IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mukhtyar khan
26th Jun 2023

Outline

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

Executive Summary

- **Summary of methodologies**
- **Data Collection**
  - **API**
    - Collect data from SpaceX's API.
  - **Web Scraping**
    - Collect data from third-party websites.
- **Data Wrangling**
  - Clean and format the data.
  - Identify and handle missing values.
  - Remove outliers.
- **Exploratory Data Analysis**
  - Use SQL to explore the data.
  - Create data visualizations to understand the data.
- **Interactive Visual Analytics**
  - Use Folium to create interactive visualizations.
- **Machine Learning Prediction**
  - Train a machine learning model to predict the success of a Falcon 9 rocket landing.

Executive Summary Continued!

- **Summary of all results**
- **Exploratory Data Analysis:**
  - Findings and insights from the EDA process
- **Interactive Analytics:**
  - Screenshots demonstrating interactive analytics using visualizations
- **Predictive Analytics:**
  - Results and outcomes from the predictive analytics model

Introduction

- Project background and context

SpaceX promotes Falcon 9 rocket launches on its website at a significantly lower cost of 62 million dollars, in contrast to other providers whose prices exceed 165 million dollars per launch. This cost advantage stems from SpaceX's ability to reuse the first stage of the rocket. To determine the cost of a launch, it is crucial to predict whether the first stage will successfully land. This predictive information becomes valuable when another company intends to compete with SpaceX in bidding for a rocket launch project. Therefore, the objective of this project is to develop a machine-learning pipeline capable of accurately predicting the successful landing of the first stage.

Introduction Continued!

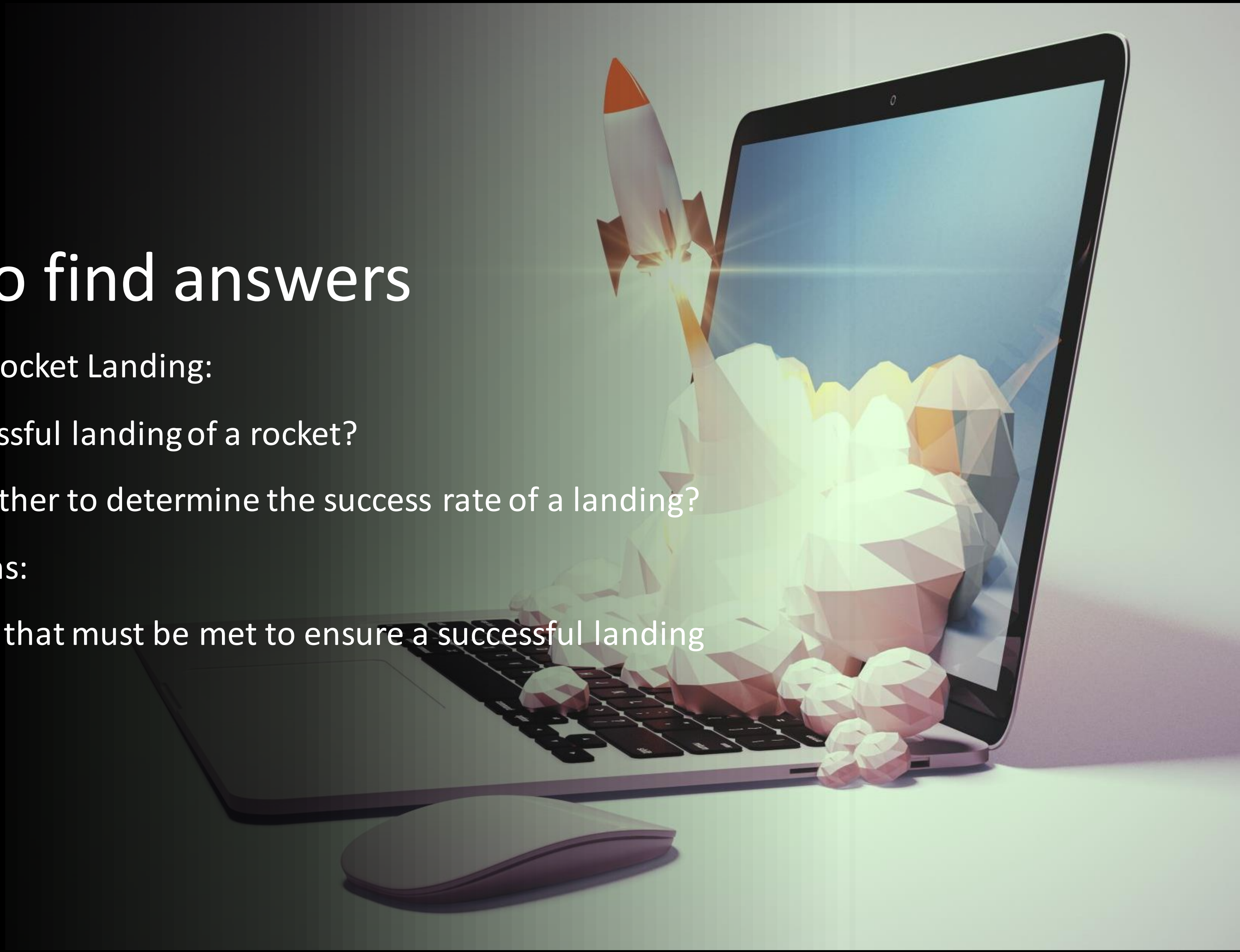
- Problems you want to find answers

1. Determining Factors for Successful Rocket Landing:

- Which specific factors influence the successful landing of a rocket?
- How do these factors interact with each other to determine the success rate of a landing?

2. Identifying Essential Operating Conditions:

- What are the critical operating conditions that must be met to ensure a successful landing program?



Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
- SpaceX API:
  - Data was collected using the SpaceX API, which provided relevant information about Falcon 9 rocket launches.
- Web Scraping:
  - Data was also collected through web scraping from Wikipedia to gather additional details.
- **Perform data wrangling**
  - Data Preprocessing:
  - The collected data underwent data wrangling processes to clean and transform it into a suitable format for analysis.
  - One-Hot Encoding:
  - Categorical features in the data were encoded using one-hot encoding to represent them numerically.

Methodology Continued!

Executive Summary

- **Exploratory Data Analysis (EDA):**
- **Visualization:**
- **EDA was performed using visualization techniques to gain insights and understand patterns and trends in the data.**
- **SQL Queries:**
- **SQL queries were utilized to explore the data and extract relevant information for analysis.**
- **Interactive Visual Analytics:**
- **Folium:**
- **Folium library was used to create interactive visualizations, allowing users to explore and interact with the data.**
- **Plotly Dash:**
- **Plotly Dash framework was employed to develop interactive dashboards for data visualization and exploration.**

Methodology Continued!

Executive Summary

- **Predictive Analysis:**
- **Classification Models:**
- **Predictive analysis was conducted using various classification models to predict the success of the first stage landing.**
- **Model Building, Tuning, and Evaluation:**
- **Classification models were built, tuned, and evaluated using appropriate techniques to optimize their performance and assess their predictive capabilities.**

Data Collection

- Data Collection Methods:
- SpaceX API:
- The data collection process involved making GET requests to the SpaceX API to retrieve relevant information.
- JSON Decoding and DataFrame Conversion:
- The response content obtained from the API was decoded as JSON using the .json() function. This JSON data was then transformed into a pandas DataFrame using .json_normalize().

Data Collection Continued!

- Data Collection Methods:
- Data Cleaning:
- After obtaining the DataFrame, we performed data cleaning operations, which included checking for missing values and filling them in where necessary to ensure data completeness and accuracy.
- Web Scraping from Wikipedia:
- Additionally, web scraping techniques were employed to gather Falcon 9 launch records from Wikipedia. This involved extracting the launch records as an HTML table and parsing it to convert it into a pandas DataFrame for further analysis.

