IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kevin Wu

8/20/2023

Outline

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

Executive Summary

- The goal of this project was to figure out if rocket launches were successful or not, since this implies the first stage could be used which reduces costs drastically. This is important to predicting the price of each launch.
- First, data was obtained from the SpaceX API and web scraping from a Wikipedia page that held past launch records.
- This data was then wrangled to only include Falcon 9 data since that was the main interest, alongside performing one hot encoding to transform categorial features and getting rid of missing values
- Exploratory data analysis was performed with SQL and visualization libraries like seaborn & Folium to get an initial idea of the relationship between different variables
- More data analysis was performed by creating an interactive dashboard with Plotly and Dash, allowing us to change certain features at will to see what the output was
- Finally, we put our dataset in four classification algorithms to find which algorithm would be the most accurate to classify each launch as successful or not.
- It was determined that later launches overall had a better success rate, and in general, launches set to orbits ES-L1, GEO, HEO, SSO, VLEO had the highest success rates. The decision tree model was the most accurate for classification, which should be used for future analysis as needed.

Introduction

- Project background and context

As a data scientist for a new rocket company, we want to be able to successfully predict the price of a rocket launch. A key factor to this is finding if the first stage of the rocket can be reused, as this will drastically reduce the cost of the launch if the first stage lands. We will use pre-existing data from SpaceX with Falcon 9 launches to determine if the first stage will be reused, so we can predict the price of each launch. From this, we know what price to set our own launches with to compete against SpaceX.

- Problems you want to find answers to

What are the best data sources of precious SpaceX records?

What factors of previous SpaceX launches contributed the most to the reuse of the first stage? Any groups of factors in particular?

What model is the best for classifying the success of the reuse of the first stage?

Section 1

Methodology

Methodology

Executive Summary

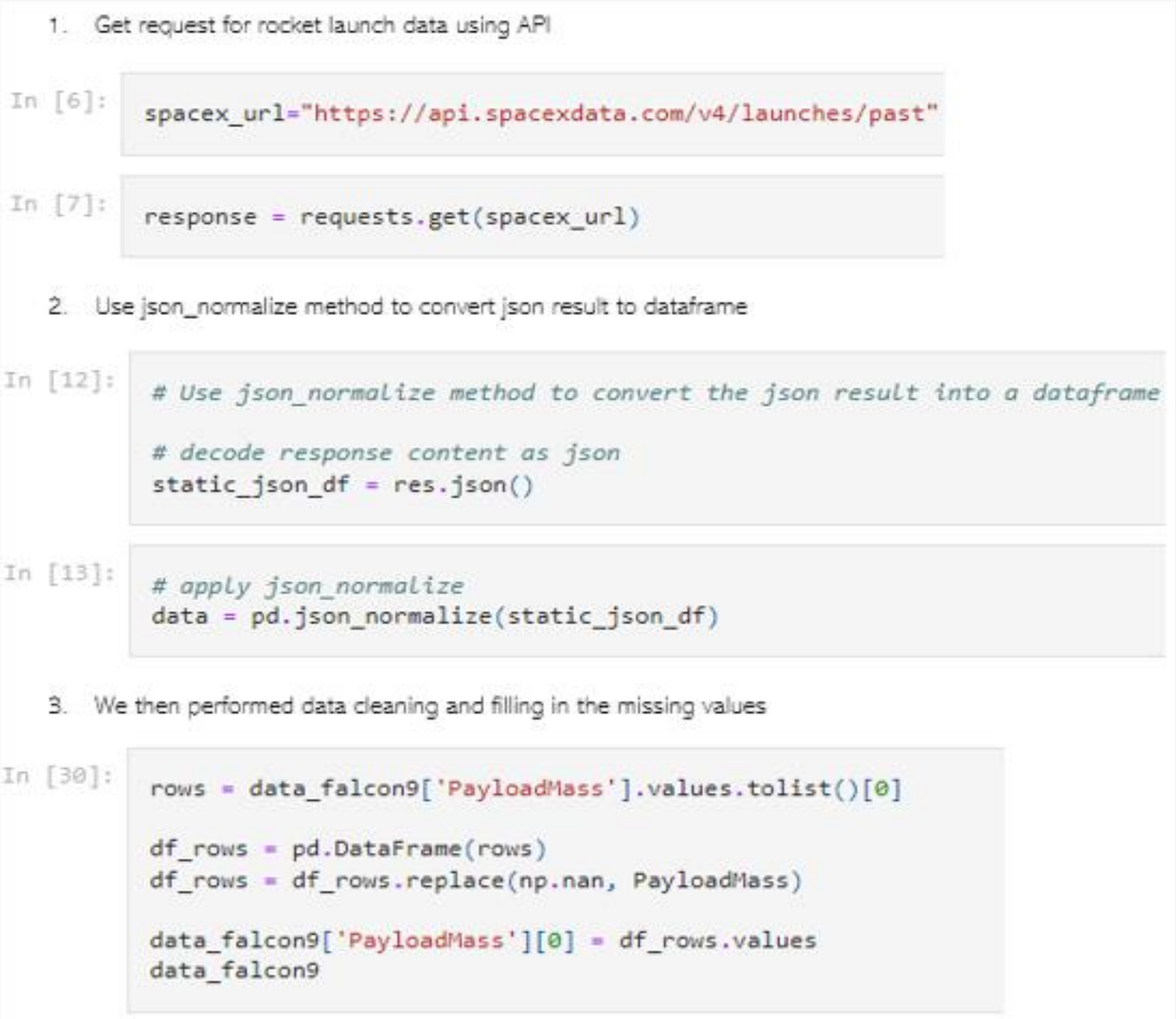
- Data collection methodology:
  - Data was collected using SpaceX API and Wikipedia via webscraping.
- Perform data wrangling
  - Categorial features are converted to binary numerical ones via one hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was collected from both the SpaceX API and from Wikipedia to get all the information we need.
- The requests library was used to start a request with the SpaceX API. With a successful response, the raw data was converted into a DataFrame by normalizing the JSON file.
- The raw data was then manipulated to extract the data we want and in the right format (such as only wanting Falcon 9 launch data and replacing NaN values with the respective mean in the column of interest); A new DataFrame was created with the adjustments.
- The requests library was used to start a request with the Wikipedia page. After a successful response, a Beautiful Soup object was created to parse through the html data on the page.
- The html tables were parsed through to get the data we need. This was formatted in a separate DataFrame.

Data Collection – SpaceX API

- Using a get request to the SpaceX API, data was extracted and then cleaned, wrangled, and formatted into a new Dataframe.
- Notebook Link: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(1)_Collecting_The_Data.ipynb



1. Get request for rocket launch data using API

In [6]:

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

In [7]:

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

2. Use json_normalize method to convert json result to dataframe

In [12]:

```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:

```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

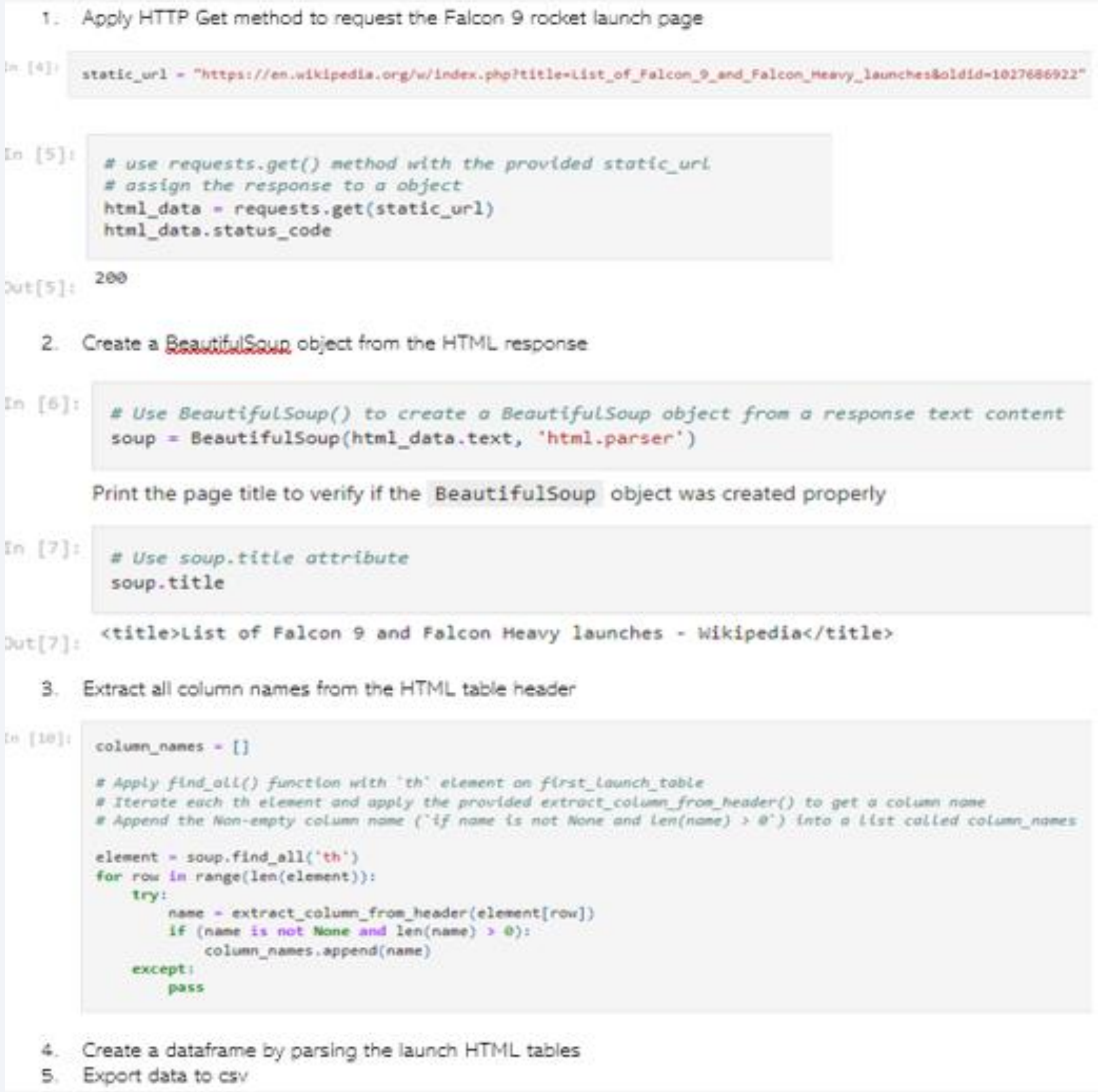
3. We then performed data cleaning and filling in the missing values

In [30]:

```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- The Wikipedia URL was webscrapped and stored in a BeautifulSoup object. The data was parsed to find the html table of interest, and then extracting the data into a Dataframe.
- Notebook Link: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(2)_Webscraping_From_Wikipedia_Records.ipynb



1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url)
        html_data.status_code

Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

