IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Tien Jie Jun (Jake)

28 May 2022

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data was collected from SpaceX API and SpaceX Wikipedia; a new column name label, Class, was created to classify successful landings; data was explored by using SQL, visualisations, folium maps and dashboards; gathered several relevant columns as features; modified all categorical variables to binary using one-hot encoding; normalised data and utilised GridSearchCV to find the best parameters for machine learning models; and visualised the accuracy score of all models.
- Summary of all results
  - All results in this project were produced with four machine learning models: Logistic Regression, Support Vector Machine, Decision Tree Classifier and K Nearest Neighbours. These results have the similar accuracy score on the test data: 83.33%. This depicts that all models have predicted more successful landings. Hence, more data is required for calculating the best accuracy and performance of the methods.

# Introduction

- Project background
  - Space X is one of the most successful companies that makes space travel affordable for the majority in this commercial space age. Space X had several achievements: sending spacecraft to the International Space Station; utilizing Starlink, a satellite internet constellation that provides the satellite internet access; and sending manned missions to Space. The main reason behind these achievements is due to inexpensive Falcon 9 rocket launches with $62 million, while other suppliers cost about $165 million. Thus, this can be determined whether the first stage will land and the cost of a launch. However, a new rocket company, Space Y, wants to compete with Space X.
- Problems you want to find answers
  - As a data scientist in Space Y, gathering information and creating dashboards for data analysis will be required as the first step. Then, a machine learning model will be trained to predict the first stage of successful recovery.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX API and SpaceX Wikipedia
- Perform data wrangling
  - Data was processed by classifying whether the booster has landed successfully or not.
- 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 collection involves the Space X Public API using GET requests, web scraping data from an HTML table on Space X's Wikipedia page, and basic data wrangling and formatting. The list will store the retrieved data and create into a new dataframe. The below depicts the column names of Space X API and Space X's Wikipedia Web Scraping respectively.

Space X API Column Names:

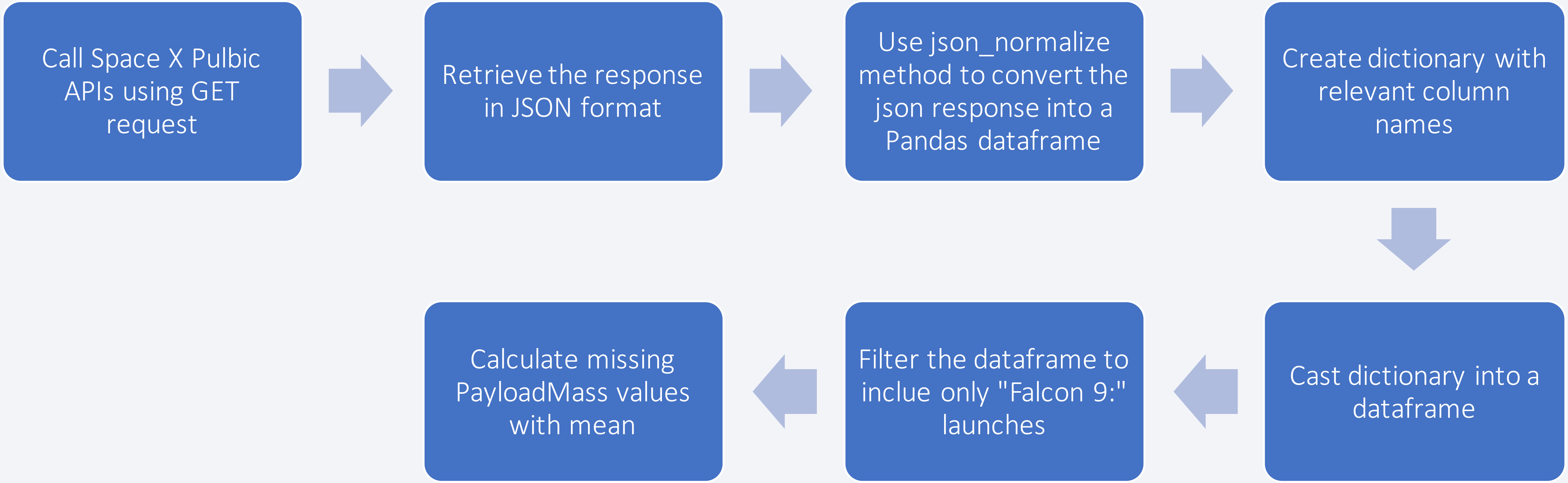
FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flight s, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Wikipedia Webscrape Column Names:

FlightNumber, Launch site, Payload, PayloadMass, Orbit, Customer, Launch out come, Version  Booster, Booster landing, Date, Time

The flowcharts will be presented in Slides 8 and 9.

# Data Collection – SpaceX API



Call Space X Pulbic APIs using GET request

Retrieve the response in JSON format

Use json_normalize method to convert the json response into a Pandas dataframe

Create dictionary with relevant column names

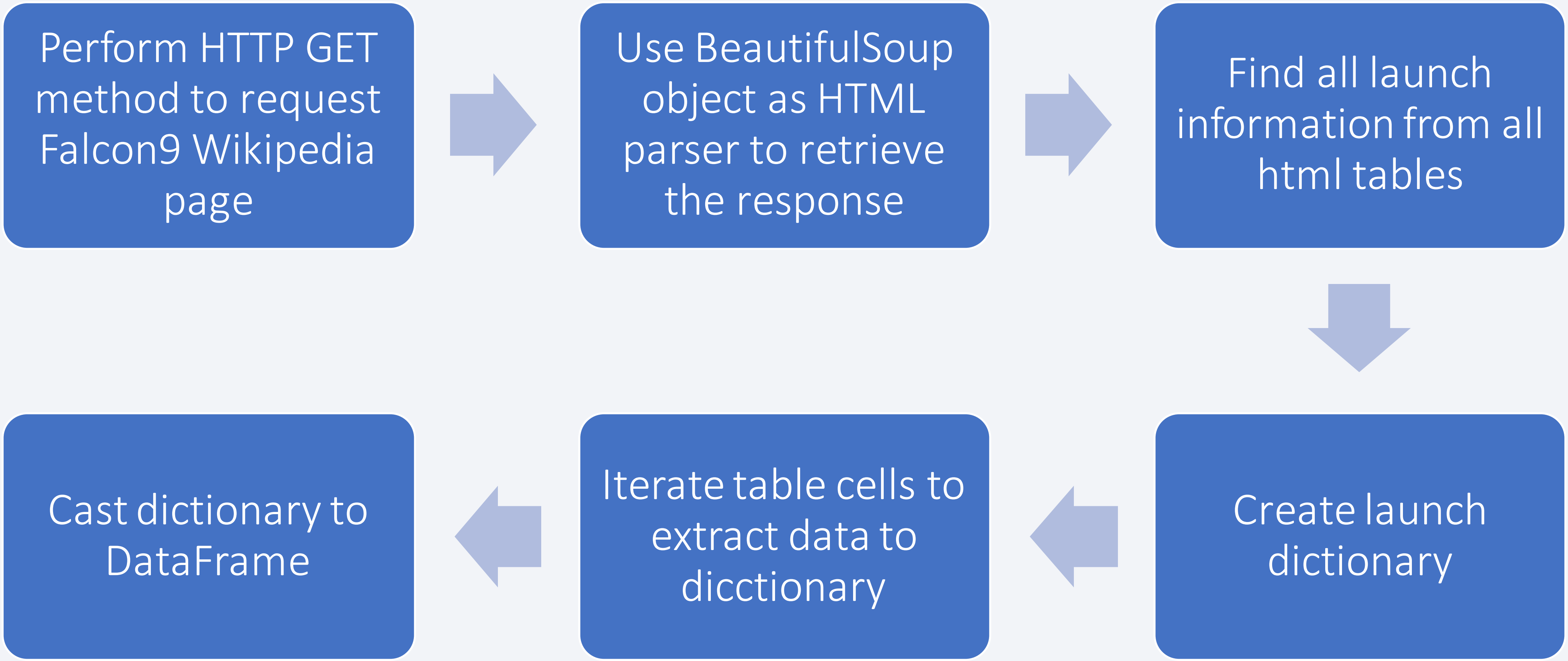
Cast dictionary into a dataframe

Filter the dataframe to inclue only "Falcon 9:" launches

Calculate missing PayloadMass values with mean

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%201/jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection - Scraping



Perform HTTP GET method to request Falcon9 Wikipedia page

Use BeautifulSoup object as HTML parser to retrieve the response

Find all launch information from all html tables

Create launch dictionary

Iterate table cells to extract data to dicctionary

Cast dictionary to DataFrame

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%201/jupyter-labs-webscraping.ipynb

# Data Wrangling

The landing outcomes convert into training labels with 1 – booster landed successfully, and 0 – booster landed unsuccessfully. Both landing outcome and mission outcome labels will be created from the Outcome column. The new training label column, Class, is the classification variable that represents the outcome of each launch. The first stage did not land successfully if the value is 0. Otherwise, the first stage landed successfully if the value is 1.The table below depicts the mapping of the outcome.

| Value | Outcome |
|---|---|
| 0 | False Ocean, False RTLS, None ASDS. None None |
| 1 | True Ocean, True RTLS, True ASDS |

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%201/labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with Data Visualization

Both Pandas and Matplotlib perform Exploratory Data Analysis (EDA) and Feature Engineering. The following variables are used to perform EDA: Flight Number, Payload Mass, Launch Site, Orbit, Class and Year. With that, the following relationships were visualised on its detailed launch records.

| Plots | Relationships |
|---|---|
| Catplot | Flight Number vs Launch Site; Payload vs Launch Site; Flgiht Number vs Orbit Type; Payload vs Orbit Type |
| Line chart | Success yearly trend |
| Bar chart | Orbit vs Success rate of each orbit type |

The above table depicts the plots use on these relationships. Should these relationships exist, they could be used for training machine learning model.

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%202/EDA%20with%20Visualization.ipynb

# EDA with SQL

- SQL queries performed
  - Load dataset into IBM DB2 Database
  - Used SQL Python Integration for SQL queries. The database connection must be established with our own Service Credentials beforehand.
  - Learnt basic SQL commands such as 'SELECT', 'FROM', 'WHERE', 'ORDER BY' and more.
  - Retrieved information such as launch site names, mission outcomes, payload mass carried by booster version F9 v1.1, landing outcomes and the names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%202/Wk2%20sql.ipynb

# Build an Interactive Map with Folium

Folium maps are utilised to label launch sites, successful and unsuccessful landings, and a proximity example to key locations such as Railway, Highway, Coast, and City. Hence, this enables us to comprehend the approximate locations of the launch sites and visualise all successful landings that are relative to their location.

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%203/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

# Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard contains both pie and scatter charts for all launch sites. The infographic will be displayed in these charts depending on all launch sites or any individual launch sites. The pie chart displays the distribution of successful landings across these sites' success rates with its visualisation while the scatter plot displays the success that varies across launch sites, payload mass and booster version category with these two inputs: All sites or any individual site; and payload mass on the HTML slider between 0 and 10000kg.

