IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

SCIFO ROBERTO
05/30/2024

Outline

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

Executive Summary

- Summary of methodologies
  - **Data Collection through API:** Using Application Programming Interfaces (APIs) to collect organized data.
  - **Data Collection with Web Scraping**: Implementing methods to pull data from web pages.
  - **Data Wrangling:** Refining, structuring, and enhancing raw data to achieve a desired format for quicker decision-making.
  - **Exploratory Data Analysis with SQL**: Leveraging Structured Query Language (SQL) to explore and examine data, uncovering patterns and insights.
  - **Exploratory Data Analysis with Data Visualization:** Developing visual data representations to identify hidden patterns and clearly communicate information.
  - **Interactive Visual Analytics with Folium:** Utilizing the Folium library to map geospatial data and create interactive maps for comprehensive analysis.
  - **Machine Learning Prediction:** Employing machine learning techniques to forecast future events or results based on historical data.
- Summary of all results
  - **Exploratory Data Analysis result:** Revealing trends and patterns discovered through preliminary data examination.
  - **Interactive analytics in screenshots:** Visual analytics that facilitate interactive data exploration, typically captured as screenshots.
  - **Predictive Analytics result:** Results from machine learning forecasts, detailing the accuracy, efficiency, and potential uses.

Introduction

- **Project background and context**

SpaceX has revolutionized the space industry with its Falcon 9 rocket, available at a price of $62 million per launch on its website. This pricing is notably lower than other providers, who charge more than $165 million. A key factor in SpaceX's cost efficiency is its ability to reuse the first stage of the Falcon 9. Predicting whether the first stage will land successfully is not only a technical challenge but also an economic one. If we can forecast successful landings, we can more accurately estimate the cost of a launch. This prediction could be invaluable for other companies looking to compete with SpaceX in the launch services market. The aim of this project is to create a machine learning pipeline that can predict the successful landing of the rocket's first stage.

- **Problems you want to find answers**

What factors influence the successful landing of a rocket?

How do the various factors interact to determine landing success?

What operational conditions are necessary to ensure a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data Acquisition via SpaceX API: The dataset was gathered using the SpaceX REST API, which offers extensive information on SpaceX launches. The particular endpoint utilized for this project was api.spacexdata.com/v4/launches/past, providing data on previous SpaceX missions.
- Perform data wrangling
  - The gathered data was enhanced by generating a landing outcome label, derived from outcome data after summarizing and analyzing key features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The collected data up to this point was normalized, split into training and test sets, and then assessed using four distinct classification models. The accuracy of each model was evaluated using various combinations of parameters.

Data Collection

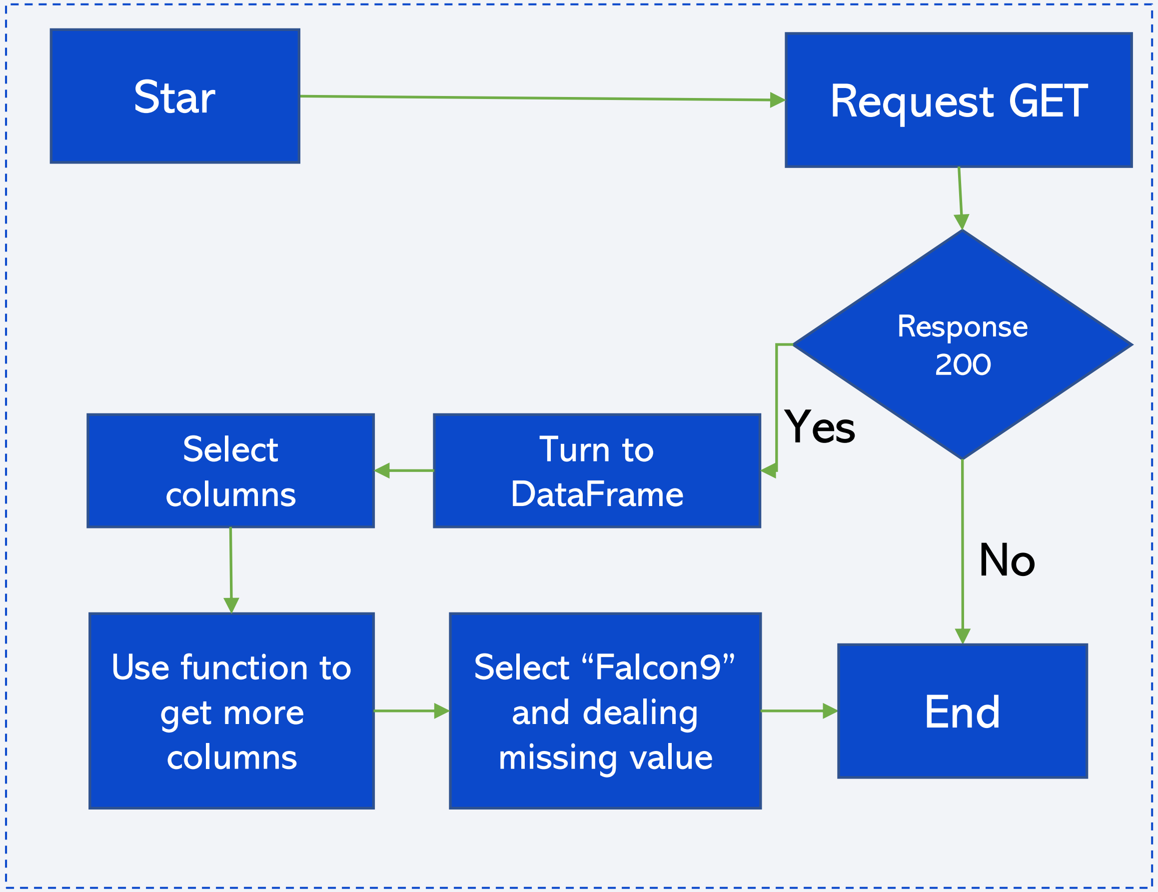
- Describe how data sets were collected.

✓ API Data Retrieval: Data was gathered from the SpaceX API (https://api.spacexdata.com/v4/rockets/).

✓ Web Scraping: Information was extracted from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches) using web scraping techniques.

