IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ABDUL AHID
22/02/2023

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data Collection Through API
  - ➢ Data Collection With Web Scraping
  - ➢ Data Wrangling
  - ➢ EDA with SQL
  - ➢ EDA with Data Visualization
  - ➢ Interactive Visual Analytics With Folium
  - ➢ Machine Learning Prediction
- Summary of all results
  - ➢ Exploratory Data Analysis Result
  - ➢ Interactive Analytics in Screenshot
  - ➢ Predictive Analytics results

Introduction

- Project background and context

  SpaceX is a company that works on reusable first stage space missions. SpaceX advertise on its website Falcon 9 rocket launch for 9 million dollars whereas competitors advertise it for 165 millions dollar. The main reason of cost effectiveness of the Falcon 9 as compared to other is because its reuses its first stage. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - What factors that determine the rocket will land successfully
  - Interaction of various features that determine the rocket will land successfully
  - What conditions help in the successful landing of the rocket

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected by Space X API and Web Scraping from Wikipedia.
- Perform data wrangling
  - Data wrangling was done by eliminating null values with mass mean and then doing one hot encoding of categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Ploty Dash
- Perform predictive analysis using classification models
  - Predictive modeling was done using various machine learning algorithm like Logistic Regression, SVM,KNN and trees algorithm and they were evaluated on the bases of accuracy.

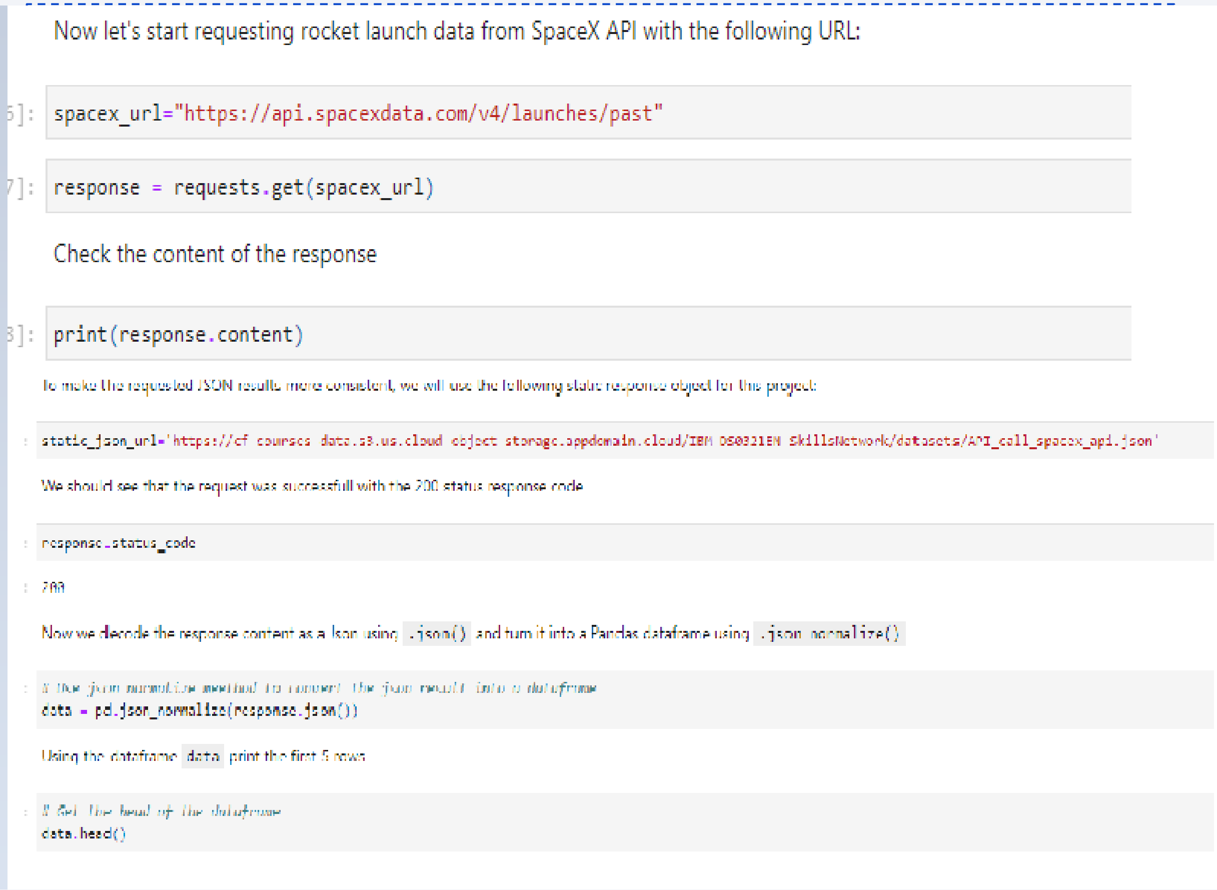
Data Collection

- Describe how data sets were collected.
  - ➢ 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 data frame 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 data frame for future analysis.

Data Collection – SpaceX API

- First of all we import request pandas and Numpy libraries from python. After that we get response from SpaceX API by request.get function. The data set was too large so we get response static by json static URL. After we converted the json to data frame by json_normalize function. As the resulted data have some falcon 1 rocket data also so we removed falcon 1 data and only kept falcon 9 data after that we go for null values and replaced null values from PayLoadMass from mean.

- The link to the notebook is https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/Data%20Collection%20API%20Lab.ipynb



Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

Check the content of the response

```
print(response.content)
```

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

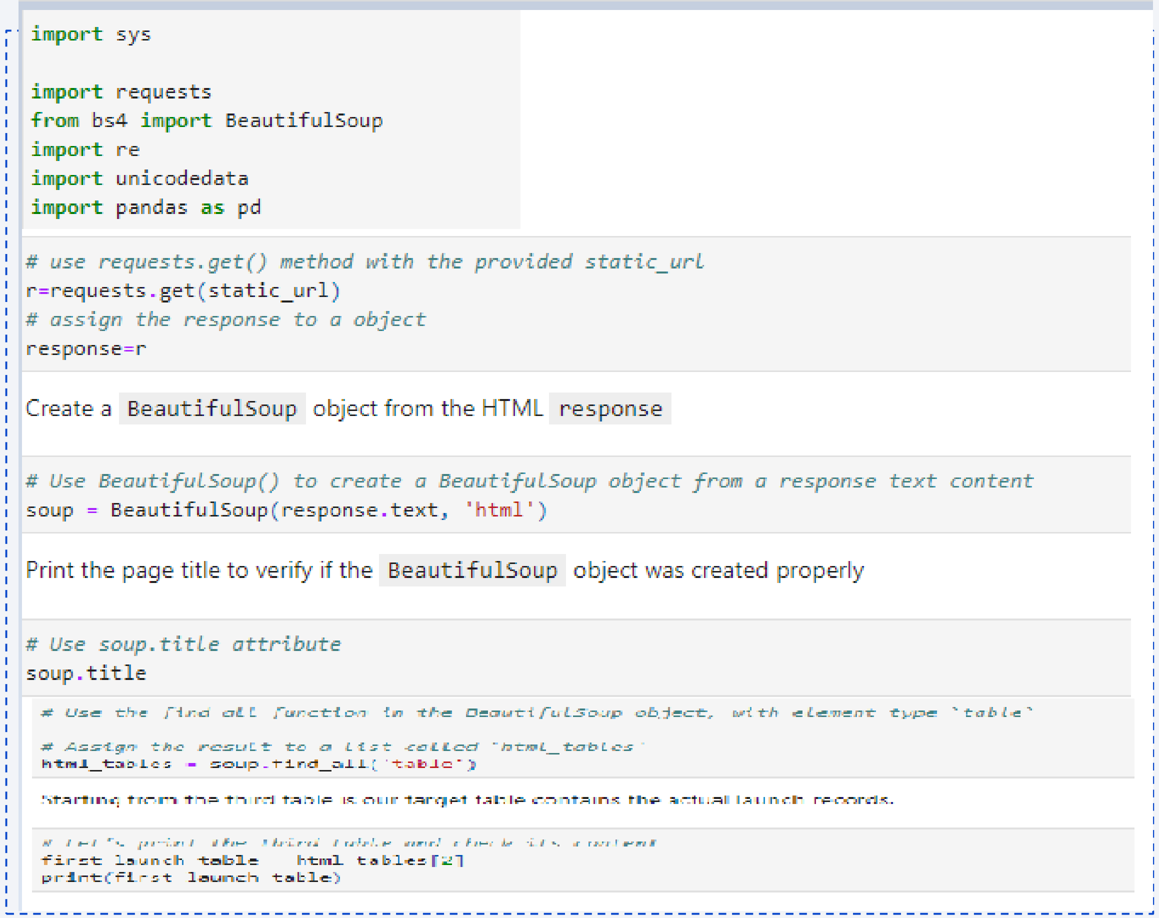
Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head()
```

Data Collection - Scraping

- First of all we imported all required liberaries.After that we define a response variable to get data from URL and then we make a soup object and parse the response into HTML.The scraped data is stored in dataframe using pandas

- Github https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/Data%20Collection%20with%20Web%20Scraping%20lab.ipynb



```python
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

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

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

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html')
```

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

```python
# Use soup.title attribute
soup.title
```

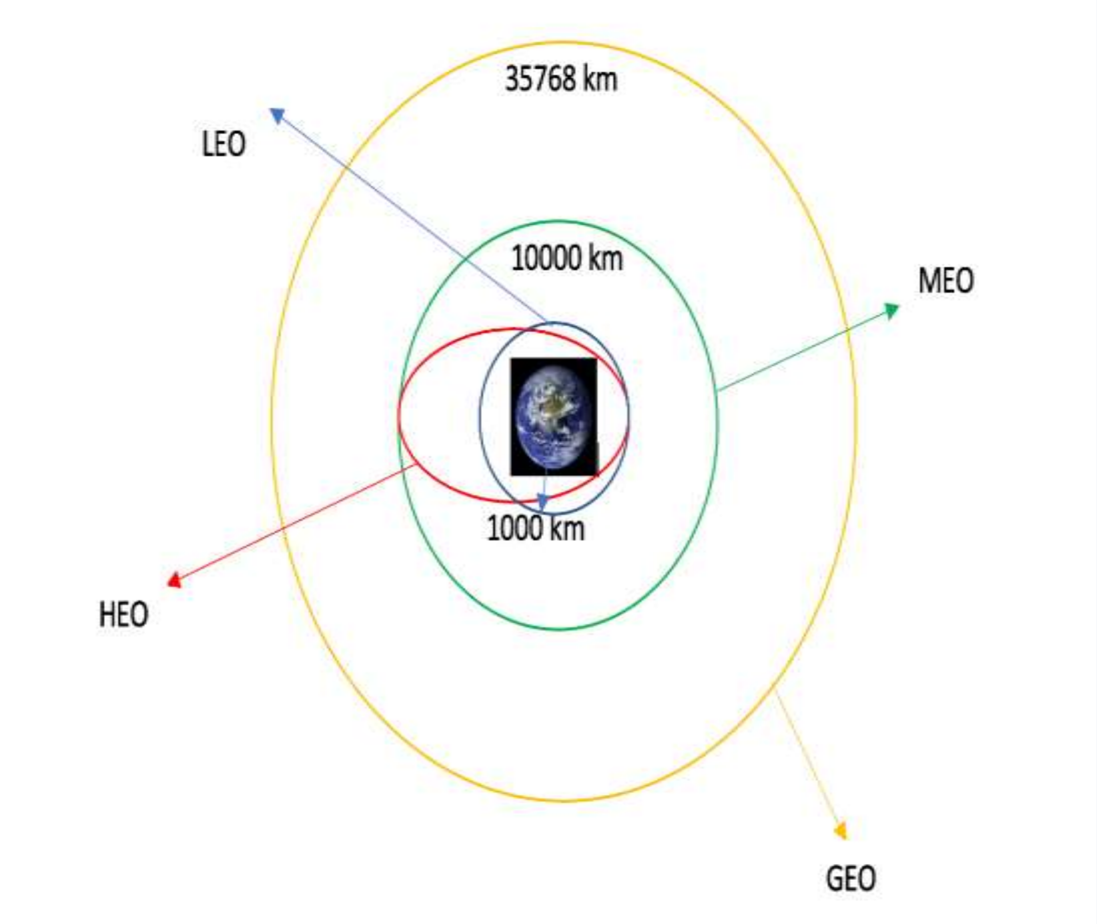
```python
# Use the find_all function in the BeautifulSoup object, with element type `table`

