IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Adhish Nanda
14th January 2021

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using SpaceX API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis using Pandas and Matplotlib
  - Data Visualization with Folium
  - Interactive Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

The goal of this Capstone is to predict whether Falcon 9's First Stage will land successfully or not. SpaceX reuses its First Stage and thus, on its website has quoted a price of 62 million dollars, whereas, the next nearest cost of sending a rocket to space is 165 million dollars and above. So, if one could determine the successful landing of the First Stage, then one could determine the cost of a launch. This information can then be used by a competitor who wants to bid against SpaceX.

- Problems we want to find answers:-

    1. Will SpaceX reuse the current Falcon 9’s First Stage, as most unsuccessful landing are already planned in advance?

    2. What are the factors that influence successful landing of Falcon 9’s First Stage?

    3. What effects does such factors or their combinations have on First Stage’s landing?

    4. What are the required outcomes that predict a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Using SpaceX REST API and web scraping Wikipedia pages
- Perform data wrangling
    - Applied One-Hot-Encoding to categorical attributes
- 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

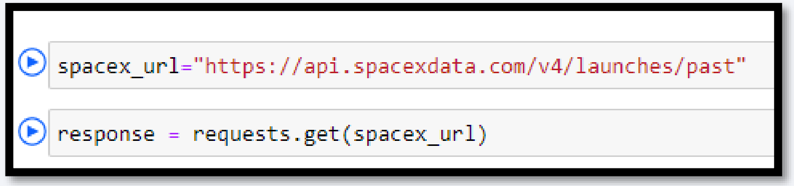
- Data Sets were collected in two parts:

  1. In first part, the SpaceX launch data was gathered from SpaceX REST API. This data was about launches, including information about the rockets used, launch specification, landing specifications and landing outcome. We use this data to predict whether space will attempt to land a rocket or not.

  2. The remaining data regarding Falcon 9's launch was collected through Web Scraping related Wikipedia pages. Python's BeautifulSoup package was used to web scrape some HTML tables that contain valuable Falcon 9 launch records.

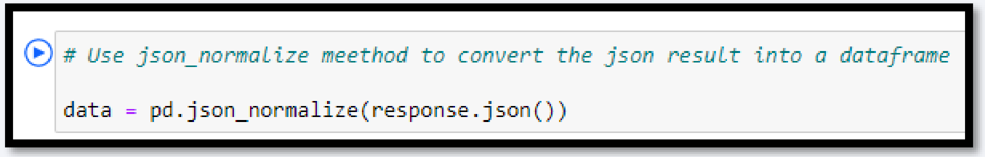
Data Collection – SpaceX API

1. Request and parse SpaceX launch data using GET, decode the response content as a Json and turn it into a Pandas dataframe.
2. Use the API again to get information about the launches using the IDs given for each launch & clean the data.
3. Filter the dataframe to include only Falcon 9 launches and deal with missing values.

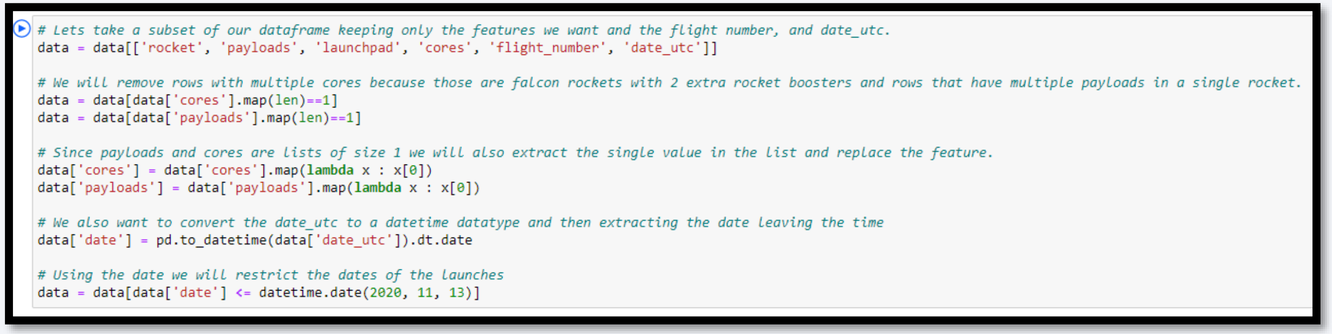


```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

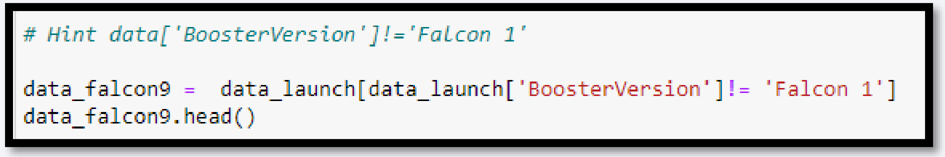
```
response = requests.get(spacex_url)
```



```
# Use json_normalize meethod to convert the json result into a dataframe

data = pd.json_normalize(response.json())
```



```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```



```
# Hint data['BoosterVersion']!='Falcon 1'

data_falcon9 =  data_launch[data_launch['BoosterVersion']!= 'Falcon 1']
data_falcon9.head()
```

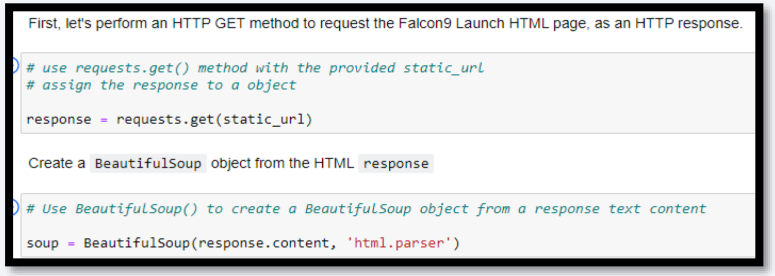
Notebook Link: https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Data%20Collection%20using%20API.ipynb

Data Collection - Scraping

1. Request the Falcon 9 Launch Wiki page from its URL your web scraping process using key phrases and flowcharts

2. Extract all column/variable names from the HTML table header.

3. Create a data frame by parsing the launch HTML tables.

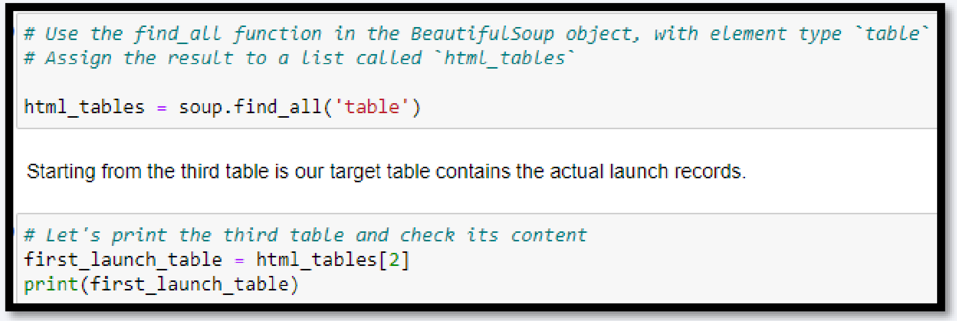


First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object

response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

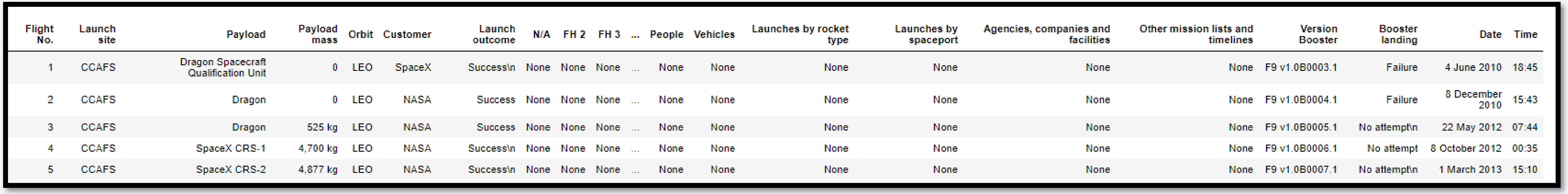
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(response.content, 'html.parser')
```



```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

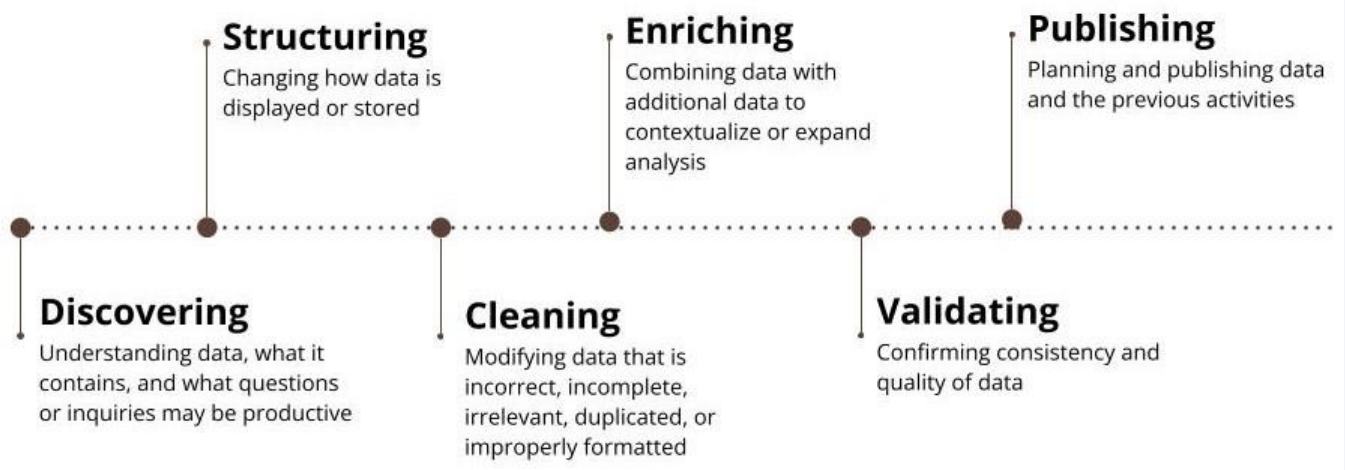
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```



Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	N/A	FH 2	FH 3	...	People	Vehicles	Launches by rocket type	Launches by spaceport	Agencies, companies and facilities	Other mission lists and timelines	Version Booster	Booster landing	Date	Time
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	None	None	None	...	None	None	None	None	None	None	F9 v1.0B0003.1	Failure	4 June 2010	18:45
2	CCAFS	Dragon	0	LEO	NASA	Success	None	None	None	...	None	None	None	None	None	None	F9 v1.0B0004.1	Failure	8 December 2010	15:43
3	CCAFS	Dragon	525 kg	LEO	NASA	Success	None	None	None	...	None	None	None	None	None	None	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	None	None	None	...	None	None	None	None	None	None	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	None	None	None	...	None	None	None	None	None	None	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

