IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Anu Ramesh
12/16/2024

# Outline

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

# Executive Summary

## Summary of methodologies

- Data collection
- Data wrangling
- Exploratory Data Analysis with Data Visualization
- Exploratory Data Analysis using SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive Analysis (Classification)

## Summary of all results

- Exploratory Data Analysis results
- Interactive analytics demo screenshots
- Graphs generated by plotly dash, Seaborn and Folium maps
- Predictive analysis results

# Introduction

## Project background and context

SpaceX is a successful company in sending spacecraft to the space. The company advertises 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Sometimes the first stage does not land. Sometimes it will crash. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Other times, Space X will sacrifice the first stage due to the mission parameters like payload, orbit, and customer.

## Problems that need answers

- Train a machine learning model to predict whether the first stage will land or not.
- How the mission parameters like payload, orbit, and customer affects the landing.
- Does the space station has any effect on the landing.
- Use public information to predict if SpaceX will reuse the first stage.

Section 1

# Methodology

# Methodology

## Executive Summary

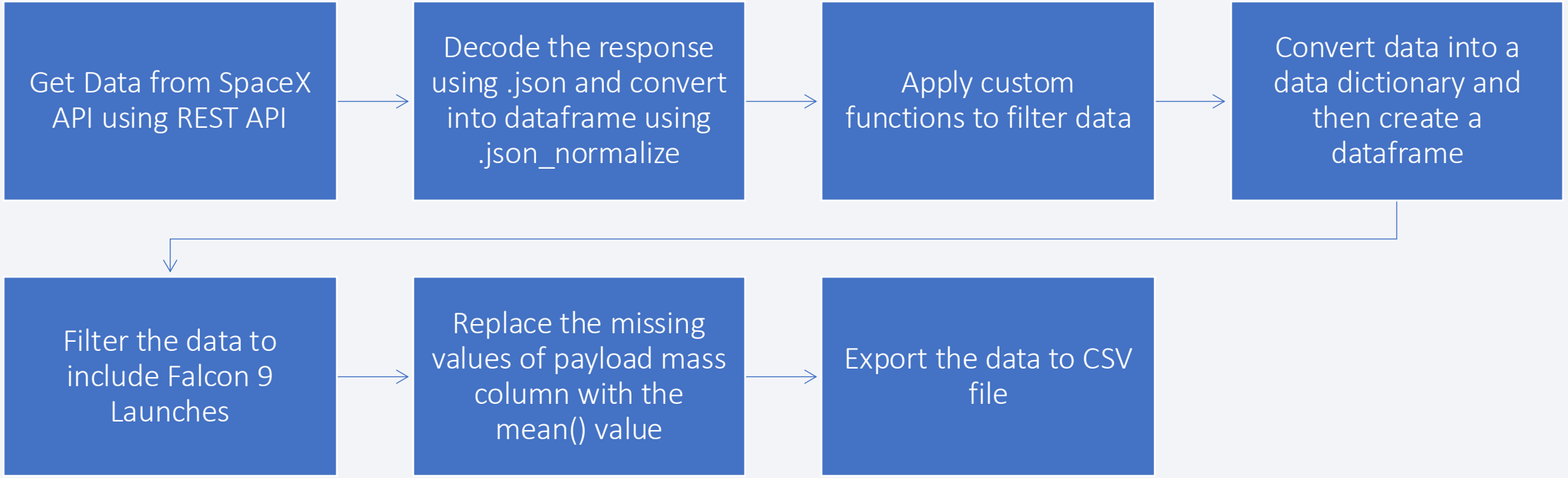
- Data collection methodology
  - Using SpaceX REST API
  - Using Webscraping
- Perform data wrangling
  - Filtering the data
  - Cleaning the data for missing values
  - Prepare the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

# Data Collection

- Data collection process involved a combination of API requests from SpaceX REST API and Web Scraping data from a table in SpaceX's Wikipedia entry.
- The REST Api endpoint used for data is ***api.spacexdata.com/v4/launches/past***
- We had to use both REST API and WebScraping data collection methods in order to get complete information about the launches for a more detailed analysis.
- Data Columns are obtained by using SpaceX REST API:
    - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude Data Columns are obtained by using Wikipedia Web Scraping: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time
- Data Columns are obtained by using Wikipedia Web Scraping:
    - Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

