



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Mohamed ayoub Essalami  
9<sup>th</sup> September 2024



# Outline

---

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

# Executive Summary

---

## Summary of methodologies

- Gathering Data via API
- Gathering Data through Web Scraping
- Data Cleaning and Preparation
- Exploratory Data Analysis using SQL
- Exploratory Data Analysis using Data Visualizations
- Interactive Visual Analysis using Folium
- Machine Learning Forecasting
- Overview of All Findings

## Summary of all results

- Results from Exploratory Data Analysis
- Screenshots of Interactive Analytics
- Results of Predictive Analytics

# Introduction

---

- Project background and context

- SpaceX promotes its Falcon 9 rocket launches on its website at a price of \$62 million, while competitors charge at least \$165 million each. A significant portion of the cost savings comes from *SpaceX's* ability to reuse the first stage of the rocket. By predicting whether the first stage will land successfully, we can estimate the launch cost. This insight can be beneficial for other companies looking to compete with SpaceX for rocket launch contracts. The *project's* objective is to develop a machine learning pipeline to forecast the successful landing of the first stage.

- Problems you want to find answers

## Factors Influencing Successful Rocket Landings

- Key Determinants: Various features interact to influence the likelihood of a successful landing.
  - Operating Conditions: Specific operating conditions must be met to ensure the effectiveness of the landing program.



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 has been utilized for the categorical variables.
- 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

---

- Describe how data sets were collected.
  - Data collection was carried out by making GET requests to the SpaceX API. Subsequently, we decoded the response content as JSON using the `.json()` function and transformed it into a pandas DataFrame using `.json_normalize()`. We then cleaned the data, checked for any missing values, and filled in those missing values as needed. Additionally, we conducted web scraping from Wikipedia to gather Falcon 9 launch records using BeautifulSoup. The goal was to extract the launch records from an HTML table, parse the table, and convert it into a pandas DataFrame for further analysis.
- You need to present your data collection process use key phrases and flowcharts

# Data Collection – SpaceX API

- <https://github.com/mohamedayoub97/Peer-graded-Assignment.git>

```
1. Get request for rocket launch data using API

In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)

2. Use json_normalize method to convert json result to dataframe

In [12]: # Use json_normalize method to convert the json result into a dataframe
          # decode response content as json
          static_json_df = res.json()

In [13]: # apply json_normalize
          data = pd.json_normalize(static_json_df)

3. We then performed data cleaning and filling in the missing values

In [30]: rows = data_falcon9['PayloadMass'].values.tolist()[0]

          df_rows = pd.DataFrame(rows)
          df_rows = df_rows.replace(np.nan, PayloadMass)

          data_falcon9['PayloadMass'][0] = df_rows.values
          data_falcon9
```



# Data Collection - Scraping

- We utilized web scraping techniques with BeautifulSoup to gather Falcon 9 launch records. After extracting the data, we parsed the table and converted it into a pandas DataFrame.
- <https://github.com/mohamedayoub97/Peer-graded-Assignment.git>

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [5]: # use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

```
Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the BeautifulSoup object was created properly

