IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Federico Neri
3rd May 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

This project aimed to analyze SpaceX's Falcon 9 rocket launches using various methodologies, including data collection, data wrangling and exploratory data analysis (EDA) visualization, SQL queries for insights, geospatial analysis with Folium, creation of a dashboard application with SNlab, and predictive analysis using classification algorithms.

Data Collection

The data collection process involved gathering information from the SpaceX API and a Wikipedia page, enabling a structured approach to extract key details such as launch outcomes, payload masses, and launch sites.

Exploratory Data Analysis

EDA visualization provided insights into launch patterns and relationships between variables such as flight number, payload mass, and launch success rates. SQL queries further enriched the analysis by revealing unique launch sites, notable achievements, and instances of success and failure based on different criteria.

Geospatial Analysis

Geospatial analysis with Folium allowed for visualizing launch sites, proximity to landmarks, and spatial relationships between various points of interest. The use of markers, circles, marker clusters, and polylines facilitated a comprehensive understanding of geographical patterns.

Dashboard application

The creation of a dashboard application using SNlab enabled interactive exploration of SpaceX launch records, with features such as dropdown menus, pie charts, range sliders, and scatter plots providing insights into success rates, payload distributions, and mission outcomes across different launch sites and booster versions.

Predictive analysis

Predictive analysis using classification algorithms involved data loading, preprocessing, model selection, hyperparameter tuning, evaluation, and comparison. The Decision Tree model emerged as the best performer based on training accuracy, although all models showed comparable performance on testing data, indicating their effectiveness in classifying mission outcomes.

Introduction

The project was initiated to analyze SpaceX's Falcon 9 rocket launches, aiming to gain insights into launch patterns, success rates, and factors influencing mission outcomes. Understanding launch features relationship is crucial for optimizing mission planning and enhancing overall efficiency.

Key problems:

- Understanding the success factors behind Falcon 9 launches: The project aimed to identify patterns and trends in successful and failed launches to determine key factors contributing to mission success.
- Analyzing payload characteristics: By examining payload masses and their relationship with launch outcomes, the project sought to understand how payload attributes influence mission success.
- Exploring geographical considerations: Geospatial analysis was conducted to investigate the impact of launch site locations, proximity to landmarks, and spatial relationships on launch outcomes.
- Building predictive models: The project aimed to develop classification models to predict mission outcomes based on various features, providing insights for future mission planning and risk assessment.

Section 1

Methodology

Methodology

Summary

- ➔ Data collection methodology
  - ◆ REST API call, webscraping
- ➔ Perform data wrangling
  - ◆ missing values, dummy variables, outcome label
- ➔ Perform exploratory data analysis (EDA) using visualization and SQL
  - ◆ scatterlots and other visualization, sql queries
- ➔ Perform interactive visual analytics using Folium and Plotly Dash
  - ◆ geospatial analysis, maps, markers, dashboards
- ➔ Perform predictive analysis using classification models
  - ◆ ML models building, refinement, evaluation

Data Collection Overall - spaceX api and wikipedia

The data for this project was collected primarily from the SpaceX API, which provides comprehensive information about Falcon 9 rocket launches. Various attributes such as booster version, payload mass, orbit, launch site, outcome of the launch, and other relevant details were extracted from the API.
Efficient extraction of historical launch data from a Wikipedia page by use of BeautifulSoup. This process involved parsing HTML tables, extracting relevant column names, and populating a Pandas dataframe with key information such as flight number, launch site, payload details, orbit, customer, and launch outcome. The structured approach ensured accurate data collection, laying the foundation for further analysis and insights into SpaceX's launch history.

Data Collection – SpaceX API

- **KEYWORDS**: Data Collection, SpaceX API, Falcon 9, Booster version, Payload mass, Orbit, Launch site, Launch outcome, Wikipedia, BeautifulSoup, HTML tables, Pandas dataframe, Flight number.

- GitHub URL : [IBM_DS_CapstoneProject/0.mod_1_jupyter-labs-spacex-data-collection-api.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)]

The data for this project was collected primarily from the SpaceX API, which provides comprehensive information about Falcon 9 rocket launches. Various attributes such as booster version, payload mass, orbit, launch site, outcome of the launch, and other relevant details were extracted from the API.
Efficient extraction of historical launch data from a Wikipedia page by use of BeautifulSoup. This process involved parsing HTML tables, extracting relevant column names, and populating a Pandas dataframe with key information such as flight number, launch site, payload details, orbit, customer, and launch outcome. The structured approach ensured accurate data collection, laying the foundation for further analysis and insights into SpaceX's launch history.

Data Collection – SpaceX API - Flowchart

Import Libraries and Define Auxiliary Functions	Request and Parse SpaceX Launch Data	Extract From Launch Data	Construct Dataset	Data Wrangling
Import requests, pandas, numpy, datetime libraries Define helper functions for extracting data from SpaceX API using identification numbers in the launch data	Make HTTP GET request to SpaceX API to retrieve launch data Parse JSON response and convert it into a Pandas dataframe Filter and preprocess the dataframe to extract relevant features and convert date format	Extract booster version, launch site details, payload data, and core data from the dataframe using API calls Store the extracted information in lists for further processing	Combine the extracted information into a dictionary Create a Pandas dataframe from the dictionary to represent the launch dataset Filter the dataframe to include only Falcon 9 launches	Handle missing values in the PayloadMass column by replacing them with the mean value Export the cleaned dataset to a CSV file for further analysis

Data Collection - Web Scraping

- **KEYWORDS**: Web scraping, BeautifulSoup, Pandas data frame, HTML table, HTTP GET method, Table parsing, Extract column names, Launch records, Flight No., Launch site, Payload mass, Orbit, Launch outcome, Version Booster, Booster landing, Date and time

- GitHub URL: IBM_DS_CapstoneProject/1.mod_1_jupyter-labs-spaceX-webscraping-checkpoint.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)

Data Collection – Web Scraping - Flowchart

Web Scraping Request	Parse HTML	Extract Table	Parse Table Data	Create DataFrame
Send an HTTP GET request to the Wikipedia page containing Falcon 9 launch records.	Use BeautifulSoup to parse the HTML content of the response and create a BeautifulSoup object.	Find and extract the Falcon 9 launch records table from the parsed HTML.	Iterate through the table rows to extract relevant information such as flight number, launch site, payload, etc., and store them in a dictionary	Convert the dictionary into a Pandas DataFrame and export it to a CSV file for further analysis.

Data Wrangling

- Key steps included identifying missing values, exploring numerical and categorical columns, and calculating the frequency of launches, orbits, and mission outcomes. Additionally, the lab involved creating a landing outcome label based on the mission's success or failure.
- The analysis revealed that approximately 66.67% of launches were successful in landing the first stage. This information will serve as a crucial factor for training supervised models in subsequent analyses.

- GitHub URL:
  IBM_DS_CapstoneProject/2.mod_1_labs-jupyter-spacex-Data wrangling.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)

Data Wrangling – Flowchart

Import Libraries

Pandas and numpy libraries for data manipulation and analysis

Data Analysis

Load the Space X dataset and perform initial data analysis, including identifying missing values and data types.

Calculate Launch Site Counts

Determine the number of launches for each launch site using the value_counts() method on the LaunchSite column.

Calculate Orbit Occurrences

Use the value_counts() method on the Orbit column to determine the number and occurrence of each orbit.

Landing Outcome Label

Based on the Outcome column, create a binary classification variable (Landing_Class) where zero indicates unsuccessful landing outcomes and one indicates successful landing outcomes.

EDA with Data Visualization

FlightNumber vs. PayloadMass: This scatter plot was used to observe the relationship between the flight number (indicating continuous launch attempts) and the payload mass. It helped in understanding how the first stage's success rate relates to these variables. It was observed that as the flight number increases, the first stage is more likely to land successfully. Additionally, the plot showed that heavier payloads are less likely to result in a successful landing.
FlightNumber vs. LaunchSite: This scatter plot was used to visualize the relationship between the flight number and the launch site. It helped in understanding any patterns or trends in launch outcomes based on the launch site.
Payload vs. LaunchSite: This scatter plot was used to explore the relationship between the payload mass and the launch site. It revealed insights into which launch sites are preferred for different payload masses.
Success Rate by Orbit Type: A bar chart was plotted to visualize the success rate of each orbit type. This helped in understanding which orbit types have higher success rates.
FlightNumber vs. Orbit type: A scatter plot was used to examine the relationship between the flight number and the orbit type. It helped in understanding if there is any relationship between the flight number and the type of orbit.
Payload vs. Orbit type: This scatter plot was used to explore the relationship between the payload mass and the orbit type. It provided insights into how the payload mass affects the success rate for different orbit types.
Average Launch Success Rate Over the Years: A line chart was plotted to visualize the average launch success rate over the years. It helped in understanding the trend of launch success over time.

