IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kavyashree V
16/10/2023

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

Introduction

SpaceX, founded by Elon Musk in 2002, has significantly advanced space exploration and transportation. The company's ambitious goals and innovative technologies have profoundly impacted the aerospace industry.

This project aims to create a machine-learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch

Problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variable and how it is affecting the outcome.

- Data Methodology

Section 1

Methodology

Methodology

In the subsequent slides we will describe

1. Data collection methodology:
    - Describe how data was collected
2. Perform data wrangling
    - Describe how data was processed
3. Perform exploratory data analysis (EDA) using visualization and SQL
4. Perform interactive visual analytics using Folium and Plotly Dash
5. Perform predictive analysis using classification models
    - How to build, tune, evaluate classification models

Data Collection Methodology

1. Web Scrapping

- Extract data from publicly available sources, such as news articles, press releases, or official documents.

2. API Integration

- Utilize application programming interfaces (APIs) provided by relevant platforms or organizations to retrieve data.

Data Cleansing & Validation

1. Data Cleaning and Validation:

- Clean and preprocess the collected data to remove duplicates, handle missing values, and ensure consistency.
- Validate the data to ensure accuracy and reliability

Data Collection – SpaceX API

```
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 number, and da
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 rocket booste
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 the list and
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 date leaving th
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_science_final_assignment/jupyter-labs-spacex-data-collection-api.ipynb at main · KavyashreeRai/Data_science_final_assignment (github.com)]

Get request for rocket launch data using API

Use Json normalize method to convert Json result to data frame

Performed data cleaning and filling the missing value

Data Collection - Scraping

