IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Web Scraping for Data Collection
  - Data Acquisition via APIs
  - Data Cleaning and Preparation
  - Exploratory Data Analysis (EDA) with SQL
  - Data Visualization and Exploration
  - Building Interactive Visualizations with Folium
  - Machine Learning for Predictive Modeling
-
  - Summary of all results
  - Insights from Exploratory Data Analysis
  - Interactive Visual Analytics (Captured Screenshots)
  - Outcomes of Predictive Modeling

Introduction

- Project background and context:
  The primary objective of this project is to predict the successful landing of SpaceX's Falcon 9 first stage. SpaceX's competitive pricing for rocket launches—$62 million compared to over $165 million offered by other providers—is largely attributed to their innovative reuse of the first stage of their rockets. Accurately predicting the likelihood of a successful landing can provide valuable insights into the overall cost of a launch. This information is crucial for other companies aiming to compete with SpaceX in the space launch industry, as it can guide their strategies and cost assessments.

- Problems you want to find answers:

Model Accuracy and Performance:

- Can machine learning models reliably predict the success of Falcon 9 first-stage landings?
- Which machine learning model demonstrates the best predictive performance?

Trends and Patterns in Launch Success:

- What patterns or trends can be identified in SpaceX's launch success rates over time?

Impact of Environmental Factors:

- How do environmental variables, such as weather conditions or geographic location, influence the outcomes of landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data Sources: Data was gathered from the SpaceX REST API, providing detailed information on launches, rockets, and missions. Additional web scraping from reliable sources was performed to enrich and supplement the dataset.
- Perform data wrangling
  - Filtered out irrelevant entries to focus on meaningful data.
  - Addressed missing values using appropriate imputation techniques to ensure data completeness and consistency.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Use SQL to query data for trends, such as success rates and environmental impacts.
  - Create visualizations like scatter plots, bar charts, and heatmaps to uncover patterns and relationships.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Plotly Dash: Build dynamic dashboards for exploring trends and filtering data interactively.
- Perform predictive analysis using classification models
  - Building: Train models like Logistic Regression, Random Forests, or Gradient Boosting using split datasets.

Data Collection

- Describe how data sets were collected.

  The dataset was collected from the SpaceX API by requesting information on rocket launches, specifically Falcon 9 missions. The API response, in JSON format, was decoded using `.json()` and then normalized into a flat structure with `.json_normalize()` for easier analysis. The data was filtered to include only Falcon 9 launches, and missing values, such as payload mass, were replaced with the mean of available values. Finally, the cleaned data was exported to a CSV file for further analysis.

- You need to present your data collection process use key phrases and flowcharts

API Request from SpaceX API] =>

[Decode with .json()] =>

[Normalize Data (.json_normalize())] =>

[Filter for Falcon 9 Launches] =>

[Clean Data (Replace Missing Values)] =>

[Export to CSV]

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- NawafTaleb/IBM-Data-science-capstone

Task 2: Filter the dataframe to only include `Falcon 9` launches

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

In [82]:

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

Out[82]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- [IBM-Data-science-capstone/jupyter-labs-webscraping.ipynb at main · NawafTaleb/IBM-Data-science-capstone]

TASK 3: Create a data frame by parsing the launch HTML tables

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

In [13]:

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Data Wrangling

- Describe how data were processed

  We begin by determining the total number of launches at each site to assess launch frequency across different locations. Following that, we examine the distribution of orbit types used in the missions to understand the frequency of each type. We then analyze the success rates for each orbit type by evaluating the outcomes of the missions, identifying patterns or trends in the data. Finally, we create a landing outcome label derived from the "Outcome" column, classifying each mission as either a success or failure based on the first-stage landing result. This structured approach provides a comprehensive view of the launch data, allowing us to identify key factors influencing mission success and prepares the dataset for predictive modeling and further analysis.

- You need to present your data wrangling process using key phrases and flowcharts

- [IBM-Data-science-capstone/labs-jupyter-spacex-Data wrangling.ipynb at main · NawafTaleb/IBM-Data-science-capstone]()

In [8]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

Out[8]:
```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

`True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean. `True RTLS` means the mission outcome was successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad. `True ASDS` means the mission outcome was successfully landed to a drone ship `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

In [9]:
```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- The first chart is a bar chart that illustrates the success rates of different orbit types, helping to visualize how the outcome of missions varies across various orbital categories. This chart is useful for identifying which orbit types have the highest or lowest success rates, offering valuable insights into mission reliability. The second chart is a line chart that tracks the trend of launch success over time, allowing us to observe any patterns or shifts in success rates across different years. This helps in understanding how SpaceX's performance has evolved and whether improvements have been made over time. Additionally, these visualizations provide a clear, concise way to communicate key trends and relationships within the data, supporting the analysis and predictions for future launches.
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- IBM-Data-science-capstone/SpaceX_Machine Learning Prediction_5.ipynb at main · NawafTaleb/IBM-Data-science-capstone

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
    The SQL queries performed included calculating the total number of launches at each site to understand launch frequency, counting the frequency of each orbit type used in the missions to identify common orbits, and analyzing the mission outcomes for each orbit type to evaluate success rates. Additionally, a query was used to filter the dataset to include only Falcon 9 launches, ensuring a focused analysis. Queries were also run to check for missing values in key columns, such as payload mass, and calculate the mean payload mass to replace missing values. Lastly, a query created a new column to label each mission based on the first-stage landing outcome, categorizing them as successful or failed.

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
  [NawafTaleb/IBM-Data-science-capstone]()

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

```
%sql ibm_db_sa://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5-d84a-4
%sql SELECT Unique(LAUNCH_SITE) FROM SPACEXTBL;
```

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

```
* sqlite:///my_data1.db
Done.
```