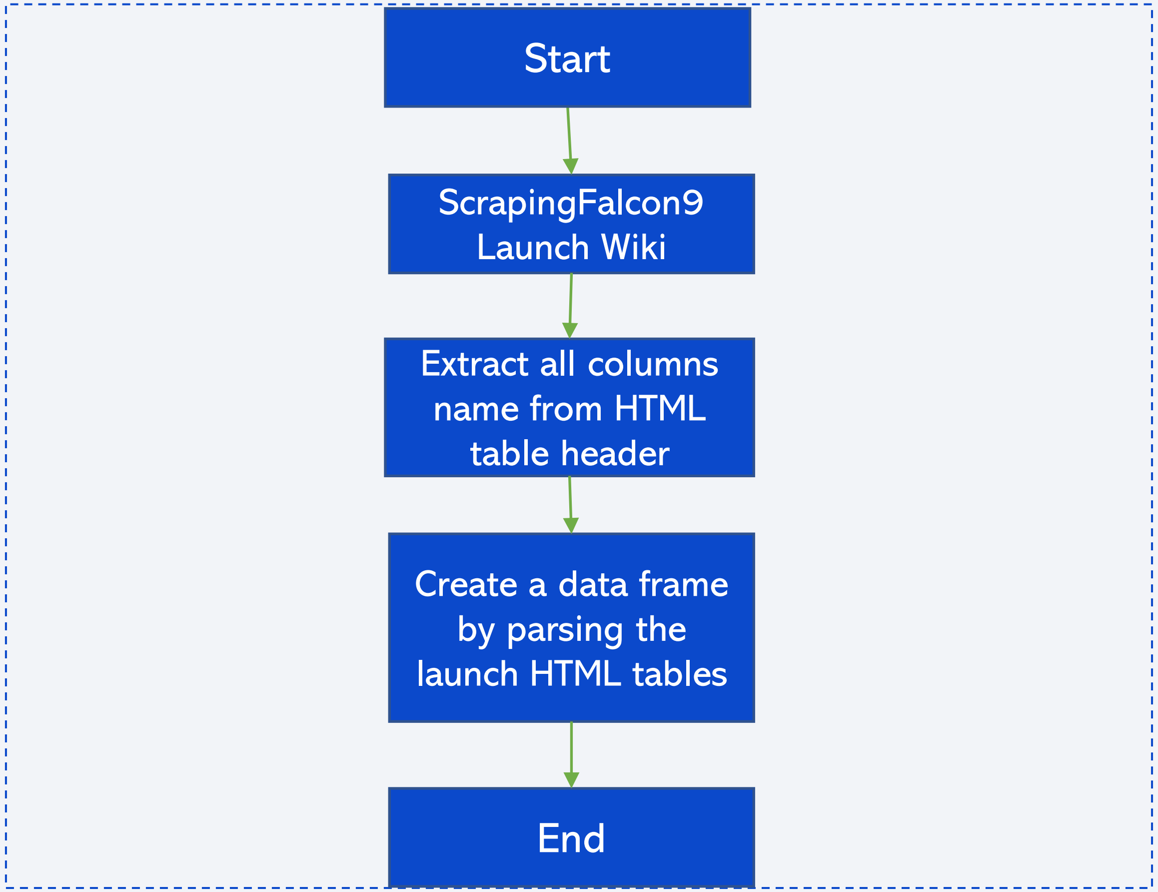
Data Collection – SpaceX API

- Fetch and Parse SpaceX Data via GET Request: Retrieve data from the SpaceX API using a GET request.
- Convert JSON to DataFrame: Transform the JSON response into a pandas DataFrame.
- Utilize Specific Columns: Focus on the columns: rocket, payloads, launchpad, and cores.
- Extract Key Information: Obtain BooterVersion, LaunchSite, PayloadData, CoreData, and other relevant details.
- Filter DataFrame for Specific Booster: Limit the DataFrame to entries where BoosterVersion is "Falcon 9".
- Handle Missing Values: Replace missing values with the column mean.
- GitHub Repository: Refer to [jupyter-labs-spacex-data-collection-api.ipynb]() for the code implementation.



Data Collection - Scraping

- Request the Falcon9 Launch Wiki Page from URL: Obtain the Falcon9 launch information from the specified web URL.
- Web Scraping with BeautifulSoup: Utilize BeautifulSoup for extracting data from web pages.
- Extract Table Column Names from HTML Header: Retrieve all column names from the headers of HTML tables.
- Parse Launch Tables into a DataFrame: Construct a data frame by parsing the HTML tables containing launch data.
- GitHub Resource: Refer to jupyter-labs-webscraping.ipynb for the complete notebook on GitHub.



Data Wrangling

Handling Missing Data

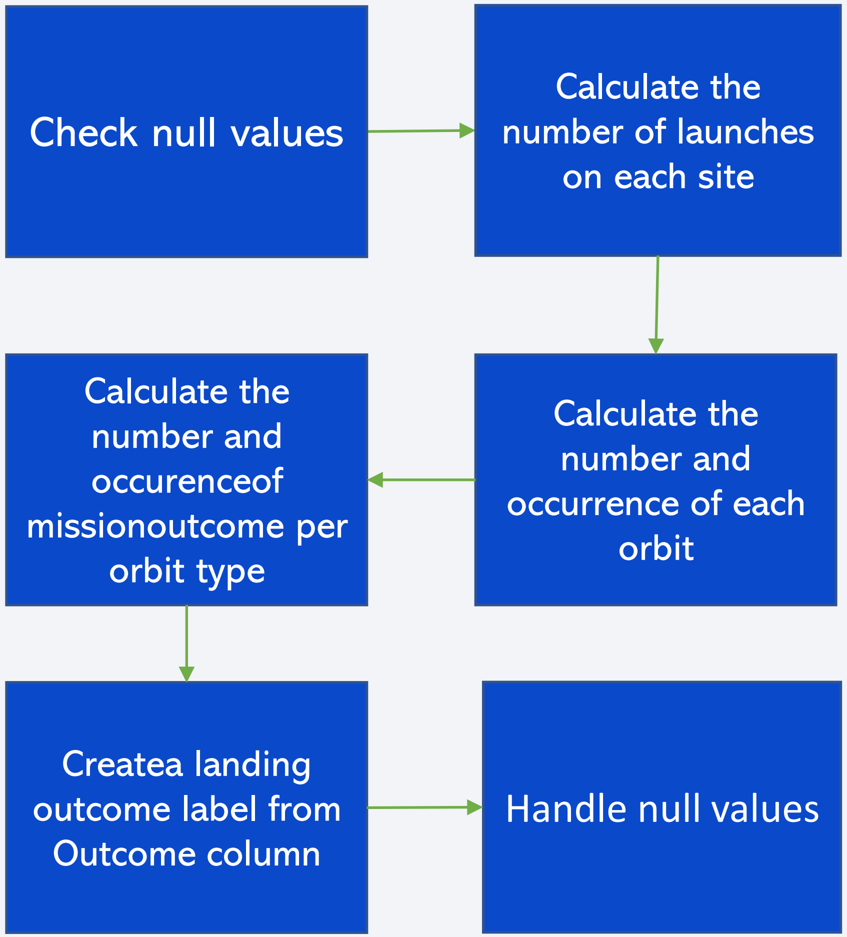
- **Null Values in Payload Mass and Landing Pad:** In the dataset, some entries for Payload Mass and Landing Pad are missing.

