IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dzerjinski Lemos
2024-10-19

Outline

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

Executive Summary

- Summary of methodologies
  - This project involved **data collection** through web scraping and API integration to gather SpaceX launch data, followed by **data wrangling** to clean and transform the data. **Exploratory Data Analysis (EDA)** was performed using **Pandas**, **matplotlib**, and **SQL** to explore the dataset's characteristics and create visualizations. **Dashboards and visual analyses** helped interpret key insights. Finally, **predictive analysis** was conducted using machine learning models like logistic regression and Random Forest to predict the success of Falcon 9 rocket landings.

- Summary of all results
  - Web scraping successfully retrieved detailed historical SpaceX launch data from Wikipedia, and API integration provided updated and structured data on rocket launches.
  - The collected data was cleaned by handling missing values, inconsistencies, and duplicates.
  - Using Pandas and SQL, key insights were extracted, including the frequency of successful launches and the factors affecting rocket landings. Visualizations, such as scatter plots and histograms, highlighted patterns in the data, such as the correlation between launch site and landing success.
  - Visual dashboards were created to summarize findings, such as the success rate of Falcon 9 landings over time, launch site performance, and the impact of payload mass on landing outcomes.
  - Machine learning models, including logistic regression and Random Forest, were used to predict the success of Falcon 9 first-stage landings. The models achieved good accuracy, with Random Forest performing particularly well in identifying the key features influencing successful landings, such as payload mass and launch site.

Introduction

- As SpaceX continues to pioneer reusable rocket technology, predicting and improving the success rate of these landings is critical to reducing costs and increasing the efficiency of space exploration. The available data from public sources like Wikipedia and APIs offered a rich dataset for understanding past launch outcomes, including details about launch sites, payloads, and landing attempts. This project leverages data science techniques to analyze this information and build predictive models that can inform future landing success rates and therefore estimate the launch costs.
- The key problems this project aims to address are: What are the main factors influencing the success or failure of Falcon 9 first-stage landings? Can we use historical data to reliably predict the outcome of future landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

SpaceX API

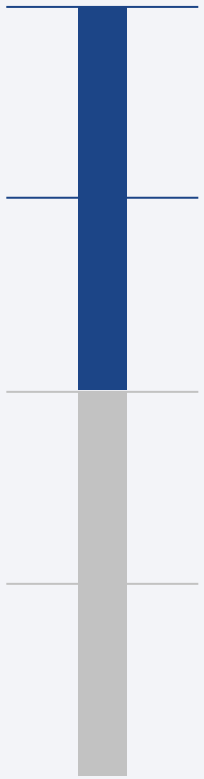
Using the SpaceX's own API , gathered up-to-date information on launches, such as detailed metrics on each mission, payload specifics, and landing attempts.

Web Scraping

Gathered historical launch data from Wikipedia, specifically from a page listing Falcon 9 and Falcon Heavy launches. BeautifulSoup were used to automate the extraction process.

Data Collection – SpaceX API

- The data collection from the SpaceX API followed a structured process to ensure efficient and accurate retrieval of launch data.
- Finally, the data from the API was saved locally for further processing and analysis.



Identifying the specific API endpoints

such as those offering details on launch histories, payloads, rockets, and landing outcomes.

API requests

using the requests library in Python, which allowed for HTTP GET requests to retrieve JSON-formatted data.

Parsing response data

into a structured format, typically as a dictionary or a DataFrame using the pandas library.

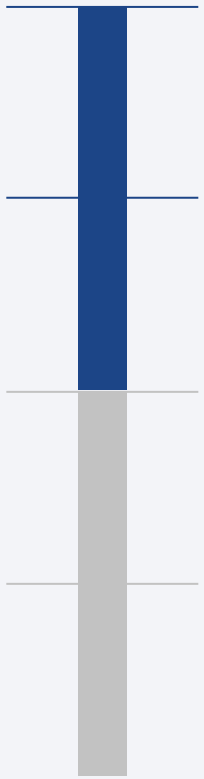
Ensure data integrity

validation checks were performed on each response, verifying the presence of required fields and handling any potential errors in API responses, such as missing data

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/jupyter-labs-spacex-data-collection-api-v2.ipynb

Data Collection – Web Scraping

- The Wikipedia webpage, provided a comprehensive list of Falcon 9 and Falcon Heavy launch records, including details on launch dates, locations, payloads, and landing results
- Finally, the data from the API was saved locally for further processing and analysis.



Download Falcon 9 Launch Wiki page

Using Python libraries like requests and BeautifulSoup, a connection was established to retrieve the HTML content of the page.

Parse response data

making it easy to navigate and extract specific elements.

Locate information tables

then further parsed to access individual rows and columns containing launch details.

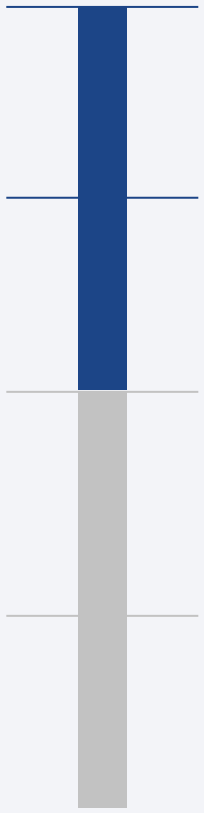
Extract rows information

such as mission name, date, site, payload, and landing outcome; and stored in a structured format, using pandas DataFrame, for easier manipulation and analysis.

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/jupyter-labs-webscraping.ipynb

Data Wrangling

- The data processing phase involved data wrangling techniques to ensure the datasets were clean, consistent, and suitable for analysis.
- The result was a clean, well-structured dataset ready for exploratory analysis and predictive modeling.



