IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Berk Demir
05.04.2023

Outline

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

Executive Summary

Methodologies:

- Collected data through API and web scraping
- Cleaned and processed data through data wrangling
- Conducted exploratory data analysis (EDA) using SQL queries and data visualization techniques
- Conducted interactive visual analytics using Folium for geographical data
- Built predictive models using machine learning algorithms
- Summarized all results, including EDA findings, interactive analytics screenshots, and predictive analytics results from the machine learning lab.

Results:

- Results of Exploratory Data Analysis
- Screenshots of Interactive Analytics
- Results of Predictive Analytics using Machine Learning

Introduction

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Spaces X's Falcon 9 launch like regular rockets. Unlike other rocket providers, SpaceX's Falcon 9 Can recover the first stage. Sometimes the first stage does not land. Sometimes it will crash as shown in this clip. Other times, Space X will sacrifice the first stage due to the mission parameters like payload, orbit, and customer. In this capstone, i will take the role of a data scientist working for a new rocket company. Space Y that would like to compete with SpaceX founded by Billionaire industrialist Allon Musk. My job is to determine the price of each launch. I will do this by gathering information about Space X and creating dashboards for my team. I will also determine if SpaceX will reuse the first stage. Instead of using rocket science to determine if the first stage will land successfully, I will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Problems Included in the Project:

- Identifying the variables that are affecting the result of landing
- The relation between each variable and how much affect they have
- How the landing performance of the rockets can be improved

Section 1

Methodology

Methodology

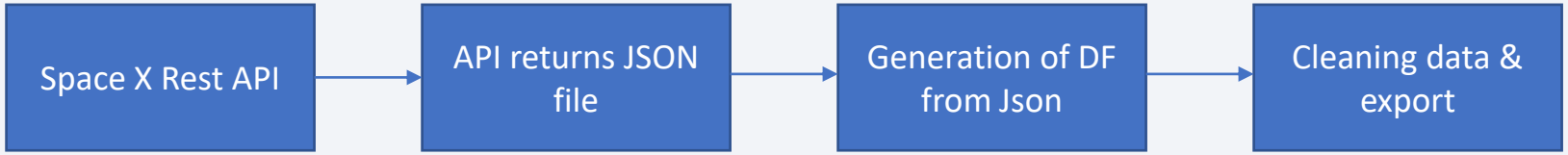
Executive Summary

- Data collection methodology:
  - Data is collected via usage of API and web scraping
- Perform data wrangling
  - Dependent variable of the dataset transformed int o 0 and 1's.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic regression, SVM, Decision tree and KNN is tried as model, grid search is used to find the best hyperparameters.

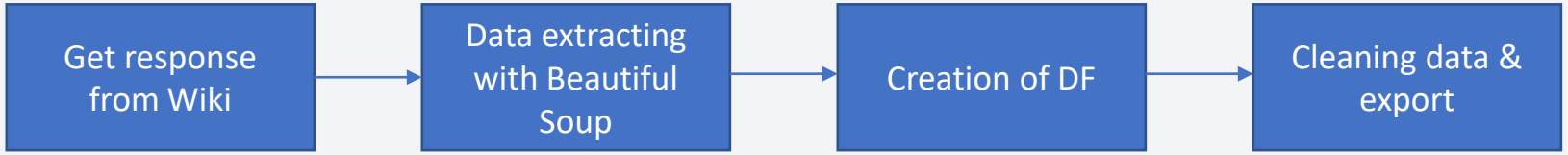
Data Collection

- Data collection refers to the procedure of accumulating and measuring data regarding specific variables within an established framework, which allows individuals to answer pertinent questions and assess outcomes. In this instance, the dataset was obtained through REST API and web scraping from Wikipedia. To utilize REST API, a "get" request was made, followed by the decoding of the response content as Json, and transforming it into a pandas dataframe via json_normalize(). Data was then cleaned, missing values were checked, and filled in with the necessary information. As for web scraping, BeautifulSoup was employed to extract launch records in the form of an HTML table. This table was parsed and then converted to a pandas dataframe for further analysis.

- Space X Rest API



- Wikipedia Webscrapping



Data Collection – SpaceX API

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

response = requests.get(spacex_url)
```

```
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want and the flight n
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
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
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 da
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)]
```

Get request for rocket launch data using API

json_normalize method is used to convert json result into a dataframe

data cleaning performed and missing values are filled

Code File

Data Collection - Scraping

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

