IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tyler Jenson
June 5th, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - The use of API's for data collection in form of CSV files
  - The use of Web Scraping for data collection in form of CSV files
  - Data Wrangling
  - The use of Structured Query Language (SQL), one of the most common data analysis tools, for Exploratory Data Analysis
  - Utilized Data Visualizations to assist with the task of Exploratory Data Analysis
  - Used Folium, interactive dashboard in the form of maps, to better understand our data
  - The use of Machine Learning for predictions and the accuracy of such predictions
- Summary of all results
  - Result of Exploratory Data Analysis
  - Screenshots of the Interactive Analytics (Folium, Plotly Dash)
  - Result of Predictive Analytics

Introduction

- Project background and context
  - On its website, Space X promotes Falcon 9 rocket launches for 62 million dollars; other suppliers charge upwards of 165 million dollars for each launch. A large portion of the savings is due to Space X's ability to reuse the first stage. So, if we can figure out whether the first stage will land, we can figure out how much a launch will cost. If another business wishes to submit a proposal for a rocket launch against space X, they can use this information. The project's objective is to build a pipeline for machine learning that can forecast if the initial stage will land successfully
- Problems you want to find answers
  - What elements determine whether the rocket will successfully land?
  - The way that different elements interact to affect the likelihood of a successful landing
  - What operational requirements must be met to guarantee a successful landing program

Section 1

Methodology

Methodology

Executive Summary

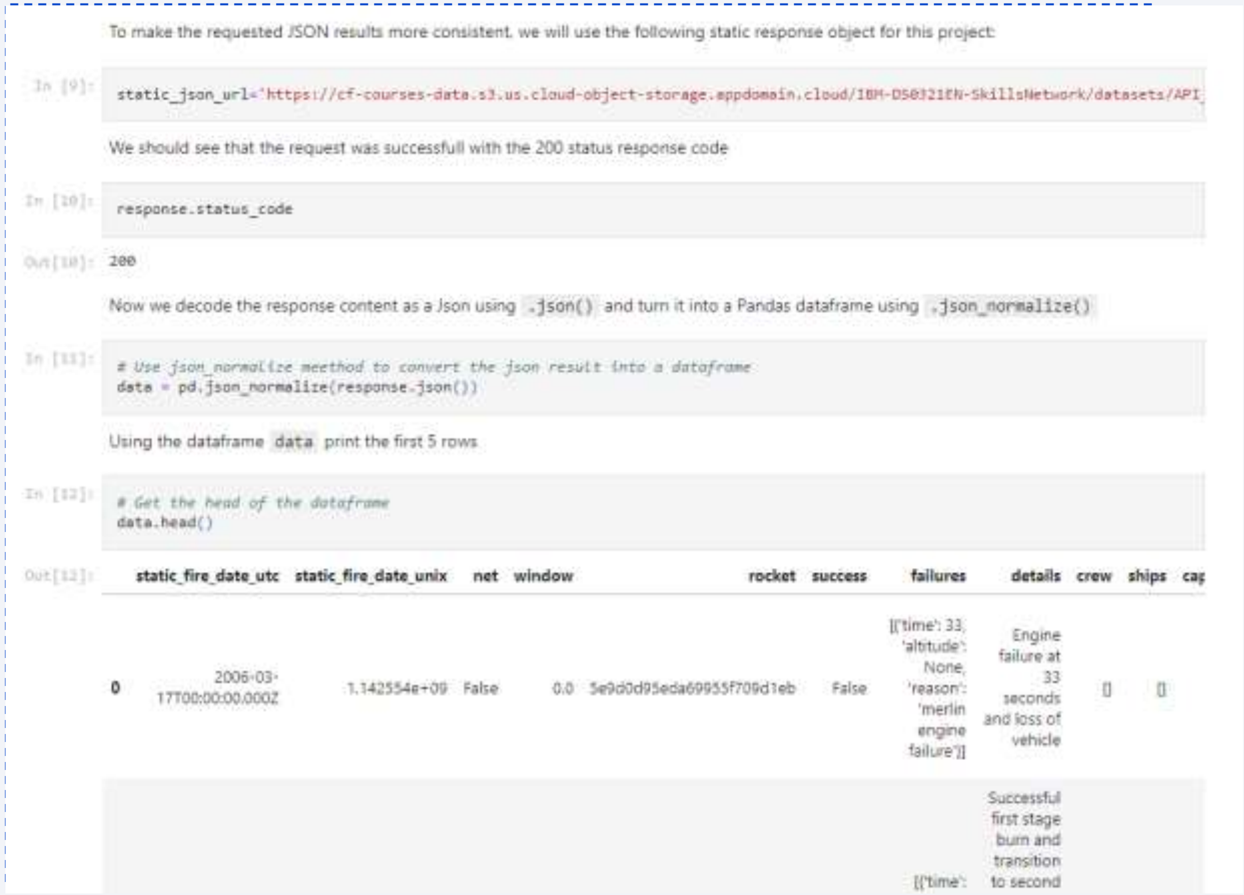
- Data collection methodology:
  - Data was gathered using the SpaceX API and Wikipedia web scraping
- Perform data wrangling
  - We used one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The information was gathered in numerous ways
  - Utilizing a get call to the SpaceX API, data was gathered
  - Next, we used the.json() function call to decode the response's content as JSON and the.json_normalize() method to convert it to a pandas dataframe
  - The data was then cleansed, missing values were checked for, and filled in as appropriate
  - Additionally, using BeautifulSoup, we scraped Wikipedia for information on Falcon 9 launch statistics
  - The goal was to extract the launch records as an HTML table, parse the table, and then transform the table into a pandas dataframe for later analysis

Data Collection – SpaceX API

- To gather the data for SpaceX, we cleaned the requested data, and did some simple data wrangling and formatting, we used the get request to the SpaceX API
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(1.1)%20Collecting%20the%20Data.ipynb



Data Collection - Scraping

- Web scraping was utilized to obtain Falcon 9 launch records using the resource BeautifulSoup
- Once we obtain the data, parsing was required and converted the original table to a pandas dataframe
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(1.2)%20Data%20Collection%20with%20Web%20Scraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:
```python
# 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 [6]:
```python
# 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 [7]:
```python
# Use soup.title attribute
print(soup.title)
```

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

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

Next, we want to collect all relevant column names from the HTML table header

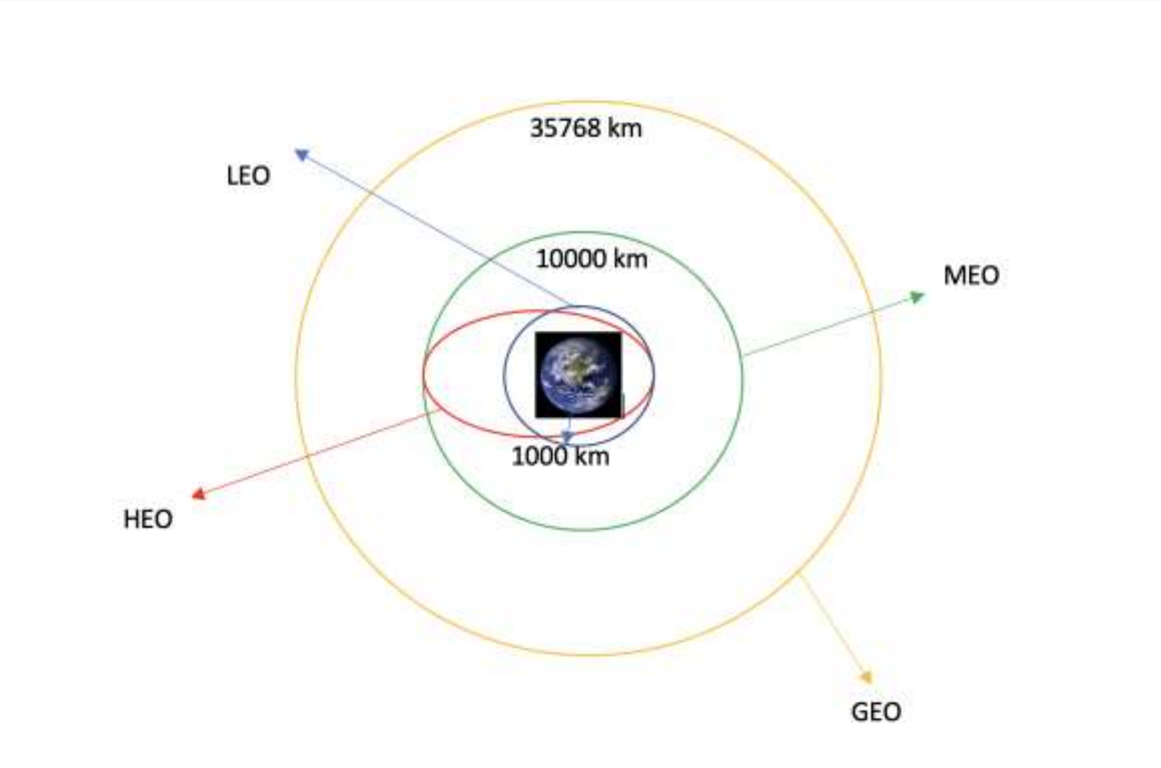
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

