IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zihe(Peter) Zhang
1/20/2024

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- **Project background and context**
  - This project centers around SpaceX, a leading aerospace company established by Elon Musk. Our objective is to examine data pertaining to SpaceX launches, delving into aspects such as success rates, payload trends, rocket performance, landing outcomes, launch site choices, and temporal patterns.
- **Problems you want to find answers**
  - Launch Success: Exploring the factors influencing the success of SpaceX launches.
  - Payload Analysis: Investigating the evolution of payload mass and types over time and assessing their correlation with launch outcomes.
  - Rocket Performance: Analyzing insights into the reliability and reusability of SpaceX's rockets.
  - Landing Outcomes: Identifying patterns in the outcomes of first stage landings.
  - Launch Site Assessment: Evaluating the impact of launch site choices on mission success.
  - Temporal Trends: Examining notable trends or seasonality in SpaceX's launch history.

Section 1

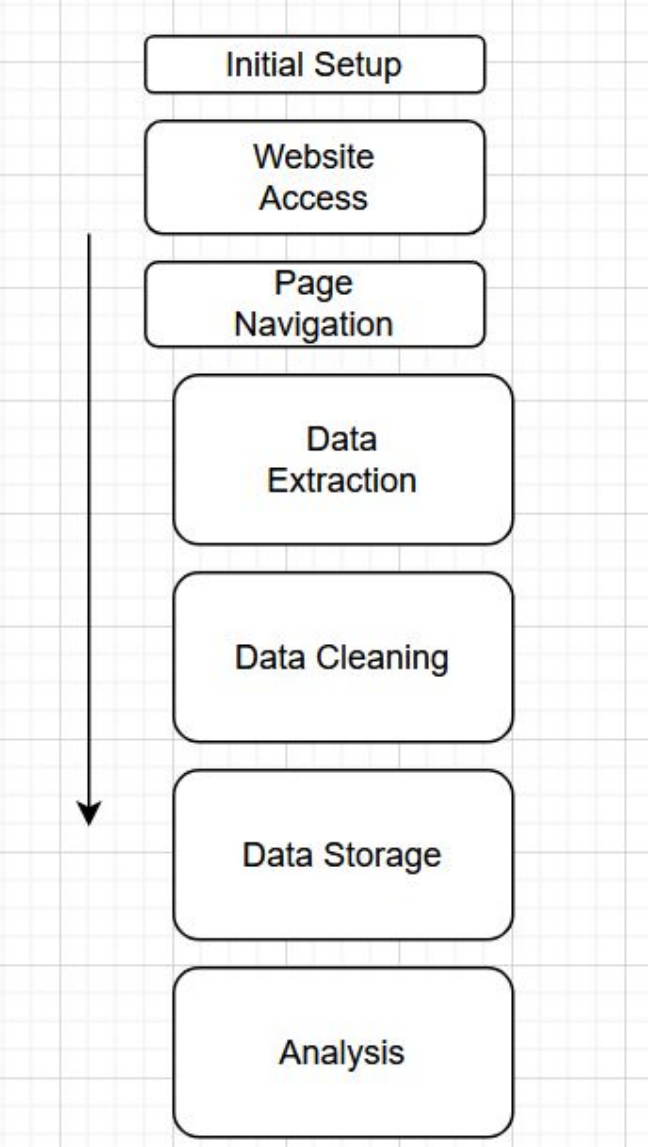
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web scraped from the SpaceX official website
- Perform data wrangling
  - Clean the dataset and only save the useful data columns
- 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:
• Implement web scraping libraries to gain access to SpaceX's website.

Website Access:
• Navigate across web pages to identify pertinent data.
• Gain entry to the SpaceX launch history page.

Page Navigation:
• Retrieve data from web pages.

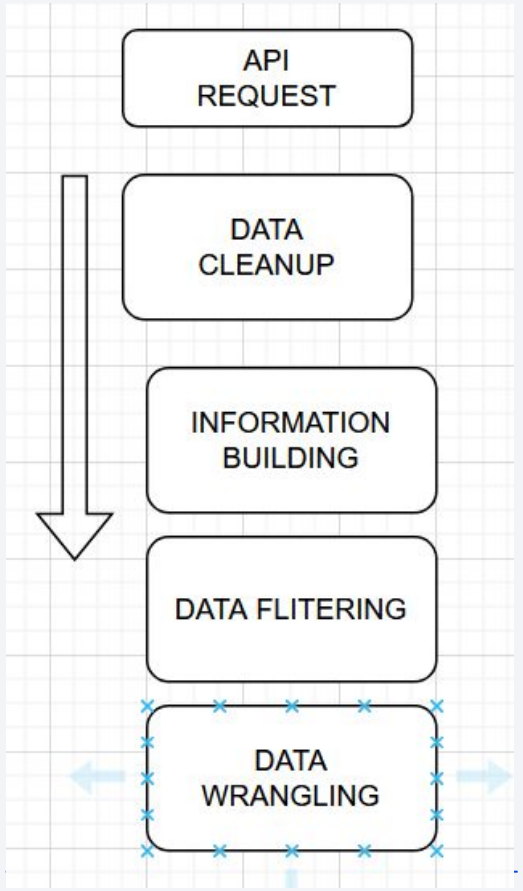
Data Extraction:
• Purify the extracted data to eliminate inconsistencies and errors.
• Scrutinize for any missing values.

Data Cleaning:
• Transfer processed data into a structured format, such as CSV.

Data Storage:
• Employ the collected data for a multitude of analyses.

Data Collection – SpaceX API

- API Request: Utilized SpaceX API to gather launch data.
- Data Cleanup: Formatted and cleaned the obtained data.
- Information Extracted:
  - Booster Version
  - Payload Mass & Orbit
  - Launch Site (Longitude & Latitude)
  - Landing Outcomes & Types
  - Flight Counts
  - Gridfins, Reuse, and Legs
  - Landing Pads
  - Core Blocks, Reuse Counts, and Serials
- Data Filtering: Retained only Falcon 9 launches.
- Data Wrangling: Imputed missing Payload Mass values with the mean



Link to API notebook on Github

Data Collection - Scraping

Objective: Extract Falcon 9 launch records from Wikipedia using web scraping.

• Tools Employed: BeautifulSoup, Requests, Pandas.

• Step 1: Initiate a request for the Falcon 9 Launch Wiki page.

• Step 2: Generate a BeautifulSoup object.

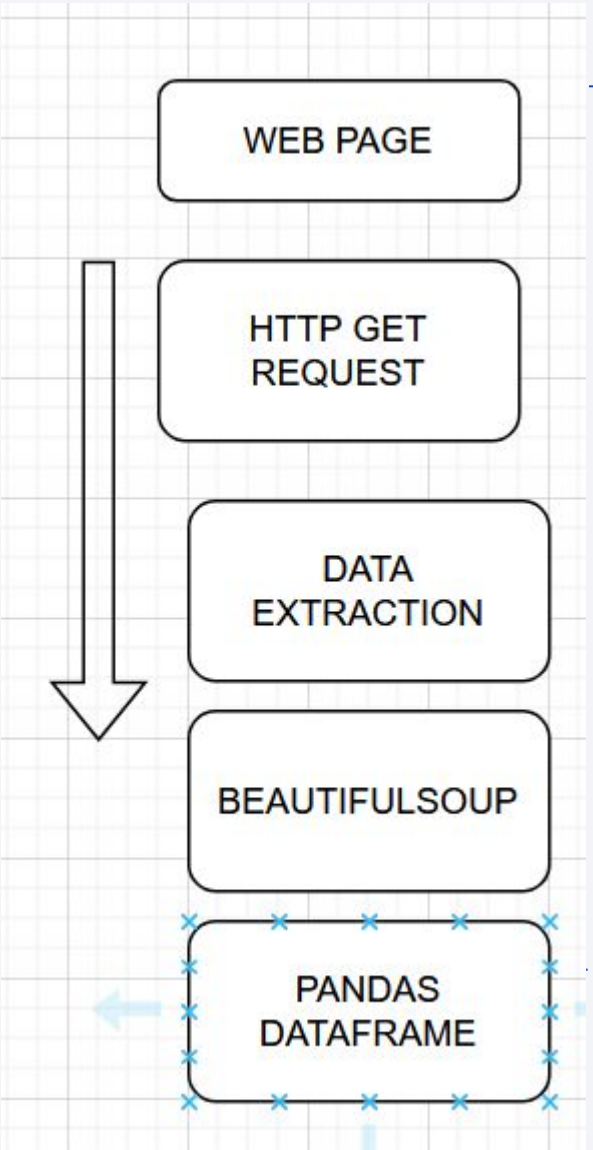
• Step 3: Retrieve column names from the HTML table header.

