IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nguyen Do
April 16, 2025

Outline

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

Executive Summary

Methodologies

1. Problem Framing
2. Data Collection
3. Data Wrangling and Cleaning
4. Exploratory Data Analysis
5. Dashboard and Visualization
6. Data Modeling

Introduction

1. Background Overview
   - This project simulates the role of a data scientist at a fictional rocket company, SpaceY, aiming to compete with SpaceX.

2. Business Problems
- Analyze public SpaceX data to predict whether the Falcon 9 rocket first stage will land successfully.
- Using machine learning to build predictive models
- Create dashboards to support launch pricing decisions without relying on traditional rocket science.

Section
1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

1. SPACEX REST API
   - **Tool Used**: requests library in Python
   - **Endpoint Accessed**: https://api.spacexdata.com/v4/launches/past
   - **Data Format Returned**: JSON – list of JSON objects, each representing a single launch
   - **Parsing**:
     - response.json() is used to parse the data
     - pandas.json_normalize() flattens the nested JSON into a DataFrame
2. Web scraping

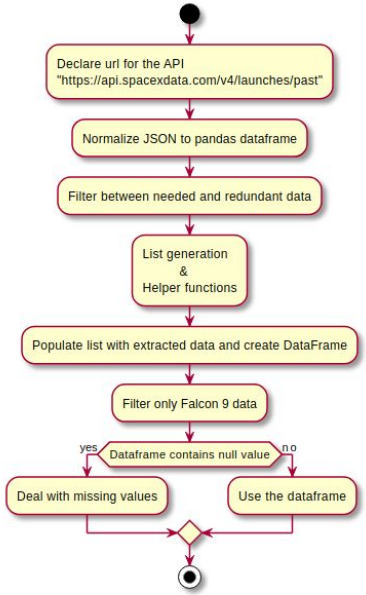
Web Scraping – Wikipedia Falcon 9 Launch Records

- **Tool Used**: BeautifulSoup in Python
- **Source**: HTML tables containing historical Falcon 9 launches
- **Parsing Process**:
  - Extract tables
  - Convert to pandas.DataFrame
  - Clean and merge with the API data

Data Collection – SpaceX API

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

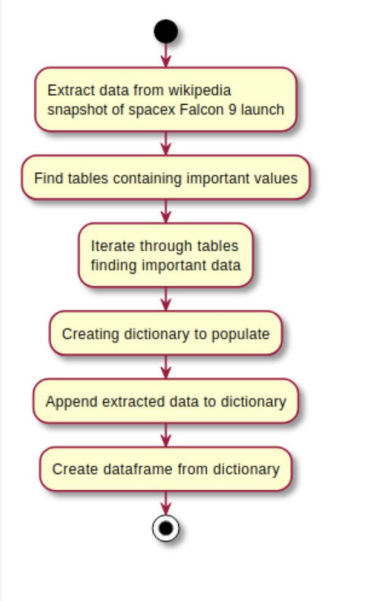
- https://github.com/nguyen010103/RocketLaunch/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

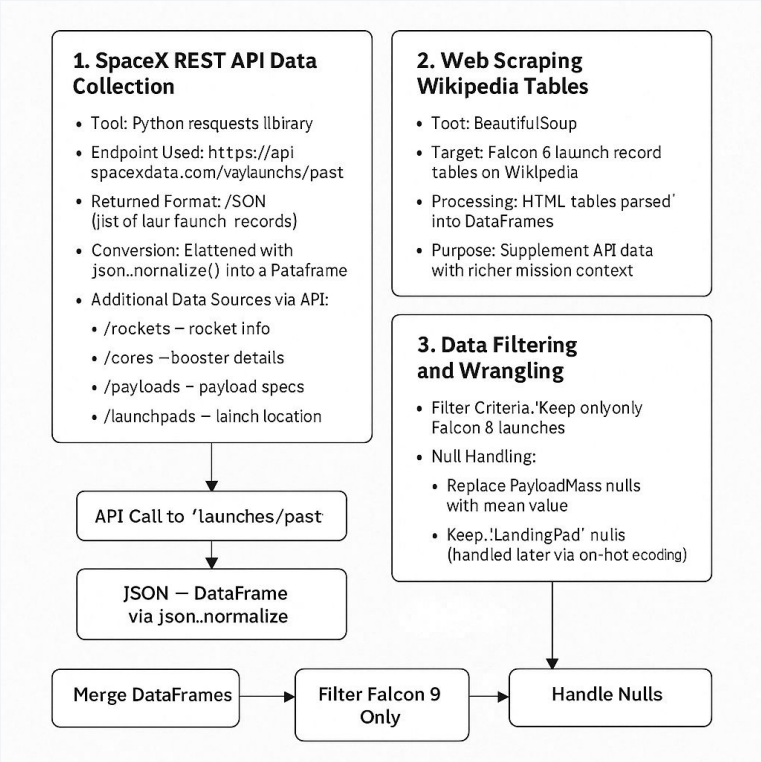
- Present your web scraping process using key phrases and flowcharts

