IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Furkan Durmus
09.10.2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction

- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

SpaceX stands as a game-changer in the realm of space exploration, redefining the economics of space missions. Their groundbreaking approach to cost-efficiency is primarily showcased through their Falcon 9 rocket launches, which are priced at a competitive 62 million dollars, a stark contrast to the hefty 165 million dollar price tags of other providers. The cornerstone of this cost reduction is SpaceX's ingenious strategy of recycling the first stage of their rockets.

By successfully landing and reusing this stage for subsequent missions, they've set the stage for even more cost savings in the future. From the perspective of a data scientist working for a startup in competition with SpaceX, the main objective of our project is to devise a machine learning model that predicts the landing outcomes of these rockets' first stages. Such a project is pivotal, as it will equip us with insights to strategically price our own launches in a bid to challenge SpaceX's market dominance. Key challenges encompass:

- Pinpointing the determinants that sway the landing results.
- Understanding the intricate interplay between variables and their impact on the landing.
- Deciphering the optimal conditions that bolster the chances of a successful touchdown.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology

  SpaceX API and Web Scraping From Wikipedia

- Perform data wrangling

  One Hot Encoding data from Machine Learning

- Perform exploratory data analysis (EDA) using visualization and SQL

  For data visualization, I utilized various plotting libraries.

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models

  I utilized Logistic Regression, KNN, SVC, and DecisionTreeClassifier for the model building process in my analysis.

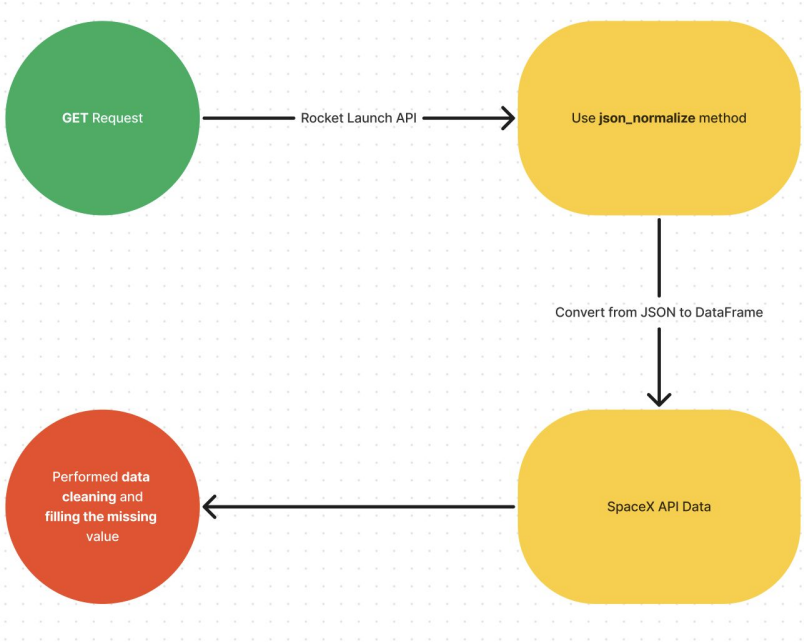
Data Collection

Data collection is pivotal in analytics, enabling informed decision-making through gathering precise information. In this analysis, we focused on SpaceX rocket data sourced from Wikipedia using both REST API and Web Scraping techniques.

The REST API method began with a 'GET' request, decoding the response into JSON, which was then transformed into a pandas dataframe using json_normalize(). Any inconsistencies or missing values in the data were addressed.

For web scraping, BeautifulSoup was utilized to extract launch records from Wikipedia's HTML tables, subsequently converting them into a structured pandas dataframe for in-depth analysis.

Data Collection – SpaceX API




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

```python
response = requests.get(url)
```

```python
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

```python
data = pd.json_normalize(response.json())
data.head()
```

```python
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Github URL: [Data Collection - SpaceX API]

Data Collection - Scraping

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

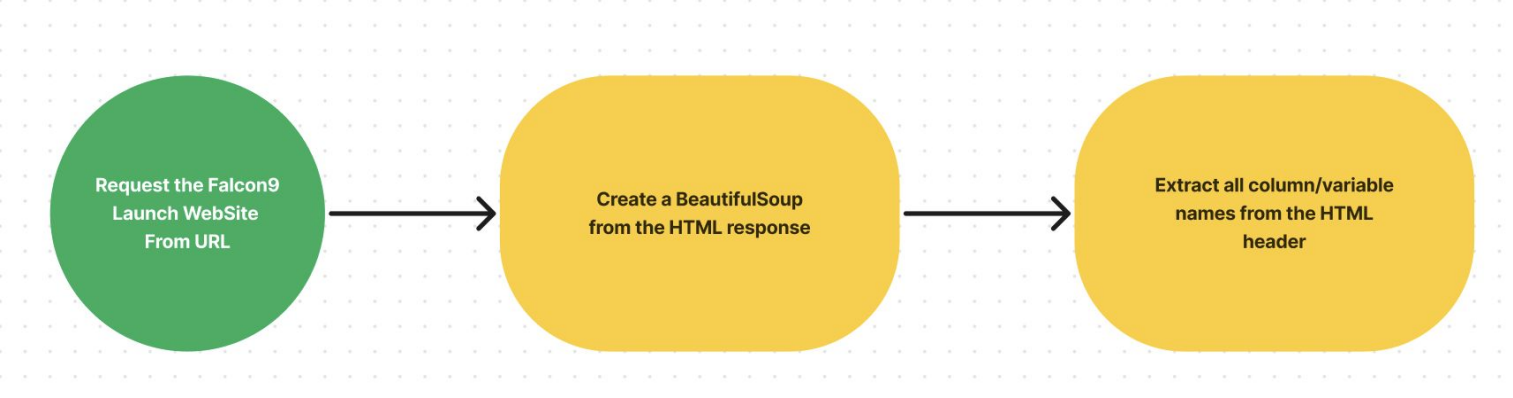
Python

```python
response = requests.get(static_url)

if response.status_code == 200:
    print(f"Success: {response.status_code}")

else:
    print(f"Failure {response.status_code}")
```

Python

```python
soup = BeautifulSoup(response.text, "html.parser")
```

Python



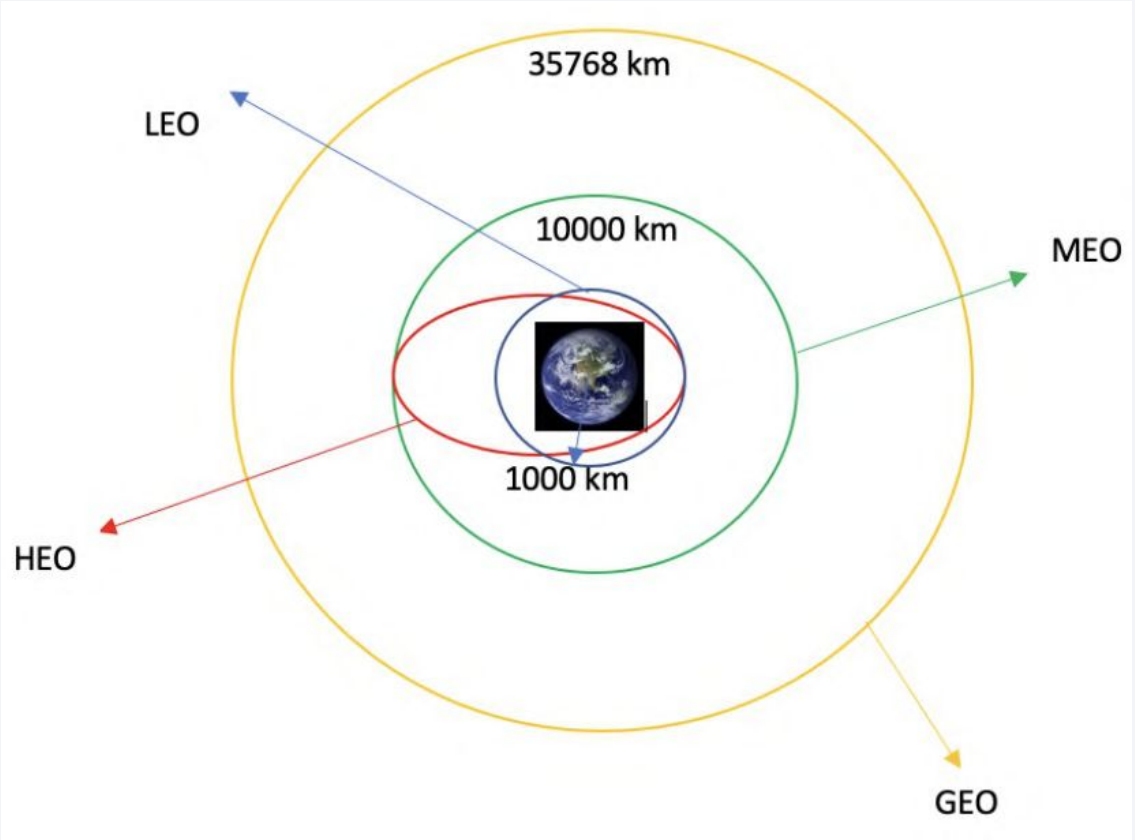
Github URL: [Data Collection - Scraping Process]

Data Wrangling

Data Cleaning involves refining and standardizing complex datasets to facilitate easier Exploratory Data Analysis (EDA). Initially, we'll determine the number of launches for each site.

