IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jose Ambrosio
September 26, 2025

Outline

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

Executive Summary

Overview of Methods Applied

- Data Acquisition via API
- Data Gathering through Web Scraping
- Data Cleaning and Preparation
- Exploratory Analysis using SQL
- Exploratory Analysis through Visualizations
- Interactive Geospatial Visualization with Folium
- Predictive Modeling with Machine Learning

Summary of Findings

- Results from Exploratory Data Analysis
- Screenshots of Interactive Dashboards/Analytics
- Outcomes from Predictive Modeling

Introduction

Project Background and Context

SpaceX lists Falcon 9 rocket launches on its website at a price of about $62 million, compared to more than $165 million charged by other providers. The cost advantage largely comes from the ability to reuse the rocket's first stage. Predicting whether the first stage will successfully land is therefore key to estimating launch costs. Such insights could also help competing companies when preparing bids against SpaceX. The main objective of this project is to develop a machine learning pipeline capable of predicting the likelihood of a first-stage landing.

Key Research Questions

- Which factors influence the probability of a rocket landing successfully?
- How do different features interact to impact landing success rates?
- What operating conditions are necessary to maximize the chances of a successful landing?

Section 1

Methodology

Methodology

Data Collection Approach:

- Data was obtained from the SpaceX API and supplemented with web scraping from Wikipedia.

Data Preparation:

- Data cleaning and wrangling were performed.
- Categorical variables were transformed using one-hot encoding.

Exploratory Analysis:

- Exploratory Data Analysis (EDA) was conducted using SQL queries and visualizations.
- Interactive visualizations were created with Folium and Plotly Dash.

Predictive Modeling:

- Classification models were developed to predict outcomes.
- Steps included building, fine-tuning, and evaluating model performance.

Data Collection

Data Collection and Preparation

- Data was retrieved from the SpaceX API using GET requests.
- The response was decoded as JSON using the .json() method and converted into a pandas DataFrame via .json_normalize().
- Data cleaning was performed, including checking for and handling any missing values.
- Additional launch records for Falcon 9 were obtained from Wikipedia using web scraping with BeautifulSoup.
- The HTML tables were extracted, parsed, and converted into pandas DataFrames to support further analysis.

Data Collection – SpaceX API

I collected data from the SpaceX API using a GET request, followed by data cleaning, basic wrangling, and formatting to prepare it for analysis.

Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2001/jupyter-labs-spacex-data-collection-api.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

I used web scraping with BeautifulSoup to extract Falcon 9 launch records, then parsed the HTML table and converted it into a pandas DataFrame.

Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2002/jupyter-labs-webscraping.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
```

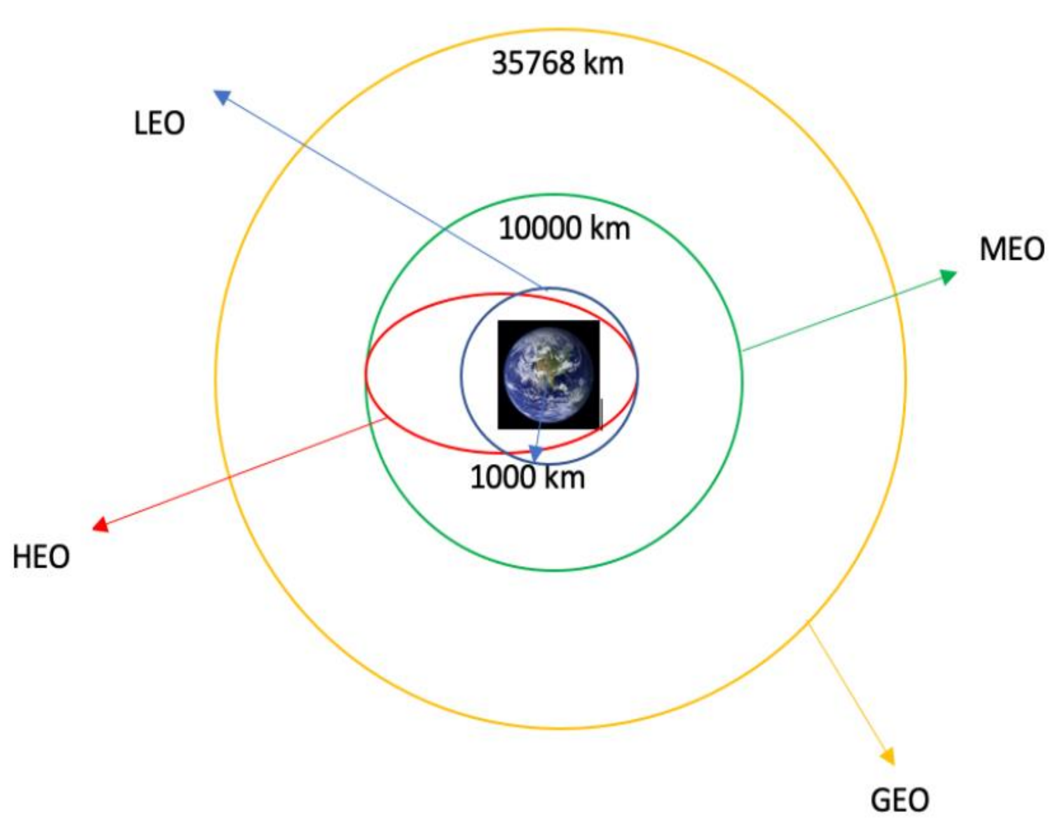
4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

I conducted exploratory data analysis and identified the training labels. I calculated the number of launches per site, as well as the frequency of each orbit type. Additionally, I generated the landing outcome labels from the outcome column and exported the results to a CSV file.

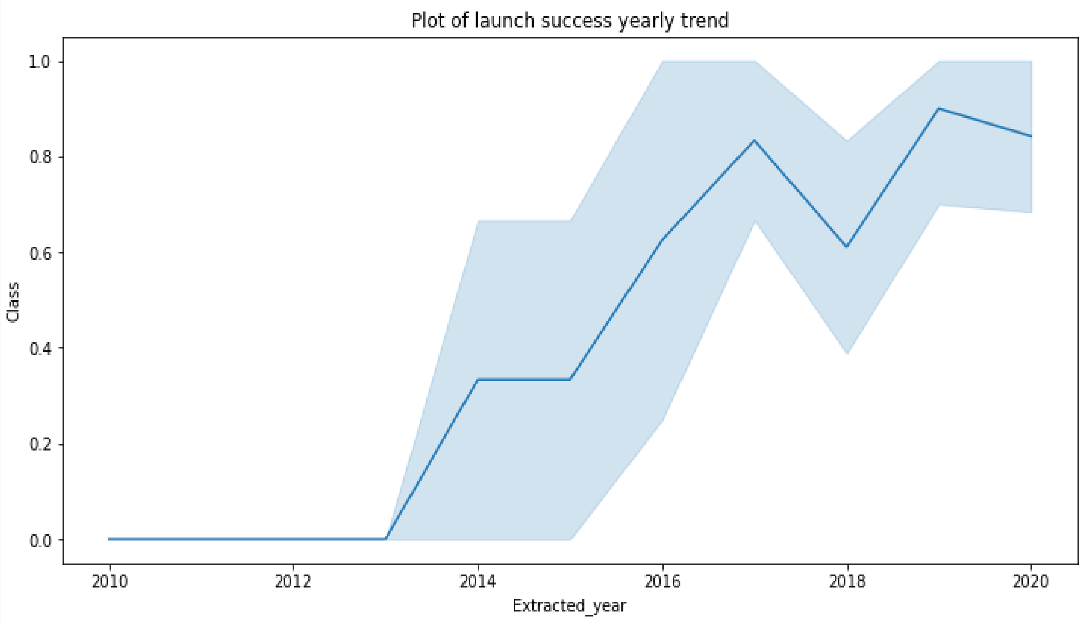
Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2002/jupyter-labs-webscraping.ipynb



EDA with Data Visualization

I analyzed the data by creating visualizations to examine relationships between flight number and launch site, payload and launch site, the success rate for each orbit type, flight number and orbit type, and trends in launch success over the years.



Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2005/edadataviz.ipynb

EDA with SQL