```
# Create a BeautifulSoup object
response = requests.get(static_url)
```

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

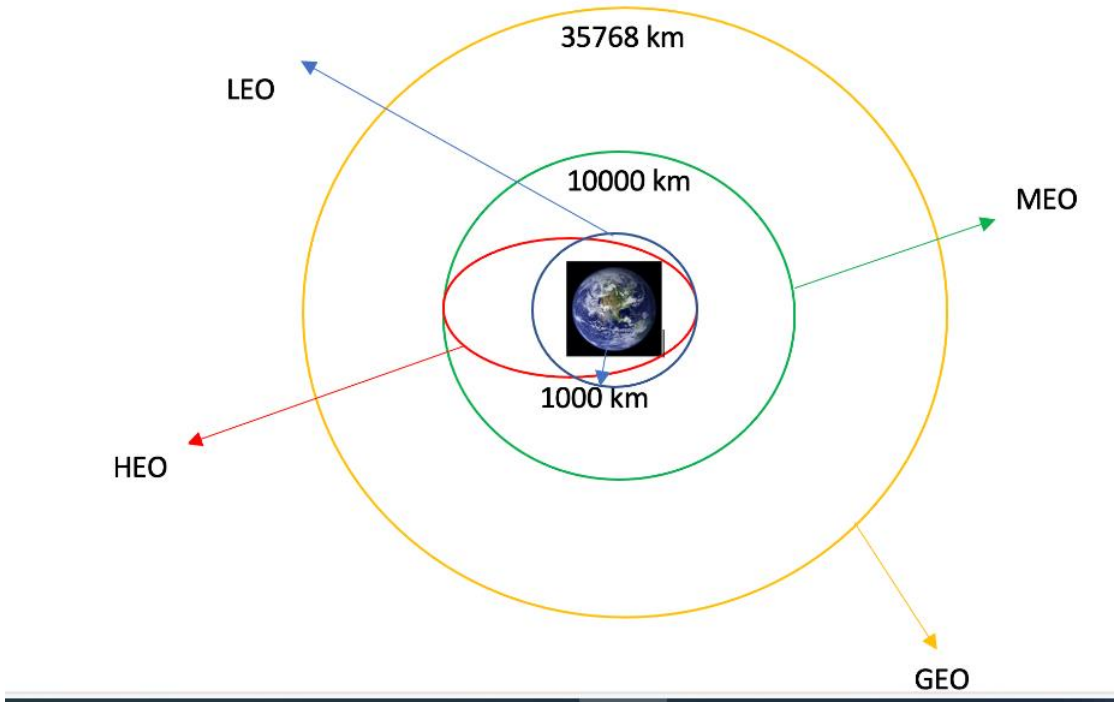
[Data_science_final_assignment/jupyter-labs-webscraping.ipynb at main · KavyashreeRai/Data_science_final_assignment (github.com)]

Request the Falcon9 Launch Wiki page from url

Create a Beautiful Soup from the HTML response

Extract all column/variable names from the HTML header

Data Wrangling

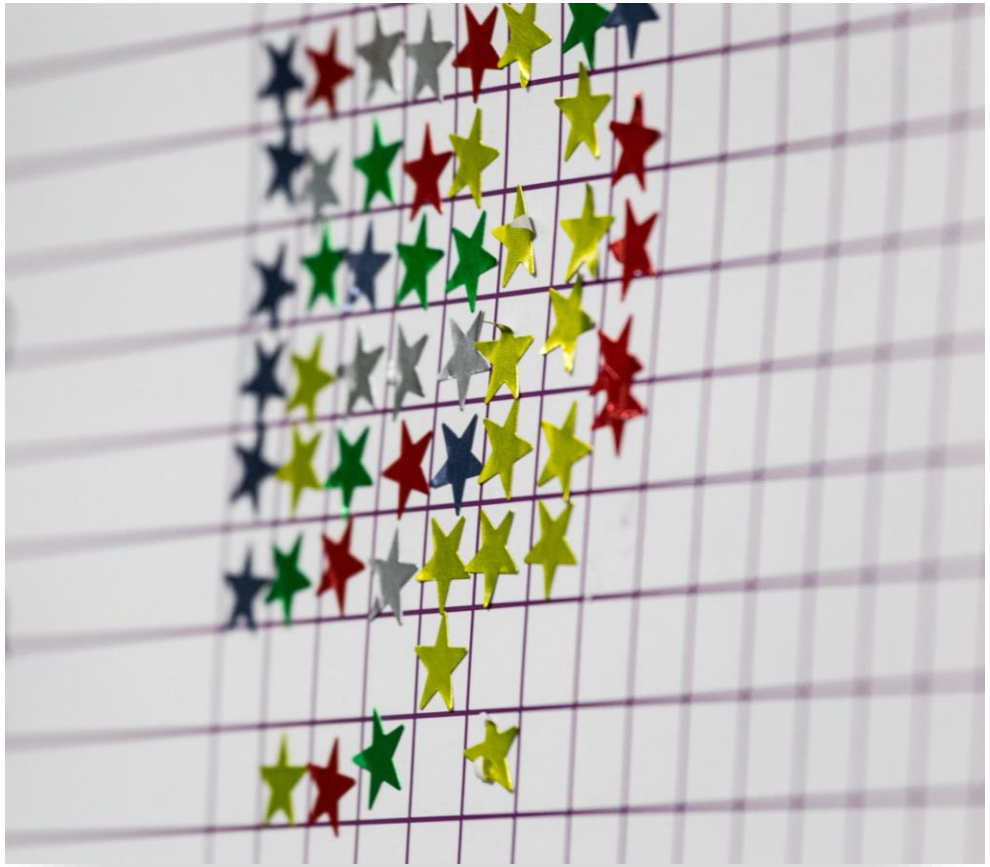


Data wrangling, also known as data munging, is the process of cleaning, transforming, and preparing raw data into a format suitable for analysis or further processing.

From the figure in the left, we will first calculate the number of launches on each site then calculate the number and occurrence of mission outcome per orbit type.

[Data_science_final_assignment/labs-jupyter-spacex-Data wrangling.ipynb at main · KavyashreeRai/Data_science_final_assignment (github.com)]

EDA with Data Visualization



We first started by using a scatter graph to find the relationship between the attributes such as :

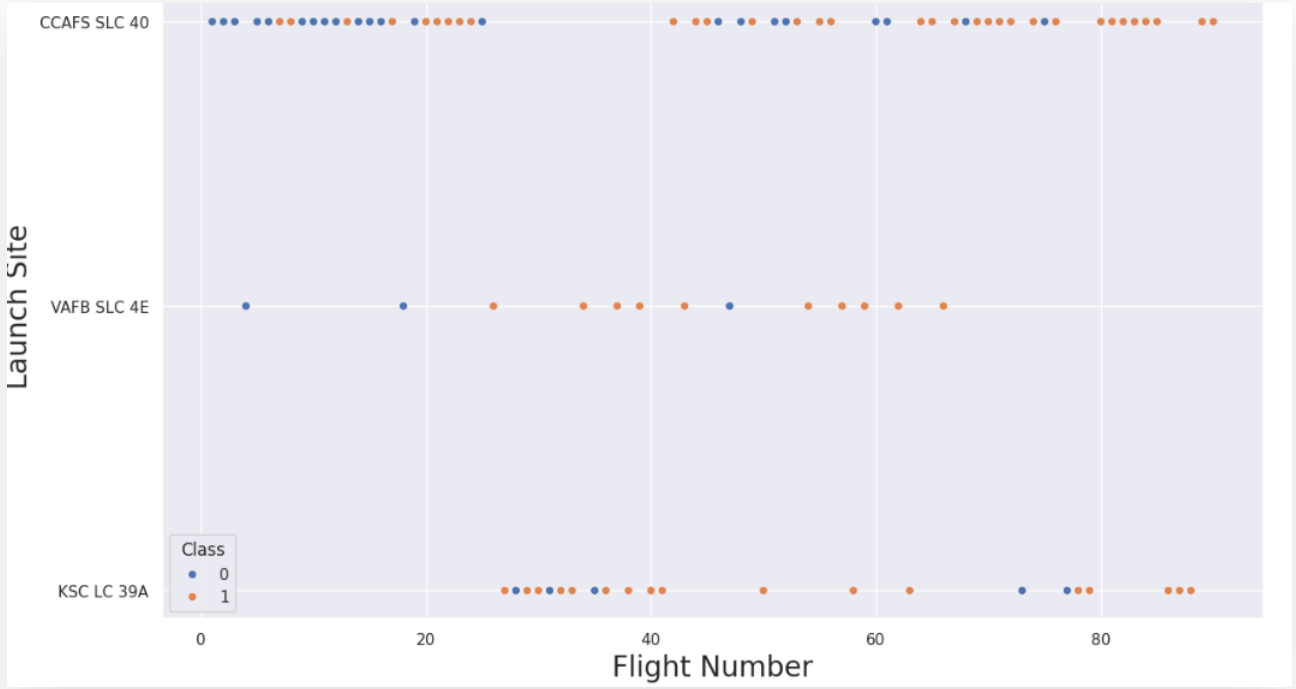
- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type

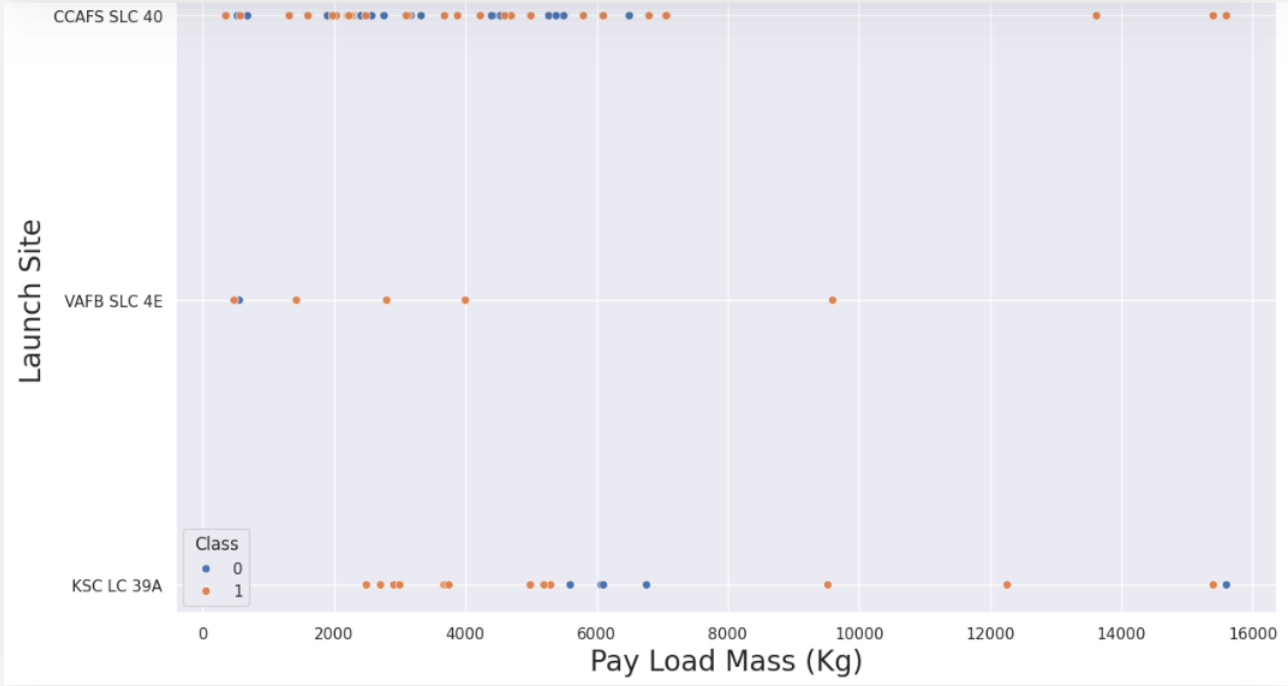
[Data_science_final_assignment/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb at main · KavyashreeRai/Data_science_final_assignment (github.com)]

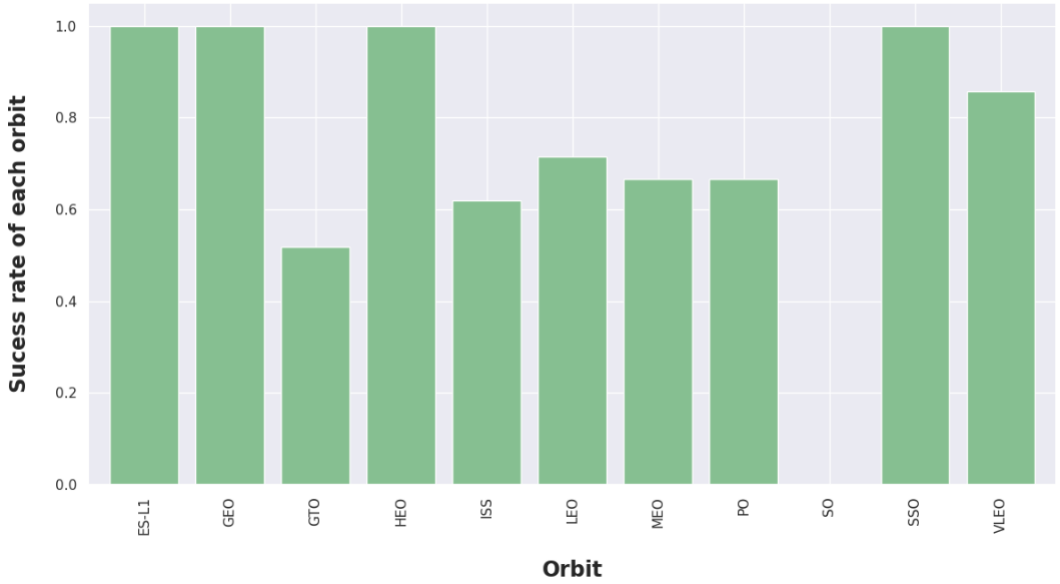
EDA with Data Visualization

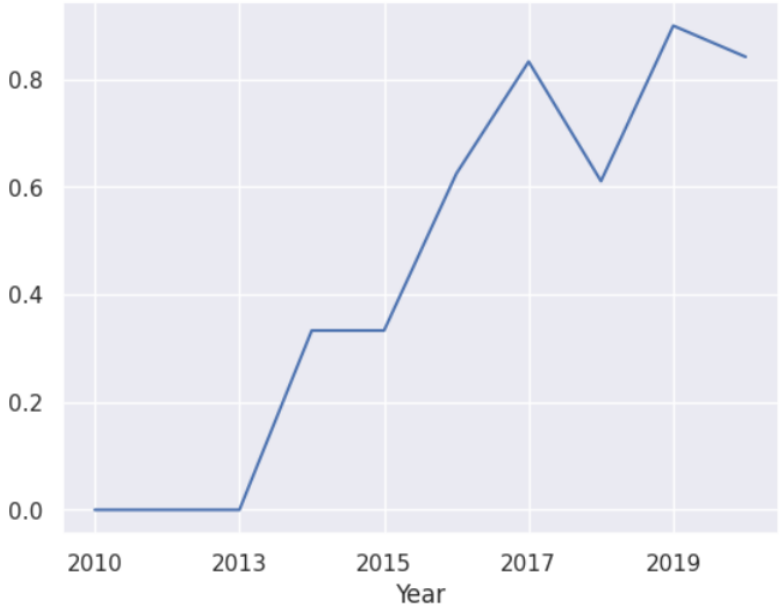
Scatter plots show the dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes.

