Winning Space Race with Data Science

Tafadzwa Muchedzi

27th July 2023

Outline

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

Executive Summary

Summary of methodologies

- - Data was acquired using API calls.
- - Web scraping was utilized to collect additional data.
- - Data was processed and cleaned through data wrangling.
- - Exploratory data analysis (EDA) was performed using SQL queries.
- - EDA was further conducted using data visualization techniques.
- - Interactive visual analytics were executed using Folium.
- - Machine learning prediction models were built for analysis.

Summary of all results

- Exploratory Data Analysis outcomes
- Interactive analytics in screenshots
- Predictive Analytics findings

Introduction

- Project background and context

SpaceX offers Falcon 9 rocket launches at a significantly lower cost of 62 million dollars on their website compared to other providers, who charge upwards of 165 million dollars per launch. The main reason for this cost advantage is SpaceX's ability to reuse the first stage of their rockets. By determining the likelihood of a successful first stage landing, we can accurately estimate the overall launch cost. This information becomes valuable when competing companies bid for rocket launch contracts against SpaceX. The project's objective is to develop a machine learning pipeline that predicts whether the first stage will successfully land or not. This predictive model will aid in cost estimation and decision-making in the rocket launch industry.

- Problems you want to find answer
- What factors determine if the rocket will land successfully?
- Identify the correlation and interactions between different features that contribute to the success rate of a successful landing.
- Determine the key operating conditions required to ensure a successful landing program.
- Analyse how specific combinations of features influence the likelihood of a successful landing.
- Investigate the dependencies between various factors and their impact on the landing success rate.
- Pinpoint the critical conditions that need to be met to maximize the chances of a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data was gathered through a combination of SpaceX API and web scraping from Wikipedia.
- Data was cleaned and organized through data wrangling techniques.
- Categorical features were converted into numerical format using one-hot encoding.
- Exploratory data analysis (EDA) was conducted using visualization techniques and SQL.
- Interactive visual analytics were performed using Folium and Plotly Dash.
- Classification models were built, tuned, and evaluated for predictive analysis.
- The process involved selecting appropriate classification algorithms, tuning hyperparameters, and evaluating model performance using relevant metrics.

Data Collection

- The data was collected using various methods

- Data collection involved a GET request to the SpaceX API.

- The JSON response content was decoded and converted into a pandas dataframe using .json_normalize().

- Data cleaning included addressing missing values by imputing where needed.

- Web scraping was utilized with BeautifulSoup to extract Falcon 9 launch records from Wikipedia.

- The primary objective was to parse the HTML table and convert it into a pandas dataframe for subsequent analysis.

Data Collection – SpaceX API

- We utilized a GET request to the SpaceX API to gather data, performed data cleaning, and conducted basic data wrangling and formatting.
- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/Collecting%20the%20data.ipynb

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 used BeautifulSoup for web scraping to extract Falcon 9 launch records. After parsing the table, we transformed the data into a pandas dataframe.
- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/Web%20scraping%20Falcon%209.ipynb