- **Imputation for Payload Mass:** For Payload Mass, we replace the missing values with the average of the other entries using the .mean() and .replace() functions from the numpy package.

- **Handling Missing Landing Pad Values:** The null values for Landing Pad indicate instances where landing pads were not utilized.

- **Creation of Class Column:** A new column named "Class" will be introduced, representing whether the landing was successful (1) or a failure (0).

- GitHub Repository: Refer to [labs-jupyter-spacex-Data wrangling.ipynb] for the code implementation.



EDA with Data Visualization

Summarize what charts were plotted and why you used those charts

- **Scatter Plots:** These graphs illustrate how one variable influences another. The connection between two variables is known as their correlation. Scatter plots typically involve a significant amount of data. Examples include: Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload vs. Launch Site, Orbit vs. Flight Number, Payload vs. Orbit Type, and Orbit vs. Payload Mass.
- **Bar Charts:** These graphs simplify comparing data sets across different groups at a glance. One axis represents categories, while the other shows a discrete value. The purpose is to display the relationship between these two axes. Bar charts can also highlight substantial changes in data over time. Example: Mean vs. Orbit.
- **Line Graphs:** These graphs are valuable for clearly displaying data variables and trends, aiding in predictions about future data not yet recorded

- GitHub Repository: Refer to [edadataviz.ipynb]() for the code implementation.

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Connect to the database
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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.

- GitHub Repository: Refer to jupyter-labs-eda-sql-coursera_sqllite.ipynb for the code implementation.

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- **Markers:** Used to denote specific points of interest, such as launch sites.
- **Circles:** Highlight areas surrounding particular coordinates, for instance, the NASA Johnson Space Center.
- **Marker Clusters:** Represent groups of events at each location, like multiple launches at a single site.
- **Lines:** Depict distances between two geographical points.

- GitHub Repository: Refer to **lab_jupyter_launch_site_location.ipynb** for the code implementation.

Build a Dashboard with Plotly Dash

Summarize what plots/graphs and interactions you have added to a dashboard

- **Dropdown Menu:** Utilize to obtain the type of launch site as input. Users can choose from the available options to display different graph styles based on their selection.
- **Pie Chart:** Illustrates the success rates of all launch sites for the "All Sites" option and shows the proportions of success versus failure for each selected launch site.
- **Scatter Plot:** Compares Payload Mass (kg) against the success class.

GitHub Repository: Refer to spacex_dash_app.py for the code implementation.

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- **Standardize and Split Data**: Begin by standardizing the data and dividing it into training and testing sets.
- **Model Selection and Parameter Setting:** Choose a model and configure its parameters.
- **Training the Model:** Train the selected model on the training dataset.
- **Evaluate Accuracy:** Compute the accuracy score using the test dataset.
- **Model Training Decision:** Determine if additional models need to be trained based on the results.
- **Model Comparison:** Compare the performance of all trained models and select the best one.

GitHub Repository: Refer to SpaceX_Predicción de aprendizaje automático_Parte_5.ipynb for the code implementation

Results

Exploratory Data Analysis Findings

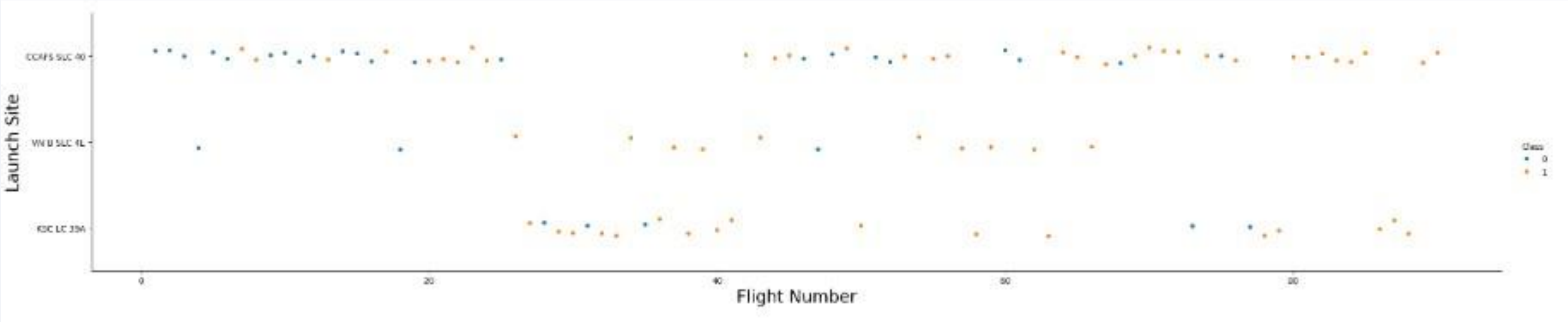
- **Launch Sites:** SpaceX utilizes four distinct launch sites.
- **Initial Launches:** The first launches were conducted for SpaceX and NASA.
- **Average Payload:** The average payload capacity of the Falcon 9 v1.1 booster is 2,928 kg.
- **First Successful Landing:** The first successful landing occurred in 2015, five years after the initial launch.
- **Successful Landings:** Many versions of the Falcon 9 booster achieved successful landings on drone ships, often carrying payloads above the average weight.
- **Mission Success Rate:** Nearly 100% of the mission outcomes were successful.
- **Landing Failures in 2015:** Two versions of the booster, F9 v1.1 B1012 and F9 v1.1 B1015, failed to land on drone ships in 2015.
- **Improved Landing Outcomes:** The success rate of landings has improved over the years.
- **Interactive Analytics Insights:** Through interactive analytics, it was identified that launch sites are typically located in safe areas, often near the sea, and have good logistical infrastructure.
- **Launch Frequency:** Most launches occur at East Coast launch sites.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