Notebook Link: https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

Data Wrangling

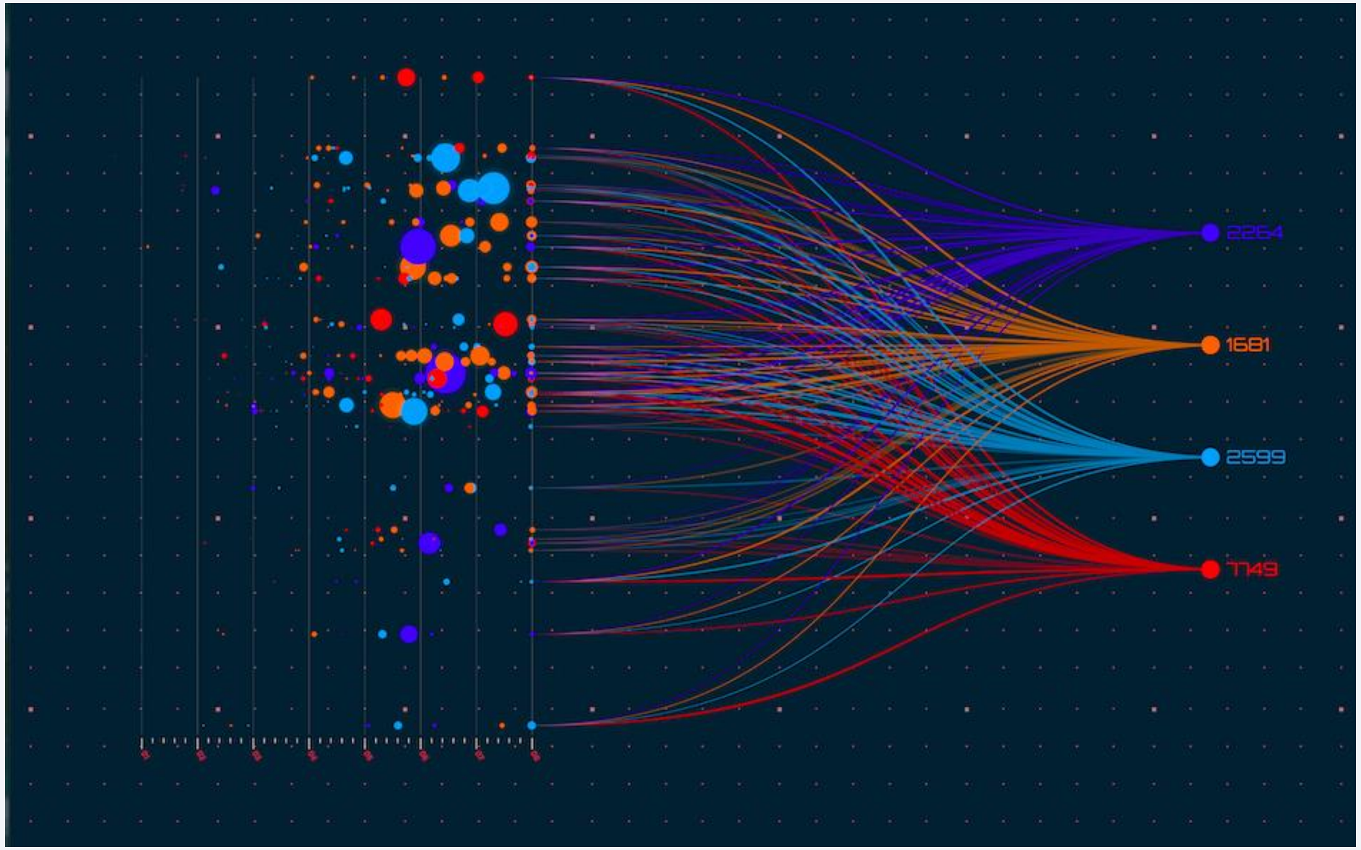

Structuring
Changing how data is displayed or stored
Enriching
Combining data with additional data to contextualize or expand analysis
Publishing
Planning and publishing data and the previous activities
Discovering
Understanding data, what it contains, and what questions or inquiries may be productive
Cleaning
Modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted
Validating
Confirming consistency and quality of data

- Calculated the number of launches on each site.
- Calculated the number and occurrence of each orbit.
- Calculated the number and occurrence of mission outcome per orbit type.
- Created a landing outcome label from Outcome column.

Notebook Link: https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

- Visualized the relationship between Flight Number and Launch Site, Payload and Launch Site, Success Rate of each Orbit type, Flight Number and Orbit type, Payload and Orbit type, and Launch Success's yearly trend.



Notebook Link: https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Exploratory%20Analysis%20using%20SQL.ipynb

EDA with SQL

- We loaded the SpaceX data into IBM DB2 database and perform queries to fetch the following data:
  - Names of the unique launch sites in the space mission.
  - Records where launch sites begin with the string 'CCA’.
  - Total payload 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 greater than 4000 but less than 6000.
  - Total number of successful and failure mission outcomes.
  - Names of the booster versions which have carried the maximum payload mass.
  - Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.
  - Ranking 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.

Notebook Link: https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Exploratory%20Analysis%20using%20SQL.ipynb

Build an Interactive Map with Folium

- Marked the locations and proximities of launch sites.
  - Used folium.Circle and folium.Marker to add a highlighted circle area with a text label on a specific coordinate.
- Calculated the distances between a launch site to its proximities.
  - Used folium.PolyLine to show the distance between a launch site and other locations.
- Discovered patterns while exploring the map.
  - We found out whether all launch sites are in proximity to equator line, coastline, railways and cities.
- Marked the success/failed launches for each site on the map.
- To choose an optimal launch site, we converted the landing outcome attribute into 0 and 1. 0 being failed outcome and I being successful. Then we used color-labeled marker_cluster to identify sites with high success rates.
- GitHub URL of the same:-

https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We built an Interactive Dashboard with Plotly and Dash.
- Added a pie chart to show the total successful launches count for all sites. If a specific launch site was selected, show the Success vs. Failed counts for that site.
- Added a scatter chart to show the correlation between payload and launch success.
- GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose:-

https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Dashboard.py

Predictive Analysis (Classification)

- Loaded the data, created a column for the Class using Numpy and Pandas, standardized the data and split it into training and testing sets.
- The training data is then divided into validation data, a second set used for testing the model; then the models are trained and hyperparameters are selected using the function GridSearchCV.
- We used ‘Accuracy’ as the scoring metric for our classification models. Used Feature Engineering to improve the model.
- GitHub URL of your completed predictive analysis lab:-

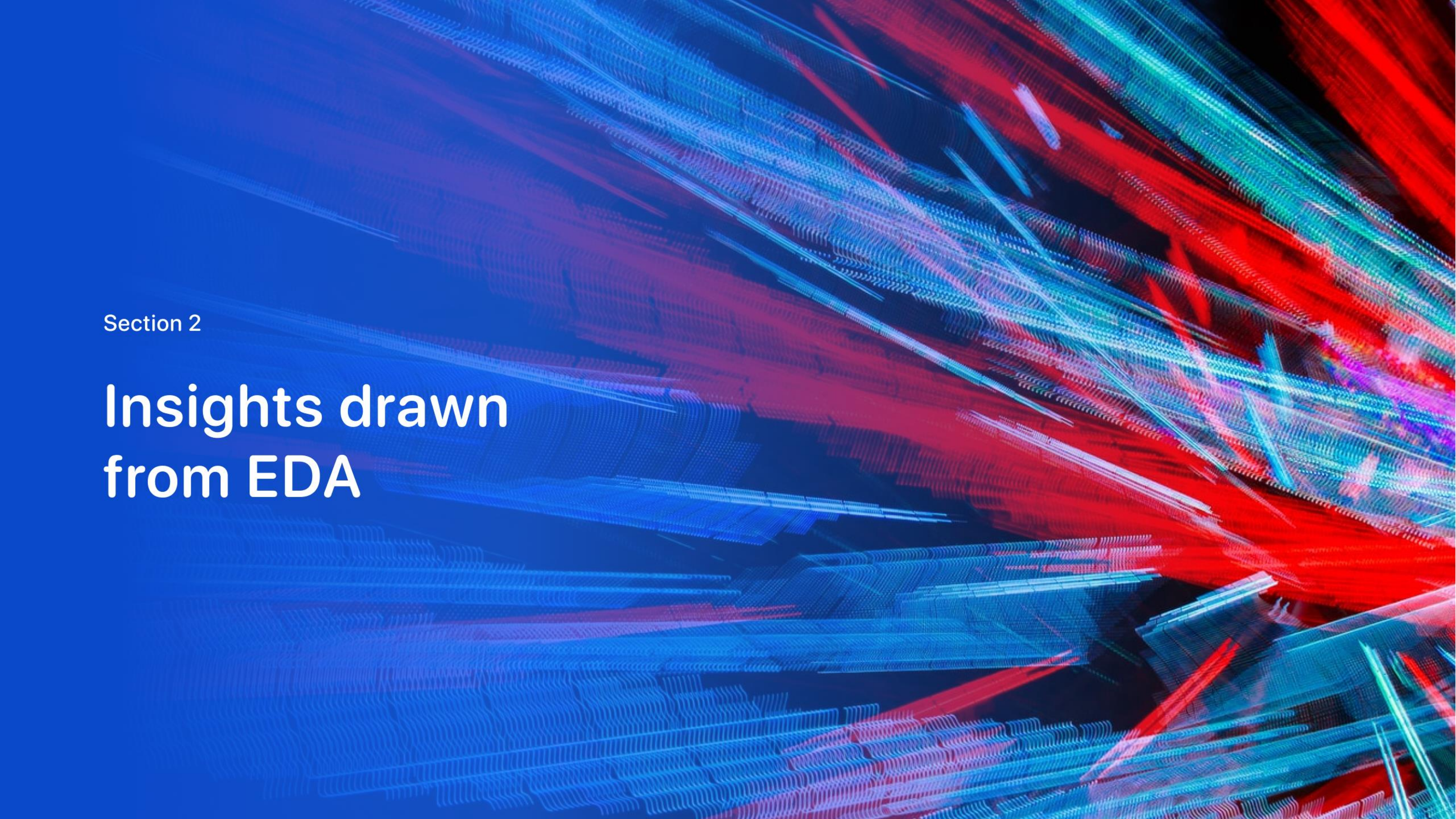
https://github.com/adhishnanda/IBM-Data-Science-Capstone-Project/blob/main/Machine%20Learning%20Prediction.ipynb

Results

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

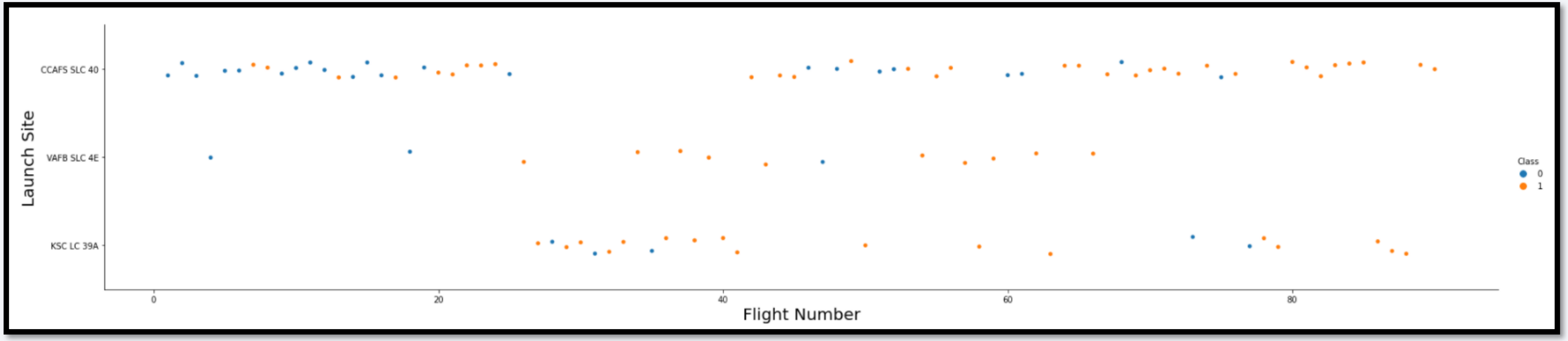
Section 2

Insights drawn from EDA



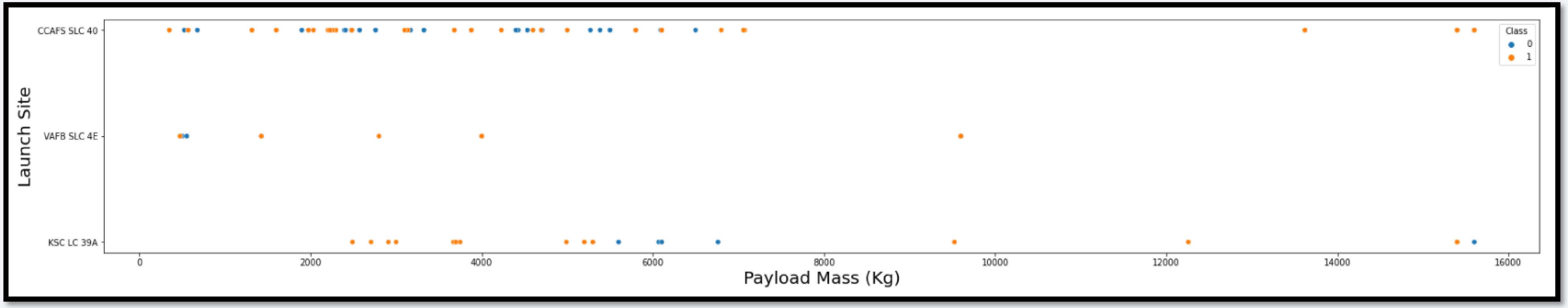
Flight Number vs. Launch Site

- Above flight numbers greater than 30, the success rate of the mission at a launch site is positively correlated to the number of flights from that site.
- 0 = Failed Mission (Blue), 1 = Successful Mission (Orange)