```
In [7]: # Use soup.title attribute
        soup.title

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

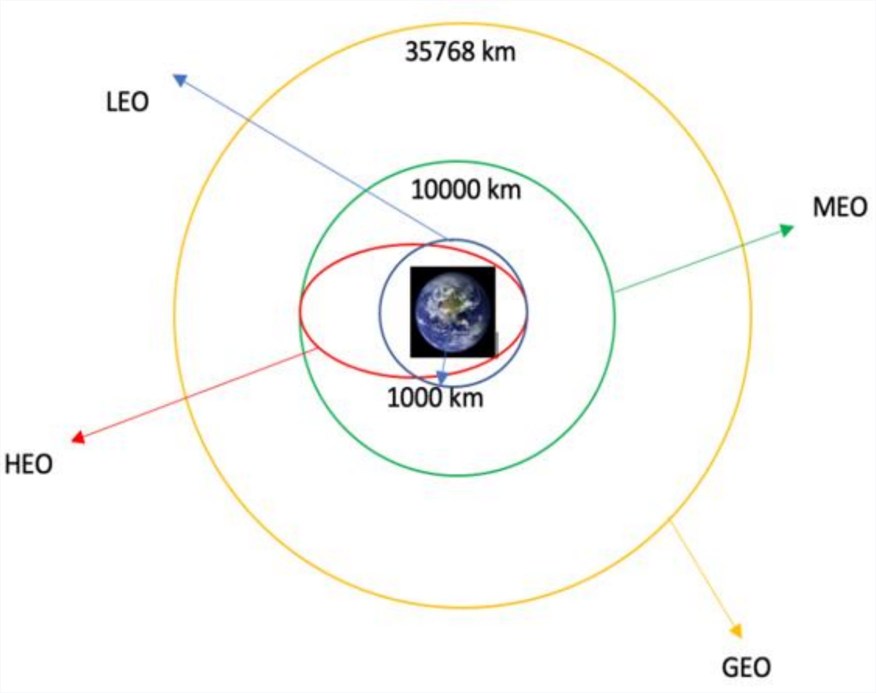
3. Extract all column names from the HTML table header

```
In [10]: 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

         element = soup.find_all('th')
         for row in range(len(element)):
             try:
                 name = extract_column_from_header(element[row])
                 if (name is not None and len(name) > 0):
                     column_names.append(name)
             except:
                 pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

- Introductory data analysis was performed by first finding the number of launches in each site by the value_counts() method.
- Each launch is aimed at a specific orbit, so this was done again by finding the number of launches in each orbit..
- The number in each landing outcome was recorded.
- From this, each landing outcome was categorized as successfully landed and not, and a new column was created to categorize each launch as having a successful landing outcome and not.
- The results were exported for future data analysis.
- Notebook link: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(3)%20Data%20Wrangling.ipynb



EDA with SQL

- Found the names of all unique launch sites alongside first five launch sites that began with ‘CCA’
- Found total payload mass for launches by NASA (CRS), and average payload mass carried by booster version F9 v1.1
- Found date of the first successful landing outcome with a ground pad
- Found total count of all mission outcomes
- Found names of booster versions that carried the maximum payload mass
- Found all records that had a failure with drone ship in the year 2015
- Found total count of all landing outcomes between certain dates in descending order
- Notebook: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(4)%20Exploratory%20Data%20Analysis%20With%20SQL%20Magic.ipynb

EDA with Data Visualization

- Displayed scatterplots with flight number vs other variables with hue set to the class parameter, as they are easy to find individual data points and recognize patterns
