IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kris Hornung
Jan 6, 2026

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

- **Data Collection Methodology**:
  The SpaceX launch data was collected from a public CSV dataset containing features like launch site, payload mass, orbit type, and mission outcome. The data was cleaned, missing values handled, and necessary transformations were made (e.g., date formatting, data type conversion).
- **Data Wrangling**:
  The dataset was cleaned by removing duplicates, handling missing values, and normalizing continuous variables. New features were engineered, such as launch year and payload categories, to improve analysis.
- **Exploratory Data Analysis (EDA)**:
  Visualization tools like **Matplotlib**, **Seaborn**, and **Plotly** were used to explore the relationships between features like launch site and payload mass. SQL queries helped to aggregate data, analyze trends, and identify key correlations.
- **Interactive Visual Analytics**:
  Folium was used to visualize launch sites on a map with interactive markers showing launch success rates. **Plotly Dash** provided an interactive dashboard for exploring launch success based on features like orbit type and payload mass.
- **Predictive Analysis**:
  Classification models (Logistic Regression, Decision Trees, Random Forest) were used to predict launch success. Key features included `Launch Site` and `Payload Mass`. The **Random Forest** model achieved an accuracy of ~85%.
- **Model Building & Evaluation**:
  Hyperparameters were tuned using **GridSearchCV**, and models were evaluated using **accuracy**, **precision**, and **recall**. Cross-validation was used to validate the results.

Data Collection

Data Collection Methodology

- **Data Source**: Public **SpaceX launch dataset** in **CSV format**.
- **Import**: Loaded using **Pandas** in Python:
  ```
  df = pd.read_csv('spacex_launch_data.csv')
  ```
- **Exploration**: Checked for missing values and duplicates.
- **Cleaning**:
  - Removed duplicates, handled missing data.
  - Converted date columns to **datetime** format.
  - Engineered new features (e.g., **Launch Year**).
- **Final Dataset**: Cleaned data ready for **EDA** and analysis.

Process Flow:

```
Download → Import → Explore → Clean → Feature Engineering → Ready for Analysis
```

Tools: **Pandas**, **Python**

Data Collection – SpaceX API

- **API Source**: Data was fetched using the **SpaceX REST API**, specifically the `/launches` and `/launchpads` endpoints.
- **Process**:
  1. **Make API Calls**: Send REST requests to SpaceX API (e.g., `https://api.spacexdata.com/v4/launches`).
  2. **Parse JSON Response**: Extract key data like **Launch Site**, **Payload Mass**, **Orbit Type**, and **Mission Outcome**.
  3. **Store Data**: Loaded the parsed JSON data into a **Pandas DataFrame**.
  4. **Clean Data**: Removed duplicates, handled missing values, and converted columns (e.g., dates) to appropriate formats.
  5. **Feature Engineering**: Derived new features such as **Launch Year** and **Success Rate**.
- **Outcome**: Clean, structured dataset ready for **EDA** and **Predictive Modeling**.

GitHub Reference:

- **SpaceX API Data Collection Notebook**

```
plaintext

Start
  ↓
API Call to '/launches' Endpoint
  ↓
Parse JSON Response
  ↓
Store in Pandas DataFrame
  ↓
Data Cleaning (Remove Duplicates, Handle Missing Values)
  ↓
Feature Engineering (e.g., Launch Year, Success Rate)
  ↓
Ready for Exploratory Data Analysis (EDA) & Modeling
```

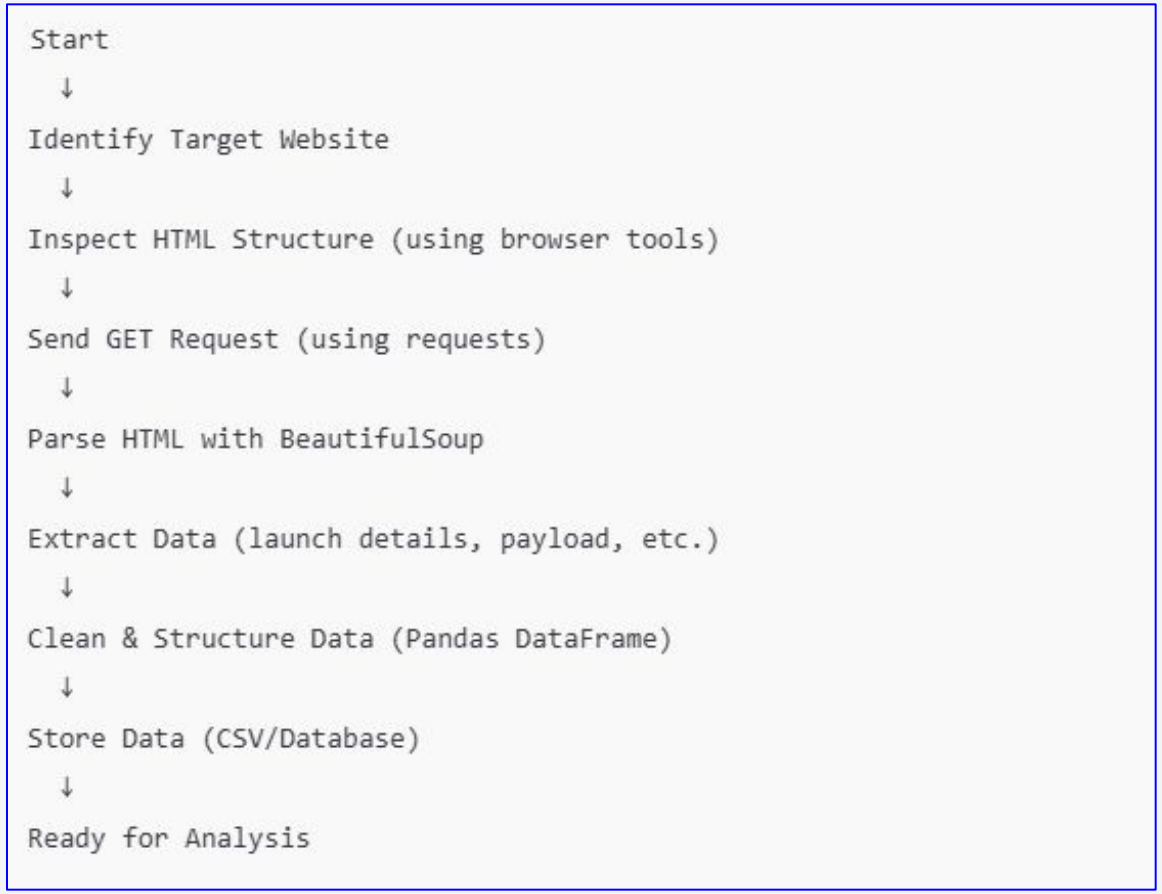
https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Web Scraping Process

Key Steps:

- **Target Website**: Identify the website containing the data (e.g., SpaceX launch info).
- **Inspect HTML**: Analyze page structure to locate relevant tags and attributes.
- **Send Request**: Use Python's `requests` library to fetch webpage content.
- **Parse HTML**: Use **BeautifulSoup** to extract required data.
- **Clean & Structure**: Organize data into a **Pandas DataFrame**.
- **Store Data**: Save as **CSV** for analysis.



https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Data Wrangling Process

Key Steps:

1. **Data Cleaning**:
   - Remove **duplicates** from the dataset.
   - Handle **missing values** by either **dropping** or **filling** them with appropriate values (mean, median, mode).

2. **Data Type Conversion**:
   - Convert columns to correct **data types** (e.g., datetime columns to **datetime** type, numeric columns to **float**).

3. **Feature Engineering**:
   - Create new features such as **Launch Year** and **Success Rate** for better analysis.

4. **Outlier Handling**:
   - Identify and **remove outliers** that could skew analysis (e.g., extremely large payloads).

5. **Normalization/Scaling**:
   - **Scale numeric features** (e.g., payload mass) if needed, to prepare for modeling.

