IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rabin Karki
October 26, 2023

Outline

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

Executive Summary

Summary of Methodologies:

The SpaceX Capstone Project aimed to analyze and visualize SpaceX launch data to gain insights into launch success rates, payload mass, launch sites, and booster versions. The project employed a data-driven approach, utilizing Python libraries such as Pandas for data manipulation, Dash for building interactive web applications, and Matplotlib, Seaborn,Folium and Plotly for data visualization.

Summary of All Results:

The analysis of the SpaceX launch data using the capstone project yielded valuable insights:

1. **Launch Site Insights:** The project identified that Kennedy Space Center Launch Complex 39A (KSC LC-39A) had the highest launch success rate. This site recorded a launch success rate of 76.9%, making it the most successful launch site.
2. **Payload Range Findings:** The analysis revealed that payloads within the range of 2000-4000 kg had the highest launch success rates. On the other hand, payloads in the range of 6000-8000 kg had relatively lower success rates.
3. **Booster Version Performance:** Among various booster versions, Falcon 9 (B5) had a 100% success rate, although it had only one successful launch. Excluding this case, Falcon 9 Full Thrust (FT) had the highest success rate, with 15 successful launches out of 23 attempts.

The project successfully provided a user-friendly interface for exploring the SpaceX launch data, enabling users to make data-driven decisions regarding launch sites, payload mass, and booster version selection for future SpaceX missions. It showcased the power of data visualization and analysis in making informed decisions in the space exploration industry.

Introduction

In the SpaceX Capstone Project, our primary objective is to predict the successful landing of the Falcon 9 first stage. SpaceX's Falcon 9 rocket launches are known for their cost-effectiveness, with advertised prices at $62 million, significantly undercutting competitors in the space launch industry, where prices often exceed $165 million per launch. One of the key cost-saving factors lies in SpaceX's ability to reuse the first stage of the Falcon 9 rocket.

The successful landing of the first stage is a pivotal event in reducing launch costs and enabling cost-effective space travel. If we can accurately predict whether the first stage will land successfully, we can effectively determine the overall cost of a launch. This information holds tremendous significance for potential competitors looking to bid against SpaceX for rocket launch contracts.

The ability to consistently achieve this outcome significantly impacts the economics of space missions.

In this capstone project, we will leverage data collection and ensure it is in the correct format from an API source. By predicting first-stage landing success, we aim to provide valuable insights not only to SpaceX but also to other space launch service providers and potential bidders in the ever-evolving space exploration industry. This project represents a critical step in facilitating cost-effective and accessible space travel, thereby contributing to the broader goal of expanding human presence in the cosmos.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**

  Data collection for this project involves acquiring the historical SpaceX launch records from a reliable API source. This data source provides information about each launch, including launch site, payload mass, booster version, booster version category, and the outcome (success or failure) of the first stage landing. The data is collected using API requests and stored in a structured format for further analysis.

- **Perform data wrangling:**

  The collected data often requires cleaning and preprocessing to ensure consistency and readiness for analysis. Data wrangling involves handling missing values, correcting data types, and removing duplicates. This step ensures that the data is in a suitable format for exploration and modeling.

- **Data Processing:**

  Following data wrangling, data processing includes feature engineering, where new variables or features are created from the existing dataset to enhance the quality of the predictive models. Additionally, data normalization and standardization may be applied to ensure that all features are on the same scale.

- **Perform exploratory data analysis (EDA) using visualization and SQL:**

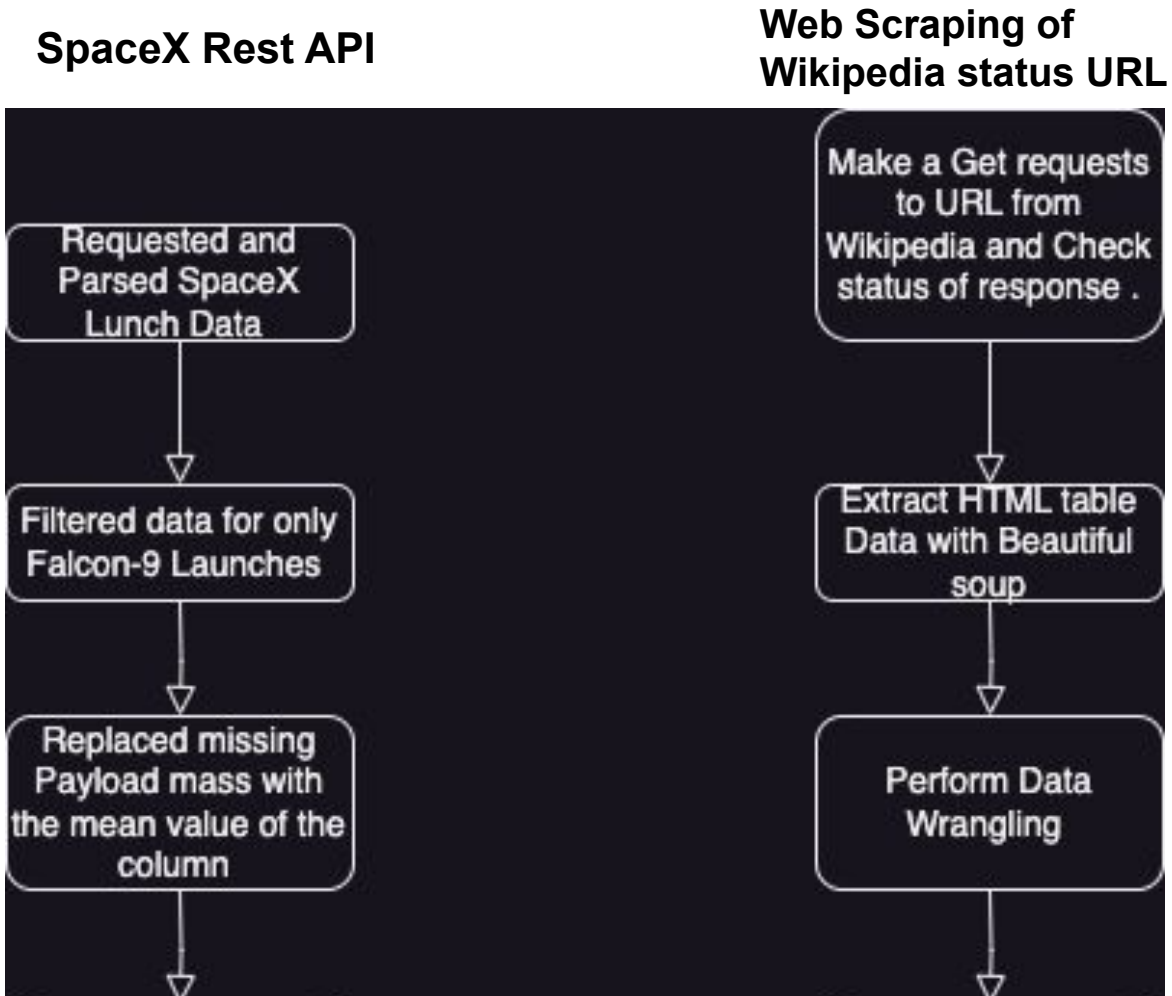
  EDA is a crucial step to gain insights into the data and understand its characteristics. Visualization techniques, including histograms,

Data Collection

The data for this project was collected from:

1. Open Source SpaceX REST API
2. Falcon-9 Launch Data from Wikipedia through Web Scraping

Data Collection Flowchart:



Data Collection – SpaceX API

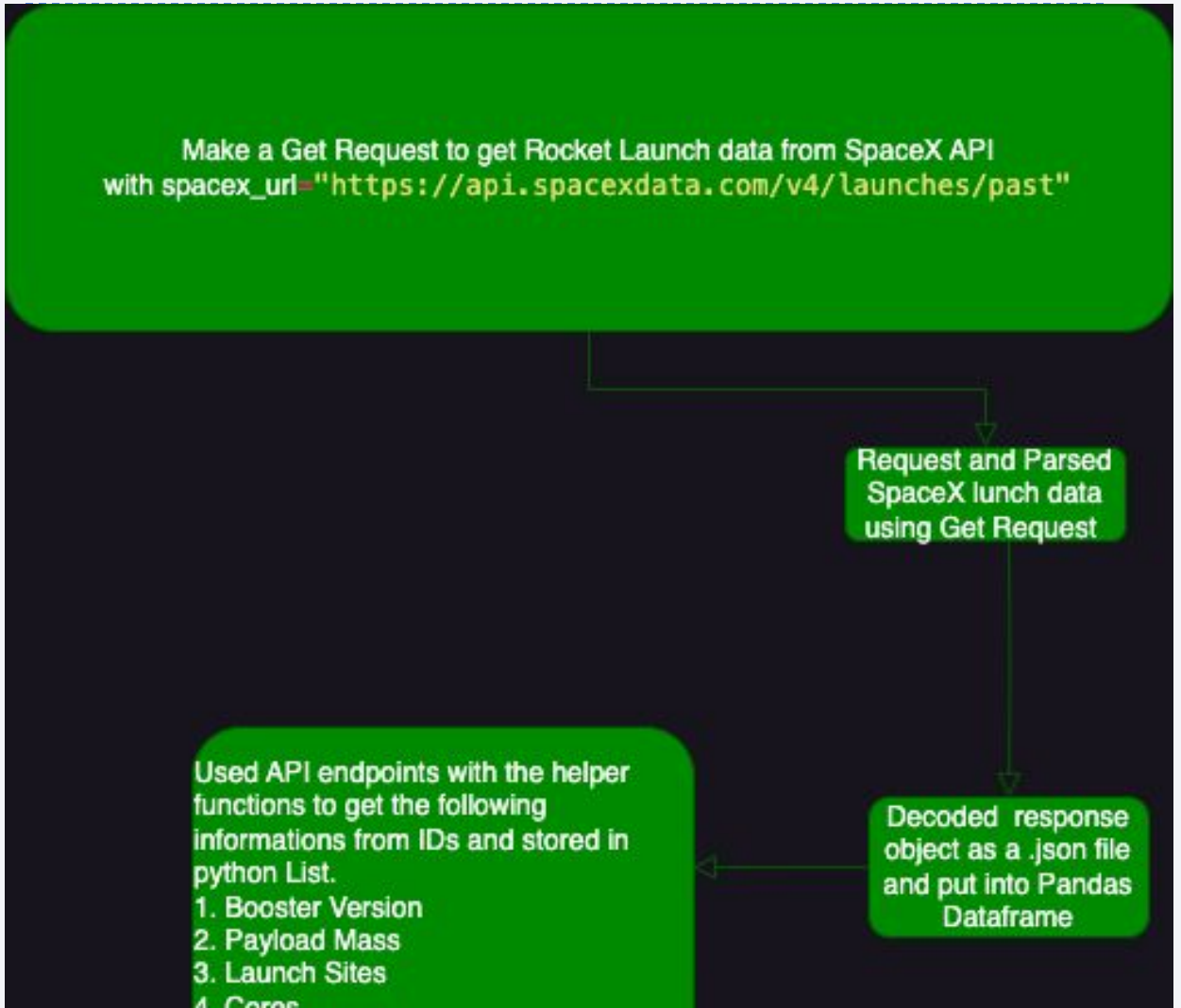
- I defined a series of helper functions that help us use the API to extract information using identification number in the lunch data .

