IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hoang Trung
05/2024

OUTLINE

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

EXECUTIVE SUMMARY

- Summary of methodologies
- Summary of all results

PROJECT BACKGROUND AND CONTEXT:

This project focuses on SpaceX, a prominent aerospace company founded by Elon Musk. We aim to analyze data related to SpaceX launches, exploring their success rates, payload trends, rocket performance, landing outcomes, launch site choices, and temporal patterns.

PROBLEMS

- Launch Success: What factors contribute to the success of SpaceX launches?
- Payload Analysis: How have payload mass and types evolved over time, and how do they correlate with launch outcomes?
- Rocket Performance: What insights can we gain about the reliability and reusability of SpaceX rockets?
- Landing Outcomes: What patterns can be observed in the outcomes of first-stage landings?
- Launch Site Assessment: How do launch site selections impact mission success?
- Temporal Trends: Are there any significant trends or seasonal patterns in SpaceX's launch history?

Section 1

Methodology

METHODOLOGY

Executive Summary

- Data collection methodology:

Describe how data was collected

- Perform data wrangling:

Describe how data was processed

- 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

Initial Setup:
- Primary Data Source: SpaceX Official Website
- Configure web scraping tools and environment

Website Access:
- Utilize web scraping libraries to access data from SpaceX's website

Data Extraction:
- Extract Data from web pages

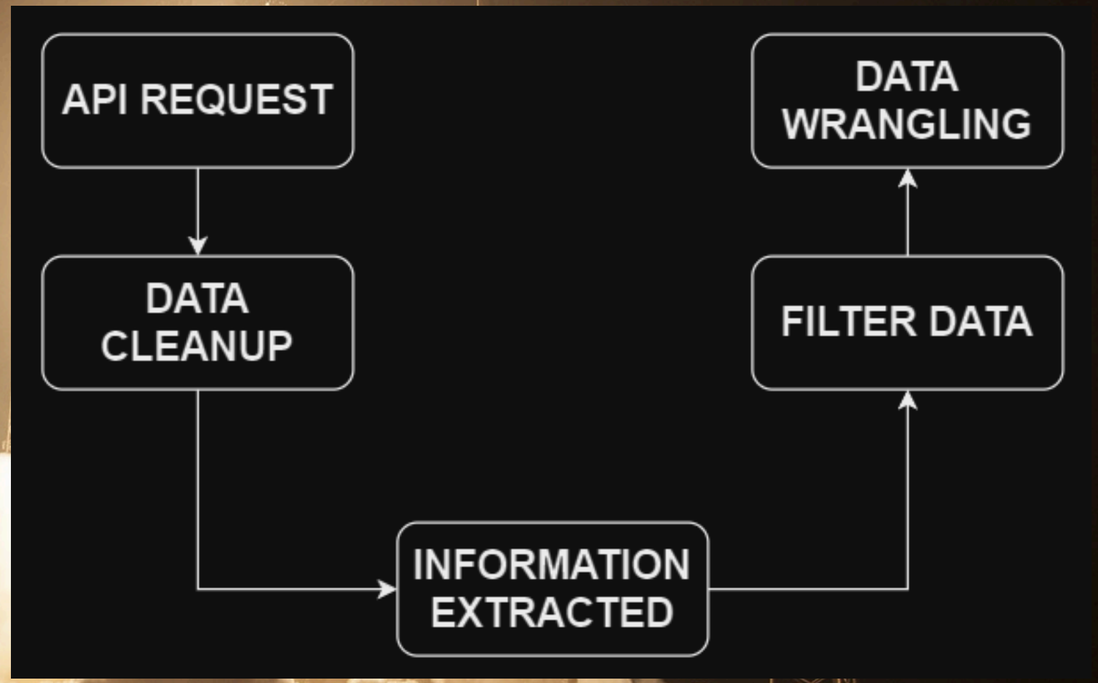
Data Cleaning:
- Clean the extracted data to remove inconsistencies and errors
- Identify and address any missing values.

Data Storage:
- Export Data to CSV Files

Data Analysis:
- Analyze Collected Data

Data Collection – SpaceX API

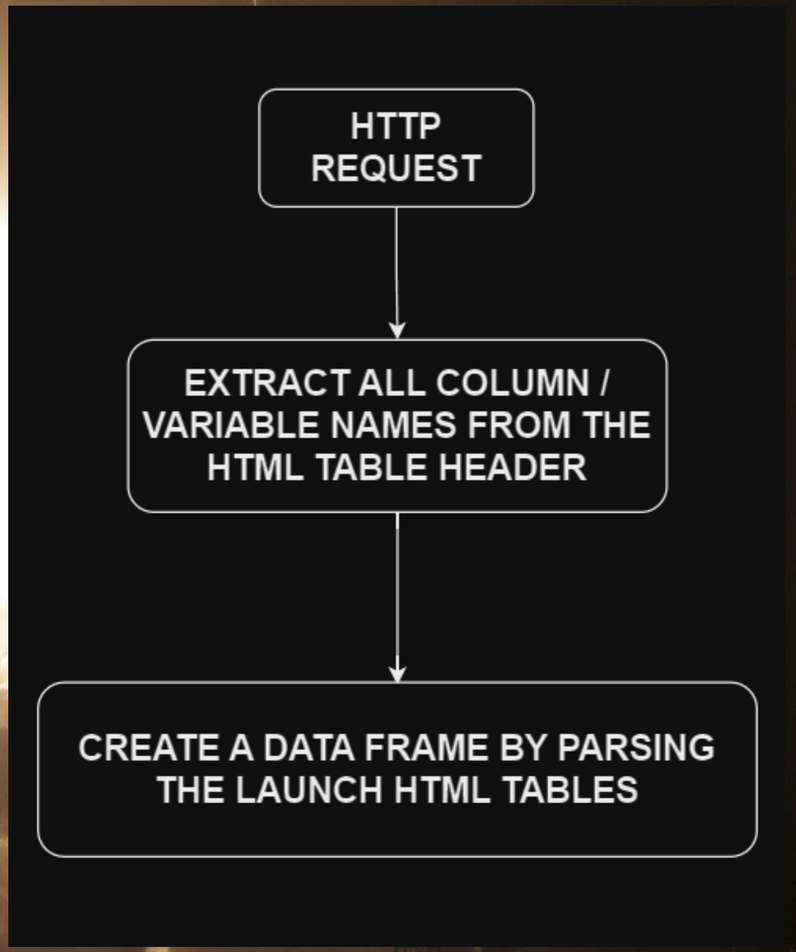
- **API Request**: Use SpaceX API to get Data From SpaceX website
- **Data CleanUp**: Formatted and refined the extracted data
- **Information Extracted**: BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- **Filter Data**: Retained only the Falcon 9 launches.
- **Data Wrangling**: Replace missing Payload Mass values with the mean of PayloadMass column.



GIT HUT: LINK

DATA COLLECTION - SCRAPING

- Request the Falcon9 Launch Wiki page from its URL
  - Using requests, BeautifulSoup to get HTML Code.
- Extract Data From HTML Code
  - Find All table in this page (use soup.find_all('table'))
  - Get Third Table and check
  - Use For Loop to extract all column names from this table.
- Create a Data Frame by parsing the launch HTML tables:
  - Create Dictionary launch_dict
  - Parse Data from HTML table to launch_dict
  - Create Data Frame from launch_dict

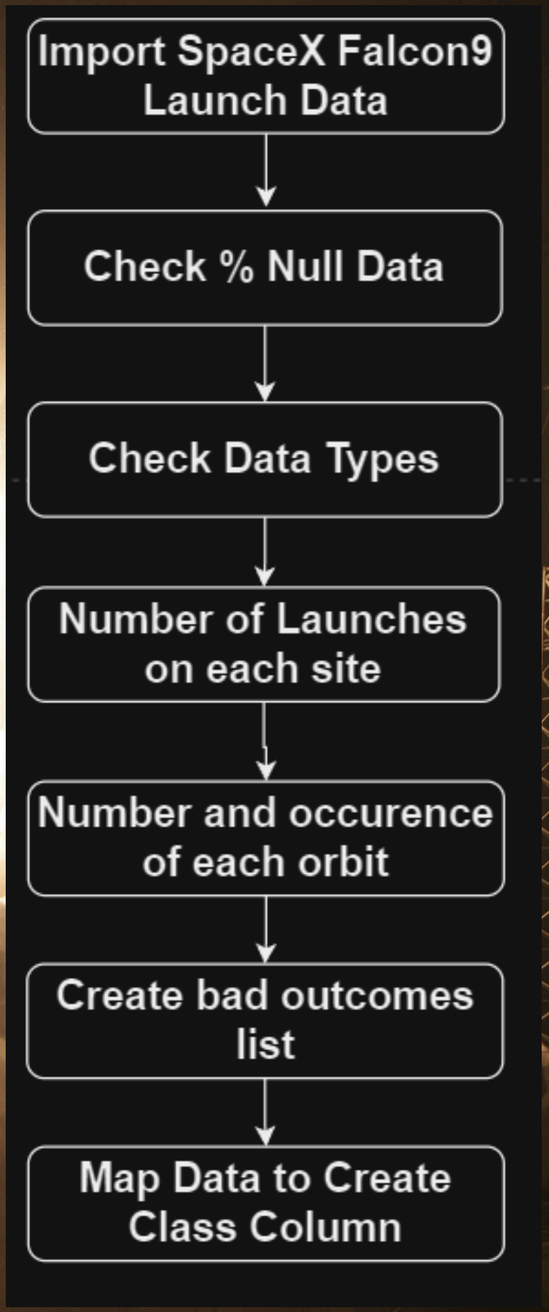


GIT HUT: LINK

DATA WRANGLING

- Import Data from a CSV file contains SpaceX Falcon9 launch data.
- Check % Null Data in your Data Frame
- Check Data Types
- Determine the number of launches on each site
- Determine the number and occurence of each orbit
- Create bad outcomes list from Launch Data
- Create new column Class with 0 (Outcome is in bad outcomes list) and 1 for others.

