IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dhanush P

18th August 2024

Outline

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

Executive Summary

- Methodologies Summary:
  - API-based Data Collection
  - Web Scraping for Data Collection
  - Data Cleaning and Transformation
  - SQL-based Exploratory Data Analysis
  - Data Visualization for Exploratory Analysis
  - Interactive Analytics with Folium
  - Predictive Modeling with Machine Learning
- Results Summary:
  - Results of Exploratory Data Analysis
  - Screenshots of Interactive Analytics
  - Outcomes of Predictive Analytics

Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches at a cost of $62 million per launch, significantly lower than the $165 million charged by other providers. This cost advantage is largely due to SpaceX's ability to reuse the first stage of the rocket. Therefore, predicting whether the first stage will successfully land is key to estimating the overall launch cost. This information is valuable for other companies that may wish to compete with SpaceX on launch bids. The project's objective is to develop a machine learning pipeline to predict the likelihood of a successful first-stage landing.

- Problems you want to find answers

  - Key Determinants of Rocket Landing Success: What are the critical factors that influence whether a rocket's first stage will land successfully?
  - Feature Interactions and Success Rate: How do various features interact and contribute to the success rate of a rocket's landing?
  - Operational Conditions for Success: What operating conditions are necessary to ensure a consistent and successful rocket 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 was applied to categorical features
  - Target variable converted to classification numerical variable
- 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 requests to the SpaceX API.

  - The response content was decoded into JSON format using the `.json()` function.

  - The JSON data was converted into a pandas DataFrame with the `.json_normalize()` function.

  - Data cleaning was performed, missing values were checked, and necessary imputations were made.

  - Web scraping was conducted on Wikipedia to gather Falcon 9 launch records using BeautifulSoup.

  - The objective was to extract launch records as HTML tables, parse the data, and convert it into a pandas DataFrame for future analysis.

Data Collection – SpaceX API

- We utilized GET requests to the SpaceX API to gather data, followed by data cleaning, basic data wrangling, and formatting to prepare the dataset for analysis.

- GitHub: https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/Data%20Collection%20API.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

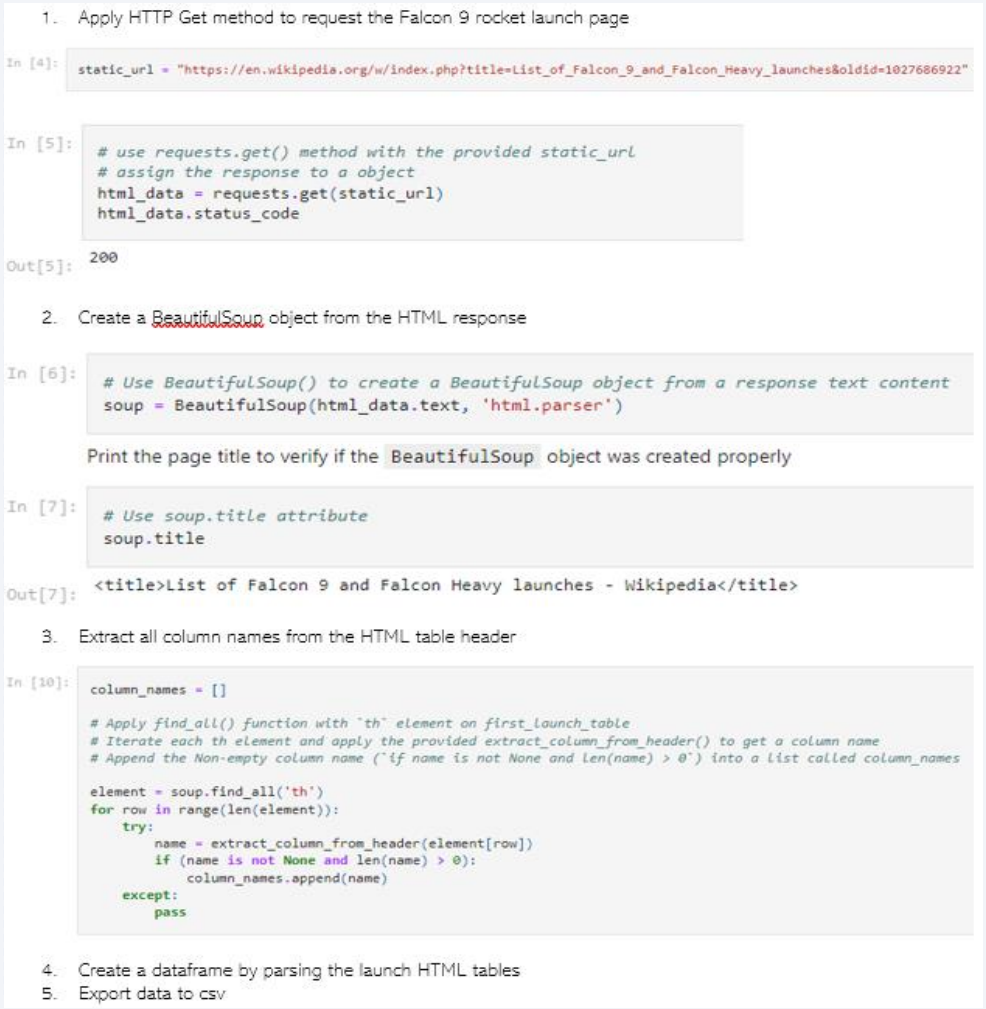
In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

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 employed web scraping techniques, using BeautifulSoup, to extract Falcon 9 launch records from a website.
- The scraped HTML tables were then parsed, and the data was converted into a pandas DataFrame, enabling easier manipulation and future analysis.
- Github: https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/Data%20Collection%20with%20Web%20Scraping.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

