IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Caio Henrique Sestari

02/11/2022

OUTLINE

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

EXECUTIVE SUMMARY

- Summary of methodologies
    - Data Collection through API
    - Data Collection with Web Scraping
    - Data Wrangling
    - Exploratory Data Analysis with SQL
    - Exploratory Data Analysis with Data Visualization
    - Interactive Visual Analytics with Folium
    - Machine Learning Prediction
- Summary of all results
    - Exploratory Data Analysis result
    - Interactive analytics in screenshots
    - Predictive Analytics result

INTRODUCTION

Project background and context

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. 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 determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

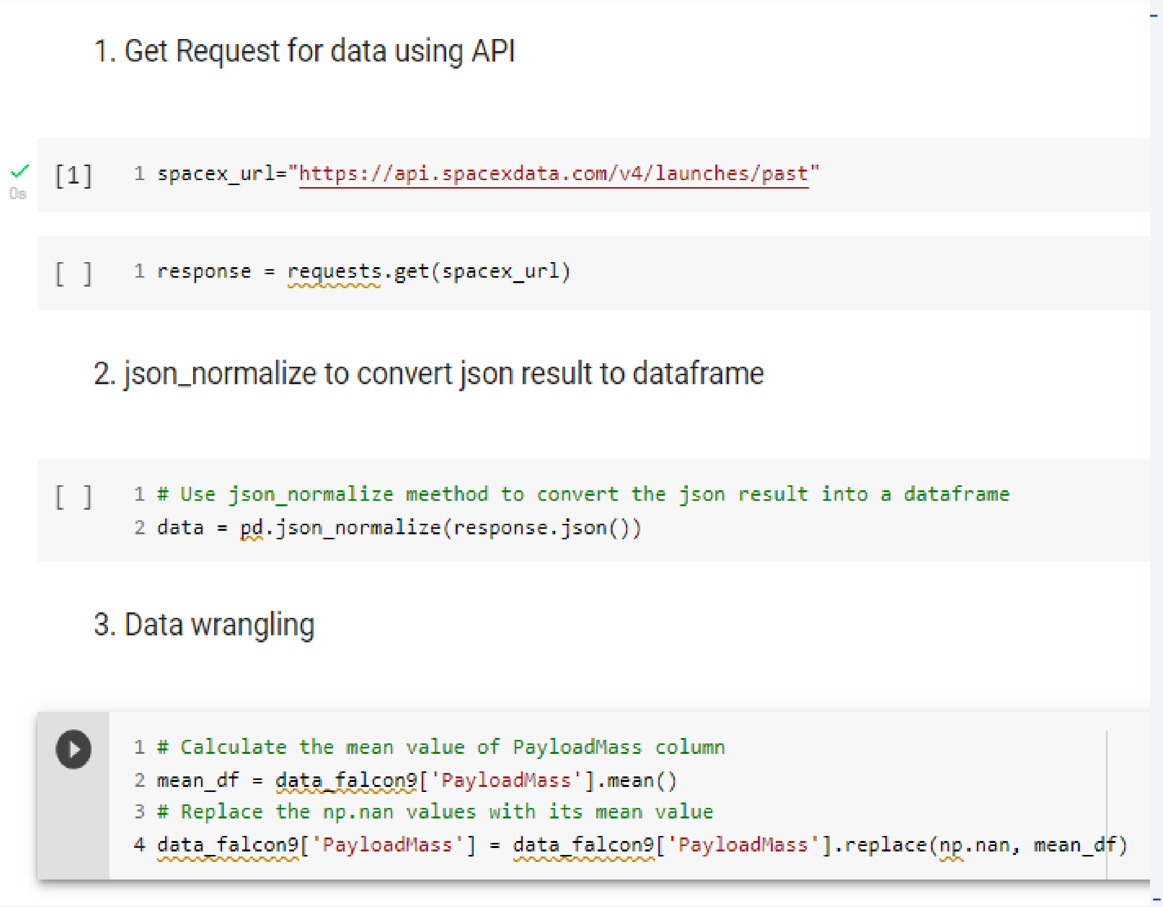
Data Collection

- The data was collected using various methods
  - The data collection was done using get request for the SpaceX API, we decoded the response content as a Json using the .json() function call and then we used .json_normalize() to transform it into a dataframe.
  - We clean the data, check for missing values, and fill in missing values when asked.
  - We web scraped Wikipedia with BeautifulSoup for Falcon 9 release logs.
  - The aim was to extract the release records as HTML table, parse the table and convert it into a pandas dataframe for further analysis

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is

  https://github.com/csestari/Capstone_IBM_DataScience/blob/main/spacex_data_collection_API.ipynb



1. Get Request for data using API

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

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

2. json_normalize to convert json result to dataframe

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

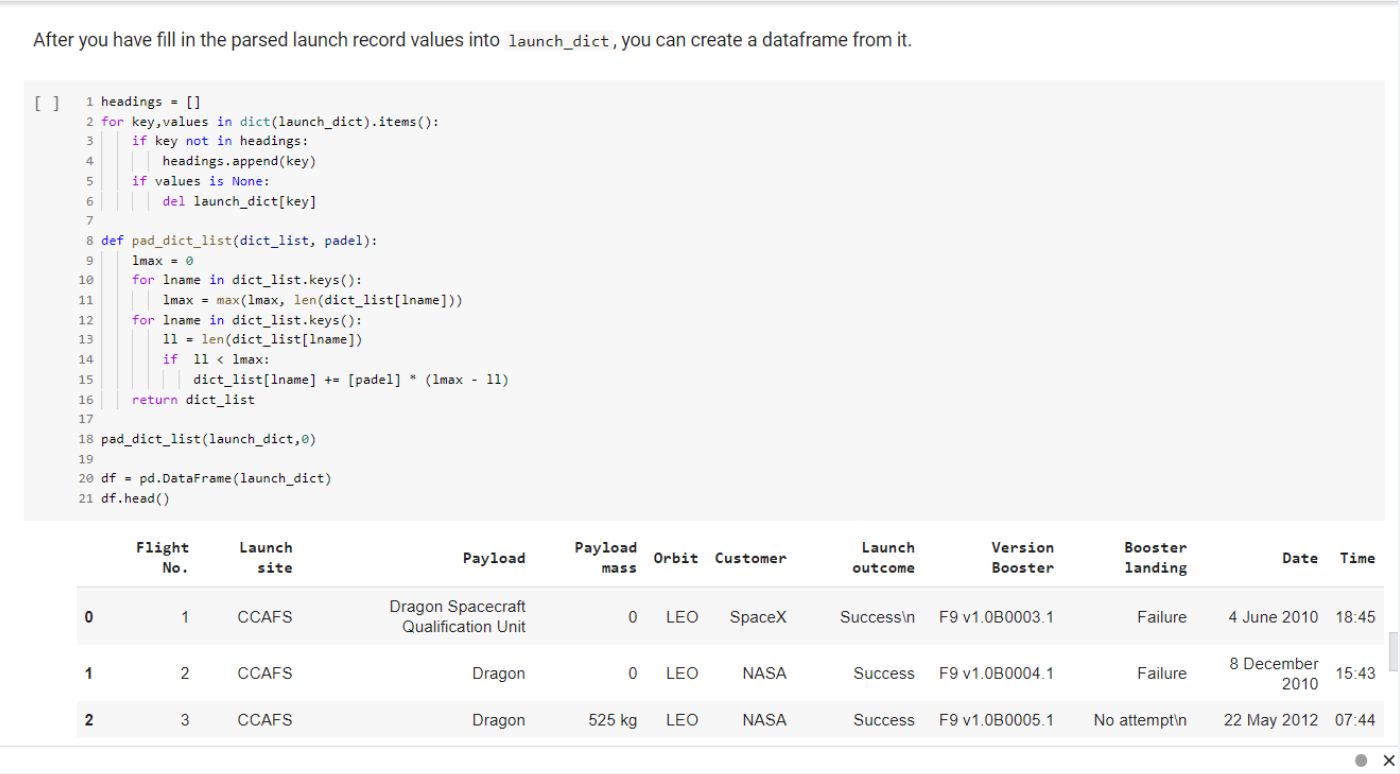
3. Data wrangling

```
# Calculate the mean value of PayloadMass column
mean_df = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean_df)
```

Data Collection - Scraping

- Used web scrapping to Falcon 9 launch logs with BeautifulSoupThe table was analyzed and transformed into a dataframe.

- The link to the notebook:https://git



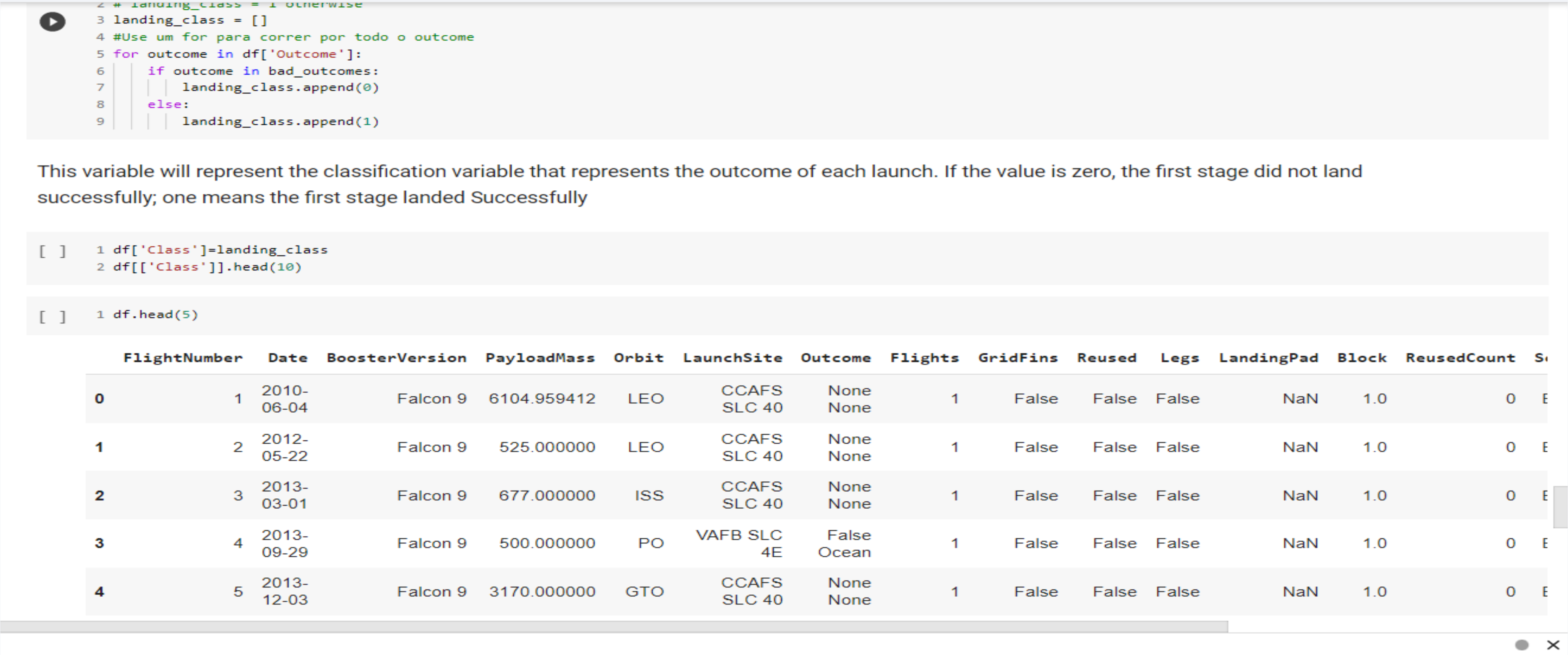
After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

```
[ ]
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

df = pd.DataFrame(launch_dict)
df.head()
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44

Data Wrangling

- Through exploratory data analysis, training labels were determined, the number of launches at each location and the number of occurrences of each orbit were also calculated.The result was exported to .csv
- The link to notebook:https://github.com/csestari/Capstone_IBM_DataScience/blob/main/webscraping.ipynb



```
# landing_class = 1 otherwise
landing_class = []
#Use um for para correr por todo o outcome
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