```
soup = BeautifulSoup(response.text, "html.parser")
```

```
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')
```

Request to get the text inside the url

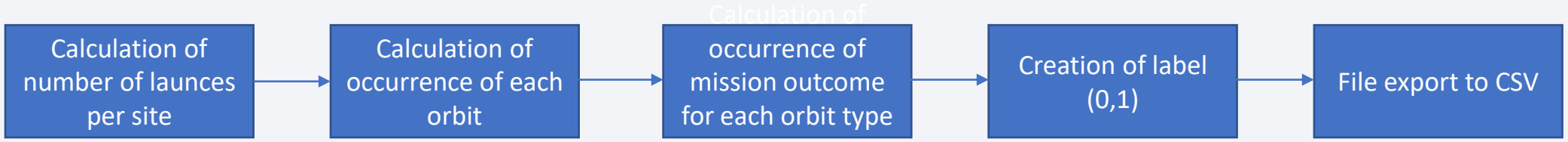
Creating a beautiful soup object based on the response variable

Extracting variable names from wikipedia

Code File

Data Wrangling

Data Wrangling refers to the procedure of tidying and integrating complex and untidy datasets for simple access and Exploratory Data Analysis (EDA). Initially, we will compute the amount of launches per site, followed by calculating the frequency and total number of mission outcomes based on the orbit type. Next, we will generate a landing outcome tag using the information from the outcome column. This will simplify further analysis, visualization, and Machine Learning (ML) processing. Finally, we will export the final results to a CSV file.



Code File

EDA with Data Visualization

Our initial approach was to employ scatter plots to identify any connections 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 provide a clear visualization of the relationship between attributes. After analyzing the graphs and detecting any patterns, it becomes apparent which factors are the most influential in determining the success of landing outcomes.

After detecting any potential correlations through scatter plots, we will employ additional visualization techniques such as bar graphs and line plots for further examination. Bar graphs provide a straightforward approach to interpret the relationship between attributes. In our situation, we will utilize a bar graph to ascertain which orbits have the greatest likelihood of success. We will then use a line graph to showcase any trends or patterns of attributes over time, such as yearly launch success trends. Lastly, we will utilize Feature Engineering to generate dummy variables for categorical columns, which can be utilized in predicting success in future modules.

Code File

EDA with SQL

- We utilized SQL queries to extract and interpret data from the dataset in the following ways:
- We displayed the names of the unique launch sites in the space mission.
- We retrieved five records where the launch sites begin with the string 'CCA.'
- We calculated the total payload mass carried by boosters launched by NASA (CRS).
- We computed the average payload mass carried by booster version F9 v1.1.
- We listed the date when the first successful landing outcome on the ground pad was achieved.
- We retrieved the names of the boosters that have achieved success on drone ships and carried a payload mass greater than 4000 but less than 6000.
- We tallied the total number of successful and failed mission outcomes.
- We compiled a list of the booster versions that have carried the maximum payload mass.
