IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Asif Shahzad
January 10, 2025

Outline

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

Executive Summary

- **Summary of methodologies**
  - Collection of Data through API
  - Collection of Data by Web Scraping
  - Data Wrangling
  - EDA with SQL
  - EDA by Visualization
  - Interactive Visual Analytics with Folium
  - Predictive Analysis with Machine learning
- **Summary of all results**
  - EDA results
  - Interactive analytics with screenshots
  - Predictive Analytics results

Introduction

- Project background and context
- SpaceX advertises the cost of a Falcon 9 rocket launch on its website as **$62 million**, significantly lower than other providers whose costs typically exceed **$165 million** per launch. The substantial savings are primarily due to SpaceX's ability to reuse the rocket's first stage.
- Determining whether the first stage of a rocket will successfully land is critical for assessing the overall cost-effectiveness of a launch. By predicting the success of the first stage landing, we can estimate the associated costs and provide competitive insights for other companies aiming to bid against SpaceX for rocket launches.
- **Problems to Answer**

1. **What factors influence the success of a rocket landing?**
   Understanding the variables and conditions that determine whether a rocket will land successfully is essential to optimizing the landing process.
2. **How do various features interact to affect the likelihood of a successful landing?**
   Investigating the relationships and dependencies among different factors is critical to identifying the key drivers of success.
3. **What operating conditions are required to ensure a reliable landing program?**
   Establishing the necessary environmental, technical, and operational criteria is vital to achieving consistent landing success.

Section 1

Methodology

Methodology

Executive Summary

- **Data Collection:**

Data was collected by using two methods (**SpaceX API** and **web scraping)** from Wikipedia to confirm comprehensive analysis of related information.

- **Data Wrangling:**

We applied **one-hot encoding** to transform categorical variables into type which was suitable for analysis and modelling.

- **Exploratory Data Analysis (EDA):**

EDA using **visualizations** and **SQL queries** were conducted to find insights, identify patterns, and make the data for progressive analytics.

- **Interactive Visual Analytics:**

With **Folium** and **Plotly Dash** interactive visual dashboards were created

- **Predictive Analysis:**

Classification models were developed by using a few machine learning algorithums to predict key outcomes. This included:

- **Model building**: Constructing robust predictive models.
- **Model tuning**: Optimizing hyperparameters for improved performance.
- **Model evaluation**: Assessing the effectiveness of models using appropriate metrics

Data Collection

The data was gathered by using the following methods:

- **SpaceX API:** GET requests were utilized to access the SpaceX API, retrieving data in JSON format
This data was then parsed using the .json() function and transformed into a structured Pandas DataFrame using .json_normalize()

- **Data Cleaning and Preprocessing:** To check data quality, we applied different functions in data wragling and cleaning process including handling missing values and addressing any inconsistencie

- **Wikipedia Web Scraping:** We applied BeautifulSoup to scrape Falcon 9 launch records from Wikipedia, extracting HTML tables and converting them into Pandas DataFrames for subsequent analysis

Data Collection – SpaceX API

- We applied the get request to the SpaceX API to gain data, then cleaned the gained data and did some other basic data wrangling and formatting process
- https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/jupyter-labs-spacex-data-collection-api%20(2).ipynb

1. Get request for rocket launch data using API

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection by WebScraping

- We used web scrapping technique to scrape Falcon 9 launch records from wikipedia with BeautifulSoup
- We parsed the required table(2) and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/jupyter-labs-webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