  The followings are the helper functions:

  - getBoosterVersion(data)
  - getLaunchSite(data)
  - getPayloadData(data)
  - getCoreData(data)

Github Link:
SpaceX_Falcon9_Landing_Predictions/**jupyter-labs-spacex-data-collection-api.ipynb**

FlowChart of Data collection Using SpaceX API

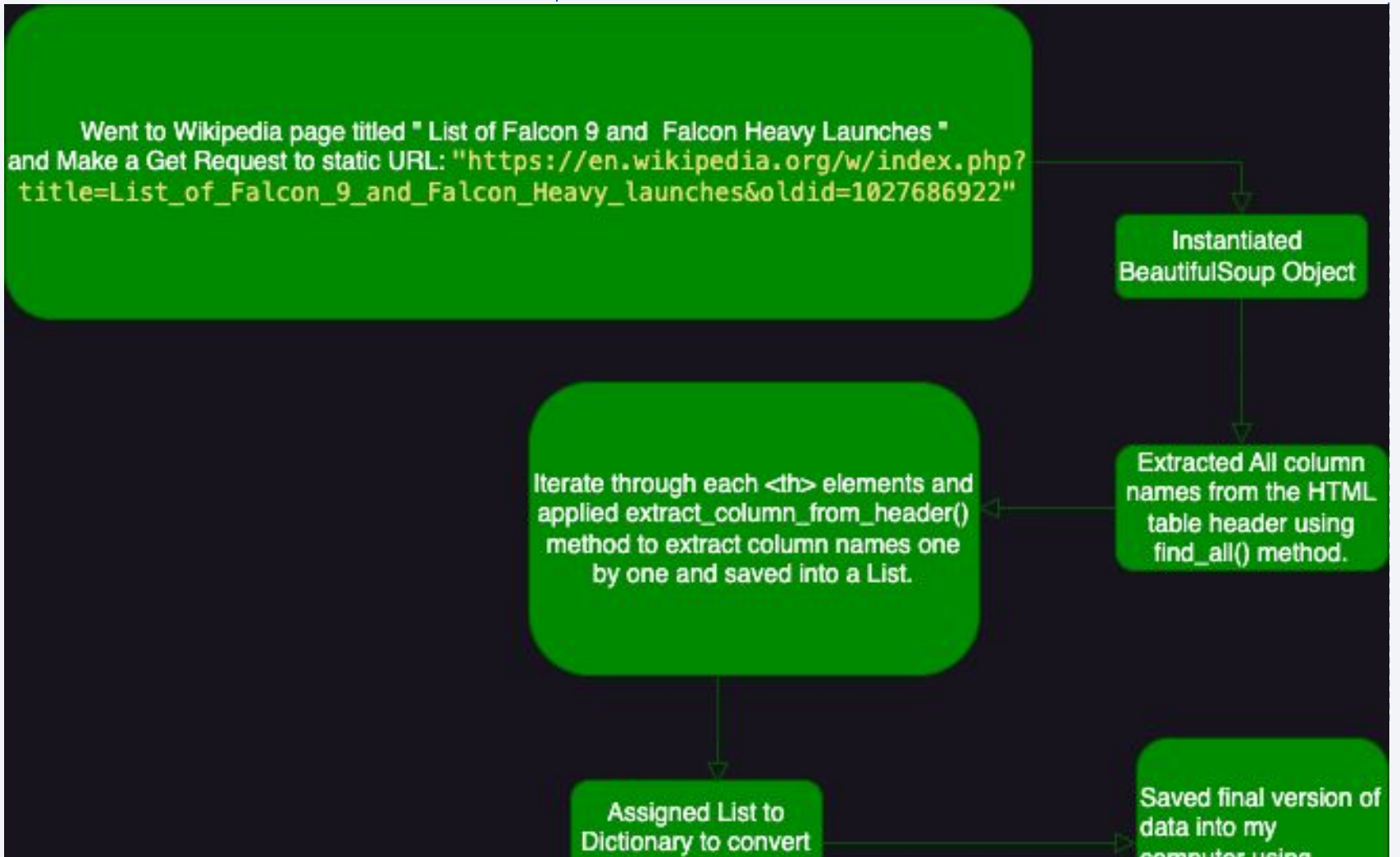


Data Collection - Scraping

- I defined series of functions to extract the specific informations from HTML table cells.
- **Here are the functions :**

1. **date_time(table_cells):** This function is used to extract the date and time from HTML table cell.
2. **booster_version(table_cells):** This function is used to extract booster version from HTML table cells.
3. **landing_status(table_cells):** This function is used to extract the landing status from HTML table cells
4. **get_mass(table_cells):** This function is used to extract payload mass from HTML table cells.
5. **extract_column_from_header(row):** This function is used to process and

Flow chart of Web scraping data collections steps



Data Wrangling

In the Data Wrangling process, I first perform EDA and then converted categorical variables into binary classification to determine the training labels .

EDA steps:

- First checked if there are any null values or missing values
- Replaced missing values of “Payload” mass column with its mean value

The Calculations:

- Calculated the number of lunches on each site
- Calculated the number and occurrence of each orbit
- Calculated the number and occurrence of missions outcome per orbit type

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

Here,

True ASDS means : The mission outcome was successfully landed on drone ship

False ASDS means : The mission outcome was failed to land on drone ship .

None None means: Failure to land

True RTLS means : The mission outcome was successfully landed in ground pad

False RTLS means : The mission outcome was failed to land in ground pad

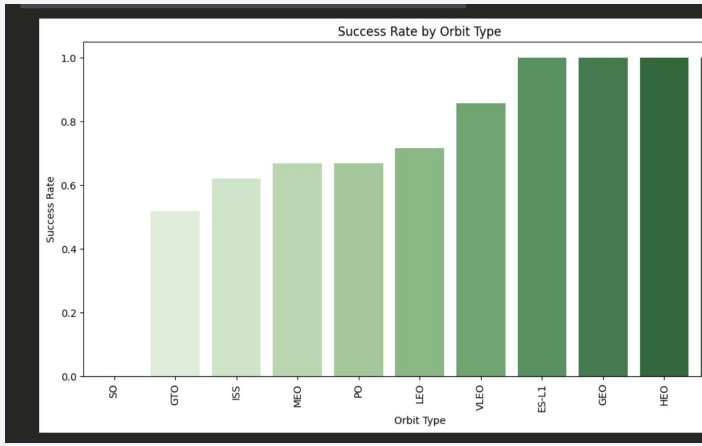
True Ocean means: The mission outcome was successfully landed on specific region of the ocean

	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Long
0	None None	1	False	False	False	NaN	1.0	0	B0003	-80.57
0	None None	1	False	False	False	NaN	1.0	0	B0005	-80.57
0	None None	1	False	False	False	NaN	1.0	0	B0007	-80.57
E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.61
0	None None	1	False	False	False	NaN	1.0	0	B1004	-80.57
	...	...	...	...	...	...	...	...	...	
A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	2	B1060	-80.60
A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	2	B1058	-80.60
A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	5	B1051	-80.60
0	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	2	B1060	-80.57
0	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	0	B1062	-80.57

EDA with Data Visualization

- In the data visualization process ,I used Matplotlib and Seaborn python libraries to plot different charts and graphs .
- There are multiple factors that can affect the success rate of Falcon 9 first stage landing outcome :
  We mainly check the following three parameters for analysis.
  1. Payload Mass
  2. Orbit Type
  3. Launch Site

- We created scatter plot to check the relationship between following two variables .
  - Payload mass vs. Flight number
  - Launch Site vs. Flight Number
  - Launch site vs Payload mass
  - Orbit vs Flight Number
  - Orbit vs Payload mass



- We also Created Bar chart to visualize the success rate of each orbit type

- Created Line chart to visualize the trend of the average lunch success rate over the years.

- Performed One hot encoding for categorical columns.
- Github Link: SpaceX_Falcon9_Landing_Predictions/**jupyter-labs-eda-dataviz.ipynb**

EDA with SQL

In order to perform EDA with SQL:

- First, connected to the database using inline magic function and sqlite3 database.
- Read csv file into the dataframe and converted into SQL table.

The following informations retrieved from SQL database table:

- Displayed the name of the unique lunch sites from the lunch_site column using DISTINCT keyword in a query.
- Displayed 5 records where the lunch_site name begins with the string “CCA” using “like” operator in a query.
- Displayed total payload mass carried by boosters launched by NASA using “sum” function in a SQL query .
- Displayed average payload mass carried by booster version F1 v1.1 using “AVG” function in a SQL query with the where clause to select the particular version.
- Displayed the list of the date when the first successfully landing outcome in ground pad was achieved using “min” function in a SQL Query.
- Displayed the list of the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 using “ and” operator in Where clause in a query .
- Analyzed mission outcomes to identify common patterns by extracting the first 7 characters of 'Mission_Outcome' and counted the occurrences. The results reveal mission success trends and were visualized to provide valuable insights.
- Identified the booster versions that carried the maximum payload mass by using a subquery to compare the payload mass with the maximum value. The unique booster versions achieving this feat were extracted for further analysis.
- Here are screenshots of some important queries that I performed .

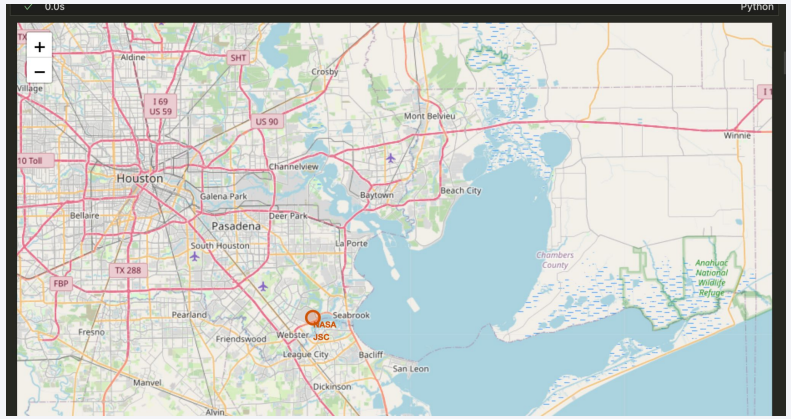

```
avg_payloadd_mass = pd.read_sql("select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1'", con)
avg_payloadd_mass
```

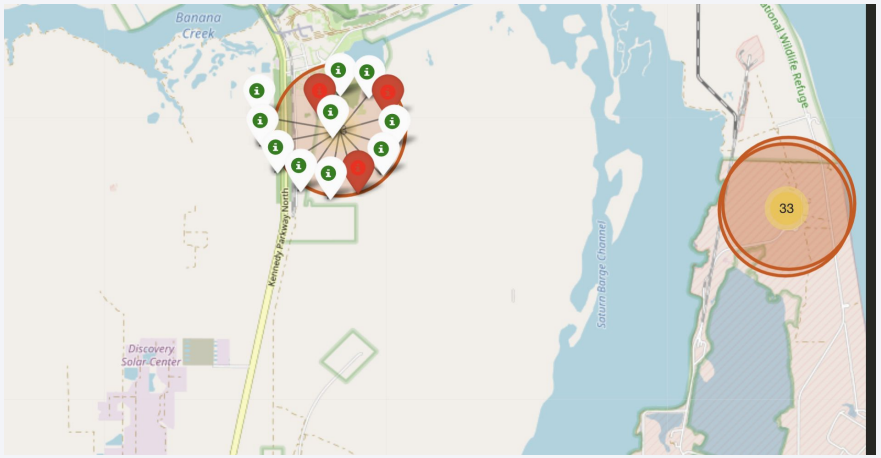
Build an Interactive Map with Folium

The Launch success rate may depend on the location and proximity of a lunch site. Folium is very useful library to show the interactive maps to analyze SpaceX launch sites .

Marking Launch Sites on a Map:

- Downloaded the spacex_launch_geo.csv dataset, which includes latitude and longitude information for launch sites.
- Selected relevant sub-columns: Launch Site, Lat(Latitude), Long(Longitude), and class.
- Created a Folium Map object with NASA Johnson Space Center as the initial center location.
- Used folium.Circle to add highlighted circle areas at specific coordinates to represent launch sites.
- Added a label to each circle with the launch site's name.
- Iterated through the launch sites and added circles for each site on the map.