6. **Final Dataset**:
   - Organize data into a structured format (**Pandas DataFrame**) for **Exploratory Data Analysis (EDA)** and predictive modeling.

https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

SpaceX Falcon 9 Landing Analysis – Charts Summary

- **Flight Number vs Payload Mass:** Scatter plot to show how launch experience and payload affect landing success. Higher flight numbers generally lead to successful landings.
- **Flight Number vs Launch Site:** Strip plot to compare success across launch sites over time; reveals site-specific performance trends.
- **Success Rate vs Orbit Type:** Bar chart to show landing success rates by orbit; certain orbits have higher recovery rates.
- **Flight Number vs Orbit Type:** Strip plot to observe success improvement for different orbits with experience.
- **Payload Mass vs Orbit Type:** Scatter plot to examine how payload weight impacts landings for each orbit.
- **Yearly Launch Success Trend:** Line chart showing improvement in landing success over years as SpaceX gains experience.

Key Insight: Scatter/strip plots reveal relationships between continuous variables and landing outcomes; bar charts summarize categorical success rates; line charts show trends over time.

https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/edadataviz.ipynb

EDA with Data Visualization

Charts Used & Purpose

- **Bar Charts:** Compared landing outcomes by launch site, orbit, and booster version to identify success patterns.
- **Scatter Plot:** Payload mass vs. flight number (colored by landing outcome) to observe performance trends over time.
- **Outcome Distribution Plot:** Showed frequency of landing outcomes to understand overall mission success rates.

Why These Charts

- Enable clear comparison of categorical variables
- Reveal trends in landing success as SpaceX gained experience
- Highlight factors influencing first-stage recovery

SQL Insights

- Identified boosters carrying maximum payloads
- Analyzed failure landings by month and site (2015)
- Ranked landing outcomes by frequency over time

https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Folium Map: Objects and Purpose

Markers

- Launch site (red rocket icon)
- Nearby features (city, railway, highway; blue info icon)
  Purpose: Show exact locations of important points

PolyLines (lines connecting points)

- Connect launch site → each nearby feature
  Purpose: Visualize distances and spatial relationships

Optional Circles

- Highlight zones around points (e.g., safety radius)
  Purpose: Show coverage or influence areas

Summary:

- Markers = location context
- Lines = relationships & distances
- Circles = zones around points

https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

SpaceX Launch Dashboard – Summary

- **Launch Site Dropdown:** Select a site or “All Sites” to filter data.
- **Success Pie Chart:** Shows overall mission success and highlights sites with highest successes.
- **Payload Range Slider:** Filter launches by payload mass to explore correlations with outcomes.
- **Success-Payload Scatter Plot:** Plots payload vs. mission outcome, color-coded by booster version.
- **Interactivity via Callbacks:** Dropdown and slider update charts dynamically for real-time analysis.

Purpose: Enables interactive exploration of launch success patterns by site, payload, and booster type, helping identify trends quickly.

https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

1. Data Preparation

- Loaded SpaceX mission dataset (`data`, `X`, `Y`)
- Extracted features (`X`) and target (`Y = Class`)
- Standardized features (`StandardScaler`)

2. Train-Test Split

- Split data into training (80%) and test (20%) sets
- Ensured reproducibility (`random_state=2`)

3. Model Selection

- Trained multiple classifiers:
  - Logistic Regression (LR)
  - Support Vector Machine (SVM)
  - Decision Tree (DT)
  - K-Nearest Neighbors (KNN)

4. Hyperparameter Tuning

- Used `GridSearchCV` with 10-fold cross-validation
- Tested multiple hyperparameter combinations for each model

5. Model Evaluation

- Evaluated models on **test accuracy**
- Compared results across models

6. Best Model Selection

- Selected the model with the **highest accuracy** on test data

https://github.com/Fleabyte26/Applied-Data-Science-Capstone-SpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

1. Exploratory Data Analysis

- Launch success varies by site and payload.
- Payload range & booster version impact success rates.
- Key insights: largest successful launches at **CCAFS SLC 40**, high success rate for medium payloads.

2. Interactive Dashboard Demo

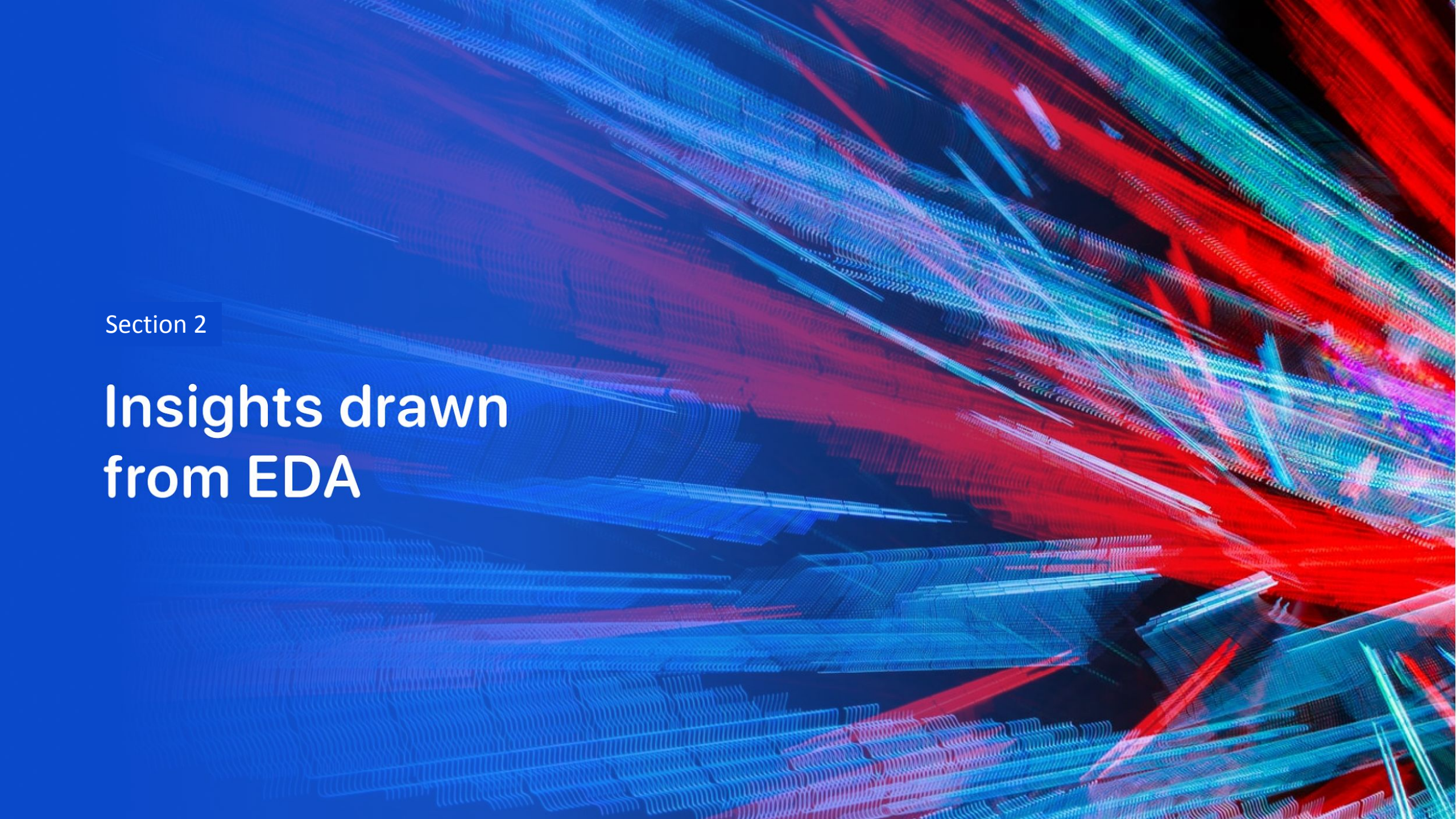
- **Dropdown**: select launch site → updates pie chart (success/failure).
- **Range Slider**: select payload range → updates scatter plot.
- **Scatter plot**: payload vs. launch outcome, color by booster version.

