IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aditha Pathiraja

19.01.2024

Outline

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

Executive Summary

This project focuses on predicting the success of SpaceX Falcon 9 first stage landings using machine learning classification algorithms. Key steps include data collection, wrangling, exploratory data analysis, interactive visualization, and machine learning prediction.

Key Highlights:

1. Data Processing: Efficiently collected, cleaned, and formatted data to create a reliable dataset.
2. Exploratory Data Analysis (EDA): Uncovered crucial patterns and correlations through in-depth analysis.
3. Interactive Visualization: Presented insights using interactive visualizations highlighting correlations.
4. Machine Learning Prediction: Applied various classification algorithms to predict Falcon 9 first stage landing outcomes.

Insights:

- Correlations identified between rocket launch features and success/failure outcomes.
- Decision tree stands out as a promising algorithm for accurate predictions.
- This project contributes valuable insights into the factors influencing Falcon 9 landings, offering a predictive model that enhances decision-making in SpaceX missions.

Introduction

Our capstone's primary focus is predicting the successful landing of the Falcon 9 first stage during SpaceX rocket launches. This prediction holds significance in determining the cost of a launch, crucial for competitive bidding against SpaceX, which advertises a cost of 62 million dollars— a substantial savings compared to other providers. SpaceX's cost efficiency is attributed to the ability to reuse the first stage, emphasizing the need for an accurate prediction of its successful landing.

An intriguing aspect is that most unsuccessful landings are intentional and planned by SpaceX, often involving controlled ocean landings. Distinguishing between planned and unplanned outcomes is pivotal. The core question guiding this project is whether, based on features like payload mass, orbit type, and launch site, we can accurately predict the successful landing of the Falcon 9 first stage. Addressing this question not only aids in cost estimation but also equips stakeholders with valuable insights for strategic decision-making in the competitive space launch industry.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Data was processed using one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Employed various algorithms including logistic regression, support vector machine (SVM), decision tree, and k-nearest neighbors (KNN) for accurate predictions.

Data Collection

Data collection involves the systematic gathering and measurement of information related to specific variables within an established system. This process enables the formulation of pertinent questions and the evaluation of outcomes. In this project, the dataset was acquired through both REST API and web scraping from Wikipedia.

For REST API, the process initiated with a GET request. Subsequently, the response content was decoded as JSON and transformed into a pandas dataframe using json_normalize(). Following this, data cleaning procedures were implemented, including the identification and handling of missing values.

In the case of web scraping, BeautifulSoup was employed to extract launch records from an HTML table. The table was then parsed and converted into a pandas dataframe for subsequent analysis. This dual approach to data collection ensures a comprehensive dataset for the project's analytical phases.

Data Collection – SpaceX API

Get request for rocket launch data using API

↓

Use json_normalize method to convert json result to dataframe

↓

Performed data cleaning and filling the missing value

https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/jupyter-labs-spacex-data-collection-api.ipynb

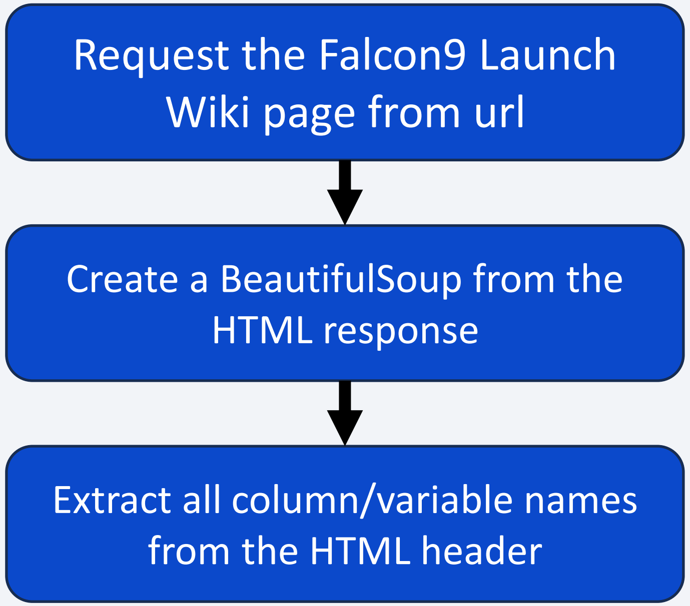
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

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

```
# Lets take a subset of our dataframe keeping only the features we want and the flight nu
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra r
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in t
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the dat
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping



https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/jupyter-labs-webscraping.ipynb

