Winning Space Race with Data Science

ABDUL WADOOD ASIM

16 JULY 2023

Outline

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

Executive Summary

- Summary of methodologies
  - ❑ Data Collection through API
  - ❑ Data Collection with Web Scraping
  - ❑ Data Wrangling
  - ❑ Exploratory Data Analysis with SQL
  - ❑ Exploratory Data Analysis with Data Visualization
  - ❑ Interactive Visual Analytics with Folium
  - ❑ Machine Learning Prediction
- Summary of all results
  - ❑ Exploratory Data Analysis result
  - ❑ Interactive analytics in screenshots
  - ❑ Predictive Analytics result from Machine Learning Lab

Introduction

SpaceX, a revolutionary company, has disrupted the space industry by offering rocket launches at an incredibly low cost of just 62 million dollars for their Falcon 9, while other providers charge upwards of 165 million dollars for each launch. This substantial cost reduction is primarily attributed to SpaceX's remarkable concept of reusing the first stage of the rocket by successfully landing it for future missions. By repeating this process, the price is further reduced. As a data scientist working for a startup that competes with SpaceX, the objective of this project is to develop a machine learning pipeline capable of predicting the landing outcome of the first stage in future missions. This project holds significant importance in determining the optimal bidding price against SpaceX for a rocket launch. The challenges involved in this project include identifying all the factors that influence the landing outcome and understanding the relationship between each variable and its impact on the outcome.

- The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- 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
  - How to build, tune, evaluate classification models

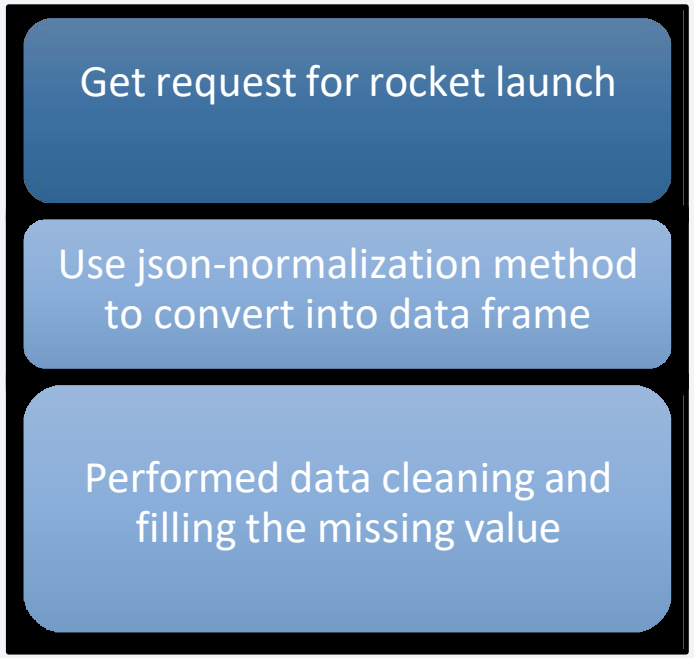
Data Collection

Data collection involves the process of gathering and measuring information regarding specific variables within an established system. This enables individuals to address pertinent questions and evaluate outcomes effectively. In this particular case, the dataset was obtained through two methods: REST API and web scraping from Wikipedia.

To retrieve data using the REST API, we initiated a "get" request. Subsequently, we decoded the response content in JSON format and transformed it into a pandas dataframe using the "json_normalize()" function. Following this, we performed data cleaning, checked for missing values, and filled them as necessary.

Regarding web scraping, we utilized BeautifulSoup to extract the launch records from the HTML table. The table was then parsed and converted into a pandas dataframe, facilitating further analysis.

Data Collection - SpaceX API



From: https://github.com/asim077/my-final-assignments/blob/main/notebook_Data_Collection_yJPxhv2oU.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 a
nd the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number',
'date_utc']]

# We will remove rows with multiple cores because those are falcon rocket
s with 2 extra rocket boosters and rows that have multiple payloads in a
single rocket.
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 s
ingle value in the list and replace the feature.
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 ex
tracting the date leaving the time
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

Request the Falcon9 Launch Wiki page from url

Create a BeautifulSoup from the HTML response

Extract all column/variable names from the HTML header

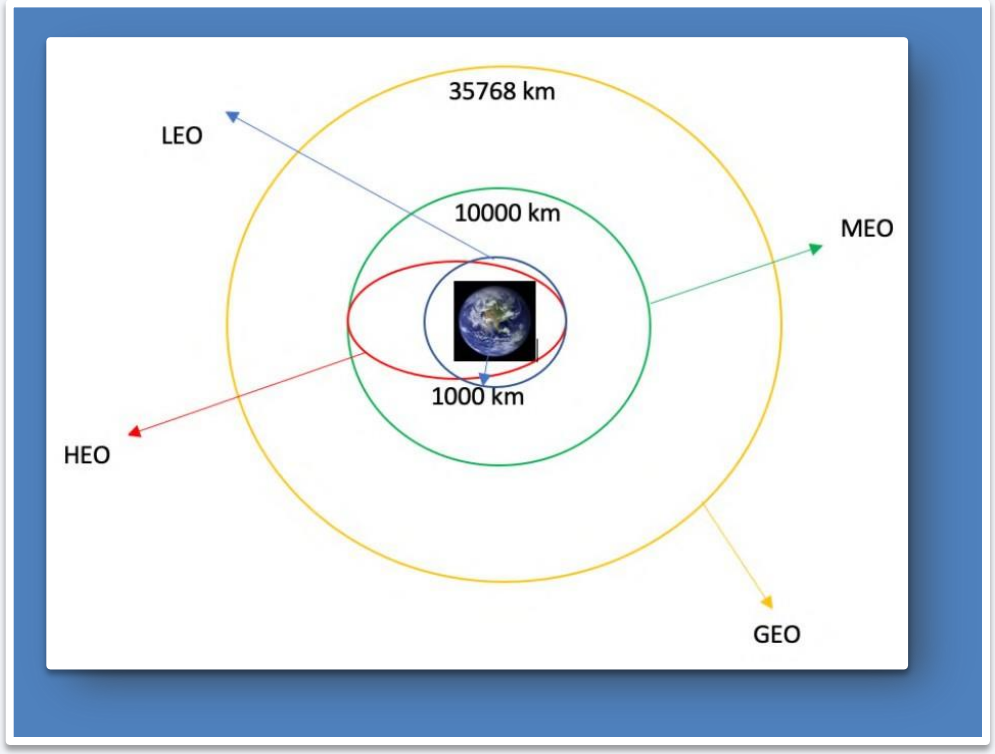
From: https://github.com/asim077/my-final-assignments/blob/main/notebook_Data_Collection_with_Web_Scraping_nI89VIRCE.ipynb

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response te
xt content
soup = BeautifulSoup(data,'html.parser')
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plai
nrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding t
o launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

