IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pablo Avila Segura
29 December 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory 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

Project Background and Context

SpaceX promotes Falcon 9 rocket launches on its website at a cost of $62 million, whereas other providers charge upwards of $165 million per launch. A significant portion of these savings is attributed to SpaceX's ability to reuse the first stage of the rocket. Consequently, by predicting the successful landing of the first stage, one can estimate the cost of a launch. This information is invaluable for any alternative company wishing to compete with SpaceX for rocket launch contracts. The primary objective of this project is to develop a machine learning pipeline capable of predicting the successful landing of the first stage.

Problems to Address

1. What factors influence the successful landing of the rocket?
2. How do various features interact to impact the success rate of a landing?
3. What operational conditions must be met to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

Data Collection Methodology:

The data for this project was sourced from two primary channels:

- 1. **SpaceX API**: Data was retrieved from the official SpaceX API, accessible at https://api.spacexdata.com/v4/rockets/, providing detailed information on various rocket launches.

- 2. **Web Scraping**: Additional data was obtained through web scraping from Wikipedia's comprehensive list of Falcon 9 and Falcon Heavy launches, available at https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches.

Data Wrangling:

The collected data underwent a thorough wrangling process to enhance its utility. This involved the creation of a landing outcome label, derived from the outcome data, which was summarized and analyzed to provide a clearer understanding of the factors influencing successful landings. The enriched dataset is now poised for further analysis and the development of predictive models.

Methodology

Executive Summary

The process of conducting an exploratory data analysis (EDA), performing interactive visual analytics, and executing predictive analysis using classification models is outlined as follows:

1. **Exploratory Data Analysis (EDA) using Visualization and SQL:**

- - **Visualization:** Utilize data visualization tools such as Matplotlib and Seaborn to create plots and charts that provideinsights into the data distribution, trends, and relationships. Common visualizations include histograms, scatter plots, box plots, and heatmaps.
- - **SQL Analysis:** Employ SQL queries to explore the dataset, identify key statistics, and filter or aggregate data as needed. This involves querying the data for summary statistics, identifying missing values, and understanding the data structure.

2. **Interactive Visual Analytics using Folium and Plotly Dash:**

- - **Folium:** Leverage Folium to create interactive maps that visualize geographical data points related to rocket launches.This can include mapping launch sites, trajectories, and landing zones to provide a spatial understanding of the data.
- - **Plotly Dash:** Implement Plotly Dash to build interactive dashboards that allow users to explore and interact with the data dynamically. Dashboards can include various charts and graphs, sliders, and dropdowns to filter and visualize data in real-time.

3. **Predictive Analysis using Classification Models:**

- - **Data Preparation:** Normalize the data to ensure it is on a consistent scale, and divide it into training and test datasets to facilitate model evaluation.
- - **Model Selection and Evaluation:** Use four different classification models, such as Logistic Regression, Decision Trees,Random Forest, and Support Vector Machines (SVM), to predict landing outcomes or other target variables.
- - **Parameter Tuning:** Evaluate the accuracy of each model by experimenting with different combinations of parameters. Techniques such as cross-validation and grid search can be employed to optimize model performance.
- - **Model Comparison:** Compare the models based on their accuracy, precision, recall, and other relevant metrics to determine the most effective model for the data.

Through these steps, a comprehensive understanding of the data is achieved, and predictive insights are generated to inform decision-making processes.

Data Collection

The datasets were collected from two main sources: the SpaceX API and Wikipedia.

1. **SpaceX API:**

   - Accessed via https://api.spacexdata.com/v4/rockets/.

   - Data was retrieved using HTTP GET requests, typically with Python libraries like `requests`.

   - The API returned data in JSON format, which was parsed to extract relevant information such as rocket specifications and launch history.

2. **Wikipedia:**

   - Data was extracted from the page [List of Falcon 9 and Falcon Heavy launches](https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches).

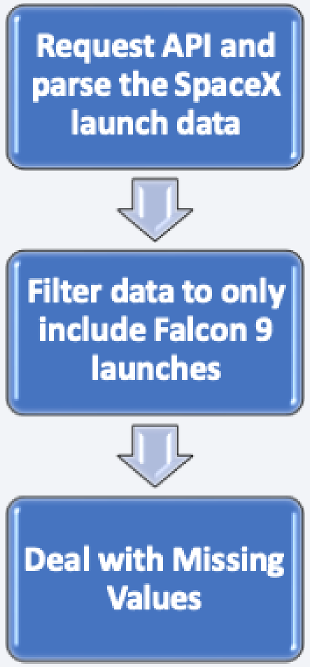
   - Web scraping techniques, using tools like BeautifulSoup, were used to parse HTML tables for launch dates, payloads, and mission outcomes.

   - The data was cleaned to ensure accuracy and consistency.

This approach provided a comprehensive dataset, combining technical details and historical launch information.

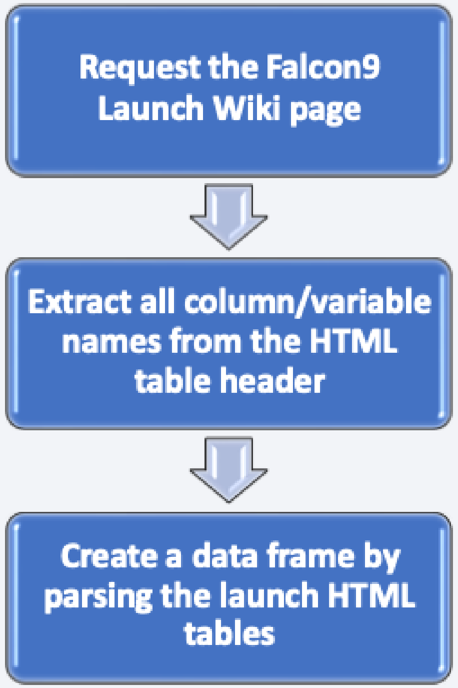
Data Collection – SpaceX API

- SpaceX offers a public API from where data can be obtained and then used;
- This API was used according to the flowchart beside and then data is persisted.
- Source code: https://github.com/SouRitra01/IBM-Data-Science-Project/blob/main/Data%20Collection%20API.ipynb


Request API and parse the SpaceX launch data
Filter data to only include Falcon 9 launches
Deal with Missing Values

Data Collection - Scraping

- Data from SpaceX launches can also be obtained from Wikipedia;
- Data are downloaded from Wikipedia according to the flowchart and then persisted.
- Source code: [https://github.com/SouRitra01/IBM-Data- Science-Project/blob/main/Data%20Collection%20wit h%20Web%20Scraping.ipynb](https://github.com/SouRitra01/IBM-Data-Science-Project/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb)



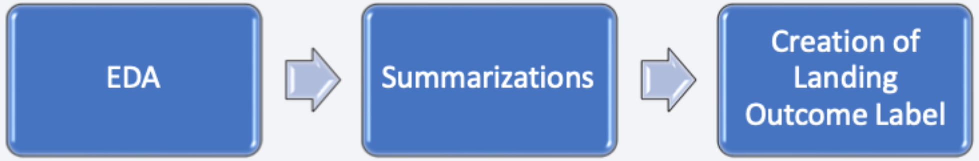
Request the Falcon9 Launch Wiki page

Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

Data Wrangling

- Initially some Exploratory Data Analysis (EDA) was performed on the dataset.
- Then the summary launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.
- Finally, the landing outcome label was created from Outcome column.



EDA

Summarizations

Creation of Landing Outcome Label

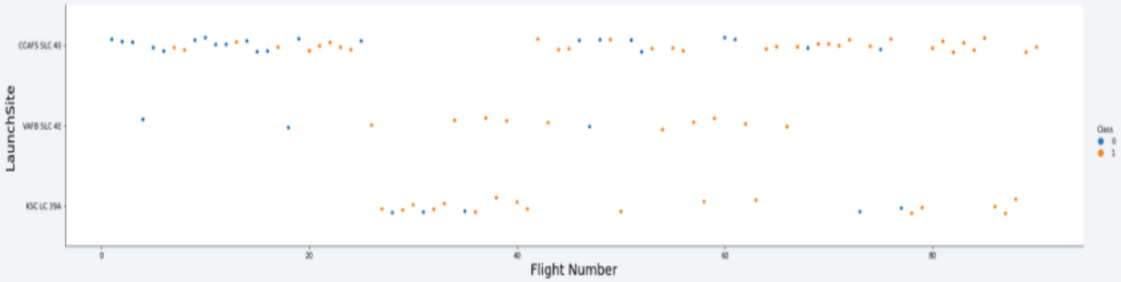
- Source code: https://github.com/SouRitra01/IBM-Data-Science-Project/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

- The following SQL queries were performed:
    - Names of the unique launch sites in the space mission;
    - Top 5 launch sites whose name begins with the string 'CCA’;
    - Total pay load mass carried by boosters launched by NASA (CRS);
    - Average payload mass carried by booster version F9 v1.1;
    - Date when the first successful landing outcome in ground pad was achieved;
    - Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
    - Total number of successful and failure mission outcomes;
    - Names of the booster versions which have carried the maximum payload mass;
    - Failed landing out comes in droneship, their booster versions, and launch site names for in year 2015; and
    - Rank of the count of landing outcomes (such as Failure (droneship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.
- Source code: https://github.com/SouRitra01/IBM-Data-Science-Project/blob/main/EDA.ipynb

EDA with SQL

- To explore data, scatterplots and bar plots were used to visualize the relationship between pair of features:
- Payload Mass X Flight Number, Launch Site X Flight Number, Launch Site X Payload Mass, Orbit and Flight Number, Payload and Orbit



- Source code: https://github.com/SouRitra01/IBM-Data-Science-Project/blob/main/EDA%20with%20Data%20Visualization.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines and marker clusters were used with Folium Maps
- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.
- Source code: https://github.com/SouRitra01/IBM-Data-Science-Project/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

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.
- The link to the notebook is https://github.com/chuksoo/IBM-Data-Science-Capstone-SpaceX/blob/main/app.py

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.
- The link to the notebook is https://github.com/chuksoo/IBM-Data-Science-Capstone-SpaceX/blob/main/Machine%20Learning%20Prediction.ipynb

Results

The exploratory data analysis of SpaceX reveals the following:

1. **Launch Sites:** SpaceX uses four different launch sites.

2. **Initial Launches:** First launches were for SpaceX and NASA.

3. **Payload Capacity:** Falcon 9 v1.1 has an average payload of 2,928 kg.

4. **Successful Landings:** First successful landing occurred in 2015, five years after the first launch.

5. **Drone Ship Landings:** Many Falcon 9 versions successfully landed on drone ships with payloads above average.

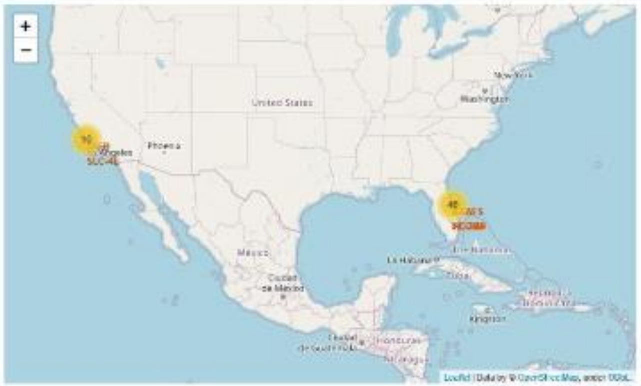
6. **Mission Success Rate:** Almost 100% of missions were successful.

7. **Landing Failures in 2015:** F9 v1.1 B1012 and F9 v1.1 B1015 failed to land on drone ships in 2015.

8. **Improvement Over Time:** Landing outcomes improved as years passed.

