IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Physicist. Jorge Arturo Rubio Iñiguez
01/02/25

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Interactive Analytics with Folium
  - Machine Learning Predictions in Python
- Summary of all results
  - Interactive visual analysis results
  - Predictive Analytics results

Introduction

- Project background and context

The future of space exploration depends on the budget allocated to that sector, which is why it is important to reduce costs. SpaceX is achieving this by recovering the first stage of the rockets. Each Falcon 9 launch costs 62 million dollars, while the cost of other providers is 165 million dollars. For this reason, this project could help reduce costs or understand them before the launch takes place.

- Problems you want to find answers
    - What factors determine the success of a landing?.
    - The interaction among various features that determine the success rate of a successful landing.
    - The location of the launch affects the probability of success?.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Categorical data was manipulated with One-hot encoding
- 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

- Describe how data sets were collected.
  - Using get reguest to the SpaceX API
  - Next, we use de ,json() function to decode the response, then we turn it into pandas dataframe using de .json_normalize() function.
  - Then, we cleaned de data and checked for missing values, filled where necessary.
  - For the web scraping from Wikipedia, we used BeautifulSoup library.
  - The goal of this was to extract the launch records as HTML table and convert it to pandas dataframe.

Data Collection – SpaceX API

- I used the get request function to the SpaceX API to collect data, clean it and formatting for further analisis.

- https://github.com/xjorgerubiox/Final-Press/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[10]: response=requests.get(static_json_url)
```

```
[11]: response.status_code
```

```
[11]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[13]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
[17]: # Get the head of the dataframe
print(data.head(5))
```

```
       static_fire_date_utc  static_fire_date_unix    tbd    net  window  \
0  2006-03-17T00:00:00.000Z           1.142554e+09  False  False     0.0
1                      None                    NaN  False  False     0.0
2                      None                    NaN  False  False     0.0
3  2008-09-20T00:00:00.000Z           1.221869e+09  False  False     0.0
4                      None                    NaN  False  False     0.0

                     rocket  success  \
0  5e9d0d95eda69955f709d1eb    False
1  5e9d0d95eda69955f709d1eb    False
2  5e9d0d95eda69955f709d1eb    False
3  5e9d0d95eda69955f709d1eb     True
4  5e9d0d95eda69955f709d1eb     True
```

Data Collection - Scraping

- I used web scraping with BeautifulSoup to the Falcon 9 launch records From Wikipedia. And create de pandas dataframe by parsing the HTML table

- https://github.com/xjorgerubiox/Final-Press/blob/main/jupyter-labs-webscraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

[6]:
```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
if response.status_code == 200:
    print("Request successful!")
else:
    print(f"Request failed with status code: {response.status_code}")

# Display the response object
print(response)
```

```
Request successful!
<Response [200]>
```

Create a `BeautifulSoup` object from the HTML `response`

[7]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

[8]:
```
# Use soup.title attribute
print(f"Page title: {soup.title.string}")
```

```
Page title: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

[28]:
```
soup.title
```

[28]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

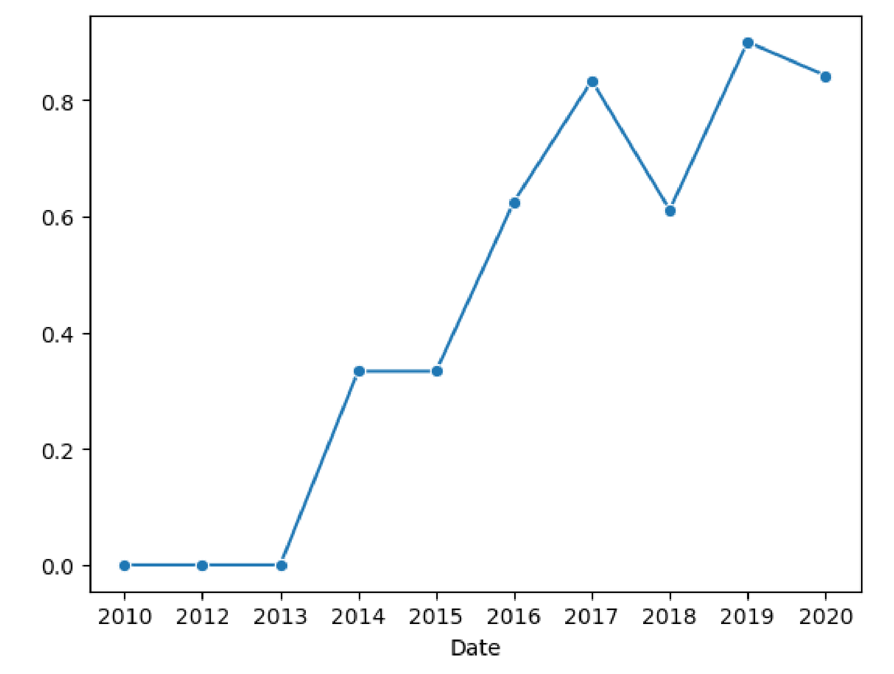
Data Wrangling

- I performed a exploratory Data analysis and determined the training labels.
- First I calculate the number of launches on each site and occurrence of each orbit.
- I create a landing outcome label from outcome column and exported the results to csv.