GIT HUT: LINK



EDA WITH DATA VISUALIZATION

- **FlightNumber vs PayloadMass Scatter Plot**:
  - **Purpose**: Investigate the connection between flight number and payload mass.
- **FlightNumber vs Launch Site Scatter Plot**:
  - **Purpose**: Investigate the relationship between flight number and launch site
- PayloadMass vs Launch Site Scatter Plot:
  - **Purpose**: Investigate the relationship between payload mass and launch site
- Success Rate vs Orbit Bar Chart:
  - **Purpose**: Visualize the success rate of different orbit types
- FlightNumber vs Orbit Scatter Plot:
  - **Purpose**: Investigate the relationship between flight number and orbit type
- PayloadMass vs Orbit Scatter Plot:
  - **Purpose**: Investigate the relationship between payload mass and orbit type
- Launch Success Yearly Trend Line Chart:
  - **Purpose**: Analyze the trend in average launch success rate over the years

GIT HUT: LINK

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.

GIT HUT: LINK

BUILD AN INTERACTIVE MAP WITH FOLIUM

On the Folium map, I added the following map objects:

- **Markers**: These indicate specific locations or points of interest on the map. Markers display information when clicked, making them useful for highlighting important spots.
- **Polylines**: I used polylines to connect multiple points on the map, creating a visual path or route. This helps users understand the connections between locations.
- **Marker Clusters**: To prevent map clutter, I grouped nearby markers into clusters. This enhances map readability, especially when many markers are close together.

I added these objects to enhance visualization, represent connectivity, and improve the user experience.

GIT HUT: LINK

BUILD A DASHBOARD WITH PLOTLY DASH

- **Dropdown Selection for Statistics**:
  - Users can select either “Yearly Statistics” or “Recession Period Statistics”.
- **Dropdown Selection for Year**:
  - Users can pick a specific year for “Yearly Statistics”
- **Disabled Year Dropdown**:
  - The year dropdown is deactivated for “Recession Period Statistics”
- **Graph Output Container**:
  - A dynamic section for showcasing interactive graphs.
- **Yearly Statistics Graph (Example)**:
  - A bar chart depicting monthly automobile sales for the chosen year.
- **Recession Period Statistics Graph (Example)**:
  - A line chart showing automobile sales during recession periods.
  - Purpose:
    - Examine historical automobile sales data.
    - Evaluate annual sales patterns.
    - Assess the impact of recessions on sales.

GIT HUT: LINK

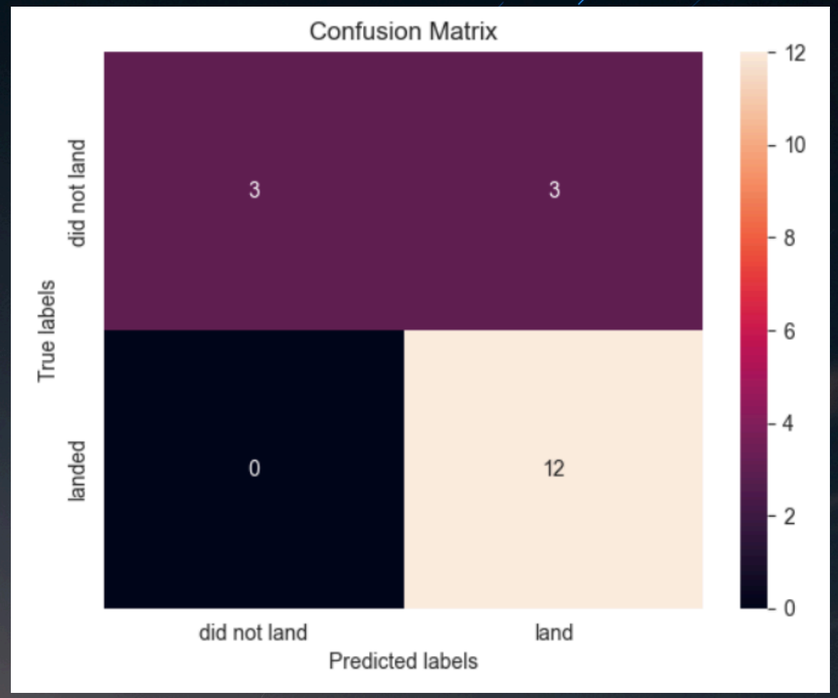
PREDICTIVE ANALYSIS (CLASSIFICATION)

- Load Data to Data Frame
- Create X, y variable
- Use StandardScaler to scale X
- Split Train, Test Data (Train: 80%, Test: 20%)
- Use Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors Algorith to predict rocket landings fail or successful.
- Use GridSearchCV to tuning model
- Evaluation the model and check accuracy score
- Plot Confusion Matrix
- Create Data Frame Report to summarize the score of 4 models.

GIT HUT: LINK

LOGISTIC REGRESSION CONFUSION MATRIX

```
In 11
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
Executed at 2024.05.15 17:32:02 in 58ms

In 12
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge

lr=LogisticRegression()
Executed at 2024.05.15 17:32:02 in 54ms

In 13
logreg_cv = GridSearchCV(lr,parameters,cv=10)

logreg_cv.fit(X_train, y_train)
Executed at 2024.05.15 17:32:02 in 114ms

Out 13
▸ GridSearchCV
▸ estimator: LogisticRegression
  ▸ LogisticRegression
```

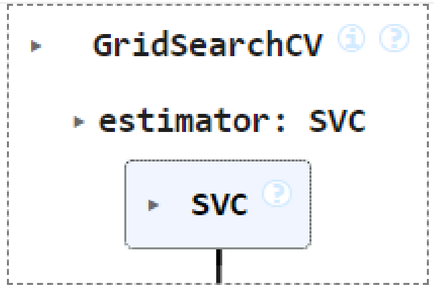


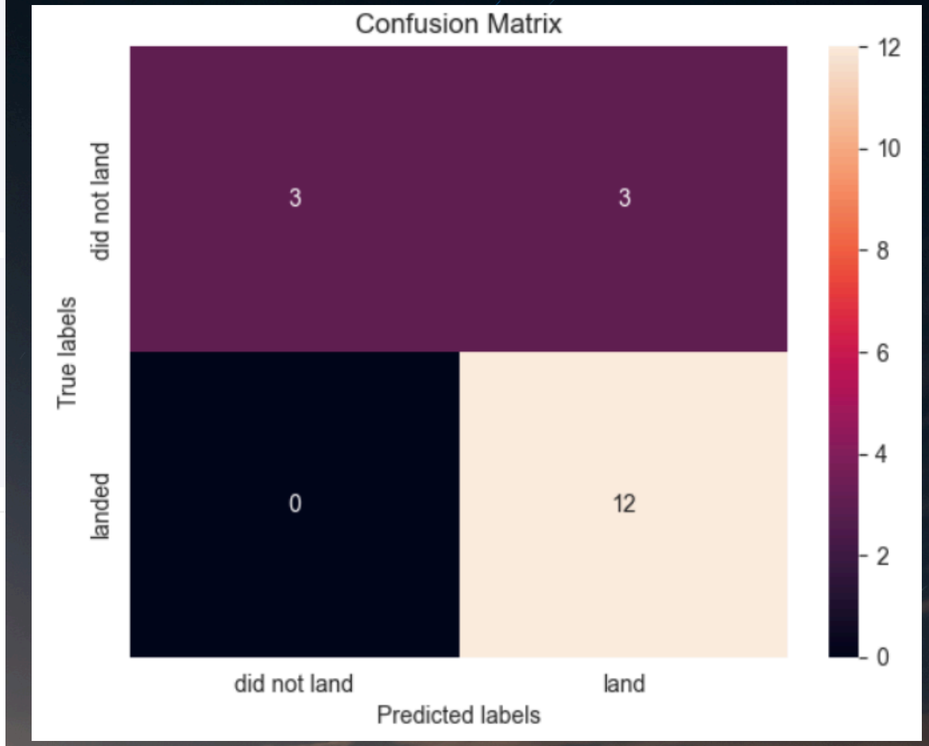
SVM CONFUSION MATRIX

In 17
```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```
Executed at 2024.05.15 17:32:02 in 5ms

In 18
```
svm_cv = GridSearchCV(svm,parameters,cv=10)

svm_cv.fit(X_train, y_train)
```
Executed at 2024.05.15 17:32:03 in 1s 323ms

Out 18

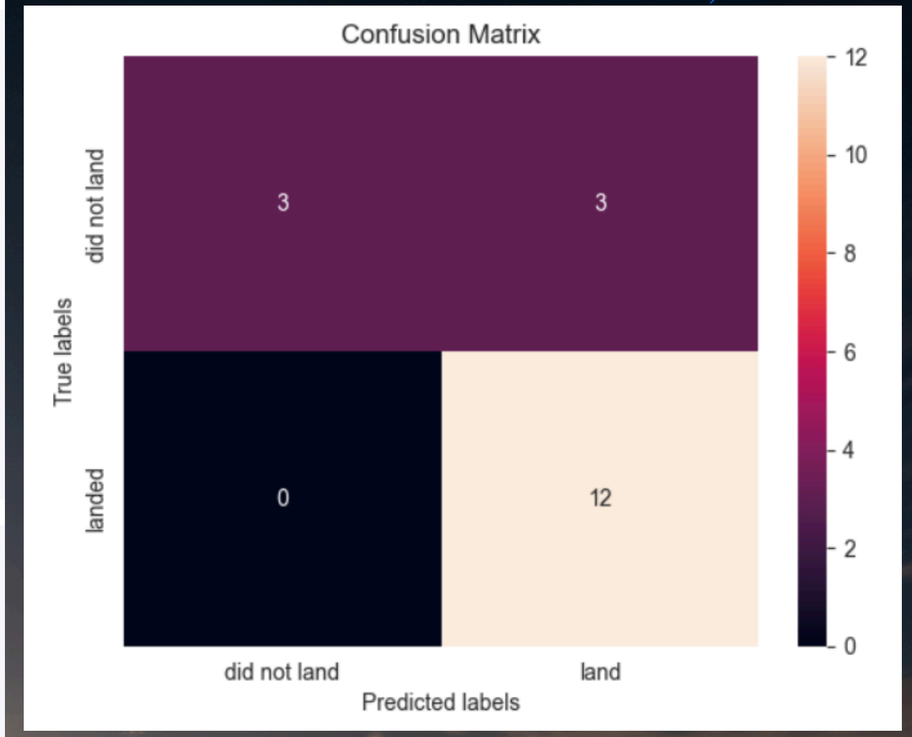




DECISION TREE CONFUSION MATRIX

In 22
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```
Executed at 2024.05.15 17:32:03 in 10ms

In 23
```
tree_cv = GridSearchCV(estimator=tree, param_grid=parameters, cv=10)

tree_cv.fit(X_train, y_train)
```
Executed at 2024.05.15 17:32:07 in 3s 859ms



K-NEAREST NEIGHBORS CONFUSION MATRIX

In 27
```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```
Executed at 2024.05.15 17:32:07 in 6ms