In [8]:
```python
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

```
[<table class="multicol" role="presentation" style="border-collapse: collapse; padding: 0; border: 0; background:transparent;
width:100%;">
<tbody><tr>
<td style="text-align: left; vertical-align: top;">
<h3><span class="mw-headline" id="Rocket_configurations">Rocket configurations</span></h3>
<div class="chart noresize" style="margin-top:1em;max-width:420px;">
<div style="position:relative;min-height:320px;min-width:420px;max-width:420px;">
<div style="float:right;position:relative;min-height:240px;min-width:320px;max-width:320px;border-left:1px black solid;border-
bottom:1px black solid;">
<div style="position:absolute;left:3px;top:224px;height:15px;min-width:18px;max-width:18px;background-color:LightSteelBlue;-we
bkit-print-color-adjust:exact;border:1px solid LightSteelBlue;border-bottom:none;overflow:hidden;" title="[[Falcon 9 v1.0]]:
2"></div>
<div style="position:absolute;left:55px;top:224px;height:15px;min-width:18px;max-width:18px;background-color:LightSteelBlue;-w
```

Data Wrangling

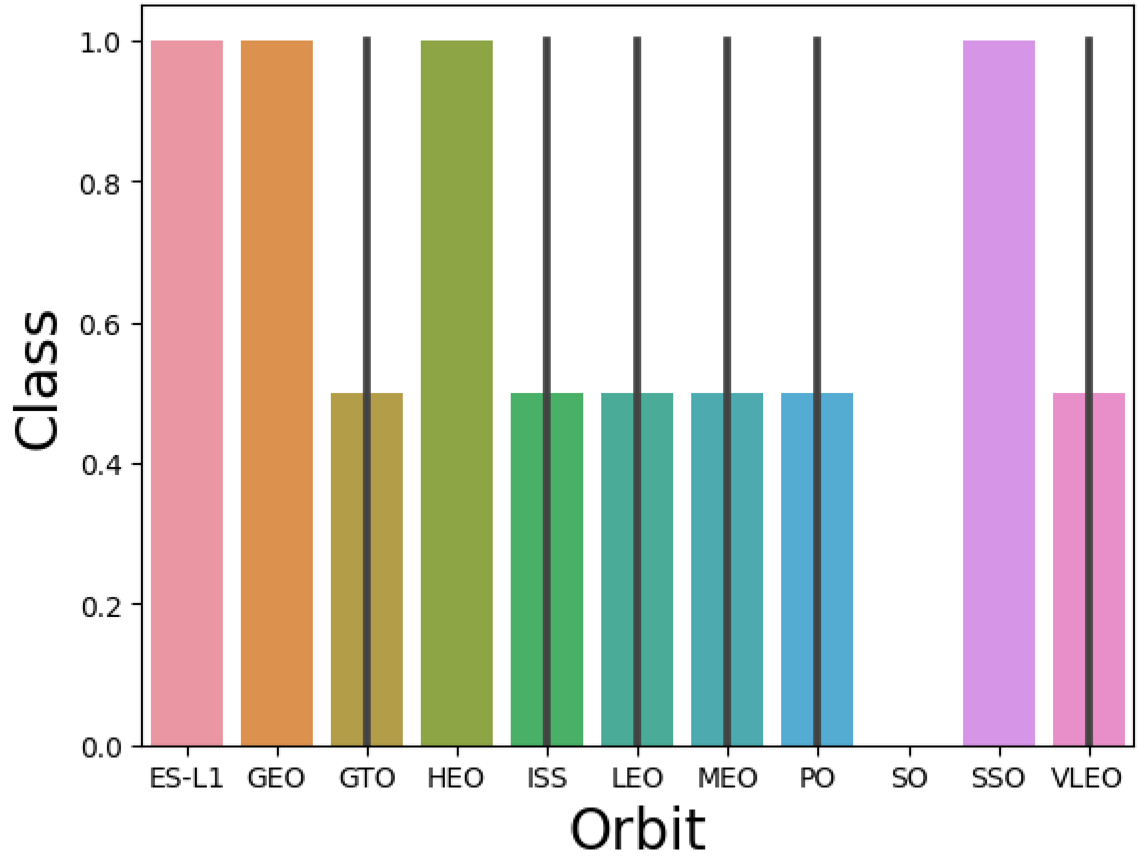
- Exploratory data analysis was done to establish the training labels
- We determined the number of launches at each location as well as the frequency and number of orbits
- From the outcome column, we established a landing outcome label, and then exported the data to a CSV file
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(1.3)%20Data%20Wrangling.ipynb

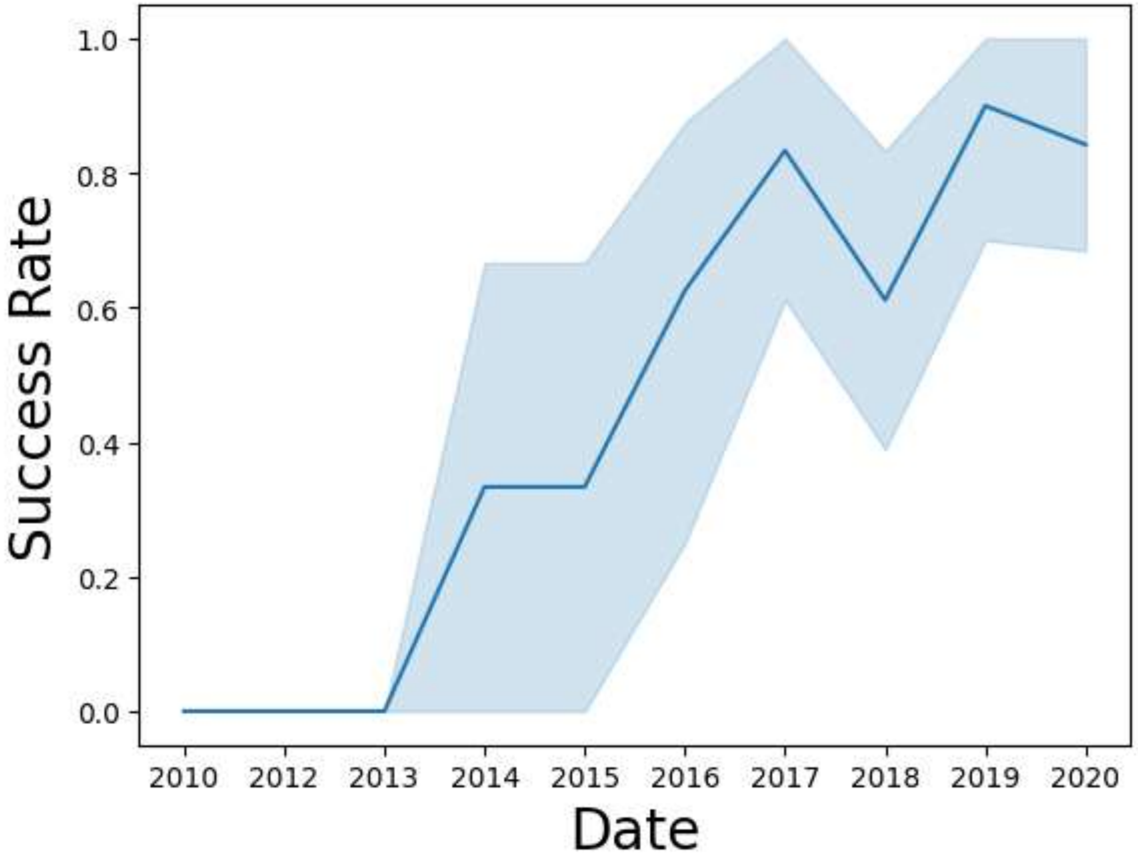


EDA with Data Visualization

- By displaying the relationship between the flight number and the launch site, the payload and the launch site, the success rate of each orbit type, the flight number and the orbit type, and the yearly trend in launch success, we investigated the data
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(2.2)%20Exploratory%20Data%20Analysis%20using%20Pandas%20%26%20Matplotlib.ipynb