- https://github.com/xjorgerubiox/Final-Press/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- I explored the data by the relationship between the independent variables such as, launch site, payload, flight number, success rate of each orbit and orbit type, for example. The launch success yearly trend.



- https://github.com/xjorgerubiox/Final-Press/blob/main/edadataviz.ipynb

EDA with SQL

- First I loaded the SpaceX data into PostgreSQL.
- Then applied and executed SQL queries to answer questions such as:
  - Names of unique launch sites.
  - Total payload mass carried by boosters launches by NASA(CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Total number of successful and failure mission outcomes

- https://github.com/xjorgerubiox/Final-Press/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- I marked all launch sites and added map objects like markers and circles for each site on the folium map.
- Then was assigned the feature launch outcomes to class, 0 for failure, 1 for succers
- Was Used the marker clusters function to identify which launch sites have high success rate.
- Whit this, it can be answered some questions like:
    - Are launch sites near coastlines.
    - Are launch sites close to equator line.
- https://github.com/xjorgerubiox/Final-Press/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with plotly dash.
- Next, plotted pie charts showing the total launches.
- Plotted scatter graph showing the relationship with Outcome and Payload mass for different booster.
- Explain why you added those plots and interactions

Predictive Analysis (Classification)

- I imported the data and loaded to a pandas dataframe, then transformed the data, split it into training and testing.
- Then built different machine learning models and tune diferente hyperparamateres using GridSearchCV function.
- Finally found the best performing model using accuracy as the metric.

- https://github.com/xjorgerubiox/Final-Press/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

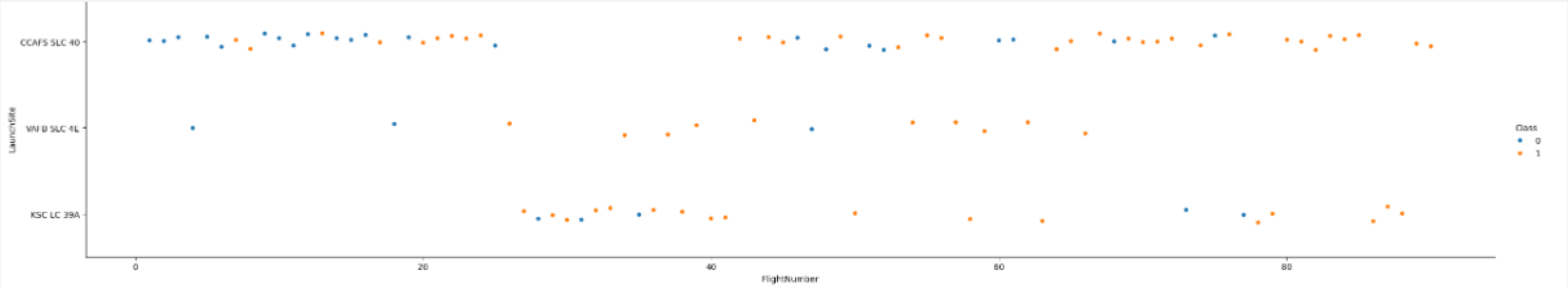
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

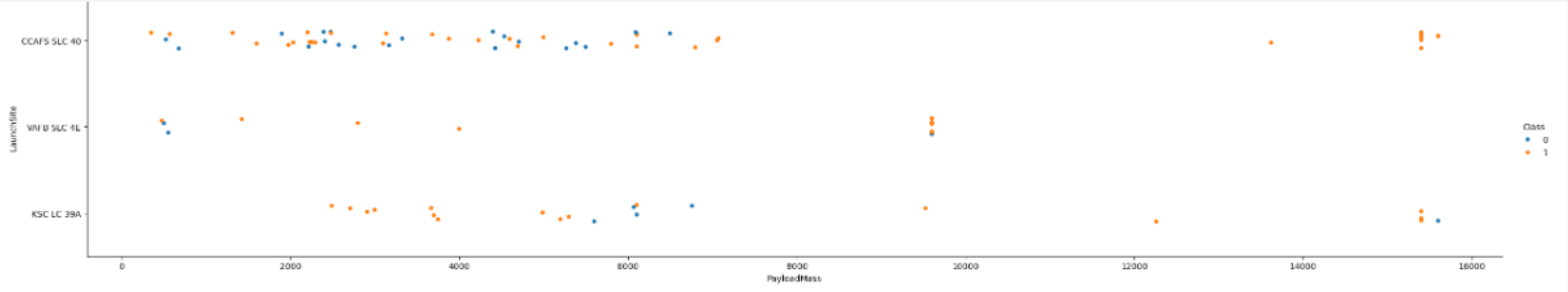
Flight Number vs. Launch Site

- From the plot, A directly proportional relationship was found between the number of flights and the success rate of the launch site.



