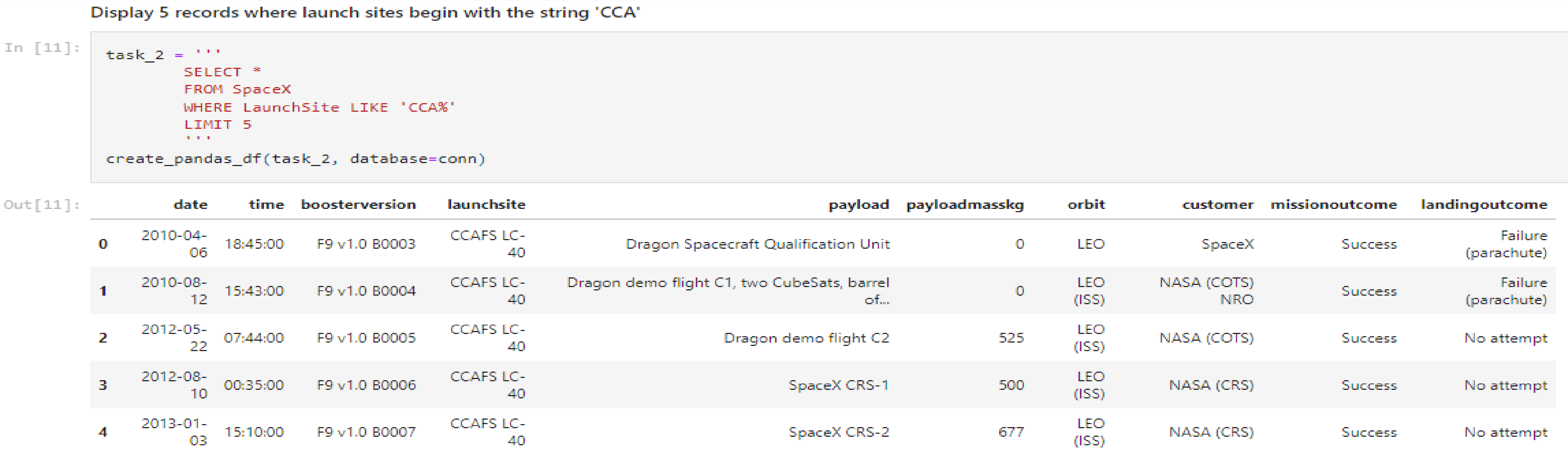
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

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'

- The query below is used to display 5 records where launch sites begin with `CCA`



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

In [11]:

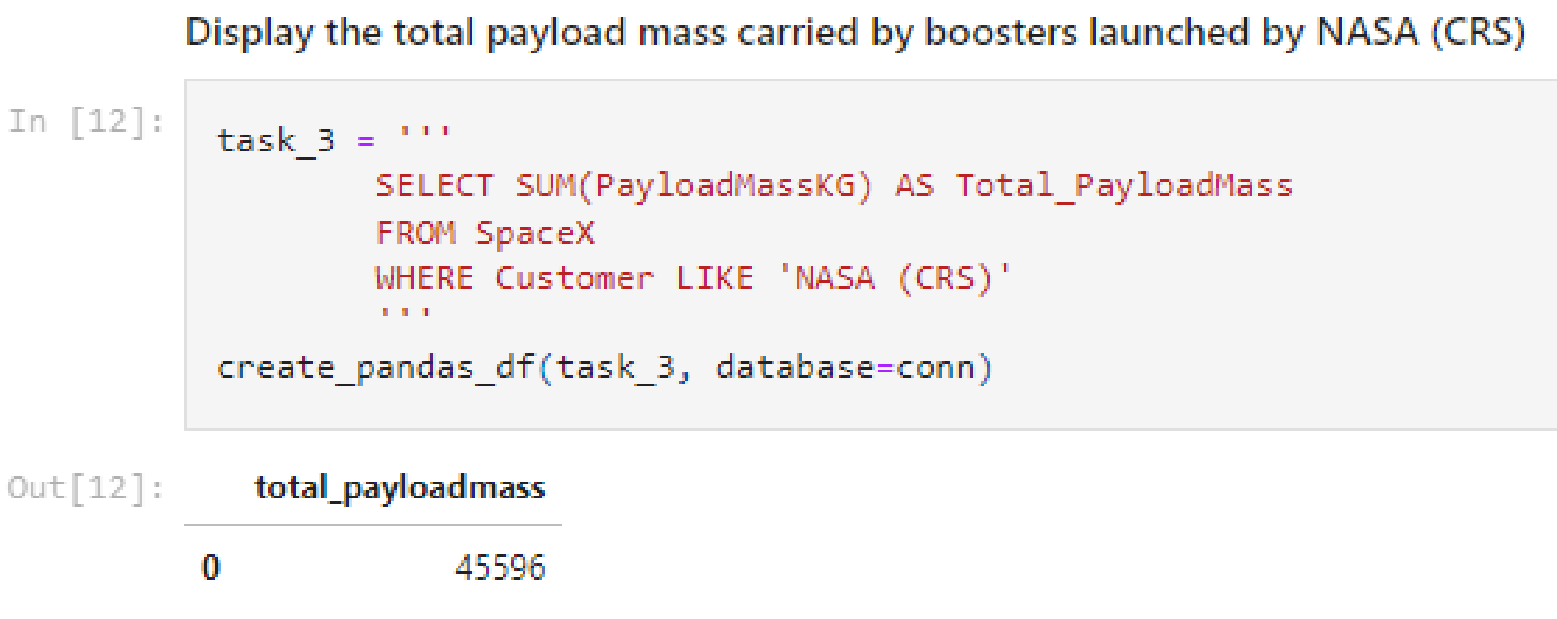
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

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

Total Payload Mass

- The total payload carried by boosters from NASA is calculated and the total payload mass is 45596 using the query below



