IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abdelhadi FADILI
11 March 2024

Outline

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

Executive Summary

- Summary of methodologies
    1. Data Collection through API
    2. Data Collection with Web Scraping
    3. Data Wrangling
    4. Exploratory Data Analysis with SQL
    5. Exploratory Data Analysis with Data Visualization
    6. Interactive Visual Analytics with Folium
    7. Machine Learning Prediction
- Summary of all results
    1. Exploratory Data analysis result
    2. Interactive Analytics in screenshots
    3. Predictive Analytics result from ML lab

Introduction

With rocket launches, notably Falcon 9, starting at under 62 million dollars, SpaceX is a ground-breaking enterprise that has completely upended the space market. In contrast, other services

cost more than $165 million apiece. The majority of these savings are attributable to SpaceX's brilliant concept to re-land the rocket after the first stage of the flight in order to use it on a later mission. The price will drop considerably more if this practice is repeated. The objective of this project, as a data scientist for a firm that competes with SpaceX, is to develop a machine learning pipeline to forecast the first stage landing result in the future.

This project is essential to determining how much to offer SpaceX for a rocket launch.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variables and how it is affecting the outcome.
- The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling
  - Data was processed using one-hot encoding from categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and PowerBI
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia

For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe using json_normalize(). We then cleaned the data, checked for missing values and fill with whatever needed.

For web scrapping, we will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection – SpaceX API

Get request for rocket launch data using API

Use json_normalize method to convert json result to dataframe

Performed data cleaning and filling the missing value

From:
Falcon9_Launch_project/capstone_falcon9_part1.ipynb at main · FADili12/Falcon9_Launch_project (github.com)

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

```
response.status_code
```

```
200
```

```
#json_noramize method to convert the json result into a dataframe
json_data = response.json()

#normalize the JSON data into a DF

data = pd.json_normalize(json_data)
```

```
# 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 r
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)]
```

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from url

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
print(response.content)
```

Create a BeautifulSoup from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text)
```

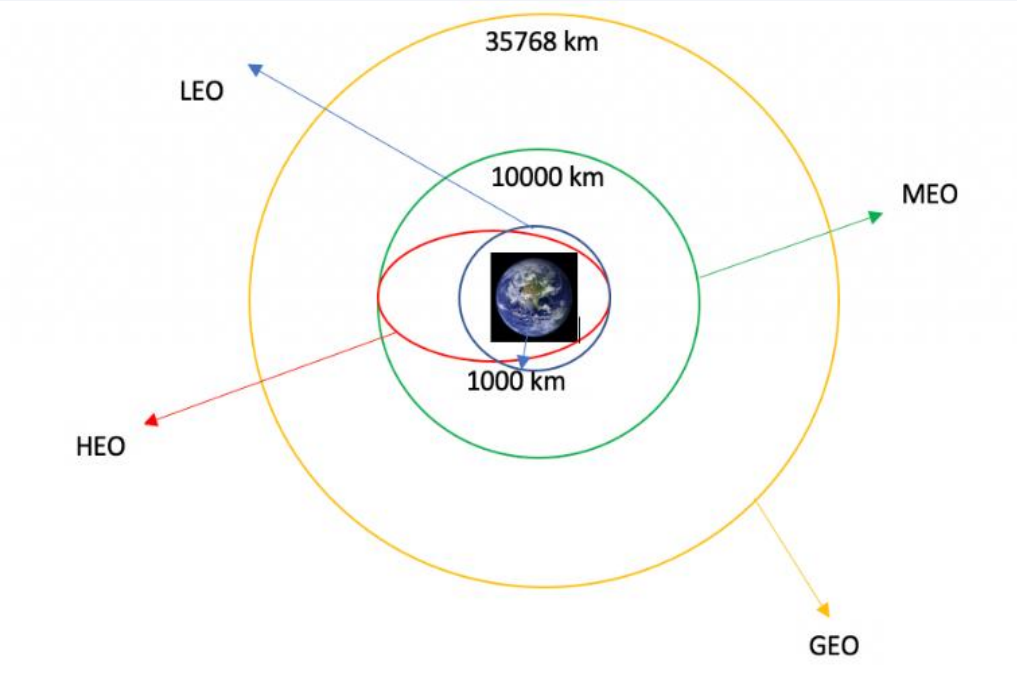
Extract all column/variable names from the HTML header

