IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Lukas O.>
<02.11.2021>

Outline

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

Executive Summary

- **Summary of methodologies:**
  - Data collection/ data wrangling
  - EDA with data visualization and SQL
  - Interactive Folium map and a plotly dashboard
  - Predicative analysis with classification ML algorithms
- **Summary of all results**

  EDA results and Interactive analytics demo:

  - Which features of the dataset are valid for the classification task?

  Preductive analysis results:

  - Can we predict the success of the landing of the first stage of the Falcon 9 rocket?

Introduction

- **Project background and context**
  - The commercial space age is here, companies are making space travel affordable for everyone. SpaceX, one popular player, advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- **Problems you want to find answers:**
  - Determine the price of each launch by gathering information about Space X and creating dashboards.
  - Determine if SpaceX will reuse the first stage.
  - Train a machine learning models instead of using rocket science and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

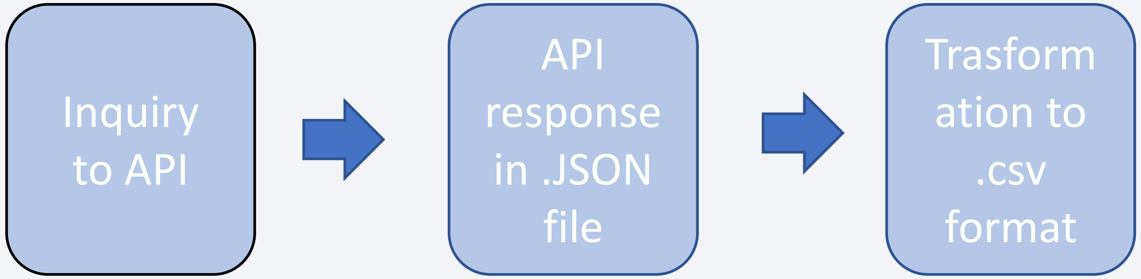
- Data collection methodology:
    - SpaceX Rest API and web scraping from Wikipedia
- Perform data wrangling:
    - One-hot-encoding for categorical features + dropping of irrelevant featurrs
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Build and finetune the hyperparameters of multiple ML classification models
    - Evaluate them on our data to determine the best model

Data Collection

1. The launch data was gathered from the SpaceX Rest API
    - The API returns information about the used rocket, its payload, launch and landing specifications and the overall landing outcome.

2. A second data source for Falcon 9 launch data was provided by webscraping Wikipedia using the Beautiful Soup Package

SpaceX API

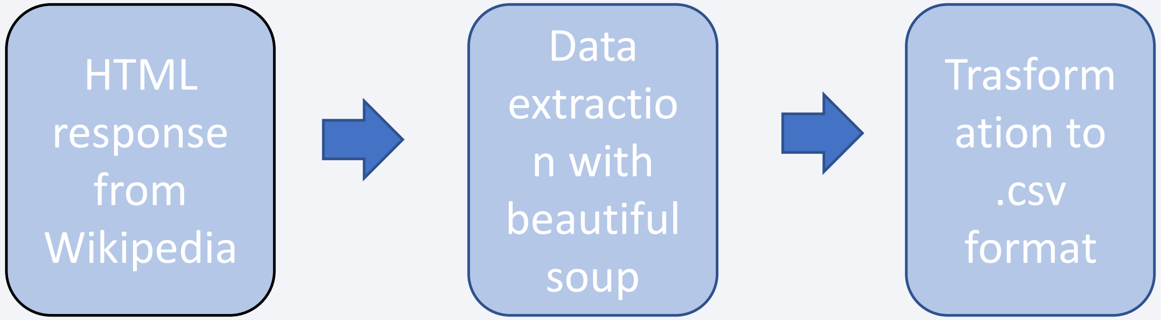


Inquiry to API

API response in .JSON file

Trasformation to .csv format

Web scraping



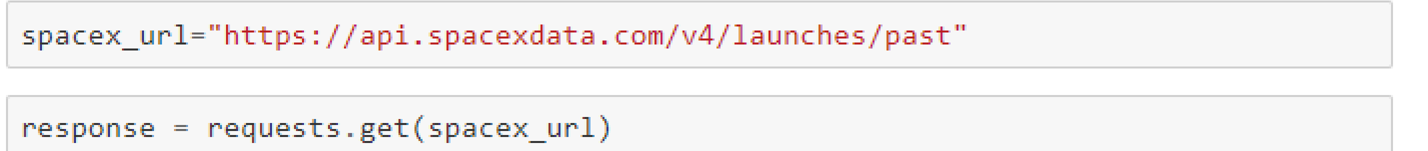
HTML response from Wikipedia

Data extraction with beautiful soup

Trasformation to .csv format

Data Collection – SpaceX API (link)

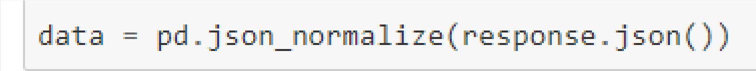
1. Getting response from API:



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

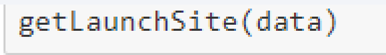
```
response = requests.get(spacex_url)
```

2. Decode response as Json and convert it to Pandas dataframe:

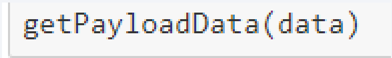


```
data = pd.json_normalize(response.json())
```

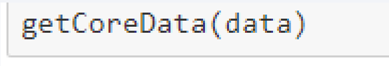
3. Apply provided functions to data to reformat it:



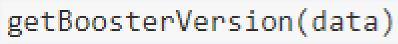
```
getLaunchSite(data)
```



```
getPayloadData(data)
```

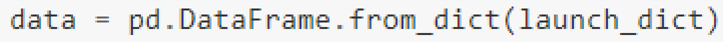


```
getCoreData(data)
```



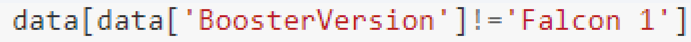
```
getBoosterVersion(data)
```

4. Combine the columns using a label dictionary:



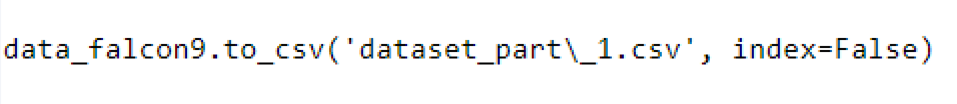
```
data = pd.DataFrame.from_dict(launch_dict)
```

5. Filter dataframe and replace missing values with column mean



```
data[data['BoosterVersion']!='Falcon 1']
```

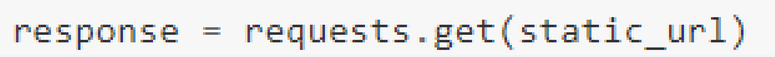
6. Export to CSV file



```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

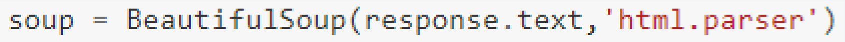
Data Collection – Scraping (link)

1. Request HTTP response from Falcon9 Launch HTML page



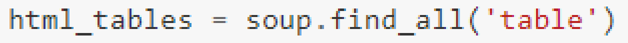
```
response = requests.get(static_url)
```

2. Create beautiful soup object



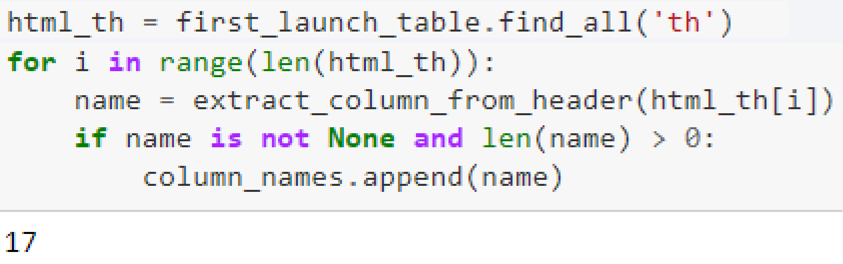
```
soup = BeautifulSoup(response.text,'html.parser')
```

3. Find tables