• Step 4: Establish a dictionary incorporating column names.

• Step 5: Parse and fill the dictionary with launch records.

• Step 6: Transform the dictionary into a Pandas dataframe.

• Step 7: Export the dataframe to a CSV file.



Link to notebook on Github

Data Wrangling

Data Import:

• Imported data from a CSV file containing SpaceX Falcon 9 launch data.

Exploratory Data Analysis (EDA):

• Uncovered patterns and addressed missing values.

• Recognized various attributes, including launch site, orbit type, and landing outcomes.

Identifying Launch Site Data:

• Calculated the number of launches for each launch site.

• Analyzed the launch site data to discern site-specific trends.

Analyzing Orbit Types:

• Examined the number and occurrence of each orbit type.

Determining Mission Outcomes:

• Identified mission outcomes (landing results) and their frequencies.

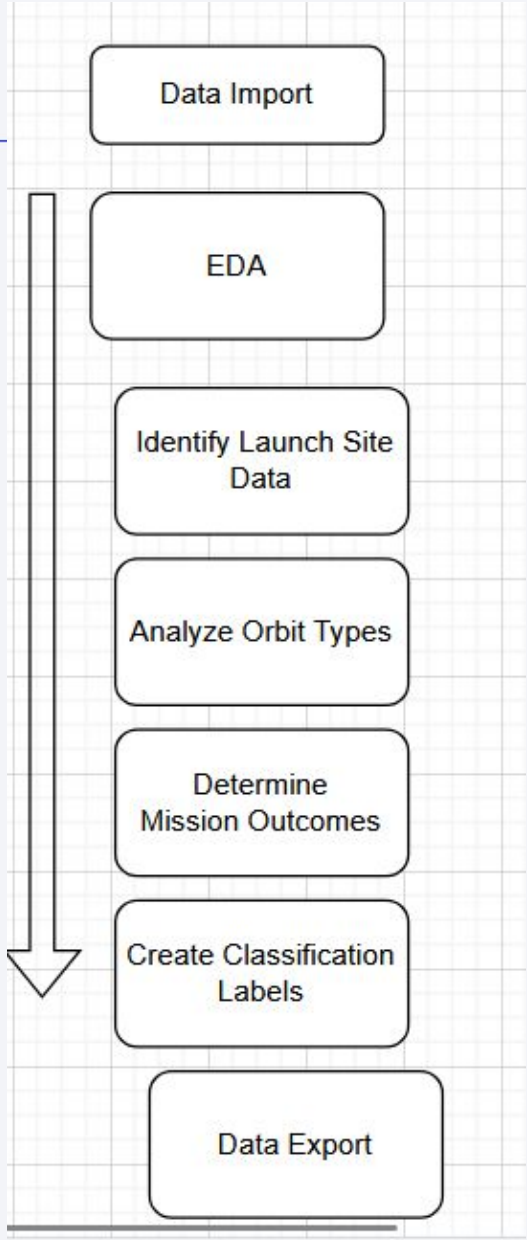
• Created a set of unsuccessful outcomes (bad_outcomes).

Creating Classification Labels:

• Generated a classification variable (landing_class) where 0 represents unsuccessful landings and 1 represents successful landings.

Data Export:

• Exported the processed data to a CSV file (dataset_part_2.csv) for further analysis.



Link to notebook on Github

EDA with Data Visualization

FlightNumber vs. PayloadMass Scatter Plot:

• Aim: Investigate the correlation between flight number and payload mass.

FlightNumber vs. Launch Site Scatter Plot:

• Aim: Examine the relationship between flight number and launch site.

PayloadMass vs. Launch Site Scatter Plot:

• Aim: Explore the connection between payload mass and launch site.

Success Rate vs. Orbit Bar Chart:

• Aim: Visualize the success rate of different orbit types.

FlightNumber vs. Orbit Scatter Plot:

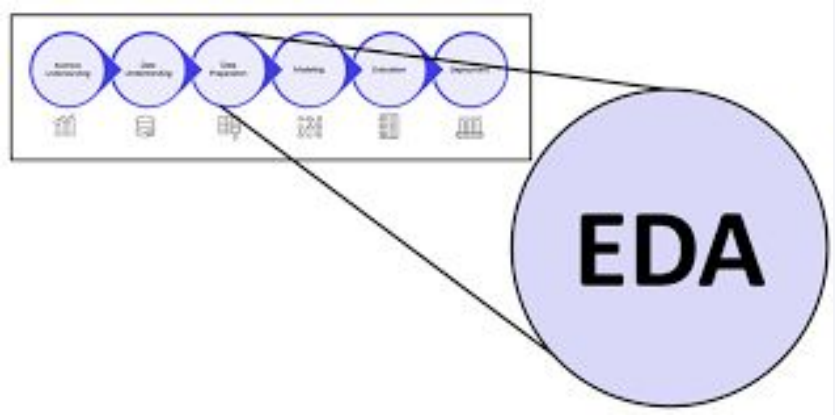
• Aim: Explore the relationship between flight number and orbit type.

PayloadMass vs. Orbit Scatter Plot:

• Aim: Investigate the relationship between payload mass and orbit type.

Launch Success Yearly Trend Line Chart:

• Aim: Analyze the trend in the average launch success rate over the years.



Link to Github notebook

EDA with SQL

Link to Github notebook

Unique Launch Sites: Presented a list of distinct launch site names.

Launch Sites Starting with 'CCA': Displayed 5 records associated with launch sites beginning with 'CCA'.

Total Payload Mass for NASA (CRS): Computed the total payload mass for NASA (CRS) missions.

Average Payload Mass for F9 v1.1: Calculated the average payload mass for booster version F9 v1.1.

First Successful Ground Pad Landing Date: Determined the date of the initial successful ground pad landing.

Boosters with Successful Drone Ship Landings: Enumerated boosters that accomplished successful landings on drone ships with a payload mass ranging between 4000 and 6000.

Mission Outcomes: Tallied and listed the total number of both successful and failed mission outcomes.

Max Payload Mass Booster Versions: Identified booster versions with the maximum payload mass using a subquery.

Failed Drone Ship Landings in 2015: Compiled records for drone ship landings that failed in 2015, including month names.

Ranking Landing Outcomes: Ranked landing outcomes between specific dates in descending order.

Build an Interactive Map with Folium

Markers: These indicate particular locations or points of interest on the map. Clicking on markers reveals information and is valuable for emphasizing important locations.

Polyline: Polylines were employed to link multiple points on the map, forming a visual path or route. This aids users in comprehending connections between locations.

Marker Cluster: To avoid map clutter, nearby markers were grouped into clusters. This enhances map readability, particularly when multiple markers are in close proximity.

These objects were incorporated to enhance visualization, depict connectivity, and elevate the overall user experience.

Link to notebook on Github

Build a Dashboard with Plotly Dash

Build a Plotly Dash Dashboard:

Dropdown Selection for Statistics:

- Users can opt for either "Yearly Statistics" or "Recession Period Statistics."

Dropdown Selection for Year:

- Users have the flexibility to choose a specific year for "Yearly Statistics."

Disabled Year Dropdown:

- The year dropdown is inactive when exploring "Recession Period Statistics."

Graph Output Container:

- A dynamic space designed to showcase interactive graphs.