- GitHub URL: [IBM_DS_CapstoneProject/4.mod_2_edadataviz.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)]

EDA with SQL/1

- **Unique launch sites** include CCAFS LC-40, VAFB SLC-4E, and KSC LC-39A.
  Query:
  ```
  SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
  ```
- Notable achievements include the **first successful landing** on a ground pad on December 22, 2015.
  Query:
  ```
  SELECT MIN("Date") AS FirstSuccessfulLandingDate
  FROM SPACEXTABLE
  WHERE "Landing_Outcome" = 'Success (ground pad)';
  ```
- Boosters achieving success on drone ships with **payload masses between 4000 and 6000 kg** include F9 FT B1022, F9 FT B1026, among others.
  Query:
  ```
  SELECT "Booster_Version" FROM SPACEXTBL
  WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;
  ```
- The dataset comprises records of both **successful and failed mission outcomes,** with varying payloads.

Query:
```
SELECT DISTINCT Mission_Outcome FROM SPACEXTBL;
```

EDA with SQL/2

- **Booster versions carrying the maximum payload** mass include F9 B5 B1048.4, F9 B5 B1049.4, F9 B5 B1051.3, and others.
  Query:
  ```
  SELECT "Booster_Version", "PAYLOAD_MASS__KG_"
  FROM SPACEXTBL
  WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
  ```
- Instances of **failure landing outcomes** on drone ships, booster versions, and launch sites were identified for the months in the year 2015.
  Query:
  ```
  SELECT strftime('%m', "Date") AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
  FROM SPACEXTBL
  WHERE "Landing_Outcome" LIKE 'Failure%' AND substr("Date", 0, 5) = '2015';
  ```
- The **count of landing outcomes between June 4, 2010, and March 20, 2017,** varied, with successes including drone ship landings and ground pad landings, as well as failures like parachute landings and controlled/ocean landings.
  Query:
  ```
  SELECT "Date", "Landing_Outcome", COUNT(*) AS OutcomeCount
  FROM SPACEXTBL
  WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
  GROUP BY "Landing_Outcome"
  ORDER BY Date DESC;
  ```

Github url: [IBM_DS_CapstoneProject/3.mod_2_jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)]

Build an Interactive Map with Folium

- **Markers**: Markers were added to represent the launch sites. Each marker indicates a specific launch site on the map. The markers were customized with labels showing the name of the launch site. Additionally, markers were used to represent the closest city, railway, and highway to the launch site. These markers were also customized with labels showing the distance from the launch site to each point of interest.
- **Circles**: were added to highlight areas around specific coordinates. In the example provided, a circle was created around NASA Johnson Space Center's coordinate to highlight its location on the map. The circle was filled with a color and included a popup label showing the name of the location.
- **Marker Clusters**: Marker clusters were utilized to simplify the map when there are multiple markers at the same location. This helps in maintaining map readability and reducing clutter. Marker clusters were used to group markers representing launch outcomes (success or failure) for each site. The color of the markers was determined based on the launch outcome (green for successful launches, red for failed launches).
- **Polyline**: Polyline was drawn between the launch site and the closest coastline point. This polyline represents the distance between the launch site and the coastline. Similarly, polylines can be drawn between the launch site and other points of interest such as cities, railways, and highways.

Analysis of launch site locations, proximity to various landmarks, and distances between different points of interest come in hand for understanding geographical patterns and exploring the spatial relationships between launch sites and surrounding areas.

Github url: [IBM_DS_CapstoneProject/5.mod_3_lab_jupyter_launch_site_location.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)]

Build a Dashboard with Plotly Dash

- **Dropdown Menu:** A dropdown menu was added to allow users to select a specific launch site. This dropdown menu enables users to filter the data based on launch sites, providing them with the flexibility to focus on data specific to a particular site of interest.
- **Pie Chart**: A pie chart was added to visualize the distribution of success and failure launches. When all sites are selected, the pie chart displays the total success launches across all sites. When a specific site is selected from the dropdown menu, the pie chart shows the success rate (percentage of successful launches) for that particular site.
- **Range Slider**: A range slider was added to enable users to filter the data based on payload mass. This slider allows users to define a payload mass range, helping them focus on launches within a specific payload range of interest.
- **Scatter Plot**: A scatter plot was added to visualize the relationship between payload mass and mission outcome. The scatter plot displays each launch as a point, with payload mass on the x-axis and mission outcome (success or failure) on the y-axis. Additionally, the scatter plot is color-coded based on the booster version category, providing further insights into the relationship between payload mass, mission outcome, and booster version.

These plots and interactions were added to enhance the dashboard's functionality and usability, allowing users to interactively explore SpaceX launch records, analyze success rates, investigate payload mass distributions, and identify patterns in mission outcomes across different launch sites and booster versions.

Github url: IBM_DS_CapstoneProject/6.mod3_spacex_dash_app.py at main · fener95/IBM_DS_CapstoneProject (github.com)

Predictive Analysis – Flowchart

Data Preparation and Labeling	Data Standardization and Splitting	Hyperparameter Tuning	Model Evaluation	Model Selection
Perform Exploratory Data Analysis (EDA) and determine training labels, including creating a column for the class in the dataset.	Standardize the data and split it into training and test datasets.	Utilize GridSearchCV to find the best hyperparameters for classification algorithms (**Logistic Regression, Support Vector Machine, Decision Tree, and K Nearest Neighbors**).	Evaluate the performance of each classification algorithm on the test dataset using the best hyperparameters obtained from GridSearchCV.	Compare the performance of different classification models based on their accuracy scores on the test dataset and select the one with the highest accuracy as the best performing classification model.

Results from EDA, interactive analytics, Predictive analysis

The exploratory data analysis revealed insights into the SpaceX launch records dataset, showcasing trends in mission outcomes, payload masses, and launch sites. Interactive analytics demonstrated in screenshots showcased features such as filtering by launch site and payload range, with visualizations like pie charts and scatter plots aiding in understanding success rates and payload outcomes.

Predictive analysis using classification algorithms, including Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors, yielded accuracy scores ranging from 83.33% to 86.25% on training data and, however all models showed the same performance on new data prediction (testing data) with an accuracy of 83.33% ,indeed, highlighting their ability to classify mission outcomes effectively. Confusion matrices provided further insights into model performance, aiding in the selection of the Decision Tree model as the best performer based on training accuracy.

However, all models achieved the same testing accuracy, indicating their comparable performance on unseen data.

Github url: IBM_DS_CapstoneProject/7.mod4_SpaceX_Machine Learning Prediction_Part_5.ipynb at main · fener95/IBM_DS_CapstoneProject (github.com)

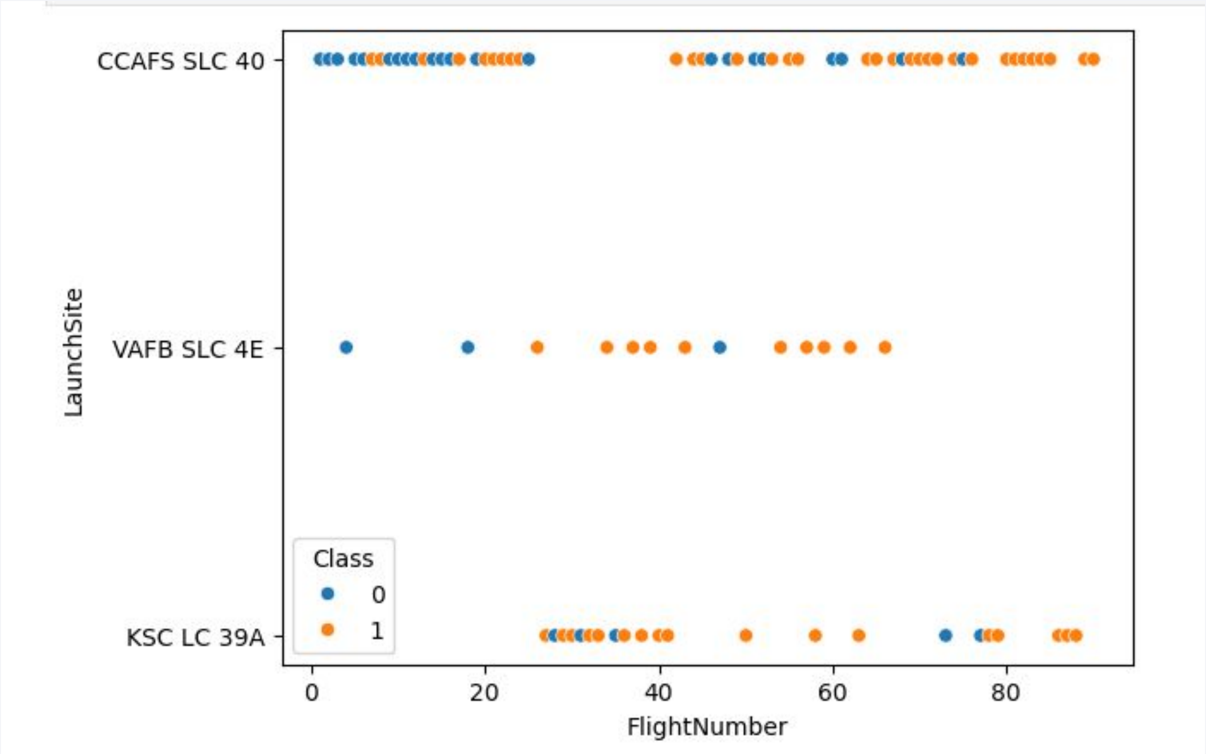
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Here we appreciate the relationship between number of flights performed and the specific launch site bases; where class ‘0’ and ‘1’ individuate, respectively, failed and succeeded missions.

