IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

<Name>
<Date>

# Outline

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

# Executive Summary

- Data collected from SpaceX API and web scraping from wikipedia
- Perform data wrangling
- One hot encoder applied on categorical features
- Perform EDA and Interactive dashboards
- Perform predictive analysis using classification models
- All classification models produces same result

# Introduction

Project Background and Context

- Space X announces that Falcon 9 launches with a cost of $62M other providers cost more than $165M
- If we can predict whether Falcon 9 launches on its first stage with the information given data

Problems you want to find answers

- What are the factors that causes the rocket to land successfully?
- What are the relationship between the variables that affects the outcome?

Section 1

# Methodology

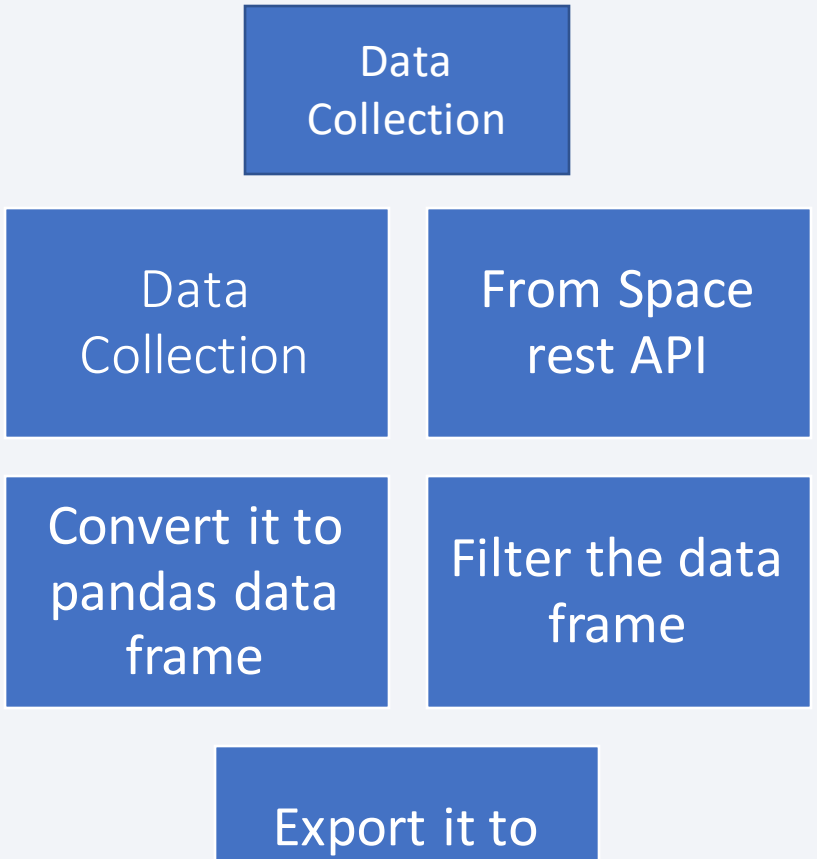
# Methodology

## Executive Summary

- Data collection methodology:
  - Data was downloaded from Space X API and web scraping from wikipedia
- Perform data wrangling
  - Cleaned the messy data and organized in structured format for further analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build the model and fit it.
  - By using their parameters and find the best parameter

# Data Collection

- Data can be downloaded from the space rest API and web scraping from wikipedia
- You need to present your data collection process use key phrases and flowcharts



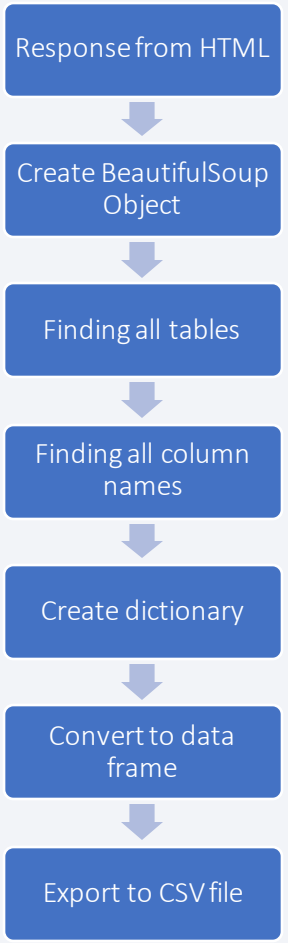
# Data Collection – SpaceX API

- Getting response from rest API
- Converting response into Json file
- Apply functions for cleaning the data
- Align list into dictionary
- Convert dictionary into Data frame
- Filter data frame
- Export it to CSV file)

- testrepo/jupyter-labs-spacex-data-collection-api (1).ipynb at master · Aravinth-Megnath/testrepo (github.com)

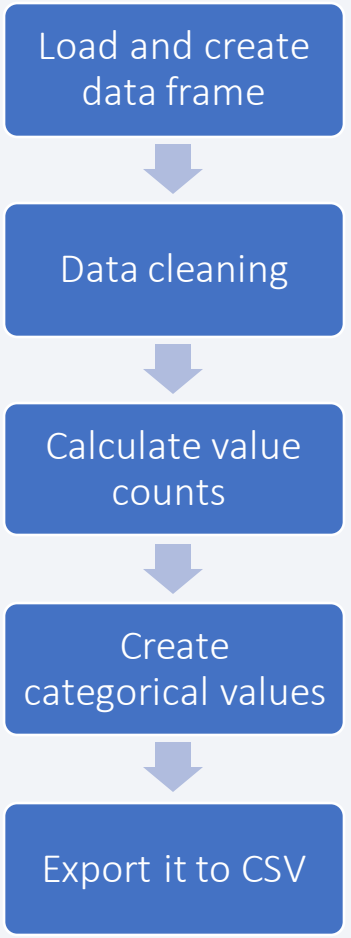
# Data Collection - Scraping

- Getting response from HTML
- Creating BeautifulSoup object
- Finding all the tables
- Getting the column names
- Create dictionary and add the values to the keys
- Convert dictionary to data frame
- Convert it to CSV file
- [testrepo/jupyter-labs-webscraping (3).ipynb at master · Aravinth-Megnath/testrepo (github.com)]



# Data Wrangling

- Cleaning the data which is unwanted and also into the right format
- Creating categorical Class variable , 1 for successful launching and 0 for launch failure
- Steps
  - Load and create data frame
  - Clean the data
  - Calculate number of launches at each launch sites
  - Calculate number of Orbits
  - Calculate number of mission outcomes per orbit type
  - Create categorical Class variable using Boolean values from Outcome column
    - 1 for successful launch
    - 0 for failure launch




- [testrepo/spacex-data_wrangling_jupyterlite.jupyterlite.ipynb at master · Aravinth-Megnath/testrepo (github.com)]

# EDA with Data Visualization

- ✓ Catplot – Relationship between Payload mass and Number of flights
- ✓ Scatter point chart – Relationship between Launch site and No.of flights
- ✓ Scatter point chart -Relationship between Launch site and Pay load mass
- ✓ Bar chart – Relationship between success rate and orbit type
- ✓ Scatter point chart – Relationship between Number of flights and Orbit type
- ✓ Scatter point chart – Relationship between Number of flights and Orbit type
- ✓ Scatter point chart – Relationship between payload and orbit type
- ✓ Line Chart – Relationship between year and success rate

testrepo/eda-dataviz.ipynb.jupyterlite (2).ipynb at master · Aravinth-Megnath/testrepo (github.com)

# EDA with SQL

- Displayed names of unique launched site
- Displayed 5 records where launch sites with a string 'CCA'
- Displayed the total payload mass carried by boosters launched by NASA(CRS)
- Average Payload mass carried by booster version F9 v1.1
- On when the first successful landing outcome in ground pad was achieved
- List of names of successful boosters in ground pad and payload mass between 4000 kg and 6000 kg
- Total number of successful and failure mission outcomes
- Names of booster versions which has maximum payload mass
- Failure landing outcomes in drone ship in the year 2015 particularly in which month
- Ranking of landing outcomes between Jun 2010 and Mar 2010 in descending order.

testrepo/eda-sql-coursera_sqllite.ipynb at master · Aravinth-Megnath/testrepo (github.com)

# Build an Interactive Map with Folium

- ❖ Folium Circle – Added a circle in a folium map to highlight NASA Johnson's Space center coordinate
- ❖ Marker – Added a marker for each Launch site in the site map
- ❖ Marker Cluster – Created color labelled markers for easily identify which launch site has high success rates
- ❖ Mouse Position – Added a mouse position on the map to get coordinate for mouse when we hover over them
- ❖ Poly line – Drew a line from the launch site and the coastal area and also calculated the distance between them

