IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Linh Le Thuy Nguyen
20th September 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection (API and Web Scraping)
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis
  - Interactive analytics
  - Predictive Analytics

Introduction

- Project background and context

We predicted if Falcon 9 first stage will land successfully. 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems needed solving
  - 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 needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  - SpaceX API

  - Web scraping from Wikipedia.

- Perform data wrangling

  - One Hot Encoding data fields for Machine Learning and data cleaning and an irrelevant columns

- Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models

  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- The data sets were collected:

- SpaceX launch data is gathered from the SpaceX REST 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

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

– Initialize API: Start with the base URL api.spacexdata.com/v4/.

– Target Endpoint: Choose the specific endpoint /launches/past for past launch data.

– GET Request: Use the requests library to make a GET request.

– Receive JSON: Data is returned as a list of JSON objects, each representing a launch.

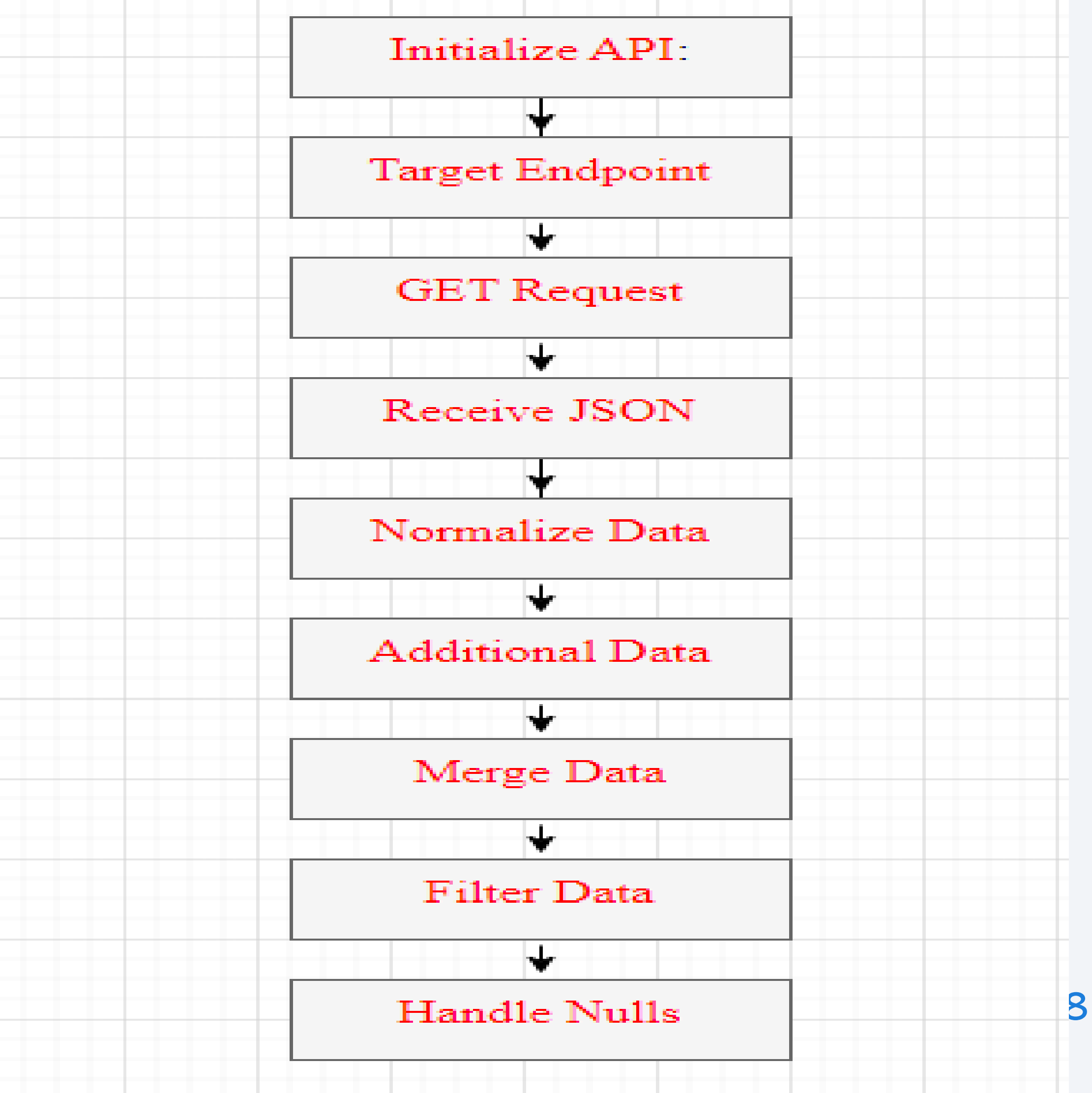
– Normalize Data: Use json_normalize to convert JSON data to a Pandas DataFrame.

– Additional Data: For columns with ID values (e.g., rocket), make additional API calls targeting related endpoints like Booster, Launchpad, payload, etc., to fetch specific data.

– Merge Data: Incorporate the retrieved data from the additional API calls into the main dataset.

– Filter Data: Retain only Falcon 9 launch data, removing Falcon 1 entries.

– Handle Nulls: Address null values in columns PayloadMass were replaced by the value of mean PayloadMass

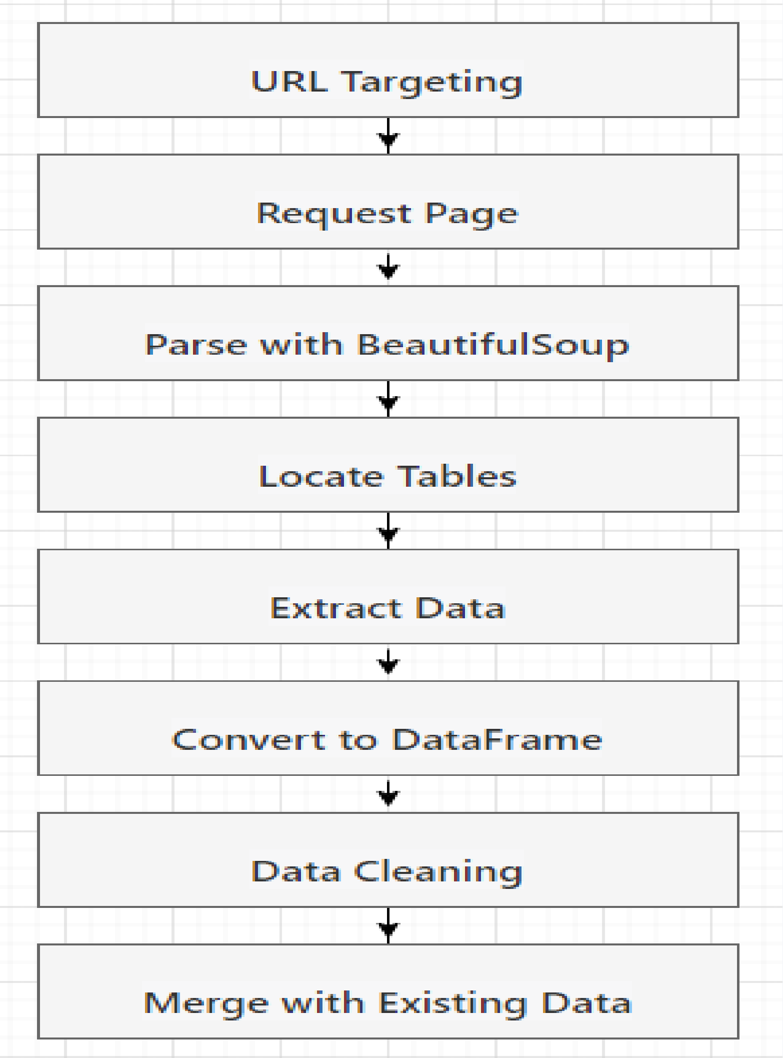


https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

– **URL Targeting**: Identify the specific webpage (Wiki pages related to Falcon 9 launches) for scraping.
– **Request Page**: Use the **requests** library to fetch the webpage's HTML content.
– **Parse with BeautifulSoup**: Utilize **BeautifulSoup** to parse the fetched HTML.
– **Locate Tables**: Identify and locate HTML tables containing Falcon 9 launch records.
– **Extract Data**: Traverse through rows and columns of the tables to extract relevant data.
– **Convert to DataFrame**: Transform the scraped data into a Pandas DataFrame for ease of analysis.
– **Data Cleaning**: Handle any inconsistencies, irrelevant data, or formatting issues.
– **Merge with Existing Data**: If needed, combine the scraped data with other datasets (e.g., from the API).



https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- How data were processed:

1.Data Loading:

•Data related to SpaceX launches is loaded from an external CSV file into a Pandas DataFrame.

2.Initial Data Exploration:

- Conducted an initial examination of the dataset:
- Determined the percentage of null values in each column.
- Checked the data types of the columns.
- Analyzed the distribution of values in specific columns like **LaunchSite** and **Orbit**.

3.Outcome Analysis and Classification:

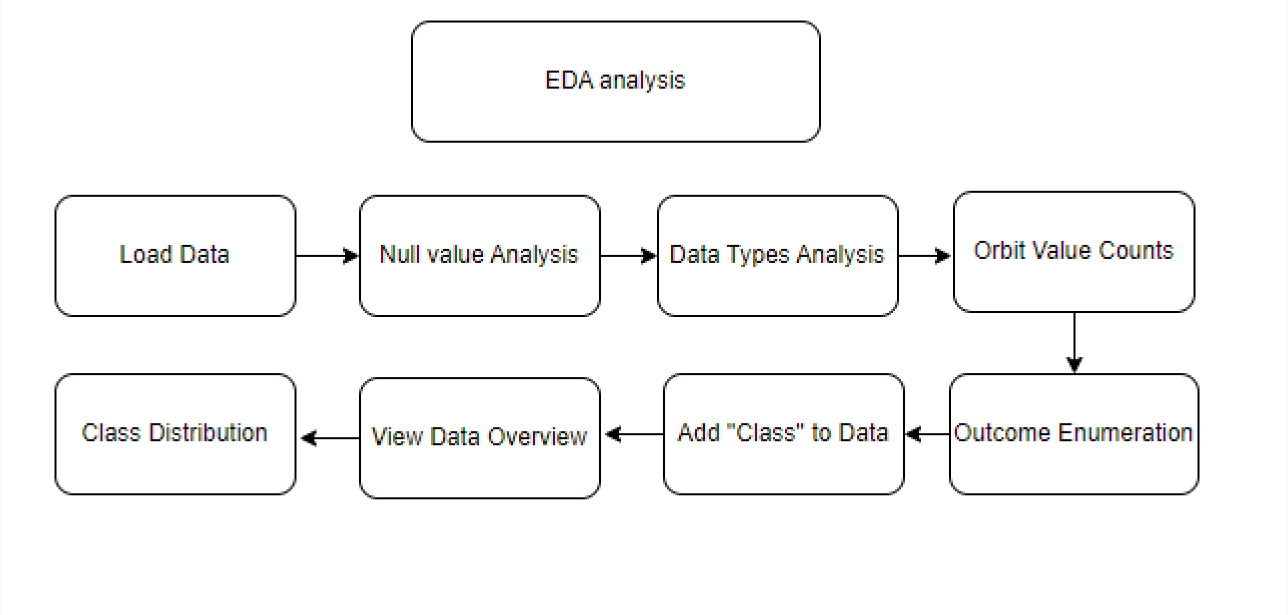
- Identified and enumerated the unique landing outcomes from the **Outcome** column.
- Proposed a classification mechanism to categorize landing outcomes as either successful (Class = 1) or unsuccessful (Class = 0).
- This classification was based on a predefined set of "bad outcomes".

4.Data Augmentation:

- Incorporated the derived classification into the dataset by adding a new column named **Class**.

5.Data Overview and Analysis:

- Displayed a segment of the dataset to understand its structure after the addition of the **Class** column.
- Calculated the mean of the **Class** column to gauge the proportion of successful vs unsuccessful outcomes in the dataset.

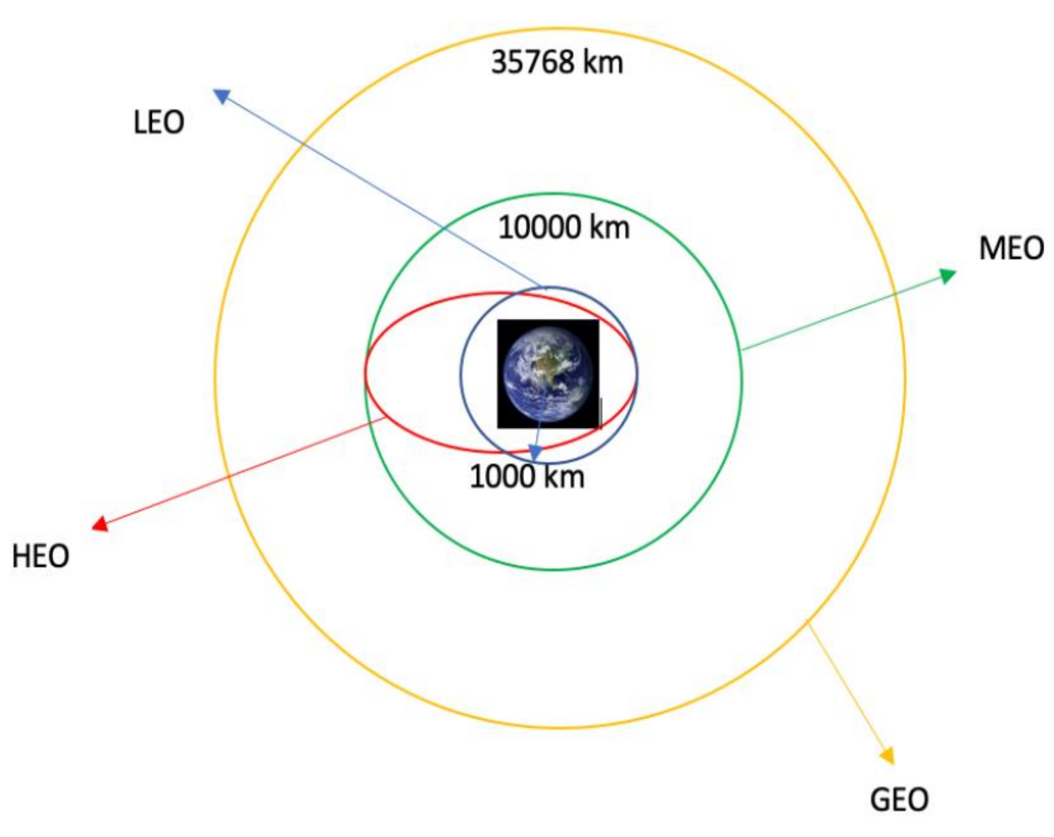


https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

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.

