IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joseph Candelaresi
2024-09-09

Outline

Executive Summary

Summary of Methodologies:

- ✓ **Data Collection through API**: This method involves using Application Programming Interfaces (APIs) to retrieve data from various sources. APIs provide a structured way to access live data feeds or databases, enabling automated and efficient data collection.
- ✓ **Data Collection with Web Scraping**: Web scraping involves extracting data from websites by parsing the HTML code. This technique is useful for gathering information that is not readily available through APIs, allowing for the collection of diverse and extensive datasets.
- ✓ **Data Wrangling**: Also known as data munging, this process involves cleaning, transforming, and organizing raw data into a more usable format. Data wrangling is essential for ensuring the quality and usability of data, making it ready for analysis.
- ✓ **Exploratory Data Analysis with SQL**: Using Structured Query Language (SQL), this phase focuses on examining and summarizing datasets. It helps in identifying patterns, spotting anomalies, testing hypotheses, and checking assumptions, providing a solid foundation for further analysis.
- ✓ **Exploratory Data Analysis with Data Visualization**: This step uses graphical representations to visually explore data. Data visualization helps in understanding trends, outliers, and patterns within the dataset, making complex data more accessible and interpretable.
- ✓ **Interactive Visual Analytics with Folium**: Folium is a Python library used for creating interactive maps. This technique involves visualizing geospatial data, allowing users to interact with the maps and explore the data in-depth, enhancing the analytical experience.
- ✓ **Machine Learning Prediction**: The final step involves applying machine learning algorithms to predict future trends or behaviors based on historical data. Machine learning models can identify patterns and make accurate predictions, providing valuable insights for decision-making

Summary of all results:

- ✓ **Exploratory Data Analysis (EDA)**: This step involves summarizing the main characteristics of the data, often using visual methods to uncover patterns, spot anomalies, and test hypotheses.
- ✓ **Interactive Analytics**: Screenshots of interactive dashboards or visual tools are used to provide a dynamic way to explore and understand the data, allowing for deeper insights through user interaction.
- ✓ **Predictive Analytics**: This involves using statistical models or machine learning algorithms to make predictions about future events based on historical data.

Introduction

Project background and context:

SpaceX advertises Falcon 9 rocket launches at a cost of $62 million, significantly lower than other providers who charge upwards of $165 million. The primary reason for this cost efficiency is SpaceX's ability to reuse the first stage of the rocket. Therefore, predicting the successful landing of the first stage is crucial for determining the overall cost of a launch. This project aims to develop a machine learning pipeline to predict the likelihood of the first stage landing successfully, providing valuable insights for companies looking to compete with SpaceX in the rocket launch market.

Problems you want to find answers:

1. **Prediction Accuracy**: How accurately can we predict the first stage landing?
2. **Cost Savings**: What are the potential cost savings from accurate predictions?
3. **Model Generalization**: How well does the model perform under different conditions?

Section 1

Methodology

Executive Summary

This project aims to predict the successful landing of SpaceX's Falcon 9 first stage using a comprehensive data analysis approach. The data collection was performed using the SpaceX API and web scraping from Wikipedia, ensuring a robust dataset. Data wrangling involved applying one-hot encoding to transform categorical features into a suitable format for modeling.

Exploratory Data Analysis (EDA) was conducted using visualization tools and SQL to uncover patterns and insights within the data. Interactive visual analytics were performed using Folium and Plotly Dash, providing dynamic and user-interactive representations of the data.

The final phase focused on predictive analysis through classification models. This involved building, tuning, and evaluating various models to optimize their performance in predicting the successful landing of the first stage. The methodology aims to provide actionable insights and accurate predictions, supporting decision-making in the competitive rocket launch market.

Data Collection

Describe how data sets were collected.

1.Data Collection from SpaceX API:

- We initiated data collection by sending a GET request to the SpaceX API. This API provides comprehensive data on SpaceX launches, rockets, and other related information.
- The response content from the API was decoded as JSON using the .json() function call. This JSON data was then transformed into a pandas DataFrame using the .json_normalize() function, which helps in flattening the nested JSON structure.

2.Data Cleaning and Preprocessing:

- The collected data was cleaned to ensure its quality and usability. This involved checking for missing values and filling them where necessary to maintain data integrity.

3.Web Scraping from Wikipedia:

- To supplement the data from the SpaceX API, we performed web scraping on Wikipedia to gather Falcon 9 launch records. Using BeautifulSoup, we extracted the launch records presented in HTML tables.
- These tables were parsed and converted into pandas Data Frames, making them ready for future analysis.

This comprehensive approach ensured that we had a robust dataset for our analysis, combining API data with additional information from web scraping.

Data Collection – SpaceX API

Data Collection Process Using SpaceX REST API:

1. **Initiate Data Collection**:
   1. **GET Request**: We used a GET request to the SpaceX API to retrieve data on launches, rockets, and other relevant information.
2. **Data Cleaning**:
   1. **Clean Data**: The requested data was cleaned to ensure accuracy and consistency. This involved removing any irrelevant information and correcting any errors.
3. **Data Wrangling and Formatting**:
   1. **Basic Data Wrangling**: We performed basic data wrangling to structure the data appropriately. This included handling missing values, normalizing data, and converting it into a usable format.
   2. **Formatting**: The cleaned and wrangled data was then formatted into a pandas DataFrame for further analysis.

GitHub URL: https://github.com/Seph71/Capstone-Project-Space-X/blob/main/Data%20Collection.PNG

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

Web Scraping Process:

1.Web Scraping Application:
We applied web scraping to extract Falcon 9 launch records from Wikipedia using BeautifulSoup.

2.Parsing HTML Table:
We parsed the HTML table containing the launch records.

3.Data Conversion:
The parsed table was converted into a pandas DataFrame for further analysis.

This process ensures that the data extracted from the web is structured and ready for analysis.

GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/Scrapping.PNG

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
```

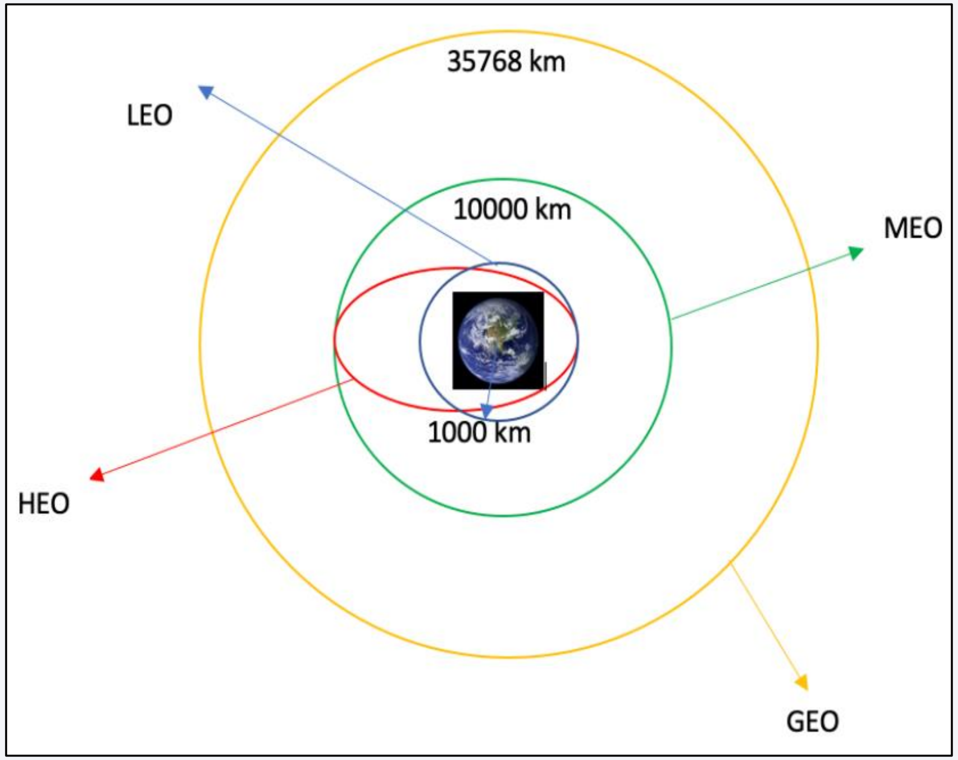
4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

How data was processed:

1. **Exploratory Data Analysis (EDA)**: We performed exploratory data analysis to understand the dataset better and identify any patterns or anomalies. This step helped us determine the training labels necessary for our analysis.
2. **Calculation of Launches and Orbits**: We calculated the number of launches at each site and analyzed the number and occurrence of each orbit type. This step was crucial for understanding the distribution and frequency of launches and orbits.
3. **Creation of Landing Outcome Label**: From the outcome column, we created a landing outcome label. This new label was essential for categorizing the landing results and facilitating further analysis.
4. **Exporting Results**: Finally, we exported the processed data, including the newly created landing outcome label, to a CSV file for easy access and further use.

This process ensured that the data was clean, well-structured, and ready for analysis.



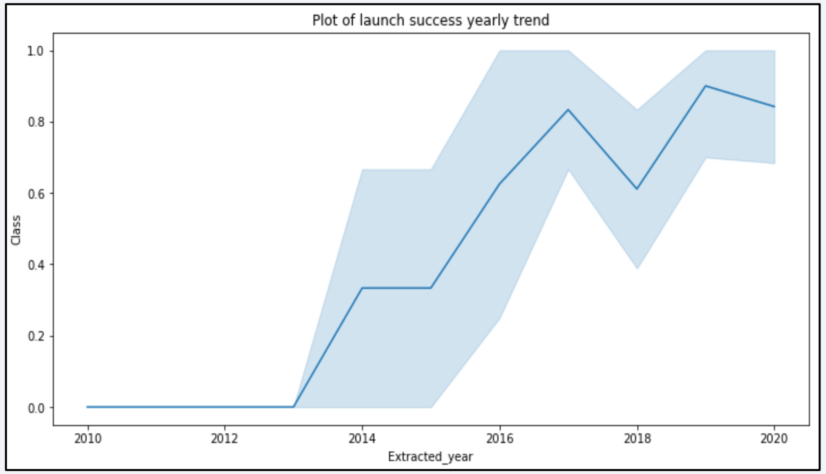
GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/Wrangling.PNG

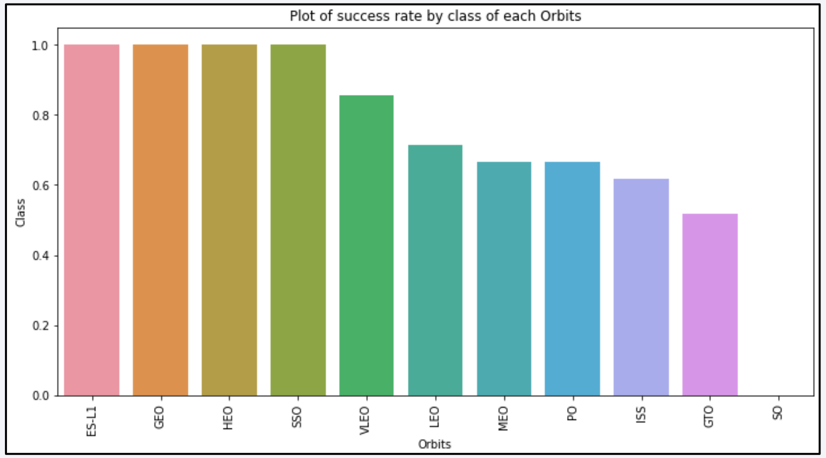
EDA with Data Visualization

Summary of charts plotted and why used:

1. **Flight Number vs. Launch Site**: Identifies patterns in flight numbers across different launch sites.
2. **Payload vs. Launch Site**: Examines the relationship between payload mass and launch site preferences.
3. **Success Rate of Each Orbit Type**: Analyzes which orbit types have higher success rates.
4. **Flight Number vs. Orbit Type**: Explores the relationship between flight numbers and orbit types.
5. **Launch Success Yearly Trend**: Tracks the success rate of launches over the years.

These charts help uncover patterns and trends crucial for strategic decision-making in space missions





GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/EDA%20Visualization.PNG

EDA with SQL

Summary of SQL queries performed:

- Retrieved unique launch sites from the SpaceX dataset.
- Displayed records where launch sites start with ‘CCA’.
- Calculated the total payload mass for NASA (CRS) missions.
- Determined the average payload mass for booster version F9 v1.1.
- Found the date of the first successful ground pad landing.
- Listed boosters with successful drone ship landings and specific payload mass criteria.