Marking Success/Failure Launches:

- Created a new column in the launch_sites dataframe called marker_color to store marker colors based on the class value.
- Assigned marker colors based on the class value:
- If class is 1, marker color is green.
- If class is 0, marker color is red.

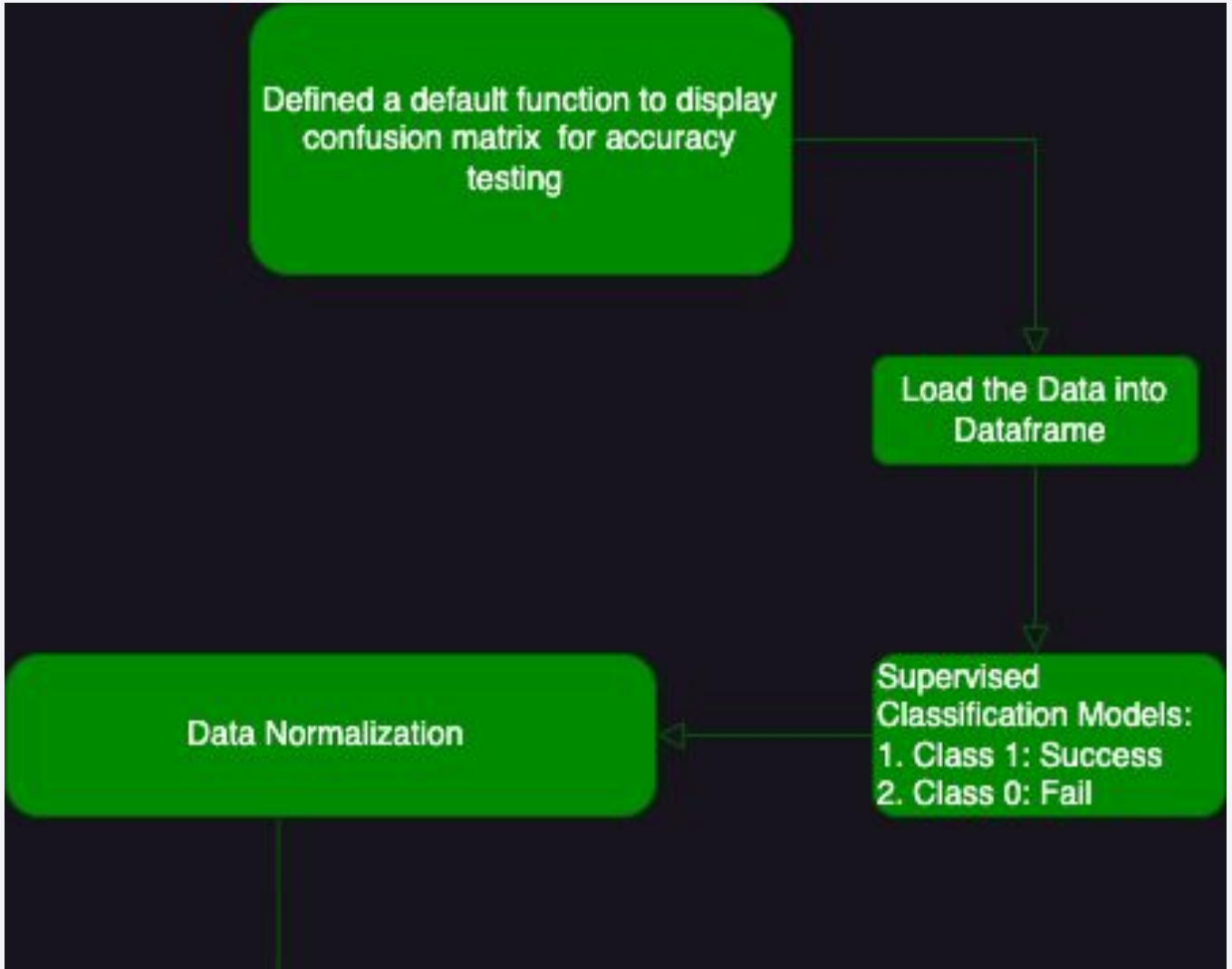
Build a Dashboard with Plotly Dash

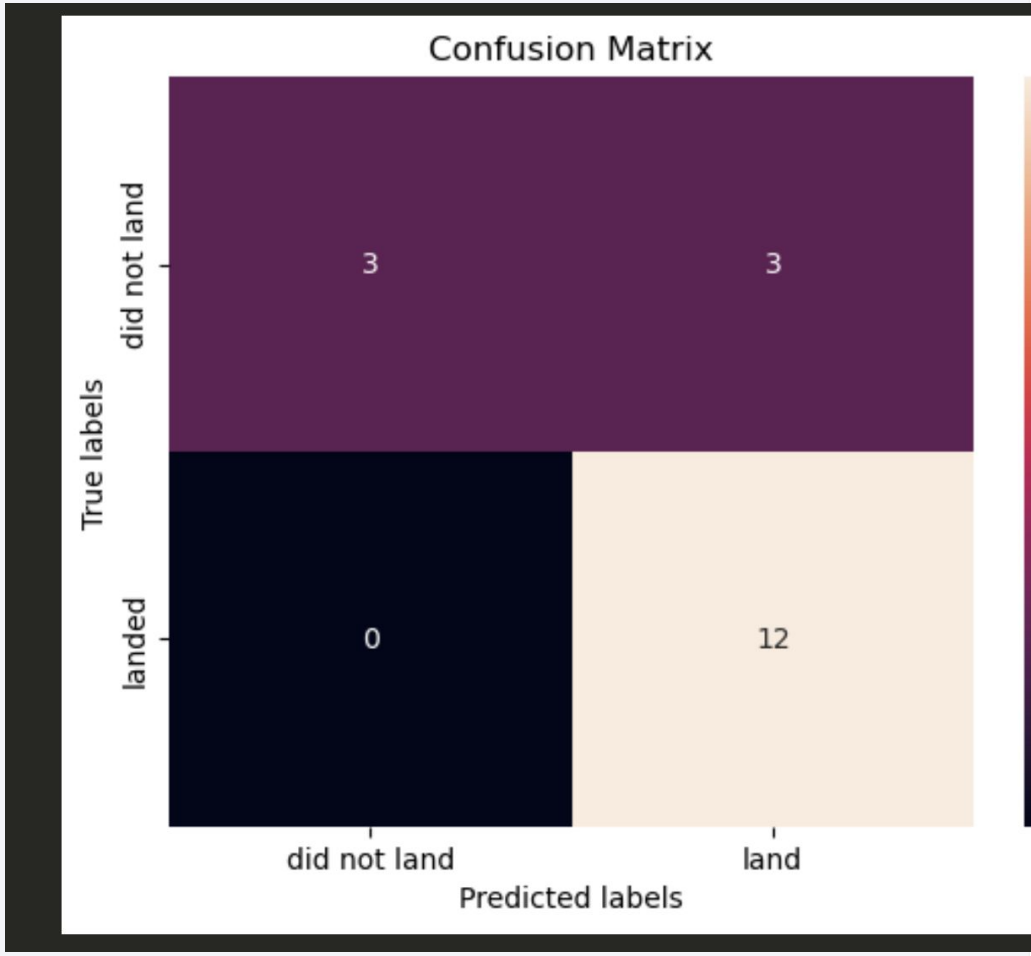
We created the following plots/ Graphs and Interactions using Plotly for analyzing SpaceX launch data and answering key questions about launch success rates, payload ranges, and their correlation with outcomes.

- Dropdown for Launch Site Selection:
  - - Enabled users to select a specific launch site.
  - - The default selection is for ALL sites.
  - - Users can search for a specific site, making site selection convenient.
- Pie Chart for Launch Success:
  - - Shows the total successful launches count for all sites.
  - - If a specific launch site is selected, it displays the Success vs. Failed counts for that site.
  - - Provides a quick overview of launch success rates.
- Payload Range Slider:
  - - A range slider allows users to select the payload range (in kilograms).
  - - Provides an interactive way to filter data based on payload mass.
  - - Users can customize the payload range to analyze specific scenarios.
- Scatter Chart for Payload vs. Launch Success:
  - - Displays a scatter chart showing the correlation between payload mass and launch success.
  - - Differentiates launches by Booster Version Category using color.
  - - Helps users understand the relationship between payload mass and launch outcomes.
- Reasons for Adding Plots and Interactions:

Predictive Analysis (Classification)

