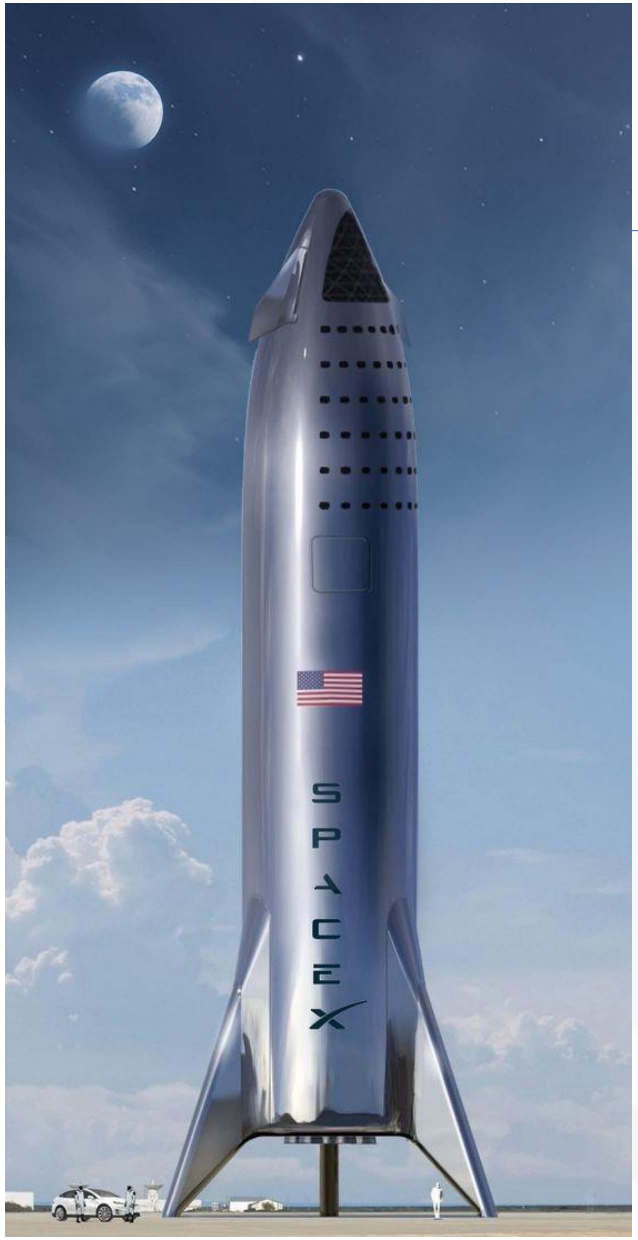


# Applied Data Science Capstone

Cassandra Raj
20th July 2022

IBM Developer

SKILLS NETWORK

# Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Discussion
- 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

- Space X 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 Space X 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 space X for a rocket launch.

## Problems you want to find answers

- 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.

# Methodology

- **Data collection methodology:**
  - Via SpaceX Rest API
  - Web Scrapping from Wikipedia
- **Perform data wrangling:**
  - One-hot encoding was applied to categorical features
- **Perform exploratory data analysis (EDA) using visualization and SQL:**
  - Scatter and bar graphs to show pattern between data
- **Perform interactive visual analytics:**
  - Using Folium and Plotly Dash Visualizations
- **Perform predictive analysis using classification models:**
  - Build and evaluate classifications models

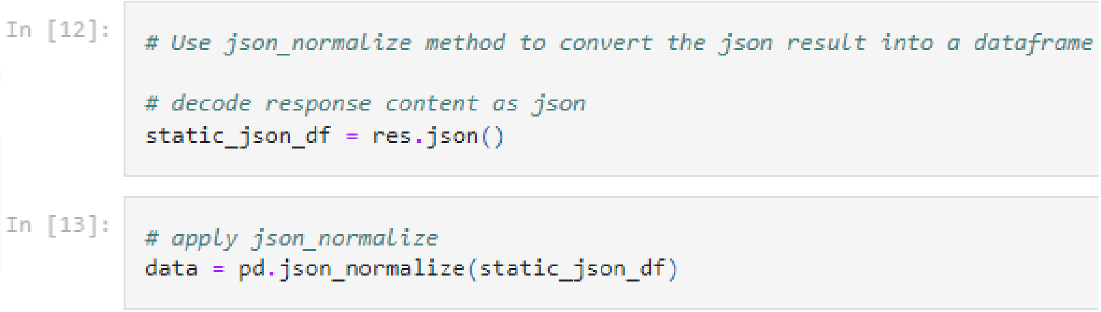
# Data Collection

- **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

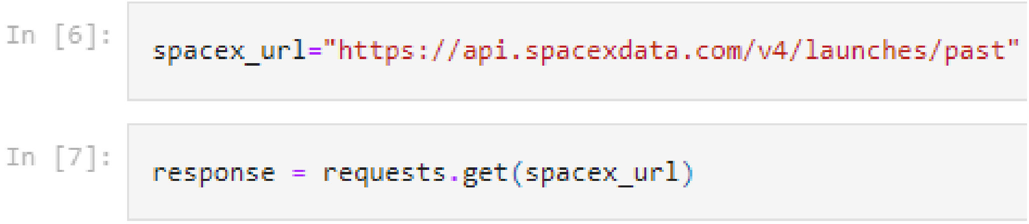
- The link to notebook: https://github.com/Claopatra/Final_Asignment/blob/main/Data%20Collection%20API%20Lab%20(1).ipynb

**2. Use json_normalize method to convert json result to dataframe**



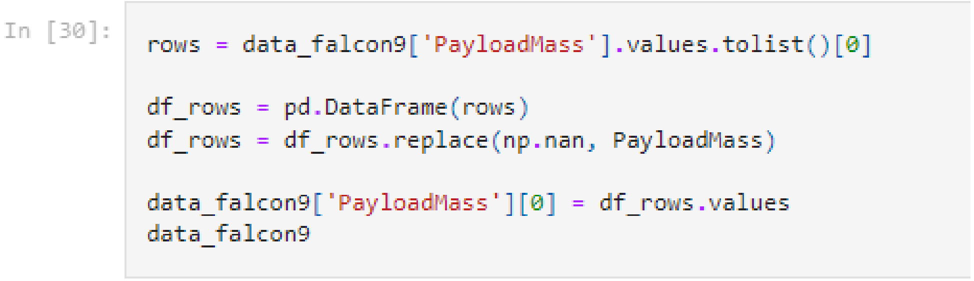
```
In [12]:  # Use json_normalize method to convert the json result into a dataframe

          # decode response content as json
          static_json_df = res.json()

In [13]:  # apply json_normalize
          data = pd.json_normalize(static_json_df)
```

**1. Get request for rocket launch data using API**



```
In [6]:  spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]:  response = requests.get(spacex_url)
```

**3. We then performed data cleaning & filling in the missing values**



```
In [30]:  rows = data_falcon9['PayloadMass'].values.tolist()[0]

          df_rows = pd.DataFrame(rows)
          df_rows = df_rows.replace(np.nan, PayloadMass)

          data_falcon9['PayloadMass'][0] = df_rows.values
          data_falcon9
```

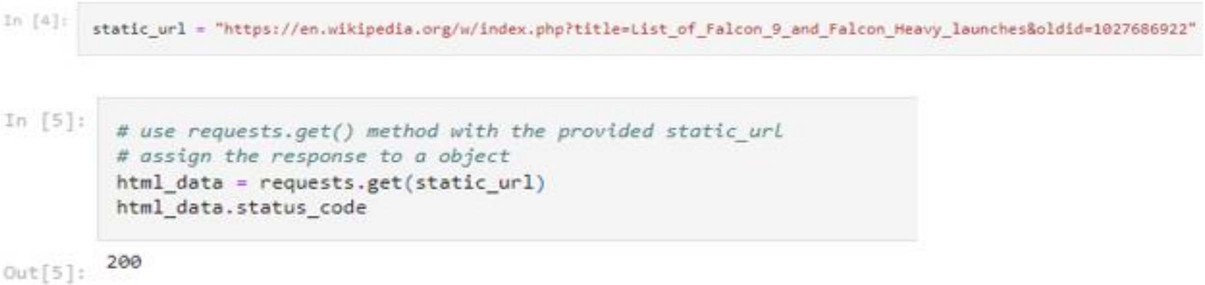
IBM Developer

SKILLS NETWORK

# Data Collection– Web Scraping

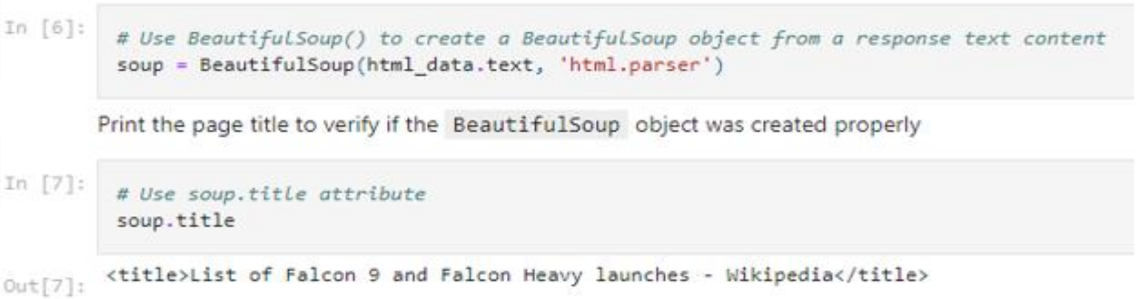
- The link to the notebook is [https://github.com/Claopatra/Final_Assignment/blob/main/Data%20Collection%20API%20Lab%20(1).ipynb](https://github.com/Claopatra/Final_Assignment/blob/main/Data%20Collection%20API%20Lab%20(1).ipynb)

### 1. Getting response from HTML



```
In [4]:  static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

In [5]:  # use requests.get() method with the provided static_url
         # assign the response to a object
         html_data = requests.get(static_url)
         html_data.status_code

Out[5]:  200
```