- We retrieved the records that display the month names, failure landing outcomes on drone ships, booster versions, and launch sites for the months in the year 2015.
- We ranked the count of successful landing outcomes between the dates 04-06-2010 and 20-03-2017 in descending order.

Code File

Build an Interactive Map with Folium

- To visualize the launch data, we used the latitude and longitude coordinates of each launch site to create an interactive map. We centered the map on the NASA Johnson Space Center in Houston, Texas and added a red circle with a label indicating its name. Then, we added red circle markers at each launch site's coordinates with labels showing their names.
- To differentiate the launch outcomes, we assigned the success and failure dataframes to classes 1 and 0, respectively, and plotted them on the map using MarkerCluster(). Successful landings were marked with green markers, while unsuccessful ones were marked with red markers.
- In addition to showing the launch sites and their outcomes, we used the Haversine's formula to calculate the distances between the launch sites and various landmarks such as railways, highways, coastlines, and nearby cities. To display this information on the map, we added markers and plotted lines to show the distances between the launch sites and these landmarks. This helped us answer questions such as how close the launch sites are to railways, highways, coastlines, and nearby cities.
- Overall, this interactive map with different markers, labels, and lines, allowed us to easily visualize the launch sites and their surroundings, as well as the number of successful and unsuccessful landings.

Code File

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.The dashboard comprises various interactive components, including dropdown, pie chart, rangeslider, and scatter plot.

- The dropdown component, using dash_core_components.Dropdown, allows the user to select one or more launch sites.
- The pie chart, created using plotly.express.pie, displays the total number of successful and failed launches based on the launch site selected with the dropdown component.
- The rangeslider, utilizing dash_core_components.RangeSlider, enables the user to select a specific payload mass range.
- The scatter plot, developed with plotly.express.scatter, demonstrates the correlation between two variables, Success vs Payload Mass.

Code File

Predictive Analysis (Classification)

Data Preparation:
- The first step was to load the dataset into the system.
- The data was then normalized to ensure consistency and improve model accuracy.
- Next, the dataset was split into training and test sets for model evaluation.

Model Preparation:
- A variety of machine learning algorithms were selected for analysis.
- Parameters were set for each algorithm using GridSearchCV to optimize their performance.
- The selected models were then trained using the training dataset.