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
th_element = first_launch_table.find_all('th')
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for th in th_element:
    name = extract_column_from_header(th)
    # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
    if name is not None and len(name) > 0:
        column_names.append(name)
```

From:
[Falcon9_Launch_project/capstone_falcon9_webscraping.ipynb at main · FADili12/Falcon9_Launch_project · GitHub]

Data Wrangling



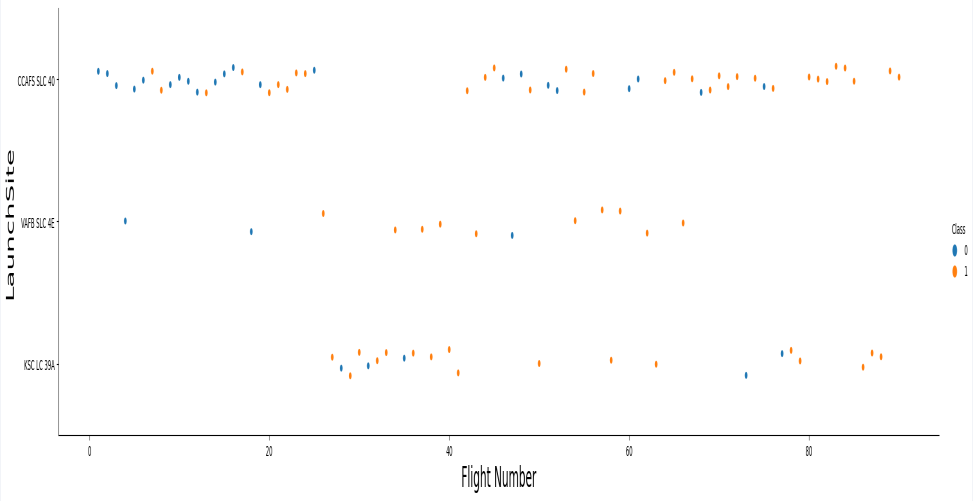
From:
[Falcon9_Launch_project/capstone_falcon9_webscraping.ipynb at main · FADili12/Falcon9_Launch_project · GitHub]

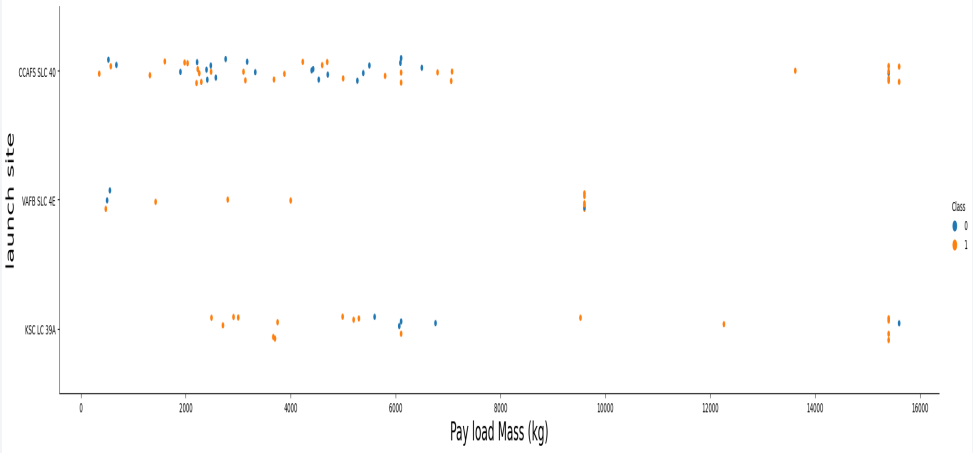
Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA) .

We will first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type

We then create a landing outcome label from the outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV.

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.

Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to
see which factors affecting the most to the success of the landing outcomes

From:
Falcon9_Launch_project/eda-datavizualization.ipynb at main · FADili12/Falcon9_Launch_project · GitHub

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

from:
Falcon9_Launch_project/EDA_with_SQL.ipynb at main · FADili12/Falcon9_Launch_project · GitHub