We may visualize that a positive relationship might exists between no of flights and one mission’s positive outcome.

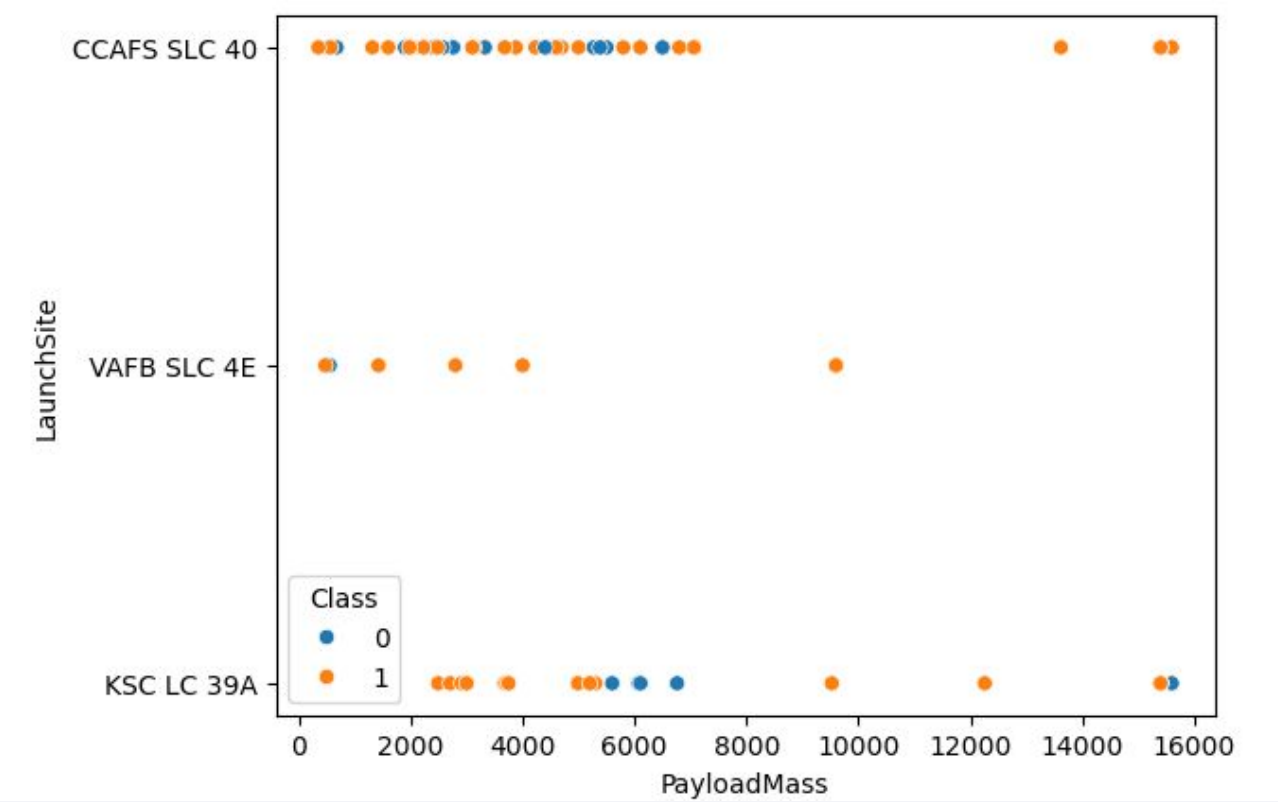


Payload vs. Launch Site

Here we appreciate the relationship between payload mass (Kg) and the specific launch site bases; where class ‘0’ and ‘1’ individuate, respectively, failed and succeeded missions.

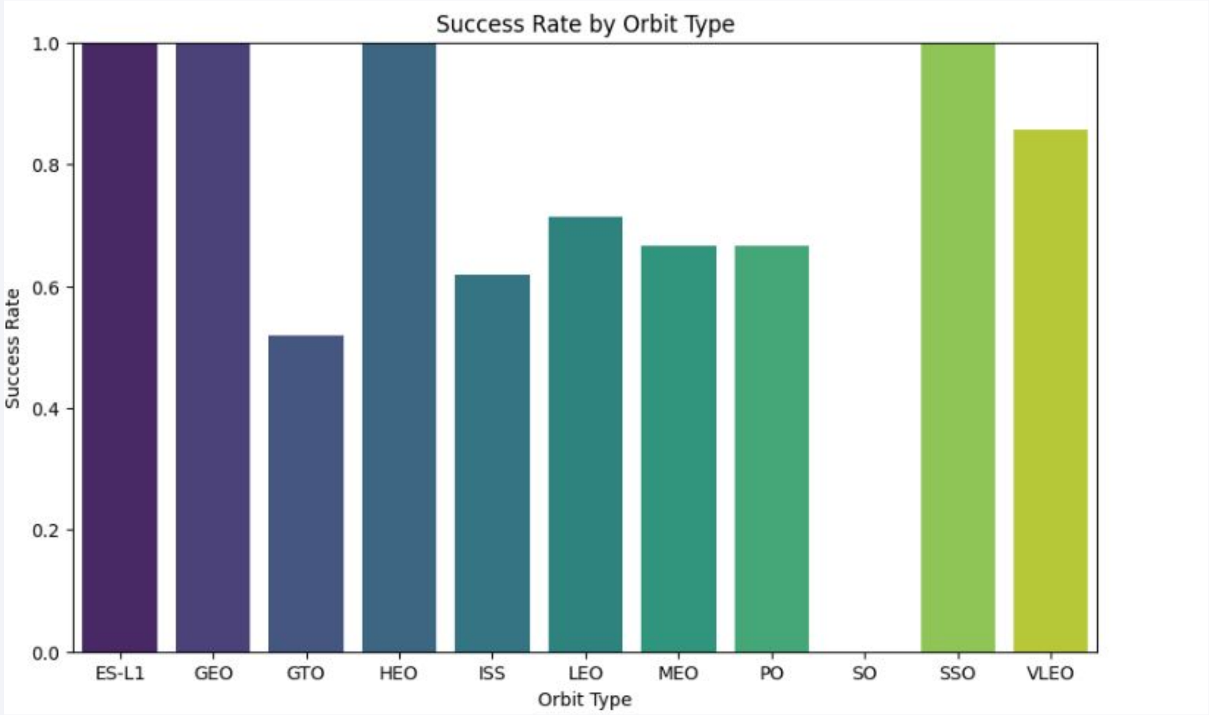
We may visualize that unsuccess is probably related with low-to-middle payload mass (4K - 7K)

Only a few data about launches with higher ranges of payloads(14K-16K) are present.