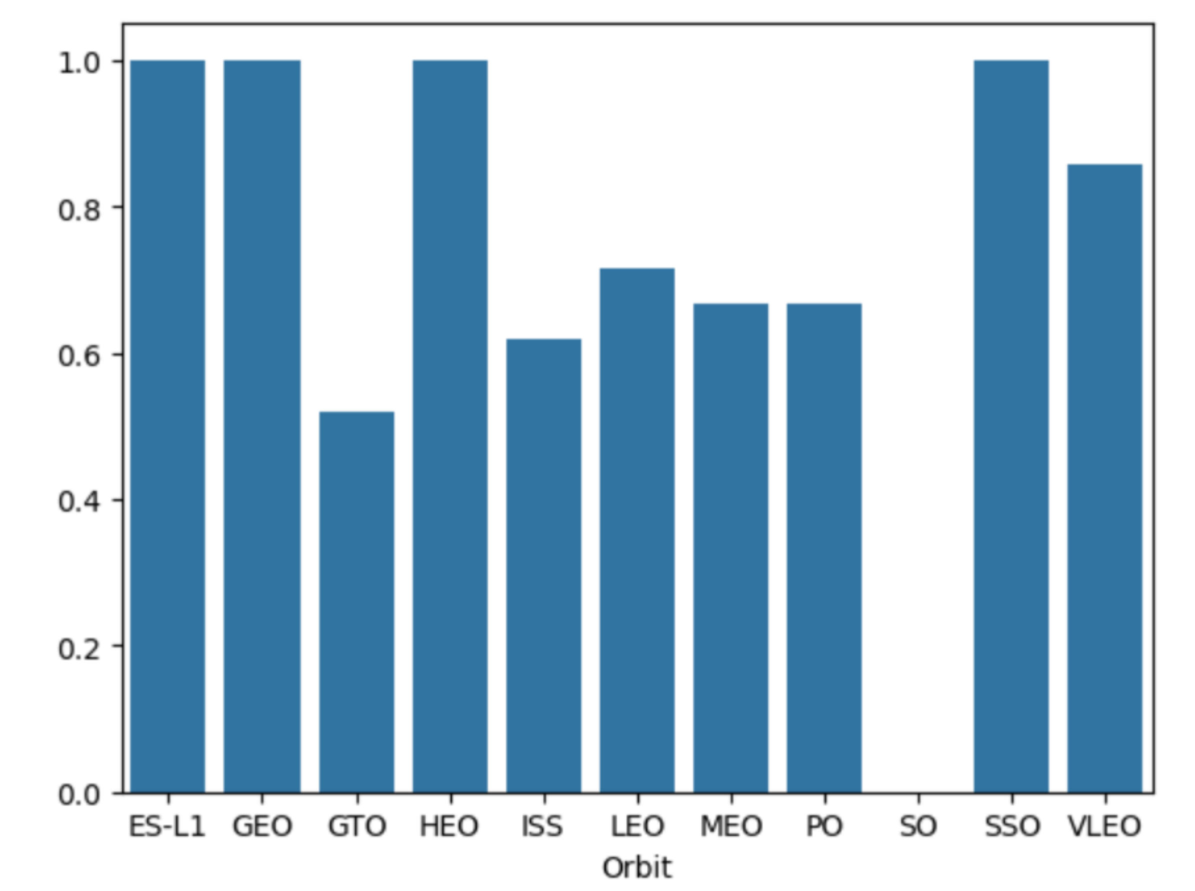
Payload vs. Launch Site

- From the plot, It can be seen that from 8,000 onward, the success rate increases