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%203/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

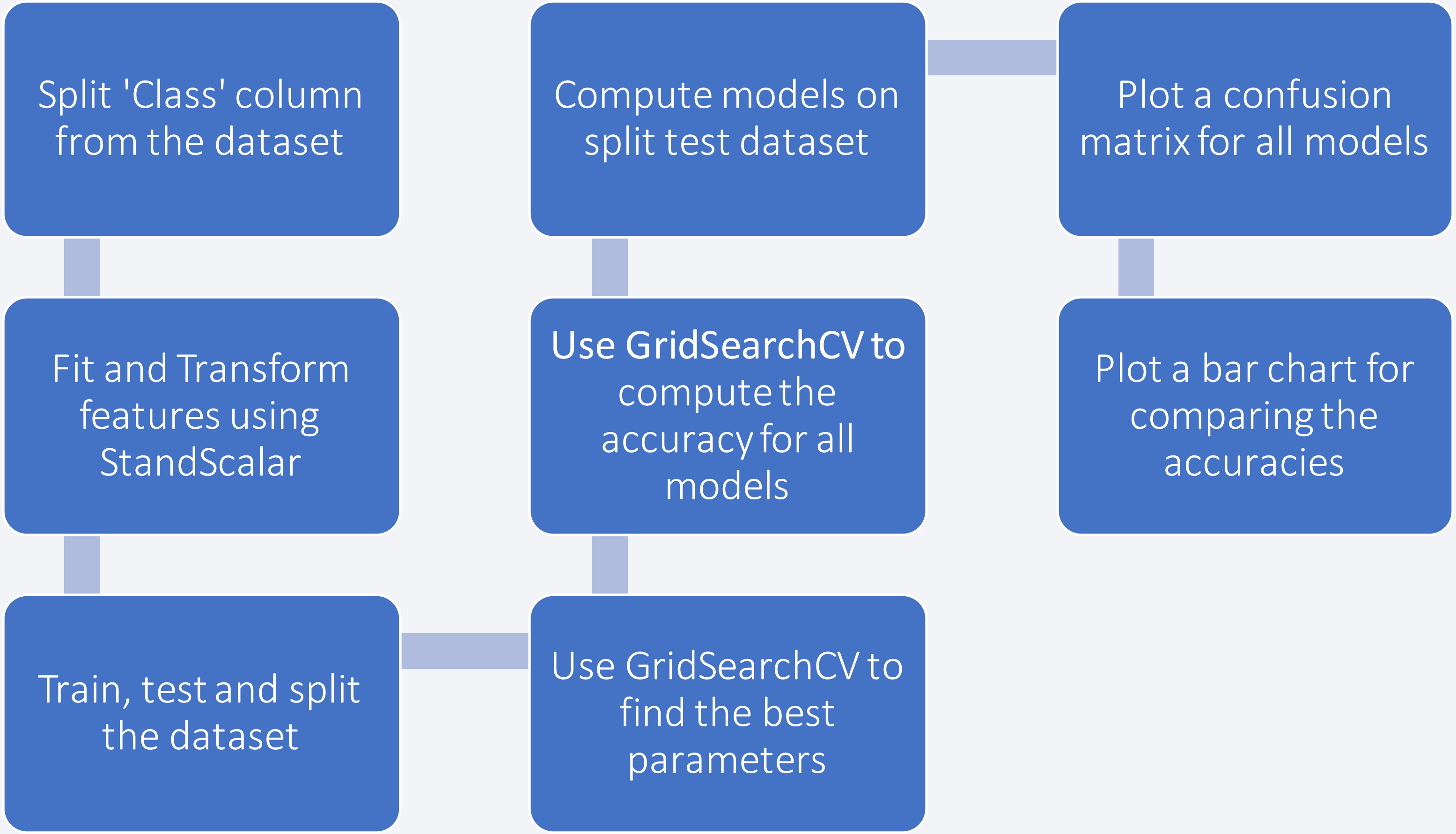
# Predictive Analysis (Classification)

First, the 'Class' column will be split from the dataframe to a NumPy array as variable Y using the method to_numpy(). Next, StandardScaler() will be used for fitting and transforming the features from the dataframe as variable X. Once the fit and transform process completes, tran_test_split function starts to split the data X and Y into training and test datasets. Thus, the number of samples will be available in both training and test datasets. Furthermore, GridSearchCV object looks for the best parameters and best accuracy. This method applies to Logistic Regression, support vector machine (SVM), decision tree classifier and K-Nearest Neighbour models. Once computing the accuracy scores of all models completes, the test dataset is used for plotting a confusion matrix. Finally, these accuracy scores will be analysed to see which method has the same performance.

The flow chart is presented in slide 16.

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%204/Machine%20Learning%20Prediction.ipynb

# Predictive Analysis (Classification)



Split 'Class' column from the dataset

Fit and Transform features using StandScalar

Train, test and split the dataset

Use GridSearchCV to find the best parameters

**Use GridSearchCV to** compute the accuracy for all models

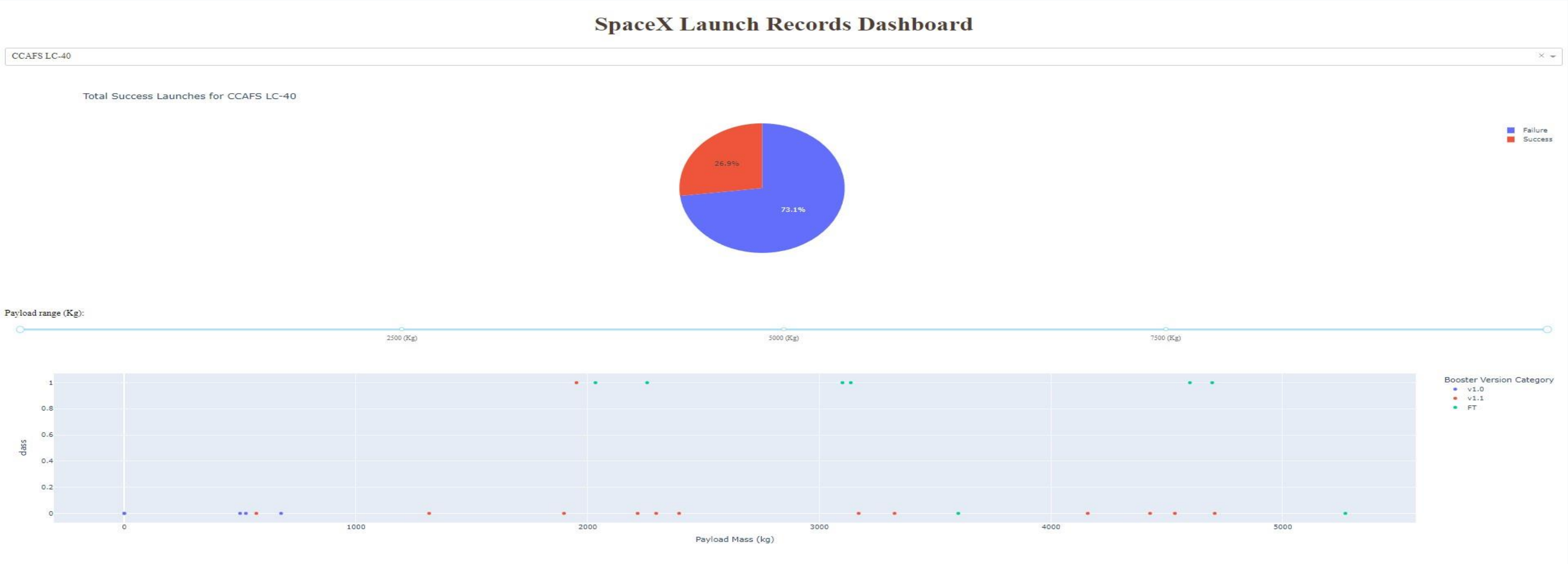
Compute models on split test dataset

Plot a confusion matrix for all models

Plot a bar chart for comparing the accuracies

Gitlab URL: https://github.com/jaketee93/capstone_project/blob/develop/wk%204/Machine%20Learning%20Prediction.ipynb

# Results



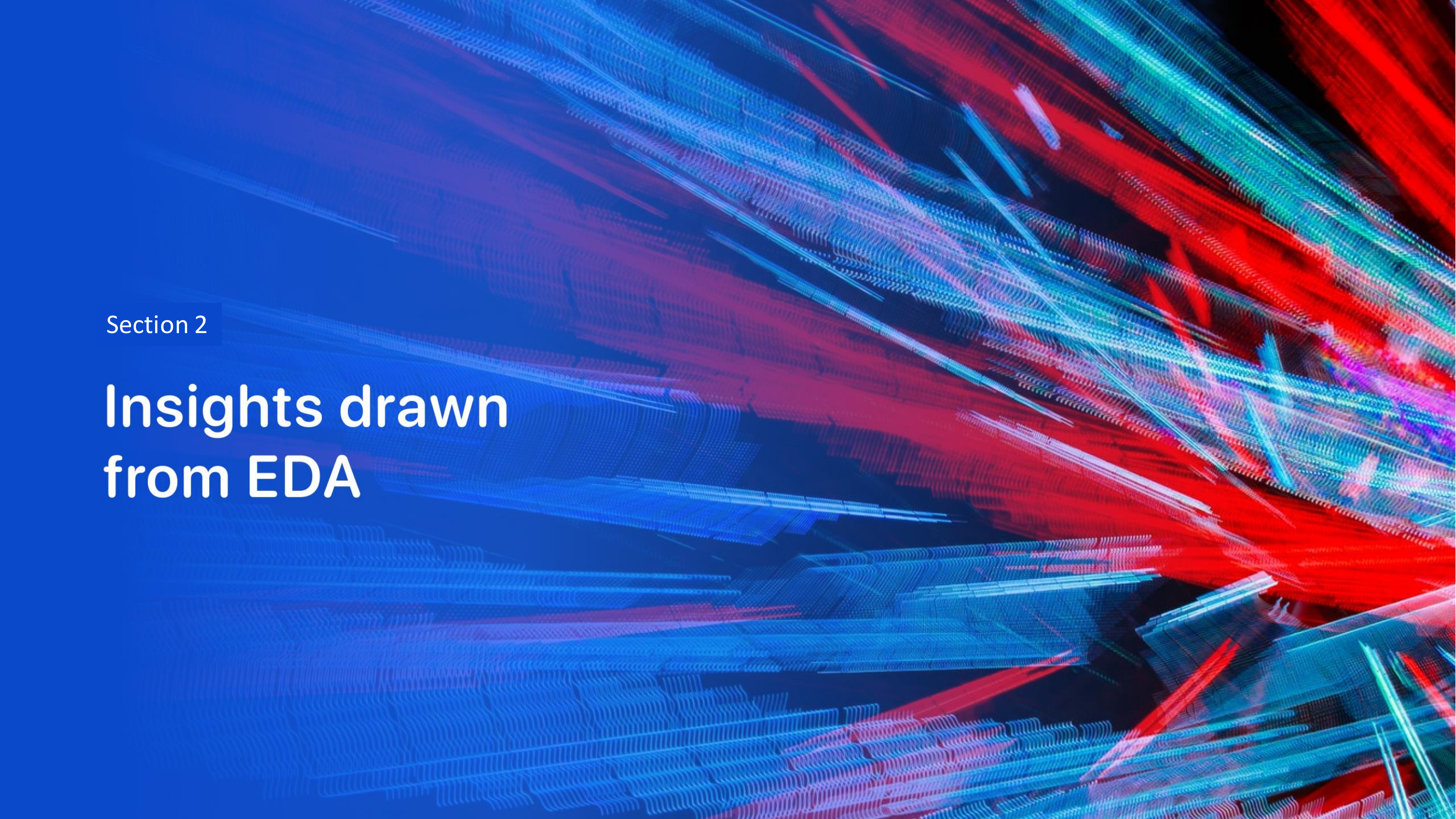
SpaceX Launch Records Dashboard

CCAFS LC-40

Total Success Launches for CCAFS LC-40

Payload range (Kg):

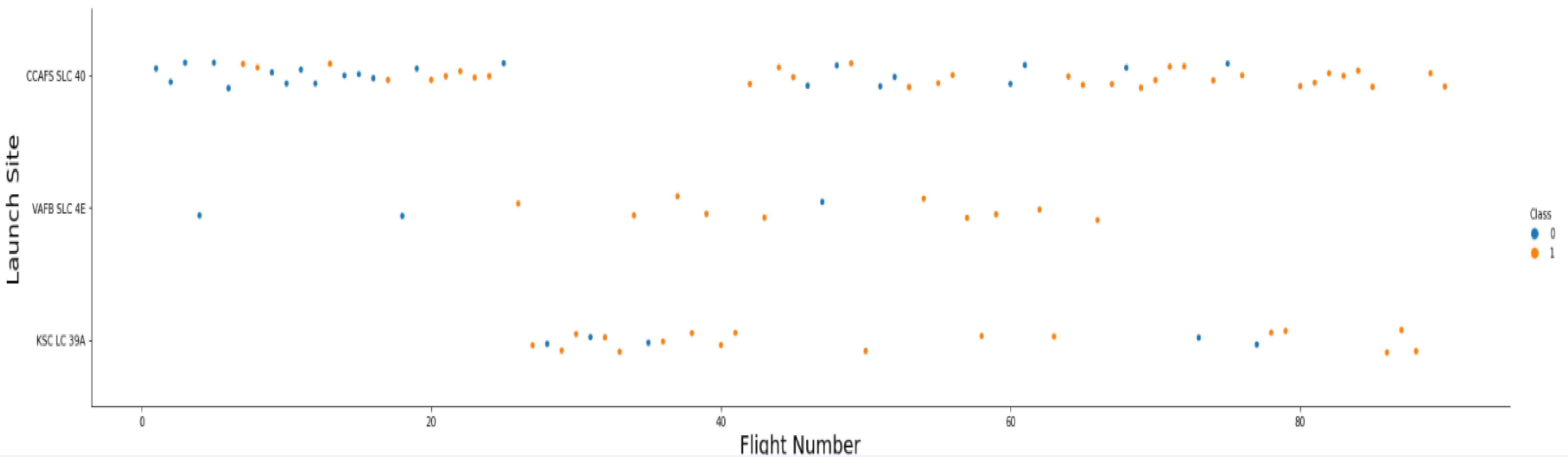
The above diagram depicts the Plotly dashboard. The results of EDA with Visualisation, EDA with SQL, Interactive Map with Folium and accuracy score of all models are displayed in the next section.



