IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hargobind S. Gill
Saturday, July 20th, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via APIs & Web Scraping
  - Data Wrangling Techniques
  - Exploratory Data Analysis with SQL & Visualization Tools
  - Interactive Visual Analytics w/ Folium
  - Interactive Dashboard w/ Plotly Dash
  - Machine Learning Predictions (KNN, Decision Trees, SVM)

- Summary of all results
  - *Success Rate Over Time:* Significant Improvement over Defined Periods
  - *Success Rate by Launch Site:* The highest success rate of all launch sites was **KSC LC-39A**.
  - *Success Rate by Orbit:* The highest success rates observed were amongst **ES-L1, SSO, HEO, & GEO**.
  - *Predictive Algorithms:* The **DecisionTreeClassifier** was most accuracte in the prediction of landing outcomes.
  - *Payload Analysis:* It is seen that heavier payloads have higher failure rates, though success of which improves over time.

Introduction

Project background and context

SpaceX Falcon 9 Rocket Launches => $62mm

Other Providers Cost => $165mm

The nearly $100mm in savings is due to SpaceX's proprietary rocket reuse technology post-first stage launch.

Problems you want to find answers

Will SpaceX's Falcon 9 Rocket land successfully?

What is the historical success rate of these past launches?

What metrics may be utilized in prediction of the probability of success for future launches?

Section 1

Methodology

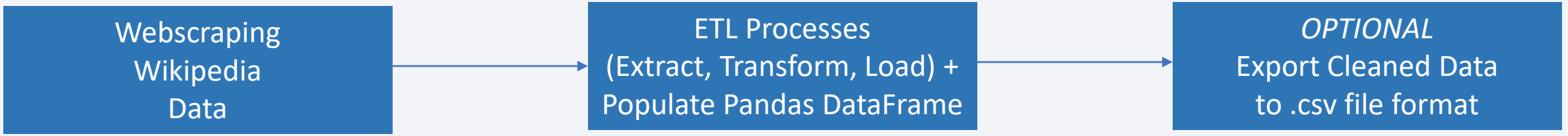
Methodology

Executive Summary

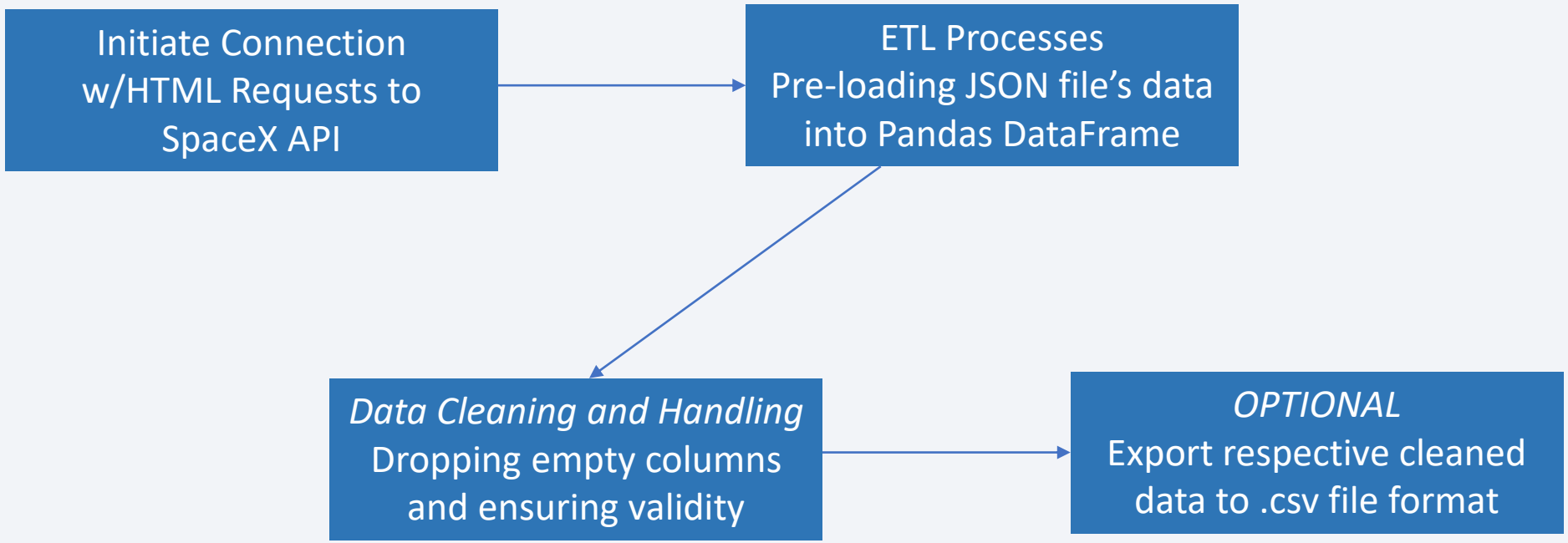
- Data collection methodology:
  - Collection was constructed through SpaceX API endpoints and webscraping via Wikipedia
- Perform data wrangling
  - Performed ETL of SpaceX data, respectively tuning appropriate records & dropping nulls
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Illustrative analysis presented through Pie, Scatter, Bar, and other charts
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning, and evaluating classification models utilizing Ensemble Tree Methods and Regression Analyses

Data Collection

- Data Collection:
  - HTTP Requests pivoting with SpaceX API Endpoints, to extract the respective raw data
  - Utilized tabular data from webscraping applications on the Wikipedia.org site

