

# Winning Space Race with Data Science

Suyog Manoj Vaidya  
12 February 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

# Introduction

---

- Project background and context

SpaceX is a revolutionary company who has disrupted the space industry by offering rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully.

- The problems included:

- Identifying all factors that influence the landing outcome.
- The interaction amongst various features that determine the success rate of a successful landing.
- 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 API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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 using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json\_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future 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 values

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

response = requests.get(spacex_url)

# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())

# 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 rocket s 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)]
```

From:

[https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX\\_Falcon9\\_Data\\_Collection\\_API.ipynb](https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX_Falcon9_Data_Collection_API.ipynb)

# Data Collection - Scraping

Request the Falcon9  
Launch Wiki page from url

Create a BeautifulSoup  
from the HTML response

Extract all column/variable  
names from the HTML  
header

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(page.text, 'html.parser')
```

```
column_names = []

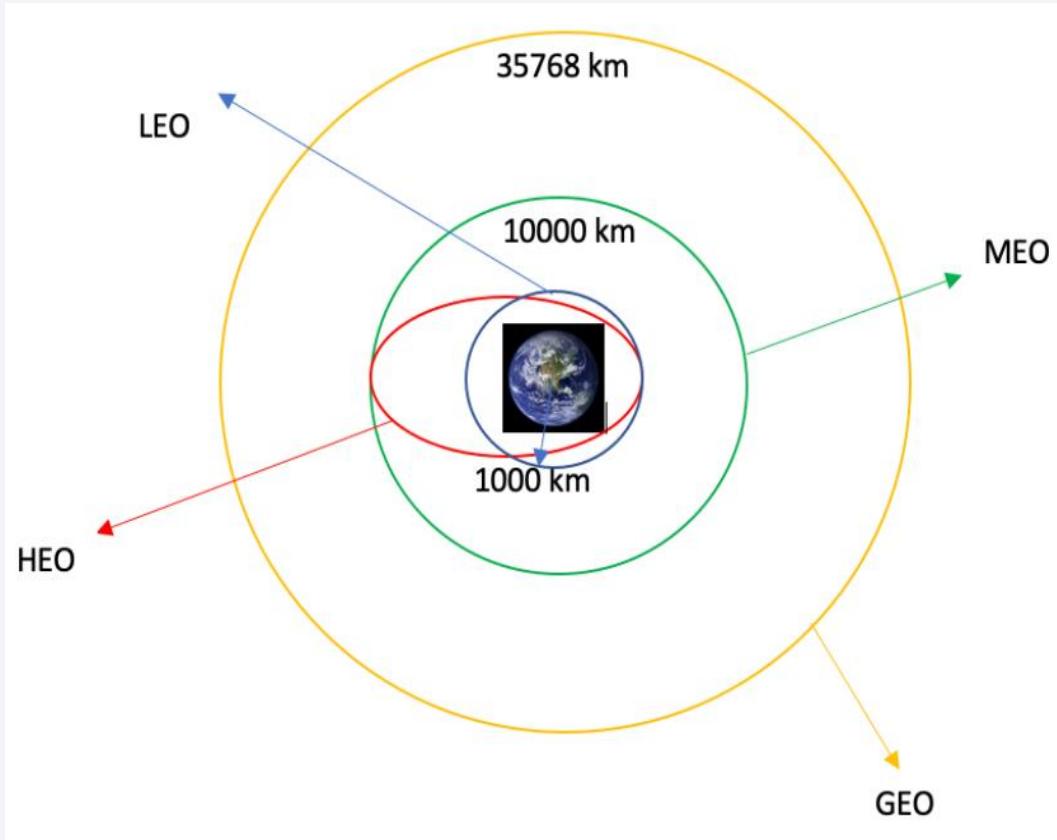
# Apply find_all() function with `th` element on first_launch_table
element = soup.find_all('th')

# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
    # 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)
    except:
        pass
```

From:

[https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX\\_Falcon9\\_Data\\_Collection\\_with\\_Web\\_Scraping.ipynb](https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX_Falcon9_Data_Collection_with_Web_Scraping.ipynb)

# Data Wrangling

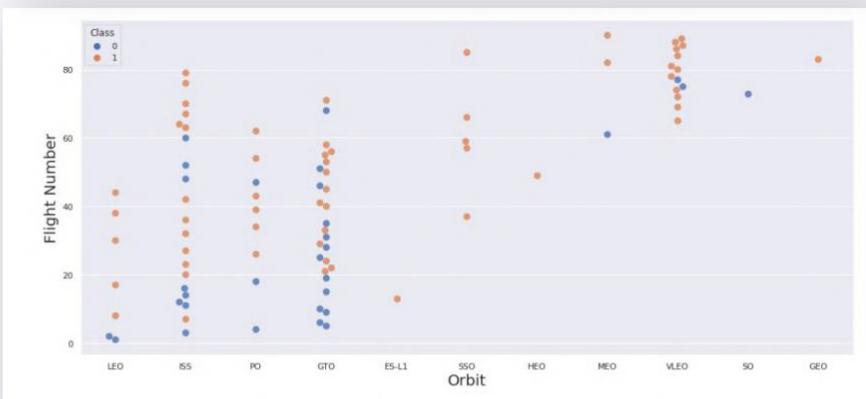
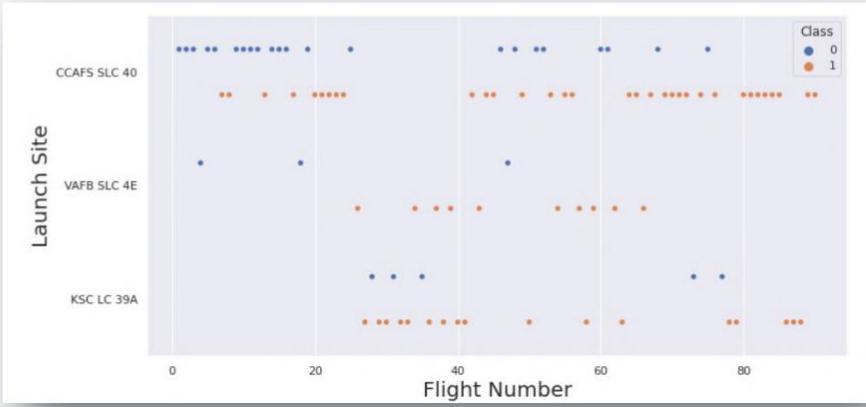


- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits.
- We created landing outcome label from outcome column and exported the results to csv.