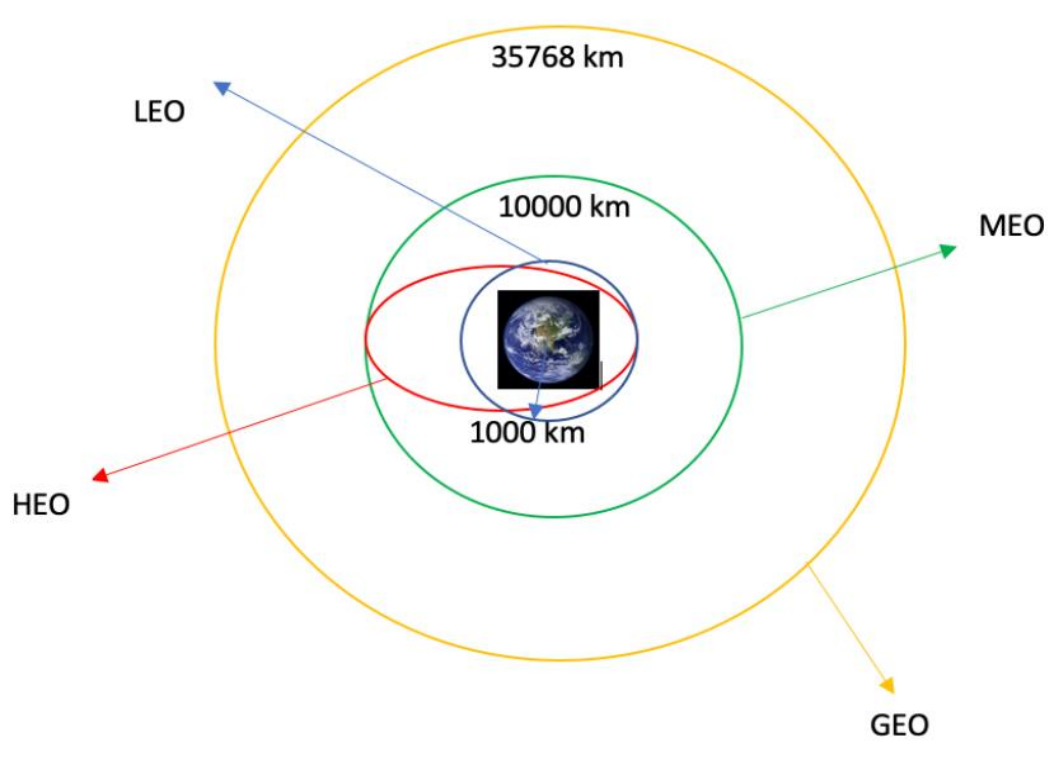
```
In [10]: column_names = []

         # 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



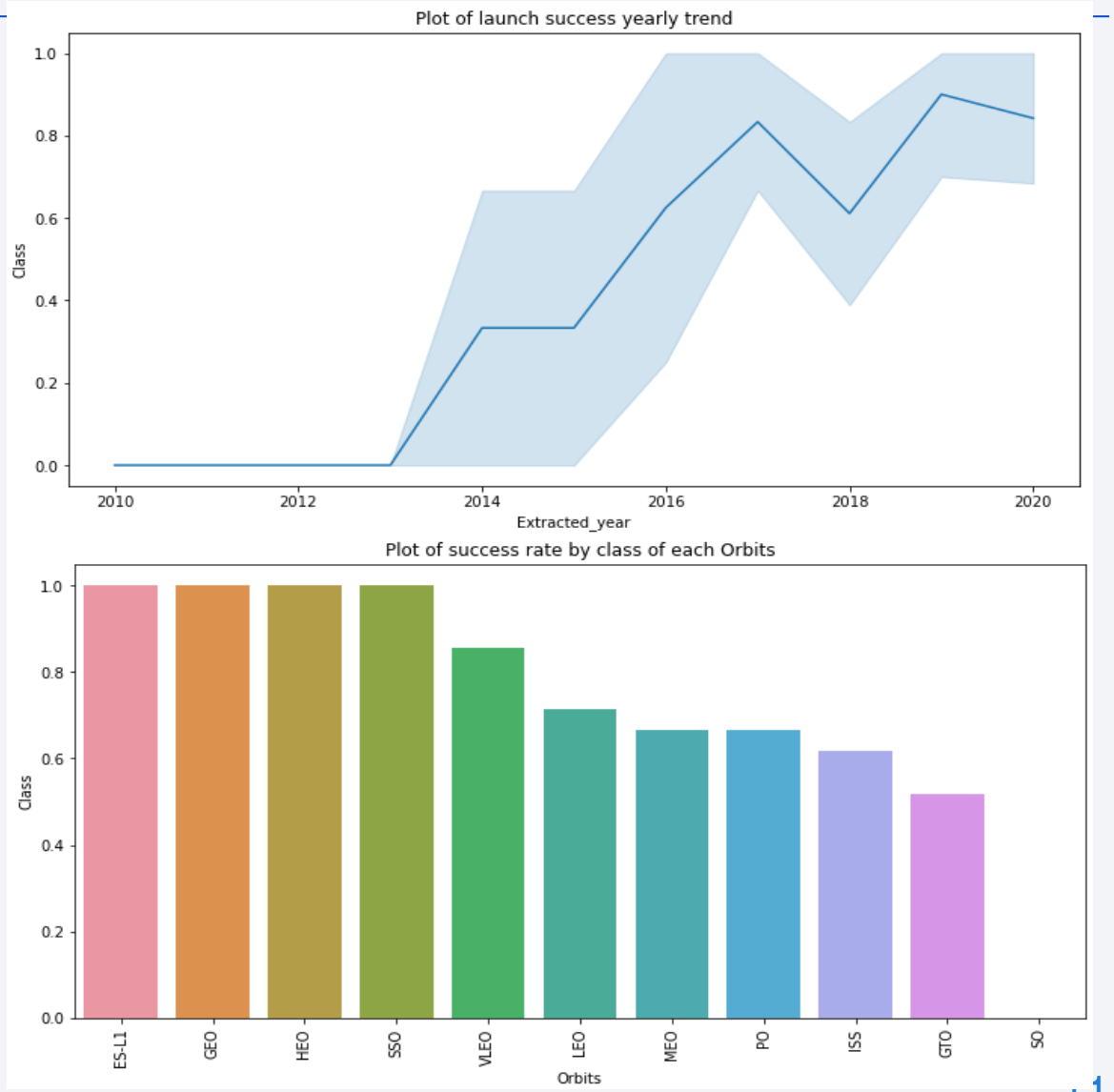
Github: https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/Data%20Wrangling.ipynb

- We conducted exploratory data analysis (EDA) to identify patterns and relationships within the dataset and defined the training labels.
- Specifically, we analyzed the number of launches at each site and examined the frequency and distribution of different orbit types.
- Additionally, we created a landing outcome label derived from the outcome column to facilitate predictive modeling.
- The processed results were then exported to a CSV file for further analysis and model training.

EDA with Data Visualization

We explored the SpaceX dataset by visualizing various relationships, including:

- The correlation between flight numbers and launch sites.
- Payload mass in relation to different launch sites.
- Success rates for each orbit type.
- The relationship between flight numbers and orbit types.
- Payload mass compared to different orbit types.
- Trends in launch success rates over the years.
- Github: https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/EDA%20with%20Data%20Visualization.ipynb



EDA with SQL

In a Jupyter notebook, we loaded the SpaceX dataset into a PostgreSQL database and performed exploratory data analysis (EDA) using SQL. Our queries provided insights into the following:

- Unique names of launch sites for space missions.

- Total payload mass carried by boosters for NASA (CRS) missions.

- Average payload mass for boosters of version F9 v1.1.

- Total counts of successful and failed mission outcomes.

- Failed landing outcomes on drone ships, including the associated booster version and launch site names.