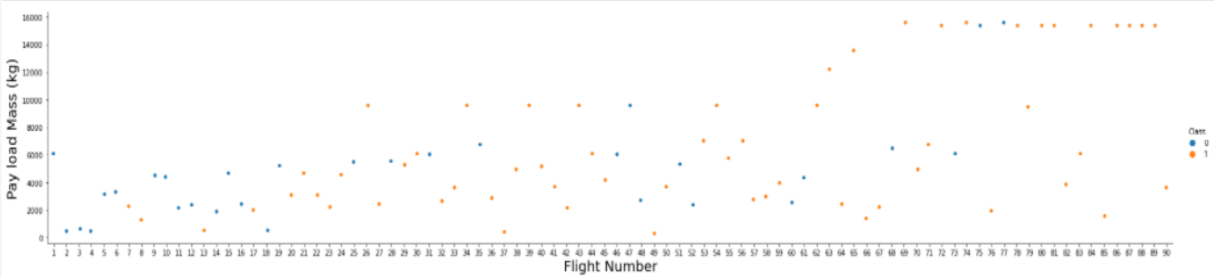
```
[ ]  df['Class']=landing_class
     df[['Class']].head(10)
```

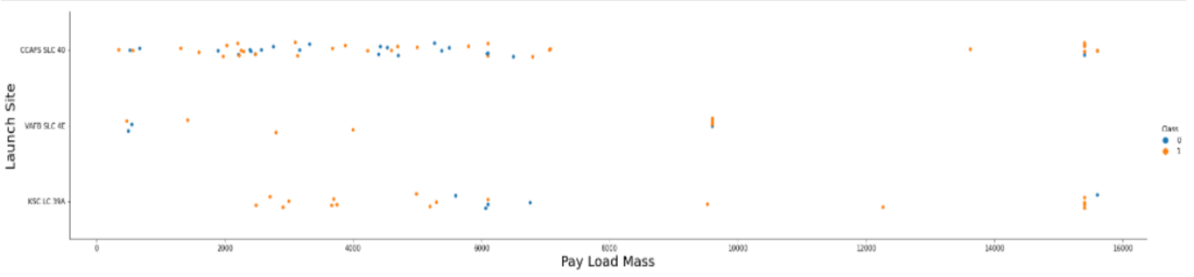
```
[ ]  df.head(5)
```

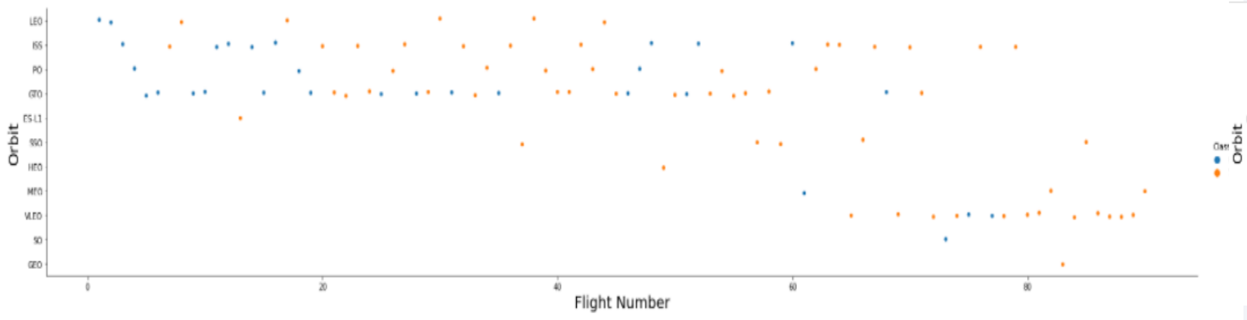
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	S
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	E
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	E
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	E
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	E
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	E

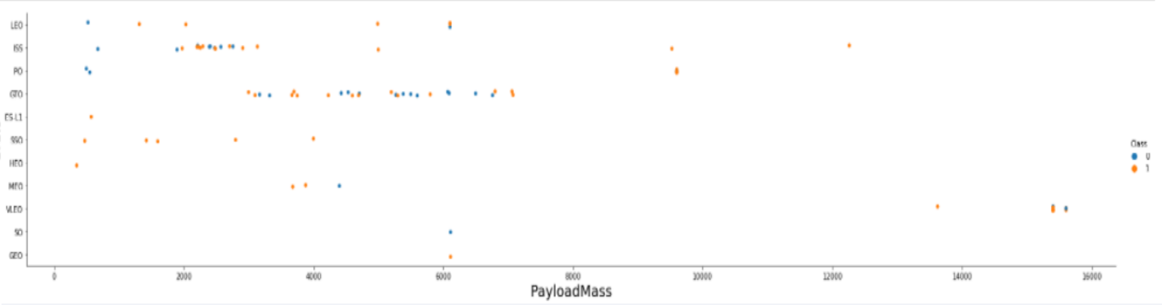
EDA with Data Visualization

- 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, launch success yearly trend.
- The link to notebook:https://github.com/csestari/Capstone_IBM_DataScience/blob/main/EDA%20with%20py.ipynb









EDA with SQL

We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

- ➢ The names of unique launch sites in the space mission.
- ➢ The total payload mass carried by boosters launched by NASA (CRS)
- ➢ The average payload mass carried by booster version F9 v1.1
- ➢ The total number of successful and failure mission outcomes
- ➢ The failed landing outcomes in drone ship, their booster version and launch site names.

- The link to notebook:https://github.com/csestari/Capstone_IBM_DataScience/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- ✓First all launch locations were marked, then we added map objects such as markers, circles, lines for success or failure of launches from each location.

- ✓Classes 0 and 1 were added where 0 for failure and 1 for success, so you can have better analysis.

- ✓It was identified which launch sites have a high success rate compared to others, and the distances between a launch site and its vicinity were also calculated.

- ➢With all these elements on the map it was possible to say: What are the launch sites near railroads, highways and coastlines?
  - ✓Launch sites apparently maintain a certain distance from cities.

- The link to notebook: https://github.com/csestari/Capstone_IBM_DataScience/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Interactive panel was built with Plotly dash
- Pie charts showing the total launches of certain sites, scatter chart showing the relationship with the Result and the Payload Mass (Kg) for the different reinforcement versions.
- The link to notebook: https://github.com/csestari/Capstone_IBM_DataScience/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Data were transformed into training and testing.
- Different machine learning models were built and adjusted to different hyperparameters using GridSearchCV.
- Score accuracy as a metric for our model, the model has been improved using feature engineering and algorithm tuning techniques.
- The classification model with the best performance was found using GridSearchCV in the same way that we used for the different models.
- The link to notebook: https://github.com/csestari/Capstone_IBM_DataScience/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

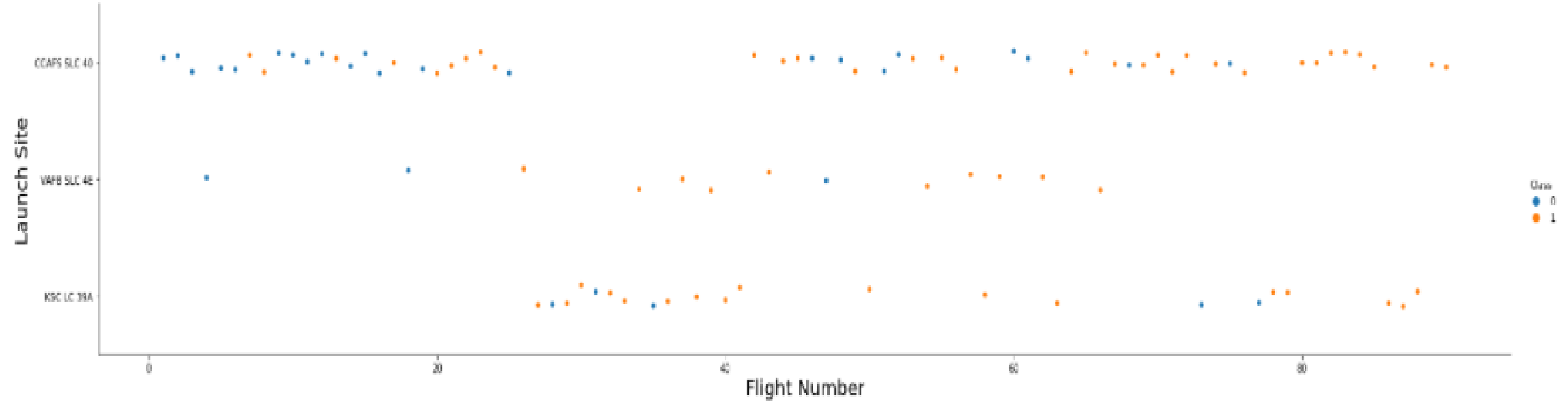
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

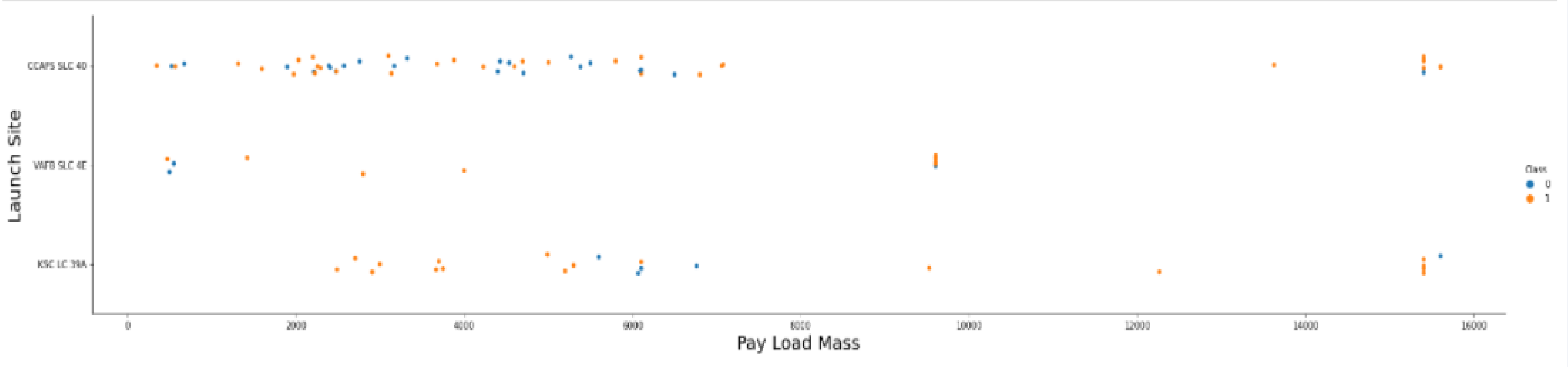
Flight Number vs. Launch Site

- With the graph, it is possible to see that the greater the number of flights in a launch location, the greater the success rate at the same location.

