IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sining Xu
04/13/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - 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

  The Space X website advertises Falcon 9 rocket launches for a cost of 62 million dollars, which is significantly lower than the upwards of 165 million dollars charged by other providers. This is largely due to the fact that Space X is able to reuse the first stage of the rocket. Therefore, predicting the success of the first stage landing can help determine the cost of a launch. This information can be valuable for other companies bidding against Space X for a rocket launch. The aim of this project is to develop a machine learning pipeline that can accurately predict the success of the first stage landing.

- 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
- 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

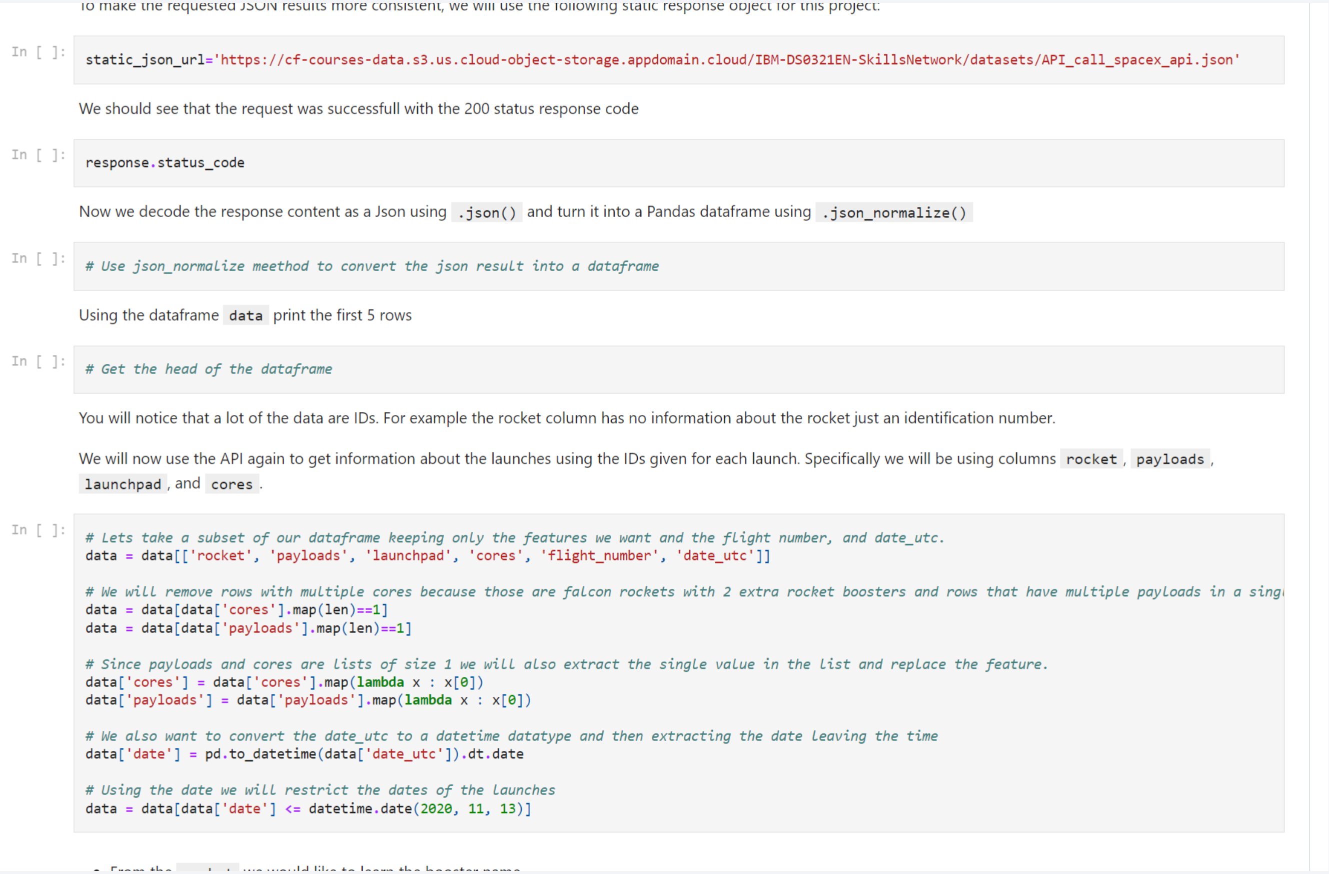
Data Collection

- Describe how data sets were collected.
- Various methods were used to collect the data, including sending get requests to the SpaceX API. The response content was decoded as a Json using the .json() function call and converted to a pandas dataframe using the .json_normalize() method. The data was then cleaned, missing values were identified and filled in as necessary. Furthermore, web scraping was conducted on Wikipedia to gather Falcon 9 launch records using BeautifulSoup. The aim was to extract the launch records from an HTML table, parse the table, and convert it to 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/SiningXu/IBM-Data-Scientist-Capstone/blob/main/Data%20Collection%20API.ipynb



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

```
In [ ]: 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

```
In [ ]: response.status_code
```

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

```
In [ ]: # Use json_normalize meethod to convert the json result into a dataframe
```

Using the dataframe `data` print the first 5 rows