### 2. Create BeautifulSoup object from HTML response

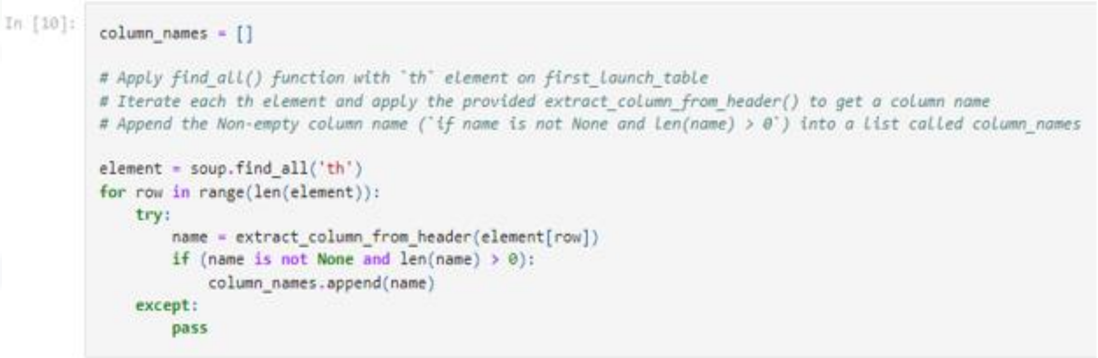


```
In [6]:  # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
         soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]:  # Use soup.title attribute
         soup.title

Out[7]:  <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

### 3. Getting column names

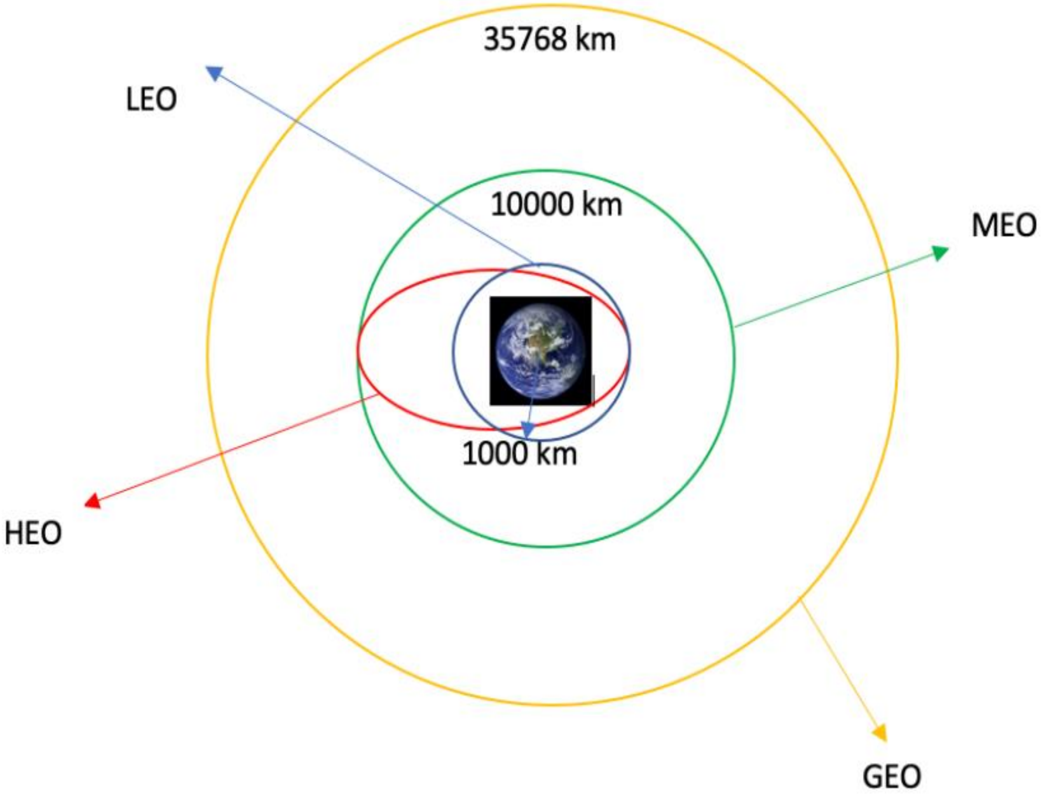


```
In [10]:  column_names = []

          # Apply find_all() function with `th` element on first_launch_table
          # Iterate each th element and apply the provided extract_column_from_header() to get a column name
          # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