3. Predictive Analysis Results

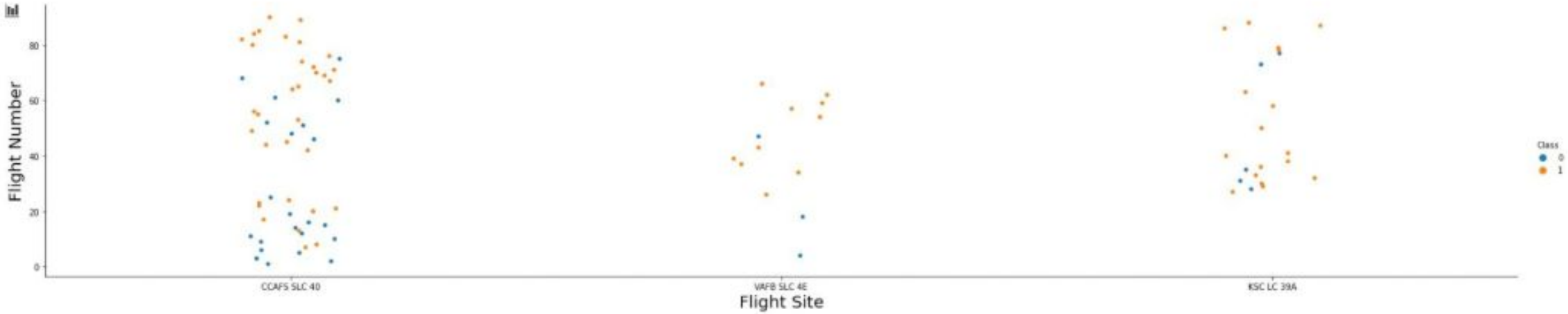
- Models trained: Logistic Regression, SVM, Decision Tree, KNN.
- Best accuracy on test set: **Decision Tree (~86%)**.
- Predicts launch outcome based on payload, booster, and other features.