I imported the SpaceX dataset directly into a PostgreSQL database from within Jupyter Notebook. Using SQL, I performed exploratory data analysis to gain insights, including:

- Identifying unique launch site names.
- Calculating the total payload mass for boosters launched by NASA (CRS).
- Determining the average payload mass for booster version F9 v1.1.
- Counting the total number of successful and failed missions.
- Analyzing failed landings on drone ships, including booster versions and launch site names.

Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2004/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

I visualized all launch sites on a Folium map, adding markers, circles, and lines to indicate the success or failure of each launch. Launch outcomes were encoded as 0 for failure and 1 for success. Using color-coded marker clusters, I identified launch sites with relatively high success rates. I also calculated distances between each launch site and nearby features to answer questions such as:

- **Are launch sites located near railways, highways, or coastlines?**
- **Do launch sites maintain a certain distance from cities?**

Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2006/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

I created an interactive dashboard using Plotly Dash. The dashboard includes pie charts displaying the total launches per site and scatter plots illustrating the relationship between launch outcomes and payload mass (kg) for different booster versions.

Dashboard Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2007/spacex-dash-app.py

Predictive Analysis (Classification)

I loaded the data with NumPy and pandas, performed data transformation, and split it into training and testing sets. I developed several machine learning models, tuning hyperparameters with GridSearchCV. Model performance was evaluated using accuracy, and improvements were made through feature engineering and algorithm optimization, ultimately identifying the best-performing classification model.

Notebook Link:

https://github.com/jose-ambrosioo/IBM-Data-Science-Professional-Certificate/blob/main/10%20-%20Applied%20Data%20Science%20Capstone/Labs/Lab%2008/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