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().

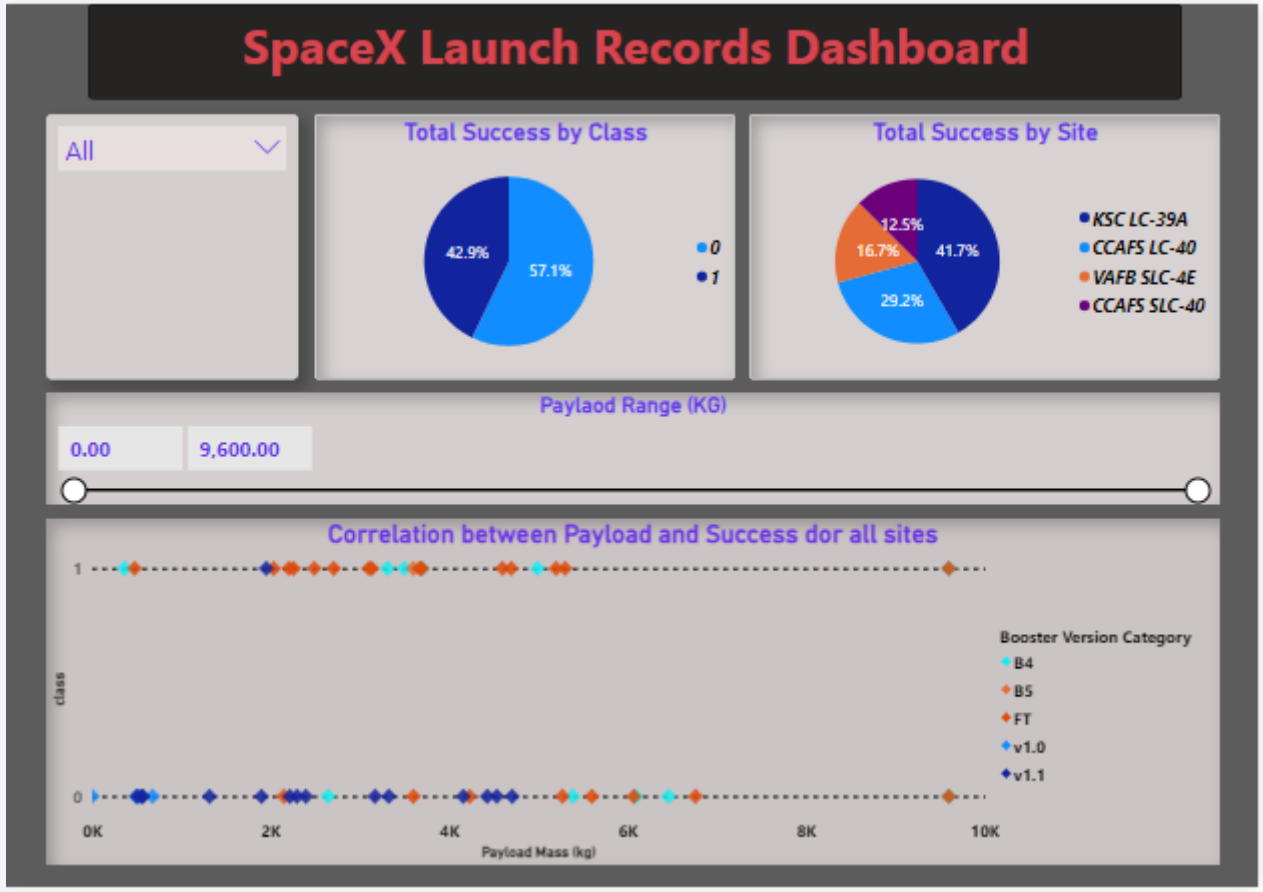
We then used the Haversine's formula to calculated the distance of the launch sites to various landmark to find answer to the questions of:

- How close the launch sites with railways, highways and coastlines?
- How close the launch sites with nearby cities?

From:
Falcon9_Launch_project/launch_site_location.ipynb at main · FADili12/Falcon9_Launch_project · GitHub

Build a Dashboard with Power BI

- The monitor request a dashboard with Plotly dash but for some problems in my python version. I built a dashboard with Power Bi.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.



From:

[Falcon9_Launch_project/SpaceX_Dashboard.pbix at main · FADili12/Falcon9_Launch_project · GitHub]

Predictive Analysis (Classification)

Load the dataset into NumPy and Pandas
•Transform the data and then split into training and test datasets
•Decide which type of ML to use
•set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model
•Check the accuracy for each model
•Get tuned hyperparameters for each type of algorithms.
•plot the confusion matrix

Improving the Model
•Use Feature Engineering and Algorithm Tuning
.

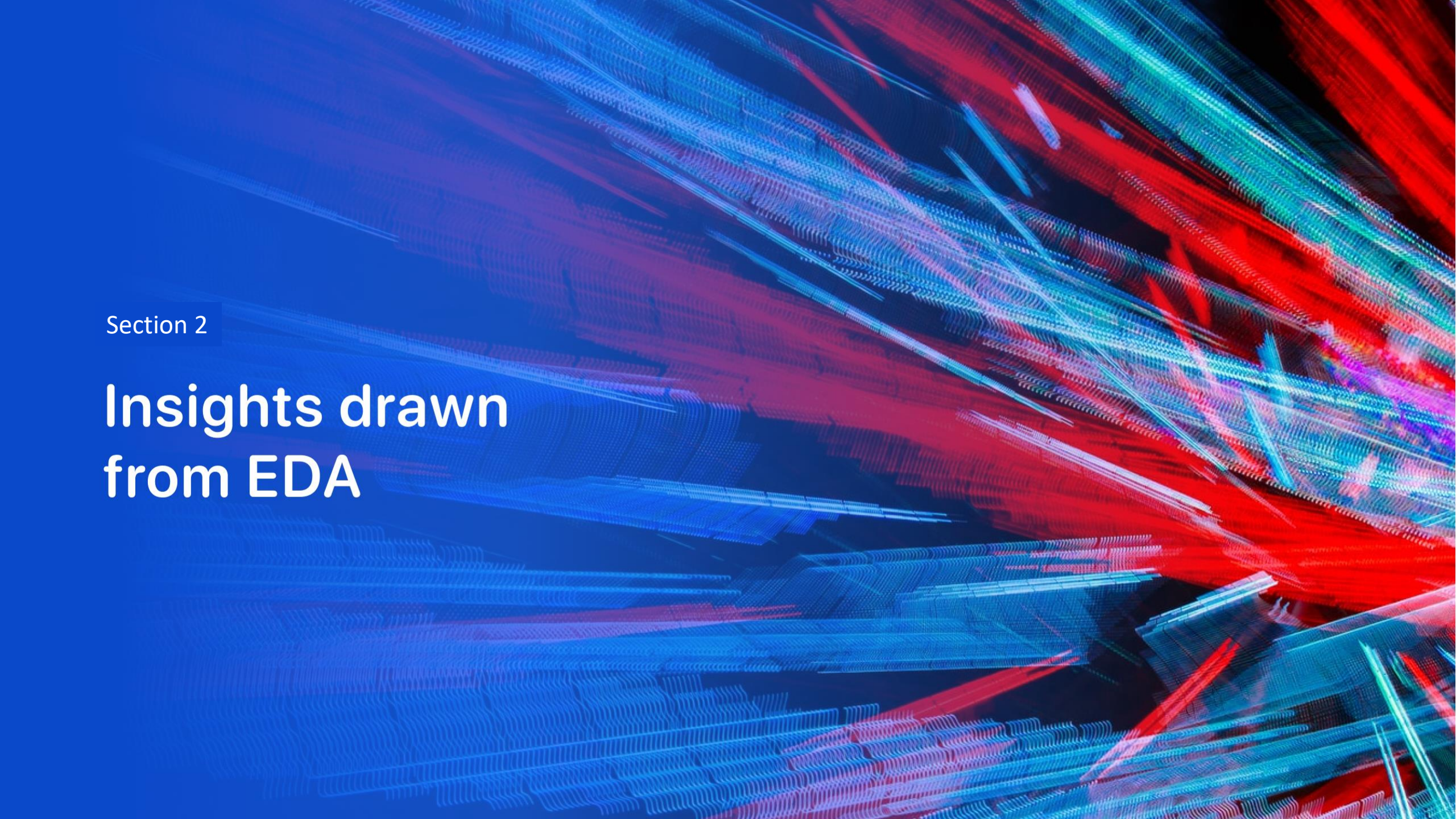
Find the Best Model