Flowchart



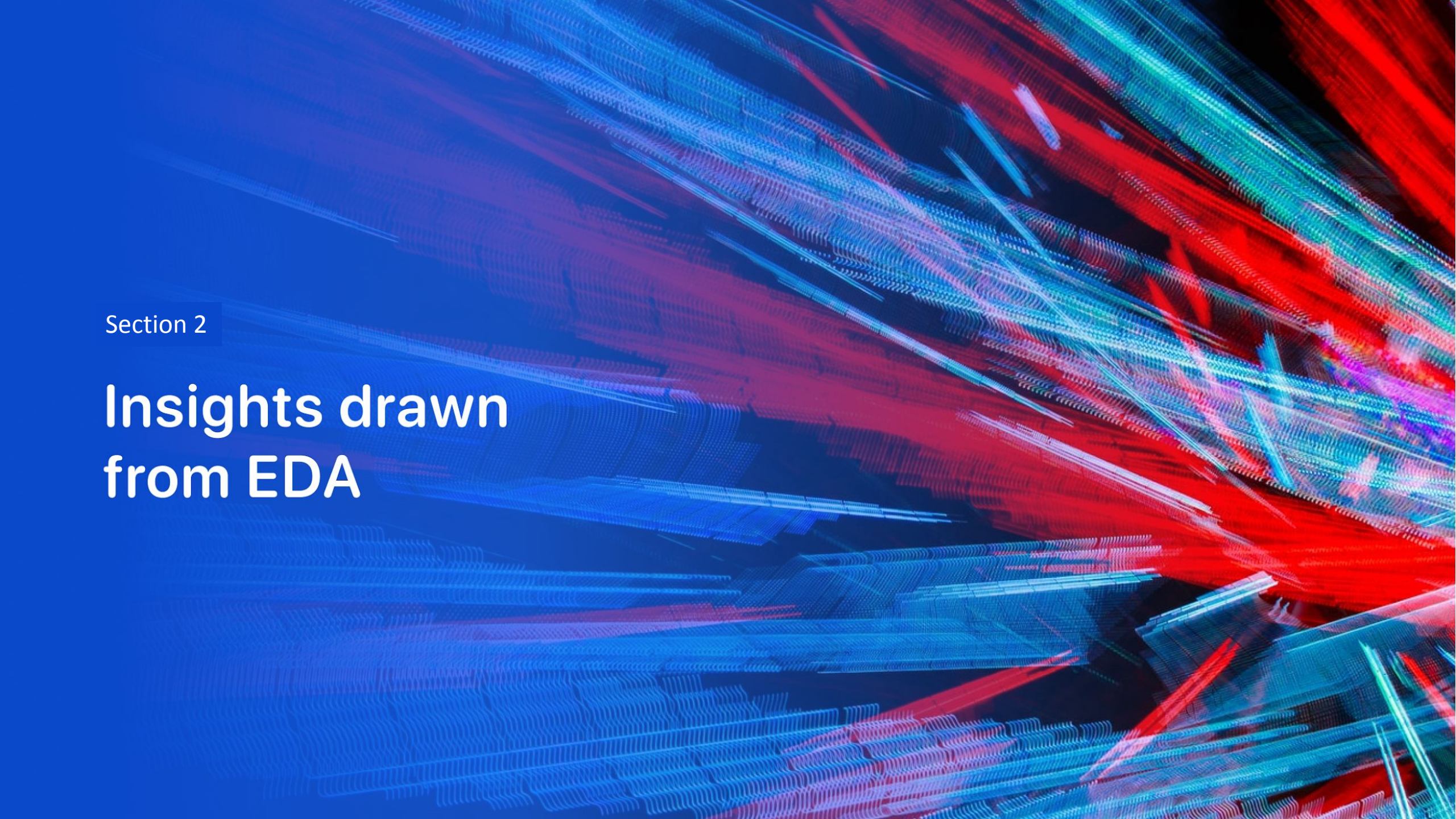


Results

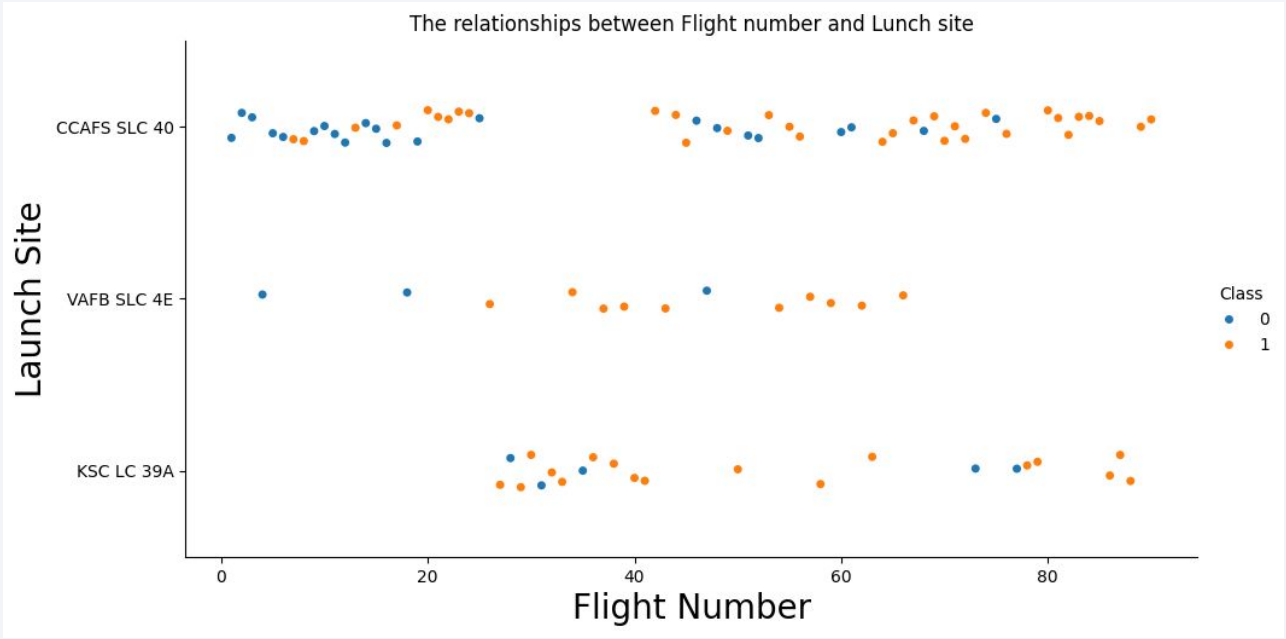
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA



Flight Number vs. Launch Site



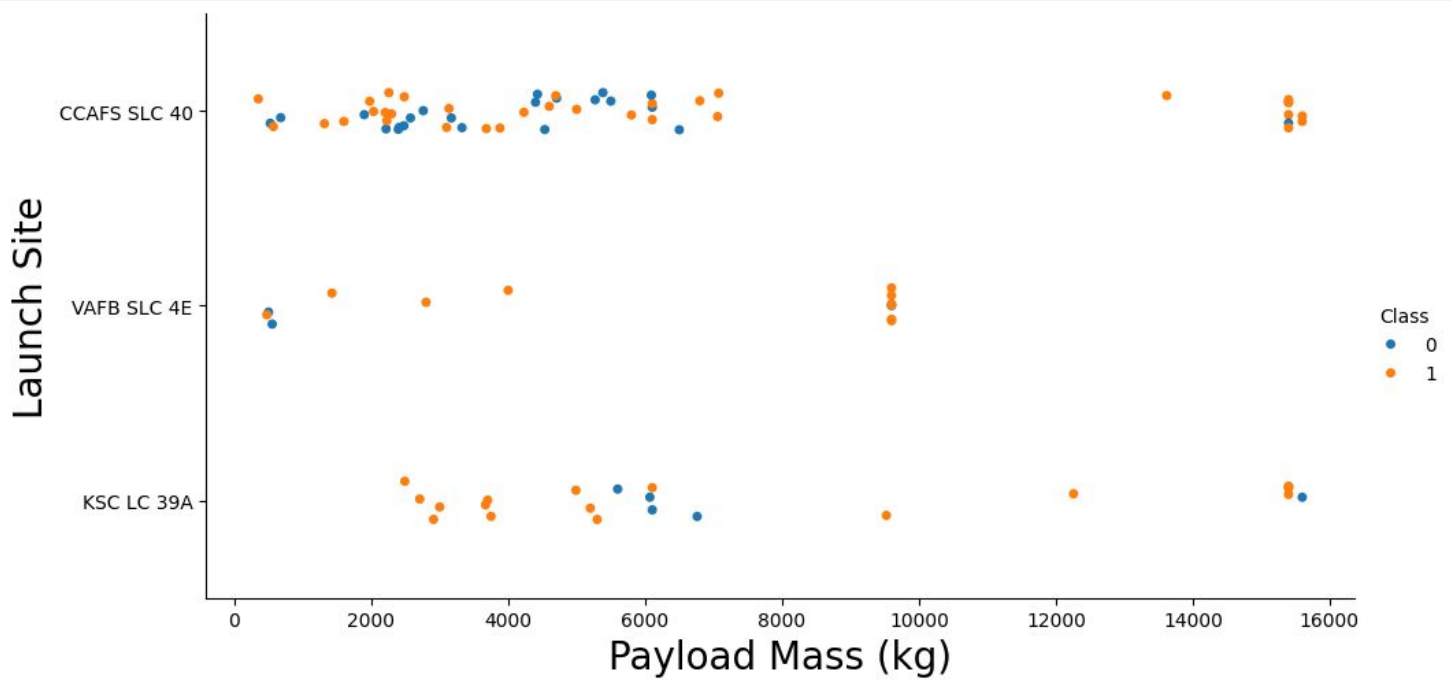
Remarks:

X-axis => The number of lunch attempts by Falcon 9

Y-axis => Name of the different Launch sites

0/1 => failure/success per lunch sites

Payload vs. Launch Site



Remarks:

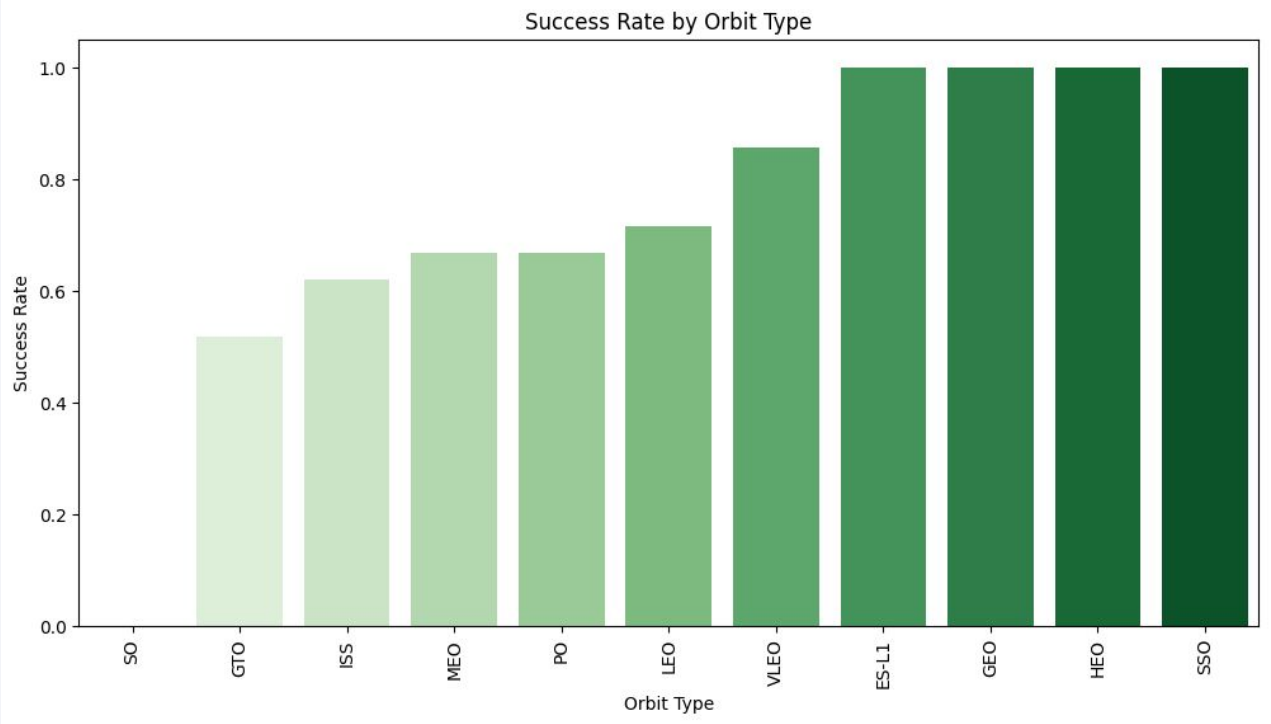
x-axis=> Total mass of cargo carried by Falcon 9 in kg

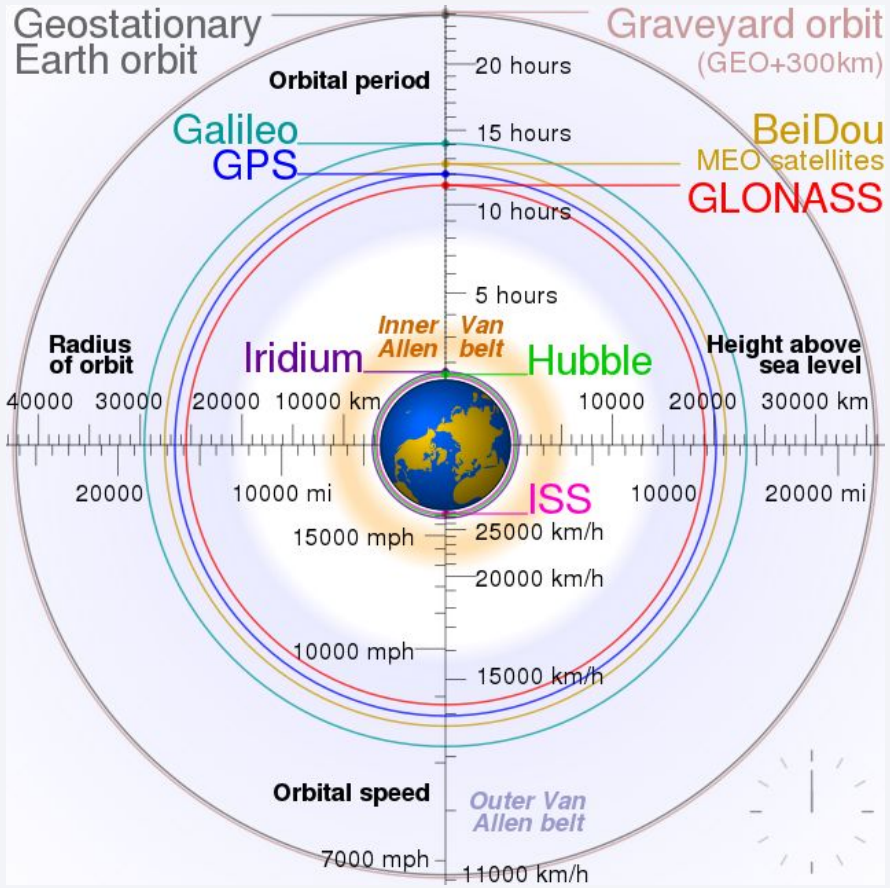
y-axis => Launch sites

0/1 => failure / success of the attempts by different weights cargo

Insights:

Success Rate vs. Orbit Type





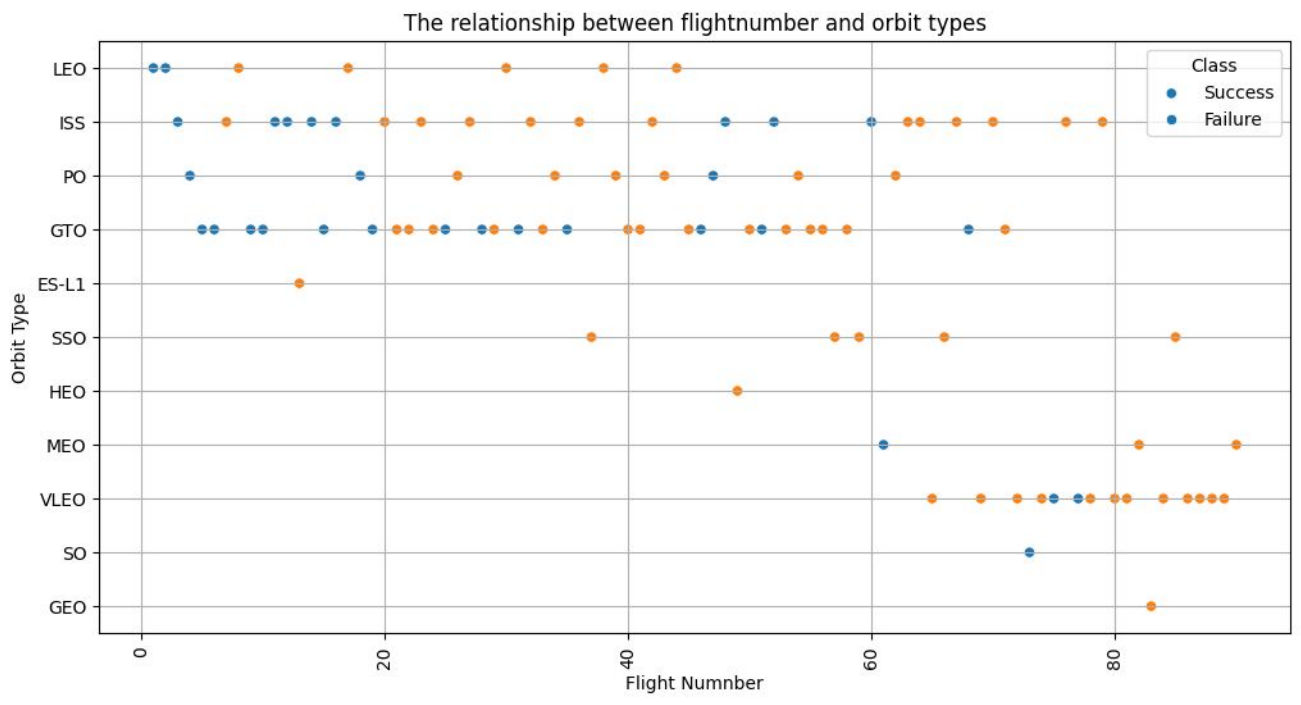
Remarks:

x-axis => The name of the different orbits that Falcon-9 can travel.

y-axis=> Success rate which is calculated by finding the mean of success/Failures based on each orbit type

The most of the human made objects are in LEO which is lower earth orbit and also most of the flights have been attempted .

Flight Number vs. Orbit Type



Remarks:

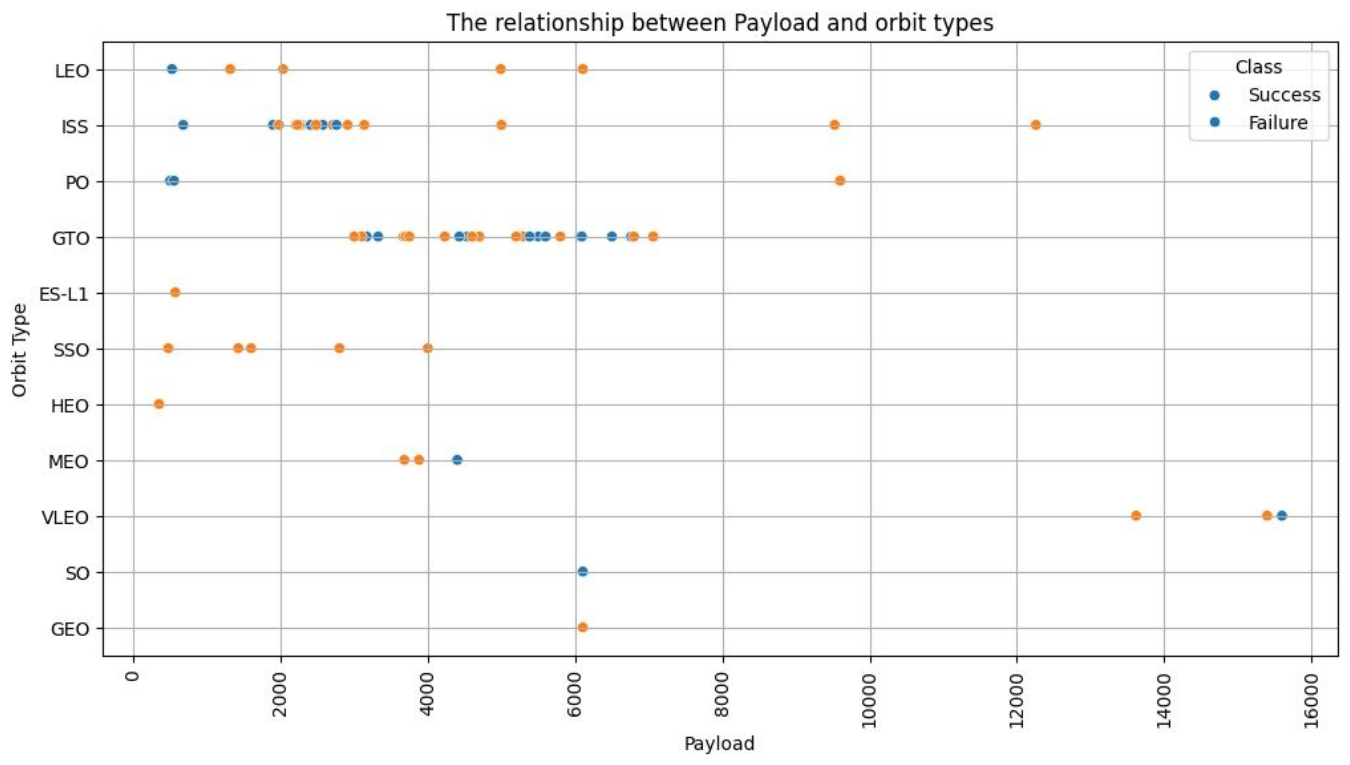
x-axis=> The number of flight attempted by Falcon 9 in different orbit types

y-axis=> The name of the different orbits that Falcon 9 have attempted their flights .

0/1=> failure/success

Insights:

Payload vs. Orbit Type



Remarks:

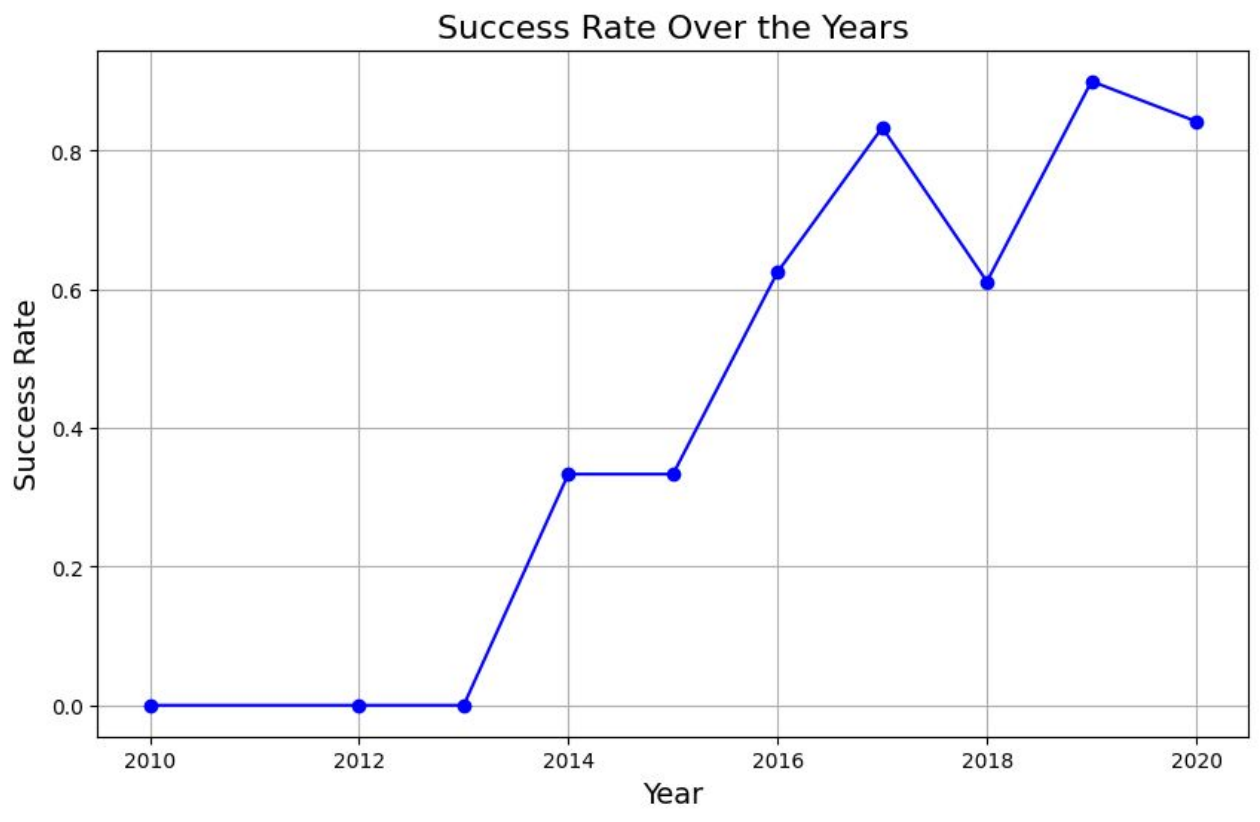
x-axis=> mass of cargo carried by Falcon-9 in kg

y-axis=> different orbit types

0/1=> failure/success

Insights:

Launch Success Yearly Trend



Remarks:

x-axis => different years that SpaceX launches Falcon 9

y-axis => success rate

Insights :

All Launch Site Names

```
[34]  q = pd.read_sql('select distinct Launch_Site from SPACEXTBL', con)
      q
```

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

Explanation:

- I selected unique launch site names from “SPACEXTBL” using “DISTINCT” keyword in a SQL Query.
- Here, “con” is a connection object that we used to connect to database.