# Data Collection – SpaceX API

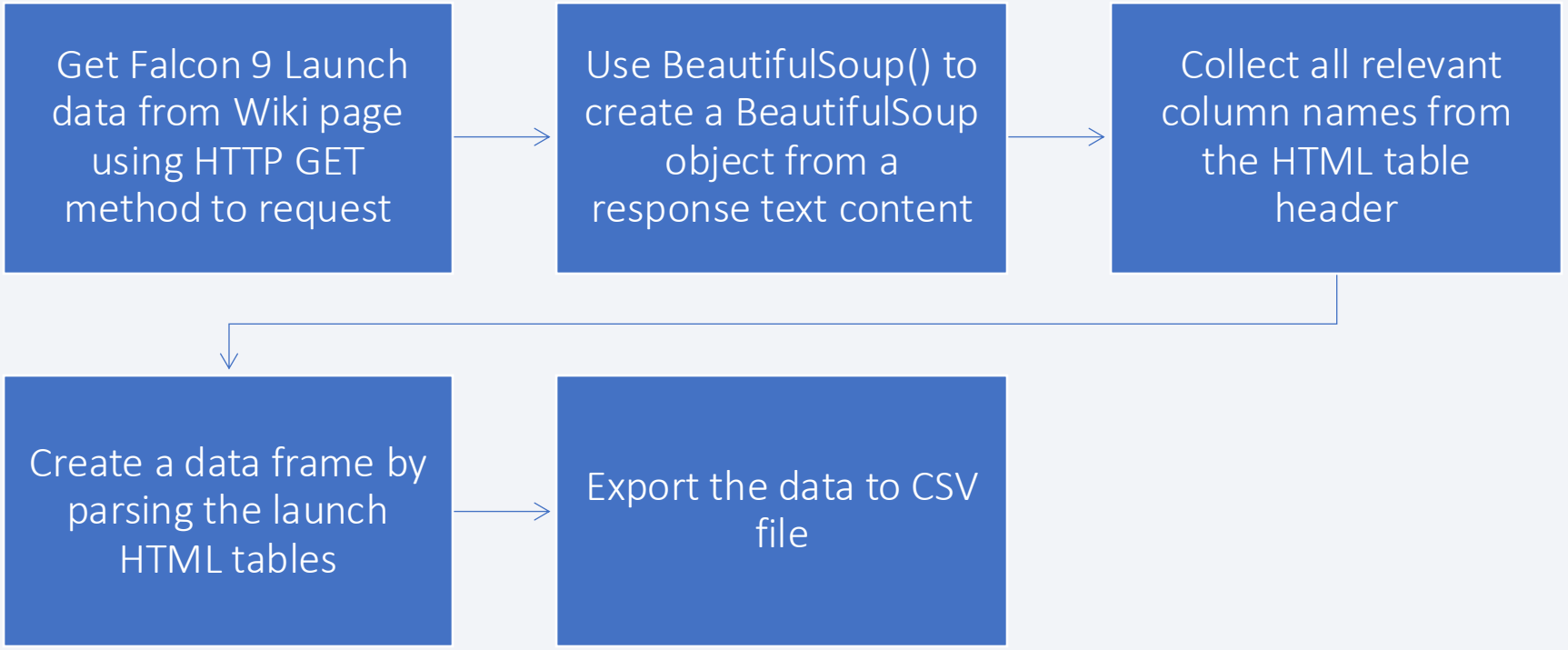
## 1. Data Collection Flowchart




## 2. GitHub URL - Data Collection Using API

# Data Collection – WebScraping

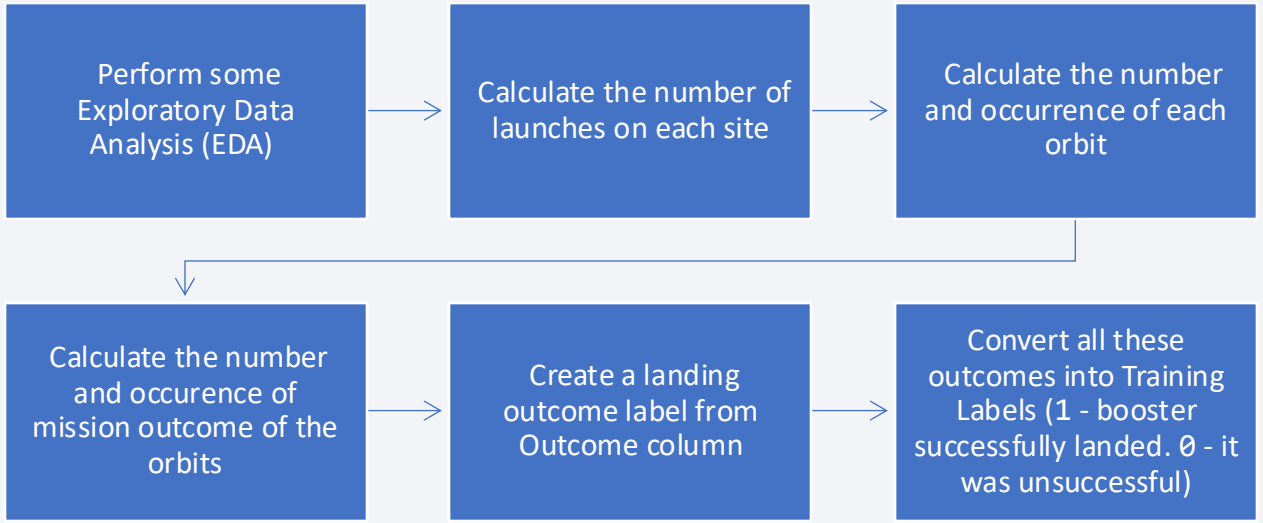
1. Data collecting using Webscraping flowchart




2. GitHub URL - Web Scraping Lab

# Data Wrangling

- **Exploratory Data Analysis (EDA):** We performed EDA to identify patterns in the dataset and understand the distribution of landing outcomes.
- **Outcome Categorization:** The dataset contained multiple landing outcome categories: ocean, ground pad (RTLS), and drone ship (ASDS), with each marked as either successful (True) or unsuccessful (False).
- **Label Conversion:** Each landing outcome (True/False) was converted into a binary label: 1 for successful landings and 0 for unsuccessful ones.
- **Data Transformation:** The original outcome categories were mapped to these binary labels to prepare the dataset for supervised model training.
- **Final Dataset:** The processed data now consists of binary labels indicating whether each booster landing attempt was successful (1) or unsuccessful (0).




GitHub URL - Data Wrangling

# EDA with Data Visualization

Plotted Charts

- *Scatter Plot* to visualize the relationship between Flight Number and Launch Site
- *Scatter Plot* to visualize the relationship between Payload Mass and Launch Site
- *Bar Plot* to visualize the relationship between success rate of each orbit type
- *Scatter Plot t*o visualize the relationship between FlightNumber and Orbit type
- *Scatter Plot t*o visualize the relationship between Payload Mass and Orbit type
- *Line Chart* to visualize the launch success yearly trend

The above charts would give the preliminary information about how each variable would affect the rate, we can select the features that can be used in training the models for prediction in the future module

GitHub URL - EDA With Data Visualization

# EDA with SQL (Part 1)

SQL Queries performed

- Display the names of the unique launch sites in the space mission
  - %sql select distinct(Launch_Site) from SPACEXTABLE
- Display 5 records where launch sites begin with the string 'CCA'
  - %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
- Display the total payload mass carried by boosters launched by NASA (CRS)
  - %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer='NASA (CRS)'
- Display average payload mass carried by booster version F9 v1.1
  - %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like 'F9 v1.1%'
- List the date when the first succesful landing outcome in ground pad was acheived.
  - %sql select min(Date) from SPACEXTABLE WHERE Landing_Outcome='Success'

GitHub URL - EDA With SQL Lab

# EDA with SQL (Part 2)

- 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 from SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
- List the total number of successful and failure mission outcomes
  - %sql select count(*) from SPACEXTABLE WHERE Mission_Outcome like 'Success%'
  - %sql select count(*) from SPACEXTABLE WHERE Mission_Outcome like 'Failure%'
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - %sql select Booster_Version from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = ( select max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE )
- 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.
  - %sql select substr(Date, 6,2) as MonthName, Booster_Version, Launch_Site from SPACEXTABLE WHERE landing_outcome = 'Failure (drone ship)' and substr(Date,0,5)='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.
  - %sql select a.*, b.* from ( (select count(landing_outcome) as FAIL_DRONE_SHIP from SPACEXTABLE WHERE landing_outcome ='Failure (drone ship)' and date > '2010-06-04' and date < '2017-03-20') a, (select count(landing_outcome) as SUCCESS_GROUND_PAD from SPACEXTABLE WHERE landing_outcome ='Success (ground pad)' and date > '2010-06-04' and date < '2017-03-20') b)
  - %sql select count(landing_outcome) as SUCCESS_GROUND_PAD from SPACEXTABLE WHERE landing_outcome ='Success (ground pad)' and date > '2010-06-04' and date < '2017-03-20'

GitHub URL - EDA With SQL Lab

# Build an Interactive Map with Folium

- Marked all launch sites on a map
  - Added markers with circle, label, and Text Label using latitude and Longitude and NASA Space center as starting location
  - Marked all launch sites using latitude and longitude to show their proximity to equator and coasts
- Marked the success/failed launches for each site on the map
  - Added color markers to differentiate the success (Green) and Failure(Red) launches using marker cluster
- Calculate the distances between a launch site to its proximities
  - Added colored Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City.

After completing the above tasks, I was able to find some geographical patterns about launch sites.

GitHub URL - Folium Map

# Build a Dashboard with Plotly Dash

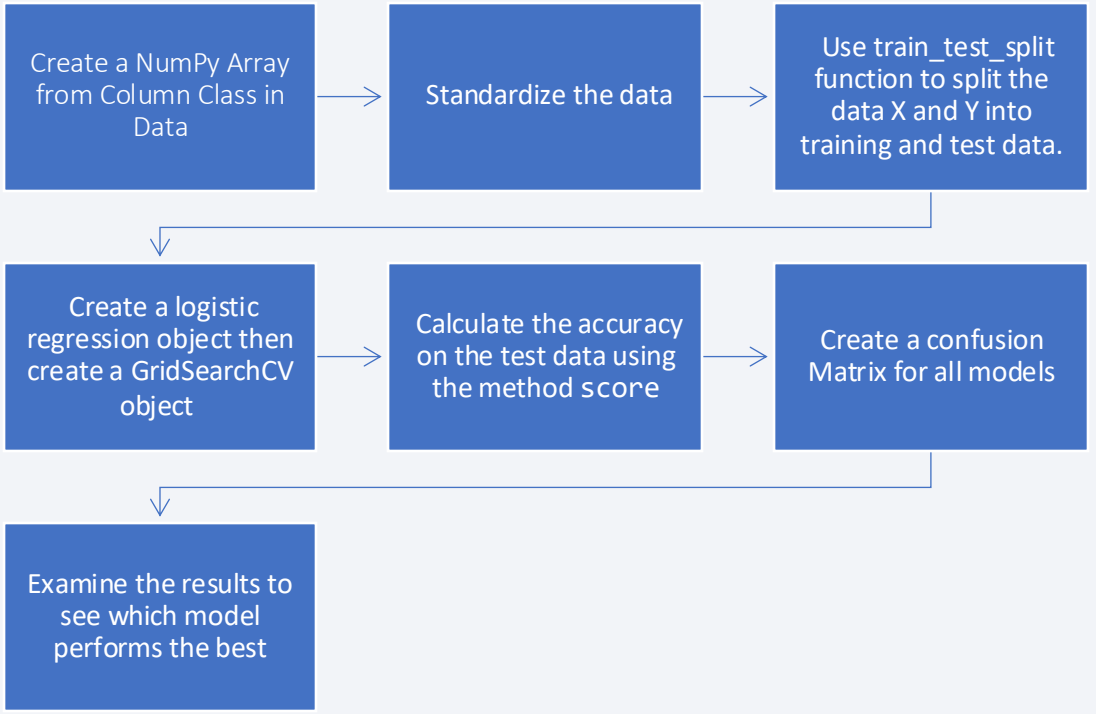
- Added a dropdown list to enable Launch Site selection.
- Pie Chart showing Success Launches (All Sites/Certain Site) - A pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site for the selected Launch Site
- Added a slider to select Payload range.
- If we adjust the slider, Scatter Chart of Payload Mass vs. Success Rate for different Booster Versions would be displayed
- Added a scatter chart to show the correlation between Payload and Launch Success