```
In [ ]: # Get the head of the dataframe
```

You will notice that a lot of the data are IDs. For example the rocket column has no information about the rocket just an identification number.

We will now use the API again to get information about the launches using the IDs given for each launch. Specifically we will be using columns `rocket`, `payloads`, `launchpad`, and `cores`.

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

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

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

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

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

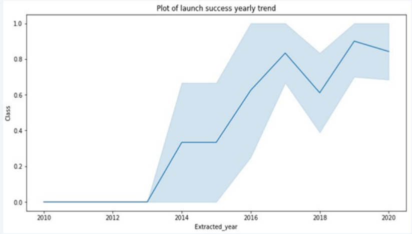
- From the we would like to learn the booster name

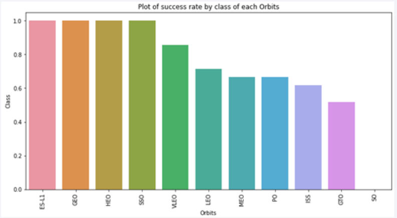
Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/SiningXu/IBM-Data-Scientist-Capstone/blob/main/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.
- Link: https://github.com/SiningXu/IBM-Data-Scientist-Capstone/blob/main/Complete%20the%20EDA%20with%20Visualization%20lab.i





EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- 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 the notebook is https://github.com/SiningXu/IBM-Data-Scientist-Capstone/blob/main/Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to
- mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