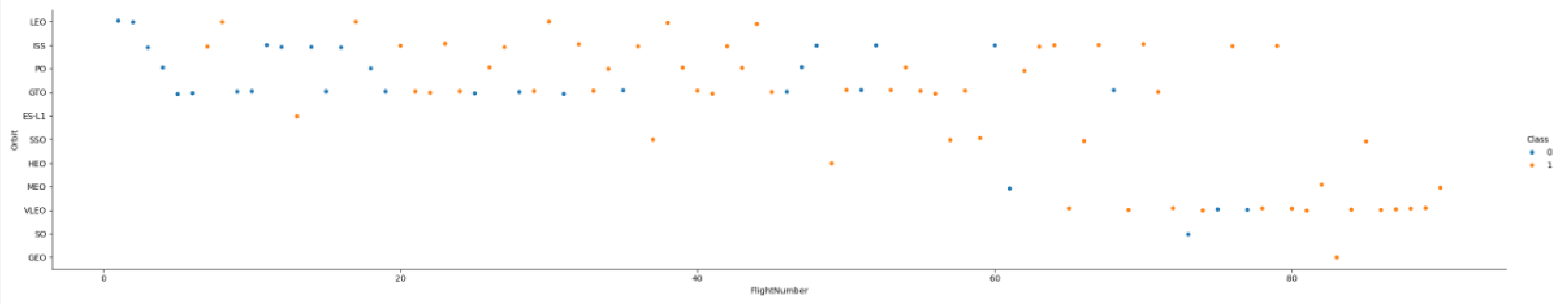
Success Rate vs. Orbit Type

- From the plot, can be seen that ES-L1, GEO, HEO and SSO had the most succes rate



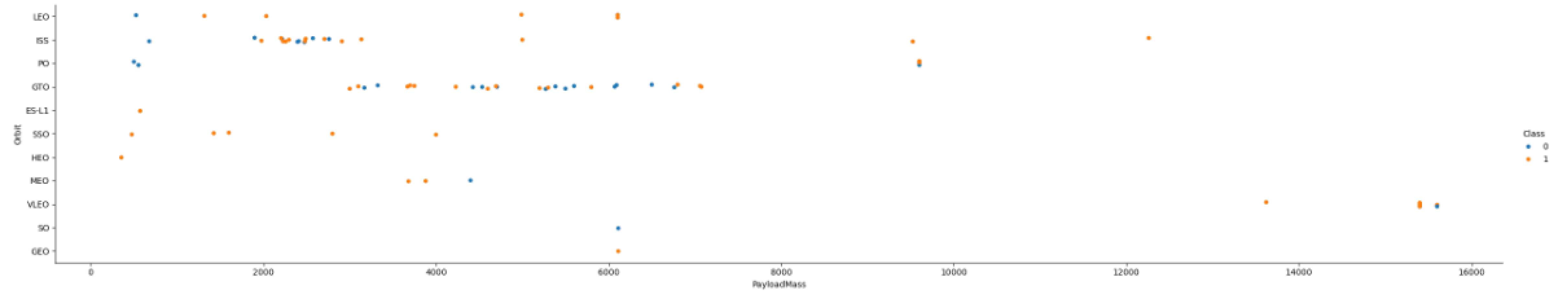
Flight Number vs. Orbit Type

- From the plot, We can observe that the success rate in different orbits increases as the number of flights rises.



