DATA SCIENCE CAPSTONE PROJECT

Ziheng Ding

Aug 14, 2021

OUTLINE

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

EXECUTIVE SUMMARY

- This capstone project collected historical launch data and then train predictive model to see if the Falcon 9 first stage will land successfully in the future.
- The trained model could have a good chance* to predict the result successfully based on current collected data set.

*over 83%

INTRODUCTION

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch, which can be used if an alternate company wants to bid against SpaceX for a rocket launch.

What data set will be used to train the model?

What features affect successful rate of launch and landing?

How reliable will the trained model be?

METHODOLOGY

Data collection methodology:

Requested from SpaceX API and scraped from Wikipedia

Perform data wrangling

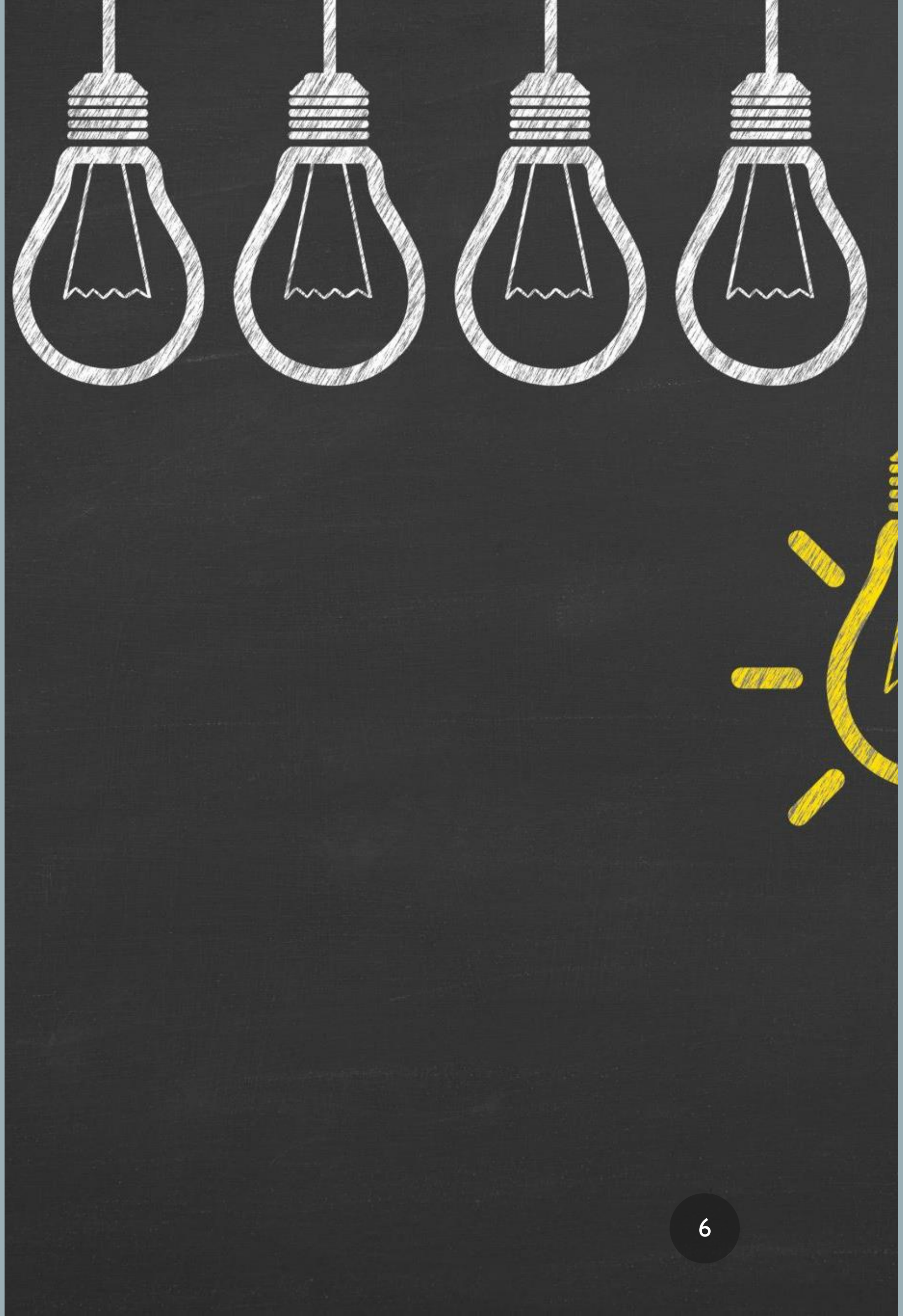
Remove irrelevant, replace missing values, and hot encoding data

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

METHODOLOGY



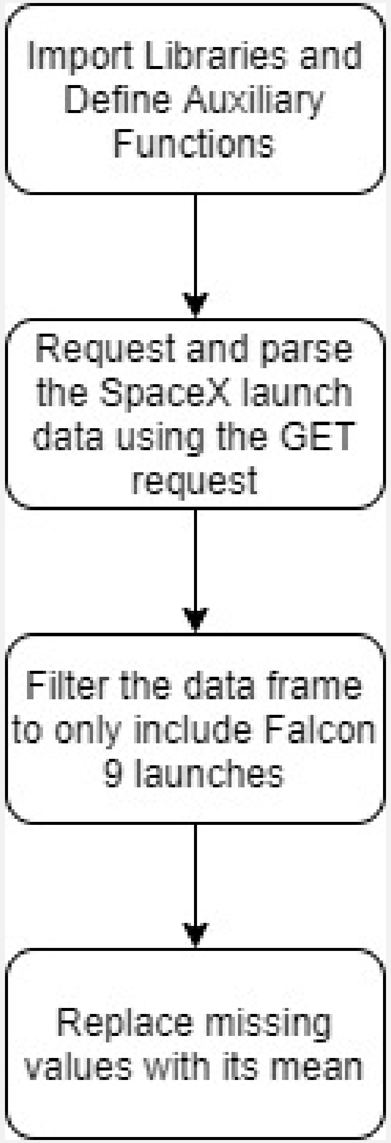
DATA COLLECTION

- Use provided **API** to get all Falcon launches record and then filter out Falcon 9.
  - Request and parse the SpaceX launch data using the GET request.
  - Filter the data frame to only include Falcon 9 launches.
  - Replace missing values with its mean.
- **Web scraping** Falcon 9 and Falcon heavy launches record from Wikipedia
  - Request the Falcon9 Launch Wiki page from its URL by using HTTP GET method.
  - Parse HTML response by using beautiful soup.
  - Extract all column/variable names from the HTML table header.
  - Create a data frame by parsing the launch HTML tables

DATA COLLECTION – SPACEX API

- Request records, parse records, filter records, and replace missing values.

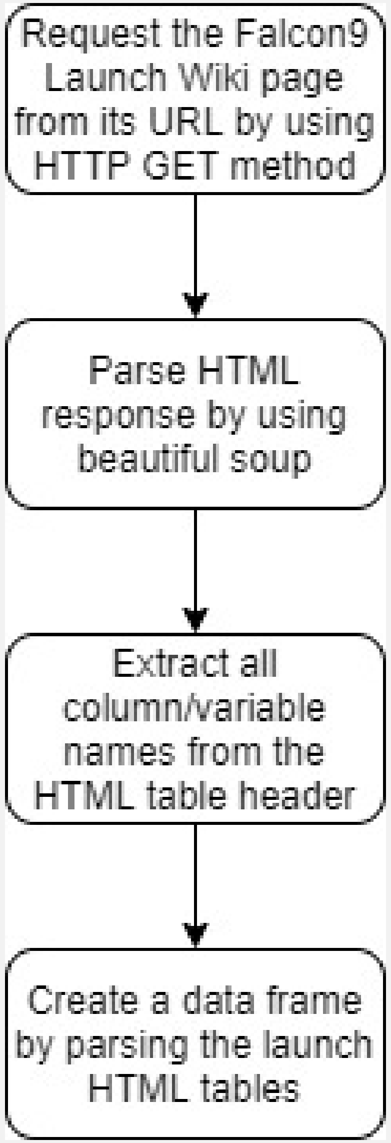
- https://github.com/xiaotied/ibm_ds_project/blob/main/W1-DataCollectionAPILab.ipynb


Import Libraries and Define Auxiliary Functions
Request and parse the SpaceX launch data using the GET request
Filter the data frame to only include Falcon 9 launches
Replace missing values with its mean

DATA COLLECTION – WEB SCRAPING

- Request, parse, extract, and create data frame

- https://github.com/xiaotied/ibm_ds_project/blob/main/W1-WebScrapingLab.ipynb



Request the Falcon9 Launch Wiki page from its URL by using HTTP GET method

Parse HTML response by using beautiful soup

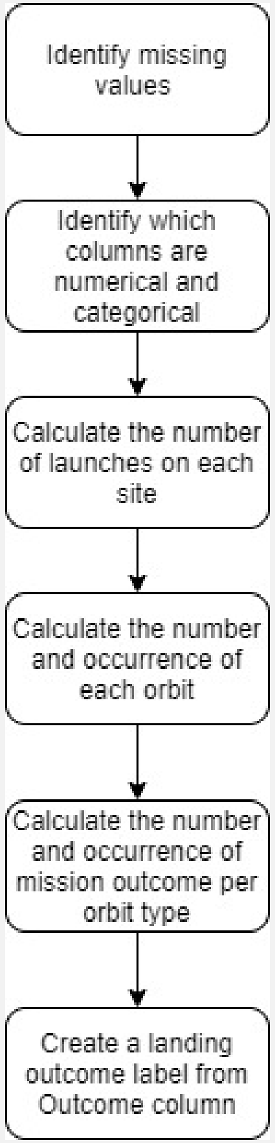
Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

DATA WRANGLING

- How data were processed
  - Identify and calculate the percentage of the missing values in each attribute
  - Identify which columns are numerical and categorical
  - Calculate the number of launches on each site
  - Calculate the number and occurrence of each orbit
  - Calculate the number and occurrence of mission outcome per orbit type
  - Create a landing outcome label from Outcome column

- https://github.com/xiaotied/ibm_ds_project/blob/main/W1-DataWrangling.ipynb


Identify missing values
Identify which columns are numerical and categorical
Calculate the number of launches on each site
Calculate the number and occurrence of each orbit
Calculate the number and occurrence of mission outcome per orbit type
Create a landing outcome label from Outcome column

EDA WITH DATA VISUALIZATION

- In the EDA, I used the following charts:
  - Scatter point chart to visualize the pattern between features
  - Bar plot to visualize the distribution of data points and comparison the values between features.
  - Line chart to visualize the relationship between features.
- https://github.com/xiaotied/ibm_ds_project/blob/main/W2-EDAwithVisualizationLab.ipynb

EDA WITH SQL

- Summarize performed SQL queries using bullet points
  - Display the names of the 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.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
  - Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- https://github.com/xiaotied/ibm_ds_project/blob/main/W2-EDAwithSQLlab.ipynb

BUILD AN INTERACTIVE MAP WITH FOLIUM

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - Added markers to all launch sites on map
    - Visualize the map distribution of launch sites
  - Marked the success/failed launches for each site on the map
    - It visualized how many times success/failed
  - Calculated and shown distances between a launch site to its nearest railway
    - Tried to understand how SPACX chooses launch site.