- https://github.com/nguyen010103/RocketLaunch/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

https://github.com/nguyen010103/RocketLaunch/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

The following charts were plotted to explore relationships between rocket launch features and the success of first stage landings (Class):

1. Catplot: Payload Mass vs. Flight Number
    - Why: To investigate how the payload mass correlates with flight number and landing success.
    - Insight: Reveals patterns across mission size, evolution, and outcome.
2. Catplot: Launch Site vs. Flight Number
    - Why: To determine if certain launch sites have higher success rates.
    - Insight: Identifies site-specific success trends.
3. Catplot: Launch Site vs. Payload Mass
    - Why: To see which launch sites handle heavier payloads and how it relates to outcome.
    - Insight: Useful for linking launch capacity with performance.
4. Boxplot: Payload Mass by Orbit Type
    - Why: To observe distribution of payload sizes across different orbit types.
    - Insight: Shows variability and central tendency in mission profiles.
5. Scatter Plot: Flight Number vs. Orbit
    - Why: To assess if orbit type changes as mission experience increases.
    - Insight: Suggests evolution in mission design over time.
6. Boxplot: Launch Site vs. Orbit
    - Why: To understand orbit variety by location.
    - Insight: Indicates specialization or capability of launch sites.
    - https://github.com/nguyen010103/RocketLaunch/blob/main/edadataviz.ipynb

EDA with SQL

Summary of the SQL

Summary of SQL Queries Performed

- **Created a new table** from an existing one, excluding records with null Date values:
  - CREATE TABLE SPACEXTABLE AS SELECT * FROM SPACEXTBL WHERE Date IS NOT NULL
- **Previewed the data**:
  - SELECT * FROM SPACEXTBL LIMIT 5
- **Retrieved unique launch sites**:
  - SELECT DISTINCT(Launch_Site) FROM SPACEXTBL
- **Filtered launches by site prefix (CCA)**:
  - SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
- **Calculated total payload mass** for NASA CRS missions:
  - SELECT SUM(Payload_Mass__kg_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
- **Found average payload mass** for Falcon 9 launches:
  - SELECT AVG(Payload_Mass__kg_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9%'
- **Counted successful landings** (Class = 1):
  - SELECT COUNT(*) FROM SPACEXTBL WHERE Class = 1
- **Grouped missions by booster version**:
  - SELECT Booster_Version, COUNT(*) FROM SPACEXTBL GROUP BY Booster_Version
- **Filtered by specific date range** and success:
  - SELECT * FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' AND Class = 1

https://github.com/nguyen010103/RocketLaunch/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

1. Markers

- **What**: Custom text-based markers using folium.Marker with DivIcon
- **Where**: Placed at each launch site (e.g., NASA Johnson Space Center)
- **Why**: To clearly label each location on the map

2. Circles

- **What**: Colored folium.Circle objects (radius: 1000m, orange color)
- **Where**: Around each launch site
- **Why**: To visually emphasize each launch site's area of interest

3. Lines (Polylines)

- **What**: folium.PolyLine from launch site to nearest coastline
- **Where**: Between site coordinates and coastline points
- **Why**: To visualize proximity and possible trajectory directions

4. Popups

- **What**: folium.Popup attached to circles
- **Why**: To display launch site names interactively

5. Marker Clusters

- **What**: MarkerCluster plugin
- **Why**: To group launch sites together on zoomed-out views for better map readability

6. Mouse Position Tracker

- **What**: MousePosition plugin
- **Why**: To display dynamic latitude and longitude while hovering over the map

https://github.com/nguyen010103/RocketLaunch/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

