IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jorge Luis Oñate Hernandez
September 27, 2022

https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone.git

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 Predictions
- **Summary of all results**
  - Exploratory Data Analysis results
  - Interactive Analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  1. The task at hand is to predict if the first stage of the SpaceX Falcon 9 rocket will land successfully/
  2. What factors will indicate us as to whether a rocket lands successfully or not.

Section 1

Methodology

Methodology

Executive Summary

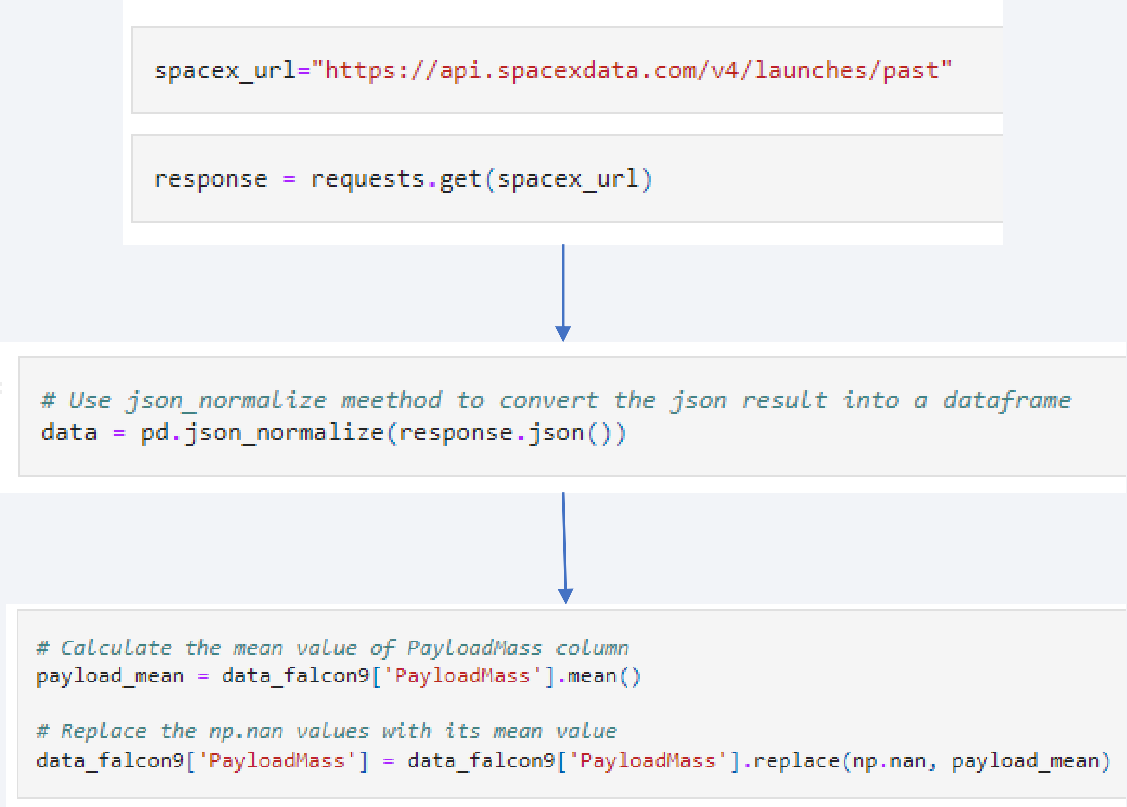
- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One-hot encoding and data cleaning of irrelevant data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models (LR, KNN, SVM, DT)

Data Collection

- The data was collected in the following ways:
  - SpaceX API was used to collect the SpaceX launch data. (APIs endpoints/URL starts with api.spacexdata.com/v4/)
  - Using the API, we gathered data about launches including information about the rockets used, launch specifications and landing outcome, and many other attributes.
  - Additionally, we performed web scrapping from Wikipedia using BeautifulSoup and converted it to a pandas dataframe to be able to perform methods to the data.

Data Collection – SpaceX API

- The SpaceX API was used to collect the data.
- The data was cleaned and formatted as to be used with ease.
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



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

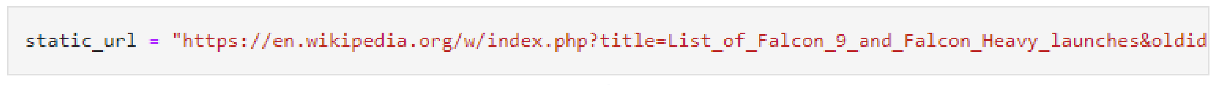
```
response = requests.get(spacex_url)
```

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

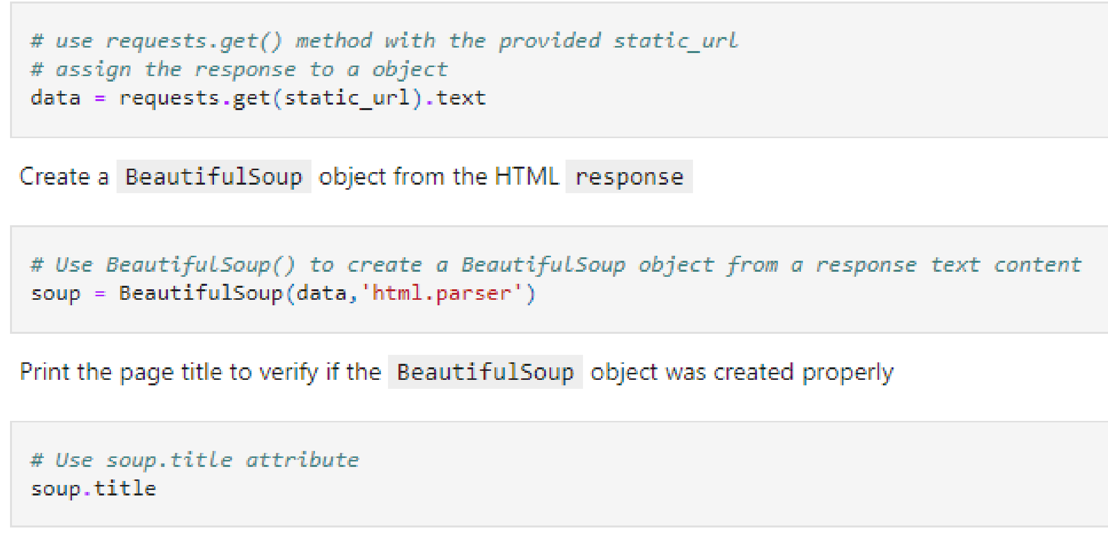
```
# Calculate the mean value of PayloadMass column
payload_mean = data_falcon9['PayloadMass'].mean()

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

Data Collection - Scraping

- The Falcon 9 launh data was collected from Wikipedia with BeautifulSoup.
- It was parsed and converted into a pandas dataframe.
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid
```



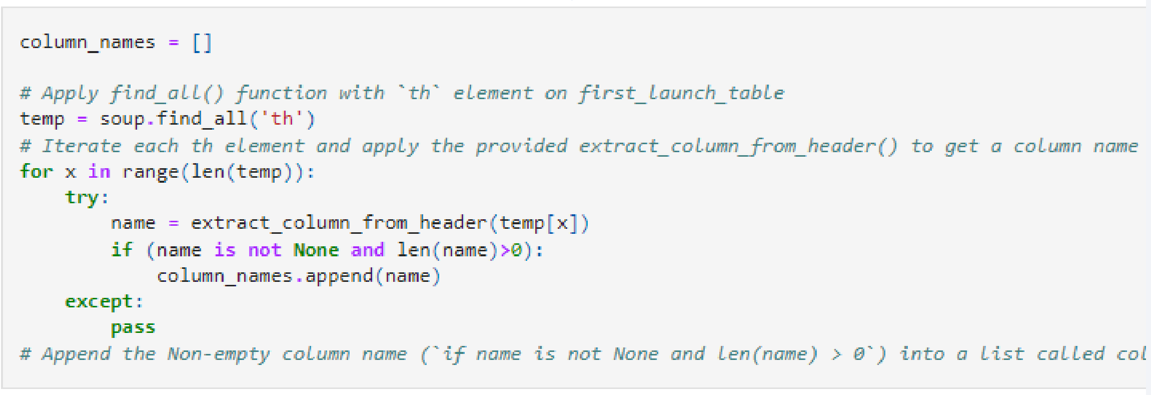
```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

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

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

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

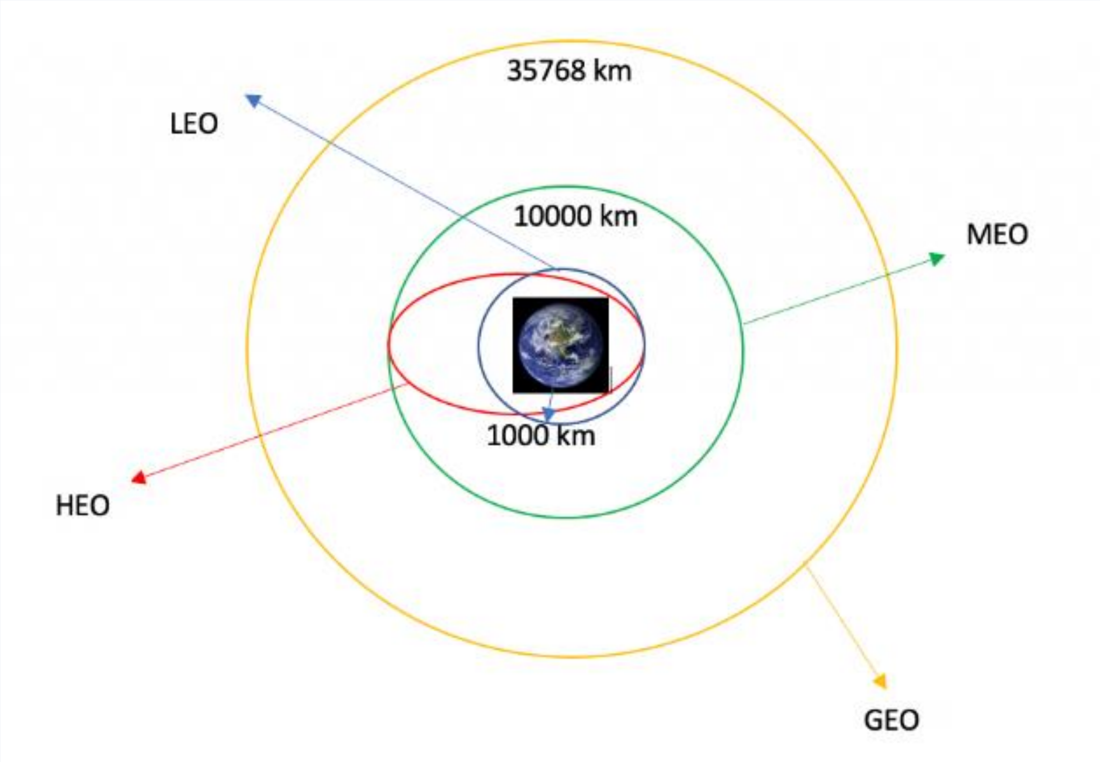
```
# Use soup.title attribute
soup.title
```



