IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mehdi Bohloul
Sep 2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- The project, executed for "Space Y," aimed to predict the successful landing of the first stage of SpaceX rockets by training machine learning models on launch data collected via a third-party API and Web Scraping. This prediction is crucial for determining accurate rocket launch costs based on reusability.
- Exploratory and interactive analytics using Folium and Plotly Dash identified key performance indicators: the launch success rate generally increased after 2013, KSC LC-39A was found to have the highest success rate, and orbits ES-L1, SSO, HEO, and GEO achieved 100% success.
- Data trends revealed that most failures occurred within the initial 20 experimental flights and in the payload range under 8000kg. Notably, every launch after flight number 80 was successful.
- Four classification models (Linear Regression, SVM, Decision Tree, and KNN) were trained and optimized using `GridSearchCV`. Although the Decision Tree had the highest cross-validation score, all models achieved the same prediction score on the test data due to class imbalance, providing the necessary predictive insights for "Space Y" to conserve resources and time.

Introduction

This project places me as a data scientist at "Space Y," a new company competing with SpaceX. My goal is to analyze how SpaceX achieves **low launch costs** through its winner strategy: **reusing the first stage of its rockets**. I will use SpaceX data to understand and predict their operations.

To answer how much the rocket launch will cost, which is directly tied to the reusability of the rocket, I must first predict if the rocket's first stage will land successfully. This will be done by training a machine learning model on SpaceX’s launch data.

Section 1

Methodology

Methodology

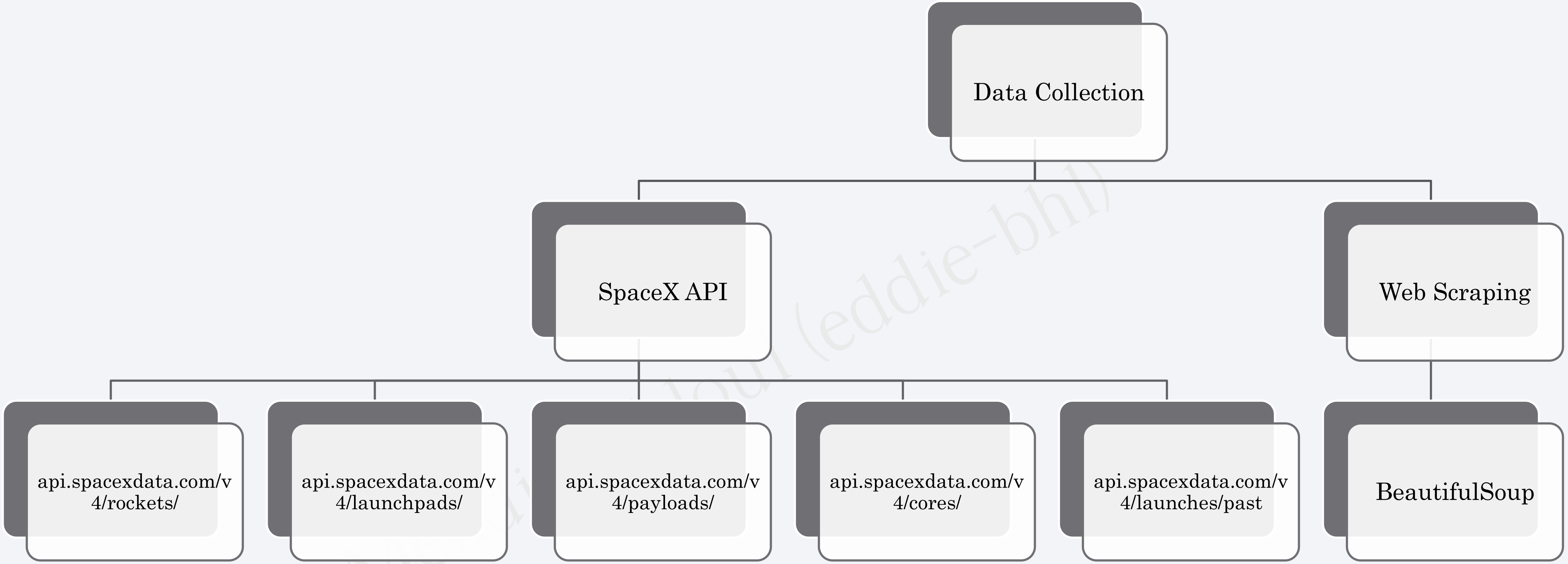
Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

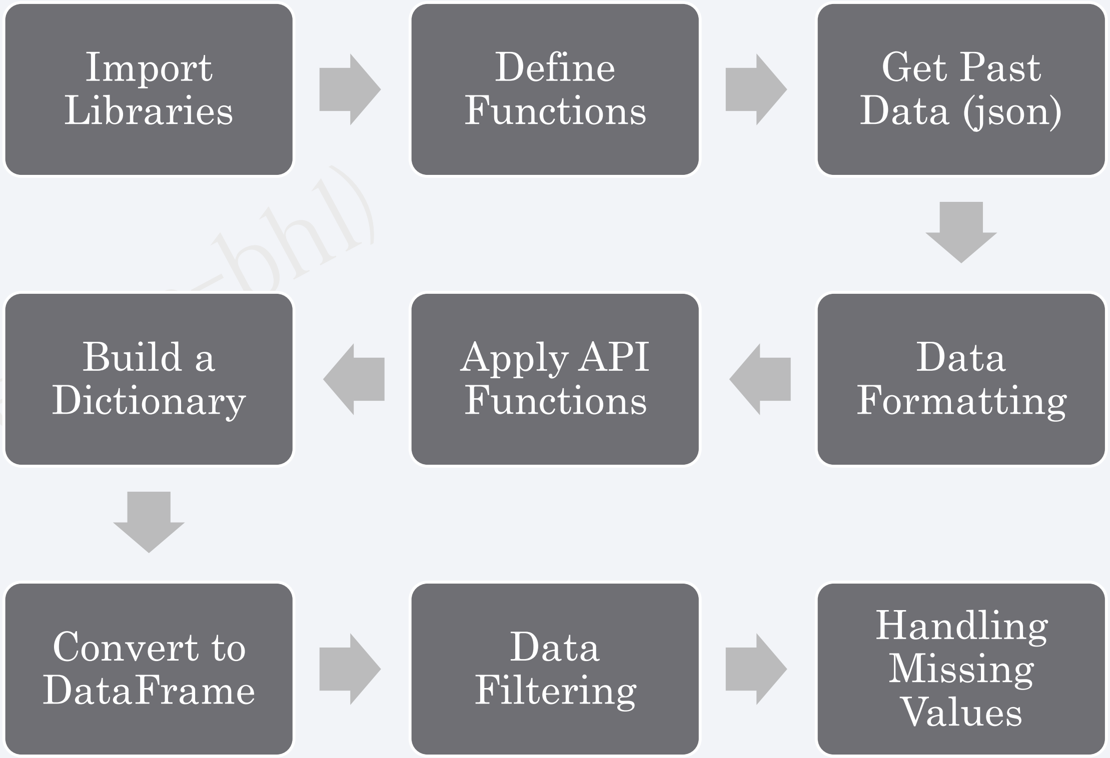
- The data was collected through two methods:1) Using a third-party SpaceX REST API. 2) Using Web Scraping
- The API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- This result can be viewed by calling the .json() method. Our response will be in the form of a list of JSON objects.
- In web scraping, we extract information of the launches from a HTML table.

Data Collection



Data Collection – SpaceX API

- I made a get request to the SpaceX API, defined a few API call functions, and then I performed a few basic data wrangling and formatting.



GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/8298a8ee5386d860ea2df7bc751d966c2b7a5f06/Notebooks/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – SpaceX API

Function	Target	Endpoint
getBoosterVersion	Rockets	api.spacexdata.com/v4/rockets/
getLaunchSite	Launchpad name & location	api.spacexdata.com/v4/launchpads
getPayloadData	Payload mass & Orbit	api.spacexdata.com/v4/payloads/
getCoreData	Core Data	api.spacexdata.com/v4/cores/

Data Collection - Scraping

- I defined a few functions to extract required information. Then I created a BeautifulSoup object to parse HTTP response from our get request.
- Then a DataFrame was created using a dictionary containing all of the required information from columns and rows in the table.