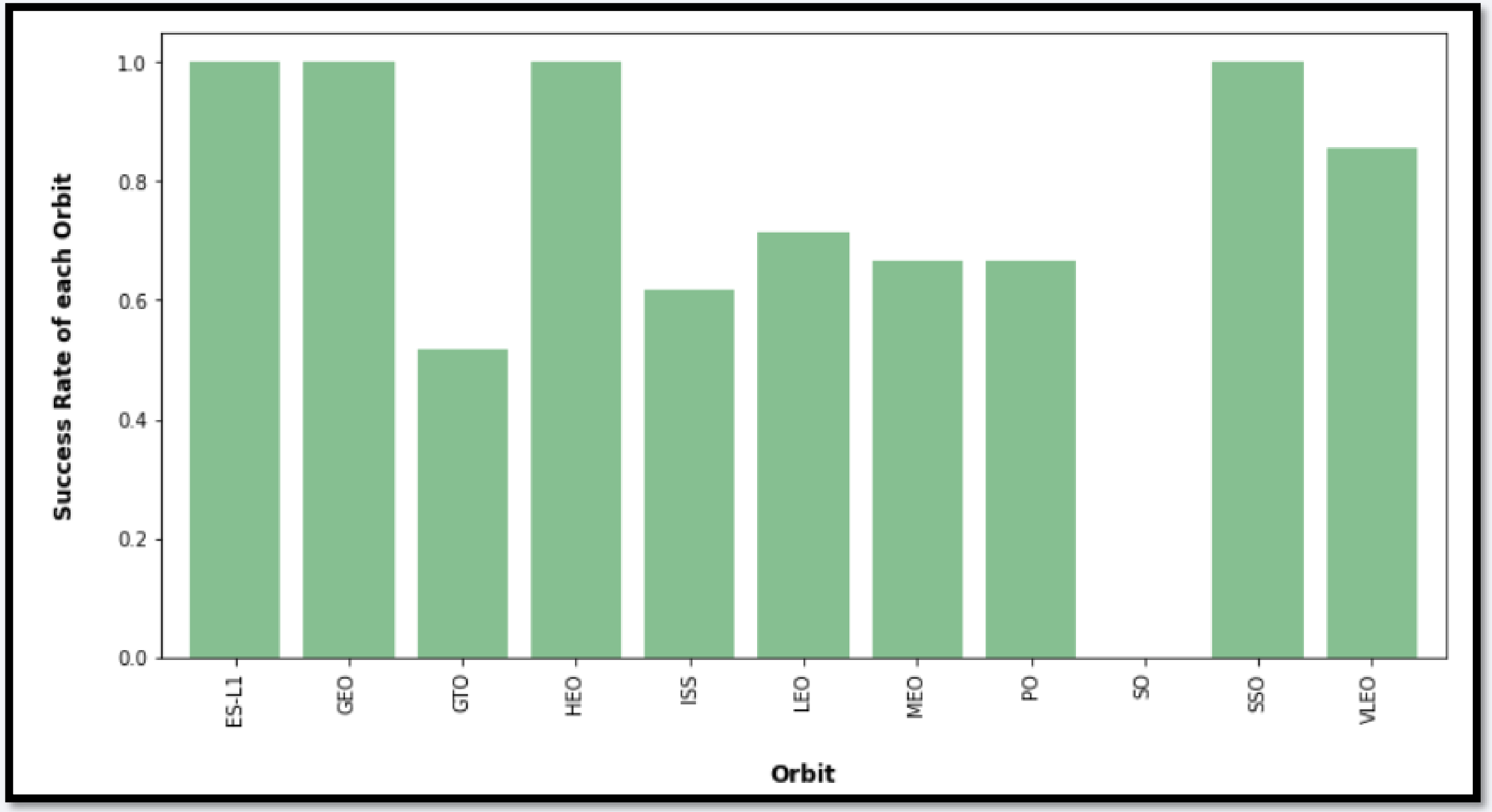
Payload vs. Launch Site

- VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000 Kgs).
- The greater the payload mass at site CCAFS SLC 40 (> 7000 Kgs), the higher the success rate.
- But its difficult to figure out whether Launch Site and Payload are related. There is no clear pattern.



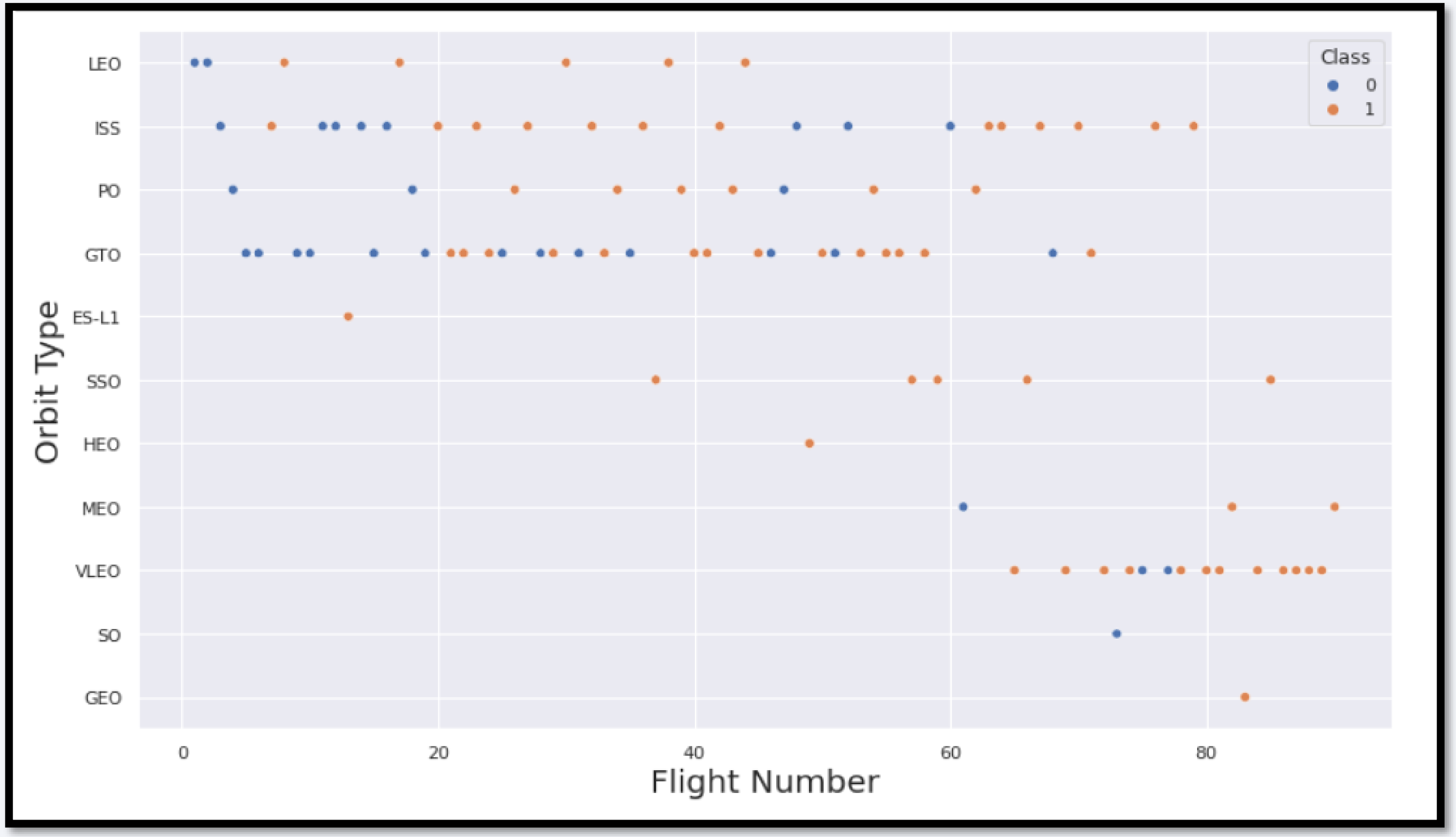
Success Rate vs. Orbit Type

- Missions to Orbits ES-L1, GEO, HEO and SSO have the highest and almost 100% success rates.



