IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Dinar Wahyu Rahman
August 2025

# Outline

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

# Executive Summary

- Summary of Methodologies
  - Data Collection
  - Data Wrangling
  - EDA using visualization and SQL
  - Interactive visual analytics using folium and plotly dash
  - Predictive analysis (machine learning)
- Summary of all result
  - EDA result
  - Interactive analytics in screenshot
  - Predictive analytics result

# Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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 we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
    - whats the most success lauch site?
    - does the rate of successful landings increase over years?
    - whats the best algorithm that can be used for this case?

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Uses the SpaceX API (REST) to retrieve launch data (flight number, launch site, payload, orbit, outcome).
  - Using web scraping from wikipedia.
- Perform data wrangling
  - Data is cleaned: changing data types, handling missing values, and merging multiple data sources.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

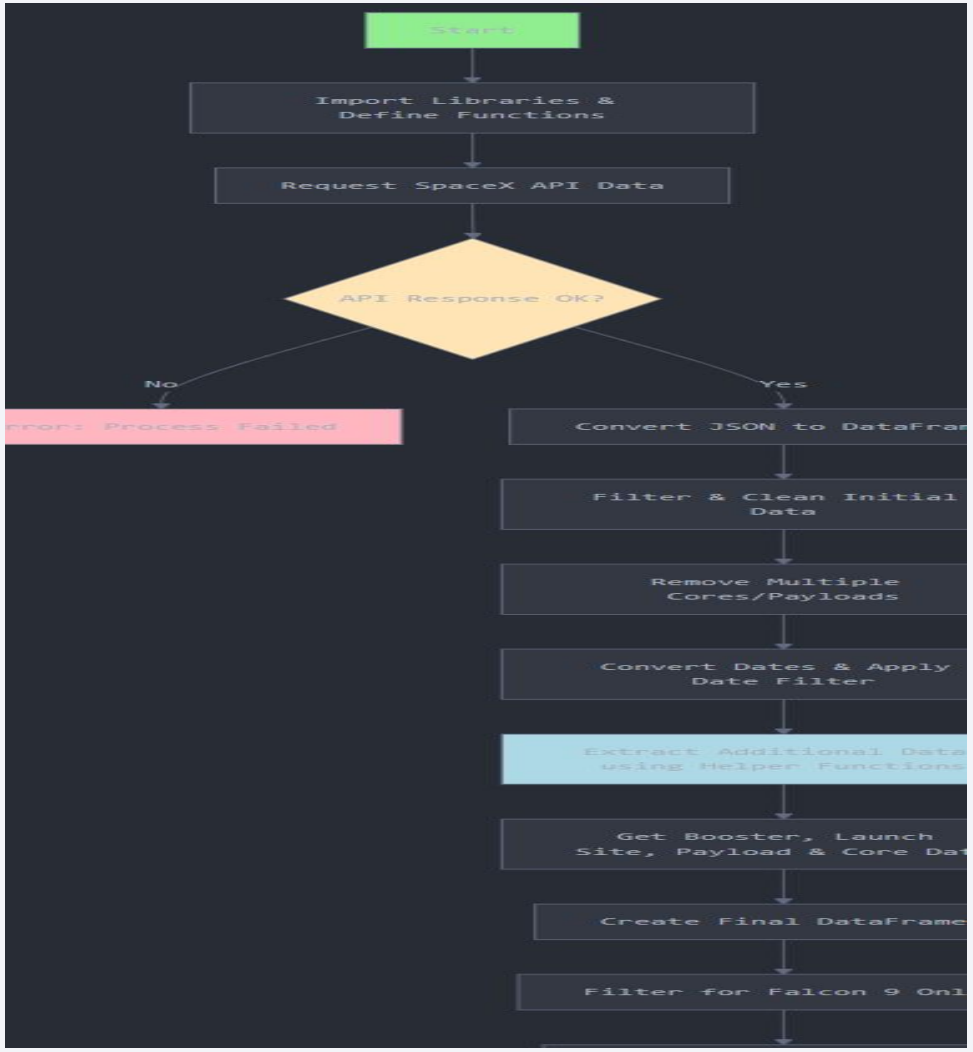
# Data Collection

- The Data was collecting using 2 methods
  - Data collection using get request to the SpaceX API
    - Decode that response content as a Json using .json() function call and turn it into pandas dataframe using .json_normalize().
    - After that, cleaned the data, checked for missing values and fill in missing value where necessary.
    - Data columns are obtained by using SpaceX API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcom, Flights, GridFins, Reused, Legs. LandingPad, Block. ReusedCount, Serial, Longitude, Latitude.
  - Data collection using web scraping from Wikipedia for Falcon 9
    - Web scraping from Wikipedia for Falcon 9 record with library BeautifulSoup.
    - The objective wa to extract the launch record as HTML table, parse the table and convert it to pandas dataframe for analysis.
    - Data columns are obtained by using web scraping website with BeautifulSoup: Flight No., Date and time ( ), Launch site, Payload, Payload mass, Orbit, Customer, Launch outcome.

# Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

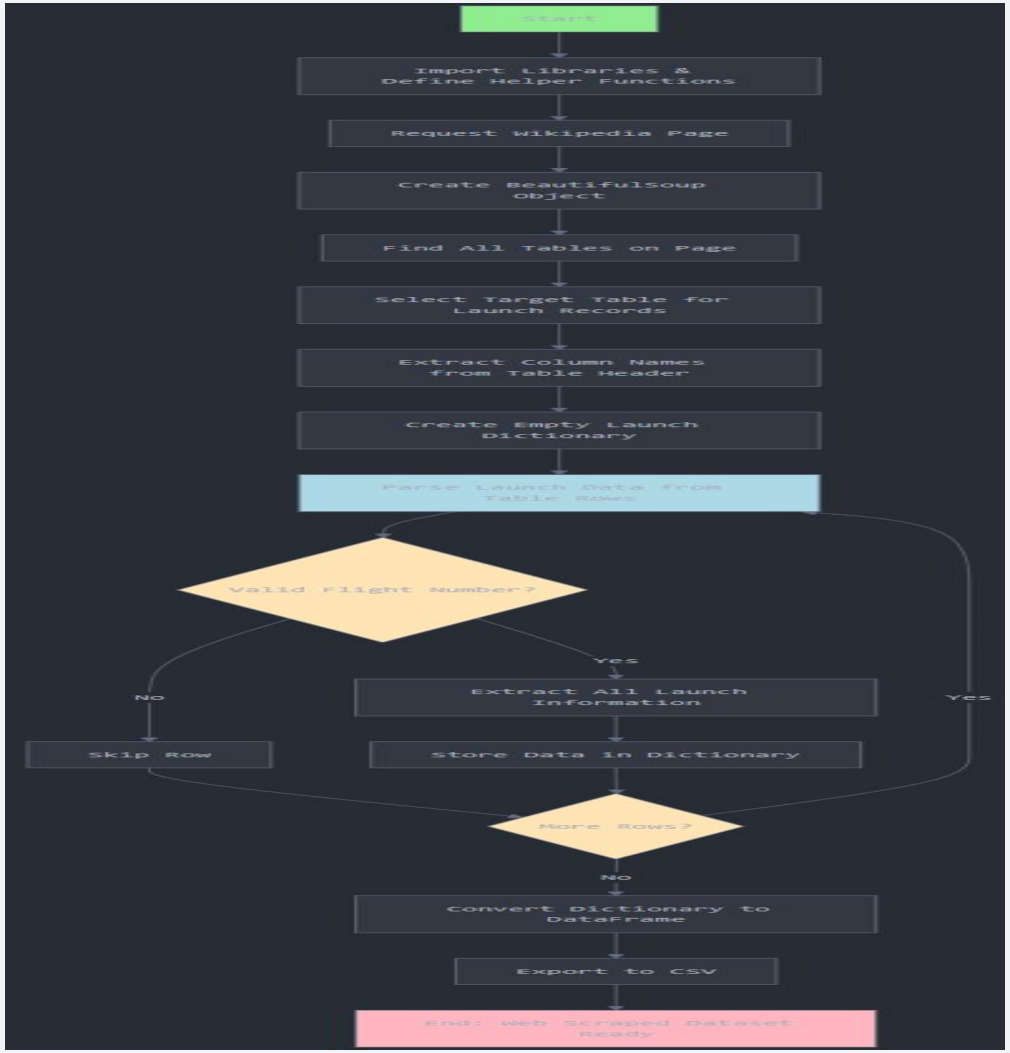
- Data Collection SpaceX API - GitHub
- Flowchart Data Collection using SpaceX API



# Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Data Collection Web Scraping from Wikipedia
- Flowchart Data Collection using Web Scraping



# Data Wrangling

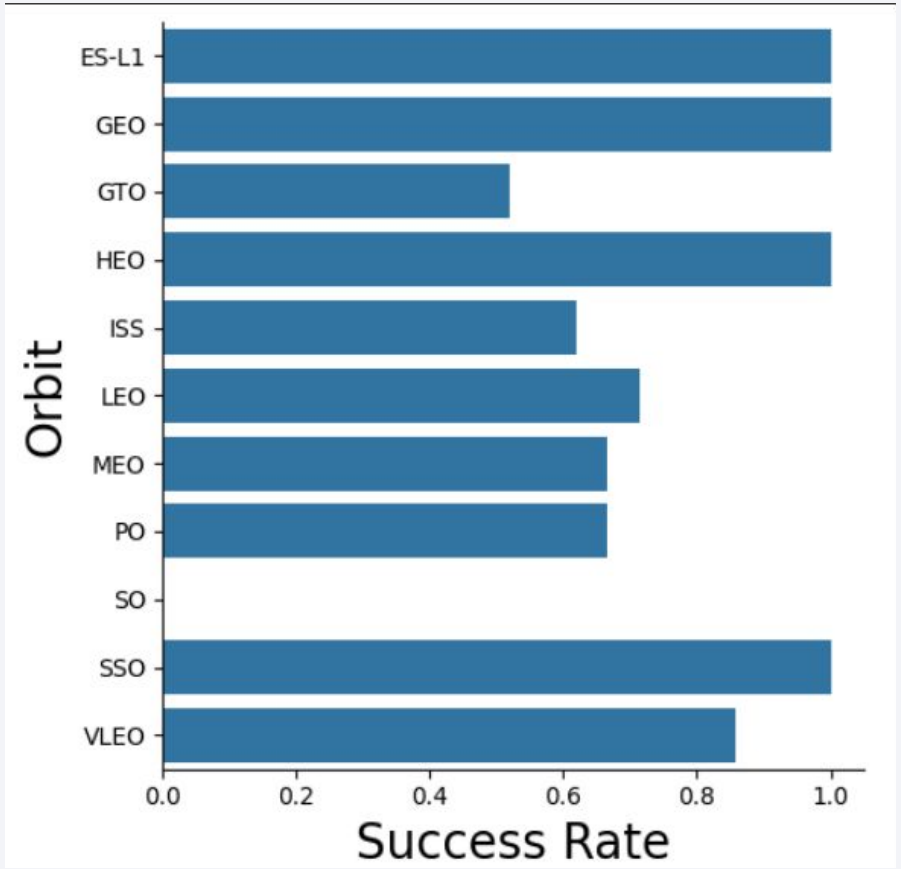
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, **True Ocean** means the mission outcome was successfully landed to a specific region of the ocean while **False Ocean** means the mission outcome was unsuccessfully landed to a specific region of the ocean. **True RTLS** means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.**True ASDS** means the mission outcome was successfully landed on a drone ship **False ASDS** means the mission outcome was unsuccessfully landed on a drone ship.

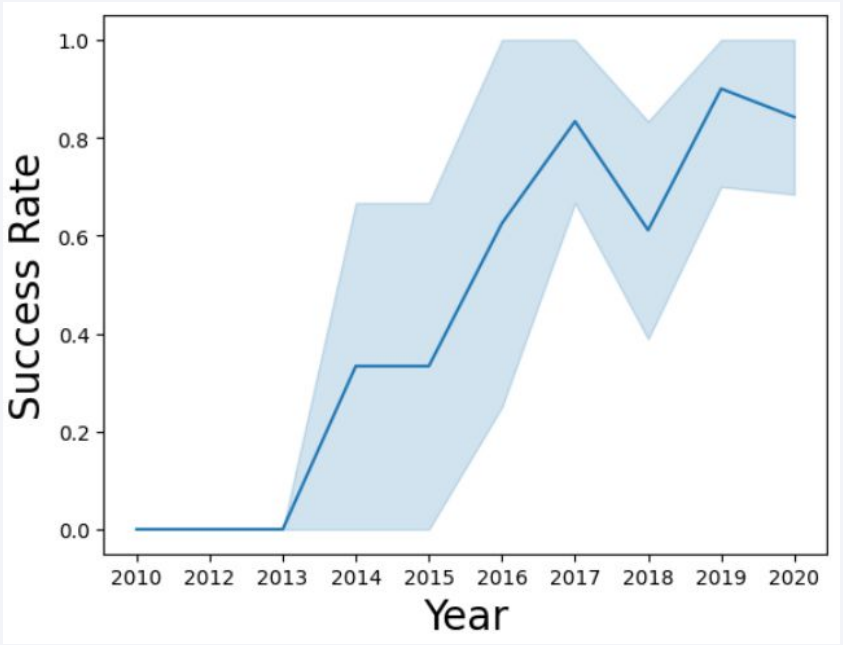
  In this lab we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

- Data Wrangling Notebook
- Flowchart Data Wrangling

# EDA with Data Visualization

- We explore the data by visualizing (matplotlib and seaborn) the relationship between columns.