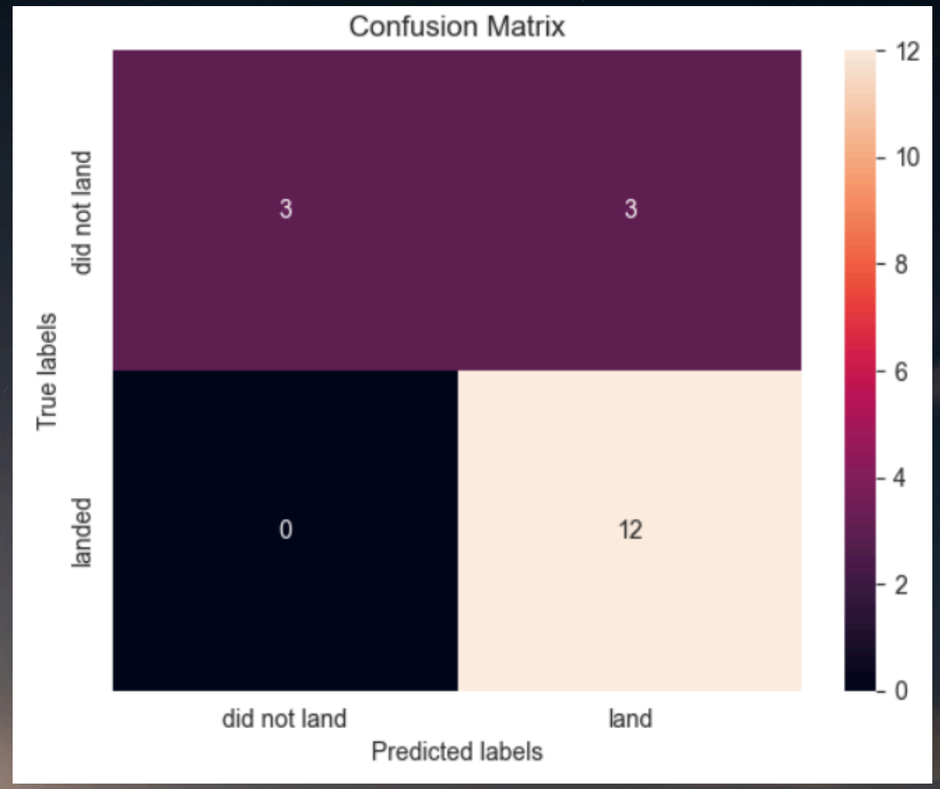
In 28
```
knn_cv = GridSearchCV(KNN,parameters,cv=10)

knn_cv.fit(X_train, y_train)
```
Executed at 2024.05.15 17:32:09 in 1s 319ms