- Displayed bar chart with orbit type vs success rate, as to easily see if a specific orbit type leads to a higher success rate
- Displayed line graph comparing date (year) vs success rate, to clearly see how success rate changes as the year progresses in a continuous motion
- Notebook: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(5)%20EDA%20With%20Data%20Visualization%20And%20Preparing%20Data.ipynb

Build an Interactive Map with Folium

- Marked all launch site locations with a dot as a central point
- Marked all individual launches with a line showing distance from launch site, and highlighted them in green/red to imply success/failure respectively; This can be used to see if a specific launch site has a better overall success rate
- Added lines to calculate distance between launch sites and notable landmarks, such as cities, railways, coastlines, and highways; This can be used to see if certain launches have better success being away/close by to these landmarks
- Notebook: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(6)%20Launch%20Site%20Mapping%20With%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Displayed a pie chart displaying all successful launch outcomes between launch sites; This can show which launch sites have the most success
- The above pie chart can also display the number of successful and unsuccessful launch outcomes for a particular launch site
- Displayed a scatterplot showing relationship between payload amount (range can be changed with a slider) and success rate, with hue as the booster version, for all launch sites; This can show if a particular payload amount or booster version leads to a higher success
- The above scatterplot can also display the relationship for just one particular launch site
- Notebook: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(7)%20SpaceX%20Dash%20App%20Code.py

Predictive Analysis (Classification)

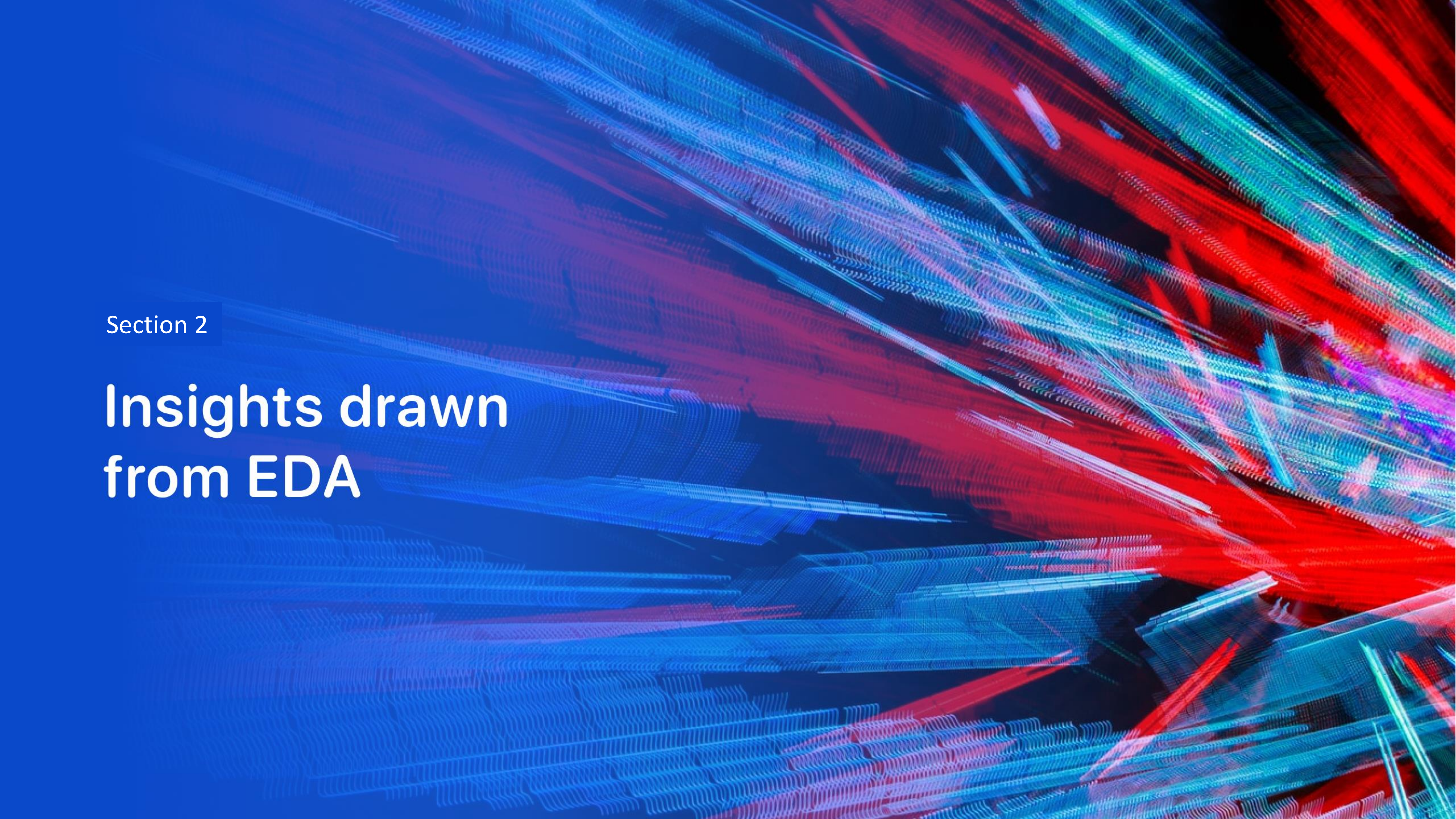
- After inputting the DataFrame with the wrangled data, the data was converted to a NumPy array so we could use the proper libraries
- Data was then standardized as some of our ML algorithms rely on distance between data points, so the ‘effect’ of a certain feature on our classification wouldn’t be unnaturally higher due to having a larger range of numbers
