IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Cynthia Appiah-Osafo>
<9/26/2023>

Outline

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

Executive Summary

- ***Methodologies Used***

- Data Collection from API and Web Scraping
- Data Wrangling.
- Exploratory Data analysis using SQL, Pandas, and MatplotLib
- Interactive Visual Analytics and Dashboard with Folium and Plotly Dash.
- Predictive Analysis with Machine Learning.

- ***Results***

- The best method used for testing data.
- Good parameters for Logistic Regression, KNN classifiers, SVM, and decision Tree

Introduction

- SpaceX : we make rocket launches which are very affordable and price effective as compared to other companies.
- The Falcon 9 rockets that we make can be reuse in its first stage.

- Therefore, we can use Data science and Prediction analysis to determine the landing in the first stage and the cost of a launch of alternate companies.

Section 1

Methodology

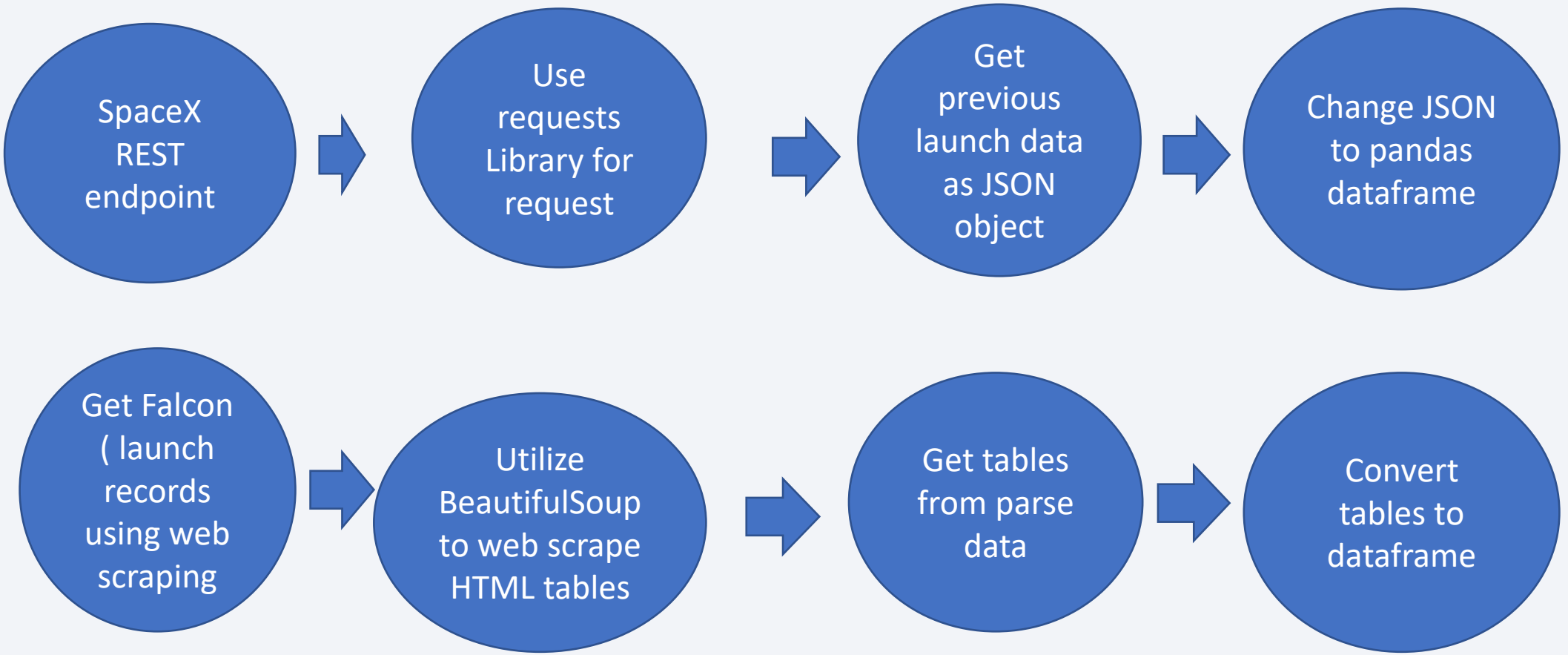
Methodology

Executive Summary

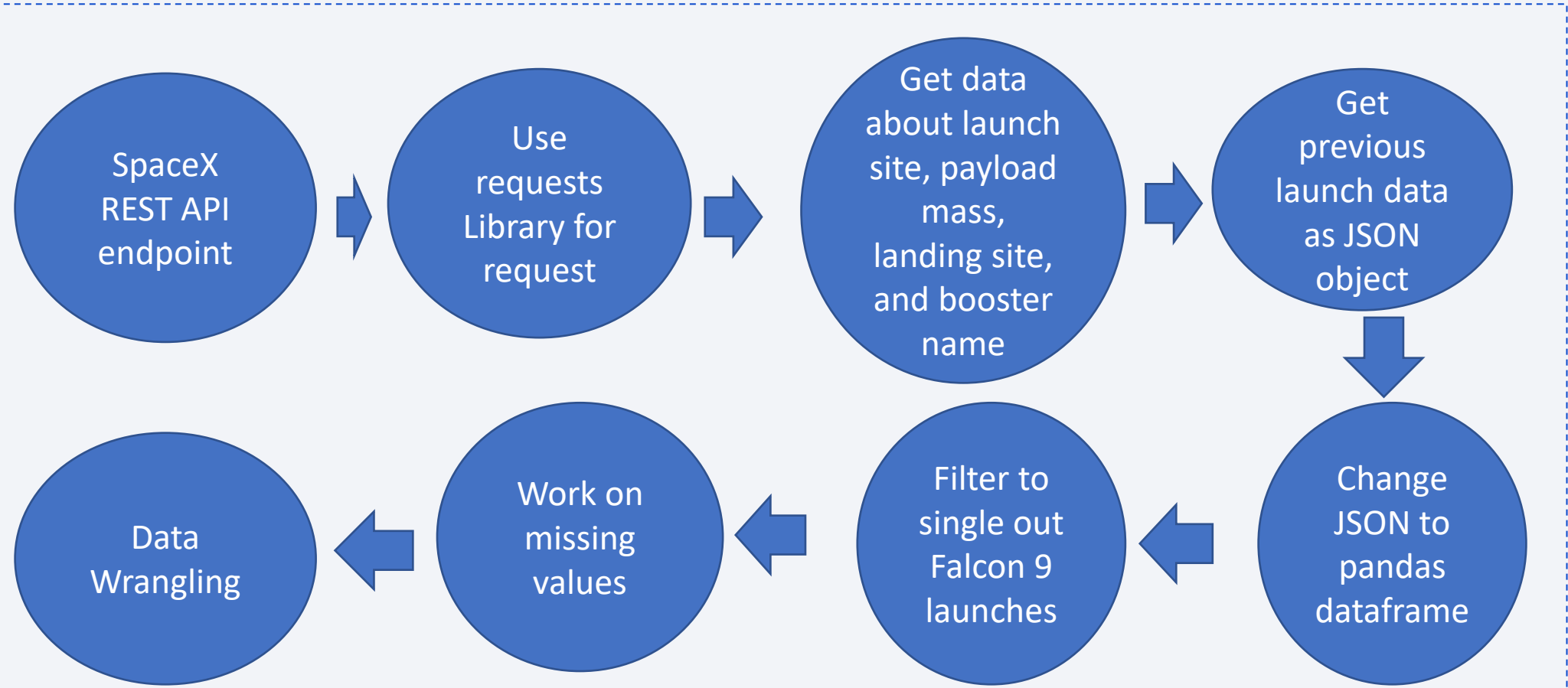
- Data collection methodology:
  - Data was collected from SpaceX Rest API and wiki pages using data wrangling.
- Perform data wrangling
  - JSON object and HTML tables were processed into pandas dataframe for better analysis and easy visualization.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using Machine Learning to predict if the first landing will be successful.