Results

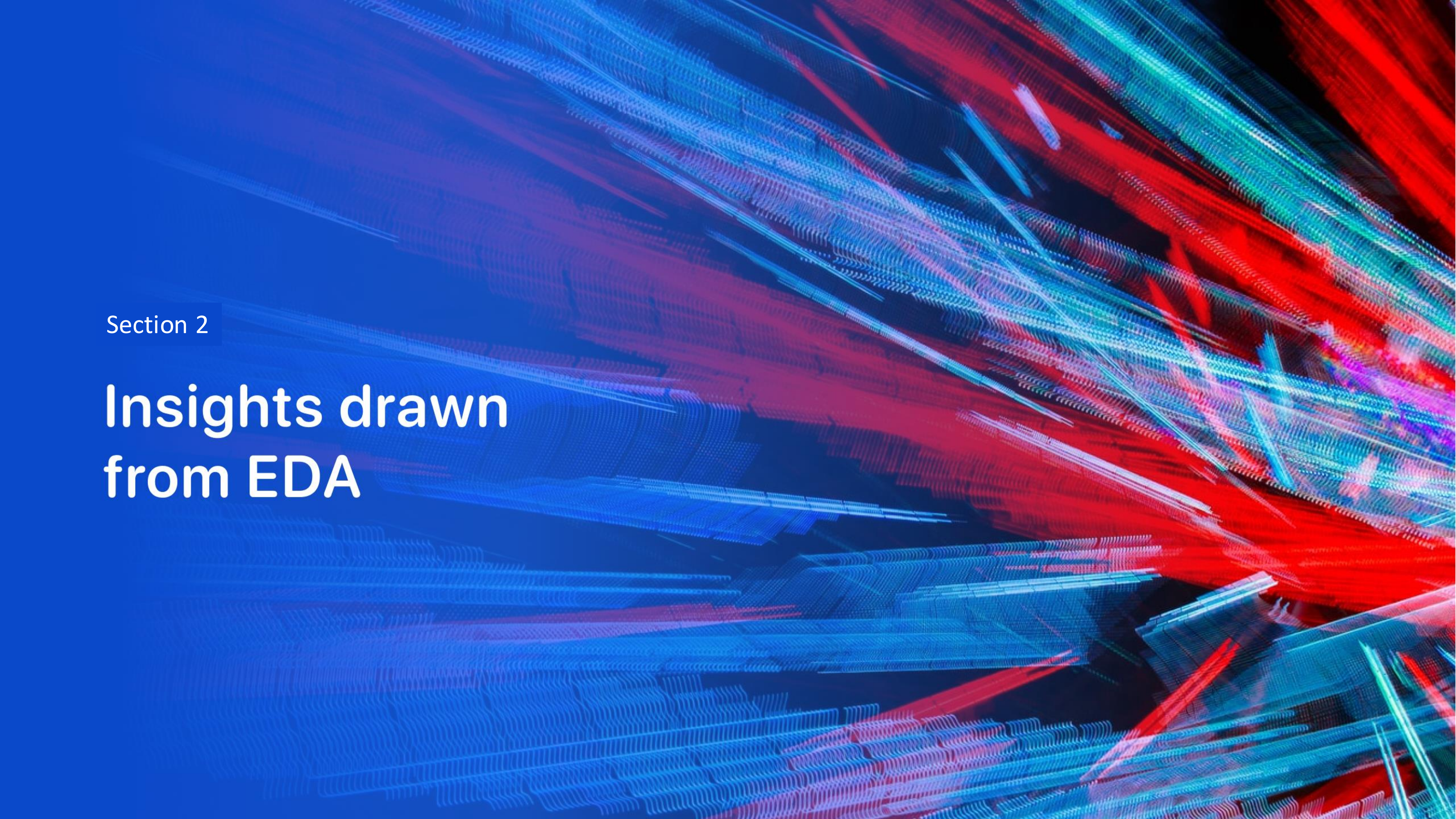
- Using interactive analytics was possible to identify that launch sites use to be in safety places, near sea, for example and have a good logistic infrastructure around.
- Most launches happens at east cost launch sites.



+
−
Lat: 28.55251 Long: -80.83166
13
KM
LC-39A
33
Leaflet | Data by © OpenStreetMap, under ODbL.