- Data was split into a large training set and a smaller testing set, so we could properly test the accuracy of our models & avoid potential overfitting
- Four ML models were trained: Logistic Regression, Decision Tree, SVM, and KNN
- The best parameters were found using grid search, and the best overall model was found with the one that had the highest score (the highest accuracy)
- Notebook: https://github.com/KevinWu41/Coursera-Capstone-Project/blob/main/(8)_Machine_Learning_Pipeline%20(1).ipynb

Results

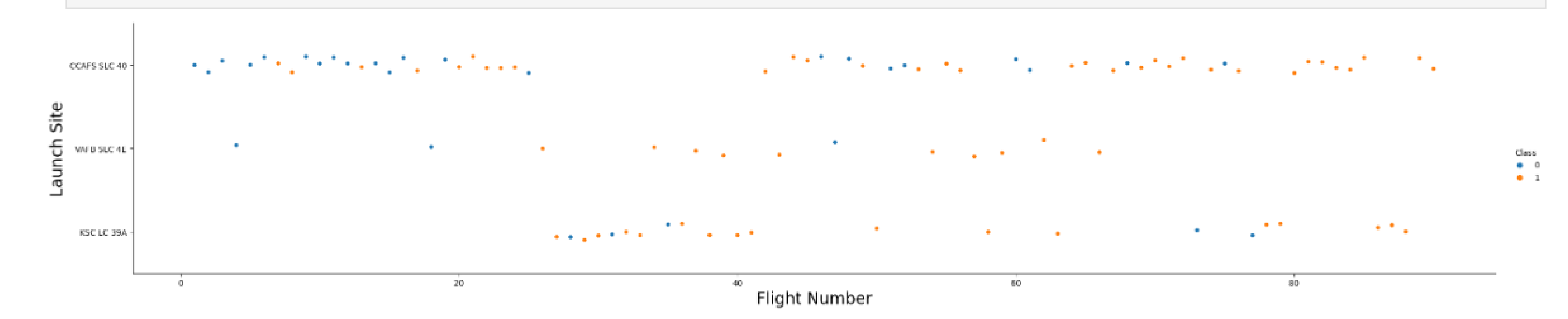
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

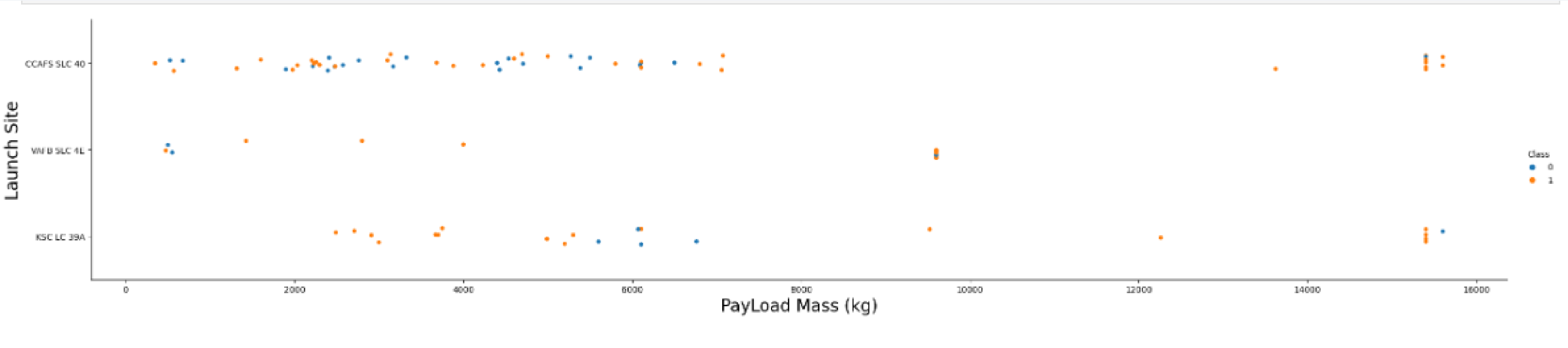


Flight Number vs. Launch Site



For launches at CCAFS SLC 40 and VAFB SLC 4E, success rate seems to go up as the flight number goes up. There does not seem to be more/less successes as the flight number changes for launches at KSC LC-39A.

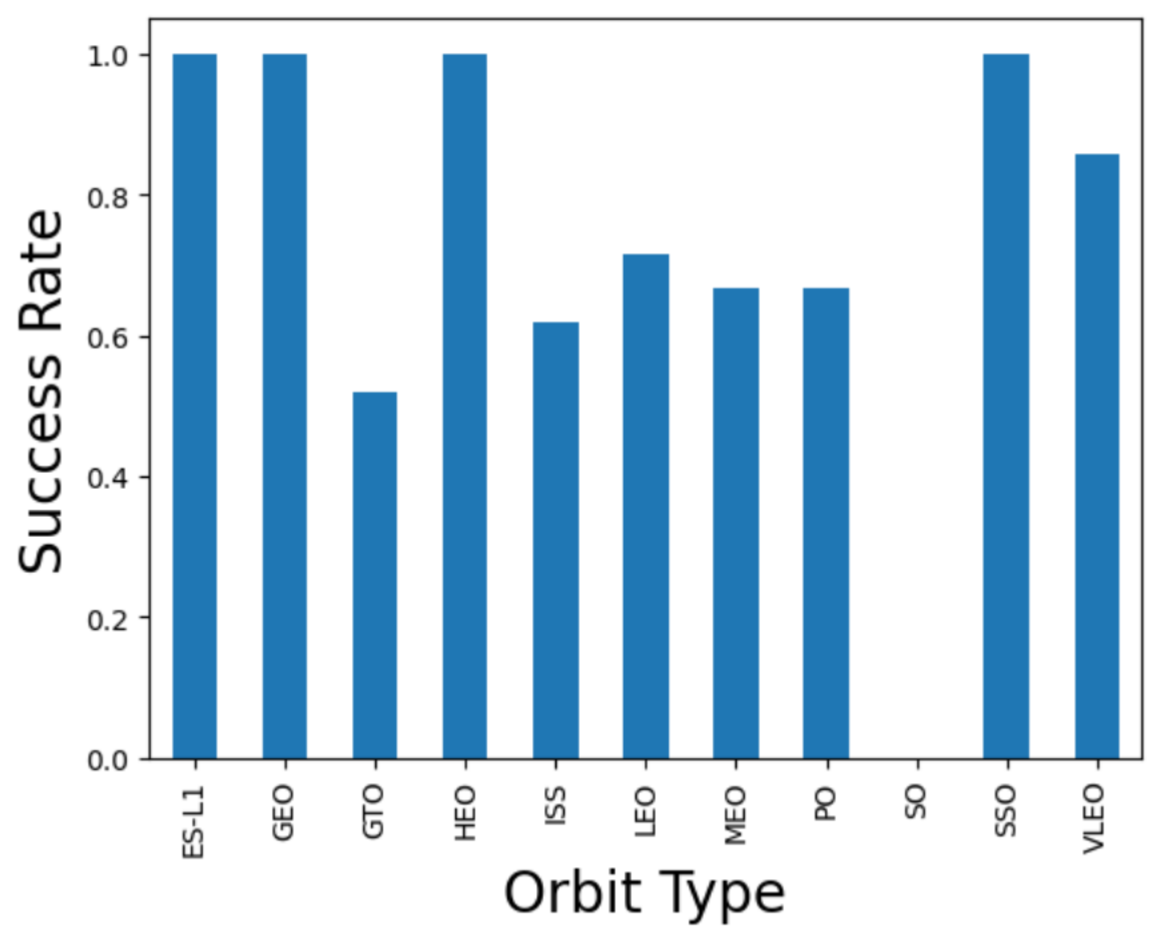
Payload vs. Launch Site



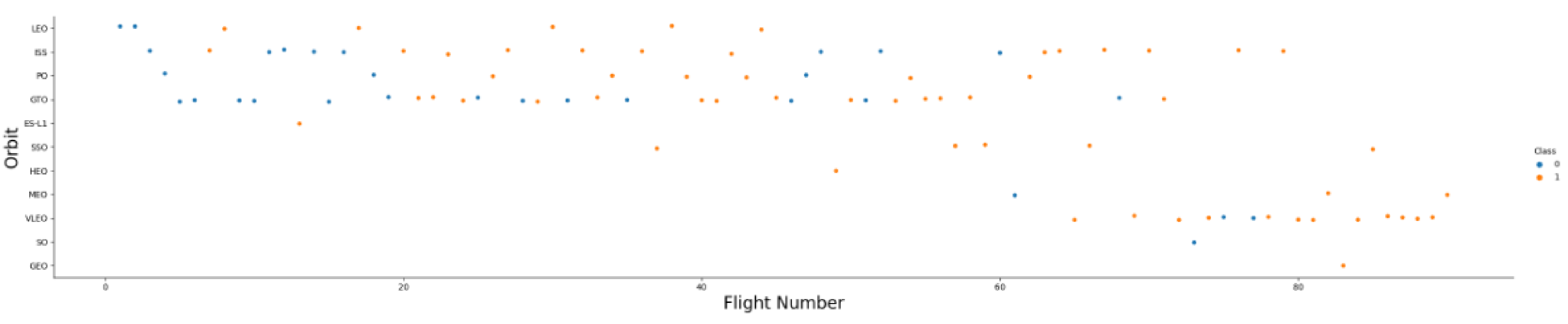
As payload mass increases, there are less launches in general at all launch sites. In particular, VAFB SLC 4E has no launches past a payload mass of 10,000 kg, and the other two launch sites have a few launch sites concentrated between 15,000 and 16,000kg for their payload mass. Payload mass does not seem to be a greater indicator for determining the success of the launch, as there are not significantly more/less successes as the payload mass increases for all launch sites.

Success Rate vs. Orbit Type

We see that aiming for certain orbit types can lead to a high, if not perfect success rate. Orbits ES-L1, GEO, HEO, and SSO have a success rate of 100%. The other orbits have an overall success rate less than 100%, with orbit SO having a success rate of 0%. Future launches aimed at this orbit are most likely not worthwhile if the goal is to get a successful launch.

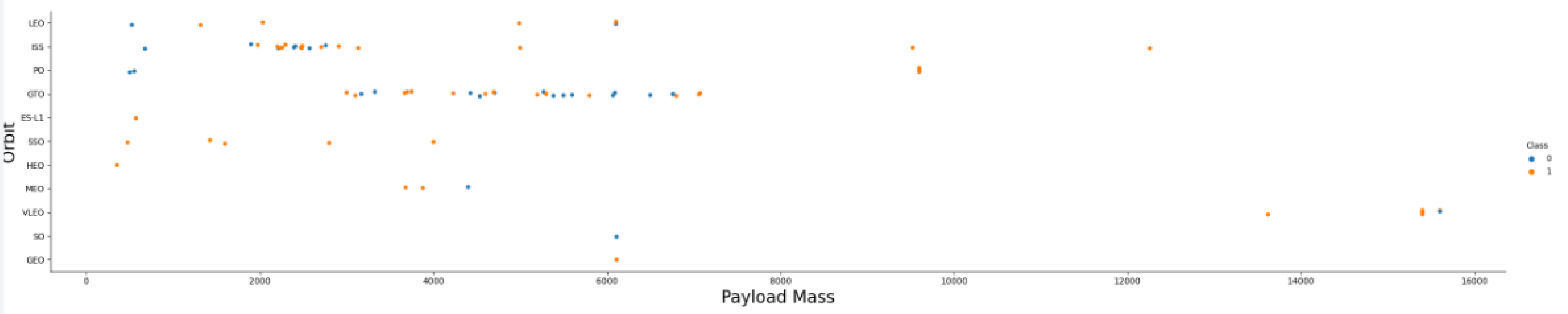


Flight Number vs. Orbit Type