Handling missing values

by either imputing values based on related information or removing incomplete records where necessary

Identifying duplicate entries

and removing them to avoid skewing the analysis.

Categorical encoding

was applied to features like launch site and payload type, converting them into numerical values that could be used in machine learning models.

Normalization

was employed to ensure that all variables operated on similar scales, which is essential for algorithms sensitive to varying magnitudes.

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/labs-jupyter-spacex-data-wrangling.ipynb

EDA with Data Visualization

Three kinds of charts were plotted:

1. **Scatter Plots** – Used to explore the relationships between variables such as payload mass, flight number, launch site and landing success. Scatter plots helped identify patterns and correlations, offering a visual understanding of how those variables might influence landing outcomes.
2. **Bar Chart** – Employed to compare landing success rate by orbit. Bar charts made it easier to see orbits had higher success rates, offering insights into potential orbit-related factors affecting landing outcomes.
3. **Line Chart** – Used to illustrate the success rate of landings across multiple years. This visualization helped highlight trends, revealing how landing success rates improved with advancements in SpaceX technology.

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/jupyter-labs-eda-dataviz-v2.ipynb

EDA with SQL

Here is a list of performed SQL queries:

- List of all launch sites.
- First 5 launches from launch sites beginning with the string CCA.
- Total payload from NASA (CRS) launches.
- Average payload mass carried by booster version F9 v1.1.
- The list of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the number of successful and unsuccessful outcomes.
- The list of booster version which carried the maximum payload mass.
- List failed landings in 2015.
- The rank of count landing outcomes between June 2010 and March 2017

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Several map objects were created land added to folium maps:
  - **Circles** to mark launch sites in the map;
  - **Marks** with text or icons to attach information to launch sites, such name or mission outcomes;
  - and **Lines** to inform about launch sites and nearby objects.

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

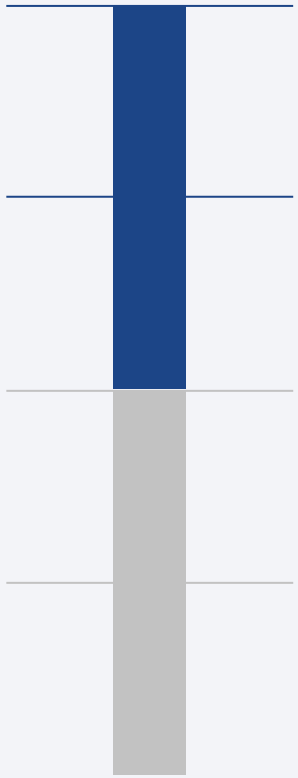
Build a Dashboard with Plotly Dash

The spacex_dash_app.py file includes the following plots and interactive elements in the dashboard:

- **Pie Chart of Launch Success Rate by Launch Site**: offers a quick, comparative view of each site's success rate, making it easier to identify sites with the best reliability and efficiency. Users can also get an insight about which of the launch sites contributes the most for success outcomes of the program.
- **Scatter Plot of Launch Outcomes by Payload Mass, Launch Site and Booster Version**: this plot is essential for analyzing how payload size affects the success rate across SpaceX's various launch locations. Providing filters by launch site allows users to focus on site-specific performance, which can highlight how conditions or resources at each location might impact launch success.
- **Dynamic Filtering Based on Launch Site and Payload Range**: This interaction supports custom queries, enhancing the dashboard’s versatility for users who want to drill down into specific conditions influencing launch success. This dual-filtering option helps discover patterns that might not be visible with a broader dataset.

The code is available at: https://github.com/kindalus/ibm-mod-6/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)



Data Preparation

used StandardScaler to normalize features, then splitted the data between training and testing sets

Model Selection and Tuning

four models were identified as possible candidates linear regression, support vector machine, decision trees and knn; then models were trained and their hyperparameters tuned using grid search

Model Evaluation

using several performance metrics such as precision, recall, f1 and jaccard score