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

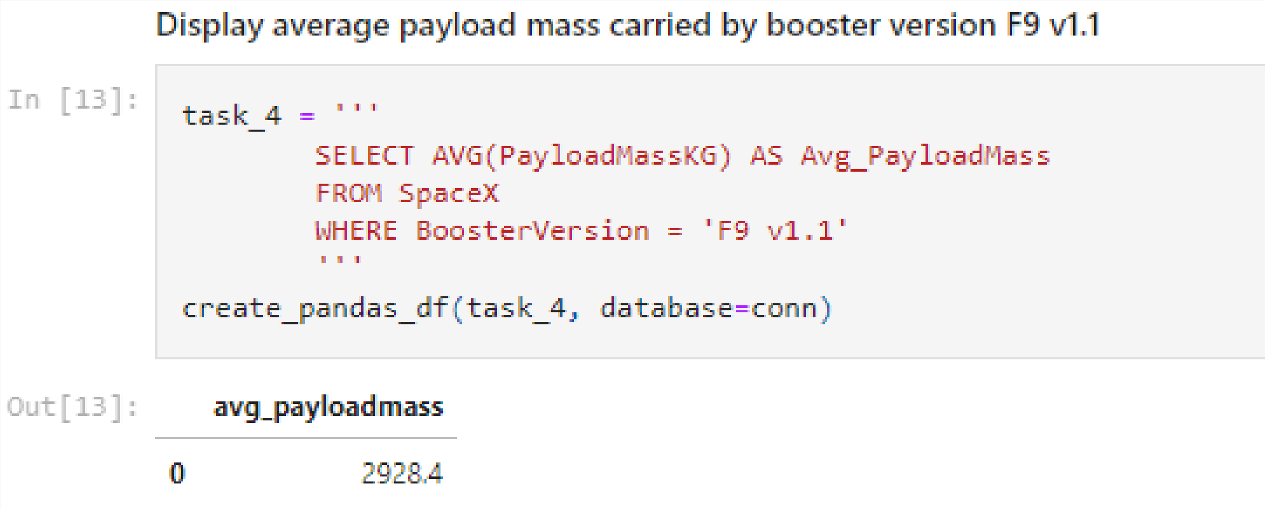
```
In [12]:
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is calculated and the result given as 2928.4



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

In [13]:

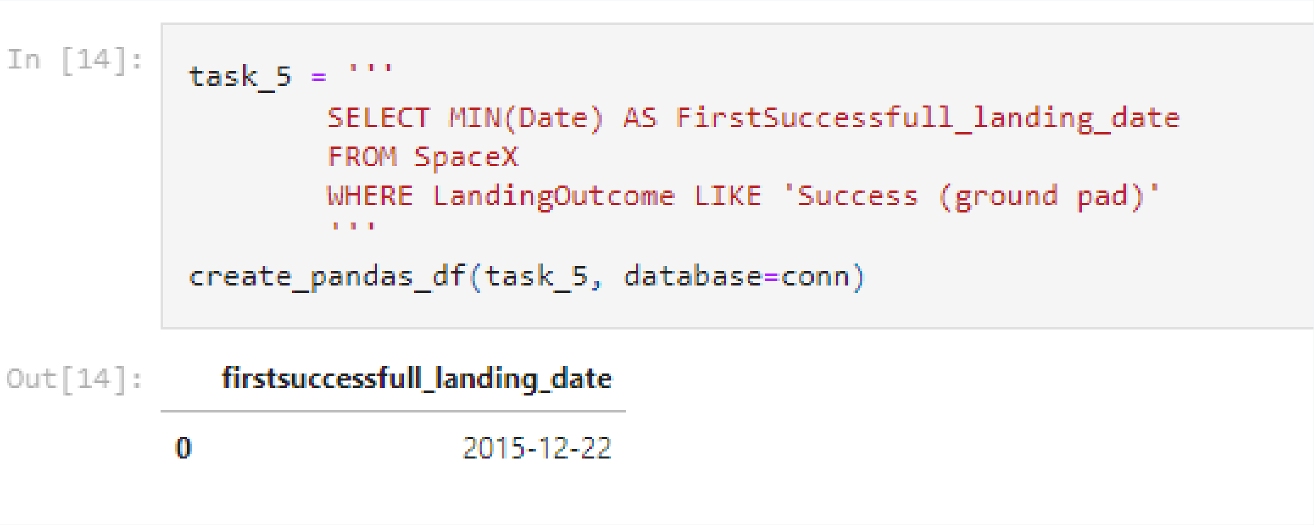
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- This query returns the first successful ground pad landing date.
- First ground pad landing wasn’t
- until the end of 2015.

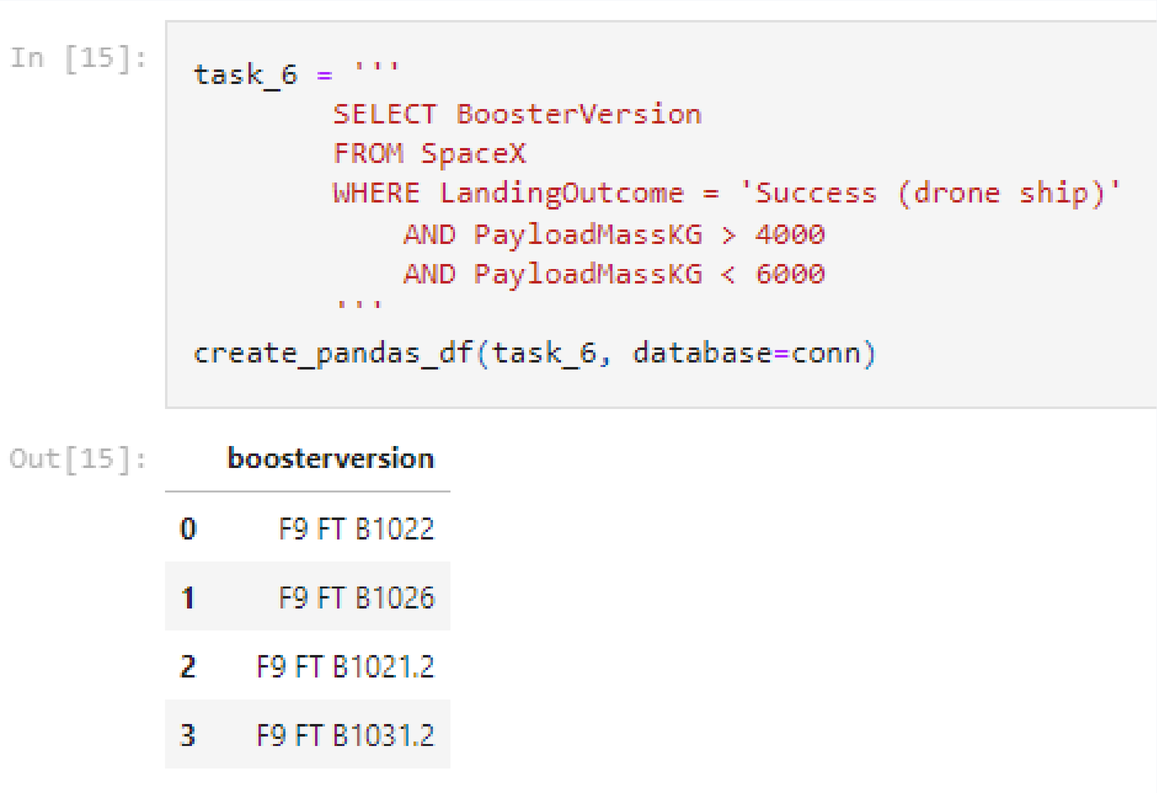