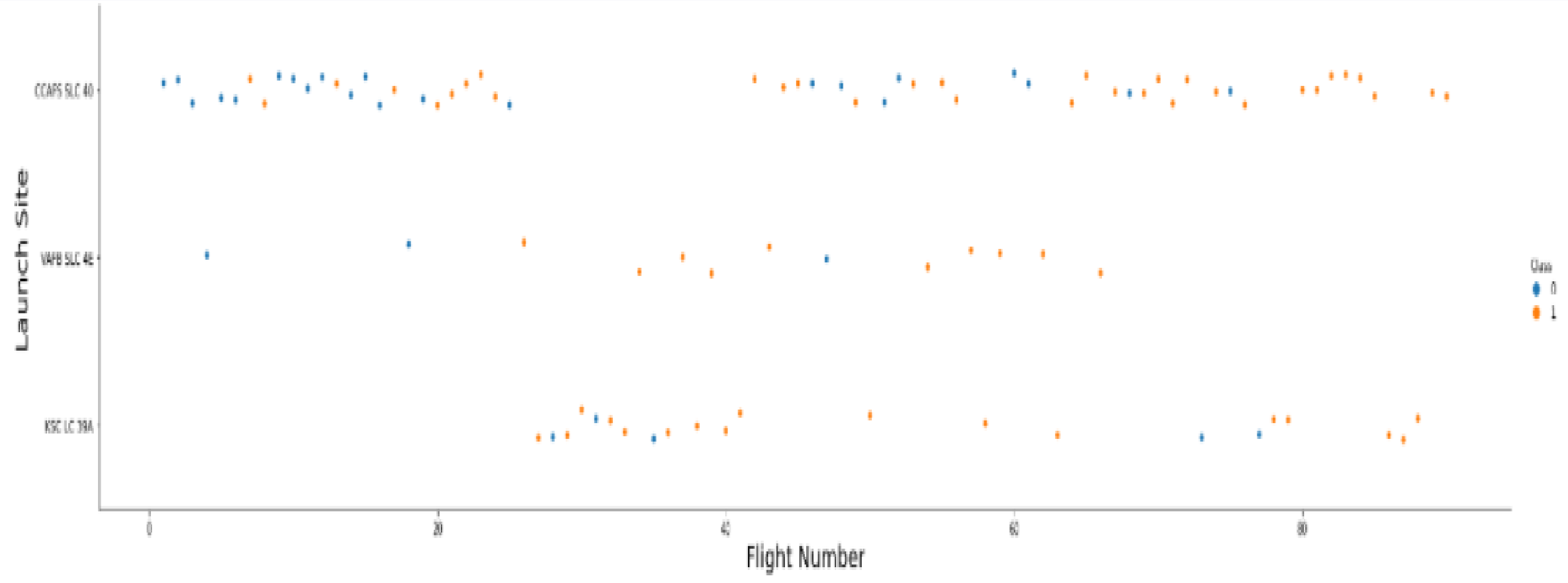
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

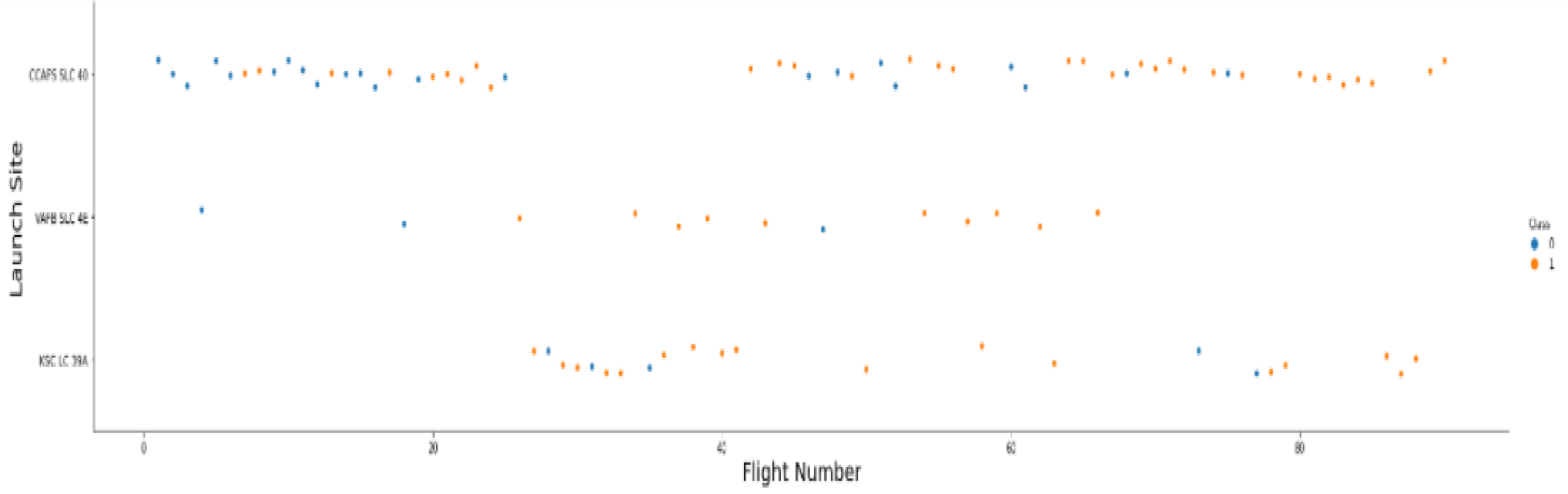
Flight Number vs. Launch Site

The plot shows that launch sites with a higher number of flights tend to have higher success rates.



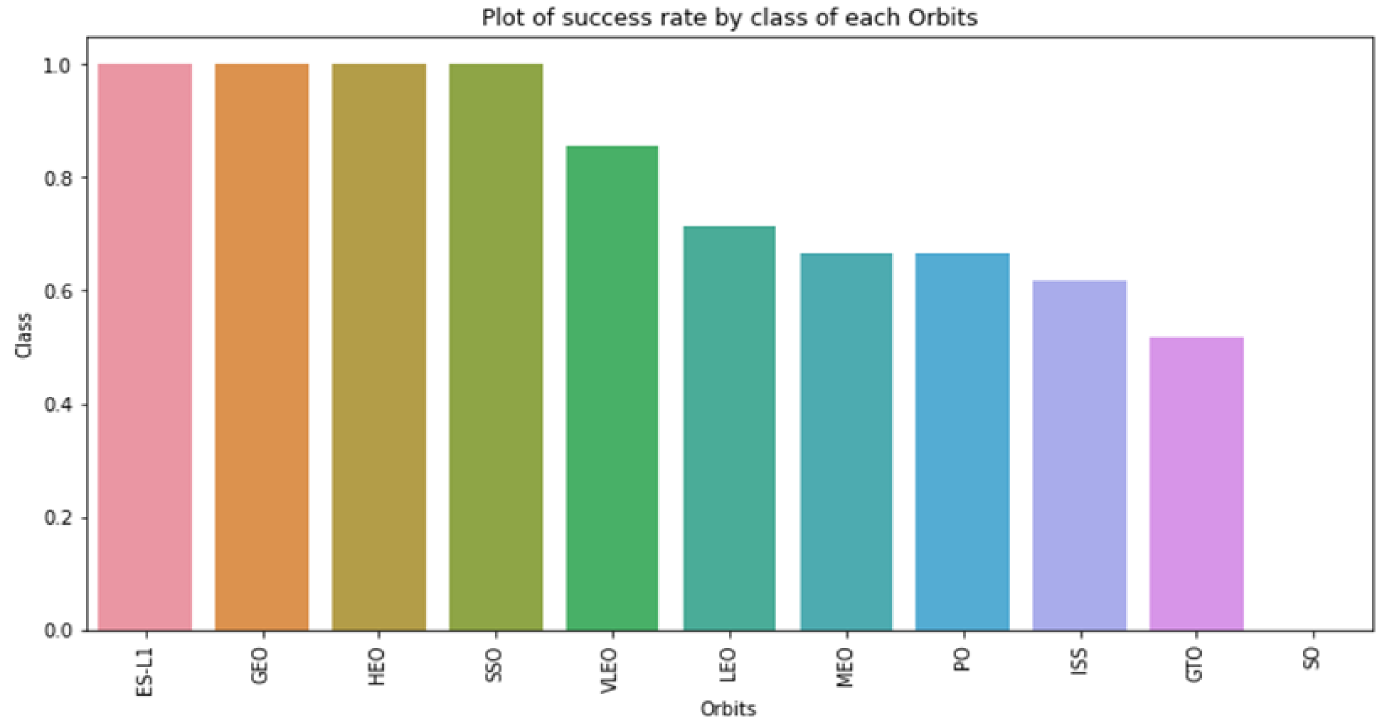
Payload vs. Launch Site

The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



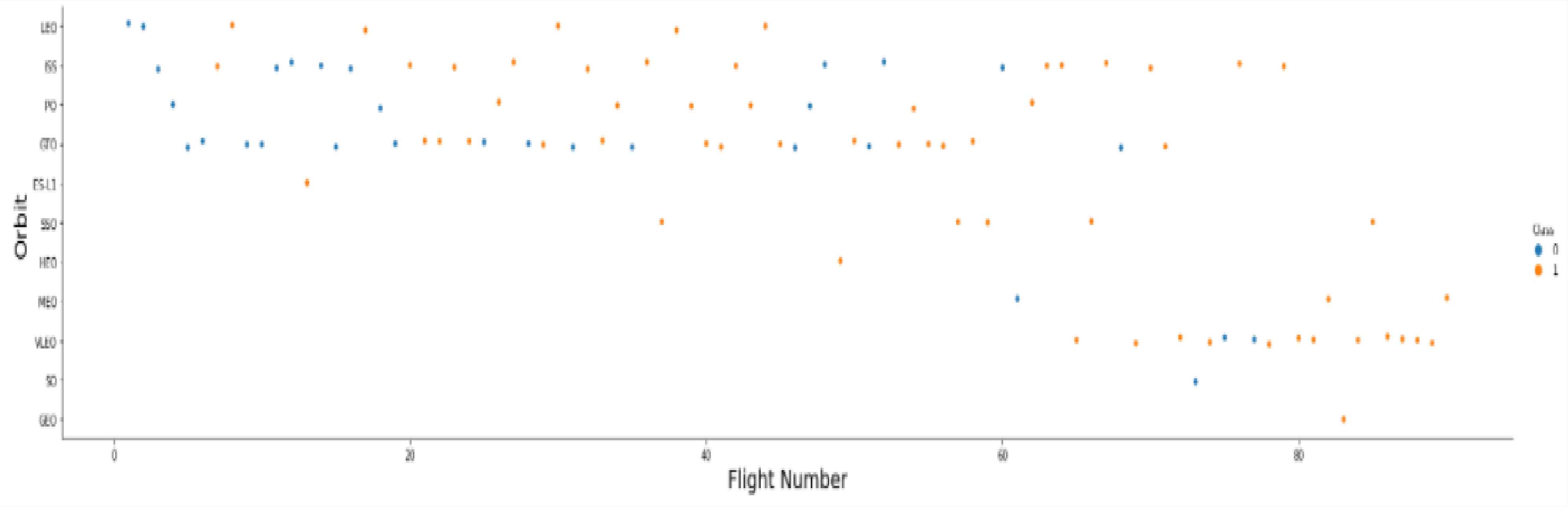
Success Rate vs. Orbit Type

The plot indicates that the orbits ES-L1, GEO, HEO, SSO, and VLEO exhibit the highest success rates.



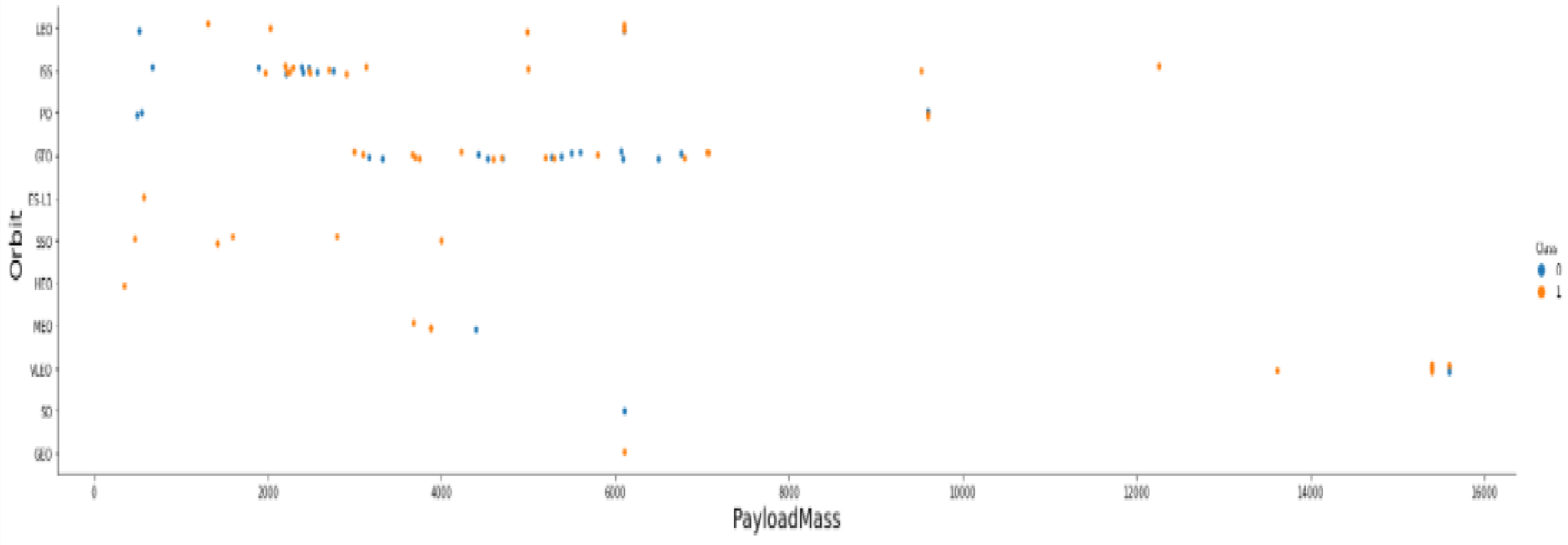
Flight Number vs. Orbit Type

The plot of Flight Number versus Orbit Type shows that for LEO orbits, success appears to increase with the number of flights, while for GTO orbits, no clear relationship is observed between flight number and success.



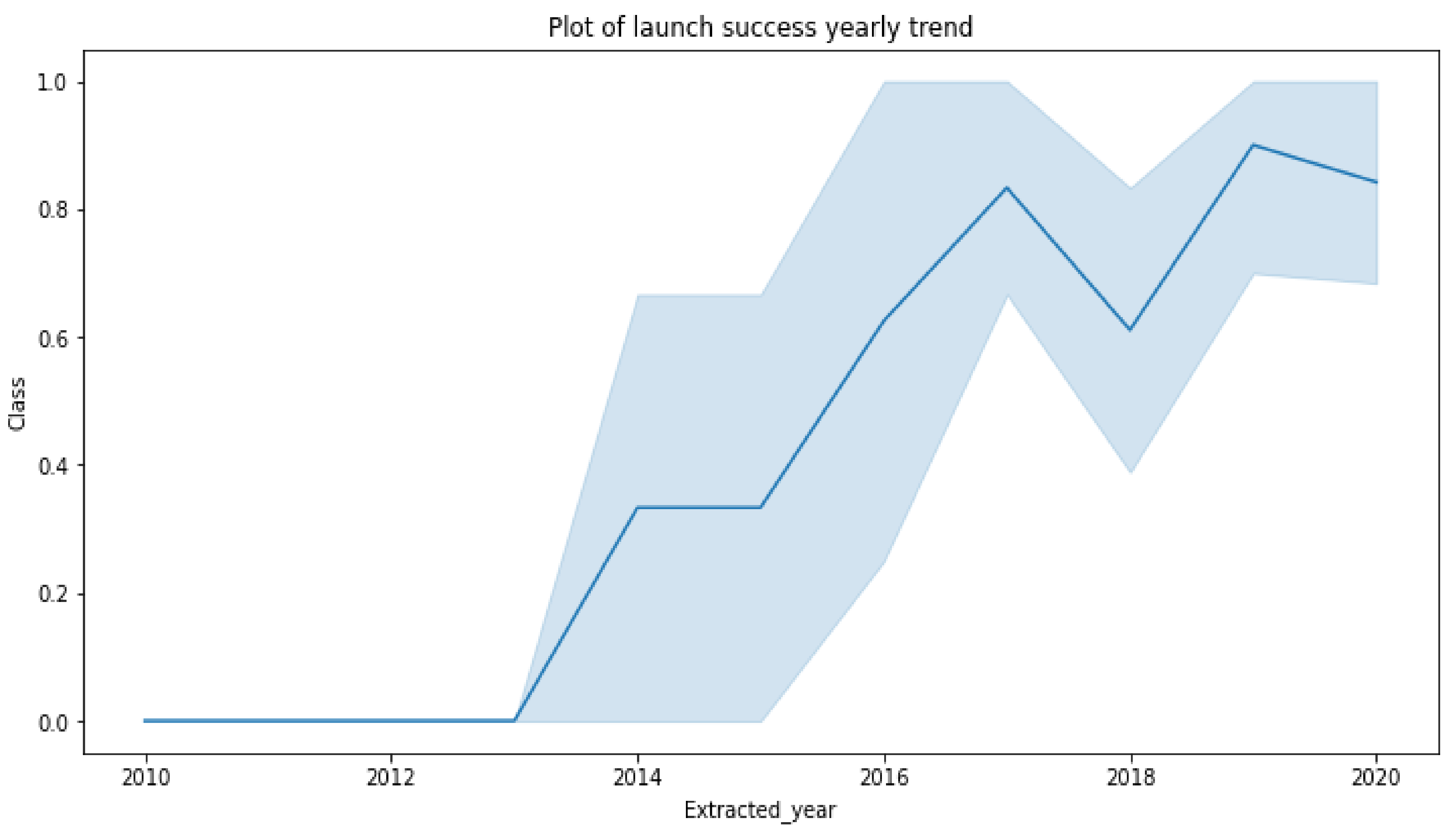
Payload vs. Orbit Type

It can be observed that heavier payloads tend to have more successful landings in PO, LEO, and ISS orbits.



Launch Success Yearly Trend

The plot shows that the success rate has steadily increased from 2013 through 2020.