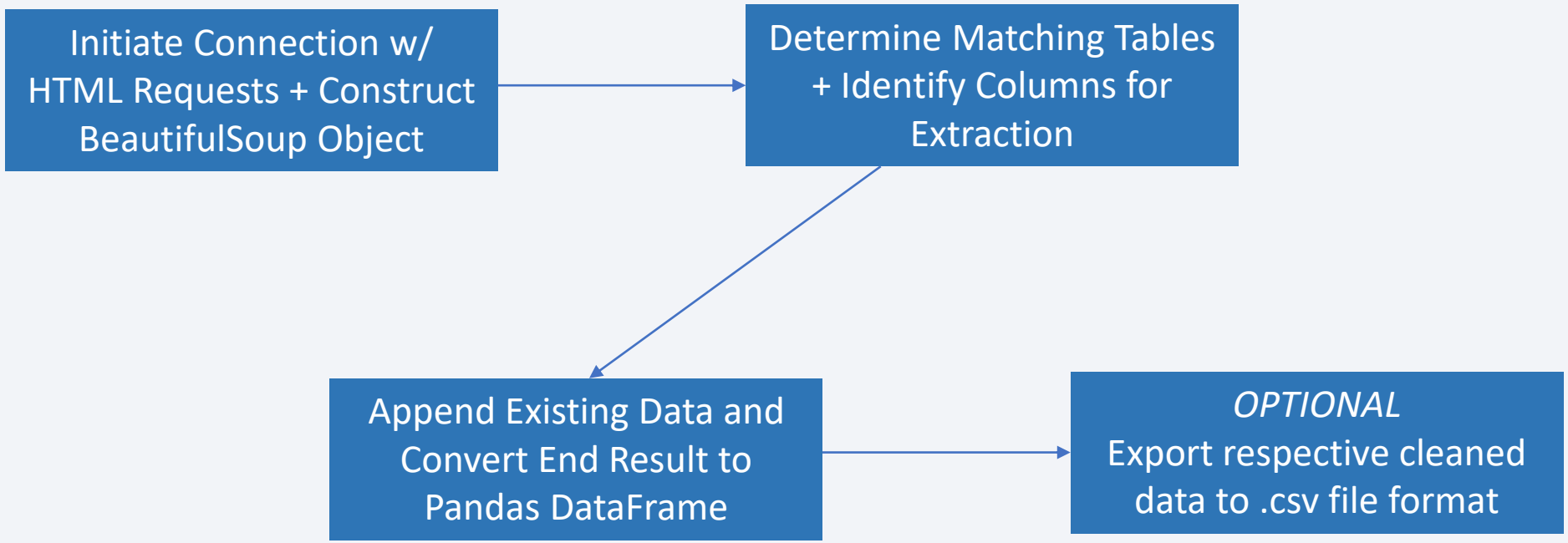


Data Collection – SpaceX API



GitHub Link

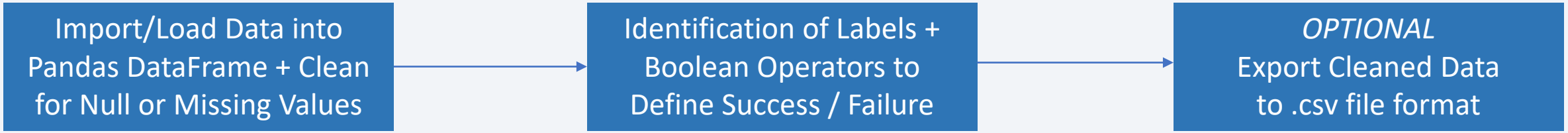
Data Collection - Scraping



GitHub Link

Data Wrangling

- Describe how data were processed
  - Utilizing Exploratory Data Analysis (EDA) toolkits in order to readily identify pattern sequences in data structures.



GitHub Link

EDA with Data Visualization

The following charts were plotted in accordance with the Jupyter Assignment

- Flight Number v. Outcomes
- Flight Number v. Launch Site
- Payload v. Launch Site
- Payload v. Orbit
- Orbit v. Outcome
- Flight Number v. Orbit
- As well as, the Year-on-Year Success Rate v. Orbit

- These charts were provisioned such as to analyze dependency arguments, impacts of variables on one another, as well as trend determination on a YoY basis.

GitHub Link

EDA with SQL

- Individuals Steps Involved:
  - Firstly, one is surveying the database in order to draw conclusions upon any patterns that exist.

  - (i) Loading + Pre-Processing of Data
  - (ii) DataFrame Manipulation – Display of entries with names beginning in 'CCA'
  - (iii) Performing Arithmetic Operations Upon Specified Date Bounds (Counts + Rank b/w certain years)
  - (iv) Illustrating Payload Carry on part of weightage >=4,000 kg and <=6,000 kg (Booster Versions List)
  - (v) Formulating Counts of Successful and Failed Operations
  - (vi) Decree booster versions which passed this requisite marking (such as Year = 2015)
  - (vii) Displaying outcome distributions between the Year and Date Bounds

GitHub Link

Build an Interactive Map with Folium

Folium Map Overview & Contextual Framework:

I. Determination of geographical markedness within data such as to note pattern recognition b/w launch sites

II. Launch Site Testing and Folium Markers were illustrated as such:

1) Principally constructing a holistic map encompassing all launch sites within the United States,
2) Pooling Successful and Failed Operations on part of Launches between the aforementioned sites
3) Displaying Distance Markers to Identify Localized Structures for Launch Sites and Proximity to Landmarks

GitHub Link

Build a Dashboard with Plotly Dash

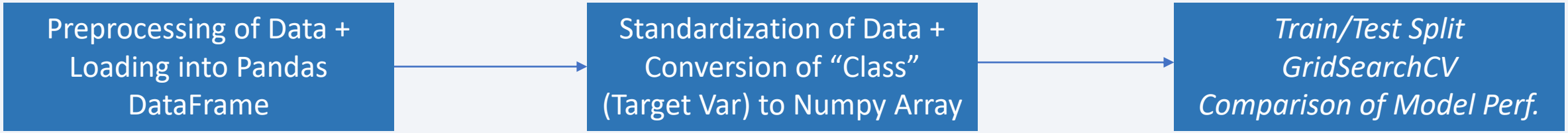
- An Overview of Plotly Dash:
  - Selection for Launch Sites
  - Relational Percentage Pie Chart – Launch Sites
  - Correlation Scatterplots – Payload & Success v. Launch Site

In basic terms, this is to provide an illustrative and interactive framework within which one can analyze the mass of the payload, success of the launch, and launch site control features as it interrelates to these categories and between launch sites.

GitHub Link

Predictive Analysis (Classification)

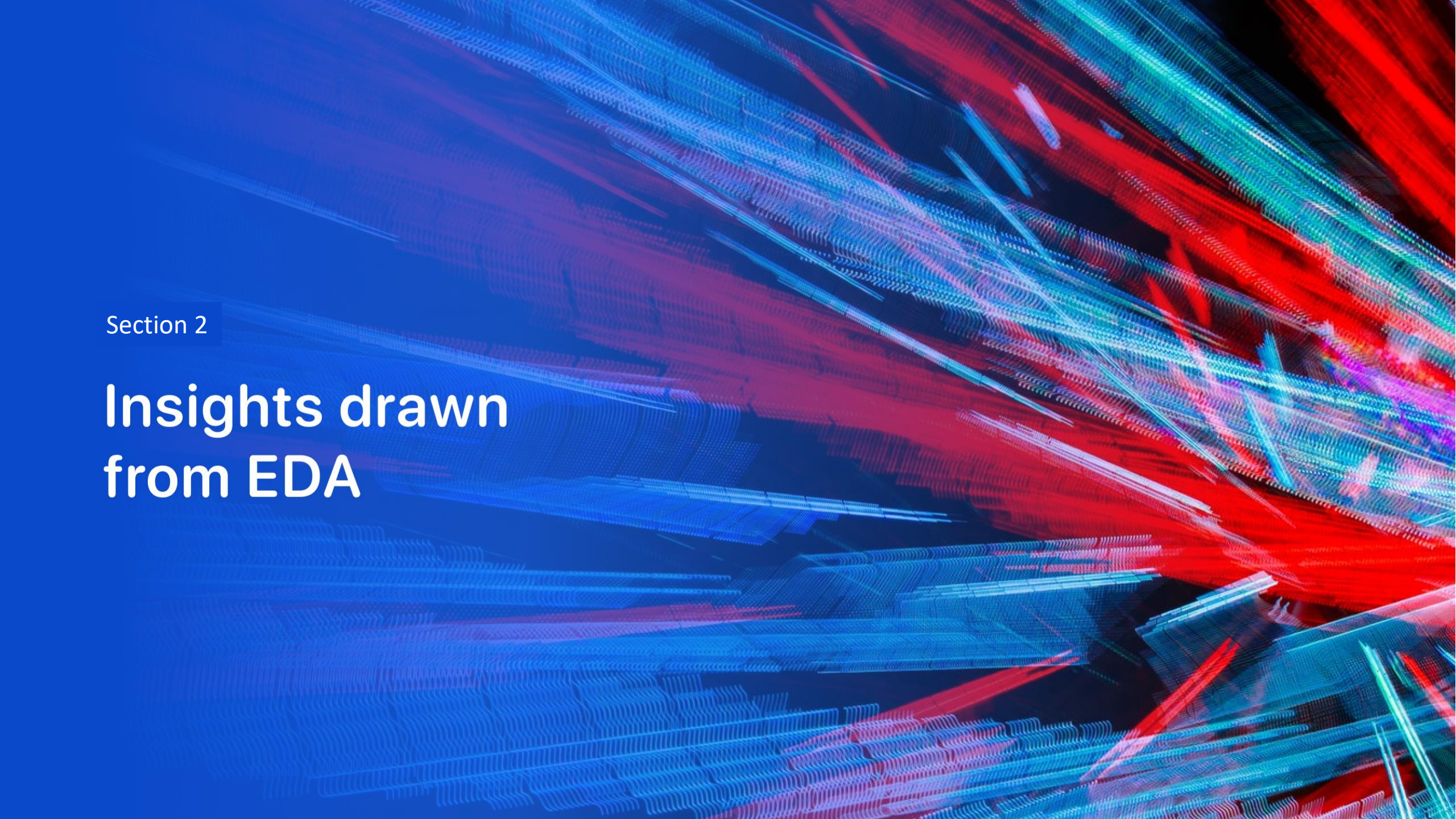
- Steps Involved:
  - Preprocessing of Data and Loading into Pandas DataFrame for manipulation purposes (ETL)
  - StandardScaler Application on X and Conversion of Y (Target Variable = “Class”) to Numpy Array
  - Splitting of Training and Testing Data using Train/Test Split, then utilizing GridSearchCV in order to optimize hyperparameters for model performance
  - Algorithms utilized in this case include:
    - Logistic Regression, SVC/SVM, Decision Tree (Classifier), K-Neighbors (Classifier)