```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
temp = soup.find_all('th')
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for x in range(len(temp)):
    try:
        name = extract_column_from_header(temp[x])
        if (name is not None and len(name)>0):
            column_names.append(name)
    except:
        pass
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called col
```

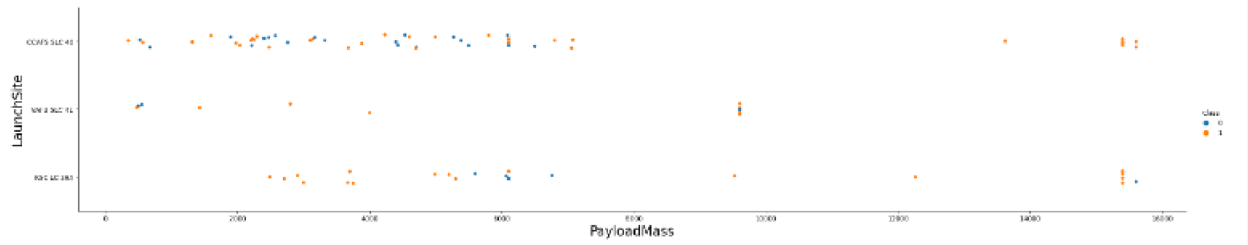
Data Wrangling

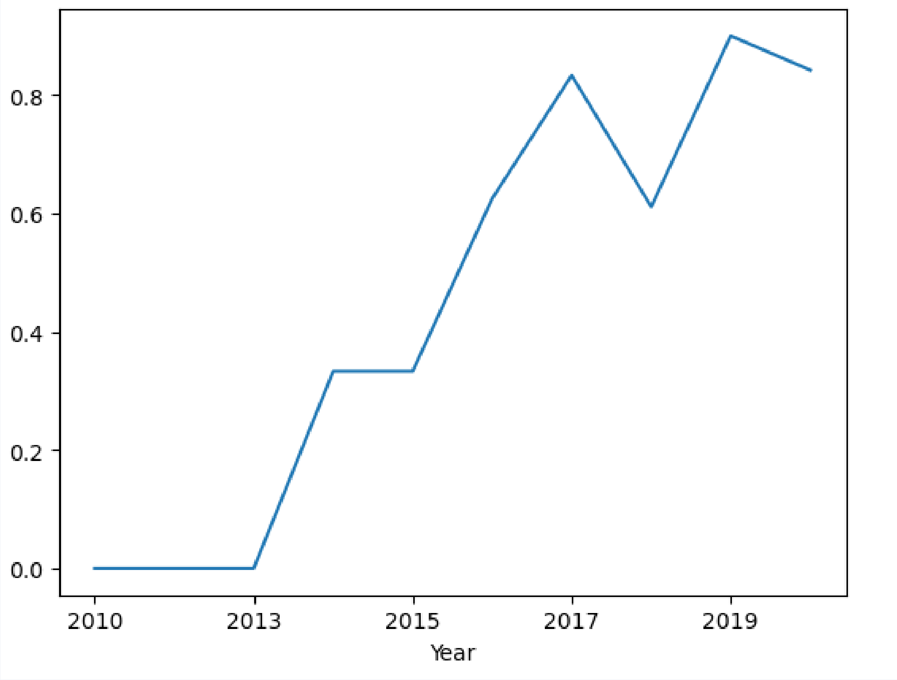
- Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- The Number of launches on each site was calculated and the occurrence of each orbits.
- Finally, we created a landing outcome label from the outcome column. (null values where also taken care of)
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

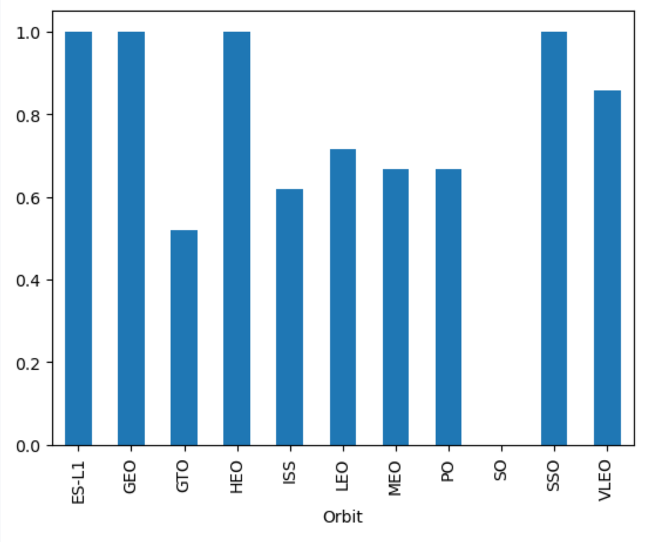
- We explored the data and their relationships. (For example, the relationship between Payload and Launch Site - Image 1)
- The average success rate was also plotted for launches based on the years. (Image 2)
- Data like flight number and orbit type we also looked at.
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb



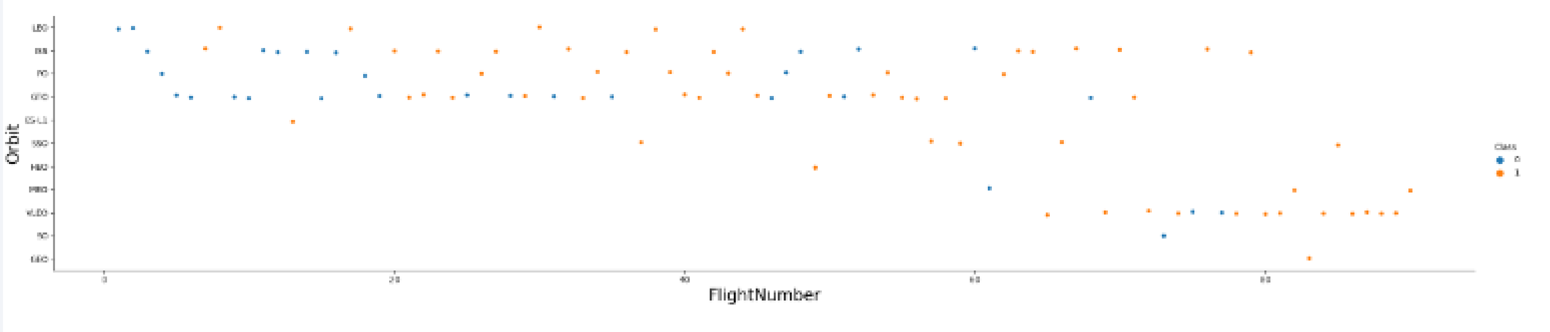


EDA with Data Visualization

Relationship between success rate of each orbit type (bar chart)



Relationship between Flight Number and Orbit type

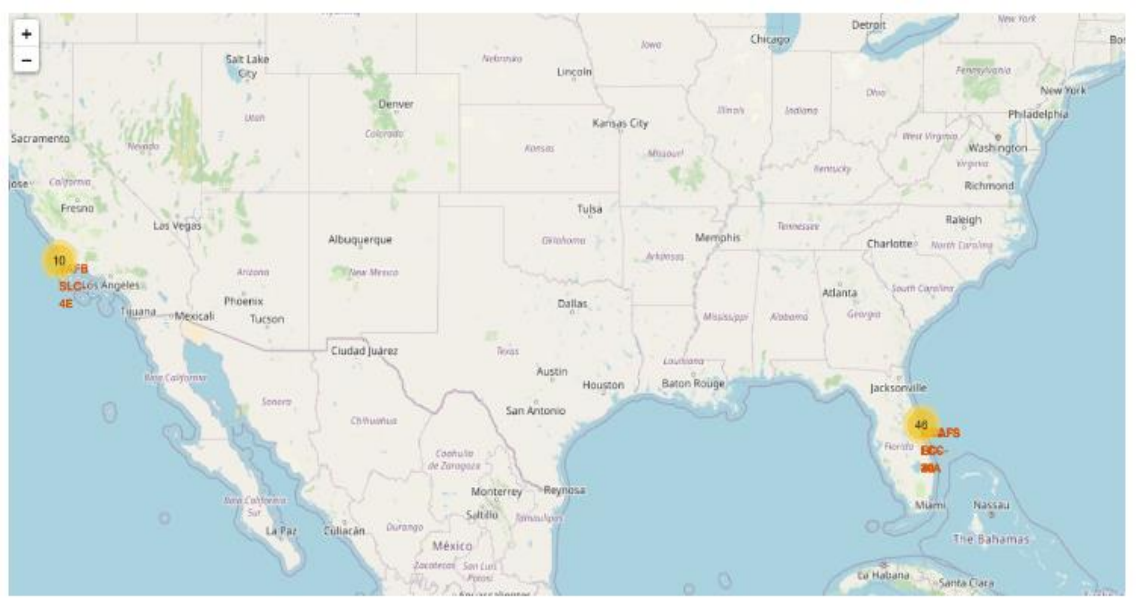


EDA with SQL

- The SQL queries that were performed are the following:
  - 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 succesful landing outcome in ground pad was acheived.
  - 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 04-06-2010 and 20-03-2017 in descending order.
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

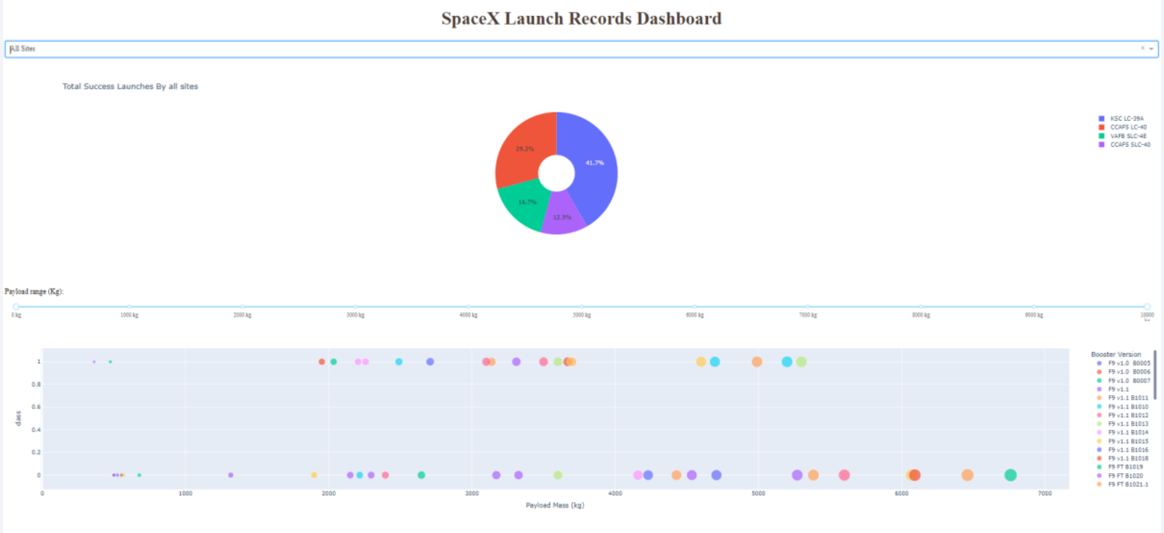
Build an Interactive Map with Folium

- Launch sites were marked on the map, other indicators likes circles or markers where also added to represent successful or unsuccessful launches.