          element = soup.find_all('th')
          for row in range(len(element)):
              try:
                  name = extract_column_from_header(element[row])
                  if (name is not None and len(name) > 0):
                      column_names.append(name)
              except:
                  pass
```

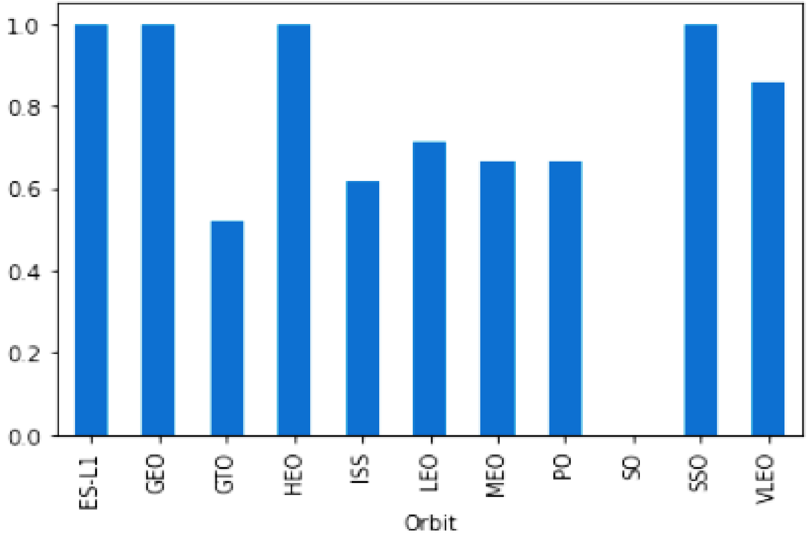
# Data Wrangling

- 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.

- The link to the notebook is: **https://github.com/Claopatra/Final_Assignment/blob/main/labs-jupyter-spacex-Data%20wrangling%20(2).ipynb**



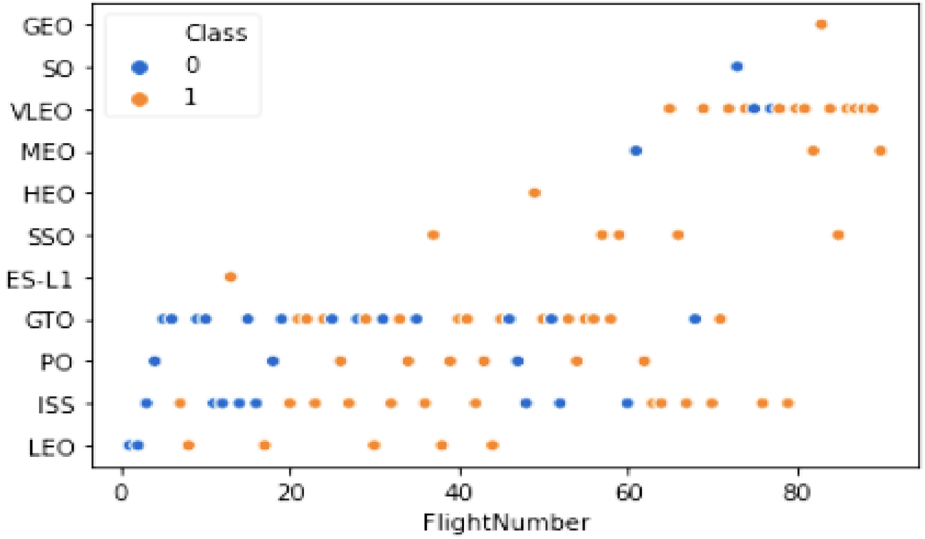
# EDA with Data Visualizations



**Bar Chart**

- The link to the notebook is: [https://github.com/Claopatra/Final_Assignment/blob/main/jupyter-labs-eda-dataviz%20(2).ipynb](https://github.com/Claopatra/Final_Assignment/blob/main/jupyter-labs-eda-dataviz%20(2).ipynb)

- 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.



**Scatter Plot**

# 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**
  - **Display 5 records where launch sites begin with the string 'CCA'**
  - **Display the total payload mass carried by boosters launched 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 achieved**

- The link to the notebook is: **https://github.com/Claopatra/Final_Assignment/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).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:
  - **Are launch sites near railways, highways and coastlines.**
  - **Do launch sites keep certain distance away from cities.**
- The link to the notebook is: **https://github.com/Claopatra/Final_Assignment/blob/main/lab_jupyter_launch_site_location%20(1).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/Claopatra/Final_Assignment/blob/main/spacex_dash_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/Claopatra/Final_Assignment/blob/main/Machine%20Learning%20Prediction.ipynb**

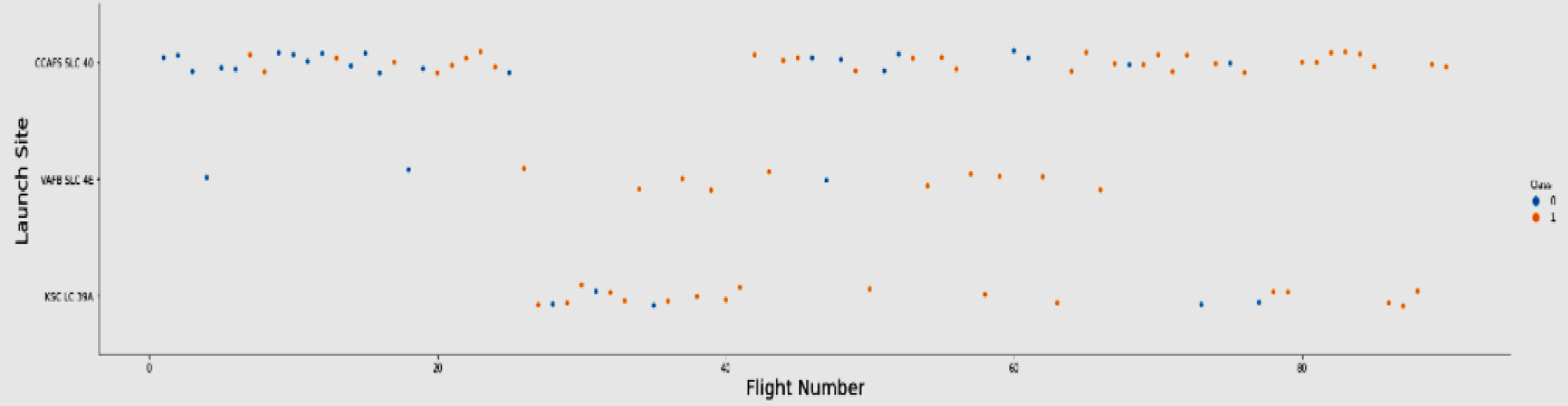
# Result

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

# EDA WITH VISUALIZATIONS

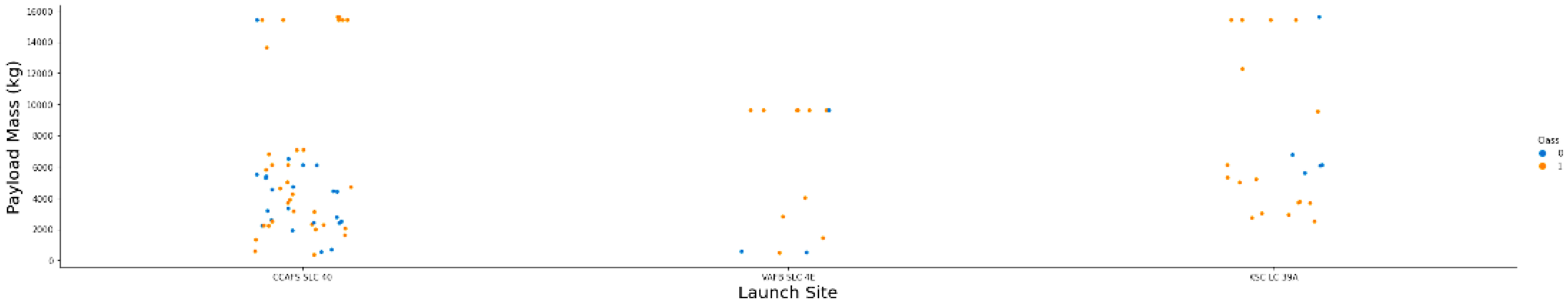
# Flight Number vs. Launch Site

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



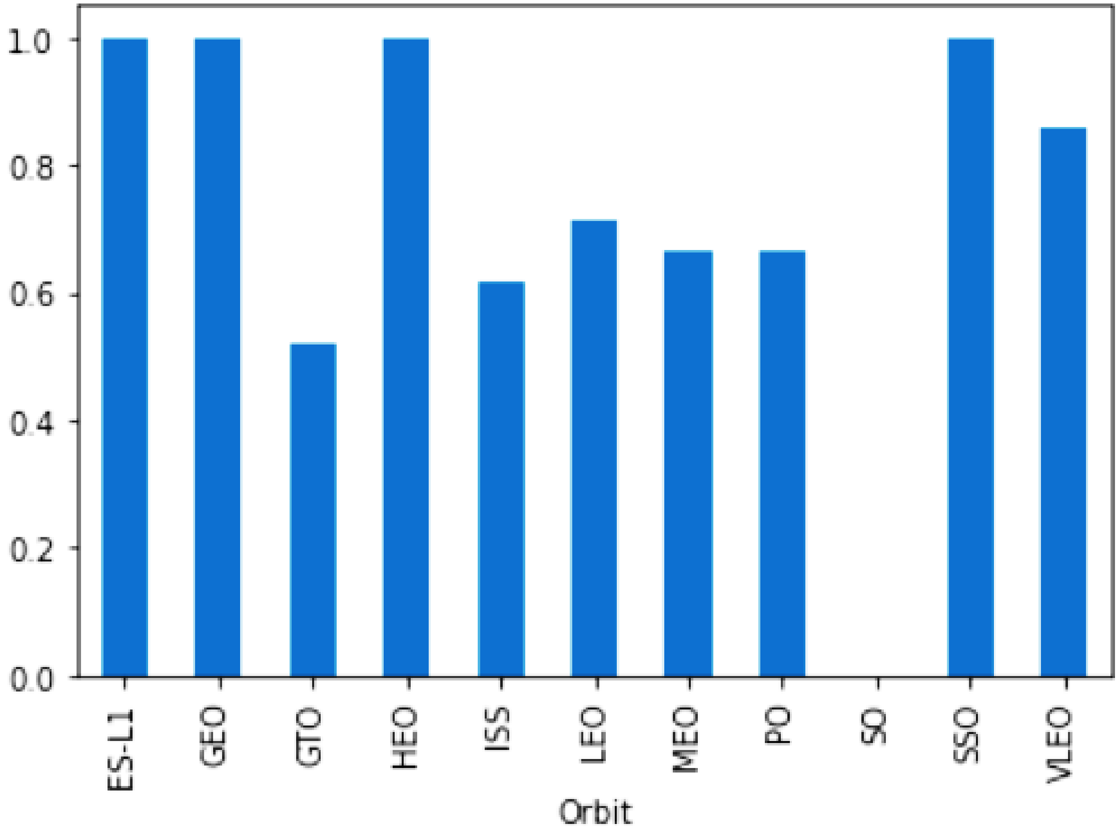
# Payload vs. Launch Site

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



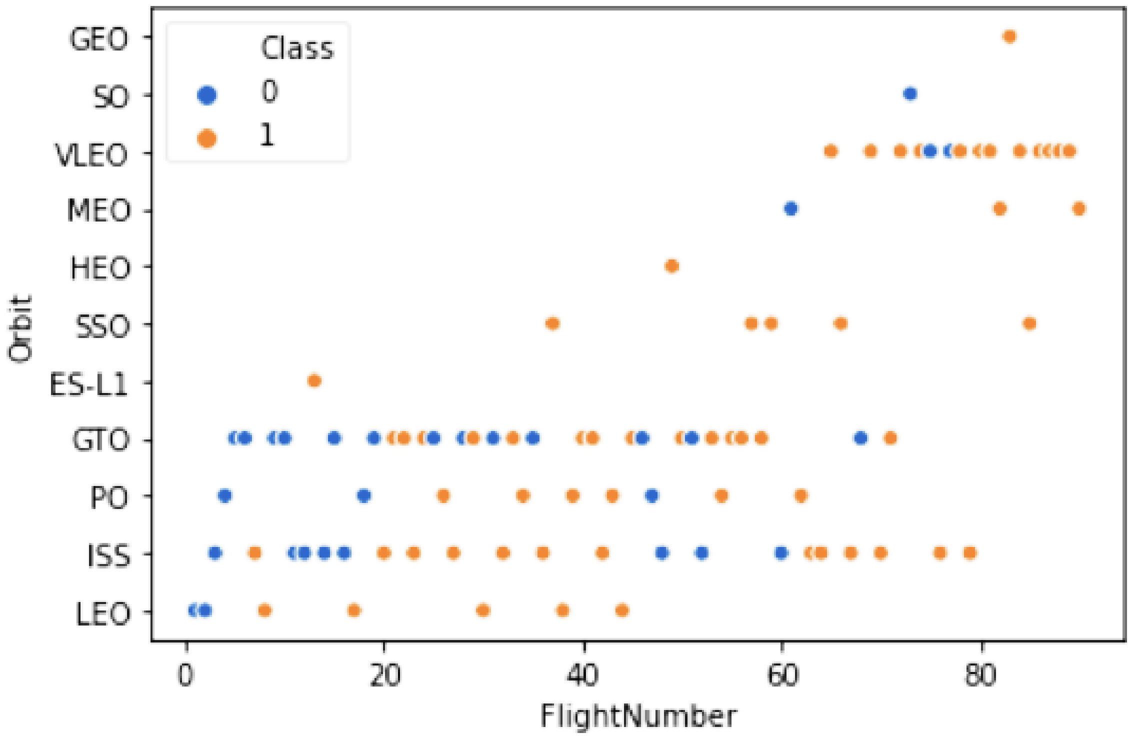
# Success Rate vs. Orbit Type

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



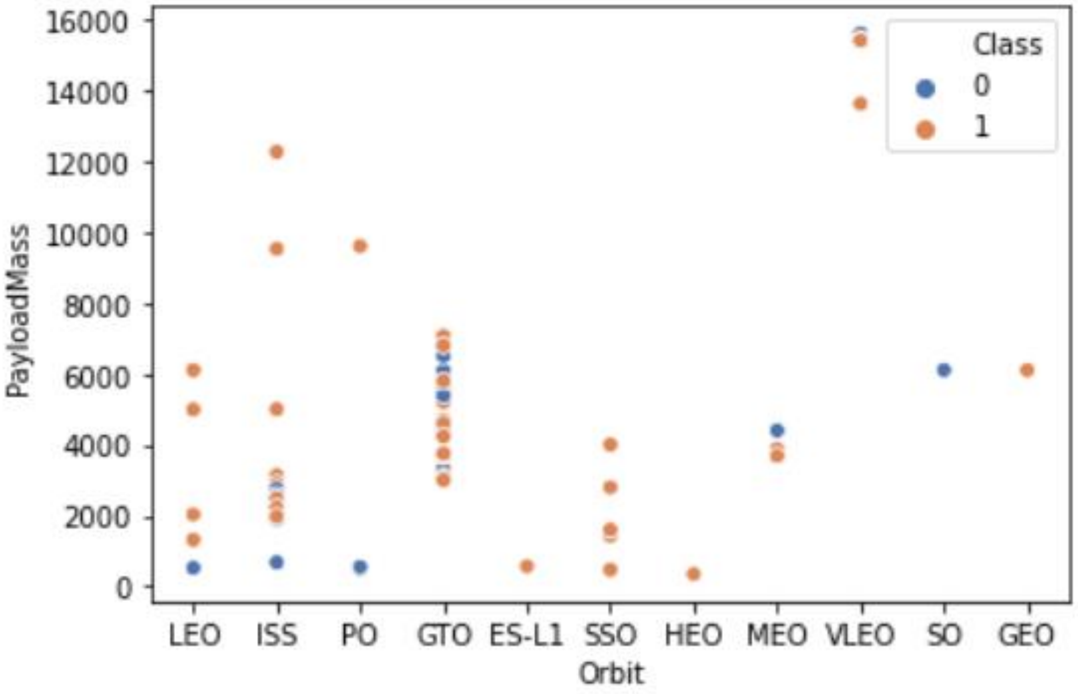
# Flight Number vs. Orbit Type

- We see that for LEO orbit the success rate increases with the number of flights
- Contrastingly, there’s no relationship between flight number and the GTO orbit



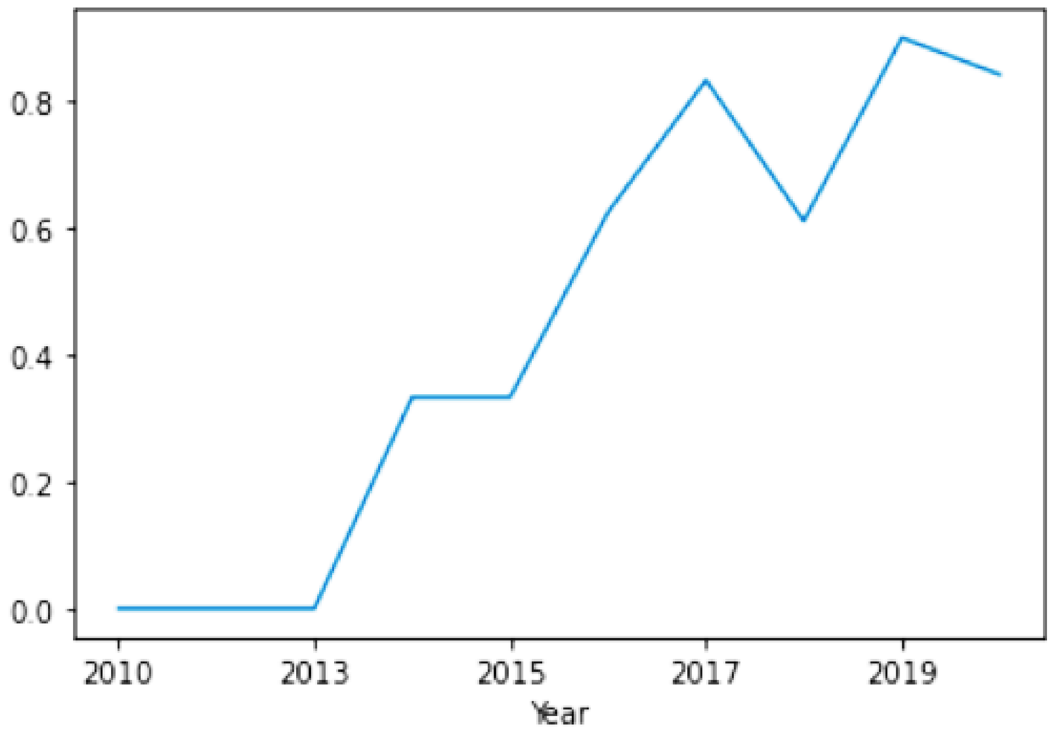
# Payload vs. Orbit Type

- 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



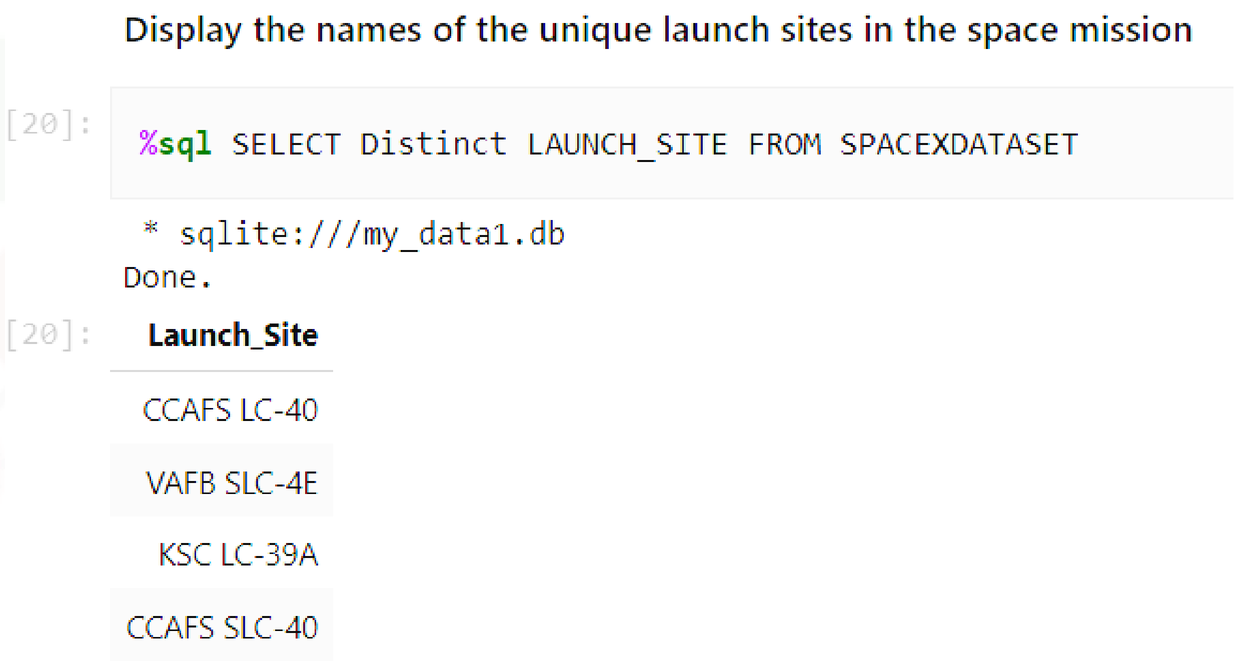
IBM Developer

SKILLS NETWORK

# EDA WITH SQL

# 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