Section 2

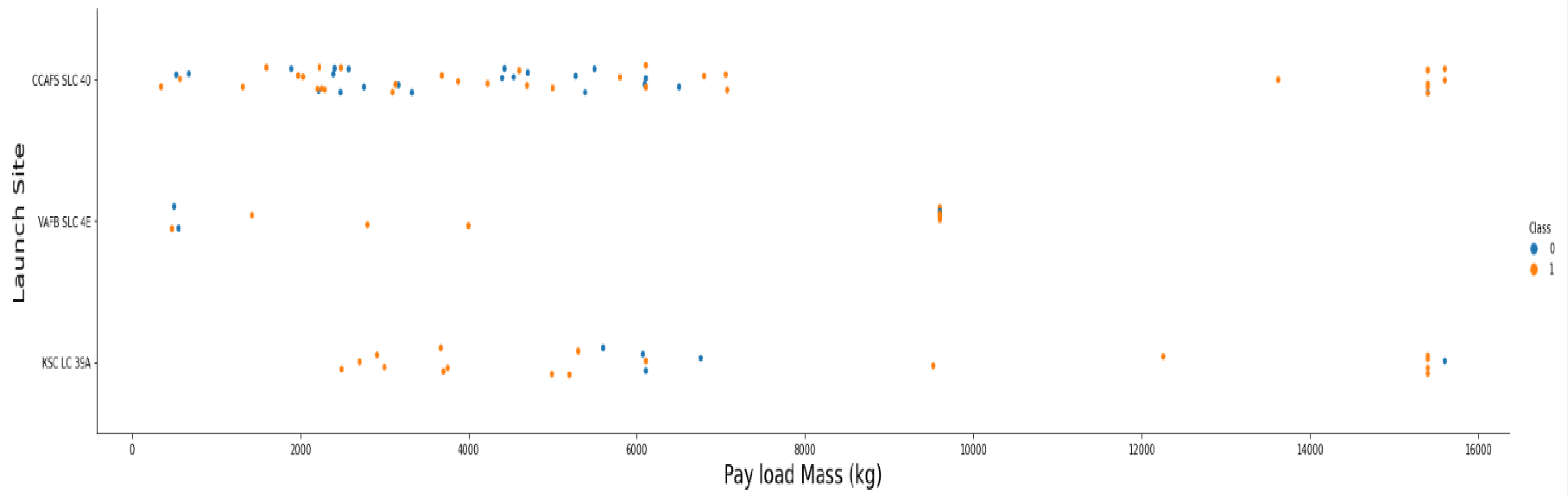
# Insights drawn from EDA

# Flight Number vs. Launch Site



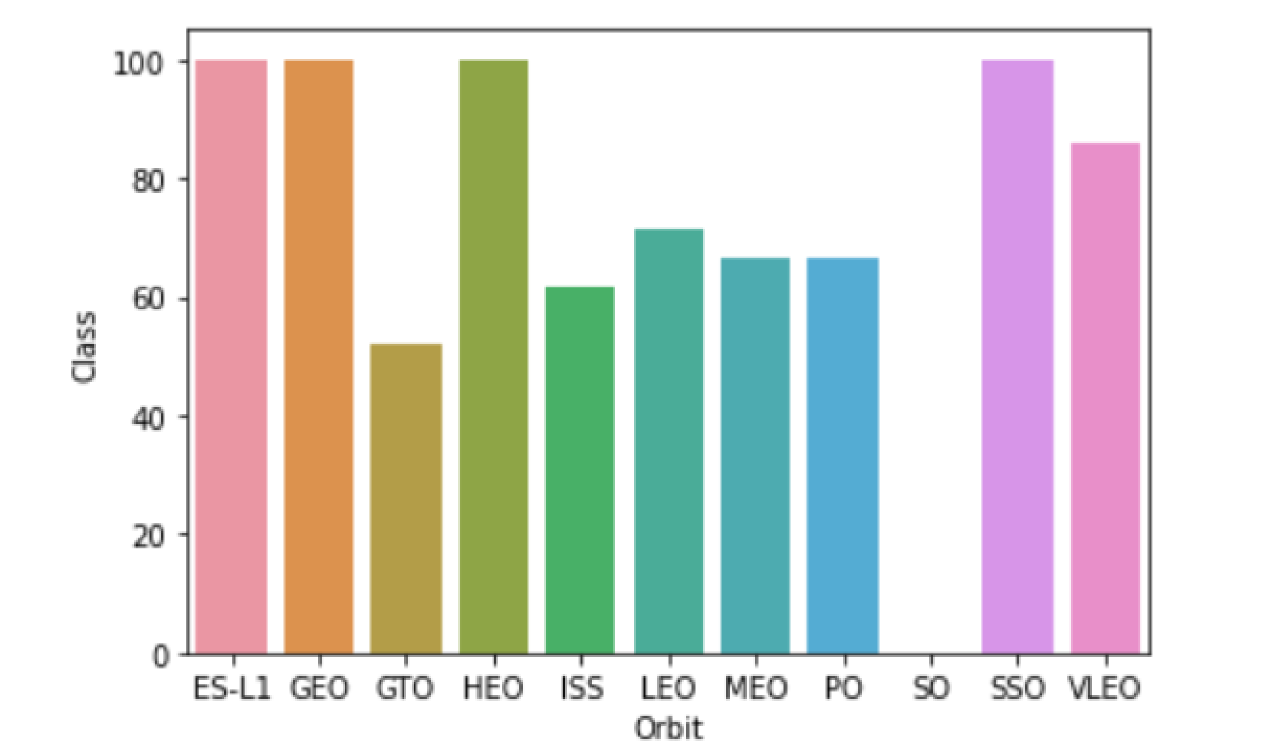
In the above infographic, orange indicates a successful launch, and blue indicates an unsuccessful launch. This infographic depicts that the success rate has increased significantly after the 20th flight. CCAFS SLC 40 is most likely the main launch due to the higher number of flights.

# Payload vs. Launch Site



In the above infographic, orange indicates a successful launch, and blue indicates an unsuccessful launch. This infographic depicts that the payload is not massive as the first stage returns between 0 and 6000kg. Different launch sites are using different payload mass.

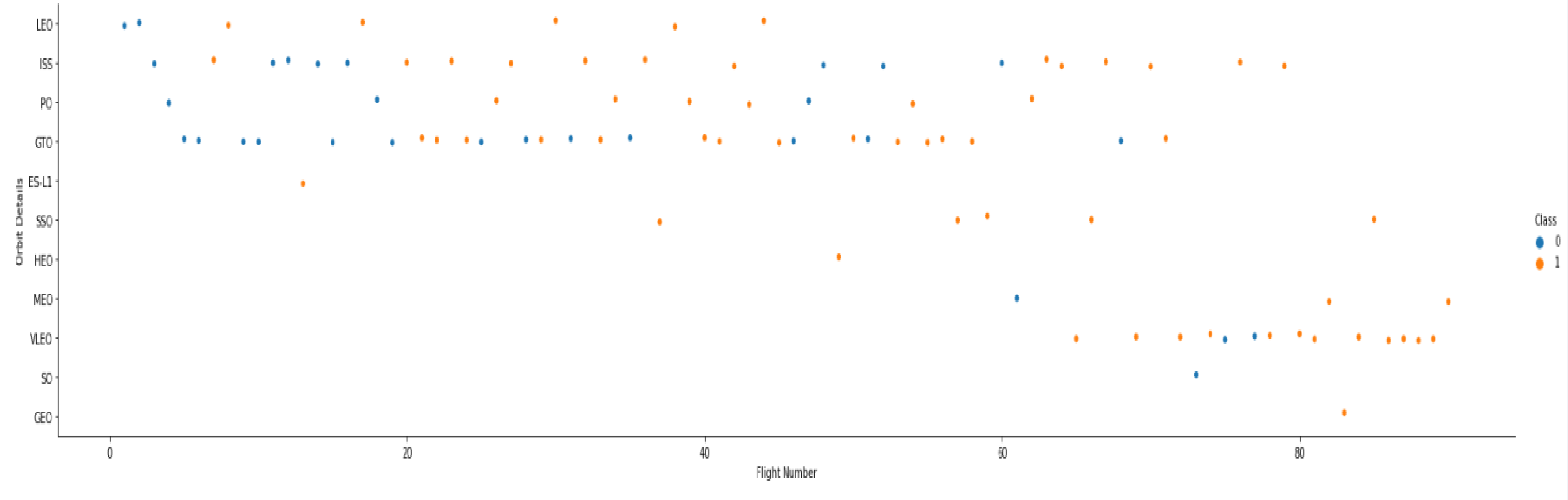
# Success Rate vs. Orbit Type



The success rate is labelled as Class.

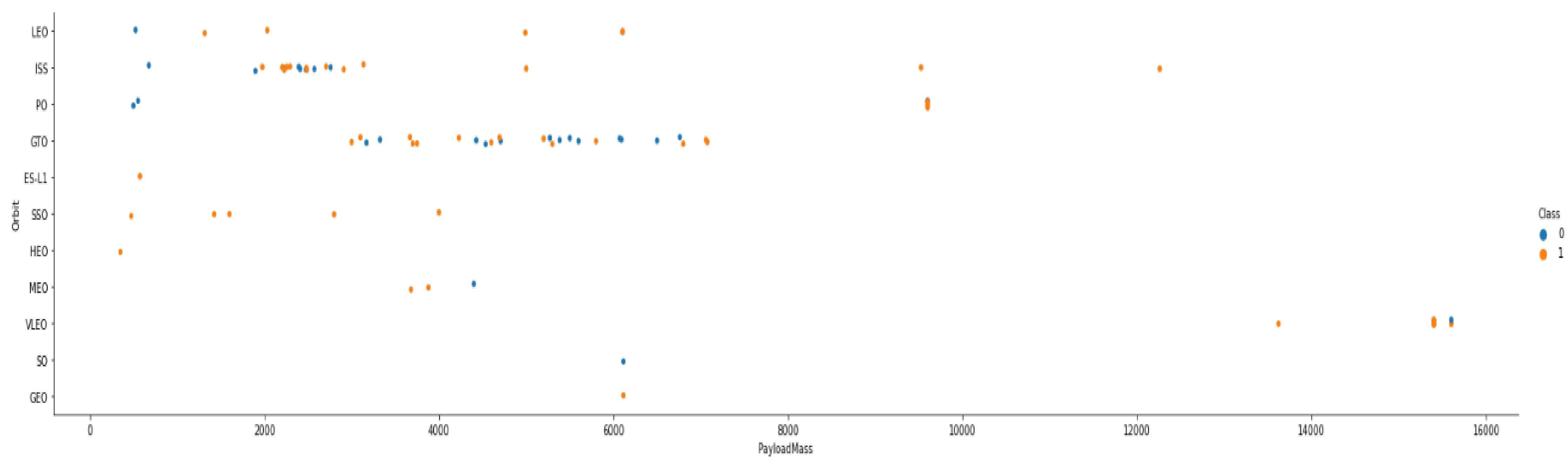
The above chart depicts that ES-L1, GEO, HEO and SSO have the highest success rate which is 100%; VLEO has 80% success rate; and SO has 0% success rate.