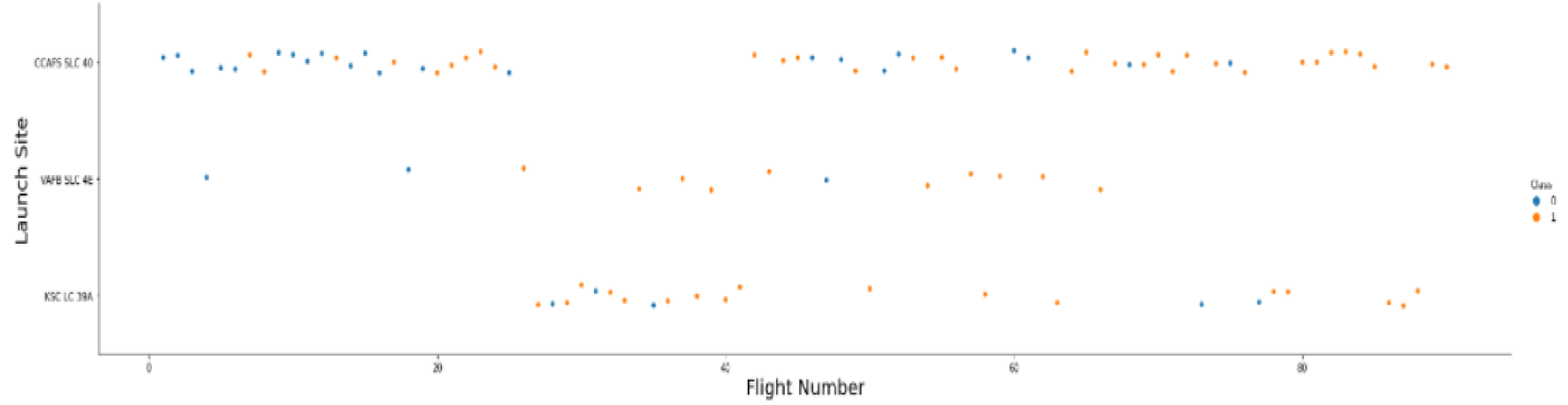
Section 2

Insights drawn from EDA



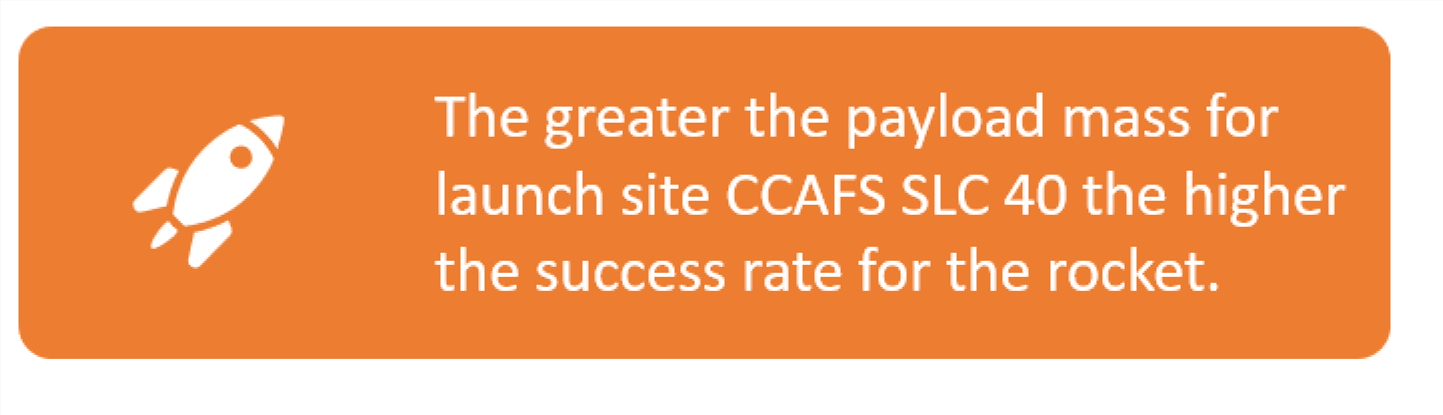
Flight Number vs. Launch Site

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

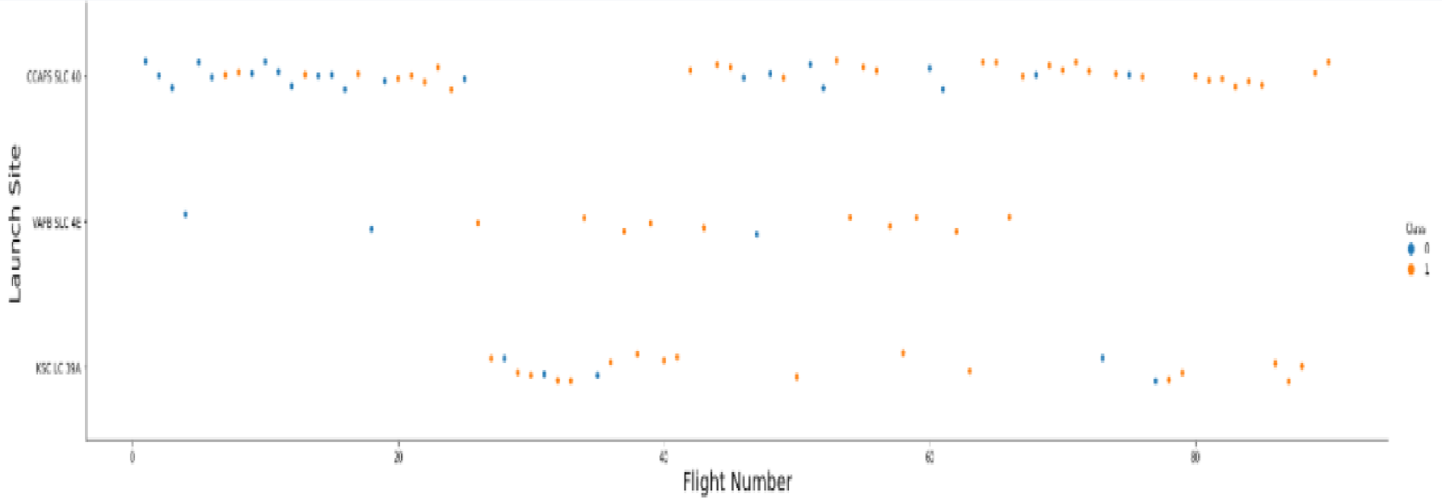


Payload vs. Launch Site

Payload vs. Launch Site

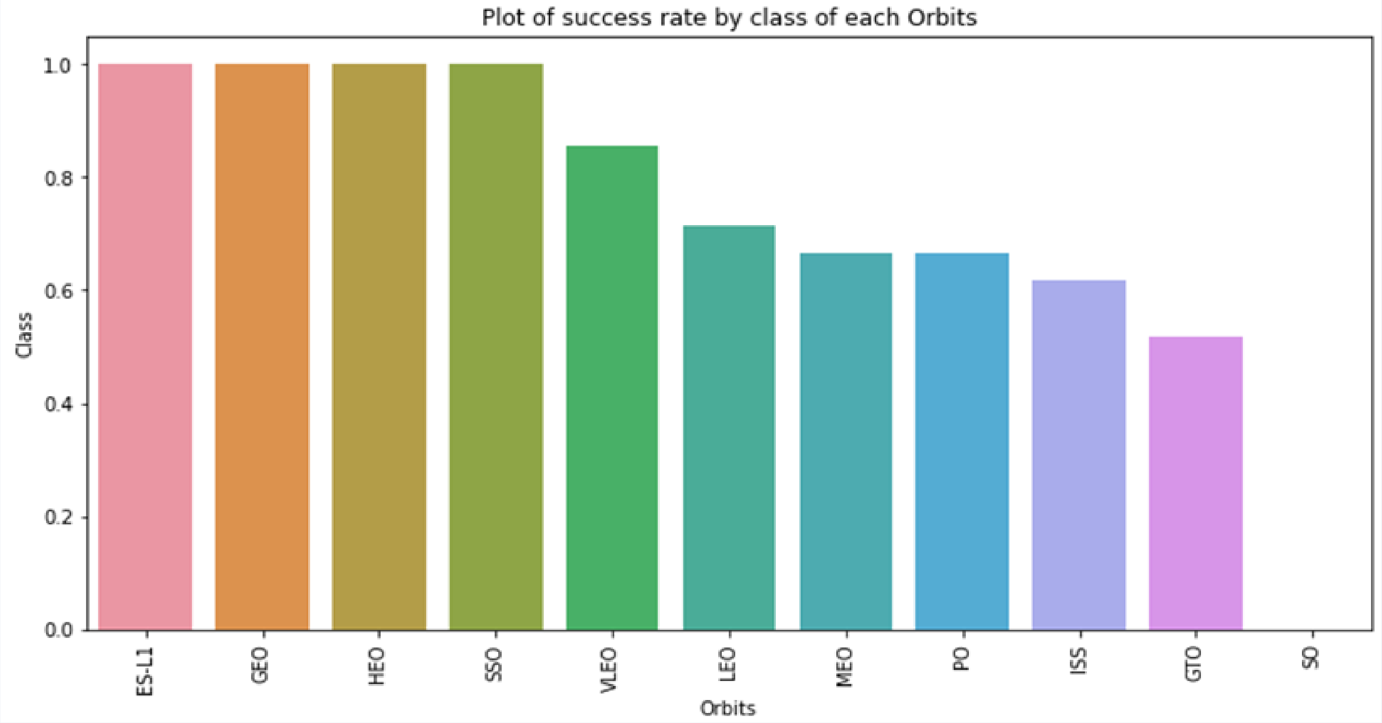


The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



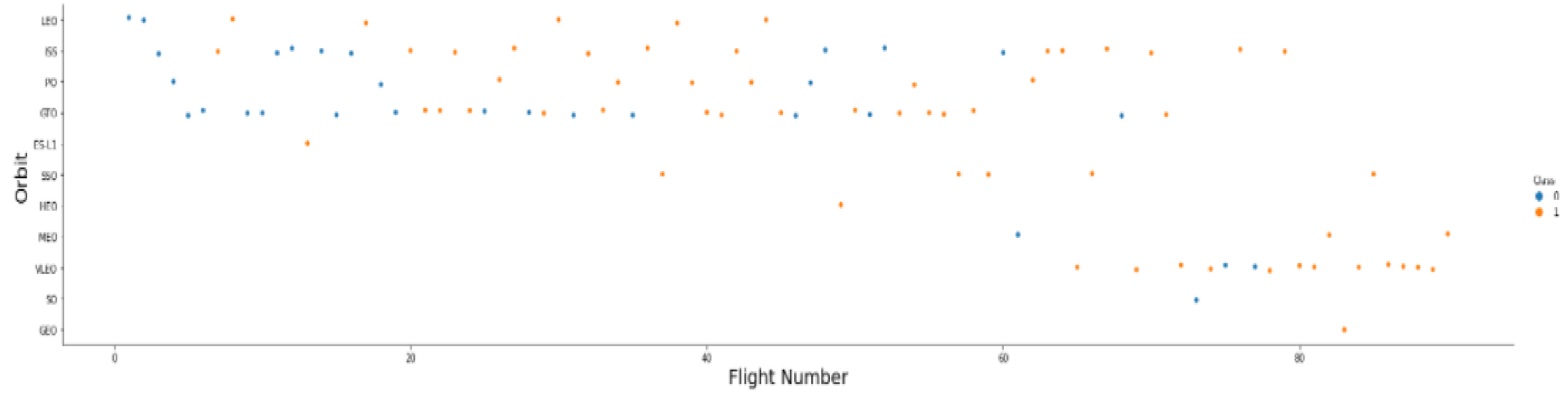
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



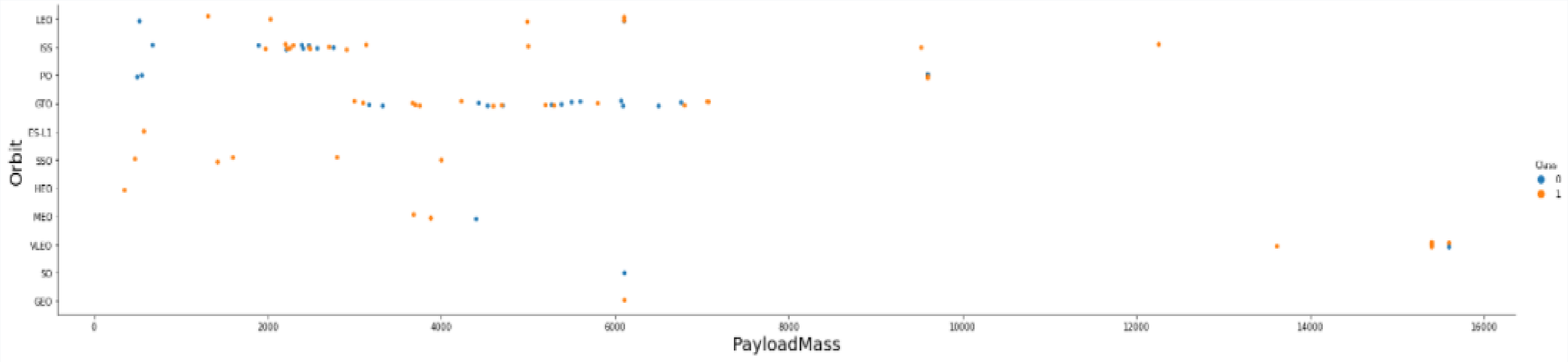
Flight Number vs. Orbit Type

The plot presented below illustrates the relationship between Flight Number and Orbit Type. It is evident that within the LEO orbit, success appears to be associated with the frequency of flights. Conversely, in the GTO orbit, there is no discernible relationship between the number of flights and the orbit.