EDA with Data Visualization, Continued





EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook
- To gain understanding from the data, we used SQL for EDA. We created queries to research, for example:
  - The names of distinctive launch sites used in space missions
  - The complete weight of payloads carried by rockets fired by NASA
  - The typical payload mass that the booster type F9 carries
  - The total number of mission successes and failures
  - The drone ship's booster version, launch location names, and the results of the botched landing
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(2.1)%20Exploratory%20Data%20Analysis%20with%20SQL%20(EXAMPLE).ipynb

Build an Interactive Map with Folium

- On the folium map, we identified every launch point and added map elements like markers, circles, and lines to indicate whether a launch was successful or unsuccessful for each location.Explain why you added those objects
- We categorized the feature launch results, whether they were a success or failure, into classes 0 and 1.0 represents failure while 1 represents success
- The launch sites with a comparatively high success rate were determined using the color-labeled marker clusters
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(3.1)%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- In addition to the interactive dashboard with Folium, we also prepared another interactive dashboard with Plotly Dash
- A pie chart was created and used to show the total launches by each site
- A scatter plot was created and used to show the relationship between variables ‘Outcome’ & ‘Payload Mass (Kg)’ with various boosters
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(3.2)%20Interactive%20Visual%20Analytics%20with%20Plotly%20Dash.pdf

Predictive Analysis (Classification)

- Using NumPy and Pandas, we loaded the data, transformed it, and divided it into training and testing sets
- Using GridSearchCV, we constructed various machine learning models and tuned various hyperparameters