**The slider, dropdown and scatter plots were added to find the** correlation between payload Mass and Launch success

GitHub URL - SpaceX Dash App

# Predictive Analysis (Classification)

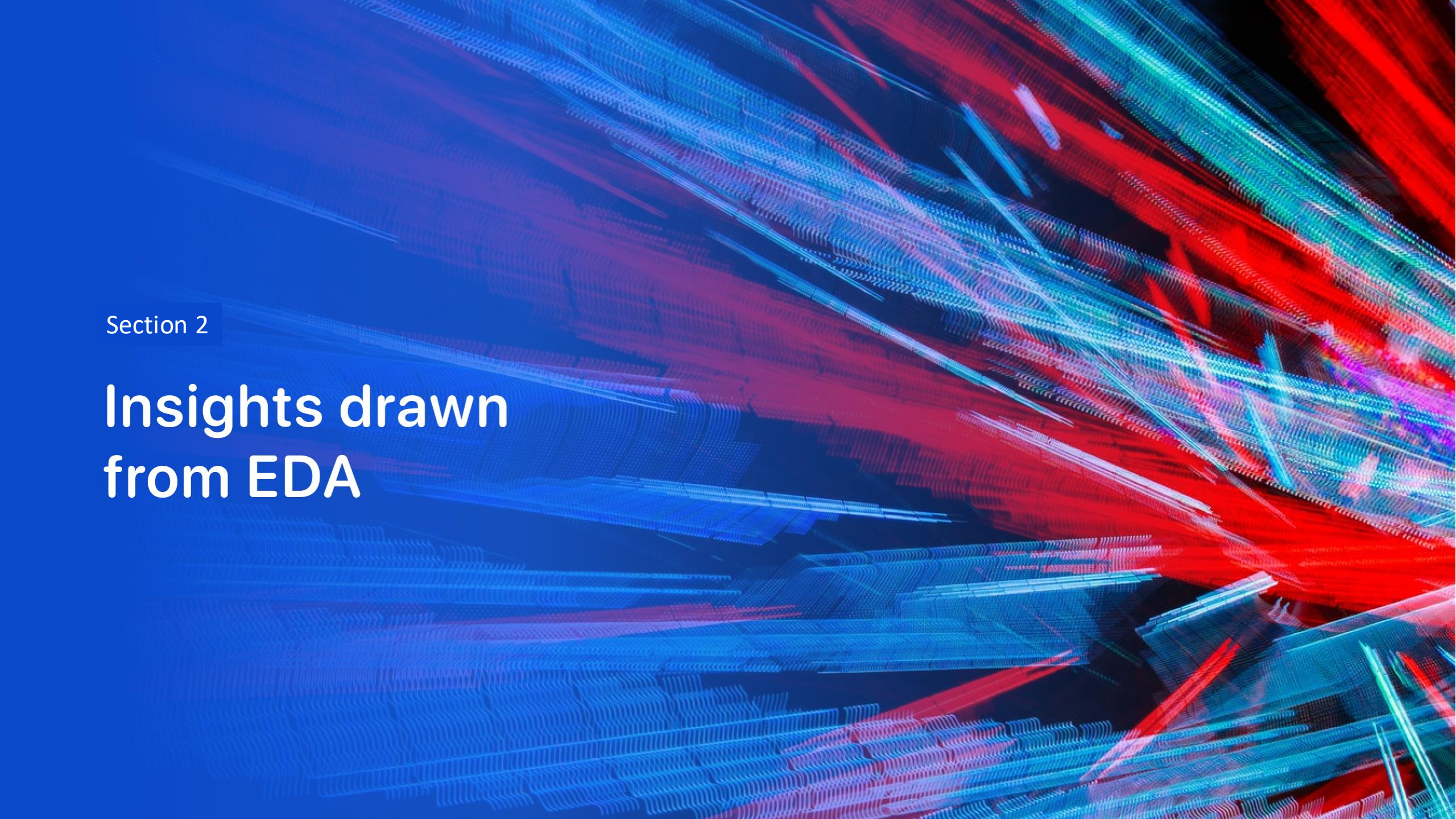
- Developed a machine learning pipeline to predict the success of Falcon 9 first-stage landings.
- Preprocessed the data to standardize it and applied train-test split to create training and testing datasets.
- Trained multiple models: Logistic Regression, Support Vector Machines, Decision Tree Classifier, and K-Nearest Neighbors.
- Performed Grid Search to find optimal hyperparameters for each algorithm and selected the best-performing model based on accuracy.
- Evaluated model performance using a confusion matrix to assess classification results.



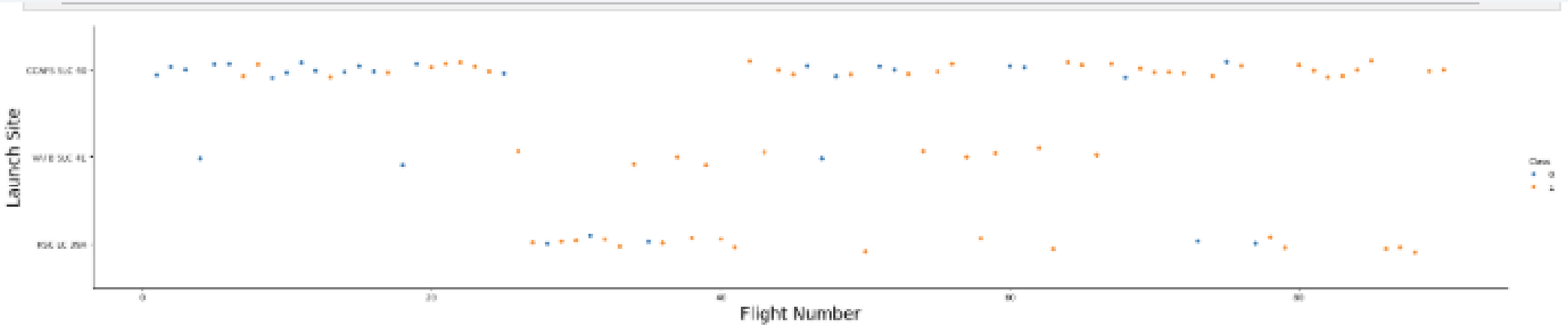

GitHub URL - [SpaceX Machine Learning Predictive Analysis Lab]

# Results

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


Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site (Scatter Plot)