•The model with the best accuracy score will be the best performing model

Falcon9_Launch_project/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · FADili12/Falcon9_Launch_project · GitHub

Results

The results will be categorized to 3 main results which is:

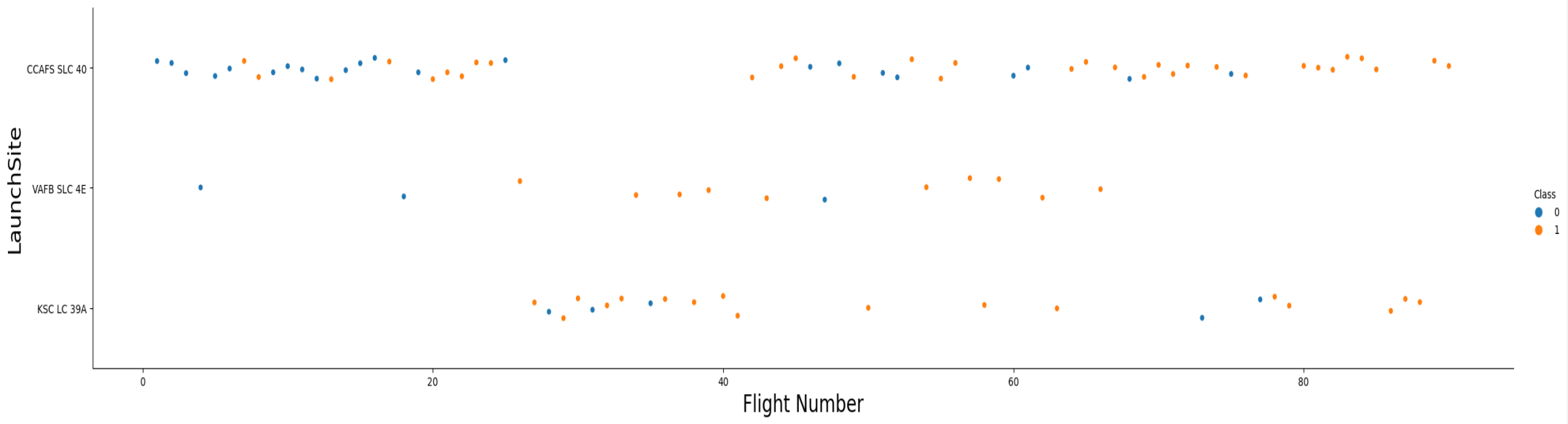
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

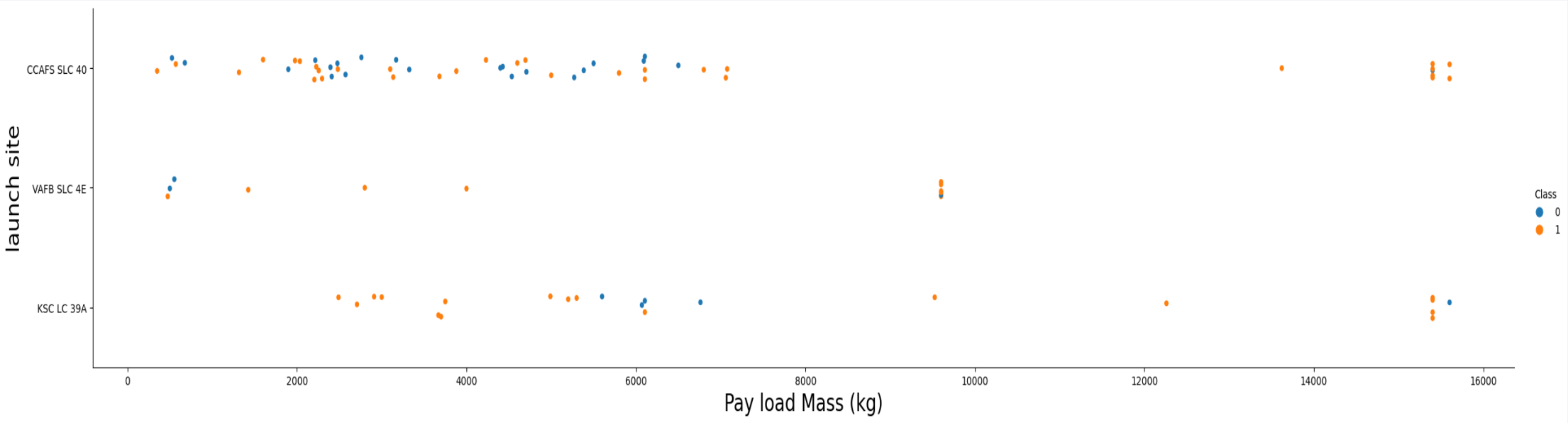
This scatter plot shows that the larger the flights amount of the launch site. The greater the success rate will be.

However, site CCAFS and SLC40 shows the least pattern of this.



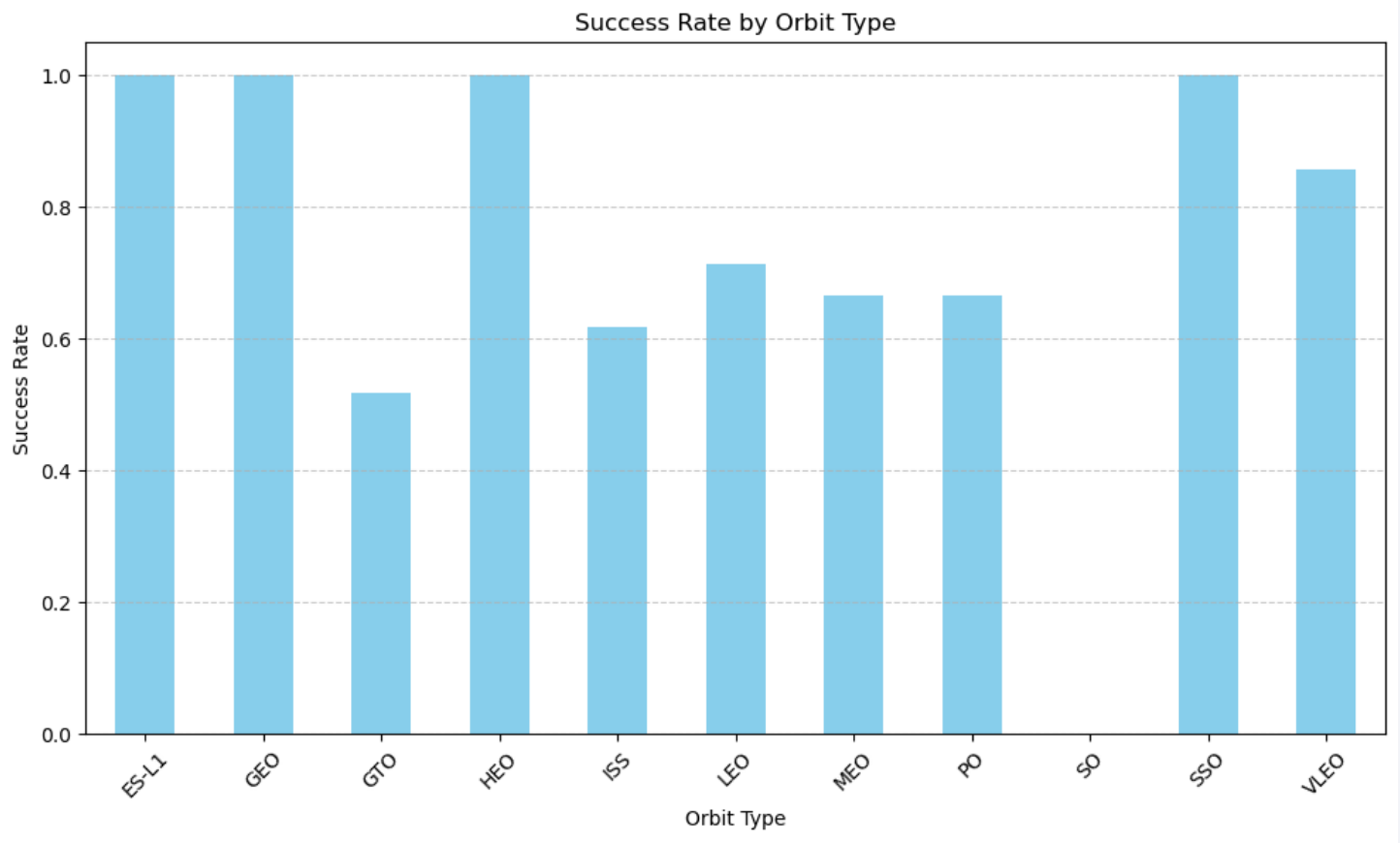
Payload vs. Launch Site

This scatter plot shows once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased.
However, there is no clear pattern to say the launch site is dependent to the pay load mass for the success rate



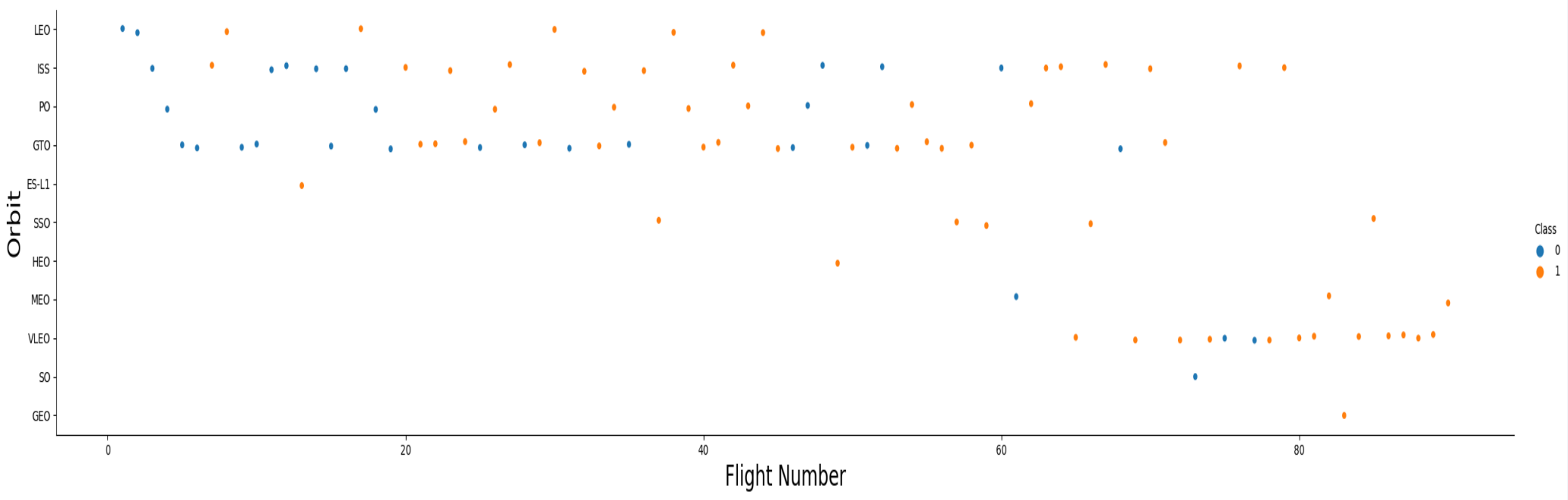
Success Rate vs. Orbit Type

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.



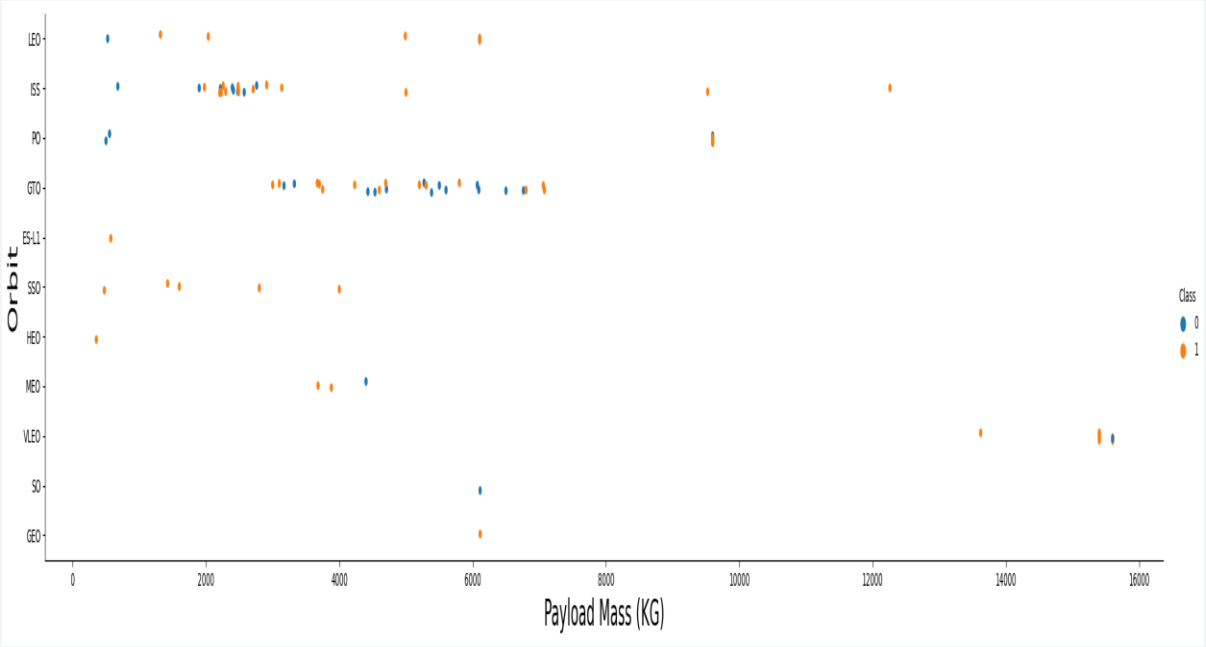
Flight Number vs. Orbit Type

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.

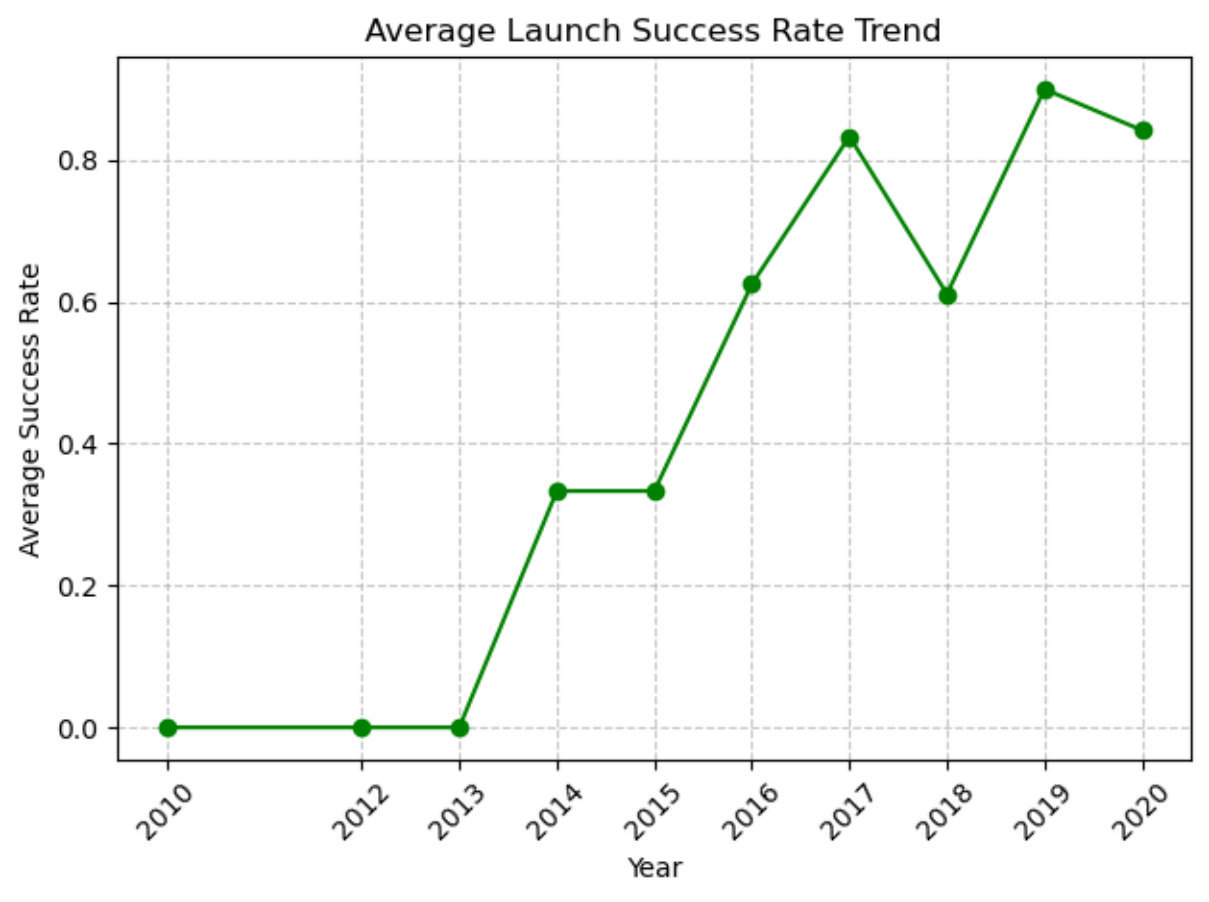


Payload vs. Orbit Type

- 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 relation 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 used the key word **DISTINCT** to show only unique launch sites from the SpaceX data