# 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.

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

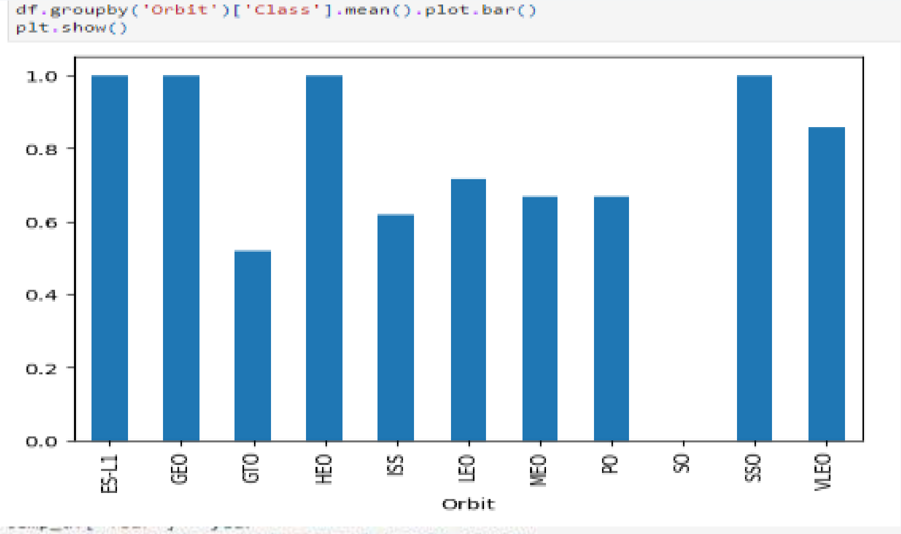
Data Wrangling

- EDA was done to get training labels for supervised learning.
- Value_count method was used to get launchsite,orbit and outcome. And after that how many landing the launchsite,orbit and outcome had.
- We used a lambda function to make all outcome 0 as bad outcomes and 1 as good outcome. Made a new column class for outcomes. After that we took mean of class to get the success rate.
- GitHub:https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/Data%20Wrangling.ipynb

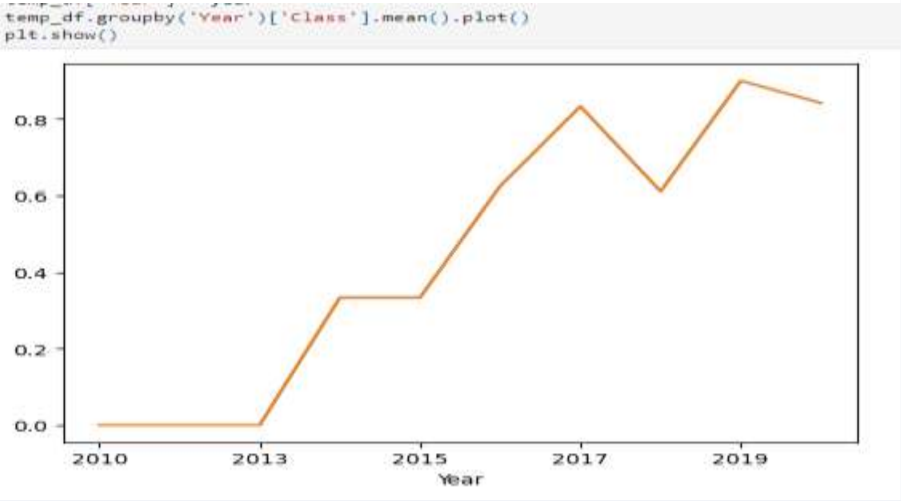


EDA with Data Visualization

- 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.
- Graphs like bar chart scatter point plot and line plot were used to get useful insight of data.
- Github:https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/EDA%20with%20Visualization%20lab.ipynb



```
df.groupby('Orbit')['Class'].mean().plot.bar()
plt.show()
```



```
temp_df.groupby('Year')['Class'].mean().plot()
plt.show()
```

EDA with SQL

- Data was loaded in IBM cloud DB2 data base by creating a new table SPACEX_TABLE.
- Connection string to DB2 data base was added in Jupiter notebook to make a connection to the DB2 database and query the required table.
- After the connection was made to the data base we did the exploratory data analysis by writing following queries
  - Name of unique site launch in the space mission
  - Total mass payload carried by booster launched by NASA (CRC)
  - Average payload carried out by booster version F9 v1.1
  - Name of boosters that have success rate in drone ship and payload mass is greater then 4000 and less then 6000 lbs
  - The total number of successful and unsuccessful outcomes
  - The failed landing outcome in booster, drone ship and launch site names
  - Github:https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/EDA%20WITH%20SQL%20LAB.ipynb

Build an Interactive Map with Folium

- We marked all launch sites and added objects on map such as markers, circles and lines to show the success and failure of the launches for each site on the map
- We assigned the feature assign outcome class I.e. 1 for success and 0 for failure.
- Through color labeled markers clusters we can easily determine the site that have high or low success rate.
- We calculated the distance between launch site and its close proximities.
- Github:https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

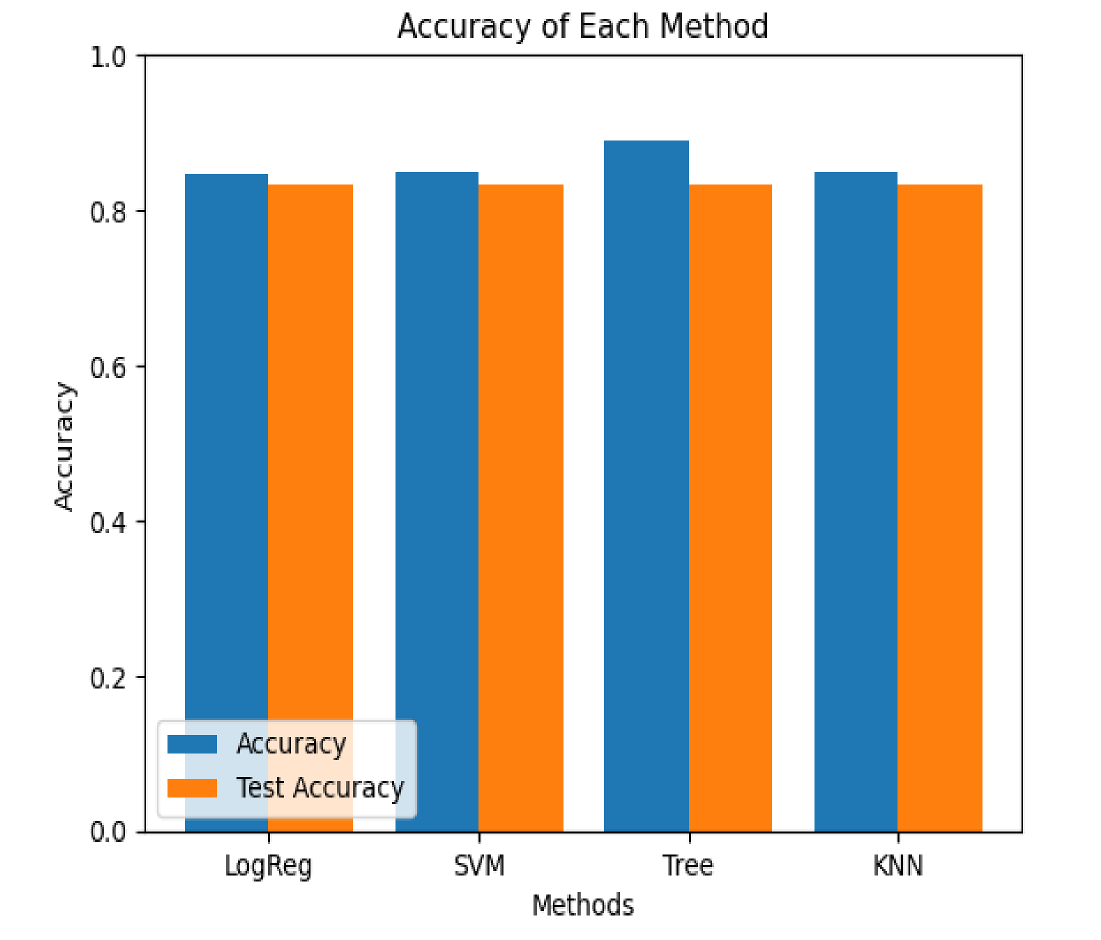
Build a Dashboard with Ploty Dash

- We build an interactive dashboard using Ploty dash
- A pie chart was built to show the outcomes of launch from a certain launch site.
- We plotted the scatter plot the show the relationships between payload mass (kg) and outcome for different booster version.
- Github:https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/Plotly%20app.ipynb

Predictive Analysis (Classification)

- The main problem was to classify that the if the first stage landing was successful or not so it’s a classification problem that is part of supervised learning.
- For this purpose we used 4 classification models that are logistic regression, support vector machine, KNN and decision tree.
- These models where trained tested and evaluated by following steps
  - ❖ First of all we define Train and Test column then we split the train and test samples with test sample of size 20% and random state of 2.
  - ❖ After that we train the data by fitting the model on train set and then compare the results of test data with actual test to find the accuracy.
  - ❖ The model was tuned by gridsearchCV with the cross validation of 10
  - ❖ Accuracy was used as a performance matrix to determine the best model and the best model performing was decision tree.
- Github:https://github.com/abdulahid-cs/IBM_DATASCIENCE_CAPSTONE/blob/master/Machine%20Learning%20Prediction%20lab.ipynb

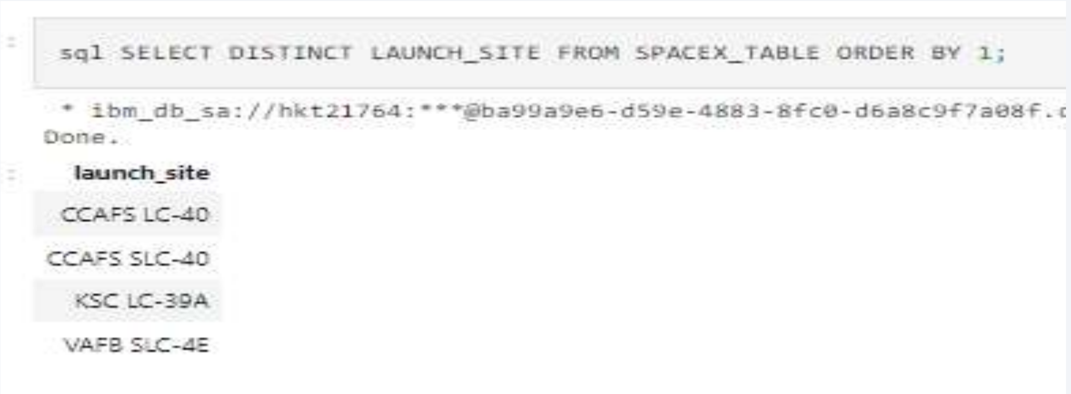
Results



Predictive analysis results

From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

Interactive analytics demo in screenshots



```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEX_TABLE ORDER BY 1;
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c
Done.
```