- Marker clusters can be a good way to simplify a map containing many markers having the same coordinate.
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb



Build a Dashboard with Plotly Dash

- An interactive dashboard application was built using Plotly dash, input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart were added to the application.
- To find if variables payload is correlated to mission outcome, we added a range slider to select payload.
- Link to the app: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py


SpaceX Launch Records Dashboard

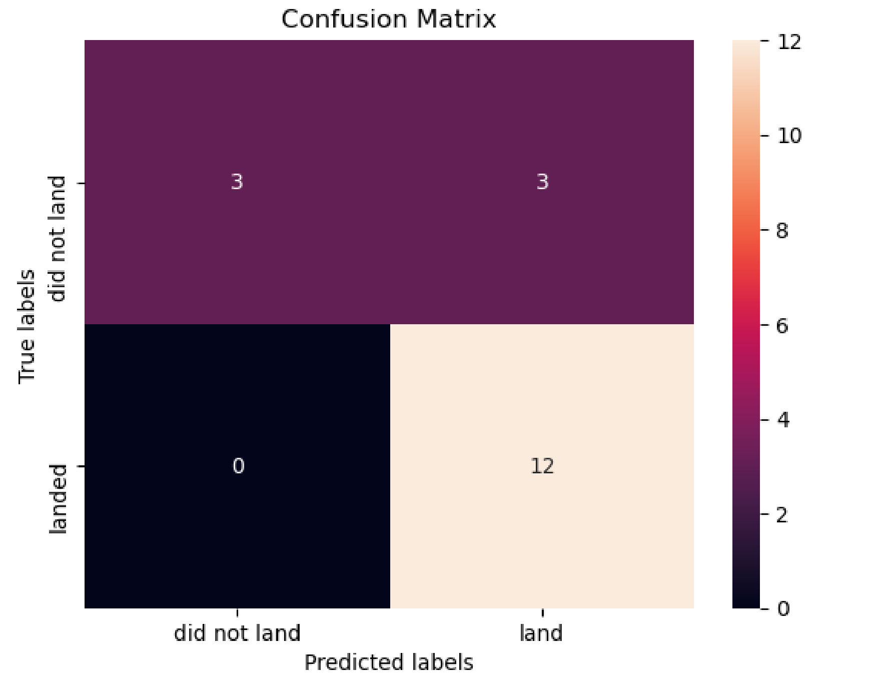
All Sites

Total Success Launches By all sites

Payload range (Kg):

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Link to the notebook: https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Results

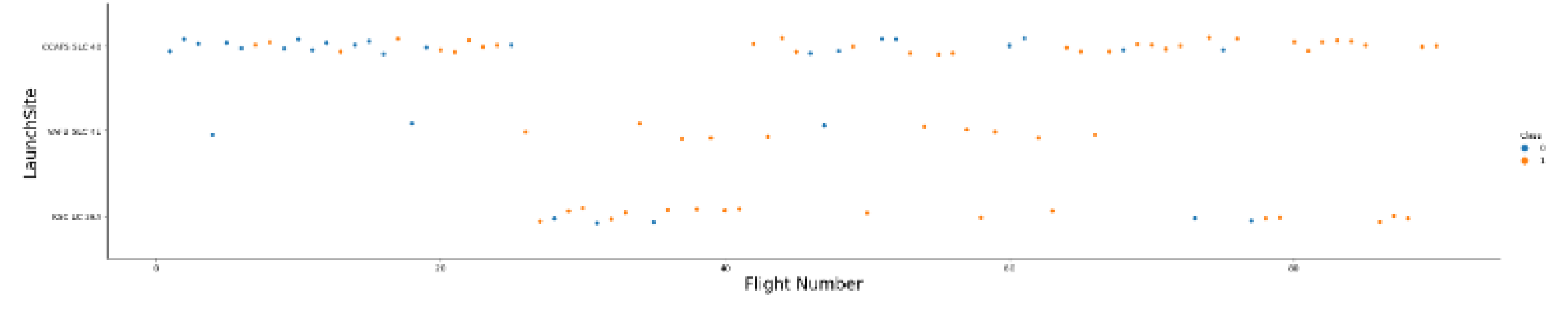
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

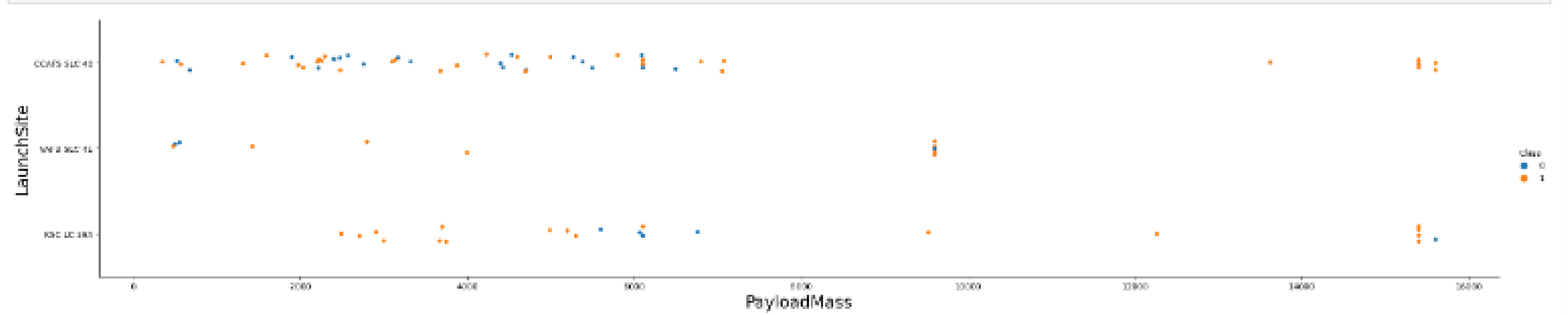
Flight Number vs. Launch Site

- The launches from CCAFS SLC-40 are significantly higher than other sites.



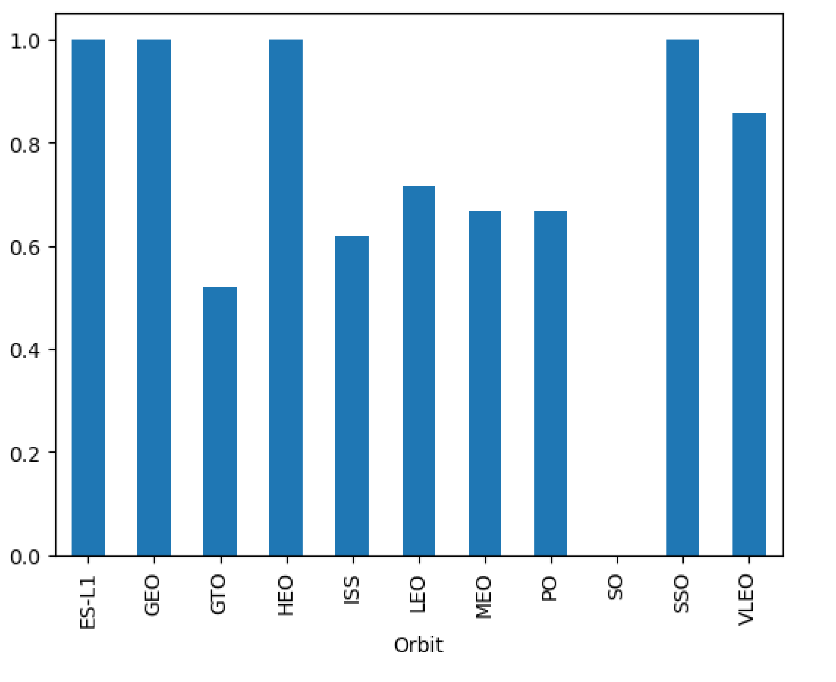
Payload vs. Launch Site

- VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- The highest number of Payloads have been launched from CCAFS SLC-40 under 8000.



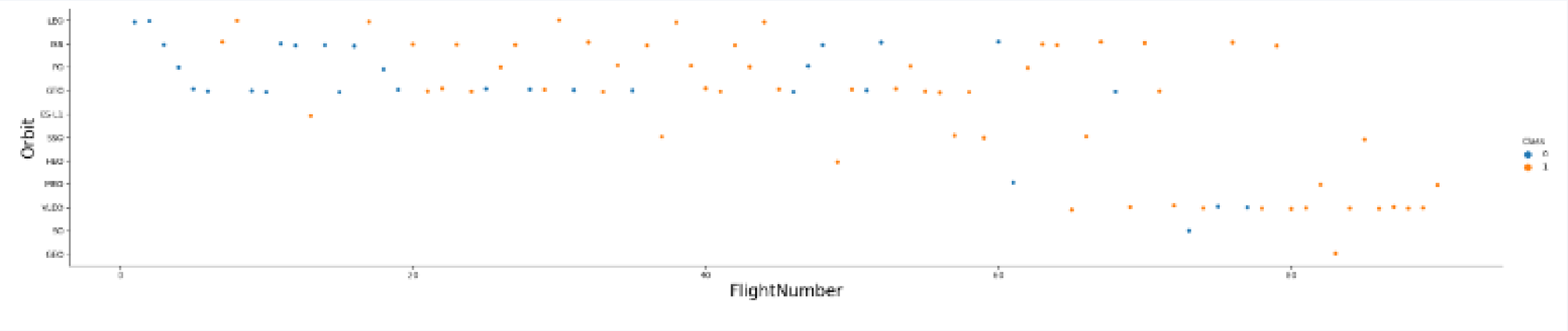
Success Rate vs. Orbit Type

- We can observe that ES-L1, GEO, HEO, SSO have the highest success rate.



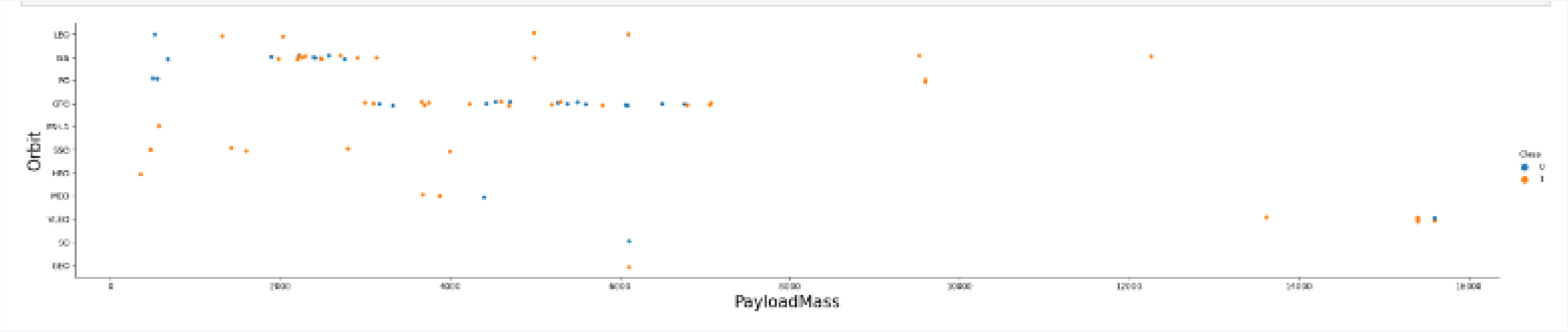
Flight Number vs. Orbit Type

- 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.



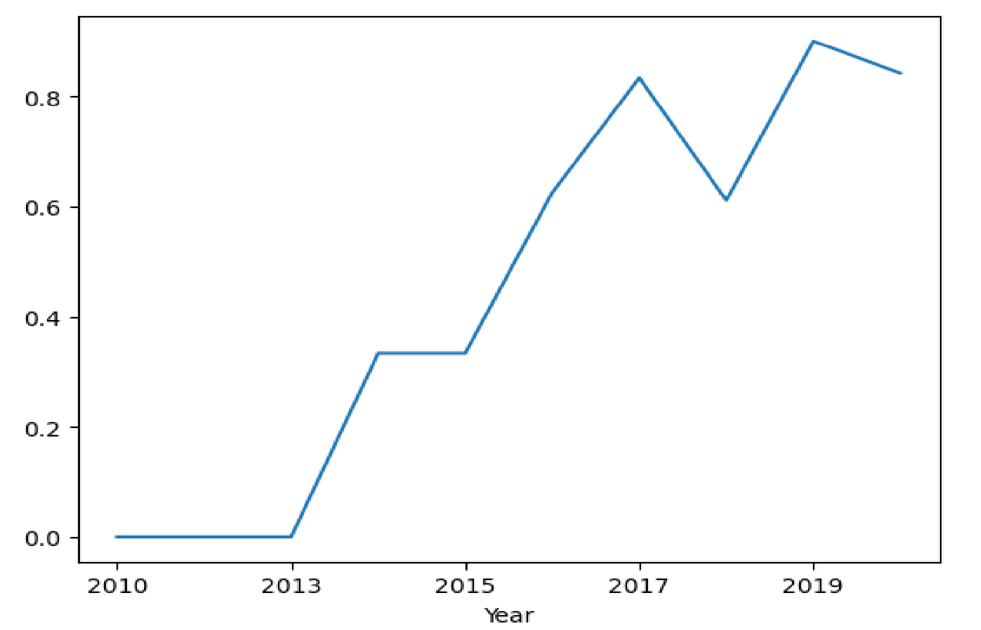
Payload vs. Orbit Type

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



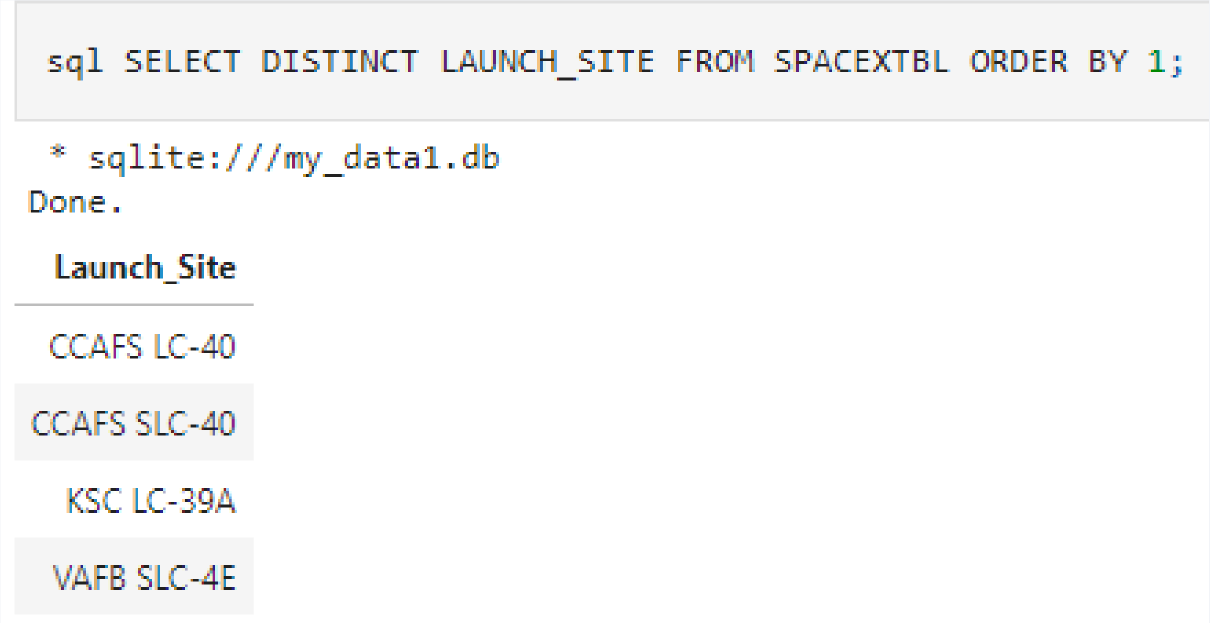
Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2020, which can possibly be attributed to growing and advancing technologies in the field.



All Launch Site Names

We are getting the unique values by using ”DISTINCT”



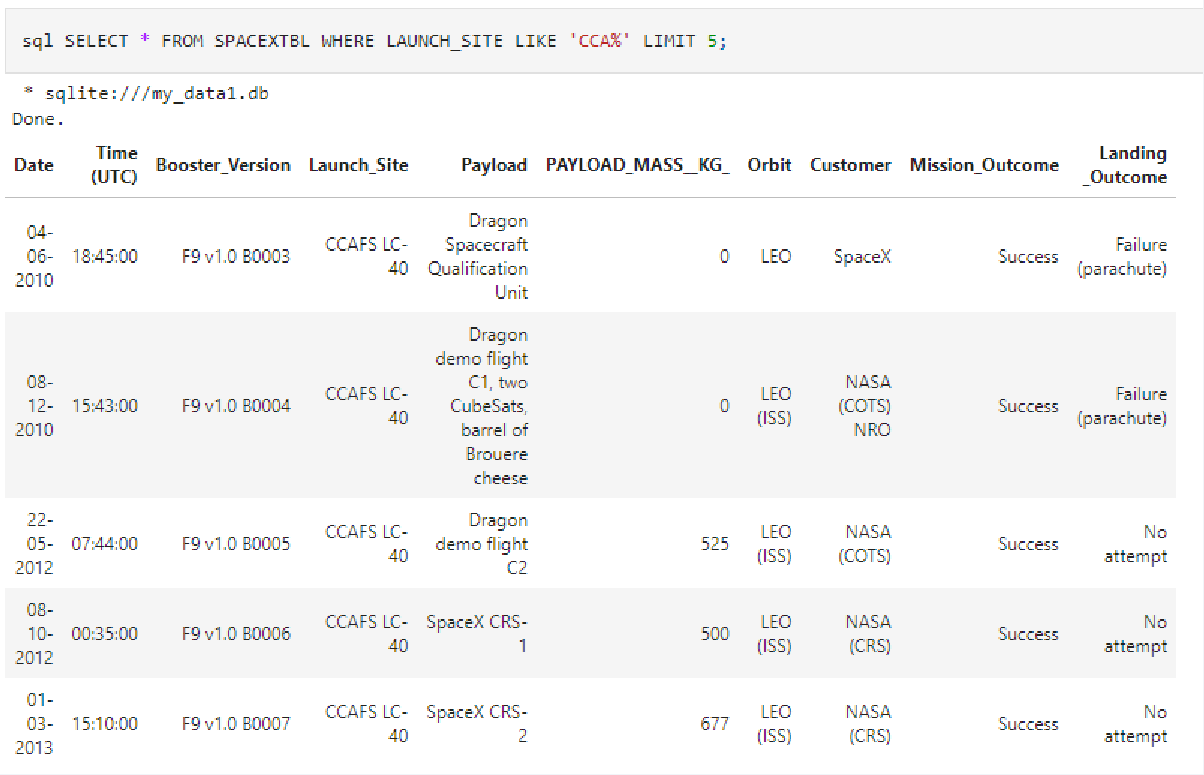
```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