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

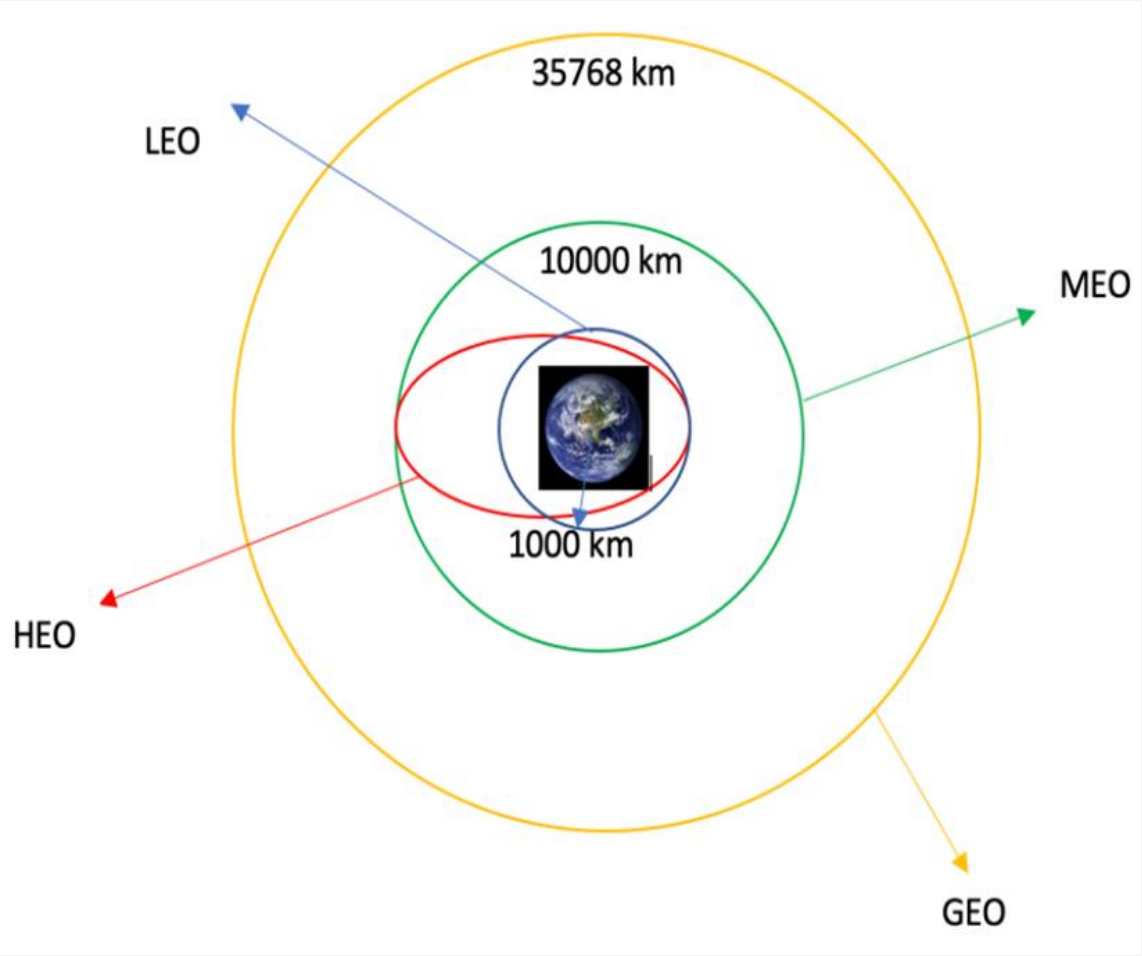
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

```
n [16]: df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

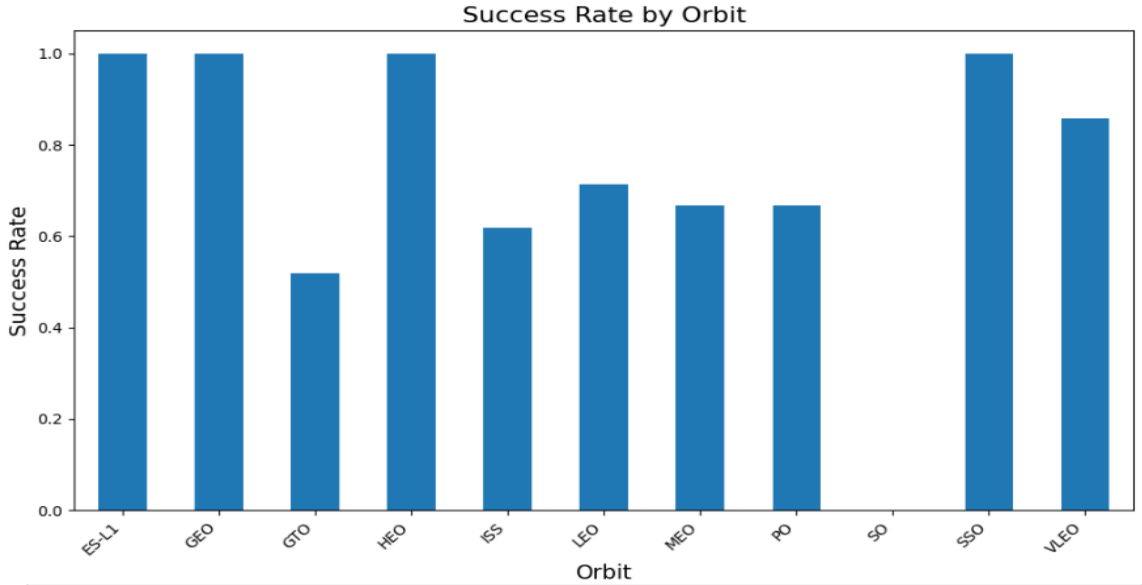
Data Wrangling

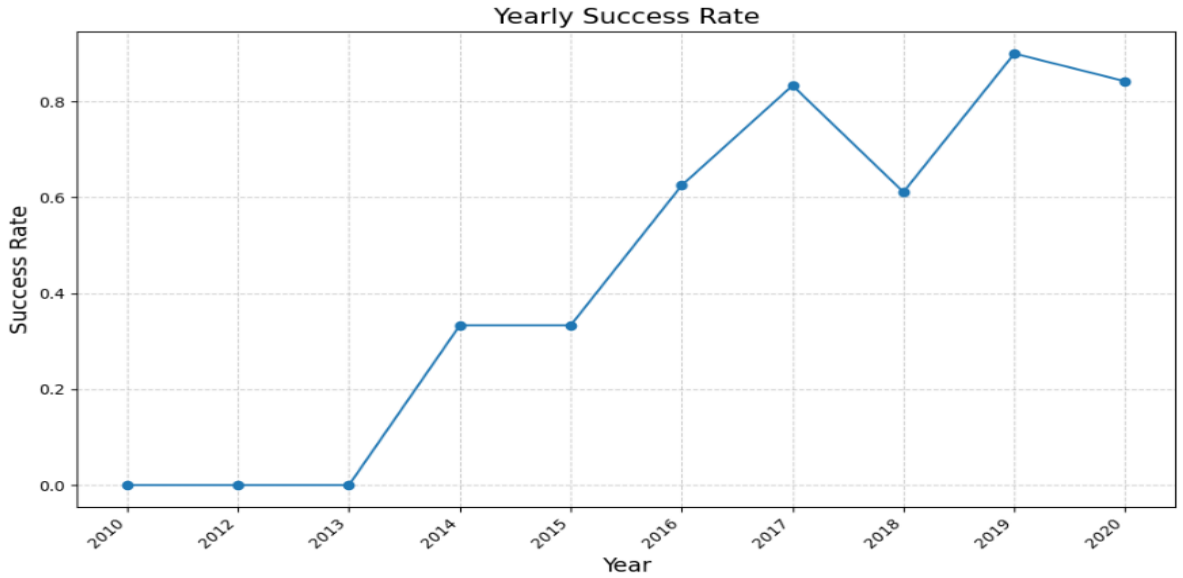
- We applied exploratory data analysis and checked the training labels.
- We calculated total number of launches at each site, and the number and occurrence of each orbits
- We developed landing outcome label from outcome column and saved the results to csv.
- The link to the notebook is https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

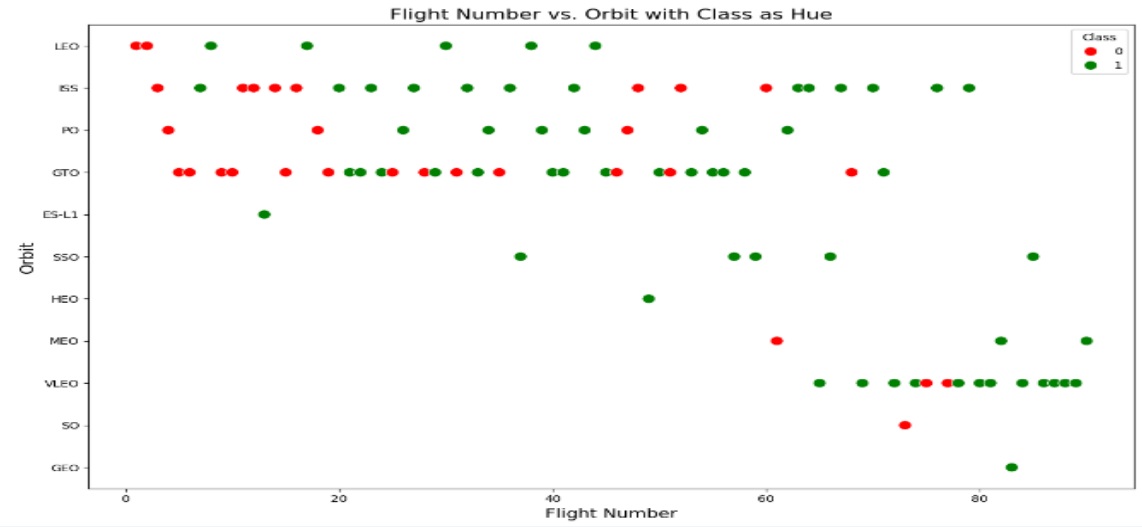


EDA with Data Visualization

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend
https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/EDA%20with%20Visualiztion.ipynb