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

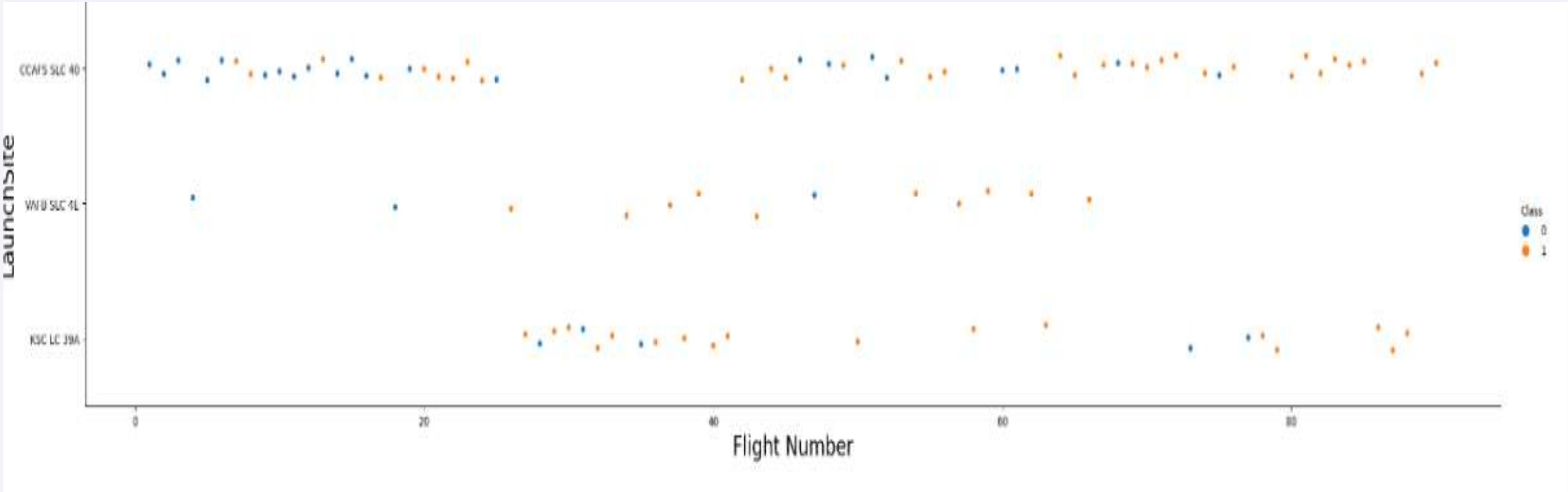
EDA WITH SQL

Section 2

Insights drawn from EDA

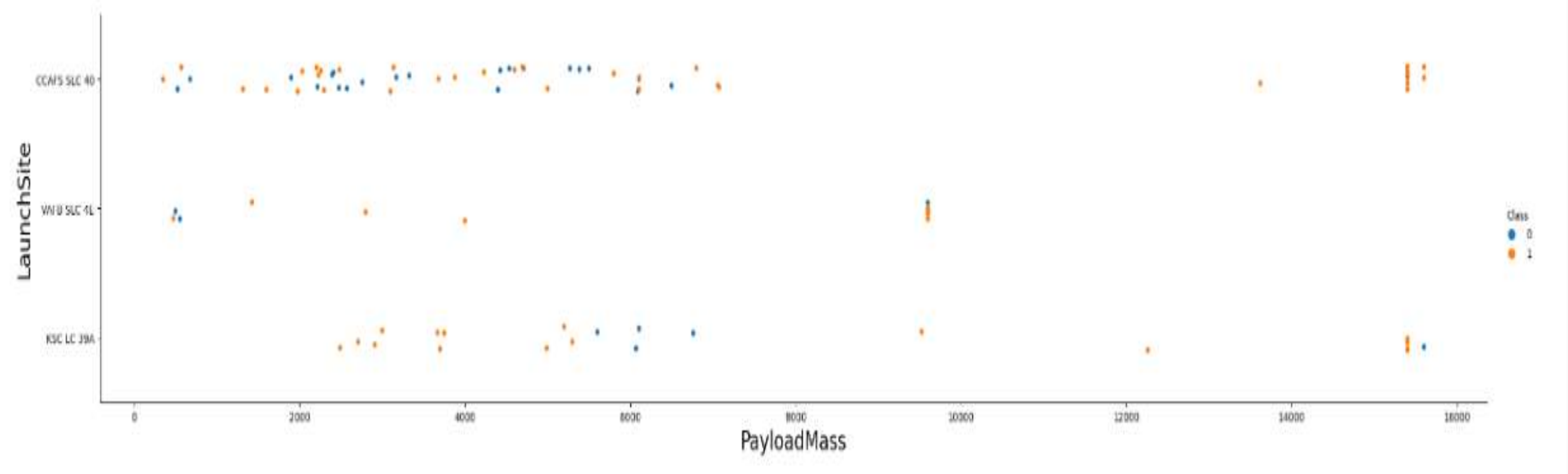
Flight Number vs. Launch Site

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



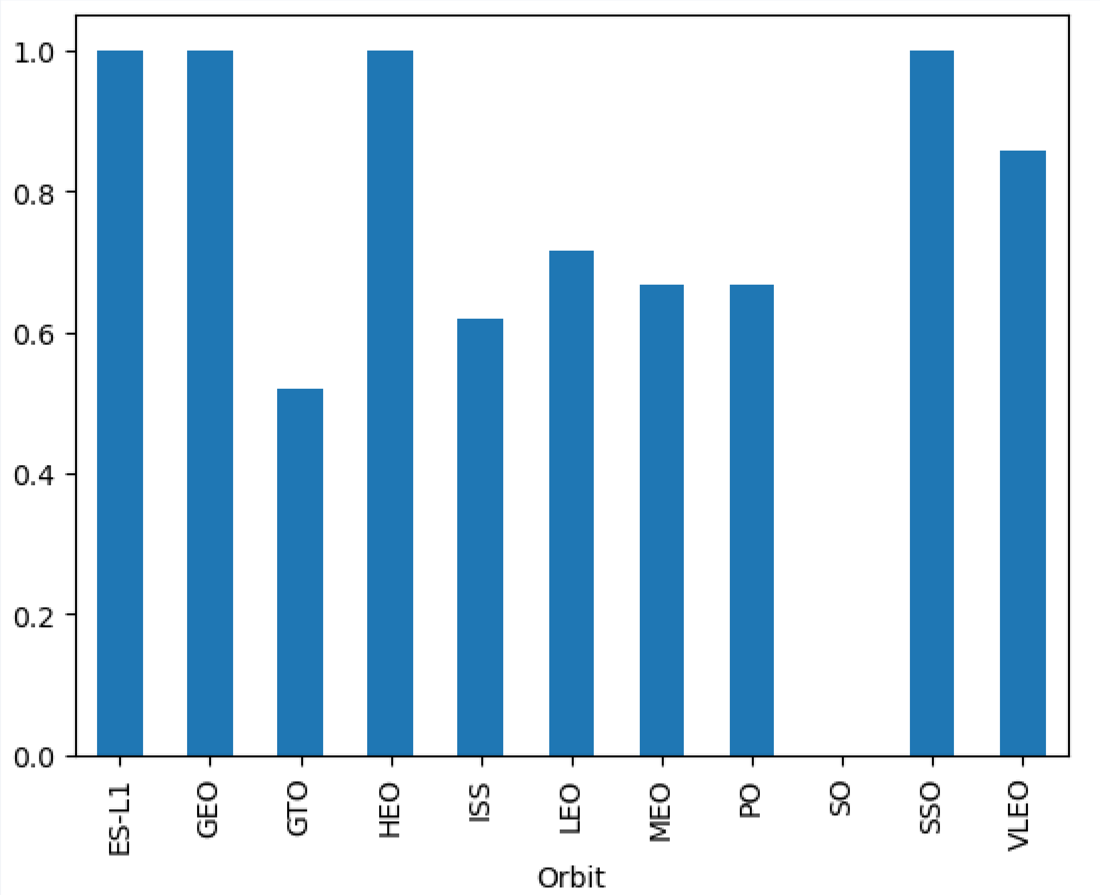
Payload vs. Launch Site

- The payload mass below 8000 kg have high success rate as compare to payload mass above that.



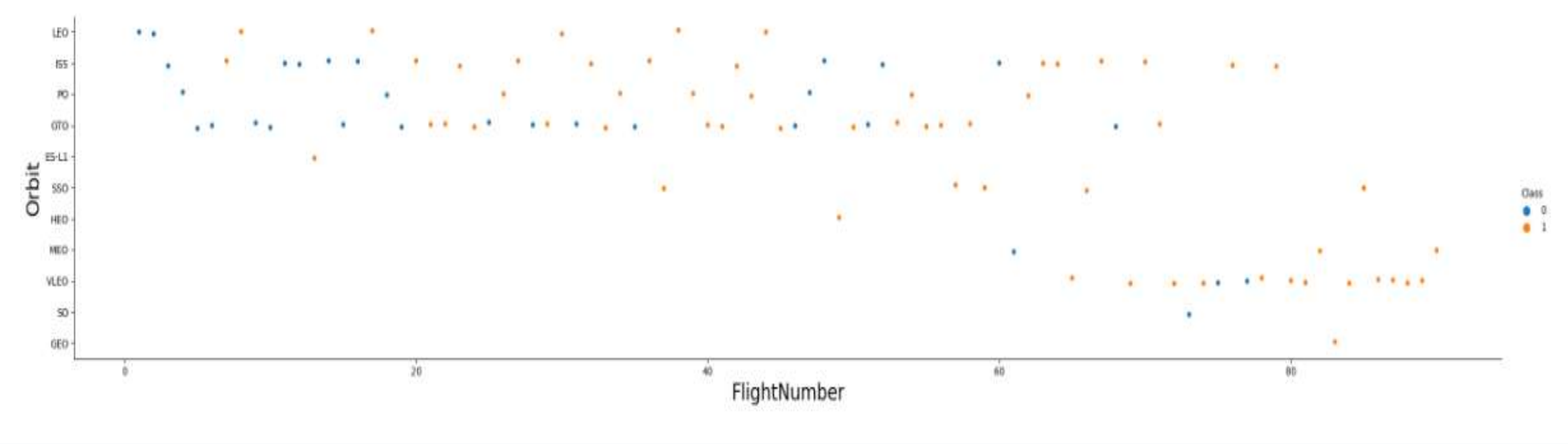
Success Rate vs. Orbit Type

- The so orbit type have no success rate and whereas the ES-L1,GEO,HEO and SOO have success rate 1



