IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Nguyen Nhat Minh
30/06/2023

# Outline

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

# Executive Summary

- Summary of methodologies
- Summary of all results

# Introduction

- 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.
- With the background above, the main idea for this project is to predict how much does the cost we need for successful landing in first stage. It can also used to compare with other competitive companies.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - The data was collected from 2 separated websites
- Perform data wrangling
  - Transform from categorical data to numerical data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using traditional machine learning models along with GridSearchCV to detect the best parameters for the best accuracy score

# Data Collection

- For SpaceX API, we collect the data from links:

+) https://api.spacexdata.com/v4/rockets/

+) https://api.spacexdata.com/v4/launchpads/

+) https://api.spacexdata.com/v4/payloads/

+) https://api.spacexdata.com/v4/cores/

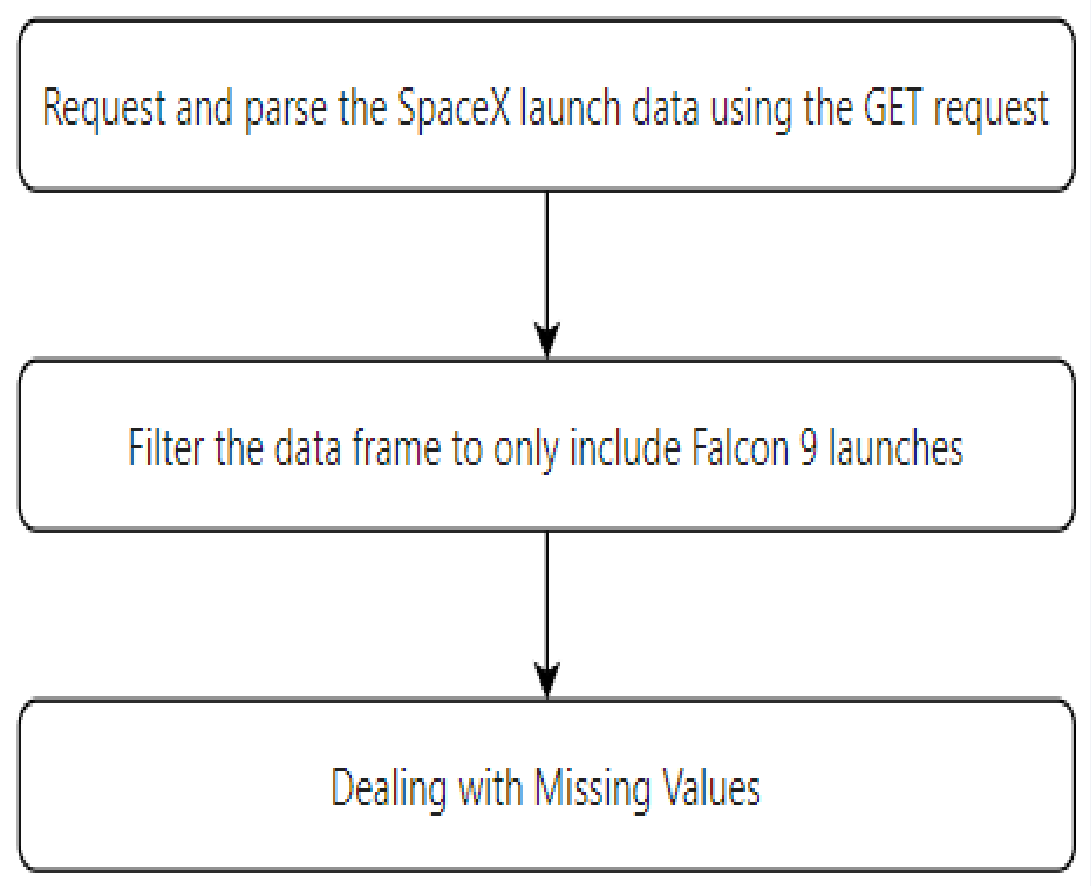
+) https://api.spacexdata.com/v4/launches/past

- For Web Scraping, we collect the data from the link:

+) https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

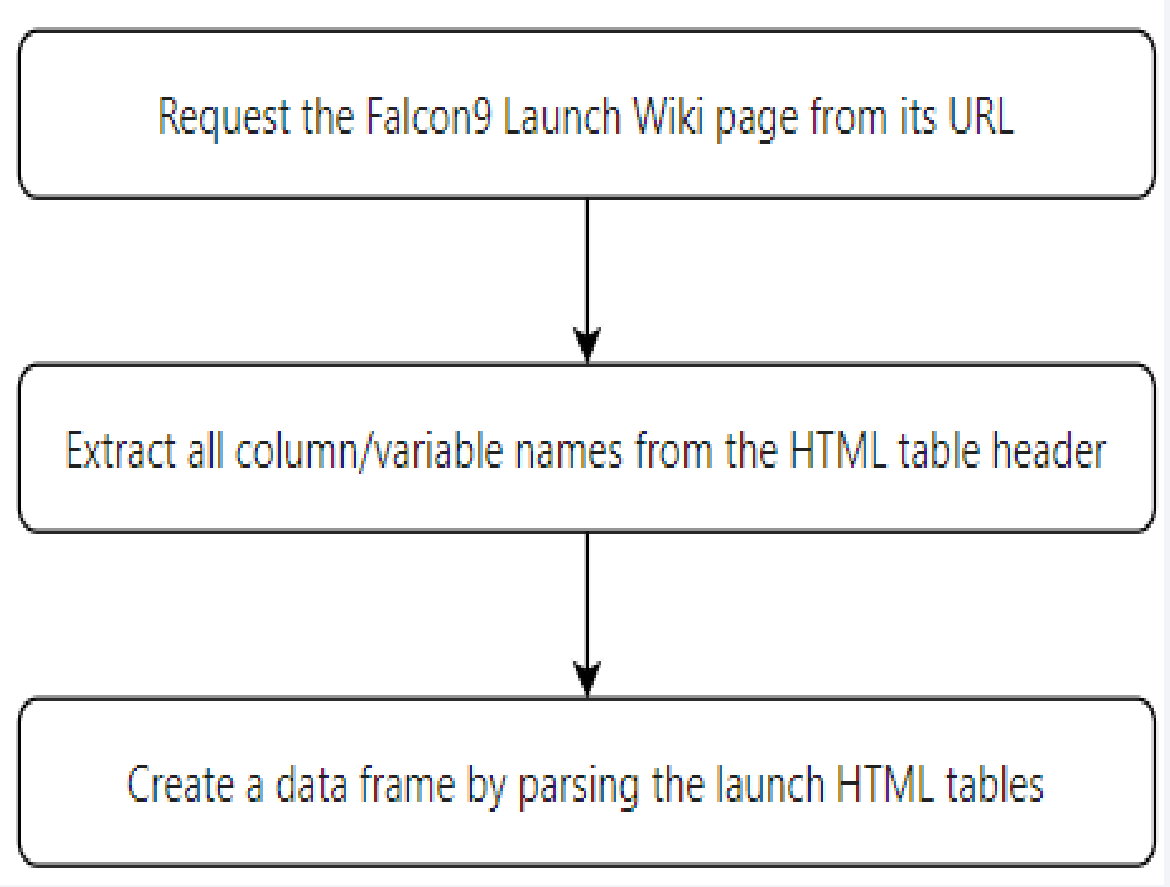
# Data Collection – SpaceX API

- Since the format for SPACEX API is json, we will convert to the data frame by using pd.json_normalize() with the response has an attribute json().
- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_1_1_3.ipynb



# Data Collection - Scraping

- We get the request from the link with attribute text and view the html structure with BeautifulSoup to discover the table that we need to convert it to the data frame.
- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_1_2_3.ipynb



# Data Wrangling

- Connected with the dataset that we did from SPACEX API, after handling missing value for numerical values, we will continue performing data wrangling with categorical values.
- We also add a column for binary classification to determine that the landing end up successfully or failure (with 0 as failure and 1 as successful)
- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_1_3_3.ipynb

# EDA with Data Visualization

- Barplot: Success rate vs orbit type
- Catplot:

+) Flight number vs launch site

+) Payload vs launch site

+) Flight number vs orbit type

+) Payload vs orbit type

- Line plot: Launch success yearly trend
- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_2_2_2.ipynb

# EDA with SQL

- 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 succesful 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 records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months 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.

- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_2_1_2.ipynb

# Build an Interactive Map with Folium

- Markers and circles for locating the place where launch site occur and marking the success/failed launches for each site on the map.
- Lines for calculating distances between launch site to its proximities
- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_3_1_2.ipynb

# Build a Dashboard with Plotly Dash

- Circle plot for calculating percentage for successful landing for all sites. And for singular sites, circle plots consist the percentage of success and fail landing.
- Scatter plot for visualizing relationship between payload mass and class to determine success or fail landing, categorized by booster version
- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_3_2_2.ipynb

# Predictive Analysis (Classification)

- After splitting train and the test data, we train the model with 4 traditional machine learning for classification, along with Grid Search CV to find the best parameters for the model for the best accuracy score:

+) K Nearest Neighbour

+) Decision Tree

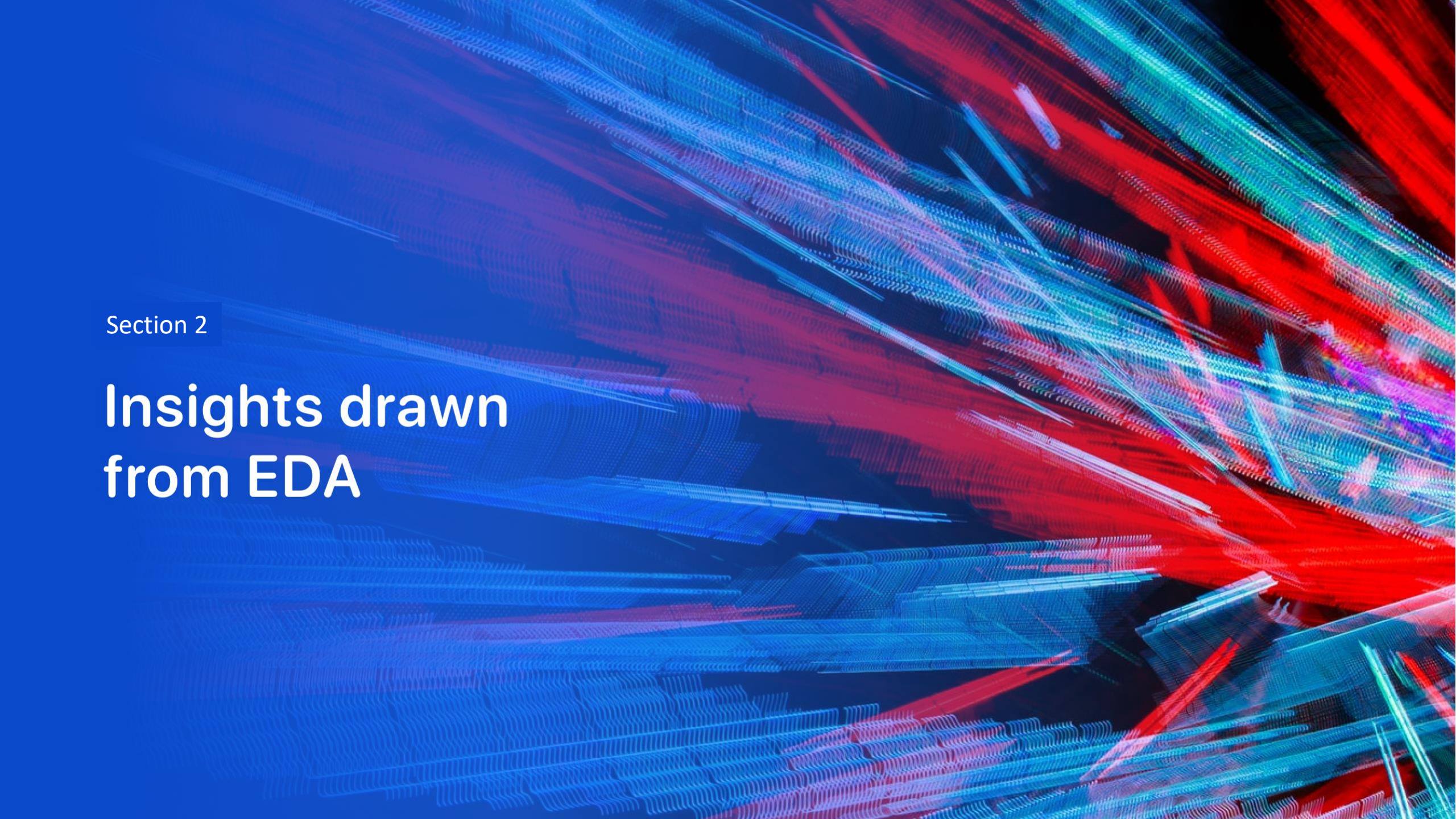
+) Logistic Regression

+) Random Forest

- https://github.com/FPTU12345/Coursera/blob/main/MOOC4_Week_4.ipynb

# Results

- All 4 machine learning models share the same accuracy score (83.33%)

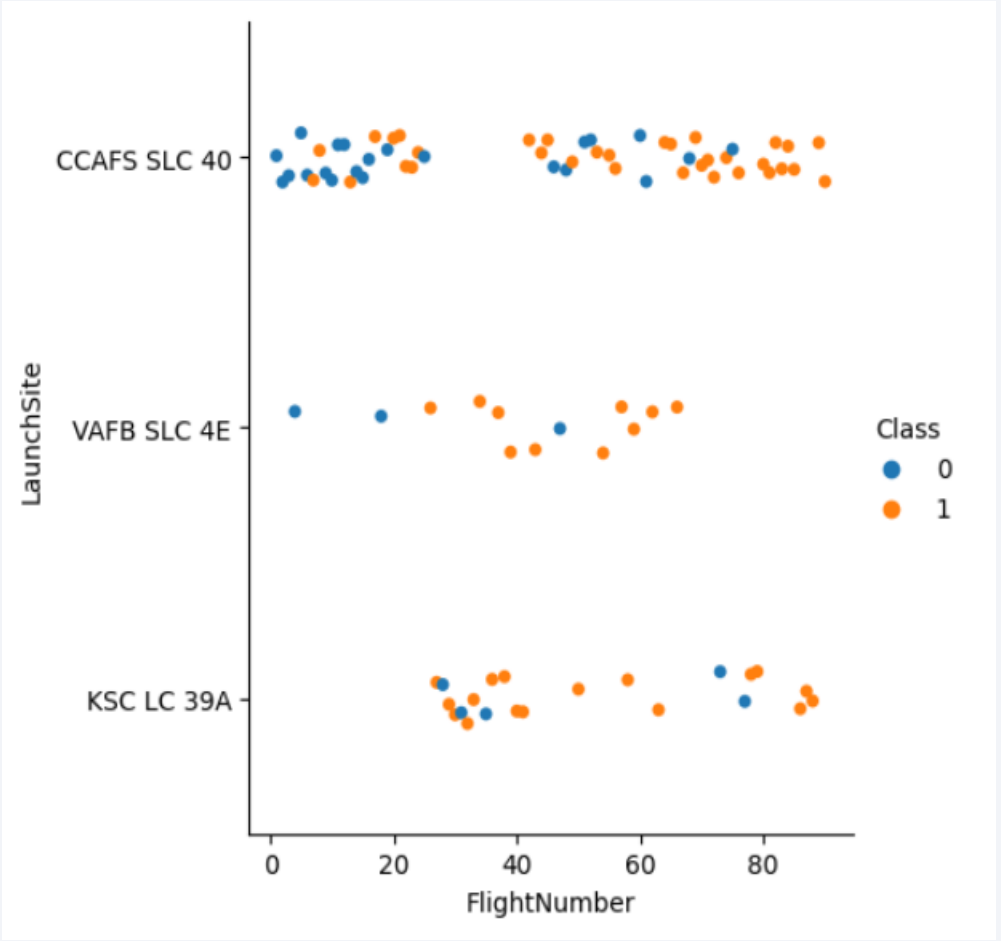

Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site

According to the chart, the number of flights for each launch site:

| | LaunchSite |
|---|---|
| CCAFS SLC 40 | 55 |
| KSC LC 39A | 22 |
| VAFB SLC 4E | 13 |

In every Launch Site, the more the flight, the higher the successful landing.