Data Collection – SpaceX API

- **We utilized the SpaceX API to collect the necessary data for our analysis.**
- **The data collection process involved making GET requests to the SpaceX API.**
- **We cleaned and formatted the obtained data to ensure its quality and usability.**
- **Basic data wrangling techniques were applied to organize and structure the collected data.**
- **For more detailed information about our data collection process using the SpaceX API, please refer to the following notebook:**

In [6]:
```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [7]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [9]:
```
response = requests.get(spacex_url)
```

Check the content of the response

In [10]:
```
print(response.content)
```

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"laun

[Link to Notebook]

Data Collection – Web Scraping

- **To complement our data collection efforts, we utilized web scraping techniques to gather Falcon 9 launch records.**
- **We employed BeautifulSoup, a Python library, to scrape the necessary data from web pages.**
- **The scraped data was parsed and converted into a structured format, specifically a pandas dataframe.**
- **This dataframe served as a valuable resource for our subsequent analysis.**
- **For a more detailed explanation of our web scraping process, please refer to the following notebook:**
- **[Link to Notebook]()**

```
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from requests) (3.3)
```

In [17]:

```python
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

and we will provide some helper functions for you to process web scraped HTML table

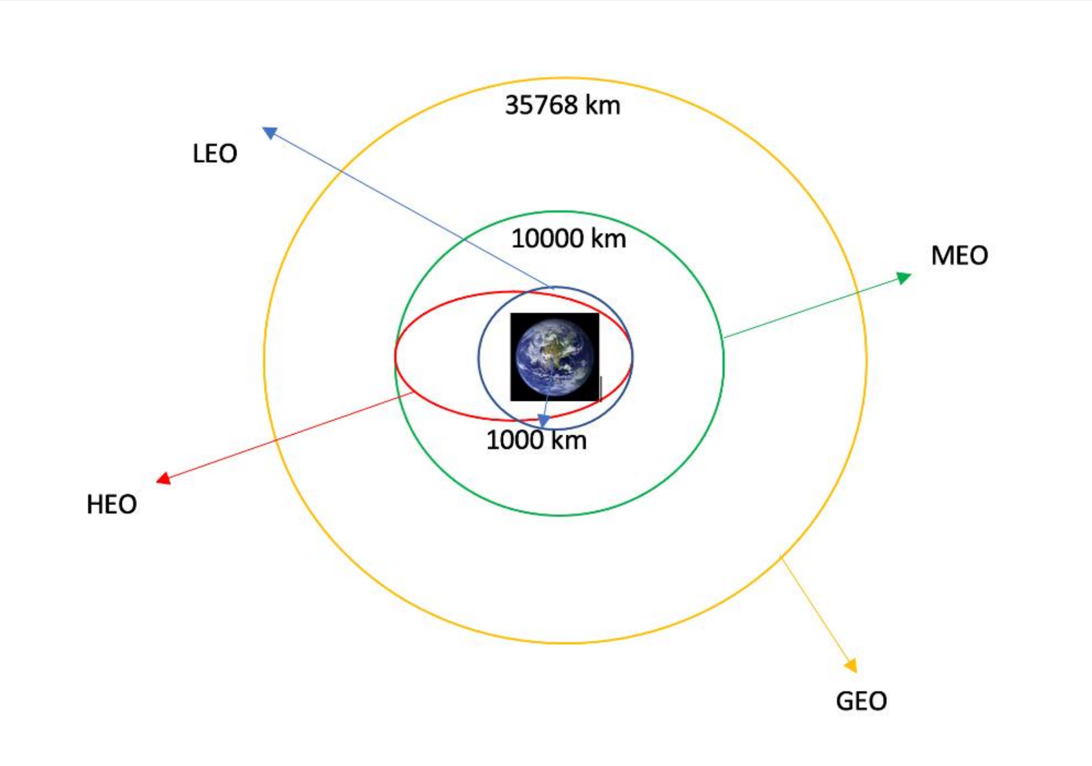
In [18]:

```python
def date_time(table_cells):
    """
    This function returns the data and time from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    return [data_time.strip() for data_time in list(table_cells.strings)][0:2]

def booster_version(table_cells):
    """
    This function returns the booster version from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=''.join([booster_version for i,booster_version in enumerate( table_cells.strings) if i%2==0][0:-1])
    return out

def landing_status(table_cells):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=[i for i in table_cells.strings][0]
    return out
```

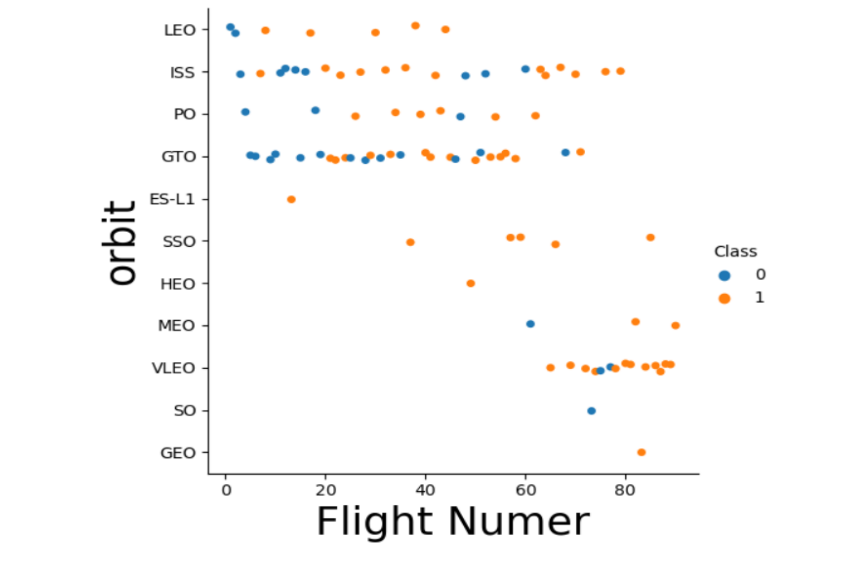
Data Wrangling

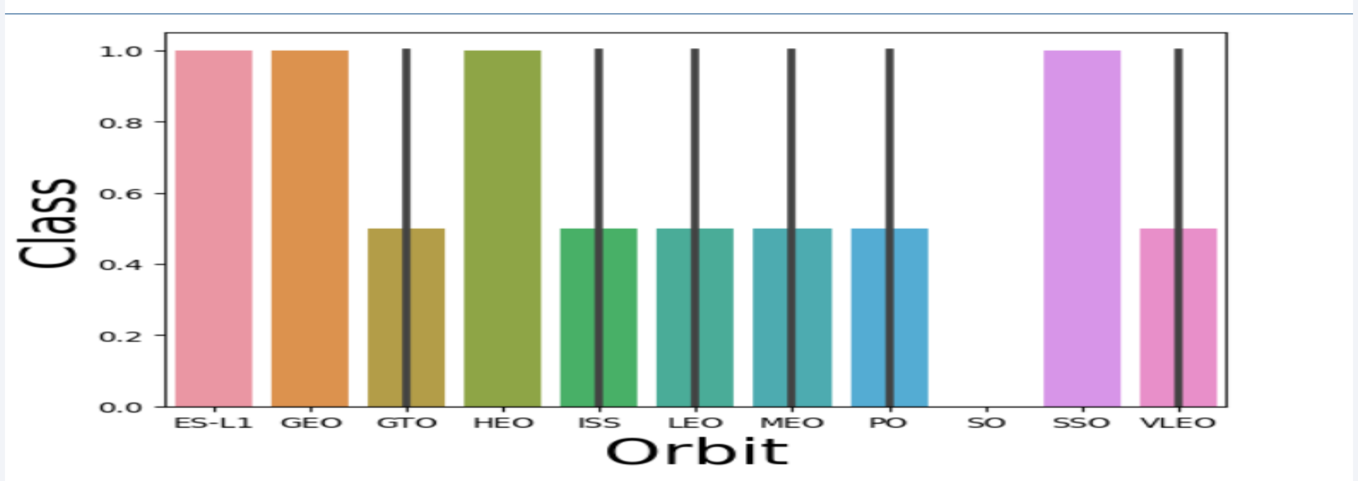


- As part of our data analysis process, we conducted exploratory data analysis (EDA) to gain insights into the dataset.
- We examined the number of launches at each site and analyzed the frequency of different orbits.
- Additionally, we created a landing outcome label based on the outcome column in the dataset.
- The results of our analysis, including the calculated statistics and labeled data, were exported to a CSV file for further use.
- For a more detailed overview of our data wrangling process, please refer to the following notebook:
- Link to Notebook

EDA with Data Visualization

- Utilized data visualization techniques to explore the dataset.
- Visualized relationships between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, and the yearly trend of launch success.
- Uncovered patterns, trends, and correlations through visual representations.





For more details and visualizations, refer to the provided notebook:

[Link to Notebook](#).