- Booster versions that have carried the maximum payload mass.

Github: https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We visualized launch sites using Folium maps, incorporating markers, circles, and lines to indicate launch outcomes (success or failure) at each site.
- We assigned launch outcomes to binary classes-0 for failure and 1 for success and used color-coded marker clusters to identify launch sites with higher success rates.
- Additionally, we calculated distances from each launch site to nearby features and investigated questions such as:
  - Are launch sites located near railways, highways, or coastlines?
  - Do launch sites maintain specific distances from urban areas?

Github:

https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We developed an interactive dashboard using Plotly Dash to enhance data visualization. The dashboard features:
  - Pie Charts: Displaying the total number of launches for each launch site.
  - Scatter Plots: Illustrating the relationship between launch outcomes and payload mass (in kilograms) for different booster versions.
- In addition, we provided interactive elements for users to explore these visualizations and gain deeper insights into launch site performance and payload metrics.

Predictive Analysis (Classification)

- We loaded the SpaceX dataset using NumPy and Pandas, transformed the data, and split it into training and testing sets.
- We built various machine learning models and utilized GridSearchCV to tune hyperparameters.
- Accuracy was used as the evaluation metric. To enhance model performance, we applied feature engineering and algorithm tuning.
- Through this process, we identified the best-performing classification model.

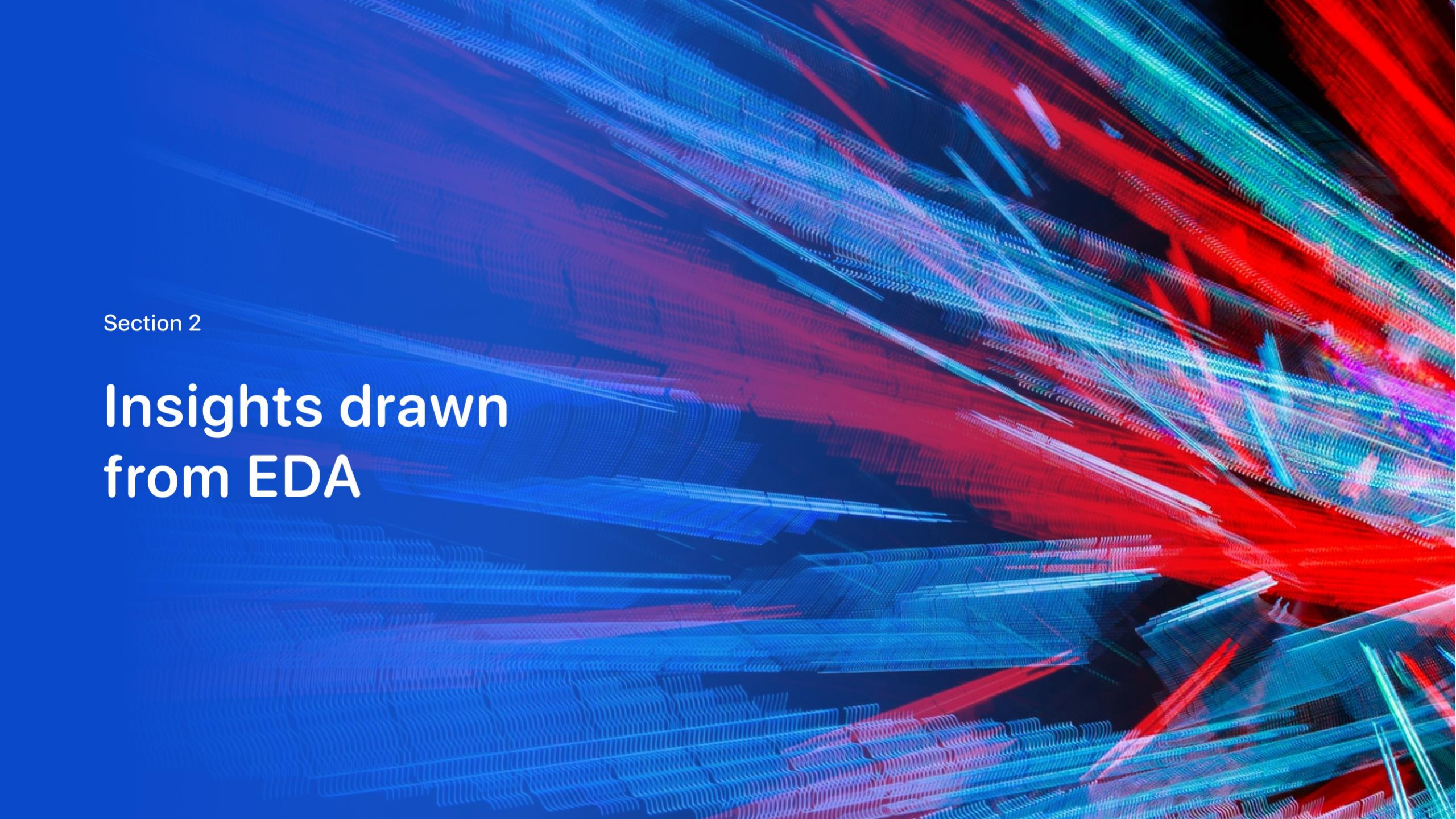
Github: https://github.com/dhanushparsa/IBM-DataScience-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

