IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Merna Tarek

4/10/2022

# 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 :

  Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers :

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

- The data was collected by :

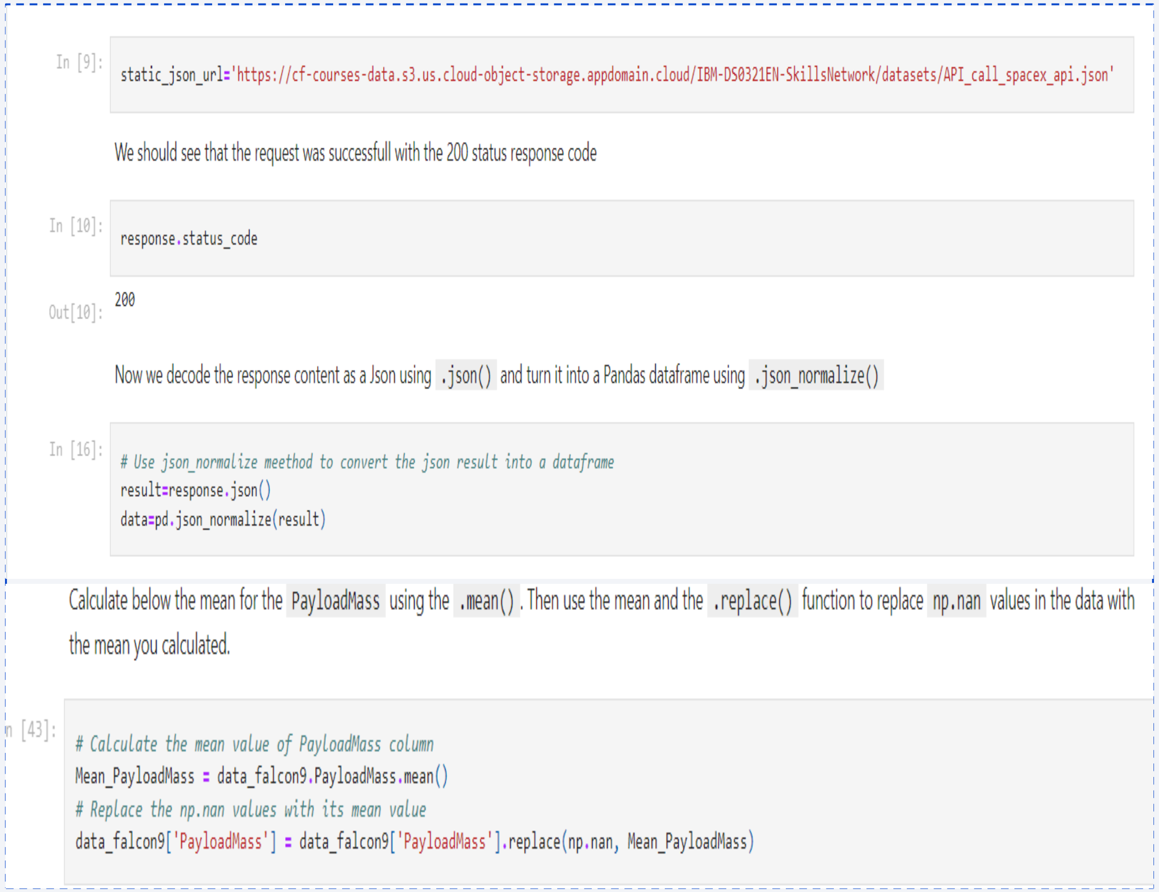
1. Using SpaceX API: Request (Space X  APIs) by using get method, use json_normalize method to convert the json result into a dataframe, then clean the data, finally check for missing values and fill them.

2. Using web scraping: perform an HTTP GET method to request the Falcon9 Launch HTML page, create a BeautifulSoup object from the response, then save the records as table and convert it into a dataframe.

# 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.

- GitHub URL: https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Collecting%20the%20data.ipynb



In [9]:
```
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 [10]:
```
response.status_code
```

Out[10]:
```
200
```

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

In [16]:
```
# Use json_normalize meethod to convert the json result into a dataframe
result=response.json()
data=pd.json_normalize(result)
```

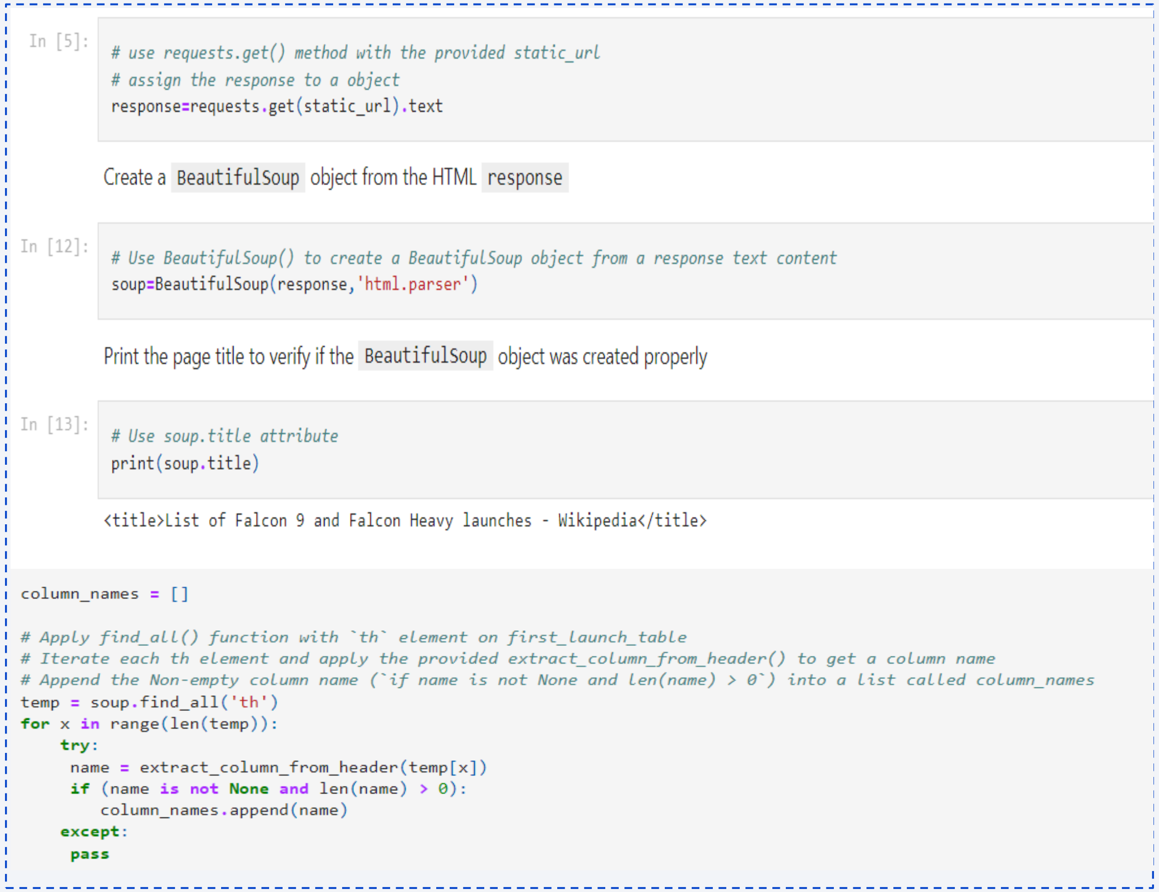
Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

n [43]:
```
# Calculate the mean value of PayloadMass column
Mean_PayloadMass = data_falcon9.PayloadMass.mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean_PayloadMass)
```

# Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup.
- We parsed the table and converted it into a pandas dataframe.
- GitHub URL:

[https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Collecting%20the%20data.ipynb](https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Collecting%20the%20data.ipynb)



In [5]:

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

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

In [12]:

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

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

In [13]:

```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

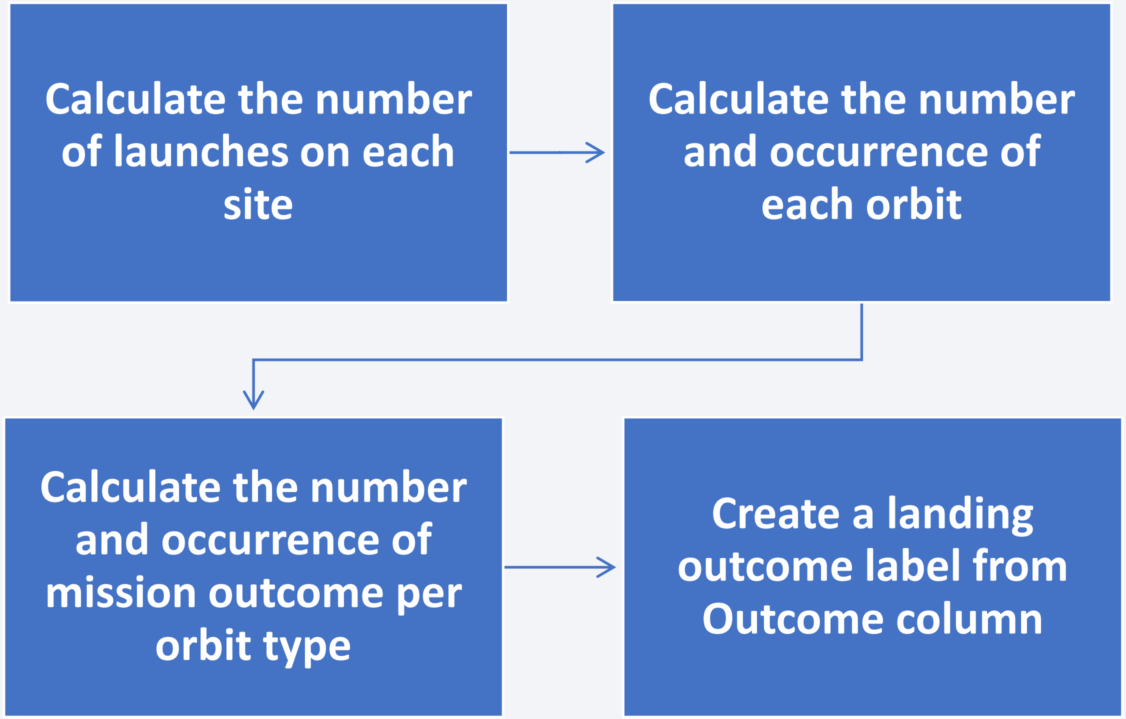
```
column_names = []

# 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
temp = soup.find_all('th')
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
```

# Data Wrangling

- GitHub URL:

[https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Data%20wrangling.ipynb](https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Data%20wrangling.ipynb)



Calculate the number of launches on each site

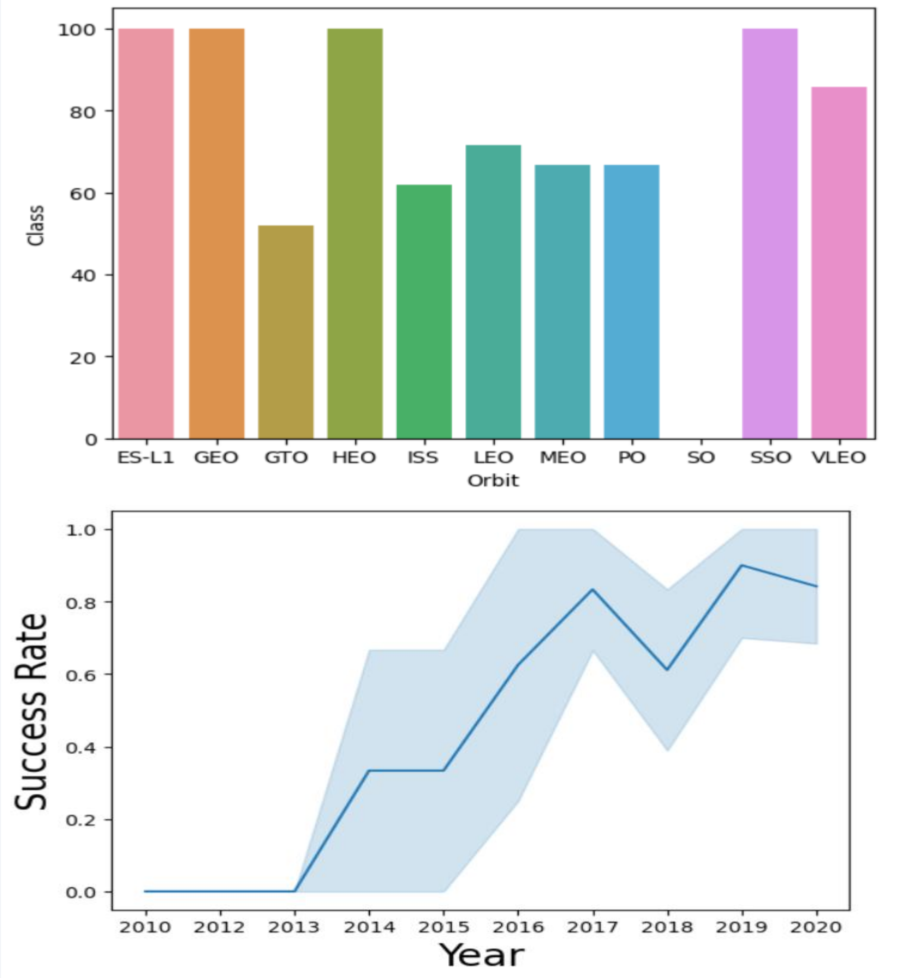
Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

# EDA with Data Visualization

- Scatter plot drown for: flightnumber and orbit type - payload and orbit type - launch sites and their payload mass - flightnumber and payload.
- Bar chart drown for: the success rate of each orbit.
- Line chart drown for: the launch success yearly trend.
- GitHub URL: [https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/eda-dataviz.ipynb](https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/eda-dataviz.ipynb)



# EDA with SQL

Performed SQL queries for:

- Displaying the names of the unique launch sites in the space mission.
- Displaying 5 records where launch sites begin with the string 'CCA‘.
- Displaying the total payload mass carried by boosters launched by NASA (CRS).
- Displaying average payload mass carried by booster version F9 v1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Ranking the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

- GitHub URL: [https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/EDA%20with%20SQL.ipynb](https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/EDA%20with%20SQL.ipynb)

# Build an Interactive Map with Folium

Markers of all Launch Sites:

- Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.
- Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.

Colored Markers of the launch outcomes for each Launch Site:

- Added colored Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates.

Distances between a Launch Site to its proximities:

- Added colored Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City.

- These objects are created in order to understand better the problem and the data.

- GitHub URL: https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

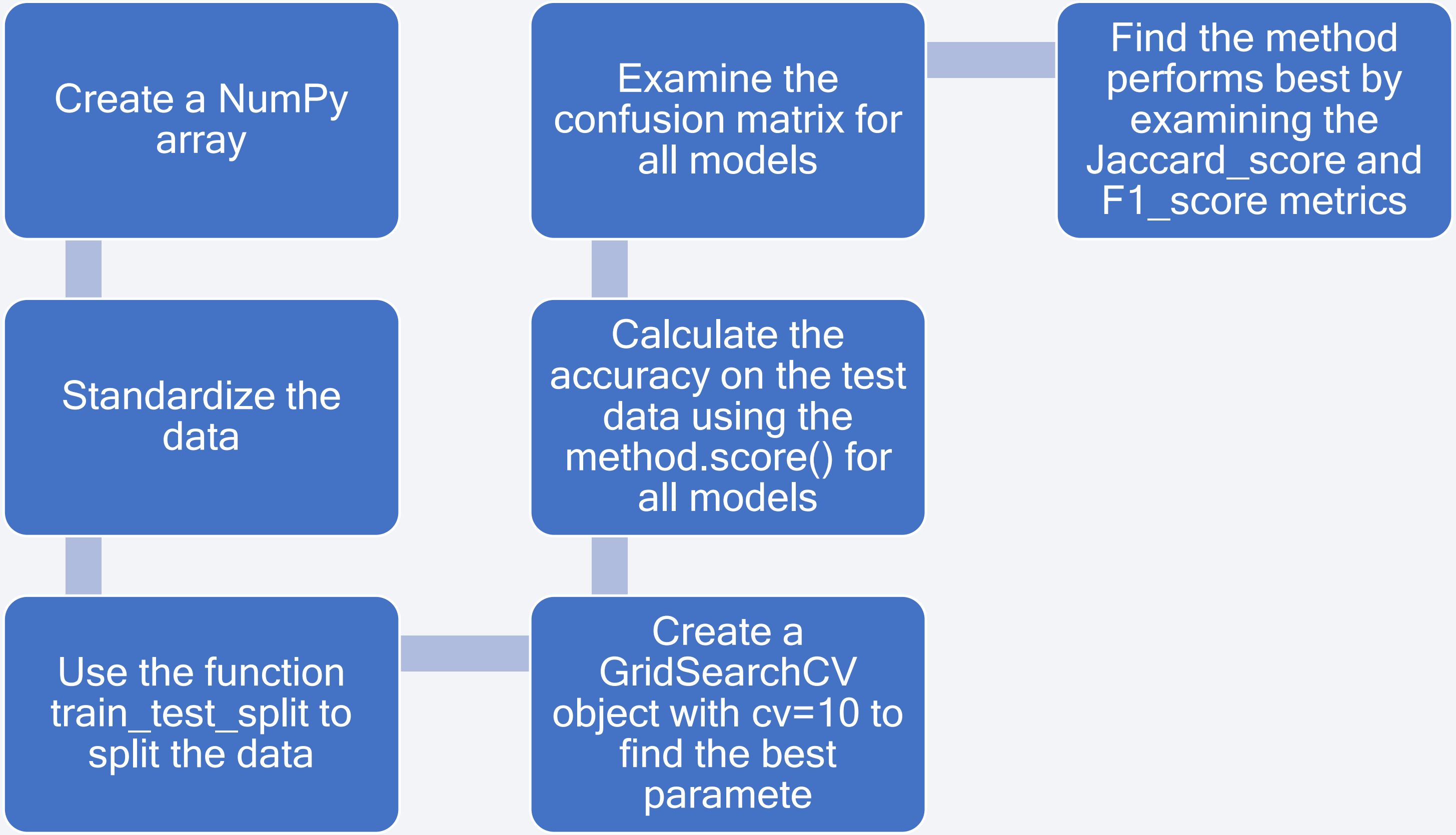
# Build a Dashboard with Plotly Dash

Dashboard has dropdown, pie chart, rangeslider and scatter plot components:

- Dropdown allows a user to choose the launch site or all launch sites.
- Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component.
- Rangeslider allows a user to select a payload mass in a fixed range.
- Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass.

- GitHub URL: https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

# Predictive Analysis (Classification)



Create a NumPy array

Standardize the data

Use the function train_test_split to split the data

Create a GridSearchCV object with cv=10 to find the best paramete

Calculate the accuracy on the test data using the method.score() for all models

Examine the confusion matrix for all models

Find the method performs best by examining the Jaccard_score and F1_score metrics

- GitHub URL: https://github.com/Merna-Tarek/Applied-Data-Science-Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