TASK 1: Request the Falcon9 Launch Wiki page from its URL

+ 6 cells hidden

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

+ 11 cells hidden

TASK 3: Create a data frame by parsing the launch HTML tables

+ 10 cells hidden

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/9bc87df1c2c66f87db842904ca552611c1784393/Notebooks/jupyter-labs-webscraping.ipynb

Data Wrangling

This step consists of two main stages:

1. Exploratory Data Analysis (EDA)
2. Determining Training Labels

Initial Data Inspection

- Identify missing values
- Identify data types

Exploratory Data Analysis (EDA)

- number of launches on each site
- number of occurrence of each orbit
- number of occurrence of mission outcomes

Feature Engineering

- Identify Unsuccessful Outcomes
- Create 'Class' Column

Validation & Export

- Determine the average success rate
- Export to CSV

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/fd6102b1339f5579fe636f1721c3e28245dce388/Notebooks/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Scatter Plot (scatterplot, catplot)

To show relationship between different variables with hue set to class category (successful vs. unsuccessful).

- FlightNumber vs. PayloadMass
- FlightNumber vs. LaunchSite
- PayloadMass vs. LaunchSite
- FlightNumber vs. Orbit
- PayloadMass vs. Orbit

Bar Plot

To show relationship between success rate and orbit.

- Class vs. Orbit

Line Plot

To represent the trend of success rate in different years.

- Date(Year) vs. Class

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/4c8cdbd4a0c854ff92f016f535eb84baf857fef3/Notebooks/edadataviz.ipynb

EDA with SQL

The SQL queries performed are as follows:

- **Creating** a database using '**Connect**' command.
- **Drop** existing **table** if any and **creating** a new one.
- **Display** the names of the **unique launch sites.**
- **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** with **success in drone ship** and **payload mass** greater than **4000 but less than 6000**.
- **List** the **total number** of successful and failure **mission outcomes**.
- **List** all **booster versions** with the **maximum payload mass**, using a **subquery** with a suitable **aggregate function**.
- **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.

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/0472d50446af2f0c5bface61e559b720a87b8f90/Notebooks/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

1. Each site location was identified using 'groupby' method.

2. A map circle and a marker was added showing NASA JSC location
   - We created a folium map object with NASA JSC as the center.

3. Four circles and markers were added for each four site locations.

4. Launch outcomes for each site were added.
   - To see which sites have high success rates
   - Green marker for successful launches and red marker for unsuccessful ones.
   - Since many markers have same coordinates, Marker Cluster object was used.

5. Mouse position indicator was added
   - To identify locations' coordinates and calculate launch sites proximity to railway, highway, coastline, etc.

6. Created a Marker and Polyline
   - To show the distance between the nearest coastline and the site location.

7. Inserted markers for highway, railway and city center
   - To find out if the launch sites are in particular distance from these locations.

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/f249510bd91128ed87bec3887c3bcaa6133b3662/Notebooks/lab-jupyter-launch-site-location-v2.ipynb

Build a Dashboard with Plotly Dash

To be able to obtain some insights to answer the following five questions, an interactive dashboard was built using Plotly Dash.

1. Which site has the largest successful launches?
2. Which site has the highest launch success rate?
3. Which payload range(s) has the highest launch success rate?
4. Which payload range(s) has the lowest launch success rate?
5. Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/3859aaba1fb5e5b6cb8148596b367a34b97310da/Applications/spacex-dash-app.py

Build a Dashboard with Plotly Dash

Added a Launch Site Drop-down

- To see overall and **each site's success rate**.

Added a Range Slider

- To select **payload mass range** and find if it is correlated to success rate.

Added Two Callback Functions

- 1. To render a **pie chart** visualizing launch success counts, based on the selected **launch site from drop-down**.
- 2. To render a **scatter plot** visualizing launch success counts, based on **payload mass** and **booster versions**.

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/3859aaba1fb5e5b6cb8148596b367a34b97310da/Applications/spacex-dash-app.py

Predictive Analysis (Classification)

1. Imported libraries and defined Confusion Matrix function.

2. Loaded the Dataframe and created an array containing ‘Class’ column data.
   - Assign it as “Y” variable to be predicted.

3. Assigned the rest of the data as “X” and preprocessed using standardization.

4. Split the data into training and testing data using train_test_split.

5. Four algorithms were used to train the model.
   1. Linear Regression 2. SVM 3. Decision Tree 4. KNN

6. A GridSearchCV object was created for each of four, to find the best parameters.

7. The best parameters and the accuracy on the training and test data were identified.

8. Finally, confusion matrices and accuracy scores were compared to select the best model.

GitHub URL:
https://github.com/eddie-bhl/IBM-Capstone-Project-SpaceX/blob/5c8269bd18d3f5ffb95181cfcf07f0f4377208b1/Notebooks/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

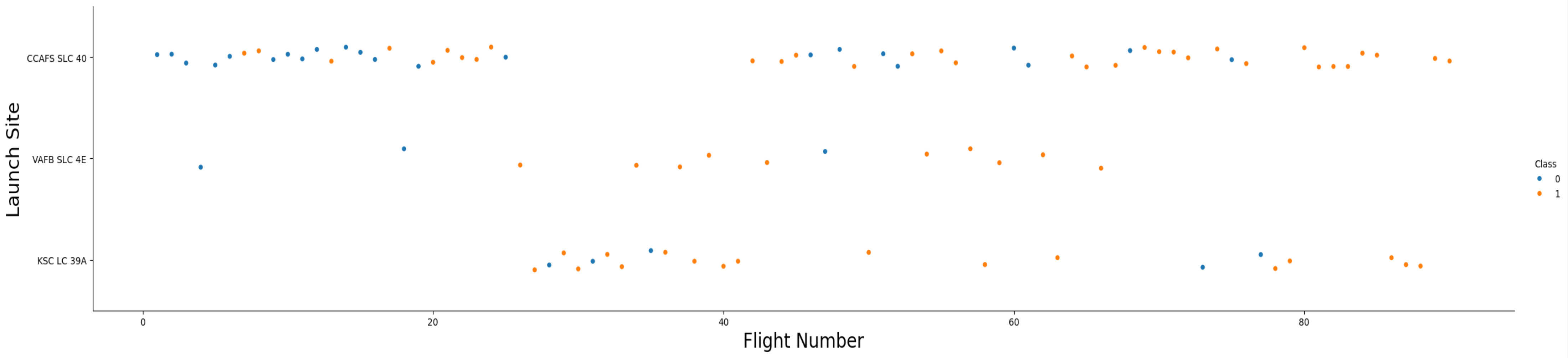
Results

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

Section 2

Insights drawn from EDA

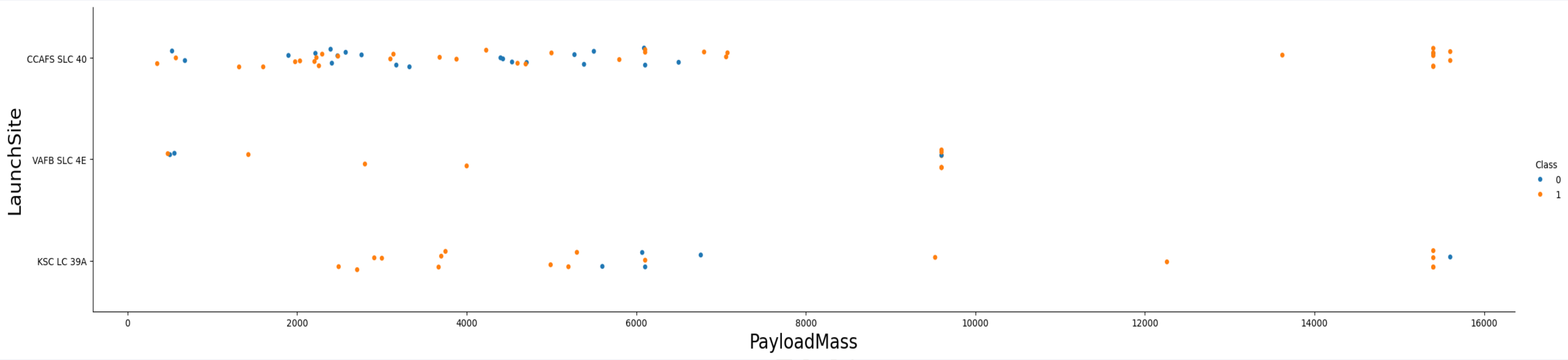
Flight Number vs. Launch Site



The chart reveals several insights about launch performance across different sites.

