IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kevin Steve Sathyanath
06/02/2023

Outline

Executive Summary

This project attempted to predict the future landing success of the first stage of the Falcon 9 rocket.

- Summary of methodologies
  1. Data Collection
  2. Data wrangling
  3. EDA with data visualization
  4. EDA with SQL
  5. Interactive map with Folium
  6. Dashboard with Plotly Dash
  7. Predictive Analysis

- Summary of results
  1. Exploratory data analysis results
  2. Interactive analysis results
  3. Predictive analysis results

Introduction

Project background and context:

The Falcon 9 is well known to save money when putting payloads into space. Much of this is done by cutting costs by reusing the first stage. If we can predict whether the first stage will land, we can ostensibly determine the price of the launch. This project aims to make exactly such a prediction using ML.

Problems you want to find answers to:

-What variables affect the chance of the rocket landing successfully?

-How likely is it that the first stage will be successful?

-How to raise the chances of this success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected using SpaceX Rest API and scraped from Wikipedia
- Perform data wrangling
    - One hot encoding was applied; less useful columns were dropped.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

As mentioned earlier, data was collected in 2 ways.

SpaceX API: The API gives all the data needed, including info on the rocket used, payload delivered, landing outcome etc. The get request is decoded into a pandas dataframe using .json_normalize(). Then it is cleaned.

Wikipedia: Data is scraped using HTML's BeautifulSoup.

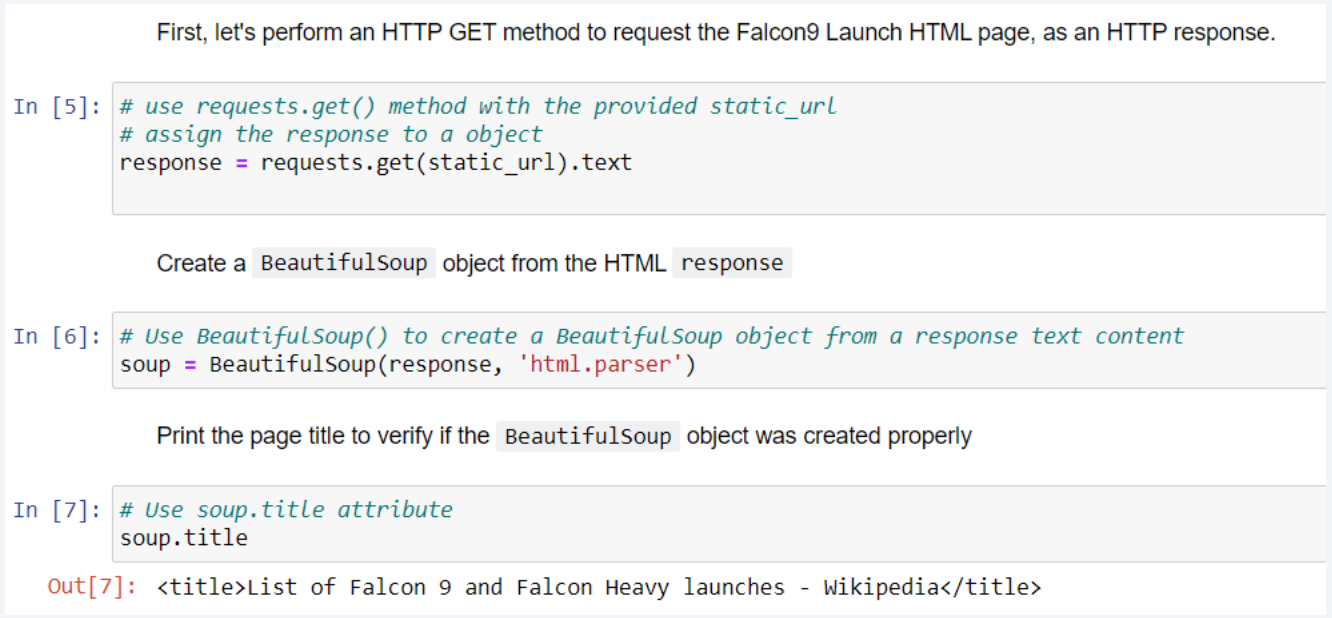
Data Collection – SpaceX API

- My URL is: https://github.com/kvnstv1/Capstone-/blob/56cfb941c4dca3dcf088375416f31daf92c96afb/REST%20API.ipynb

1. First we use the Get request for the launch data using the API.
2. Next we use the json_normalize() method to convert the json output to a dataframe.
3. Finally, we performed data cleaning and filled in the missing values.

Data Collection - Scraping

- Using BeautifulSoup for scraping data from a Wikipedia page, we then parse the table and convert it into a pandas dataframe.
- The URL is: https://github.com/kvnstv1/Capstone-/blob/17b6205d4399c7b8d04e136a7706f90f9f72e629/Web%20scraping.ipynb



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

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

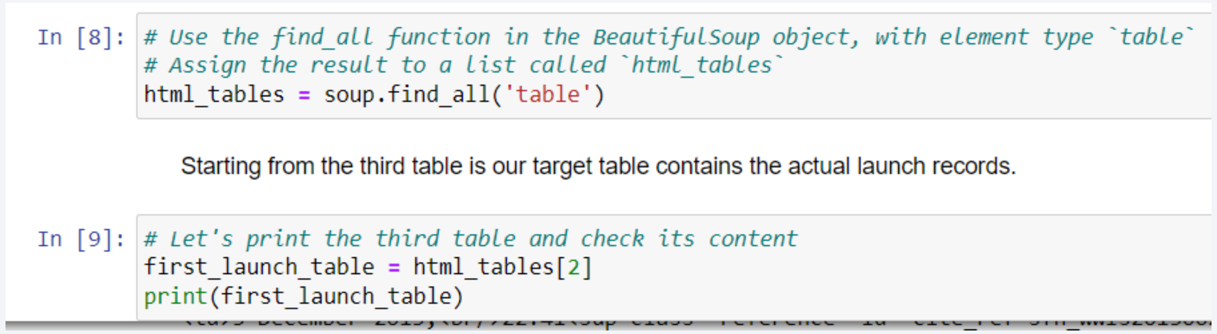
Create a `BeautifulSoup` object from the HTML `response`

```
In [6]: # 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]: # Use soup.title attribute
        soup.title

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



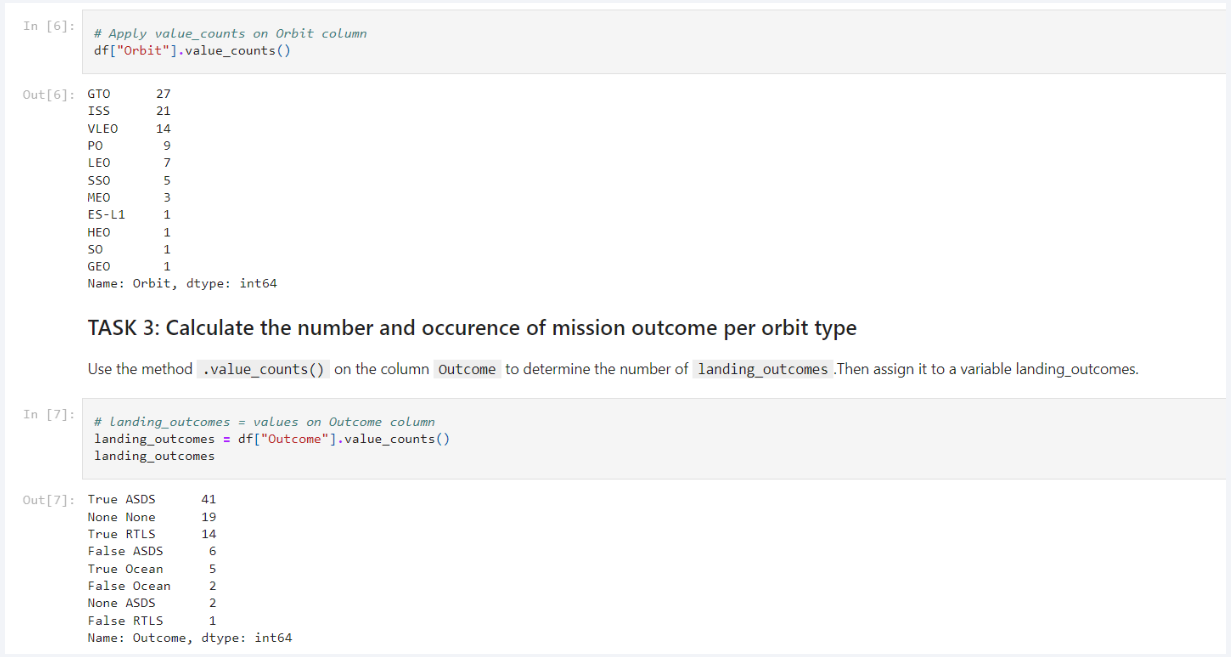
```
In [8]: # Use the find_all function in the BeautifulSoup object, with element type `table`
        # Assign the result to a list called `html_tables`
        html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
In [9]: # Let's print the third table and check its content
        first_launch_table = html_tables[2]
        print(first_launch_table)
```

Data Wrangling

- Data analysis was done by calculating the number of launches, calculating the number and occurrence of each orbit, the number of missions of each type and creating a landing outcome label.
- My github URL is: https://github.com/kvnstv1/Capstone-/blob/381625c51016fc16689931cba1dd6e78545f9984/Data%20wrangling%20(2).ipynb