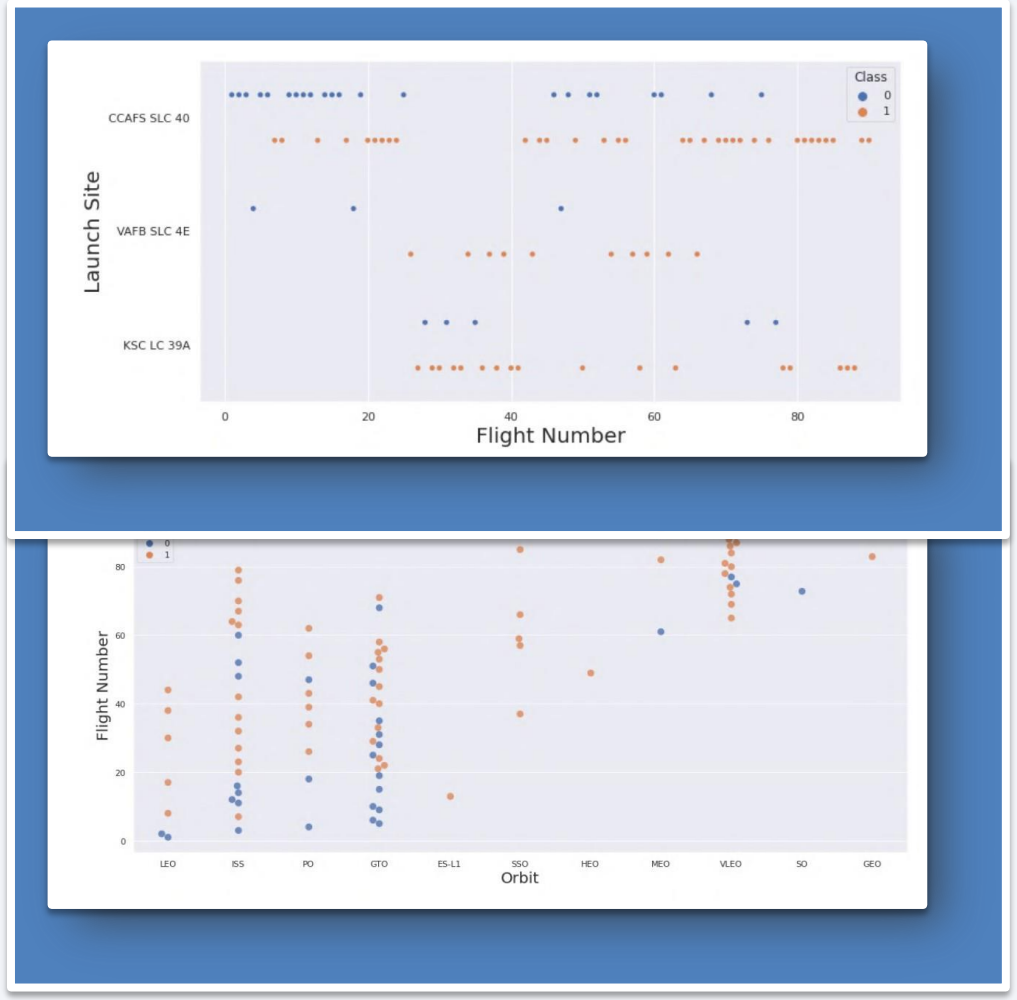
Data Wrangling



From: https://github.com/asim077/my-final-assignments/blob/main/notebook_Data_Wrangling_9HnvfsJ5G.ipynb

Data wrangling refers to the process of cleaning and organizing intricate and disorderly datasets, making them more accessible and suitable for exploratory data analysis (EDA).
In this project, our initial step will involve determining the number of launches that took place at each site. Subsequently, we will calculate the frequency and occurrence of mission outcomes based on different orbit types.
create a landing outcome label from the outcome column. for further analysis, visualization, and ML. Lastly, we will export the result to a CSV.

EDA with Data Visualization



To begin our analysis, we employed scatter graphs to explore the relationships between various attributes. Specifically, we examined the relationships between the following pairs:

Payload and Flight Number.

Flight Number and Launch Site.

Payload and Launch Site.

Flight Number and Orbit Type.

Payload and Orbit Type.

Scatter plots are effective tools for visualizing the dependencies between different attributes. By studying the patterns revealed in these graphs, we can easily identify which factors have the greatest impact on the success of landing outcomes.https://github.com/asim077/my-final-assignments/blob/main/notebook_Exploratory_Data_Analysis_with_SQL__eqznon1EA.ipynb

EDA with Data Visualization

After gaining insights from scatter plots, we will utilize additional visualization tools for further analysis. Two such tools are bar graphs and line plots.

Bar graphs are an intuitive way to interpret the relationships between attributes. In our case, we will employ bar graphs to determine which orbits exhibit the highest probability of success.

Next, we will utilize line graphs to showcase trends or patterns in attributes over time. Specifically, we will examine the yearly trend of launch success, enabling us to observe any notable patterns or changes.

To prepare for success prediction in future modules, we will employ feature engineering techniques. This involves creating dummy variables for categorical columns, allowing us to incorporate them into our predictive models effectively.

https://github.com/asim077/my-final-assignments/blob/main/notebook_Exploratory_Data_Analysis_with_SQL__eqznon1EA.ipynb

EDA with SQL

sing SQL, we executed several queries to gain a better understanding of the dataset. Here are some examples:

Displayed the names of the launch sites.

Displayed 5 records where launch sites begin with the string 'CCA'.

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

Displayed the average payload mass carried by booster version F9 v1.1.

Listed the date when the first successful landing outcome on a ground pad was achieved.

Listed the names of boosters that successfully landed on a drone ship with a payload mass greater than 4000 but less than 6000.

Listed the total number of successful and failed mission outcomes.

Listed the names of booster versions that carried the maximum payload mass.

Listed the failed landing outcomes on a drone ship, along with their corresponding booster versions and launch site names, for the year 2015.

Ranked the count of landing outcomes or successes between the dates 2010-06-04 and 2017-03-20 in descending order.

https://github.com/asim077/my-final-assignments/blob/main/notebook_Exploratory_Data_Analysis_with_SQL__eqznon1EA.ipynb

Build an Interactive Map with Folium

To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.

We then assigned the dataframe launch_outcomes(failure,success) to classes 0 and 1 with Red and Green markers on the map in MarkerCluster().

We then used the Haversine's formula to calculated the distance of the launch sites to various landmark to find answer to the questions of:

- How close the launch sites with railways, highways and coastlines?
- How close the launch sites with nearby cities?

From: https://github.com/asim077/my-final-assignments/blob/main/notebook_Interactive_Visual_Analytics_with_Folium_M8uUhCmHY.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

The link of the app.py:: https://github.com/asim077/my-final-assignments/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Building the Model

- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of ML to use
- set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- plot the confusion matrix.

Improving the Model

- Use Feature Engineering and Algorithm Tuning

Find the Best Model

- The model with the best accuracy score will be the best performing model.

From: https://github.com/asim077/my-final-assignments/blob/main/spacex_dash_app.py

Results

The results obtained from the analysis will be categorized into three main categories:

Exploratory Data Analysis (EDA) Results: These findings will include insights and discoveries derived from the initial exploration of the data. They will help us understand the relationships, patterns, and trends within the dataset.

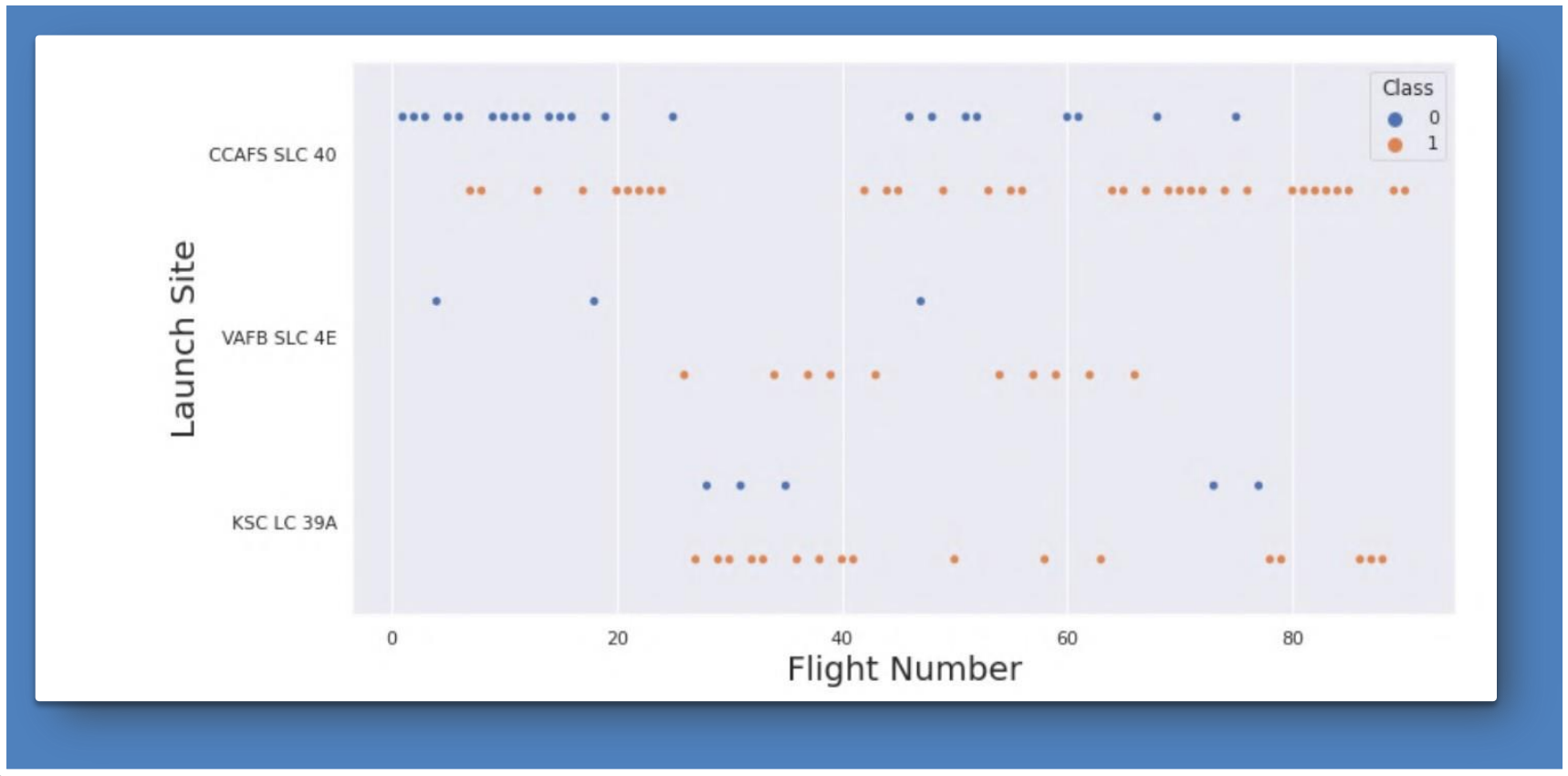
Interactive Analytics Demo in Screenshots: This category will consist of visual demonstrations showcasing interactive analytics using screenshots. It will provide an interactive experience to navigate and explore the data, allowing users to gain a deeper understanding of the dataset.