```python
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

```
200
```

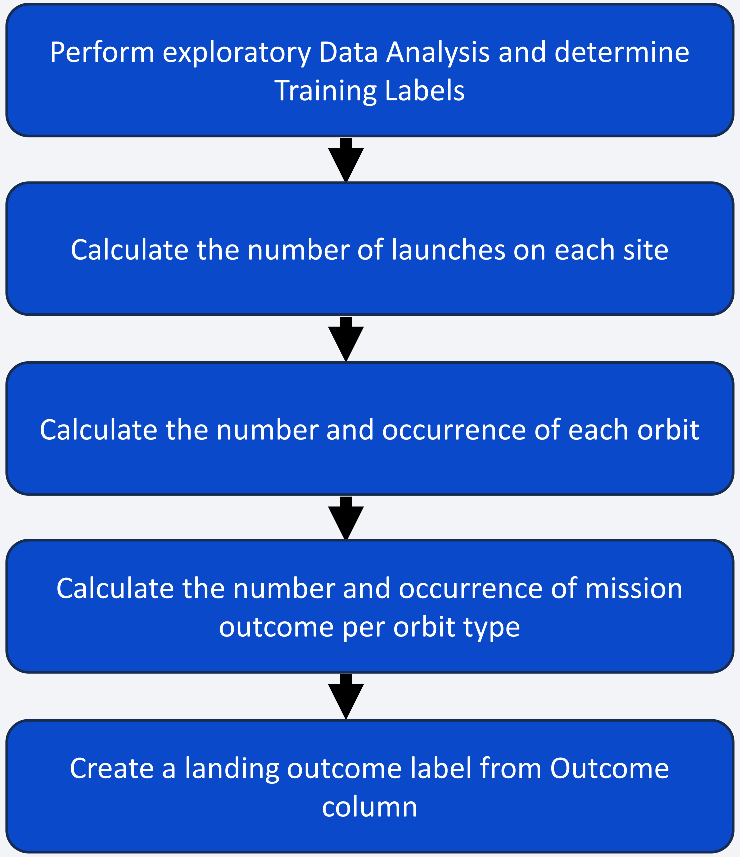
Create a `BeautifulSoup` object from the HTML `response`

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text)
```

```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)

