Fig. 1 Source: SpaceX

SpaceX Capstone Project

Samer Alhaddad

12 August 2023

Table of Contents

EXECUTIVE SUMMARY

In this project, the primary objective is to predict the successful landing of the Falcon 9 first stage, a critical factor in determining the cost of a SpaceX launch. This is achieved through the utilization of data from the SpaceX REST API and web scraping of Falcon 9 launch records from Wikipedia.

The project consists of the following key stages:

- **Data Collection and Data Wrangling:**
  Data is collected from both the SpaceX public API and Falcon 9 launch records extracted through web scraping. This combined dataset is then subject to data wrangling.
- **Exploratory Data Analysis (EDA):**
  Exploratory data analysis is conducted using visualizations with Pandas and SQL queries. Relationships between variables such as Flight Number, Payload Mass, Launch Site, Orbit, Class (Success or Failure), and Year are examined. These insights aid in selecting relevant variables for training the machine learning model.
- **Interactive Visual Analytics:**
  Interactive visual analytics are enabled using tools like Folium and Plotly Dashboard Web Application. Folium maps depict launch sites, successful and unsuccessful landings, and nearby landmarks. The dashboard includes pie charts and scatter plots that allow users to explore success rates across different launch sites and payload masses.
- **Predictive Analysis:**
  The predictive analysis stage involves the development of multiple machine learning models and choosing the best performant model. By leveraging the collected data and features, the model predicts the success of Falcon 9 first stage landings. The model achieves an accuracy of ~83.3%, providing the ability to make informed launch decisions.

INTRODUCTION

Fig. 2 Source: The Verge

- In space exploration, where it is now possible to restore and reuse the rocket's first stage, a challenging question persists: ***Can we predict whether SpaceX Falcon 9's first stage will land successfully?*** This question holds within it the potential to reshape the cost dynamics and environmental impact of space exploration.
- Our journey into answering this question revolves around data, seeking patterns and insights that could unravel the secrets of successful landings. By dissecting information from SpaceX's launches, we aspire to construct a predictive model that can anticipate the outcome of these landings.
- The implications could be profound, ranging from enabling calculated mission decisions to shaping the competitive landscape of space exploration.

METHODOLOGY

- We will discuss the methodologies applied through each stage of the project, including:
  - **Data Collection and Data Wrangling Methodologies**
  - **EDA and Interactive Visual Analytics Methodologies**
  - **Predictive Analysis Methodologies**

METHODOLOGY: Data Collection (API)

1. Request and parse the SpaceX launch data from static URL using the HTTP GET request
2. decode the response content as a Json using json() function and turn it into a Pandas data frame using json normalize() function.
3. Substitute ID values we got with corresponding values by extracting data from SpaceX API and storing each feature values in a separate list:
    1. from rockets endpoint we extracted: booster name
    2. from launchpads endpoint we extracted: launch site being used, the longitude, and the latitude
    3. from payloads endpoint we extracted: mass of the payload and the orbit that it is going to
    4. from cores endpoint we extracted: outcome of the landing, the type of the landing, number of flights with that core, whether grid fins were used, whether the core is reused, whether legs were used, the landing pad used, the block of the core which is a number used to separate version of cores, the number of times this specific core has been reused, and the serial of the core
4. We combine all data into a dictionary and convert into a Pandas data frame.
5. After we have collected the data, we filter the data frame to only include `Falcon 9` launches.

METHODOLOGY: Data Collection (Web Scraping)

1. Request the Falcon9 Launch Wikipedia page
2. Create a `BeautifulSoup` object from the HTML `response`
3. Extract and store all html tables into a list using BeautifulSoup find_all() function.
4. Get the 3rd table from the list, which contains the required data.
5. Extract column names from the HTML table header into a list.
6. Parse the HTML table and add the values to a dictionary, similar to what we did when collecting from REST API.
7. Convert dictionary into a Pandas data frame.

METHODOLOGY: Data Wrangling

- We have to deal with missing values.
- We replaced payload missing values with the mean.
- We left missing values of landing pad since it represents no landing pad was used.

