IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Juan Ricardo Ortiz
2 December 2024

Outline

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

Executive Summary

This capstone project demonstrates the application of **data science methodologies** to address a real-world problem in aerospace technology. The project focuses on developing a **predictive model** to determine the likelihood of SpaceX's Falcon 9 first stage successfully landing. Key steps include:

1. **Data Manipulation and Analysis**:
   - Leveraging Python and Pandas to process and analyze data.
   - Converting JSON files into structured Pandas DataFrames.
2. **Model Development**:
   - Employing advanced data science techniques to train and evaluate a predictive model.
   - Utilizing insights from historical data to enhance model accuracy.
3. **Knowledge Sharing**:
   - Documenting the entire workflow in Jupyter Notebooks.
   - Publishing sharable and reproducible analysis on GitHub.

(**Source:** IBM Data Science Professional Certificate, Coursera.)

This project highlights the integration of programing, data analysis, and machine learning to extract actionable insights and solve complex challenge

Introduction

This project represents the culmination of months of learning advanced theories and applying software to address complex challenges from a Data Science perspective.

Background: SpaceX advertises its Falcon 9 rocket launches at a cost of $62 million, significantly lower than competitors, who charge upwards of $165 million per launch. This substantial cost reduction is largely attributed to SpaceX's ability to reuse the rocket's first stage. Accurately predicting whether the first stage will successfully land is critical, as it directly influences the cost efficiency of each launch (**Source:** IBM Data Science Professional Certificate, Coursera.)

Goal: Develop a predictive model to determine if SpaceX's Falcon 9 first stage will successfully land.

Significance: Insights from this analysis can empower potential competitors to strategically bid against SpaceX for rocket launch contracts, leveraging the predicted outcomes to optimize their offers (**Source:** IBM Data Science Professional Certificate, Coursera.).

Section 1

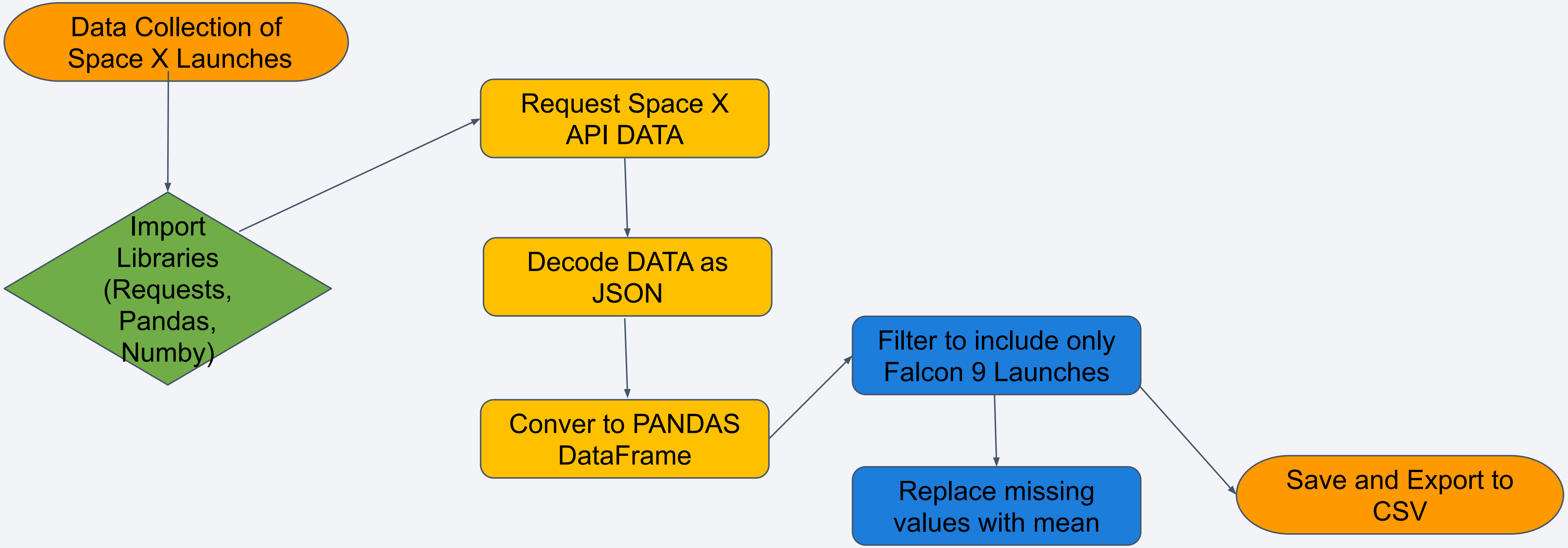
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Rest API from Space X API and Web Scraping from Wikipedia to convert a JSON file & HTML table to a Panda's DataFrame
- Perform data wrangling
  - Utilized Pandas and Numpy to explore the data and define the training label 'Class', which classifies landing outcomes as successful or unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used sklearn to standardize data, split into training and test sets, optimize hyperparameters for SVM, Classification Trees, and Logistic Regression, and selected the best model.

Data Collection – SpaceX API



Data Collection of Space X Launches

Import Libraries (Requests, Pandas, Numby)

Request Space X API DATA

Decode DATA as JSON

Conver to PANDAS DataFrame

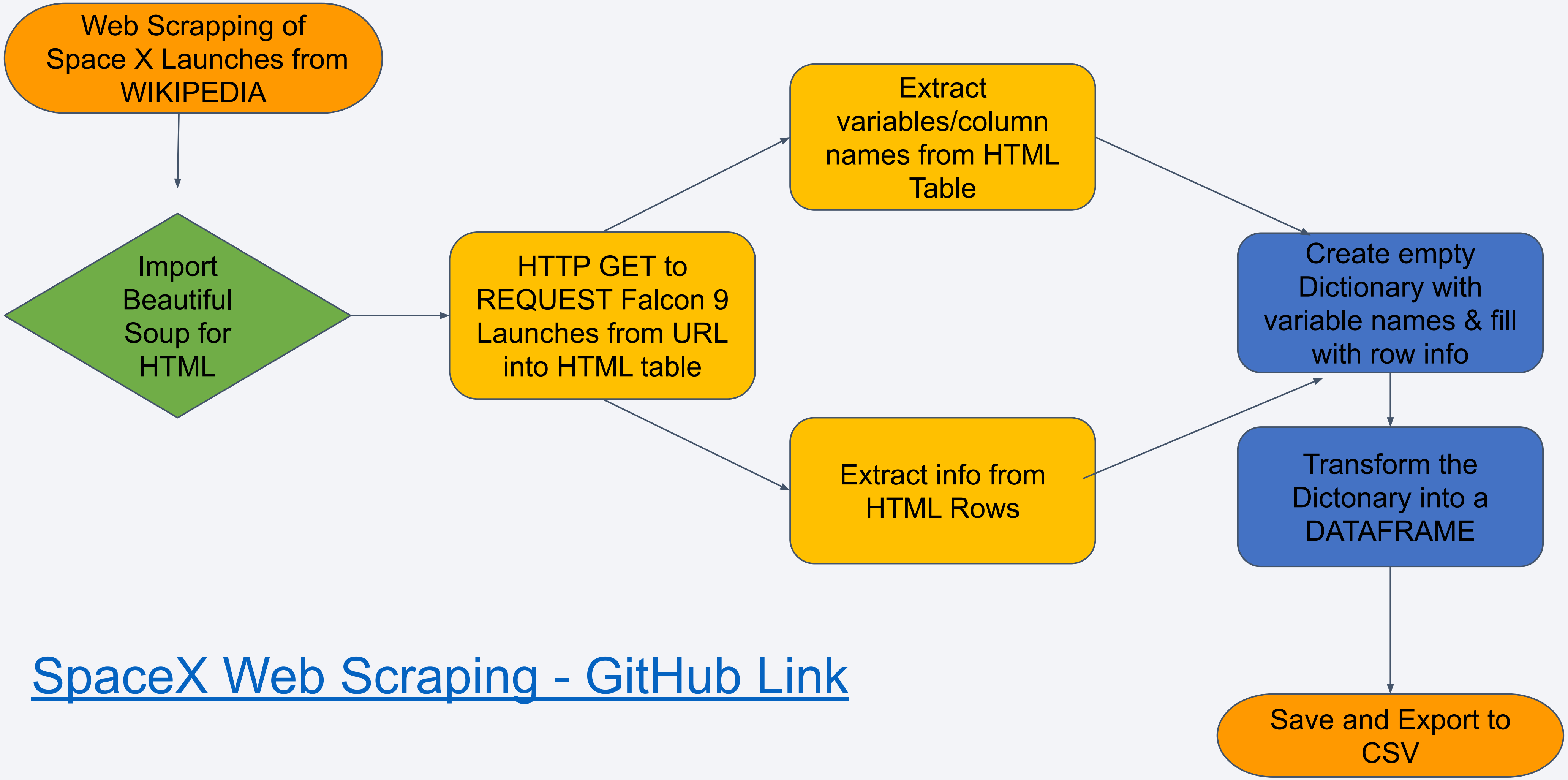
Filter to include only Falcon 9 Launches

Replace missing values with mean

Save and Export to CSV

[SpaceX API Calls - GitHub link]