```
In [7]: # Use soup.title attribute
soup.title
```

```
Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

```
In [10]: column_names = []

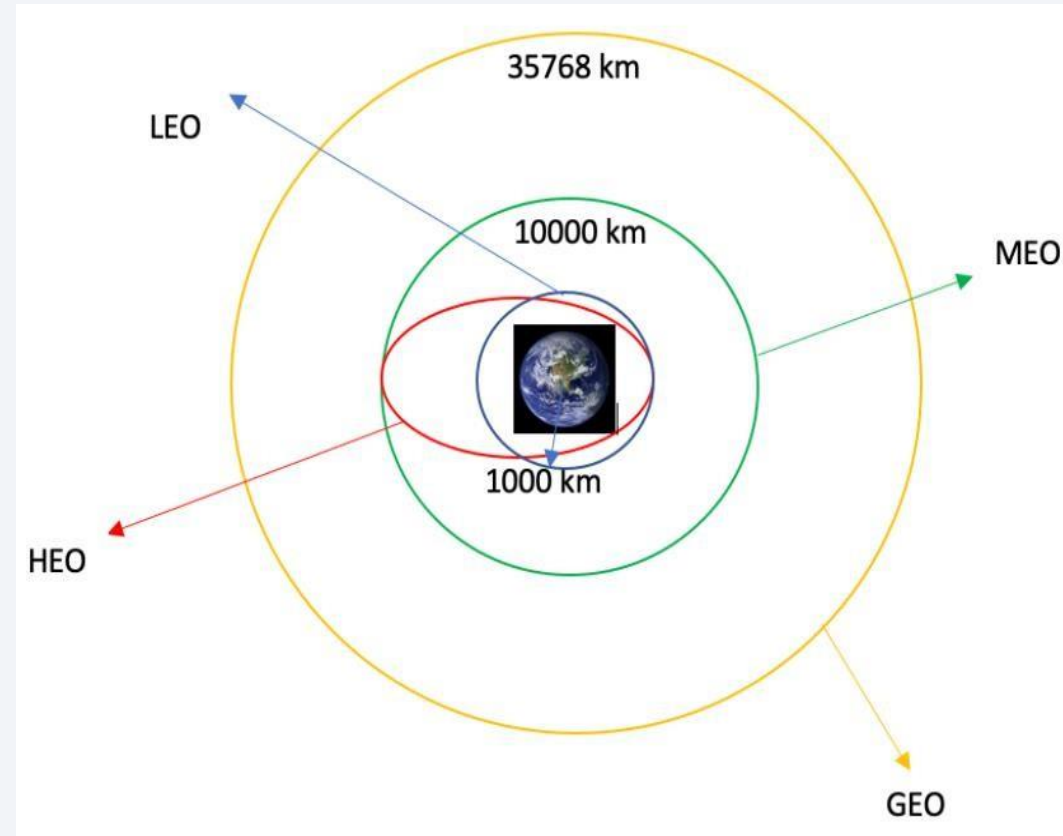
# Apply find_all() function with 'th' element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name ('if name is not None and len(name) > 0') into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

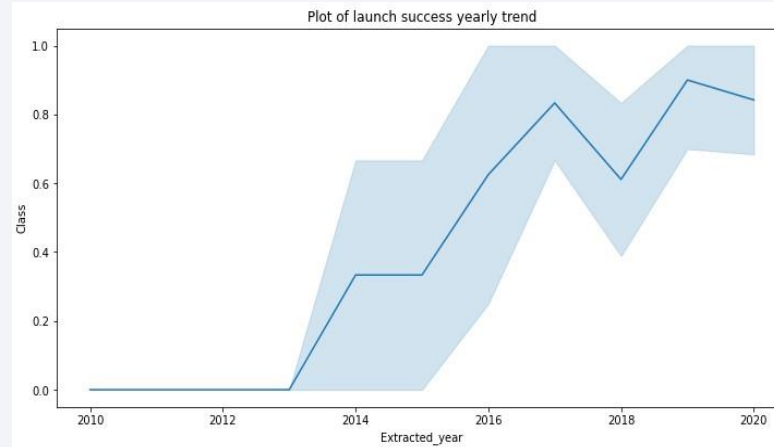
# Data Wrangling

- We conducted exploratory data analysis to identify the training labels. We counted the total number of launches at each site as well as the frequency of each orbit. Additionally, we generated a landing outcome label from the outcome column and exported the results to a CSV file.
- <https://github.com/mohamedayoub97/Peer-graded-Assignment.git>

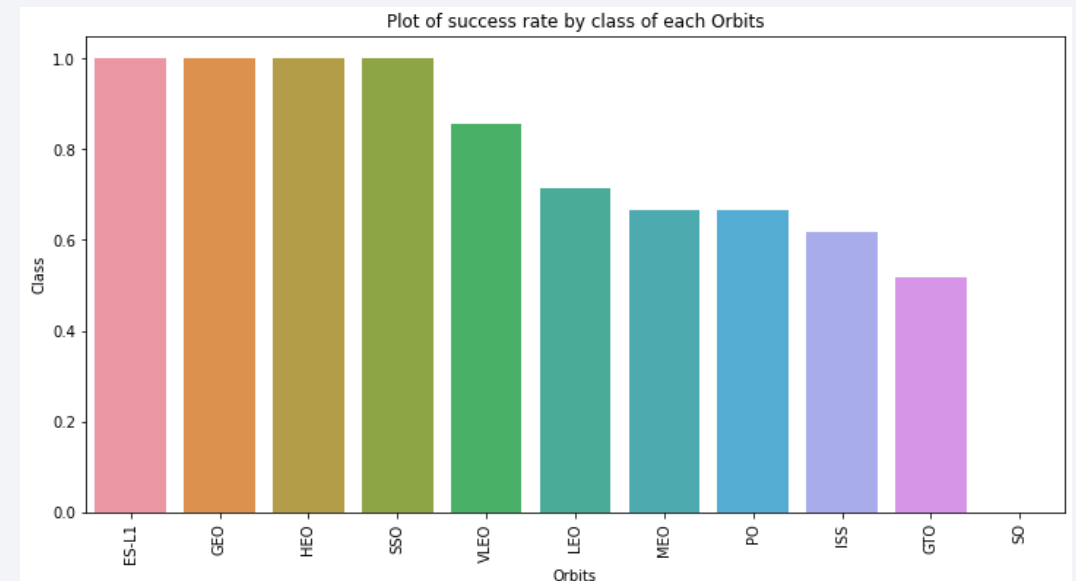


# EDA with Data Visualization

- We examined the data by visualizing various relationships, including the correlation between flight number and launch site, payload versus launch site, success rates for each type of orbit, flight number in relation to orbit type, and the annual trends in launch success.



<https://github.com/mohamedayoub97/Peer-graded-Assignment.git>



# EDA with SQL

---

- We imported the SpaceX dataset into a PostgreSQL database directly from the Jupyter Notebook.
- We conducted exploratory data analysis (EDA) using SQL to glean insights from the data. We formulated queries to determine, for example:
  - The names of distinct launch sites involved in the space missions.
  - The total payload mass transported by NASA-launched boosters (CRS).
  - The average payload mass carried by the F9 v1.1 booster version.
  - The overall count of successful and failed mission outcomes.
