IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Xinyi Li
7/14/2022

Outline

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

Executive Summary

- **Summary of methodologies**

1. Data Collection
2. Data Wrangling
3. Exploratory Data Analysis
4. Interactive Visual Analytics and Dashboard
5. Predictive Analysis

- **Summary of all results**

1. Exploratory data analysis results
2. Interactive analytics demo in screenshots
3. Predictive analysis results

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.

- Problems you want to find answers

What factors determine if the rocket will land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected through API and web scrapping from Wikipedia.
- Perform data wrangling
    - We performed EDA to find the patterns in the data and determine the training labels.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - We use GridSearchCV to find the best parameters.

Data Collection

We collected data sets through API and web scrapping from Wikipedia.

API

- Request to the SpaceX API
- Clean the requested data

Web Scrapping

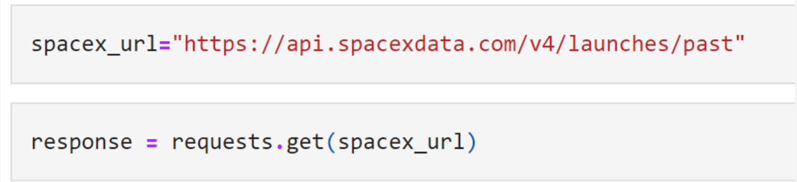
- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

Data Collection – SpaceX API

- We used GET request to request and parse the SpaceX launch data and did data wrangling by filling missing values.

- The GitHub URL of the completed SpaceX API calls notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Collection%20API.ipynb

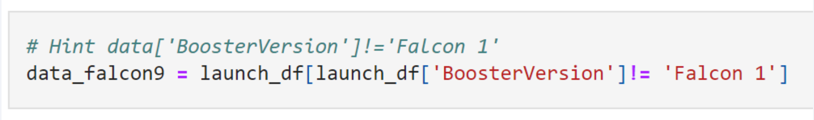
1. Request and parse the SpaceX launch data using the GET request



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

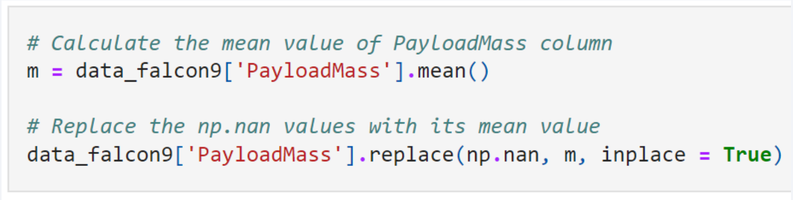
```
response = requests.get(spacex_url)
```

2. Filter the dataframe to only include Falcon 9 launches



```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion']!= 'Falcon 1']
```

3. Data Wrangling (Dealing with Missing Values)



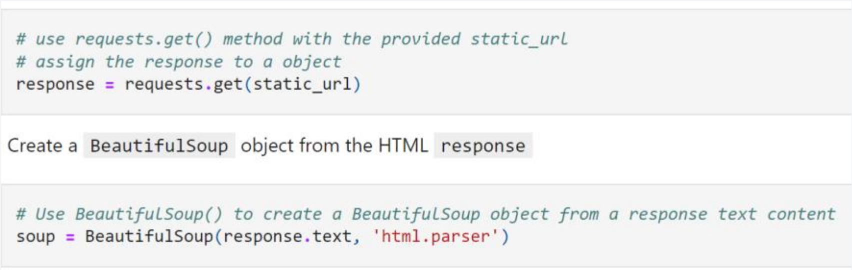
```
# Calculate the mean value of PayloadMass column
m = data_falcon9['PayloadMass'].mean()

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

Data Collection - Scraping

- We Extracted a Falcon 9 launch records HTML table from Wikipedia and parsed the table and converted it into a Pandas data frame.
- The GitHub URL of the completed web scraping notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

1. Request the Falcon9 Launch Wiki page from its URL

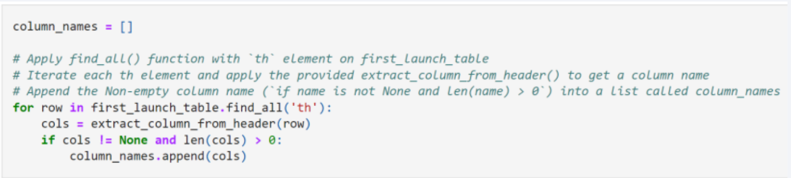


```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

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

2. Extract all column/variable names from the HTML table header



```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for row in first_launch_table.find_all('th'):
    cols = extract_column_from_header(row)
    if cols != None and len(cols) > 0:
        column_names.append(cols)
```

3. Create a data frame by parsing the launch HTML tables

Data Wrangling

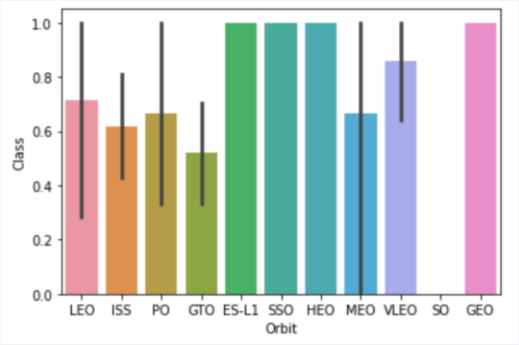
- In Data Wrangling, we performed two main tasks:

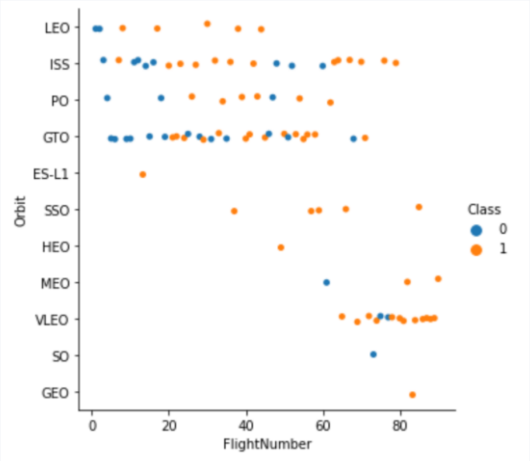
1. Exploratory Data Analysis
   - Calculate the number of launches on each site
   - Calculate the number and occurrence of each orbit
   - Calculate the number and occurence of mission outcome per orbit type
2. Determine Training Labels
   - Create a landing outcome label from Outcome column

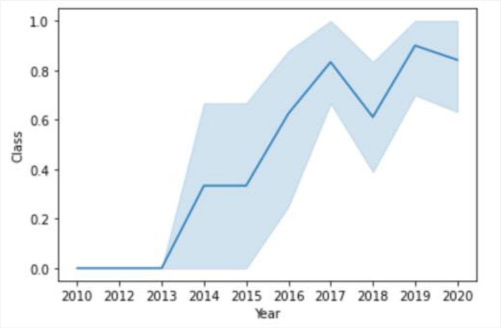
- The GitHub URL of the completed data wrangling related notebooks is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

- In Exploratory Data Analysis for Data Visualization, we used scatterplots and barplot to see how and if different variables such as flight numbers, payload mass and orbit would affect launch outcome. Also, we used lineplot to visualize the launch success yearly trend.
- The GitHub URL of the completed EDA with data visualization notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Exploratory%20Data%20Analysis%20for%20Data%20Visualization.ipynb





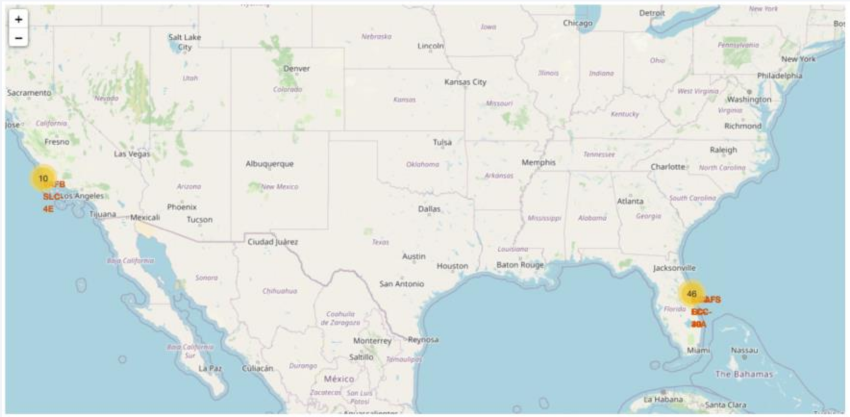


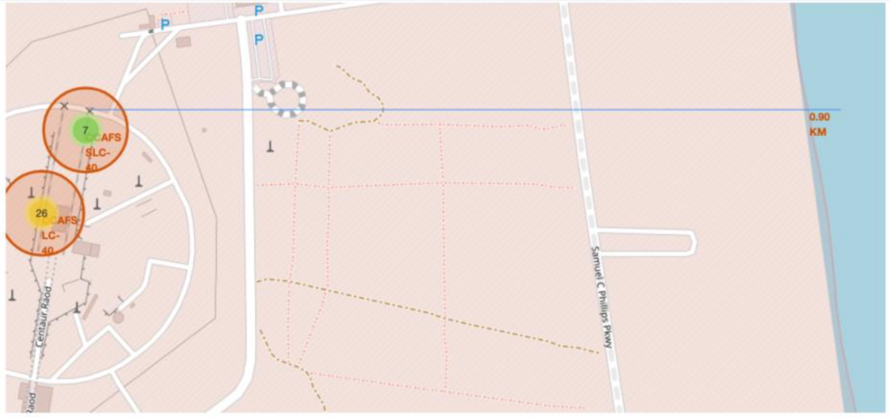
EDA with SQL

- We loaded the dataset in a Db2 database and performed the SQL queries such as:
  - Display the names of the unique launch sites in the space mission
  - List the date when the first succesful landing outcome in ground pad was achieved
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass
  - 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
- The GitHub URL of the completed EDA with SQL notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Exploratory%20Data%20Analysis%20Using%20SQL_Sqllite.ipynb