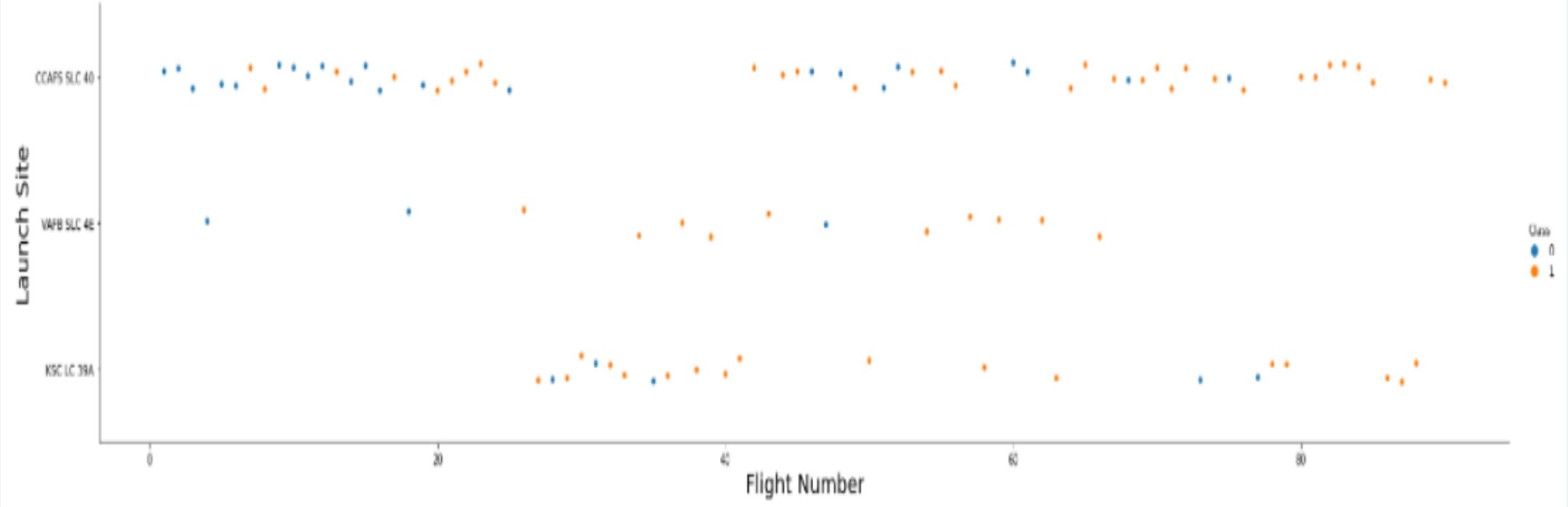
Section 2

Insights drawn from EDA



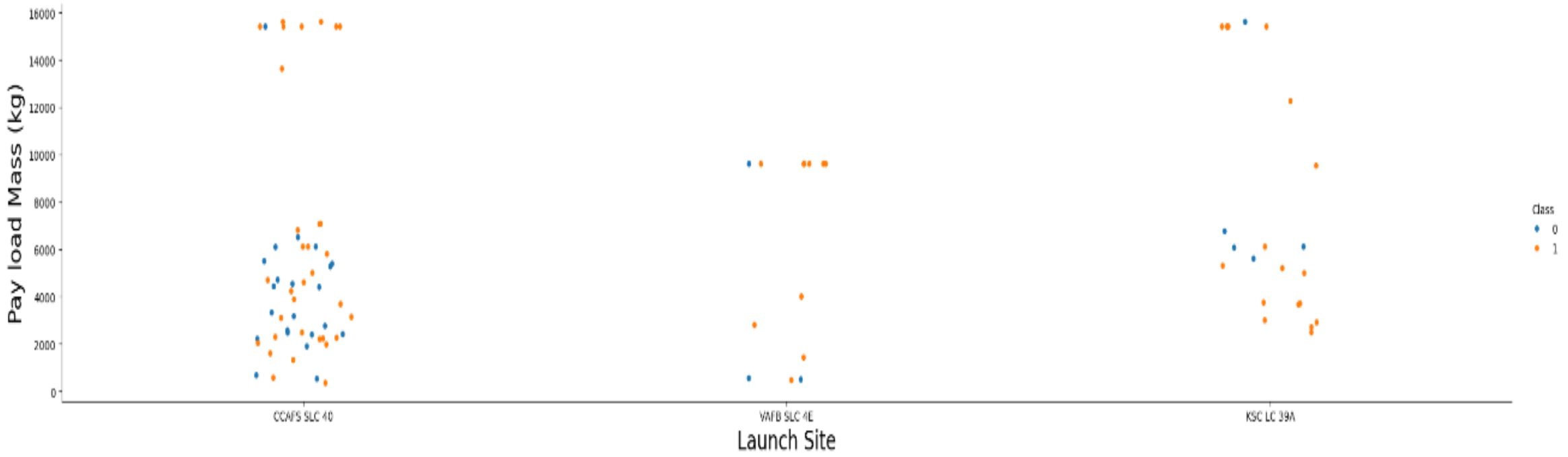
Flight Number vs. Launch Site

From the plot, we found that the with increasing flight number, the success rate at a launch site is improving.



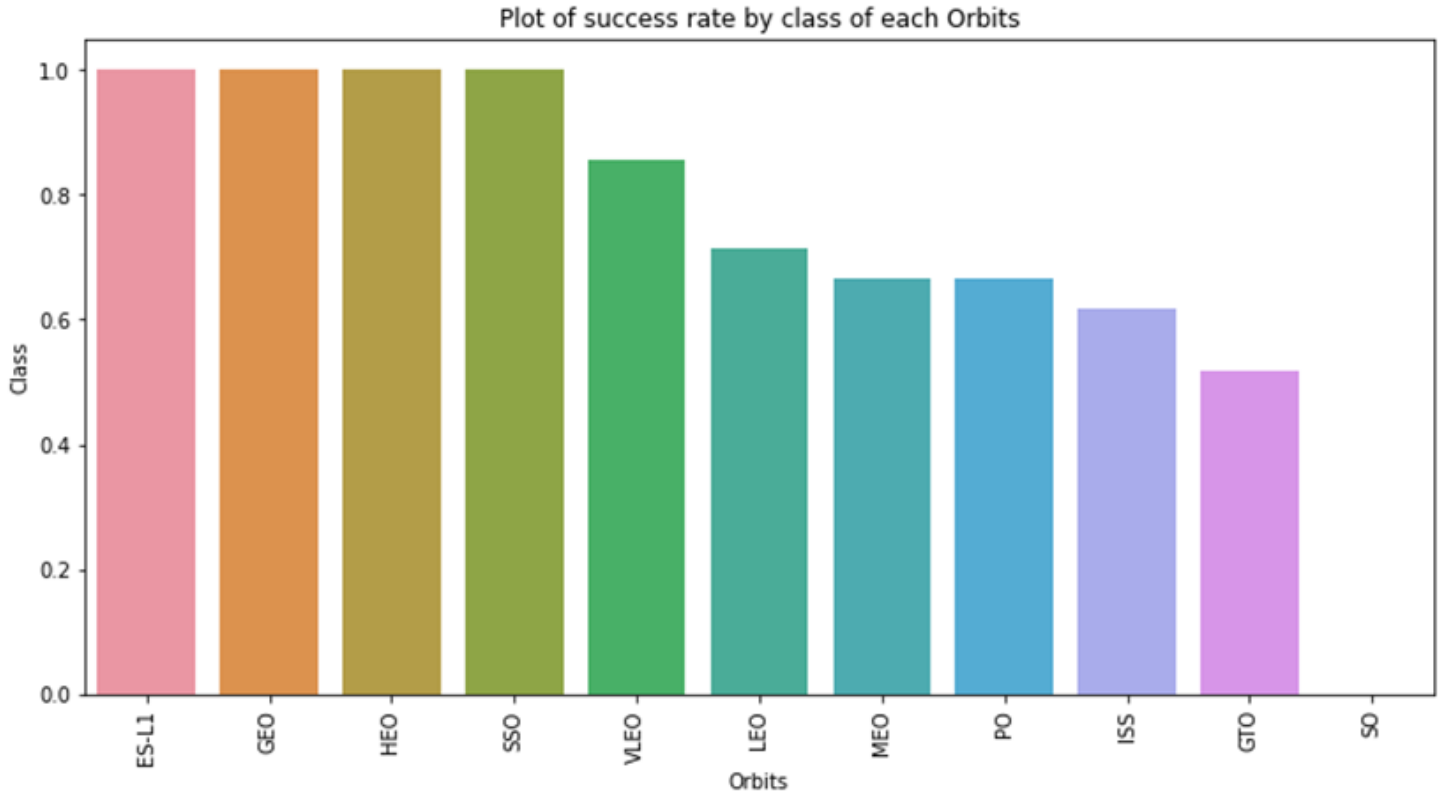
Payload vs. Launch Site

In the scatter plot of Payload Mass versus Launch Site, it is evident that the VAFB-SLC launch site does not have any rockets launched with a payload mass greater than 10,000 kg.



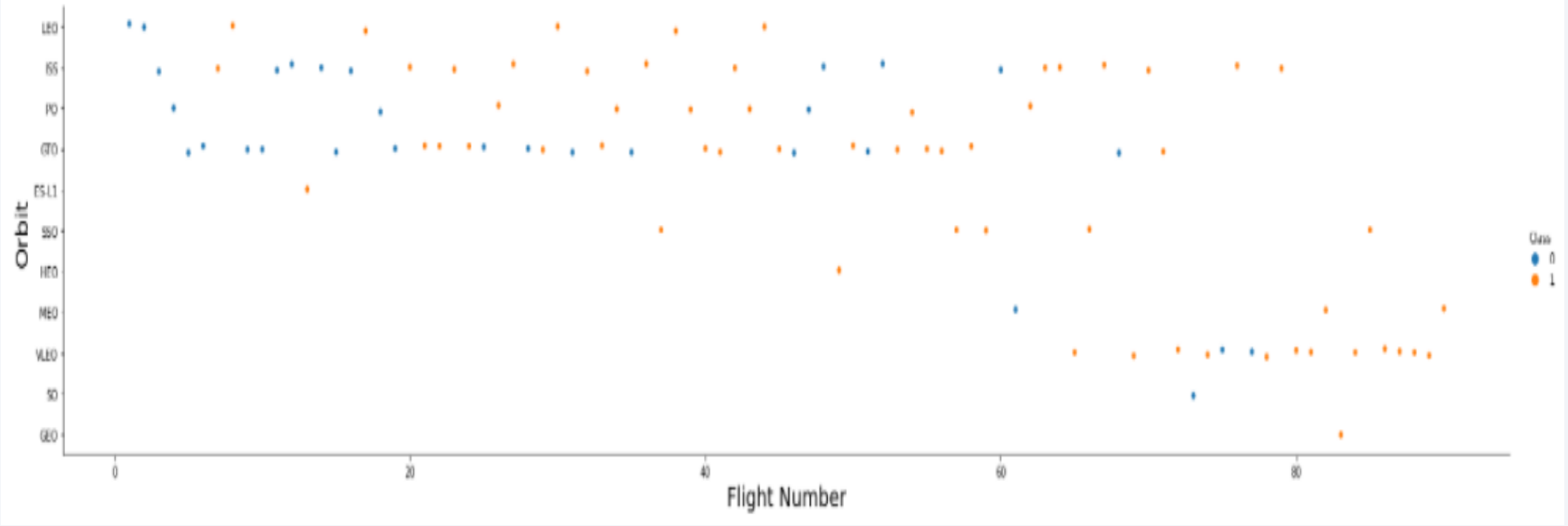
Success Rate vs. Orbit Type

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



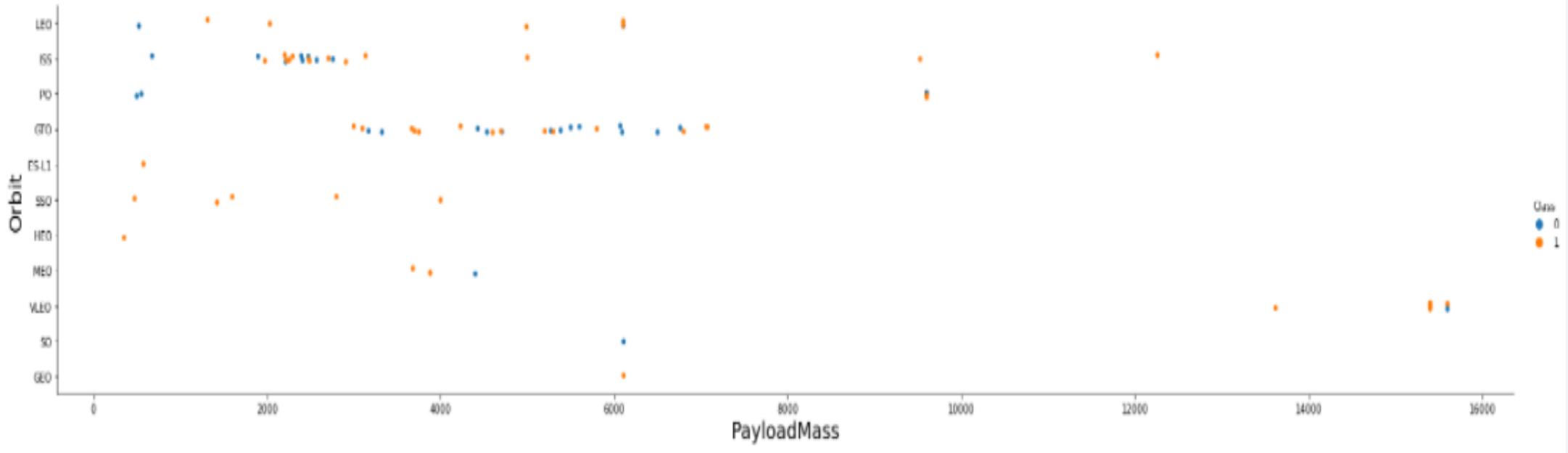
Flight Number vs. Orbit Type

- In the plot, it appears that in Low Earth Orbit (LEO), success is correlated with the number of flights. However, for Geostationary Transfer Orbit (GTO), there is no apparent relationship between the number of flights and success rates.



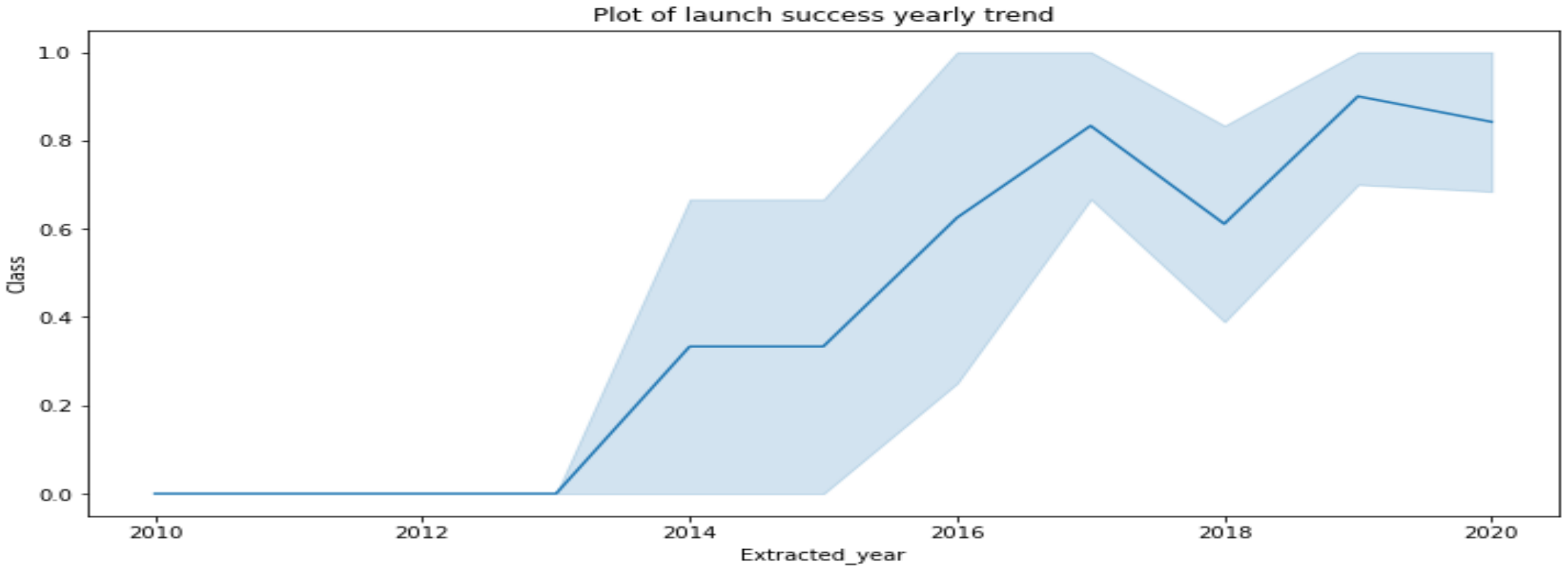
Payload vs. Orbit Type

- For heavy payloads, the success rate of landings is higher for Polar, Low Earth Orbit (LEO), and International Space Station (ISS) missions. In contrast, for Geostationary Transfer Orbit (GTO) missions, it is challenging to differentiate between successful and unsuccessful landings, as both outcomes are present.



Launch Success Yearly Trend

The plot reveals that the success rate has been steadily increasing from 2013 through 2020.



All Launch Site Names

- We used the DISTINCT keyword to display only the 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 a query to display 5 records where launch sites start with CCA, and applied the LIMIT function to restrict the results to 5 rows.

Total Payload Mass

- We calculated the total payload mass carried by boosters for NASA (CRS) missions to be 45,596 kg using the query provided below.