- testrepo/folium-launch_site_location.jupyterlite.ipynb at master · Aravinth-Megnath/testrepo (github.com)

# Build a Dashboard with Plotly Dash

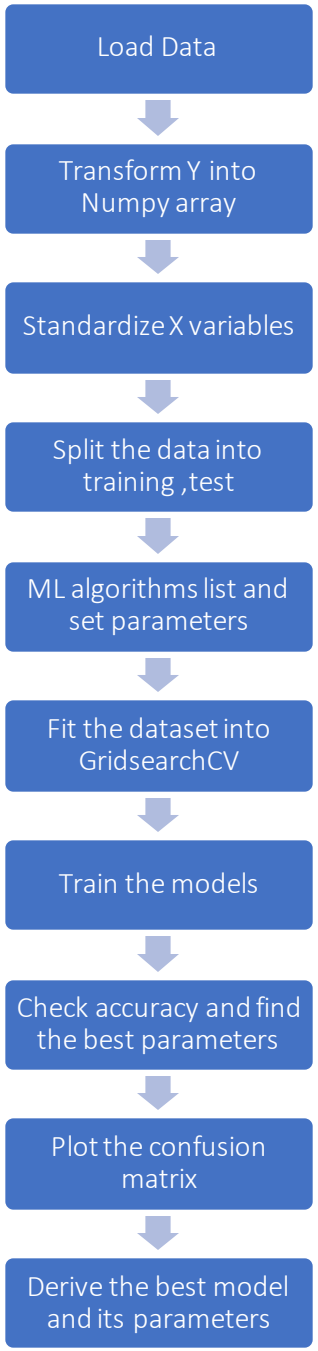
- ❑ Pie Charts – It shows the total success for all sites or by certain sites
- ❑ Scatter chart – It shows the correlation between the payload and success rate for all sites or by certain sites.

- testrepo/Dash Final Assignment.ipynb at master · Aravinth-Megnath/testrepo (github.com)

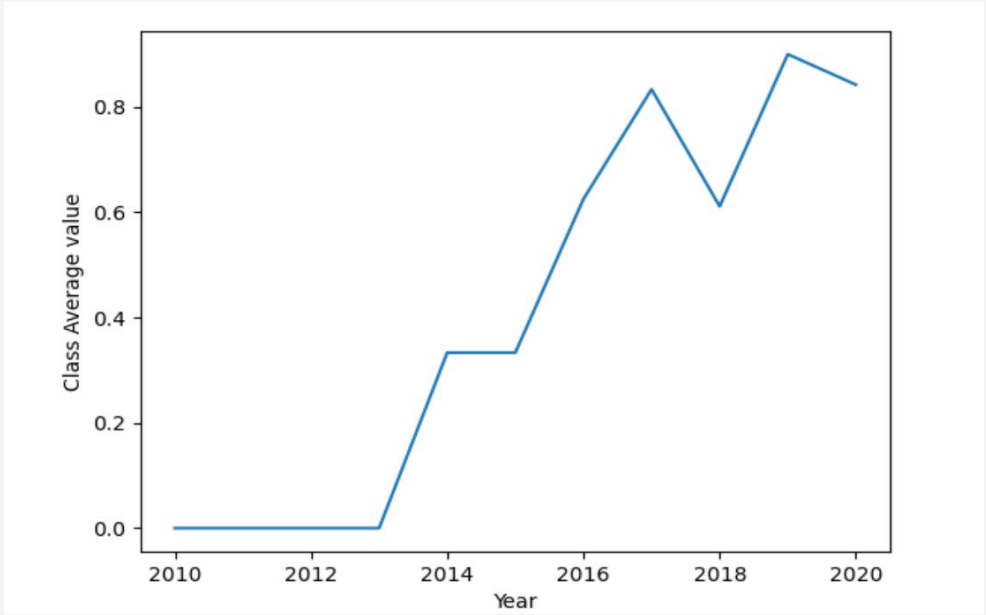
# Predictive Analysis (Classification)

- Loaded the data into data frame
- Transform the target variable (Y) into numpy array
- Standardize the Input variables (X) by using StandardScaler
- Split the dataset which is having 20 % data for testing by train-test-split
- List the Classification ML algorithm and set the range of parameters
- Fit our datasets into GridSearchCv and train all our models
- Check accuracy and find the best parameters for each models
- Plot confusion matrix
- Derive the best model with the highest accuracy with best parameters

testrepo/module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at master · Aravinth-Megnath/testrepo (github.com)


Load Data
Transform Y into Numpy array
Standardize X variables
Split the data into training ,test
ML algorithms list and set parameters
Fit the dataset into GridsearchCV
Train the models
Check accuracy and find the best parameters
Plot the confusion matrix
Derive the best model and its parameters

# Results

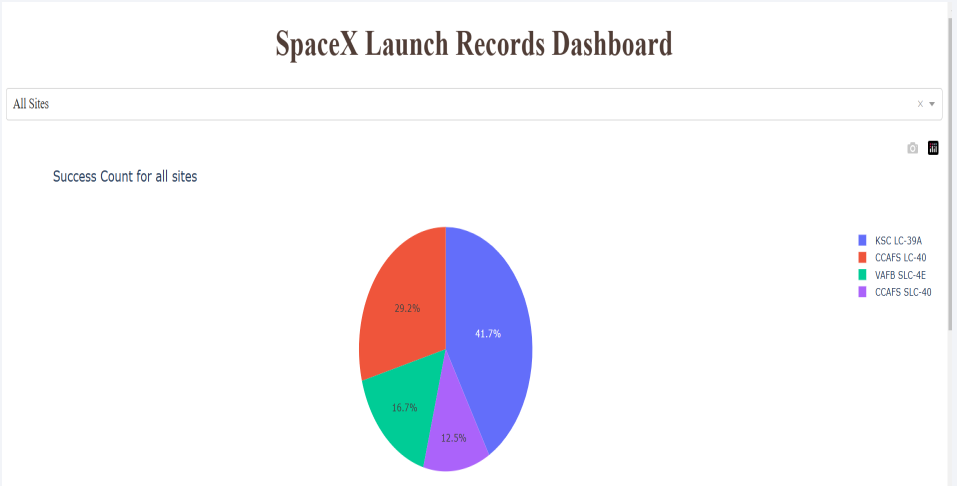



Accuracy score for Logistic Regression is 0.8333333333333334

Accuracy score for Support Vector Machine is 0.8333333333333334

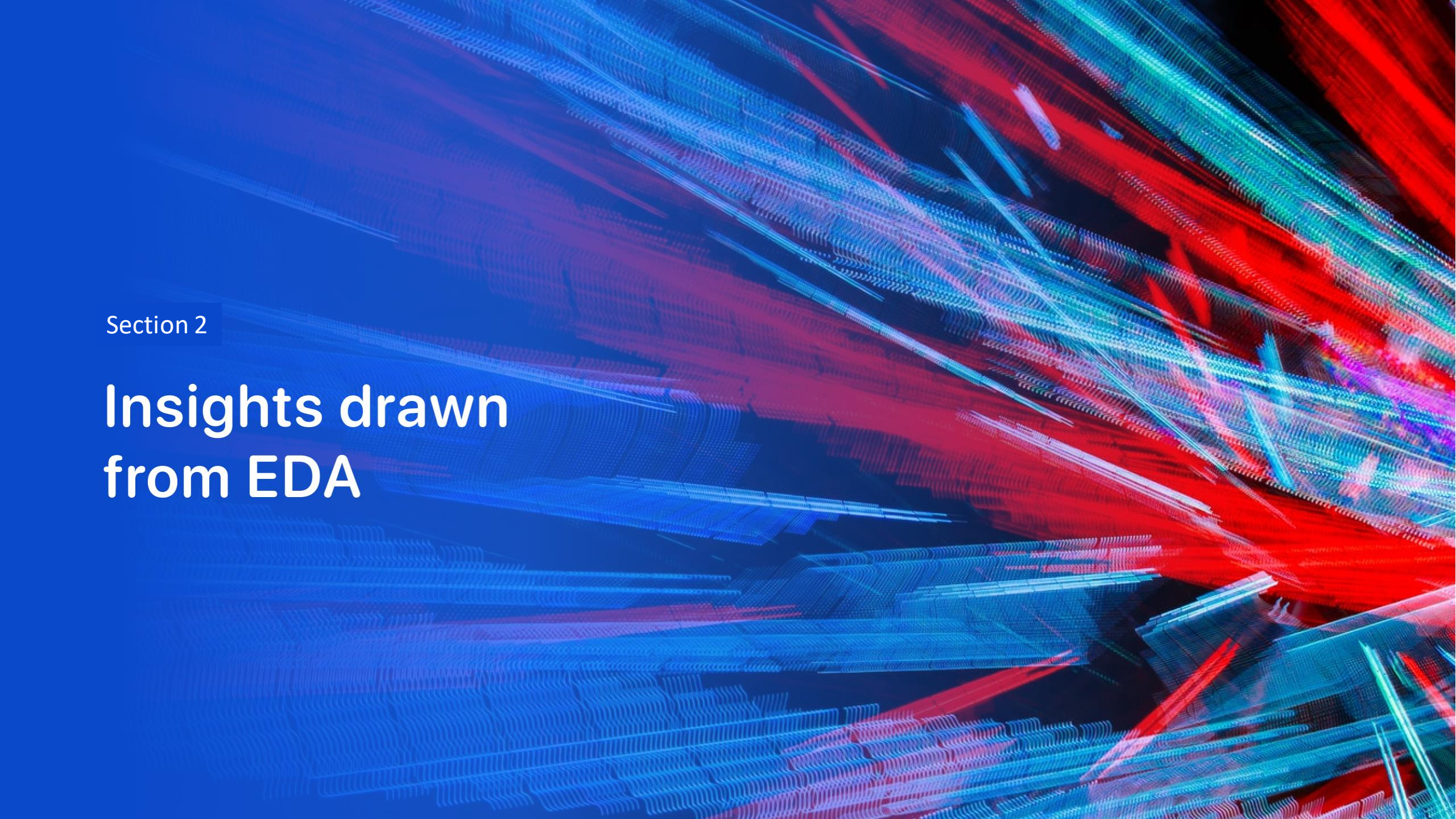
Accuracy score for decision tree classifier is 0.8333333333333334