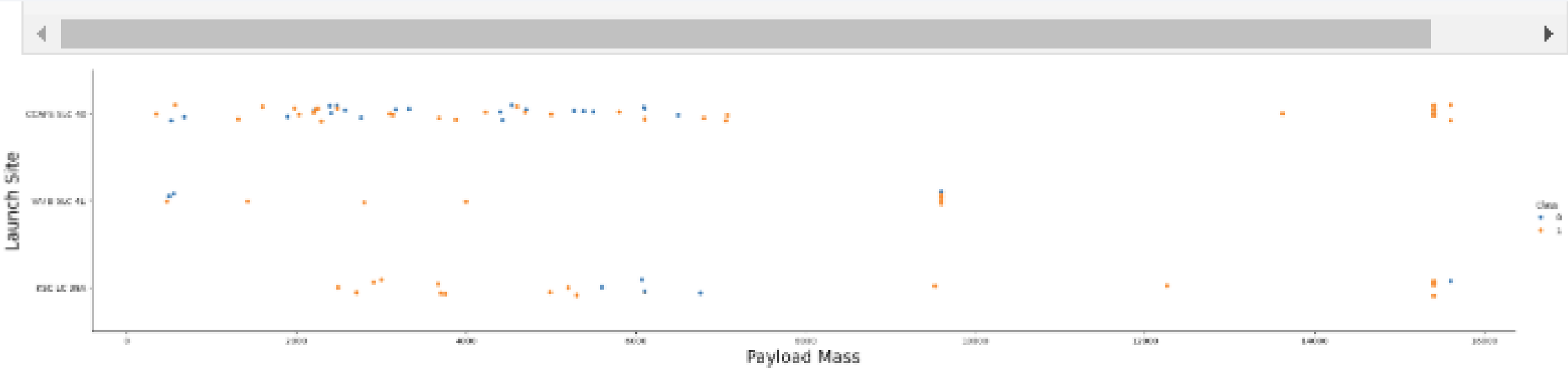



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

## Explanation

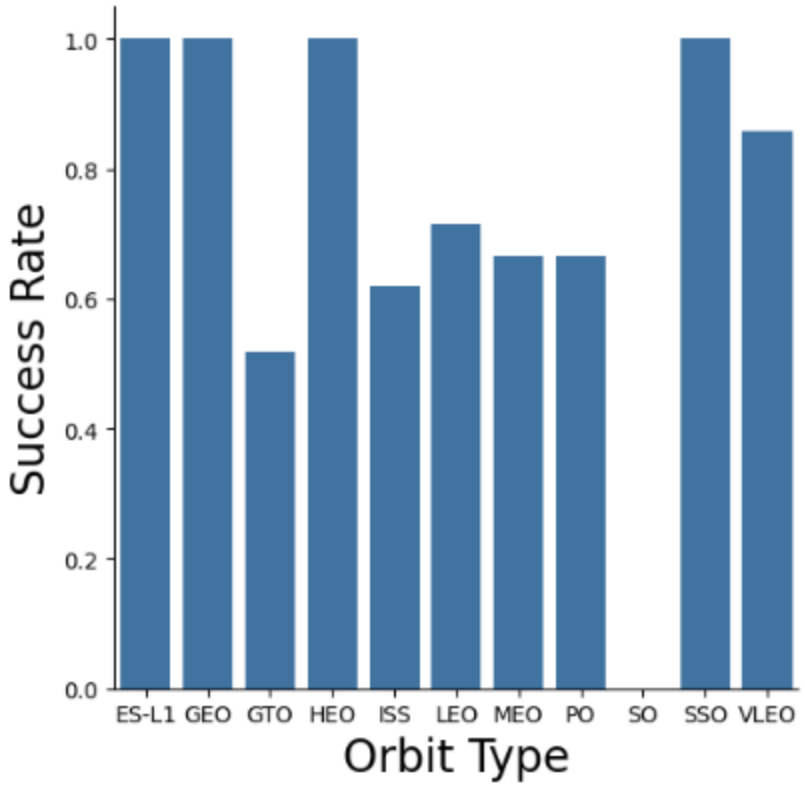
- Most of the flights were launched from CCAF5 SLC 40 Launch Site.
- The landings of SpaceX Falcon 9 First Stage during the earliest launch times resulted in failure.
- The landings of SpaceX Falcon 9 First Stage during the latest launch times resulted in continous success.

# Payload vs. Launch Site (Scatter Plot)




Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

# Success Rate vs. Orbit Type (Bar Chart)

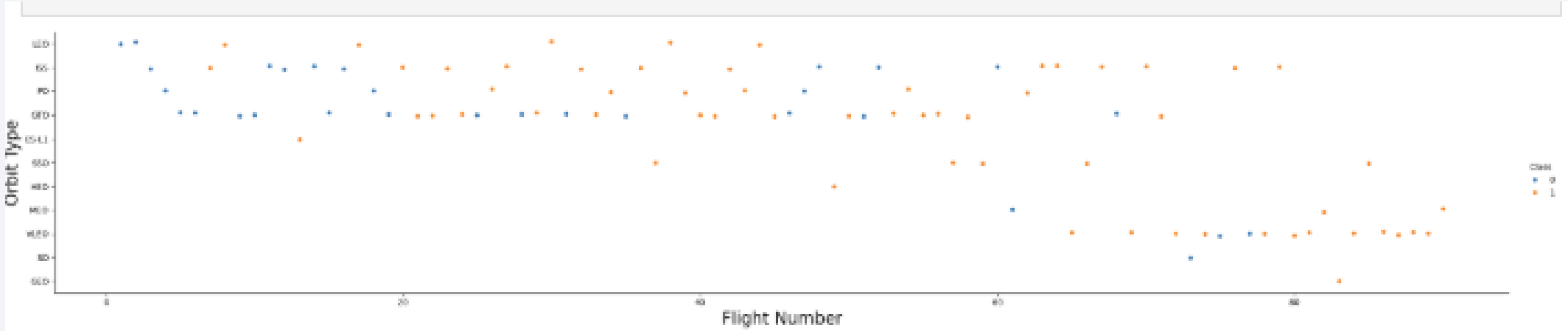



Analyze the plotted bar chart to identify which orbits have the highest success rates.

## Explanation

- Orbits with 100% success rate -ES-L1, GEO, HEO, SSO
- Orbits with 50-75% success rate are -GTO, ISS, LEO, MEO, PO
- SO Orbit has 0% success rate