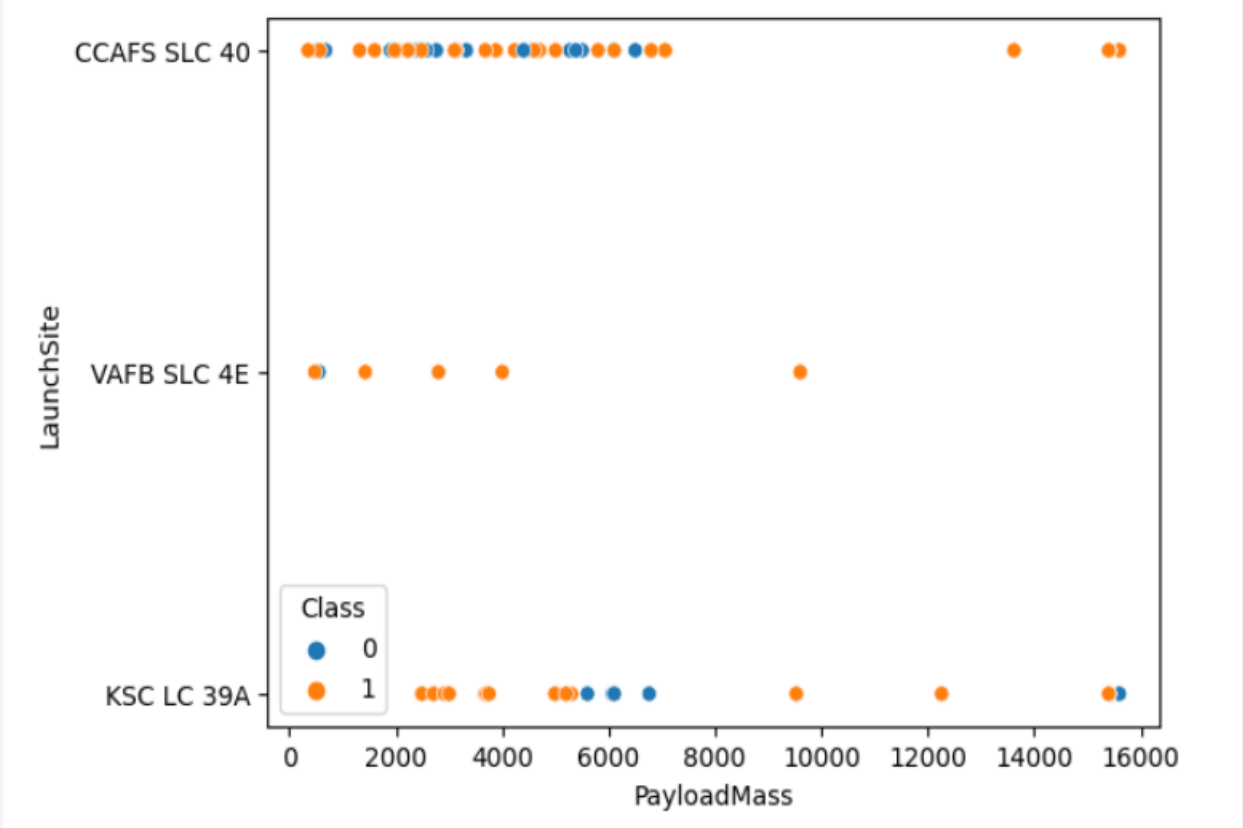
# Payload vs. Launch Site

According to the chart:

+) VAFB SLC 4E: The successful landing is guaranteed when Payload is up to 1000.

+) CCAFS SLC 40: The successful landing is guaranteed when Payload is up to 7000.

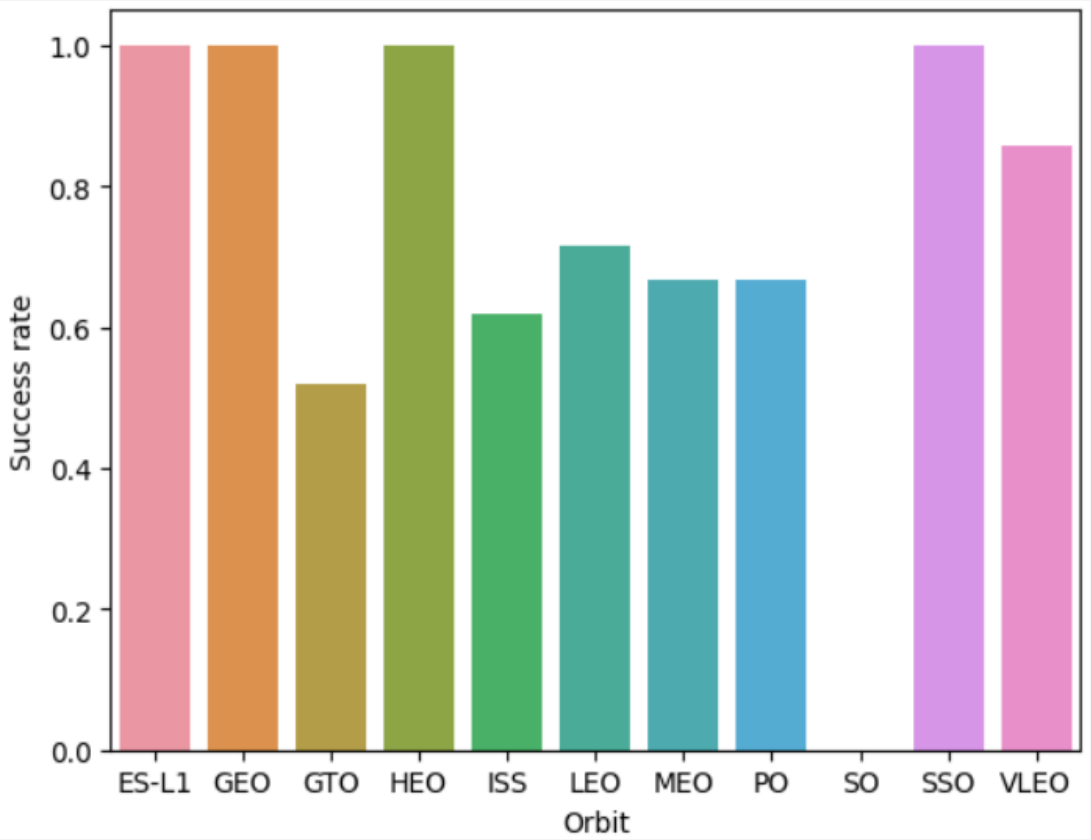
+) KSC LC 39 A: The successful landing is guaranteed when Payload is between 2000 and 5000, and between 9000 and near to 16000.



# Success Rate vs. Orbit Type

According to the chart, orbits have the most success rate are:

| Orbit | Class |
|---|---|
| ES-L1 | 1.000000 |
| GEO | 1.000000 |
| HEO | 1.000000 |
| SSO | 1.000000 |
| VLEO | 0.857143 |
| LEO | 0.714286 |
| MEO | 0.666667 |
| PO | 0.666667 |
| ISS | 0.619048 |
| GTO | 0.518519 |
| SO | 0.000000 |



