IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Athari Khalid
03-Oct-2022

# Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- References

# Executive Summary

**Summary of methodologies**

- Data collection
- Data wrangling
- Data Analysis using SQL
- EDA with data visualization
- Visual Analytics with Folium
- Predictive analysis (Classification)

**Summary of all results**

- Interactive Visual Analytics
- Exploratory Data Analysis using SQL
- Predictive analysis using a machine learning model

# Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars, other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if you can accurately predict the likelihood of the first stage rocket landing successfully, you can determine the cost of a launch. With the help of your Data Science findings and models, the competing start-up you have been hired by can make more informed bids against SpaceX for a rocket launch.

- Problems you want to find answers
  - Determined the price of each launch.
  - Predict if the Falcon 9 first stage will land successfully.

Section 1

# Methodology

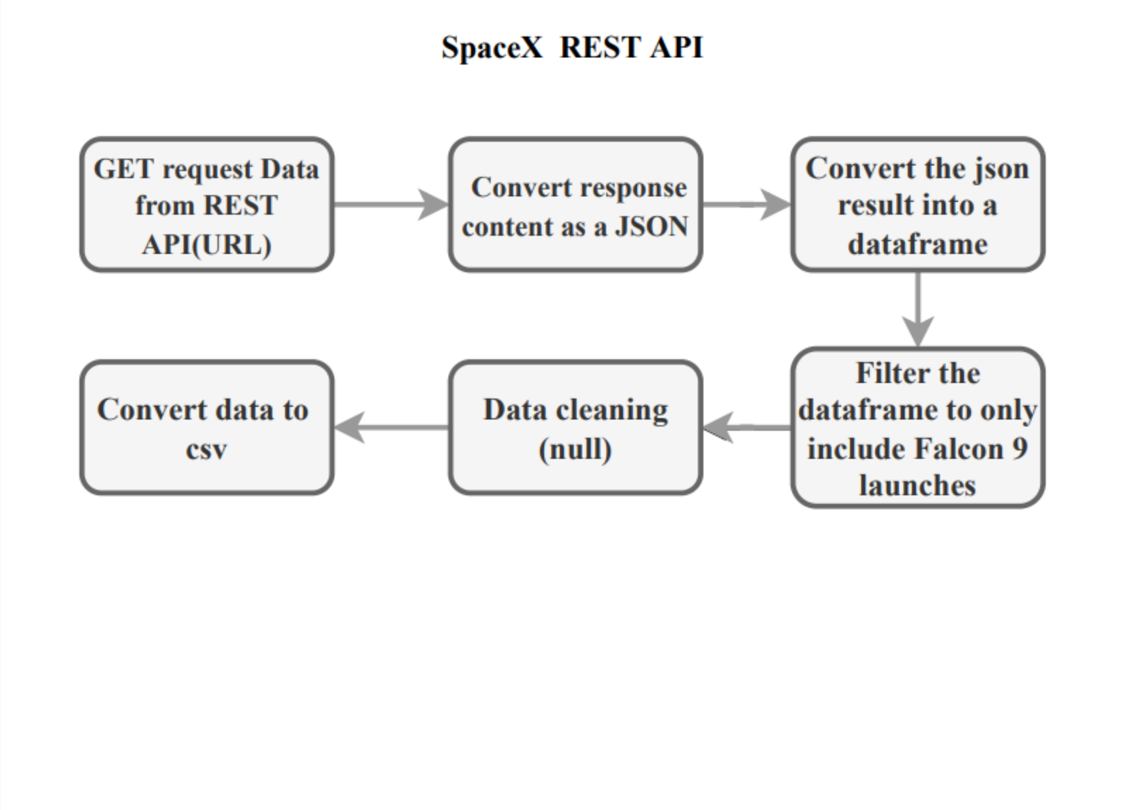
# Methodology

## Executive Summary

- Data collection methodology:
  - Rest API and Web Scraping
- Perform data wrangling
  - Data are clean of null values, and irrelevant columns and transformed into one Hot Encoding to be applied to Machine Learning.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use RL, KNN, SVM, DT models have been built and evaluated for the best classifier.

# Data Collection

- Data that is collected launch data from SpaceX  REST API.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.



# Data Collection – SpaceX API

### 1) Request and parse the SpaceX launch data using the GET request

```
In [11]: # Use json_normalize method to convert the json result into a dataframe
         import json
         from pandas import json_normalize

         response =  requests.get(static_json_url)
         df = response.json()
         data = json_normalize(df)
```

### 2) Filter the dataframe to only include `Falcon 9` launches

```
In [24]: # Hint data['BoosterVersion']!='Falcon 1'

         data_falcon9 = data_1[data_1['BoosterVersion'] == 'Falcon 9']
         data_falcon9
Out[24]:
```

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | LandingPad | Block | Reu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 6 | 2010-06-04 | Falcon 9 | NaN | LEO | CCSFS SLC 40 | None None | 1 | False | False | False | None | 1.0 | |
| 5 | 8 | 2012-05-22 | Falcon 9 | 525.0 | LEO | CCSFS SLC 40 | None None | 1 | False | False | False | None | 1.0 | |
| 6 | 10 | 2013-03-01 | Falcon 9 | 677.0 | ISS | CCSFS SLC 40 | None None | 1 | False | False | False | None | 1.0 | |
| 7 | 11 | 2013-09-29 | Falcon 9 | 500.0 | PO | VAFB SLC 4E | False Ocean | 1 | False | False | False | None | 1.0 | |

### 3) Dealing with Missing Values

```
In [28]: # Calculate the mean value of PayloadMass column
         Mean_PayloadMass = data_falcon9['PayloadMass'].mean()
         # Replace the np.nan values with its mean value
         data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean_PayloadMass)
```

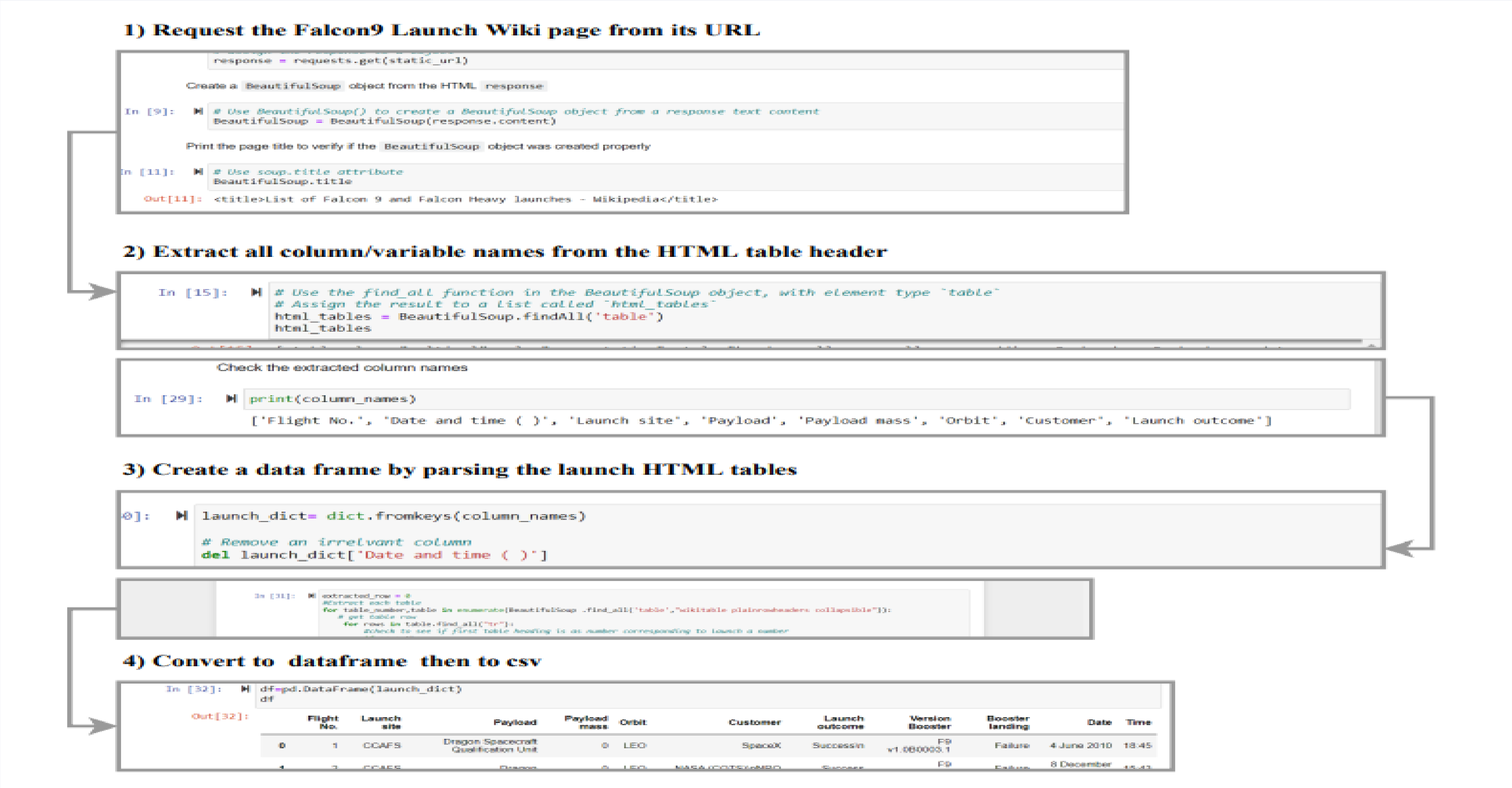
### 4) Convert to CSV

We can now export it to a **CSV** for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

```
In [31]: data_falcon9.to_csv('1.csv', index=False)
```