The link to the notebook is:

<https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX%20Falcon9%20Data%20Wrangling.ipynb>

# 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.

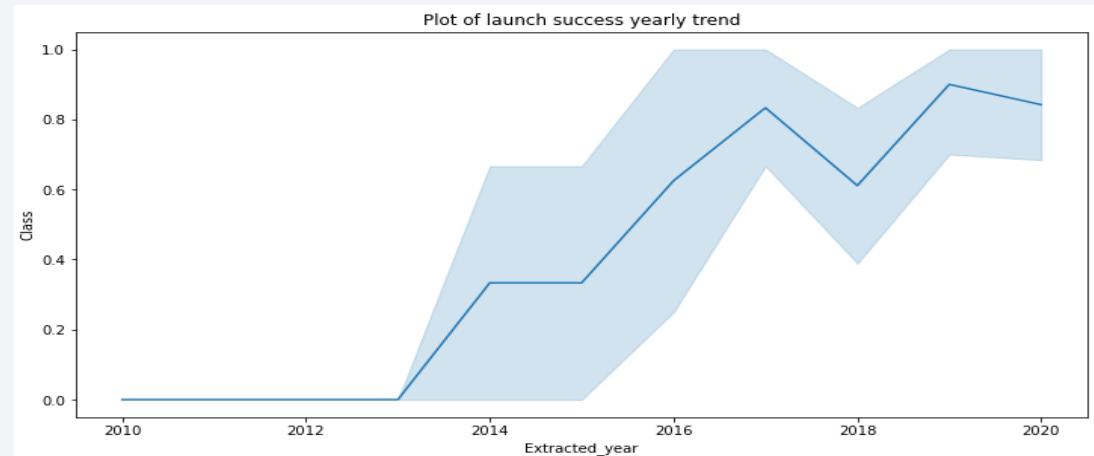
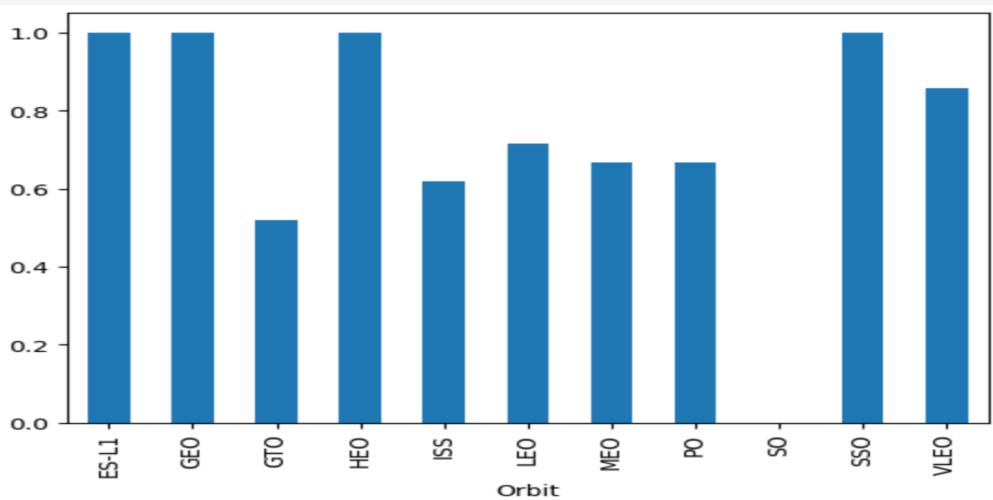
Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes.

The link to the notebook is:

<https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX%20Falcon9%20EDA%20with%20visualization.ipynb>

# 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 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 the trends or patterns of the attribute over time which in this case, is used to see the launch success yearly trend.
- We then use Feature Engineering to be used in success prediction in the future module by creating the dummy variables to categorical columns.
- From: <https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX%20Falcon9%20EDA%20with%20visualization.ipynb>

# EDA with SQL

---

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

The link to the notebook is: <https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/SpaceX%20Falcon9%20EDA.ipynb>

# Build an Interactive Map with Folium

---

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - How close the launch sites with railways, highways, and coastlines?
  - How close the launch sites with nearby cities?

Link to the notebook is:

<https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb>

# Build a Dashboard with Plotly Dash

---

- We built an interactive dashboard with Plotly dash.
- 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.

The link to the notebook is: [https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/spacex\\_dash\\_app.py](https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/spacex_dash_app.py)

# Predictive Analysis (Classification)

---

## Building the Model

- 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.

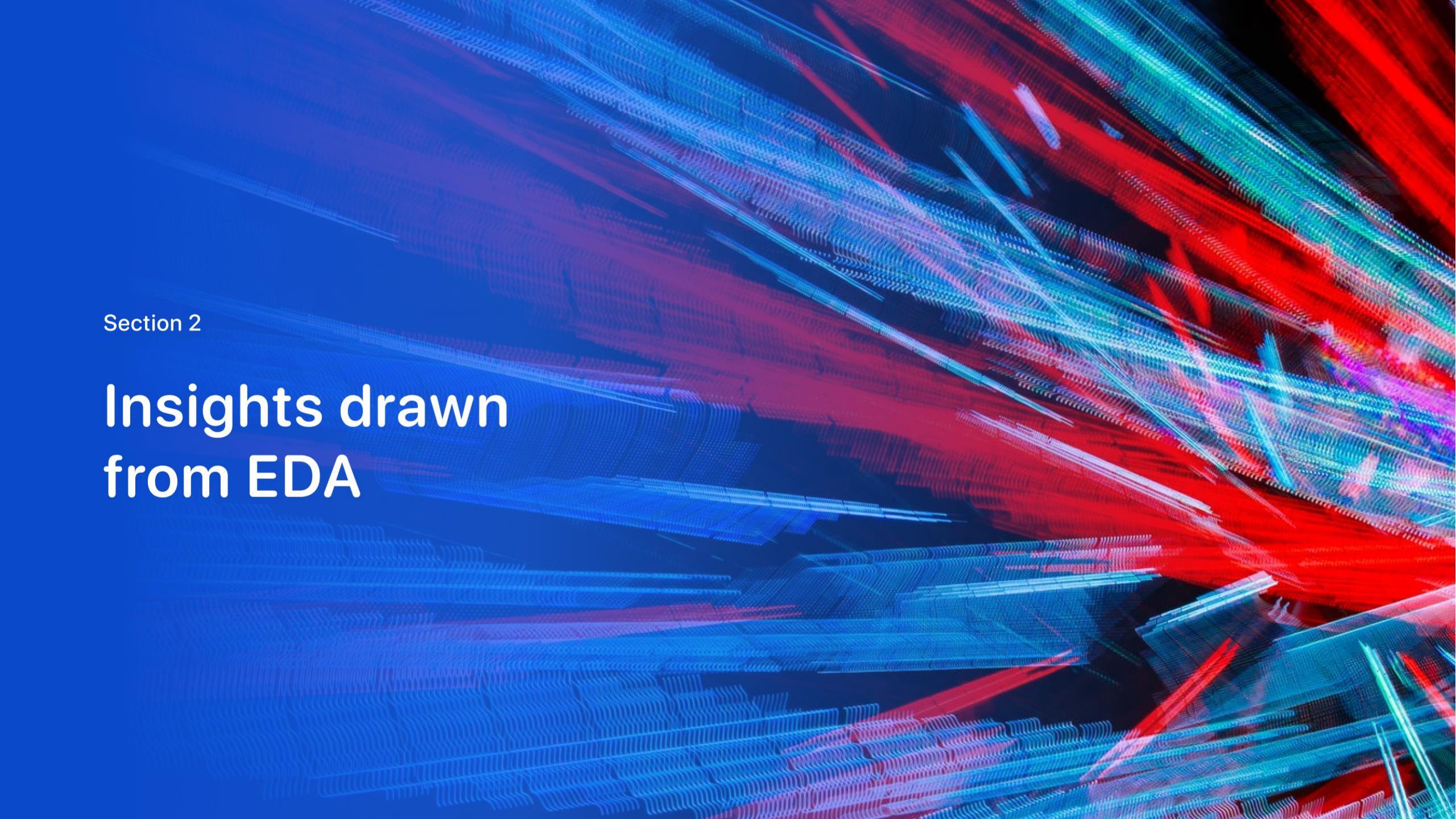
The link to the notebook is:

[https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/Predictive\\_Analysis\\_Machine\\_Learning.ipynb](https://github.com/SuyogV1999/Data-Science-Capstone/blob/main/Predictive_Analysis_Machine_Learning.ipynb)

# Results

---

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

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

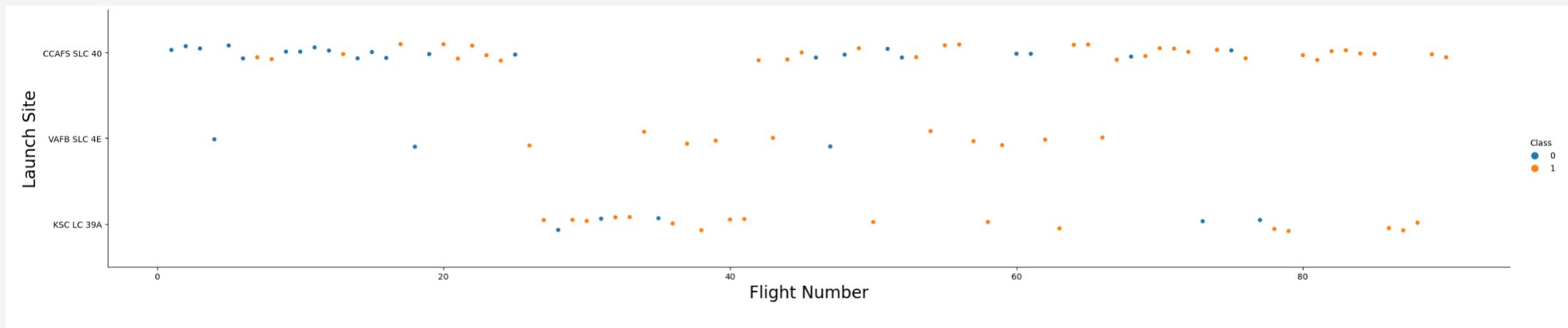
Section 2

## Insights drawn from EDA

# Flight Number vs. Launch Site

---

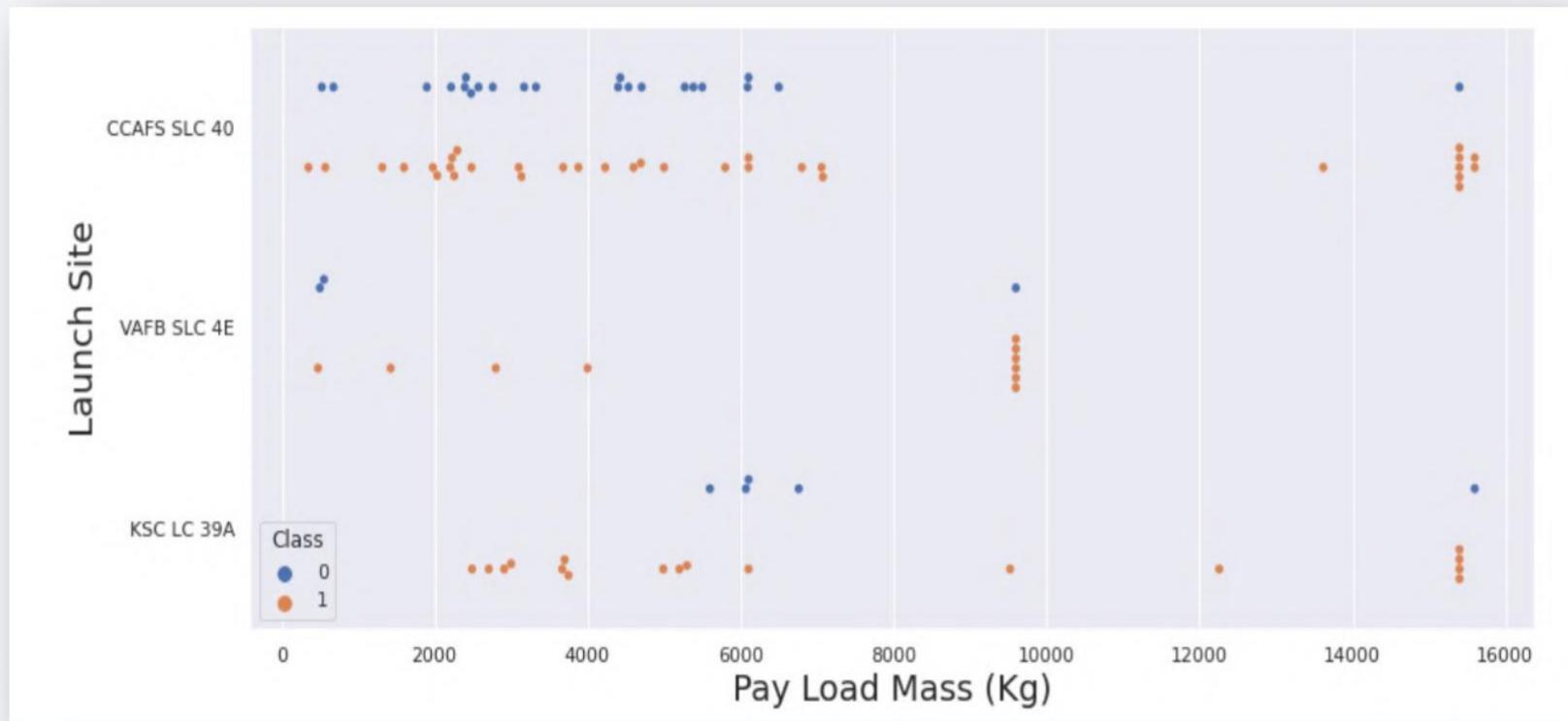
- From the plot, we found that the larger the flight amount of the launch site, the greater the success will be.



