IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mehar Ali

31-03-2023

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

- **Project background and context**

  The SpaceX was founded in 2002 to revolutionize space technology, with the ultimate goal of enabling people to live on other planets SpaceX has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. One reason behind SpaceX's success is that the rocket launches are relatively inexpensive thanks to the reusability of the first stage. As a data scientist objective of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. Determine the price of each launch which depends on whether the first stage would land successfully to be reused.

- **Problems you want to find answers**

  Identifying all factors that influence the landing outcome and relationship between them and also estimation of condition how we can increase the of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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 is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. Data collection method was carried out by following two methods.

- 1- REST API we get request for data of launch of rocket. Then, we decoded the response content as Json and turn it into a pandas dataframe using json_normalize(). After the cleaning of the data we checked for missing values and fill with appropriate need.

- 2- Web scrapping we use request .get() to collect data of falcon9 launch using method the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis.

Data Collection – SpaceX API

1-For request to get data of rocket launch using API data collection method

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

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

2- Conversion of jason result into panda datafram with the help jason normalize method

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

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

3- Data cleaning with relative feature and handling of missing values

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

From :
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/Spacex%20data%20collection%20API.ipynb

Data Collection - Scraping

1- Performed requests.get() method with the provided static_url assign the response to a object for data collection

2-For request to get data from HTML of rocket launch we used BeautifulSoup method

3-From HTML header in order to extract all column and variables names we used the piece of code

From:
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/Data%20collection%20by%20Webscraping.ipynb