https://github.com/KavyashreeRai/Data_science_final_assignment/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb









EDA with Data Visualization

Bar graphs is one of the easiest way to interpret the relationship between the attributes. In this case, we will use the bar graph to determine which orbits have the highest probability of success.

We then use the line graph to show a trends or pattern of the attribute over time which in this case, is used to see the launch success trending by the year.

Data_science_final_assignment/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb at main · KavyashreeRai/Data_science_final_assignment (github.com)

EDA with SQL

- 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 the ground pad was achieved.
- List the names of the boosters that have success in drone ships and have a payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failed mission outcomes.
- Listing the names of the booster_versions that have carried the maximum payload mass.
- Listing the failed outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- Rank the count of landing outcomes or success between the dates 2010-06-04 and 2017-03-20, in descending order.

Data_science_final_assignment/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · KavyashreeRai/Data_science_final_assignment (github.com)

Build an Interactive Map with Folium

Based on the graph we are using the Folium library, which is a Python library used for creating interactive maps

Markers:

- Markers are used to indicate specific points of interest on the map. They are typically represented by icons or symbols

Circles:

- Circles on a map represent a circular area, often used to highlight regions of interest or to indicate a certain radius around a point. These circles can be used to show areas like coverage zones or proximity.

Lines:

- Lines are used to represent paths, routes, or connections between points on a map. They can be used to show the trajectory of a journey or to indicate boundaries or routes.

Data_science_final_assignment/jupyter_launch_site_location.jupyterlite.ipynb at main· KavyashreeRai/Data_science_final_assignment (github.com)

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash which allows the user to play around with the data as they need.
- We plotted pie charts showing the total launches by certain sites.

Data_science_final_assignment/Dashboard.JPG at main · KavyashreeRai/Data_science_final_assignment (github.com)

Predictive Analysis (Classification)

Building the Model

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

Evaluating the Model

- Check the accuracy of each model
- Tune hyperparameters for each type of algorithm.
- Plot the confusion matrix.

Improving the Model

- Use Feature Engineering and Algorithm Tuning

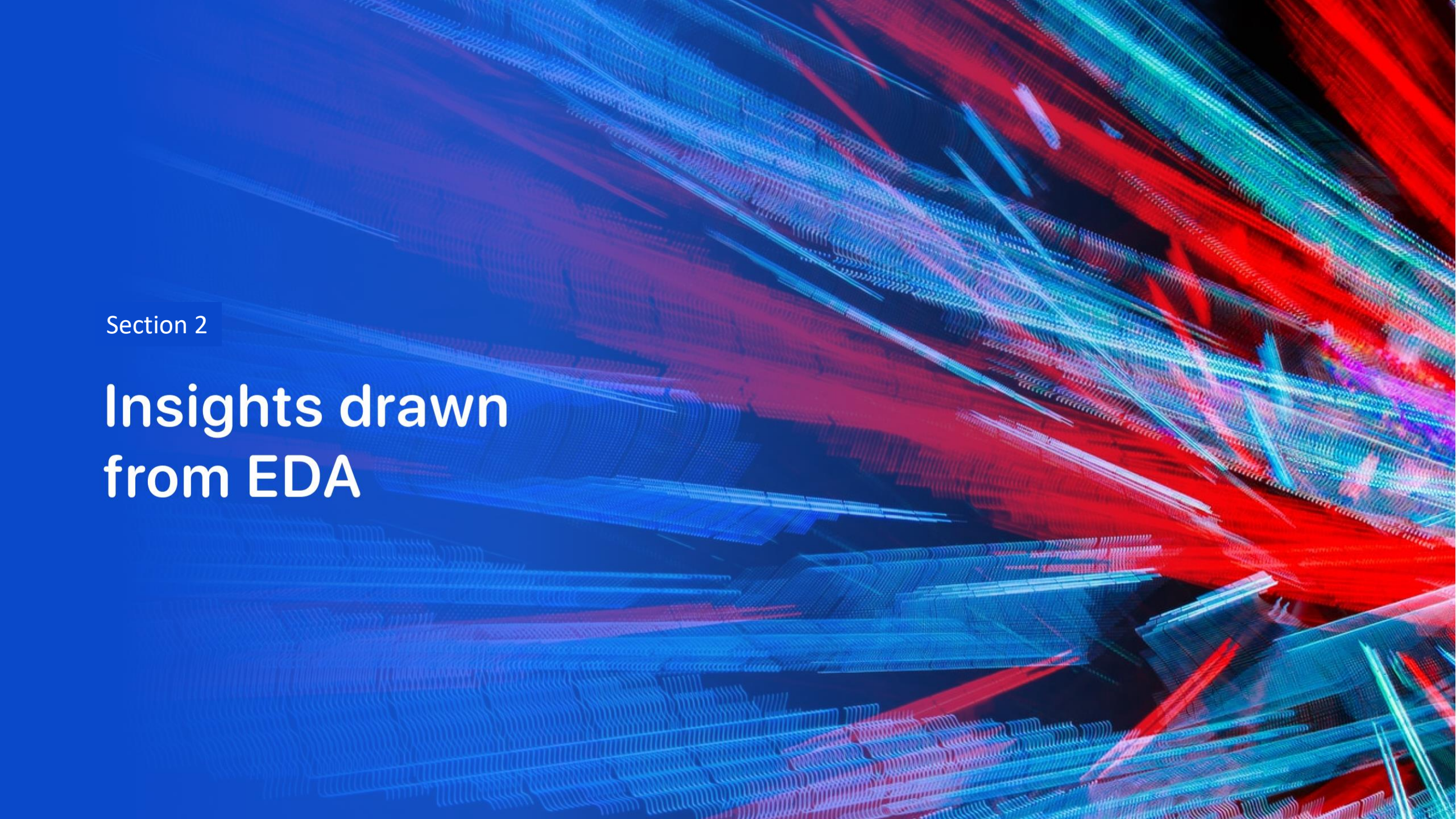
Find the Best Model

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

KavyashreeRai/Data_science_final_assignment (github.com)

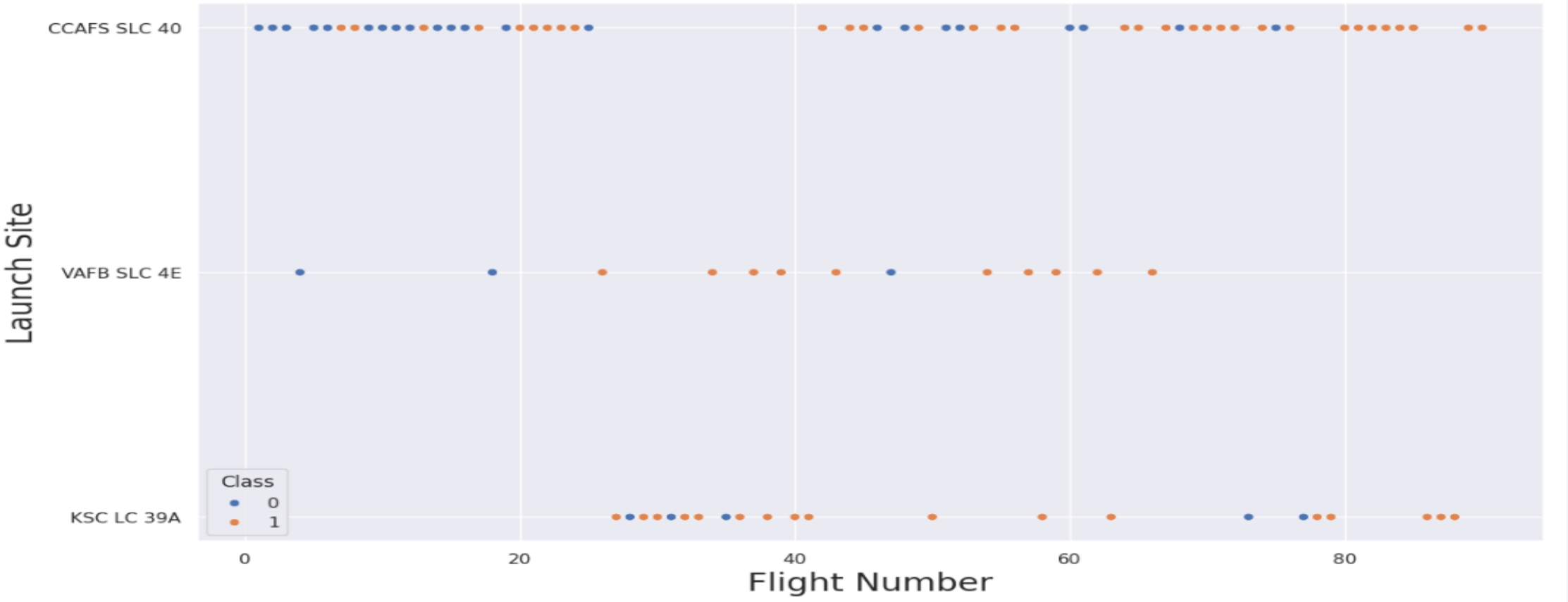
Results

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

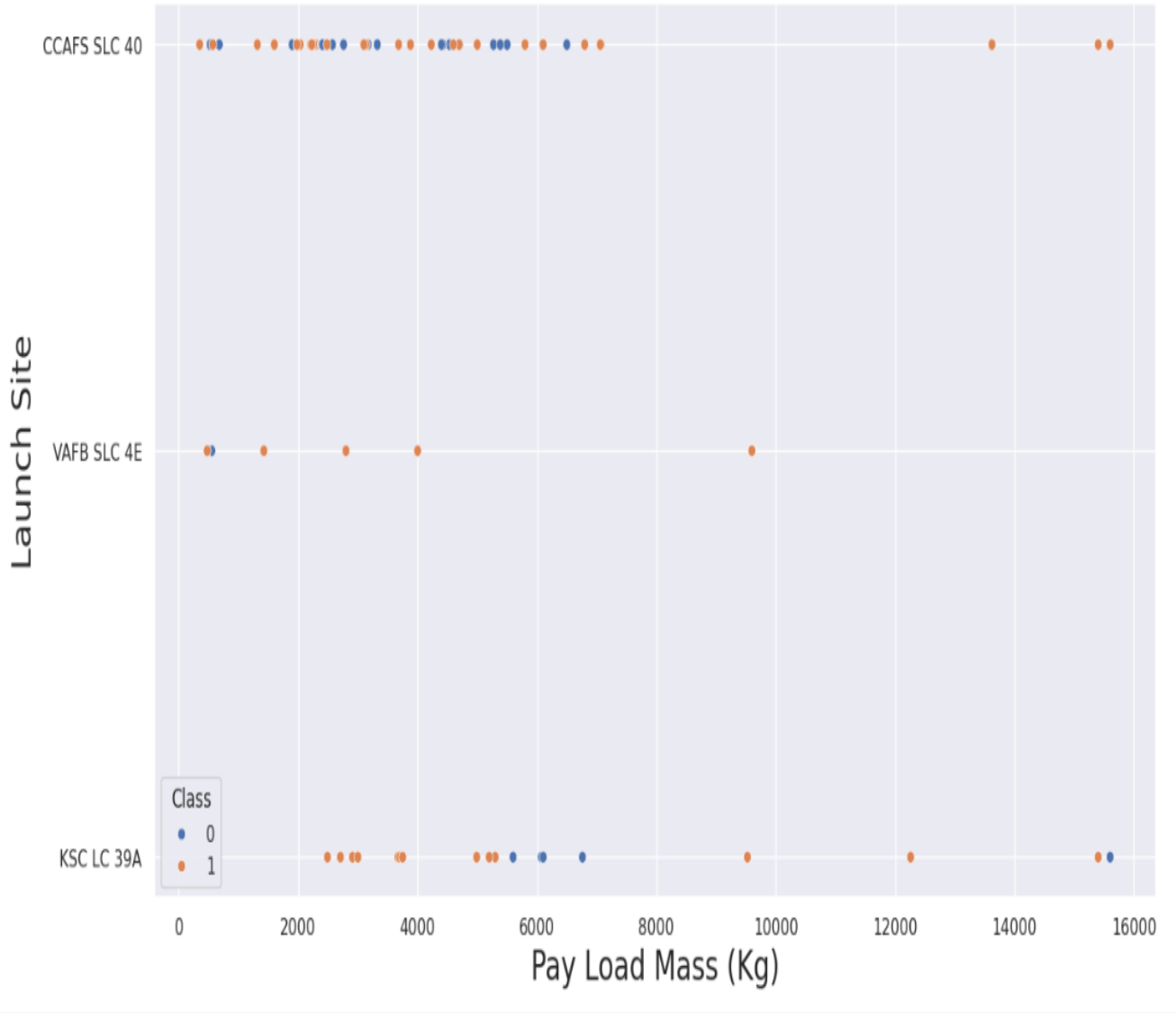

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