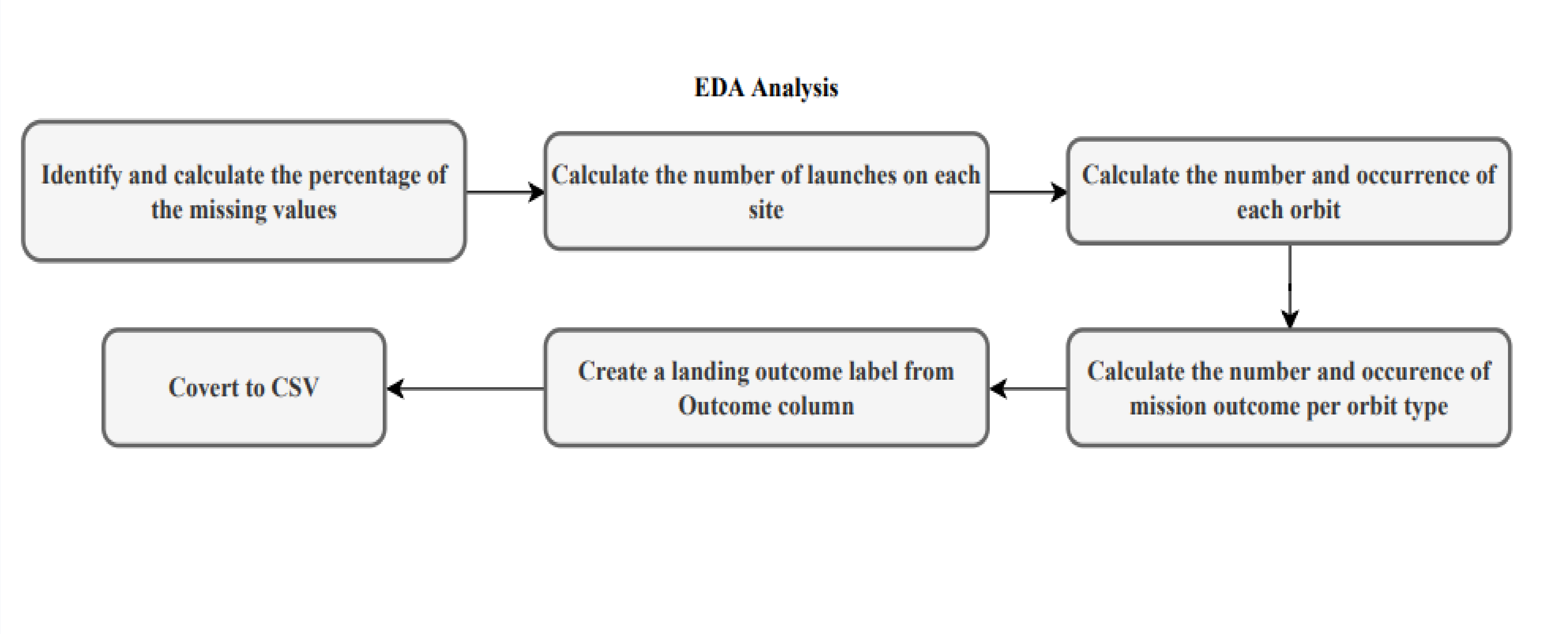
**Github**: Data_collection

# Data Collection - Scraping



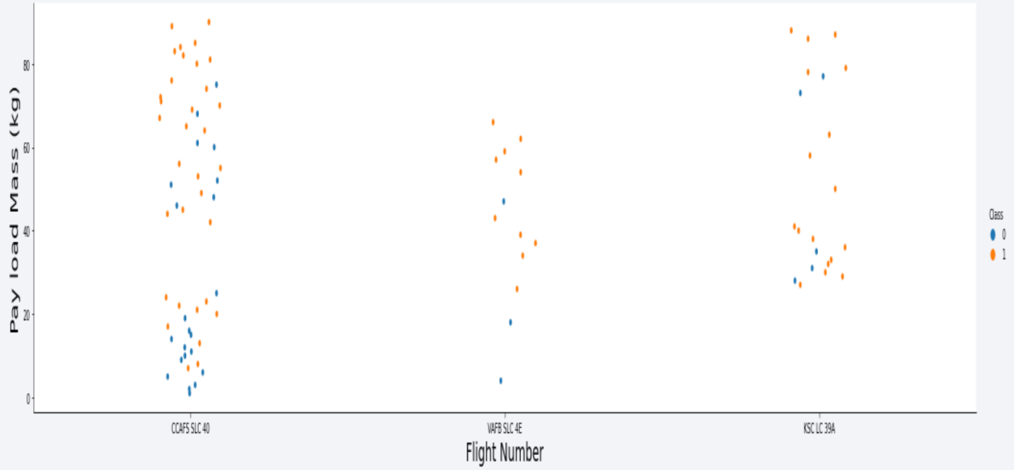

**Github**: [Data_scraping]

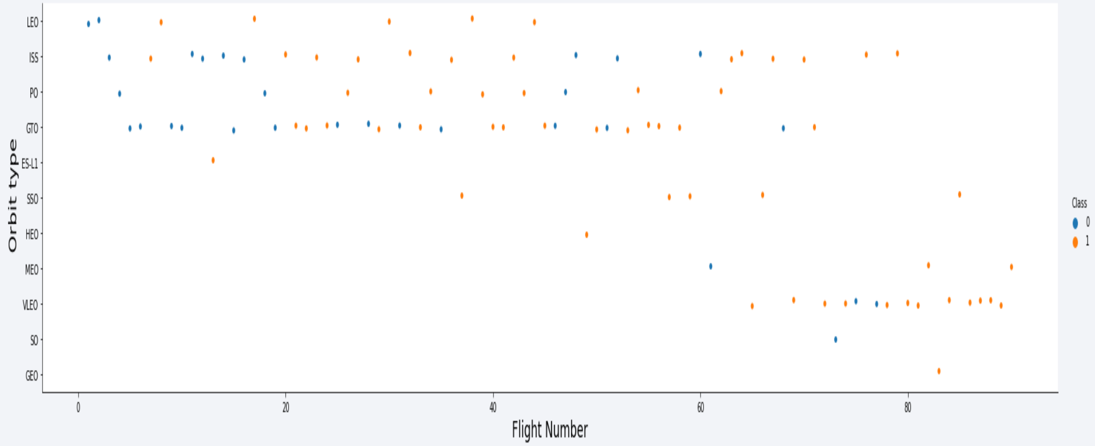
# Data Wrangling

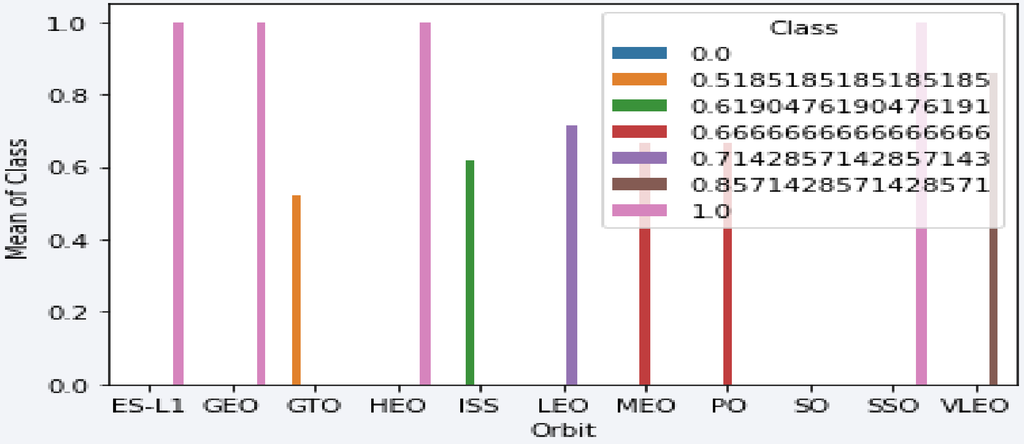


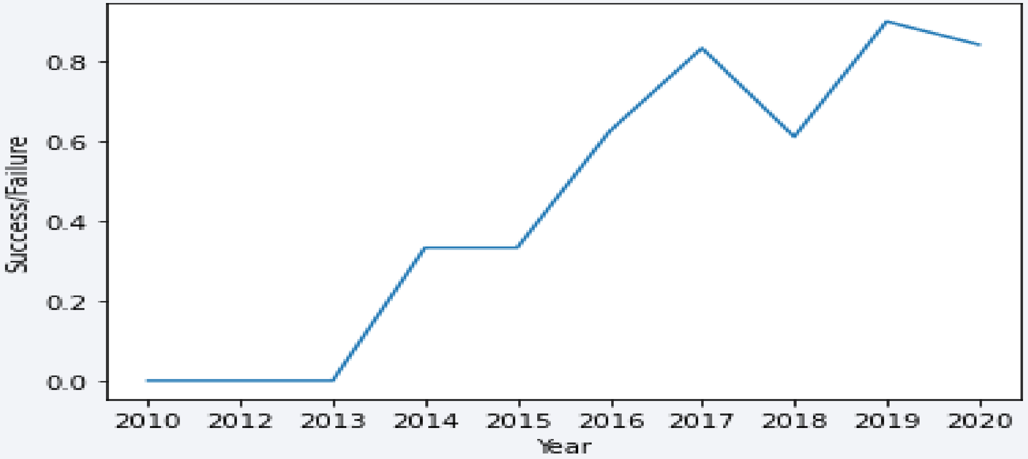

Github:[Data_Wrangling]