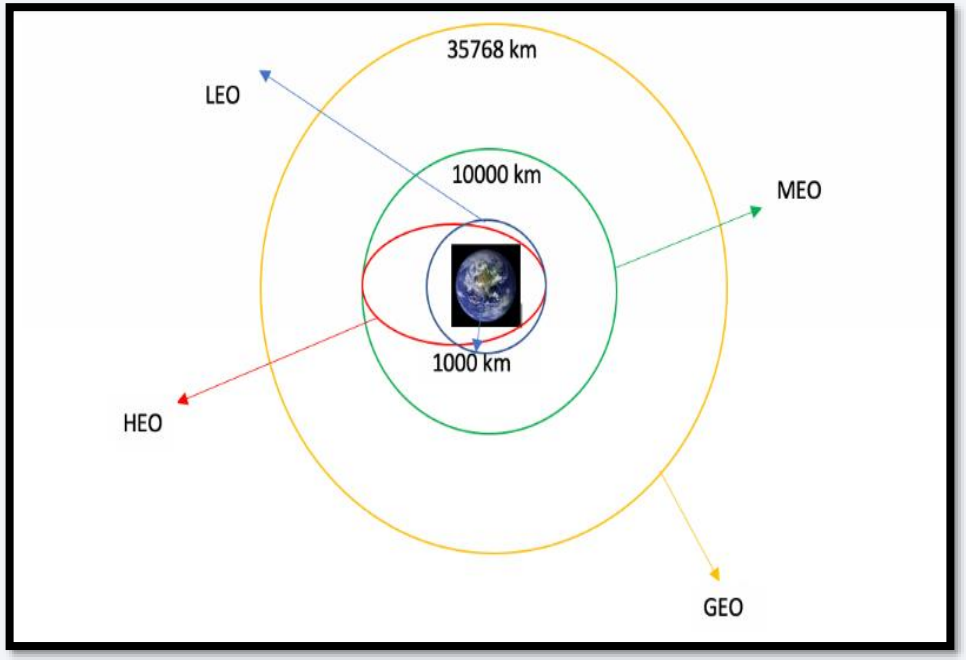
First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# 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 text content
soup = BeautifulSoup(data,'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
```

Data Wrangling

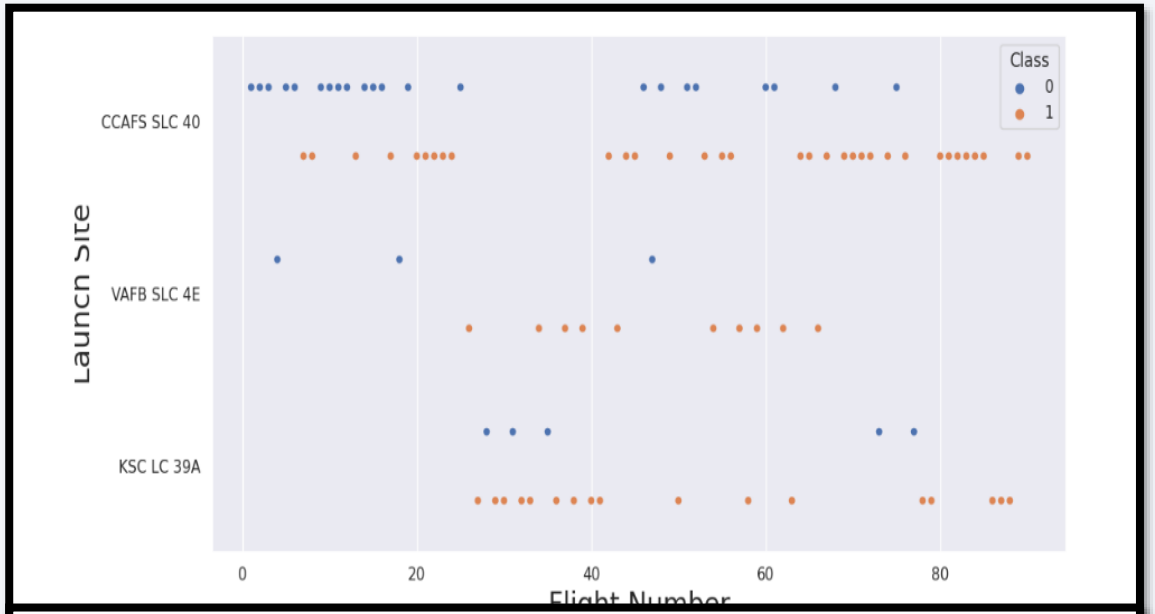


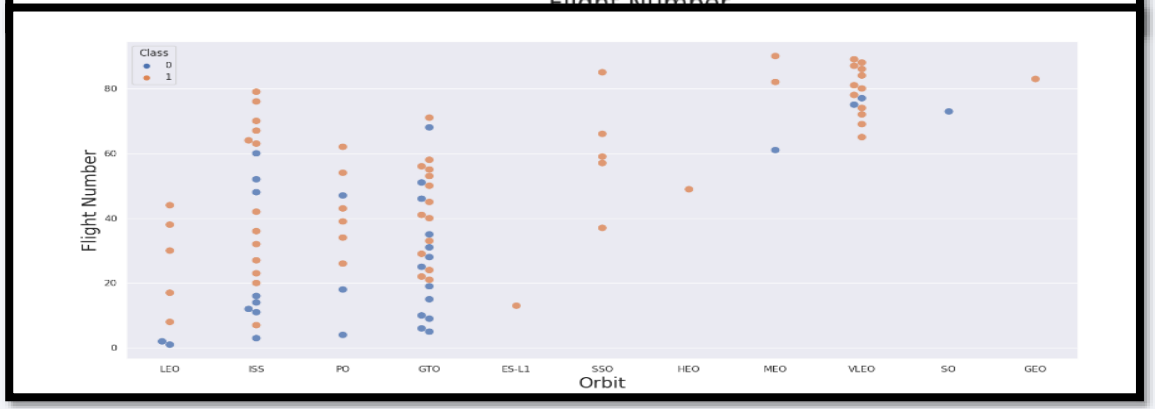
Data wrangling is the process of removing errors and combining complex data sets to make them more accessible and easier to analyze. In lab work we calculated number of launches on each side and the probability of mission outcomes . For the data analysis and visualization and for prediction by using machine learning techniques we exported csv files and labeled the landing outcomes in a appropriate way.

From:
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/Spacex%20Data%20wrangling.ipynb

EDA with Data Visualization

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods . In EDA with Data Visualization Lab we plated different scatter graphs including Payload and Flight Number, Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit Type and Payload and Orbit Type and also we find the relationship between these attributes





From:
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

In EDA with SQL Lb we performed following SQL queries

- 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.

From:
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/EDA%20With%20SQL.ipynb

Build an Interactive Map with Folium

In Build an Interactive Map with Folium Lab we have visualize the launch data into an interactive map and also we have taken 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 tovarious landmark to determination of closeness of the launch sites with railways, highways and coastlines and closeness of the launch sites with nearby cities.

From:
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

In Build a Dashboard with Plotly Dash lab 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 then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version. We have Build a Dashboard with plotly dash on jupyter notebook with uploading csv file in this environment.

From:
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/Interactive%20Dashboard%20with%20Ploty%20Dash%20Lab.ipynb

Predictive Analysis (Classification)

Building the Mode

In Model building section we load the dataset into NumPy array and Pandas to transform the data after completion this process we split into training and test datasets. After splitting data we opted which type of machine learning would be best, after this process set the parameters and algorithms to Grid SearchCV and fit it to dataset.

Evaluating the Model

In the Model evaluation process we checked the accuracy for each model and then tuned hyperparameters for each type of algorithms. After the setting the hyperperparameters plotted the confusion matrix.

Finding the Best Model

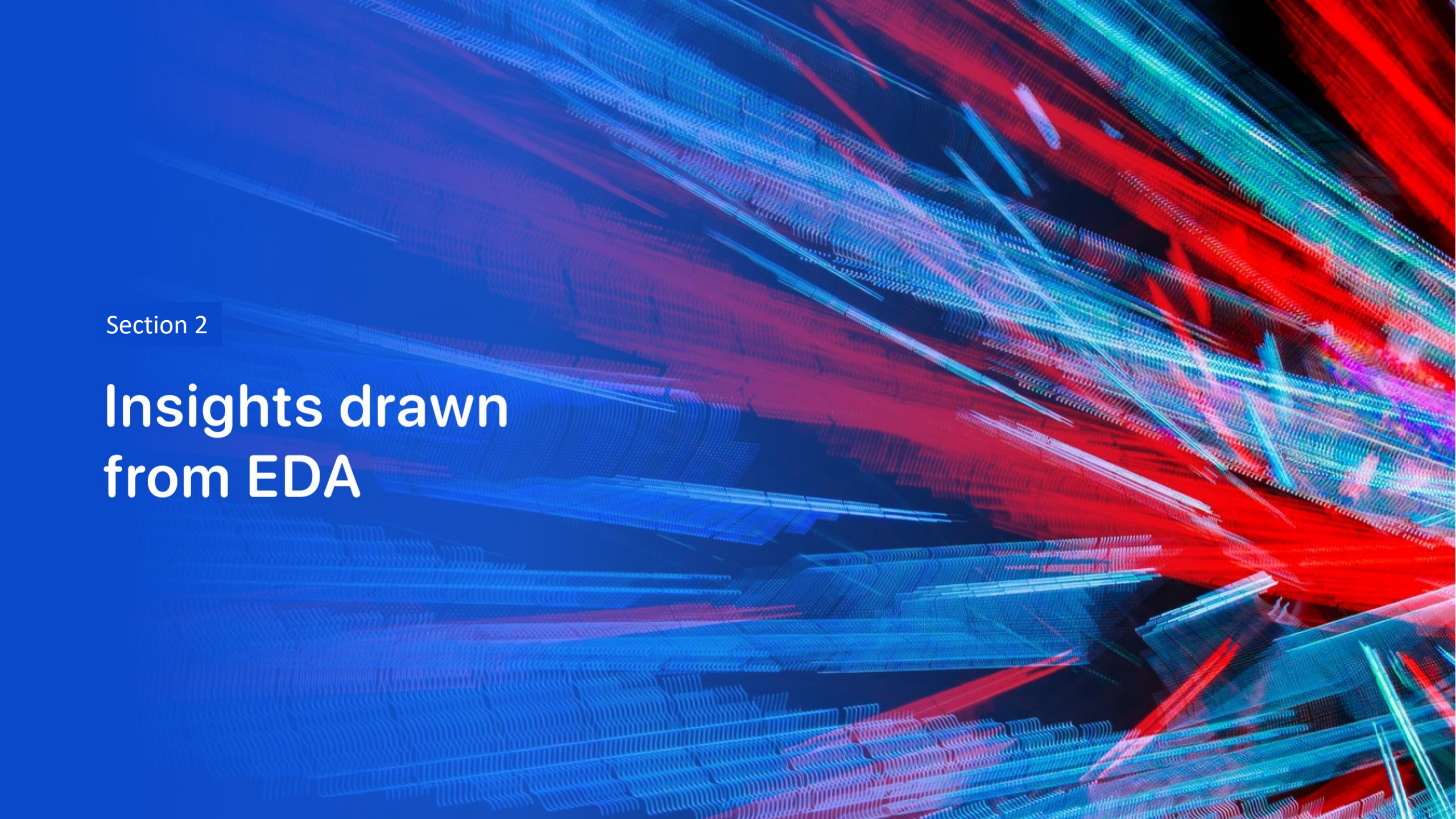
After the improving of model using feature engineering and tuning of algorithm next step is to find the best model that can be fit with better accuracy score.

From:

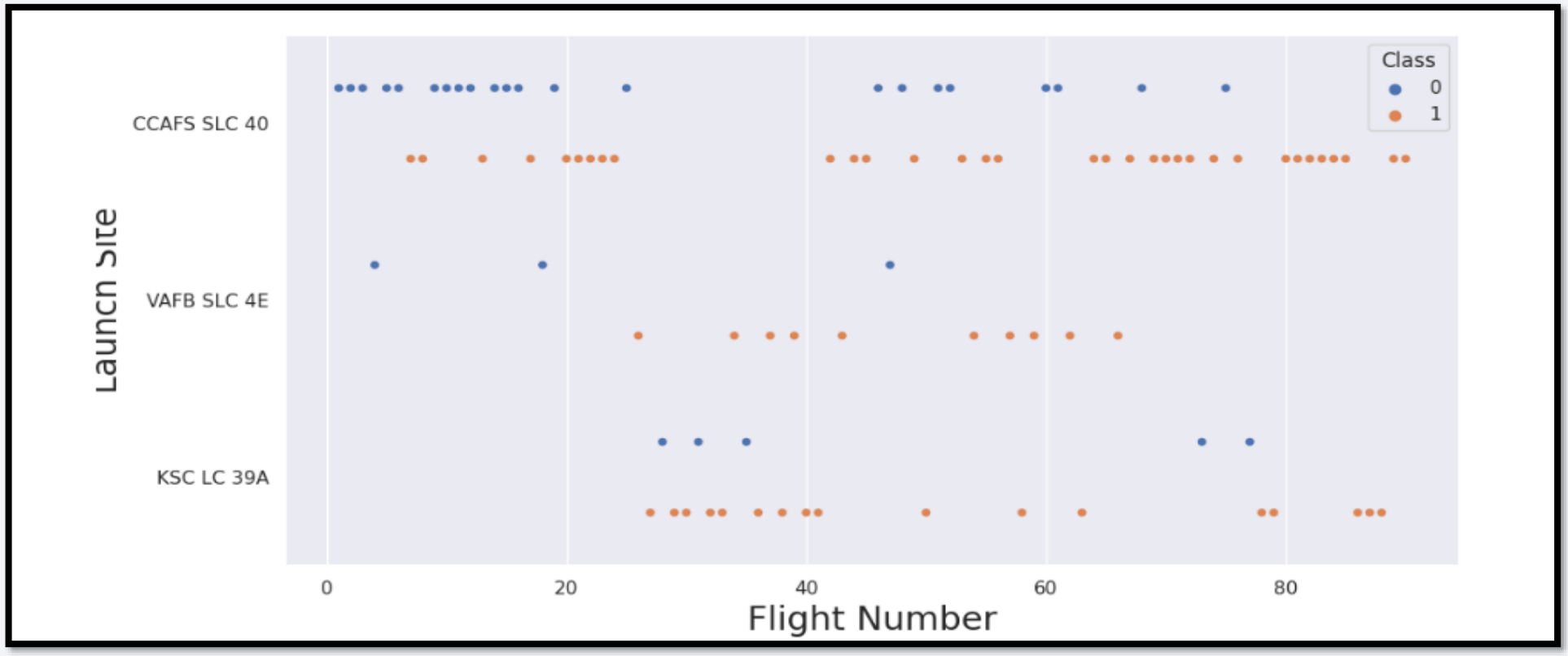
https://github.com/Mehar-Ali1122/Applied-Data-Science-Capstone/blob/main/SpaceX%20Machine%20Learning%20Prediction.jupyterlite.ipynb

Results

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

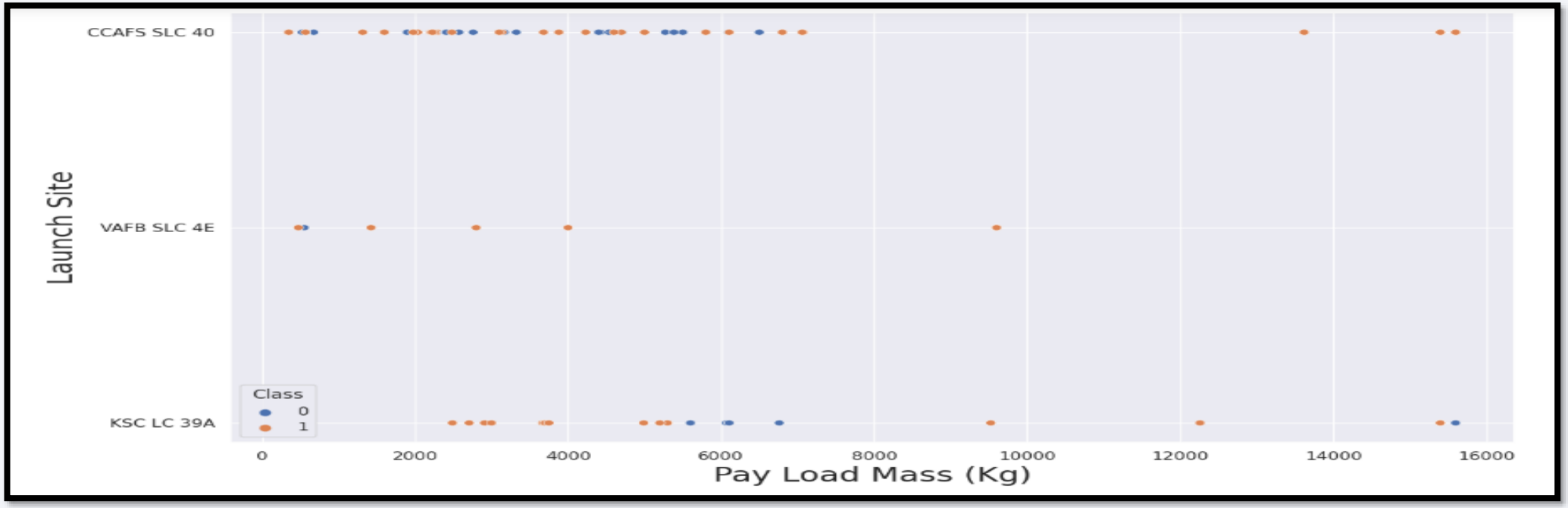

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



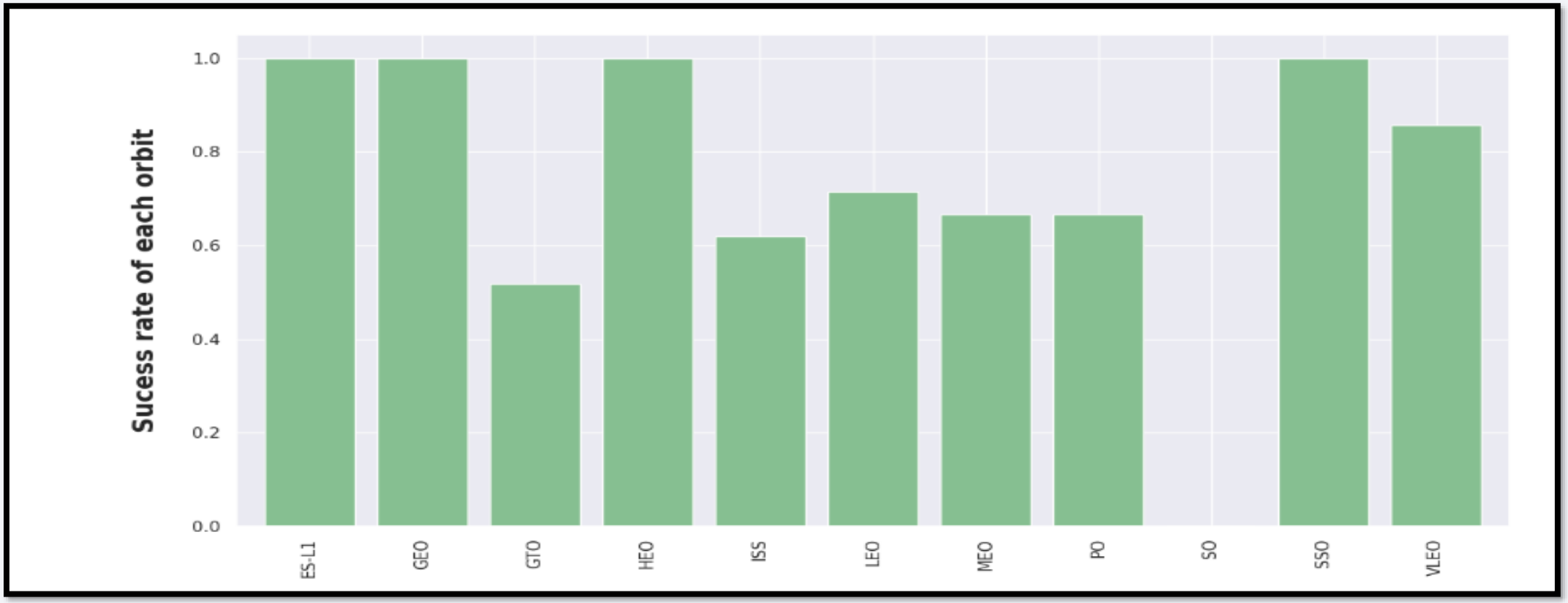
The Above scatter graph shows that launches from site CCAFS SLC 40 are higher as compared to the other sites

Payload vs. Launch Site



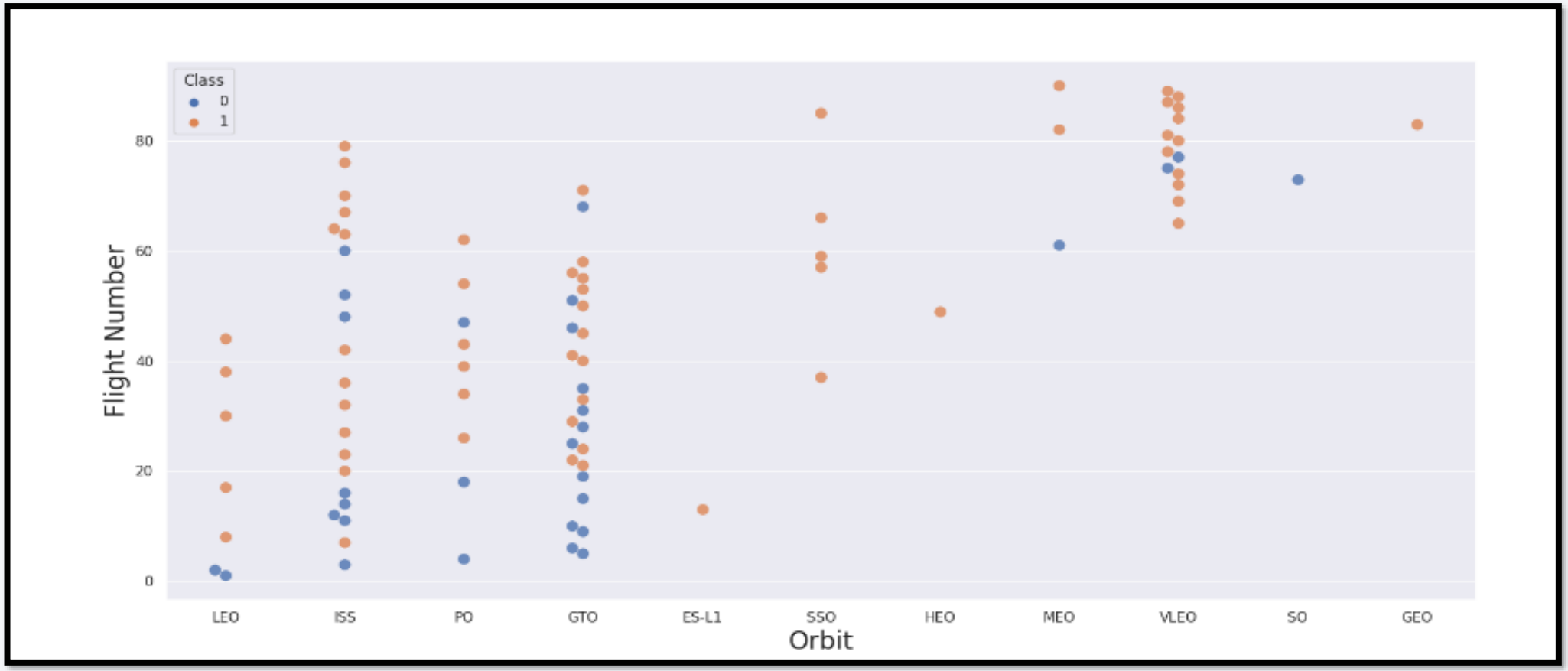