We are Limiting the number of results shown by 5 and looking for names that start with CCA using “LIKE”.



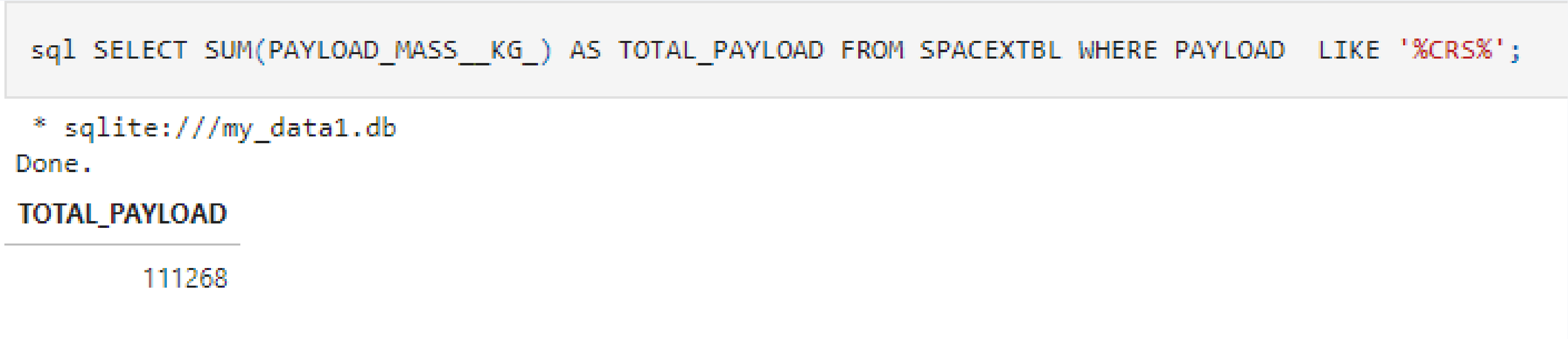
```
sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

We are getting the SUM of the “PAYLOAD_MASS_KG_”



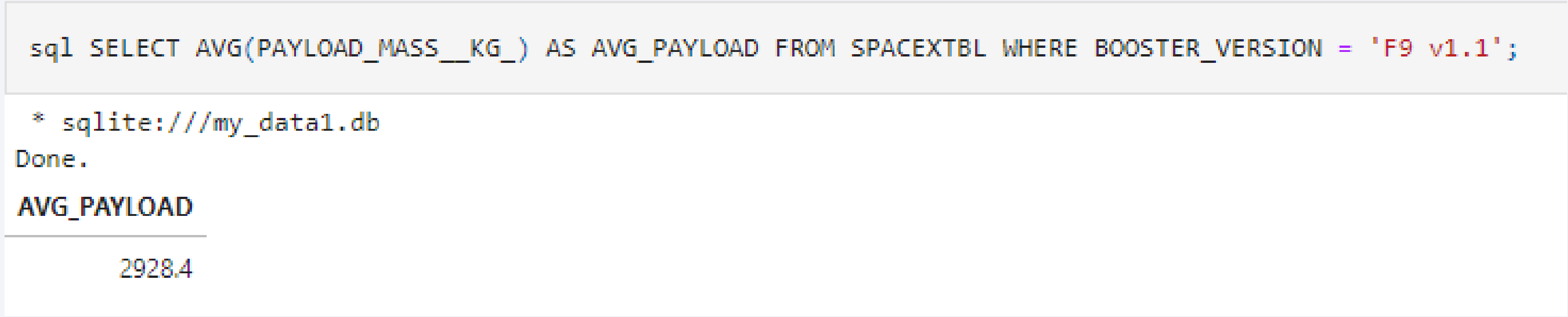
```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

```
 * sqlite:///my_data1.db
Done.
```