- During the initial 20 flights, the **CCAFS SLC-40** site was primarily used, with only two launches conducted from **VAFB SLC-4E**. Most of these early missions were unsuccessful, largely due to their experimental nature.
- Overall, **CCAFS SLC-40** recorded the highest number of launches, while **VAFB SLC-4E** had the fewest, with 13 in total; three of which failed. Following the first 20 flights, the success rate improved significantly across all sites.
- Notably, every launch after flight number 80 was successful, all taking place at **CCAFS SLC-40** and **KSC LC-39A**.

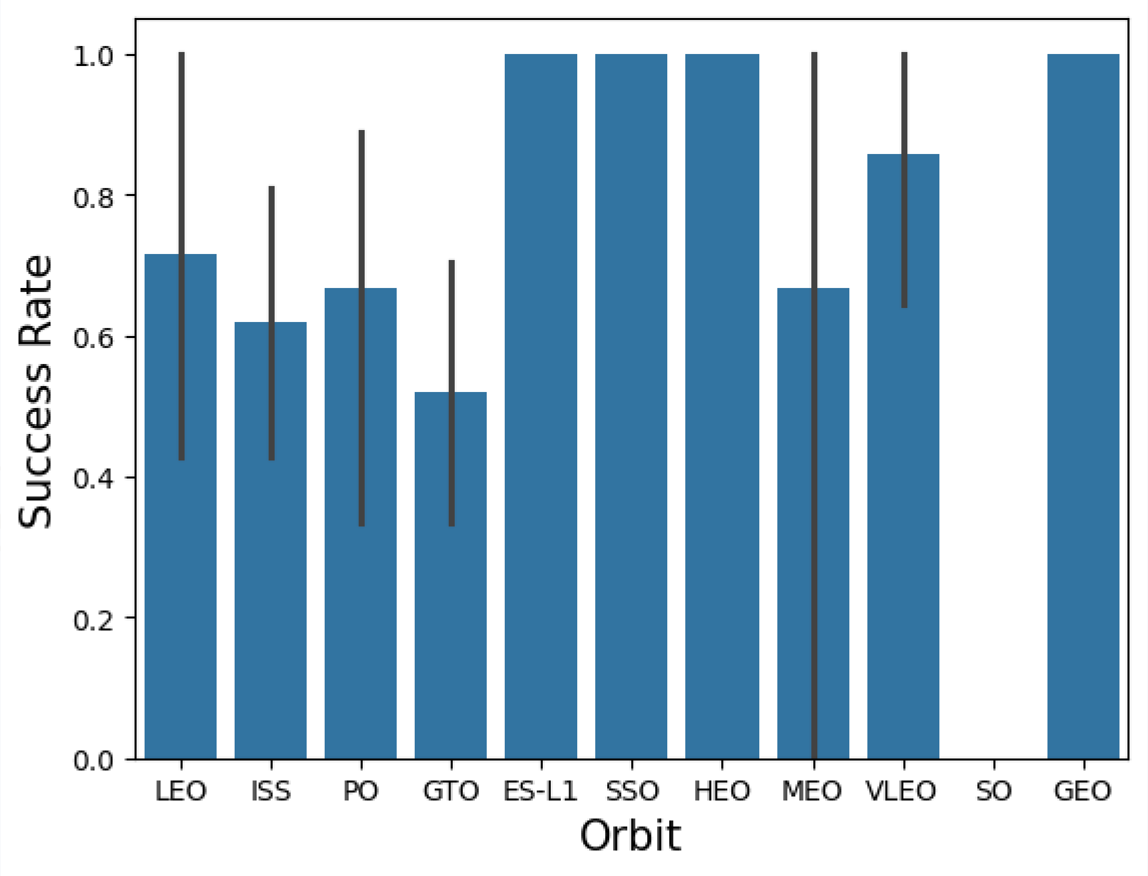
Payload vs. Launch Site



- More than **80%** of launches had payload mass **less than 8000kg**. That is the range where **most of the failures** occurred too.
- There were a total of **17** launches with payload mass of between **8000kg and 16000kg** of which, only **two** in sites **VAFB SLC 4E** and **KSC LC 39A** were unsuccessful.
- The results for site **CCAFS SLC 40** are not consistent as payload mass increases, whereas for site **KSC LC 39A** all the launches with payload mass of less than around **5000kg** were successful.
- There are significant unsuccessful attempts at payload mass between **5000kg and 6000kg**. Therefore, there is a problem in that range.

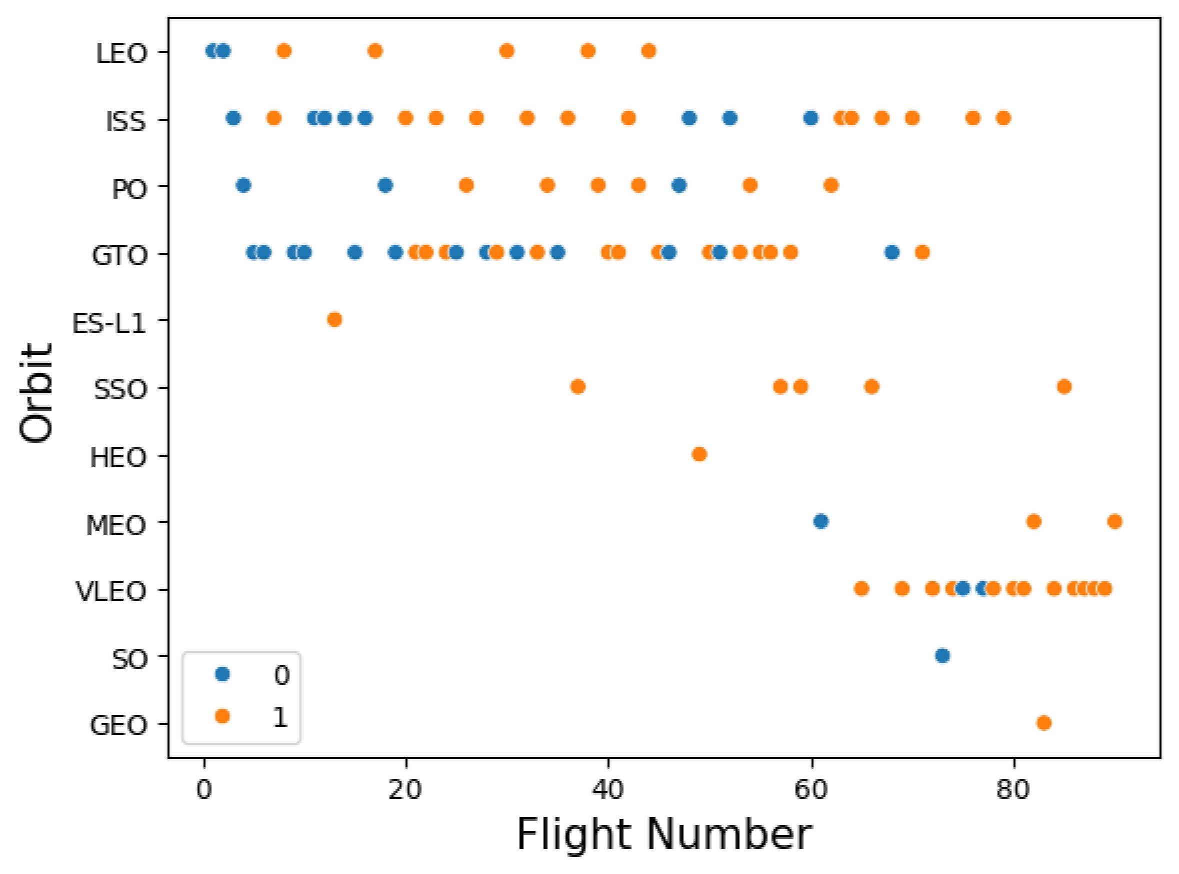
Success Rate vs. Orbit Type

- This chart represents success rate in each orbit the SpaceX rockets went through.
- Orbits **ES-L1**, **SSO**, **HEO** and **GEO** have the highest success rate of 100% among others.
- In this chart, **data are not normalized** and the number of launches in every orbit is needed to determine the significance rate.