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

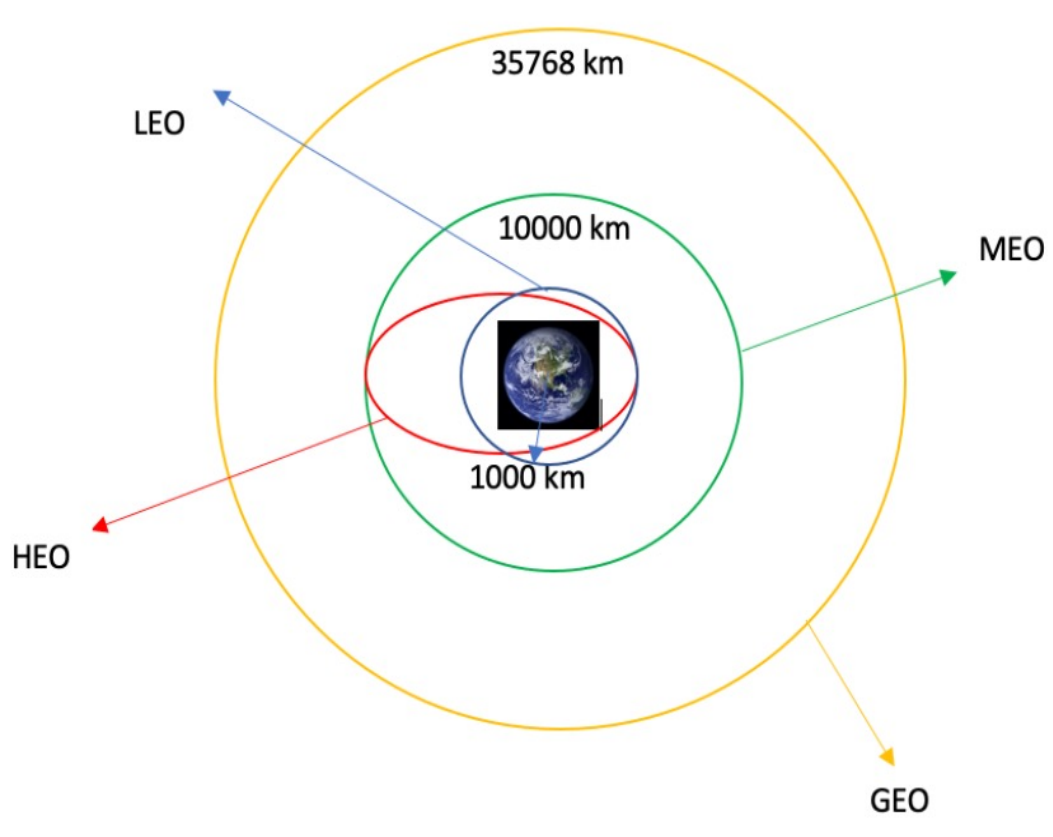
```
In [10]: column_names = []

         # 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

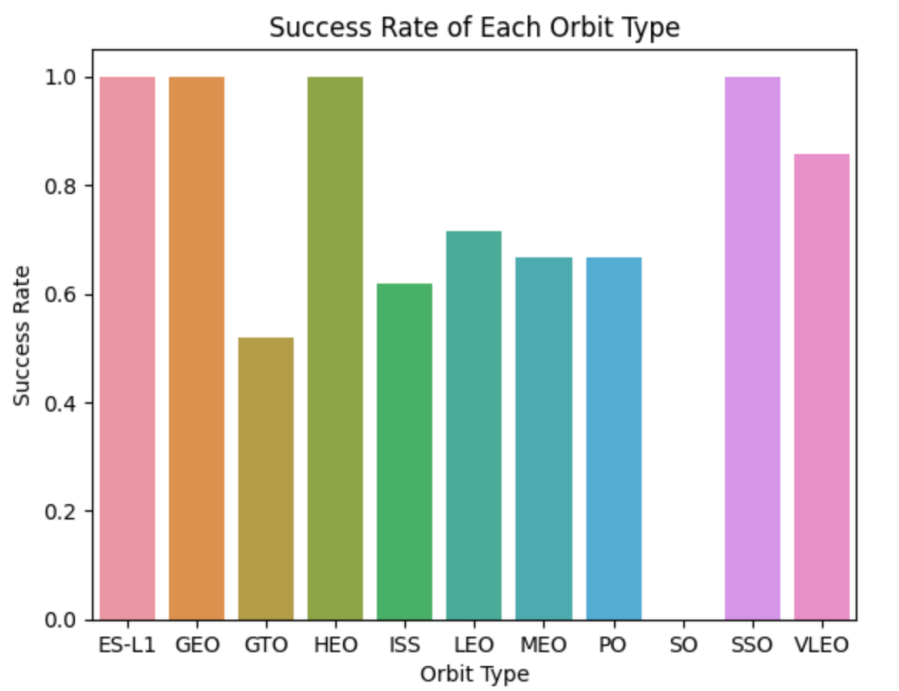
Data Wrangling

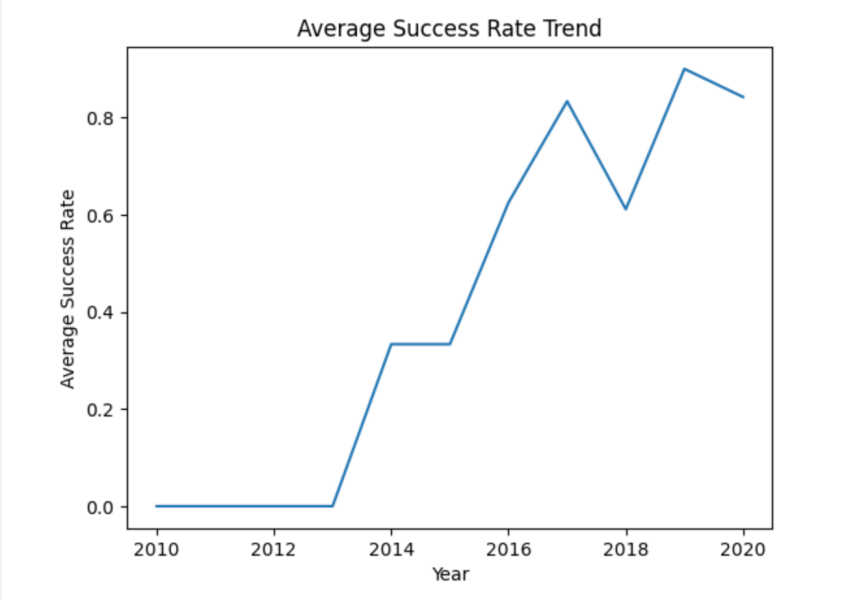


- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/%20Data%20wrangling.ipynb

EDA with Data Visualization

- We performed data exploration through visualizations to analyze the relationships between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, and the annual trend of launch success.





- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- - The SpaceX dataset was seamlessly loaded into a PostgreSQL database directly from the Jupyter notebook.
- - EDA with SQL was conducted to gain valuable insights from the data.
- - Queries were written to identify unique launch site names in the space missions.
- - The total payload mass carried by NASA (CRS) launched boosters was calculated.
- - The average payload mass carried by booster version F9 v1.1 was determined.
- - The total count of successful and failure mission outcomes was obtained.