EDA with SQL

- We loaded the SpaceX dataset into a MySQL database
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities

https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20.ipynb

Build a Dashboard with Plotly Dash

- We constructed an interactive dashboard with Plotly dash
- We designed pie charts displaying the total launches by a certain sites
- We schemed scatter graph presenting the relationship with Outcome and Payload Mass (Kg) for the diverse booster version.

https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/spacex_dash_app%20(1).py

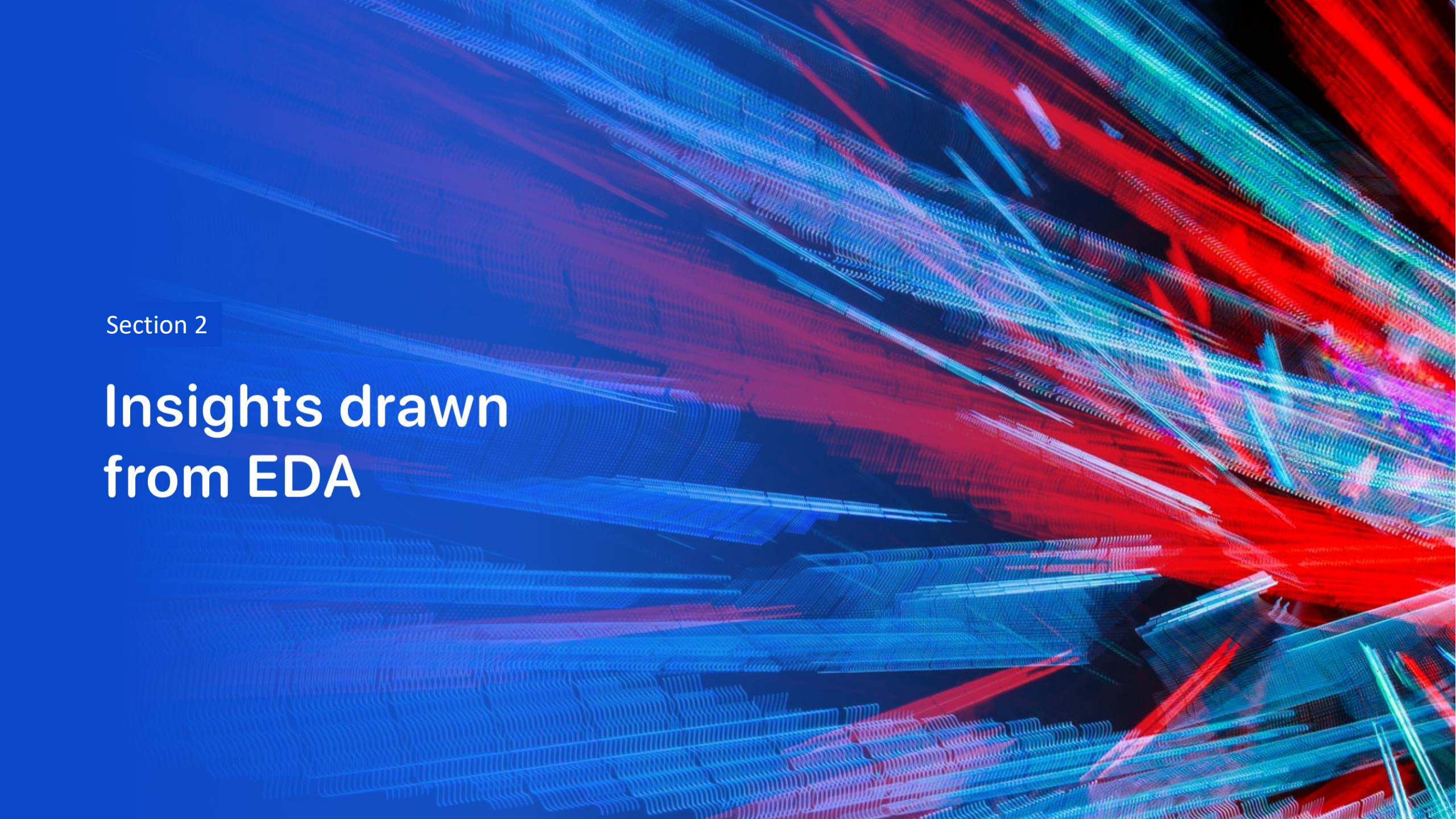
Predictive Analysis (Classification)

- For prediction we used the data set with numpy and pandas, transformed it , split it into training and testing set