- Link: https://github.com/SiningXu/IBM-Data-Scientist-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

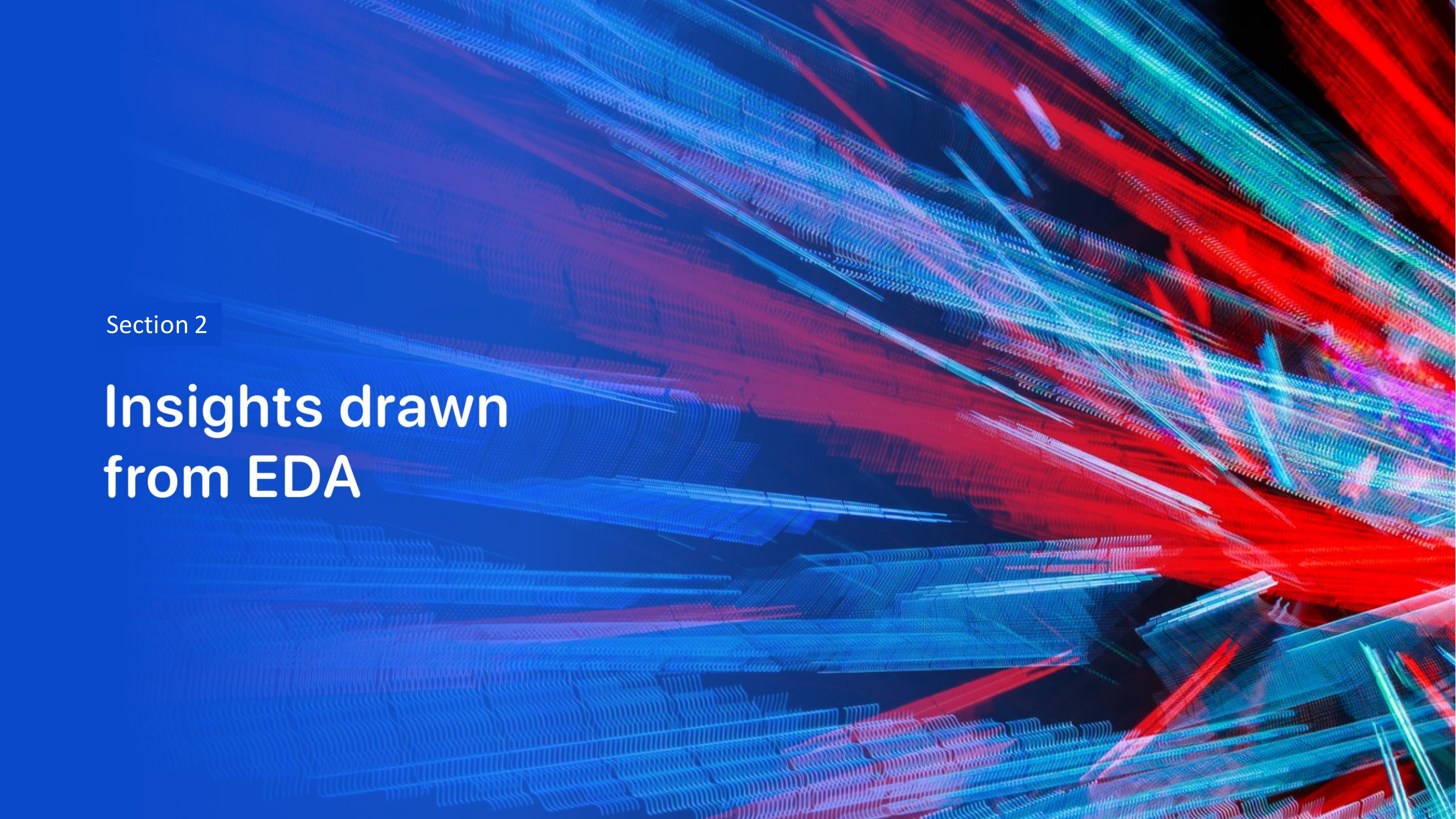
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.

Results

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

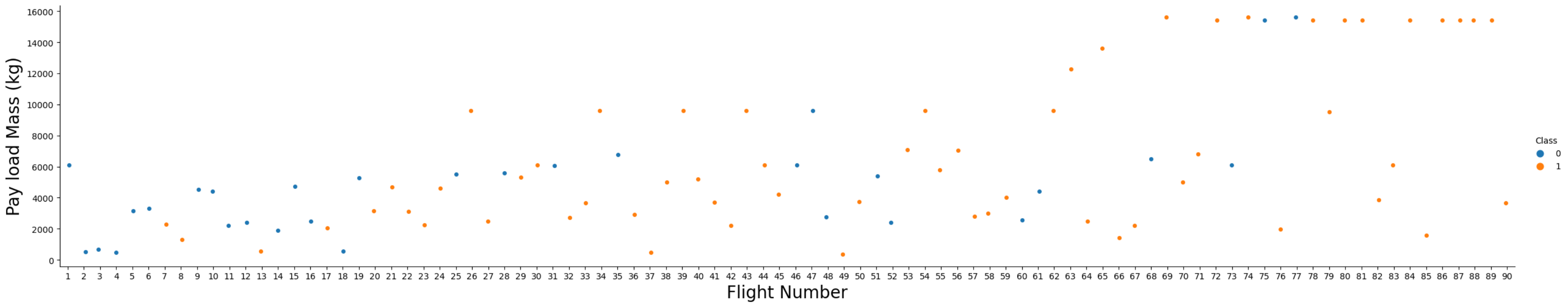


Section 2

Insights drawn from EDA

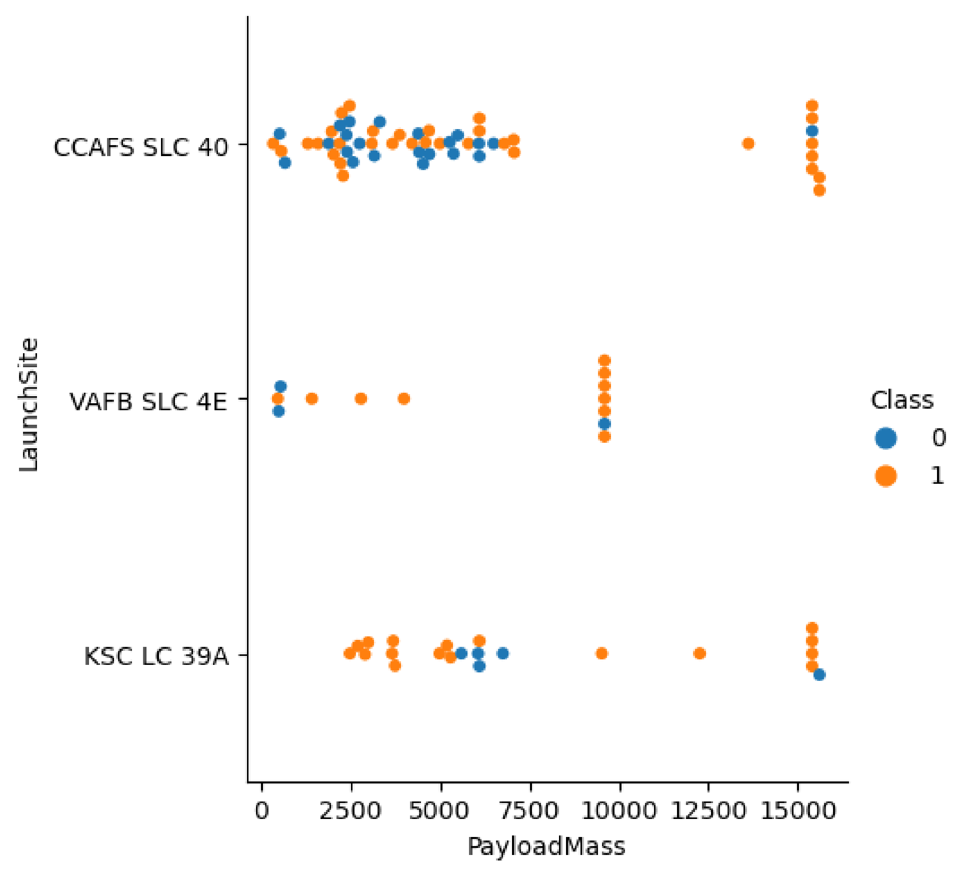
Flight Number vs. Launch Site

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



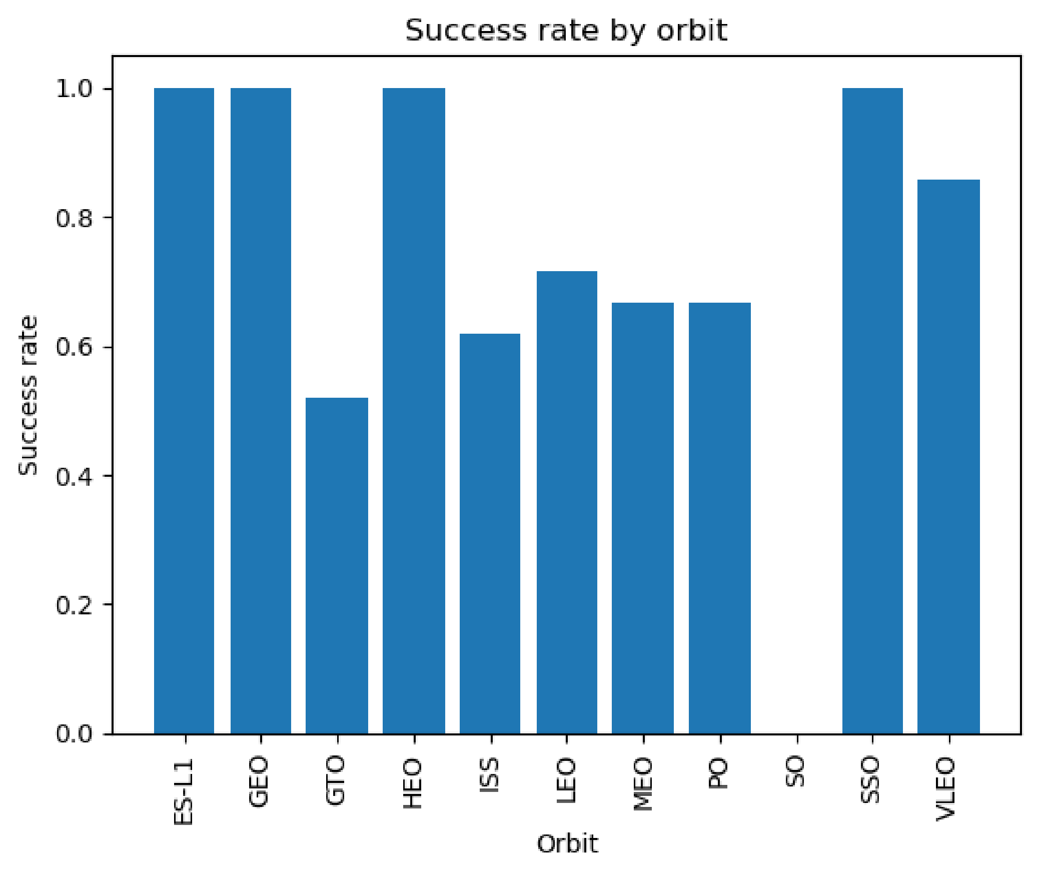
Payload vs. Launch Site

- Payload vs. Launch Site



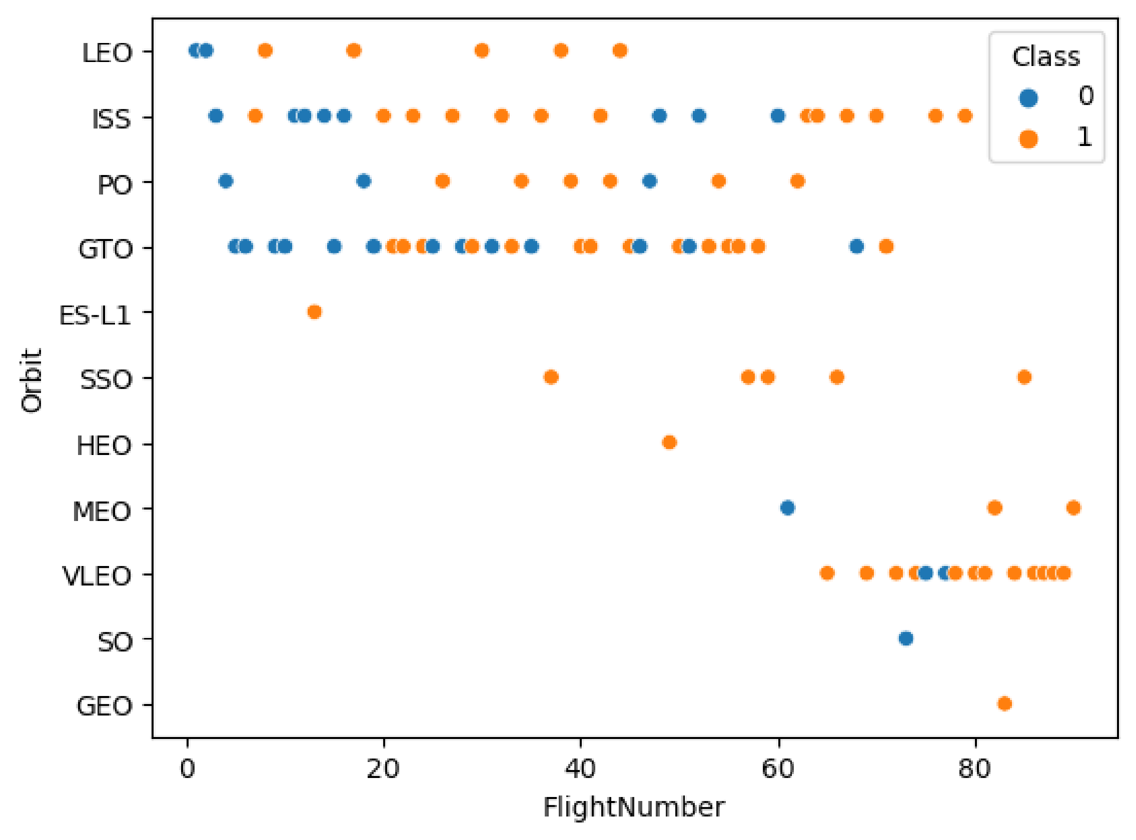
Success Rate vs. Orbit Type

- A bar chart for the success rate of each orbit type



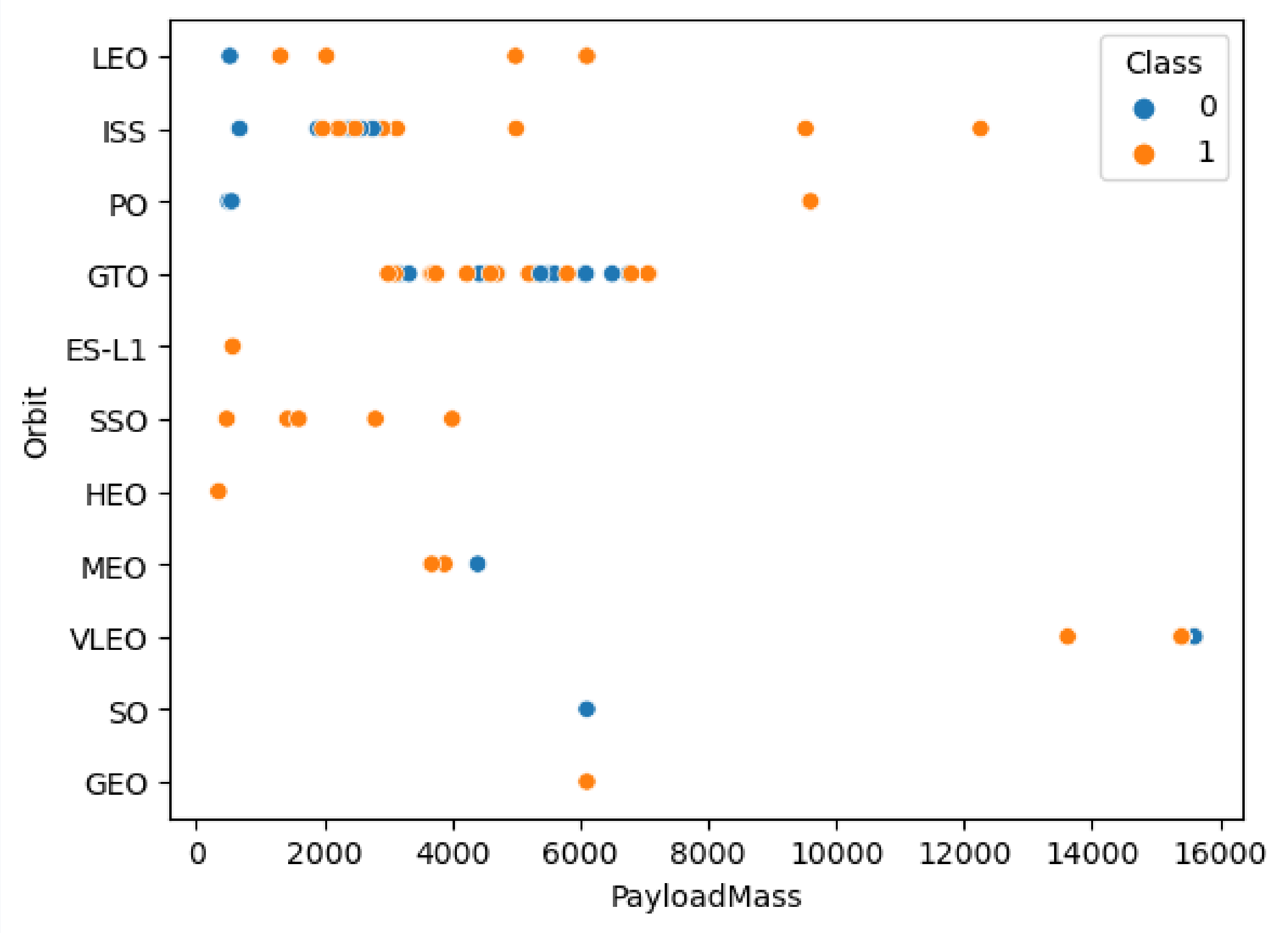
Flight Number vs. Orbit Type

- A scatter point of Flight number vs. Orbit type



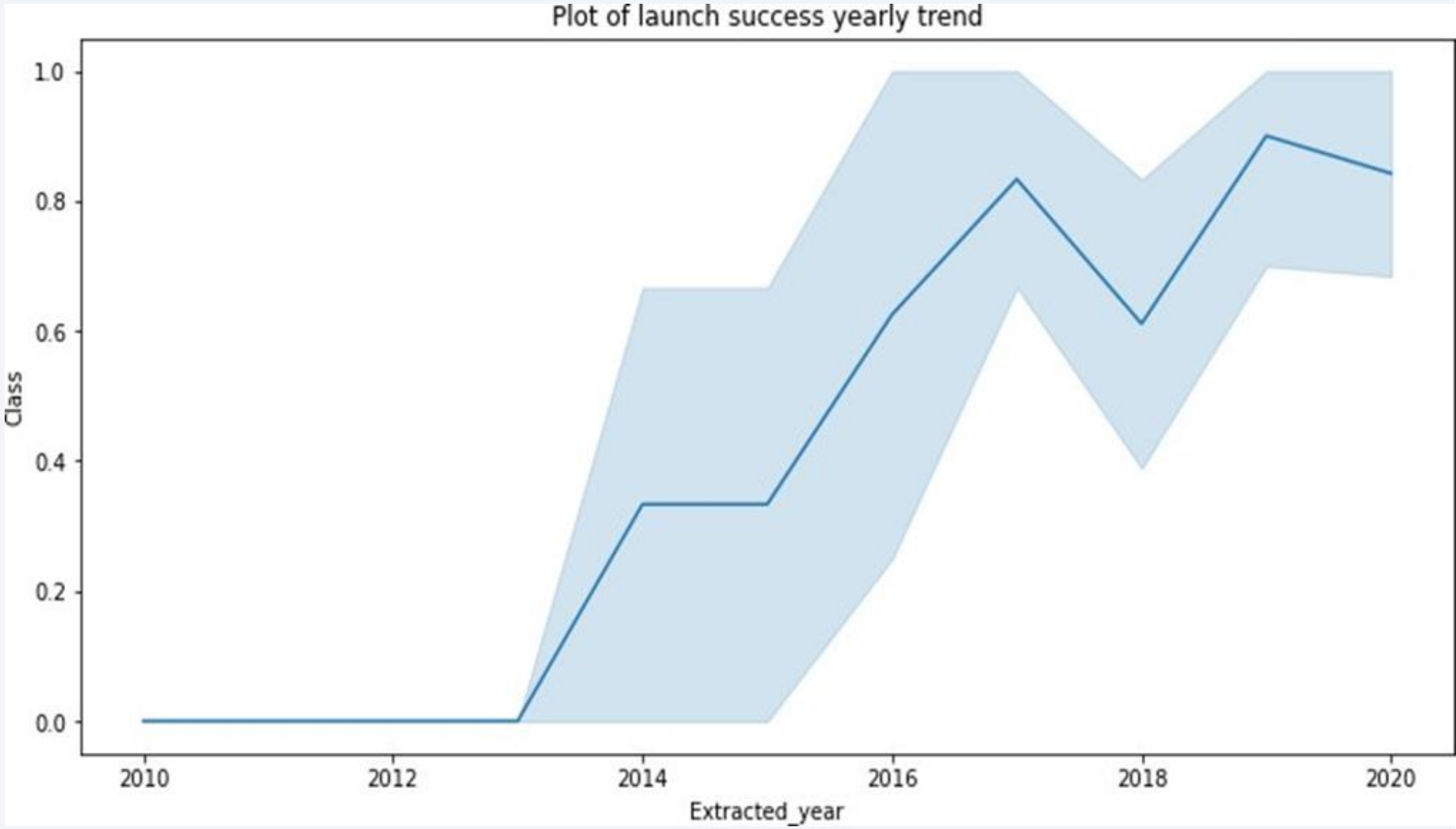
Payload vs. Orbit Type

- A scatter point of payload vs. orbit type



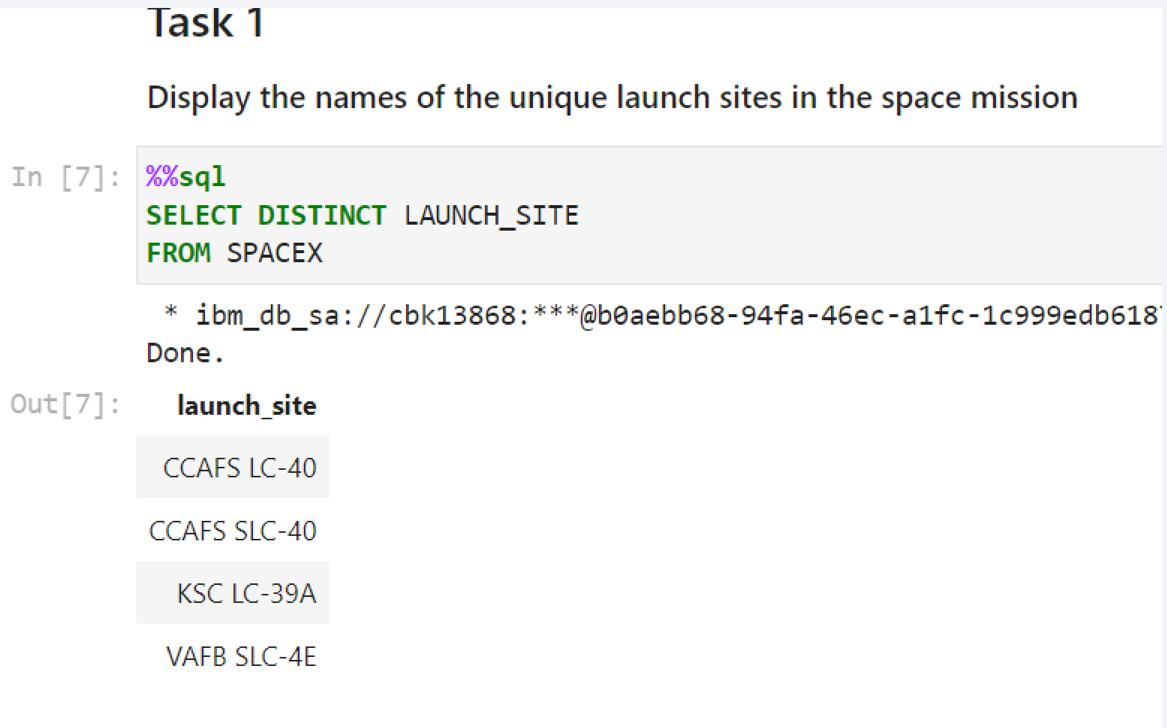
Launch Success Yearly Trend

- A line chart of yearly average success rate



All Launch Site Names

- the names of the unique launch sites



Task 1

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

In [7]:
```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEX
```

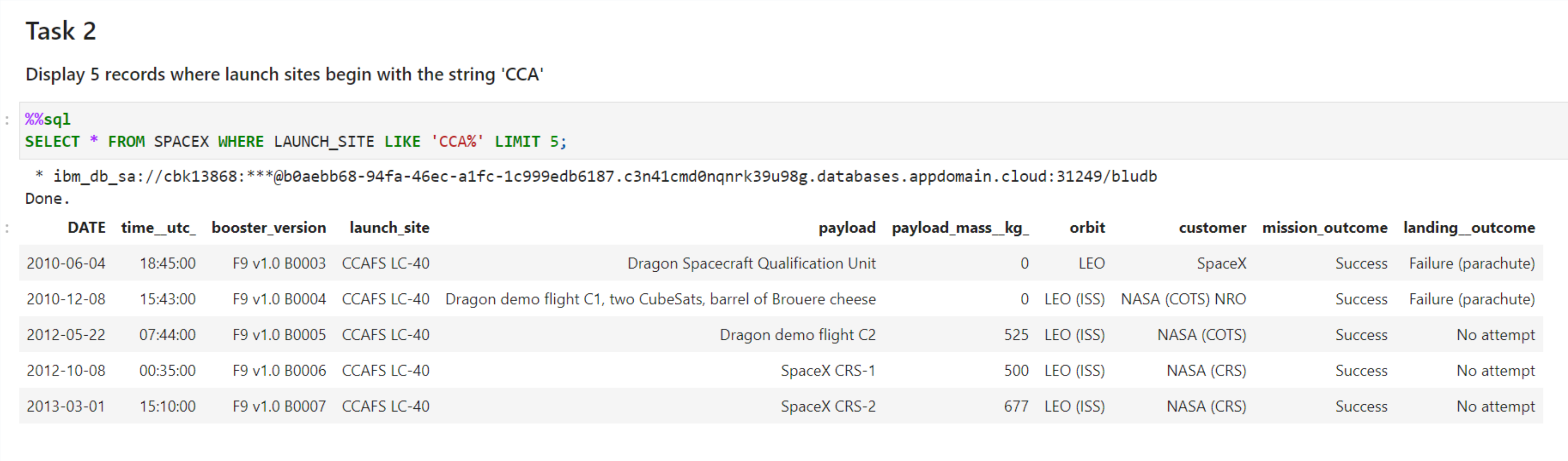
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb618
Done.
```