This scatter plot indicates a positive correlation between the size of the launch site and the success rate of flights, suggesting that larger launch facilities tend to exhibit higher success rates. However, it's noteworthy that CCAFS SLC40, while still exhibiting a positive trend, demonstrates a less pronounced pattern in comparison to the other launch sites.



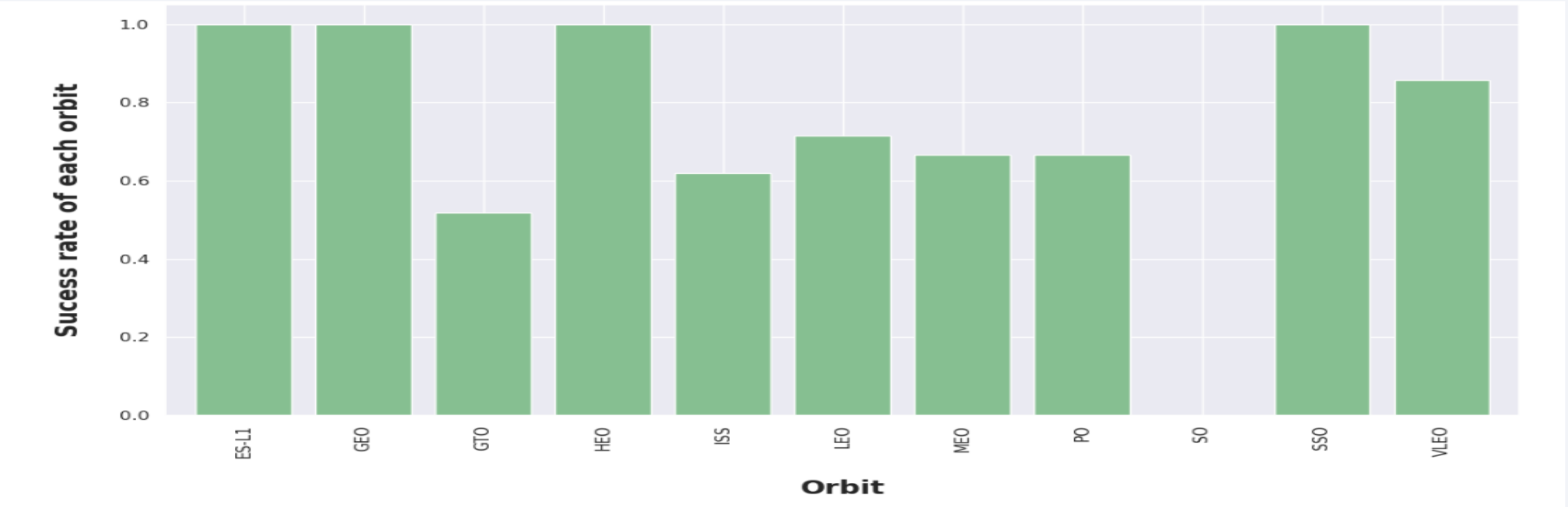
Payload vs. Launch Site



The scatter plot reveals a notable trend: when the payload mass exceeds 7000kg, there is a substantial increase in the probability of success. However, it's important to note that there is no distinct pattern indicating a strong dependency between the launch site and the payload mass for determining the success rate.

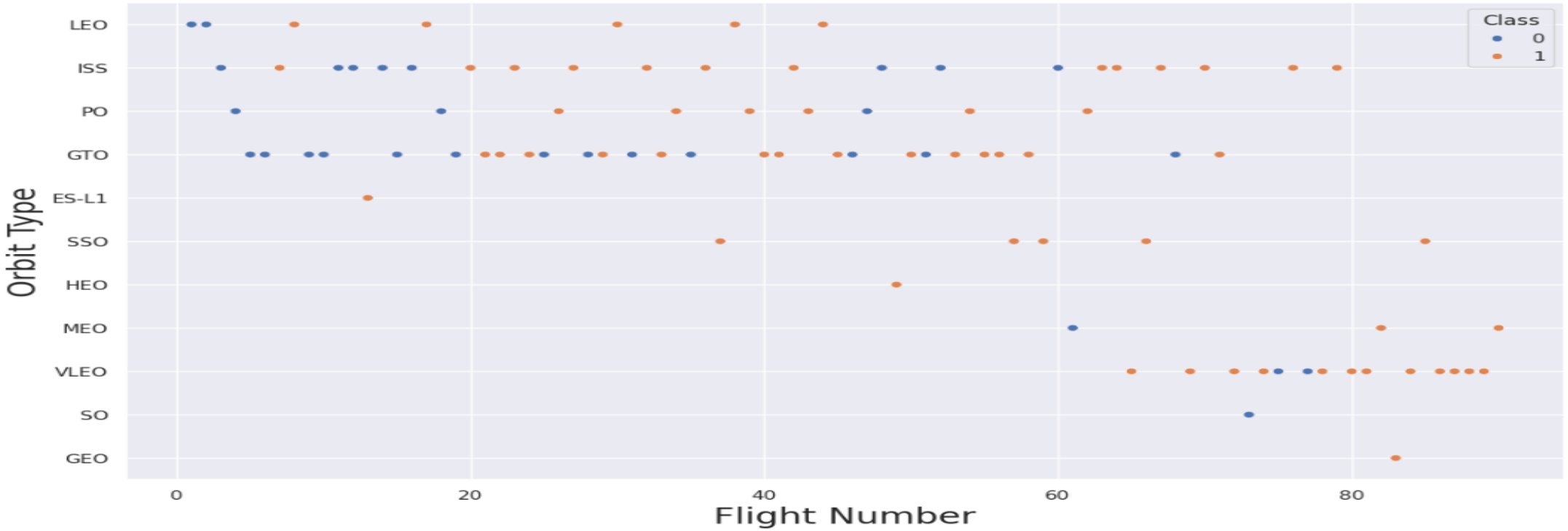
Success Rate vs. Orbit Type

This figure illustrates a discernible connection between the choice of orbit and the corresponding landing outcomes. Orbits like SSO, HEO, GEO, and ES-L1 exhibit a 100% success rate, while SO orbit shows a 0% success rate. However, upon closer examination, it's crucial to note that some of these orbits only have one occurrence, as seen in GEO, SO, HEO, and ES-L1. This highlights the need for a more extensive dataset to discern any definitive patterns or trends before drawing conclusive assessments.

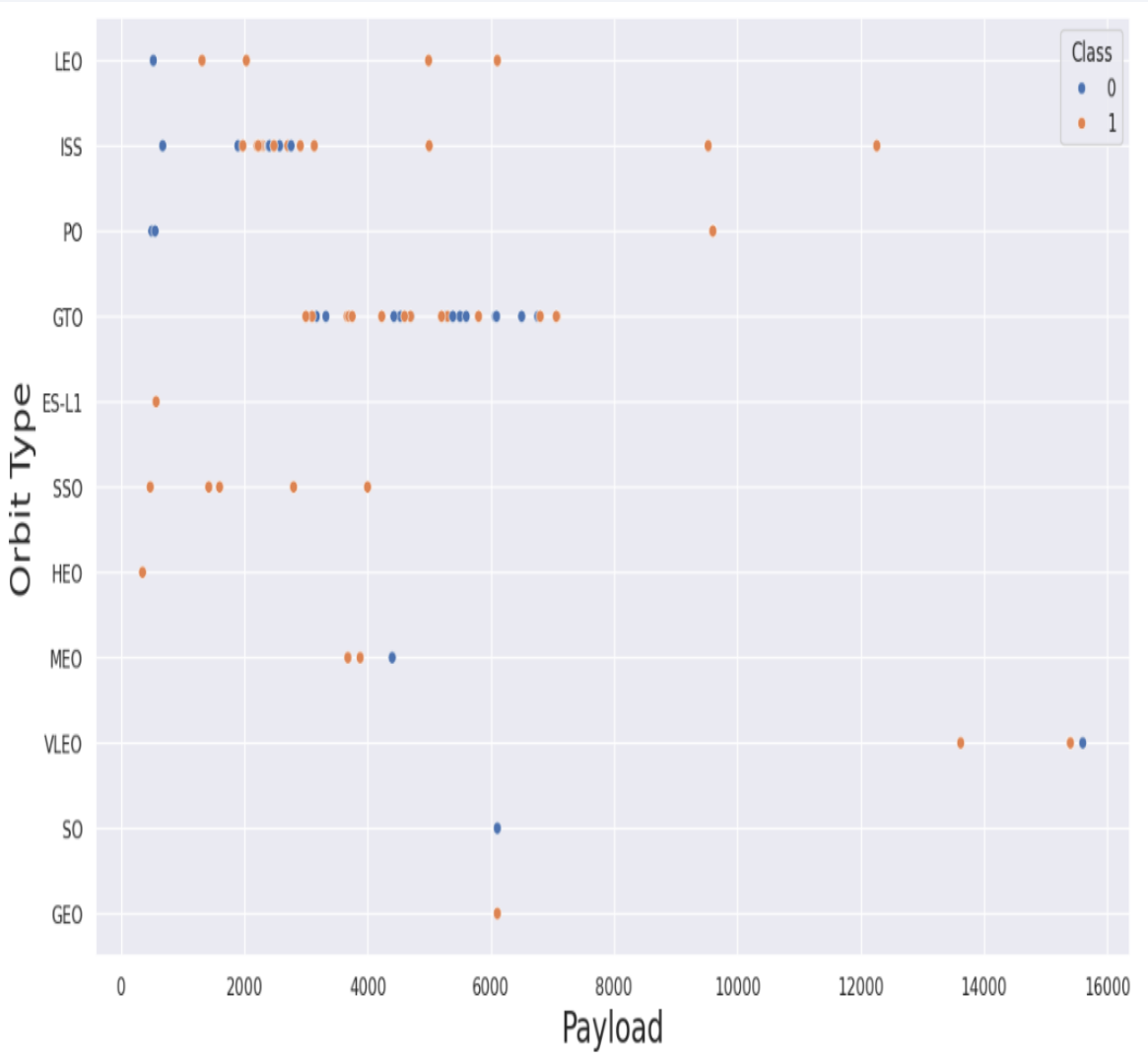


Flight Number vs. Orbit Type

- This scatter plot reveals a notable trend: in general, as the flight number increases within each orbit, the success rate tends to rise, particularly evident in the LEO orbit. However, this pattern doesn't hold true for GTO orbit, where no discernible relationship between these attributes is observed.