  - The details of failed landing outcomes on the drone ship, including the corresponding booster versions and launch site names.
- <https://github.com/mohamedayoub97/Peer-graded-Assignment.git>

# Build an Interactive Map with Folium

---

- We marked all launch sites on the Folium map and incorporated map objects such as markers, circles, and lines to denote the success or failure of launches at each site. We categorized the launch outcomes, assigning class 0 for failure and class 1 for success. By using color-coded marker clusters, we identified launch sites with relatively high success rates. Additionally, we calculated the distances from each launch site to nearby landmarks and addressed questions such as:
  - Are launch sites located near railways, highways, and coastlines?
  - Do launch sites maintain a certain distance from urban areas?

-<https://github.com/mohamedayoub97/Peer-graded-Assignment.git>



# Build a Dashboard with Plotly Dash

---

- We developed an interactive dashboard using Plotly Dash. Within this dashboard, we displayed pie charts illustrating the total number of launches by specific sites. Additionally, we created scatter plots to visualize the relationship between launch outcomes and payload mass (in kilograms) for various booster versions.
- <https://github.com/mohamedayoub97/Peer-graded-Assignment.git>

# Predictive Analysis (Classification)

---

- We imported the data using NumPy and pandas, transformed the dataset, and divided it into training and testing sets. We developed various machine learning models and optimized their hyperparameters using GridSearchCV. Using accuracy as the evaluation metric, we enhanced the model through feature engineering and algorithm tuning. Ultimately, we identified the best-performing classification model.