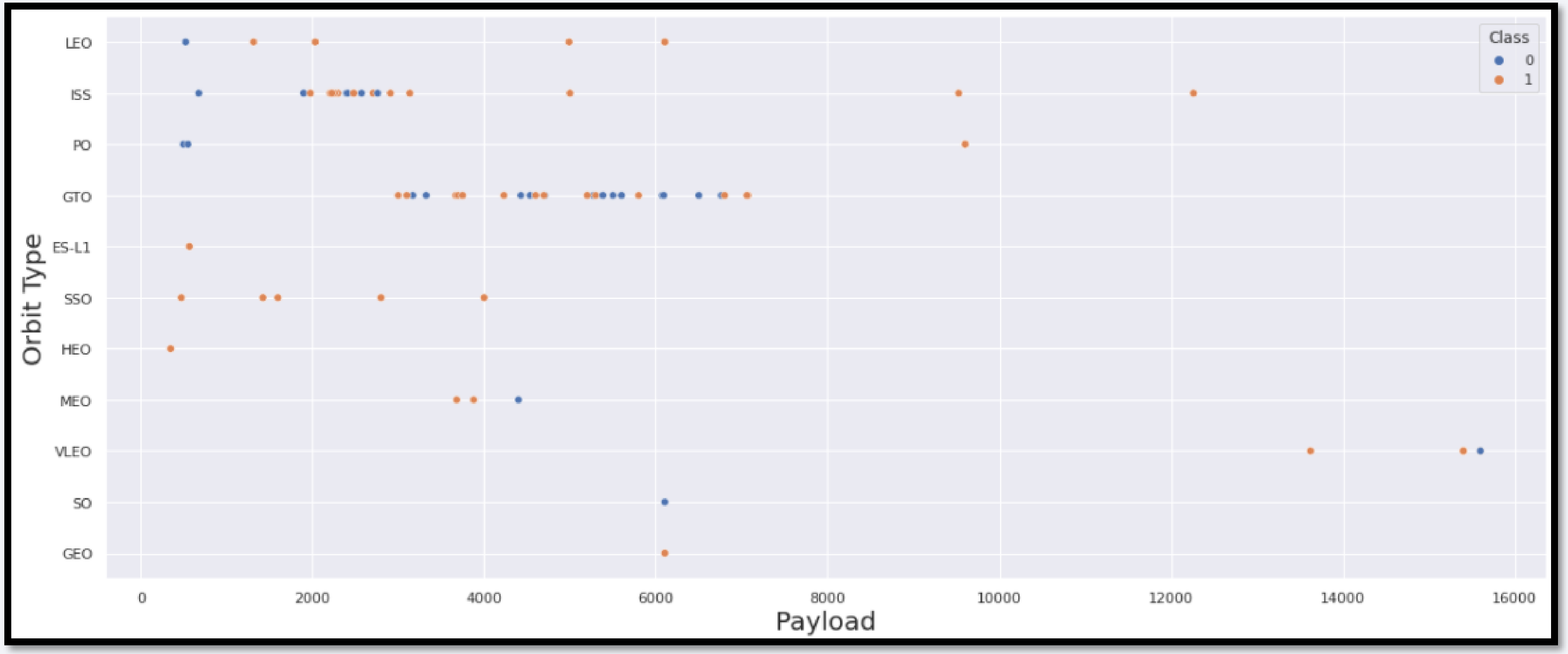
Flight Number vs. Orbit Type

- We observe that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



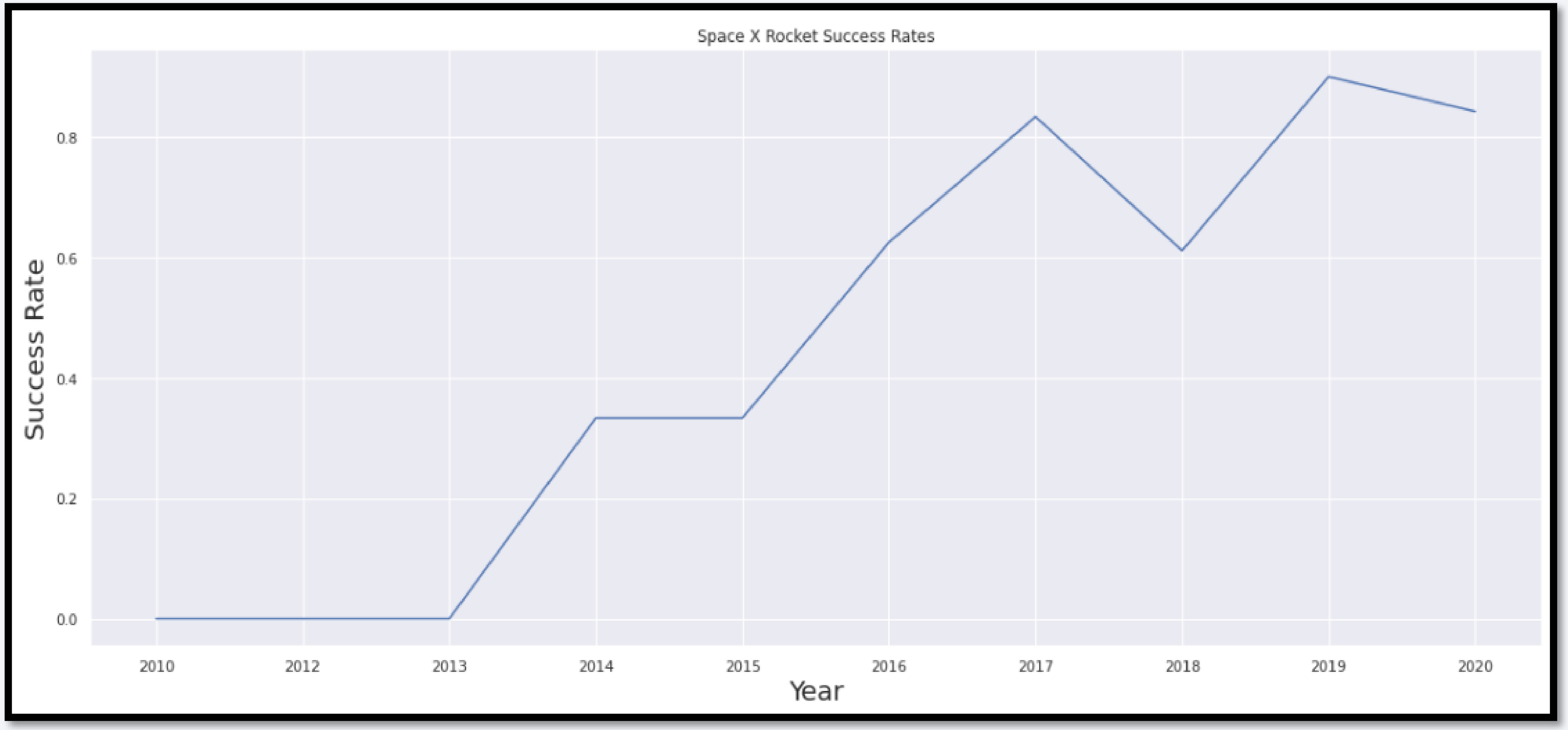
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS Orbits. However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are equally present.



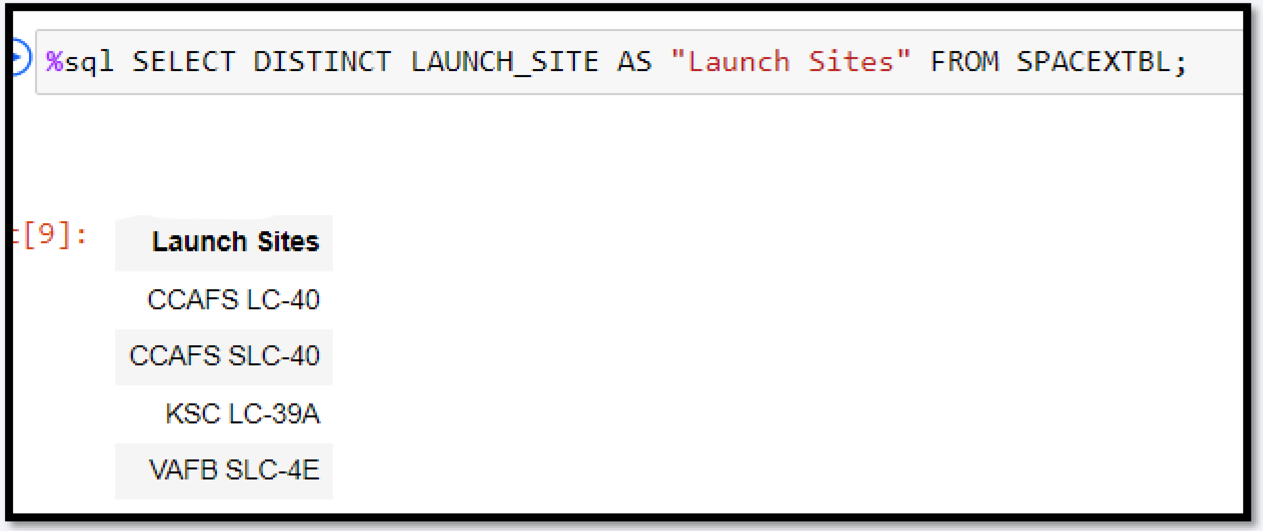
Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

- Found the names of the unique launch sites from the SpaceX data loaded as SPACEXTBL on IBM DB2 using the keyword ‘DISTINCT’.

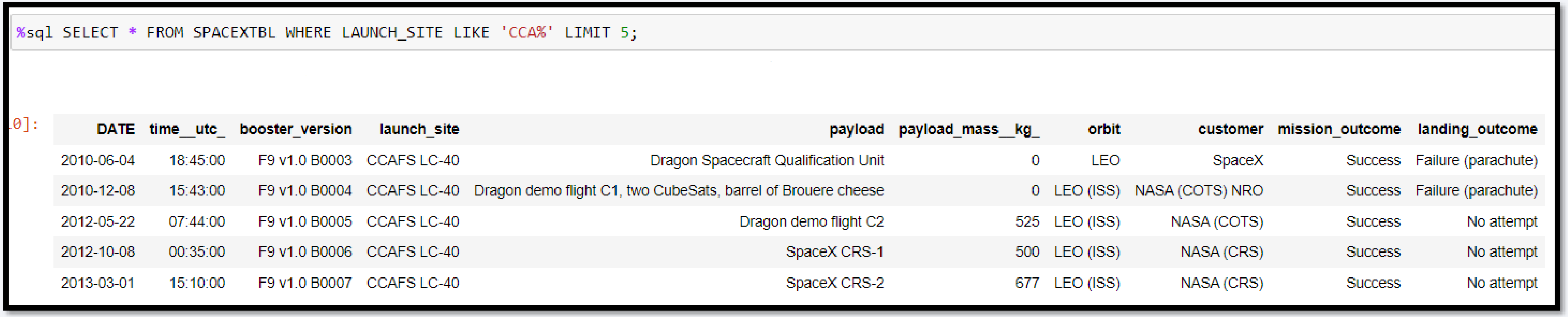


```
%sql SELECT DISTINCT LAUNCH_SITE AS "Launch Sites" FROM SPACEXTBL;
```

Launch Sites
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Found 5 records where launch sites begin with `CCA`
- The query result shows 5 launch sites starting with ‘CCA’ and all the information related to them.



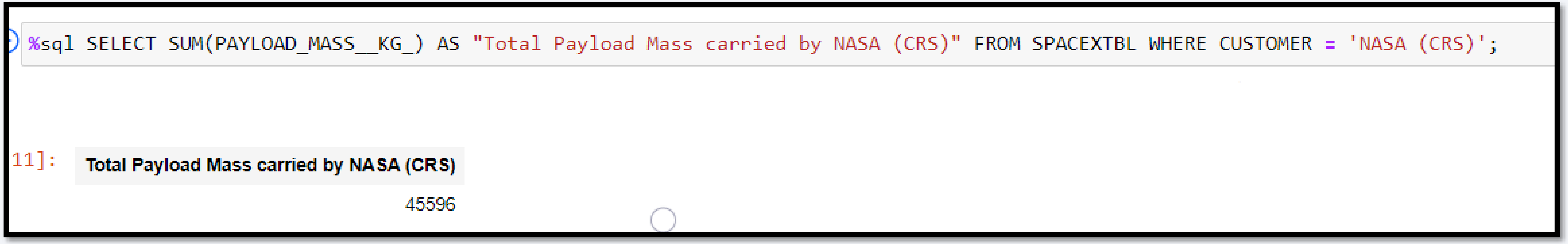
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

0]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
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
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculated the total payload carried by boosters from NASA
- The total payload mass carried by NASA (CRS) is 45596 Kgs.

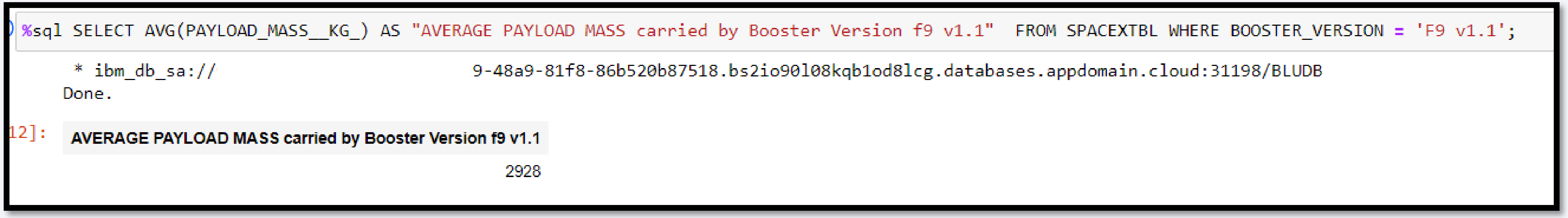


```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass carried by NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

```
11]:  Total Payload Mass carried by NASA (CRS)
                                         45596
```

Average Payload Mass by F9 v1.1

- Calculated the average payload mass carried by booster version F9 v1.1
  - The query result shows that the average payload carried by F9 v1.1 is 2928 Kgs.



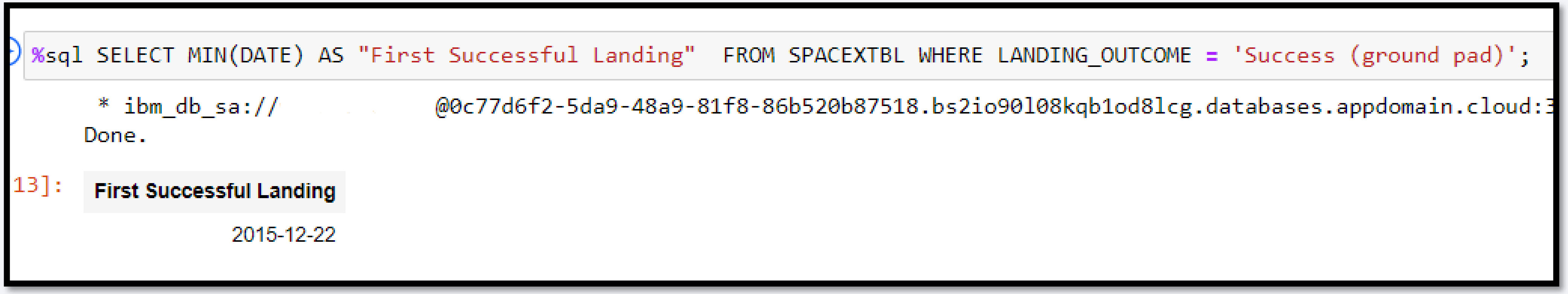
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "AVERAGE PAYLOAD MASS carried by Booster Version f9 v1.1"  FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';

 * ibm_db_sa://                        9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

12]:

AVERAGE PAYLOAD MASS carried by Booster Version f9 v1.1
2928

First Successful Ground Landing Date

- Found the dates of the first successful landing outcome on ground pad.
    - The first successful landing was on 22nd December 2015.