```
In [14]:  task_5 = '''
                  SELECT MIN(Date) AS FirstSuccessfull_landing_date
                  FROM SpaceX
                  WHERE LandingOutcome LIKE 'Success (ground pad)'
                  '''
          create_pandas_df(task_5, database=conn)
```

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively.



In [15]:

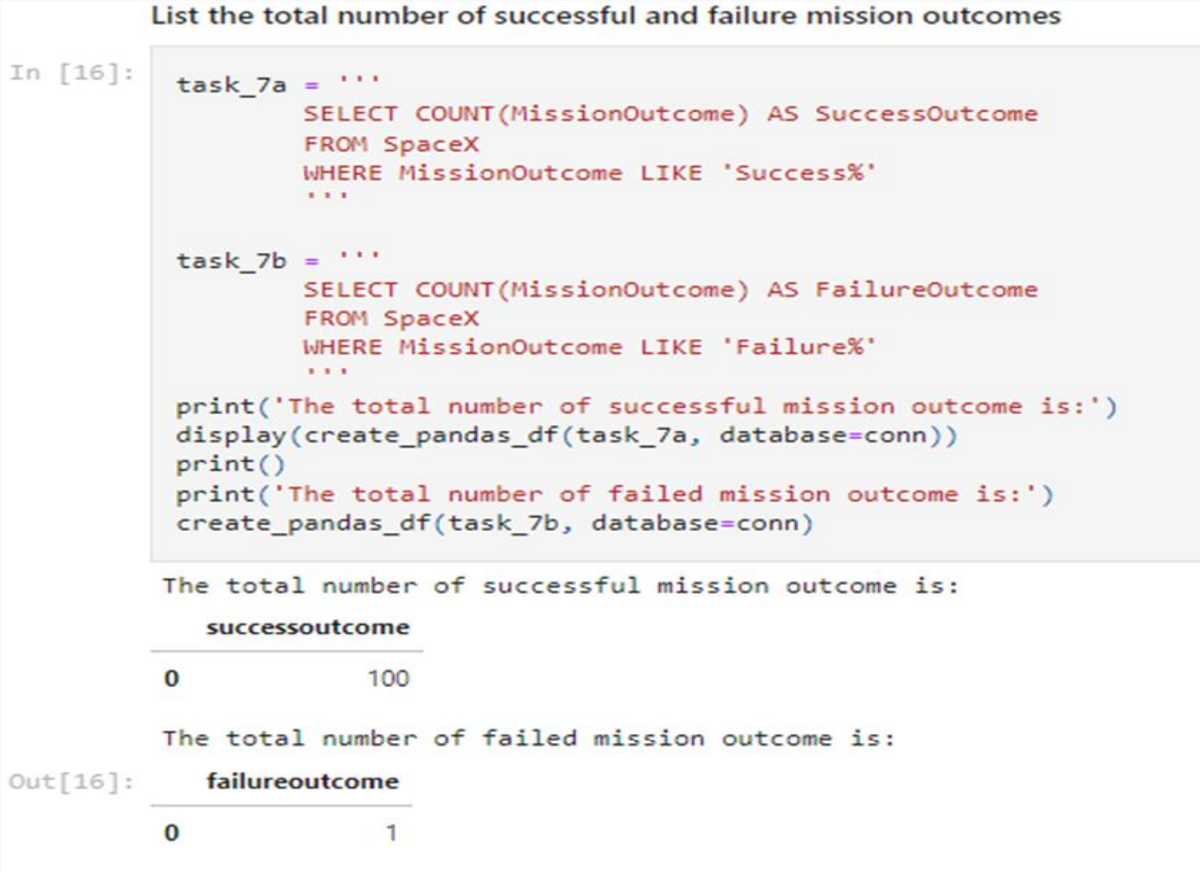
```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

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

- This query returns a count of each mission outcome.