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

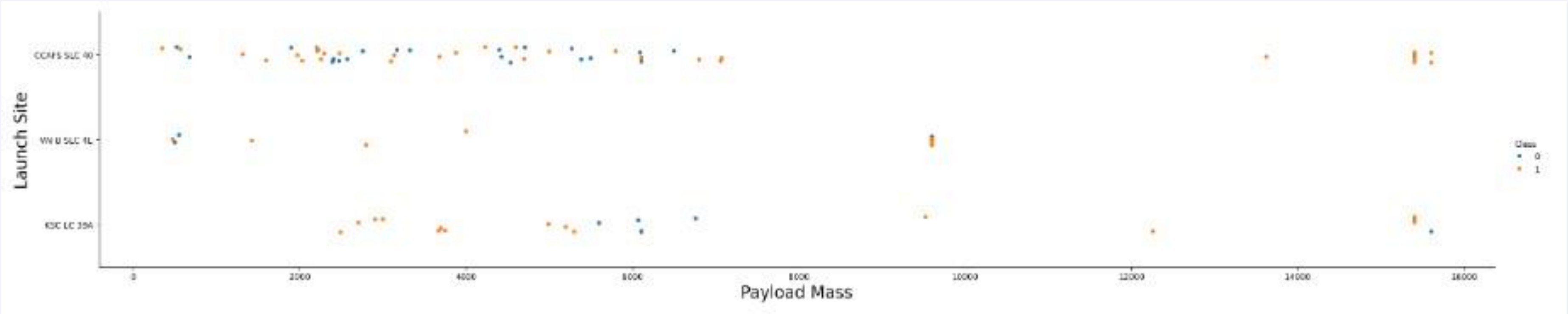


- Show the screenshot of the scatter plot with explanations

- **Scatter Plot Description:** A scatter plot of Flight Number versus Launch Site, with outcomes color-coded (blue for unsuccessful, orange for successful).
- **Launch Site Performance:** CCAFS has conducted the most tests. However, VAFB and KSC exhibit higher success rates, likely due to their usage in later flights where reliability may have improved.

Payload vs. Launch Site

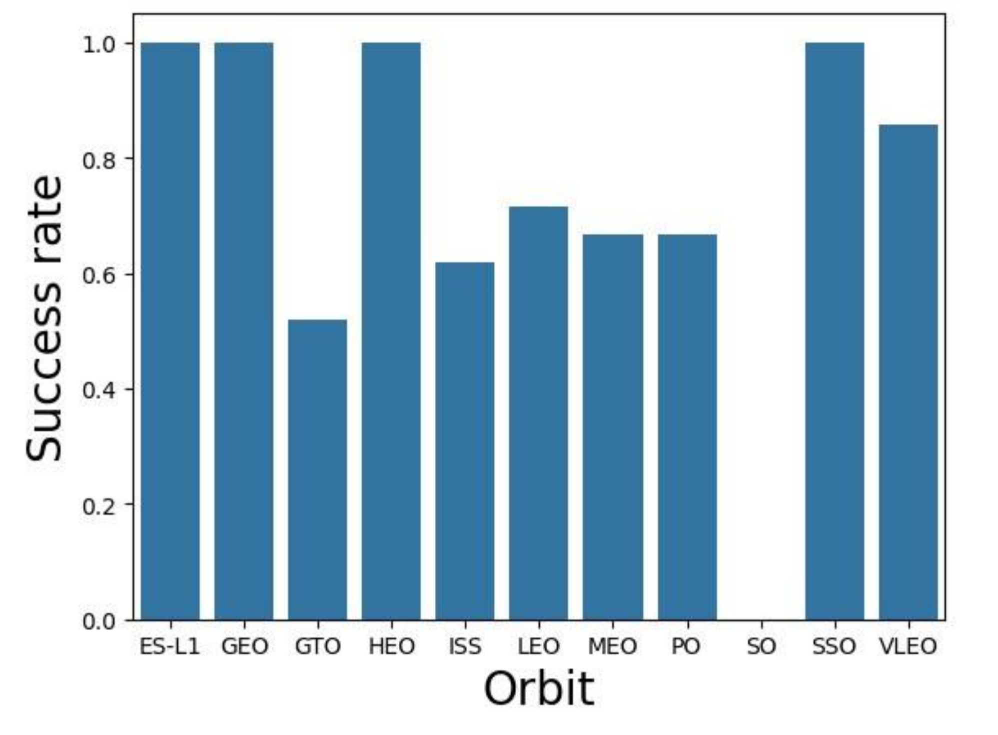
- Show a scatter plot of Payload vs. Launch Site



- Show the screenshot of the scatter plot with explanations
- **Scatter Plot of Payload vs. Launch Site:** The data visualization depicts payload against the launch site, with outcomes color-coded (blue for unsuccessful, orange for successful).
- **Observations on VAFB Launches:** VAFB has recorded fewer heavy payload launches.
- **Correlation Between Payload and Success:** Higher payload launches tended to be more successful. This trend might be attributed to engineers opting for such launches only when they were confident in the improved reliability and the safety of the additional payload cost.

Success Rate vs. Orbit Type

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



Show the screenshot of the scatter plot with explanations

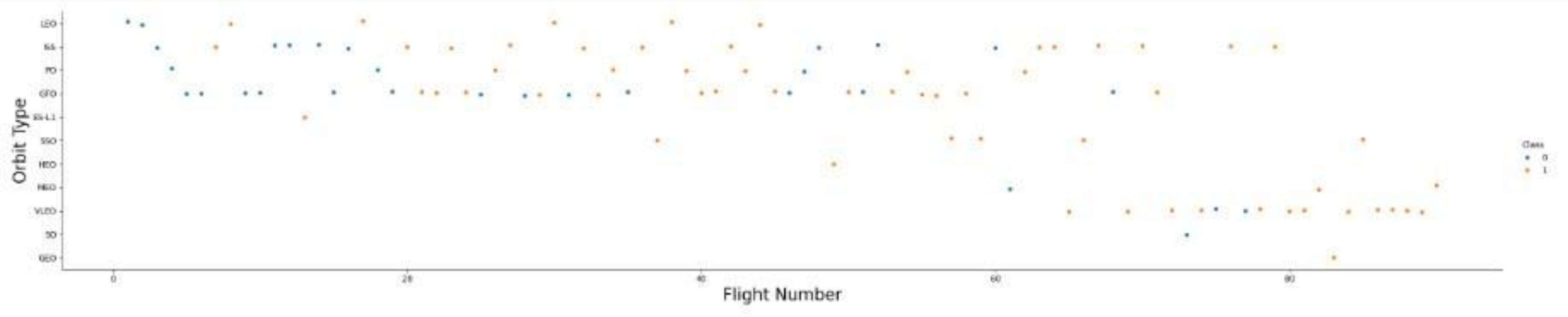
Bar Chart of Success Rates by Orbit Type: This visualization presents the success rates associated with different types of orbits.

Success in Higher Orbits: Higher orbits, such as GEO and HEO, generally demonstrate greater success rates.