            datatimelist=date_time(row[0])
            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
```

Data Wrangling

Data wrangling involves the systematic cleaning and organizing of intricate and disorderly datasets, making them more accessible for both analysis and Exploratory Data Analysis (EDA). Our initial step involves determining the count of launches at each site, followed by the computation of the number and frequency of mission outcomes per orbit type. Subsequently, we generate a landing outcome label derived from the outcome column, streamlining it for subsequent analysis, visualization, and machine learning applications. The final step involves exporting the results to a CSV file for further use.



https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Initially, we employed scatter plots to explore relationships between various attributes, including:

- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

Scatter plots serve as visual tools to depict dependencies between attributes. By analyzing these plots, patterns and correlations among factors influencing the success of landing outcomes become evident. This graphical exploration facilitates a clearer understanding of the key factors impacting the overall success of the landing outcomes.

https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Using SQL, we had performed many queries to get better understanding of the dataset,

- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string ‘CCA’.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015.
- Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order

https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

To create an interactive map visualizing launch data, we extracted latitude and longitude coordinates for each launch site. Utilizing these coordinates, we incorporated circle markers around each launch site, accompanied by a label denoting the site's name. The launch outcomes, categorized into success and failure (assigned as classes 0 and 1), were represented by red and green markers on the map using MarkerCluster().

To assess proximity, we employed Haversine's formula to calculate distances between launch sites and various landmarks, addressing questions such as:

- The proximity of launch sites to railways, highways, and coastlines.
- The closeness of launch sites to nearby cities.

This approach not only enhances the visualization of launch data but also provides valuable insights into the geographical relationships between launch sites and surrounding landmarks.

https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

We developed an interactive dashboard using Plotly Dash, providing users with the flexibility to explore the data dynamically.

The dashboard features pie charts displaying the total launches from specific sites.

Additionally, we included scatter graphs illustrating the relationship between the outcome and payload mass (in kilograms) across different booster versions.

This interactive platform empowers users to interact with and analyze the data based on their specific needs and preferences.

https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/spacex_dash_app.py