- Counted the total number of successful versus failed missions.
- Identified booster versions with the maximum payload mass.
- Extracted records of failed drone ship landings, including booster versions and launch sites for 2015.
- Ranked the count of various landing outcomes within a specified date range in descending order

GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/EDA%20SQL.PNG

Build an Interactive Map with Folium

Summary statement of the map objects added to the folium map and the reasons for their inclusion:

Summary Statement:

We created a folium map to visualize launch sites and their outcomes. The map includes the following objects:

1. **Markers**: Placed at each launch site to indicate its location.
2. **Circles**: Added around each marker to represent the success or failure of launches. The circles are color-coded based on the outcome (0 for failure, 1 for success).
3. **Lines**: Drawn to show distances between launch sites and nearby features such as railways, highways, coastlines, and cities.

Explanation:

- **Markers**: These were added to pinpoint the exact locations of all launch sites on the map, providing a clear visual reference.
- **Circles**: By using color-coded circles, we could easily distinguish between successful and failed launches. This visual differentiation helps in quickly assessing the performance of each site.
- **Lines**: These were included to measure and display the distances between launch sites and their proximities to railways, highways, coastlines, and cities. This information is crucial for answering questions about the strategic placement of launch sites, such as their accessibility and safety considerations.
- Using these map objects, we were able to identify which launch sites have relatively high success rates through color-labeled marker clusters. Additionally, we could analyze the geographical factors influencing the placement of launch sites, such as their proximity to infrastructure and urban areas.

GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/Map.PNG

Build a Dashboard with Plotly Dash

Summary of plots/graphs and interactions added to dashboard:

We developed an interactive dashboard using Plotly Dash, incorporating the following visualizations and interactions:

1. **Pie Charts**: Displaying the total launches by specific sites.
2. **Scatter Graph**: Illustrating the relationship between Outcome and Payload Mass (Kg) for different booster versions.

Explanation:

Pie Charts:

- **Purpose**: To provide a clear and immediate visual representation of the distribution of total launches across various sites.