- - The drone ship landing failures, along with their respective booster versions and launch site names, were extracted through queries.
- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/SQL%20NOTEBOOK%20sqllite.ipynb

Build an Interactive Map with Folium

- We plotted all launch sites on the folium map and used map objects like markers, circles, and lines to indicate the success or failure of launches for each site.
- The launch outcomes were categorized as class 0 for failure and class 1 for success.
- By employing color-labeled marker clusters, we identified launch sites with relatively high success rates.
- Calculating distances between launch sites and their surroundings, we addressed questions such as proximity to railways, highways, and coastlines.
- We also explored whether launch sites maintain a certain distance from cities in the analysis.

Build a Dashboard with Plotly Dash

- - An interactive dashboard was constructed using Plotly Dash.
- - Pie charts were utilized to visualize the total number of launches from specific sites.
- - Scatter graphs were plotted to illustrate the relationship between Outcome and Payload Mass (Kg) for various booster versions.
- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/Folium2ipynb

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is: https://github.com/chibhanz/SPACEX/blob/main/Machine%20Learning%20Prediction.ipynb

Results

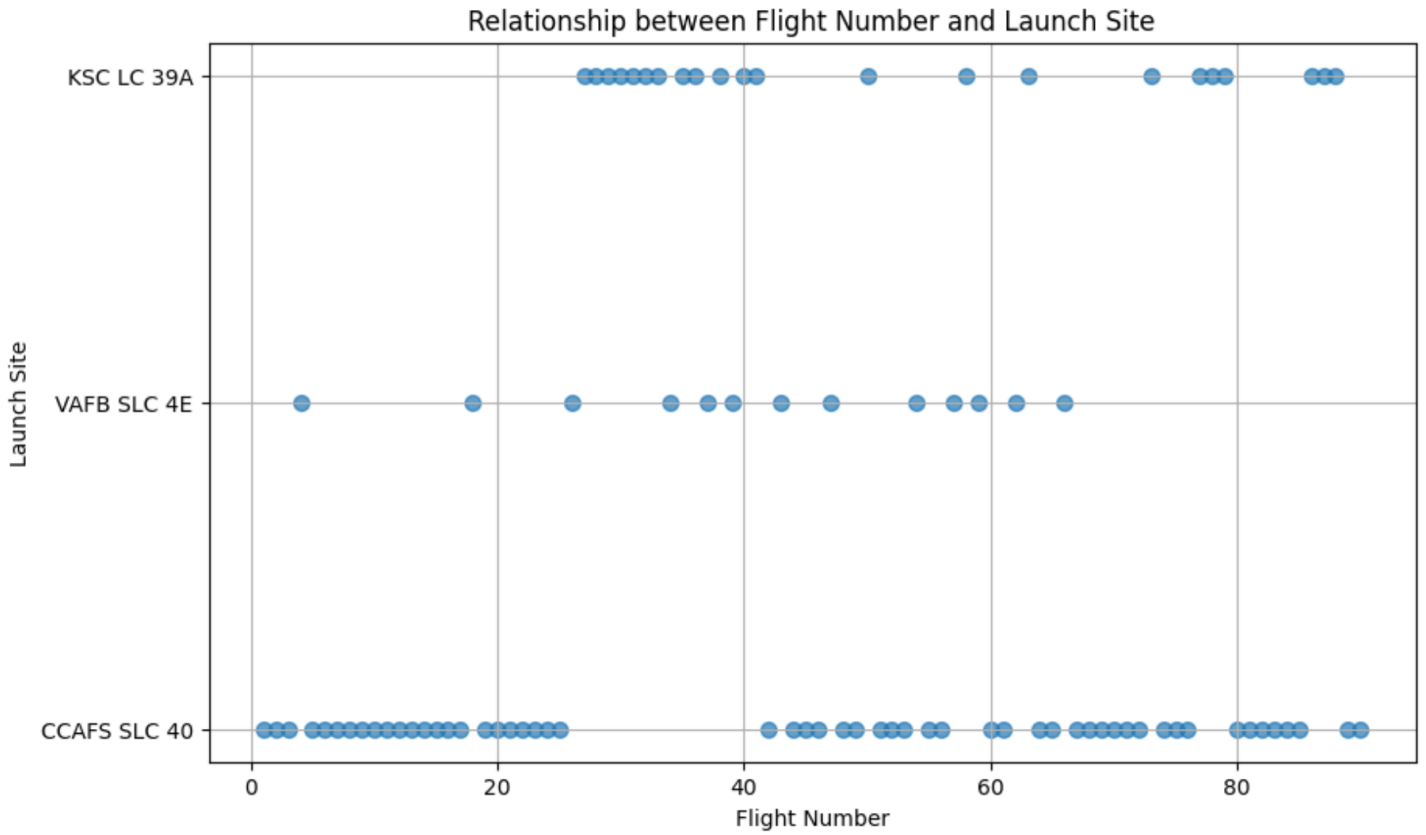
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

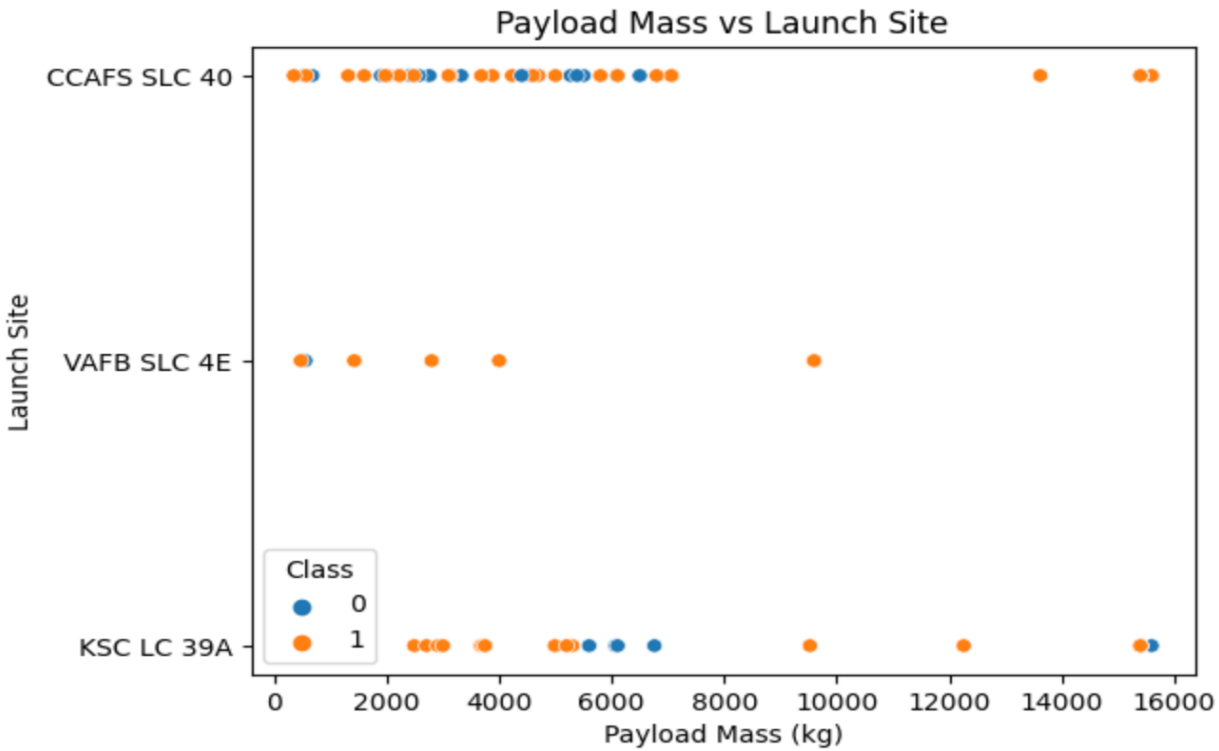
Insights drawn from EDA

Flight Number vs. Launch Site

- Based on the plot, we observed that higher flight amounts at a launch site correspond to a higher success rate at that site.



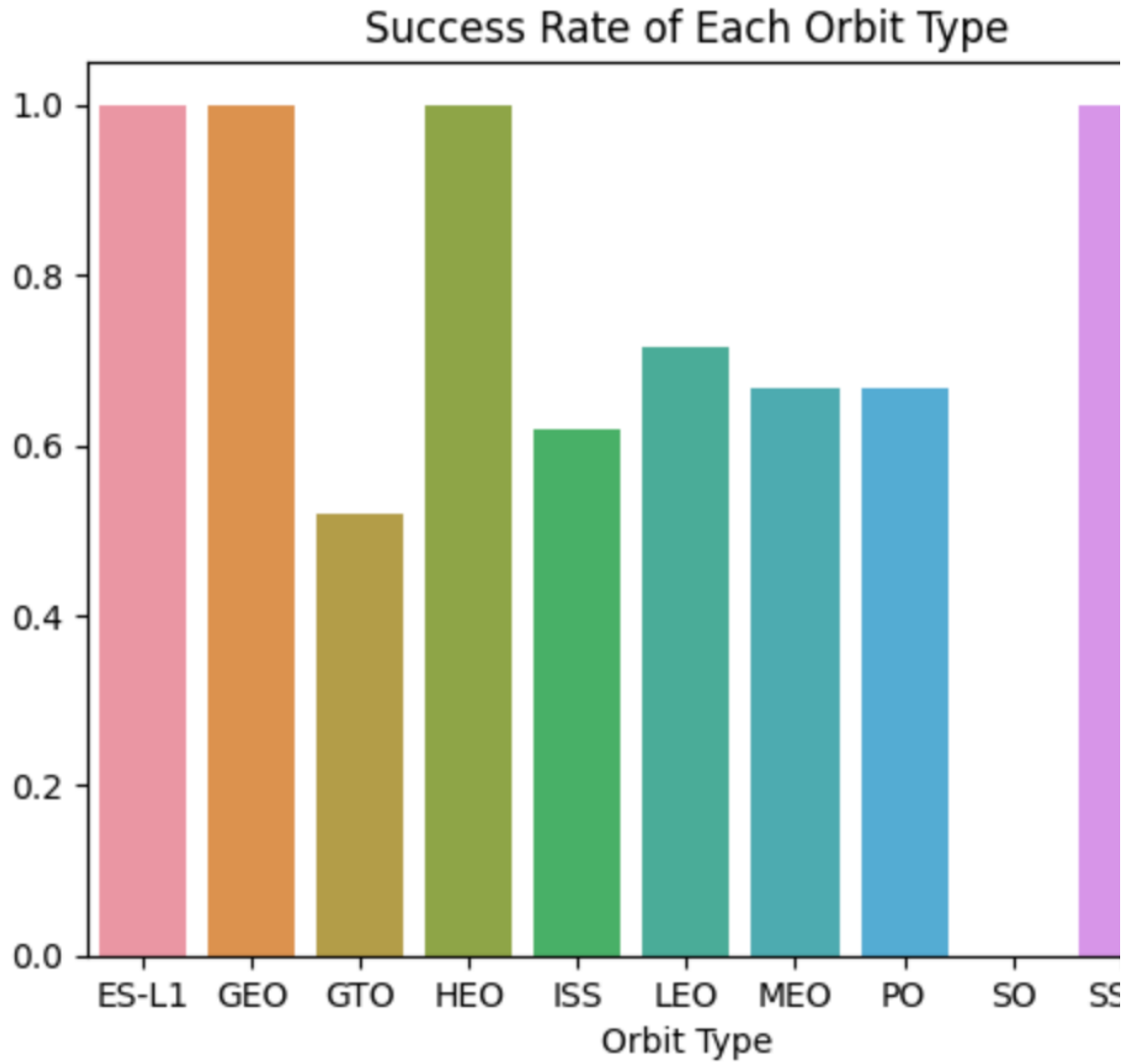
Payload vs. Launch Site



- For Site CCAFS SLC 40 Payload mass and success are positively correlated

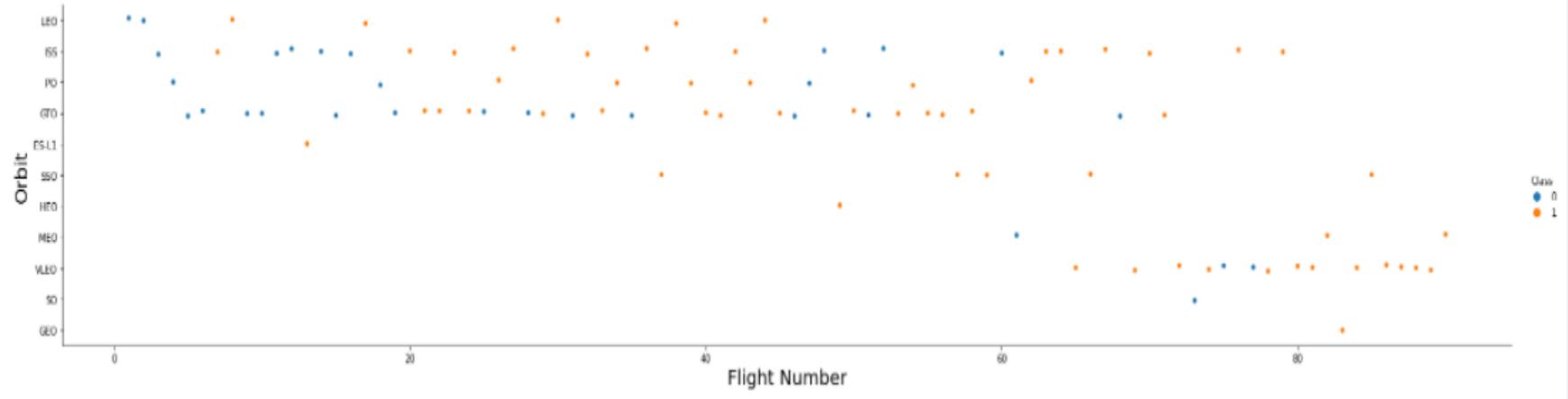
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



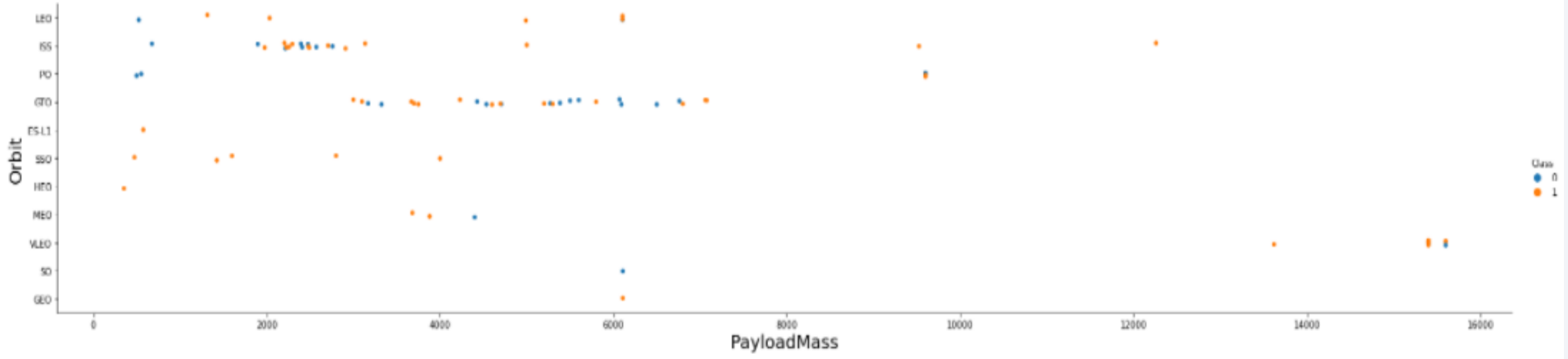
Flight Number vs. Orbit Type

- The plot illustrates the relationship between Flight Number and Orbit type. We notice that in the LEO orbit, the success rate is influenced by the number of flights, while in the GTO orbit, there is no discernible relationship between flight number and the orbit's success.