- EDA using Visualization Library Notebook

# EDA with SQL

- We Loaded the SpaceX dataset into SQL database in jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - display the names of the unique launch sites in the space mission.
  - display 5 records where launch site begin with the string ‘CCA’.
  - display the total payload mass carried by boosters launched bu NASA (CRS).
  - display average payload mass carried by boosters 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.
- EDA using SQL Notebook

# Build an Interactive Map with Folium

- We marked all launch site and added map objects such as markers, circle, lines to mark the success or failure of launches for each site on the folium map.
- Coloured marked of launch outcomes for each launch site. green (success) and red (failed) to identify which launch site have relatively high success rate.
- Interactive map with Folium

# Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly Dash.
- Launch sites dropdown list.
- Slider of payload mass range
- Scatter chart of payload mass vs. success rate for the difference booster version.
- Plotted pie chart showing success Lauches

# Predictive Analysis (Classification)

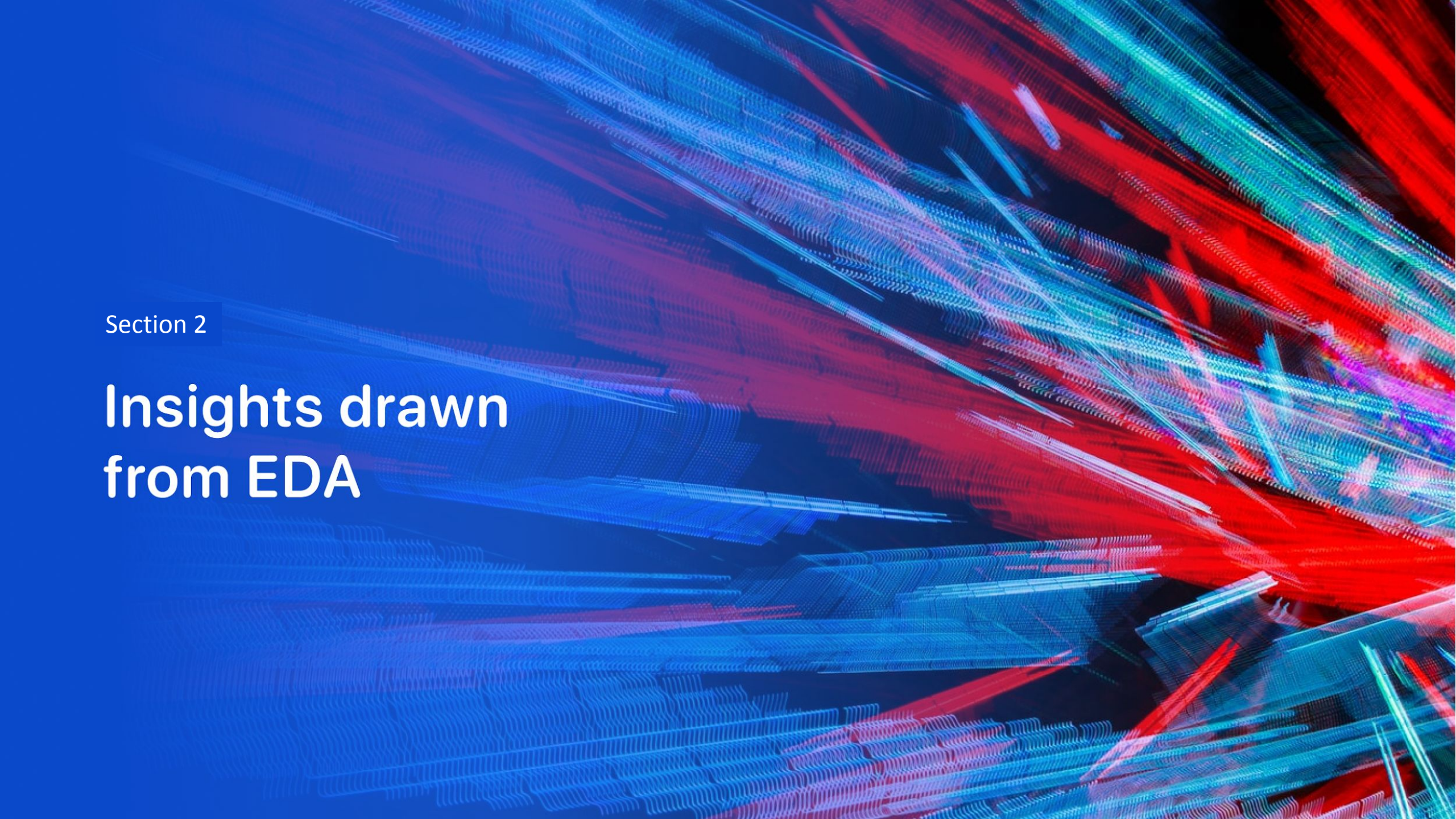
- Loaded the data using numpy and pandas, transformed the data, split data into training dan testing 80:20.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Use accuracy as the metric for model, and improved the model using feature engineering and algorithm tuning.
- Found the best performing  classification model.
- Prediction Analysis Model Notebook

# Results

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

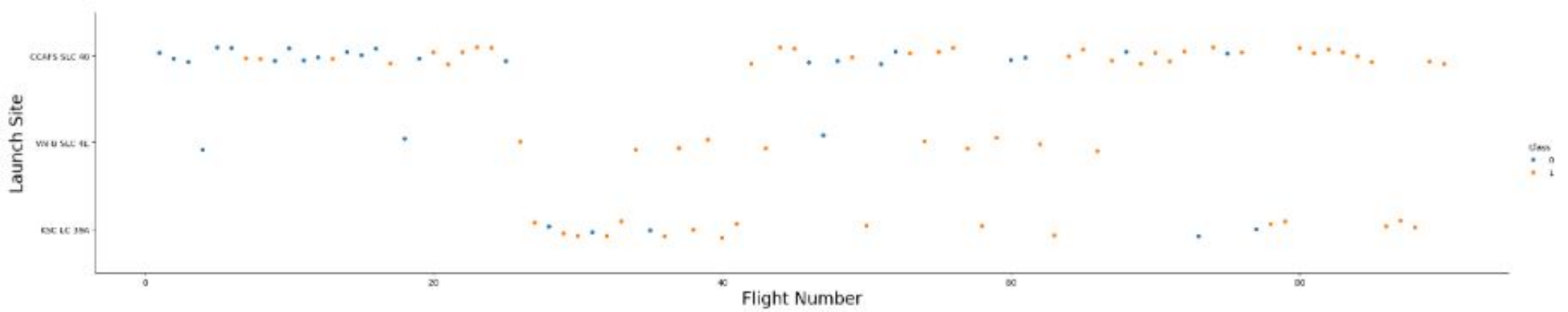
Section 2

# Insights drawn from EDA



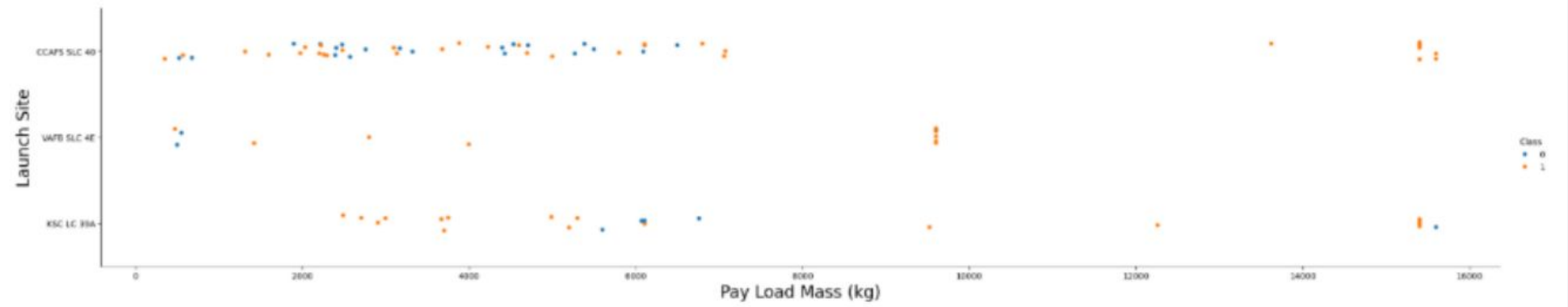
# Flight Number vs. Launch Site

- This visualization effectively demonstrates that launch success rates tend to increase with increasing **flight number** (experience). Furthermore, KSC LC-39A **launch site** has a better success record than the other two sites.