- <https://github.com/mohamedayoub97/Peer-graded-Assignment.git>

# Results

---

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



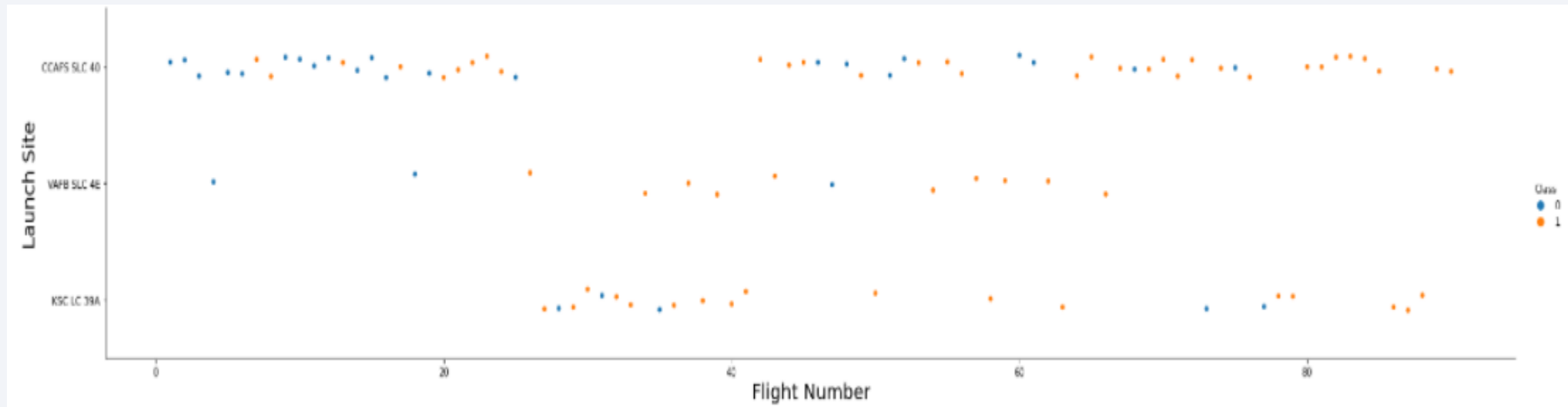
The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue, red, and cyan on the right. These streaks are layered over a faint, dark grid pattern, creating a sense of depth and movement.

Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

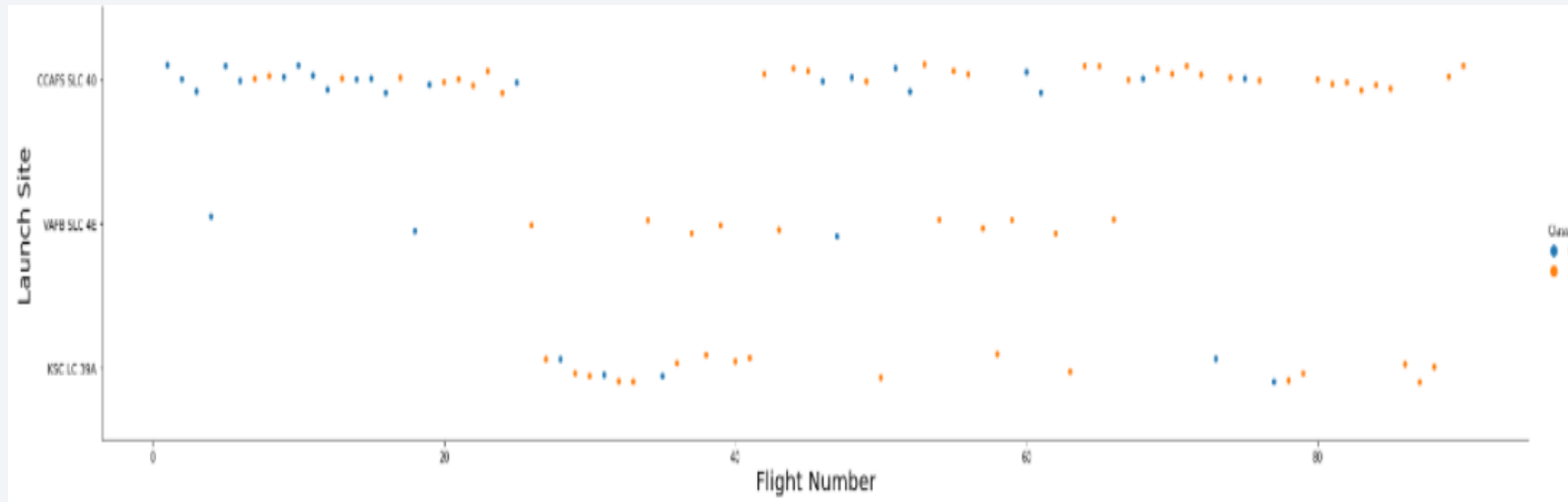


From the plot, we observed that an increase in the number of flights at a launch site correlates with a higher success rate for that site.



# Payload vs. Launch Site

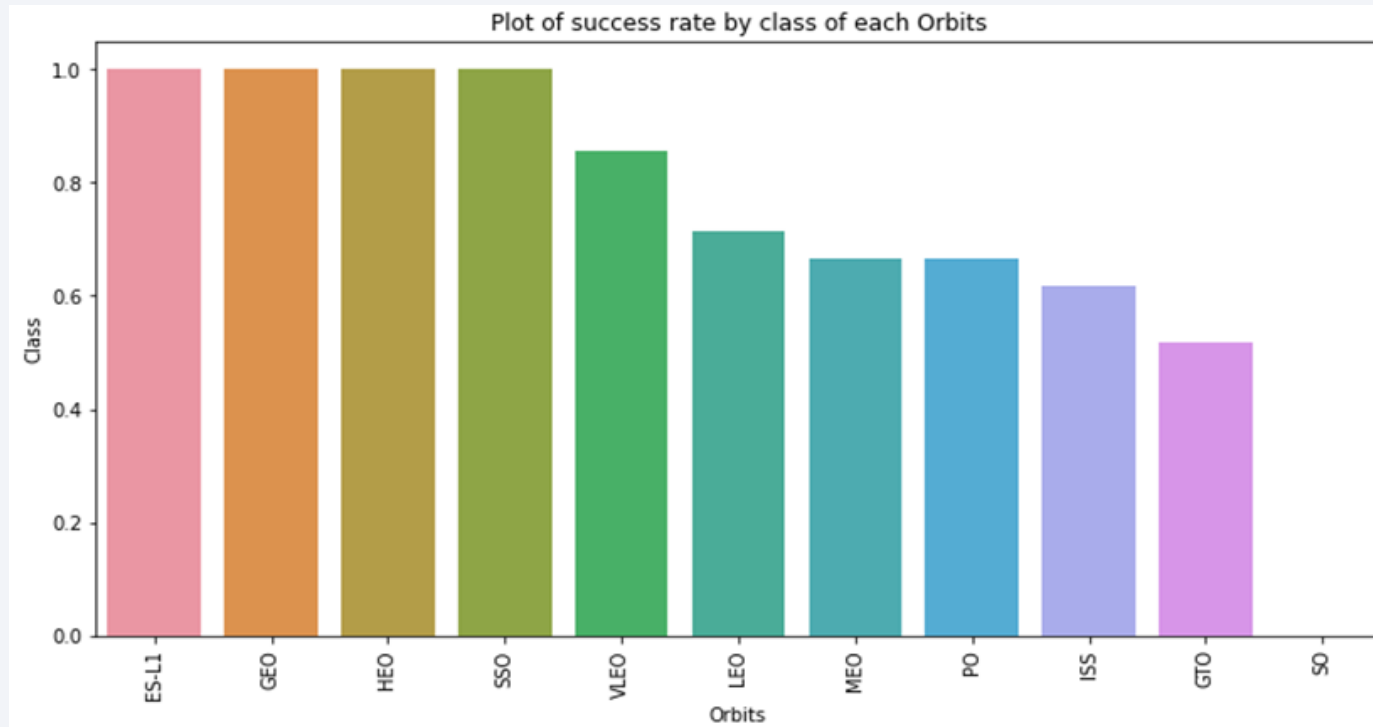
---



The greater the payload the higher the success rate.

# Success Rate vs. Orbit Type

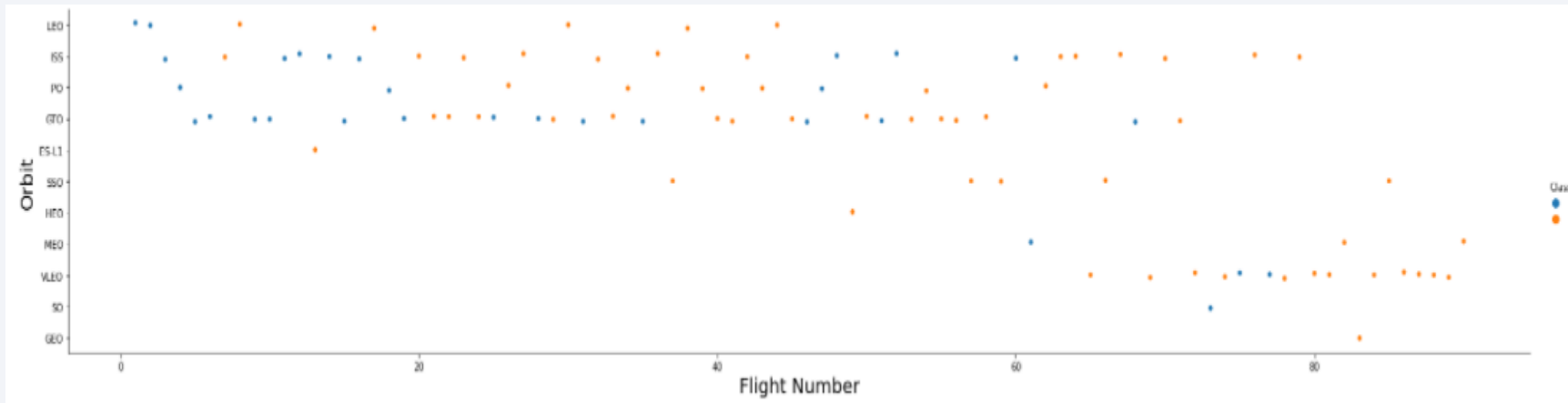
---



- ES-L1, GEO, HEO, SSO and VLEO had the most success rate

# Flight Number vs. Orbit Type

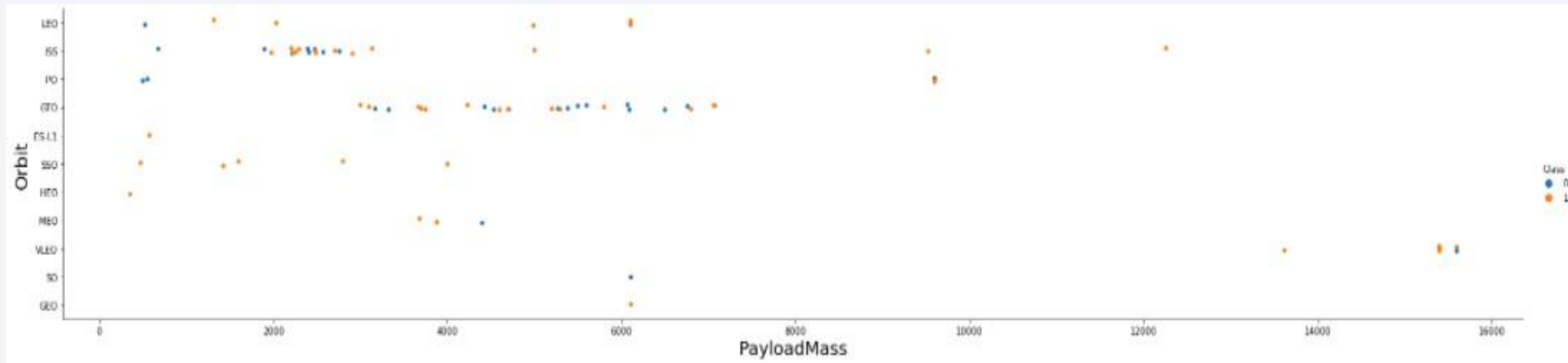
---



- The plot below illustrates the relationship between Flight Number and Orbit type. We note that in the LEO orbit, the success rate appears to be connected to the number of flights, while in the GTO orbit, there is no evident relationship between the flight number and the orbit.

# Payload vs. Orbit Type

---

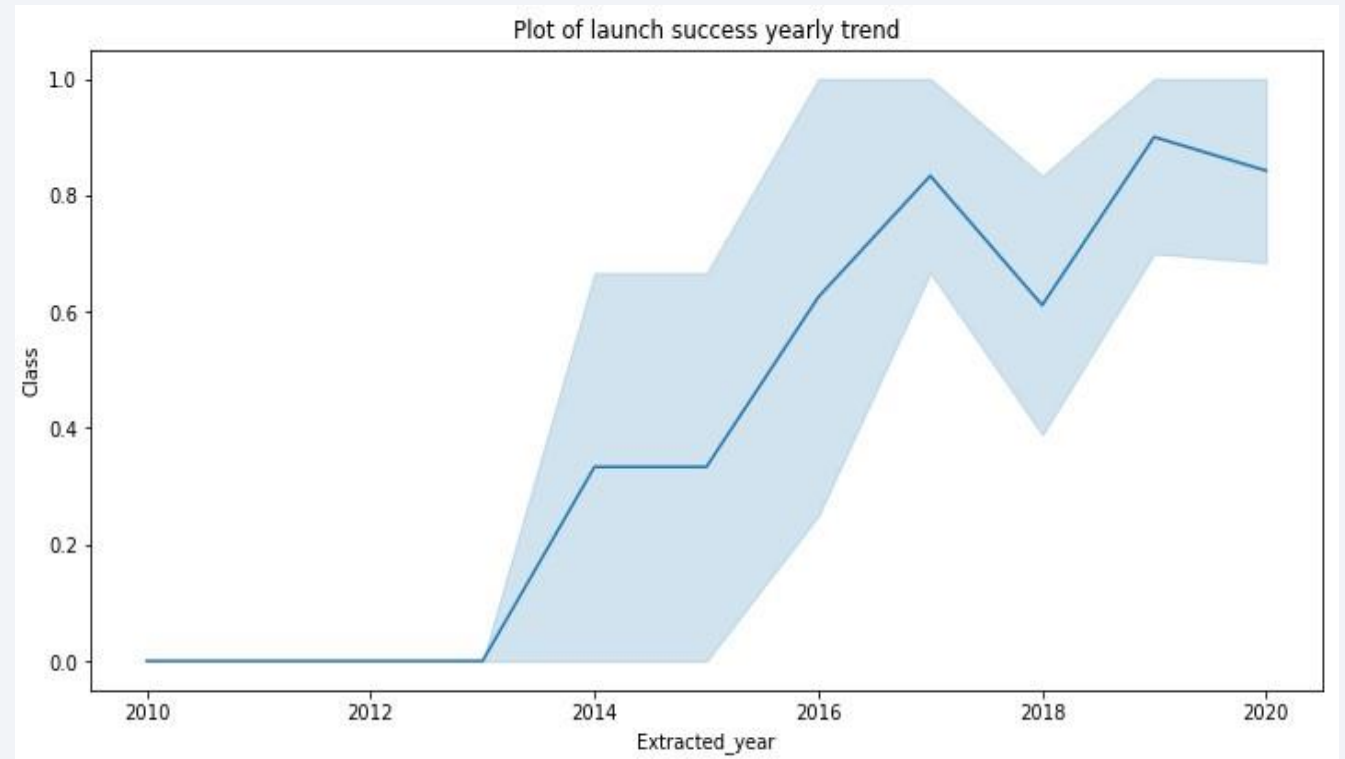


We can see that successful landings are more frequent for heavy payloads in the PO, LEO, and ISS orbits.

# Launch Success Yearly Trend

---

- Success rate increased from 2013 till 2020





# All Launch Site Names

---

KEYWORD - DISTINCT

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