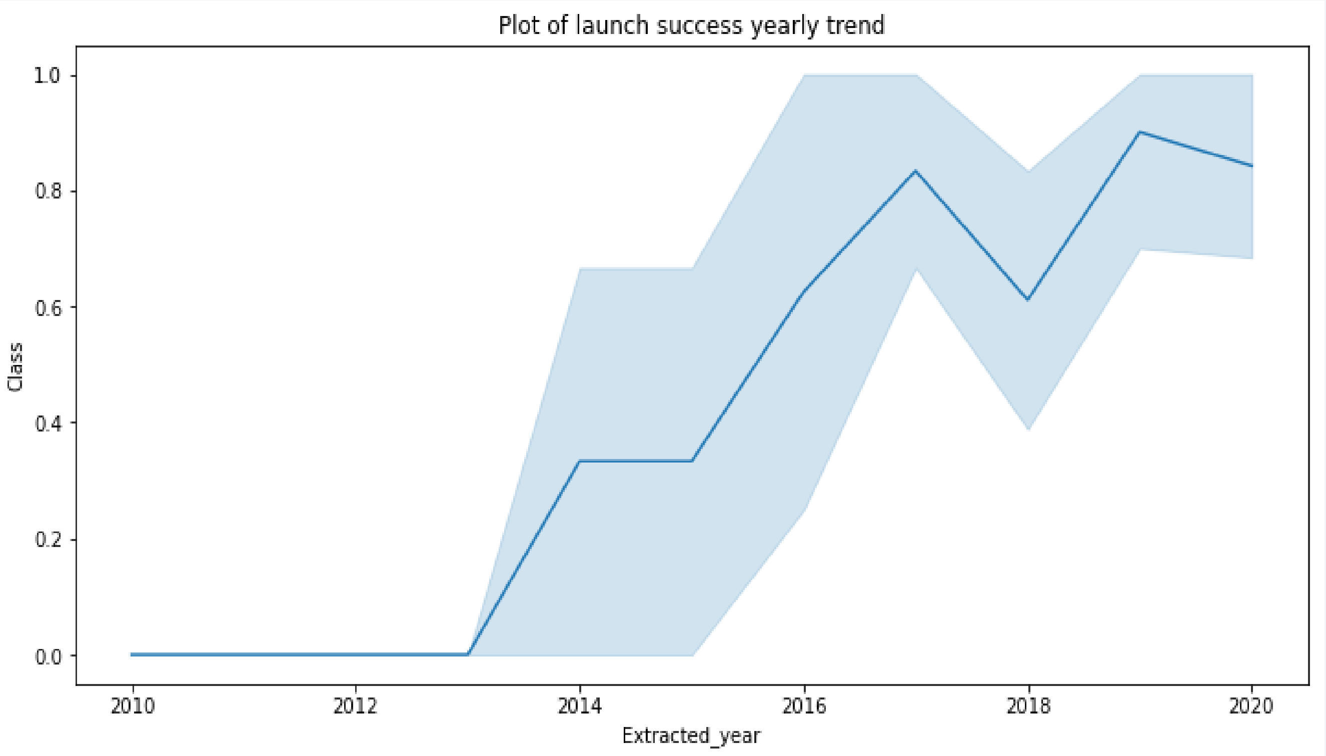
Payload vs. Orbit Type

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



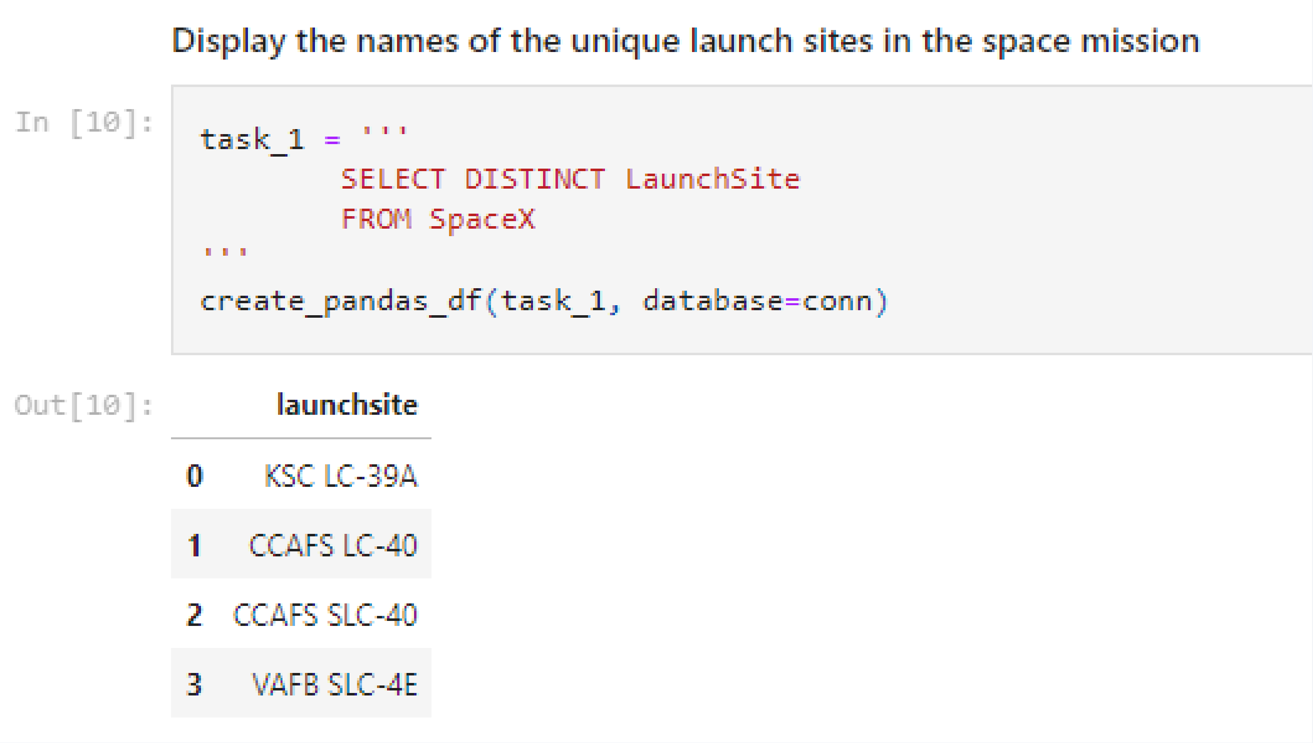
Launch Success Yearly Trend

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



All Launch Site Names

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



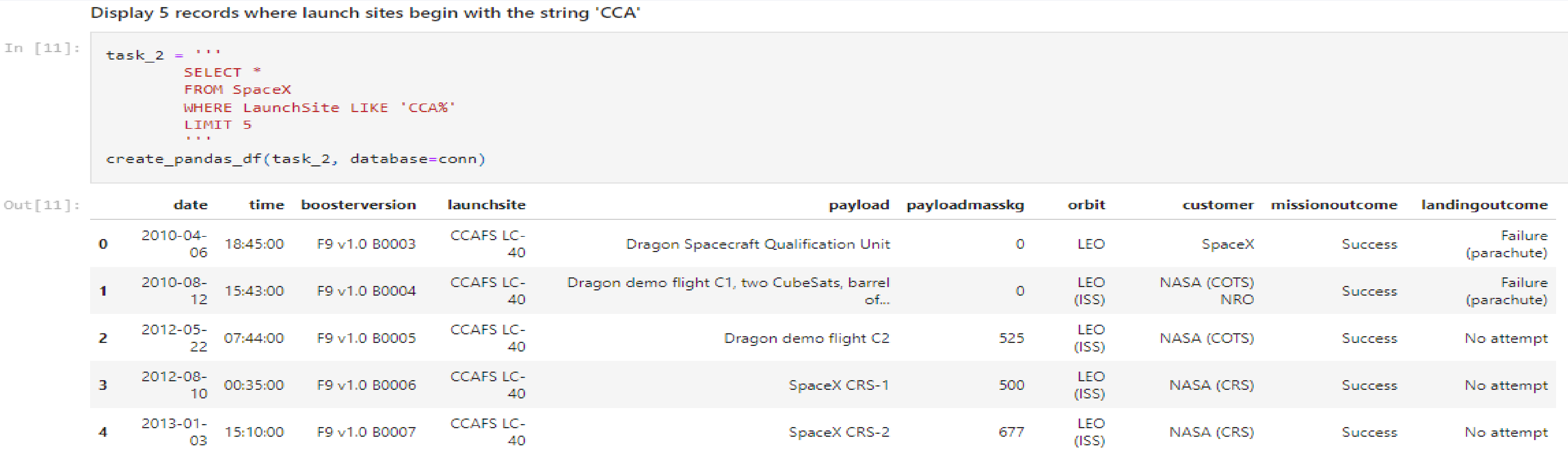
Display the names of the unique launch sites in the space mission

```
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'



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)
```

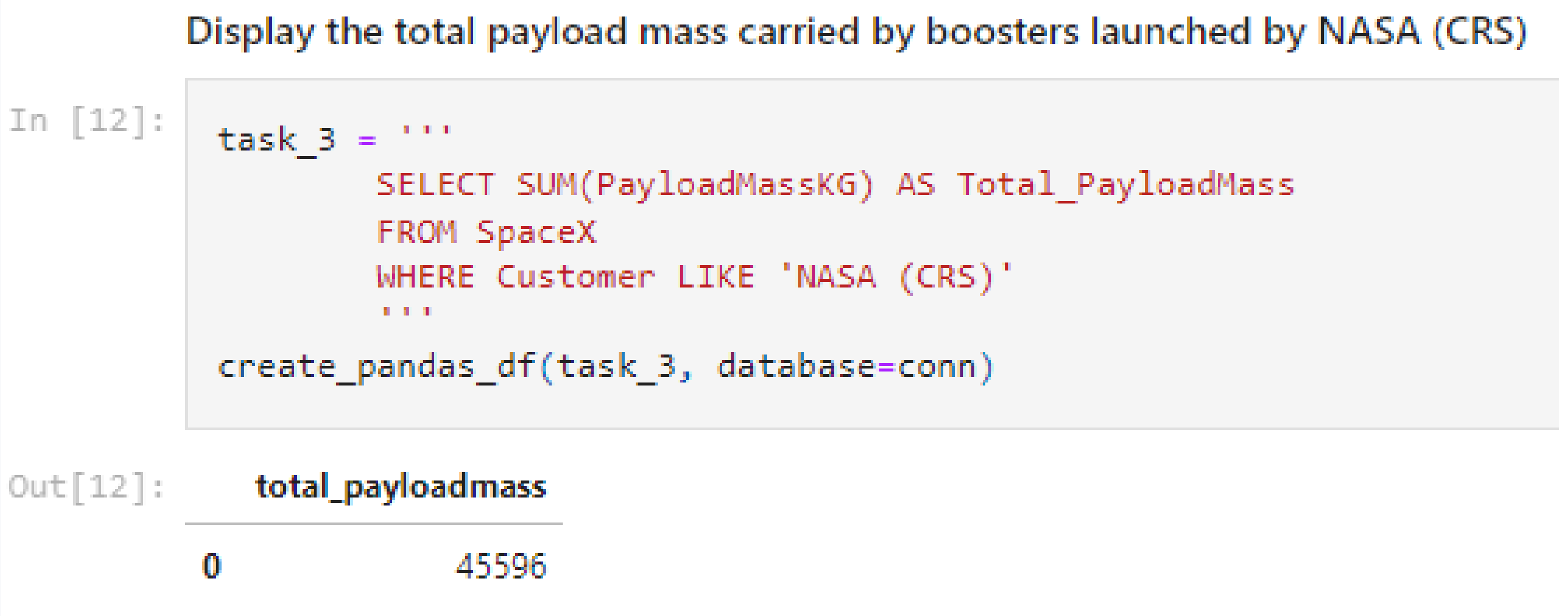
Out[11]:

	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

- We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below



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

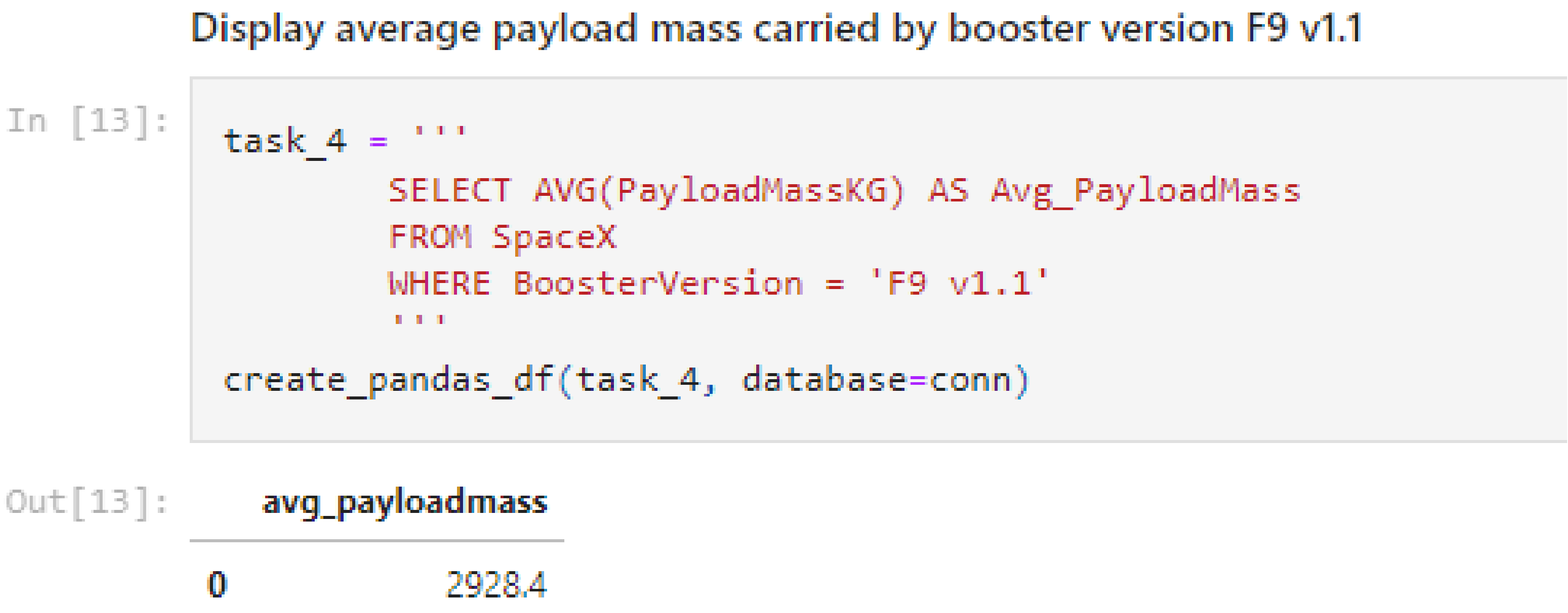
```
In [12]: task_3 = '''
                 SELECT SUM(PayloadMassKG) AS Total_PayloadMass
                 FROM SpaceX
                 WHERE Customer LIKE 'NASA (CRS)'
                 '''
         create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4



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

In [13]:

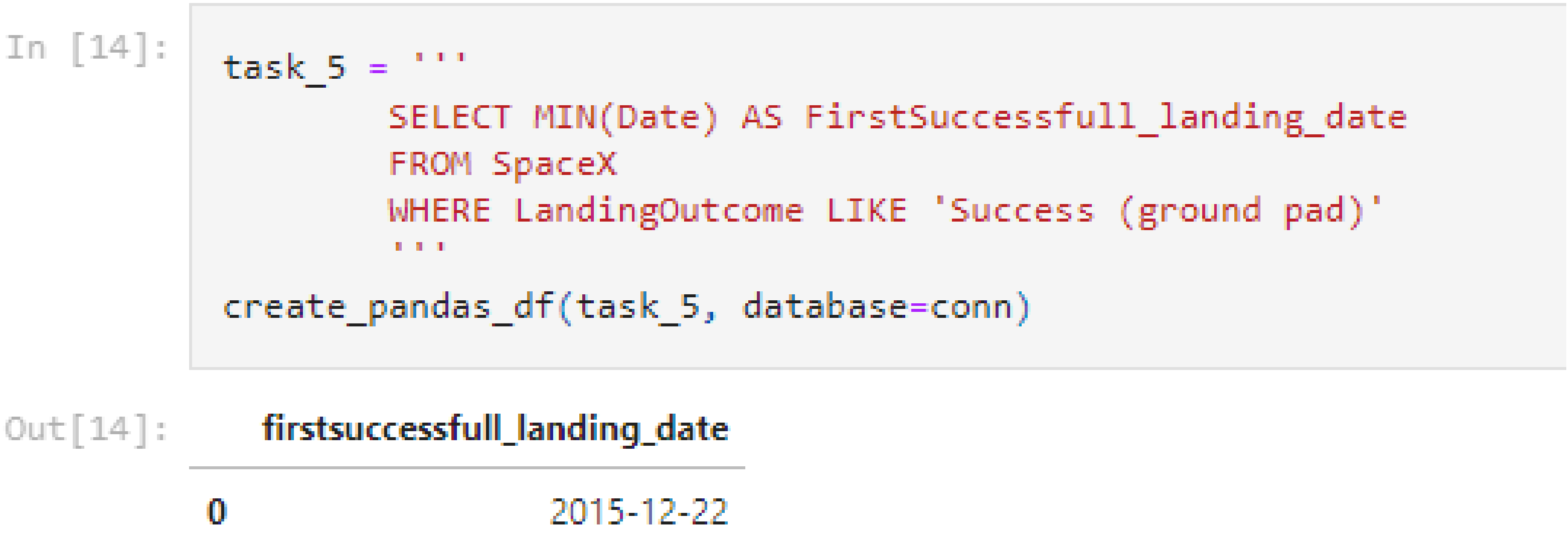
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

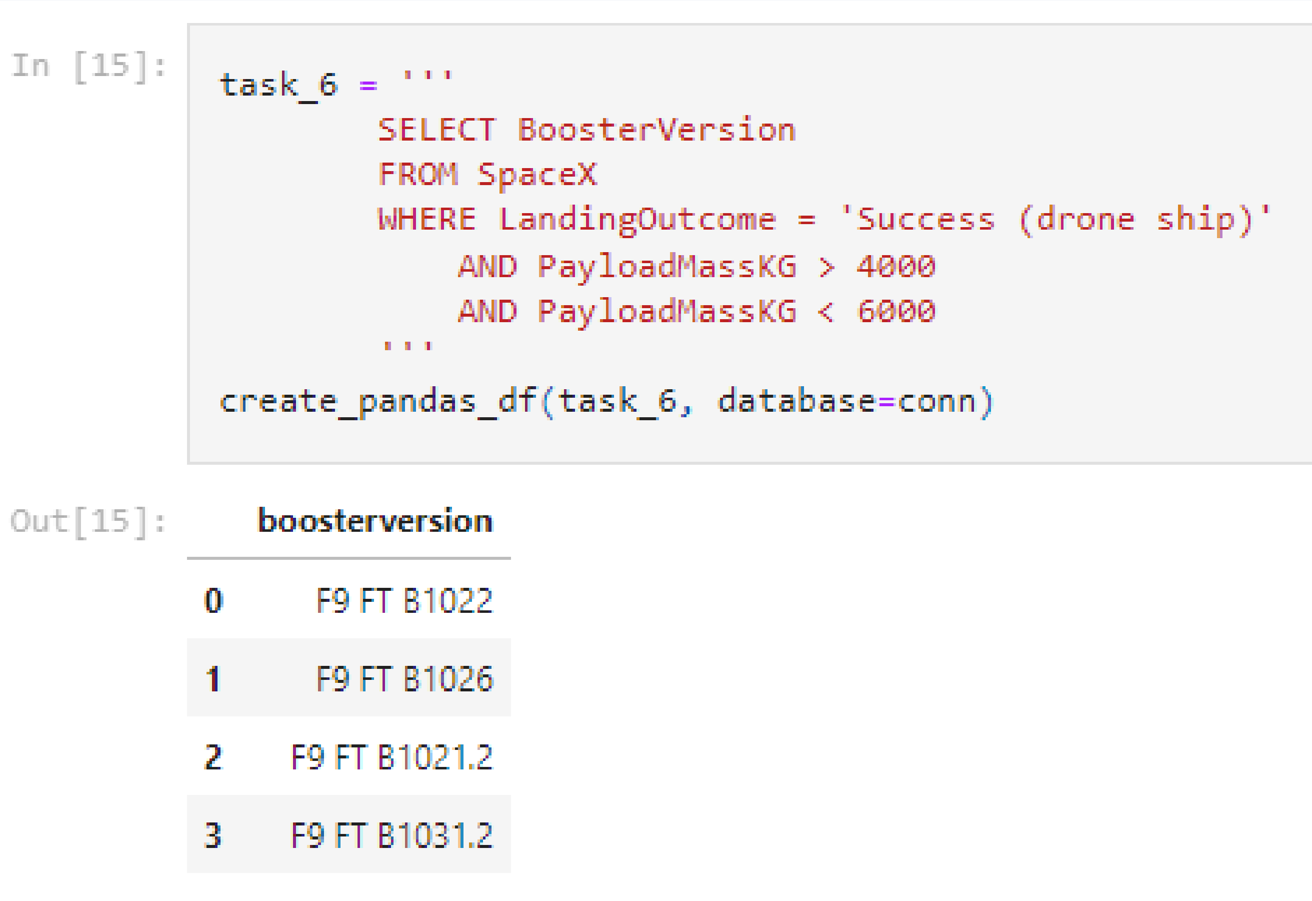
- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015