```
[20]: %sql SELECT Distinct LAUNCH_SITE FROM SPACEXDATASET
```

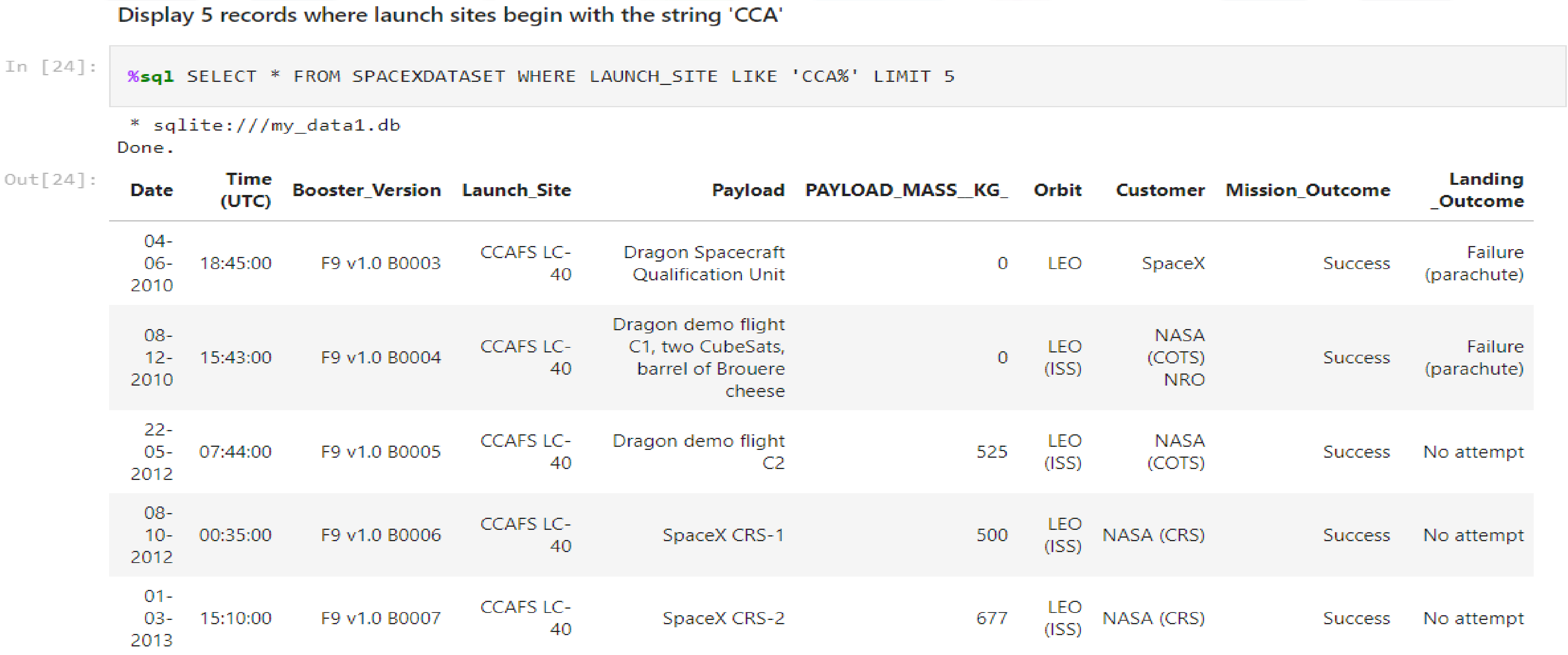
```
 * sqlite:///my_data1.db
Done.
```

[20]:

| Launch_Site |
|---|
| CCAFS LC-40 |
| VAFB SLC-4E |
| KSC LC-39A |
| CCAFS SLC-40 |

# Launch Site Names Begin with 'CCA'

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



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

```
In [24]: %sql SELECT * FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

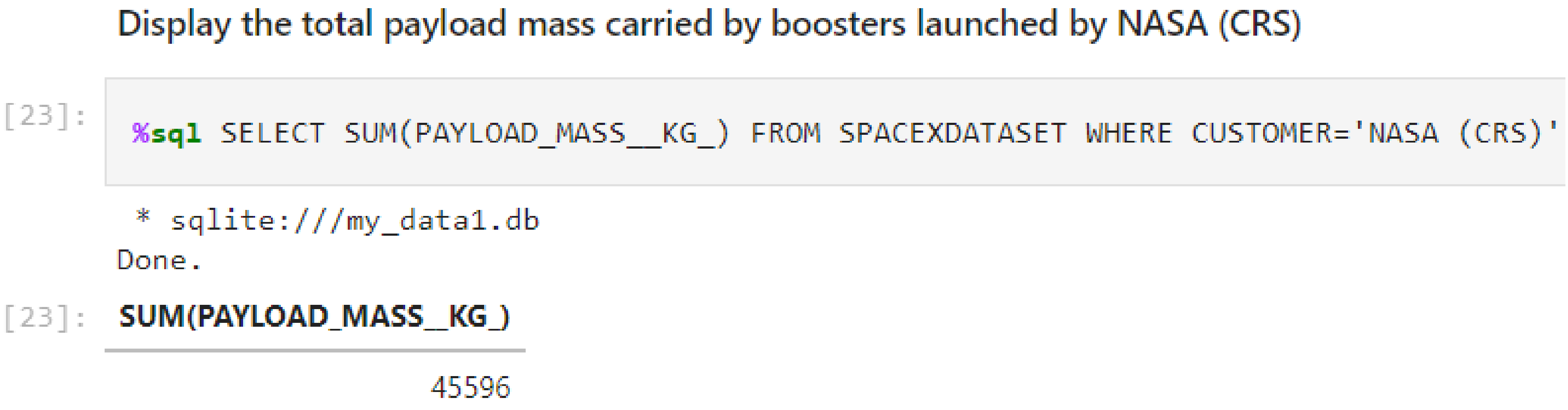
```
 * sqlite:///my_data1.db
Done.
```

Out[24]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (parachute) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# 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)