Clarification on SSO and SO: It is important to recognize that SSO and SO are considered the same, which means this orbit type does not achieve a perfect 100% success rate.

Flight Number vs. Orbit Type

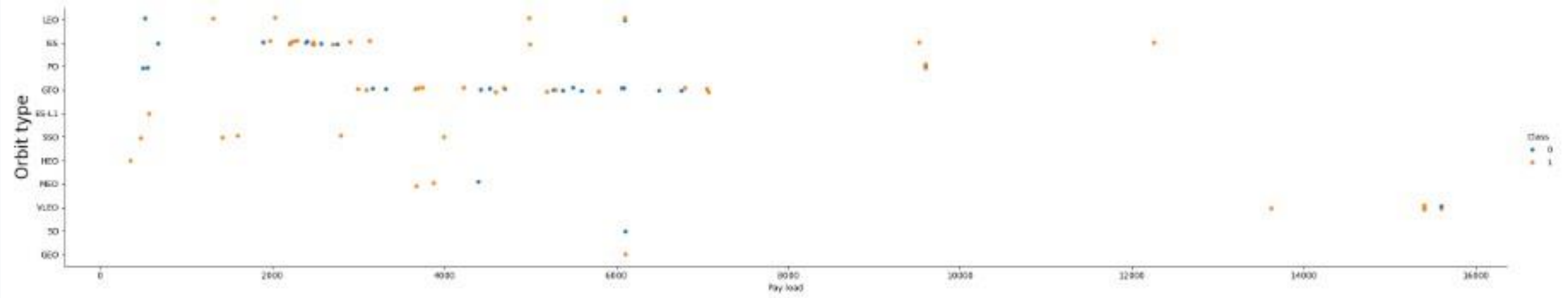
- Show a scatter point of Flight number vs. Orbit type



- Show the screenshot of the scatter plot with explanations
- **Scatter Plot of Flight Number vs. Orbit Type:** This scatter plot displays the relationship between flight numbers and orbit types, with outcomes color-coded (blue indicating unsuccessful outcomes, and orange indicating successful ones).
- **Improvement Over Time:** As the number of flights has increased, the success rate across various orbit types appears to have improved.

Payload vs. Orbit Type

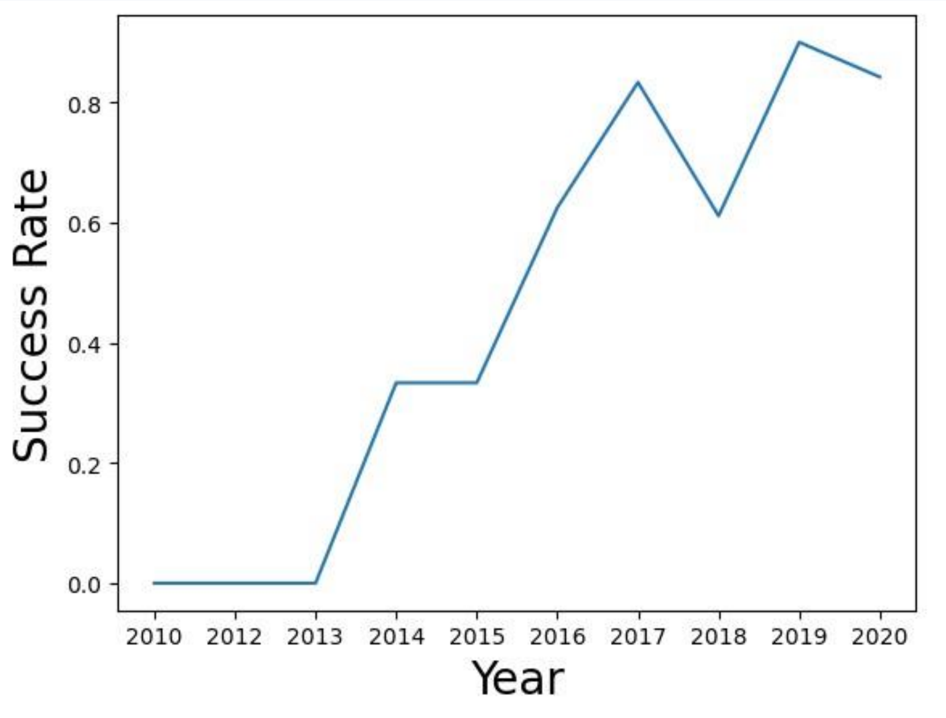
- Show a scatter point of payload vs. orbit type



- Show the screenshot of the scatter plot with explanations
- **Scatter Plot of Payload vs. Orbit Type**: This scatter plot illustrates the relationship between payload and orbit type, with outcomes color-coded (blue for unsuccessful, orange for successful).
- **Lack of Clear Patterns:** No distinct pattern of increased success is observed with higher payloads for any specific orbit type.
- **Consistency in SSO Orbit**: The SSO orbit appears to maintain a high success rate, though it should be noted that SSO is equivalent to SO, indicating that the success rate is not a perfect 100%.

Launch Success Yearly Trend

Show a line chart of yearly average success rate



Show the screenshot of the scatter plot with explanations

The line chart supports the conclusion that the overall success rate has significantly improved over the years, reflecting advancements and enhanced reliability in operations. Despite the setback in 2018, the general upward trend indicates a positive trajectory in performance and success rates over time.

%%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;

All Launch Site Names

Find the names of the unique launch sites

%%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;

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

%%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

%%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXDATASET WHERE customer = 'NASA (CRS)';
```

- Present your query result with a short explanation here

To determine the total payload mass carried by NASA boosters, we use the **SUM** function to aggregate the values in the **PAYLOAD_MASS__KG_** column. The **WHERE** clause ensures that only records where the customer is 'NASA (CRS)' are included in this calculation. This query sums up all relevant payload masses to provide the total payload carried by NASA for their CRS missions.

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

%sqlselect avg(PAYLOAD_MASS__KG_) as payloadmassfrom SPACEXDATASET where Booster_Version= 'F9 v1.1';

- Present your query result with a short explanation here

To determine the average value in the **PAYLOAD_MASS__KG_** column, use the **AVG** function. The **WHERE** clause filters the dataset to calculate the average only for rows where **Booster_Version** equals **'F9 v1.1'**.

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

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

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

Hint:Use min function

In [27]:
```
%%sql
SELECT MIN(Date) FROM SPACEXTBL
    WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Out[27]:

MIN(Date)
2015-12-22

- Present your query result with a short explanation here

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

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

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

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- Present your query result with a short explanation here

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome) AS COUNT FROM SPACEXTBL
    GROUP BY Mission_Outcome;
```

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

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

- Present your query result with a short explanation here

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