EDA with SQL

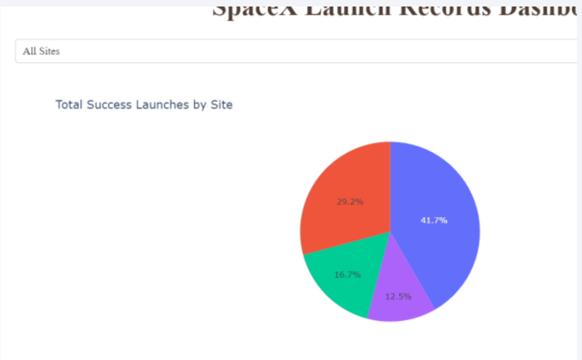
- We loaded the SpaceX dataset into a PostgreSQL database without leaving the Jupyter notebook.
- We applied EDA with SQL to get insight from the data.
- We wrote queries to find out:
    - The names of unique launch sites in the space mission.
    - The total payload mass carried by boosters launched by NASA (CRS).
    - The average payload mass carried by booster version F9 v1.1.
    - The total number of successful and failure mission outcomes.
    - The failed landing outcomes in drone ship, their booster version and launch site names.
- Link to notebook

Build an Interactive Map with Folium

- Created an interactive map using the Folium library.
- Marked all launch sites on the map and added map objects such as markers, circles, and lines to represent the success or failure of launches at each site.
- Assigned a binary class label (0 for failure and 1 for success) to the launch outcomes.
- Utilized color-labeled marker clusters to identify launch sites with higher success rates.
- Calculated the distances between each launch site and its surroundings.
- Explored questions such as proximity to railways, highways, and coastlines, as well as the distance between launch sites and cities.
- For a visual representation of the interactive map and detailed analysis, please refer to the provided notebook: Link to Notebook.

Build a Dashboard with Plotly Dash

- Built an interactive dashboard using Plotly Dash.
- Pie charts showcase total launches by different sites.
- Scatter graph reveals the relationship between Outcome and Payload Mass (Kg) for various booster versions.
- Link to the notebook: [GitHub Notebook]



Predictive Analysis (Classification)

- **Data Loading and Preprocessing**
  - Loaded the data using NumPy and Pandas.
  - Transformed the data to fit the machine learning models.
  - Split the data into training and testing sets.
- **Model Building and Hyperparameter Tuning**
  - Built different machine learning models.
  - Tuned different hyperparameters using GridSearchCV.
  - Used accuracy as the metric for the models.
  - Improved the models using feature engineering and algorithm tuning.
- **Model Evaluation**
  - Found the best performing classification model.
  - Link to the notebook: GitHub Notebook

Results

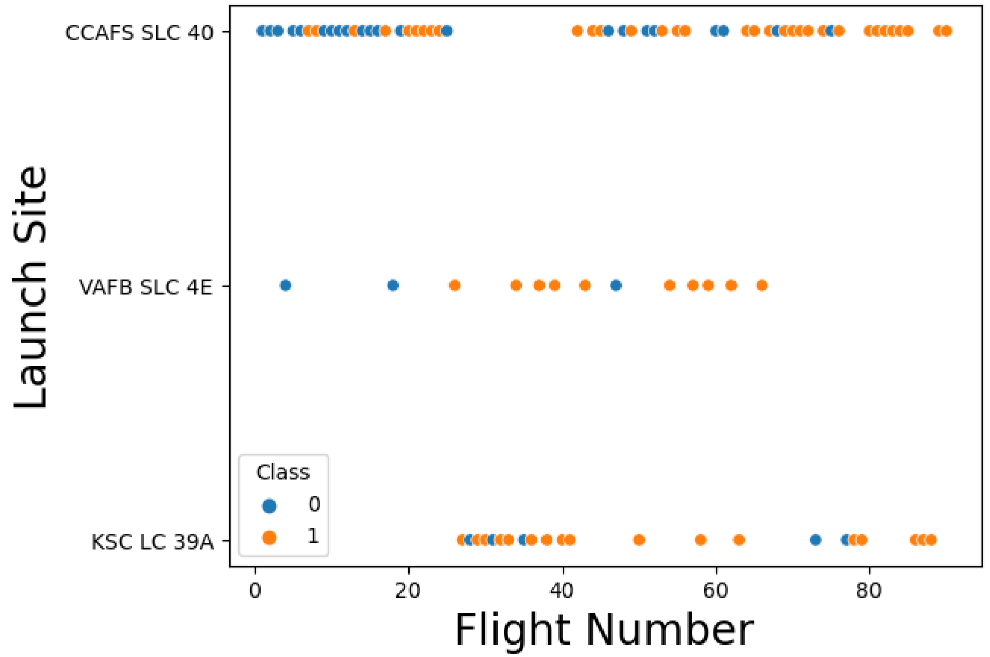
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

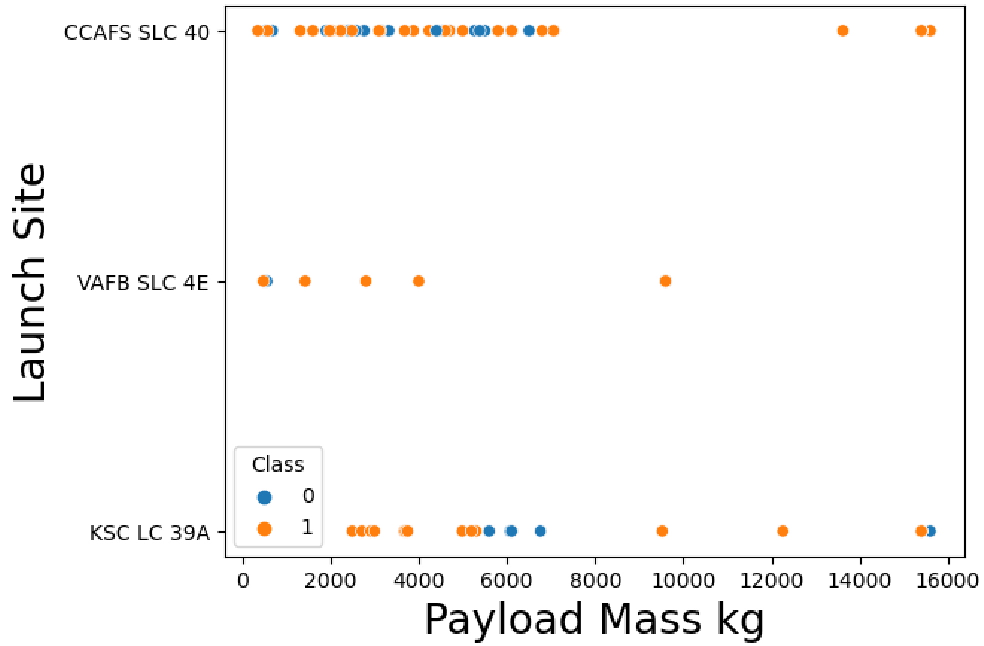
- We plotted the number of flights against the success rate for each launch site. The plot showed a positive correlation between the two variables, meaning that the larger the number of flights at a launch site, the greater the success rate at that launch site.
- The plot also showed some outliers. For example, the launch site at Vandenberg Air Force Base (VAFB) has a success rate of 94.1%, even though it has hosted only 27 launches. This suggests that VAFB may have a particularly favorable launch environment.



Payload vs. Launch Site

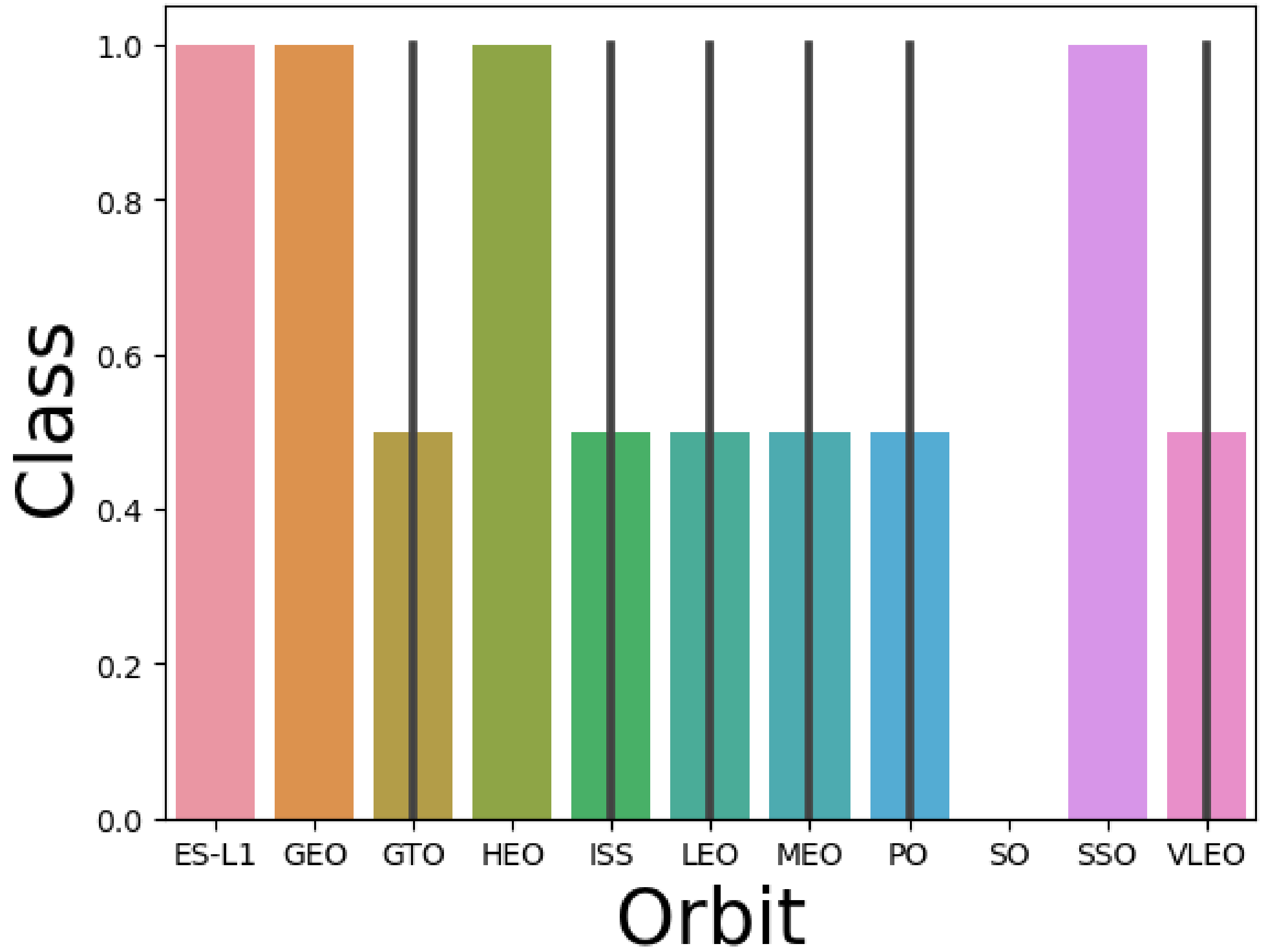
- Analyzing the Payload vs. Launch Site scatter point chart.