# EDA with Data Visualization






**Github**: Data_Visualization

# EDA with SQL

## SQL queries you performed include:

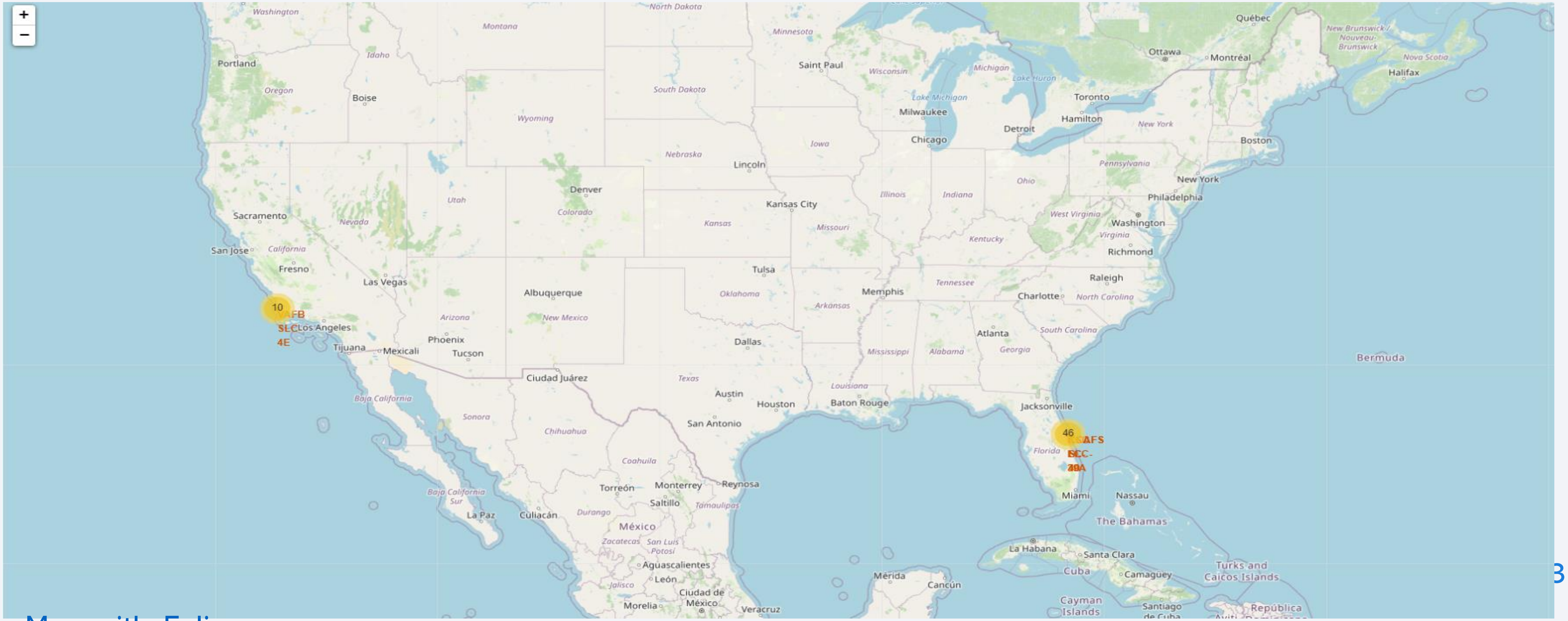
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015¶
- 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



**Github**: EDA_SQL

# Build an Interactive Map with Folium

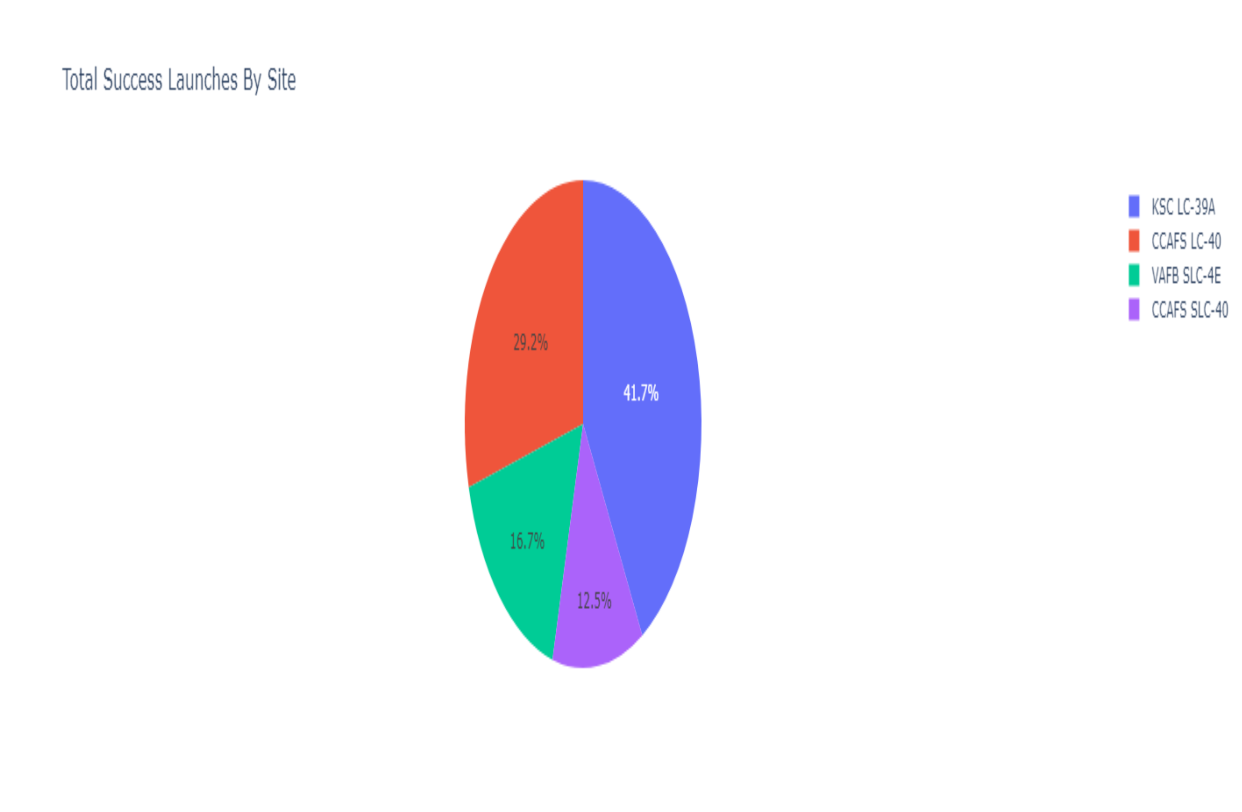
By using Marker Cluster, I'm able to add the markers having the same coordinate for launch outcomes to each site.