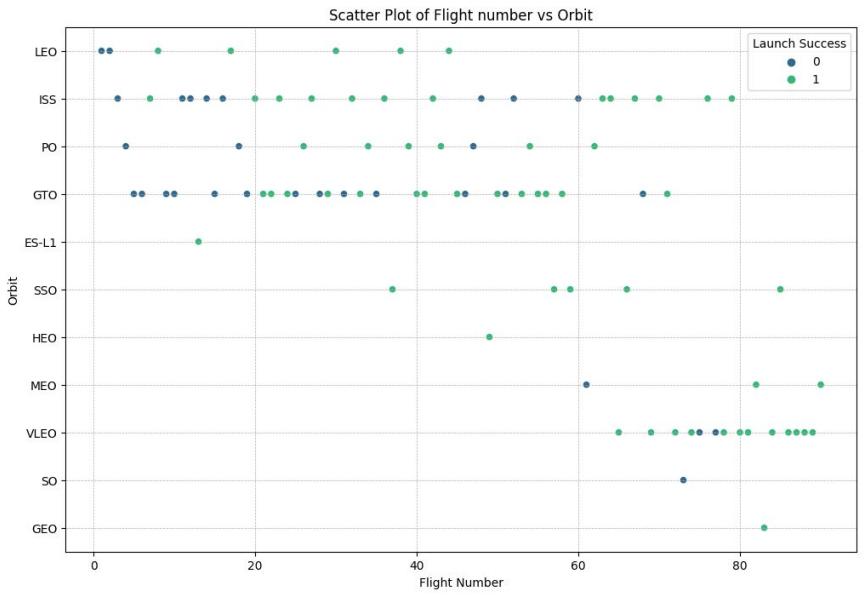
Subsequently, we will assess the frequency and results of missions for each orbit type.

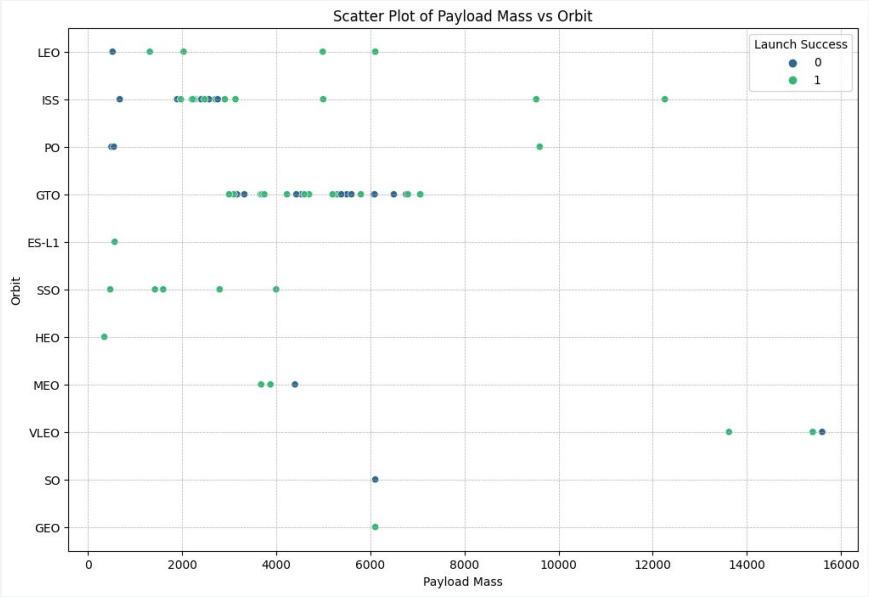
From the outcome column, we'll generate a label indicating the landing outcome, enhancing the ease of subsequent analysis, visualization, and ML tasks. Ultimately, the refined data will be saved to a CSV file.



Github URL: [Data Wrangling]

EDA with Data Visualization



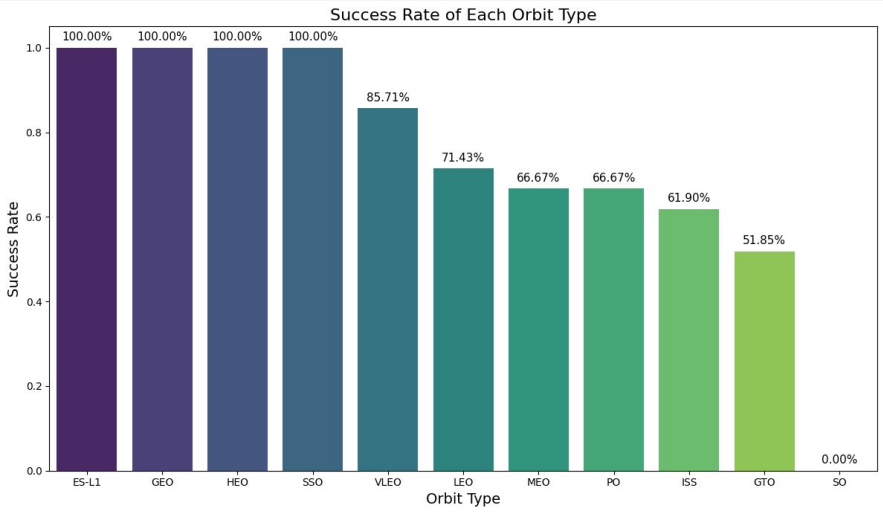


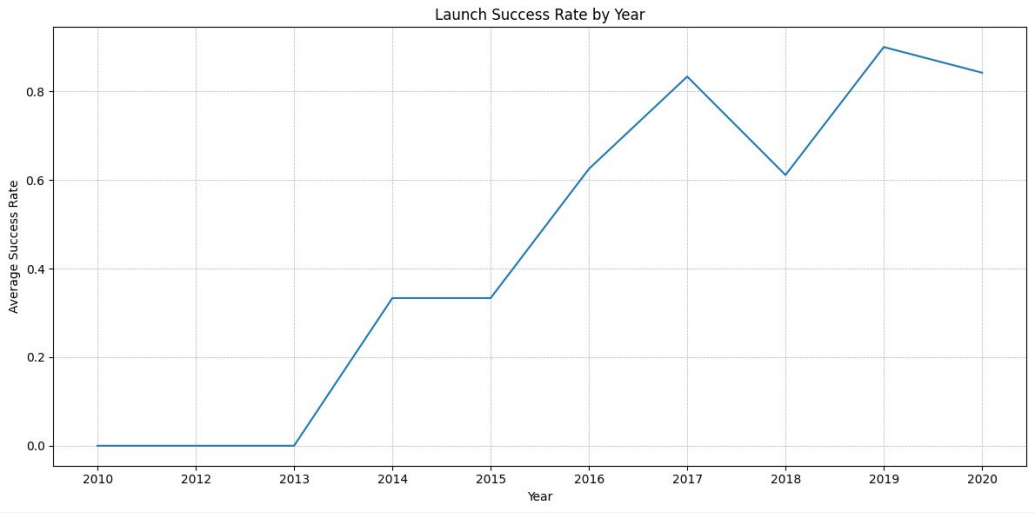
We first started by using scatter graph to find the relationship between the attributes such as between:

- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

Github URL: [EDA With Data Visualization]

EDA with Data Visualization





Once we get a hint of the relationships using scatter plot. We will then use further visualization tools such as bar graph and line plots graph for further analysis.