- **Reason**: Pie charts are effective for showing proportions and making it easy to compare the number of launches from different sites at a glance.

Scatter Graph:

- **Purpose**: To explore and visualize the relationship between the outcome of launches and the payload mass for different booster versions.
- **Reason**: Scatter graphs are ideal for identifying patterns, correlations, and potential outliers in the data, helping to understand how payload mass impacts launch outcomes across different booster versions.

These visualizations and interactions were chosen to enhance the user's ability to analyze and interpret the data effectively, providing insights into launch distributions and performance metrics.

GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/PieCharts.PNG

Predictive Analysis (Classification)

Summary of Model Development Process

1. **Data Loading and Transformation**:
    - We loaded the data using **numpy** and **pandas**.
    - Transformed the data to prepare it for modeling.
2. **Data Splitting**:
    - Split the data into **training and testing** sets to ensure unbiased evaluation.
3. **Model Building and Hyperparameter Tuning**:
    - Built various machine learning models.
    - Tuned different hyperparameters using **GridSearchCV** to find the optimal settings.
4. **Model Evaluation**:
    - Used **accuracy** as the primary metric to evaluate model performance.
5. **Model Improvement**:
    - Improved the model through **feature engineering** and **algorithm tuning**.
6. **Best Performing Model**:
    - Identified the best performing classification model based on the evaluation metrics.

This structured approach ensured a thorough and effective model development process, leading to the identification of the most accurate classification model.

GitHub URL:https://github.com/Seph71/Capstone-Project-Space-X/blob/main/Prediction%20Outcome.PNG

Results

✓Exploratory data analysis results

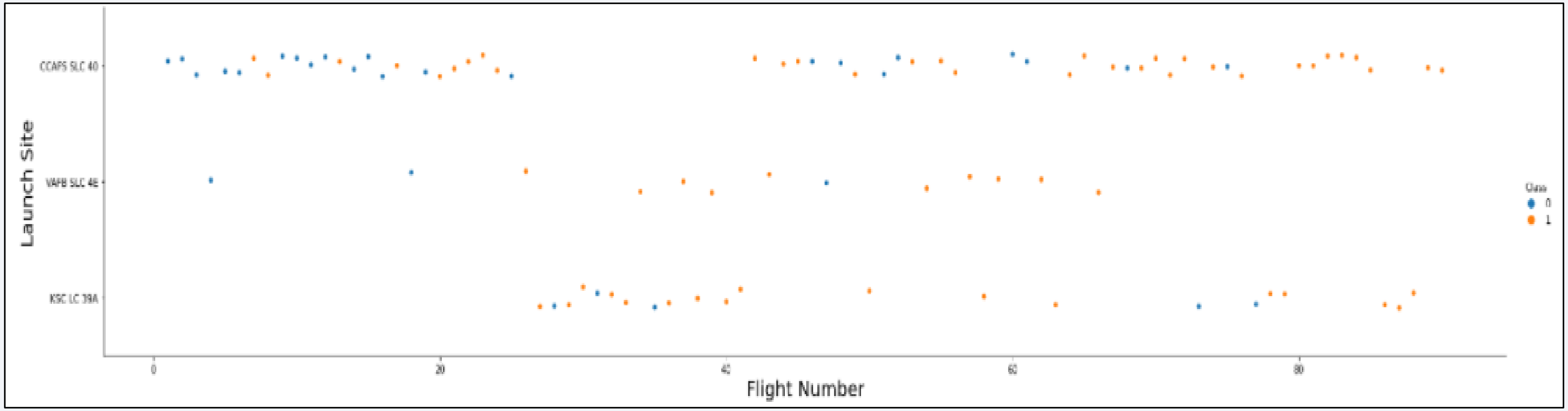
✓Interactive analytics demo in screenshots

✓Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



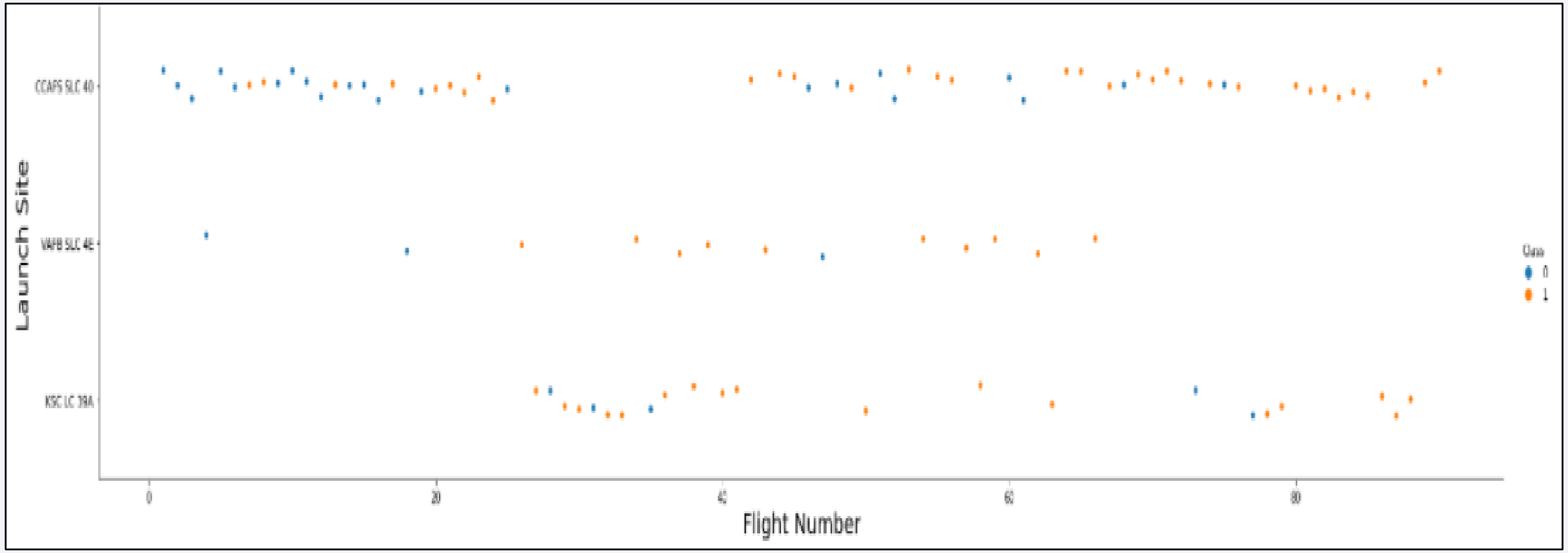
Explanation of Scatter Plot Insights

The scatter plot reveals that:

1. **No Clear Correlation**: Success rates (orange points) don't clearly correlate with flight numbers.
2. **Mixed Outcomes**: Both successful (orange) and unsuccessful (blue) launches are spread across all sites.
3. **Higher Flight Volume, Higher Success**: Sites with more flights tend to have higher success rates.