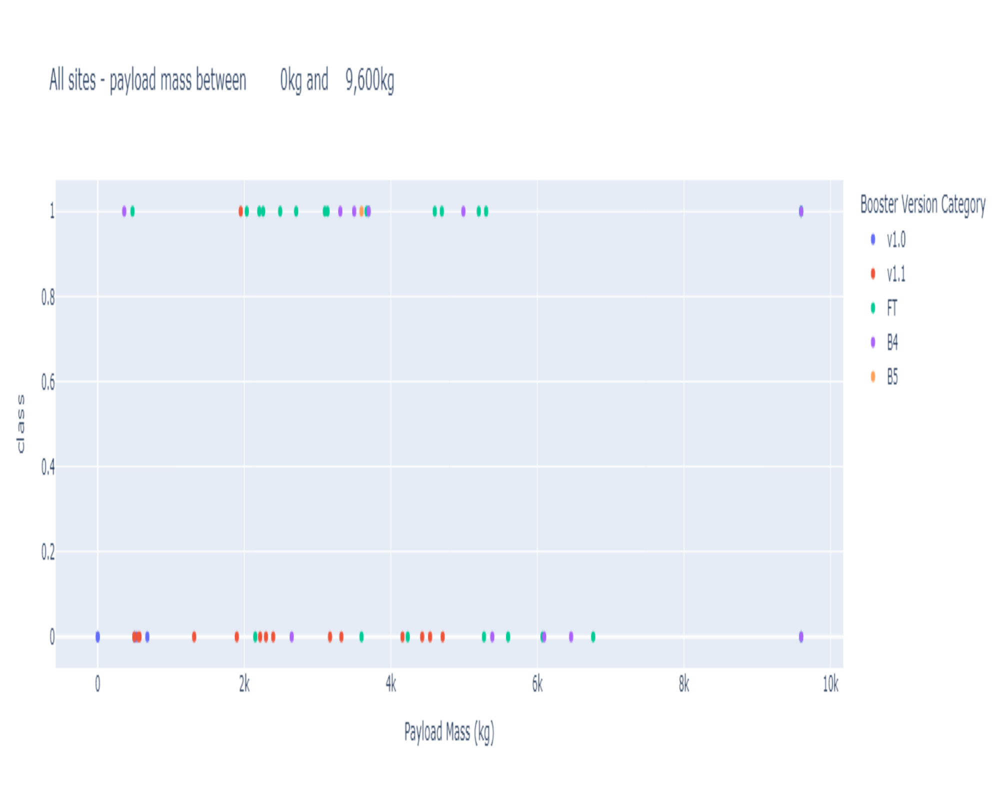





**Github**: [Map_with_Folium]

# Build a Dashboard with Plotly Dash



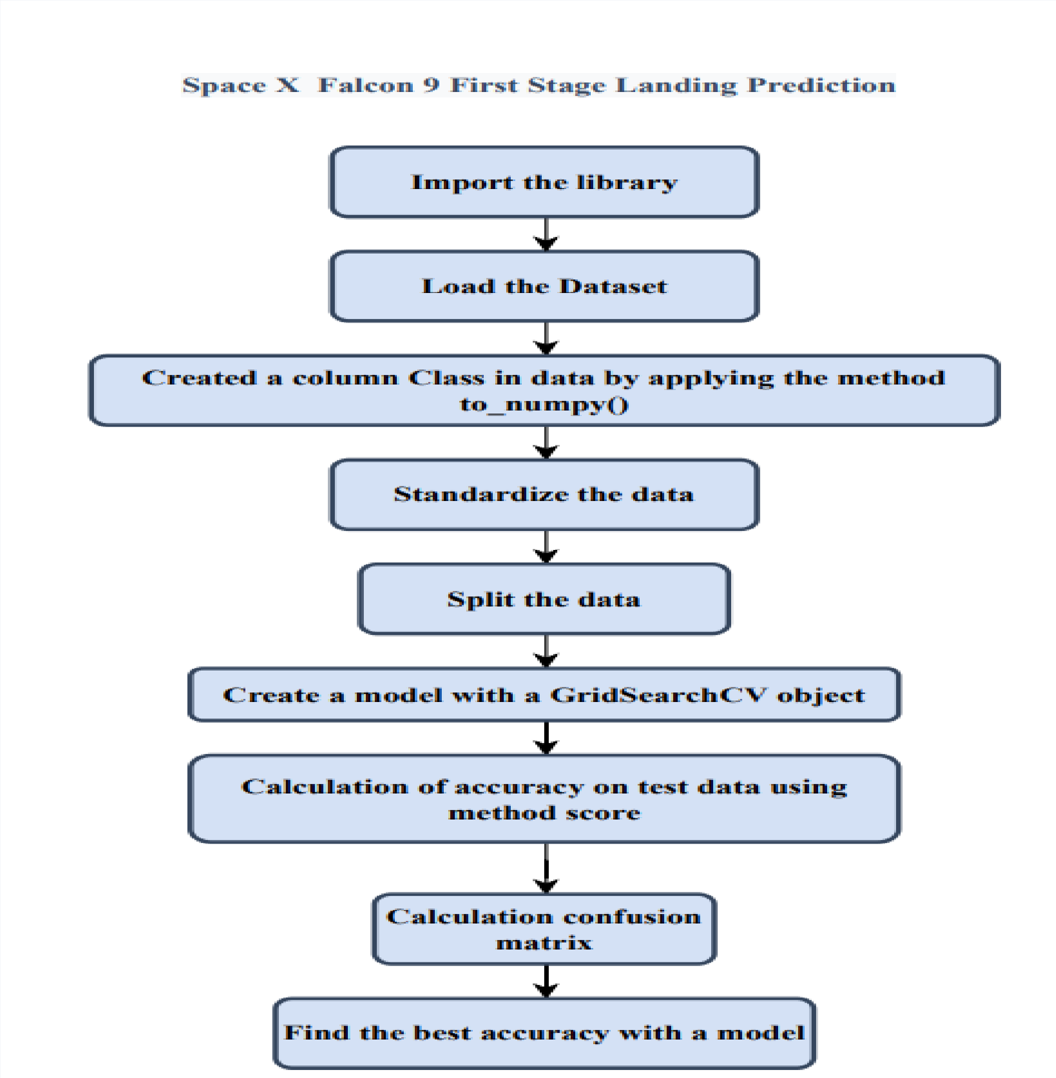






**Github**: [Dashboard_with_Plotly]

# Predictive Analysis (Classification)

- Load the Dataset
- Standardize the data
- Split the data into training and testing data
- Create a model with a GridSearchCV object
- Calculation of accuracy on test data using method score
- Calculation confusion matrix
- Find the best accuracy with a model




**Github**: [Predictive_Analysis]

# Results

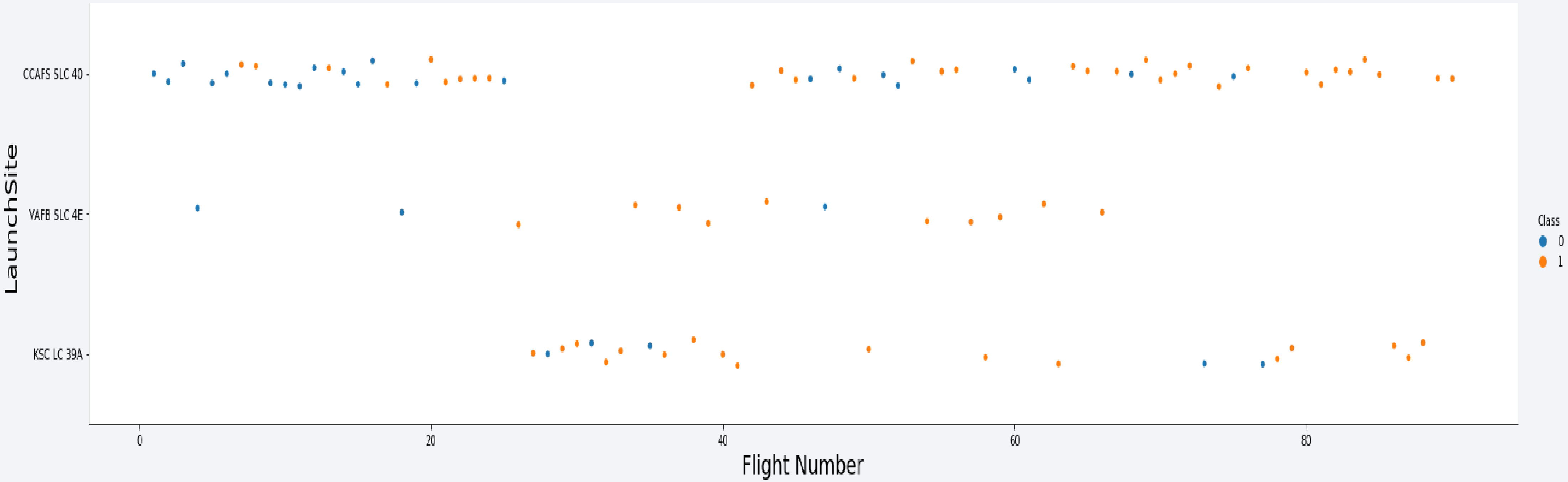
- After analysis data the CCAFS SLC-40 site and KSC LC-39A site are has most successful launches from all the sites.
- Orbit GEO,HEO,SSOES L1 has the best Success Rate.
- Decision Tree  model are the best in terms of prediction accuracy for this dataset.

Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site

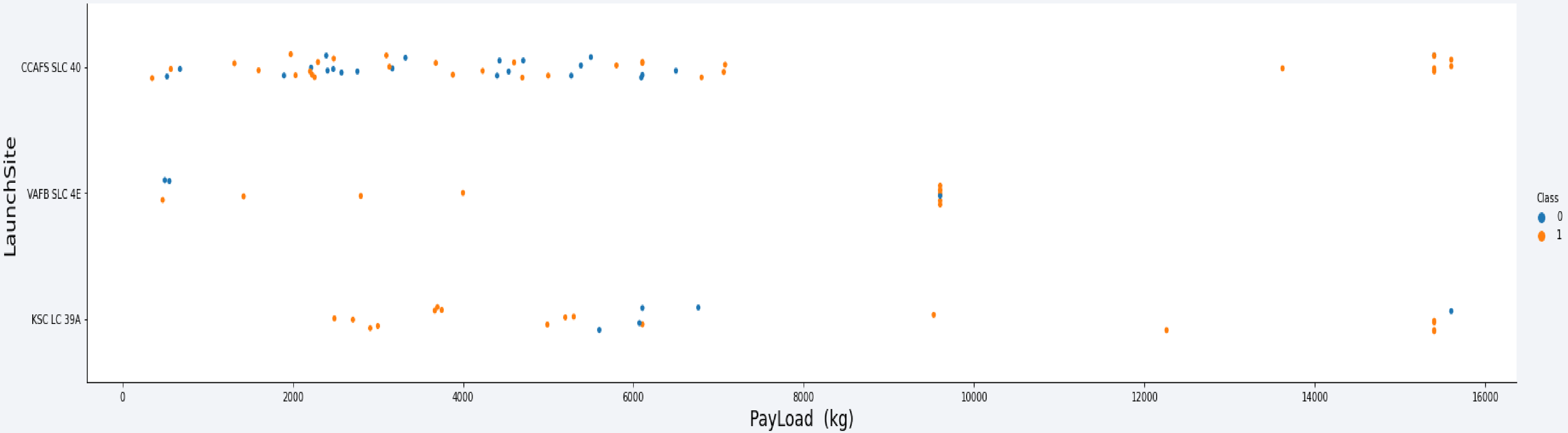
**Visualize the relationship between Flight Number and Launch Site**
We see that different launch sites have different success rates. But as we increase the number of flights the success rate increase.