```
In [6]:  # Apply value_counts on Orbit column
         df["Orbit"].value_counts()

Out[6]:  GTO      27
         ISS      21
         VLEO     14
         PO        9
         LEO       7
         SSO       5
         MEO       3
         ES-L1     1
         HEO       1
         SO        1
         GEO       1
         Name: Orbit, dtype: int64
```

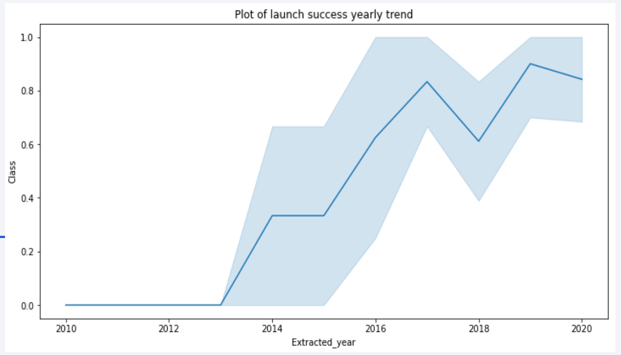
TASK 3: Calculate the number and occurence of mission outcome per orbit type

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

```
In [7]:  # landing_outcomes = values on Outcome column
         landing_outcomes = df["Outcome"].value_counts()
         landing_outcomes

Out[7]:  True ASDS      41
         None None      19
         True RTLS      14
         False ASDS      6
         True Ocean      5
         False Ocean     2
         None ASDS       2
         False RTLS      1
         Name: Outcome, dtype: int64
```

EDA with Data Visualization



- I used 3 different types of charts.
  - Scatter graphs were used because they are very good to show how much one variable is affected by another i.e. correlation.
  - I used bar graphs because I can compare sets of data at a glance, as stated in the question prompt.
  - I used a line graph because it can map out trends over time and can come in very handy for making predictions. This is part of the state goal of the exercise.
- *URL: https://github.com/kvnstv1/Capstone-/blob/8c250db3b61ecd946a57764cc7427deec65a4b95/Data%20visualization%20(1).ipynb*

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Connect to the database
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA’
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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
- *https://github.com/kvnstv1/Capstone-/blob/e73df4cfe33eebea9453b8fe6607dbfac4b2f263/EDA%20with%20SQL.ipynb*

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map. Explain why you added those objects.

1. I marked all the launch sites on a map using latitude and longitude. I made a circle marker around each launch site with a label of the name of the site as well as the number of of occurrences.

2. I marked the successful or failed launches on a map.

The link is: https://github.com/kvnstv1/Capstone-/blob/072917ace14dce378424292bddf5afcc5ea4f859/lab_jupyter_launch_site_location%20(1).ipynb

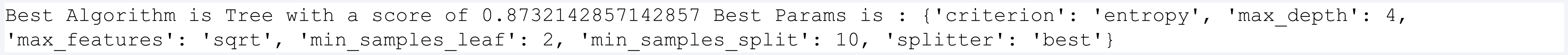
Build a Dashboard with Plotly Dash

- Made an interactive dashboard in python and plotted pie charts showing total launches by select sites
- URL: https://github.com/kvnstv1/Capstone-/blob/ffdbbb5200a4bffd5e197591b48cec021a9c8a8c/SpacexApp.py

Predictive Analysis (Classification)

- Loading the data using numpy and pandas, we transformed it and split our data into training and testing.
- We tested different machine learning algorithms presented during the course.
- Using accuracy as a metric, we improved available models.
- We found the best performing classification
- URL below:
- https://github.com/kvnstv1/Capstone-/blob/b376eab7dfa59d5c054e8797ec5ff31209228588/SpaceX_Machine%20Learning%20Prediction_Part_5%20(2).ipynb

Results