Out[7]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`



Task 2

Display 5 records where launch sites begin with the string 'CCA'

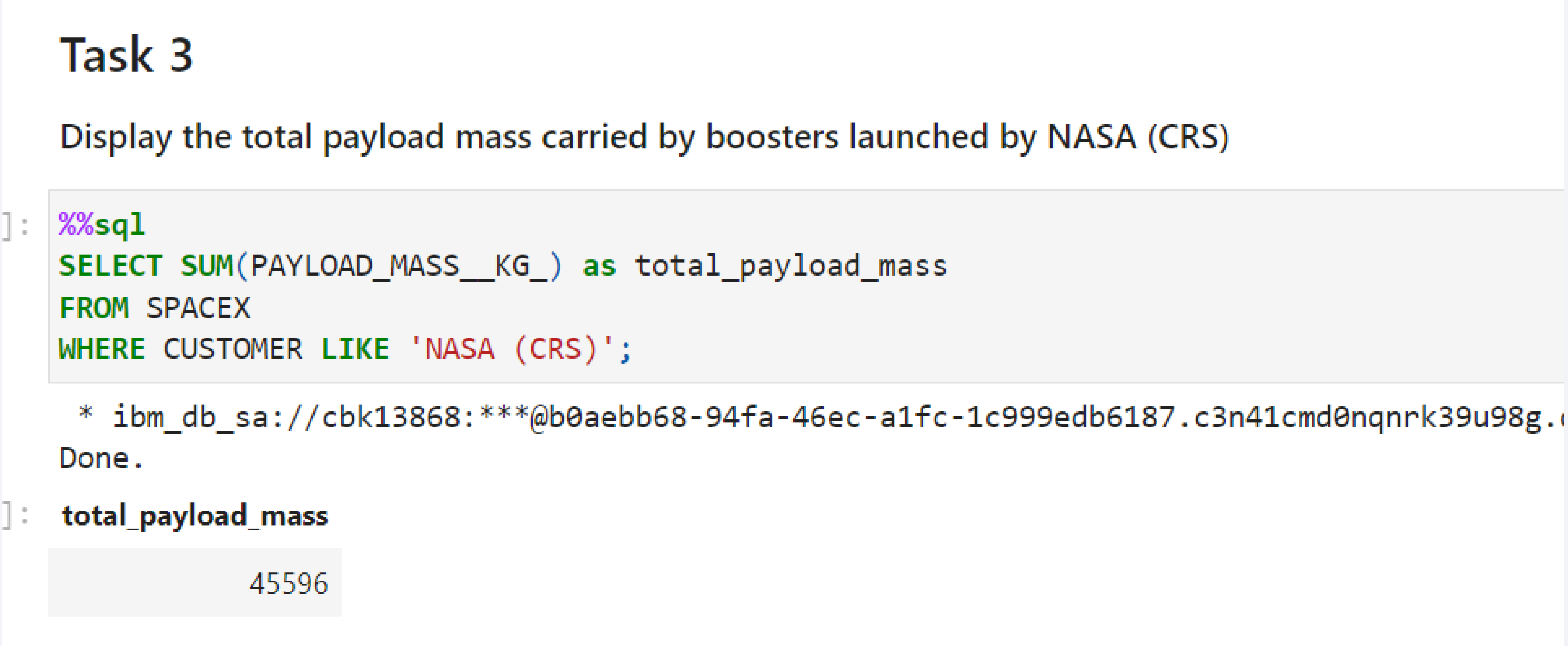
```
%%sql
SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



Task 3

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

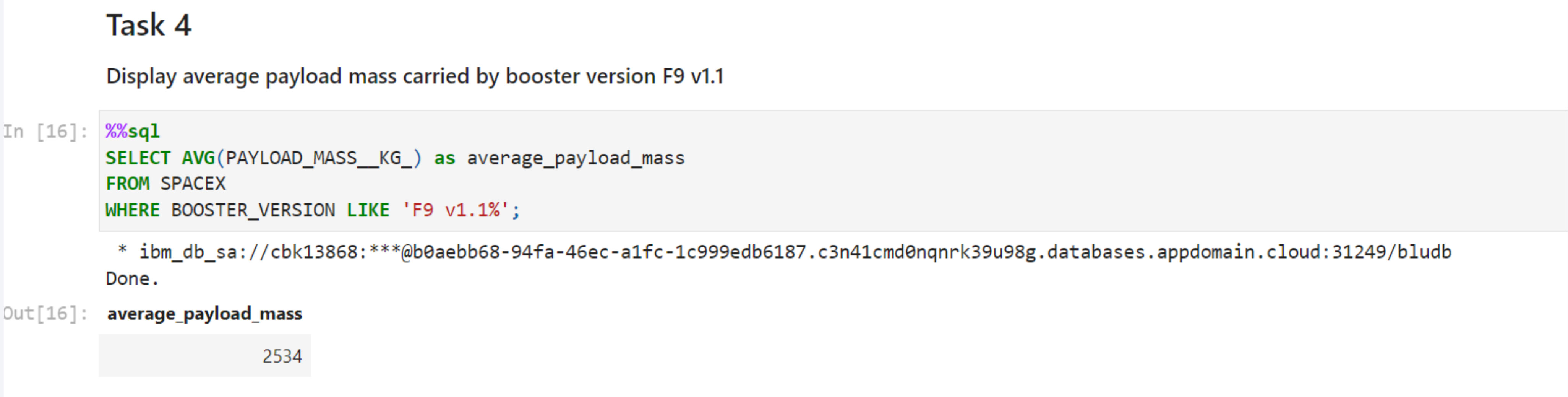
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) as total_payload_mass
FROM SPACEX
WHERE CUSTOMER LIKE 'NASA (CRS)';
```

```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1



Task 4

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

In [16]:

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) as average_payload_mass
FROM SPACEX
WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

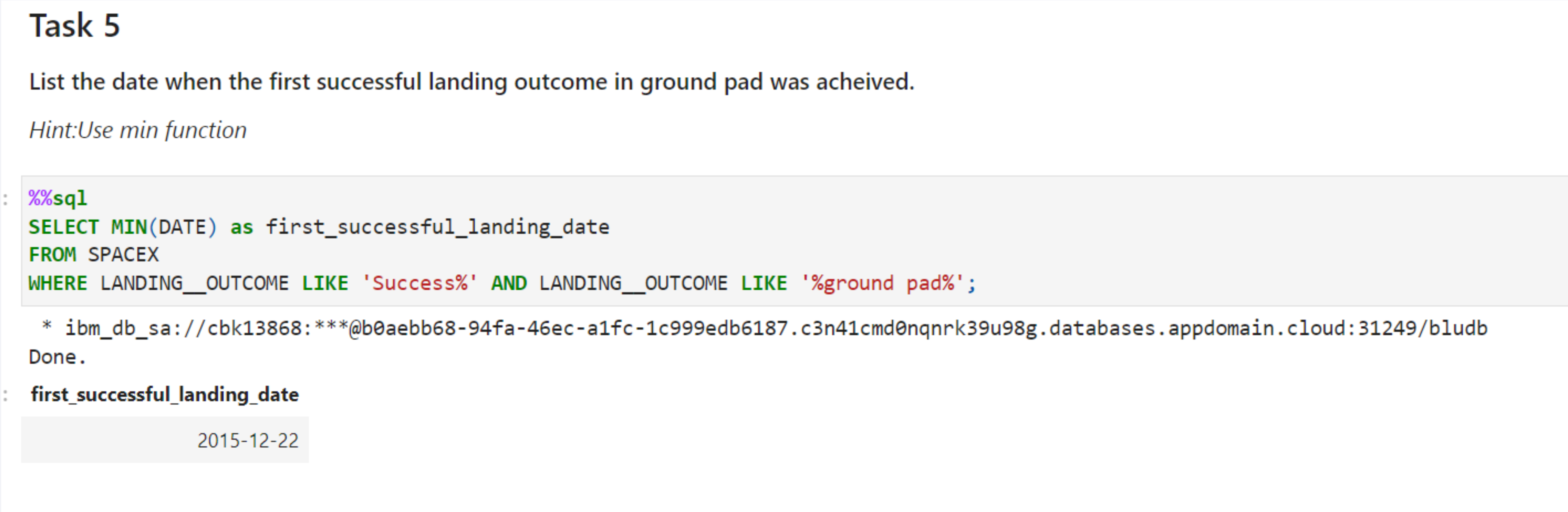
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[16]:

average_payload_mass
2534

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



Task 5

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

Hint:Use min function

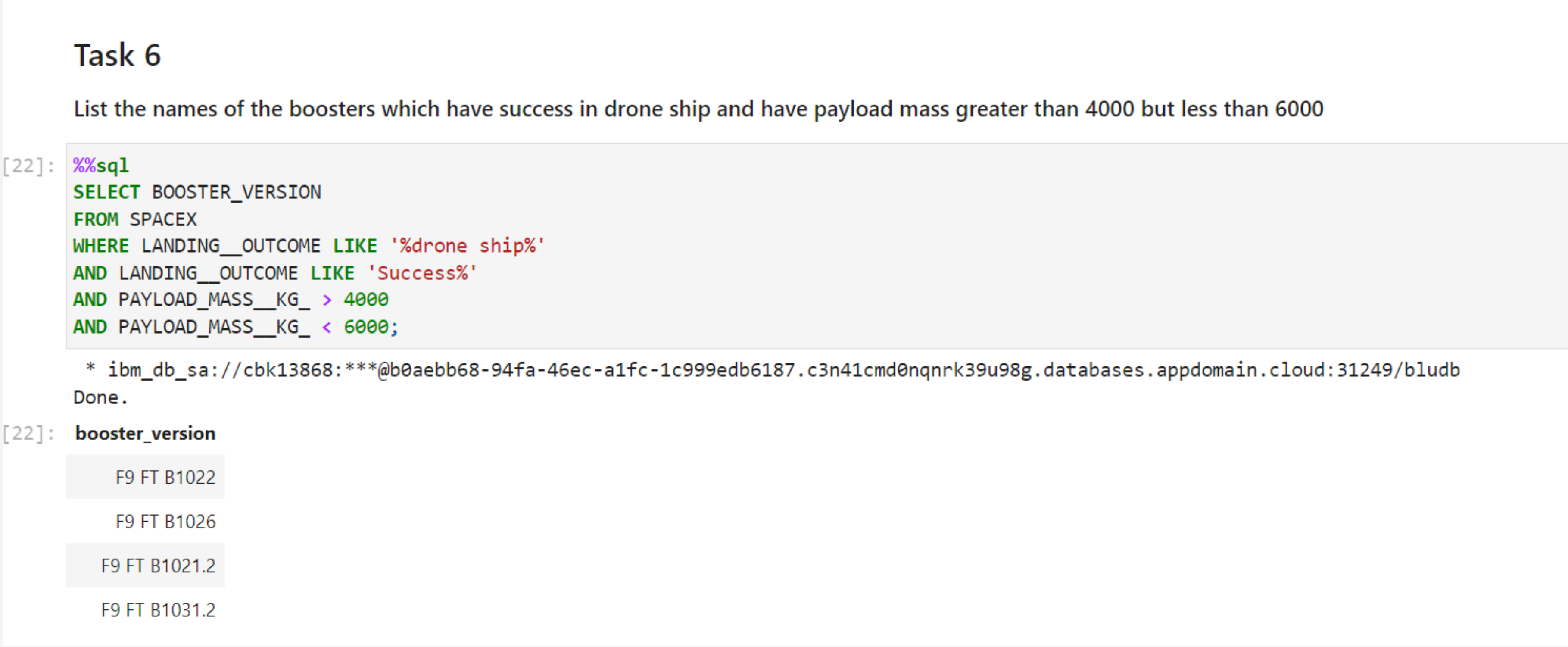
```
%%sql
SELECT MIN(DATE) as first_successful_landing_date
FROM SPACEX
WHERE LANDING__OUTCOME LIKE 'Success%' AND LANDING__OUTCOME LIKE '%ground pad%';
```