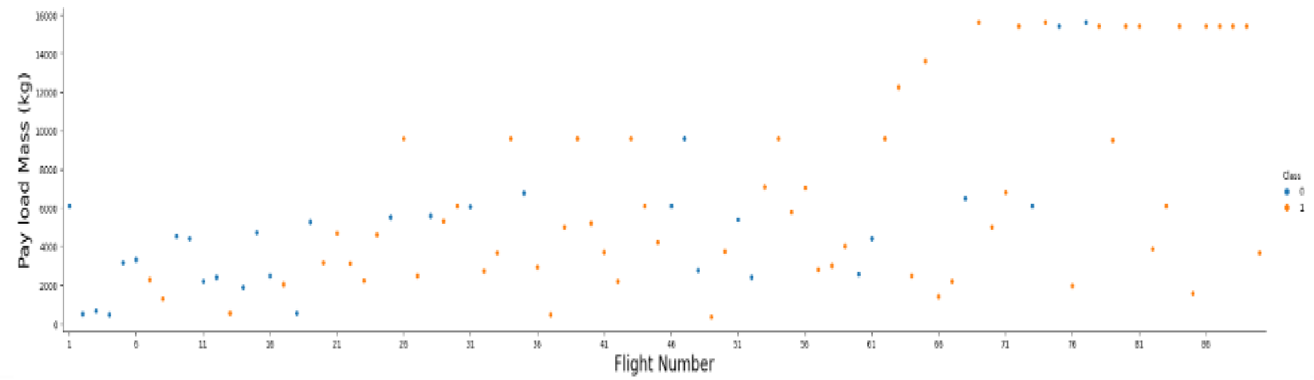
https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

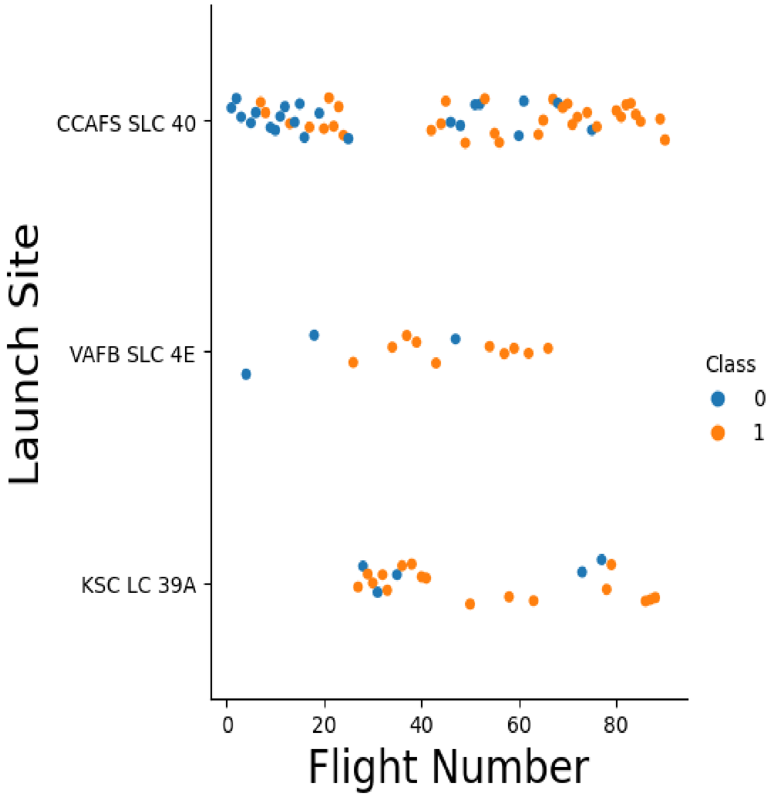


EDA with Data Visualization

- 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.

https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/jupyter-labs-eda-dataviz.ipynb





EDA with Data Visualization

- We first started by using scatter graph to find the relationship between the attributes such as between:
- Payload Mass and Flight number
- Flight number and Launch Site
- Payload Mass and Launch Site
- Flight Number and Orbit type
- Payload Mass and Orbit type
- Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graph. It is easy to see which factors affecting the most to the success of the landing outcomes
- Once we get a hint of the relationship using scatter plot. We can the use further visualization such as bar graph, line plots graph to understand more about data
- Bar graph is one of the easiest way to interpret the relationship between the attributes. In this case, we use the bar graph to determine which orbits have the higest probability of success
- Then, using Feature Engineering to be used in success prediction in the future module by create the dummy variables to categorial columns

https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/jupyter-labs-eda-dataviz.ipynb

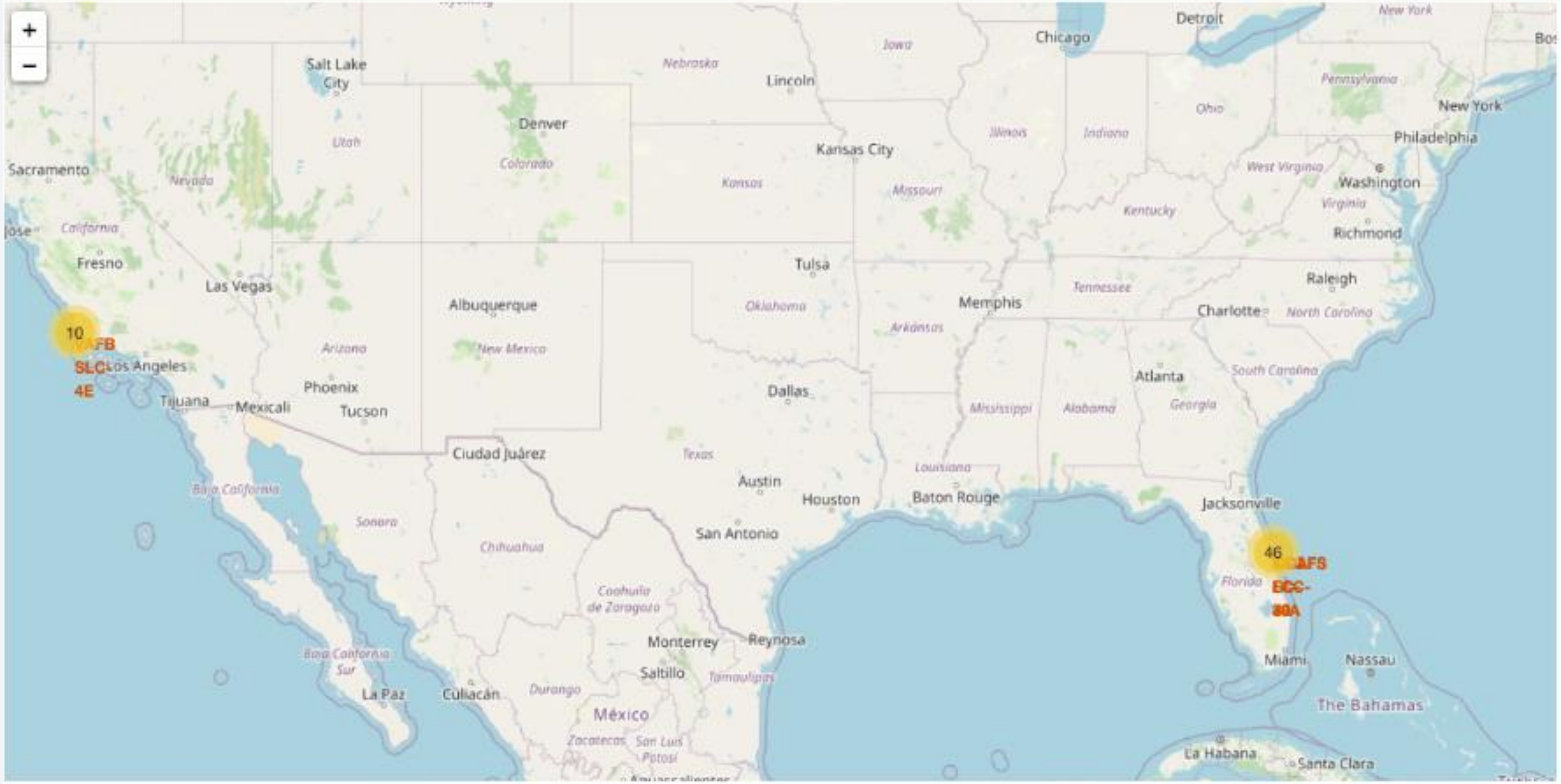
EDA with SQL

The SQL queries performed the following points:

- Display the names of the unique launch sites in the space mission
- Display 5 records where lauch sites begin with thestring 'CCA'
- Display the total payload mass carried by boosters lauched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achived
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and lauch site names for the years 2015
- Rank the count of landing outcomes (such as Failure (drone ship) of Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium



Map makers have been added to the map with the aim to find an aptimal location for buiding a launch site

https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/lab_jupyter_launch_site_location.ipynb

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:
  - How close the launch sites with railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?

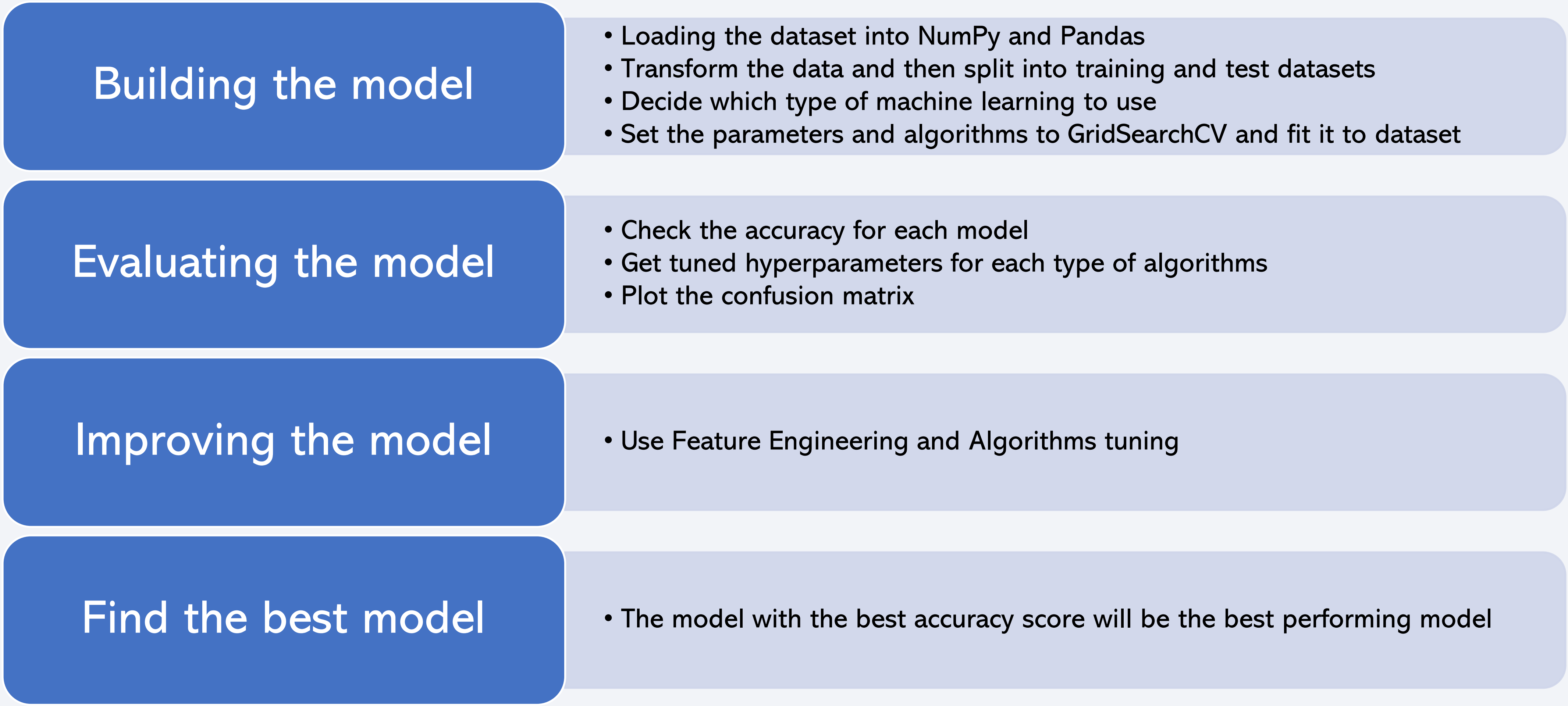
https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/lab_jupyter_launch_site_location.ipynb

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.

https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/spacex_dash_app1.py

Predictive Analysis (Classification)



Building the model

- Loading the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of machine learning to use
- Set the parameters and algorithms to GridSearchCV and fit it to dataset

Evaluating the model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot the confusion matrix

Improving the model

- Use Feature Engineering and Algorithms tuning

Find the best model

- The model with the best accuracy score will be the best performing model

https://github.com/linhnguyenlethuy1611/linh_datascience/blob/main/Predictive_Analysis_-_Machine_Learning_Lab_.ipynb

Results

The results will be categorized to 3 parts:

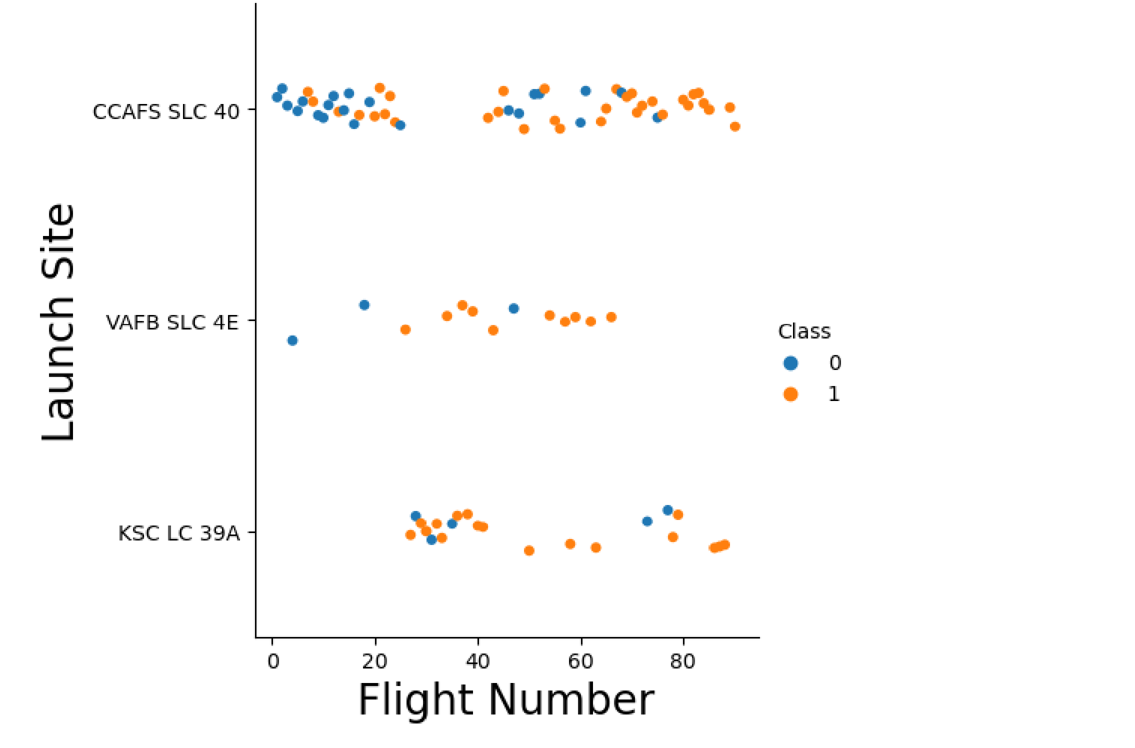
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