Yearly Statistics Graph (Example):

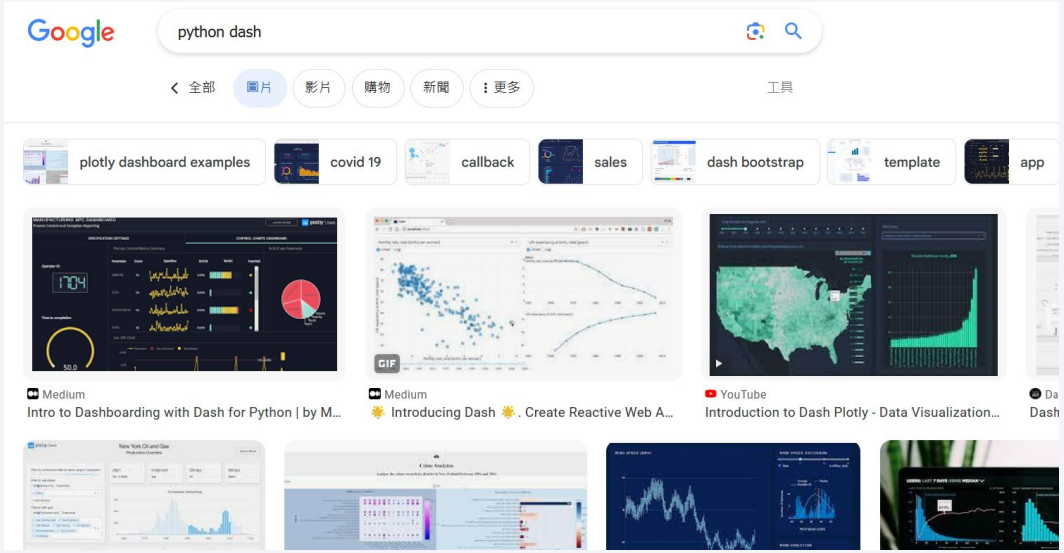
- A bar chart illustrating monthly automobile sales for a selected year.

Recession Period Statistics Graph (Example):

- A line chart depicting automobile sales during recession periods.

Purpose:

- Investigate historical automobile sales data.
- Analyze annual sales trends.
- Comprehend the impact of recessions on sales.



[link to github dash file]

Predictive Analysis (Classification)

Data Exploration:

- Load the dataset
- Explore the data

Data Preprocessing:

- Formulate a binary target variable
- Standardize the data
- Split the data (80% train, 20% test)

Model Selection:

- Consider Logistic Regression, SVM, Decision Tree, and KNN

Hyperparameter Tuning:

- Employ GridSearchCV (cv=10)
- Fine-tune hyperparameters

Model Training & Evaluation:

- Train with the best parameters
- Assess accuracy on the test data

Confusion Matrix:

- Analyze false positives/negatives

Best Model:

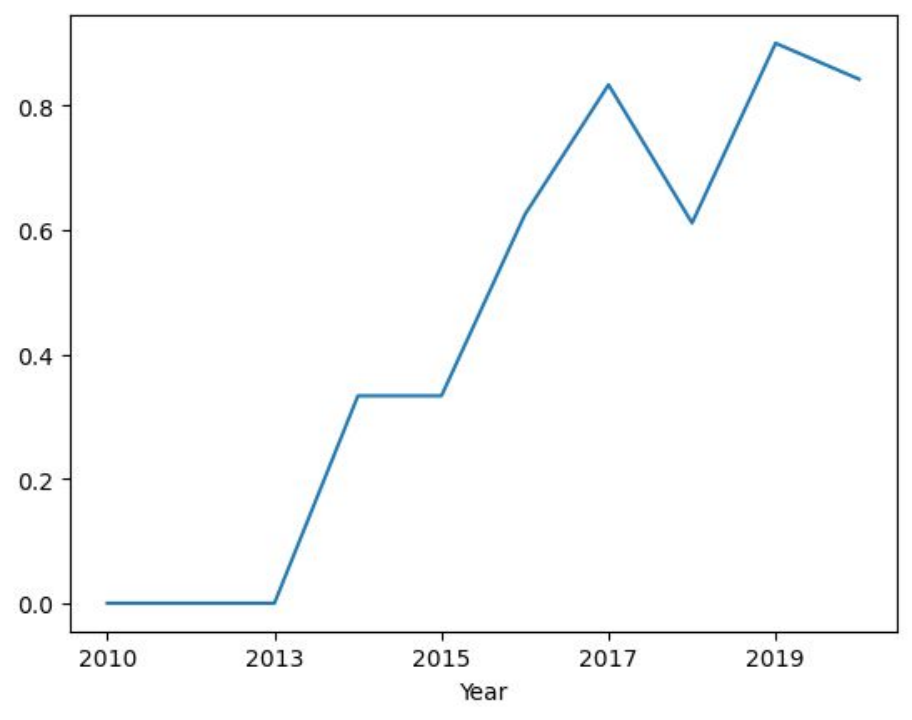
- Decision Tree (83.33% accuracy)

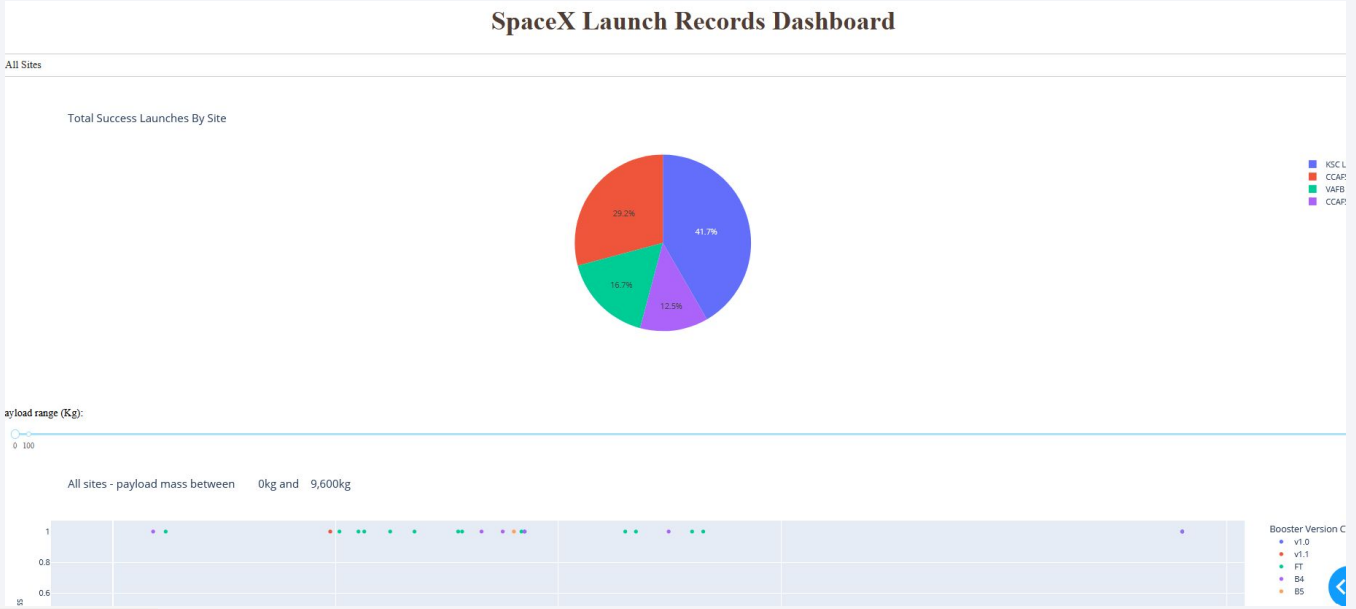
Conclusion:

- The model predicts rocket landings
- There is room for improvement

Link to github

Results

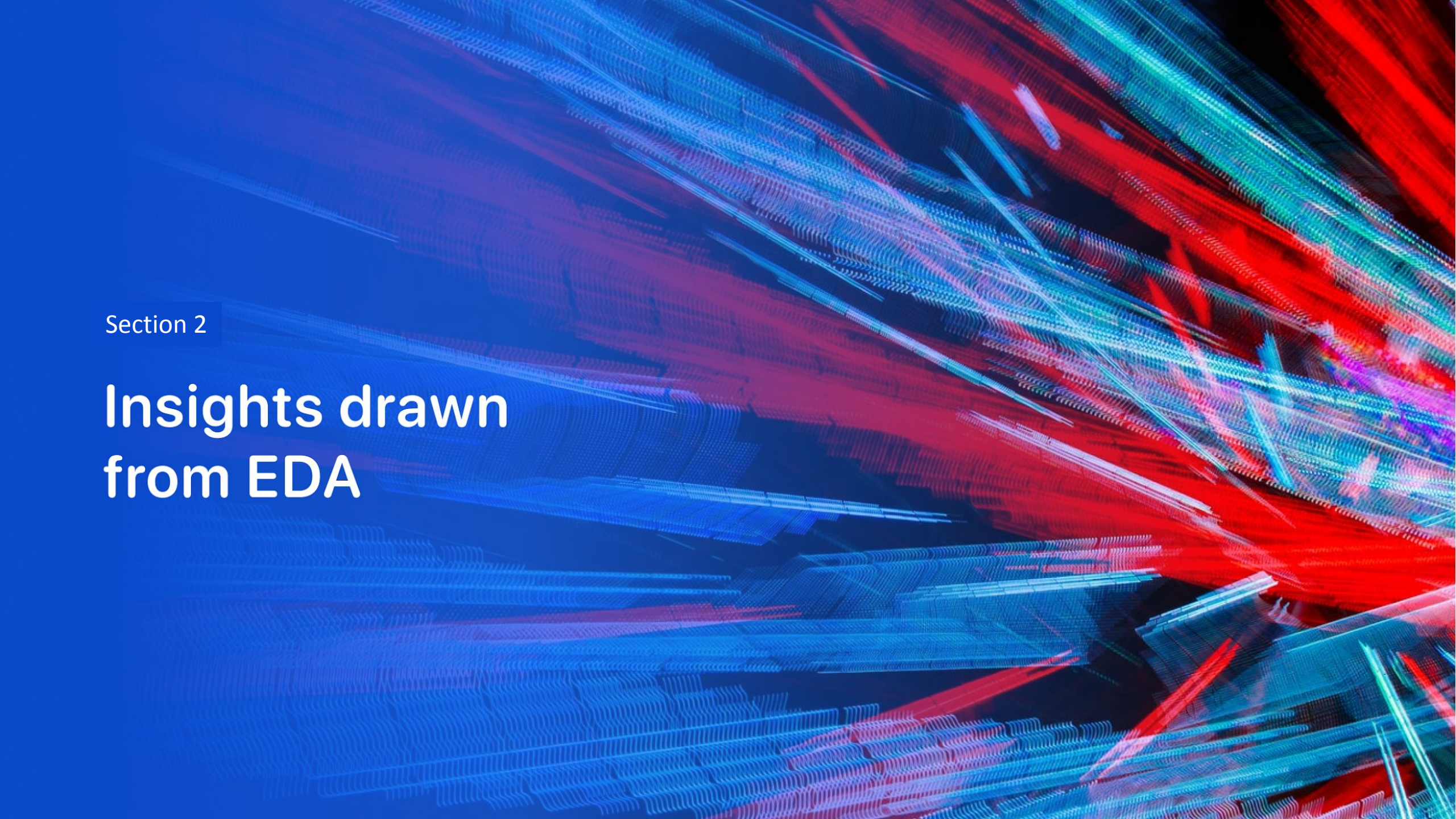




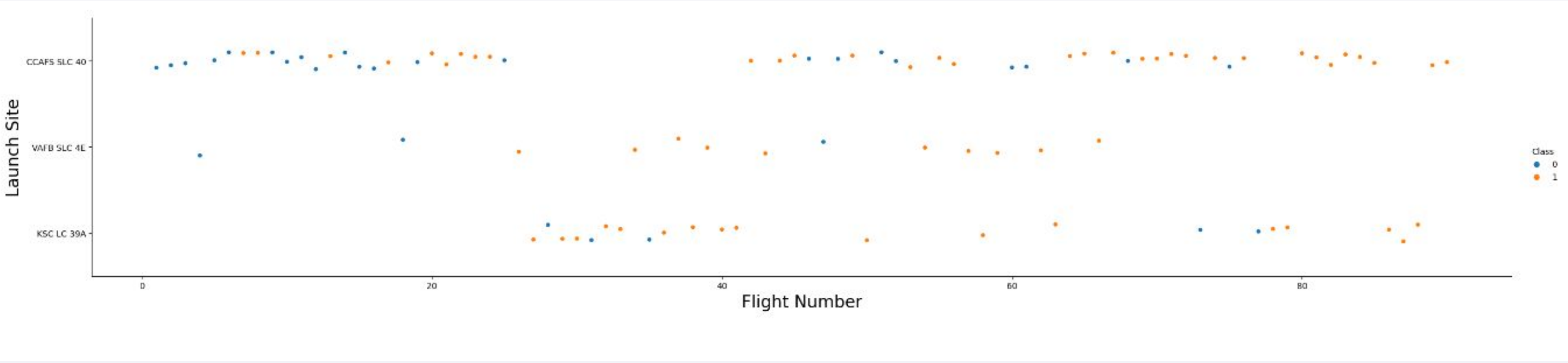
Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.8875	0.83333
KNN	0.84821	0.83333

Section 2

Insights drawn from EDA

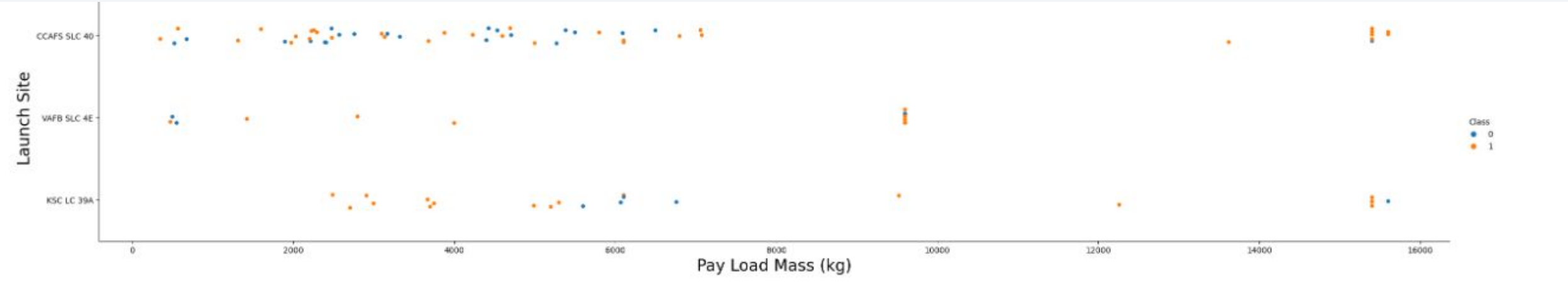


Flight Number vs. Launch Site



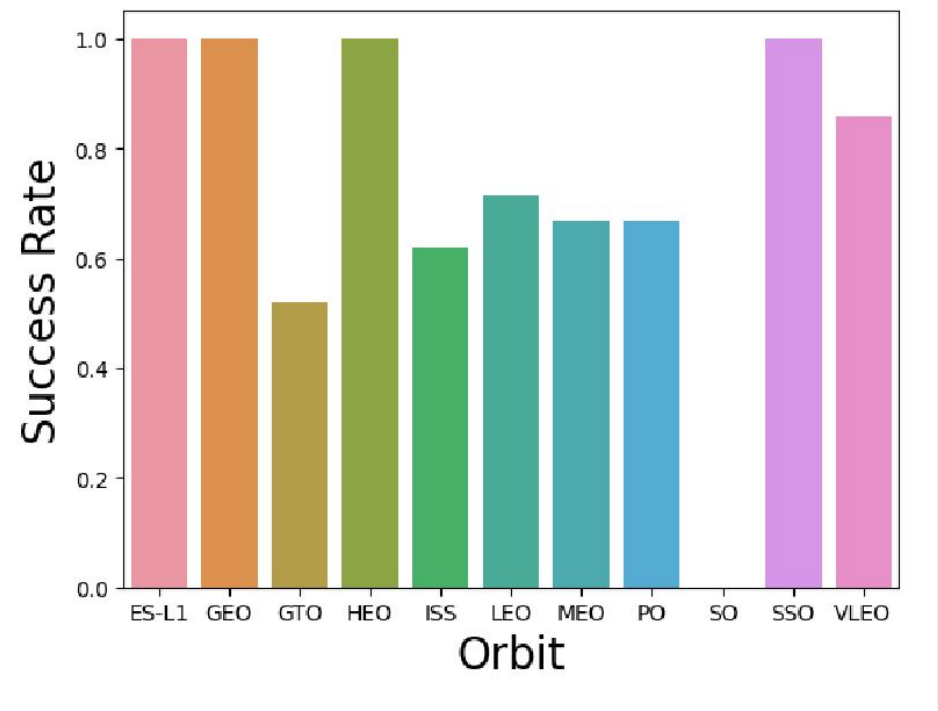
• It is evident that Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC 40) experienced a higher number of flights compared to the other launch sites.