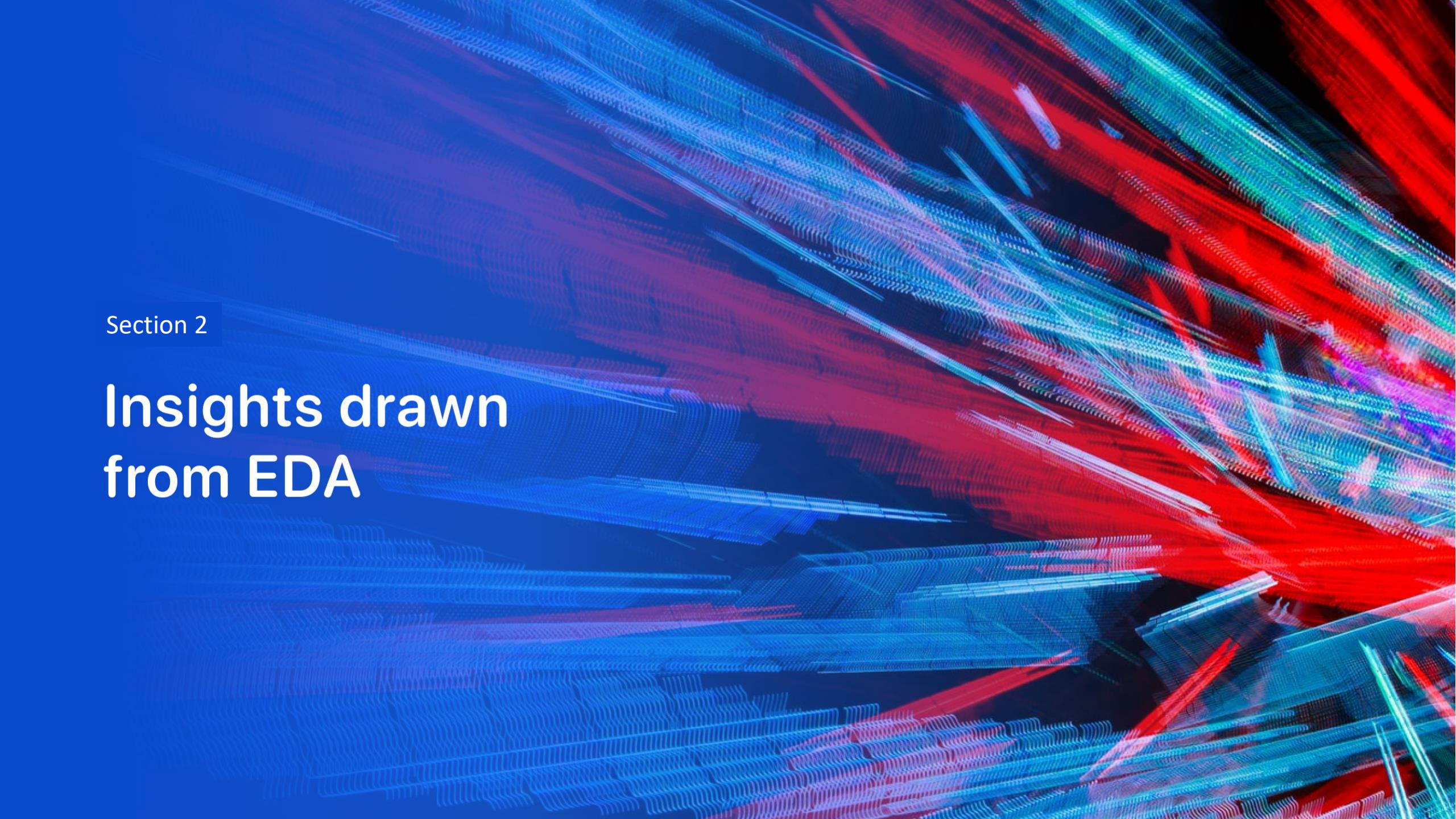
Model Evaluation:
- The best hyperparameters for each model were determined.
- The accuracy of each model was computed using the test dataset.
- A Confusion Matrix was plotted to visually assess the model's performance.

Model Comparison:
- The accuracy of each model was compared to determine which one performed the best.
- The model with the highest accuracy was selected as the final model (see Notebook for the result).

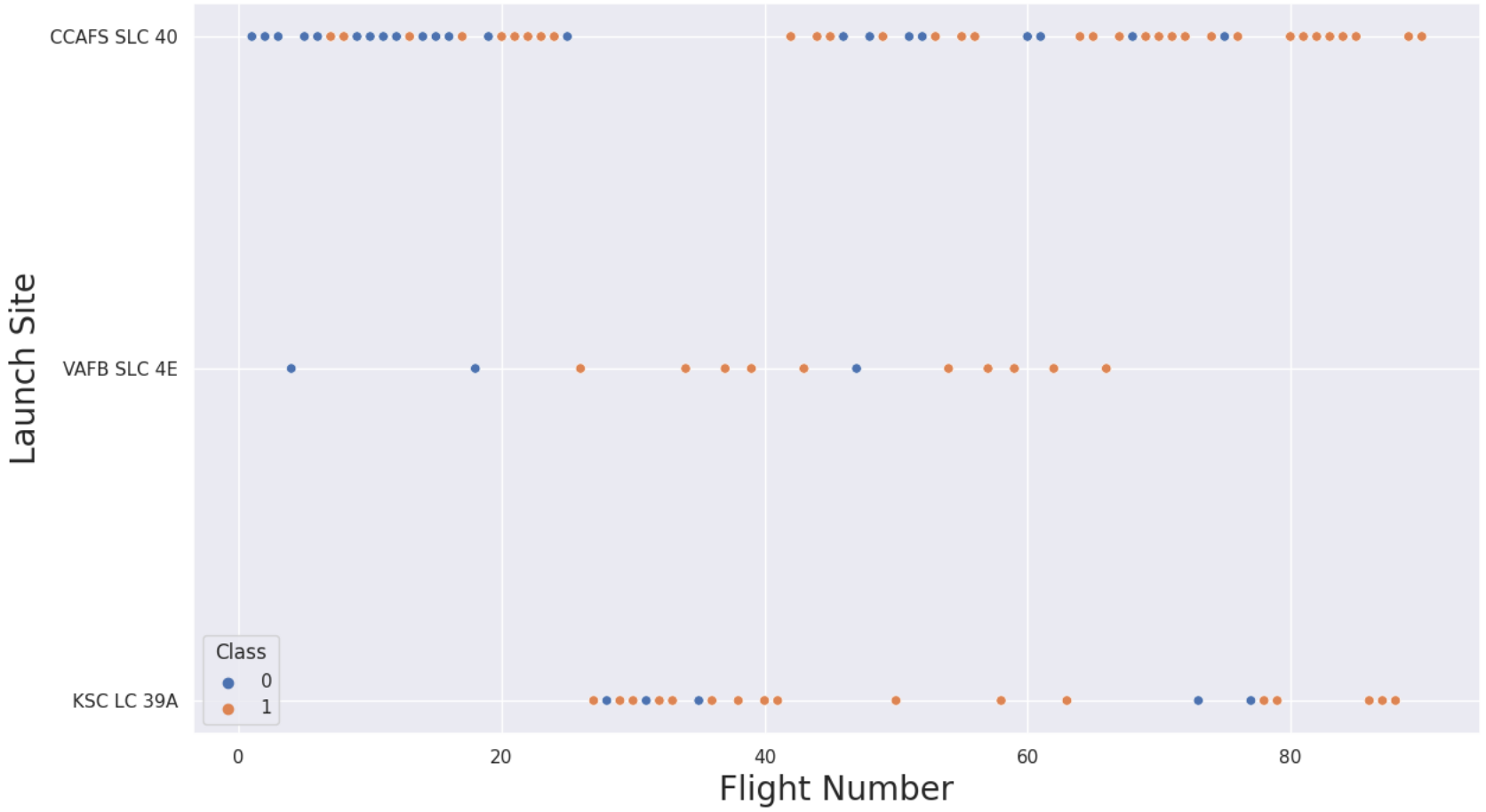
Code File

Results

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

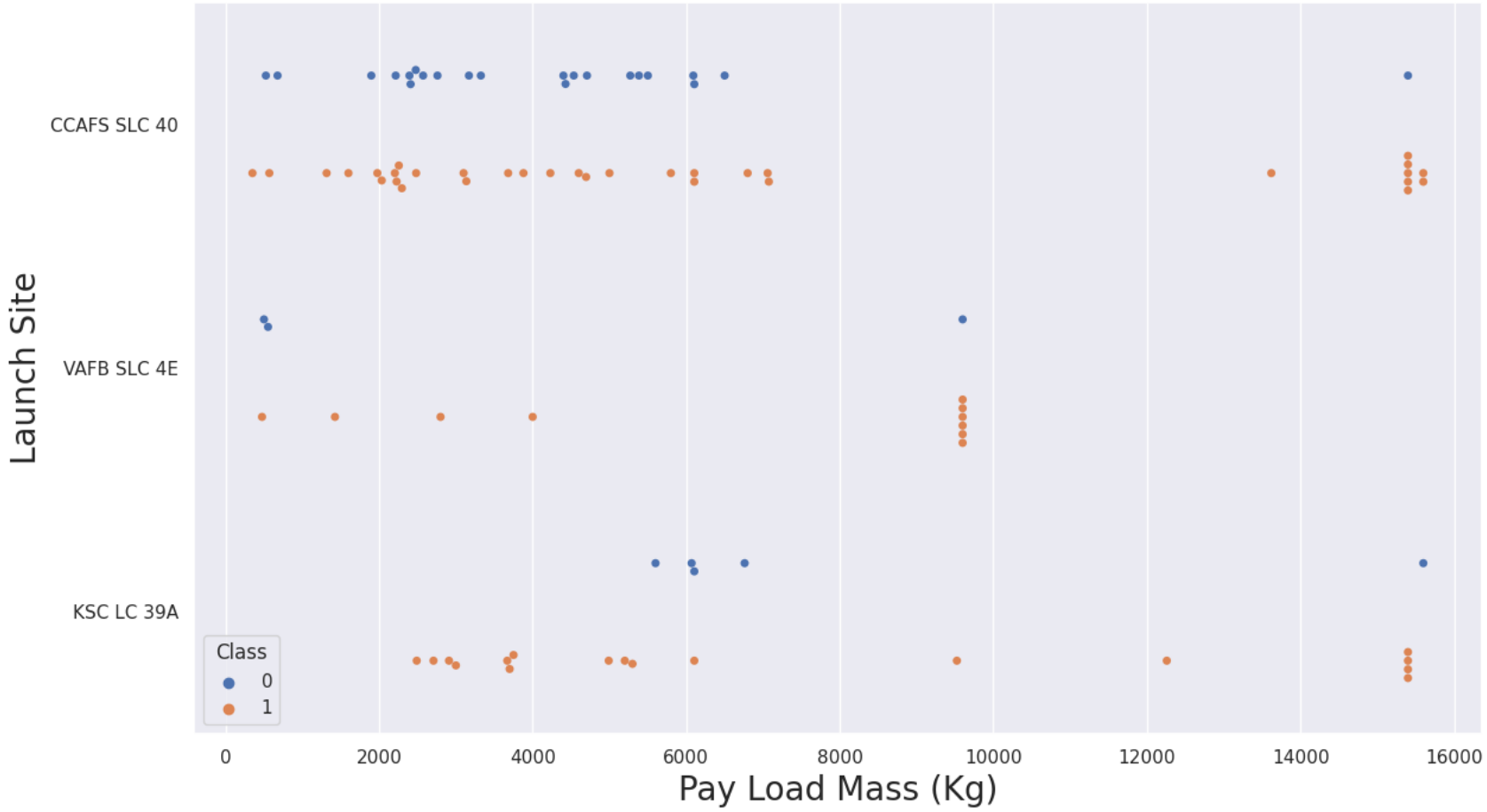

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



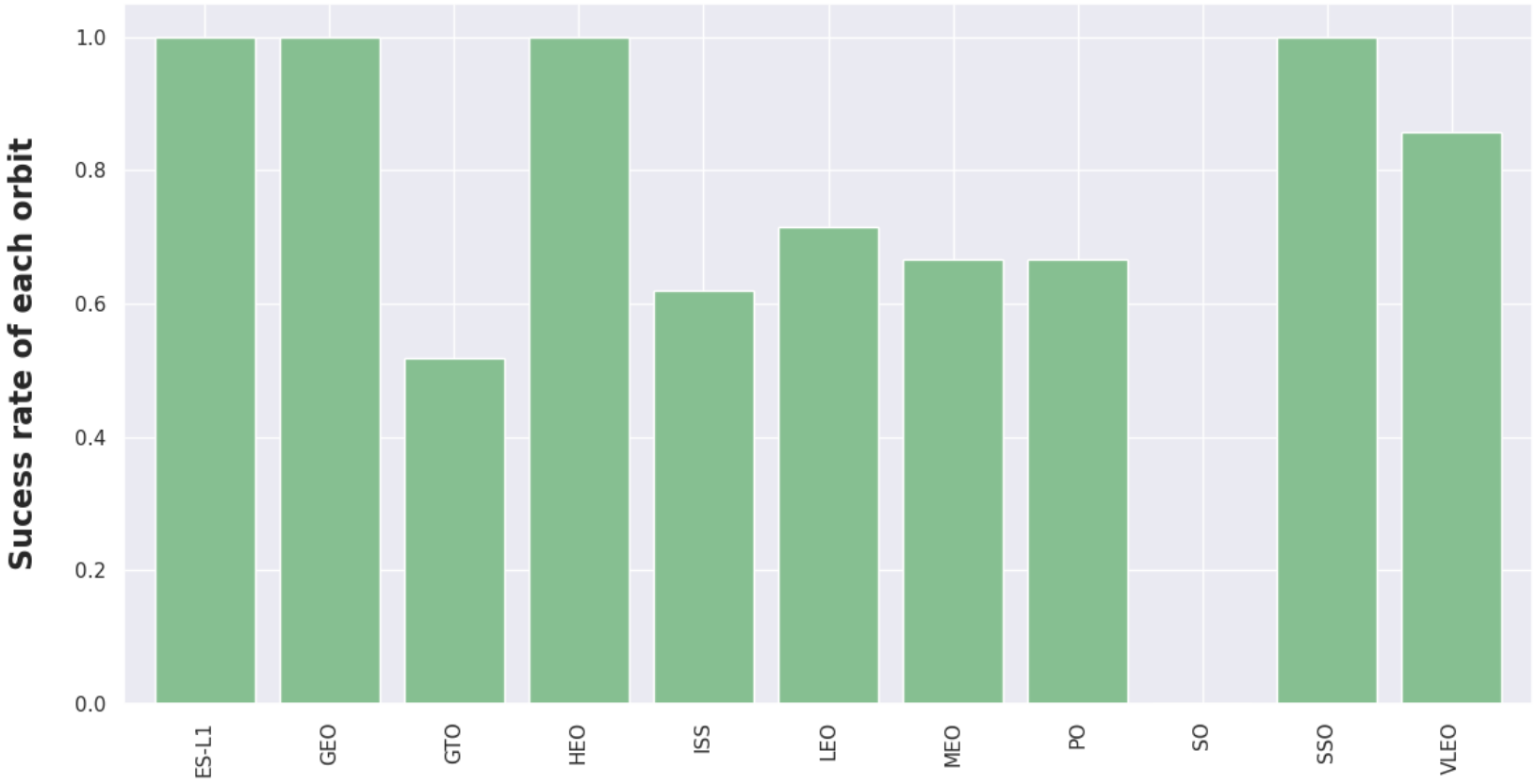
The scatter plot visualizes that as the flights amount of the launch site increases, the success rate also increases. However, the launch site CCAFS SLC40 does not follow this pattern and exhibits the lowest success rate among the launch sites.

Payload vs. Launch Site



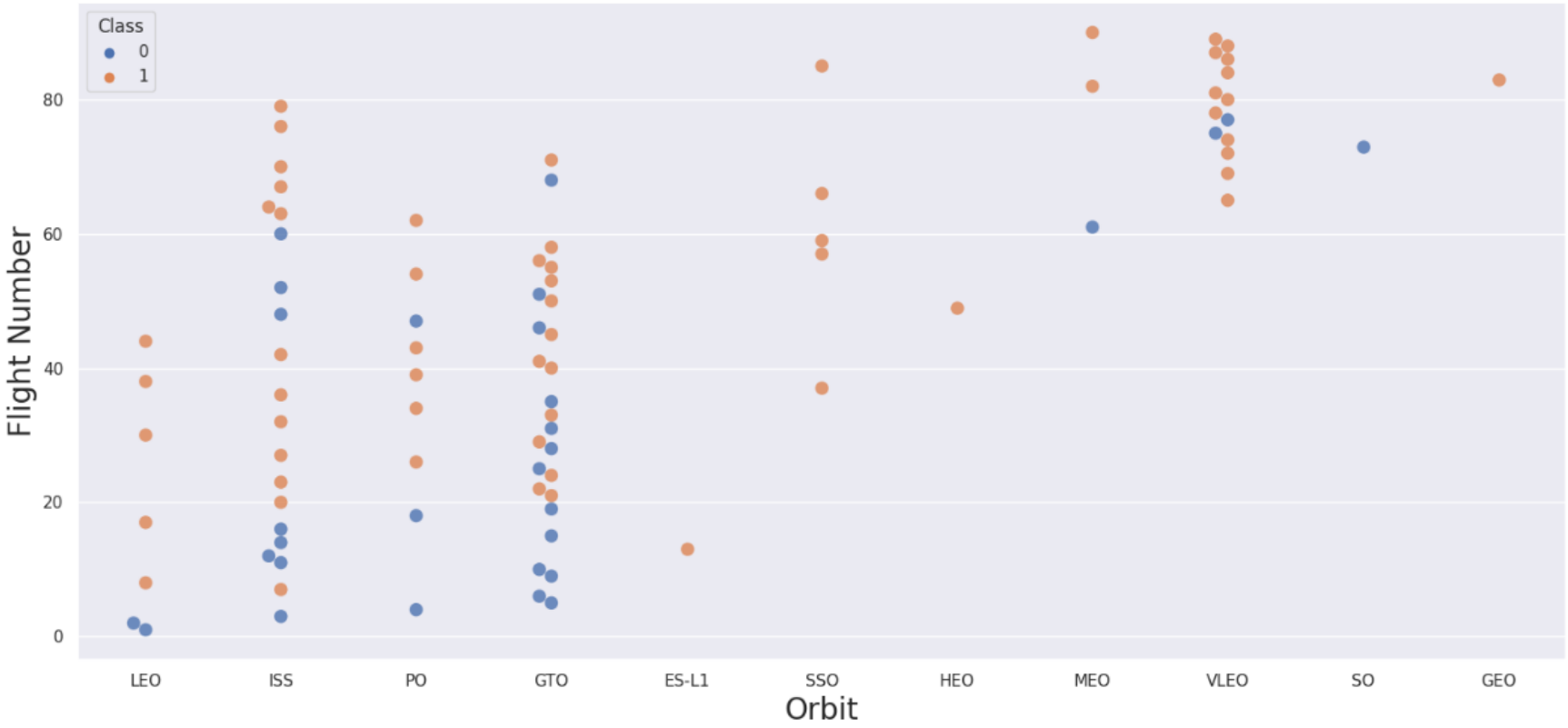
The scatter plot illustrates that when the payload mass exceeds 7000kg, the success probability significantly increases. However, there is no conclusive evidence to suggest that the success rate is dependent on the launch site's payload mass.

Success Rate vs. Orbit Type



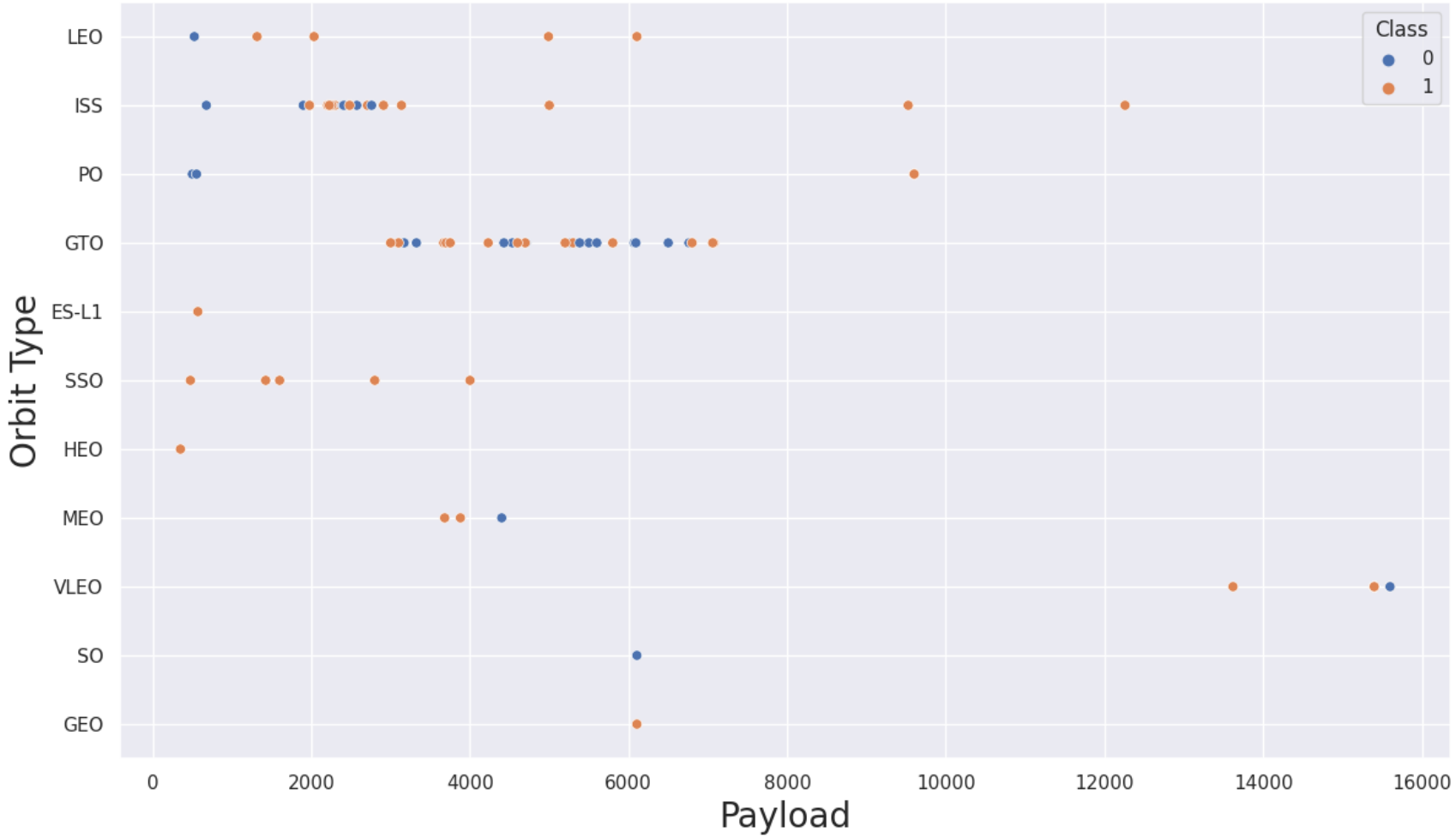
The diagram illustrates how the success rate of landings can be affected by the type of orbit, with certain orbits such as SSO, HEO, GEO, and ES-L1 having a 100% success rate, while SO orbit showed a 0% success rate. However, a more detailed analysis reveals that some of these orbits only had one occurrence, including GEO, SO, HEO, and ES-L1. As a result, more data is required to determine any patterns or trends before any conclusions can be drawn.

Flight Number vs. Orbit Type



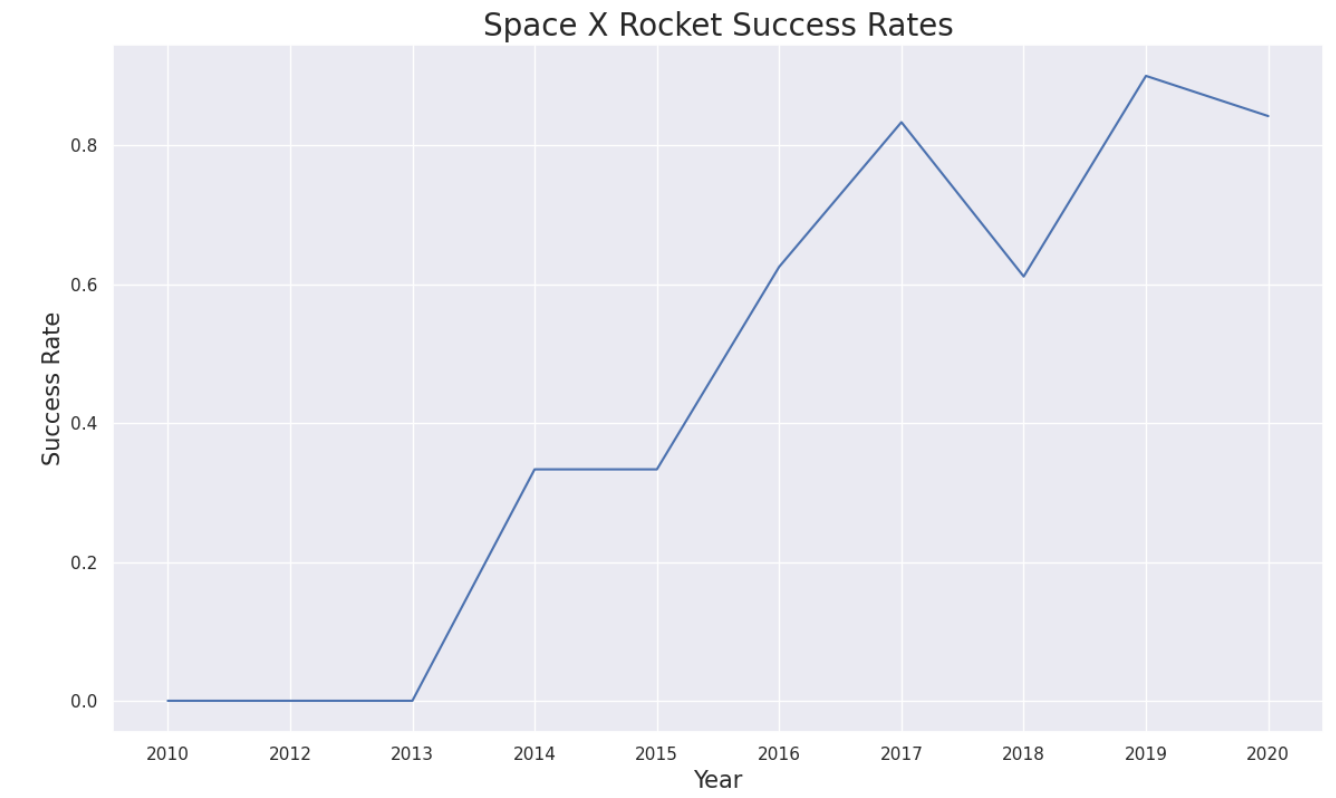
The scatter plot reveals that, in most cases, the success rate tends to increase as the flight number for each orbit increases, particularly for the LEO orbit. However, this relationship does not hold true for the GTO orbit as there is no apparent correlation between the two attributes. Moreover, it is important to exclude the orbits with only one occurrence from the aforementioned statement since they require additional data to draw any meaningful conclusions.

Payload vs. Orbit Type



Payload weight plays a significant role in the success rate of launches in specific orbits. For instance, heavier payloads tend to increase the success rate of the LEO orbit, while reducing payload weight for a GTO orbit can enhance the likelihood of a successful launch.

Launch Success Yearly Trend



Since 2013, we can say that there is an increase in the Space X Rocket success rate. But only in 2018 value drops compared to the previous year.

All Launch Site Names