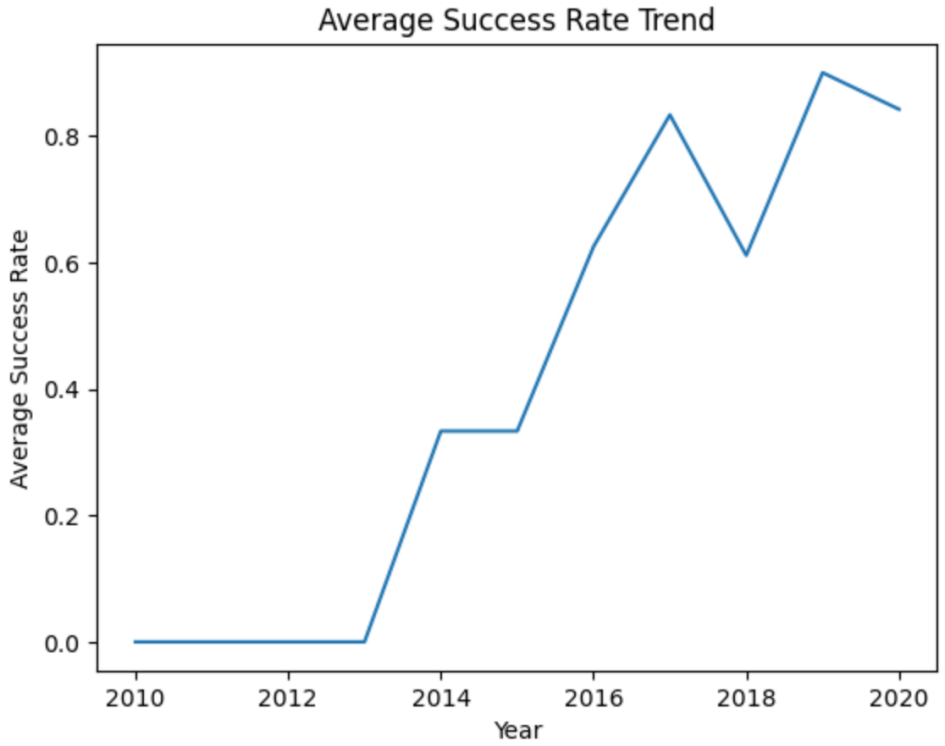
Payload vs. Orbit Type

- It can be observed that for PO, LEO, and ISS orbits, there is a higher frequency of successful landings when heavy payloads are involved.



Launch Success Yearly Trend

- Based on the plot, it is evident that the success rate has been consistently increasing from 2013 to 2020.



All Launch Site Names

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

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'

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 kgs using the query as shown 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 calculated the average payload mass carried by booster version F9 v1.1 as 2928.4kgs

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

- 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 4000kgs but less than 6000kgs

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

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

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)
```