Data Collection

Collecting Data form Space X REST API / using web scraping to gather information.



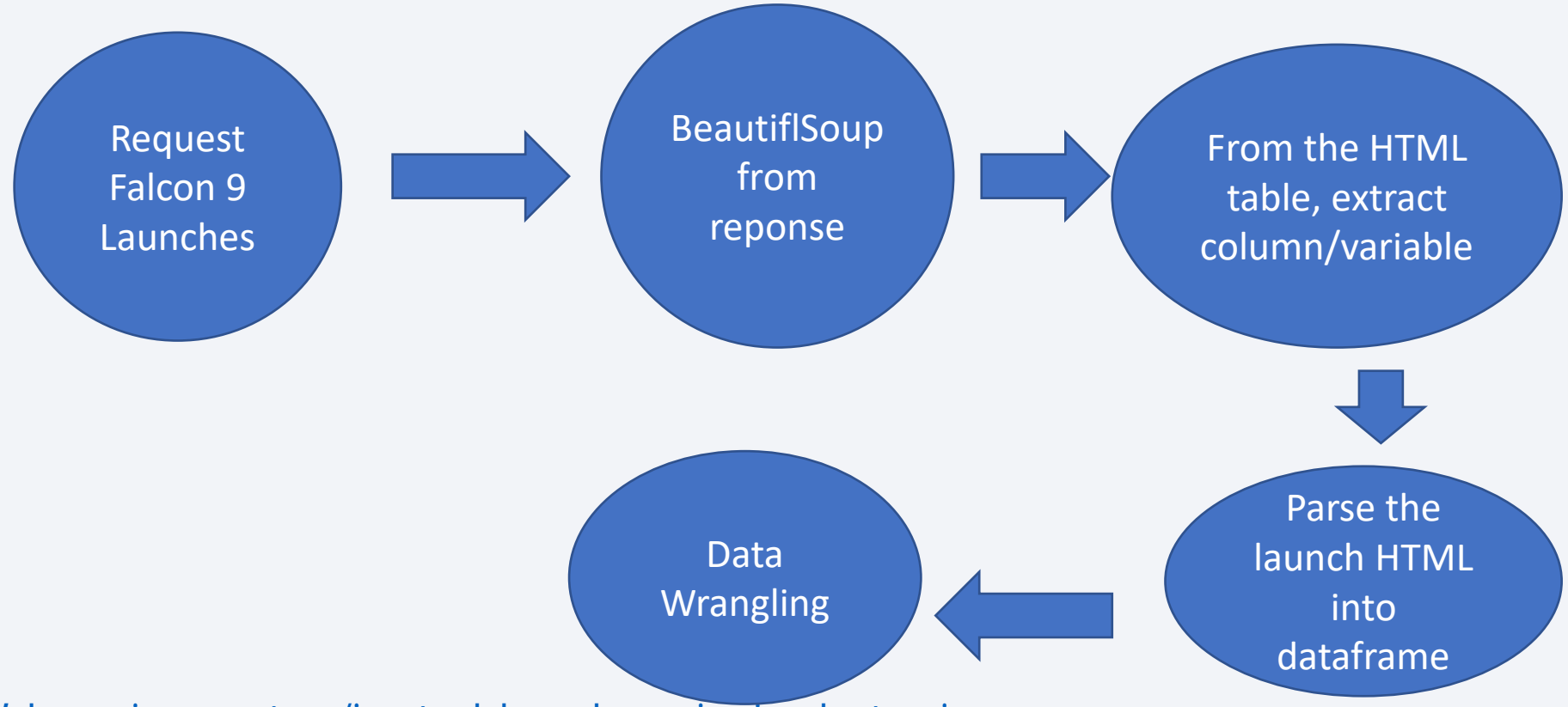
Data Collection – SpaceX API



- Lab_1/jupyter-labs-spacex-data-collection-api.ipynb at main · Cynthkwabs14/Lab_1 (github.com)

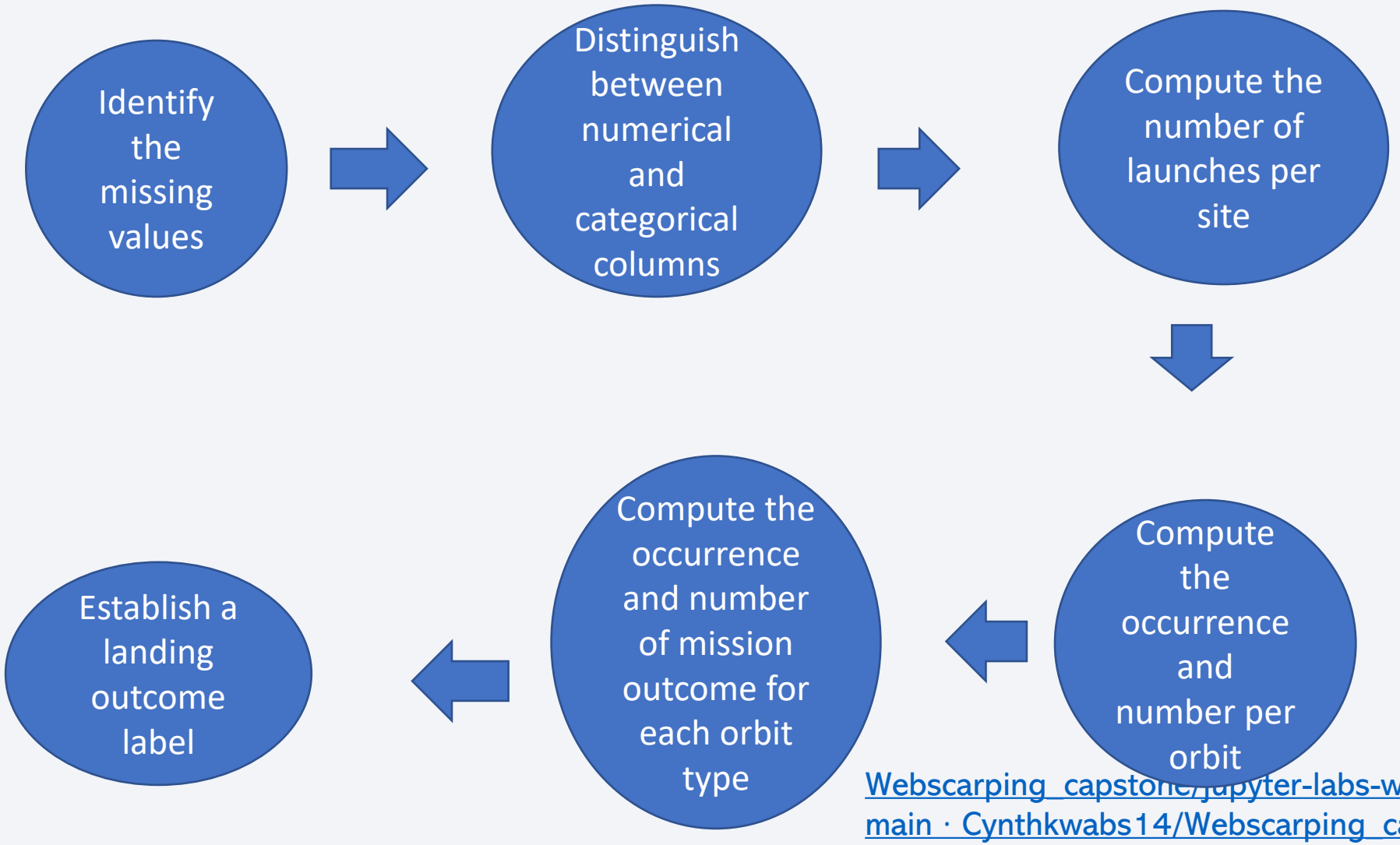
Data Collection - Scraping

Collecting the history of Falcon 9.