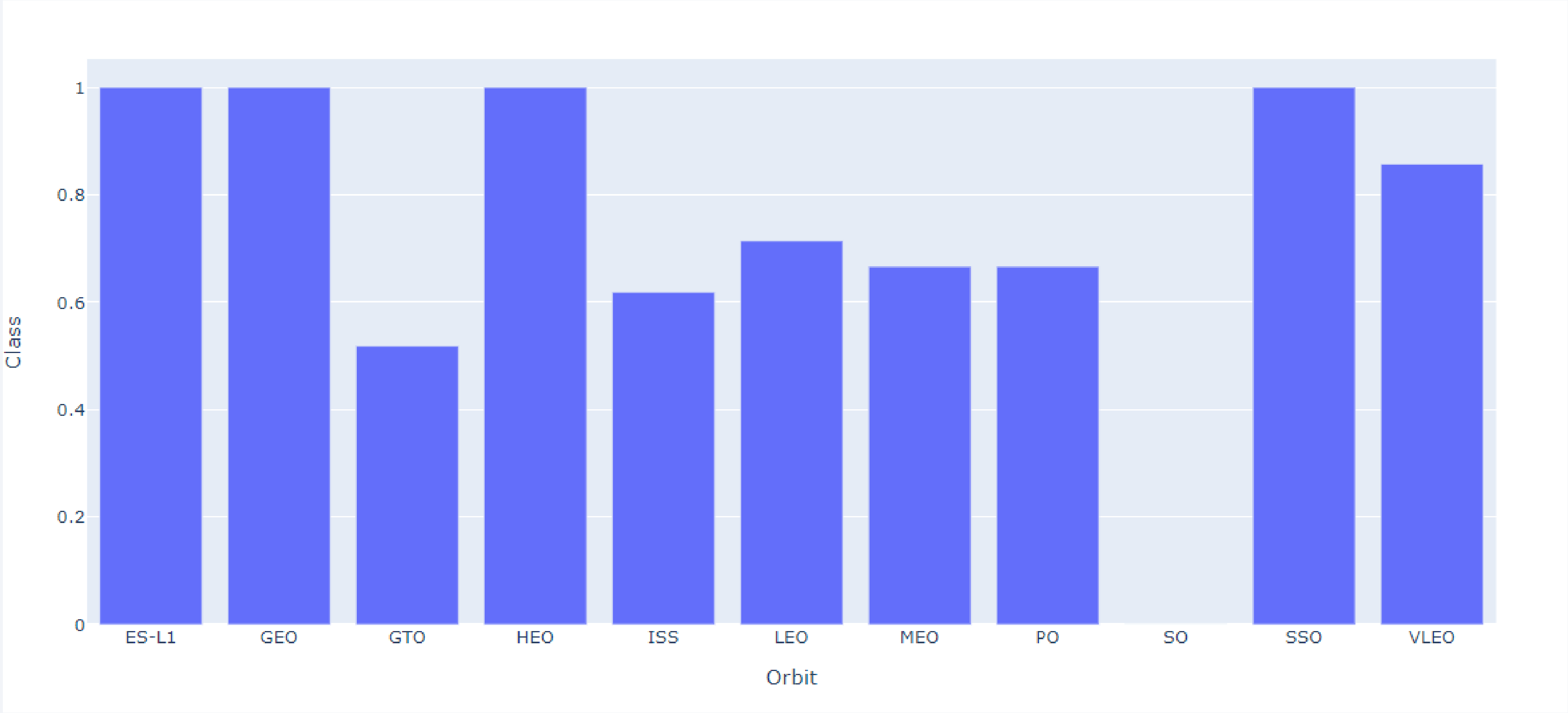


Payload vs. Launch Site

- Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

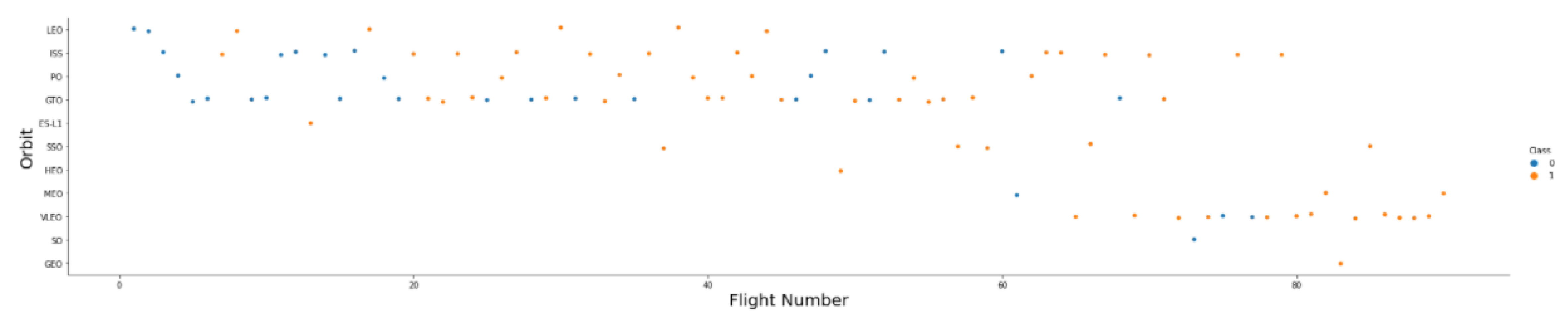


Success Rate vs. Orbit Type



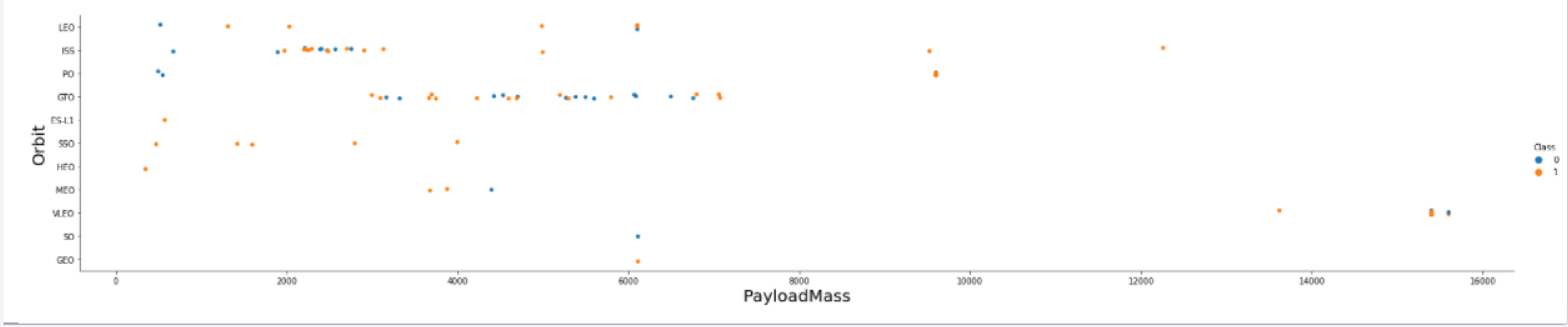
Flight Number vs. Orbit Type

- You should see 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.

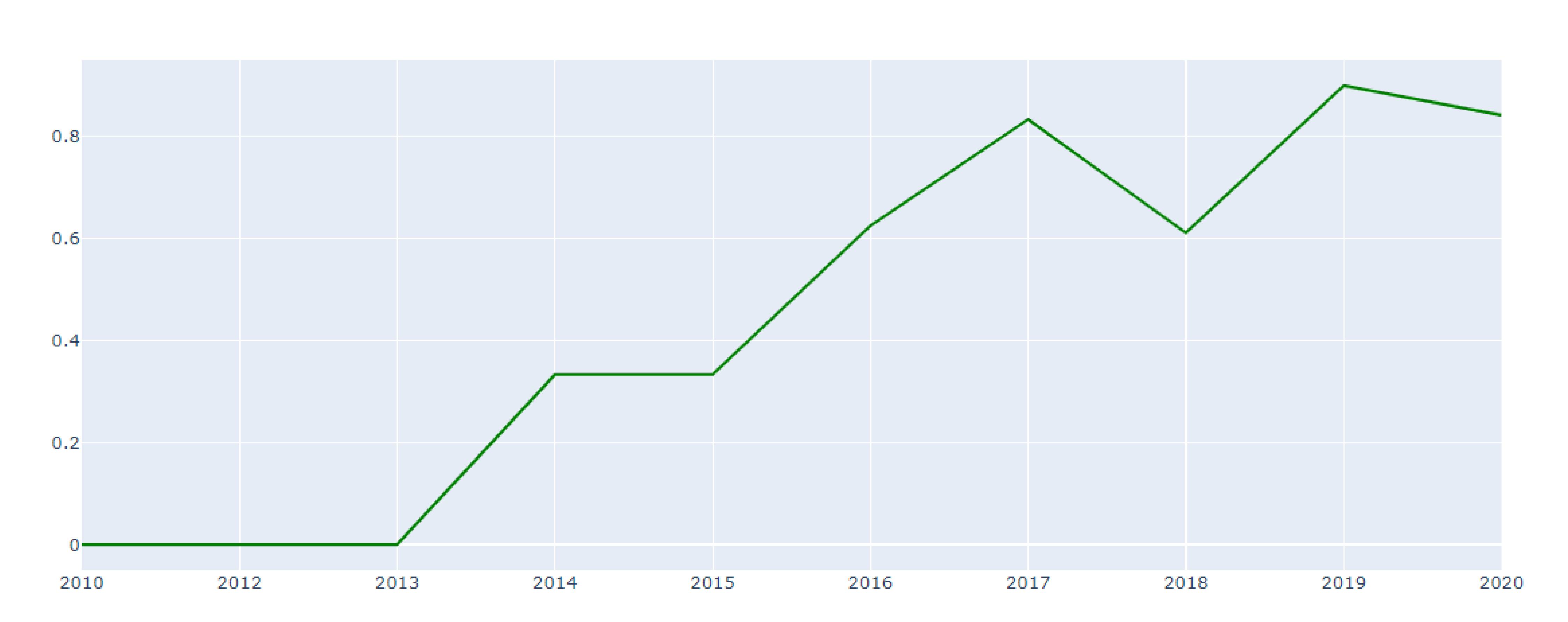


Payload vs. Orbit Type

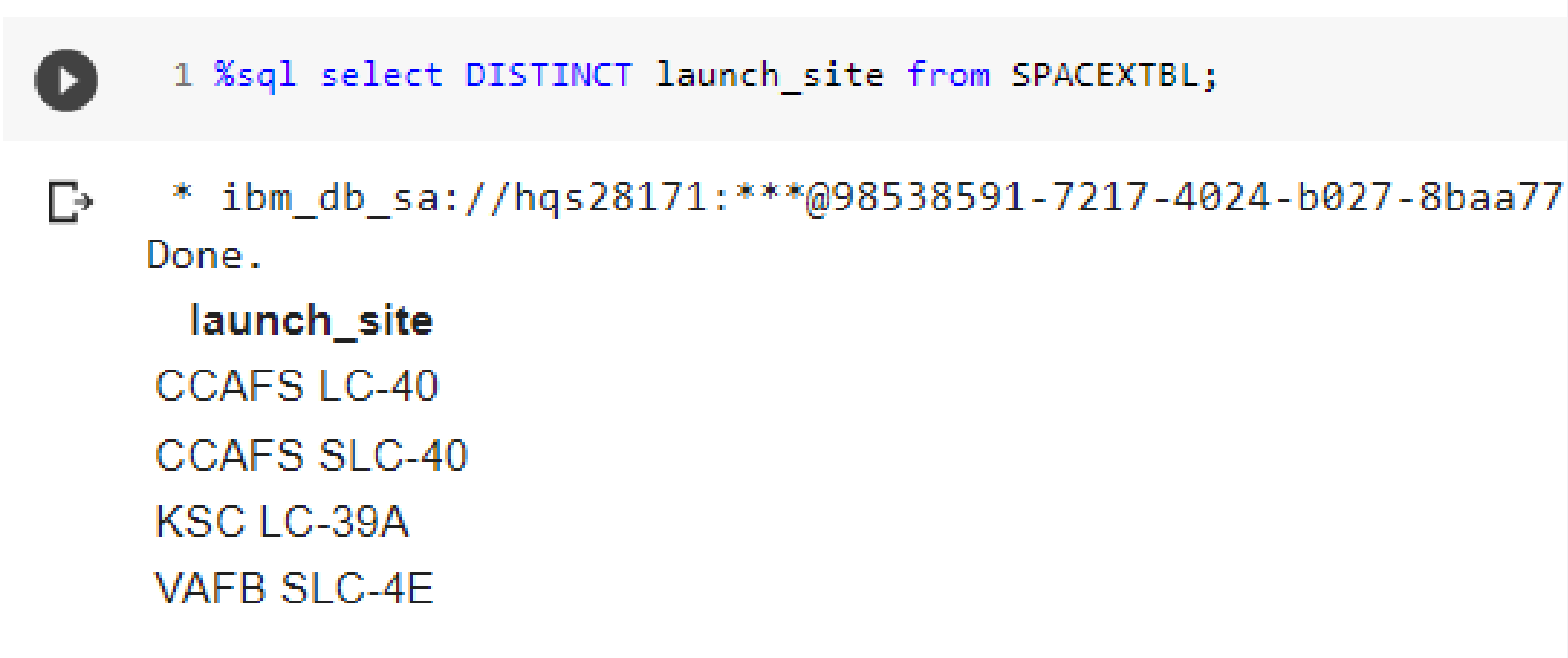
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend



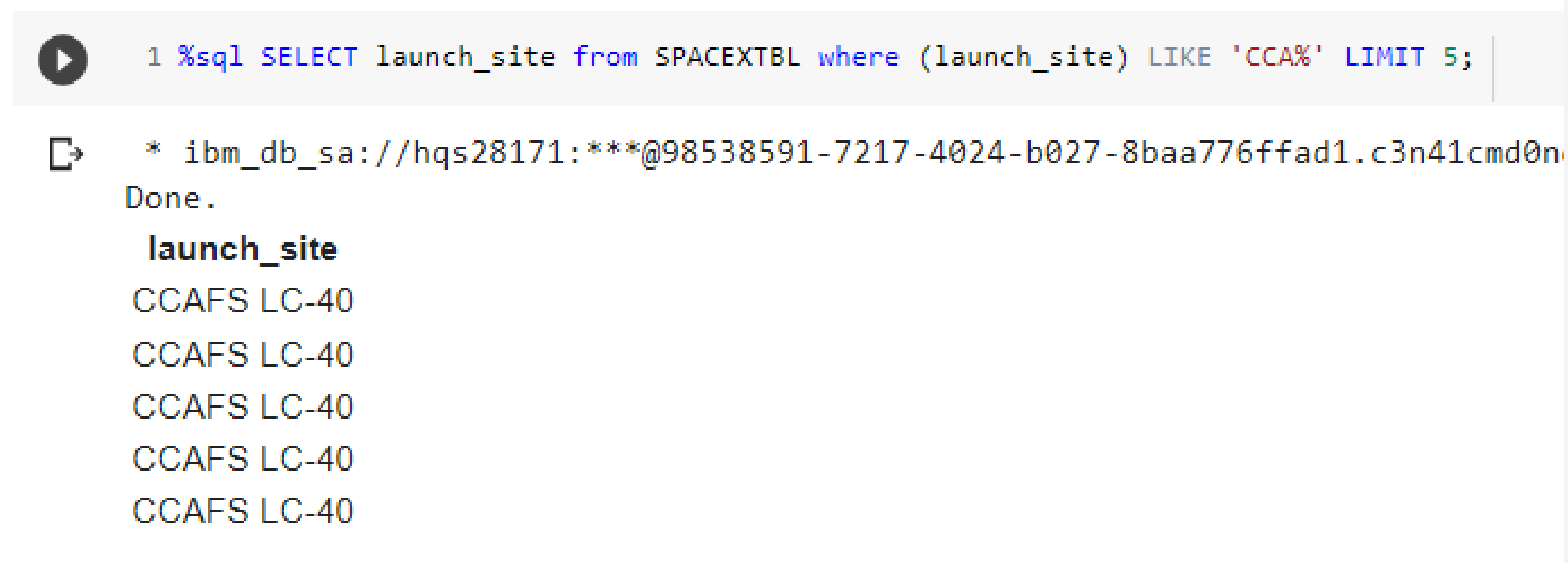
All Launch Site Names



```
%sql select DISTINCT launch_site from SPACEXTBL;
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa77
Done.
  launch_site
 CCAFS LC-40
 CCAFS SLC-40
 KSC LC-39A
 VAFB SLC-4E
```

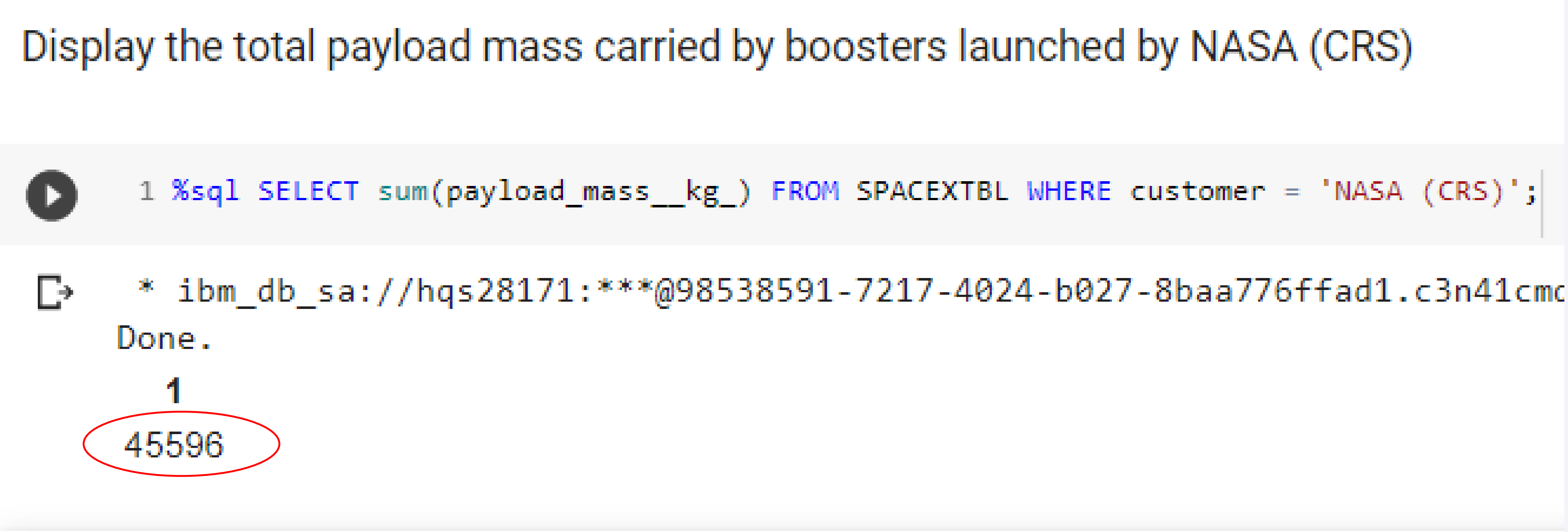
Launch Site Names Begin with 'CCA'



```
%sql SELECT launch_site from SPACEXTBL where (launch_site) LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0n
Done.
 launch_site
 CCAFS LC-40
 CCAFS LC-40
 CCAFS LC-40
 CCAFS LC-40
 CCAFS LC-40
```

Total Payload Mass

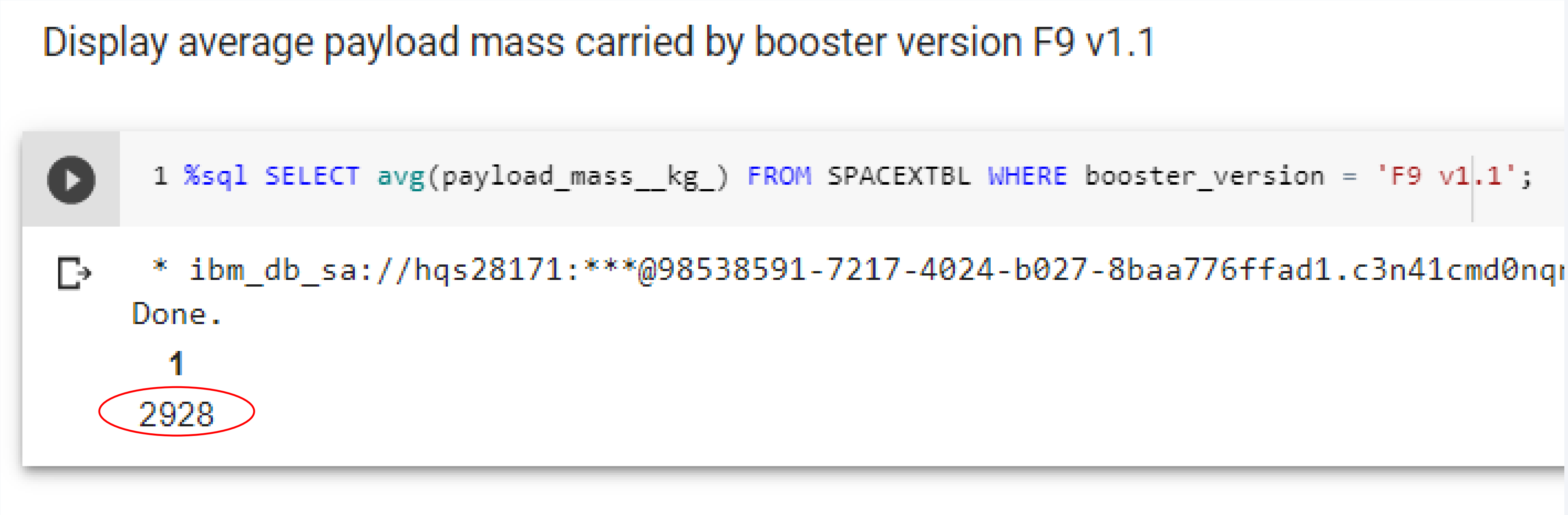


Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT sum(payload_mass__kg_) FROM SPACEXTBL WHERE customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd
Done.
   1
 45596
```

Average Payload Mass by F9 v1.1

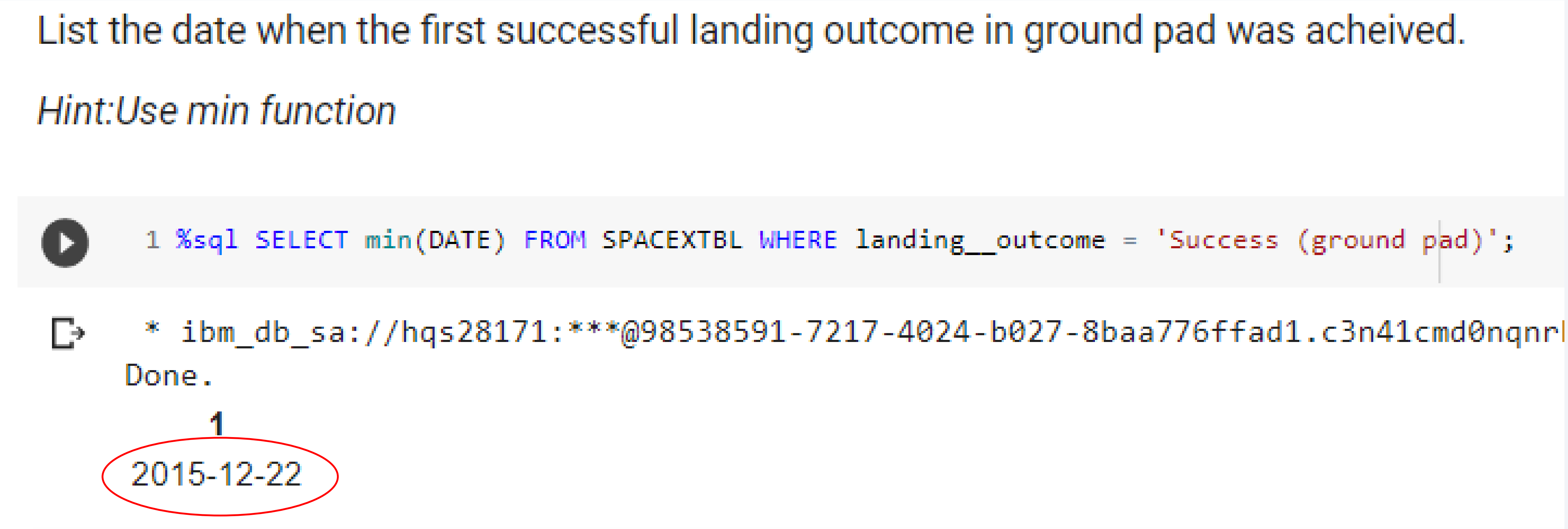


Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT avg(payload_mass__kg_) FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nq
Done.
  1
2928
```

First Successful Ground Landing Date



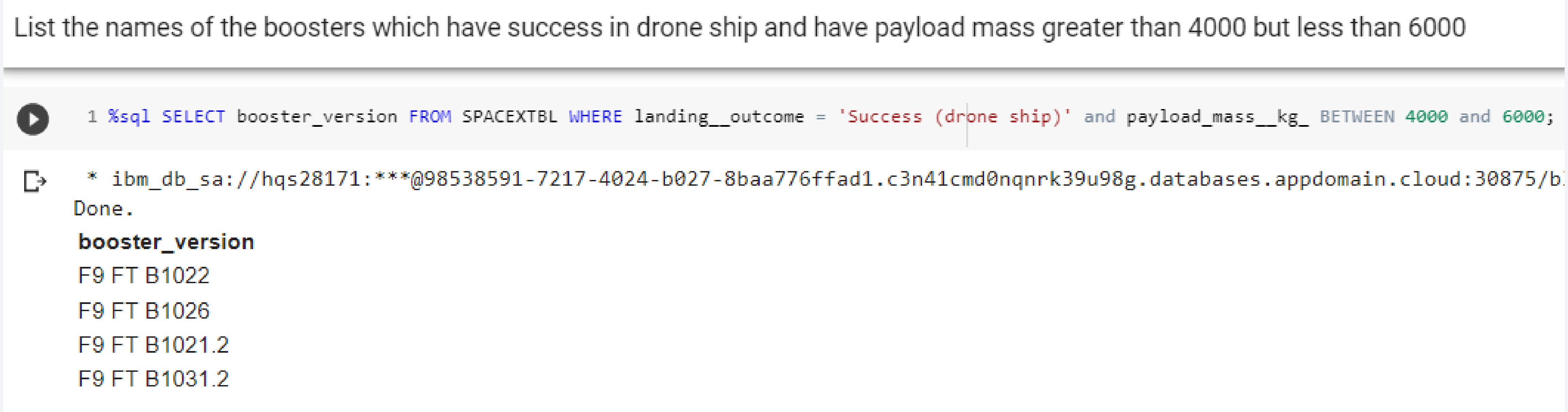
List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnr
Done.
     1
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000