```
%sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

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

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

The key word DISTINCT is used to show only the unique values for launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Limit functionality is used to get the first five launch sites that contains ‘CCA’

Total Payload Mass

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

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

SUM("PAYLOAD_MASS__KG_")
45596

Total payload is found as 45596 where customer is Nasa

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

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

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

The average payload mass is calculated as 2534 for the booster version F9 v1.1.

First Successful Ground Landing Date

```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

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

MIN("DATE")
01-05-2017

The first successful landing date was found as 01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%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.
```

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

The WHERE clause was utilized to filter for boosters that have landed successfully on a drone ship. Additionally, an AND condition was applied to identify successful landings with payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

SUCCESS	FAILURE
100	1

In the initial SELECT statement, presented the subqueries that provide outcomes. The first subquery counts the number of successful missions, while the second subquery counts the number of unsuccessful missions. The WHERE clause followed by the LIKE clause is used to filter the mission outcomes, and then the COUNT function is used to count the records that have been filtered.

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

```
 * 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

A subquery was employed to filter the data by extracting the maximum payload mass using the MAX function. The primary query utilized the results obtained from the subquery to return the booster version that is distinct (SELECT DISTINCT) with the maximum payload mass.

2015 Launch Records

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

The following query retrieves data on unsuccessful landings and launch details that occurred in 2015, including the month, booster version, and launch site. To obtain the month from the landing date, the Substr function is used, which processes the date and extracts the month or year. The expression Substr(DATE, 4, 2) returns the month, while Substr(DATE, 7, 4) returns the year.

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

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

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

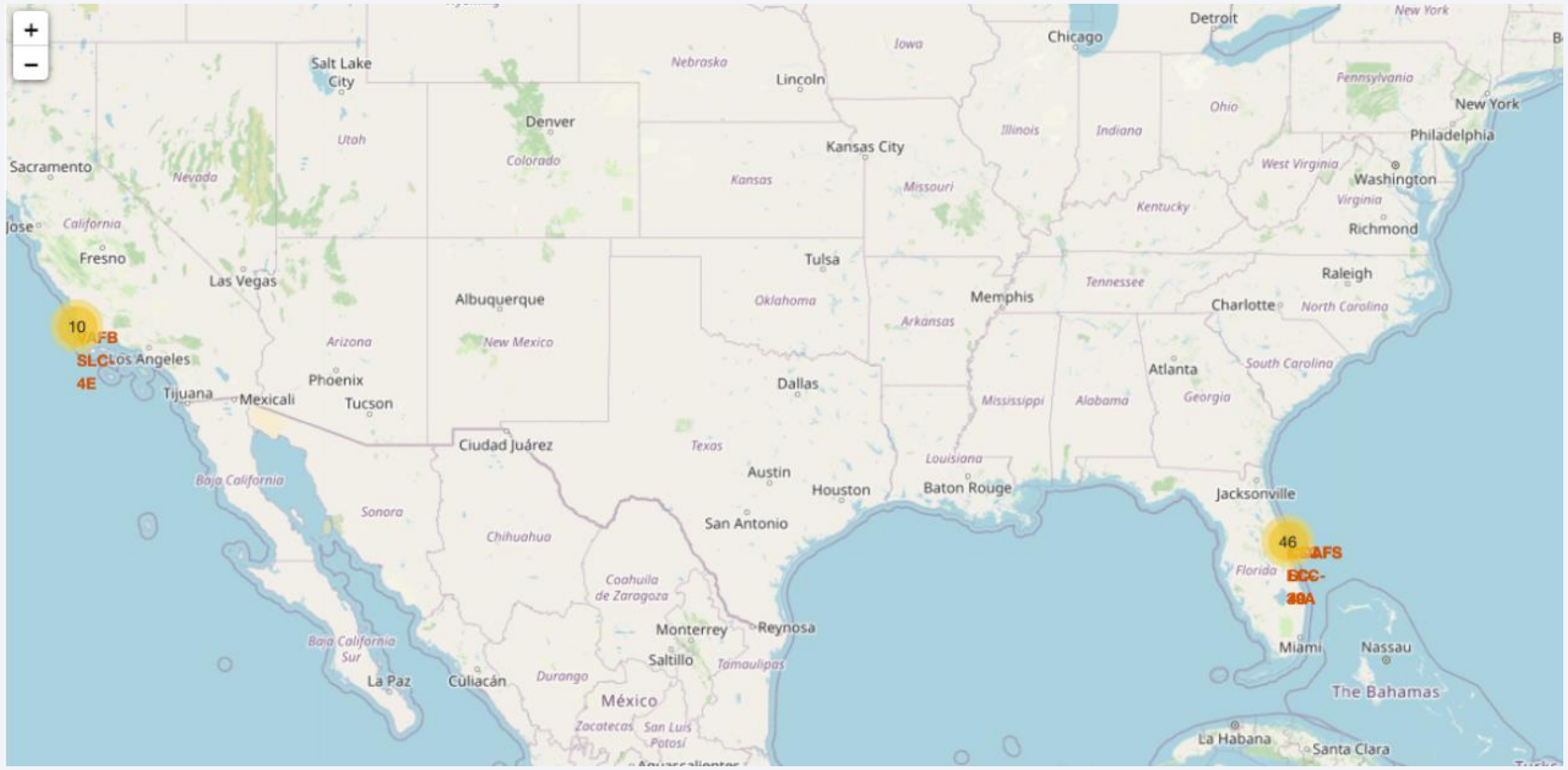
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

The purpose of this query is to retrieve the count of successful landing outcomes within a specific date range from 04/06/2010 to 20/03/2017. The GROUP BY clause groups the results by landing outcome, while the ORDER BY COUNT DESC sorts the results in decreasing order of count.

Section 3