```
%sql SELECT MIN(DATE) AS "First Successful Landing"  FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

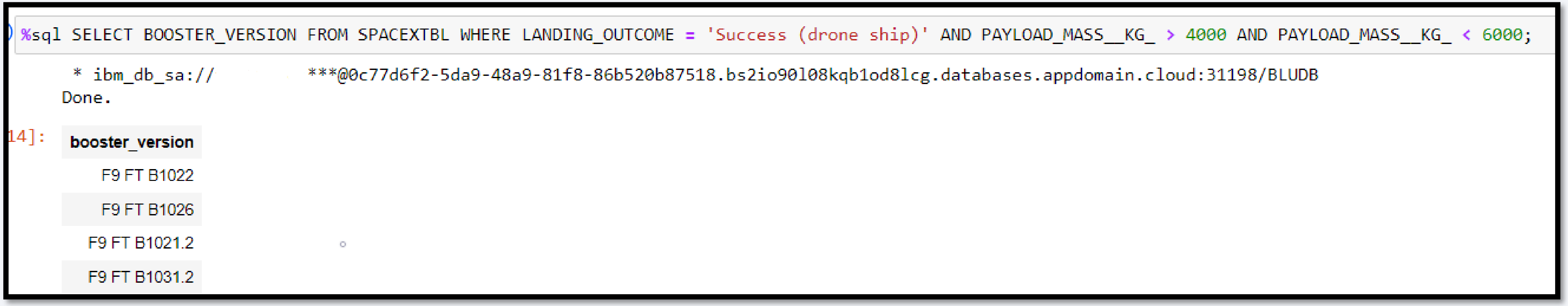
```
 * ibm_db_sa://          @0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3
Done.
```

13]:

First Successful Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Listed the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
    - The boosters that successfully landed on drone ship with a payload mass greater than 4000 kg but less than 6000 kg are F9 FT B1022, B1026, B1021.2 and B1031.2 resp.



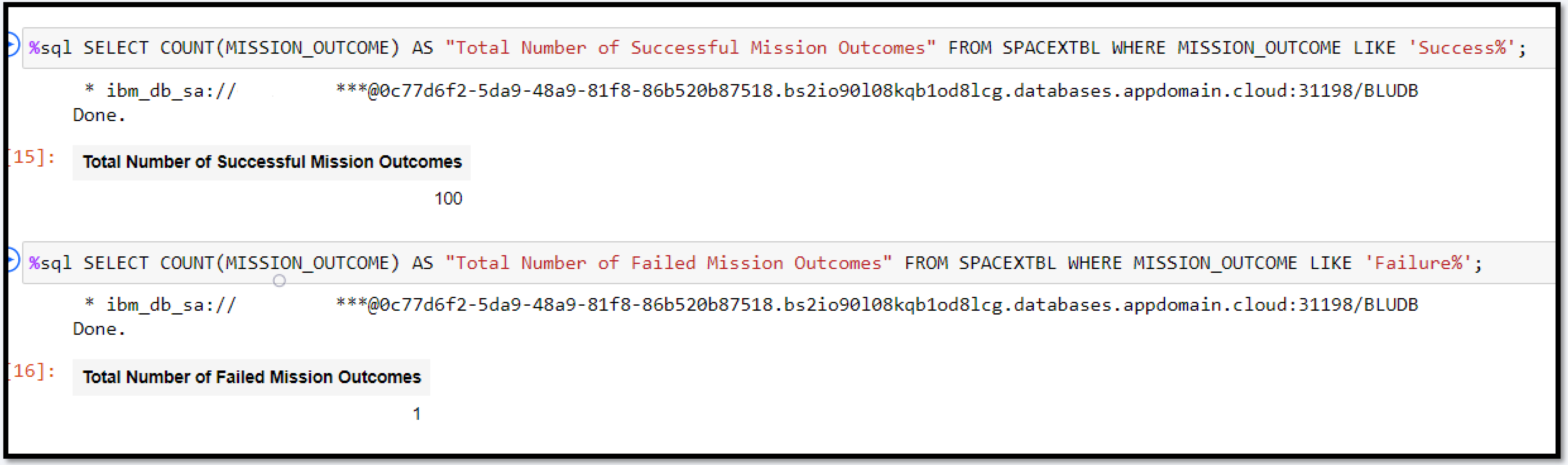
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

 * ibm_db_sa://          ***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

14]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculated the total number of successful and failure mission outcomes
  - Total successful missions are 100 and total failed missions is 1.



```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Total Number of Successful Mission Outcomes" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://          ***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

[15]:

Total Number of Successful Mission Outcomes
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Total Number of Failed Mission Outcomes" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%';
```

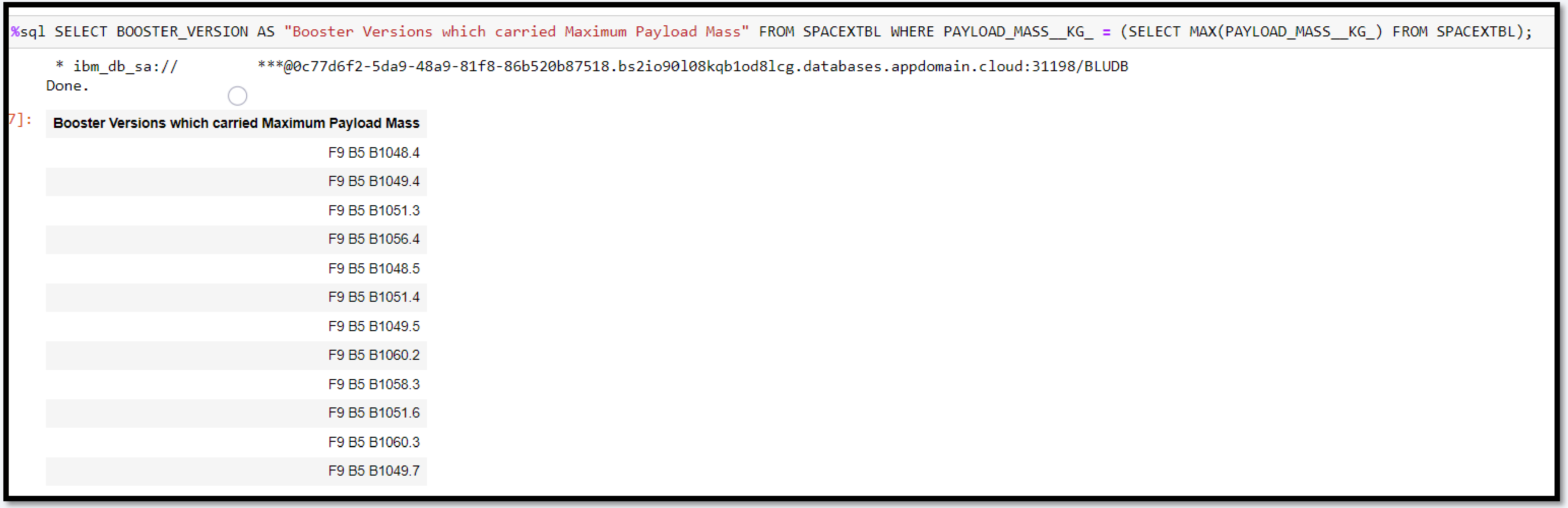
```
 * ibm_db_sa://          ***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

[16]:

Total Number of Failed Mission Outcomes
1

Boosters Carried Maximum Payload

- Listed the names of the booster which have carried the maximum payload mass.
  - These are the boosters that carried maximum payload mass. We used a subquery here to extract them.



```
%sql SELECT BOOSTER_VERSION AS "Booster Versions which carried Maximum Payload Mass" FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

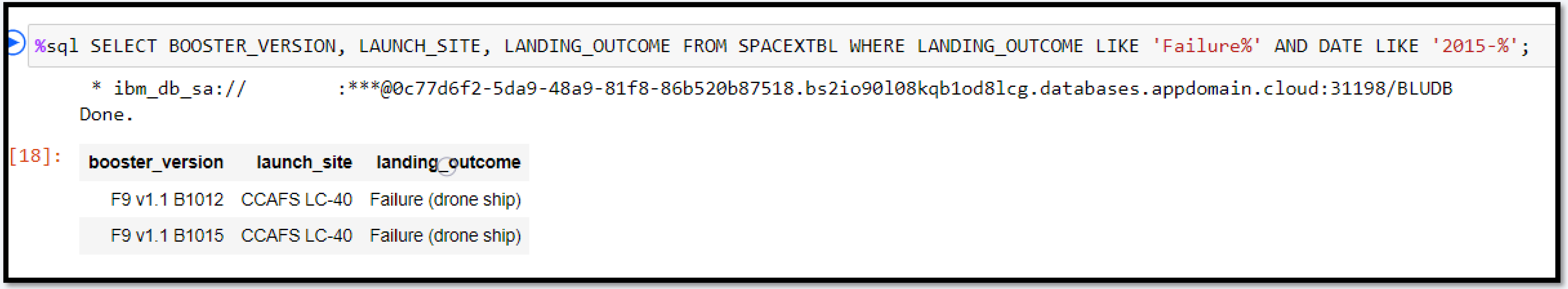
```
 * ibm_db_sa://        ***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

7]:

Booster Versions which carried Maximum Payload Mass
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

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



```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Failure%' AND DATE LIKE '2015-%';
```

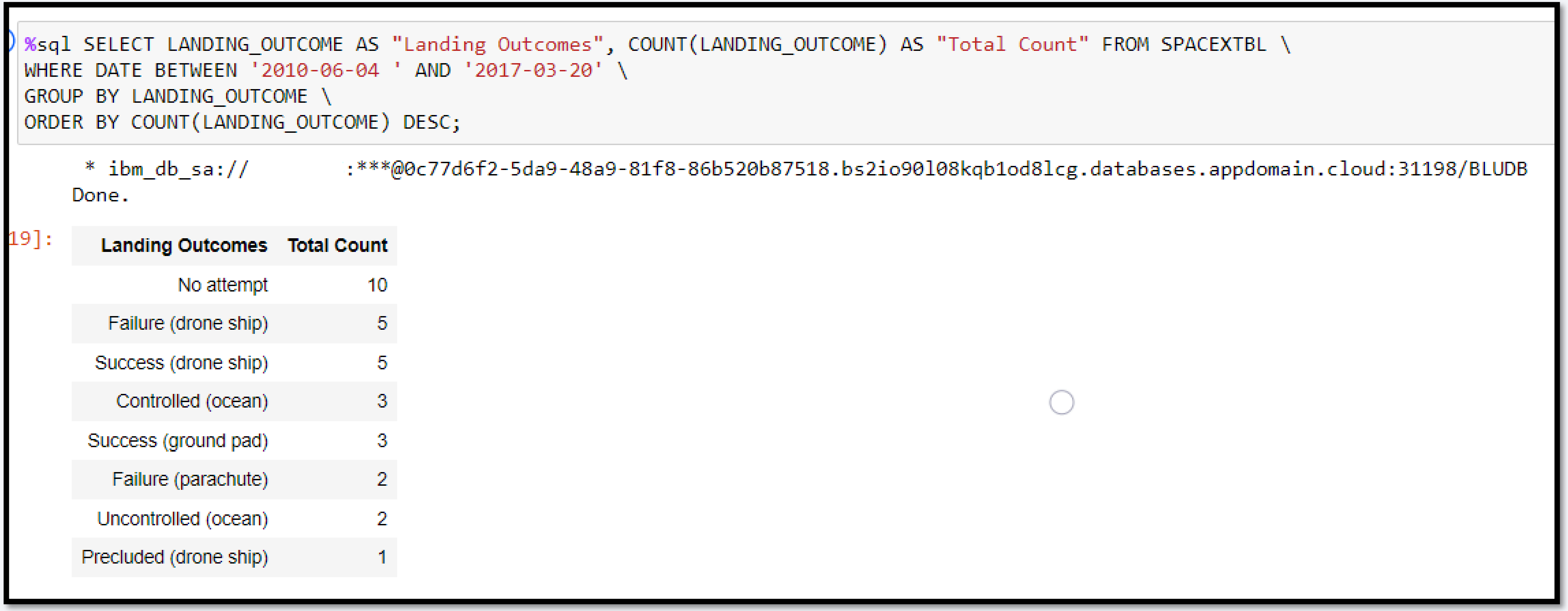
```
 * ibm_db_sa://        :***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