Bar graphs is one of the easiest way to interpret the relationship between the attributes. In this case, we will use the bar graph to determine which orbits have the highest probability of success.

We then use the line graph to show a trends or pattern of the attribute over time which in this case, is used for see the launch success yearly trend.

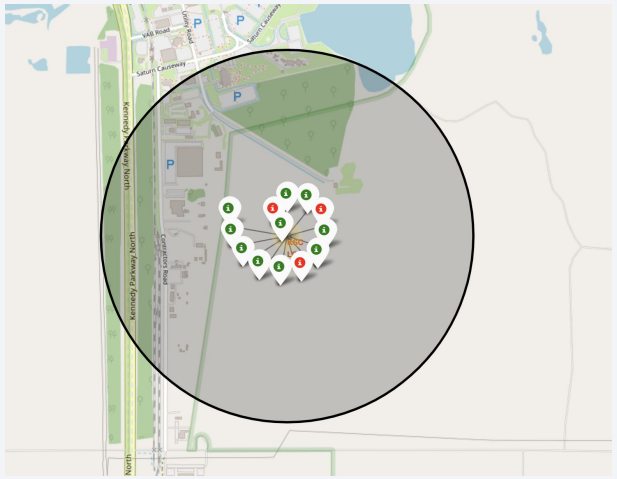
We then use Feature Engineering to be used in success prediction in the future module by created the dummy variables to categorical columns.

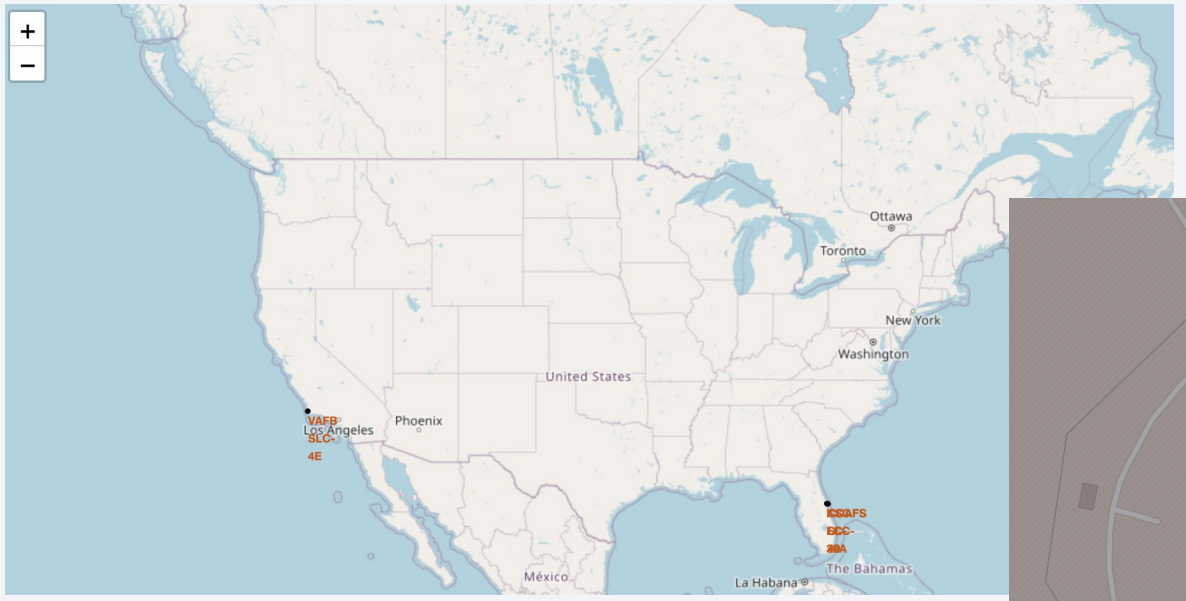
EDA with SQL

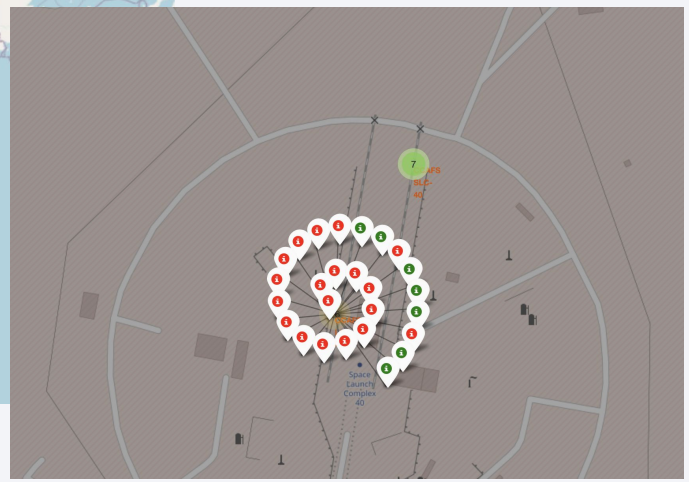
Using SQL, we had performed many queries to get better understanding of the dataset, Ex:

- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string ‘CCA’.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015.
- Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order.

Build an Interactive Map with Folium



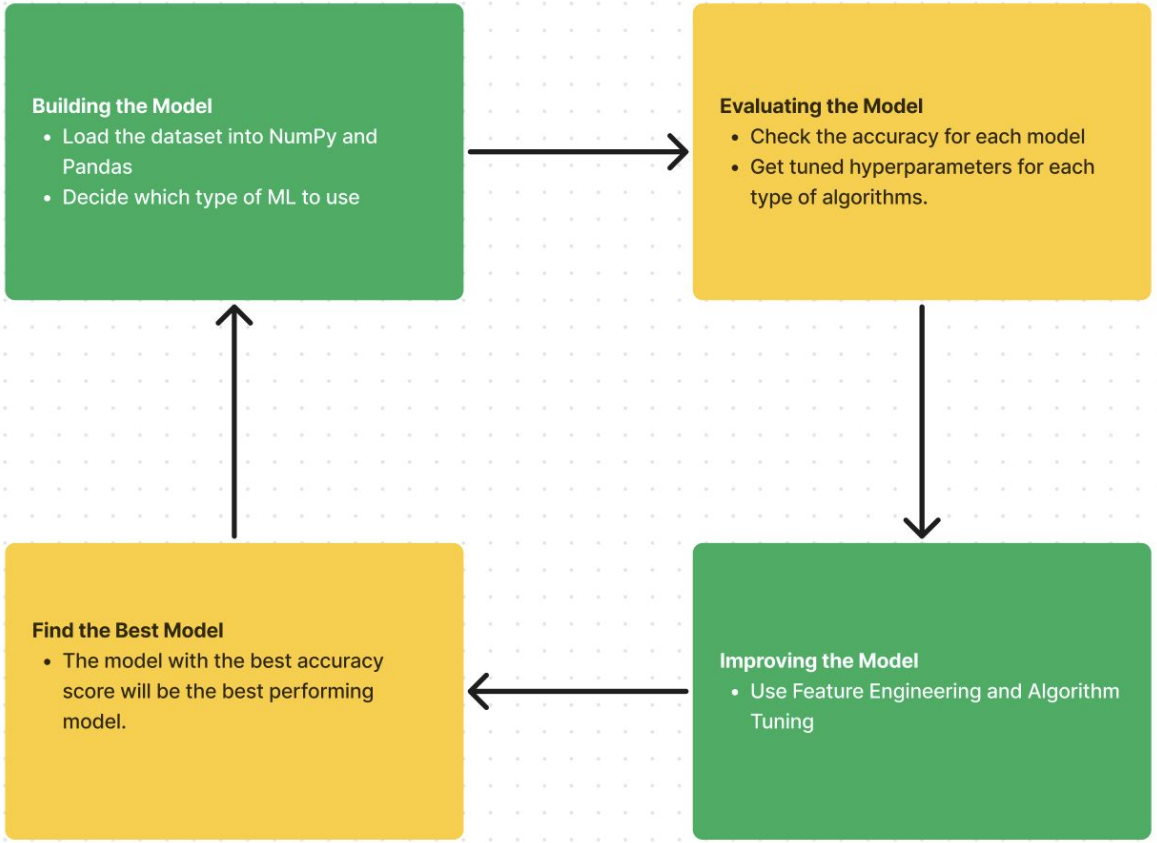




To visualize the launch data into an interactive map. We took the latitude and longitude
coordinates at each launch site and added a circle marker around each launch site with a
label of the name of the launch site.
We then assigned the dataframe **launch_outcomes(failure,success)** to classes 0 and 1 with
Red and **Green** markers on the map in **MarkerCluster()**.