# Payload vs. Launch Site

**Visualize the relationship between Payload and Launch Site**

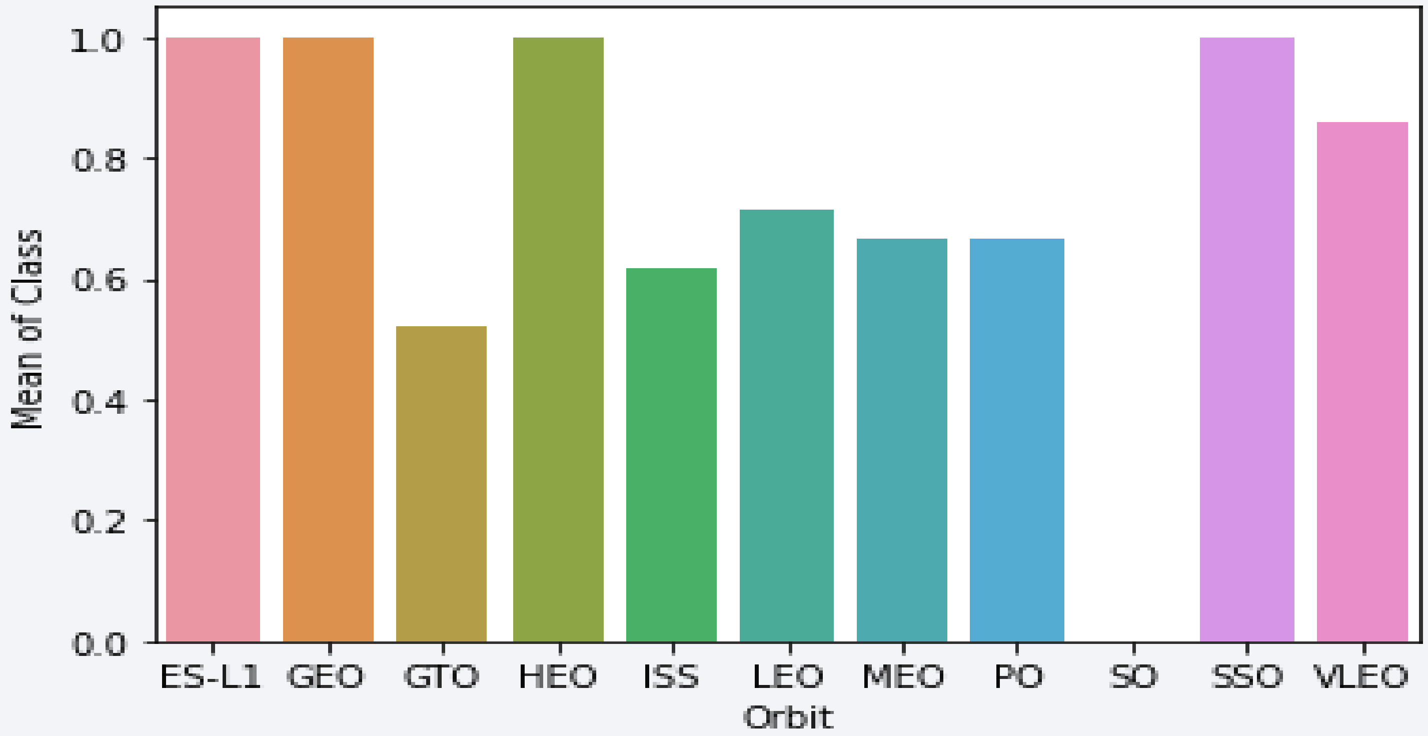
As well, if increase the number of we Pay Load Mass (kg) the success rate increase.



# Success Rate vs. Orbit Type

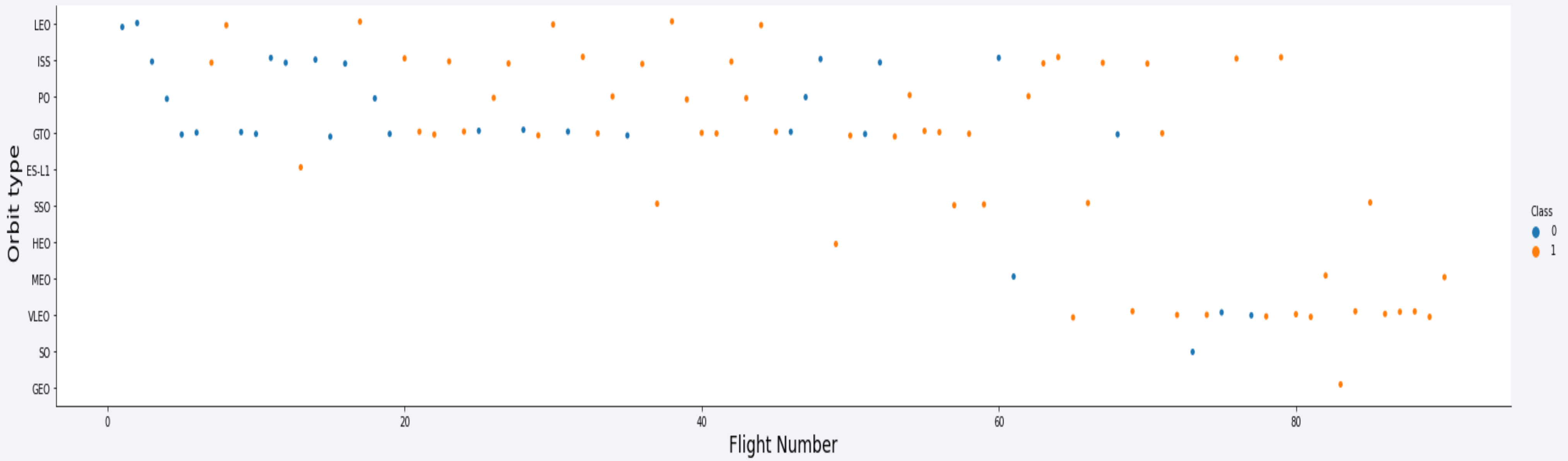
**Visualize the relationship between success rate of each orbit type**

- AS we can see ES-L1 , GEO, HEO and SSO have a success rates 100%.



# Flight Number vs. Orbit Type

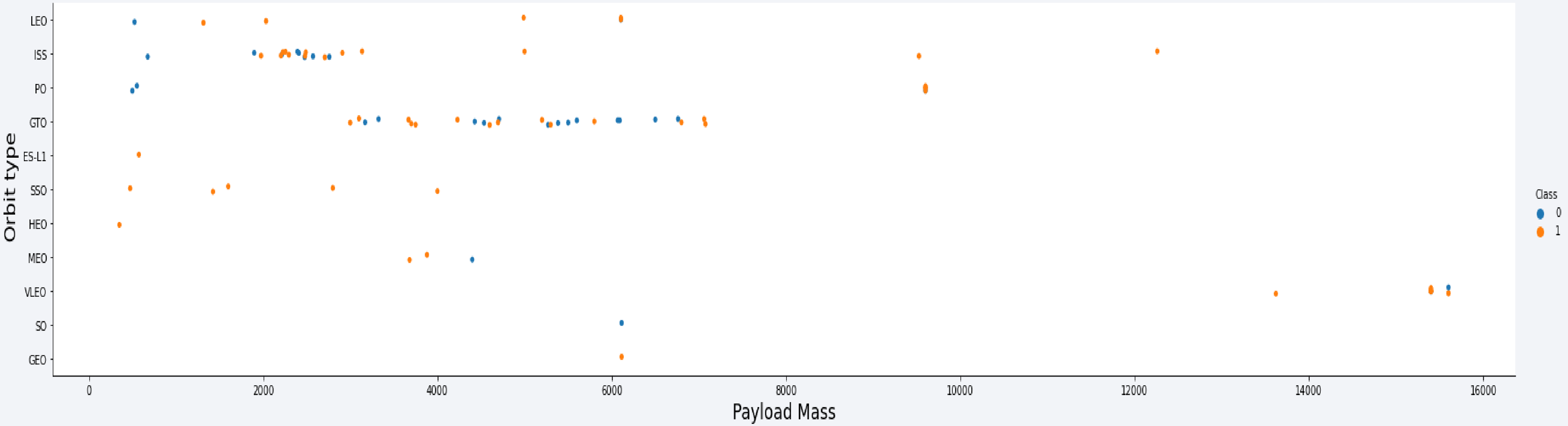
**Visualize the relationship between FlightNumber and Orbit type**



# Payload vs. Orbit Type

**Visualize the relationship between Payload and Orbit type**

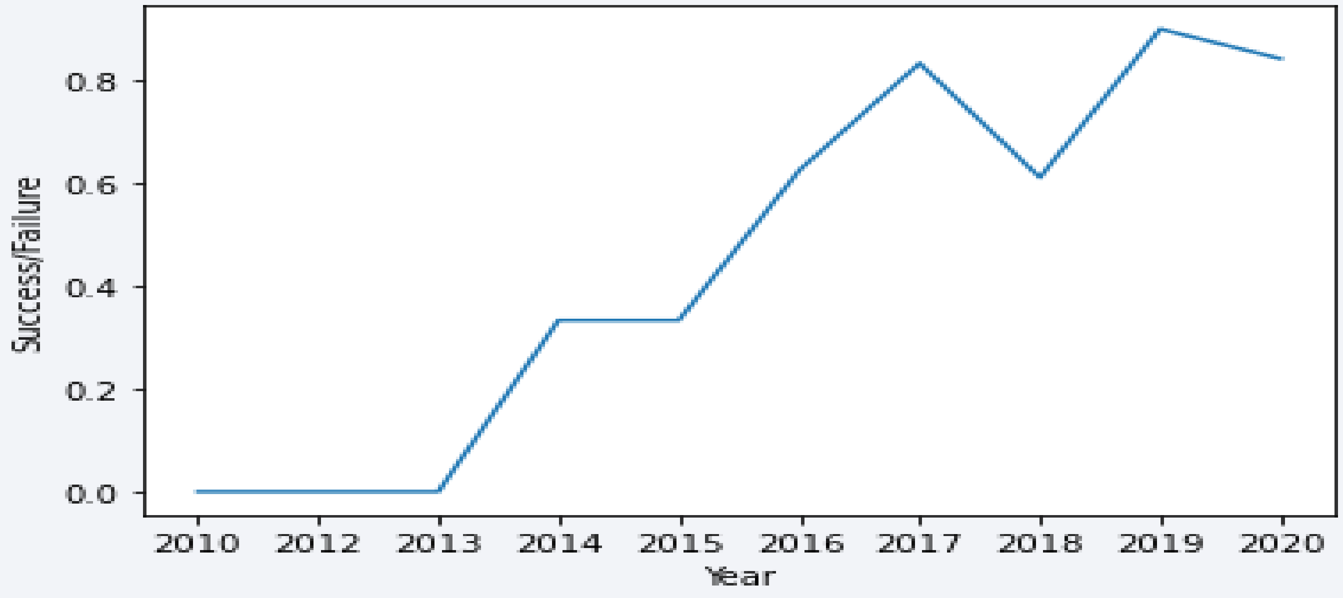
There is a connection between ISS and  payload in the range of 2000 to 3000. Also between GTE and Payload  at 4000 to 8000.