# Payload vs. Launch Site

- This analysis confirms that there is a clear relationship between **launch site** and **payload mass**. Each location appears to have a specialty or operational limitation: VAFB for light payloads, CCAFS for extended ranges, and KSC for the heaviest payloads.

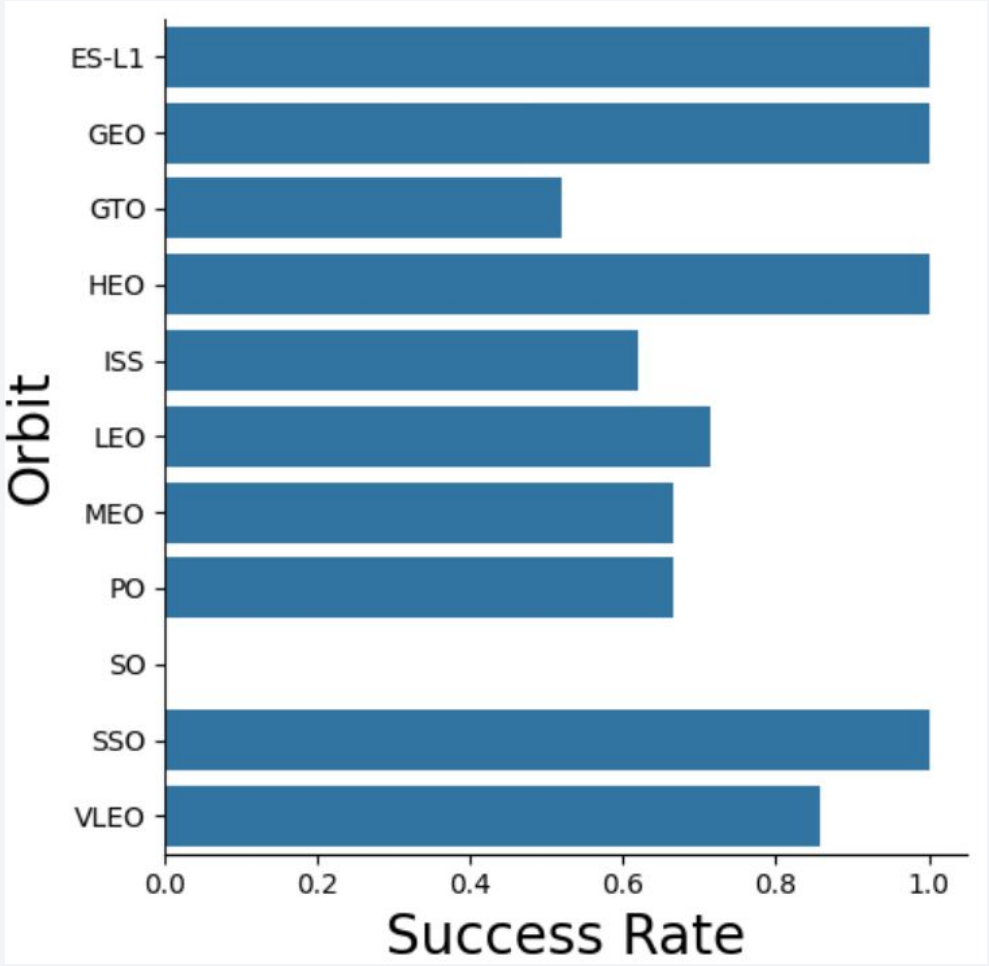


# Success Rate vs. Orbit Type

Space mission success rates vary significantly depending on the destination orbit.

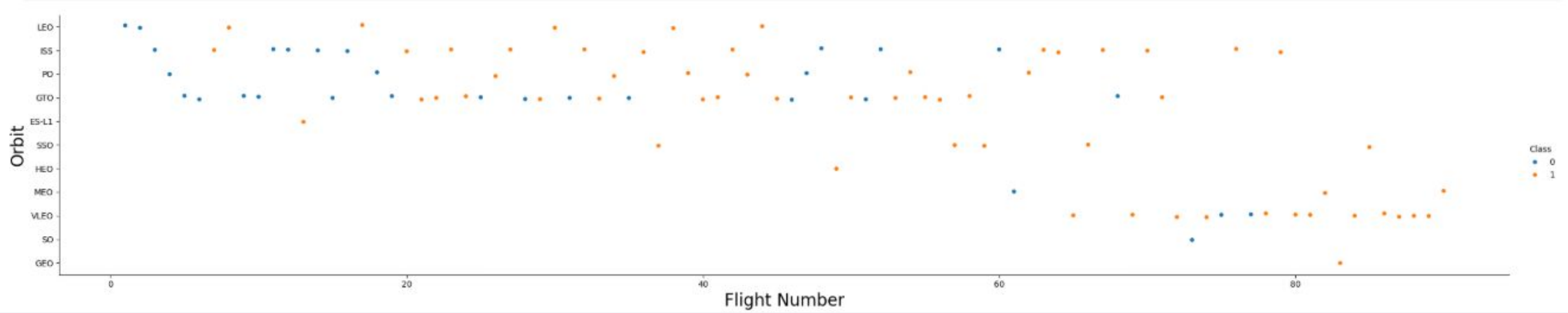
- Highly Successful (Near 100%): Missions to ES-L1, GEO, HEO, and SSO orbits are highly reliable.
- Riskiest (50% or Less): GTO (Geostationary Transfer Orbit) orbits are the most challenging, with a success rate of only 50%, while SO (Sub-orbital) missions have a success rate of 0%.

Bottom Line: The biggest challenge is often not the final destination, but rather the "transfer" phase of reaching that orbit, as evidenced by the low success rate of GTO.



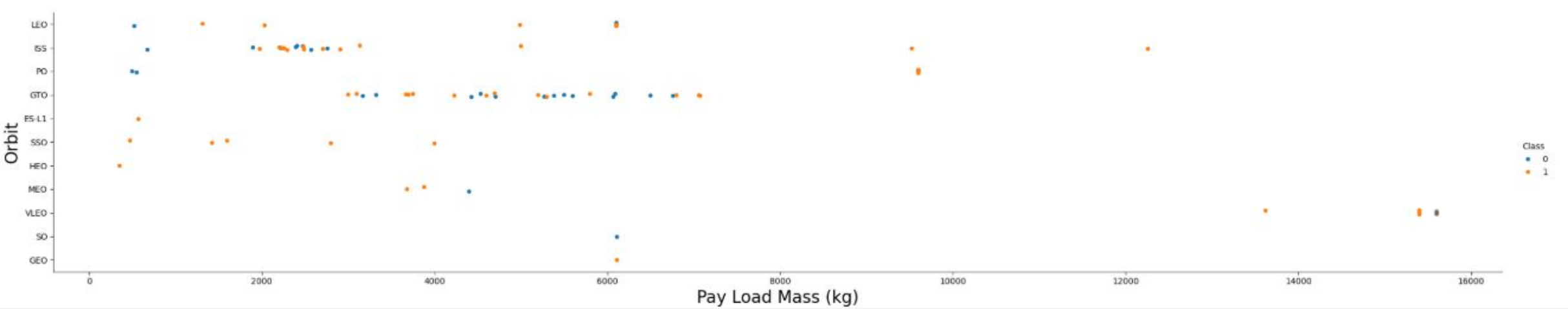
# Flight Number vs. Orbit Type

- Based on observations in LEO orbit, success appears to be related to the number of flights. In contrast, in GTO orbit, there appears to be no relationship between flight number and success. Launch success rates depend heavily on orbit type and experience (as represented by FlightNumber).



