IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nolan Rink
8/30/24

Outline

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

Executive Summary

- **API Data Collection**:
- **Web Scraping**: Collected data from websites using BeautifulSoup and Requests in Python.
- **Exploratory Data Analysis (EDA)**: Visualized data using Python libraries (Pandas, Matplotlib, Seaborn) to identify trends and patterns.
- **SQL Analysis**: Used SQL queries in Python to extract and analyze data from an SQLite database.
- **Geospatial Analysis**: Mapped SpaceX launch sites using Folium for geographic insights.
- **Data Wrangling**: Cleaned and prepared data using Pandas to ensure quality and consistency.
- **Machine Learning Models**: Developed and evaluated predictive models (logistic regression, decision trees) for SpaceX launch outcomes.

- **API Data Success**: Collected and prepared real-time SpaceX data for analysis.
- **Web Scraping Efficiency**: Automated collection of relevant online data.
- **EDA Insights**: Identified key trends and patterns in data attributes.
- **SQL Data Insights**: Successfully manipulated large datasets to uncover insights.
- **Geospatial Insights**: Visualized and analyzed SpaceX launch site locations.
- **Data Quality**: Produced clean, consistent datasets ready for analysis.
- **Predictive Insights**: Identified effective models for predicting SpaceX launch success.

Introduction

Project Background and Context

- The project involves analyzing SpaceX data, utilizing various data science techniques such as EDA, SQL analysis, API data collection, web scraping, geospatial analysis, data wrangling, and machine learning.

- The primary focus is on understanding and predicting the outcomes related to SpaceX launches, exploring factors that influence launch success, and analyzing data from various sources to provide actionable insights.

Questions We Answer

- Predicting SpaceX Launch Outcomes: Developing machine learning models to predict the success of SpaceX launches based on historical data.

- Analyzing Factors Influencing Launch Success: Identifying and understanding various factors that may affect the success of launches, such as weather, location, and technical parameters.

- Geospatial Analysis of Launch Sites: Understanding the geographical distribution of launch sites and its strategic implications.

- Optimizing Launch Strategies: Using predictive models and data analysis to suggest improvements in SpaceX's future launch strategies.

Section 1

Methodology

Methodology

Executive Summary

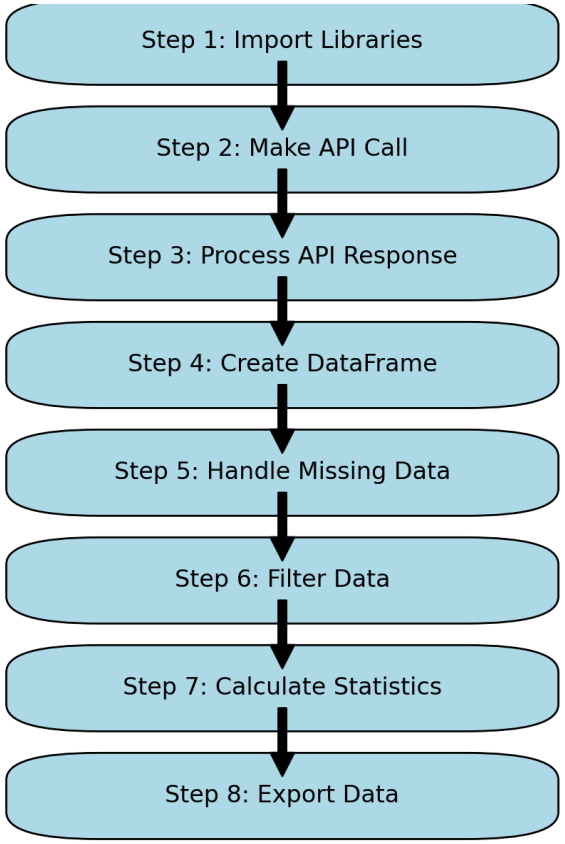
- Data collection methodology:
  - Collected data from Wikipedia using BeautifulSoup and Requests in Python.
  - Retrieved and processed real-time data from the SpaceX API using Python.
- Perform data wrangling
  - Used SQL queries in Python to extract and analyze data from an SQLite database.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- API Data Collection:
  - Used SpaceX API to collect real-time launch data.
  - Python libraries (e.g., requests) for API calls.
  - Data processed and cleaned for analysis.
- Web Scraping:
  - Gathered supplementary data from SpaceX-related websites.
  - Utilized BeautifulSoup and Requests for HTML parsing.
  - Automated extraction of relevant information (e.g., launch details, historical data).
  - Database Queries (SQL):
- Employed SQL queries to retrieve data from an SQLite database
  - Combined multiple datasets using joins and aggregations.
  - Filtered and extracted data to focus on specific analysis needs.
- Geospatial Data:
  - Collected geographical data on SpaceX launch sites.
  - Utilized Python libraries (e.g., Folium) for mapping and spatial analysis.
  - You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

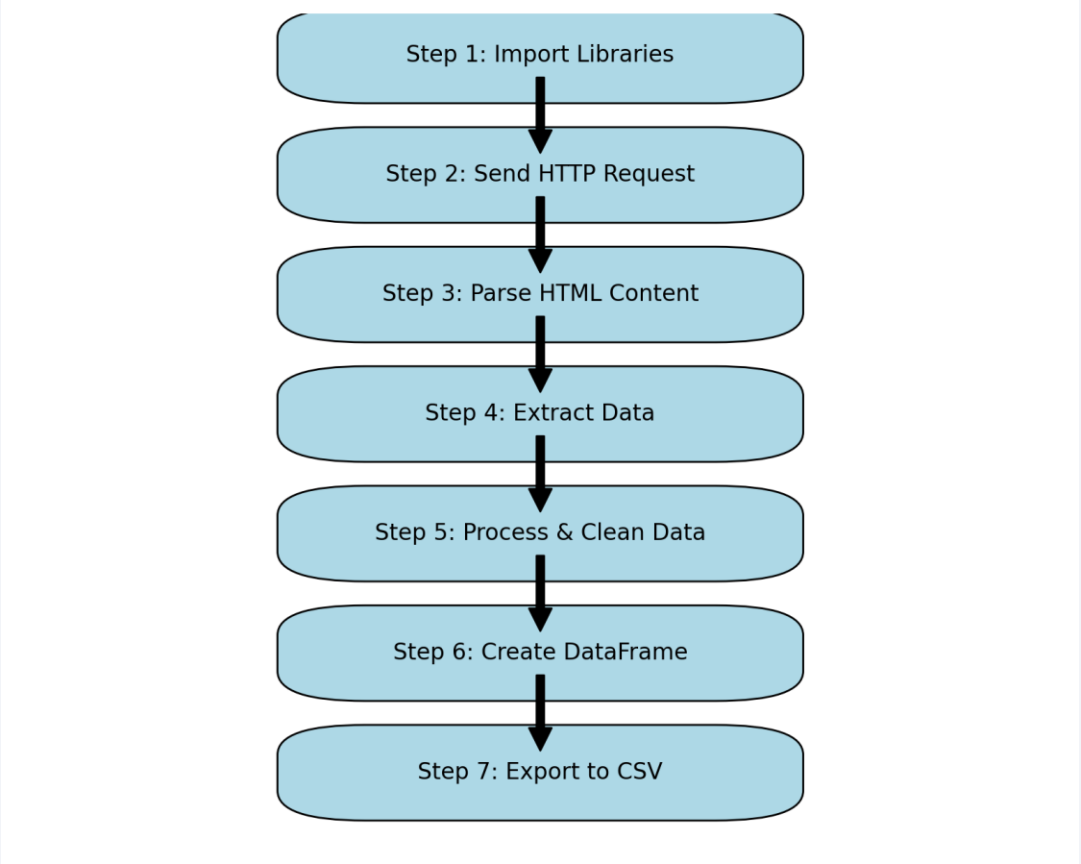
- Made a GET request to the SpaceX API
- Converted JSON to a pandas DataFrame for analysis
- Handled missing values in columns
- Calculated statistics like mean payload mass
- Exported the cleaned data as CSV for further use
- Data Collection GitHub - SpaceX API



Data Collection - Scraping

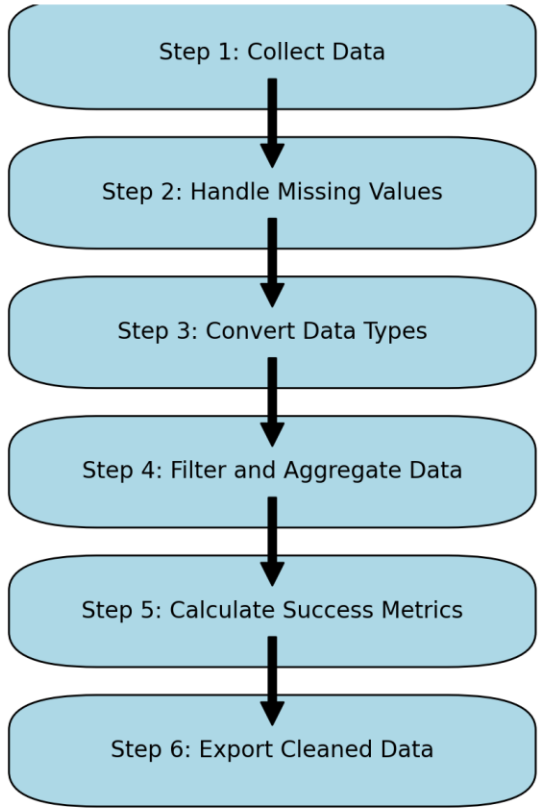
- Sent HTTP request to retrieve HTML content
- Parsed HTML content using BeautifulSoup
- Extracted table
- Created DataFrame from web-scraped data
- Exported to CSV for further analysis

- Data Collection GitHub - Scraping



Data Wrangling

- Handled missing data by replacing with mean
- Converted date formats and ensuring numeric consistency
- Filtered successful landings and payload mass
- Exported the final cleaned dataset



- Data Wrangling GitHub

EDA with Data Visualization

- **Summary of Charts in the Dataset Analysis:**
  - **Distribution Plots**: These were used to show the frequency and spread of different features within the dataset, providing insight into the data's overall shape and any potential outliers.
  - **Correlation Heatmap**: This chart highlighted the relationships between different variables in the dataset, allowing us to see which features are strongly or weakly correlated.
  - **Bar Charts**: These were employed to compare categorical data, showing the count or proportion of observations for various categories.
  - **Box Plots**: Used to visualize the spread and quartiles of the data, helping to identify the range, median, and potential outliers in numerical features.