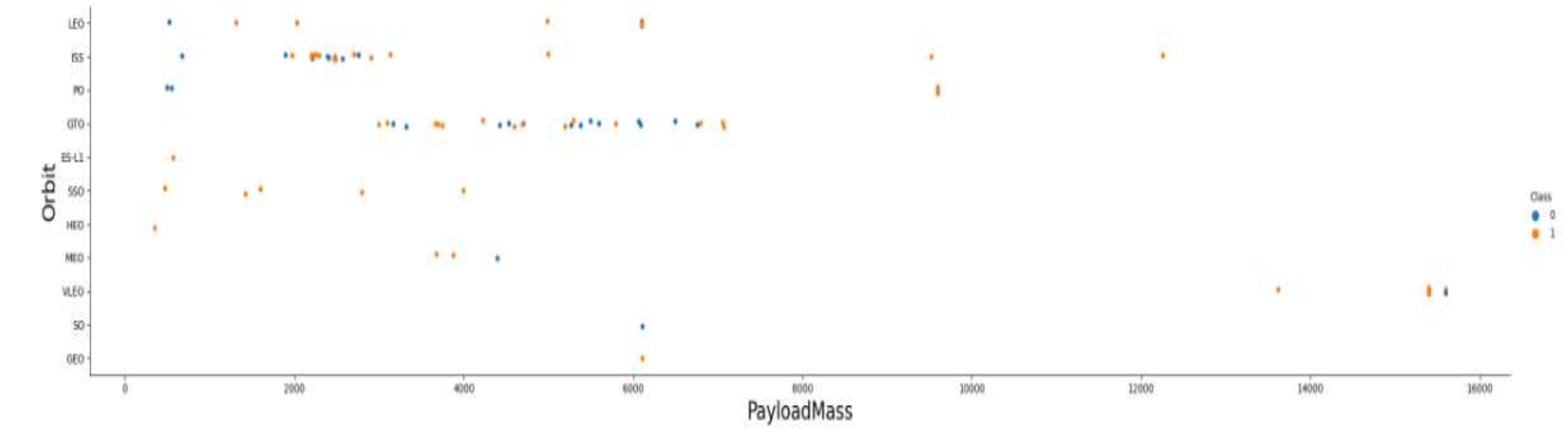
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit



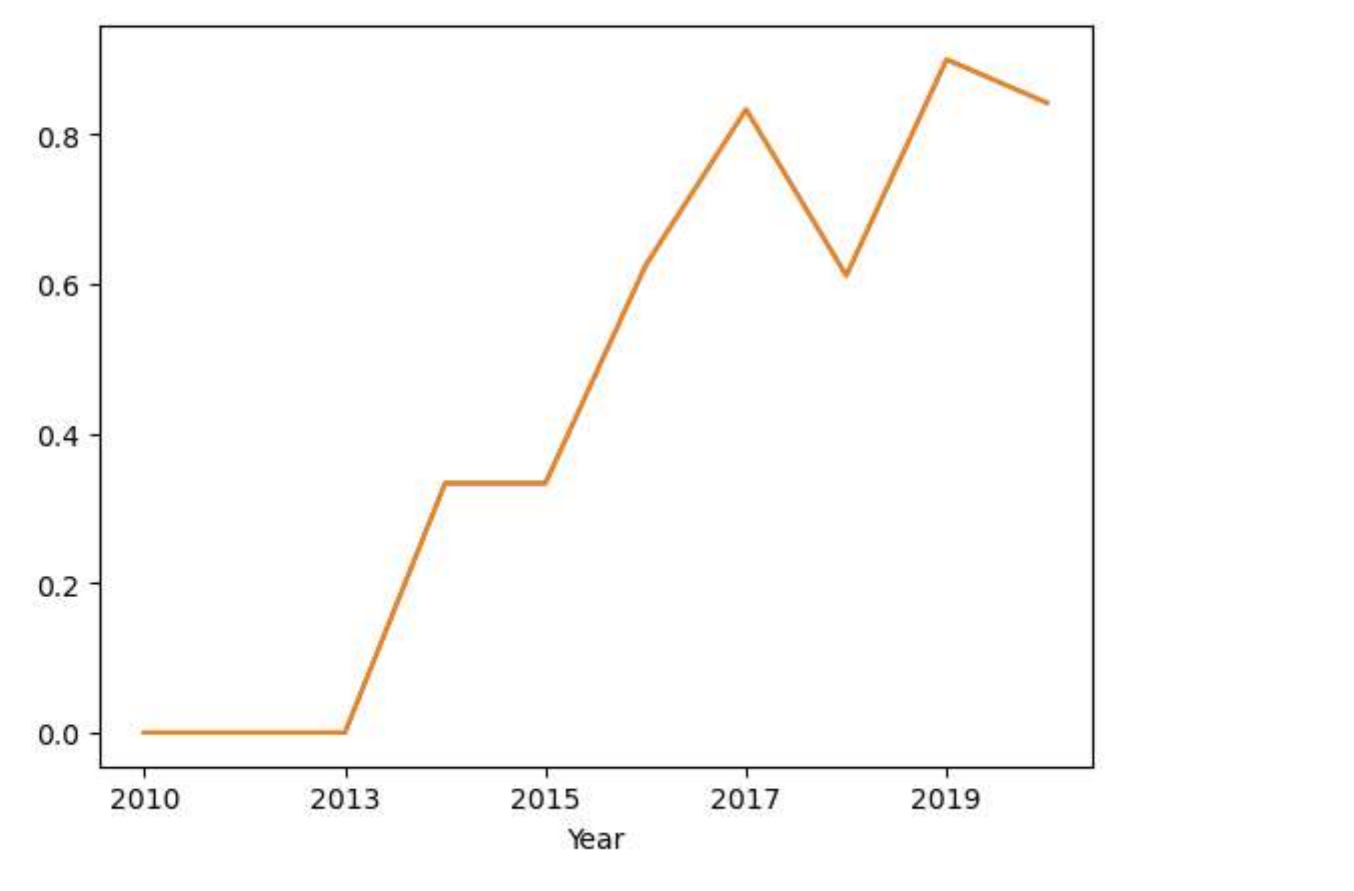
Payload vs. Orbit Type

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



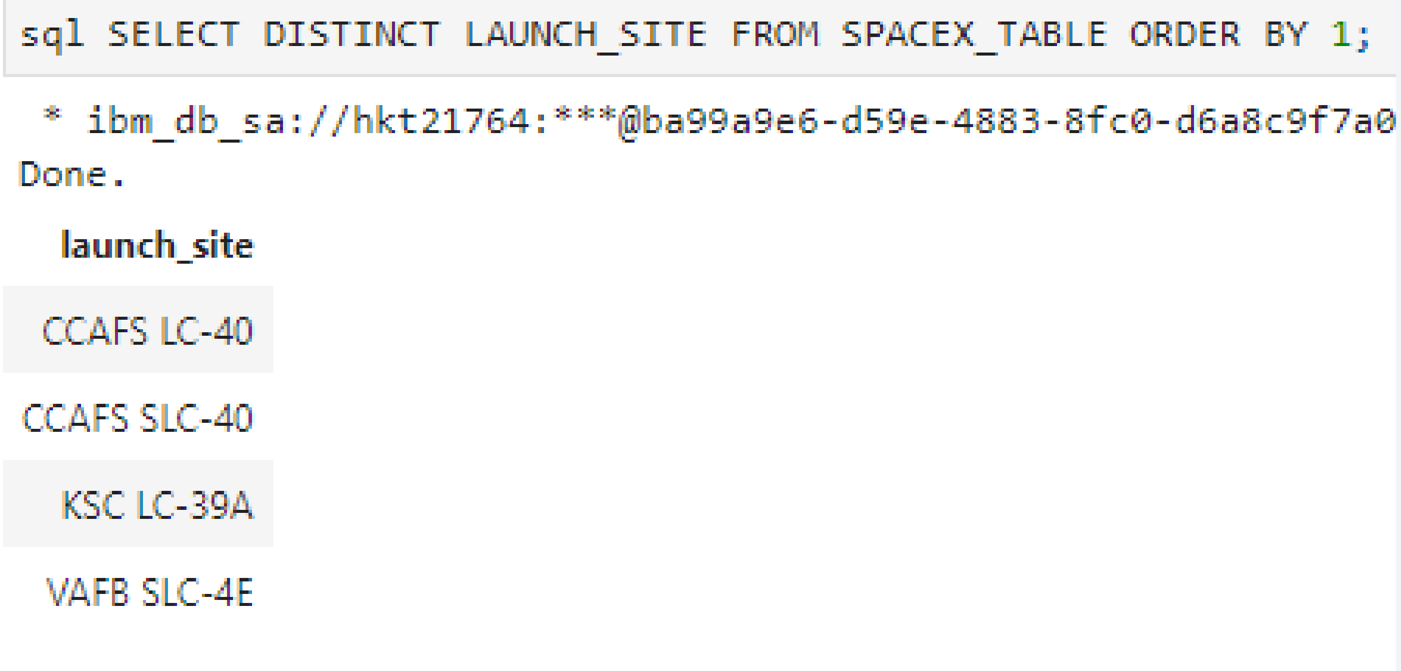
Launch Success Yearly Trend

- The highest success rate was established in between 2018 and 2019 after 2019 it started declining again



All Launch Site Names

- Distinct key word is used in SQL query to get the distinct launch site names from the table SPACEX_TABLE



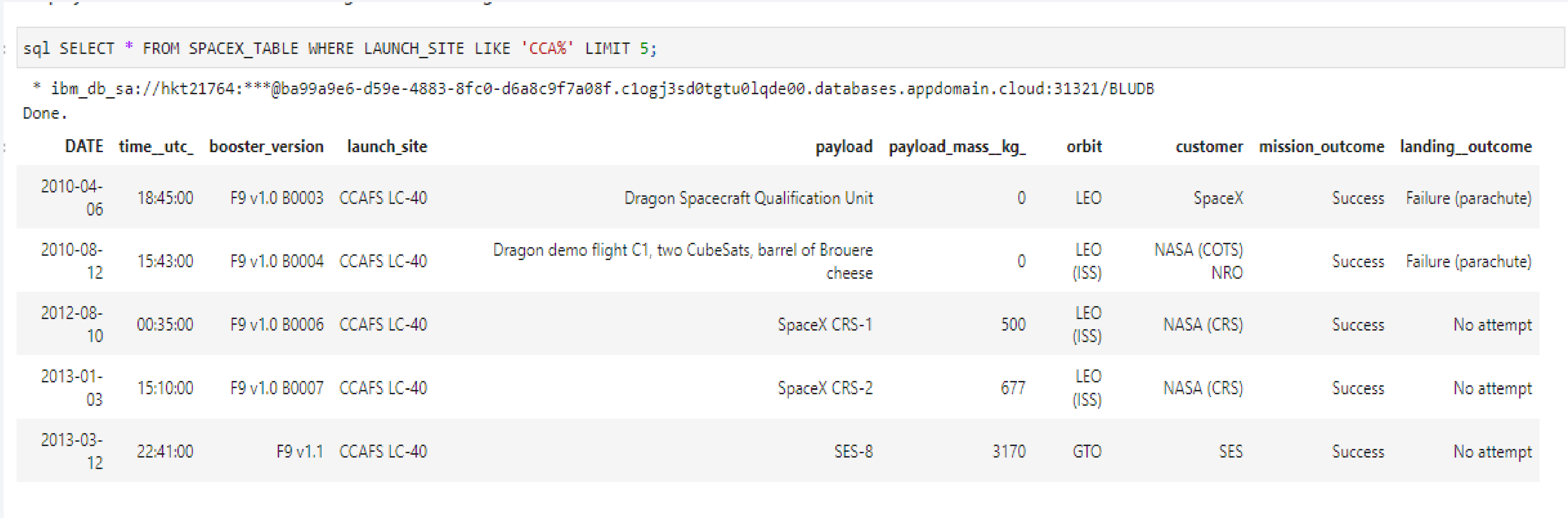
```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEX_TABLE ORDER BY 1;
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a0
Done.
```

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

Launch Site Names Begin with 'CCA'

- We used like with keyword ‘CCA’ and then limit the result for 5 to see the top 5 launch sites name beginning with CCA