Accuracy score for KNN is 0.8333333333333334



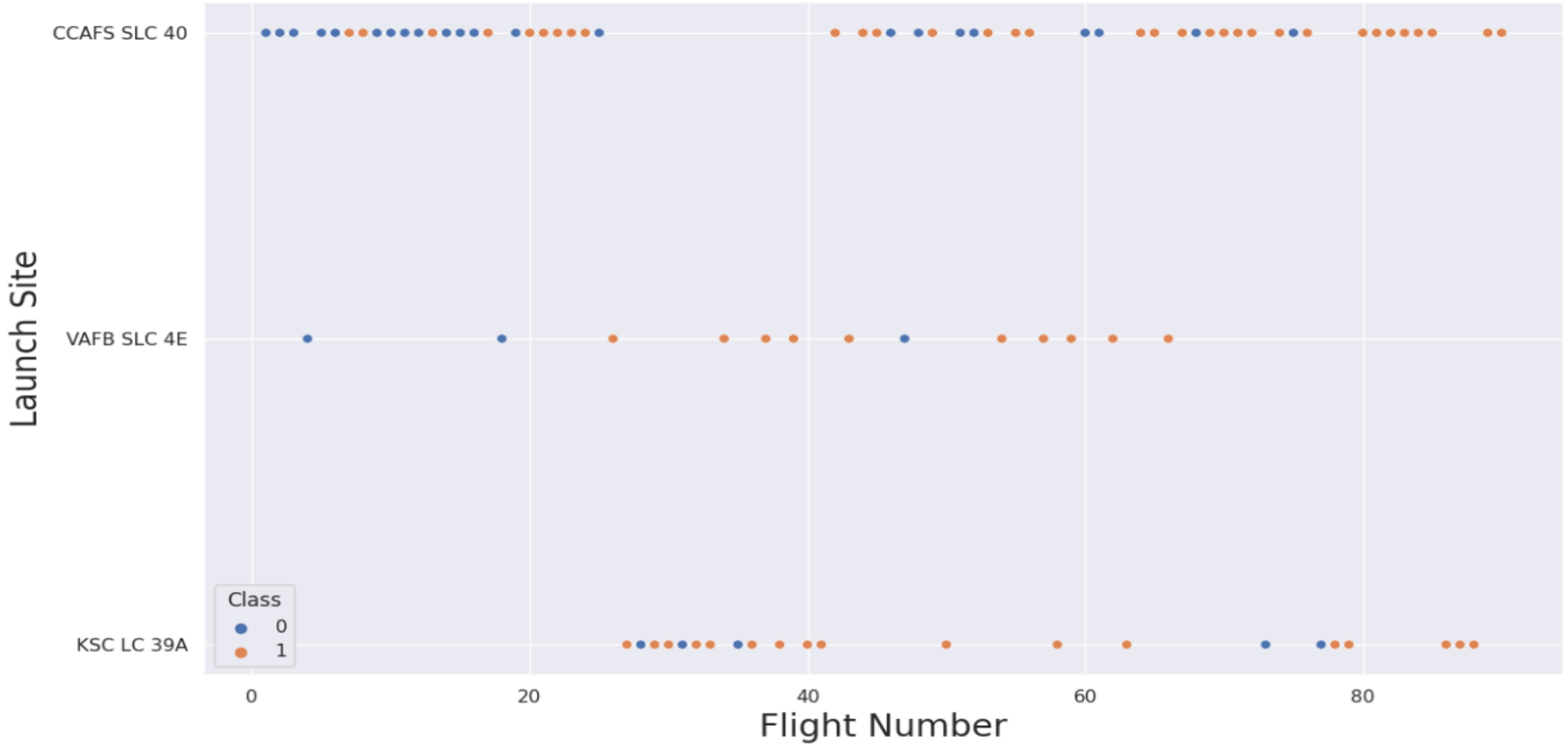

Section 2

# Insights drawn from EDA



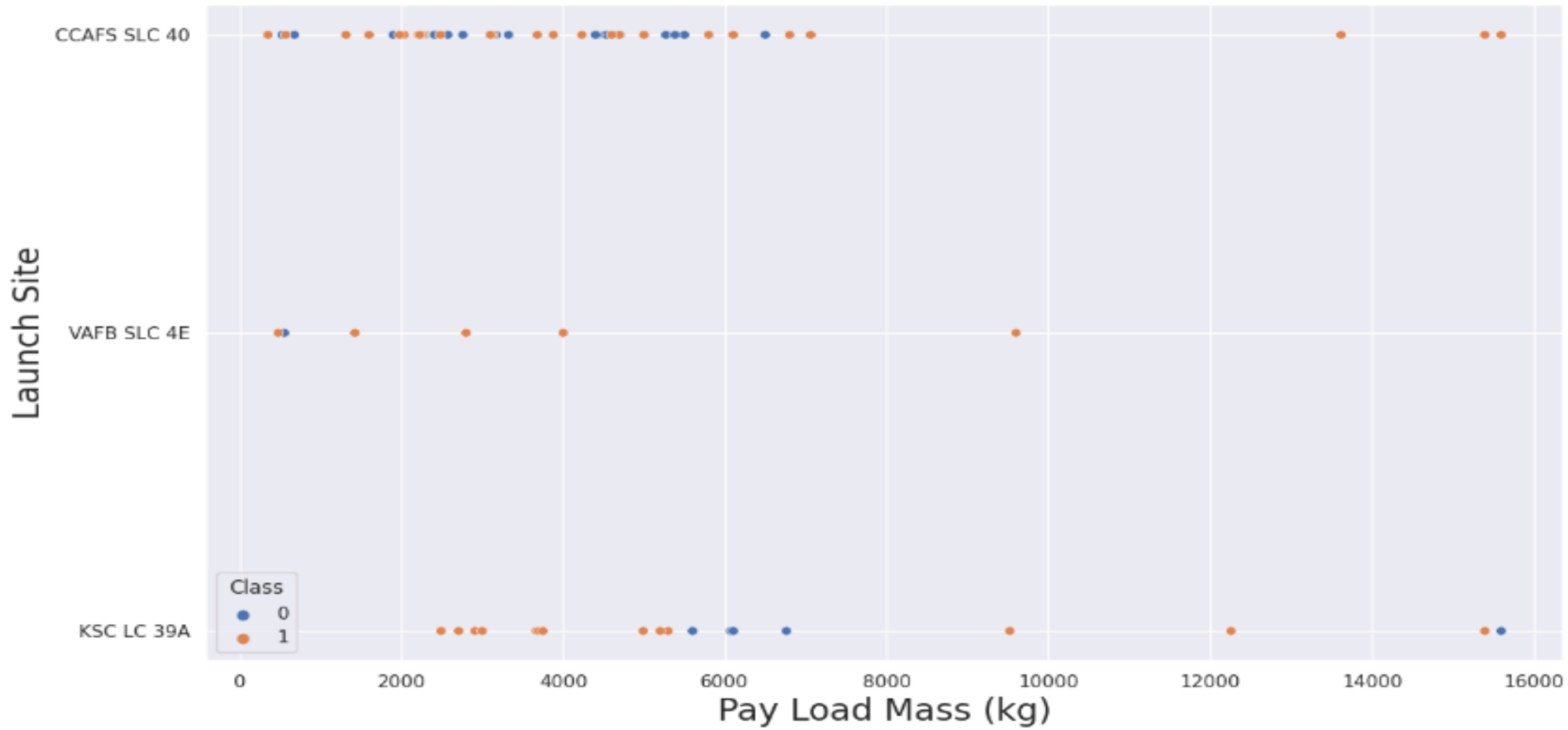
- From the graph we can see that the success rate increases with the number of flights