```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



Task 6

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

[22]:

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEX
WHERE LANDING__OUTCOME LIKE '%drone ship%'
AND LANDING__OUTCOME LIKE 'Success%'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000;
```

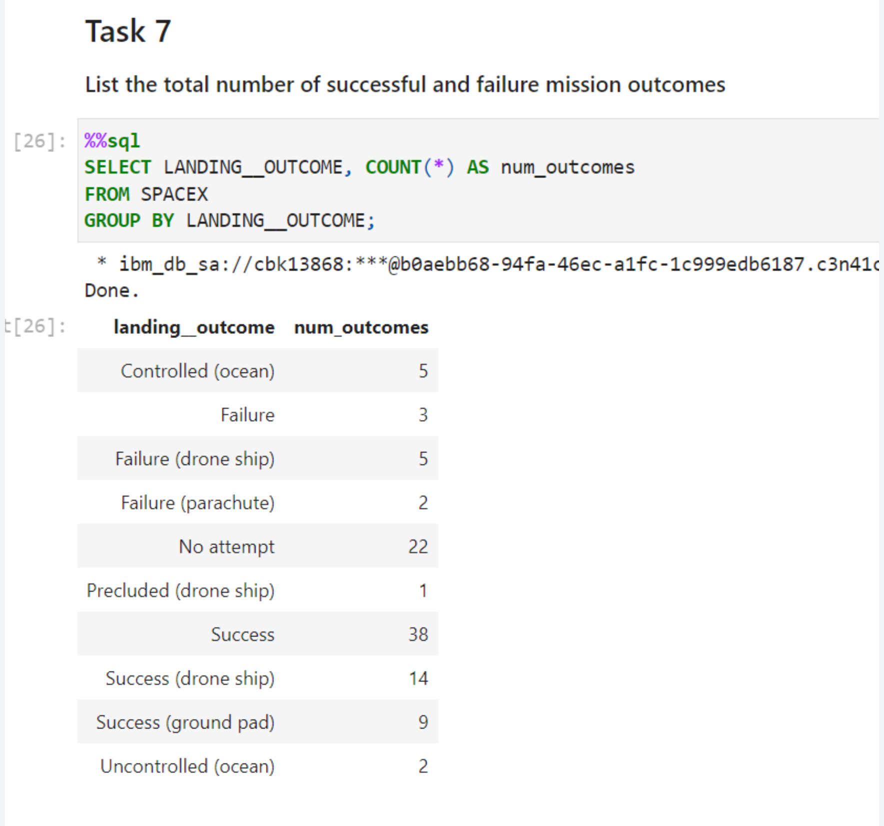
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

[22]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



Task 7

List the total number of successful and failure mission outcomes

```
[26]: %%sql
      SELECT LANDING__OUTCOME, COUNT(*) AS num_outcomes
      FROM SPACEX
      GROUP BY LANDING__OUTCOME;
```

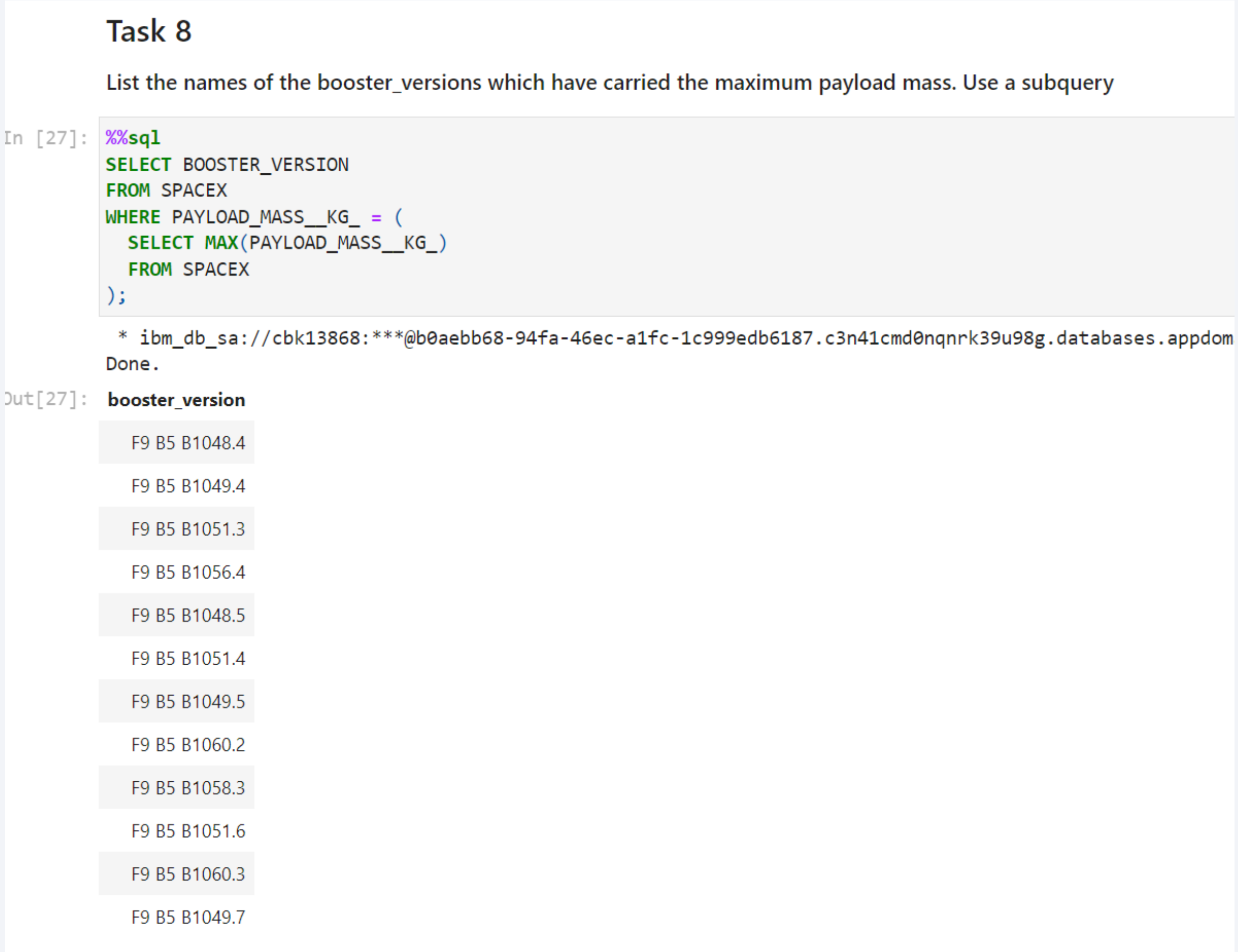
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41c
Done.
```

t[26]:

landing__outcome	num_outcomes
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	22
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



Task 8

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

In [27]:

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEX
WHERE PAYLOAD_MASS__KG_ = (
  SELECT MAX(PAYLOAD_MASS__KG_)
  FROM SPACEX
);
```

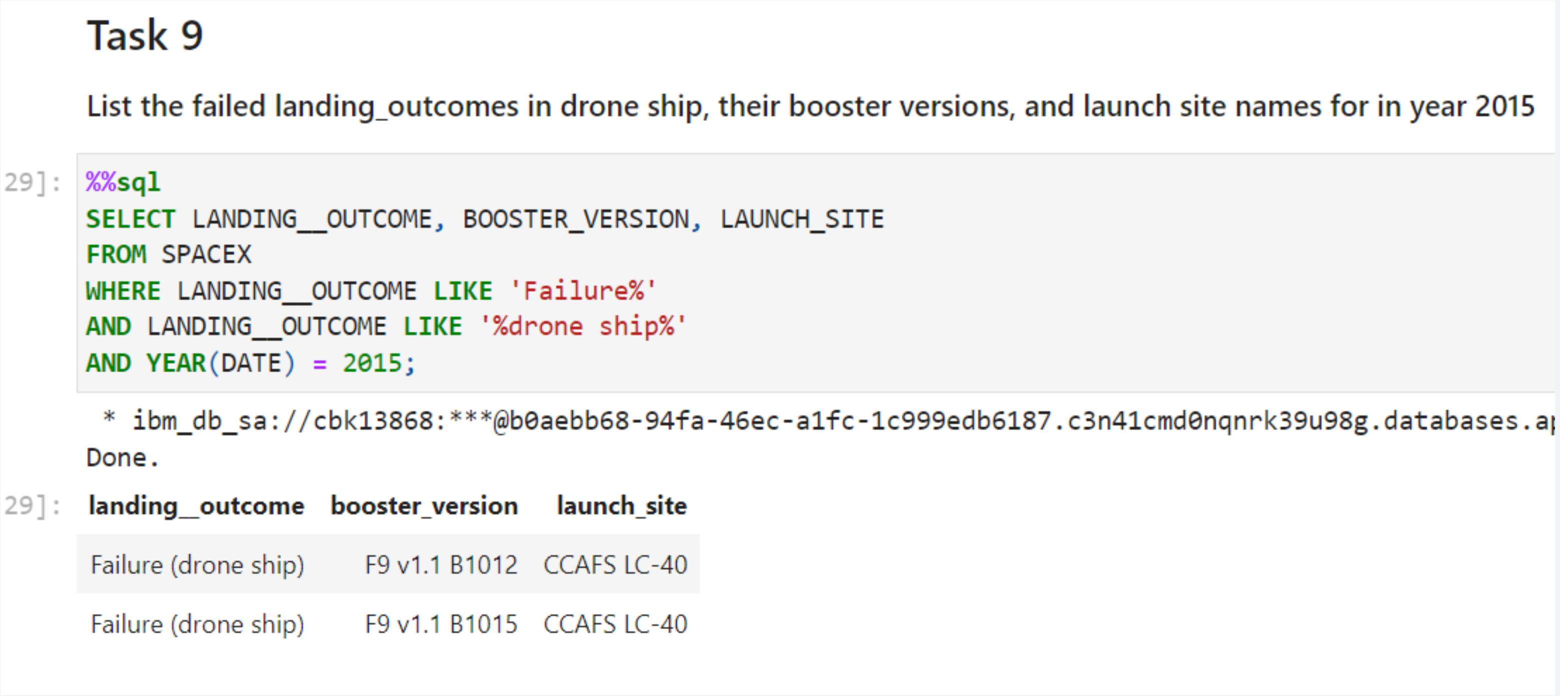
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdom
Done.
```