- Our model was measured by accuracy, and it was enhanced through feature engineering and algorithm tweaking
- We then ran some code to make the decision of the most effective classification model we performed
- Link: https://github.com/TylerJenson/Applied-Data-Science-Capstone---IBM/blob/main/(4.1)%20Machine%20Learning%20Prediction.ipynb

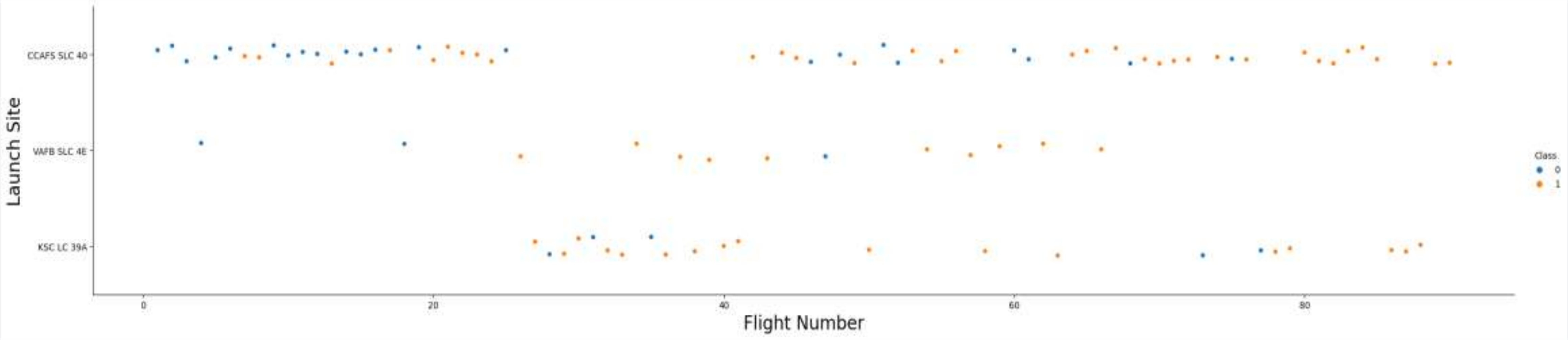
Results

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

Section 2

Insights drawn from EDA

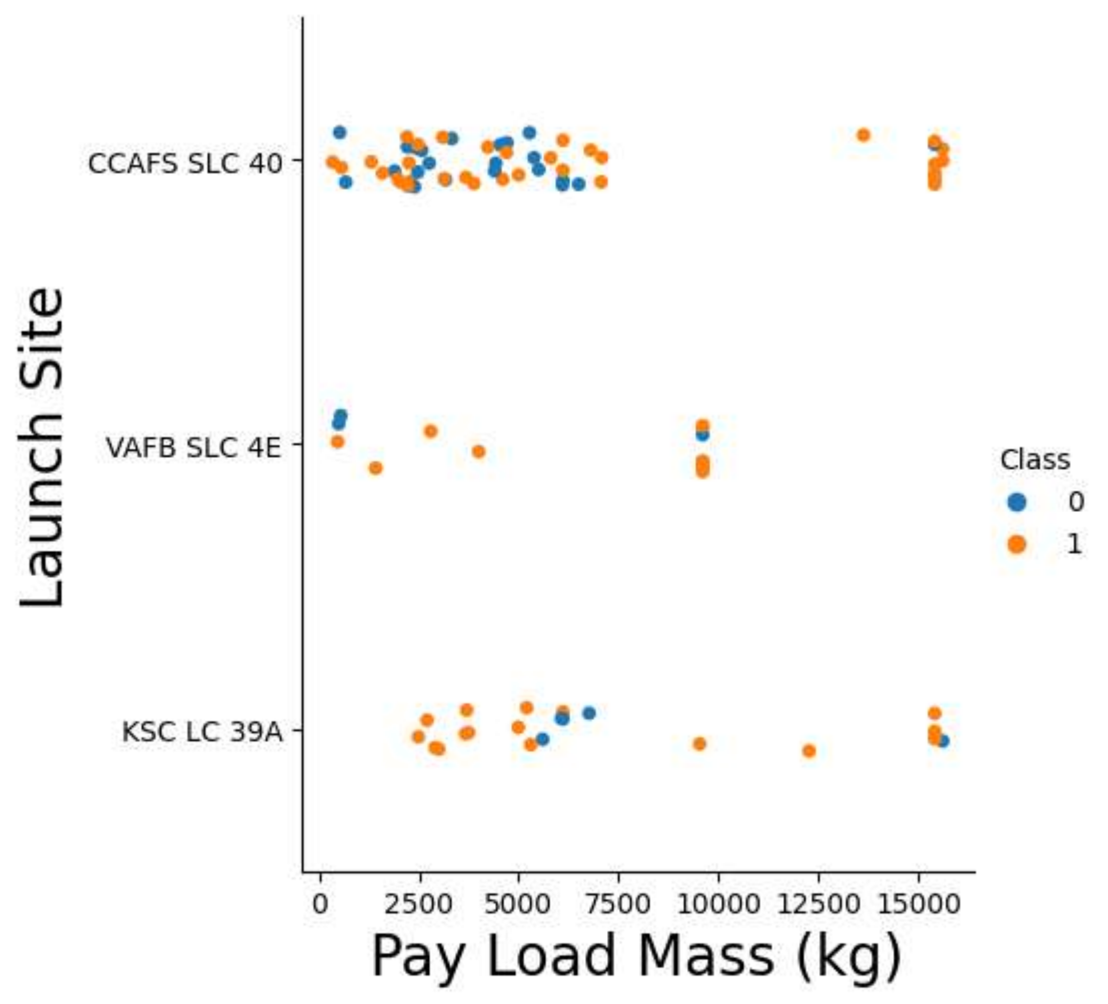
Flight Number vs. Launch Site



- The plot led us to the conclusion that a launch site's success rate increases with the size of the flight quantity

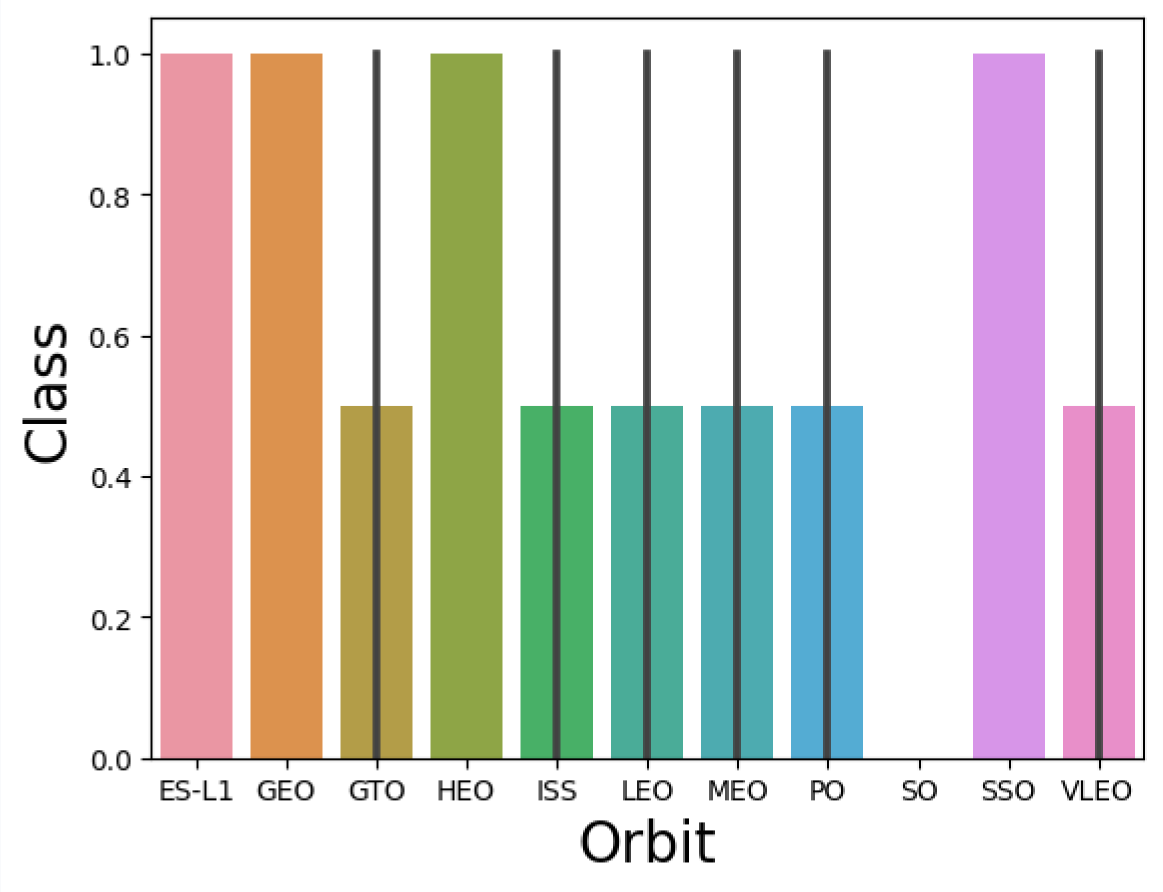
Payload vs. Launch Site

- We found that for launch point CCAFS SLC 40, the heavier the payload mass the higher the rocker's success rate



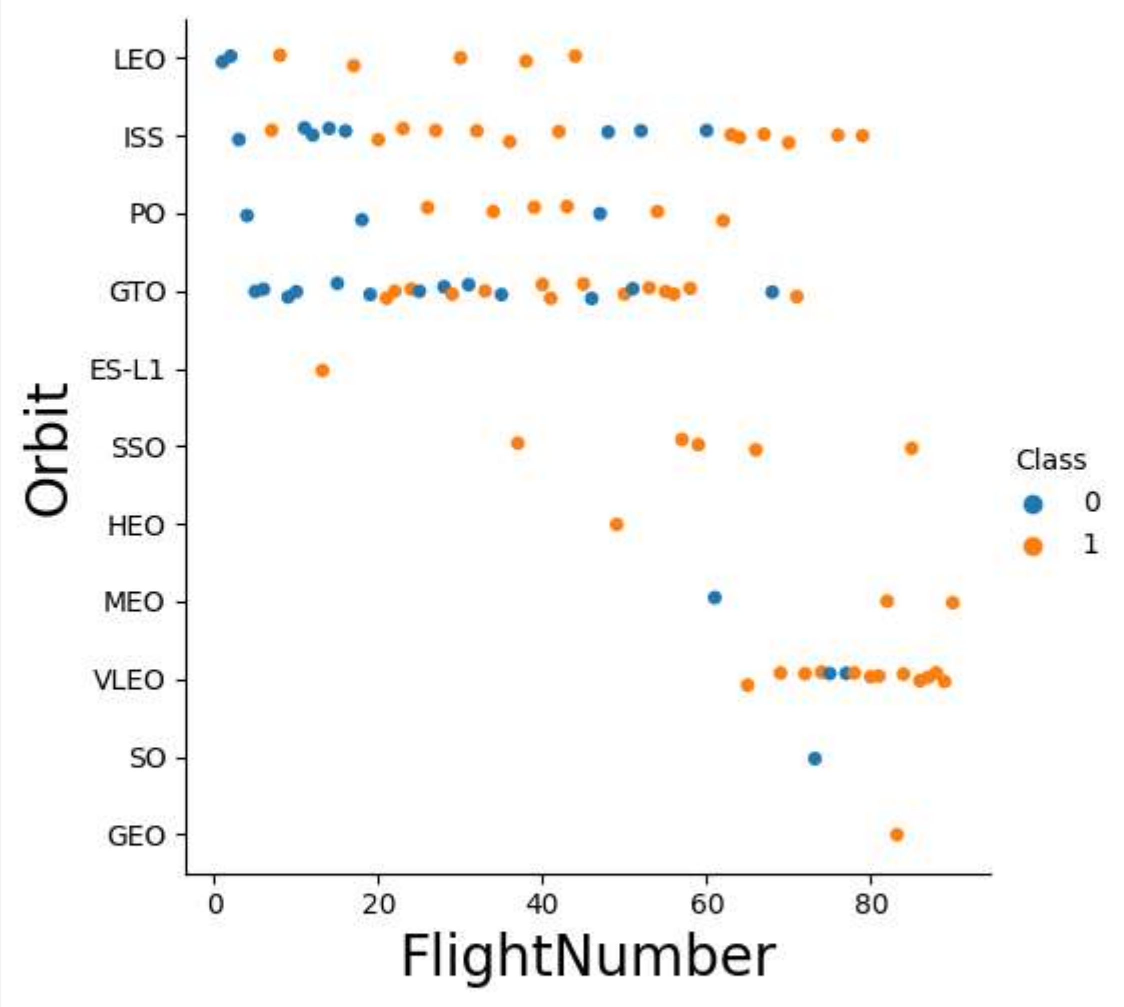
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, and SSO had the most success rate compared to the other types of orbit



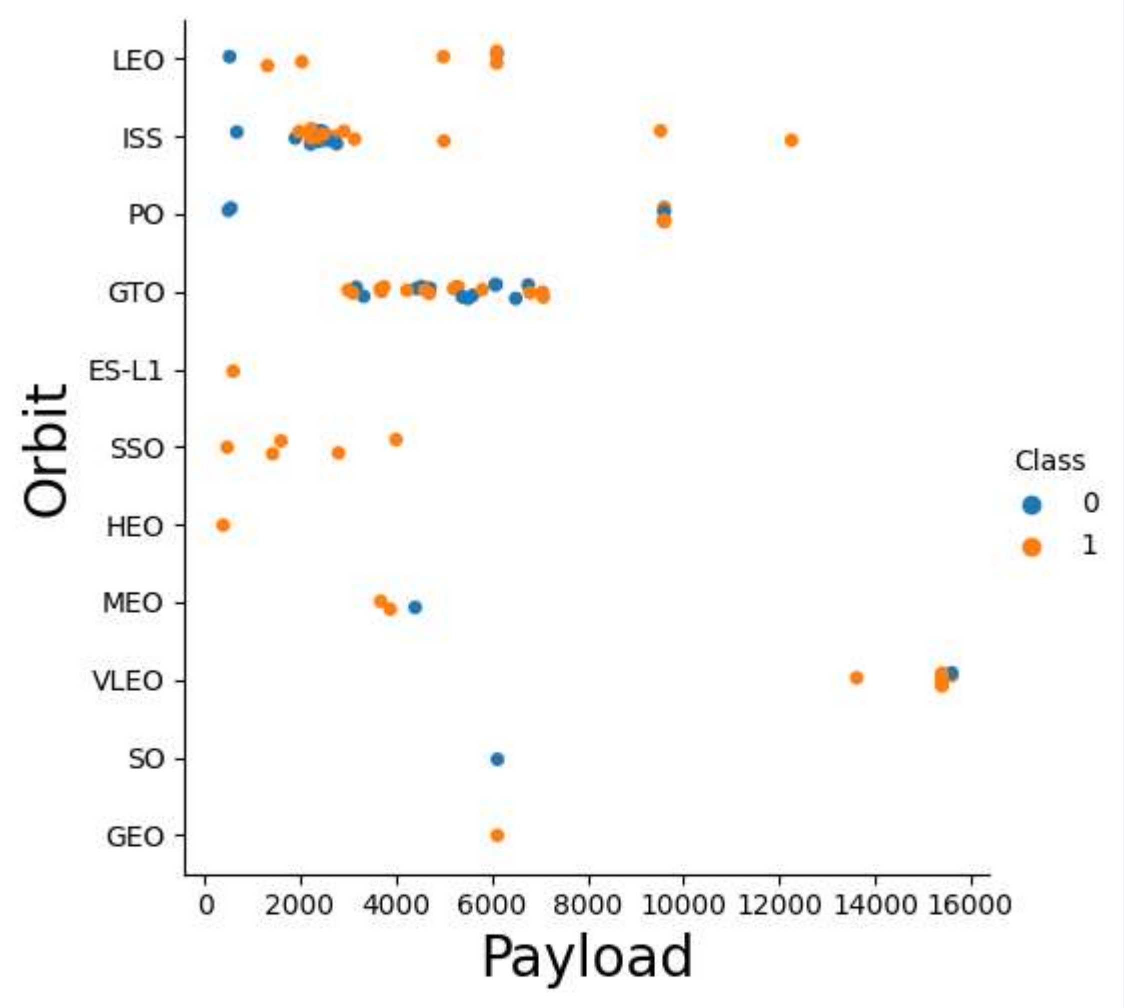
Flight Number vs. Orbit Type

- The plot of the Flight Number versus Orbit type is shown below. We note that success in the LEO orbit is correlated with the number of flights, however there is no correlation between the number of flights and the GTO orbit



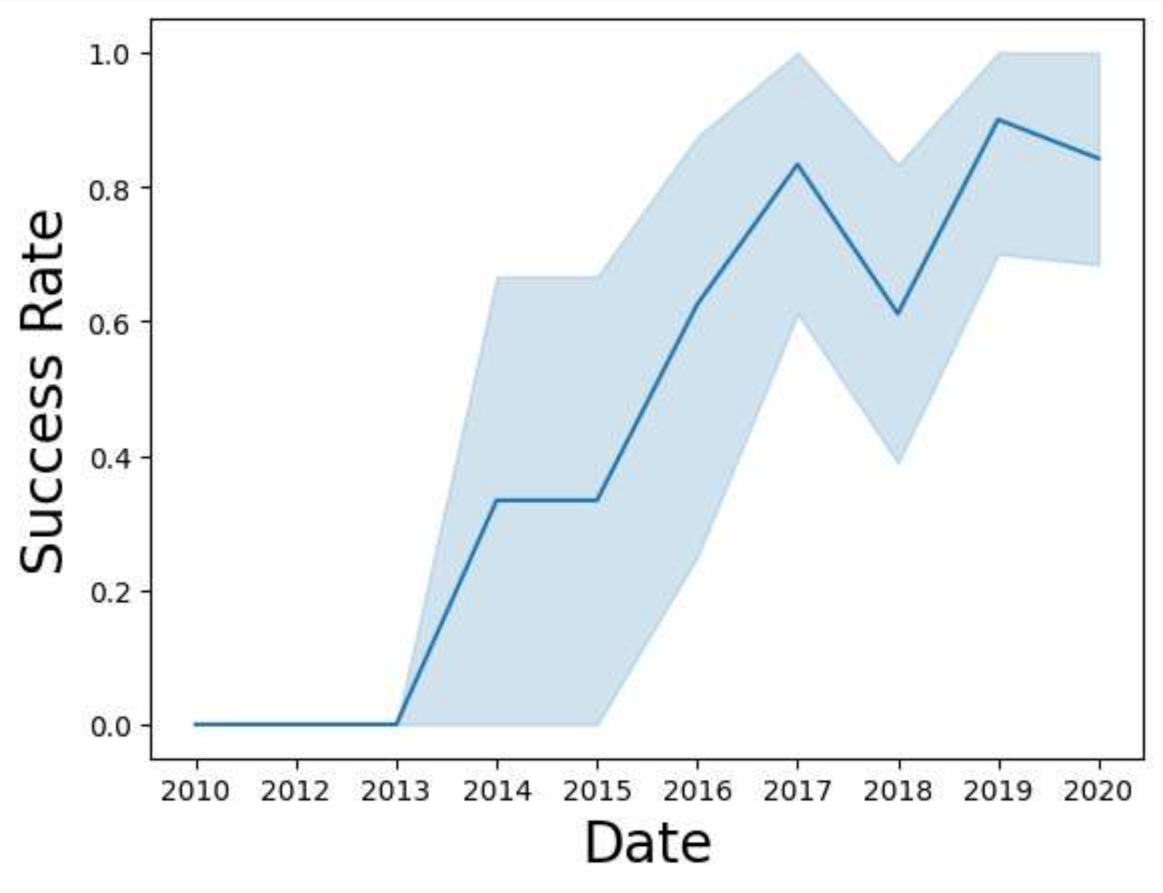
Payload vs. Orbit Type

- We can see that landings with heavier payloads tend to be more successful in PO, LEO, and ISS orbits



Launch Success Yearly Trend

- Based on the line plot, we can conclude that the success rates have increased since 2013, into 2020



All Launch Site Names

- When finding the distinct launch names, we utilized Structured Query Language
- The DISTINCT keyword was used to only show the unique launch sites

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

In [39]:
```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[39]:

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

Launch Site Names Begin with 'CCA'

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

In [40]:
```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[40]:

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

- We performed the aforementioned query to show 5 records for launch sites that start with "CCA"

Total Payload Mass

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

```
In [41]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'

 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[41]:

1
45596

- Using the above query, we determined that NASA's boosters carried a total of 45596 kilograms of payload

Average Payload Mass by F9 v1.1

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

In [42]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[42]:

1
2928.400000

- The calculated average payload mass carried by the Falcon 9 booster was 2928.4 kg

First Successful Ground Landing Date

- We noted that the first successful landing result on the ground pad occurred on December 22, 2015

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

Hint:Use min function

In [43]:
```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[43]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- In order to find boosters that have successfully landed on drone ships, we employed the WHERE clause. We then used the AND condition to identify successful landings with payload masses larger than 4,000 but less than 6,000

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

In [44]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[44]:

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

Total Number of Successful and Failure Mission Outcomes

- The SELECT statement was used along with the subsequent count, from, and where syntax to arrive at the outcomes

List the total number of successful and failure mission outcomes

```
In [45]: %sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in fligh

 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[45]:

1
100

Boosters Carried Maximum Payload

- Using a subquery in the WHERE clause and the MAX() method, we were able to identify the booster that had carried the most payload

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

In [46]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[46]:

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

- In order to filter for failure landing outcomes in drone ship, their booster versions, and launch site names for the year 2015, we employed permutations of the WHERE clause, LIKE, AND, and BETWEEN conditions

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

```
In [47]: %sql SELECT EXTRACT(MONTH, select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)')
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
(ibm_db_dbi.ProgrammingError) ibm_db_dbi::ProgrammingError: SQLNumResultCols failed: [IBM][CLI Driver][DB2/LINUXX8664] SQL0104
N  An unexpected token "EXTRACT(MONTH, select" was found following "SELECT ".  Expected tokens may include:  "<space>".  SQLST
ATE=42601 SQLCODE=-104
[SQL: SELECT EXTRACT(MONTH, select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)' )]
(Background on this error at: http://sqlalche.me/e/f405)
```

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

- With the help of the WHERE clause, we were able to filter the data for landing outcomes between 2010-06-04 and 2010-03-20 by choosing Landing outcomes and the COUNT of landing outcomes
- The landing outcomes were grouped using the GROUP BY clause, and the grouped landing outcomes were then sorted by the ORDER BY clause