# Flight Number vs. Launch Site



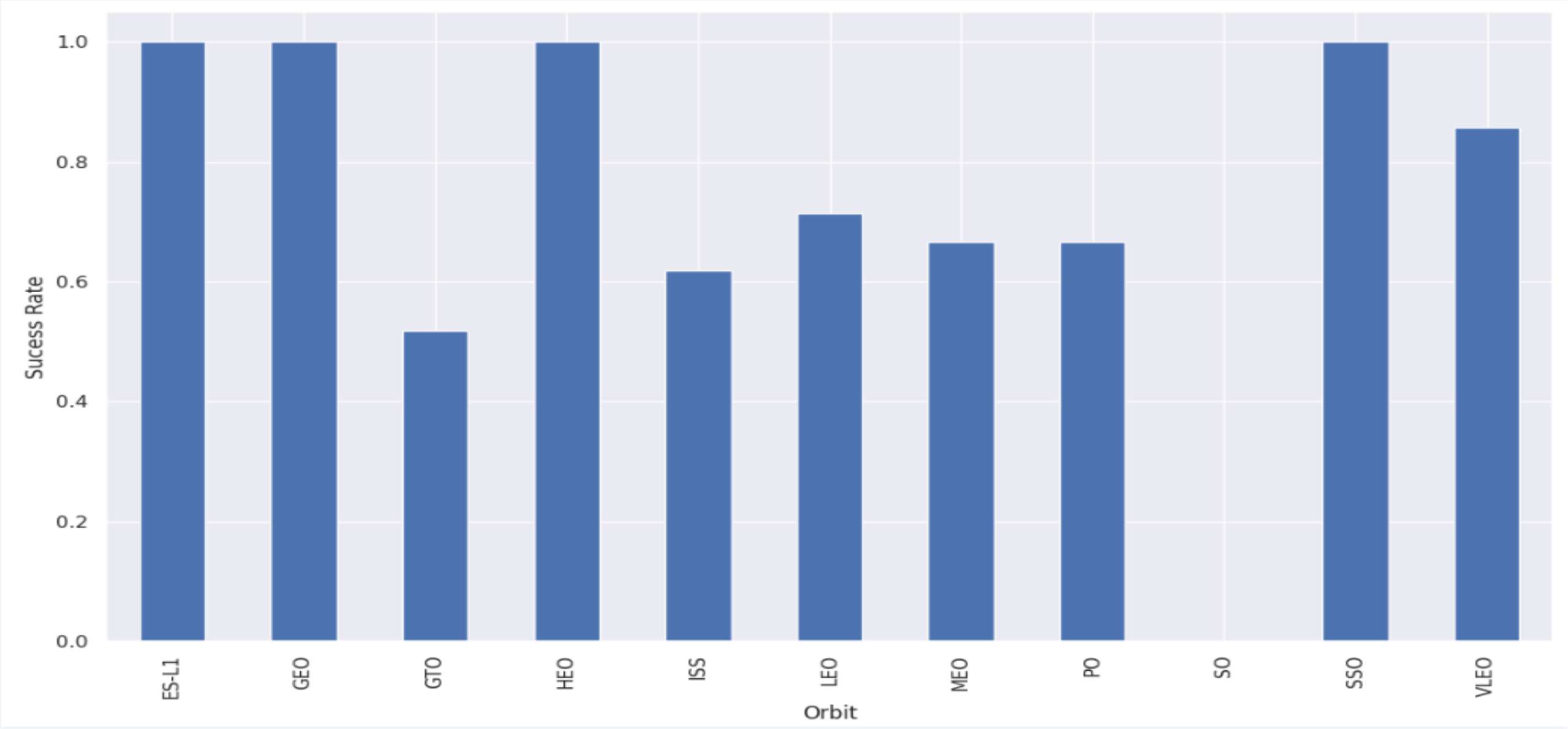
# Payload vs. Launch Site

- VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000)
- For CCAFS SLC 40 success rate is good with the increase of pay load mass