Out[27]:

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



Task 9

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

```
29]: %%sql
     SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
     FROM SPACEX
     WHERE LANDING__OUTCOME LIKE 'Failure%'
     AND LANDING__OUTCOME LIKE '%drone ship%'
     AND YEAR(DATE) = 2015;
```

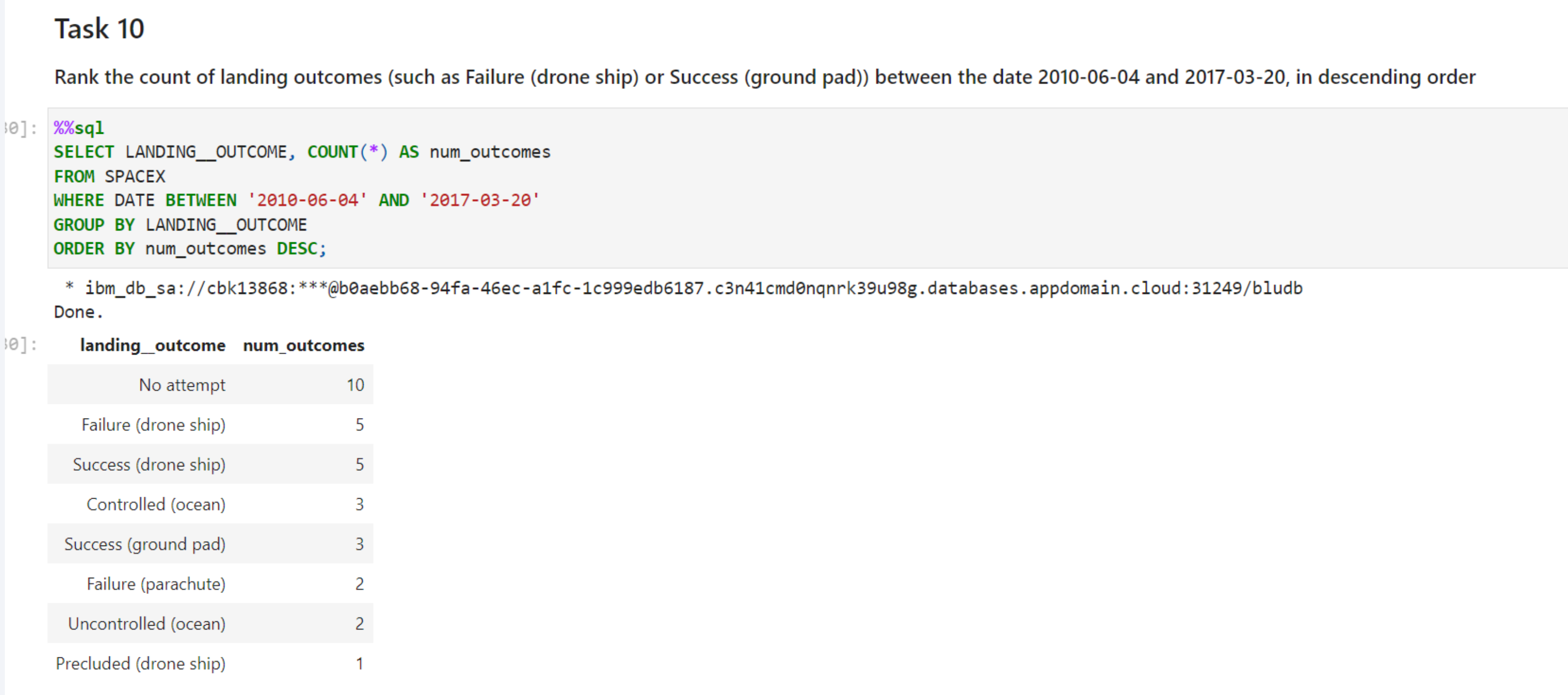
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.a
Done.
```

29]:

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

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



Task 10

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 num_outcomes
FROM SPACEX
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY num_outcomes DESC;
```