```
%%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
F9 B5 B1049.7

- Present your query result with a short explanation here

2015 Launch Records

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

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%%sql
SELECT SUBSTR(Date,6,2), Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL
    WHERE Landing_Outcome LIKE 'Failure (drone ship)' AND
        SUBSTR(Date,0,5) = '2015'
```

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

SUBSTR(Date,6,2)	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

- Present your query result with a short explanation here

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

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.

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

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

Landing_Outcome	OutcomeCount
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

- Present your query result with a short explanation here

Section 3

Launch Sites Proximities Analysis

Launch site locations

- **Interactive Map Exploration:** Navigate the generated Folium map and capture a comprehensive screenshot that displays all launch site location markers on a global scale.
- **Explanation of Key Elements and Insights:** Provide a detailed explanation of the significant elements and observations visible in the screenshot.
- **Locations of SpaceX Launch Sites:** SpaceX's launch sites are situated along the American coastlines, specifically in Florida and California.

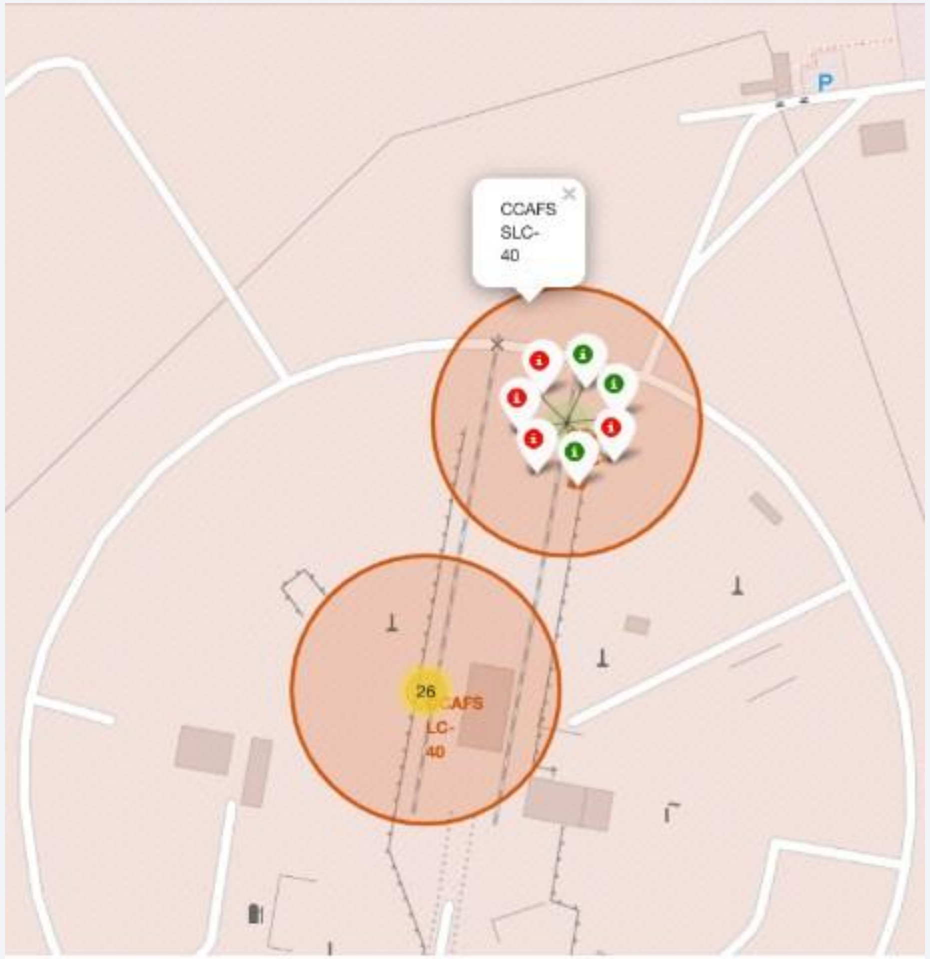
Launch Locations Success/Failures

- **Explore the Folium Map:** Investigate the interactive map created with the Folium library to visualize launch outcomes, color-coded for clarity.