- Observation: No rockets launched for heavy payload mass (greater than 10,000 kg) at the VAFB-SLC launch site.
- Explanation:
- The scatter point chart illustrates the relationship between the Payload Mass and Launch Site.
- On examining the chart, it is evident that there are no instances of rockets launched at the VAFB-SLC launch site for heavy payload masses exceeding 10,000 kg.



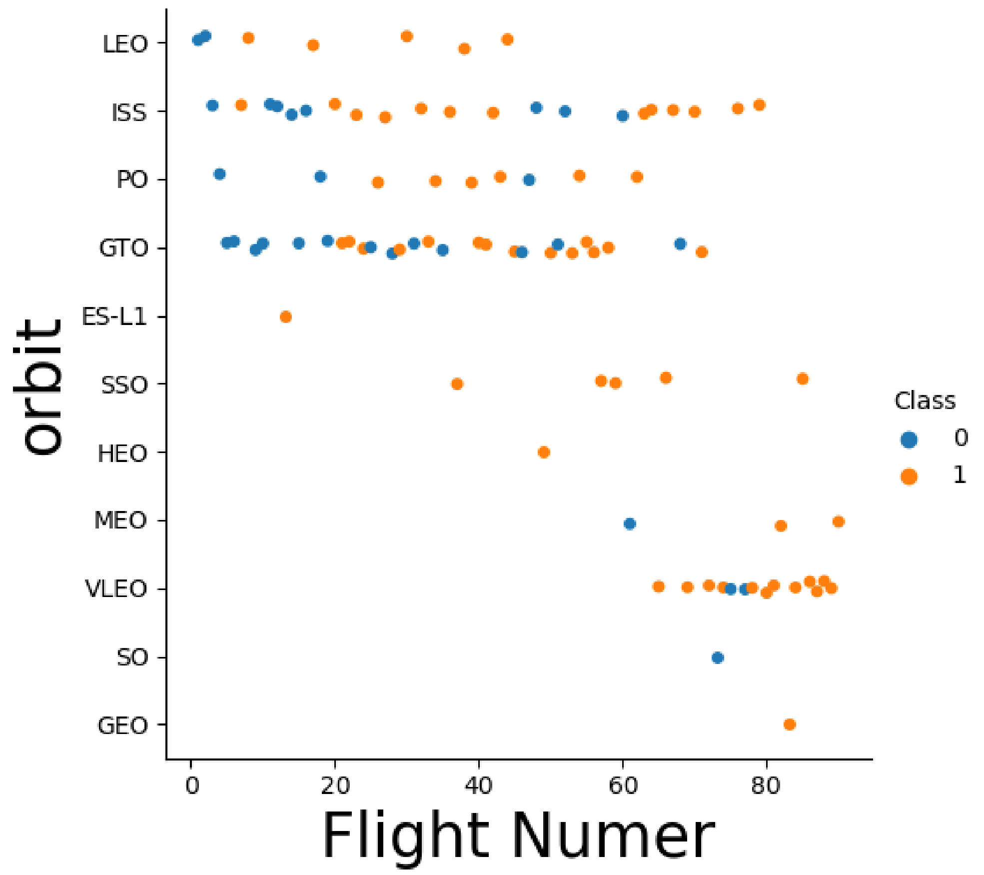
Success Rate vs. Orbit Type

- Observation: ES-L1, GEO, HEO, SSO, and VLEO orbits have the highest success rates.
- Insights:
- The success rates of these orbits indicate their reliability and effectiveness for successful launches.
- These orbits may have specific characteristics or requirements that contribute to their high success rates.



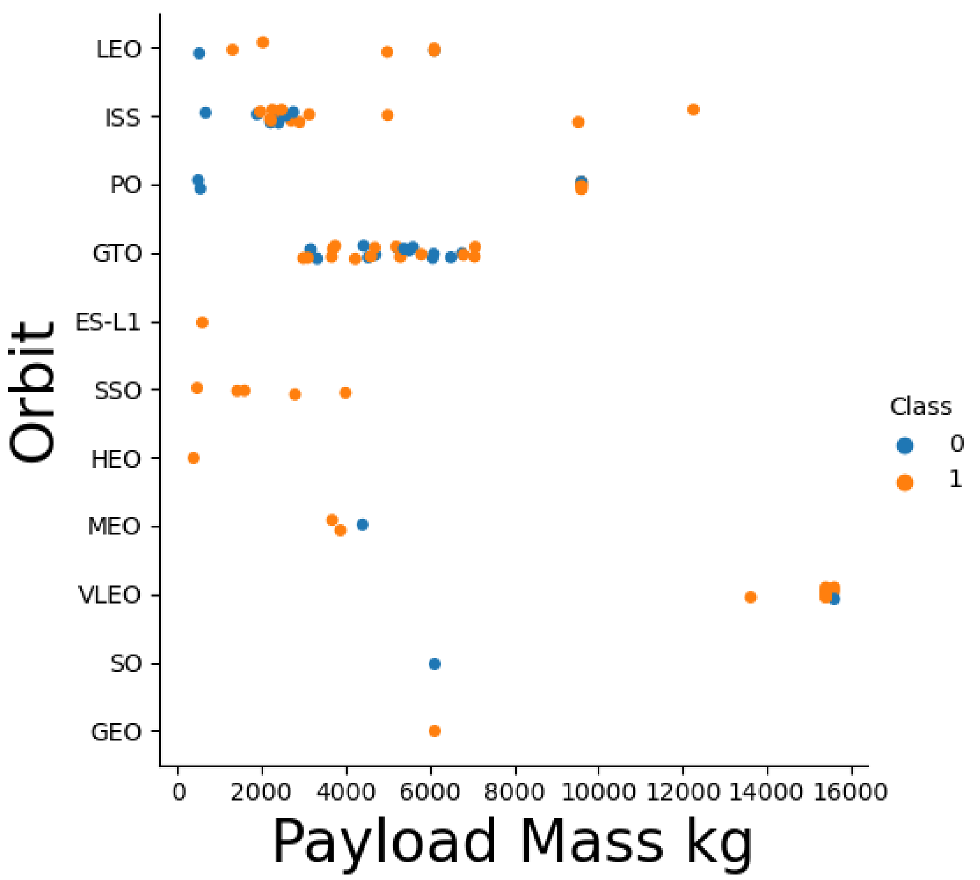
Flight Number vs. Orbit Type

- Observation: In the LEO orbit, the success appears to be related to the number of flights, while in the GTO orbit, there seems to be no relationship between flight number and success.
- Insights:
- LEO Orbit:
  - Higher flight numbers in LEO orbit indicate more experienced launches, leading to a higher success rate.
  - The relationship between flight number and success suggests that accumulated knowledge and experience contribute to improved success rates in LEO.
- GTO Orbit:
  - There is no clear relationship between flight number and success in the GTO orbit.
  - The success rate in GTO orbit might depend on other factors, such as payload