Predictive Analysis (Classification)

- Creating a NumPy array from the column “Class” in data
- Standardizing the data with StandardScaler, then fitting and transforming it
- Splitting the data into training and testing sets with train_test_split function
- Creating a GridSearchCV object with cv = 10 to find the best parameters
- Applying GridSearchCV on LogReg, SVM, Decision Tree, and KNN models
- Calculating the accuracy on the test data using the method .score() for all models
- Examining the confusion matrix for all models
- Finding the method performs best by examining the Jaccard_score and F1_score metrics

https://github.com/adithapathiraja/Capstone-Project/blob/0cd7f0e6c71e9b9a72cc0d7ab55a4a2940293331/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

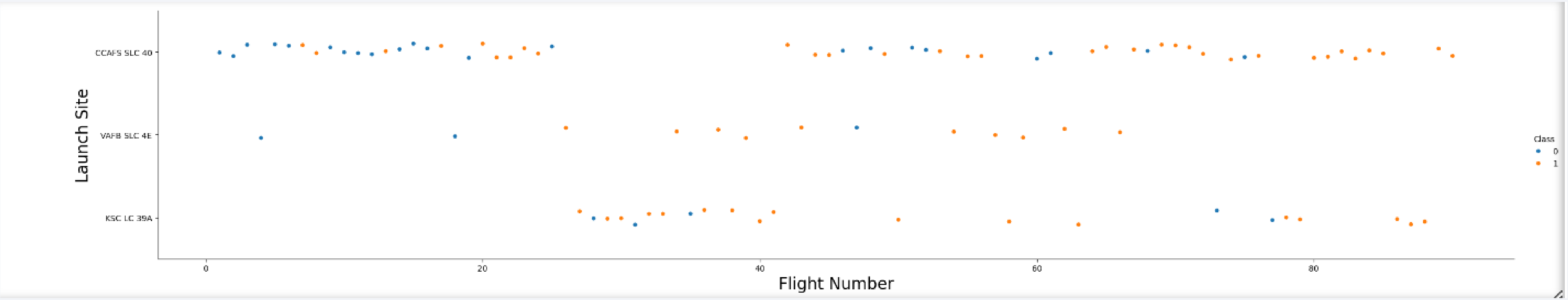
The results will be categorized to 3 main results which is:

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

Section 2

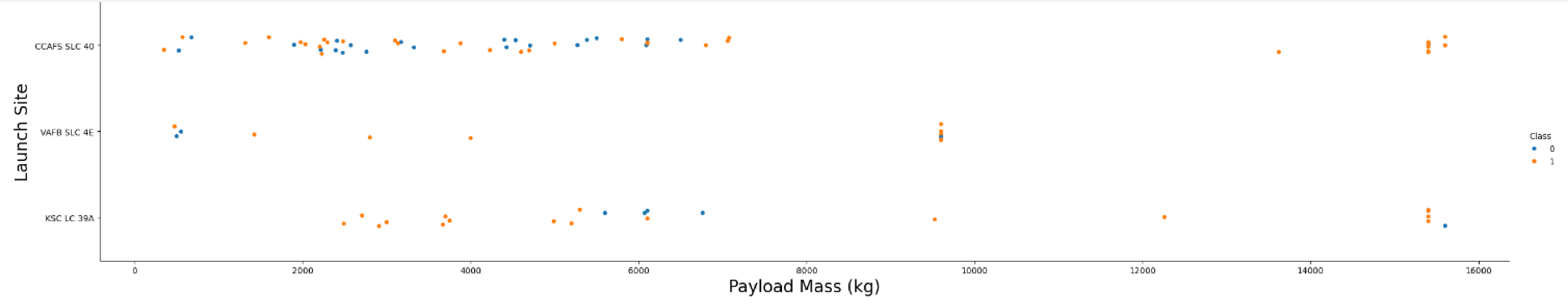
Insights drawn from EDA

Flight Number vs. Launch Site



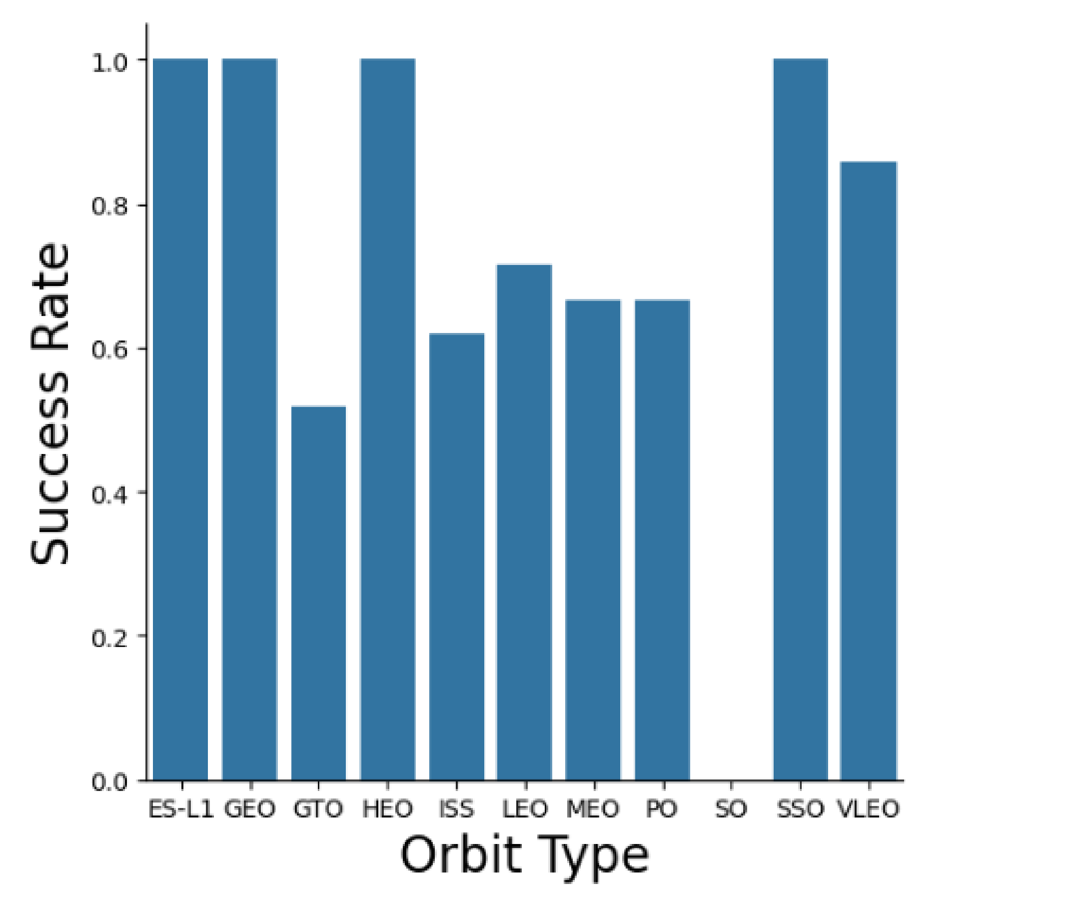
This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be. However, site CCAFS SLC40 shows the least pattern of this

Payload vs. Launch Site



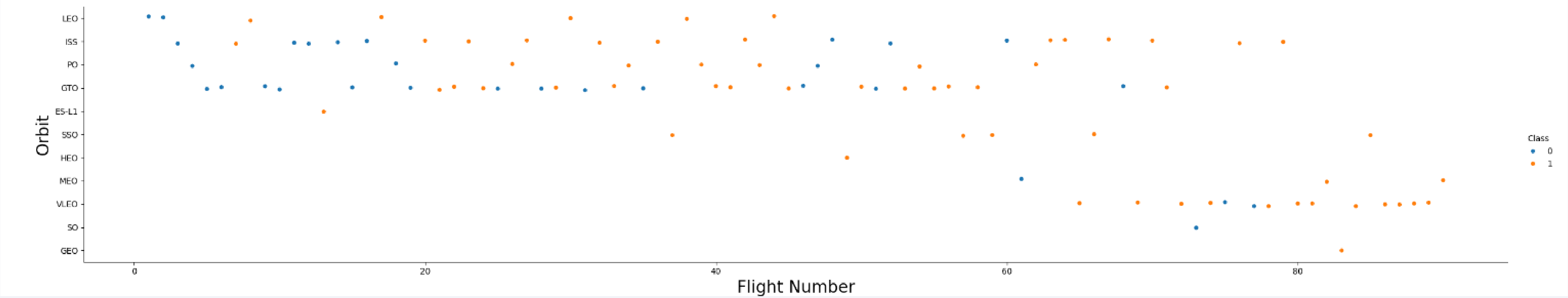
This scatter plot shows once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased. However, there is no clear pattern to say the launch site is dependent to the pay load mass for the success rate

Success Rate vs. Orbit Type



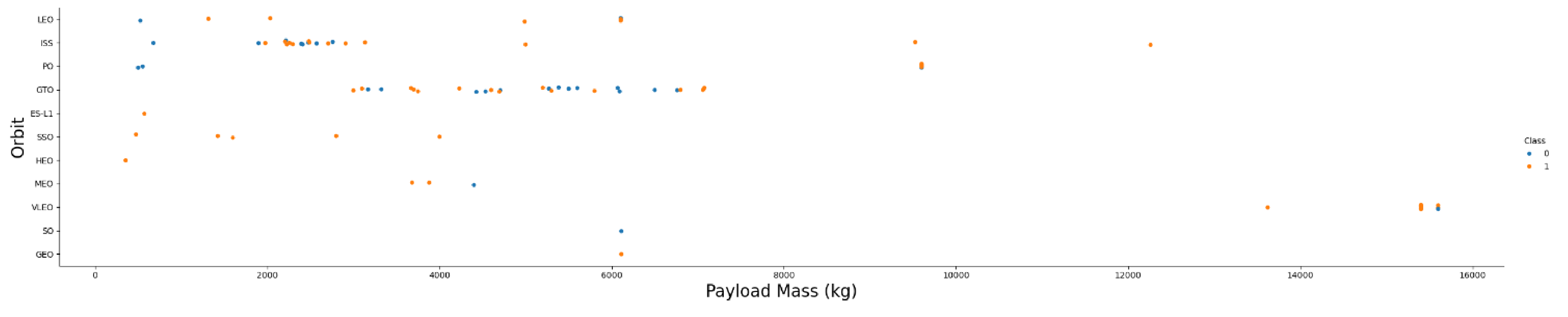
This figure depicted the possibility of the orbits to influences the landing outcomes as some orbits has 100% success rate such as SSO, HEO, GEO AND ES-L1 while SO orbit produced 0% rate of success. However, deeper analysis show that some of this orbits has only 1 occurrence such as GEO, SO, HEO and ES-L1 which mean this data need more dataset to see pattern or trend before we draw any conclusion.

Flight Number vs. Orbit Type



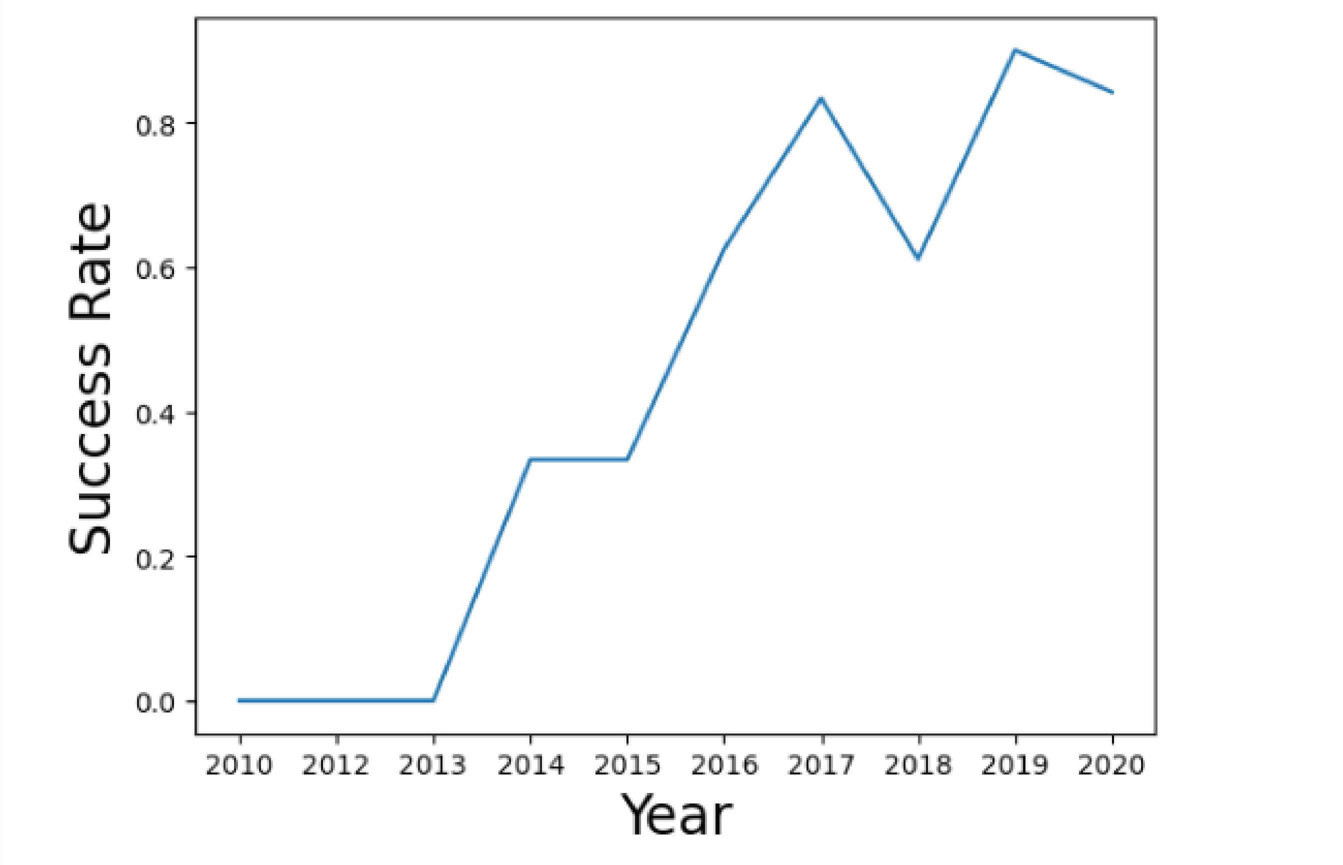
This scatter plot shows that generally, the larger the flight number on each orbits, the greater the success rate (especially LEO orbit) except for GTO orbit which depicts no relationship between both attributes. Orbit that only has 1 occurrence should also be excluded from above statement as it's needed more dataset.

Payload vs. Orbit Type



Heavier payload has positive impact on LEO, ISS and P0 orbit. However, it has negative impact on MEO and VLEO orbit. GTO orbit seem to depict no relation between the attributes. Meanwhile, again, SO, GEO and HEO orbit need more dataset to see any pattern or trend.

Launch Success Yearly Trend



This figures clearly depicted and increasing trend from the year 2013 until 2020. If this trend continue for the next year onward. The success rate will steadily increase until reaching 1/100% success rate.

All Launch Site Names

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
[18]: %sql select distinct launch_site from SPACEXTABLE;

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

[18]:

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

Launch Site Names Begin with 'CCA'

We used the query above to display 5 records where launch sites begin with `CCA`