Predictive Analysis Results: This section will present the outcomes of predictive analysis, which involves utilizing machine learning techniques to make predictions or forecasts based on the dataset.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

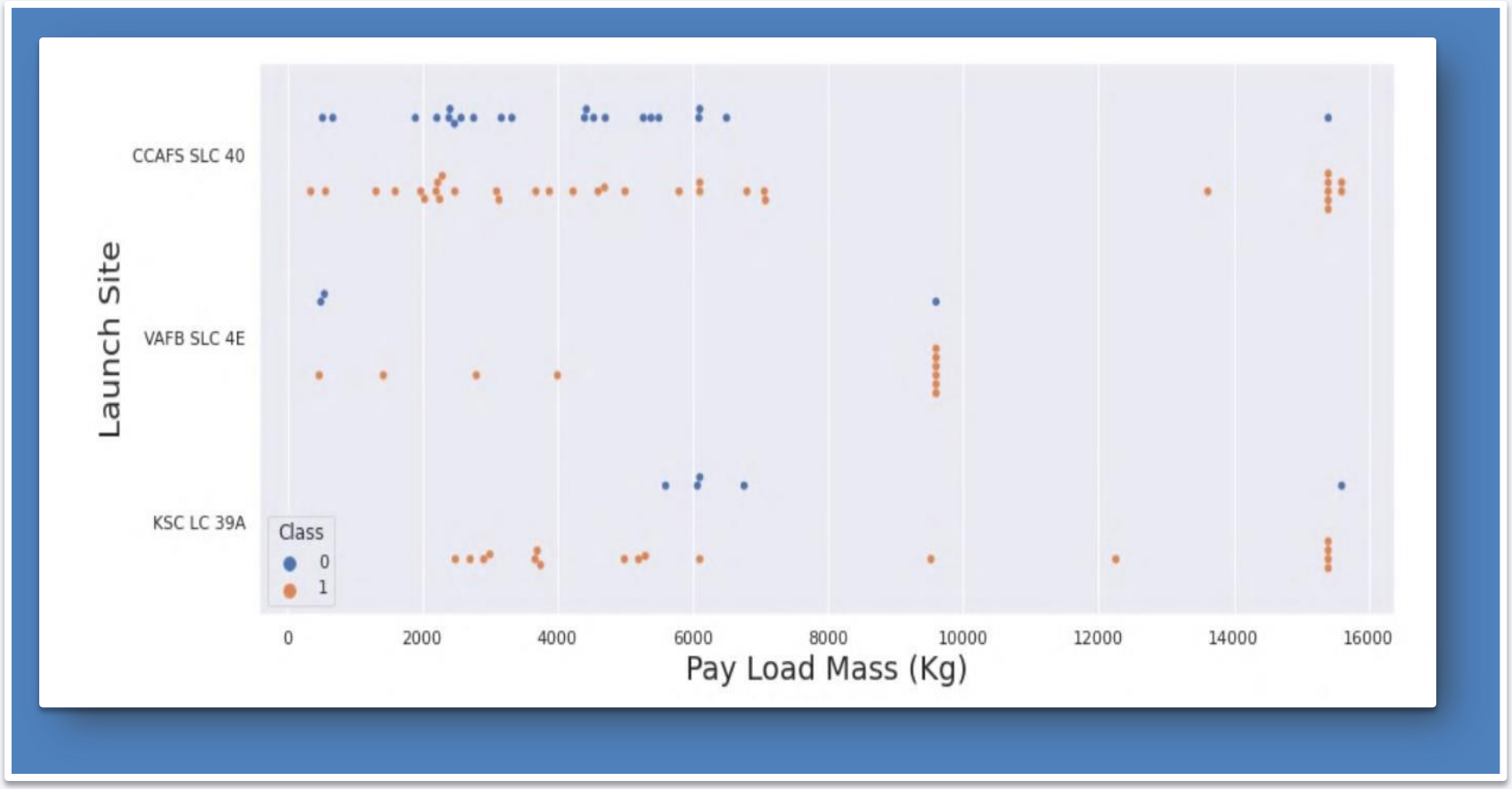


Based on the scatter plot, it is evident that there is a positive relationship between the number of flights from a launch site and the success rate. Generally, as the flights amount increases, the success rate tends to be higher. However, it is worth noting that site CCAFS SLC40 exhibits the least apparent pattern in this relationship.