Build an Interactive Map with Folium

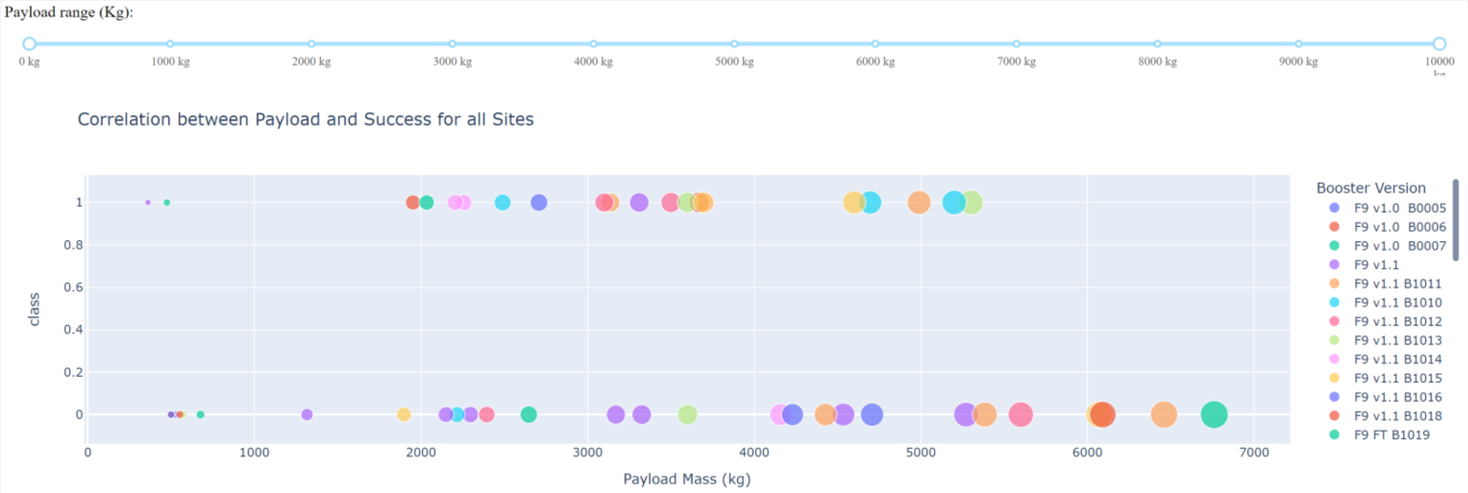
- We used markers and circles to mark all launch sites and the success/failed launches for each site on the map. Then we calculated the distances between a launch site to its proximities and marked it using lines.
- The GitHub URL of the completed interactive map with Folium map notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb





Build a Dashboard with Plotly Dash

- We built SpaceX Launch Records Dashboard. In the dashboard, we used pie charts to visualize launch success by sites and used scatterplot to demonstrate correlation between payload and success for all sites.
- The GitHub URL of the completed Plotly Dash lab notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Interactive%20Dashboard%20with%20Ploty%20Dash.py



Predictive Analysis (Classification)

- In Predictive Analysis, we performed Exploratory Data Analysis and determined Training Labels:
    - Create a column for the class
    - Standardize the data
    - Split into training data and test data
- Then, we found best hyperparameter for SVM, Classification Trees and Logistic Regression using GridSearchCV method.
- The GitHub URL of the completed predictive analysis lab notebook is https://github.com/BeatriceXL/IBM-Applied-Data-Science-Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

Results

See section 2 for details:

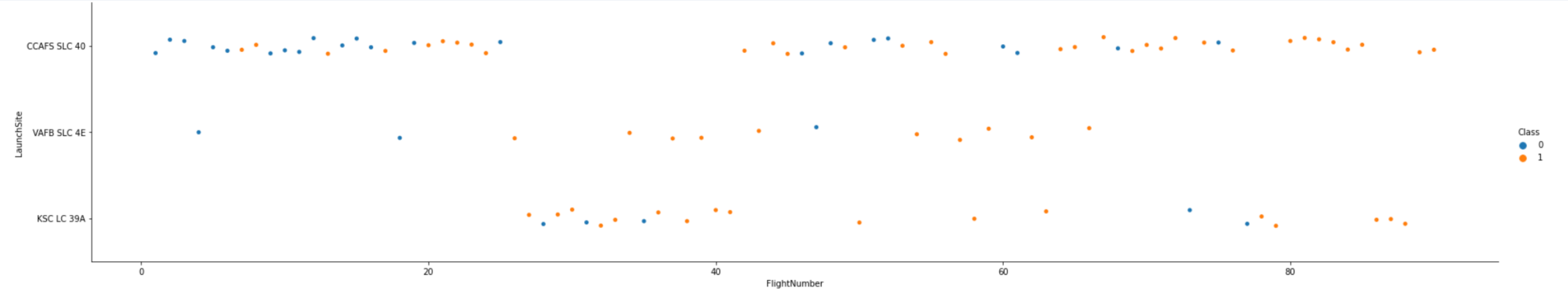
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

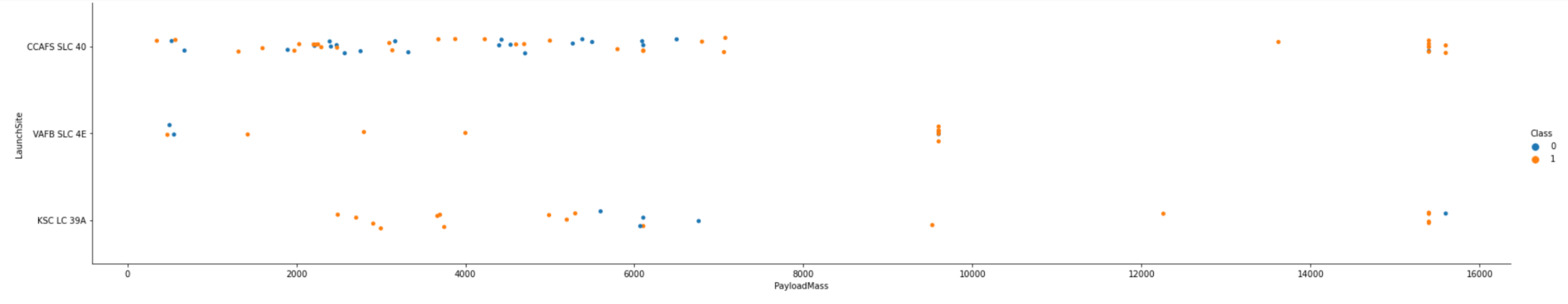
Flight Number vs. Launch Site

- From the scatterplot, we found that the larger the flight number at a launch site, the greater the success rate.



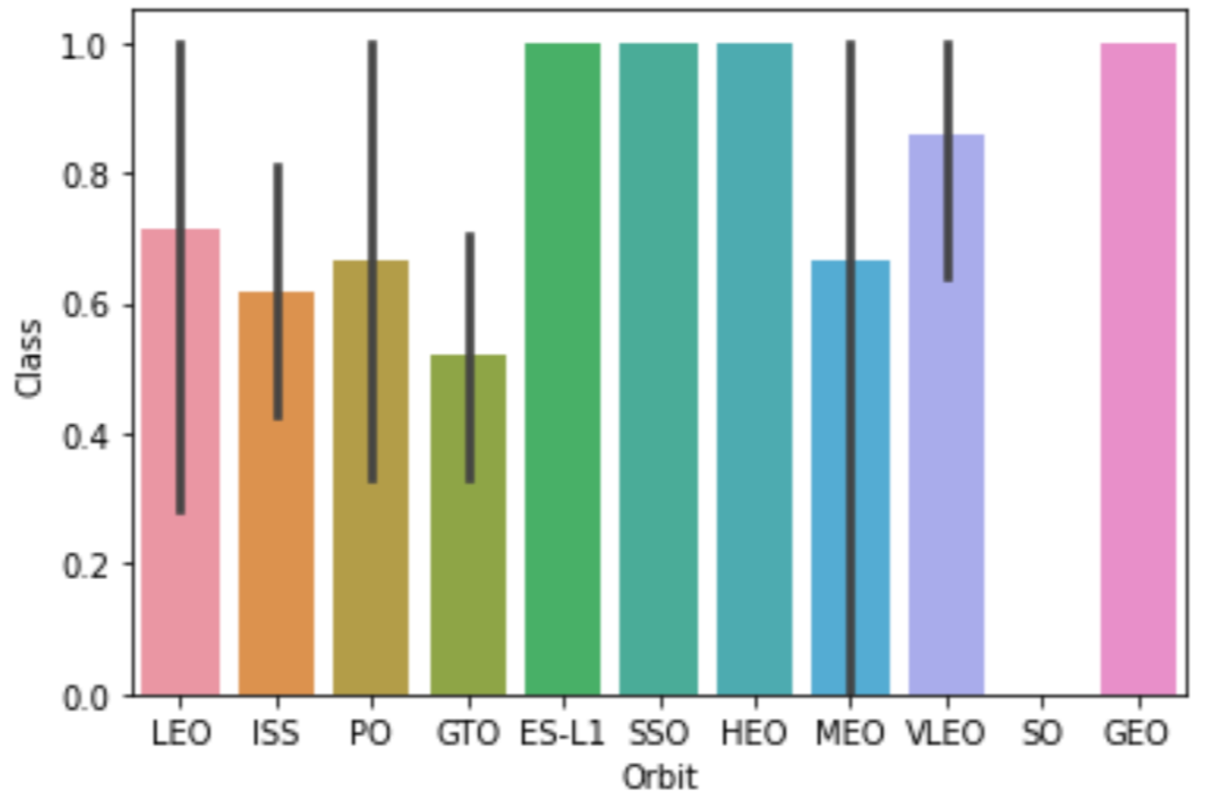
Payload vs. Launch Site

- From the scatterplot, we found that the larger the payloadmass at a launch site, the greater the success rate, especially for site CCAFS SLC 40 and site VAFB SLC 4E.
- Also, for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