These insights highlight that experience and volume of launches may improve success rates.

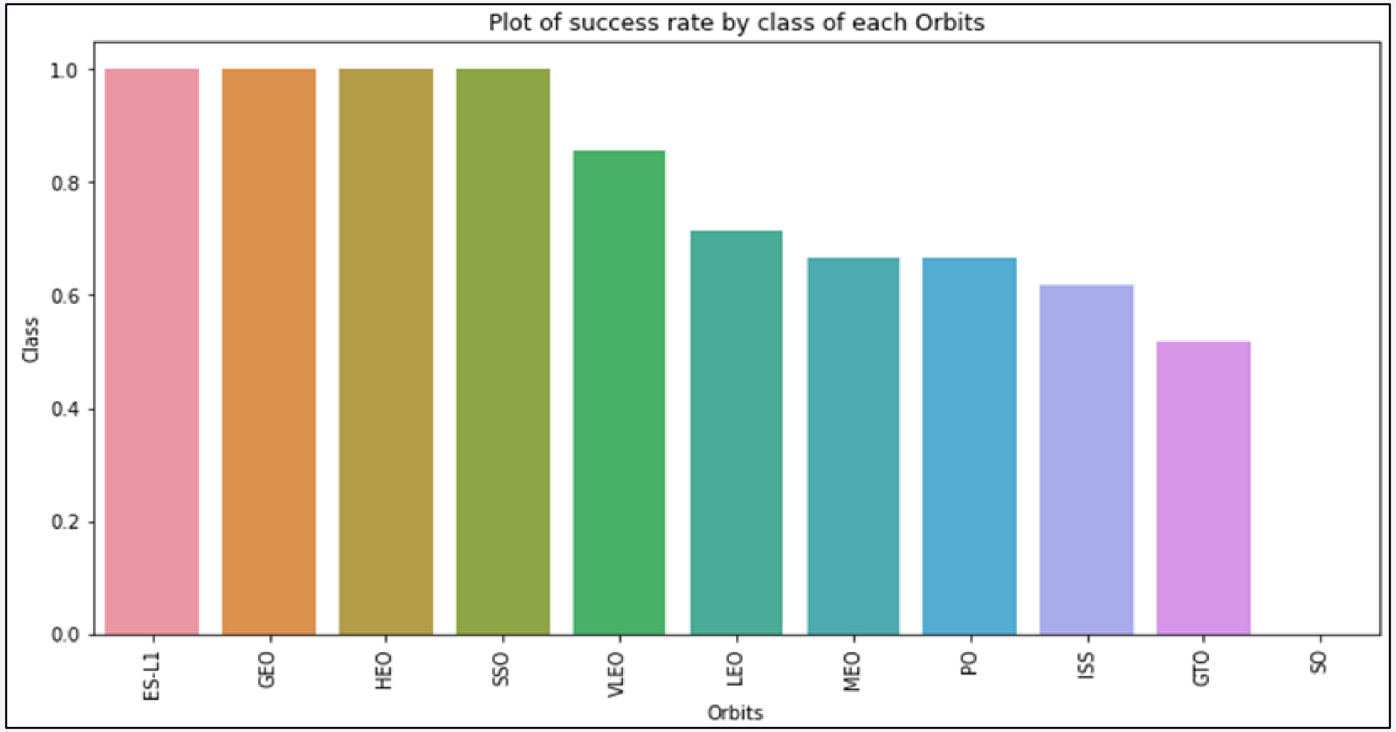
Payload vs. Launch Site



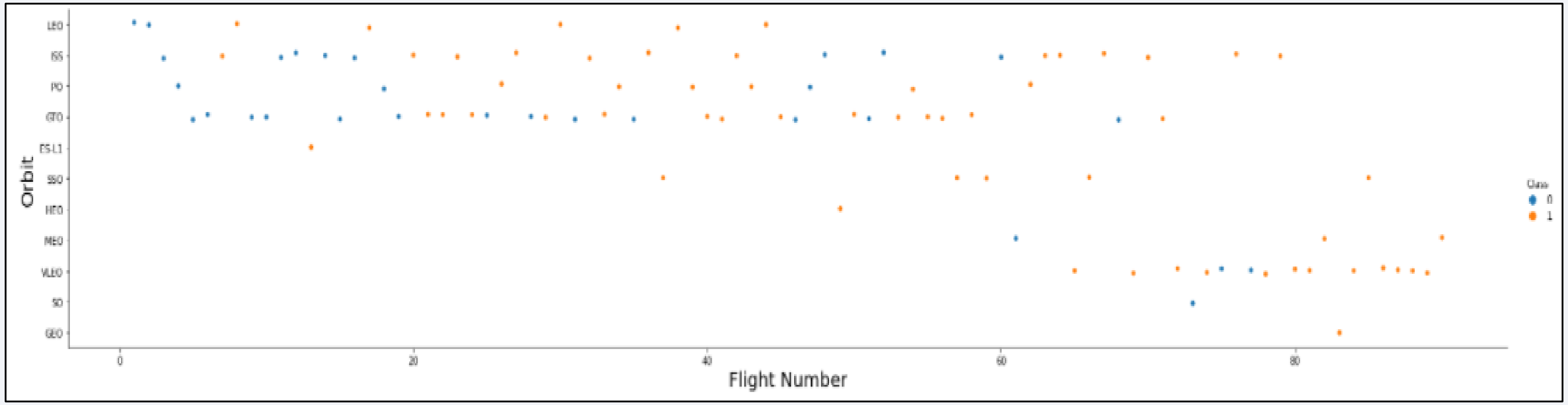
The scatter plot shows that successful launches (orange points) occur across a wide range of payload masses, with no clear pattern linking success to payload size. Both successful and unsuccessful launches (blue points) are seen at various payload sizes for each launch site

Success Rate vs. Orbit Type

The bar chart reveals that ES-L1, SSO, and VLEO orbits have the highest success rates, while GTO and MEO have lower success rates. This suggests that certain orbit types may present more challenges or have different mission profiles affecting their success rates. Additionally, ES-L1, GEO, HEO, SSO, and VLEO are the orbits with the most success.

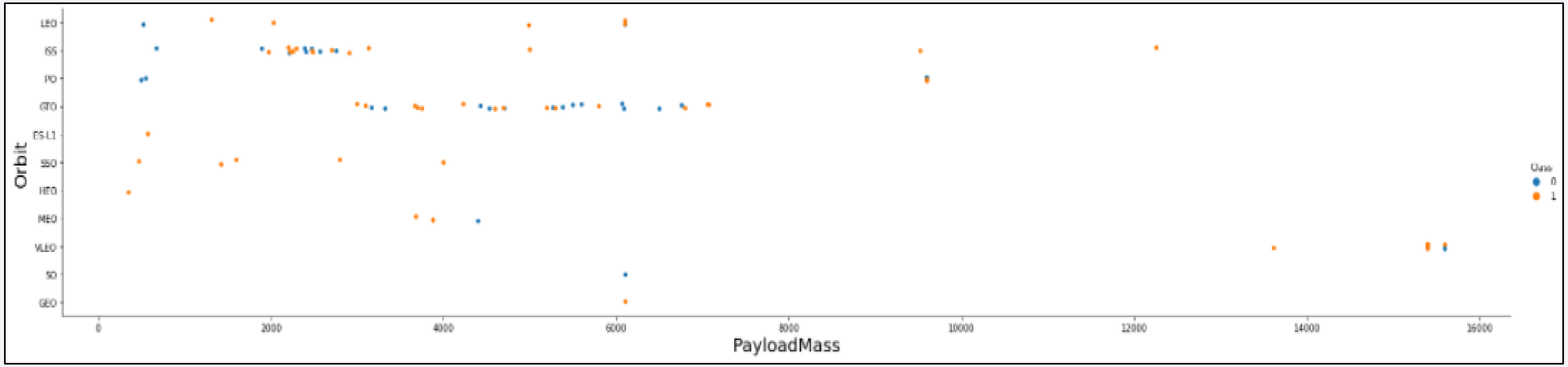


Flight Number vs. Orbit Type



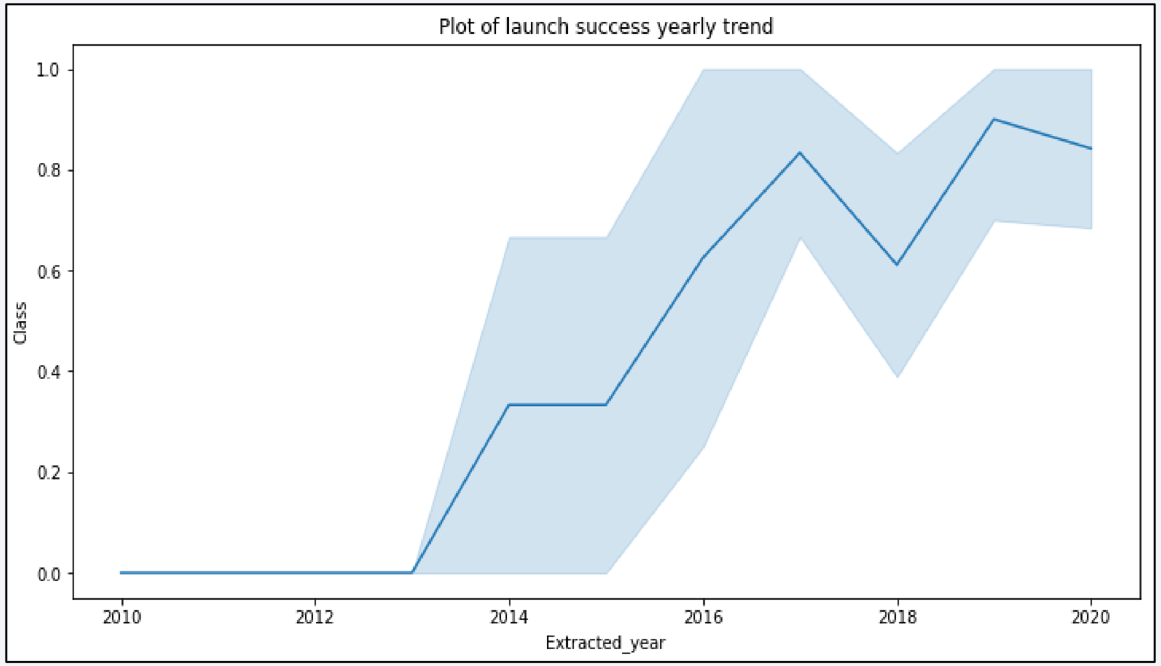
The plot shows that in the LEO orbit, success is related to the number of flights, while in the GTO orbit, there is no such relationship. The scatter plot, with blue points for failures (Class 0) and orange points for successes (Class 1), suggests that success rates generally increase with flight number, indicating improvements in technology or processes. It also highlights that certain orbits like ISS have high success rates, whereas others like SO face more challenges.

Payload vs. Orbit Type



The scatter plot shows that successful landings with heavy payloads are more common for PO, LEO, and ISS orbits. Most successful launches carry payloads under 10,000 kg, with some exceptions. GTO launches have a broad range of payload masses, indicating flexibility. There is also a cluster of successful launches with higher payloads, suggesting advancements in launch vehicle capabilities or mission profiles.

Launch Success Yearly Trend



The line graph shows that the success rate of launches has been increasing since 2013, peaking around 2017. Although it fluctuates after 2017, the success rate generally remains high, indicating improved reliability over the years. The shaded area likely represents the confidence interval or variance in the data.

All Launch Site Names

We displayed the unique launch site names by using the keyword **DISTINCT** to filter and show only the unique entries from the SpaceX data.

	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 displayed the first 5 records where launch sites begin with CCA using the query above.

Total Payload Mass