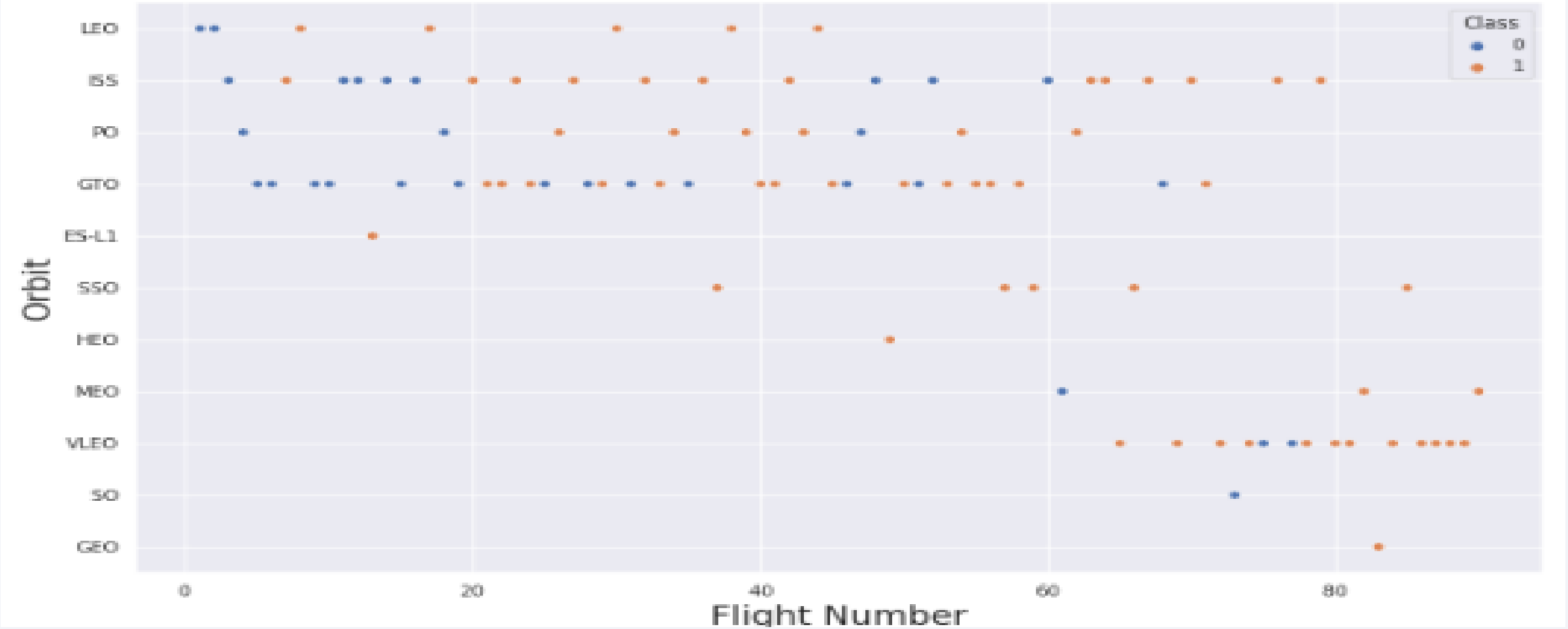
# Success Rate vs. Orbit Type

- ES – L1,GEO,HEO,SSO are the orbits which have good success rate



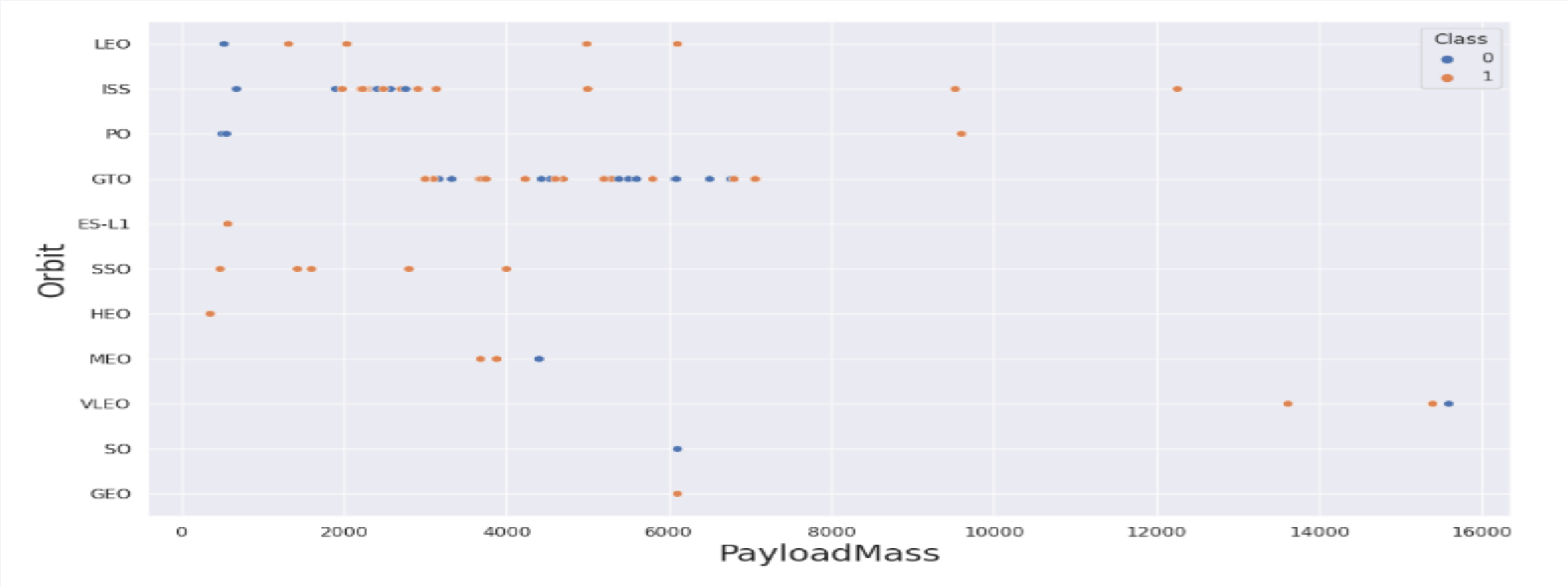
# Flight Number vs. Orbit Type

- Only LEO orbit has the increase in success rate with respect to increase in number of flights, while others doesn't have any relations



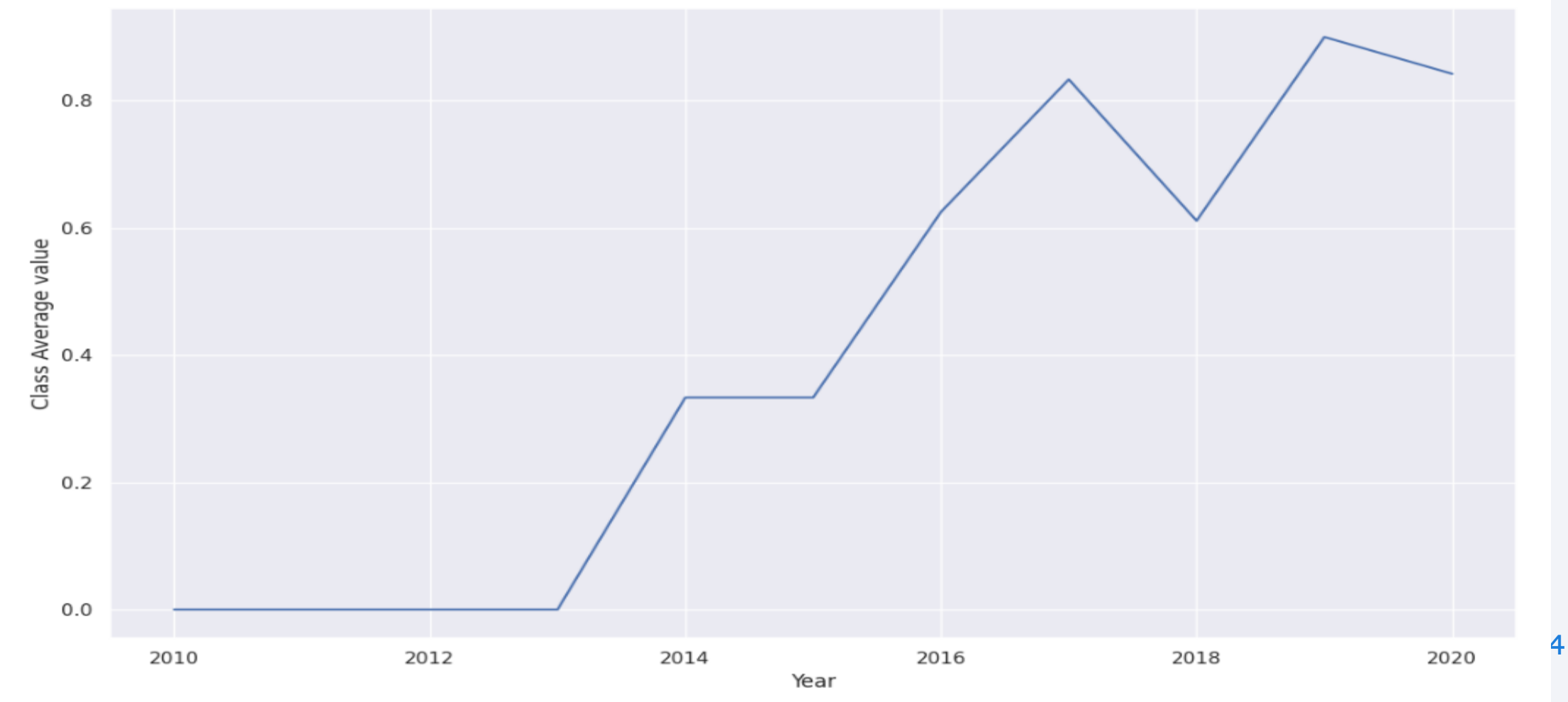
# Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



# Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2020



# All Launch Site Names

- We can obtain this by using Distinct function in our query

```
%sql select  distinct Launch_site 'Launch_Site' from SPACEXTBL ;
```

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

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

# Launch Site Names Begin with 'CCA'

- We should use wildcards where and like to get this result

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

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

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload |
|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit |
| 08-12-2010 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 |

# Total Payload Mass

- The total payload carried by boosters from NASA
- We should use sum function in pay load mass where customer like NASA

```
%sql select sum(PAYLOAD_MASS__KG_) as 'Total Pay Load Mass in Kg',\
Customer from SPACEXTBL where Customer like '%NASA (CRS)%'
```

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

| Total Pay Load Mass in Kg | Customer |
|---|---|
| 48213 | NASA (CRS) |

# Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1
- We should average function in Payload Mass where booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as 'Average pay load mass in Kg',\
Booster_Version from SPACEXTBL where Booster_Version = 'F9 v1.1'
```

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

| Average pay load mass in Kg | Booster_Version |
|---|---|
| 2928.4 | F9 v1.1 |

# First Successful Ground Landing Date

- Date when is the first successful landing outcome on ground pad
- We should use min function on Date where landing outcome is ground pad

```
%sql select min(Date) as 'First successful landing', "LANDING _OUTCOME" from SPACEXTBL \
WHERE  "LANDING _OUTCOME" like '%Success (ground pad)%'
```

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

| First successful landing | Landing _Outcome |
|---|---|
| 01-05-2017 | Success (ground pad) |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- We should select Booster version , landing outcome, payload mass where landing outcome is successful in drone ship and payload mass between 4000 and 6000

```
%sql select Booster_Version ,"LANDING _OUTCOME",PAYLOAD_MASS__KG_ \
from SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 and "LANDING _OUTCOME" = 'Success (drone ship)'
```

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

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

# Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes
- We should use count function on Mission outcome and group by mission outcome

```
%sql select count(Mission_Outcome), Mission_Outcome from   SPACEXTBL GROUP BY Mission_Outcome
```

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

| count(Mission_Outcome) | Mission_Outcome |
|---|---|
| 1 | Failure (in flight) |
| 98 | Success |
| 1 | Success |
| 1 | Success (payload status unclear) |

# Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass
- We should use max function on payload mass in the sub query

```
%sql select Booster_Version, PAYLOAD_MASS__KG_ as 'Maximum Payload Mass in Kg'\
from SPACEXTBL where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

| Booster_Version | Maximum Payload Mass in Kg |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- We should use substr for date and two where conditions using 'and'

```
%sql select Date ,substr(Date, 4, 2) as Month , "LANDING _OUTCOME",Booster_Version,Launch_Site \
from SPACEXTBL where  substr(Date,7,4)='2015' and "LANDING _OUTCOME" like '%failure (drone ship)%'
```

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

| Date | Month | Landing _Outcome | Booster_Version | Launch_Site |
|---|---|---|---|---|
| 10-01-2015 | 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 14-04-2015 | 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