Data Collection - Scraping



Web Scrapping of Space X Launches from WIKIPEDIA

Import Beautiful Soup for HTML

HTTP GET to REQUEST Falcon 9 Launches from URL into HTML table

Extract variables/column names from HTML Table

Extract info from HTML Rows

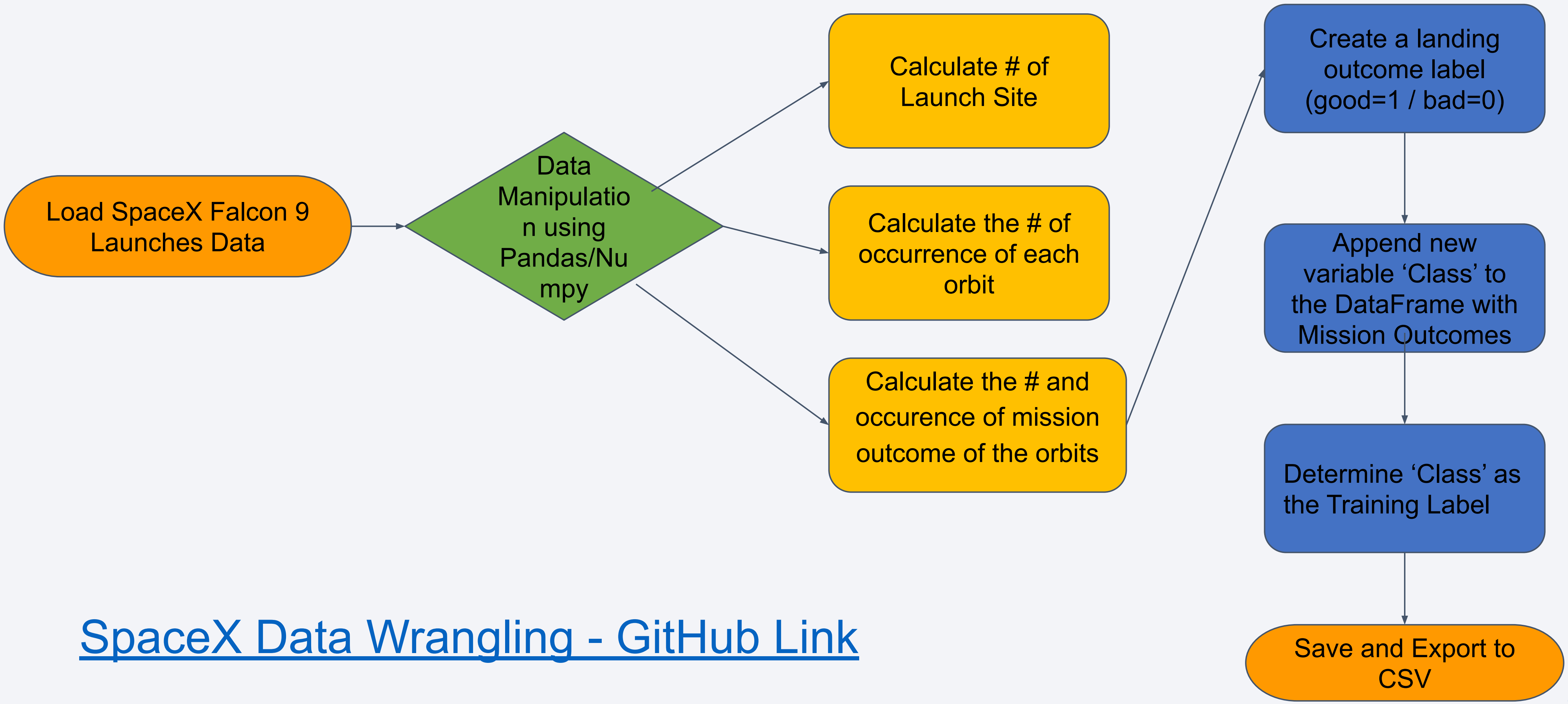
Create empty Dictionary with variable names & fill with row info

Transform the Dictonary into a DATAFRAME

Save and Export to CSV

SpaceX Web Scraping - GitHub Link

Data Wrangling - Label Mission Outcomes



Load SpaceX Falcon 9 Launches Data

Data Manipulation using Pandas/Numpy

Calculate # of Launch Site

Calculate the # of occurrence of each orbit

Calculate the # and occurence of mission outcome of the orbits

Create a landing outcome label (good=1 / bad=0)

Append new variable ‘Class’ to the DataFrame with Mission Outcomes

Determine ‘Class’ as the Training Label

Save and Export to CSV

SpaceX Data Wrangling - GitHub Link

EDA with Data Visualization - Types of Plots Used

- **Scatter Plot:** used to visualize the relationship between different quantitative variables, and the success of a SpaceX launch.E.g.
    - In a Payload Vs. Launch Site scatter point we find there are VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000)
- **Bar chart:** Intuitive way to show exact percentages for each category, allowing for easy comparison across categories. Used to visualize the success rate for each orbit type. Findings:
    - Orbits ES-L1, GEO, HEO, and SSO have the highest success rates
- **Line chart:** Used to understand trend overtime. Used to observe the success rate of launch outcomes from 2010 to 2020. Findings:
    - You can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

[EDA with Data Visualization- GitHub Link]

EDA with SQL - Slide 1

- **Displayed unique launch site names:** %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
- **Displayed 5 records where launch sites begin with the string 'CCA':** %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
- **Displayed the total payload mass carried by boosters launched by NASA**:%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)’;
- **Displayed average payload mass carried by booster version F9 v1.1:** %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
- **Listed the date when the first succesful landing outcome in ground pad was acheived.:** %sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
- **Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:** %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

EDA with SQL - Slide 2

- **Listed the total number of successful and failure mission outcomes:** %sql SELECT Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY Mission_Outcome;
- **Listed the names of the booster_versions which have carried the maximum payload mass. Use a subquery:** %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT **MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);**
- **Listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015:** %%sq SELECT substr(Date,6,2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE substr(Date,0,5) = '2015' AND Landing_Outcome LIKE 'Failure (drone ship)';
- **Ranked 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 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;

EDA with SQL- GitHub Link

Build an Interactive Map with Folium

Map Objects Utilized:

- **Markers:** used to visually locate the geographical location of a point of interest on the map. Used it for all the locations, most notably for the starting location NASA Johnson Space Center and Launch Sites.
- **Circles:** makes a point in the map stand out and facilitates its visual location. These circles where used on launch sites.
- **MousePosition:** added to the on the map to get coordinates when hovering over a point of interest on the map. Used to get coordinates for a point on the coastline, highway, railway, and city center.
- **Polylines**: used to connect two or more points on the map. Used to connect a launch site to the selected coastline point, highway, railway, and city center.
- **Distance Marker:** used to label the distance between two points connected by a polyline. E.g. distance from coastline to chosen launch site.
- **Popups:** Display information when clicking on a marker or feature on the map. Used for the NASA Johnson Space Center and Launch Sites..
- **Marker Cluster:** Marker clusters can be a good way to simplify a map containing many markers having the same coordinate. Created a market cluster for each launch records for each launch site.

Folium- GitHub Link

Build a Dashboard with Plotly Dash

Plots/Graphs and Interactions:

- **Launch Site Drop Down Selector**: added to allow for the selection of all launch sites or just a particular one, updating the pie chart and scatter plot.
- **Pie Chart:** added to display the total count of successful launches for all sites or success vs. failure counts for a selected site. Data can be updated dynamically based on the selected site on the dropdown or by clicking on the legend.
- **Payload Range:** allows the user to control the content of the Scatter plot by filtering the data based on payload range. This enables focusing on particular ranges as the business question changes/evolves.
- **Scatter Plot (Payload vs. Launch Outcome):** Allows the user to visually establish relationship between the Plots payload mass (x-axis) against launch success (y-axis) with points color-coded by booster version. It updates dynamically based on both the selected site (dropdown) and the payload range (slider).

Plotly Dashboard Code- GitHub Link
Plotly Dashboard Outcome - Guithub Link

Predictive Analysis (Classification)

Import Data & Libraries

- Import Data
- We import the sklearn library for python to:
    - Facilitate data preprocessing,
    - Data splitting (test/train)
    - Data standardization
    - selecting the different ML classification algorithms (Logistic Regression, SVM, Decision Tree, KNN).

Prepare Data

- Select the launch outcome variable ‘Class’ as the dependent variable Y and save it as a Numpy Array
- Select the independent variables, save them as X, and standardize its data using the preprocessing.StandardScaler()
- Split the data into training and test data

Modeling & Prediction

- Select the Machine Learning model GridSearch CV (Logistic Regression, SVM, Decision Tree, KNN).
- Define parameters
- Fit the model to the data and find the best Hyperparameters
- Determine the accuracy of the model on the validation data using the method *score* and the *Confusion Matrix*
- Draw Conclusions

Machine Learning Prediction- GitHub Link

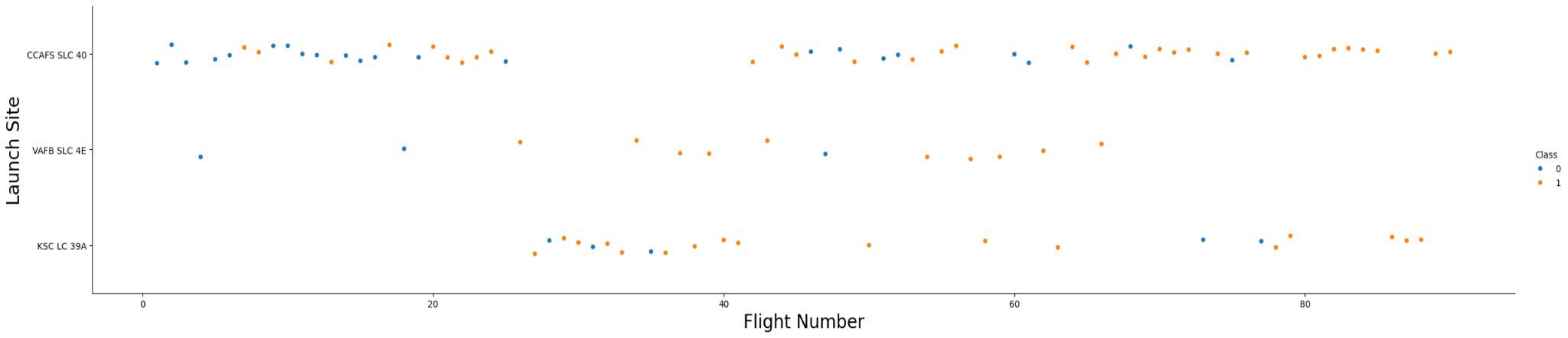
Results

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

Section 2

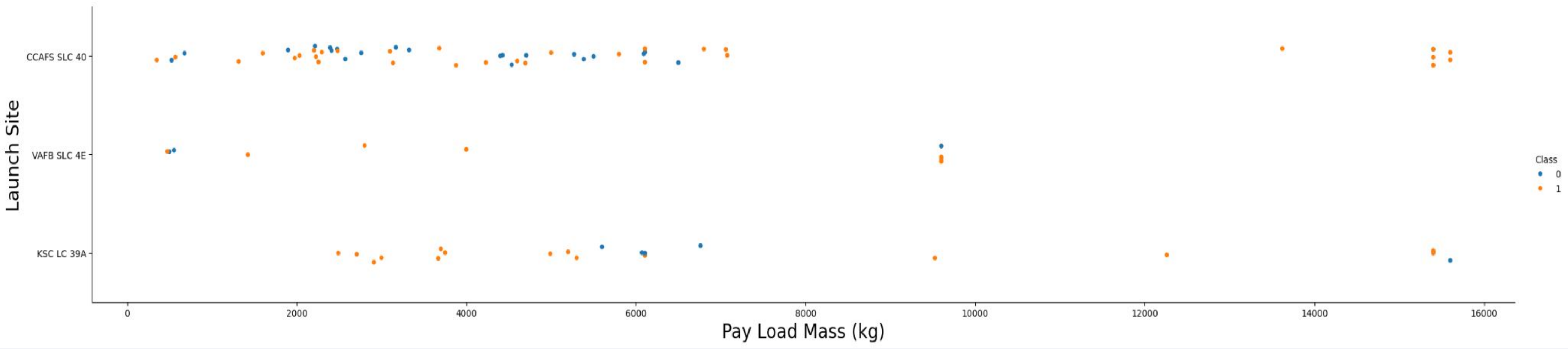
Insights drawn from EDA

Flight Number vs. Launch Site



We can observe that at higher number of flights (80+), there are more successful launches from CCAFS SLC 40 and KSC LC 39A launch sites. While launch site VAFB SLC 4E doesn’t share the same volume of flight numbers, most launches at this site are successful.