Webscarping_capstone/jupyter-labs-webscraping.ipynb at main · Cynthkwabs14/Webscarping_capstone (github.com)

Data Wrangling



Webscarping_capstone/jupyter-labs-webscraping.ipynb at main · Cynthkwabs14/Webscarping_capstone (github.com)

EDA with Data Visualization

Plots that was used are:

1. Scatter Plot: Show the relationship between variables.
- Flight number VS. Payload mass
- Flight number VS. Launch site
- Payload mass VS. Launch site
- Flight number VS. Orbit
- Payload mass VS. Orbit type
- Orbit VS. Payload mass

2. Bar chart: It is used for comparison of variables. It shows relationship between categorical and numerical variables.

- Orbits VS. Success rate (mean)

3. Line chart: It is used to show behavior of variables and make connections to show unseen predictions.

- Success Rate VS. Year

- capstone_EDA_Visualization_Lab/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb at main · Cynthkwabs14/capstone_EDA_Visualization_Lab (github.com)

EDA with SQL

SQL task Performed:

- first load the SQL extension and establish a connection with the database.
- 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 successful landing outcome in ground pad was achieved.
- 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.
- 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.

- SQL_For_Peer_Assignment/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · Cynthkwabs14/SQL_For_Peer_Assignment (github.com)

Build an Interactive Map with Folium

Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas

• Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle,folium.map.Marker).

• Red circles at each launch site coordinates with label showing launch site name (folium.Circle,folium.map.Marker, folium.features.DivIcon).

The grouping of points in a cluster to display multiple and different information for the same coordinates. (folium.plugins.MarkerCluster).

Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (folium.map.Marker, folium.Icon).

Markers to show distance between launchsite to key locations (railway, highway,coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)

These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successfuland unsuccessful landings.

- Capstone_Analysis_with_Folium/lab_jupyter_launch_site_location.jupyterlite.ipynb at main · Cynthkwabs14/Capstone_Analysis_with_Folium (github.com)

Build a Dashboard with Plotly Dash

- **The Interactive element:**
- Dropdown allows a user to choose the launch site or all launch sites.
- Pie chart shows the total success and failure for the launch site chosen with the dropdown.
- Rangeslider allows a user to select a payload mass in a fixed range.
- Scatter chart shows the relationship between two variables, e.g., Success vs Payload Mass.

Predictive Analysis (Classification)

Data preparation

Load dataset
Normalize data
Split data into training and test sets.

Model preparation

Selectionof machine learning algorithms
Set parameters for each algorithm to GridSearchCV
Training GridSearchModel models with training dataset

Model evaluation

Get best hyperparameters for each type of model
Compute accuracy for each model with test dataset
Plot Confusion Matrix

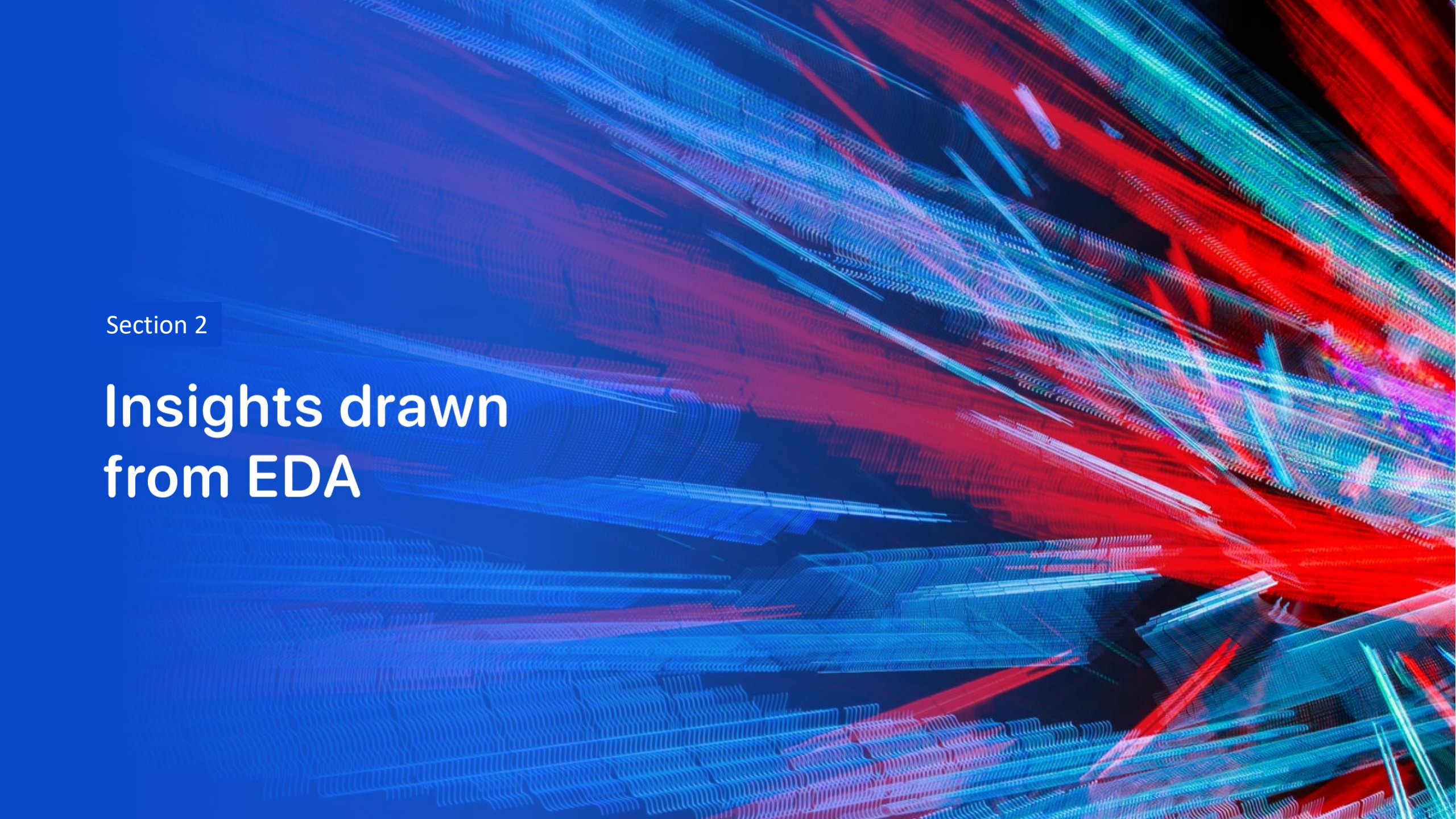
Model comparison

Comparison of models according to their accuracy
Choose the model with the best accuracy.

Machine_Learning_Capstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · Cynthkwabs14/Machine_Learning_Capstone (github.com)