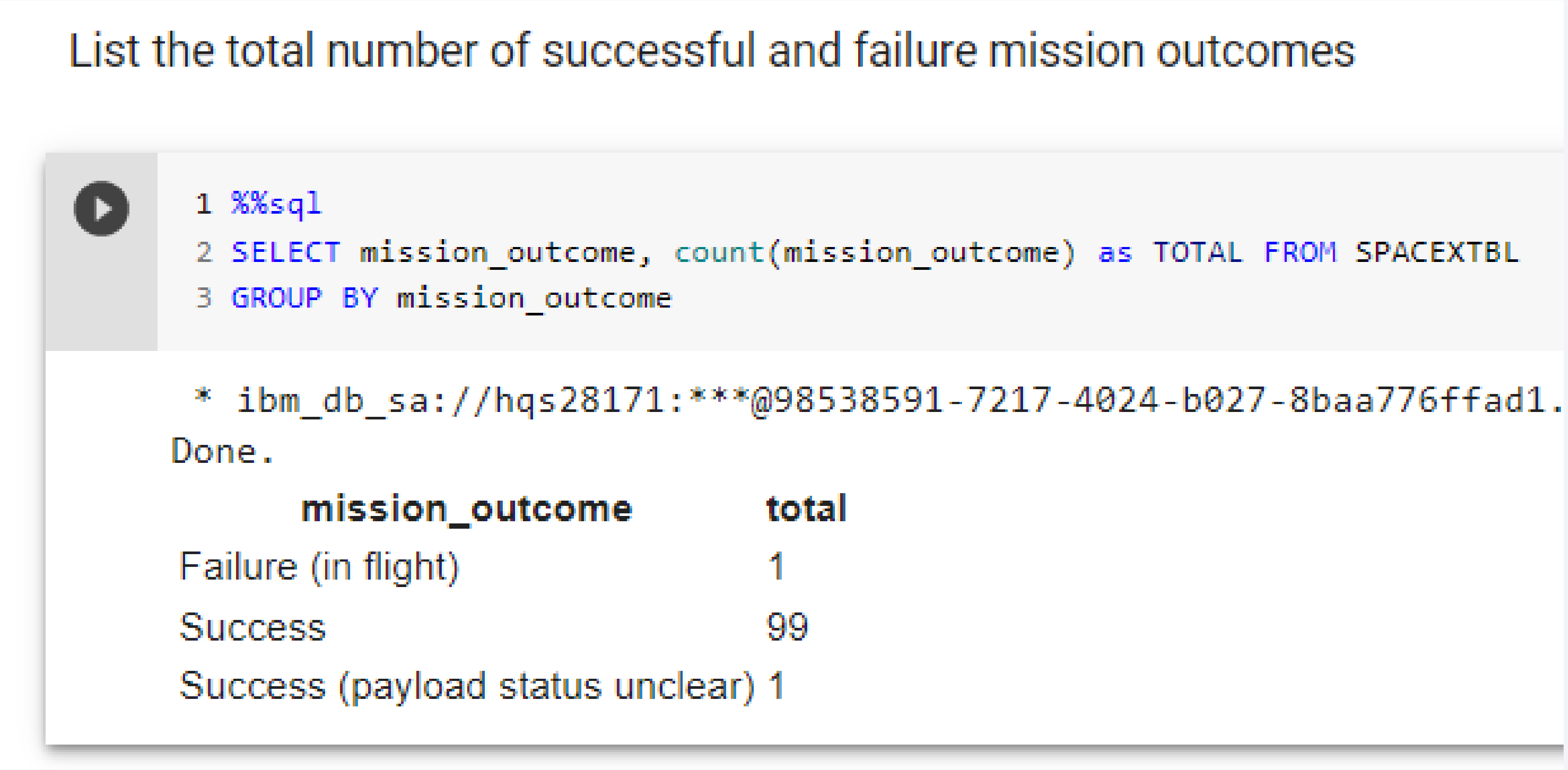
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT booster_version FROM SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 and 6000;
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/b
Done.
```

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

Total Number of Successful and Failure Mission Outcomes



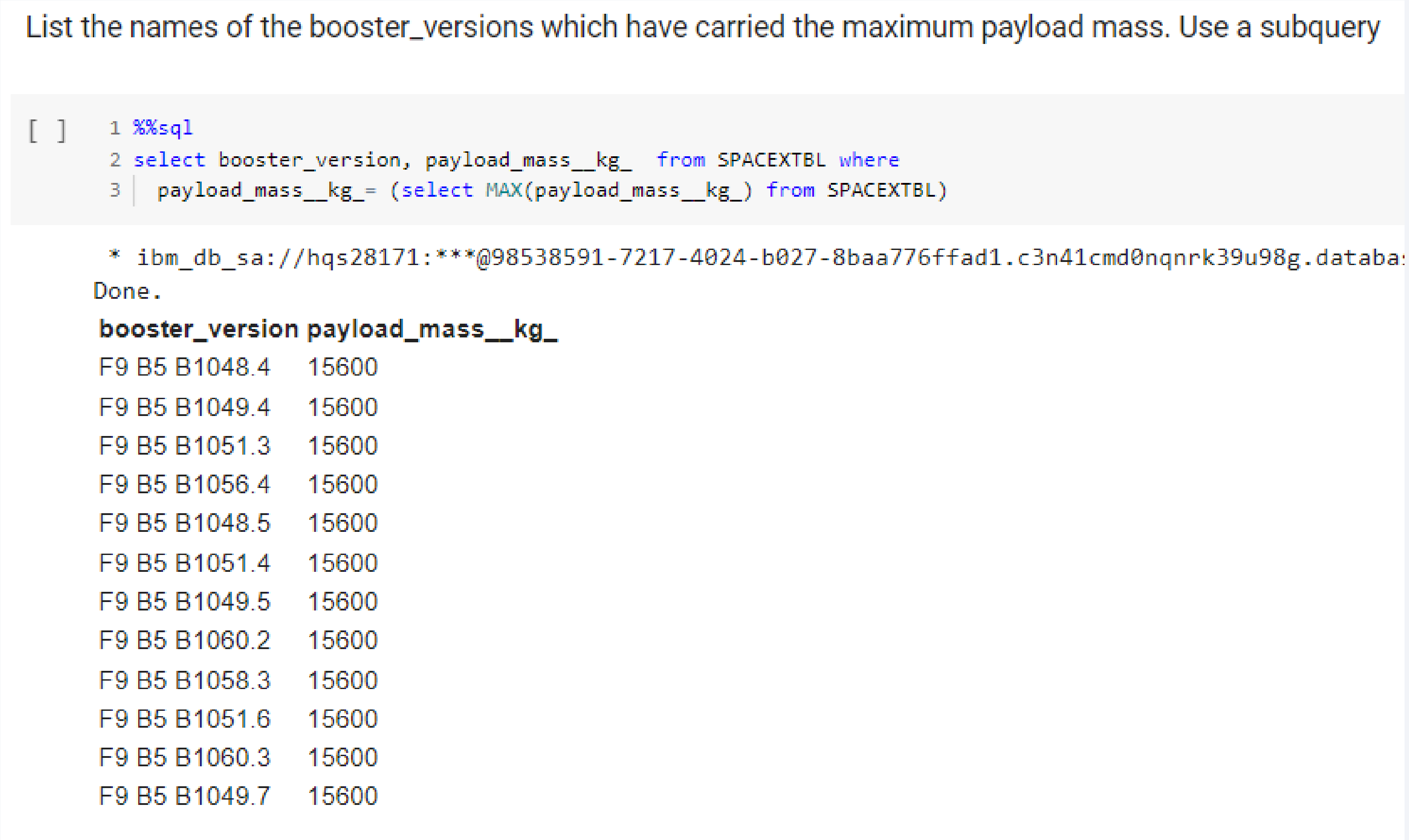
List the total number of successful and failure mission outcomes

```
%%sql
SELECT mission_outcome, count(mission_outcome) as TOTAL FROM SPACEXTBL
GROUP BY mission_outcome
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.
Done.
```

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

Boosters Carried Maximum Payload



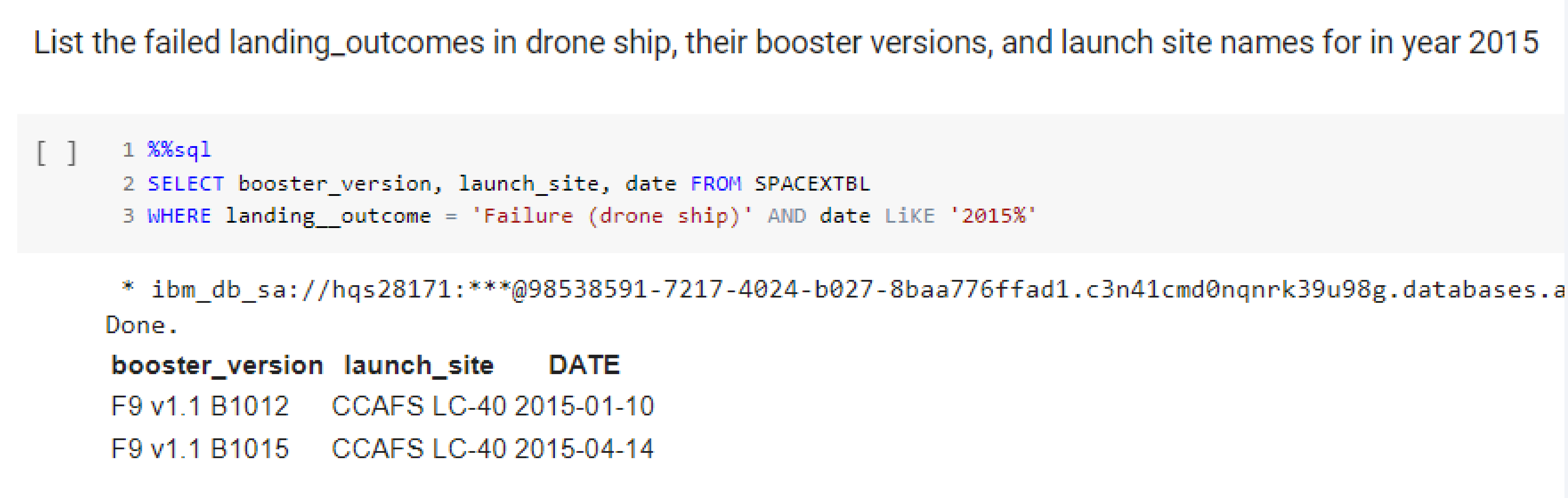
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[ ]  %%sql
     select booster_version, payload_mass__kg_  from SPACEXTBL where
      payload_mass__kg_= (select MAX(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databa
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

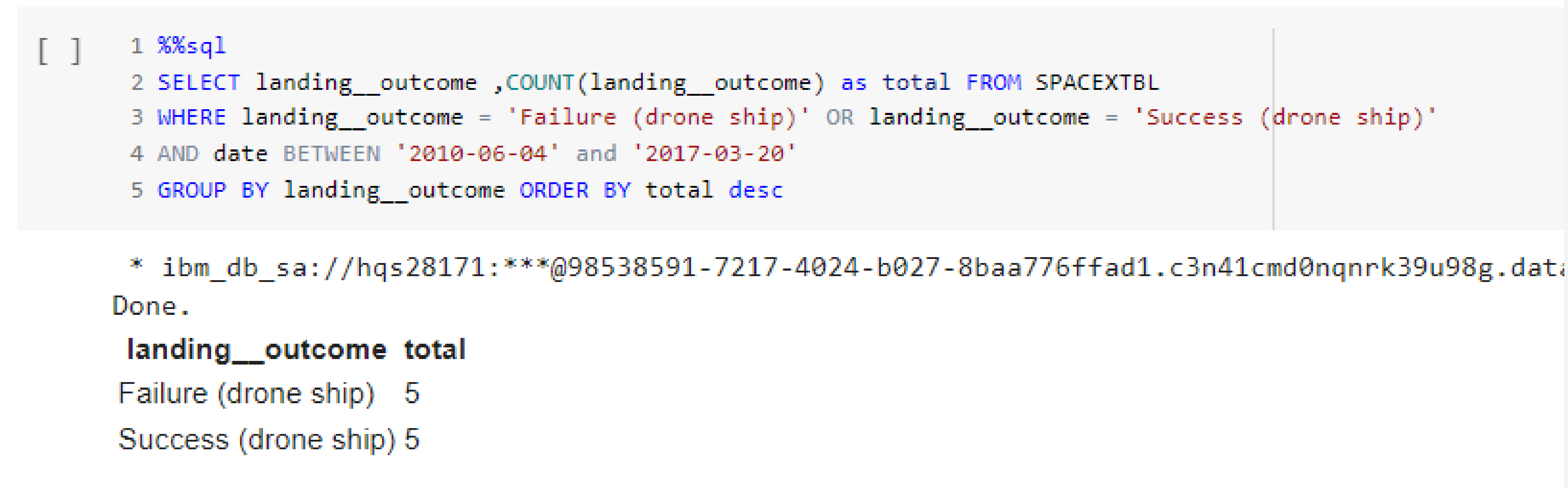


List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
SELECT booster_version, launch_site, date FROM SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)' AND date LiKE '2015%'
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.a
Done.
booster_version  launch_site     DATE
F9 v1.1 B1012    CCAFS LC-40 2015-01-10
F9 v1.1 B1015    CCAFS LC-40 2015-04-14
```