# Launch Success Yearly Trend

**Visualize the launch success yearly trend**

We can note in the figure that the launch sites begin to increase in success rate from 2013 to approximately 2018, decreases slightly, and then returns to increase with the passage of the year



# All Launch Site Names

- %sql SELECT DISTINCT(launch_site) FROM Spacex

| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS SLC-40 |
| KSC LC-39A |
| VAFB SLC-4E |

# Launch Site Names Begin with 'CCA'

- %sql SELECT * FROM Spacex WHERE launch_site LIKE'CCA%' LIMIT 5

| 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) |
| 2012-08-10 | 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 |
| 2013-03-12 | 22:41:00 | F9 v1.1 | CCAFS LC-40 | SES-8 | 3170 | GTO | SES | Success | No attempt |

# Total Payload Mass

- %sql SELECT SUM(payload_mass__kg_) FROM Spacex WHERE customer='NASA (CRS)'

| 1 |
| --- |
| 45596 |

# Average Payload Mass by F9 v1.1

- %sql SELECT AVG(payload_mass__kg_) FROM Spacex  WHERE booster_version='F9 v1.1'

| 1 |
| --- |
| 2928 |

# First Successful Ground Landing Date

- %sql SELECT MIN(DATE) FROM Spacex WHERE landing__outcome='Success (ground pad)'

| 1 |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql SELECT booster_version FROM Spacex  WHERE landing__outcome='Success (drone ship)' AND payload_mass__kg_ BETWEEN 4000 AND 6000

| booster_version |
| --- |
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

# Total Number of Successful and Failure Mission Outcomes

- %sql SELECT COUNT(MISSION_OUTCOME) AS missionoutcomes FROM Spacex  WHERE mission_outcome LIKE 'Success%'

| missionoutcomes |
|---|
| 100 |

- %sql SELECT COUNT(MISSION_OUTCOME) AS missionoutcomes FROM Spacex  WHERE mission_outcome LIKE 'Failure

| missionoutcomes |
|---|
| 1 |

# Boosters Carried Maximum Payload

- %sql SELECT booster_version AS Maxboosterversion FROM Spacex WHERE payload_mass__kg_=(SELECT MAX(payload_mass__kg_) FROM Spacex)

| maxboosterversion |
|---|
| 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 landing__outcome,booster_version,launch_site,DATE FROM Spacex WHERE landing__outcome='Failure (drone ship)' AND EXTRACT(YEAR FROM DATE)='2015'

| landing__outcome | booster_version | launch_site | DATE |
|---|---|---|---|
| Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 | 2015-10-01 |
| Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 | 2015-04-14 |

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

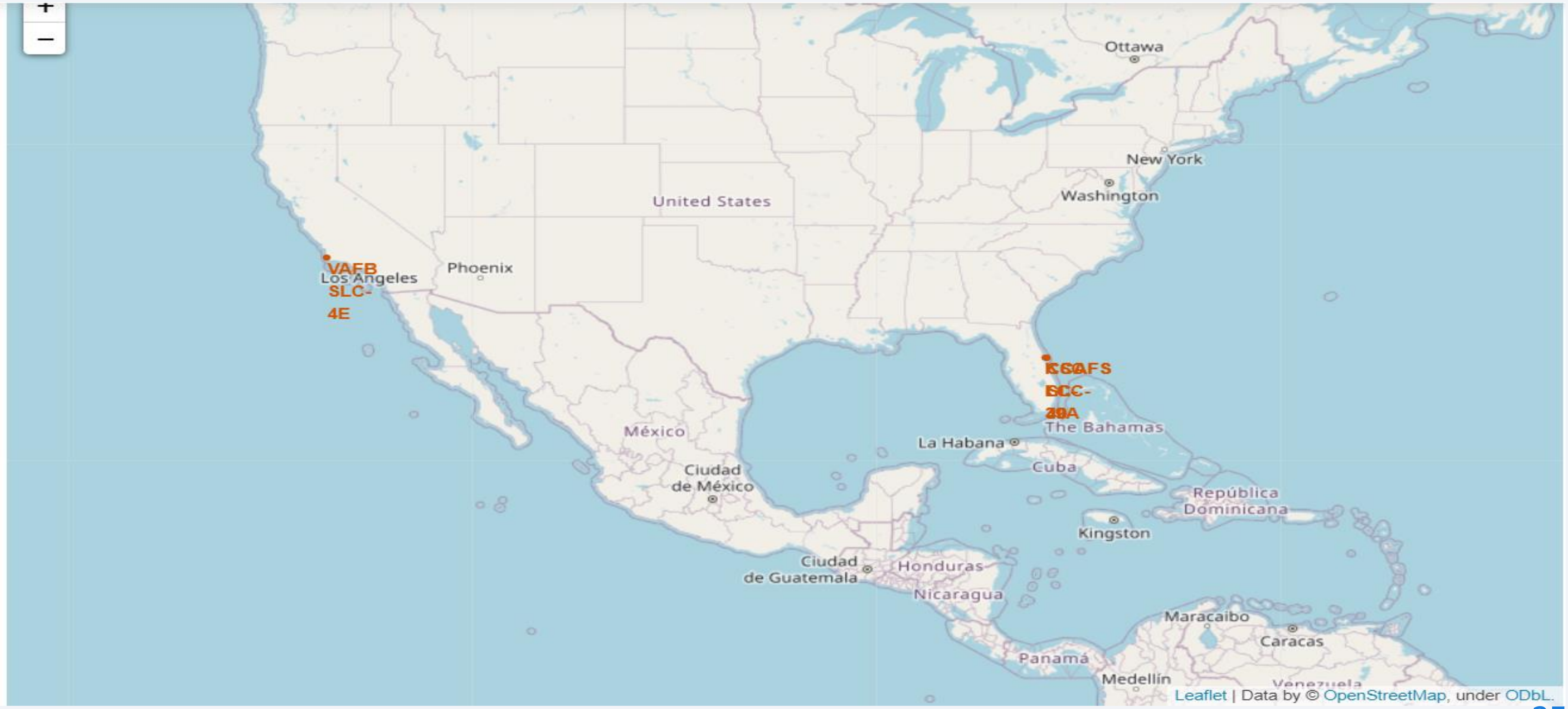
- %sql SELECT landing__outcome , COUNT(landing__outcome) FROM Spacex WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY COUNT(landing__outcome) DESC

| landing__outcome | 2 |
|---:|---:|
| No attempt | 10 |
| Success (drone ship) | 6 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 5 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 1 |
| Precluded (drone ship) | 1 |