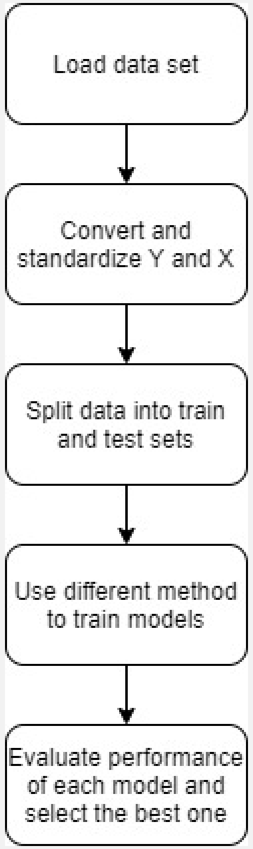
- https://github.com/xiaotied/ibm_ds_project/blob/main/W3-Interactive%20Visual%20Analytics%20with%20Folium.ipynb

BUILD A DASHBOARD WITH PLOTLY DASH

- A drop-down list has been added to let users select each launch sites and able to interact with dashboard.
- A pie chart has been added to show the number of success vs unsuccess for each and all location.
- A scatter chart has been added to display the correlation between payload mass and success rate for each and all sites.
- A slider bar for payload mass range, which allows users to select payload mass range.

- https://github.com/xiaotied/ibm_ds_project/blob/main/W3%20-%20dash/W3-dash.py

PREDICTIVE ANALYSIS (CLASSIFICATION)



Load data set

Convert and standardize Y and X

Split data into train and test sets

Use different method to train models

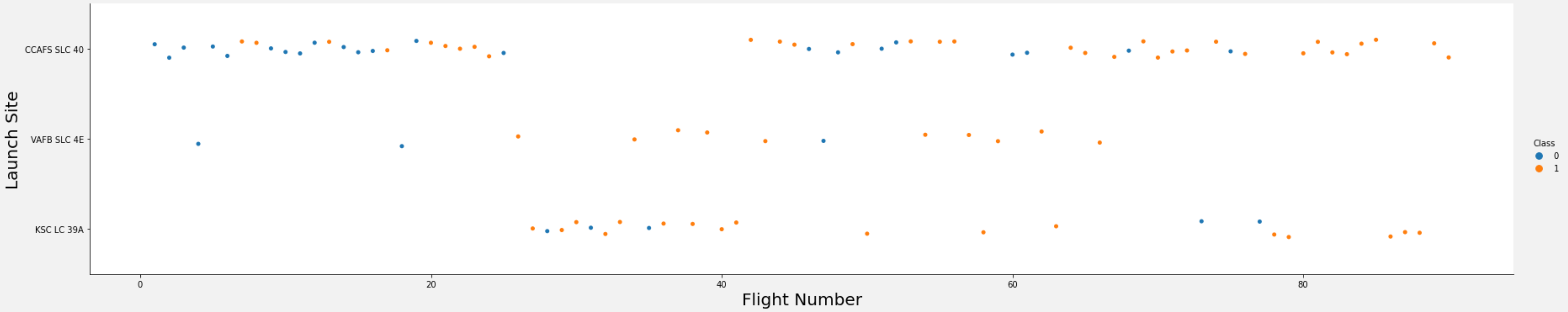
Evaluate performance of each model and select the best one

Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

RESULTS

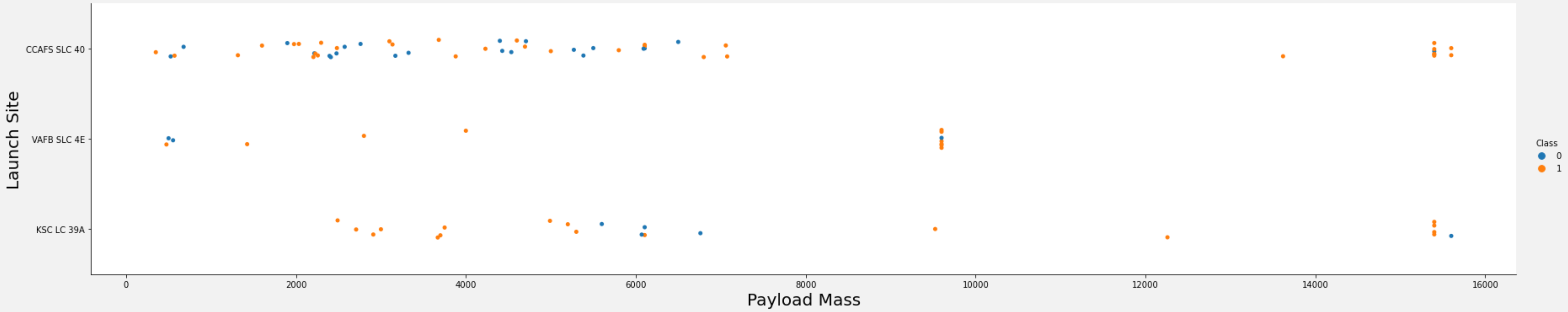
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

EDA WITH VISUALIZATION



FLIGHT NUMBER VS. LAUNCH SITE

- The most used location is CCAFS SLC 40
- The Flight number after 80 are all succussed

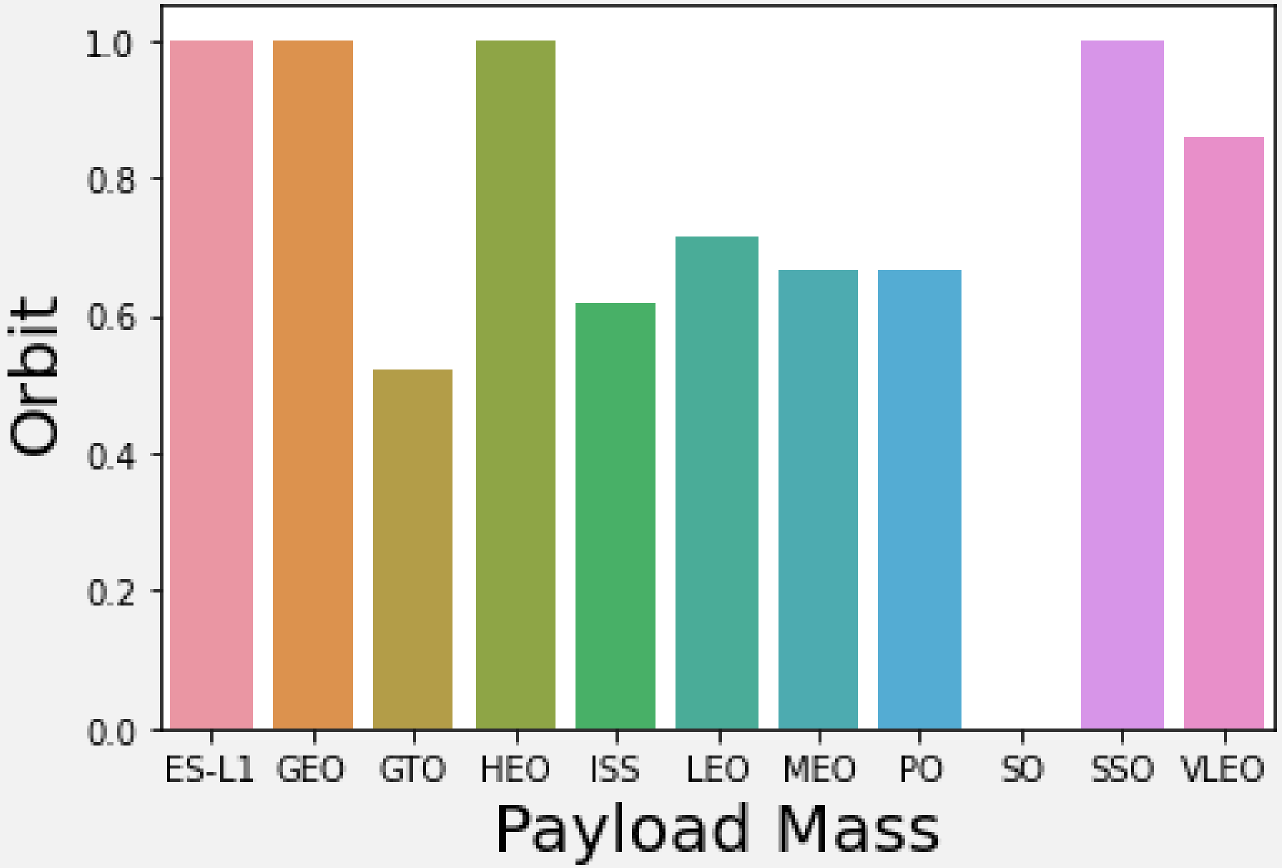


PAYLOAD VS. LAUNCH SITE

- CCAF SLC 40 is most use to launch payload mass under 8000
- VAFB SLC $E is most use to launch payload mass around 10000

SUCCESS RATE VS. ORBIT TYPE

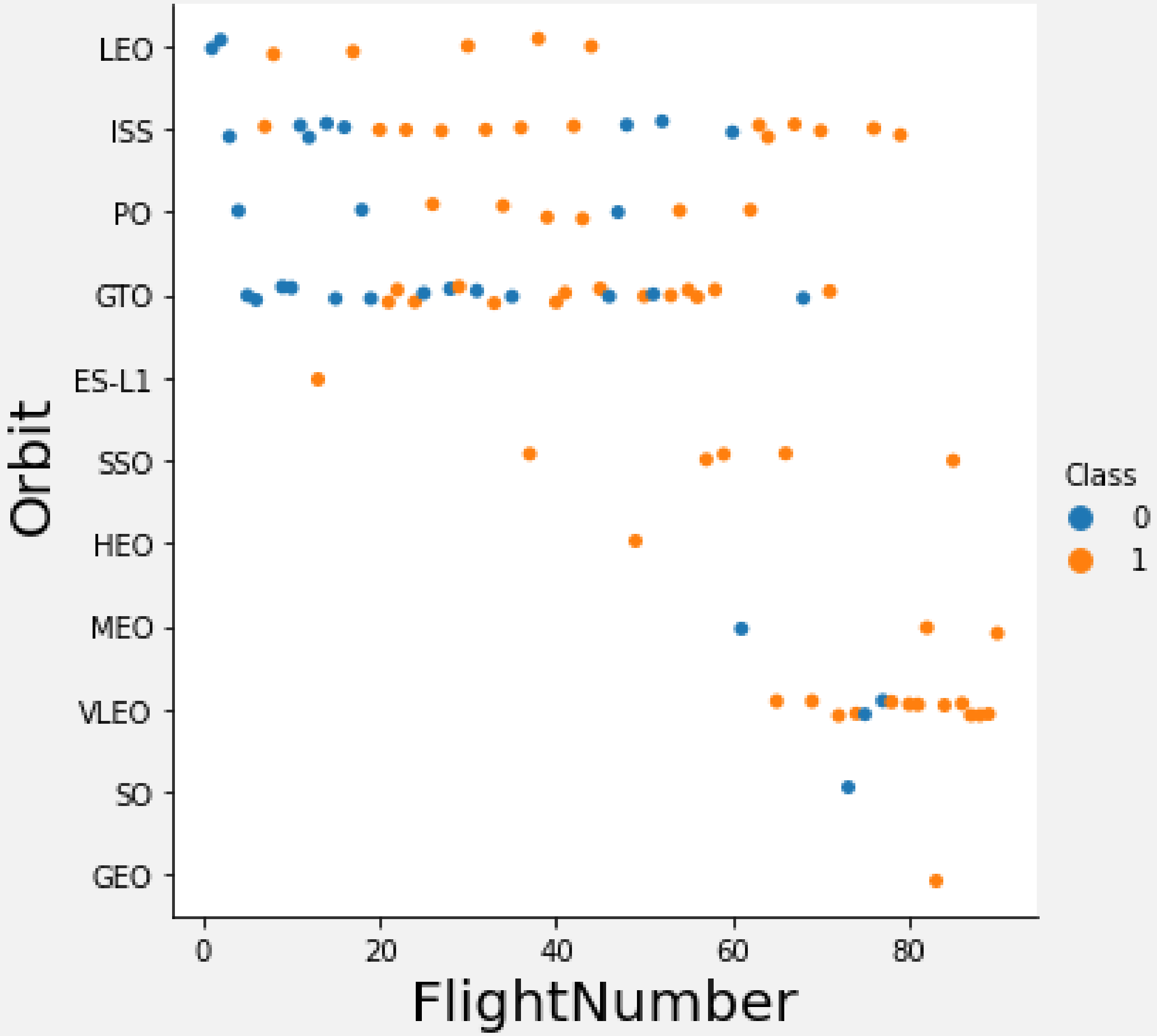
- ES-L1, GEO, HEO, and SSO have the highest success rate, 100%
- SO has 0% success rate