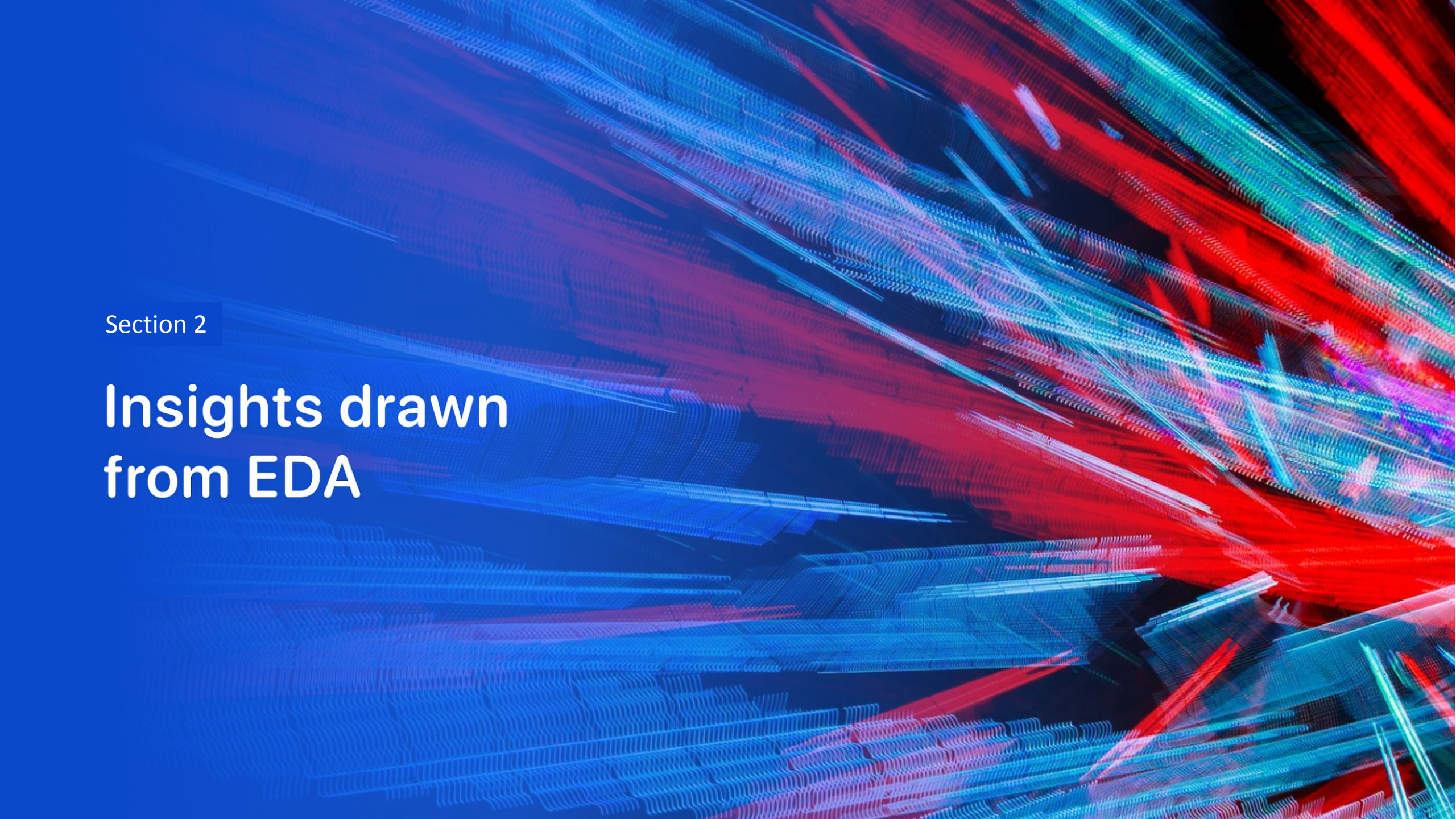
Model Selection

the selected model would be the one with the highest precision score since higher the precision the score the lesser the false positives that would increase the costs of the project

https://github.com/kindalus/ibm-mod-6/blob/master/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb