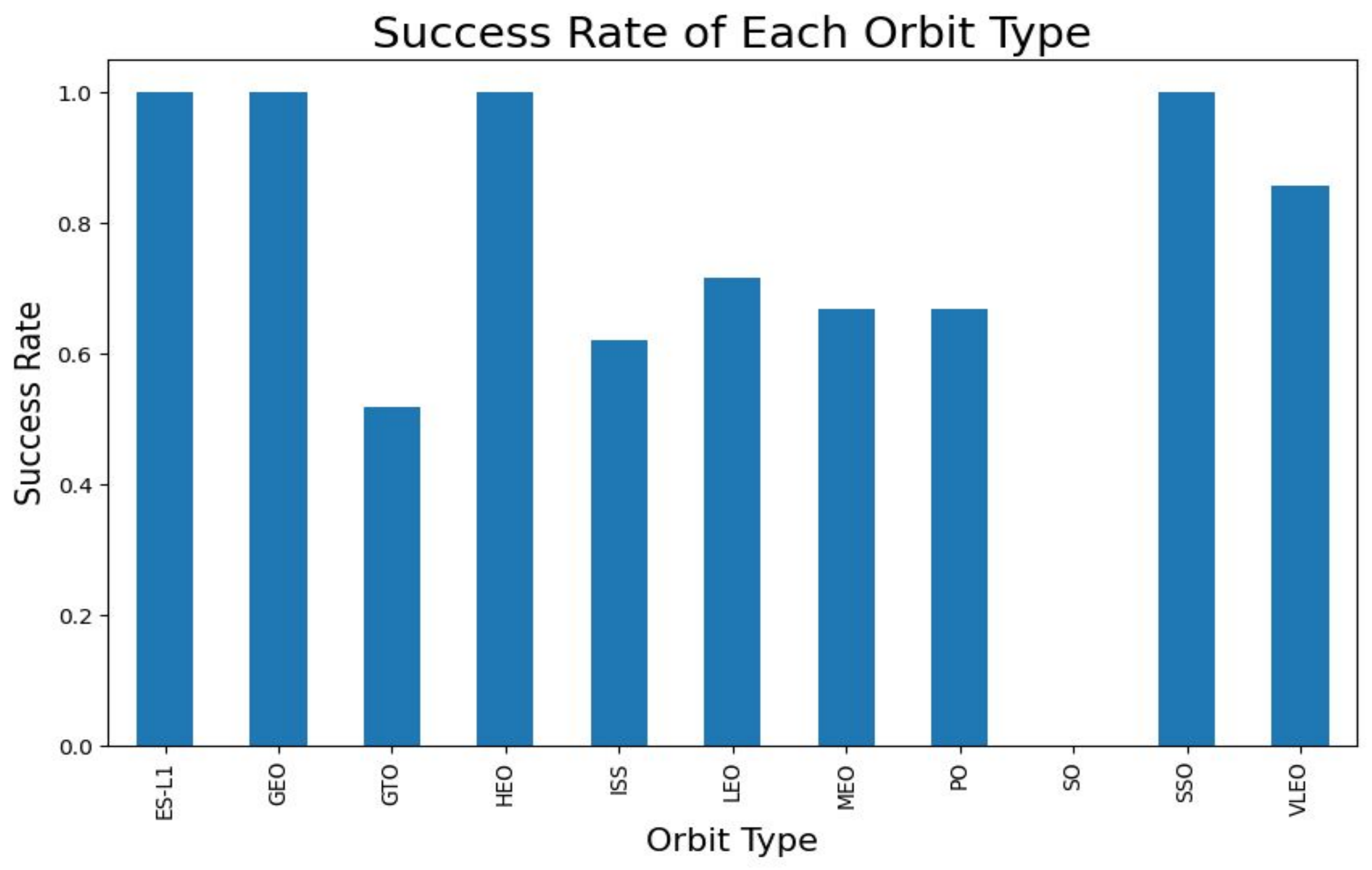
Payload vs. Launch Site



In this Payload Vs. Launch Site scatter point chart we find for the VAFB-SLC launchsite, there are no rockets launched for heavy payload mass(greater than 10000).

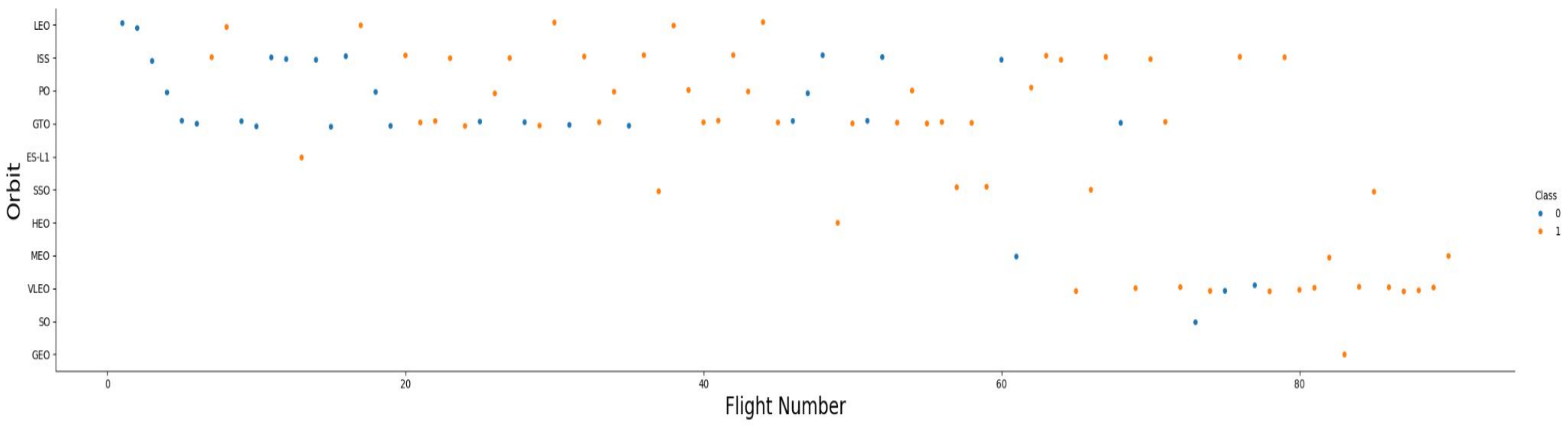
(**Source:** IBM Data Science Professional Certificate, Coursera.)

Success Rate vs. Orbit Type



We find that the launch success rate is the highest for orbit's ES-L1, GEO, HEO, and SSO.

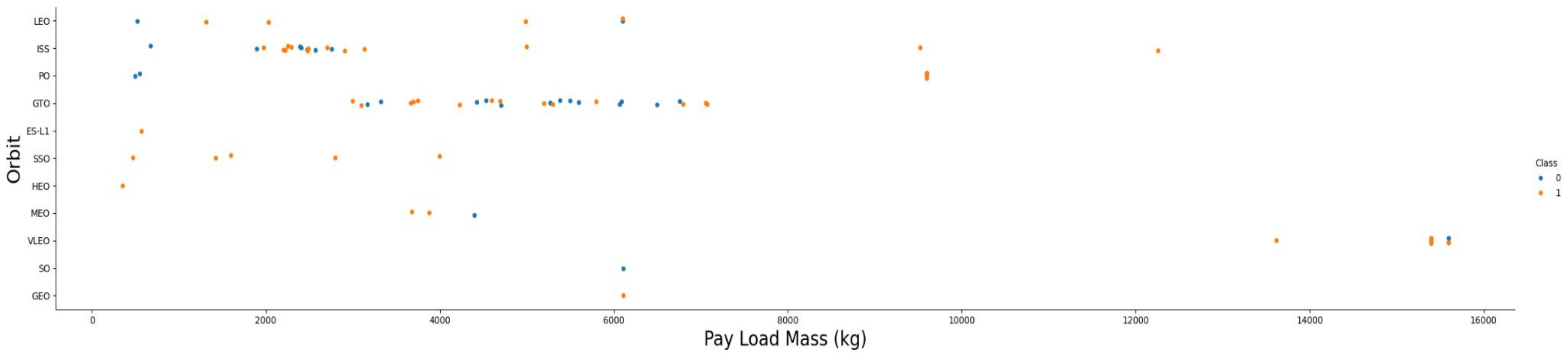
Flight Number vs. Orbit Type



You should see that in 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.

(**Source:** IBM Data Science Professional Certificate, Coursera.)

Payload vs. Orbit Type

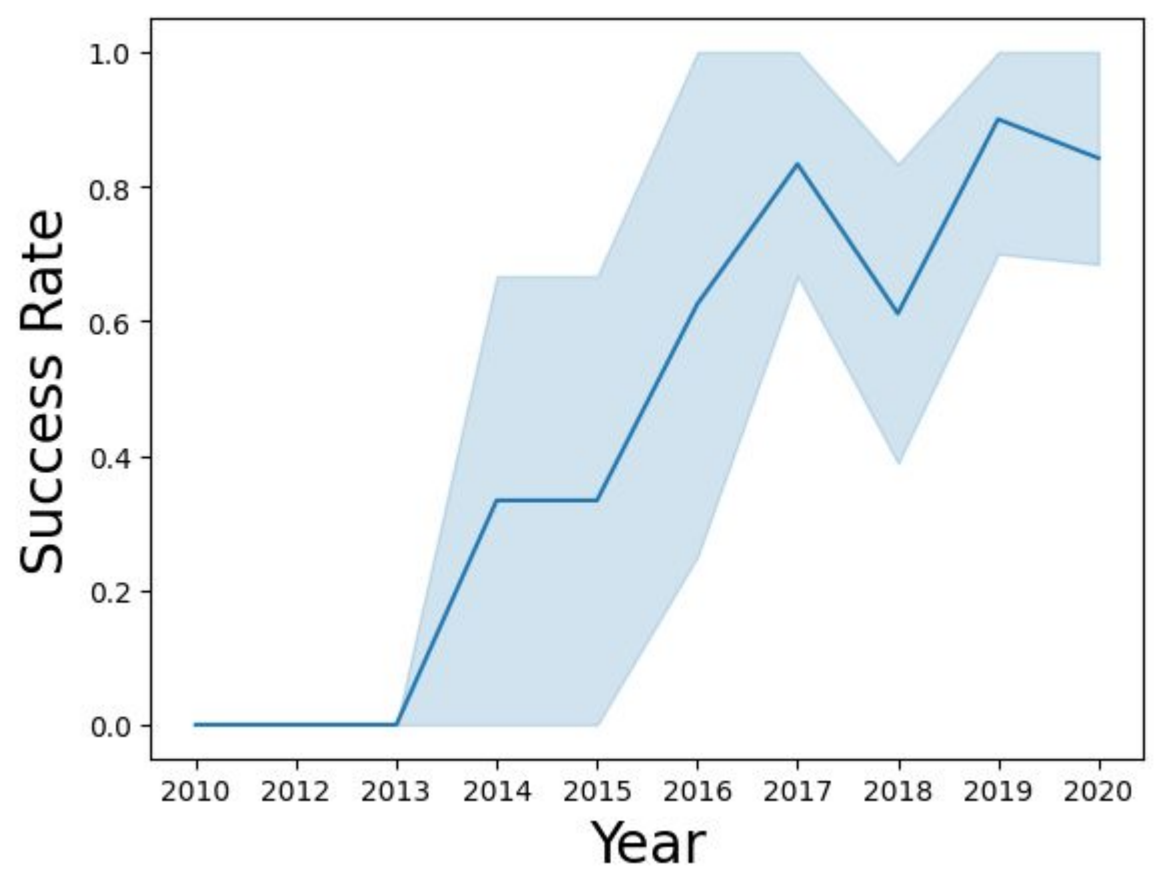


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

(**Source:** IBM Data Science Professional Certificate, Coursera.)