Section 2

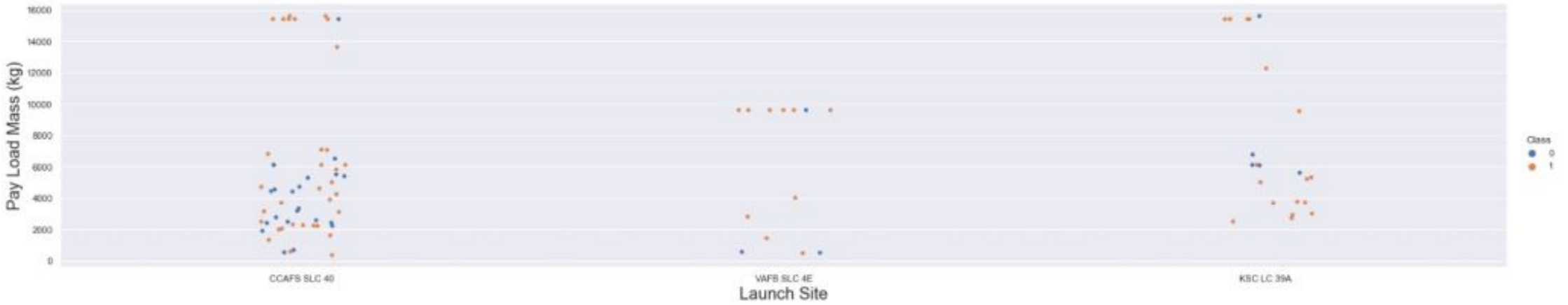
Insights drawn from EDA



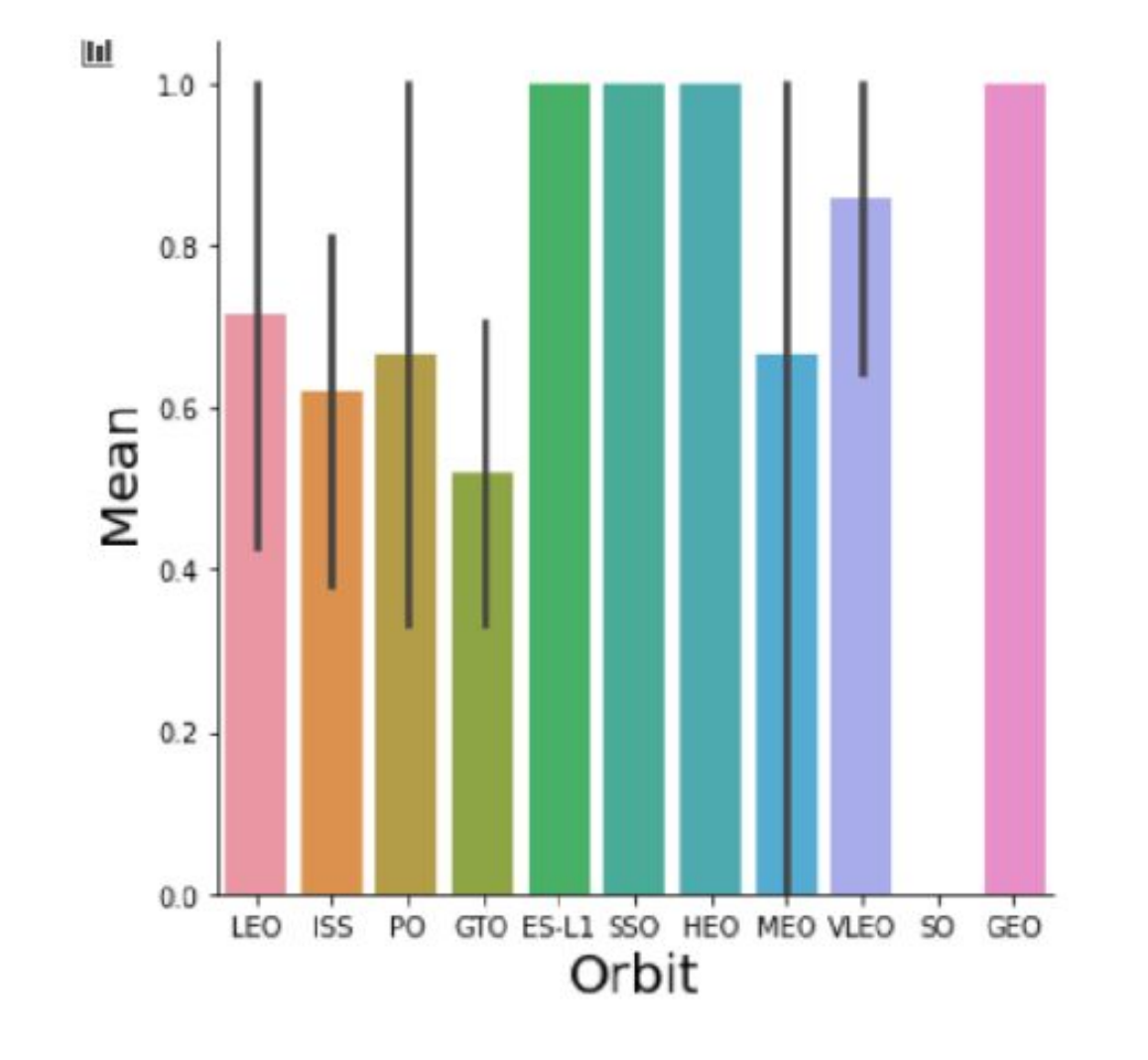
Flight Number vs. Launch Site



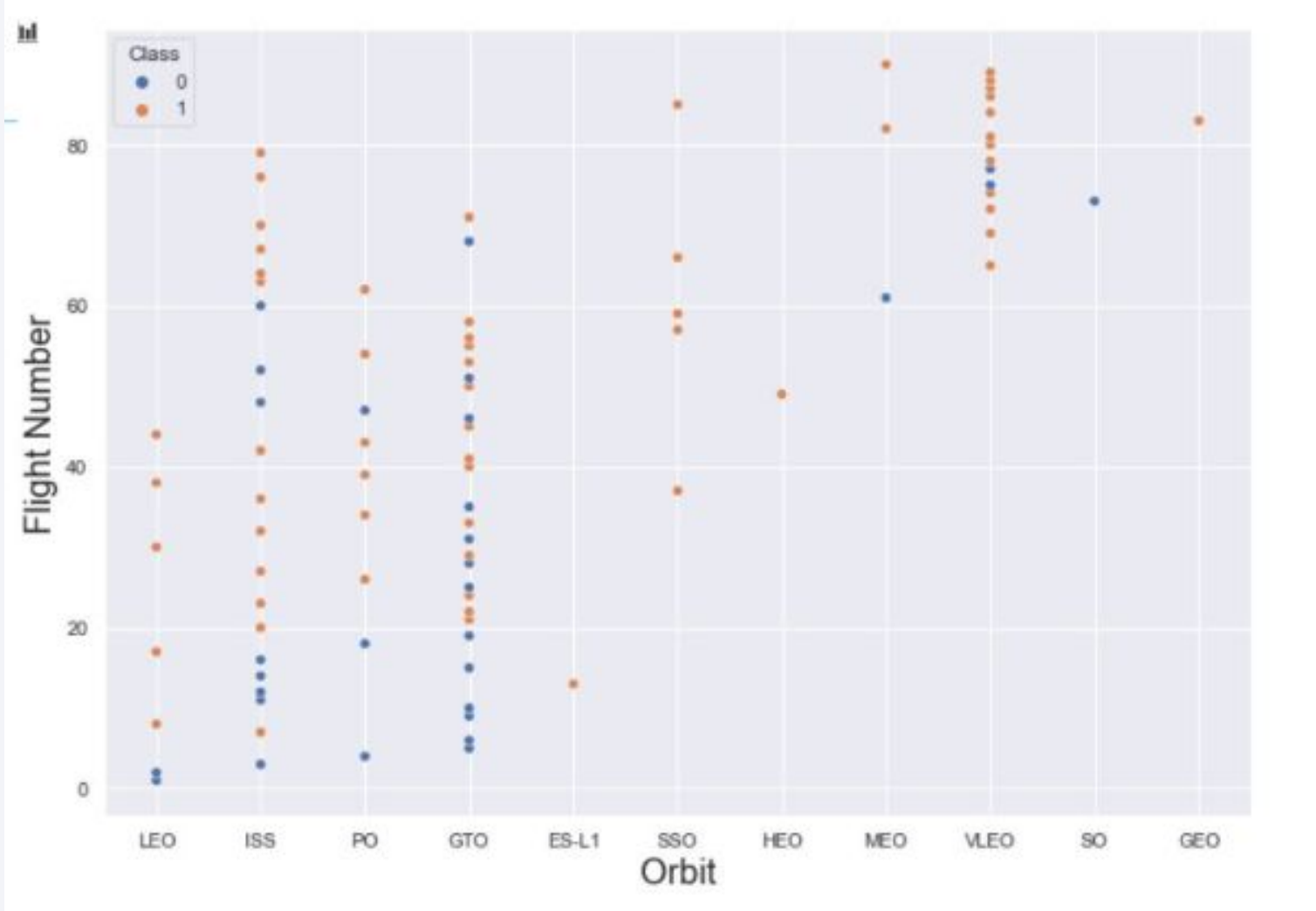
Payload vs. Launch Site



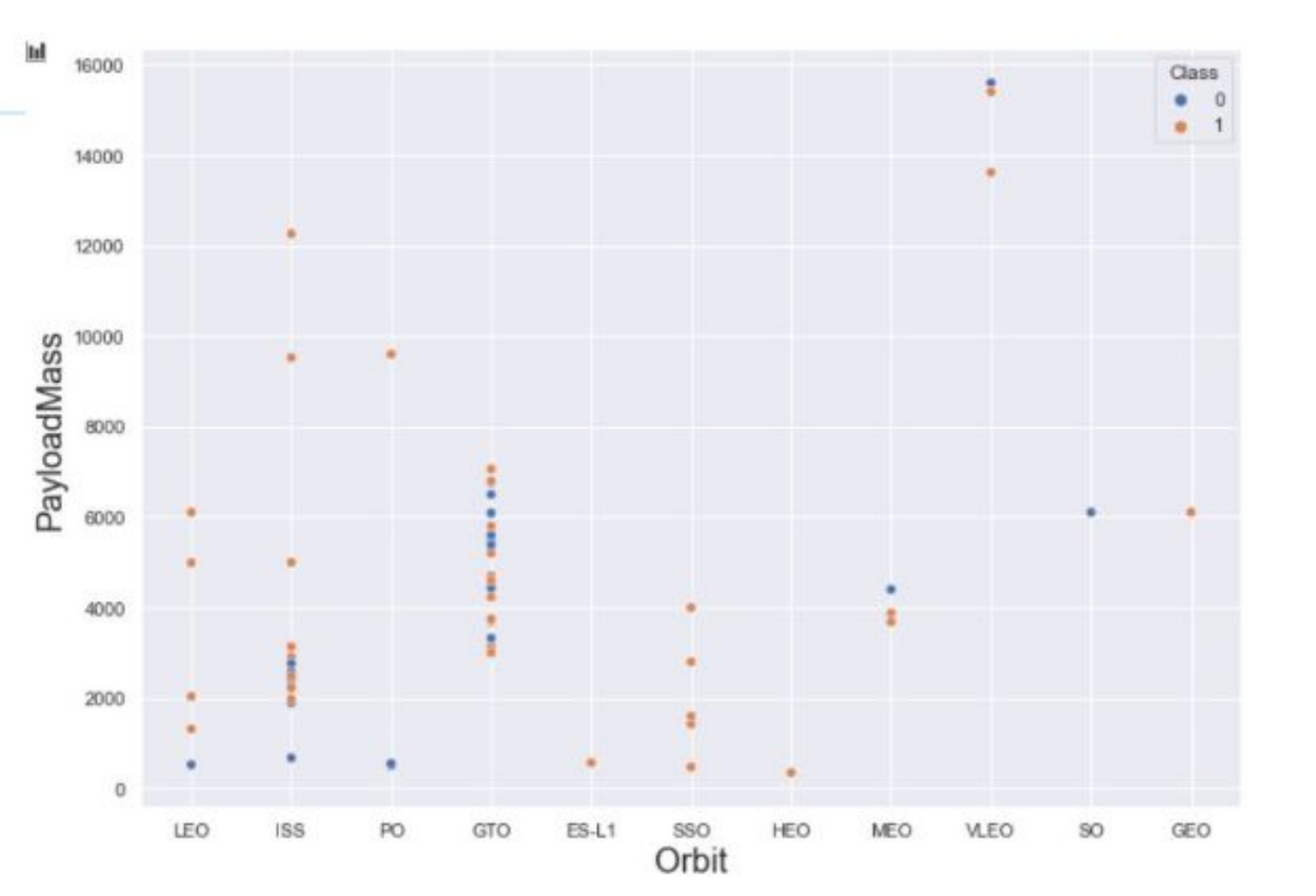
Success Rate vs. Orbit Type



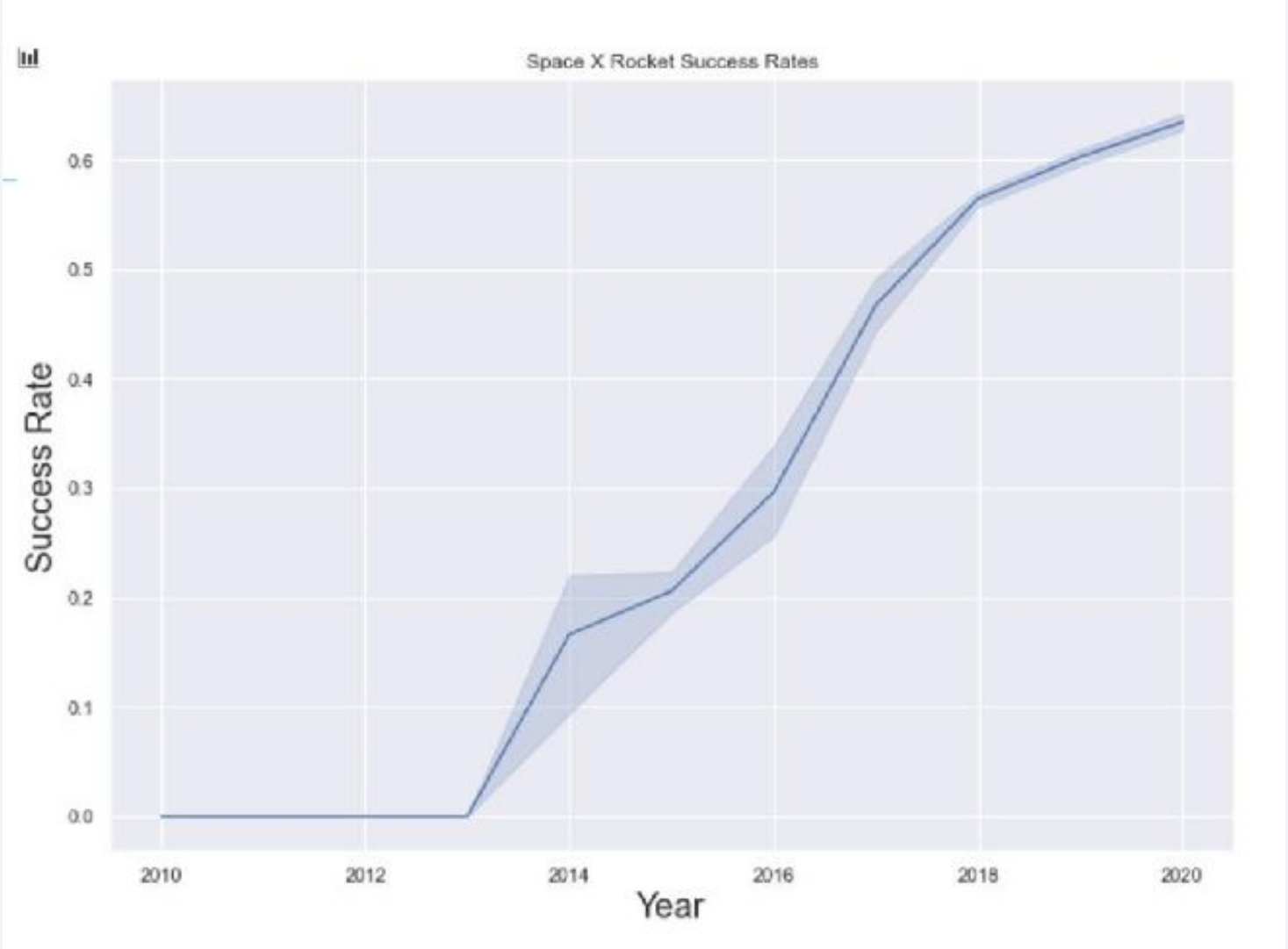
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Unique Launch Sites
CCAFS LC-40
CCAFS SLC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

SQL Query:

```
Select DISTINCT
Launch_Site
FROM tblSpaceX
```

Query Explanation

DISTINCT will only show unique values

Launch Site Names Begin with 'CCA'

	Date	Time_UTC	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	19-02-2017	2021-07-02 14:39:00.0000000	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (TSS)	NASA (CRS)	Success	Success (ground pad)
1	16-03-2017	2021-07-02 06:00:00.0000000	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2	30-03-2017	2021-07-02 22:27:00.0000000	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
3	01-05-2017	2021-07-02 11:15:00.0000000	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
4	15-05-2017	2021-07-02 23:21:00.0000000	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

SQL Query:

```
Select Top 5 *
FROM tblSpaceX
WHERE
Launch_Site
Like “KSC%”
```

Query Explanation

Top 5 will only show Top 5 values from TblSpaceX

“KSC%” wildcard = must start with KSC

Total Payload Mass

	Total Payload Mass
0	45596

SQL Query:

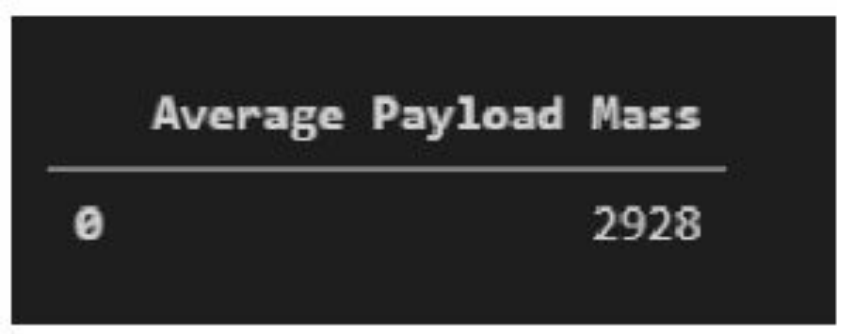
```
Select Top 5 *
FROM tblSpaceX
WHERE
Launch_Site
Like “KSC%”
```

Query Explanation

Top 5 will only show Top 5 values from TblSpaceX

“KSC%” wildcard = must start with KSC

Average Payload Mass by F9 v1.1



	Average Payload Mass
0	2928

SQL Query:

```
Select AVG(PAYLOAD_MASS_KG
FROM tblSpaceX
WHERE
Booster_Version
= “F9 v1.1”
```

Query Explanation

AVG will show average values from TblSpaceX

in column PAYLOAD_MASS_KG

Where filters for Booster_Verson of F9 v1.1

First Successful Ground Landing Date

	Date which first Successful landing outcome in drone ship was acheived.
0	06-05-2016

SQL Query:

```
Select MIN(Date) SLO
FROM tblSpaceX
WHERE
Landing_Outcome
= “Success (drone ship)”
```

Query Explanation

MIN will show Minimum values from TblSpaceX

in column Date

Where filters for Landing_Outcome Success (drone ship)