```
result = %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
result
```

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

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with ‘CCA’

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

```
record_5launch = %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
record_5launch
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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) NRC
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)

Total Payload Mass

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

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

```
result = %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
result
```

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

TotalPayloadMass
45596

Average Payload Mass by F9 v1.1

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

```
result = %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
result
```

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

AveragePayloadMass
2928.4

First Successful Ground Landing Date

We use the min() function to find the result.

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

```
result = %sql SELECT MIN(Date) AS FirstSuccessfulLandingDate FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground p
result
```

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

FirstSuccessfulLandingDate
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
.

```
result = %sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAI
result
```

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

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** MissionOutcome was a success or a failure

```
result = %sql SELECT Mission_Outcome, COUNT(*) AS TotalCount FROM SPACEXTBL GROUP BY Mission_Outcome;
result
```

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

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

Boosters Carried Maximum Payload

```
result = %sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
result
```

```
 * 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

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

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

```
result = %sql SELECT CASE WHEN substr(Date, 6, 2) = '01' THEN 'January' \
WHEN substr(Date, 6, 2) = '02' THEN 'February' WHEN substr(Date, 6, 2) = '03' \
THEN 'March' WHEN substr(Date, 6, 2) = '04' THEN 'April' WHEN substr(Date, 6, 2) = '05' \
THEN 'May' WHEN substr(Date, 6, 2) = '06' THEN 'June' WHEN substr(Date, 6, 2) = '07' THEN 'July' WHEN\
substr(Date, 6, 2) = '08' THEN 'August' WHEN substr(Date, 6, 2) = '09' THEN 'September' WHEN substr(Date, 6, 2) = '10
'October' WHEN substr(Date, 6, 2) = '11' THEN 'November' WHEN substr(Date, 6, 2) = '12' THEN 'December' END \
AS MonthName, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date, 0, 5) = '2015' AND Land
```

```
 * sqlite:///my_data1.db
)one.
```

```
result
```

MonthName	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	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

```
result = %sql SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '20
result
```

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

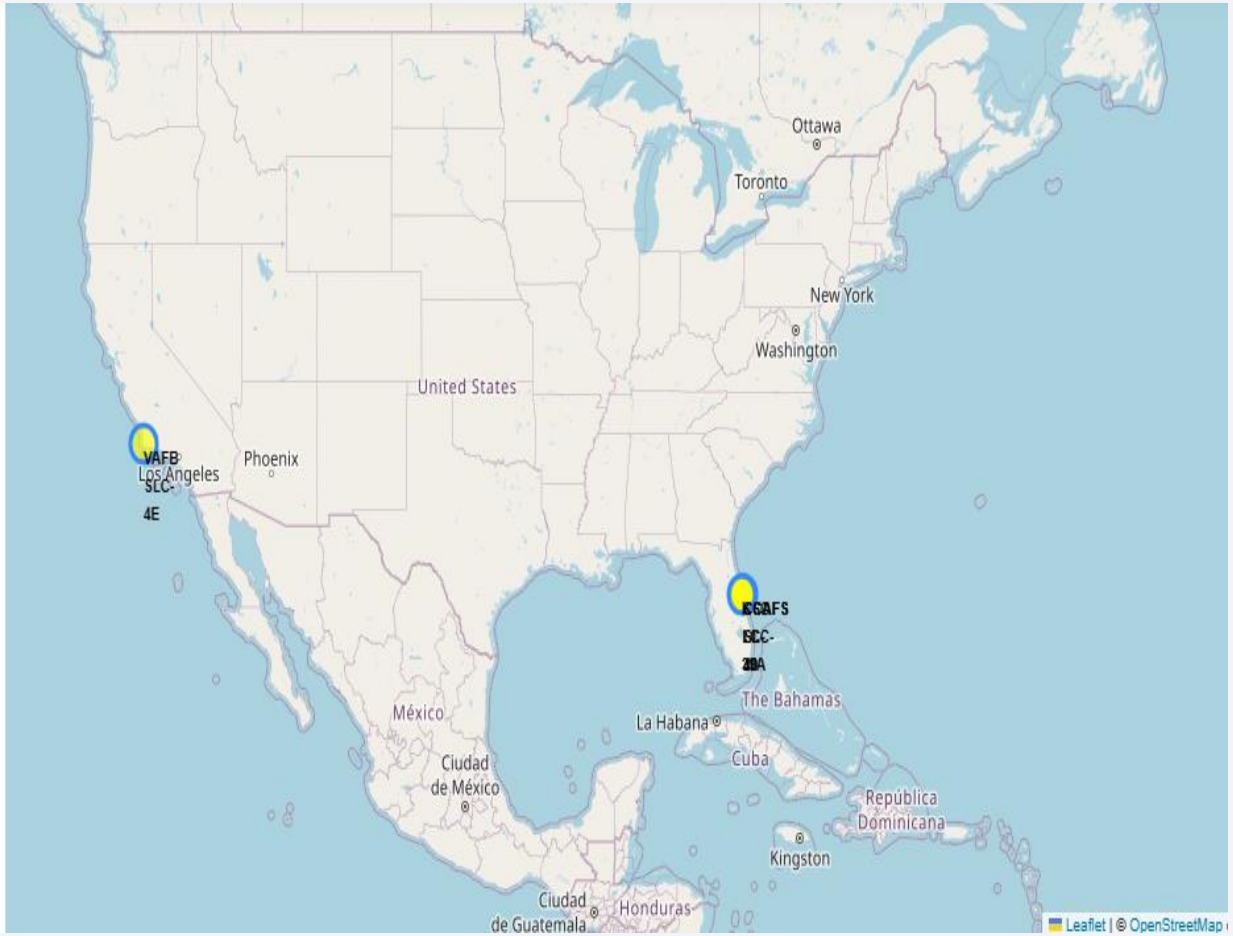
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

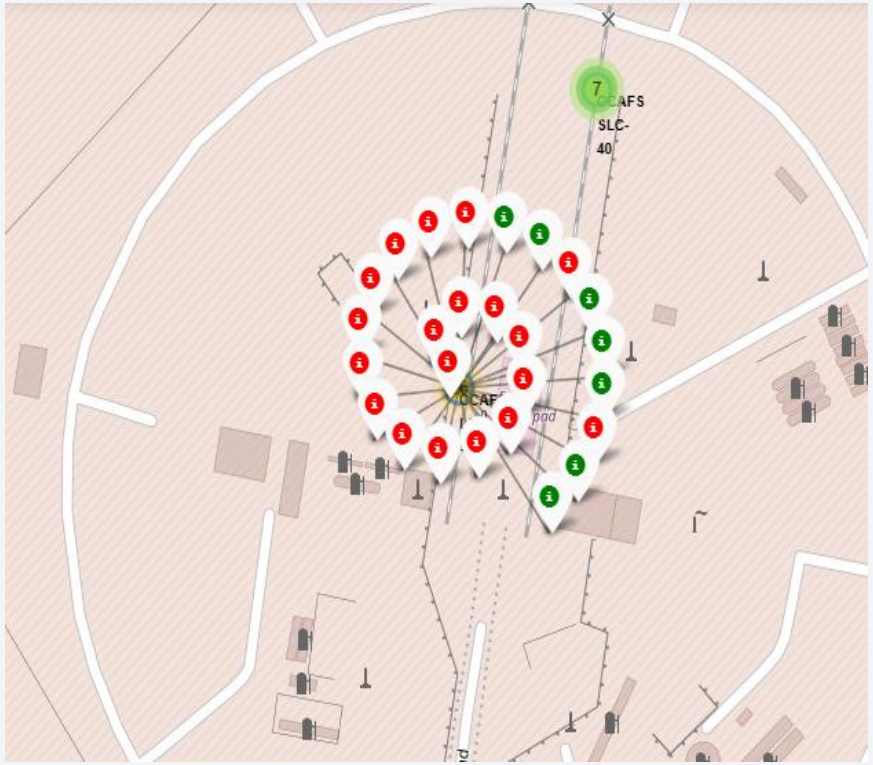
Launch Sites Proximities Analysis

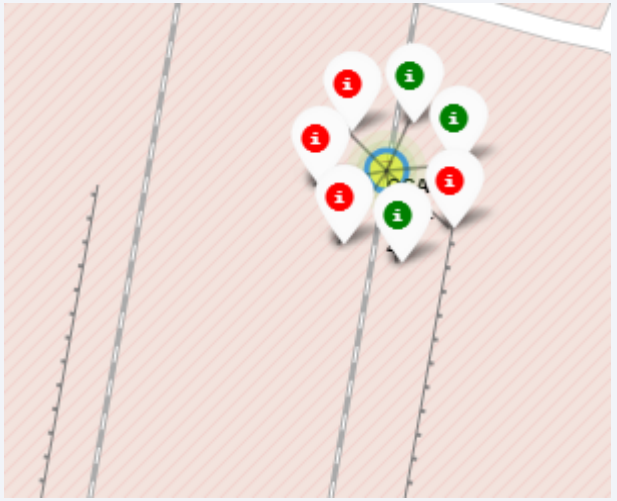
Location of all the launch site

The figure shows the location of all the SpaceX launch sites, which are located in United States.



Markers showing launch sites with color labels





Green marker shows successful launches and Red marker shows failures.

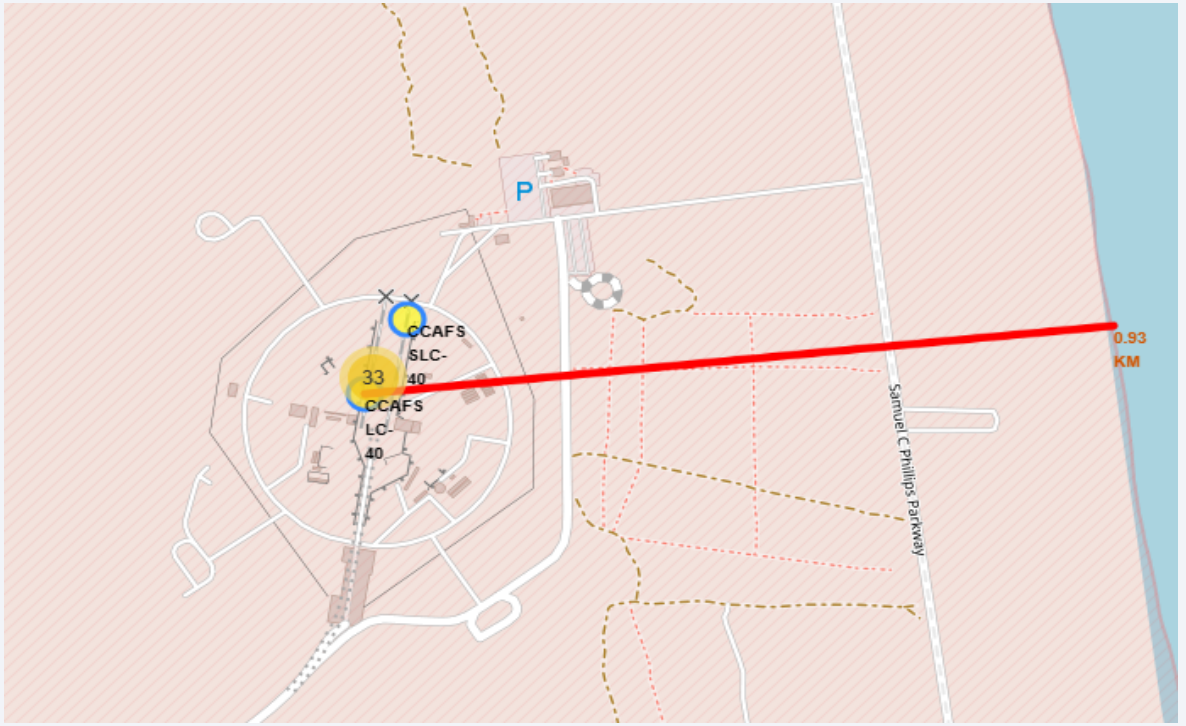
Launch Sites and Landmarks

We calculated the distances between a launch site to its proximities, to explore and analyze the proximities of launch sites.