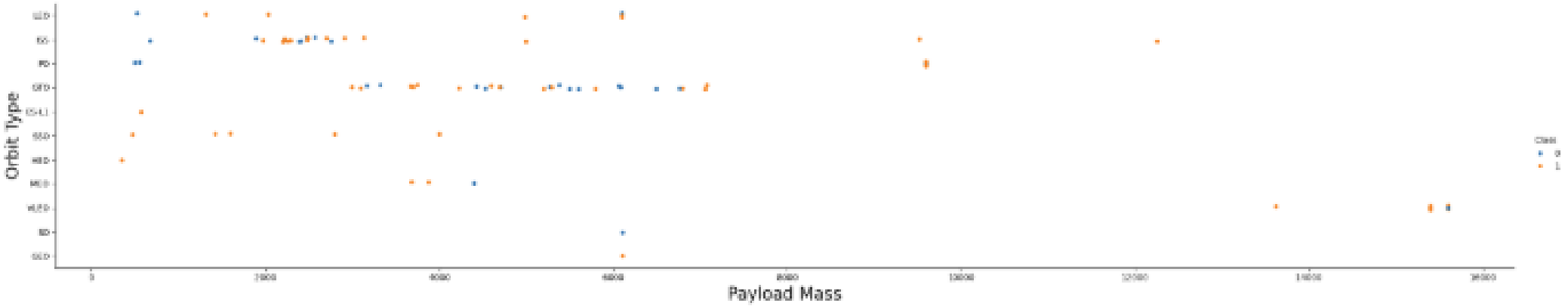
# Flight Number vs. Orbit Type




You 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

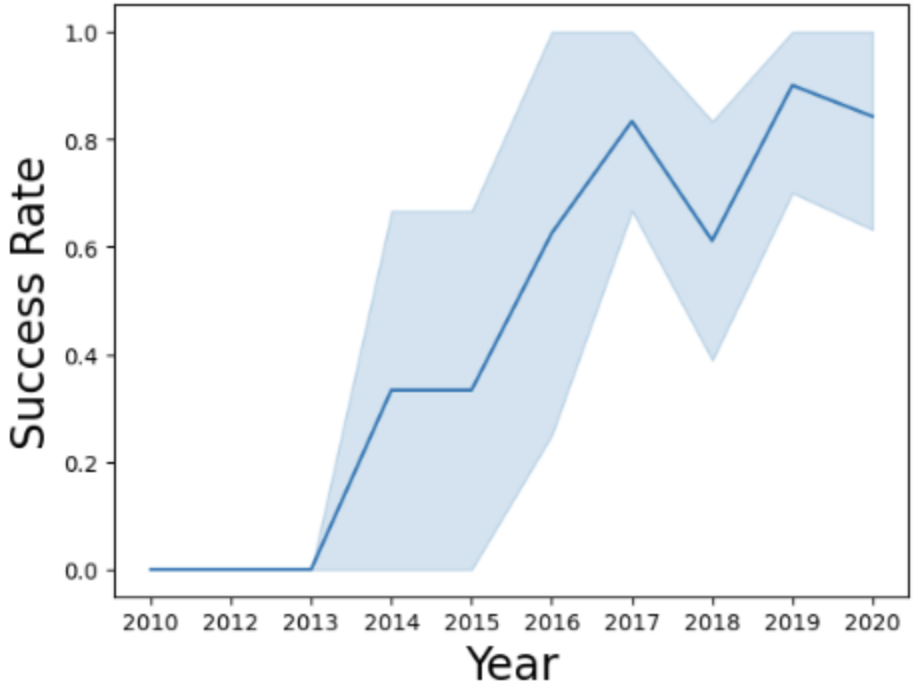
```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Orbit Type",fontsize=20)
plt.show()
```




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




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

# All Launch Site Names

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

```
[12]: %sql select distinct(Launch_Site) from  SPACEXTABLE
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
Done.
```

t[12]:

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

- Explanation – The Distinct keyword is used to find unique values of the queried column

# Launch Site Names Begin with 'CCA'

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

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

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

| 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 |

Explanation

- The SQL condition like with % in the string is used to queried values of any string that begins with the specified string

# Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer='NASA (CRS)'
```

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

| sum(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

Explanation

- The SUM() aggregate function is used to compute the sum of payload mass carried by the customer NASA (CRS)

# Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like 'F9 v1.1%'
```

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

| avg(PAYLOAD_MASS__KG_) |
|---|
| 2534.6666666666665 |

Explanation

- The **AVG( )** is an **aggregate function** in SQL. It is used to calculate the **average** of a numeric column, in this case **PAYLOAD_MASS__KG_**

# First Successful Ground Landing Date

**Task 3**

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

*Hint:Use min function*

```
%sql select min(Date) from SPACEXTABLE WHERE Landing_Outcome='Success'
```

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

| min(Date) |
| --- |
| 2018-07-22 |

Explanation

- The **min()** function on the date column returns the first date that matches the condition.
- Here we are using **Landing_Outcome='Success'** as condition to determine the first successful landing

# 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 from SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 an
```

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

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

## Explanation

- The **between** condition is used to select values that falls between 2 numerical values.
- Here we are using **PAYLOAD_MASS__KG_** value to filter the data between 4000 and 6000

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql select count(*) from SPACEXTABLE WHERE Mission_Outcome like 'Success%'
```

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

| count(*) |
|---|
| 100 |

```
%sql select count(*) from SPACEXTABLE WHERE Mission_Outcome like 'Failure%'
```

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

| count(*) |
|---|
| 1 |

## Explanation

- The **Mission_outcome** condition is used to filter the 'Success' and 'Failure' mission outcomes

# Boosters Carried Maximum Payload

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

```
%sql select Booster_Version from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = ( select max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE )
```

```
 * 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 |
| F9 B5 B1051.6 |
| F9 B5 B1060.3 |
| F9 B5 B1049.7 |

## Explanation

- Here a sub query is used with max() aggregate function to filter the maximum payload mass first, and then used as a condition to filter the Booster_version that used that 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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.**

```
%sql select substr(Date, 6,2) as MonthName, Booster_Version, Launch_Site from SPACEXTABLE WHERE landing_outcome = 'Failure
```

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

| MonthName | Booster_Version | Launch_Site |
|---|---|---|
| 01 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 |

## Explanation

- First the data is fetched for the 2015 year by using substr() function on the date field and for the landing_outcome which resulted in Failure (Drone Ship)
- Second, only the month field is extracted from date for the display along with booster_version and launch_site

# 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.

```
%sql select a.*, b.* from ( (select count(landing_outcome) as FAIL_DRONE_SHIP from SPACEXTABLE WHERE landing_outcome ='Fail
```

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

| FAIL_DRONE_SHIP | SUCCESS_GROUND_PAD |
|---|---|
| 5 | 3 |