Payload vs. Launch Site



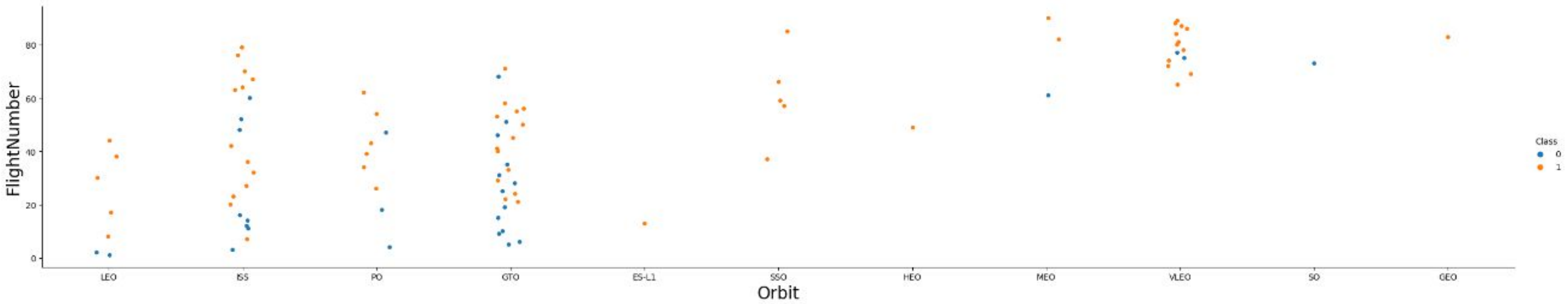
In the scatter point chart for the Launch Site, it can be observed that there are no rockets launched with a heavy payload mass (greater than 10000) at the VAFB-SLC launch site.

Success Rate vs. Orbit Type



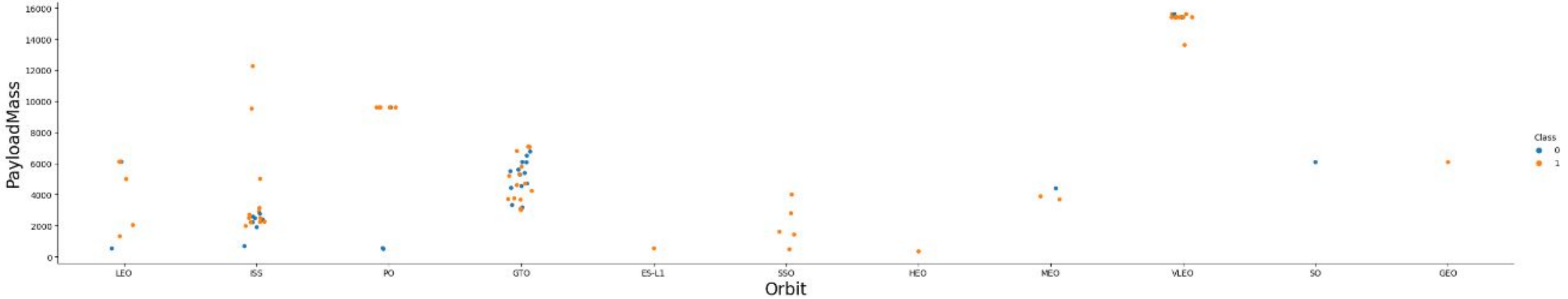
It is evident that missions targeting Earth-Sun L1 (ES-L1), Geostationary Orbit (GEO), Sun-Synchronous Orbit (SSO), and Highly Elliptical Orbit (HEO) exhibited the highest success rates.

Flight Number vs. Orbit Type



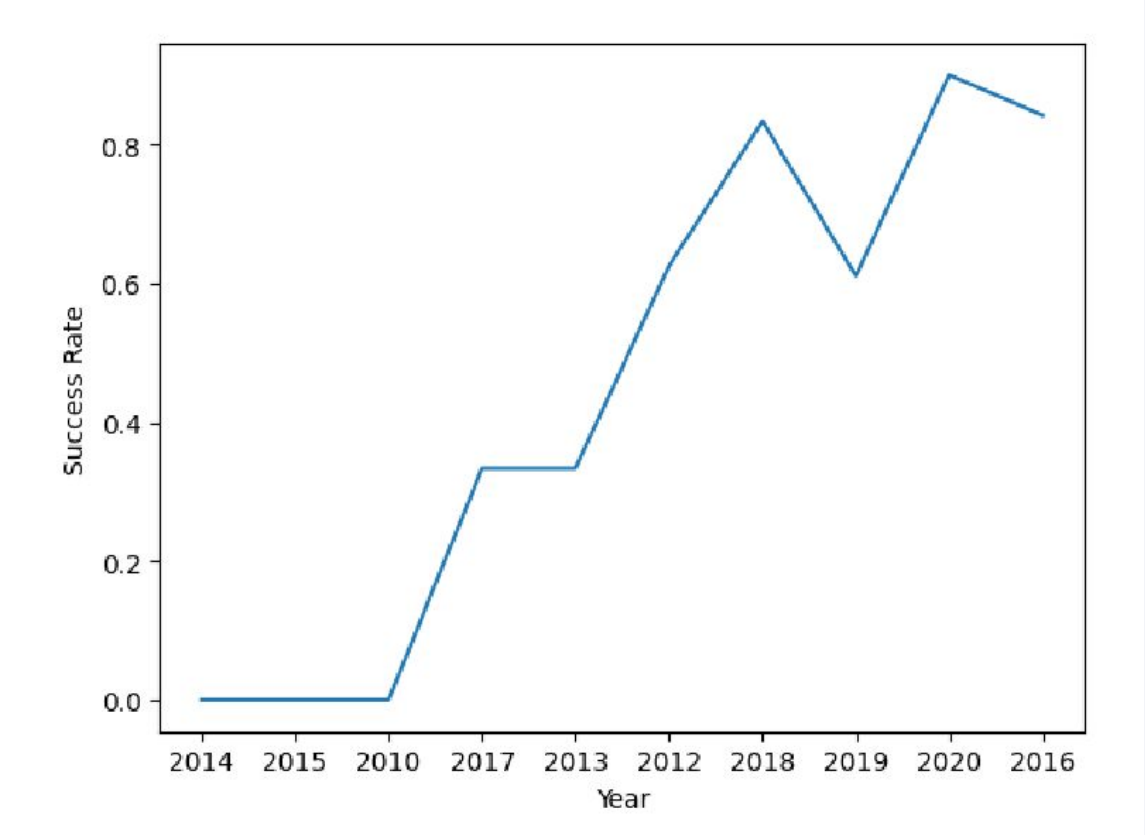
For the Low Earth Orbit (LEO), success appears to be correlated with the number of flights. Conversely, there seems to be no discernible relationship between flight number and success in the Geostationary Transfer Orbit (GTO).

Payload vs. Orbit Type



When dealing with heavy payloads, the rate of successful or positive landings is higher for Polar, Low Earth Orbit (LEO), and International Space Station (ISS). However, for Geostationary Transfer Orbit (GTO), it is challenging to distinguish well, as both positive landing rates and negative landing (unsuccessful mission) occurrences are observed.

Launch Success Yearly Trend



The success rate has consistently increased since 2013, reaching its peak in 2020.

All Launch Site Names

```
cur.execute('select distinct Launch_Site from SPACEXTABLE')
cur.fetchall()
```

```
[('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

SQL query using the SELECT DISTINCT statement to retrieve unique valu
es from the "Launch_Site" column in a table called "SPACEXTABLE."

Launch Site Names Begin with 'CCA'

```
cur.execute('select * from SPACEXTABLE where Launch_Site like "CCA%" limit 5')
cur.fetchall()
```

Retrieve the initial 5 rows from the "SPACEXTABLE" table where the entries in the "Launch_Site" column begin with "CCA."

Total Payload Mass

```
cur.execute('select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = "NASA (CRS)"')
cur.fetchall()
```

```
[(45596,)]
```

Calculate the sum of the "PAYLOAD_MASS__KG_" column for rows where the "Customer" column is equal to "NASA (CRS)."

Average Payload Mass by F9 v1.1

```
cur.execute('select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = "F9 v1.1"')
cur.fetchall()
```

```
[(2928.4,)]
```

Compute the average payload mass for rows in the "SPACEXTABLE" where the "Booster_Version" is equal to "F9 v1.1."

First Successful Ground Landing Date

```
cur.execute('select min(Date) from SPACEXTABLE where Landing_Outcome = "Success (ground pad)"')
cur.fetchall()
```

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

The date when the first successful landing outcome on the ground pad was achieved is not provided in the information given. If you have the specific dataset or details, you can query or search for the exact date in your data.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
cur.execute('select Booster_Version from SPACEXTABLE where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ be
cur.fetchall()
```

```
[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

Provide the names of the boosters that have achieved success in landing on a drone ship and have a payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
cur.execute('select Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome')
cur.fetchall()
```

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

Provide the total number of successful and failed mission outcomes.