# 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.

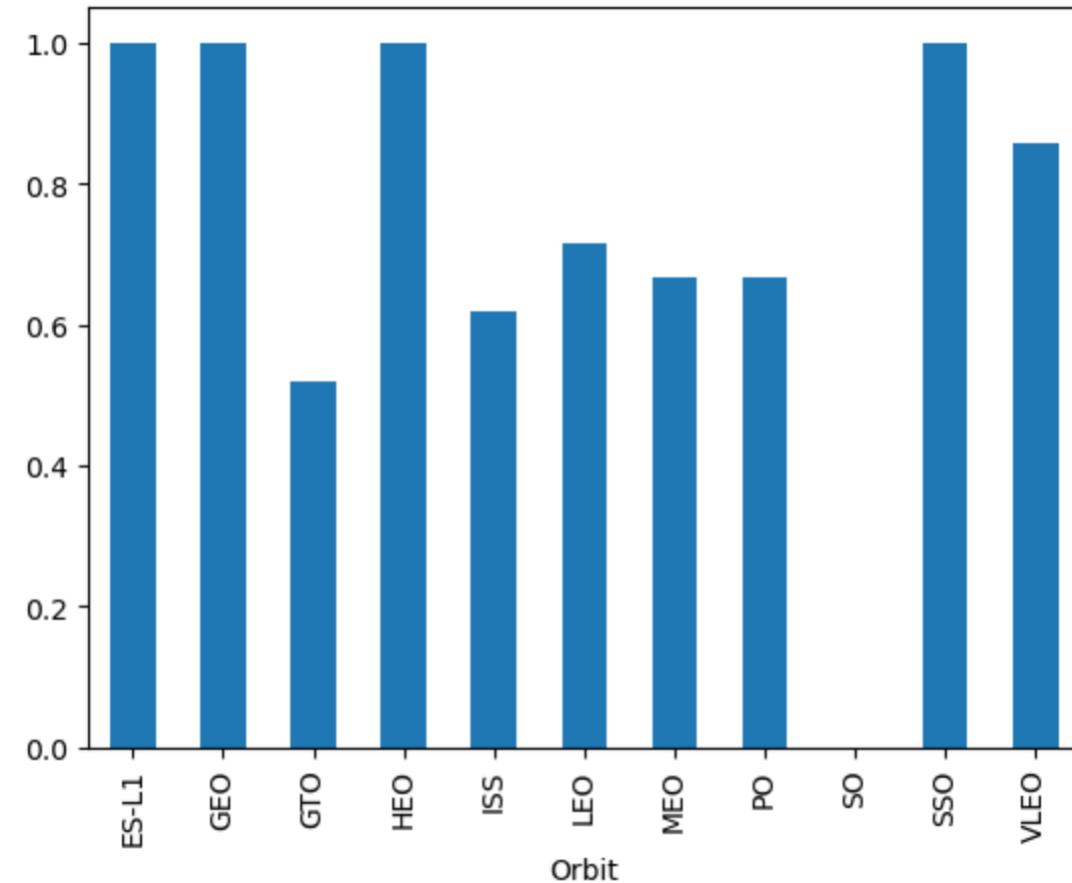
However, there is no clear pattern to say the launch site is dependent to the pay load mass for the success rate.



# 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.

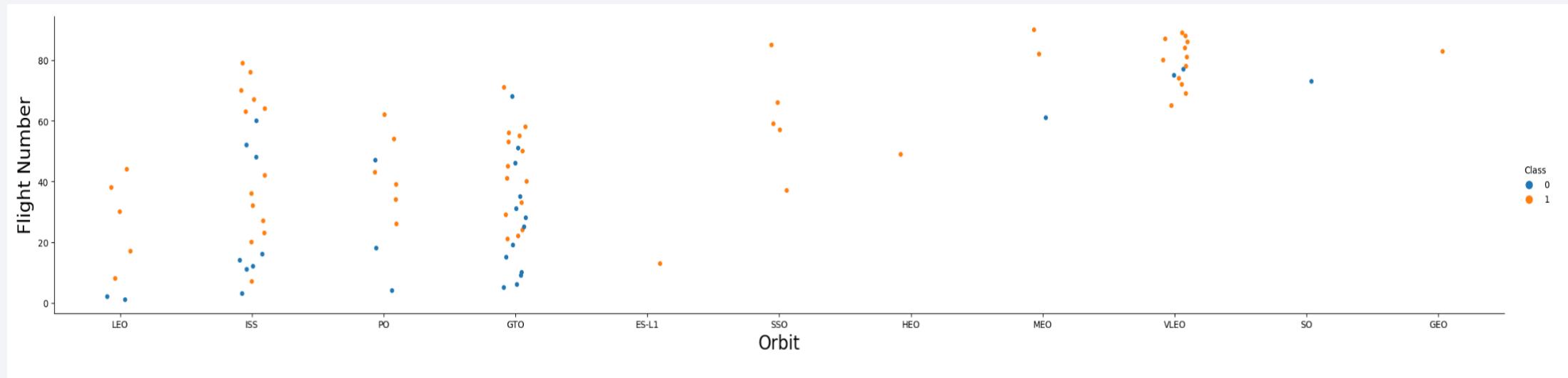
However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.



# Flight Number vs. Orbit Type

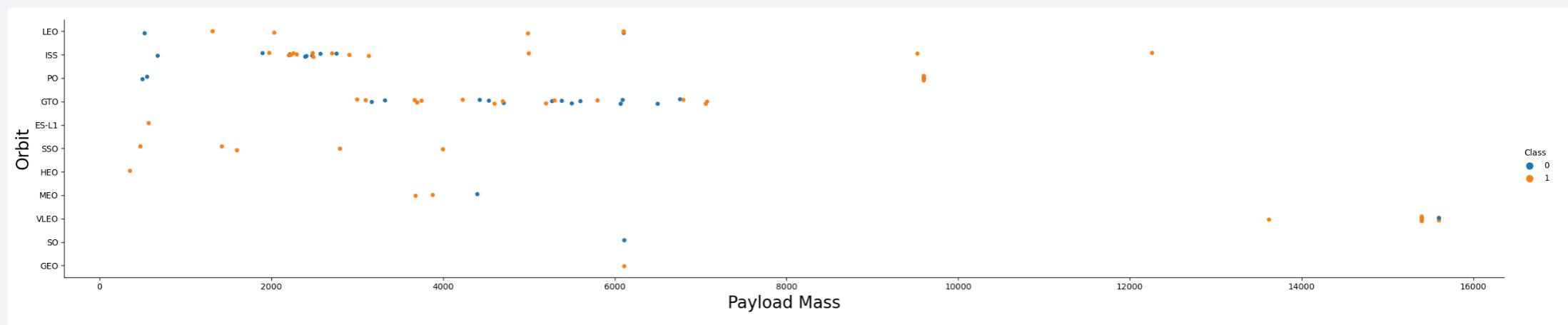
This scatter plot shows that generally, the larger the flight number on each orbits, the greater the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes.