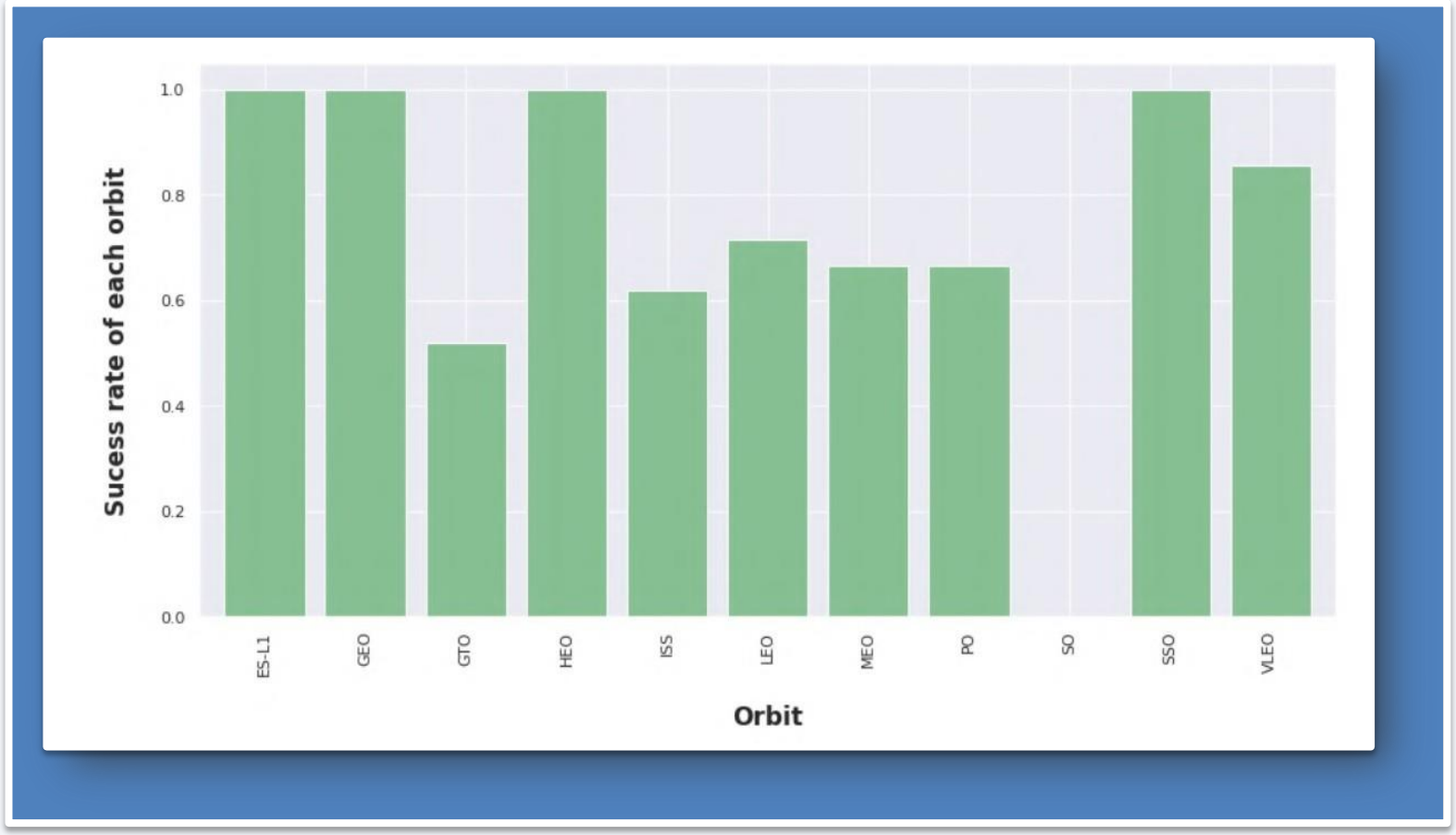
Payload vs. Launch Site

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

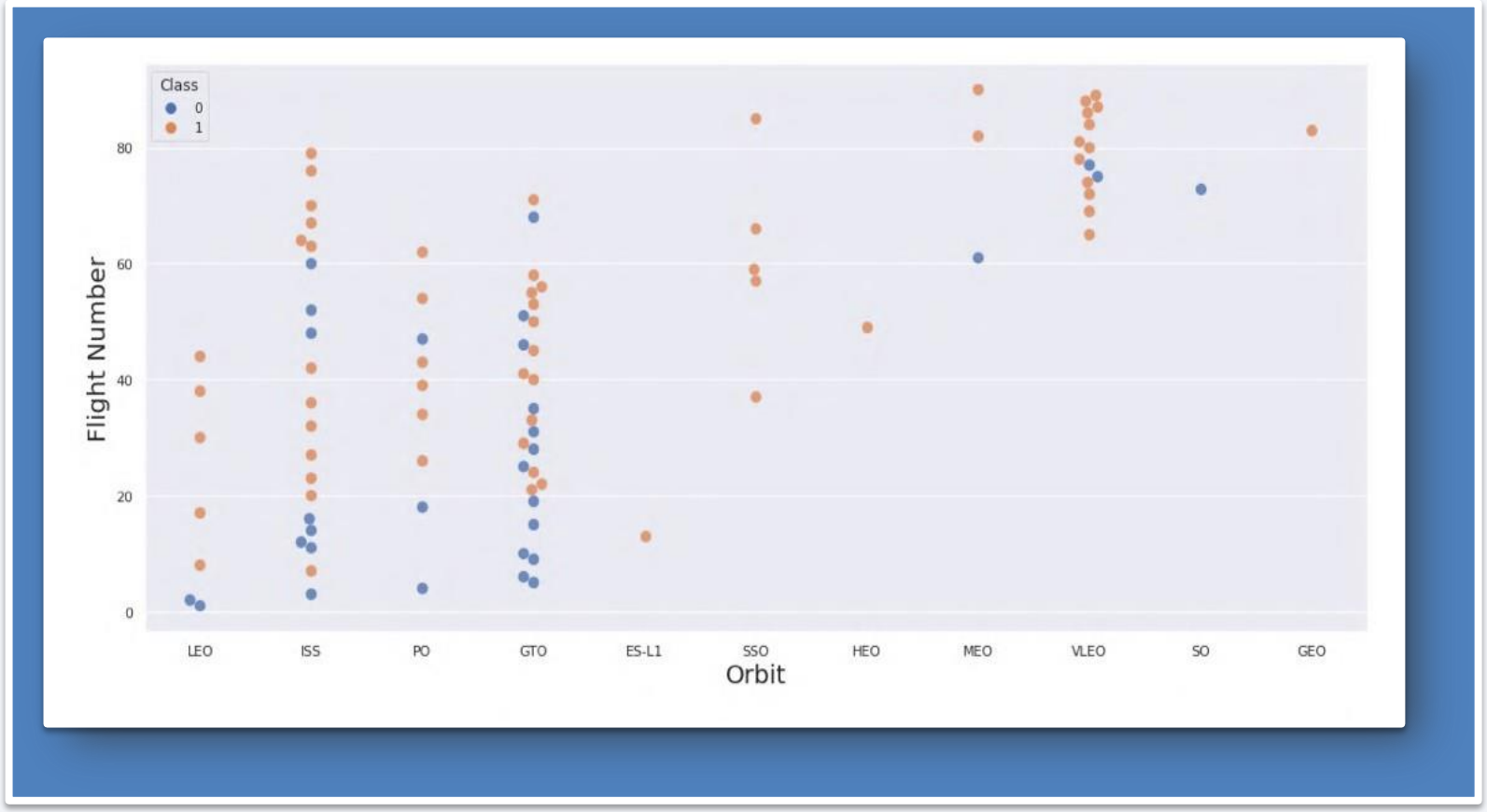


This figure illustrates how different orbits can impact landing outcomes. It indicates that certain orbits, such as SSO, HEO, GEO, and ES-L1, have a 100% success rate, while the SO orbit has a 0% success rate. However, upon closer analysis, it is important to note that some of these orbits only have one occurrence in the dataset, namely GEO, SO, HEO, and ES-L1. Therefore, it is necessary to gather more data to identify any discernible patterns or trends before drawing conclusive conclusions.