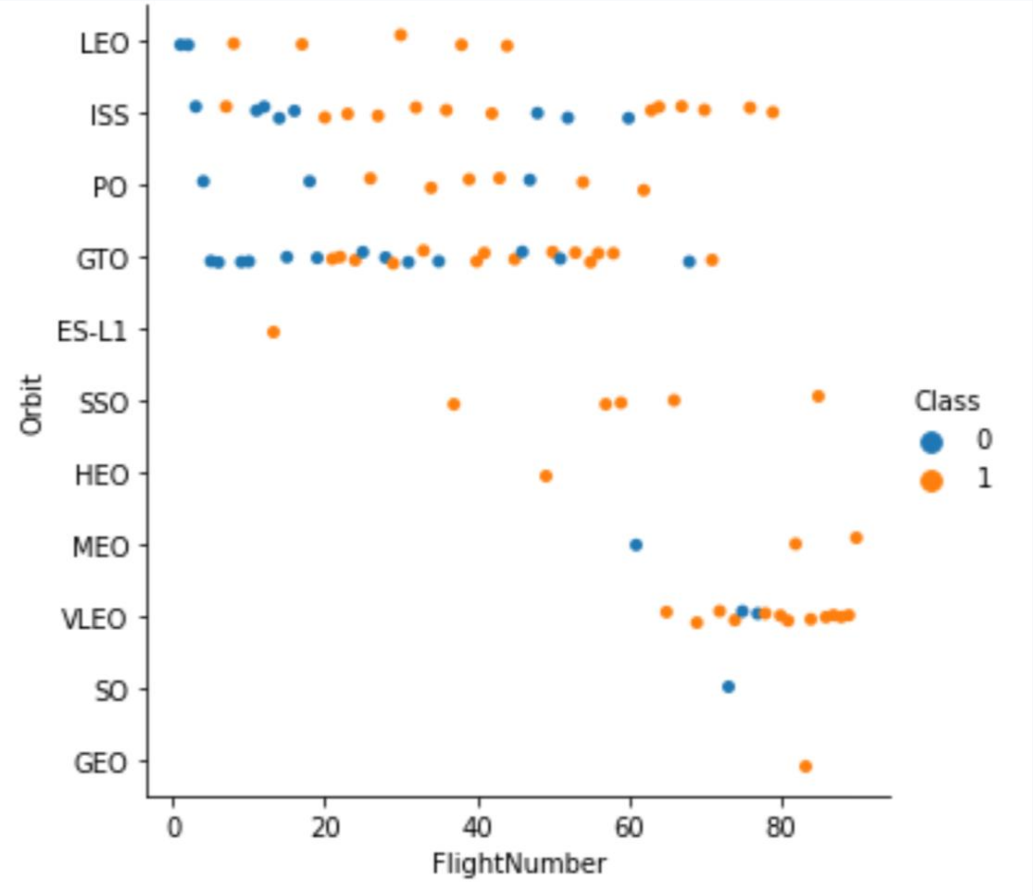
Success Rate vs. Orbit Type

- From the barplot, we can see that ES-L1, SSO, HEO and GEO had the most success rate.



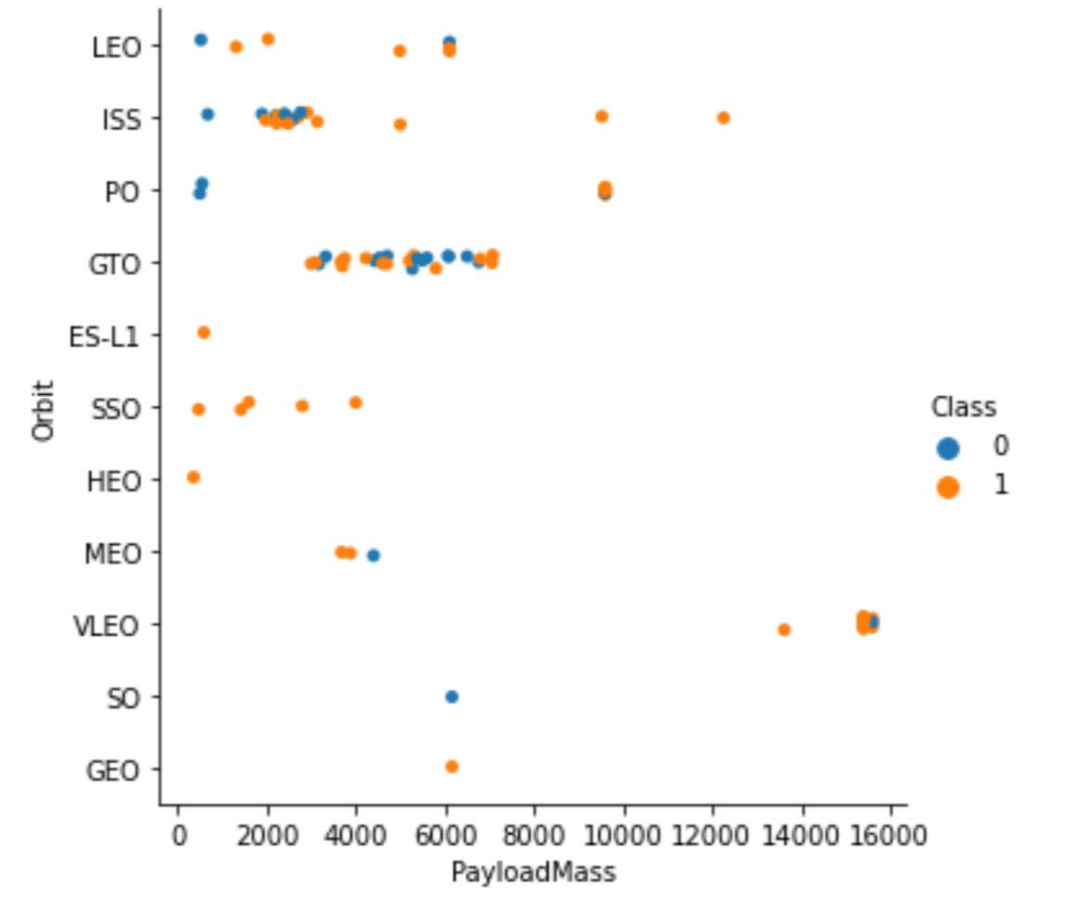
Flight Number vs. Orbit Type

- We observe 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.



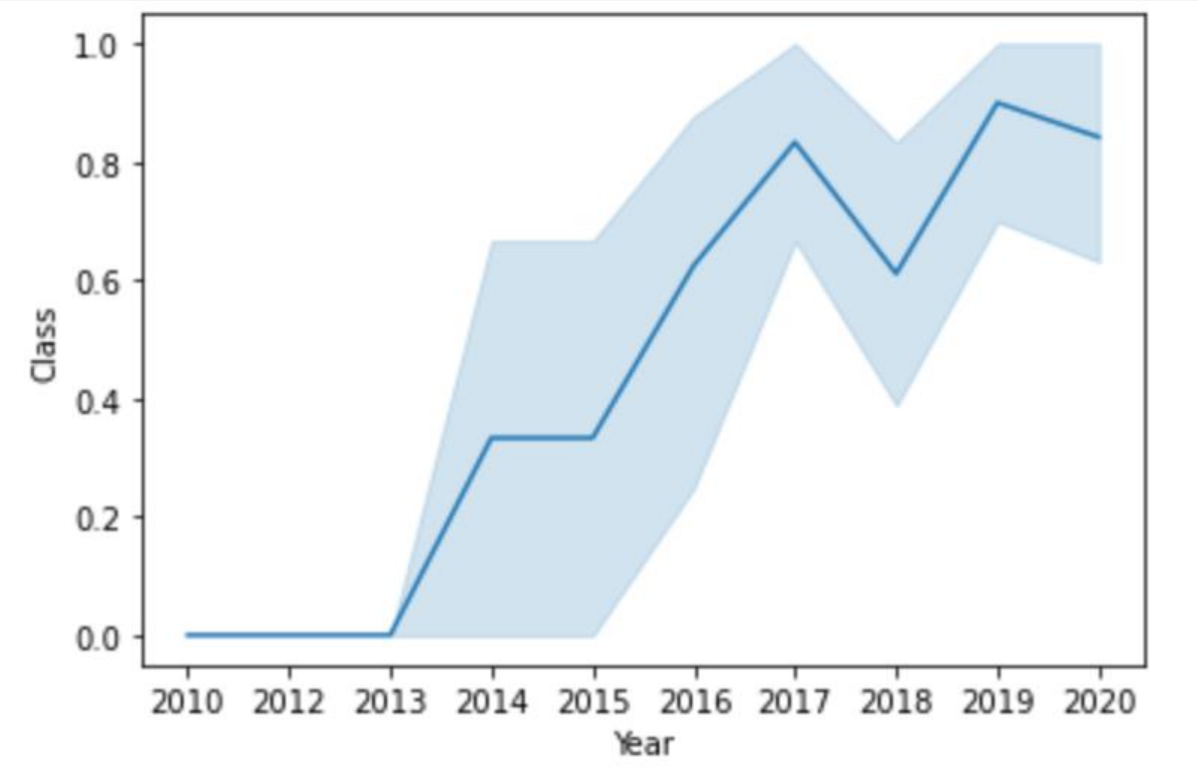
Payload vs. Orbit Type

- 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.