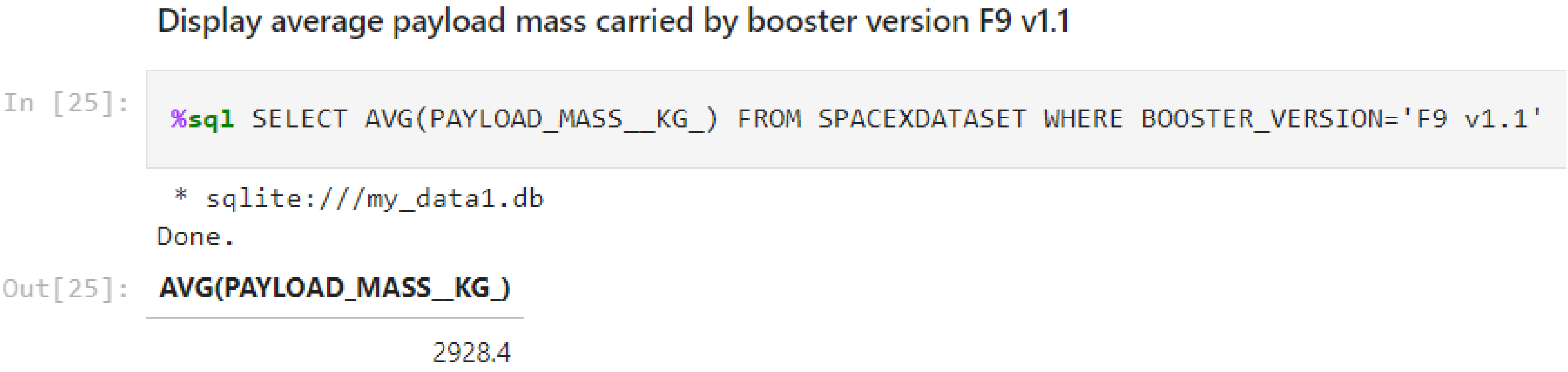
```
[23]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE CUSTOMER='NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

```
[23]: SUM(PAYLOAD_MASS__KG_)
                       45596
```

# Average Payload Mass By F9 v1.1

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

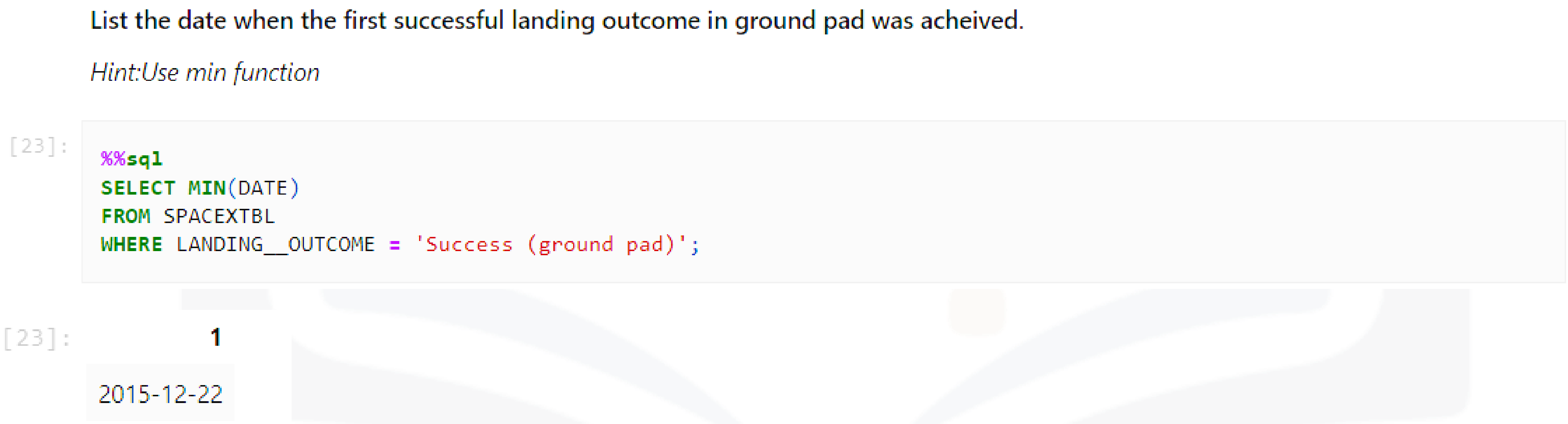


Display average payload mass carried by booster version F9 v1.1

```
In [25]:  %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE BOOSTER_VERSION='F9 v1.1'

           * sqlite:///my_data1.db
          Done.

Out[25]:  AVG(PAYLOAD_MASS__KG_)
                          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



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

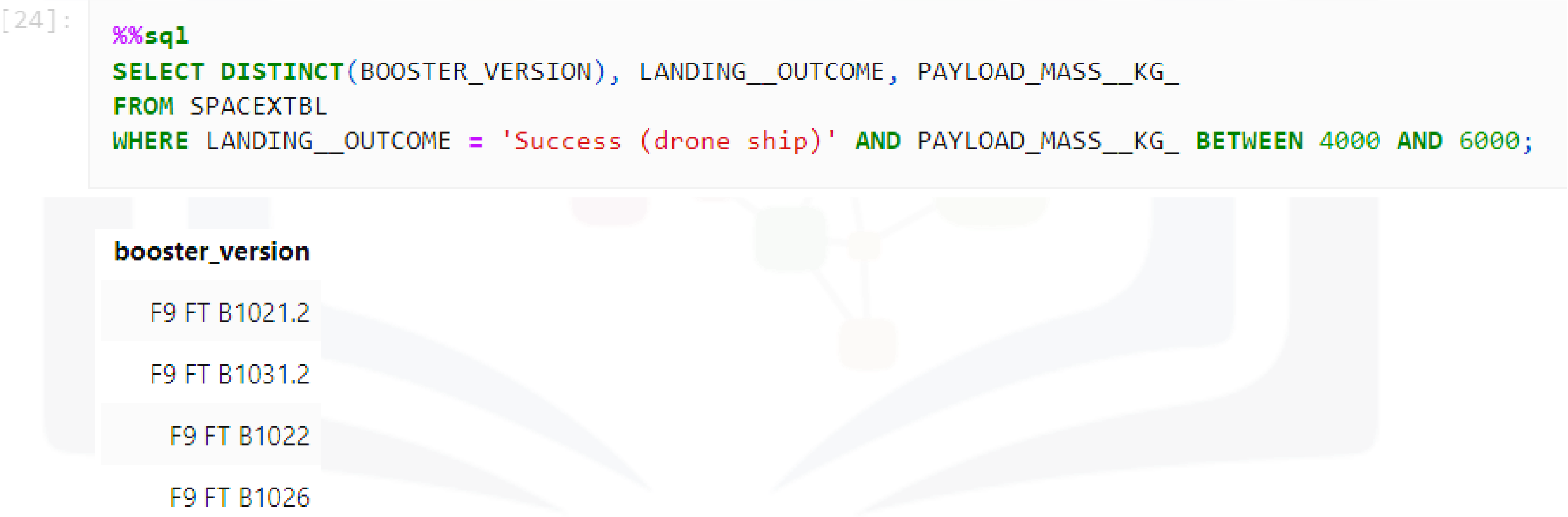
*Hint:Use min function*

```
[23]:  %%sql
       SELECT MIN(DATE)
       FROM SPACEXTBL
       WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
[23]:           1
       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

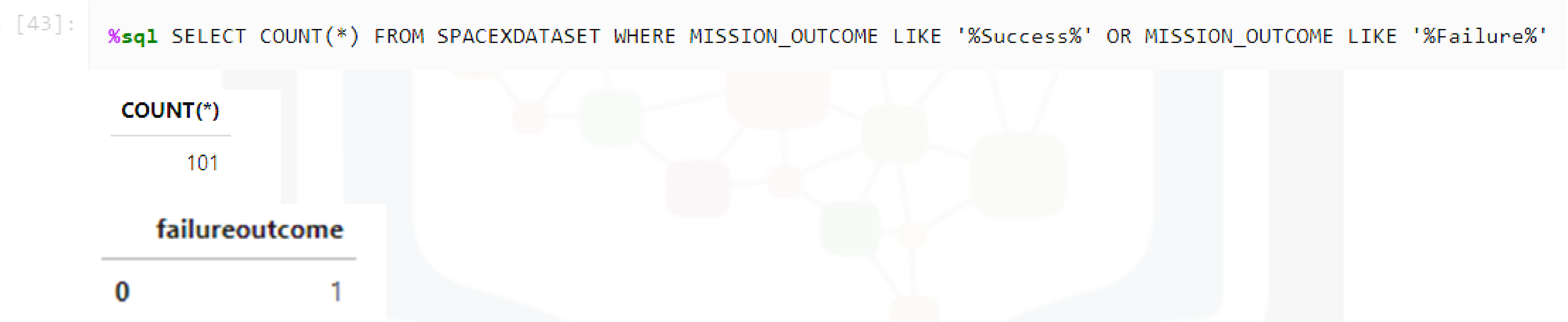


```
[24]:
%%sql
SELECT DISTINCT(BOOSTER_VERSION), LANDING__OUTCOME, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

| booster_version |
|---|
| F9 FT B1021.2 |
| F9 FT B1031.2 |
| F9 FT B1022 |
| F9 FT B1026 |

# Total Number of Successful and Failure Mission Outcomes

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

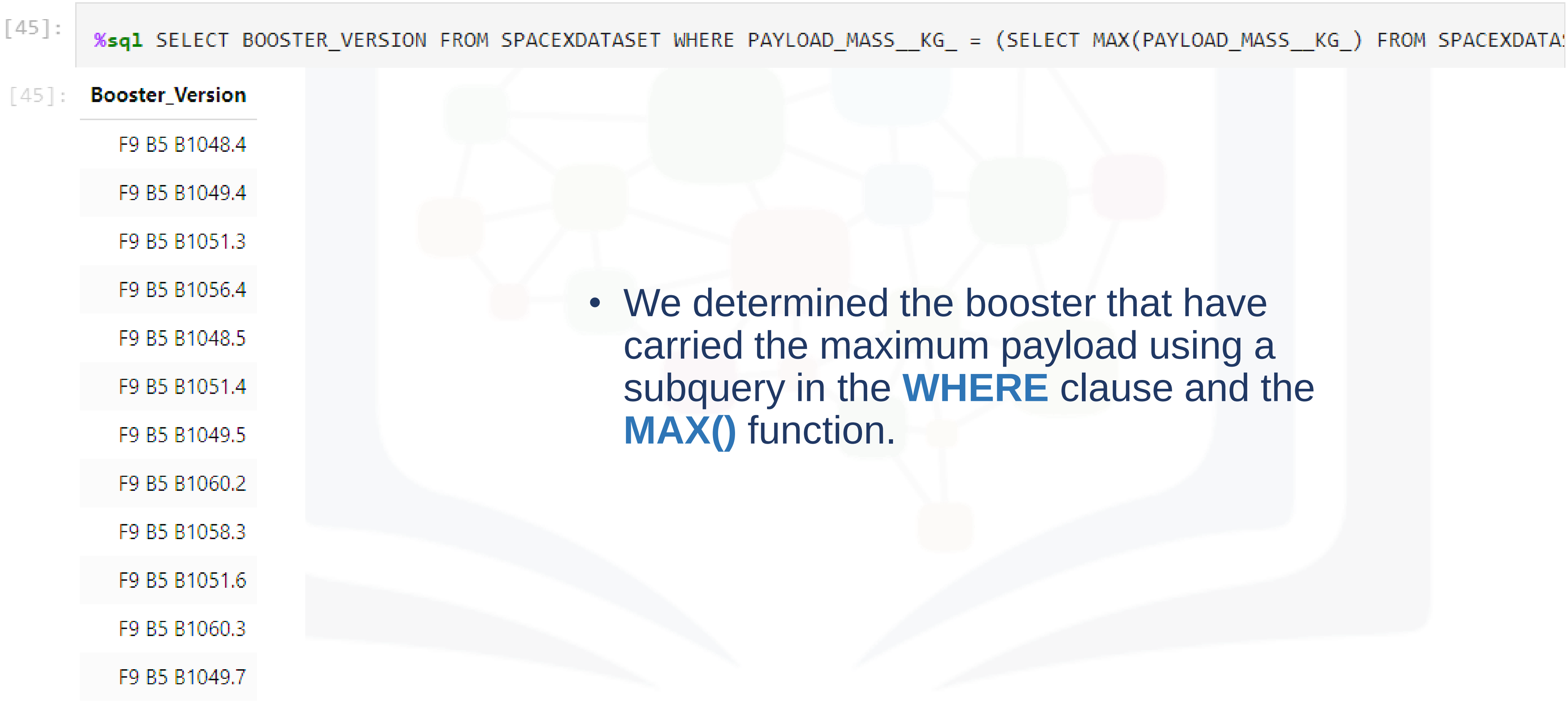


```
[43]:  %sql SELECT COUNT(*) FROM SPACEXDATASET WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

| COUNT(*) |
|---|
| 101 |

| | failureoutcome |
|---|---|
| 0 | 1 |

# Boosters Carried Maximum Payload