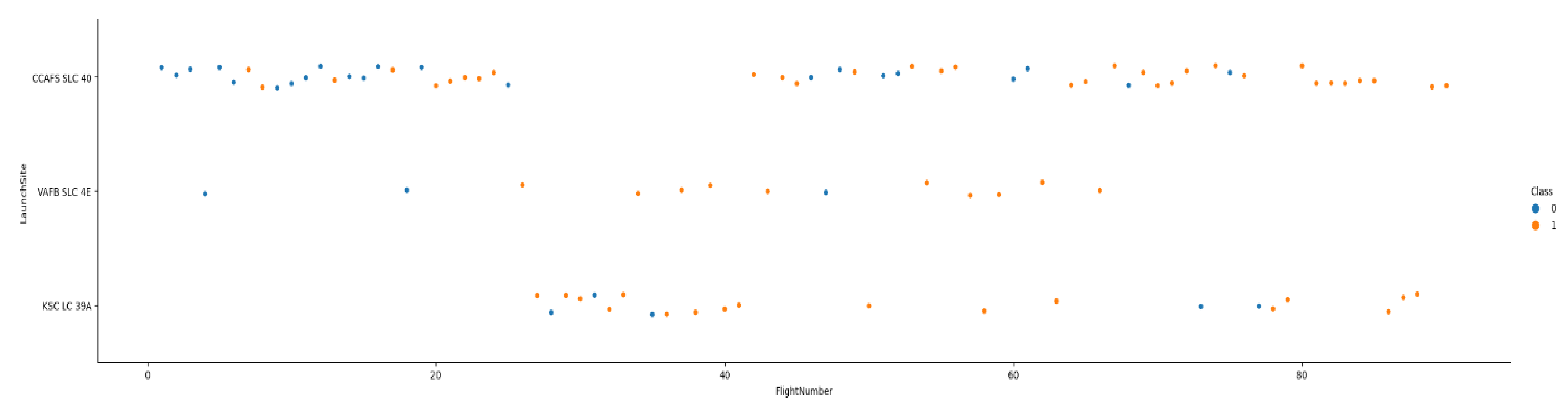
# Results

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

Section 2

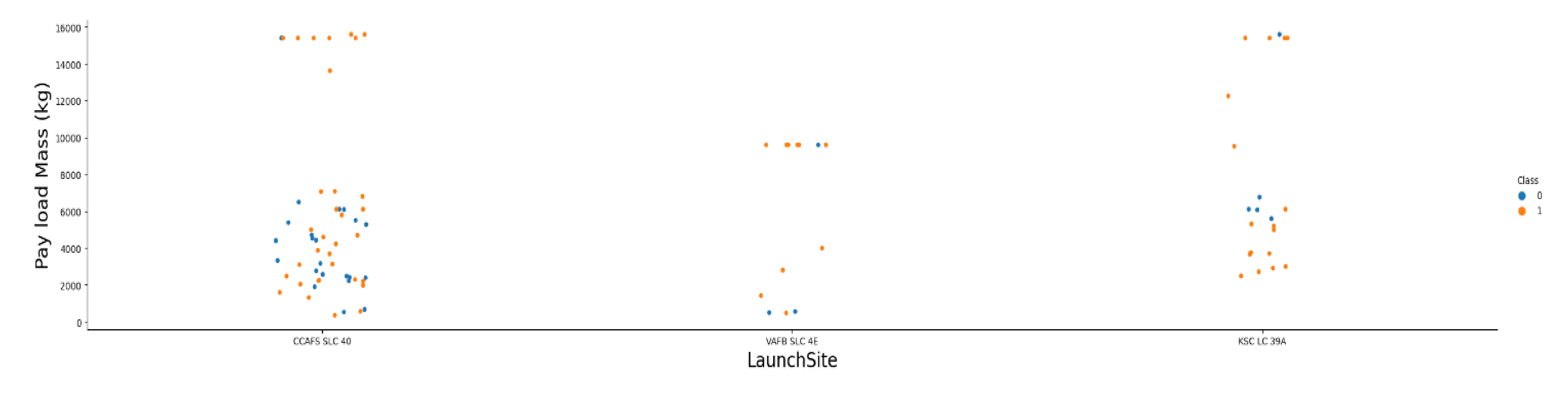
# Insights drawn from EDA

# Flight Number vs. Launch Site



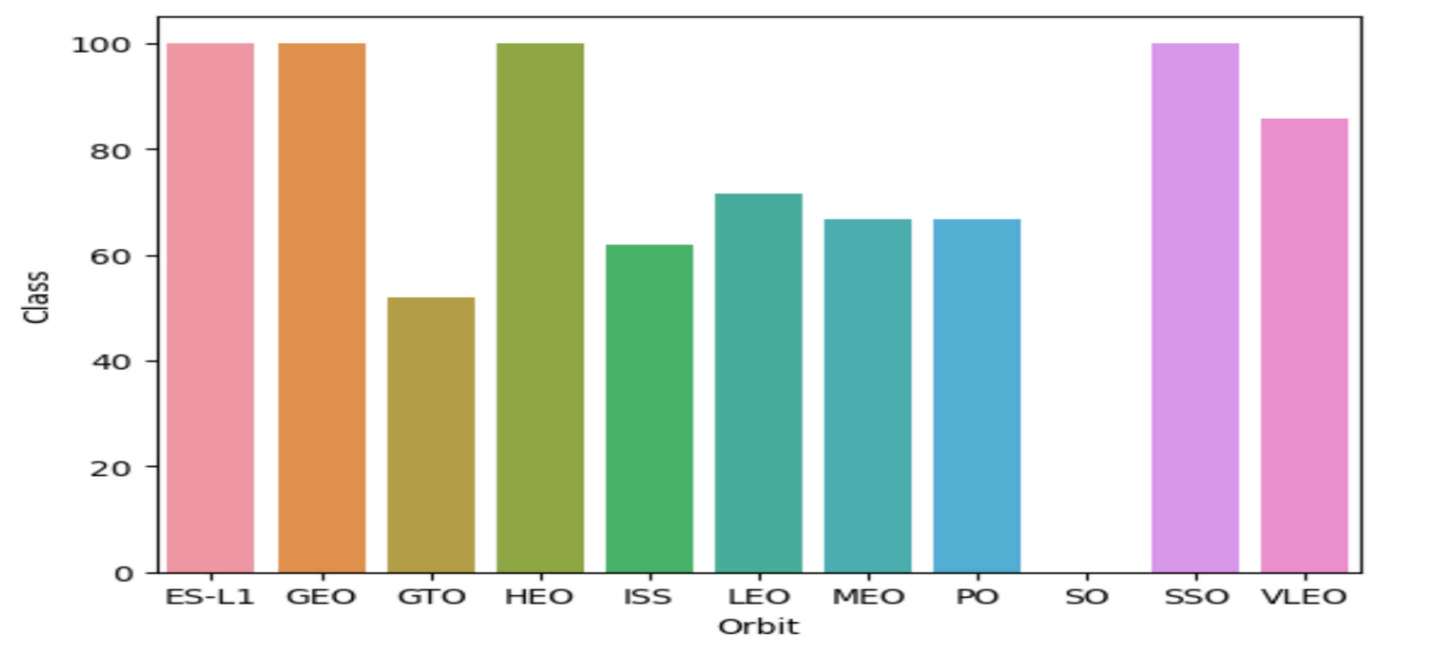
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

# Payload vs. Launch Site



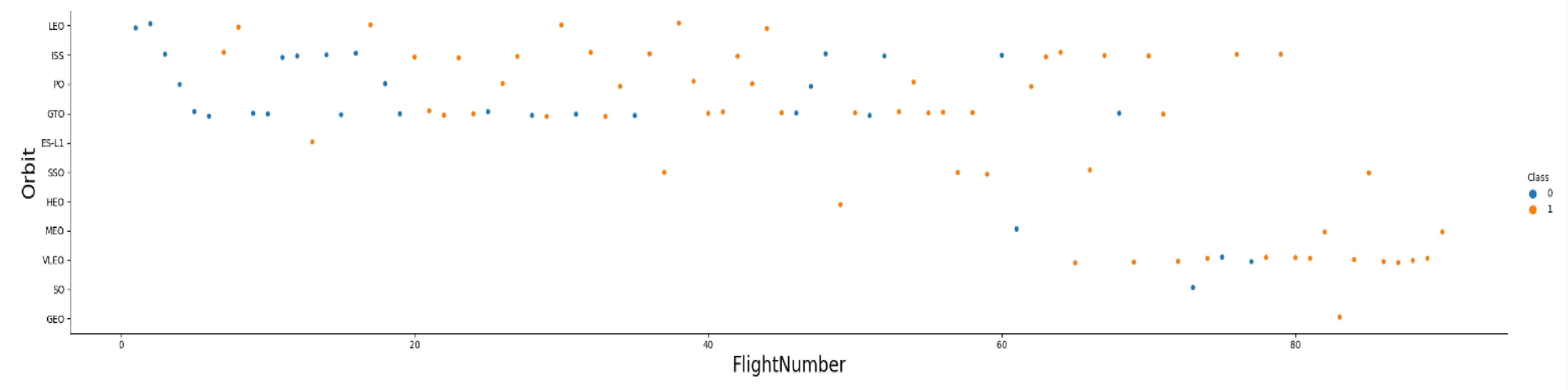
- From the plot, for every launch site the higher the payload mass, the higher the success rate.

# Success Rate vs. Orbit Type



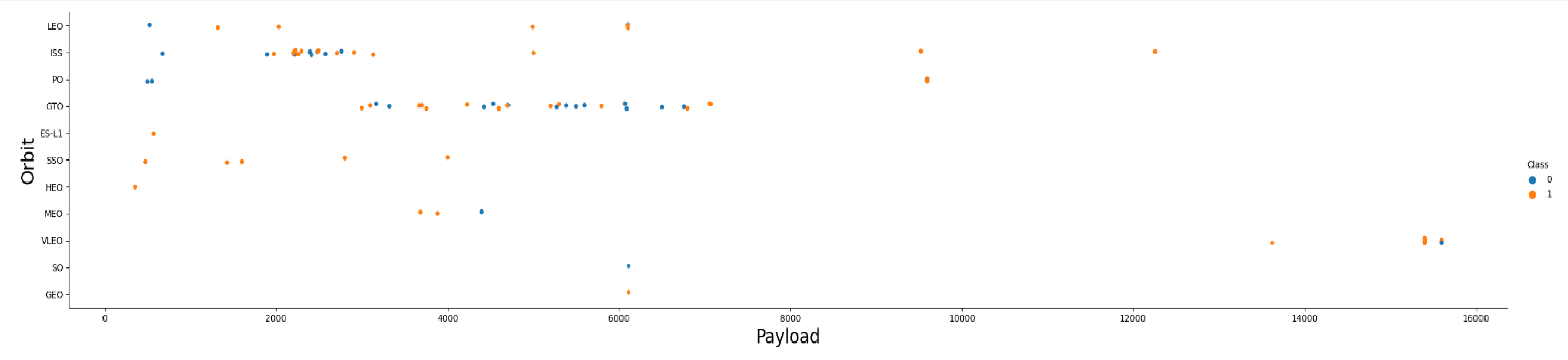
- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- Orbits with 0% success rate: SO