```
[19]: %sql select * from SPACEXTABLE where launch_site like 'CCA%' limit 5;

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

[19]:

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

We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTABLE where customer = 'NASA (CRS)';
```

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

total_payload_mass
45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2534.66

```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTABLE where booster_version like '%F9 v1.1%';
```

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

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

We use the min() function to find the result We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min(date) as first_successful_landing from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)';
```

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

first_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%sql select booster_version from SPACEXTABLE where landing_outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

We used cound and groupby methods to get the total number of Success and Failures

```
%sql select mission_outcome, count(*) as total_number from SPACEXTABLE group by mission_outcome;
```

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

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

Boosters Carried Maximum Payload

```
%sql select booster_version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);
```

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

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

We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

2015 Launch Records

```
%%sql
SELECT
  CASE substr(date, 6, 2)
    WHEN '01' THEN 'January'
    WHEN '02' THEN 'February'
    WHEN '03' THEN 'March'
    WHEN '04' THEN 'April'
    WHEN '05' THEN 'May'
    WHEN '06' THEN 'June'
    WHEN '07' THEN 'July'
    WHEN '08' THEN 'August'
    WHEN '09' THEN 'September'
    WHEN '10' THEN 'October'
    WHEN '11' THEN 'November'
    WHEN '12' THEN 'December'
  END AS month,
  date,
  booster_version,
  launch_site,
  landing_outcome
FROM
  SPACEXTABLE
WHERE
  landing_outcome = 'Failure (drone ship)' AND substr(date, 1, 4) = '2015';
```

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

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

We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20. We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

```
%%sql select landing_outcome, count(*) as count_outcomes from SPACEXTABLE
      where date between '2010-06-04' and '2017-03-20'
      group by landing_outcome
      order by count_outcomes desc;
```