- We constructed diverse machine learning models and adjust different hyperparameters by GridSearchCV
- We checked accuracy of the testing model as the metric for our model, upgraded the model using feature engineering and algorithm tuning
- We created the best fit classification model.

https://github.com/Shahzad627/IBM-Data-science-Capstone-SpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

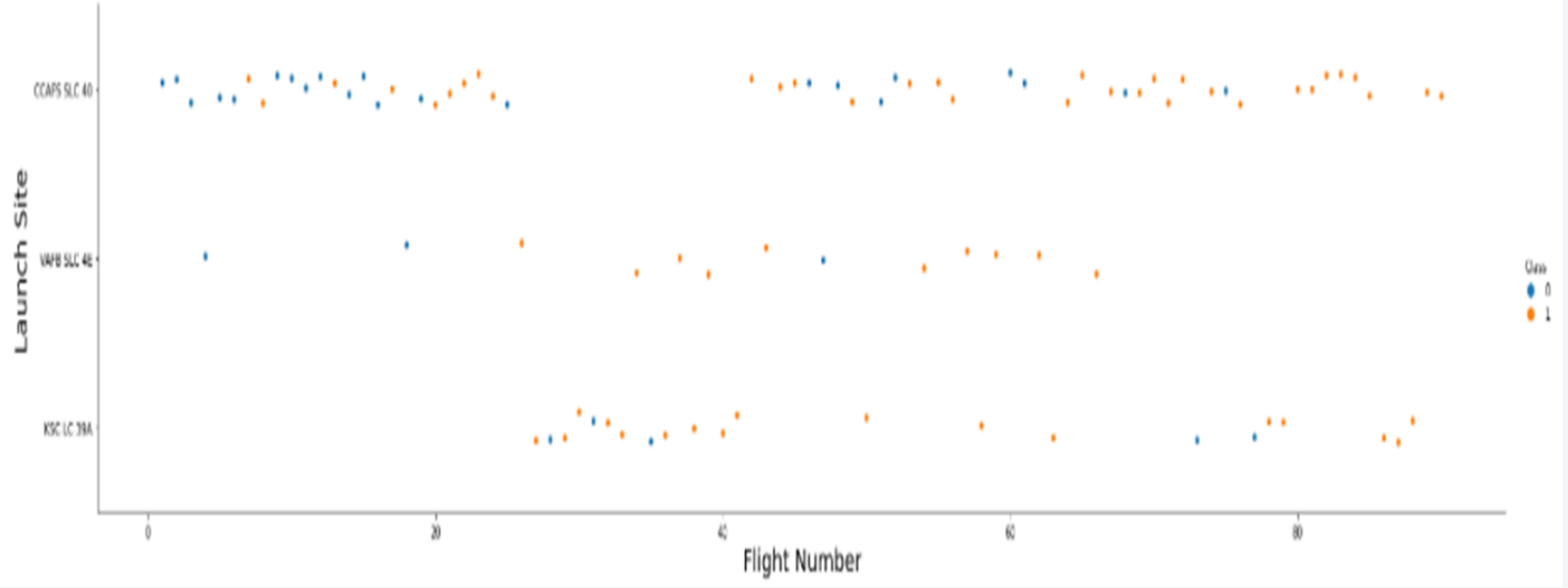
Results

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


Section 2
Insights drawn from EDA

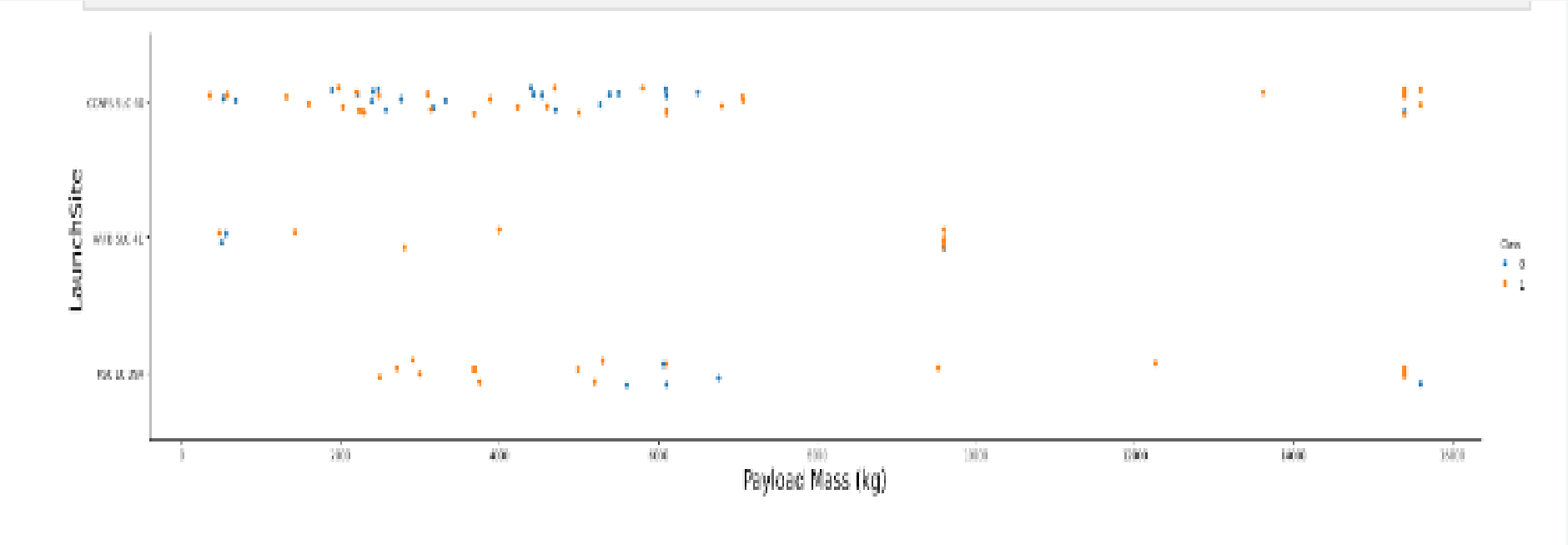
Flight Number vs. Launch Site

- We plotted a scatter chart between Flight Number and Launch site
- we observed that the larger the flight number at a launch site, the greater the success rate at a launch site



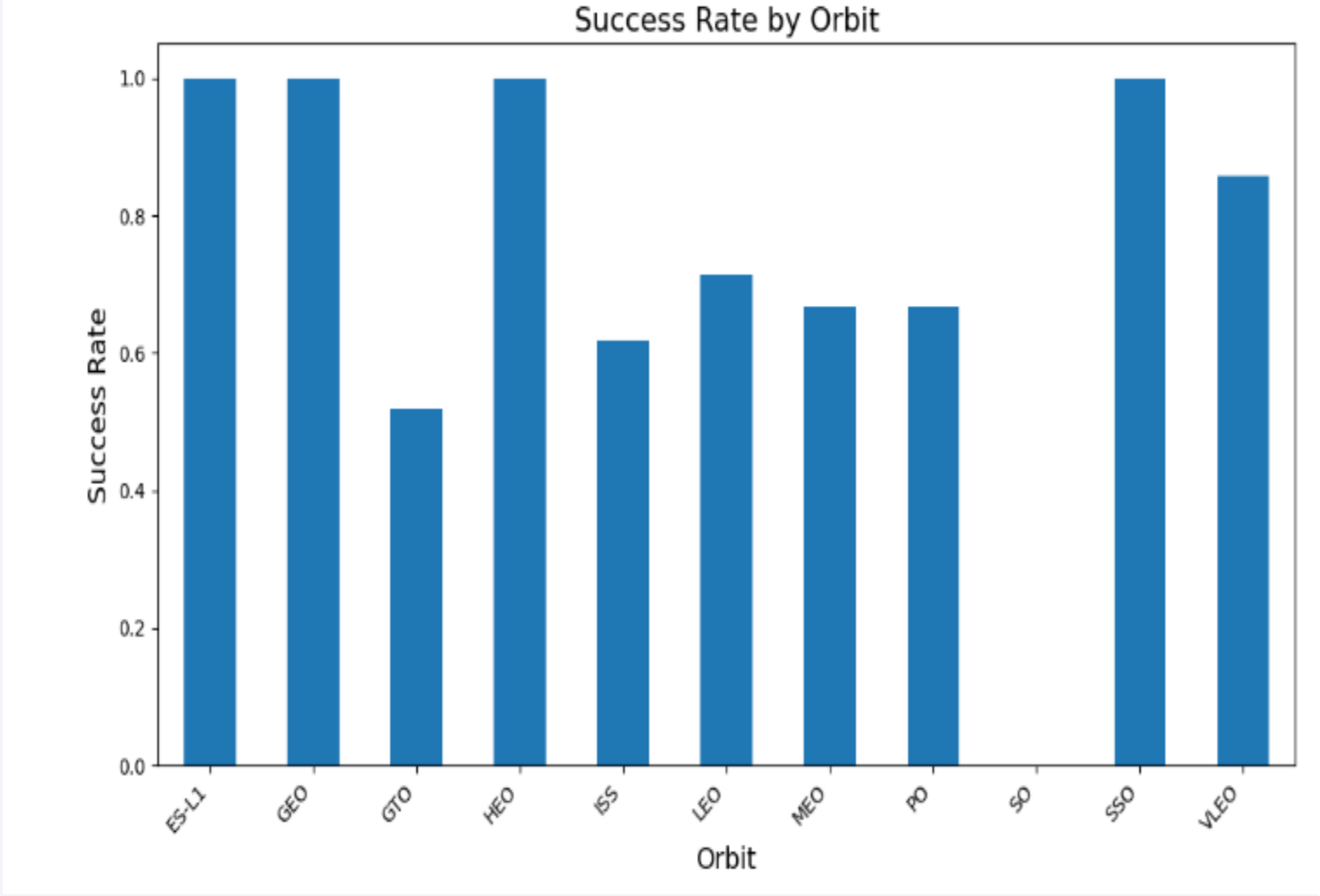
Payload vs. Launch Site