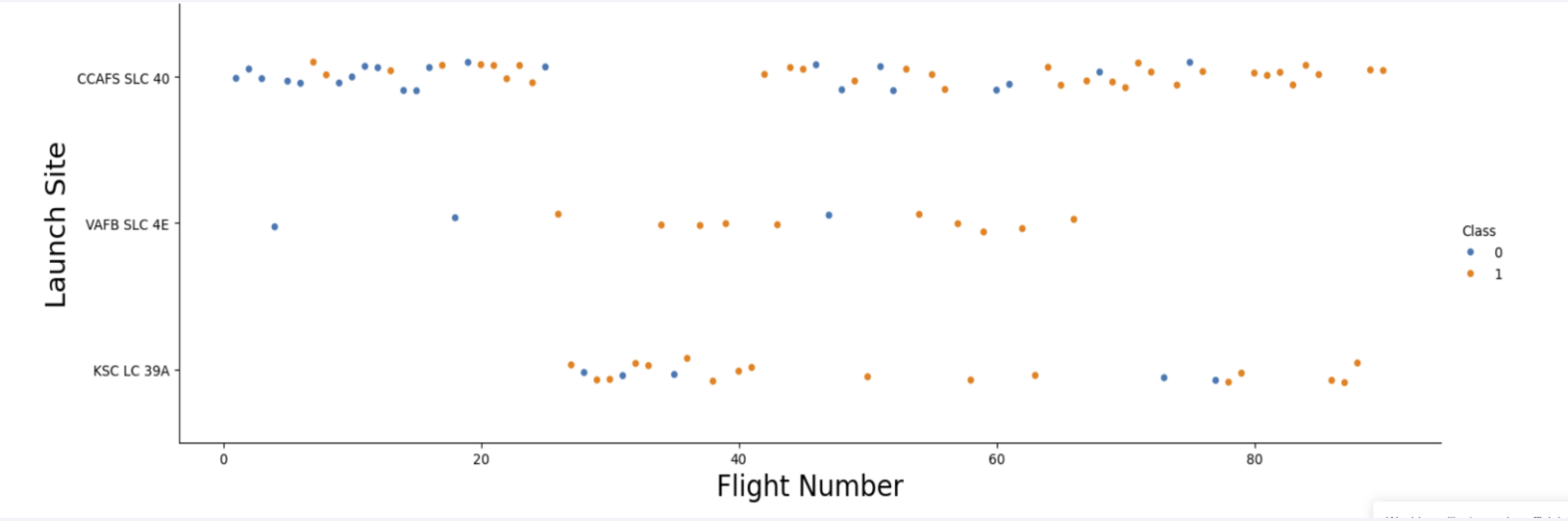
GitHub Link

Results

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

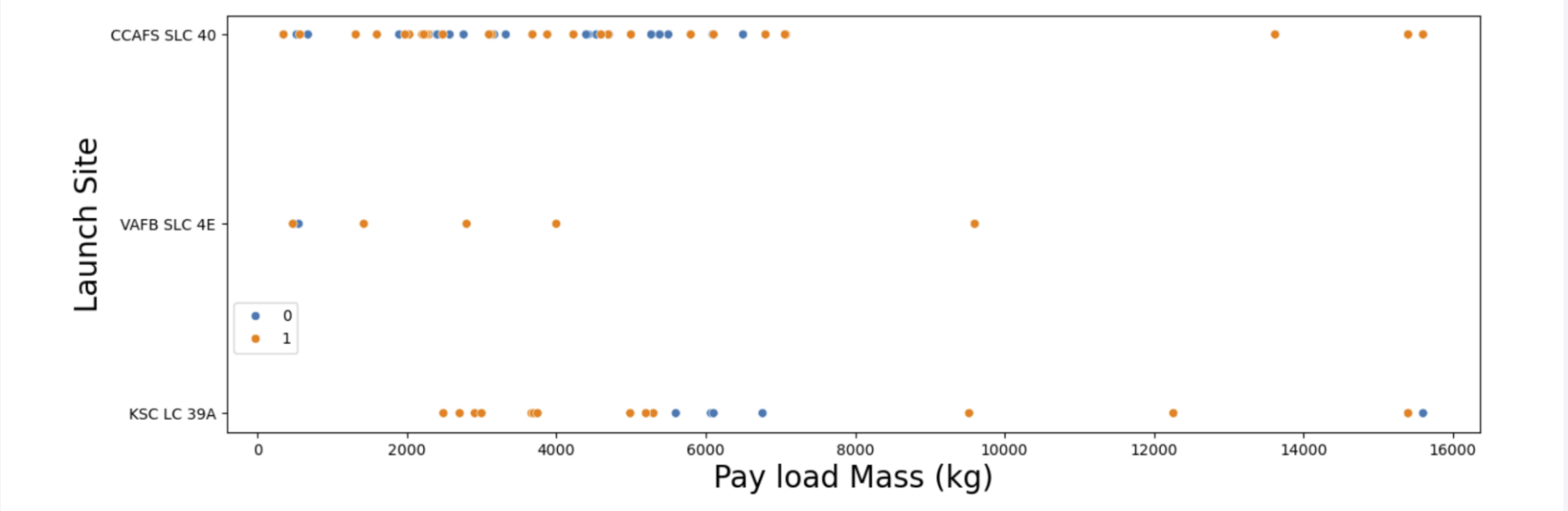

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



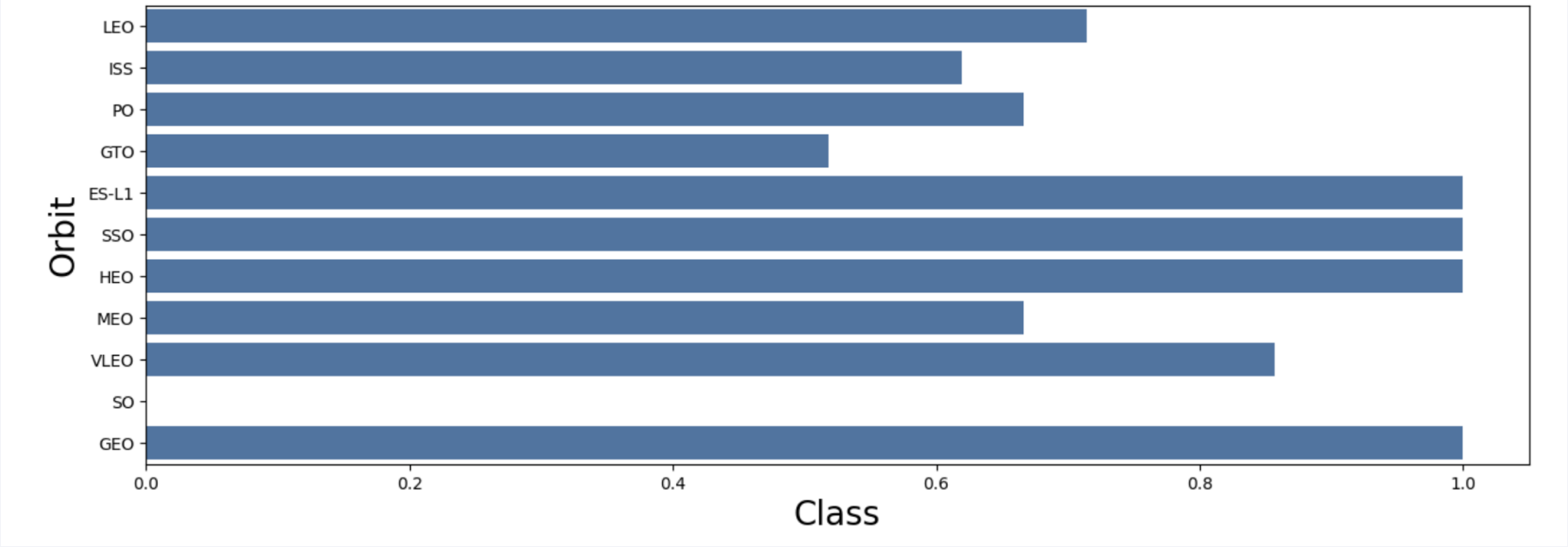
- Sites originally struggled with failures, positive checks strengthened over time
- **VAFB SLC 4E** had the best performance in terms of percentage Class Successes
- Earlier flights principally showcased signs of weakness and failure in Class determination

Payload vs. Launch Site