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

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

Data Wrangling

We can see below that some of the rows are missing values in our dataset.

```
[36] 1 data_falcon9.isnull().sum()
```

```
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       5
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```

METHODOLOGY: Data Wrangling

- Next, we need to derive our target (Class) column based on Outcome column which can have the following values:
  - True Ocean – successfully landed to a specific region of the ocean
  - False Ocean –unsuccessfully landed to a specific region of the ocean.
  - True RTLS – successfully landed to a ground pad
  - False RTLS – unsuccessfully landed to a ground pad.
  - True ASDS –successfully landed to a drone ship
  - False ASDS –unsuccessfully landed to a drone ship.
  - None ASDS and None None – failure to land.
- And we set the value to 1 if Outcome was successful and 0 otherwise.

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

METHODOLOGY: EDA with Visualizations

- We used Scatter plots, Line plots, and Bar charts to visualize relationships and correlation between variables such as Payload Mass, Launch Site, Flight Number, Orbit, and Year to have better understanding of their impact on the target attribute.

- We use Scatter plots to visualize the relationship between two variables.
- We use Line plots to visualize changes over time.
- We use Bar charts to compare categorical variables.

METHODOLOGY: EDA with SQL

We performed SQL queries after loading our data into a DB2 database to gather more information.

Queries:

- **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.**