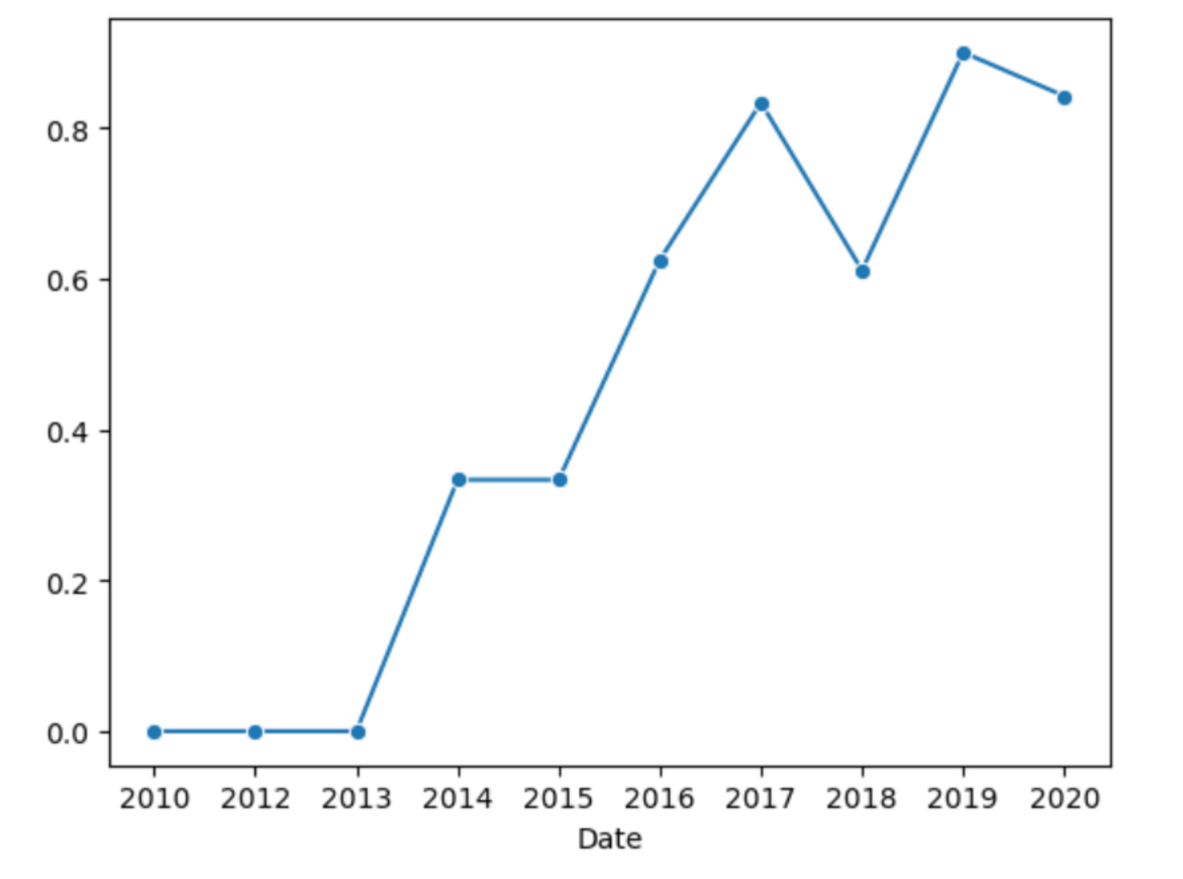
Payload vs. Orbit Type

- From the plot we can observe that with heavy payloads the succesful landing rate are more for PO, LEO and ISS.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing in a general way.



All Launch Site Names

- I used the Magic SQL to perform the DISTINCT key word to show unique launch sites.

Display the names of the unique launch sites in the space mission

In [17]:
```
%sql SELECT DISTINCT launch_site FROM SPACEXTABLE;
```

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

Out[17]:

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

Launch Site Names Begin with 'CCA'

- I Used de Magic SQL to retrieve the firs 5 record where launch sites begin with CCA.

Display 5 records where launch sites begin with the string 'CCA'

In [18]:
```
%sql SELECT * FROM SPACEXTABLE WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

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

Out[18]:

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

- With Magic SQL and the sum function we can retrieve the total payload carried by boosters launched by NASA (CRS)

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

```
In [20]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTABLE WHERE customer = 'NASA (CRS)';

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

Out[20]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- Same process but with avg function to retrieve the average payload mass carried by booster version F9 v1.1

```
In [21]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEXTABLE WHERE booster_version = 'F9 v1.1';

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

Out[21]:

average_payload_mass
2928.4

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
[23]: %sql SELECT MIN(Date) AS First_Success_Date FROM SPACEXTABLE WHERE Mission_Outcome = 'Success' AND Landing_Outcome = 'Success (ground pad)';

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

[23]:

First_Success_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- I used the WHERE clause to filter the booster which have successfully landed on a drone ship with the corresponding payload.

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

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

[24]:

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

Total Number of Successful and Failure Mission Outcomes

- I used the COUNT function to list the total number of successful and failure mission outcomes.

```
[25]: %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Mission_Outcome;

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

[25]:

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

Boosters Carried Maximum Payload

- With a subquery, can retrieve the list of names of the booster which have carried the maximum payload mass.

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

```
[27]: %%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE
);
```

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

[27]:

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

- I used a combination of functions and clause to separate the month and the year, then select the data from the SPACEXTABLE and filtered the landing outcome to know the month that a failure dron ship outcome occurs.

[29]:
```
%%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'
    END AS Month_Name,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Failure (drone ship)'
  AND substr(Date, 1, 4) = '2015';
```

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

[29]:

Month_Name	Booster_Version	Launch_Site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

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

[30]:
```
%%sql
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;
```

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

[30]:

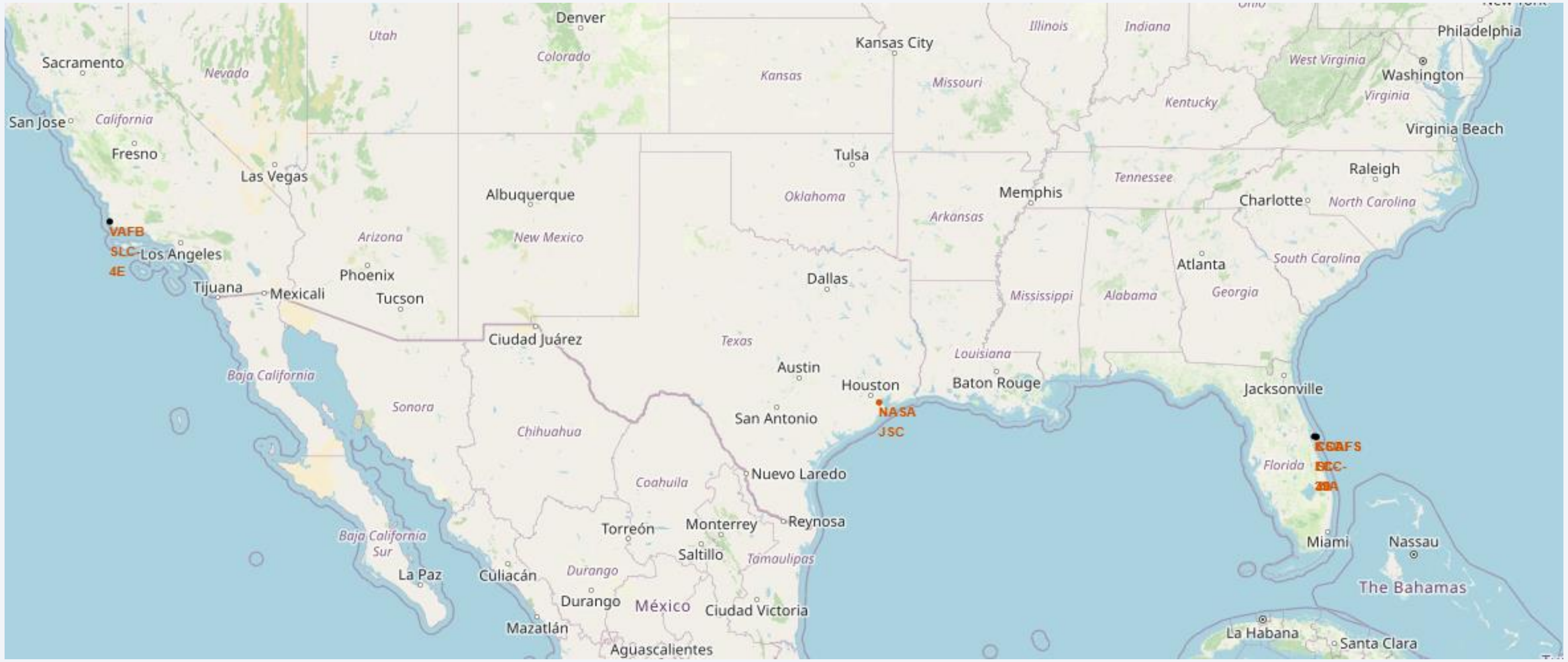
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

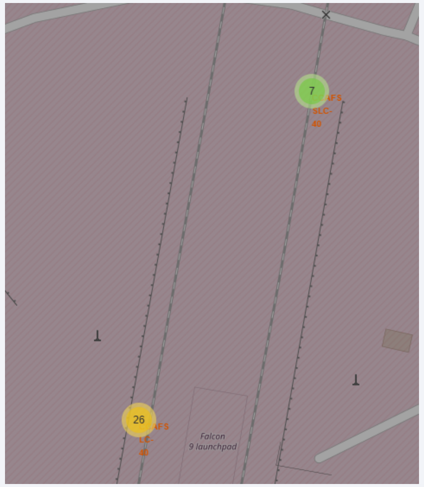
Launch Sites Proximities Analysis

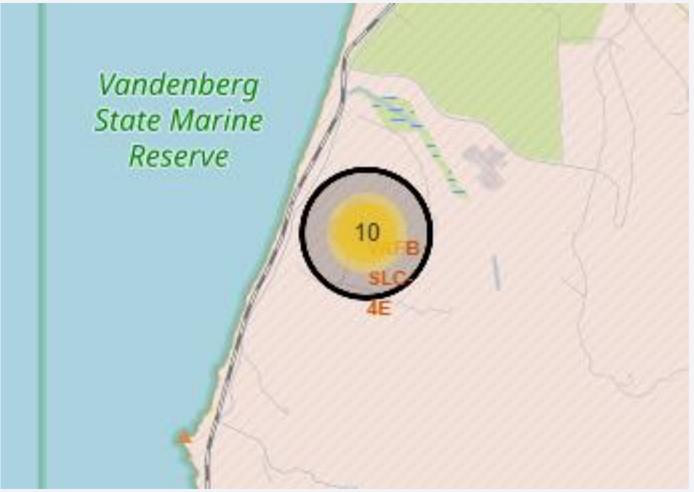
Launch sites global markers

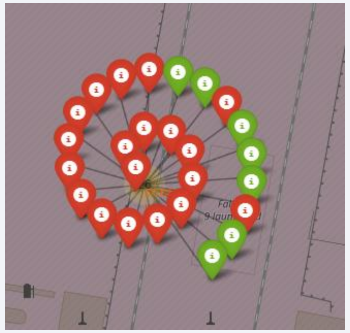
- We can see that the launch sites are in the USA coasts near as possible to the equator line