- EDA with Data Viz GitHub

EDA with SQL

Summary of Queries Used

- Selected distinct values of the "Launch_Site" from the SPACEXTABLE.
- Retrieved the "Launch_Site" where the value started with "CCA" and limited the results to 5.
- Selected the "Customer" and summed the "PAYLOAD_MASS__KG_" for rows where the "Customer" contained "NASA (CRS)".
- Calculated the average "PAYLOAD_MASS__KG_" for entries where "Booster_Version" matched "F9 v1.1".
- Retrieved the earliest "Date" and "Landing_Outcome" for successful landings on a ground pad.

EDA With SQL GitHub

Build an Interactive Map with Folium

Folium Map Objects Created:

- **Circles**:
  - Added at NASA Johnson Space Center and launch sites (radius 1000m).
  - Popups display site names.
  - **Purpose**: Visual identification of key locations.
- **Markers**:
  - NASA JSC marker with text icon.
  - Markers at launch sites with custom icons.
  - Distance markers at the closest:
    - Coastline, Highway, Railroad, City (showing distances from the launch site).
  - **Purpose**: Display proximity to infrastructure and visually highlight launch sites.
- **Polylines**:
  - Lines connecting the launch site to the nearest:
    - Coastline, Highway, Railroad, City.
  - **Purpose**: Visualize paths and distances for planning and transportation analysis.
- Folium GitHub

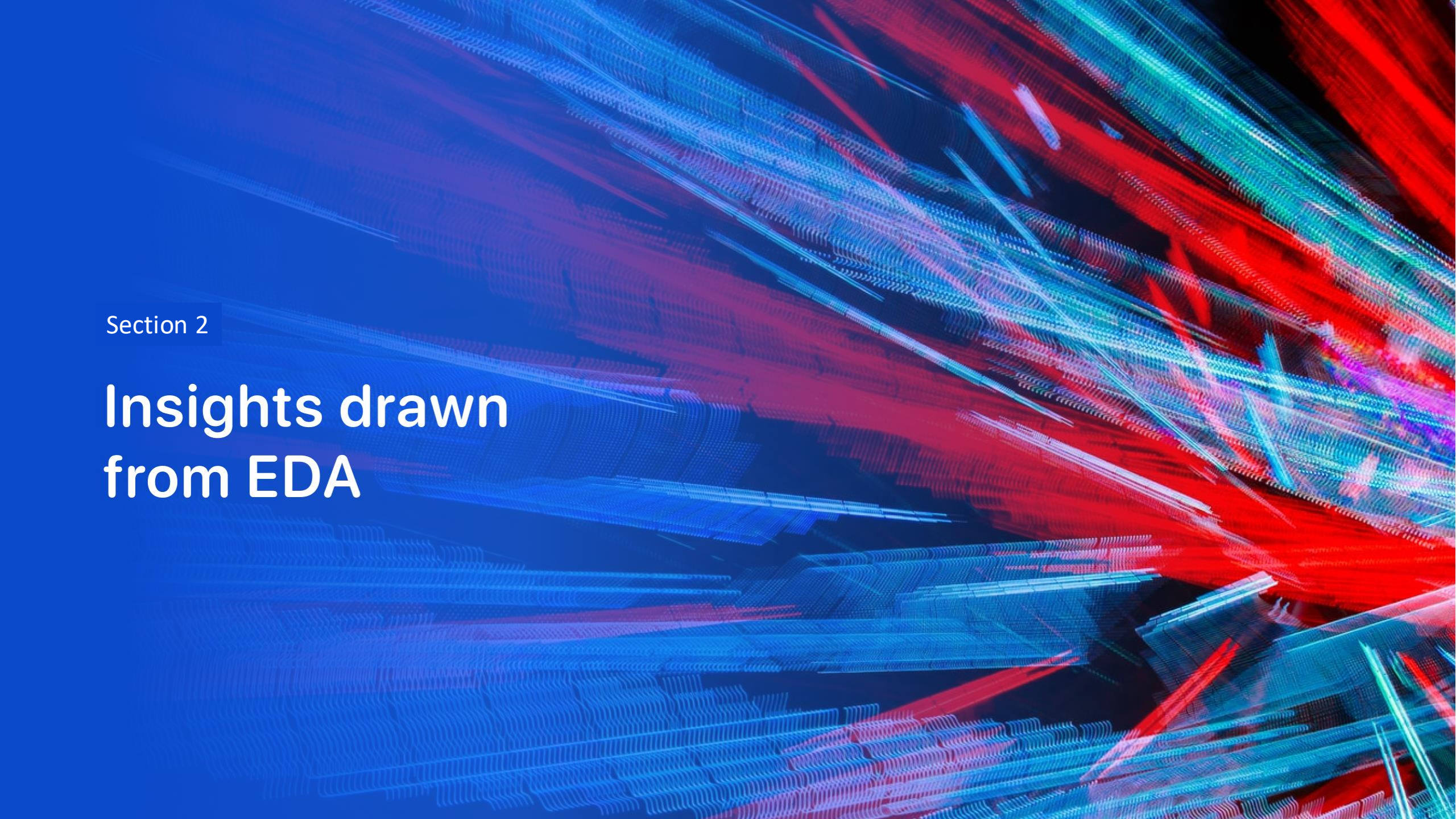
Build a Dashboard with Plotly Dash

- Dashboard Elements:
- **Dropdown Menu**:
  - A dropdown allowed users to select a launch site or view all sites.
  - **Purpose**: Filtered data by specific launch sites or showed all data.
- **Pie Chart**:
  - Displayed the success count of launches for all sites or the selected site.
  - **Purpose**: Provided an overview of launch success rates.
- **Range Slider**:
  - Allowed users to filter data by payload mass (kg).
  - **Purpose**: Let users explore the impact of payload mass on launch success.
- **Scatter Plot**:
  - Showed the relationship between payload mass and launch success.
  - **Purpose**: Visualized how payload mass influenced launch outcomes.
  - Interactions:
- **Dropdown**: Updated the pie chart based on the selected launch site.
- **Range Slider**: Updated the scatter plot based on the chosen payload range.
- Plotly Dash GitHub

Predictive Analysis (Classification)

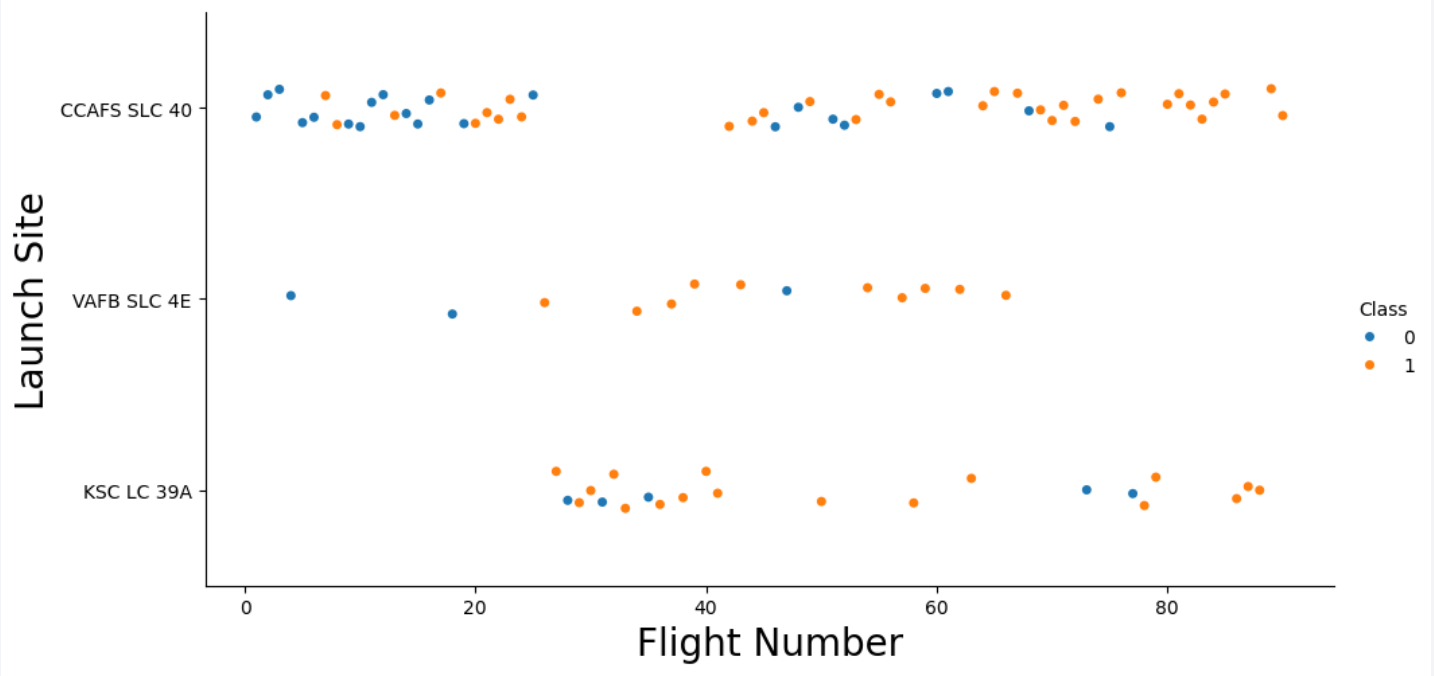
1. **Data Preparation**: Collected and cleaned data on SpaceX launches, including features like payload mass, orbit, and landing outcomes. The data was standardized for model compatibility.
2. **Model Building**: Multiple models were built, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).
3. **Evaluation and Tuning**: GridSearchCV was used to perform hyperparameter tuning for each model. The performance was evaluated using accuracy and confusion matrices.
4. **Best Model**: Models were compared using test data, and the best model was selected based on the highest accuracy after tuning.