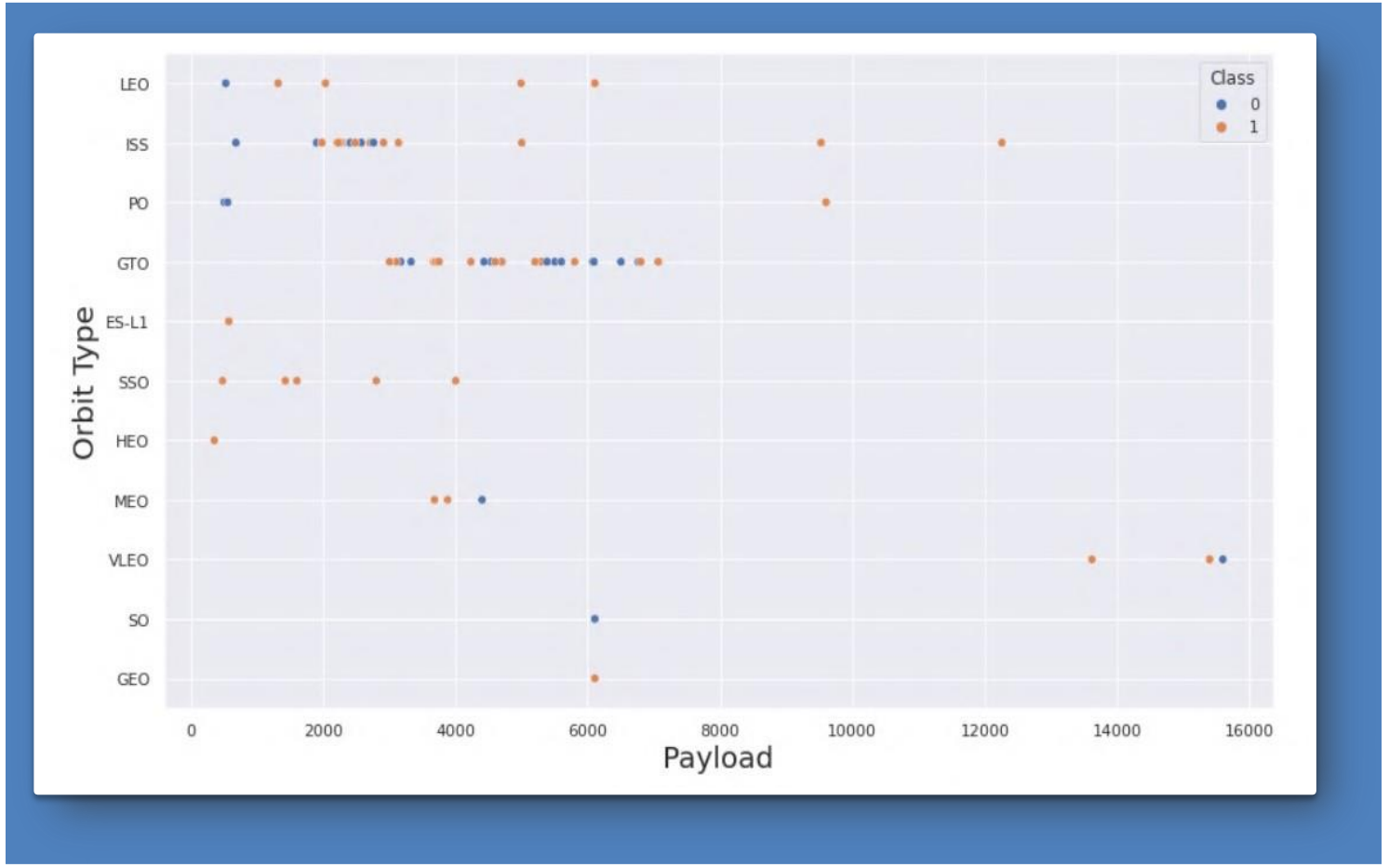
Flight Number vs. Orbit Type

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 based on the available data, a heavier payload demonstrates a positive impact on LEO (Low Earth Orbit), ISS (International Space Station), and P0 (Polar Orbit). In these orbits, increasing the payload mass correlates with a more favorable landing outcome. Conversely, in MEO (Medium Earth Orbit) and VLEO (Very Low Earth Orbit), a heavier payload has a negative impact on the landing outcome.egative impact on MEO and VLEO orbit.

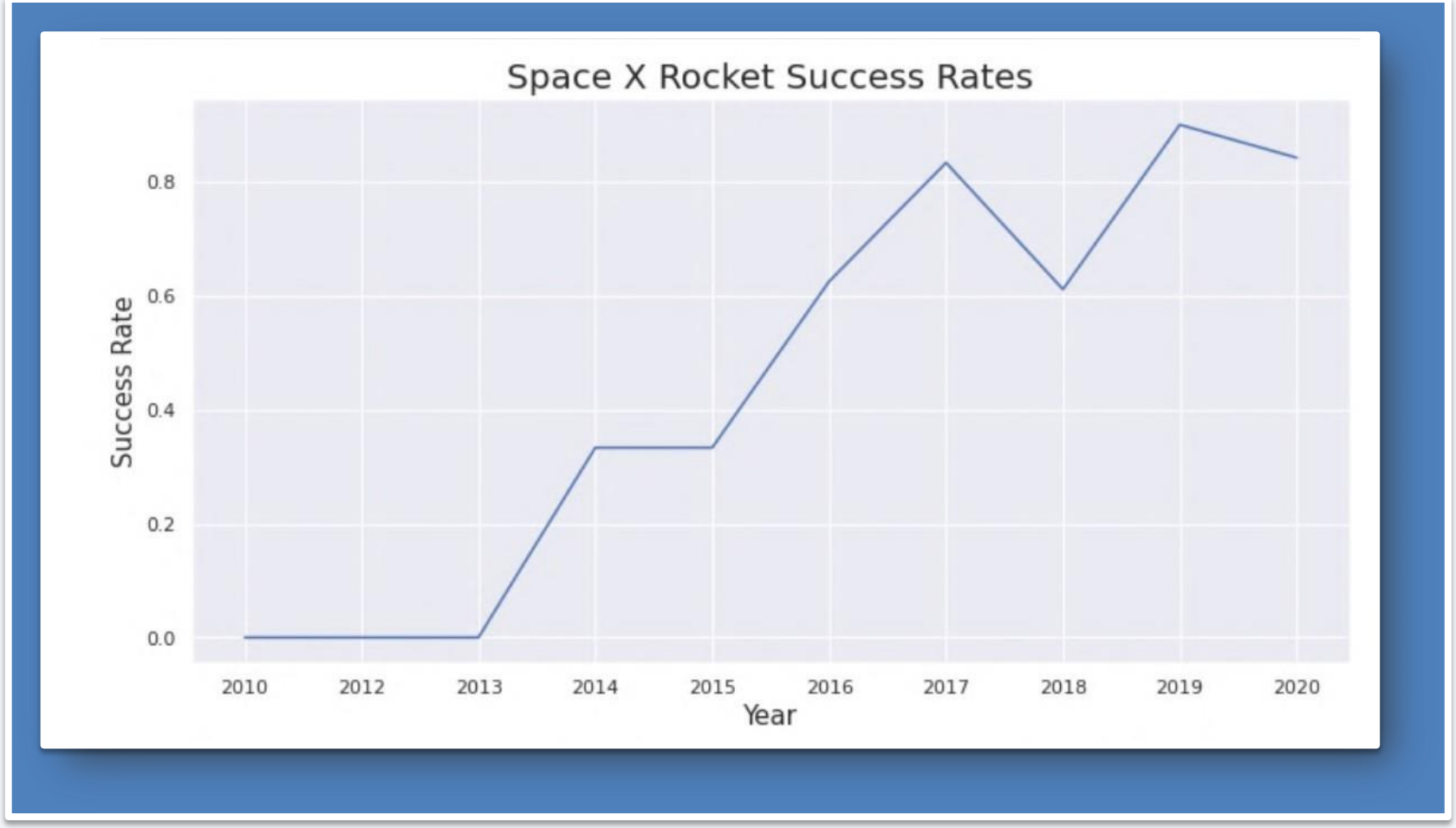
GTO orbit seem to depict no relation between the attributes.

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.

In [5]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

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

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

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Success	Failure (parachute)
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	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

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Average Payload Mass by Booster Version F9 v1.1
2928

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 Succesful Landing Outcome in Ground Pad
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3
sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

First Succesful Landing Outcome in Ground Pad
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 SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
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 wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

Failure Mission
1

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:32731/bludb
Done.
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