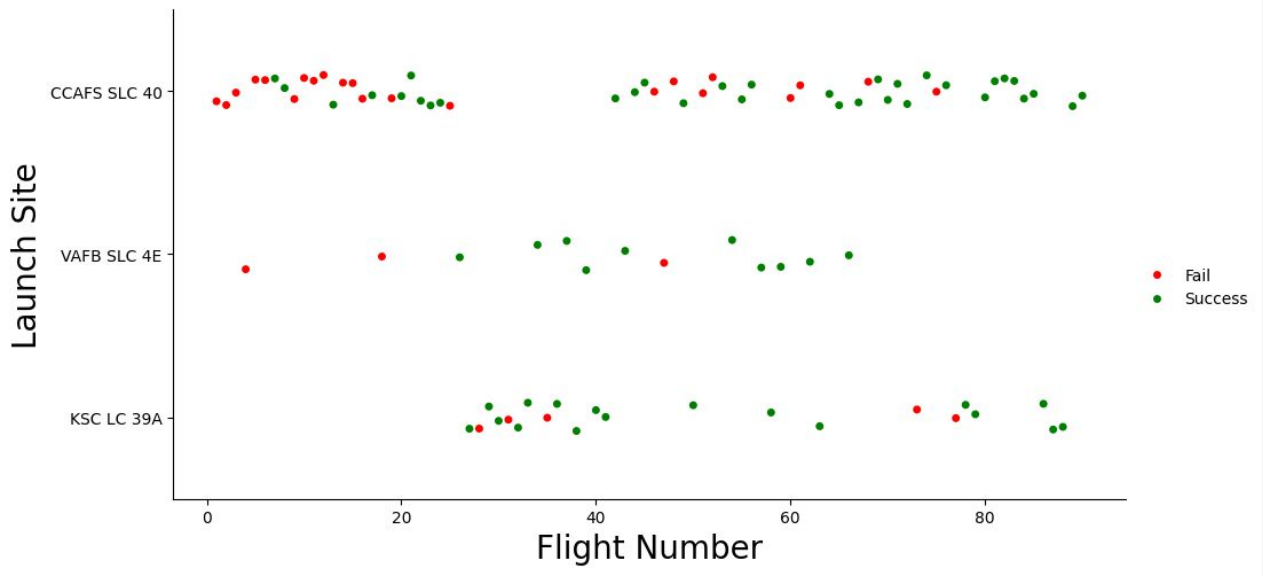
Section 2

Insights drawn from EDA



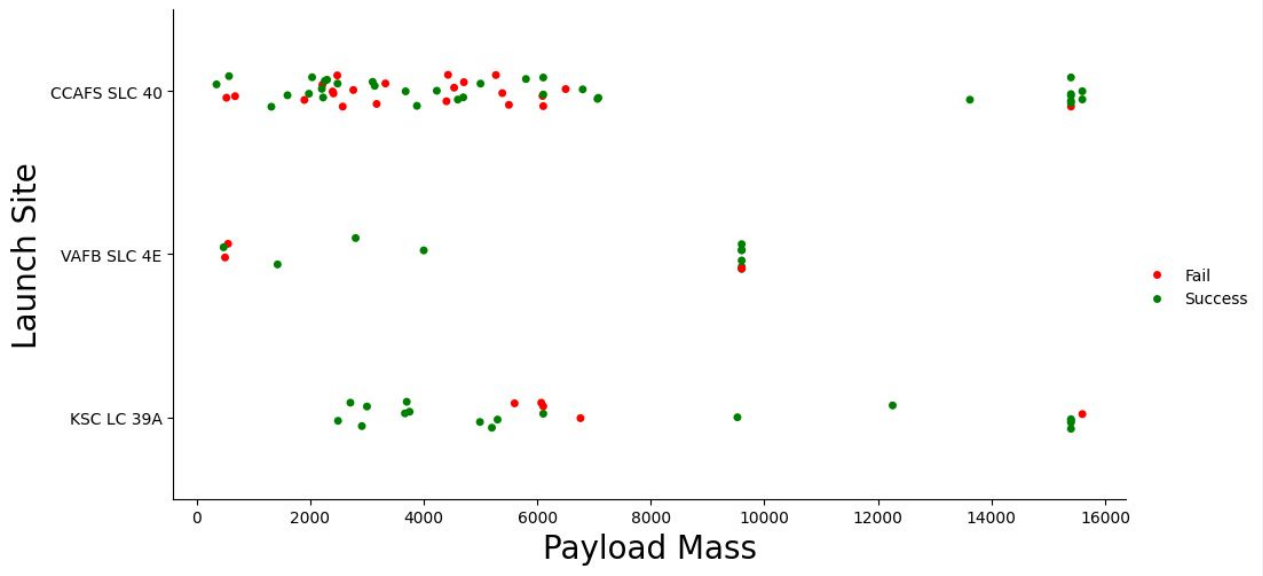
Flight Number vs. Launch Site

- There’s clearly a drop in launch failures with the increase of flight numbers;
- Although, like other launch sites CCAFS SLC 40 failures decrease with flight number, it is the site with the most failures.



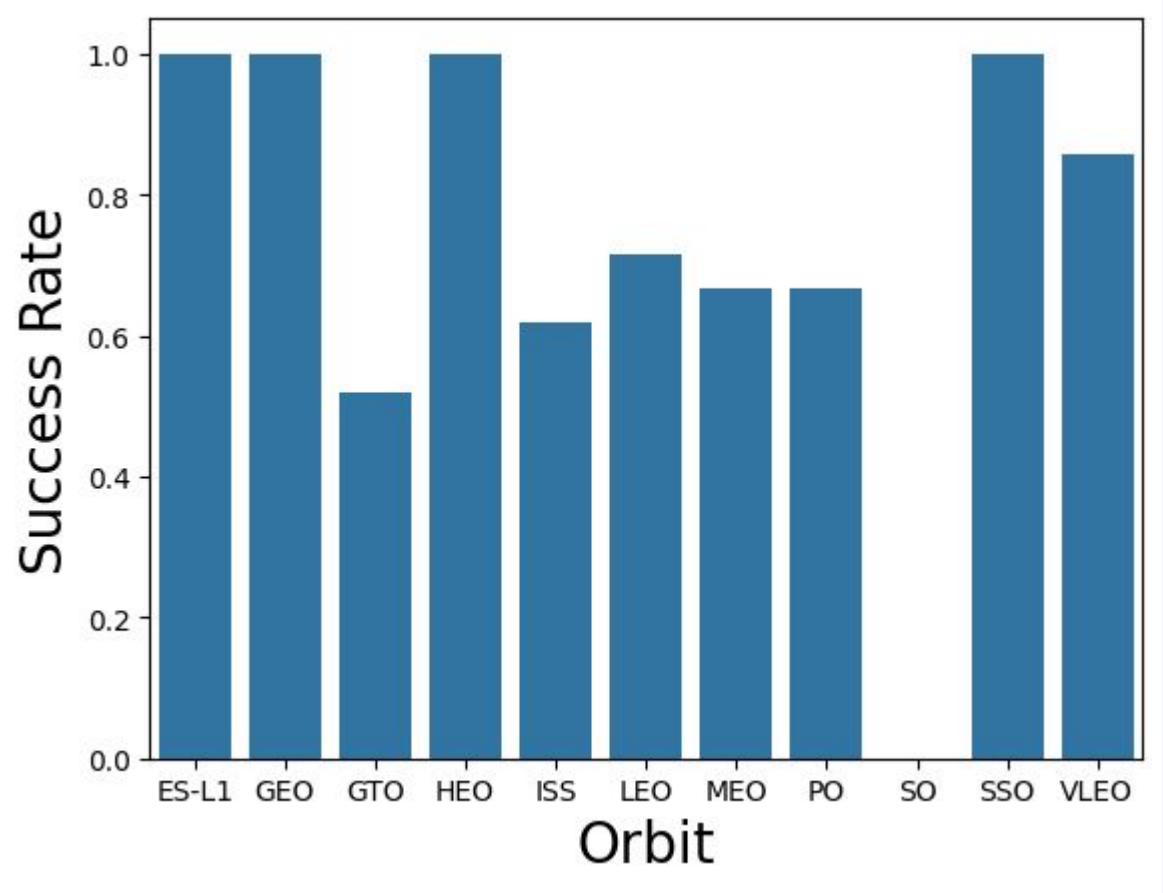
Payload vs. Launch Site

- The heavier the payload, the higher the success rate in launches;
- There is a 83.35% success rate when the payload is heavier than 8000KG