```
%sql select count(landing_outcome) as SUCCESS_GROUND_PAD from SPACEXTABLE WHERE landing_outcome ='Success (ground pad)' and
```

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

| SUCCESS_GROUND_PAD |
|---|
| 3 |

Explanation

Here 2 inner tables are used after grouping the success and failure outcomes

Section 3

# Launch Sites Proximities Analysis

# All launch sites on a map

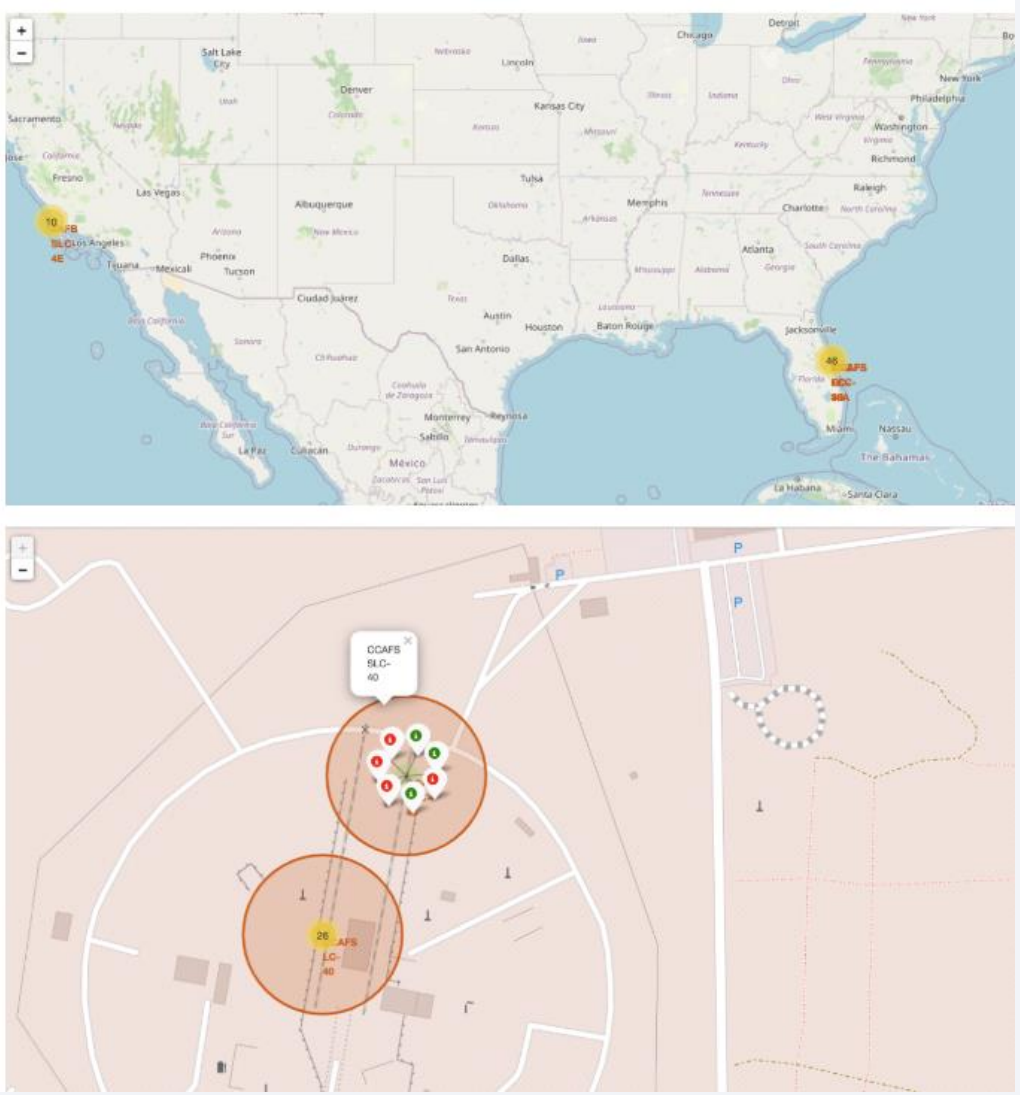
## Explanation

- Most of Launch sites are in proximity to the Equator line. The land is moving faster at the equator than any other place on the surface of the Earth.
- If a ship is launched from the equator it goes up into space, and it is also moving around the Earth at the same speed it was moving before launching.
- This speed will help the spacecraft keep up a good enough speed to stay in orbit.
- All launch sites are in very close proximity to the coast, while launching rockets towards the ocean it minimizes the risk of having any debris or falling near any domestic lives

# Success/Failed Launches for each site on the map

## Explanation

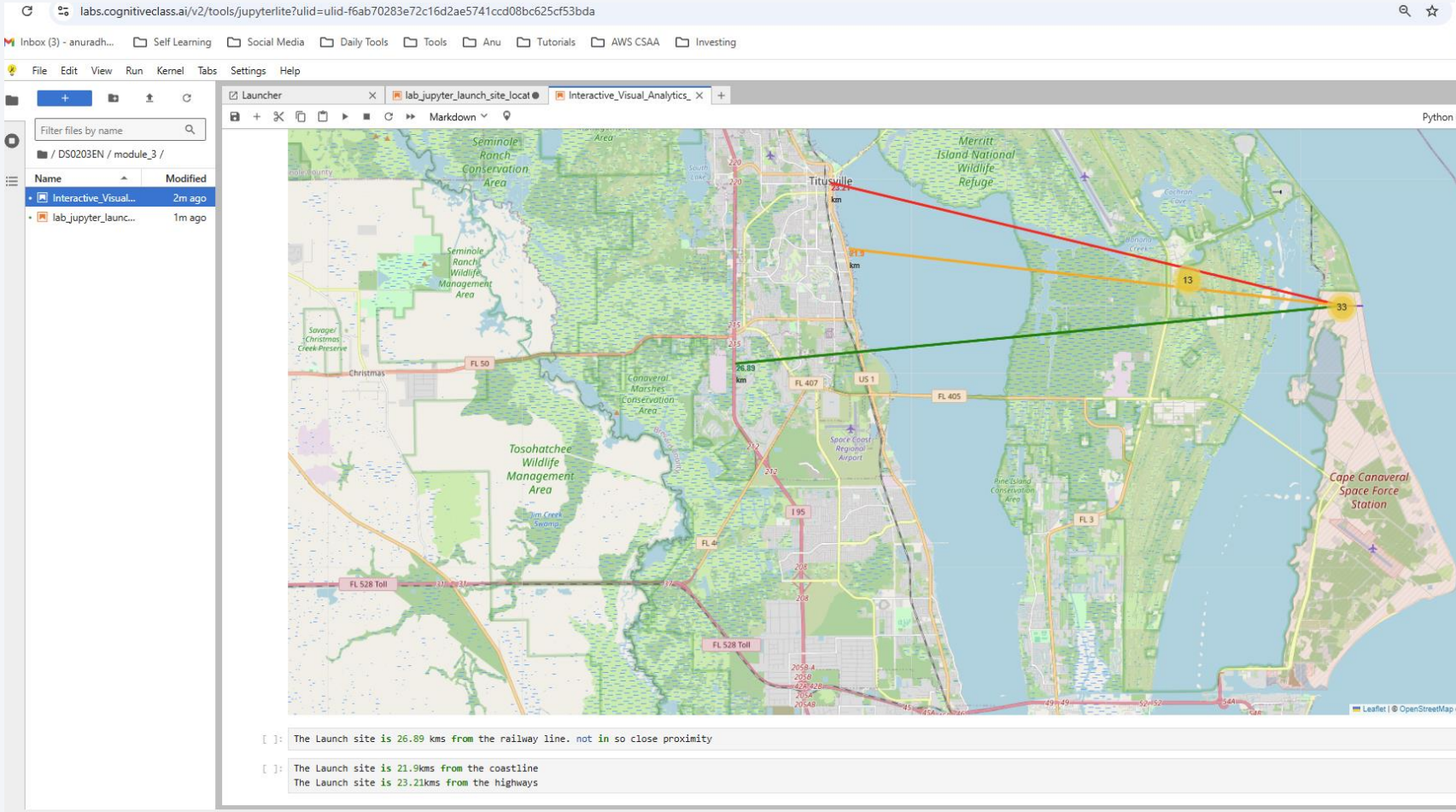
- Using folium map we can create marker clusters with different colors
- The color-labeled markers Red(Failure), Green(Success) in marker clusters we can easily identify which launch sites have relatively higher rate of success

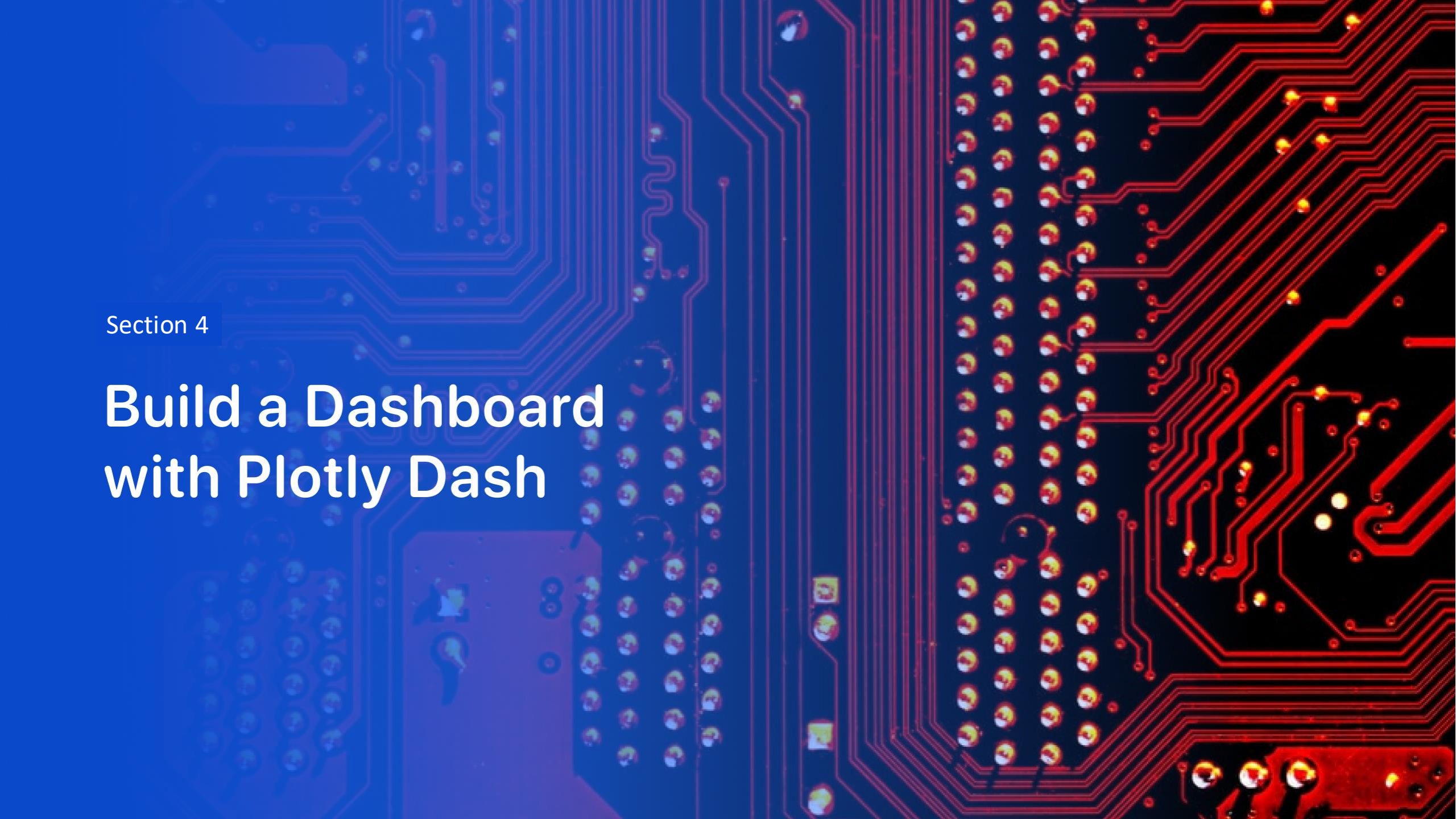