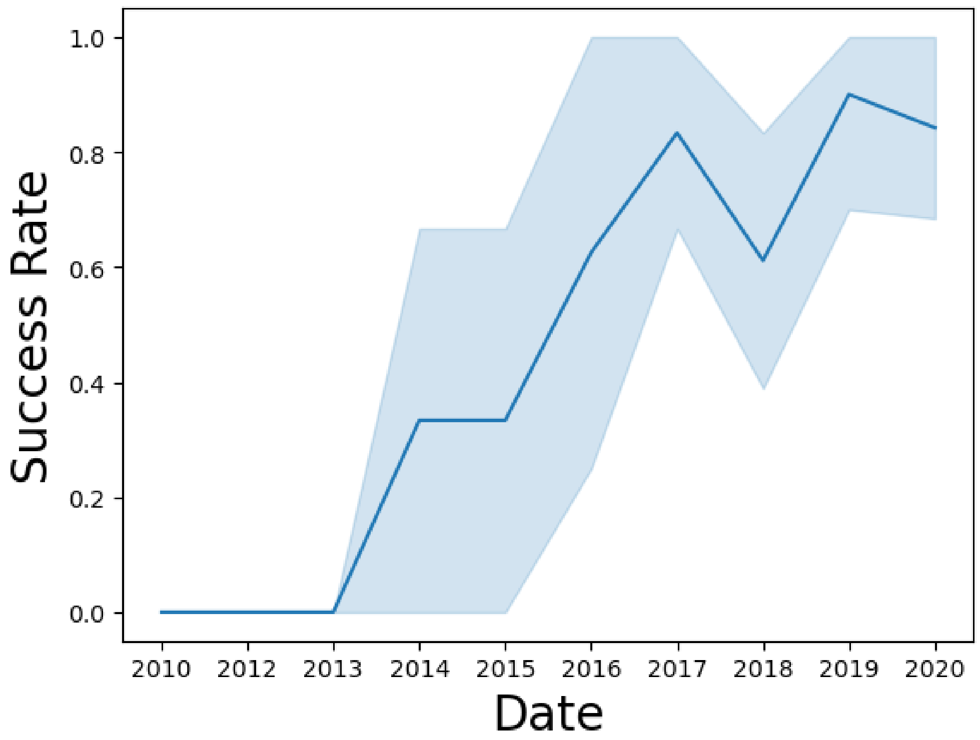
Payload vs. Orbit Type

- Observation: With heavy payloads, the successful landing rate is higher for Polar, LEO, and ISS orbits. However, distinguishing success and failure rates becomes challenging in the GTO orbit.
- Insights:
- Heavy Payloads:
  - Polar, LEO, and ISS orbits show a higher rate of successful landings for heavy payloads.
  - This indicates that these launch sites are more suitable for handling heavier payloads and ensuring successful landings.
- GTO Orbit:
  - In the GTO orbit, distinguishing between successful and unsuccessful missions becomes difficult for heavy payloads.
  - Both positive landing rates (successful missions) and negative landing rates (unsuccessful missions) are present in the GTO orbit, suggesting varying outcomes.



Launch Success Yearly Trend

- Observation: The success rate of SpaceX launches has shown a consistent increase from 2013 to 2020.
- Insights:
- Trend: Starting from 2013, the success rate of SpaceX launches has been on a positive trajectory.
- Steady Improvement: Over the years, the success rate has steadily increased, indicating the company's continuous efforts to improve their launch capabilities and mission outcomes.
- Reliability and Experience: The upward trend in success rate reflects SpaceX's growing experience, technological advancements, and enhanced launch procedures.



All Launch Site Names

- Explanation: To obtain the unique launch sites from the SpaceX data, we used the keyword "DISTINCT" in our query. This allowed us to retrieve only the unique values for the launch sites, eliminating any duplicates. The resulting list shows the distinct launch sites mentioned in the SpaceX data.

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

In [6]:
```
%sql  SELECT DISTINCT LAUNCH_SITE  FROM SPACEXTBL;
```

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

Out[6]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Explanation: The query utilized the SQL statement "SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5" to retrieve 5 records from the SpaceX data where the launch site starts with "CCA". The "LIKE" operator with the pattern 'CCA%' was used to match any launch site that begins with "CCA". The resulting records displayed the flight numbers, launch sites, class, payload mass, booster version, and booster version category for the identified launches at those specific CCA launch sites.

In [7]:
```
%sql SELECT * from SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[7]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-	SpaceX	500.0	LEO	NASA	Success	No att

Total Payload Mass

- Explanation: The query "SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA(CRS)';" attempted to calculate the sum of payload mass (in kilograms) carried by boosters from NASA. However, the result returned "None", indicating that there were no records matching the condition where the customer is specifically "NASA(CRS)". This could mean that there were no recorded payload masses for boosters associated with NASA in the provided dataset.

Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [8]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA(CRS)';

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

Out[8]:

SUM(PAYLOAD_MASS__KG_)
None

Average Payload Mass by F9 v1.1

- Explanation: The query "SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';" calculates the average payload mass (in kilograms) carried by boosters of the F9 v1.1 version. The result, 2928.4 kilograms, represents the average payload mass for this specific booster version. This information provides insight into the typical payload capacity of boosters belonging to the F9 v1.1 version.

Task 4

Display average payload mass carried by booster version F9 v1.1

In [9]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

Out[9]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Explanation: The query "SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';" retrieves the minimum date from the SPACEXTBL table where the landing outcome is recorded as a successful landing on a ground pad. The result, January 8, 2018, represents the date when the first successful landing on a ground pad occurred according to the available data. This information provides insight into the historical timeline of SpaceX's achievements in successfully landing boosters on ground pads.

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [12]:
```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Out[12]:

min(DATE)
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- Explanation: The query "SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;" was executed to retrieve the names of boosters that meet the specified conditions. However, the query did not return any results, indicating that there are no records in the dataset that satisfy the given criteria. This suggests that there are no instances where a booster successfully landed on a drone ship with a payload mass within the specified range.

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [13]:

```
%sql select Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (ground pad)' and PAYLOAD_MASS__KG_>400 and PAY
```

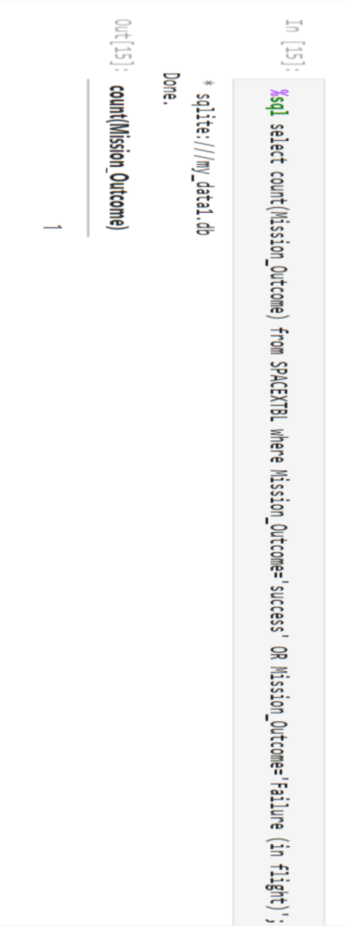
```
 * sqlite:///my_data1.db
Done.
```

Out[13]:

Booster_Version

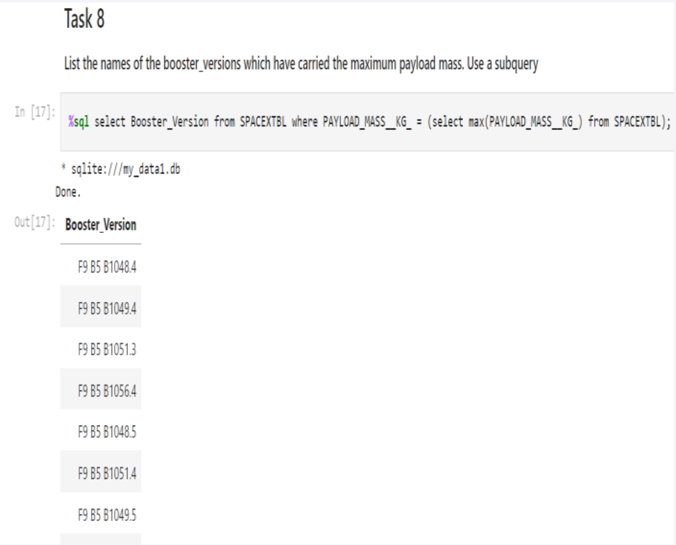
Total Number of Successful and Failure Mission Outcomes

- Explanation: The query "SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome='success' OR Mission_Outcome='Failure (in flight)';" was executed to count the occurrences of successful and failure mission outcomes in the SPACEXTBL table. The condition Mission_Outcome='success' OR Mission_Outcome='Failure (in flight)' filters the rows where the mission outcome is either "success" or "Failure (in flight)". The query returned the count of such occurrences, which is 1 in this case.