# Flight Number vs. Orbit Type



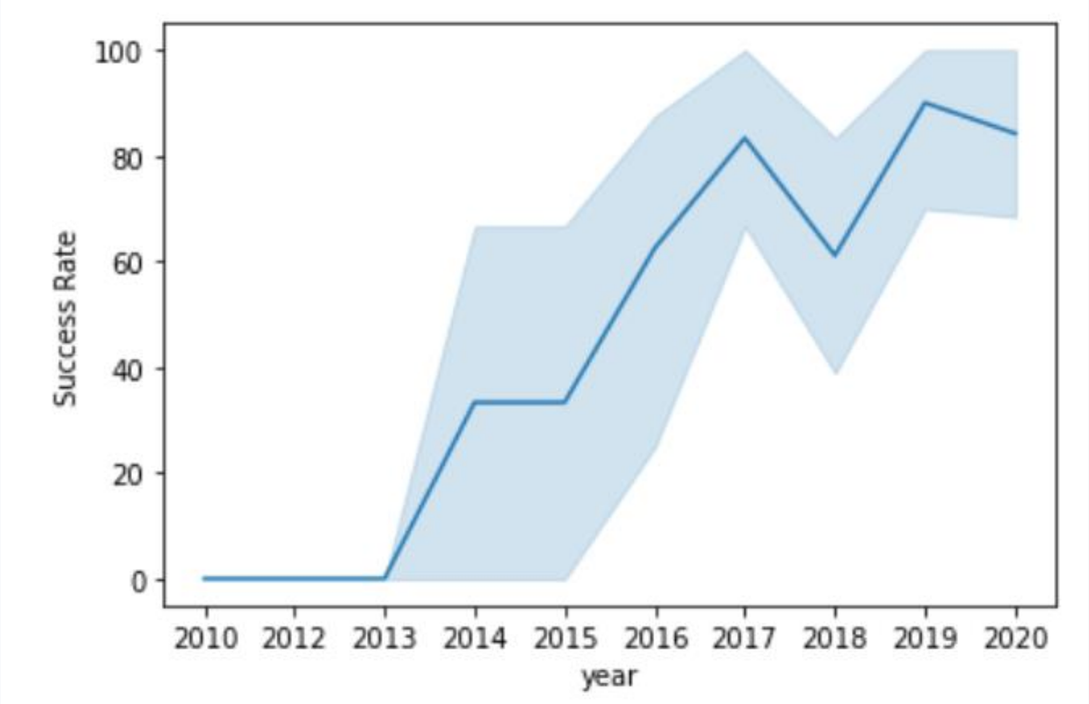
In the above infographic, orange indicates a successful launch, and blue indicates an unsuccessful launch. This infographic depicts that the launches of the orbit types change over flight numbers. Launch outcome is correlated with this preference. Therefore, SpaceX started launching LEO orbits with moderate success and then returned to VLEO in the recent launches. Thus, this shows that SpaceX can perform better in lower orbits.

# Payload vs. Orbit Type



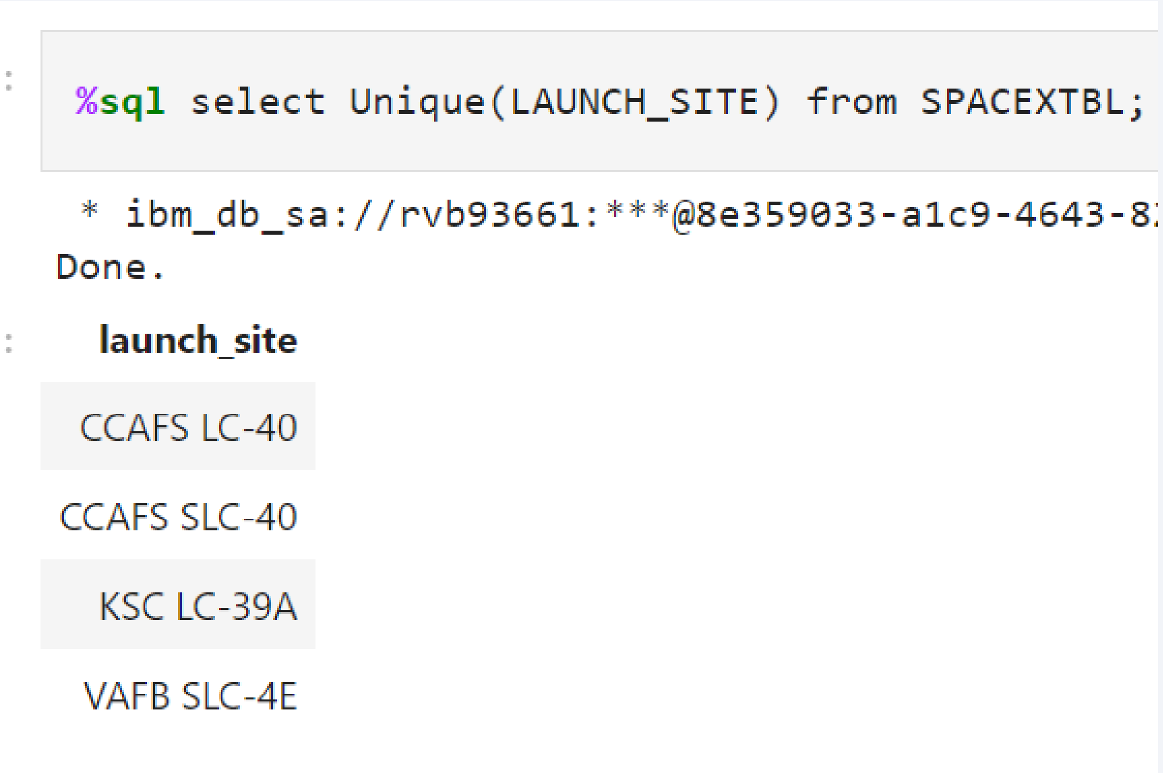
In the above infographic, orange indicates a successful launch, and blue indicates an unsuccessful launch. This infographic depicts that Orbits LEO, ES-L1, SSO and HEO have the lowest Payload Mass with below 2000kg. On the other hand, VELO has the highest Payload Mass nearer to 16000kg.

# Launch Success Yearly Trend



The above line chart displays that the success rate has increased rapidly to 80% from 2010 to 2017. Despite the slight decline to 60% success rate in 2018, it has surged to 80% success rate approximately in 2020. Light blue shaded area indicates that 95% confidence interval is used in this line chart.

# All Launch Site Names



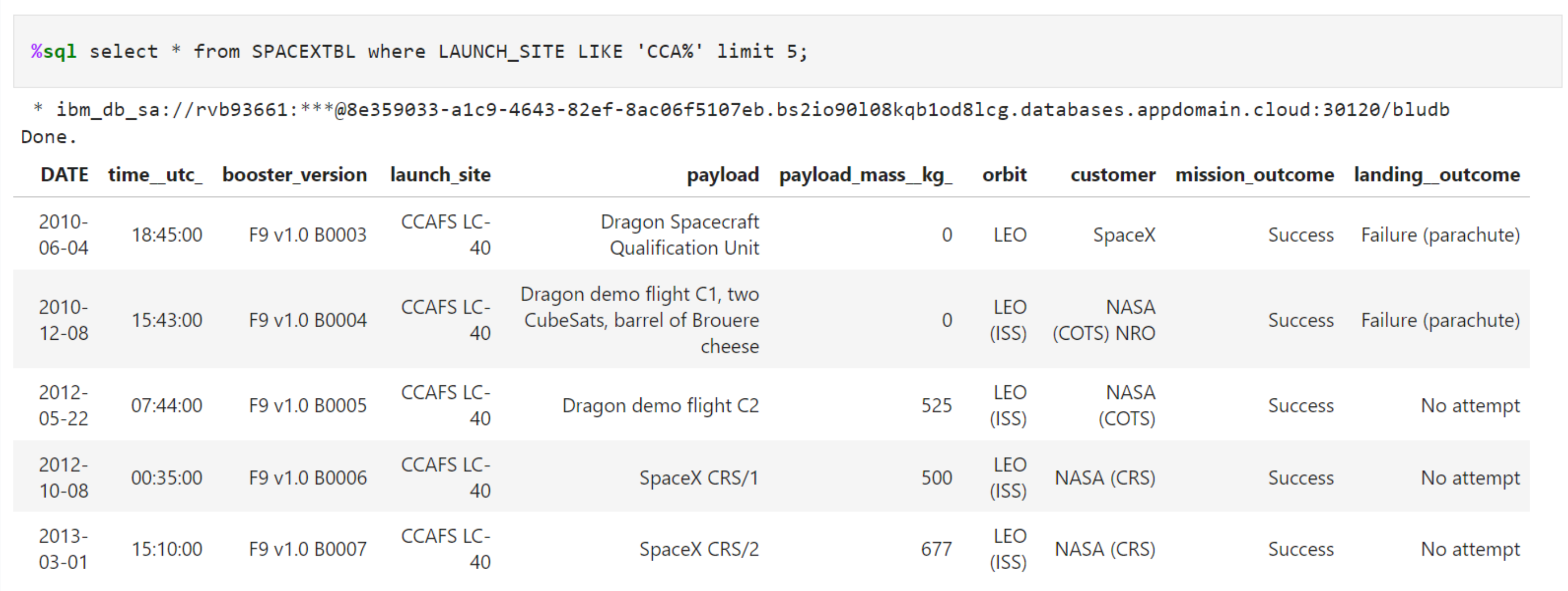
```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-8
Done.
```

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

The query in the above image retrieves all unique names of the launch sites. All launch site names: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E.