- **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.**
- **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.**

METHODOLOGY: Interactive Analytics

We used Folium library to mark launch sites on a map and try to find geographical patterns about them that could help with the analysis.

Mark all launch sites on map

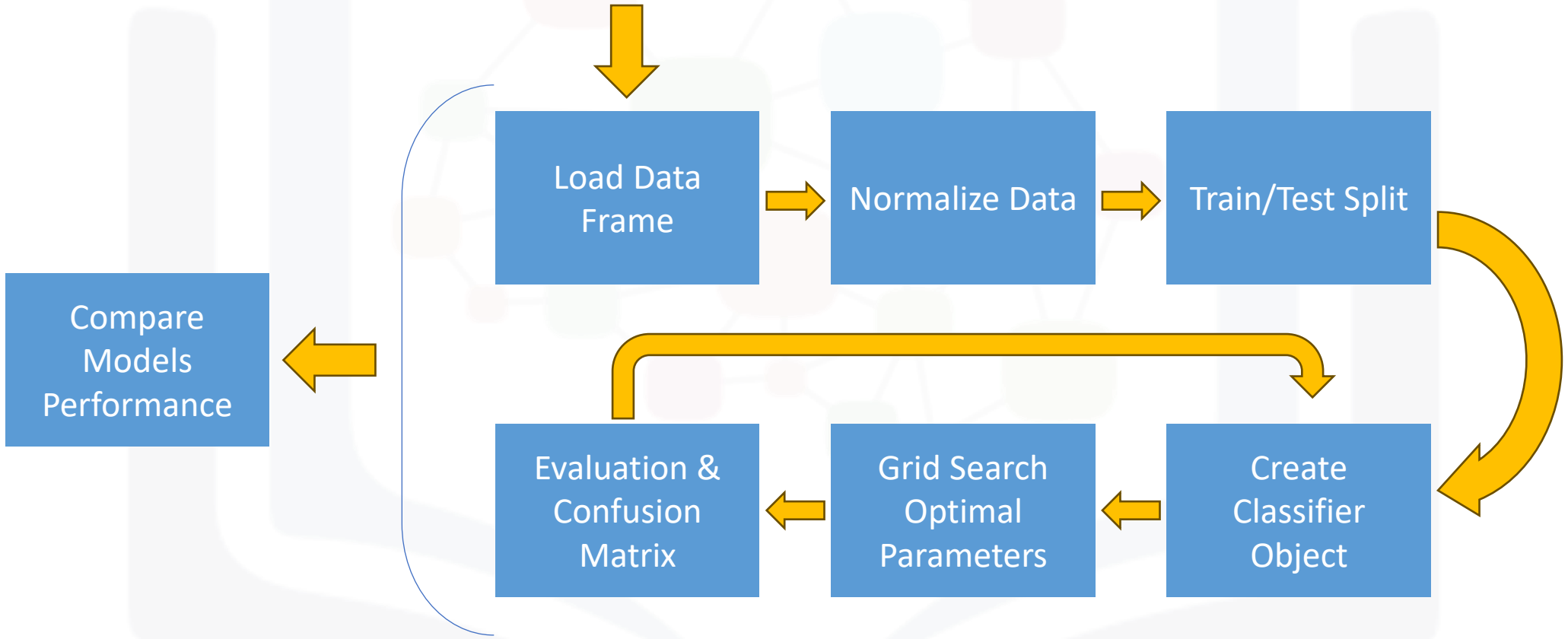
Mark the success/failed launches for each site on the map

Calculate the distances between a launch site to its proximities

METHODOLOGY: Plotly Dashboard

- We used Plotly Dash to build an interactive web-based dashboard that we can use to streamline the analytics process and have dynamic plots instantly generated rather than static ones.
- We added a dropdown menu to select one of the launch sites which will then plot a pie chart of its success rate and a scatter plot of success launches of payloads by having a slider range to select payload mass.
- We can also select ‘All’ which will then plot the statistics for all sites.
- The aim of this dashboard is to help us answer:
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range has the highest launch success rate?
  - Which payload range has the lowest launch success rate?
  - Which F9 Booster version has the highest launch success rate?

METHODOLOGY: Predictive Analysis

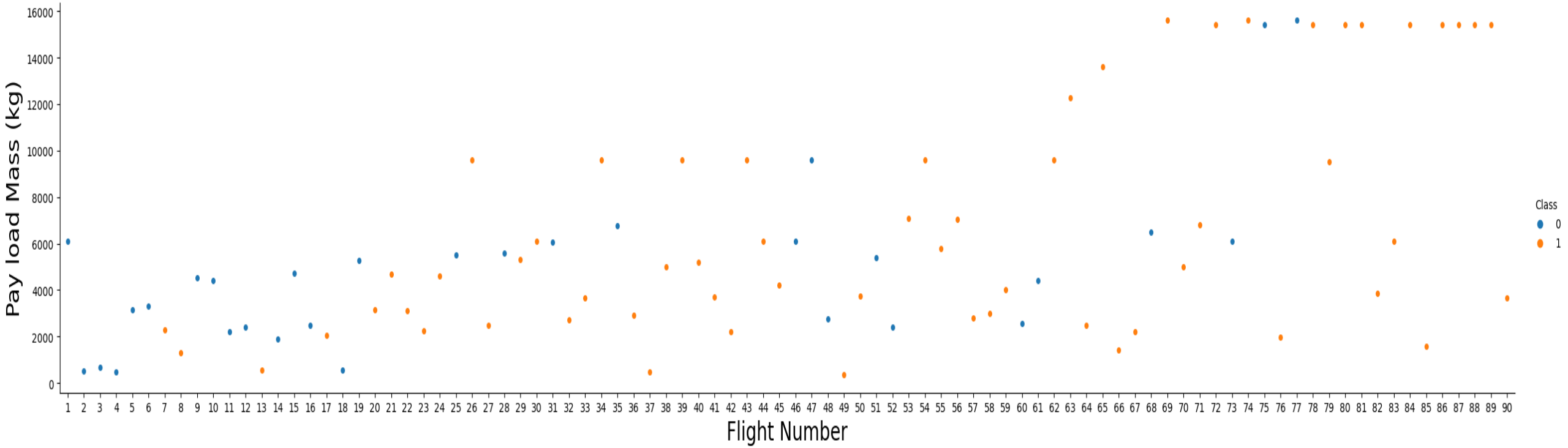


RESULTS

- We will discuss the results of the project:
  - **EDA with visualization**
  - **EDA with SQL**
  - **Interactive map with Folium**
  - **Plotly Dashboard**
  - **Predictive Analysis (Classification)**

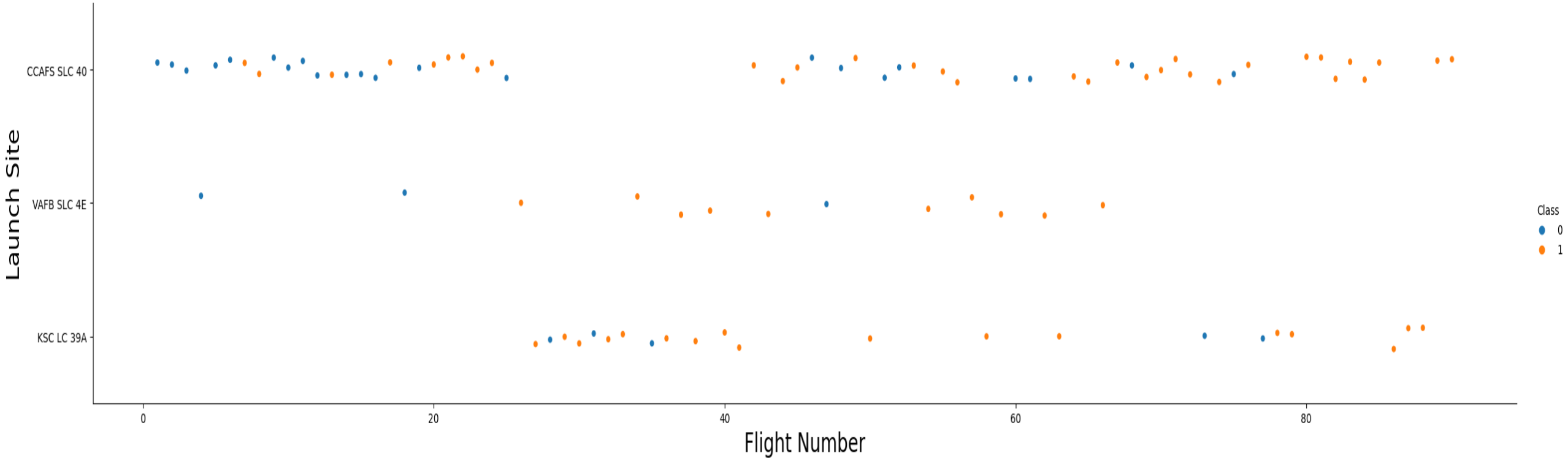
EDA with Visualization

Flight Number vs. Payload Mass



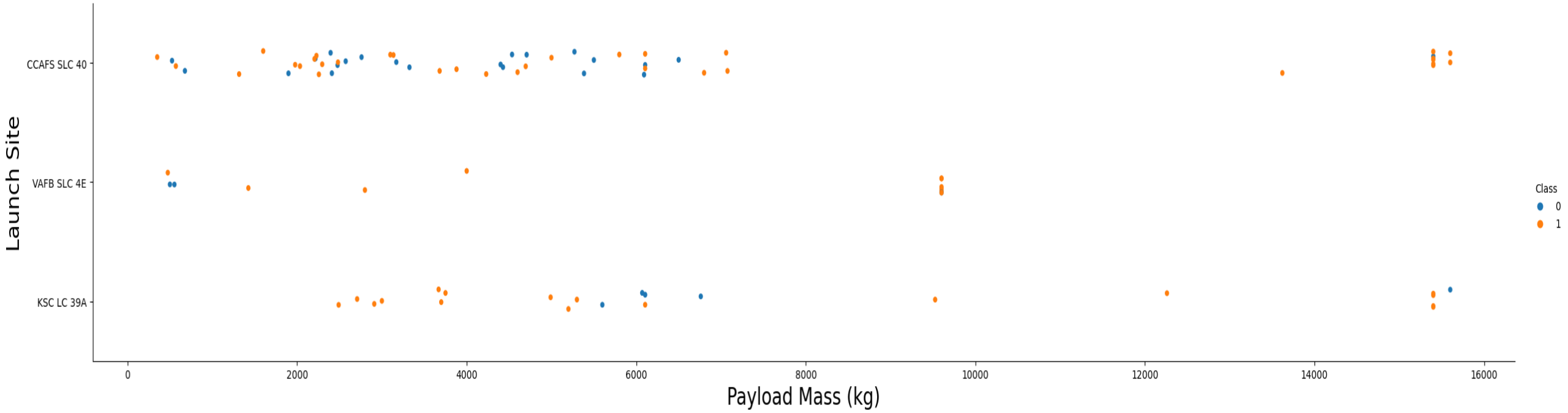
We notice an upward trend in successful launches for heavy payloads over the years.