```
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

We calculated the total payload carried by boosters from NASA as 45596 using the query above.

Average Payload Mass by F9 v1.1

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

	avg_payloadmass
0	2928.4

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

First Successful Ground Landing Date

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

Observation: The date of the first successful landing outcome on ground pad was 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

5]:

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

We used the **WHERE** clause to filter for boosters that successfully landed on a drone ship and applied the **AND** condition to identify those with a payload mass between 4000 and 6000 kg. Following this, we displayed the list of booster names that met these criteria.

Total Number of Successful and Failure Mission Outcomes

Outcomes of Success: 100

Outcomes of Failure: 1

```
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

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

We determined the boosters that carried the maximum payload using a subquery in the **WHERE** clause along with the **MAX()** function. Following this, we displayed the names of the boosters that have carried the maximum payload mass.

2015 Launch Records

	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)

We used a combination of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes on drone ships, their booster versions, and launch site names for the year 2015. Following this, the failed landing outcomes, booster versions, and launch site names for 2015 are displayed

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

	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 their counts from the data, using the **WHERE** clause to filter for outcomes between 2010-06-04 and 2017-03-20. We then applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to sort them in descending order. Following this, the ranked counts of landing outcomes (such as Failure on drone ship or Success on ground pad) between these dates are displayed in descending order.

Section 3

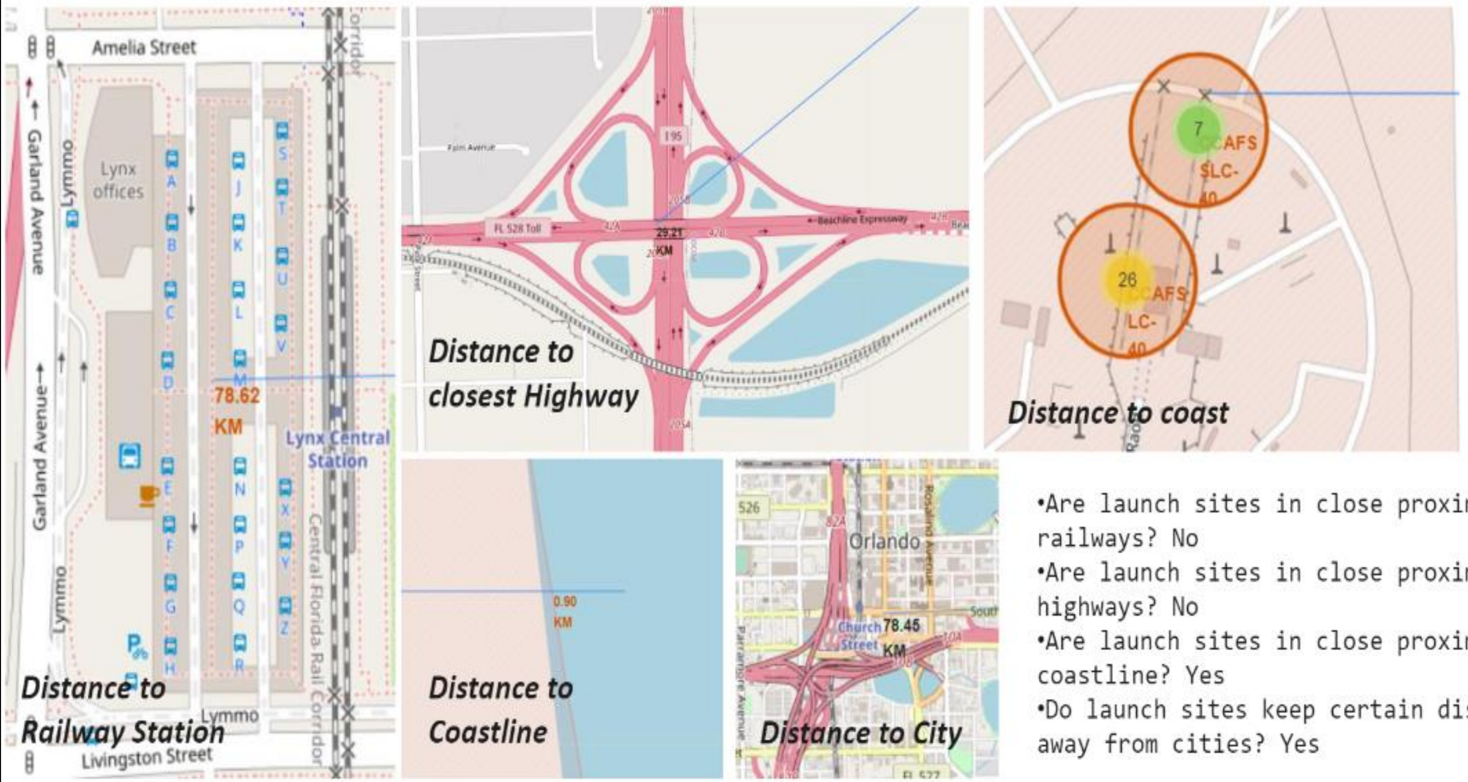
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

Launch Sites are located from coast to coast. Florida and California.

<Folium Map Screenshot 2>

<Folium Map Screenshot 3>

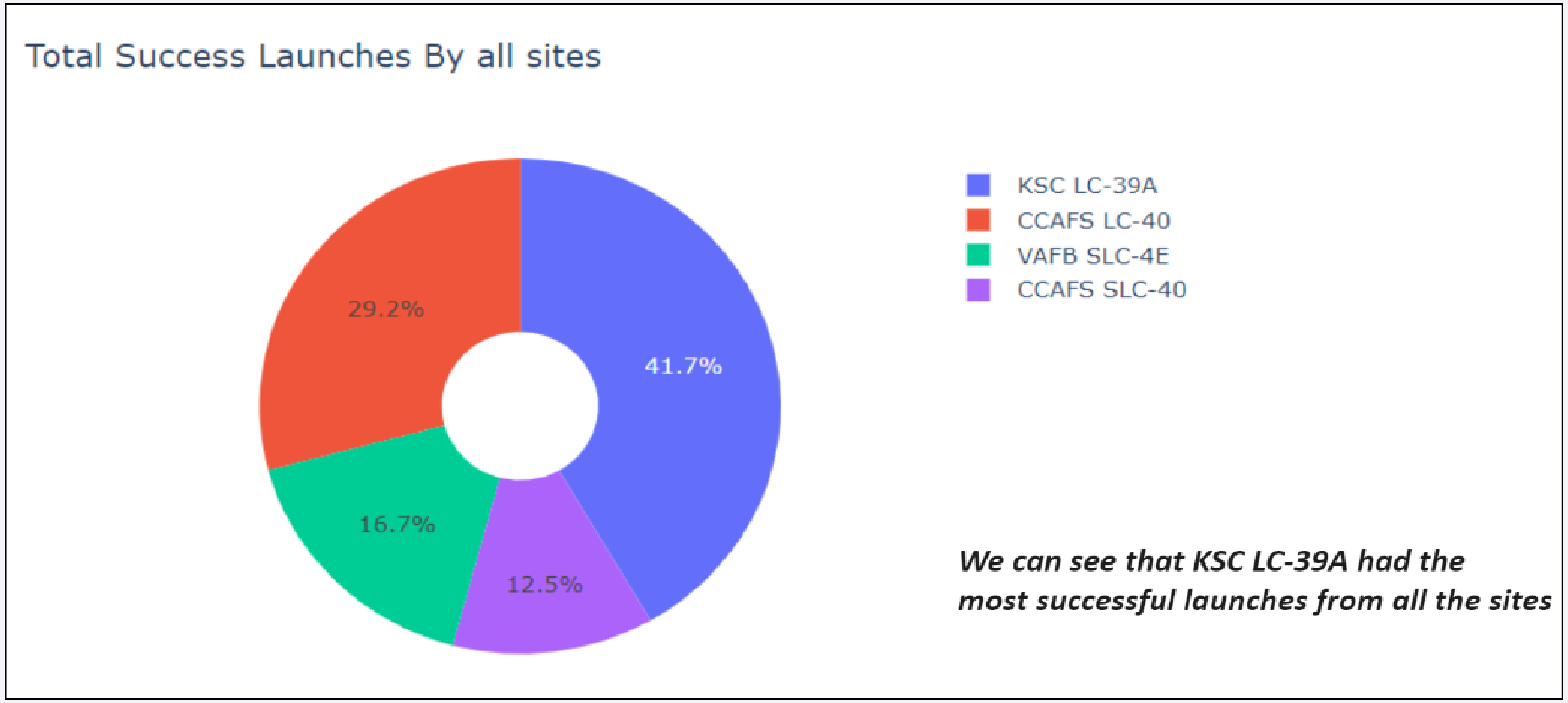


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

Build a Dashboard with Plotly Dash

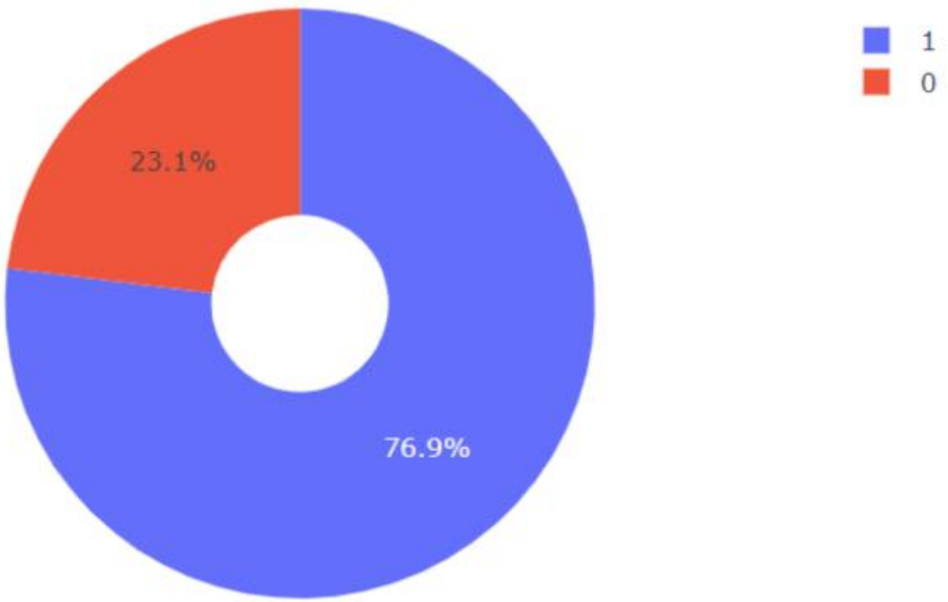
<Dashboard Screenshot 1>



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

<Dashboard Screenshot 2>

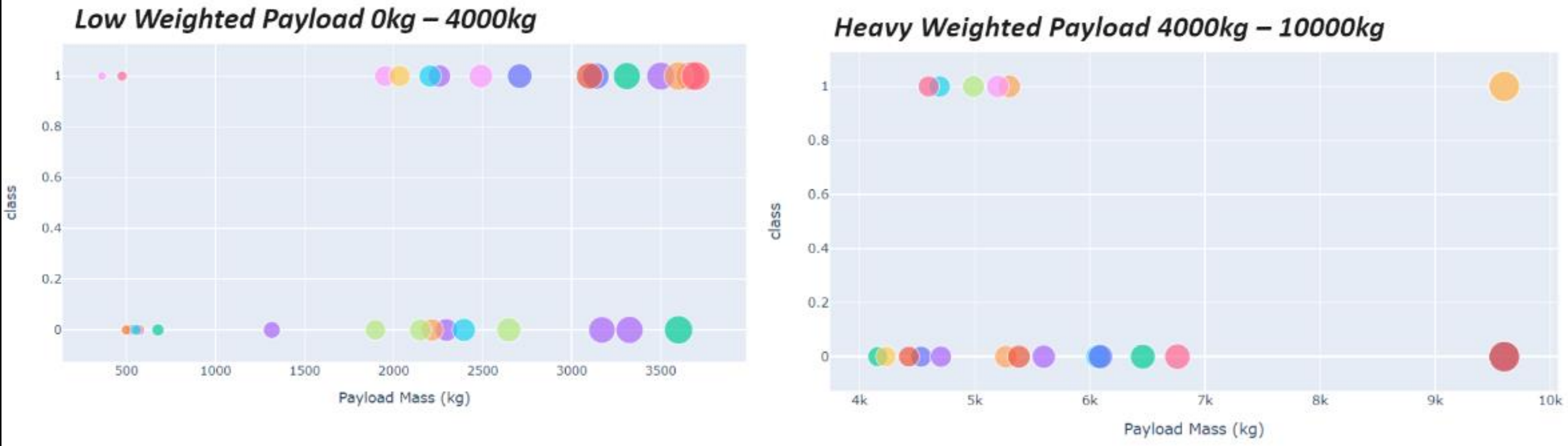
Pie chart showing the Launch site with the highest launch success ratio



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

<Dashboard Screenshot 3>

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

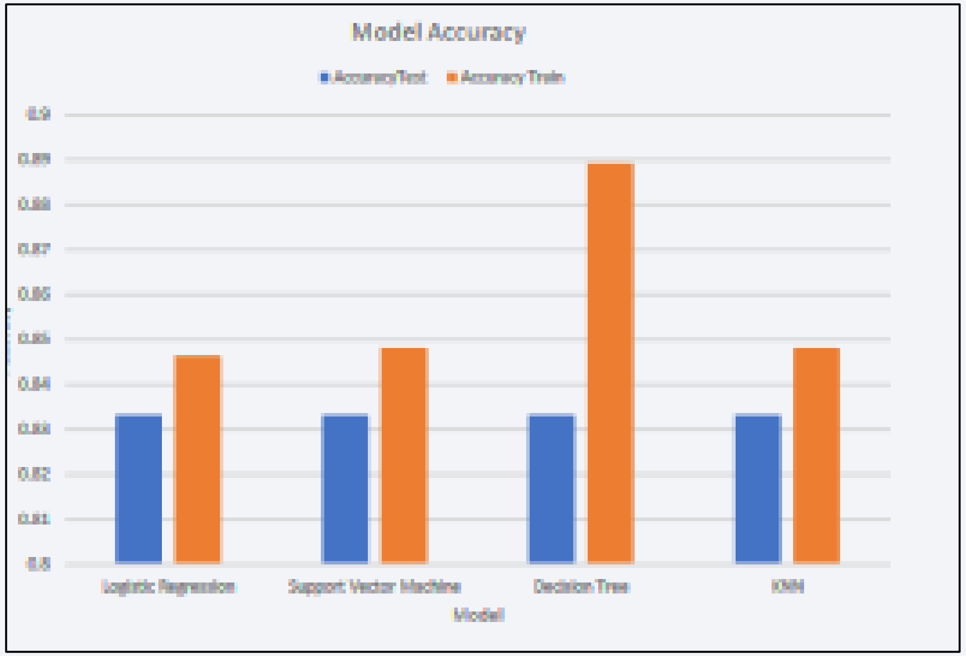


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

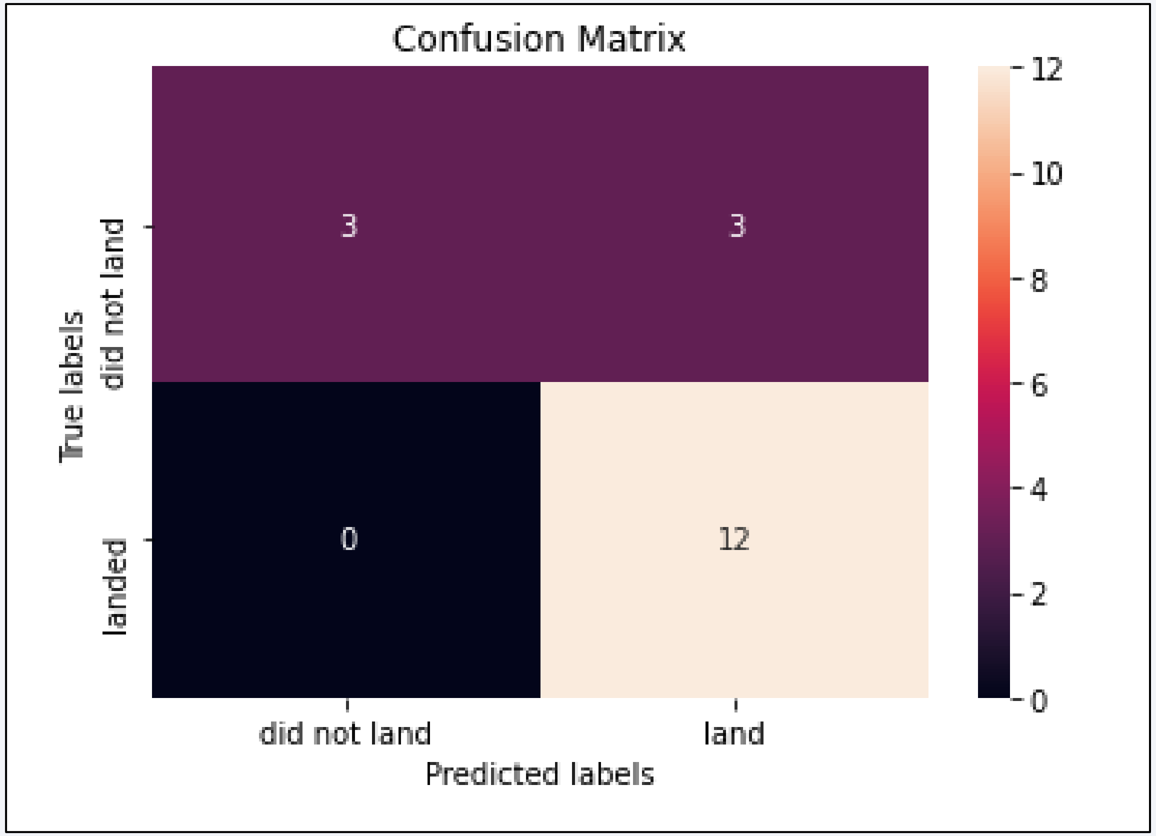
Predictive Analysis (Classification)

Classification Accuracy



- Decision Tree has the highest accuracy on training data (0.889)