Out 28

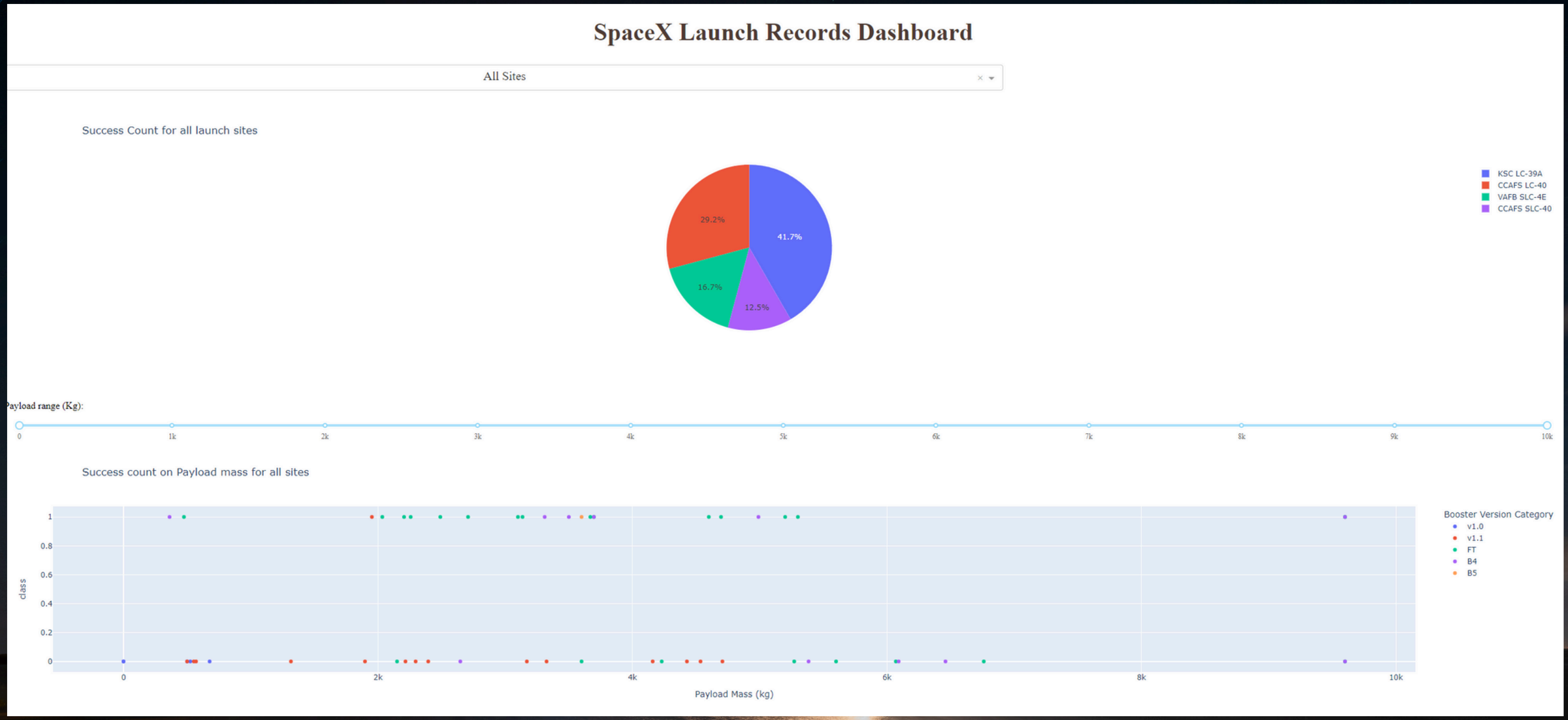
GridSearchCV

estimator: KNeighborsClassifier

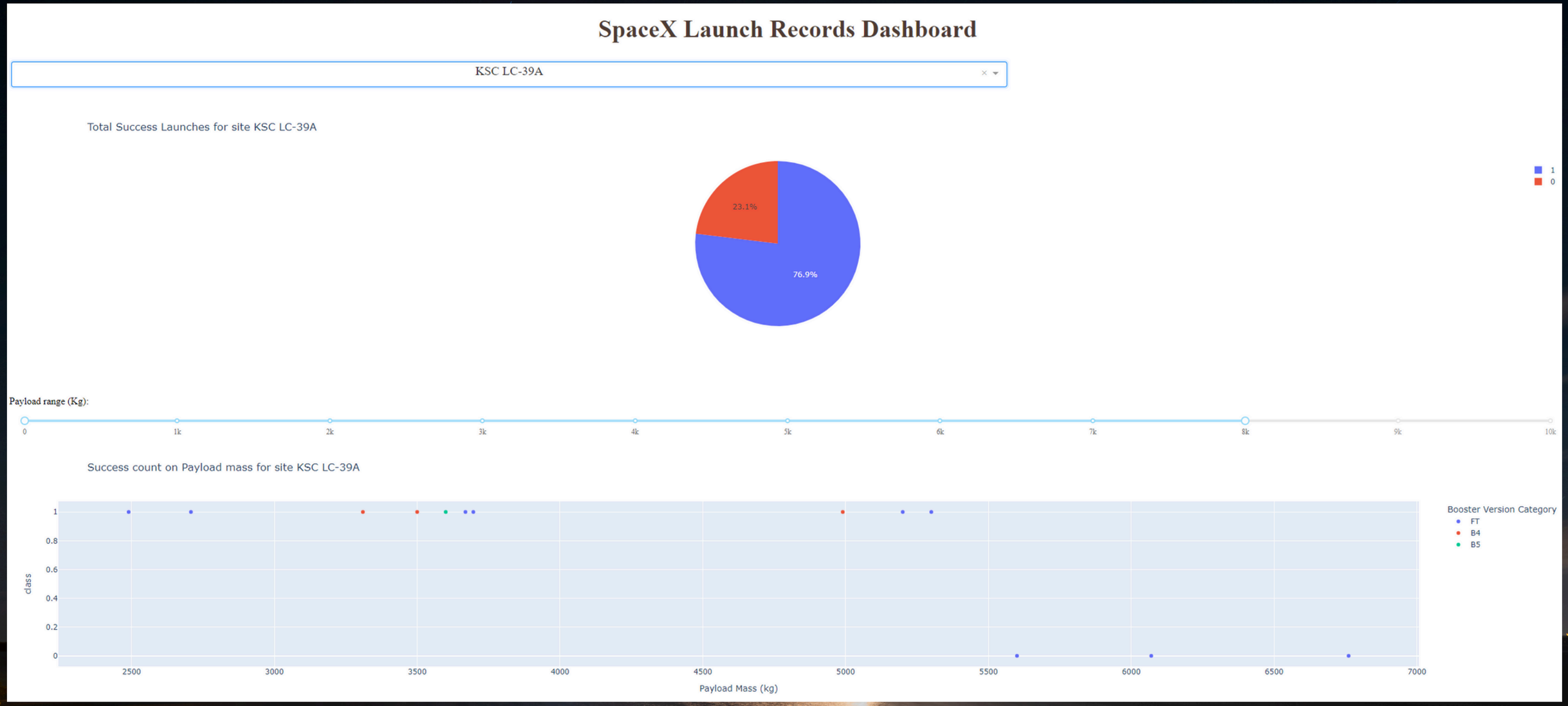
KNeighborsClassifier



INTERACTIVE DASHBOARD

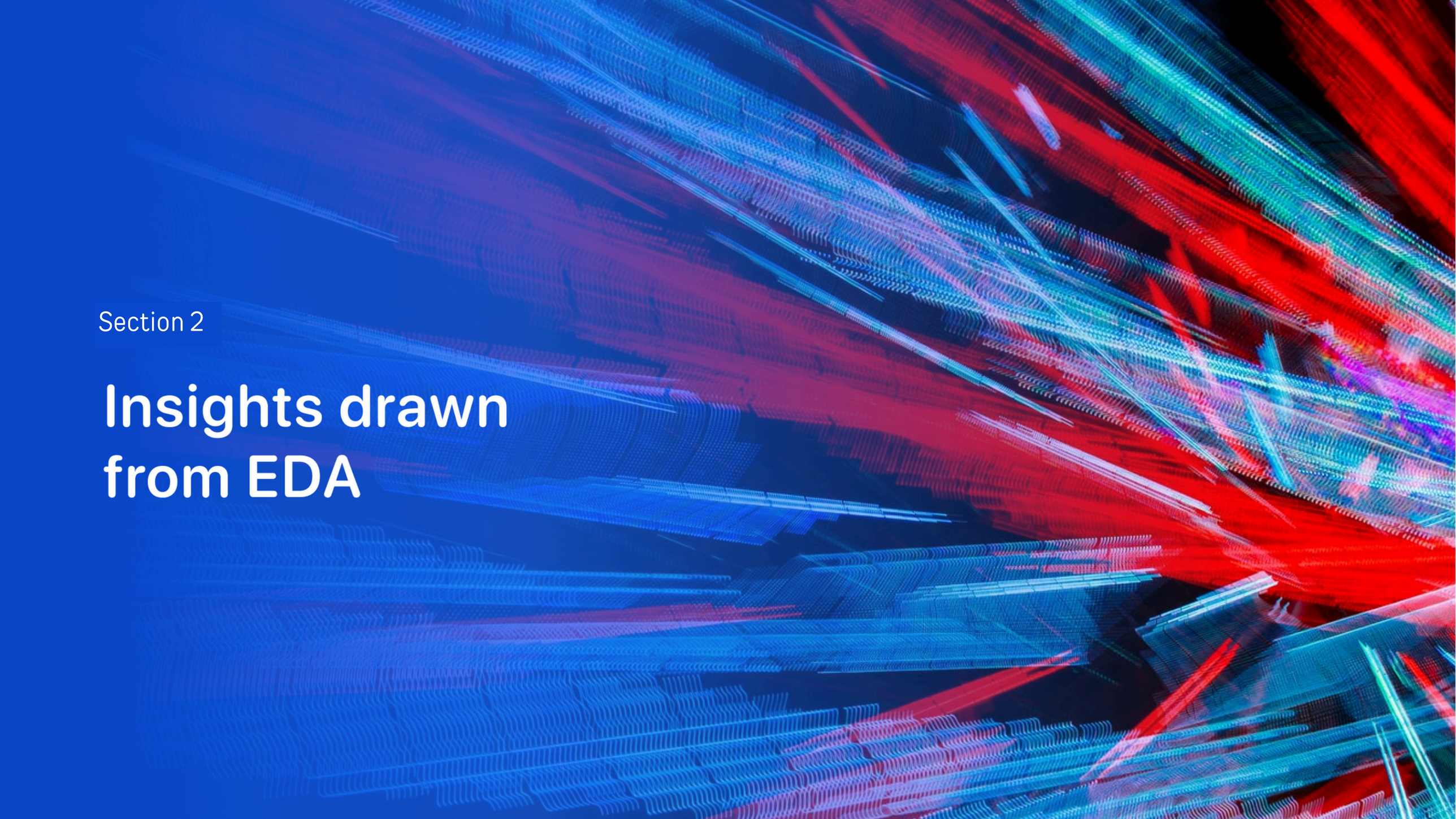


INTERACTIVE DASHBOARD

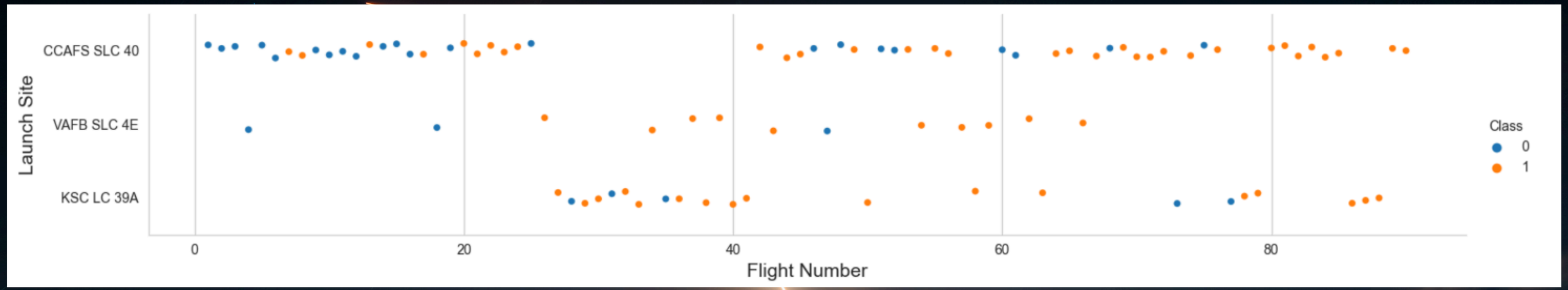


Section 2

Insights drawn from EDA



FLIGHT NUMBER VS. LAUNCH SITE

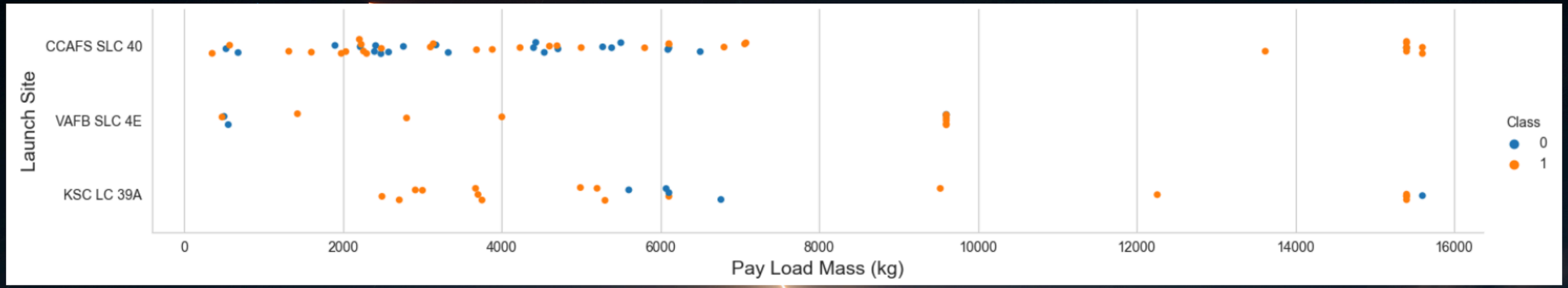


Before Launch Site No 20, they fail so much.

But after Launch Site No 80, the successful rate is **100%**

They only have 3 launch site and more of launching comes from CCAFS SLC 40

PAYLOAD VS. LAUNCH SITE



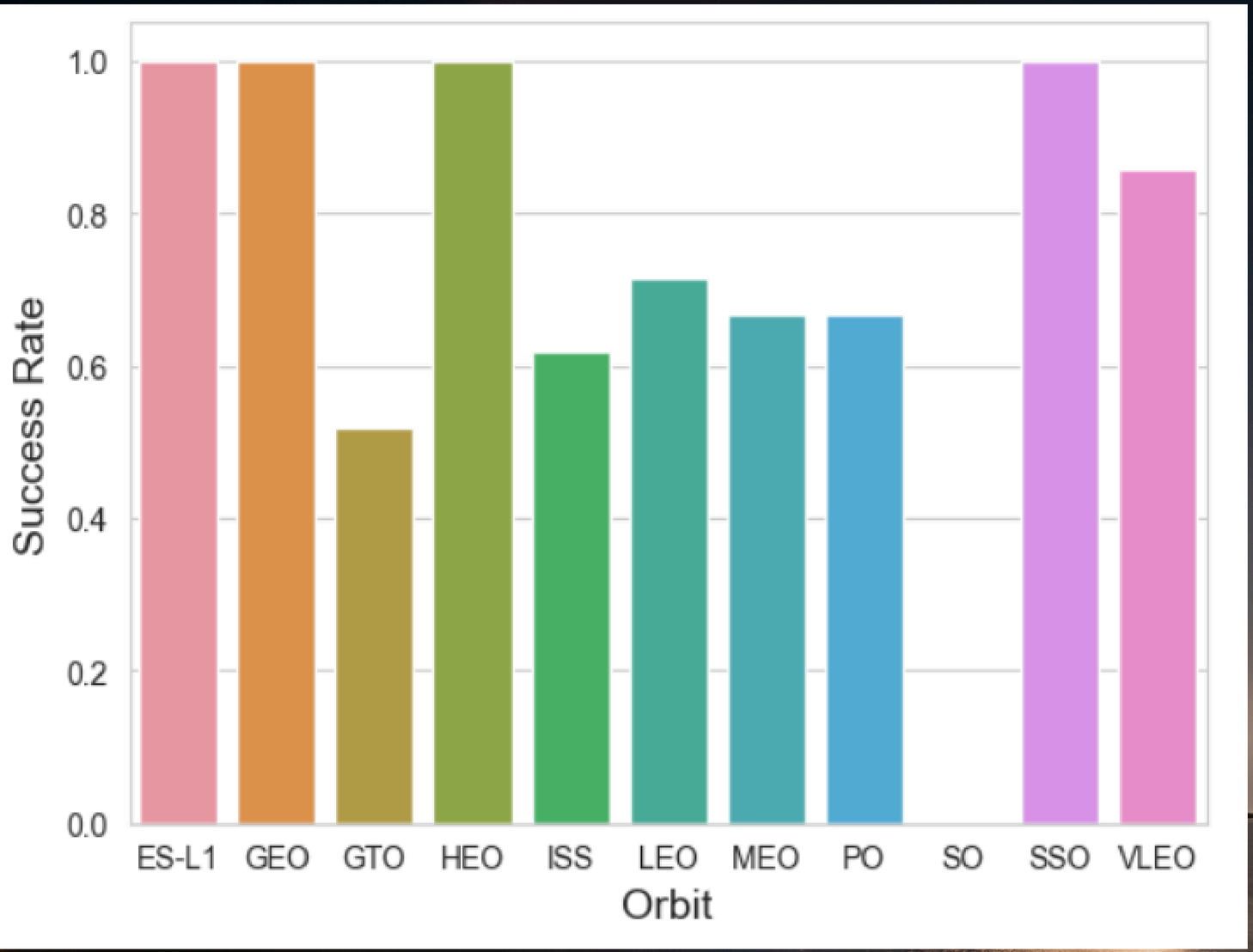
There are no rockets launched for heavypayload mass(greater than 10000).