Display average payload mass carried by booster versio

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
* sqlite:///my_data1.db
```

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects

  We plotted all the launch sites on a map using their latitude and longitude coordinates, providing a clear visual representation of the locations. Markers were added to indicate the launch sites, with different colors used to distinguish between successful and failed launches, allowing for a quick assessment of performance at each site. Additionally, we drew lines to represent the distances between the launch sites and nearby areas of interest, such as railways, highways, and coastlines. These visualizations helped highlight key insights: the sites' proximity to railways for transporting heavy components, easy access to highways for personnel, location near coastlines to minimize risks of debris falling on populated areas, and sufficient distance from cities to ensure safety in case of rocket failure. These elements were added to better understand the strategic placement and safety measures associated with each launch site.

- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

NawafTaleb/IBM-Data-science-capstone

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
  An interactive dashboard was built using Plotly Dash, featuring various visualizations. Pie charts were incorporated to display the total number of launches at each site, offering a clear overview of the launch distribution. A scatter plot was also included to examine the connection between launch outcomes and payload mass across different booster versions, helping to understand how payload size influences success rates. These interactive visualizations allow users to explore the data dynamically and uncover insights related to launch performance and booster types.
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
  NawafTaleb/IBM-Data-science-capstone

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
  The model development process began by selecting various classification algorithms, including Logistic Regression, Decision Trees, Random Forest, and Support Vector Machine (SVM), to predict landing success based on features such as launch site, orbit type, and payload mass. Each model was then evaluated using cross-validation and key performance metrics, including accuracy, precision, recall, and F1-score, to assess their predictive performance. To improve the models, hyperparameter tuning techniques like Grid Search and Randomized Search were employed to optimize each algorithm's settings. Finally, after evaluating the performance of all models, the one with the highest F1-score and balanced metrics was selected as the best-performing model.
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
  IBM-Data-science-capstone/SpaceX_Machine Learning Prediction_5.ipynb at main · NawafTaleb/IBM-Data-science-capstone

Results

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

Section 2

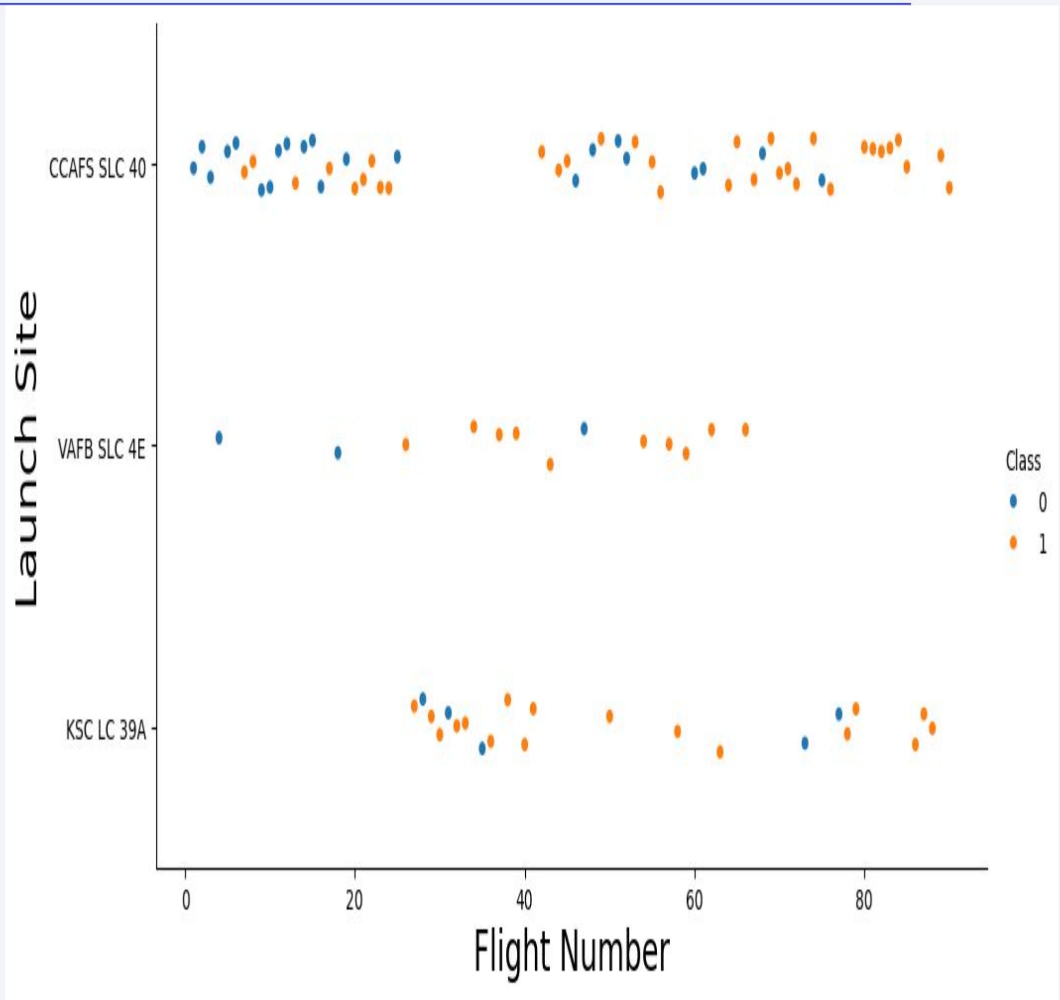
Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

The highest number of launches took place at CCAFS SLC 40, followed by KSC LC 39A, and then VAFB SLC 4E. A noticeable increase in successful launches is observed across all sites, especially after the initial failures.

- Show the screenshot of the scatter plot with explanations