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

2015 Launch Records

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.
databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu01qde00.databases.appdomain.cloud:32731/bludb
Done.
```

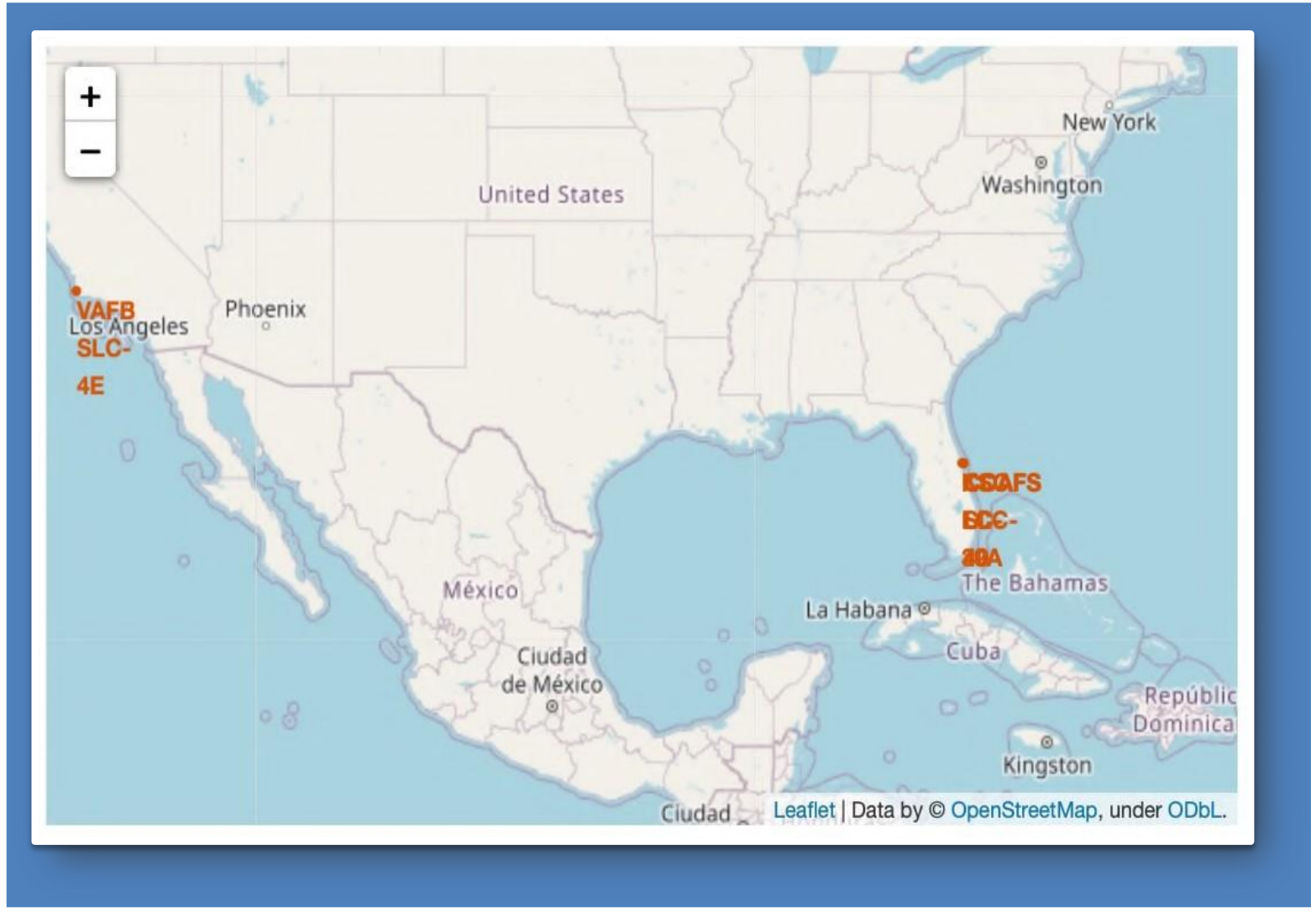
Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

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.


Section 3
Launch Sites
Proximities Analysis

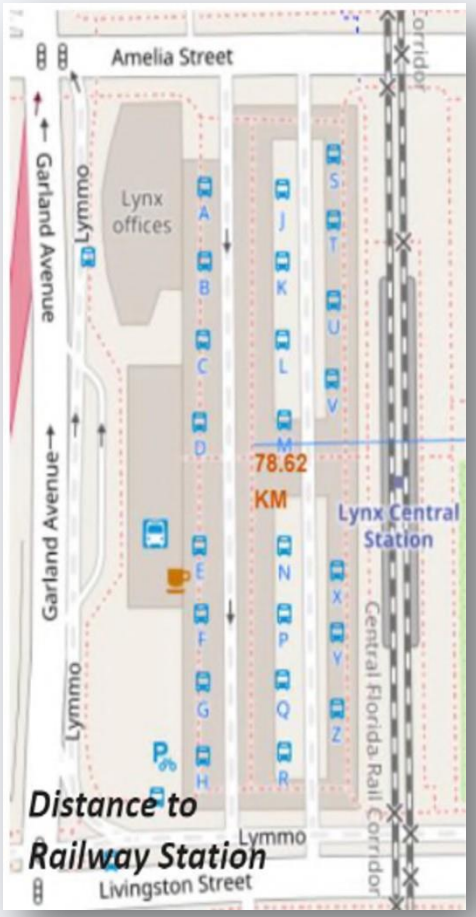
Location of all the Launch Sites

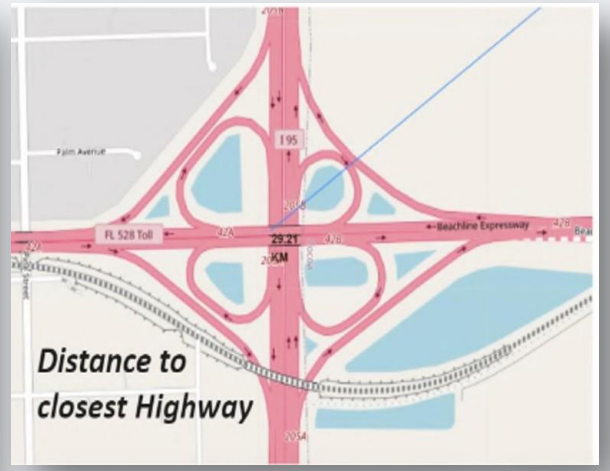


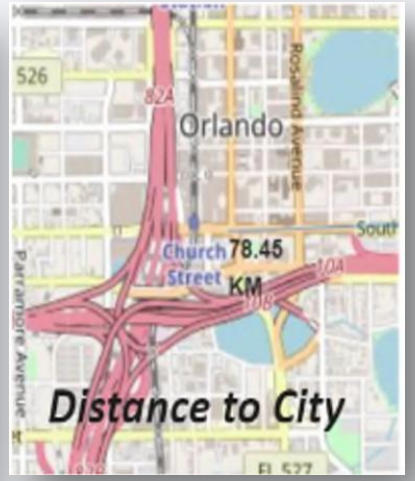