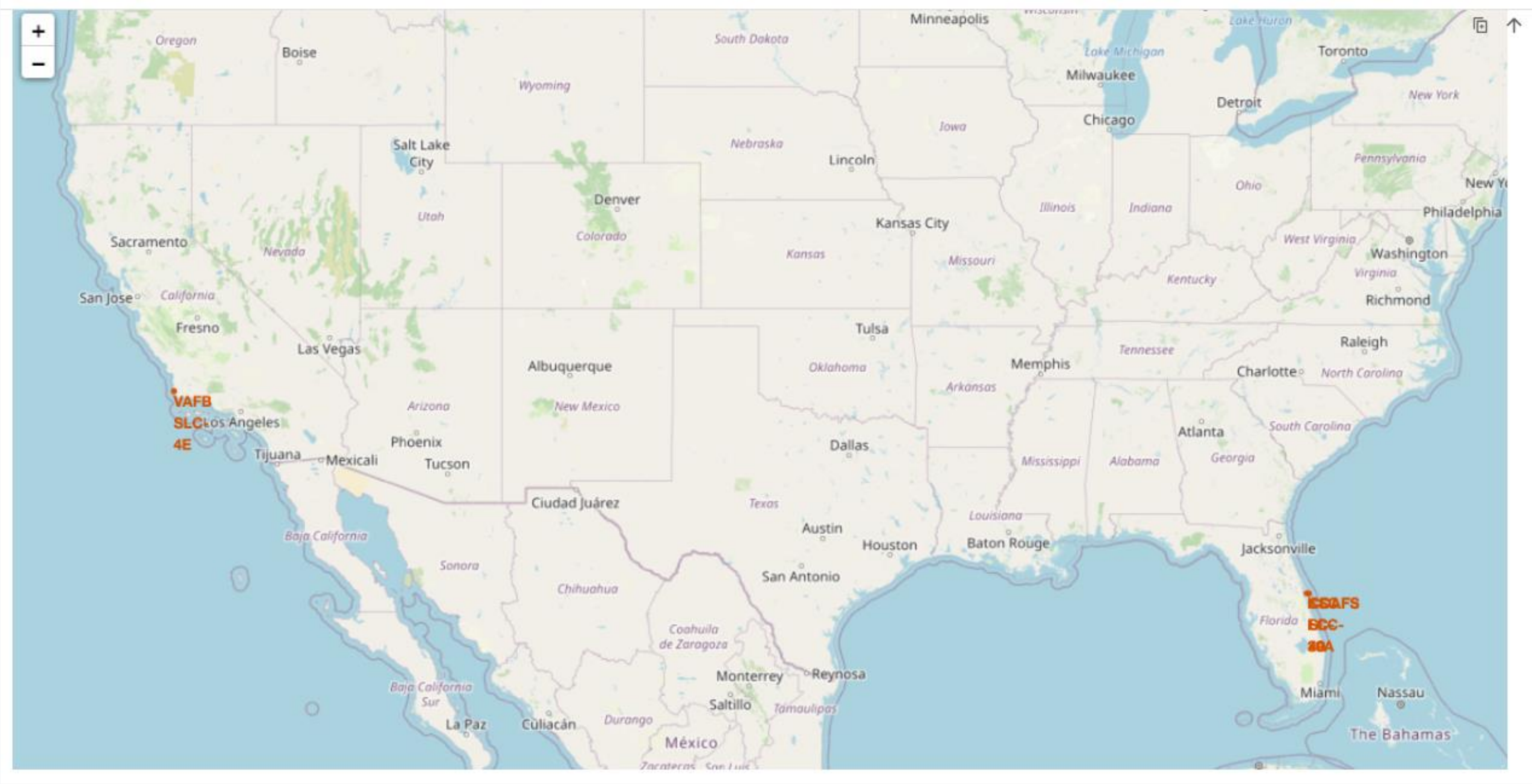
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count_outcomes
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

Location of all the Launch Sites



We can see that all the SpaceX launch sites are located inside the United States in California and LA areas.

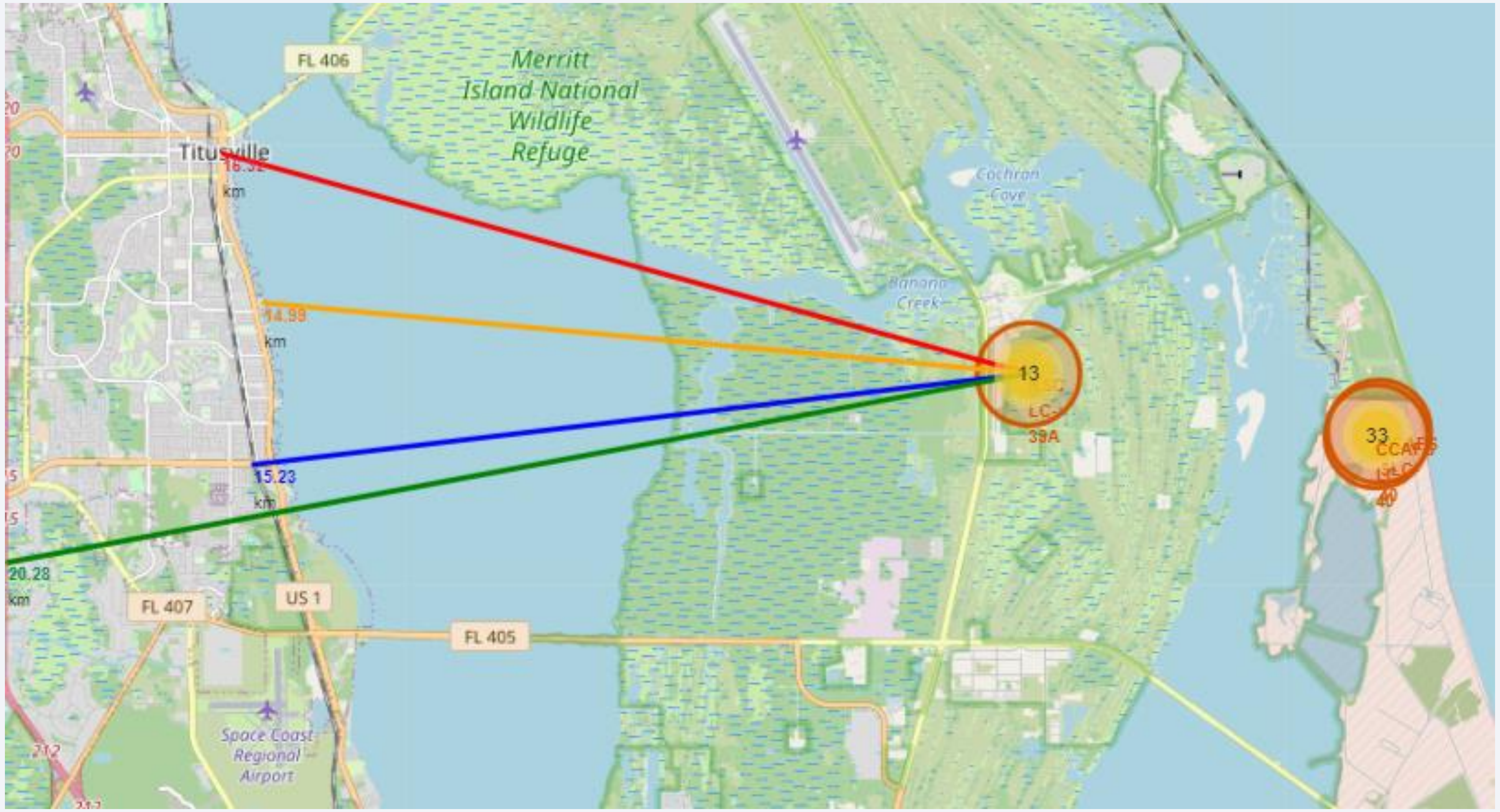
Markers showing launch sites with color labels