```
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

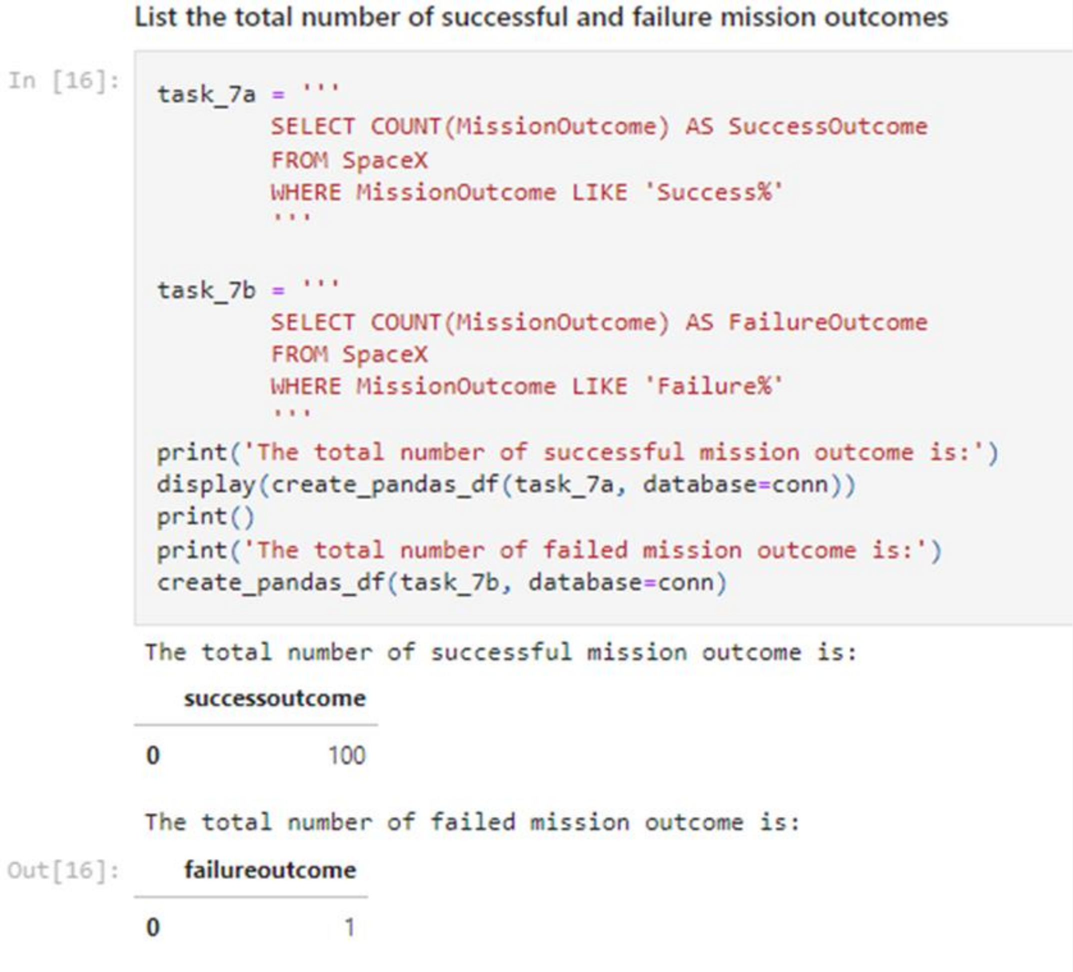


```
In [15]:  task_6 = '''
                  SELECT BoosterVersion
                  FROM SpaceX
                  WHERE LandingOutcome = 'Success (drone ship)'
                      AND PayloadMassKG > 4000
                      AND PayloadMassKG < 6000
                  '''
          create_pandas_df(task_6, database=conn)
```

```
Out[15]:    boosterversion
         0     F9 FT B1022
         1     F9 FT B1026
         2   F9 FT B1021.2
         3   F9 FT B1031.2
```

- 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

Total Number of Successful and Failure Mission Outcomes



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:
```

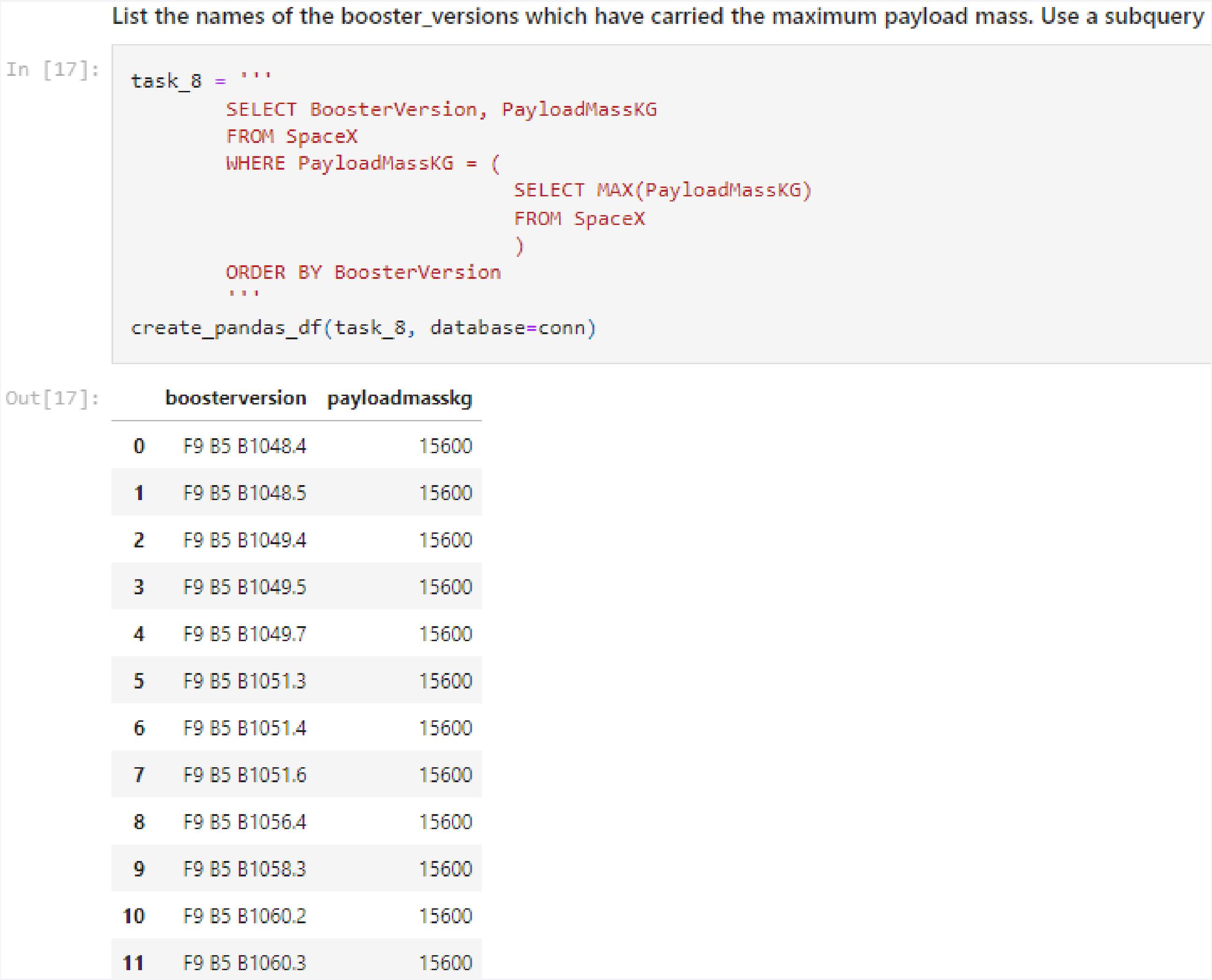
Out[16]:

	failureoutcome
0	1

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

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.



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

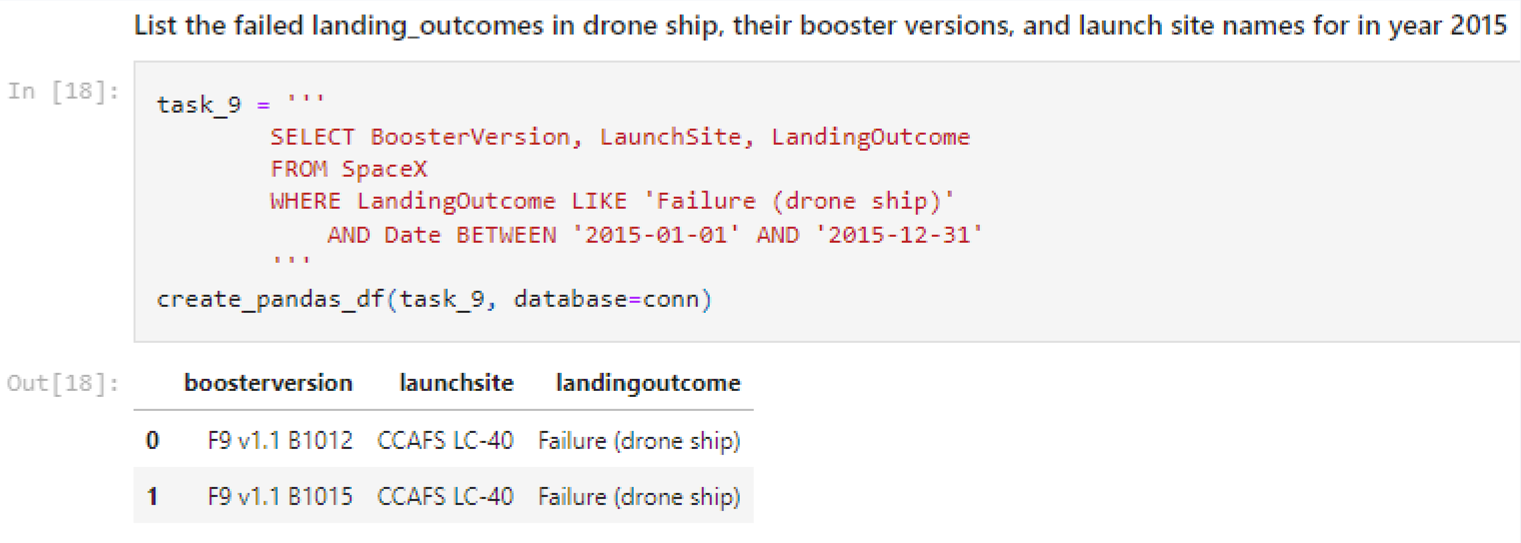
```
In [17]:
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''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