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



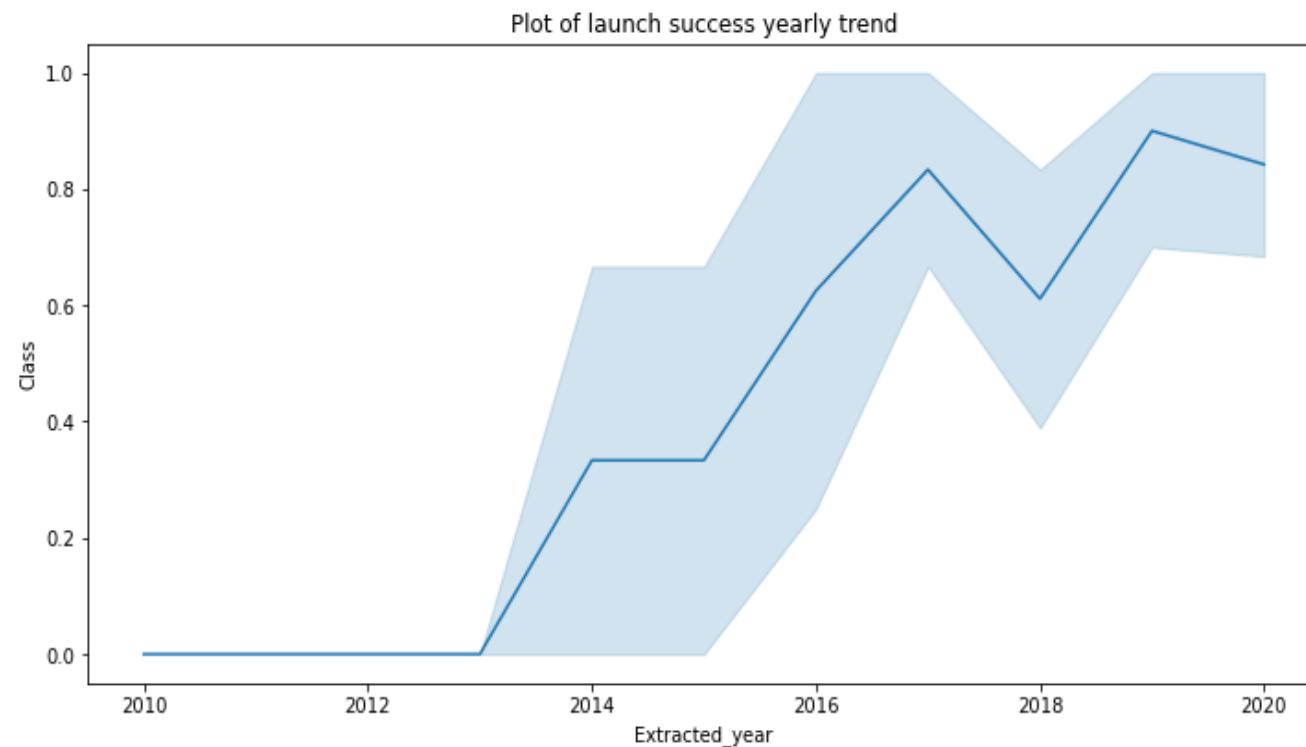
# Payload vs. Orbit Type

- 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.

In [5]:

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;  
  
* ibm_db_sa://zpw86771:***@fdbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3  
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

Out[5]: Launch\_Sites

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

# Launch Site Names Begin with 'CCA'

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

```
In [11]: task_2 = """
    SELECT *
    FROM SpaceX
    WHERE LaunchSite LIKE 'CCA%'
    LIMIT 5
"""
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- We used the query above to display 5 records where launch sites begin with 'CCA'

# 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)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)
```

```
* ibm_db_sa://zpw86771:***@fbdb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

**Total Payload Mass by NASA (CRS)**

---

45596

# Average Payload Mass by F9 v1.1

---

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

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster  
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
* ibm_db_sa://zpw86771:***@fdb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3  
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

**Average Payload Mass by Booster Version F9 v1.1**

---

2928

# 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<sup>nd</sup> December 2015

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad  
WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

```
* ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3  
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

**First Succesful Landing Outcome in Ground Pad**

---

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING_OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS_KG_ > 4000 AND PAYLOAD_MASS_KG_ < 6000;
```

```
* ibm_db_sa://zpw86771:***@fdb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
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.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
* ibm_db_sa://zpw86771:***@fdb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

**Successful Mission**

---

100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
* ibm_db_sa://zpw86771:***@fdb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb  
Done.
```

**Failure Mission**

---

1

# Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX  
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);  
  
* ibm_db_sa://zpw86771:***@fdb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lgde00.databases.appdomain.cloud:32731/bludb  
Done.  
  
Booster Versions which carried the Maximum Payload Mass  
F9 B5 B1048.4  
F9 B5 B1048.5  
F9 B5 B1049.4  
F9 B5 B1049.5  
F9 B5 B1049.7  
F9 B5 B1051.3  
F9 B5 B1051.4  
F9 B5 B1051.6  
F9 B5 B1056.4  
F9 B5 B1058.3  
F9 B5 B1060.2  
F9 B5 B1060.3
```

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
* ibm_db_sa://zpw86771:***@fdb88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.
databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

---

```
%sql SELECT LANDING_OUTCOME as "Landing Outcome", COUNT(LANDING_OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING_OUTCOME \
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

```
* ibm_db_sa://zpw86771:***@fb88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

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.

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where a large urban area is illuminated. In the upper right corner, there is a faint, greenish glow of the aurora borealis or a similar atmospheric phenomenon.