[18]:

booster_version	launch_site	landing_outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Ranked 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.



```
%sql SELECT LANDING_OUTCOME AS "Landing Outcomes", COUNT(LANDING_OUTCOME) AS "Total Count" FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04 ' AND '2017-03-20' \
GROUP BY LANDING_OUTCOME \
ORDER BY COUNT(LANDING_OUTCOME) DESC;
```

```
 * ibm_db_sa://        :***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/BLUDB
Done.
```

19]:

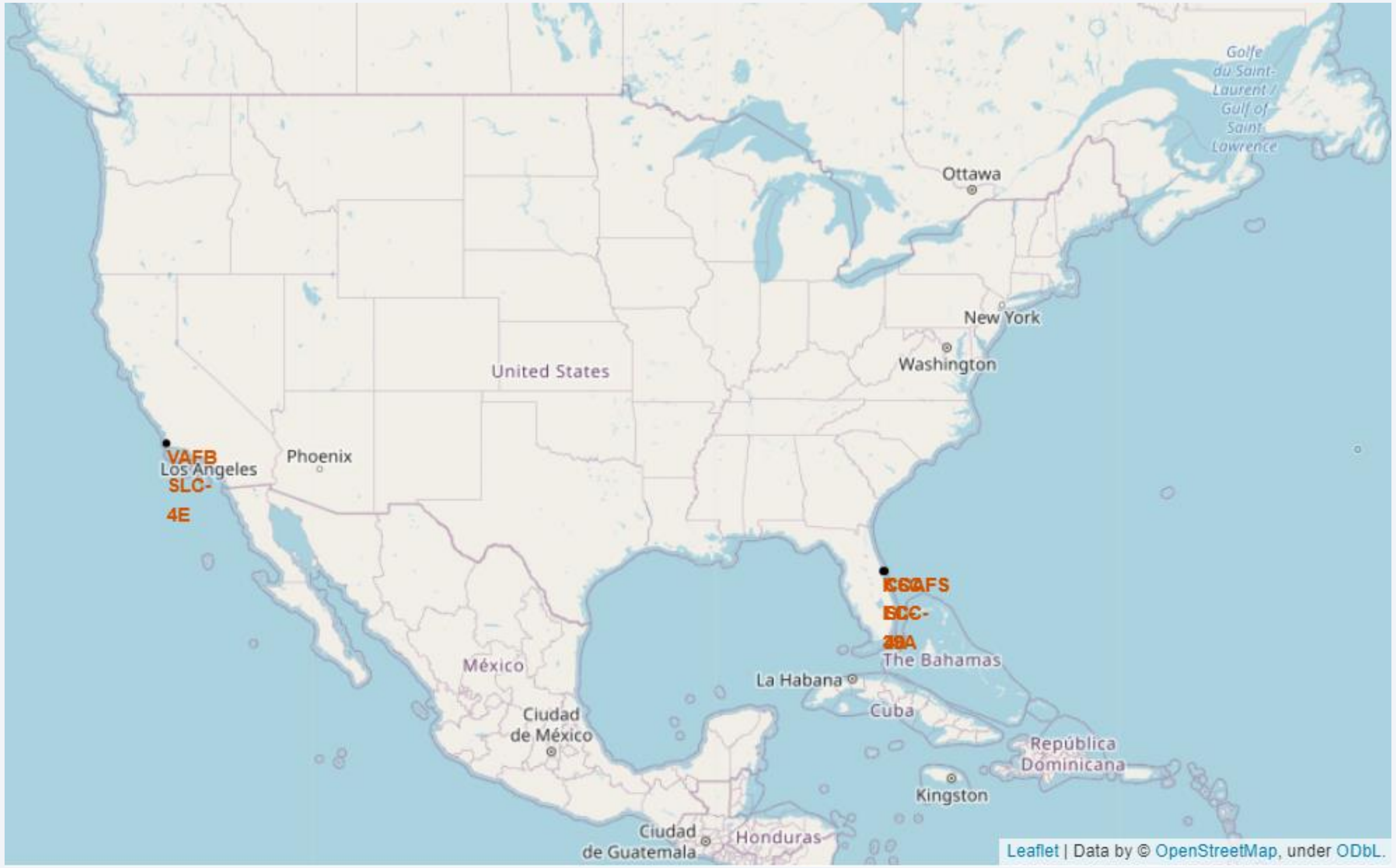
Landing Outcomes	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

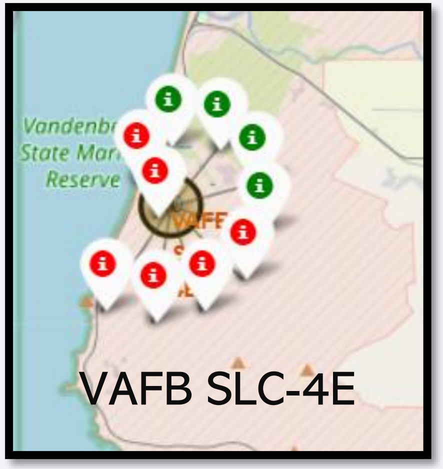
All SpaceX launch sites

- All launch sites are located in America. One on West Coast (VAFB SLC-4E), and other three on East Coast (KSC LC-39A, CCAFS SLC-40 and CCAFS LC-40)

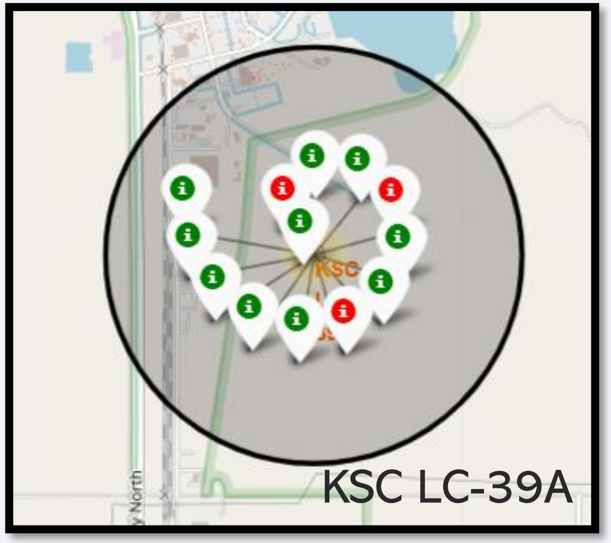


Markers with Successful/Failed launches at each site

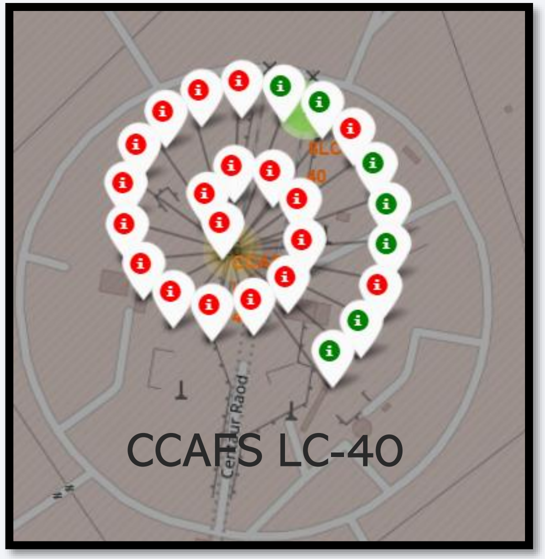
- Green Markers reveal successful landings.
- Red Markers reveal failed landings.