```
pd.read_sql("select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer ='NASA (CRS)'",con)
```

	sum(PAYLOAD_MASS__KG_)
0	45596

Average Payload Mass by F9 v1.1

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

```
pd.read_sql("select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version ='F9 v1.1'",con)
```

	avg(PAYLOAD_MASS__KG_)
0	2928.4

First Successful Ground Landing Date

We noted that the first successful landing outcome on a 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

- We used the WHERE clause to filter for boosters that successfully landed on a drone ship, and applied an AND condition to further narrow down the results to those with a payload mass between 4,000 and 6,000 kg.

```
import pandas as pd

query = """
SELECT Booster_Version, Landing_Outcome
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
"""
df = pd.read_sql(query, con)
df
```

	Booster_Version	Landing_Outcome
0	F9 FT B1022	Success (drone ship)
1	F9 FT B1026	Success (drone ship)
2	F9 FT B1021.2	Success (drone ship)
3	F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- We used the wildcard % in the WHERE clause to filter for records where the MissionOutcome was either 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

```
pd.read_sql("select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)",con)
```

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

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

2015 Launch Records

- We combined the WHERE clause with LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes on drone ships, along with the corresponding booster versions and launch site names, specifically 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

- We selected landing outcomes and their counts from the data, using the WHERE clause to filter for landing outcomes between 2010-06-04 and 2010-03-20.
- We then applied the GROUP BY clause to group the landing outcomes and used the ORDER BY clause to sort the results in descending order.

the count of landing outcomes (such as Failure (drone ship)