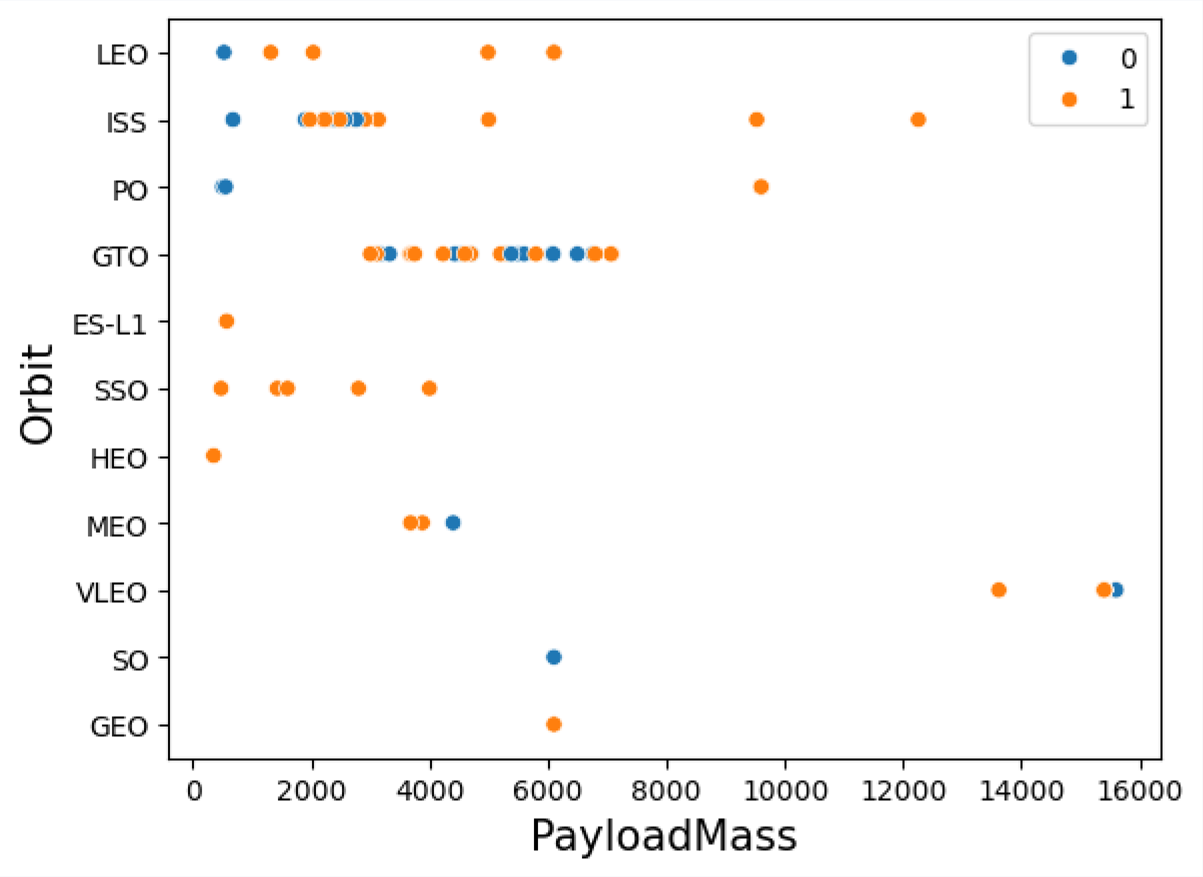
Flight Number vs. Orbit Type

- This chart represents the number of successful and unsuccessful flights in every orbit.
- It is observable that data are not normalized and around 70% of the flights occurred in **ISS**, **GTO** and **VLEO** orbits.
- The first 20 launches account for most of the failures. After 60 flights, there is a near consistent trend of success in every orbit.
- After 20 flights, there is a significant increase in successful flights.



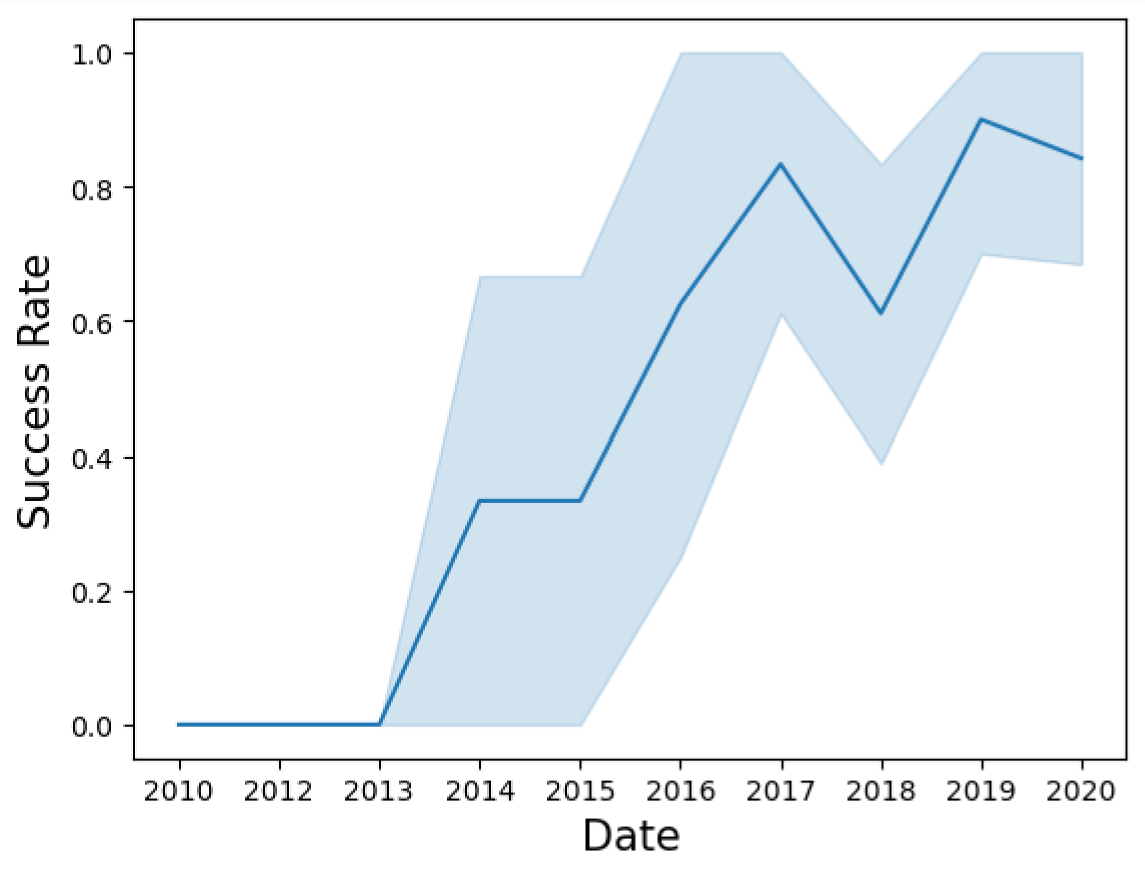
Payload vs. Orbit Type

- Most of the launches have less than 8000kg payload mass.
- It is notable that every flight in **SSO** was successful, but the majority of flights were taken place in **ISS** and **GTO** orbits.
- Most of the failures had payload mass of **500kg** and around **6000kg**.
- Also, launches with payload masses of **8000kg** and greater, have higher chance of success.



Launch Success Yearly Trend

- The chart displays the success rate from 2010 to 2020, with shaded areas indicating variability.
- From 2010 to 2013, the success rate remained at zero, suggesting no progress.
- Then it generally increases over time due to implementing new technology with a drop in 2018.