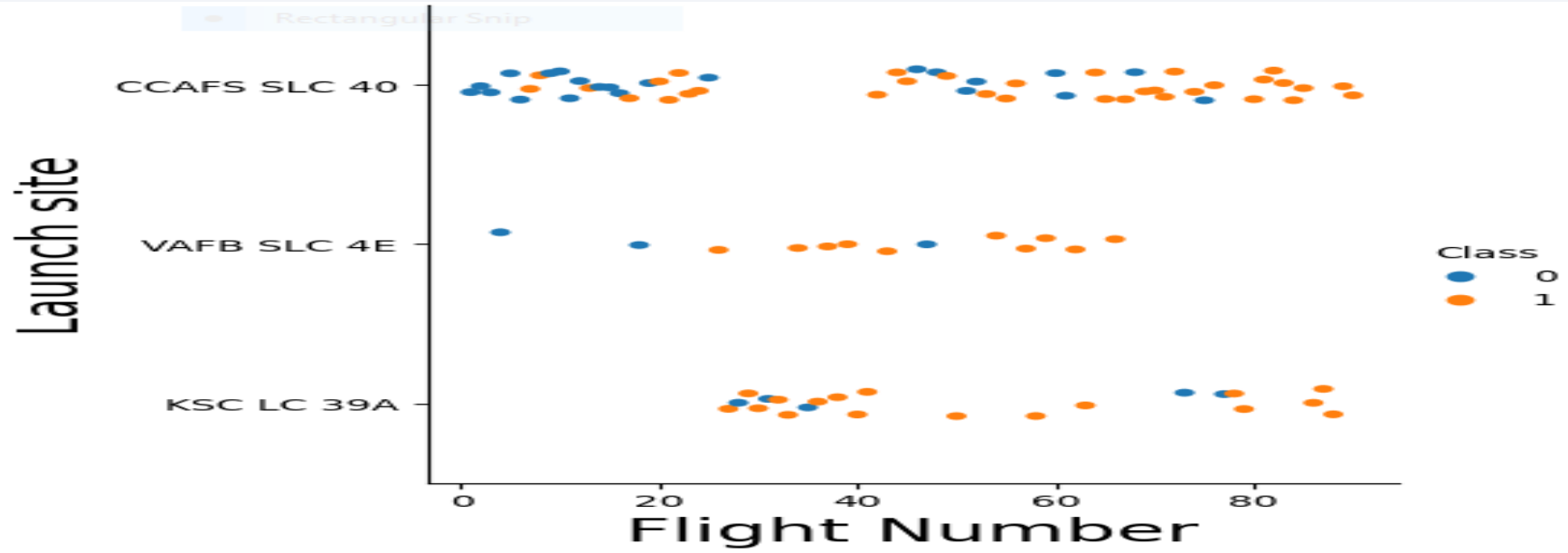
Results

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


Section 2
Insights drawn from EDA

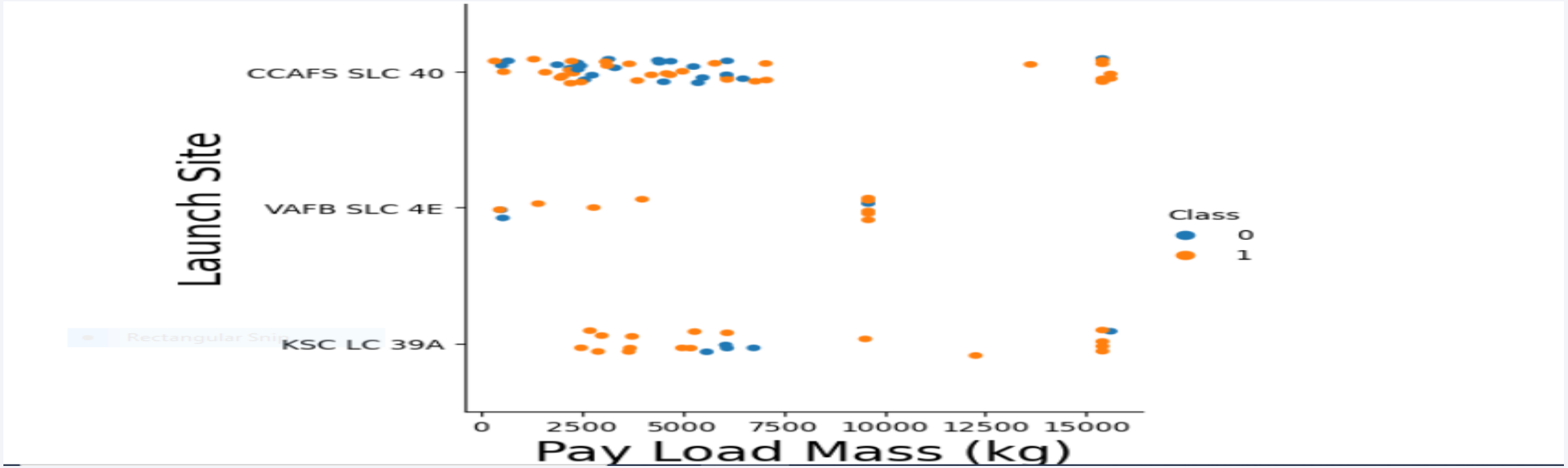
Flight Number vs. Launch Site

The higher the number of flight for the launch site, the higher success rate for the landing.



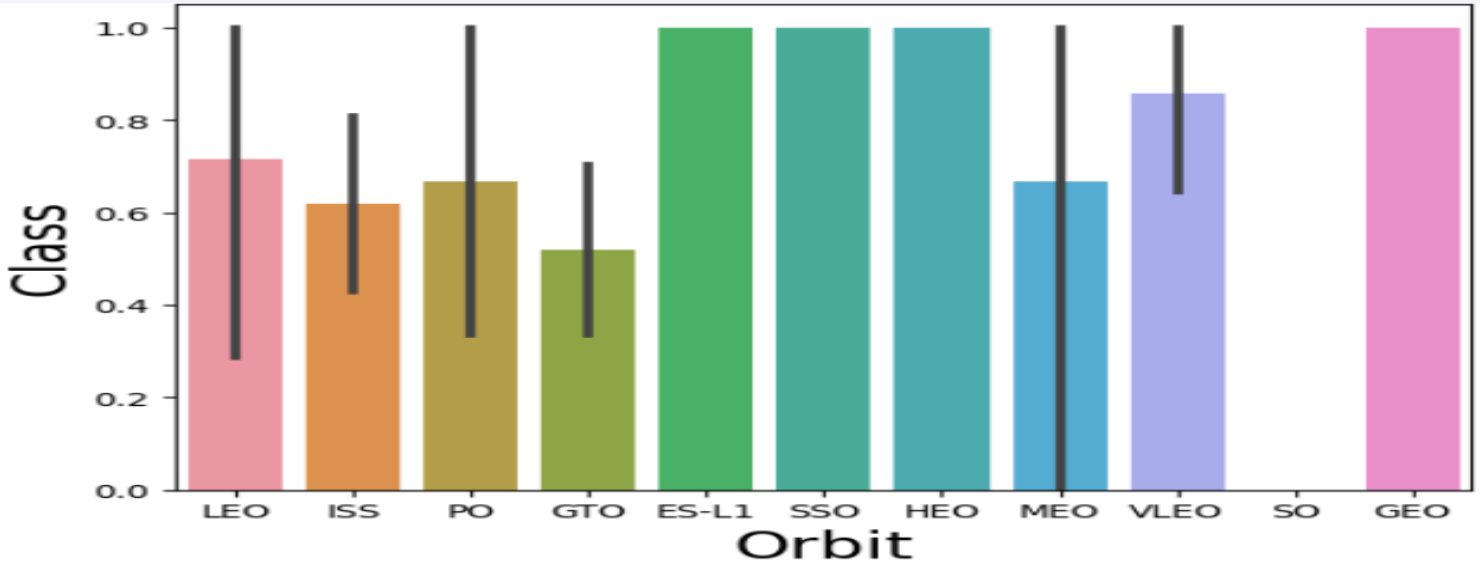
Payload vs. Launch Site

Payload Mass below 5500kg and above 7000kg has a tendency of achieving higher success rate. VAFB SLC 4E has no payload mass above 10000kg which will make it difficult to find and trend for this site.



Success Rate vs. Orbit Type

ES-L1, SSO, HEO, and GEO has the best success rate.