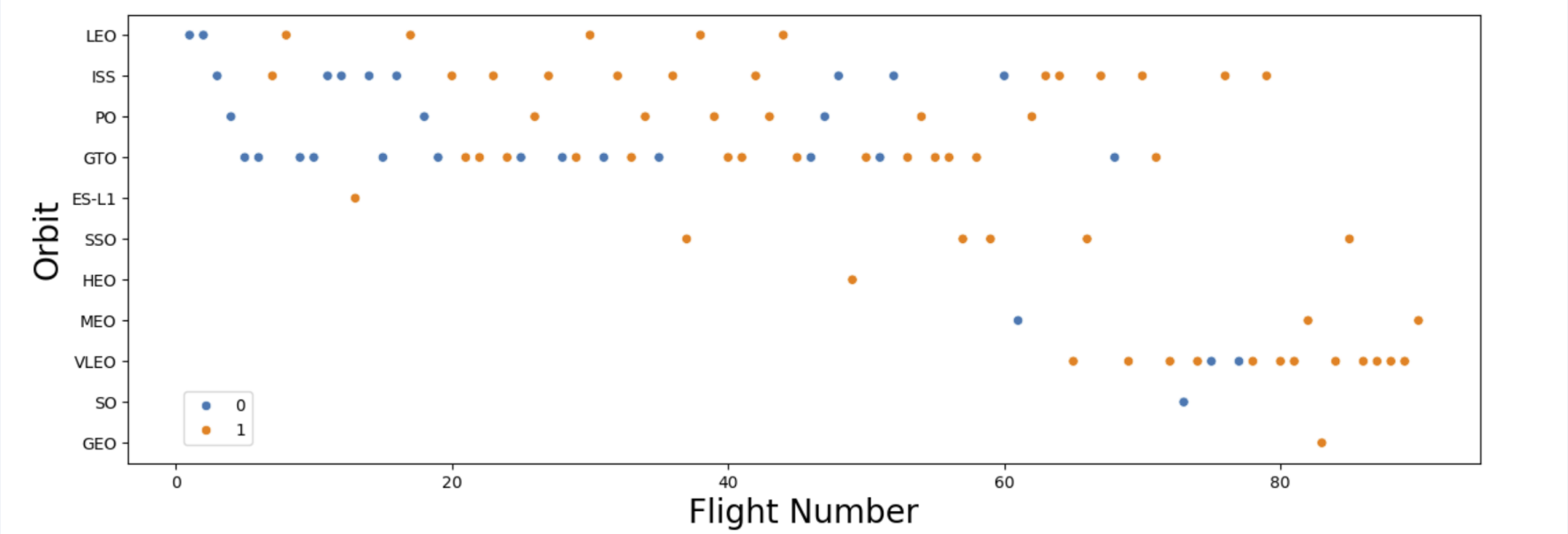
- There exists a success breaking point for all launch sites above 7,000 kg, where most launches were 1 in the classification system.
- It is seen that launches with higher Payload Mass (in kg) had higher chances of success than the overall distribution.

Success Rate vs. Orbit Type



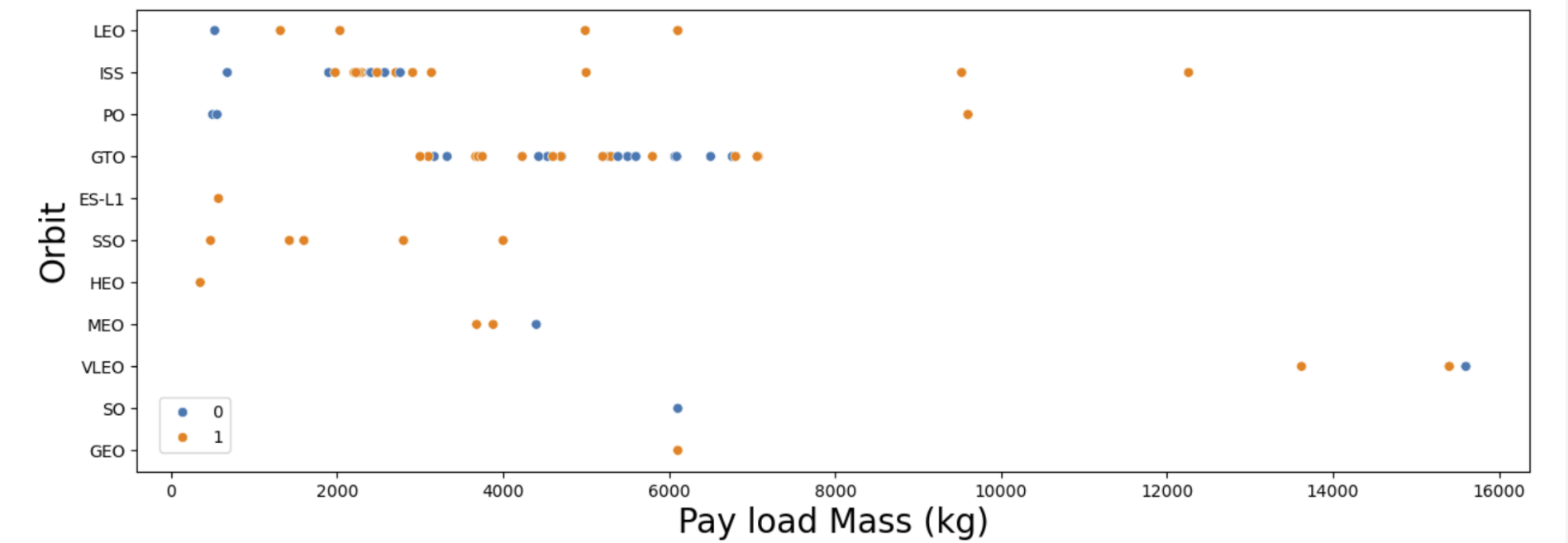
- Orbits such as ES-L1, SSO, HEO, and GEO consistently performed the best amongst the rest, with success rates near or at 1.
- SO, having only one launch, could not display any tangible data to be studied.