All Launch Site Names

- There are only 4 launch sites used.

```
%sql select "Launch_Site" from SPACEXTBL group by "Launch_Site"
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql Select * from SPACEXTBL where "Launch_Site" Like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- This query finds 5 records where launch sites begin with `CCA`.

Total Payload Mass

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where "Customer"='NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

- This query calculates the total payload carried by boosters from NASA.

Average Payload Mass by F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where "Booster_Version" like 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- This query calculates the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
%sql select MIN(Date) from SPACEXTBL where "Landing_Outcome"='Success (ground pad)'
```

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

MIN(Date)
2015-12-22

- This query finds the date of the first successful landing outcome on ground pad.
- Since flights started in 2010, this means that it took almost six years until the first rocket succeeded.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql select "Booster_Version" from SPACEXTBL
    where "Landing_Outcome"='Success (drone ship)' AND PAYLOAD_MASS__KG_ between 4000 and 6000;
```

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

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

- This lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000kg but less than 6000kg.
- Since there are only 4 entities output, this suggests that most of unsuccessful attempts had a payload mass between 4000kg and 6000kg

Total Number of Successful and Failure Mission Outcomes

```
%sql select "Mission_Outcome",count("Mission_Outcome") from SPACEXTBL group by "Mission_Outcome"
```

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

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

- Here is the total number of successful and failure mission outcomes.
- We remember that we had many failures in the **landing outcomes.** However, most of the flights had successful **mission outcomes. Be careful not to confuse those terms.**

Boosters Carried Maximum Payload

```
%sql select "Booster_Version" from SPACEXTBL where PAYLOAD_MASS__KG_=(select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

- This is the list of the names of the boosters which carried the maximum payload mass. A subquery has been used in this SQL example.

2015 Launch Records

```
%%sql select substr(Date,6,2) as month, substr(Date,0,5) as year, "Booster_Version", "Launch_Site", "Landing_Outcome"
    from SPACEXTBL where "Landing_Outcome" like 'Fail%' AND substr(Date,0,5)='2015'
```

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

month	year	Booster_Version	Launch_Site	Landing_Outcome
01	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- This query lists the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- Since SQLite does not support month names, we need to use **substr(Date, 6,2)** as month to get the months and **substr(Date,0,5)='2015'** for year.

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

```
%%sql select Date, count("Landing_Outcome"), "Landing_Outcome" from SPACEXTBL where Date between '2010-06-04' and '2017-03-20'
    group by "Landing_Outcome" order by Date DESC
```

* sqlite:///my_data1.db
Done.

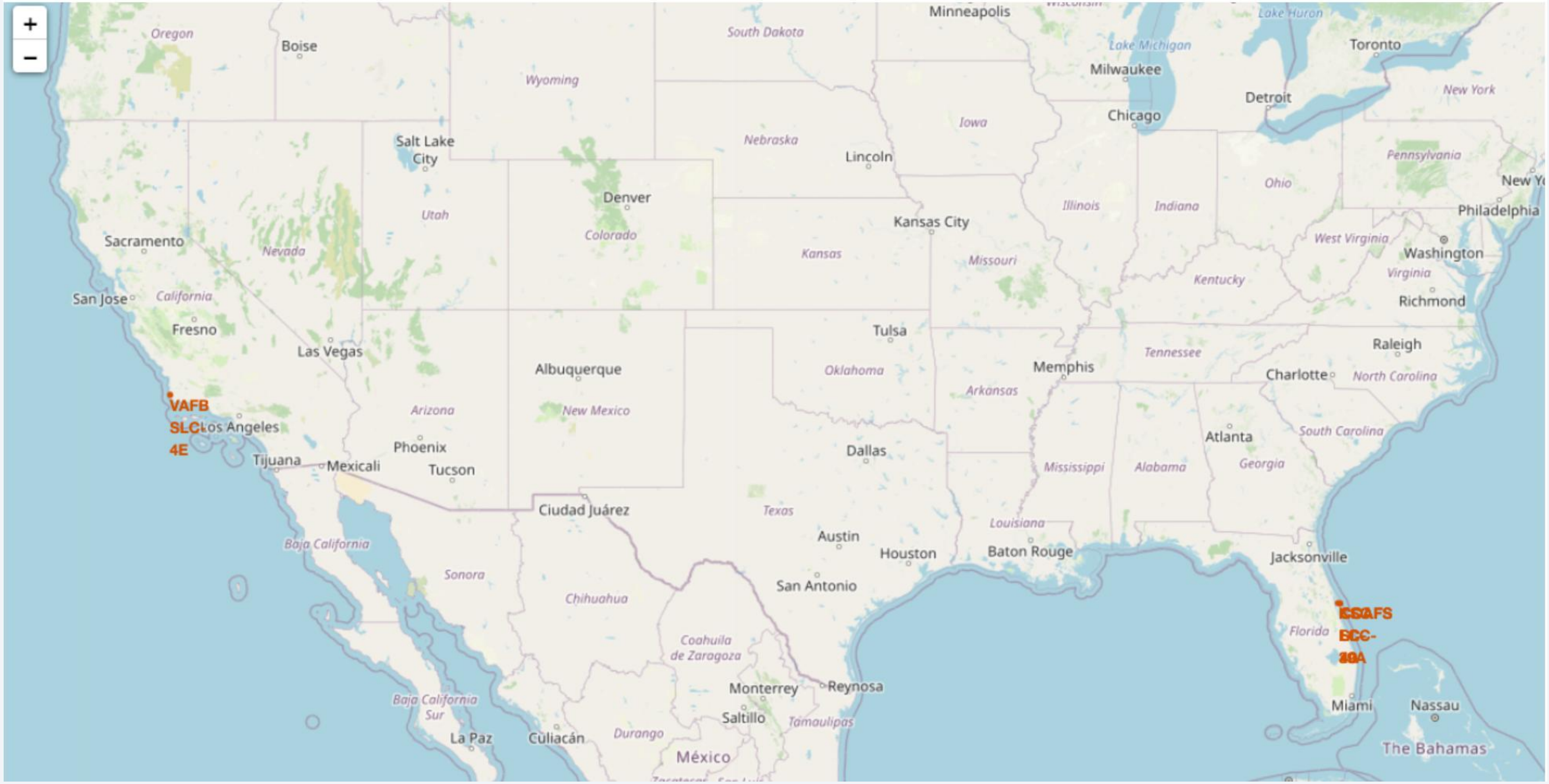
Date	count("Landing_Outcome")	Landing_Outcome
2016-04-08	5	Success (drone ship)
2015-12-22	3	Success (ground pad)
2015-06-28	1	Precluded (drone ship)
2015-01-10	5	Failure (drone ship)
2014-04-18	3	Controlled (ocean)
2013-09-29	2	Uncontrolled (ocean)
2012-05-22	10	No attempt
2010-06-04	2	Failure (parachute)

- This query ranks 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.
- It should be noted that landing outcomes with **“No attempt”** value, should be excluded from calculations for outcome prediction.