- It's important to note that orbits with only one occurrence should be excluded from the statement. These orbits require a more extensive dataset for a reliable assessment.



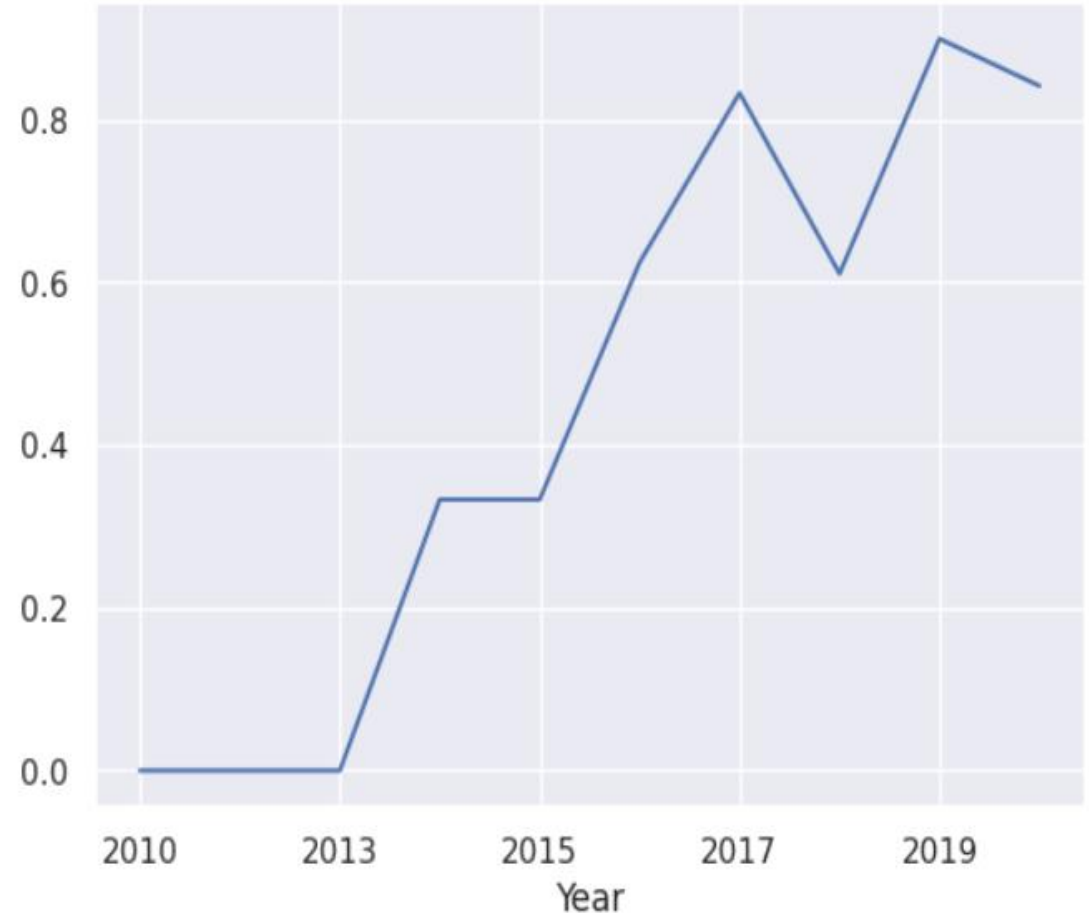
Payload vs. Orbit Type



A heavier payload demonstrates a positive impact on the success rates for LEO, ISS, and P0 orbits. Conversely, it exerts a negative influence on the success rates for MEO and VLEO orbits. In the case of GTO orbit, there appears to be no discernible relationship between payload mass and success rates.

Meanwhile, it's worth noting that for SO, GEO, and HEO orbits, additional data is needed to confidently identify any underlying patterns or trends.

Launch Success Yearly Trend



The figures unmistakably illustrate a consistent upward trend from 2013 to 2020. If this trend persists into the coming years, we can anticipate a steady rise in the success rate, eventually reaching a remarkable 100% success rate.

All Launch Site Names

Names of the unique launch sites

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

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

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

Launch Site Names Begin with 'CCA'

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

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

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

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

Find 5 records where launch sites begin with `CCA`

Total Payload Mass