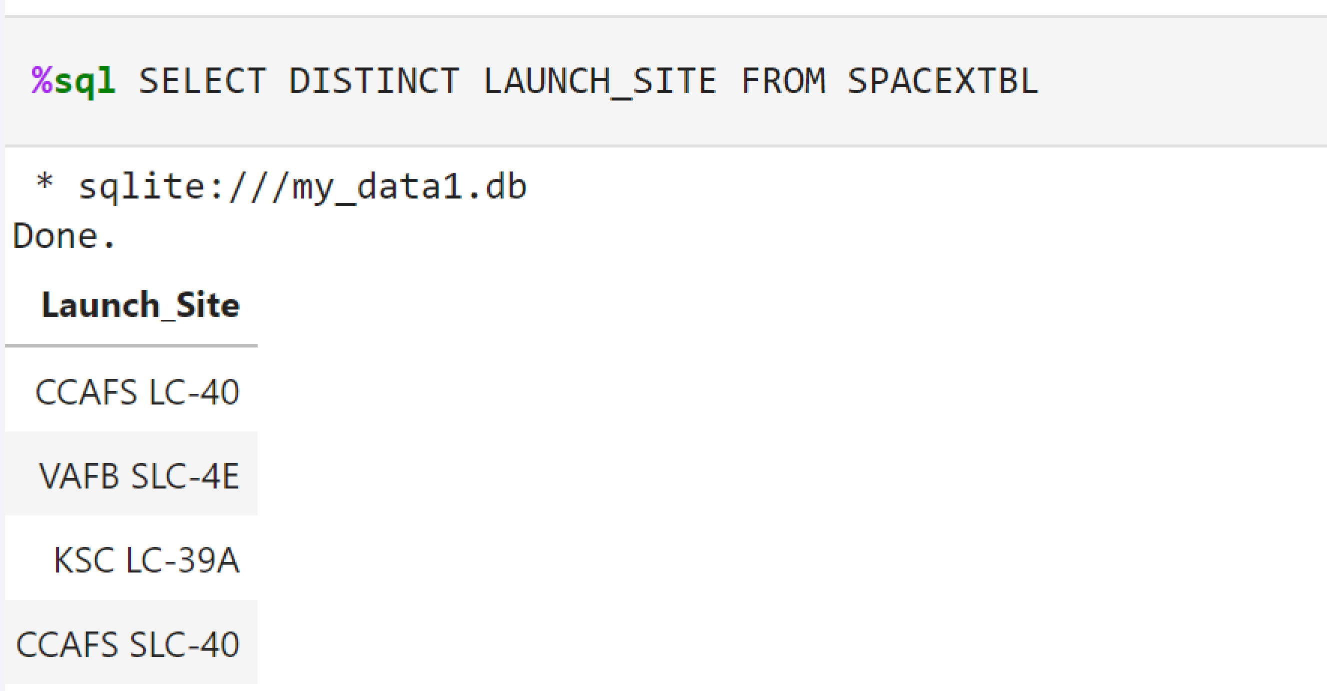
Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020 in spite of a drop in 2018.



All Launch Site Names

- We used DISTINCT function to find the unique values.



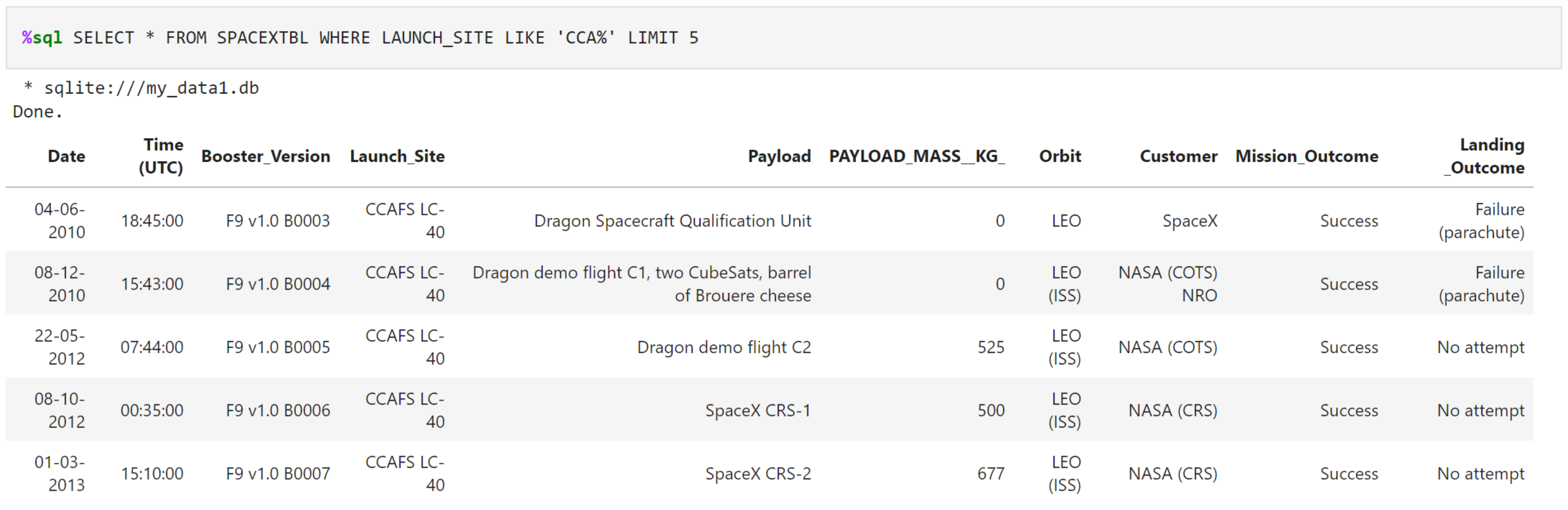
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- We matched ‘CCA’ using LIKE and % function.



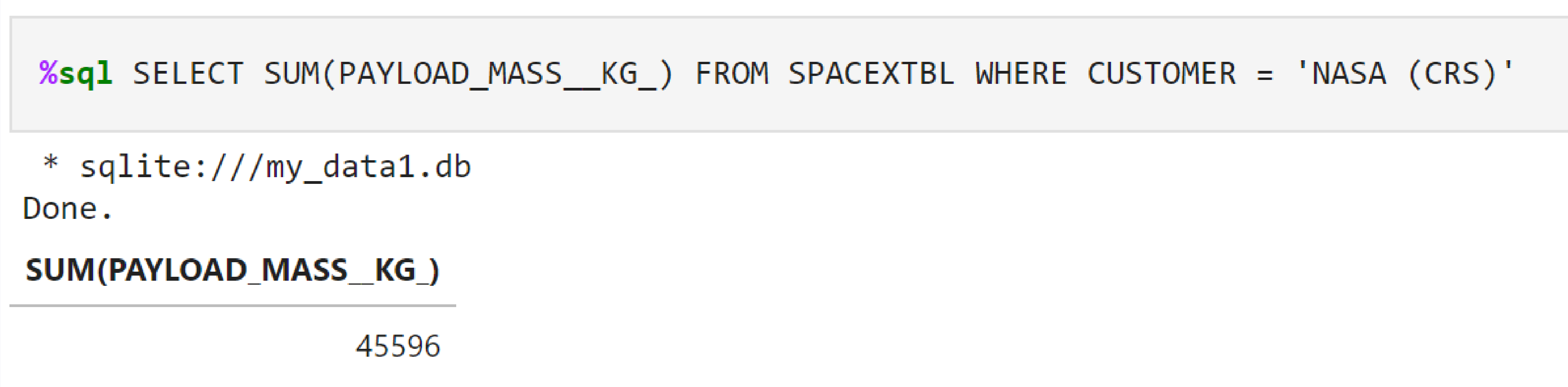
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * 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 used SUM function to calculate the total payload carried by boosters from NASA.



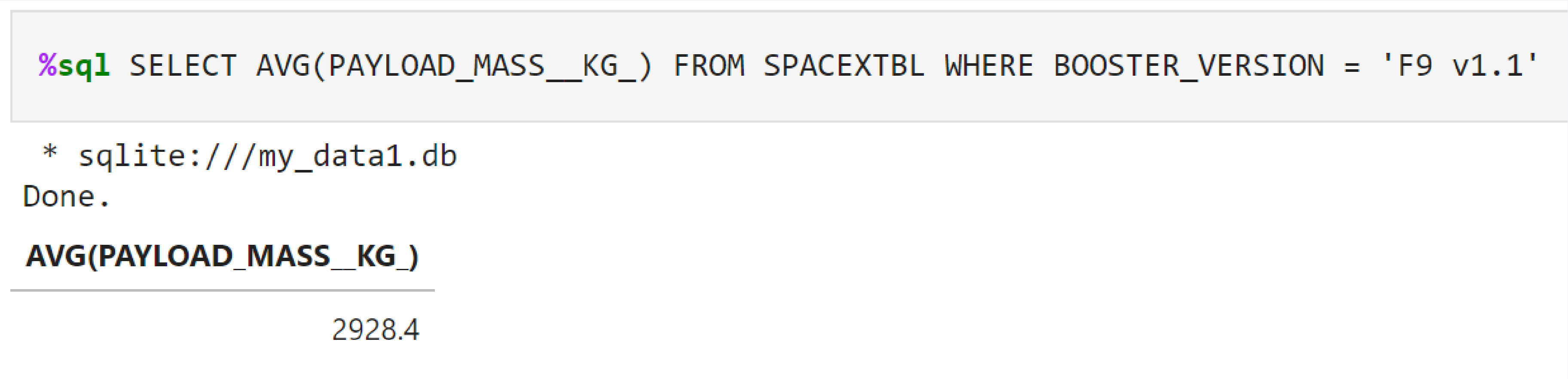
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We used AVG function to calculate the average payload mass carried by booster version F9 v1.1



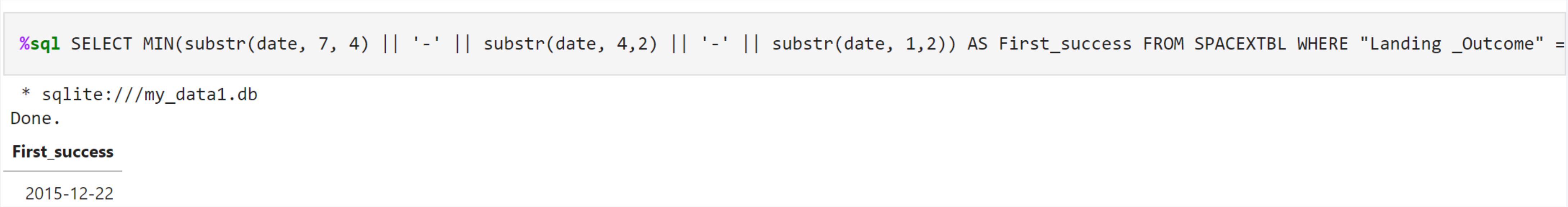
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Because SQLLite does not support monthnames, we need to use substr(Date,7,4) to get year, substr(Date, 4, 2) for month and substr(Date, 1, 2) for day.