Pay Load Mass greater than 8000 has very good successful rate.

But Pay Load Mass below 8000 has not good successful rate.

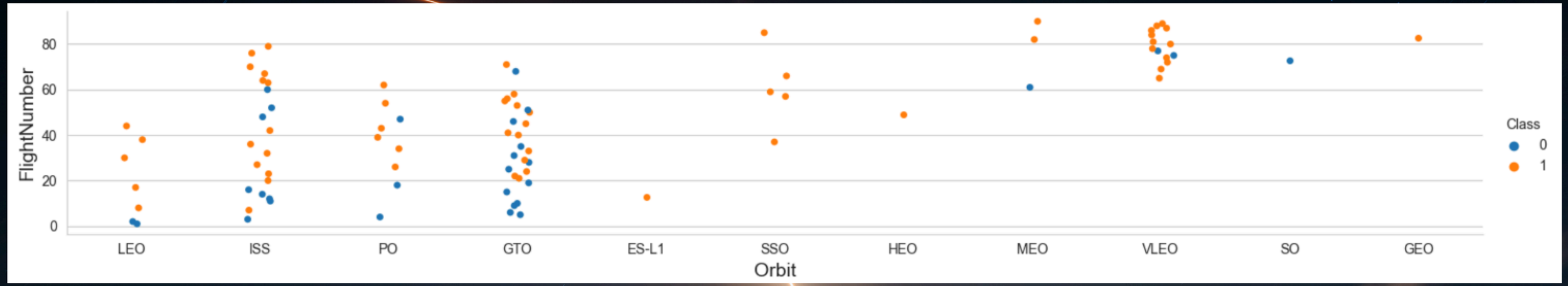
Pay Load Mass greater than 12000 only launch in CCAFS SLC 40 and KSC LC 39A.

SUCCESS RATE VS. ORBIT TYPE



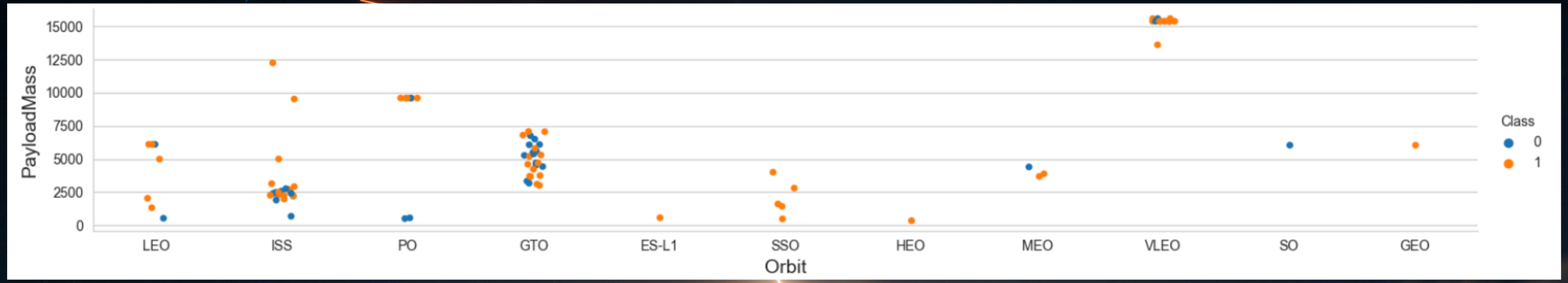
ES-L1, GEO, HEO, SSO Orbit have **100% successful rate**. SO is the worst orbit with **0% successful rate**.

FLIGHT NUMBER VS. ORBIT TYPE



The LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

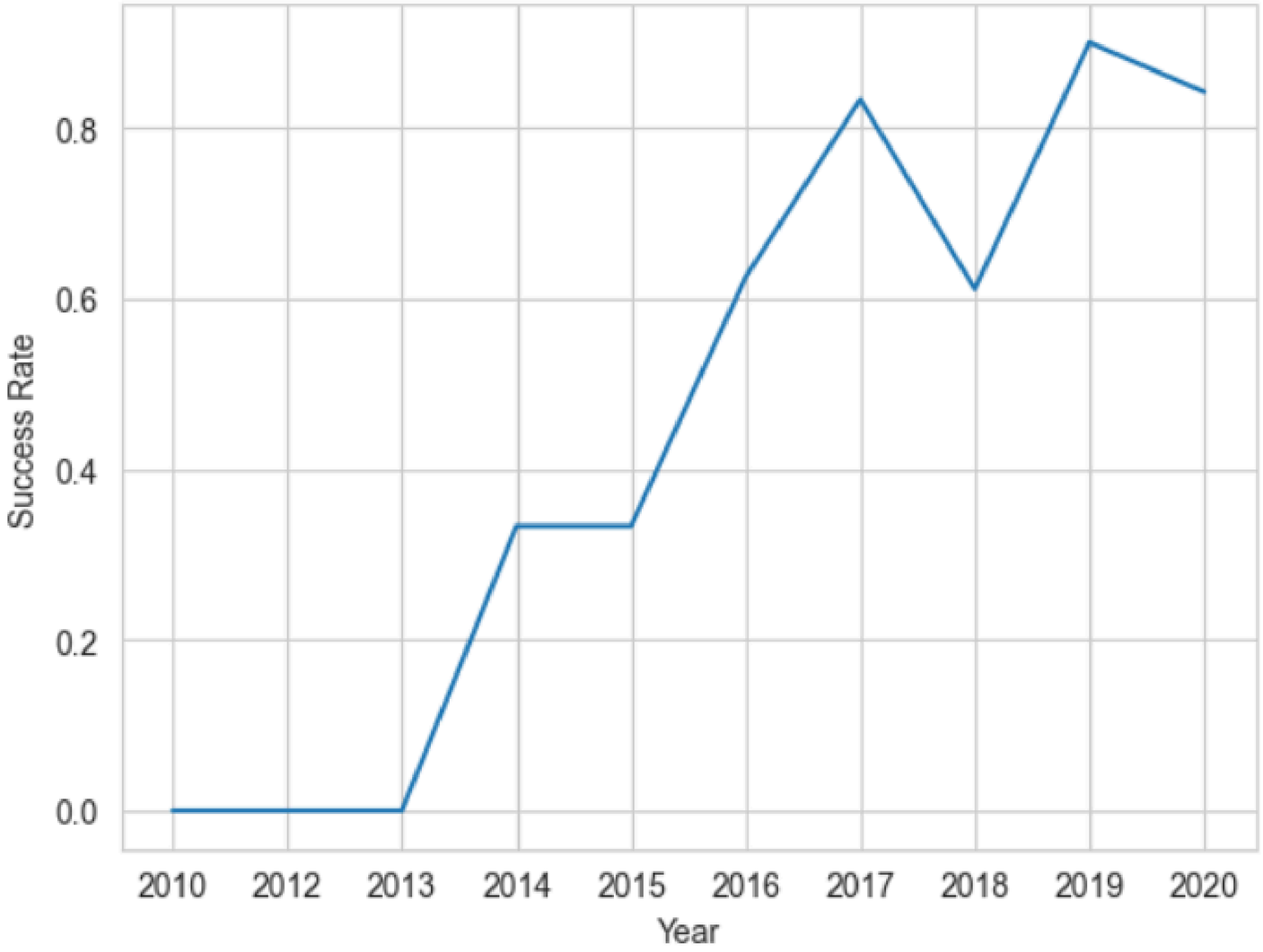
PAYLOAD VS. ORBIT TYPE



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



The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

ALL LAUNCH SITE NAMES

- Use SELECT DISTINCT to find all the Launch Site from SpaceXTBL
- We only have 4 Launch Site:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
In 8   1  %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
          Executed at 2024.05.15 11:30:21 in 7ms

           * sqlite:///my_data1.db
          Done.
Out 8     [('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

LAUNCH SITE NAMES BEGIN WITH 'CCA'

- SELECT 5 rows from SPACEXTBL using WHERE statement and wildcard ‘%’ to get the result.

```
In 10  1  %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
          Executed at 2024.05.15 11:31:32 in 9ms

           * sqlite:///my_data1.db
          Done.

Out 10   [('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', 'Succ
          ('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

- Use SUM(PAYLOAD_MASS__KG_) to get the Total Payload KG from Customer = ‘NASA (CRS)’

```
In 11   1  %sql SELECT SUM(PAYLOAD_MASS__KG_) as TOTAL_PAYLOAD_KG FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
           Executed at 2024.05.15 11:32:57 in 8ms

            * sqlite:///my_data1.db
           Done.

Out 11     [(45596,)]
```

AVERAGE PAYLOAD MASS BY F9 V1.1

- Using WHERE statement to get all the rows of Booster_Version = ‘F9 v1.1’. After that use AVG(PAYLOAD_MASS__KG_) to get the result

```
In 12   1  %sql SELECT AVG(PAYLOAD_MASS__KG_) as AVG_PAYLOAD_KG FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1'
           Executed at 2024.05.15 11:33:39 in 9ms

            * sqlite:///my_data1.db
           Done.

Out 12     [(2928.4,)]
```

FIRST SUCCESSFUL GROUND LANDING DATE

- List the date when the first successful landing outcome in ground pad was achieved

```
In 13   1  %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)'
           Executed at 2024.05.15 11:34:26 in 8ms

            * sqlite:///my_data1.db
           Done.