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

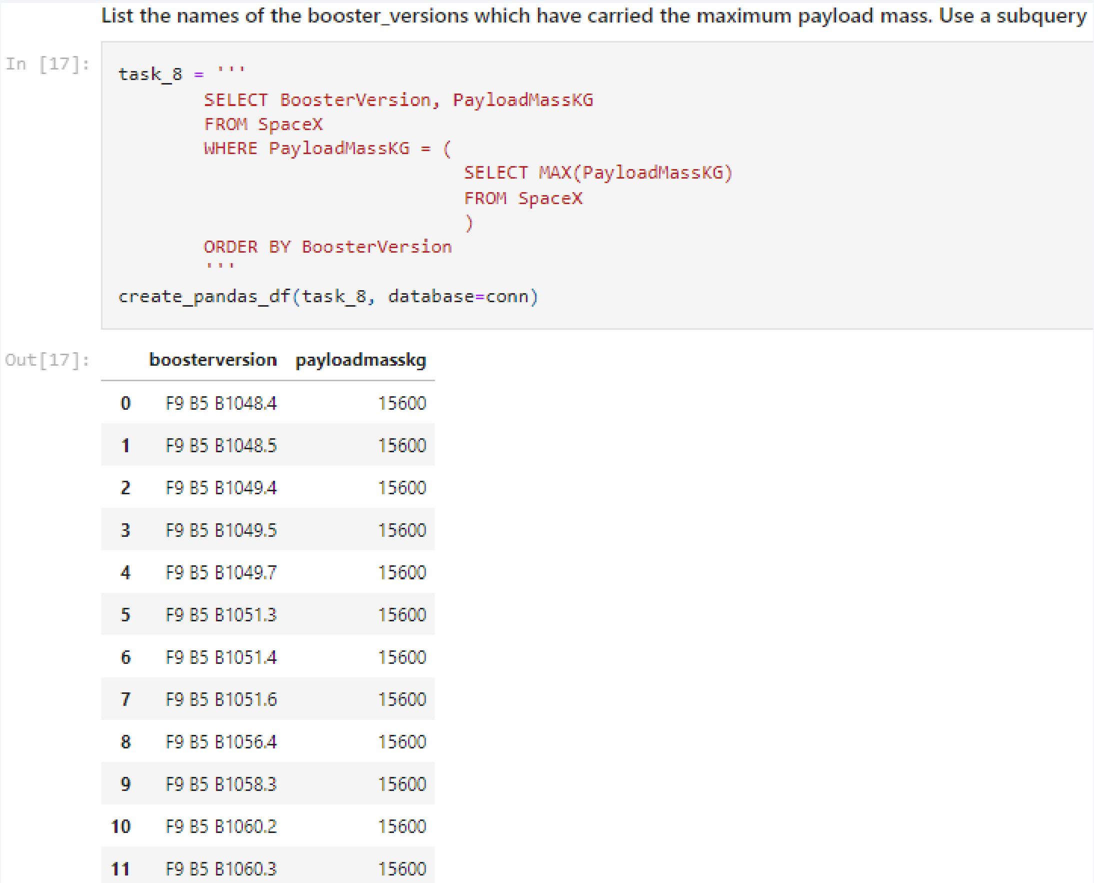
```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Boosters Carried Maximum Payload

- The booster that have carried the maximum payload is determined using a subquery in the **WHERE** clause and the **MAX()** function.



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

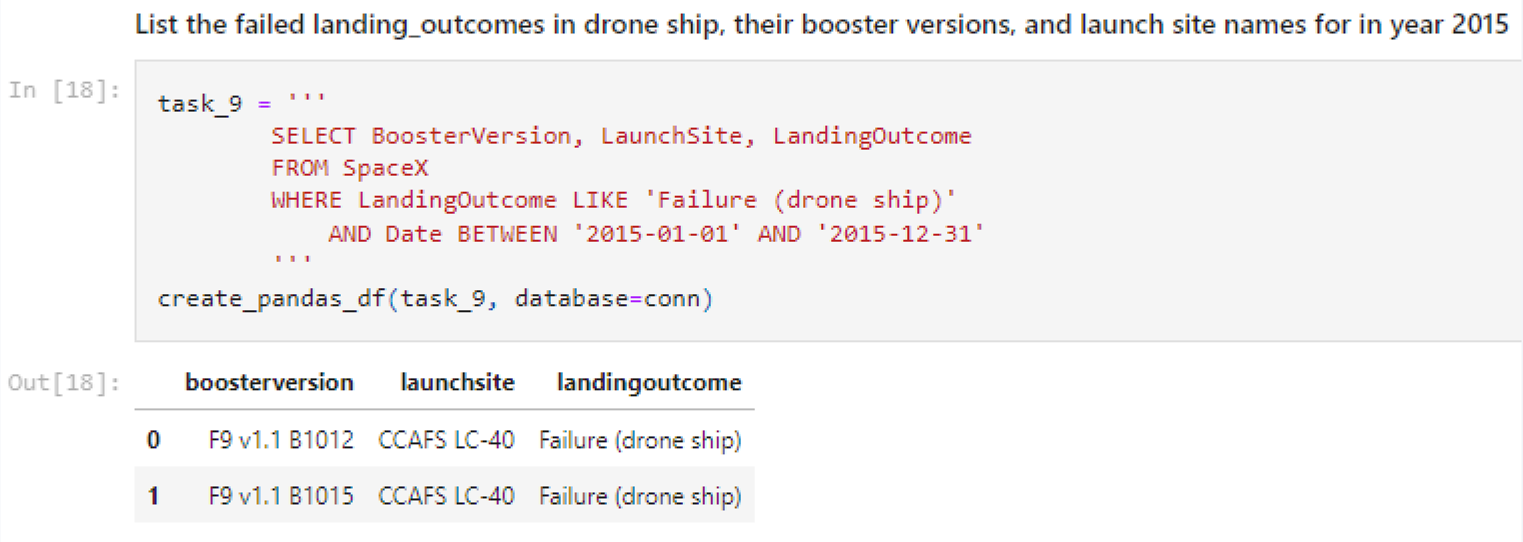
```
In [17]:  task_8 = '''
                  SELECT BoosterVersion, PayloadMassKG
                  FROM SpaceX
                  WHERE PayloadMassKG = (
                                          SELECT MAX(PayloadMassKG)
                                          FROM SpaceX
                                          )
                  ORDER BY BoosterVersion
                  '''
          create_pandas_df(task_8, database=conn)
```

Out[17]:

	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

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



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

In [18]:

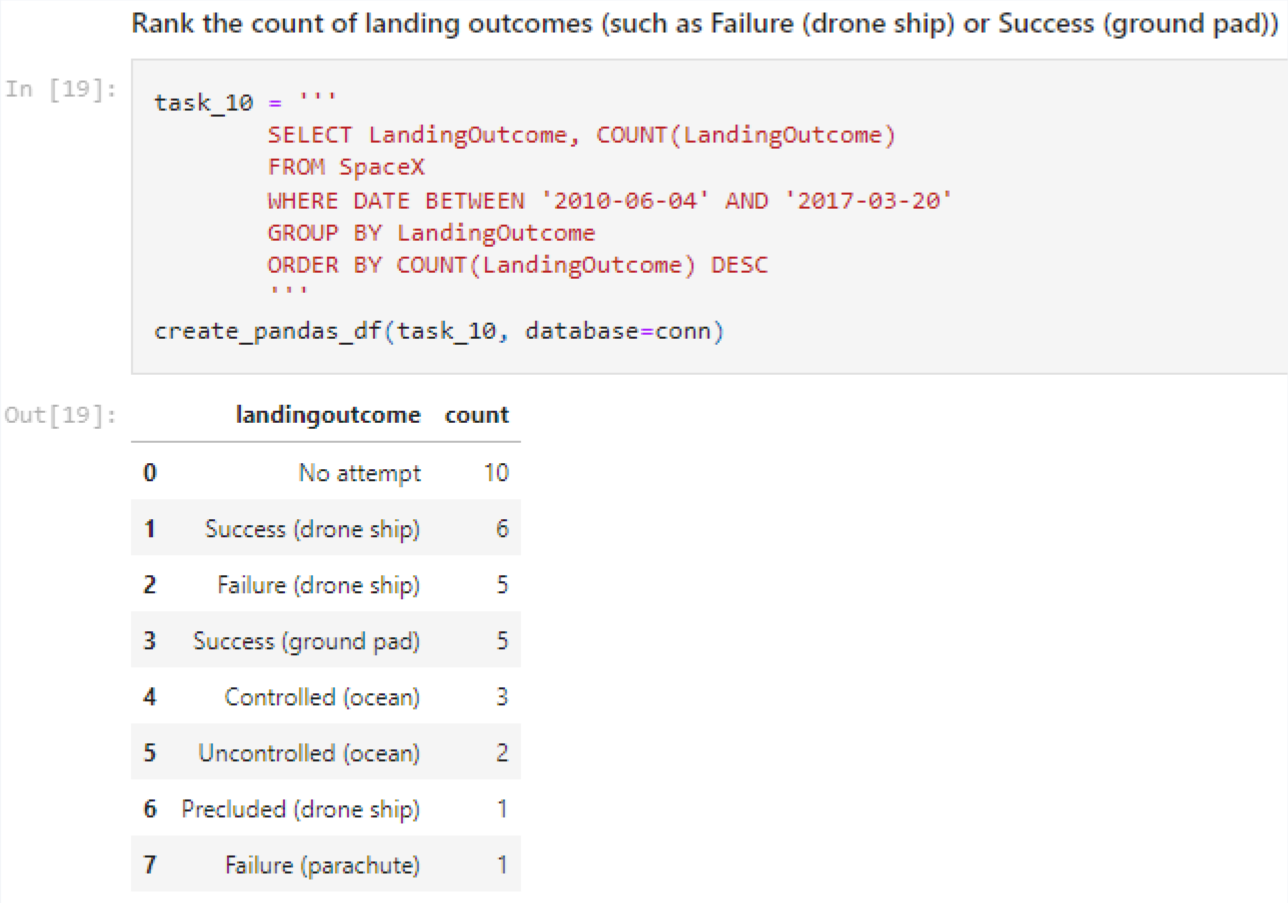
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	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

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



Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