The Above scatter graph shows that from site CCAFS SLC 40 mass of pay loads launches in less quantity as compared to the other sites

Success Rate vs. Orbit Type



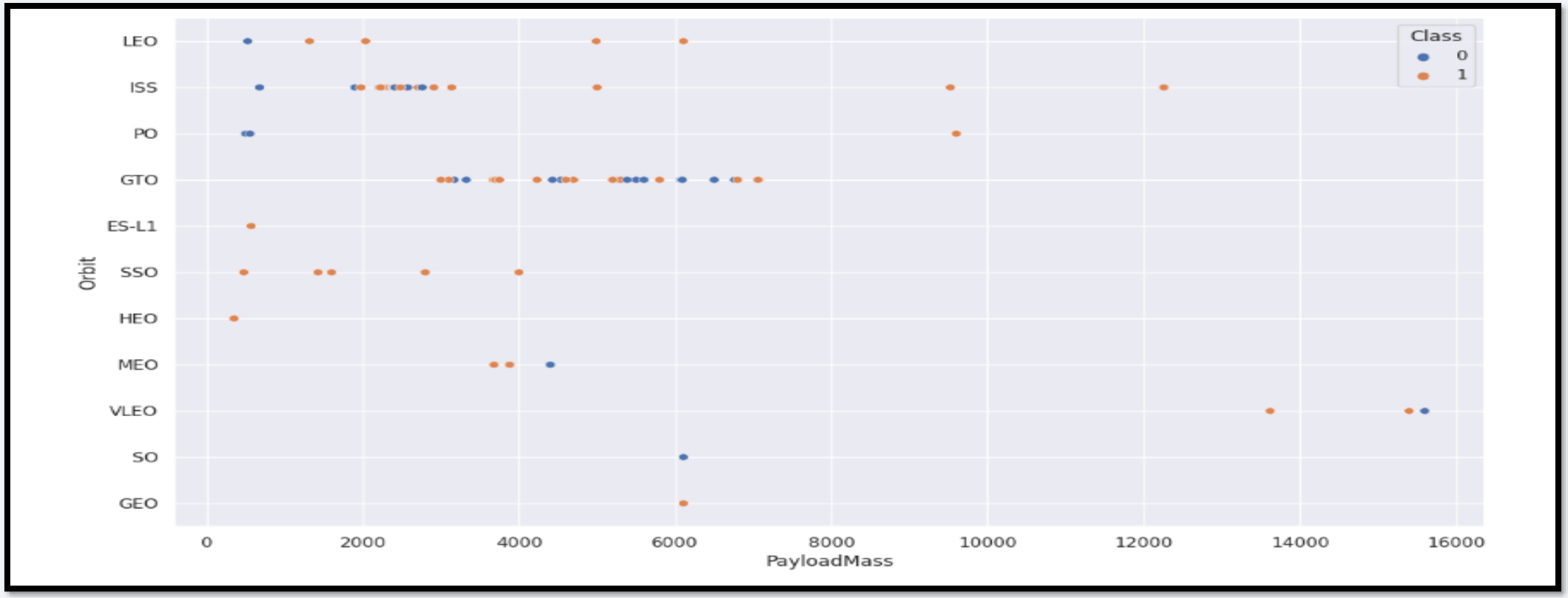
The Above bar graph shows that Orbit type ES-L1, GEO,HEO and SSO has highest success.

Flight Number vs. Orbit Type



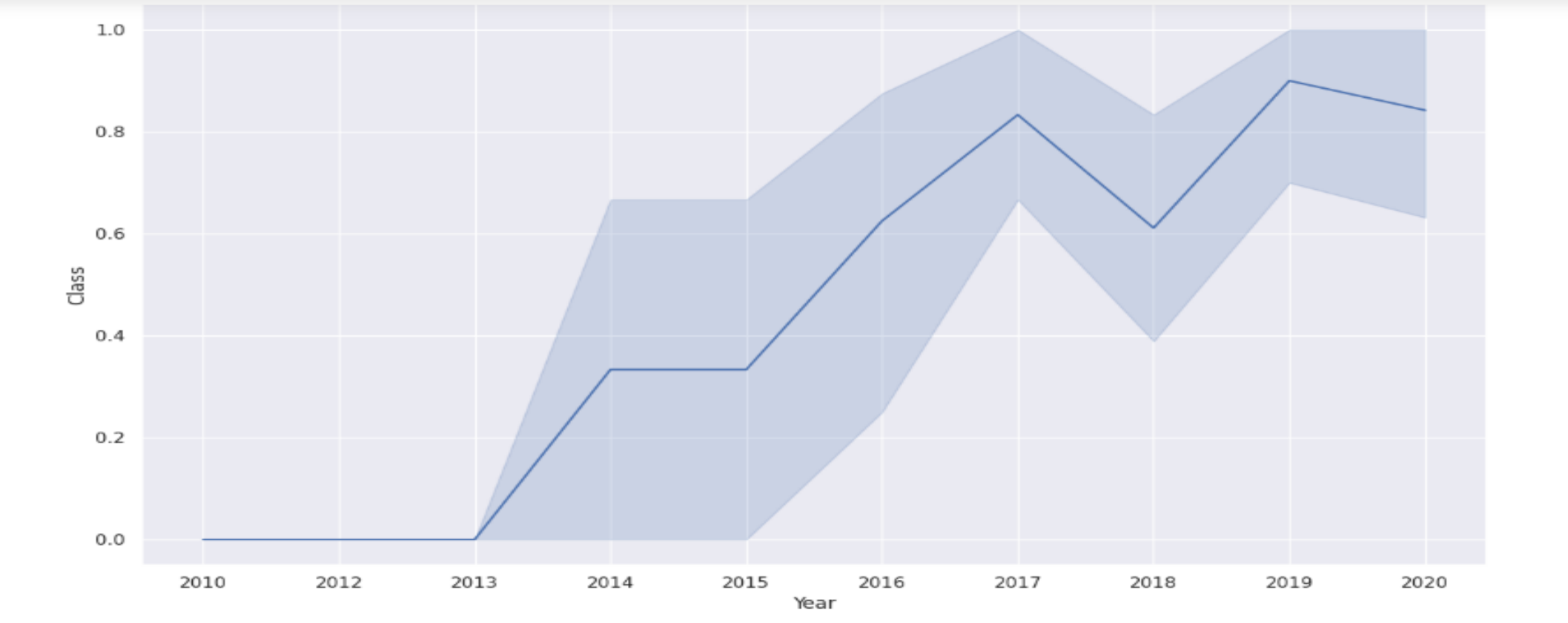
The Above bar graph shows that Orbit type LEU has highest success as number of flight are increases.

Payload vs. Orbit Type



The Above bar graph shows that increases in load (heavy load) has positive influence orbit LEO and ISS while negative influence on MEO, GTO and VLEO .

Launch Success Yearly Trend



The Above graph shows that success rate has increasing trend from 2013 while slightly a dip in 2018.

All Launch Site Names

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

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

We used word 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 LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

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

Total Payload Mass

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL \
    WHERE CUSTOMER = 'NASA (CRS)';
```