FLIGHT NUMBER VS. ORBIT TYPE

- In the early launches, most are above GTO

- Recent launches most are VLEO

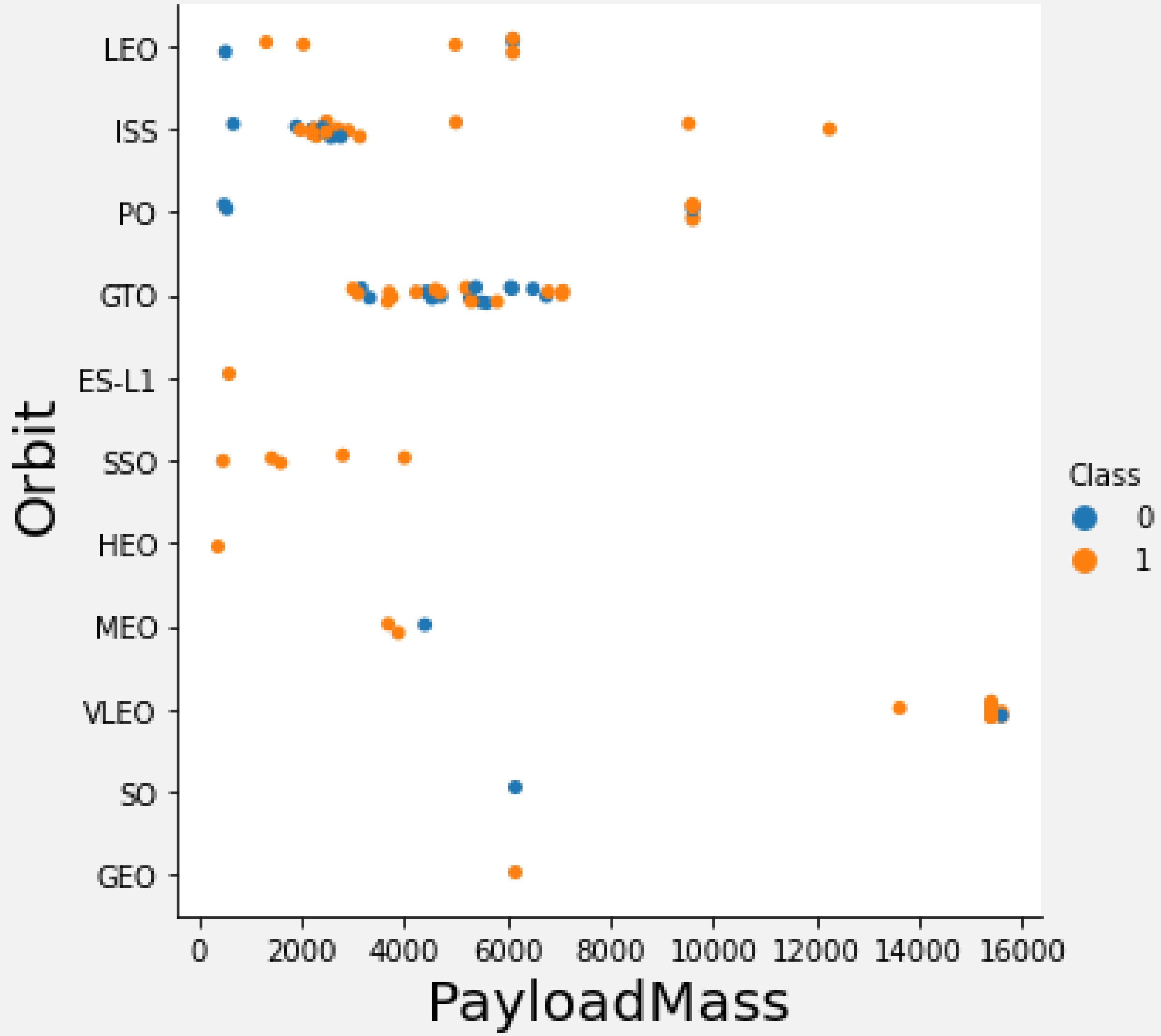


PAYLOAD VS. ORBIT TYPE

- Heavy payload are launch to VLEO

- Normal payload are launch to GTO

- Light payload are launch to ISS

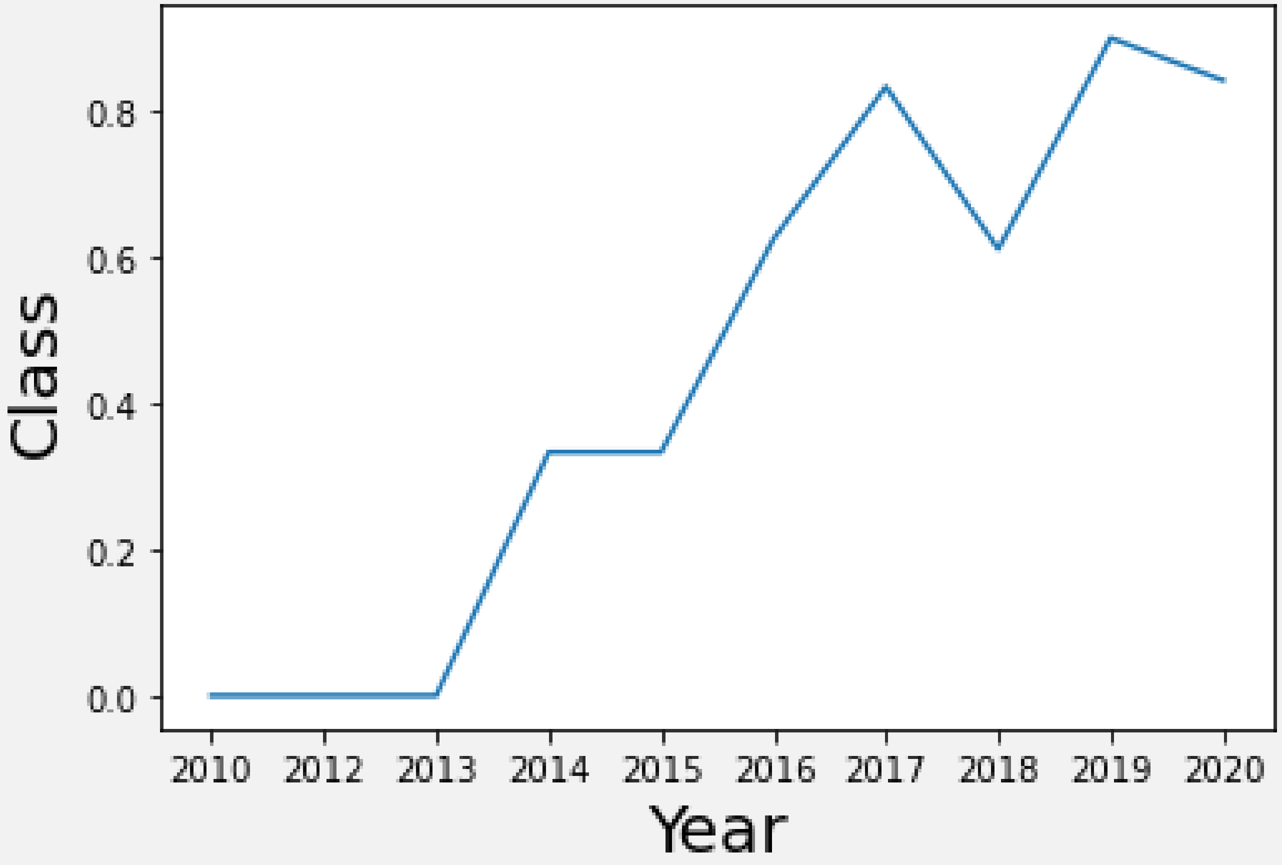


LAUNCH SUCCESS YEARLY TREND

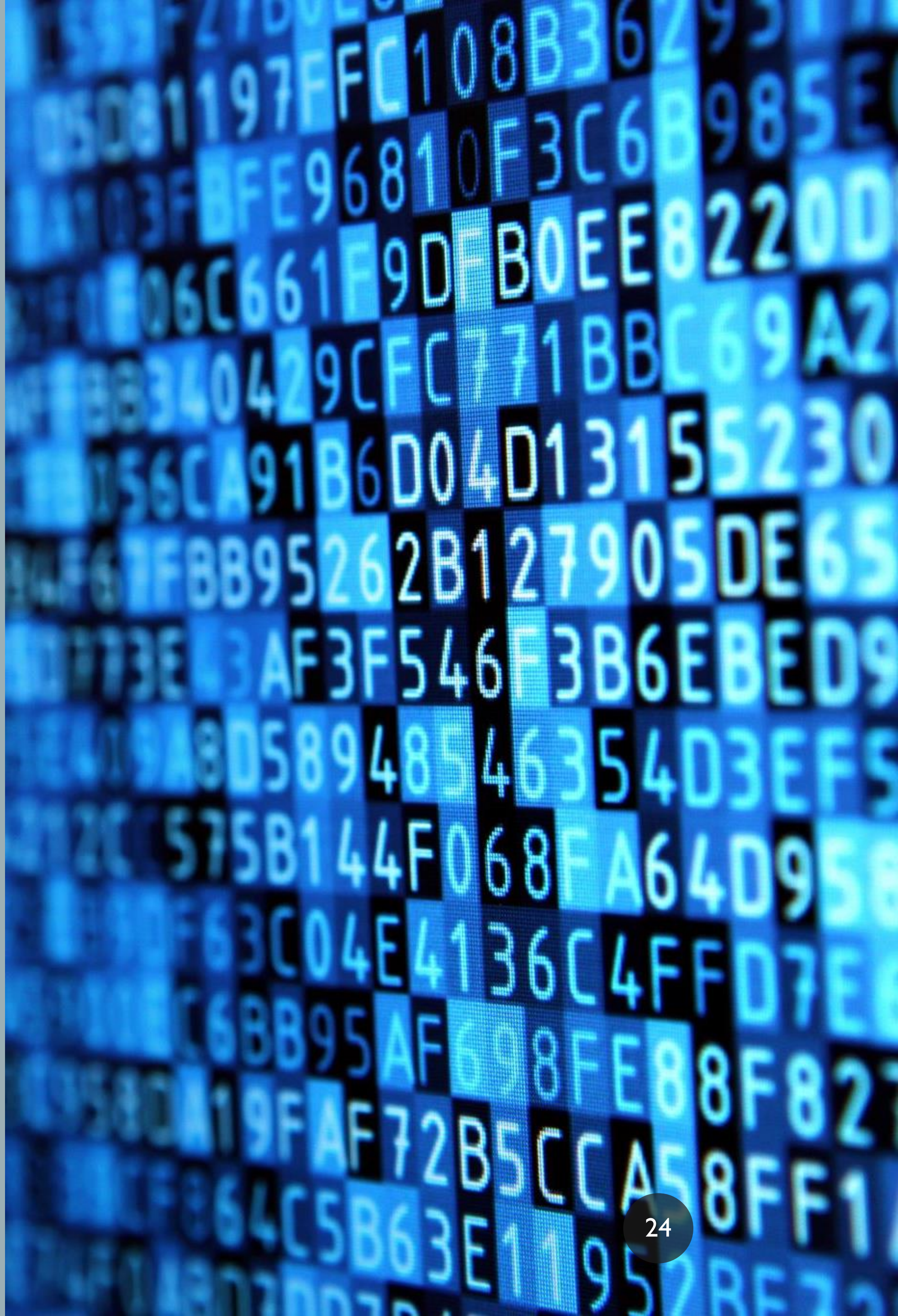
- Success rate increased from 2013 to 2016