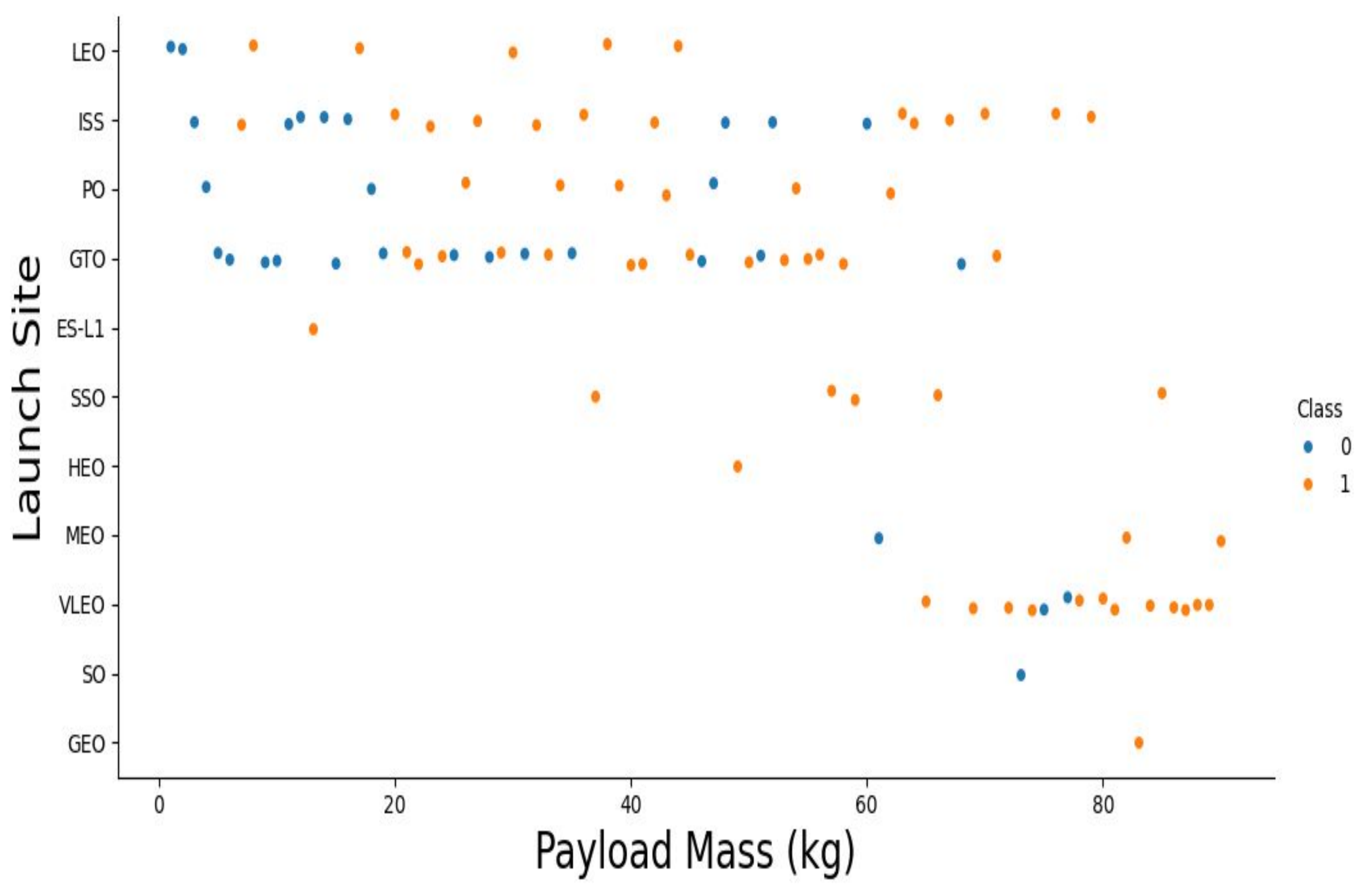


Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

  Payload masses between 4,000 kg and 6,000 kg have the highest failure rate.

- Show the screenshot of the scatter plot with explanations

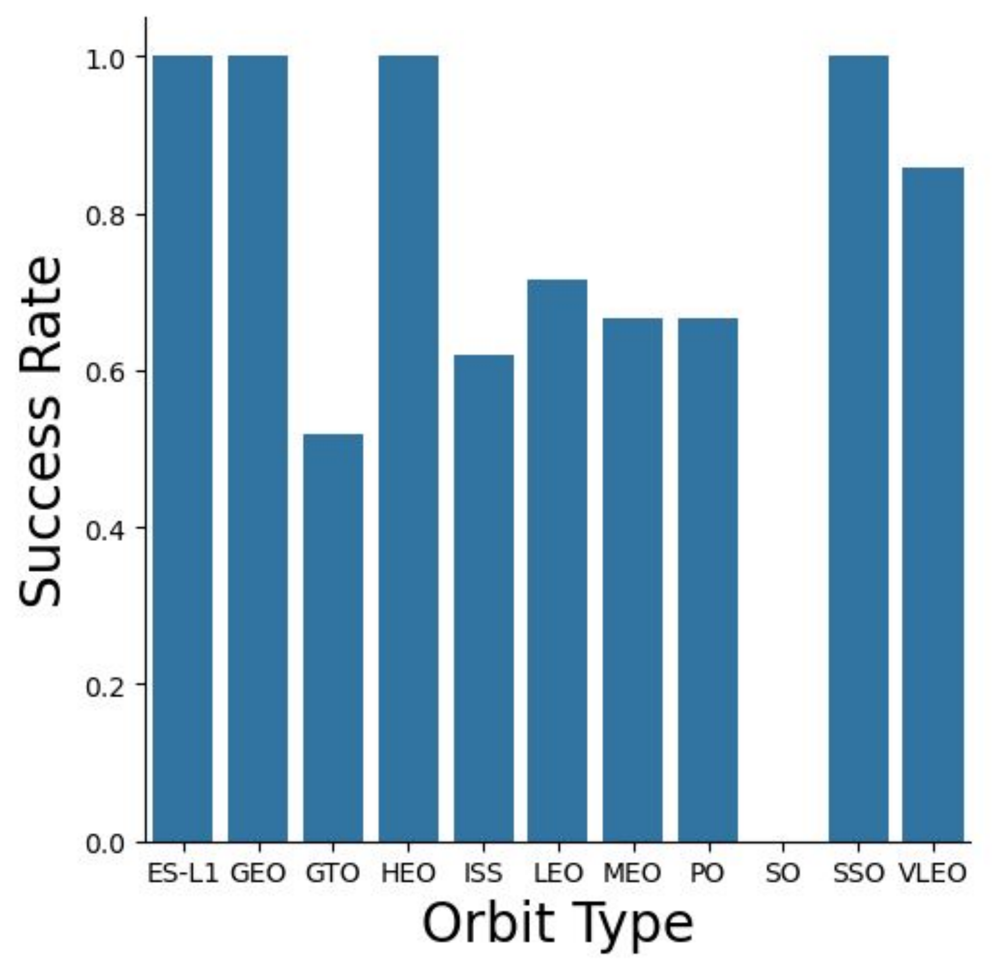


Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

There were no successful launches in the SO orbit. The top four orbits in the chart—ES-L1, GEO, HEO, and SSO—have the same success rate.

- Show the screenshot of the scatter plot with explanations

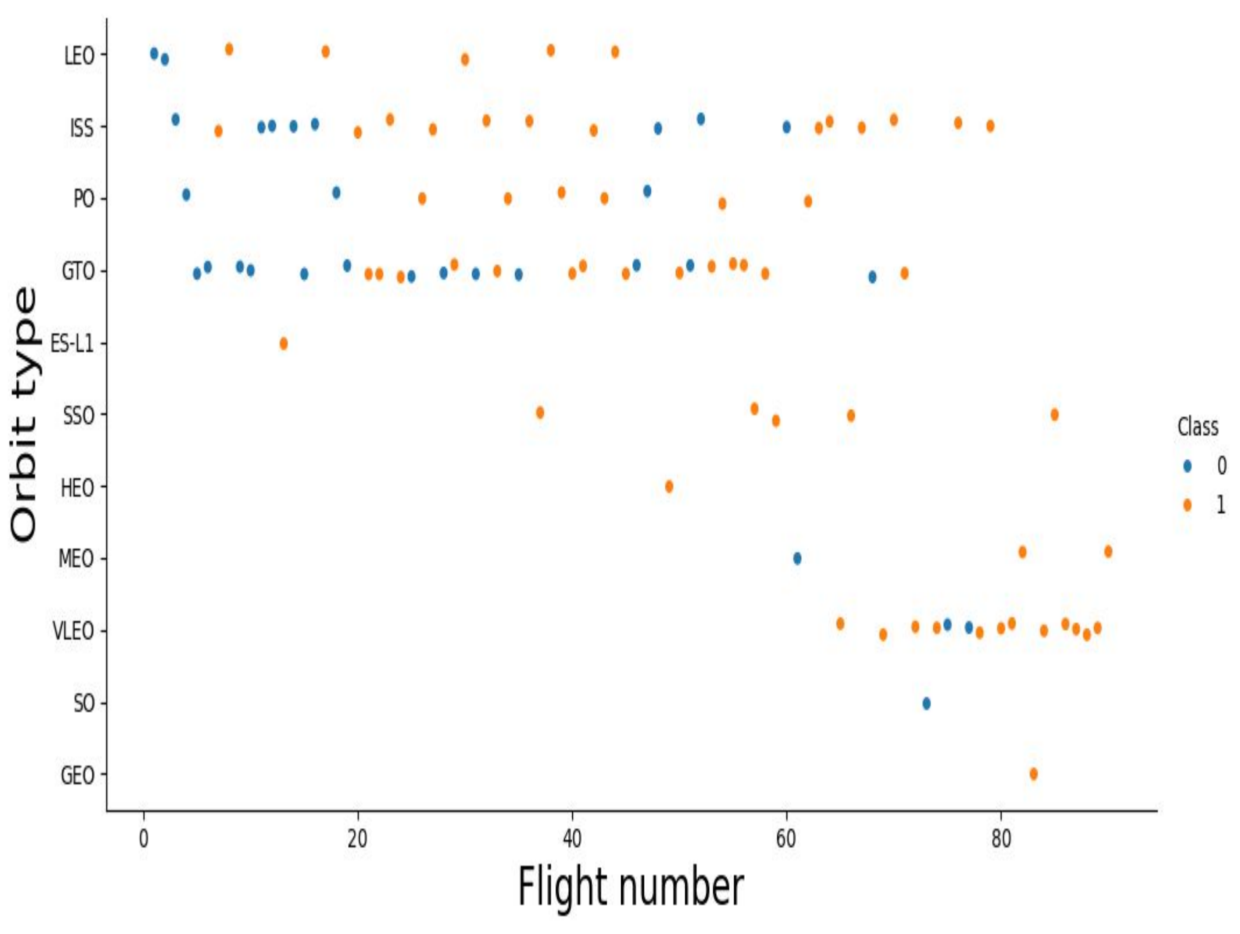


Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

Launches to the VLEO orbit have been on the rise in recent years. Meanwhile, the success rate for LEO orbit has improved as the number of flights increases.

- Show the screenshot of the scatter plot with explanations

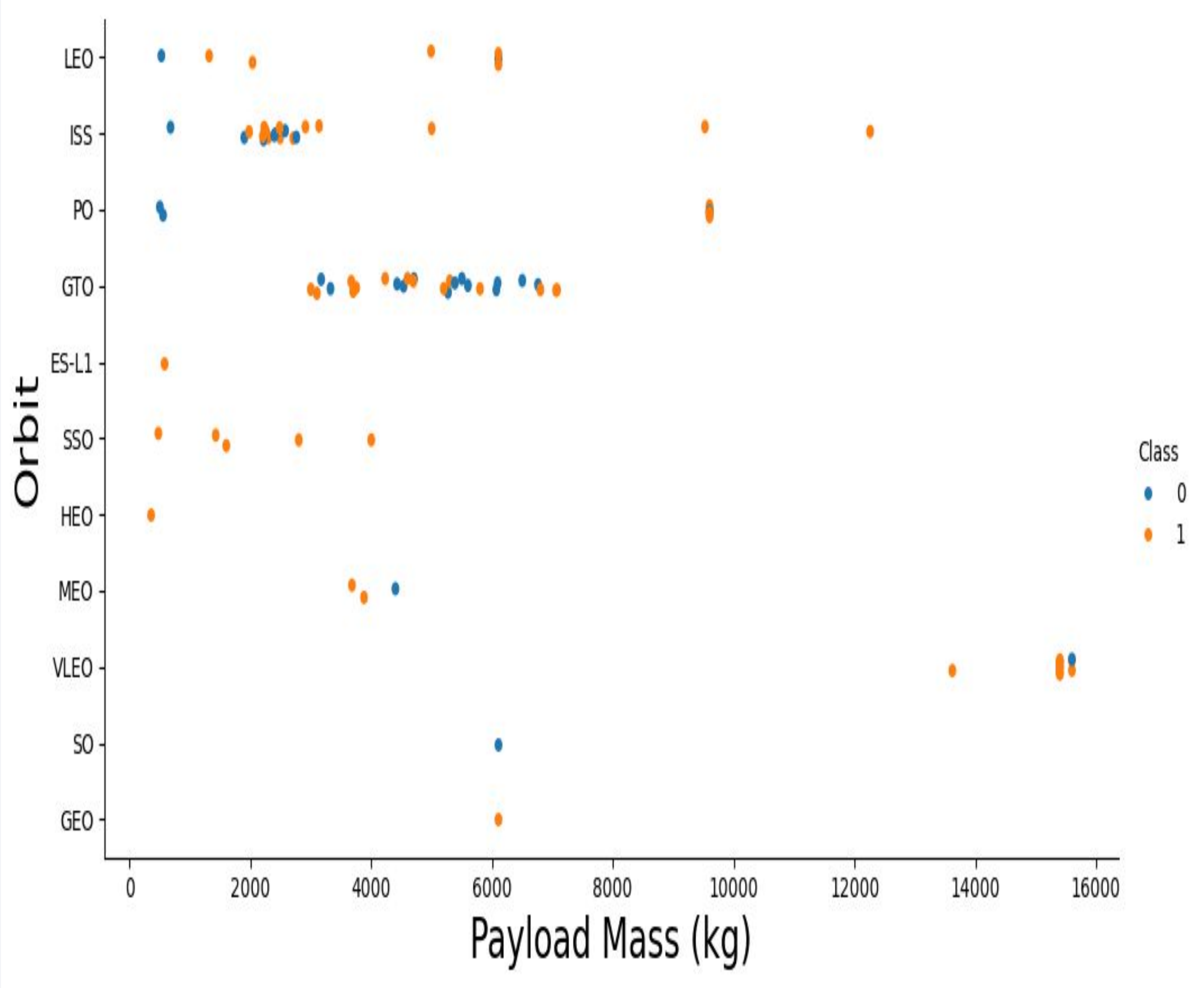


Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

Payload masses ranging from 3,000 to 8,000 kg have a notable impact on the GTO orbit. For heavier payloads, the success rate is higher in the ISS, LEO, and PO orbits.

- Show the screenshot of the scatter plot with explanations

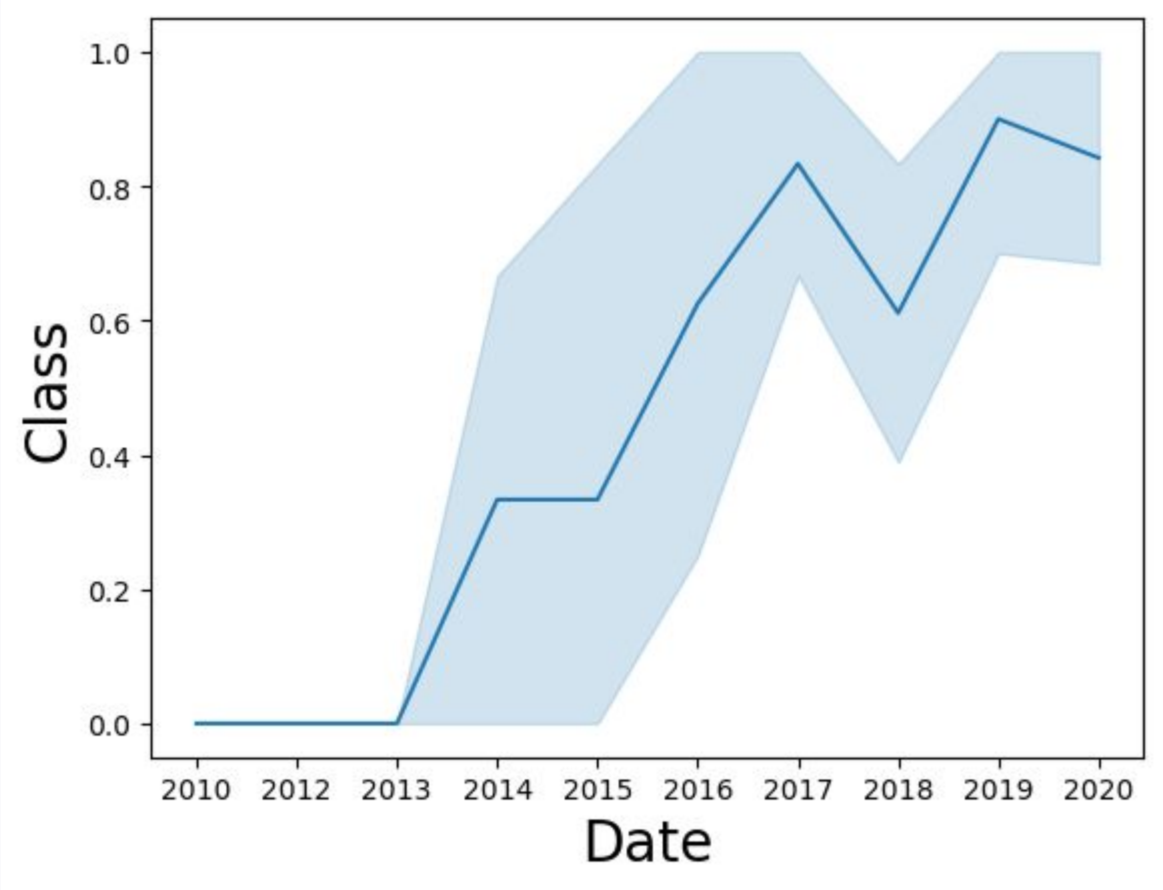


Launch Success Yearly Trend

- Show a line chart of yearly average success rate

The success rate has gradually increased from 2013 to 2020, indicating progress in launch technology, operational experience, and mission reliability. In the initial years (2013-2015), the success rate was relatively lower, with a higher occurrence of mission failures.

- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

[17]:
```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