Flight Number vs. Launch Site



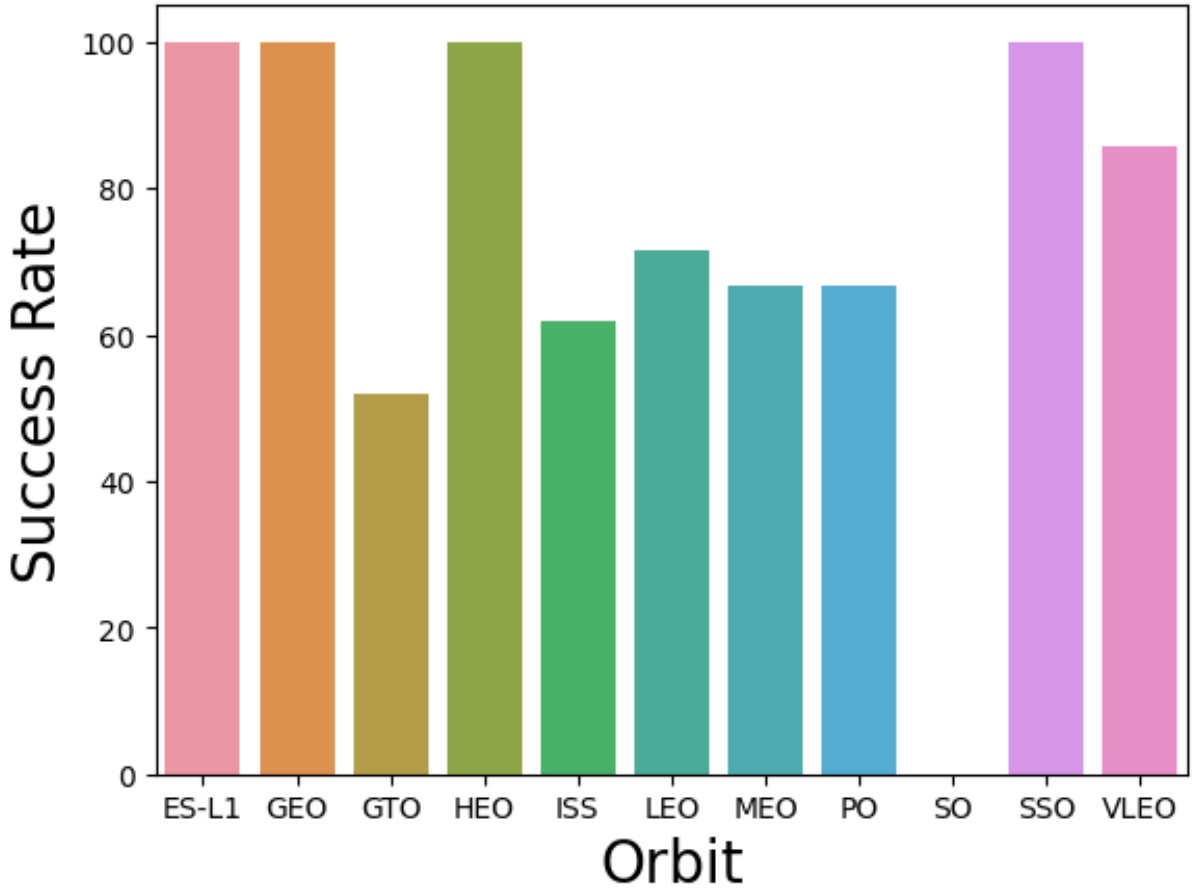
We notice a general improvement in success rate among all sites as flight number increases

Payload Mass vs. Launch Site



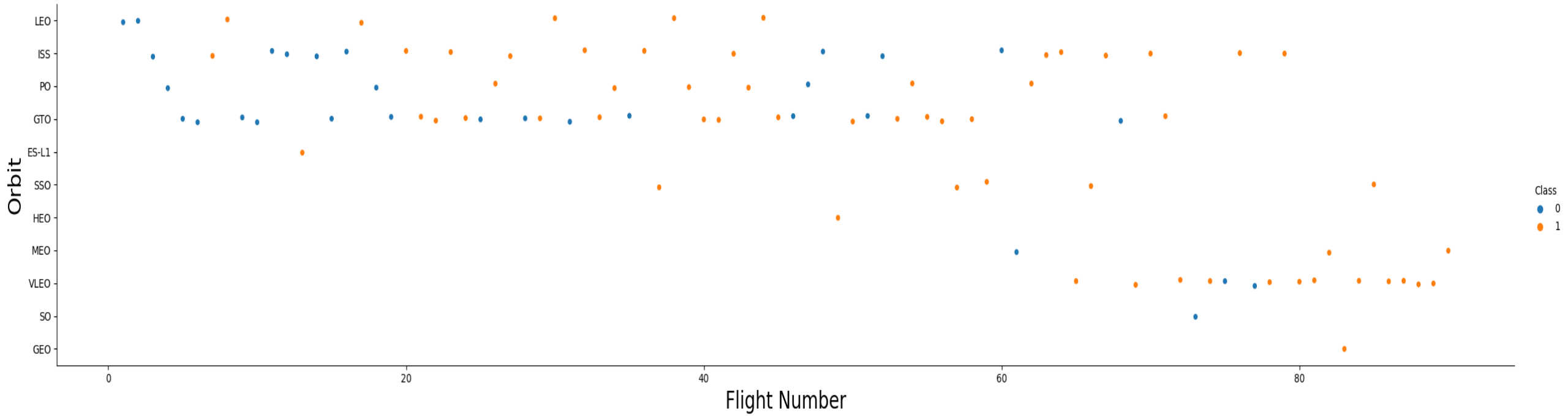
We notice VAFB doesn't launch missions with payloads exceeding 9000 kg.
Additionally, most payloads seem to be ranging between 2000 and 7000 kg.

Success Rate for each Orbit



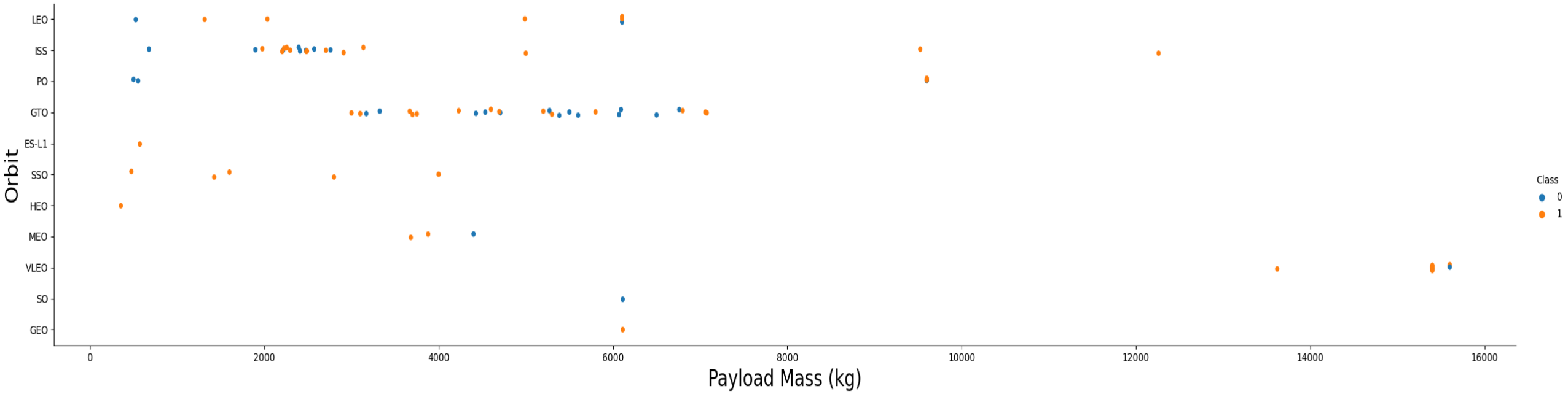
We notice perfect mission success rate in orbits ES-L1, GEO, HEO, and SSO. While the worst success rate is for SO with 0%, followed by GTO around 50%.

Flight Number vs. Orbit



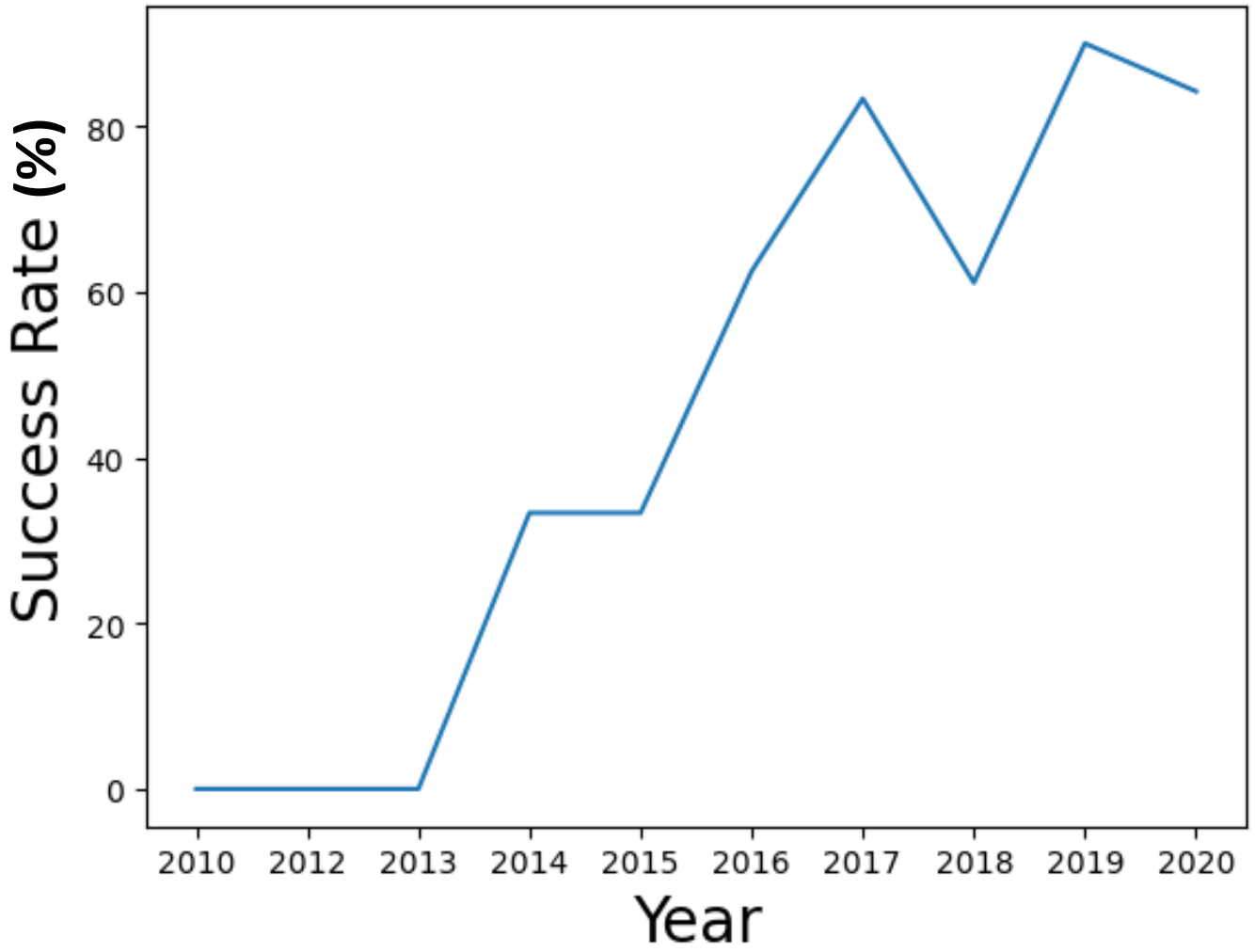
There is an obvious change in orbits that took place around flight 60.
SpaceX started trying VLEO, SO, and GEO in addition to SSO and ISS.
Doing so resulted in a higher success flights prior to flight 60.

Payload Mass vs. Orbit



ES-L1, SSO, HEO, MEO have high success with light payload.
VLEO has high success with heavy payload.
We can't determine a relationship between payload and ISS or payload and GTO.

Launch Success yearly trend



- Sharp increase in success rate after 2013, reaching its peak in 2019 at 90%.
- Slight decrease in 2018.
- Followed by another one in 2020 possibly due to the pandemic.

EDA with SQL

Unique launch sites

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

[32]:
```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE
```

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

[32]:

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

Site names beginning with ‘CCA’

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

[33]:
```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