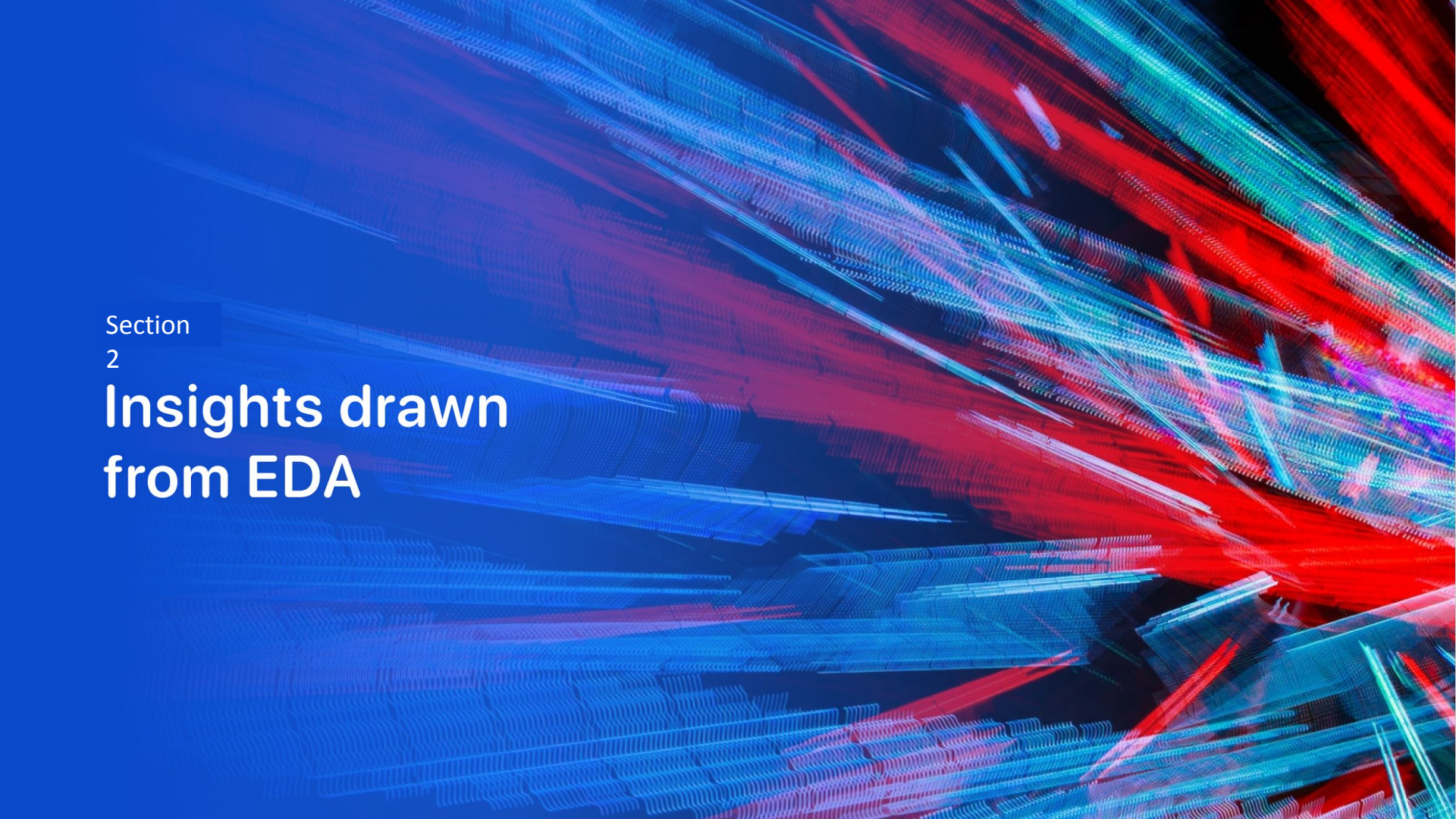
- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

https://github.com/nguyen010103/RocketLaunch/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