- We plotted a scatter chart between Payload mass and Launch site
- we observed that greater the mass of at a launch site (CCAFS SLC40, higher the success rate



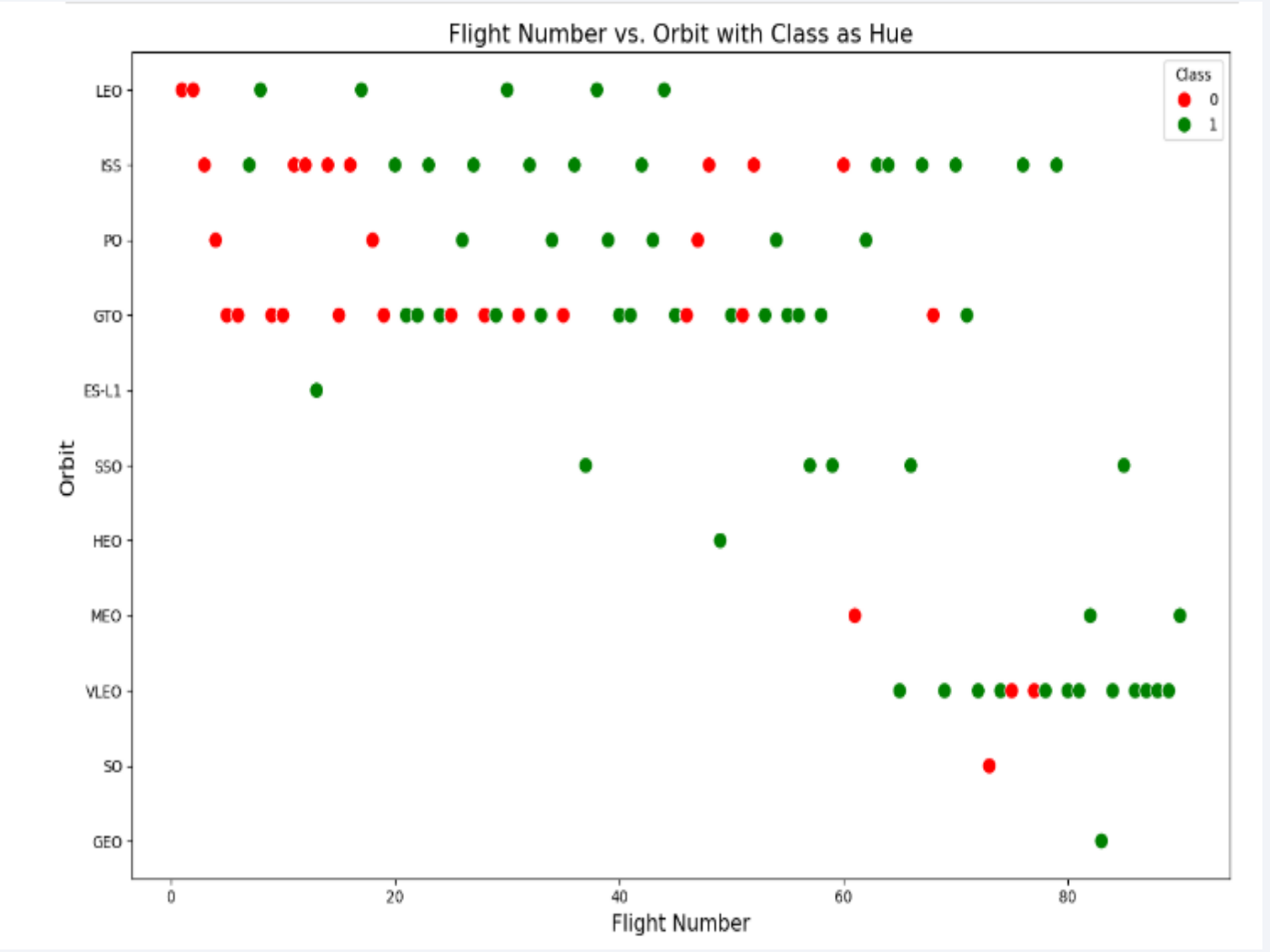
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the greatest success rate



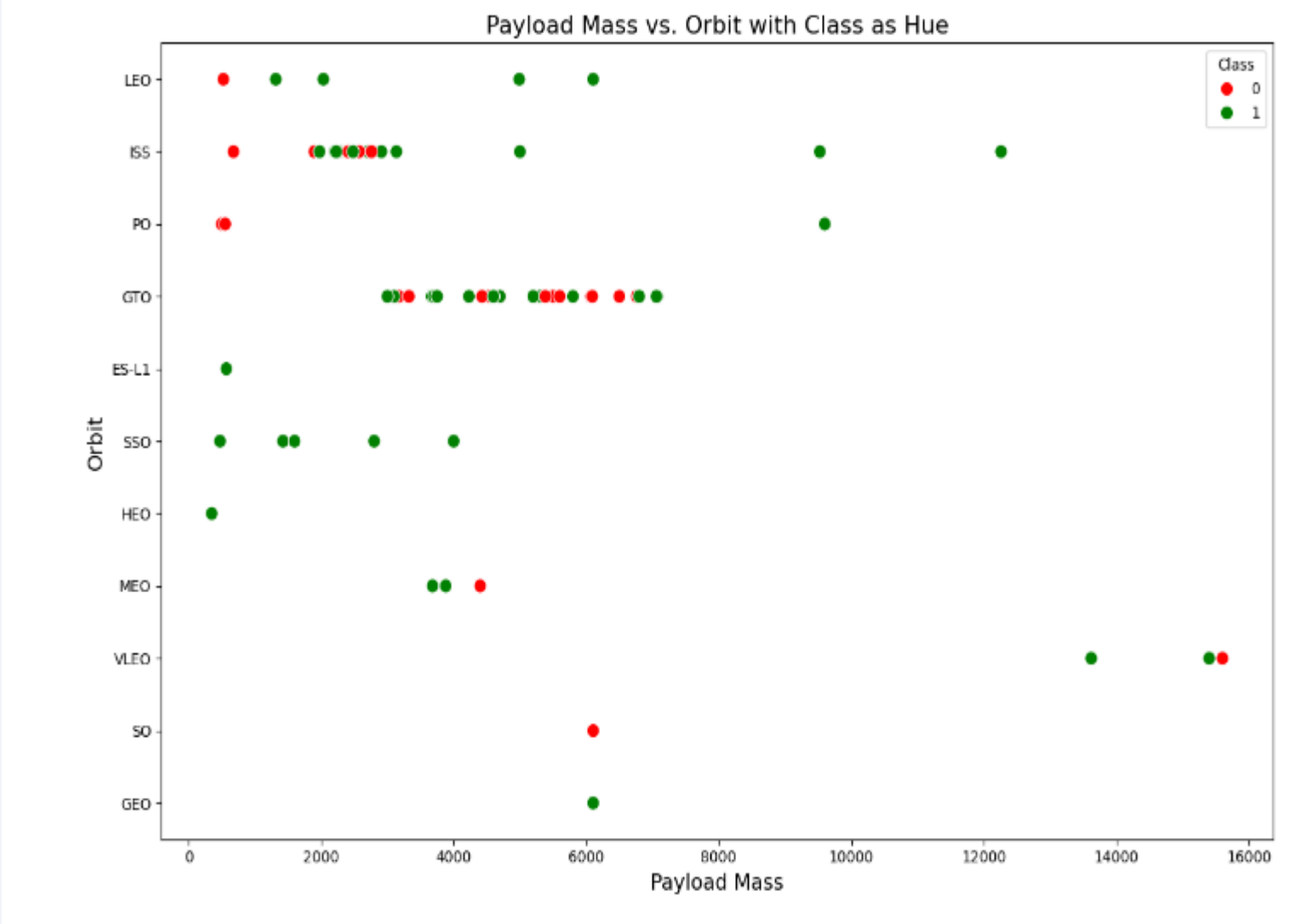
Flight Number vs. Orbit Type

- We can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success



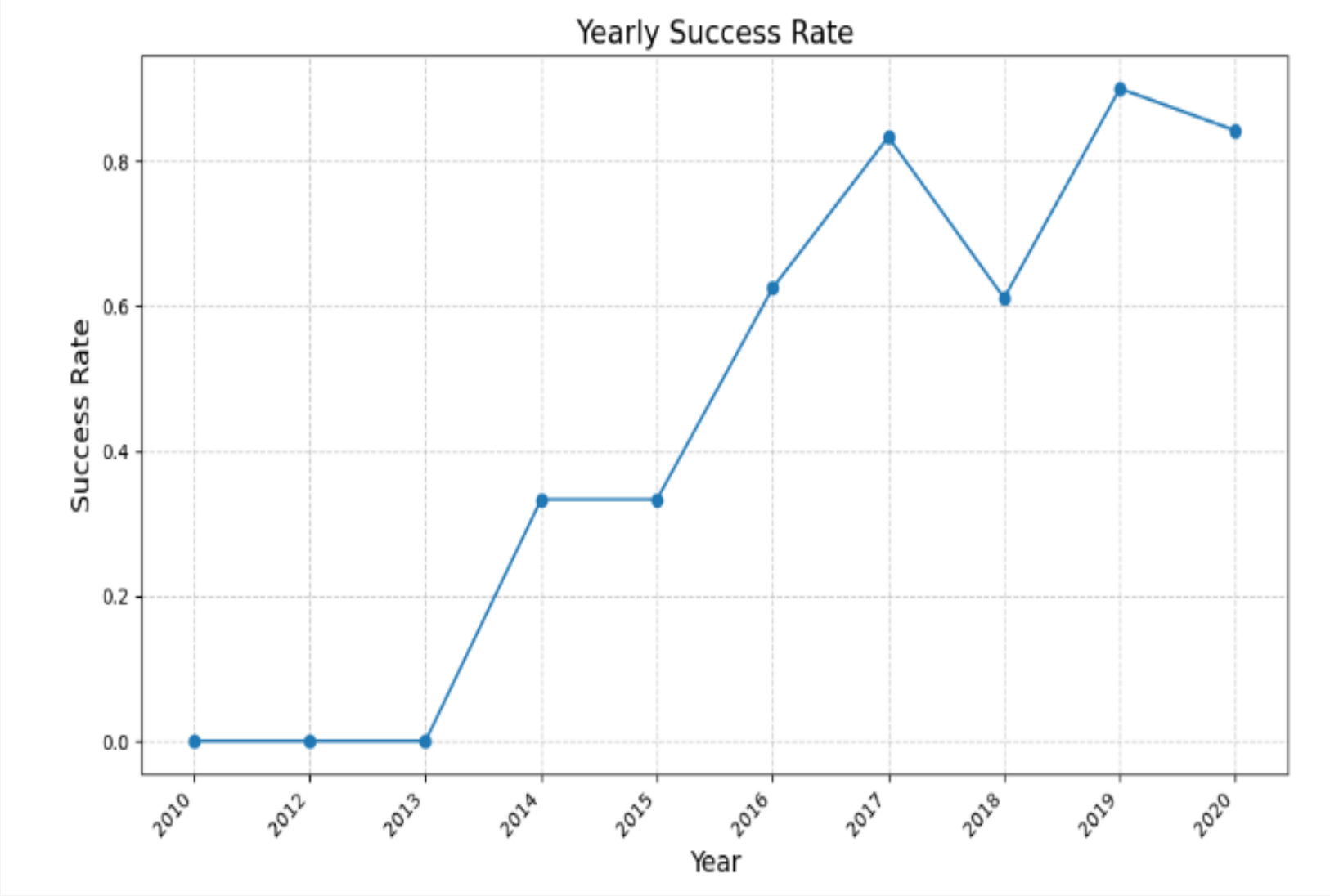
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present



Launch Success Yearly Trend

WE can observe that the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- We applied the key word **DISTINCT** to display only unique launch sites from the SpaceX data

Task 1

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

In [10]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE;
```