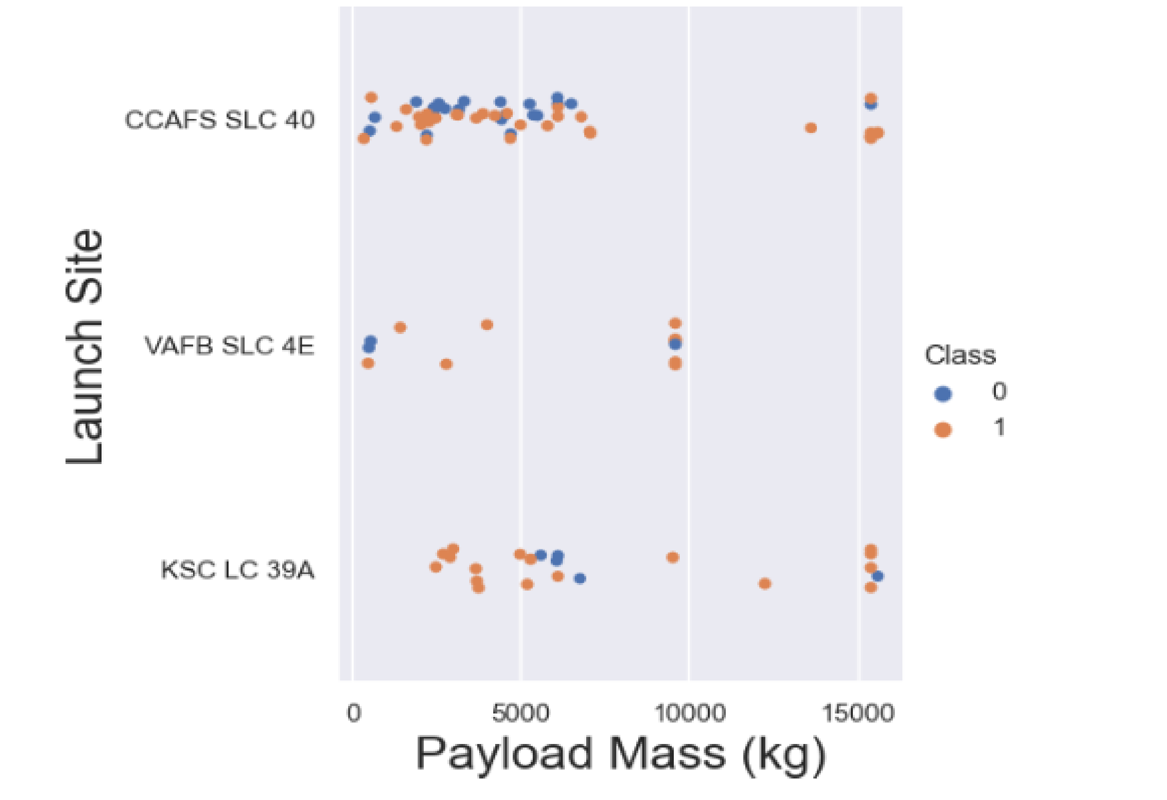
This scatter plot shows that the larger the flights amount of the launch site, the greater the success rate will be However site CCAFS SLC40 shows the least pattern of this



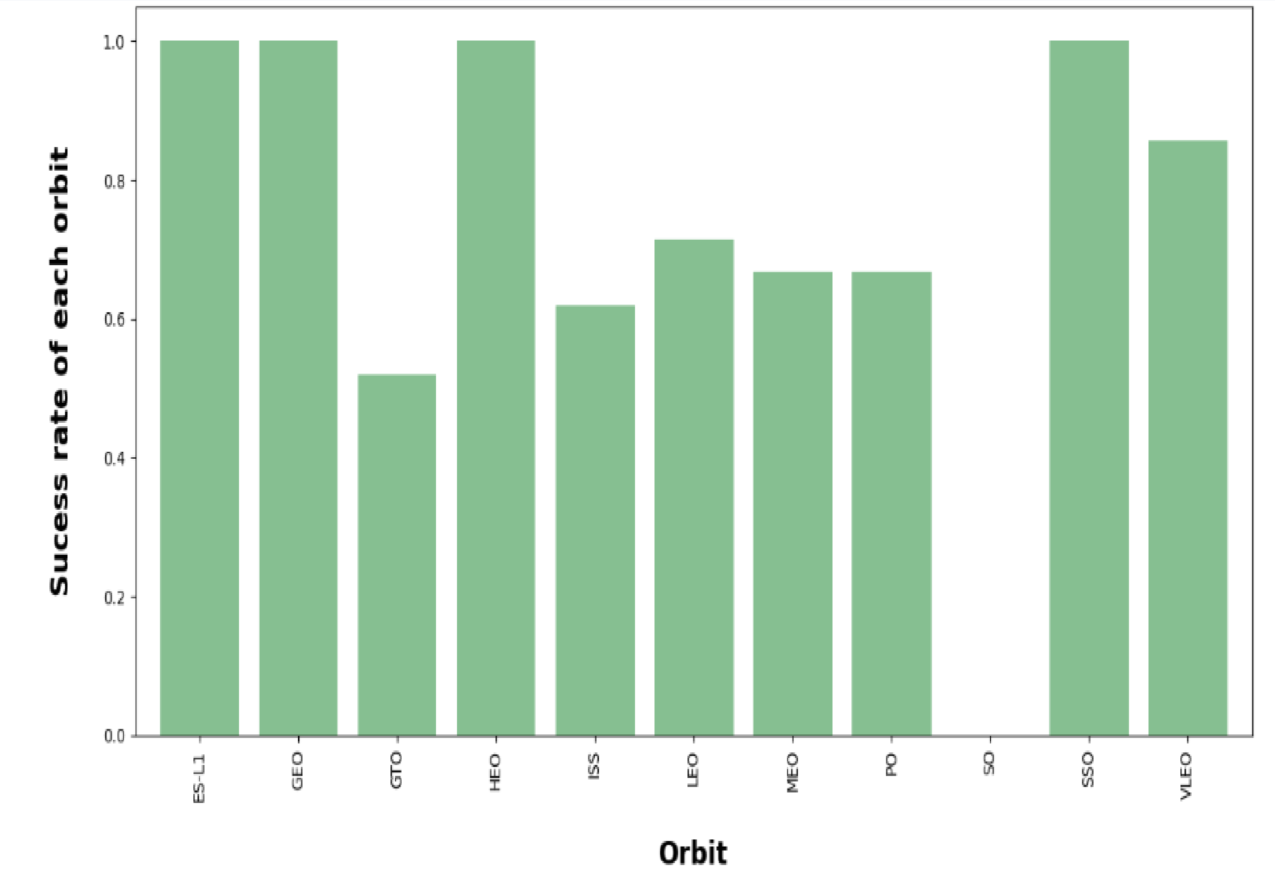
Payload vs. Launch Site

This scatter plot shows that when the payload mass is greater than 7000kg, the probability of the success rate will be highly increased

There isn’t clear pattern to show that the success rate of launch site depends on the payload mass.



Success Rate vs. Orbit Type



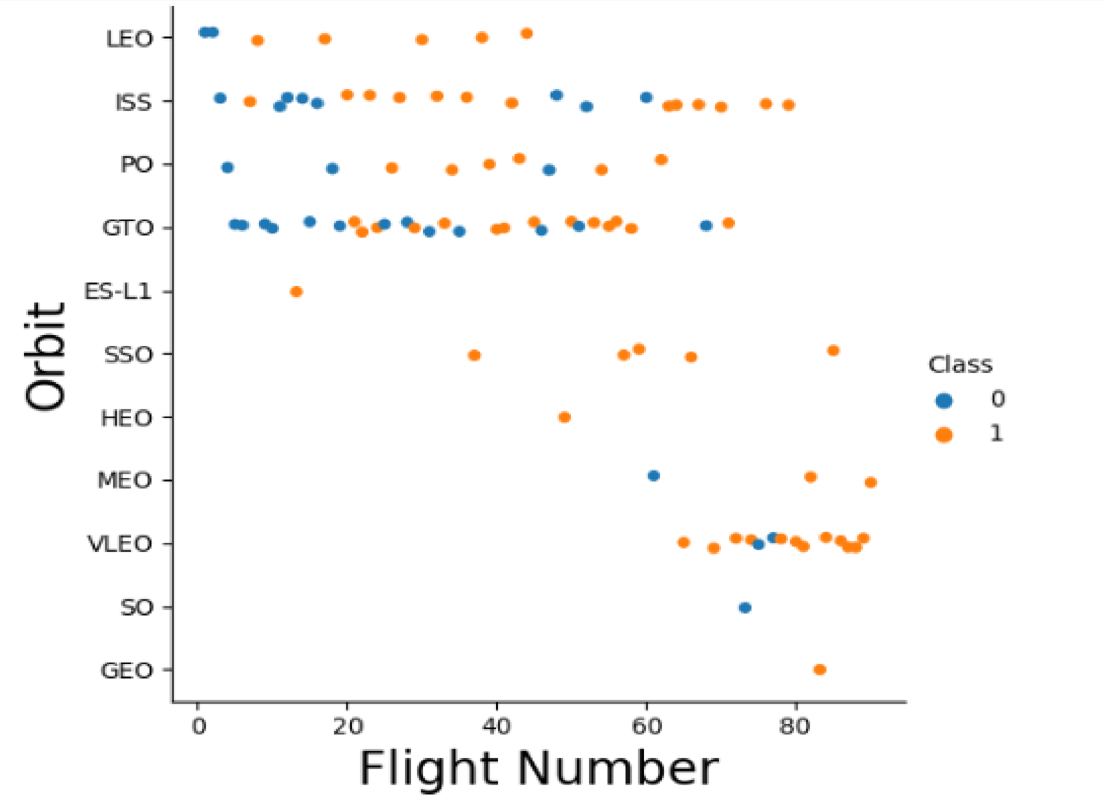
- This graph shows the coralation between type of orbits and their success rate.

- ES-L1, GEO, HEO and SSO have a success rate of 100%, however, SO has a success rate of 0%

- To deeper understanding, we need more dataset to know the trend or pattern of the orbits types

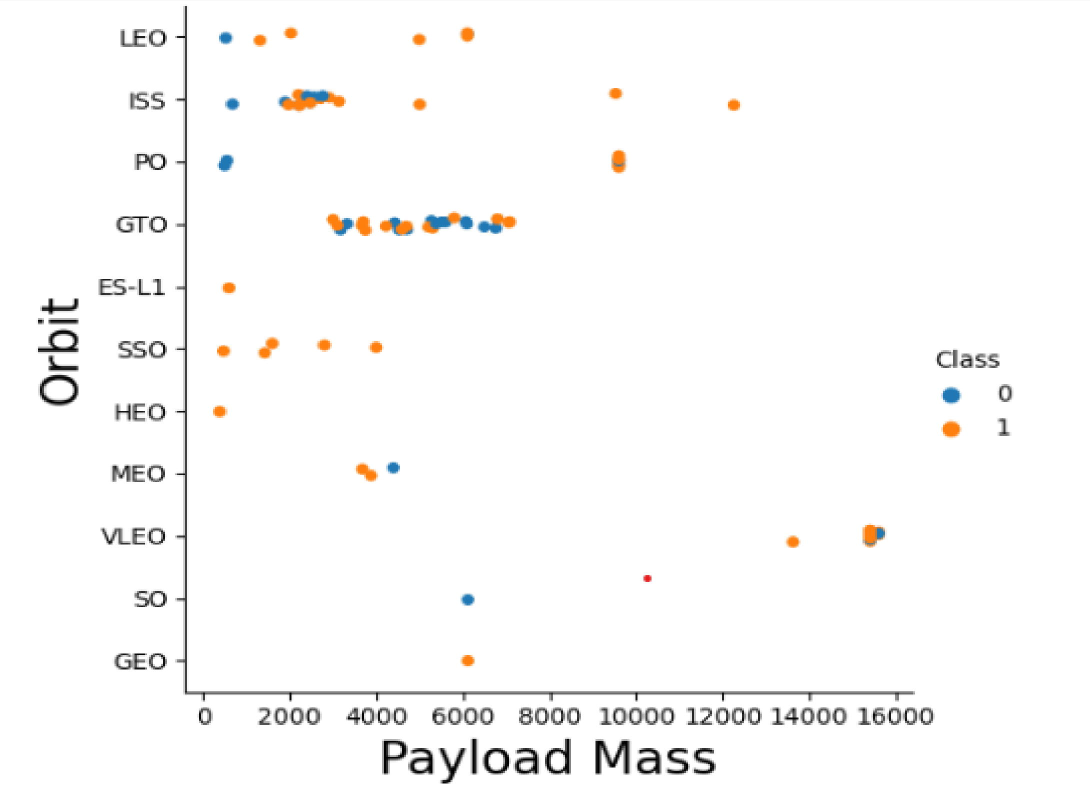
Flight Number vs. Orbit Type

- The larger the flight numbers on each orbits, the greater the success rate especially LEO.
- There is not relationship between Flight number and Orbit
- Orbit that has only q occurrence maybe given the less accuracy results, however, we need more dataset for conclusion