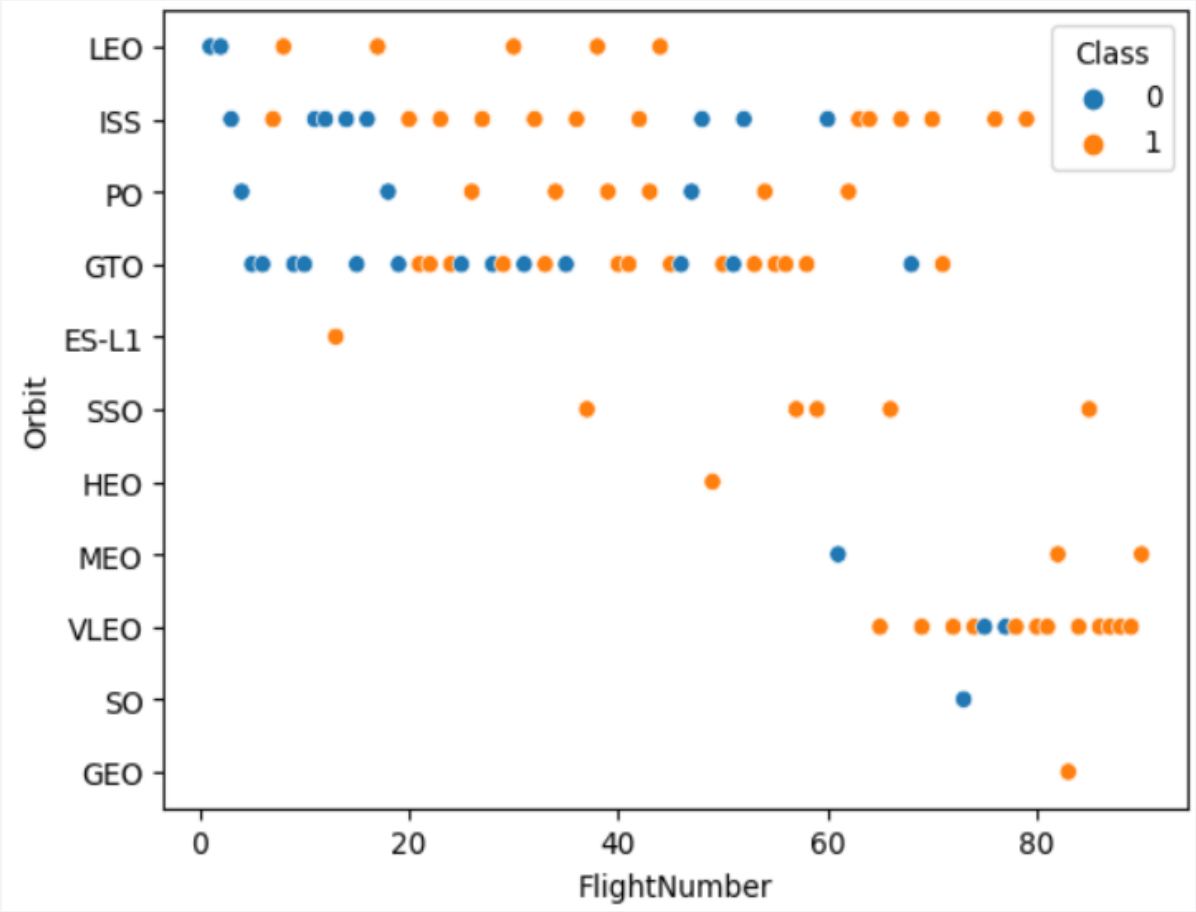
# Flight Number vs. Orbit Type

According to the chart:

+) The LEO orbit: The higher the number of flight, the more successful the landing.

+) The SSO orbit, HEO orbit, ES-L1 orbit and GEO orbit: Despite they have the least flights, they have successful landing without any fails.

+) The rest orbits: There are no correlation between Flight Number and Orbit



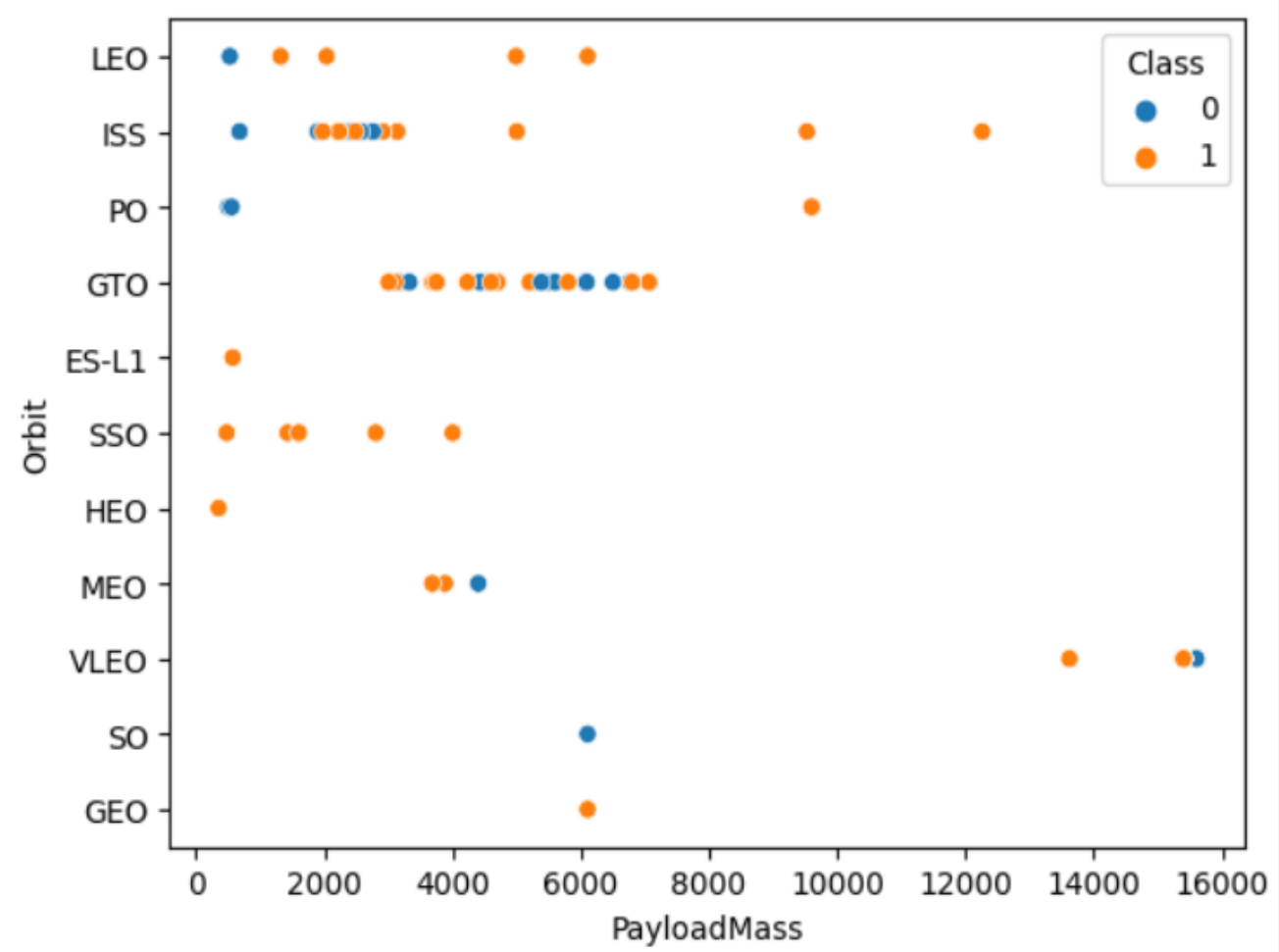
# Payload vs. Orbit Type

According to the chart:

+) The LEO orbit and PO orbit: The higher the pay load mass, the more successful the landing.