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

Out[10]:

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

Launch Site Names Begin with 'CCA'

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

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

In [11]:
```
%sql SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

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

In [12]:
```
%sql SELECT SUM(payload_mass__kg_) FROM SPACEXTABLE WHERE customer = 'NASA (CRS)';
```

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

Out[12]:

SUM(payload_mass__kg_)
45596

Average Payload Mass by F9 v1.1

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

Task 4

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

In [13]:
```
%sql SELECT AVG(payload_mass__kg_) FROM SPACEXTABLE WHERE booster_version = 'F9 v1.1';
```

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

Out[13]:

AVG(payload_mass__kg_)
2928.4

First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was June 04, 2010

Task 5

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

Hint:Use min function

In [14]:
```
%sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE mission_outcome = 'Success';
```

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

Out[14]:

MIN(DATE)
2010-06-04

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

In [15]:

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We used count method to calculate the total number of successful and failure mission outcomes
- The result of total mission-outcome was 101

Task 7

List the total number of successful and failure mission outcomes

In [20]:
```
%sql SELECT COUNT(mission_outcome) FROM SPACEXTABLE;
```

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

Out[20]:

COUNT(mission_outcome)
101

Boosters Carried Maximum Payload

- We find the boosters that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function

Task 8

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

In [21]:
```
%sql SELECT booster_version FROM SPACEXTABLE ORDER BY payload_mass__kg_ LIMIT 10
```

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

Out[21]:

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 B4 B1045.1
F9 FT B1038.1
F9 v1.0 B0006
F9 v1.1 B1003
F9 v1.0 B0005
F9 v1.1 B1017
F9 v1.1 B1013
F9 v1.0 B0007

2015 Launch Records

- We used a combinations of the **WHERE** clause and **Substr(Date)** function to filter for failed landing outcomes in done ship, their booster versions, and launch site names for year 2015

Task 9

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [38]:
```
%sql SELECT substr(Date, 6, 2) AS Month, landing_outcome, booster_version, launch_site FROM SPACEXTABLE WHERE substr(Date,
```

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

Out[38]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

In [42]:
```
%sql SELECT LOWER(landing_outcome) AS Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' ANI
```

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

Out[42]:

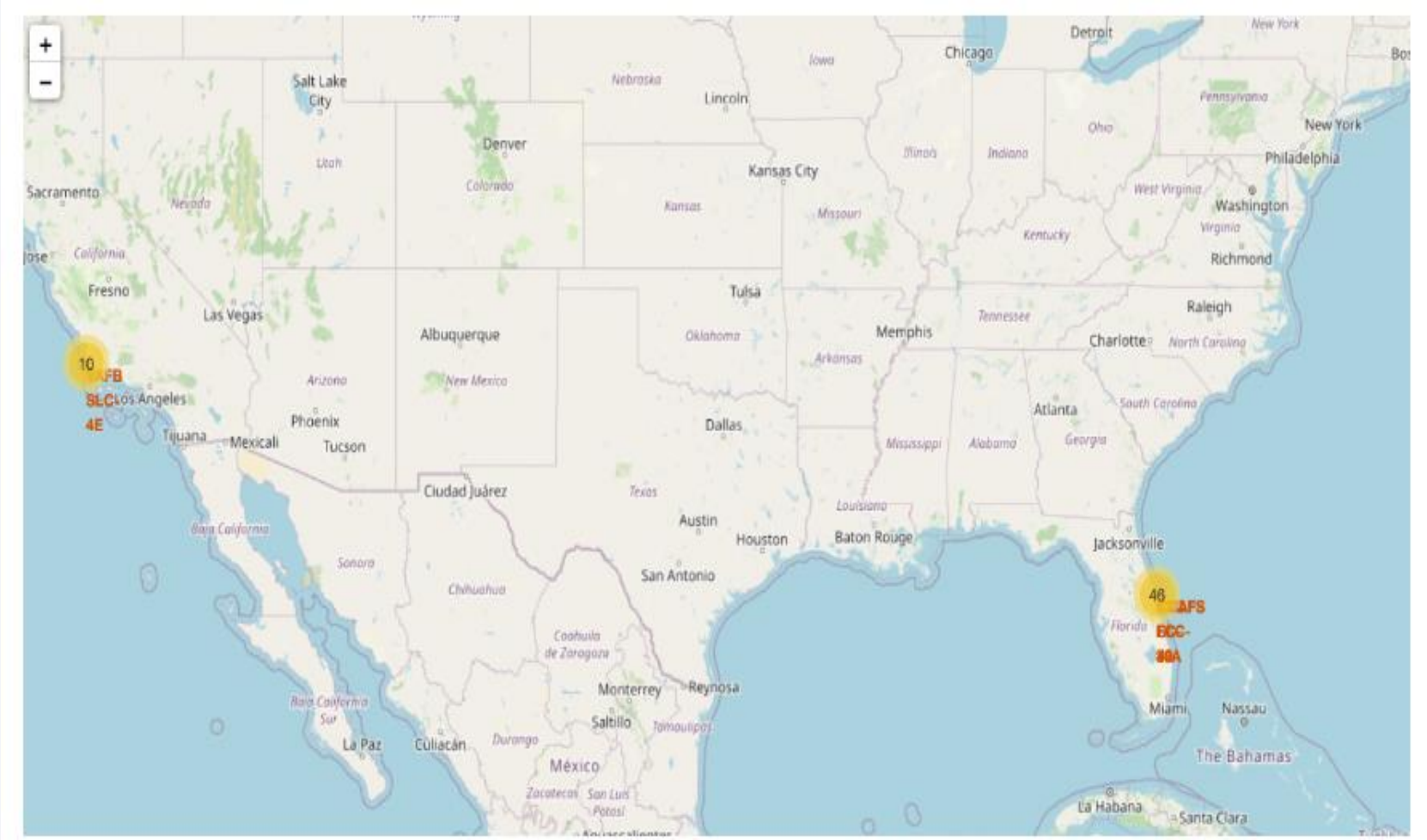
Outcome	OutcomeCount
no attempt	10
success (drone ship)	5
failure (drone ship)	5
success (ground pad)	3
controlled (ocean)	3
uncontrolled (ocean)	2
failure (parachute)	2
precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Launch sites with global map markers

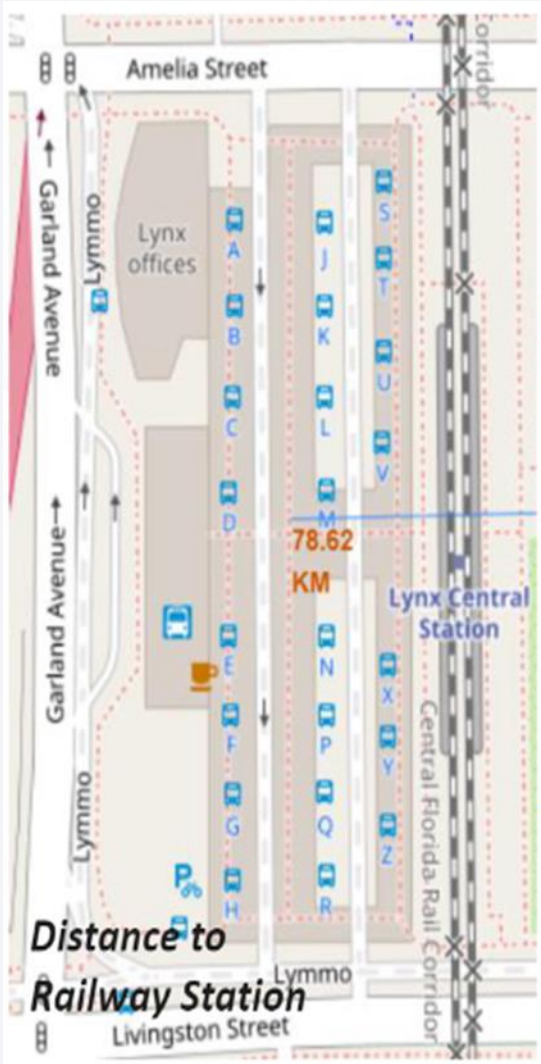
We can observe all launch sites with map markers on this map and can check the exact location of the sites



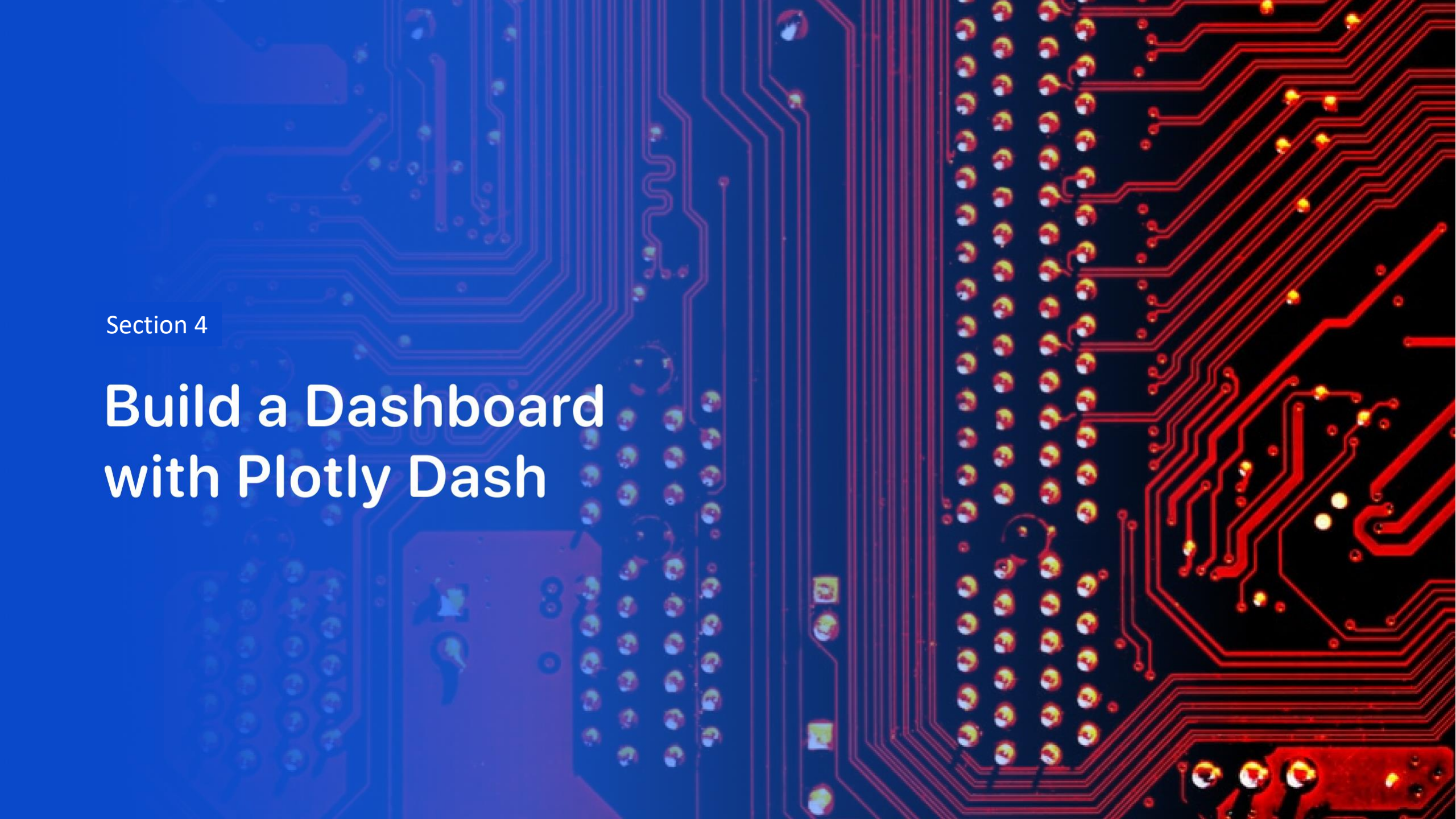
Markers displaying launch sites with color labels