- There is a decrease in 2017 to 2018

- It continue to increase after 2018

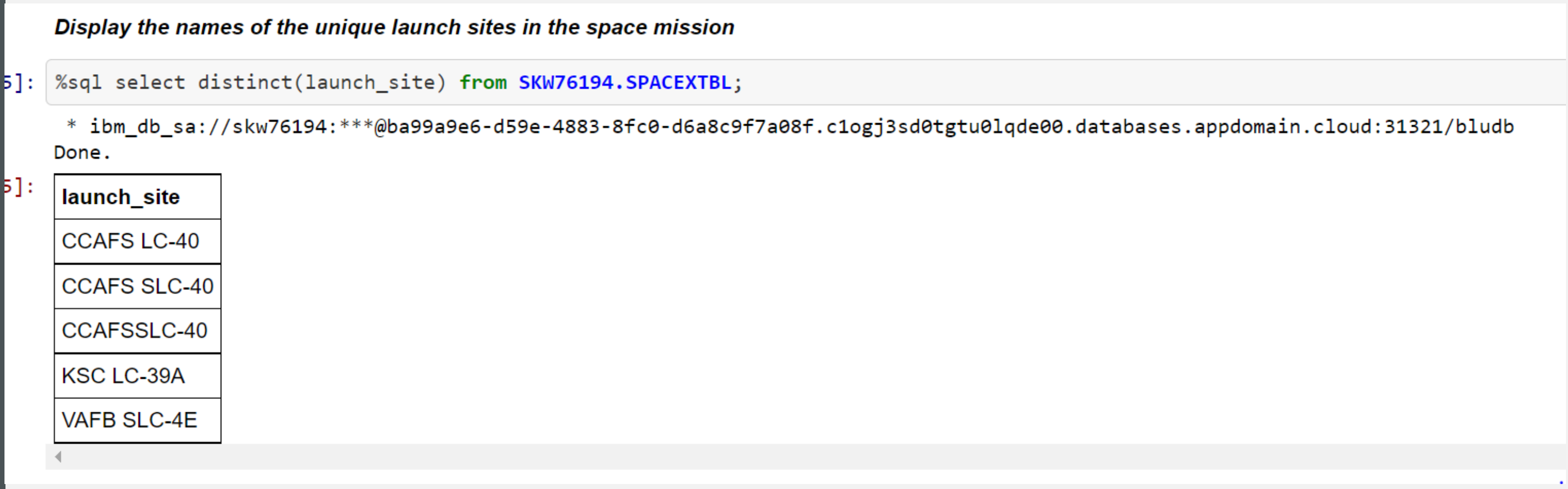


EDA WITH SQL



ALL LAUNCH SITE NAMES

- There are five unique launch sites.
- CCAFS LC-40, CCAFS SLC 40, KSC LC 39A, VAFB SLC 4E



Display the names of the unique launch sites in the space mission

```
5]: %sql select distinct(launch_site) from SKW76194.SPACEXTBL;
```

```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

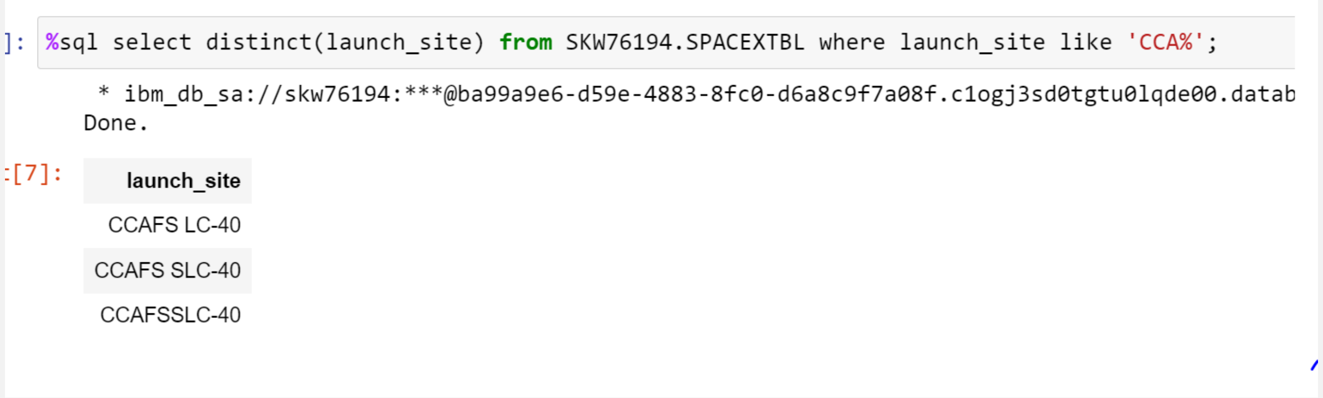
5]:

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

LAUNCH SITE NAMES BEGIN WITH `CCA`

- There are three launch sites begin with `CCA`

- CCAFS LC-40, CCAFS SLC 40, CCAFSSLC 40



```
]: %sql select distinct(launch_site) from SKW76194.SPACEXTBL where launch_site like 'CCA%';
```

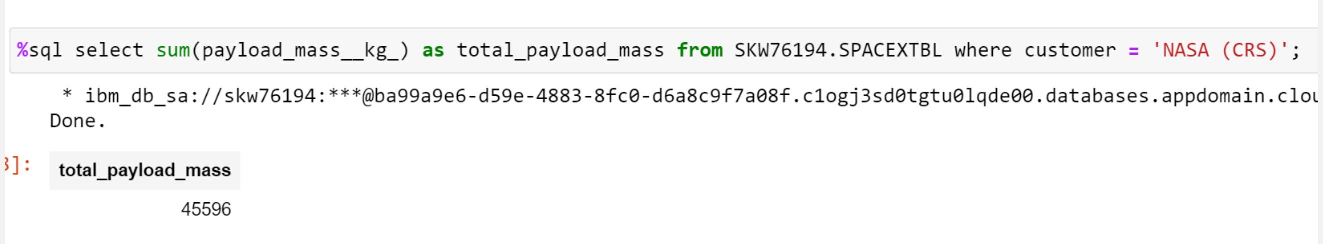
```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.datab
Done.
```

:[7]:

launch_site
CCAFS LC-40
CCAFS SLC-40
CCAFSSLC-40

TOTAL PAYLOAD MASS

- The total payload carried by boosters from NASA is 45596kg



```
%sql select sum(payload_mass__kg_) as total_payload_mass from SKW76194.SPACEXTBL where customer = 'NASA (CRS)';
```

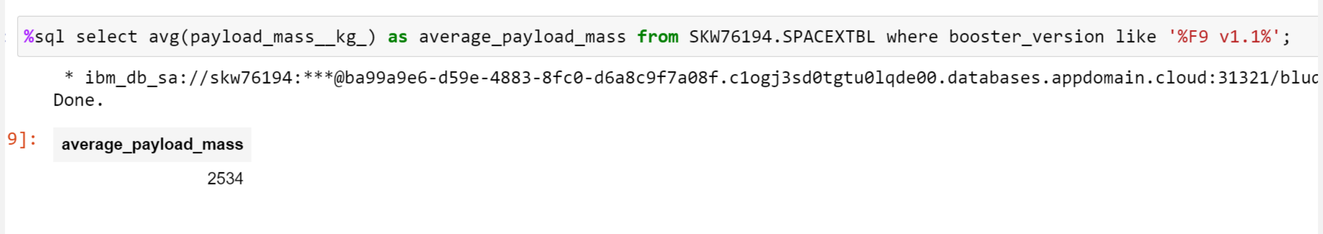
```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
Done.
```

3]:

total_payload_mass
45596

AVERAGE PAYLOAD MASS BY F9 V1.1

- The average payload mass carried by booster version F9 v1.1 is 2534 kg.



```
%sql select avg(payload_mass__kg_) as average_payload_mass from SKW76194.SPACEXTBL where booster_version like '%F9 v1.1%';
```

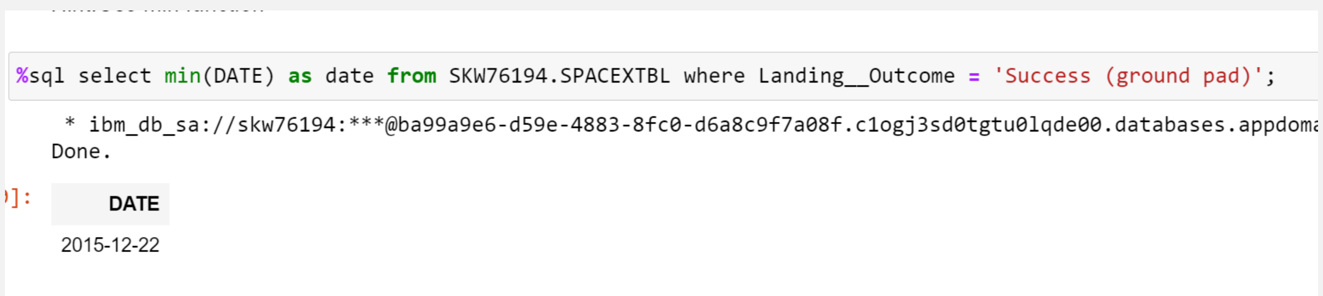
```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bluc
Done.
```

9]:

average_payload_mass
2534

FIRST SUCCESSFUL GROUND LANDING DATE

- Find the date when the first successful landing outcome in ground pad is 2015-12-22



```
%sql select min(DATE) as date from SKW76194.SPACEXTBL where Landing__Outcome = 'Success (ground pad)';
```

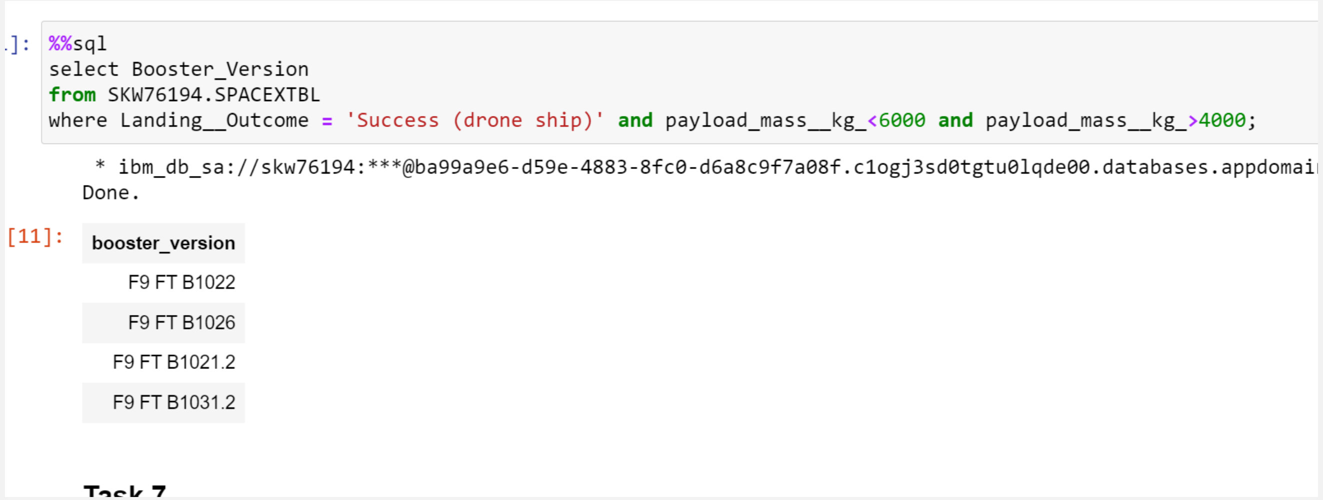
```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdoma
Done.
```

]:

DATE
2015-12-22

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

- There are 4 boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000, which list on the right side.