Launch Site Names Begin with 'CCA'

```
lunch_record = pd.read_sql("select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5", con)
lunch_record
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Missi
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	

Explanation:

Here, I used “Like” operator to select launch sites which name starts from string “CCA”. We limit first 5 records from the table using “LIMIT” keyword .

Total Payload Mass

```
total_payload = pd.read_sql("select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)' ", con)
total_payload
```

	sum(PAYLOAD_MASS__KG_)
0	45596

Explanation:

Here, I used “sum” function to find the total payload mass with in a query where customer is NASA.

Average Payload Mass by F9 v1.1

```
avg_payloadd_mass = pd.read_sql("select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1'", con)
avg_payloadd_mass
```

	avg(PAYLOAD_MASS__KG_)
0	2928.4

Explanation:

Here, I used “avg” function within a sql query to get the average payload mass for the Booster version F9 v1.1

First Successful Ground Landing Date

```
fsl_date = pd.read_sql("select min(Date) as Minimum_Date from SPACEXTBL where Landing_Outcome ='Success (ground pad)'",
fsl_date
```

Pyth

	Minimum_Date
0	2015-12-22

Explanation:

Here, I used “min” function to get the first landing successful ground landing date within a query .

Successful Drone Ship Landing with Payload between 4000 and 6000

```
booster_name= pd.read_sql("select distinct  Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' a
booster_name
                                  BL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000 ", con)
```

Python

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Explanation:

Here, I used “between” operator to select payload mass between 4000 and 6000 for successful Drone ship landing.

Total Number of Successful and Failure Mission Outcomes

```
ess = pd.read_sql("SELECT count(*) AS Successful_Outcomes FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'", c
ess
ure = pd.read_sql("SELECT COUNT(*) AS Failed_Outcomes FROM SPACEXTBL WHERE Landing_Outcome = 'Failure (drone ship)'", con)
ure
th success and failure mission outcome in the same table using substring method
d group by the firs selected clause in the column
pd.read_sql("select substr(Mission_Outcome , 1, 7) as Mission_Outcomes, count(*) as count from SPACEXTBL group by 1 ", con
```

Python

	Mission_Outcomes	count
0	Failure	1
1	Success	100

Explanation:

Here, I wrote 3 separate queries but last query gives both numbers of failures and successes counts in a same table using substring method to select the string and “count” method to find the total number of success and failures.

Boosters Carried Maximum Payload

```
.("select distinct Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)",
```

Python

	Booster_Version
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

Explanation:

Here, I used subquery to select the maximum mass using “max” function within a query.

2015 Launch Records

Explanation:

Here, I used case statements to switch months based on-substring values and assigned as a MonthName- in a table.

I also used “Like” Operator to select the failing landing - outcomes in drone as you can see in the screenshot.

```
a = pd.read_sql("""
    SELECT
        CASE
            WHEN substr(Date, 6, 2) = '01' THEN 'January'
            WHEN substr(Date, 6, 2) = '02' THEN 'February'
            WHEN substr(Date, 6, 2) = '03' THEN 'March'
            WHEN substr(Date, 6, 2) = '04' THEN 'April'
            WHEN substr(Date, 6, 2) = '05' THEN 'May'
            WHEN substr(Date, 6, 2) = '06' THEN 'June'
            WHEN substr(Date, 6, 2) = '07' THEN 'July'
            WHEN substr(Date, 6, 2) = '08' THEN 'August'
            WHEN substr(Date, 6, 2) = '09' THEN 'September'
            WHEN substr(Date, 6, 2) = '10' THEN 'October'
            WHEN substr(Date, 6, 2) = '11' THEN 'November'
            WHEN substr(Date, 6, 2) = '12' THEN 'December'
            ELSE 'Unknown'
        END AS MonthName,
        Landing_Outcome AS FailureLandingOutcomes,
        Booster_Version,
        Launch_Site
    FROM SPACEXTBL
    WHERE substr(Date, 1, 4) = '2015'
    AND Landing_Outcome LIKE '%drone ship%';
""", con)

a
```

	MonthName	FailureLandingOutcomes	Booster_Version	Launch_Site
0	October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
2	June	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

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

Explanation:

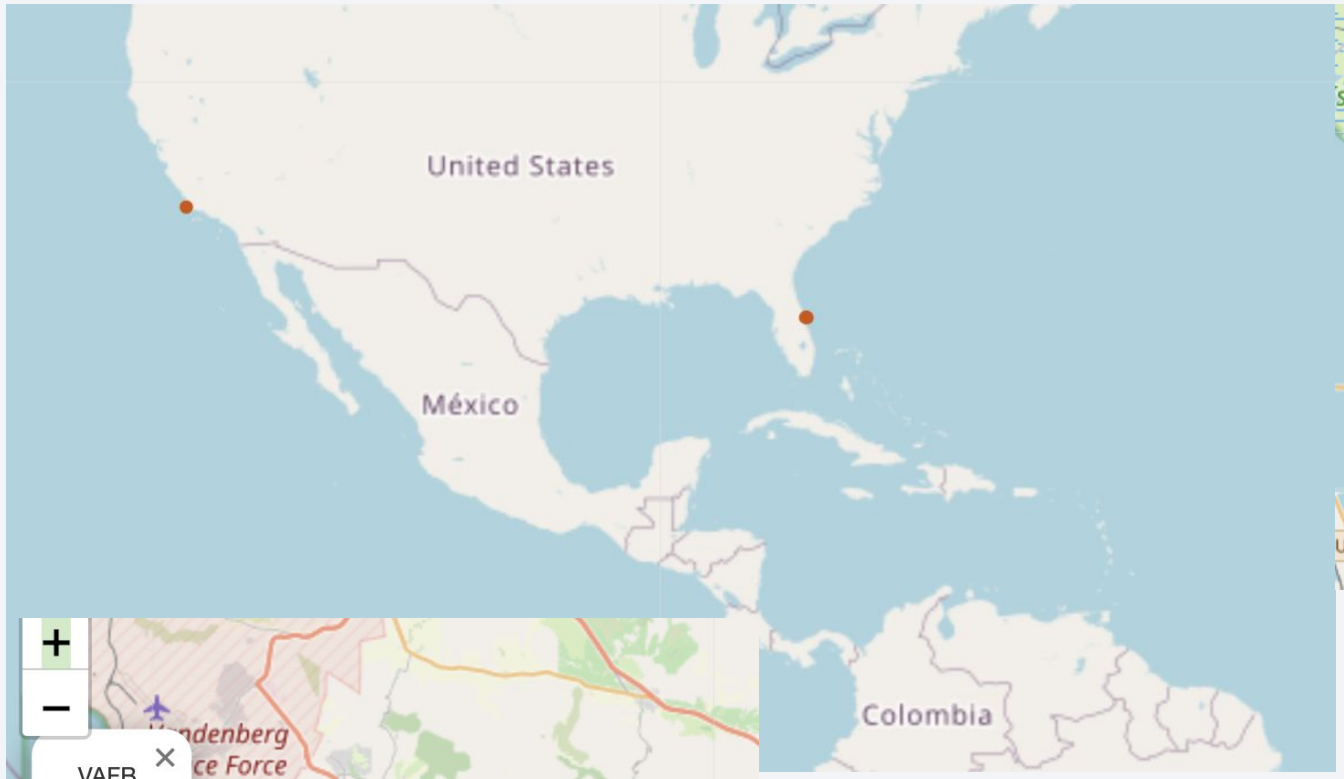
Here, I used “Group By” clause to group-landing outcomes and count their total-occurrences using “count” method and-showed in descending order.

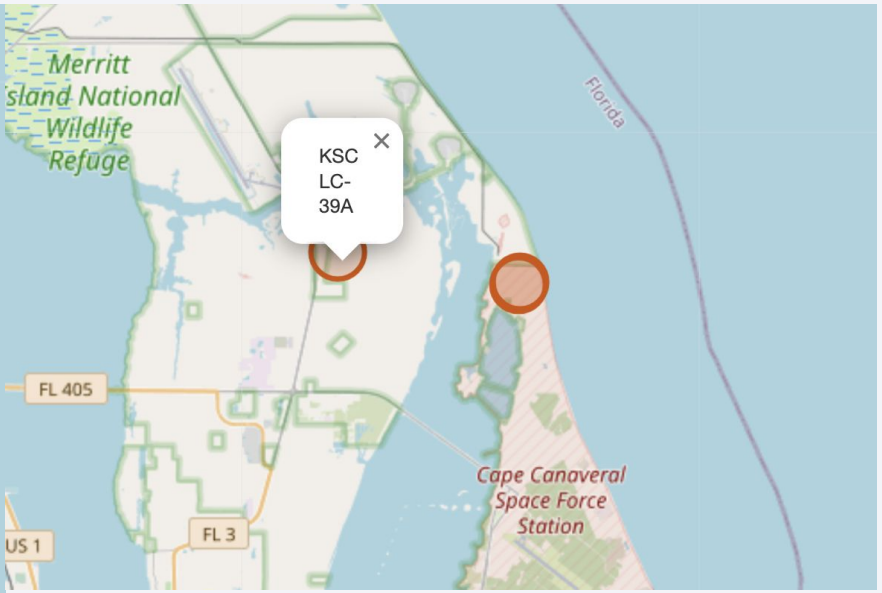
```
a = pd.read_sql("""
    SELECT
        Landing_Outcome,
        COUNT(*) AS Count
    FROM SPACEXTBL
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY Count DESC
""", con)