- Here we can see Launch sites with color labels showing the location of the sites and a circle for each site with green and red markers
- This screen shot is clearly displaying the CCAFS SLC 40 site with green colors showing successful launches and red markers with failures
- From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates

Estimation of launch Site distance to Coastline, Railway, Highway

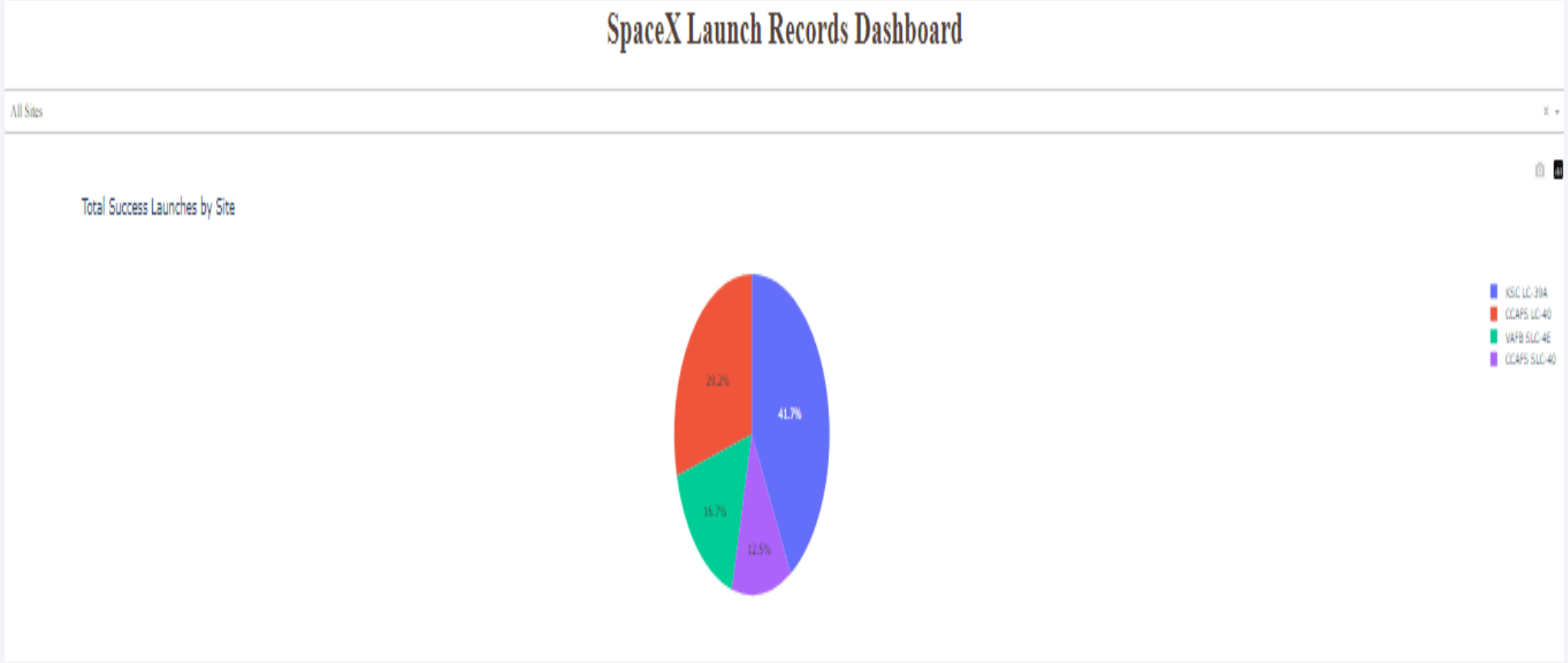


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

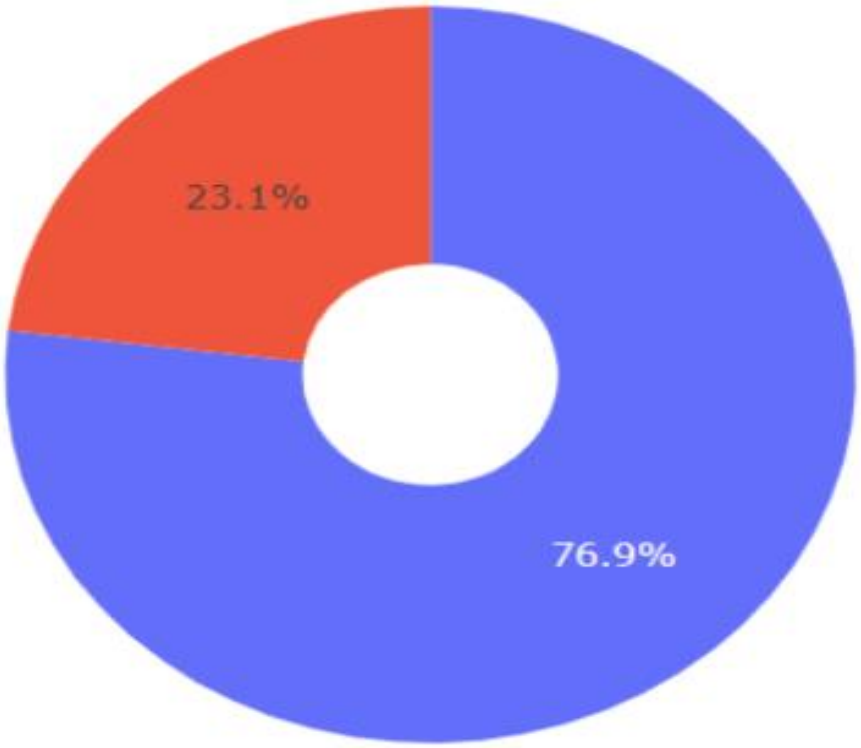

Section 4
Build a Dashboard with Plotly Dash

Dashboard showing the success rate of launch sites

KSC LC-39 with blue had the most successful launches from all the sites



Dashboard showing the KSC LC-39 site with launch success ratio



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

Dashboard showing a range slider of Payload vs. Launch Outcome scatter plot for all sites

We can observe that low weighted payload had high success rate than that of high weighted payload mass

Section 5

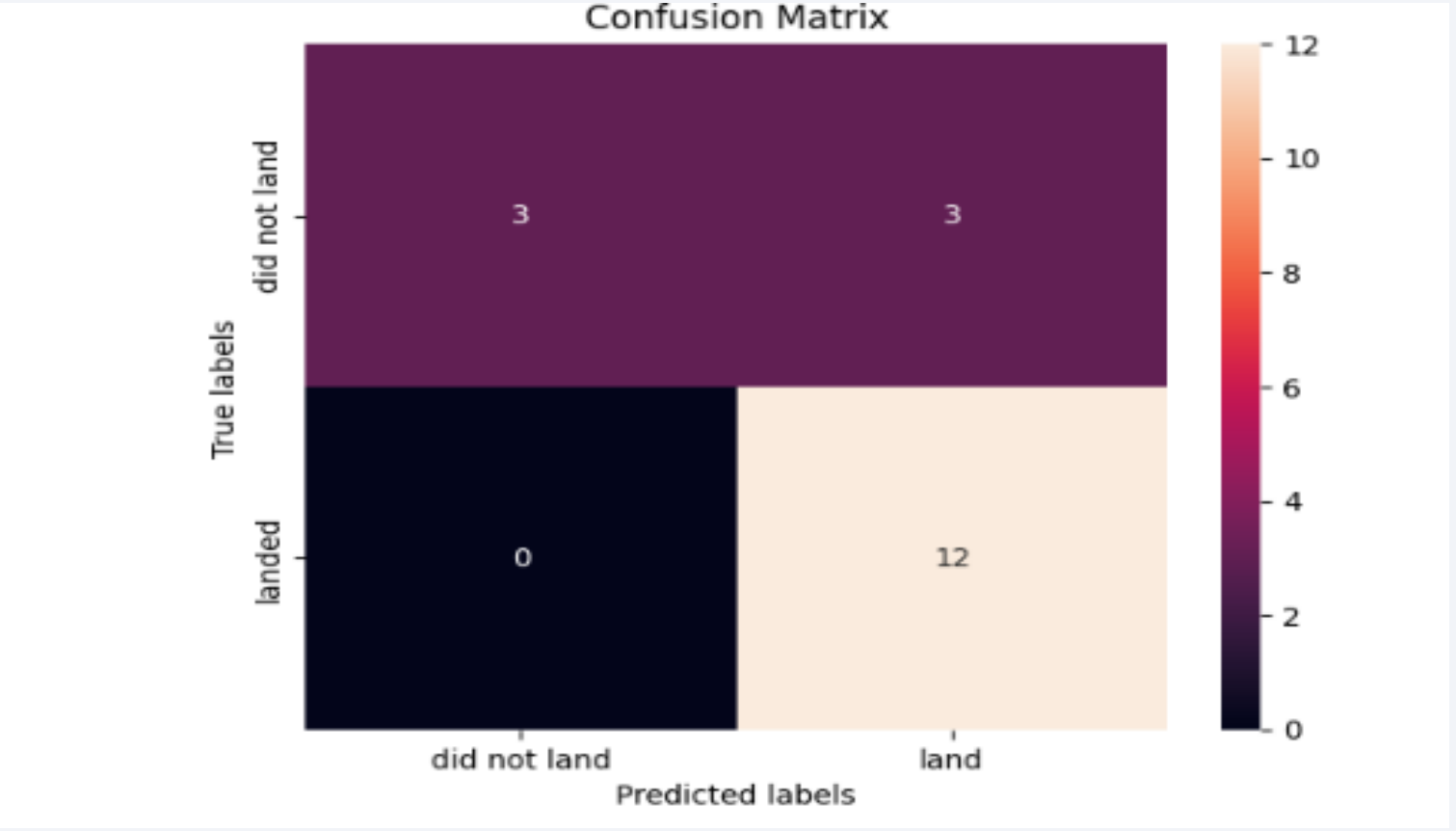
Predictive Analysis (Classification)

Classification Accuracy

All the model almost performed similarly

Confusion Matrix

- Showing the confusion matrix of the best performing model (KNN)
- All machine learning models performed similarly not much difference is noticed but slightly better accuracy is for KNN model



Conclusions

We can conclude that:

- The larger the flight number at , greater the success rate at a launch site.
- Launch success rate started to increasing in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the greatest success rate.
- KSC LC-39A become the most successful launches of any sites.
- All classifier models performed almost similar for this task.

Thank you!