```
.]: %%sql
    select Booster_Version
    from SKW76194.SPACEXTBL
    where Landing__Outcome = 'Success (drone ship)' and payload_mass__kg_<6000 and payload_mass__kg_>4000;
```

```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomai
Done.
```

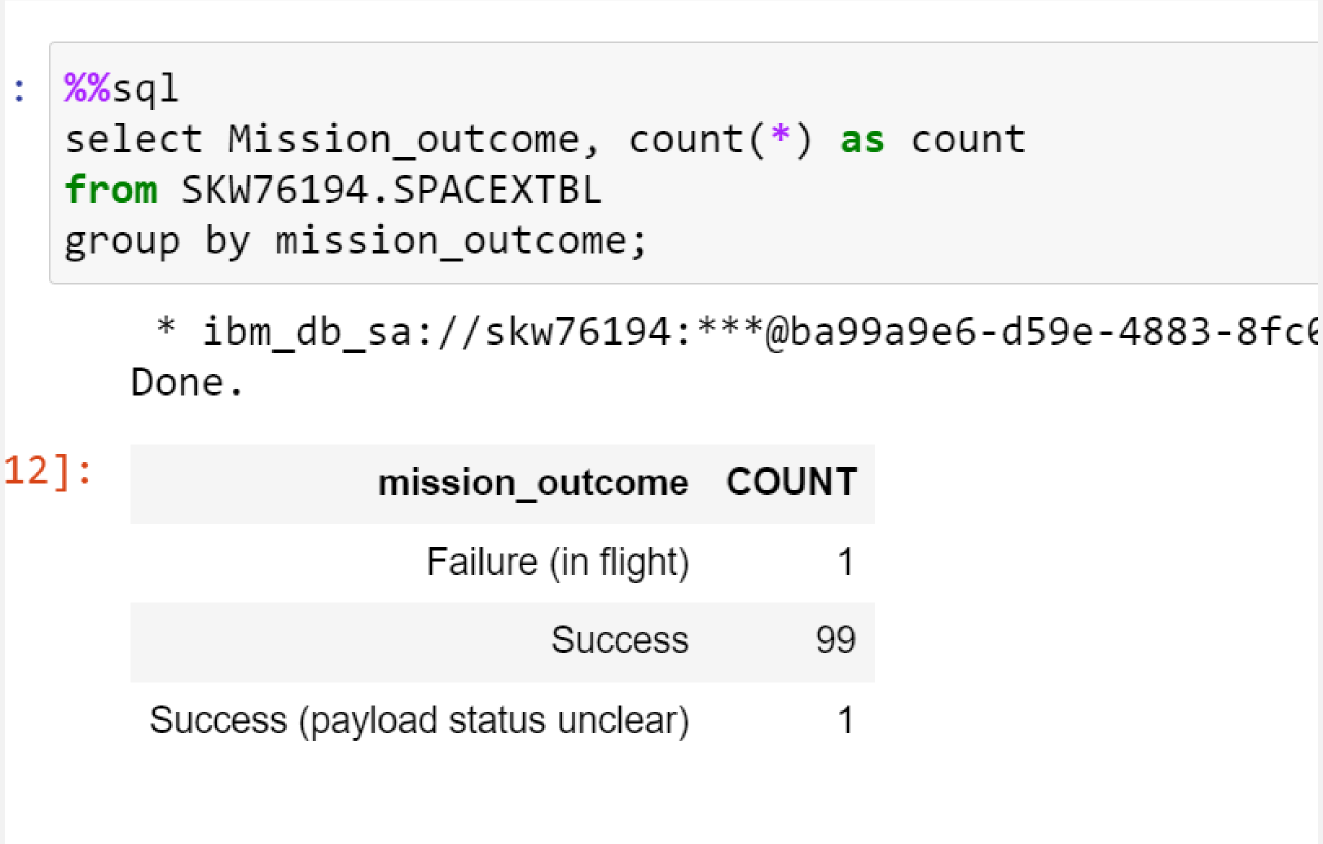
[11]:

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

Task 7

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

- There are 100 successful outcomes and 1 failure mission outcome.



```
%%sql
select Mission_outcome, count(*) as count
from SKW76194.SPACEXTBL
group by mission_outcome;
```

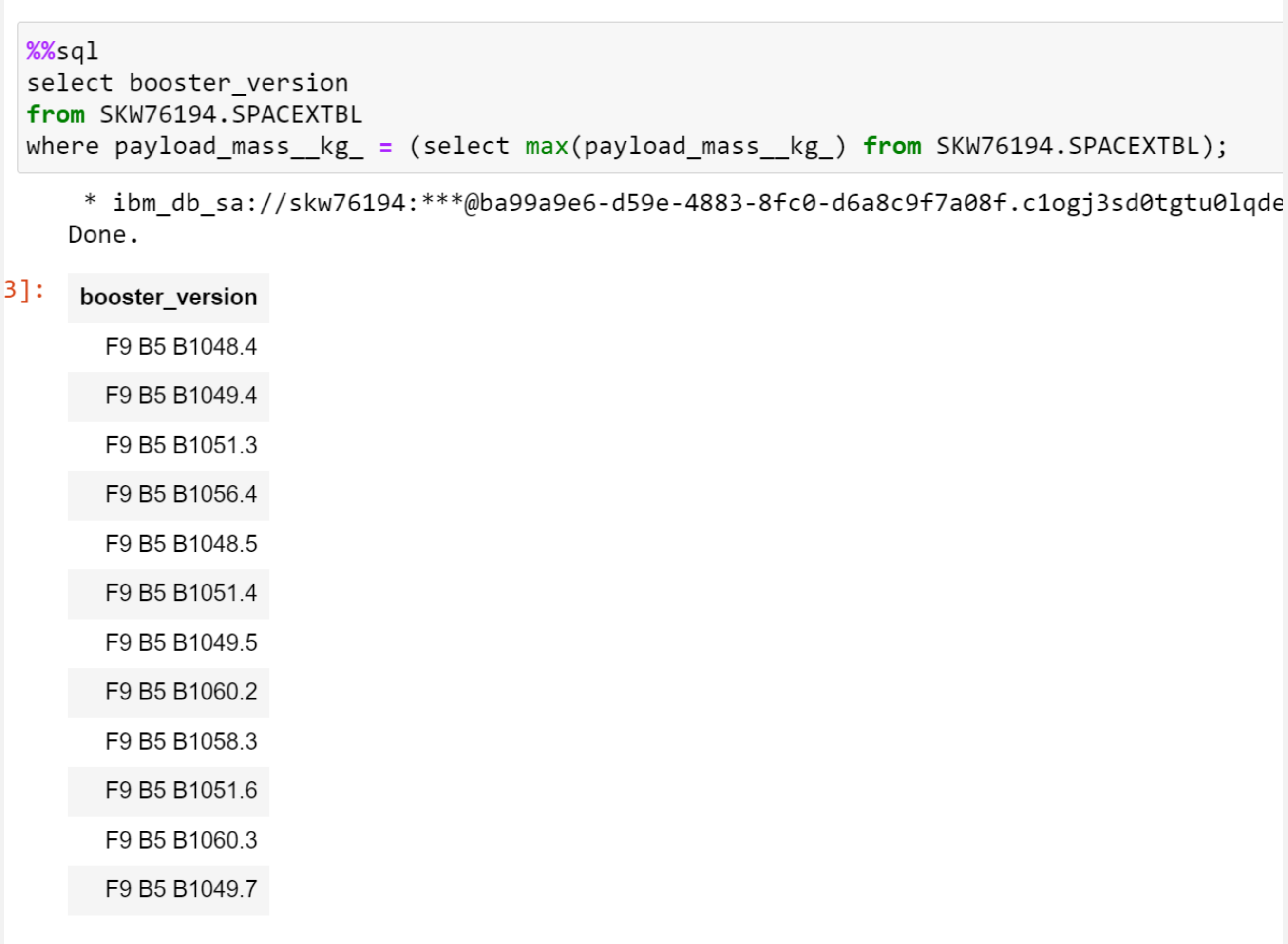
```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc
Done.
```

12]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

BOOSTERS CARRIED MAXIMUM PAYLOAD

- There are 12 booster which have carried the maximum payload mass



```
%%sql
select booster_version
from SKW76194.SPACEXTBL
where payload_mass__kg_ = (select max(payload_mass__kg_) from SKW76194.SPACEXTBL);
```

```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde
Done.
```

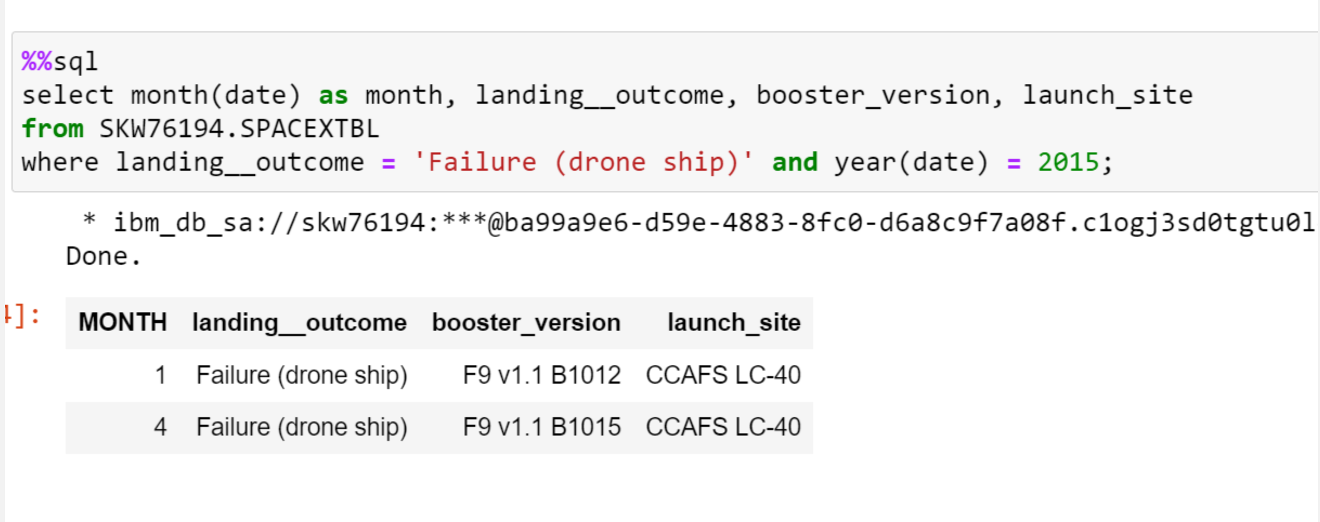
3]:

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

2015 LAUNCH RECORDS

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

- There are two records for 2015