Section 3

Launch Sites Proximities Analysis

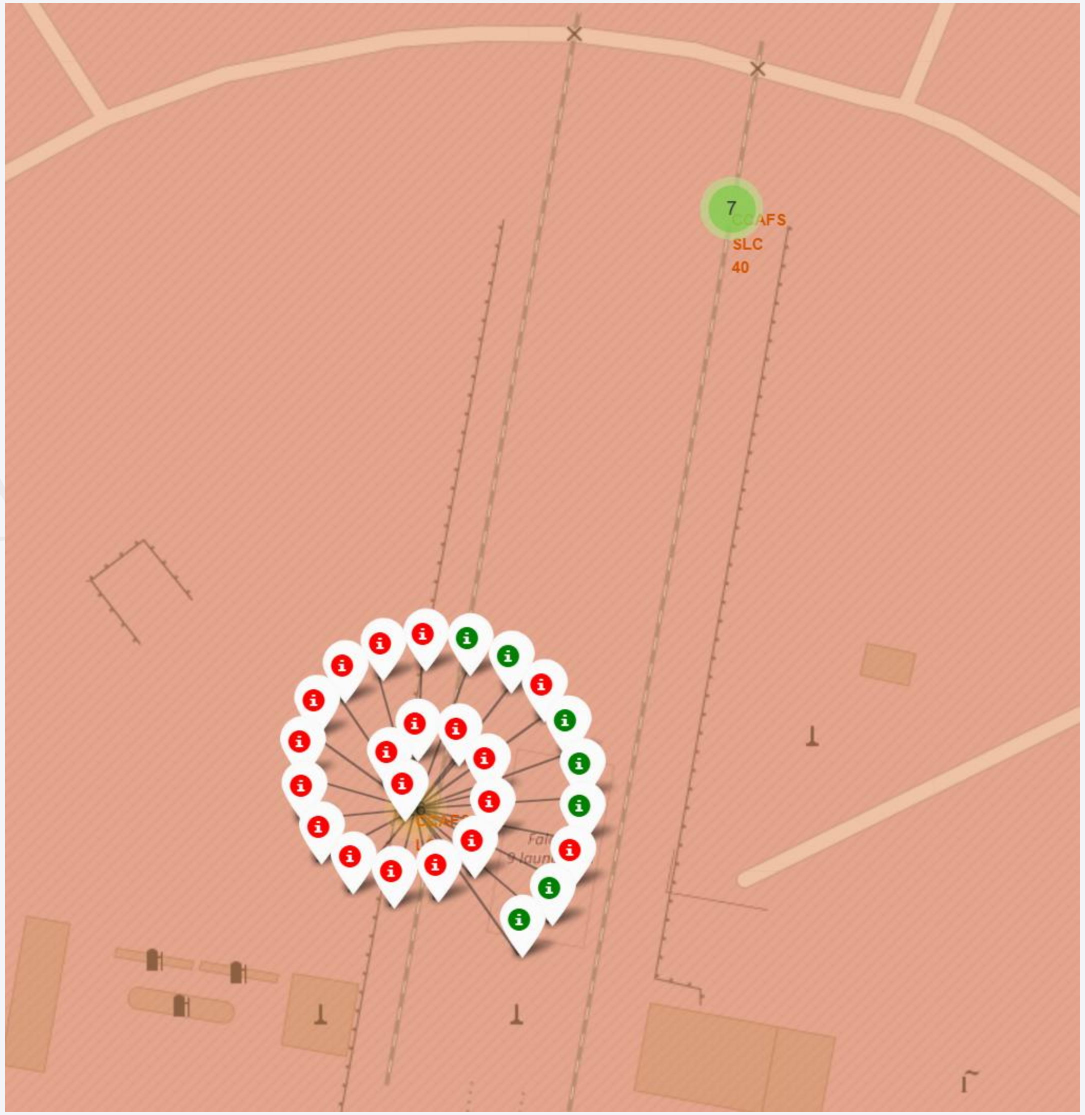
Launch Sites Markers



- As can be seen, there are three launch sites in **Eastern US** and one launch site in **Western US.**

Success/Failed Launches For Each Site

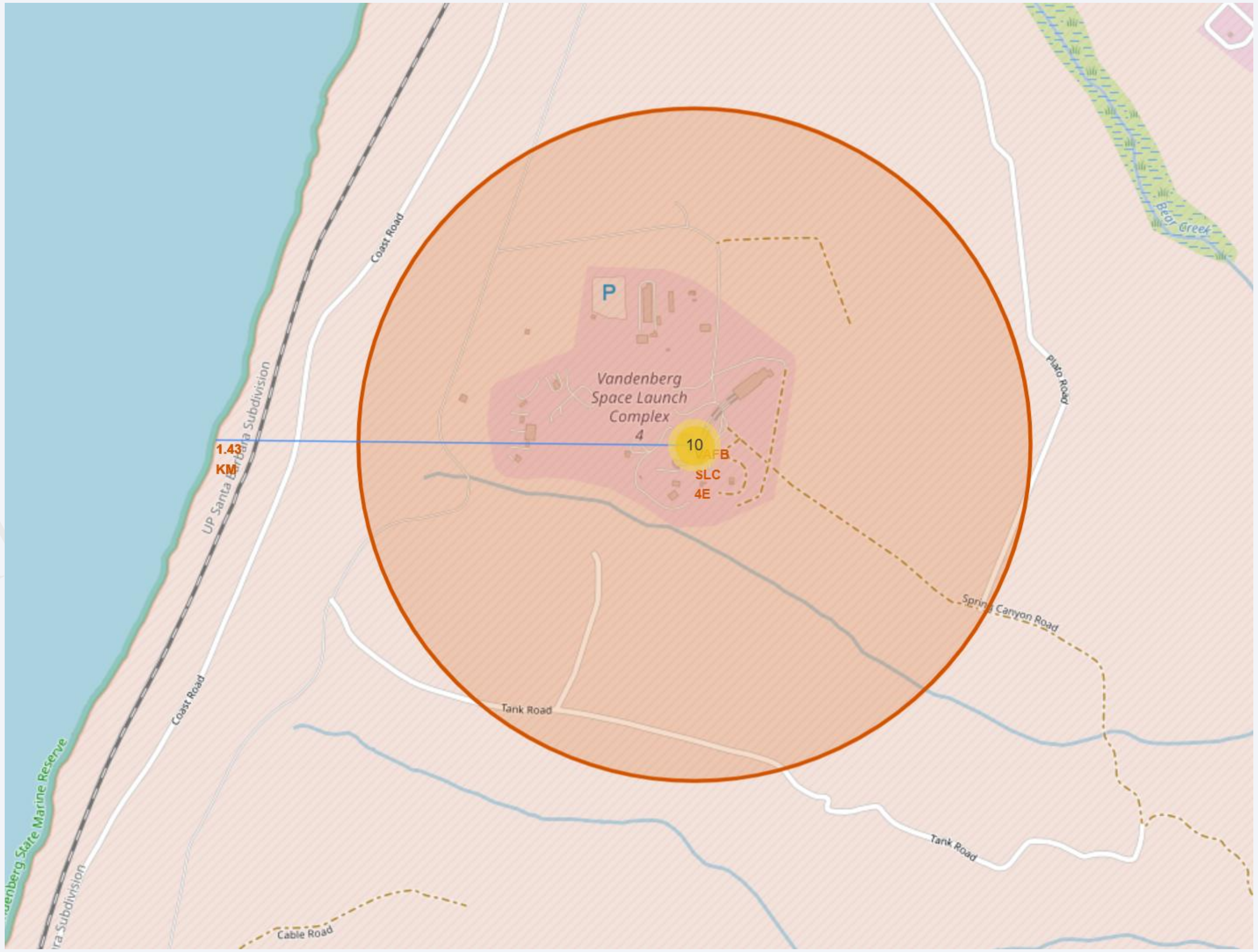
- Here is the color-labeled launch outcomes in marker clusters on the map.
- From the colors, we can easily identify each site's success rate.
- Folium map library were used.



Distances Between a Launch Site To Its Proximities

- Distances to different map objects such as highways, railways, airports, etc. can be measured by defining a calculation function in Python which uses mouse pointer coordinates to calculate the distance between two points.

- In this example, the closest distance between the center of a launch site and coastline has been measured which is 1.43 KM.