For launches in the LEO orbit, as the flight number increases, the number of successes appears to increase, excluding flights in orbit GTO. Notice orbit SO has only one flight record which was a failure, and orbits ES-L1, GEO, HEO, and SSO have all records being successes (which makes sense from our bar graph in the previous slide).

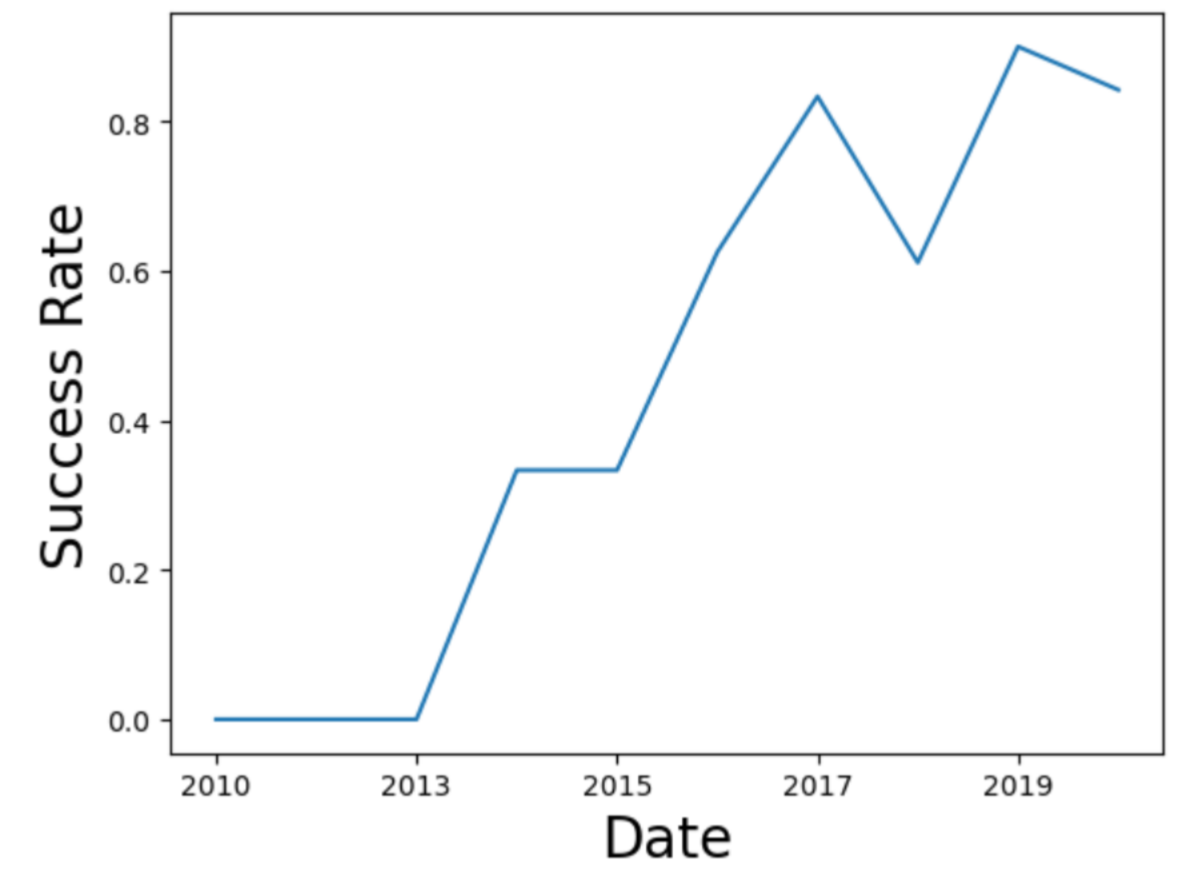
Payload vs. Orbit Type



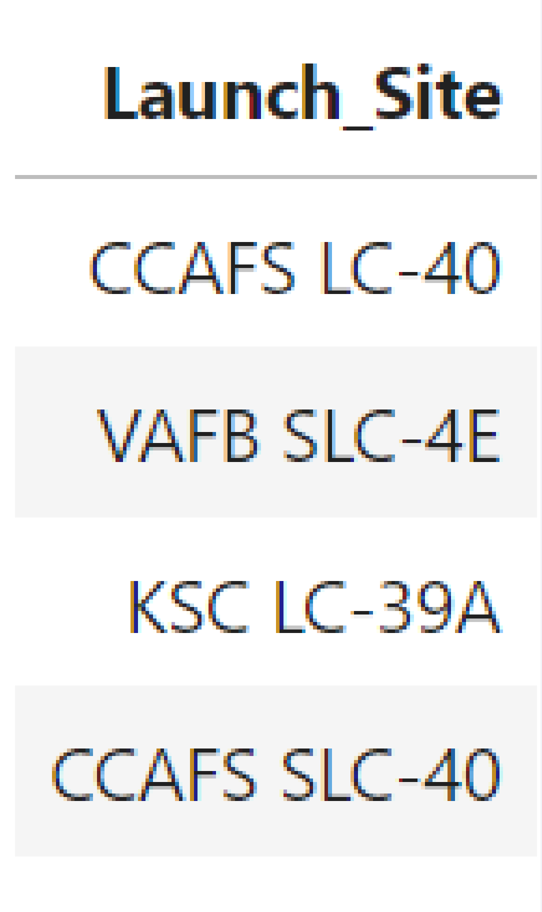
For launches with heavier payloads, there is a higher success rate for the orbits Polar, LEO and ISS. For orbits ES-L1, SSO, and HEO, there is a perfect success rate regardless of payload mass. However, for orbit GTO, payload mass does not seem to be a good feature to predict success as there are many successes and failures scattered in the range of payload mass.

Launch Success Yearly Trend

As time progresses, the success rate generally increases, with a flatline period between 2013 and 2014, and a small dip in success rate between 2017 to 2018 and 2019 to 2020.



All Launch Site Names

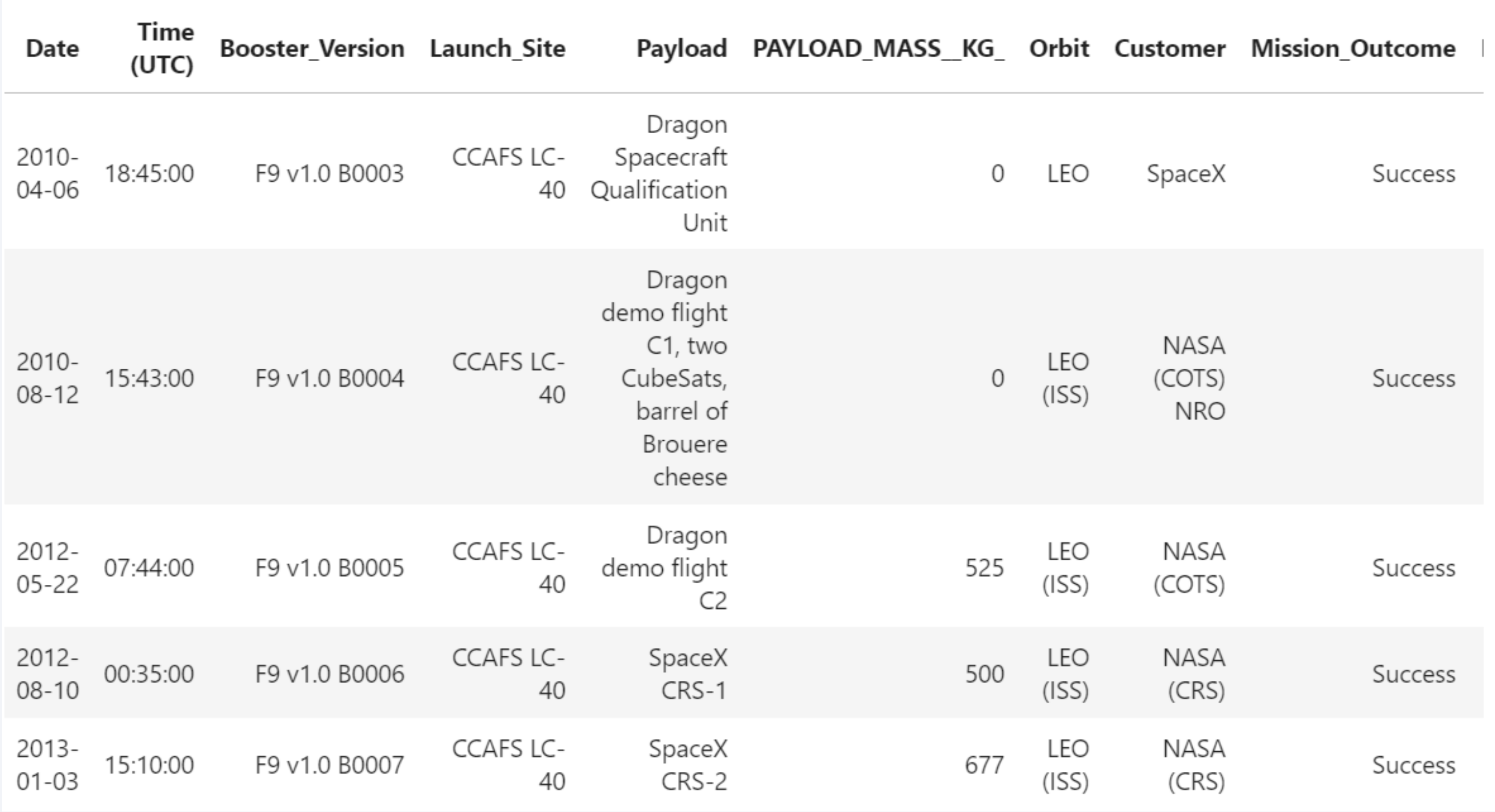


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Query:
%sql SELECT DISTINCT (Launch_Site) FROM SPACEXTABLE;

Retrieves all the unique names listed under the column ‘Launch_Site’

Launch Site Names Begin with 'CCA'

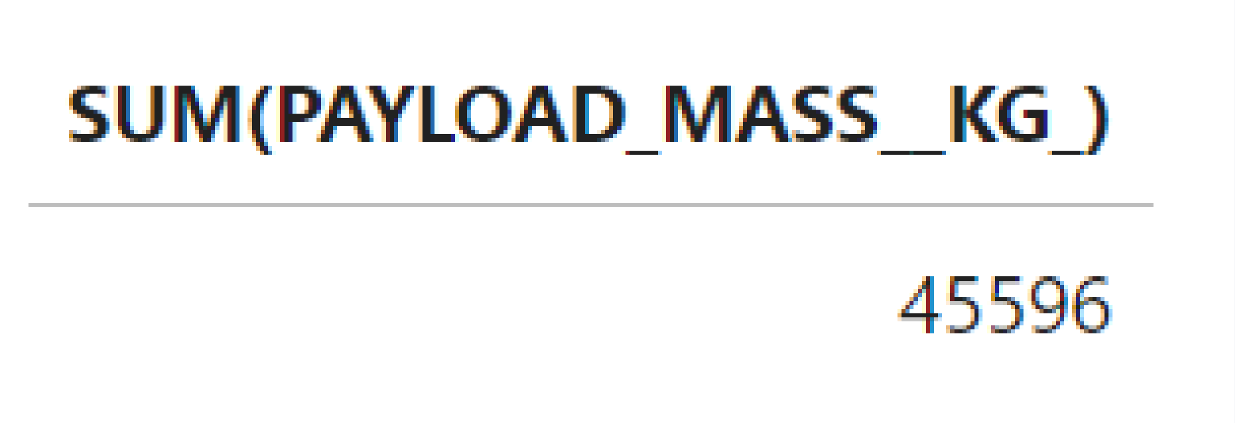


Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Query:
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

Retrieves all columns where the corresponding string in the Launch_Site column starts with ‘CCA’; Displays the first five results

Total Payload Mass

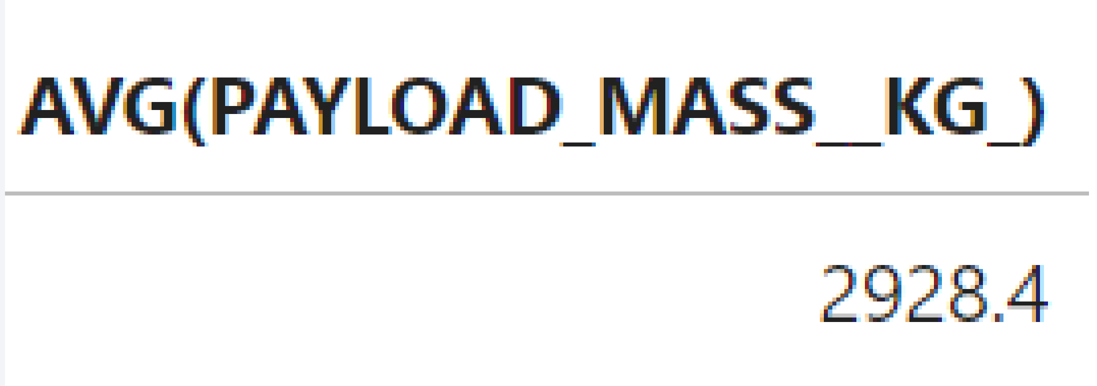


SUM(PAYLOAD_MASS__KG_)
45596

Query:
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer == 'NASA (CRS)’;

Displays the sum of all numerical values in the PAYLOAD_MASS__KG_ column where the corresponding value in the Customer column is ‘NASA (CRS)’

Average Payload Mass by F9 v1.1

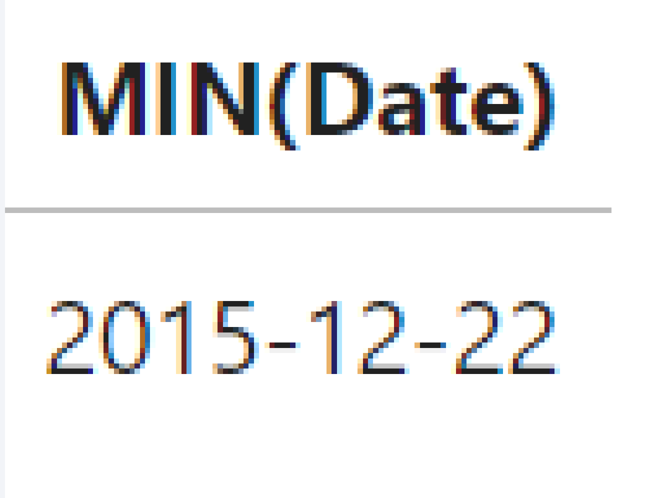


AVG(PAYLOAD_MASS__KG_)
2928.4

Query:
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1’;

Displays the average value of the PAYLOAD_MASS__KG_ column as long as the corresponding string in the Booster_Version column is ‘F9 v1.1’

First Successful Ground Landing Date

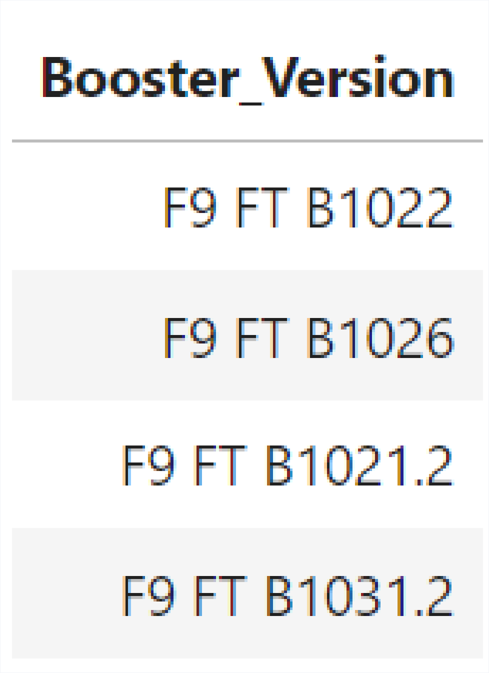


MIN(Date)
2015-12-22

Query:
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)’

Displays the earliest date in the ‘Date’ column where the corresponding string in the Landing_Outcome column is ‘Success (ground pad)’

Successful Drone Ship Landing with Payload between 4000 and 6000

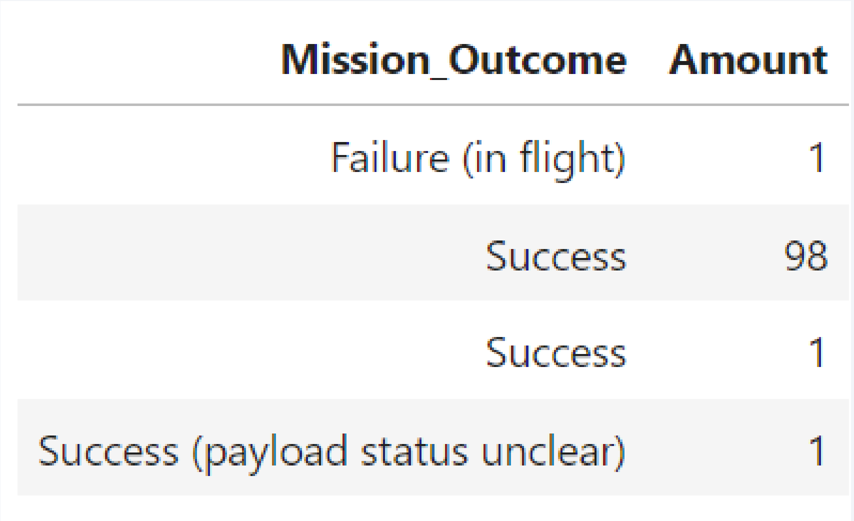


Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Query:
%sql SELECT Booster_Version FROM
SPACEXTABLE WHERE
Landing_Outcome = 'Success (drone ship)'
AND (PAYLOAD_MASS__KG_> 4000
AND PAYLOAD_MASS__KG_< 6000);

Displays the value in the Booster_Version column where the corresponding string in the Landing_Outcome column is ‘Success (drone ship)’, and the corresponding value in the PAYLOAD_MASS__KG_ column is in between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

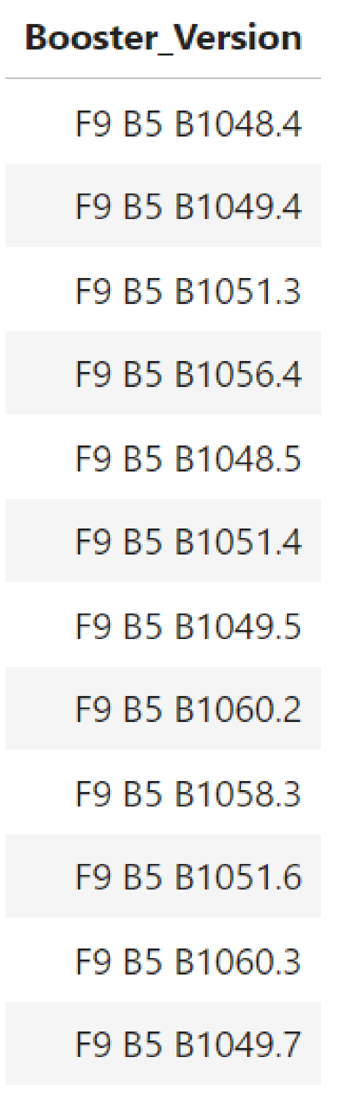


Mission_Outcome	Amount
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Query:
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) as 'Amount' FROM SPACEXTABLE GROUP BY Mission_Outcome;

Displays a sub table displaying the number of times each Mission_Outcome appears

Boosters Carried Maximum Payload

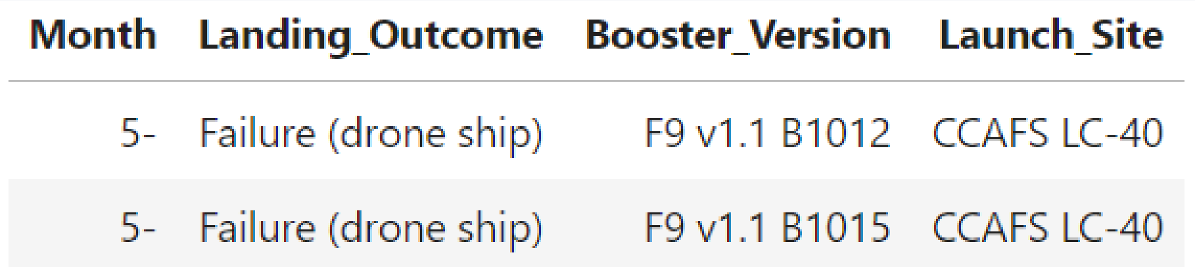


Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

Query:
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);

Displays the Booster_Version for each launch that had the maximum payload mass (when compared to all other launches, not the max capacity)

2015 Launch Records

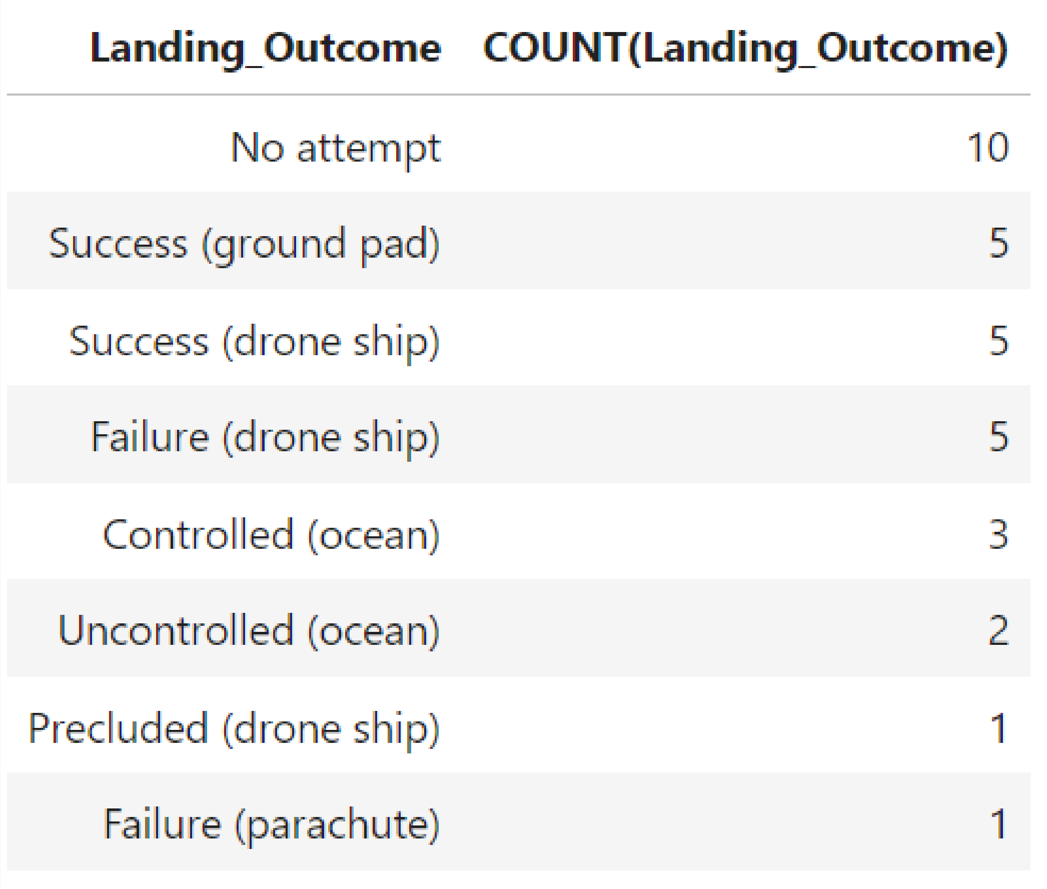


Month	Landing_Outcome	Booster_Version	Launch_Site
5-	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
5-	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Query:
%sql SELECT substr(Date, 4, 2) as 'Month' , Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE DATE Like '2015%' AND Landing_Outcome = 'Failure (drone ship)’;

Displays information about month of date, booster version, and launch site for launches that happened in 2015 and the landing outcome was a failure by drone ship

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



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

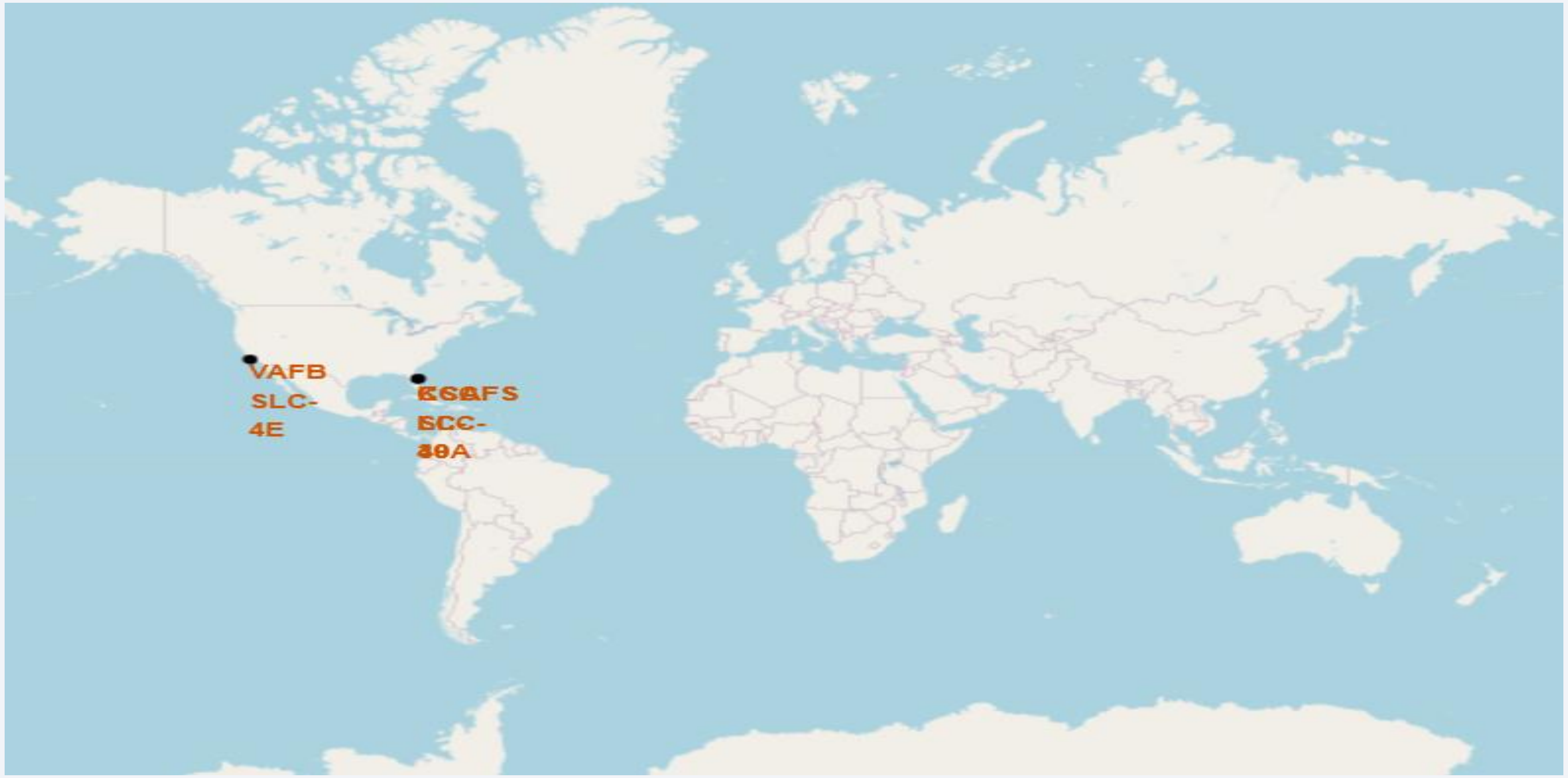
Query:
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE Date > '2010-06-04' AND Date < '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC;

Displays the total count for each possible landing outcome for launches in descending order for launches that occurred between 06/04/2010 and 03/20/2017

Section 3

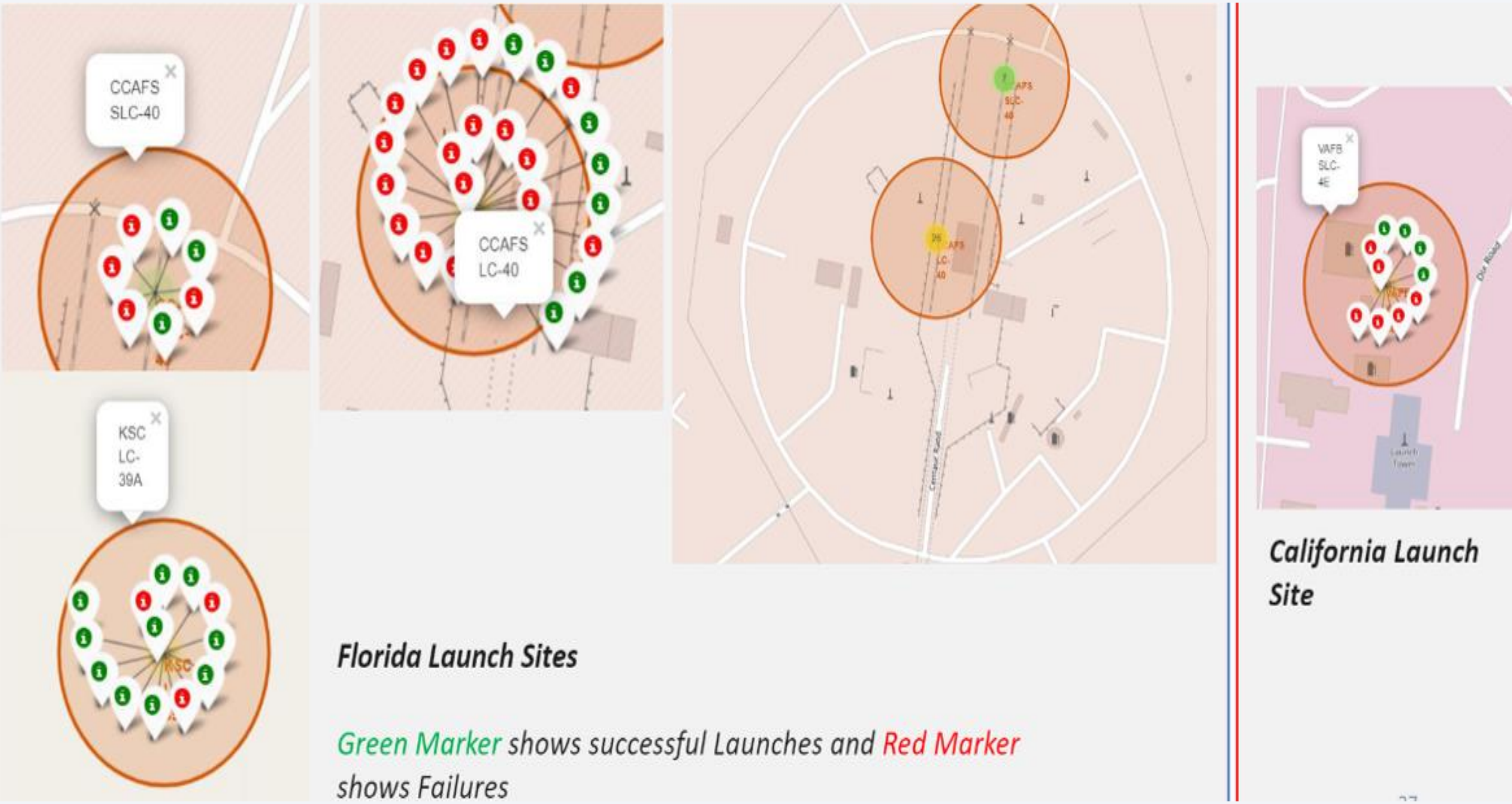
Launch Sites Proximities Analysis

Launch Site Locations



We can see the launch sites are most notably located near the west and east coast of the US.

Individual Launch Locations & Outcomes From Site



Florida Launch Sites

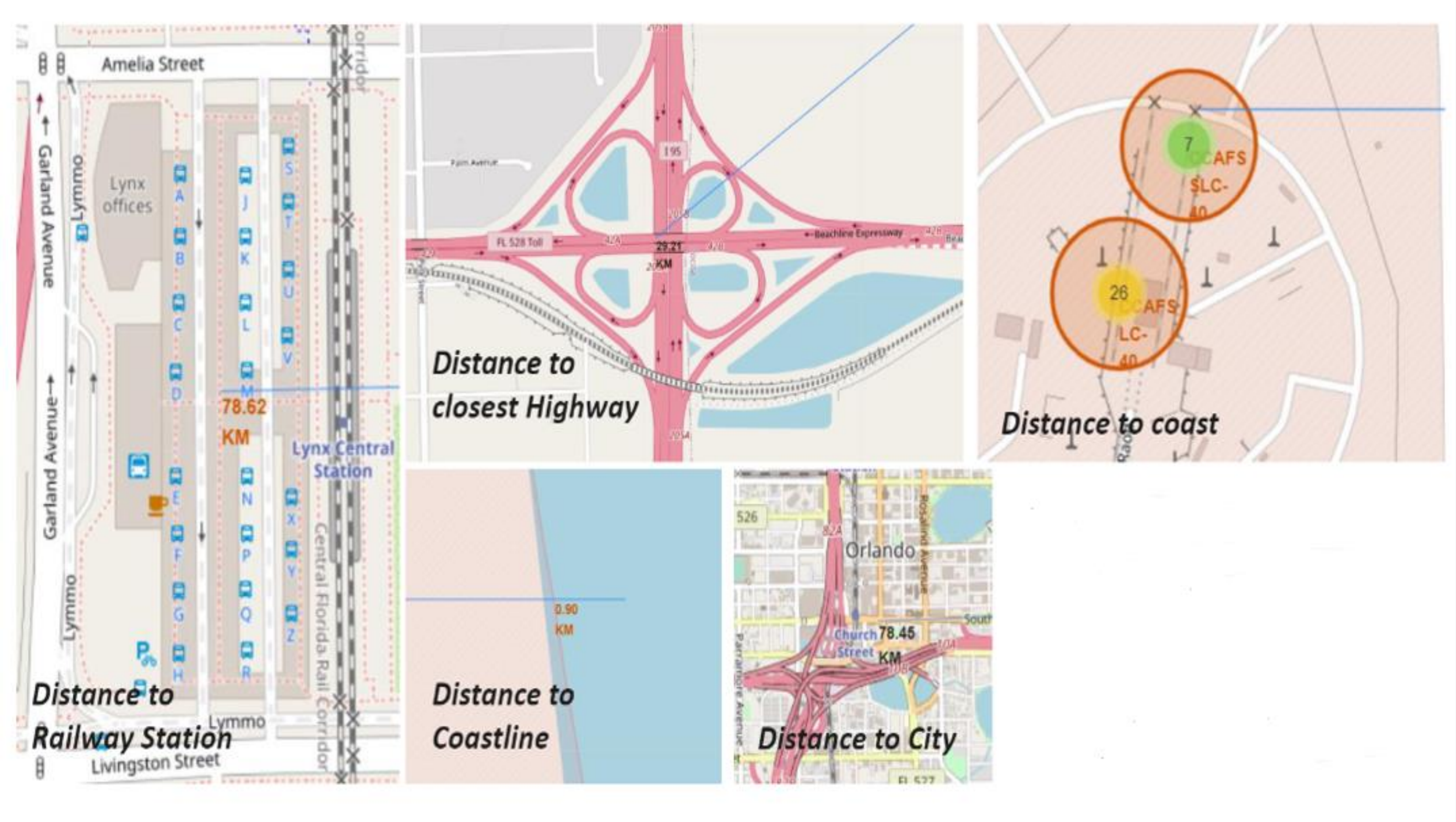
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

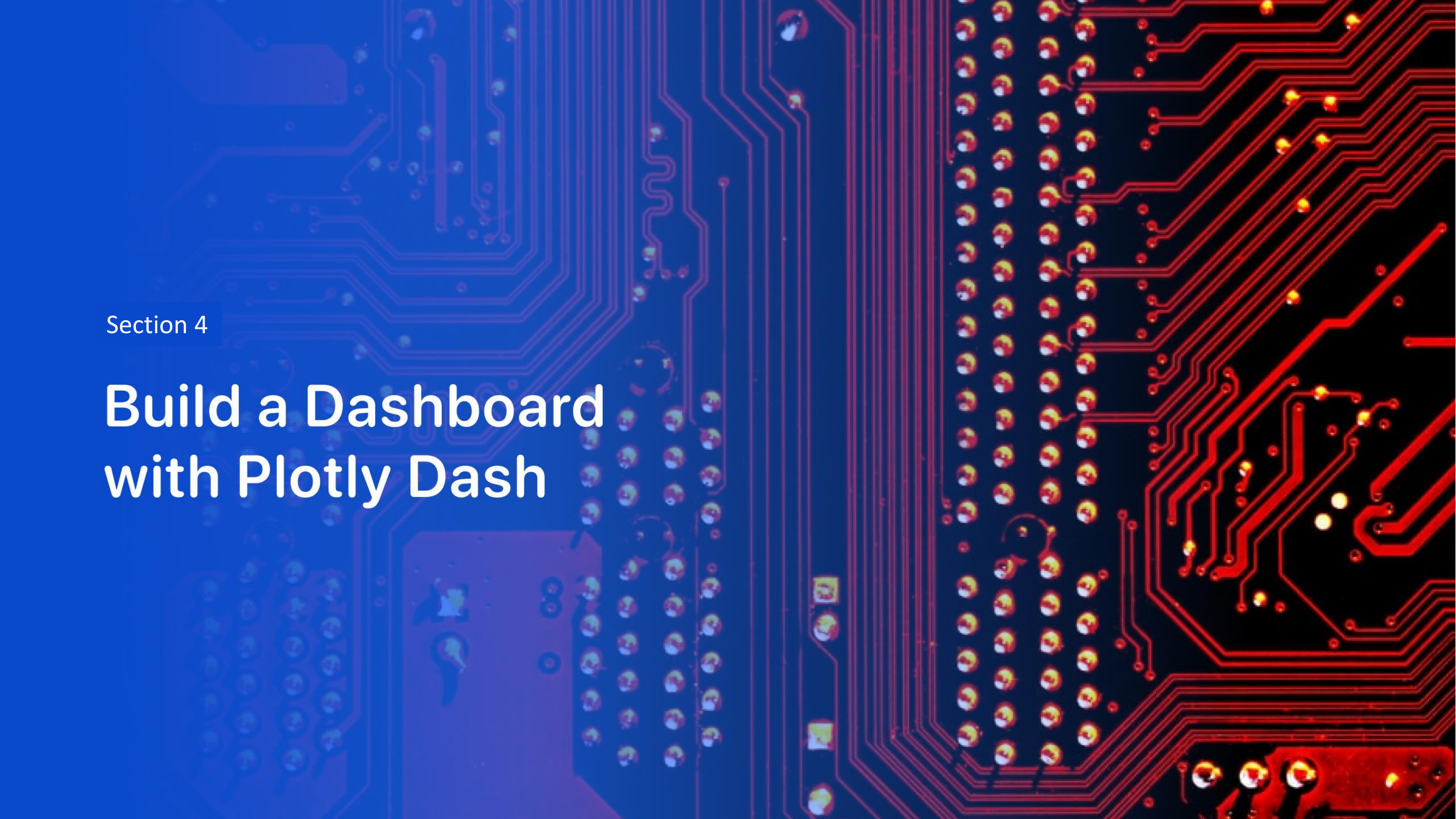
Here we see that most of the launches happen at one of the Florida launch sites.

The site KSC LC-39A has a significantly higher rate of success compared to all other launches, which have a success rate close to 50%.

Landmarks Nearby The Launch Site CCAFS SLC-40



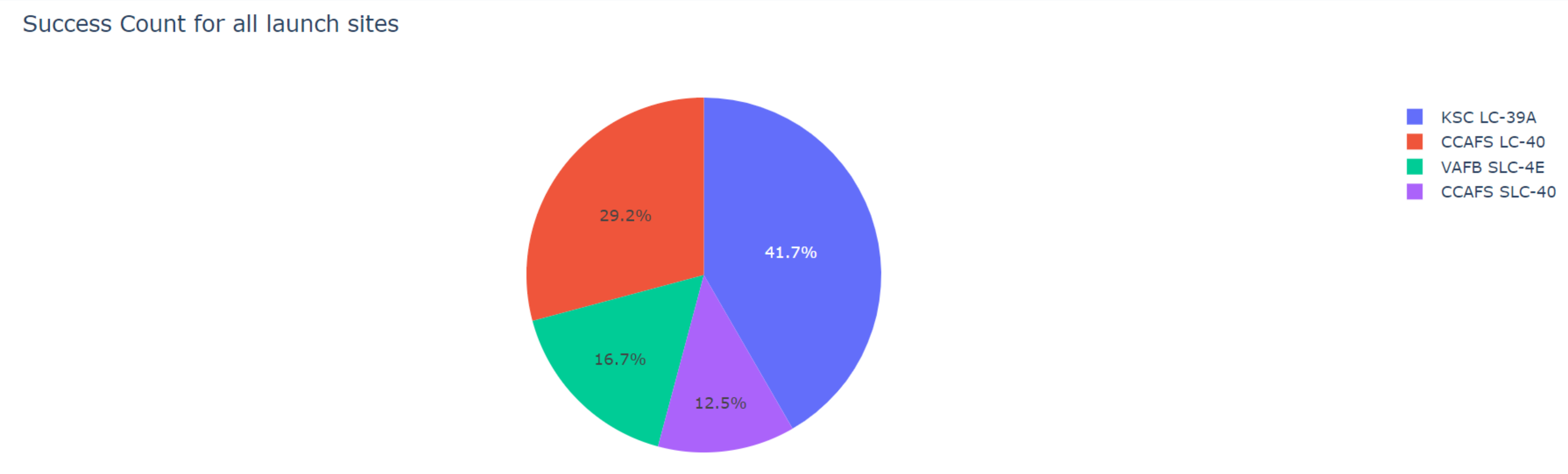
Here, we see that this launch site is very close to the coastline but is quite far away from cities. Railway stations and highways are also a bit further away, but they do not seem to influence the location of a launch site.



Section 4

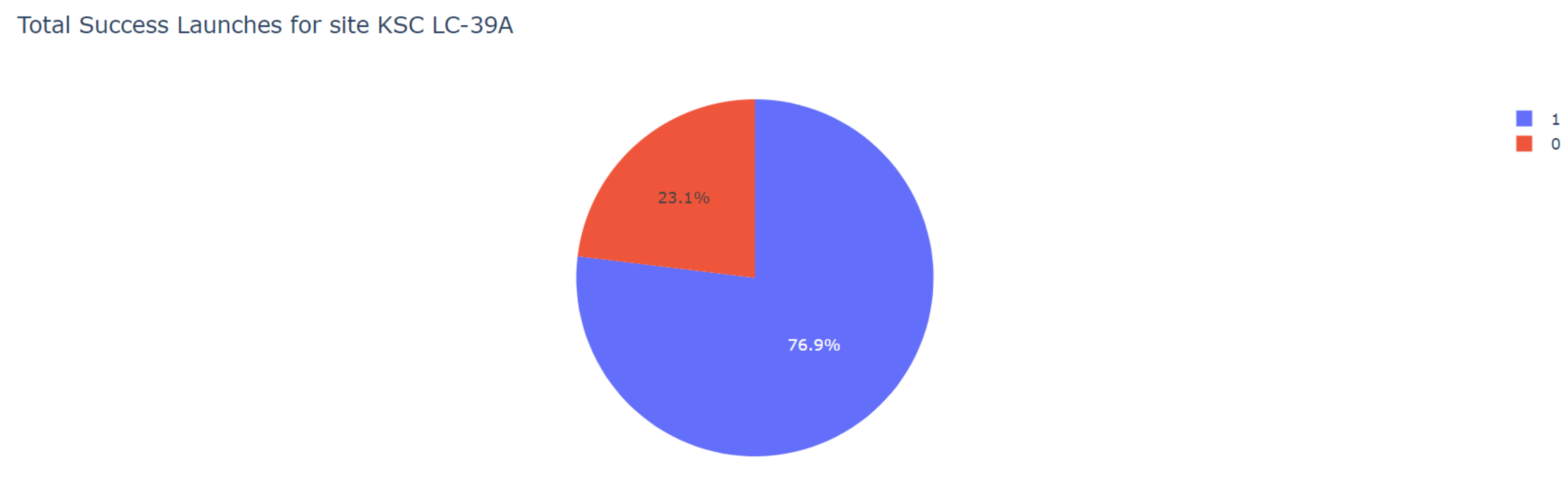
Build a Dashboard with Plotly Dash

Success Count For All Launch Sites



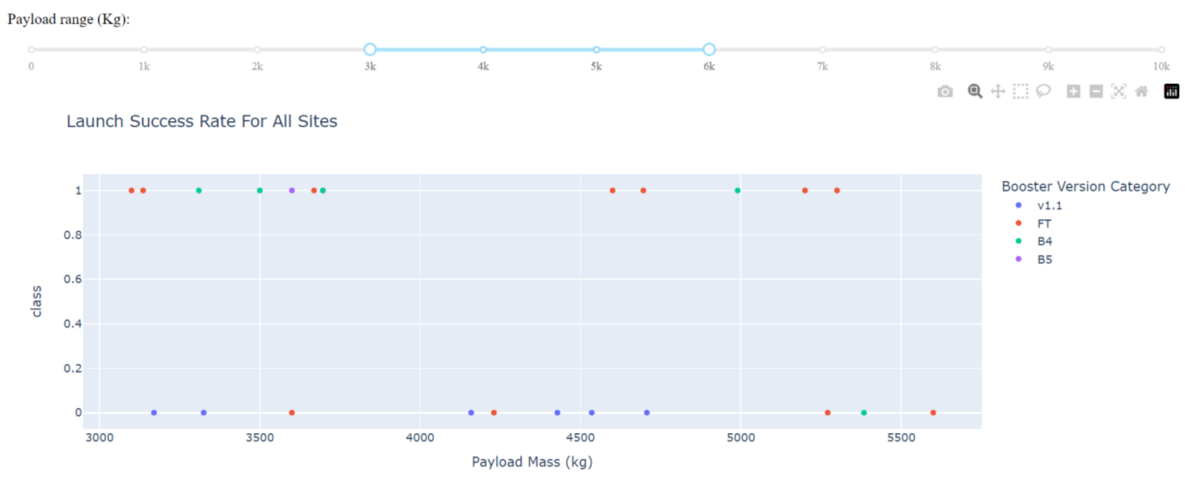
This pie chart displays the percentage of success from each launch site, with KSC-LC-39A having the highest percentage of successes.

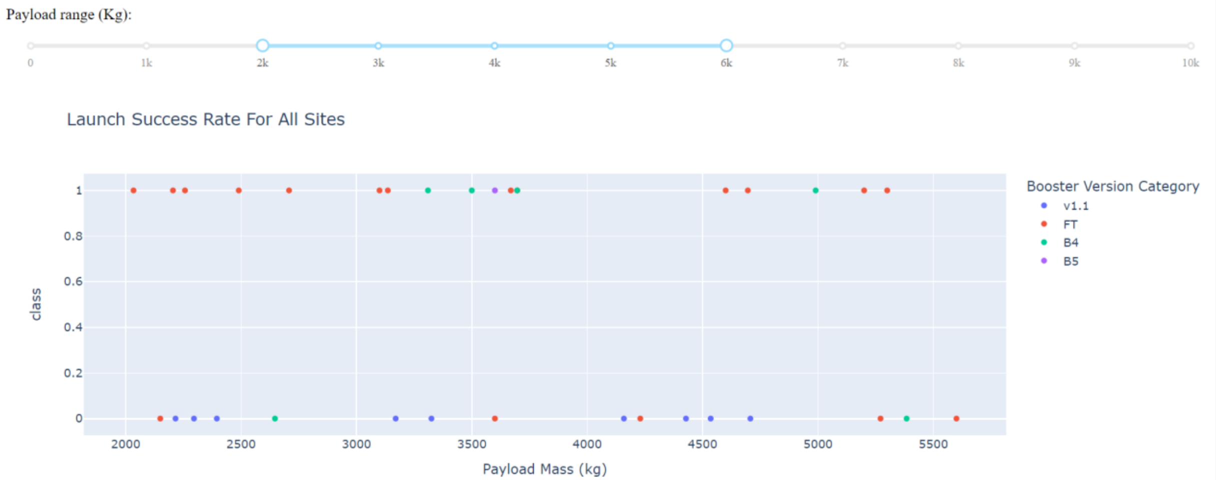
Launch Site With Highest Launch Success



This pie chart displays the success (1) ratio to failure (0) ratio, with the site KSC LC-39A having the highest ratio of successes to failures.

Payload Mass vs. Outcome



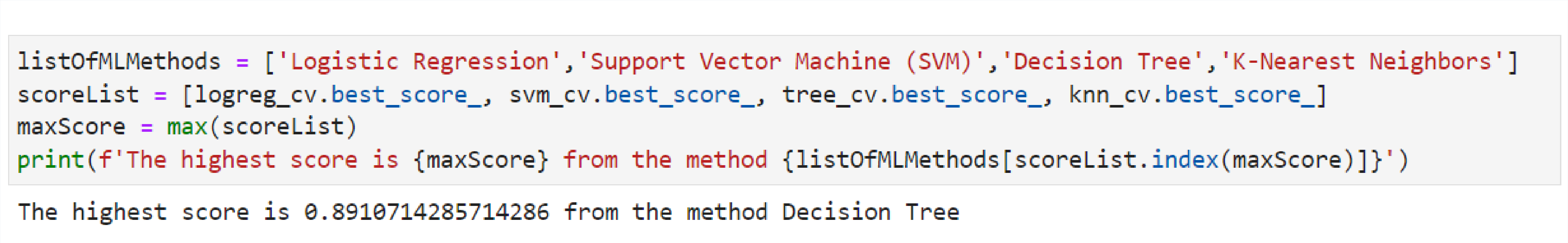


This scatterplot plots the payload mass of launches to outcome (success labeled as 1.0 and failure labeled as 0.0). We can adjust the range of the payload mass, and we can see most of the successes being in between the payload mass of 2,000 kg and 5,500 kg. Notably, there are a few more successes with launches holding lighter loads vs heavier loads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