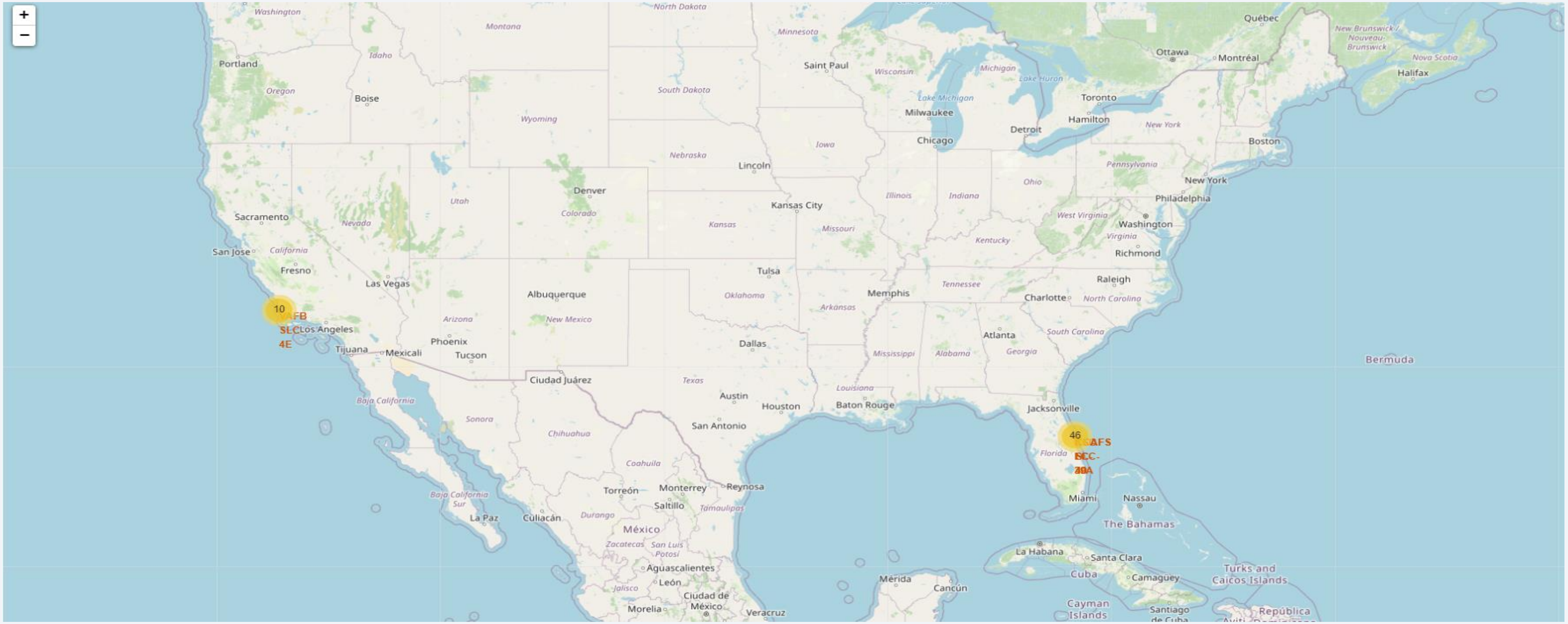
Section 3

# Launch Sites Proximities Analysis

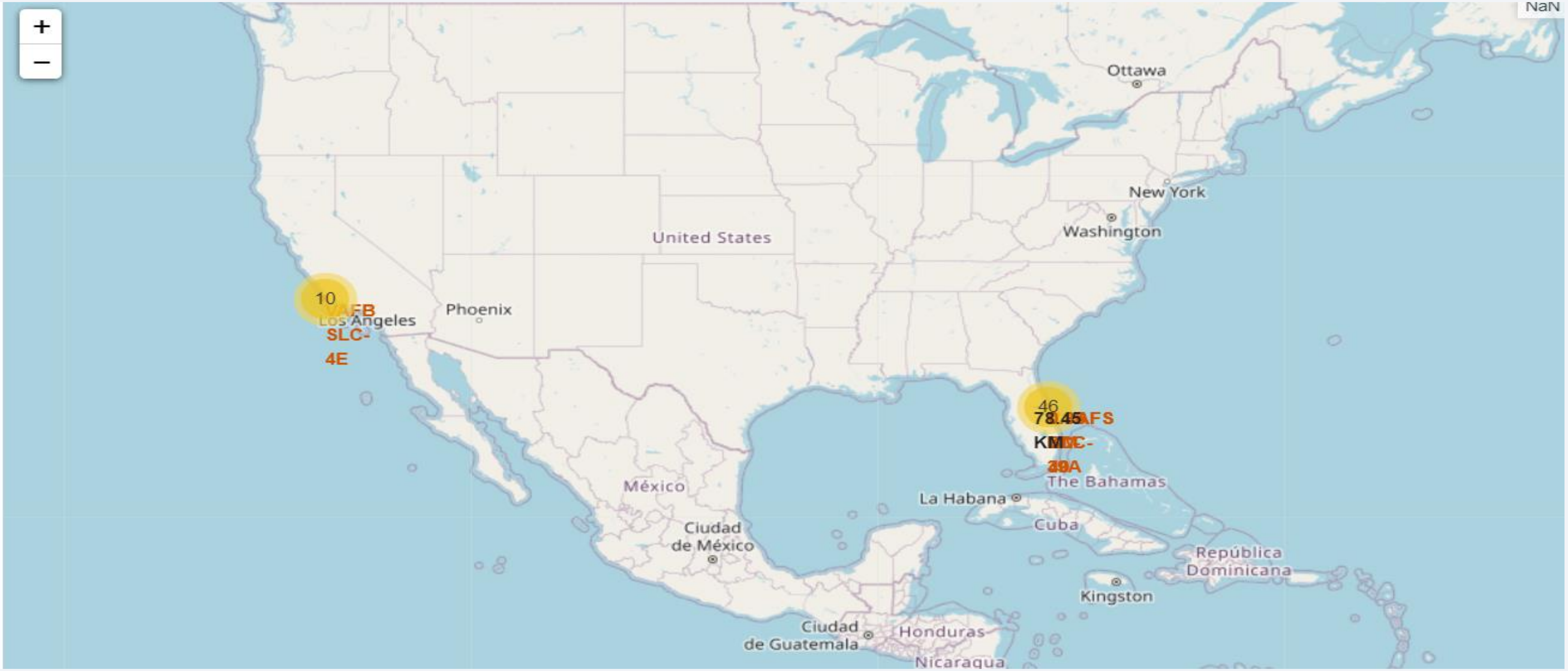
# All launch sites on the site map



# launch sites by using Marker Cluster



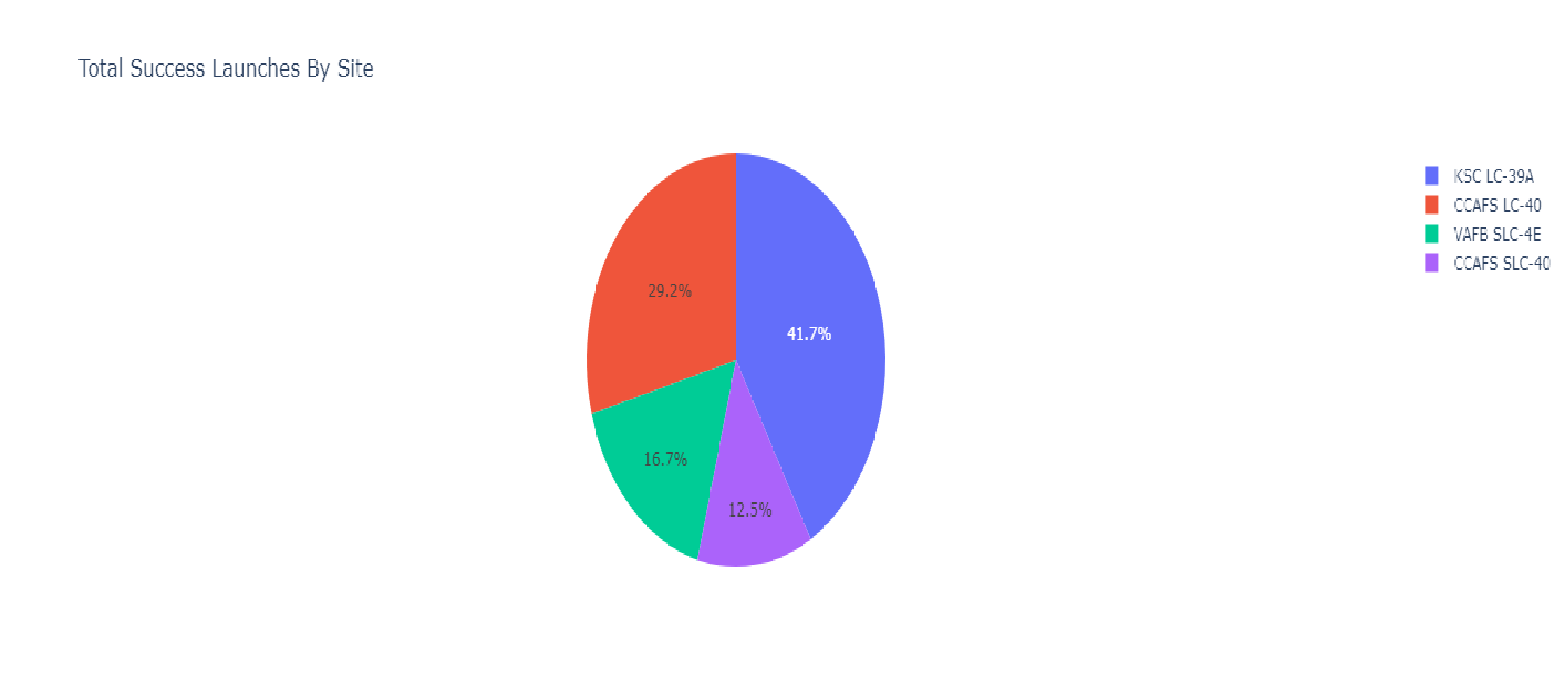
# Marker distance between the coastline point and the launch site