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

TOTAL_PAYLOAD_MASS
45596

We calculated the total payload carried by boosters from NASA as Total Payload mass equal to 45596 using the query as shown in the above figure.

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 AVERAGE_PAYLOAD_MASS FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

AVERAGE_PAYLOAD_MASS
2928.4

Average payload mass carried by booster version F9 v1.1 as 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(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)) from SPACEXTBL where  "Landing _Outcome"='Success (grou
```

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

min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2))
20151222

The dates of the first successful landing outcome on ground pad was 22nd December 2015

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 booster_version,PAYLOAD_MASS__KG_ ,"Landing _Outcome" from SPACEXTBL where  "Landing _Outcome"='Success (drone ship)
```

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

Booster_Version	PAYLOAD_MASS__KG_	Landing _Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

By using 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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER FROM SPACEXTBL 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

By using wildcard like '%' to filter for WHERE Mission Outcome was a success or a failure. From above failure is one and total success is 100.

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

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

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql SELECT substr(Date, 4, 2) as month,booster_version,"Landing _Outcome"from SPACEXTBL where "Landing _Outcome"='Failure (drone
```

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

month	Booster_Version	Landing _Outcome
01	F9 v1.1 B1012	Failure (drone ship)
04	F9 v1.1 B1015	Failure (drone ship)

By using a 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

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT "Landing _Outcome",count("Landing _Outcome")as LANDING_OUTCOME_COUNT\
from SPACEXTBL where  DATE between '04-06-2010' and '20-03-2017' \
group by "Landing _Outcome" order by count("Landing _Outcome") desc
```

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

Landing _Outcome	LANDING_OUTCOME_COUNT
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

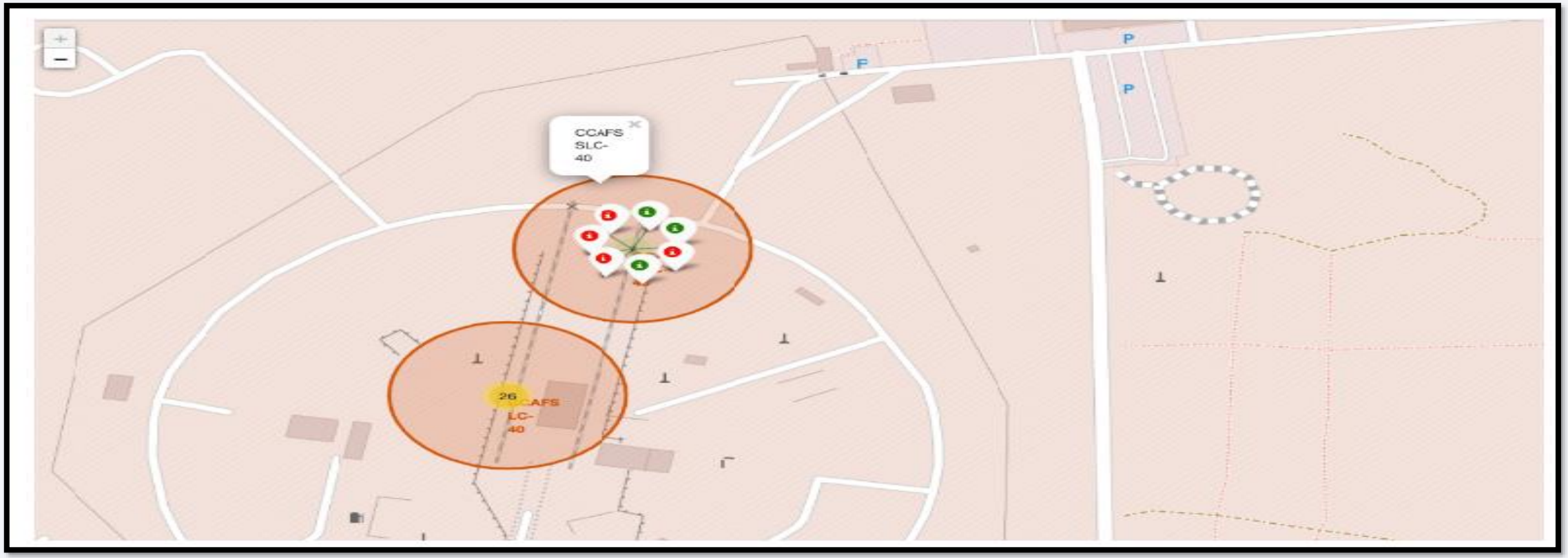
WHERE clause is used to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20 and used ORDER BYclause to order the grouped landing outcome in descending order.


Section 3
Launch Sites
Proximities Analysis

Location of all the Launch Sites

Above figure shows all launching sites which are located in USA.

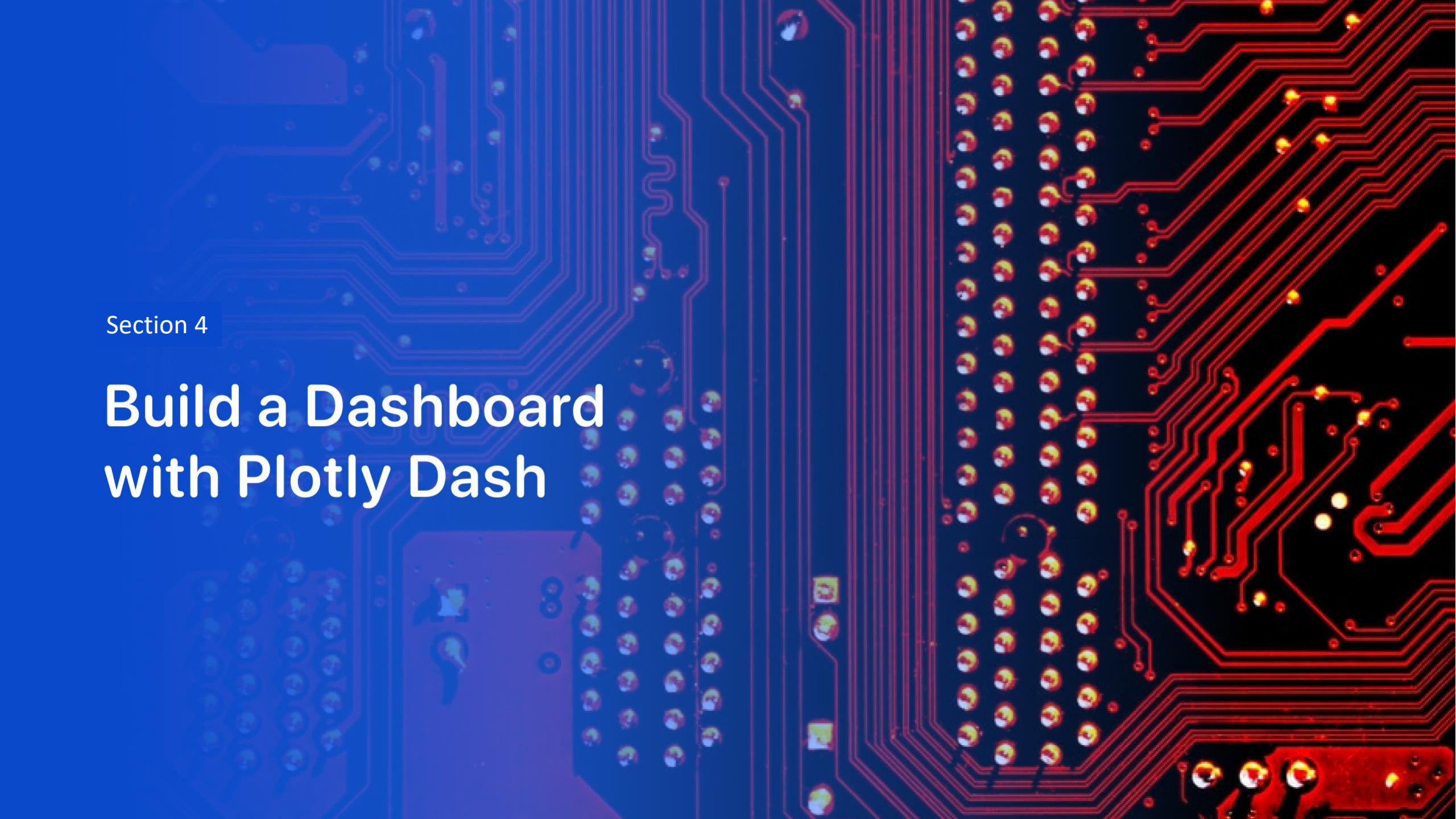
Folium Markers showing launch sites with color labels