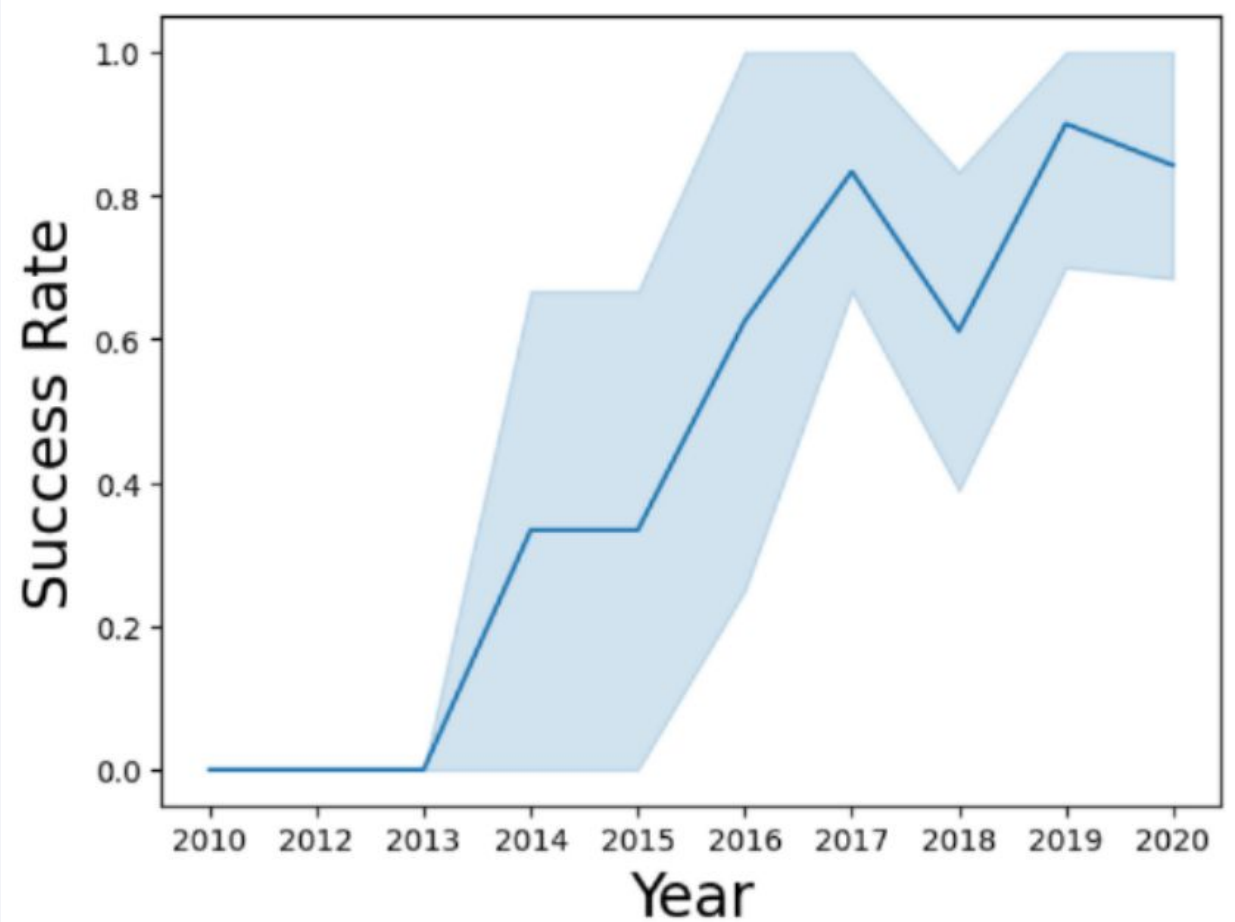
# Payload vs. Orbit Type

- 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

- Success rate trend from 2010 to 2020. Stagnant in 2010-2013, it gradually increased starting from 2014-2020, although there was a decrease in the success rate in 2019 and increased again in 2020.



# All Launch Site Names

- Use DISTINCT to show only unique launch sites from the SpaceX data.

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

```
 * 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'

- Displaying 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 |

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Displaying 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 |

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

| AVG("PAYLOAD_MASS__KG_") |
|---|
| 2928.4 |

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad as 22 December 2015

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

| MIN(Date) |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- Listing the names of boosters which have successfully landed on drone ship and had 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_" BETWEEN 4000 AND 6000;
```

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

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

# Total Number of Successful and Failure Mission Outcomes

- Used wildcard like ‘%’ to filter for WHERE MissionOutcome was a success or failure.

```
%sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

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

# Boosters Carried Maximum Payload

- Listing the names of the booster which have carried the maximum payload mass

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

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

| | boosterversion | payloadmasskg |
|---|---|---|
| 0 | F9 B5 B1048.4 | 15600 |
| 1 | F9 B5 B1048.5 | 15600 |
| 2 | F9 B5 B1049.4 | 15600 |
| 3 | F9 B5 B1049.5 | 15600 |
| 4 | F9 B5 B1049.7 | 15600 |
| 5 | F9 B5 B1051.3 | 15600 |
| 6 | F9 B5 B1051.4 | 15600 |
| 7 | F9 B5 B1051.6 | 15600 |
| 8 | F9 B5 B1056.4 | 15600 |
| 9 | F9 B5 B1058.3 | 15600 |
| 10 | F9 B5 B1060.2 | 15600 |
| 11 | F9 B5 B1060.3 | 15600 |

# 2015 Launch Records

- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in 2015.

```
%sql SELECT substr(Date, 6, 2) AS month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr(Date, 0, 5) = '2015';
 * sqlite:///my_data1.db
Done.
```

| 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

- Ranking 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 "Landing_Outcome", COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY OutcomeCount DESC;
```