```
html_tables = soup.find_all('table')
```

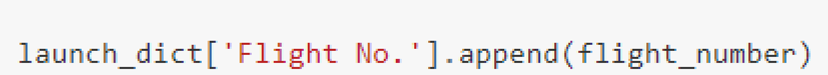
4. Find all column names



```
html_th = first_launch_table.find_all('th')
for i in range(len(html_th)):
    name = extract_column_from_header(html_th[i])
    if name is not None and len(name) > 0:
        column_names.append(name)
```

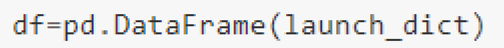
```
17
```

5. Use column name to create dictionary
6. Append data to dictionary



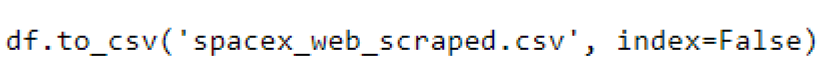
```
launch_dict['Flight No.'].append(flight_number)
```

7. Convert dict. to dataframe



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

8. Convert dataframe to CSV



```
df.to_csv('spacex_web_scraped.csv', index=False)
```

In this lab we will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.

Data Wrangling (link)

- We perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- We the landing outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

EDA with Data Visualization (link)

- We use a number of different plotting methods to get insights into the relationship of different features in our data:

  - **Scatter plots:** Flight Number and Launch Site | Payload and Launch Site | FlightNumber and Orbit type | Payload and Orbit type
  - **Bar plot:** success rate and orbit type
  - **Line plot:** success rate and year

EDA with SQL (link)

- We use the ibm-db-sa interface to evaluate queries on our database. With those we aim to get deeper insights into our data. Specifically, we aim to:

  1. ***Display the names of the unique launch sites in the space mission***
  2. ***Display 5 records where launch sites begin with the string 'CCA'***
  3. ***Display the total payload mass carried by boosters launched by NASA (CRS)***
  4. ***Display average payload mass carried by booster version F9 v1.1***
  5. ***List the date when the first successful landing outcome in ground pad was acheived.***
  6. ***List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000***
  7. ***List the total number of successful and failure mission outcomes***
  8. ***List the names of the booster_versions which have carried the maximum payload mass. Use a subquery***
  9. ***List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015***
  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***

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

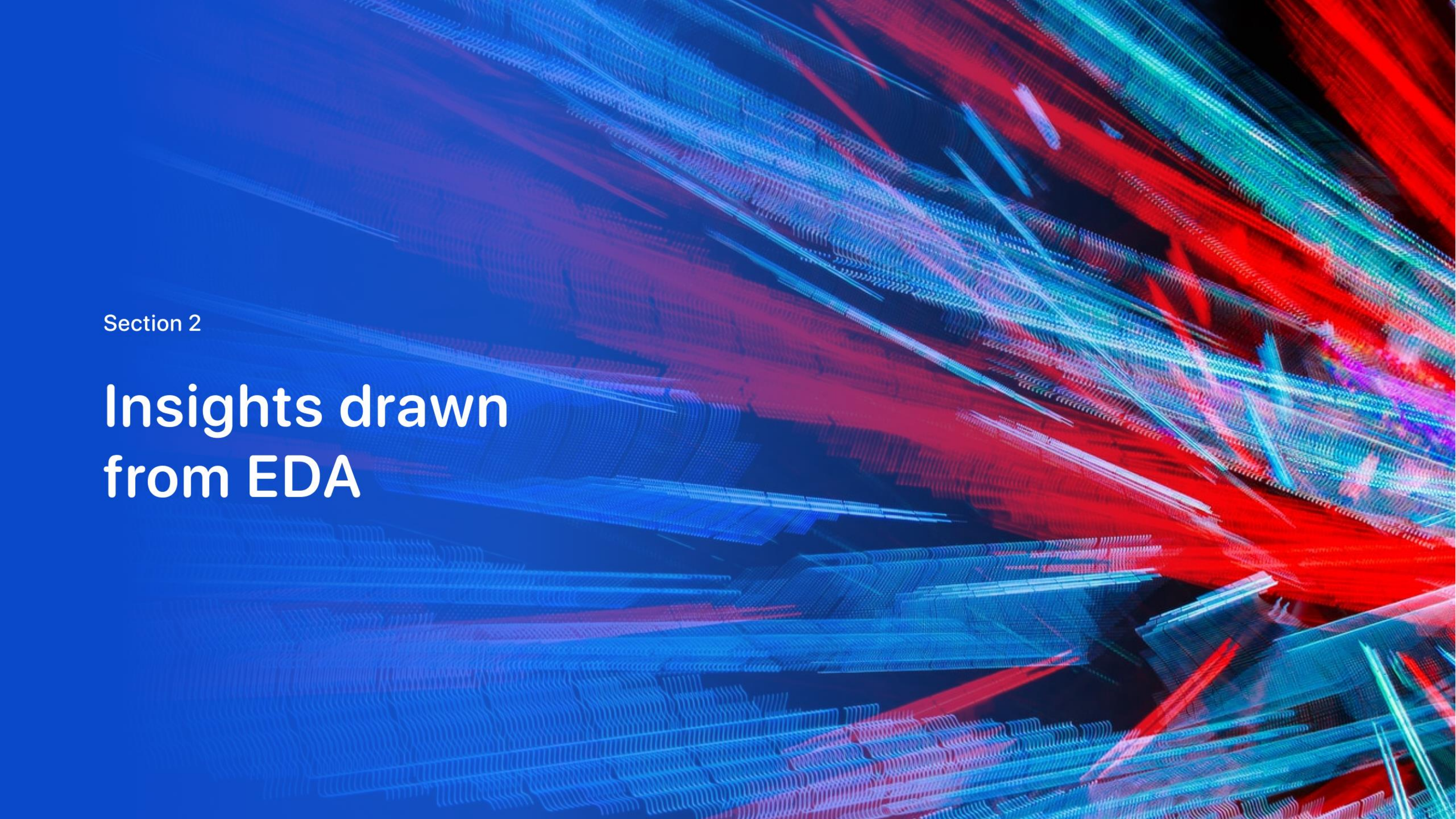
- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification) (link)

- Building the model:
  - Train/Test split; development on training set/evaluation with test set
  - Evaluate train/test set sizes
  - Folded Crossvalidation for hyperparameter fine-tuning (GridSearchCV)
- Evaluation the modles:
  - Model accuracy
  - Confusion matrix evaluation for model biases
- Best model:
  - Highest accuracy

Results

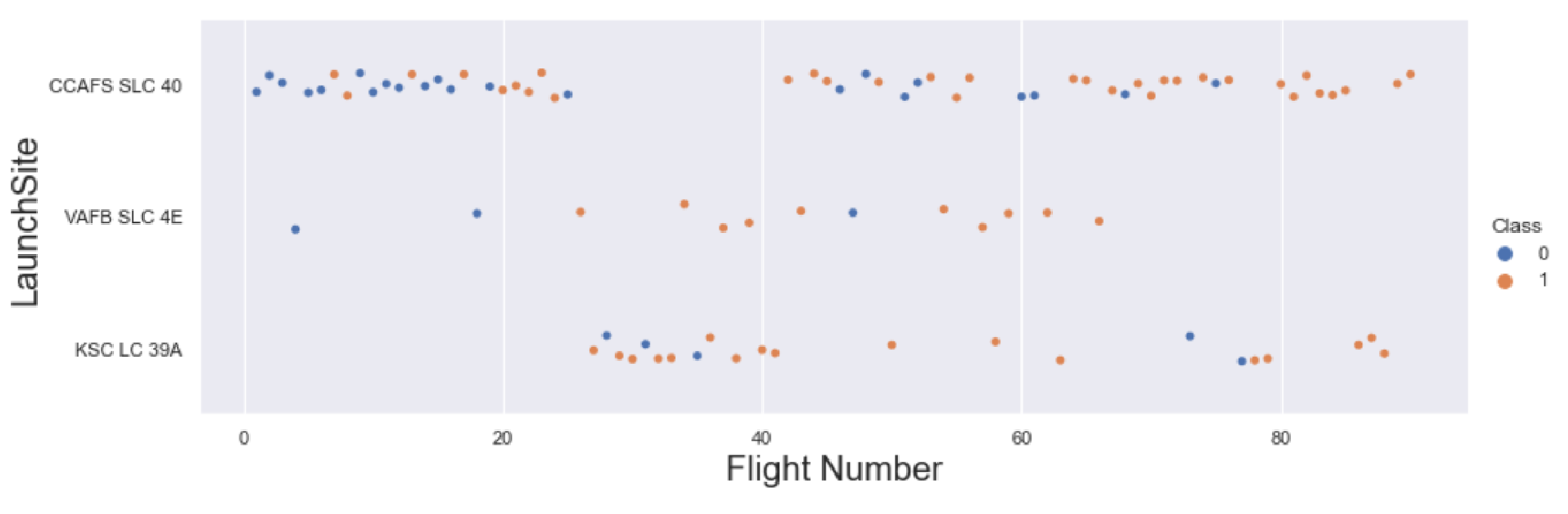
- All ML methods perform equally well on the data
- An accuracy of up to 83% is possible for the first stage landing outcome prediction
- Low weighted payloads perform better than the heavy ones