Launch Success Yearly Trend

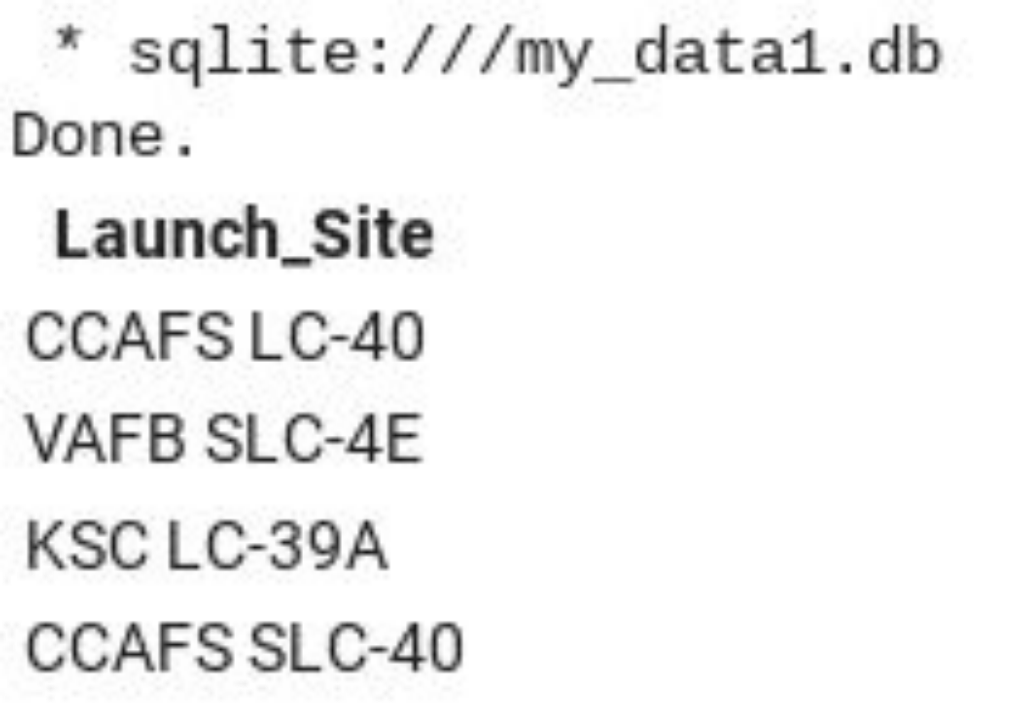


We observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

(**Source:** IBM Data Science Professional Certificate, Coursera.)

All Launch Site Names

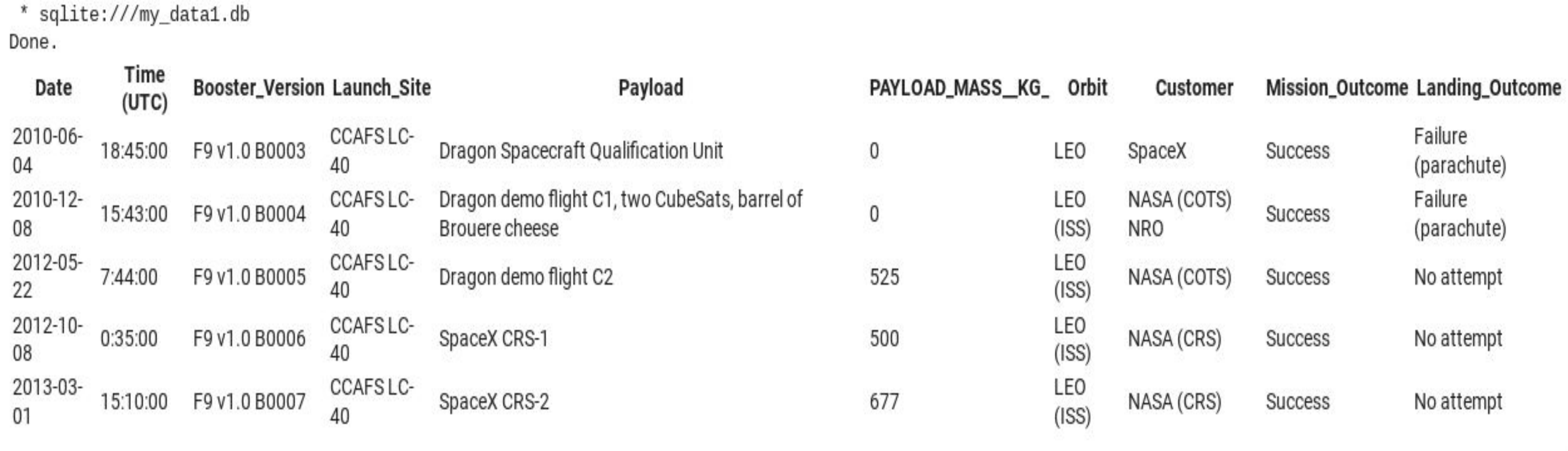
We used the DISTINCT keyword in SQL to retrieve the unique launch site names from the table



```
 * sqlite:///my_data1.db
Done.
 Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

We used the LIKE keyword to find launch site names that begin with `CCA%’ and added the LIMIT keyword to only provide 5 record

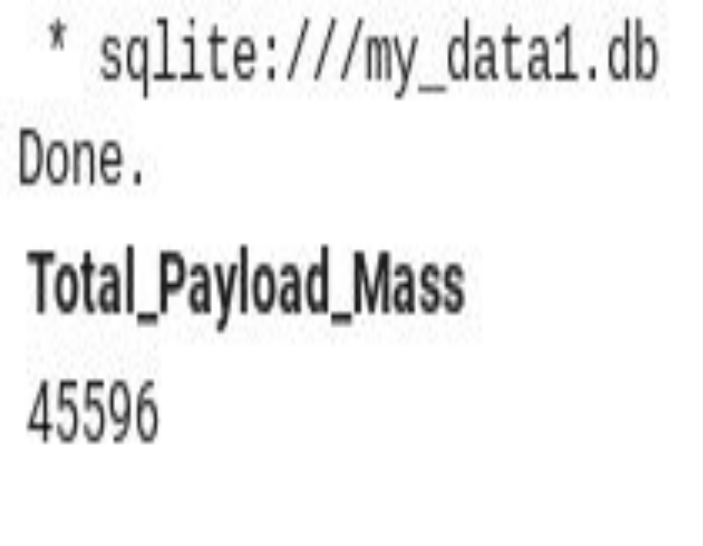


```
 * sqlite:///my_data1.db
Done.
```

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

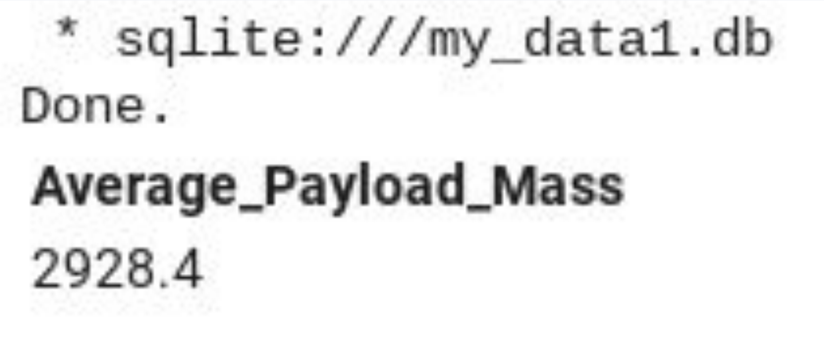
Calculate the total payload carried by boosters from NASA by using the SUM function on the Payload_Mass_Kg column and AS keyword to label the output as Total_Payload_Mass



```
 * sqlite:///my_data1.db
Done.
Total_Payload_Mass
45596
```

Average Payload Mass by F9 v1.1

Calculated the average payload mass carried by booster version F9 v1.1 by using the AVG function on the Payload_Mass_KG column and detailed in the WHERE clause to only include rows in the Booster_Version column that had the F9 v1.1 name. The AS keyword was used to add the Average_Payload_Mass label to the output.

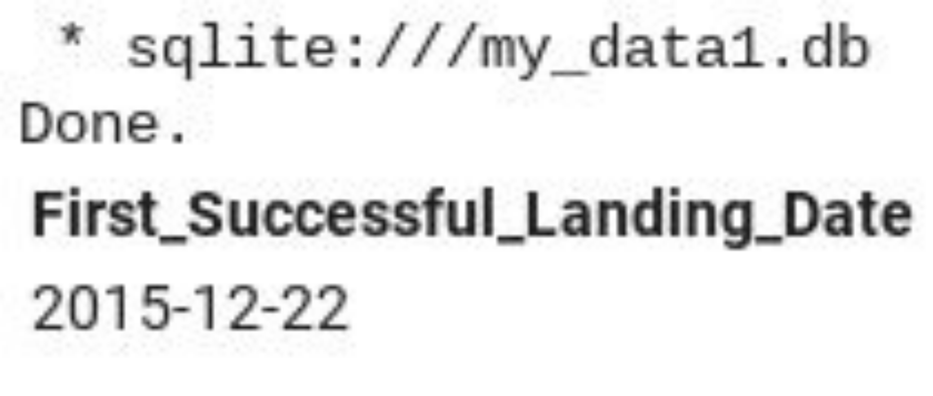


```
 * sqlite:///my_data1.db
Done.
```

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

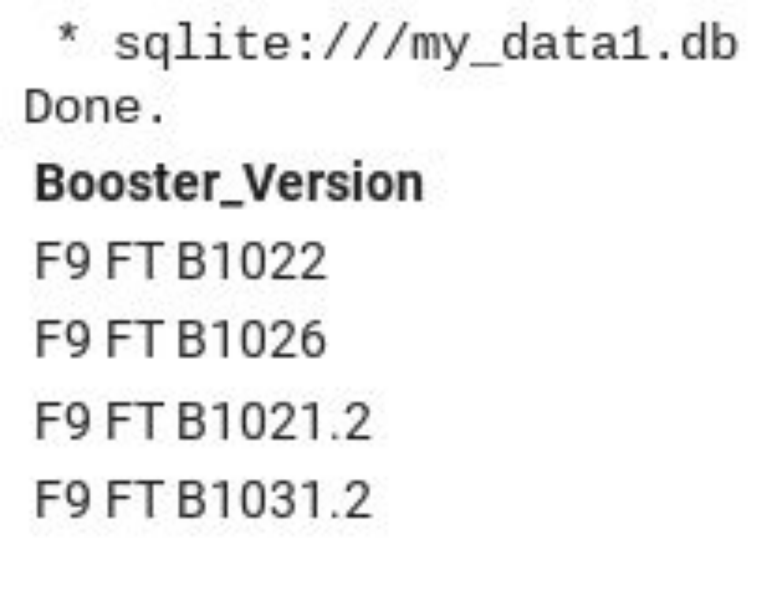
In the SELECT clause the MIN function was used on the DATE column and in the WHERE clause we only included landing outcomes that were successful. In this way, the first successful ground landing date was found and labeled as such with the AS keyword.