Predictive Analysis GitHub


Section 2
Insights drawn from EDA

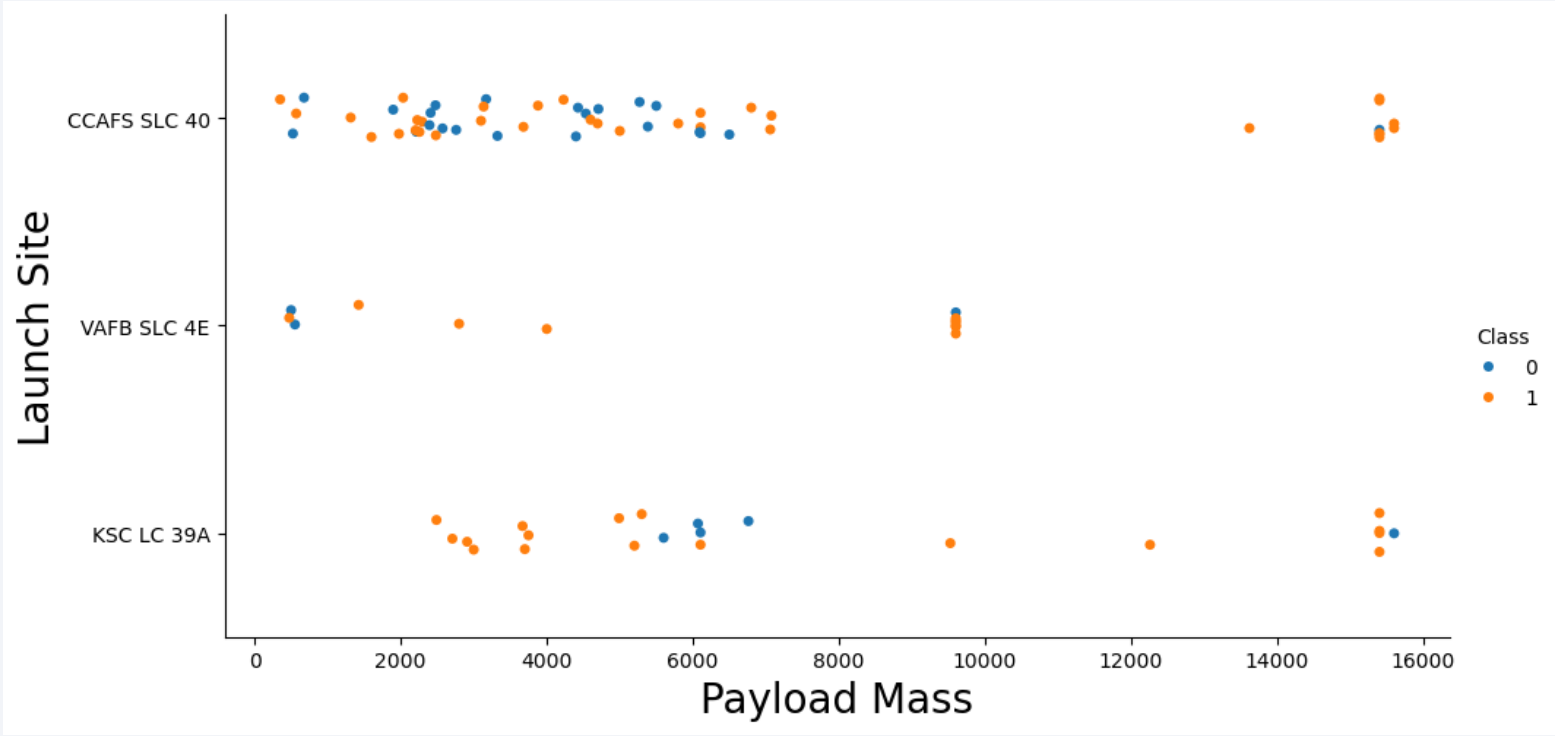
Flight Number vs. Launch Site

- This scatter plot shows the relationship between Payload Mass and Launch Sites for SpaceX launches. The color of the points represents the launch outcome, where blue points indicate unsuccessful landings, and orange points represent successful landings. It helps visualize how payload mass and launch site affect the success of launches.



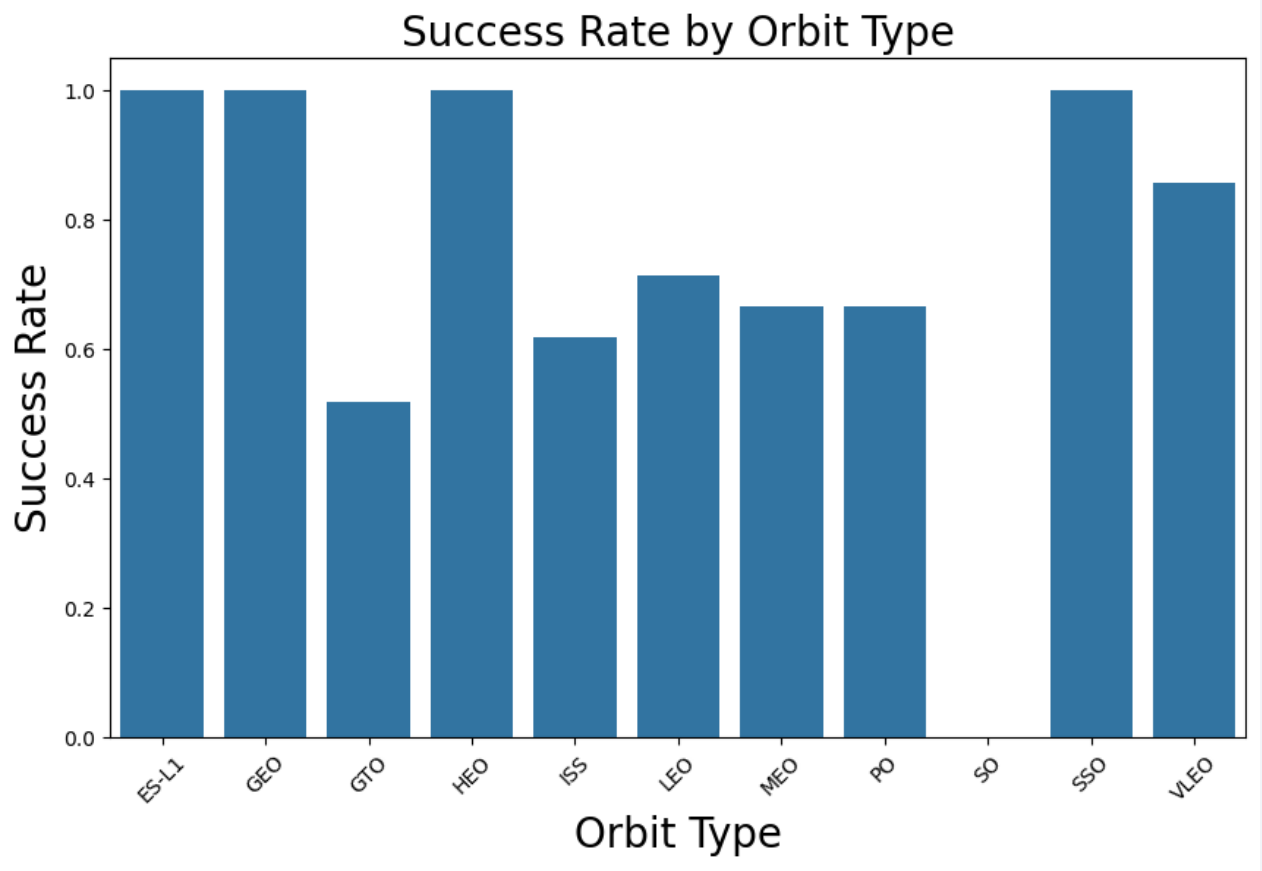
Payload vs. Launch Site

- This scatter plot shows the relationship between Payload Mass and Launch Sites, with the points colored by launch success. Blue dots represent unsuccessful landings, while orange dots indicate successful landings, providing insight into how launch site and payload mass correlate with launch outcomes.