# Flight Number vs. Orbit Type



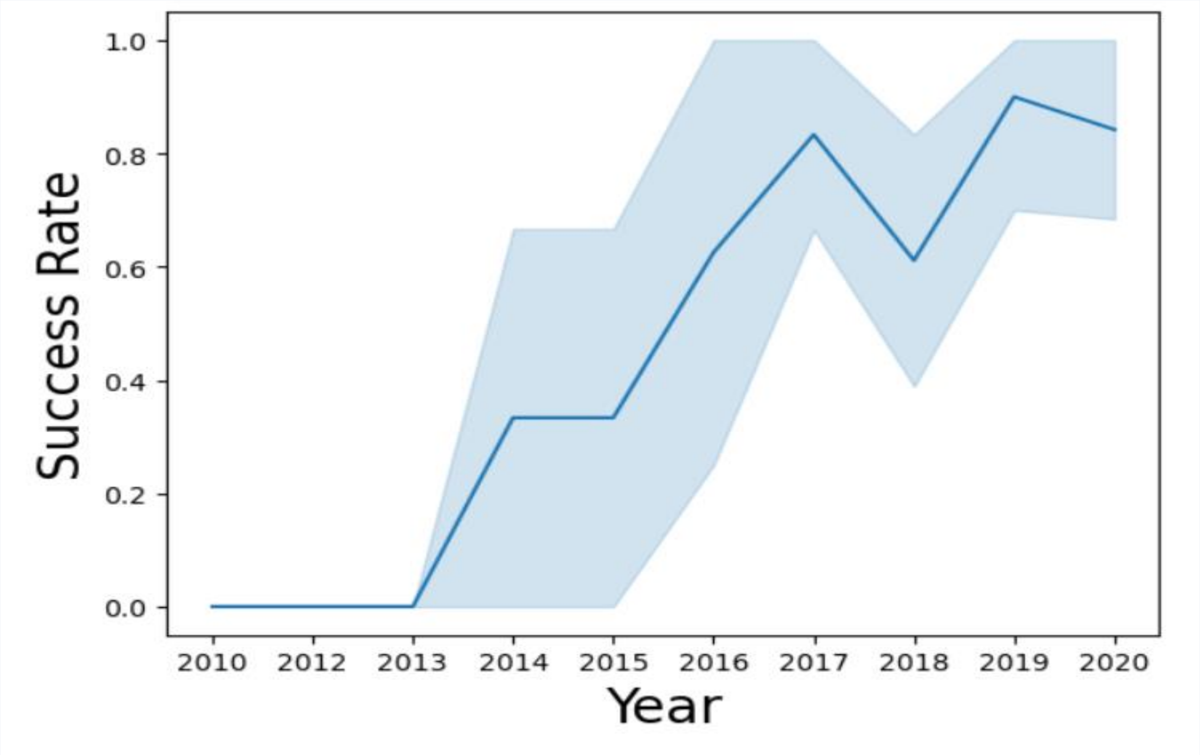
- From the plot, in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

# Payload vs. Orbit Type



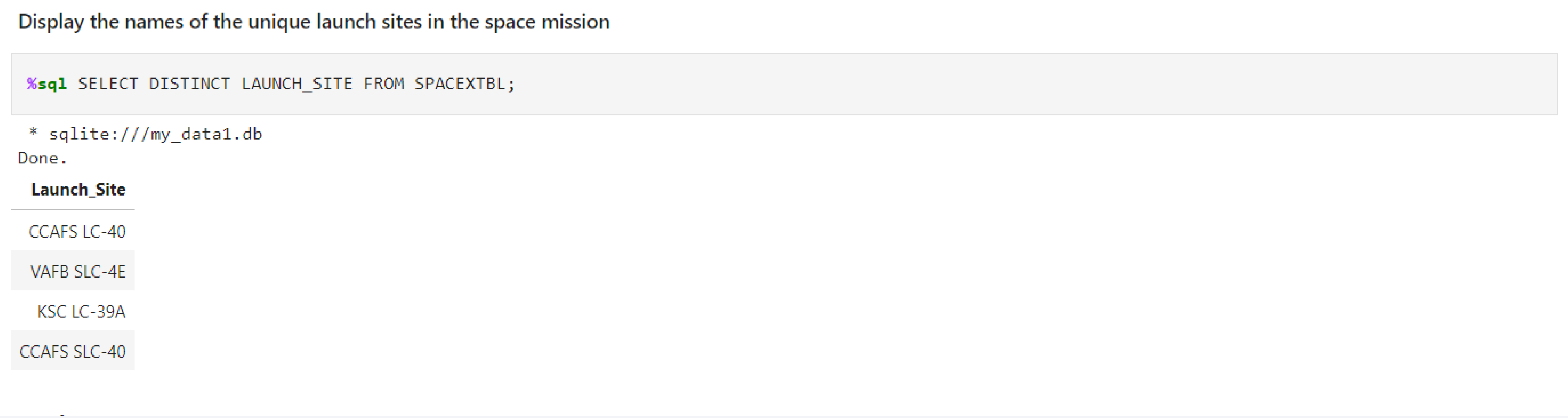
- From the plot, the weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that decreasing the payload weight for a GTO orbit improves the success of a launch.

# Launch Success Yearly Trend



- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.

# All Launch Site Names



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

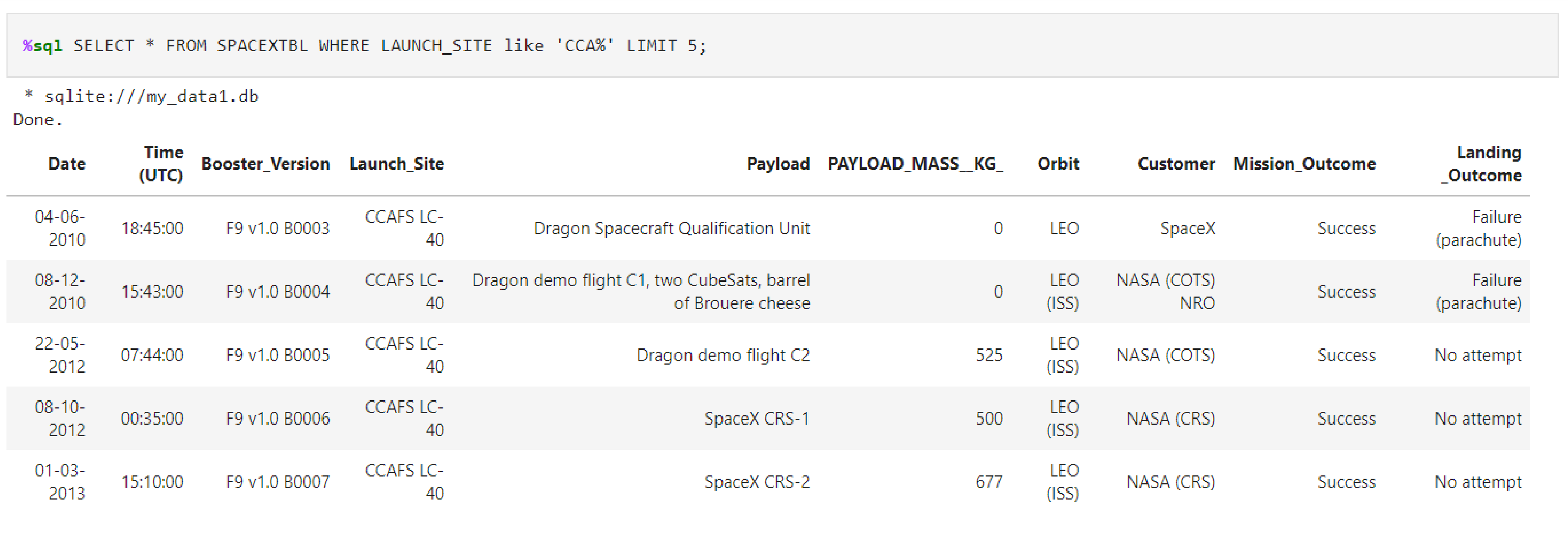
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

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

- The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

# Launch Site Names Begin with 'CCA'



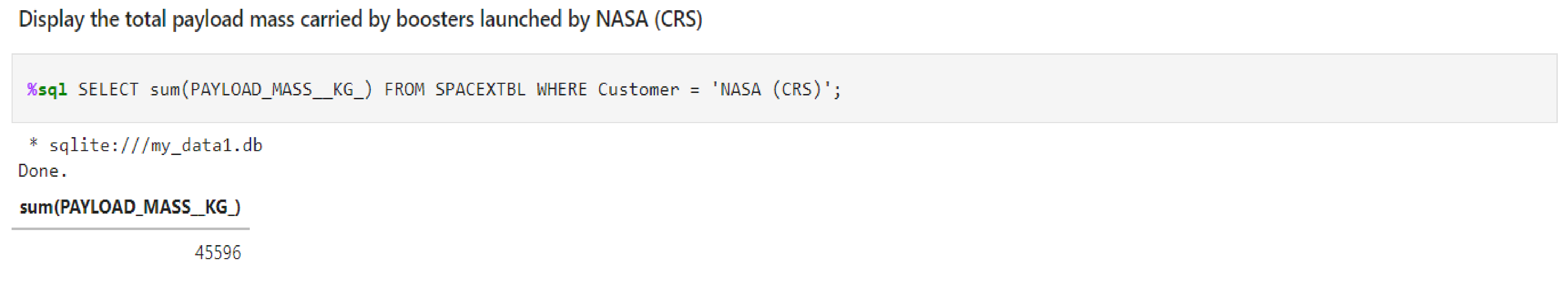
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE like 'CCA%' LIMIT 5;
```

```
 * 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 |

- By using The where clause followed by like clause filters launch sites that contain the substring CCA, LIMIT 5 shows 5 records from filtering.

# Total Payload Mass



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

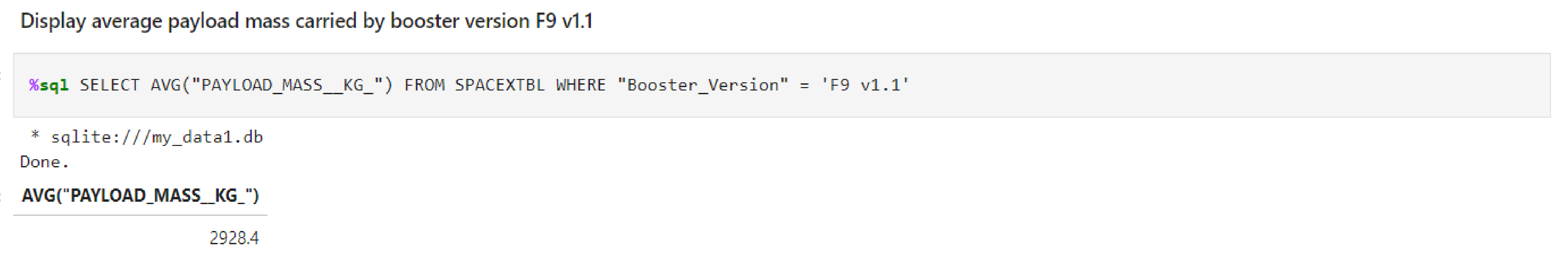
```
%sql SELECT sum(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

| sum(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

- By using the sum function returns the sum of all payload masses where the customer is NASA (CRS).

# Average Payload Mass by F9 v1.1



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

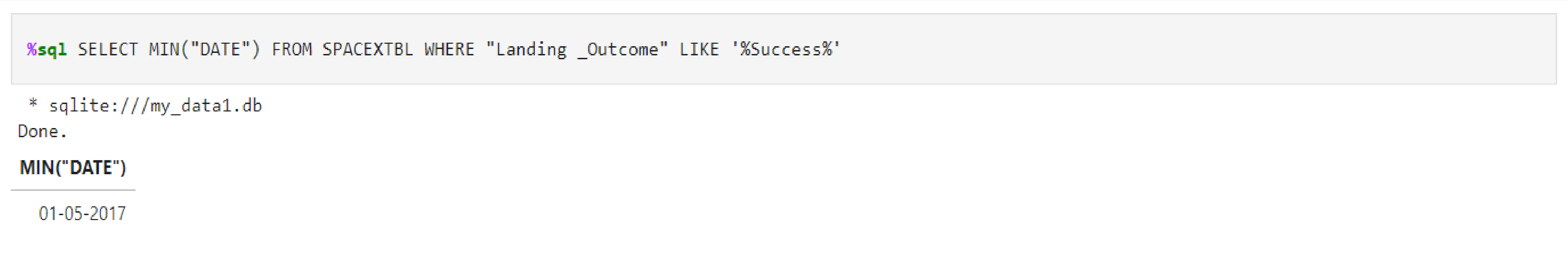
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" = 'F9 v1.1'
```

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

| AVG("PAYLOAD_MASS__KG_") |
|---|
| 2928.4 |

- By using the AVG function, we calculated the average payload mass carried by booster version F9 v1.1.