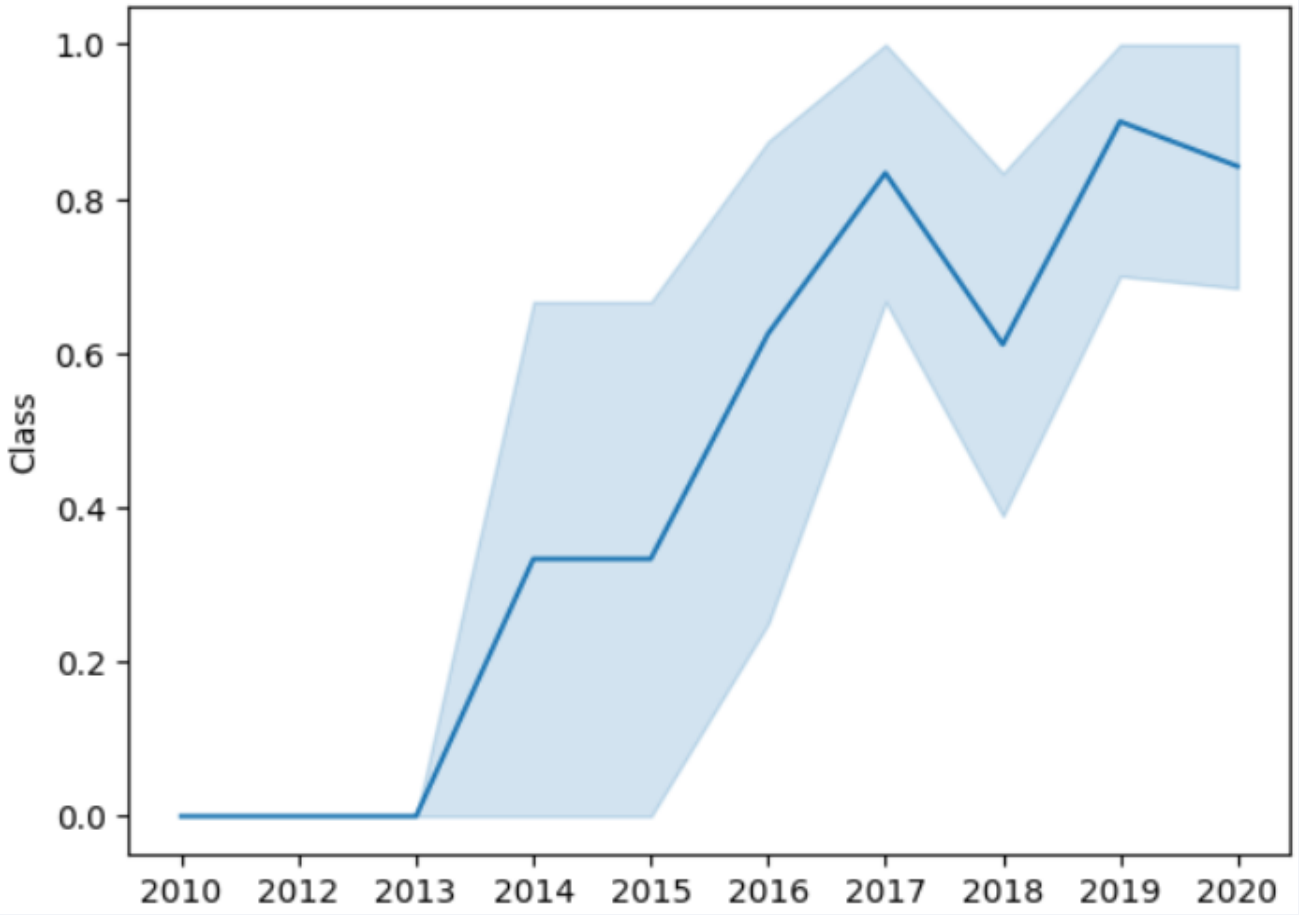
+) The SSO orbit, HEO orbit, ES-L1 orbit and GEO orbit: They have successful landing without any fails.

+) The rest orbits: There are no correlation between Pay load mass and Orbit



# Launch Success Yearly Trend

There is an increment between successful landing and years, in spite of having decrement minorly.



# All Launch Site Names

## Task 1

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

In [ ]:
```
%%sql
select distinct(launch_site)
from SPACEX
where launch_site not like 'None'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

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

# Launch Site Names Begin with 'CCA'

## Task 2

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

In [ ]:

```
%%sql
select *
from SPACEX
where launch_site like 'CCA%'
limit 5
```

```
* ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

| 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 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-08-10 | 0:35:00 | 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 |

# Total Payload Mass

## Task 3

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

In [ ]:
```
%%sql
select sum(payload_mass__kg_)
from SPACEX
where customer like 'NASA (CRS)'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

| 1 |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

## Task 4

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

In [ ]:
```
%%sql
select avg(payload_mass__kg_)
from SPACEX
where booster_version like 'F9 v1.1'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

| 1 |
|---|
| 2928 |

# First Successful Ground Landing Date

## Task 5

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

*Hint:Use min function*

In [ ]:

```
%%sql
select min(DATE)
from SPACEX
where landing_outcome like 'Success (ground pad)'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

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

# Successful Drone Ship Landing with Payload between 4000 and 6000

## Task 6

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

In [ ]:
```
%%sql
select booster_version
from SPACEX
where 4000 < payload_mass__kg_ and payload_mass__kg_ < 6000 and landing_outcome like 'Success (drone ship)'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

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

# Total Number of Successful and Failure Mission Outcomes

## Task 7

List the total number of successful and failure mission outcomes

In [ ]:
```
%%sql
select count(mission_outcome)
from SPACEX
where mission_outcome like 'Failure%' or mission_outcome like 'Success%'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

| 1 |
|---|
| 101 |

# Boosters Carried Maximum Payload

### Task 8

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

In [ ]:
```
%%sql
select booster_version
from SPACEX
where payload_mass__kg_ = (select max(payload_mass__kg_)
                           from SPACEX)
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

| 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

## 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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.**

In [ ]:
```
%%sql
select substr(Date, 6, 2) as MONTH, landing_outcome, booster_version, launch_site
from SPACEX
where landing_outcome like 'Failure (drone ship)' and DATE like '2015%'
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[ ]:

| MONTH | landing_outcome | booster_version | launch_site |
|---|---|---|---|
| 10 | 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

## 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 [ ]:

```
%%sql
select landing_outcome , count(*) as count
from SPACEX
where (DATE between '2010-06-04' and '2017-03-20')
group by landing_outcome
order by count desc
```

```
 * ibm_db_sa://vsx13718:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

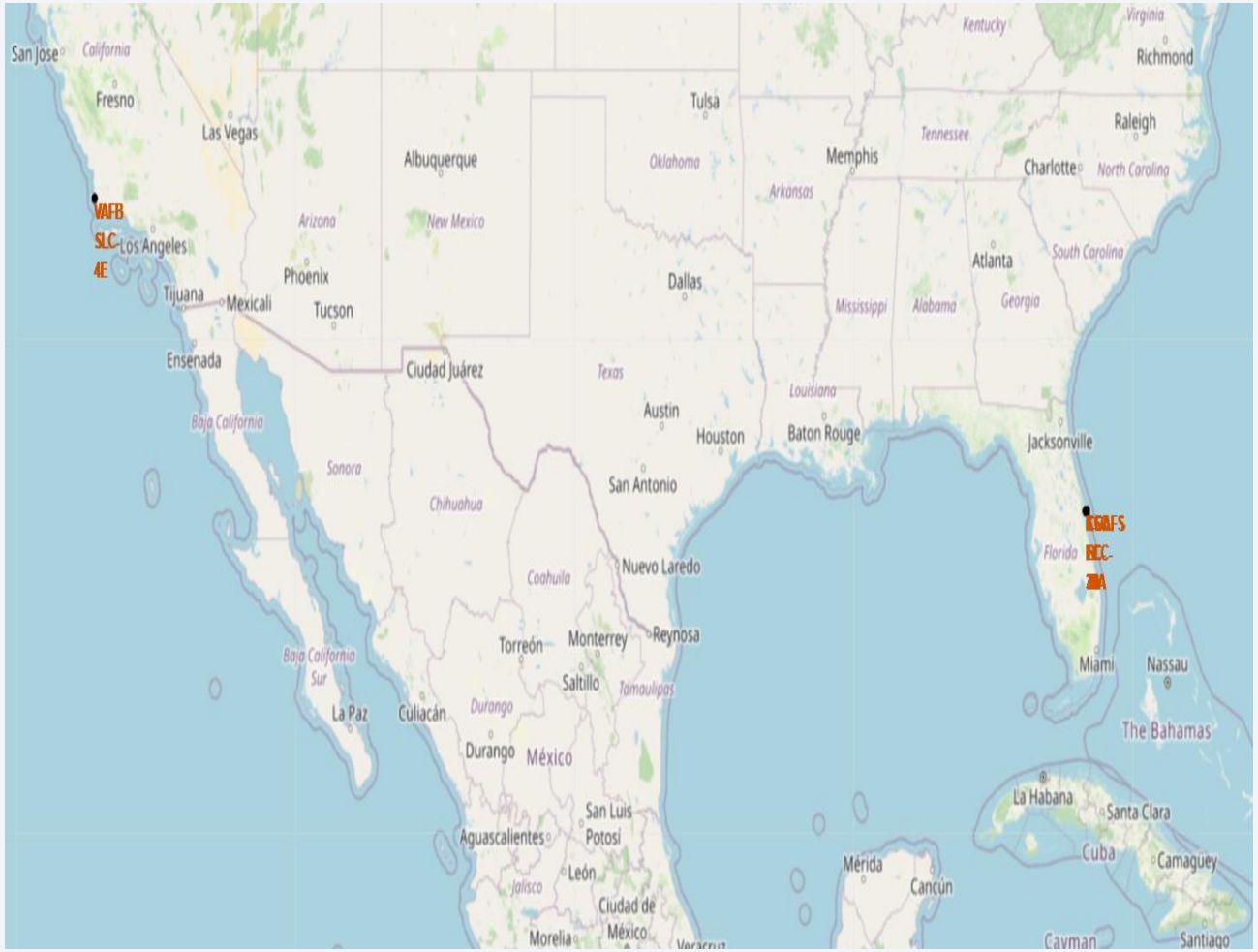
Out[ ]:

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

Section 3

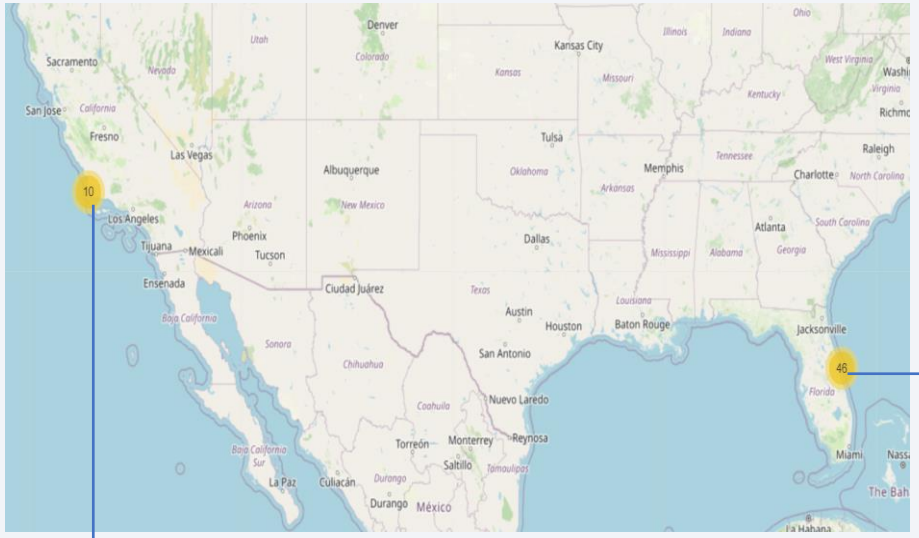
# Launch Sites Proximities Analysis

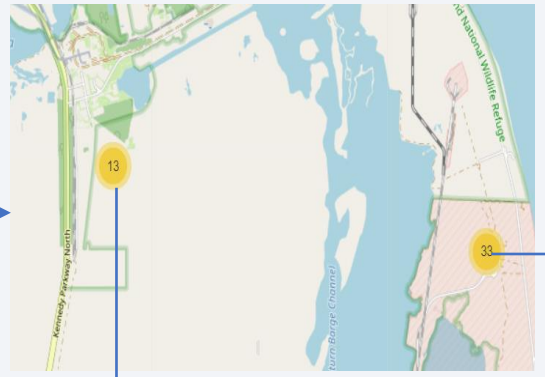
# Mark all launch sites on a map

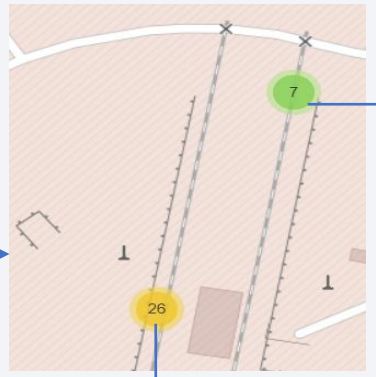


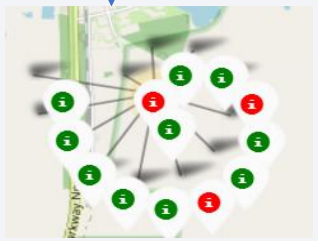

All launch sites are likely to be occurred at the near of the sea.

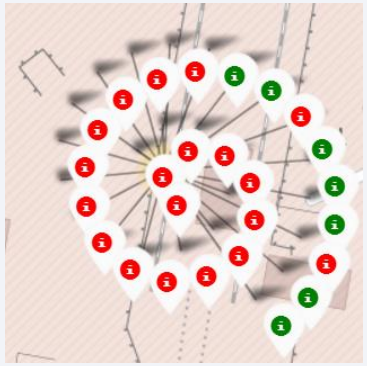
# Mark the success/failed launches for each site on the map