Github URL: [Build an Interactive Map with Folium]

Predictive Analysis (Classification)



Github URL: [Predictive Analysis (Classification)]

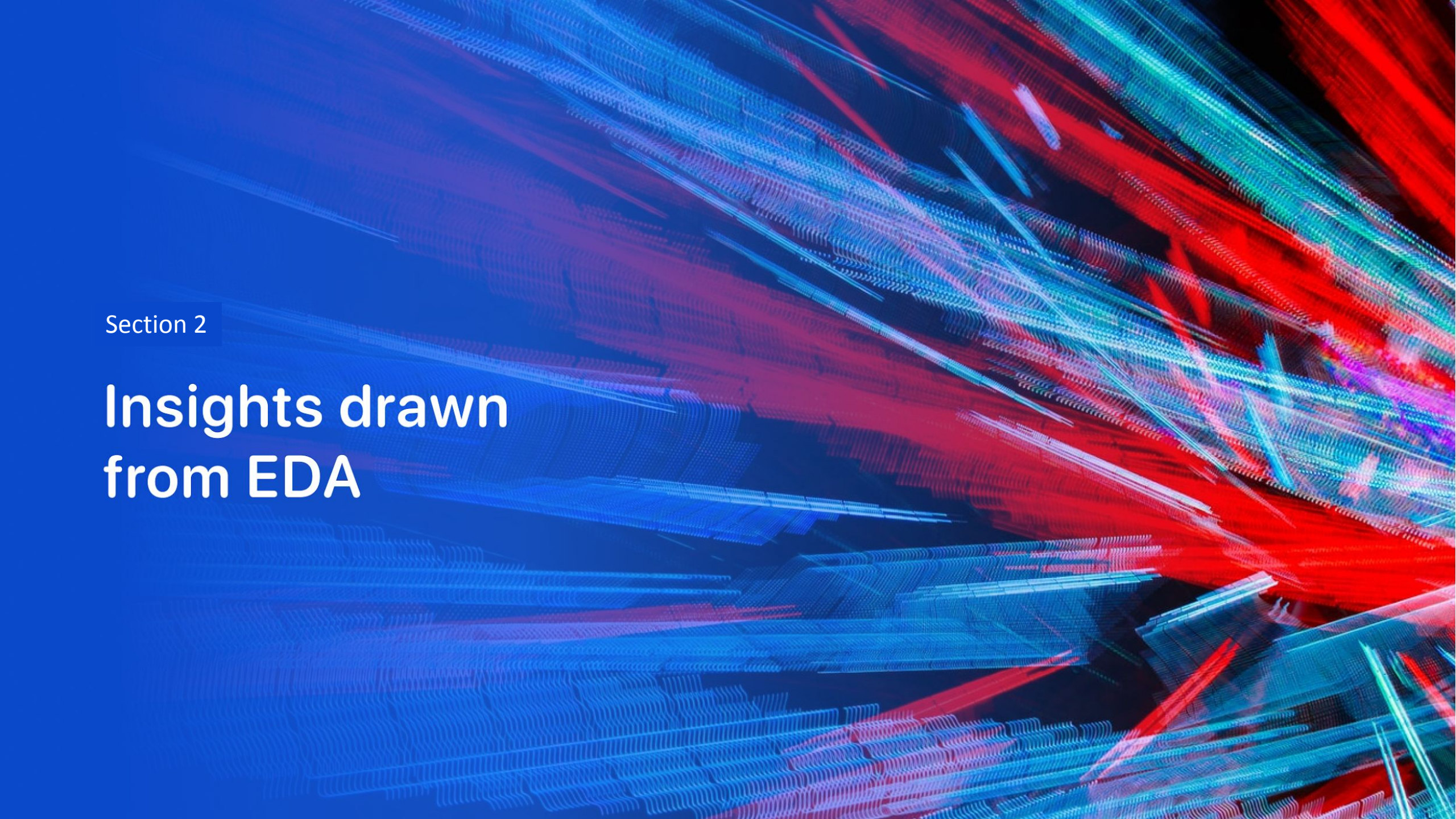
Results

The result will be categorized to 3 main result which is

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

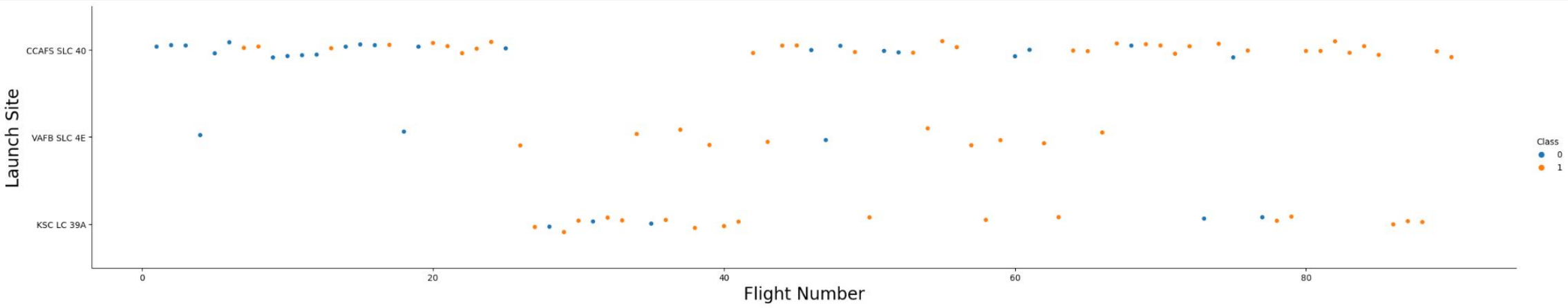
Section 2

Insights drawn from EDA



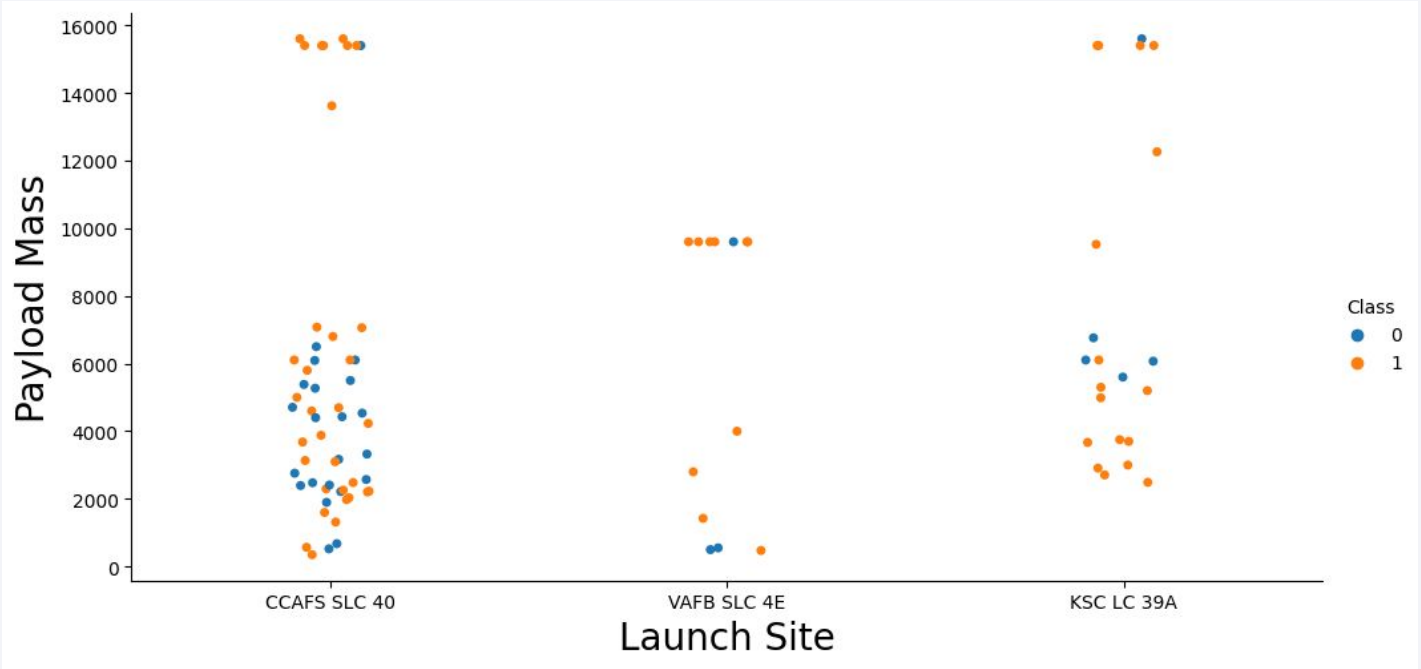
Flight Number vs. Launch Site

In the chart, the CCAFS SLC40 Launch Site appears to be the most successful. The yellow points indicate successful flights, while the blue ones indicate unsuccessful flights.