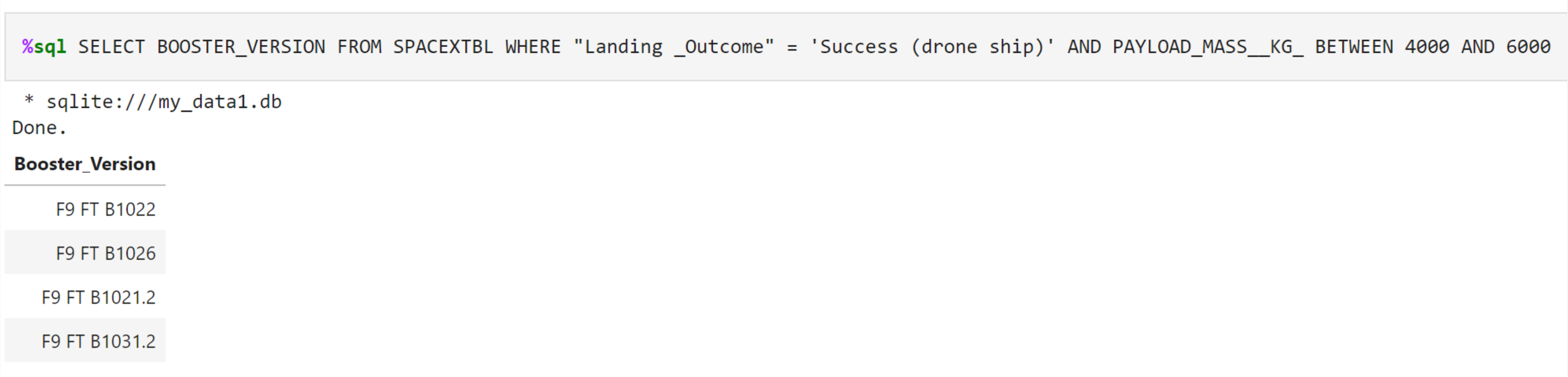
```
%sql SELECT MIN(substr(date, 7, 4) || '-' || substr(date, 4,2) || '-' || substr(date, 1,2)) AS First_success FROM SPACEXTBL WHERE "Landing _Outcome" =
```

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

First_success
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause and BETWEEN condition to filter for boosters which have successfully landed on drone ship with payload mass greater than 4000 but less than 6000



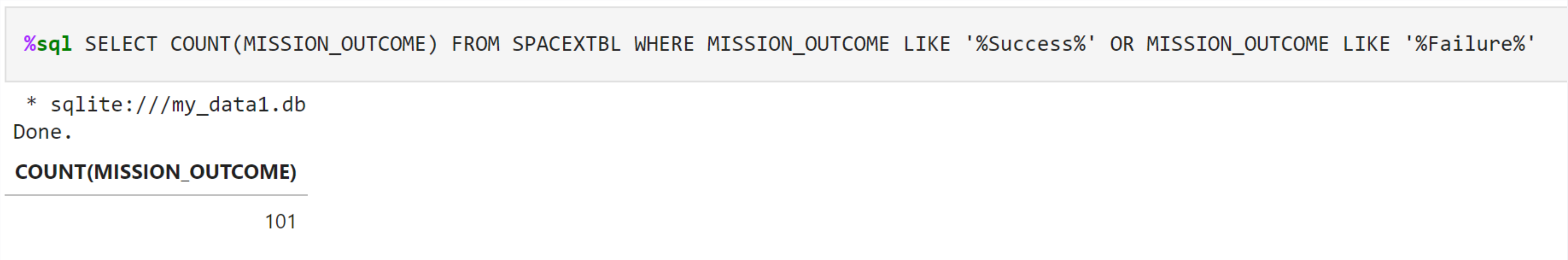
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

- We used COUNT function to calculate the total number of successful and failure mission outcomes



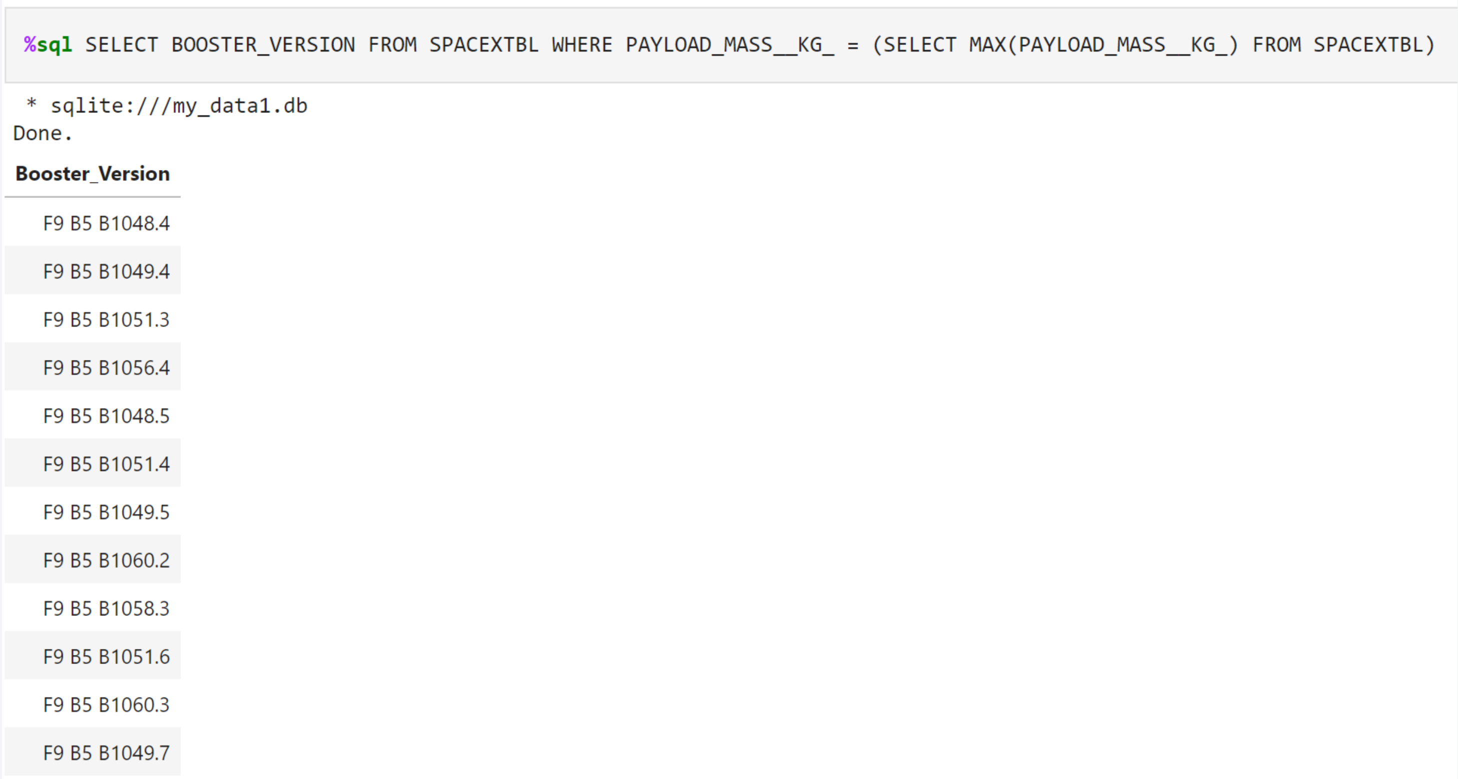
```
%sql SELECT COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

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

COUNT(MISSION_OUTCOME)
101

Boosters Carried Maximum Payload

- We used subquery to find the names of the booster which have carried the maximum payload mass



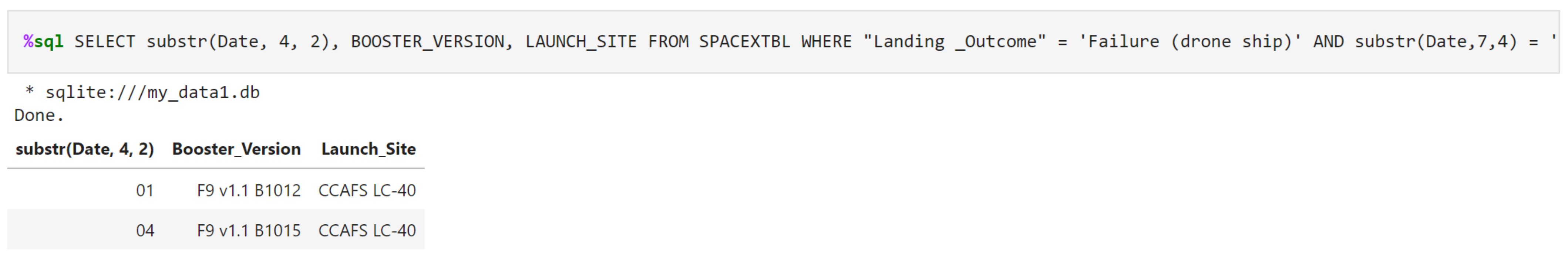
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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



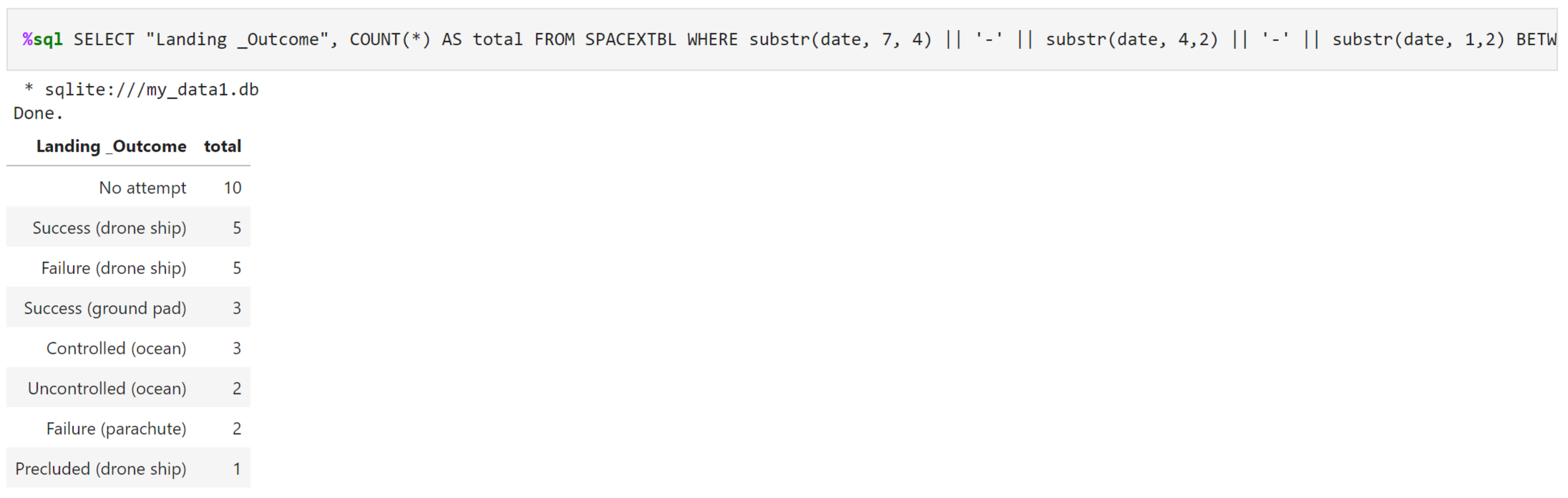
```
%sql SELECT substr(Date, 4, 2), BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE "Landing _Outcome" = 'Failure (drone ship)' AND substr(Date,7,4) = '
```

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

substr(Date, 4, 2)	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	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