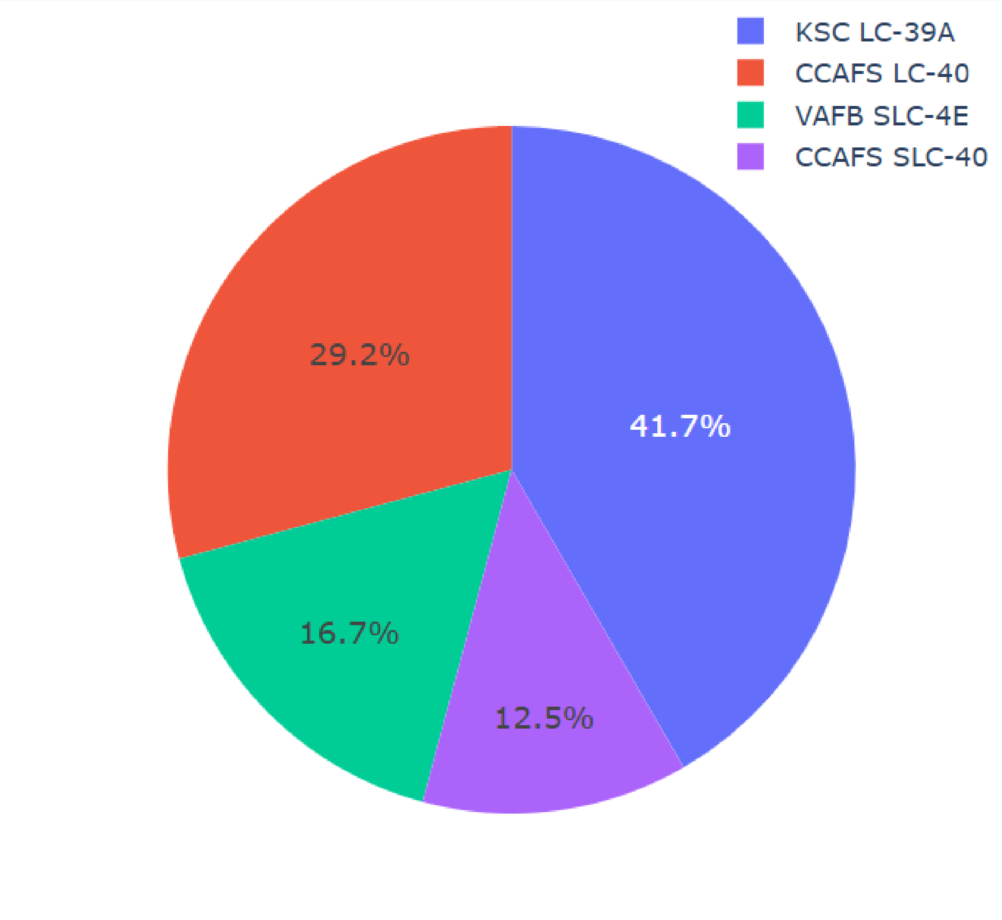


Section 4

Build a Dashboard with Plotly Dash

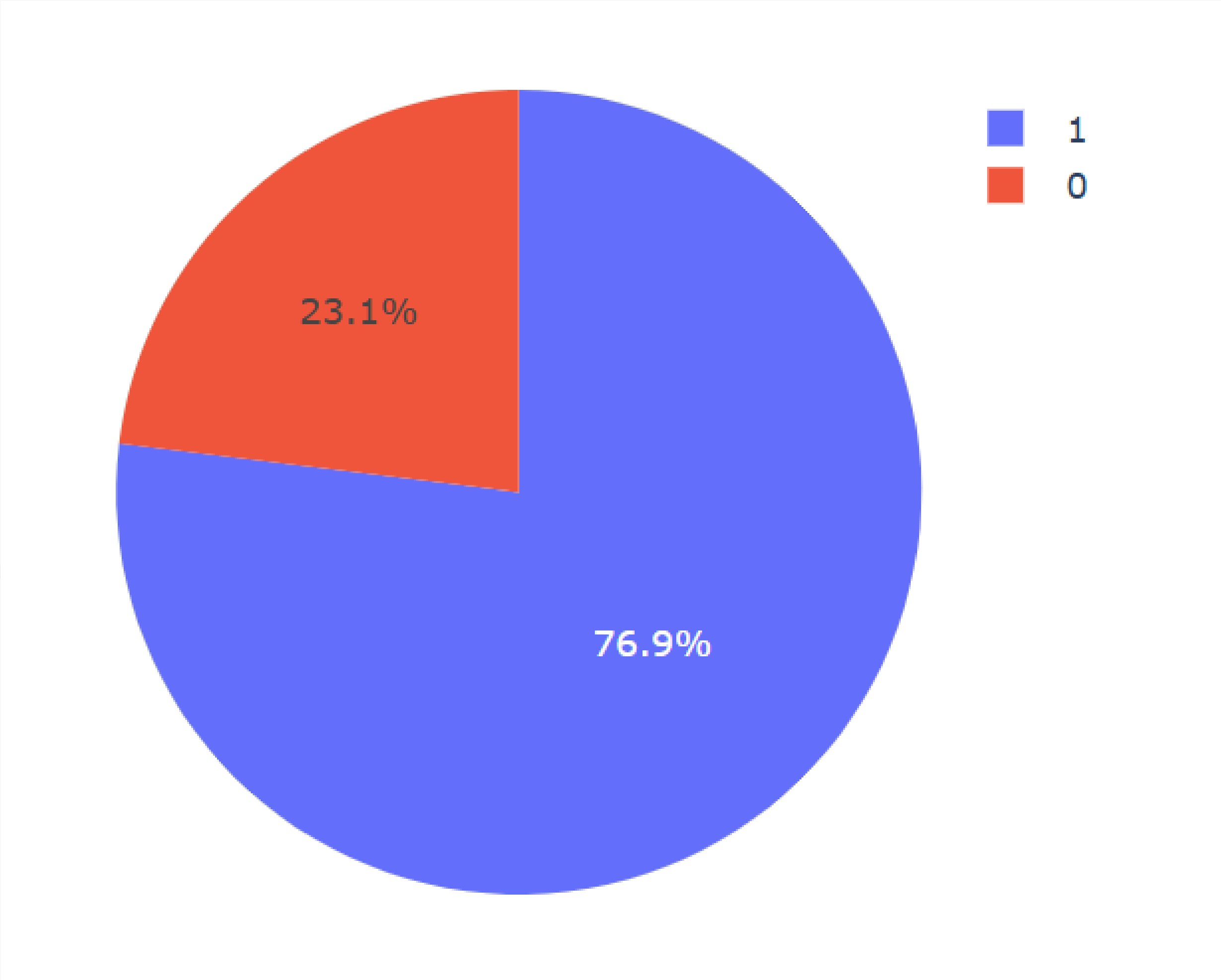
Total Successful Launches by Site

- Here is the screenshot of launch success count for all sites, in a pie chart using Plotly Dash web application.
- It can be seen that launch site **“KSC LC-39A”** has the highest success rate among the others.

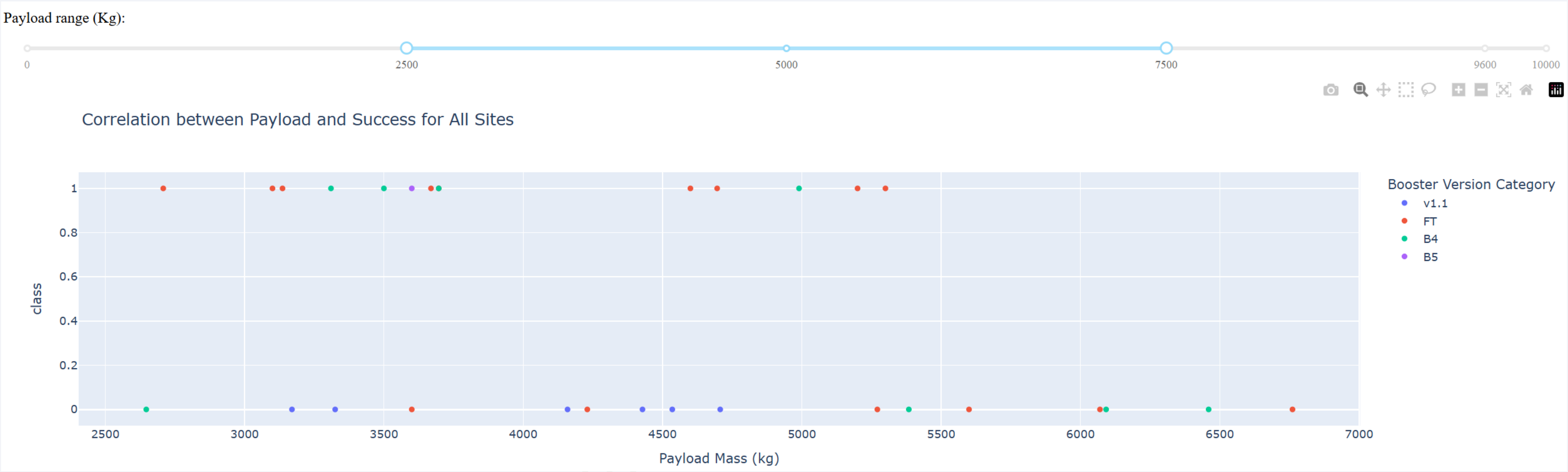


Success Rate For Site “KSC LC-39A”

- This pie chart represents the success rate for **“KSC LC-39A”** which has the highest success rate among the others.