Boosters Carried Maximum Payload

- Explanation: The query "SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);" was executed using a subquery to retrieve the names of booster versions that have carried the maximum payload mass. The subquery (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) finds the maximum payload mass from the SPACEXTBL table, and the outer query selects the booster versions where the payload mass matches the maximum value. The query returned the names of booster versions that have achieved the maximum payload mass.


Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:
```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

Out[17]:

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

2015 Launch Records

- Explanation: The query "SELECT substr(Date, 4, 2) AS month, booster_version, launch_site FROM SPACEXTBL WHERE substr(Date, 7, 4) = '2015';" was executed to retrieve the month, booster version, and launch site for failed landing outcomes on drone ship in the year 2015. The function substr() was used to extract the month from the Date column. The condition substr(Date, 7, 4) = '2015' filters the rows where the year is 2015. The query returned the month, booster version, and launch site information for the matching records.

In [23]:
```
%sql SELECT substr(Date, 4, 2) AS month, booster_version, launch_site FROM SPACEXTBL WHERE substr(Date, 7,
```
```
 * sqlite:///my_data1.db
Done.
```
Out[23]:

month	Booster_Version	Launch_Site
10	F9 v1.1 B1012	CCAFS LC-40
11	F9 v1.1 B1013	CCAFS LC-40
02	F9 v1.1 B1014	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40
04	F9 v1.1 B1016	CCAFS LC-40
06	F9 v1.1 B1018	CCAFS LC-40
12	F9 FT B1019	CCAFS LC-40

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

- Explanation: The query "SELECT * FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)' AND Date BETWEEN '2010-06-04' AND '2017-03-20';" was executed to retrieve the records from the SPACEXTBL table where the Landing_Outcome is 'Success (ground pad)' and the Date is between '2010-06-04' and '2017-03-20'. The query returned the Flight_Number, Date, Booster_Version, Launch_Site, and other columns for the matching records. This query helps identify successful landings on ground pads within the specified date range.

Section 3

Launch Sites Proximities Analysis

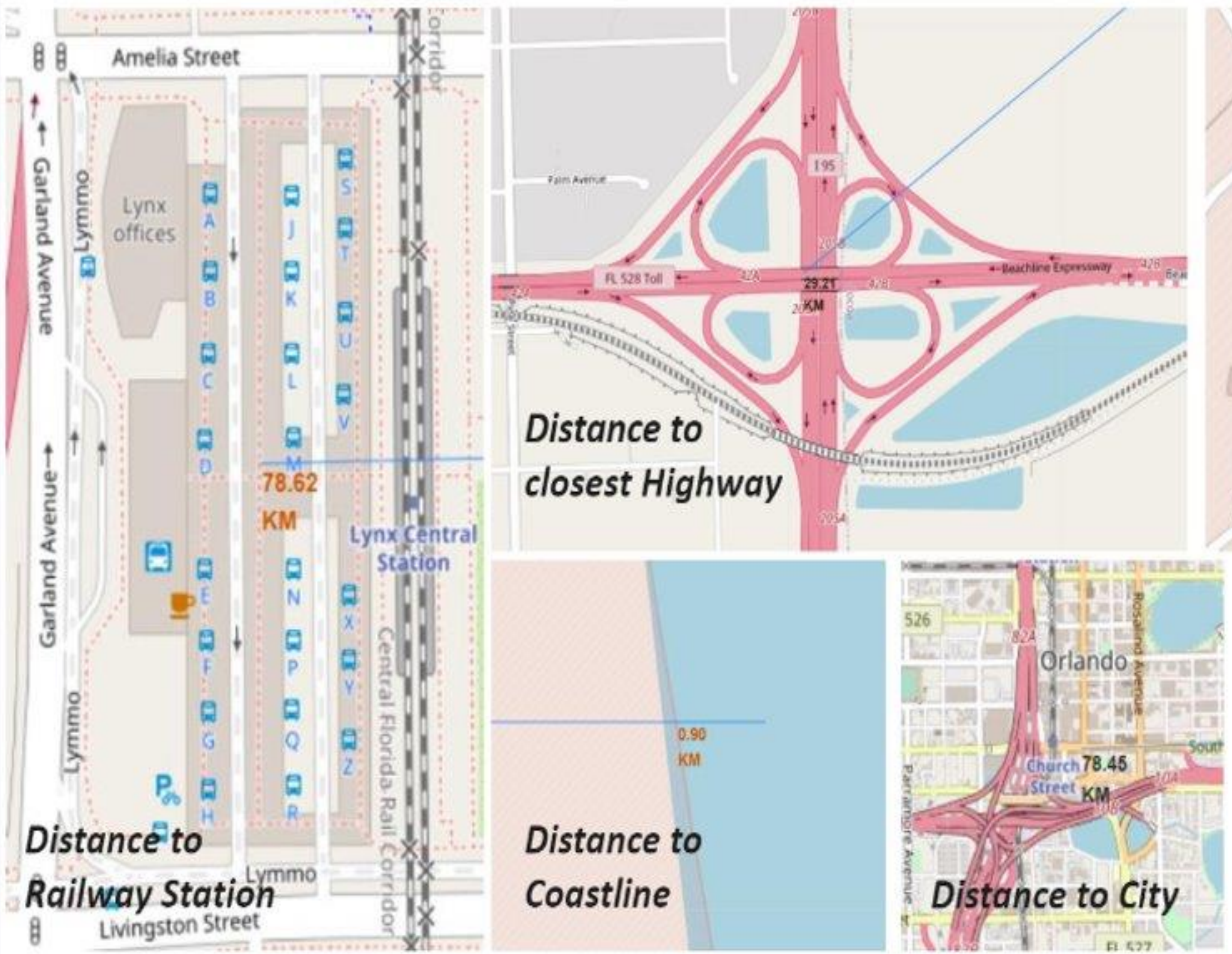
Global Launch Sites Map