```
%sql SELECT "Landing _Outcome", COUNT(*) AS total FROM SPACEXTBL WHERE substr(date, 7, 4) || '-' || substr(date, 4,2) || '-' || substr(date, 1,2) BETW
```

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

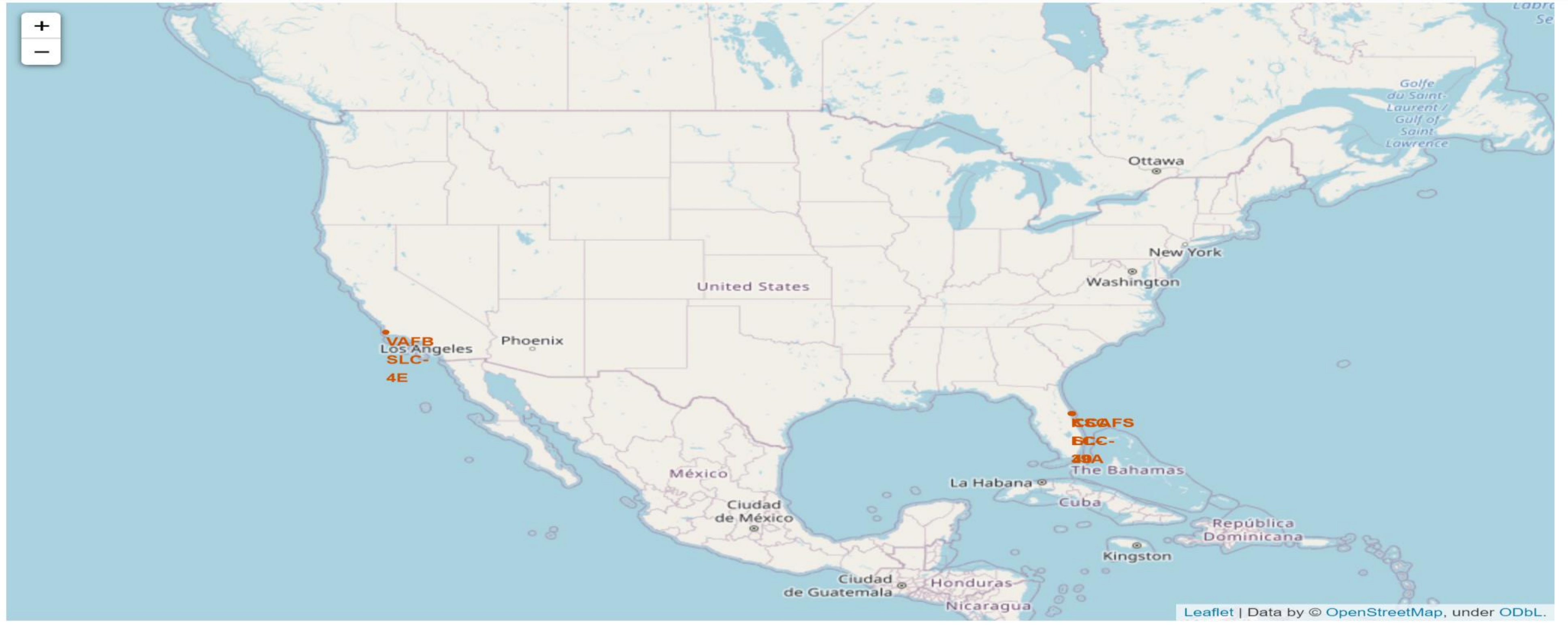
Landing _Outcome	total
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

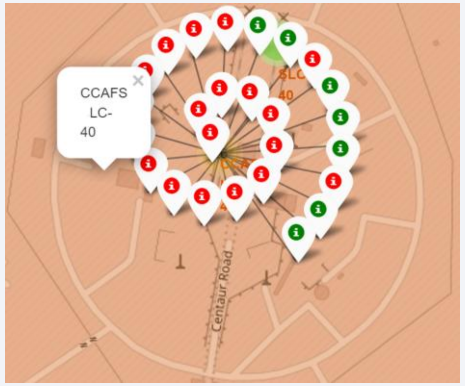
All launch sites map

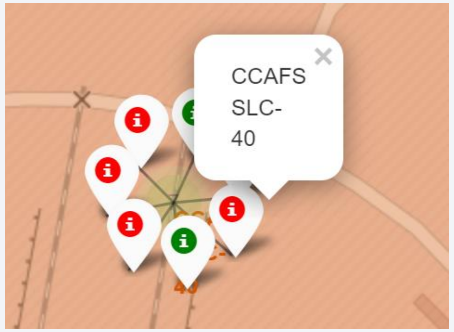
- We found that all launch sites are near coast.

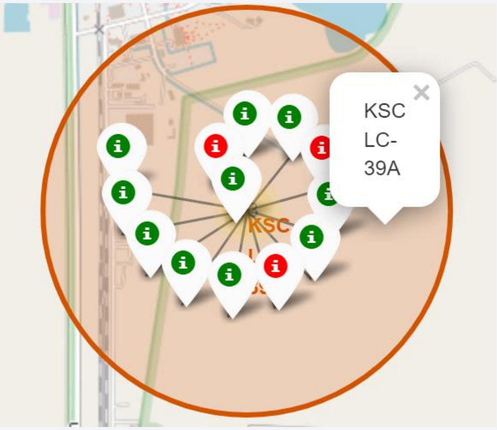


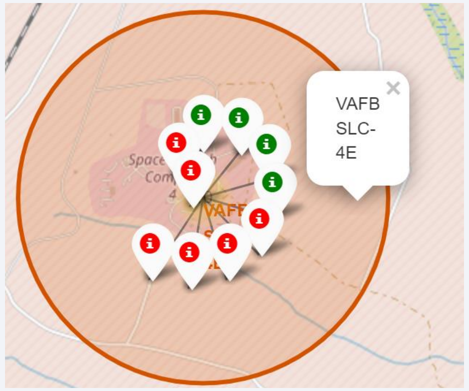
Color-labeled launch outcomes by sites

- Green markers show successful launches and red markers show failed launches.



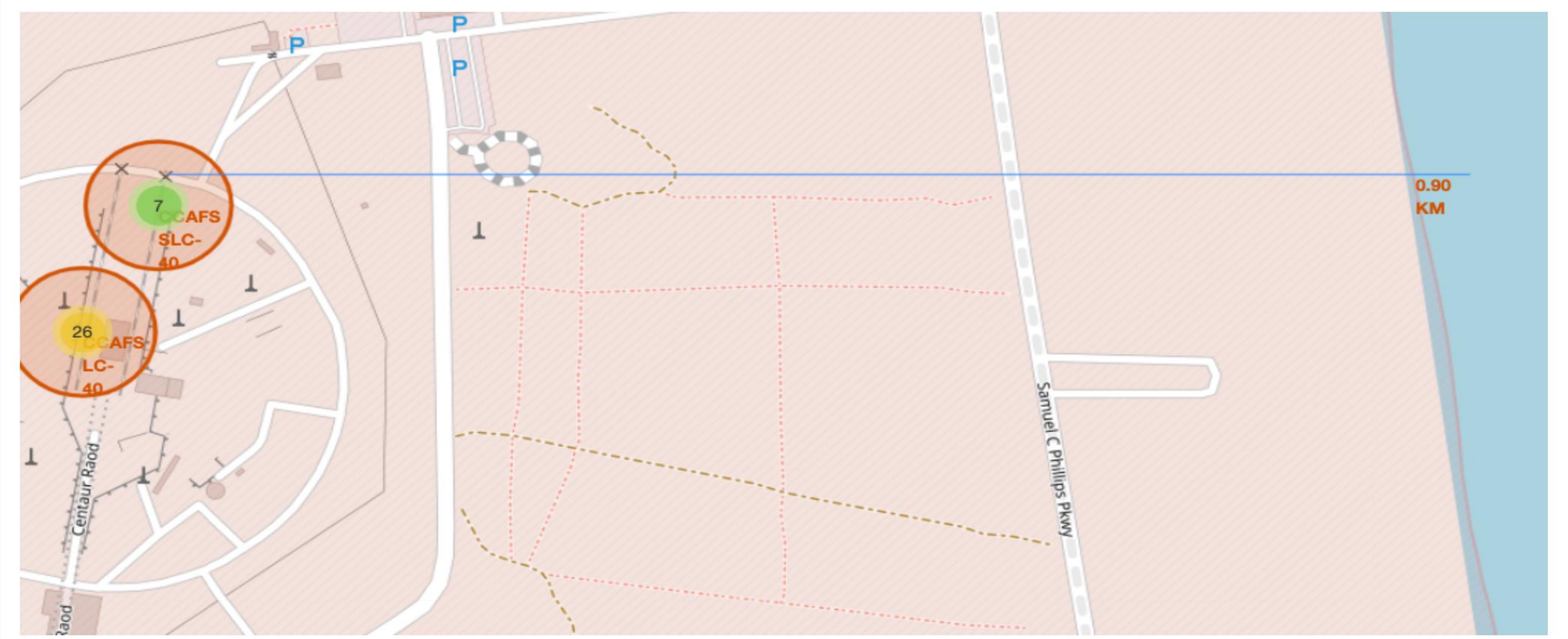


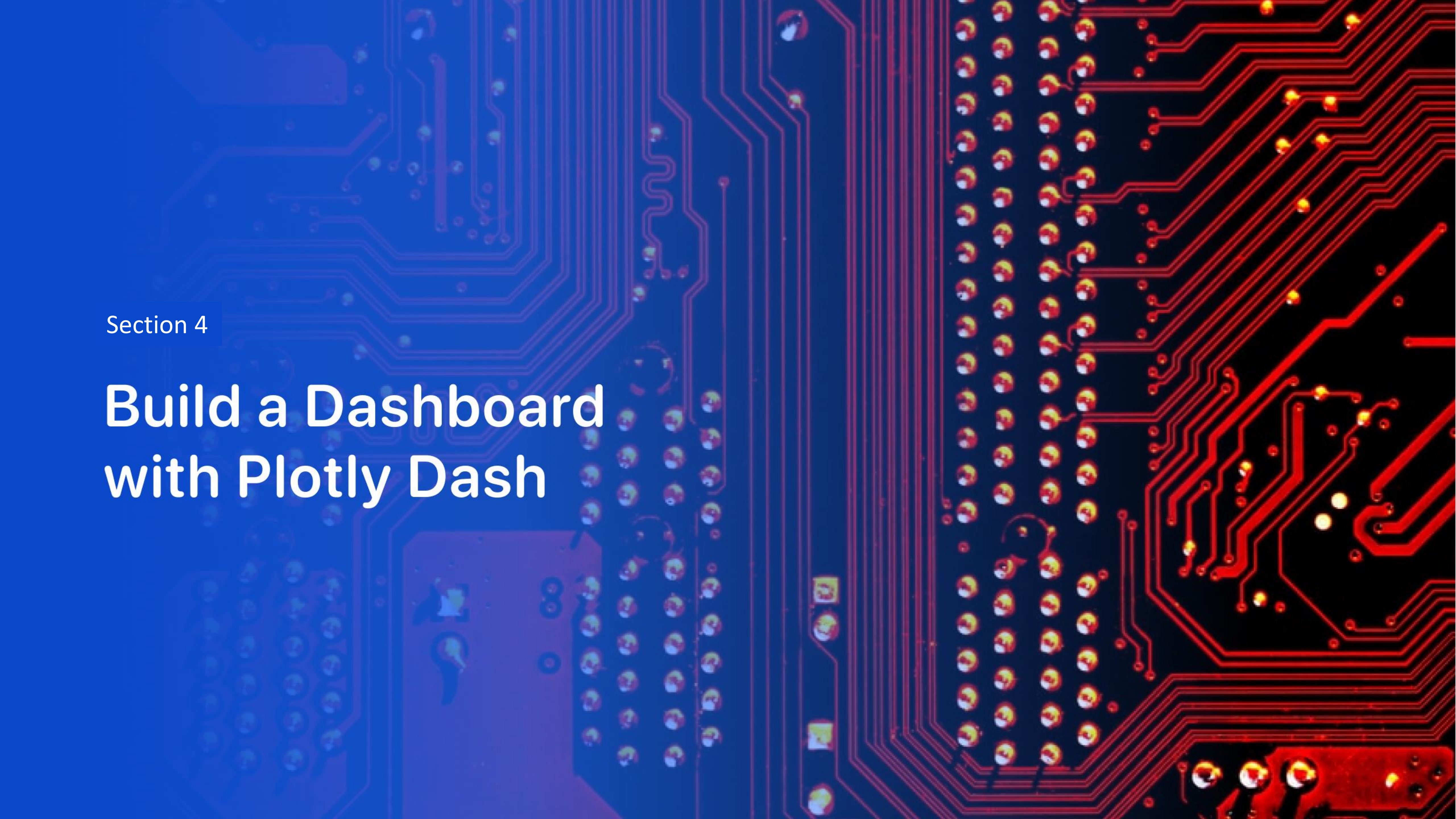




Distance of a site to its proximities