Success Rate vs. Orbit Type

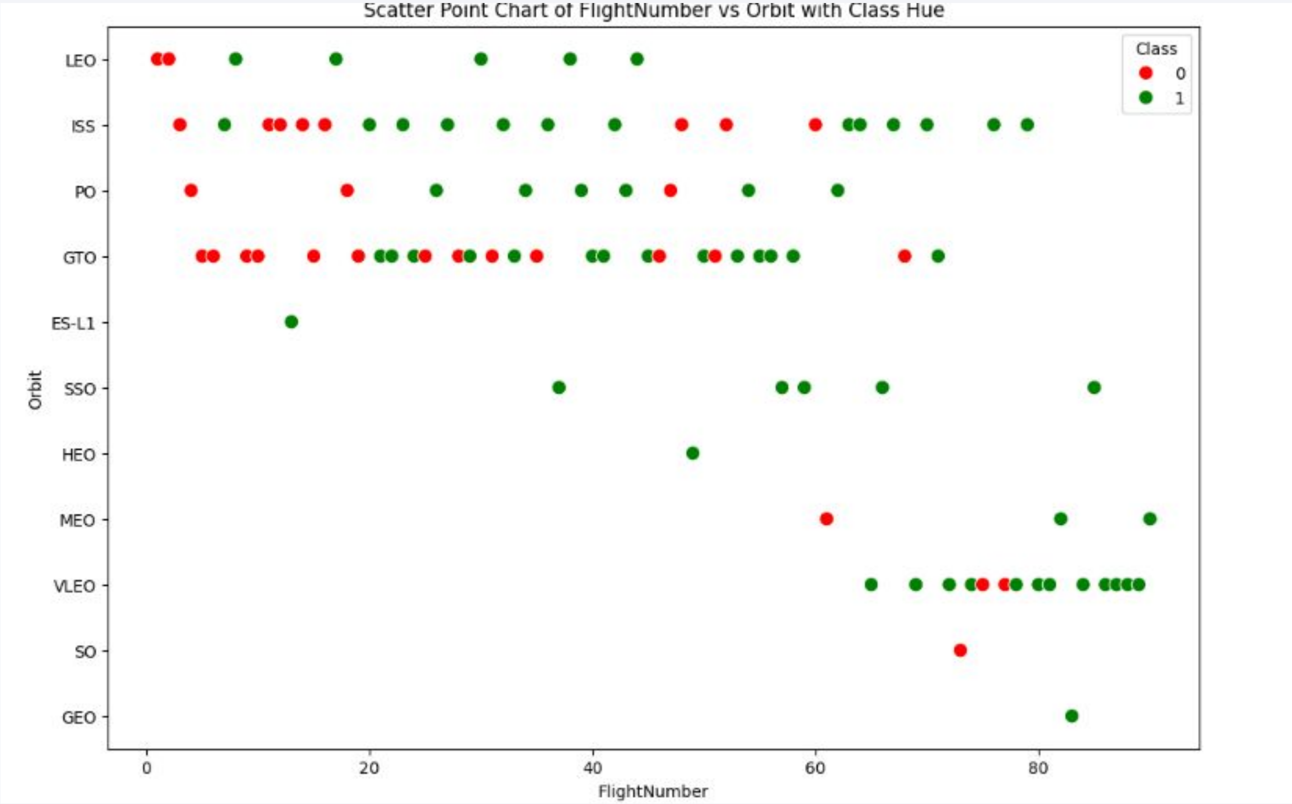
- Higher success rate seems visually strongly related to the orbit type, as displayed here, hence variability in the result is presented with one orbit at zero success rate, an half with highest (1) success rate and an other half with around 50-60% frequency of success.



Flight Number vs. Orbit Type

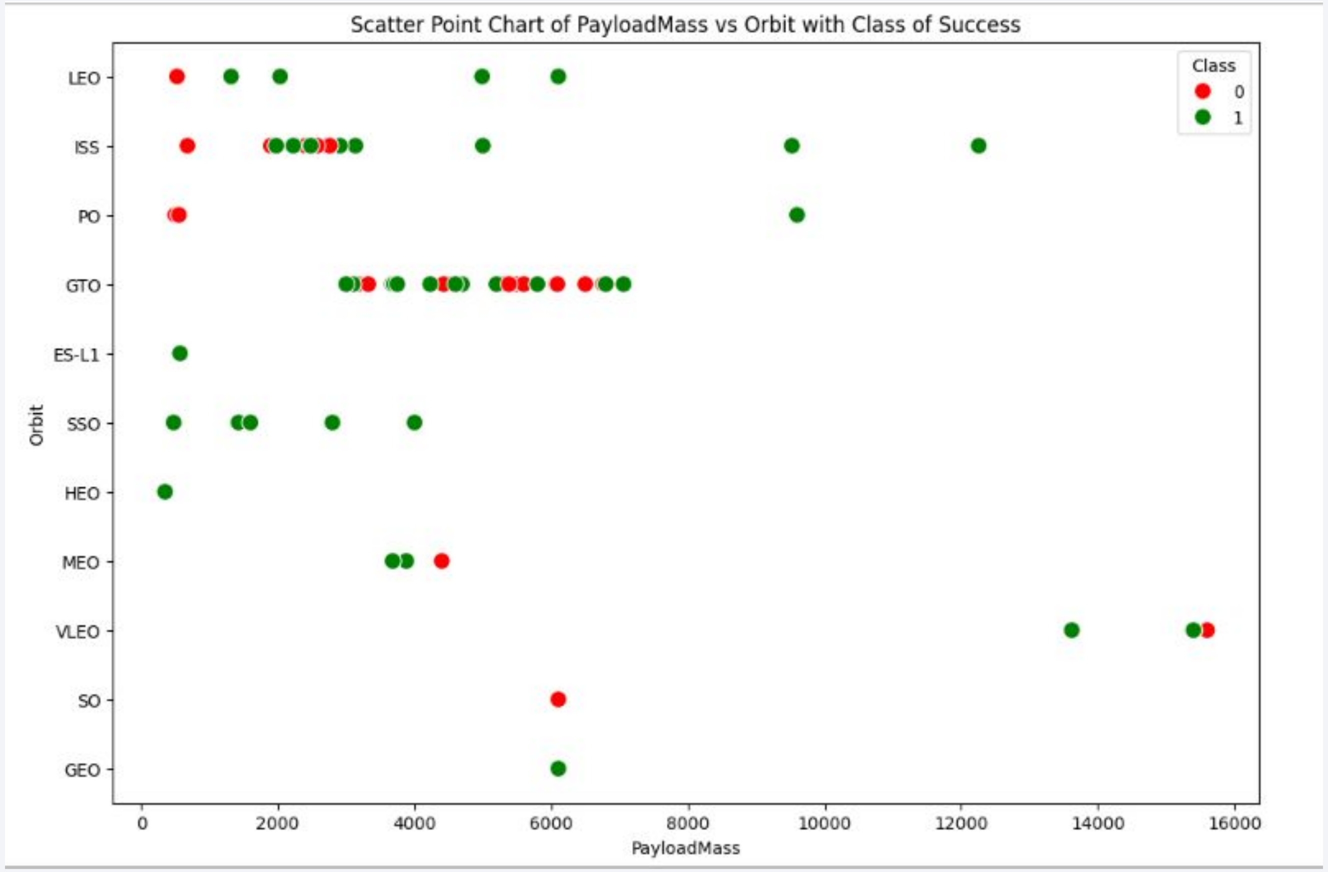
- Here we appreciate the relationship between flight number Vs orbit type; where class ‘0’ and ‘1’ individuate, respectively, failed and succeeded missions.

- Note that all orbits are experimented only after several flights; at the early stage, with low flight number, less than half of the orbit are exploited.Indeed, success increase always with higher no of flights performed.



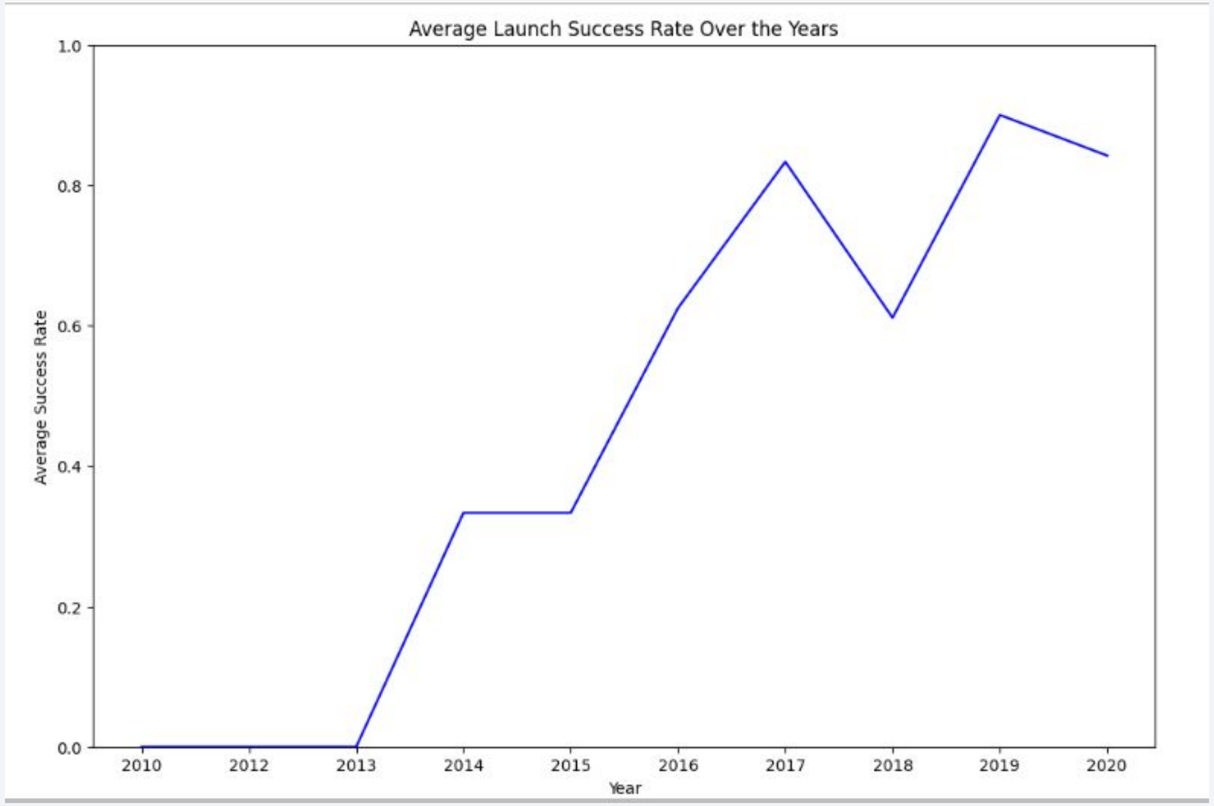
Payload vs. Orbit Type

- The scatterplot of payload vs orbit type (with green : success and red : failure), highlights as some payload ranges are more frequently experimented in specific orbit types.