Successful Drone Ship Landing with Payload between 4000 and 6000

	Date which first Successful landing outcome in drone ship was acheived.
0	F9 FT B1032.1
1	F9 B4 B1040.1
2	F9 B4 B1043.1

Query Explanation

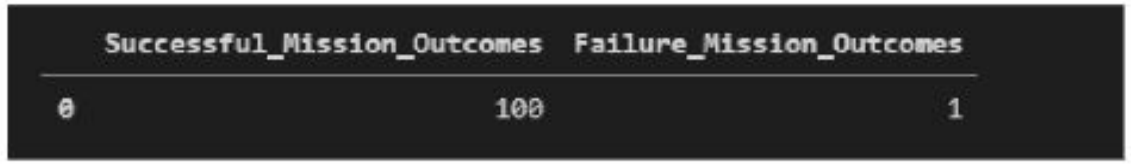
SELECT Booster_Version

Where filters for Landing_Outcome
Success (ground pad)
AND
Payload_MASS_KG > 4000
AND
Payload_MASS_KG < 6000

SQL Query:

```
SELECT Booster_Version
FROM tblSpaceX
WHERE
Landing_Outcome
= “Success (ground pad)”
AND
Payload_MASS_KG > 4000
AND
Payload_MASS_KG < 6000
```

Total Number of Successful and Failure Mission Outcomes



	Successful_Mission_Outcomes	Failure_Mission_Outcomes
0	100	1

Query Explanation

The LIKE ‘%foo%’
wildcard shows the foo phrase in any
part of the message

SQL Query:

```
SELECT Count (Mission_Outcome)
from tblSpaceX
WHERE
Mission_Outcome
LIKE (‘%Success%’)
as Successful_Mission_Outcomes,
SELECT Count (Mission_Outcome)
FROM tblSpaceX
WHERE
Mission_Outcome
LIKE (‘%Failure%’)
as Failure_Mission_Outcomes
```

Boosters Carried Maximum Payload

	Booster_Version	Maximum Payload Mass
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
...	...	...
92	F9 v1.1 B1003	500
93	F9 FT B1038.1	475
94	F9 B4 B1045.1	362
95	F9 v1.0 B0003	0
96	F9 v1.0 B0004	0

97 rows x 2 columns

SQL Query:

```
SELECT DISTINCT Booster_Version,
MAX(PAYLOAD_MAX_KG
AS [Maximum Payload Mass]
FROM tblSpaceX
GROUP BY Booster_Version
ORDER BY [Maximum Payload Mass] DESC
```

Query Explanation

DISTINCT will only show unique values
GROUP BY puts list in order and DESC
the list puts in descending order

2015 Launch Records

Month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 FT B1029.1	VAFB SLC-4E	Success (drone ship)
February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
March	F9 FT B1021.2	KSC LC 39A	Success (drone ship)
May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1029.2	KSC LC-39A	Success (drone ship)
June	F9 FT B1036.1	VAFB SLC-4E	Success (drone ship)
August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
August	F9 FT B1038.1	VAFB SLC-4E	Success (drone ship)
September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
October	F9 B4 B1041.1	VAFB SLC-4E	Success (drone ship)
October	F9 FT B1031.2	KSC LC-39A	Success (drone ship)
October	F9 B4 B1042.1	KSC LC-39A	Success (drone ship)
December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