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

In [48]:

```
%sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order
```

```
 * ibm_db_sa://sdk38546:***@dashdb-txn-sbox-yp-lon02-07.services.eu-gb.bluemix.net:50000/BLUDB
Done.
```

Out[48]:

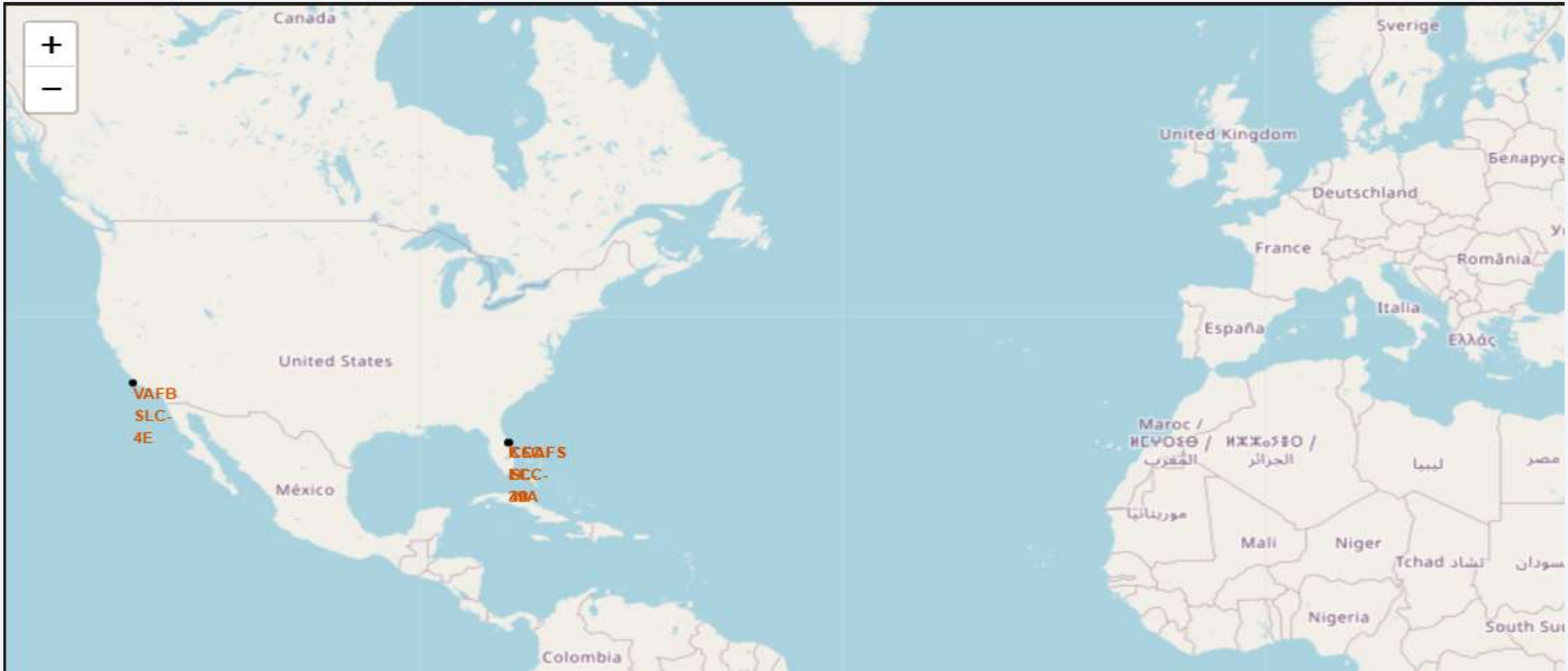
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcon
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (groun pa
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (dron shi
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (dron shi
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (groun pa
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (dron shi
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (dron shi
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (dron shi
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (groun pa

Section 3

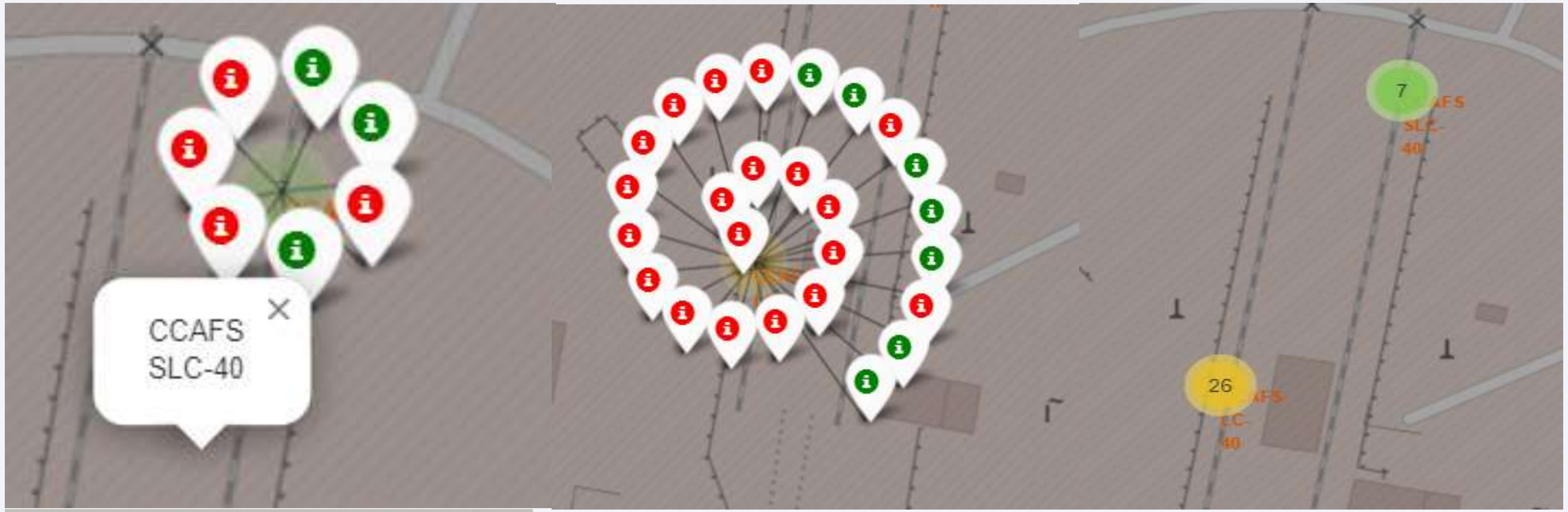
Launch Sites Proximities Analysis

Launch sites global map markers

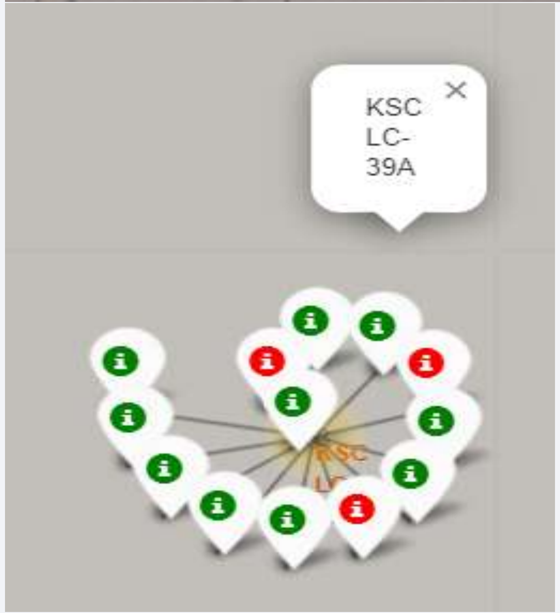
- We can observe that the SpaceX launch locations are on American coastlines. California and Florida