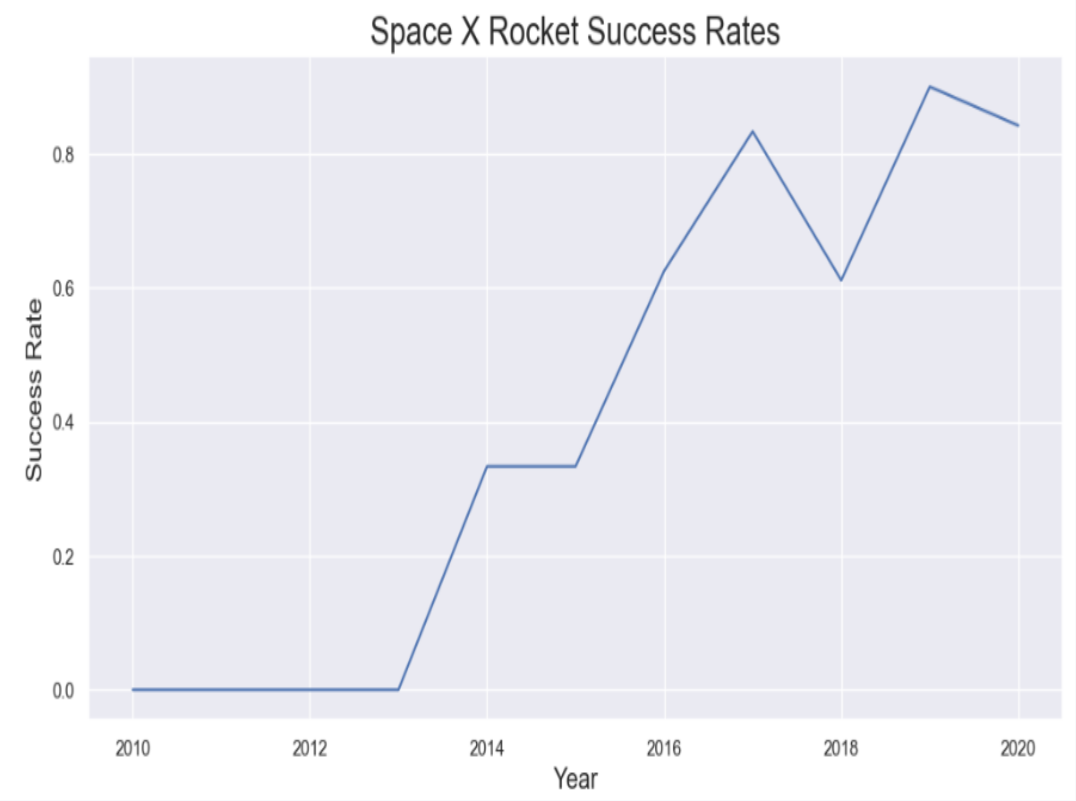
Payload vs. Orbit Type

- Heavier payload mass has positive impact on LEO, ISS and PO orbit, but it has negative impact on MEO and VLEO orbit.
- There is not relation between orbit type and Payload mass for GTO orbit
- Meanwhile, SO, GEO, HEO orbit need more data to understand their pattern



Launch Success Yearly Trend

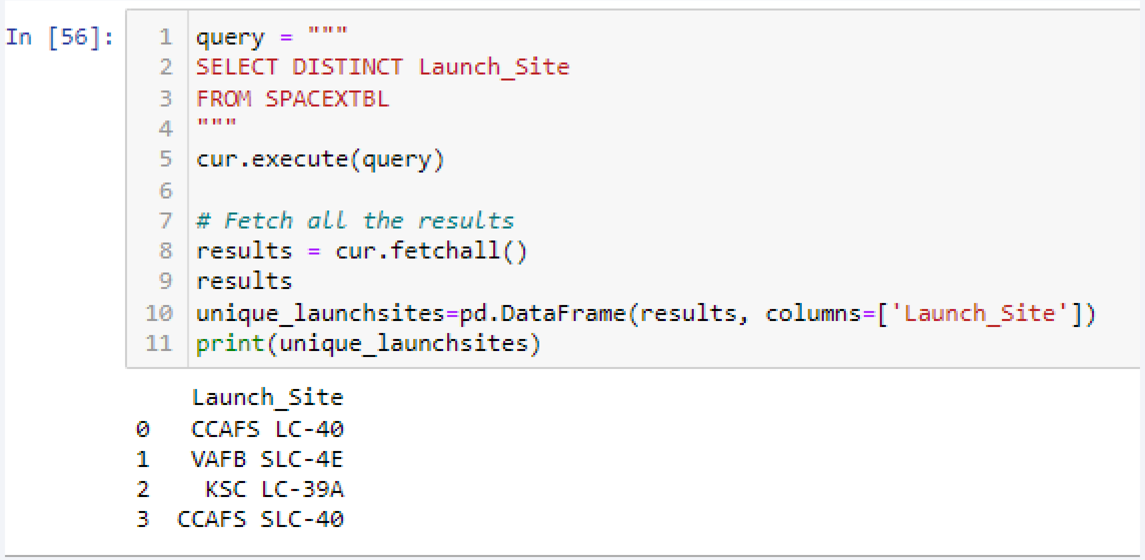
- There is an increasing trend from the year of 2013 to 2020 for Space X Rocket Success Rate, although in 2018, the success rate have a slightly decrease.



All Launch Site Names

- Launch Site names

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

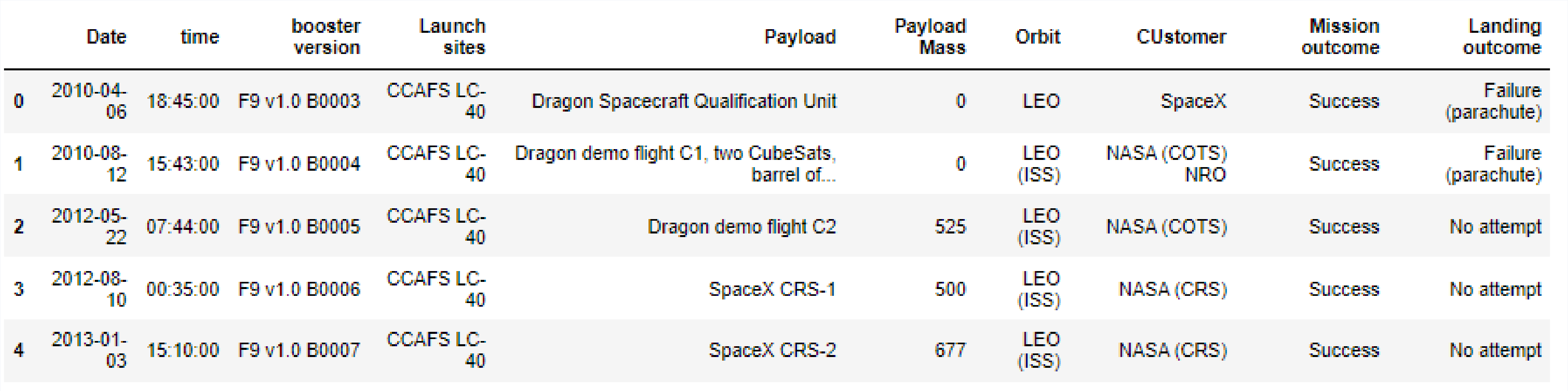


```
query = """
SELECT DISTINCT Launch_Site
FROM SPACEXTBL
"""
cur.execute(query)

# Fetch all the results
results = cur.fetchall()
results
unique_launchsites=pd.DataFrame(results, columns=['Launch_Site'])
print(unique_launchsites)
```

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

Launch Site Names Begin with 'CCA'

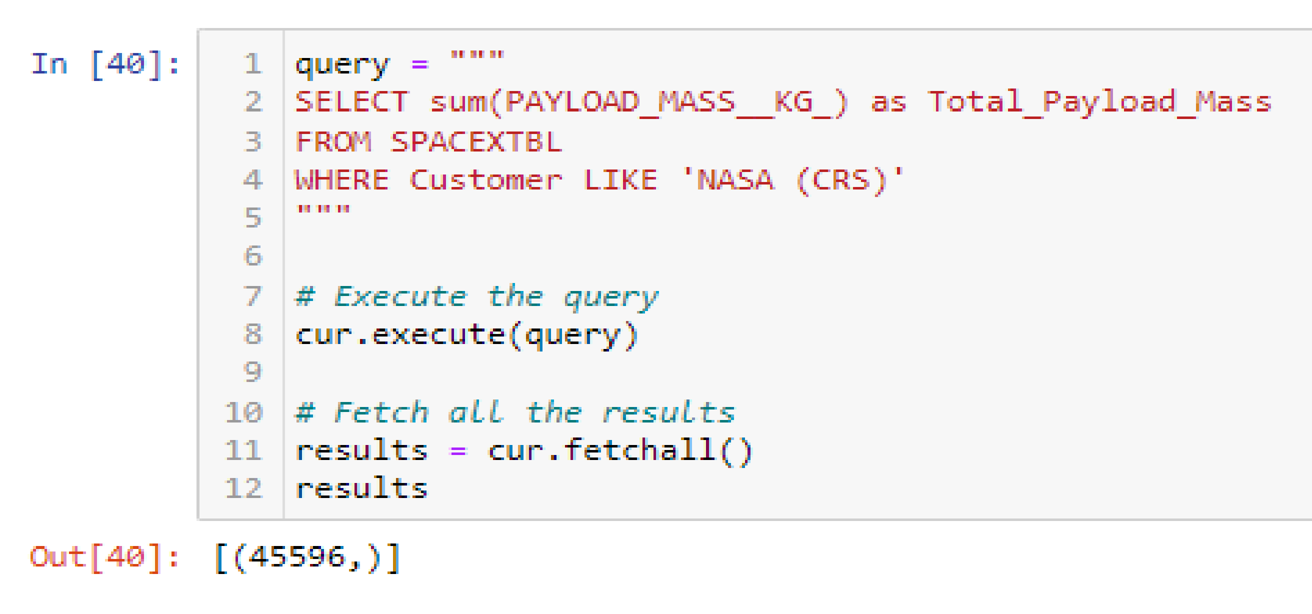
- Five records where launch sites begin with `CCA`



	Date	time	booster version	Launch sites	Payload	Payload Mass	Orbit	CUstomer	Mission outcome	Landing outcome
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



In [40]:

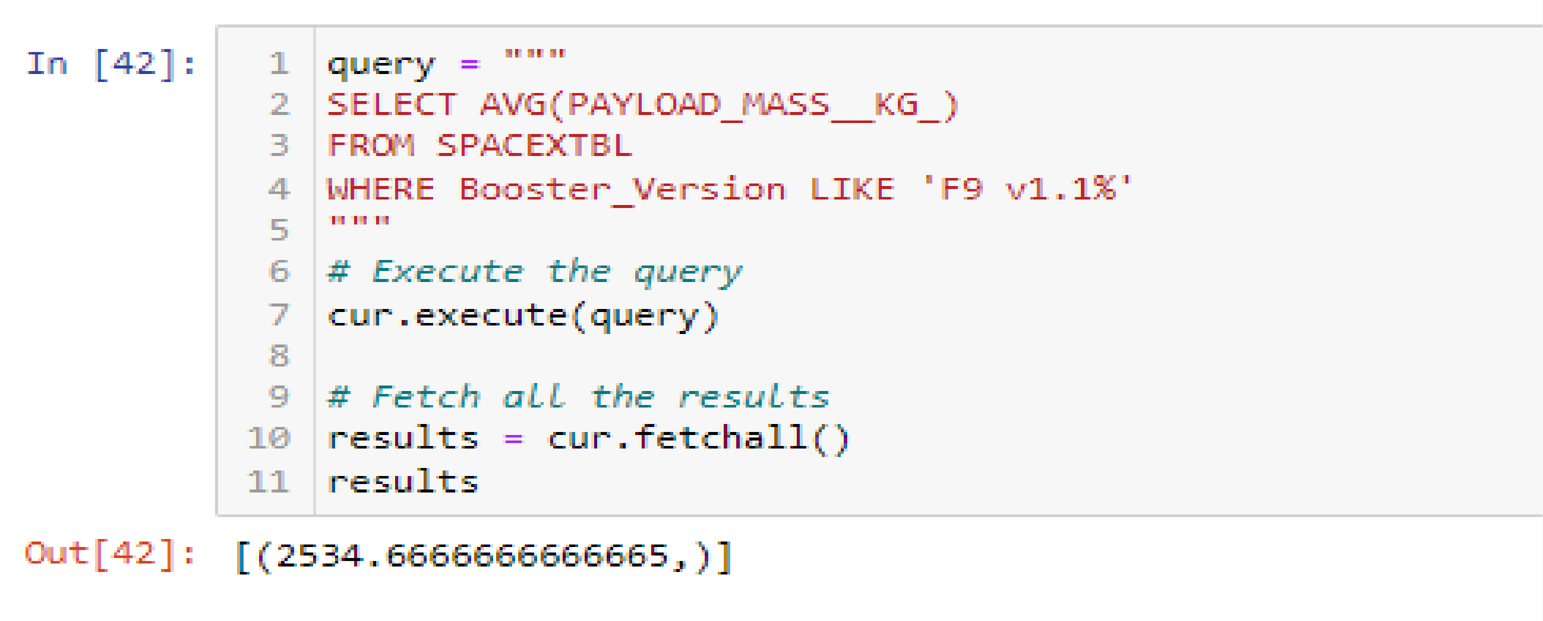
```
query = """
SELECT sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass
FROM SPACEXTBL
WHERE Customer LIKE 'NASA (CRS)'
"""

# Execute the query
cur.execute(query)

# Fetch all the results
results = cur.fetchall()
results
```

Out[40]:

```
[(45596,)]
```

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1



In [42]:

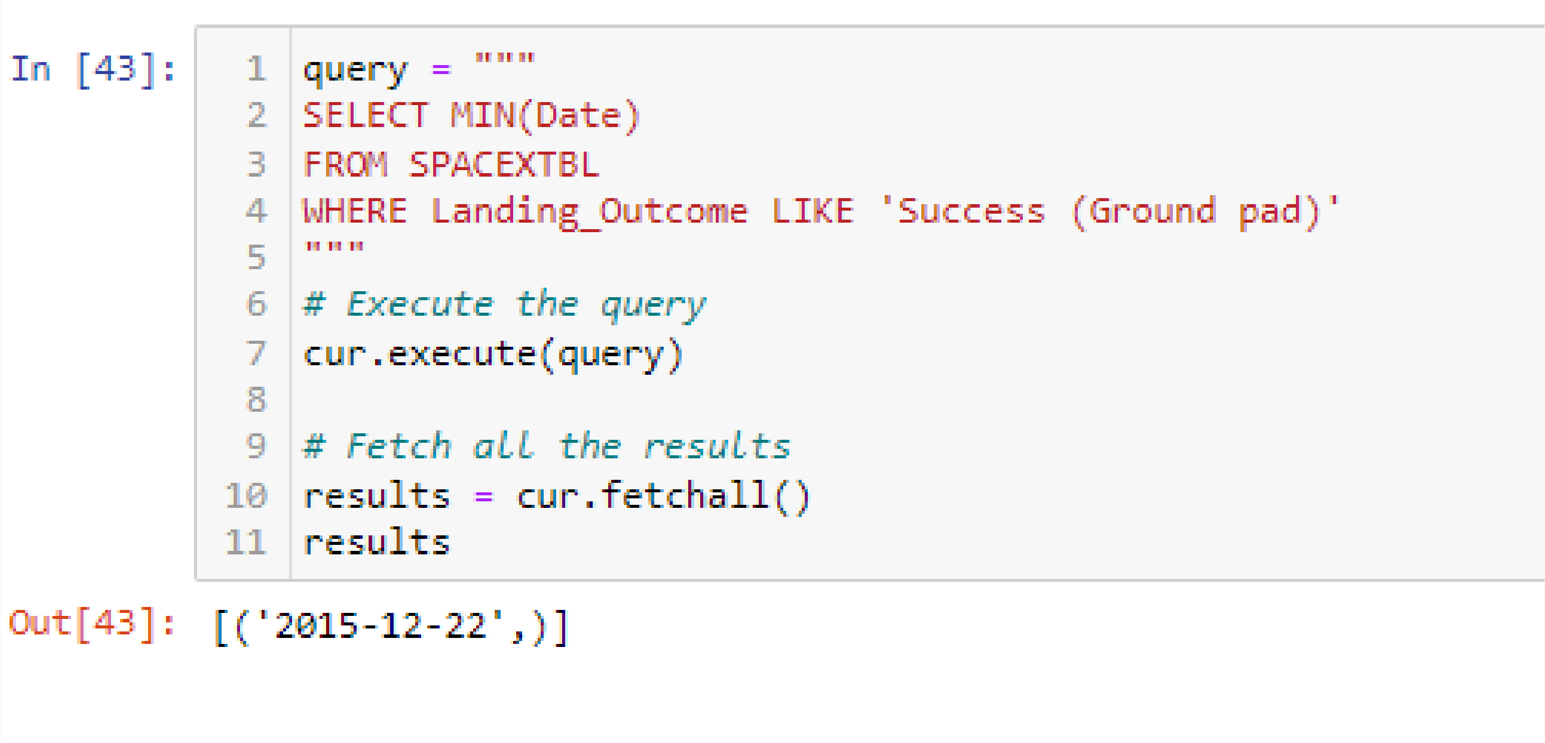
```
query = """
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1%'
"""
# Execute the query
cur.execute(query)

# Fetch all the results
results = cur.fetchall()
results
```