```
In [10]: task_1 = '''  
          SELECT DISTINCT LaunchSite  
          FROM SpaceX  
          ...  
          create_pandas_df(task_1, database=conn)
```

```
Out[10]:
```

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

# Launch Site Names Begin with 'CCA'

- Query using launch site beginning 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

# Total Payload Mass

---

We determined that the total payload carried by NASA boosters is 45,596, using the query provided below.

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

In [12]:

```
task_3 = '''
    SELECT SUM(PayloadMassKG) AS Total_PayloadMass
    FROM SpaceX
    WHERE Customer LIKE 'NASA (CRS)'
    '''

create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

# Average Payload Mass by F9 v1.1

---

We found that the average payload mass carried by the booster version F9 v1.1 is 2,928.4.

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

In [13]: task_4 = '''
          SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
          FROM SpaceX
          WHERE BoosterVersion = 'F9 v1.1'
          '''
          create_pandas_df(task_4, database=conn)

Out[13]:
```

	avg_payloadmass
0	2928.4

# First Successful Ground Landing Date

---

We noted that the date of the first successful landing outcome on the ground pad was December 22, 2015.

```
In [14]: task_5 = '''
          SELECT MIN(Date) AS FirstSuccessfull_landing_date
          FROM SpaceX
          WHERE LandingOutcome LIKE 'Success (ground pad)'
          '''
          create_pandas_df(task_5, database=conn)
```

```
Out[14]:
```

	firstsuccessfull_landing_date
0	2015-12-22



## Successful Drone Ship Landing with Payload between 4000 and 6000

---

```
In [15]: task_6 = '''
          SELECT BoosterVersion
          FROM SpaceX
          WHERE LandingOutcome = 'Success (drone ship)'
             AND PayloadMassKG > 4000
             AND PayloadMassKG < 6000
          ...
          create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

We utilized the WHERE clause to filter for boosters that successfully landed on the drone ship, applying an AND condition to specify a payload mass greater than 4,000 but less than 6,000.

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
In [16]: task_7a = '''
          SELECT COUNT(MissionOutcome) AS SuccessOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Success%'
          '''

          task_7b = '''
          SELECT COUNT(MissionOutcome) AS FailureOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Failure%'
          '''

          print('The total number of successful mission outcome is:')
          display(create_pandas_df(task_7a, database=conn))
          print()
          print('The total number of failed mission outcome is:')
          create_pandas_df(task_7b, database=conn)
```

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

```
Out[16]:
```

	failureoutcome
0	1

We employed the wildcard '%' in the WHERE clause to filter for records where the MissionOutcome was either a success or a failure.

# Boosters Carried Maximum Payload

We identified the booster that transported the maximum payload by using a subquery in the WHERE clause along with the MAX() function.

List the names of the booster\_versions which have carried the maximum payload mass. Use a subquery

```
In [17]: task_8 = '''
          SELECT BoosterVersion, PayloadMassKG
          FROM SpaceX
          WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
          ORDER BY BoosterVersion
          '''
          create_pandas_df(task_8, database=conn)
```

```
Out[17]:
```

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

# 2015 Launch Records

---

We employed a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes on the drone ship, along with their corresponding booster versions and launch site names for the year 2015.

List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [18]: task_9 = '''
          SELECT BoosterVersion, LaunchSite, LandingOutcome
          FROM SpaceX
          WHERE LandingOutcome LIKE 'Failure (drone ship)'
             AND Date BETWEEN '2015-01-01' AND '2015-12-31'
          ...
          create_pandas_df(task_9, database=conn)
```

```
Out[18]:
```

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

```
In [19]: task_10 = '''
          SELECT LandingOutcome, COUNT(LandingOutcome)
          FROM SpaceX
          WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
          GROUP BY LandingOutcome
          ORDER BY COUNT(LandingOutcome) DESC
          '''
          create_pandas_df(task_10, database=conn)
```

```
Out[19]:
```

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

We selected landing outcomes and the count of those outcomes from the dataset, using the WHERE clause to filter for landing outcomes between June 4, 2010, and March 20, 2010. We then applied the GROUP BY clause to group the landing outcomes and used the ORDER BY clause to sort the grouped landing outcomes in descending order.