[17]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcom
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachut
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachut
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attemp
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attemp
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attemp

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
  Using the SUM function on payload mass (in kg) provides the total sum of payload masses in the table.
- Present your query result with a short explanation here

```
[18]: %sql SELECT SUM(PAYLOAD_MASS__KG_) \
          FROM SPACEXTBL \
          WHERE CUSTOMER = 'NASA (CRS)';

 * sqlite:///my_data1.db
Done.
```

[18]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

By using the AVG function on payload mass (in kg), we obtain the average payload mass for Falcon 9 v1.1 from the data in the table.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';

 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

The first successful ground landing pad date (2015-12-22) can be retrieved using the MIN keyword, as it identifies the earliest date in the dataset.

- Present your query result with a short explanation here

```
%sql SELECT MIN(DATE) \
FROM SPACEXTBL \
WHERE LANDING_OUTCOME = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Present your query result with a short explanation here

```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING_OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

```
%%sql
SELECT substr("Date", 6, 2) AS "Month",
       "Landing_Outcome",
       "Booster_Version",
       "Launch_Site"
FROM SPACEXTABLE
WHERE substr("Date", 0, 5) = '2015'
AND "Landing_Outcome" LIKE '%Failure (drone ship)%';
```

```
 * sqlite:///my_data1.db
Done.
```

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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)) between the date 2010-06-04 and 2017-03-20, in descending order