```
[45]:  %sql SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATA
```

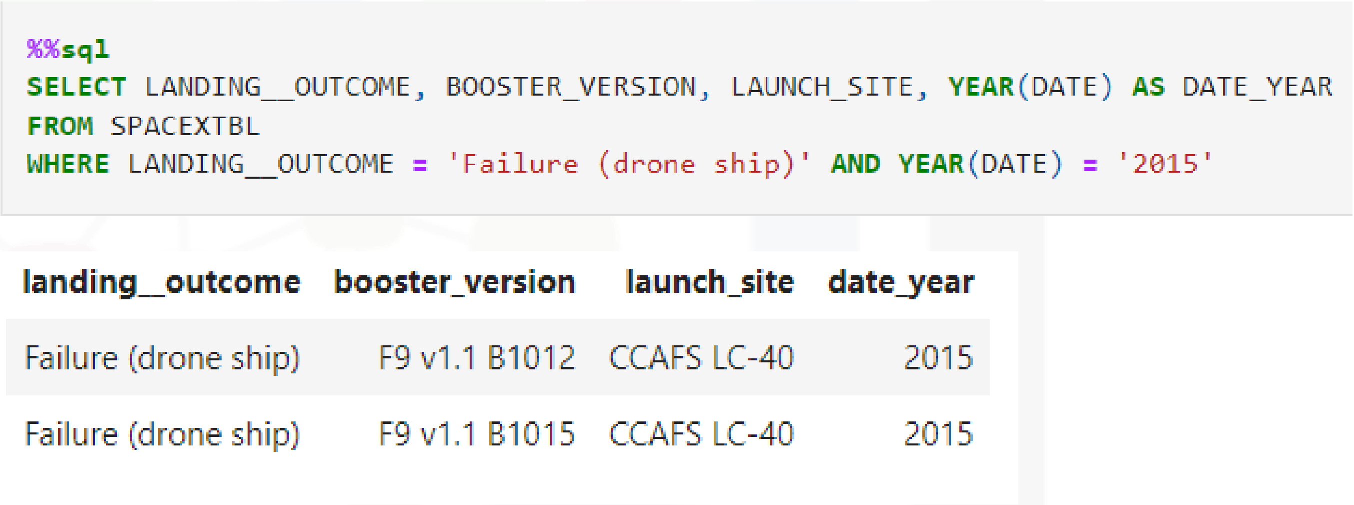
[45]:

| Booster_Version |
|---|
| 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 |

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

# 2015 Launch Records

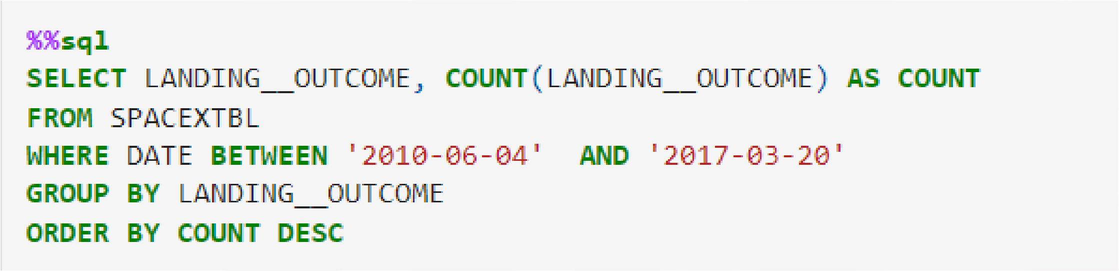
- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015



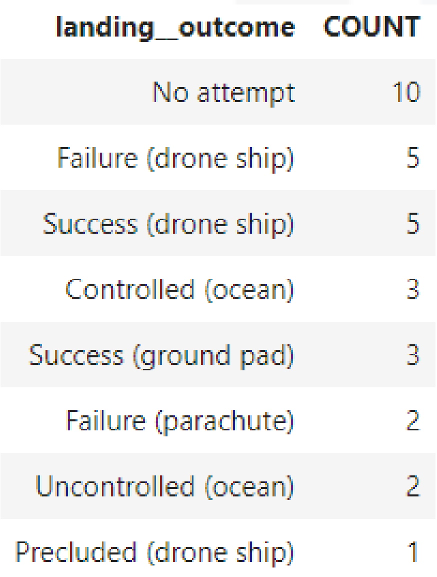
```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, YEAR(DATE) AS DATE_YEAR
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```

| landing__outcome | booster_version | launch_site | date_year |
|---|---|---|---|
| Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 | 2015 |
| Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 | 2015 |

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



```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS COUNT
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04'  AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY COUNT DESC
```



| landing__outcome | COUNT |
|---:|---:|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 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.