Out[42]:

```
[(2534.6666666666665,)]
```

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad

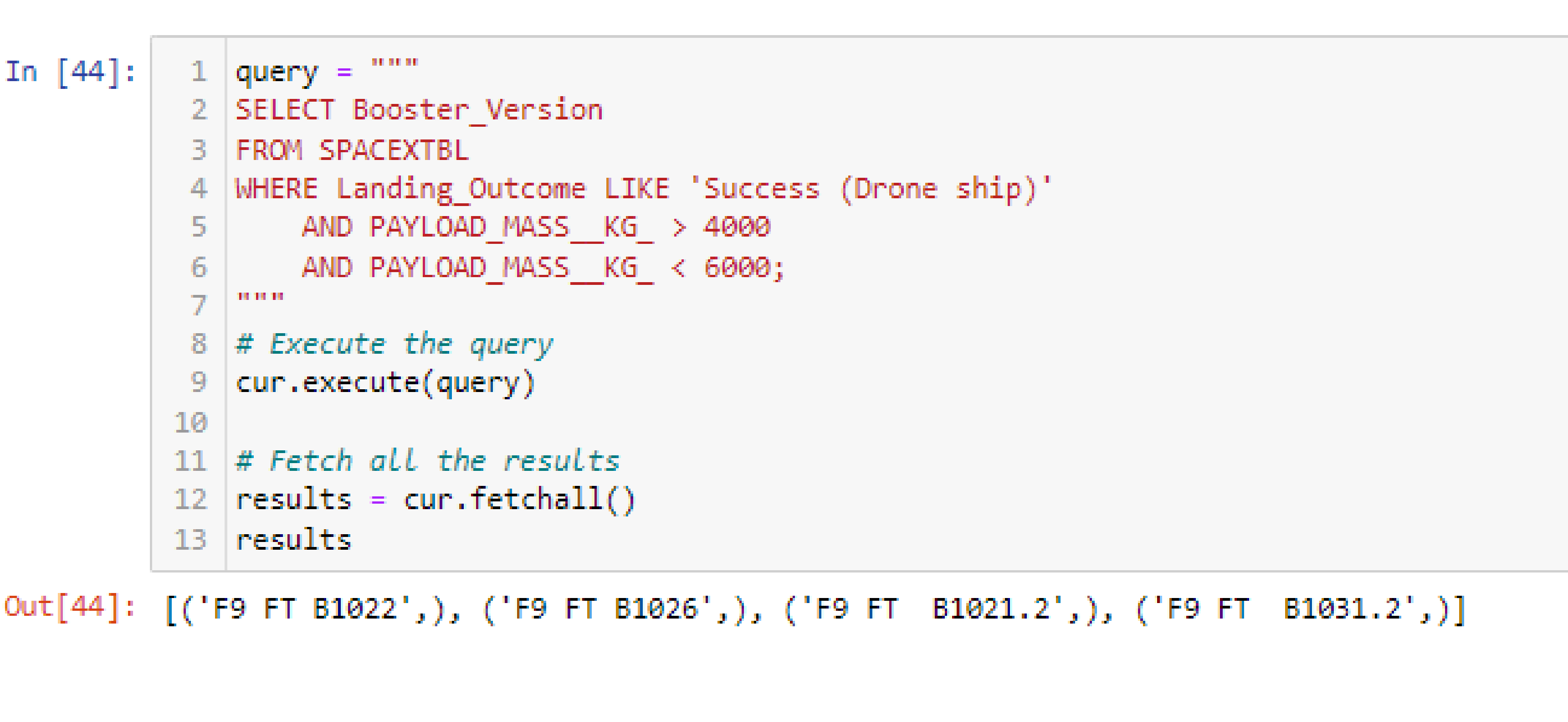


```
In [43]:
query = """
SELECT MIN(Date)
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Success (Ground pad)'
"""
# Execute the query
cur.execute(query)

# Fetch all the results
results = cur.fetchall()
results
```

```
Out[43]: [('2015-12-22',)]
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



In [44]:

```
query = """
SELECT Booster_Version
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Success (Drone ship)'
    AND PAYLOAD_MASS__KG_ > 4000
    AND PAYLOAD_MASS__KG_ < 6000;
"""
# Execute the query
cur.execute(query)

# Fetch all the results
results = cur.fetchall()
results
```

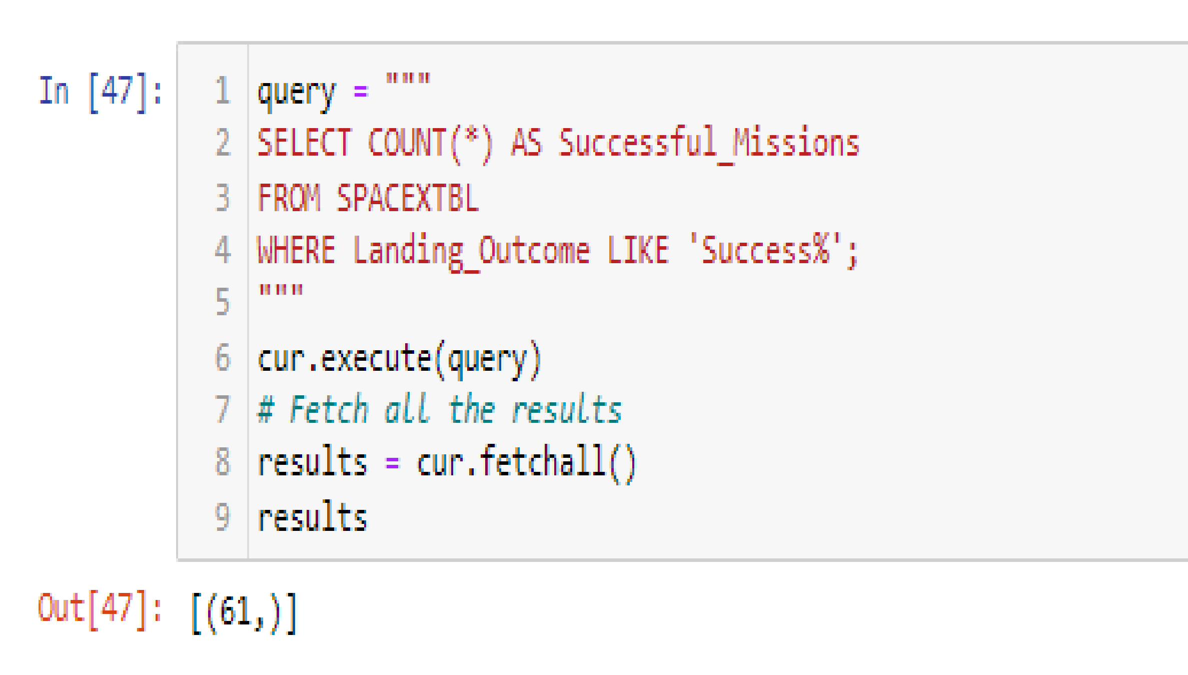
Out[44]:

```
[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

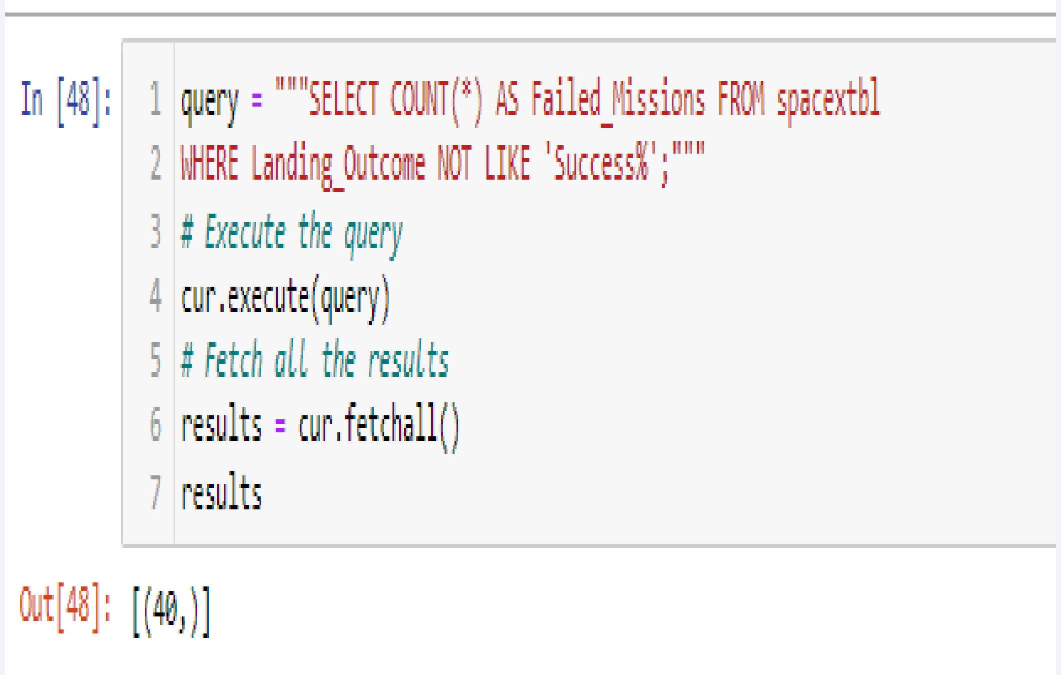
Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes

The total number of failure mission outcomes



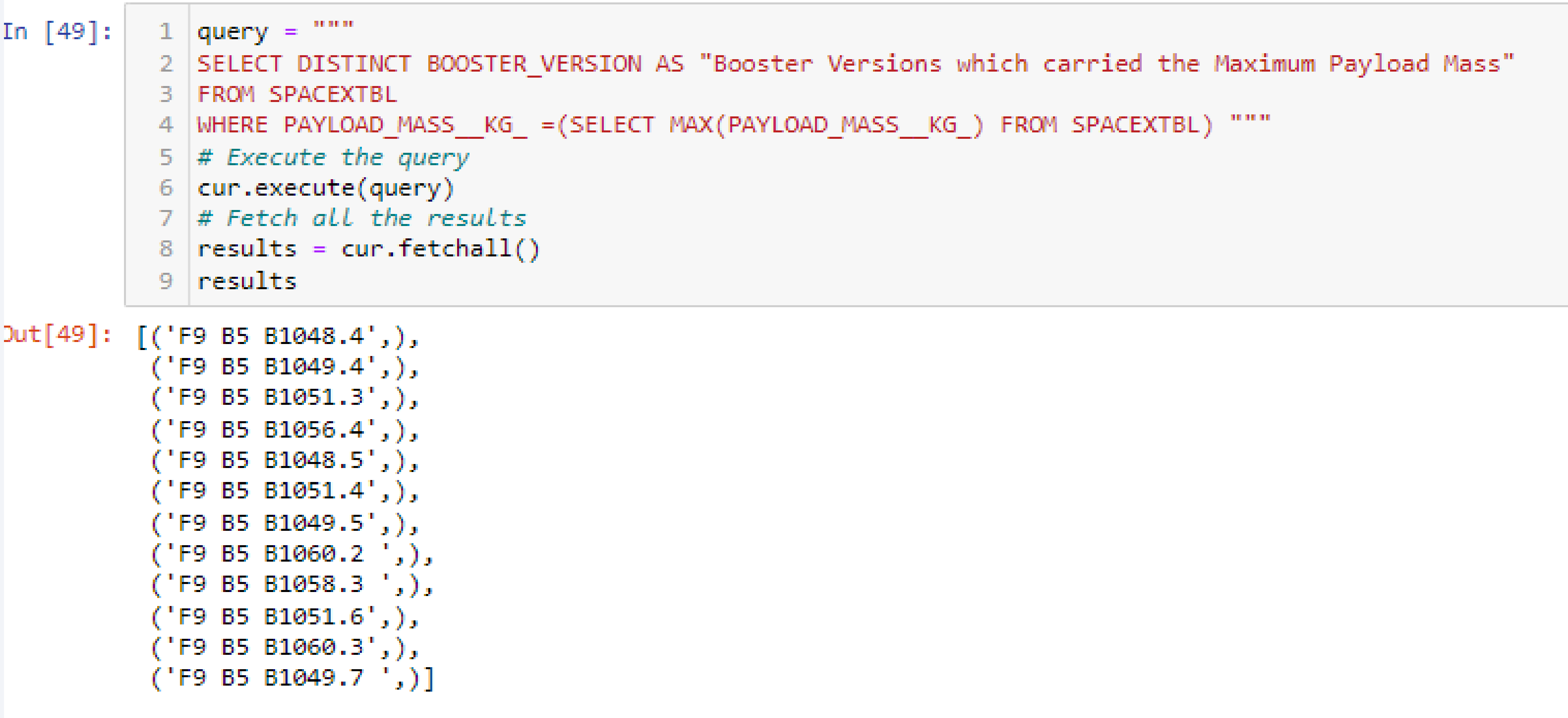
```
In [47]: query = """
         SELECT COUNT(*) AS Successful_Missions
         FROM SPACEXTBL
         WHERE Landing_Outcome LIKE 'Success%';
         """
         cur.execute(query)
         # Fetch all the results
         results = cur.fetchall()
         results

Out[47]: [(61,)]
```



```
In [48]: query = """SELECT COUNT(*) AS Failed_Missions FROM spacextbl
         WHERE Landing_Outcome NOT LIKE 'Success%';"""
         # Execute the query
         cur.execute(query)
         # Fetch all the results
         results = cur.fetchall()
         results

Out[48]: [(40,)]
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

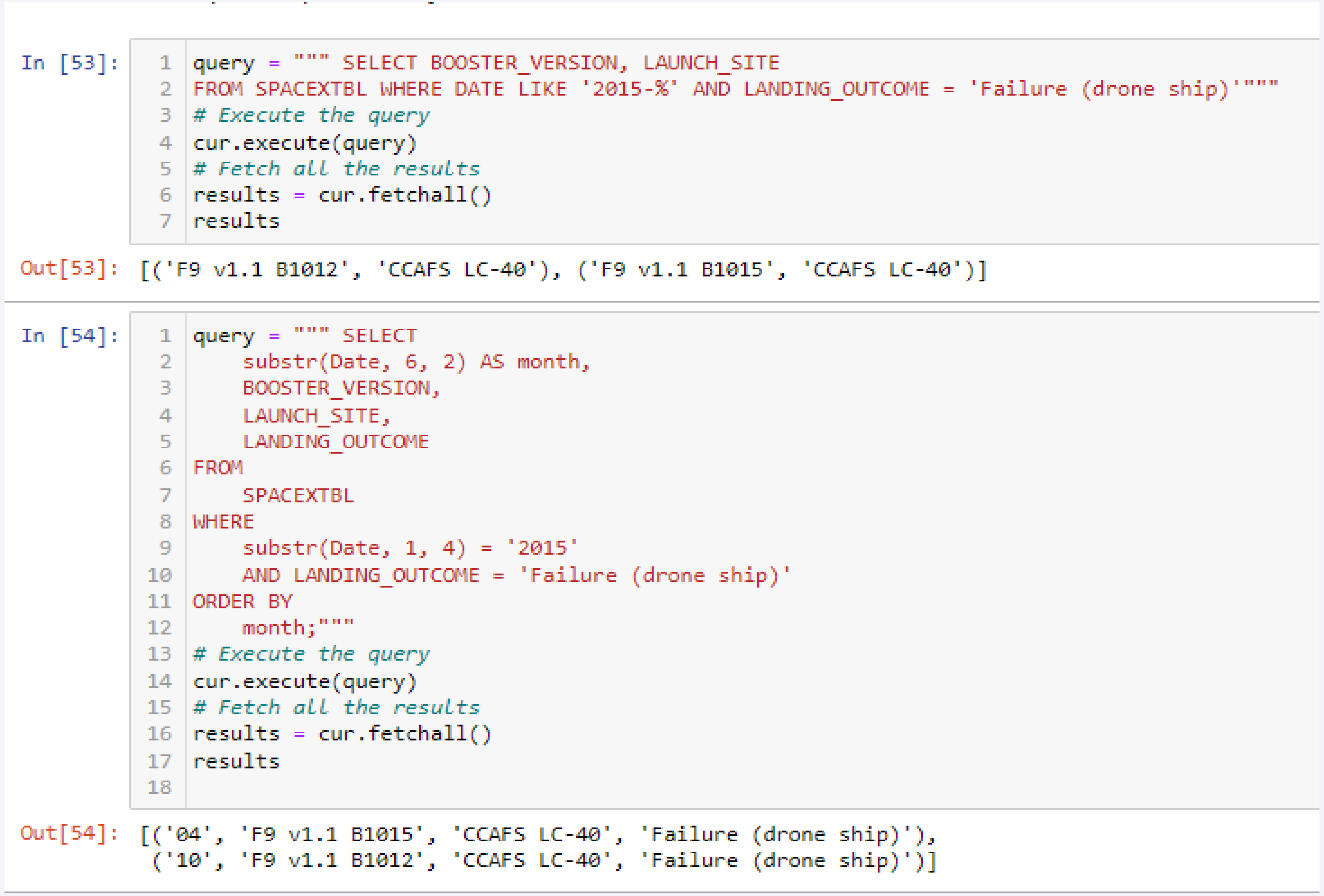