```
%%sql
select month(date) as month, landing__outcome, booster_version, launch_site
from SKW76194.SPACEXTBL
where landing__outcome = 'Failure (drone ship)' and year(date) = 2015;
```

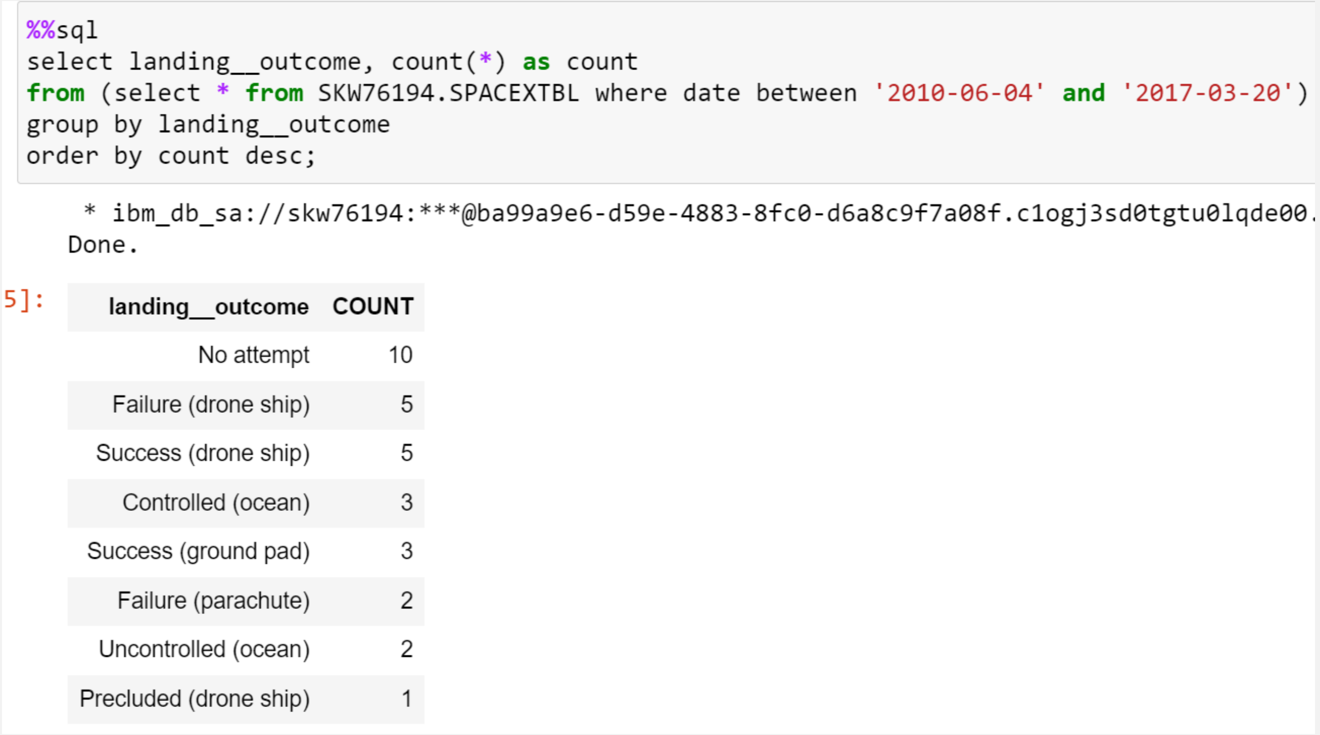
```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0l
Done.
```

4]:

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

RANK SUCCESS COUNT BETWEEN 2010-06-04 AND 2017-03-20

- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.



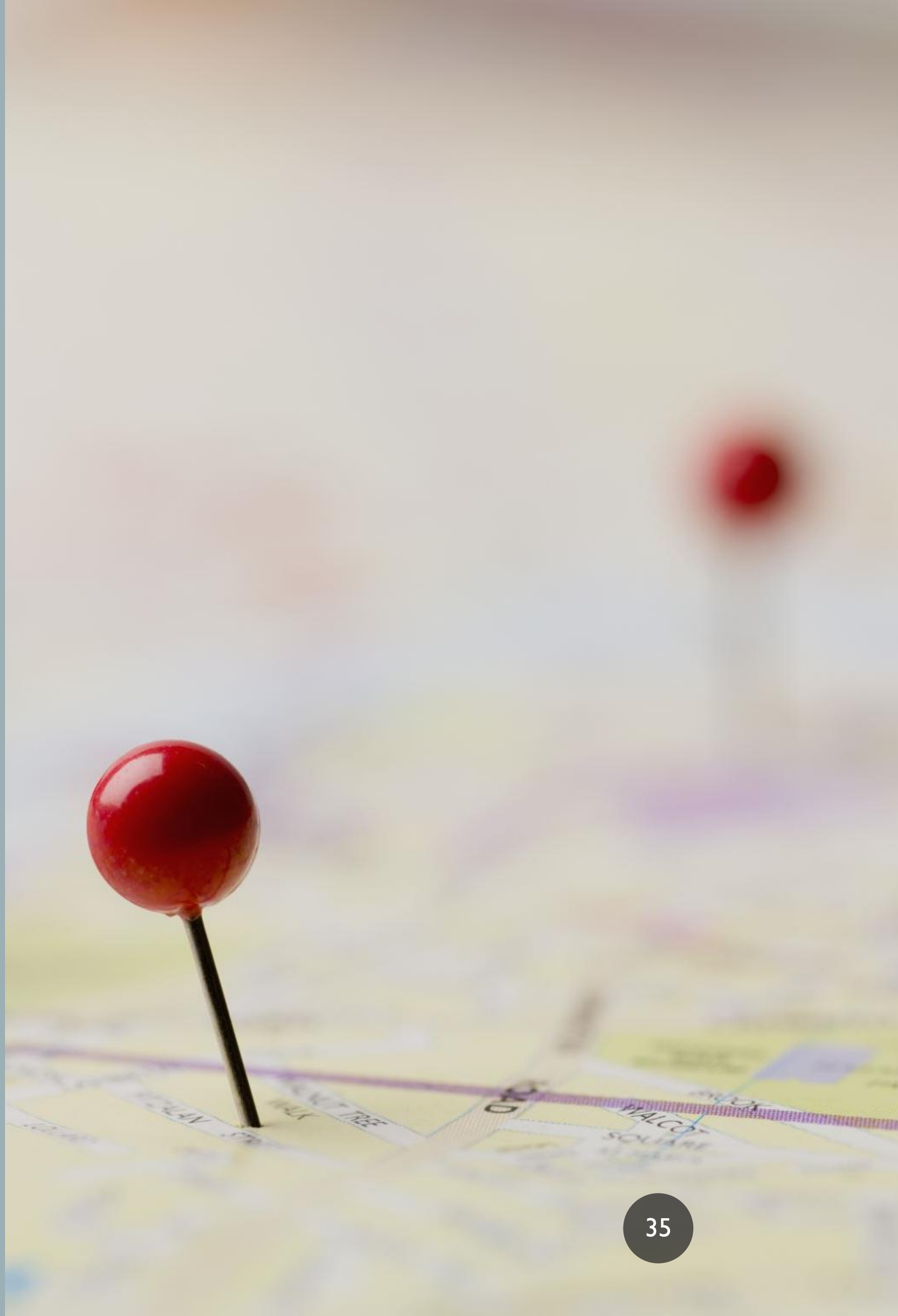
```
%%sql
select landing__outcome, count(*) as count
from (select * from SKW76194.SPACEXTBL where date between '2010-06-04' and '2017-03-20')
group by landing__outcome
order by count desc;
```