A satellite view of Earth from space, showing the curvature of the planet and the glow of city lights at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

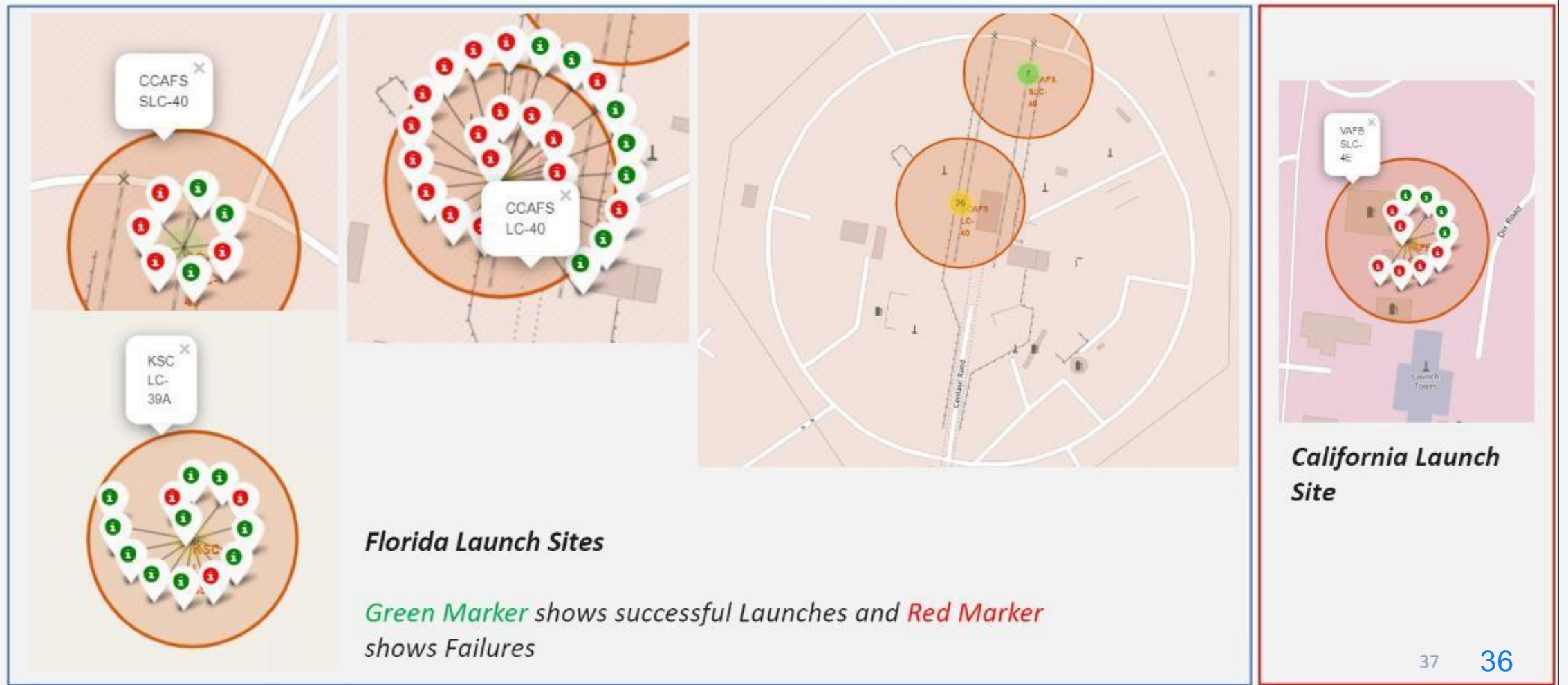


# All launch sites global map markers

---

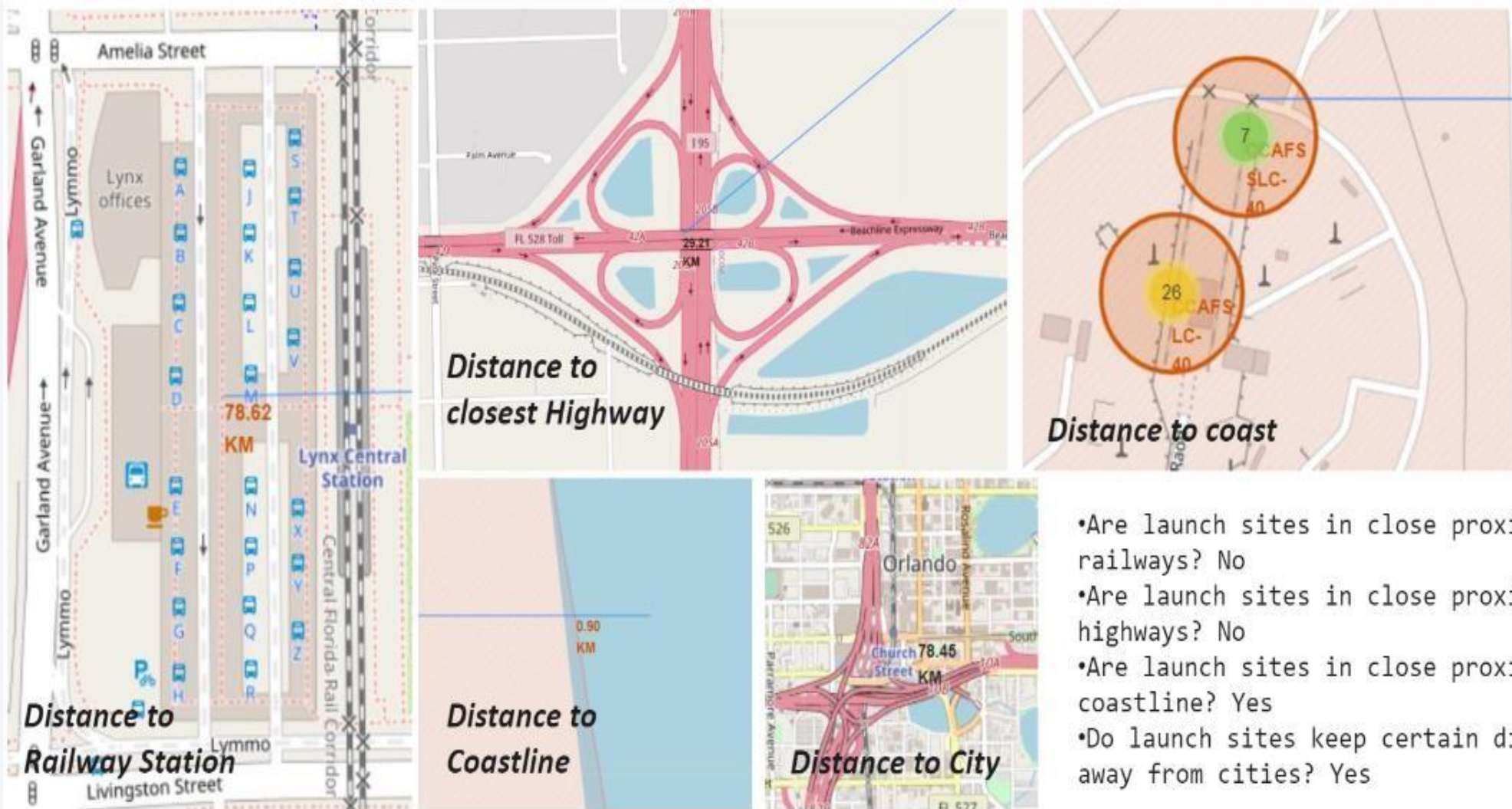


# Markers showing launch sites with color labels





# 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



The background of the slide is a close-up, artistic photograph of a printed circuit board (PCB). The board is dark, and the intricate circuit traces are highlighted in a vibrant, glowing red. Numerous small, circular components, likely solder joints or micro-components, are visible along the traces, some of which also appear to be glowing. The overall effect is one of high-tech complexity and digital energy.