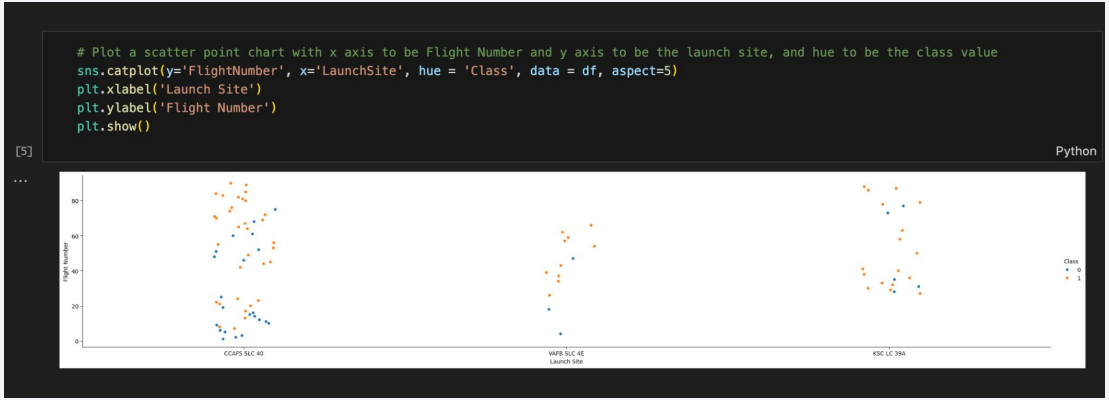


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations

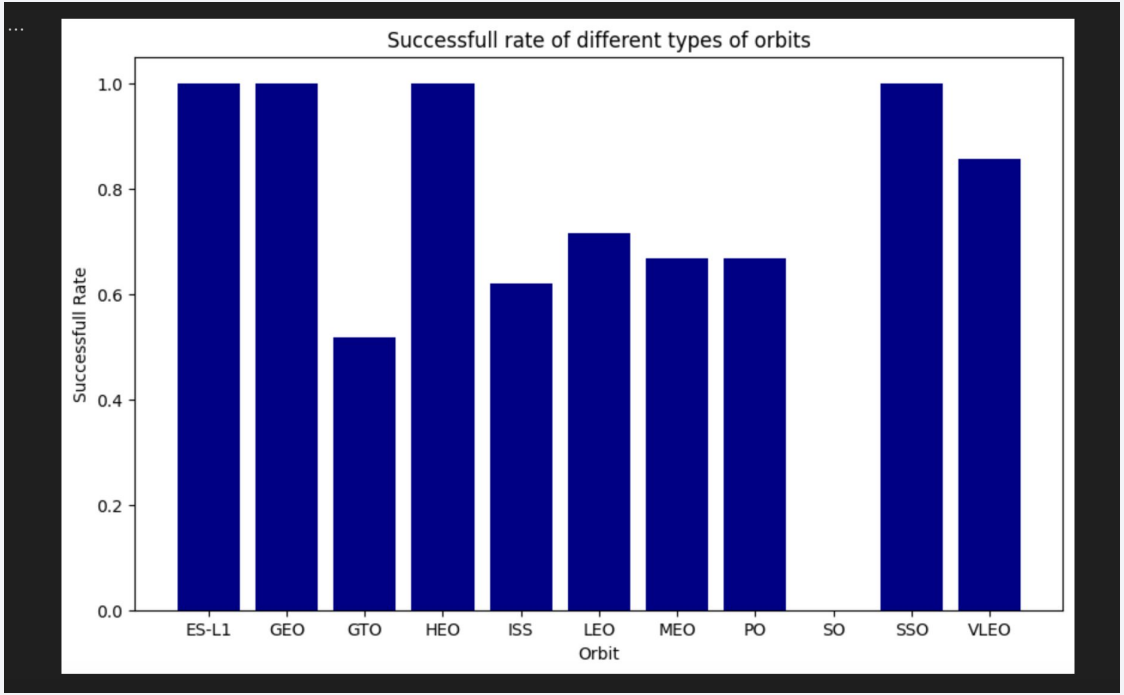


Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations

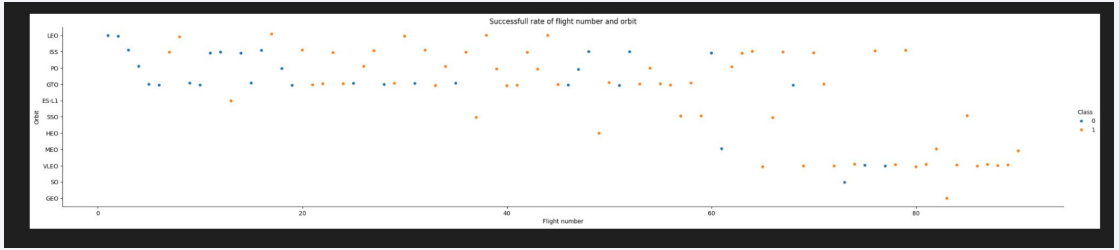
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations



Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations

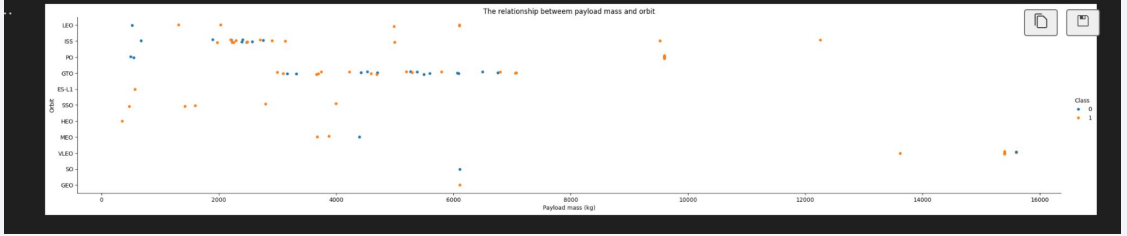


Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

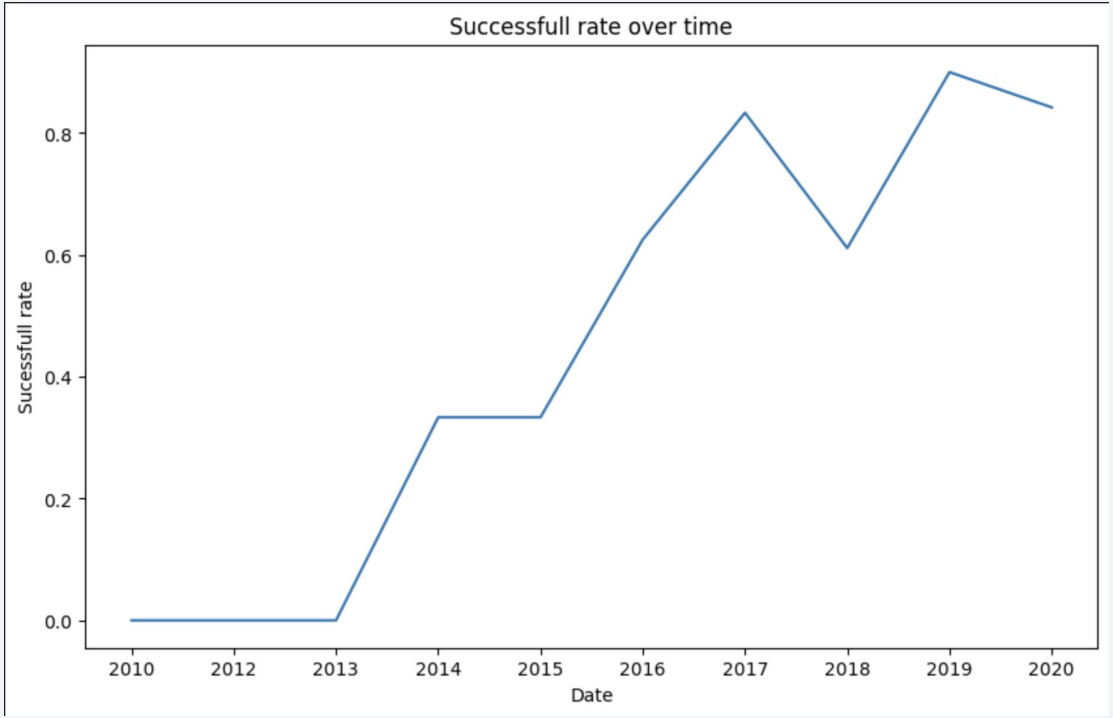
```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(x = 'PayloadMass', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Payload mass (kg)')
plt.ylabel('Orbit')
plt.title('The relationship betweem payload mass and orbit')
plt.show()
```

Python



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- Find the names of the unique launch sites

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

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

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

2015 Launch Records

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

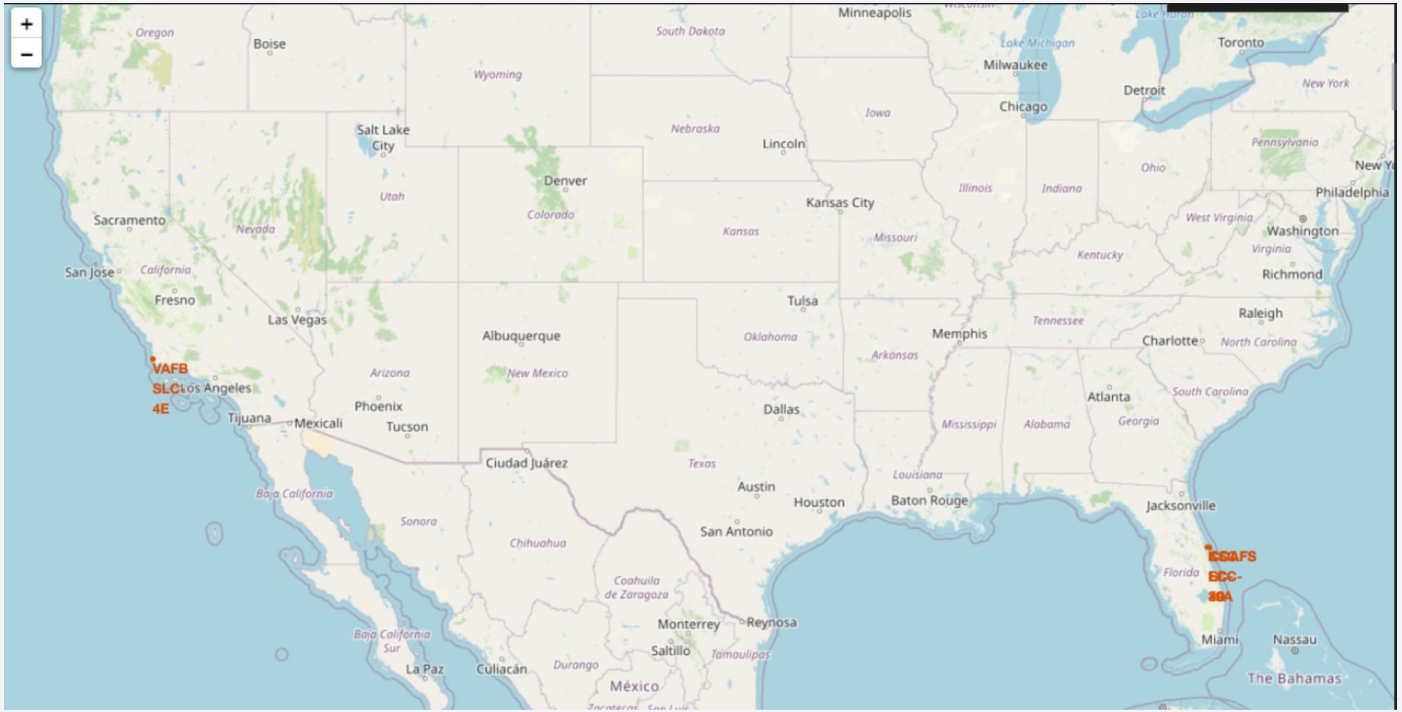
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