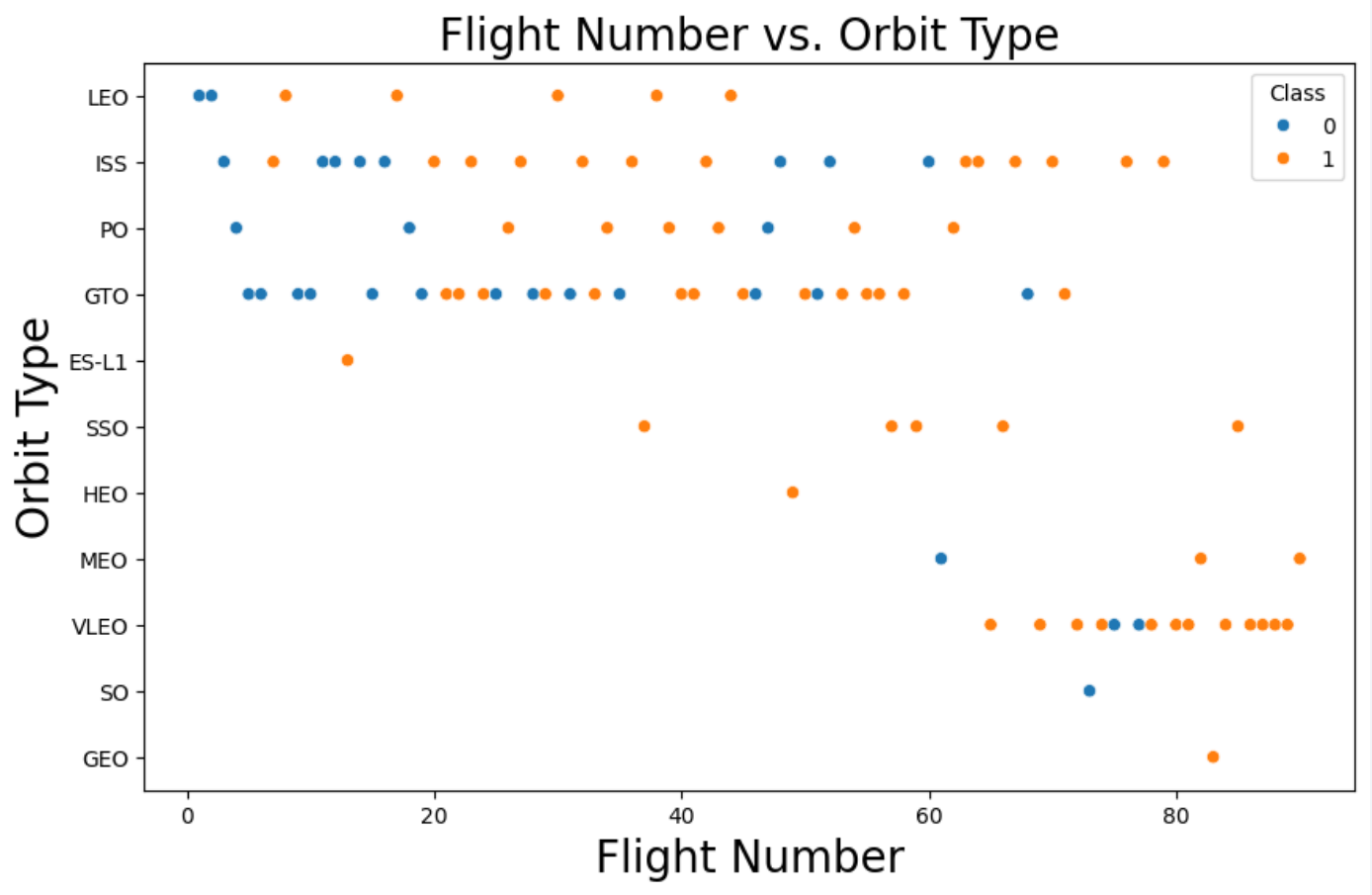
Success Rate vs. Orbit Type

- This bar chart displays the Success Rate of launches for different Orbit Types. The chart highlights that certain orbits like ES-L1, GEO, HEO, and SSO have a near-perfect success rate, while orbits like GTO show lower success rates, indicating varying reliability based on orbit type.



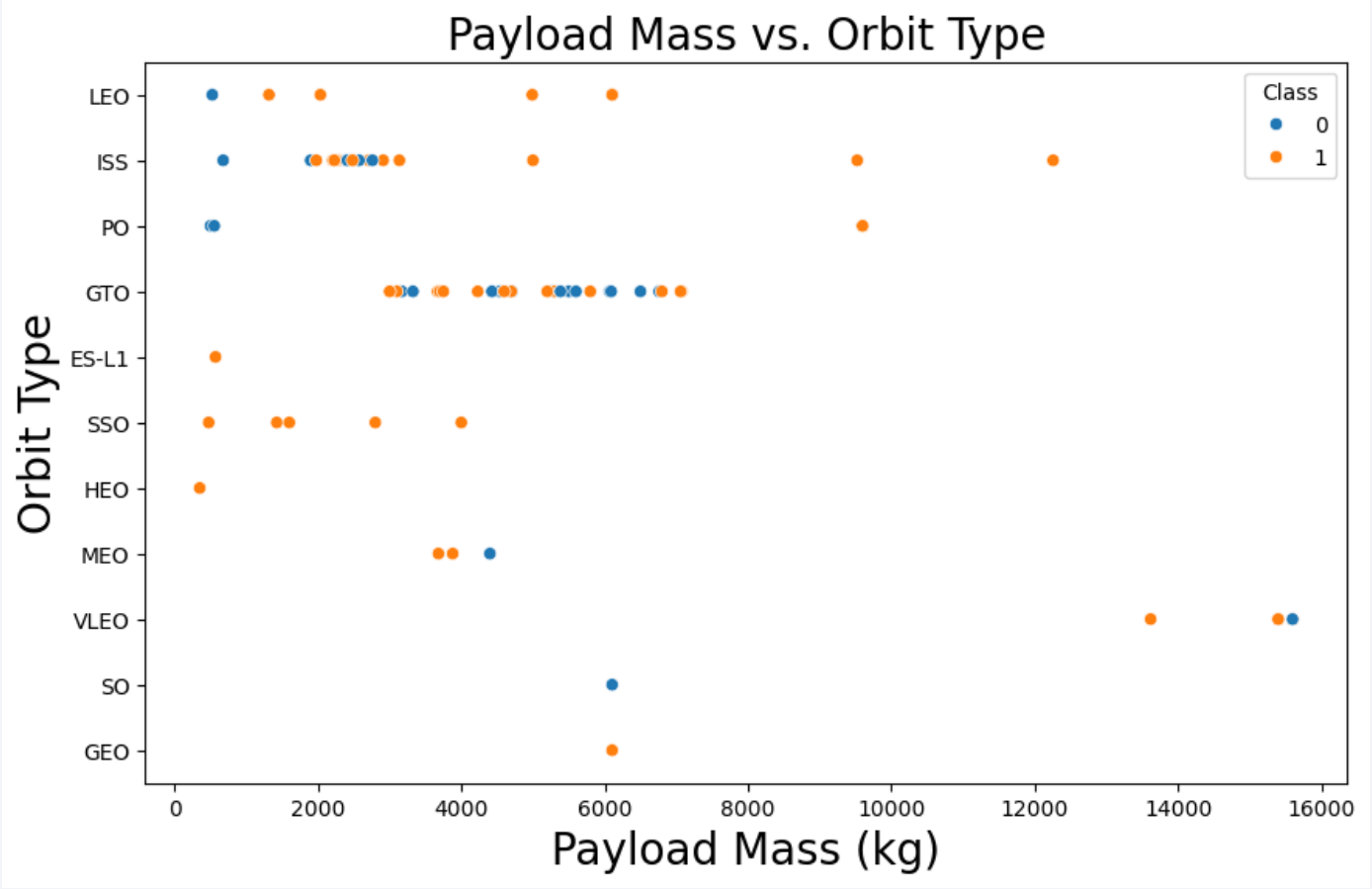
Flight Number vs. Orbit Type

- This scatter plot shows the relationship between Flight Number and Orbit Type, with each point representing a flight. The color of the points represents the launch outcome, where blue points indicate unsuccessful landings and orange points represent successful landings. It illustrates how different orbit types and flight numbers relate to the success of SpaceX launches.