Flight Number vs. Orbit Type



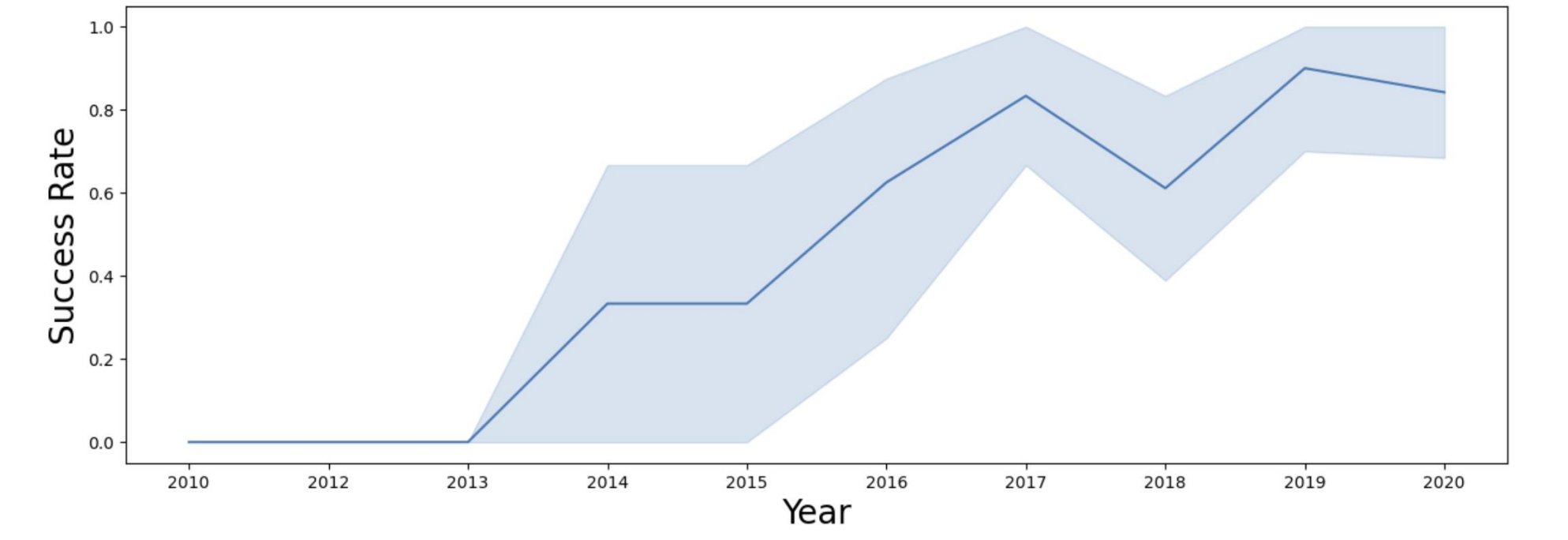
- There exists a wide diversity amongst orbits and flight numbers, with some participating across the broadness of the spectrum and some only at the upper echelons of the flight numbers.
- As flight numbers increase, generally, the success rate also increases (which can generally be seen as FL > 60).

Payload vs. Orbit Type



- The most successful orbits typically have tighter ranges for payload masses such as SSO, ES-L1, HEO, and GEO.
- As such, orbits with tighter ranges showcase better success rates than wider ones.

Launch Success Yearly Trend



- Success rate has generally increased year on year, with only a significant downturn present between 2017 and 2018.
- This perhaps indicates technological improvements and better testing/safety measures in place to ensure successful launches.

All Launch Site Names

Utilizing the DISTINCT operator to locate unique launch sites.

Task 1

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

```
[28]: %sql select DISTINCT LAUNCH_SITE from SPACEXTABLE;
```

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

[28]:

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

Launch Site Names Begin with 'CCA'

Simply utilizing the LIMIT statement to display 5 records, where launch sites start with ‘CCA’.

Task 2

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

[30]:
```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

[30]:

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

Utilizing the built-in SUM function to display the total payload mass carried by boosters launched by NASA.

Task 3

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

```
[31]: %sql SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

[31]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

Using the AVG function in SQL to illustrate the average payload mass carried by booster version F9 v1.1