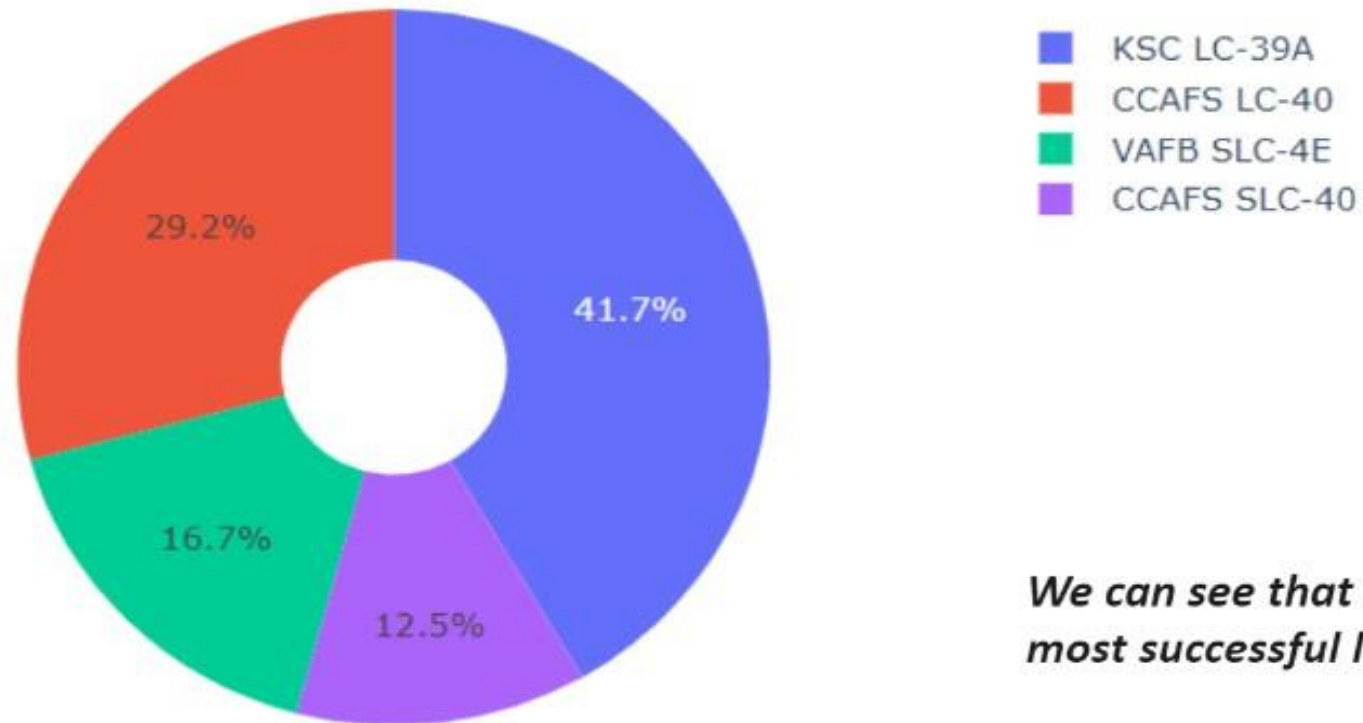
Section 4

# Build a Dashboard with Plotly Dash

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

---

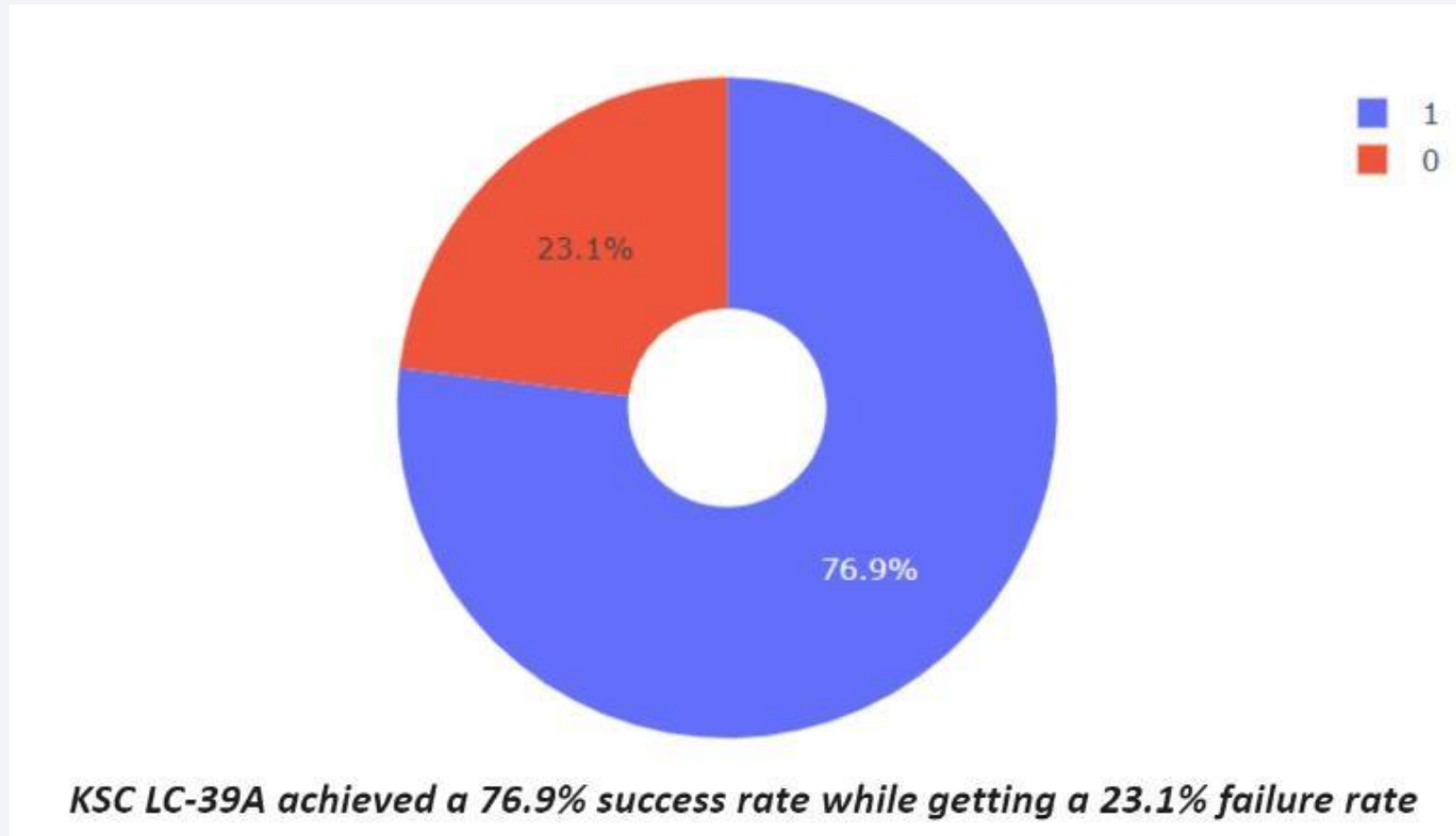
Total Success Launches By all sites



***We can see that KSC LC-39A had the most successful launches from all the sites***

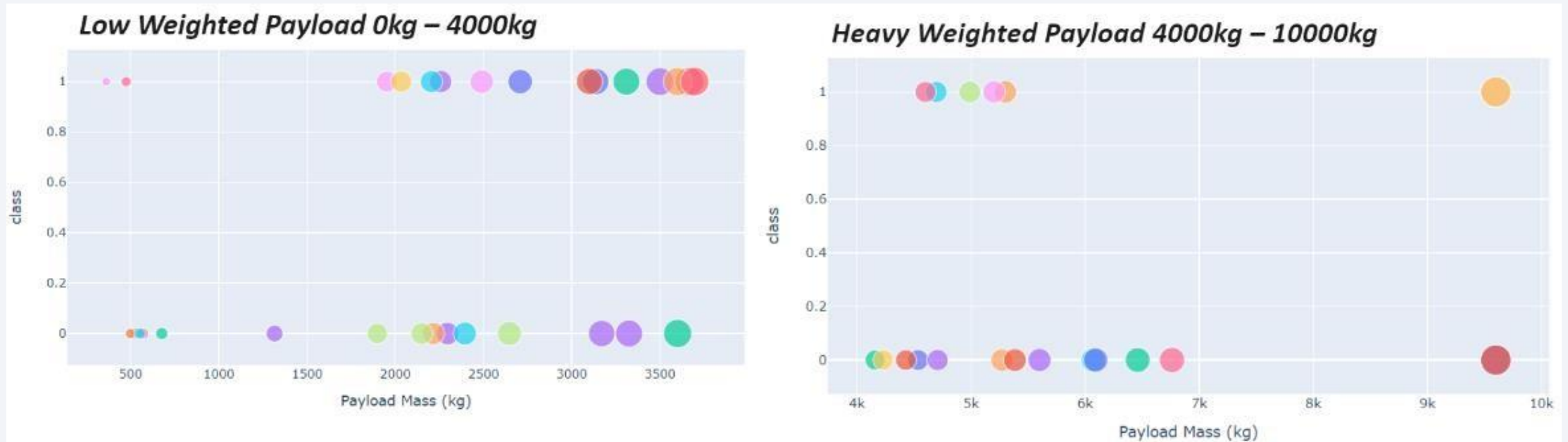
## Pie chart showing the Launch site with the highest launch success ratio

---



A scatter plot illustrating the relationship between Payload and Launch Outcome for all sites, featuring a range slider to select different payload values

---



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



Section 5

# Predictive Analysis (Classification)

# 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

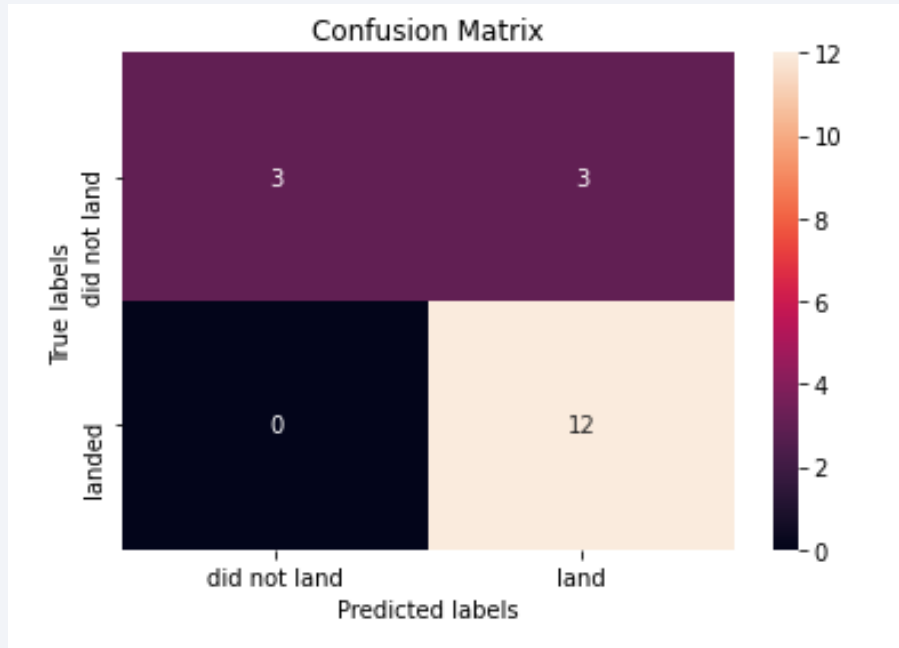
Best params is : {'criterion': 'gini', 'max\_depth': 6, 'max\_features': 'auto', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'splitter': 'random'}



# Confusion Matrix

---

The confusion matrix for the decision tree classifier indicates that the model is capable of differentiating between the various classes. However, a significant issue is the occurrence of false positives, where unsuccessful landings are incorrectly classified as successful landings by the classifier.



# Conclusions

---

- We can conclude that:
  - A higher number of flights at a launch site correlates with a greater success rate.
  - The launch success rate began to rise from 2013 and continued to increase until 2020.
  - The orbits ES-L1, GEO, HEO, SSO, and VLEO achieved the highest success rates.
  - KSC LC-39A recorded the most successful launches among all sites.
  - The decision tree classifier is the most effective machine learning algorithm for this specific task.

Thank you!