Boosters Carried Maximum Payload

```
cur.execute('select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max (PAYLOAD_MASS__KG_) from SPACEXTA
cur.fetchall()
```

```
[('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 ',)]
```

List the names of the booster_versions which have carried the maximum payload mass

2015 Launch Records

```
import calendar
cur.execute('select substr(Date, 6, 2) as month_name, Booster_Version, Launch_Site from SPACEXTABLE where Landing_Outcome =
res = cur.fetchall()
res = [(calendar.month_name[int(t[0])], t[1], t[2]) for t in res]
res
```

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

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

```
cur.execute('select count(*) as slurp , Landing_Outcome from SPACEXTABLE where Date between "2010-06-04" and "2017-03-20" gr
cur.fetchall()
```

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

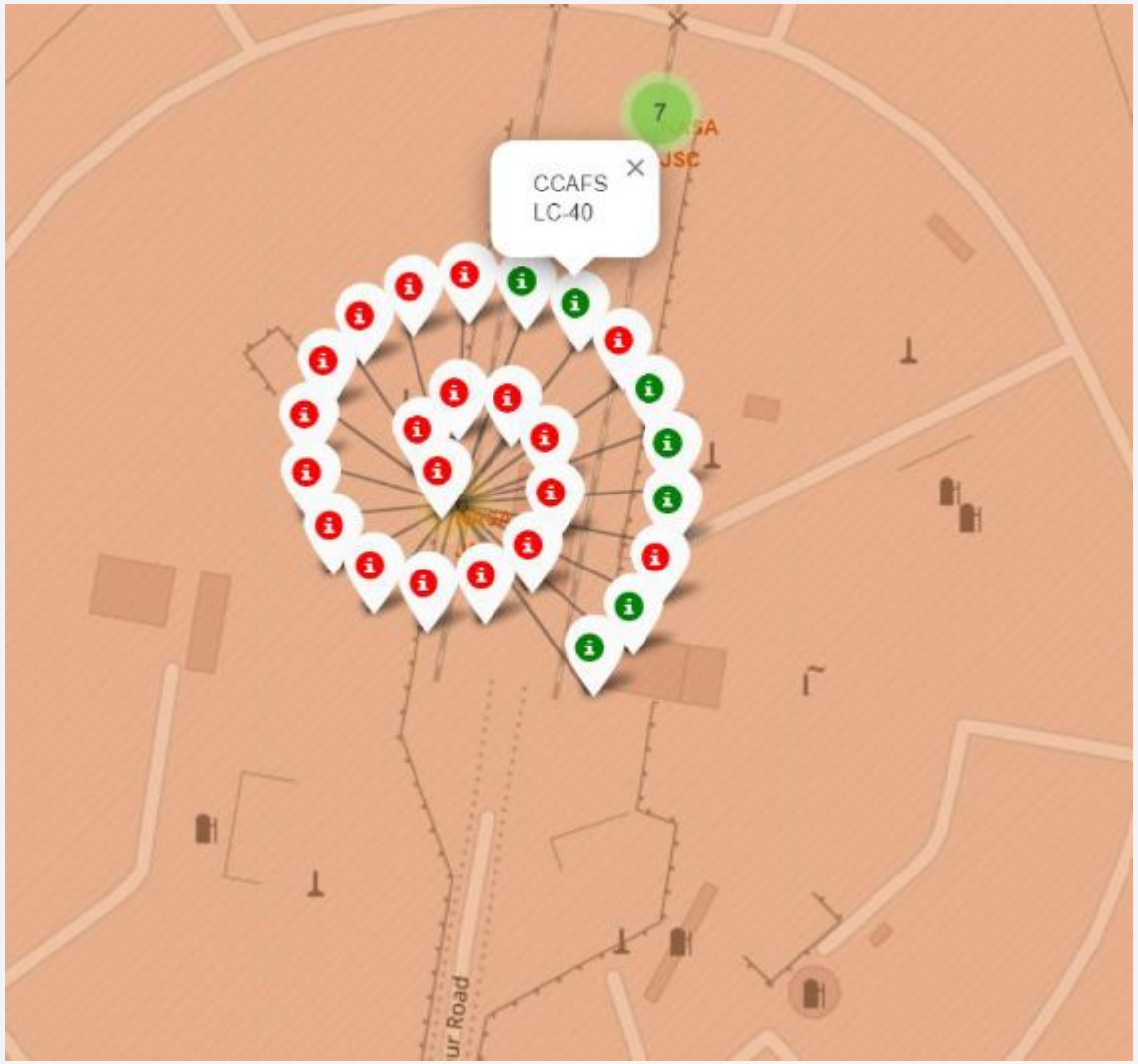
Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20 in descending order.


Section 3
Launch Sites
Proximities Analysis

All launch Sites

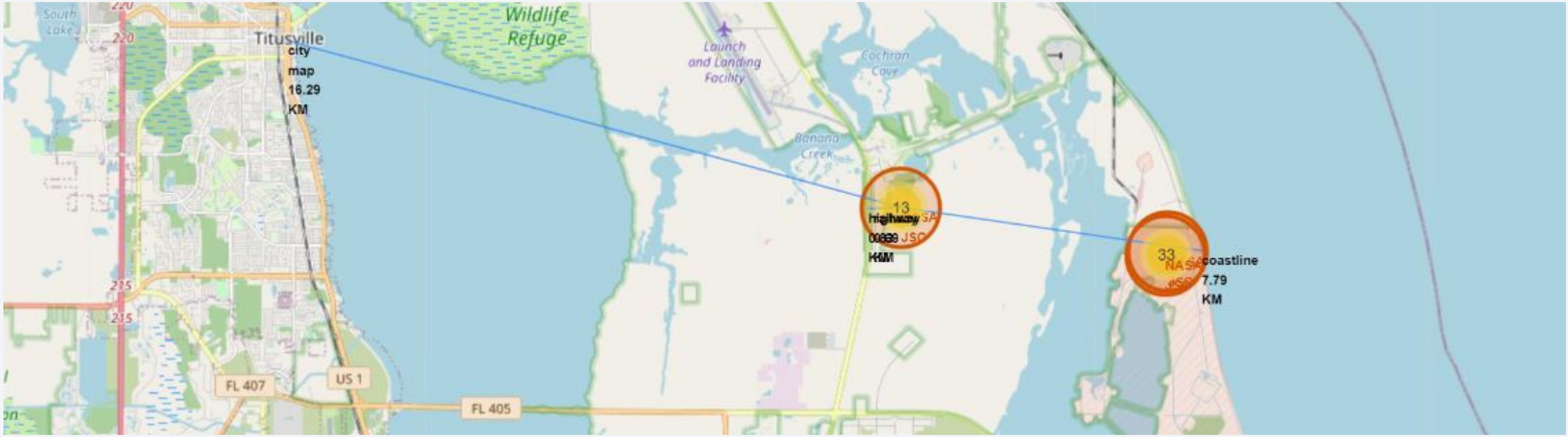
It's noteworthy that all SpaceX launch locations are situated close to the coastal line.