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



[ ]

```
%%sql
SELECT landing__outcome ,COUNT(landing__outcome) as total FROM SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)' OR landing__outcome = 'Success (drone ship)'
AND date BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing__outcome ORDER BY total desc
```

```
 * ibm_db_sa://hqs28171:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.data
Done.
```

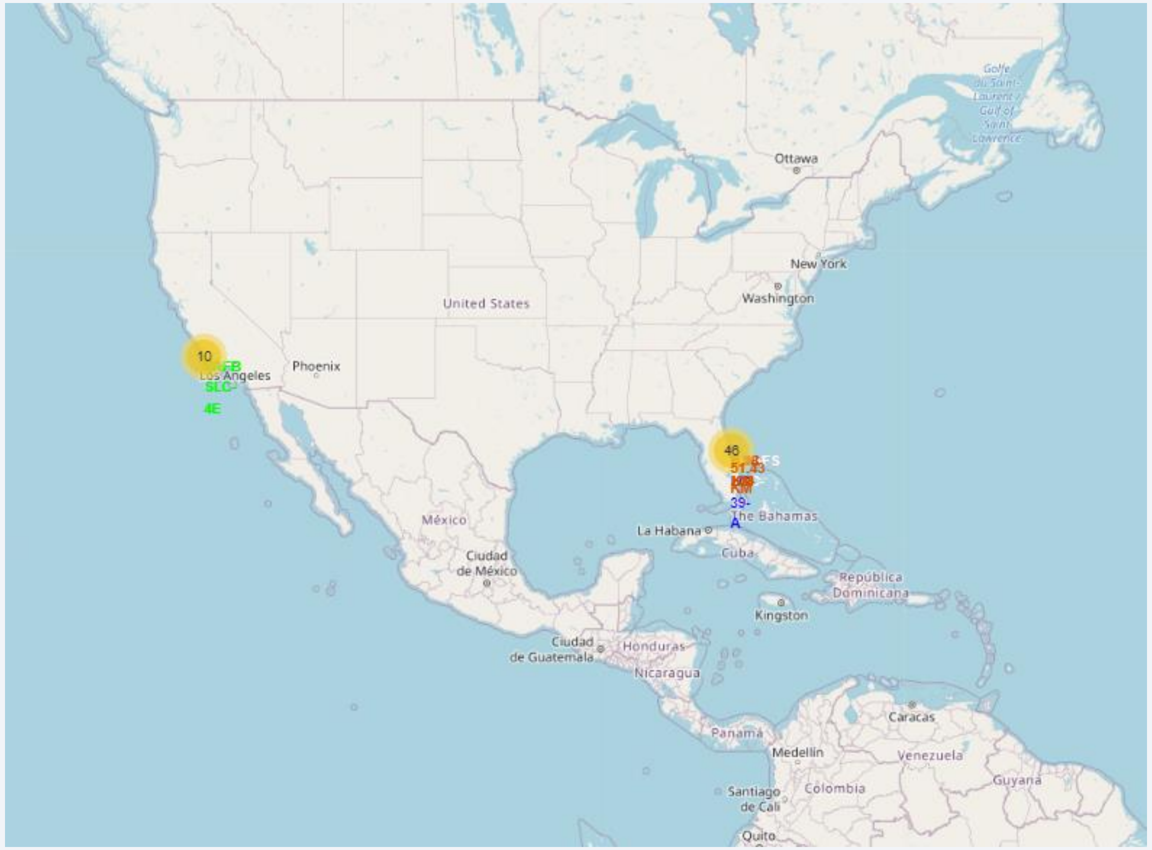
landing__outcome	total
Failure (drone ship)	5
Success (drone ship)	5

Section 3

Launch Sites Proximities Analysis

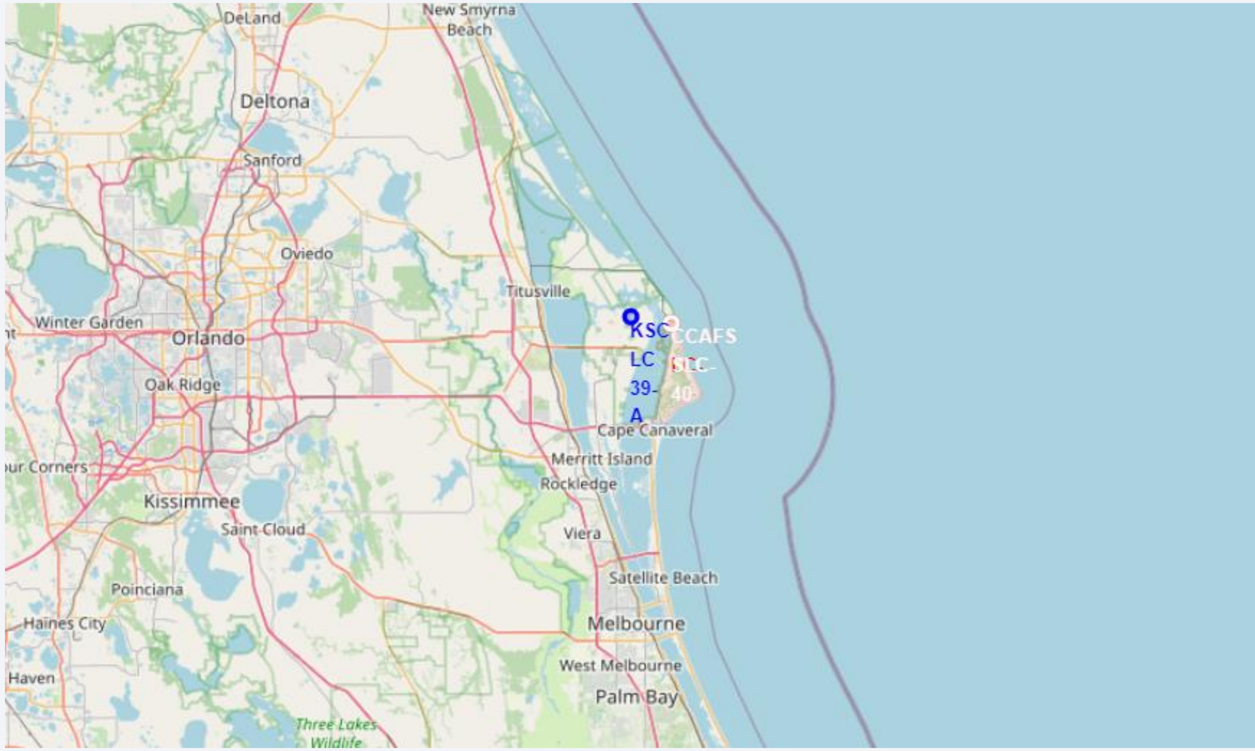
Spacex – Launch Sites Location Map

You can see that all launches are made on the US coast, and most are made in the Florida region.