```
 * sqlite:///my_data1.db
Done.
First_Successful_Landing_Date
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

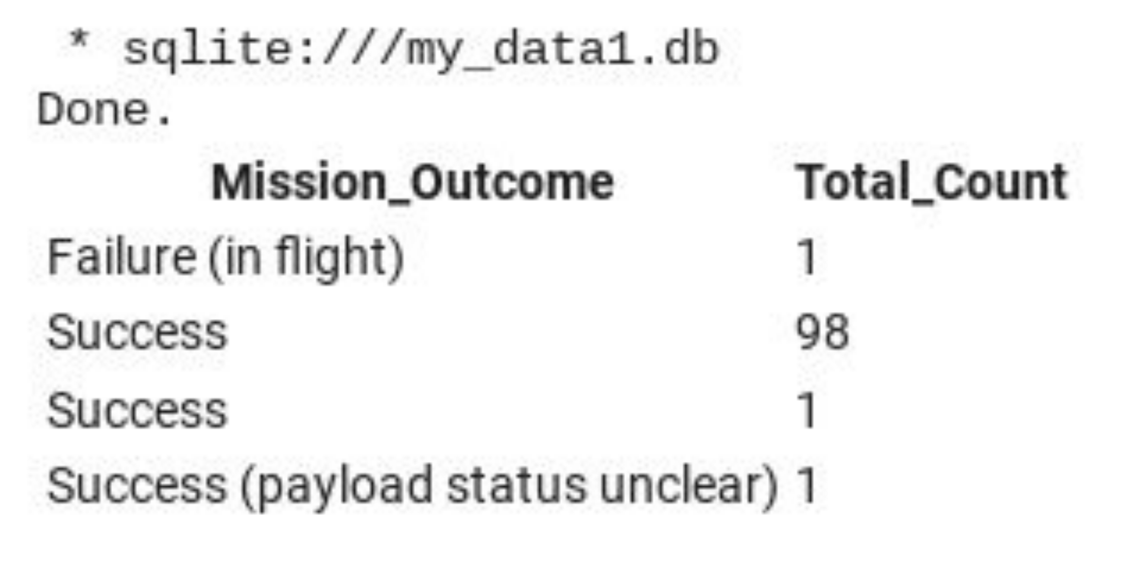
In the WHERE clause, conditions were added to sort the data. The landing outcome column was set to include only rows with successful outcomes and the payload mass column was set to only include payloads between 4000 and 6000. This enabled to obtain the booster version of those Drone Ships that met those conditions



```
 * sqlite:///my_data1.db
Done.
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2
```

Total Number of Successful and Failure Mission Outcomes

To obtain the below query result, we had to use the GROUP BY and COUNT function to arrange the data into successful and failed missions.

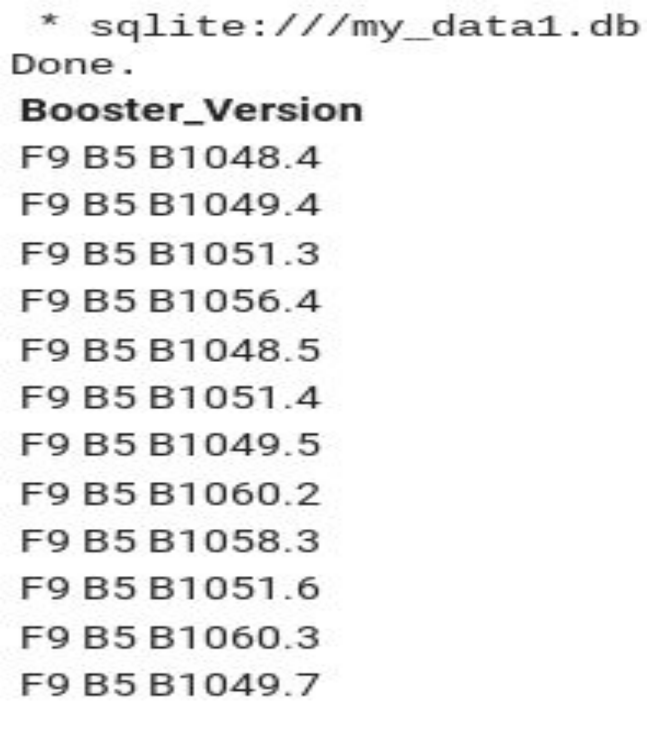


```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Total_Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

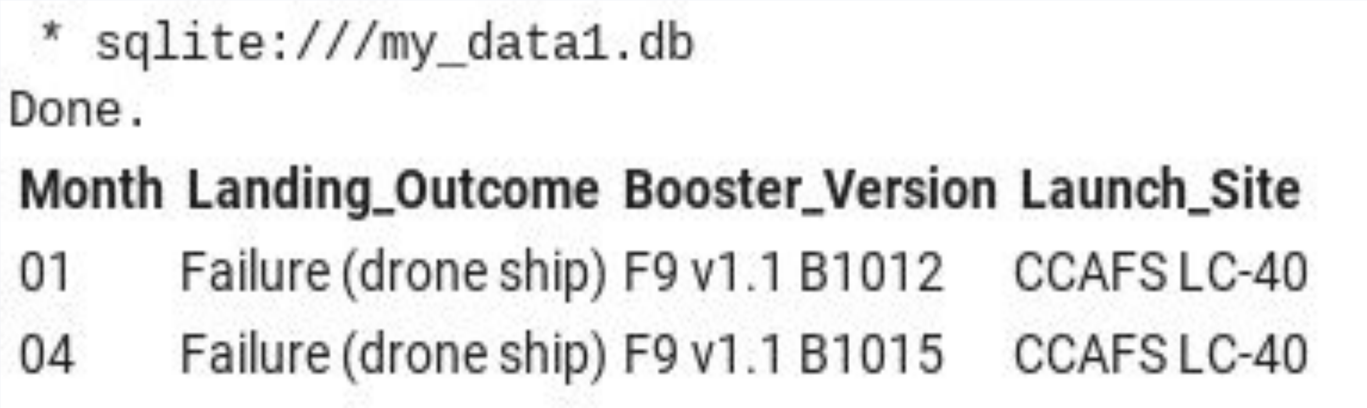
To obtain this query result, we had to use a subquery with the MAX function to find the max payload to then select the Booster_Versions which carried it.



```
 * 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
 F9 B5 B1060.2
 F9 B5 B1058.3
 F9 B5 B1051.6
 F9 B5 B1060.3
 F9 B5 B1049.7
```

2015 Launch Records

For the Launch Records we included information from failed landing_outcomes in drone ship, their booster versions, and launch site names column for the year 2015

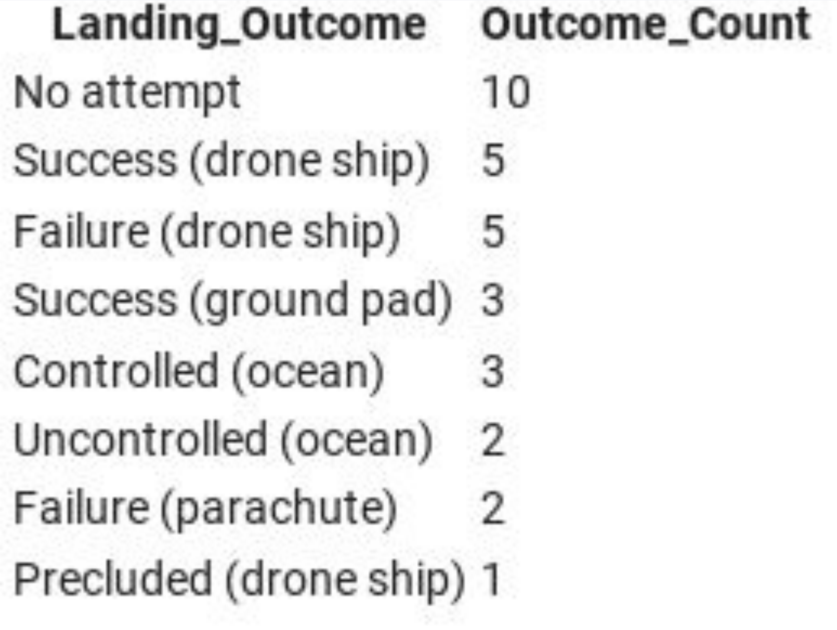