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

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

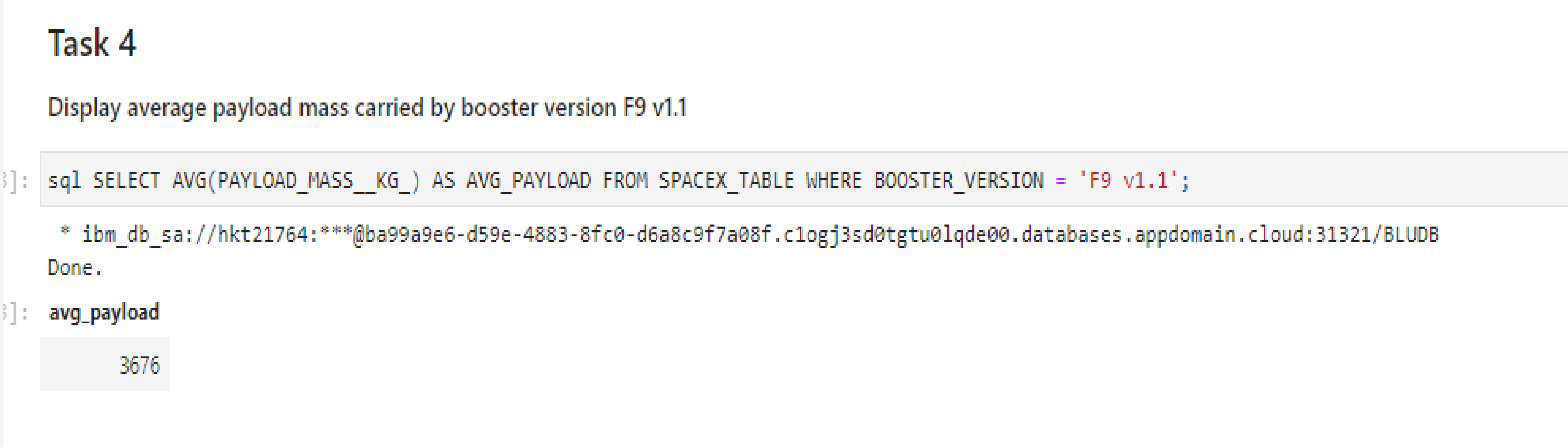
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

- Average function is used to get the average payload mass where booster version is F9 v1.1



Task 4

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

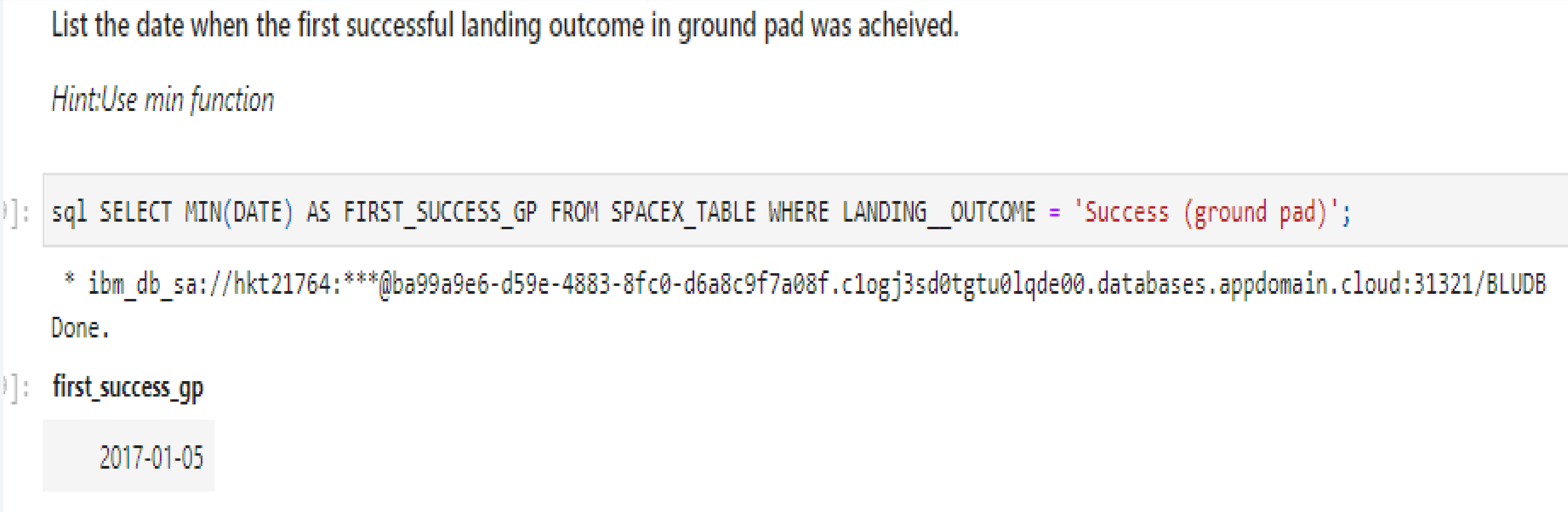
```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEX_TABLE WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

avg_payload
3676

First Successful Ground Landing Date

- Min function was use to get the first successful ground landing date



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

Hint:Use min function

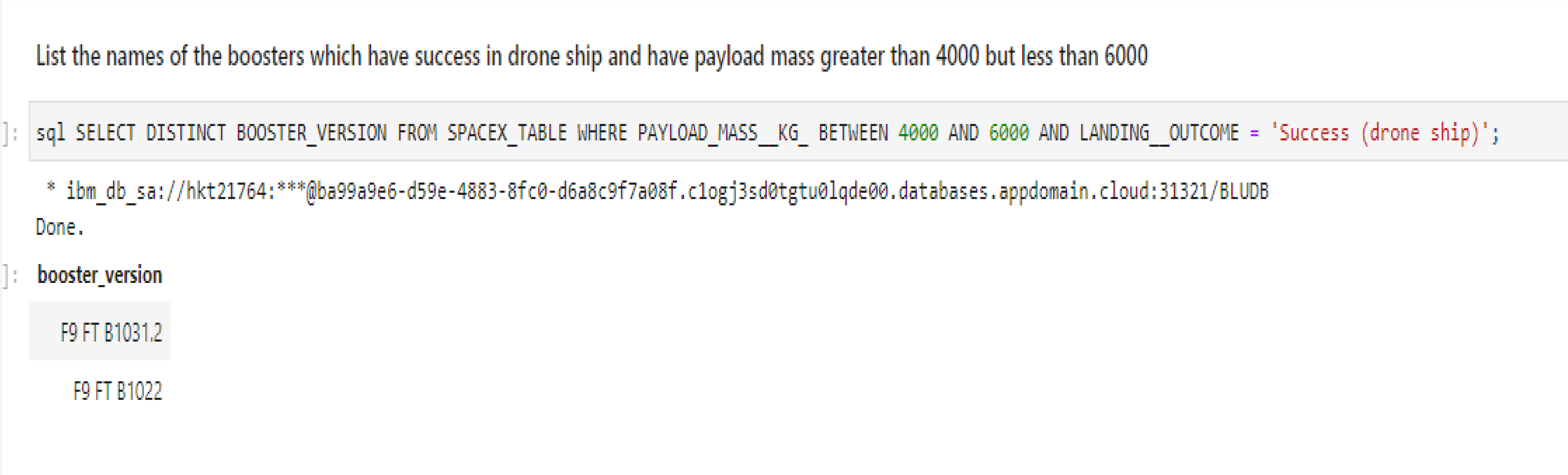
```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEX_TABLE WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

first_success_gp
2017-01-05

Successful Drone Ship Landing with Payload between 4000 and 6000

- The distinct method was used to get the names of successful booster version where drone ship payload was 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 DISTINCT BOOSTER_VERSION FROM SPACEX_TABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)';
```

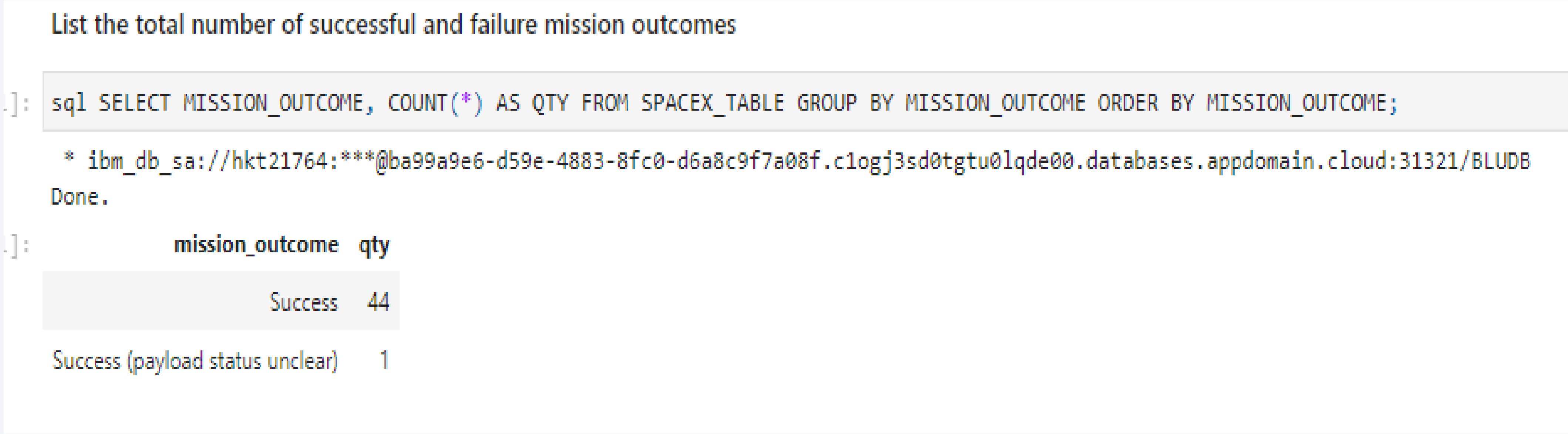
```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