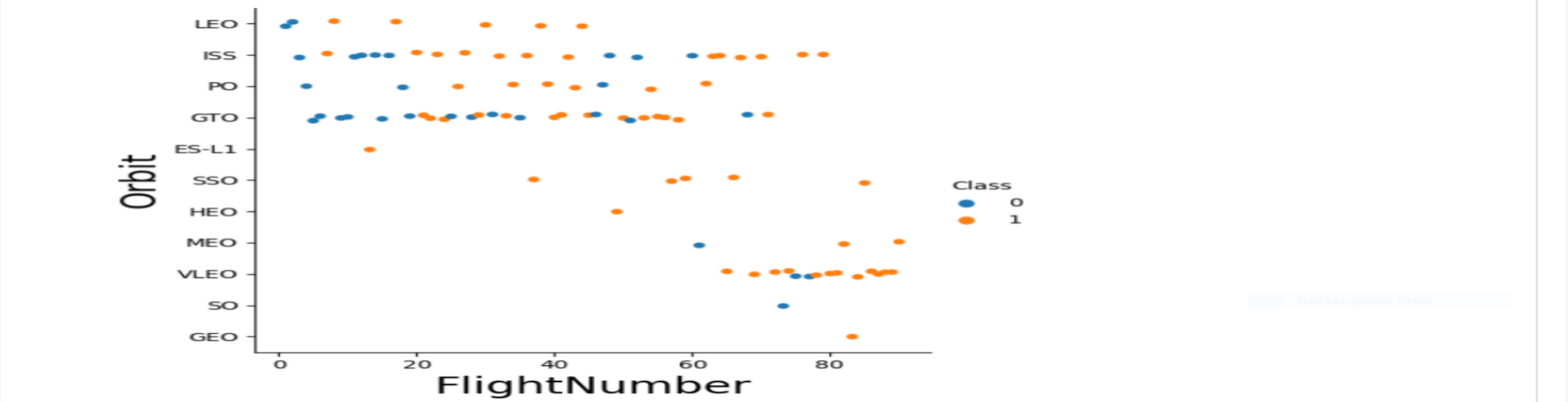


Flight Number vs. Orbit Type

the success rate increases with the number of flights for the LEO orbit.

For some orbits like GTO,there is no relation between the success rate and the number of flights.

But we can suppose that the high success rate of some orbits like SSOor HEO is due to the knowledge learned during former launches for other orbits.



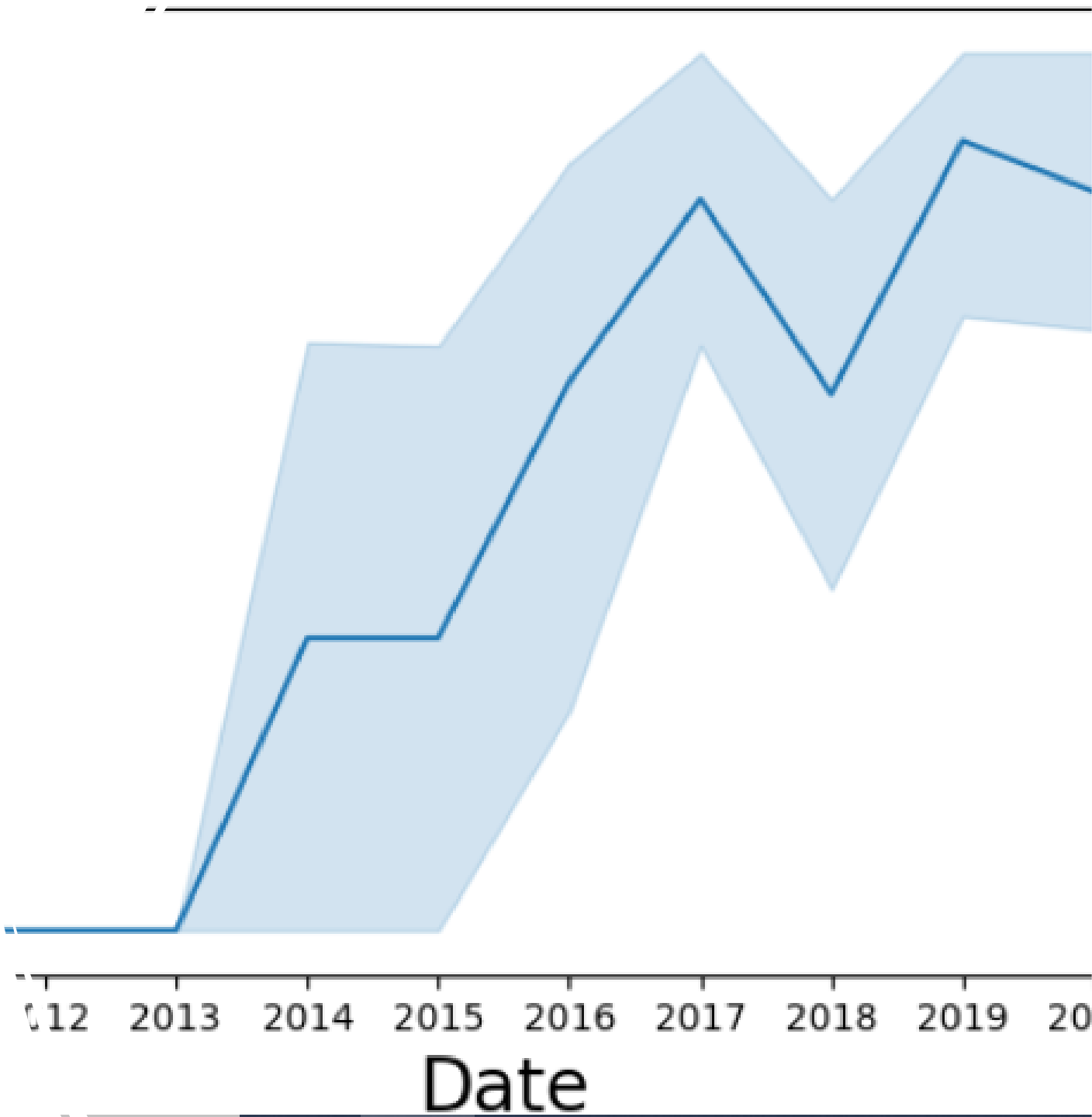
Payload vs. Orbit Type

As payload mass increase, the orbit PO, LEO, and ISS has an increased success rate.

- For GTO, we cannot distinguish well as both positive landing rate, and negative landing are both present.

Launch Success Yearly Trend

- The success rate started to have a sharp increase in the year 2013 and kept increasing until 2020.



All Launch Site Names

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

The UNIQUE statement is used to find all unique launch sites from the Launch_Site column.

```
%sql SELECT DISTINCT("Launch_Site") from SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" like "CCA%" LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landin
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure
2010-08-12	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
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

The LIKE operator was used in the query to find launch site names that begins with 'CCA.'

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE "Customer" = "NASA (CRS)"
```

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

SUM(PAYLOAD_MASS__KG_)
45596

This query used the SUM function to total the amount of payload mass from the column payload_mass_kg_ launched by the customer ‘NASA (CRS)’

Average Payload Mass by F9 v1.1

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

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

AVG(PAYLOAD_MASS__KG_)
2928.4

The AVG function was used to retrieve the average payload mass from column payload_mass_kg_ that is carried by the booster ‘F9 v1.1’

First Successful Ground Landing Date

```
%sql select min("Date") from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)'
```

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

min("Date")
2015-12-22

With this query, we select the oldest successful landing. The WHERE clause filters dataset in order to keep only records where landing was successful. With the min function, we select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select "Booster_Version" from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__K
```

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

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

This query was used to get the booster version that successfully landed on a drone ship and also carried a payload mass between 4000kg and 6000kg

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT count("Mission_Outcome") FROM SPACEXTBL WHERE "Mission_Outcome" = "Success" or  "Mission_Outcome" = "
```

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

count("Mission_Outcome")
98

The total number of successful and failed mission were counted.

Boosters Carried Maximum Payload

```
%sql select ("Booster_Version") from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPAC
```

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

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

This query was used to get booster versions and payload mass with the condition that the booster carried the maximum payload mass.

2015 Launch Records

```
%sql SELECT "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTBL where "Landing_Outcome" = 'failure
```

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

Landing_Outcome	Booster_Version	Launch_Site

The query was used to find failures in 2015 launch records.