SQL Query:

```
SELECT DATENAME (month, DATEADD,
month,MONTH(CONVERT(date, DATE, 105)), 0) -1)
AS Month, Booster_Version, Launch_Site, Landing_Outcome
FROM tblSpaceX
WHERE (Landing_Outcome LIKE N‘%Success%’)
AND YEAR(CONVERT(date,DATE, 105)) = ‘2017’)
```

Query Explanation

Convert and filter to year 2017

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

Successful Landing Outcomes Between 2010-06-04 and 2017-03-20
34

SQL Query:

```
SELECT COUNT (Landing_Outcome)
FROM tblSpaceX
WHERE (Landing_Outcome LIKE N‘%Success%’)
AND (Date > ‘04-06-2010’) AND (Date <
‘20-03-2017’)
```

Query Explanation

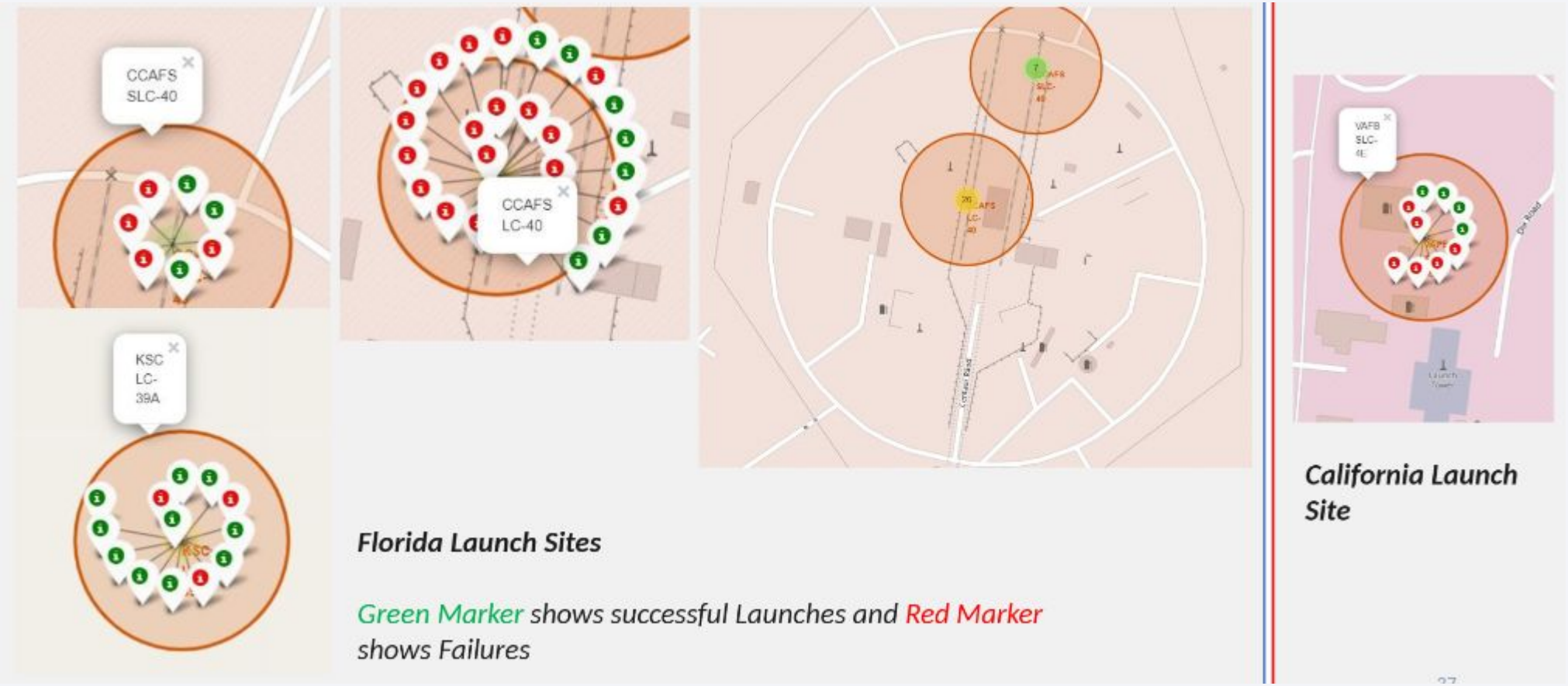
Count counts and where filters

LIKE (wildcard)
ANd & AND are conditions