TOTAL_PAYLOAD
111268

Average Payload Mass by F9 v1.1

Using “AVG” gives us the average of all values we are using



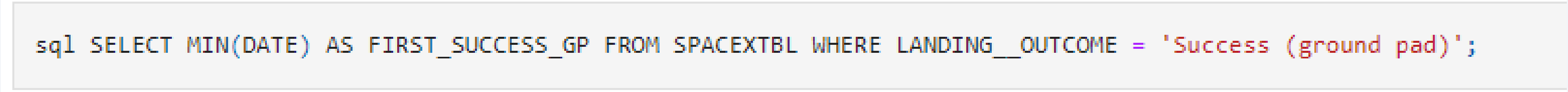
```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

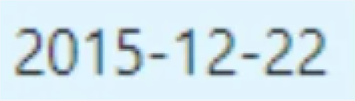
AVG_PAYLOAD
2928.4

First Successful Ground Landing Date

Using “MIN” we can get the first successful landing outcome. (MIN date will return the first minimum date)



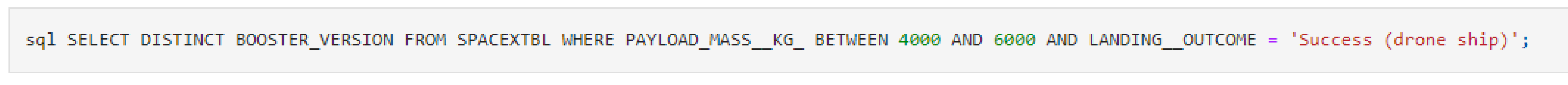
```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```



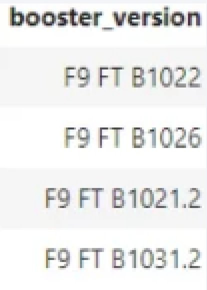
```
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

The payload mass data was only considered if the ranges where from 4000 to 6000 and if the outcome was successful



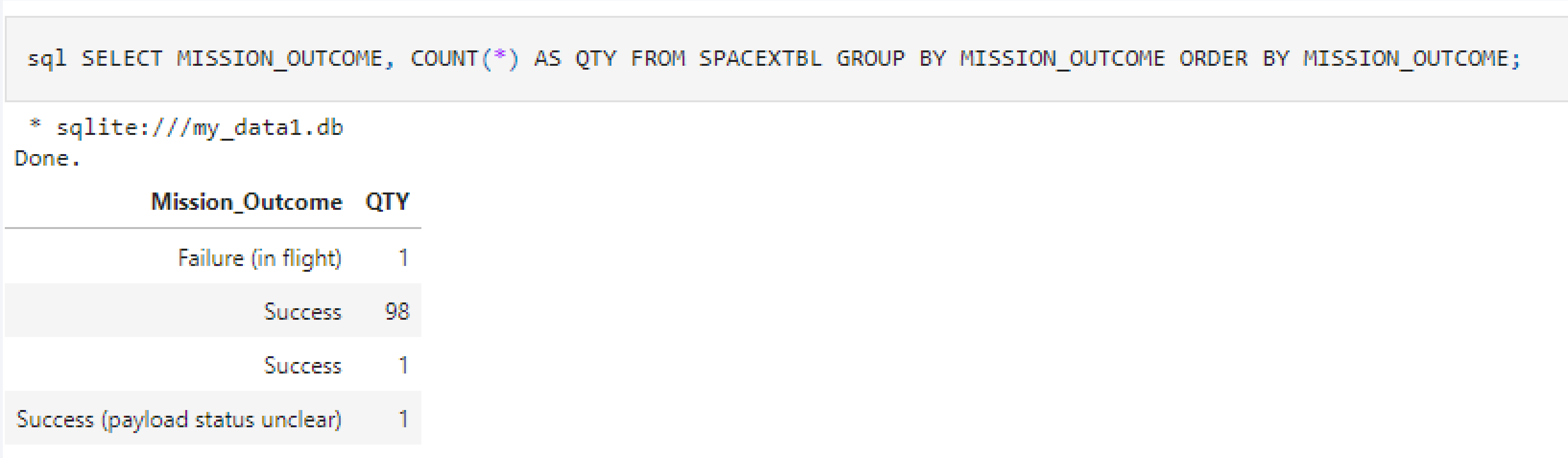
```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)';
```



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

Total Number of Successful and Failure Mission Outcomes

We looked at all the mission outcomes and used COUNT(*) to calculate this values.



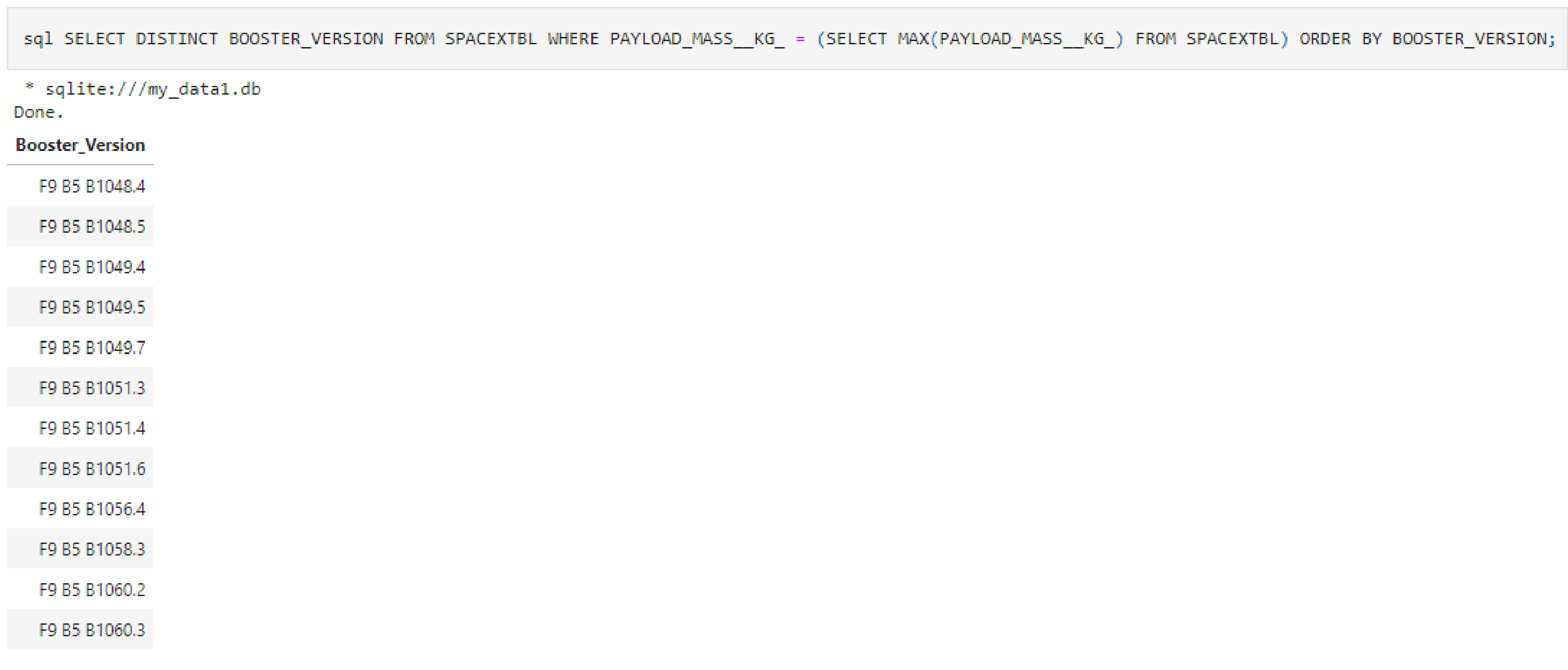
```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	QTY
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Using DISCTINT we could see every individual case, as well as using “MAX” we could also get the maximum payload masses .