Payload Mass and Success Rate Correlation



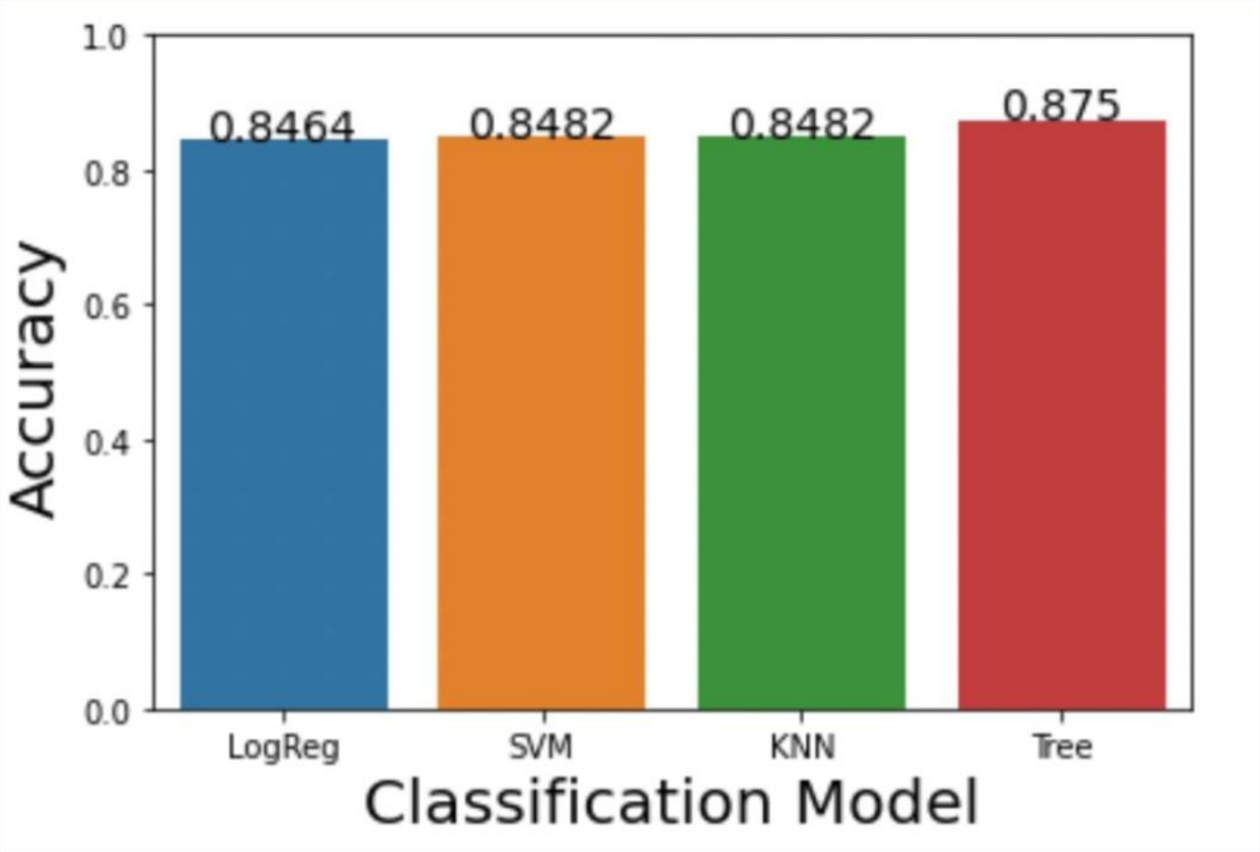
- I have also developed a scatter plot showing correlation between **Payload Mass** and **success rate** which can be filtered using **Payload Range slider**.
- This plot tells us which **booster version category** has the highest success rate in different payload mass ranges.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

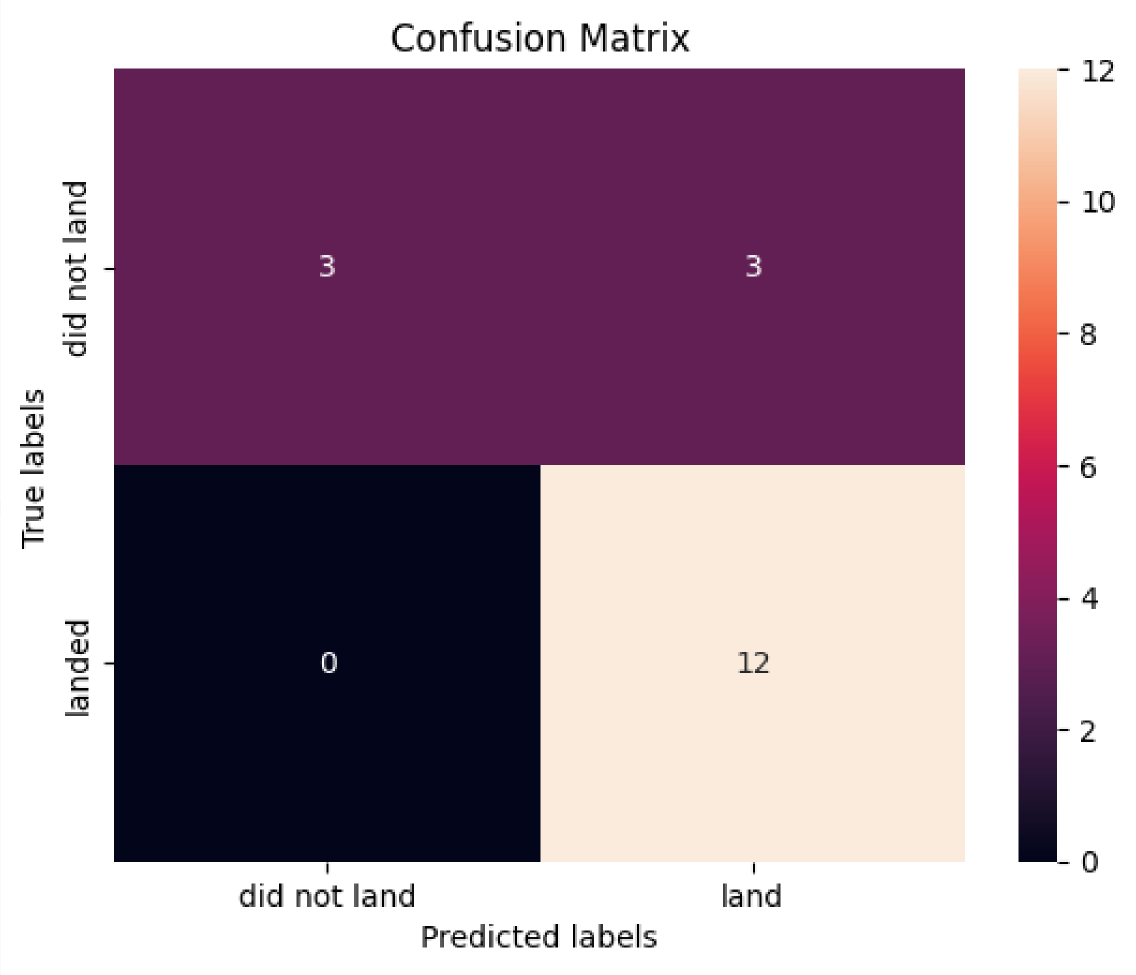
- Here is the **average cross-validation** model score (on the training dataset ONLY) for all classification models, represented in a bar chart.
- The steps to build models were discussed earlier.
- As can be seen, Decision Tree model has the highest average classification accuracy.



Confusion Matrix

- In our case, all of the models had the same prediction scores on test data for the reasons that will be discussed later in conclusion.

- Precision : 1.0
- Recall: 0.5
- Accuracy: 83.33%



Conclusions

- In the predictive analysis stage, the models showed same amount of prediction scores mainly due to **class imbalance** and **data inconsistencies**. As mentioned before, about 80% of the data lies within payload mass of less than 8000kg. Furthermore, landing outcomes with “no attempt” value should be excluded from the algorithms.
- To improve model accuracy, external factors such as weather or different technical components should be considered.
- With the information we gained from predictive analysis, we now have found out which sites with what properties have the highest success rates, so that we could use those to predict if the launches will land successfully, which accordingly saves us a lot of money and time.

Thank you!