a
```

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


Section 3
Launch Sites
Proximities Analysis

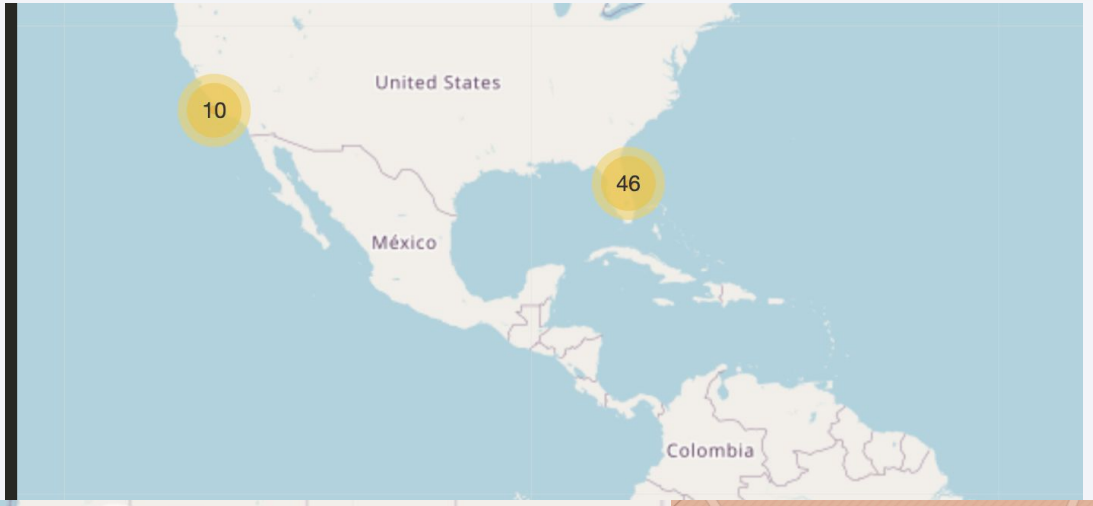
SpaceX Launch Sites

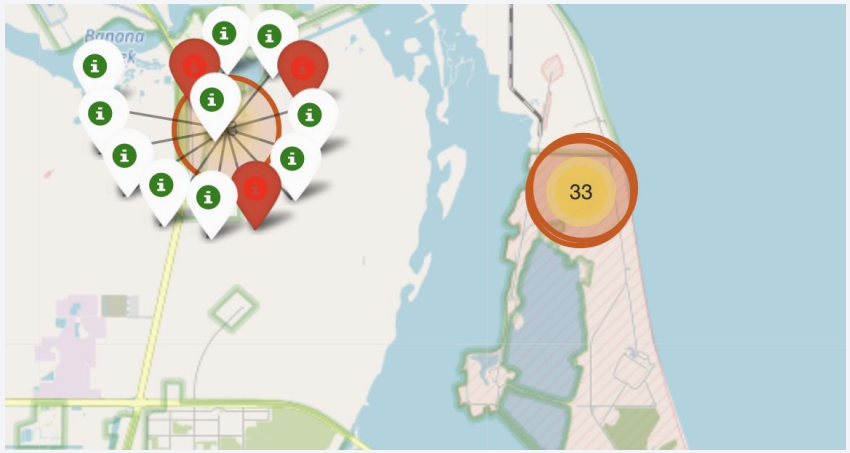


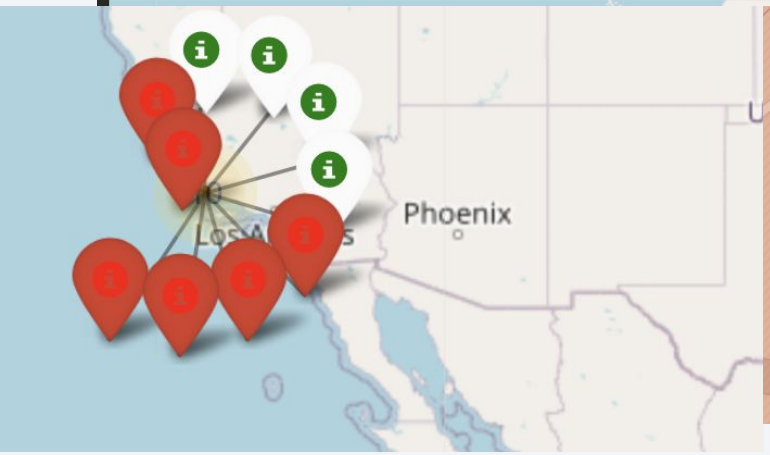


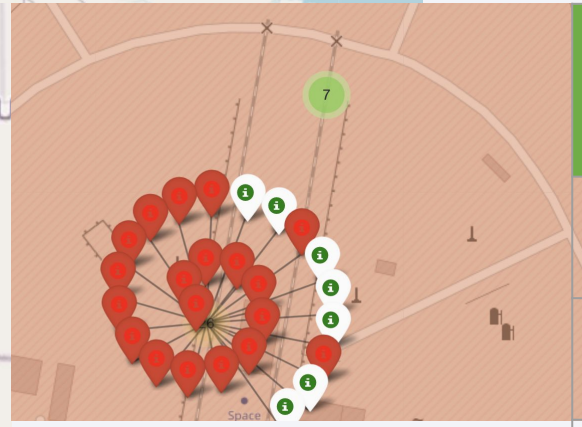
- **Name of the Launch Sites :**
- CCAF-LC40: Cape Canaveral Launch Complex (FL)
- CCAF-SLC40: Cape Canaveral Space Launch Complex(FL)
- KSC-LC29A: Kennedy Space Center-Merritt Island (FL)
- VAFB SLC-4E: Vandenberg Space Launch Complex(CA)

Falcon 9 Success/Failed Numbers For each launch sites





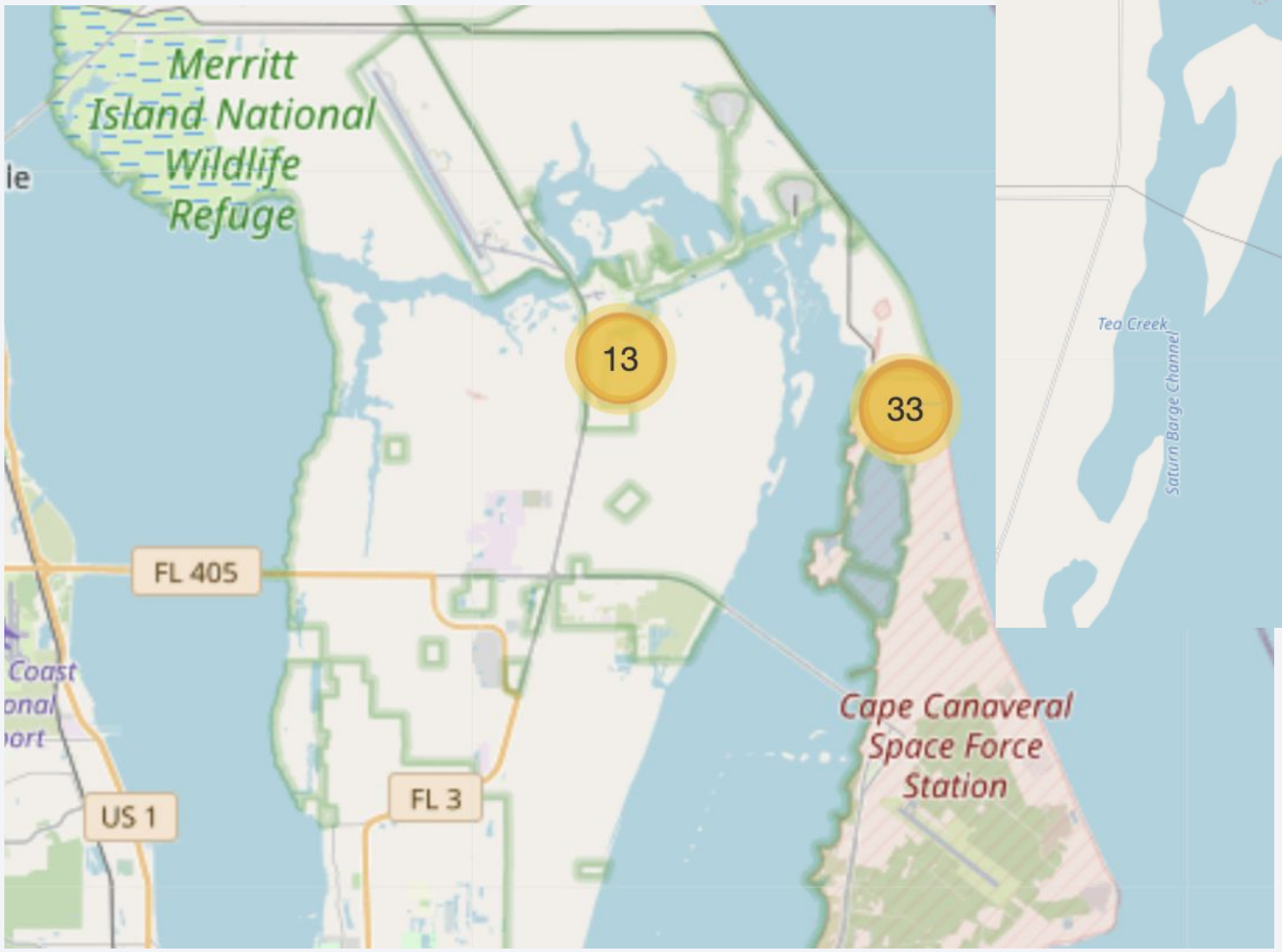


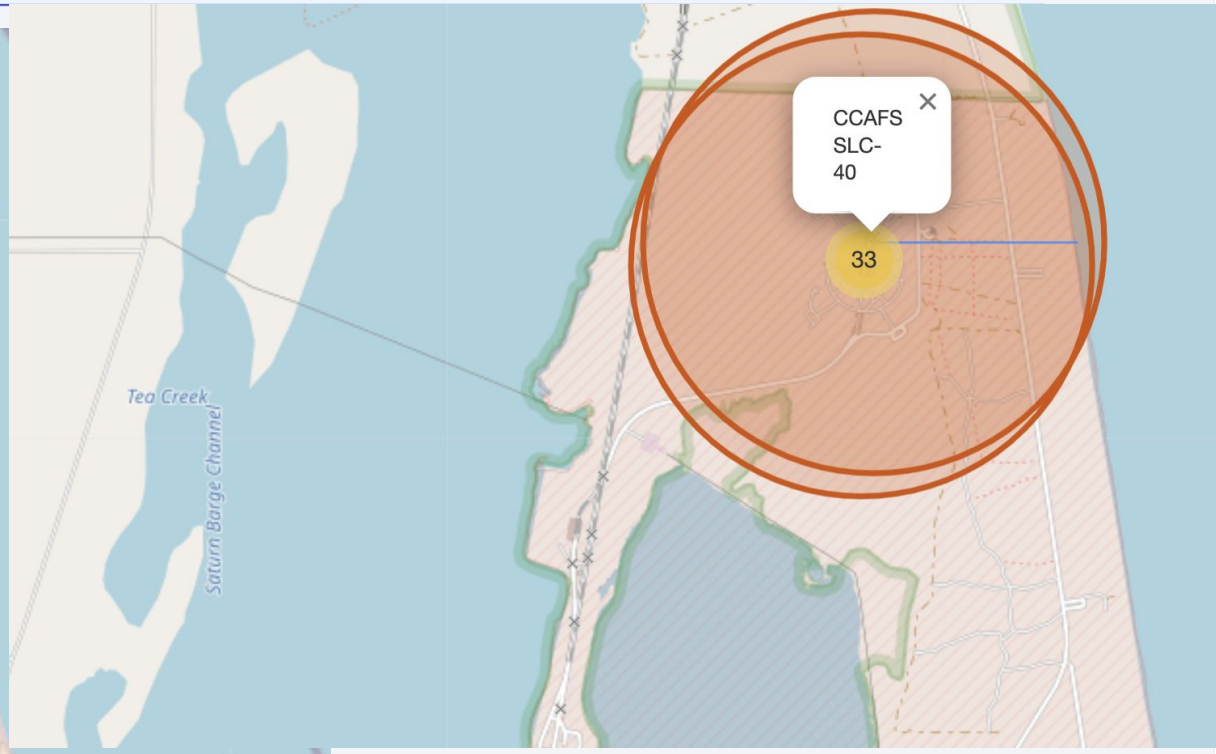


Name of the Launch sites	The total number of successful launches	The total number of failures
CCAFS LC-40	7	19
CCAFS SLC-40	3	4
KSC LC-39A	10	3
VAFB SLC-4E	4	6

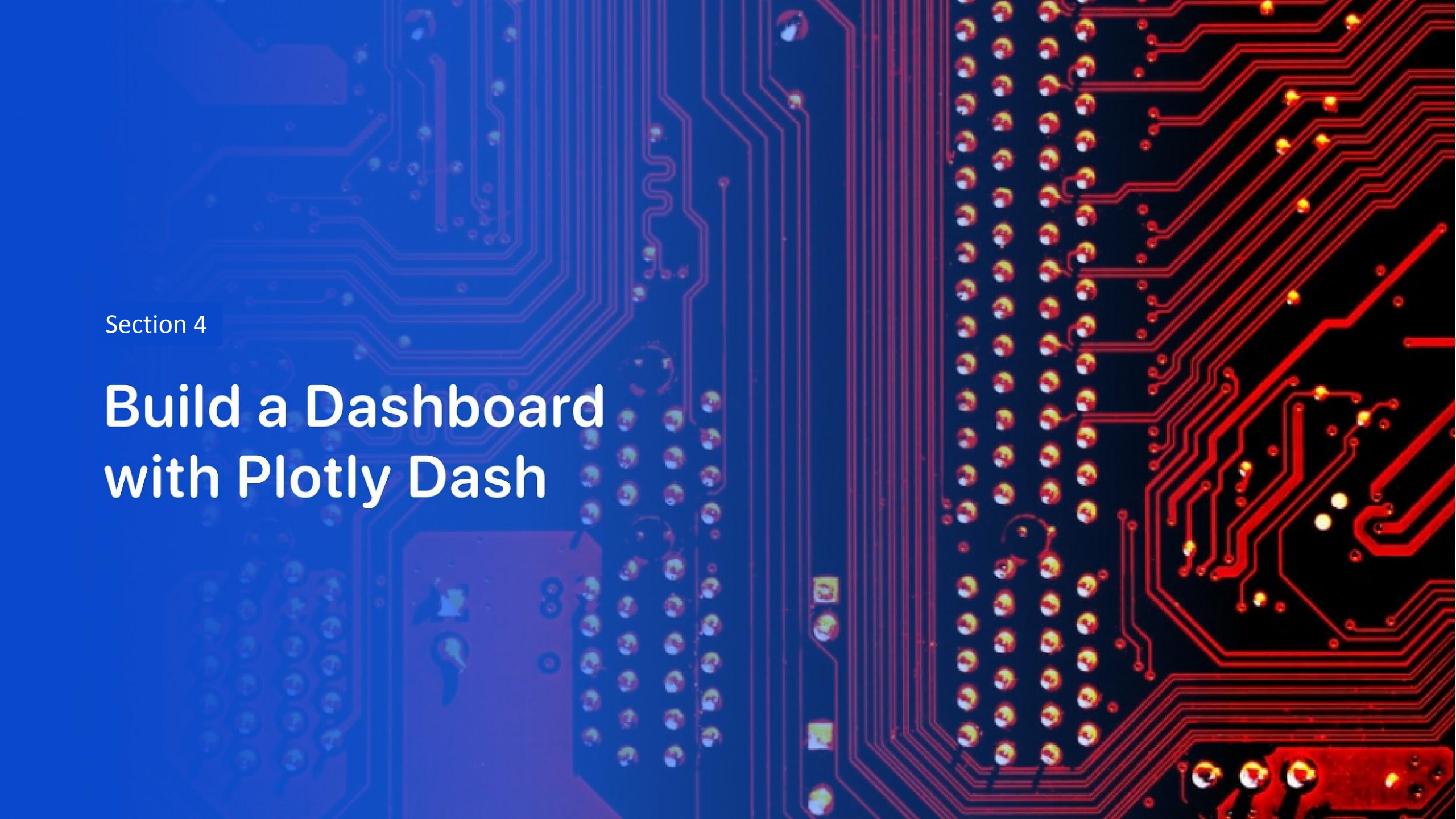
Here in the map, Red marker means unsuccessful launch and green means

Distance Between Launch Sites with its Proximities





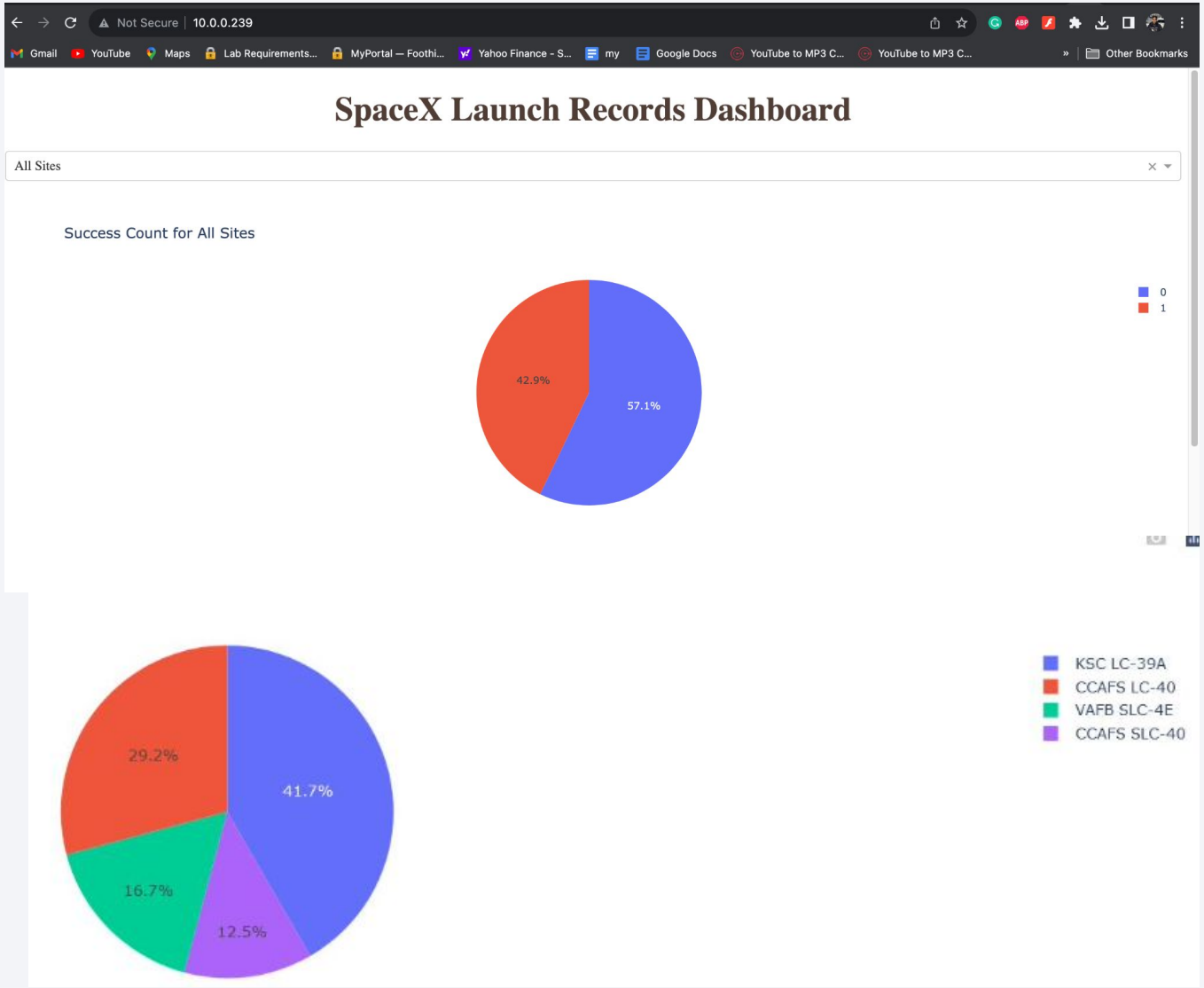
All Launches sites are close to the Coasts because of safety issues .


Section 4
Build a Dashboard with Plotly Dash

Falcon 9 : Launch Success count for all the sites

Insights :

- The highest success rate is in KSC LC-39A.
- The smallest success rate is in VAFB SLC-4E



Falcon 9: Highest success ratio amongst all launch sites

Insights:

- Heavy payload(greater than 5500 kg) is carried by booster version FT(Full Thrust).
- KSC-LC-39A has the highest success rate ratio, where out of 13 flights, 10 are successful.

Payload vs. Launch Outcome for all sites

Insights:

- FT has the highest success rate for the payload mass between 2000 kg and 6000 kg .