# Launch Site Names Begin with 'CCA'



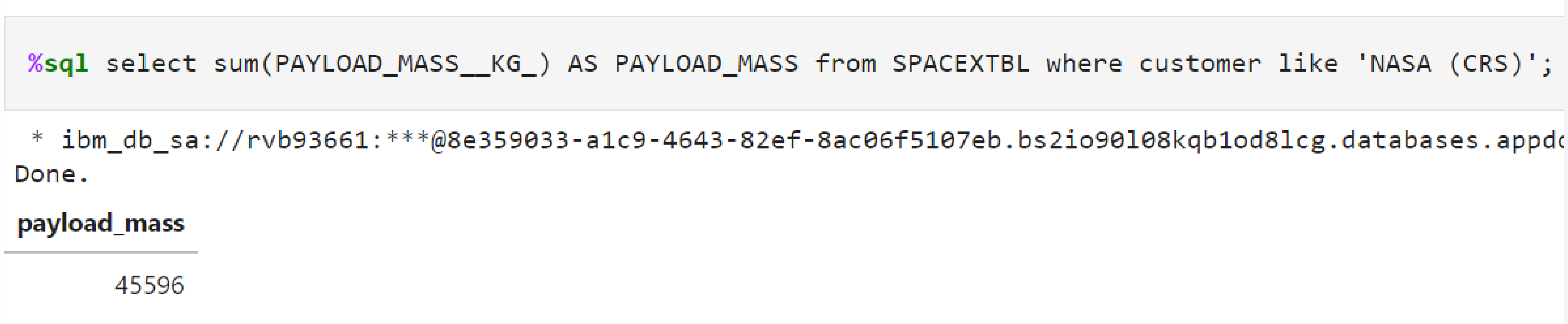
```
%sql select * from SPACEXTBL where LAUNCH_SITE LIKE 'CCA%' limit 5;
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

| 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 |

The query in the above image retrieves the first five entries with the launch site names that start with 'CCA' in the database.

# Total Payload Mass



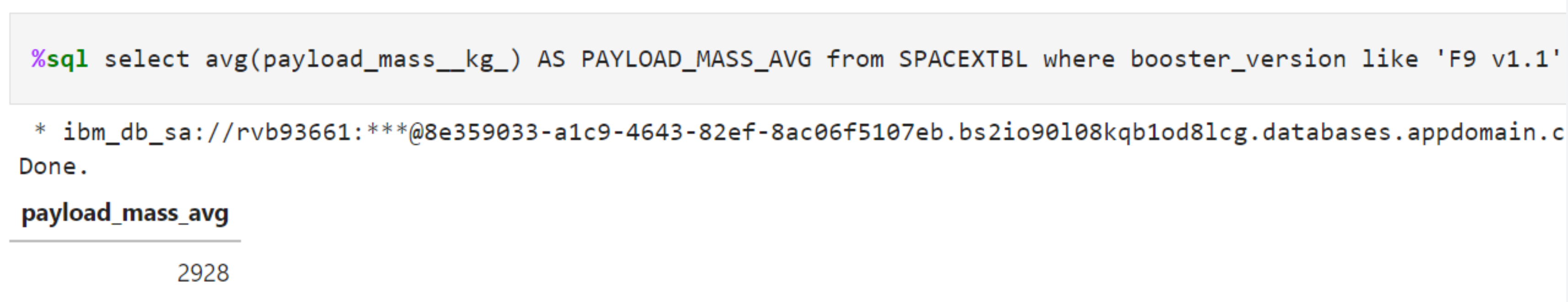
```
%sql select sum(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS from SPACEXTBL where customer like 'NASA (CRS)';
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appd
Done.
```

| payload_mass |
|---|
| 45596 |

The query in the above image calculates the total payload mass carried by boosters launched by NASA (CRS).

# Average Payload Mass by F9 v1.1



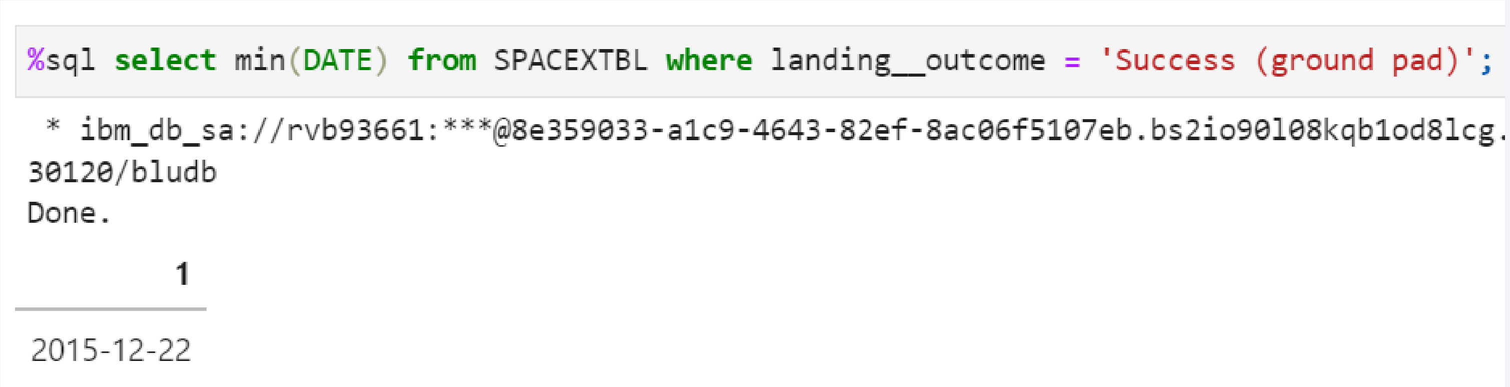
```
%sql select avg(payload_mass__kg_) AS PAYLOAD_MASS_AVG from SPACEXTBL where booster_version like 'F9 v1.1'
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.c
Done.
```

| payload_mass_avg |
|---:|
| 2928 |

The query in the above image calculates the average payload mass carried by boosters version F9 v1.1.

# First Successful Ground Landing Date



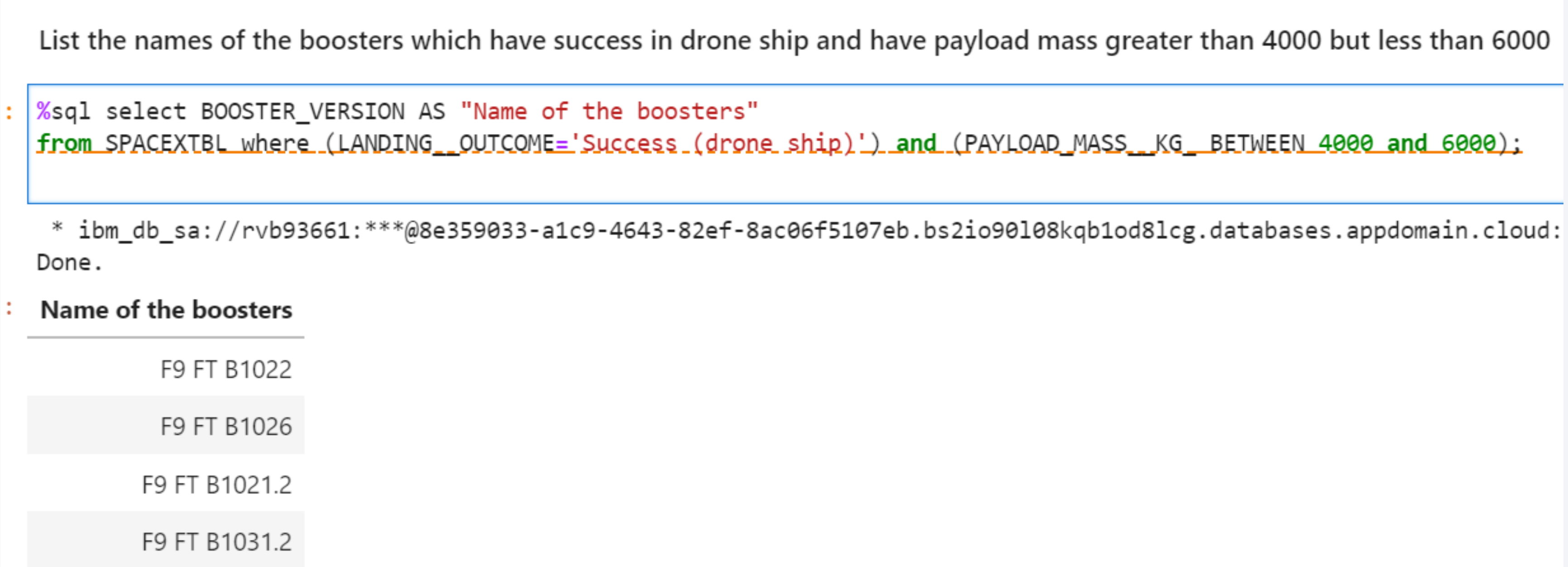
```
%sql select min(DATE) from SPACEXTBL where landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.
30120/bludb
Done.
```

| 1 |
|---|
| 2015-12-22 |

The query in the above image queries the first successful ground landing date on 22 December 2015.

# Successful Drone Ship Landing with Payload between 4000 and 6000



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

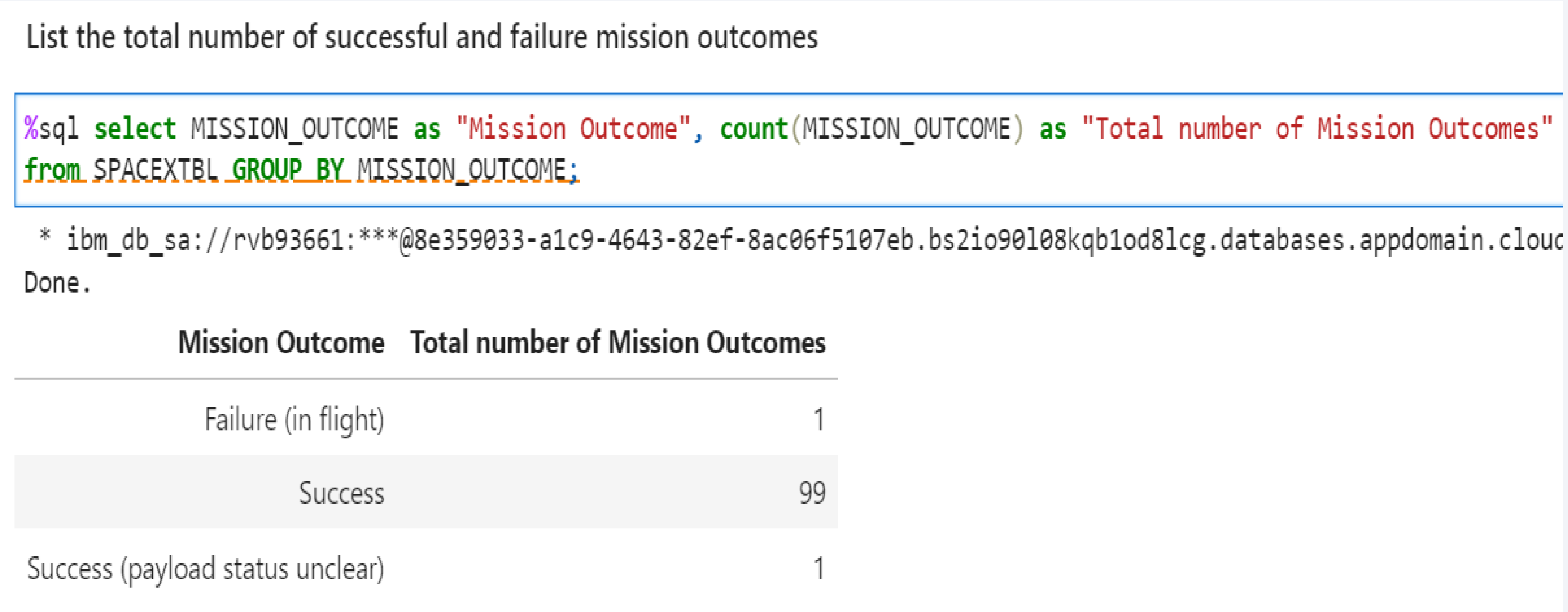
```
%sql select BOOSTER_VERSION AS "Name of the boosters"
from SPACEXTBL where (LANDING__OUTCOME='Success (drone ship)') and (PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000);
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:
Done.
```

| Name of the boosters |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

The query in the above image queries the list of boosters names that have success in drone ship and payload mass between 4000 to 6000kg.

# Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

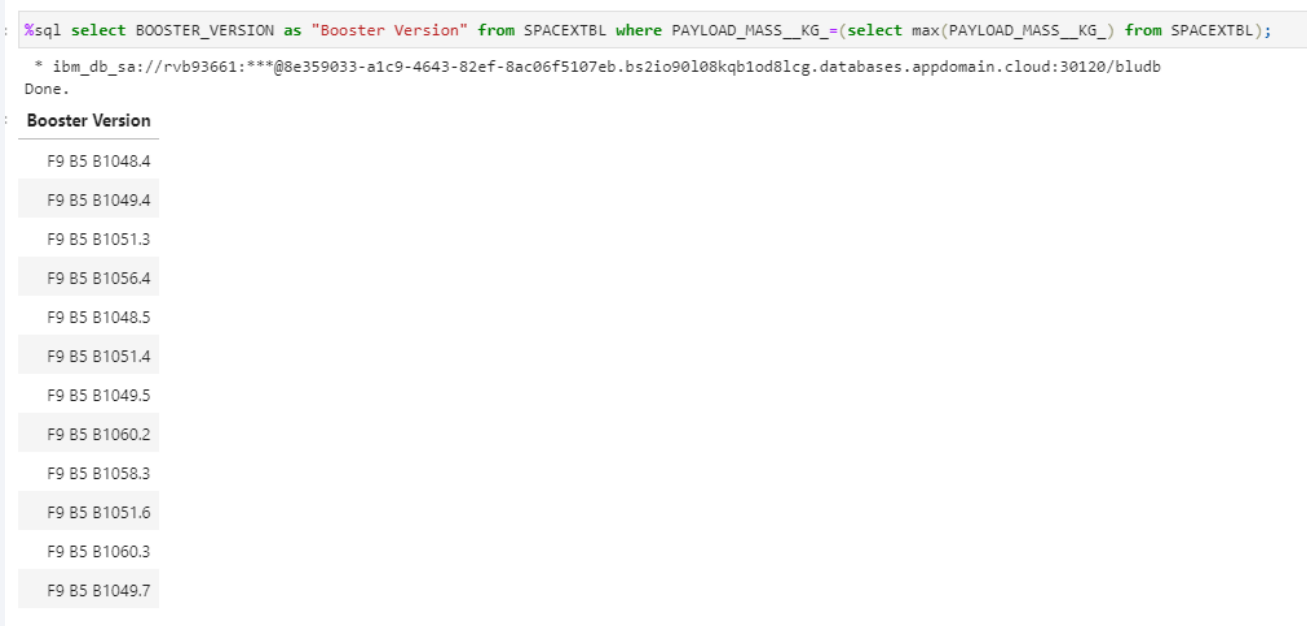
```
%sql select MISSION_OUTCOME as "Mission Outcome", count(MISSION_OUTCOME) as "Total number of Mission Outcomes"
from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.clouc
Done.
```

| Mission Outcome | Total number of Mission Outcomes |
|---:|---:|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

The query in the above image queries the <u>total number of successful and failure mission outcomes.</u>

# Boosters Carried Maximum Payload



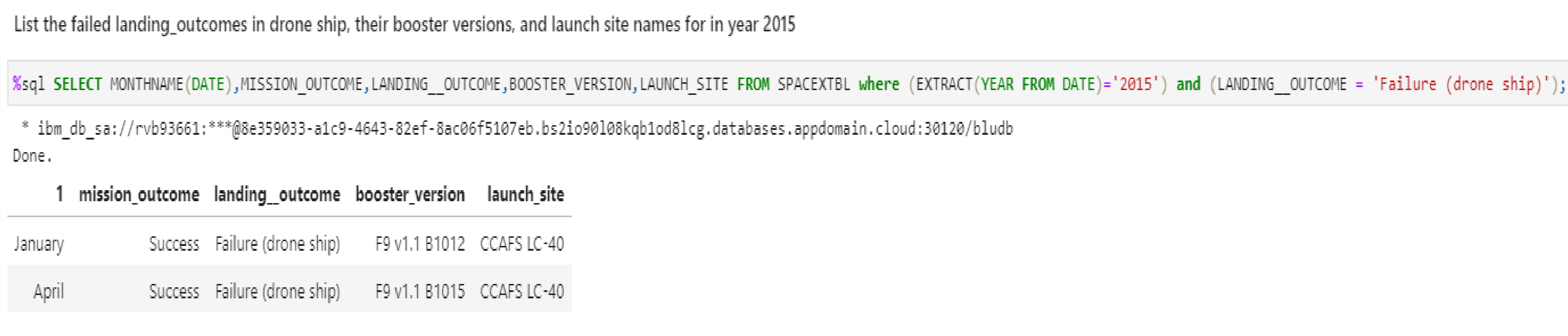
```
%sql select BOOSTER_VERSION as "Booster Version" from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

| 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 |

The query in the above image queries the names of the booster version that carry the maximum payload mass.