```
 * sqlite:///my_data1.db
Done.
```

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

To 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, the data had to be grouped by the Landing Outcome and the Outcome Count (calculated with the COUNT function) in descending order.



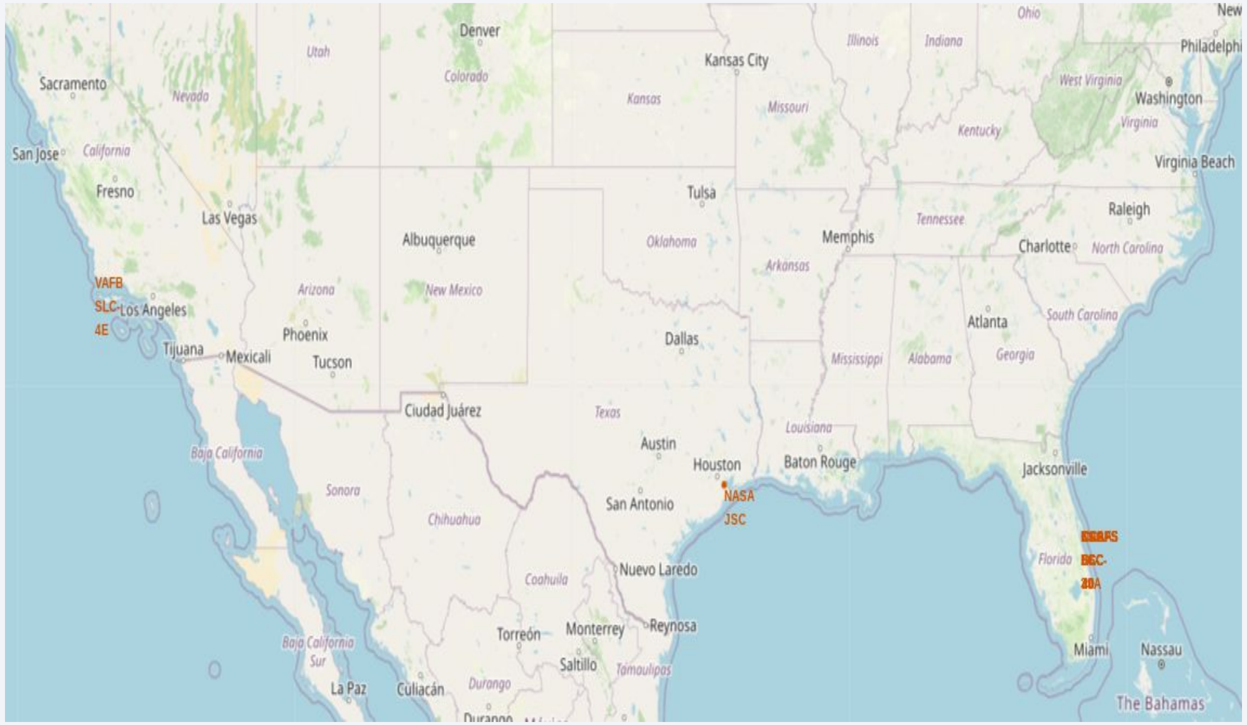
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

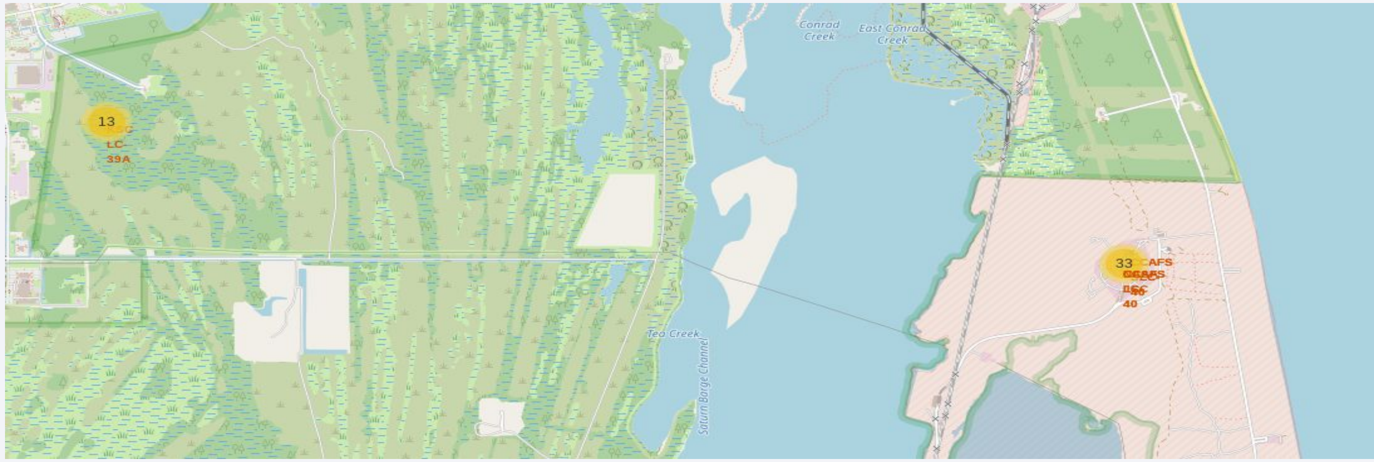
Launch Sites’ Location - Global View

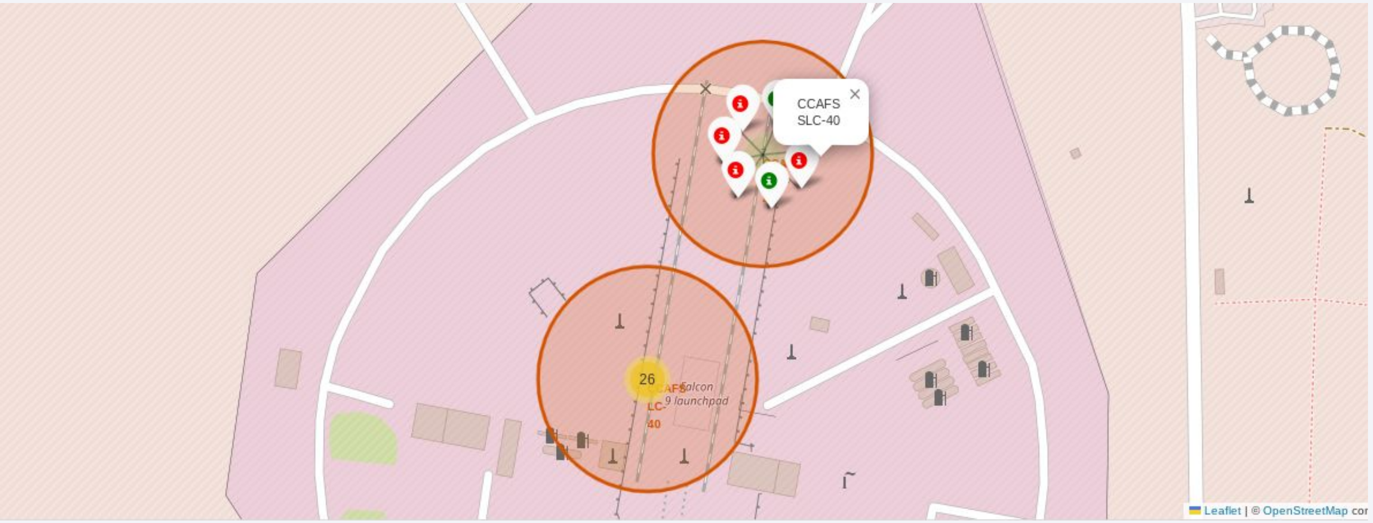
- Initial center location on the map is the NASA Johnson Space Center in Houston, Texas.
- There are a total of 4 launch site locations clearly marked
- Most Launch Sites are located in the East Coast. Only one is located in the West coast
- Without zooming in, it is difficult to see all the Launch sites in the East Cast since they are located very close to each other.



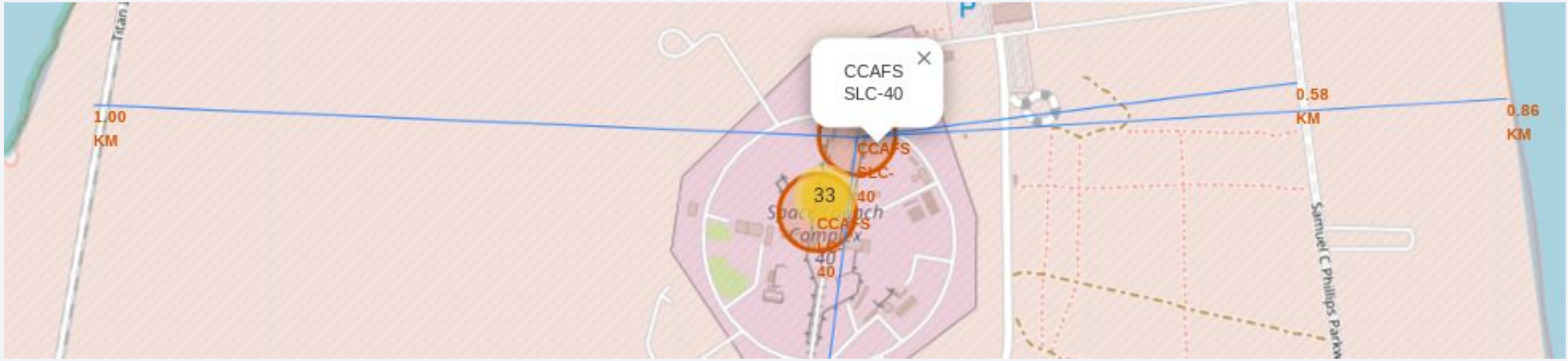
Launch Sites’ Outcomes

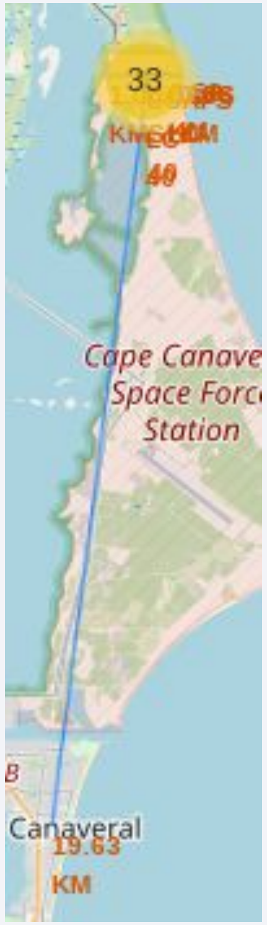
- This map contains the total number of launches per launch site as well as the outcomes, with green being successful and red unsuccessful
- Since the markercluster object was used, the view of the east coast launch sites is less cluttered as you zoom in.
- The total number of launches can be seen in the yellow circle.
- After clicking on the circle, the total number of launches is broken down further by outcome,
- In the case of launch site CCAFS SLC-40, you can see there’s a total of 7 launches, 3 of which successful while 4 of them failed.





Launch site CCAFS SLC-40 distance to its proximities

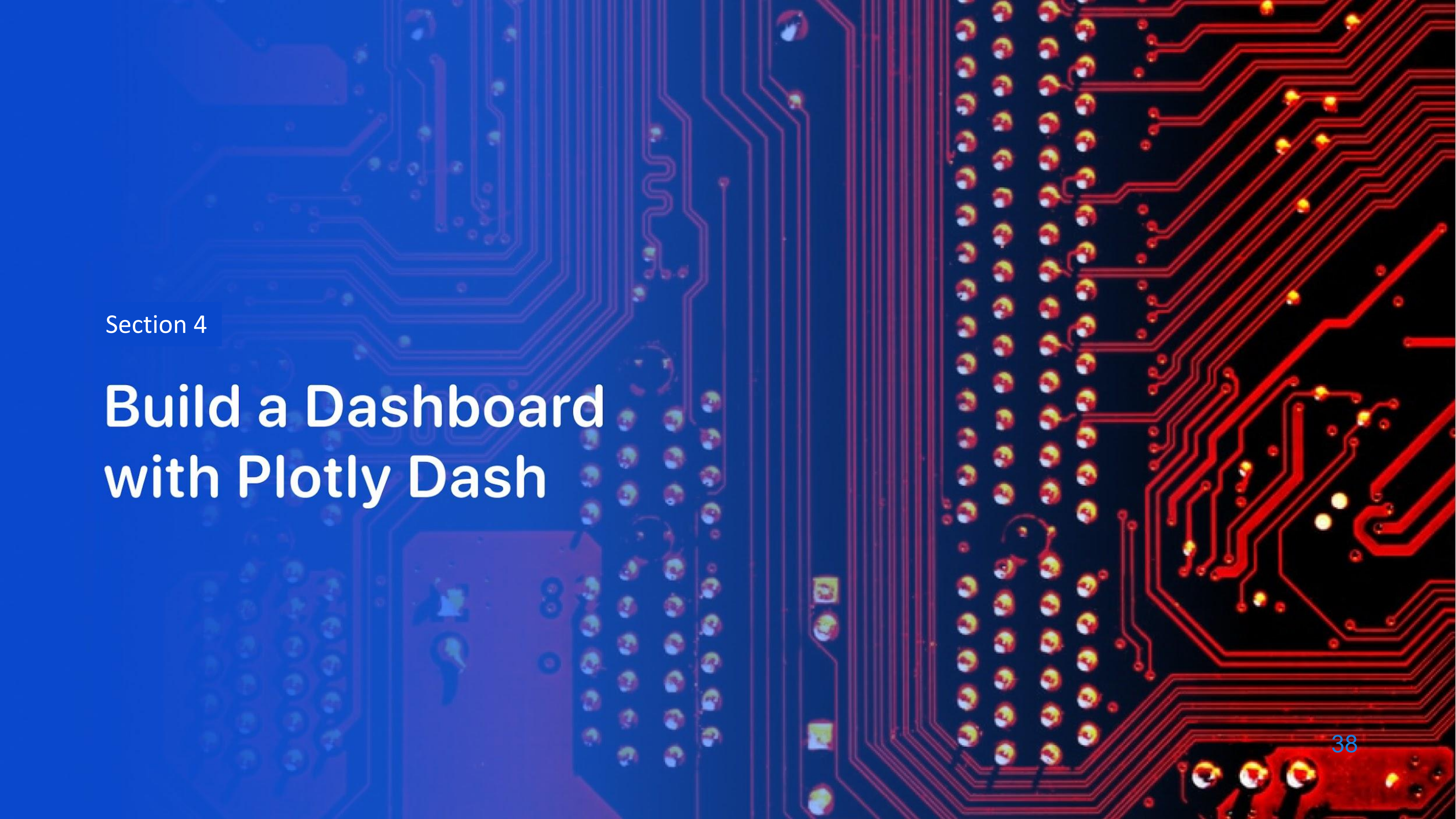




In this map, a *mouse_pad* and *distance* function enabled us to get coordinates and distance from CCAFS SLC-40 to the closest coastline, highway, railway, and city center.