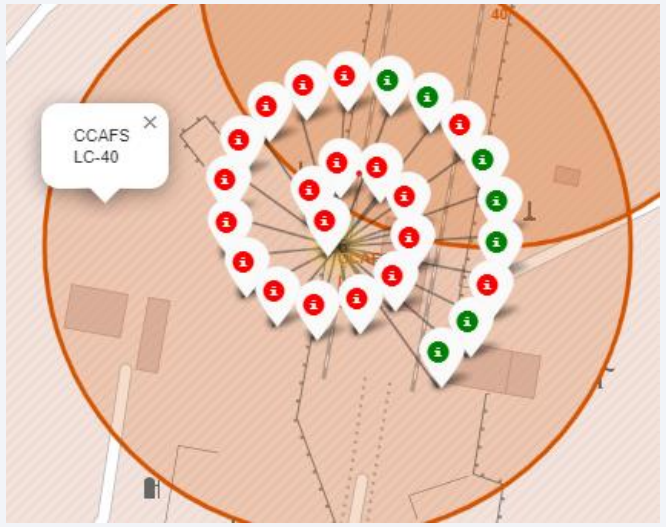
Launch Sites Proximities Analysis

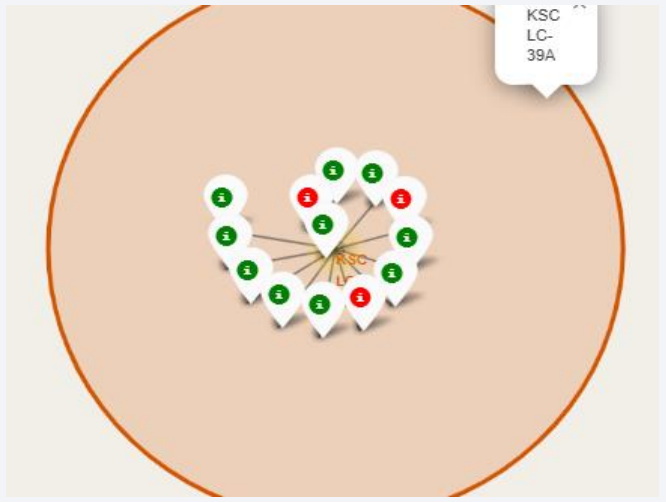
Folium Map – Station Locations



All the SpaceX launch sites are located inside the United States and close to coast side.

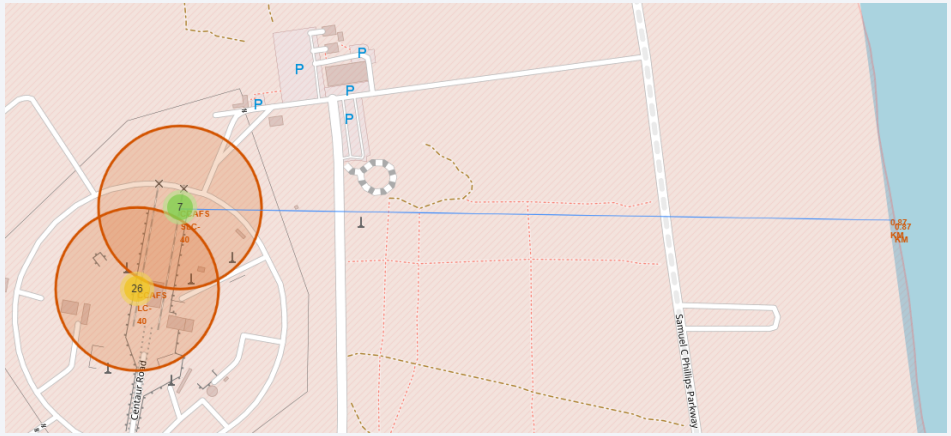
Folium Map - Color Labeled Markers for Sites

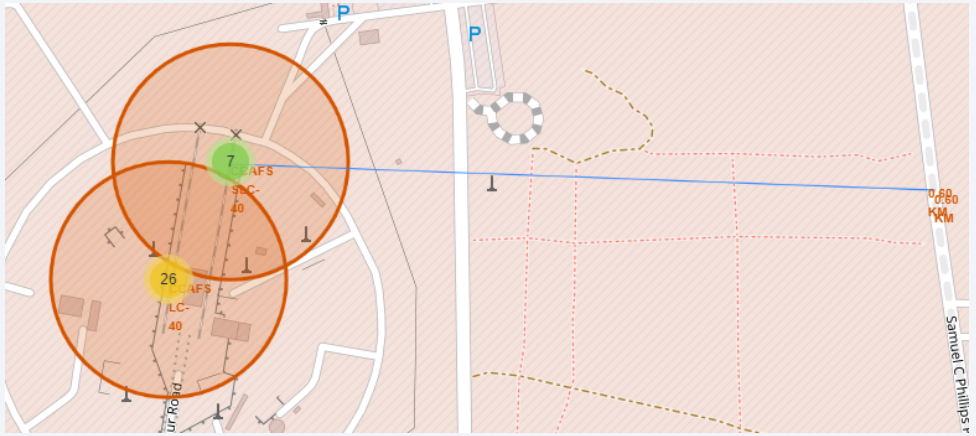


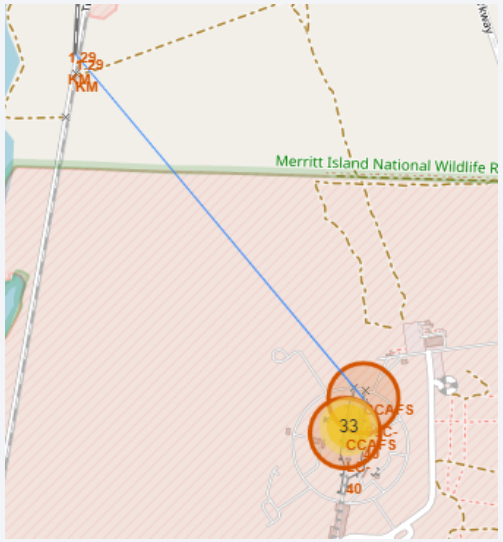


The successful launches are indicated by green markers while the unsuccessful ones are shown by red markers. It can be observed that KSC LC-39A has a relatively higher rate of successful launches.

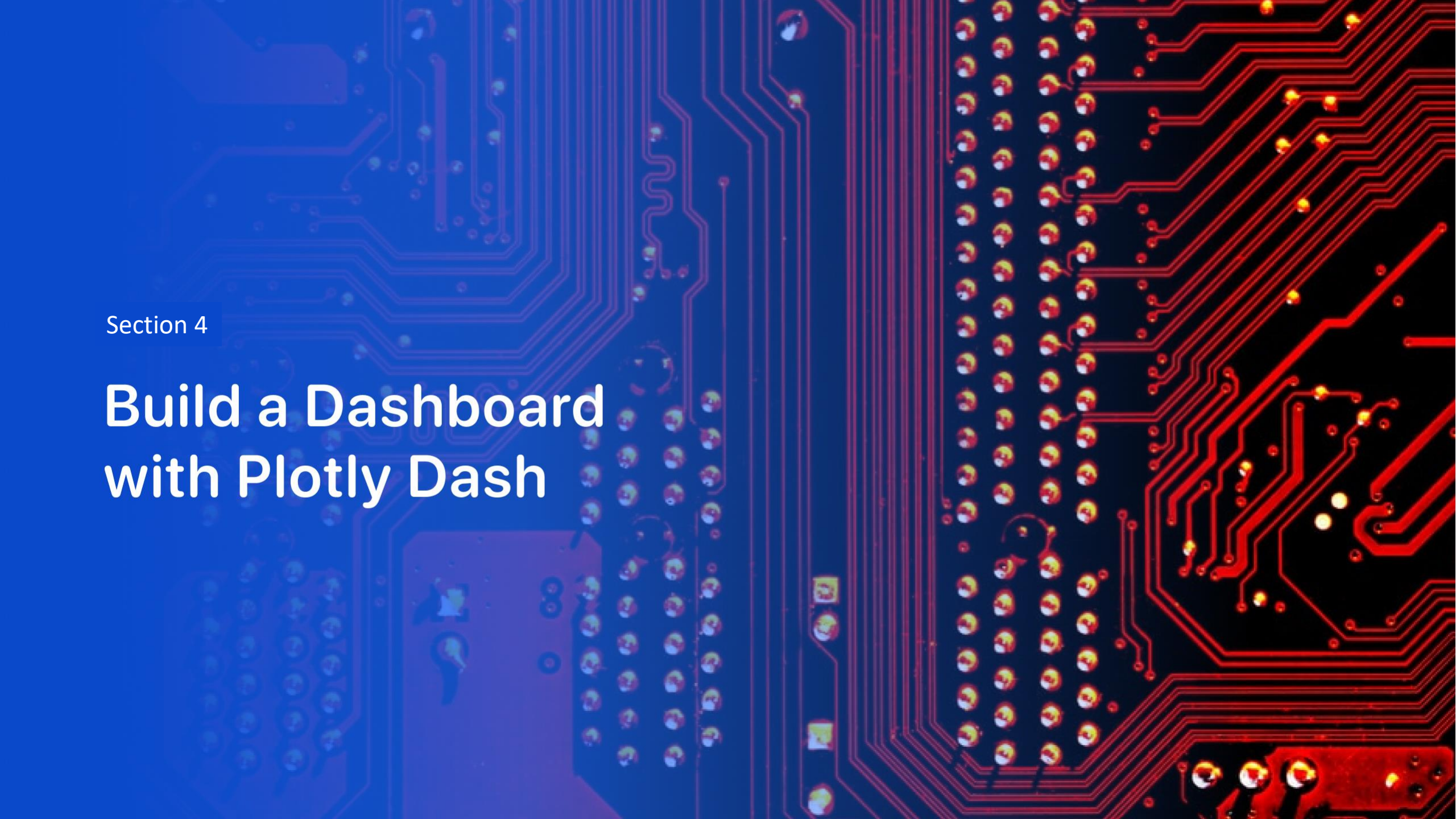
<Folium Map Screenshot 3>



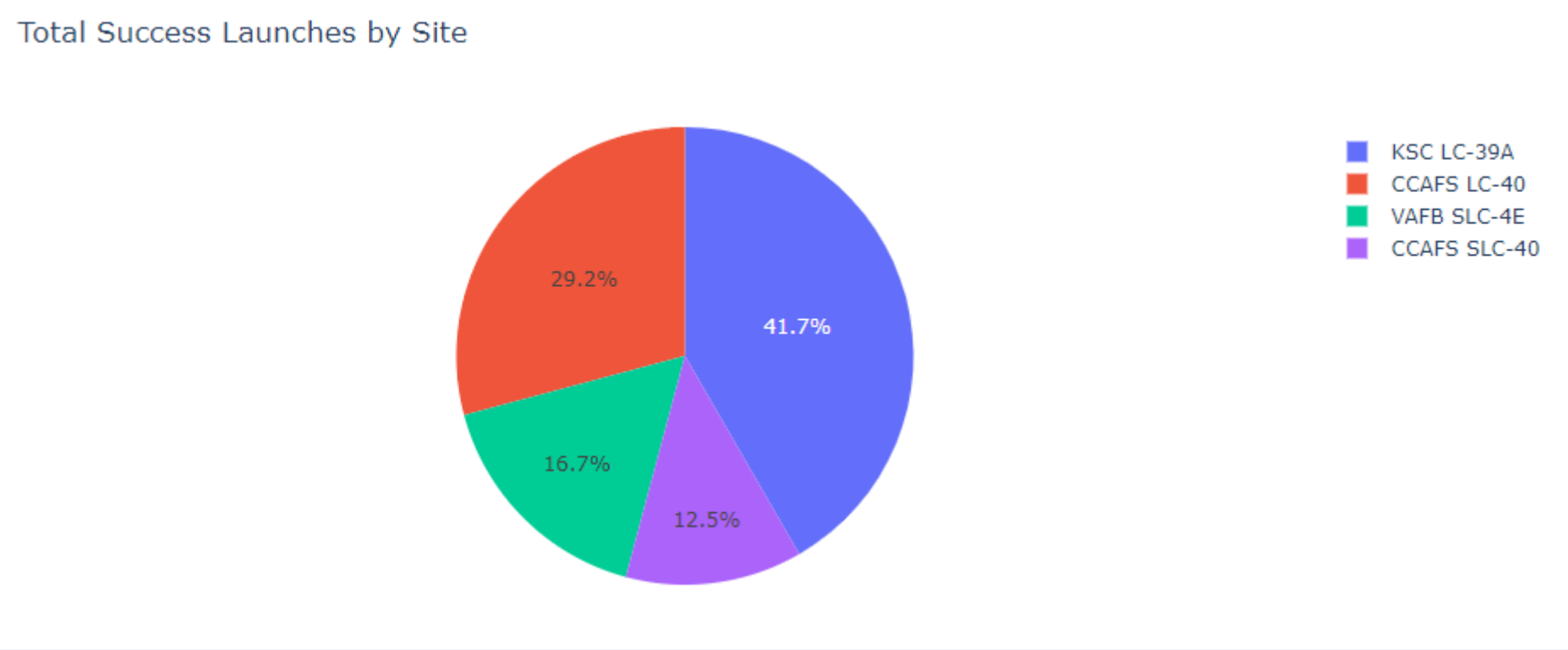




Is CCAFS SLC-40 in close proximity to railways ?
Yes
Is CCAFS SLC-40 in close proximity to highways ?
Yes
Is CCAFS SLC-40 in close proximity to coastline ?
Yes
Do CCAFS SLC-40 keeps certain distance away from cities ?
Yes

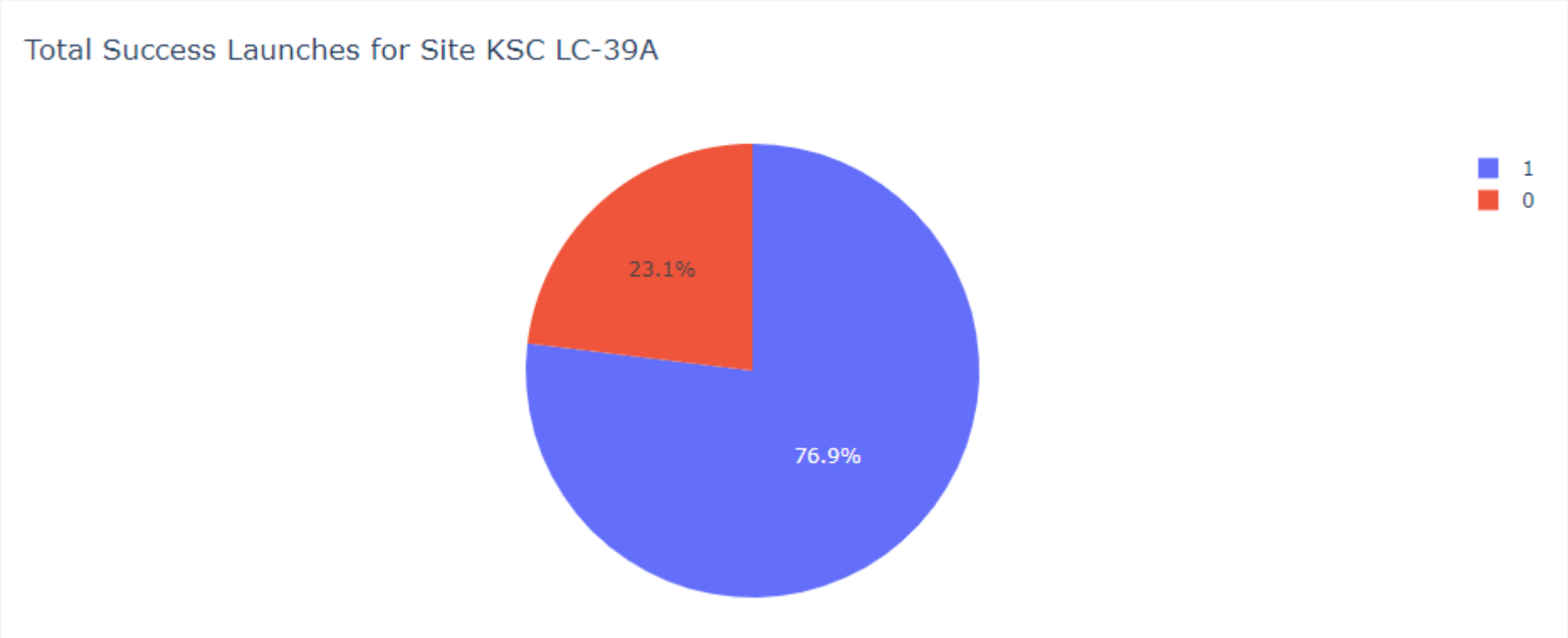

Section 4
Build a Dashboard with Plotly Dash

Success Percentage by Site



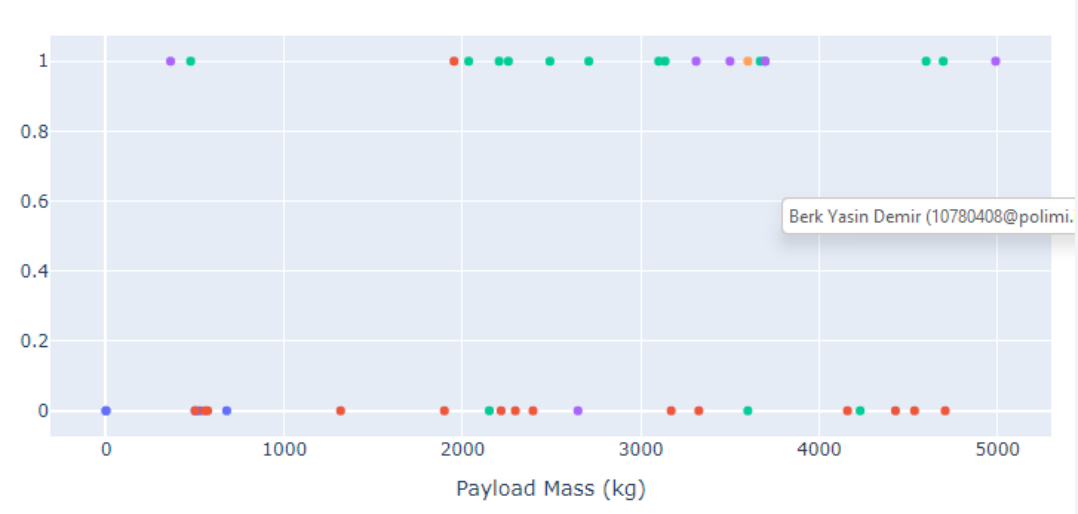
It can be said that KSC LC-39A has the best success rate of launches.

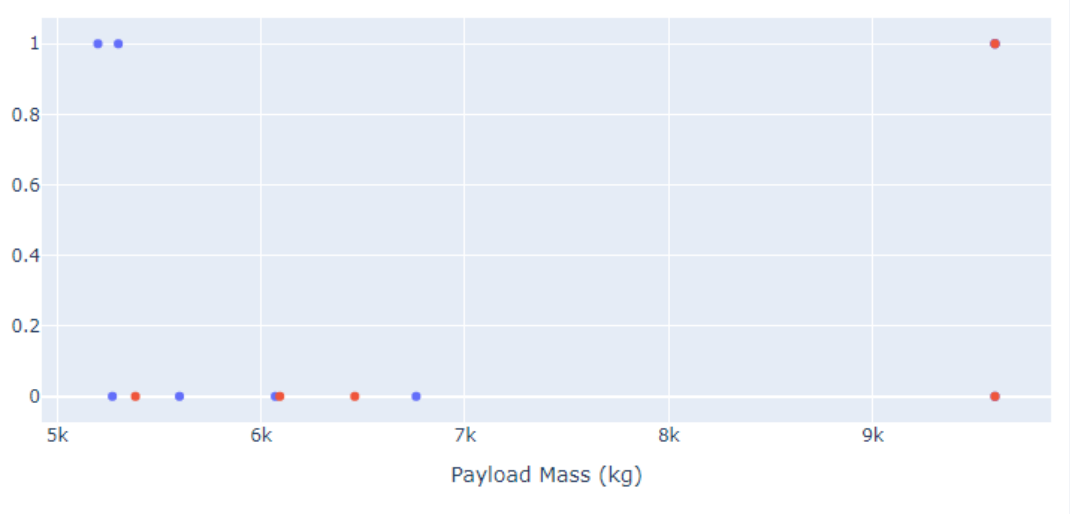
Success and Failuire Pie Chart for KSC LC-39A



It can be said that more than ¾ launches were successful in KSC LC-39A

Success Comparison with low & high weight



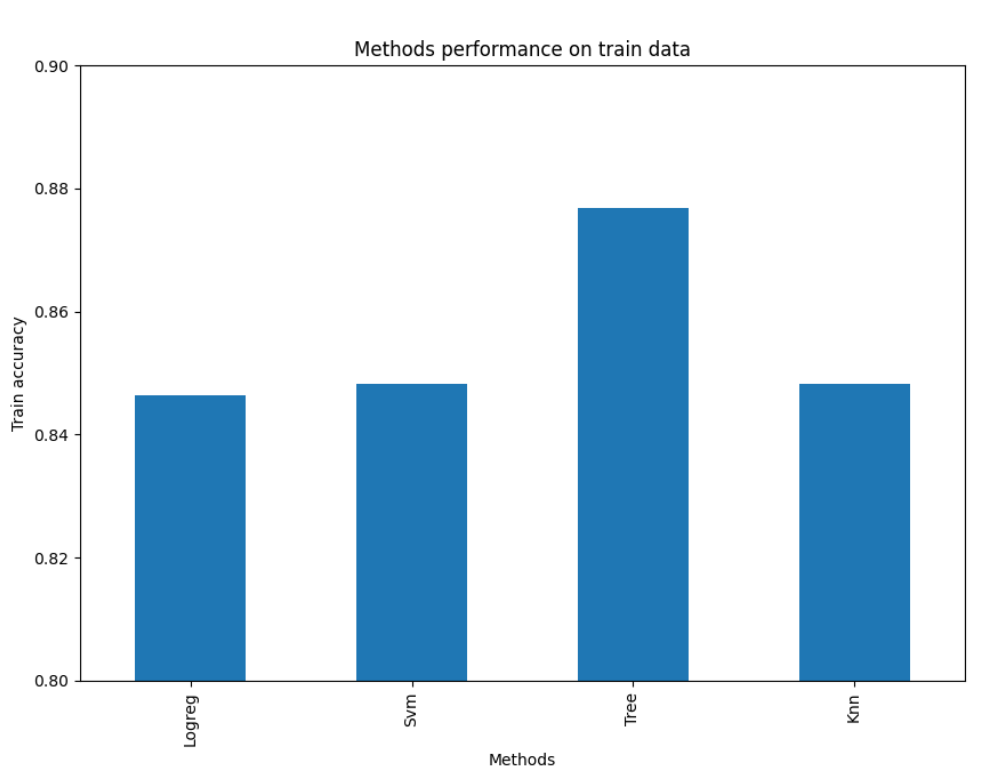


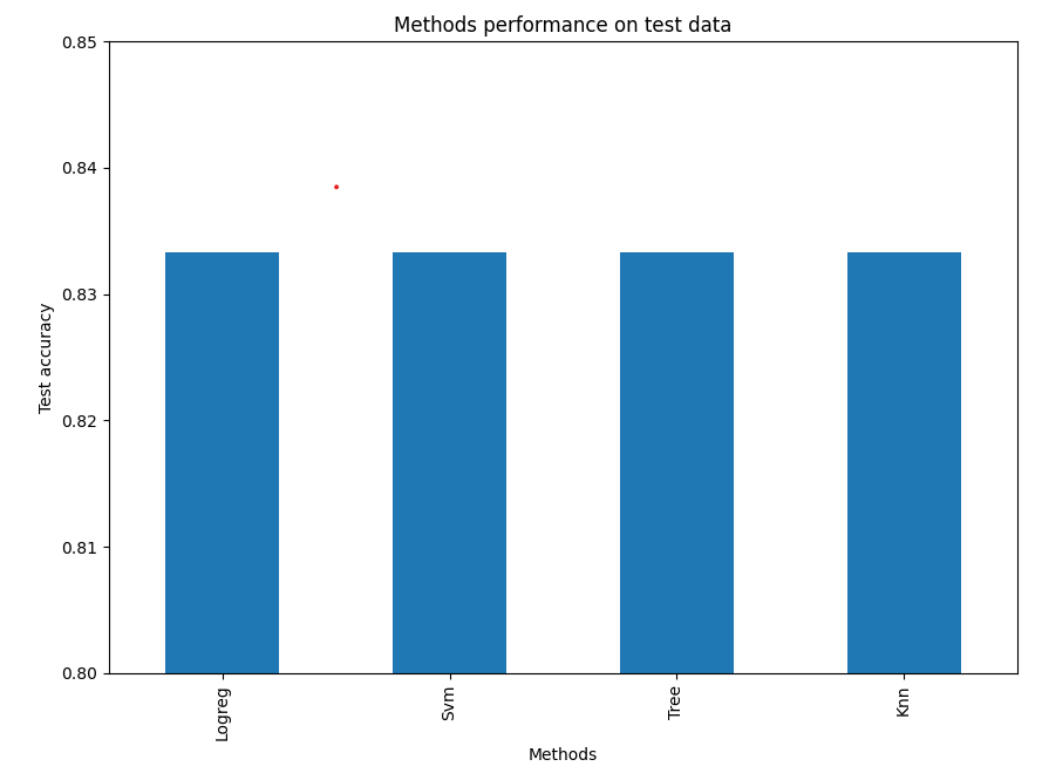
Payloads with lower weight have a higher probability of success compared to payloads with heavier weight.

Section 5

Predictive Analysis (Classification)