# 2015 Launch Records



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

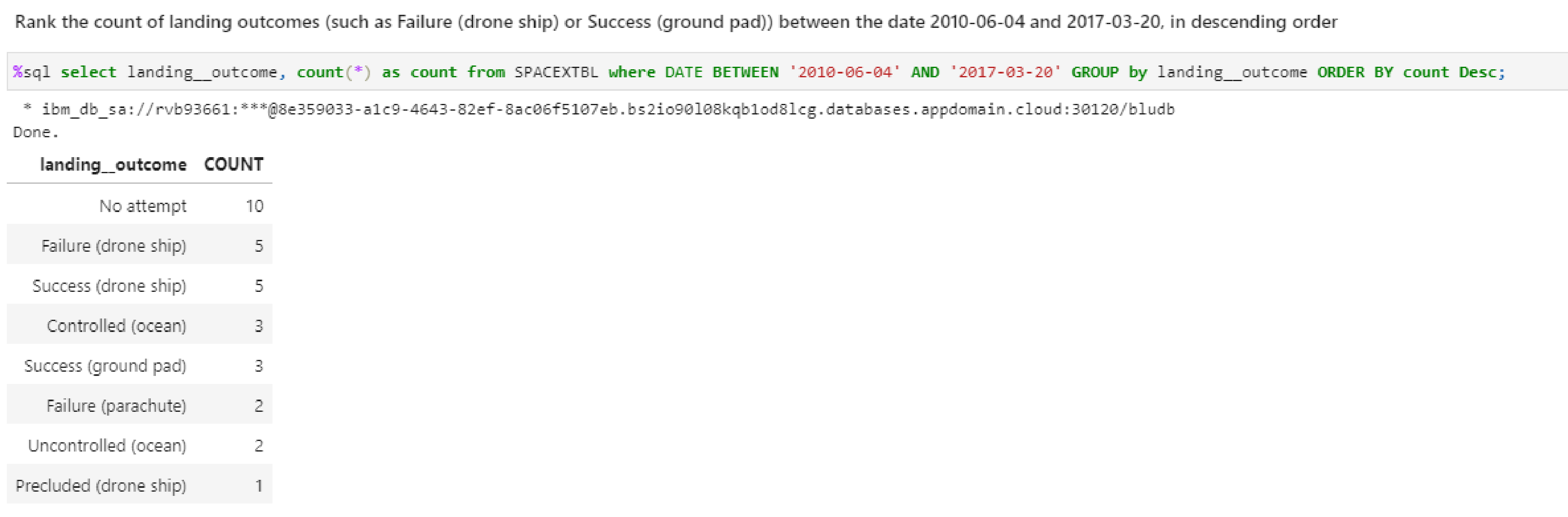
```
%sql SELECT MONTHNAME(DATE),MISSION_OUTCOME,LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where (EXTRACT(YEAR FROM DATE)='2015') and (LANDING__OUTCOME = 'Failure (drone ship)');
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

| 1 | mission_outcome | landing__outcome | booster_version | launch_site |
|---|---|---|---|---|
| January | Success | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| April | Success | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

The query in the above image queries the names of the booster version that carry the maximum payload mass.

# 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

```
%sql select landing__outcome, count(*) as count from SPACEXTBL where DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP by landing__outcome ORDER BY count Desc;
```

```
 * ibm_db_sa://rvb93661:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

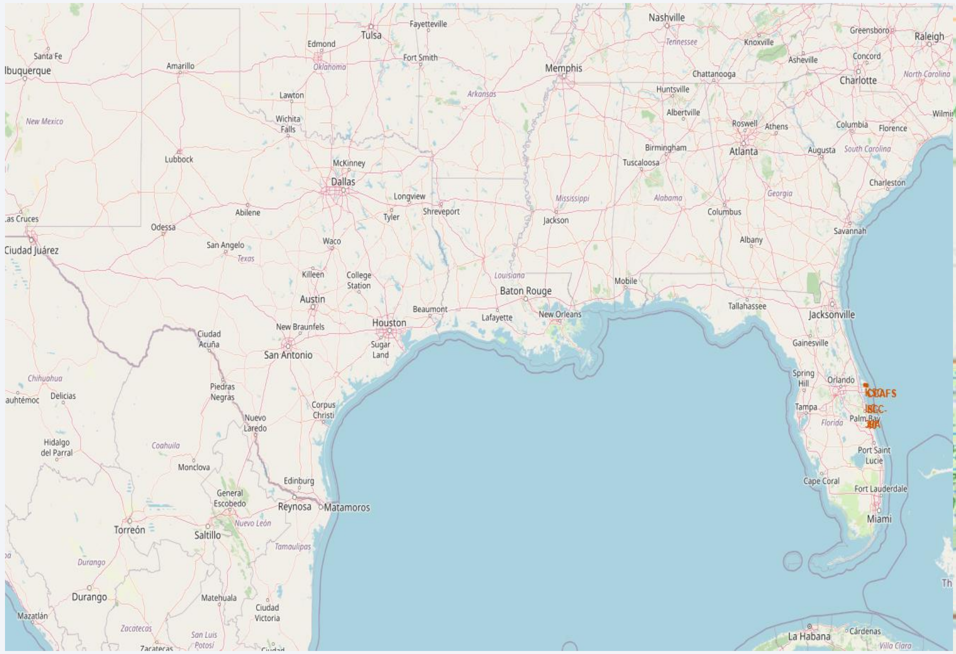
| landing__outcome | COUNT |
|---|---|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

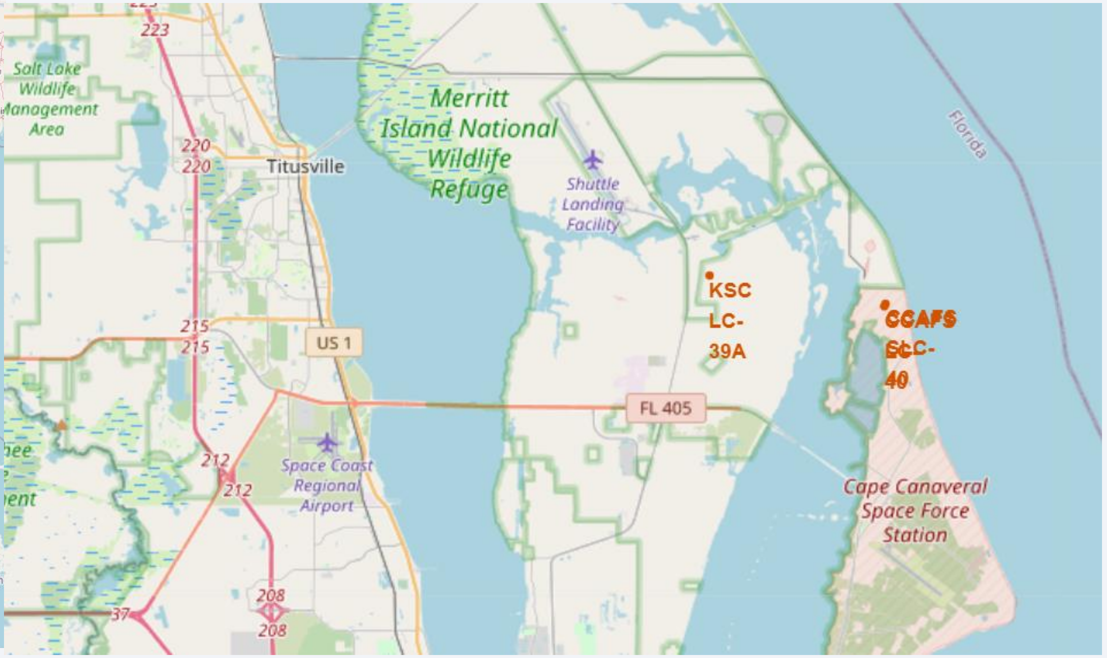
The query in the above image queries the number of landing outcome, such as Failure (drone ship), between 4 June 2010 and 20 March 2017 in descending order.

Section 3

# Launch Sites Proximities Analysis

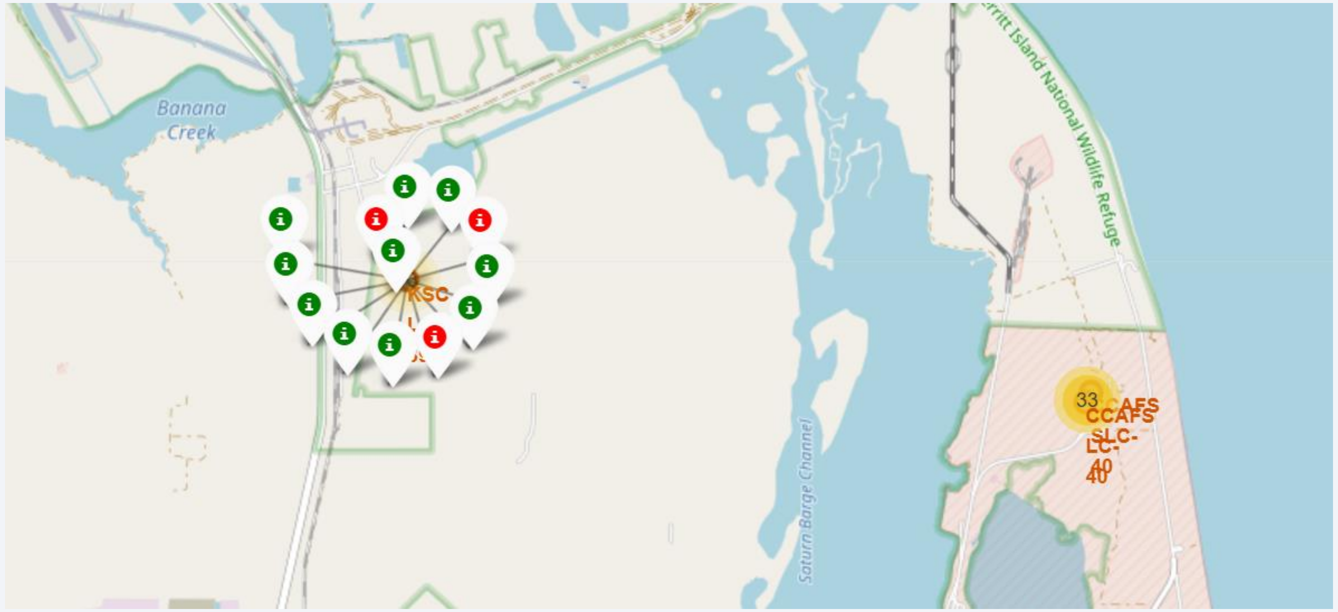
# Folium Map Screenshot 1: Launch Site Locations