all the SpaceX launch sites are located inside the United States

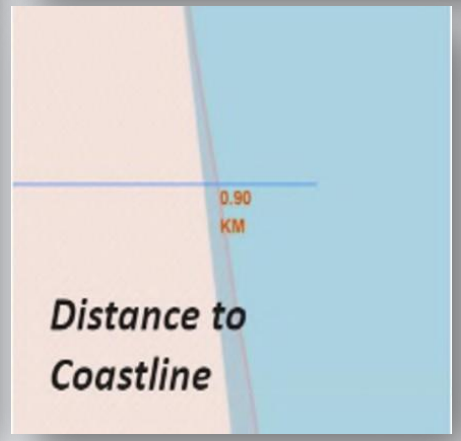
Markers showing launch sites with color labels

Launch Sites Distance to Landmarks

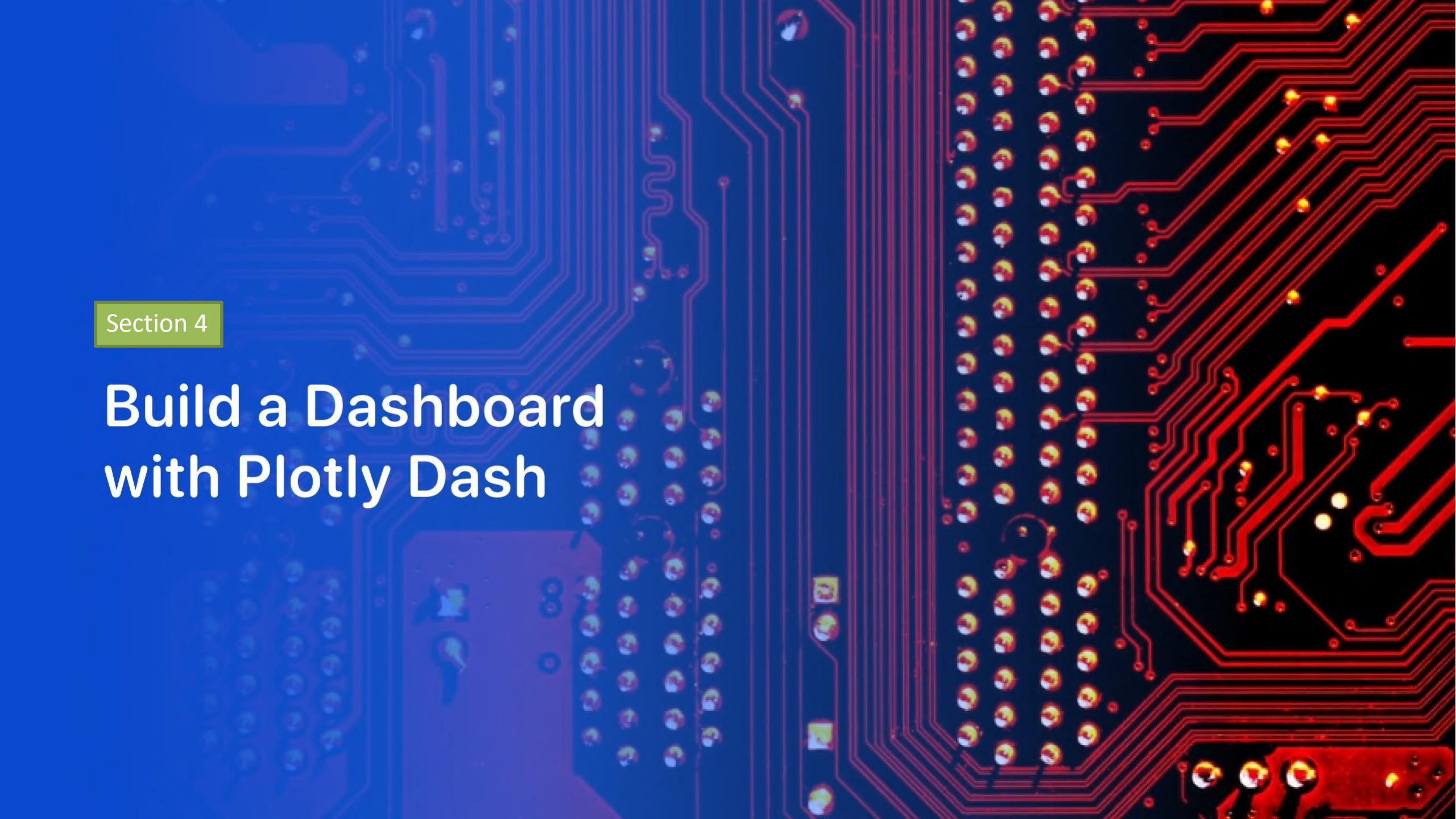




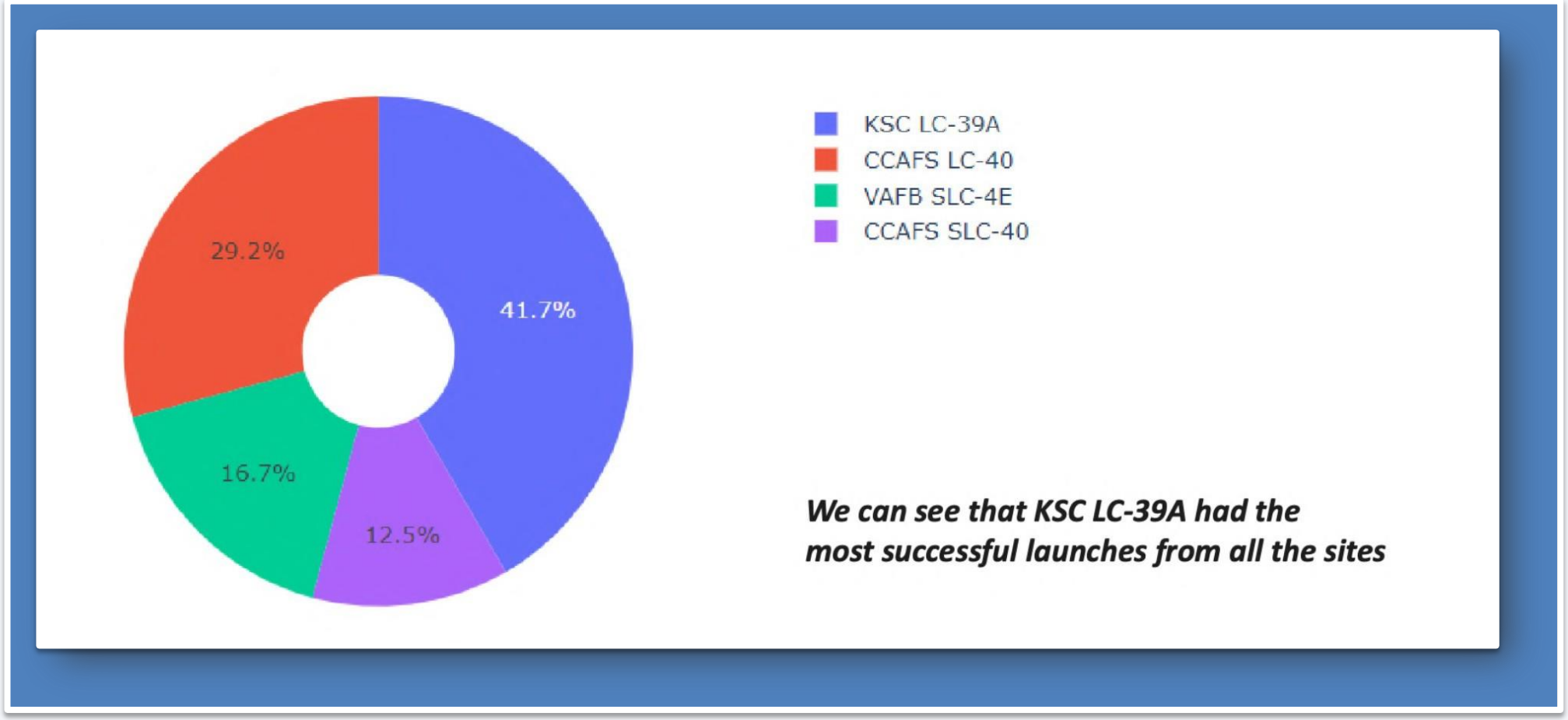




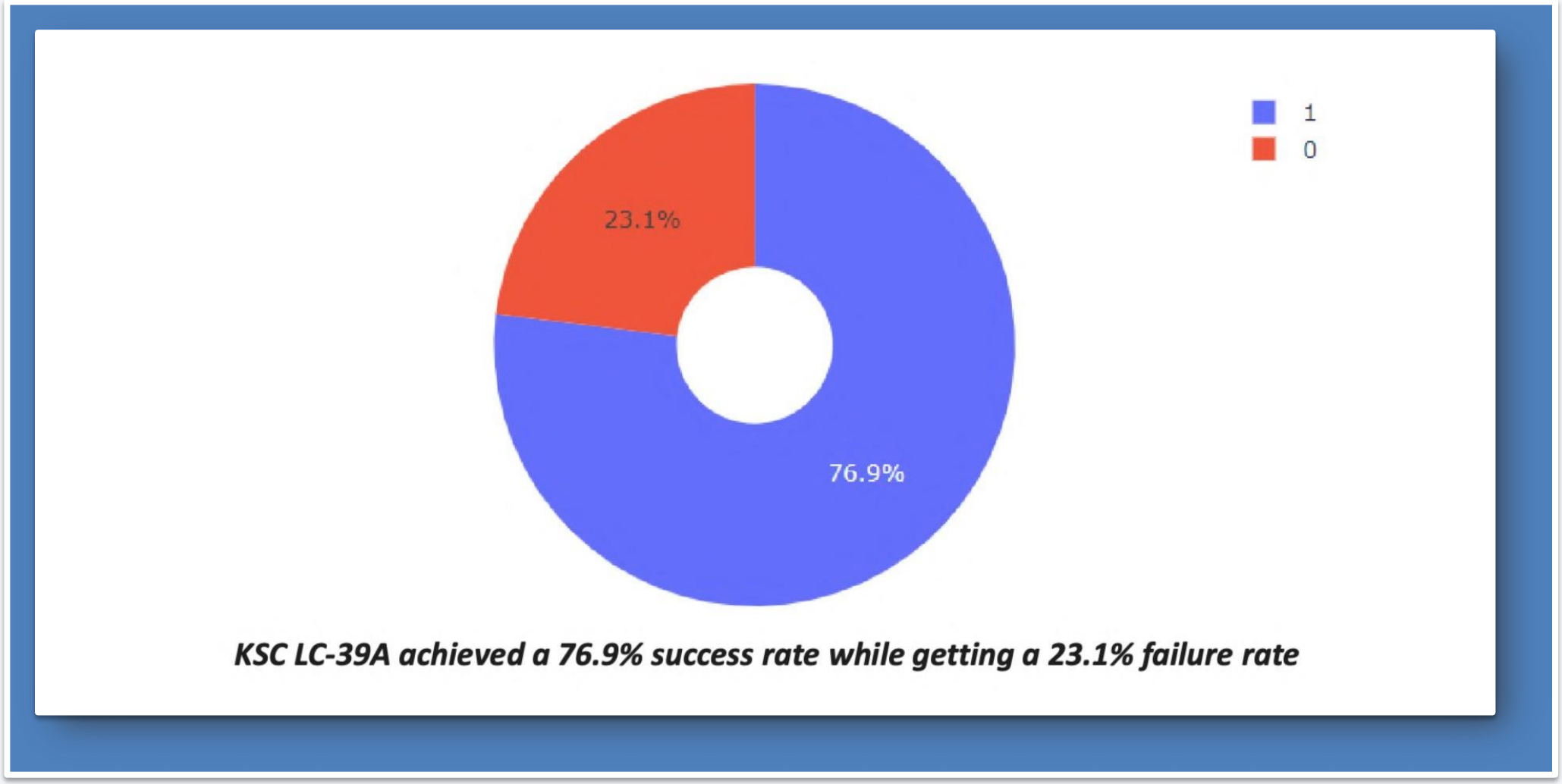
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

The success percentage by each sites.



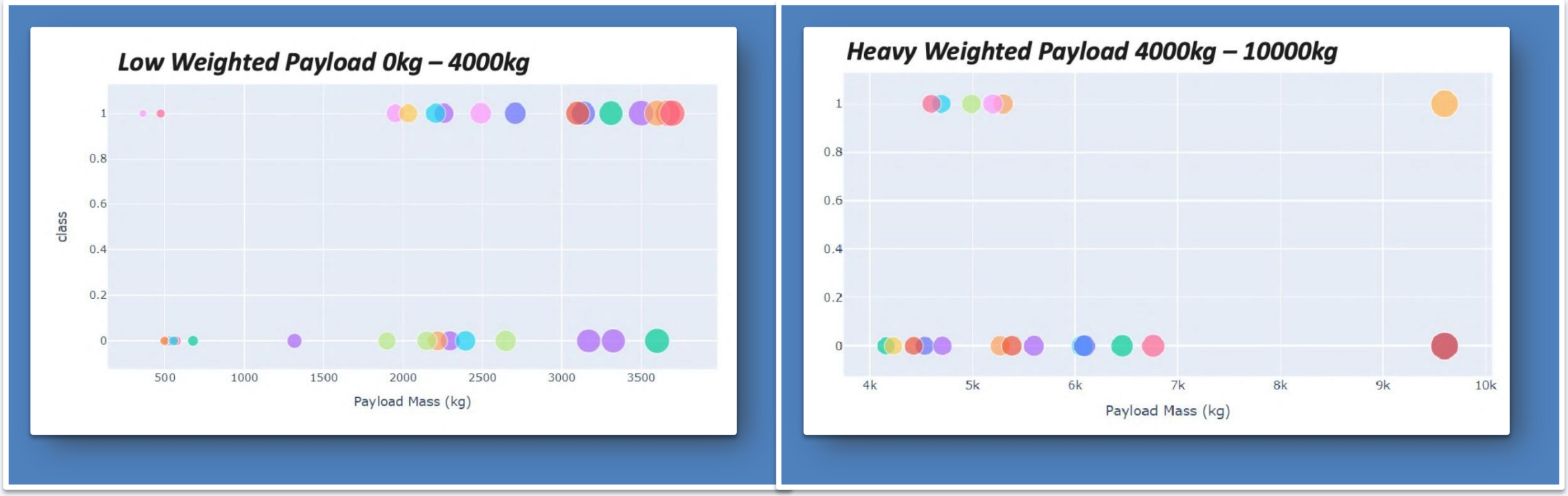
The highest launch-success ratio: KSC LC-39A



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs Launch Outcome Scatter Plot