Payload vs. Launch Site

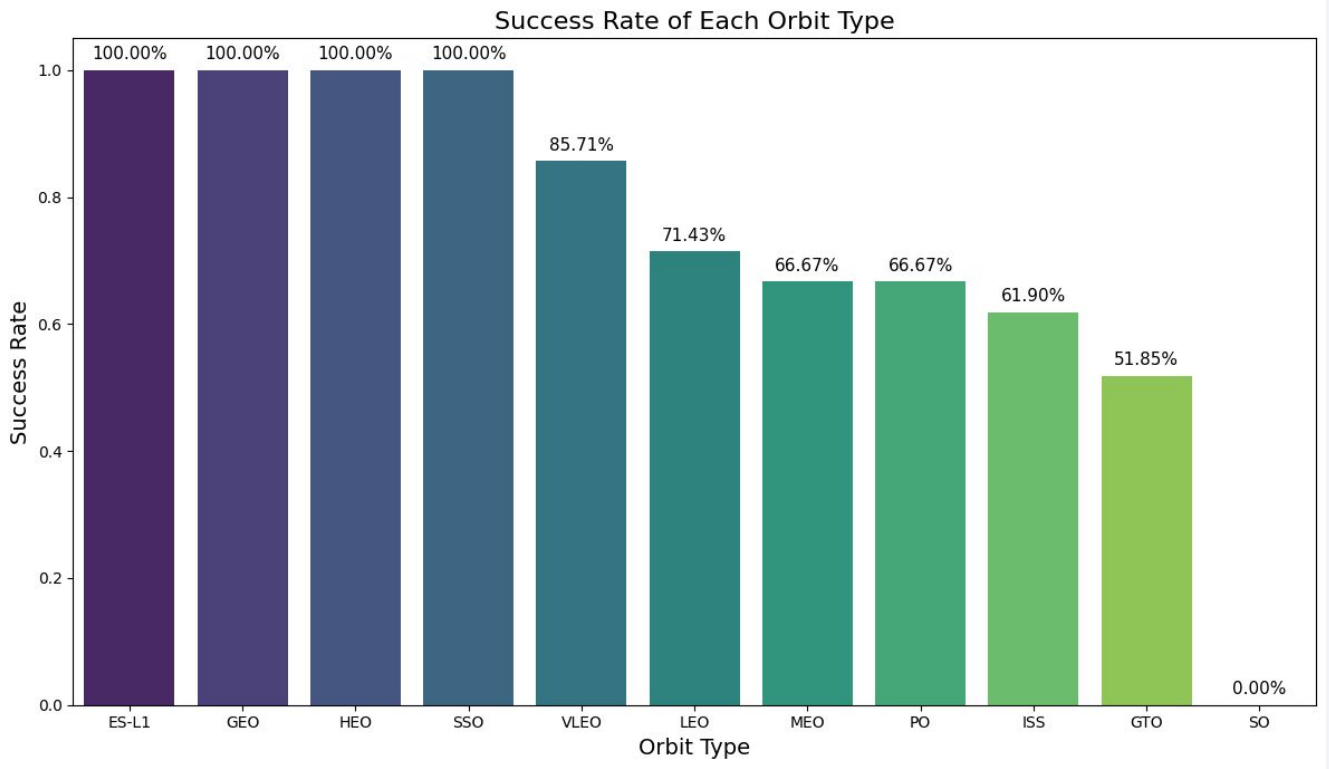
In the chart, it is evident that once the payload mass exceeds 7000kg, the probability of success significantly increases



Success Rate vs. Orbit Type

This figure depicted the possibility of the orbits to influences the landing outcomes as some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced 0% rate of success.

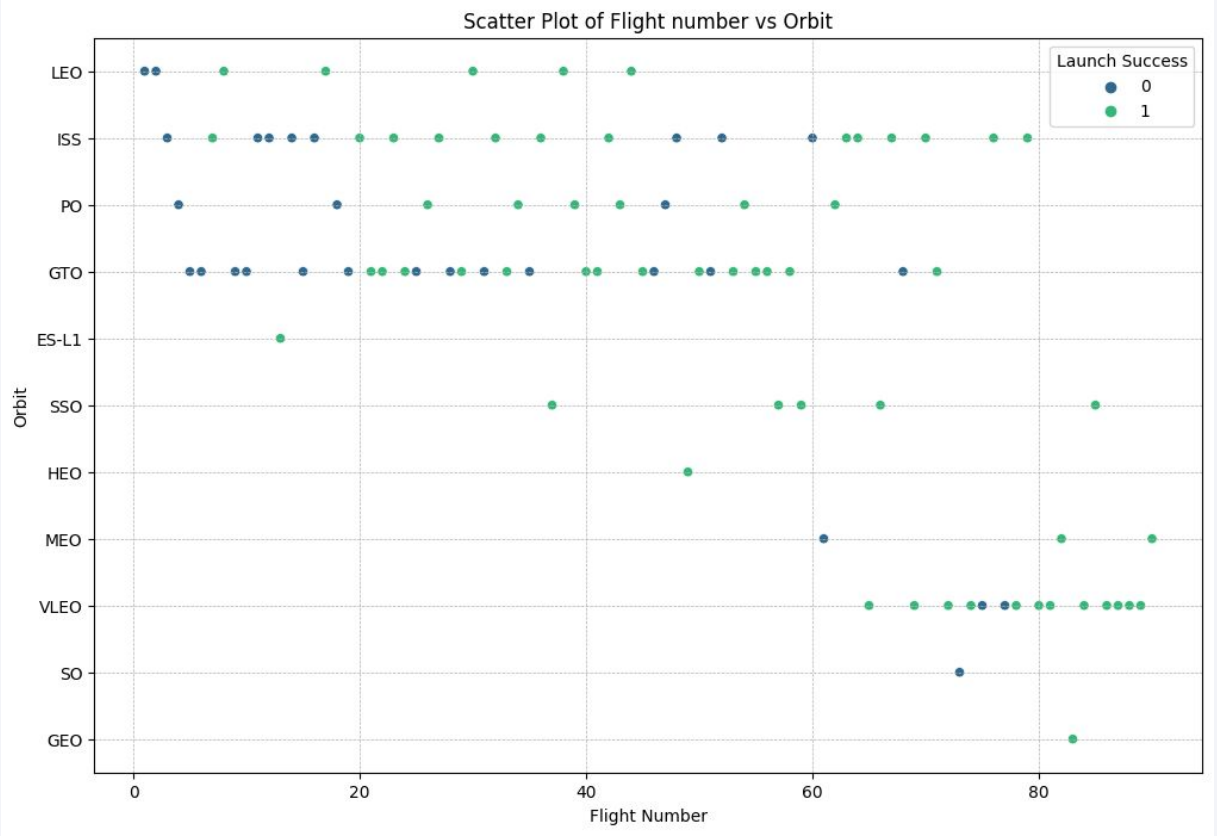
However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.



Flight Number vs. Orbit Type

This scatter plot shows that generally, the larger the flight number on each orbits, the greater the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes.