Markers clusters showing launch sites with color labels

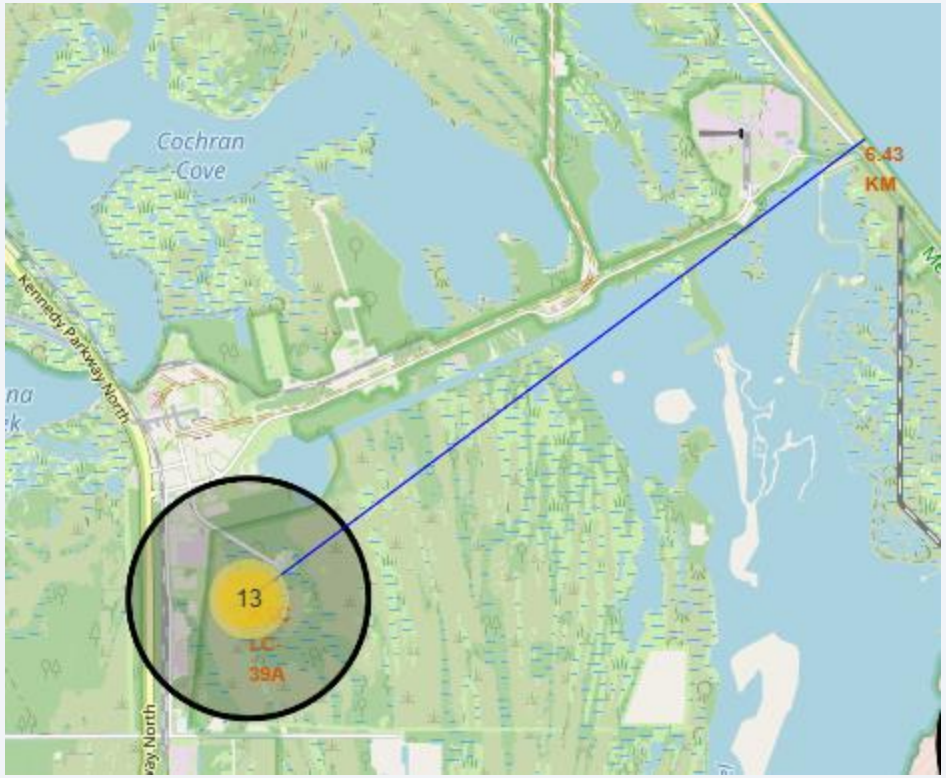




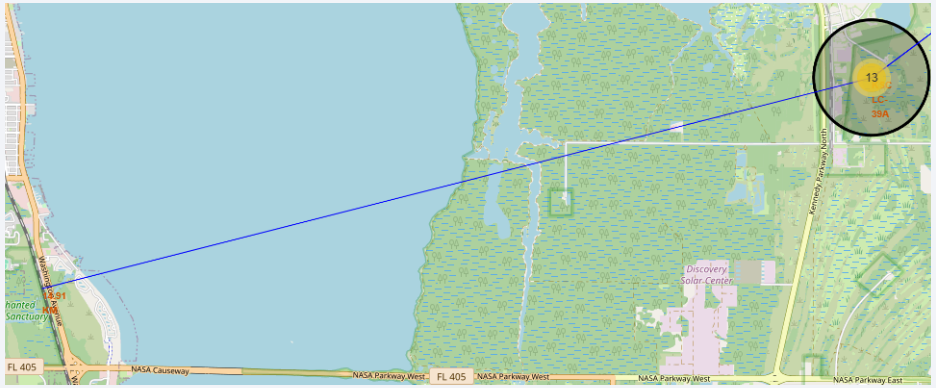


- Green marker shows successful launches and Red marker shows failures.

<Folium Map Screenshot 3>



- Launch sites are close to coastline but are apparently far from highways, railways and cities.