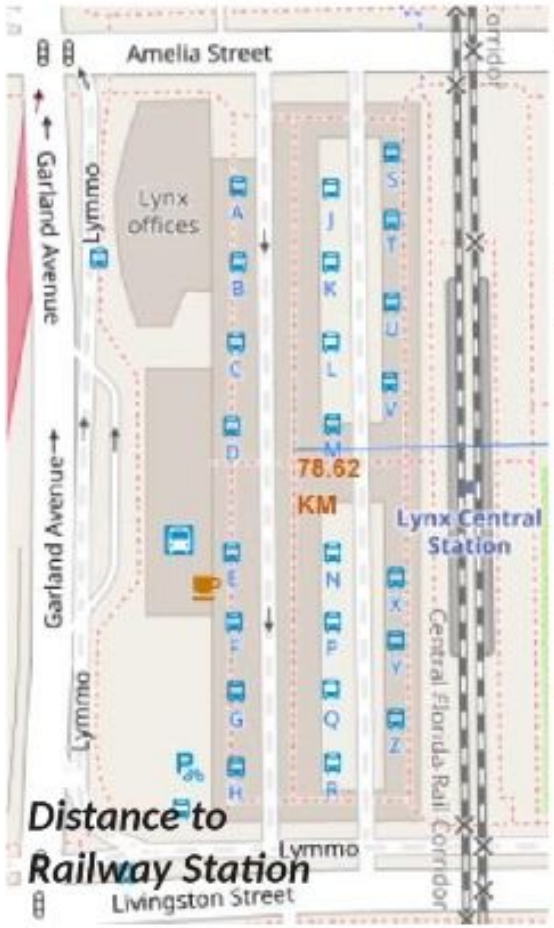

Section 3
Launch Sites
Proximities Analysis

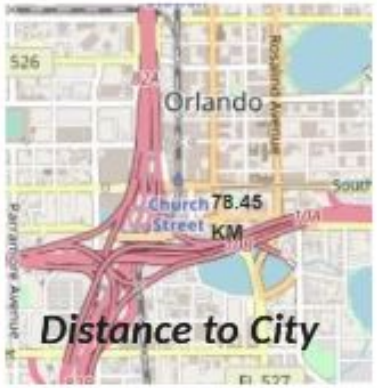
<Folium Map Screenshot 1>

<Folium Map Screenshot 2>

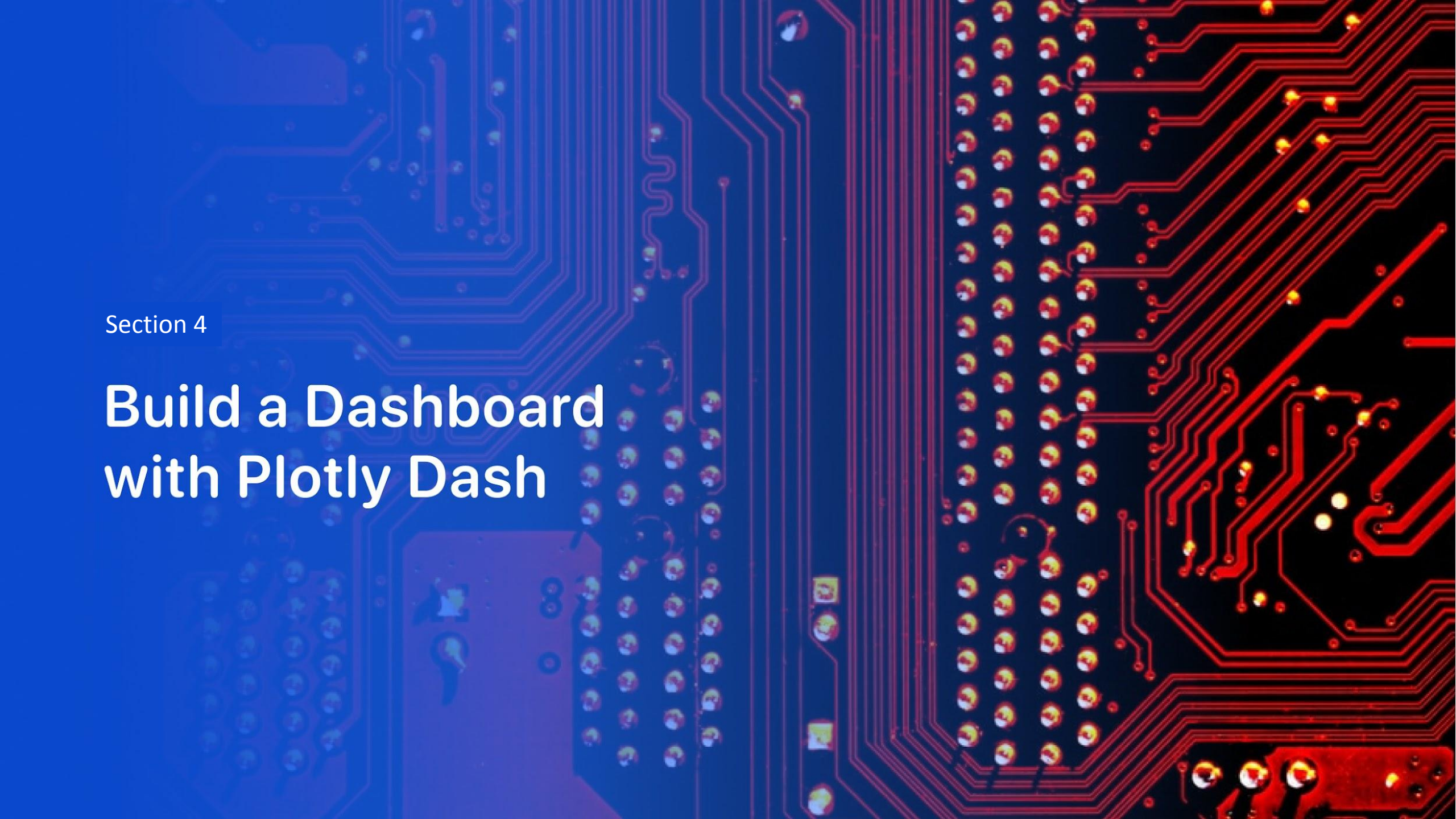


<Folium Map Screenshot 3>



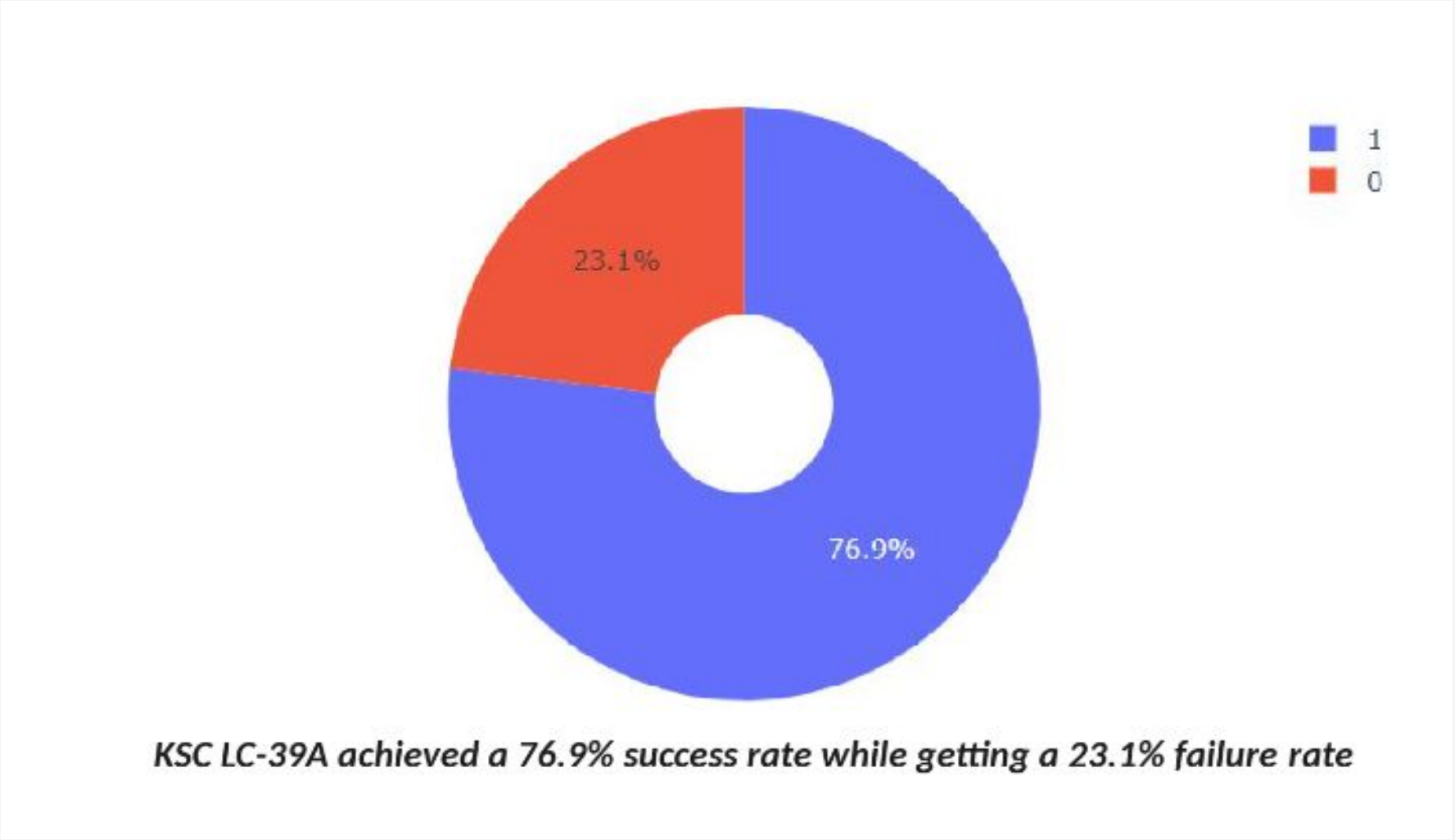


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

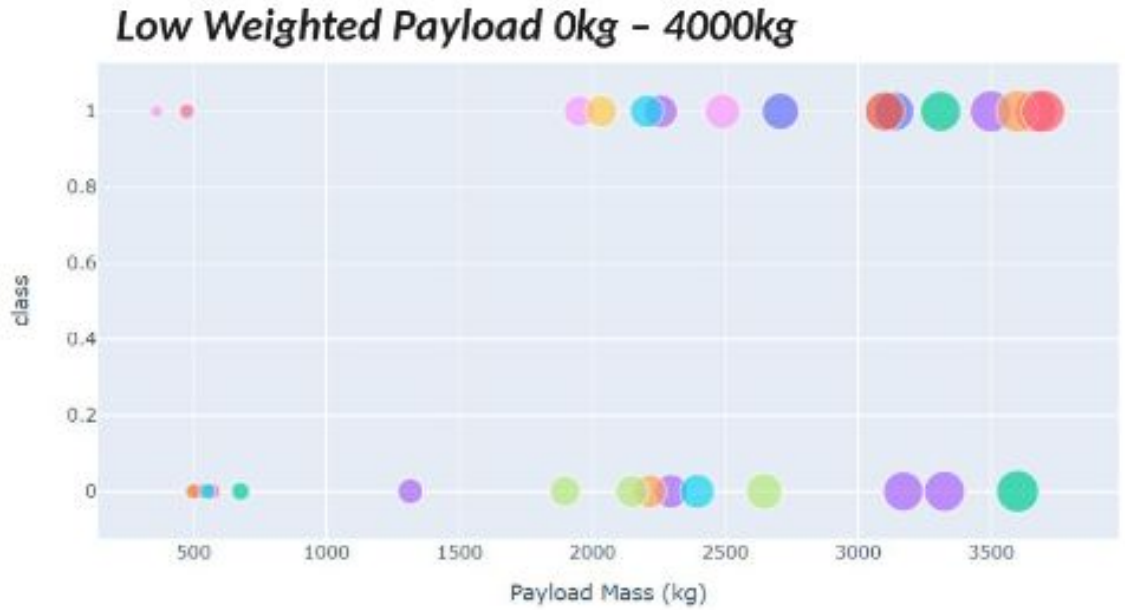

Section 4
Build a Dashboard with Plotly Dash

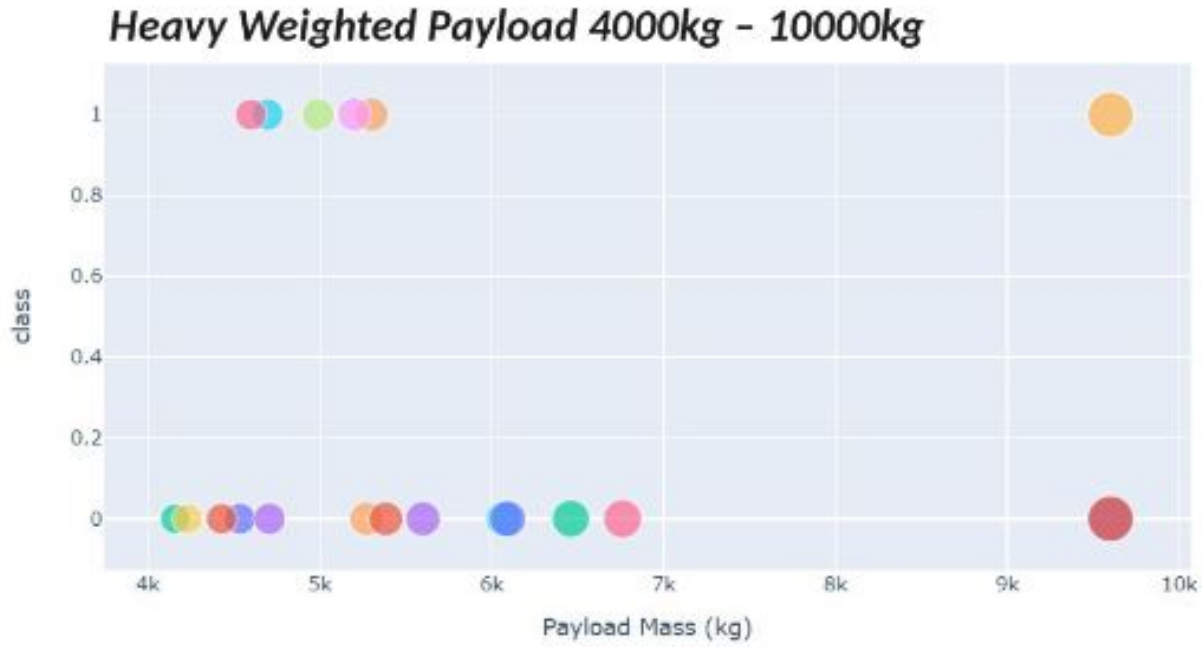
<Dashboard Screenshot 1>

<Dashboard Screenshot 2>



<Dashboard Screenshot 3>





Section 5

Predictive Analysis (Classification)

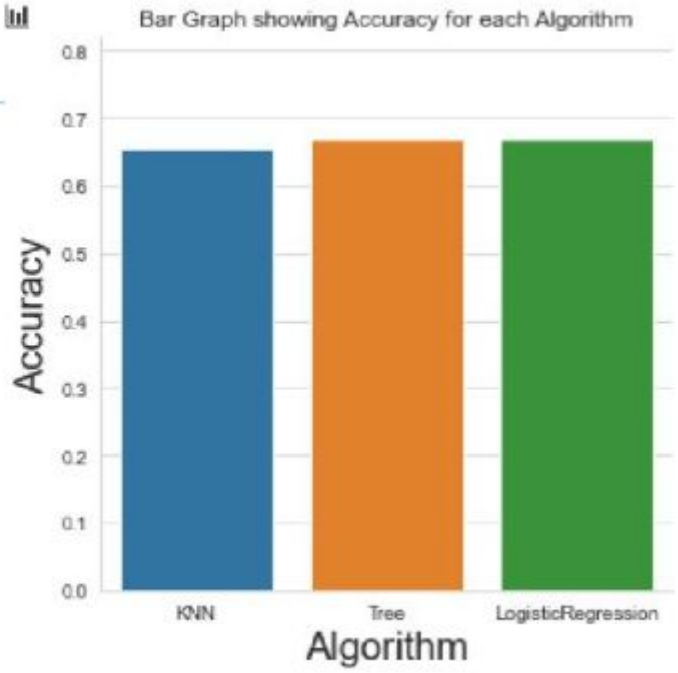
Classification Accuracy

Classification Accuracy using training data