```
In [49]:
query = """
SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass"
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) """
# Execute the query
cur.execute(query)
# Fetch all the results
results = cur.fetchall()
results
```

```
Out[49]: [('F9 B5 B1048.4',),
          ('F9 B5 B1049.4',),
          ('F9 B5 B1051.3',),
          ('F9 B5 B1056.4',),
          ('F9 B5 B1048.5',),
          ('F9 B5 B1051.4',),
          ('F9 B5 B1049.5',),
          ('F9 B5 B1060.2 ',),
          ('F9 B5 B1058.3 ',),
          ('F9 B5 B1051.6',),
          ('F9 B5 B1060.3',),
          ('F9 B5 B1049.7 ',)]
```

2015 Launch Records

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



In [53]:
```
query = """ SELECT BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND LANDING_OUTCOME = 'Failure (drone ship)'"""
# Execute the query
cur.execute(query)
# Fetch all the results
results = cur.fetchall()
results
```

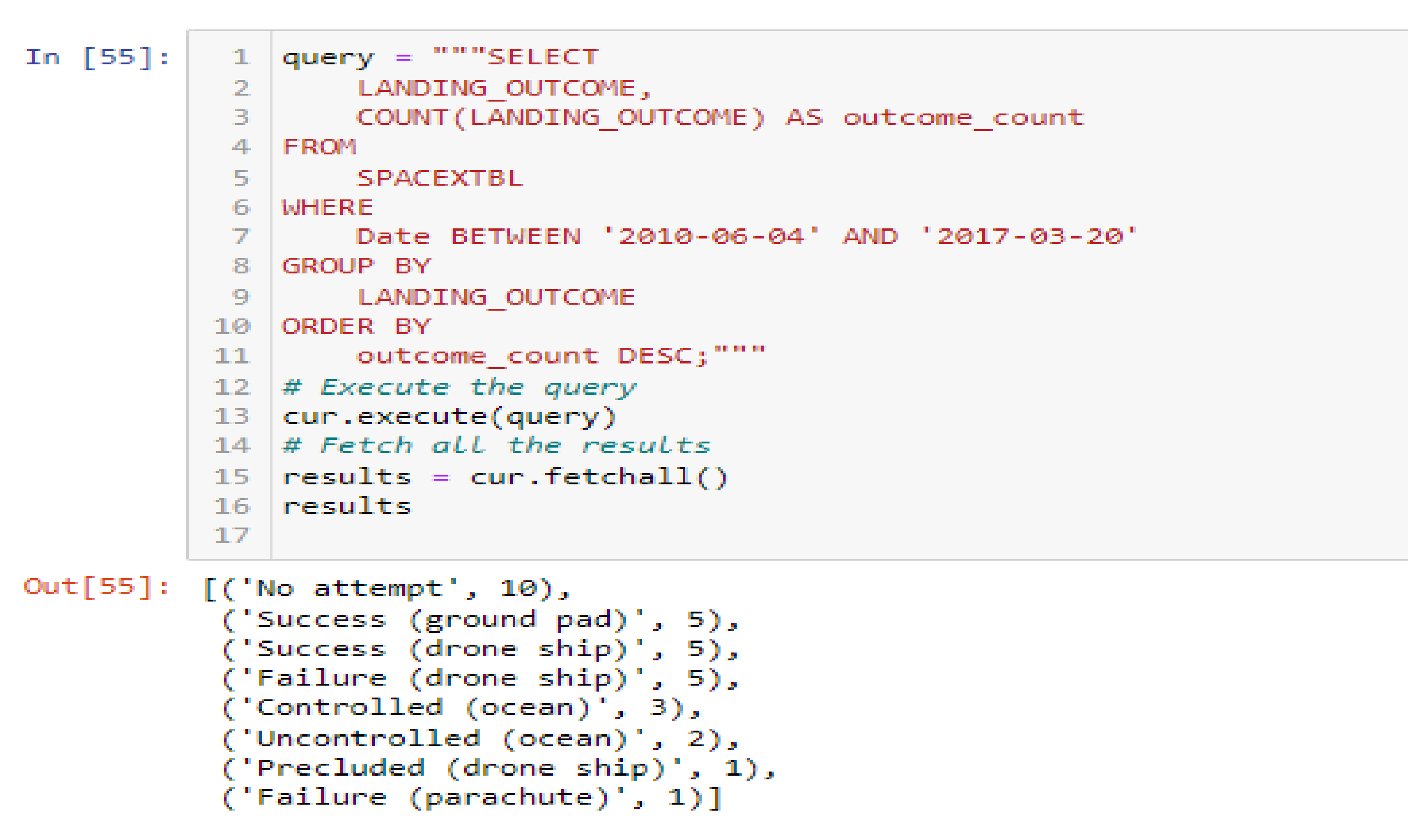
Out[53]:
```
[('F9 v1.1 B1012', 'CCAFS LC-40'), ('F9 v1.1 B1015', 'CCAFS LC-40')]
```

In [54]:
```
query = """ SELECT
    substr(Date, 6, 2) AS month,
    BOOSTER_VERSION,
    LAUNCH_SITE,
    LANDING_OUTCOME
FROM
    SPACEXTBL
WHERE
    substr(Date, 1, 4) = '2015'
    AND LANDING_OUTCOME = 'Failure (drone ship)'
ORDER BY
    month;"""
# Execute the query
cur.execute(query)
# Fetch all the results
results = cur.fetchall()
results

```

Out[54]:
```
[('04', 'F9 v1.1 B1015', 'CCAFS LC-40', 'Failure (drone ship)'),
 ('10', 'F9 v1.1 B1012', '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



In [55]:

```
query = """SELECT
    LANDING_OUTCOME,
    COUNT(LANDING_OUTCOME) AS outcome_count
FROM
    SPACEXTBL
WHERE
    Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    LANDING_OUTCOME
ORDER BY
    outcome_count DESC;"""
# Execute the query
cur.execute(query)
# Fetch all the results
results = cur.fetchall()
results

```

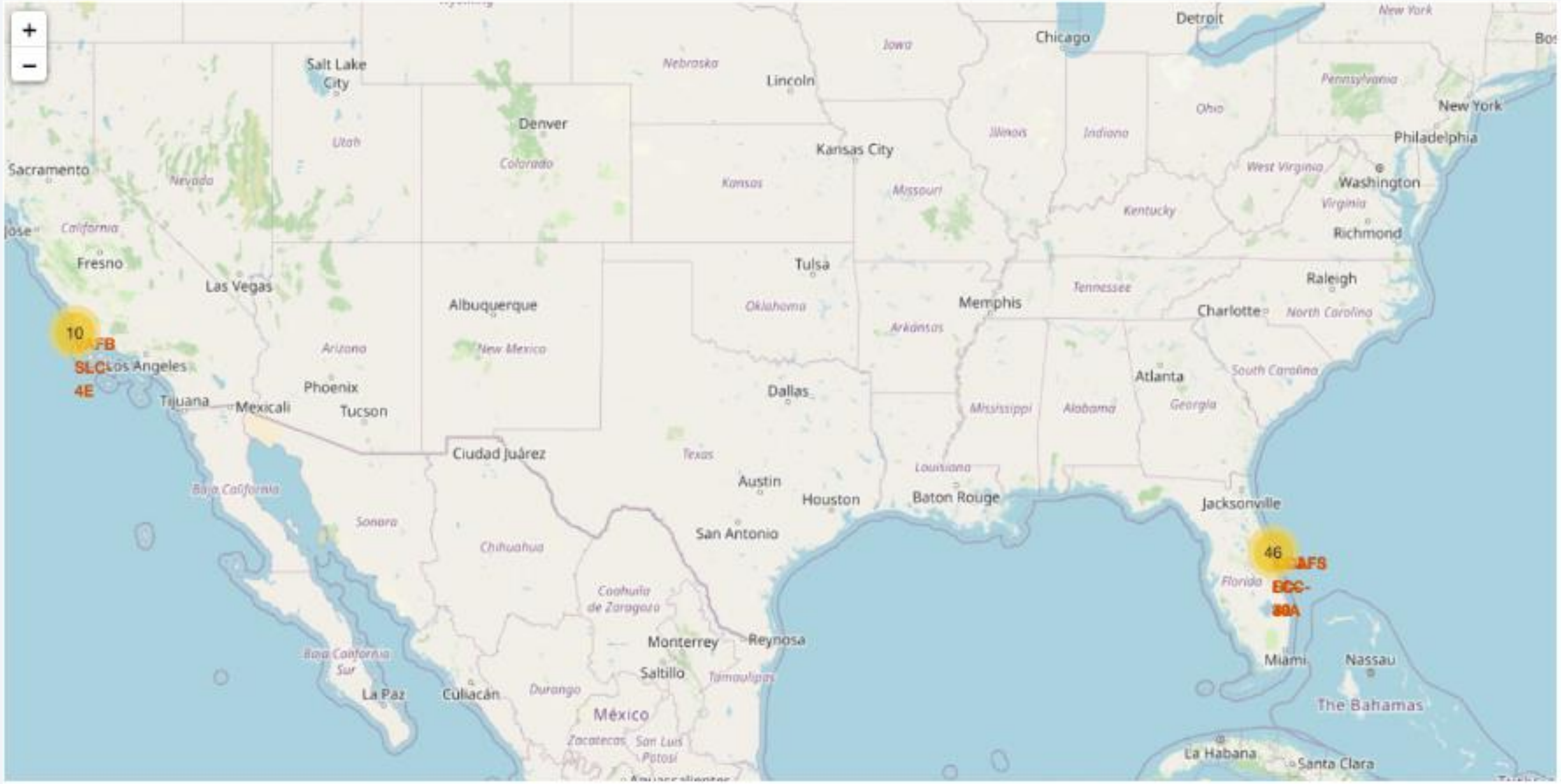
Out[55]:

```
[('No attempt', 10),
 ('Success (ground pad)', 5),
 ('Success (drone ship)', 5),
 ('Failure (drone ship)', 5),
 ('Controlled (ocean)', 3),
 ('Uncontrolled (ocean)', 2),
 ('Precluded (drone ship)', 1),
 ('Failure (parachute)', 1)]
```