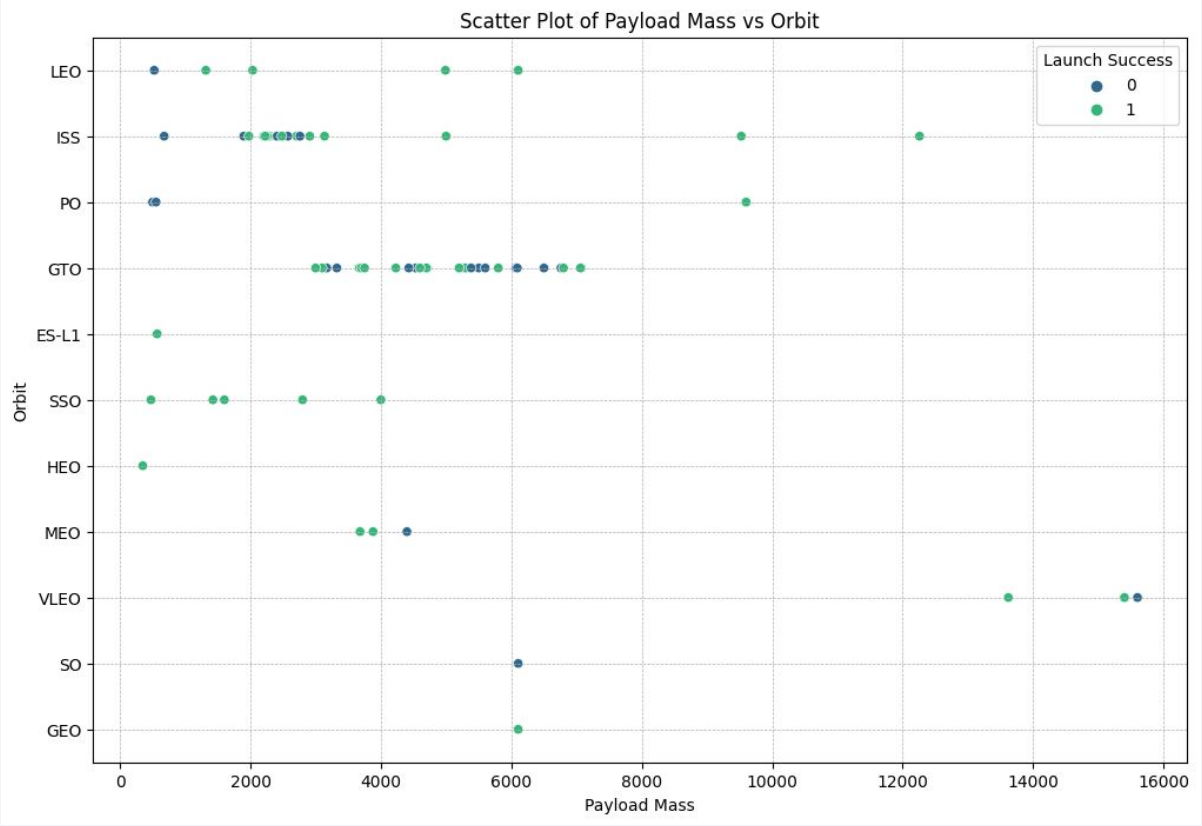
Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset.



Payload vs. Orbit Type

Heavier payload has positive impact on LEO, ISS and P0 orbit. However, it has negative impact on MEO and VLEO orbit. GTO orbit seem to depict no elation between the attributes.

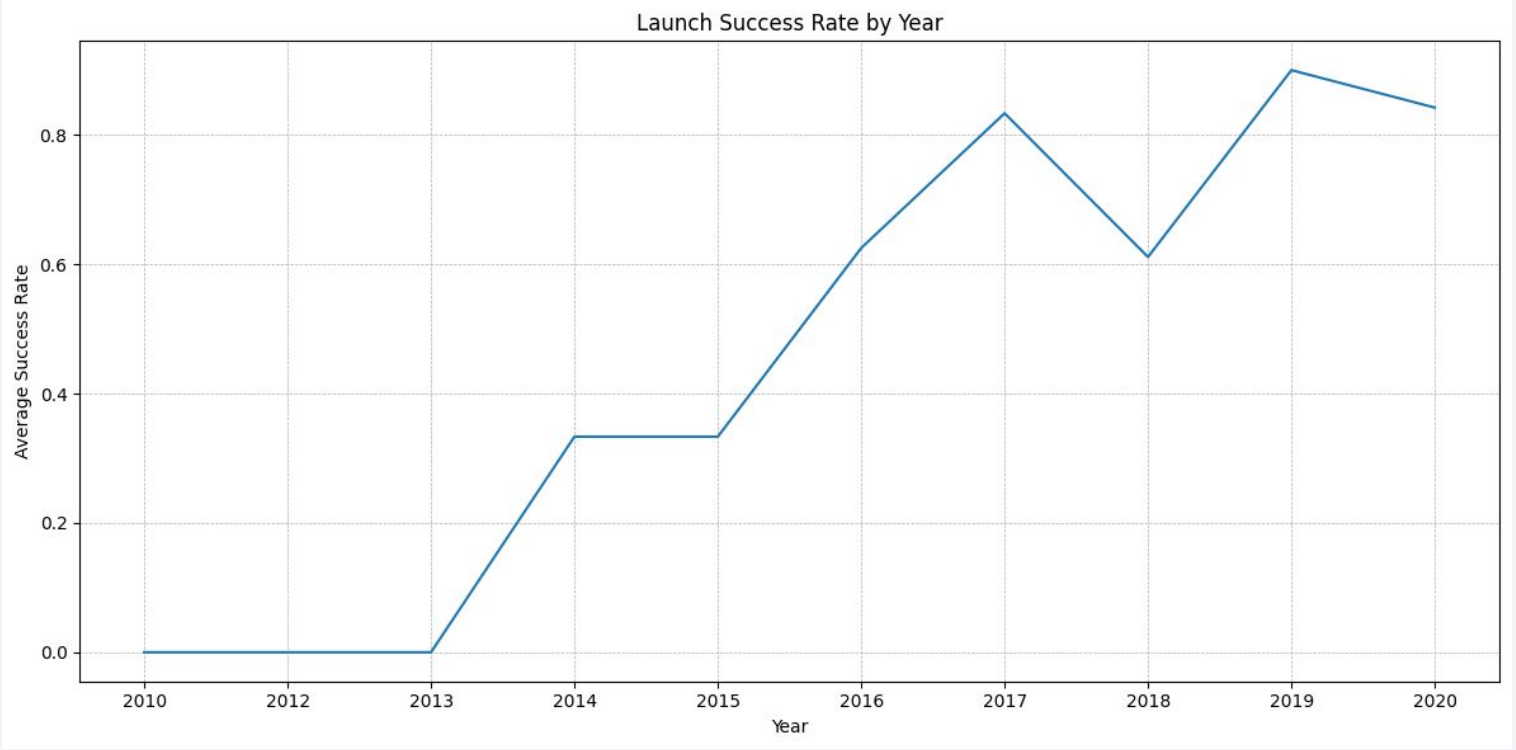
Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.



Launch Success Yearly Trend

This figures clearly depicted and increasing trend from the year 2013 until 2020.

If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.



All Launch Site Names

We retrieved data from the Launch Site and then looked at the unique counts.

```
%sql SELECT DISTINCT "Launch_Site" FROM "SPACEXTABLE"
✓ 0.4s
```

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with `CCA`.

```
%sql SELECT * FROM "SPACEXTABLE" WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

✓ 0.2s

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 Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below.

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") as total_payload_mass FROM "SPACEXTABLE" WHERE "Customer" LIKE '%NASA (CRS)%';
```

✓ 0.4s Python

total_payload_mass
48213

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") as avg_payload_mass FROM "SPACEXTABLE" WHERE "Booster_Version" = 'F9 v1.1';
```

✓ 0.2s Python

avg_payload_mass
2928.4000000000000000

First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015.

```
%sql SELECT MIN("Date") as first_success_launch_date FROM "SPACEXTABLE" WHERE "Landing_Outcome" LIKE 'Success (ground pad)';
```

✓ 0.2s Python

first_success_launch_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
l SELECT "Booster_Version" FROM "SPACEXTABLE" WHERE "Landing_Outcome" LIKE 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000  AND "PAYLOAD_MASS__KG_" <
```

✓ 0.2s Python

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

Total Number of Successful and Failure Mission Outcomes

We used wildcard like '%' to filter for WHERE Mission_Outcome was a success or a failure.