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

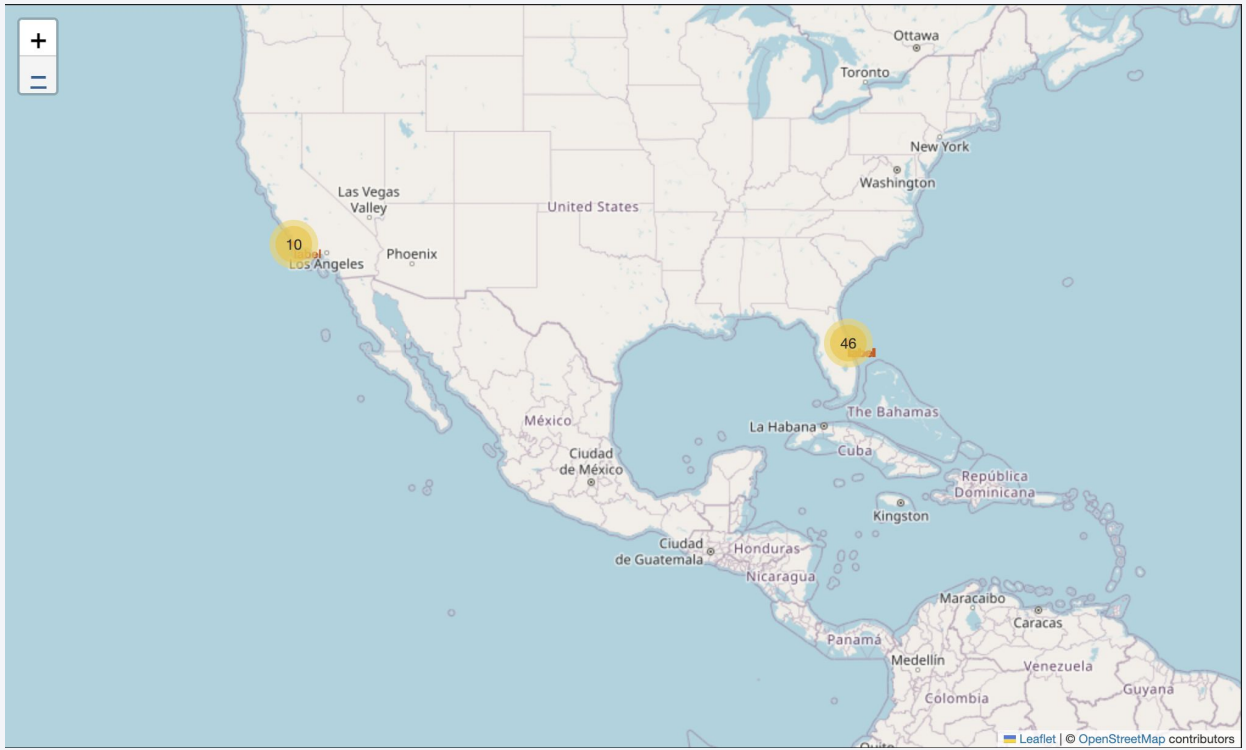
Section 3

Launch Sites Proximities Analysis

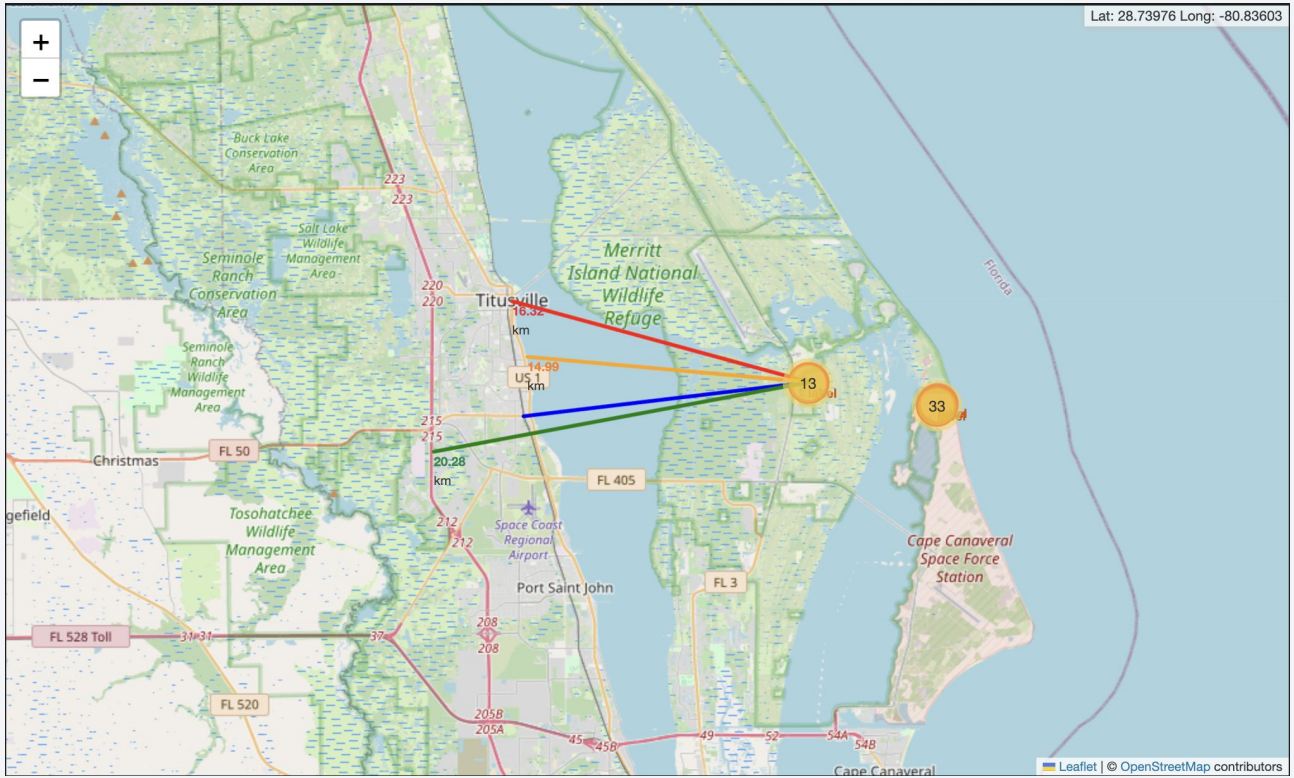
All launches sites in the United State

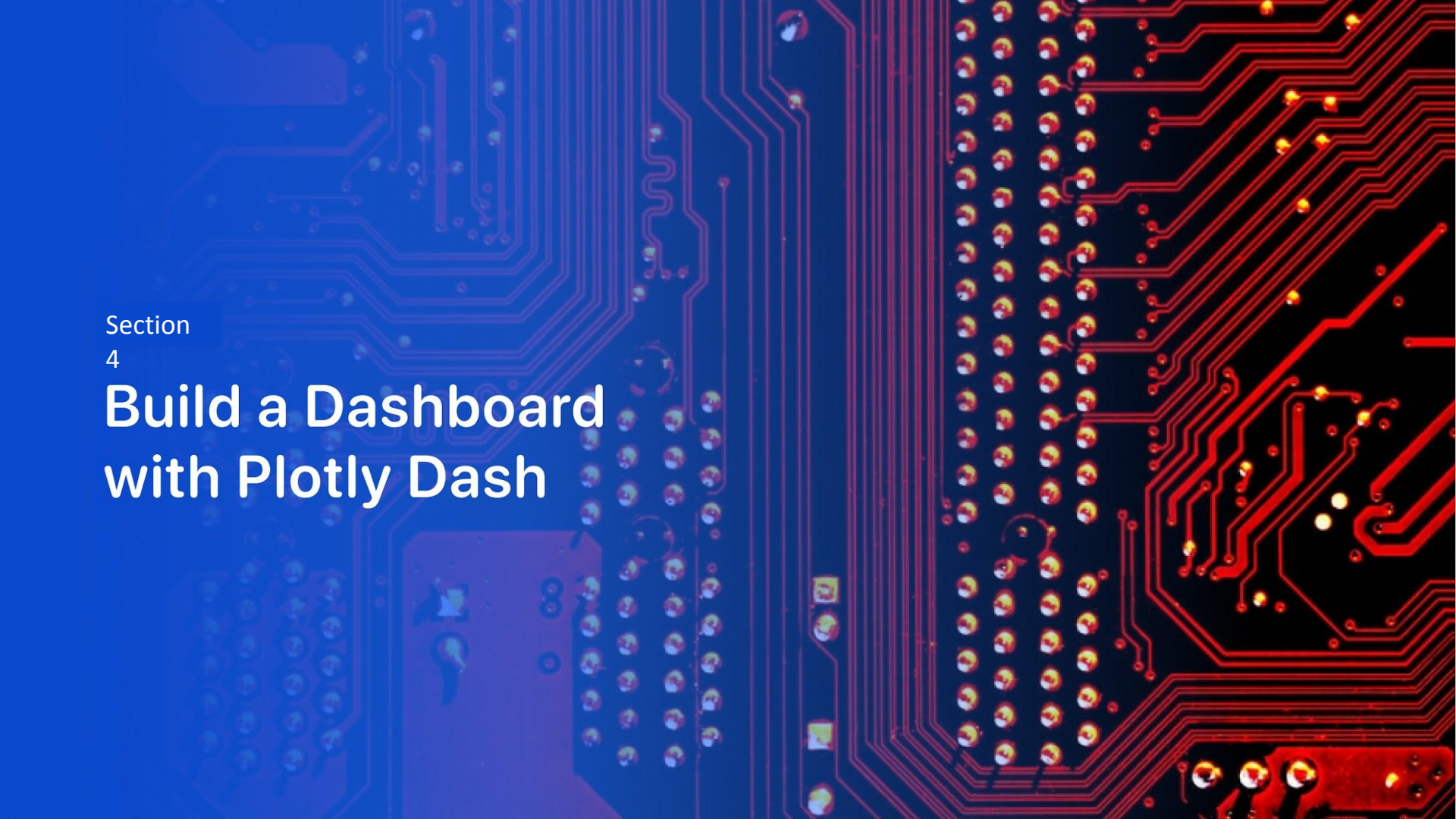


Success and Failed Launches on each Site

- 