Success/failed launches

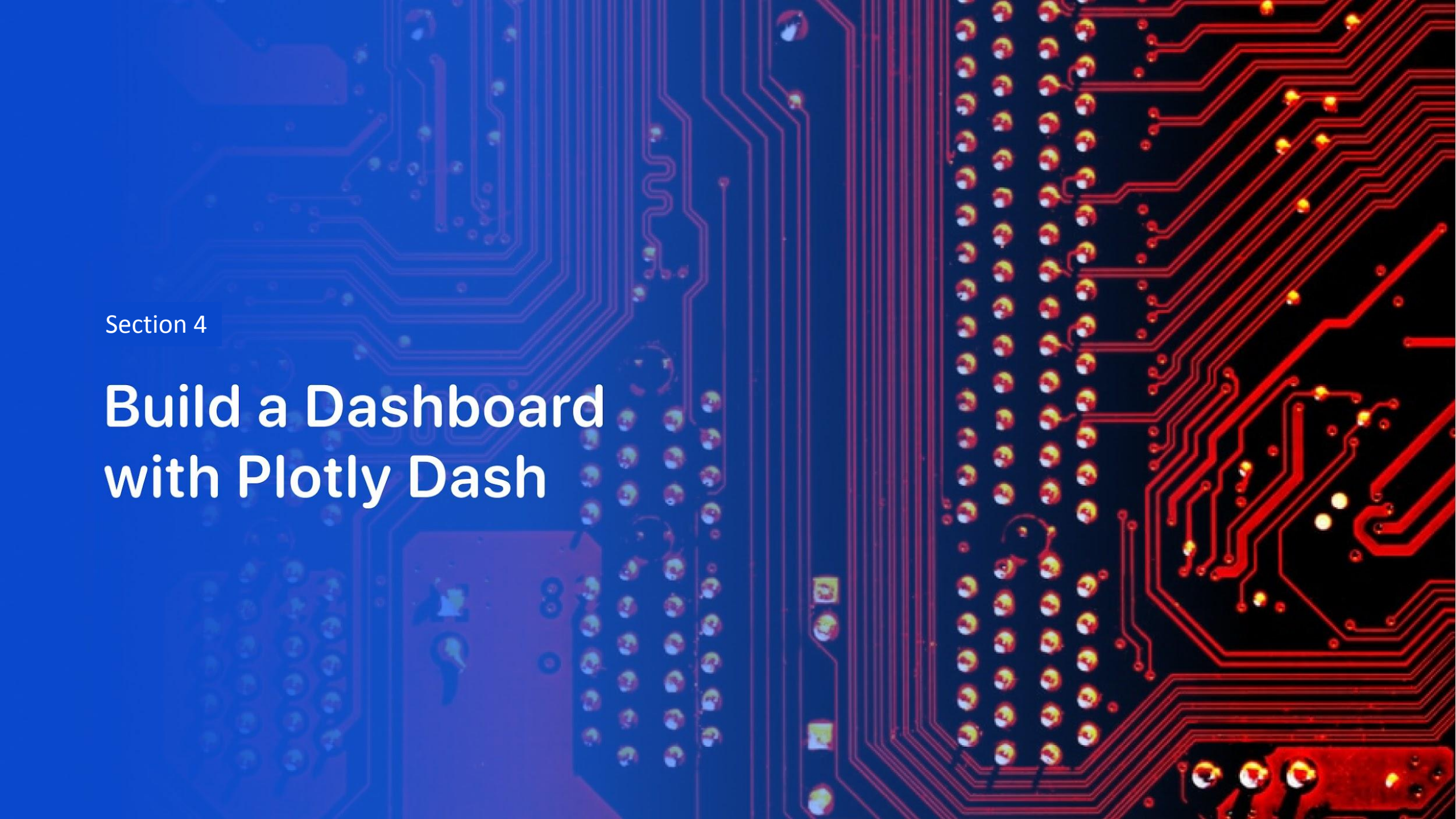


Assign a color to the launch outcomes for each site based on their success.

Distances between a launch site to its proximities



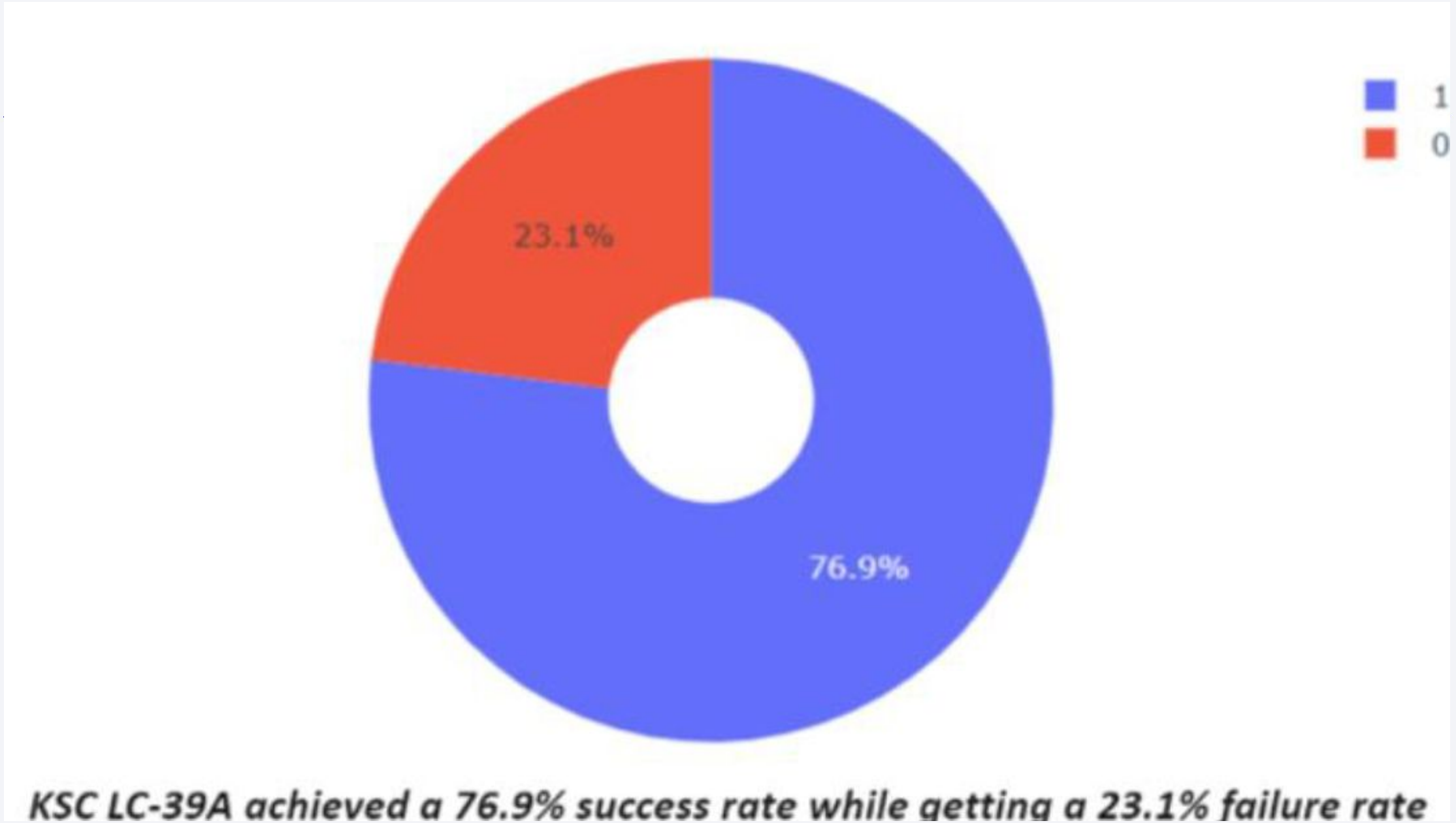
Distance by KM


Section 4
Build a Dashboard with Plotly Dash

Total Success Launches By all sites

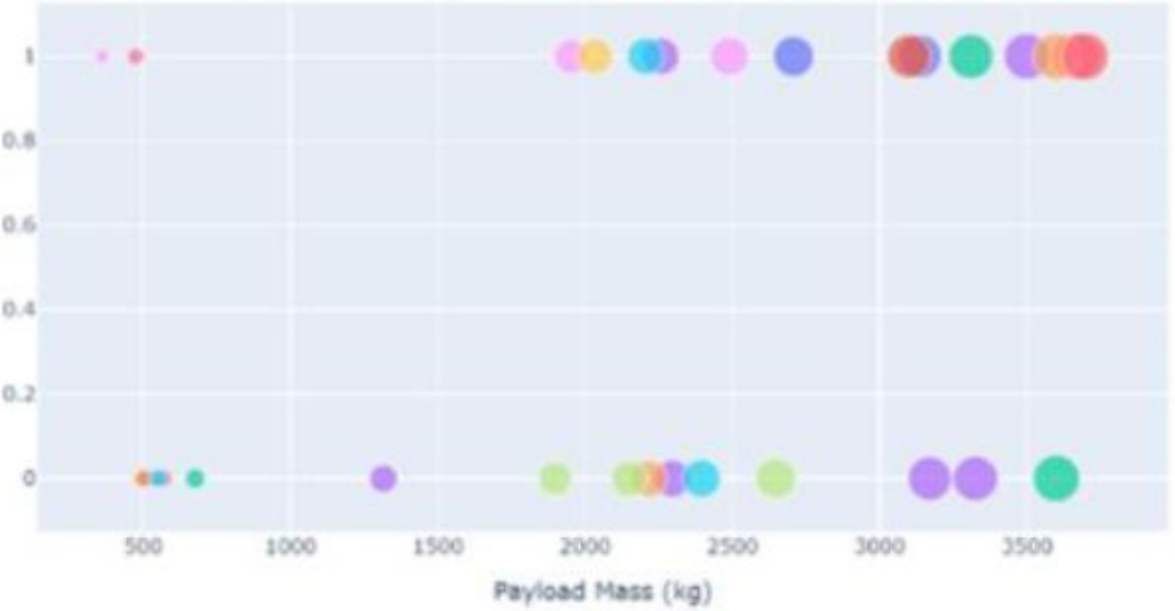
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40