# First Successful Ground Landing Date



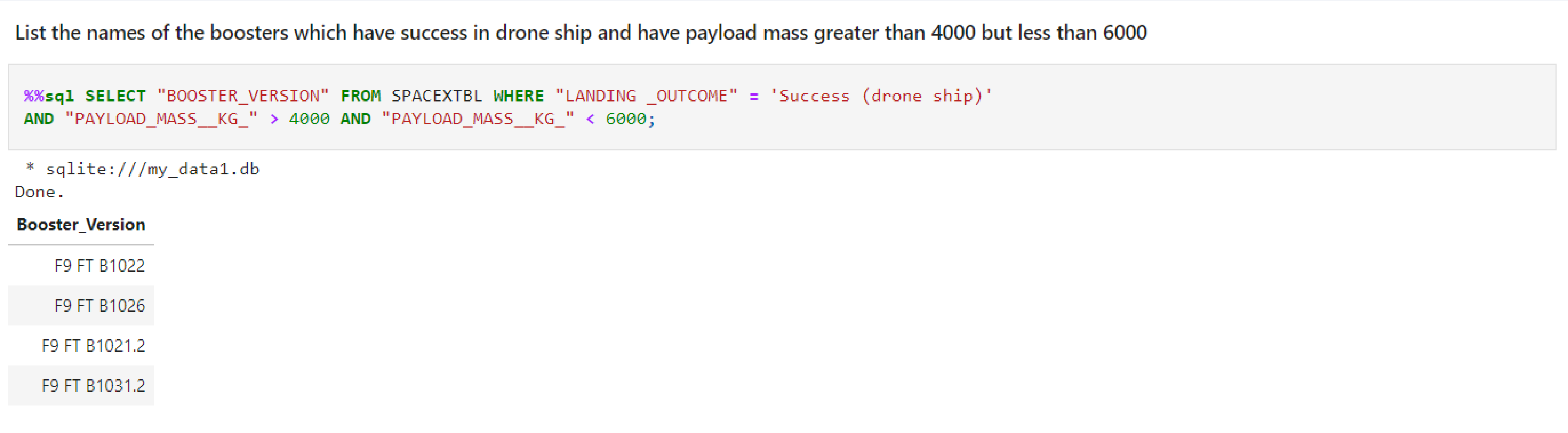
```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

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

| MIN("DATE") |
|---|
| 01-05-2017 |

- By using the MIN function, we selected the record with the oldest date.

# Successful Drone Ship Landing with Payload between 4000 and 6000



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

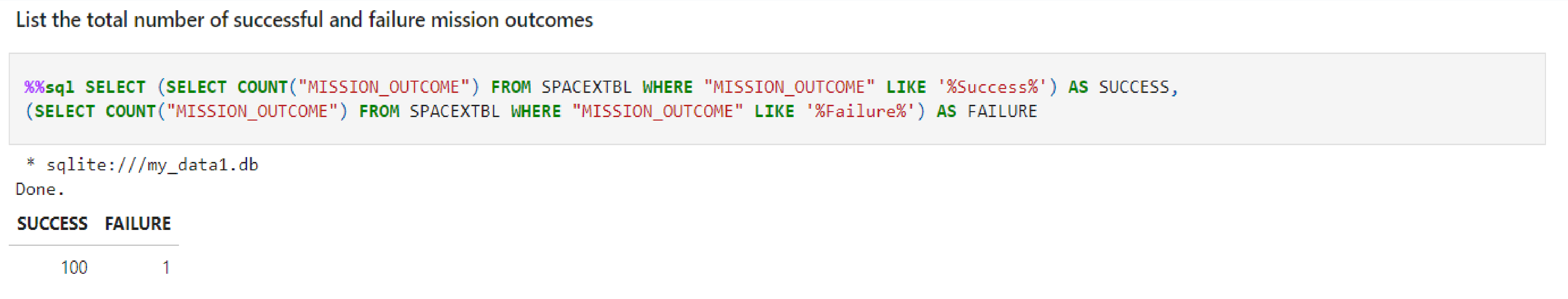
```
%%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)'
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

| Booster_Version |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

- This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

# Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

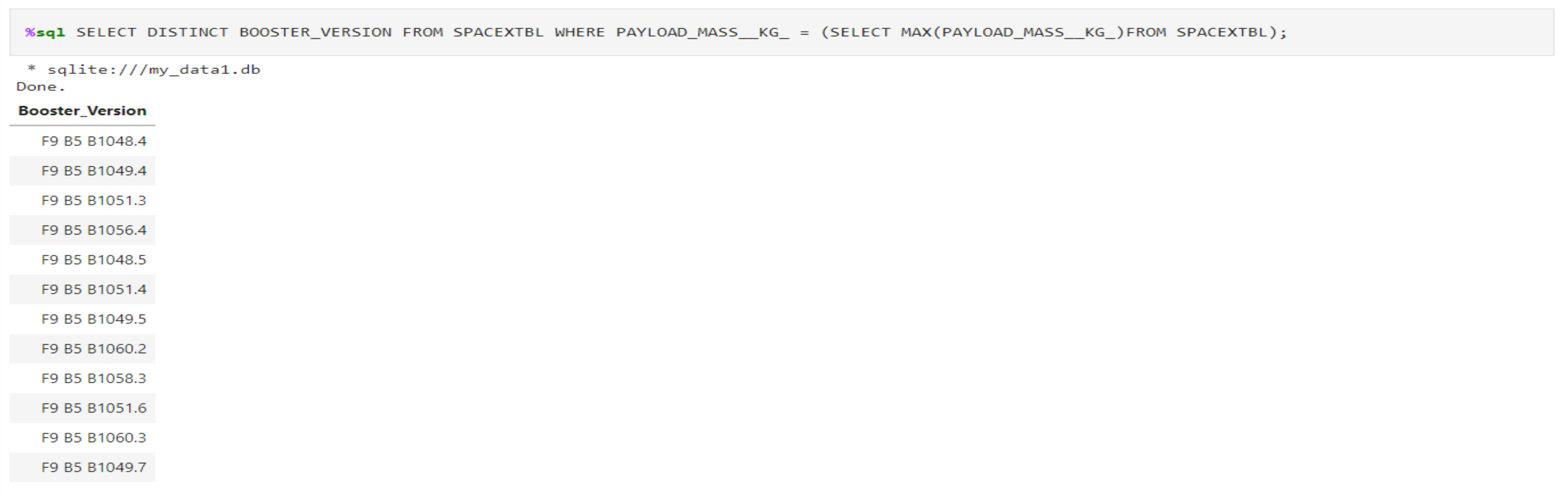
```
%%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS,
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

| SUCCESS | FAILURE |
|---|---|
| 100 | 1 |

- With the first SELECT, we show the subqueries that return results. the first subquery counts the successful mission, the second subquery counts the unsuccessful mission, the WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

# Boosters Carried Maximum Payload



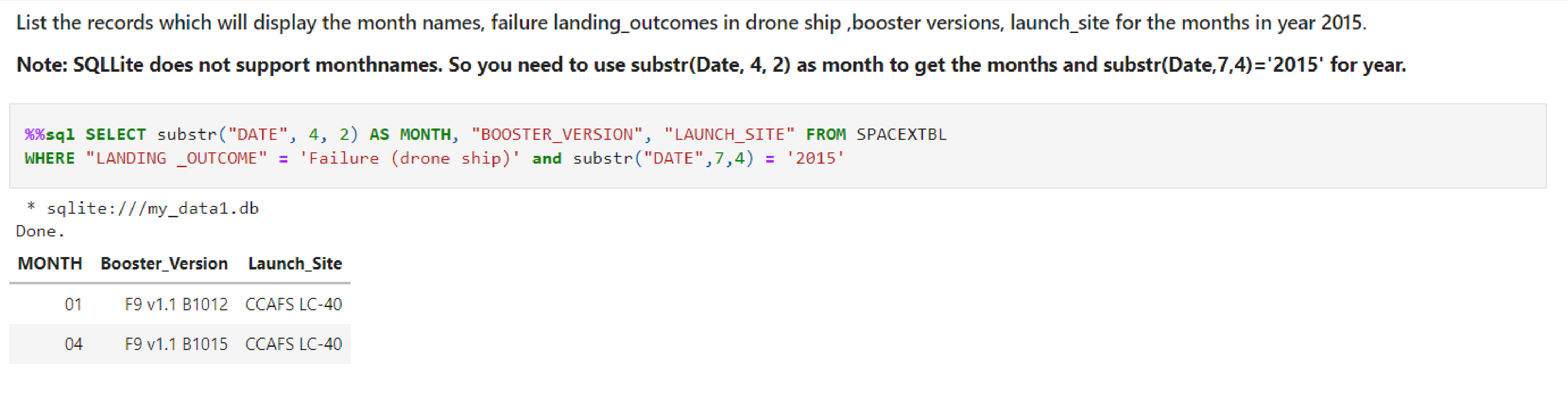
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)FROM SPACEXTBL);
```

```
 * 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 |

- We used a subquery to filter data by returning only the heaviest payload mass with MAX function, the main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

# 2015 Launch Records



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.

**Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.**

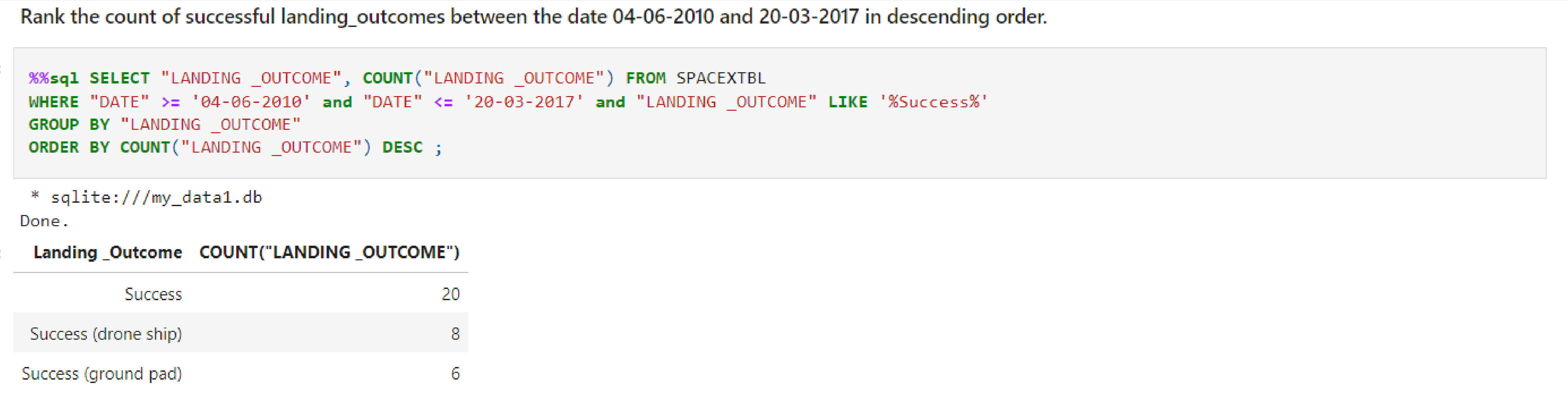
```
%%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