Markers with colored labels indicating launch sites

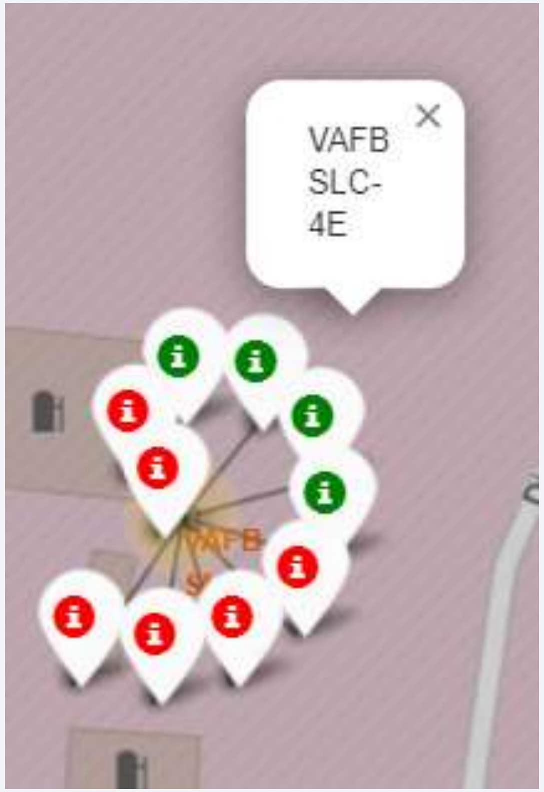


CCAFS LC-40 (fig. above)



Florida Launch Sites

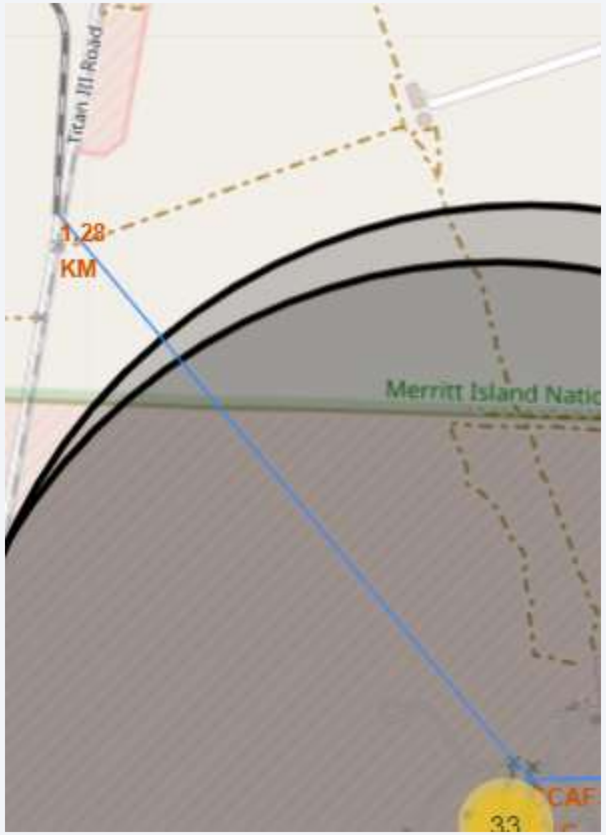
Green Marker shows successful launches, and the Red Marker shows failed launches



California Launch Site

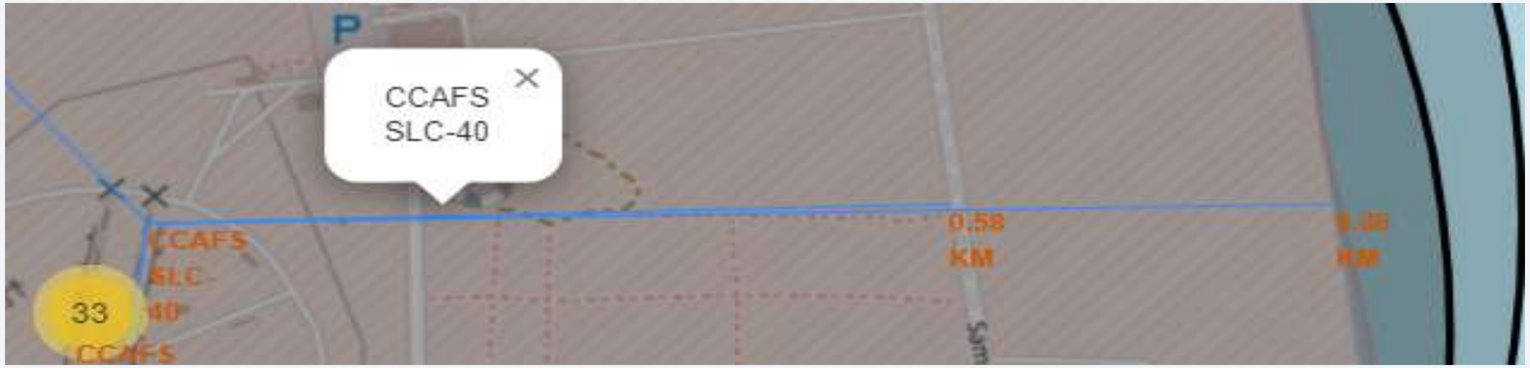
CCAFS SLC-40 Launch Site distance to nearest landmarks

Distance to coast: 0.86 KM



Distance to railway: 1.28 KM

CCAFS SLC-40 Launch Site distance to nearest landmarks



Distance to coast: 0.86 KM

Distance to airport: 51.43 KM

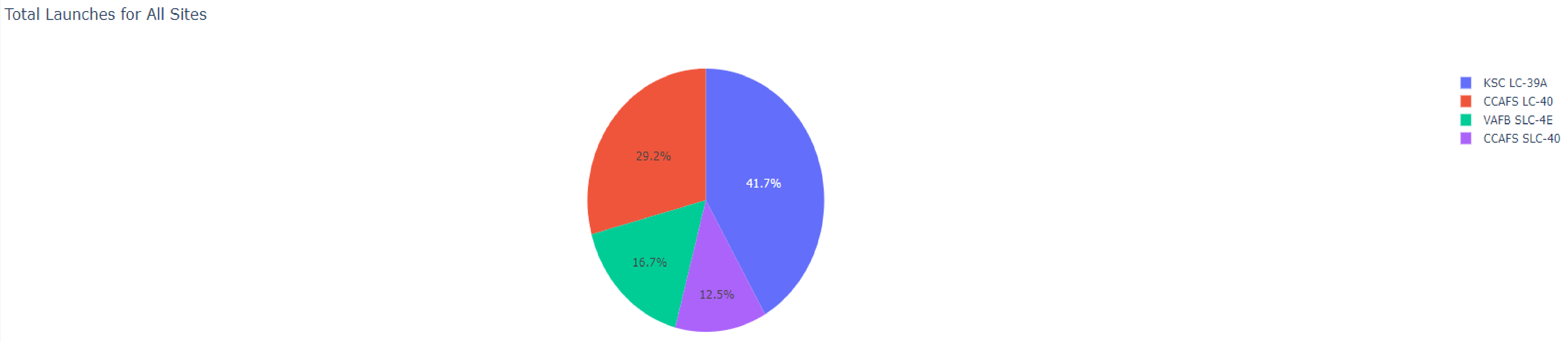
Distance to railway: 1.28 KM

- Are launch sites in close proximity to the coast? Yes
- Are launch sites in close proximity to the airport? No
- Are launch sites in close proximity to the railway? No