```
In [19]:
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)
```

Out[19]:

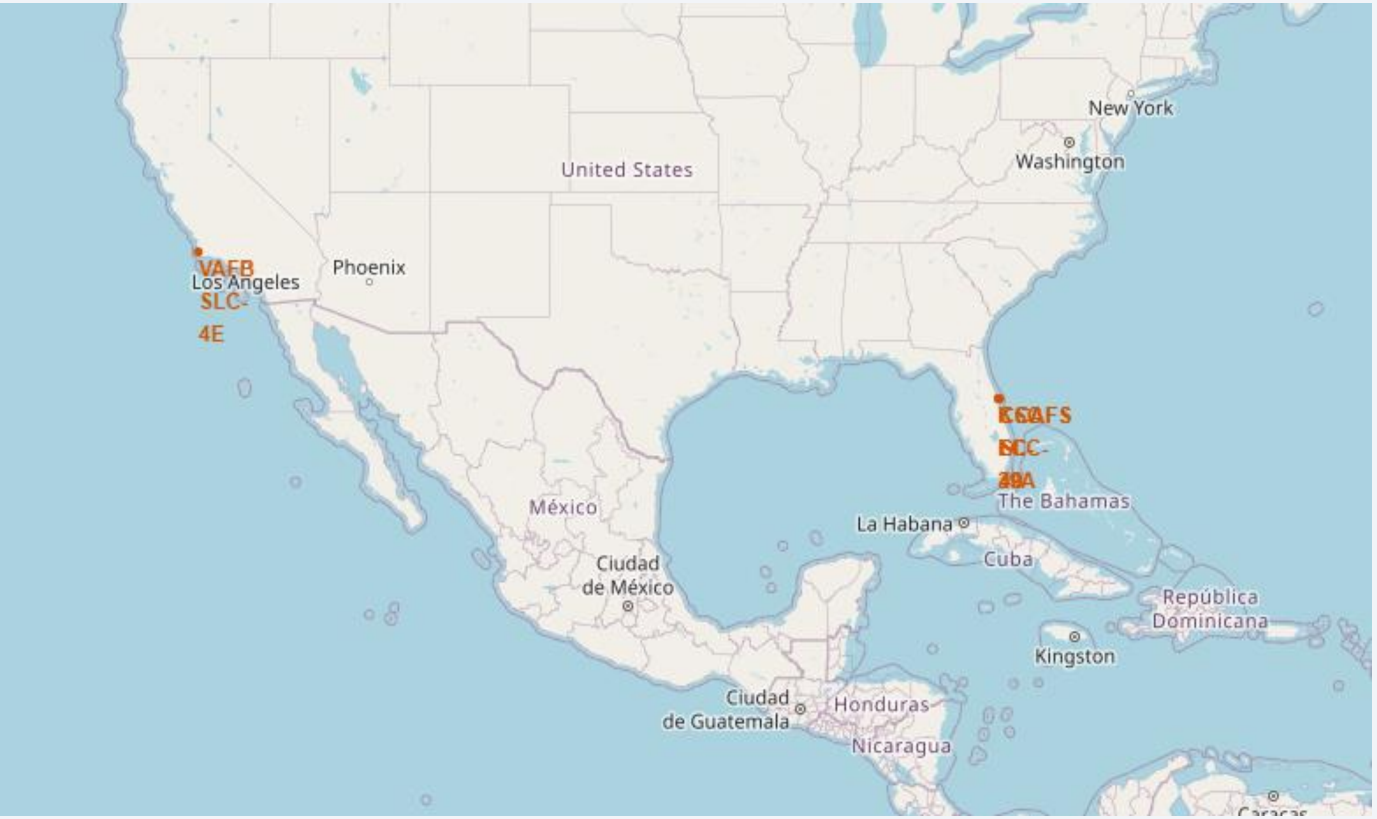
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

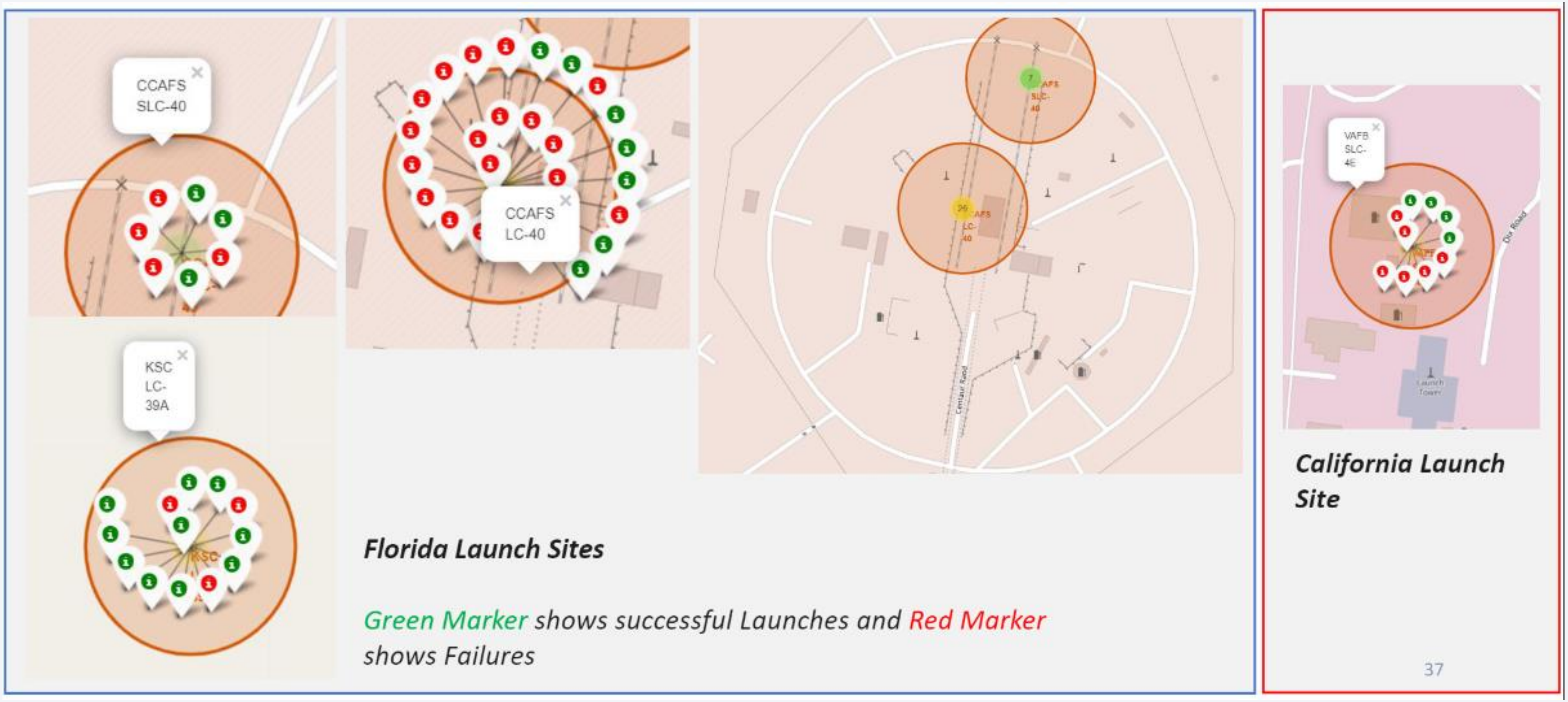
Launch Sites Proximities Analysis

Launch Site Locations

- The launch side can be seen which is in Los Angeles and Florida Coasts in the United States.



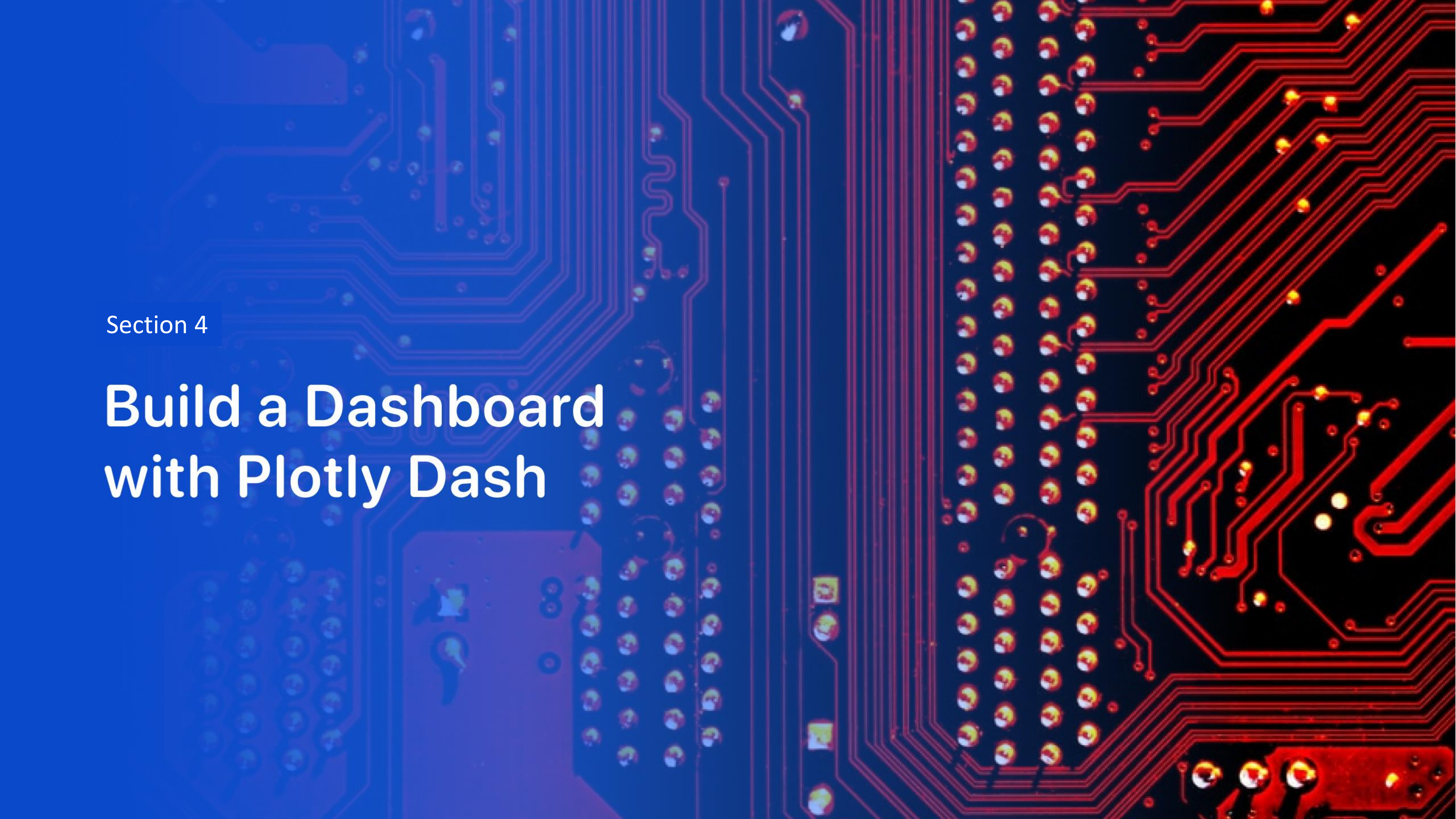
Markers showing launch sites with color labels



Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

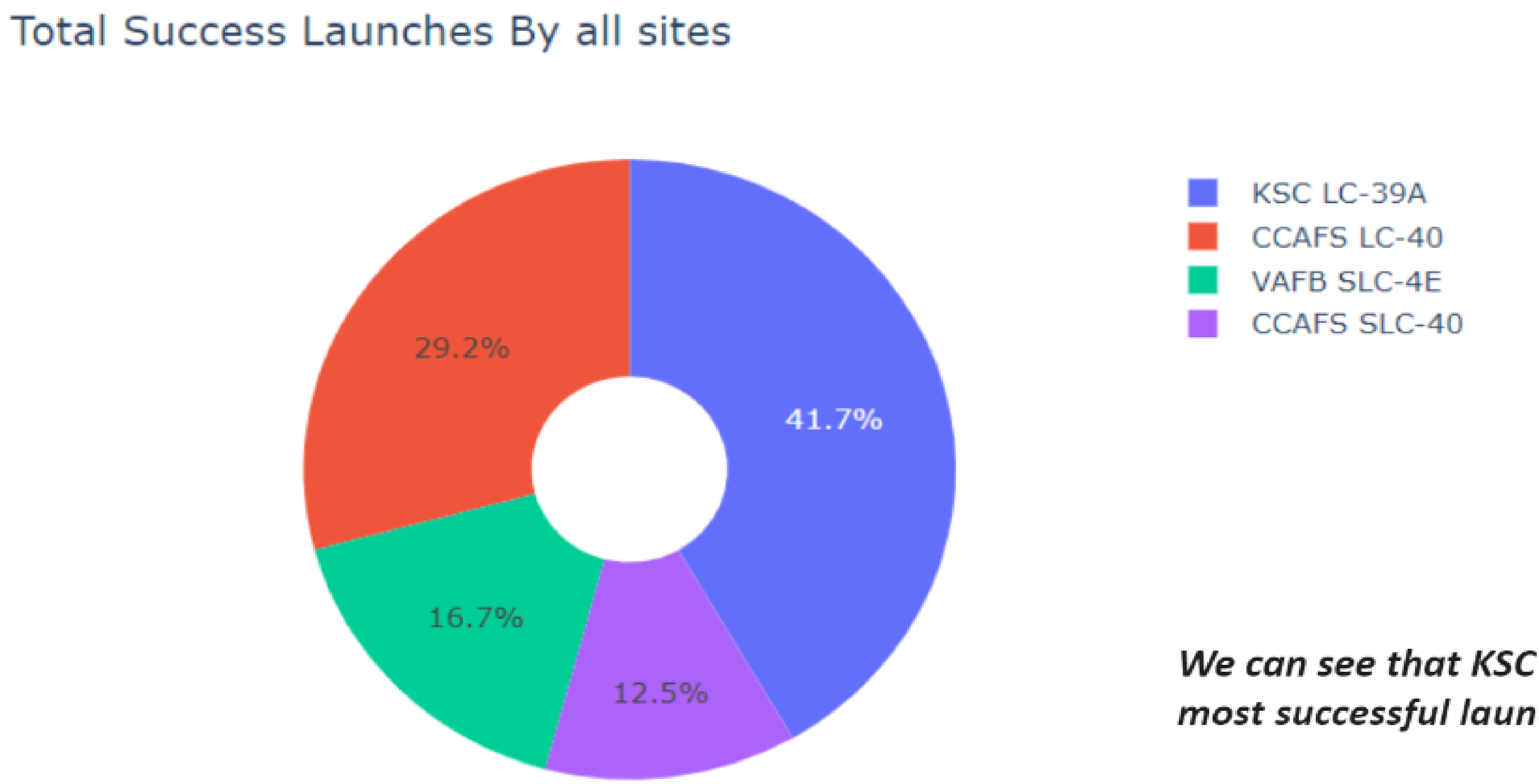
California Launch Site



Section 4

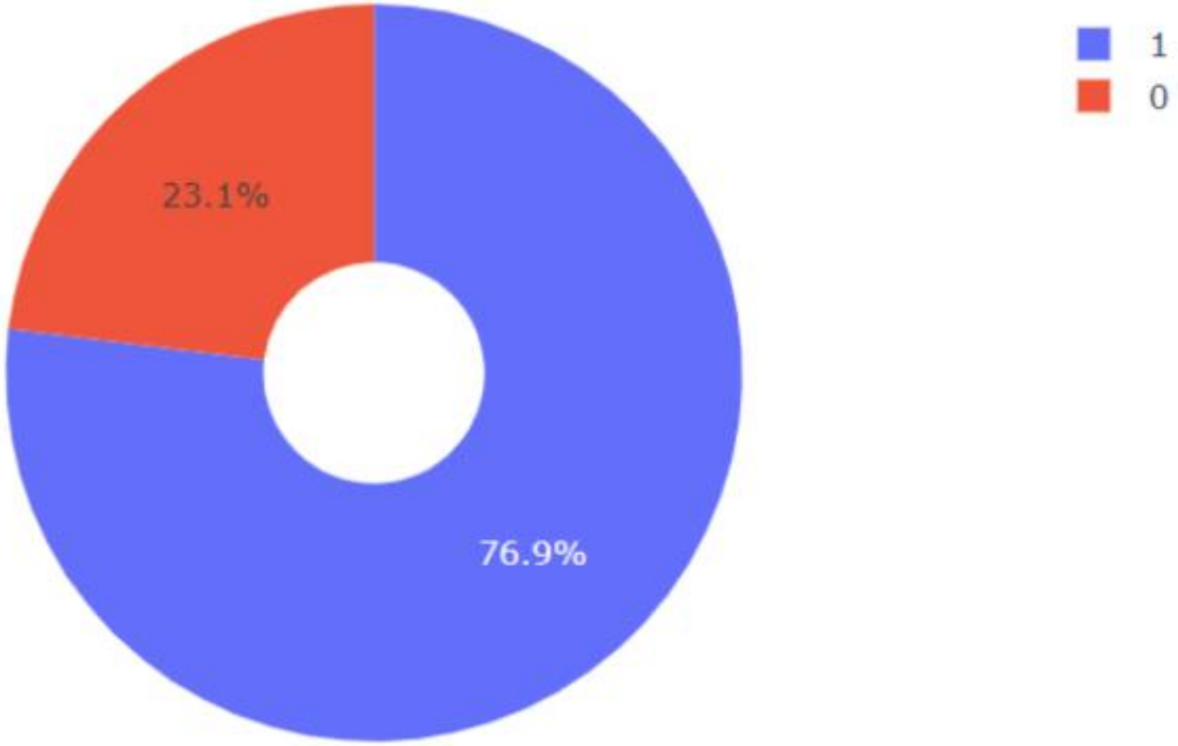
Build a Dashboard with Plotly Dash

The success percentage achieved by each launch site



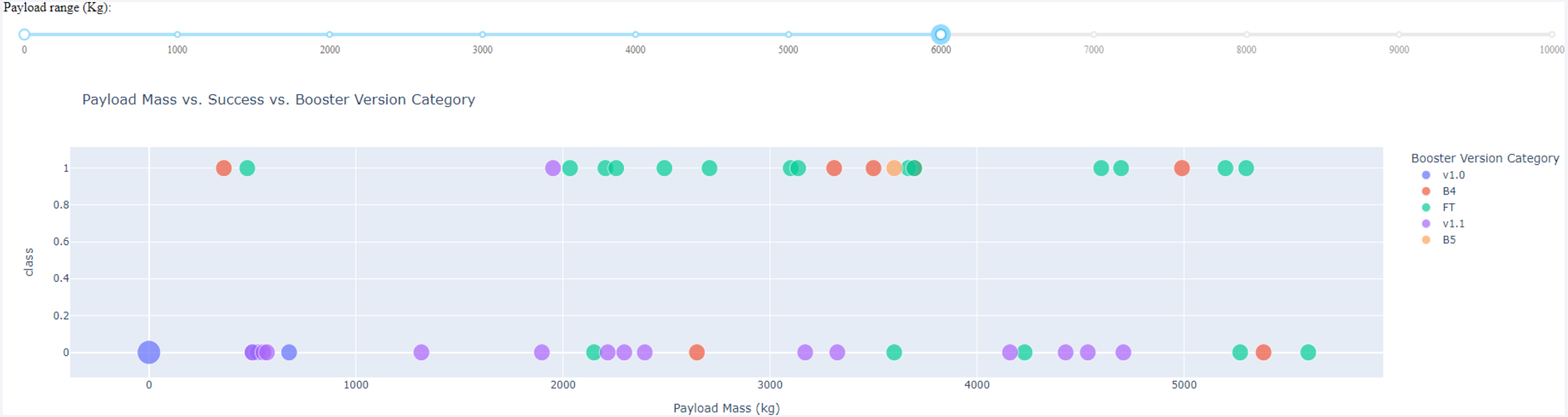
We can see that KSC LC-39A had the most successful launches from all the sites

Launch site with the highest launch success ratio