The Distance between Launch Sites




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

<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

<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

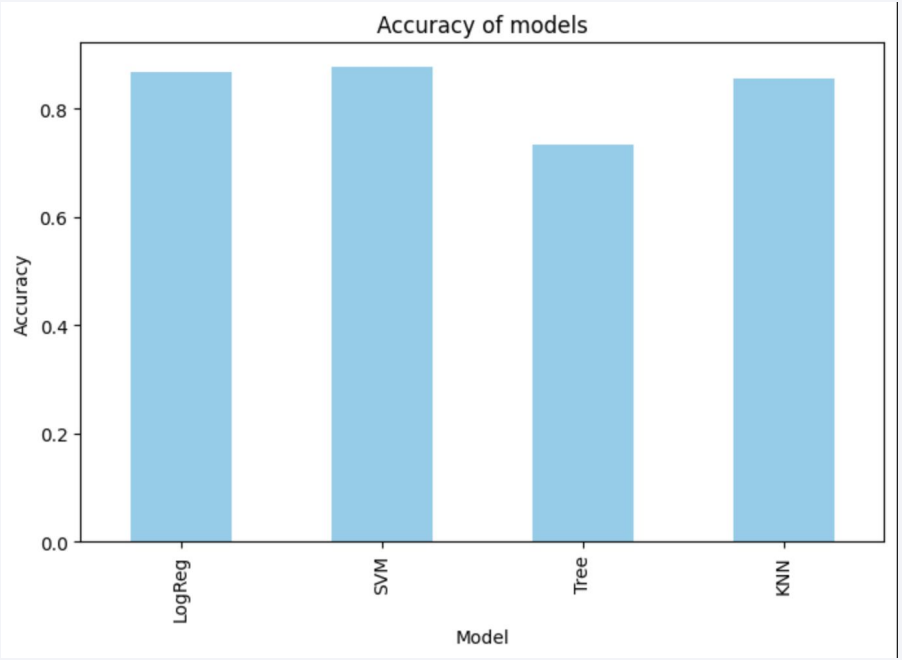
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section
5