All Launch Site Names

I used the SQL keyword **DISTINCT** to display only the unique launch sites from the SpaceX dataset.

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'

I executed the query below to retrieve five records for launch sites starting 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

Using the query below, I calculated that boosters launched by NASA carried a total payload of 45,596.

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

I determined that the average payload mass for 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

I noted that the first successful landing on the ground pad occurred on 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

I applied a **WHERE** clause to filter for boosters that successfully landed on the drone ship and used an **AND** condition to select those with payload masses between 4,000 and 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

I used wildcard like '%' to filter for **WHERE MissionOutcome** was a success or a failure.

Boosters Carried Maximum Payload

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

I identified the booster that carried the maximum payload using a subquery within the **WHERE** clause along with the **MAX()** function.

2015 Launch Records

I applied a combination of **WHERE**, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landings on the drone ship in 2015, along with their booster versions and launch site names.

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

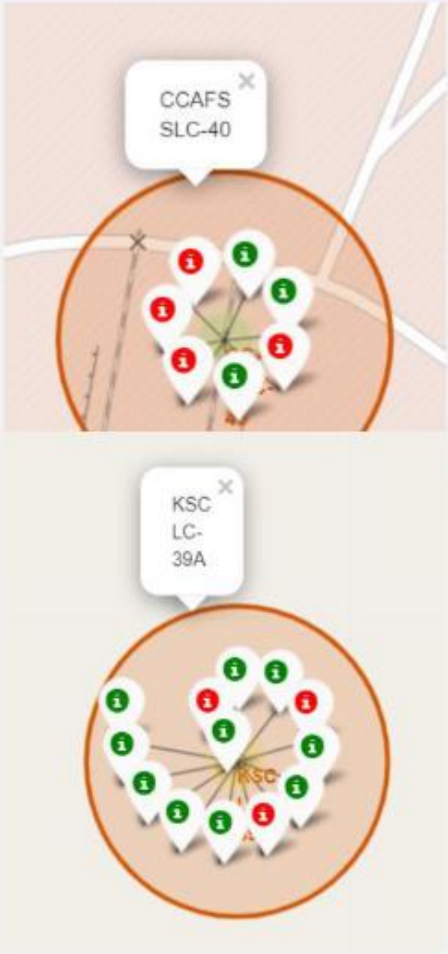
I selected landing outcomes and their counts, using a **WHERE** clause to filter for dates between 2010-03-20 and 2010-06-04. I then grouped the results with **GROUP BY** and sorted them in descending order using **ORDER BY**.

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels



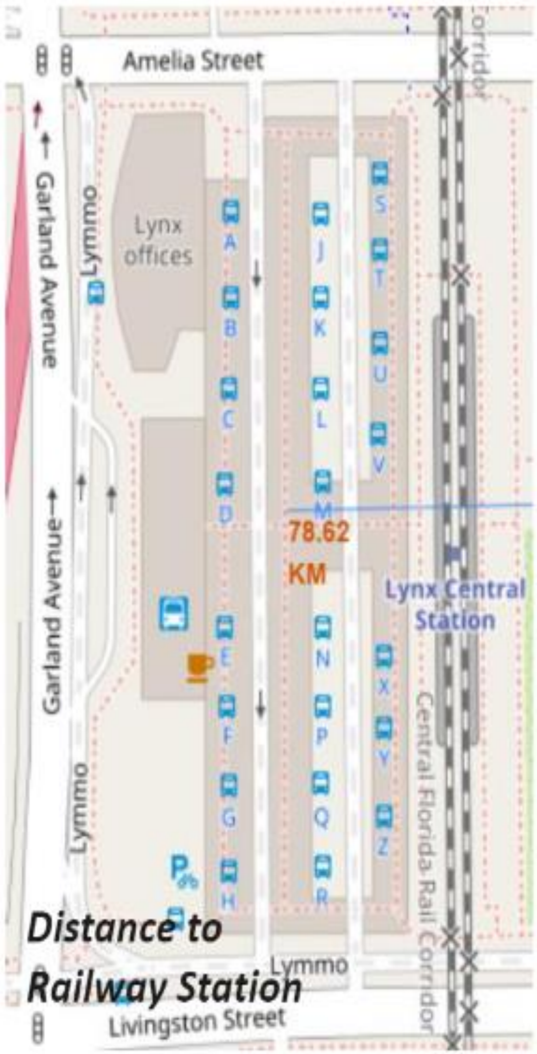
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

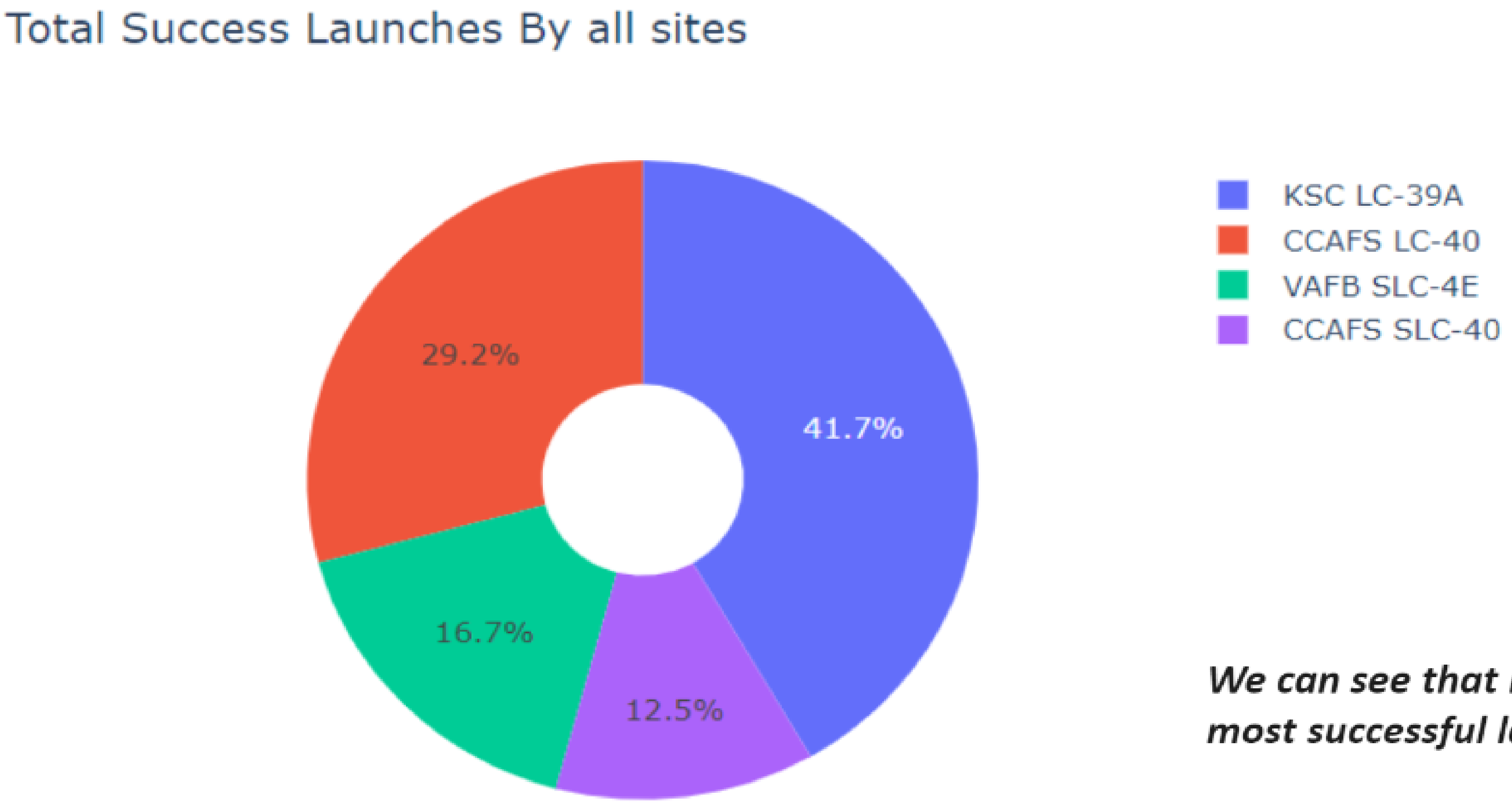


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

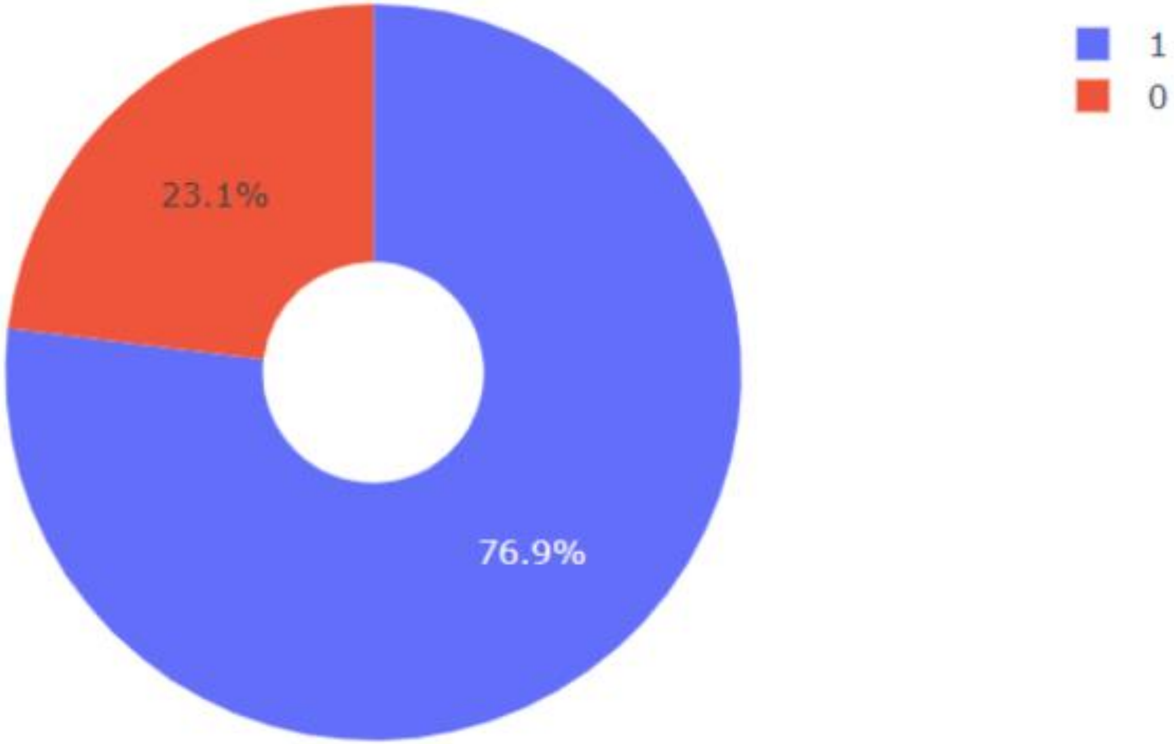
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



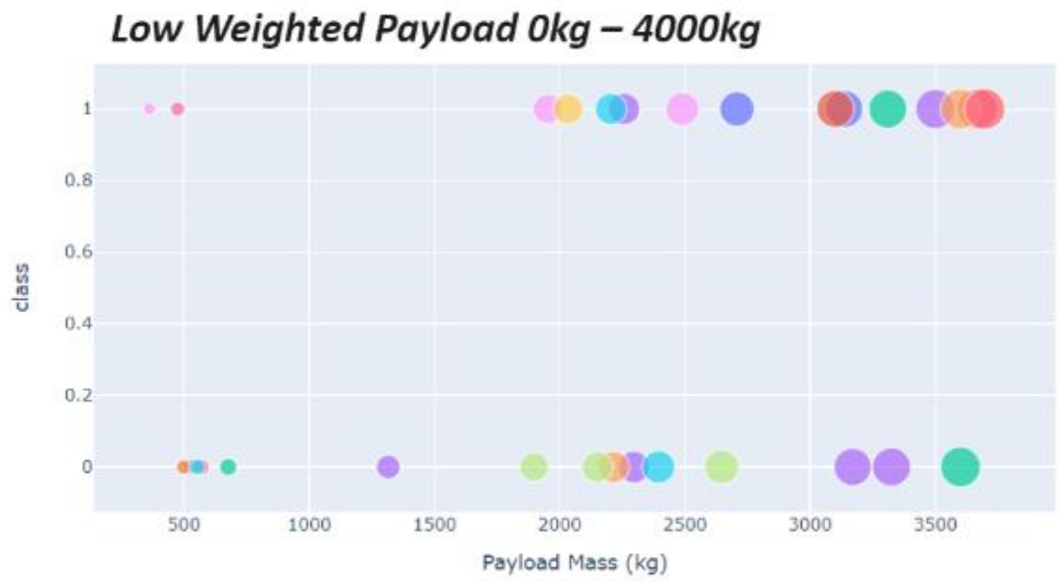