```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

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

TOTAL_PAYLOAD
111268

Average Payload Mass by F9 v1.1

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

```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

AVG_PAYLOAD
2928.4

First Successful Ground Landing Date

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

Hint:Use min function

```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

FIRST_SUCCESS_GP
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome = 'Succ
```

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

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

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

```
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

2015 Launch Records

```
%sql SELECT DATE, BOOSTER_VERSION, LAUNCH_SITE,
LANDING__OUTCOME FROM SPACEXTBL WHERE
LANDING__OUTCOME = 'Failure (drone ship)' AND
YEAR(DATE) = 2015
```

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

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

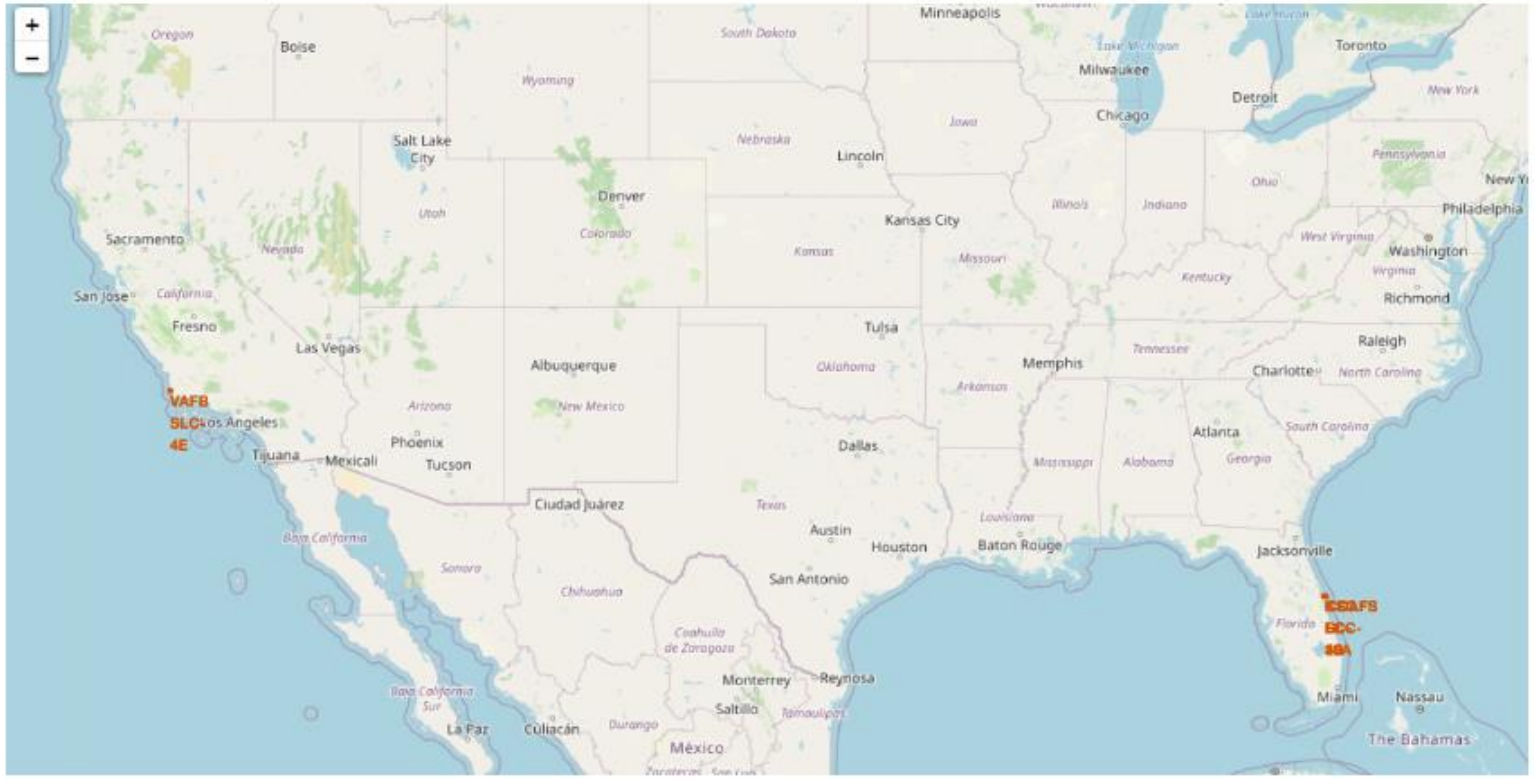
We conducted a data query to select 'Landing Outcomes' and retrieved the count of occurrences for each landing outcome. Using a WHERE clause, we filtered for landing outcomes between June 4, 2010, and March 20, 2010. We then applied a GROUP BY clause to group the landing outcomes, followed by an ORDER BY clause to arrange the grouped outcomes in descending order.

```
%sql SELECT LANDING__OUTCOME, COUNT(*) AS COUNT_LAUNCHES
FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME ORDER BY COUNT_LAUNCHES DESC;
```