Out[16]:

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

	failureoutcome
0	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.

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 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

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

- 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.

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

Section 4

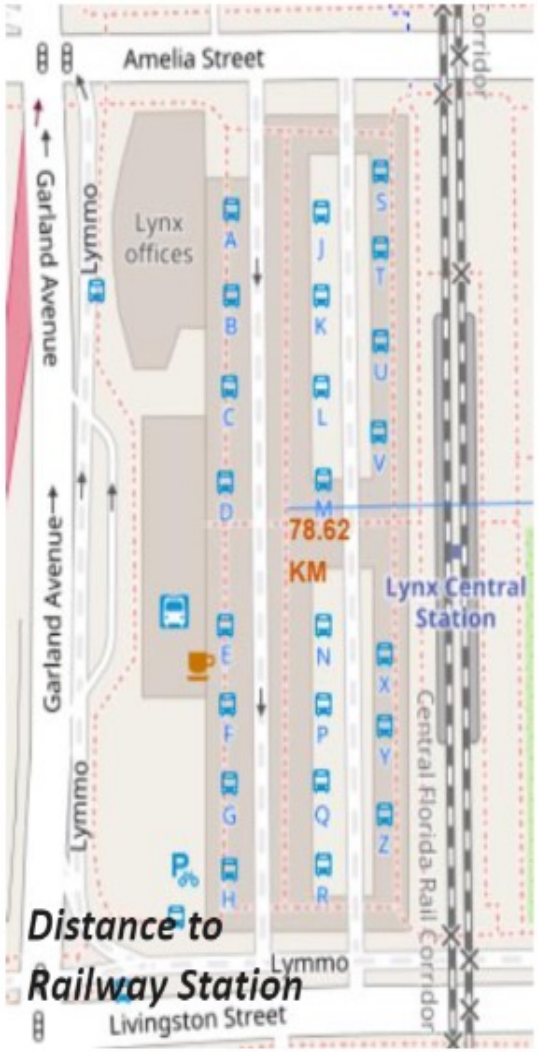
Launch Sites Proximities Analysis

All launch sites global map markers

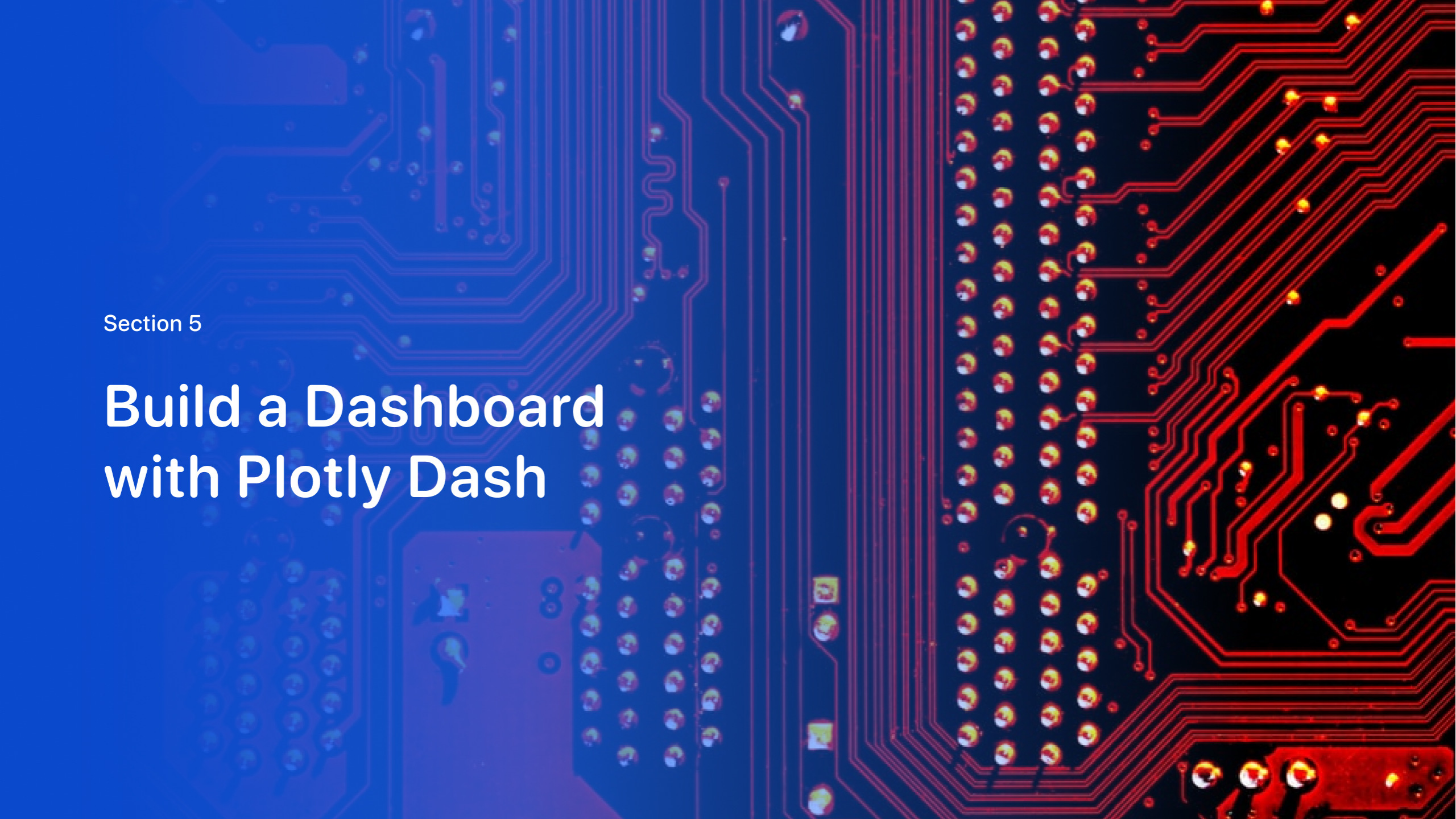
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

Launch Site distance to landmarks



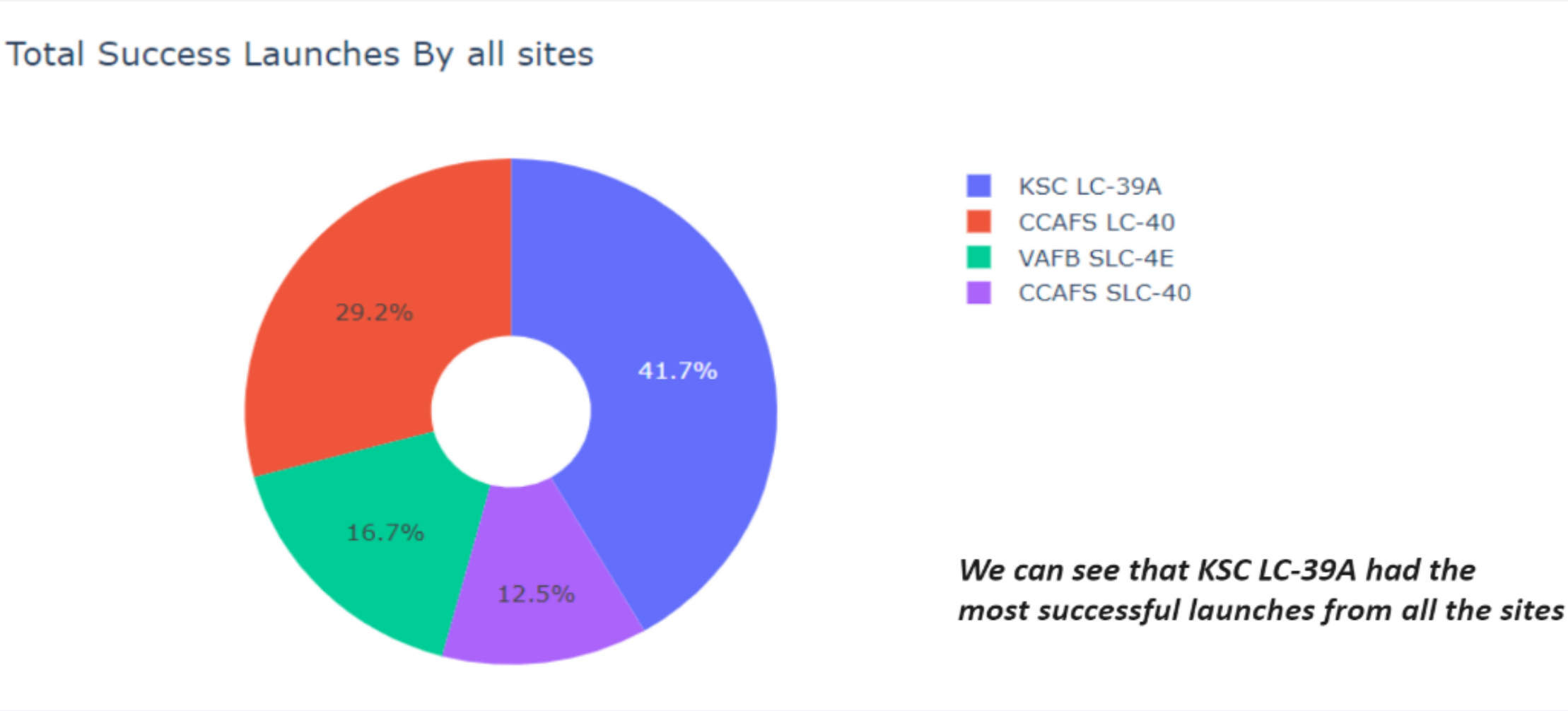
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 5