```
Best Algorithm is Tree with a score of 0.8732142857142857 Best Params is : {'criterion': 'entropy', 'max_depth': 4,
'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'best'}
```

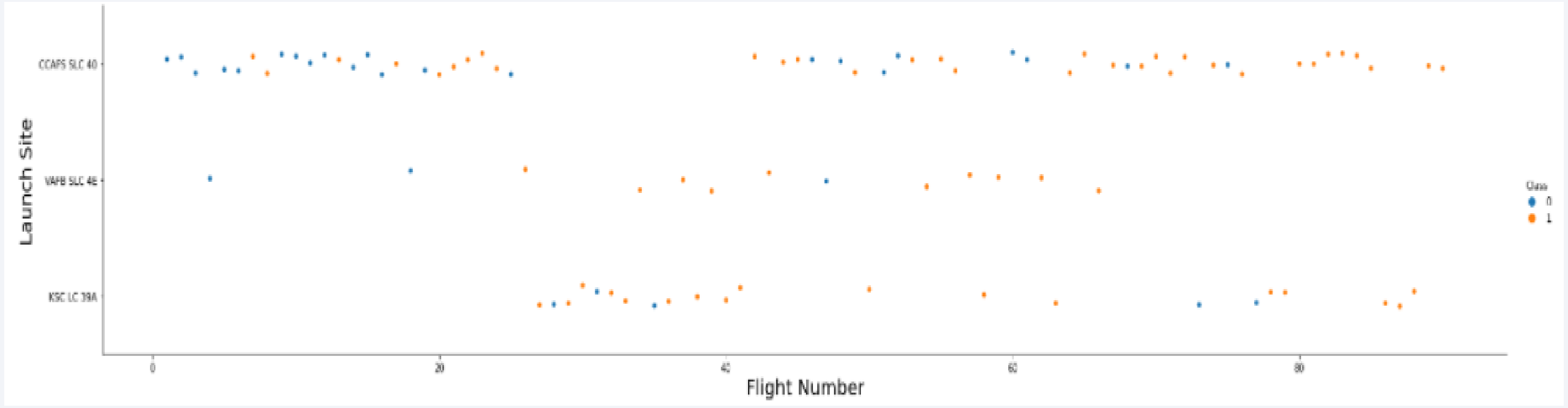
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

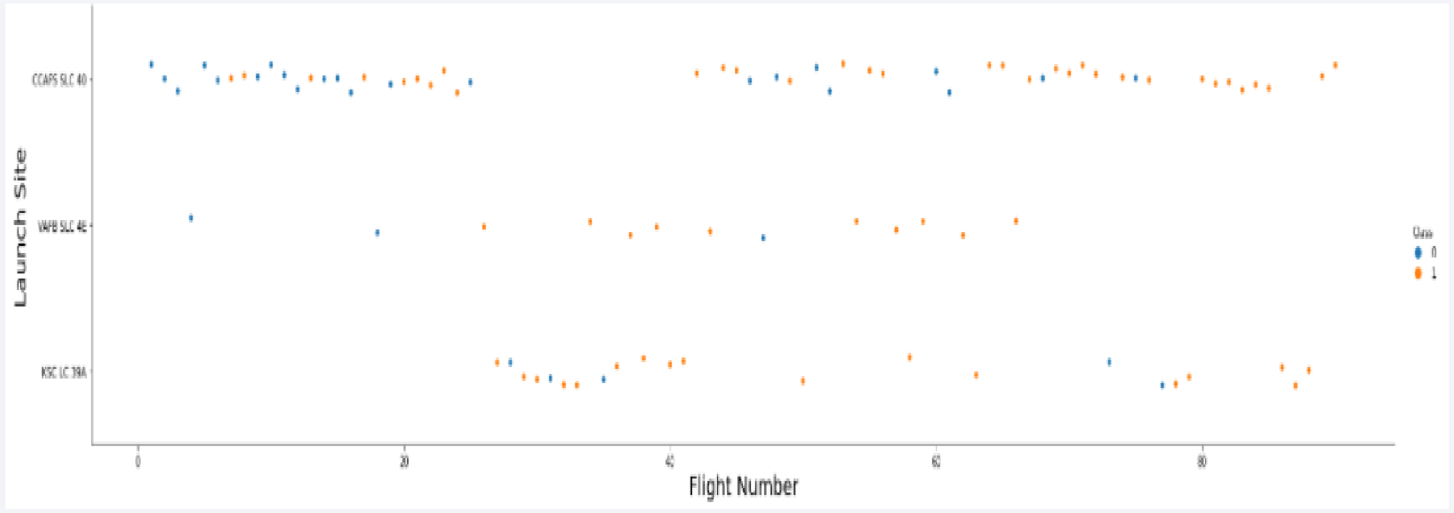
Flight Number vs. Launch Site

- From the plot, we a direct correlation between flight amount at a launch site and the success rate at a launch site.



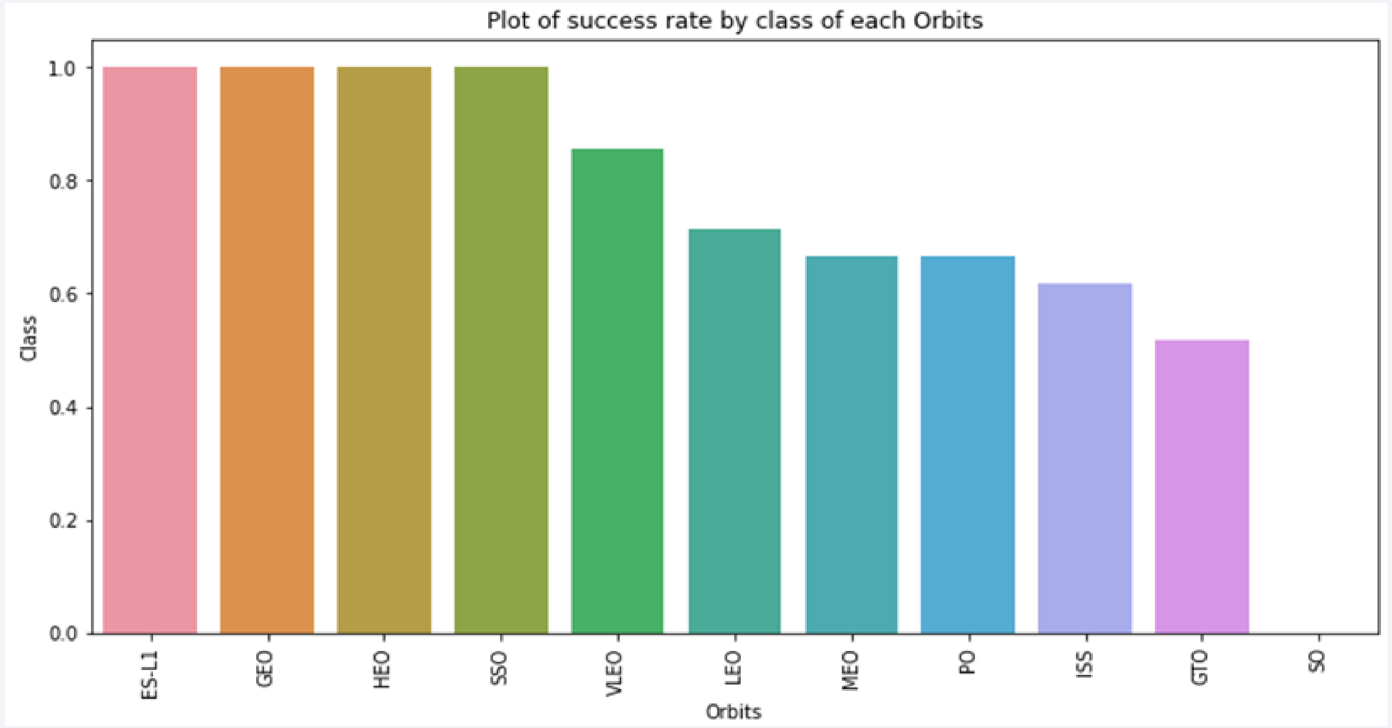
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40, the higher the success rate.



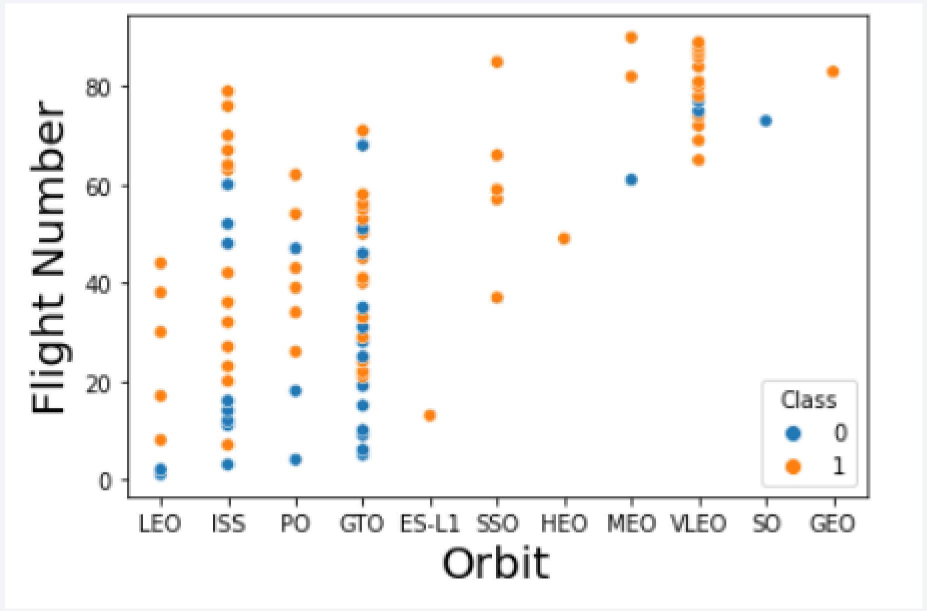
Success Rate vs. Orbit Type

Orbit GEO, HEO, SSO, ES-L1 has the best success rate.



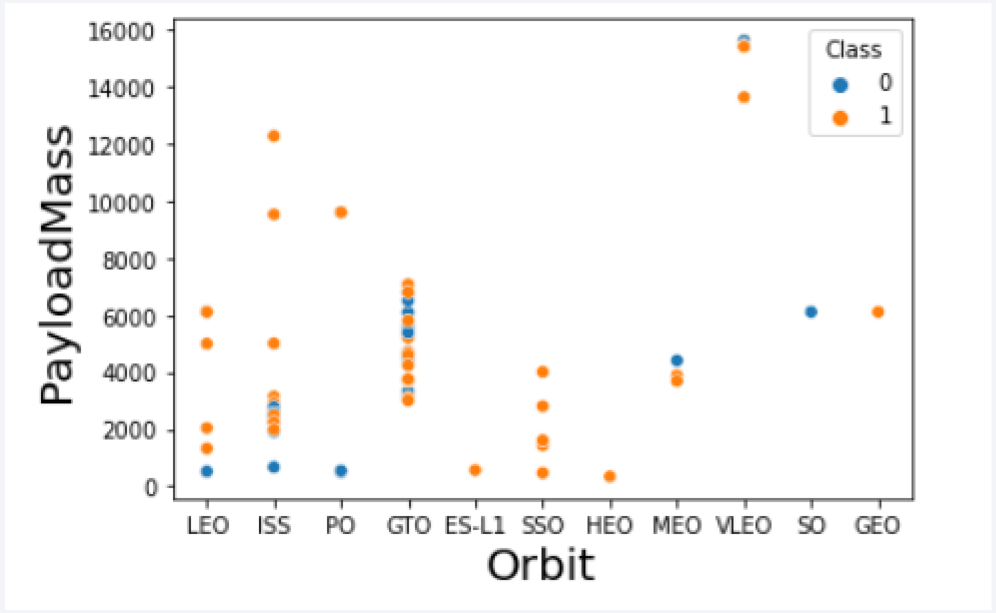
Flight Number vs. Orbit Type

Evidently, it can be seen 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.



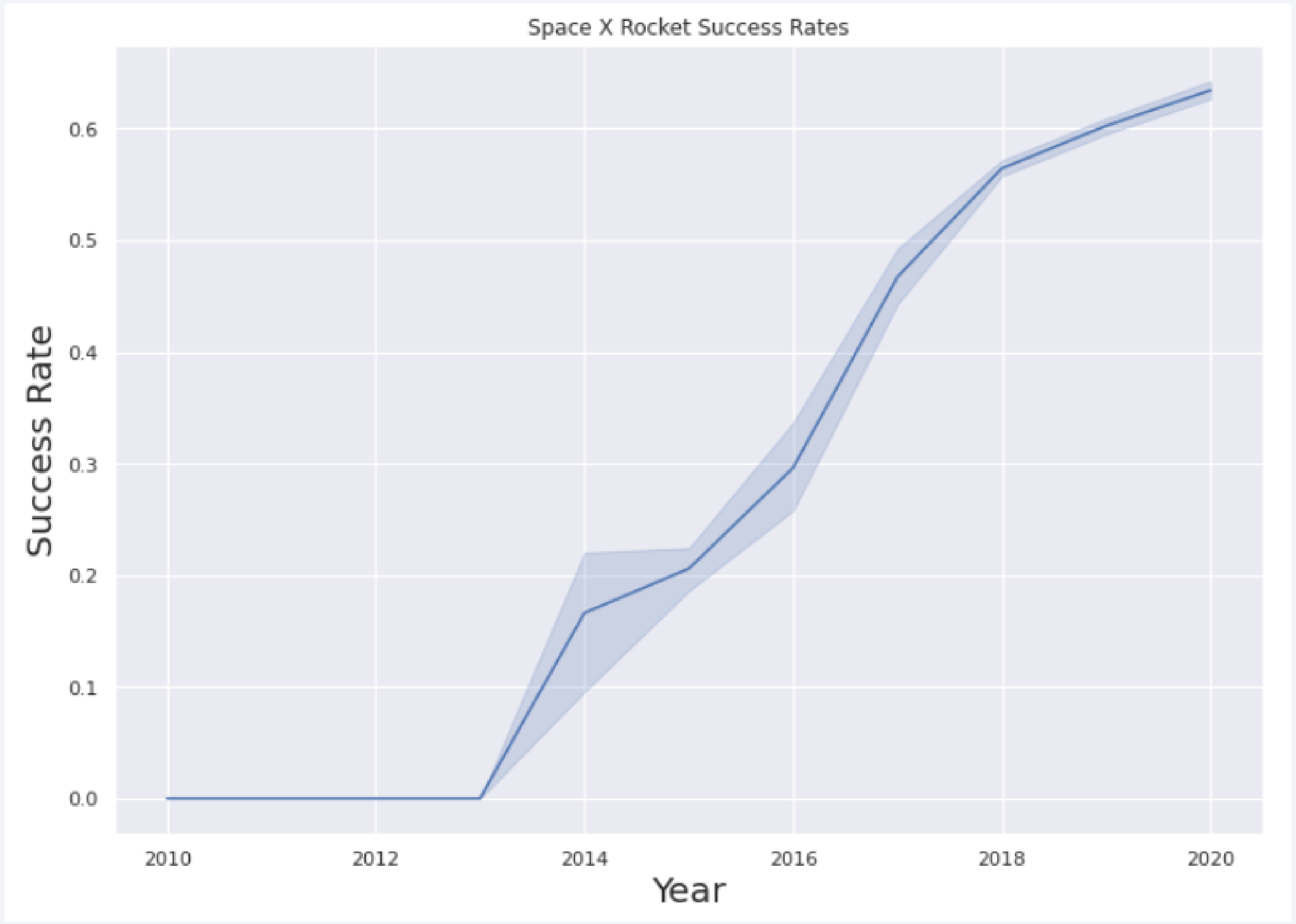
Payload vs. Orbit Type

Heavy payloads have a negative influence on GTO orbits and positive influence on GTO and polar LEO orbits.



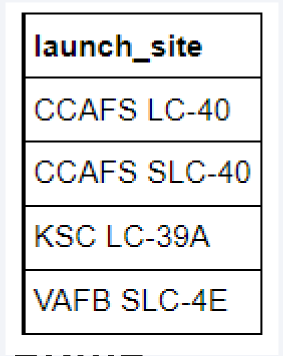
Launch Success Yearly Trend

The success rate since 2013 has been increasing!



All Launch Site Names

- *Find the names of the unique launch sites*
- %sql select Unique(LAUNCH_SITE) from SPACEXDATASET;

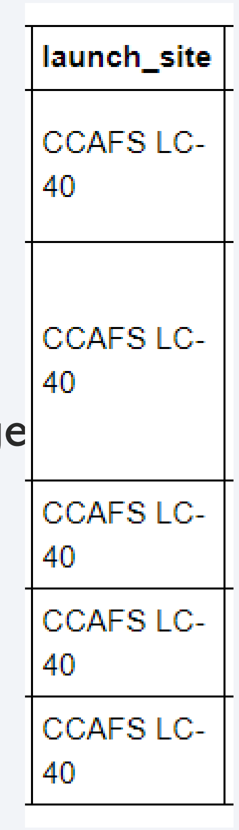


launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

- *Present your query result with a short explanation here*
- Using the word DISTINCT in the query means that it will only unique in the LAUNCH_SITE column from SPACEXTBL. I used [illegible] network lab because DB2 **REFUSED** to work.

Launch Site Names Begin with 'CCA'

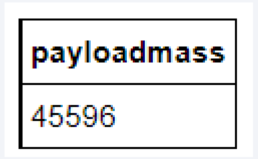
- Find 5 records where launch sites begin with `CCA`
  - %sql SELECT * from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
- Present your query result with a short explanation here
  - Using the word LIMIT 5 in the query means that it will only show 5 records from SPACEXTBL and LIKE keyword has a wild card with the words 'CCA%' the percentage the end suggests that the launch_site name must start with CCA



launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

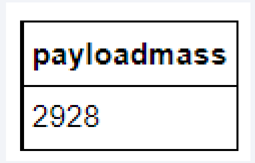
- Calculate the total payload carried by boosters from NASA
  - %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL WHERE customer = 'NASA (CRS)';
- Present your query result with a short explanation here
  - Using the function SUM summates the total in the column payload_mass_kg_. The where clause filters the dataset to only perform calculations on customer NASA (CRS)



payloadmass
45596

Average Payload Mass by F9 v1.1

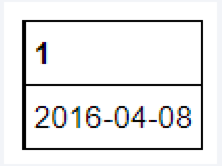