- **Capture a Proper Screenshot:** Take a clear screenshot of the map displaying the color-labeled launch outcomes.

Explanation of Elements and Findings:

- **Green Markers:** Indicate successful launches.
- **Red Markers:** Represent failed launches.



Calculate Distances from Launch Site to Nearby Locations

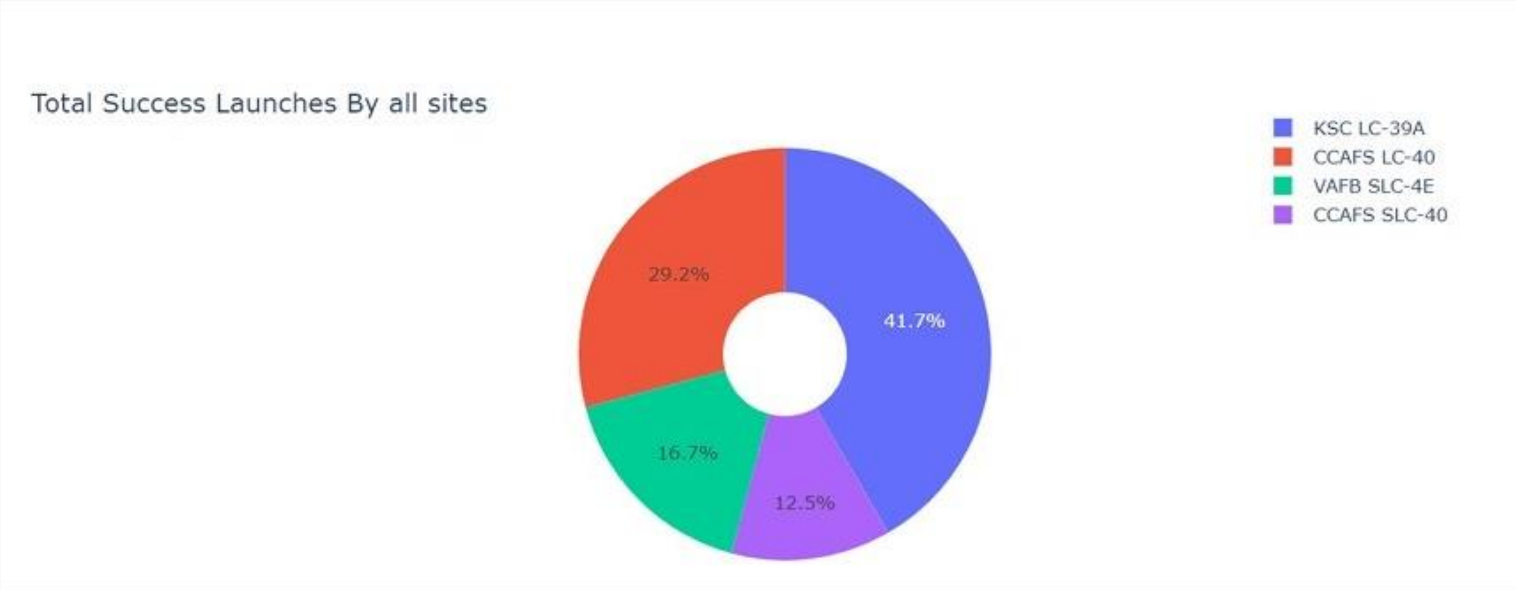
- **Explore the Generated Folium Map**
- **Task:** Examine the folium map created and capture a screenshot of a chosen launch site, including its surrounding areas such as railways, highways, and coastlines. Ensure that the distances are calculated and shown on the map.
- **Explanation:** Describe the key components and discoveries visible in the screenshot. For instance, illustrate the distance to the nearest coastline.

Section 4

Build a Dashboard with Plotly Dash

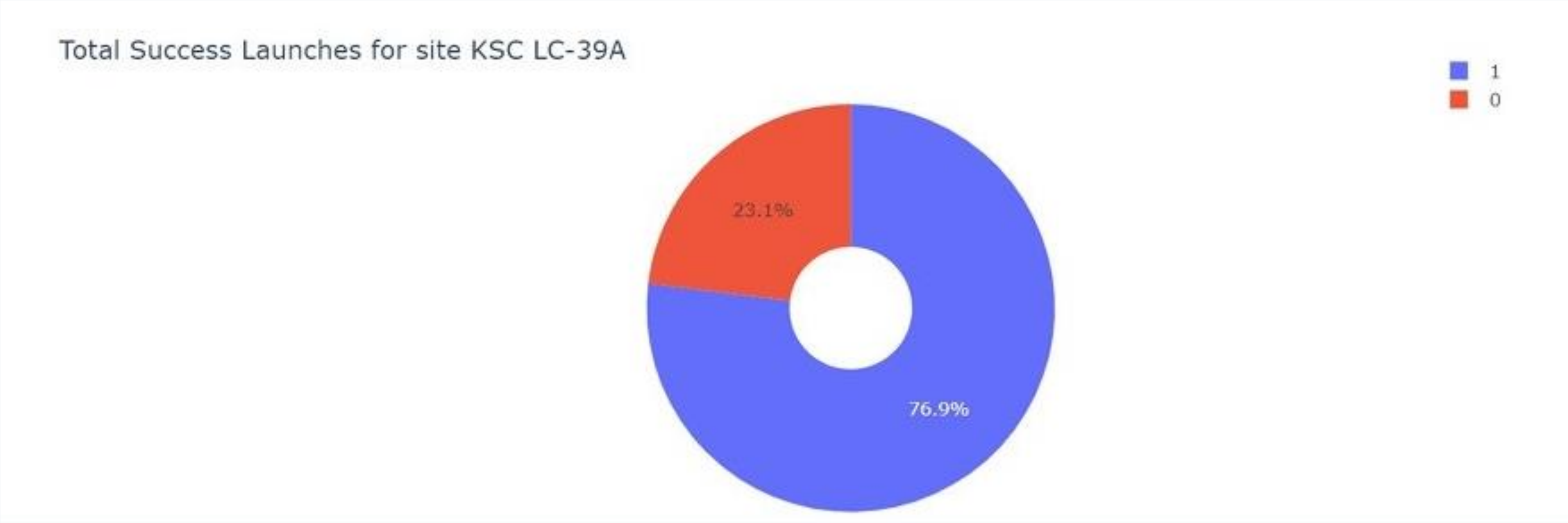
Complete Overview of Successful Launches by All Sites

- **Display the Screenshot:** Present a pie chart illustrating the count of successful launches for all sites.
- **Explanation of Key Elements and Findings:** Break down and discuss the significant components and insights from the screenshot.
- **Notable Observation:** It's evident that KSC LC-39A achieved the highest number of successful launches.



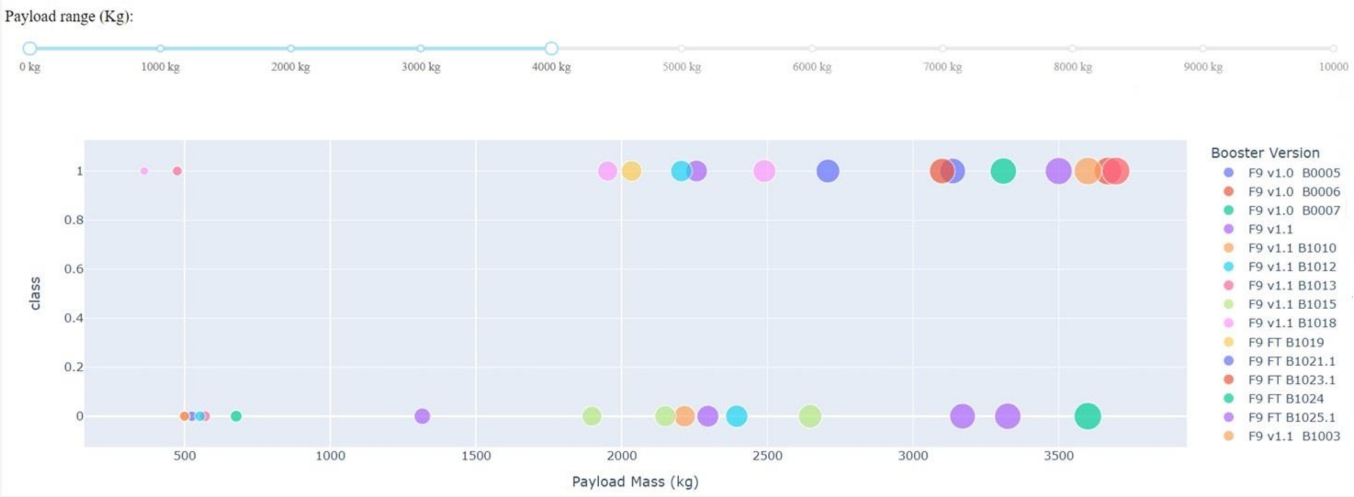
Complete Success Rate for Site KSC LC-39A

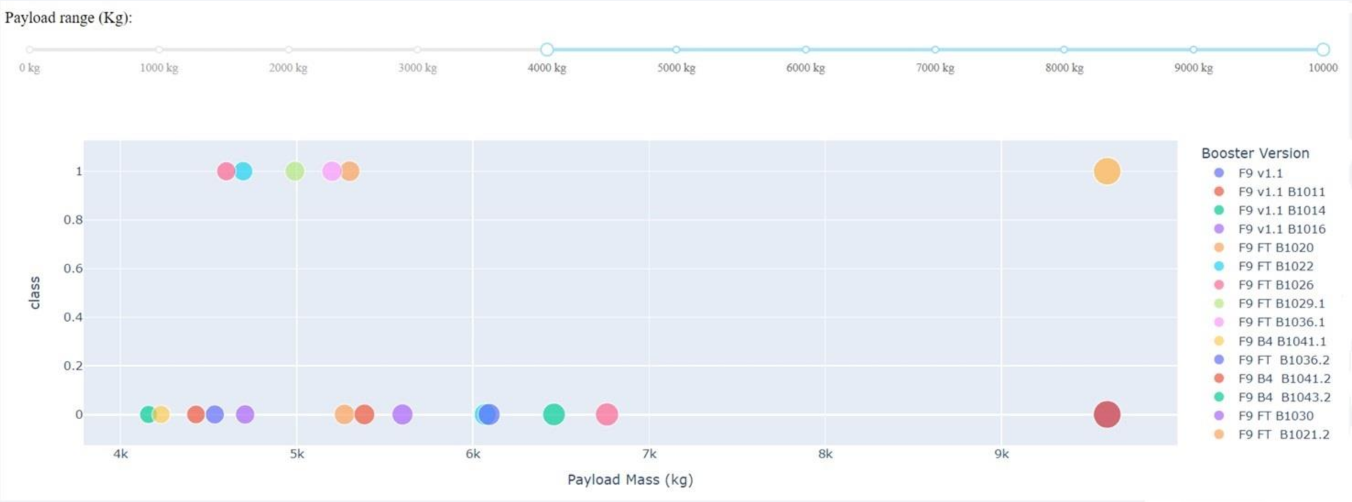
- **Display the Pie Chart Screenshot:** Present the pie chart for the launch site with the highest success rate.
- **Explanation of Key Elements and Findings:** Describe the significant components and insights from the screenshot.
- **KSC LC-39A Performance:** The launch site KSC LC-39A recorded a 76.9% success rate and a 23.1% failure rate.



Payload vs. Mass with Different Payloads

- Display screenshots of Payload vs. Launch Outcome scatter plot for all sites, with varying payloads selected in the range slider.