Section 3

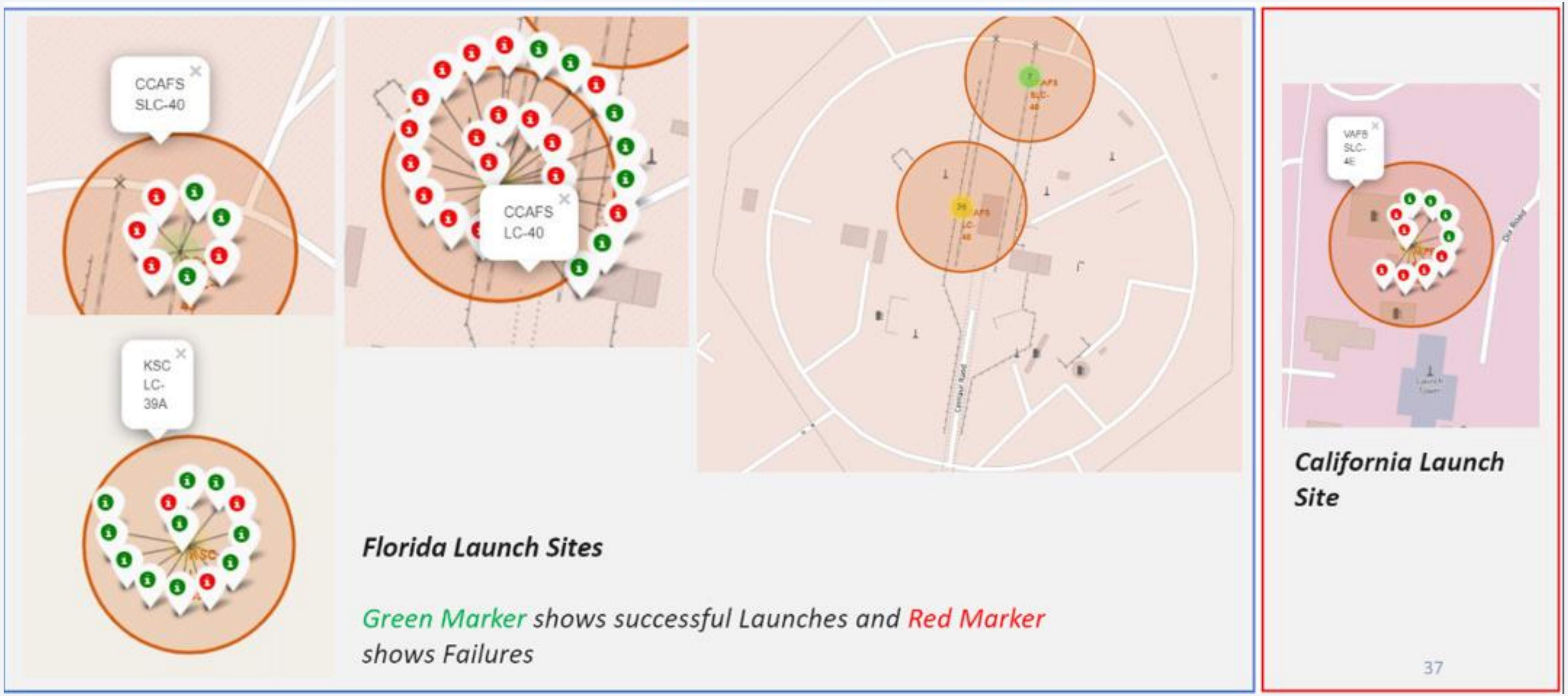
Launch Sites Proximities Analysis

All Launch Sites Location Markers



All the launch sites are near the ocean

Color labeled Launch Outcomes



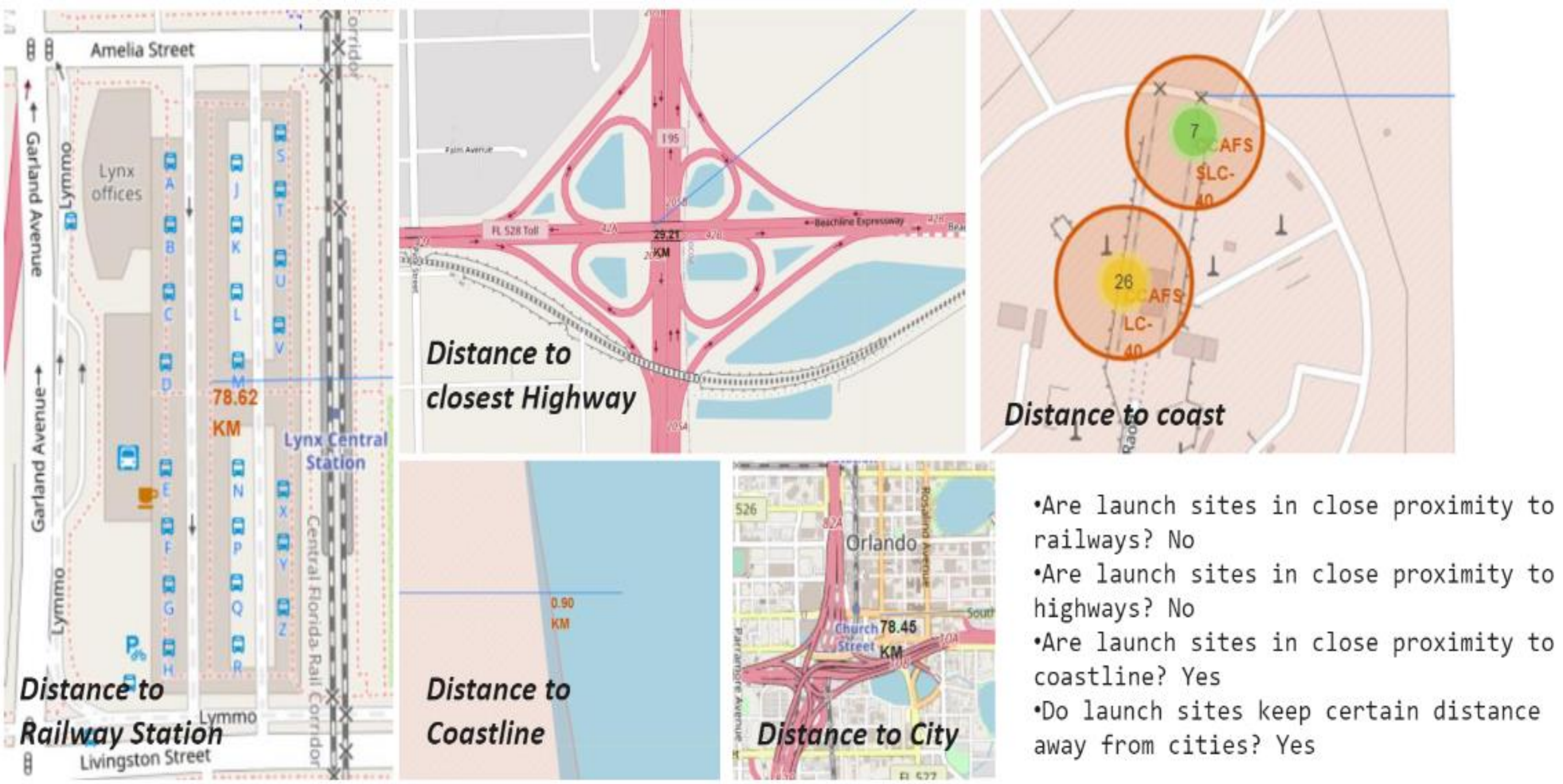
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

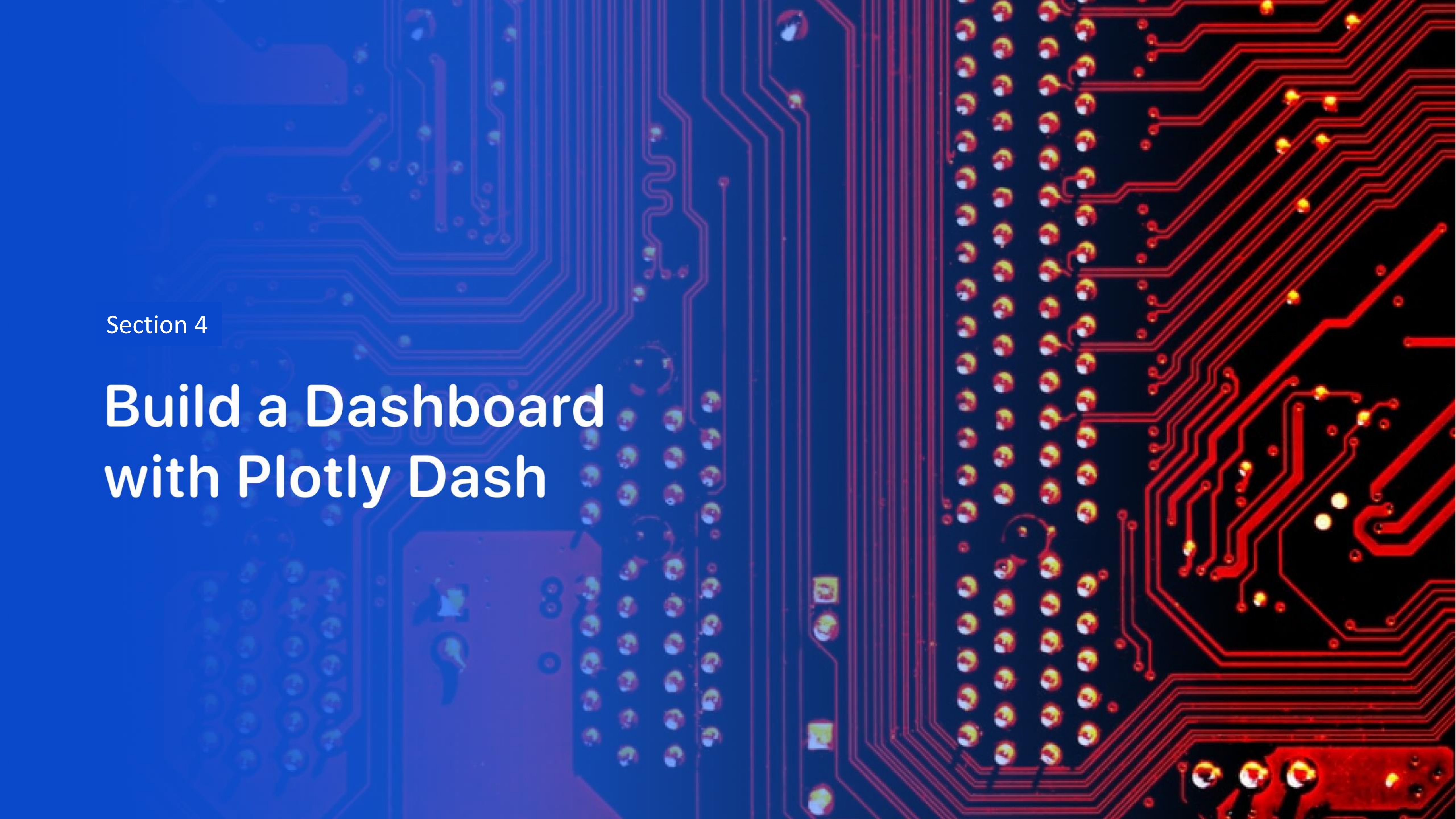
Launch Sites Distances



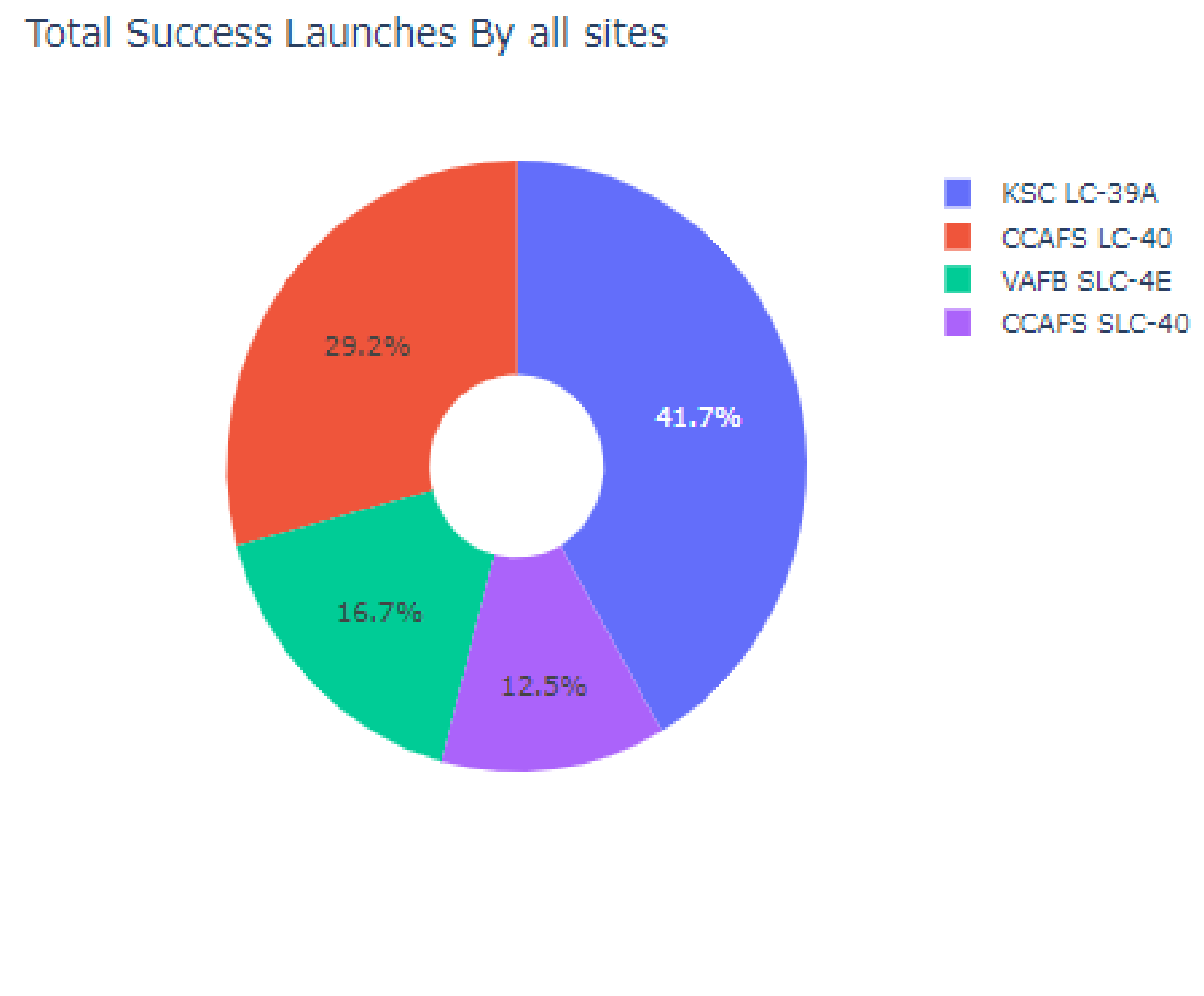
- •Are launch sites in close proximity to railways? No
- •Are launch sites in close proximity to highways? No
- •Are launch sites in close proximity to coastline? Yes
- •Do launch sites keep certain distance away from cities? Yes