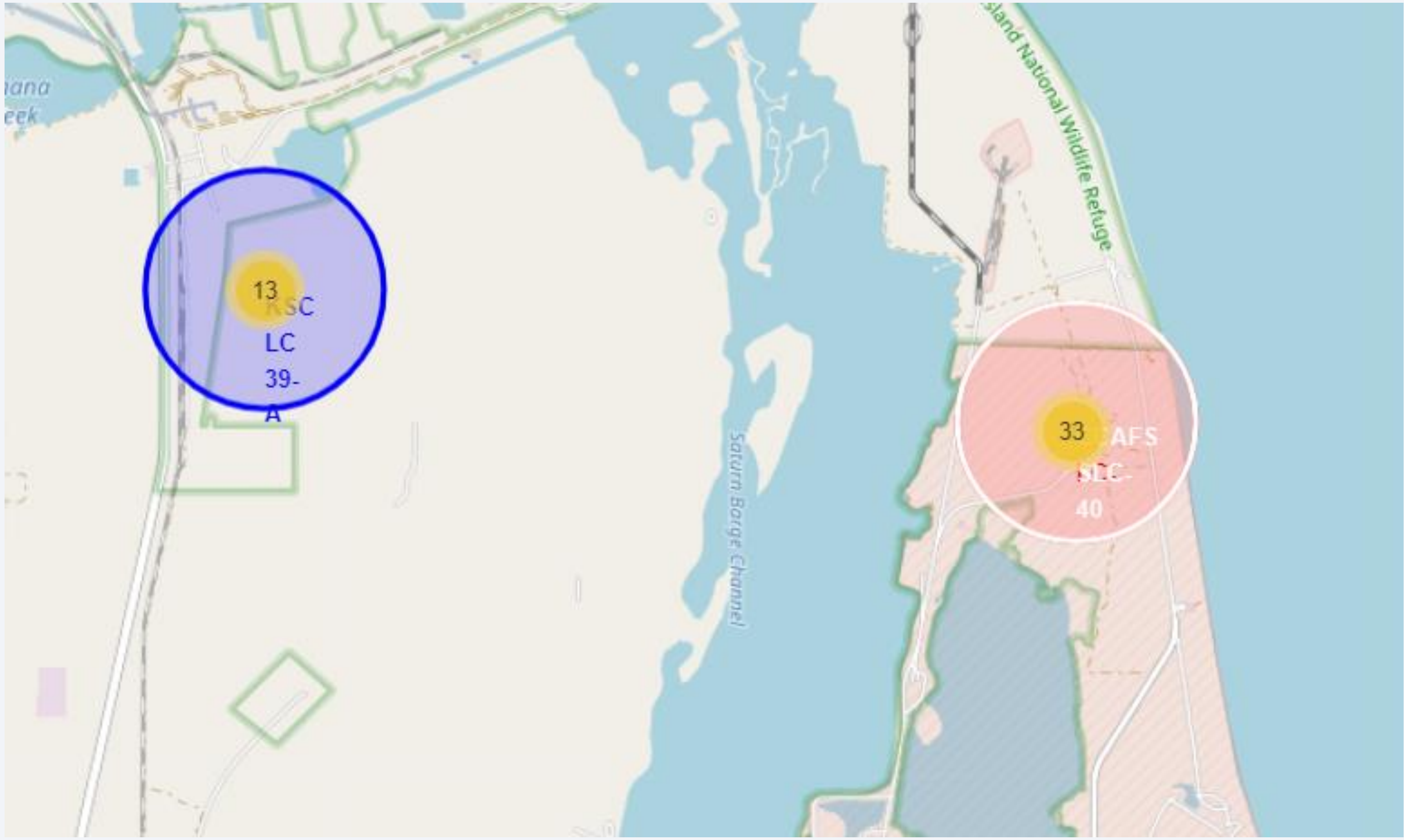


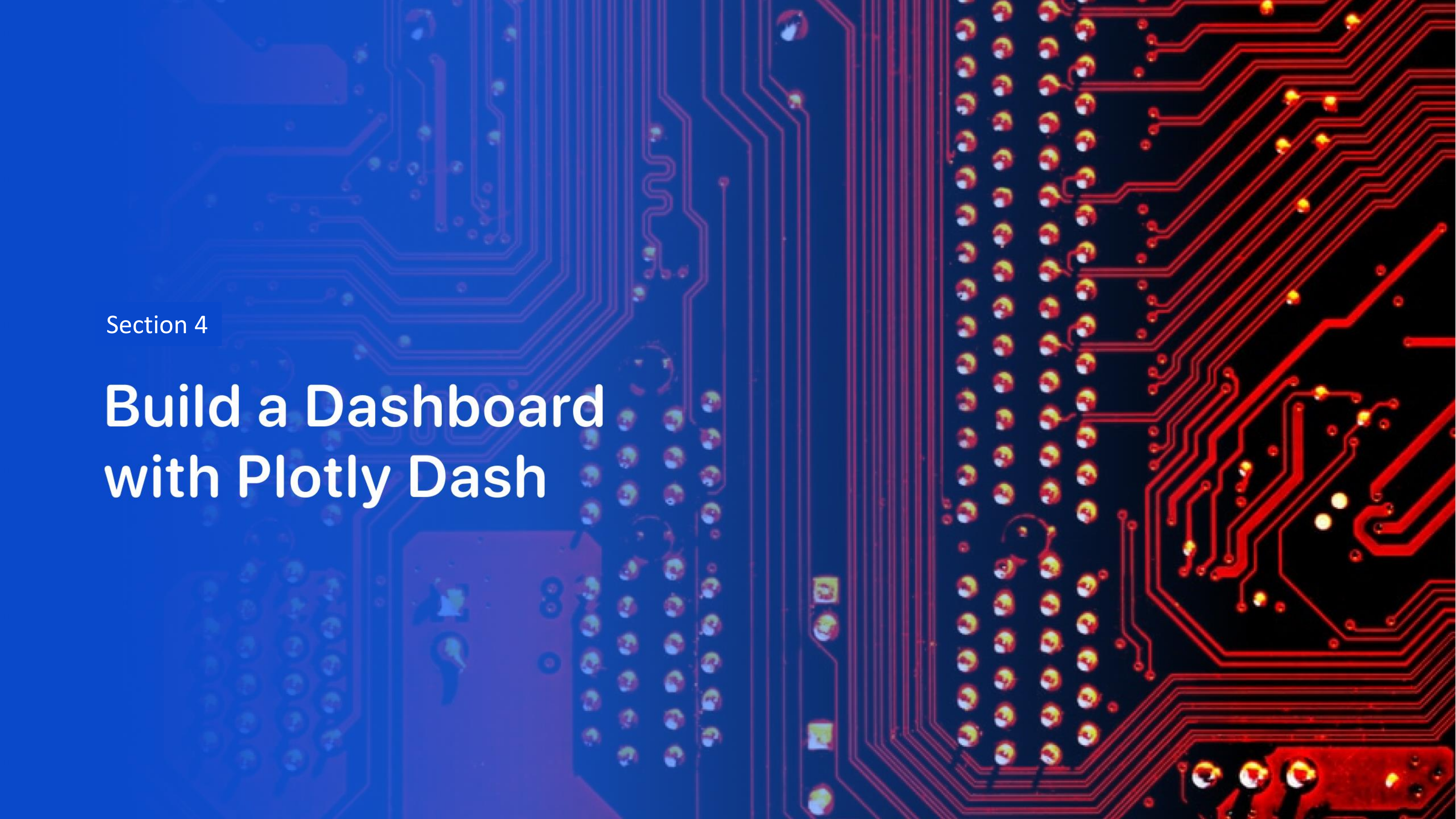
Spacex – Launch Sites Location Map 2

You can see that there are three launch sites nearby, and two in the same location.