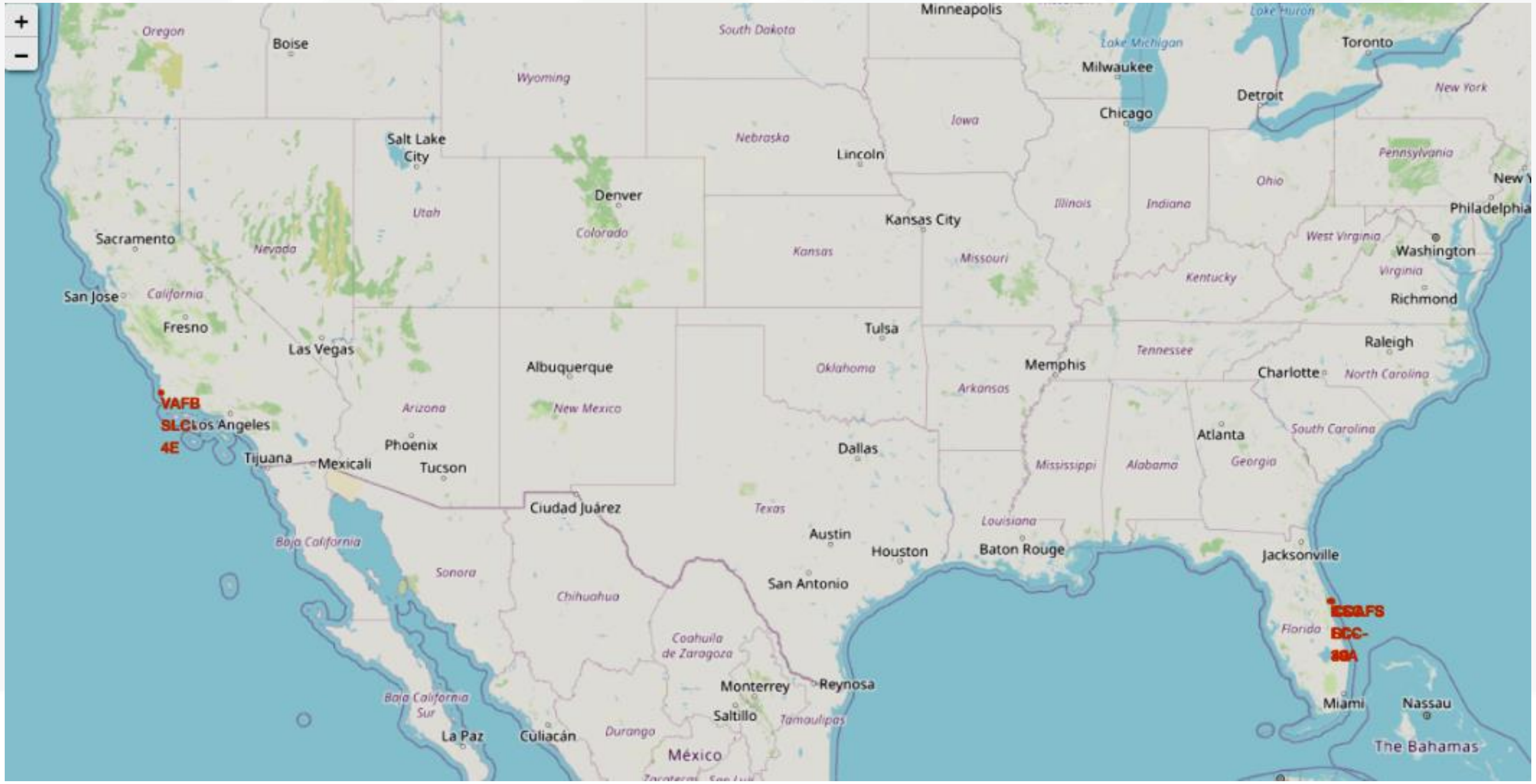
IBM Developer

SKILLS NETWORK

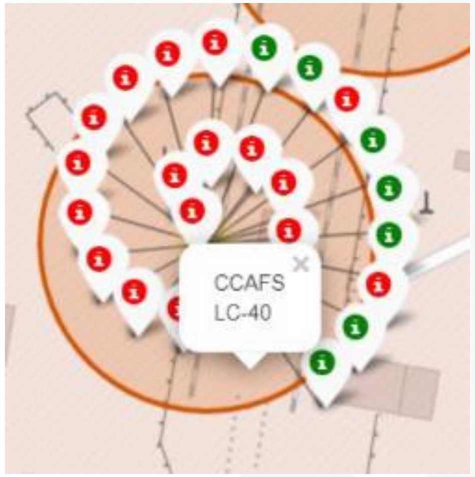
# INTERACTIVE MAP WITH FOLIUM

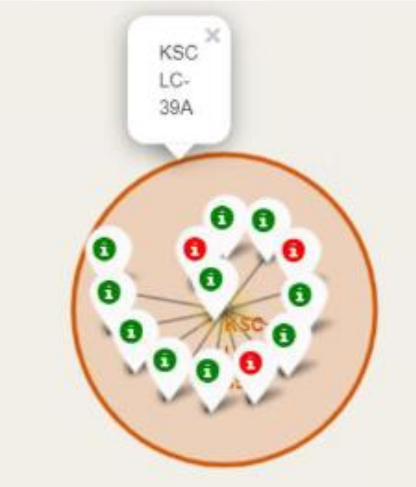
# All Launch Sites Global Map Markers

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

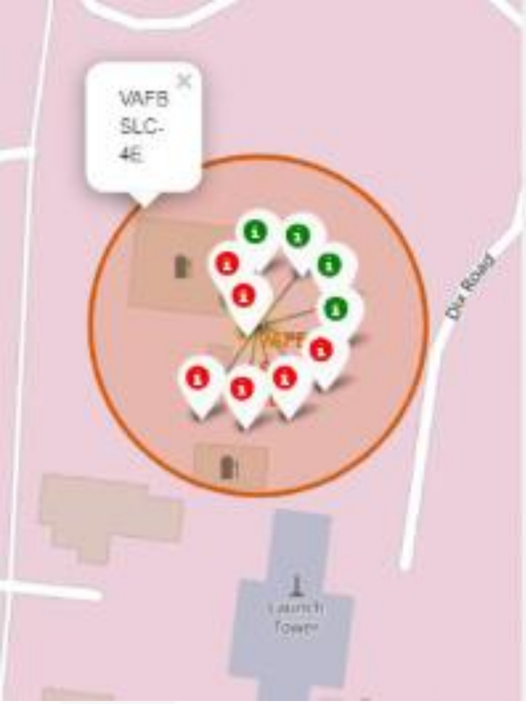


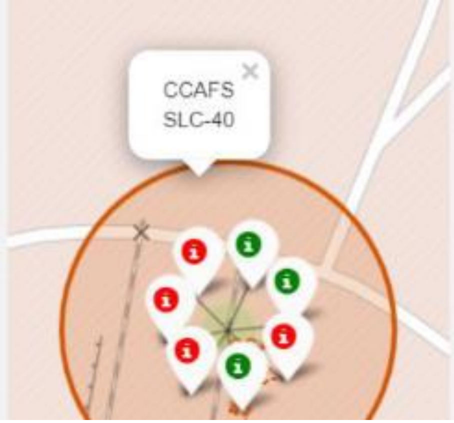
# Markers showing launch sites with color labels





7
26
Centaur Road



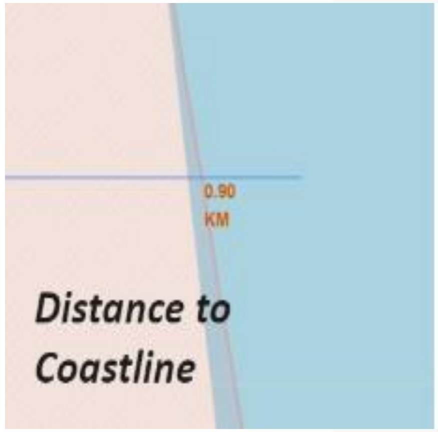


***Florida Launch Site***

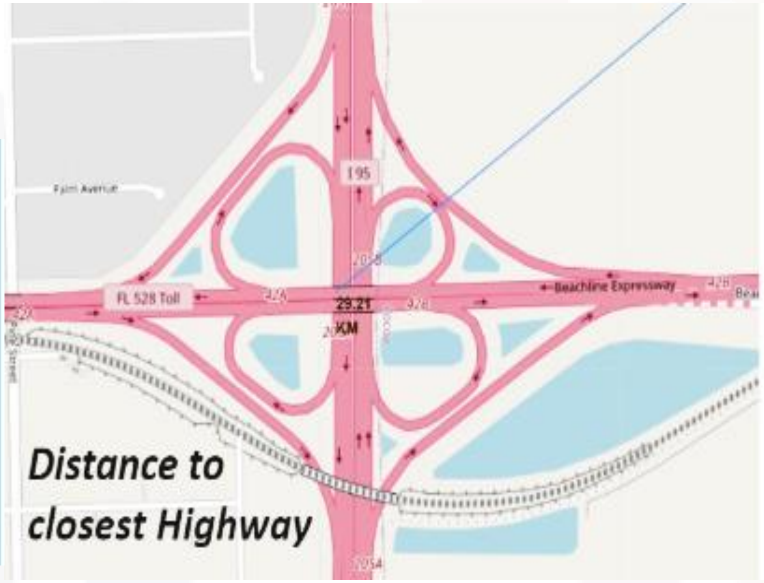
***Green Marker*** shows successful launches and ***Red Marker*** shows failures