From the color-labeled markers we should be able to easily identify which launch sites have relatively high success rates.

- Green Marker = Successful Launch
- Red Marker = Failed Launch

Launch Site KSC LC-39A has a very high Success Rate.

Distance from the launch site KSC LC-39A to its proximities



From the visual analysis of the launch site KSC LC-39A we can clearly see that it is:

- relative close to railway (15.23 km)
- relative close to highway (20.28 km)
- relative close to coastline (14.99 km)

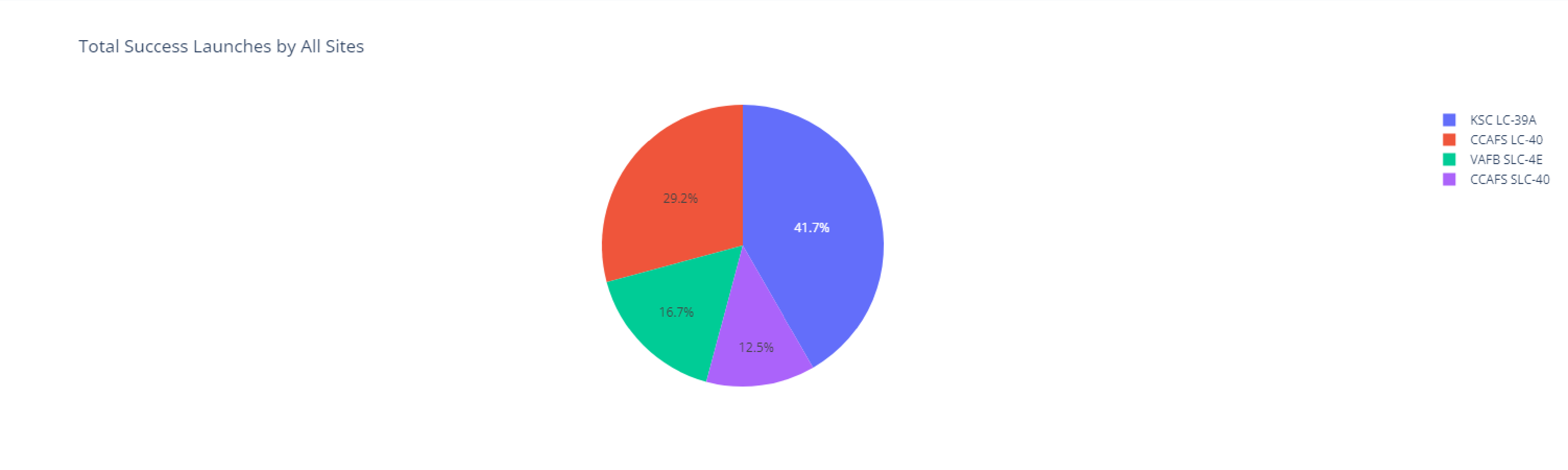
Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km). Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas.