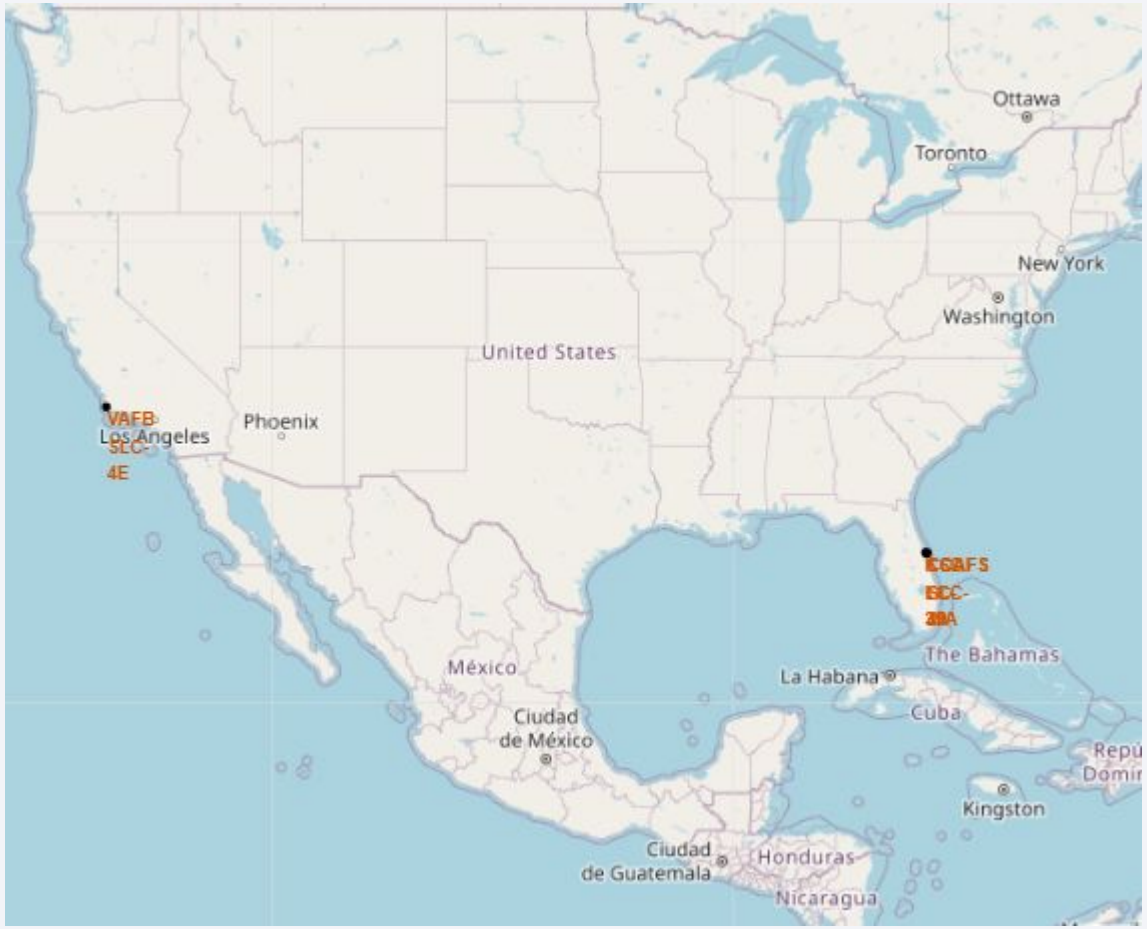
```
 * sqlite:///my_data1.db
Done.
```

| Landing_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

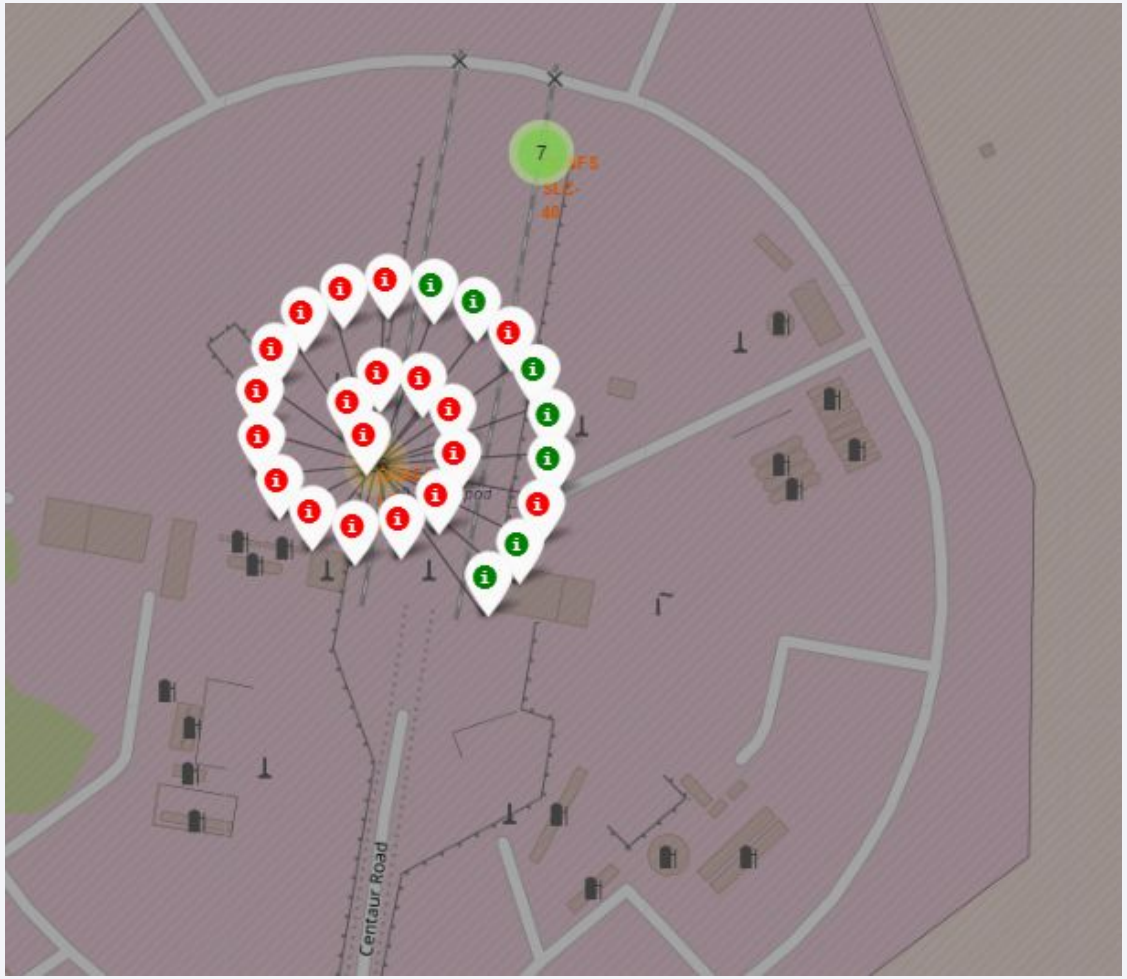
# All launch sites global map markers

- Most of launch sites are in proximity to the equator line. All launch sites are in very close proximity to the coast, while launching rockets towards the ocean it minimize the risk having any debris dropping or exploding near people.



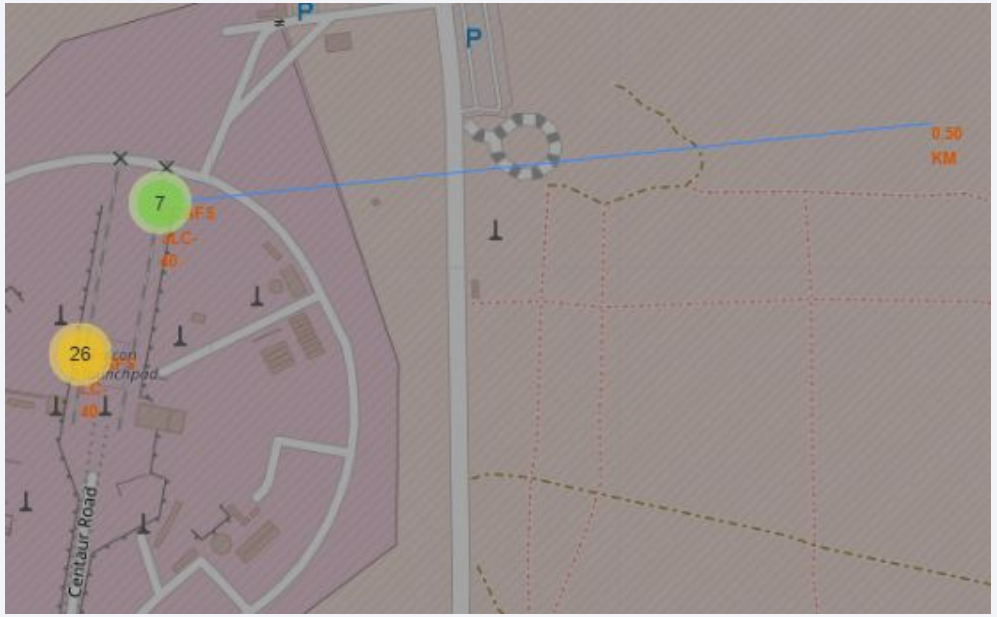
# Markers showing launch sites with color labels