| MONTH | Booster_Version | Launch_Site |
|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 |

- This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year, Substr(DATE, 4, 2) shows month.

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



Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'
GROUP BY "LANDING _OUTCOME"
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

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

| Landing _Outcome | COUNT("LANDING _OUTCOME") |
|---:|---:|
| Success | 20 |
| Success (drone ship) | 8 |
| Success (ground pad) | 6 |

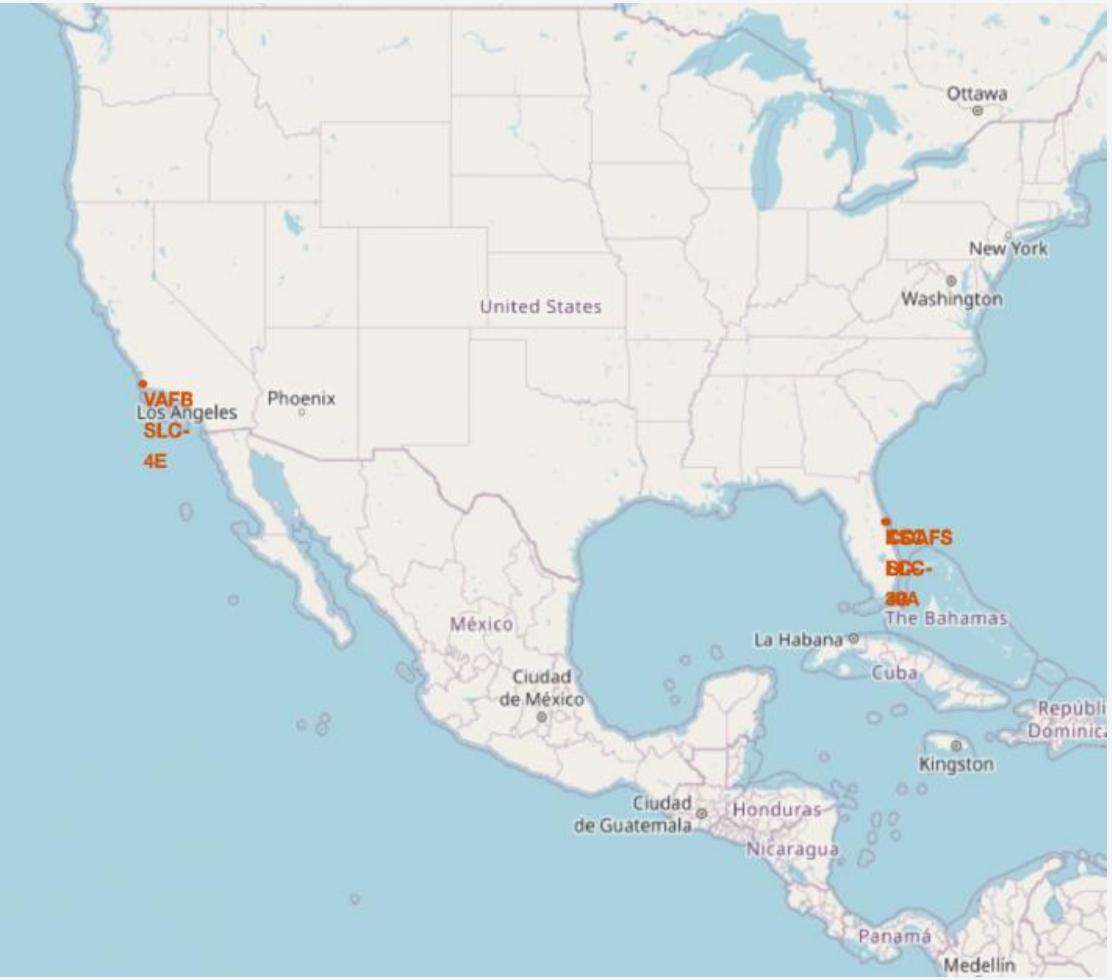
- This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Section 3

# Launch Sites Proximities Analysis

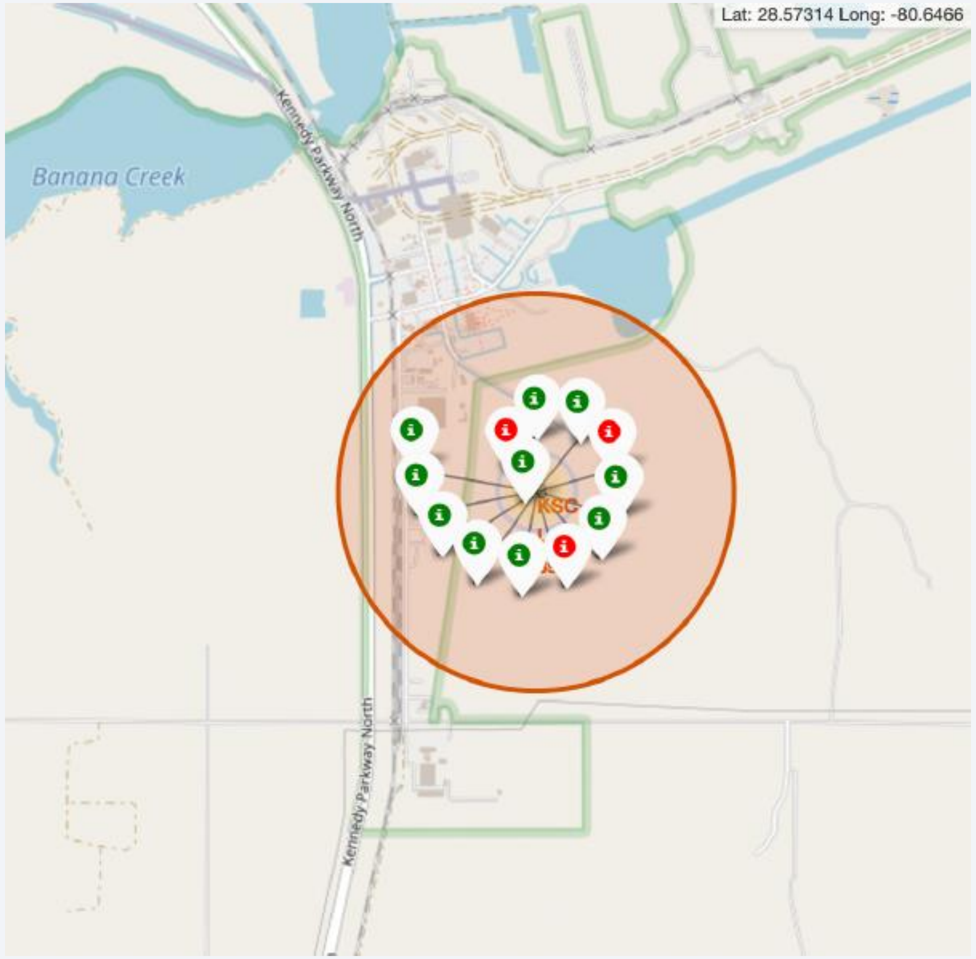
# All launch sites global on Folium map

- We see that SpaceX launch sites are located on the coast of the United States like Florida and California.