```
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)
```

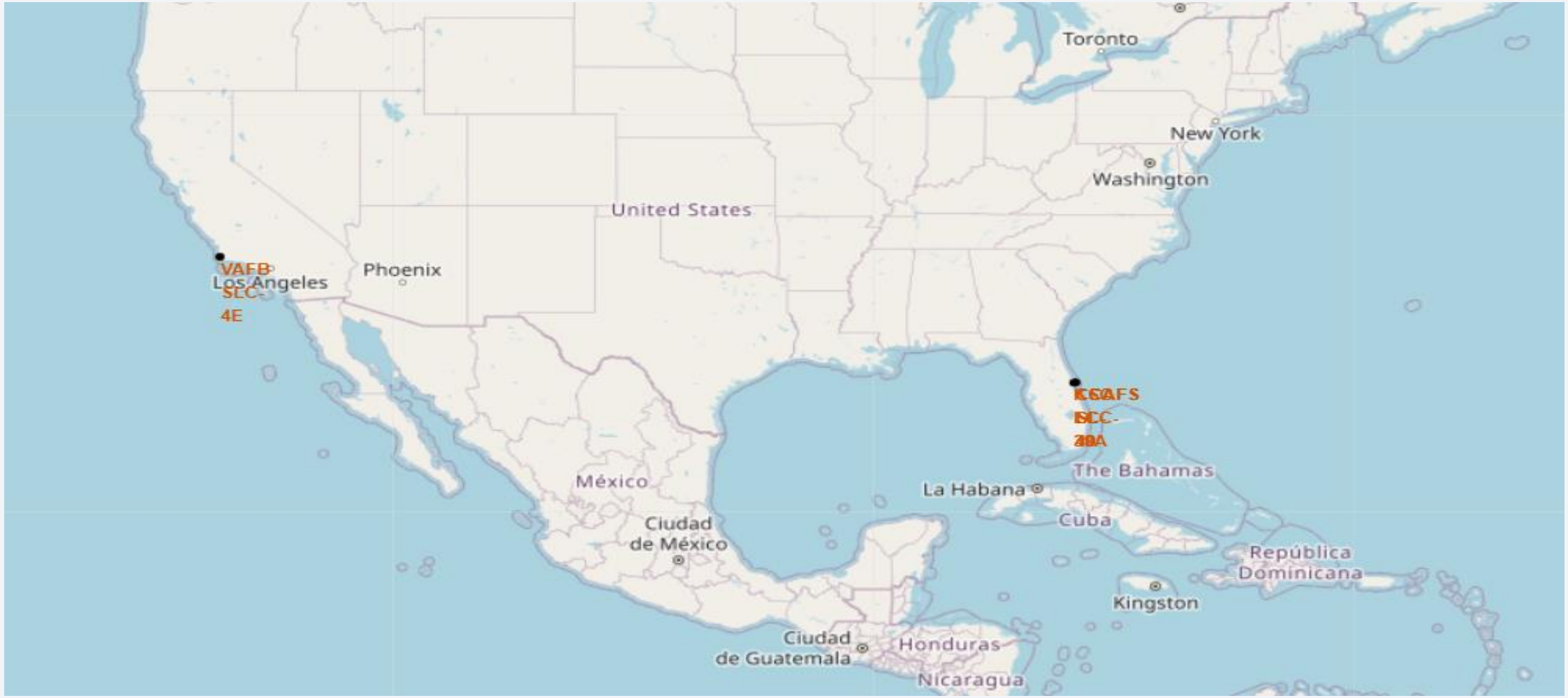
19]:

	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

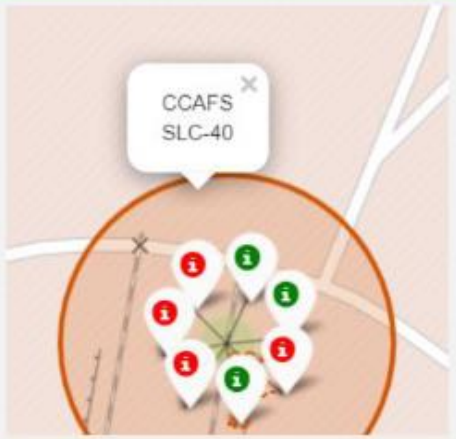
Section 4

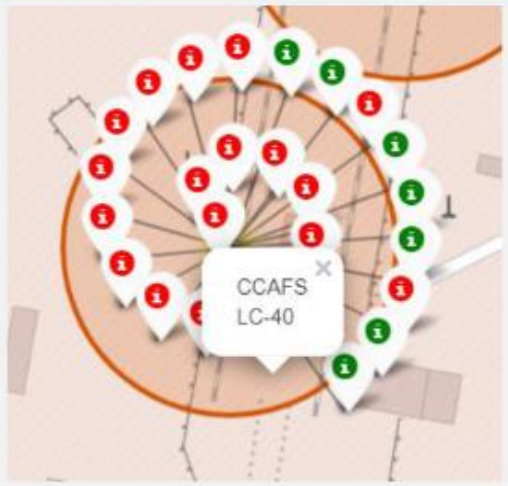
Launch Sites Proximities Analysis

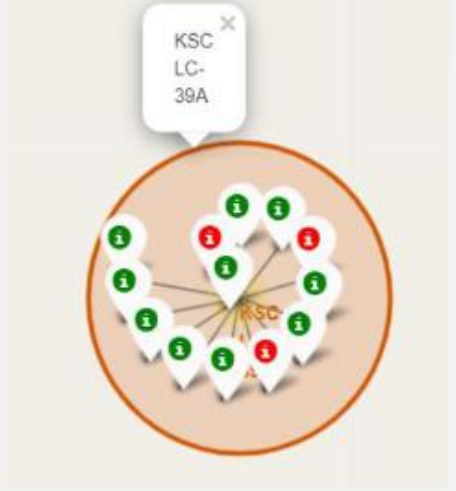
All launch sites global map markers



Markers showing launch sites with color labels





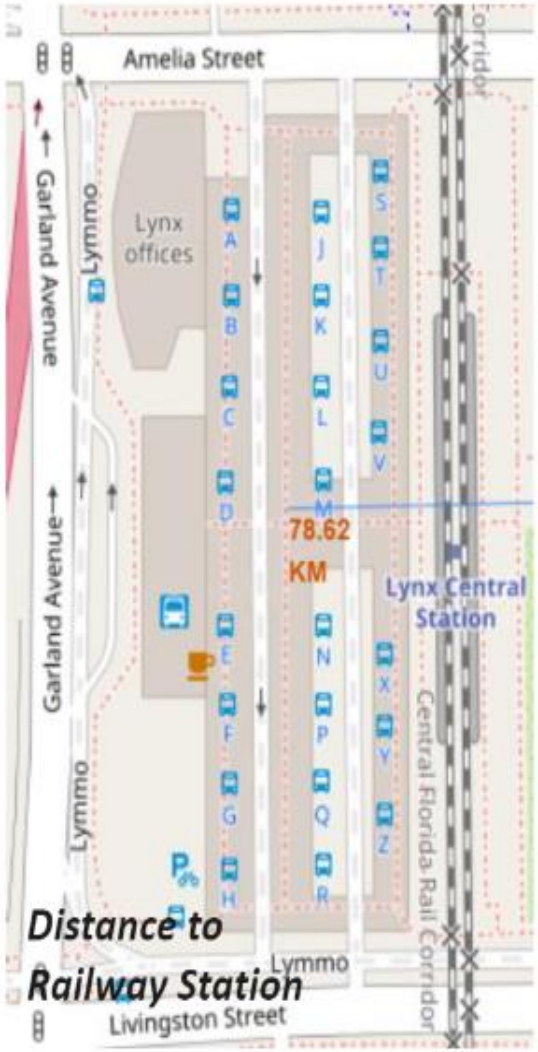


Florida Site:
Green marker – Successful launch
Red Marker - Failure

California
Launching Site

37

Launch Site distance to landmarks



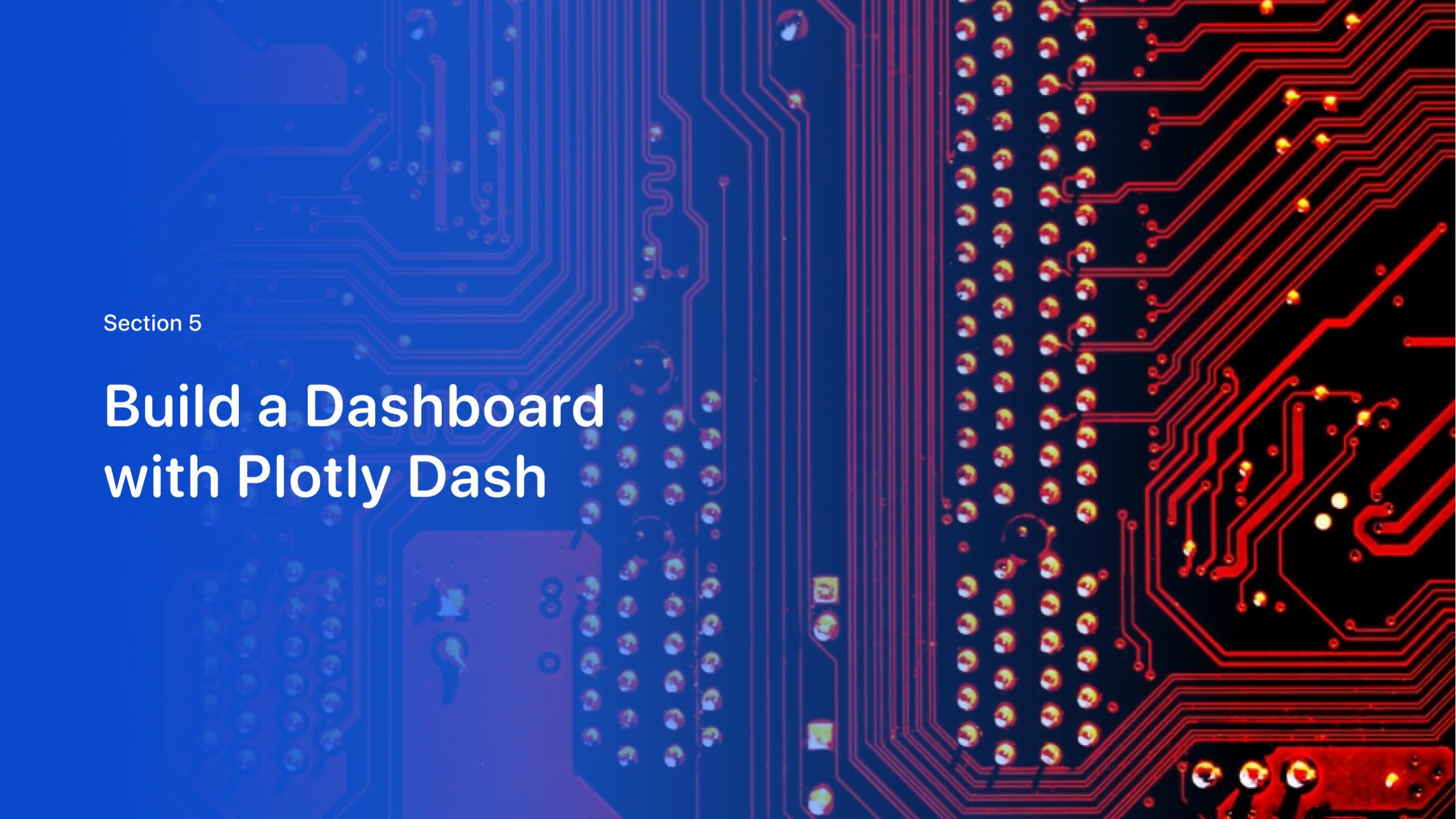
Launching site are not in close proximity to:

- Railways
- Highways
- Cities

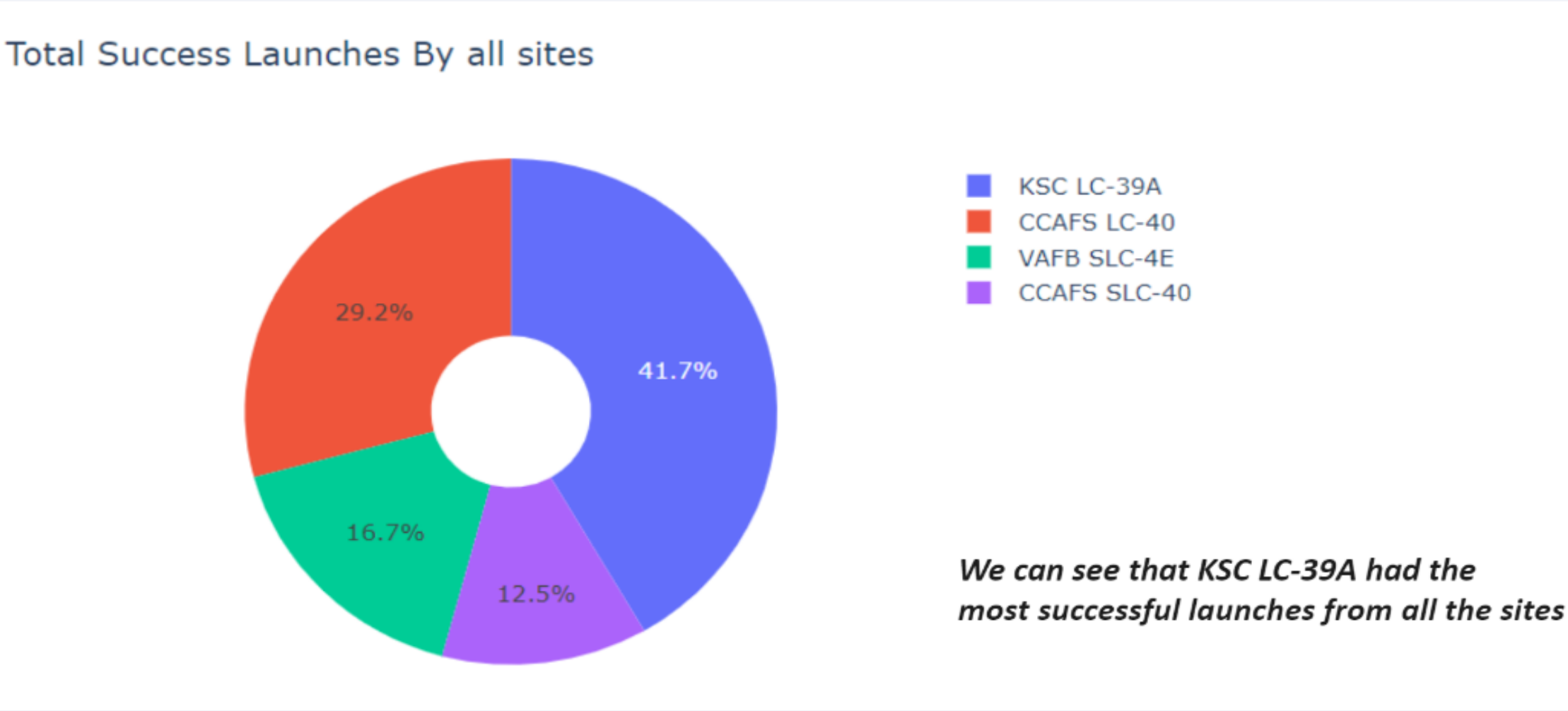
They are in close proximity to coastline

Section 5

Build a Dashboard with Plotly Dash

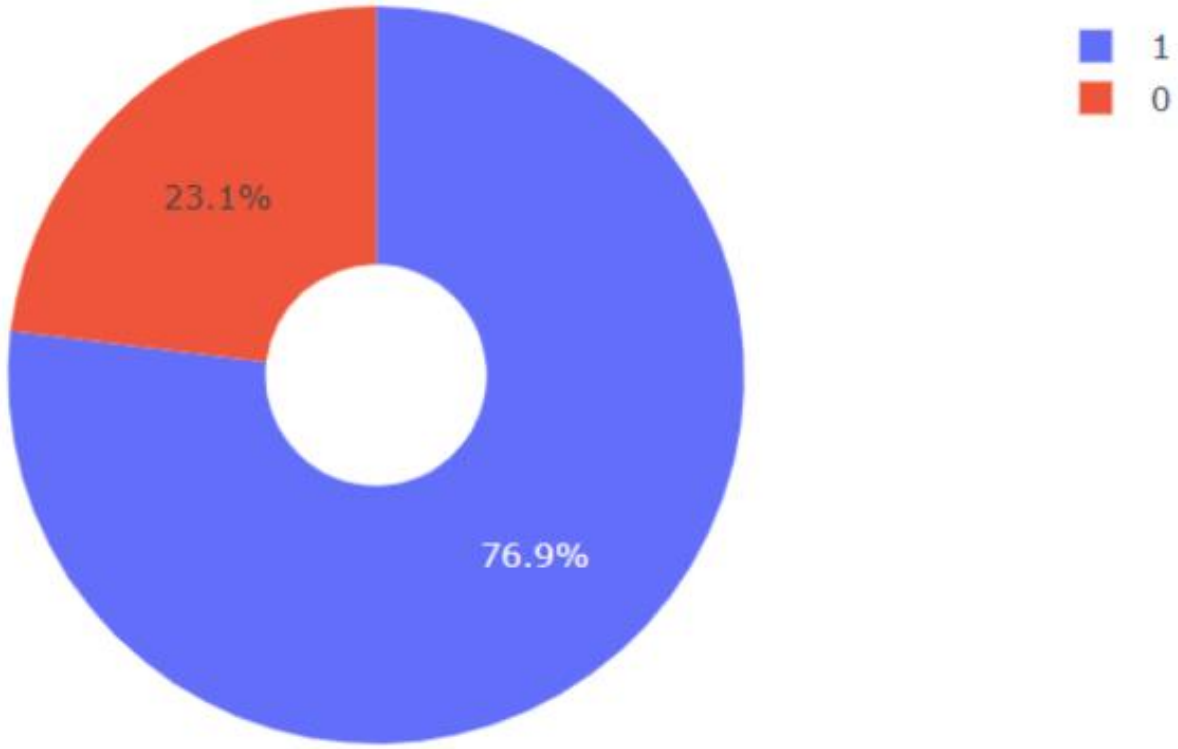


Pie chart showing the success percentage achieved by each launch site