- Calculate the average payload mass carried by booster version F9 v1.1
  - %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where Booster_Version = 'F9 v1.1';
- Present your query result with a short explanation here
  - Using the function AVG works out the average in the column PAYLOAD_MASS__KG_. The where clause filters the dataset to only perform calculate on Booster_Version = 'F9 v1.1'.



payloadmass
2928

First Successful Ground Landing Date

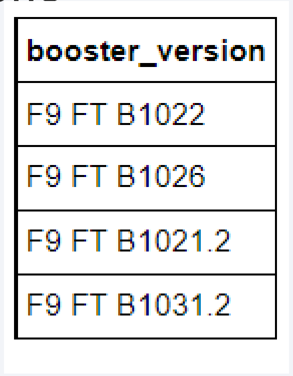
- Find the dates of the first successful landing outcome on ground pad
  - %sql select min(DATE) from SPACEXTBL where landing__outcome = 'Success (drone ship)';
- Present your query result with a short explanation here
  - Using the function MIN works out the minimum data in the column DATE, the where clause filters the dataset to only perform calculates on landing__outcome = 'Success (drone ship)'.



1
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

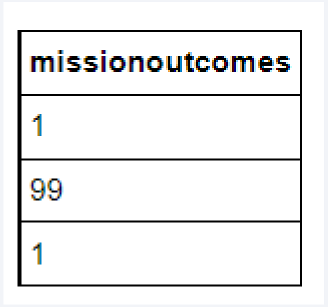
- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - %sql select Booster_Version from SPACEXTBL where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000;
- Present your query result with a short explanation here
  - Selecting only Booster_Version, the where filters the dateset to landing__outcome = 'Success (drone ship)', the AND clause specifies additional filter conditions payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000.