Salt Lake Wildlife Management Area
Merritt Island National Wildlife Refuge
Titusville
Shuttle Landing Facility
KSC
LC-
39A
CCAFS
SLC-
40
US 1
FL 405
Space Coast Regional Airport
Cape Canaveral Space Force Station
Florida
223
220
215
212
208
37

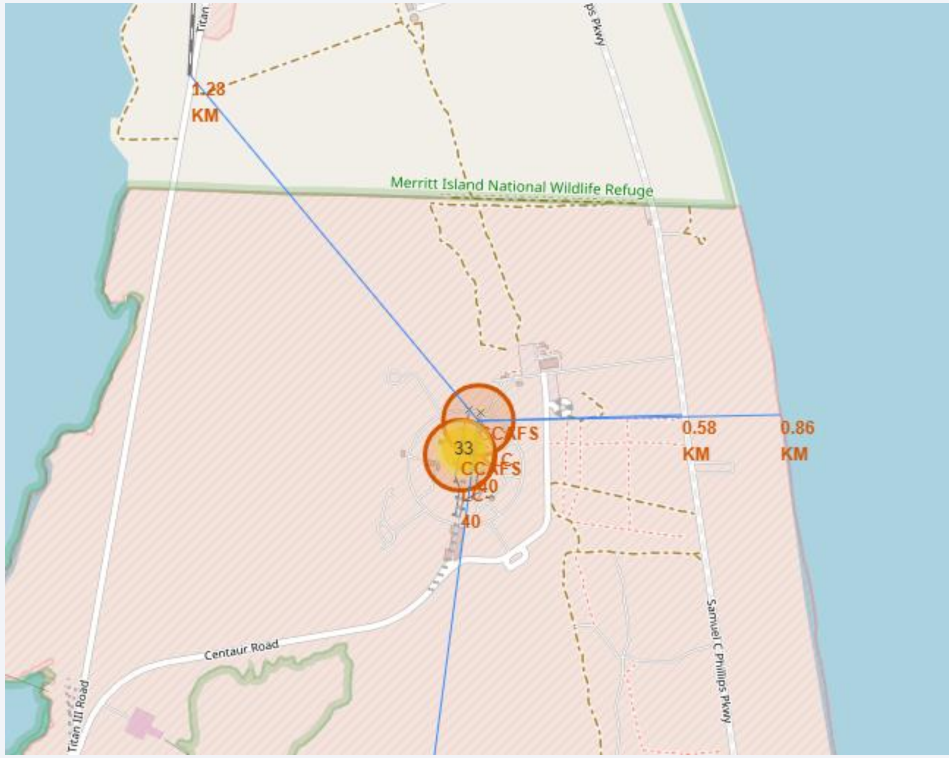
The map on the left shows all launch sites in US, while the other map shows all launch sites in Florida are nearer to one another. These launch sites are nearer to the ocean

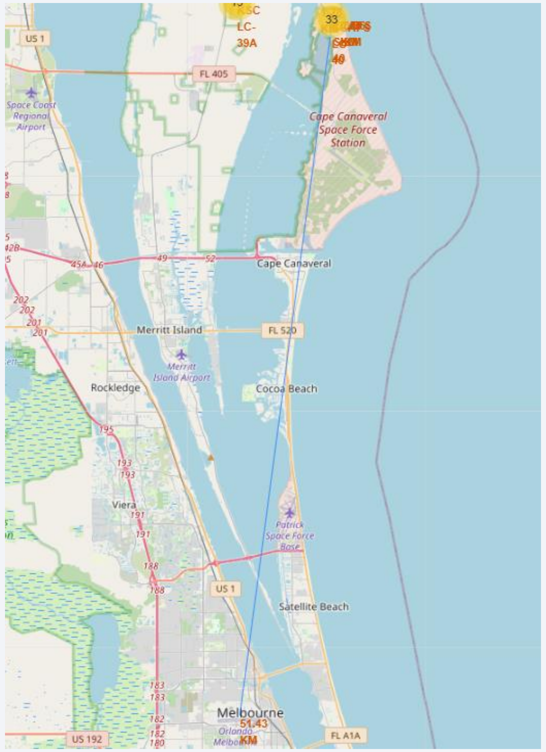
# Folium Map Screenshot 2: Launch Site Markers with Colour Code



Each cluster shows the total number of launches from their launch sites. If one cluster is selected in the map, it will display each successful landing location (green) and failed landing location (red). For example, KSC LC-39A has 10 successful landings and 3 failed landings.

# Folium Map Screenshot 3: Key Location Proximities

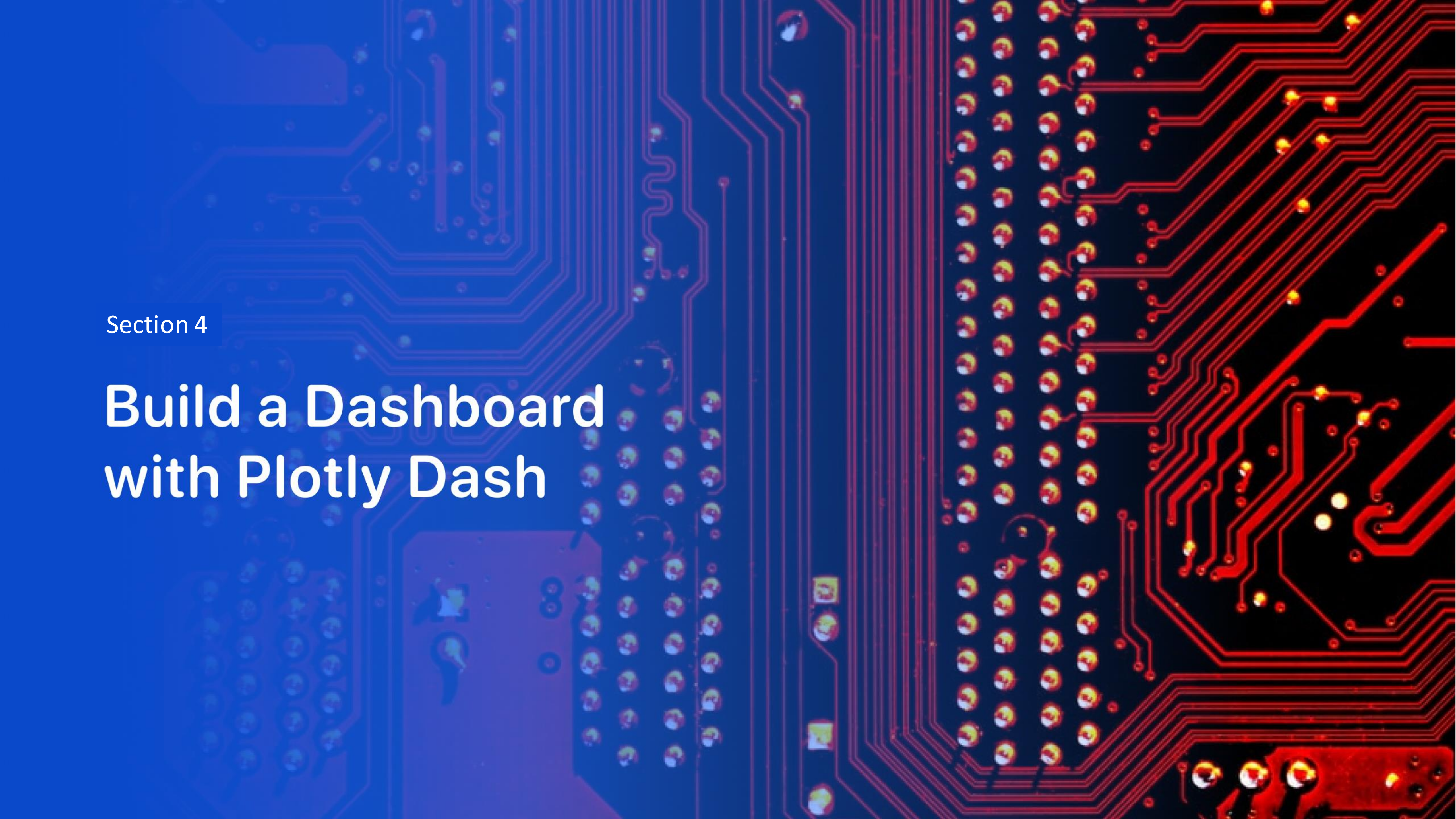




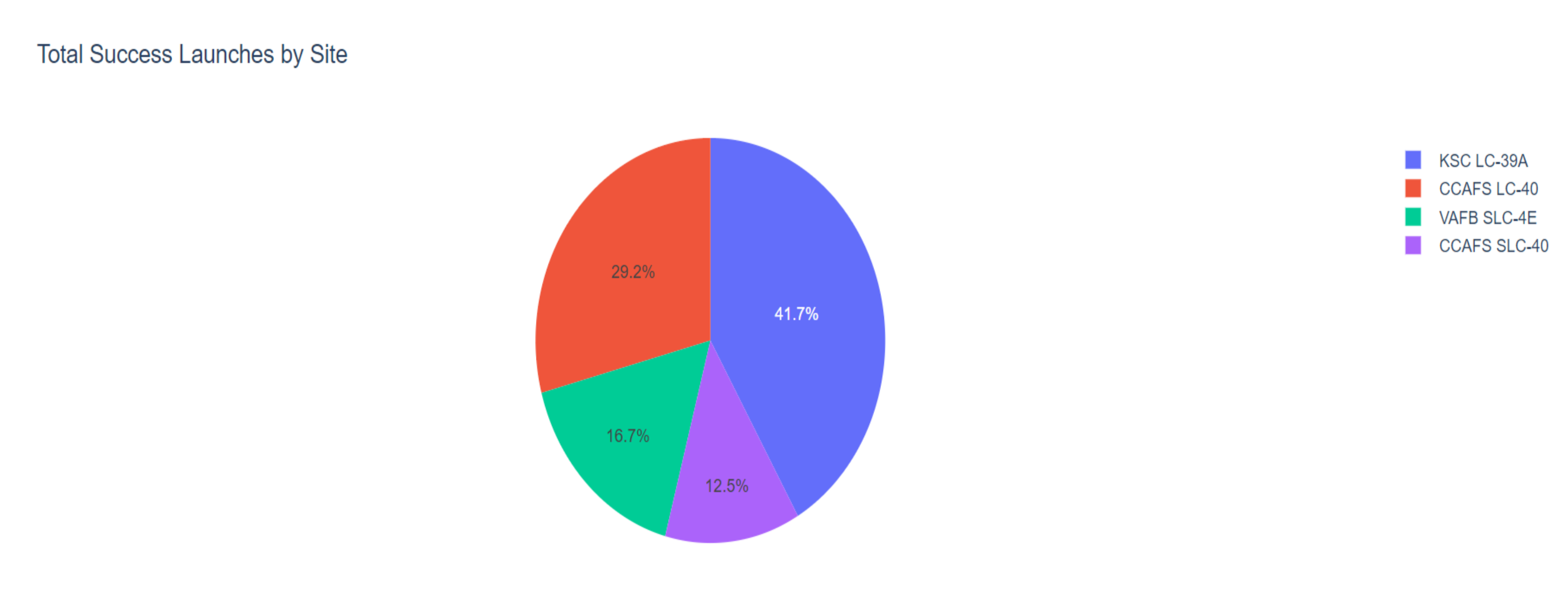
CCAFS SLC-40 will be used as an example in this section. This shows that CCAFS SLC-40 is nearer to several landmarks such as the road (0.58km), Samuel C Phillips Parkway, coastlines (0.86km) and railroad (1.28km). The farthest landmark is Olando Melbourne International Airport (51.43km).