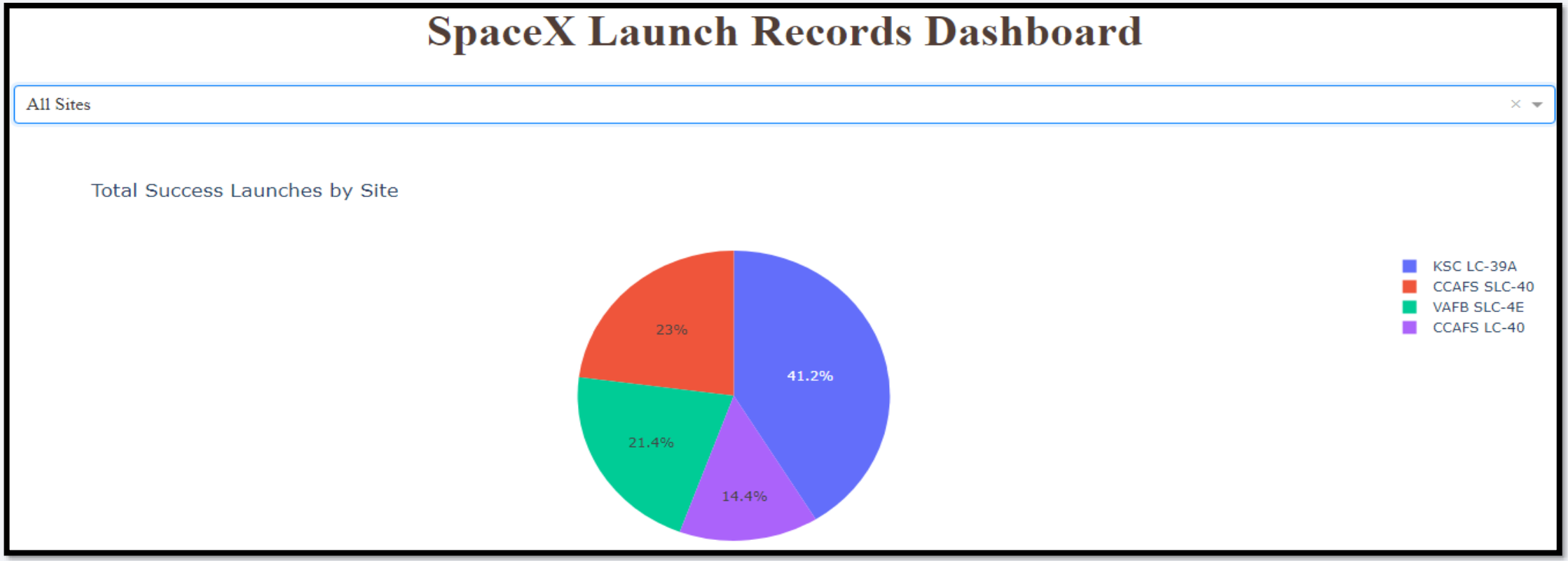
Above figure shows all launching sites marked with green are represent successful launches while red represent that launches are failed.

Launch Sites Distance to Landmarks

Above figure shows all launching sites in close proximity to coastline.