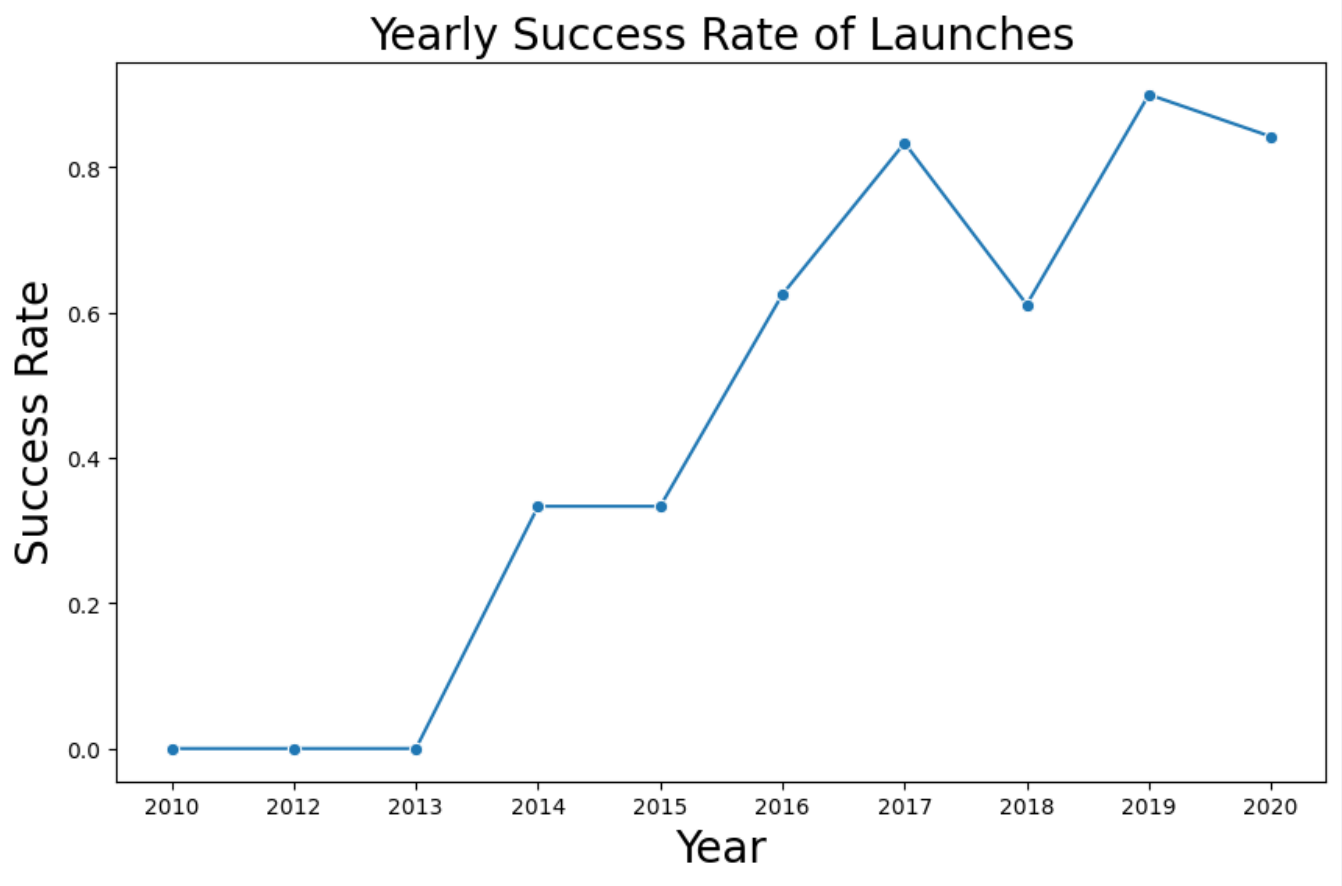
Payload vs. Orbit Type

- This scatter plot shows the relationship between Payload Mass (kg) and Orbit Type, with each point colored based on the launch outcome, where blue points indicate unsuccessful landings and orange points indicate successful landings. It illustrates how different payload masses and orbit types impact the success of SpaceX launches.



Launch Success Yearly Trend

- This scatter plot shows the relationship between Payload Mass (kg) and Orbit Type, with each point colored based on the launch outcome, where blue points indicate unsuccessful landings and orange points indicate successful landings. It illustrates how different payload masses and orbit types impact the success of SpaceX launches.



All Launch Site Names

- Query used

```
query = """
SELECT
    DISTINCT "Launch_Site"
FROM
    SPACEXTABLE
"""
df_result = pd.read_sql_query(query, con)
df_result
```

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

- This SQL query selects distinct values of the "Launch_Site" column from the SPACEXTABLE table. The purpose is to retrieve a list of unique launch sites from the dataset, avoiding any

Launch Site Names Begin with 'CCA'

- Query used

```
query = """
SELECT
    "Launch_Site"
FROM
    SPACEXTABLE
WHERE
    "Launch_Site" LIKE "CCA%"
Limit 5
"""
df_result = pd.read_sql_query(query, con)
df_result
```

	Launch_Site
0	CCAFS LC-40
1	CCAFS LC-40
2	CCAFS LC-40
3	CCAFS LC-40
4	CCAFS LC-40

Total Payload Mass

- Query used

```
query = """
SELECT
    "Customer",
    SUM("PAYLOAD_MASS__KG_")
FROM
    SPACEXTABLE
WHERE
    "Customer" LIKE "NASA (CRS)"
"""


df_result = pd.read_sql_query(query, con)

df_result
```

Customer	SUM("PAYLOAD_MASS__KG_")
NASA (CRS)	45596

- This query selects the "Customer" and the total sum of "PAYLOAD_MASS__KG_" from the SPACEXTABLE where the customer is NASA under the Commercial Resupply Services (CRS) program. It calculates the total payload mass for NASA CRS launches.

Average Payload Mass by F9 v1.1

- Query used

```
query = """
SELECT
    "Booster_Version",
    AVG("PAYLOAD_MASS__KG_")
FROM
    SPACEXTABLE
WHERE
    "Booster_Version" LIKE "F9 v1.1"
"""
df_result = pd.read_sql_query(query, con)

df_result
```

Booster_Version	AVG("PAYLOAD_MASS__KG_")
F9 v1.1	2928.4

- This query selects the "Booster_Version" and calculates the average "PAYLOAD_MASS__KG_" from the SPACEXTABLE where the "Booster_Version" is "F9 v1.1". It provides the average payload mass for launches using the Falcon 9 v1.1 booster version.

First Successful Ground Landing Date

- Query used

```
query = """
SELECT
    "Date",
    "Landing_Outcome",
    MIN("Date")
FROM
    SPACEXTABLE
WHERE
    "Landing_Outcome" LIKE "Success (ground pad)"
"""

df_result = pd.read_sql_query(query, con)

df_result
```

Date	Landing_Outcome	MIN("Date")
2015-12-22	Success (ground pad)	2015-12-22

- This query selects the "Date" and "Landing_Outcome" from the SPACEXTABLE and retrieves the earliest launch date where the landing outcome was a "Success (ground pad)." It finds the first successful ground pad landing in the dataset.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query used

```
query = """
SELECT
    "Booster_Version",
    "PAYLOAD_MASS__KG_",
    "Landing_Outcome"
    FROM
    SPACEXTABLE
WHERE
    "PAYLOAD_MASS__KG_" >= 4000 and "PAYLOAD_MASS__KG_" <= 6000
    and "Landing_Outcome" LIKE "Success (ground pad)"
"""

df_result = pd.read_sql_query(query, con)

df_result
```

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1032.1	5300	Success (ground pad)
F9 B4 B1040.1	4990	Success (ground pad)
F9 B4 B1043.1	5000	Success (ground pad)

- This query selects the "Booster_Version", "PAYLOAD_MASS__KG_", and "Landing_Outcome" from the SPACEXTABLE, filtering the results to only include rows where the payload mass is between 4000 and 6000 kg, and the landing outcome was a "Success (ground pad)." It retrieves information about successful ground pad landings with payloads in this weight range.

Total Number of Successful and Failure Mission Outcomes

- Query used

```
query = """
SELECT
    "Mission_Outcome",
    COUNT("Mission_Outcome")
    FROM
    SPACEXTABLE
    Group By "Mission_Outcome"

"""
df_result = pd.read_sql_query(query, con)

df_result
```

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

- This query selects the "Mission_Outcome" and counts the number of occurrences for each distinct mission outcome from the SPACEXTABLE. It groups the results by "Mission_Outcome" to provide the total number of missions for each possible outcome.

Boosters Carried Maximum Payload

- Query used

```
query = """
SELECT
    "Mission_Outcome",
    COUNT("Mission_Outcome")
    FROM
    SPACEXTABLE
    Group By
        "Mission_Outcome"
"""

df_result = pd.read_sql_query(query, con)

df_result
```

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

- This query retrieves the "Mission_Outcome" and counts how many times each distinct mission outcome appears in the SPACEXTABLE. It groups the results by "Mission_Outcome" to show the total number of missions for each specific outcome.

2015 Launch Records

- Query used

```
query = """
SELECT
    substr(Date, 6, 2) AS month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM
    SPACEXTABLE
WHERE
    substr(Date, 1, 4) = '2015'
    AND "Landing_Outcome" LIKE 'Failure (drone ship)';

"""
df_result = pd.read_sql_query(query, con)

df_result
```

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

- This query selects the month (extracted from the "Date"), "Landing_Outcome", "Booster_Version", and "Launch_Site" from the SPACEXTABLE for records where the year is 2015 and the landing outcome was a "Failure (drone ship)." It filters the data to show details of failed drone ship landings in 2015.

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

- Query used

```
query = """
SELECT
    "Landing_Outcome",
    COUNT(*) as outcome_count
FROM
    SPACEXTABLE
WHERE
    Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    "Landing_Outcome"
ORDER BY
    outcome_count DESC;
"""
df_result = pd.read_sql_query(query, con)

df_result
```

Landing_Outcome	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

- This query retrieves the "Landing_Outcome" and the count of occurrences (labeled as "outcome_count") from the SPACEXTABLE for launches that occurred between June 4, 2010, and March 20, 2017. It groups the results by "Landing_Outcome" and orders them by the count of outcomes in descending order, showing the most frequent landing outcomes during this period.