```
Distance between VAFB_SLC_4E and  coastline: 1.33 km
Distance between VAFB_SLC_4E and  highway: 0.06 km
Distance between VAFB_SLC_4E and  railway: 1.27 km
Distance between VAFB_SLC_4E and  city: 12.04 km
```

```
2]: lat1 = 28.573255
lon1 = -80.646895
for key, value in KSC_LC_39A_points.items():
    distance = calculate_distance(lat1, lon1, value[0], value[1
    # Round the distance to two decimal places
    distance = round(distance, 2)
    print(f"Distance between KSC_LC_39A and  {key}: {distance}
```

```
Distance between KSC_LC_39A and  coastline: 0.88 km
Distance between KSC_LC_39A and  highway: 0.32 km
Distance between KSC_LC_39A and  railway: 0.7 km
```



Launch Sites and Landmarks

Based on the provided distance results, we can make some observations:

All three launch sites are relatively close to railways, with distances ranging from 0.7 km to 1.27 km. Therefore, we can conclude that launch sites are generally in close proximity to railways.

Similar to railways, all three launch sites are close to highways. Distances range from 0.06 km to 1.05 km, indicating that launch sites are often situated near highways.

Launch sites show a close proximity to the coastline as well, with distances ranging from 0.88 km to 1.33 km. This suggests that launch sites are typically located near coastlines.

CCAFS SLC-40 is about 1.51 km away from a city, while VAFB_SLC_4E is significantly further at 12.04 km. KSC_LC_39A's distance from cities is not provided. However, based on the available data, it seems that launch sites tend to maintain a certain distance from cities, with varying degrees of separation.

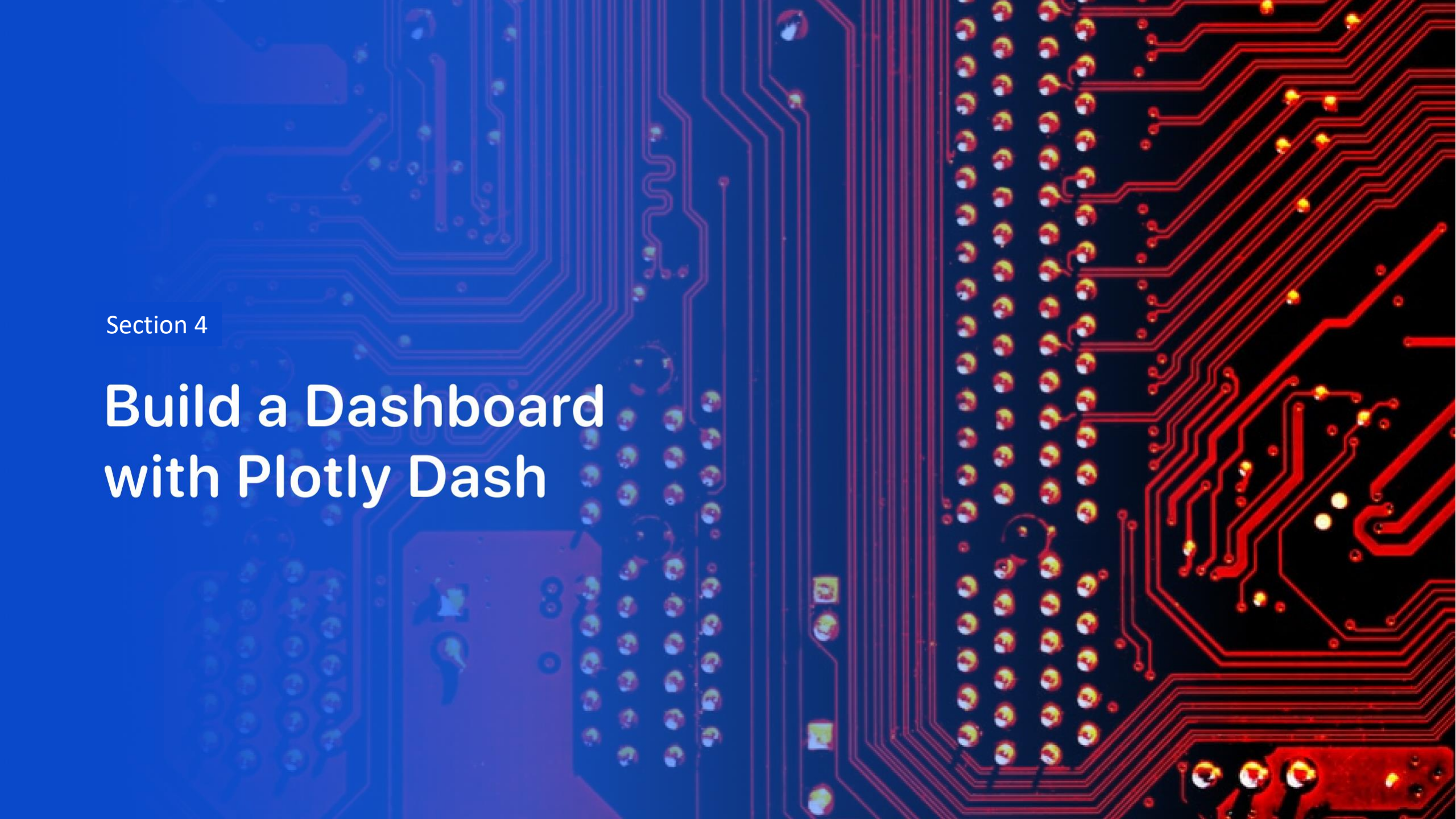
In summary, launch sites generally tend to be located close to railways, highways, and coastlines, while also maintaining a certain distance from populated cities