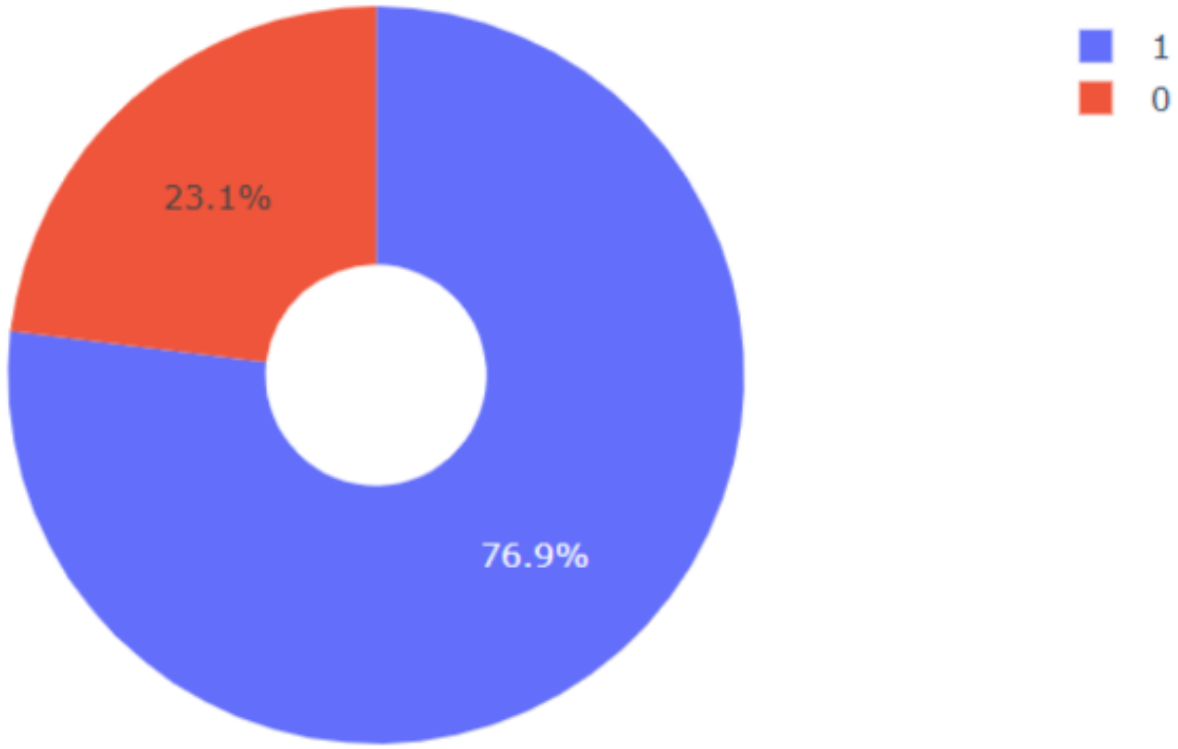
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



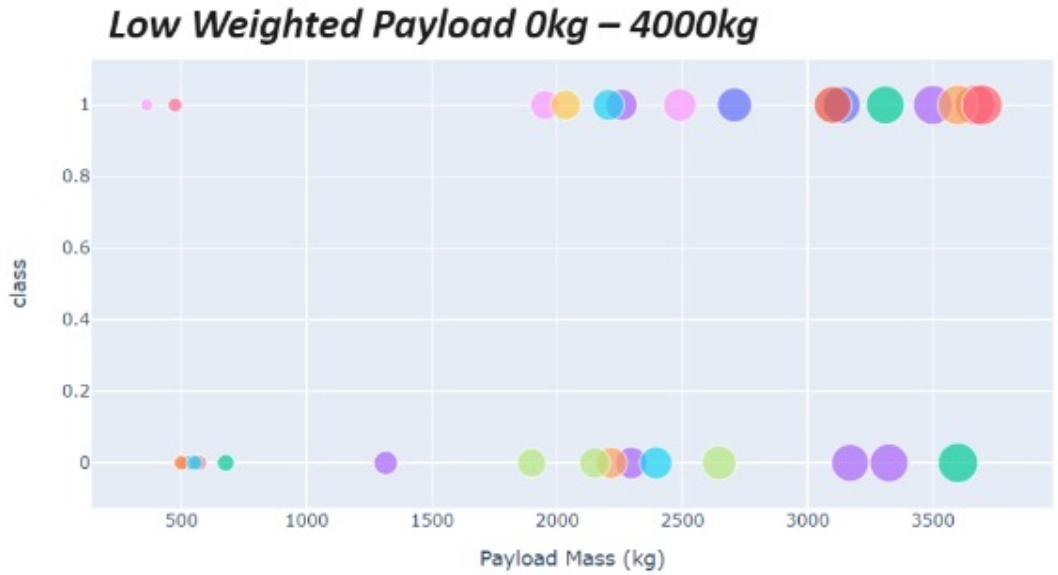
We can see that KSC LC-39A had the most successful launches from all the sites