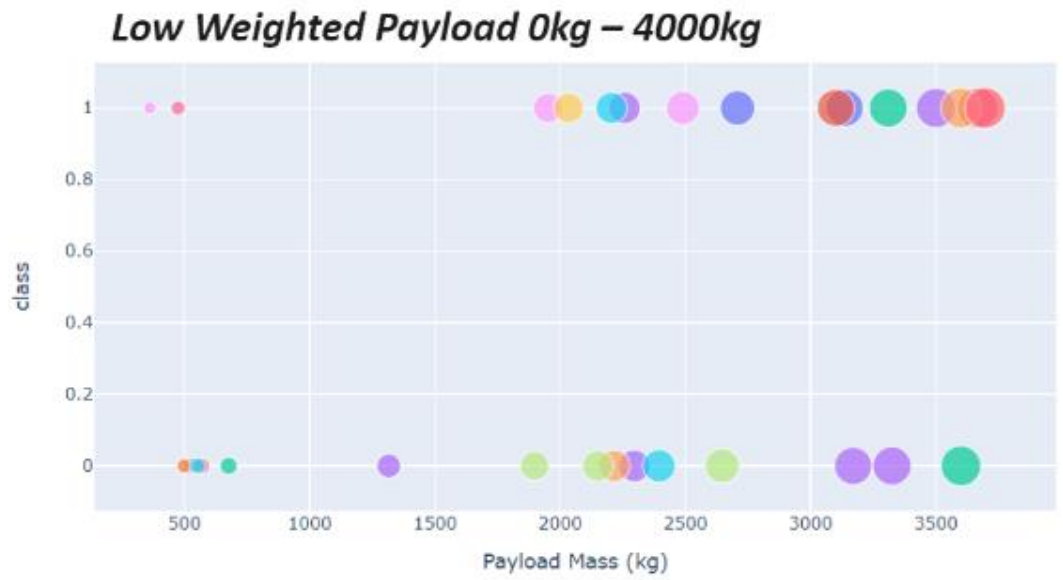
We can see that KSC LC-39A had the most successful launches from all the sites

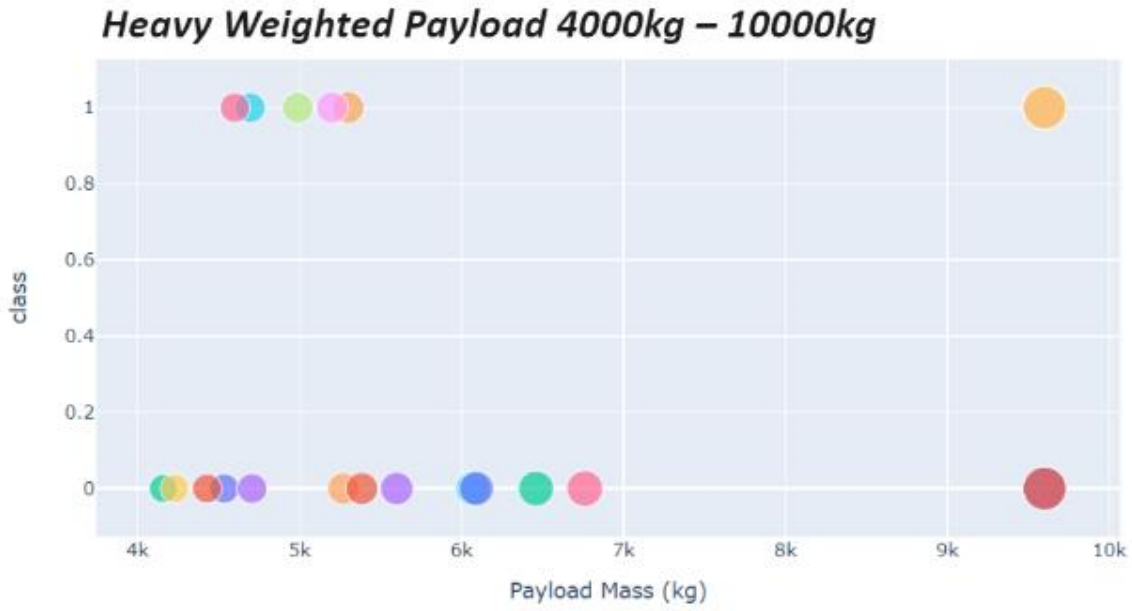
Pie chart showing the Launch site with the highest launch success ratio



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

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

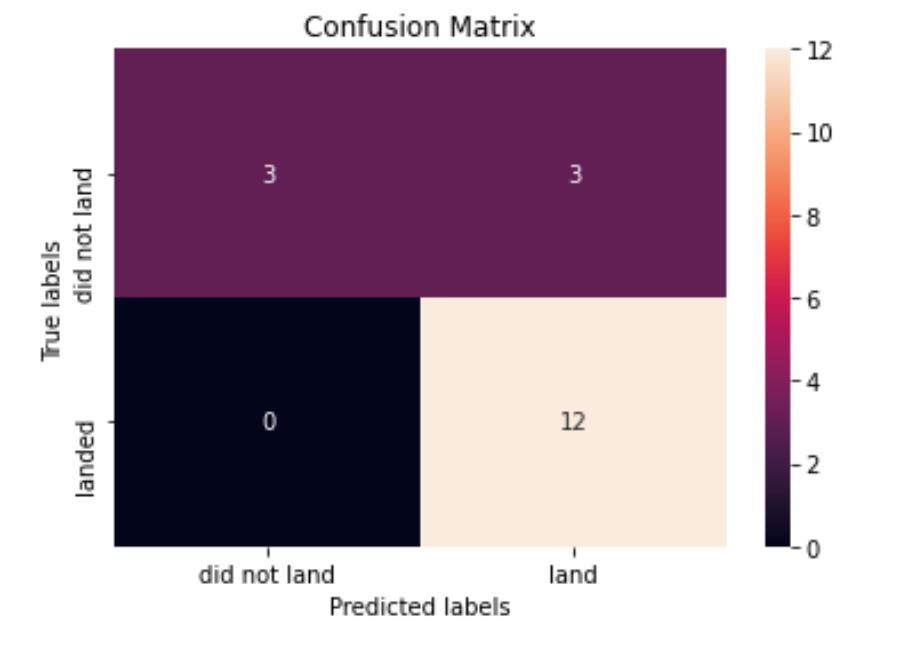
```
tree_cv=GridSearchCV(tree,param_grid=parameters,cv=10)
tree_cv.fit(X_train,Y_train)
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 1
0, 'splitter': 'best'}
accuracy : 0.8857142857142858
```

Confusion Matrix

- The confusion matrix for the decision tree classifier indicates that the model can distinguish between the different classes. However, a significant issue is the presence of false positives, where the classifier incorrectly marks unsuccessful landings as successful ones.



Conclusions

- Flight Volume and Success Rate: There is a positive correlation between the number of flights at a launch site and its success rate—higher flight volumes tend to result in higher success rates.

- Launch Success Trend: The launch success rate showed a steady increase from 2013 to 2020, indicating continuous improvements in technology and processes.

- Top Orbits: Orbits such as ES-L1, GEO, HEO, SSO, and VLEO demonstrated the highest success rates, showcasing their reliability for space missions.

- KSC LC-39A Dominance: The Kennedy Space Center's LC-39A launch site achieved the highest number of successful launches among all sites, establishing it as a critical location for mission success.

- Best Performing Model: Among the machine learning models evaluated, the Decision Tree classifier emerged as the best-performing algorithm for this task, despite the challenge of false positives. This model effectively distinguishes between different classes and can be fine-tuned further for even better accuracy.

Thank you!