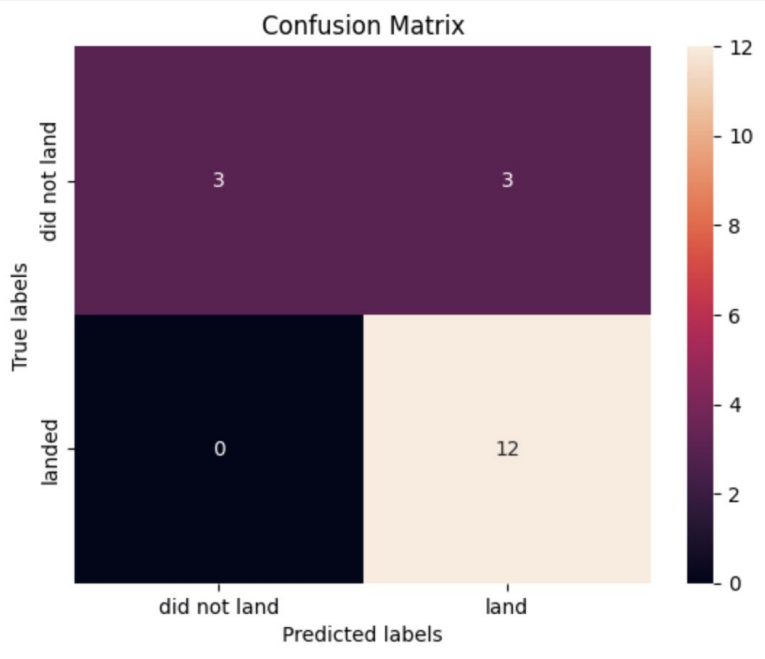
Predictive Analysis (Classification)

Classification Accuracy

- Highest classification accuracy is Logistic Regression and SVM



Confusion Matrix



Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

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

Thank you!