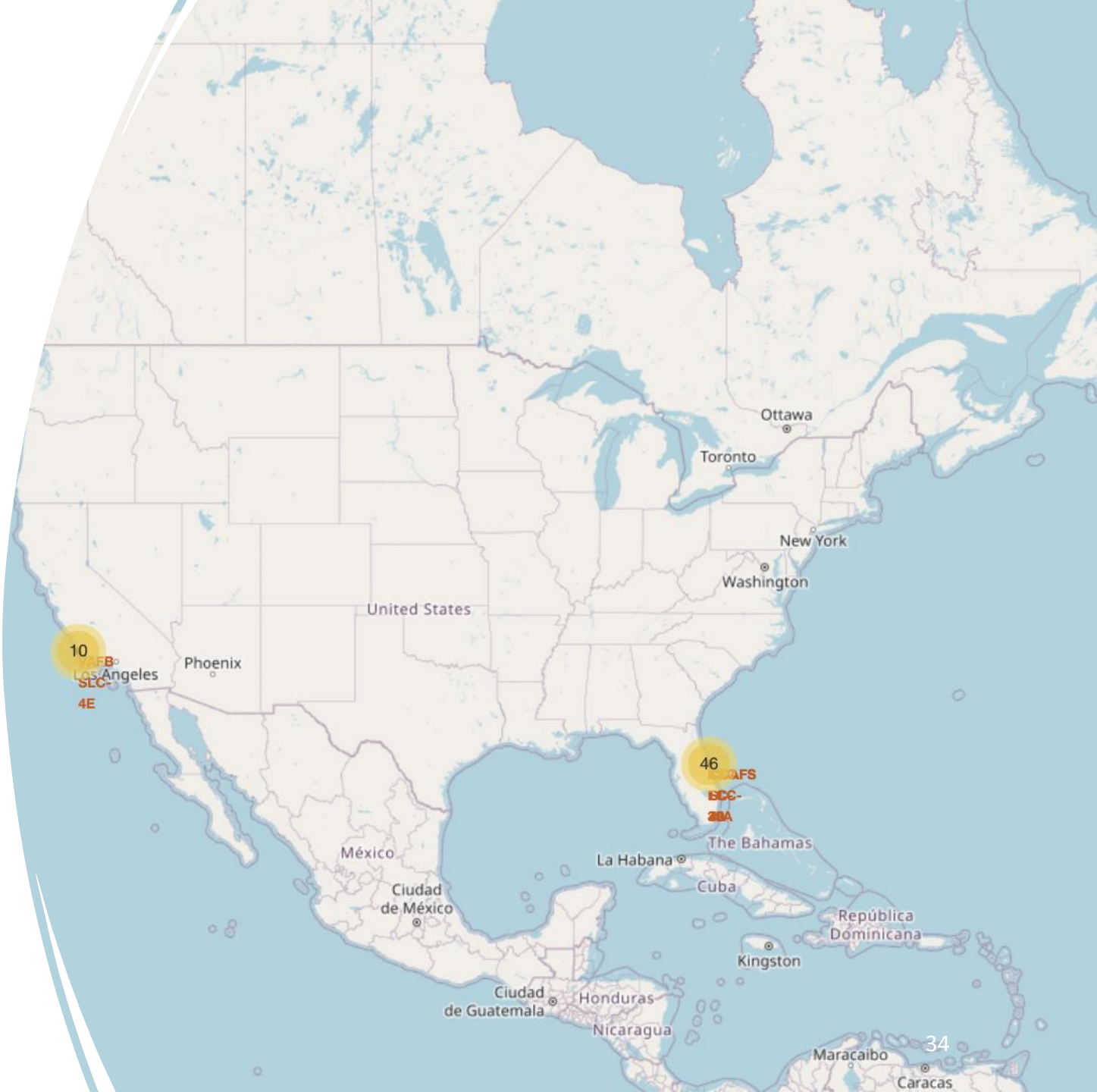
Section 3

Launch Sites Proximities Analysis

Launch Sites With Folium

- The image to the right shows the locations of all the Falcon-9 launches.

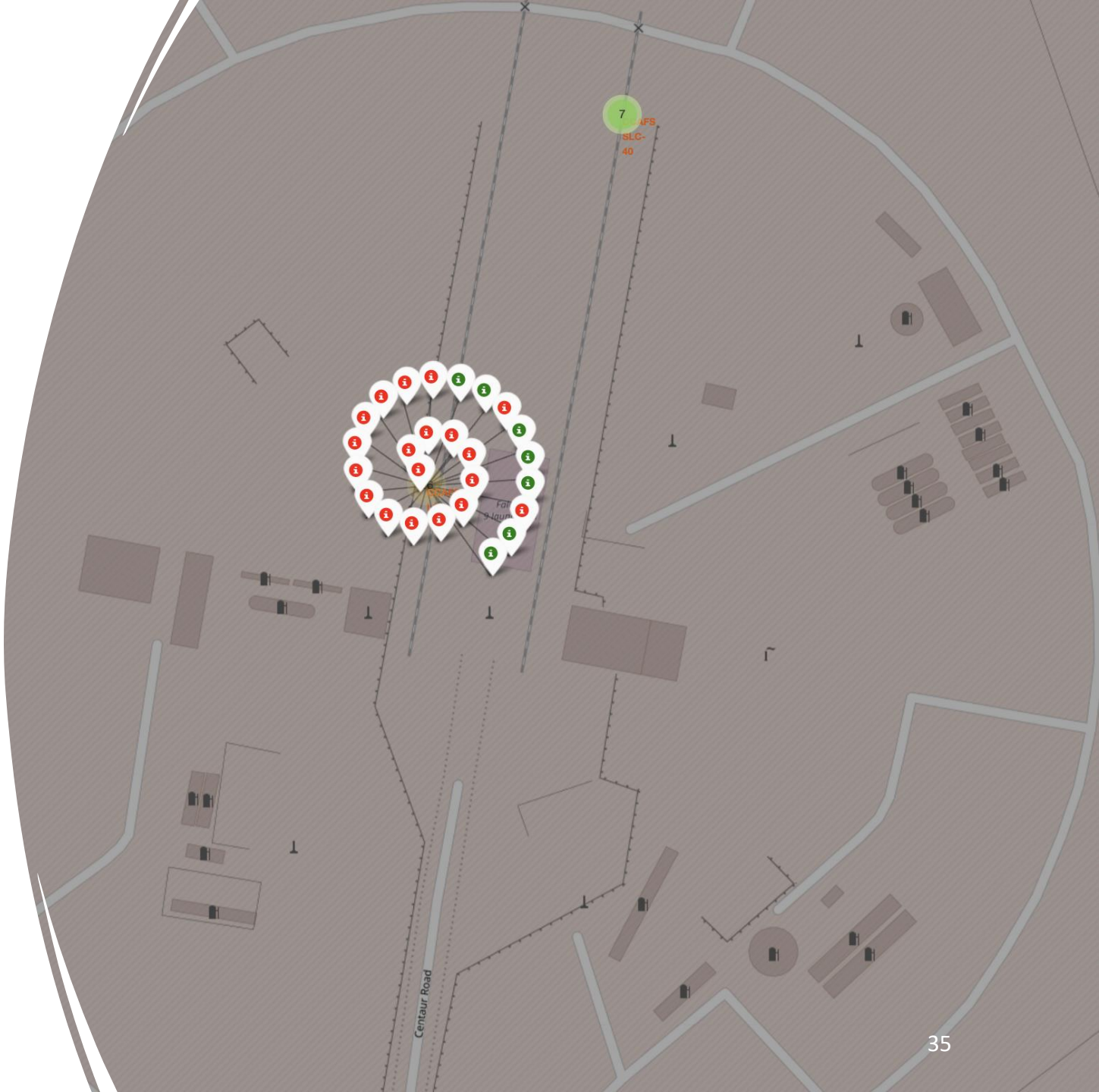
- From this screenshot we can see that 46 launches took place in Florida while 10 took place in California



Launch Outcomes

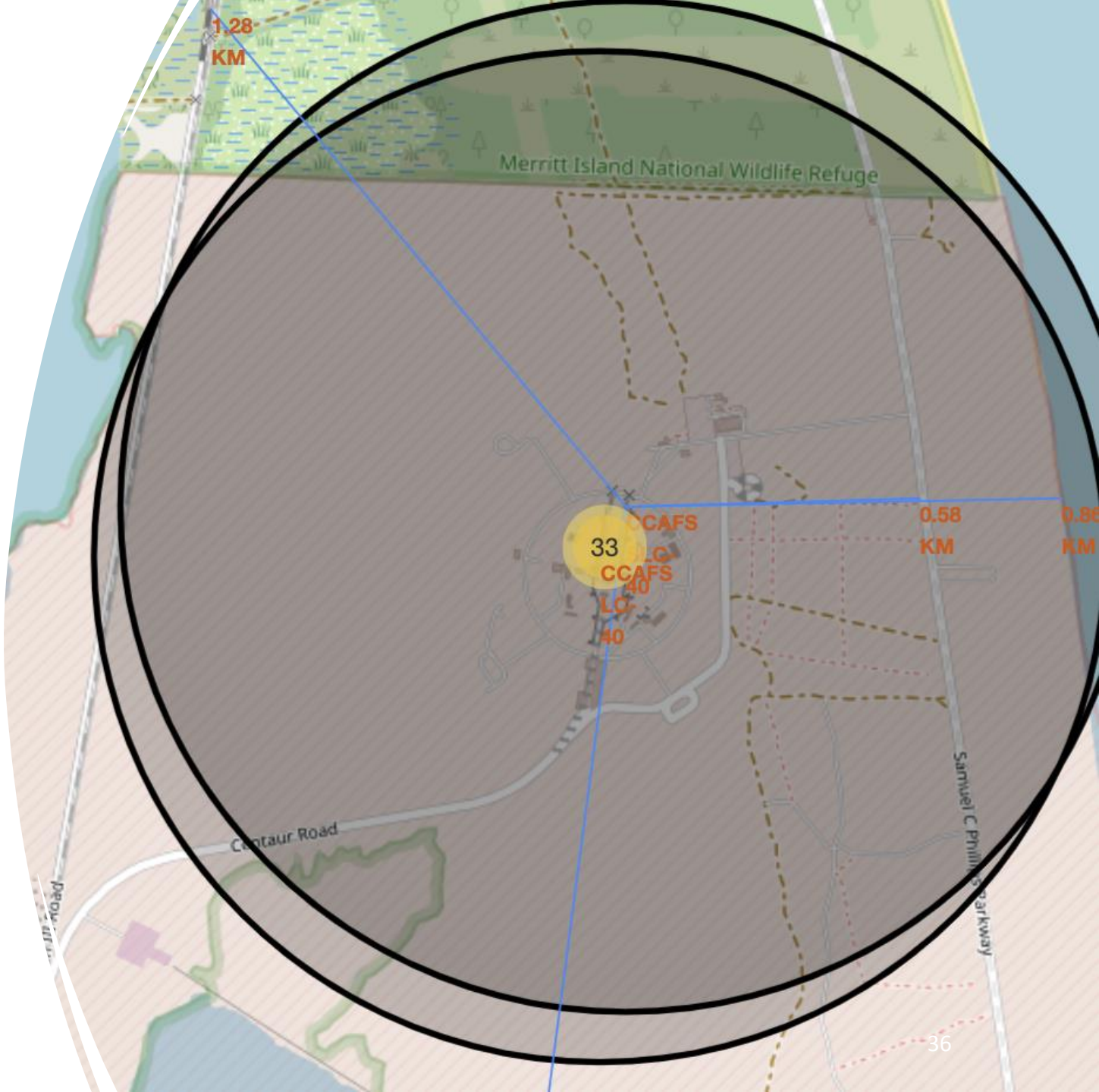
- When we zoom in to a launch location we see green or red icons.