- 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



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

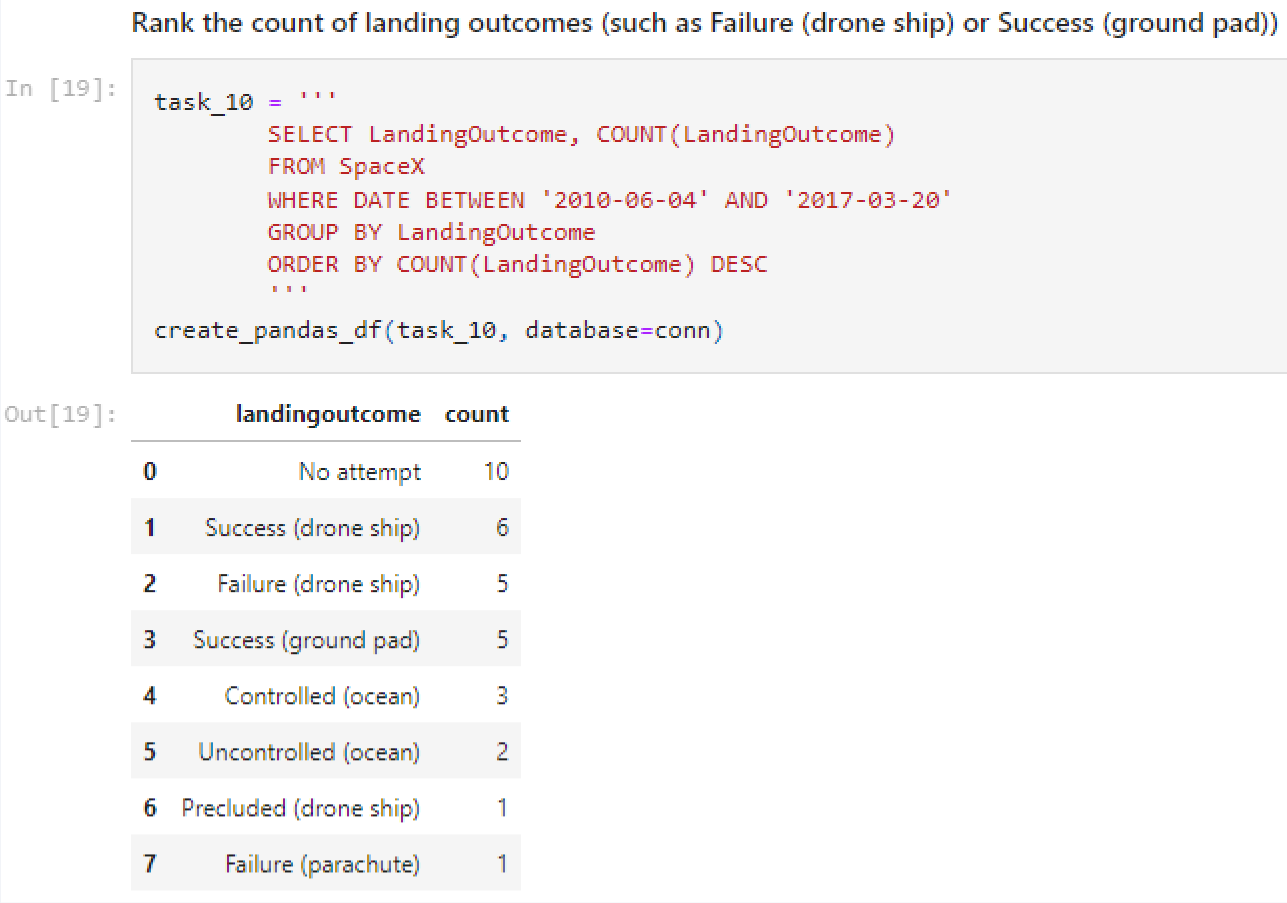
In [18]:

```
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



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

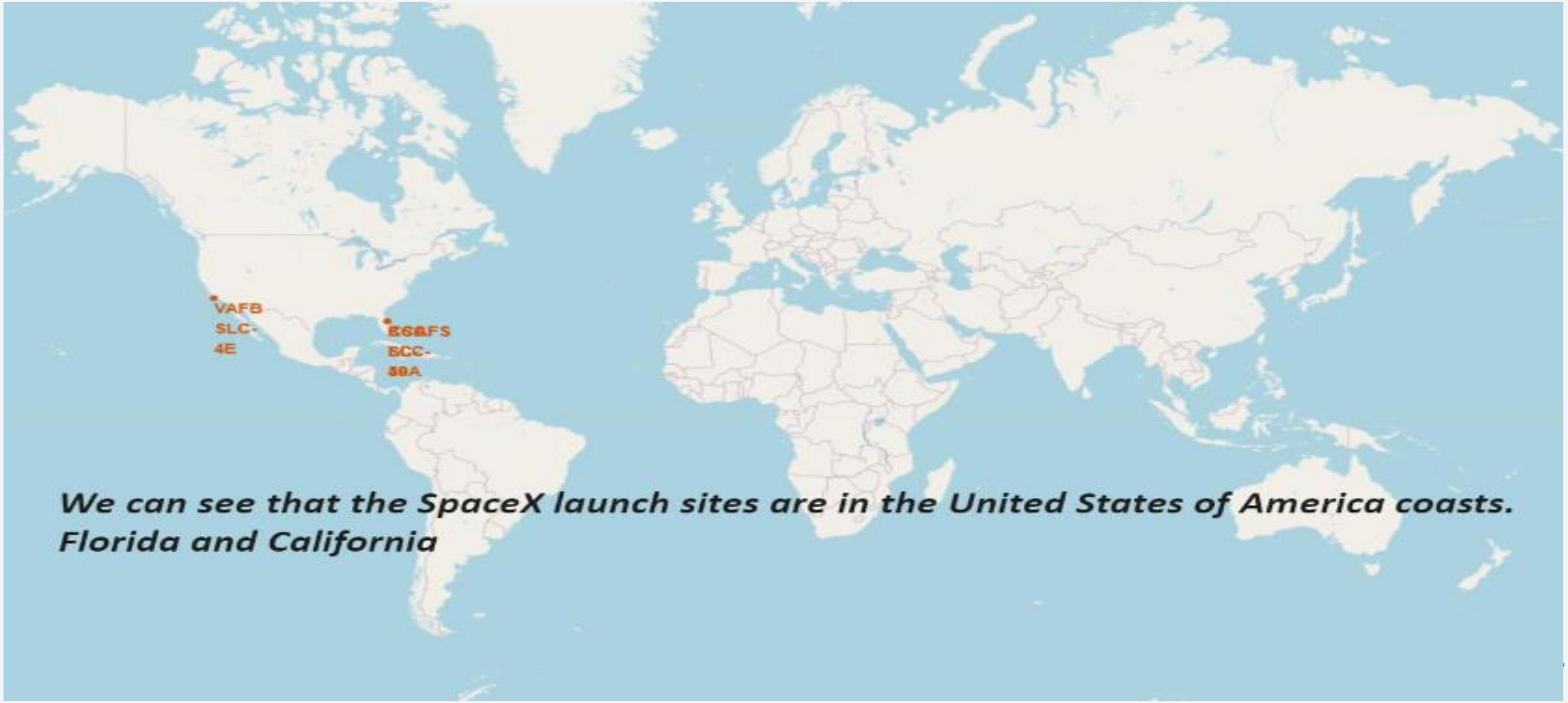
- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

- 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.

Section 3

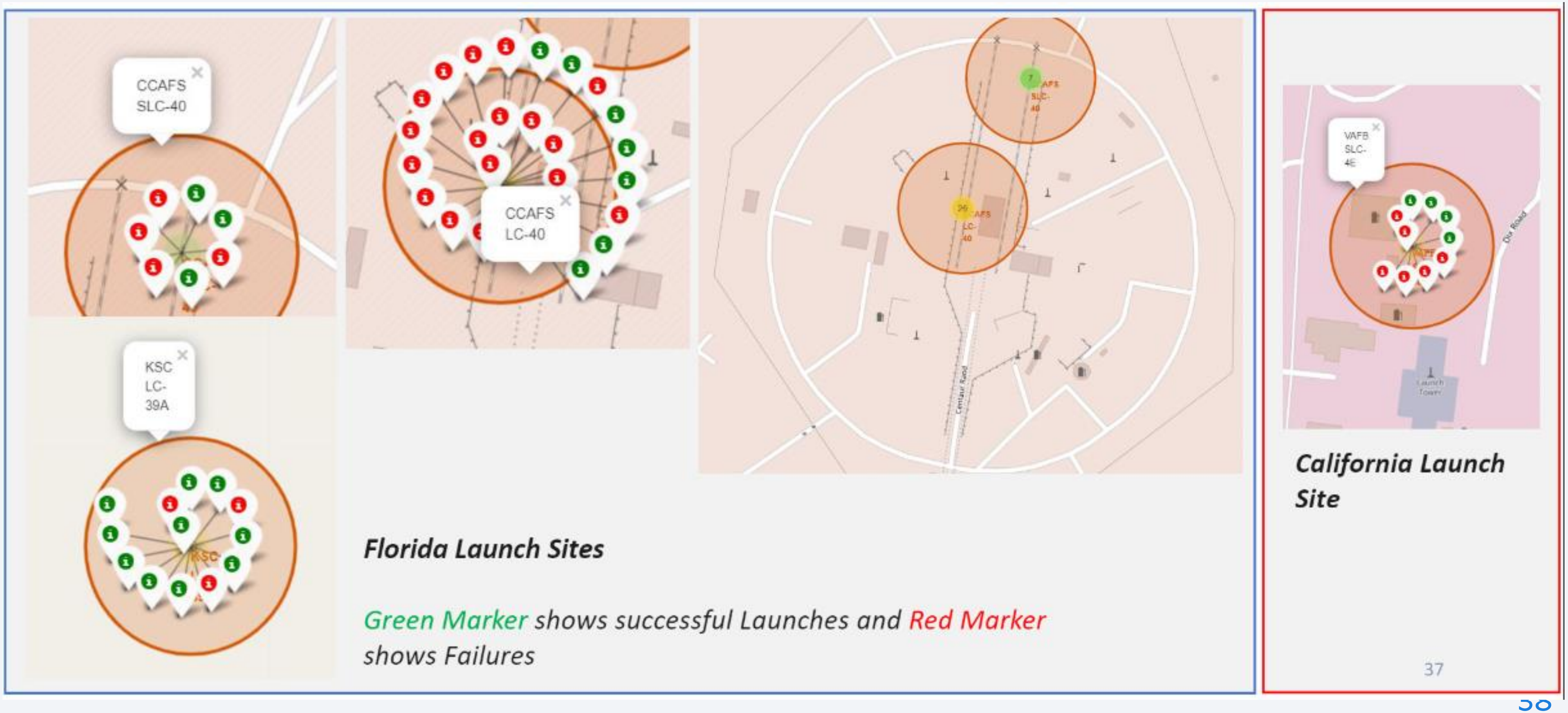
Launch Sites Proximities Analysis

All launch sites global map markers



We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

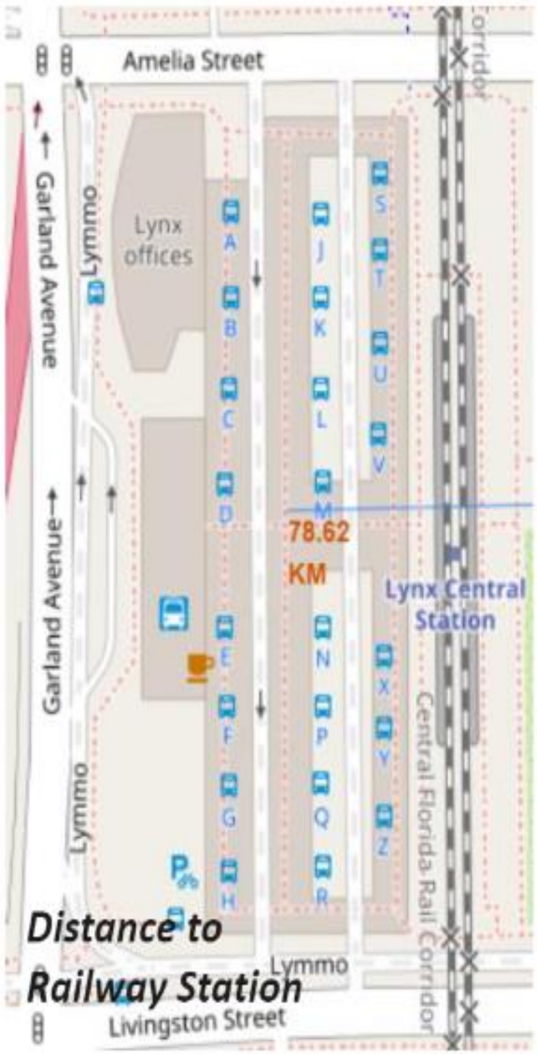


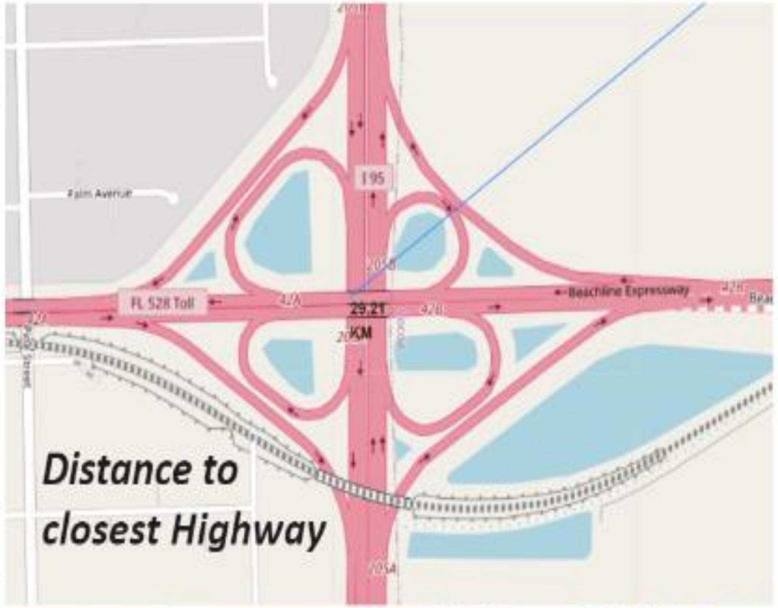
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

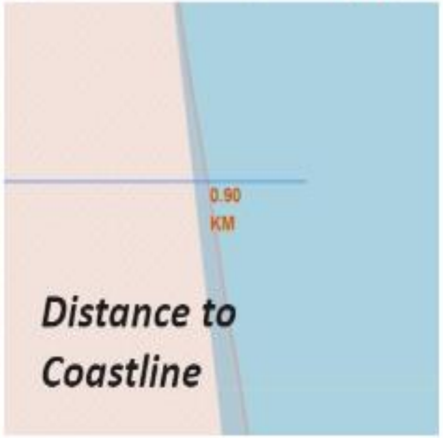
California Launch Site

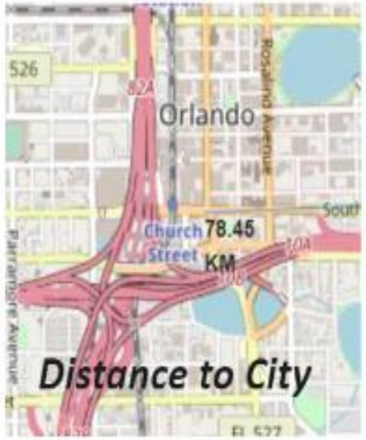
Launch Site distance to landmarks