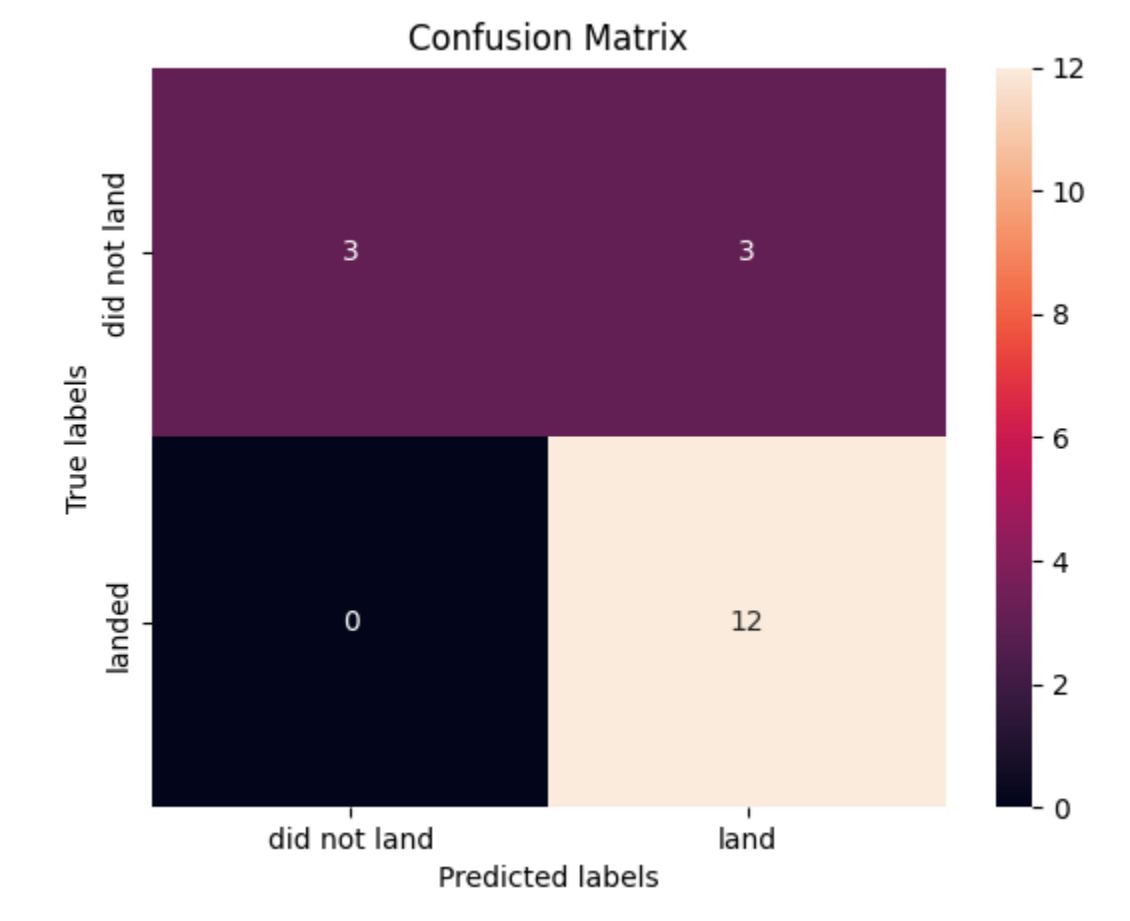
Classification Accuracy





It can be said that all the models performed same in the test set. All of them have a good accuracy score. But in the train set Tree performed better. Finally, it might be better to increase the dataset size to be sure that which model is performing better.

Confusion Matrix



The decision tree classifier's confusion matrix indicates that it can differentiate between various categories. However, the primary issue is false positives, which means that the classifier misidentifies unsuccessful landings as successful ones. This situation is also also same for other classifiers.

Conclusions

- • Launch site, orbit, and the number of previous launches all play a role in the success of a mission. The KSC LC-39A site has the highest success rate.
- • Orbits with the highest success rates are GEO, HEO, SSO, and ES-L1. Payload mass also affects the success of a mission, with some orbits requiring light or heavy payloads.
- • Low-weighted payloads (less than 4000kg) have a better success rate than heavy-weighted payloads.
- • The Decision Tree Algorithm was chosen as the best model for this dataset, as it had better train accuracy, even though the test accuracy was the same as other models used.
- • The success rate of SpaceX launches has increased since 2013, with a direct relationship to time, and is expected to continue improving in the future.
- • KSC LC-39A has the most successful launches of any site, with a success rate of 76.9%.
- • The SSO orbit has the highest success rate, with a success rate of 100% and more than one occurrence.

Thank you!