- The mission Success rate is proportional to and increases with the number of years spent on the project
- The ES-L1, SSO,HEO and GEO are most successful



Section 2

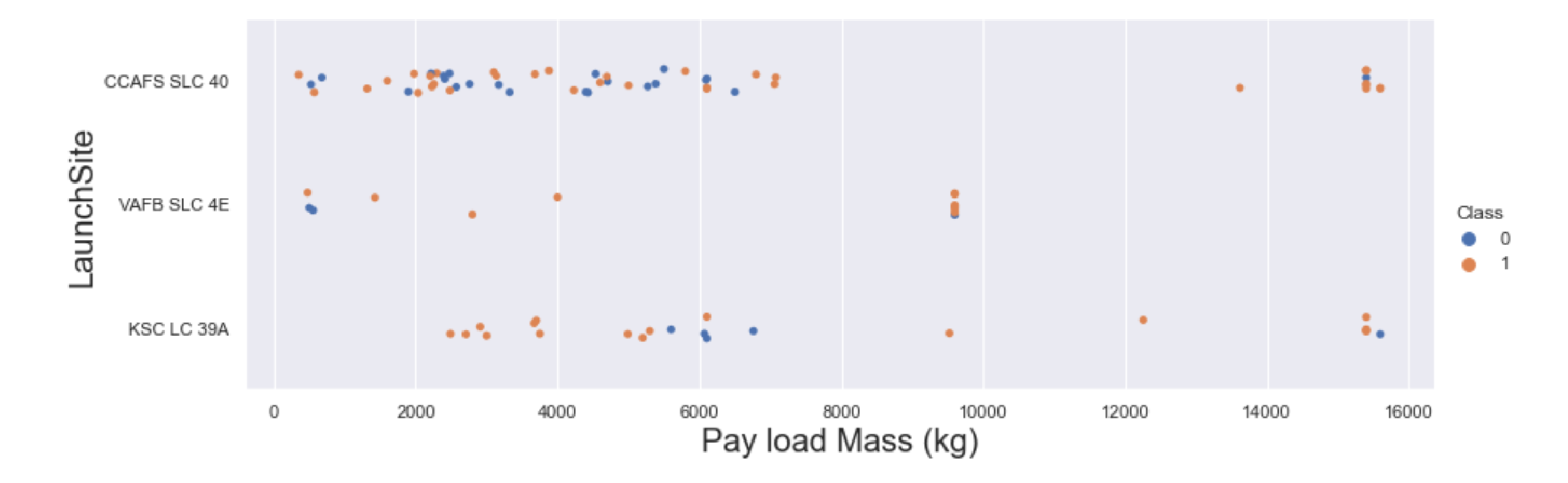
Insights drawn from EDA

Flight Number vs. Launch Site



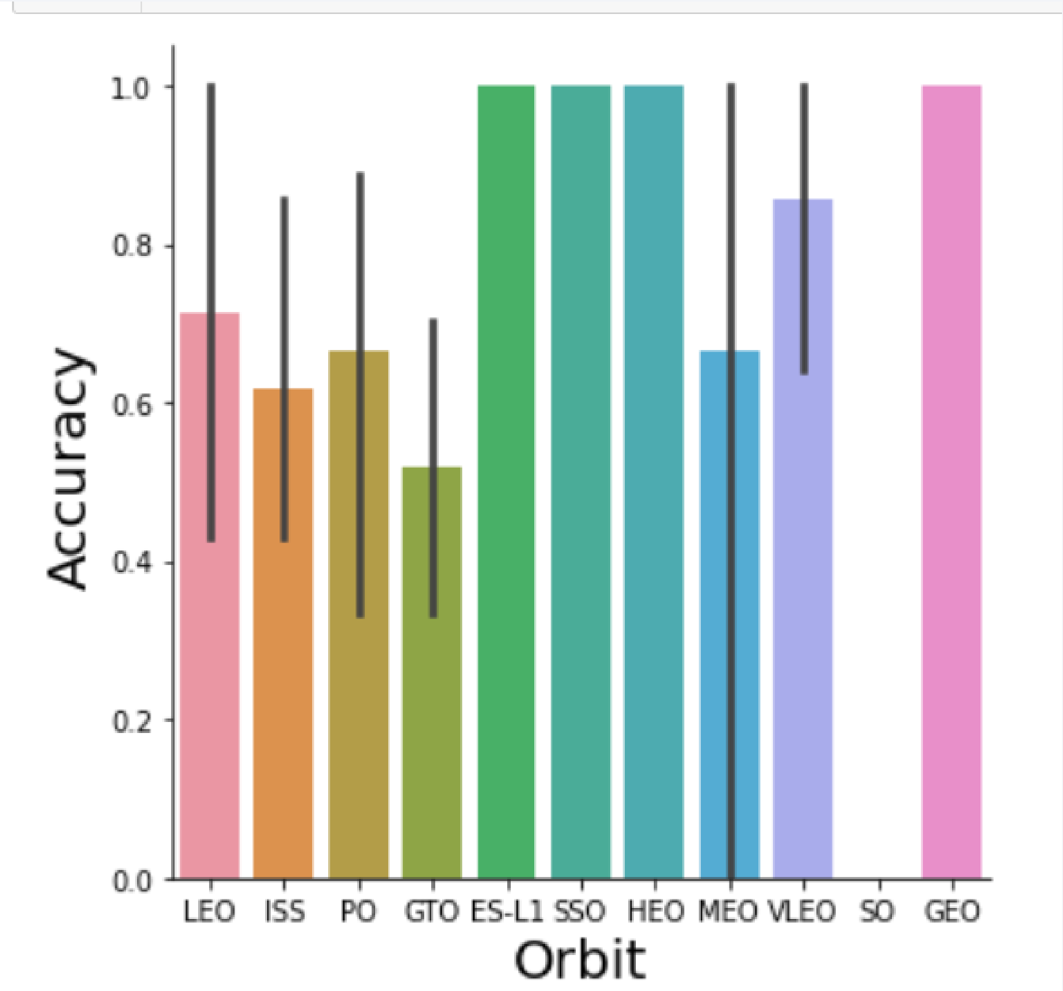
- The success rate increases with a higher flight number

Payload vs. Launch Site



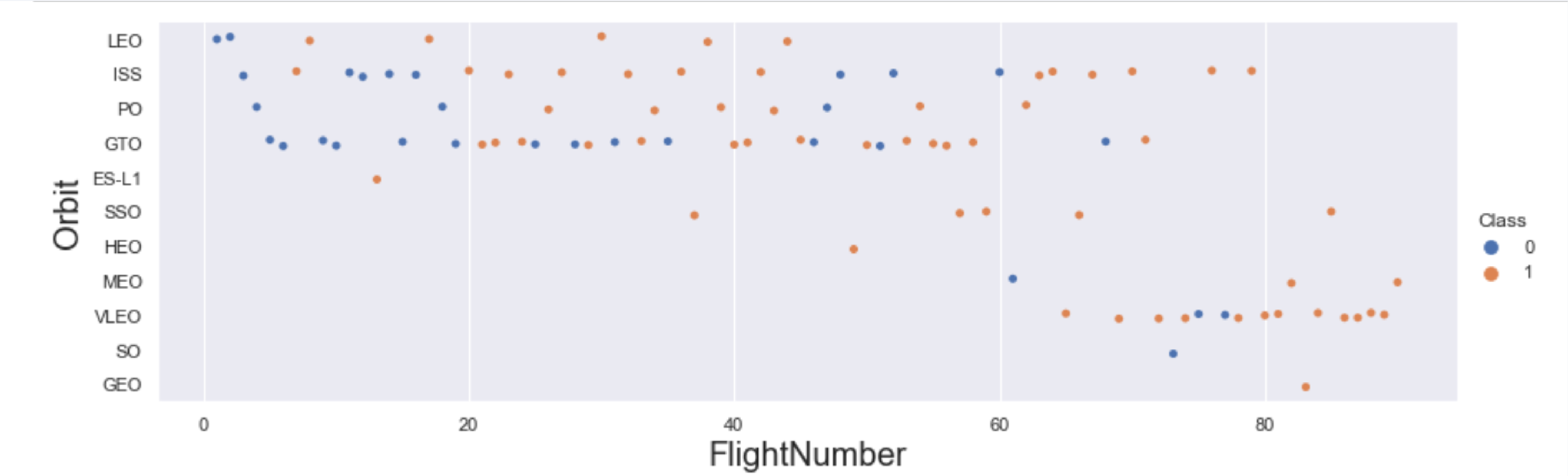
- Clusters of success/failure depending on the payload mass

Success Rate vs. Orbit Type



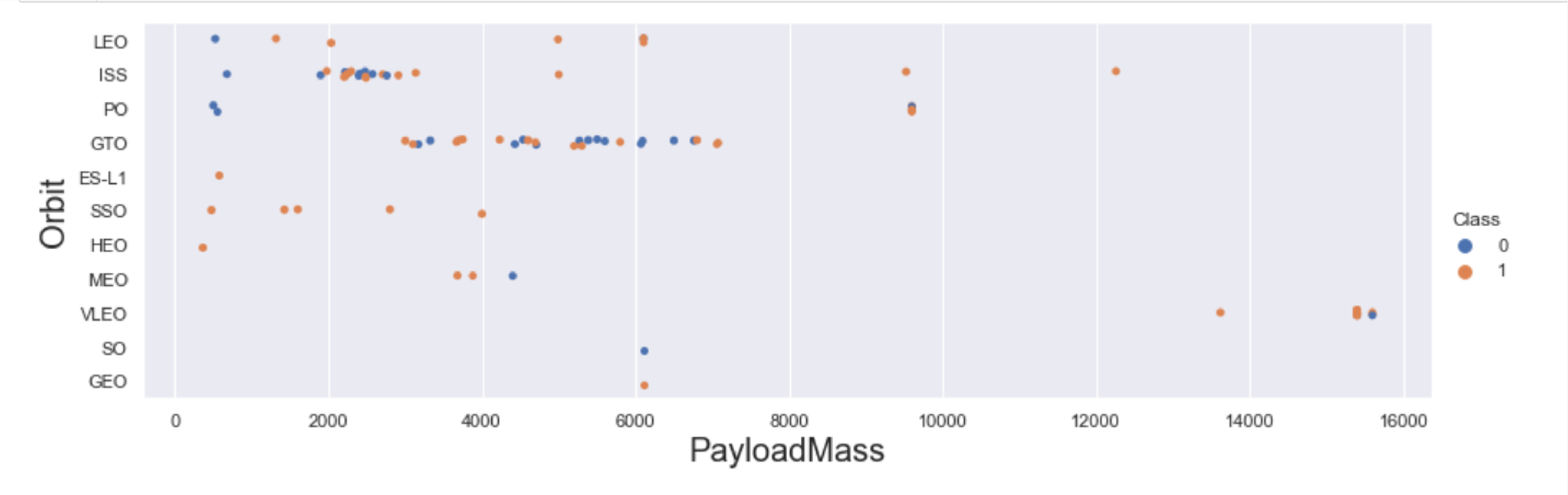
- The ES-L1, SSO,HEO and GEO are most successful.

Flight Number vs. Orbit Type



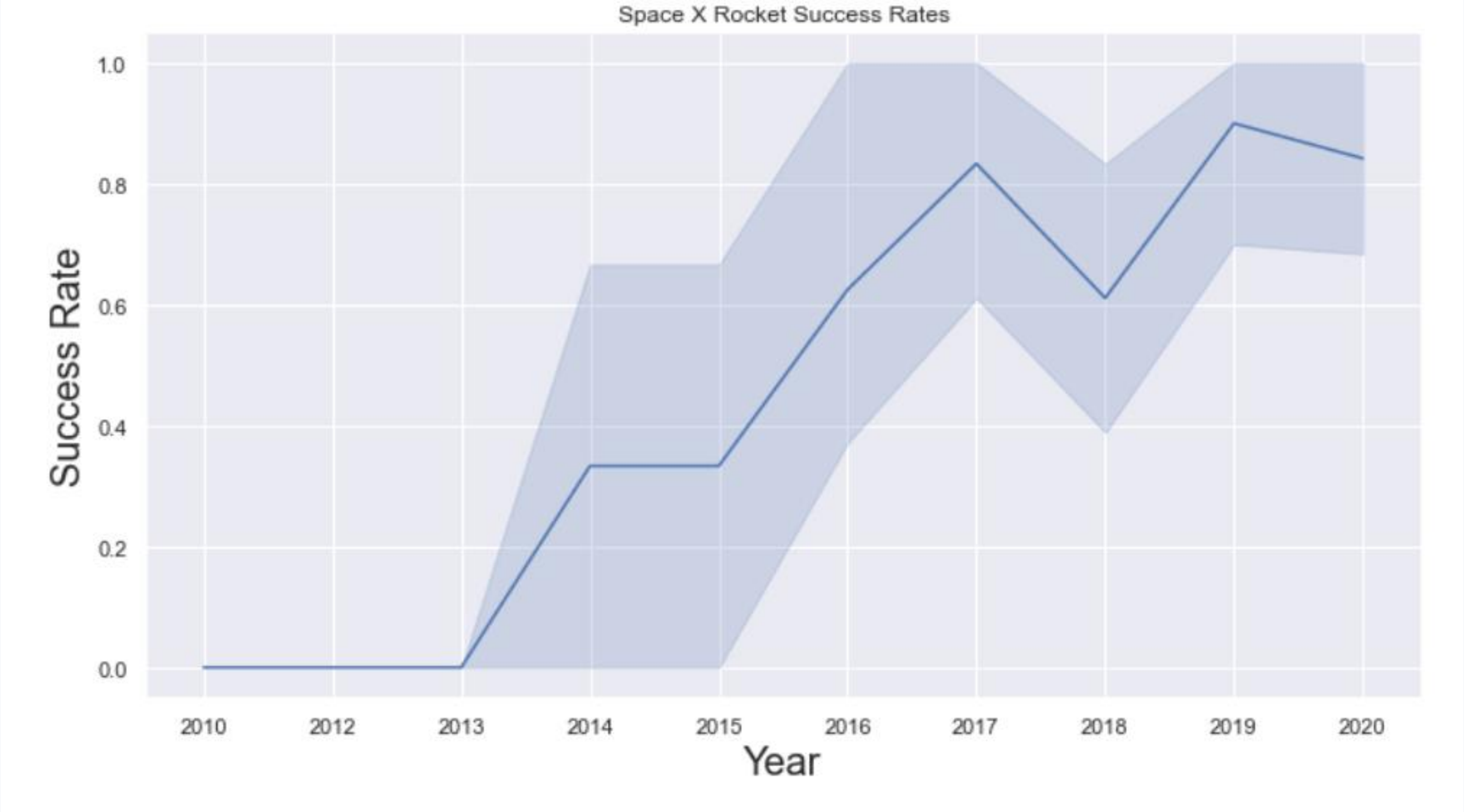
- In some orbits (e.g. LEO) the success rate increases with more flights

Payload vs. Orbit Type



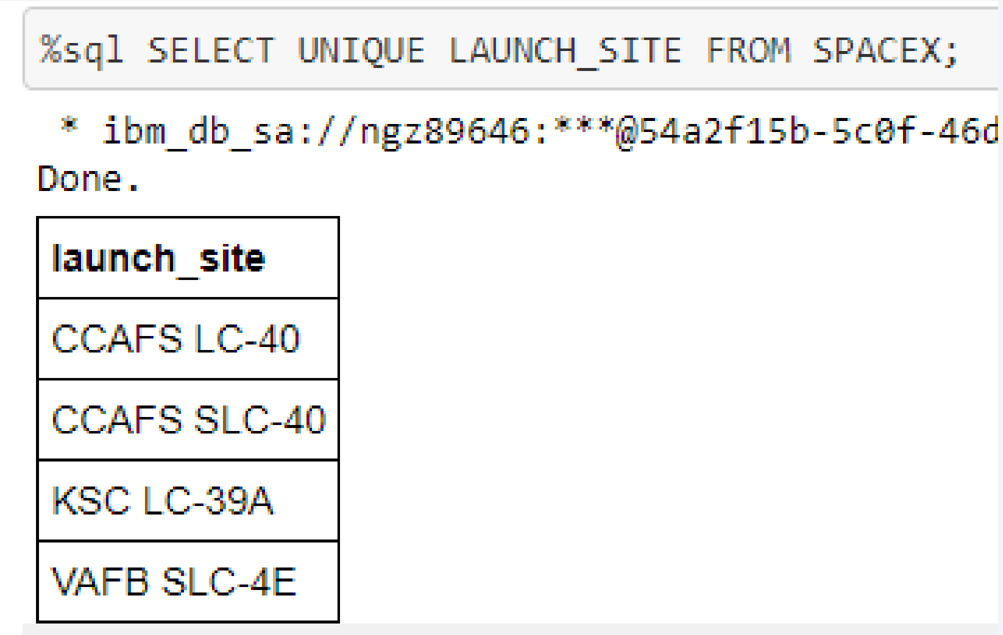
- Heavy payloads increase the success rate on some orbits

Launch Success Yearly Trend



- The success rate is directly proportional to the year

All Launch Site Names



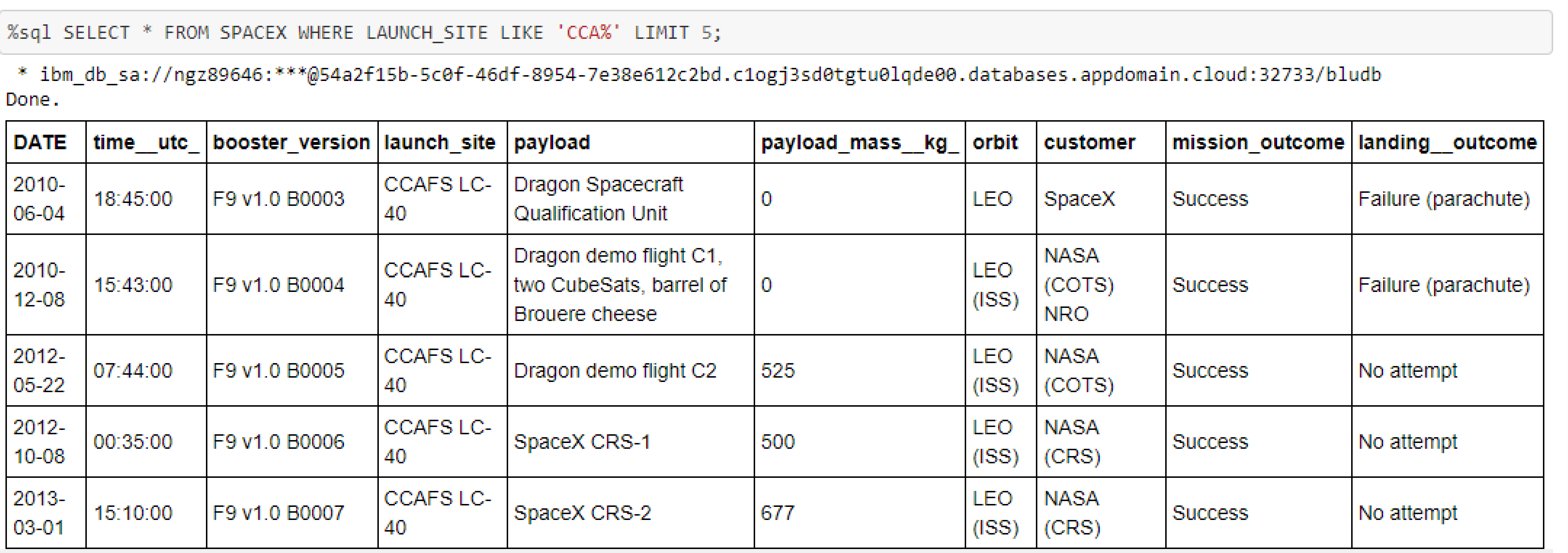
```
%sql SELECT UNIQUE LAUNCH_SITE FROM SPACEX;
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46d
Done.
```

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

- As shown in the visualization of the dataframe we have four different launch sites

Launch Site Names Begin with 'CCA'



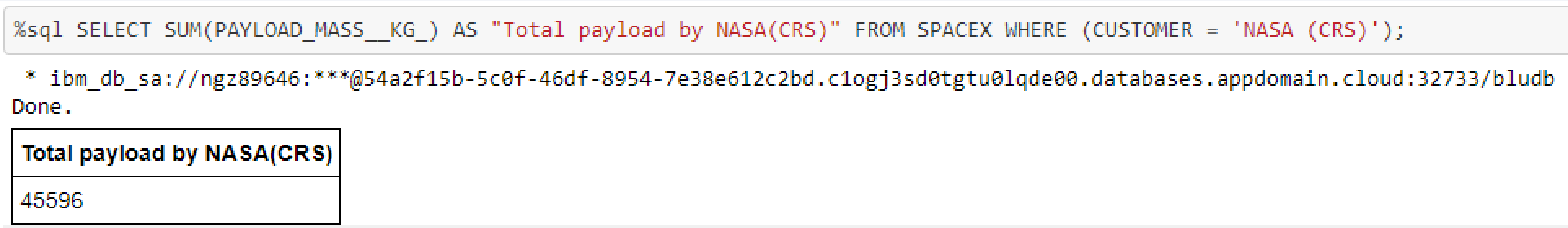
```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/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