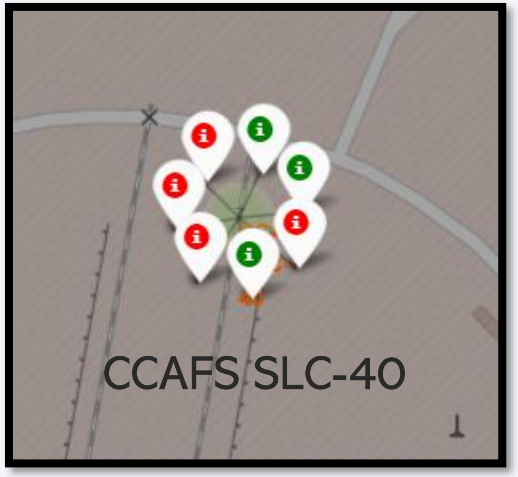
VAFB SLC-4E



KSC LC-39A



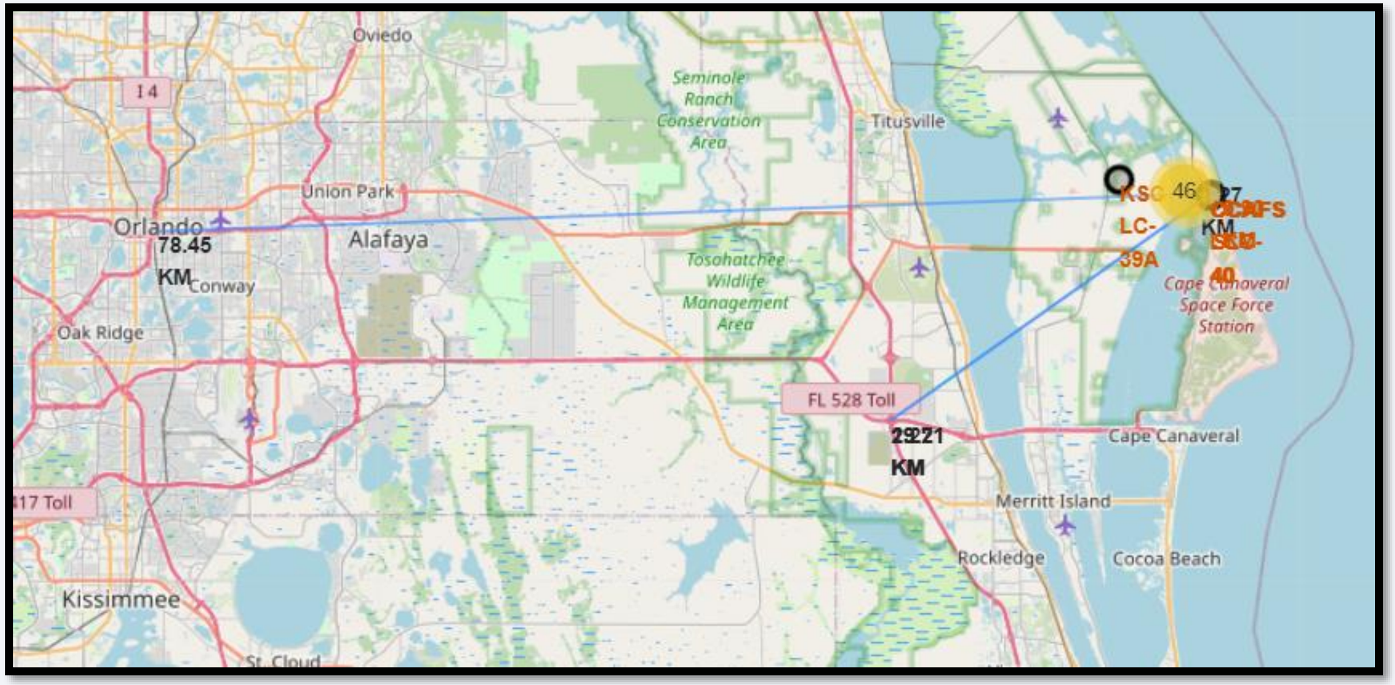
CCAFS LC-40

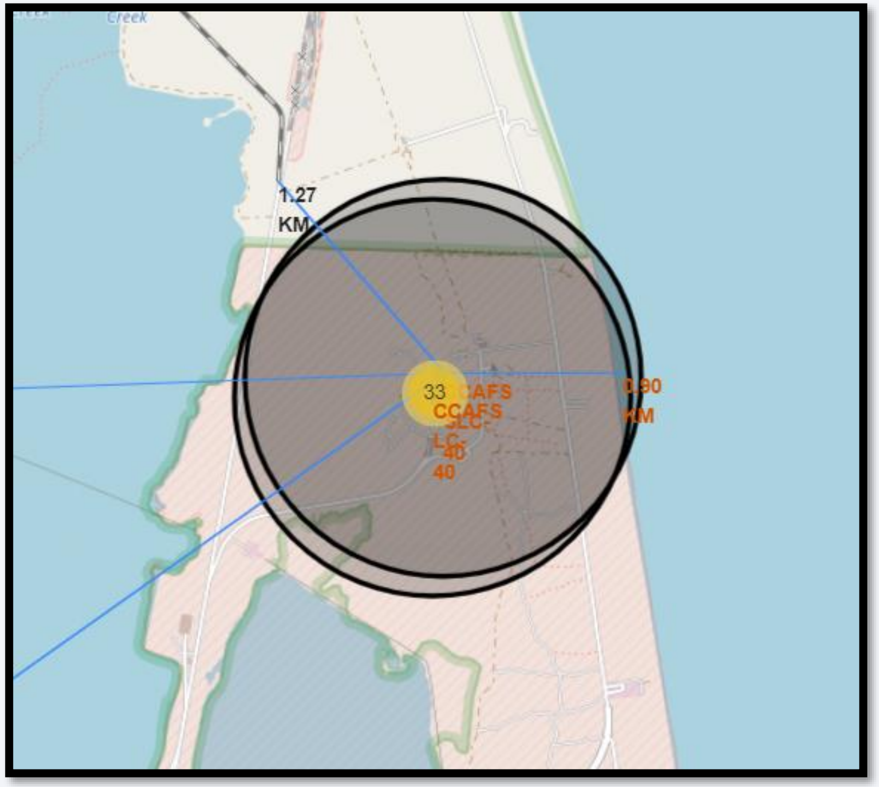


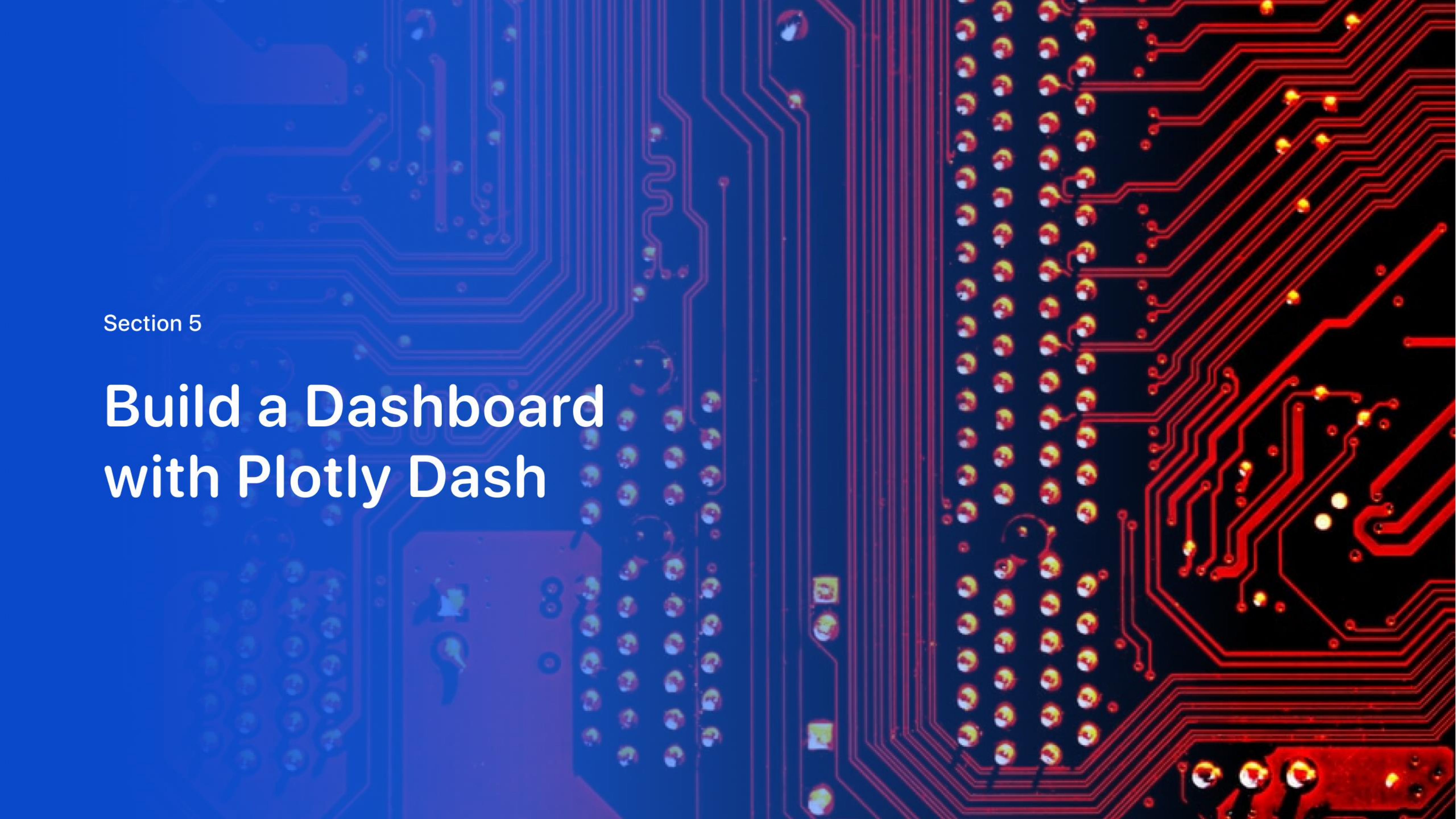
CCAFS SLC-40

Launch sites and their proximities

- We can see that launch sites are usually closer to coastlines and railways, and away from cities and highways.