- V1.0 and V1.1 are the early launches with the low reliability as we can see in the screenshot .
- B4 is the only one booster version who carries big payload (greater than 8000 kg) and had a success .

Section 5

Predictive Analysis (Classification)

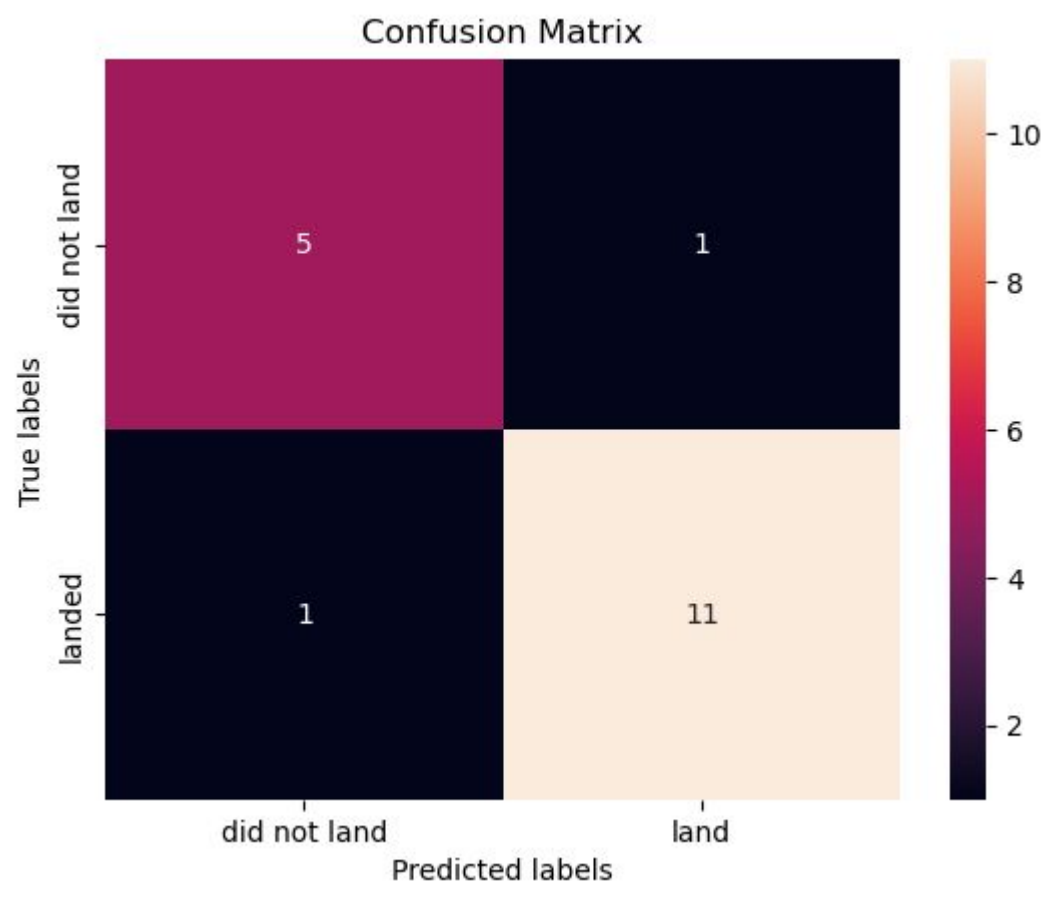
Classification Accuracy

```
# Find the method with the highest accuracy score
best_method = max(accuracy_scores, key=accuracy_scores.get)
best_accuracy = accuracy_scores[best_method]

# Print the best-performing method and its accuracy
print(f"The best-performing method is {best_method} with an accuracy of {best_accu
```

```
The best-performing method is Decision Tree with an accuracy of 0.89
```

The best performing method on test data is Decision Tree with an accuracy of 0.89 as you can see in the screenshot.

Confusion Matrix



- The decision Tree algorithm is the best predictor .
- 1 False negative for successful booster landing and 1 false positive .

Conclusions

- From SpaceX API to web scraping data from Wikipedia, we developed a model to predict Falcon 9 successful booster recovery by selecting important features from the dataset .
- Payload mass of the rocket, orbit types , Launching sites , Booster Version are the most important factors determining the outcome of booster recovery .
- The decision tree model gave us the most accurate predictions with almost 90% accuracy
- The Falcon 9 Booster recovery success rate increases over time .
- SpaceX GEO/GTO flights are more risky compared to LEO/VLEO/ISSO orbits .
- The most of the cargo payloads are between 4000 kg and 6000 kg ranges.

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!