This map shows the location of all active launch sites in the world. The launch sites are color-coded by region. The size of the launch site markers is proportional to the number of launches that have taken place from that site. The map also shows the launch success rate for each launch site.

Important elements:

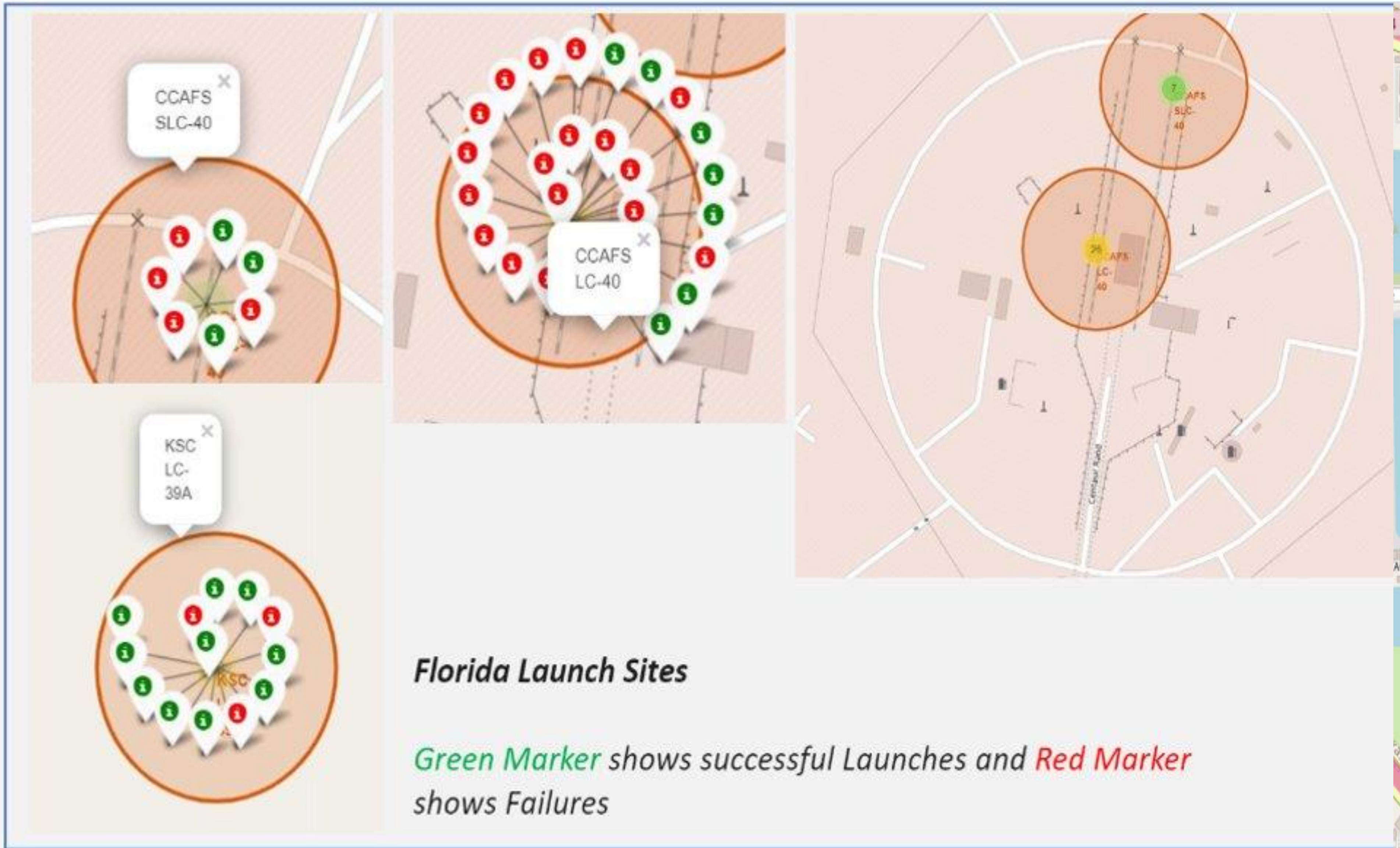
- The map shows that most launch sites are located near the Equator and the coast.
- The launch sites are color-coded by region.
- The size of the launch site markers is proportional to the number of launches that have taken place from that site.
- The map also shows the launch success rate for each launch site.

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

Markers showing launch sites with color labels



Section 4

Build a Dashboard with Plotly Dash

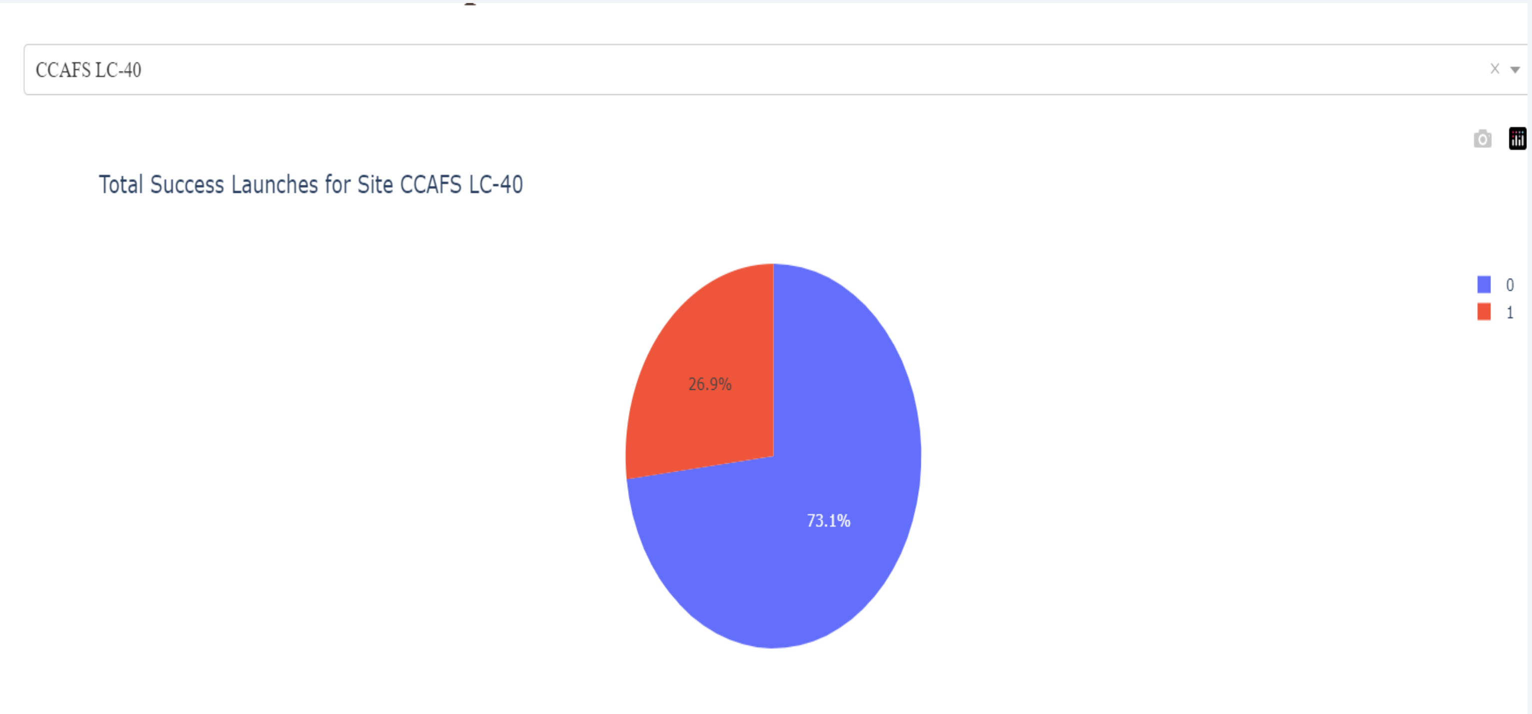
Pie chart showing the success percentage achieved by each launch site

Total Success Launches by Site

Explanation of the Screenshot: The screenshot shows a pie chart representing the distribution of launch success among different SpaceX launch sites. The chart clearly indicates that "KSC LC-39A" has the highest number of successful launches compared to other sites. This suggests that "KSC LC-39A" has been the top-performing launch site for SpaceX, demonstrating its reliability and success in mission outcomes.

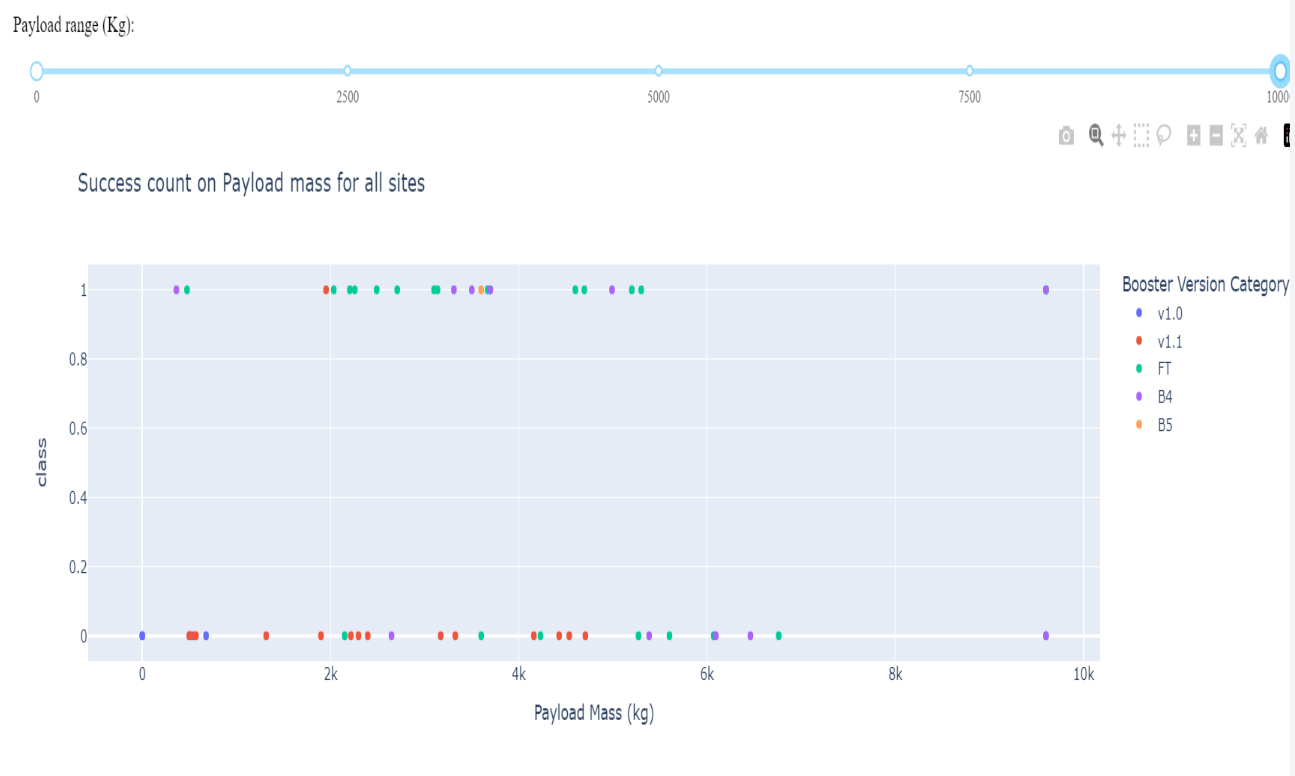
Pie chart showing the Launch site with the highest launch success ratio

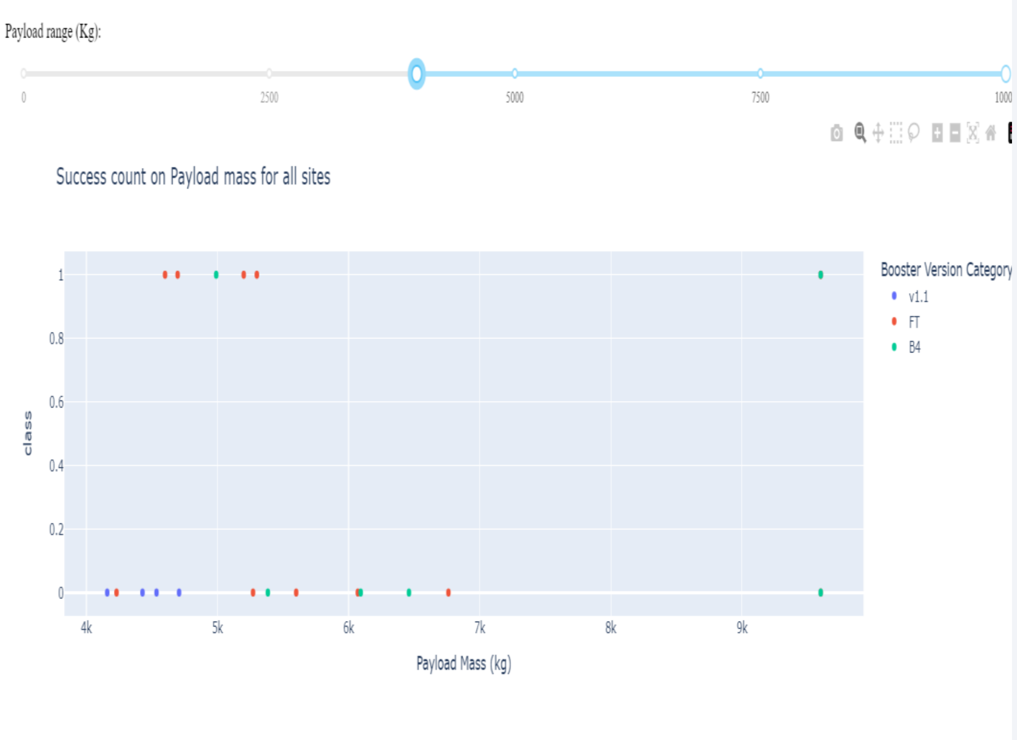
Explanation of the Screenshot: The screenshot displays a pie chart representing the launch success ratio for the launch site with the highest success rate. The chart reveals that "KSC LC-39A" achieved an impressive 73.1% success rate, indicating that a significant majority of launches from this site have been successful.



Additionally, the chart highlights a 26.9% failure rate, signifying that there have been some unsuccessful missions as well. This finding emphasizes the overall strong performance of "KSC LC-39A" in terms of launch success and underscores its position as the top-performing launch site for SpaceX

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





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier has the highest classification accuracy of 0.90, making it the most accurate model for predicting the class of a data point.
- The decision tree classifier is the best model for this task. Let's use it to make predictions.

ıte a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the c