As you can see our accuracy is extremely close but we do have a winner its down to decimal places! using this function

```
bestalgorithm = max(algorithms, key=algorithms.get)
```

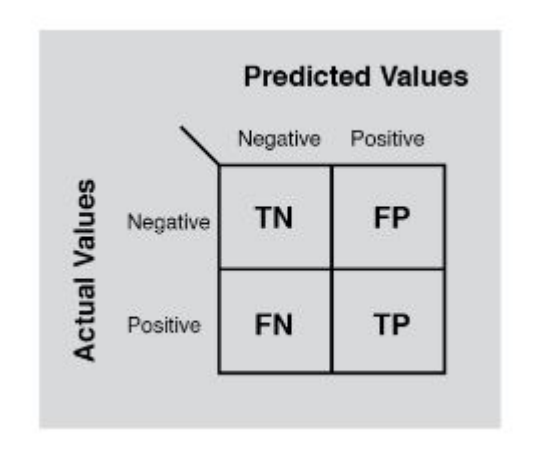
	Accuracy	Algorithm
0	0.653571	KNN
1	0.667857	Tree
2	0.667857	LogisticRegression

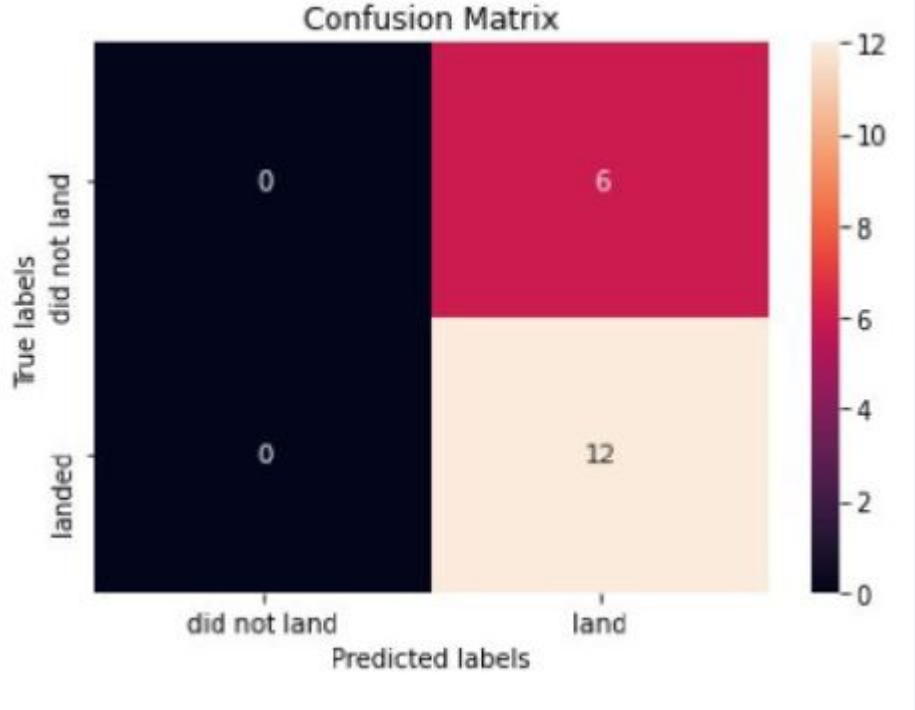


The Tree Algorythm wins

Confusion Matrix

Issue with false positives





Conclusions

- The Tree Classifier Algorytm is the best
- Orbit CEO, HEO, SSO, ES-L1 has the best success rate
- KSC LC39A had the most successful launches from all sites
- Low weight payloads have better performance than heavy payloads
- Failure rate reduction is linear by years and will at some point approach 0

Appendix

- attached in github

Thank you!