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

Payload Mass vs. Success vs. Booster Version Category



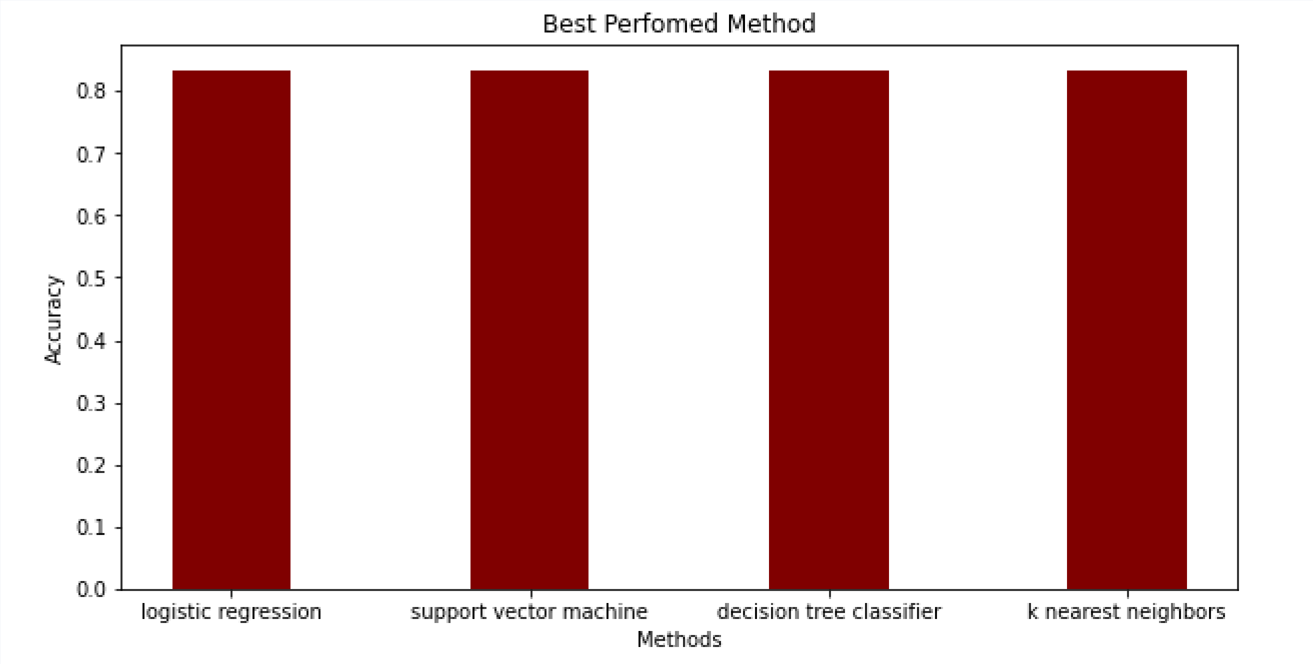
Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.

Section 5

Predictive Analysis (Classification)

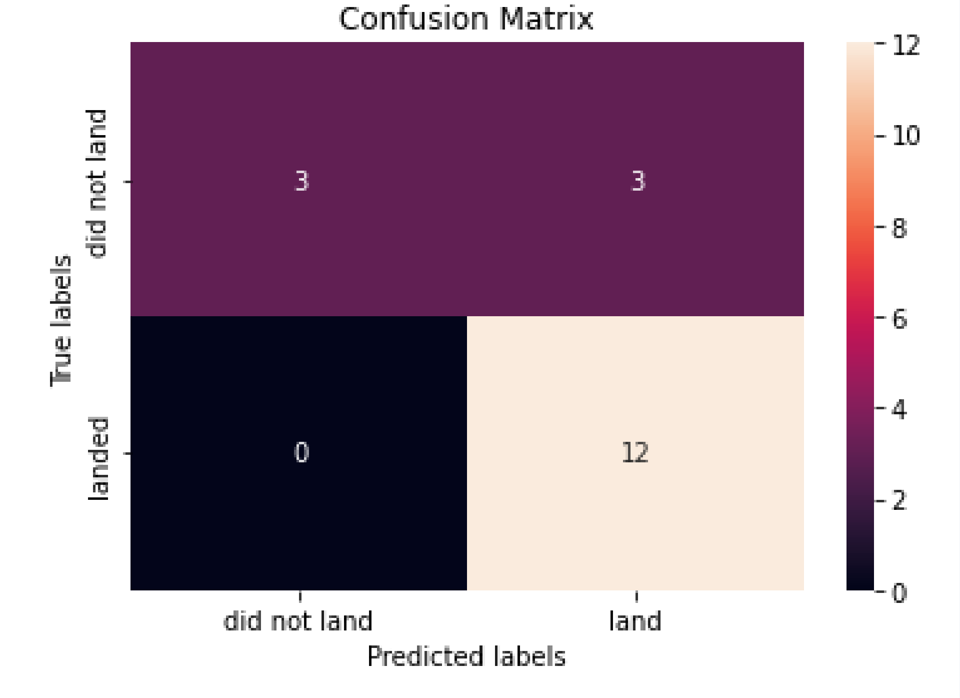
Classification Accuracy

- All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that the test size is small at only a sample size of 18.
- This can cause a large variance in accuracy results, such as those in the Decision Tree Classifier model in repeated runs.
- We likely need more data to determine the best model.



Confusion Matrix

- The confusion matrix for the classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives.



Conclusions

In conclusion:

- 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.
- Machine learning was able to provide accuracy up to 80% rate.

Thank you!