]:

booster_version
F9 FT B1031.2
F9 FT B1022

Total Number of Successful and Failure Mission Outcomes

- Count was used to get successful and failure mission outcomes



List the total number of successful and failure mission outcomes

```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEX_TABLE GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

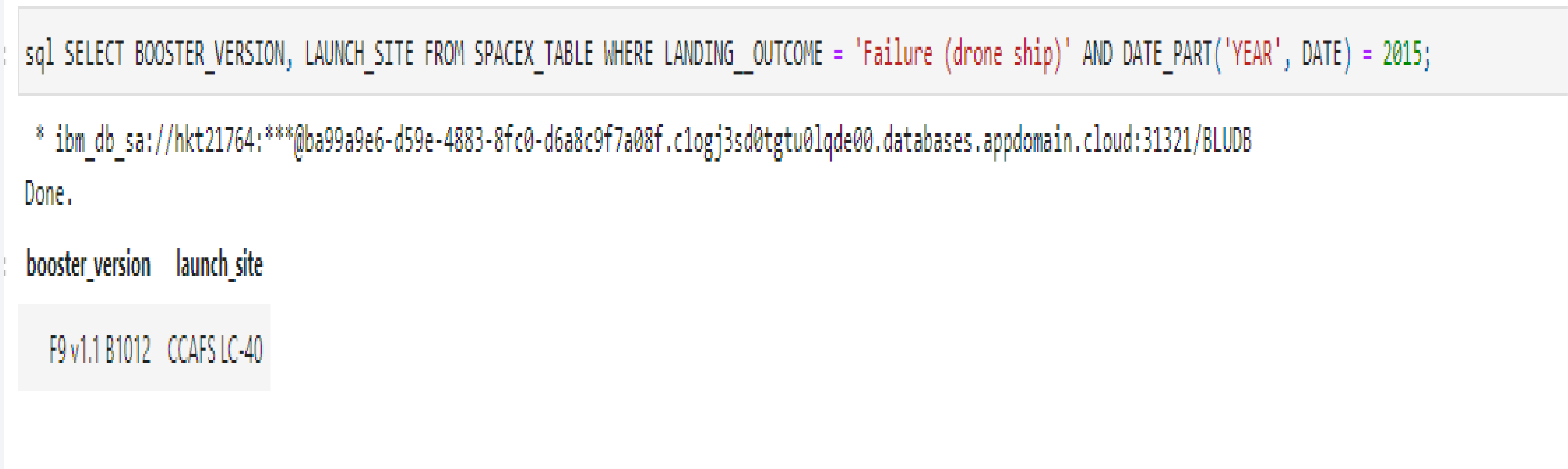
mission_outcome	qty
Success	44
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

2015 Launch Records

- Where was used and Date_part function was used



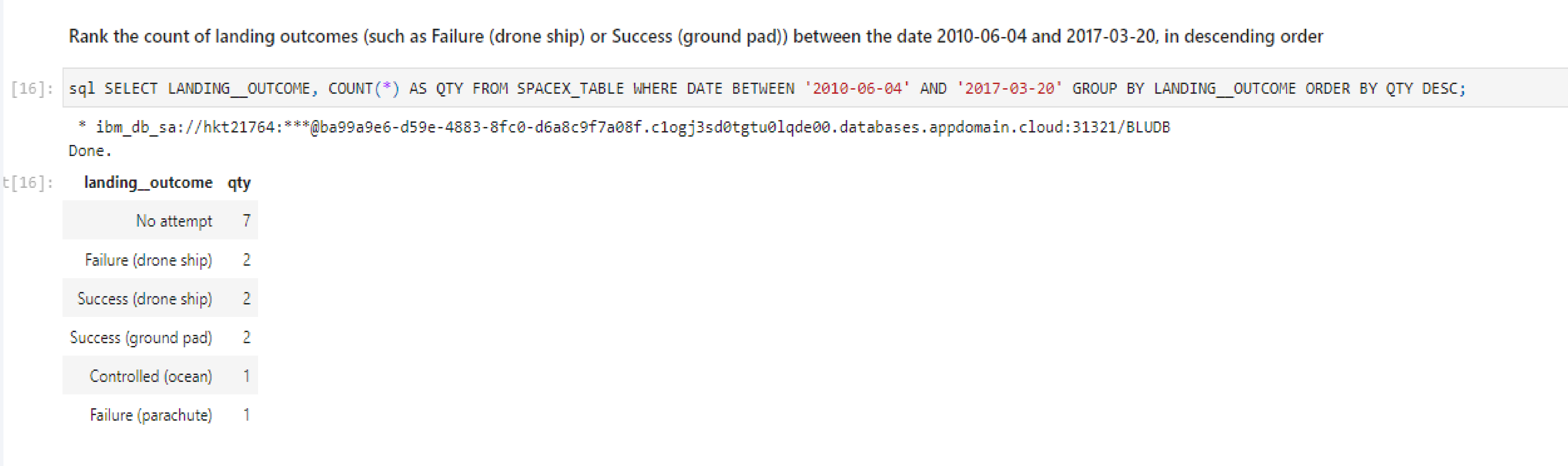
```
sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX_TABLE WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DATE) = 2015;
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40