- See results in the dataframe

Total Payload Mass



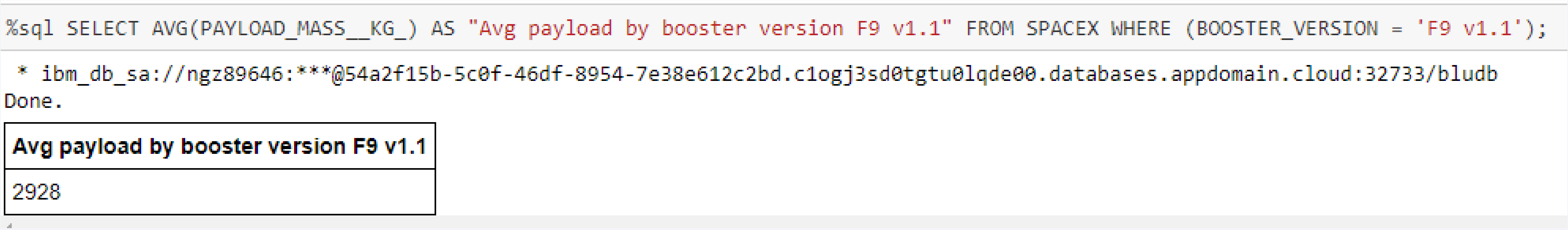
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total payload by NASA(CRS)" FROM SPACEX WHERE (CUSTOMER = 'NASA (CRS)');
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Total payload by NASA(CRS)
45596

- The boosters from NASA carried a total payload of 45596kg.