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

[33]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total payload carried by NASA(CRS)

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

[34]:
```
%%sql
    SELECT "Customer", SUM(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
    WHERE "Customer" = "NASA (CRS)"
```

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

[34]:

Customer	SUM(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

AVG payload carried by F9 v1.1

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

[16]:
```
%%sql
    SELECT "Booster_Version" AS BV, AVG(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
    WHERE BV LIKE 'F9 v1.1%'
```

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

[16]:

BV	AVG(payload_mass__kg_)
F9 v1.1 B1003	2534.6666666666665

1st landing in ground pad

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

Hint:Use min function

[28]:
```
%%sql
    SELECT MIN("Date")
    FROM SPACEXTABLE
    WHERE "Landing_Outcome" = "Success (ground pad)"
```

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

[28]:

MIN("Date")
2015-12-22

Boosters with successful drone ship landing and 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 ¶

[35]:
```
%%sql
    SELECT DISTINCT("Booster_Version")
    FROM SPACEXTABLE
    WHERE "Landing_Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

[35]:

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

Success and Failure Count

List the total number of successful and failure mission outcomes

[31]:
```
%%sql
    SELECT "Mission_Outcome", COUNT(*)
    FROM SPACEXTABLE
    GROUP BY "Mission_Outcome"
```

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

[31]:

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

Boosters with maximum payload

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

[37]:
```
%%sql
    SELECT "Booster_Version"
    FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

[37]:

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

Drone ship failure landing (2015)

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.

[46]:
```
%%sql
    SELECT SUBSTR("Date", 6, 2) AS Month, SUBSTR("Date",1,4) AS Year, "Landing_Outcome", "Booster_Version", "Launch_Site"
    FROM SPACEXTABLE
    WHERE "Landing_Outcome" = "Failure (drone ship)" AND SUBSTR("Date",1,4)='2015'
```

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

[46]:

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcome

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.

[48]:
```
%%sql
    SELECT "Date", "Landing_Outcome", COUNT(*) AS "Count"
    FROM SPACEXTABLE
    WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY "Landing_Outcome"
    ORDER BY "Count" DESC
```

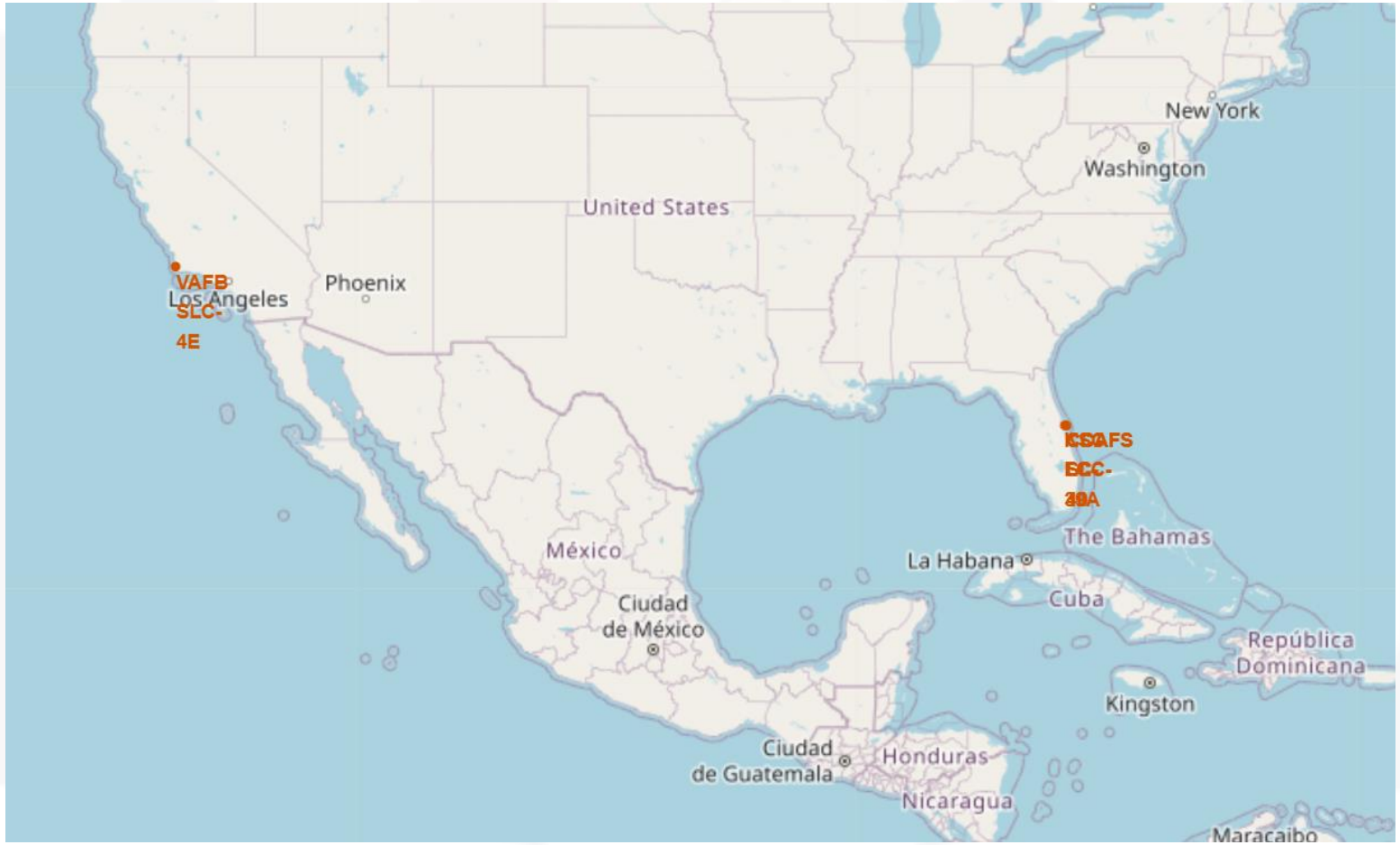
```
 * sqlite:///my_data1.db
Done.
```

[48]:

Date	Landing_Outcome	Count
2012-05-22	No attempt	10
2015-12-22	Success (ground pad)	5
2016-08-04	Success (drone ship)	5
2015-10-01	Failure (drone ship)	5
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2015-06-28	Precluded (drone ship)	1
2010-08-12	Failure (parachute)	1

Interactive map with Folium

Launch Sites

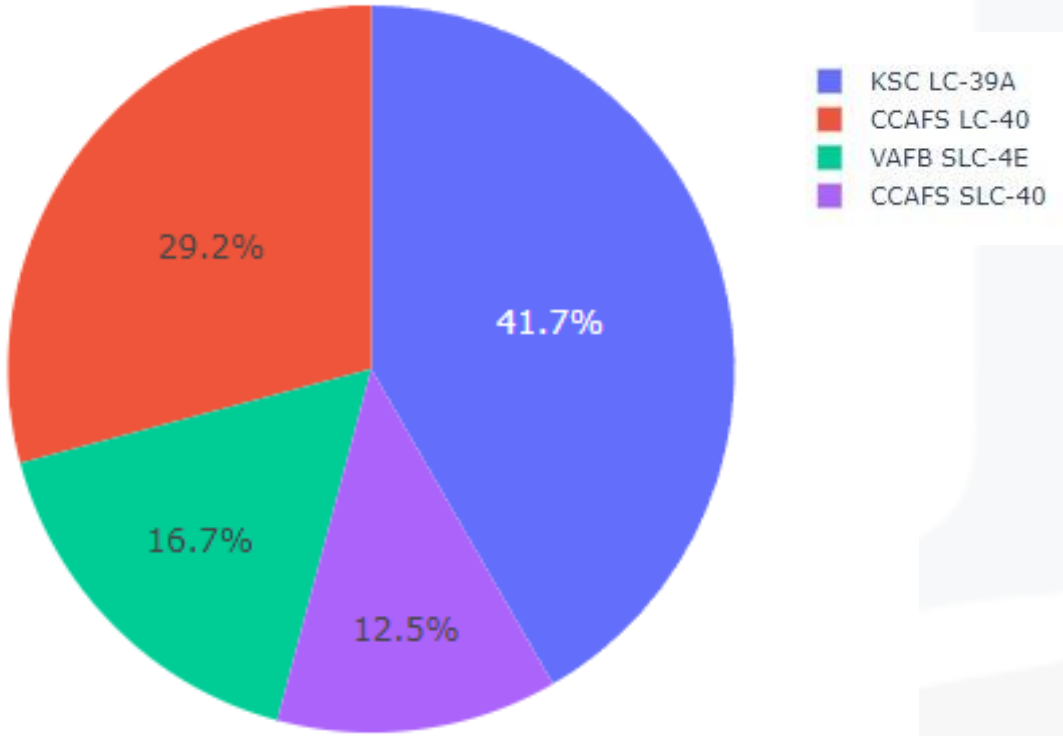


Success/Failure Markers

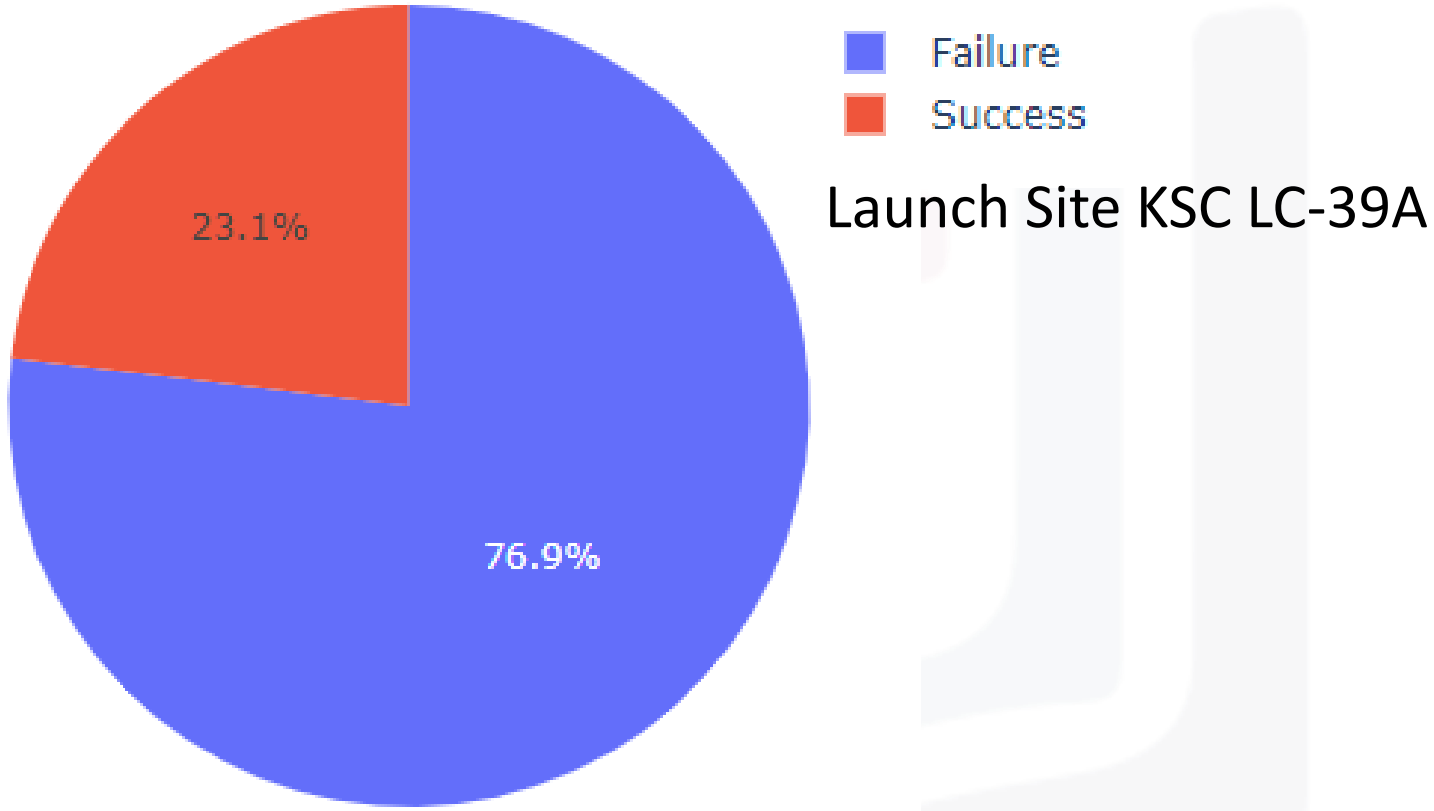
Proximity to coastline

Plotly Dashboard

Success for each site

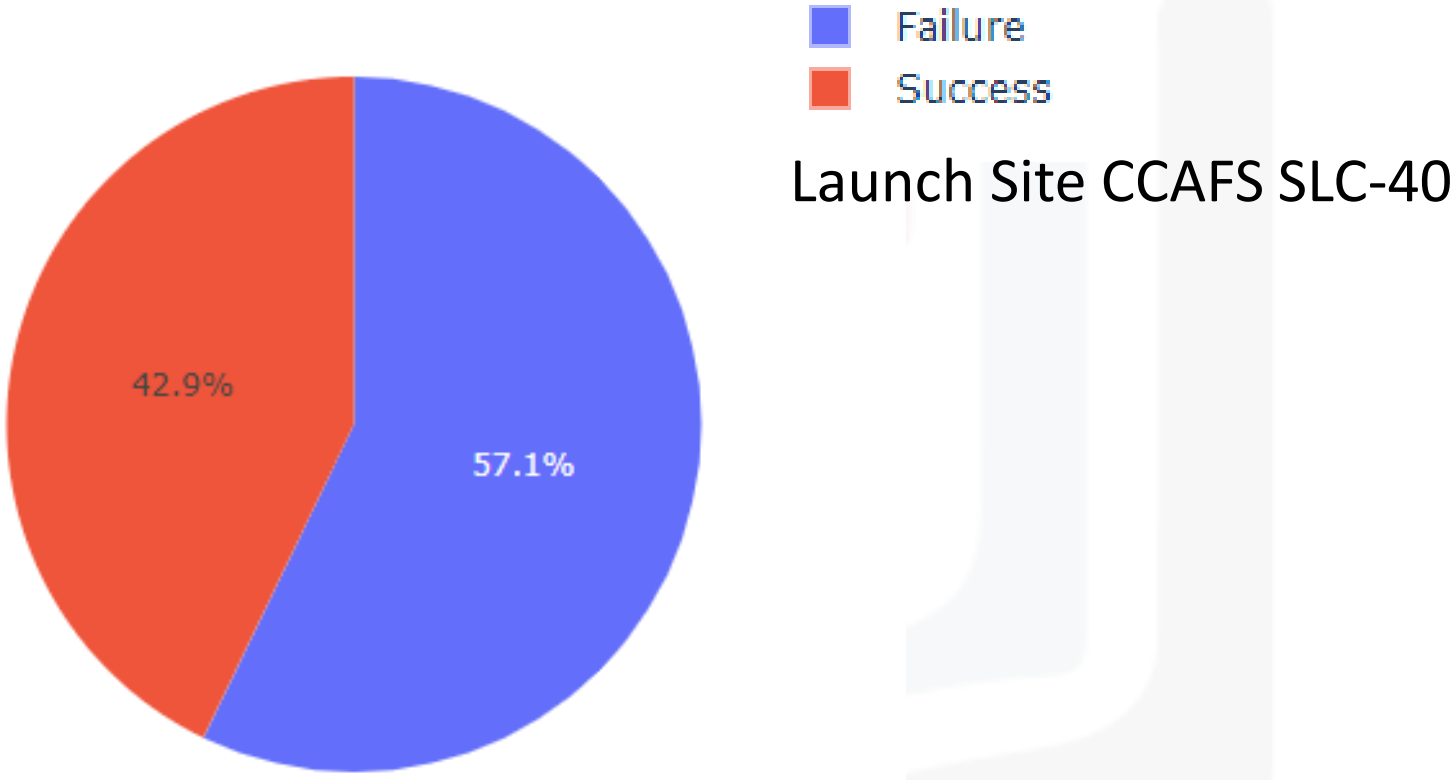


Highest Success rate



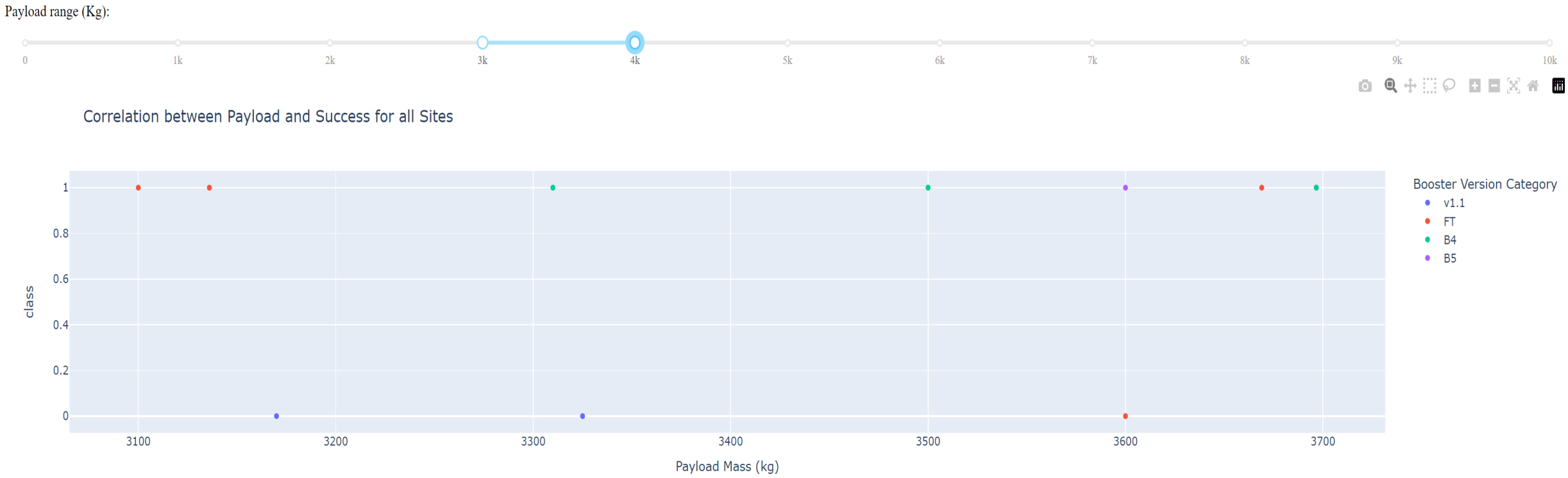
Launch Site KSC LC-39A

Lowest Success rate



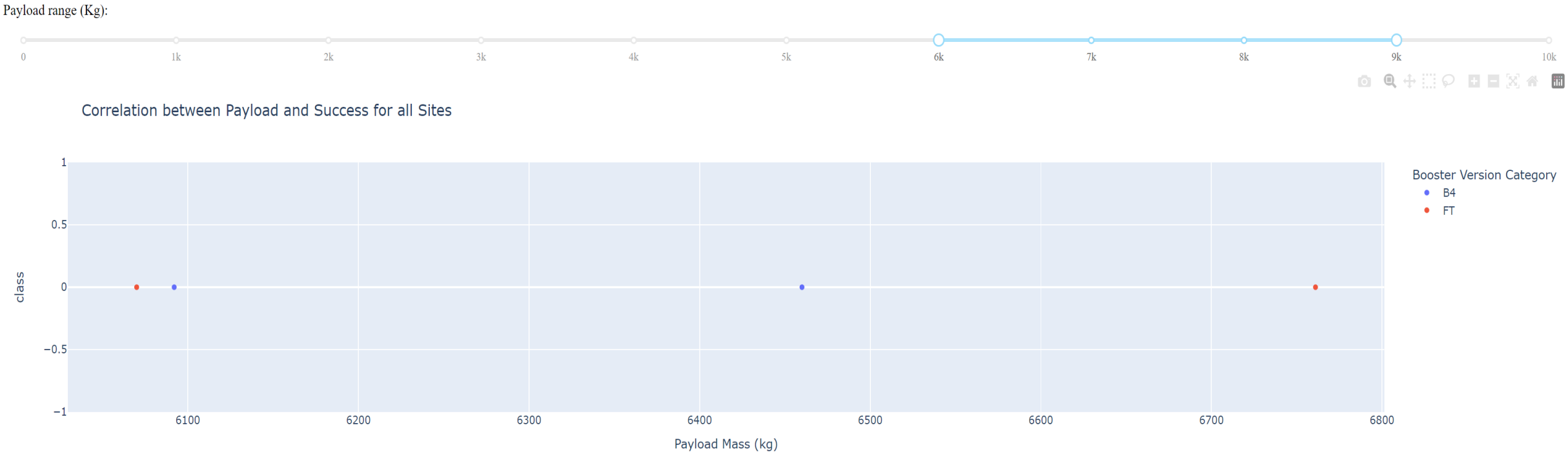
Launch Site CCAFS SLC-40

Most successful payload range



We can see the most efficient payload is between 3000 and 4000 kg having the highest number of successful landings.

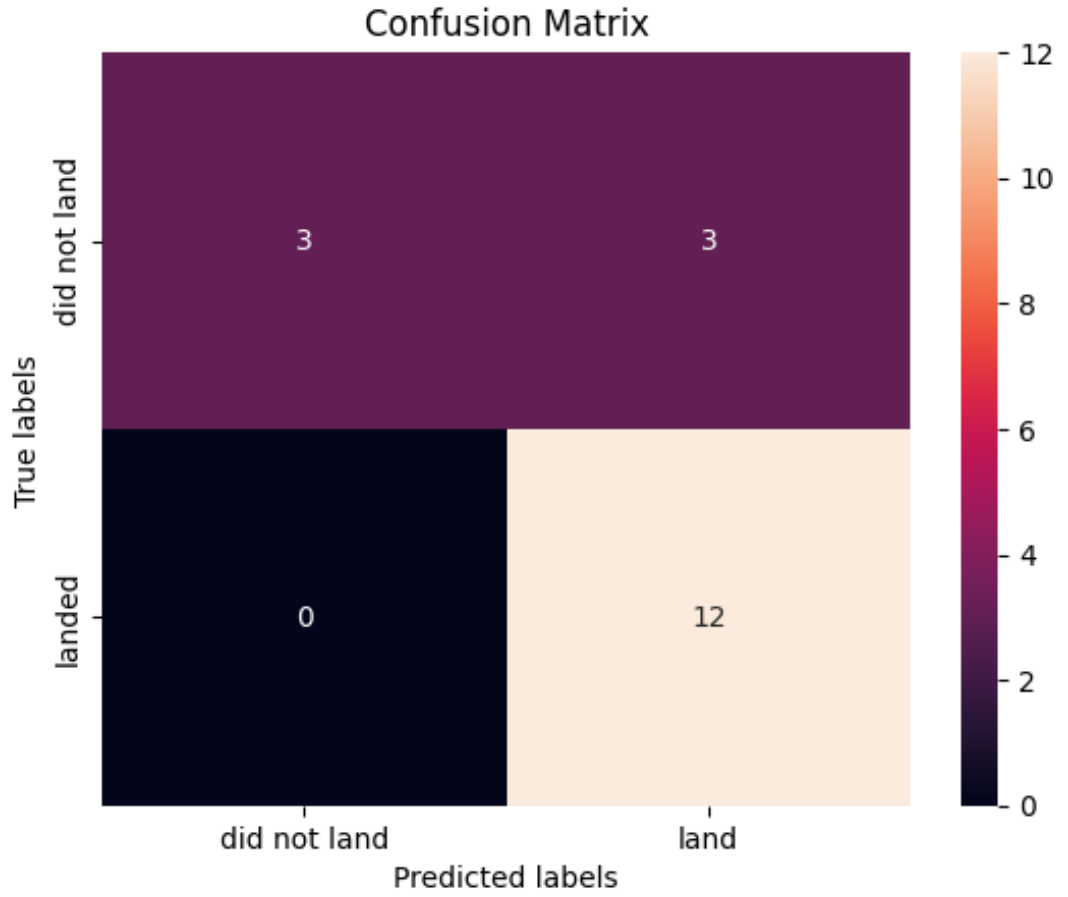