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM "SPACEXTABLE" WHERE "Mission_Outcome" LIKE 'Success%' OR "Mission_Outcome" LIKE 'Failure%' GRO
```

✓ 0.2s Python

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

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

```
%sql SELECT "Booster_Version" FROM "SPACEXTABLE" WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM "SPACEXTABLE");
```

✓ 0.2s Python

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

2015 Launch Records

We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql SELECT EXTRACT(MONTH FROM CAST("Date" AS DATE)) AS "Month_Name", "Landing_Outcome", "Booster_Version", "Launch_Site" FROM  "SPACEXTABLE"
WHERE EXTRACT(YEAR FROM CAST("Date" AS DATE)) = 2015 AND "Landing_Outcome" LIKE 'Failure (drone ship)%' ORDER BY "Month_Name";
```

✓ 0.2s Python

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
4	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

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

We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20. We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

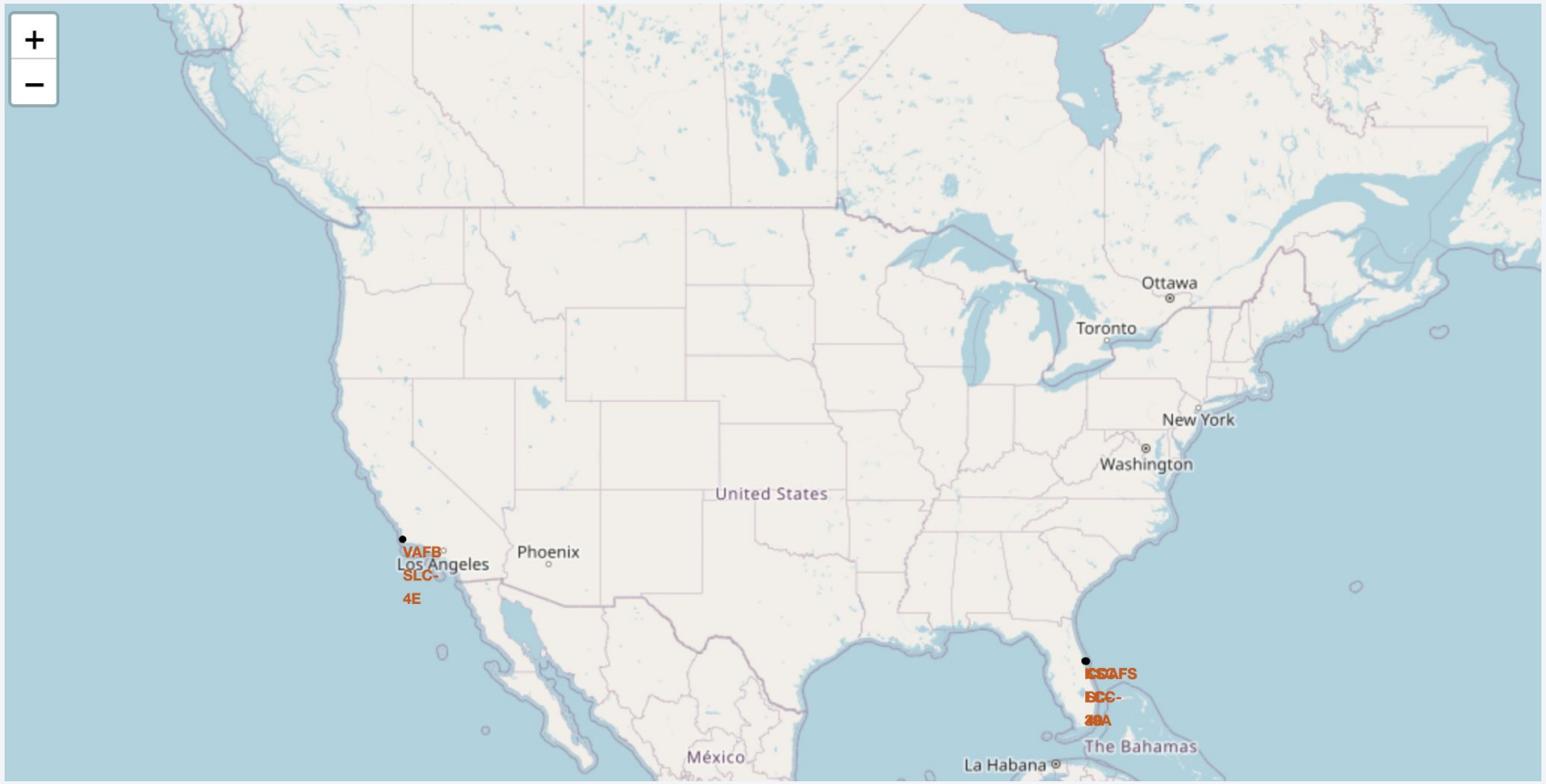
```
%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS count_outcome FROM "SPACEXTABLE"
WHERE CAST("Date" AS DATE) BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY count_outcome DESC;
```