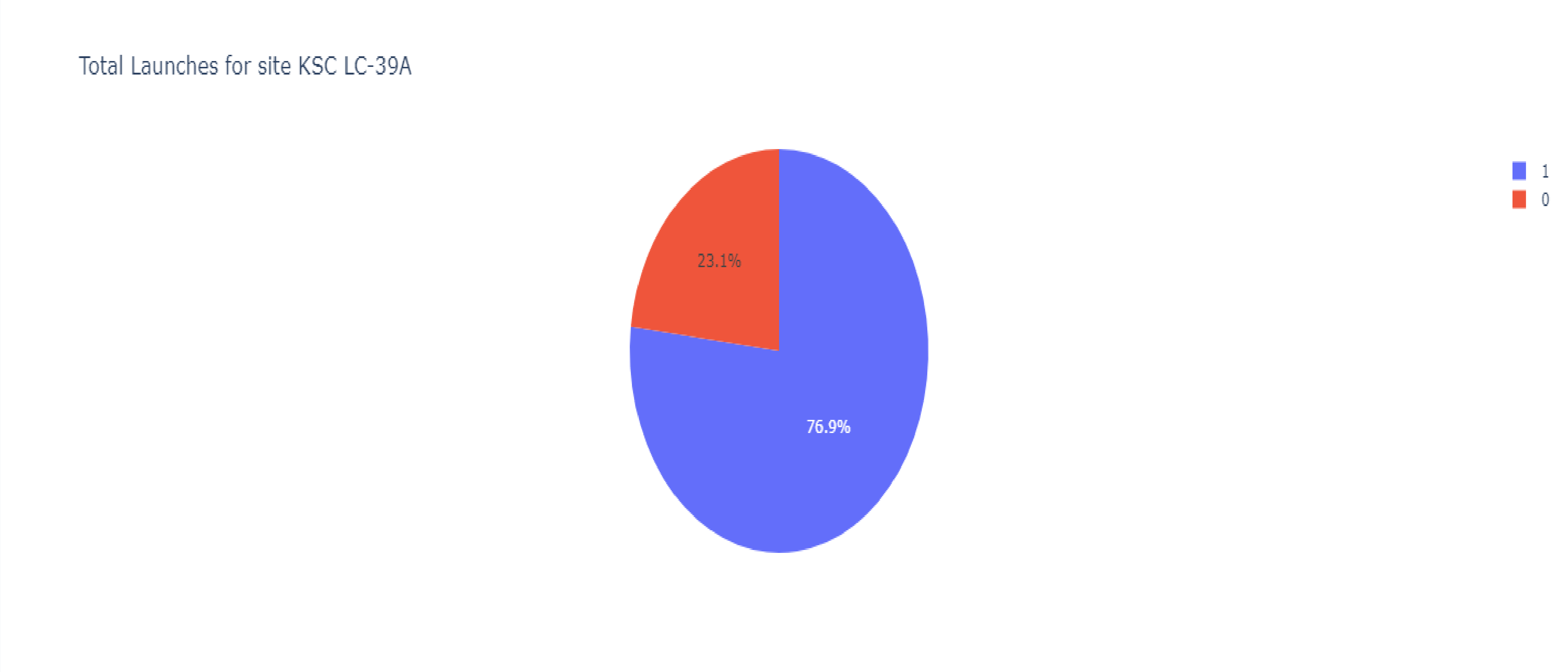
Section 4

# Build a Dashboard with Plotly Dash

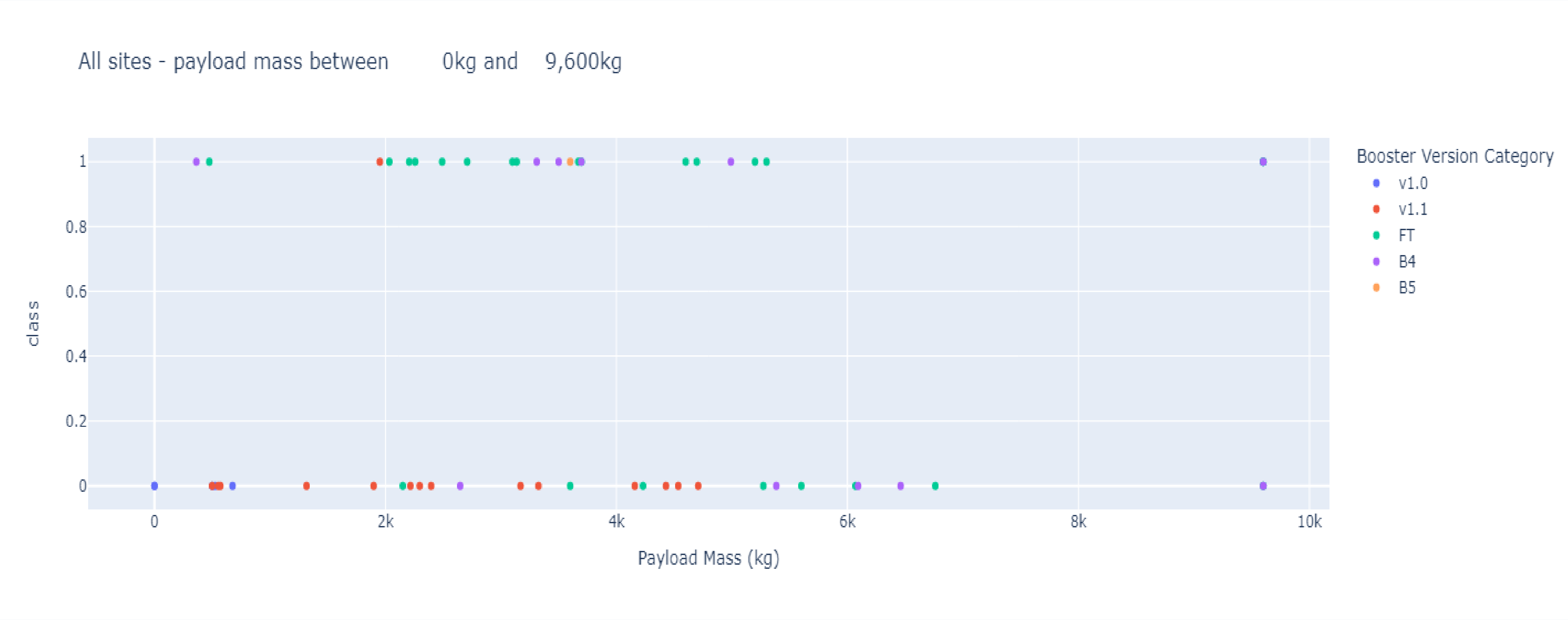
# Total Success Launches



# Launches for Hight site Score



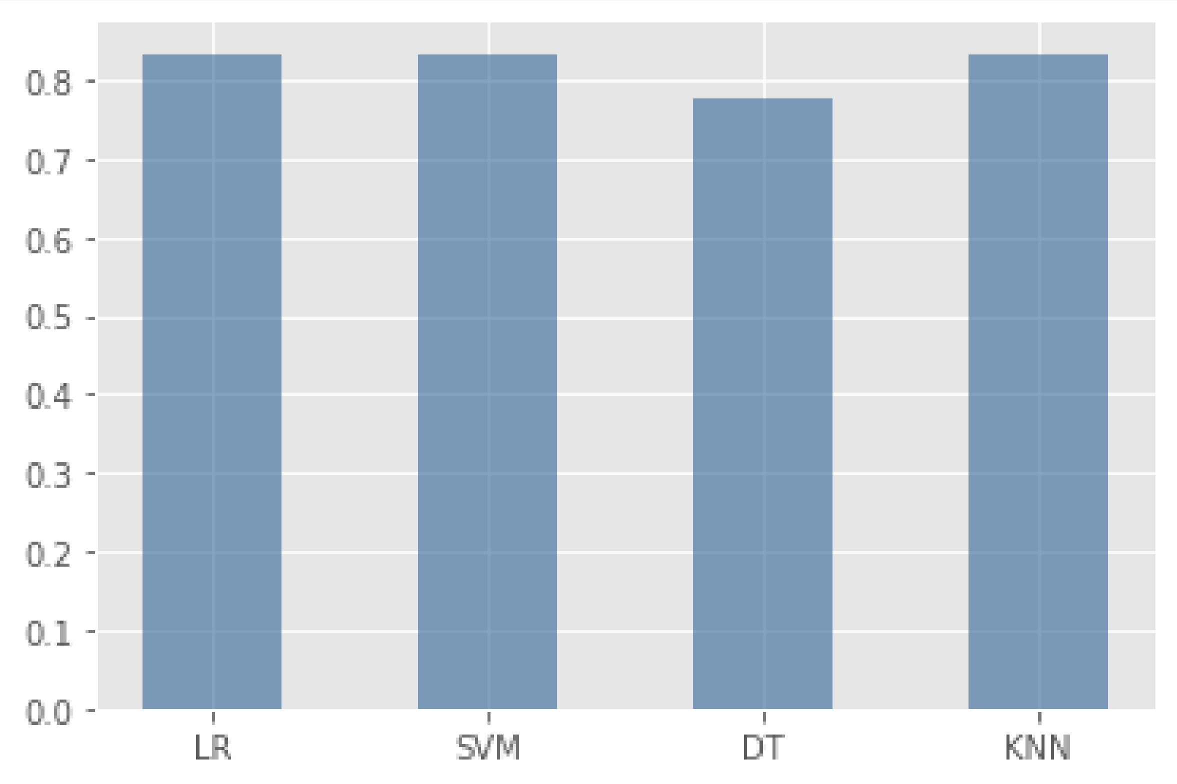
# Payload vs. All Launch



Section 5

# Predictive Analysis (Classification)

# Classification Accuracy




As we can see Decision Tree has the highest accuracy with almost 0.89, then comes the remaining models with almost the same accuracy of 0.84.

# Confusion Matrix


| Measure | Derivations | Result |
|---|---|---|
| **Precision** | PPV = TP / (TP + FP) | 0.67 |
| **Accuracy** | ACC = (TP + TN) / (P + N) | 0.89 |
| **F1 Score** | F1 = 2TP / (2TP + FP + FN) | 0.80 |

# Conclusions

- After analysis data the CCAFS SLC-40 site and KSC LC-39A site are has most successful launches from all the sites.
- Orbit GEO,HEO,SSOES L1 has the best Success Rate.
- The payload of O kg to 5000 kg was more diverse than 6000 kg to 10000
- The Decision Tree model is the best in terms of prediction accuracy for this dataset.

# References


- Confusion Matrix :https://onlineconfusionmatrix.com/
- Matplotlib - Bar Plot : https://www.tutorialspoint.com/matplotlib/matplotlib_bar_plot.htm
- https://towardsdatascience.com/7-points-to-create-better-histograms-with-seaborn-5fb542763169
- https://www.researchgate.net/figure/Bar-chart-showing-the-performance-evaluation-in-our-data-loading-tests_fig4_268150621

# Thank you!