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

Total Number of Successful and Failure Mission Outcomes

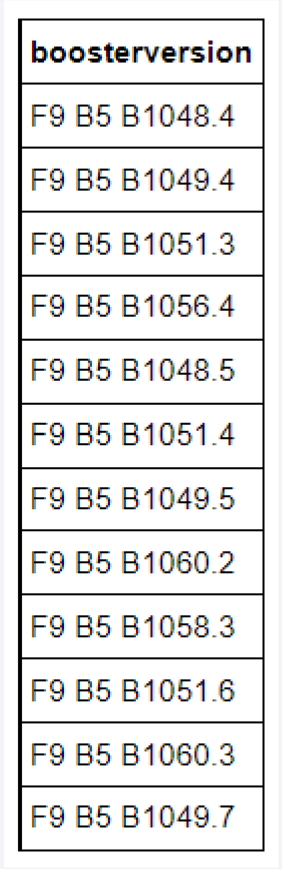
- Calculate the total number of successful and failure mission outcomes
  - %sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
- Present your query result with a short explanation here
  - Count shows the total number, group by shows different kinds.



missionoutcomes
1
99
1

Boosters Carried Maximum Payload

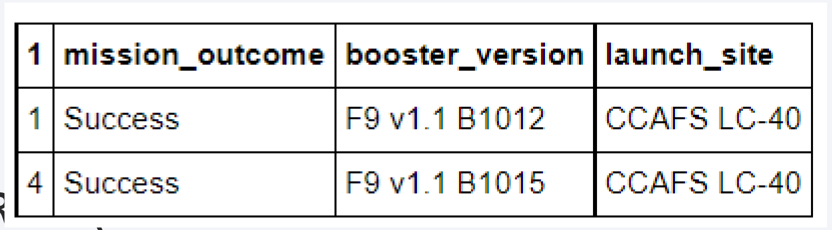
- List the names of the booster which have carried the maximum payload mass
  - %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
- Present your query result with a short explanation here
  - Subquery put the maximum payload mass, select BOOSTER_VERSION from SPACEXTBL, and where evaluates PAYLOAD_MASS__KG_= max value.



boosterversion
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

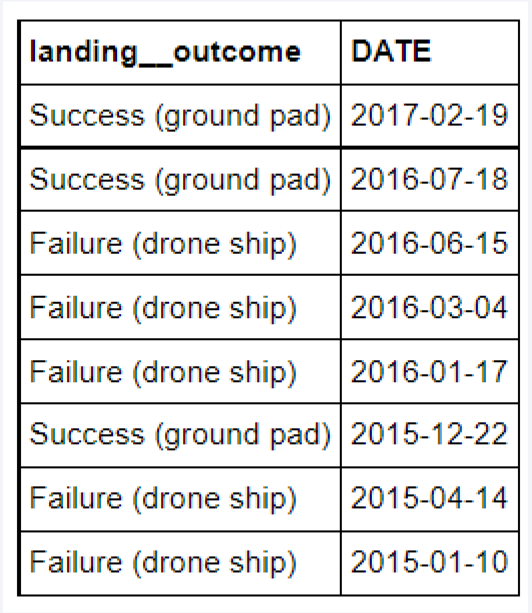
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - %sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where landing__outcome = 'Failure (drone ship)' and EXTRACT(YEAR FROM DATE)='2015';
- Present your query result with a short explanation here
  - where filters landing__outcome = 'Failure (drone ship)' and EXTR (
    DATE)='2015', and select shows information