✓ 0.2s Python

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

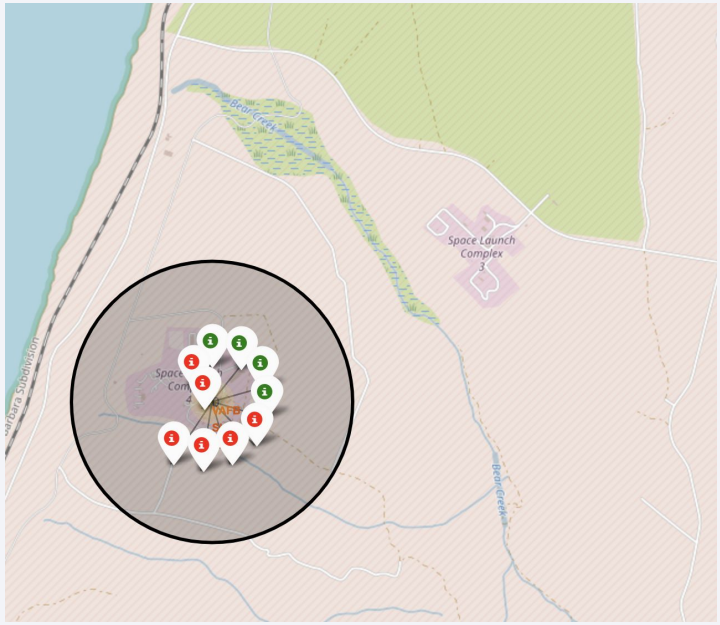

Section 3
Launch Sites
Proximities Analysis

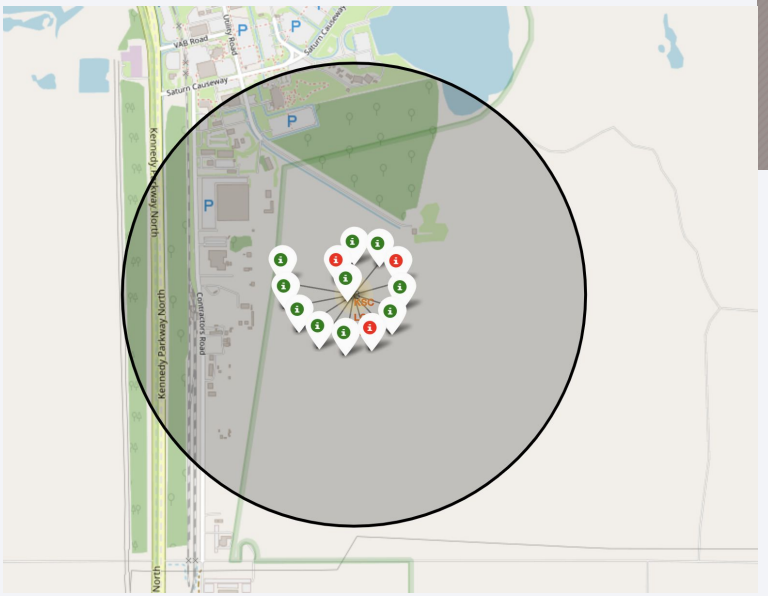
Location of all the Launch Sites

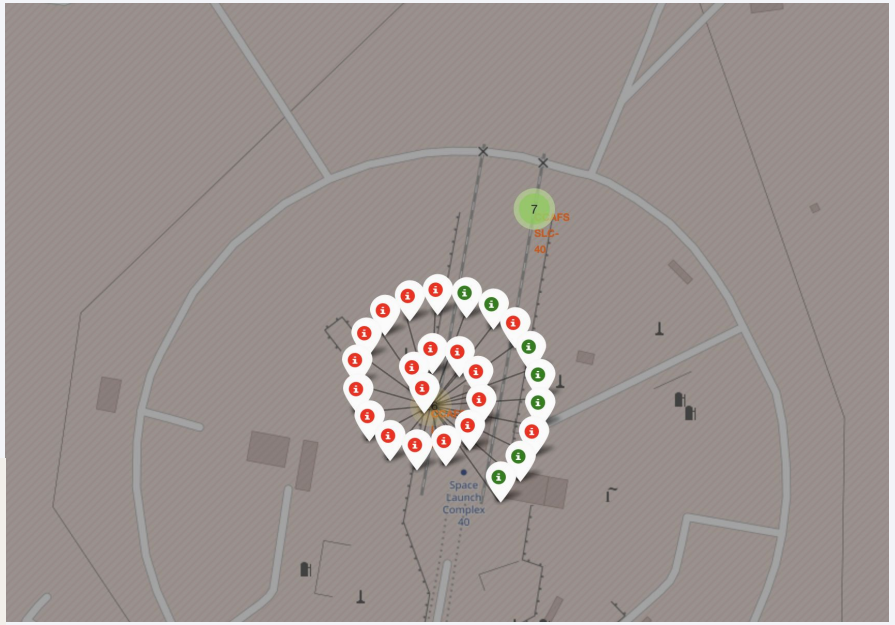


We can see that all the SpaceX launch sites are located inside the United States

Markers showing launch sites with color labels





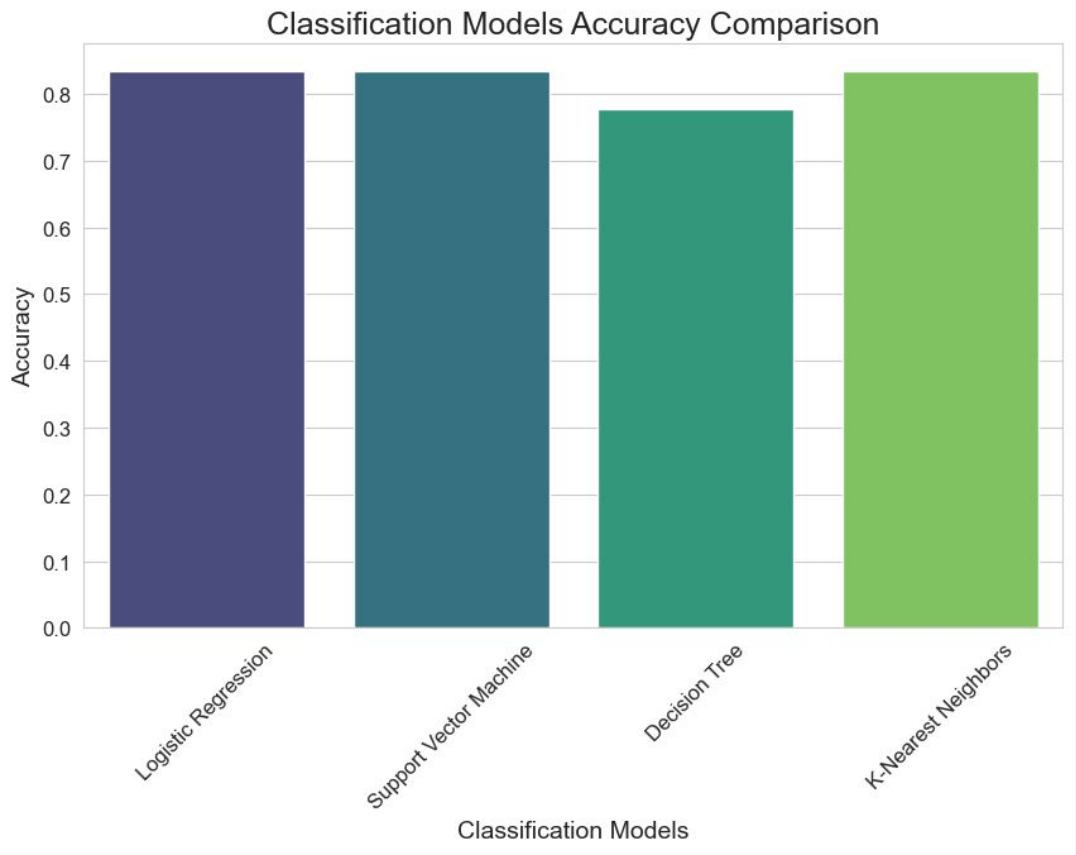


Section 5

Predictive Analysis (Classification)

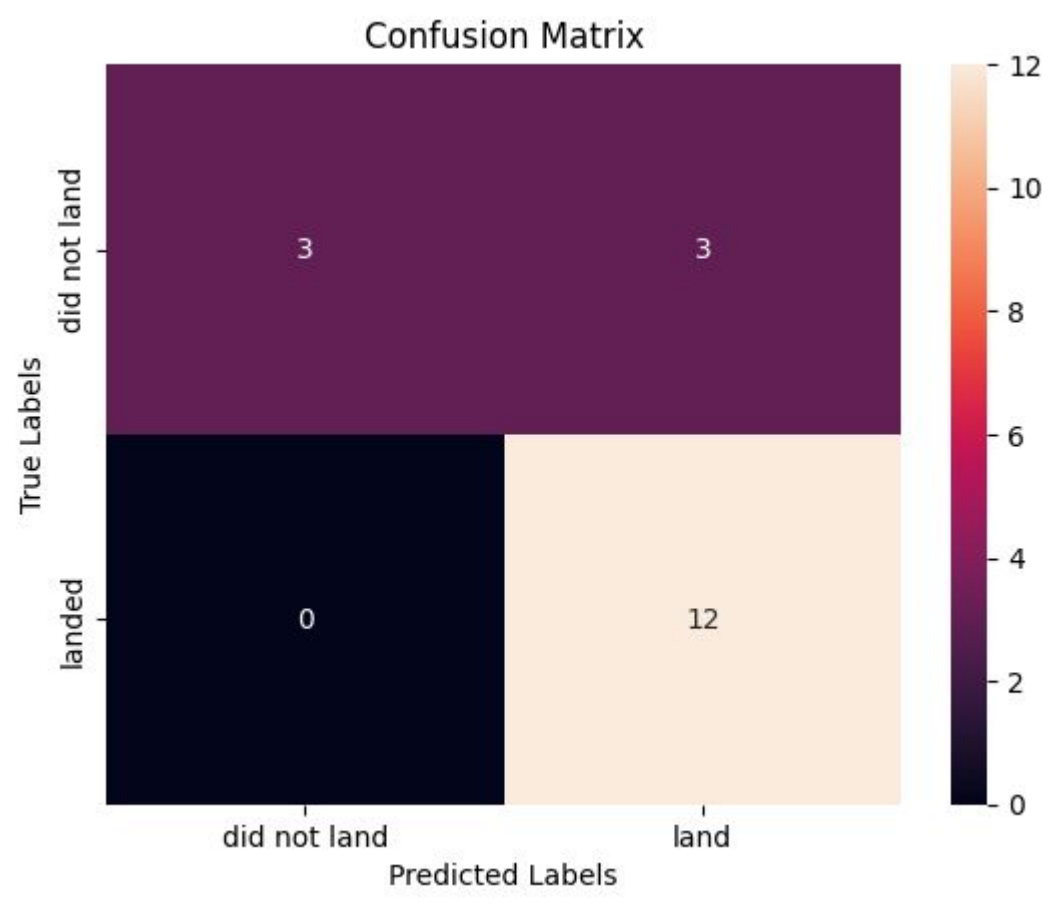
Classification Accuracy

After evaluating multiple machine learning classification models, including Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors, it was determined that the Logistic Regression model outperformed the others. This model achieved the highest accuracy of 0.83 on the test dataset. This indicates that, for this particular dataset and given the features used, Logistic Regression is the most reliable algorithm for making accurate predictions.



Confusion Matrix

After evaluating several classification methods for our data, Logistic Regression emerged as the top performer. This technique provided us with the most accurate predictions compared to other models. It's crucial to choose the best model for predictions, and in our case, Logistic Regression has proven to be the most reliable. By using this method, we can make more confident decisions based on the data and achieve better outcomes in our projects.



Conclusions

From our data analysis, we have derived the following conclusions:

- The Logistic Regression algorithm demonstrated the most reliable results for our dataset.
- Lighter payloads have exhibited a more consistent success rate compared to their heavier counterparts.
- Starting from 2014, we've observed a significant uptick in the success rate of SpaceX launches. This trend illustrates the maturation of technology and experience over time.
- The KSC LC-39A launch site boasts the highest success rate, suggesting potential technical and logistical advantages associated with this location.
- The high success rate for the SSO orbit indicates that this trajectory might be more stable and safer for launches.

Appendix

Clicking this link will give you access to the project code and more

Applied Data Science Project IBM

Thank you!