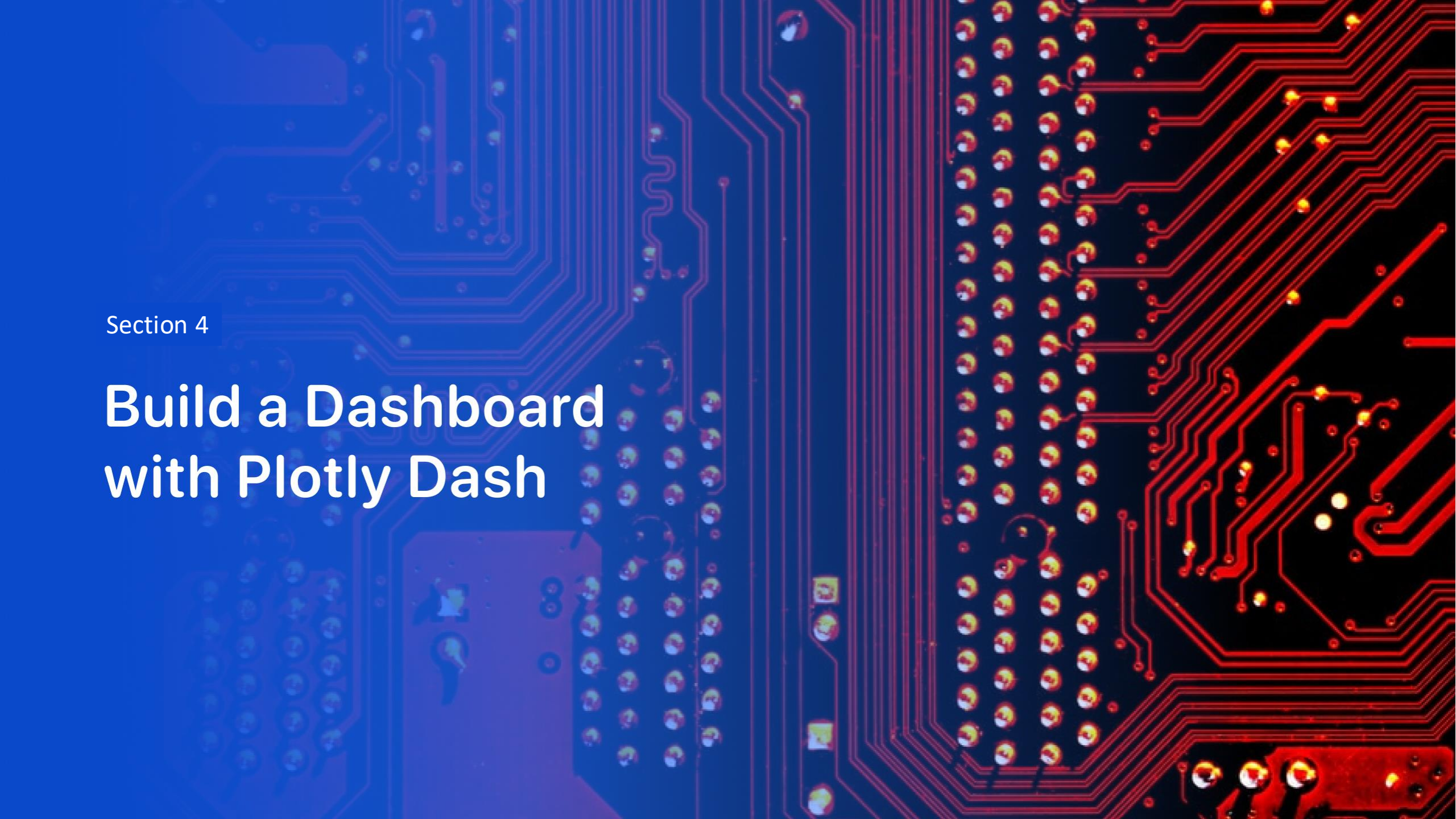
- Each icon represents a launch and if the icon is green that means the launch was successful, red unsuccessful



Poly Lines

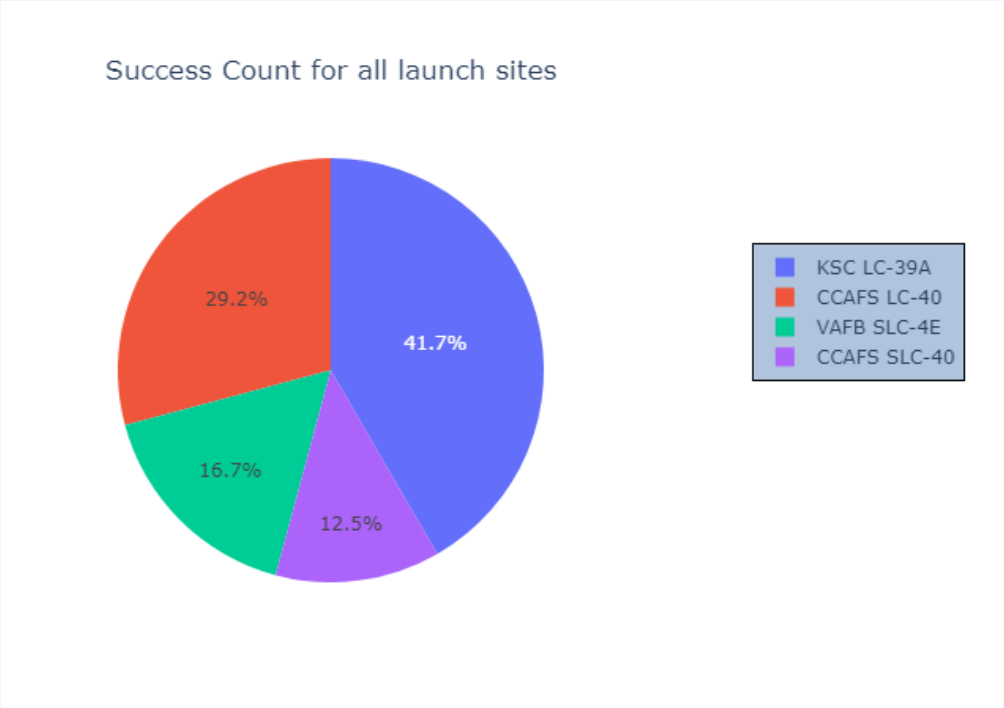
- As seen in the screenshot to the right, the launch sites have polylines to the closest highway coastline and railway.
- The closest highway is 0.58 km
- The closest railway is 1.28 km
- The closets coastline is 0.86 km




Section 4
Build a Dashboard with Plotly Dash

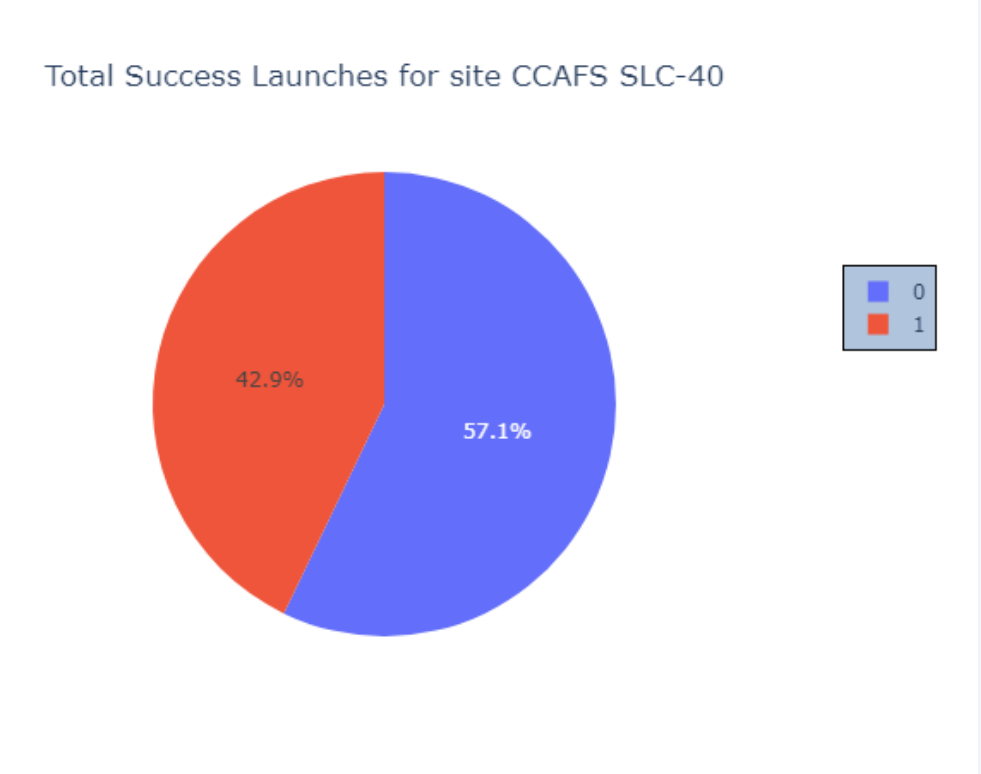
Success Count by Launch Site

- When all sites are selected for the dropdown the pie chart to the right is produced.
- The pie chart displays the distribution of successful Falcon 9 first stage landing outcomes between the different launch sites.
- With 41.7%, the greatest share of successful Falcon 9 first stage landing outcomes occurred at KSC LC-39A.