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

```
%sql SELECT "Date", count("Landin_Outcome") from SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' AND
```

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

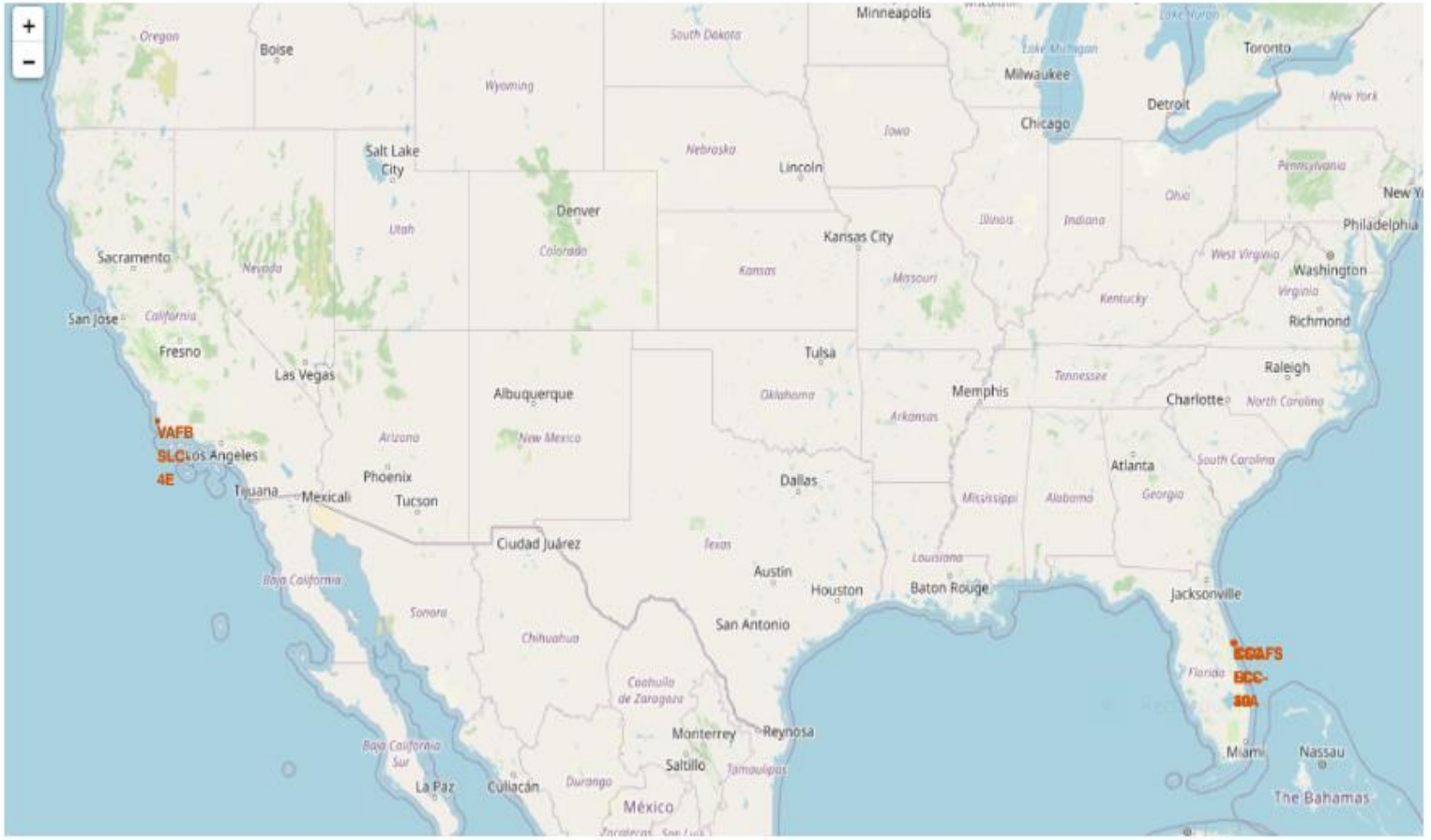
Date	count("Landin_Outcome")
2017-03-06	1
2017-02-19	1
2017-01-14	1
2017-01-05	1
2016-08-14	1
2016-08-04	1
2016-07-18	1
2016-06-05	1
2016-05-27	1
2015-12-22	1

We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECTDISTINCT) with the heaviest payload mass.

Section 3

Launch Sites Proximities Analysis

Marked Launch Sites

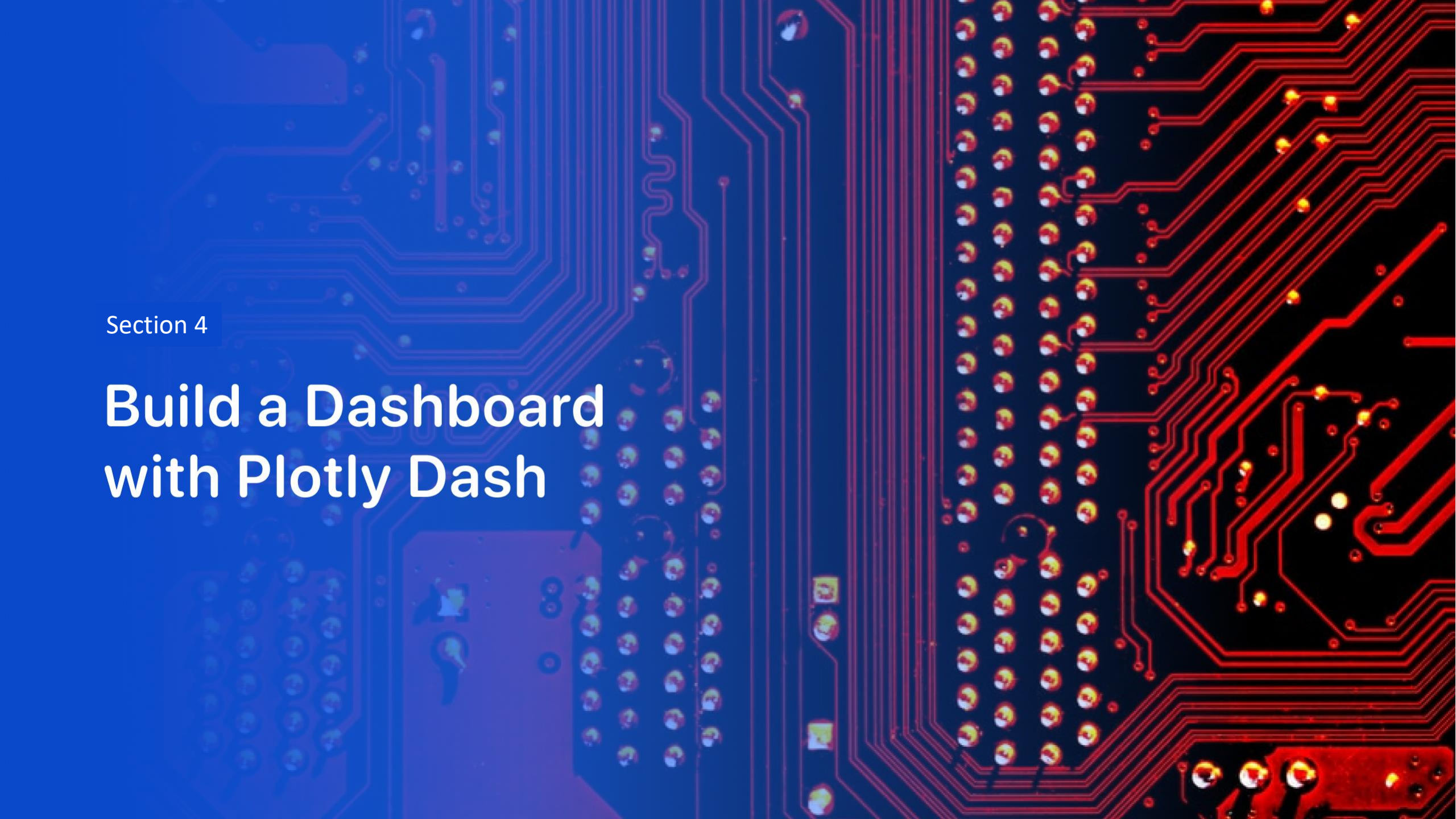


launch sites were marked and labeled for the names of the launch sites. • All launch sites are located on the United States coasts of Florida and California