Section 4

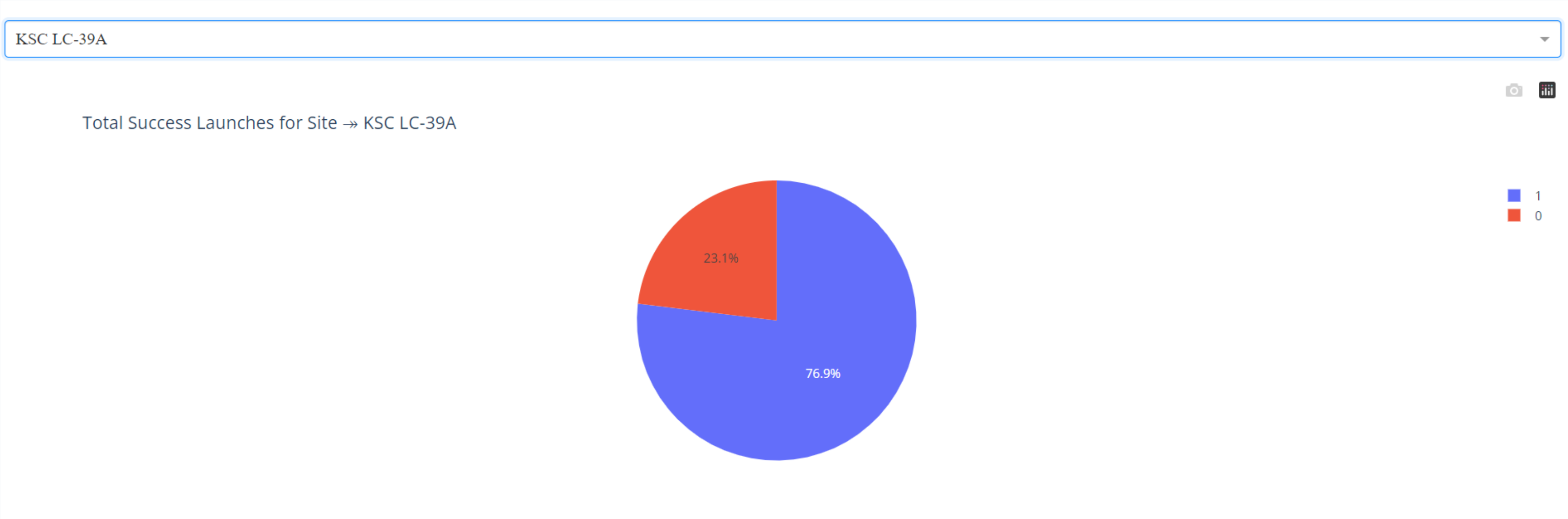
Build a Dashboard with Plotly Dash

The success percentage by each site

The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.



The highest launch-success ratio: KSC LC-39A



KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

Payload Mass vs. Launch Outcome for all sites

We can see that all the success rate for low weighted payload is higher than heavy weighted payload

Section 5

Predictive Analysis (Classification)

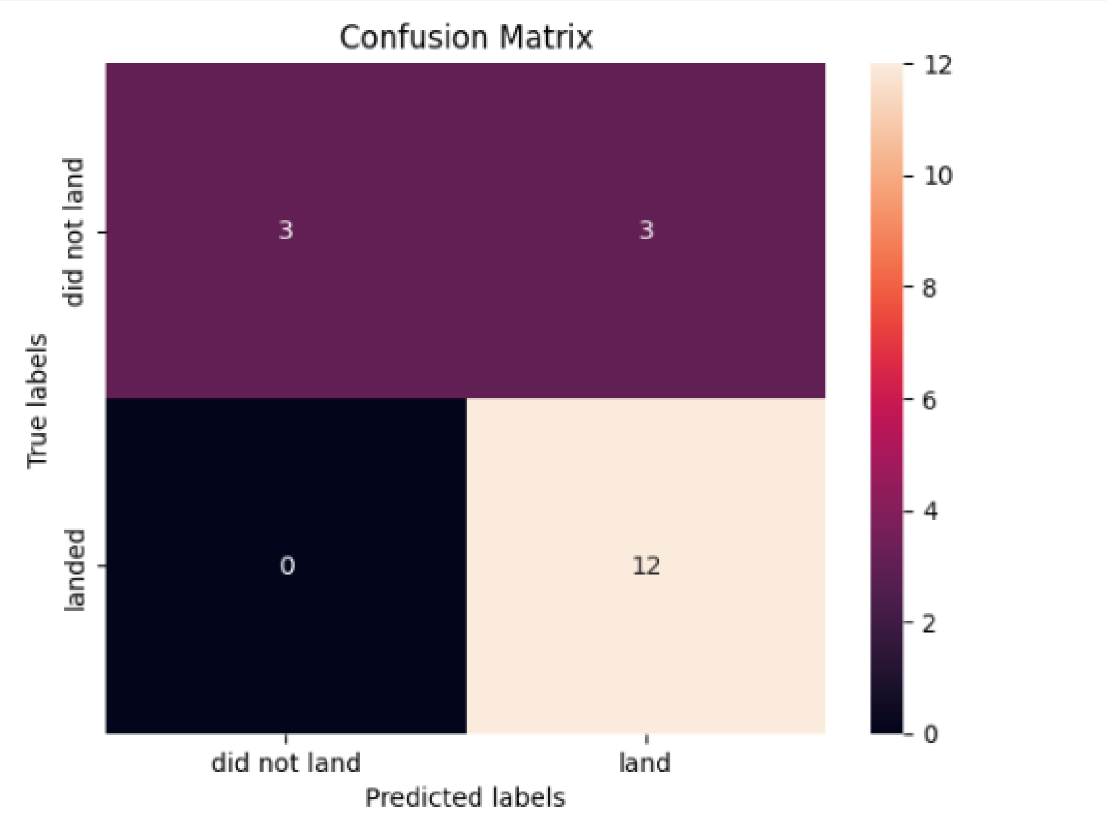
Classification Accuracy

by using the code as below: we could identify that the best algorithm to be the Tree Algorithm which have the highest classification accuracy. Parameters are shown in the output.

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8857142857142856
Best Params is : {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
```

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

Conclusions

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence

Thank you!