1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
4	Success	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, DATE FROM SPACEXTBL WHERE (landing__outcome IN ('Failure (drone ship)', 'Success (ground pad)')) AND (DATE BETWEEN '2010-06-04' AND '2017-03-20') ORDER BY DATE DESC;
- Present your query result with a short explanation here
  - Select LANDING__OUTCOME and DATE columns, from SPACEXTBL, Where filters date between 2010-06-04 and 2017-03-20 and landing__outcome is Failure (drone ship) or Success (ground pad).



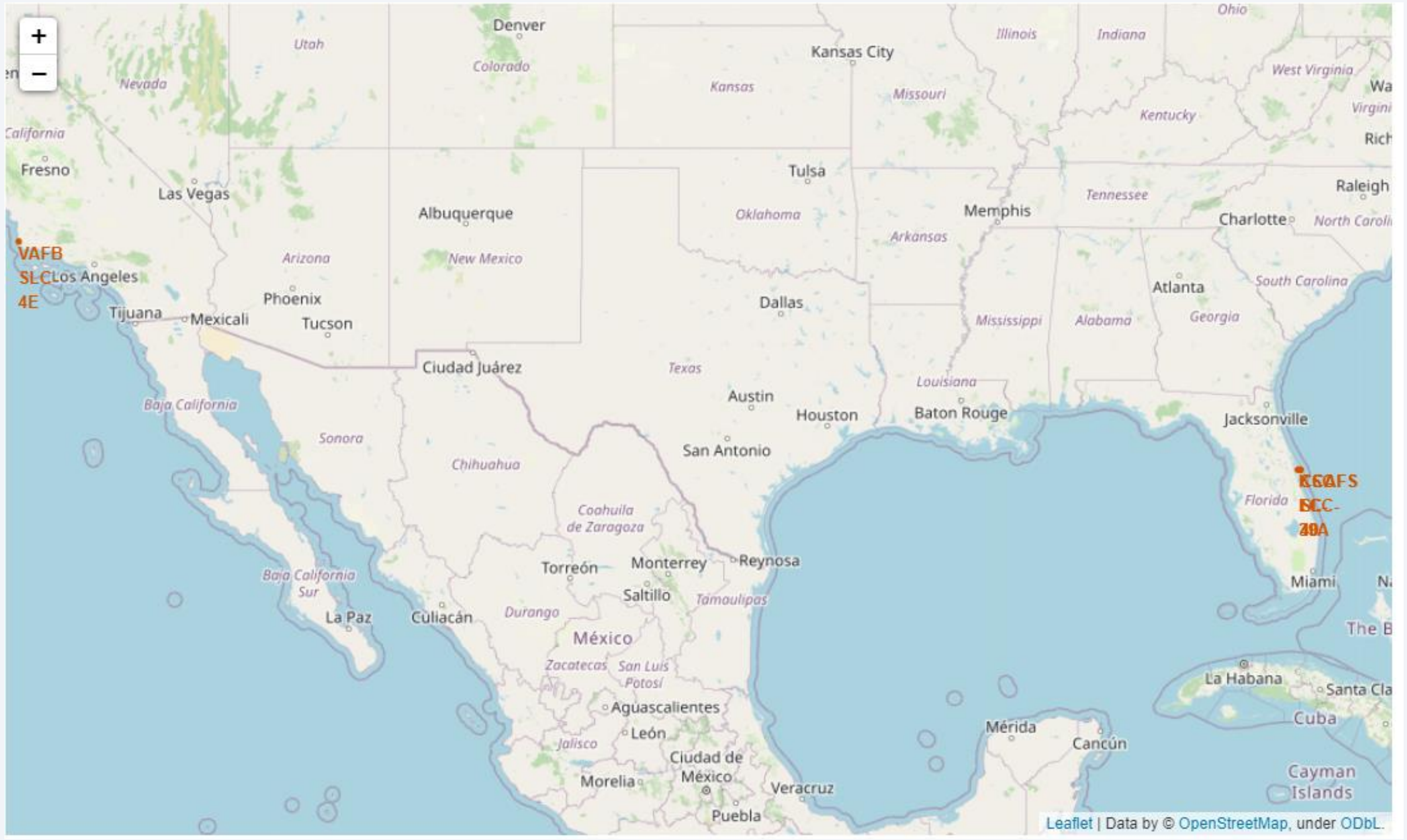
landing__outcome	DATE
Success (ground pad)	2017-02-19
Success (ground pad)	2016-07-18
Failure (drone ship)	2016-06-15
Failure (drone ship)	2016-03-04
Failure (drone ship)	2016-01-17
Success (ground pad)	2015-12-22
Failure (drone ship)	2015-04-14
Failure (drone ship)	2015-01-10

Section 3

Launch Sites Proximities Analysis

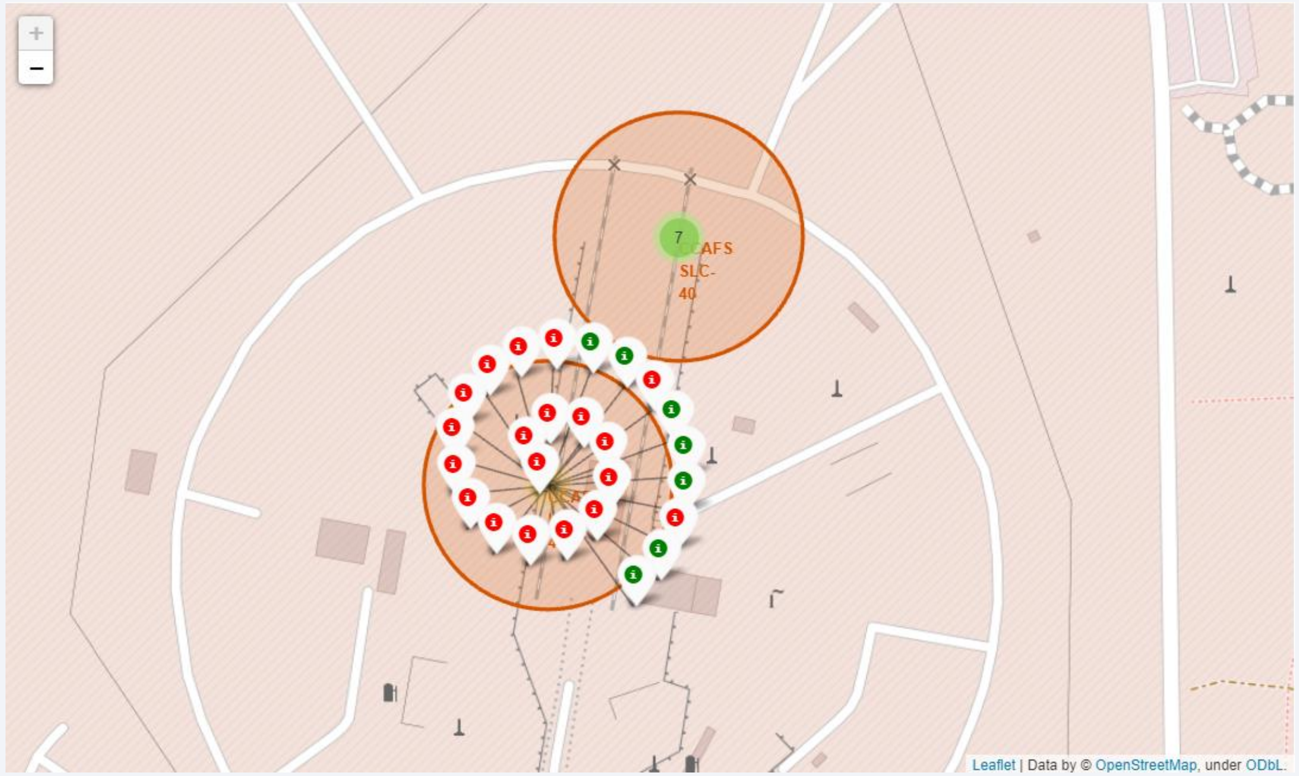
All launch sites marked on a map

- SpaceX launch sites are along American coasts, in FL and CA.



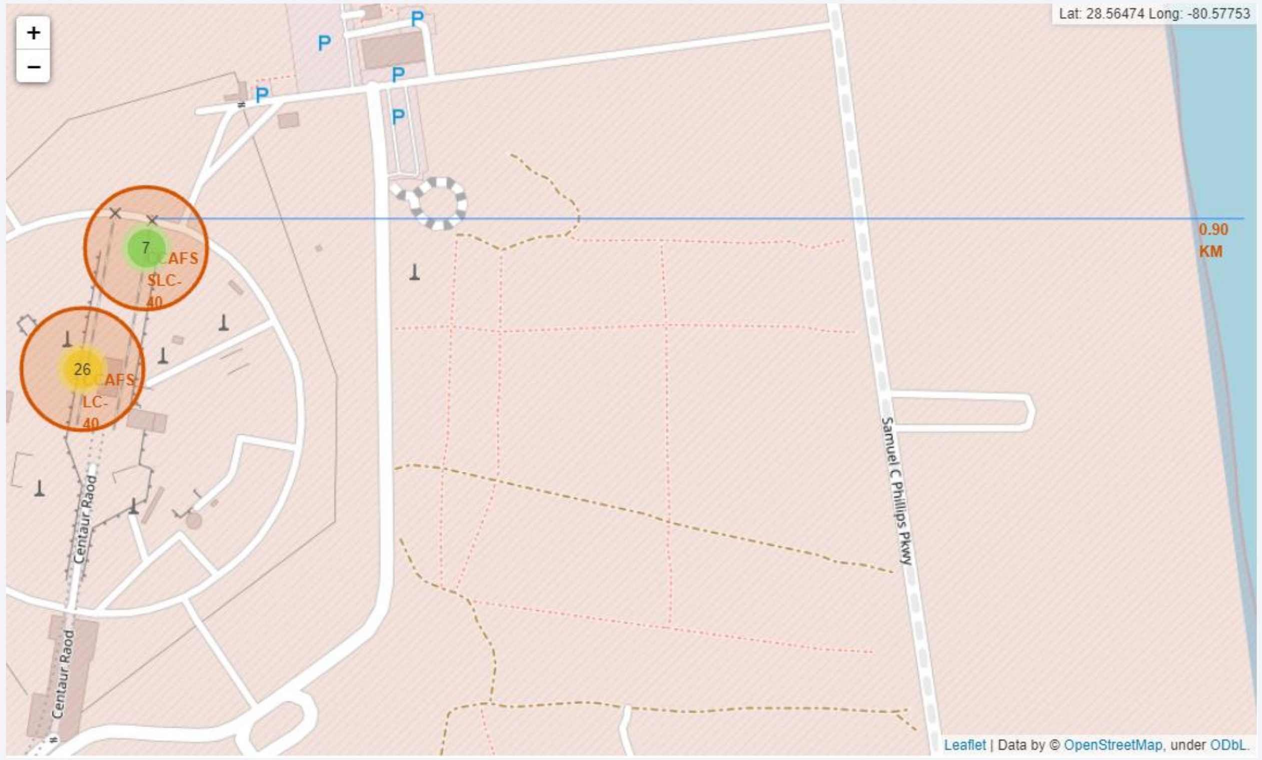
Mark successful and failed launches

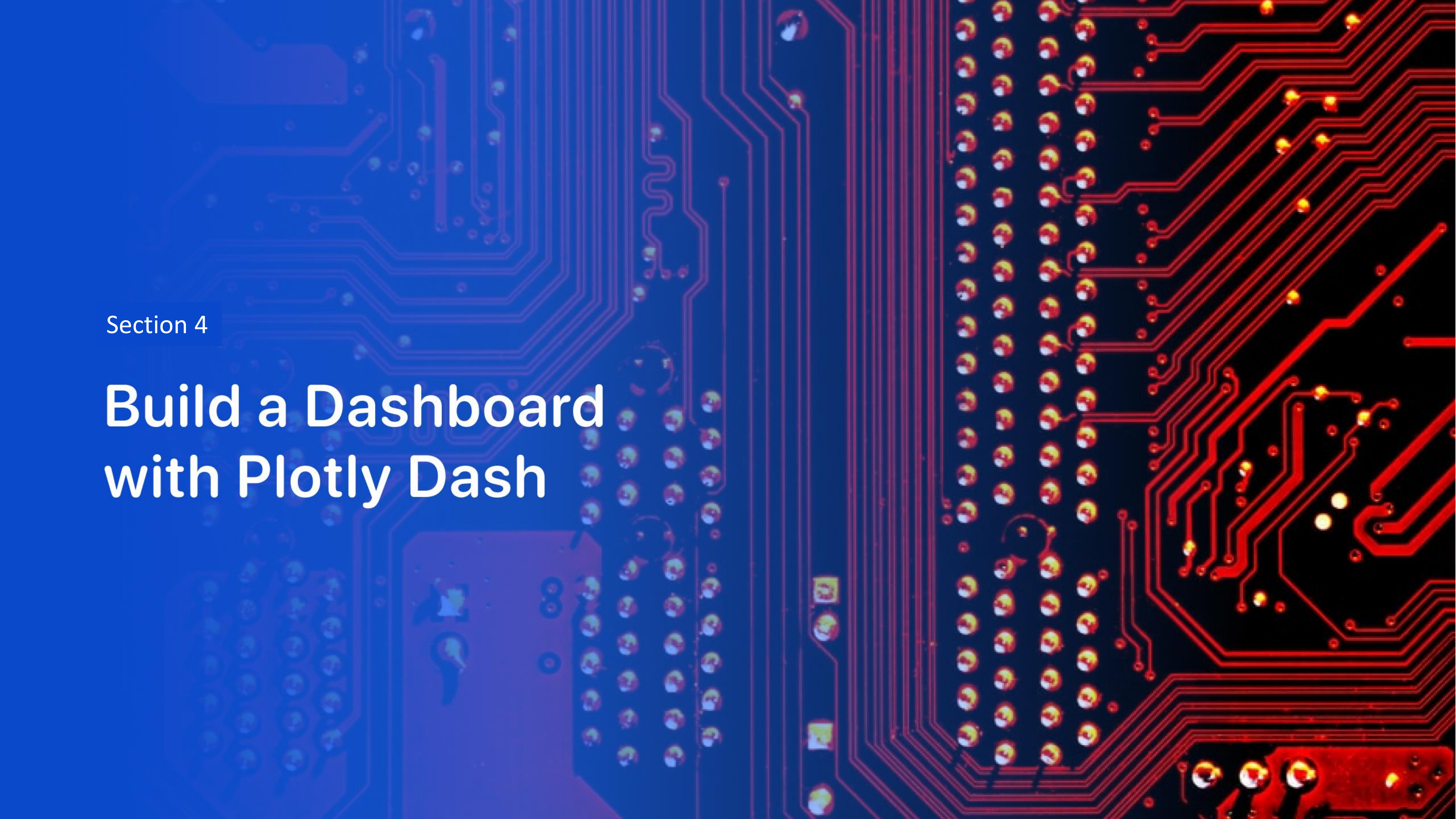
- Green shows successful launches; red shows failures…



Calculate distances

Purely as an example, I show the distance to the coastline.


Lat: 28.56474 Long: -80.57753
0.90 KM
7
CCAFS SLC-40
26
CCAFS LC-40