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

- 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.
- We should use count function on landing outcome where the date between the given date and group by landing outcome and also order by landing outcome in a descending order.

```
%sql select Date, "LANDING _OUTCOME" ,\
count("LANDING _OUTCOME") as 'Successful landing outcomes  between the date 04-06-2010 and 20-03-2017' \
from SPACEXTBL \
where "LANDING _OUTCOME" like '%Suc%' and DATE between '04-06-2010' and '20-03-2017'\
group by "LANDING _OUTCOME" \
order by "LANDING _OUTCOME"
```

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

| Date | Landing _Outcome | Successful landing outcomes between the date 04-06-2010 and 20-03-2017 |
|---|---|---|
| 07-08-2018 | Success | 20 |
| 08-04-2016 | Success (drone ship) | 8 |
| 18-07-2016 | Success (ground pad) | 6 |

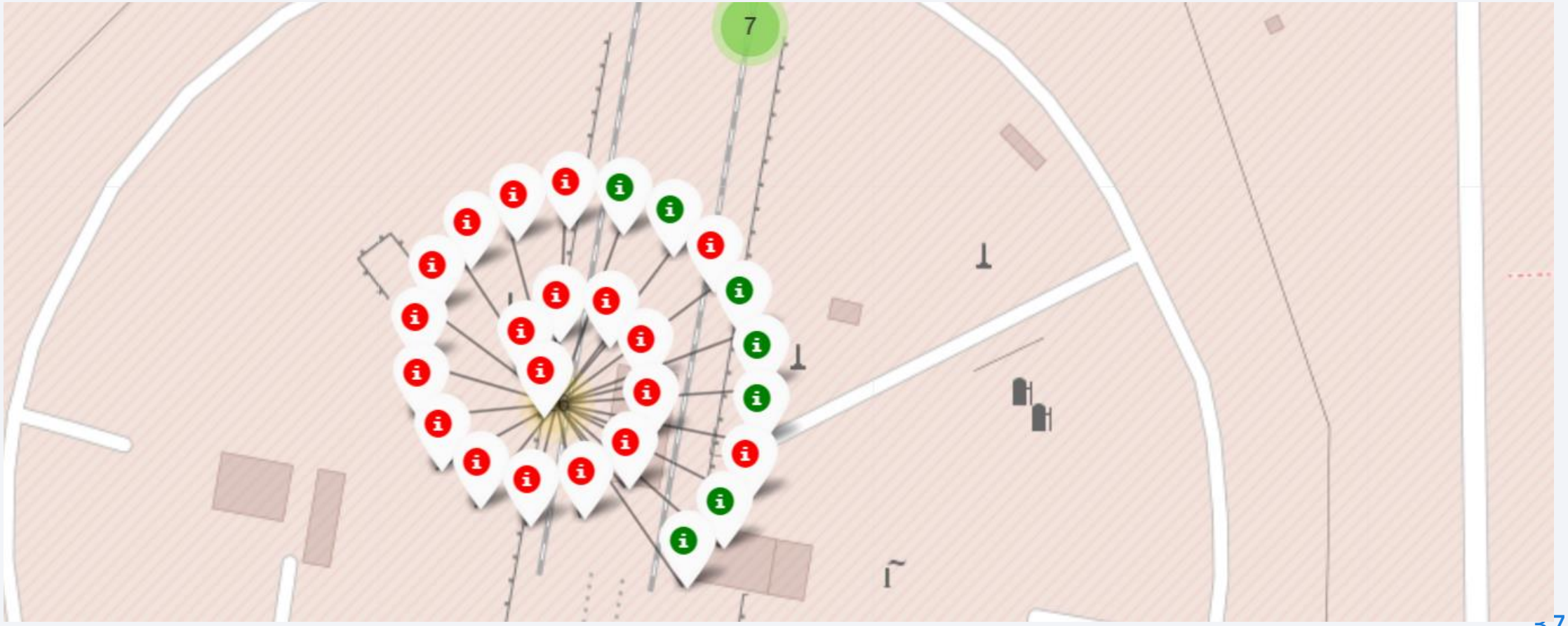

Section 3
Launch Sites
Proximities Analysis

# All Launch sites location markers

- All the Launch sites are located just near the coastal area

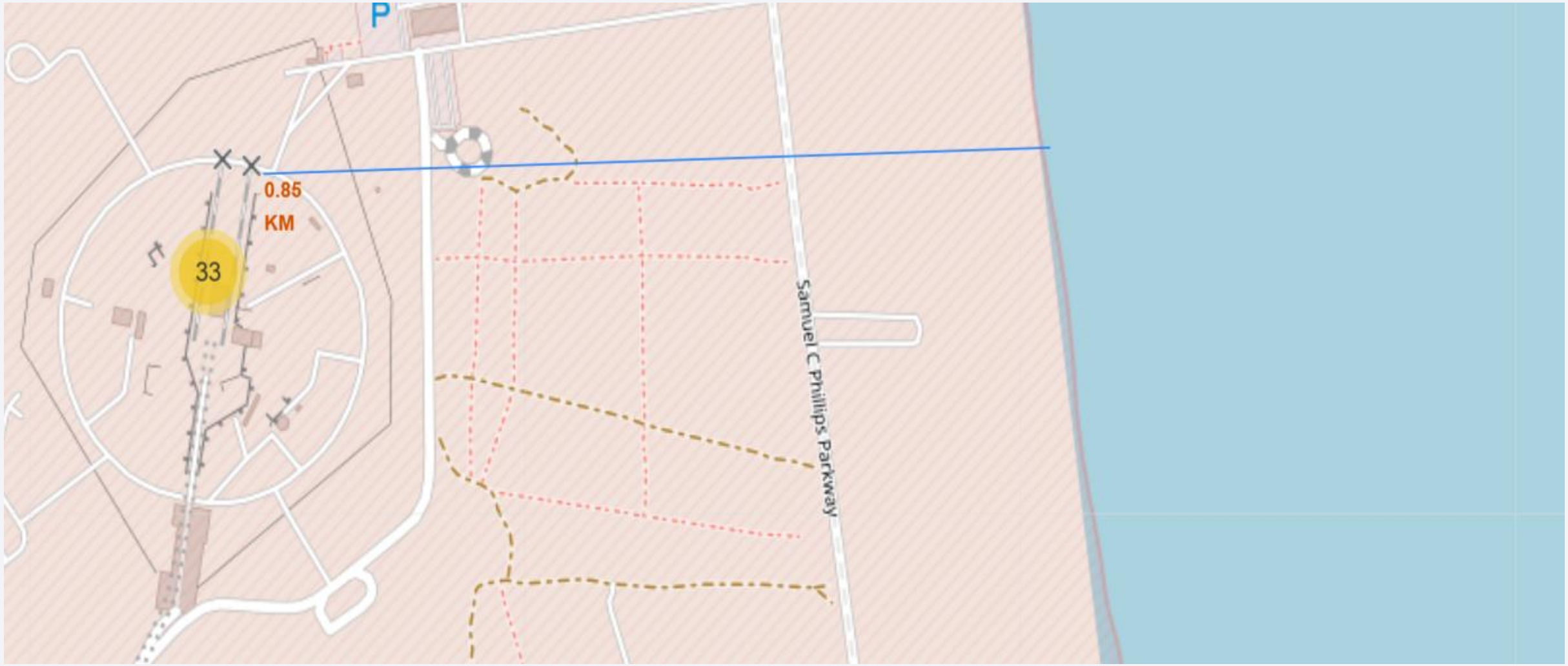
# Color Labeled Launch Outcomes

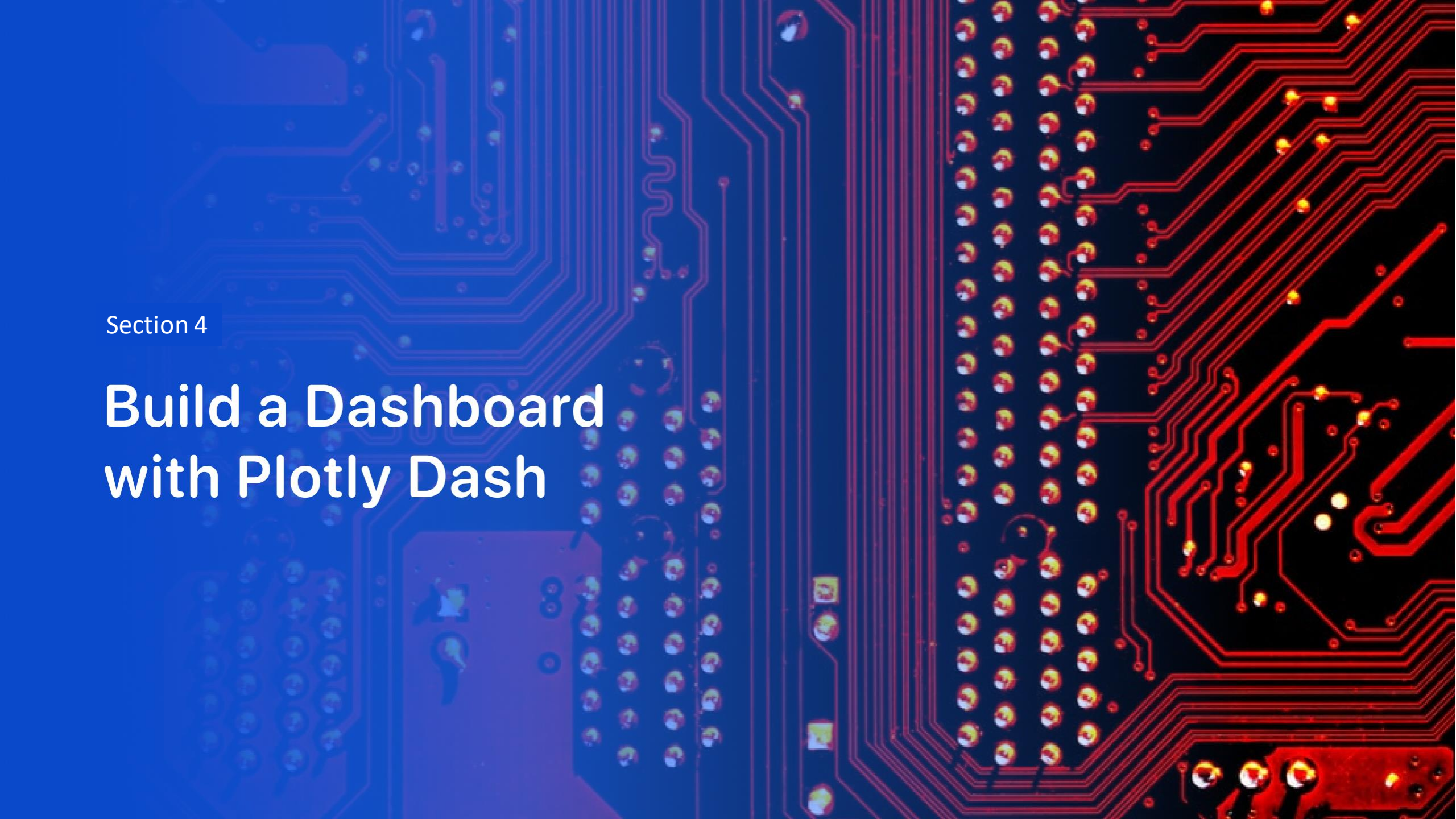
- We can easily identify which launch sites have relatively high success rates.



# Distance between Coastline and the Launch site

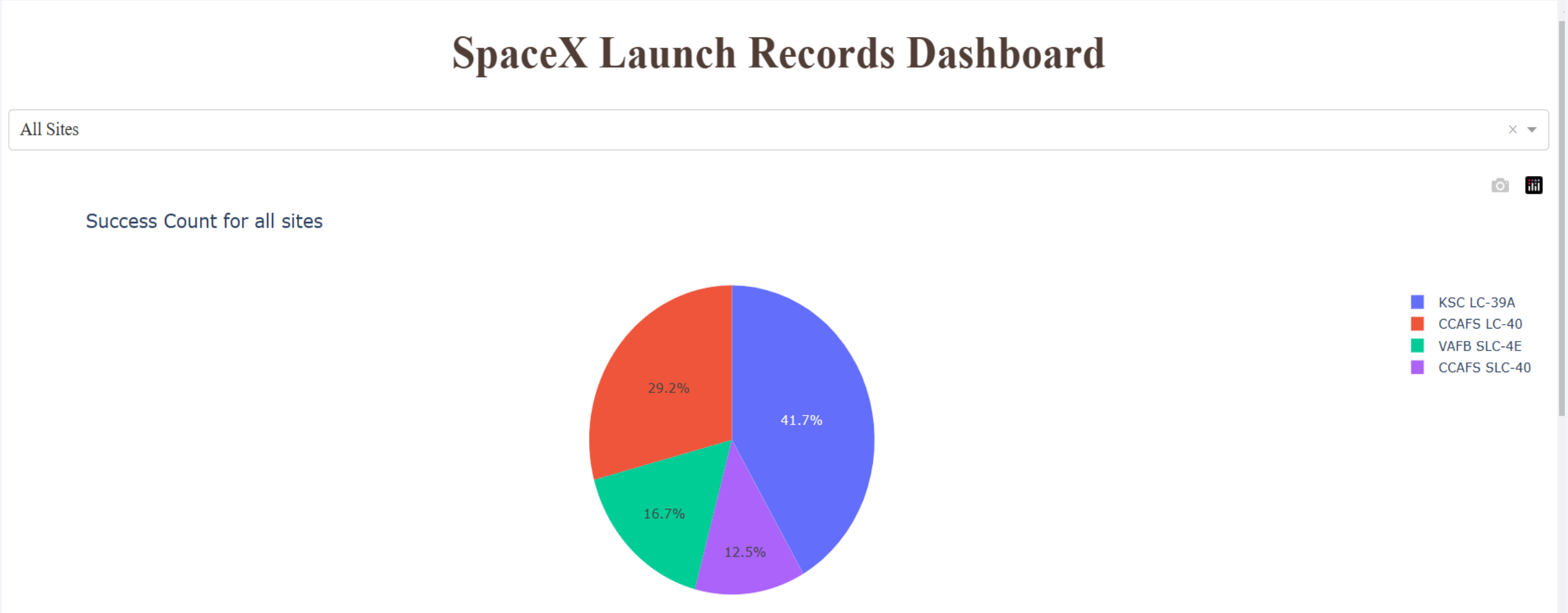
- Distance between coastline and the launch site CCAFS SLC-40 is 0.85 Km