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

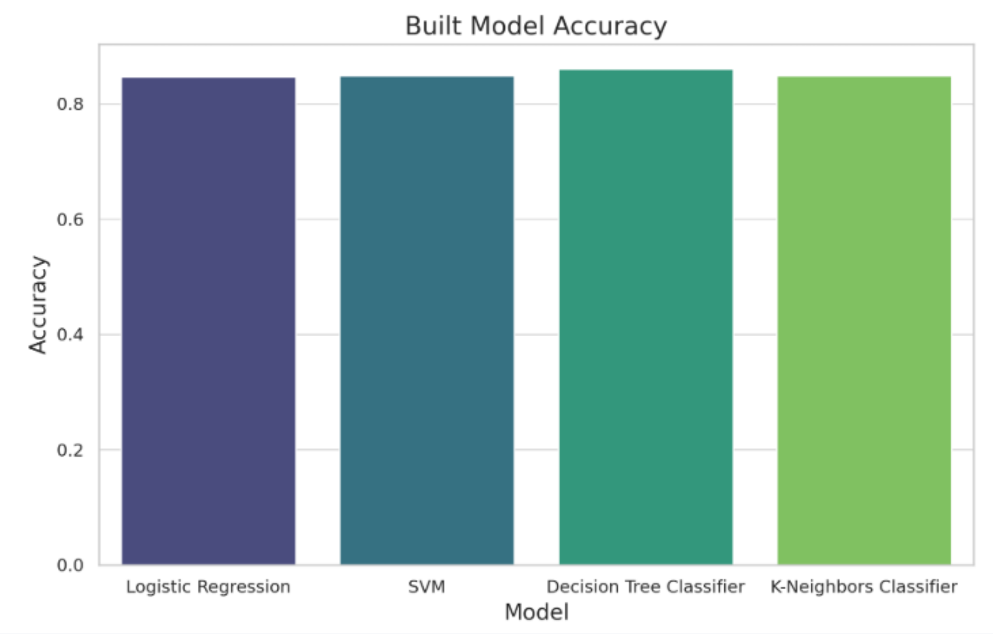
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

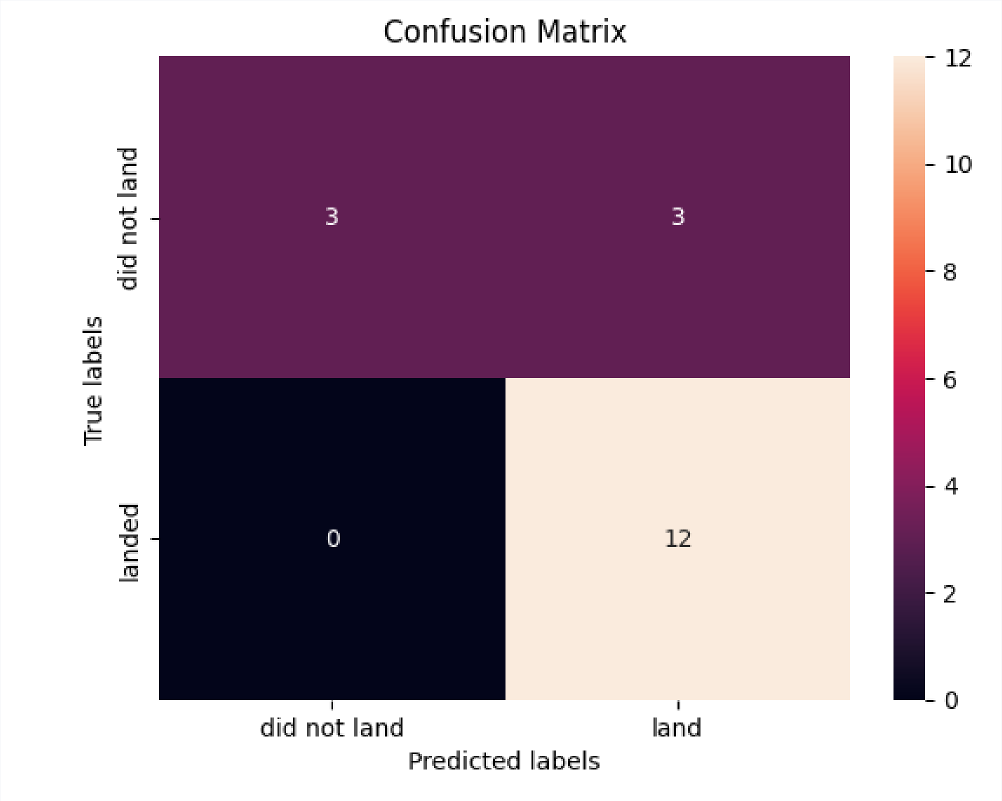
Predictive Analysis (Classification)

Classification Accuracy

- From the plot, the model with the highest classification accuracy, it is the decision tree classifier.



Confusion Matrix



- The confusion matrix for the decision tree classifier shows that the classifier can predict de successful lands very well, but the major problem is the false positives. Unsuccessful landing marked as successful landing by the classifier.

Conclusions

We can conclude that:

- The success rate is directly proportional to the number of flights.
- The success rate has increased over time.
- Orbits like GEO, HEO, and ES-L1 have the highest success rate.
- The launch sites are located far from populated areas but close to a coastline and as close as possible to the equatorial line.
- The decision tree classifier is the best machine learning algorithm for this task.

Thank you!