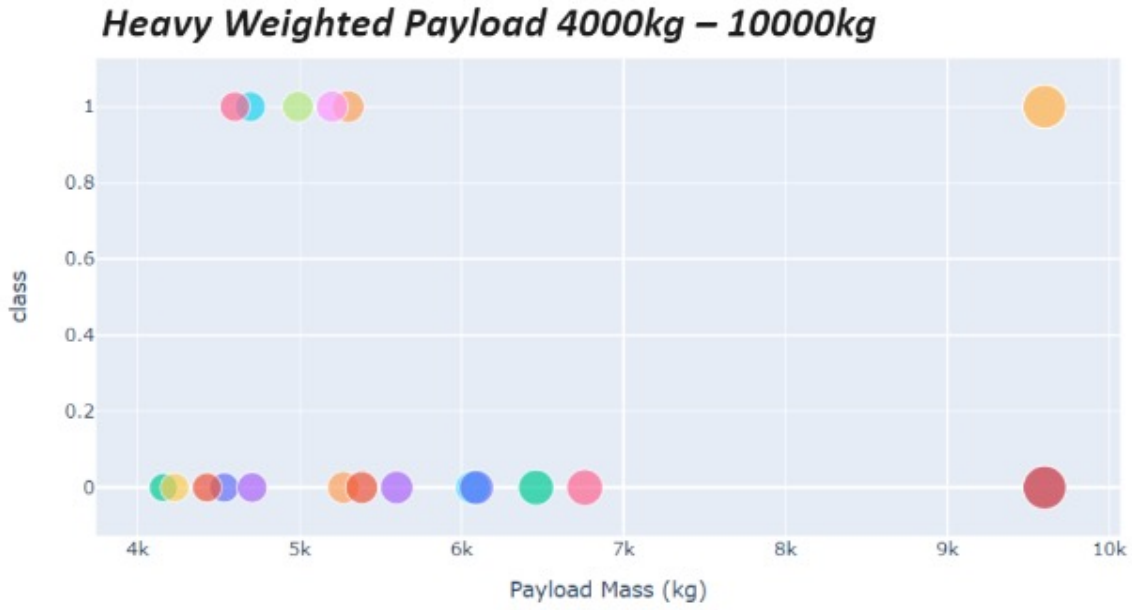
Pie chart showing the Launch site with the highest launch success ratio



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

Section 6

Predictive Analysis (Classification)

Classification Accuracy

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

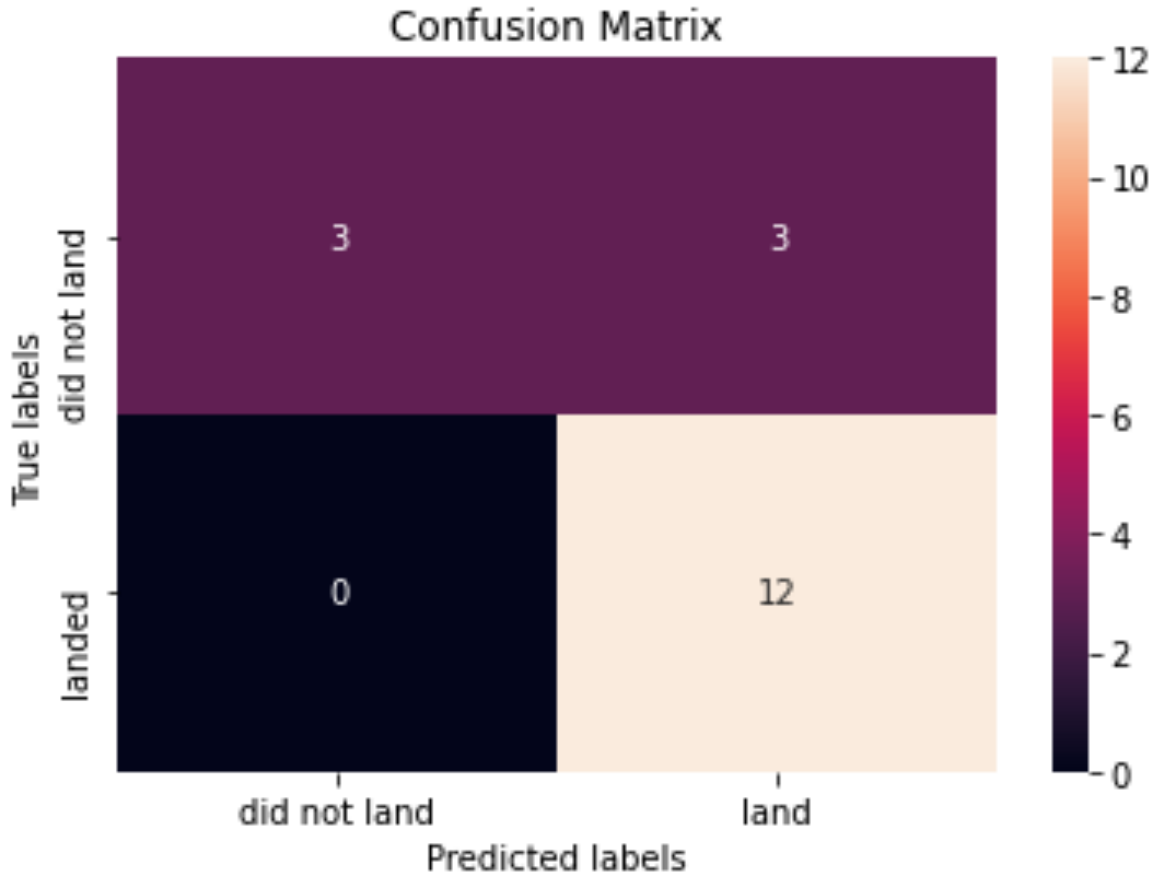
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 indicates that the classifier is capable of distinguishing between the different classes. However, the primary issue lies in the false positives, where unsuccessful landings are misclassified as successful landings by the classifier.



Conclusions

We can conclude that:

- There is a positive correlation between the flight amount at a launch site and the success rate at that site.
- The launch success rate has been on the rise since 2013 and continued to increase until 2020.
- Orbits ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.
- KSC LC-39A had the highest number of successful launches among all sites.
- The Decision tree classifier is considered the most suitable machine learning algorithm for this particular task.

Thank you!