- For example, the distance of CCAFS SLC-40 to coast line is 0.9km.



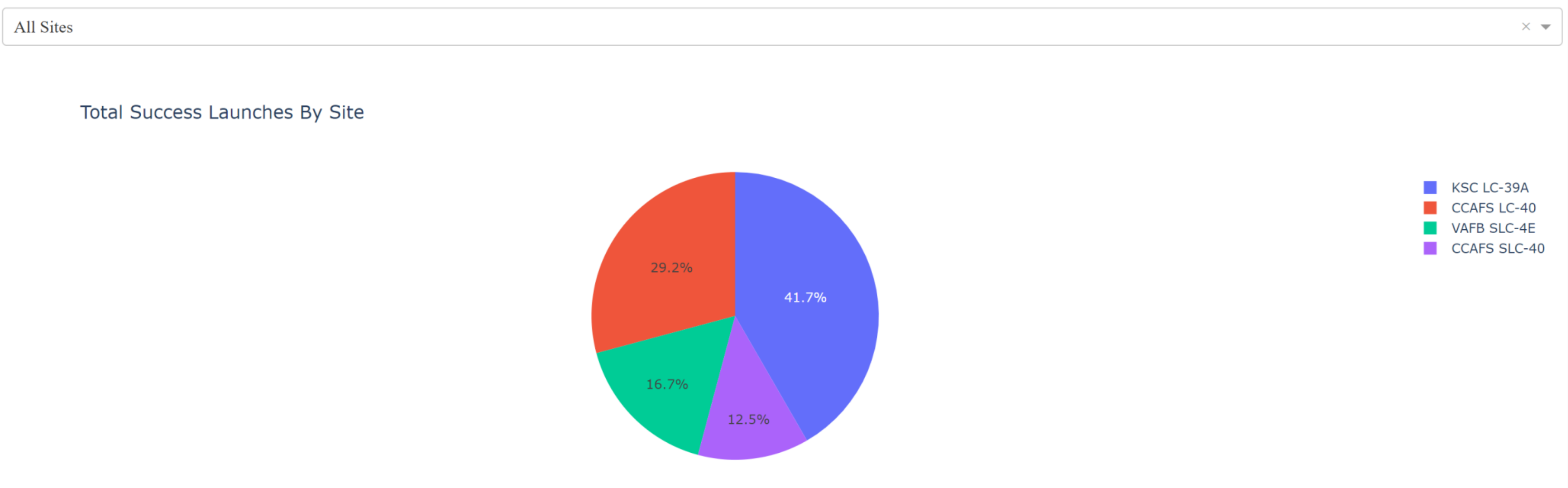


Section 4

Build a Dashboard with Plotly Dash

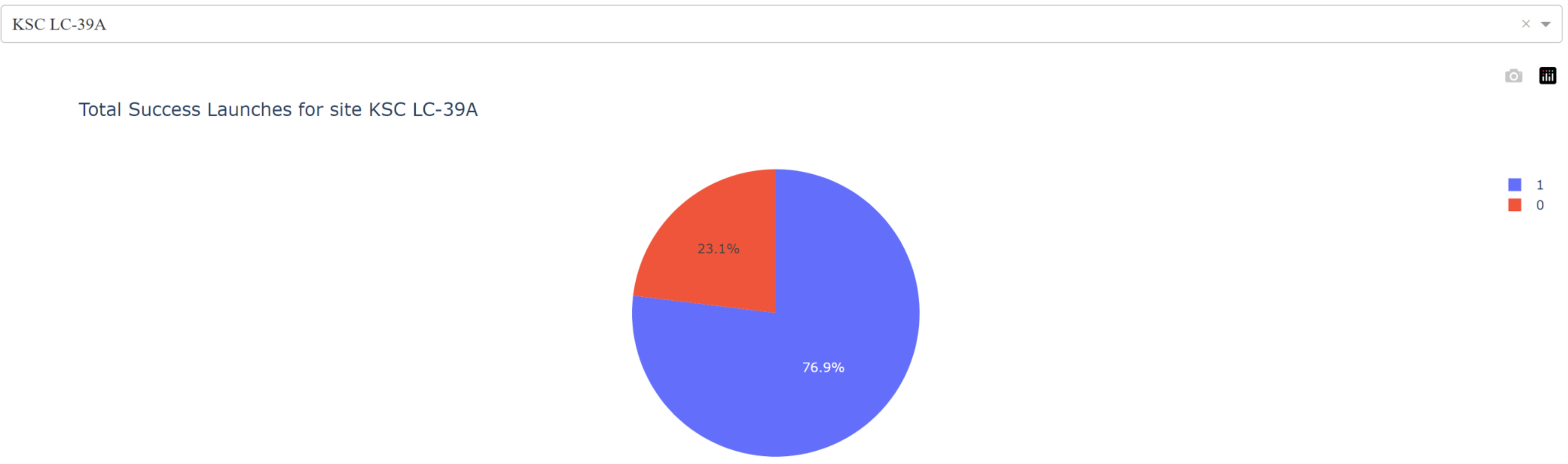
Pie Chart – Launch Success count for all sites

- KSC LC-39A has the highest success rate.



Launch detail for site with highest launch success

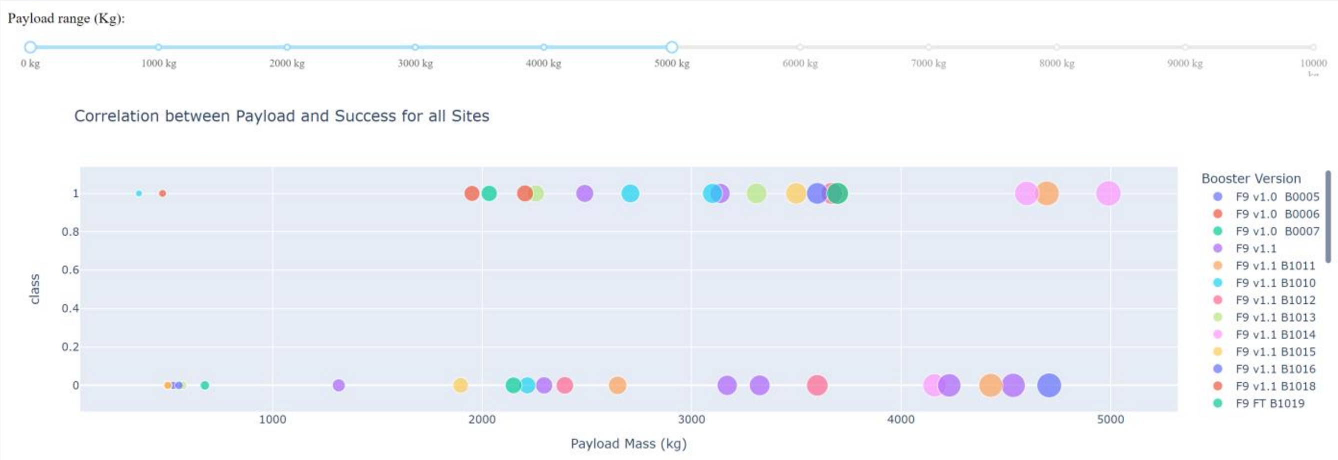
- KSC LC-39A has 76.9% of success rate and 23.1% of failure rate.

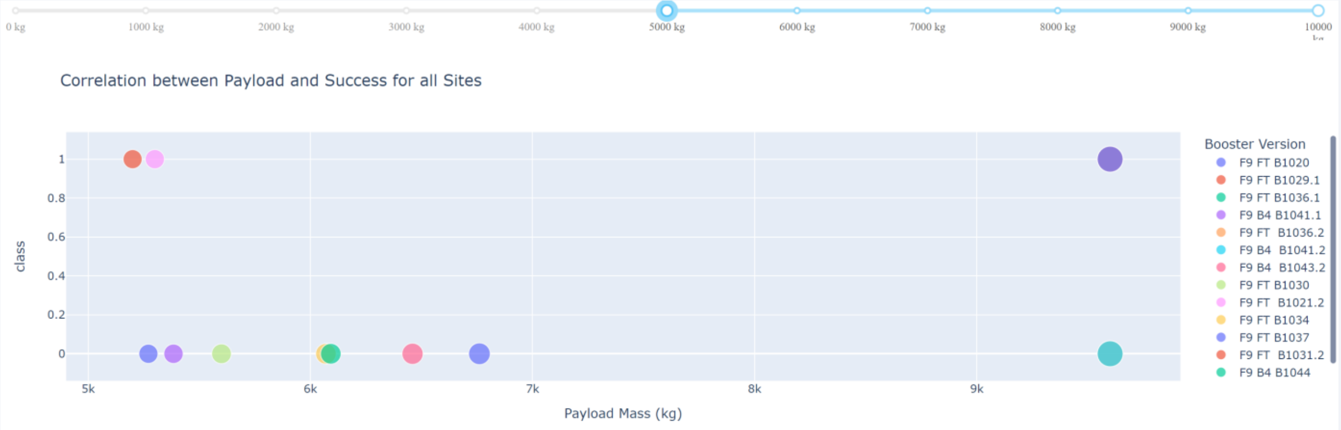


KSC LC-39A

Scatterplot of Payload vs Launch outcomes

- We can see launch outcomes for low payload (0-5000kg) and high payload (5000-10000kg).



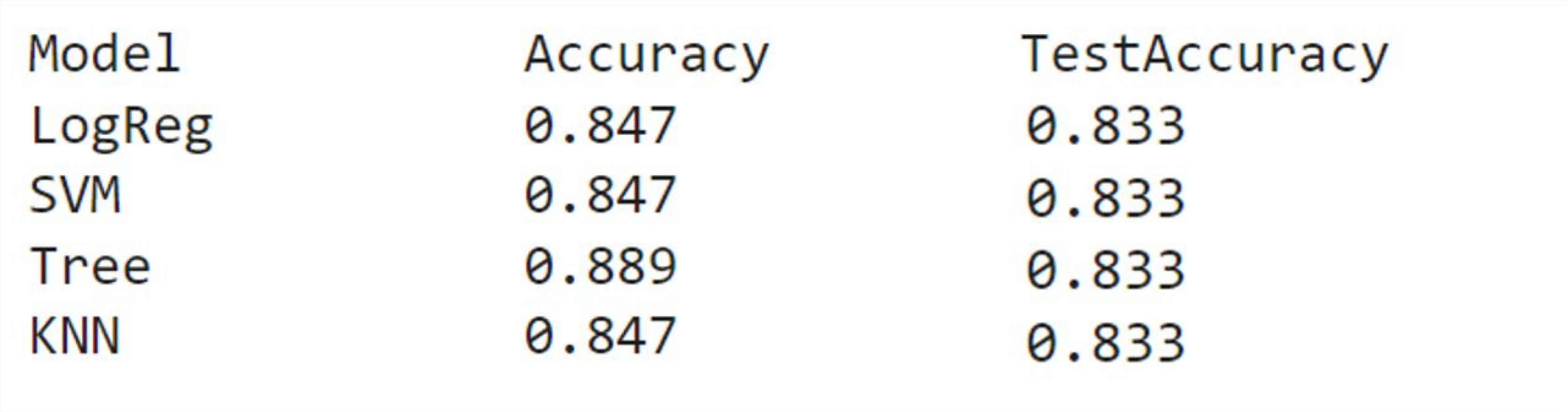


Section 5

Predictive Analysis (Classification)

Classification Accuracy

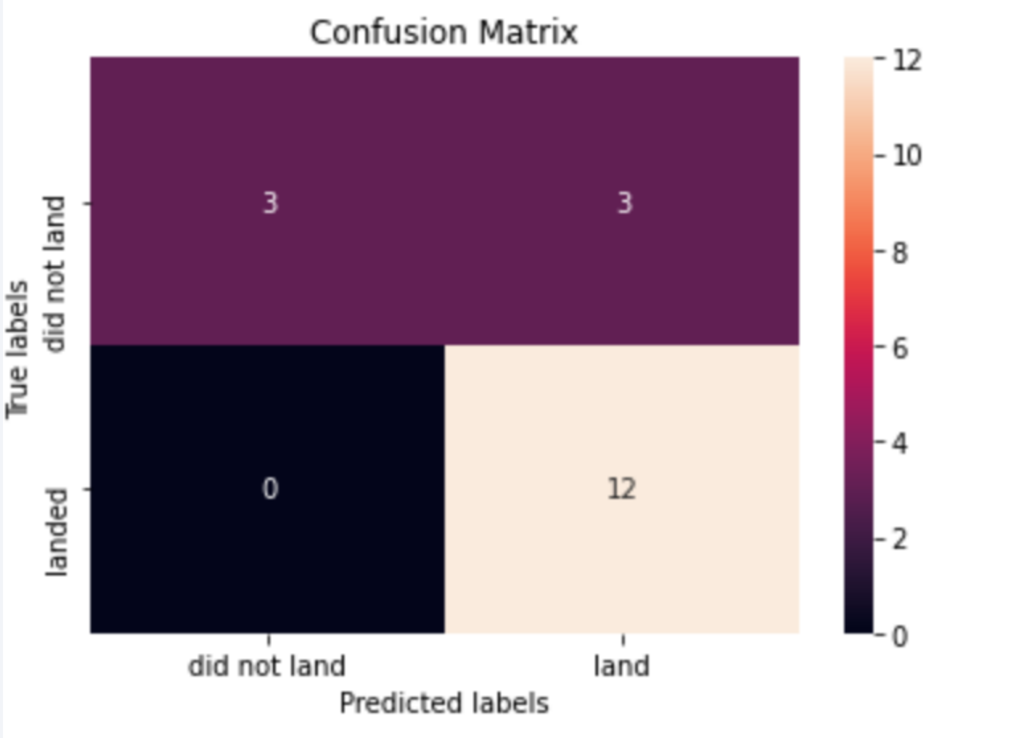
- From the result table, we can find that decision tree has the highest accuracy.



Model	Accuracy	TestAccuracy
LogReg	0.847	0.833
SVM	0.847	0.833
Tree	0.889	0.833
KNN	0.847	0.833

Confusion Matrix

- The confusion matrix of the best performing model (decision tree):



Conclusions

- We can conclude that:
- The larger the flight number at a launch site, the greater the success rate at a launch site.
- The success rate since 2013 kept increasing till 2020 in spite of a drop in 2018.
- Orbits ES-L1, SSO, HEO and GEO had the most success rate.
- KSC LC-39A had the most successful launches among all sites.
- The Decision tree classifier performed best in predicting launch outcomes.

Appendix

- For all the notebook codes, you can refer to this Github URL for detailed information:

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

Thank you!