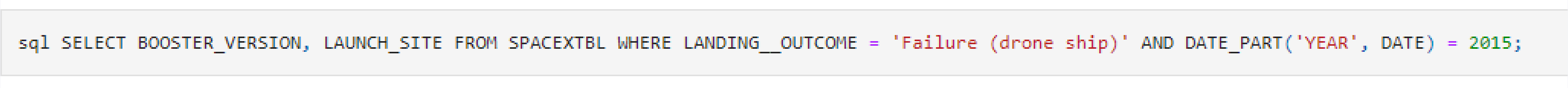
```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY BOOSTER_VERSION;
```

```
 * sqlite:///my_data1.db
Done.
```

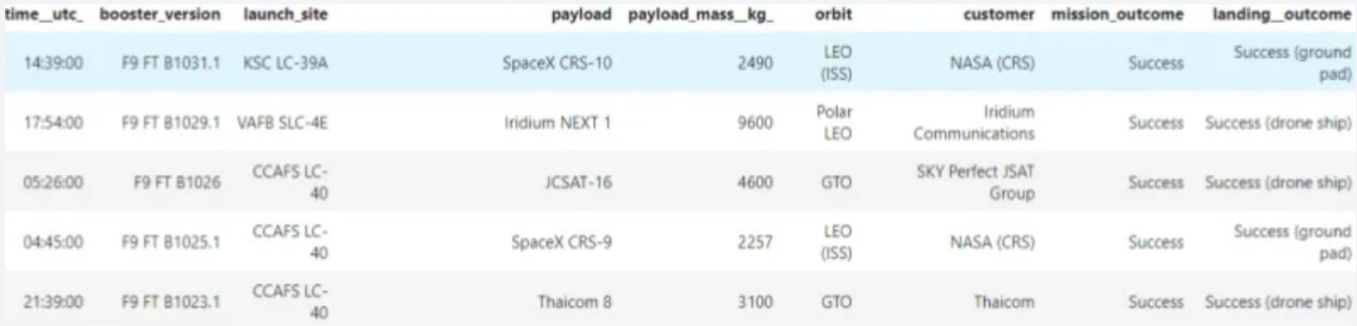
Booster_Version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

We got the year by using DATE_PART(‘YEAR’ , DATE) and making sure it was 2015



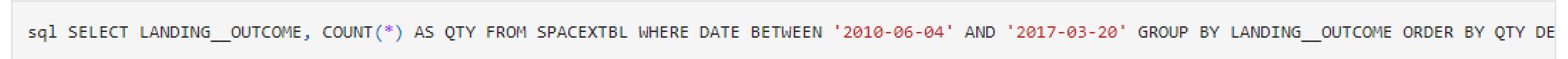
```
sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DATE) = 2015;
```



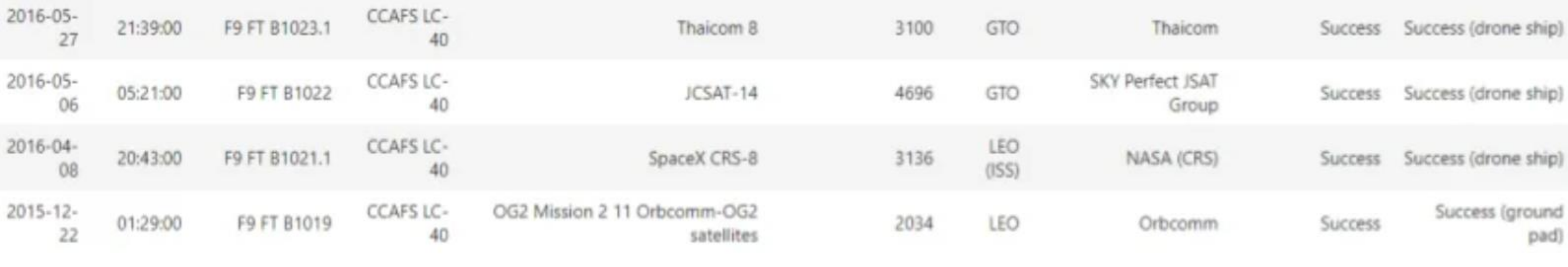
time_utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)

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

We ordered the dates between the 2 dates specified and used GROUP as well as ORDER in order to get the values in descending order



```
sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY QTY DE
```