Based on findings in CCAFS SLC-40 launch site, we can can extrapolate these results to infer that launch sites maintain close proximity to railways and highways to facilitate the transportation of the Falcon 9 Rocket to the site. In addition, they are close to the coastline so the water dampens any falling debry in case the launch fails or there are other problems along the rocket's flight path. As for the distance from city centers, the launch sites seem to be maintaining a 'safe' distance from city centers to perhaps avoid inconveniencing their citizens with the logistics of transportation, noise pollution, overcrowding with visitors, etc.

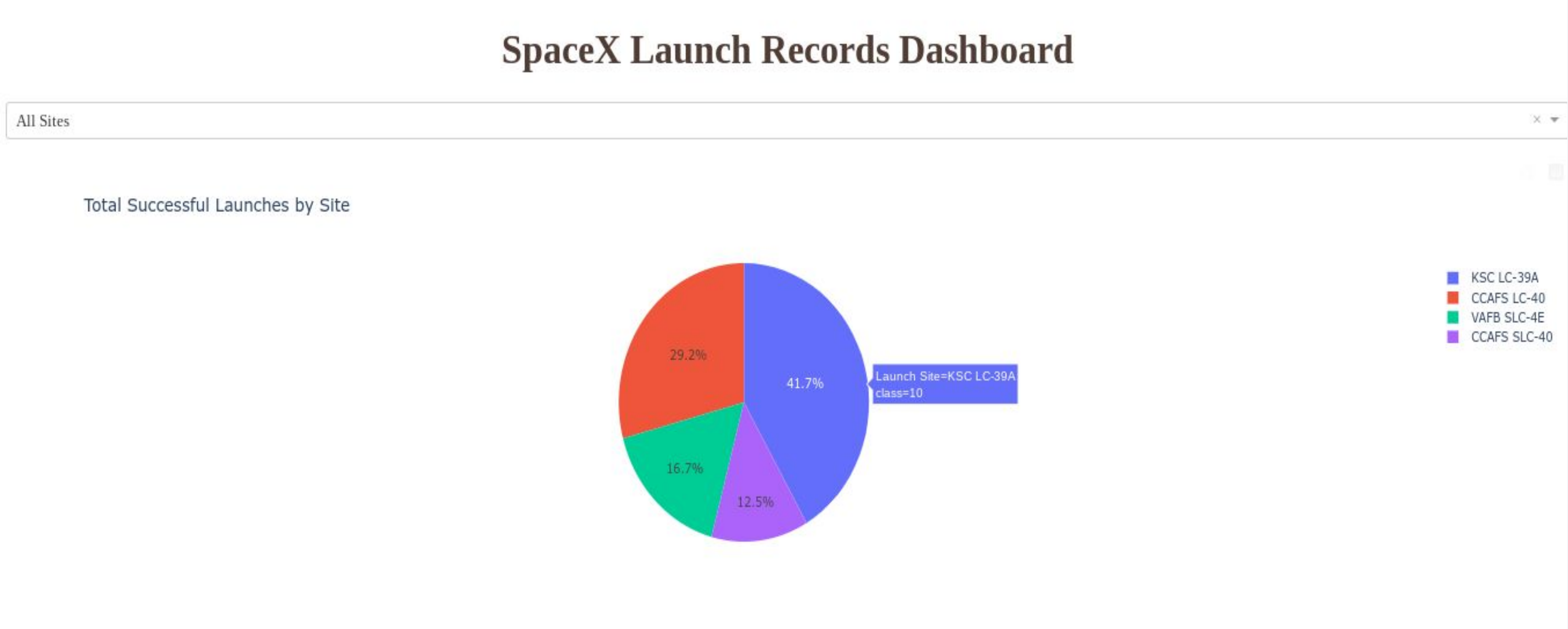
[Folium Launch Site Locations- GitHub Link]



Section 4

Build a Dashboard with Plotly Dash

Successful Launches by Site



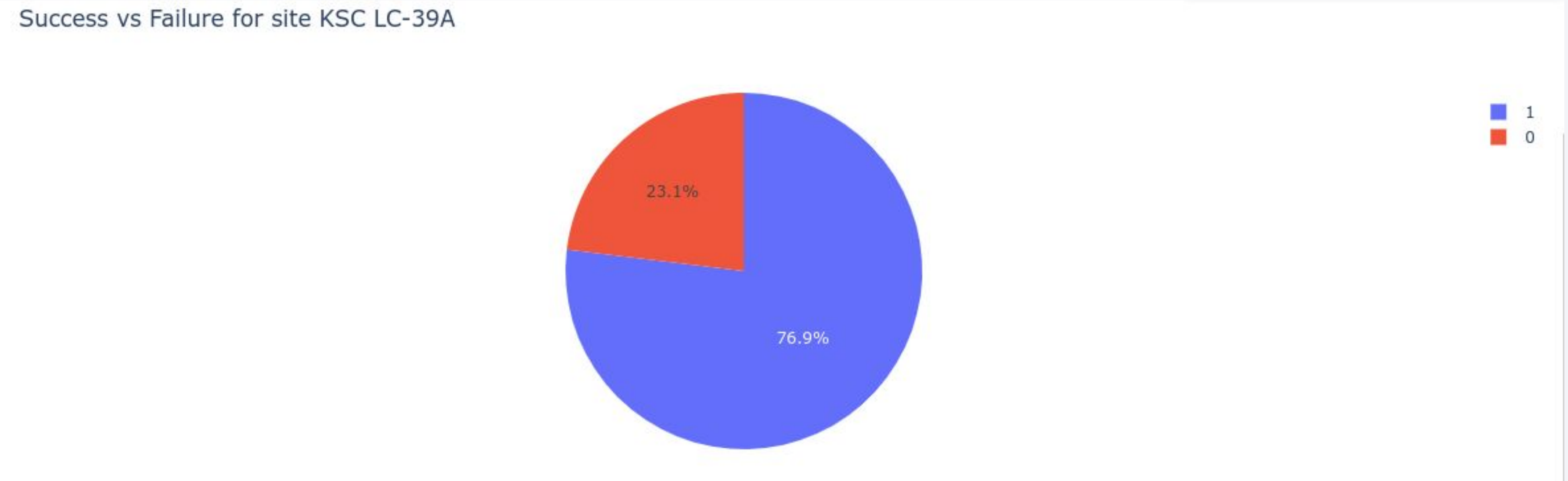
SpaceX Launch Records Dashboard

All Sites

Total Successful Launches by Site

44.7% of all successful launches come from KSC LC-39A site, making it the most successful.

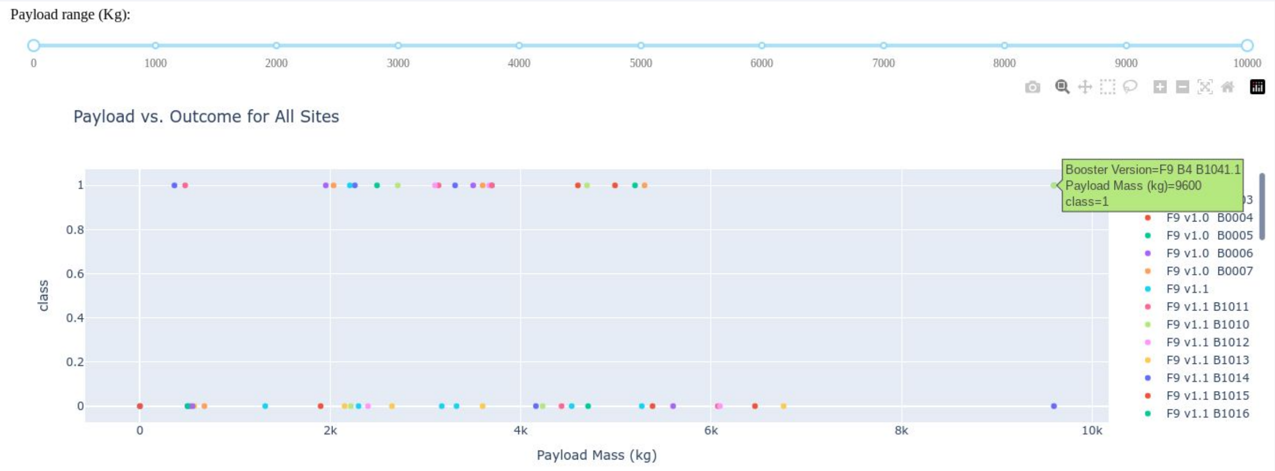
Most Successful Launch Site’s Success Rate

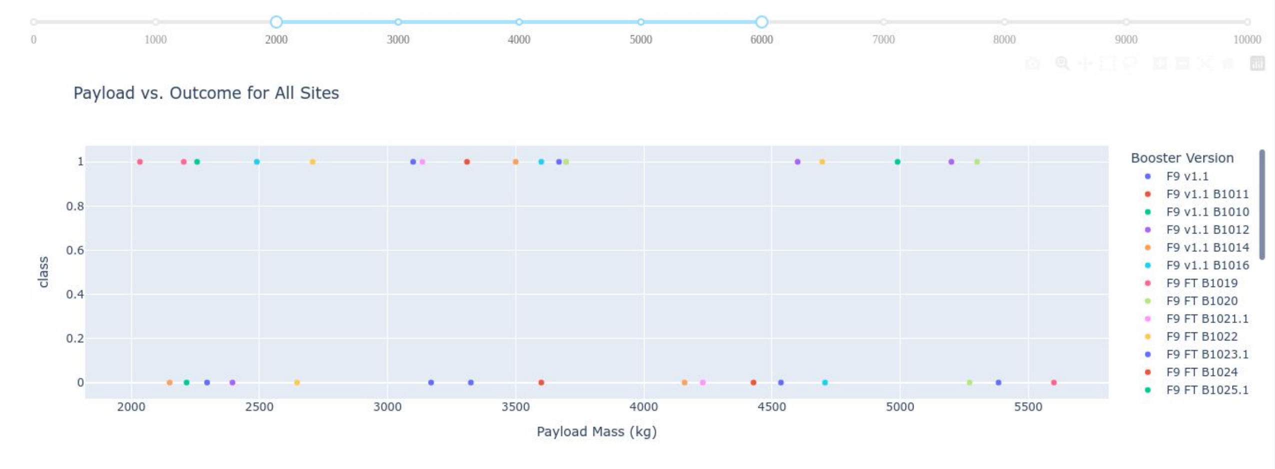


76.9% of all launches in KSC LC-39A are successful, while 23.1% are unsuccessful. Using plotly, we are able to close in on a launch site through the filter and dynamically update the pie chart.

Boosters’ Success Rate based on Payload

- At Full Payload Range Selected , we can see that:
    - Only one Booster Version can succeed at a high payload (9,600 KG)
    - Most boosters operate with a payload between 2000 and 6000 KG
- In the 2000 to 600 KG payload range, there’s a higher concentration of booster versions that operate and have both successful and unsuccessful outcomes
- The Scatter plot dynamically updates based on the site dropdown and payload range slider. It also has pop-ups when selecting a data point to get more detailed information.



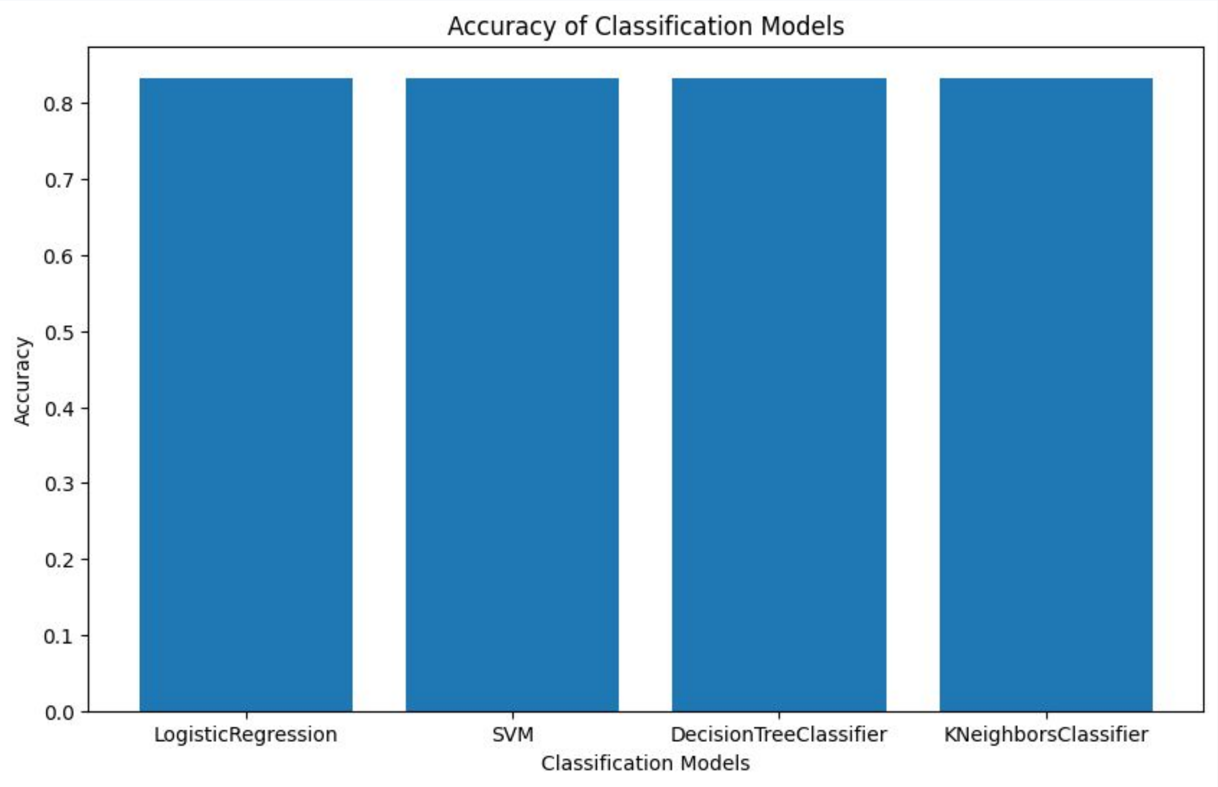


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- We can observe that ALL classification models have the same test accuracy of **0.8333333333333334.**