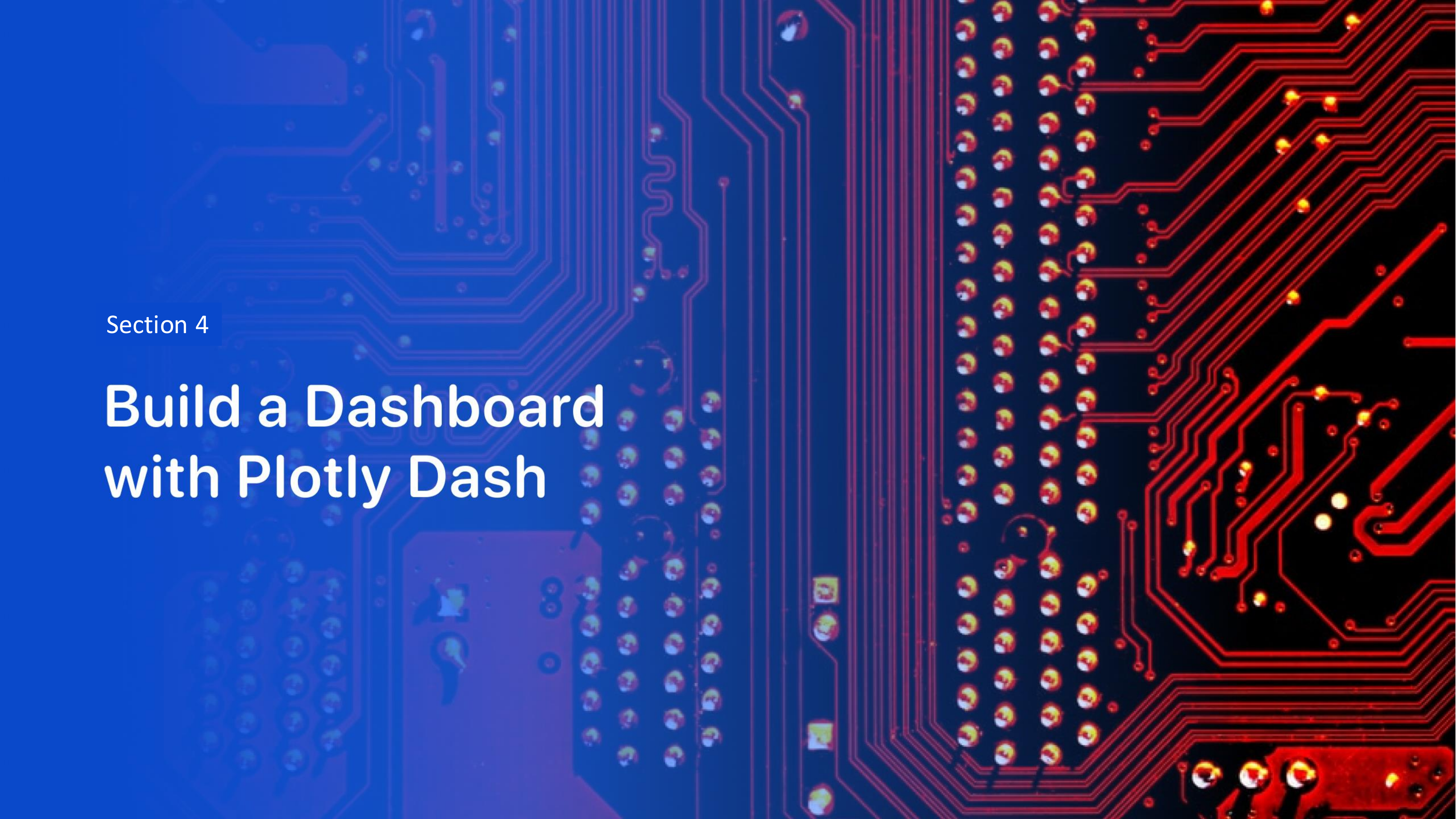


7
CCAFS SLC-40
26
CCAFS LC-40
Distance to coast





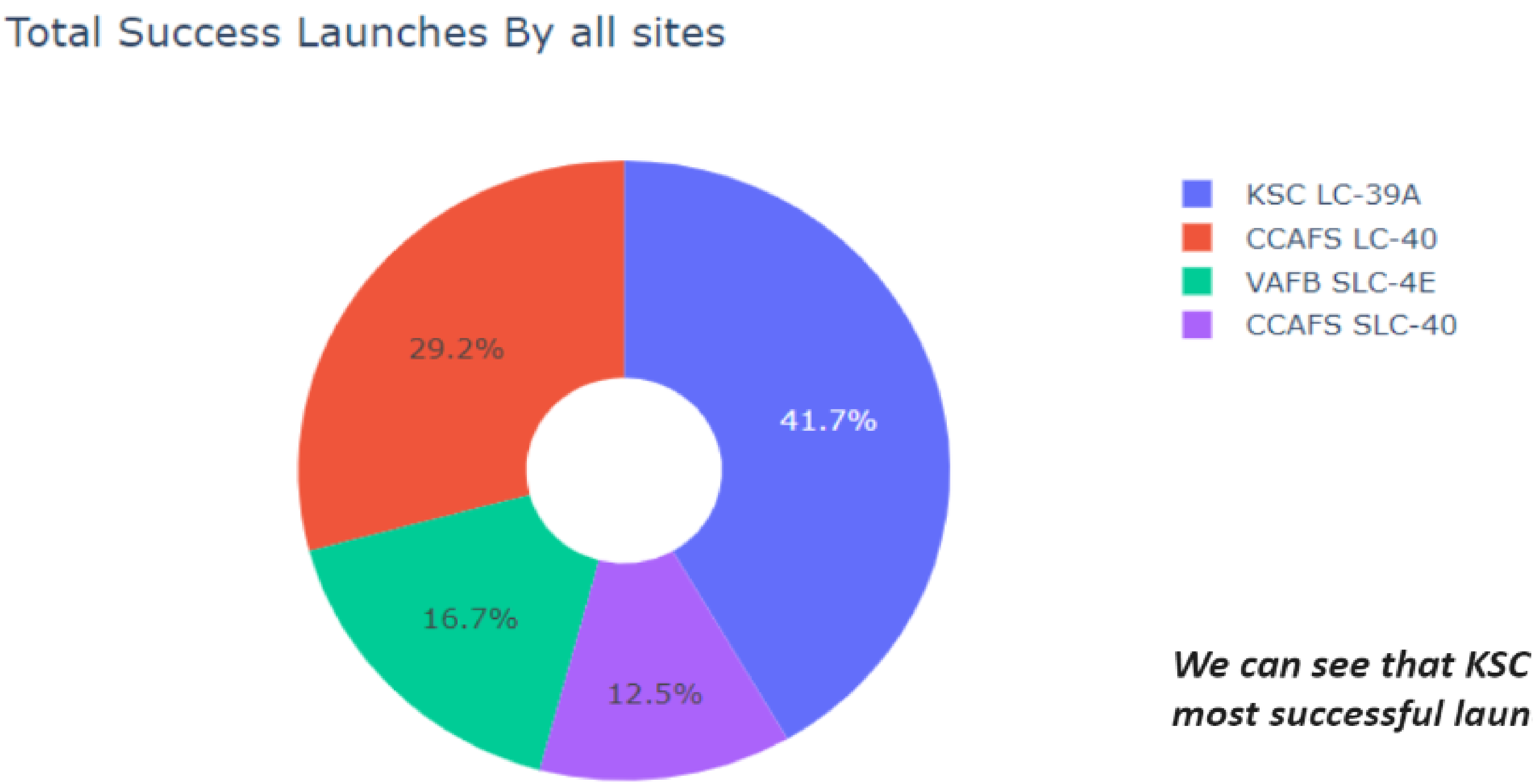
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 4

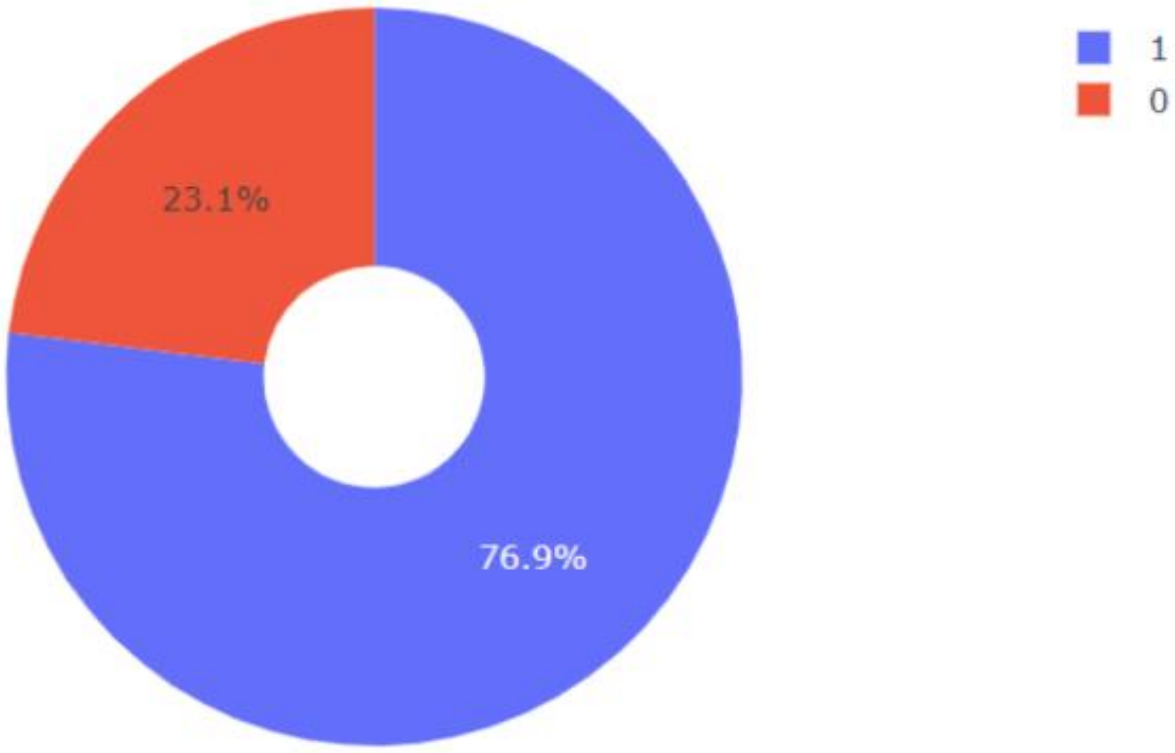
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



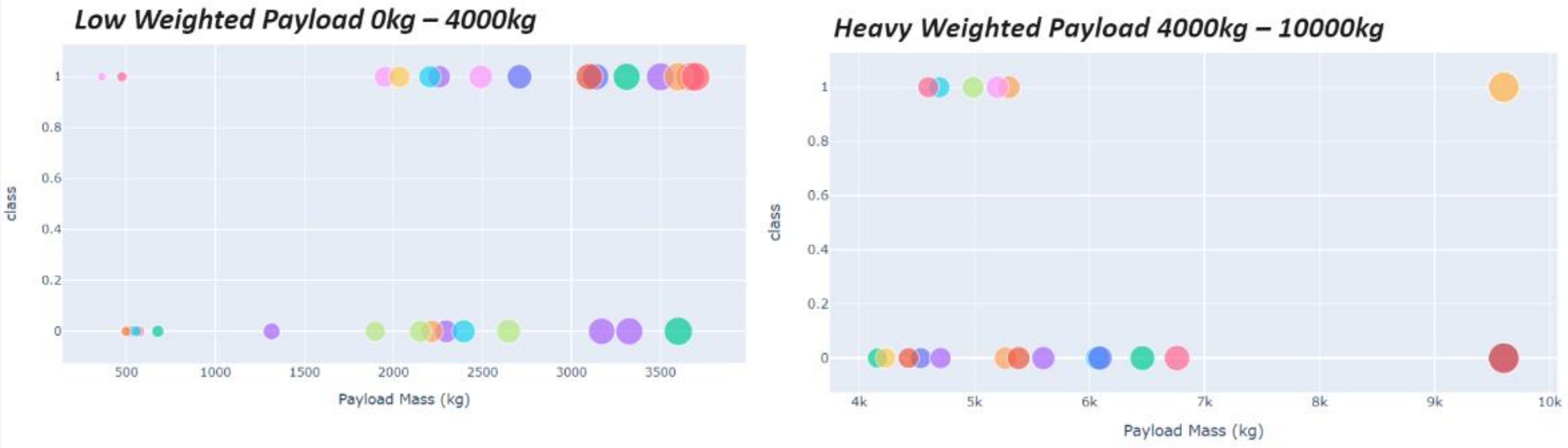
We can see that KSC LC-39A had the most successful launches from all the sites

Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



Low Weighted Payload 0kg – 4000kg

Heavy Weighted Payload 4000kg – 10000kg

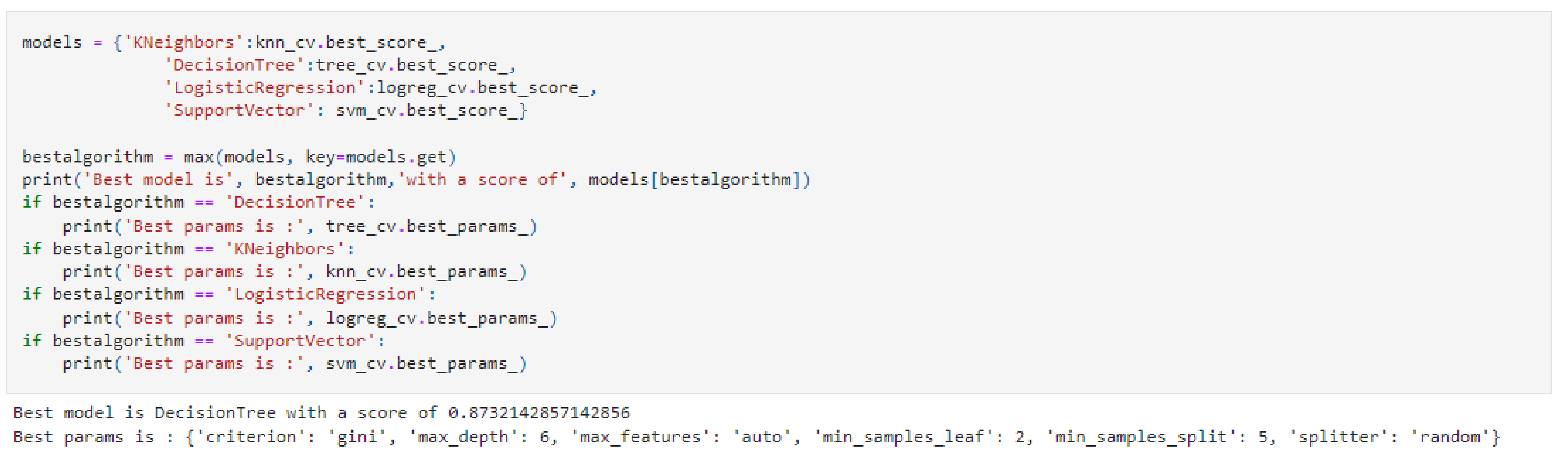
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy



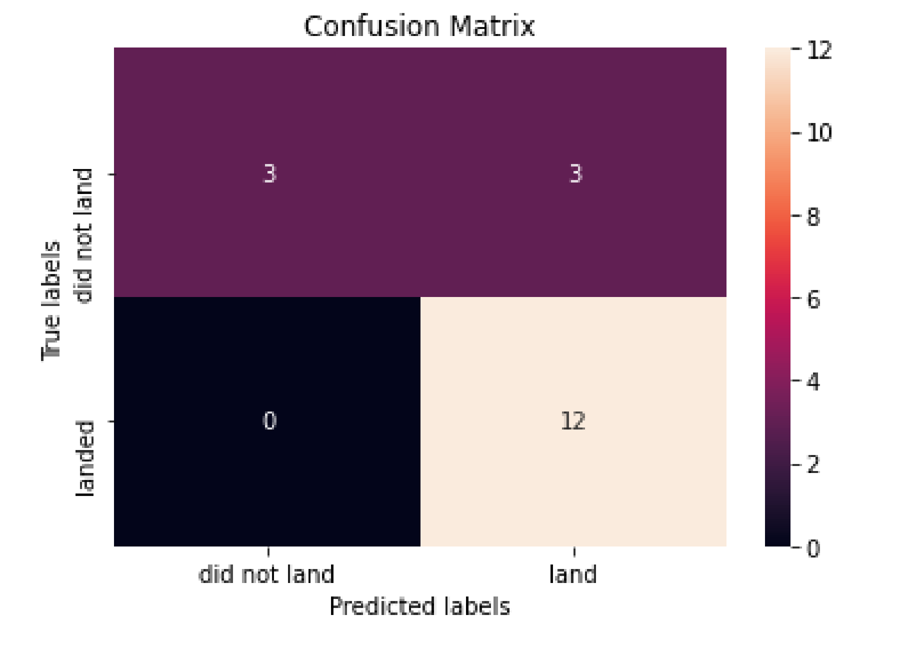
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

The analysis of the confusion matrix for the decision tree classifier indicates that while the classifier is capable of distinguishing between the different classes, there is a notable issue with false positives. Specifically, the classifier tends to incorrectly identify unsuccessful landings as successful ones. This misclassification could lead to an overestimation of the success rate, and addressing this issue is crucial for improving the model's accuracy and reliability. Consider refining the model or adjusting the decision threshold to mitigate these false positives.



Conclusions

1. **Flight Volume and Success**: More flights at a launch site correlate with higher success rates.

2. **Success Rate Increase (2013-2020)**: Launch success rates improved significantly during this period.

3. **Successful Orbits**: ES-L1, GEO, HEO, SSO, and VLEO orbits showed the highest success rates.

4. **KSC LC-39A's Excellence**: This site had the most successful launches among all sites.

5. **Decision Tree Classifier**: Identified as the best machine learning algorithm for this task.

Thank you!