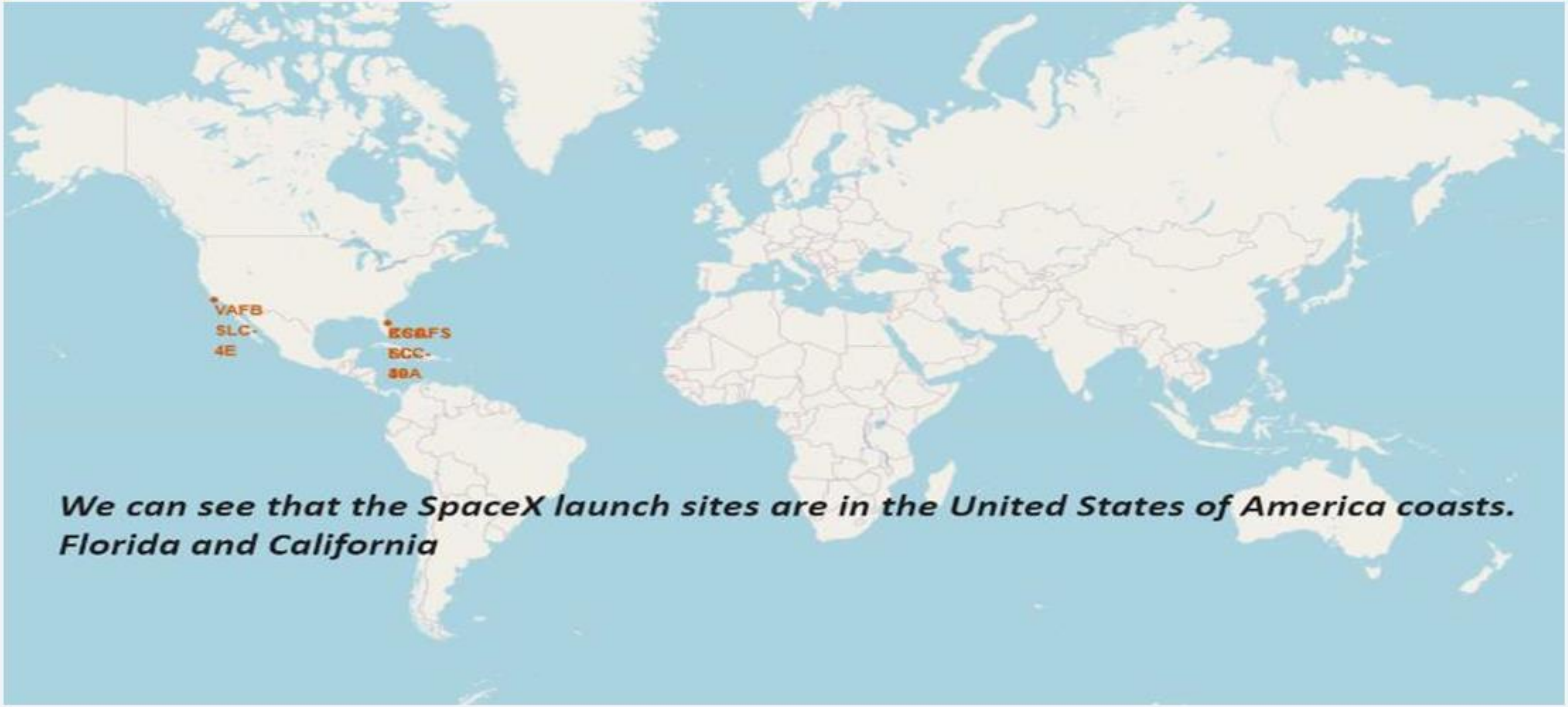
```
 * ibm_db_sa://cbk13868:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

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

Section 3

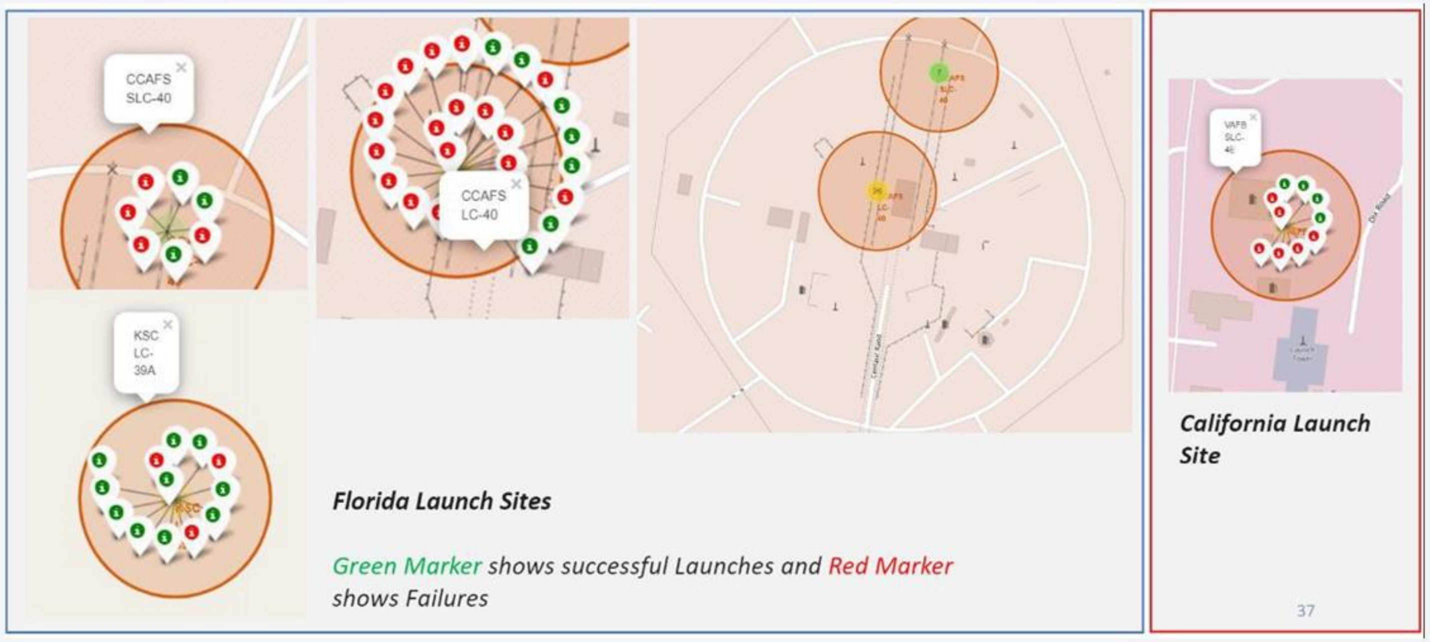
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>



We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

<Folium Map Screenshot 2>

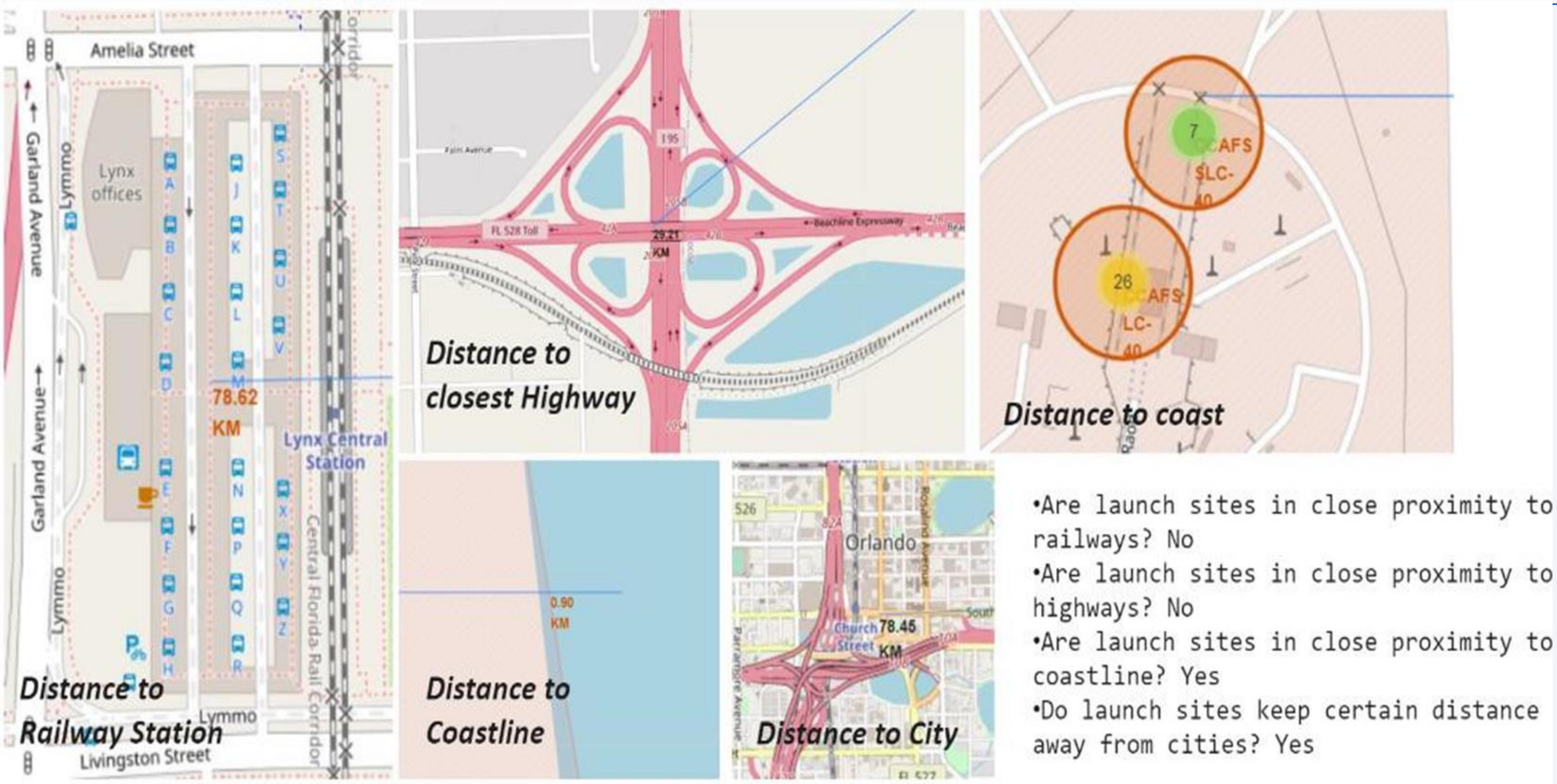


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

<Folium Map Screenshot 3>



Distance to Railway Station

Distance to closest Highway

Distance to coast

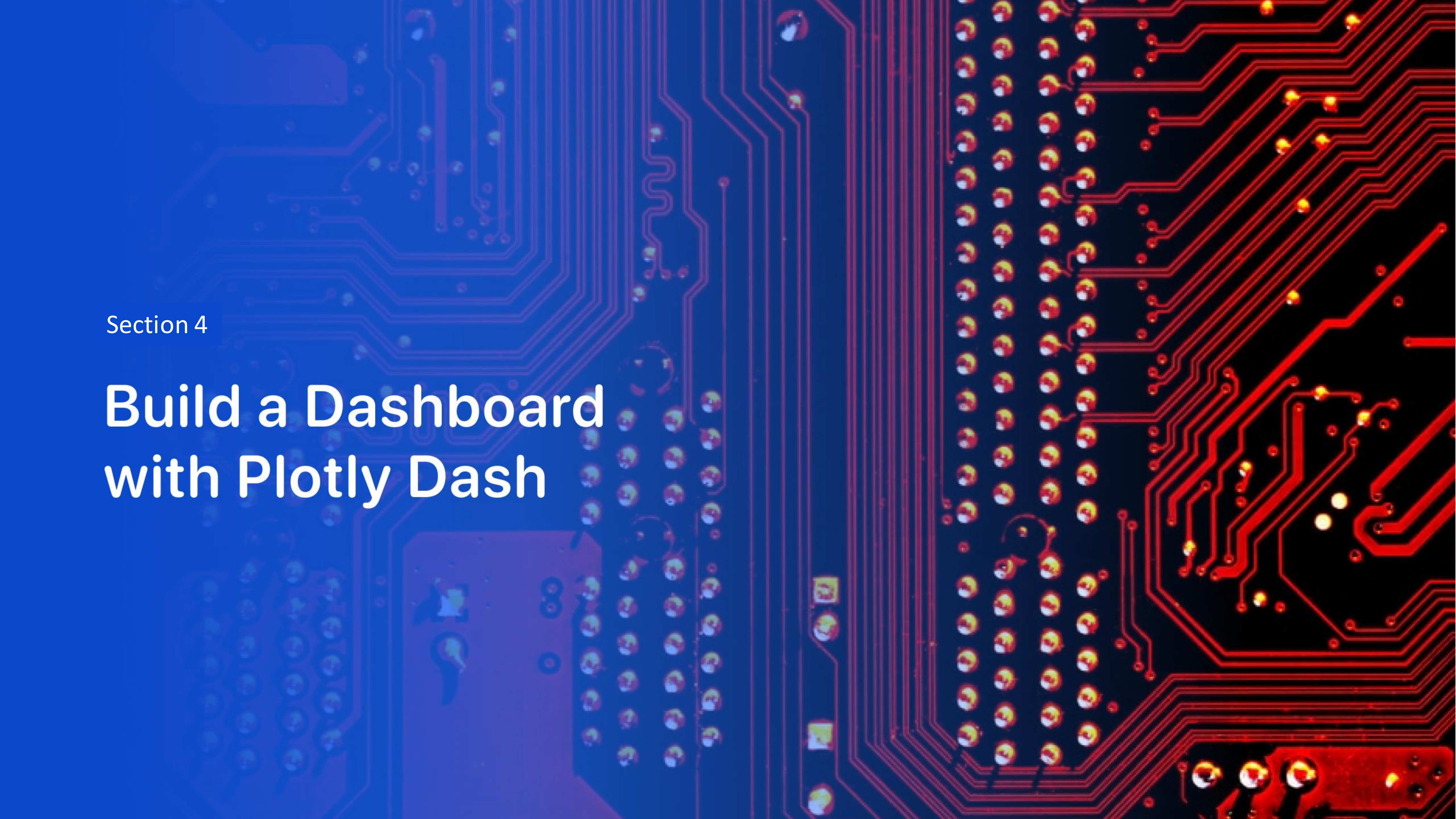
Distance to Coastline

Distance to City

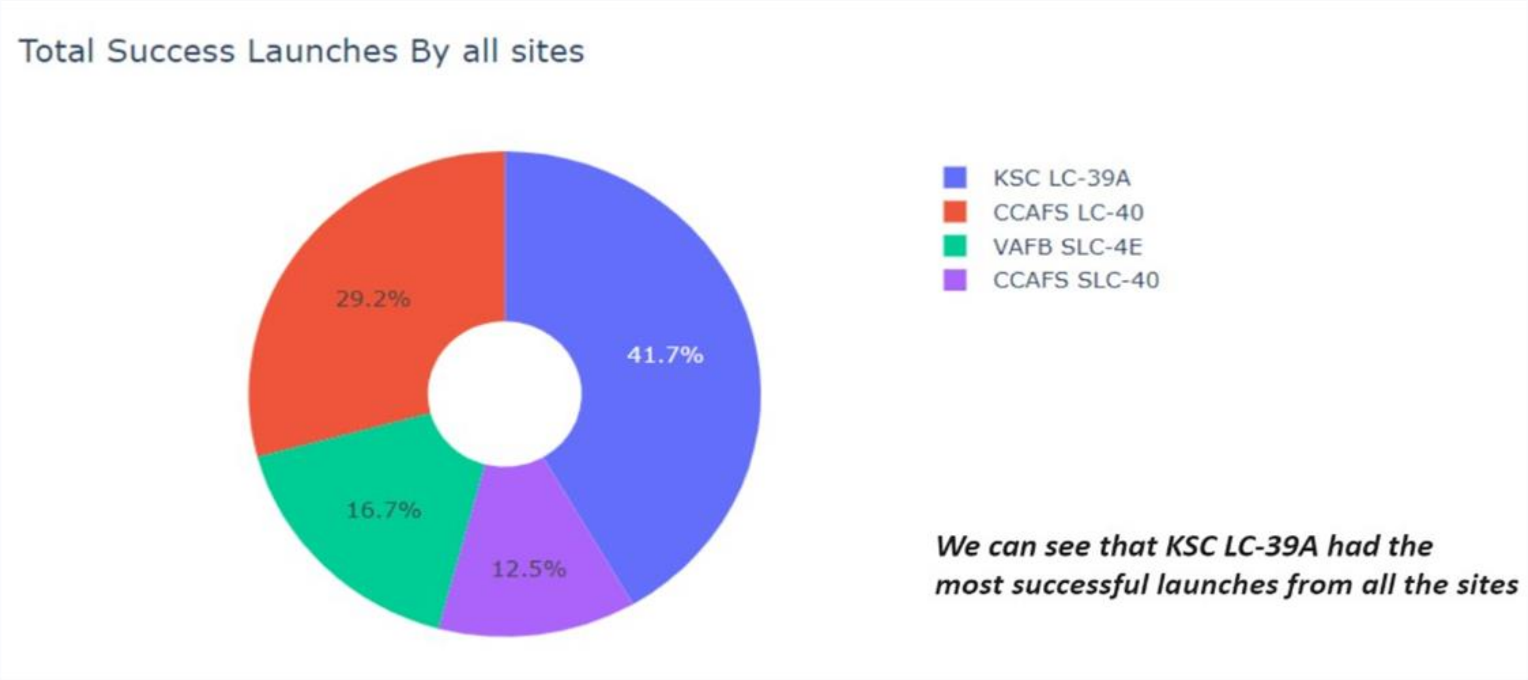