```
 * ibm_db_sa://skw76194:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.
Done.
```

5]:

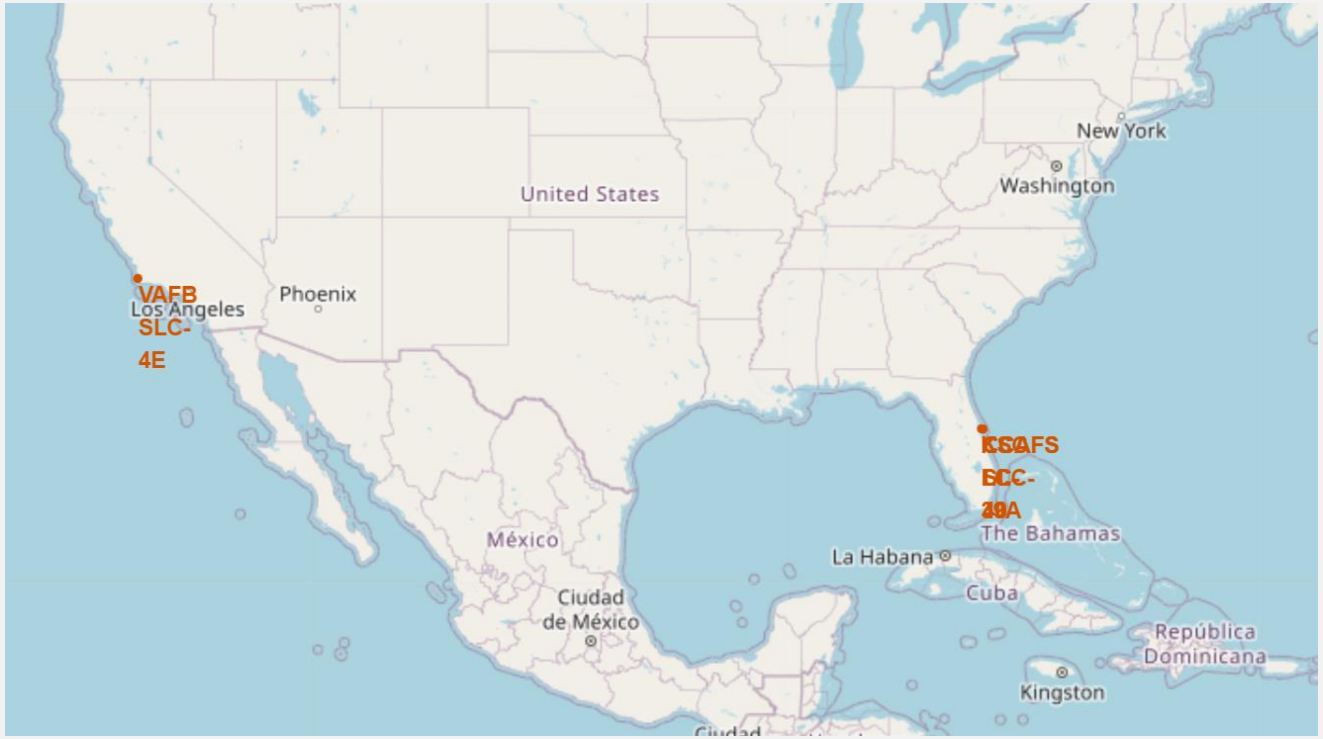
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

INTERACTIVE MAP WITH FOLIUM



LAUNCH SITES' LOCATION

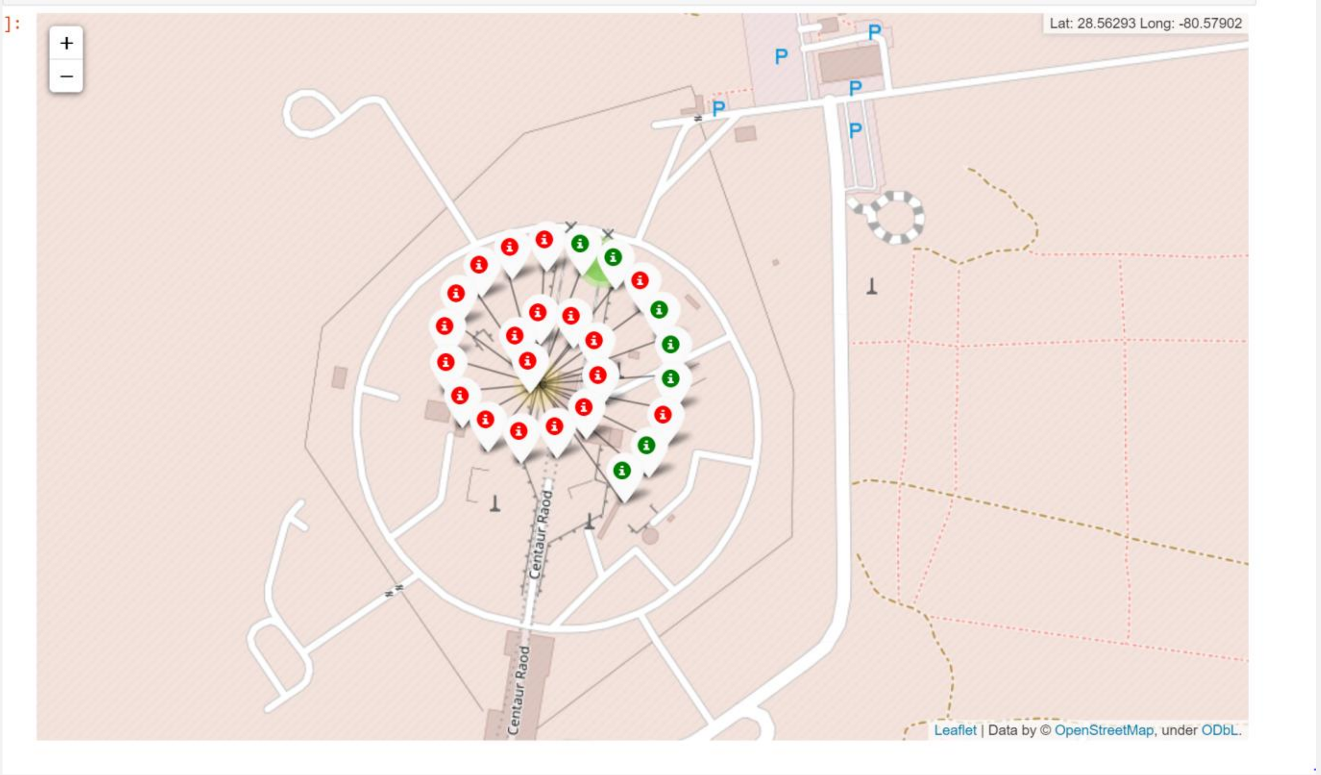
- The two places have similar latitude.
- Two locations are close to equator line
- They are close to coastline



SUCCESS RATE FOR CCAFS LC 40

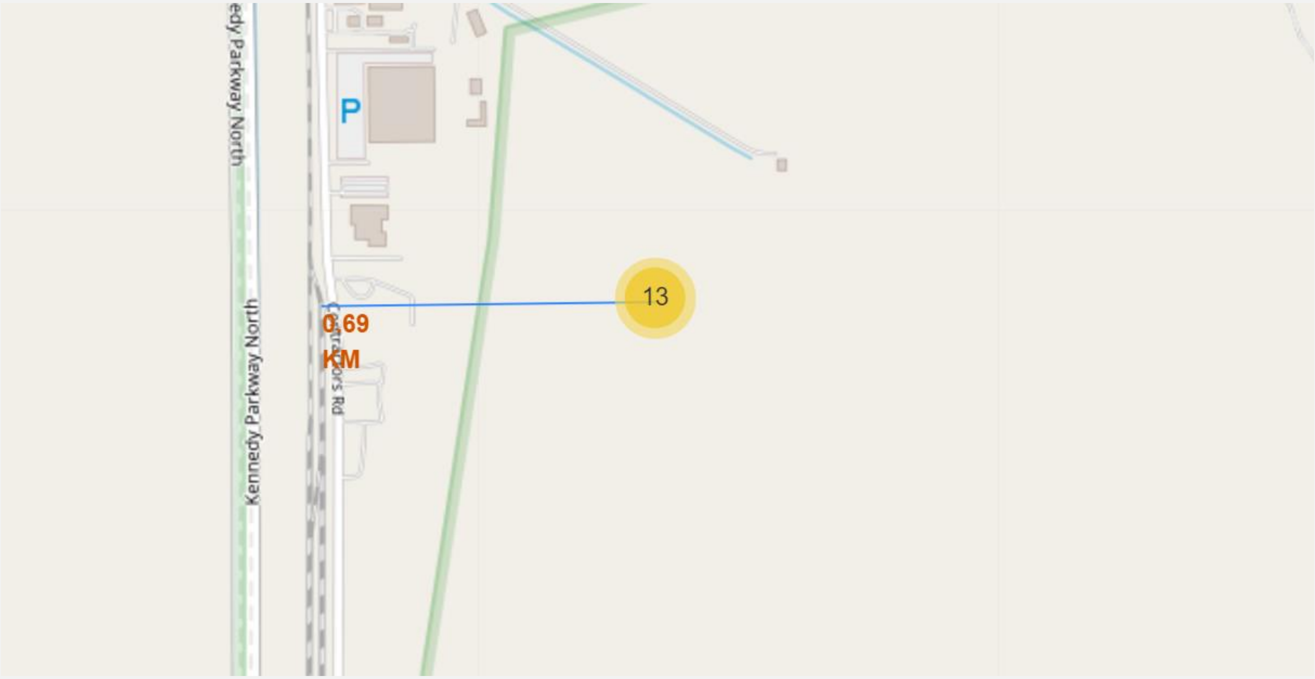
- The map shows all success and unsuccess outcomes for the site CCAFS LC 40

- This location has low success rate

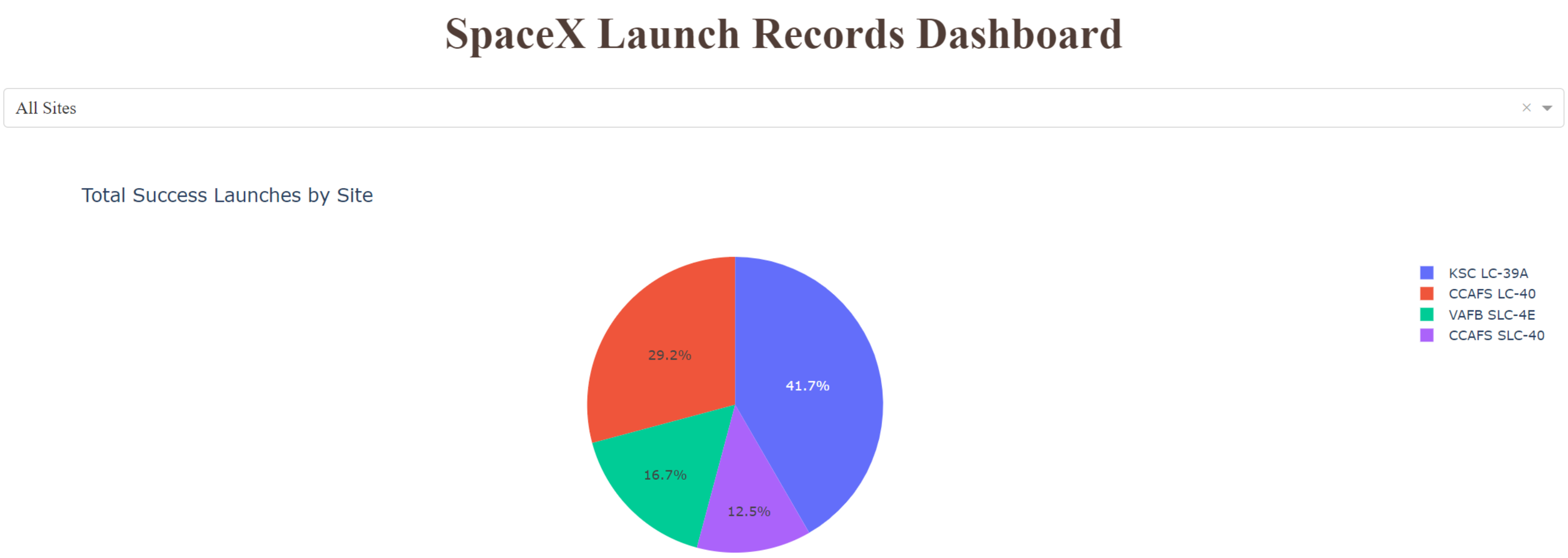

]:
+
−
Lat: 28.56293 Long: -80.57902
P
P
P
P
P
Centaur Raod
Centaur Raod
Leaflet | Data by © OpenStreetMap, under ODbL.

CLOSEST RAILWAY TO KSC LC-39A

- The map shows the closest railway to KSC LC-39A is about 0.69 KM



BUILD A DASHBOARD WITH PLOTLY DASH

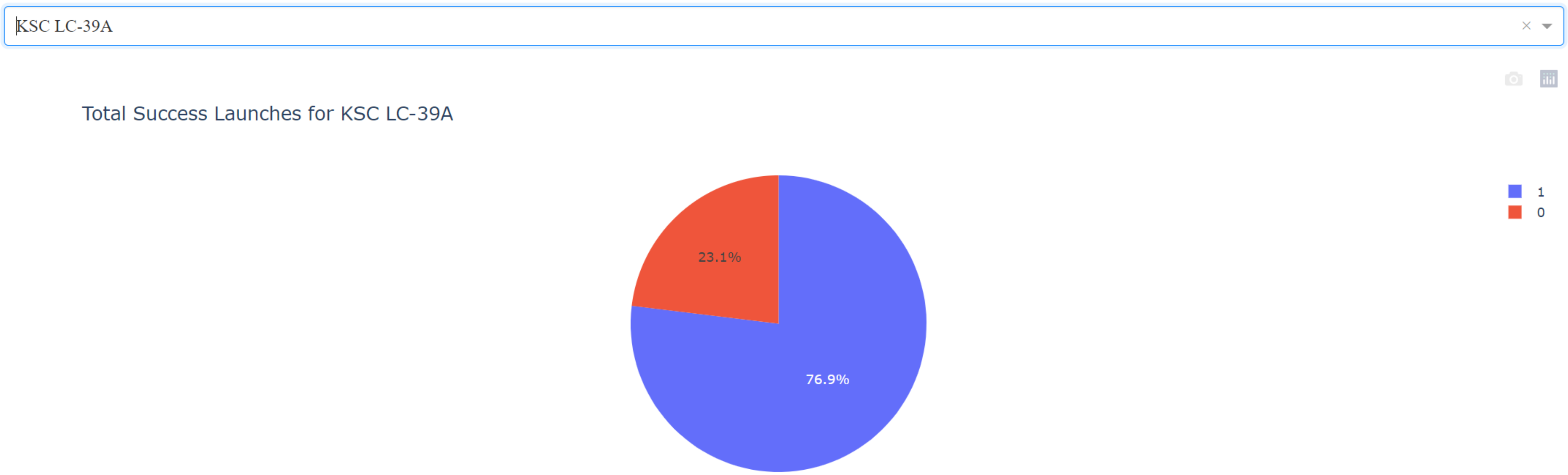


SpaceX Launch Records Dashboard

All Sites × ▾

Total Success Launches by Site

SUCCESS COUNT FOR ALL SITES

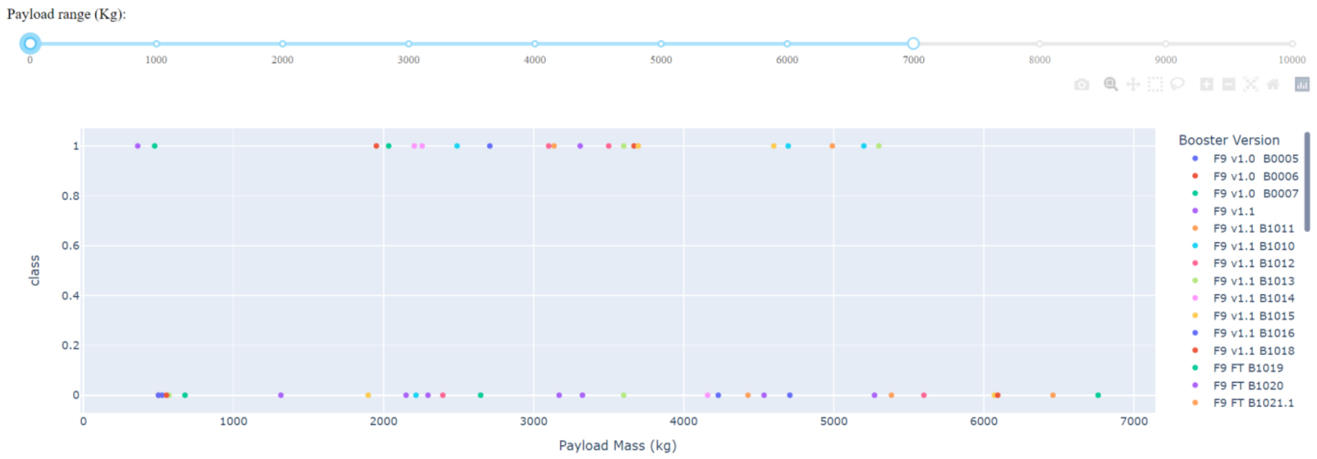


KSC LC-39A

THE LAUNCH SITE WITH HIGHEST SUCCESS RATIO

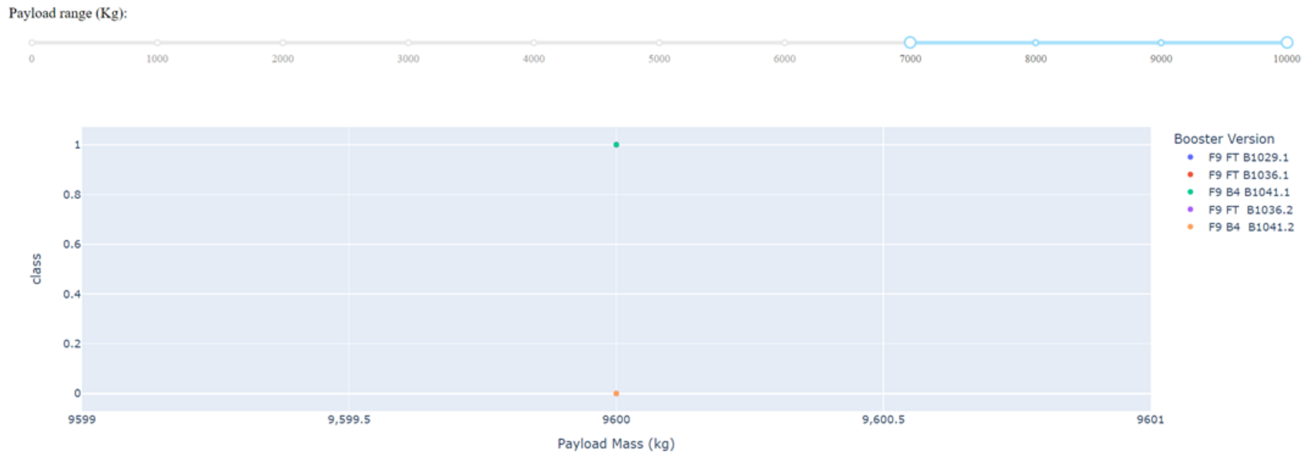
LAUNCH OUTCOME WITH DIFFERENT PAYLOAD RANGES

Payload range from 0 to 7000



LAUNCH OUTCOME WITH DIFFERENT PAYLOAD RANGES

Payload range from 7000 to 10000



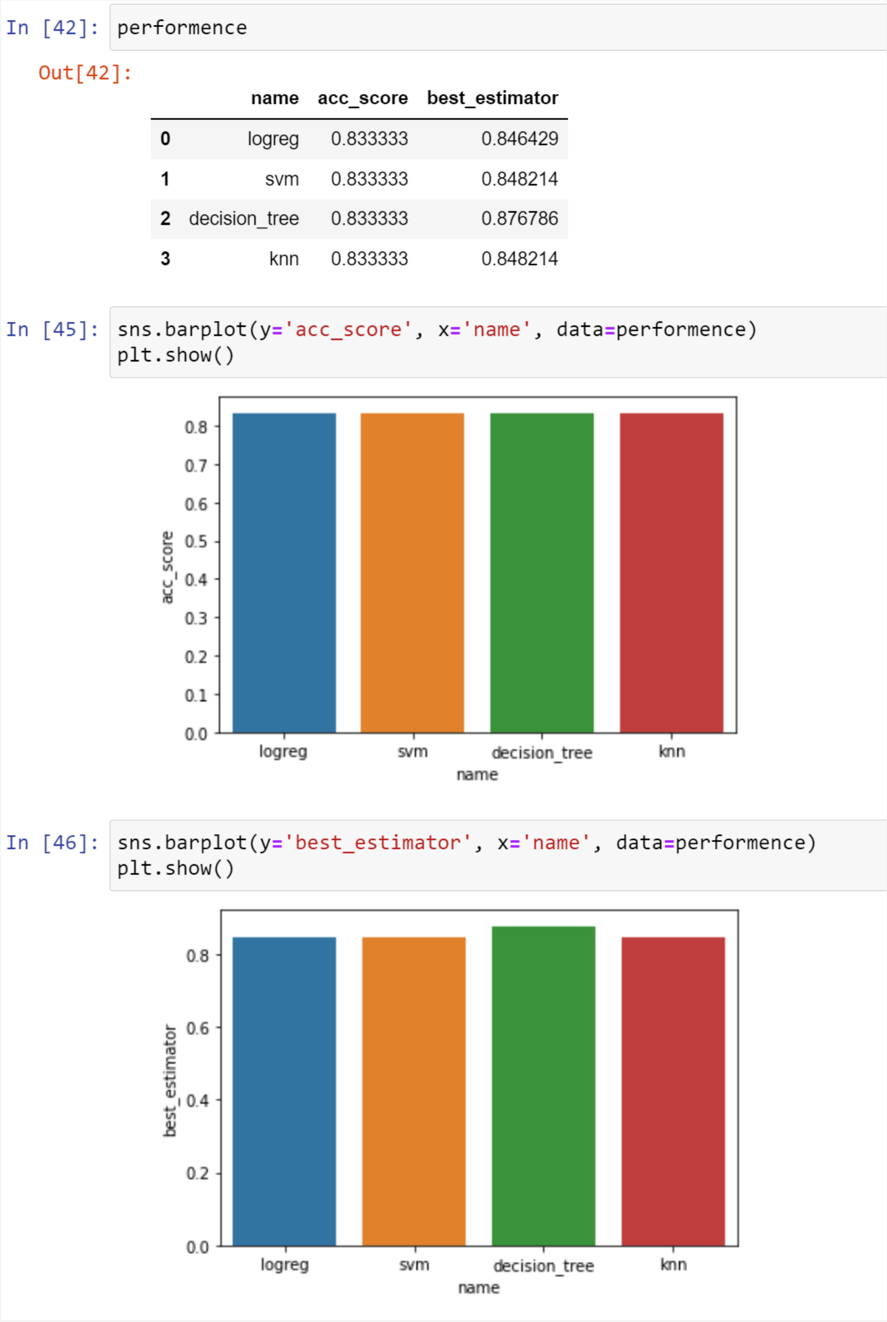
Payload range (Kg):

PREDICTIVE ANALYSIS (CLASSIFICATION)

CLASSIFICATION ACCURACY

- Visualize all the built model accuracy for all built models, in a bar chart

- Decision tree has better performance on best_score__