Out 13     [('2015-12-22',)]
```

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

- List the names of the booster which have success in drone ship and have payload mass between 4000 and 6000

```
In 15  1  %sql select Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
          Executed at 2024.05.15 11:34:57 in 8ms

           * sqlite:///my_data1.db
          Done.
Out 15    [('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

- List the total number of lauching outcomes use COUNT and GROUP BY

```
In 16   1  %sql select Mission_Outcome, count(*) as Total_Count from SPACEXTABLE group by Mission_Outcome
           Executed at 2024.05.15 11:35:18 in 7ms

            * sqlite:///my_data1.db
           Done.

Out 16     [('Failure (in flight)', 1),
            ('Success', 98),
            ('Success ', 1),
            ('Success (payload status unclear)', 1)]
```

BOOSTERS CARRIED MAXIMUM PAYLOAD

- List the names of the booster which have carried the maximum payload mass
- I use Sub Query to find the MAX Payload Mass. After that using WHERE to find all the Booster Version have Max Payload Mass.

```
In 17  1  %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max (PAYLOAD_MASS__KG_) from SPACEXTABLE)
          Executed at 2024.05.15 11:35:33 in 10ms

           * sqlite:///my_data1.db
          Done.

Out 17   [('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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Use strftime(‘%Y’, Date) = ‘2015’ to get all rows in Year = 2015. And i use SELECT CASE and strftime to get the MONTH NAME.

In 28

```
%%sql
select
    CASE strftime('%m', Date)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS month_name,
    Booster_Version, Launch_Site
from SPACEXTABLE
where (Landing_Outcome = 'Failure (drone ship)') AND
    (strftime('%Y', Date) = '2015');
```

Executed at 2024.05.15 11:40:24 in 7ms

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

Out 28

```
[('January', 'F9 v1.1 B1012', 'CCAFS LC-40'),
 ('April', 'F9 v1.1 B1015', 'CCAFS LC-40')]
```

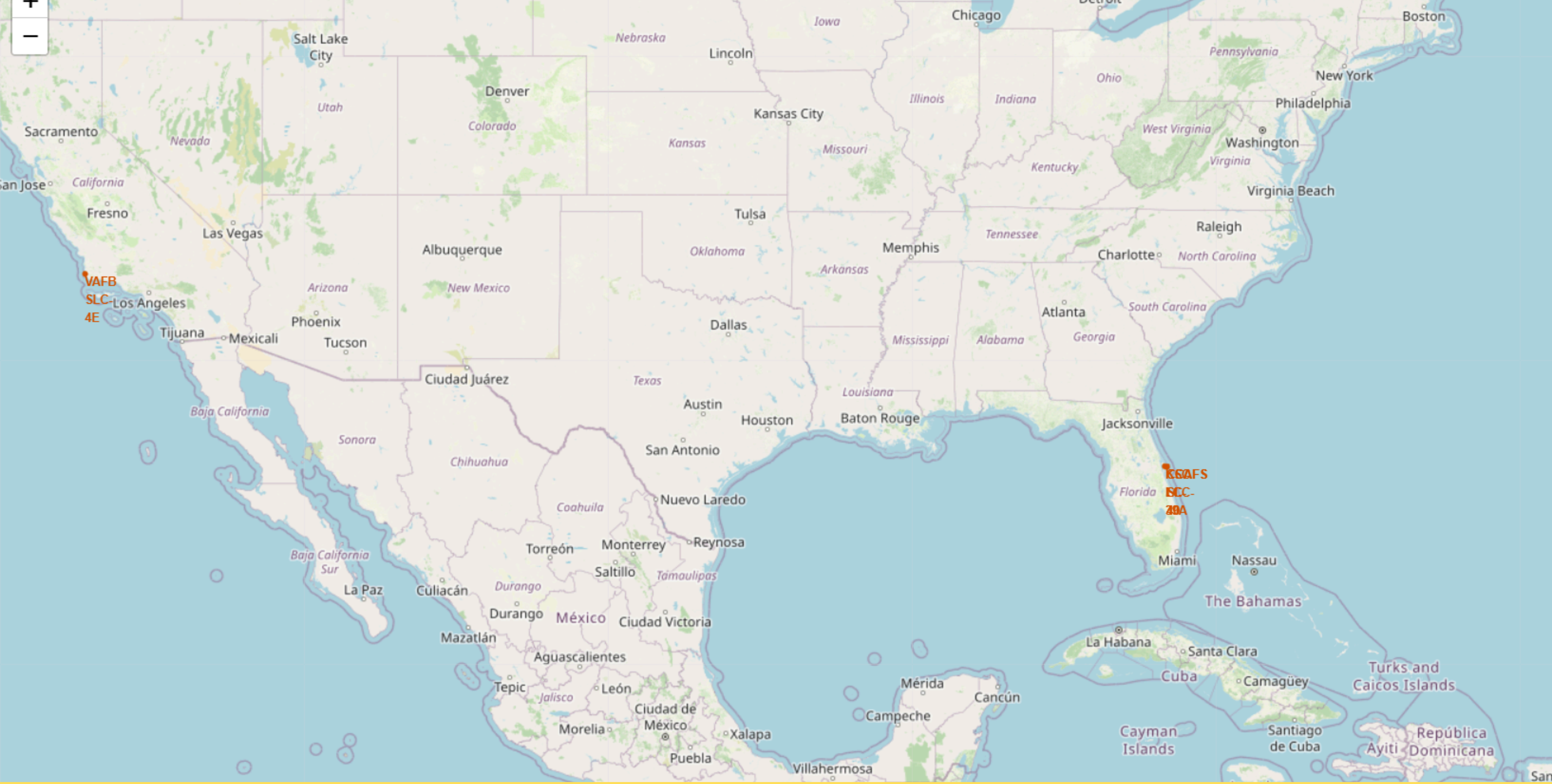
RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

- 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.
- Filter the Data first. Then i use order by slurp DESC to sort the value in descending order.

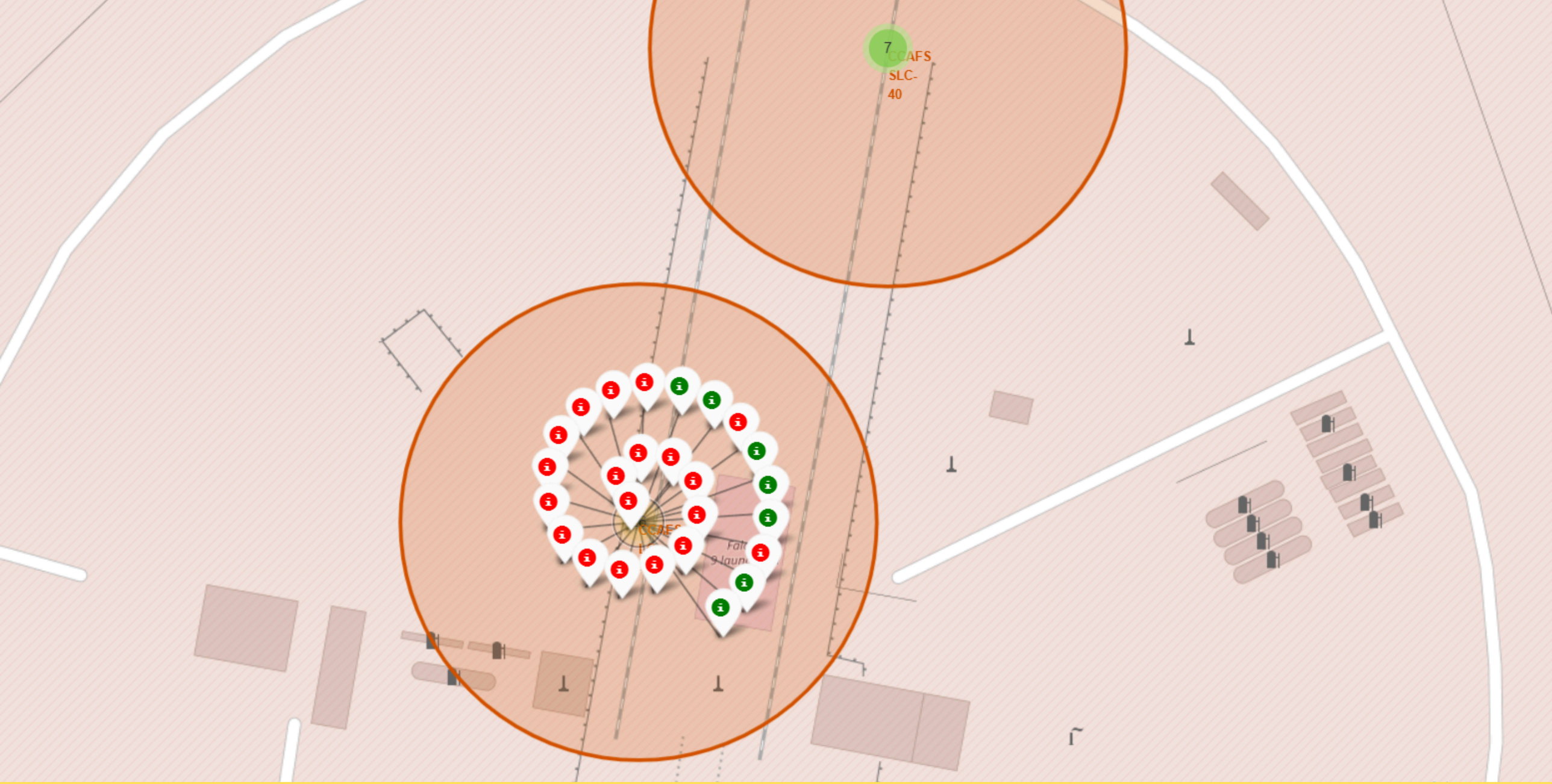
```
In 30
%%sql
SELECT COUNT(*) as slurp , Landing_Outcome
FROM SPACEXTABLE
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY slurp DESC
Executed at 2024.05.15 11:41:48 in 6ms

 * sqlite:///my_data1.db
Done.