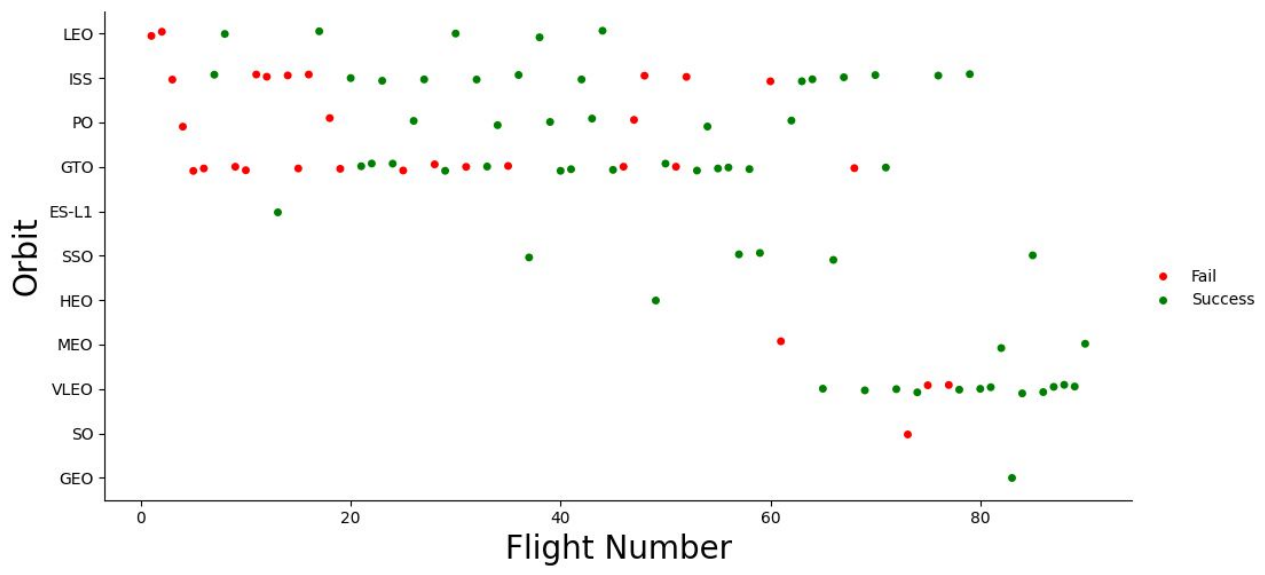
Success Rate vs. Orbit Type

- There are four orbits with only success launches;
- There’s no success lunch to the “SO” orbit



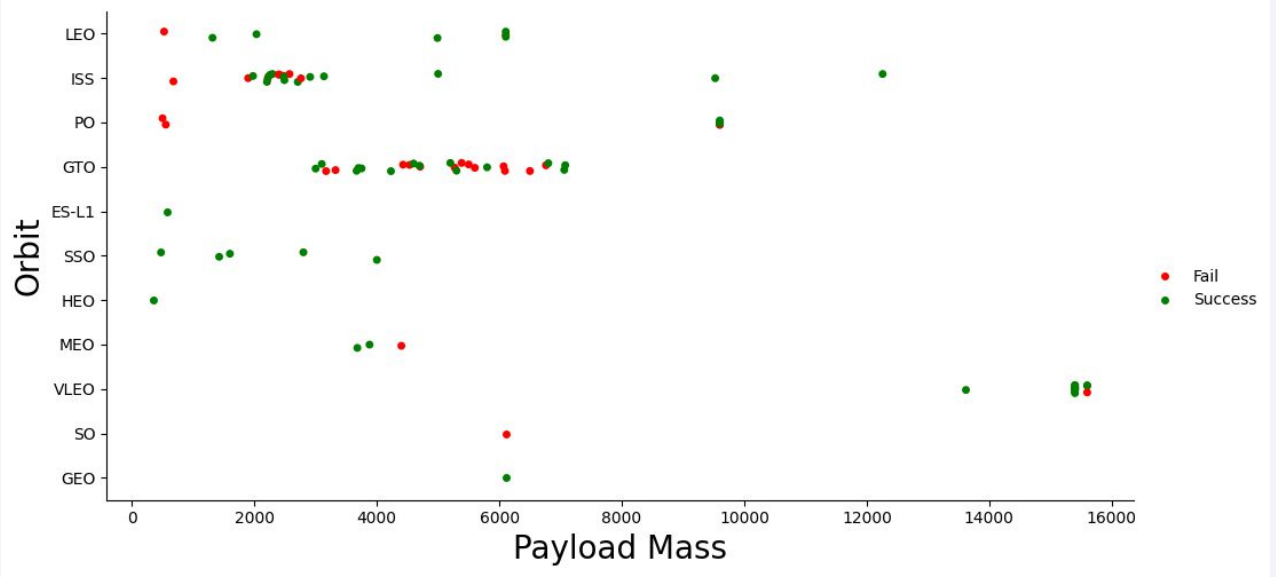
Flight Number vs. Orbit Type

- For GTO orbit it is not clear that success launches increase with more flight
- SSO orbit has a 100% success rate with 5 launches
- Other orbits with 100% success rates have too few launches to be statistical relevant