Average Payload Mass by F9 v1.1



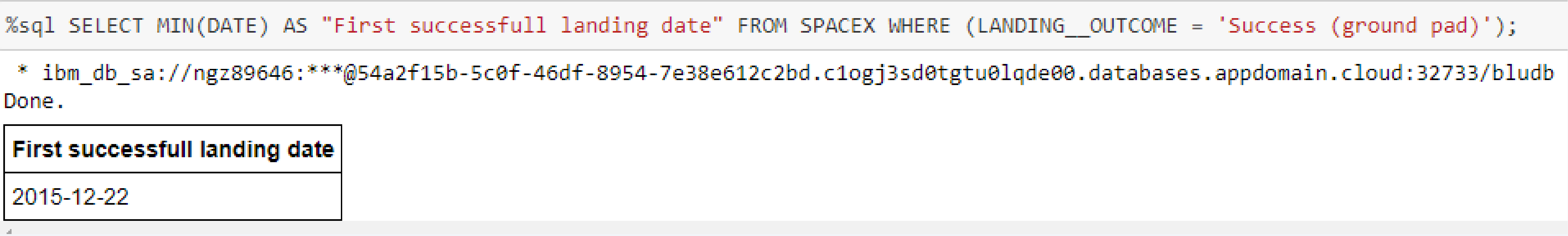
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Avg payload by booster version F9 v1.1" FROM SPACEX WHERE (BOOSTER_VERSION = 'F9 v1.1');
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Avg payload by booster version F9 v1.1
2928

- The average payload mass carried by booster version F9 v1.1 is 2928 kg.

First Successful Ground Landing Date



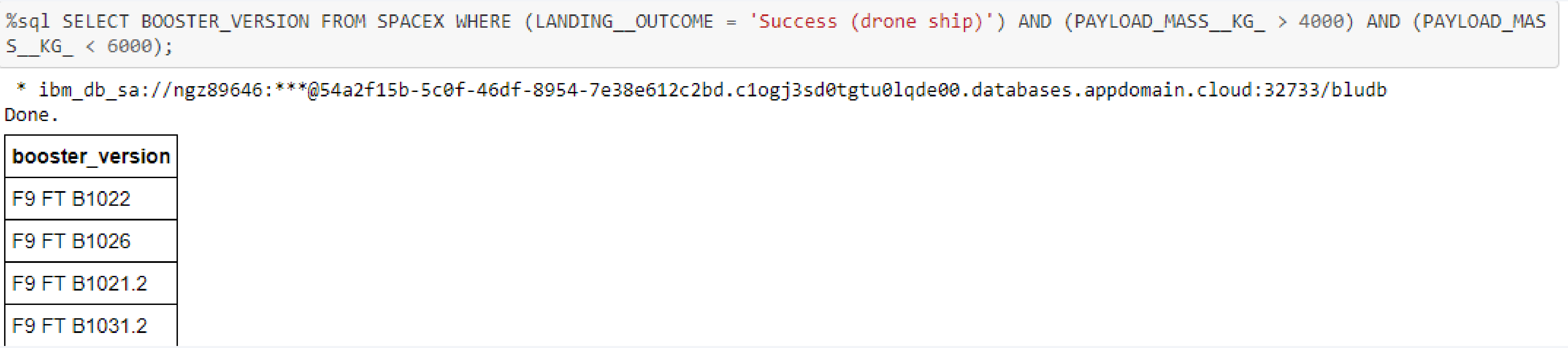
```
%sql SELECT MIN(DATE) AS "First successfull landing date" FROM SPACEX WHERE (LANDING__OUTCOME = 'Success (ground pad)');
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

First successfull landing date
2015-12-22

- First successful landing outcome on ground pad happened on the 22.12.2015.

Successful Drone Ship Landing with Payload between 4000 and 6000



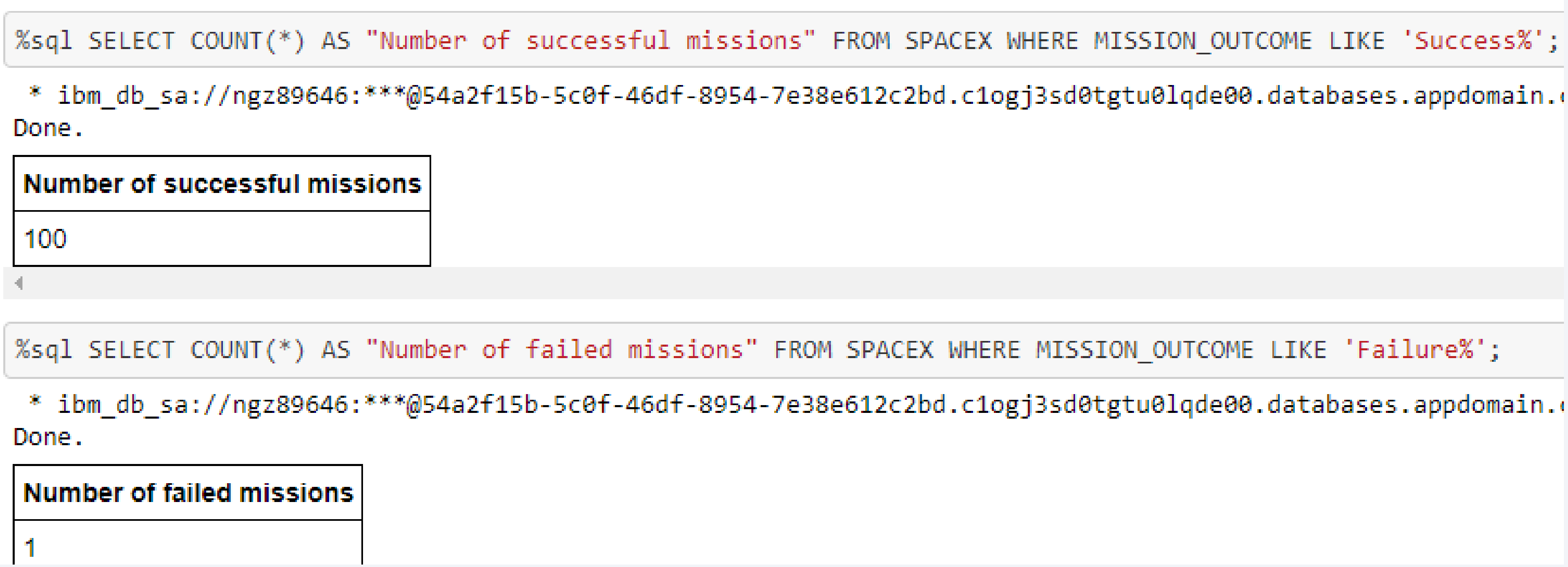
```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE (LANDING__OUTCOME = 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ > 4000) AND (PAYLOAD_MAS
S__KG_ < 6000);
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

- We have four different boosters in this category.