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

Build a Dashboard with Plotly Dash

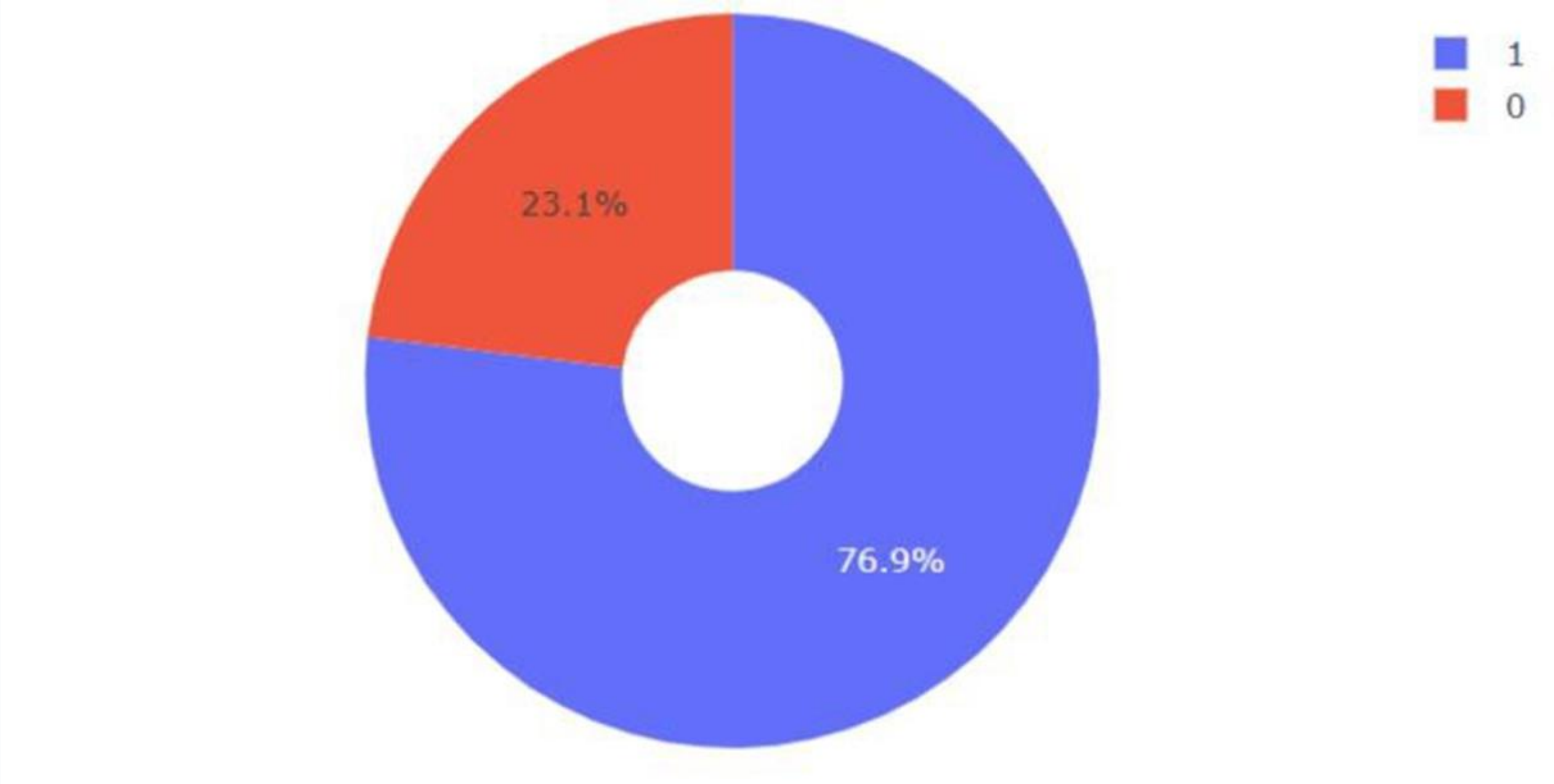


<Dashboard Screenshot 1>



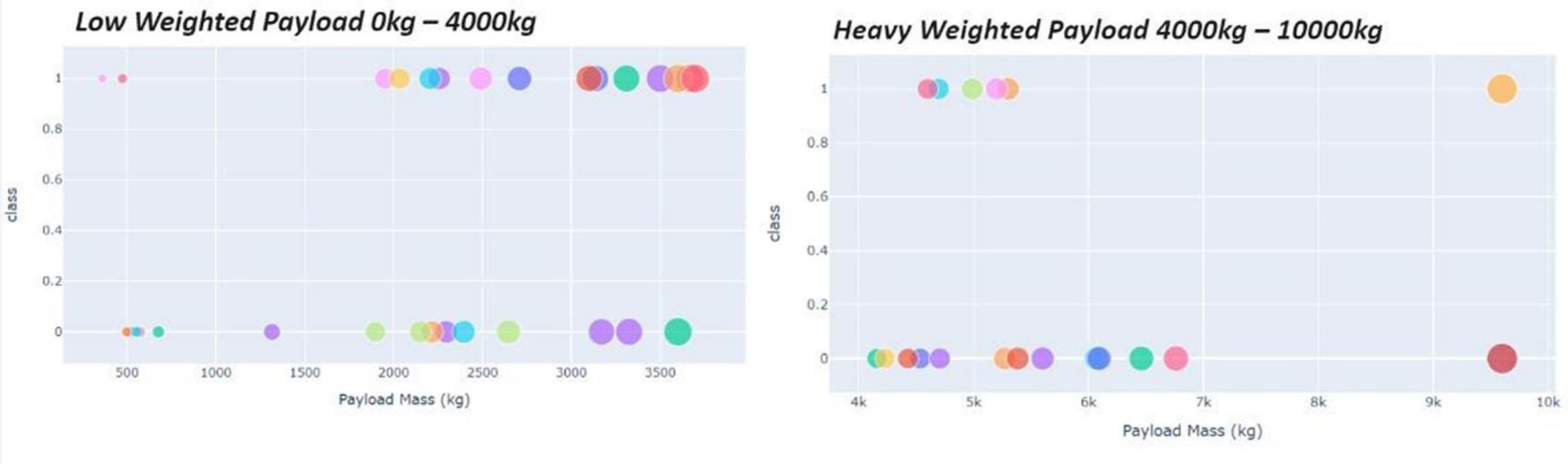
We can see that KSC LC-39A had the most successful launches from all the sites

<Dashboard Screenshot 2>



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Dashboard Screenshot 3>



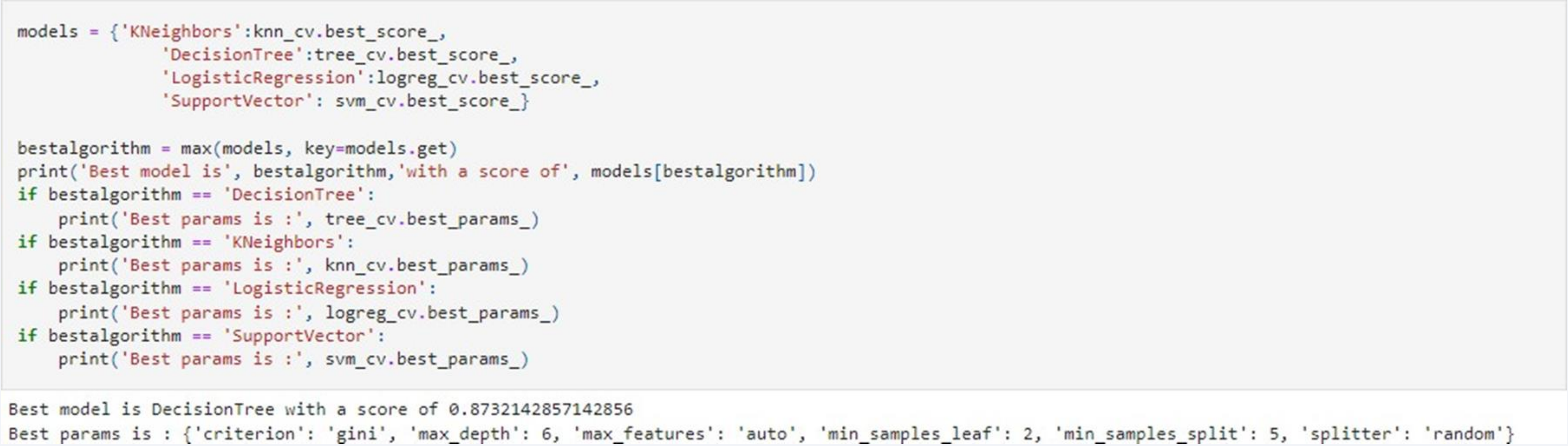
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy



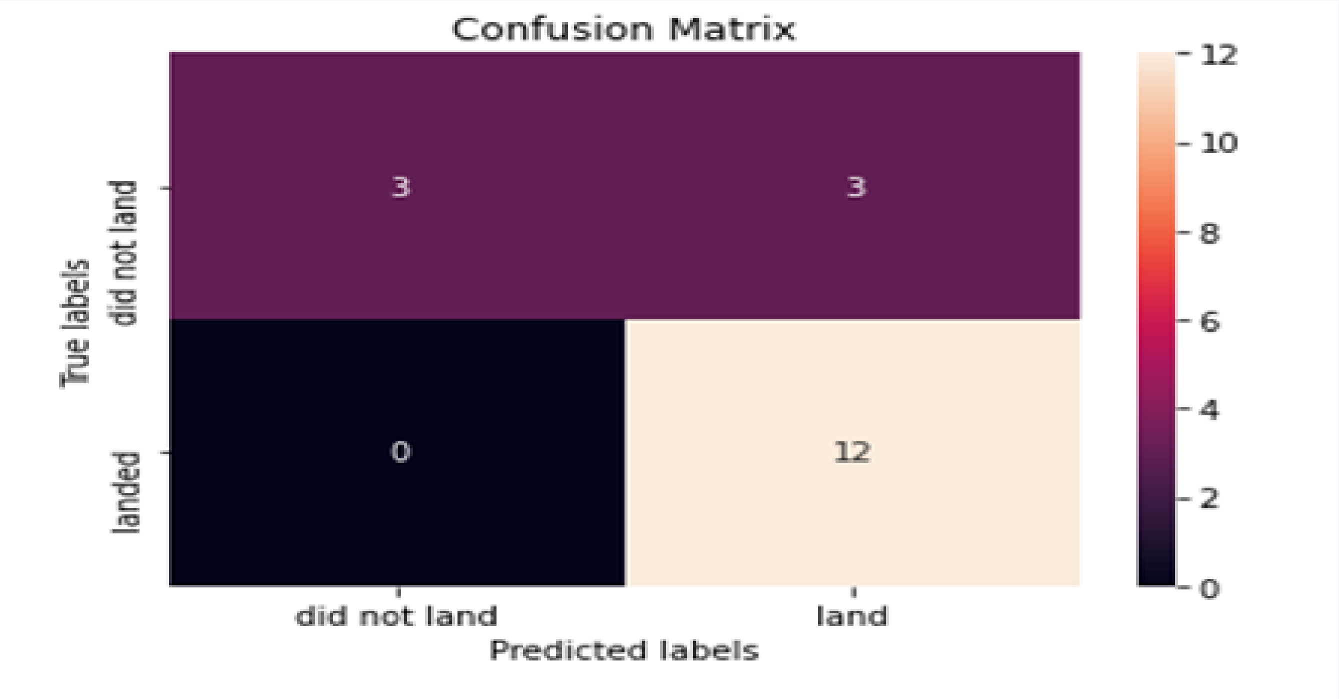
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

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 flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!