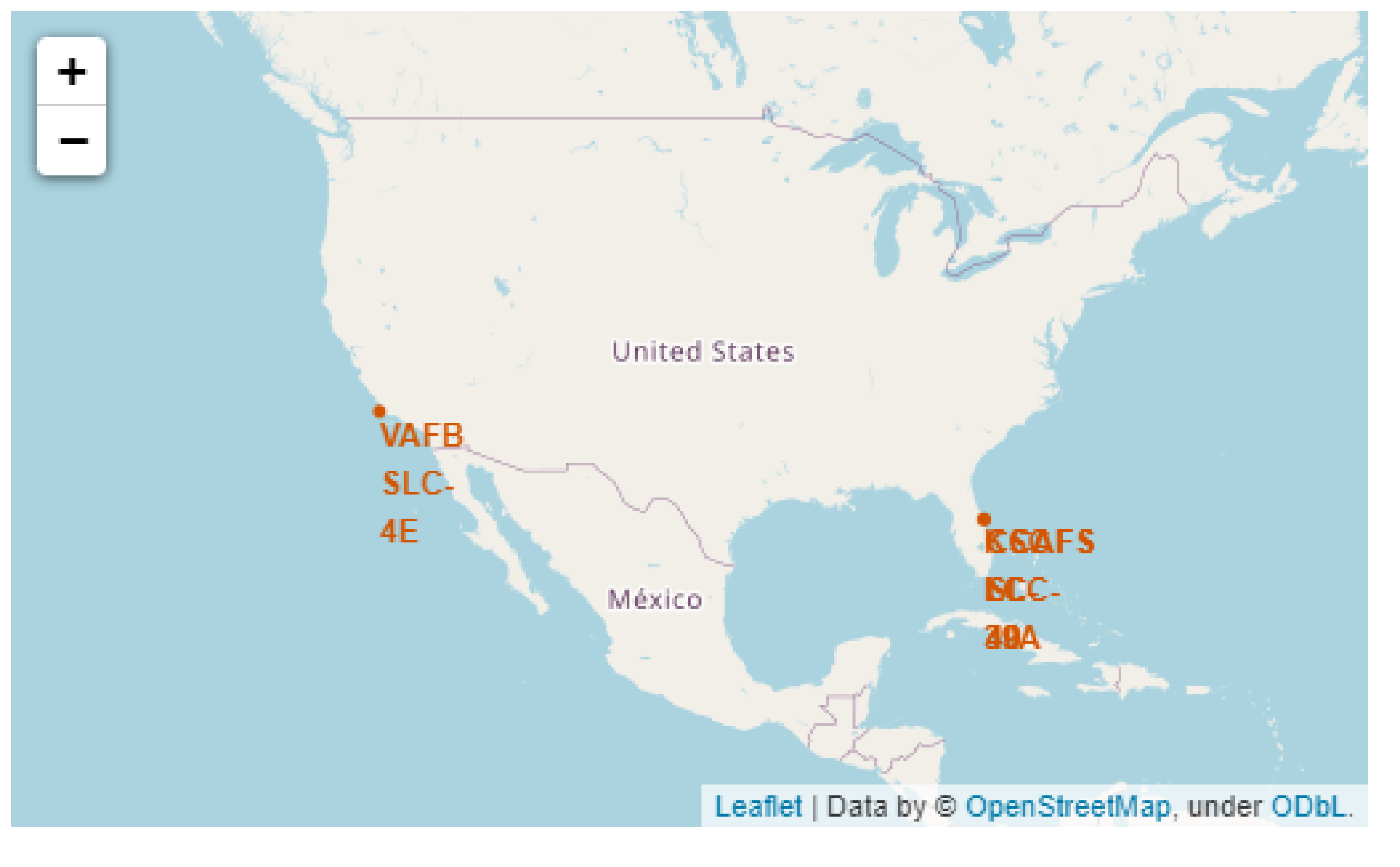


2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

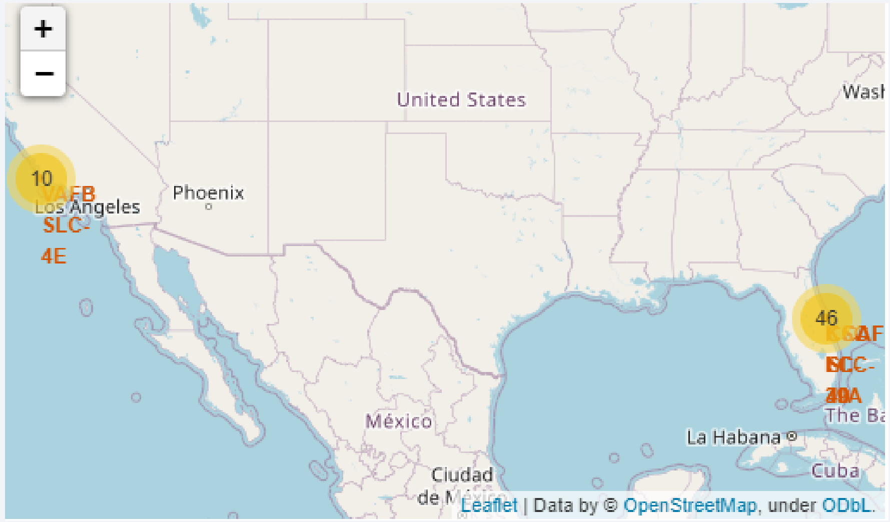
Launch Sites Proximities Analysis

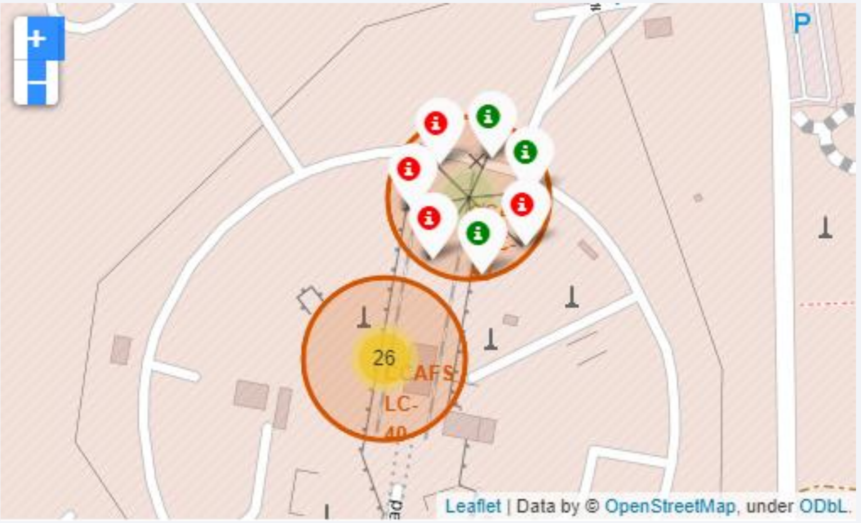
Marking all launch sites on a map



All the launch sites are displayed on the map

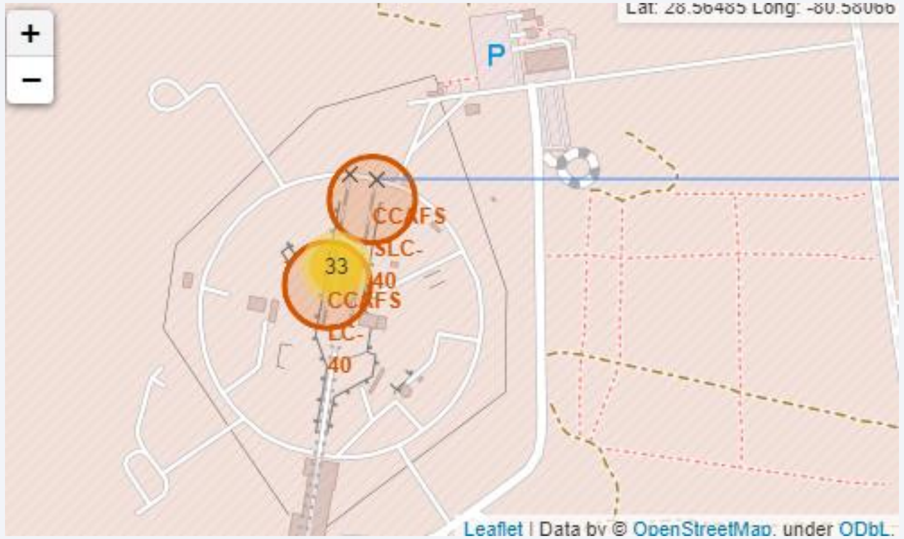
Success/failed launches for each site

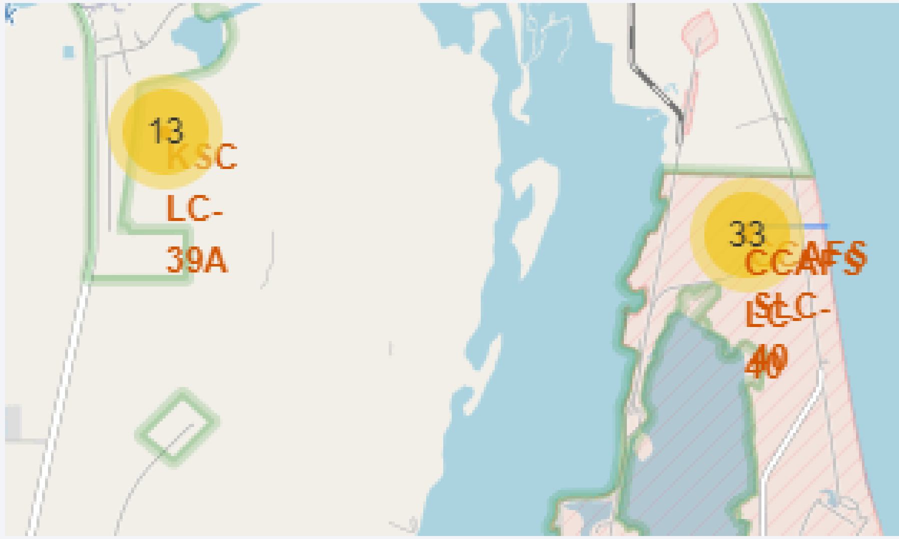




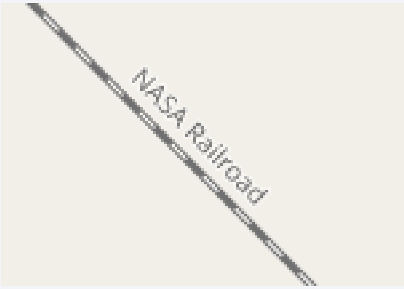
- By adding the launch outcomes for each site, and see which sites have high success rates.
- Green means it was successful.
- Red means it was a failure.

Distances between a launch site


Lat: 28.56485 Long: -80.58066
+
−
P
CCAFS
SLC-
40
33
CCAFS
LC-
40
Leaflet | Data by © OpenStreetMap, under ODbL.

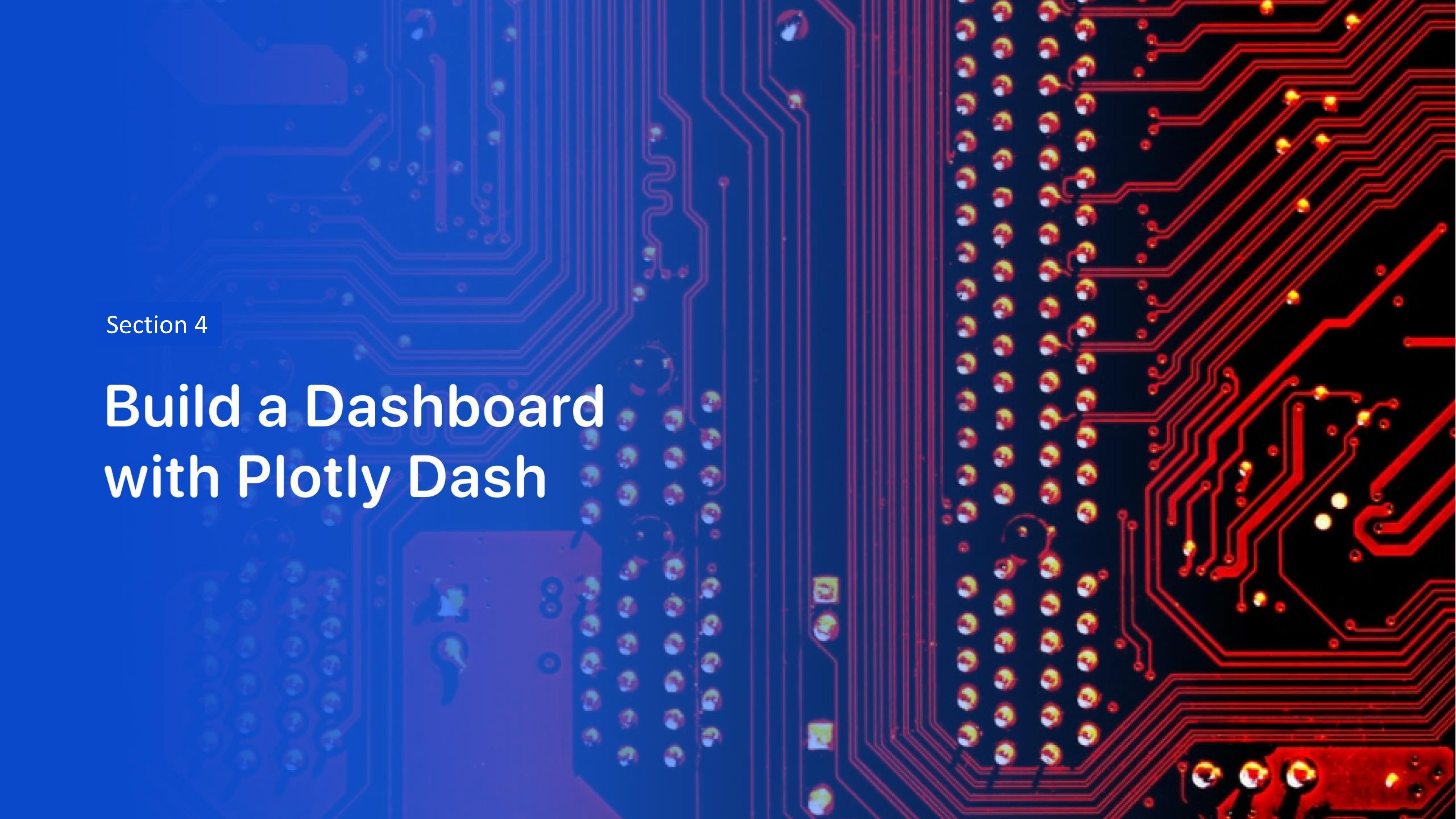


- By zooming in to a launch site and explore its proximity to see if you can easily find any railway, highway, coastline, etc.