Launch Success Yearly Trend

- Generally the average success rate increased overtime, despite a single drop in 2018.



All Launch Site Names

Here we may appreciate the sql queries to obtain launch sites names.

```
[9]: %sql SELECT DISTINCT "Launch_Site" \
     FROM SPACEXTABLE;

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

[9]:

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

Launch Site Names Filtering

SQL queries to find 5 launch sites names starting with ‘CCA’

```
[10]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%'\
      LIMIT 5;

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

[10]:

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 Filtering & Calculation

- SQL queries to calculate the total payload carried by boosters from NASA (CRS)

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

```
]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS NASA_CRS_TotalPayloadMass\
FROM SPACEXTBL\
WHERE "Customer" like '%NASA (CRS)%';
```

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

NASA_CRS_TotalPayloadMass
48213

Average Payload Mass by F9 v1.1

- SQL queries to calculate the average payload mass carried by booster version F9 v1.1

Task 4

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

```
[12]: %sql SELECT round(AVG(PAYLOAD_MASS__KG_),3) AS  F9_v1_1_AVG_PayloadMass\
      FROM SPACEXTBL\
      WHERE Booster_Version like 'F9 v1.1';
```

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

[12]:

F9_v1_1_AVG_PayloadMass
2928.4

First Successful Ground Landing Date

The dates of the first successful landing outcome on ground pad with sql queries: 22nd Dec 2015.

Task 5

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

Hint:Use min function

[13]:
```
%sql SELECT MIN("Date") AS FirstSuccessfulLandingDate\
FROM SPACEXTABLE\
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

[13]:

FirstSuccessfulLandingDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL queries to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

```
[14]: %sql SELECT "Booster_Version" FROM SPACEXTBL\
      WHERE "Landing_Outcome" = 'Success (drone ship)'\
        AND "PAYLOAD_MASS__KG_" > 4000\
        AND "PAYLOAD_MASS__KG_" < 6000;

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

[14]:

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

Total Number of Successful and Failure Mission Outcomes

- SQL queries to calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
[15]: %sql select distinct Mission_Outcome from SPACEXTBL
```

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

[15]:

Mission_Outcome
Success
Failure (in flight)
Success (payload status unclear)
Success

Boosters Carried Maximum Payload

SQL queries to select the names of the booster which have carried the maximum payload mass (15.600 Kg)

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

```
6]: %sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" \
FROM SPACEXTBL\
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
```

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

6]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- SQL queries to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Task 9

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 strftime('%m', "Date") AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site"\
FROM SPACEXTBL\
WHERE "Landing_Outcome" LIKE 'Failure%' AND substr("Date", 0, 5) = '2015';
```

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

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

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

SQl queries to 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

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

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

[18]:

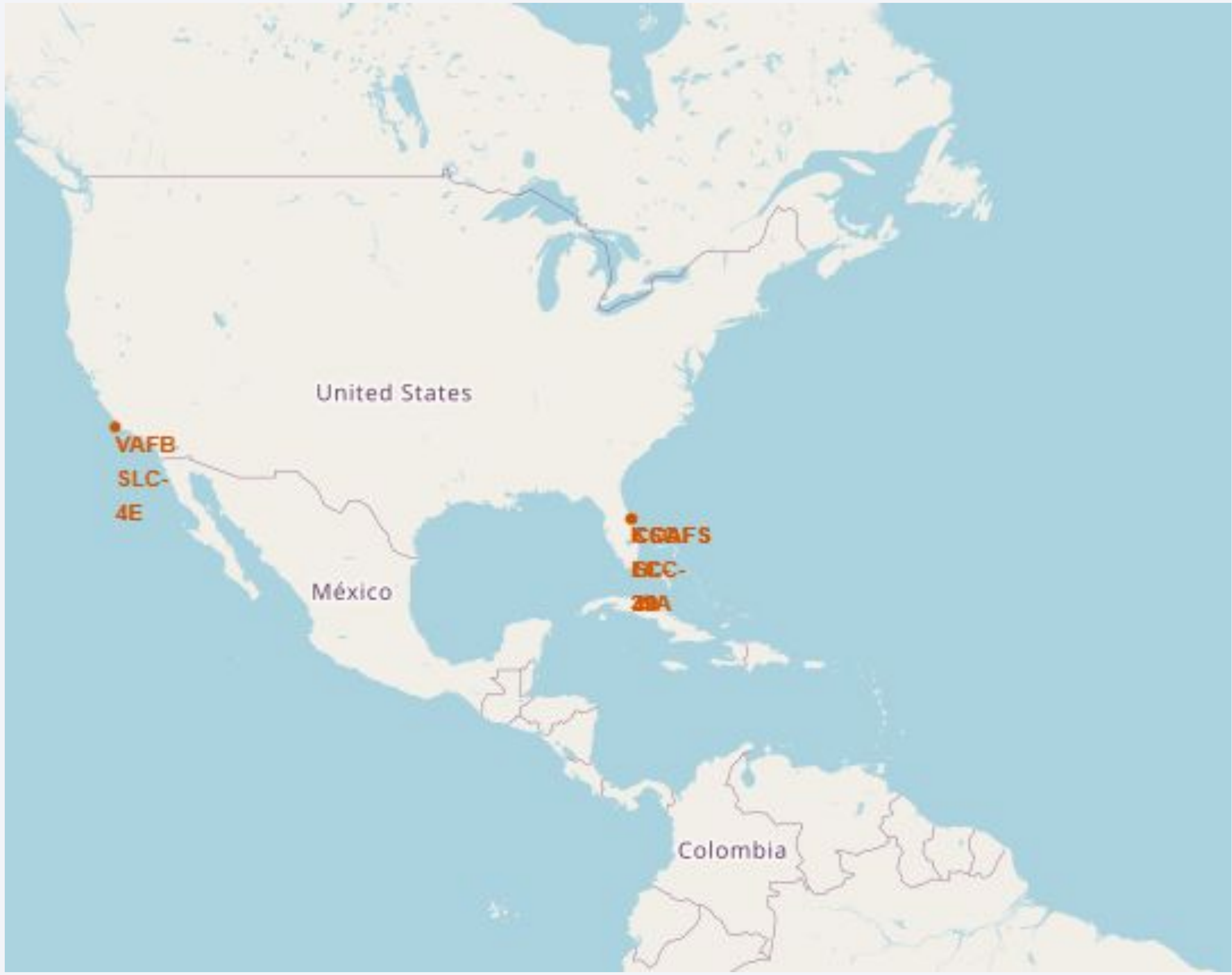
Date	Landing_Outcome	OutcomeCount
2016-04-08	Success (drone ship)	5
2015-12-22	Success (ground pad)	3
2015-06-28	Precluded (drone ship)	1
2015-01-10	Failure (drone ship)	5
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2012-05-22	No attempt	10
2010-06-04	Failure (parachute)	2