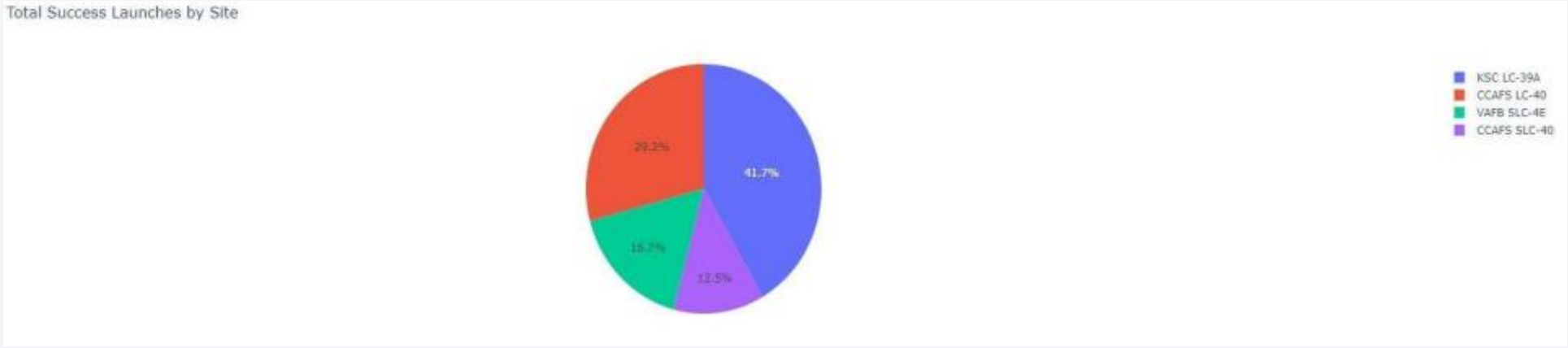
Folium Proximity Map

Green marker represents successful launches. Redmarker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

<Folium Map Screenshot 3>

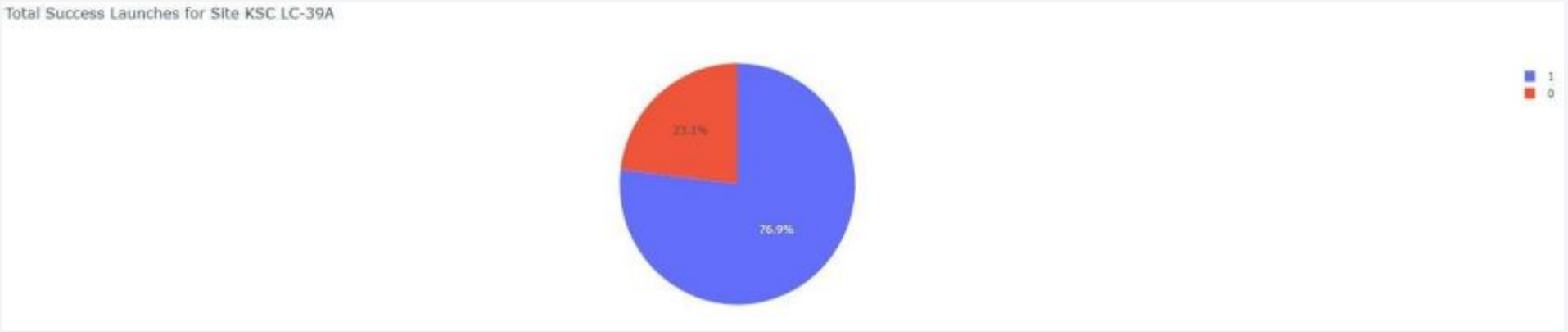

Section 4
Build a Dashboard with Plotly Dash

Launch Site Total Success



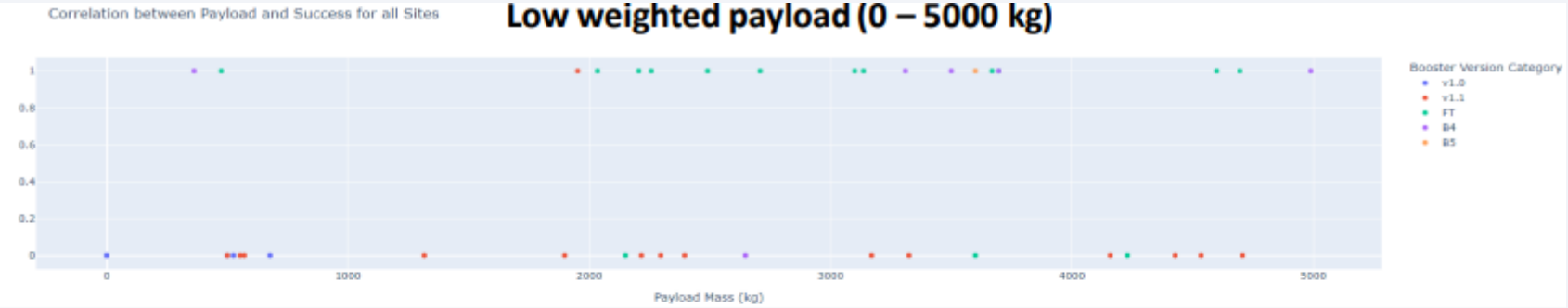
KSCLC-39A has the best success rate of launches.

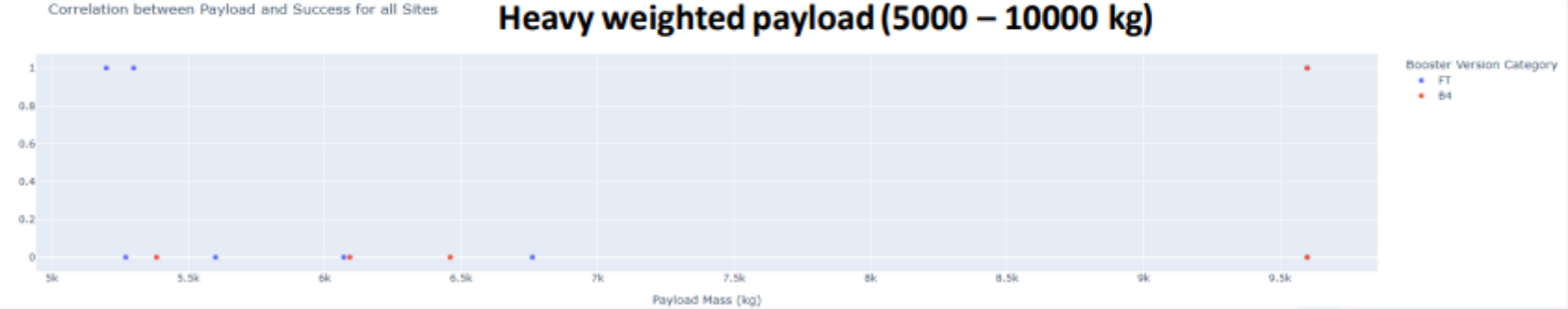
KSC LC-39A total success launches.