# Distance between a Launch Site and it's proximities

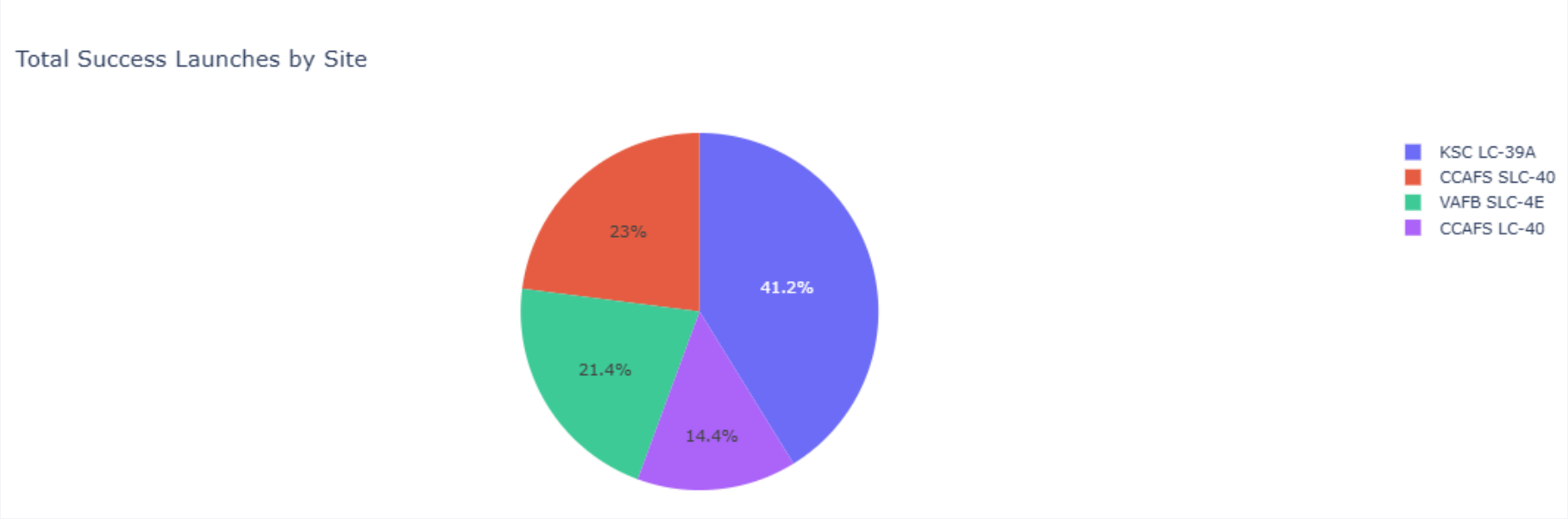
## Explanation

- The Launch site **is** 26.89 kms **from** the railway line. **not in** so close proximity
- The Launch site **is** 21.9kms **from** the coastline
- The Launch site **is** 23.21kms **from** the highways
- All three are close to Cape Canaveral city




Section 4
Build a Dashboard with Plotly Dash

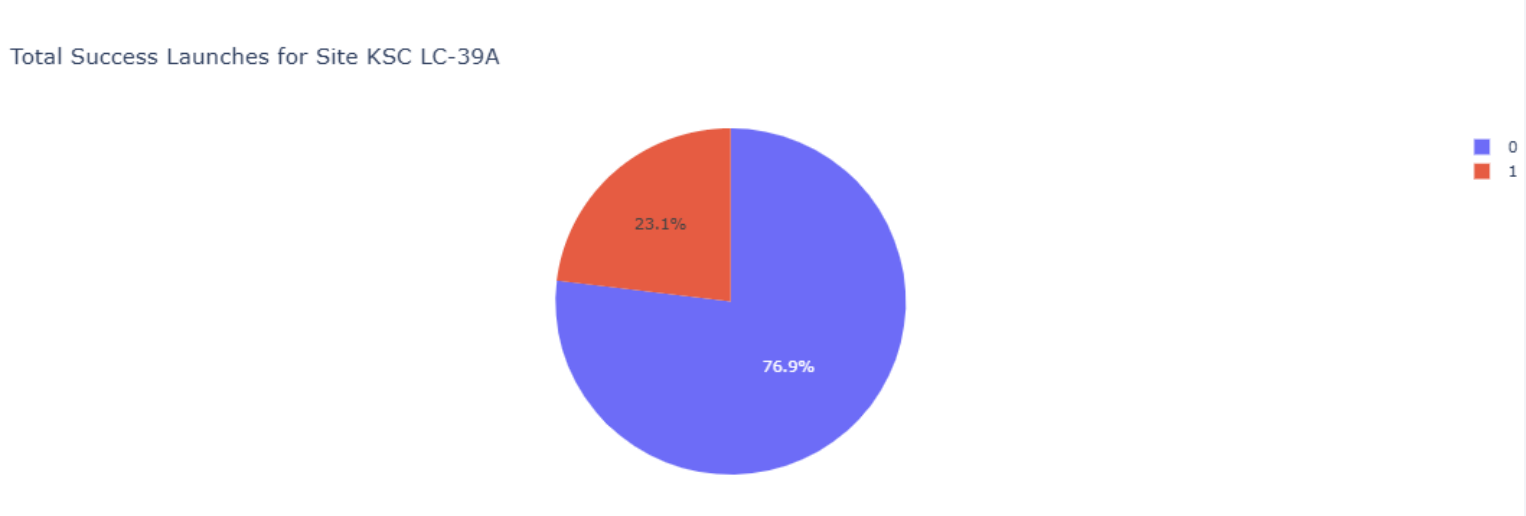
# SpaceX Launch Dashboard




## Explanation

- The Dropdown shows all the Launch Sites for selection. The default is 'All Sites'
- By Default, the Pie chart shows the success rate of all sites in different colors.
- If any of the specific launch site is selected, the pie chart will refresh and show the success rate of the selected launch site.
- From the above pie chart, the 'KSC LC-39A' has the highest success rate of launch outcomes.

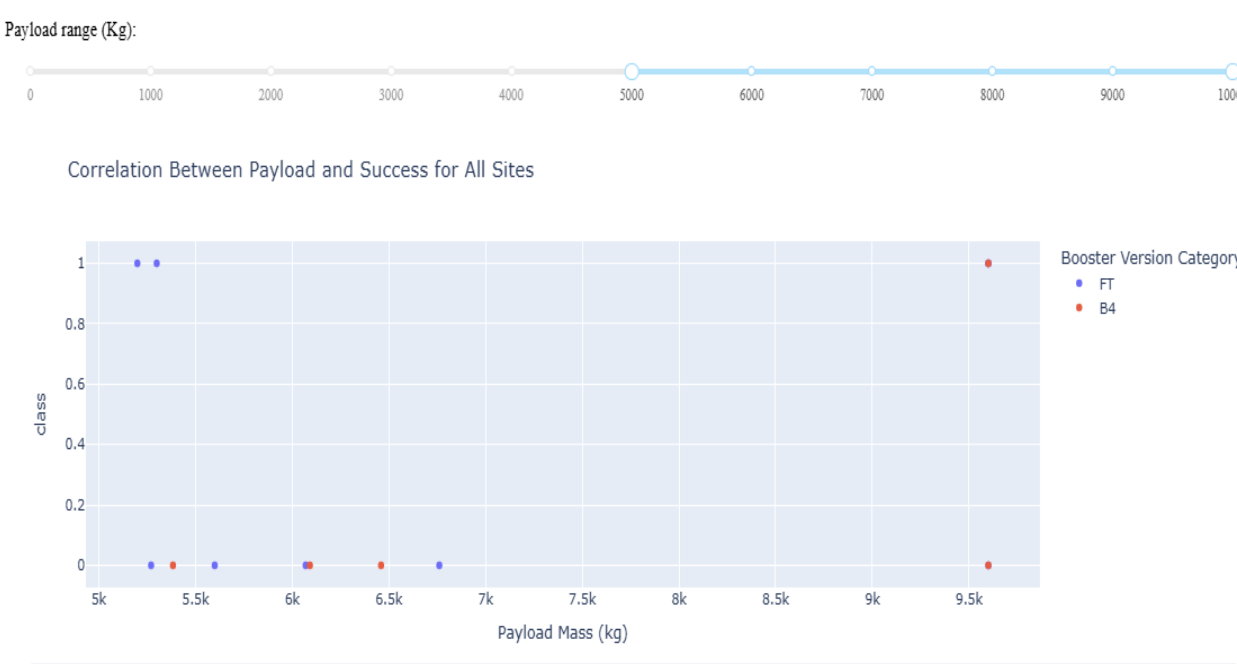
# Success Based on Selected Site




## Explanation

- The dropdown is selected with the Launch Site that has highest success rate that is 'KSC LC –39A'
- The Pie chart shows the success and failure rate for the selected launch site.
- From this selected launch site, about 76.9% of all mission launches were successful, and remaining 23.1% resulted in failure.

# Payload Vs Launch Outcome


## Explanation

- The above charts show the payload between 2000 and 6000 shows the highest success rate