Section 3

Launch Sites Proximities Analysis

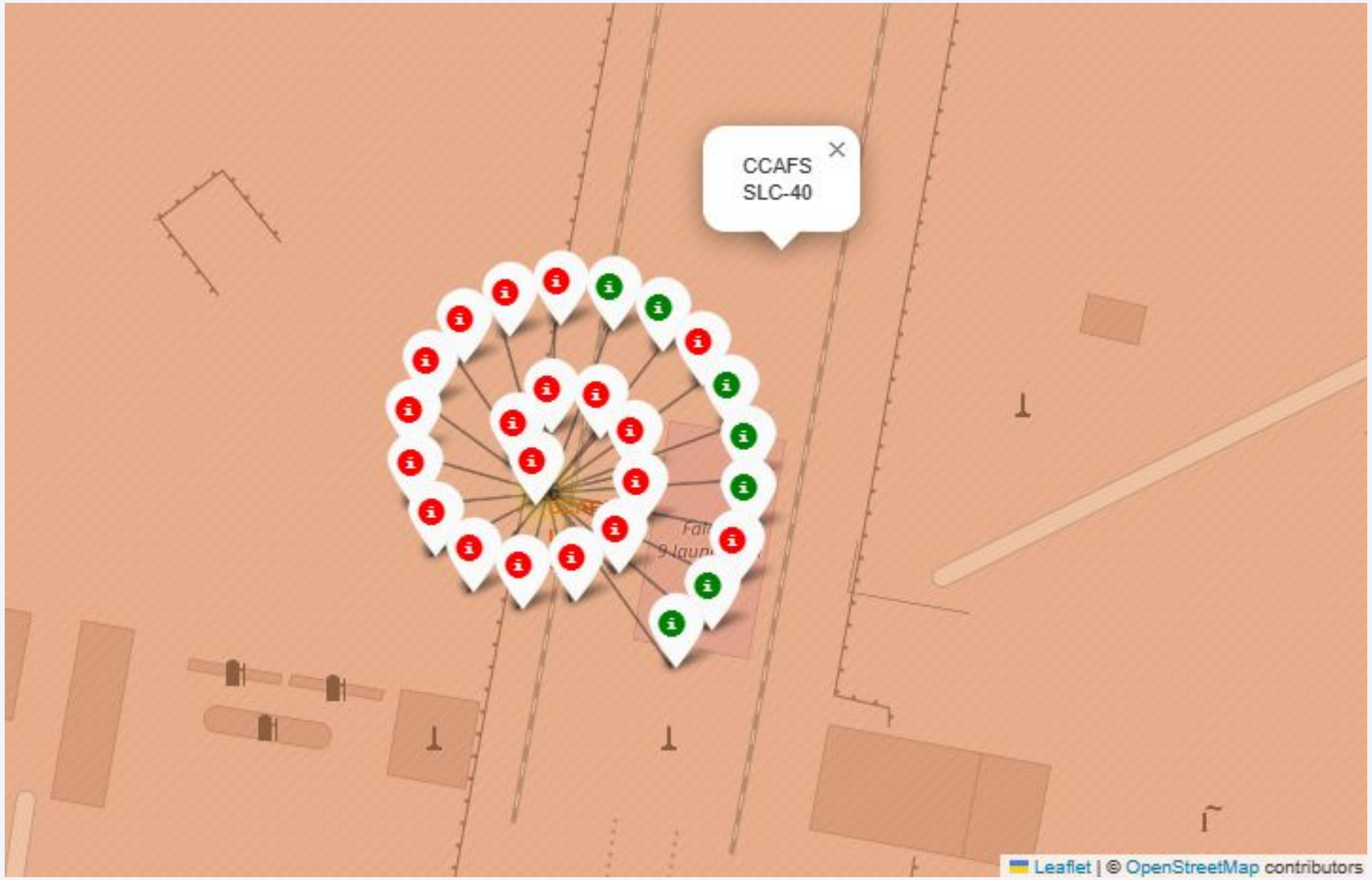
Launch Sites location

- Here we may see that all launch sites are in proximities of coast, east and west south coasts of USA.



Launch Sites Outcomes

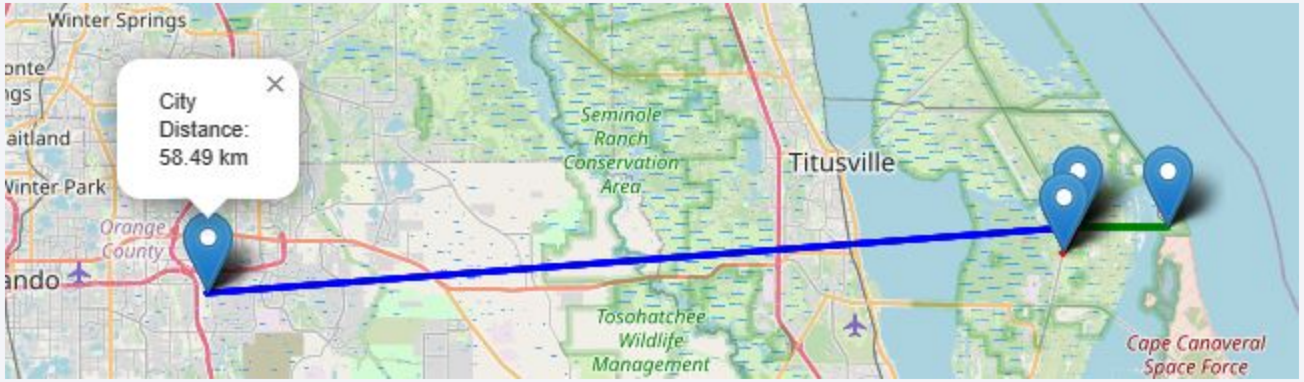
Here is the visually mapped success rate of the selected launch site CCAFS SLC-40. Red and green stands for fail and success outcome.



Launch Site Proximities

Leafmap or Folium are all python packages and libraries which permits to conduct from basic to more advanced geospatial analysis.

Here a simple straight line gives an idea of proximities to major points of interest (city of Orlando, coast, highway).