success rate for low weighted payload is higher than heavy weighted payload



Section 5

Predictive Analysis (Classification)

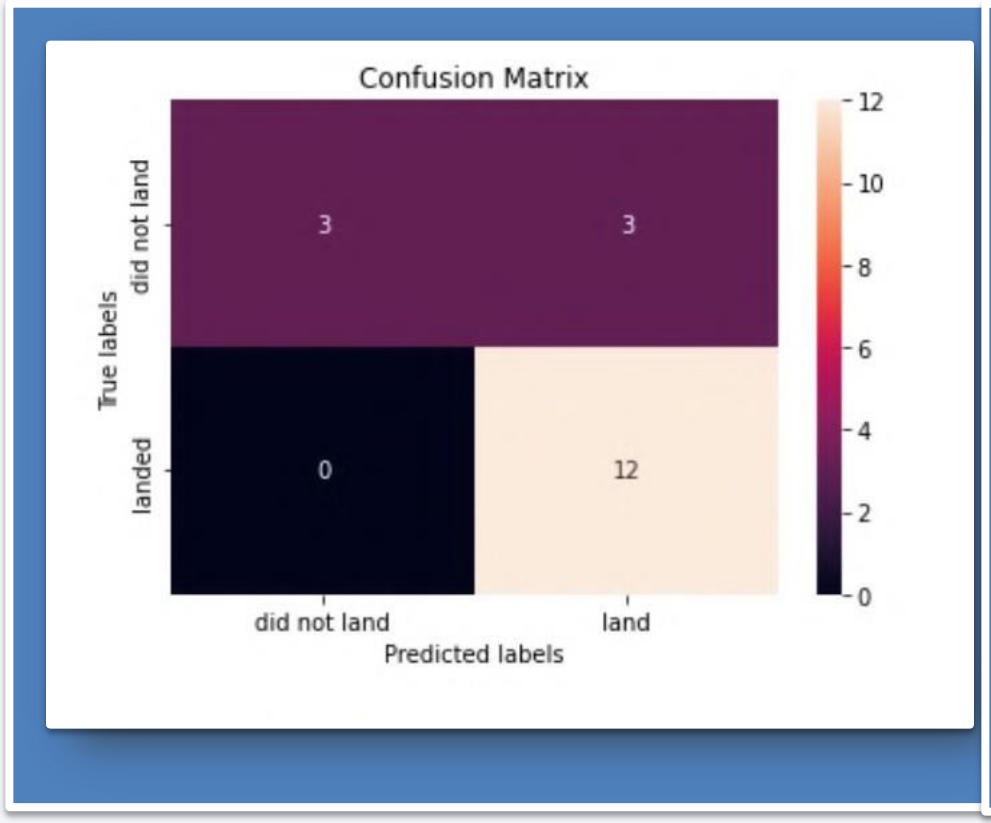
Classification Accuracy

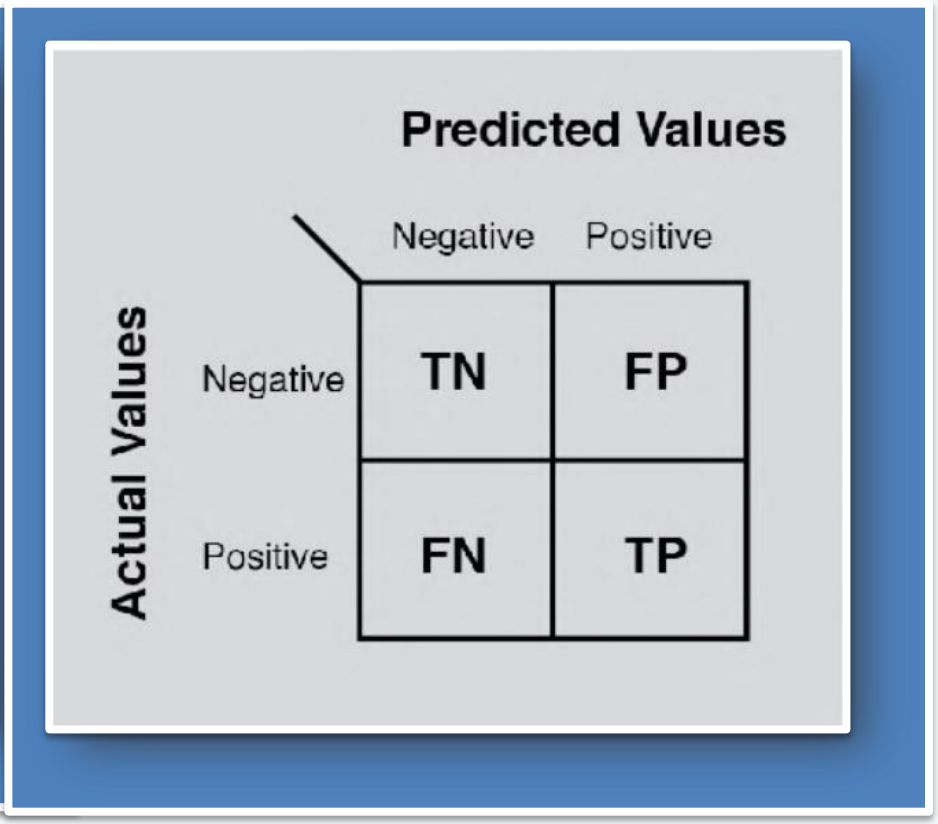
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params 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 0.9017857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_sampl
es_split': 10, 'splitter': 'random'}
```

Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.





Conclusions

We can conclude that:

- 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!