Section 3

# Launch Sites Proximities Analysis

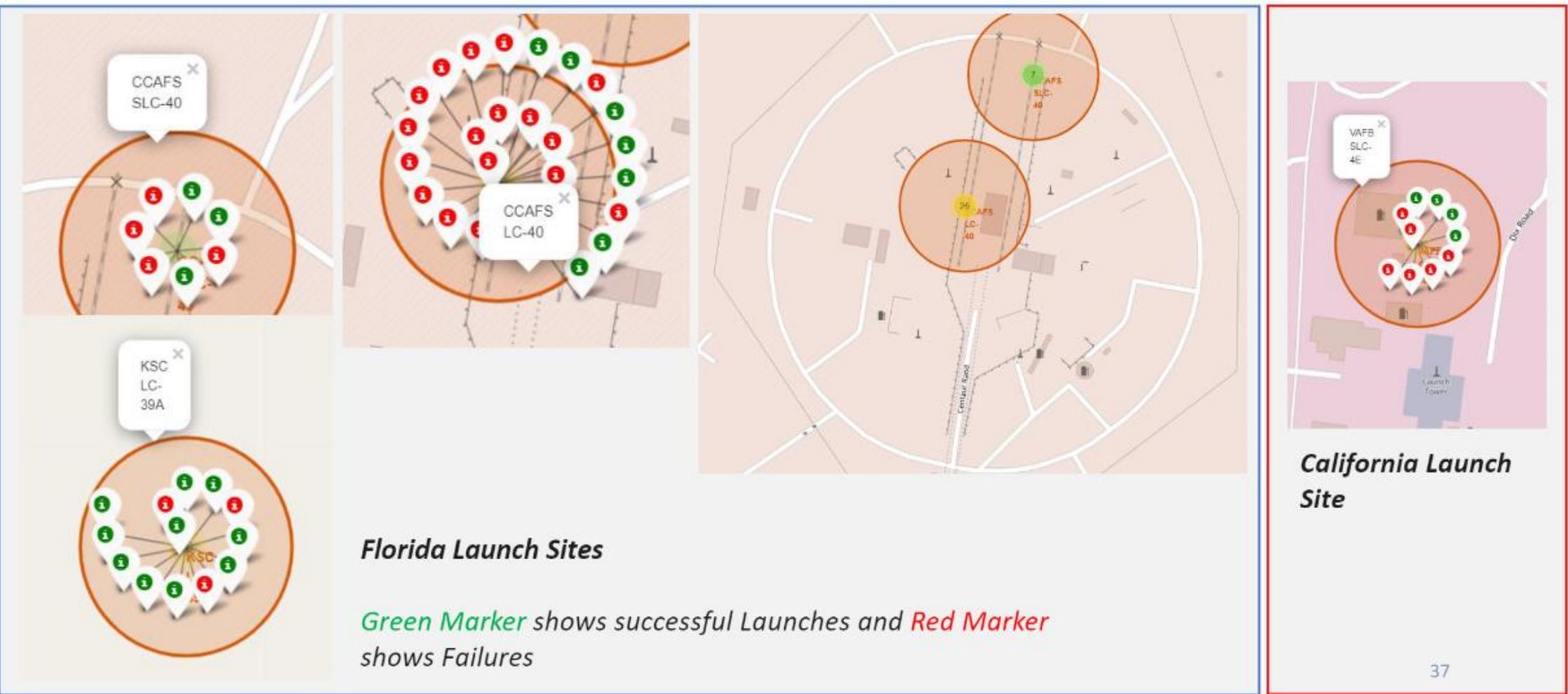
# Location of all the Launch Sites

---



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