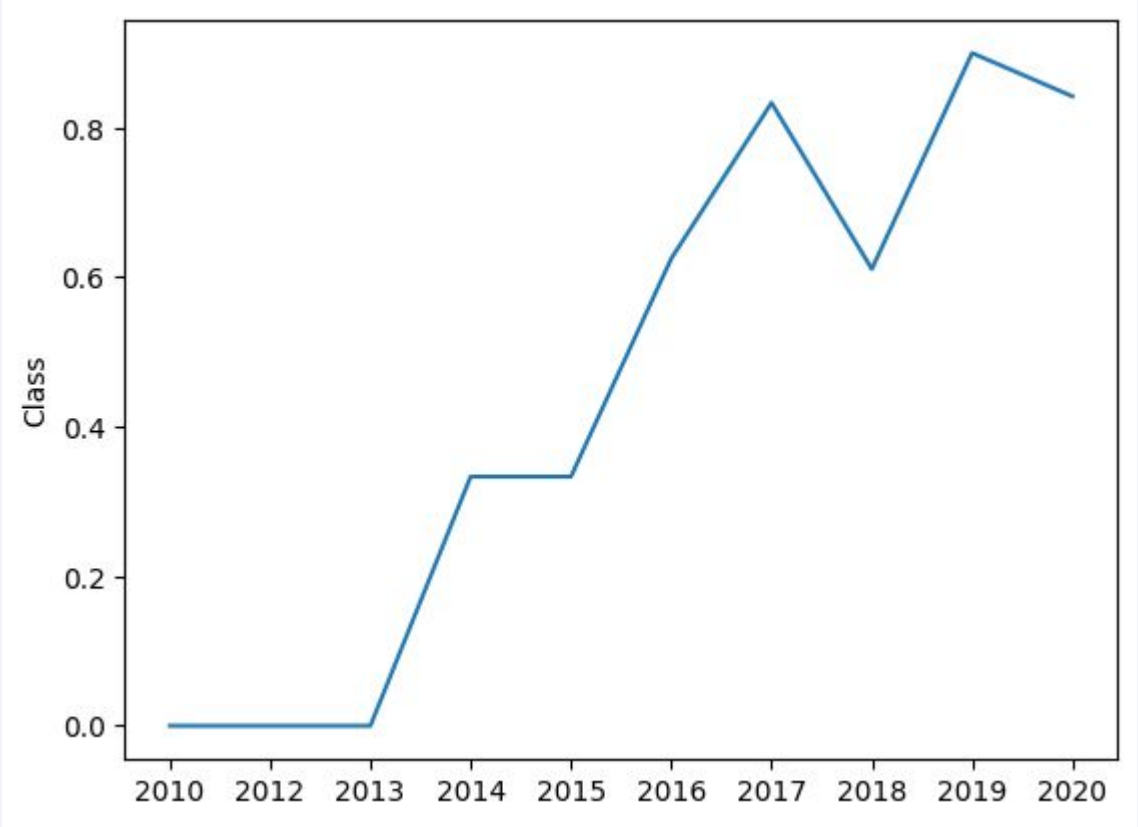
Payload vs. Orbit Type

- Again, the chances of success just rise when the payload in heavier than 8000KG
- For GTO orbit it seems not the matter the payload mass, looks like payload mass have no influence on the launch outcome



Launch Success Yearly Trend

- The program has seen progress its successful launch since 2014
- For some reason there was a momentaneous set back 2018 when the success suddenly dropped from around 80% to 60%, but in 2019 the program got back in the +80% success rate



All Launch Site Names

- The result was obtained using the DISTINCT keyword to the query
- Here is the query:
  - `SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;`

Launch Site Names Begin with 'CCA'

- In this query the wildcard ‘%’ tells the database that before CCA it can have any sequence of characters
- To retrieve only 5 records the ‘LIMIT’ keyword was used

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

- To calculate the total payload mass, it was necessary to use an aggregation function; in this case, the function SUM that adds all values for the same column across the table rows.

```
%%sql

SELECT SUM("PAYLOAD_MASS__KG_") AS "total_payload" FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

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

total_payload
45596

Average Payload Mass by F9 v1.1

- To calculate the total payload mass, it was necessary to use an aggregation function; in this case, the function AVG that adds all values for the same column across the table rows.

```
%%sql

SELECT AVG("PAYLOAD_MASS__KG_") AS "average_mass" FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

average_mass
2928.4

First Successful Ground Landing Date

- Using the min function was possible to select the first launch with Landing Outcome = ‘Success (ground pad)’

```
%%sql

SELECT MIN(DATE) FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- For this query two constraint had to be used, the first one to select only Landing outcomes equal to ‘Success (drone ship)’ the keywork BETWEEN to select row where the referenced column value is between two given values

```
%%sql

SELECT "Booster_Version", PAYLOAD_MASS__KG_ AS "Payload", "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing_Outcome" =
'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Booster_Version	Payload	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- For this query used two subqueries; one to select total number of successful outcomes; and other to select failed mission outcomes.

```
%%sql

SELECT (SELECT COUNT(*) FROM SPACEXTABLE WHERE "Mission_Outcome" = 'Failure (in flight)') AS
"Failure", (SELECT COUNT(*) FROM SPACEXTABLE WHERE "Mission_Outcome" <> 'Failure (in flight)')
AS "Success";
```

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

Failure	Success
1	100

Boosters Carried Maximum Payload

```
%%sql

SELECT DISTINCT "Booster_Version", PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_
= (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

✓ 0.0s Pytho

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- To extract the month from the string containing the date, the substr function from sqlite was used

```
%%sql

SELECT substr(Date, 6, 2) AS "month", "Booster_Version", "Launch_Site", "Landing_Outcome" FROM
SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)';
```

Python

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

month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