- Present your query result with a short explanation here

```
%%sql
SELECT "Landing_Outcome", COUNT(*) AS "Count"
FROM SPACEXTABLE
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY COUNT(*) DESC;
```

```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

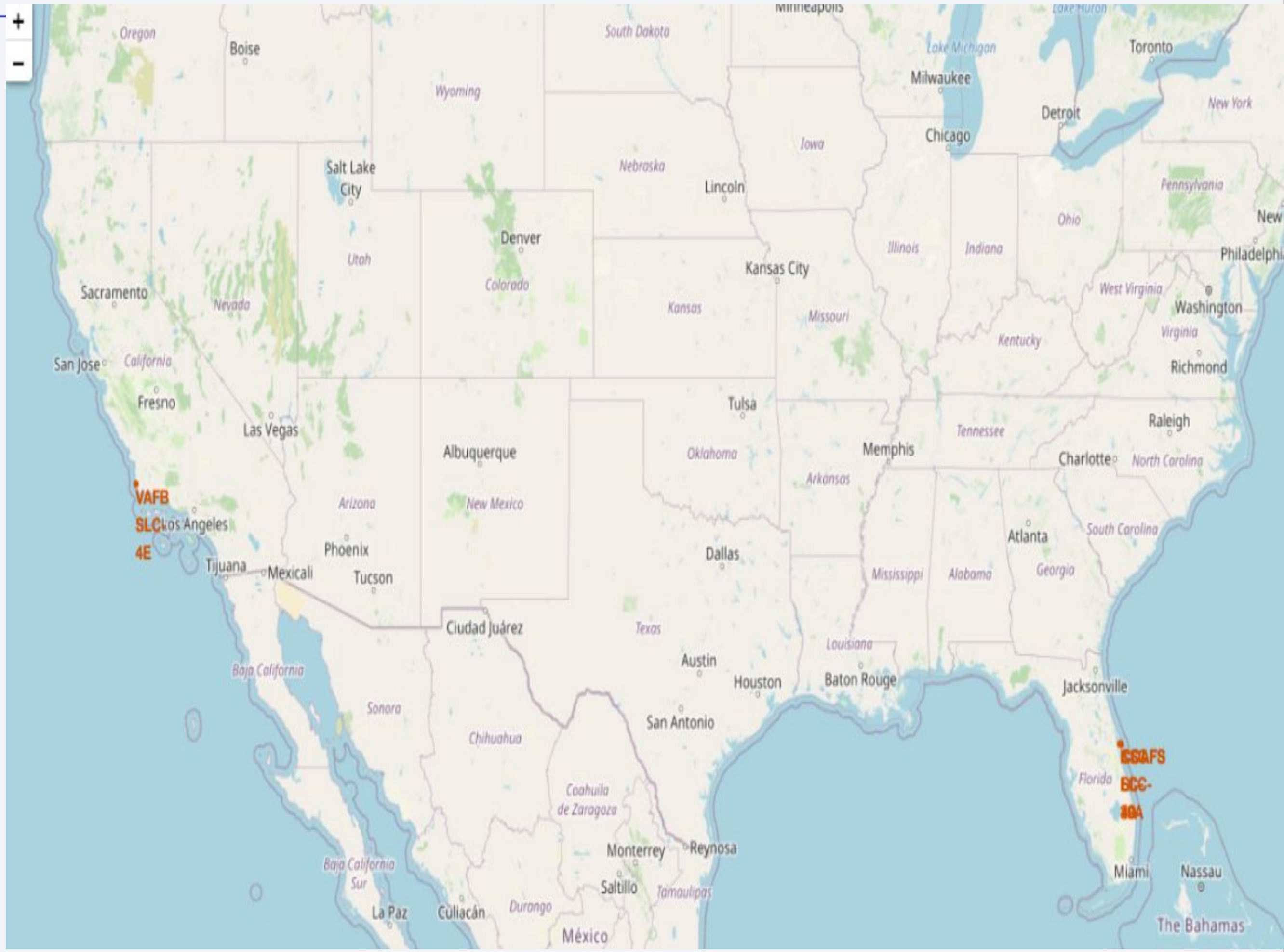
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

SpaceX launch sites are located along the coasts of California and Florida in the United States.



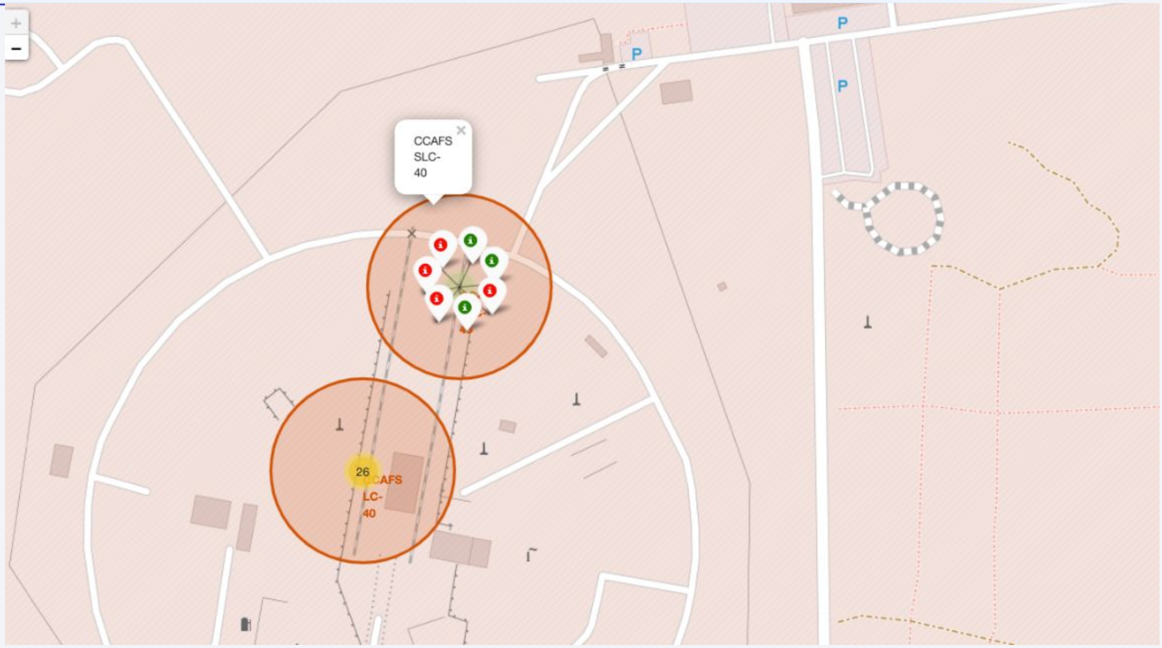
<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

- Explain the important elements and findings on the screenshot

Markers were placed on the map to represent SpaceX's launch sites in Florida and California, highlighting their geographic locations for easy visualization.



<Folium Map Screenshot 3>

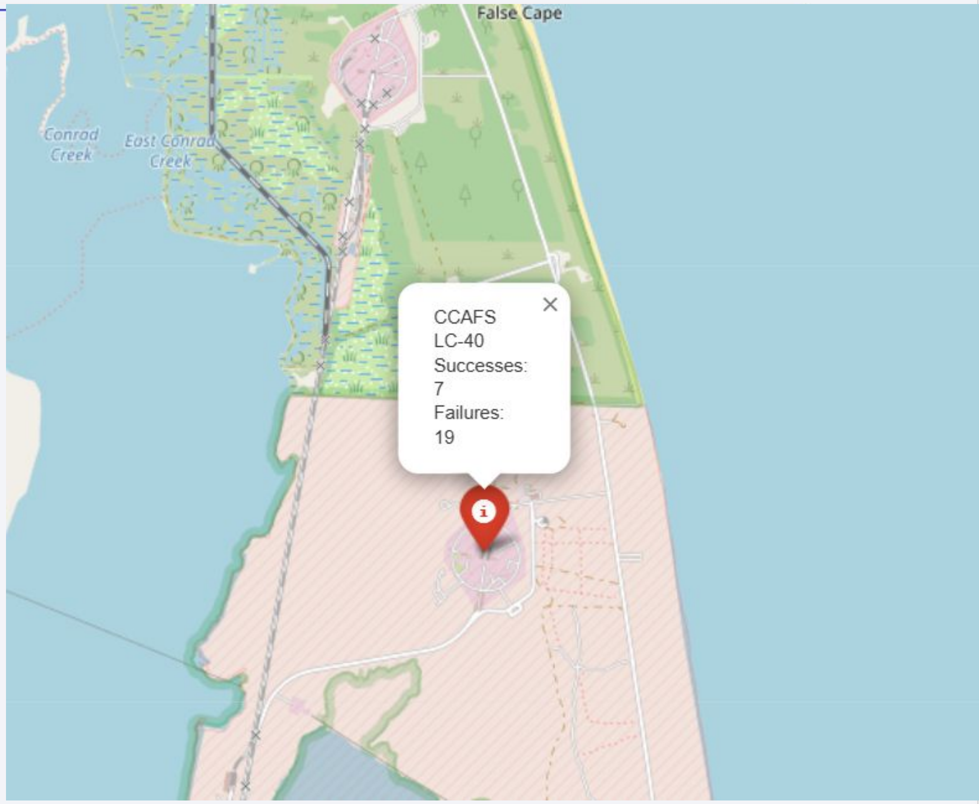
- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

The following distances are displayed between the launch sites and key landmarks:

- Distance to the nearest coastline
- Distance to the closest highway