- Describe the key elements and discoveries in the screenshot, such as identifying which payload range or booster version exhibits the highest success rate.
- Observe that the success rates for lighter payloads are higher compared to heavier ones.





Section 5

Predictive Analysis (Classification)

Classification Accuracy

Accuracy Evaluation: Employing the "score" method, the precision for each predictive technique was found to be identical.

Accuracy of logistic regression method: 0.8333333333333334

Accuracy of support vector machine method: 0.8333333333333334

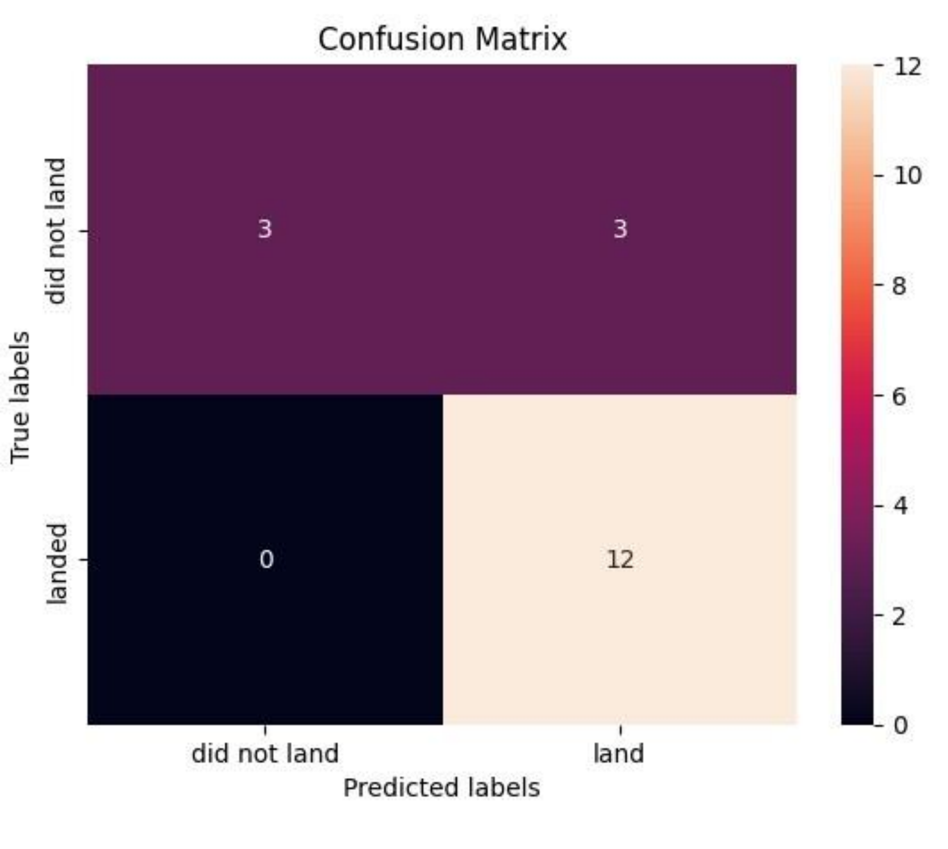
Accuracy of decision tree method: 0.8333333333333334

Accuracy of k nearest neighbors method: 0.8333333333333334

Confusion Matrix

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

- Analyzing the Confusion Matrix: The confusion matrix reveals that the Tree model can differentiate between various classes effectively. However, it highlights a significant issue: a high rate of false positives.



Conclusions

- Consistently lighter payloads have shown better performance metrics, suggesting a potential focus area for future payload designs.

- Historical data indicates a positive trend in SpaceX's launch success rates, reinforcing their progressive improvements in technology and execution.

- The decision tree model's efficacy highlights its suitability for this type of classification problem, making it a valuable tool for predicting future launch outcomes.

Appendix

- **Storage of Relevant Assets:** The essential assets are hosted on GitHub, encompassing all Notebooks and Python scripts developed throughout this project.

- GitHub Repository: Refer to https://github.com/rscifo/IBM-DataScience for the code implementation.

Thank you!