Section 5

# Predictive Analysis (Classification)

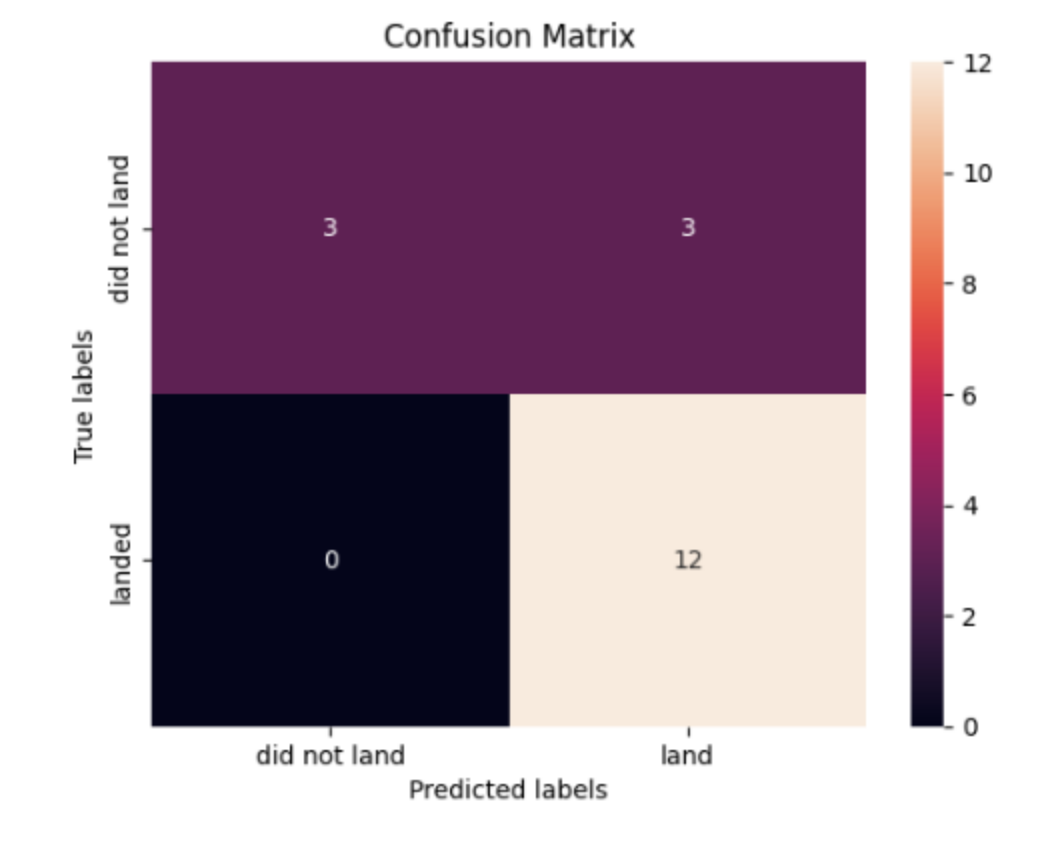
# Classification Accuracy

| | LogReg | SVM | Tree | KNN |
|---|---|---|---|---|
| Best_Score | 0.8464285714285713 | 0.8482142857142856 | 0.8767857142857143 | 0.8482142857142858 |
| Accuracy | 0.8333333333333334 | 0.8333333333333334 | 0.8333333333333334 | 0.8333333333333334 |

- **LogReg** – LogisticRegression, **SVM** – Support Vector Machine, **Tree** – Decision Tree, **KNN** – K-Nearest Neighbour
- Most models have similar results, however the 'Decision Tree Classifier' model precited with accuracy slighted above the other models.

# Confusion Matrix

- Almost all model predicted same accuracy. Despite of that, Decision tree model F1 score if slightly above the other 3 models.
- Therefore, in this given dataset I think 'Decision Tree Classifier' model prediction is the best



# Conclusions

- KSC LC-39A has the highest success rate of the launches from all the sites.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- Launches with a low payload mass show better results than launches with a larger payload mass
- Decision Tree Model is found to be the best algorithm for this dataset.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate

# Appendix

- My Final Project Files and Presentation

Special Thanks To:

- Instructors
- Coursera
- IBM

# Thank you!