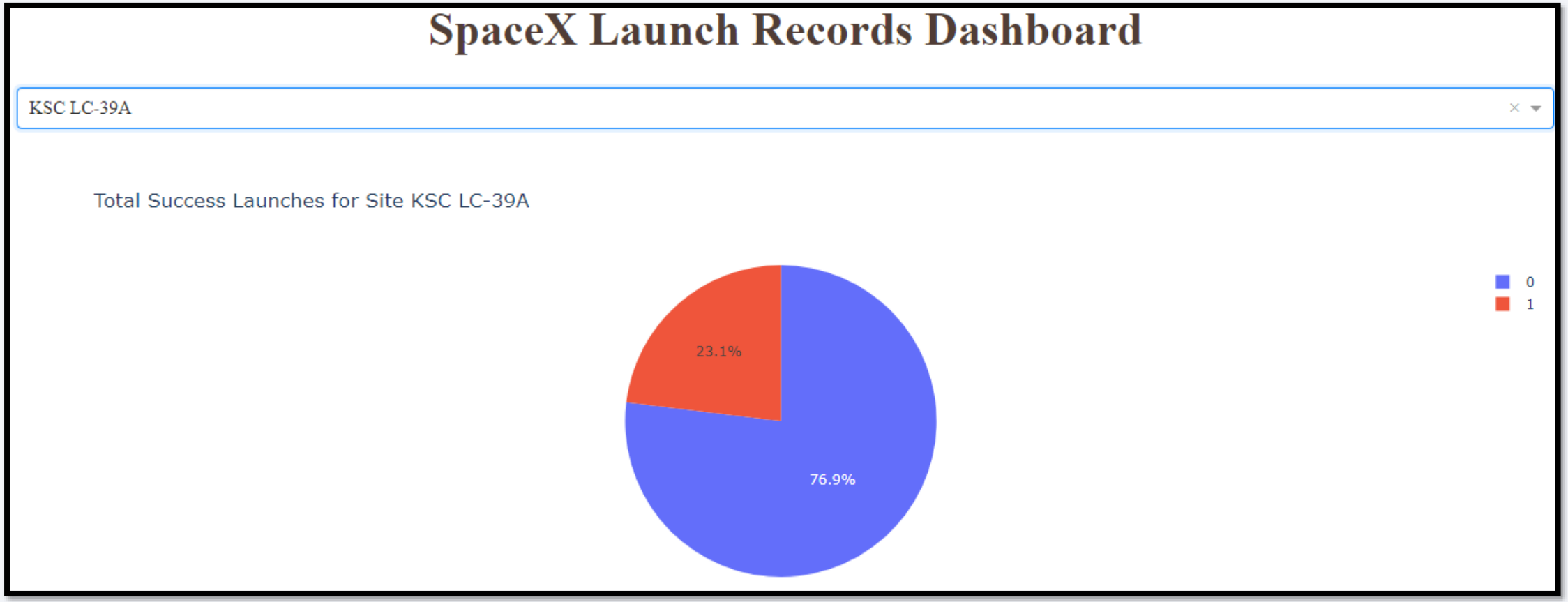

Section 4
Build a Dashboard with Plotly Dash

The success percentage by each sites



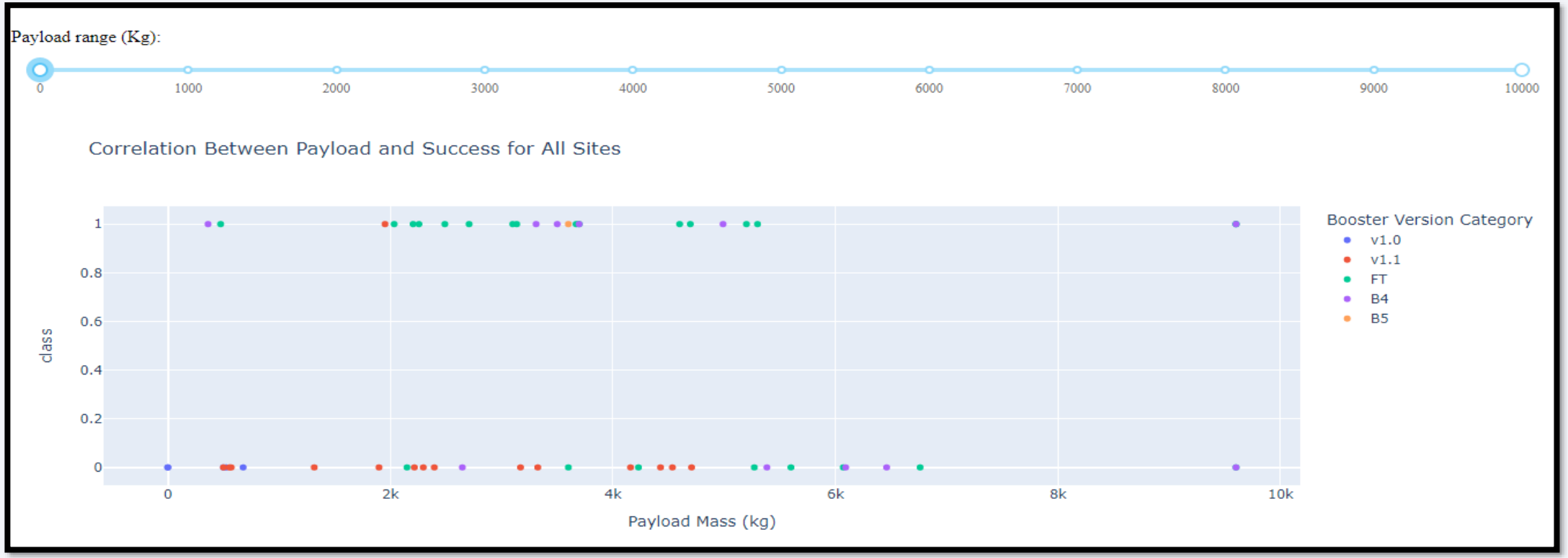
Above figure shows from all launching sites CCAFS SLC-40 is most successful.

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



Above figure shows from launching site KSC LC-39A is got 76.1% success rate while 23.1% failure rate.

Payload vs Launch Outcome Scatter Plot



Above figure shows all the success rate for low weighted payload is higher than heavy weighted payload

Section 5

Predictive Analysis (Classification)

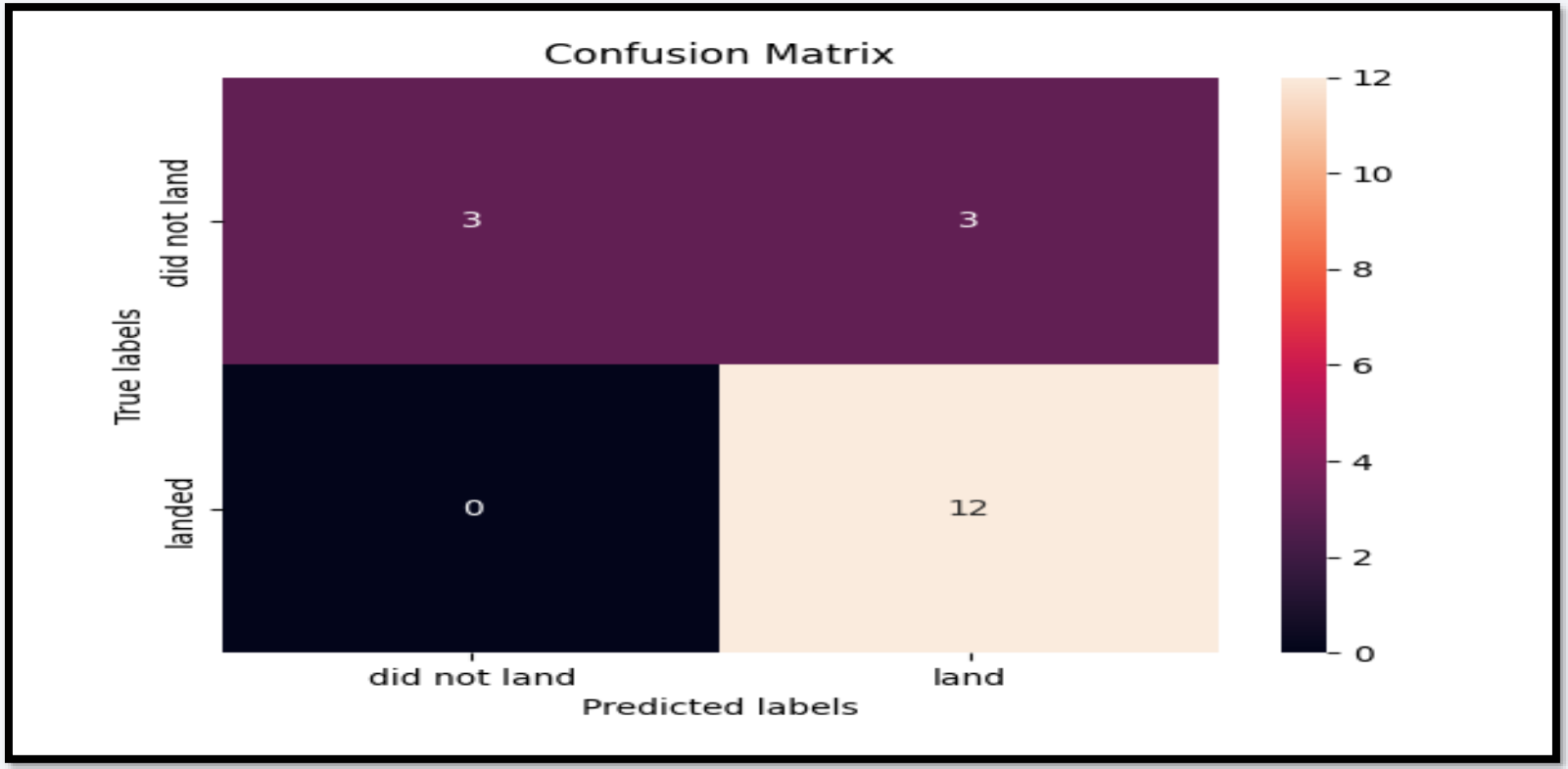
Classification Accuracy

```
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 0.8874999999999998
Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10,
'splitter': 'random'}
```

Above figure shows best algorithm to be the Tree Algorithm which have the highest classification accuracy

Confusion Matrix



Above figure shows that the main issue is the false positives so the unsuccessful landing marked as successful landing by the classifier

Conclusions

We can conclude that:

- Orbit type ES-L1, GEO,HEO and SSO has highest success rate
- KSC LC-39A have the most successful launches of any sites; 76.9%
- 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.
- SSO orbit have the most success rate; 100% and more than 1 occurrence.
- Decision Tree classifier algorithm is suitable and best for machine learning model

Thank you!