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

- Count Where Group by was used in SQL query to get the result



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

```
[16]: sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEX_TABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY QTY DESC;
```

```
 * ibm_db_sa://hkt21764:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

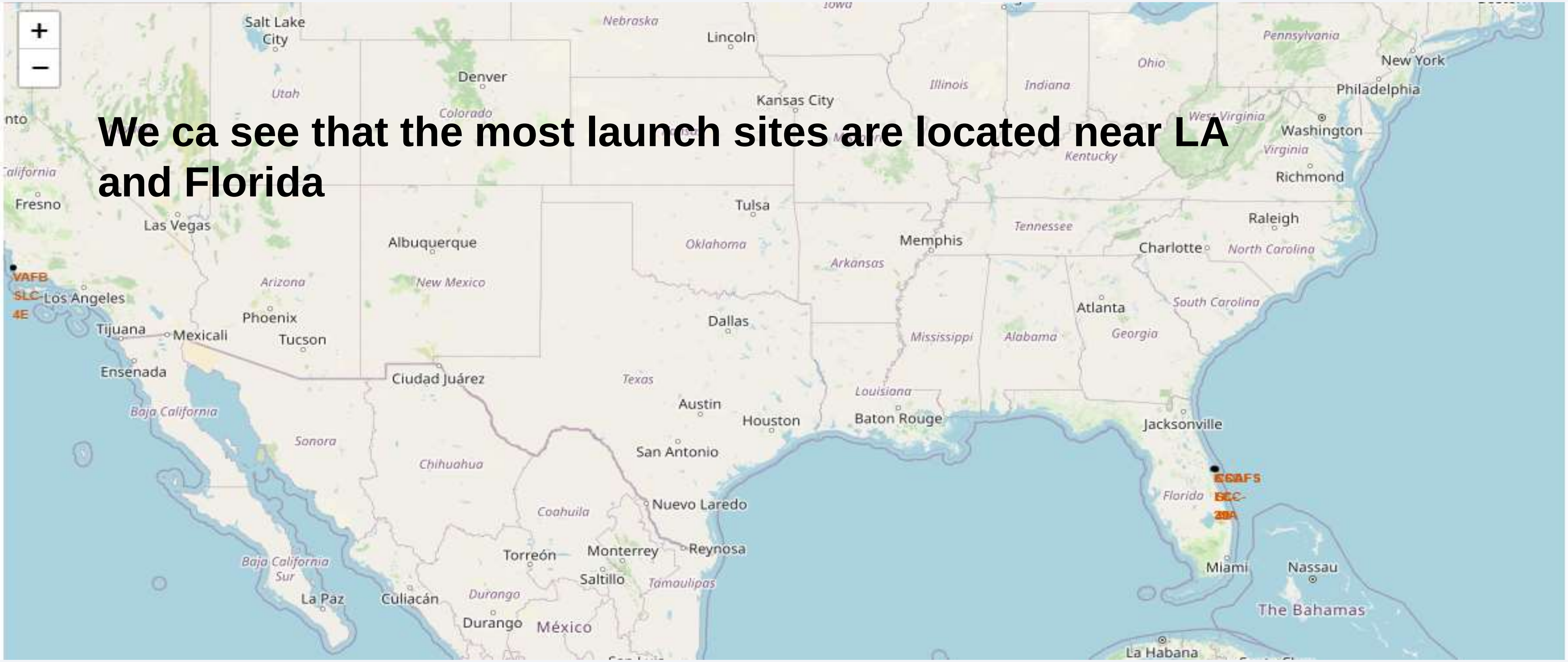
t[16]:

landing__outcome	qty
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

Section 3

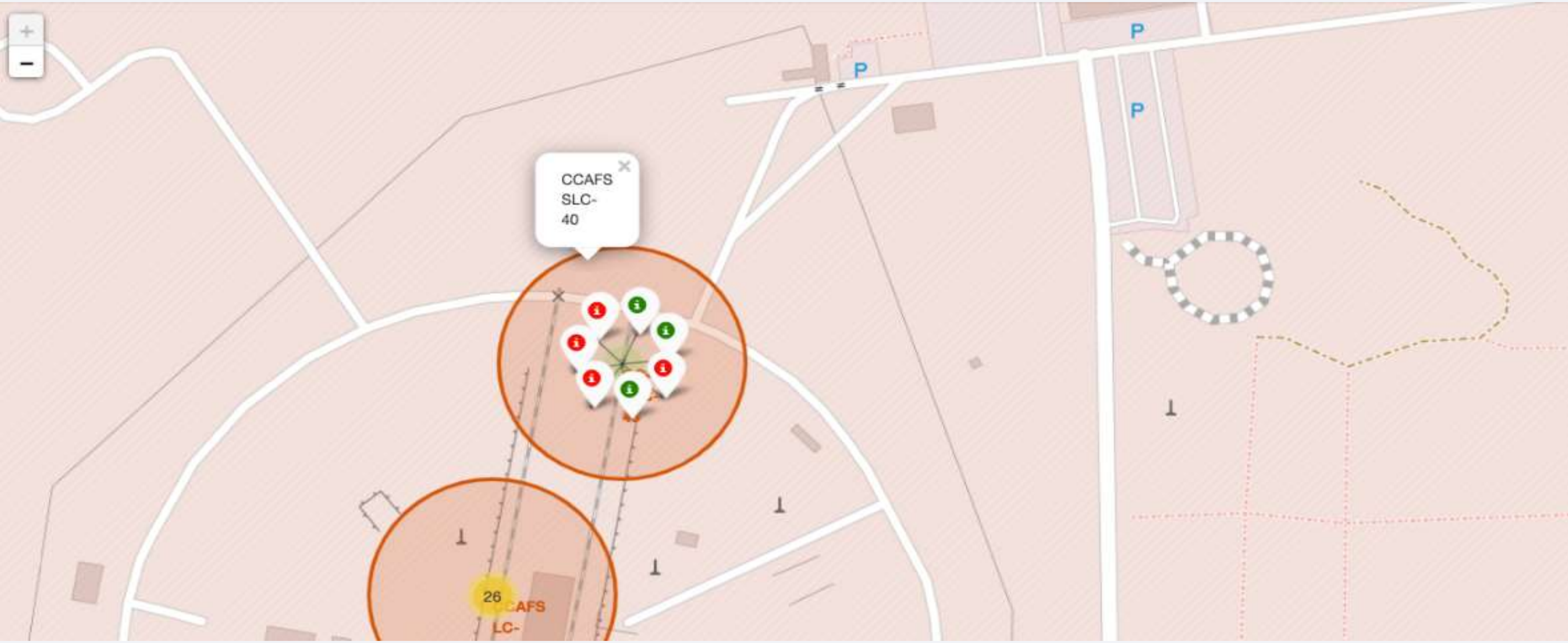
Launch Sites Proximities Analysis

ALL LAUNCH SITES GLOBAL MAP MARKERS

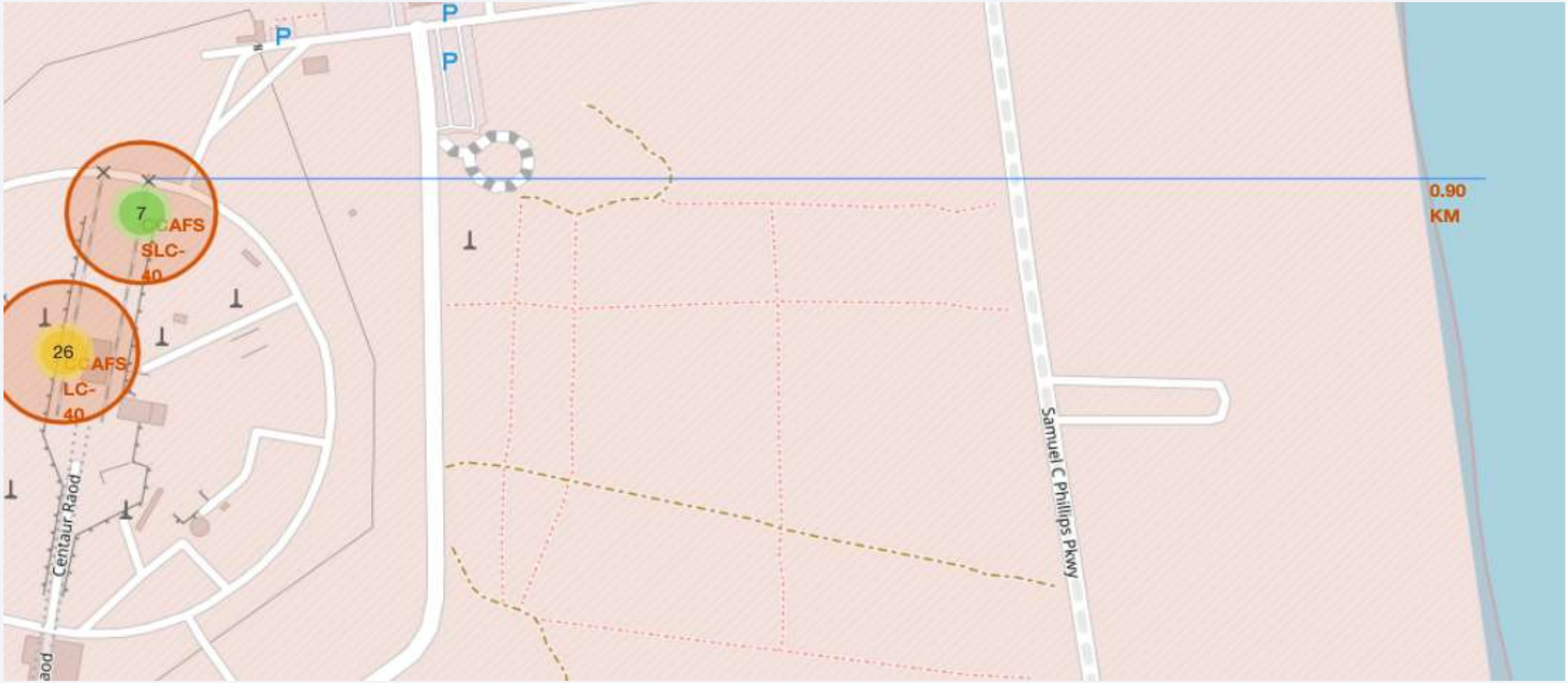


We ca see that the most launch sites are located near LA and Florida

MARK SHOWING LAUNCH SITES WITH COLOR LABELS

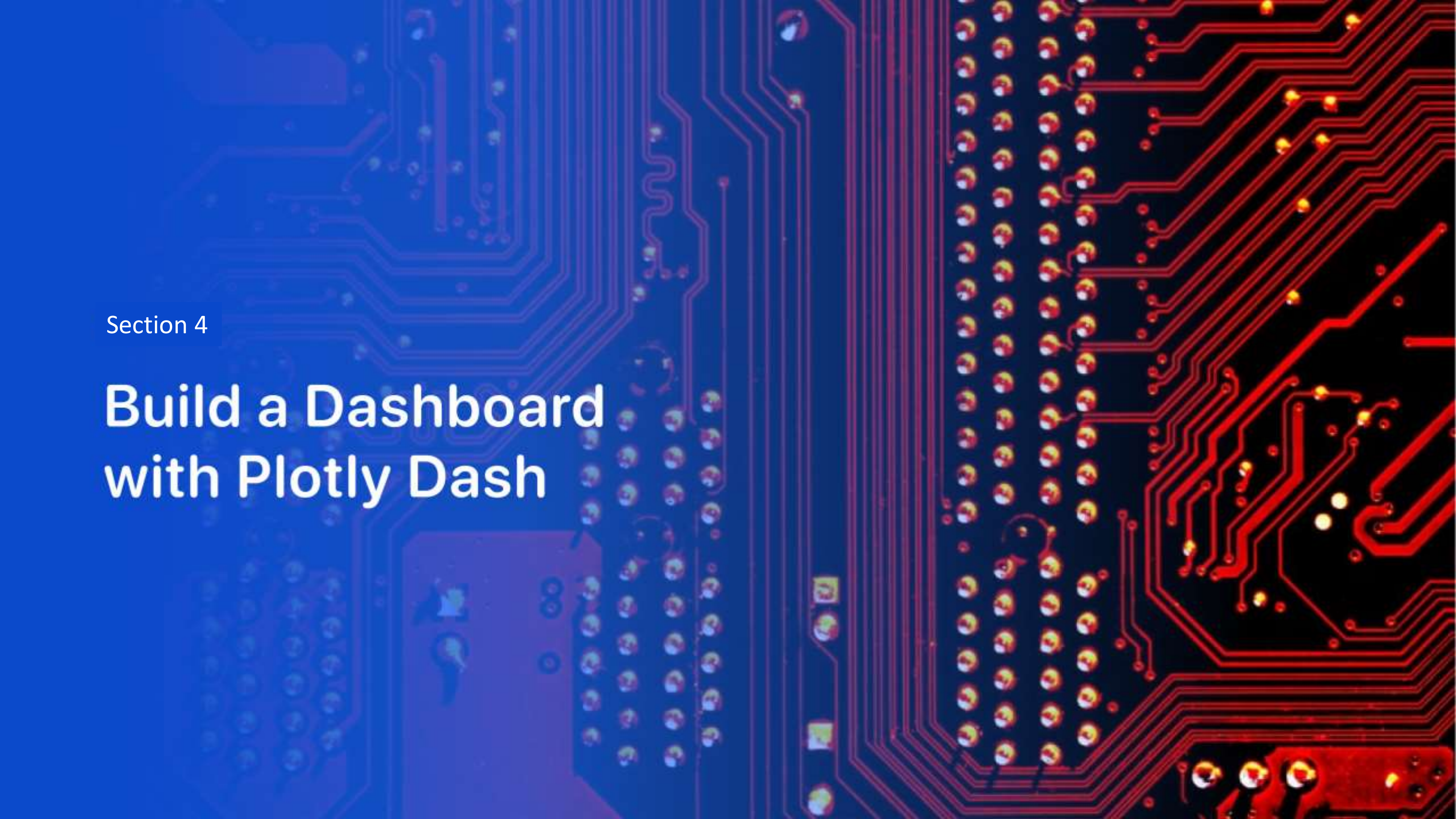


LANCH SITES DISTANCE TO LANDMARKS

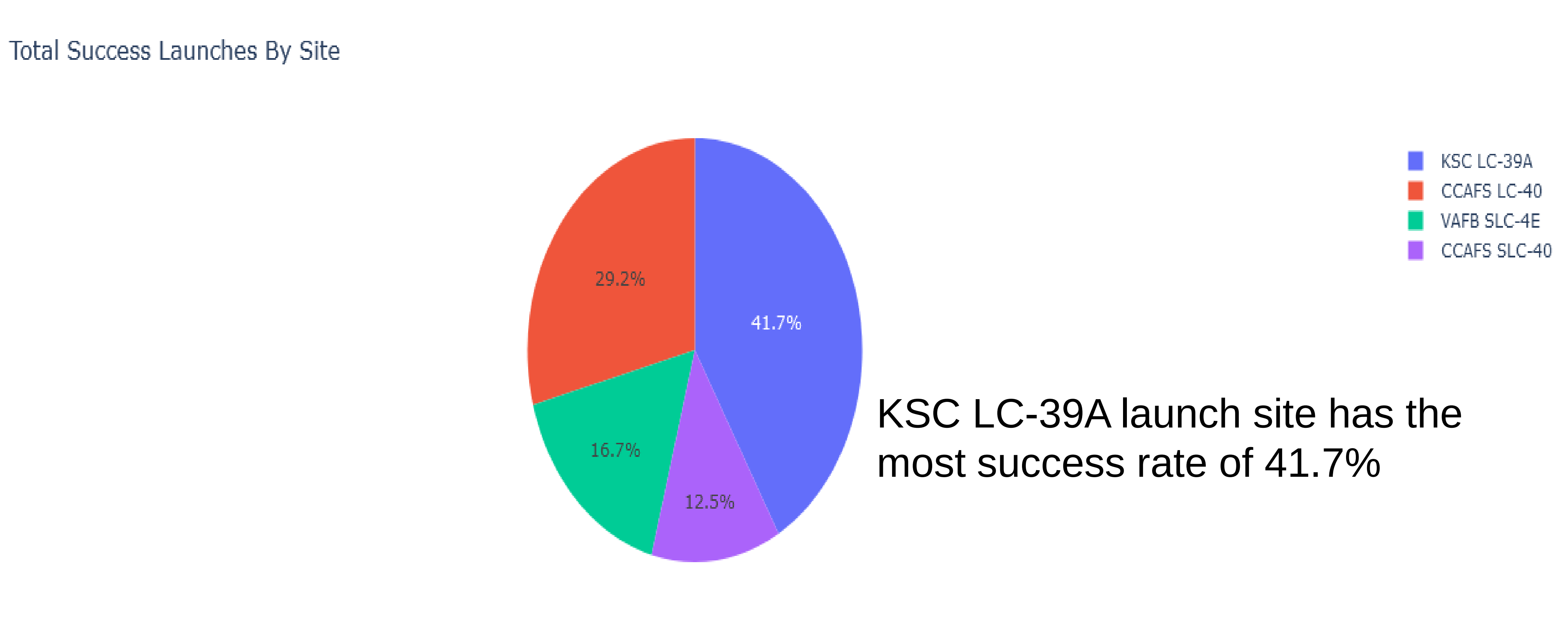


Section 4

Build a Dashboard with Plotly Dash

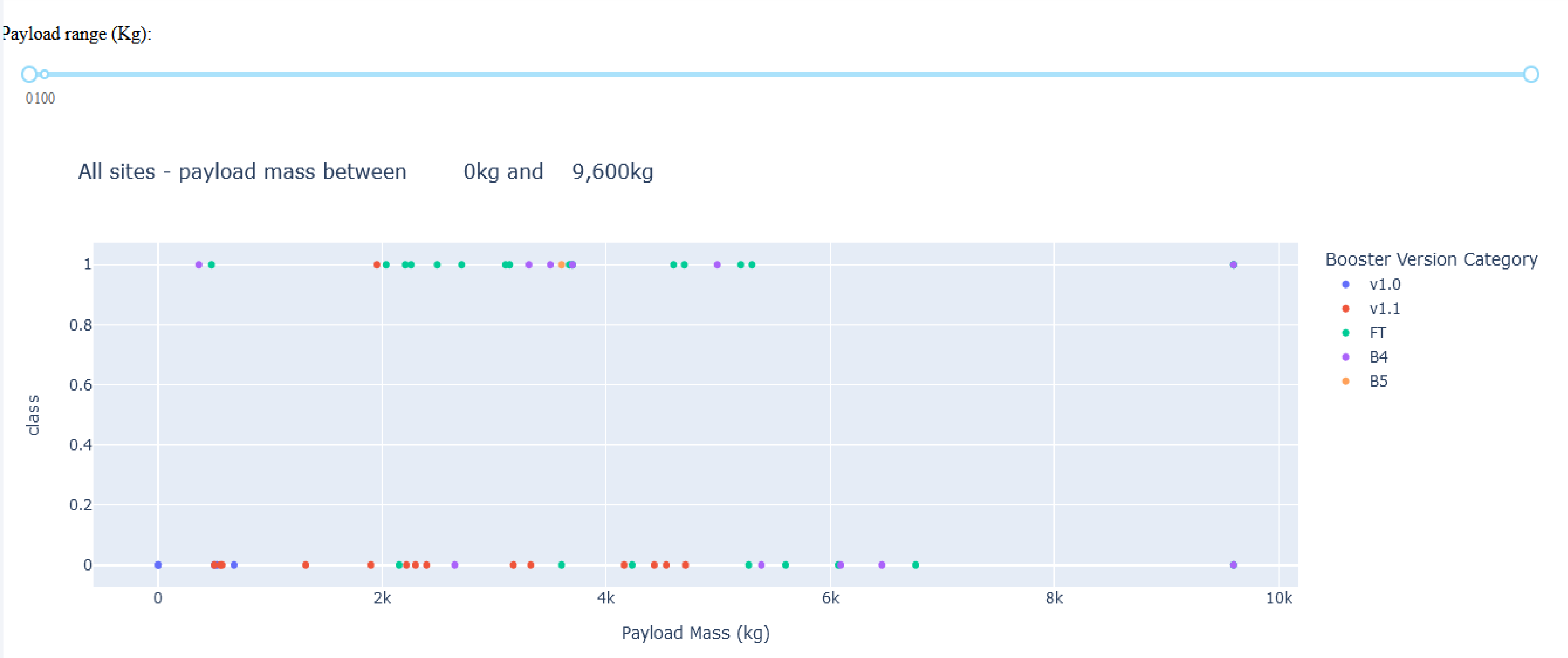


Total Success rate by All Launch sites