- Colour-labeled markers are able to easily identify which launch sites have relatively high success rate.
- Green markers: successful launch
- Red markers: Failed launch.

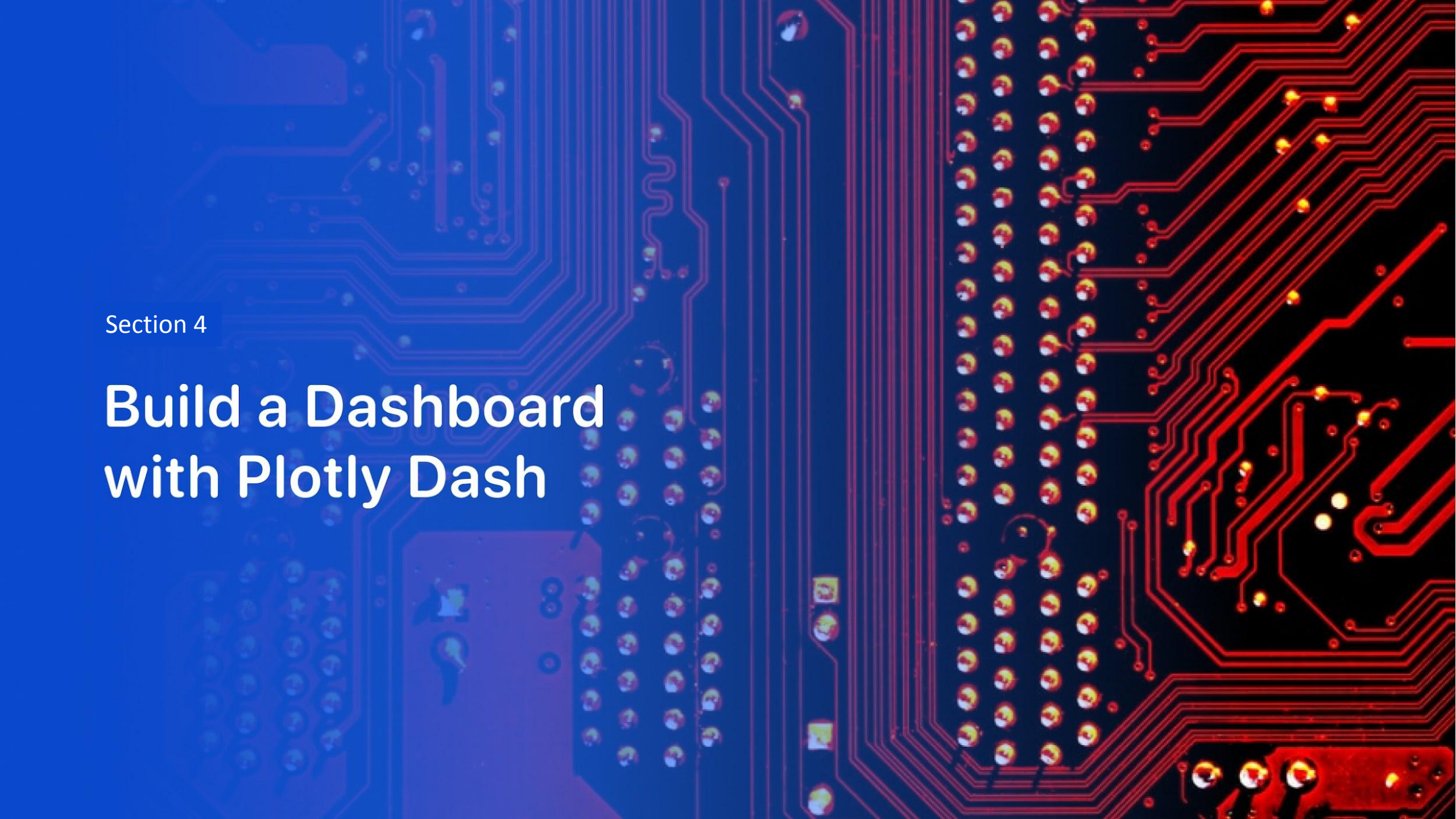


# Lanch site distance to landmarks

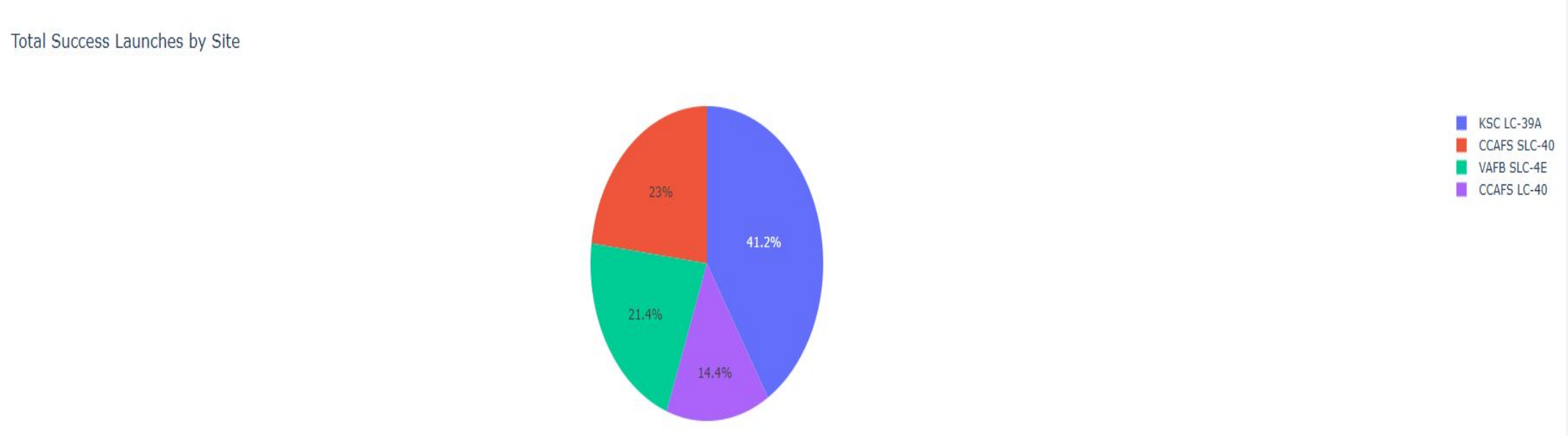
- Findings:
  - Proximity to railways: ,1.28 KM
  - Proximity to highways: 0.01 KM
  - Proximity to coastline: 0.50 KM
  - Distance from cities: 16.29 KM