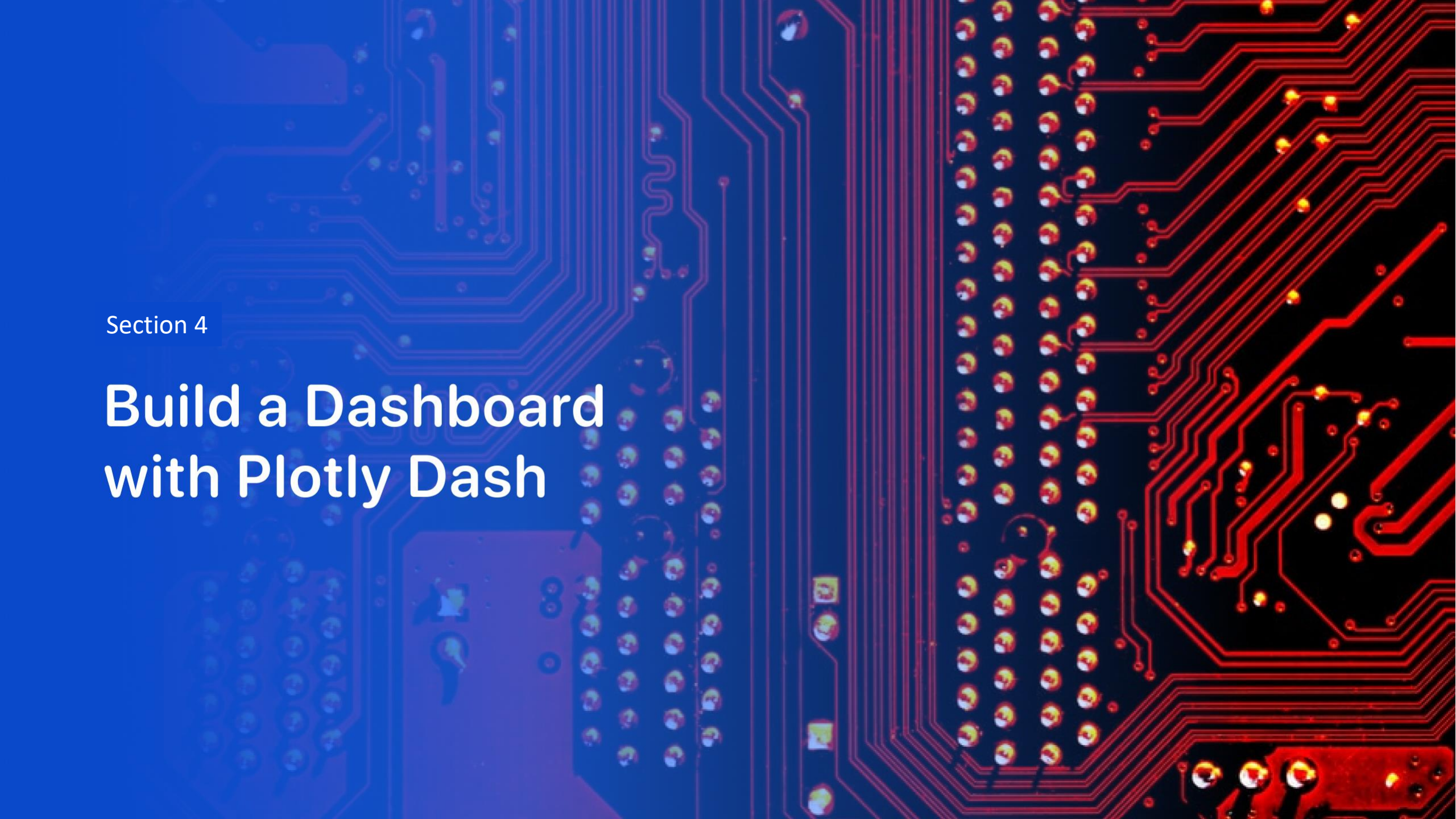
Section 3

Launch Sites Proximities Analysis

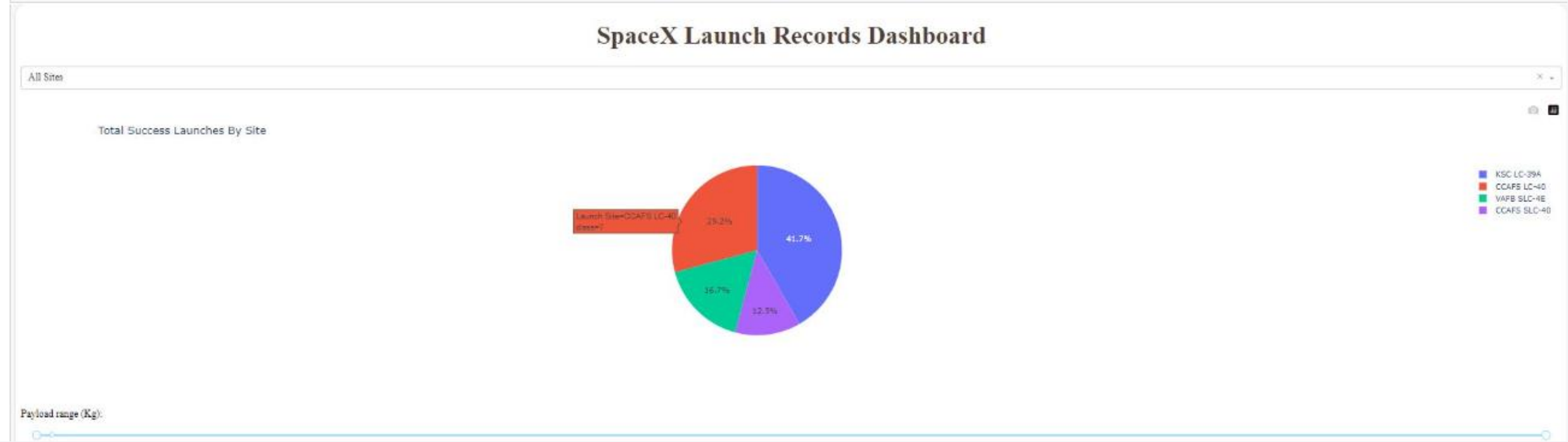
Location of all the Launch Sites



We can see that all the SpaceX launch sites are located inside the United States


Section 4
Build a Dashboard with Plotly Dash

Success percentage by each site



Success percentage by each site - Contd

Section 5

Predictive Analysis (Classification)

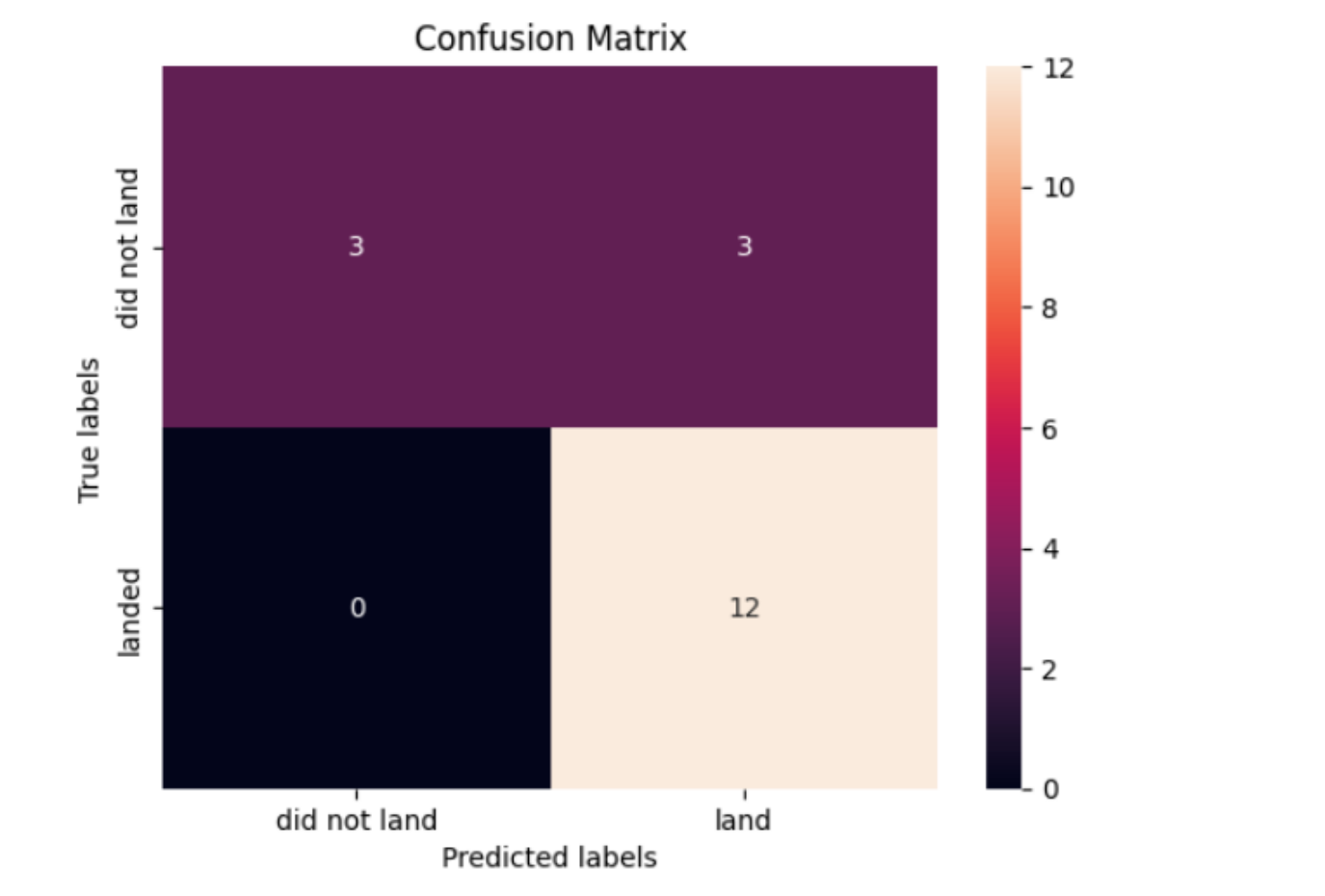
Classification Accuracy

Calculate the accuracy on the test data using the method `score`:

```
print("test set accuracy :",svm_cv.score(X_test, Y_test))
```

```
test set accuracy : 0.8333333333333334
```

Confusion Matrix



The confusion matrix for the decision tree classifier reveals its ability to effectively differentiate between various classes. However, a notable concern arises from the occurrence of false positives, where unsuccessful landings are erroneously classified as successful. This suggests a need for further refinement in the classifier's performance, particularly in reducing false positives.

Conclusions

- The Tree Classifier Algorithm emerges as the optimal Machine Learning approach for effectively classifying this dataset.
- Interestingly, the low-weighted payloads (defined as 4000kg and below) demonstrate superior performance compared to their heavier counterparts.
- Since 2013, there has been a noticeable upward trend in the success rate of SpaceX launches. This trend, which exhibits a direct proportionality with time, is expected to culminate in perfected launches in the future
- Among all launch sites, KSC LC-39A stands out with the highest number of successful launches
- The SSO orbit boasts the highest success rate, achieving a perfect 100% success rate and having more than one successful occurrence.

Thank you!