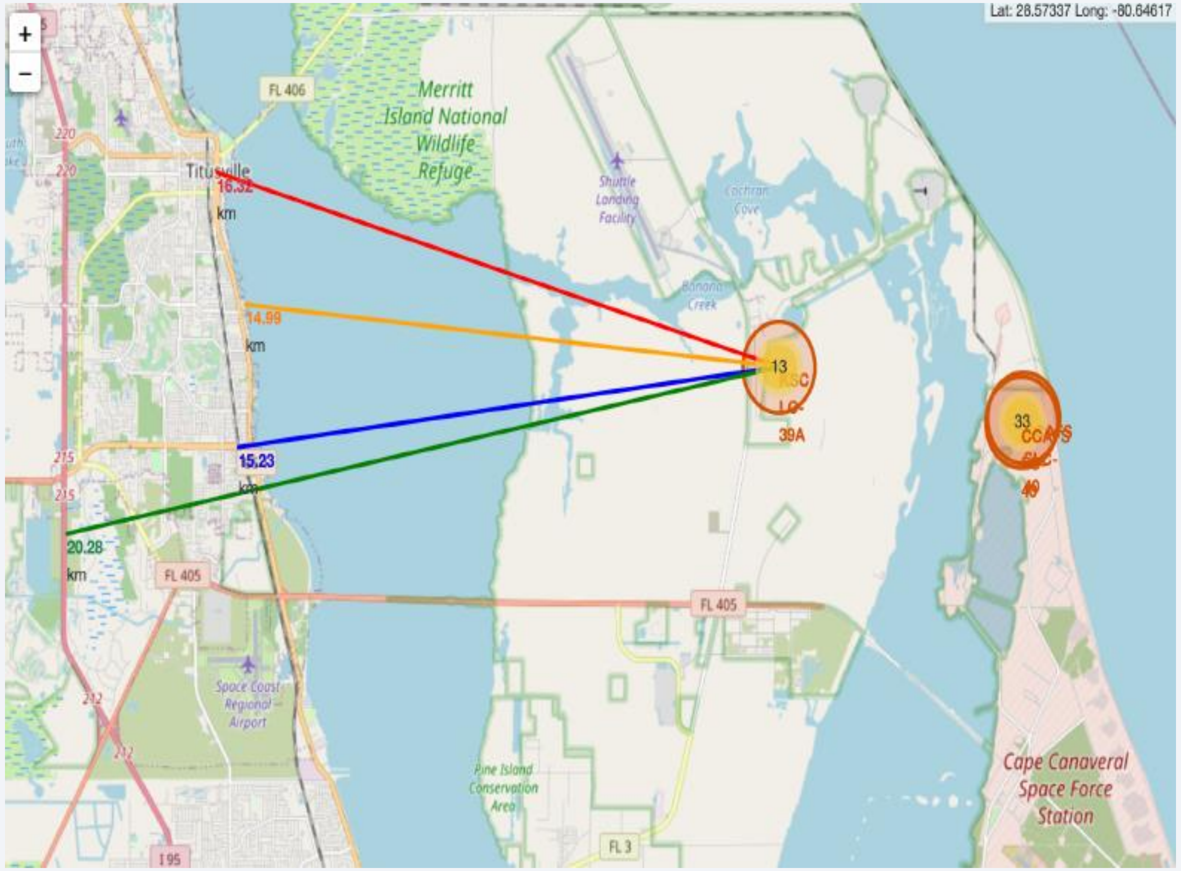
# Color-labeled launch records on Folium map

- Green marker represents successful launches.
- Red marker represents unsuccessful launches.

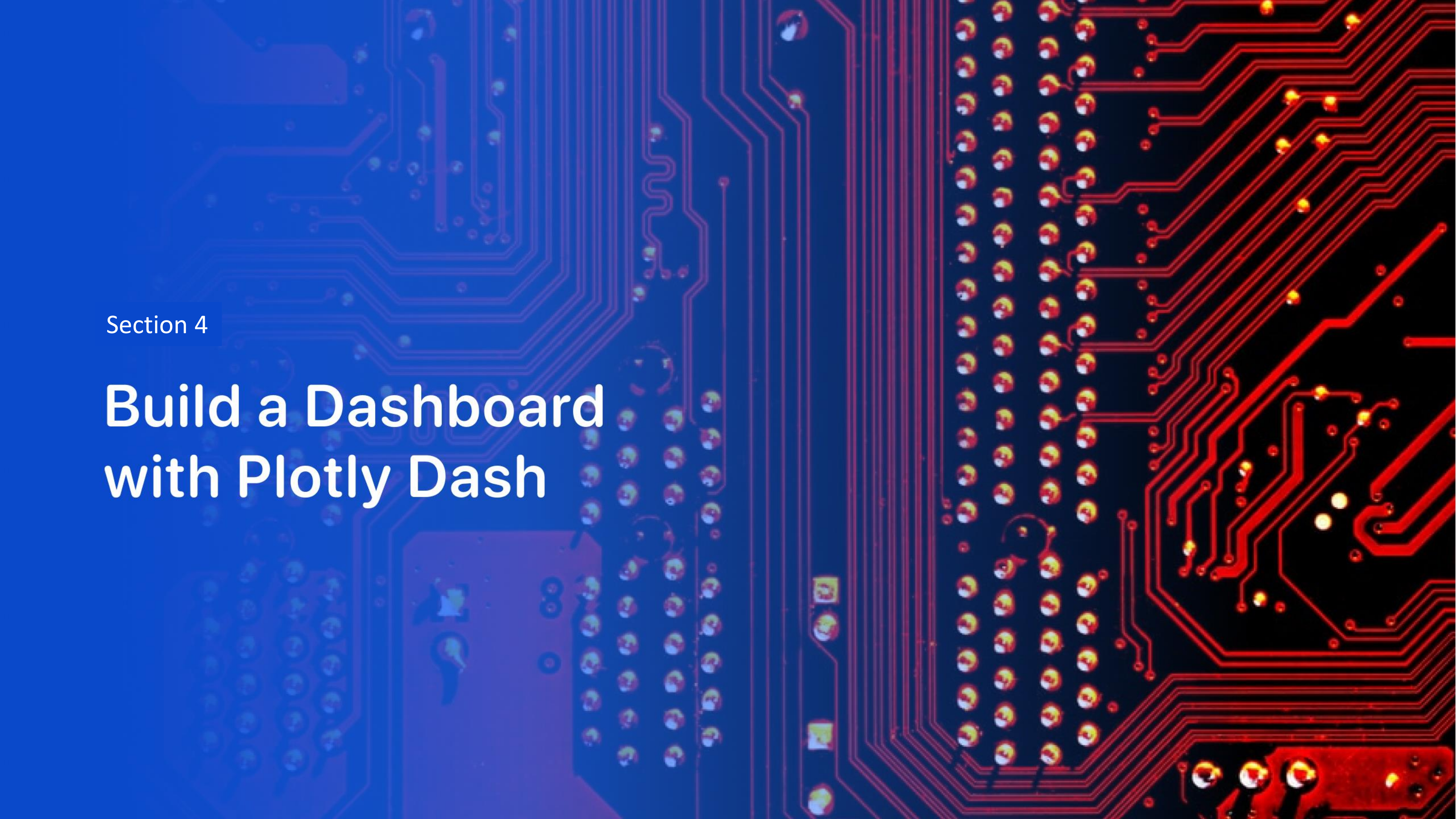


# Distance from the launch site KSC LC-39A to its proximities on Folium map

- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40in close proximity to highways ? Yes
- Is CCAFS SLC-40in close proximity to coastline ? Yes
- Do CCAFS SLC-40keeps certain distance away from cities ? No



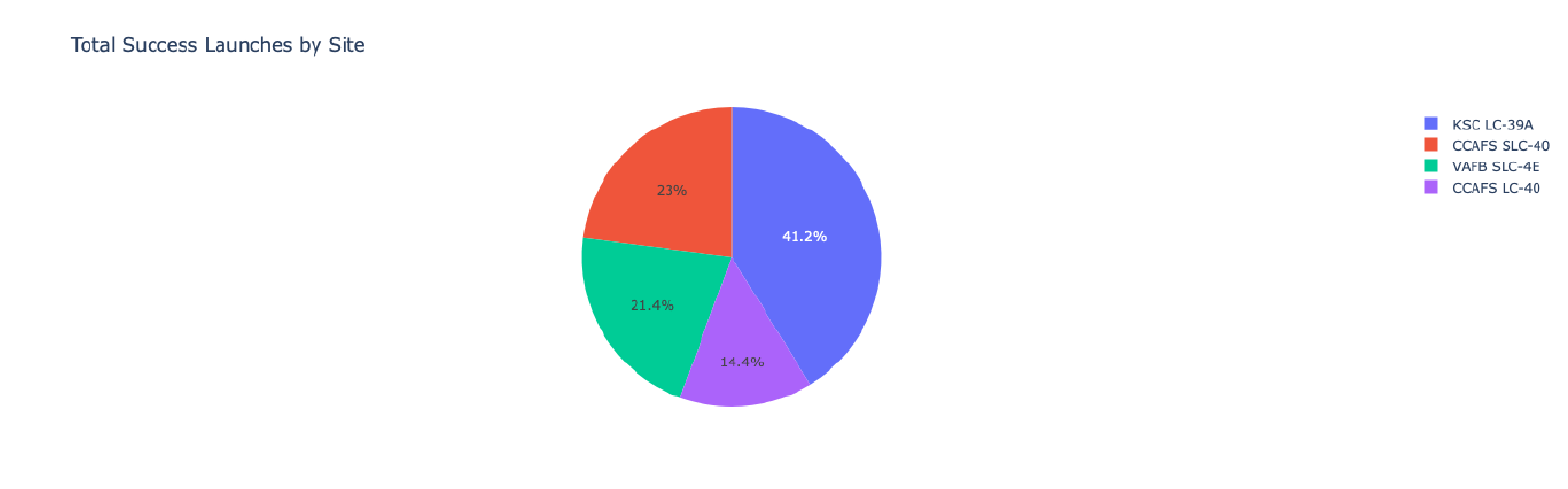
+
−
Lat: 28.57337 Long: -80.64617
FL 406
Merritt Island National Wildlife Refuge
Titusville
16.32 km
14.99 km
15.23 km
20.28 km
Shuttle Landing Facility
Cochran Cove
Banana Creek
13
KSC LC-39A
33
CCAFS SLC-40
220
215
212
FL 405
FL 405
Space Coast Regional Airport
Pine Island Conservation Area
FL 3
I 95
Cape Canaveral Space Force Station