- While performing a line of code to return the model with the best test accuracy, the outcome is **logistic regression** since it’s the first key the code encountered.



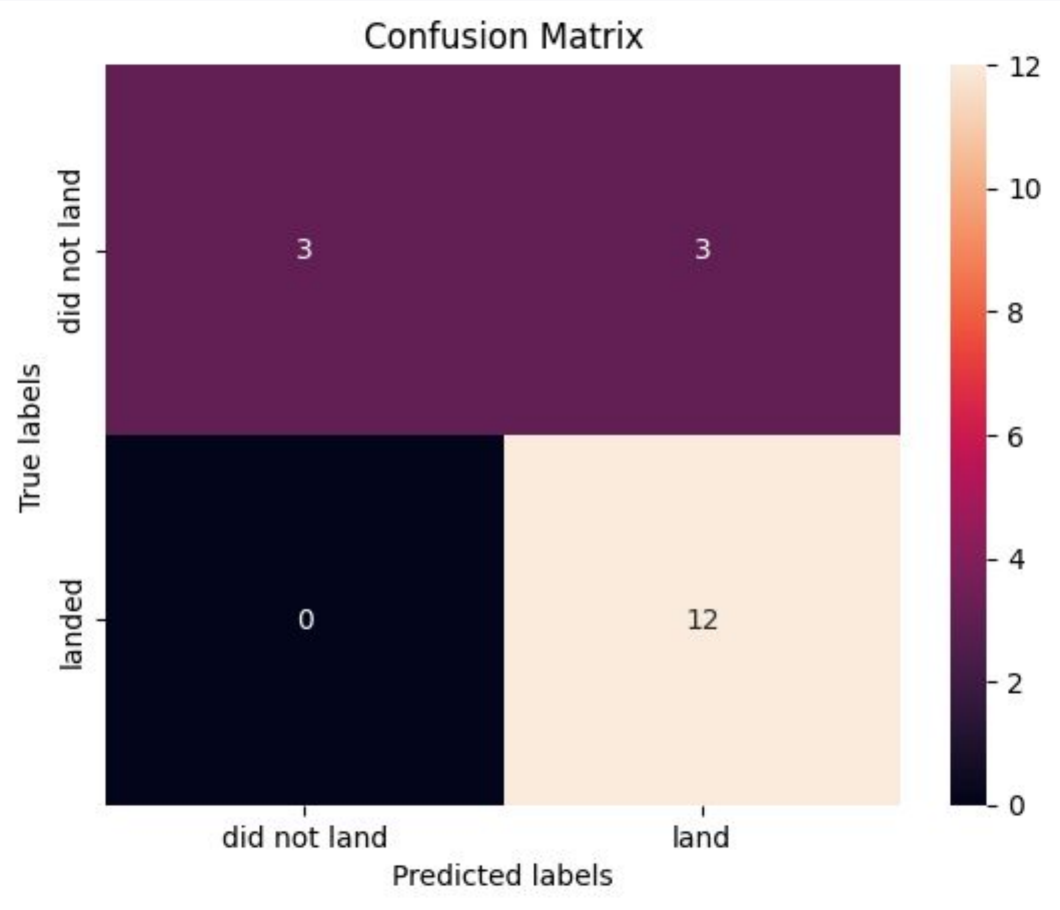
Confusion Matrix

We see that logistic regression (as well as all other models) can distinguish between the different classes.

However, there is a problem of false positives.

- **True Postive** - 12 (True label is landed, Predicted label is also landed)
- **False Postive** - 3 (True label is not landed, Predicted label is landed)

(**Source:** IBM Data Science Professional Certificate, Coursera.)



Conclusions

- **The features** payload mass, launch site, and orbit type, significantly influence the likelihood of a successful landing.
- The **Hyperparameter Tuning and Data Standardization improved model convergence and performance.** GridSearchCV for hyperparameter tuning improved model performance.
- Splitting the data into training and testing sets allowed for the evaluation of model generalization, ensuring that the models are able to perform well on unseen data.
- While all models had an **identical accuracy in predicting landing outcomes**, a Decision Tree Classifier may be considered the best model for practical reasons due to its easy visual interpretation and computational ease.
- The model has a problem of **false positive**, which could cause the overestimation of landing success. This in turn can create an unrealistic expectation of cost savings due to reuse.

Appendix

- Jupyter Notebooks- https://github.com/jrortiz88/capstone/tree/main
- Data Set Part 1 - https://github.com/jrortiz88/capstone/blob/main/dataset_part_1.csv
- Data Set Part 2 - https://github.com/jrortiz88/capstone/blob/main/dataset_part_2.csv
- Data Set Part 3 - https://github.com/jrortiz88/capstone/blob/main/dataset_part_3.csv
- Folium Output - https://github.com/jrortiz88/capstone/blob/main/foliumlocationoutput.pdf
- Dashboard Ouput - https://github.com/jrortiz88/capstone/blob/main/plotlydashboardoutput.pdf

Thank you!