Section 4

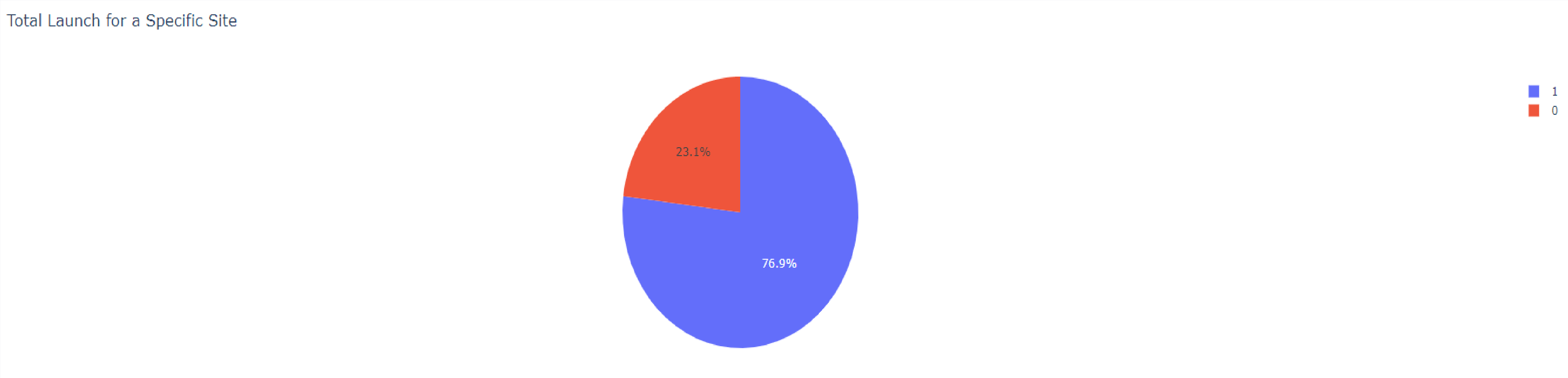
Build a Dashboard with Plotly Dash

Pie chart demonstrating the success rate attained by each launch site



KSC-LC-39A had the most successful launches of all the launch sites

Pie chart displaying the launch site with the highest launch success ratio

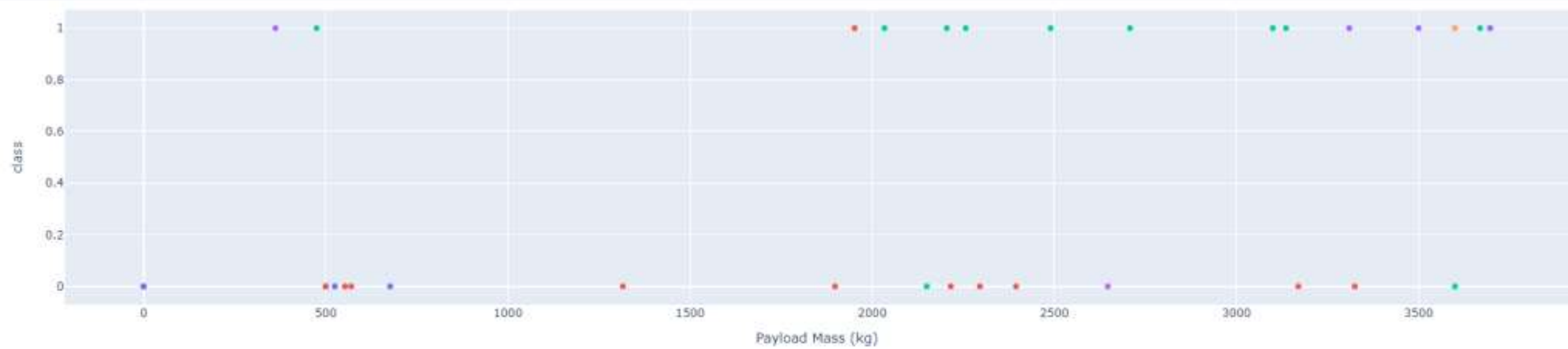


- At a success rate of 76.9%, KSC LC-39A had the highest of all the launch sites

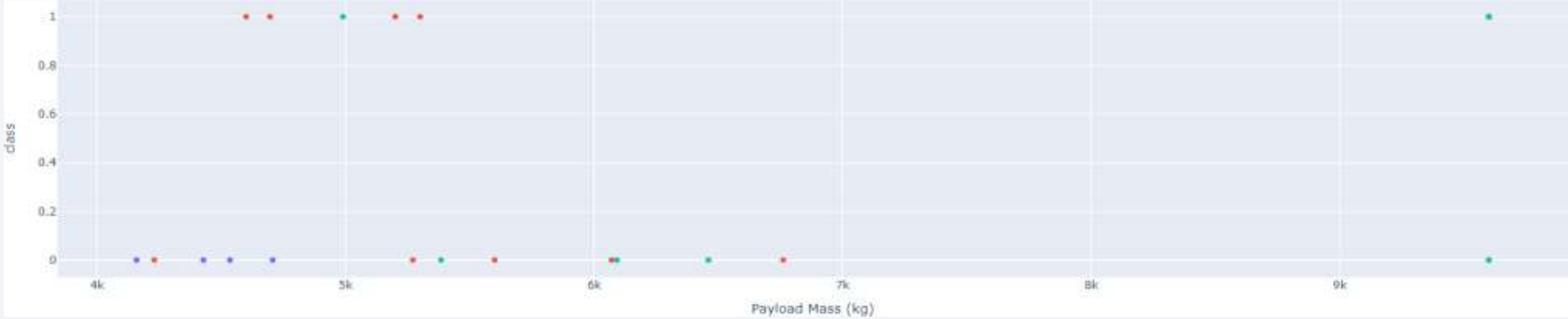
Payload vs. Launch Outcome scatter plot for all sites, with various payloads selected using the range slider

Class: 1 signifies success, 0 failure

Payload: 0 – 4,000 kg



Payload: 4,000 – 10,000 kg



- Compared to heavy weighted payloads, low weighted payloads have higher success rates

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Not one classification method outperformed one another, so the conclusion could be made in this case that either one of four methods would be optimal

Find the method performs best:

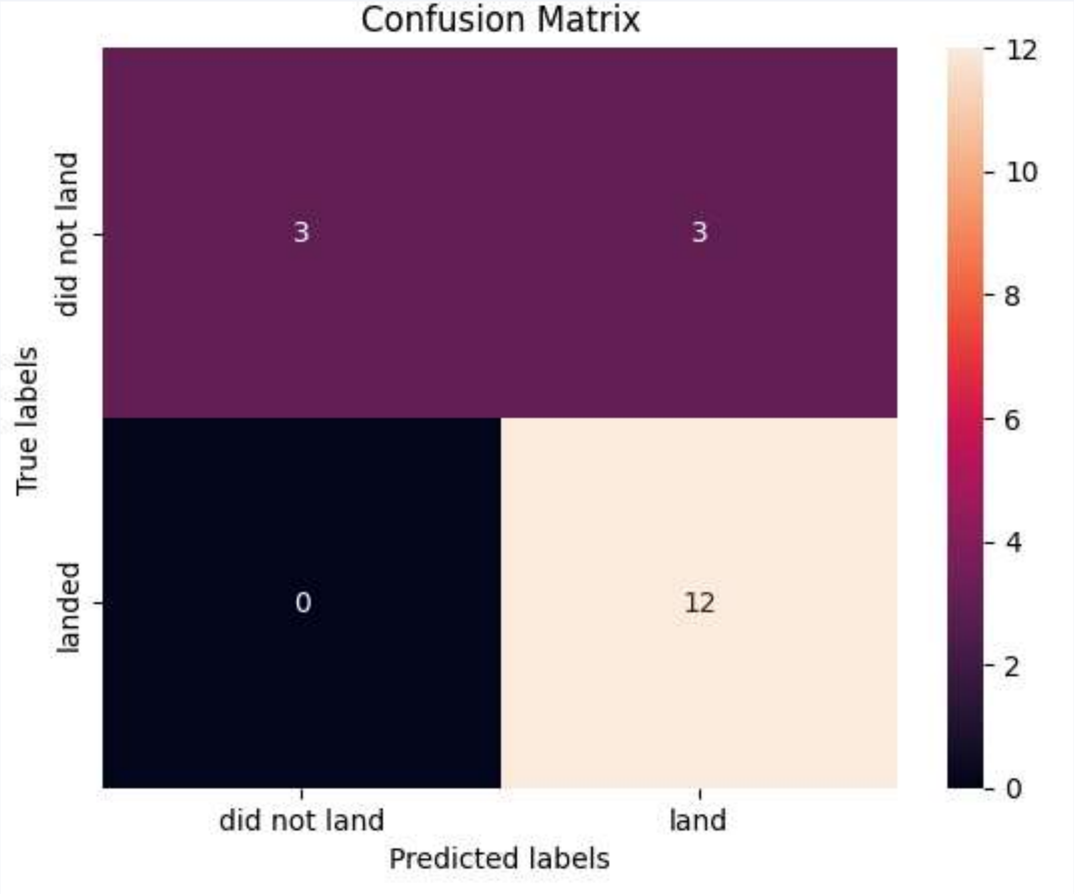
In [34]:

```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearest neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8333333333333334
Accuracy for K nearest neighbors method: 0.8333333333333334
```

Confusion Matrix

- Since neither four classification methods outperformed each other I will be demonstrating the decision tree classifier
- The decision tree classifier's confusion matrix demonstrates that it is capable of differentiating between the various classes. False positives are the main issue. In other words, the classifier misclassified a failure landing as a successful one



Conclusions

- The success rate at a launch site increases with the size of the flight quantity
- Beginning in 2013, the launch success rate will rise through 2020
- ES-L1, GEO, HEO, SSO, and VLEO orbits had the highest success rates
- The most successful launches of any sites were at KSC LC-39A
- Neither one of four classifiers outperformed one another according to the dataset

Thank you!