Section 4

# Build a Dashboard with Plotly Dash

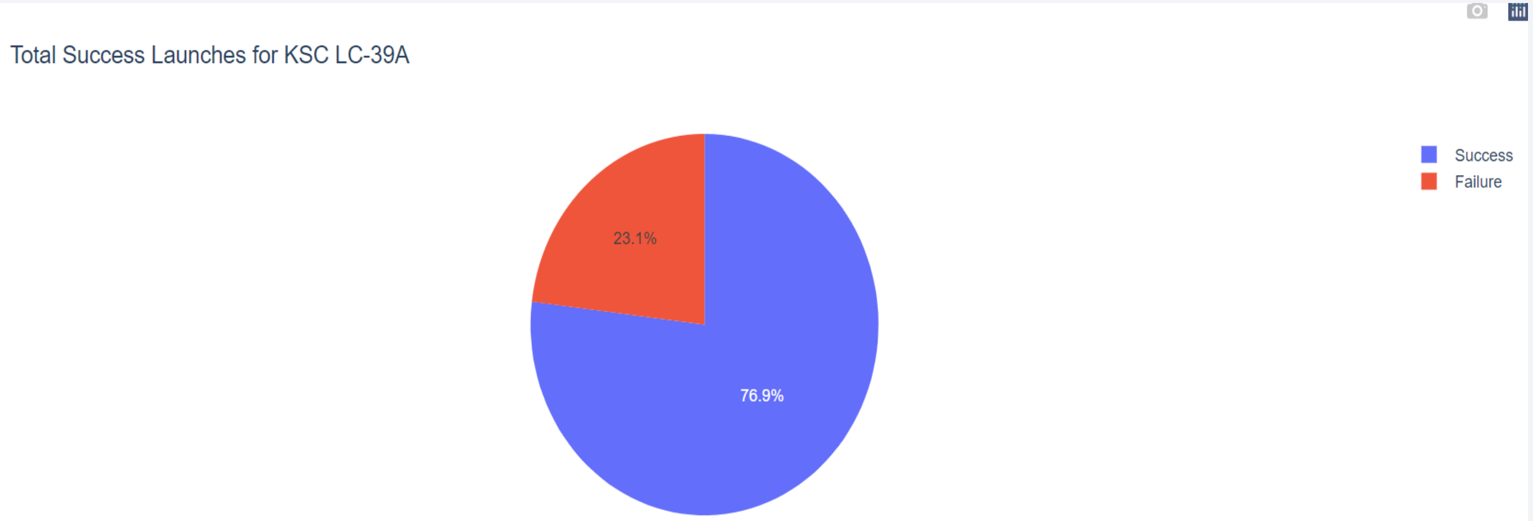


# Dashboard Screenshot 1: Successful Launches Across All Launch Sites



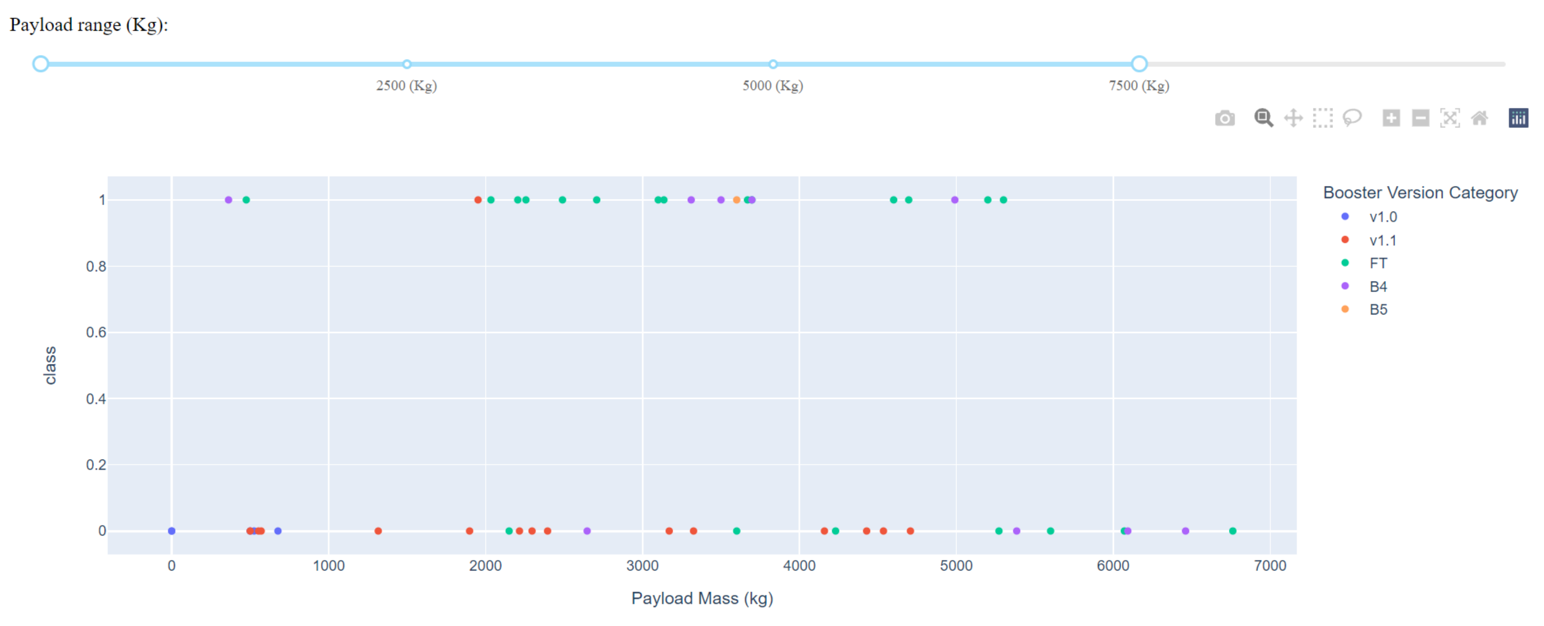
The pie chart displays the distribution of successful landings across all launch sites. CCAFS LC-40 is an old name of CCAFS SLC-40, so both CCAFS and KSC sites have the same highest weightage of successful launches (41.7%). On the other hand, VAFB has the lowest weightage of the successful launches due to the smaller sample and higher difficulty of launching in the west coast.

# Dashboard Screenshot 2: Highest Launch Success Ratio



KSC LC-39A has the highest launch success ratio with 10 successful landings and 3 failed landings.

# Dashboard Screenshot 3: Payload vs Launch Outcome

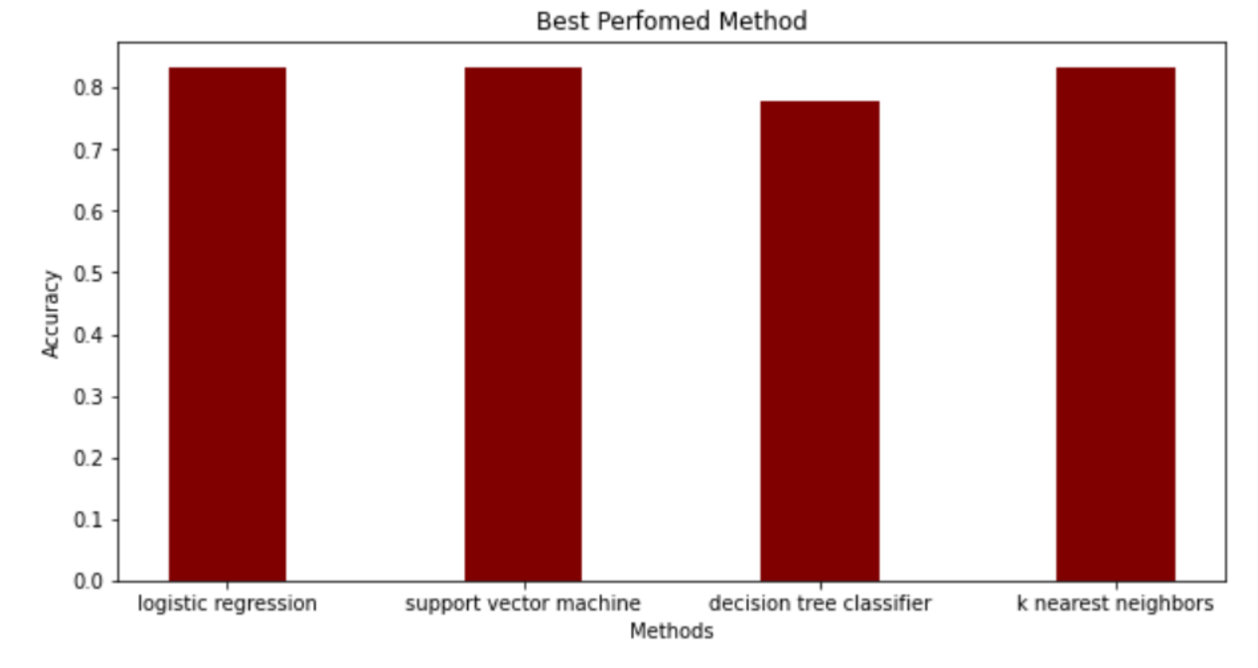


Plotly Dashboard has a Payload Mass range slider with the range of 0 – 10000. The above scatter plot shows an example of the payload mass range from 0 to 7500kg. Class 1 indicates successful landings whil Class 0 indicates unsuccessful landings. Interestingly, the scatter plot shows that there are numerous unsuccessful landings as indicated in Class 0 as compared to Class 1.

Section 5

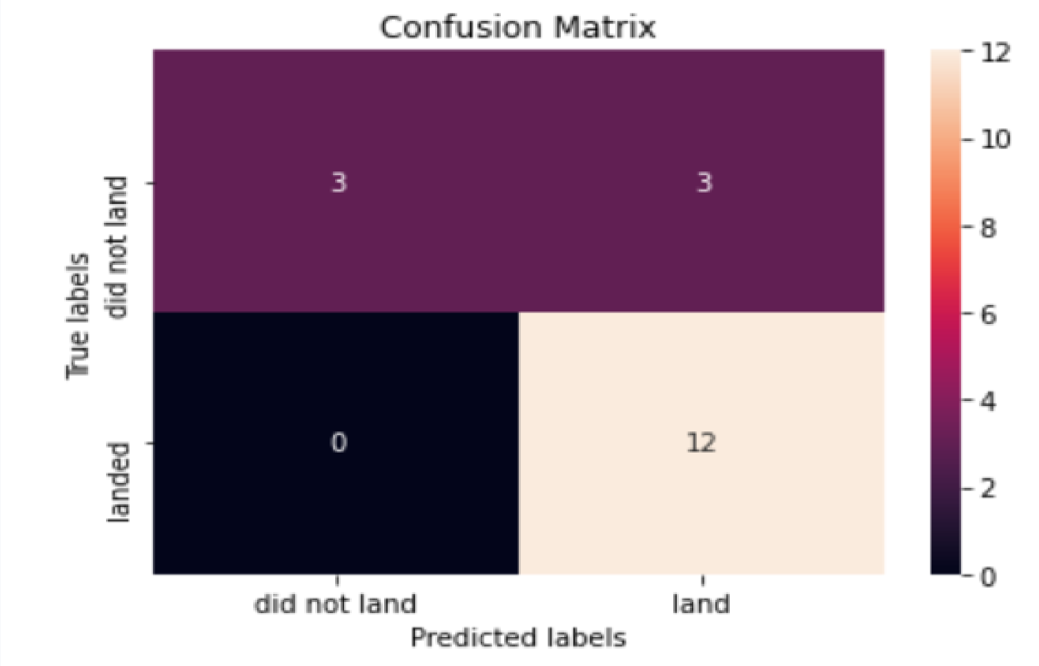
# Predictive Analysis (Classification)

# Classification Accuracy



All methods have the <u>same accuracy</u> (80%) on the test set with the <u>test size = 0.2</u> and <u>random state = 2</u>. Since the test size is smaller, we only have <u>18 test samples</u> altogether. This would cause large variance in accuracy, such as Logistic Regression and Decision Tree Classifier, in the repeated runs. Hence, we require more data, test size and random state to determine the best training model in terms of accuracy.

# Confusion Matrix



Since all training models have the same results as shown in slide 44, the confusion matrix is the same across all of them. As shown in the above confusion matrix, the models have predicted that there are 12 successful landings when the true label was successful landing (True Negative); and no unsuccessful landings when the true label was unsuccessful landing (False Negative). On the other hand, the models have predicted that there are 3 unsuccessful landings when the true label was unsuccessful landing (True Positive); and there are 3 successful landings when the true label was unsuccessful landings (False Positive).

# Conclusions

- SpaceY wants to compete with SpaceX.
- As a data scientist in Space Y, the objective is to develop a machine learning model by predicting the first stage of successful recovery to save USD $100 million approximately.
- Collected data from SpaceX API using GET request and used web scraping in SpaceX Wikipedia page.
- Created dashboards for visualisations.
- Executed SQL queries in IBM DB2 SQL database.
- Created a machine learning model with 80% accuracy.
- Will be better to collect more data for determining the best machine learning model and enhance its accuracy.

# Appendix

GitHub repository for IBM Data Science Capstone Project:

https://github.com/jaketee93/capstone_project

Special Thanks To All IBM Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors

Instructors:

Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

# Thank you!