Section 4
Build a Dashboard with Plotly Dash

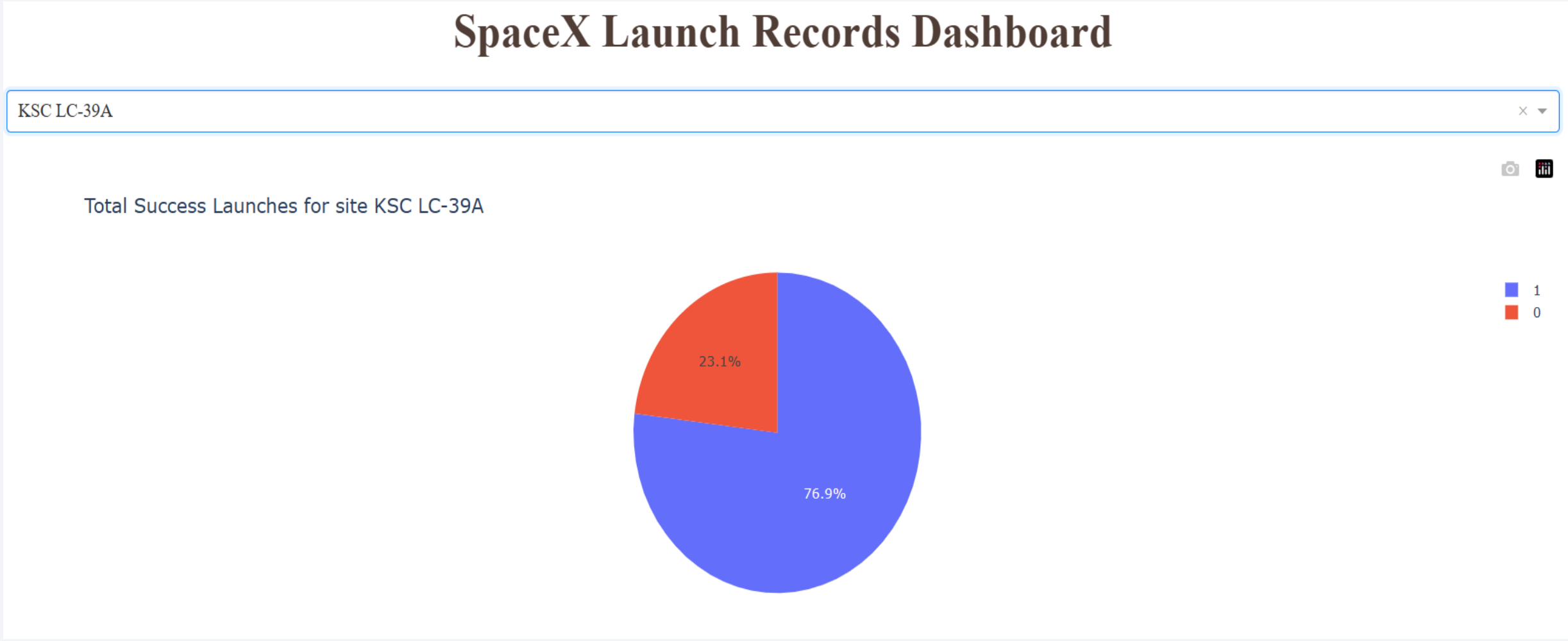
# Launch success count for all sites

- KSC LC-39A has the most success rate
- VAFB SLC-40 has the less success rate



# Launch site with highest success rate

- KSC LC –39A has the success rate of 76.9 %



# Pay load vs Launch Outcome

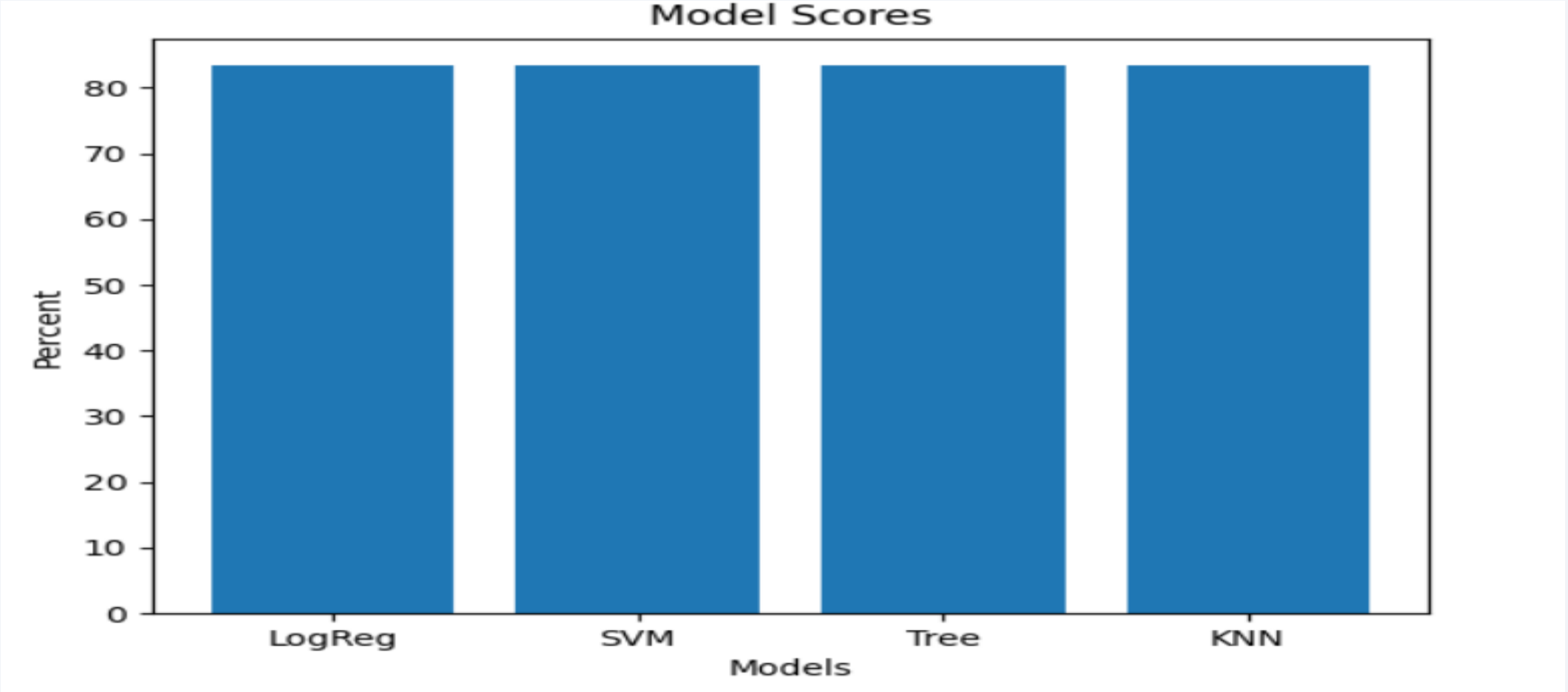
- 2000 to 10000kg payload range has the highest success rate

Section 5

# Predictive Analysis (Classification)

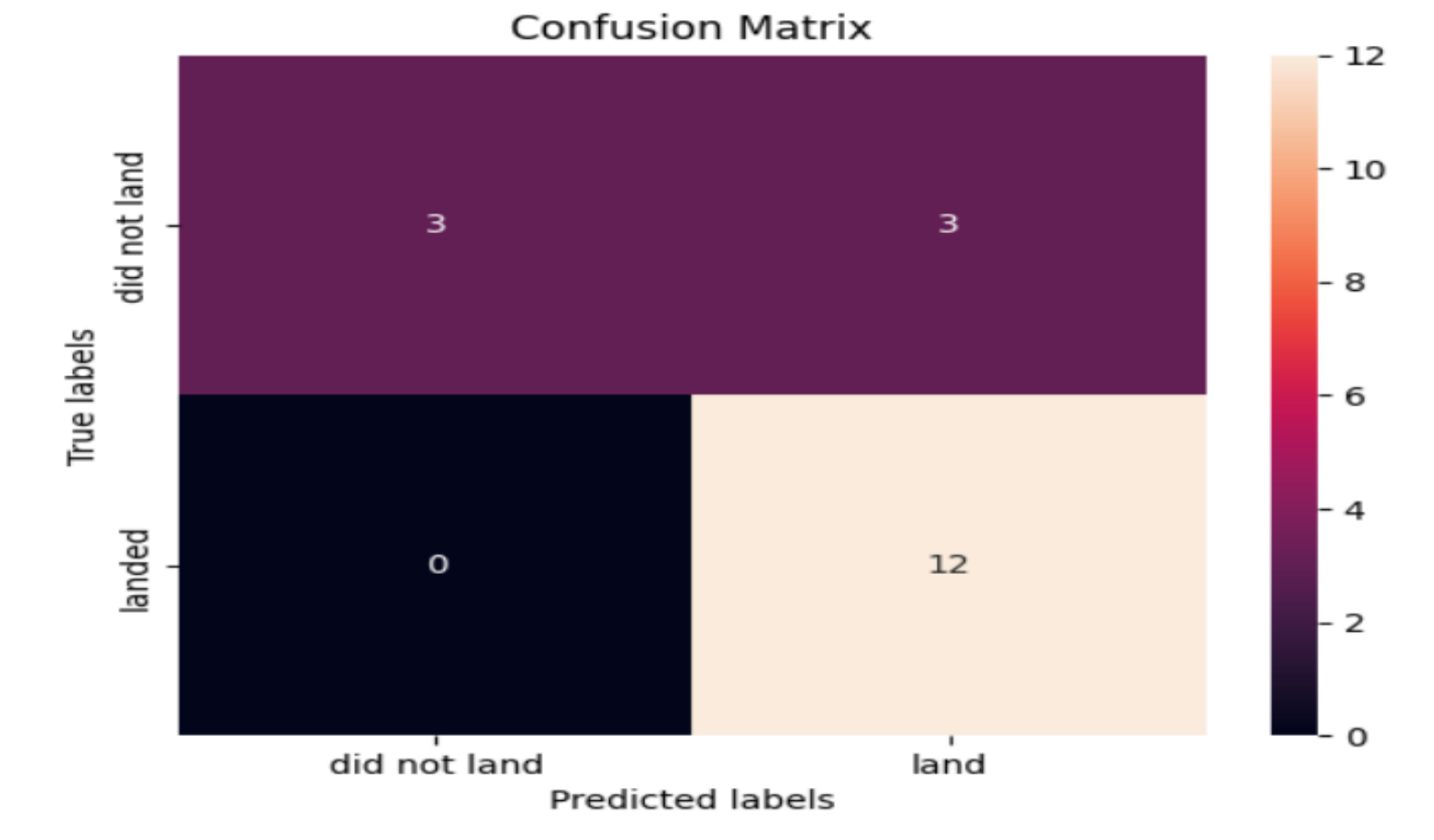
# Classification Accuracy

All Models performs equally



# Confusion Matrix

- All models performs equally and their confusion matrix is also same



# Conclusions

- We have identified that the payload and payload mass are primary variables that affect successful rates
- Higher the pay load mass , higher the success rate.
- Higher the number of flights , higher the success rate.
- ES-L1,HEO,GEO,SSO has the highest success rates among the other launching sites.

# 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!