# Markers showing launch sites with color labels



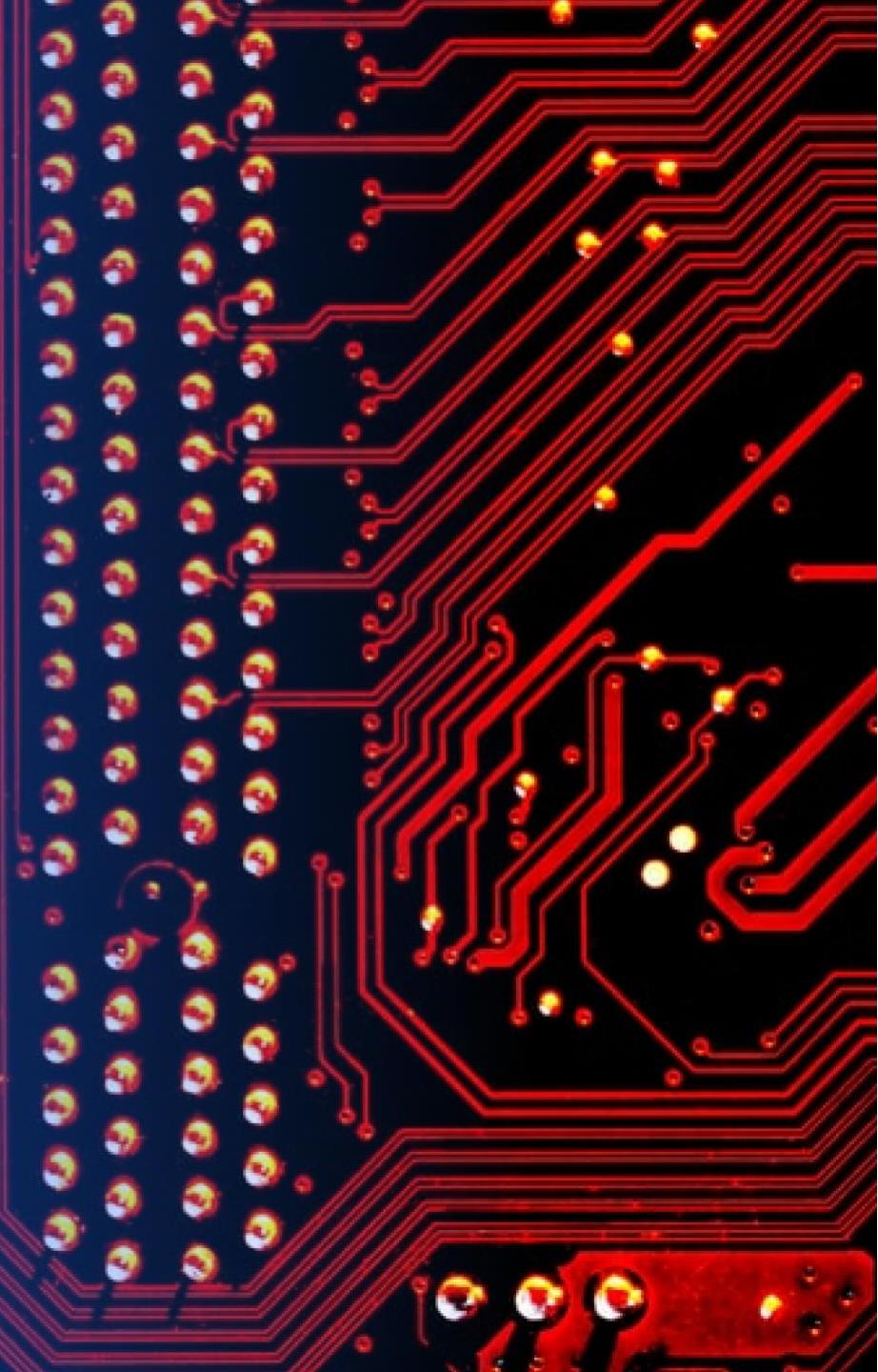
# Launch Site distance to landmarks



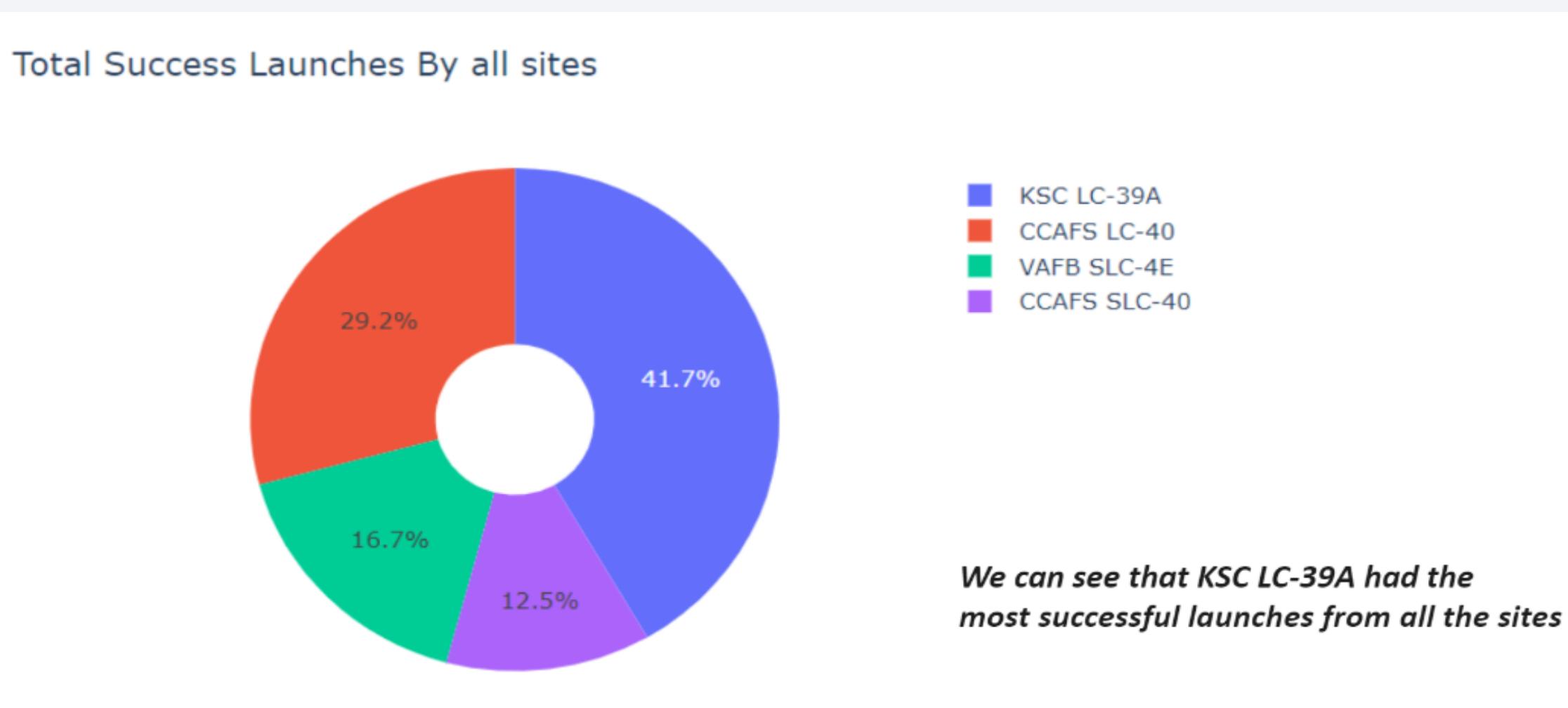
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