Total Number of Successful and Failure Mission Outcomes



```
%sql SELECT COUNT(*) AS "Number of successful missions" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.
Done.
```

Number of successful missions
100

```
%sql SELECT COUNT(*) AS "Number of failed missions" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.
Done.
```

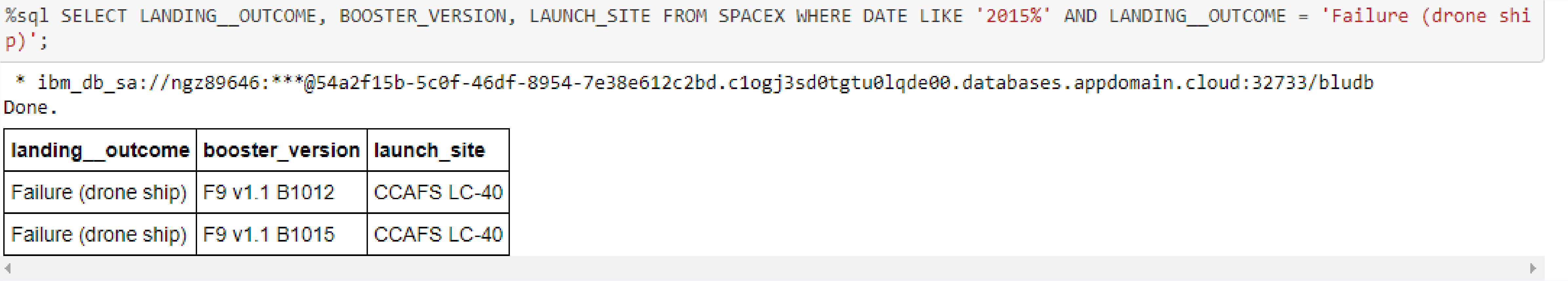
Number of failed missions
1

- We have 100 successful and one failed mission

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



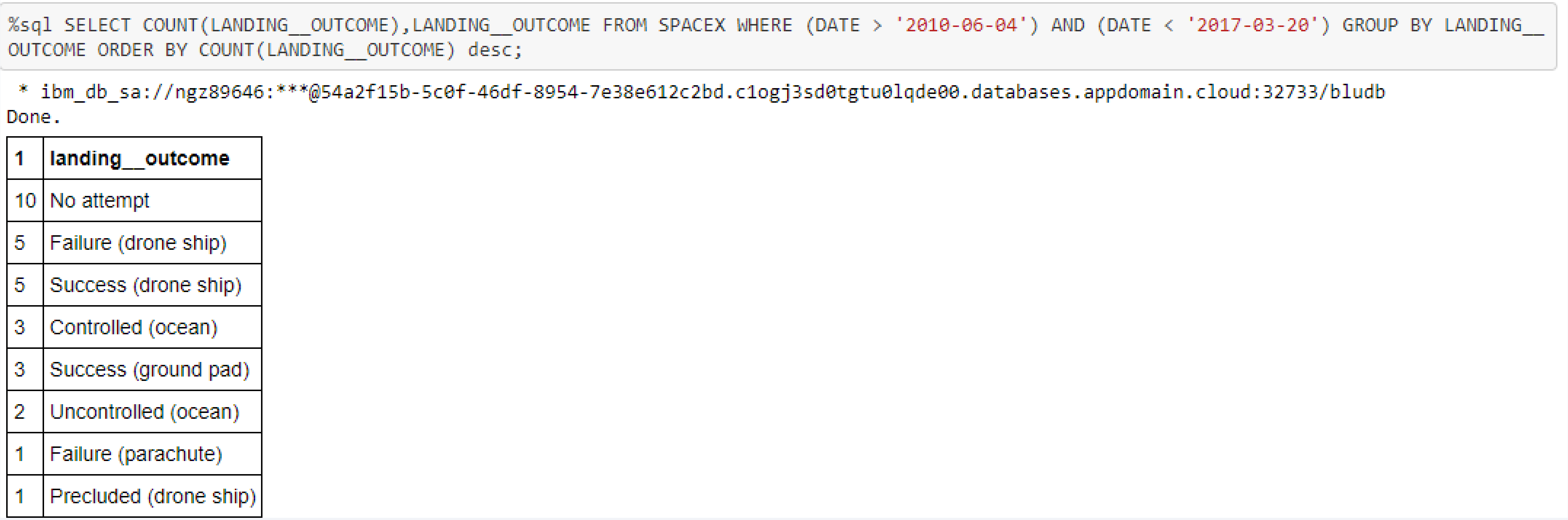
```
%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015%' AND LANDING__OUTCOME = 'Failure (drone shi
p)';
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

- See dataframe visualization for a list of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

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



```
%sql SELECT COUNT(LANDING__OUTCOME),LANDING__OUTCOME FROM SPACEX WHERE (DATE > '2010-06-04') AND (DATE < '2017-03-20') GROUP BY LANDING__
OUTCOME ORDER BY COUNT(LANDING__OUTCOME) desc;
```

```
 * ibm_db_sa://ngz89646:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

- See dataframe visualization for a ranking of 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.

Section 4

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

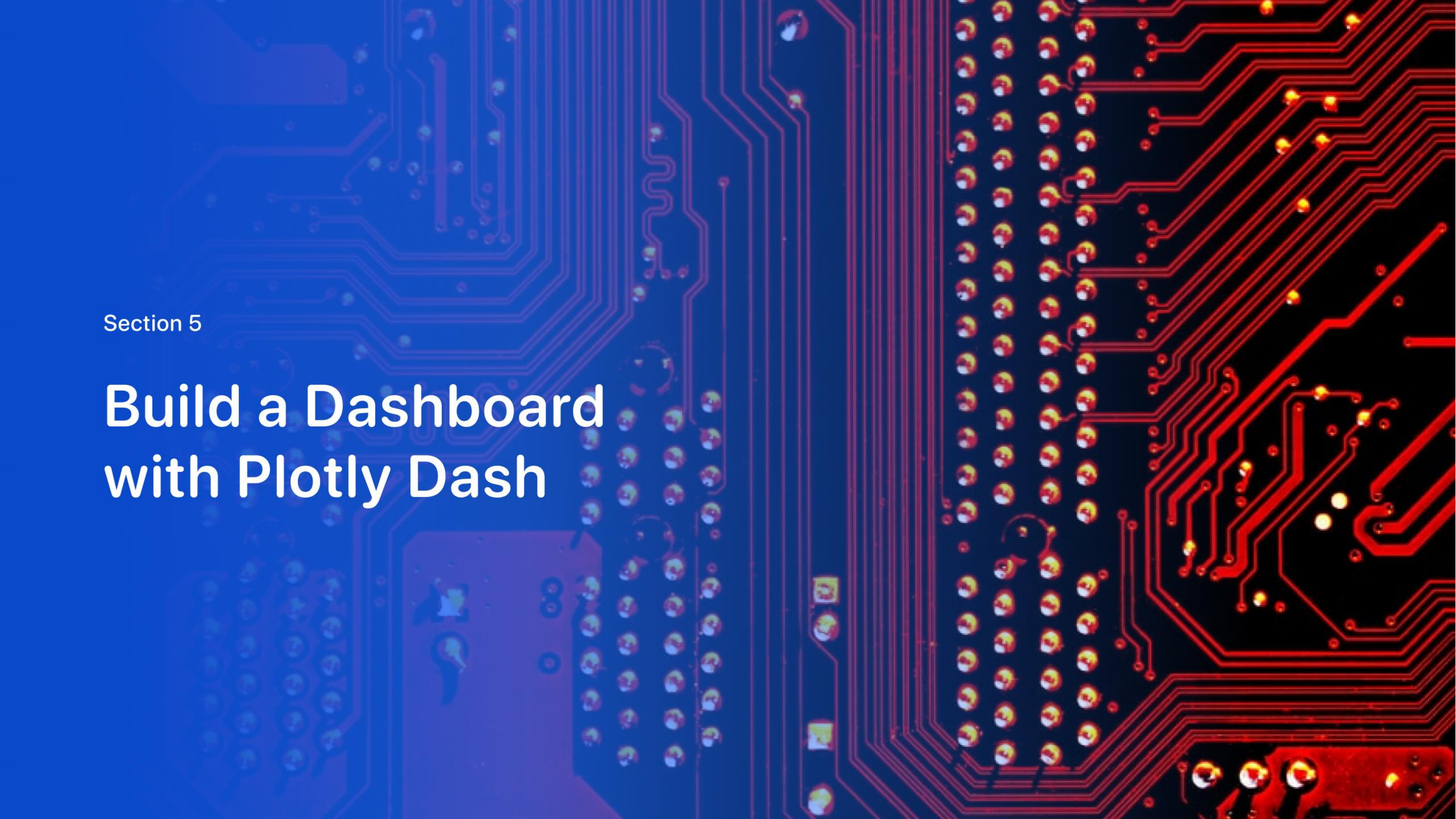
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot



Section 5

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title

- Show the screenshot of launch success count for all sites, in a piechart

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

- Show the screenshot of the piechart for the launch site with highest launch success ratio

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

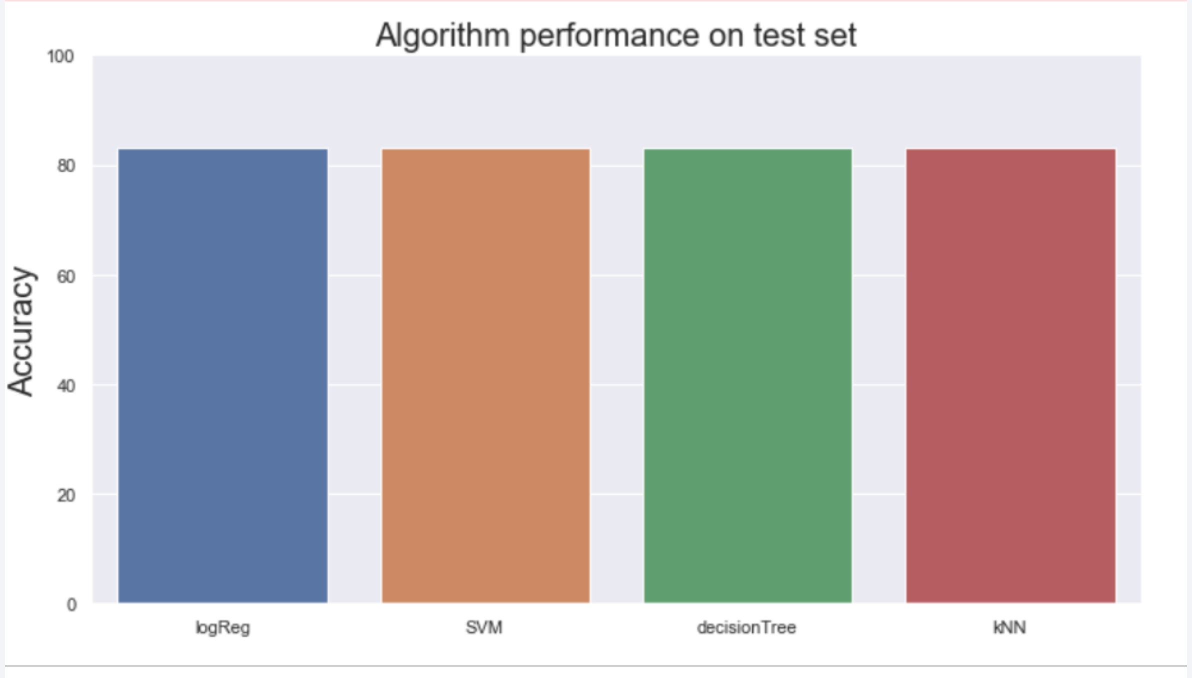
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

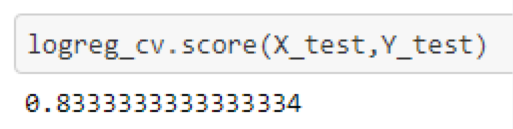
Section 6

Predictive Analysis (Classification)

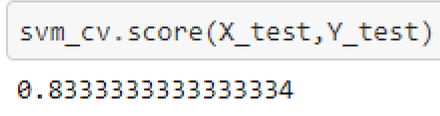
Classification Accuracy



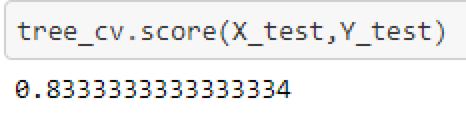
Results:



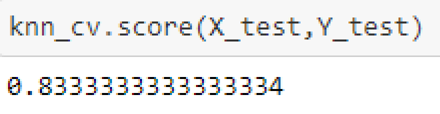
```
logreg_cv.score(X_test,Y_test)
0.8333333333333334
```



```
svm_cv.score(X_test,Y_test)
0.8333333333333334
```



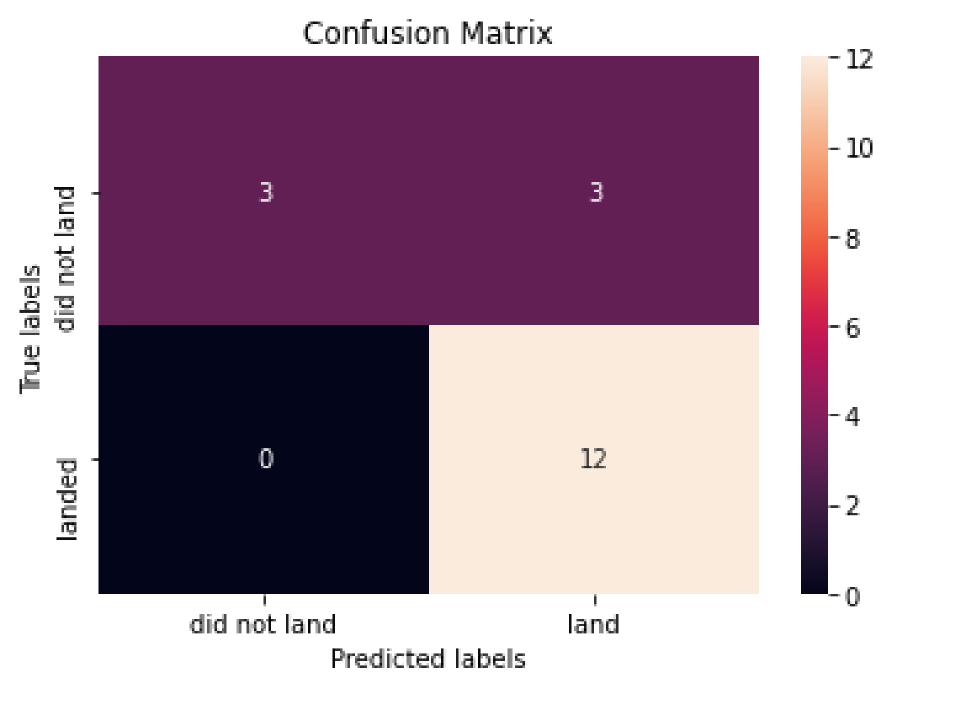
```
tree_cv.score(X_test,Y_test)
0.8333333333333334
```



```
knn_cv.score(X_test,Y_test)
0.8333333333333334
```

- All four algorithms perform equally well with a performance of 83.3% accuracy on the test data

Confusion Matrix



Confusion matrix of the decision tree model.

We see that logistic regression can distinguish between the different classes.

We see that the major problem is the high number of false positives.

Conclusions

- All ML methods perform equally well on the data
- An accuracy of up to 83% is possible for the first stage landing outcome prediction
- Low weighted payloads perform better than the heavy ones
- The mission Success rate is proportional to and increases with the number of years spent on the project
- The ES-L1, SSO,HEO and GEO are most successful

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!