KSC LC-39A launch site has the most success rate of 41.7%

All payload sites mass between 0 to 9600 kg

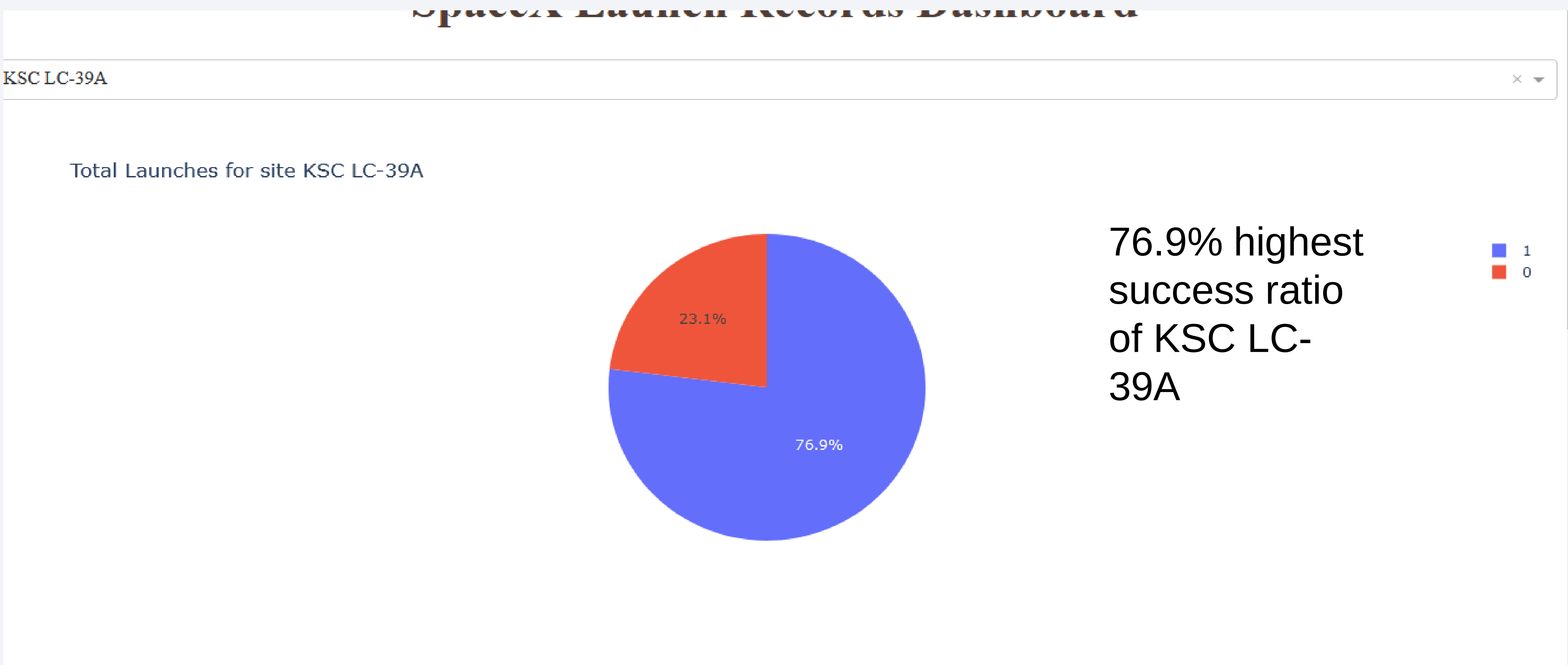


Payload range (Kg):

0100

All sites - payload mass between 0kg and 9,600kg

Highest success ratio as compared to failure



SpaceX Launch Records Dashboard

KSC LC-39A

Total Launches for site KSC LC-39A

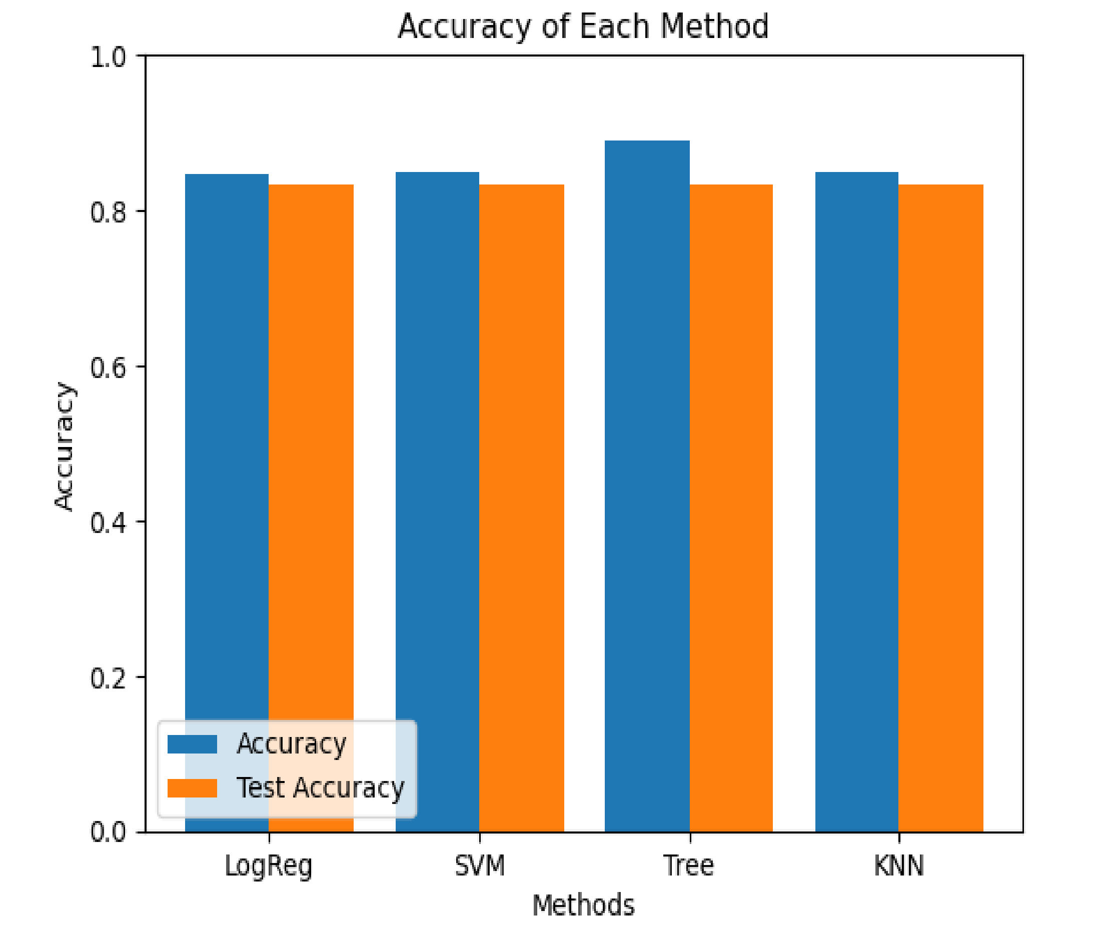
76.9% highest success ratio of KSC LC-39A

Section 5

Predictive Analysis (Classification)

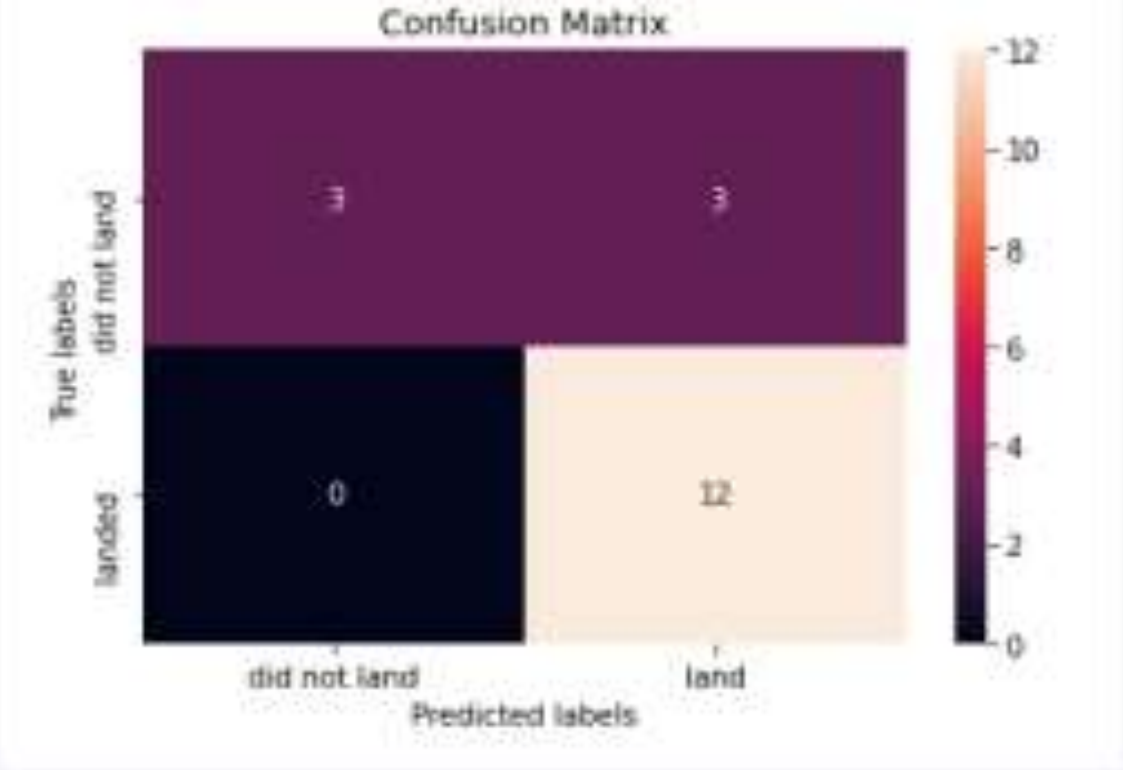
Classification Accuracy

- The highest accuracy from the bar chart can be seen of tree classifier



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.



Conclusions

- The larger the amount of launch from a launch site the more is the success rate
- The payload mass between 0 to 8000 have more success ratio as compared to the mass above then 8000
- The highest success ratio was achieved in 2018 to 2019
- Tree classifier is best for predictive analysis
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate

Appendix

- IBM data science specialization Coursera

Thank you!