- All models have the same accuracy on testing data (0.8333)

Confusion Matrix



The confusion matrix for the decision tree classifier indicates that the classifier can distinguish between different classes. However, it struggles with false positives, where unsuccessful landings are incorrectly marked as successful.

The metrics are as follows:
•**True Positives**: 12
•**False Positives**: 3
•**True Negatives**: 3
•**False Negatives**: 0
•**Accuracy**: 0.833

This suggests that while the classifier is generally accurate, it needs improvement in reducing false positives

Conclusions

- **Flight Volume and Success Rate**: The larger the number of flights at a launch site, the higher the success rate.
- **Increasing Success Rate**: Launch success rates have been rising since 2013, peaking around 2020.
- **Successful Orbits**: Orbits like ES-L1, GEO, HEO, SSO, and VLEO have the highest success rates.
- **Top Launch Site**: KSC LC-39A has the most successful launches among all sites.
- **Best Algorithm**: The Decision Tree Classifier is the best machine learning algorithm for predicting launch success, with an 84% accuracy rate.
- **Impact of Launch Site Location**: Certain launch sites show higher success rates, indicating location impacts success.
- **Payload and Success**: Heavier payloads are generally associated with a lower chance of successful landing.
- **Orbit Performance**: Orbits like ES-L1 and SSO have higher success rates, while SO orbits perform the worst.
- **Technological Improvements**: The rise in success rates from 2013 to 2020 reflects advancements in technology and procedures.

Appendix

Thank you!