Section 4

Build a Dashboard with Plotly Dash

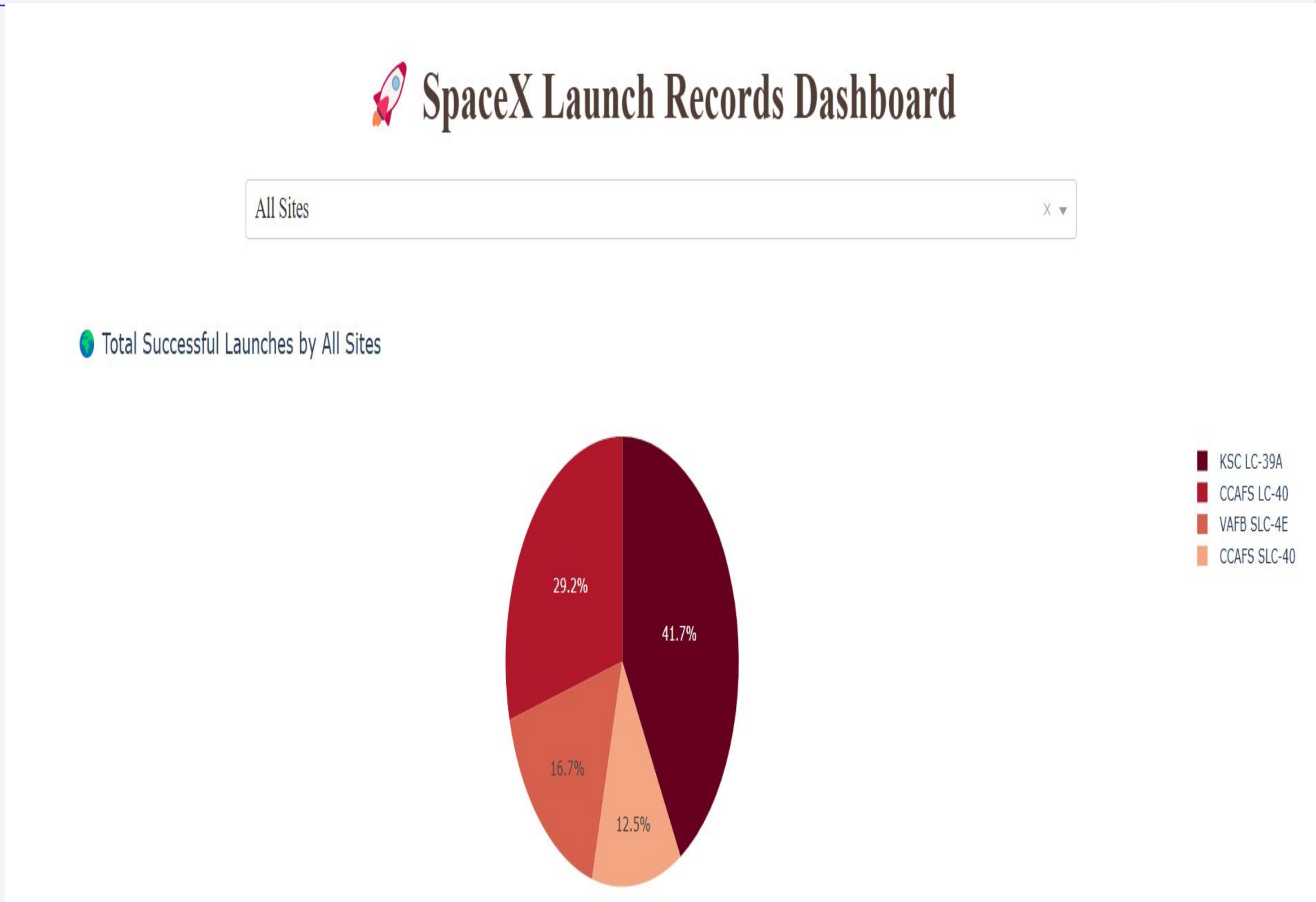
<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title

- Show the screenshot of launch success count for all sites, in a piechart

- Explain the important elements and findings on the screenshot:

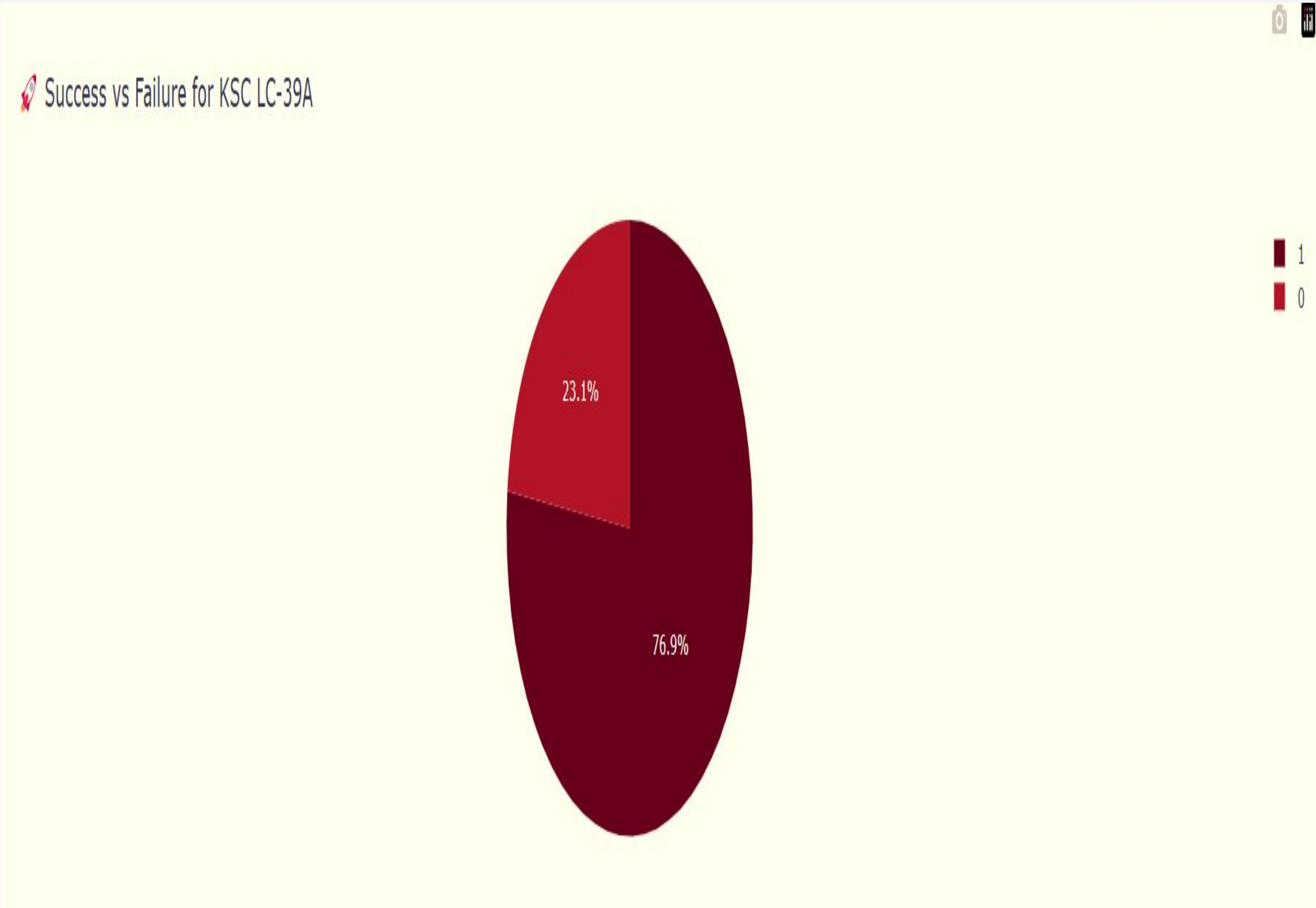
It is evident that KSC LC-39A has the highest success rate.



<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

KSC LC-39A boasts a success rate of 76.9%, with a failure rate of just 23.1%.



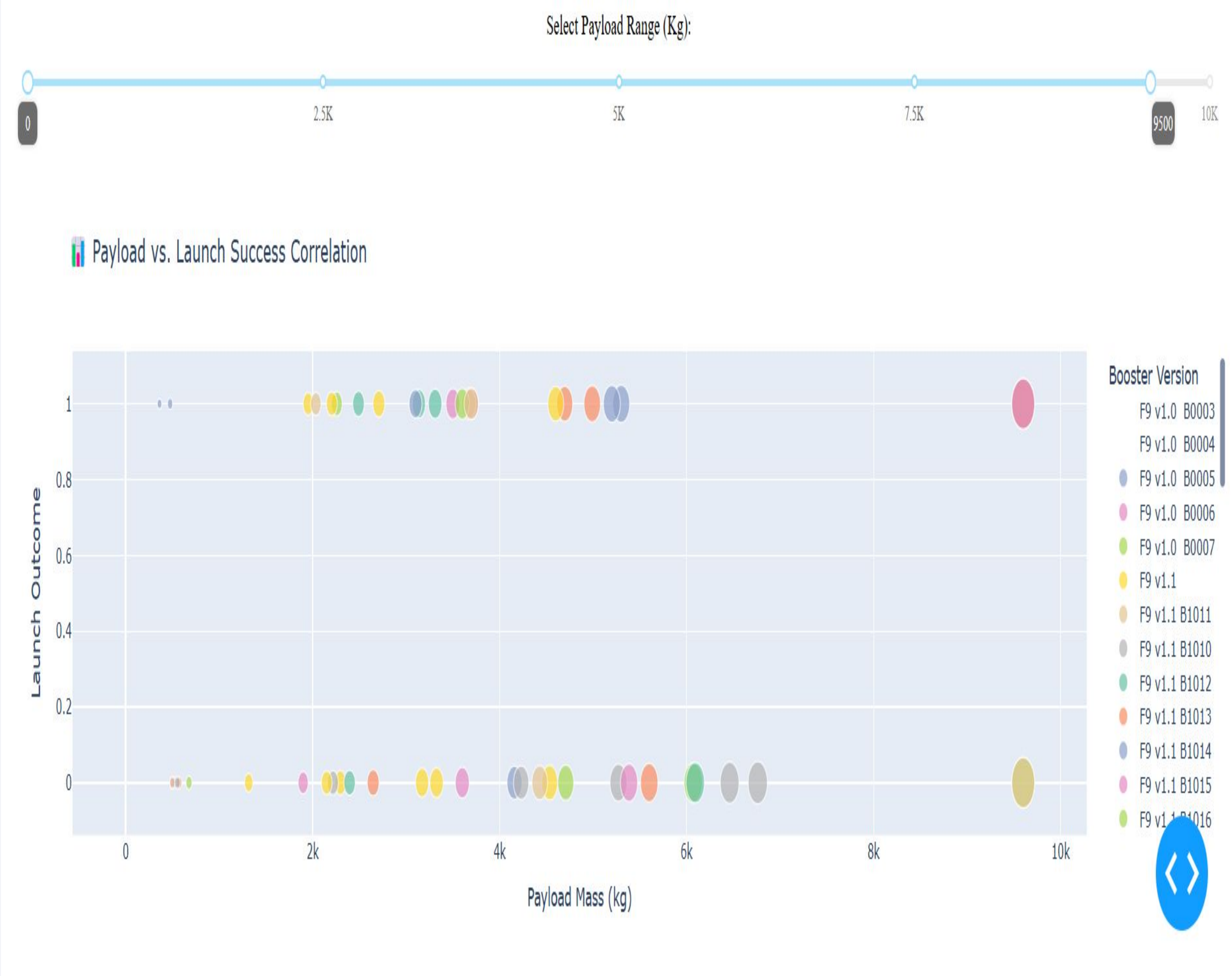
<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Lighter payloads, ranging from zero to five thousand kg, have a higher success rate compared to heavier payloads in the five thousand to ten thousand kg range.



Section 5

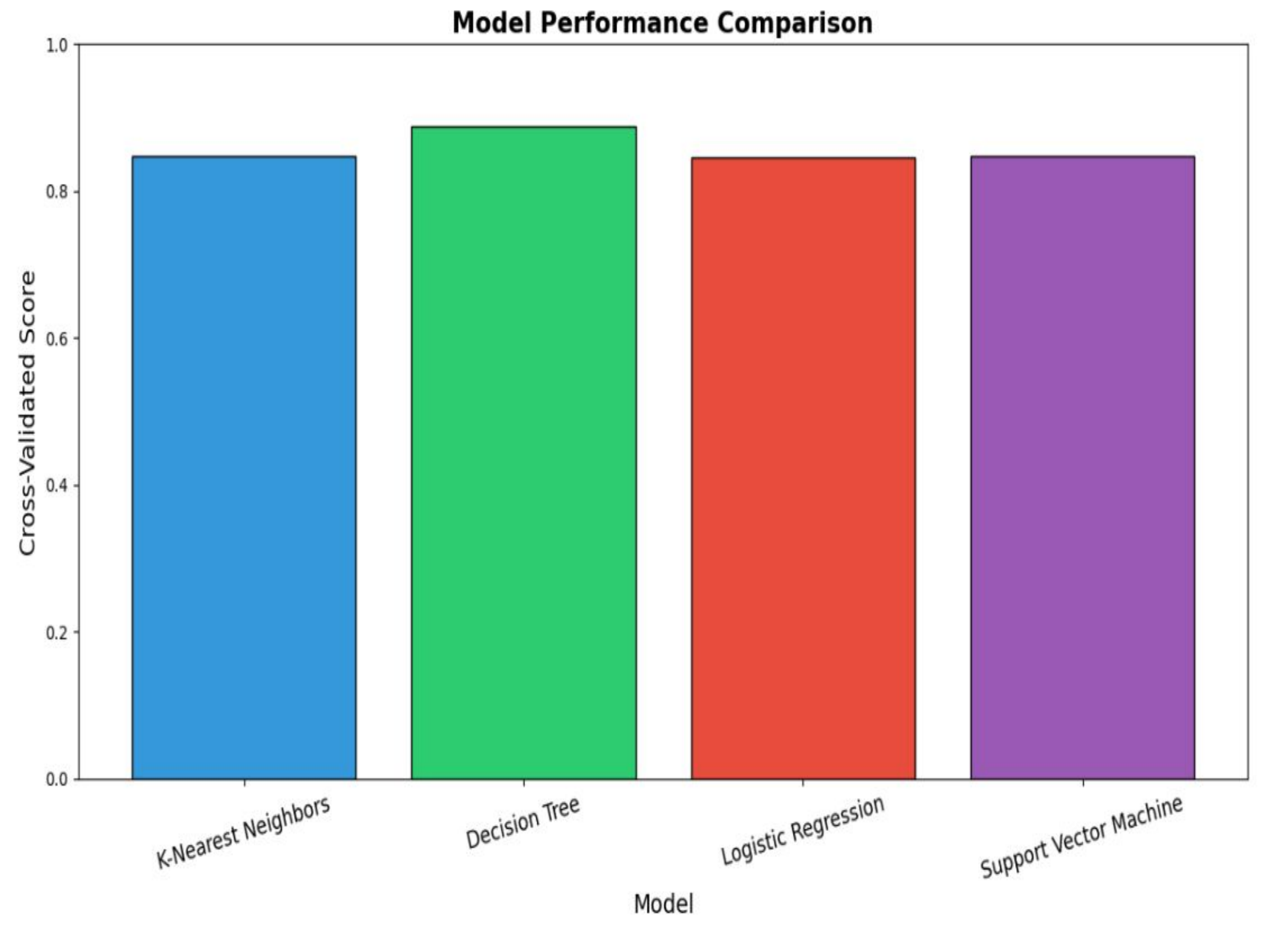
Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

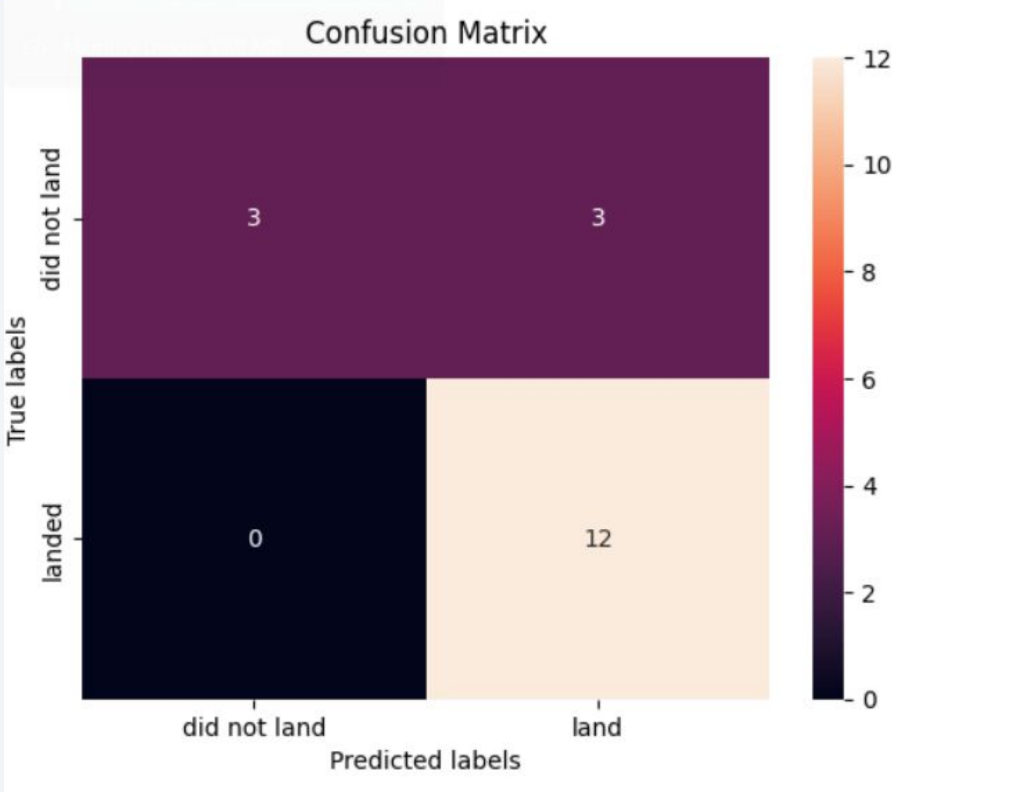
  The decision tree emerged as the best model, achieving an accuracy score of 0.88.



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

  The confusion matrix for the decision tree classifier demonstrates its ability to differentiate between classes, although it also reveals an issue with false positive values.



Conclusions

- The Decision Tree model is the best-performing, effectively distinguishing between classes, though it has an issue with false positives that requires attention. The launch success rate saw a significant increase starting in 2013, peaking in 2019, which reflects the advancements in SpaceX's capabilities and technologies. Ideally, launch sites should be located near coastlines, highways, and railroads while maintaining a safe distance from populated areas, ensuring both accessibility and safety during launches. KSC LC-39A stands out as the launch site with the highest success rate, demonstrating its efficiency and reliability in operations.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

[IBM-Data-science-capstone/SpaceX_Machine Learning Prediction_5.ipynb at main · NawafTaleb/IBM-Data-science-capstone]

Thank you!