Section 4

Build a Dashboard with Plotly Dash

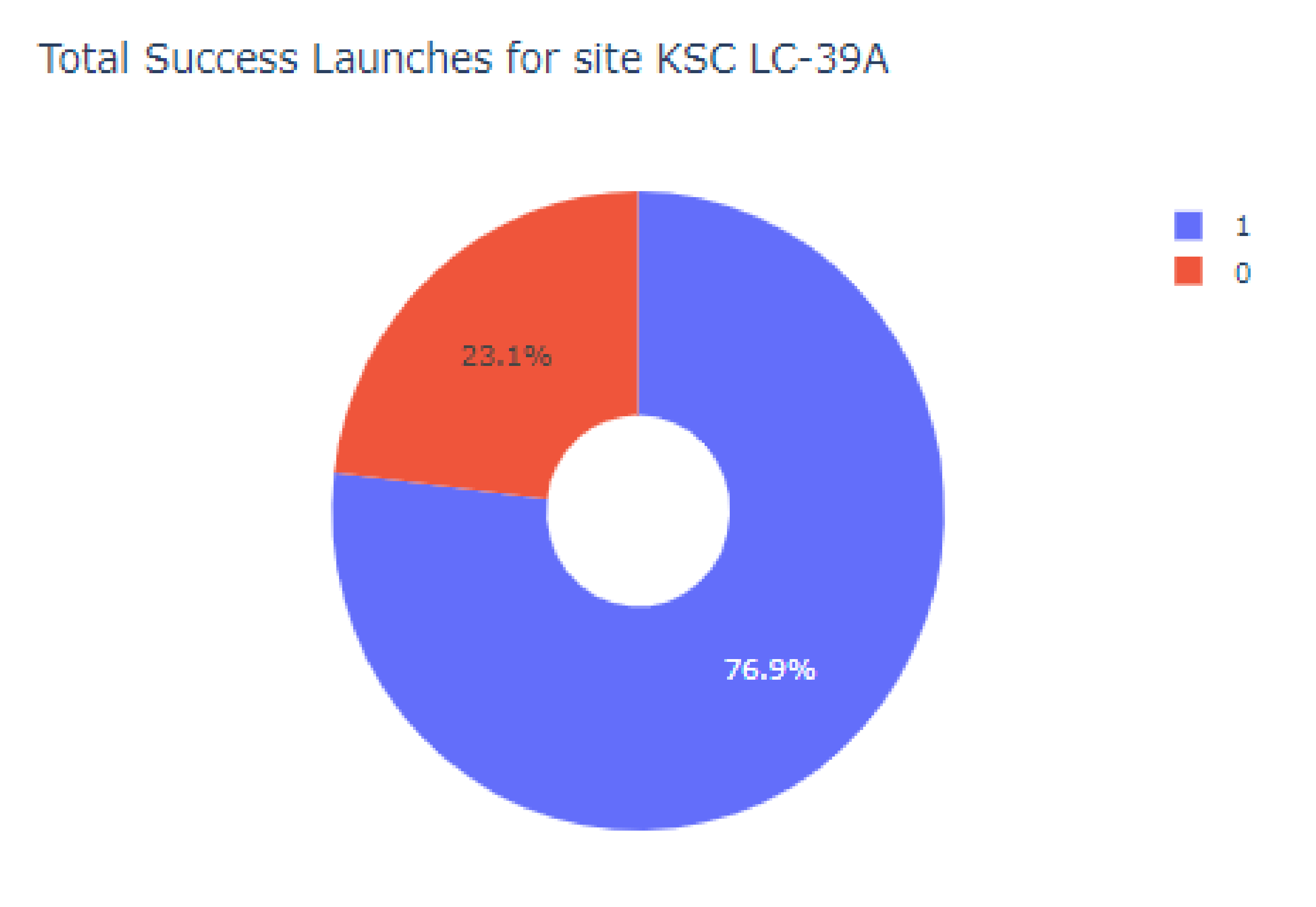


Total success rate of Launch sites



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

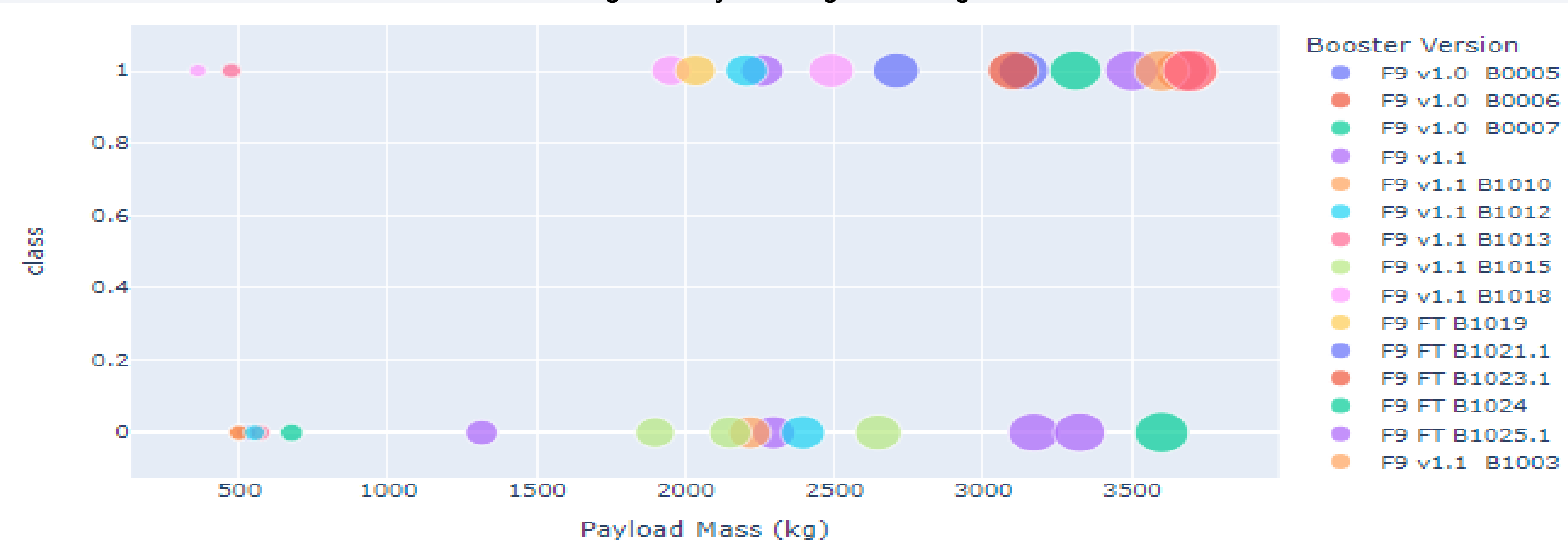
The highest launch site success

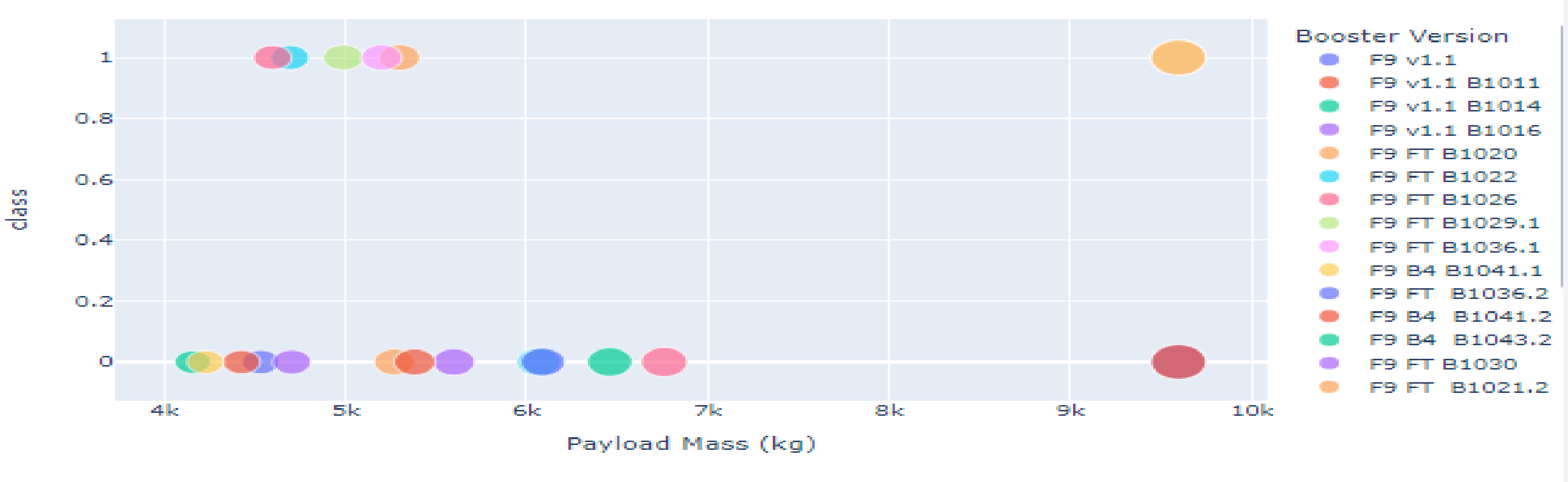


KSC LC-39A achived a 76.9% success rate while getting a 23.1% failure rate

Correlation between Payload and Launch Outcome

Low weighted Payload 0kg – 4000kg





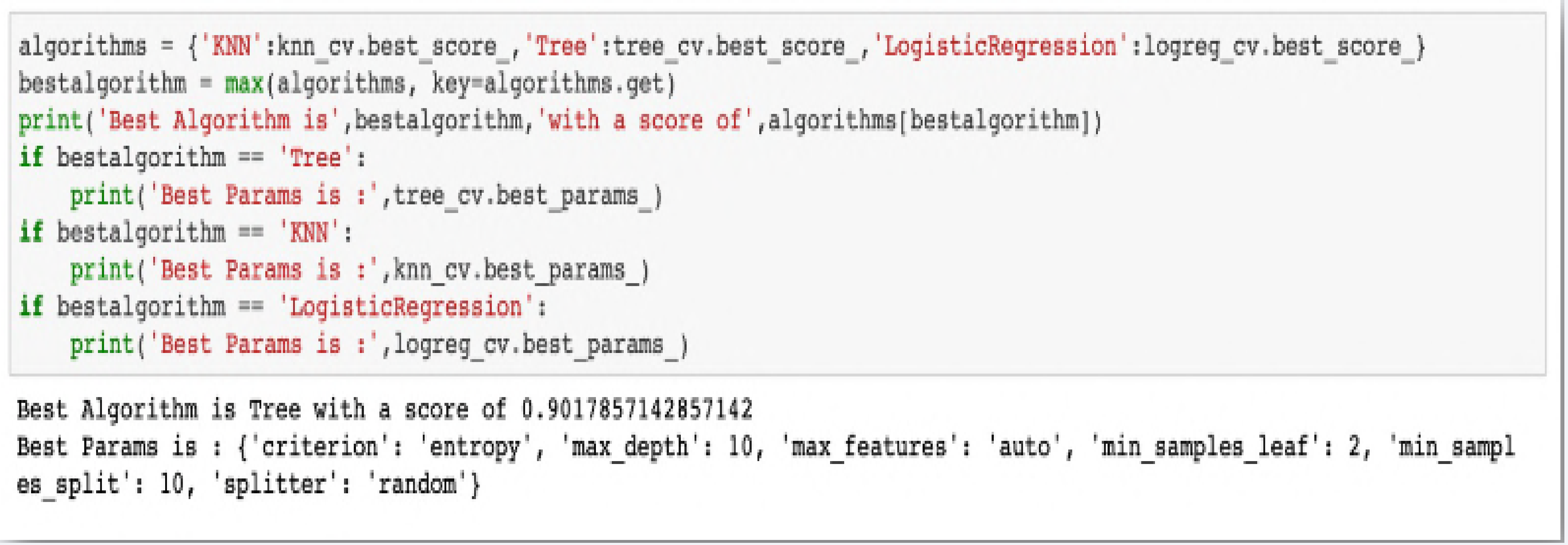
Heavy weighted Payload 4000kg – 10000kg

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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

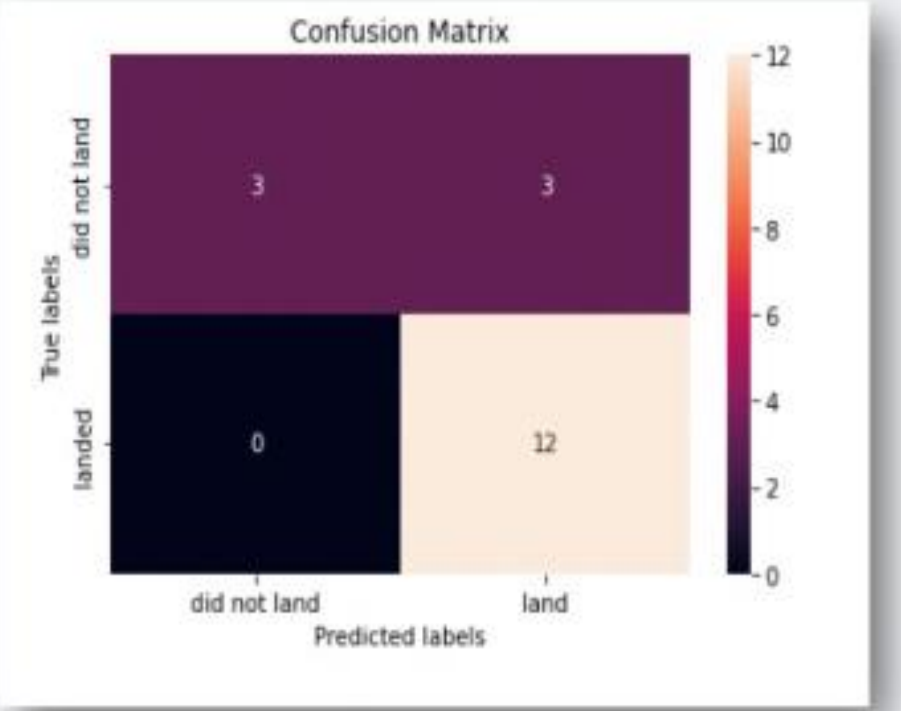


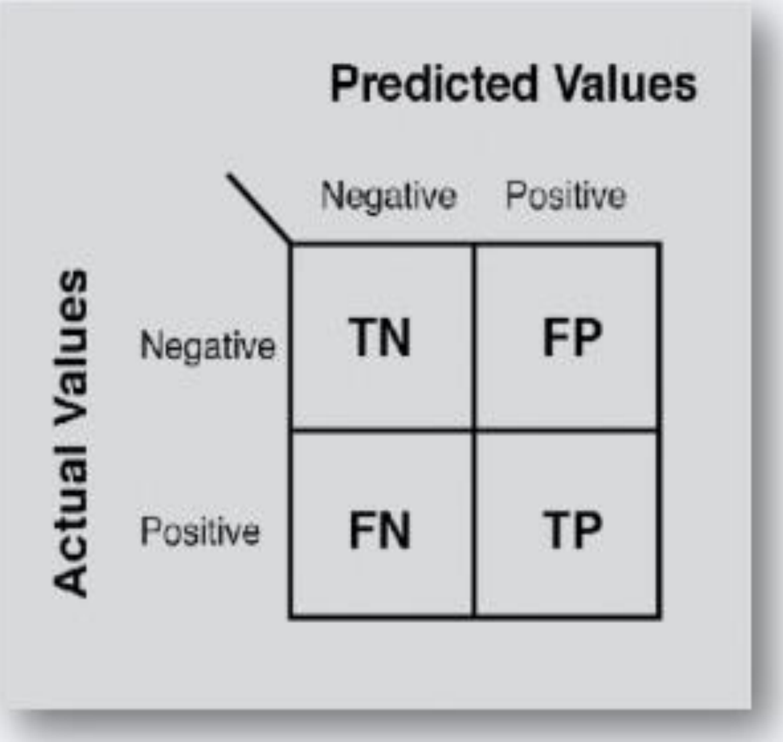
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9017857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_sampl
es_split': 10, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.





		Predicted Values	
		Negative	Positive
Actual Values	Negative	TN	FP
	Positive	FN	TP

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.
- Low weighted payloads perform better than the heavier payloads
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

All my codes can be found on my GitHub

https://github.com/linhnguyenlethuy1611/linh_datascience

Thank you!