***California Launch Sites***

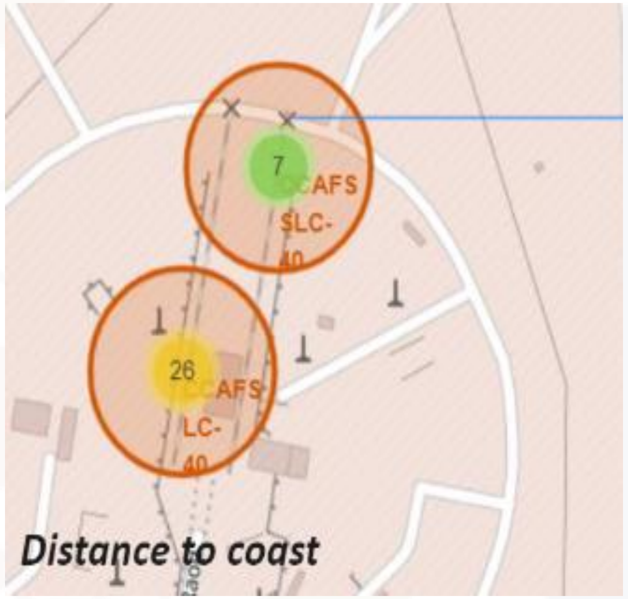
# Launch Site Distance To Landmarks



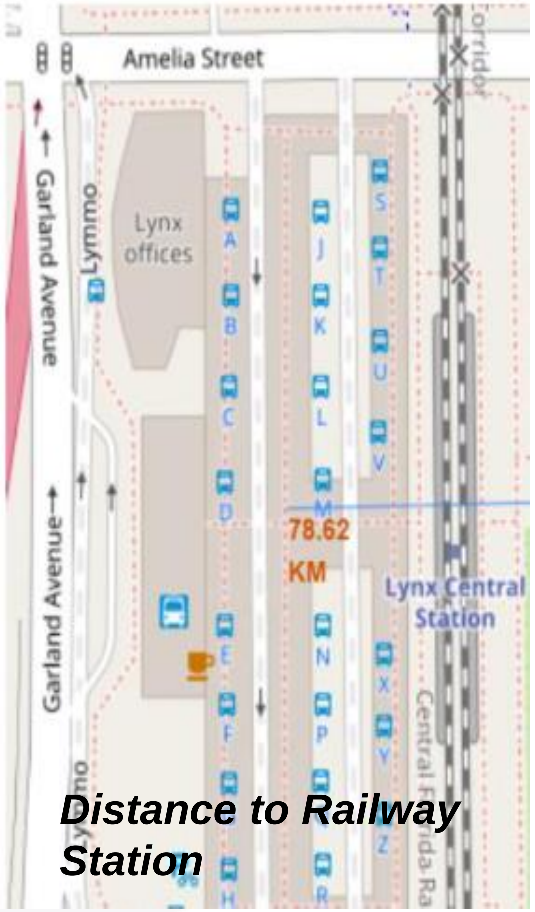
Distance to Coastline



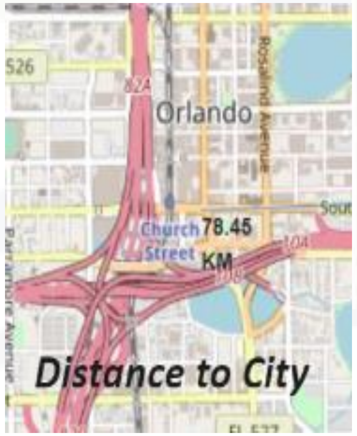
Distance to closest Highway



Distance to coast



Distance to Railway Station

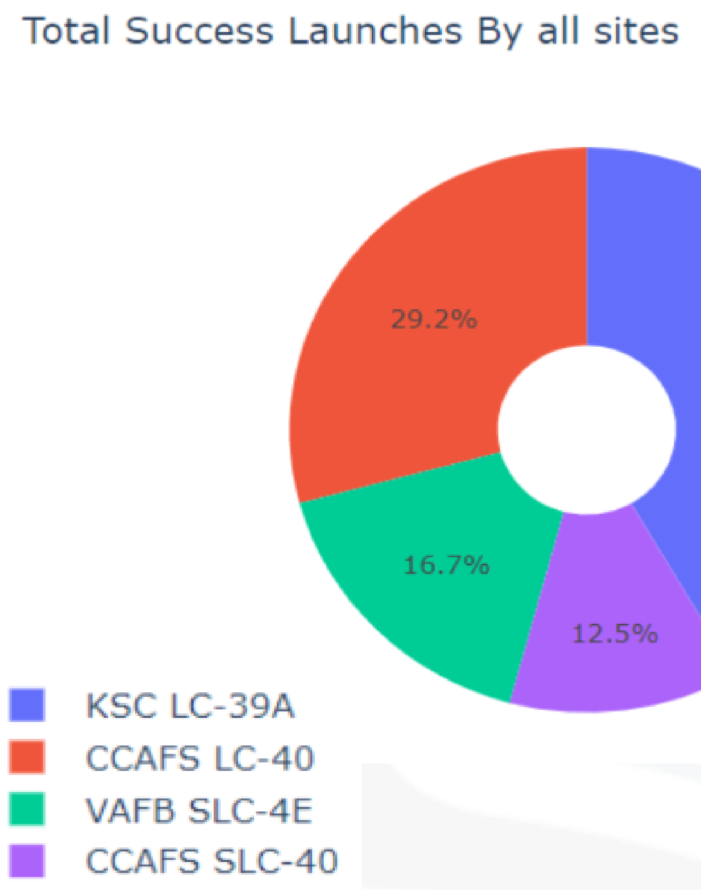


Distance to City

- ***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***

# BUILD A DASHBOARD WITH PLOTLY DASH

# Launch Success Counts For All Sites

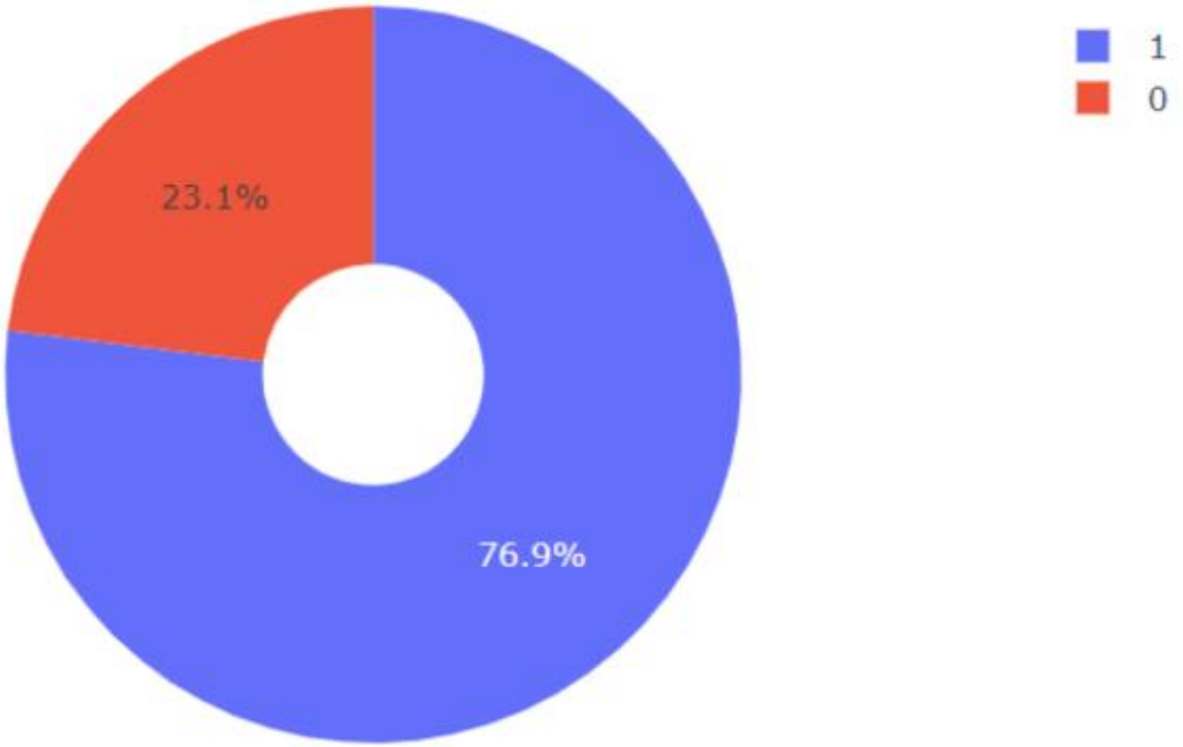


**KSC LC-39A achieved a 76.9% success rate while getting 23.1% failure rate.**

After visual analysis using the dashboard, we are able to obtain some insights to answer these questions:

- Which site has the highest launch success rate ?
  **KSC LC-39A**
- Which payloads range(s) has the highest launch success rate ?
  **2000 Kg – 10000 Kg**
- Which payloads range(s) has the low launch success rate ?
  **0 – 1000 Kg**
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate ?
  **FT**

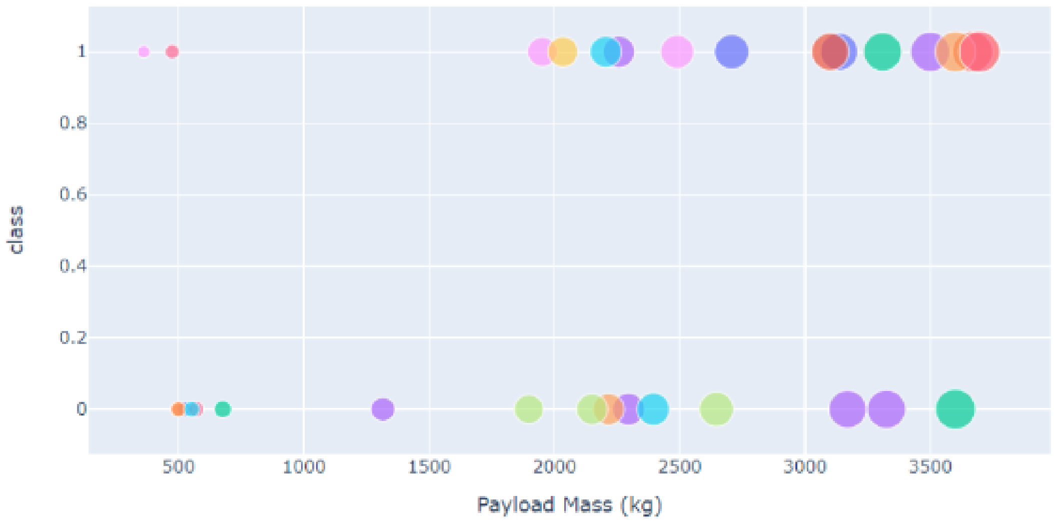
# Pie Chart Showing The Launch Site With The Highest Launch Success Ratio



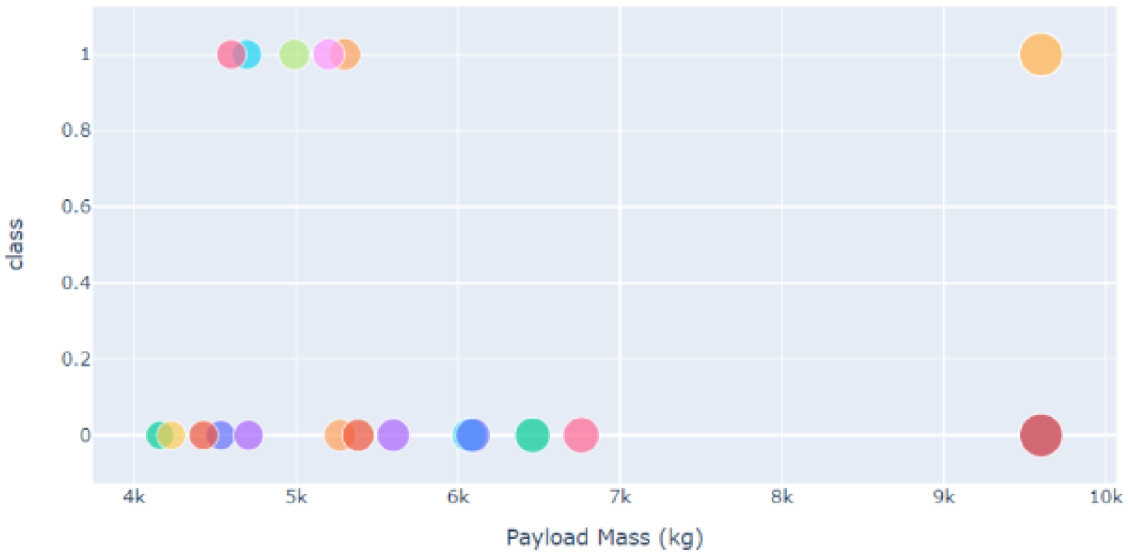
*KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate*

# 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 heavy weighted payloads***

# Predictive Analysis (Classification)

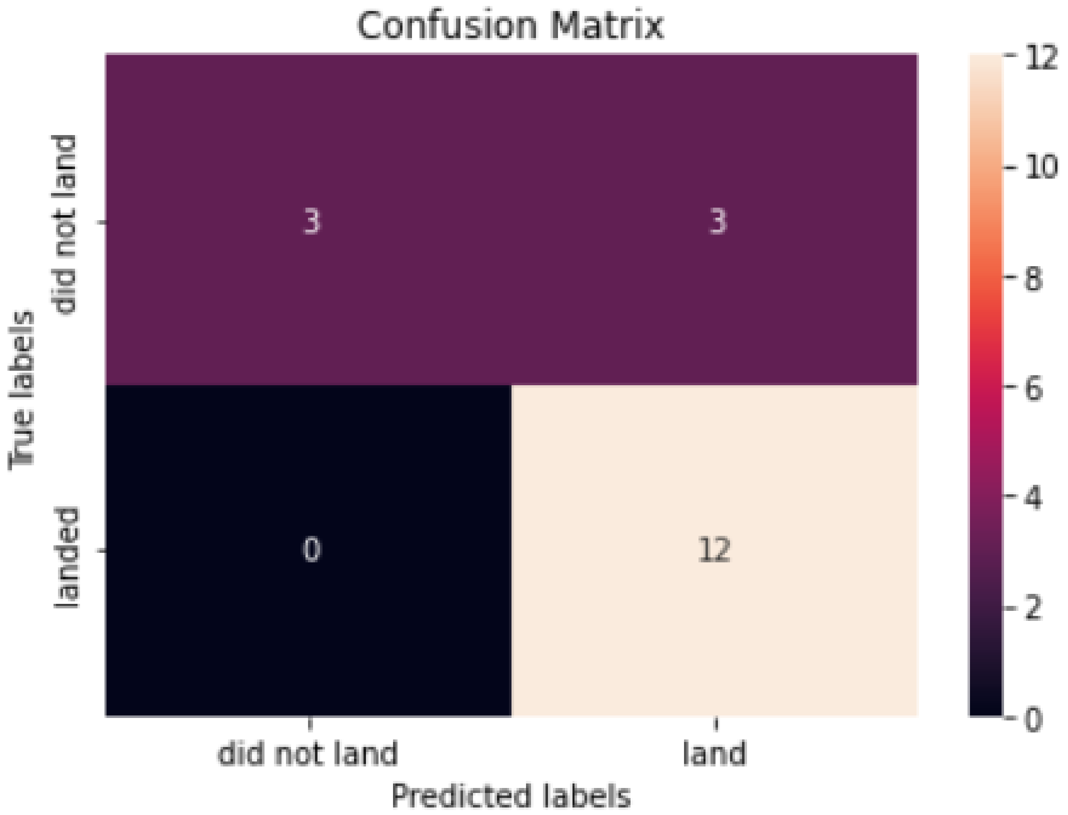
# Classification Accuracy

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

| Algorithm | Accuracy | Accuracy on Test Data |
|---|---|---|
| Logistic Regression | 0.846429 | 0.0833334 |
| SVM | 0.848214 | 0.0833334 |
| KNN | 0.848214 | 0.0833334 |
| Decision Tree | 0.875 | 0.0833334 |

# 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.



***Decision Tree***

# Conclusion

We can conclude that:

- 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.

# THE END