```
listOfMLMethods = ['Logistic Regression','Support Vector Machine (SVM)','Decision Tree','K-Nearest Neighbors']
scoreList = [logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_]
maxScore = max(scoreList)
print(f'The highest score is {maxScore} from the method {listOfMLMethods[scoreList.index(maxScore)]}')
```

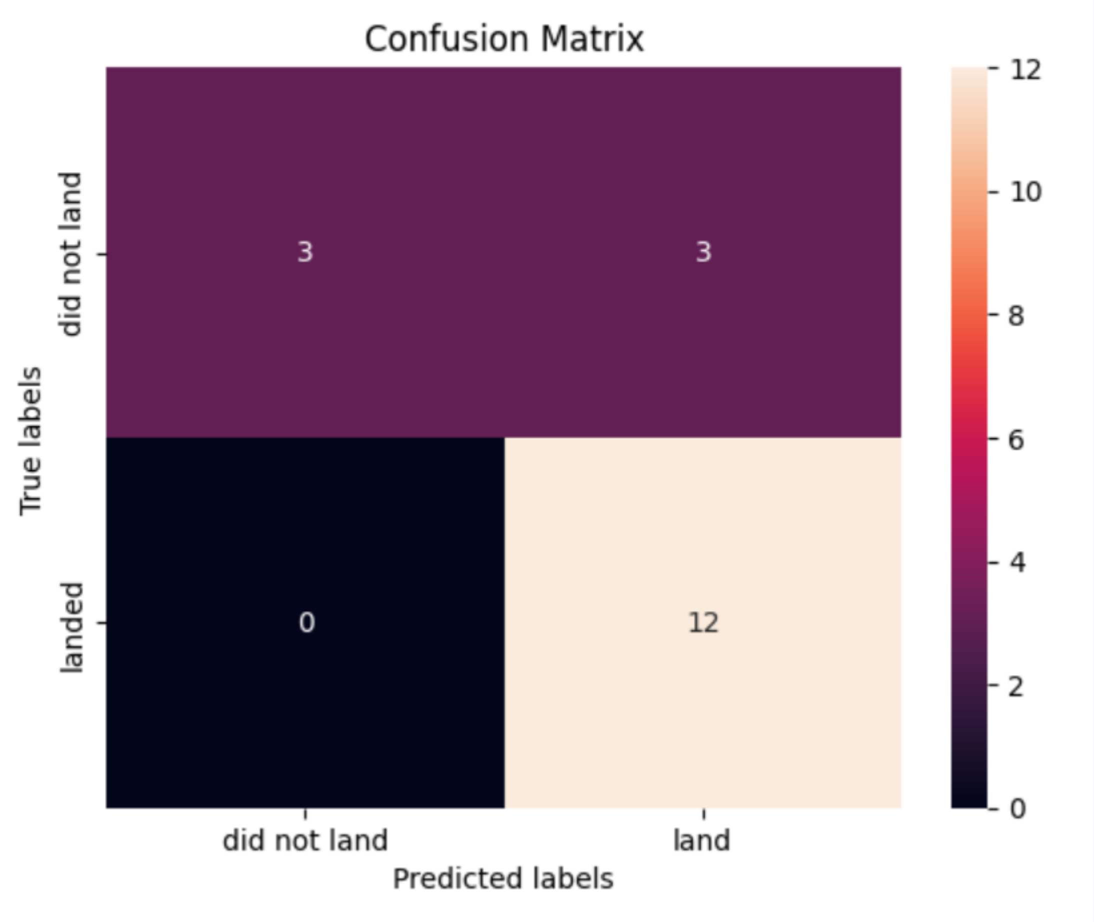
```
The highest score is 0.8910714285714286 from the method Decision Tree
```

From the four classification algorithms used on our dataset, we see that the decision tree appears to have the highest accuracy when using our test data. While the models were all tested with the initial training dataset, they use cross validation for finding the best parameters for each model alongside the best score, so they are still accurate and was technically using out of sample data due to how cross validation works.

Confusion Matrix For Decision Tree

Here, we see the decision tree model was able to correctly classify most of the data in our test data set (which is what the confusion matrix is based of). However, it incorrectly classified 3 launches as successes (they did land), when they actually did not land (failure), so it is obviously not perfect.

Note that for all confusion matrices generated from the four models, they were all identical with these results.



Conclusions

- The greater the flight number, the higher the success rate was for that launch
- Launch success rate overall began increasing from 2013 up to 2020
- Launches set to orbits ES-L1, GEO, HEO, SSO, VLEO had the largest success rate
- KSC LC-39A had the most successful launches of any sites, with it aimed at some of the orbits mentioned in the previous observation
- The decision tree was the most accurate model for classifying launch records as being successful or not; However, the other models that were trained had a very similar accuracy score compared to the decision tree, so they are still valid for future observation and testing.

Thank you!