```
lat1 = 28.563197
lon1 = -80.576820
for key, value in other_launch_sites.items():
    distance = calculate_distance(lat1, lon1, value[0], value[1])
    # Round the distance to two decimal places
    distance = round(distance, 2)
    print(f"Distance between CCAFS SLC-40 and  {key}: {distance} km")
```

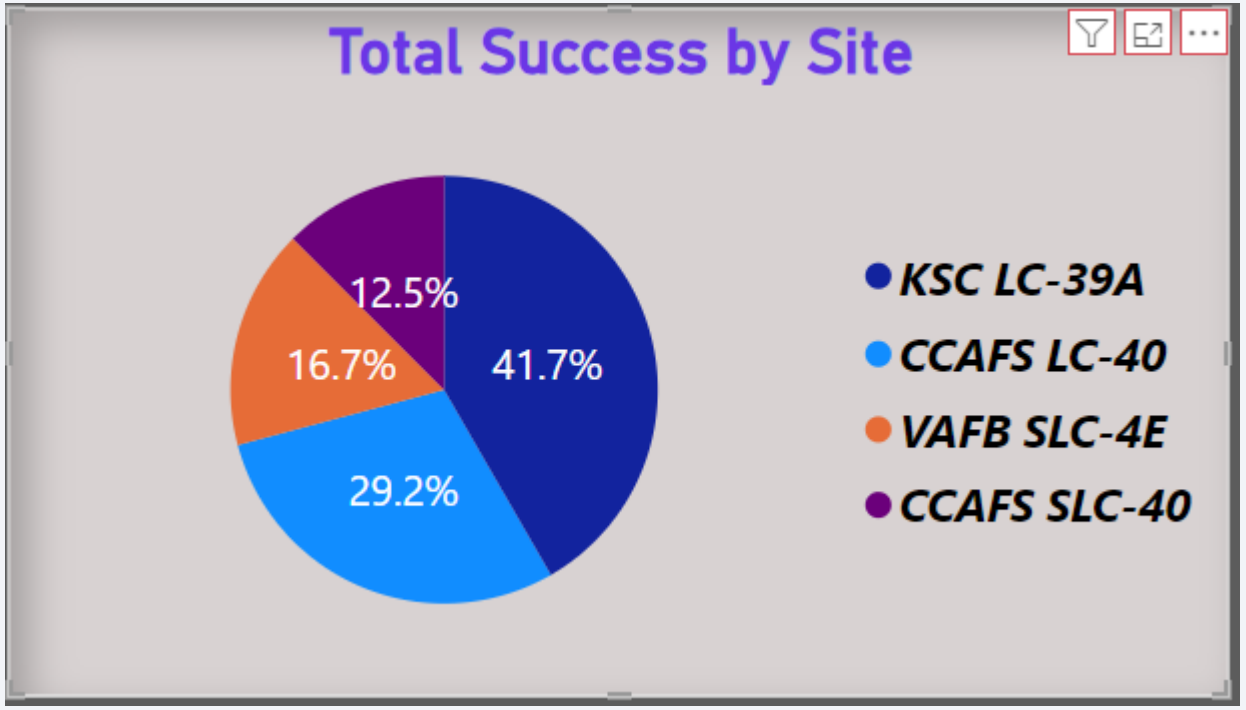
```
Distance between CCAFS SLC-40 and  coastline: 1.23 km
Distance between CCAFS SLC-40 and  highway: 1.05 km
Distance between CCAFS SLC-40 and  railway: 1.08 km
Distance between CCAFS SLC-40 and  city: 1.51 km
```

After you plot distance lines to the proximities, you can answer the following questions easily:

- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities?

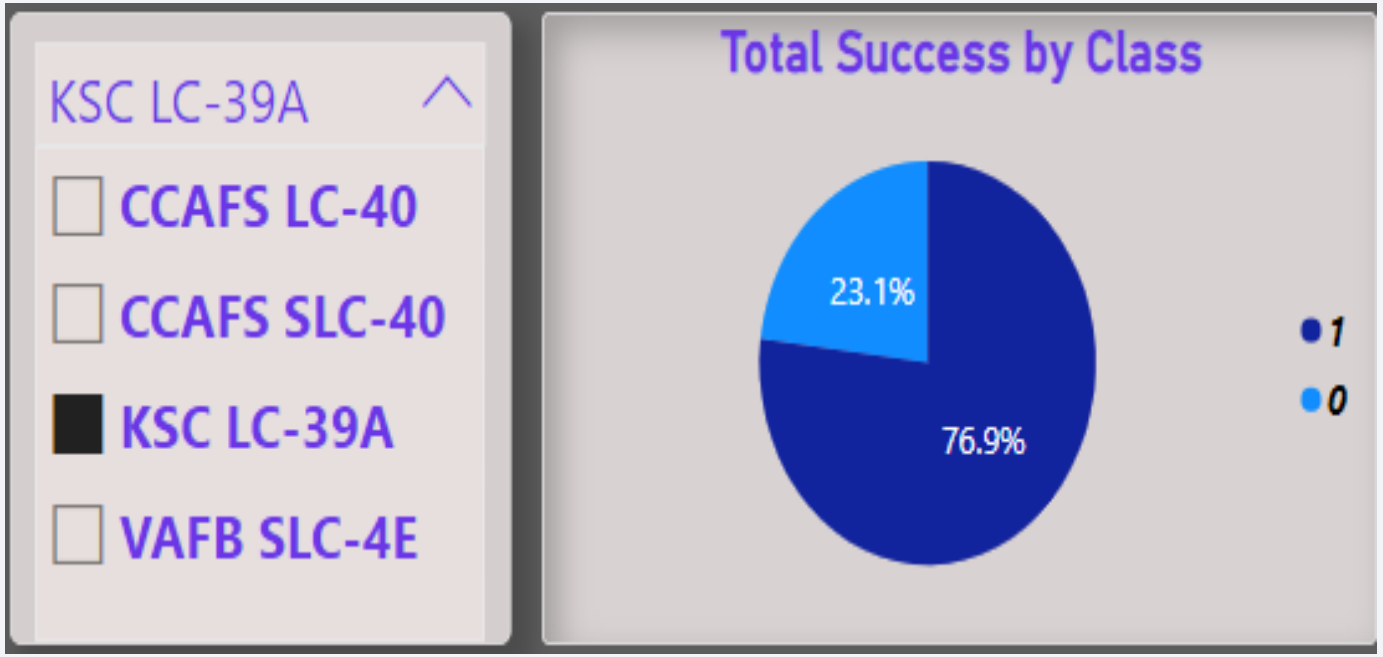

Section 4
Build a Dashboard with Plotly Dash

Success percentage by sites



The pie chart shows that, the KSC LC-39A site has the highest percentage.

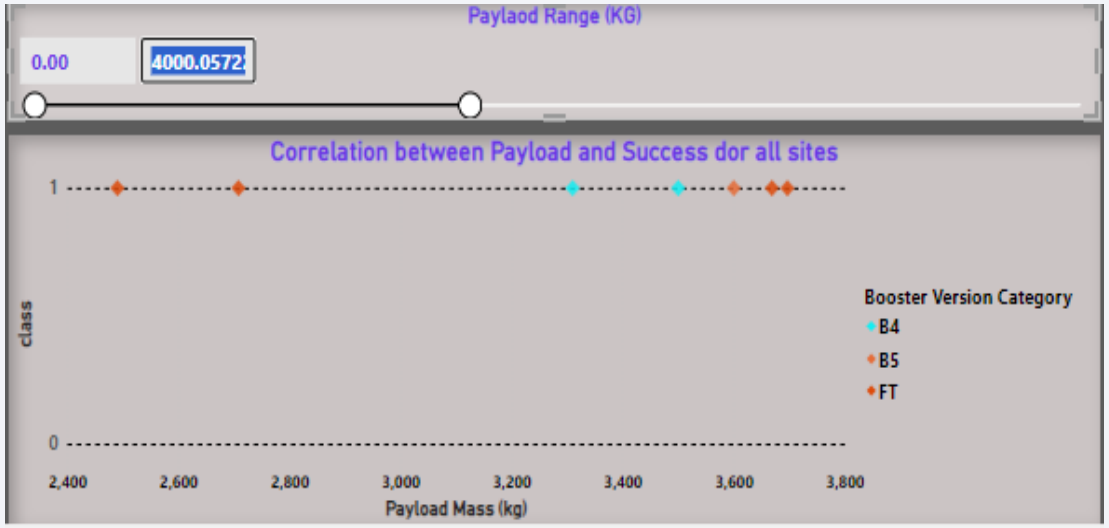
The highest launch-success ratio: KSC LC-39A

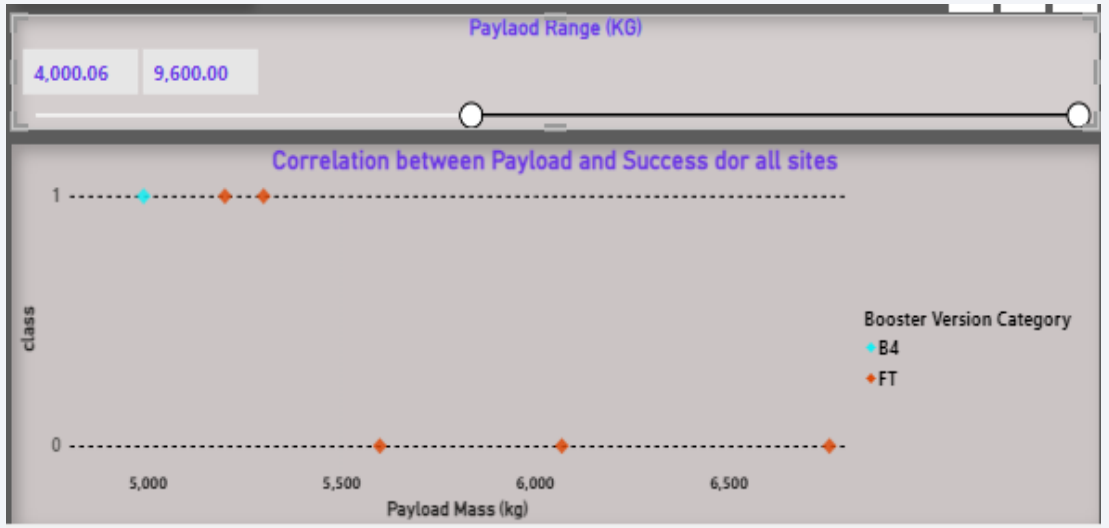