```
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]}

     = DecisionTreeClassifier()
```

```
GridSearchCV(tree, parameters, cv=10)
ʼx train, y_train)
```

```
GridSearchCV
rchCV(cv=10, estimator=DecisionTreeClassifier(),
      param_grid={'criterion': ['gini', 'entropy'],
                  'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                  'max_features': ['auto', 'sqrt'],
                  'min_samples_leaf': [1, 2, 4],
                  'min_samples_split': [2, 5, 10],
                  'splitter': ['best', 'random']})
▾ estimator: DecisionTreeClassifier
DecisionTreeClassifier()
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
int("tuneu hpyerparameters :(best parameters) ",tree_cv.best_params_)
rint("accuracy :",tree_cv.best_score_)
```

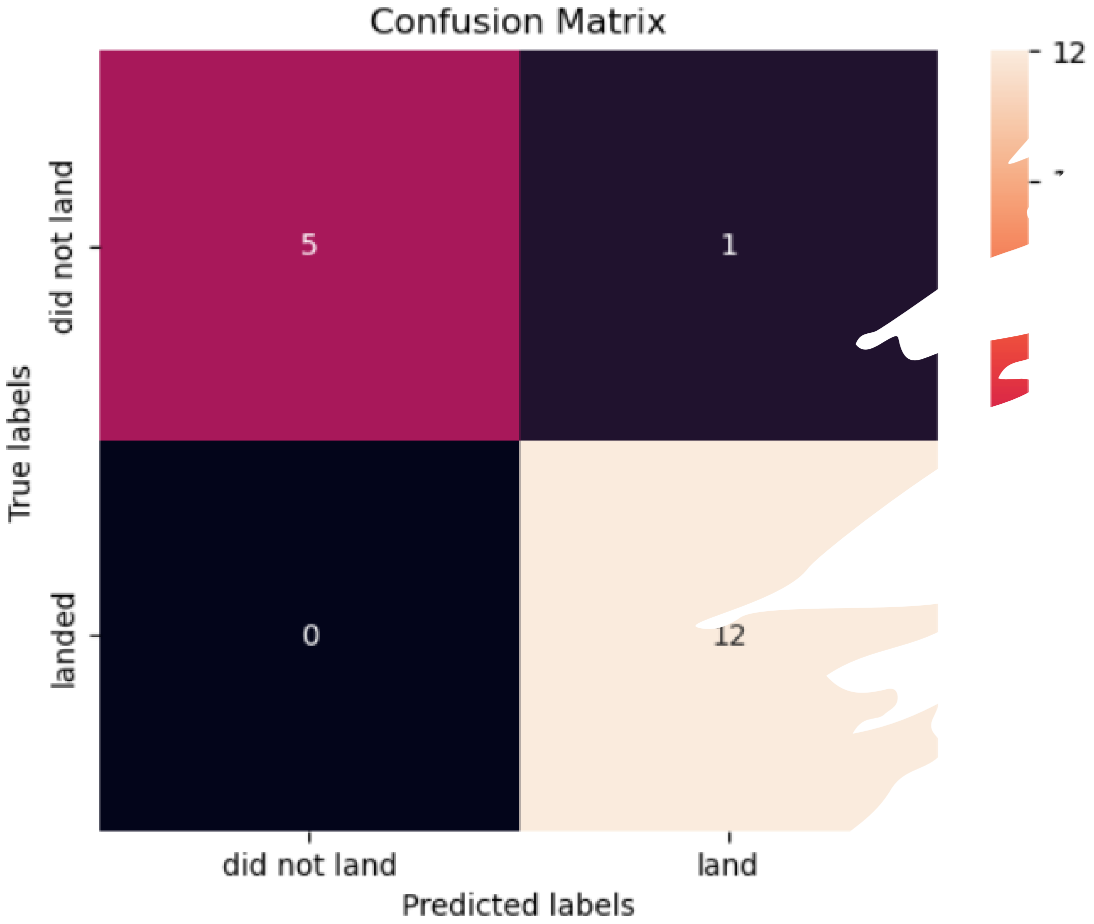
24] ✓ 0.0s

```
uned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 16, 'max_features': 'auto', 'min_sample
accu. cy : 0.8892857142857145
```

We can plot the confusion matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(y_test,yhat)
```

[26] ✓ 0.5s



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier is able to distinguish between successful and unsuccessful landings. However, the major problem is that the classifier is more likely to incorrectly classify an unsuccessful landing as a successful landing. This is a major problem, as it could lead to a plane being incorrectly cleared to land.

TASK 10

[1] Flight Amount and Success Rate

There is a positive correlation between the flight amount at a launch site and its success rate.

Launch sites with larger flight amounts tend to have higher success rates.

[2] Increase in Launch Success Rate

The launch success rate has shown a consistent increase since 2013 until 2020.

This trend indicates the improving efficiency and effectiveness of space missions.

[3] Successful Orbit Types

Orbits ES-L1, GEO, HEO, SSO, and VLEO have demonstrated the highest success rates.

These orbit types have consistently shown successful mission outcomes.

[4] Most Successful Launch Site

KSC LC-39A stands out as the launch site with the highest number of successful launches.

Its track record showcases a remarkable success rate compared to other sites.

[5] Best Machine Learning Algorithm

The Decision tree classifier emerges as the most suitable machine learning algorithm for this task.

It outperforms other algorithms in terms of accuracy and predictive capabilities.

Conclusion:

The analysis highlights important findings related to flight amount, launch success rate, orbit types, launch sites, and machine learning algorithms.

These insights provide valuable information for future space missions and decision-making processes.

Appendix

- All relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or datasets are available on GitHub

Thank you!