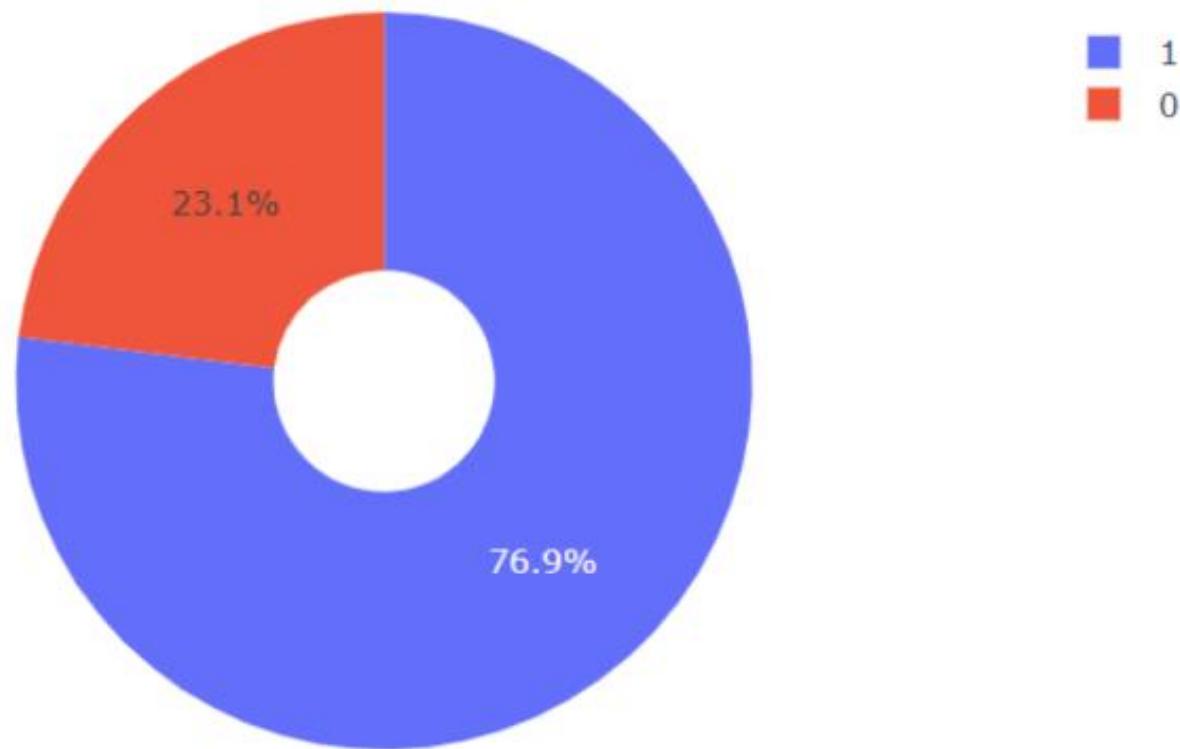
# Build a Dashboard with Plotly Dash



## Pie chart showing the success percentage achieved by each launch site

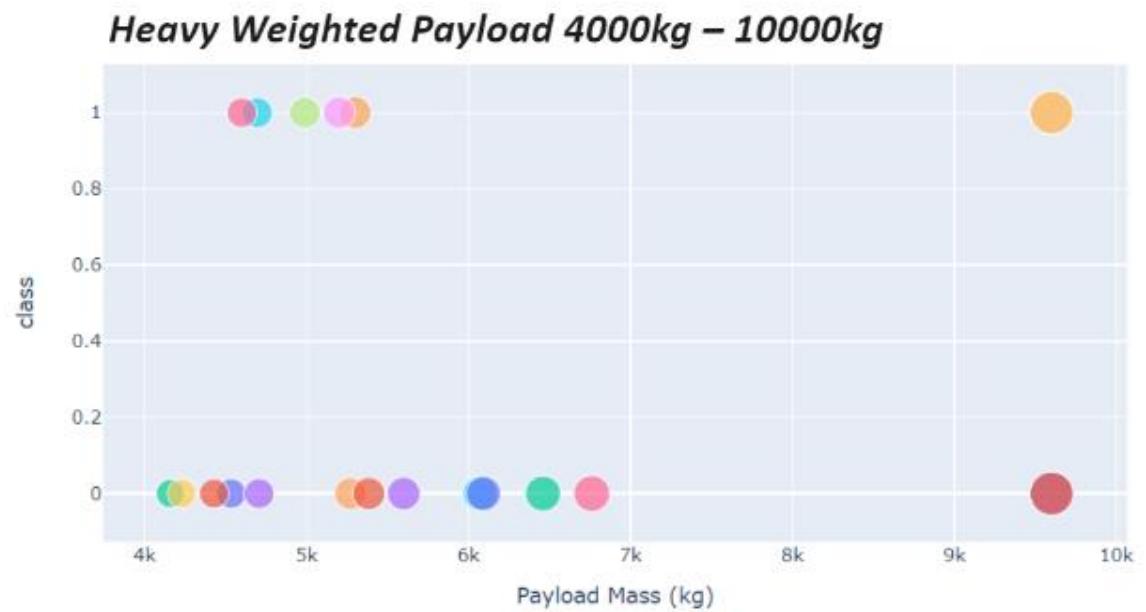
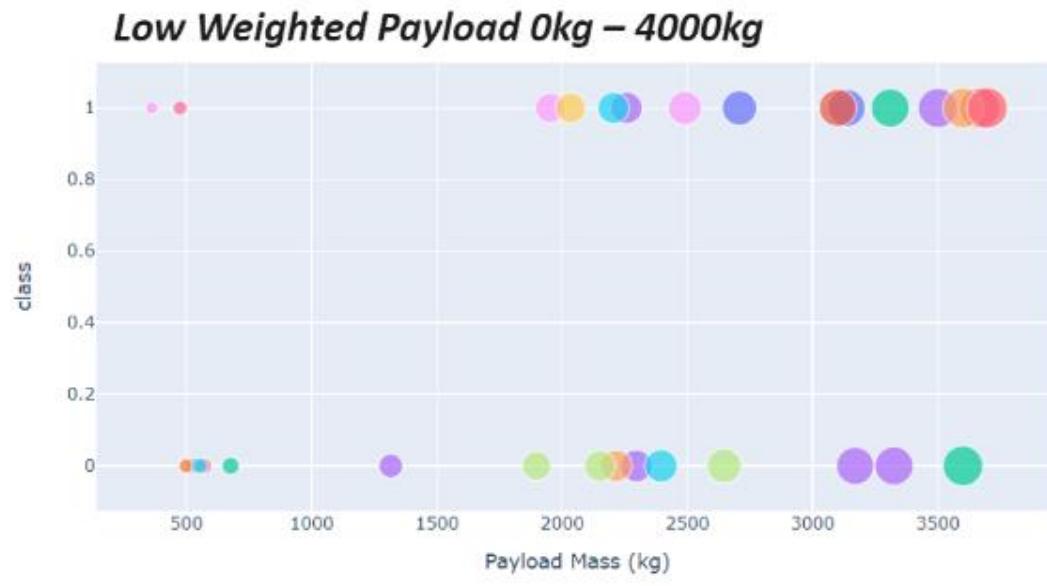


## Pie chart showing the Launch site with the highest launch success ratio



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

## Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines that transition from a bright yellow-green at the top right to a deep blue at the bottom left. These curves are set against a lighter, off-white background that has subtle diagonal stripes, giving it a sense of motion or depth.

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

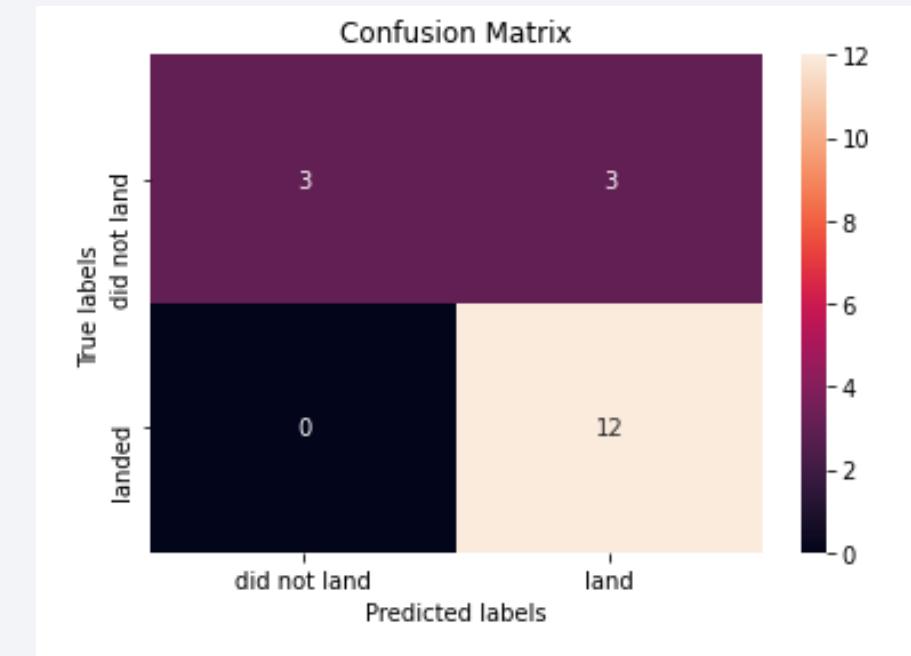
```
models = {'KNeighbors':knn_cv.best_score_,  
          'DecisionTree':tree_cv.best_score_,  
          'LogisticRegression':logreg_cv.best_score_,  
          'SupportVector': svm_cv.best_score_}  
  
bestalgorithm = max(models, key=models.get)  
print('Best model is', bestalgorithm, 'with a score of', models[bestalgorithm])  
if bestalgorithm == 'DecisionTree':  
    print('Best params is :', tree_cv.best_params_)  
if bestalgorithm == 'KNeighbors':  
    print('Best params is :', knn_cv.best_params_)  
if bestalgorithm == 'LogisticRegression':  
    print('Best params is :', logreg_cv.best_params_)  
if bestalgorithm == 'SupportVector':  
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856  
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

# Confusion Matrix

---

- The confusion matrix for the decision tree classifier 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!