Task 4

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

[32]:
```
%sql SELECT AVG("Payload_Mass__kg_") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

[32]:

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

Constructing a SQL statement around the MIN function and LIKE operator to narrow down data findings

Task 5

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

Hint:Use min function

[33]:
```
%sql SELECT MIN("Date") AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (ground pad)';
```

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

[33]:

First_Successful_Landing_Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Proffering the AND operator to initiate logical statements for purposes of filtration and calculation

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

[34]:
```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Success (drone ship)' AND "Payload_Mass__kg_" > 4000 AND "Payload_Mass__kg_" < 6000;
```

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

[34]:

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

Total Number of Successful and Failure Mission Outcomes

Developing the statement through the GROUP BY operator, decreeing counts by outcomes

Task 7

List the total number of successful and failure mission outcomes

[35]:
```
%sql SELECT "Landing_Outcome", COUNT(*) AS Total_Missions FROM SPACEXTABLE GROUP BY "Landing_Outcome";
```

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

[35]:

Landing_Outcome	Total_Missions
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

Decreeing a subquery to further hone in requirements satisfied for max payload mass

Task 8

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

[36]:
```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE);
```

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

[36]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

Constructing the statement utilizing the SUBSTR function which aids in filtration through date initialization

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[37]:
```
%sql SELECT SUBSTR("Date", 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'Failure (drone ship)' AND SUBSTR("Date", 1, 4) = '2015';
```

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

[37]:

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

Lastly ranking landing outcomes through the ORDER BY clause to showcase counts in a desc. order

Task 10

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.

[38]:
```
%sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Outcome_Count DESC;
```

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

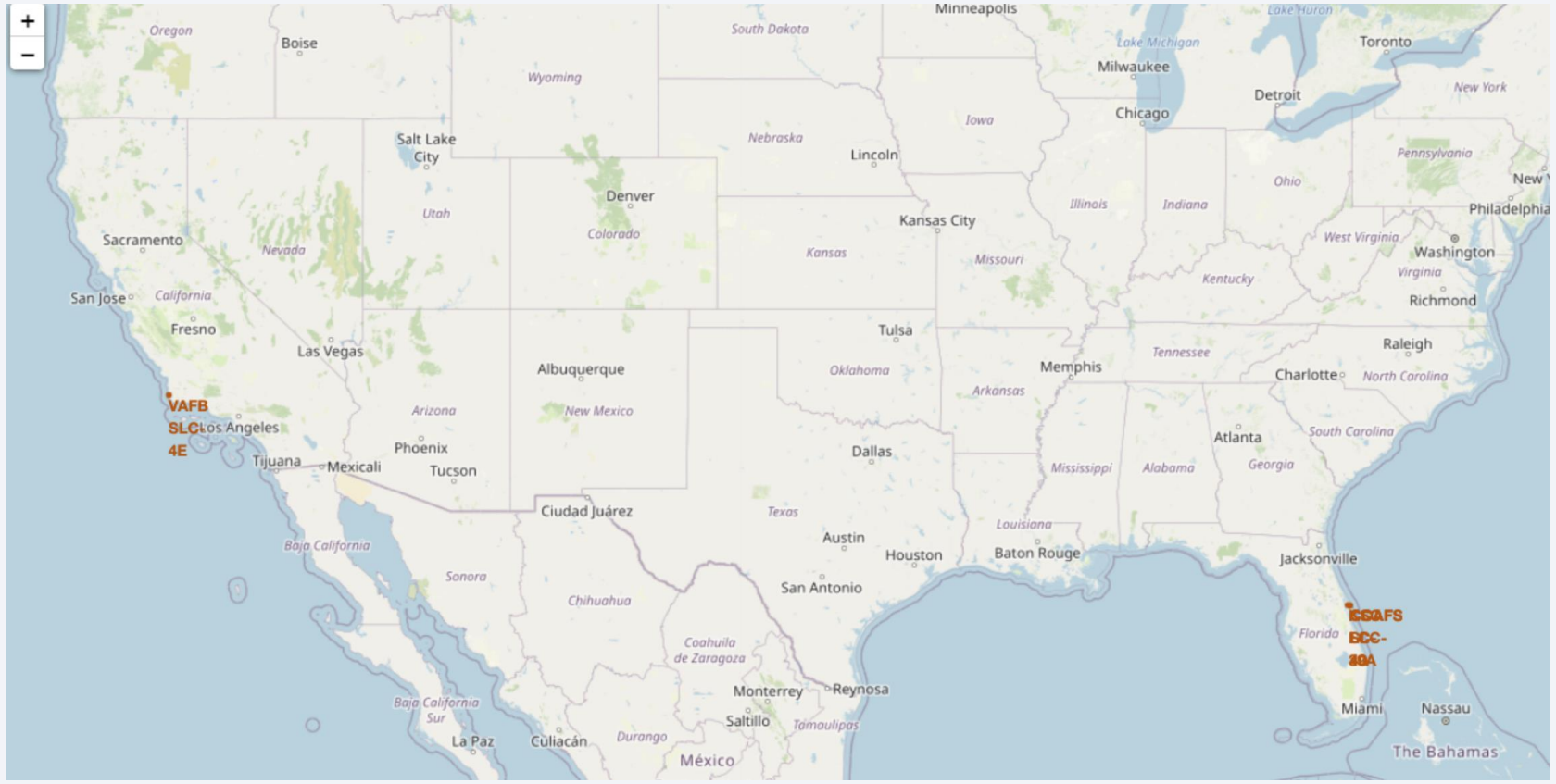
[38]:

Landing_Outcome	Outcome_Count
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

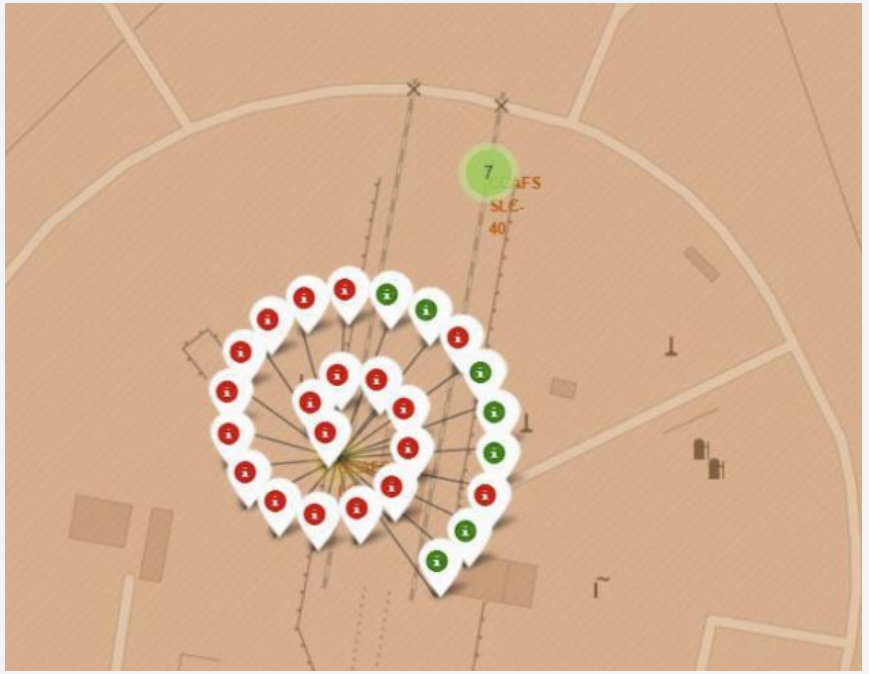
Launch Sites Proximities Analysis

Nationwide Map of Launch Sites



The major launch sites are located in California and Florida.

A Closer Zoom-In on Launch Site Success Rates



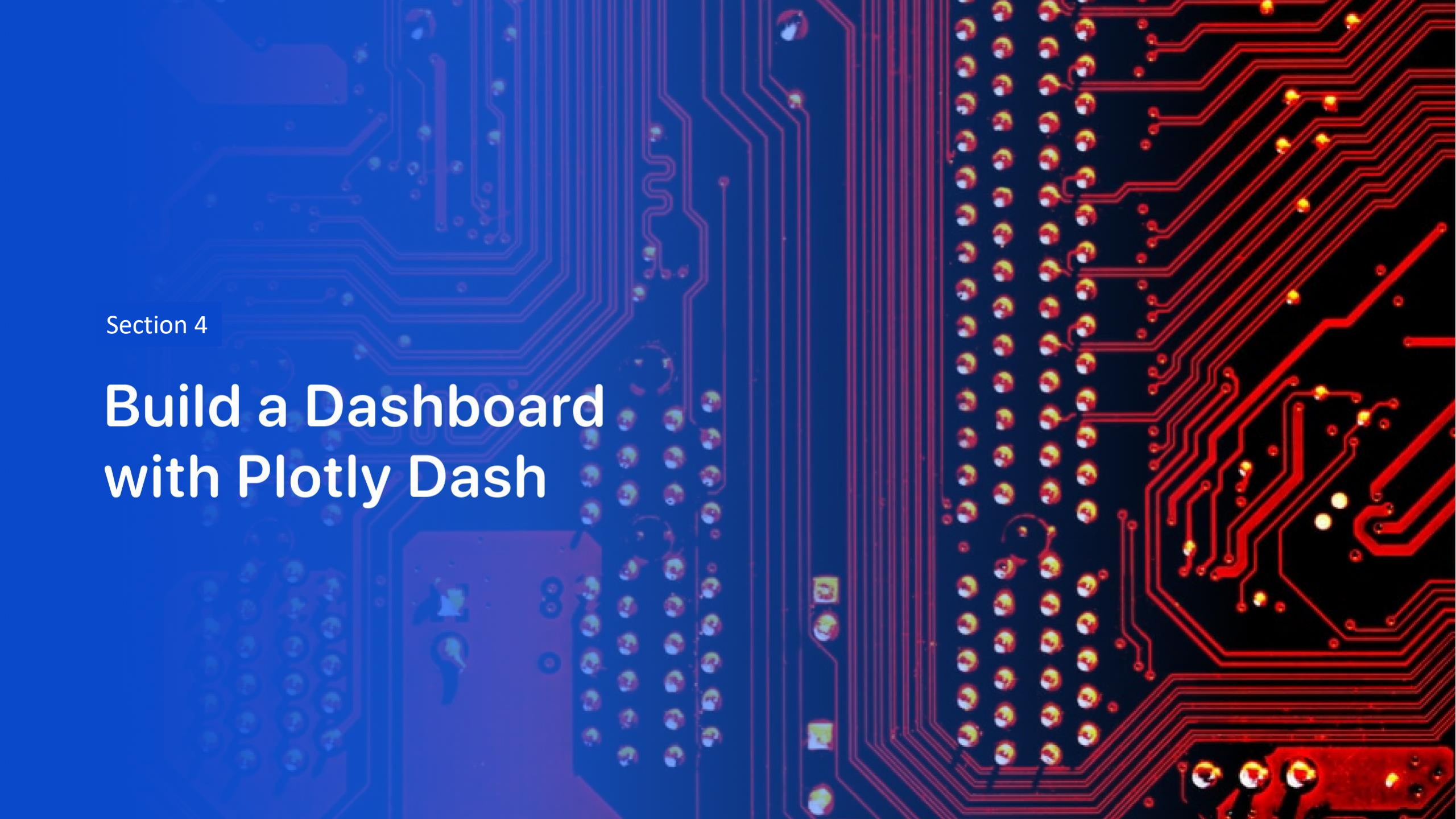
There exist more launch sites present within Florida than California.

In so, Folium markers were added to showcase the success and failure rates at launch sites.

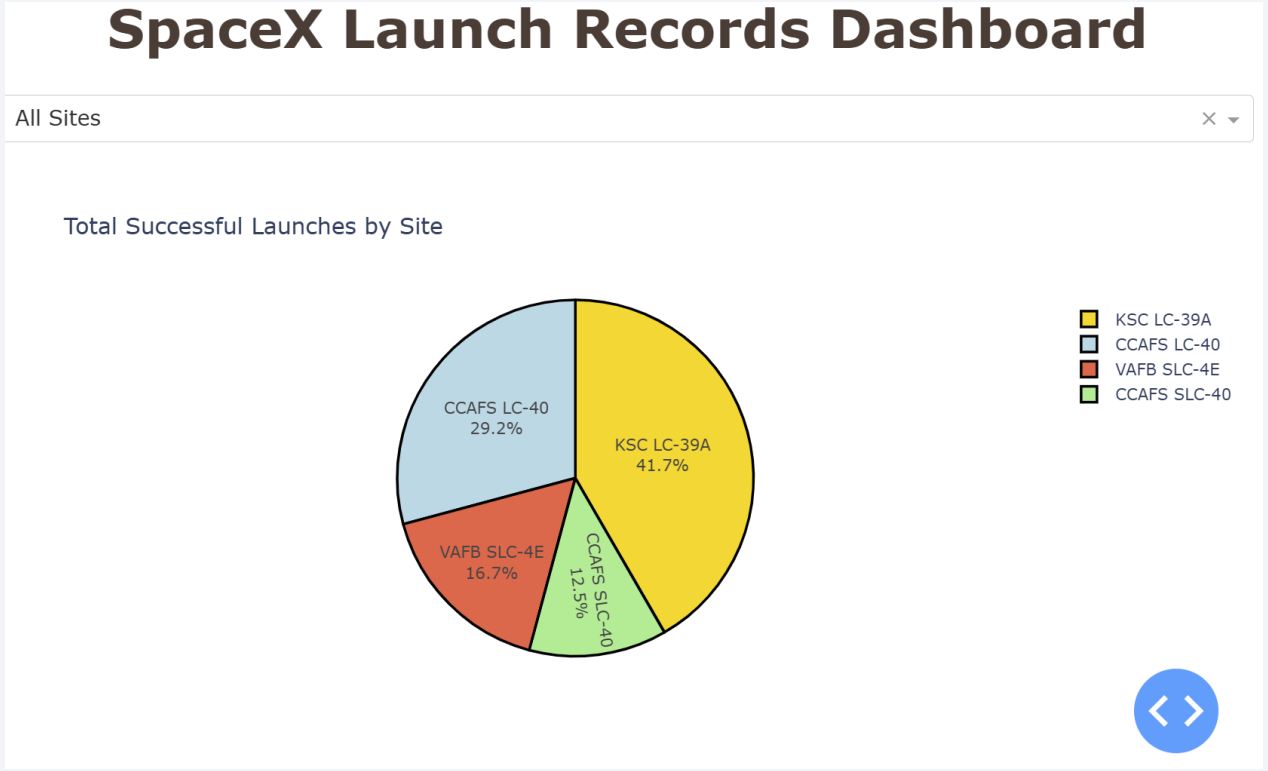
Notable Locations and their Distances Computed

This Folium line constructs the distance (straight-line) between landmarks and the launch sites.

A couple keynote examples of this include the image above, where the road is 0.6km away and the coast is 0.9 km afar.


Section 4
Build a Dashboard with Plotly Dash

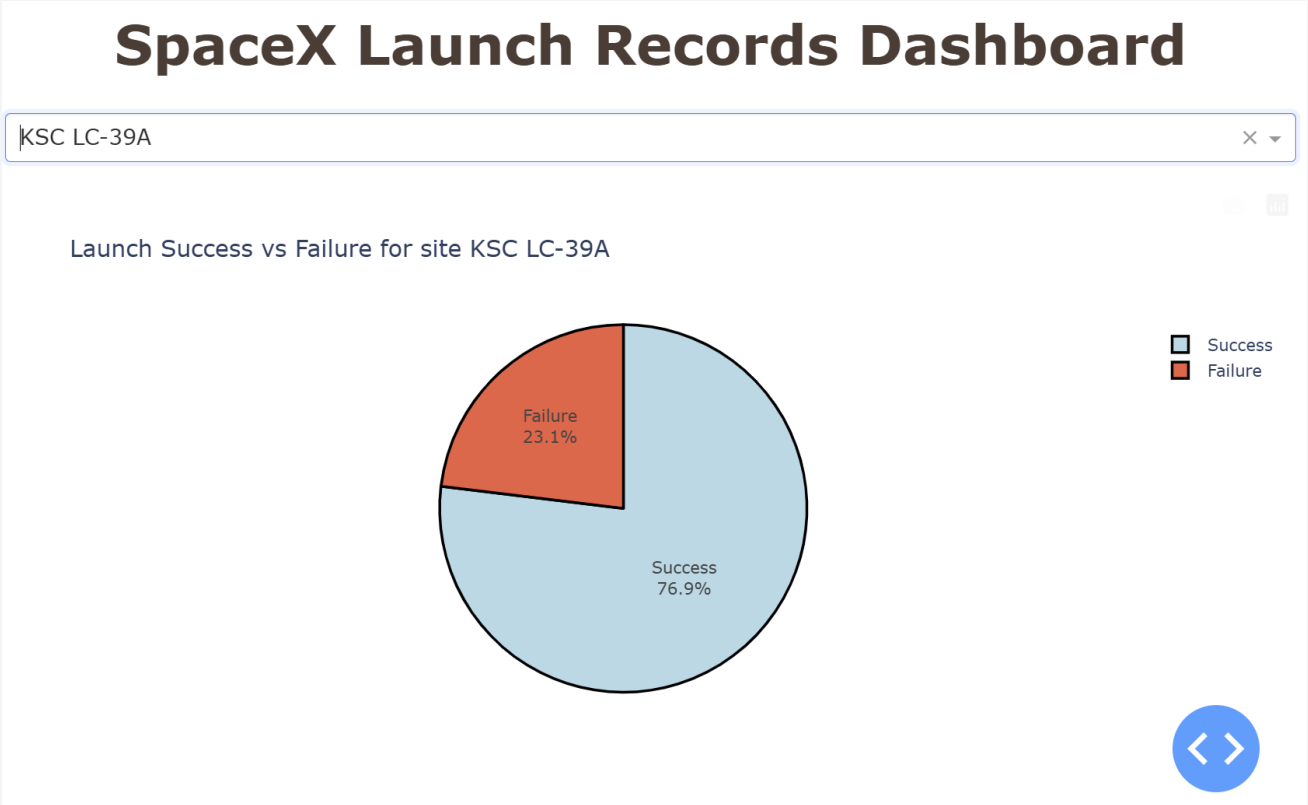
Pie Chart of Success Rates by Launch Site (SpaceX)



KSC LC-39 had the highest percentage of successful launches amongst all the other sites.

CCAFS SLC-40 had the least successful, at nearly a quarter of the percentage of KSC LC-39.