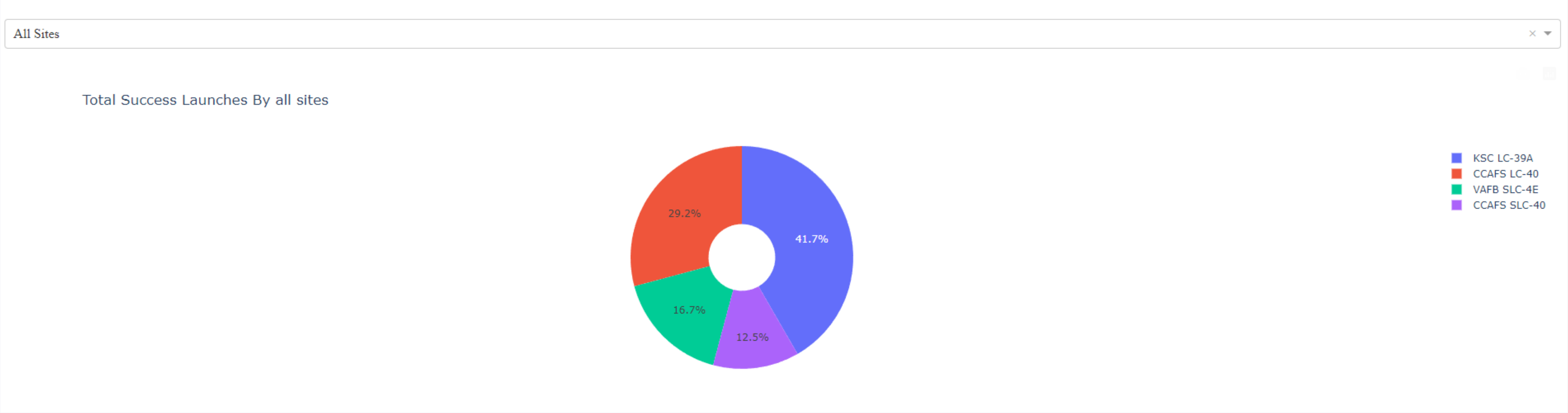
Section 4

Build a Dashboard with Plotly Dash



Total success launches by all sites

As we can see KSC LC-39A
had the most successful
launches

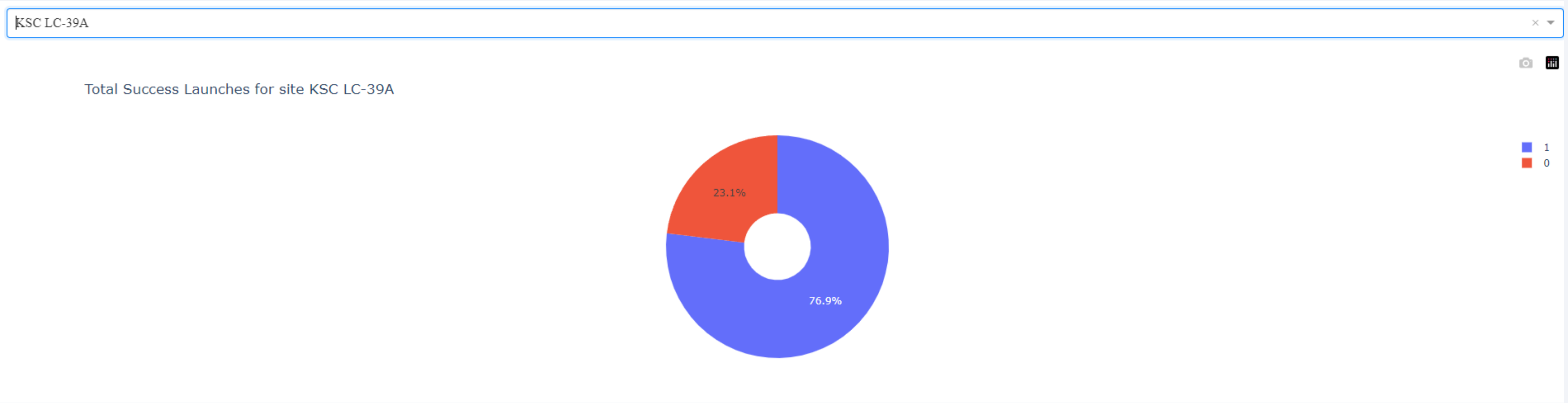


All Sites

Total Success Launches By all sites

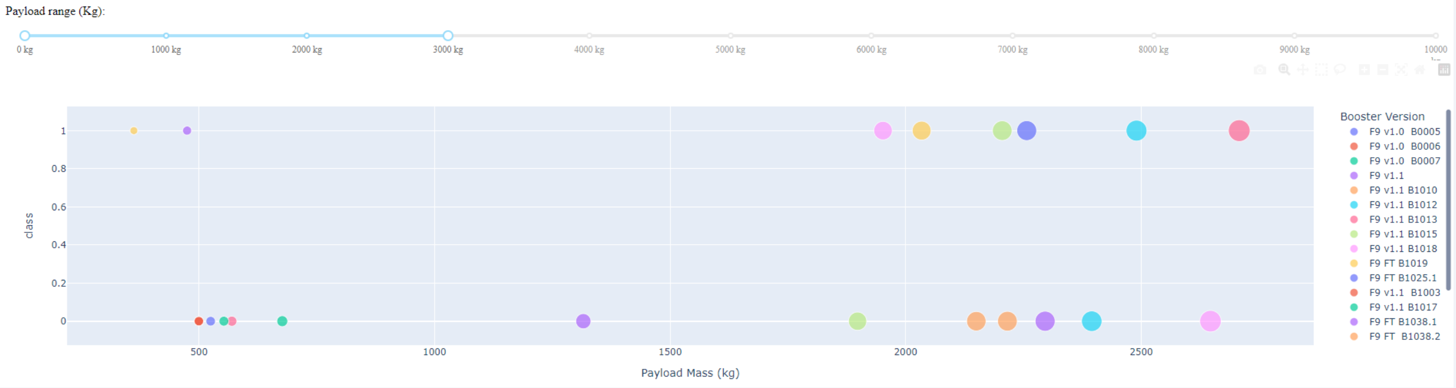
Success rate by highest site

KSC LC-39A has the highest score with a total of 76.9% success rate and only a 23.1% failure rate


KSC LC-39A

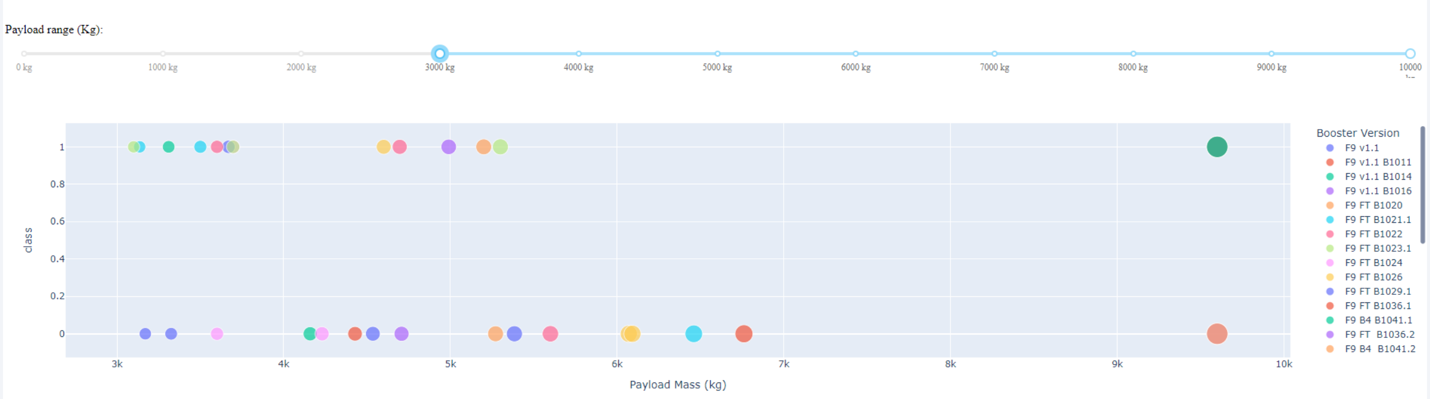
Total Success Launches for site KSC LC-39A

Payload vs. Launch Outcome



Payload from

0 kg – 3000 kg



Payload from

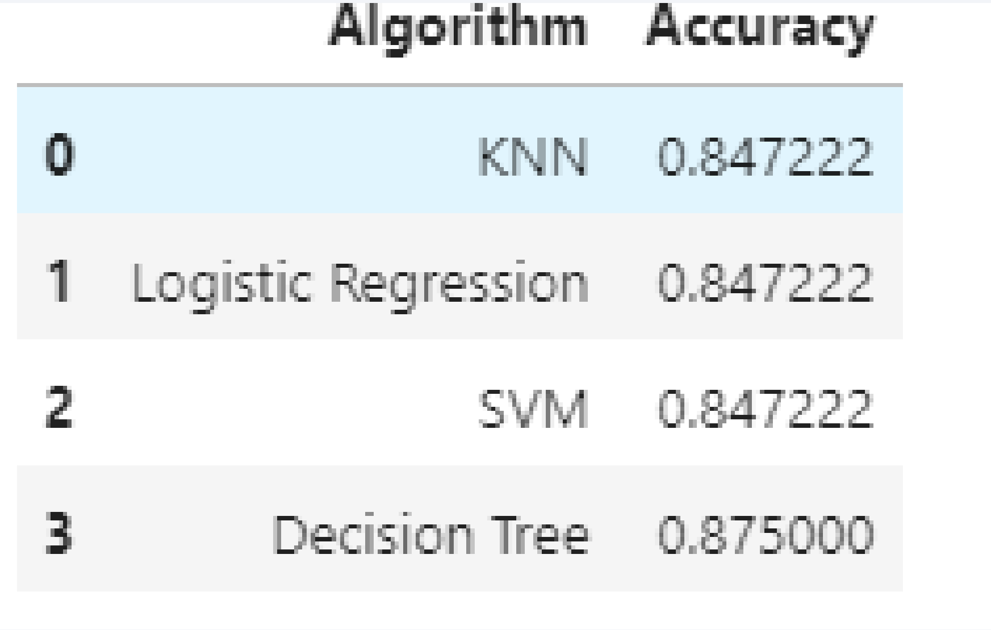
3000 kg – 10000 kg

Section 5

Predictive Analysis (Classification)

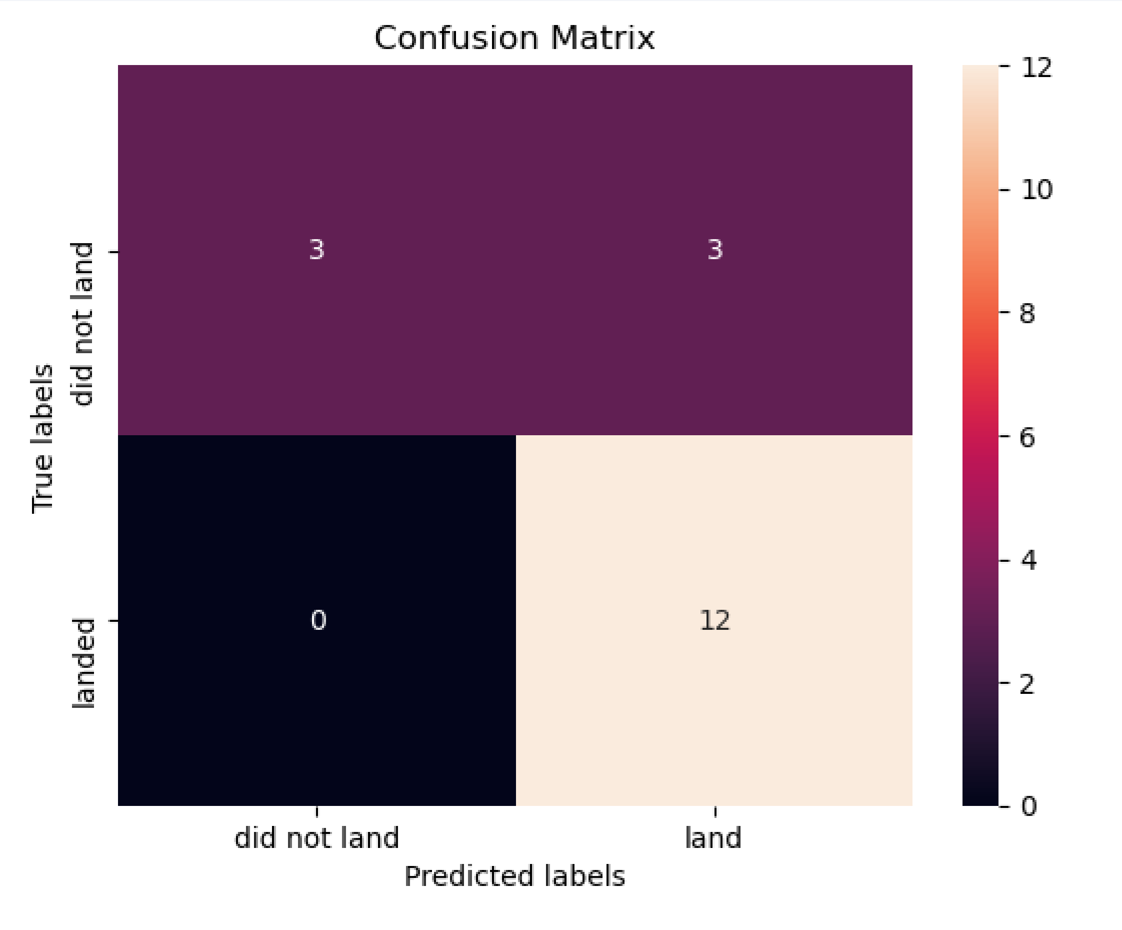
Classification Accuracy

The highest accuracy was the Decision Tree with close to 88% and the remaining models only had an 84%.



	Algorithm	Accuracy
0	KNN	0.847222
1	Logistic Regression	0.847222
2	SVM	0.847222
3	Decision Tree	0.875000

Confusion Matrix



Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

Conclusions

- KSC LC-39A had the most successful launches overall.
- The Decision tree classifier turn out to be the best predictor for the task at hand.
- The number of successful launches drastically increased throughout the years (2013-2020)
- Lower weighted payloads tended to perform better than those with heavier payloads.

Appendix

All code and notebook can be found on :

https://github.com/JorgeOnate/IBM-Applied-Data-Science-Capstone

Thank you!