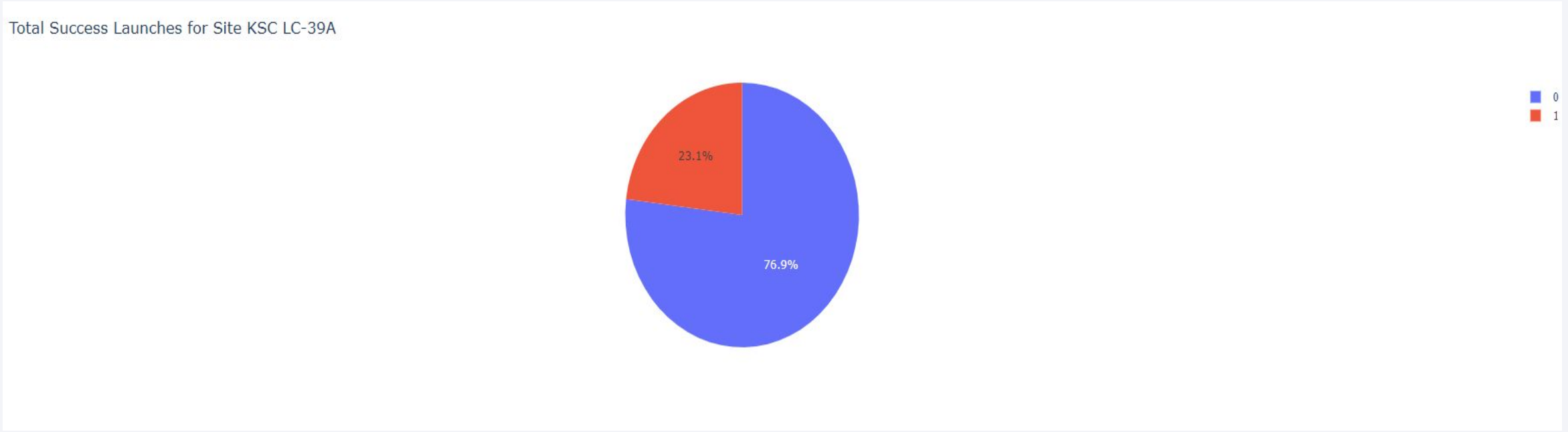

Section 4
Build a Dashboard with Plotly Dash

# Launch success count for all sites




- The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.

# Lauch site with highest launch succes rate




- KSC LC-39A has the highest launch success rate 78.9%.

# <Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- Best performing model is Logistic Regression with accuracy 83%.

```
results = {
    'Logistic Regression': logreg_test_accuracy,
    'SVM': svm_test_accuracy,
    'Decision Tree': tree_test_accuracy,
    'KNN': knn_test_accuracy
}

best_method = max(results, key=results.get)

print("Accuracy on test data for each method:")
for method, accuracy in results.items():
    print(f"{method}: {accuracy:.4f}")

print(f"\nBest performing method: {best_method} with accuracy {results[best_method]:.4f}")
```

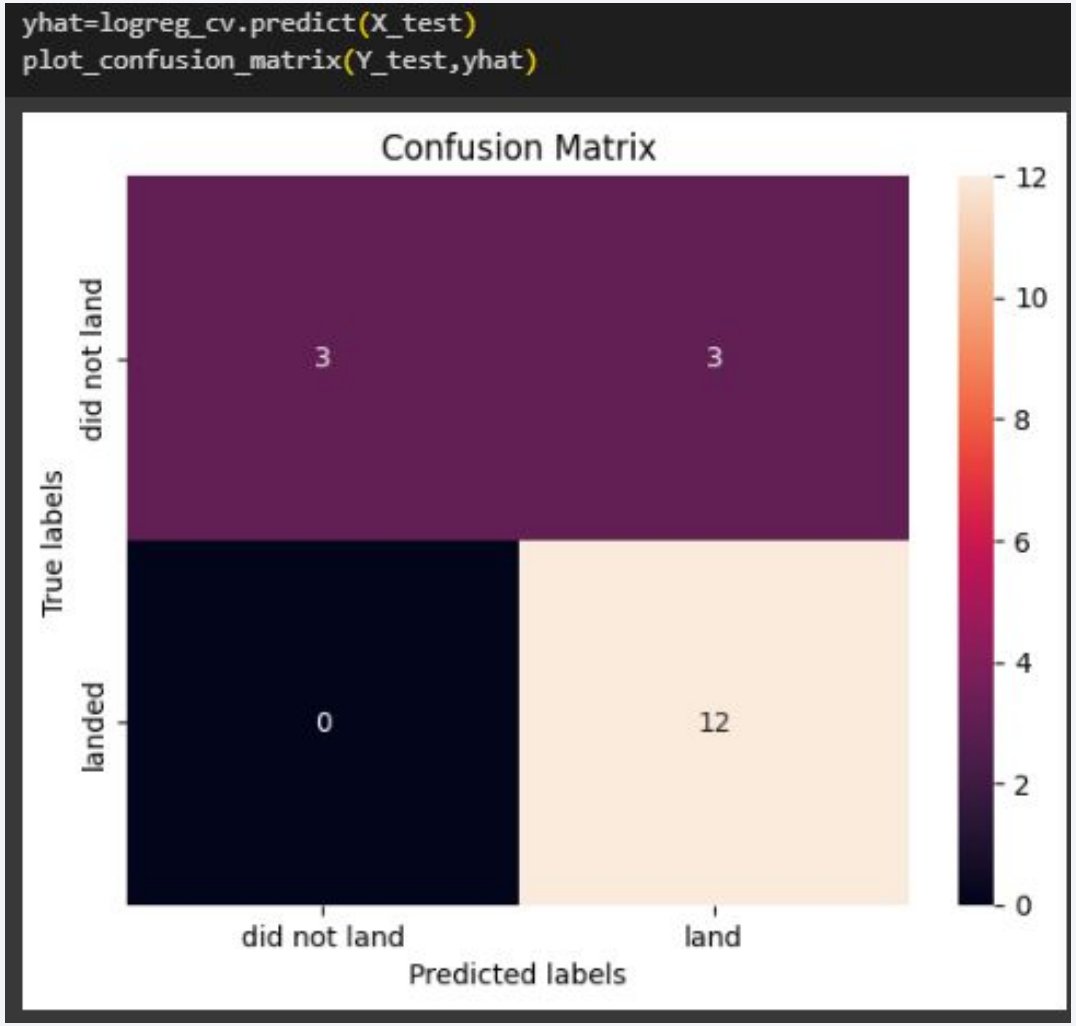
```
Accuracy on test data for each method:
Logistic Regression: 0.8333
SVM: 0.6667
Decision Tree: 0.7778
KNN: 0.6111

Best performing method: Logistic Regression with accuracy 0.8333
```

# Confusion Matrix

- The major problem in Logistic Regression is false positives.

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



# Conclusions

- Logistic Regression model is the best algorithm for this dataset.
- Launch success rate started to increase in 2013-2020. But peak the success rate of launches increases over the years.
- KSC LC-39A had the most successful launches of any sites.
- Orbit ES-L1, GEO, HEO, dan SSO have 100% success rate.

# Thank you!