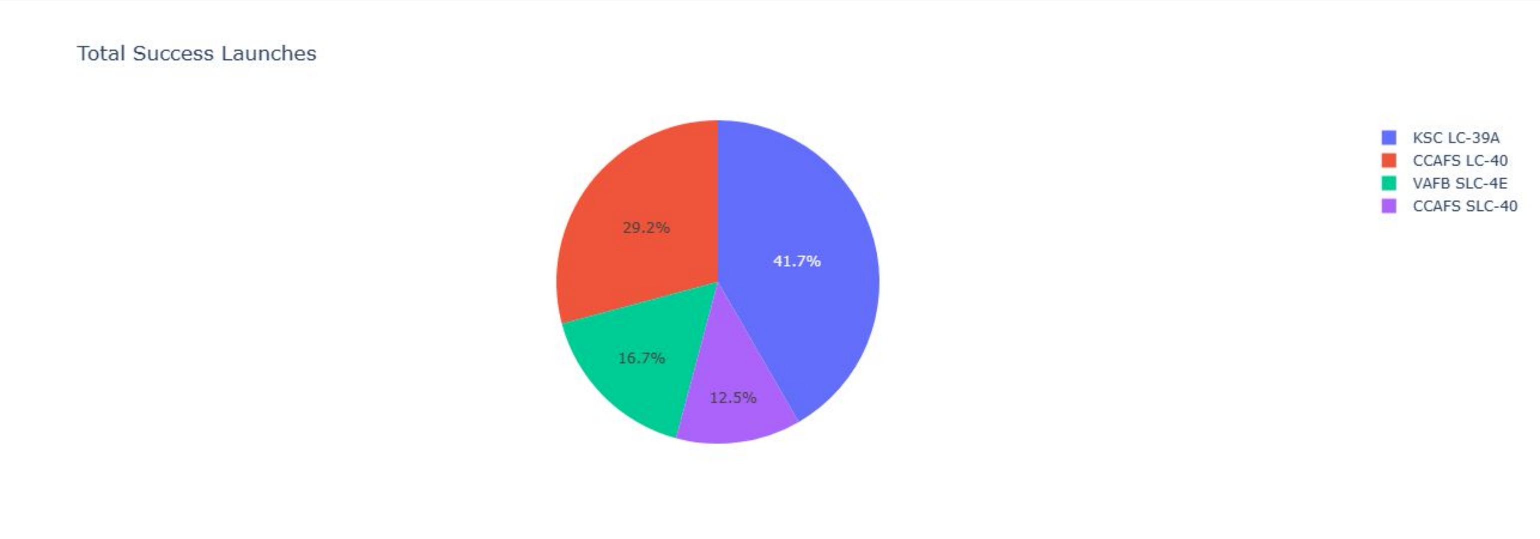


Section 4

Build a Dashboard with Plotly Dash

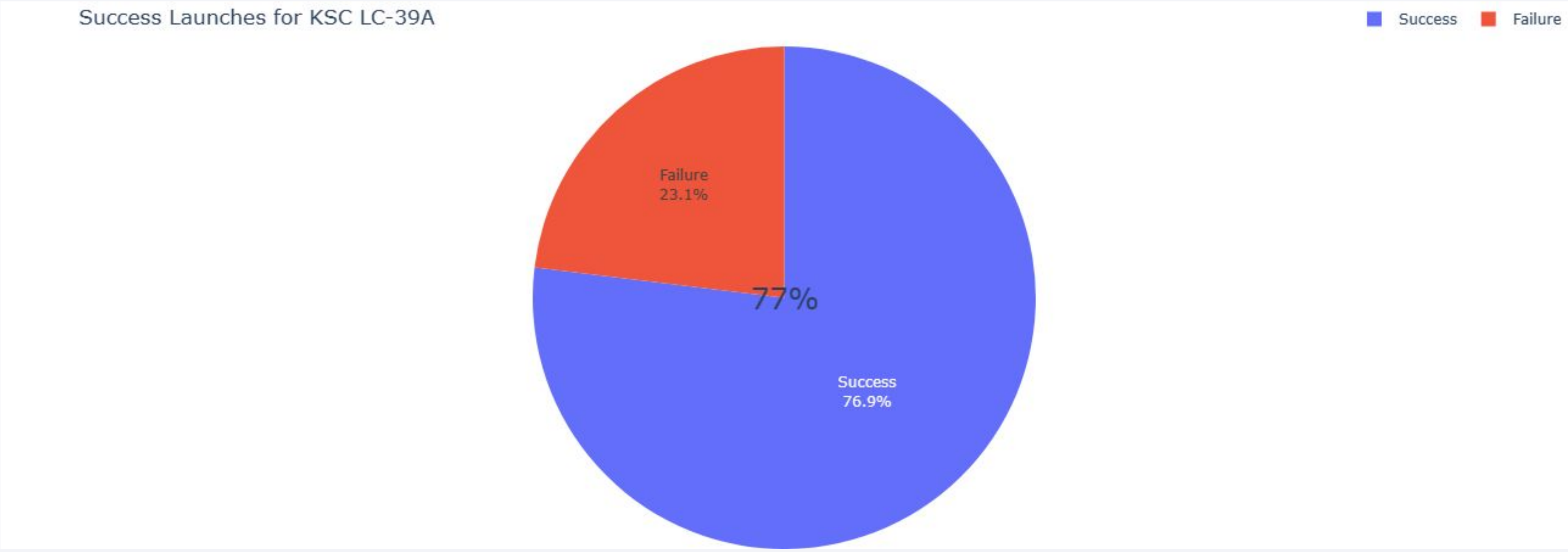
Dashboard All Sites Successful Launches

This pie-chart dynamically show the success rate per launch site(here we appreciate the all launch sites success proportions)

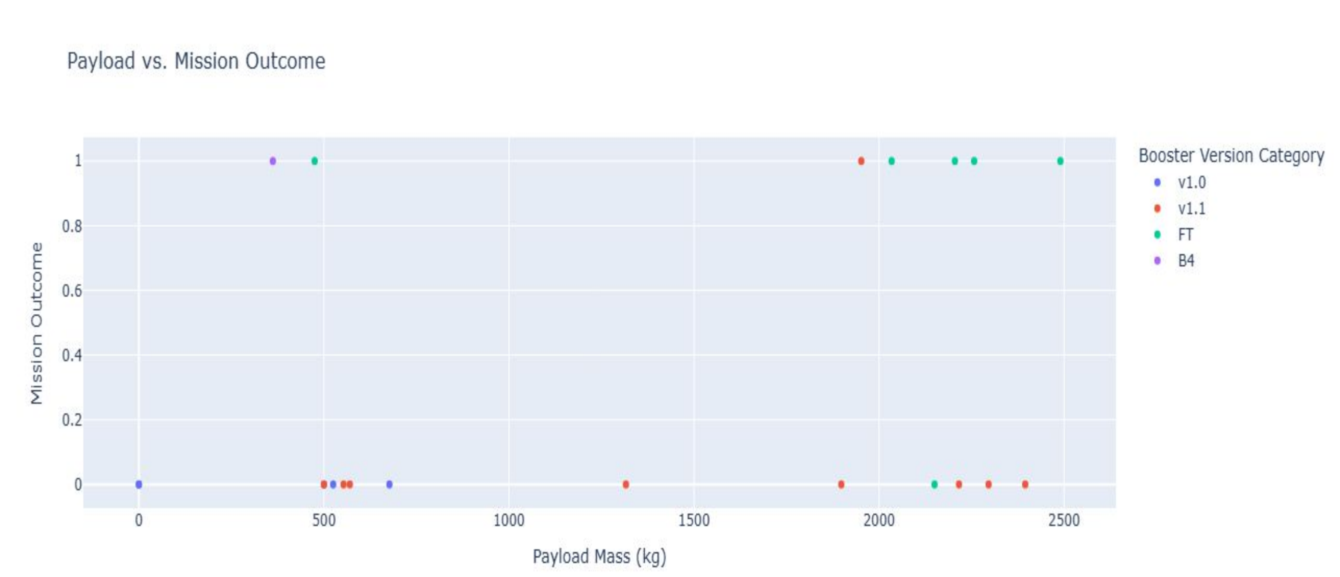


Highest Launch Success Rate: KSC-LC39A

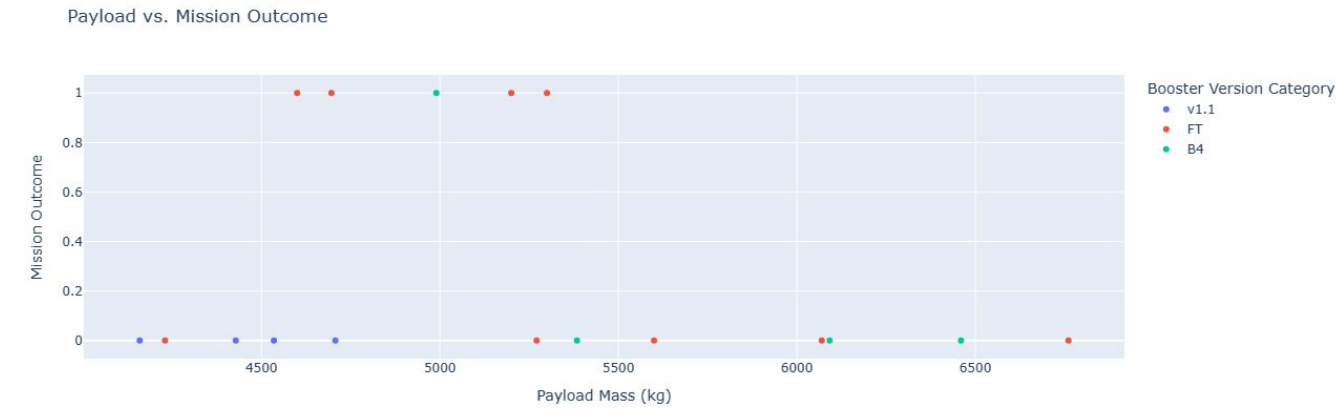
KSC-LC39A share the highest success outcome rate with around 77%.

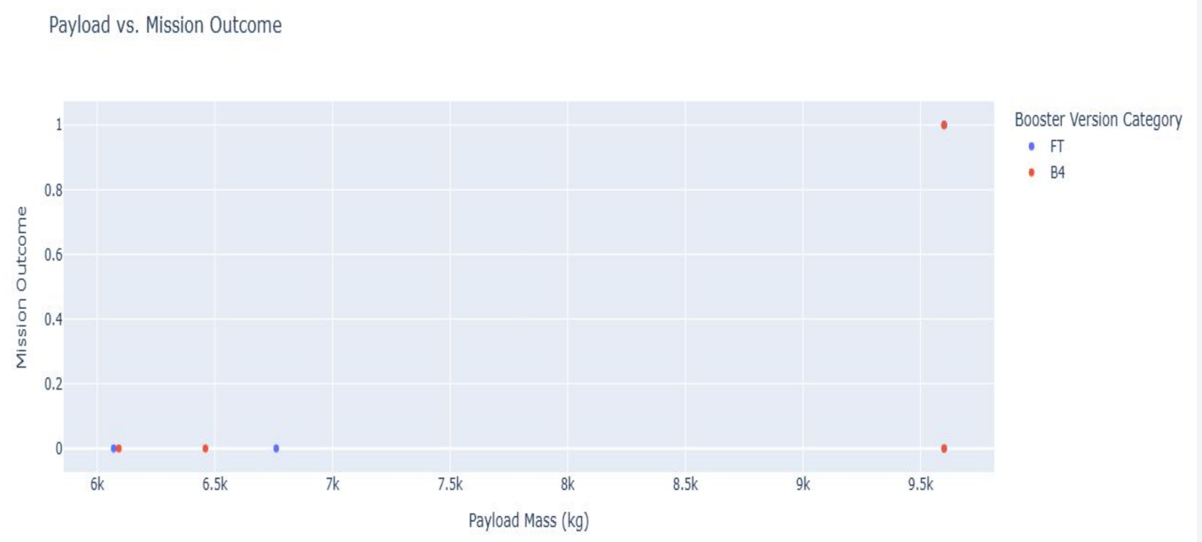


Payload Vs Launch Outcome



Applying the slider to select a custom payload range, first we notice that specific boosters release a positive outcome in relation to payload employed; furthermore, only B4 showed success at the highest payload range (bottom right corner of slide); while the majority of success for all boosters are in a range close to 2000-5500 kg.