Out 30
[(10, 'No attempt'),
 (5, 'Success (drone ship)'),
 (5, 'Failure (drone ship)'),
 (3, 'Success (ground pad)'),
 (3, 'Controlled (ocean)'),
 (2, 'Uncontrolled (ocean)'),
 (2, 'Failure (parachute)'),
 (1, 'Precluded (drone ship)')]
```

Section 3

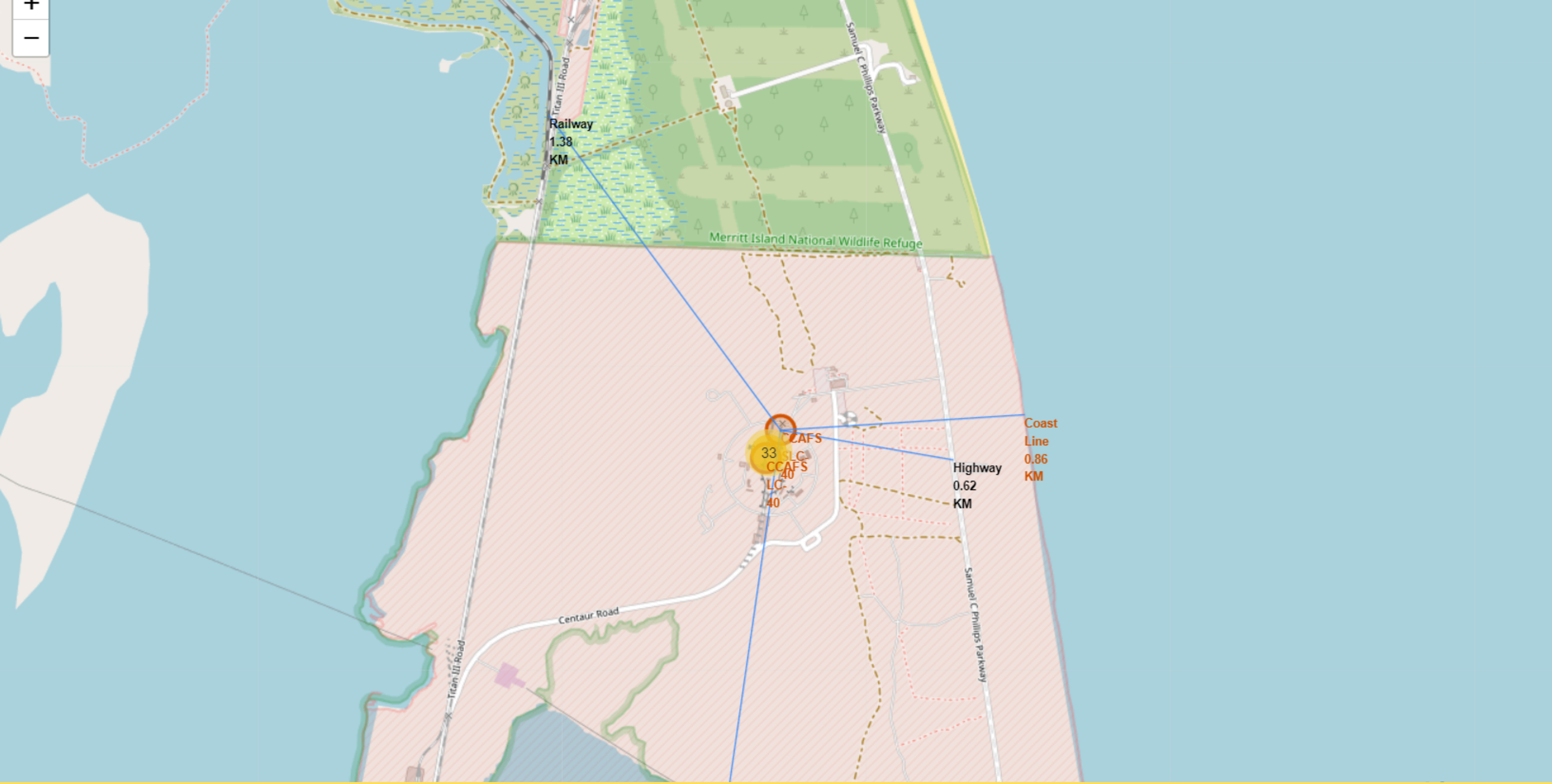
Launch Sites Proximities Analysis



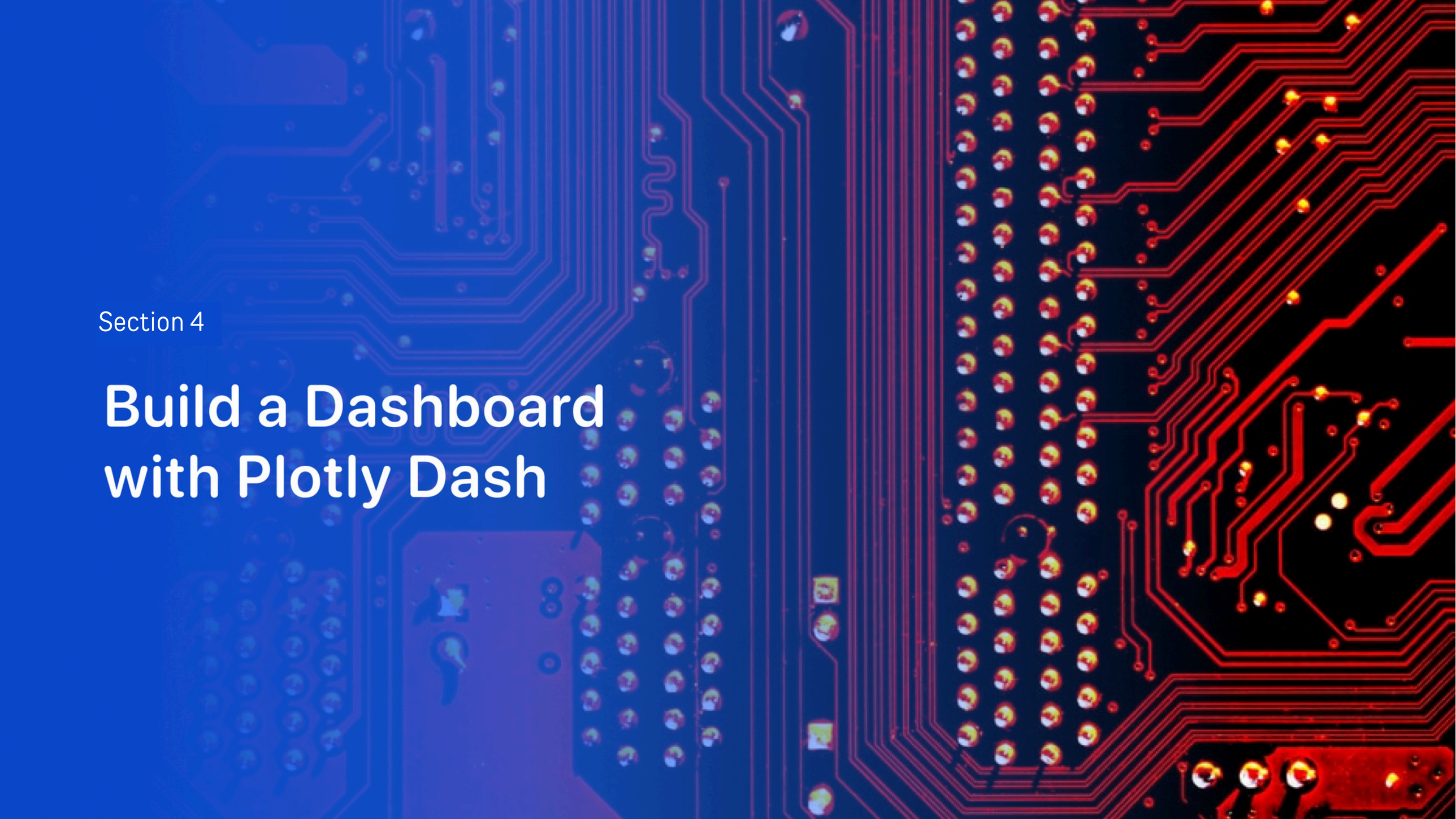
ALL SPACEX LAUNCH SITES



SUCCESS / FAILED LAUNCHES



DISTANCES BETWEEN A LAUNCH SITE TO ITS PROXIMITIES



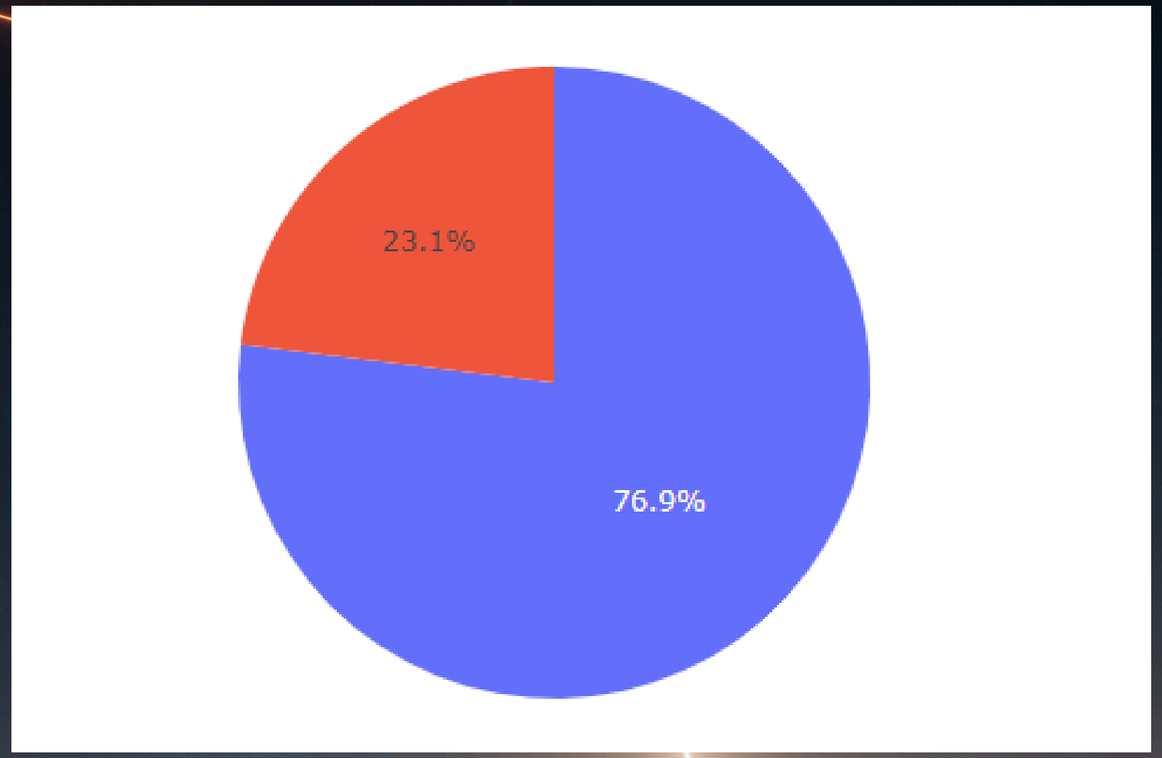
Section 4

Build a Dashboard with Plotly Dash

TOTAL SUCCESSFUL LAUNCHES BY ALL SITES

We can see that KSC LC-39A has the most successful launches from all the sites

TOTAL SUCCESS LAUNCHES FOR SITE KSC LC-39A

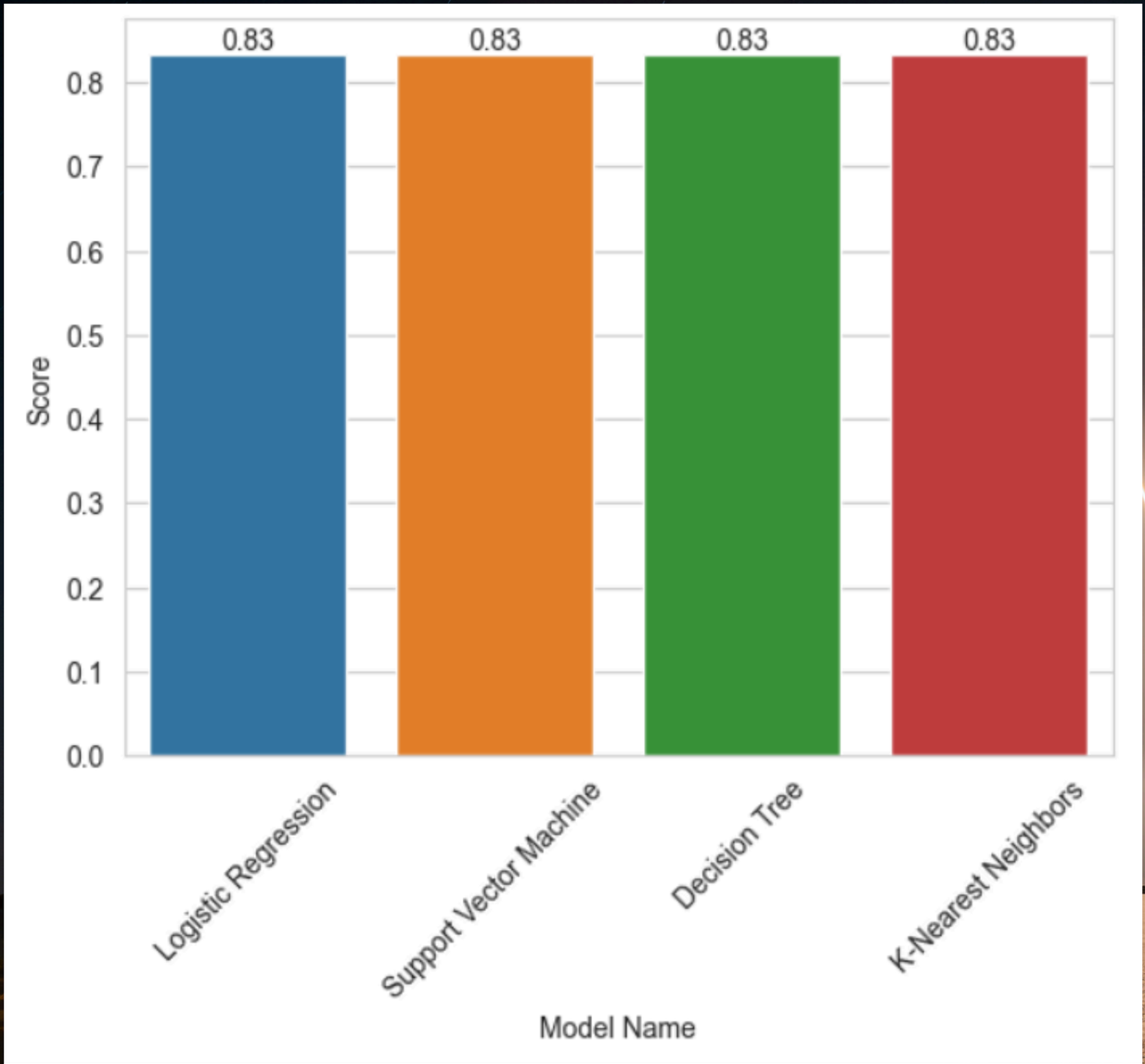


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.

Section 5

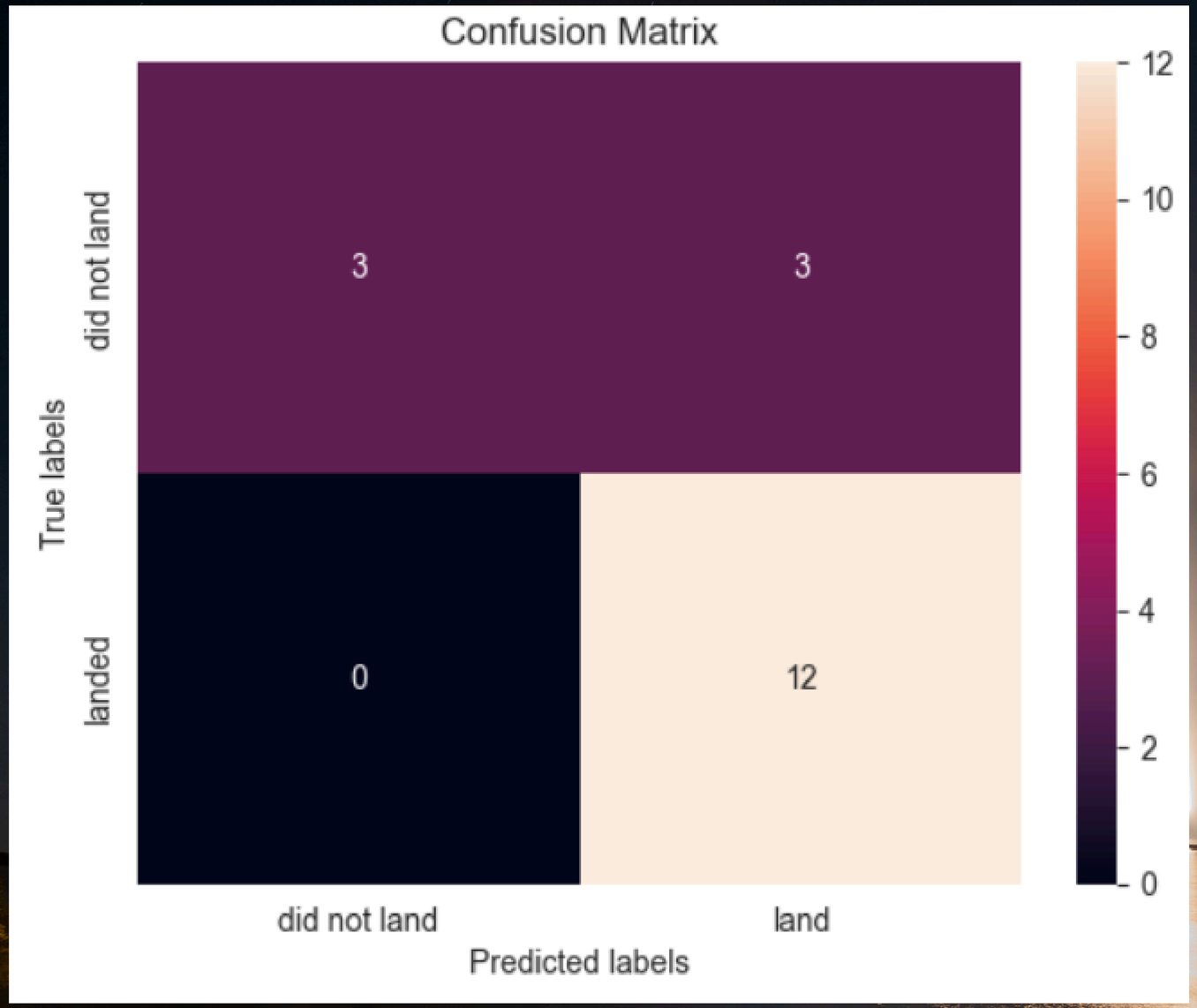
Predictive Analysis (Classification)

CLASSIFICATION ACCURACY



All the model i use have the same accuracy is 0.83. It is good accuracy score and you can use 1 of the 4 models.

Confusion Matrix



The accuracy score is so high. The rate of predicting landing successful is good.

CONCLUSIONS

In this project, we undertook a thorough examination of SpaceX's launch data to derive insights into the company's launch history, success rates, and payload trends. Our investigation led to several important discoveries:

- **Launch Site Analysis**: We determined that Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC 40) had the highest number of launches among the evaluated sites.
- **Success Rate**: Launches to Geostationary Transfer Orbit (GTO), Highly Elliptical Orbit (HEO), and Sun-Synchronous Orbit (SSO) demonstrated the highest success rates, highlighting the dependability of these missions. • Payload Mass Trends: There was a noticeable increase in payload mass over the years, reflecting SpaceX's advancing capabilities and expanding ambitions.
- **Customer Relations**: NASA (CRS) emerged as SpaceX's main customer, significantly contributing to the total payload mass launched.

APPENDIX

- **Python Code Snippets**: A variety of Python code snippets were employed for data collection, preprocessing, and analysis. These snippets are included in the project's source code.
- **SQL Queries**: SQL queries were used to extract and analyze data from the database. The specific queries are available in the project's SQL script.
- **Charts and Visualizations**: Visual representations, such as bar charts and line graphs, were created to depict key findings and trends. These visualizations are part of the project's report.
- **Jupyter Notebook Outputs**: A Jupyter Notebook was utilized for comprehensive data analysis and exploration. The complete notebook and its outputs are accessible in the project's repository.
- **Data Sets**: Links to both the raw and processed data sets used in the project are provided for reference. These data sets can be found in the project's data directory.

Thank you!