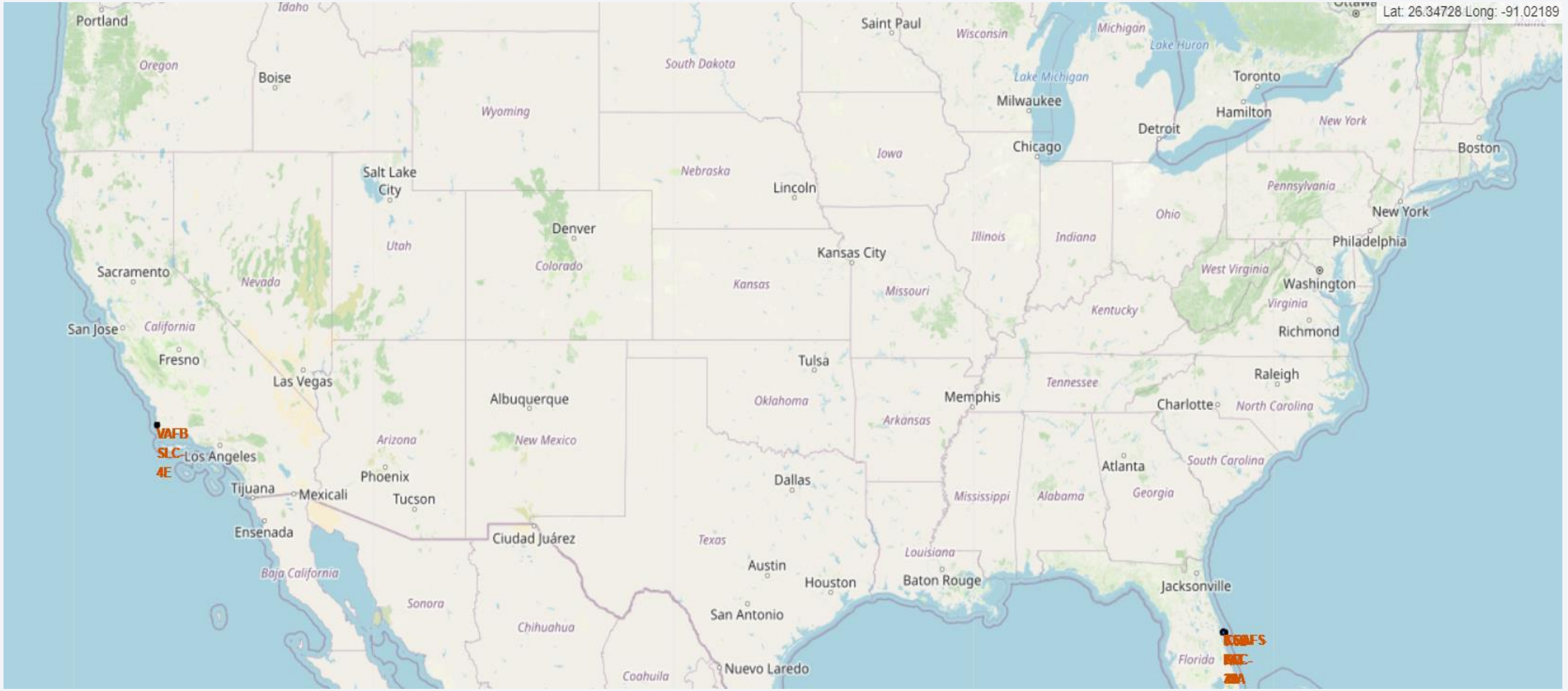



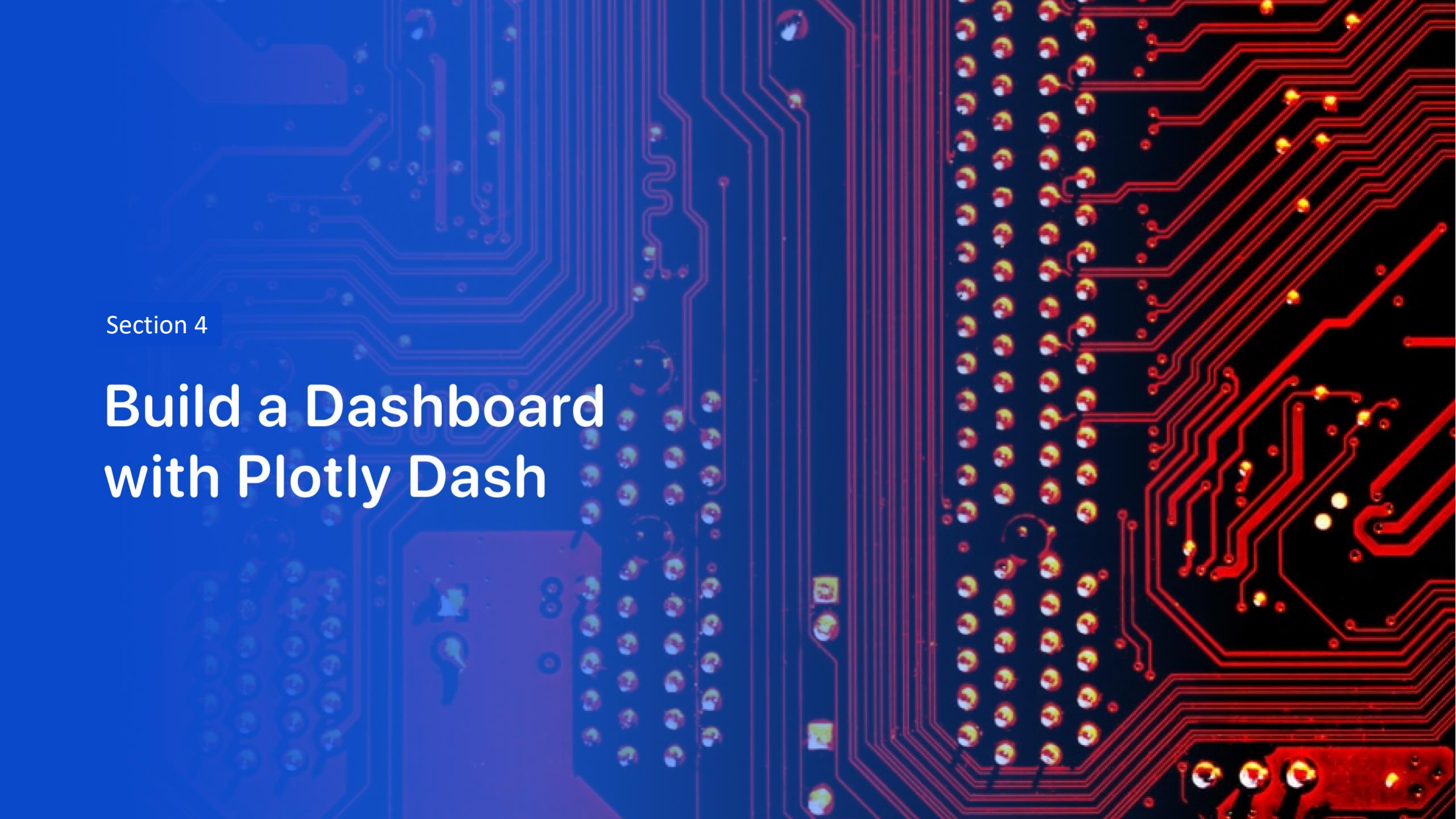





The green mark stands for successful landing and the red mark stands for failure landing.

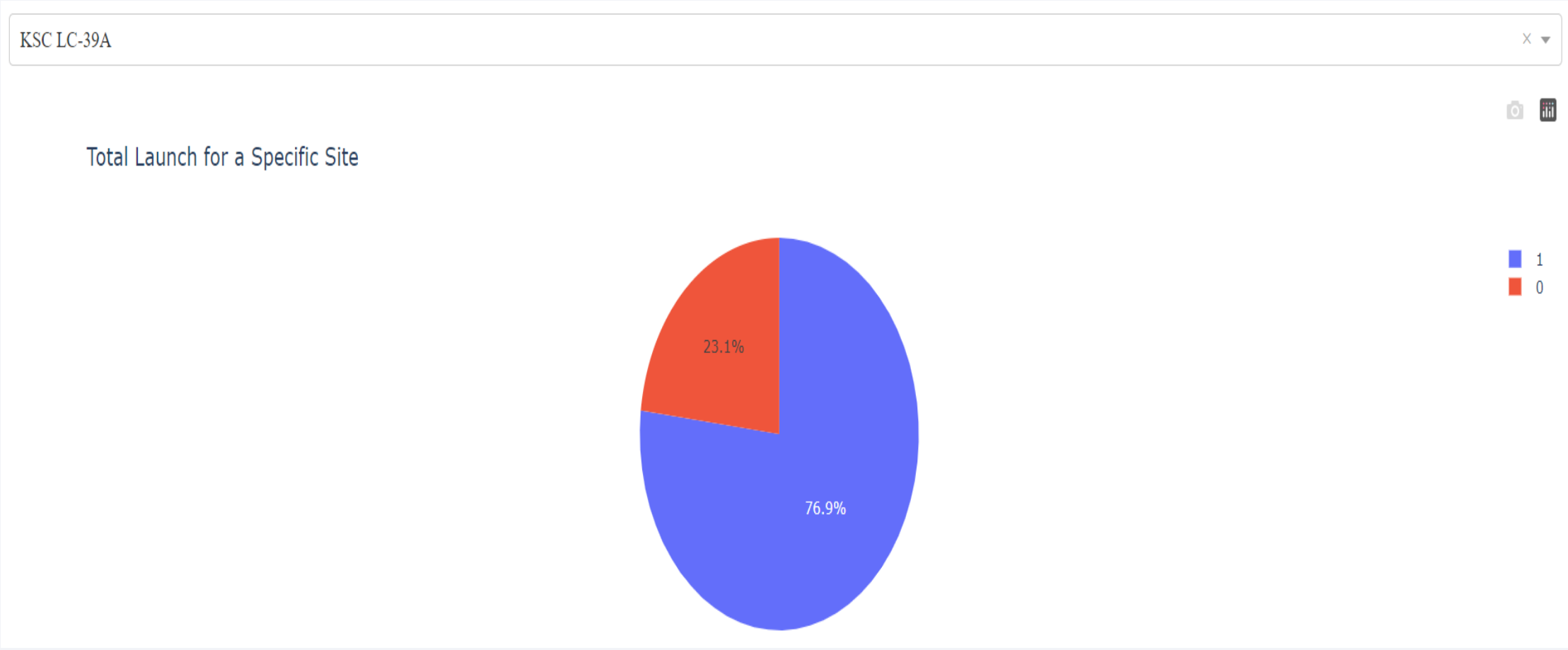
# Calculate the distances between a launch site to its proxities