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.

Payload vs Launch Outcome scatter plot

We see that all the success rate for low weighted payload is higher the heavy weighted payload





Section 5

Predictive Analysis (Classification)

Classification Accuracy

As we can see, by using the code blow, we could identity the best algorithm which is Logistic Regression.

TASK 12

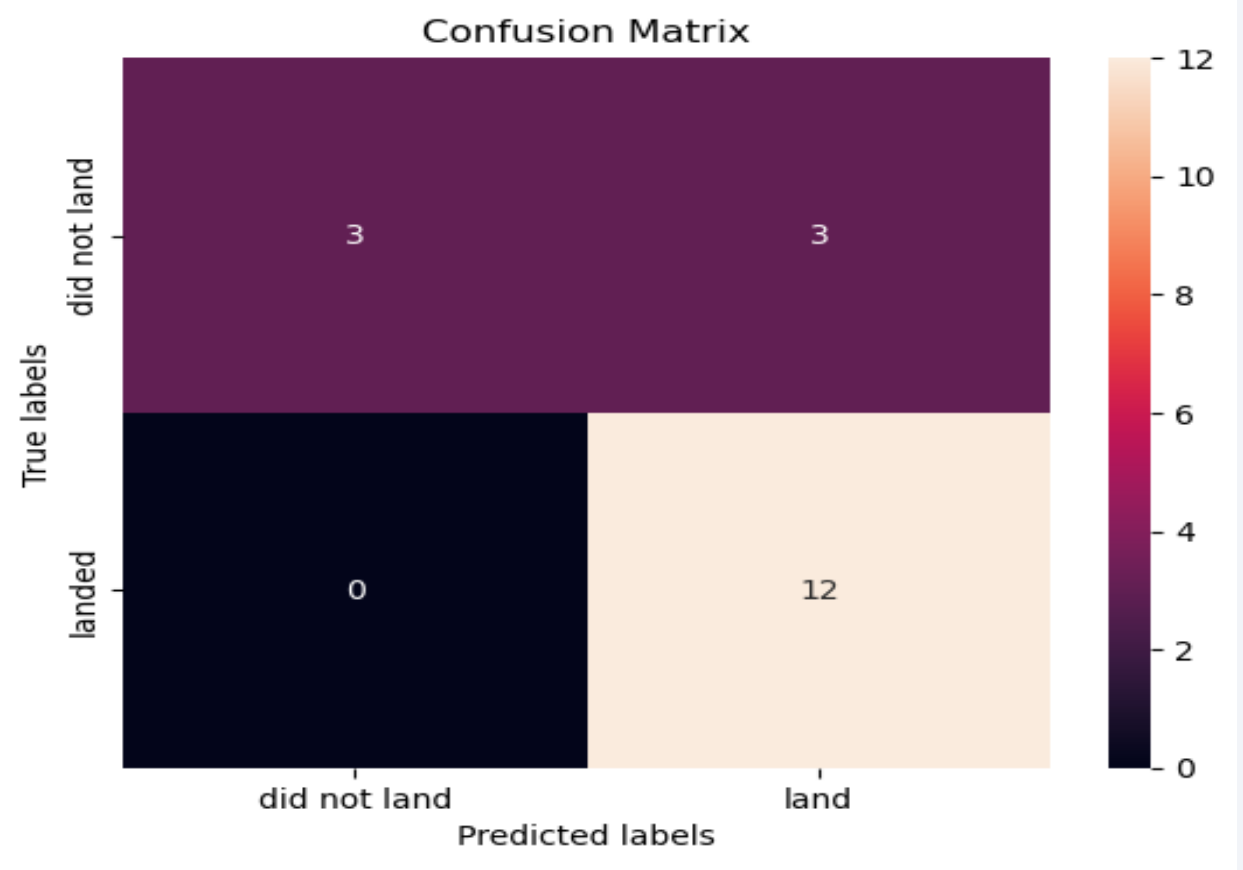
Find the method performs best:

```
best_accuracy = max(accuracy_knn,accuracy_log,accuracy_tree,accuracy_svm)
if best_accuracy == accuracy_log:
    print("Logistic Regression performs the best.")
elif best_accuracy == accuracy_svm:
    print("Support Vector Machine performs the best.")
elif best_accuracy == accuracy_tree:
    print("Decision Tree performs the best.")
else:
    print("k-Nearest Neighbors performs the best.")
```

```
Logistic Regression performs the best.
```

Confusion Matrix

The confusion matrix for the **Logistic Regression** shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.

Thank you!