Launch Site with Highest Success Ratio

- This pie chart shows the first stage landing success rate at the CCAFS SLC-40 launch site.
- The successes are indicated by the red portion of the chart while the blue is the failures.
- CCAFS SLC-40 had the highest first stage landing success rate at 42.9%



Payload vs. Launch Outcome Scatter Plot

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

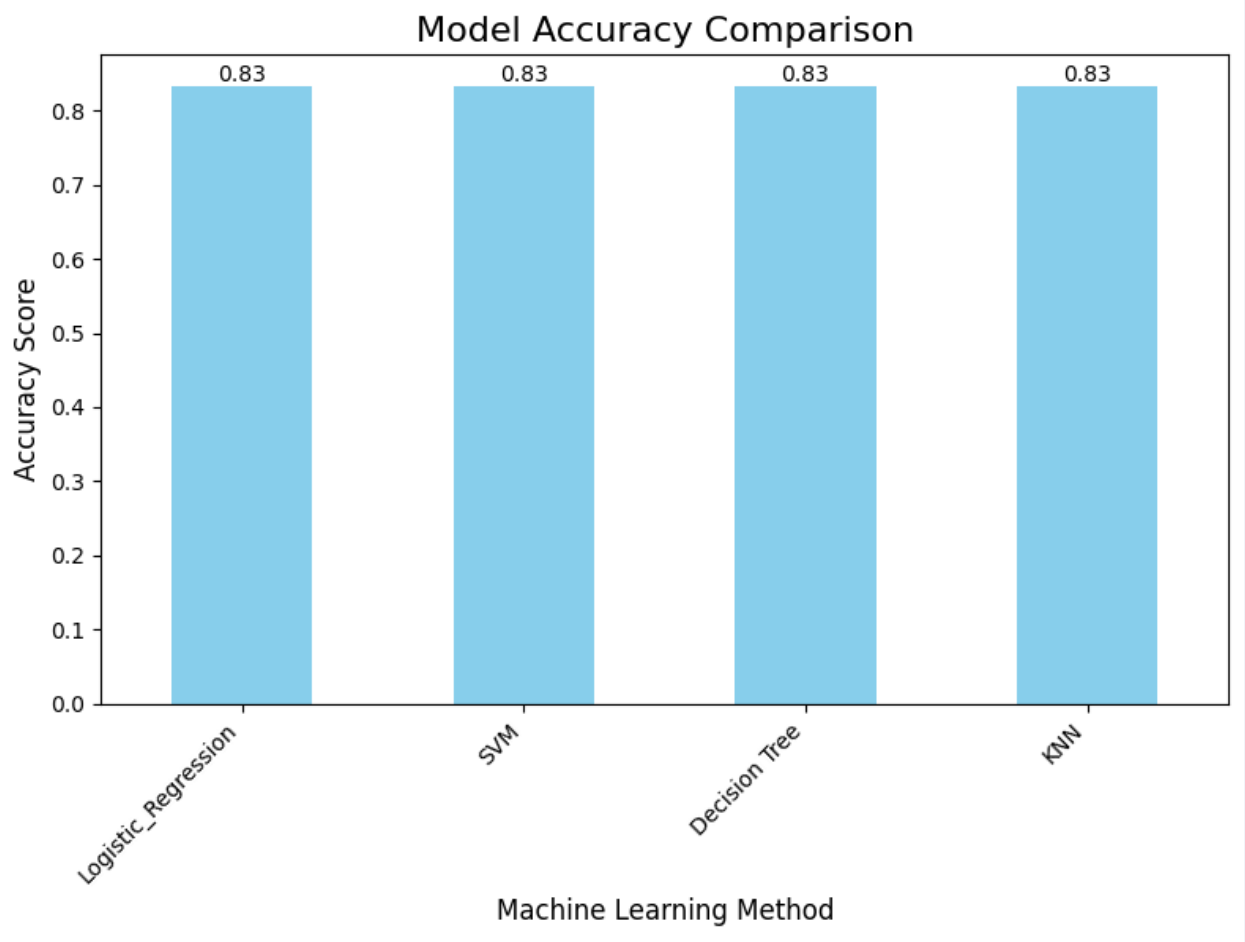
- The success rate appears to be highest when the weight is between 2000 and 3000 kg's.

Section 5

Predictive Analysis (Classification)

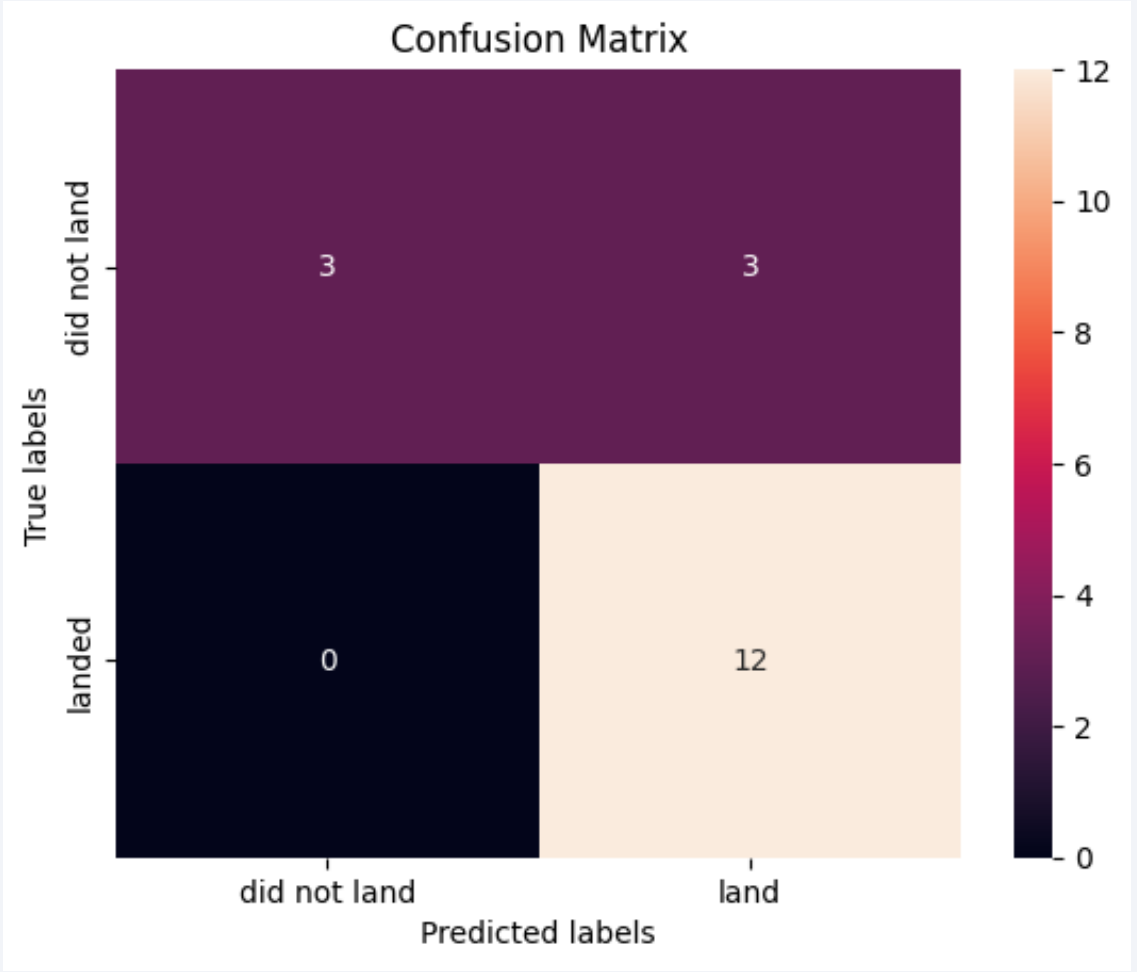
Classification Accuracy

- All of the models had the exact same accuracy score.



Confusion Matrix

- The image is a confusion matrix visualizing the performance of a classification model. It shows that the model correctly predicted 12 landings and 3 non-landings while making 3 incorrect predictions for non-landings and none for landings, indicating that the model has a high accuracy for landing predictions.



Conclusions

- SpaceX doesn't have a flawless record for Falcon 9 first stage landing outcomes.
- The success rate of Falcon 9 first stage landings has been improving with more launches.
- Machine learning models can help predict future outcomes of SpaceX Falcon 9 first stage landings.

Thank you!