KSCLC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

PayLoad VS Class Plot





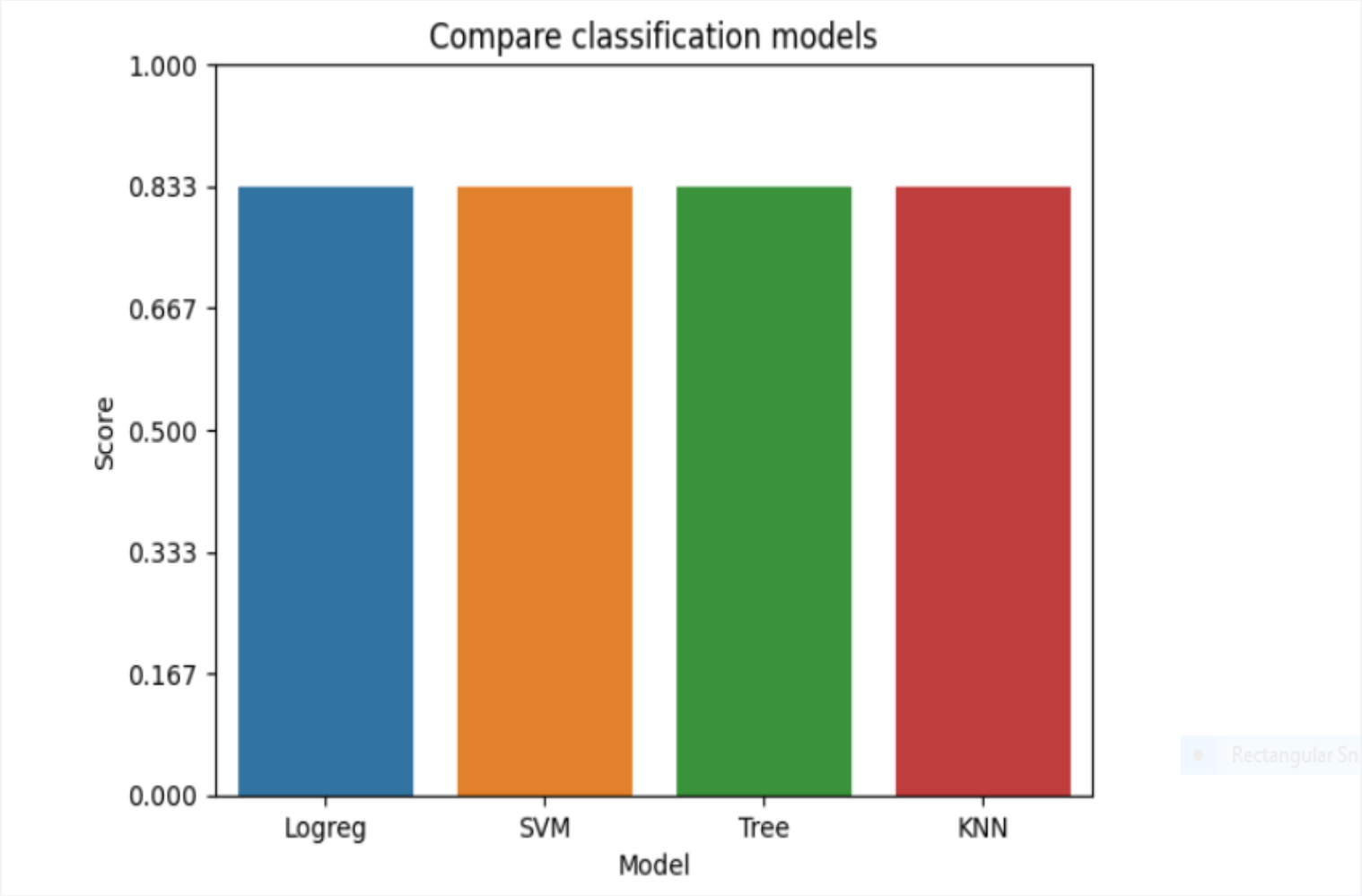
Low weighted payloads have a better success rate than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

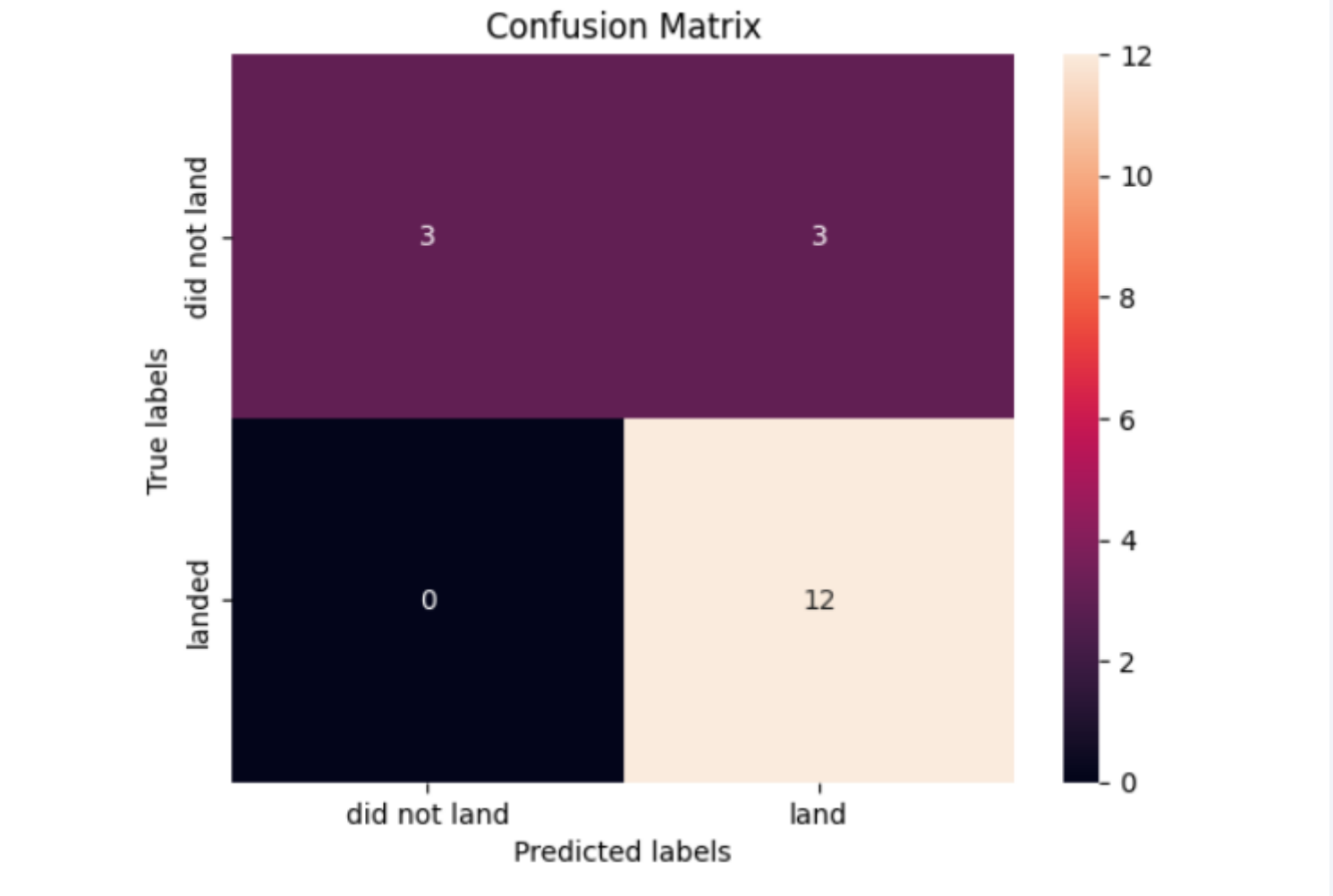
Classification Accuracy

All models has high accuracy levels.



Confusion Matrix

As the test accuracy are all equal, the confusion matrices are also identical.



Conclusions

The success of a mission can be explained by several factors such as the launch site, the orbit and • especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success. The orbits with the best success rates are GEO, HEO,SSO,ES-L1. • Depending on the orbits, the payload mass can be a criterion to take into account for the success of a • mission. Some orbits require a light or heavy payload mass, but generally low weighted payloads perform better than the heavy weighted payloads. With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is • the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data. For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy • between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

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!