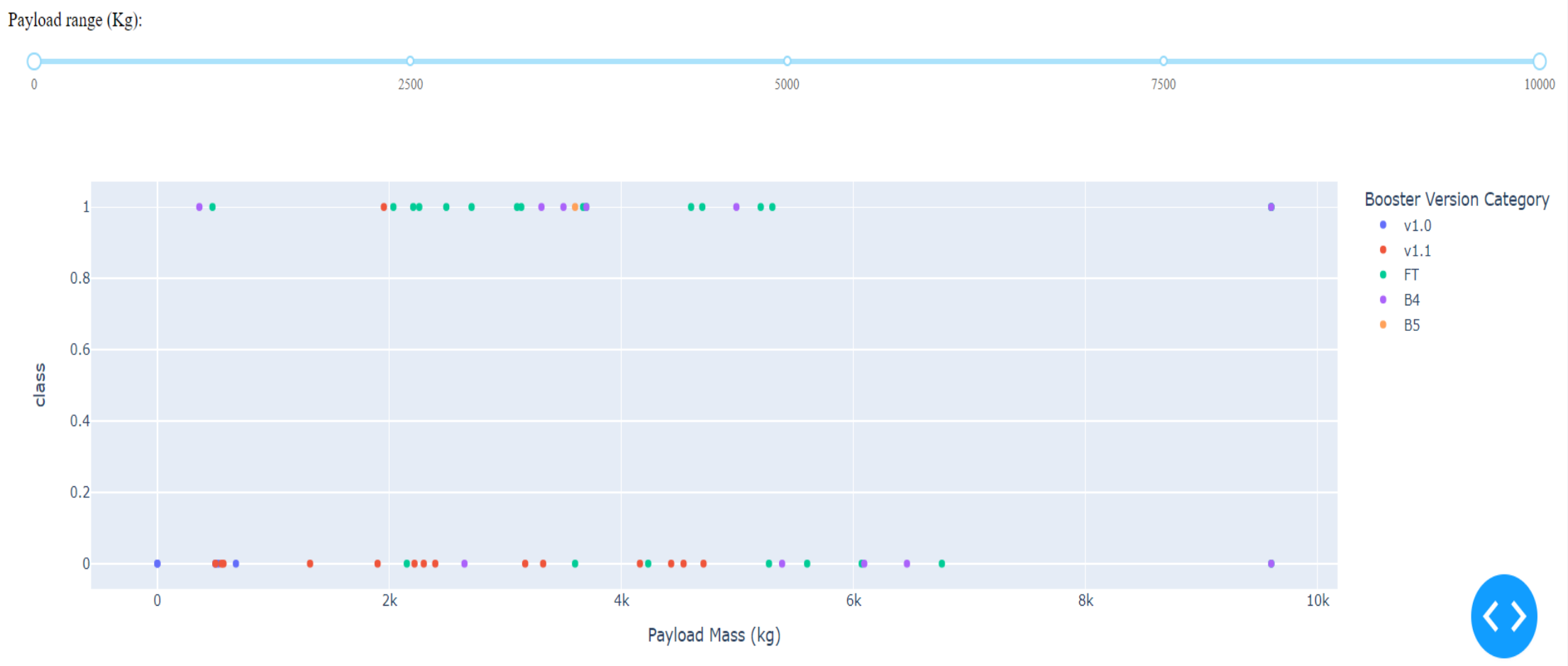

Section 4
Build a Dashboard with Plotly Dash

# A pie chart visualize the success count for all sites

# A pie chart for the launch site with highest launch success ratio



# A scatter plot for Payload vs Launch Outcome for all sites

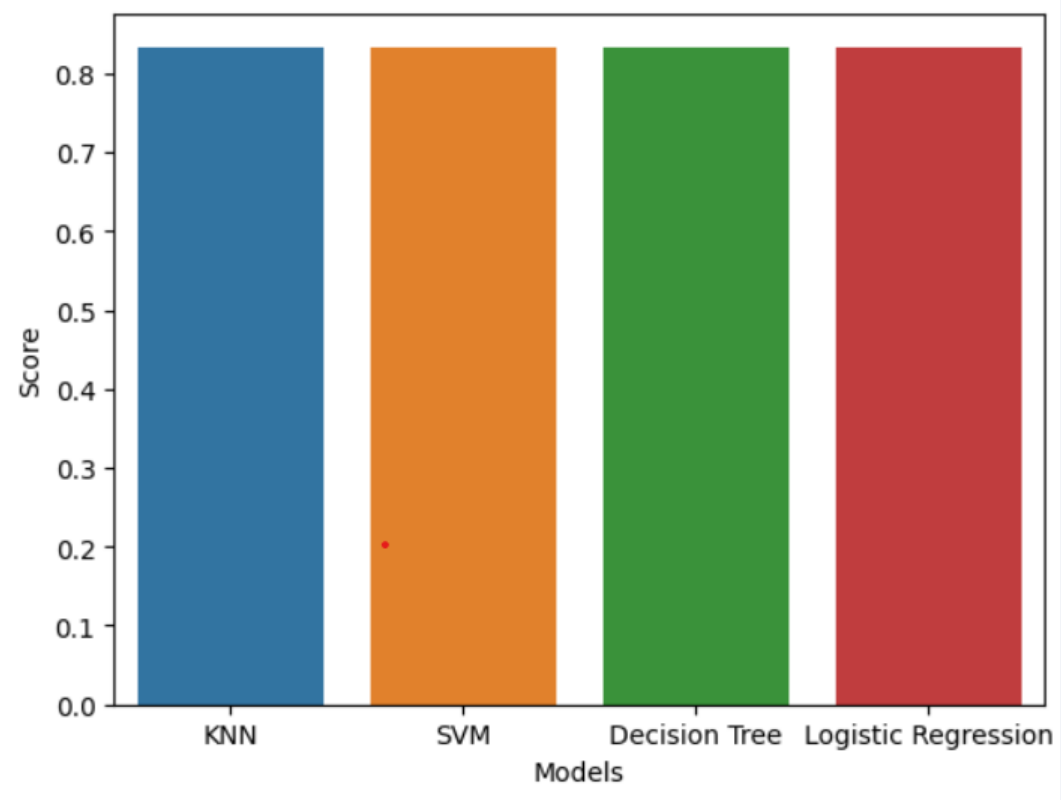


Section 5

# Predictive Analysis (Classification)

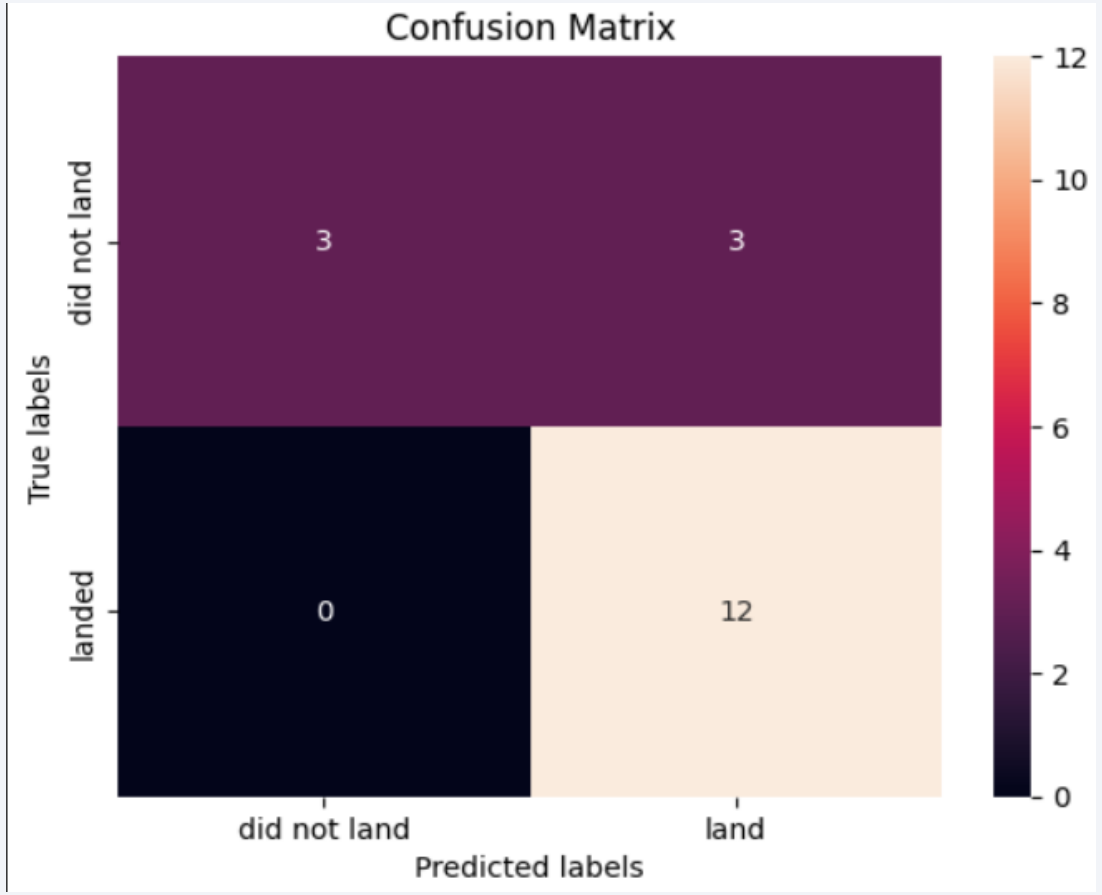
# Classification Accuracy

Accuracy for each selected models are equal to each other.



# Confusion Matrix

Every selected models shares the same result of confusion matrix.



# Conclusions

- Every models works the same, as they share the same accuracy score (83.33%)

# Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

# Thank you!