We can see that KSC LC-39A had the most successful launches from all the sites

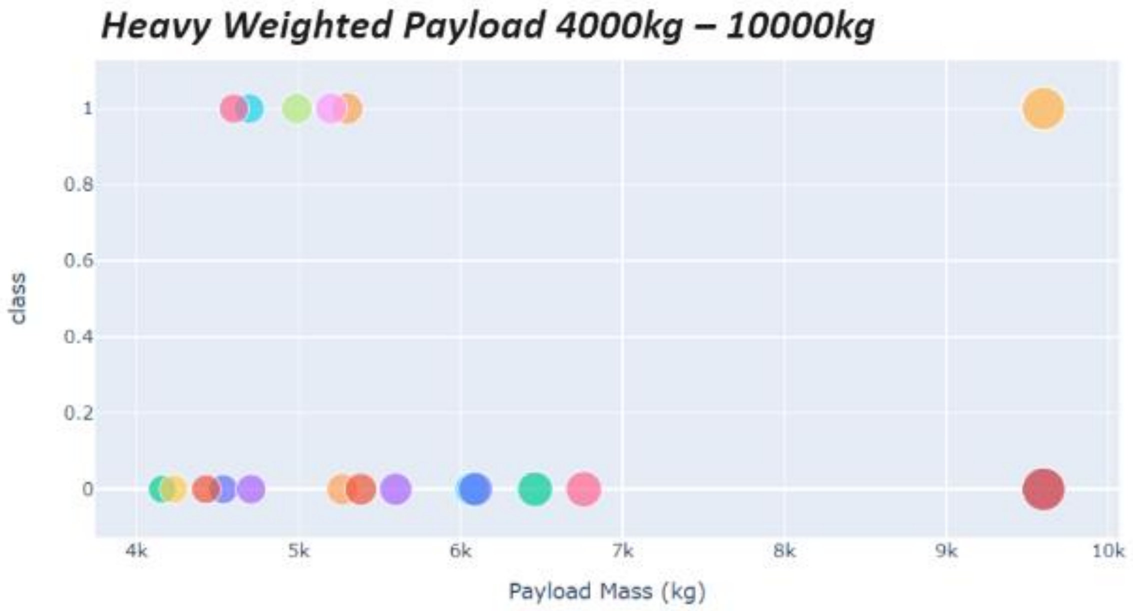
Pie chart showing the Launch site with the highest launch success ratio



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 5

Predictive Analysis (Classification)

Classification Accuracy

The decision tree classifier achieved the highest accuracy among all the models.

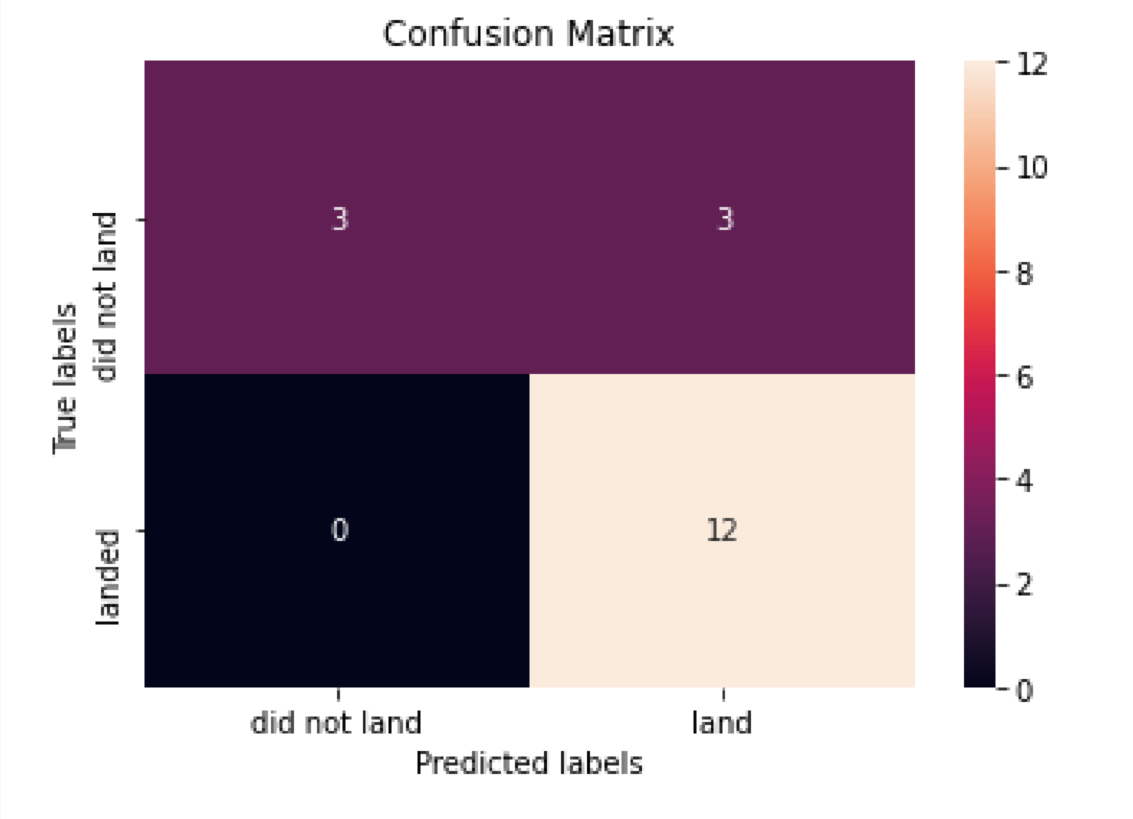
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 it can differentiate between classes. The main issue observed is false positives, where unsuccessful landings are incorrectly predicted as successful.



Conclusions

Analysis of the SpaceX launch data reveals that sites with more flights generally achieve higher success rates, with overall success improving steadily from 2013 to 2020. Orbits such as ES-L1, GEO, HEO, SSO, and VLEO show the highest success, and KSC LC-39A stands out as the most successful launch site. Among the machine learning models tested, the decision tree classifier performed best for predicting landing outcomes.

Thank you!