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

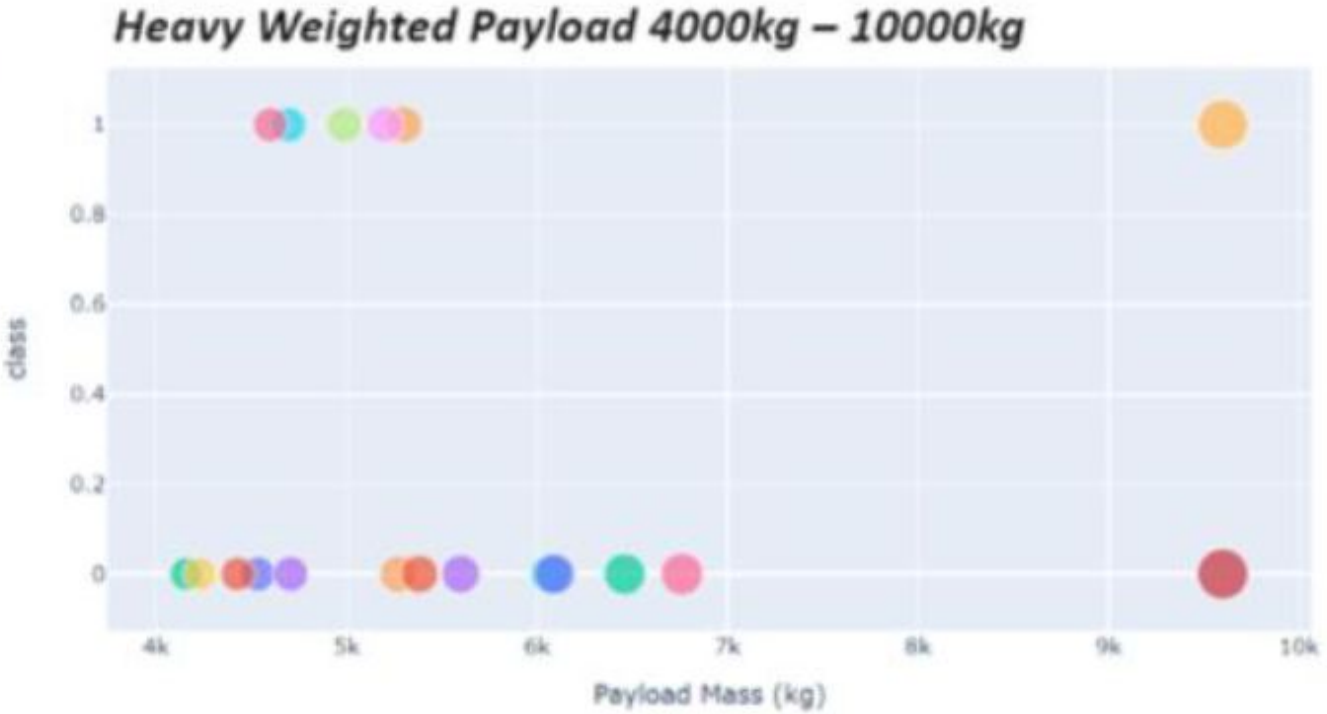


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

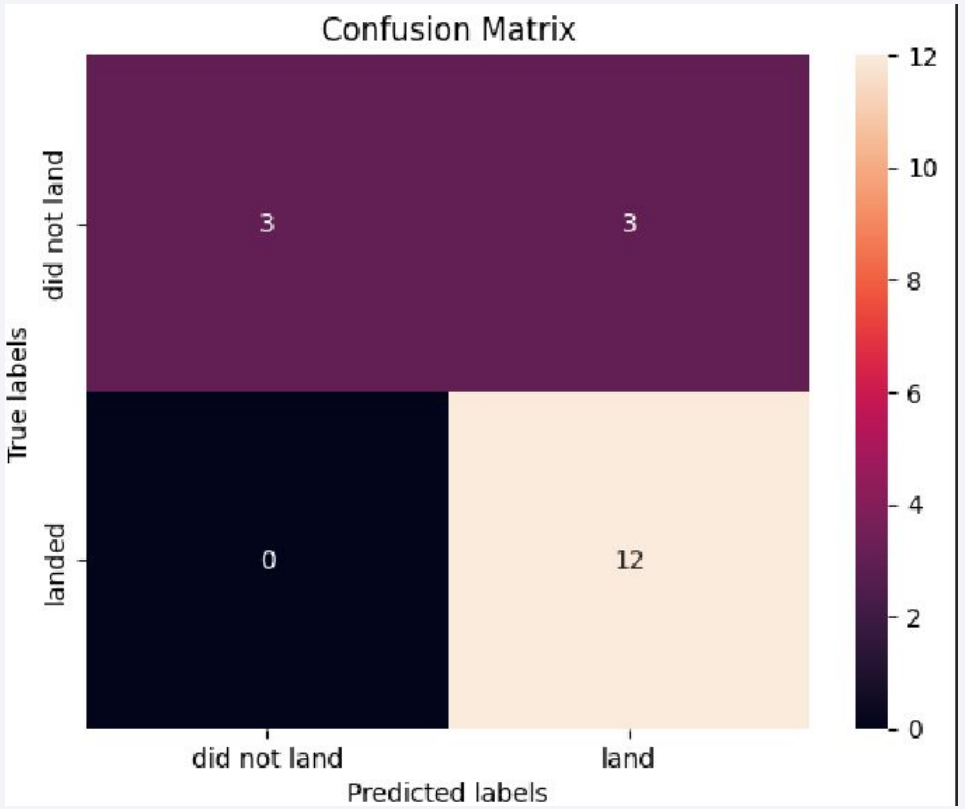
Predictive Analysis (Classification)

Classification Accuracy

Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.8875	0.83333
KNN	0.84821	0.83333

The test accuracy is similar for all models, but it's worth noting that the decision tree also exhibits the highest training accuracy.

Confusion Matrix



It is evident that the accuracy is quite high, matching 12 landings out of 12.

Conclusions

In this project, we undertook a comprehensive analysis of SpaceX's launch data with the goal of gaining insights into the company's launch history, success rates, and payload trends. Our exploration yielded several key findings:

- **Launch Site Analysis:** We discovered that Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC 40) had the highest number of launches among the considered sites.

- **Success Rate:** Launches to Geostationary Transfer Orbit (GTO), Highly Elliptical Orbit (HEO), and Sun-Synchronous Orbit (SSO) demonstrated the highest success rates, emphasizing the reliability of these missions.

- **Payload Mass Trends:** An upward trend in payload mass over the years was observed, reflecting SpaceX's evolving capabilities and growing ambitions.

- **Customer Relations**: NASA (CRS) emerged as the primary customer for SpaceX, making substantial contributions to the launched payload mass.

Appendix

Python Code Snippets: Diverse Python code snippets were utilized for tasks such as data collection, preprocessing, and analysis. These code snippets are accessible within the project's source code.

SQL Queries: SQL queries played a crucial role in retrieving and analyzing data from the database. The specific queries are detailed in the project's SQL script.

Charts and Visualizations: Visual representations, including bar charts and line charts, were generated to illustrate key findings and trends. These charts are available in the project's report.

Jupyter Notebook Outputs: A Jupyter Notebook served as a valuable tool for in-depth 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 cleaned data sets used in the project are provided for reference. These data sets can be accessed in the project's data directory.

Thank you!