Pie Chart of Success v. Failure Rates for site KSC LC-39A



This graph serves as an illustration of the most successful launch site's success v. failure rate.

KSC LC-39 had a 3-1 ratio of success to failure, indicating a very strong domain for launches.

Payload Success Rate Scatterplot for All Sites

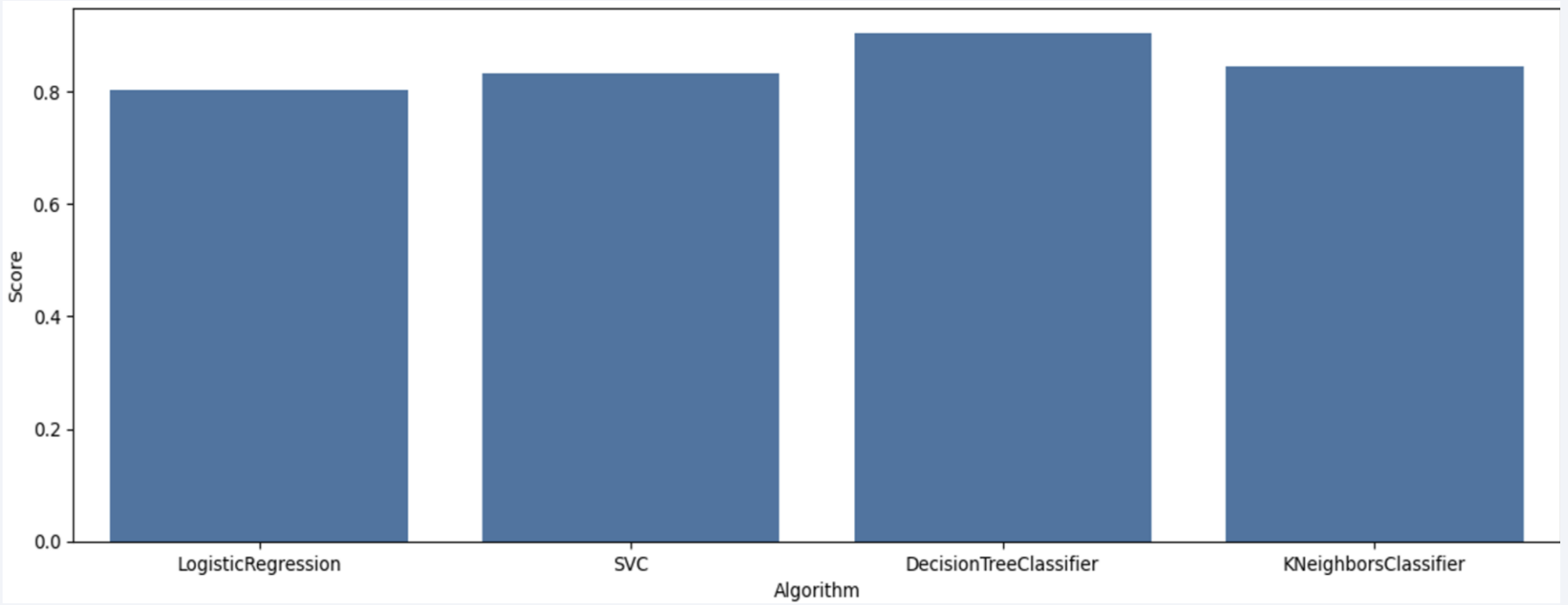
The FT and B4 boosters performed the best across all major versions and had the highest success rates evdent by the collection of 1s situated at the top of the binary chart.

Having the bounds set for payload masses between 3,000 - 5,000 kg, it can be seen that v1.1 performs under par.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

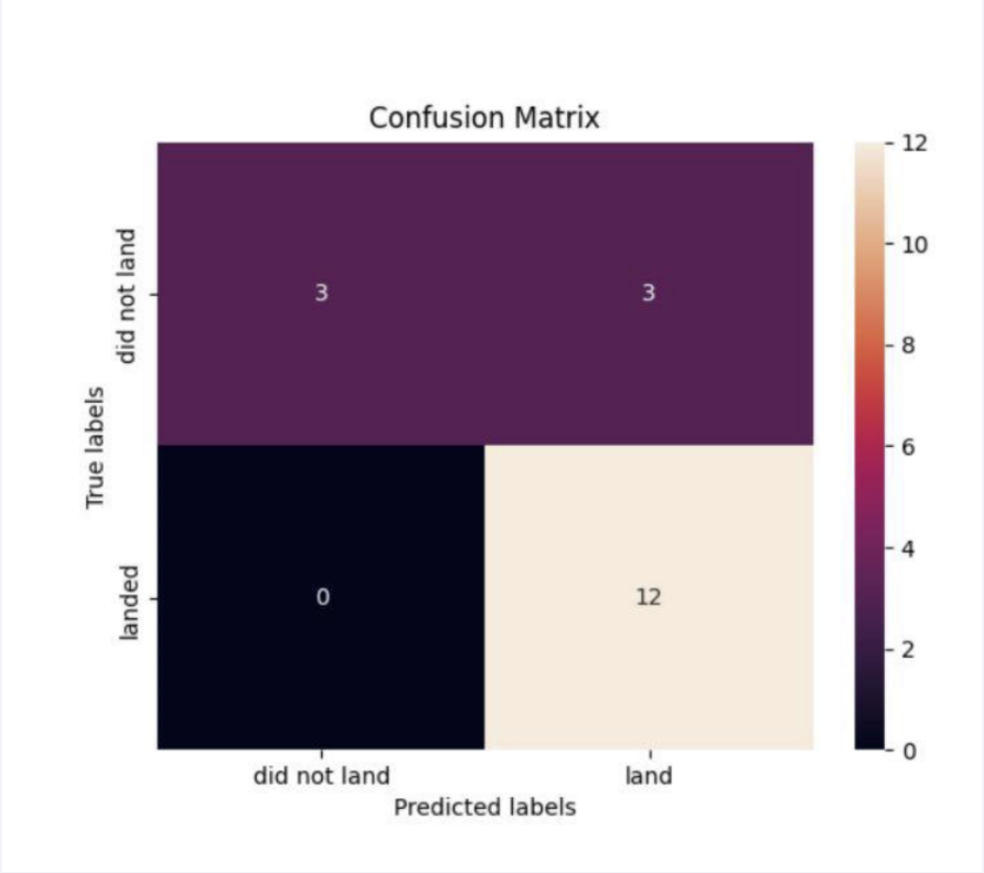


The DecisionTreeClassifier had the highest predictive accuracy as is compared with all the other algorithms, and is the only one closest to performing at nearly 90% accuracy.

The other algorithms follow in accuracy order of KNeighborsClassifier, SVC, and LogisticRegression in last.

Confusion Matrix

- This image represents the Confusion Matrix for the DecisionTreeClassifier.
- Overall it can be seen that 15/18 predictions were true, though false positives are evidently an issue here, skewing perception of the model.



Conclusions

- Summary of all results
  - *Success Rate Over Time:* Significant Improvement over Defined Periods
  - *Success Rate by Launch Site:* The highest success rate of all launch sites was **KSC LC-39A**.
  - *Success Rate by Orbit:* The highest success rates observed were amongst **ES-L1, SSO, HEO, & GEO**.
  - *Predictive Algorithms:* The **DecisionTreeClassifier** was most accuracte in the prediction of landing outcomes.
  - *Payload Analysis:* It is seen that heavier payloads have higher failure rates, though success of which improves over time.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!