Centaur Raod
Samuel C Phillips Pkwy
Leaflet | Data by © OpenStreetMap, under ODbL.

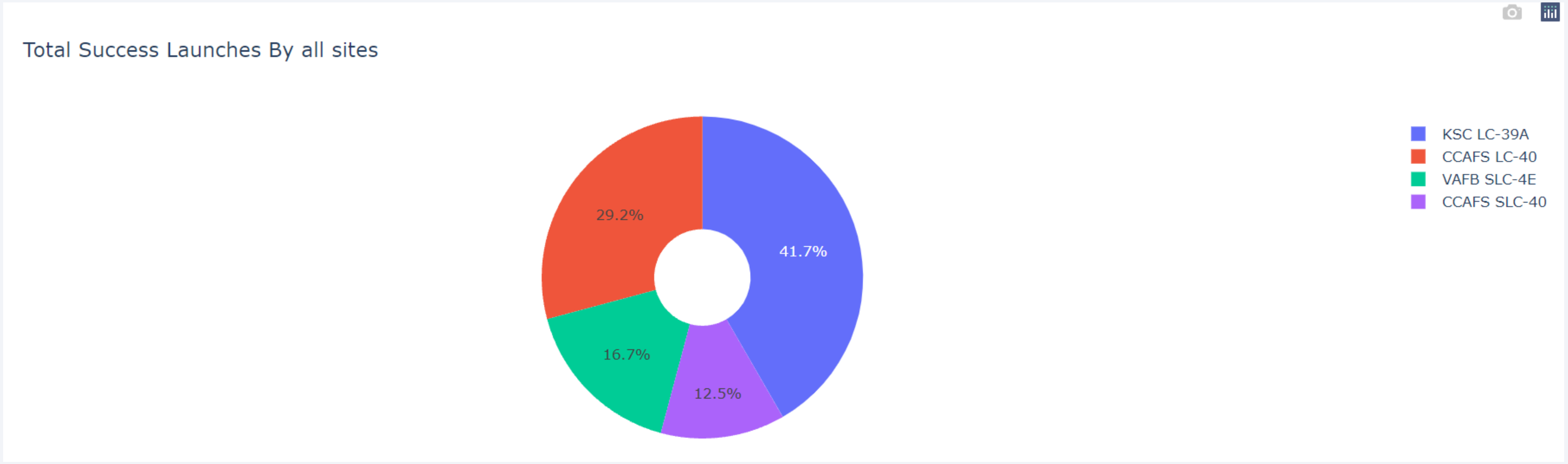


Section 4

Build a Dashboard with Plotly Dash

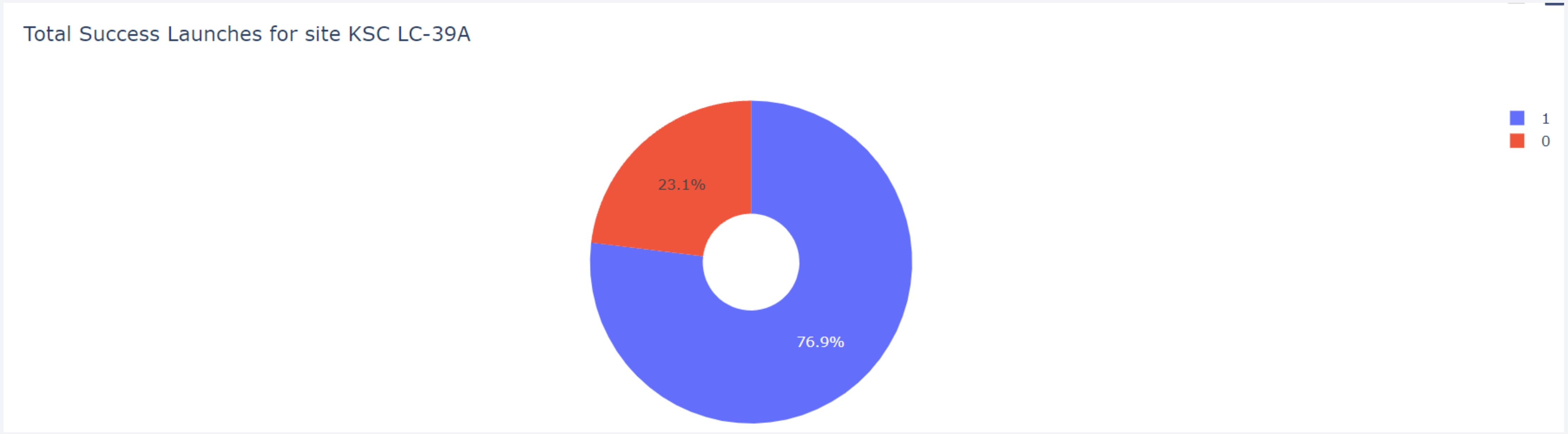
Total success launch by all sites

We can see that KSC LC-39A had the most successful launches.



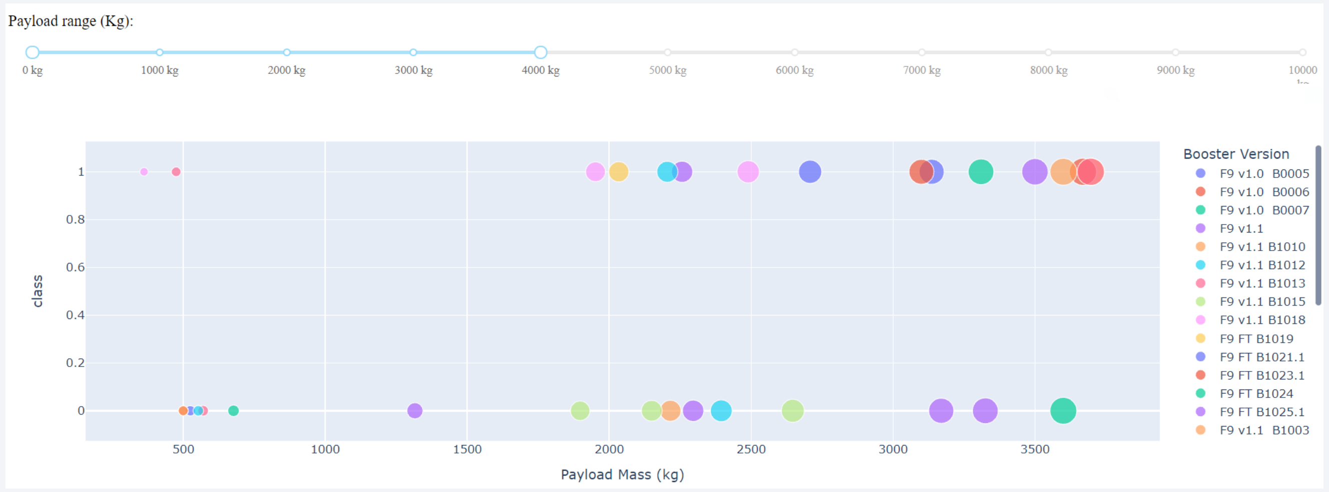
Total successful launches for KSC LC-39A

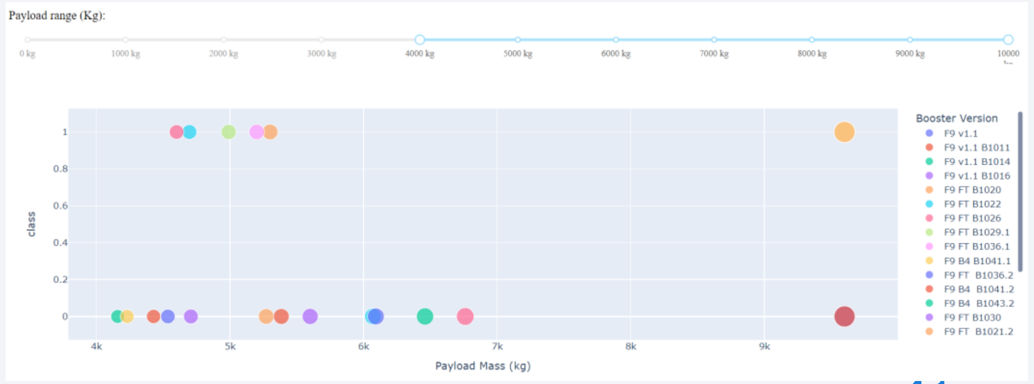
- KSC LC-39A achieved a 76.9% success rate!



Payload vs. Mass with different payload

- The success rate for low weighted payloads is higher than the heavy.



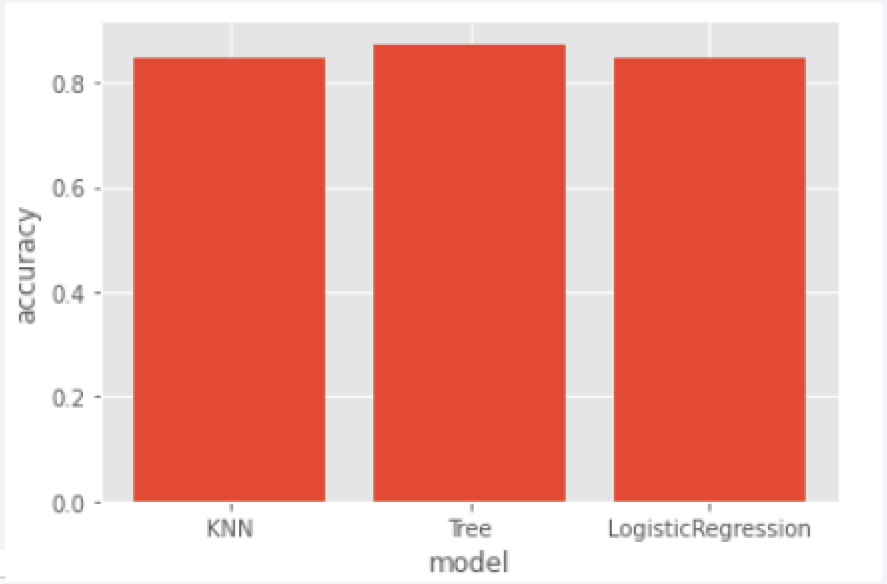


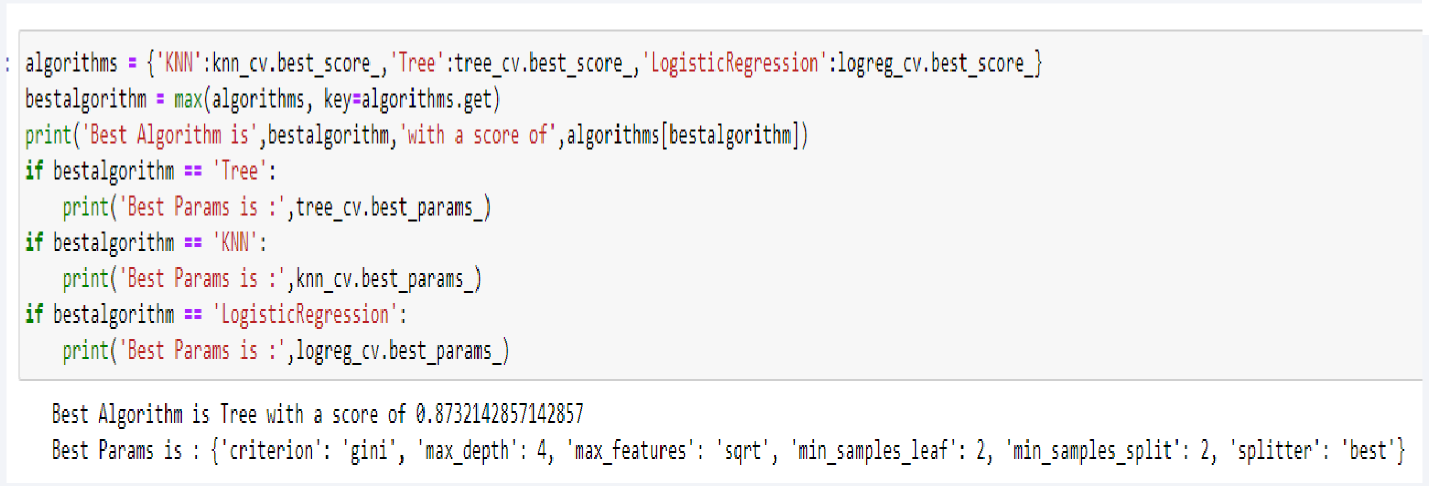
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Classification tree is most accurate!



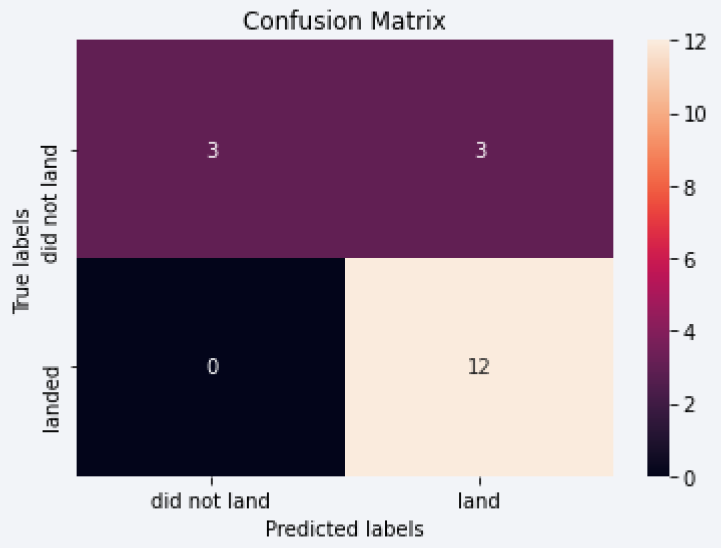


```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8732142857142857
Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'best'}
```

Confusion Matrix

- Tree can distinguish between the different classes. False positives are a major problem.



Conclusions

- The tree is best for machine learning in this dataset.
- Low weighted payloads perform better
- Success rates rise over time.
- KSC LC-39A had the most successful launch history
- Orbit GEO, HEO, SSO, ED-L1 had equally successful launch rates.

Appendix

- Nothing here I guess?

Thank you!