Section 4

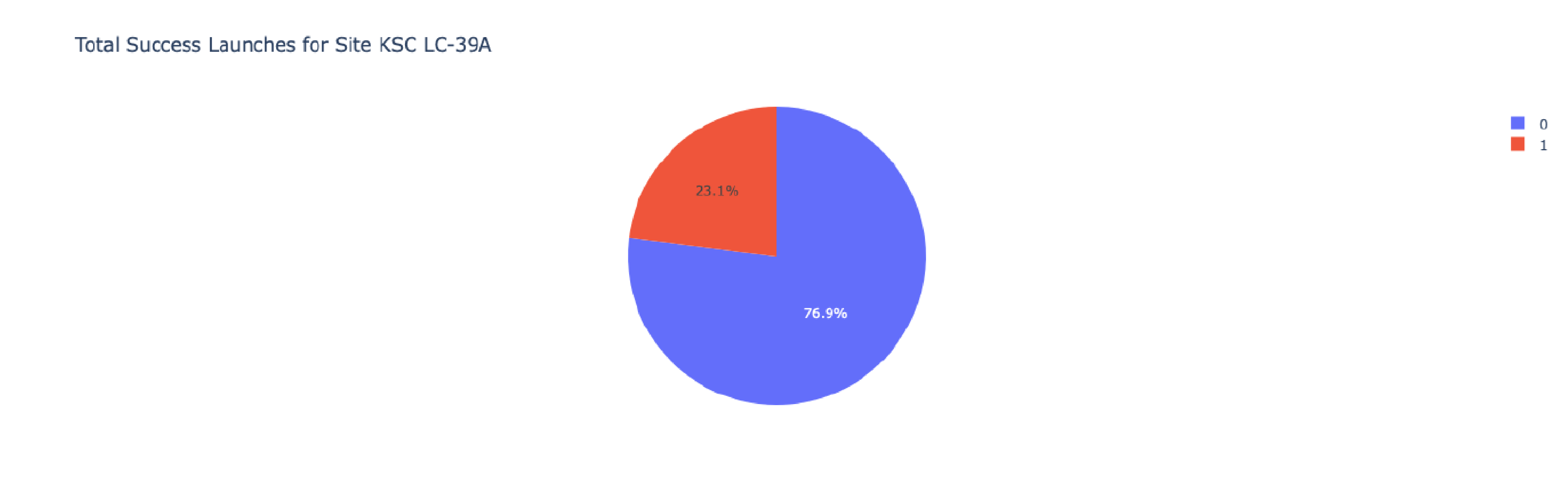
# Build a Dashboard with Plotly Dash

# Launch success count for all sites



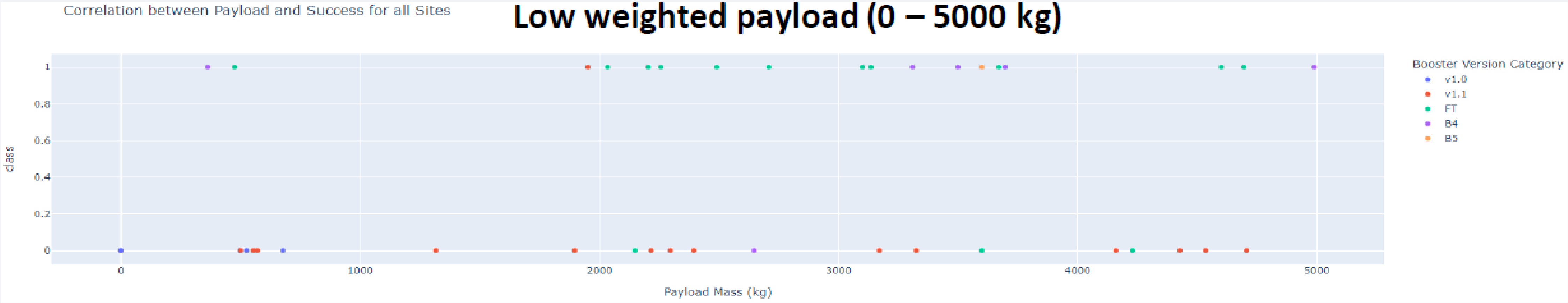
- In This chart, We see that KSC LC-39A has the best success rate of launches.

# Total success launches for Site KSC LC-39A

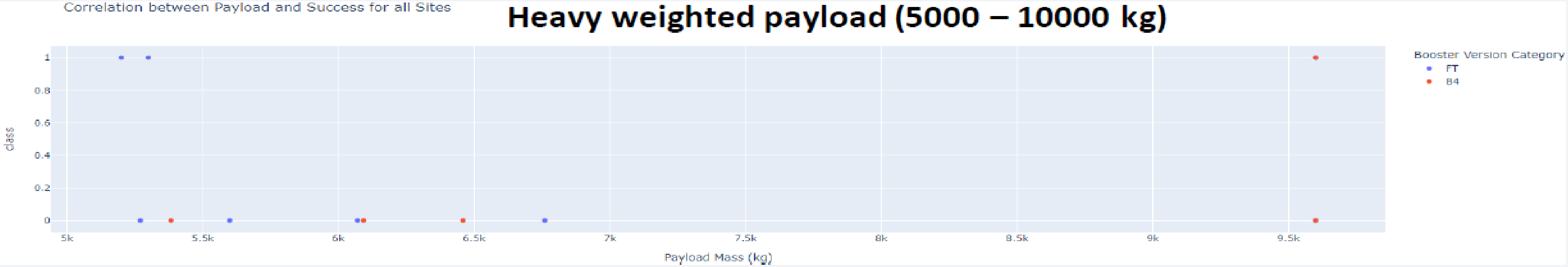


- In this chart, We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

# Payload mass vs Outcome for all sites with different payload mass selected



## Low weighted payload (0 – 5000 kg)



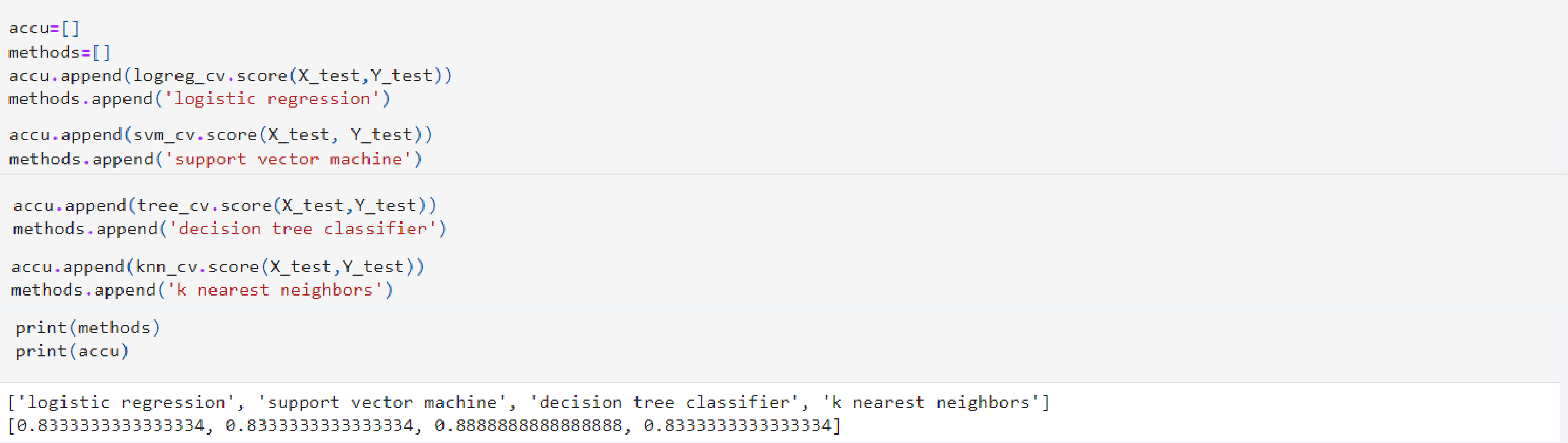
## Heavy weighted payload (5000 – 10000 kg)

- The charts show that low weighted payloads have a better success rate than the heavy weighted payloads.

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy



```
accu=[]
methods=[]
accu.append(logreg_cv.score(X_test,Y_test))
methods.append('logistic regression')
```

```
accu.append(svm_cv.score(X_test, Y_test))
methods.append('support vector machine')
```

```
accu.append(tree_cv.score(X_test,Y_test))
methods.append('decision tree classifier')
```

```
accu.append(knn_cv.score(X_test,Y_test))
methods.append('k nearest neighbors')
```

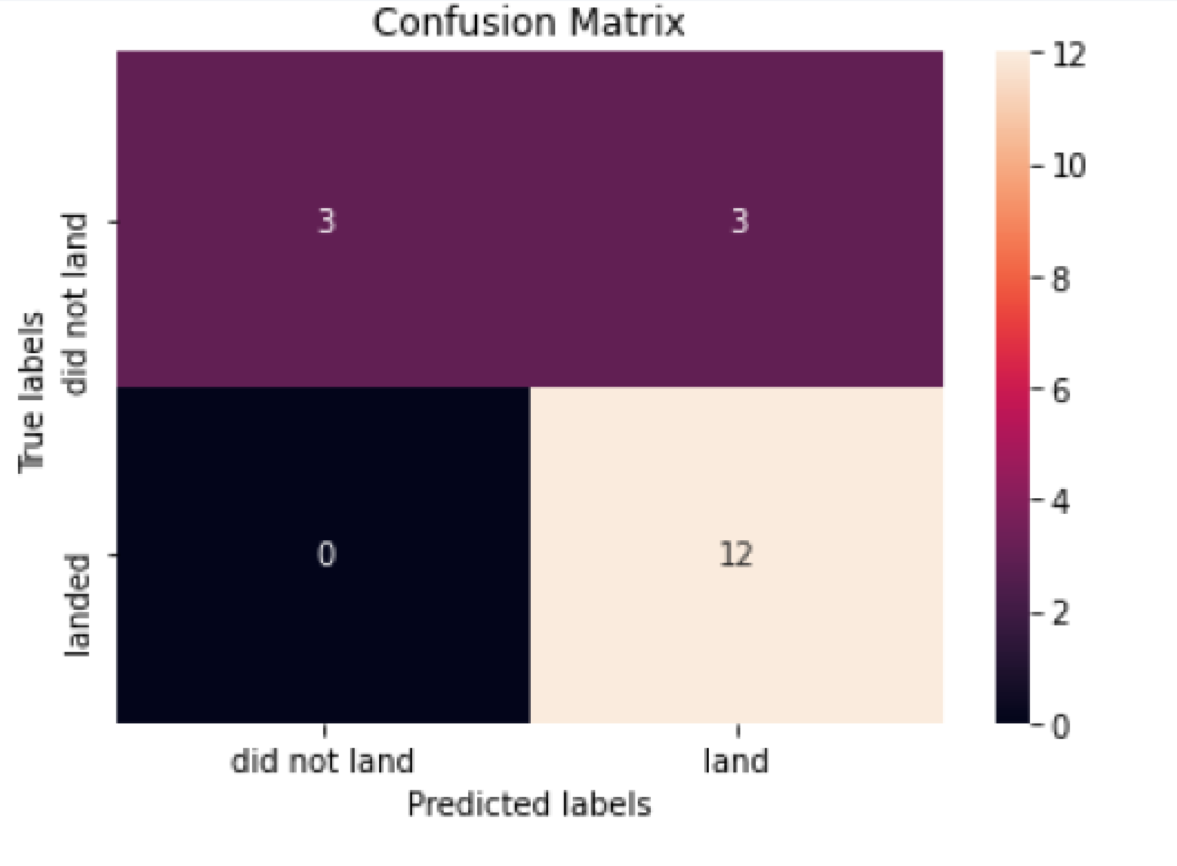
```
print(methods)
print(accu)
```

```
['logistic regression', 'support vector machine', 'decision tree classifier', 'k nearest neighbors']
[0.8333333333333334, 0.8333333333333334, 0.8888888888888888, 0.8333333333333334]
```

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

# Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.
- We see that the major problem is false positives.



# Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1, Orbits with 0% success rate: SO
- The success rate of launches increases over the years.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

# Thank you!