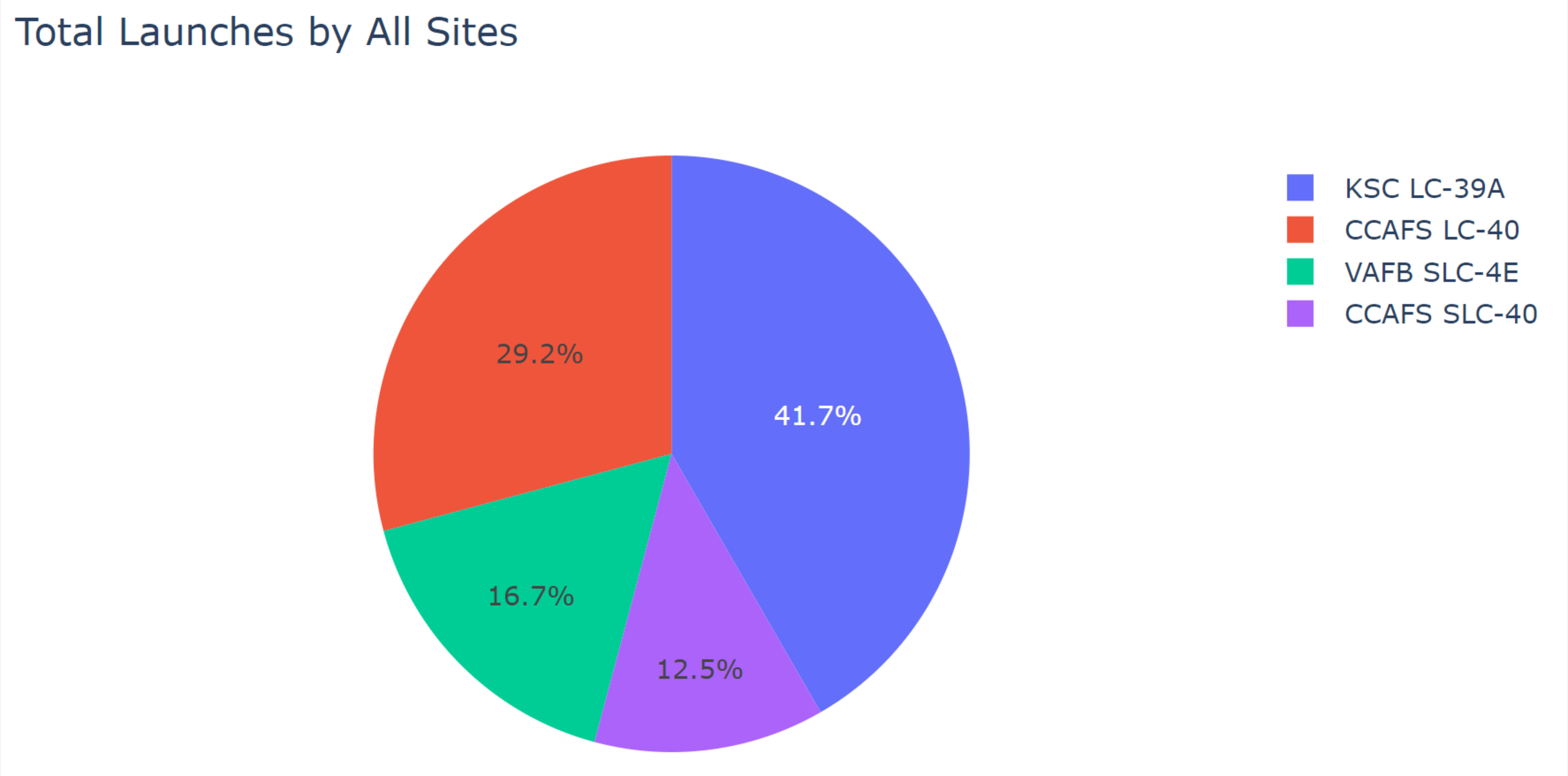


Section 5

Build a Dashboard with Plotly Dash

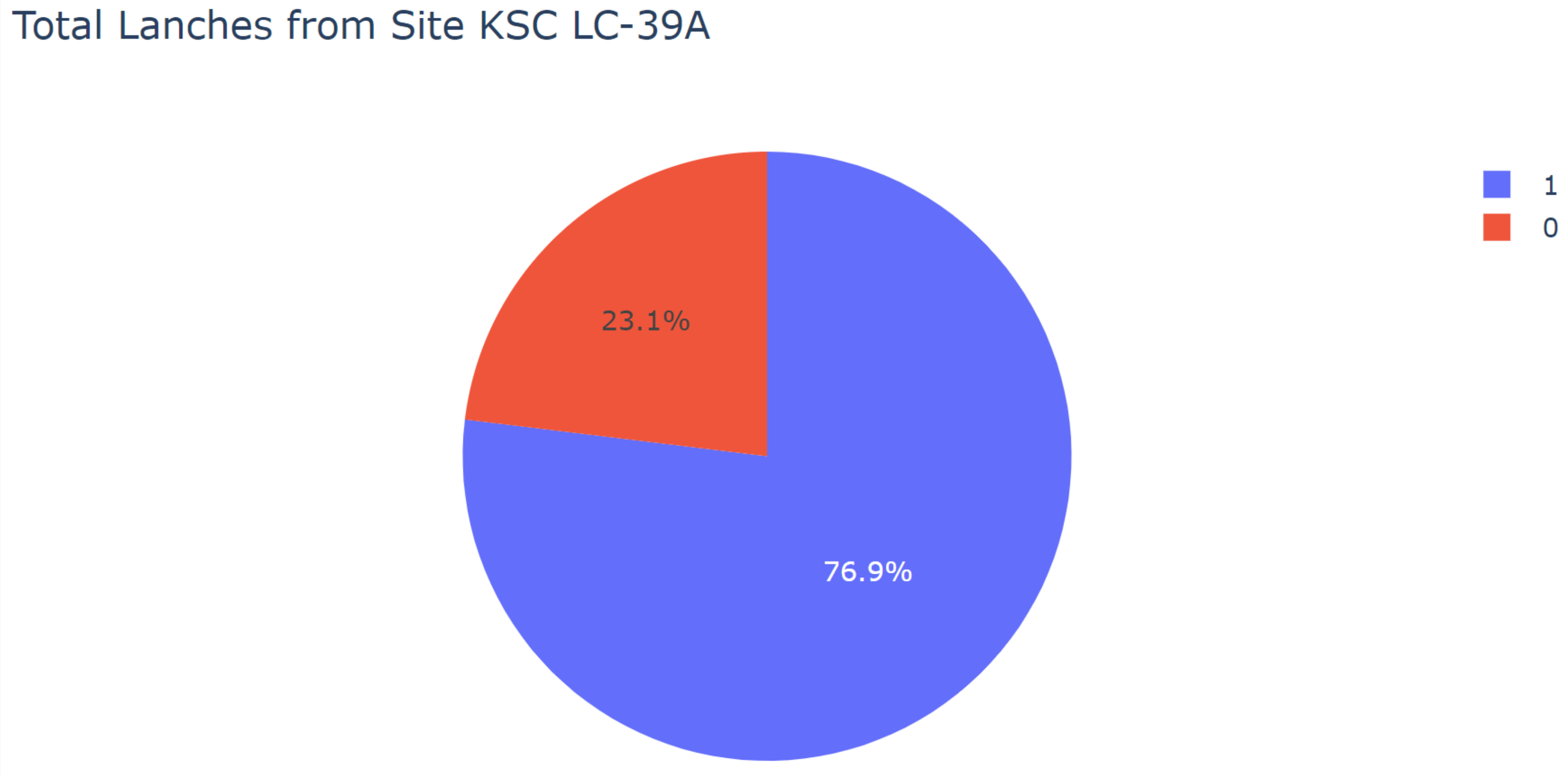
Percentage of missions from each launch site

- KSC LC-39A has the most missions to its name.

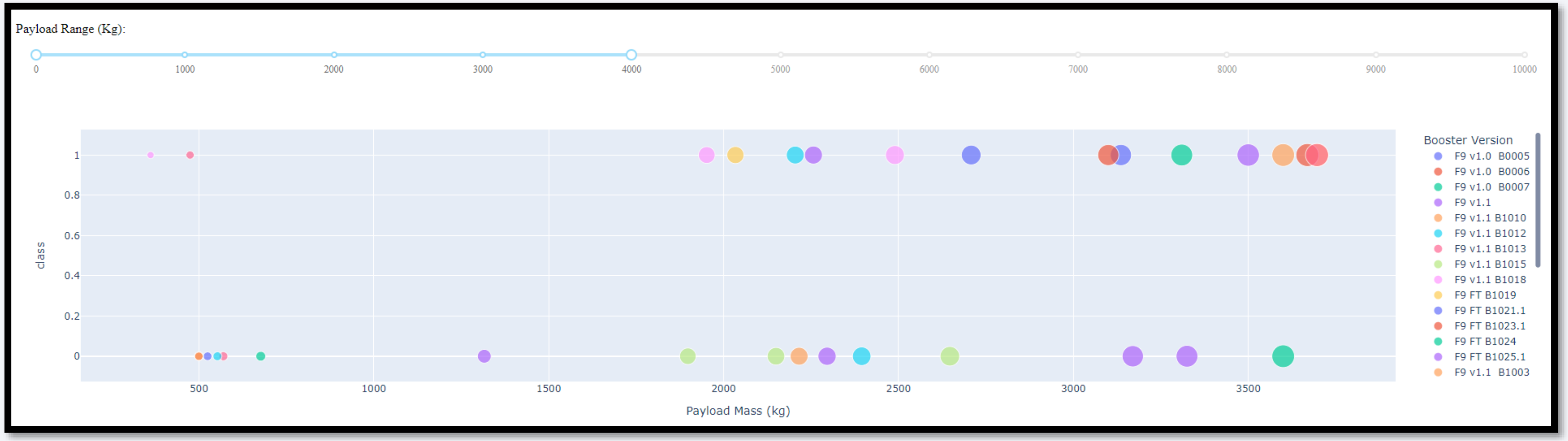


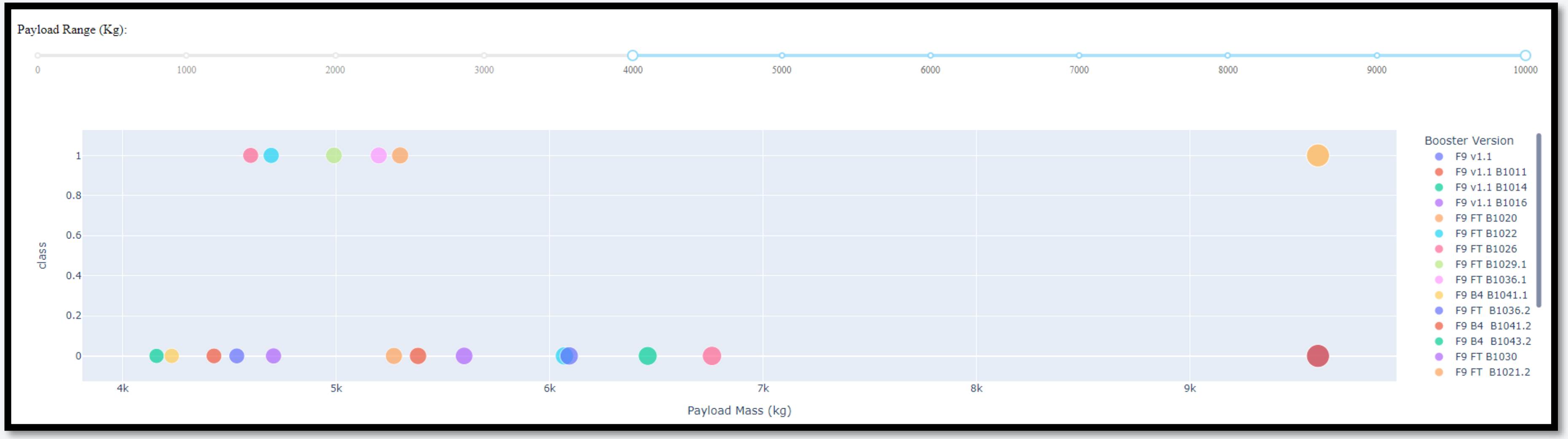
Launch site with best success/failure ratio

- 76.9 missions out of 100 are successful when launched from KSC LC-39A, with only 23.1 missions failing.



Scatterplot of Payload vs Launch Outcome for all sites





- Success rates for payload less than 4000 kgs is higher than the success rate of payloads above 4000 kgs.

Section 6

Predictive Analysis (Classification)

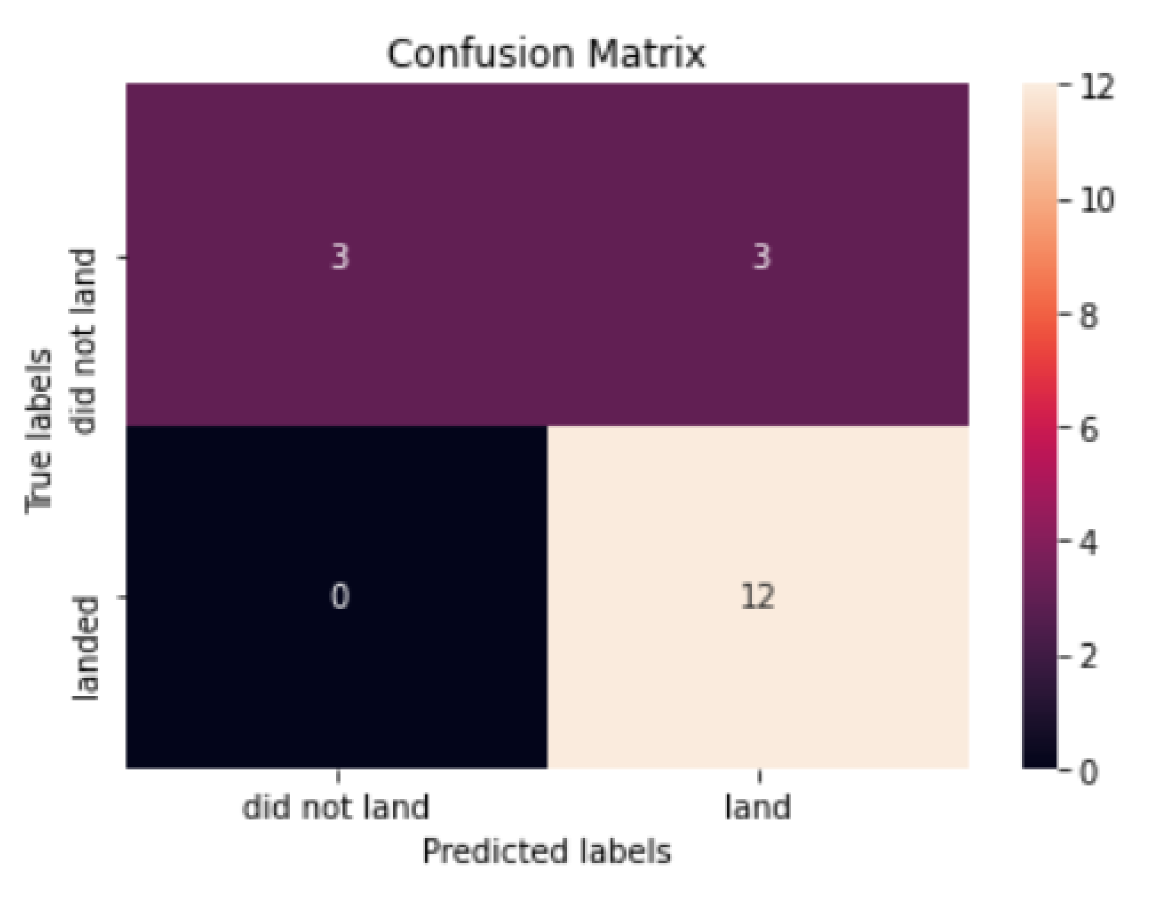
Classification Accuracy

Algorithm	Accuracy on Train Data	Accuracy on Test Data	Hyperparameters
Logistic Regression	0.8464285714285713	0.8333333333333334	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs’}
SVM	0.8482142857142856	0.8333333333333334	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid’}
Decision Trees	0.8892857142857142	0.8333333333333334	{'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random’}
KNN	0.8482142857142858	0.8333333333333334	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}

Our accuracies are extremely close, but we do have a winner, that is, the best model for this data is Decision Trees.

Confusion Matrix

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



Conclusions

- KSC LC-39A had the most successful launches of any sites.
- The larger the amount of flights at a launch site, the greater the success rate at that launch site.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rates.
- The rate of successful missions has been increasing from 2015.
- The best Machine Learning model for this task is the Decision Tree Classifier.

Thank you!