Spacex – Launch Sites Location Map 3

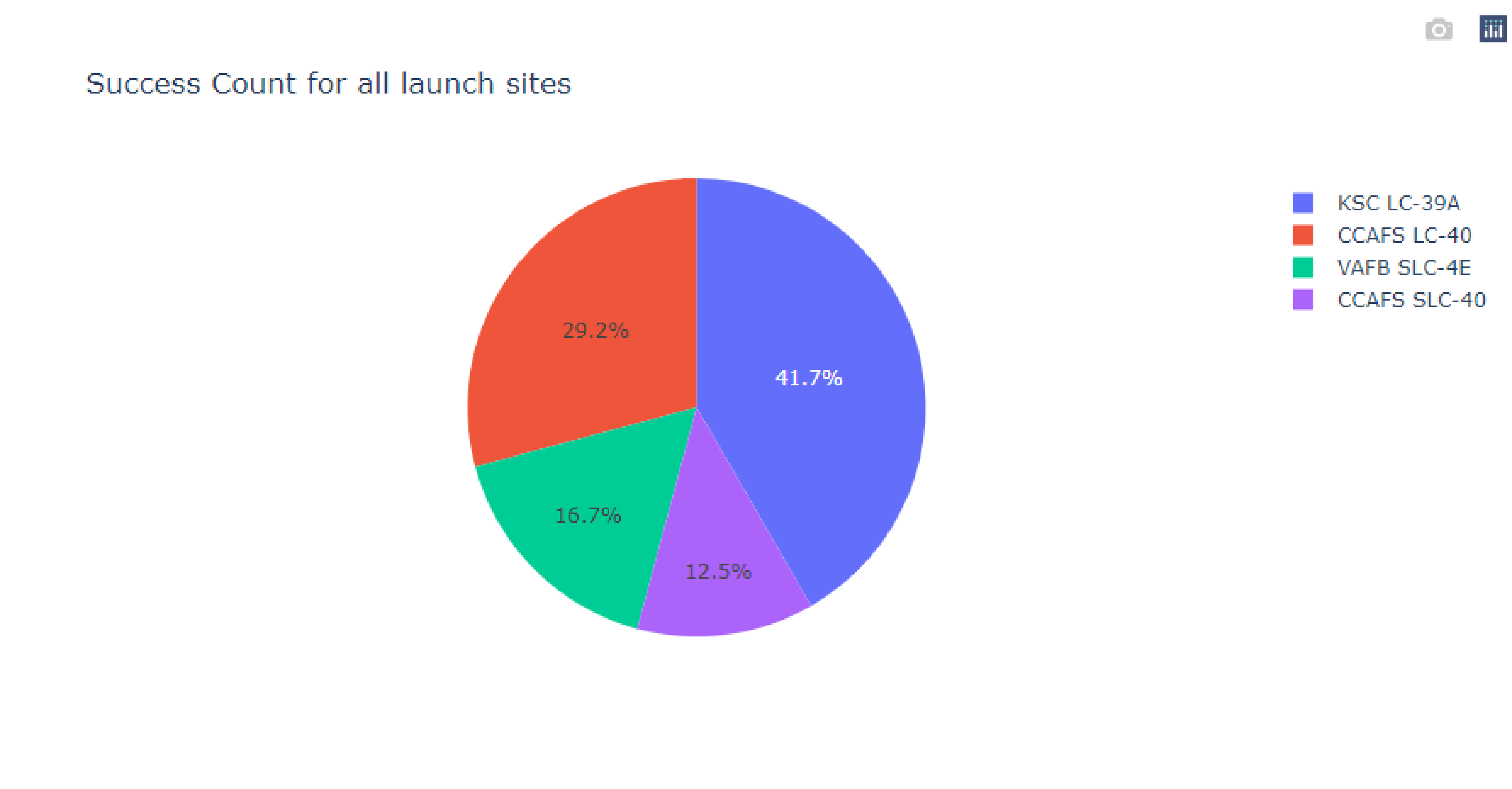




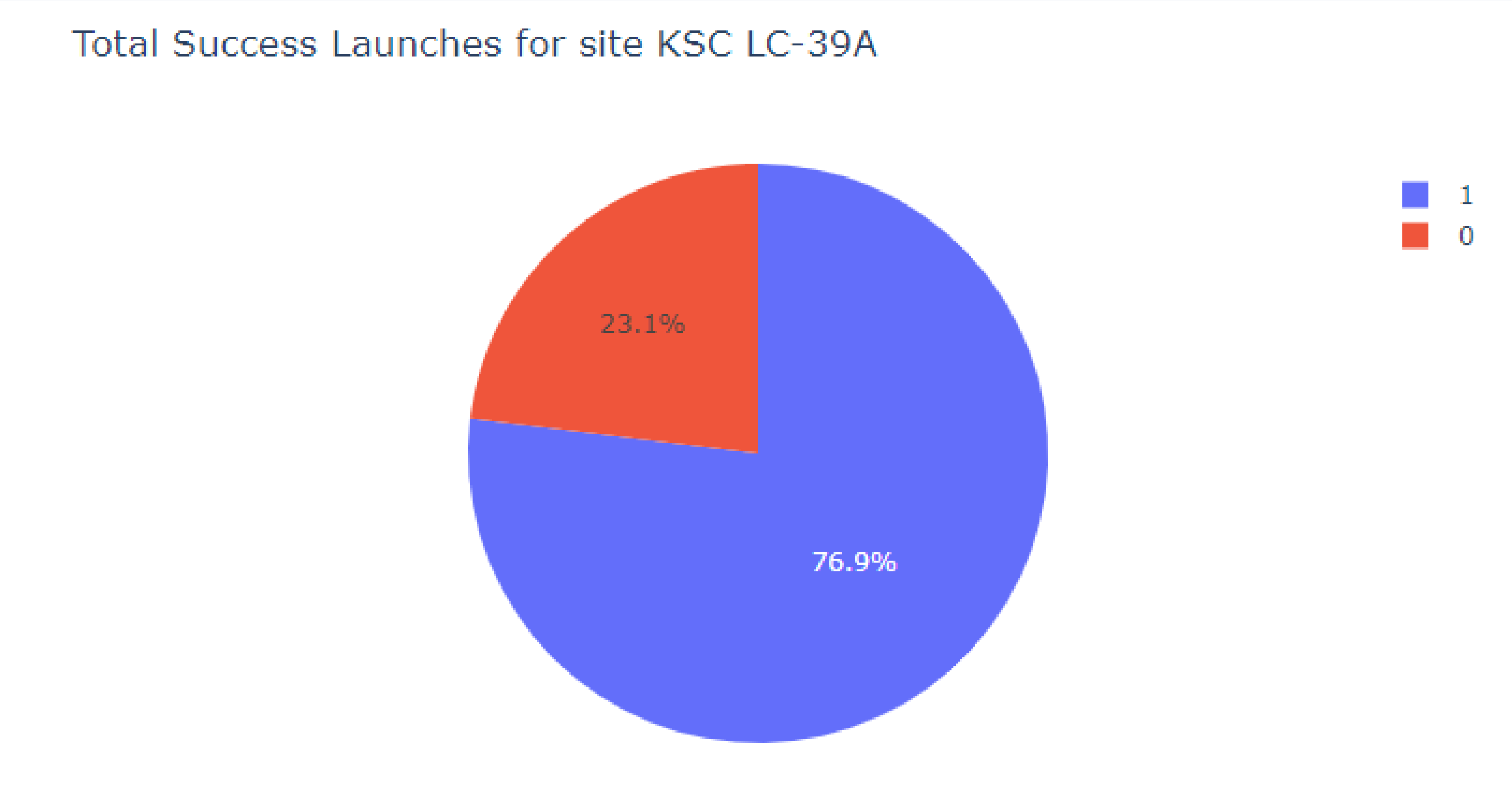
Section 4

Build a Dashboard with Plotly Dash

Dashboard - Representation of all sites

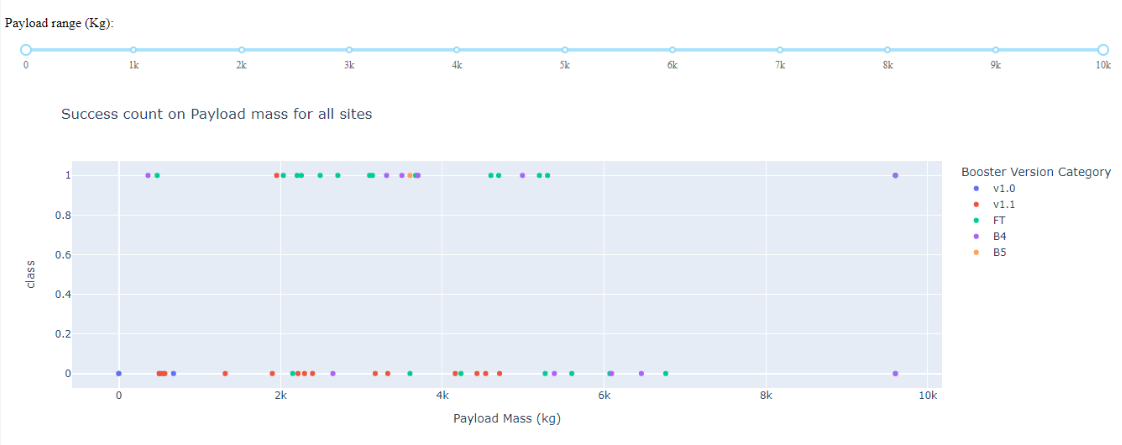


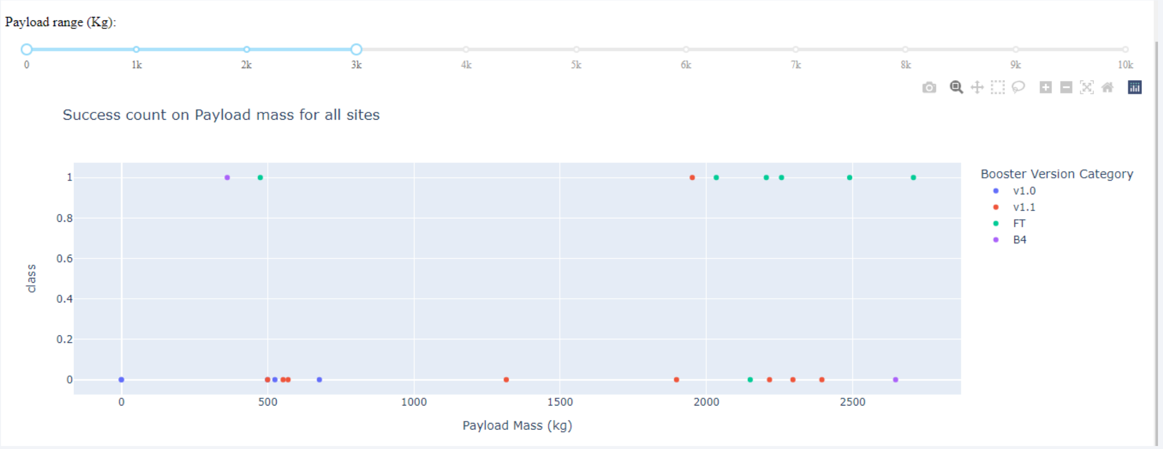
Dashboard - Site with the highest success rate



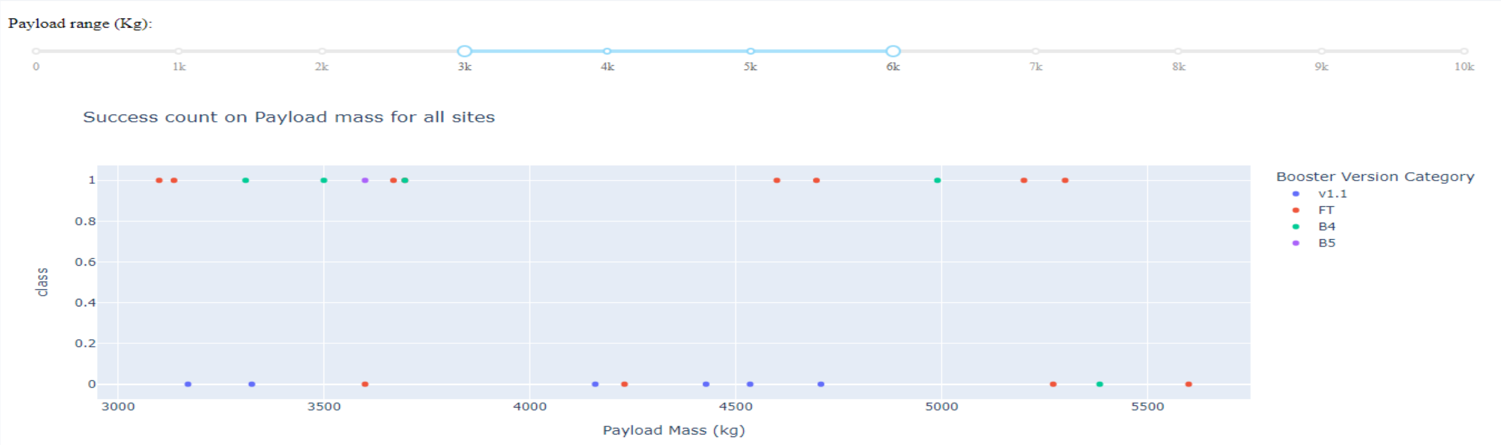
We see that they have a success rate of 76.9%

Success count on Payload for all sites in different ranges





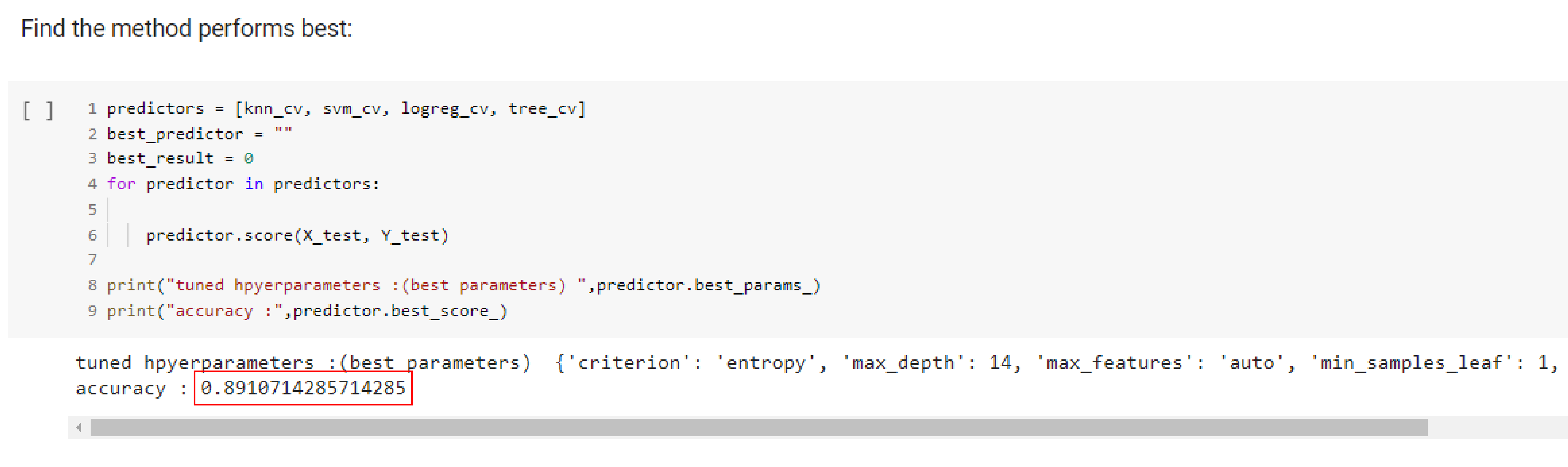
Best Range :



Section 5

Predictive Analysis (Classification)

Classification Accuracy

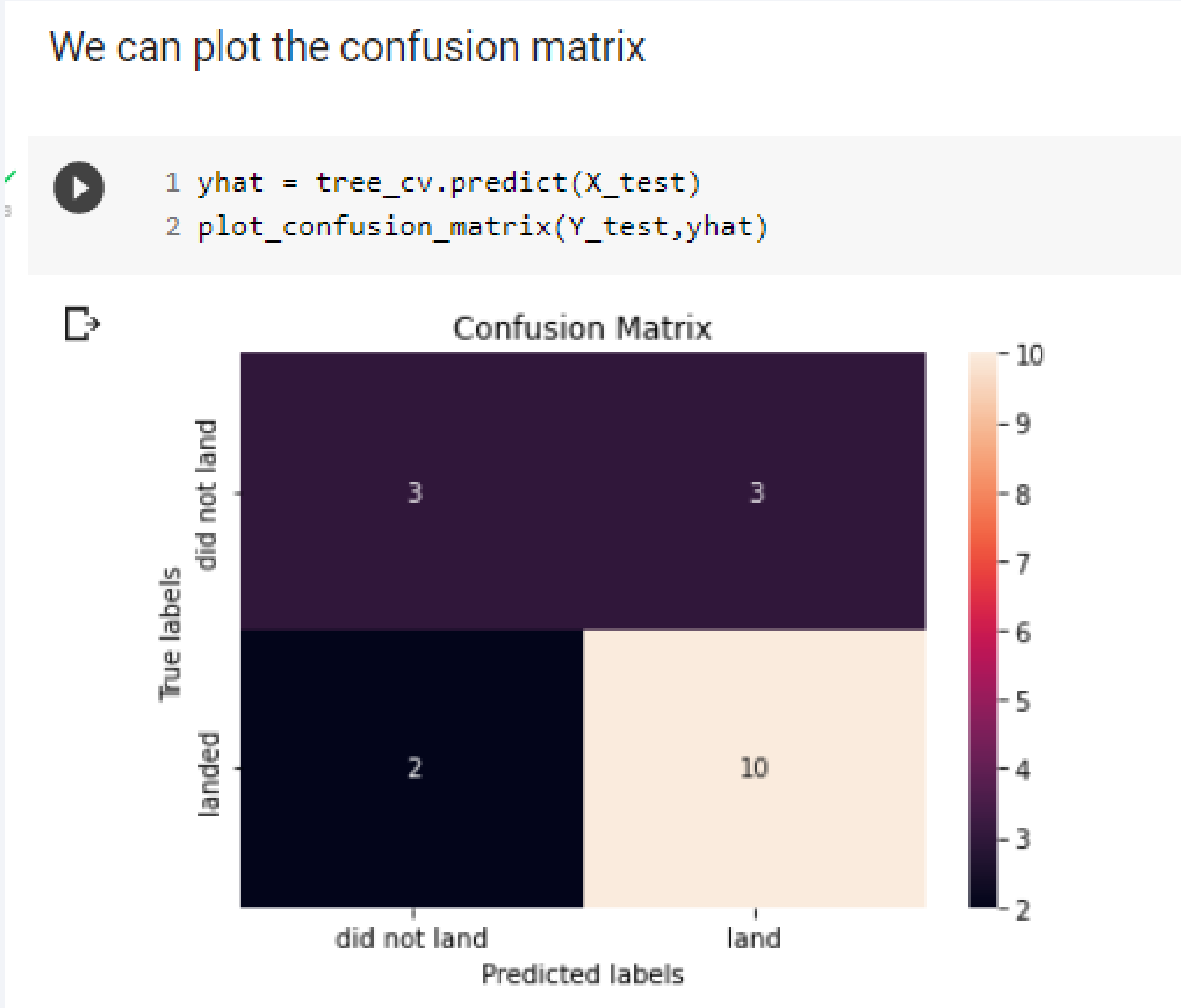


Find the method performs best:

```
predictors = [knn_cv, svm_cv, logreg_cv, tree_cv]
best_predictor = ""
best_result = 0
for predictor in predictors:

    predictor.score(X_test, Y_test)

print("tuned hpyerparameters :(best parameters) ",predictor.best_params_)
print("accuracy :",predictor.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 1,
accuracy : 0.8910714285714285
```

Confusion Matrix



We can plot the confusion matrix

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

Conclusions

- The greater the amount of flying at a launch site, the higher the success rate at a launch site.
- The ES-L1, GEO, HEO, SSO, VLEO orbits had the highest success rate of all.
- Launches started to be successful in 2013 and since then the success rate has only increased.
- KSC LC-39A is the most successful site.
- Decision tree classifier is the best ML algorithm.

Thank you!