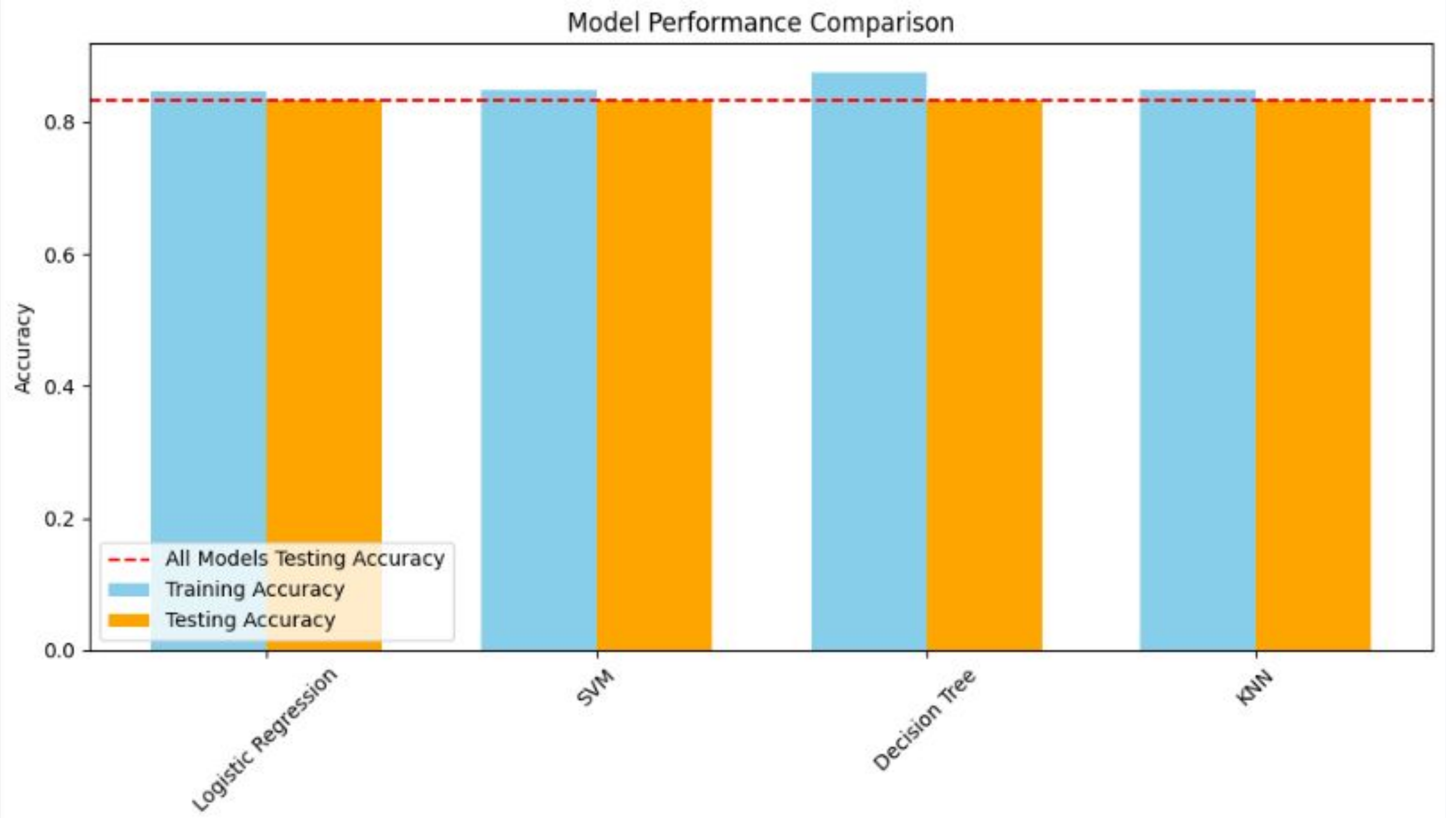


Section 5

Predictive Analysis (Classification)

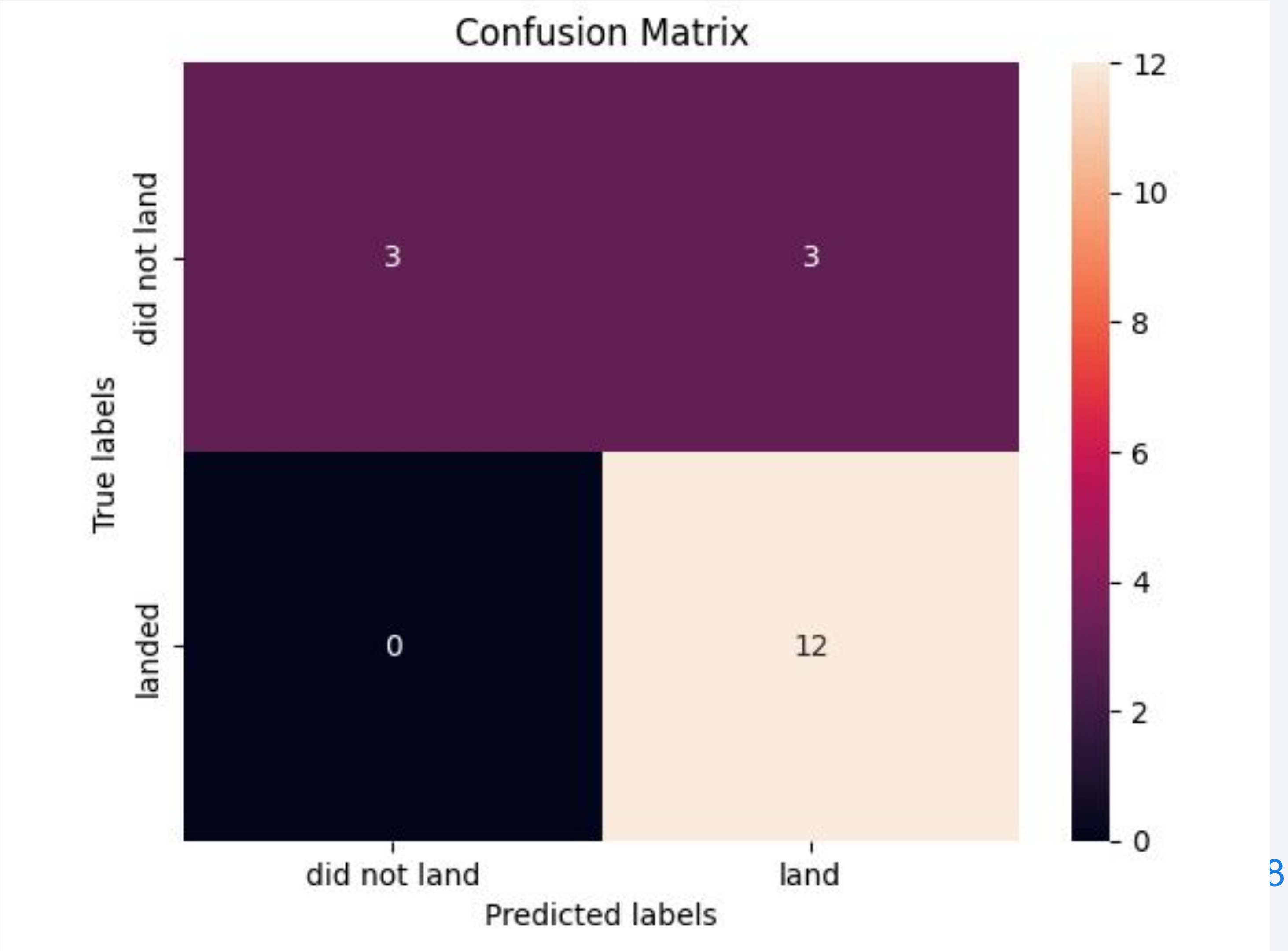
Classification Accuracy

All models performed almost equally and nicely; hence i selected **‘Decision Tree classifier**’ as best performing model, because at first training displayed better results, while classification accuracy was the same for all ML algorithm, at this early stage of analysis.



Confusion Matrix - decision tree

Here we see that the main concern and point of improvement for the model is : **false-positive**; notice that 3 out of 18 observations were wrongly predicted as successful.



Conclusions

1. SpaceX api in conjunction with web scraping of an official source are optimal means to carry an exploratory analysis and the huge amount of data available open to possible ML application for prediction in a supervised manner.

2. Among all the features of each observed flight, payload mass and booster version play a clearly pivotal role in the final outcome, furthermore, the specific launch site location seems to be relevant as well, though causal relationships are not investigated in this kind of analysis and hence further study from this side may elucidate the role launch site in mission outcome.

3. Geospatial features, displayed in a geo-referenced map may comes in hand to uncover some hidden relations among features, however spatial features were not inglobed in the ML predictive analysis, while they may constitute an expansion of the consideration of launch site location in the analysis, which is mostly uncovered at the moment step.
4. Finally, with an initial 83.33% classification accuracy, all the models, specifically the Decision Tree, are reasonably able to predict on next launch outcome indeed!

Thank you!