```
In [42]: performence
```

Out[42]:

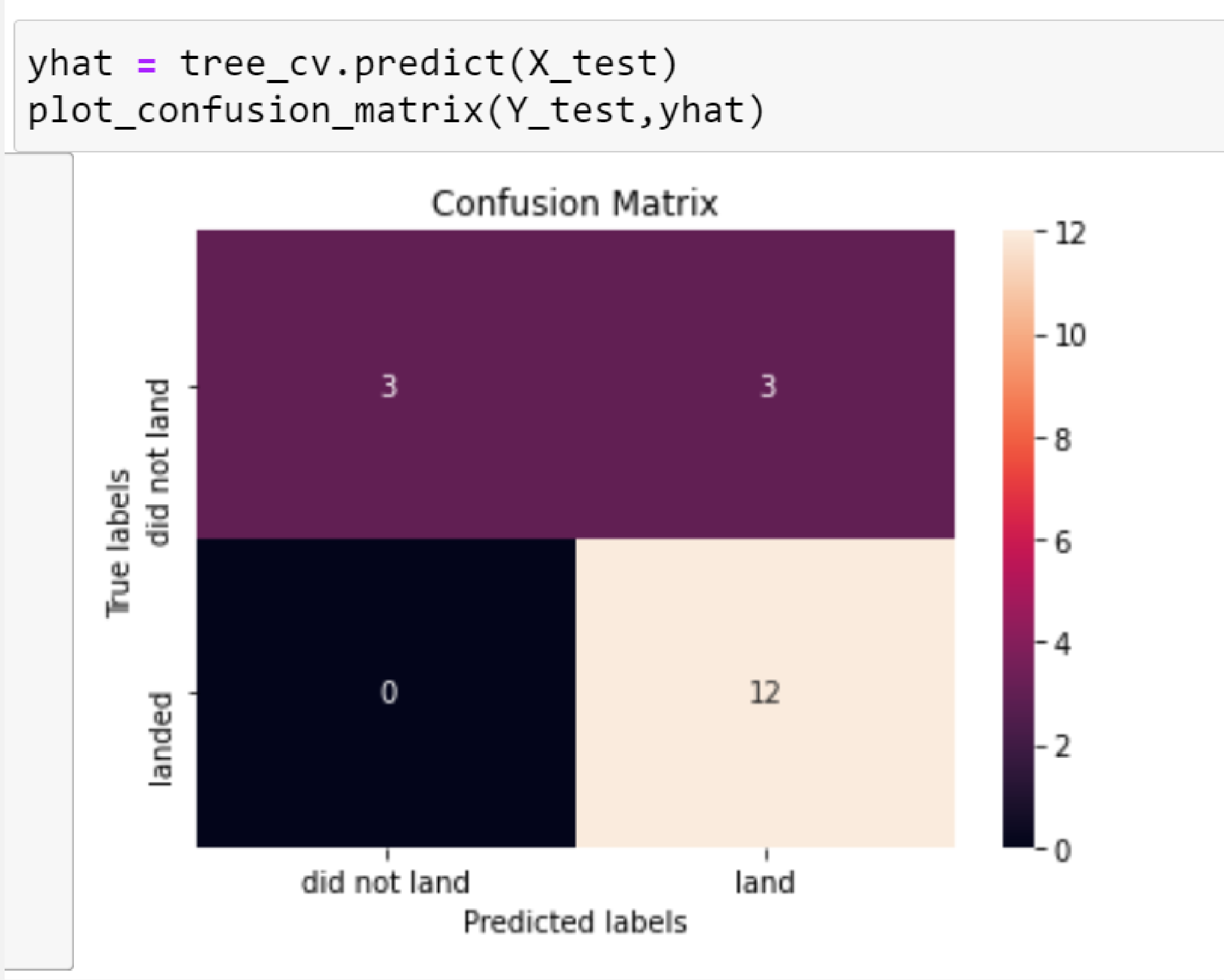
	name	acc_score	best_estimator
0	logreg	0.833333	0.846429
1	svm	0.833333	0.848214
2	decision_tree	0.833333	0.876786
3	knn	0.833333	0.848214

```
In [45]: sns.barplot(y='acc_score', x='name', data=performence)
         plt.show()
```

```
In [46]: sns.barplot(y='best_estimator', x='name', data=performence)
         plt.show()
```

CONFUSION MATRIX FOR DT

- Show the confusion matrix of the best performing model, which is decision tree.



```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

CONCLUSION

- There are three model have been trained and they all have very high accuracy, which is 83.33%

APPENDIX

- Please refer to https://github.com/xiaotied/ibm_ds_project for any codes.

- Thank you!