Least successful payload range



The least efficient payload is between 6000 and 9000 kg with no successful landings.

Predictive Analysis

Confusion Matrix (all models)



Comparing Model Performance

Find the method performs best:

[32]:
```
accuracy = {'Logistic Regression':logreg_cv.score(X_test, Y_test),
            'Suuport Vector Machine':svm_cv.score(X_test, Y_test),
            'Decision Tree':tree_cv.score(X_test, Y_test),
            'K Nearest Neighbors':knn_cv.score(X_test, Y_test)}
for k in accuracy:
    print(k, accuracy[k])
```

```
Logistic Regression 0.8333333333333334
Suuport Vector Machine 0.8333333333333334
Decision Tree 0.8333333333333334
K Nearest Neighbors 0.8333333333333334
```

We conclude that all models have the same accuracy

CONCLUSION

- Our objective was to design a machine learning model aimed at competing with SpaceX. The model's primary aim is to predict the successful landing of first stage, potentially leading to savings in space mission costs. Our approach involved utilizing data from the public SpaceX API and conducting web scraping of SpaceX Wikipedia pages.

- The culmination of our efforts resulted in the creation of a machine learning model reaching an impressive accuracy of 83.3%. This model provides the capability to make informed predictions regarding the likelihood of a successful landing before a launch occurs, enabling crucial pre-launch decisions.

- While our model stands as a significant achievement, the potential for further enhancement remains. The collection of more data is encouraged, as it holds the potential to refine the machine learning model further, potentially leading to increased accuracy and efficiency in the decision-making process.

APPENDIX

GitHub Repo Link:

https://github.com/Samer-Haddad/IBM-Data-Science/tree/main/10.%20Applied%20Data%20Science%20Capstone