```
%%sql

SELECT "Landing_Outcome", COUNT(*) AS "Total" FROM SPACEXTABLE
    WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        AND "Landing_Outcome" IN ('Failure (drone ship)', 'Success (ground pad)')
    GROUP BY "Landing_Outcome"
    ORDER BY "Total" DESC;
```

✓ 0.0s Python

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

Landing_Outcome	Total
Failure (drone ship)	5
Success (ground pad)	3

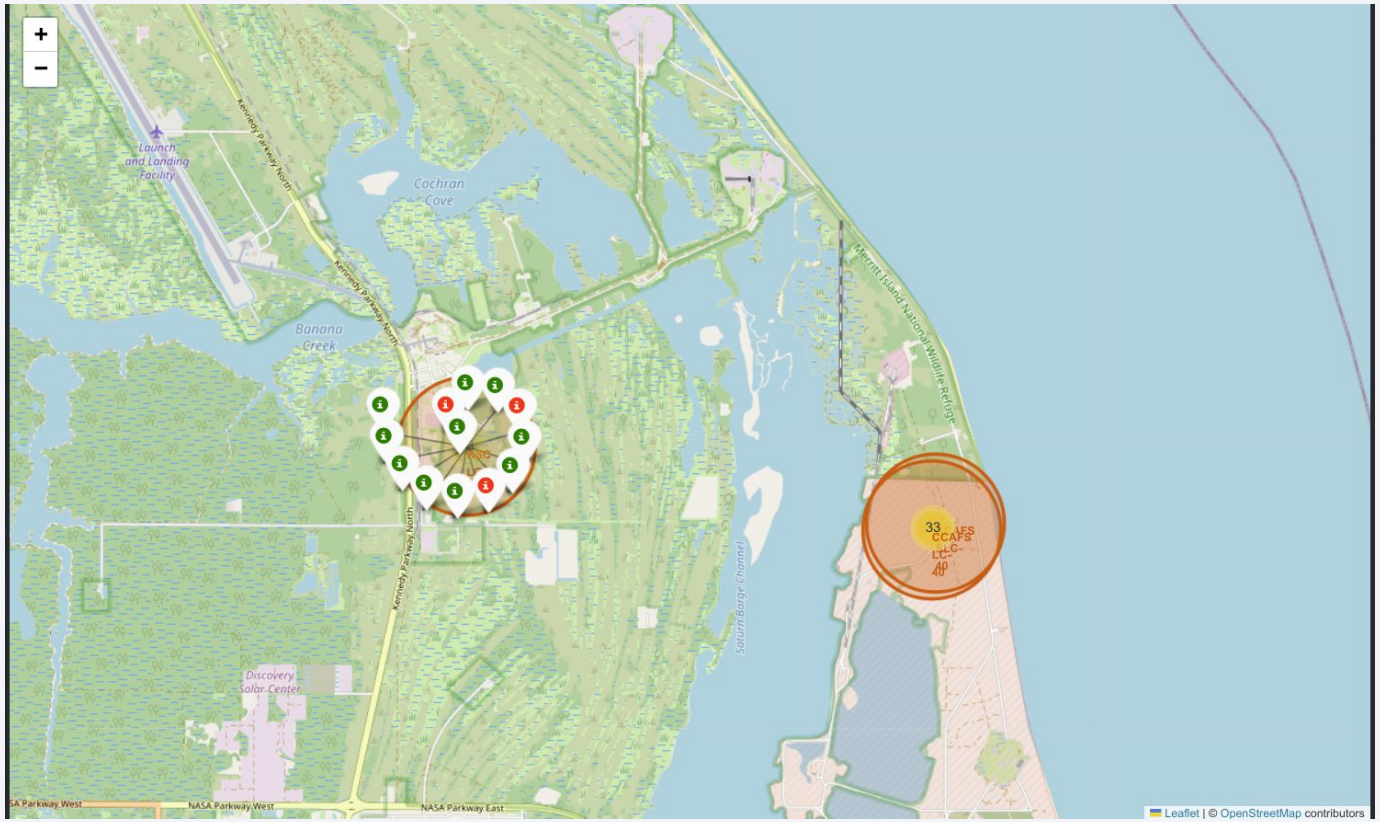

Section 3
Launch Sites
Proximities Analysis

Launch Sites Locations

- All launch sites are near the cost line.
- There’s only one site in the west coast

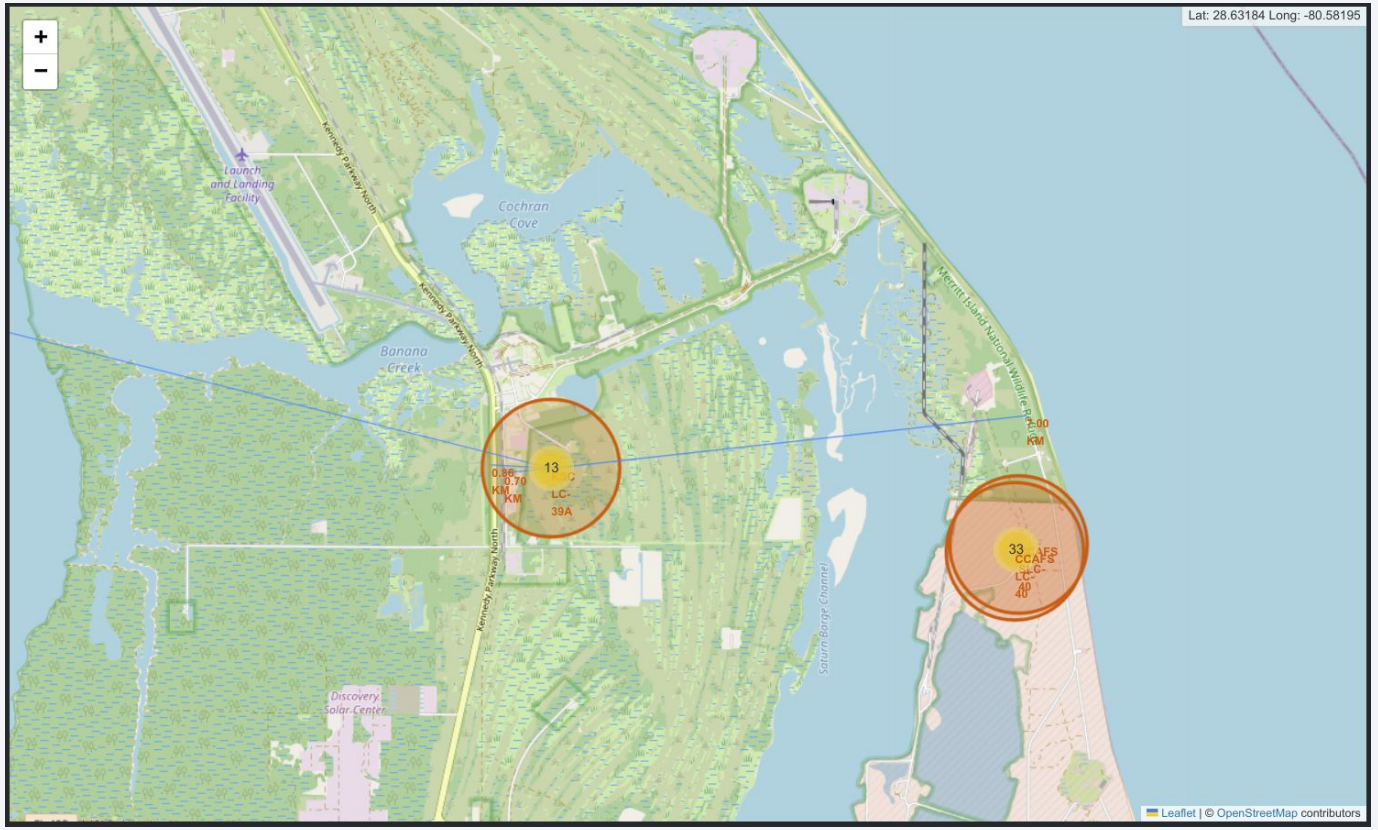
Launch Outcome Map

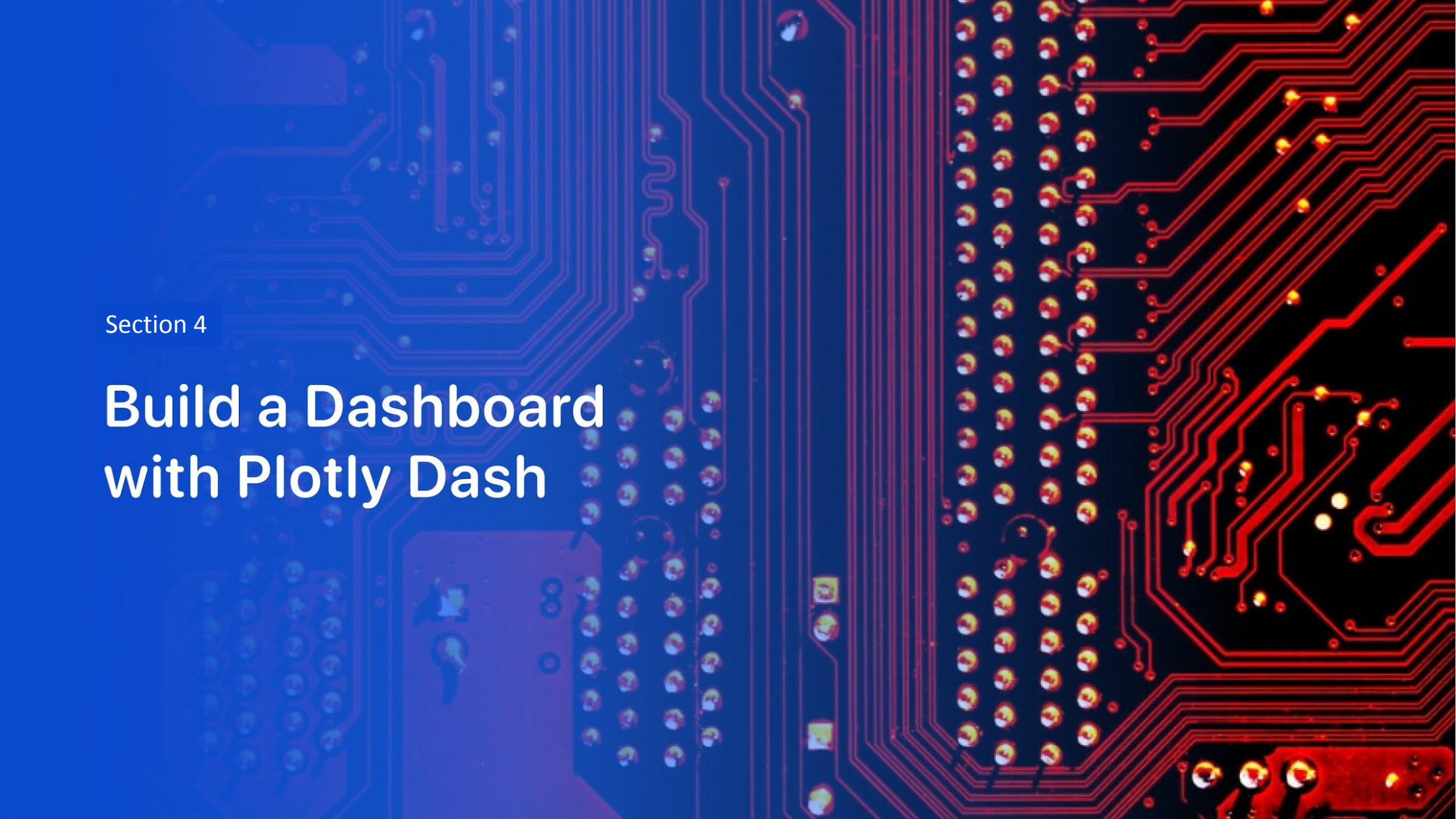
- With this map, one can clearly see the outcomes by location in a more impactful way than a table with numbers



Launch Site Proximities

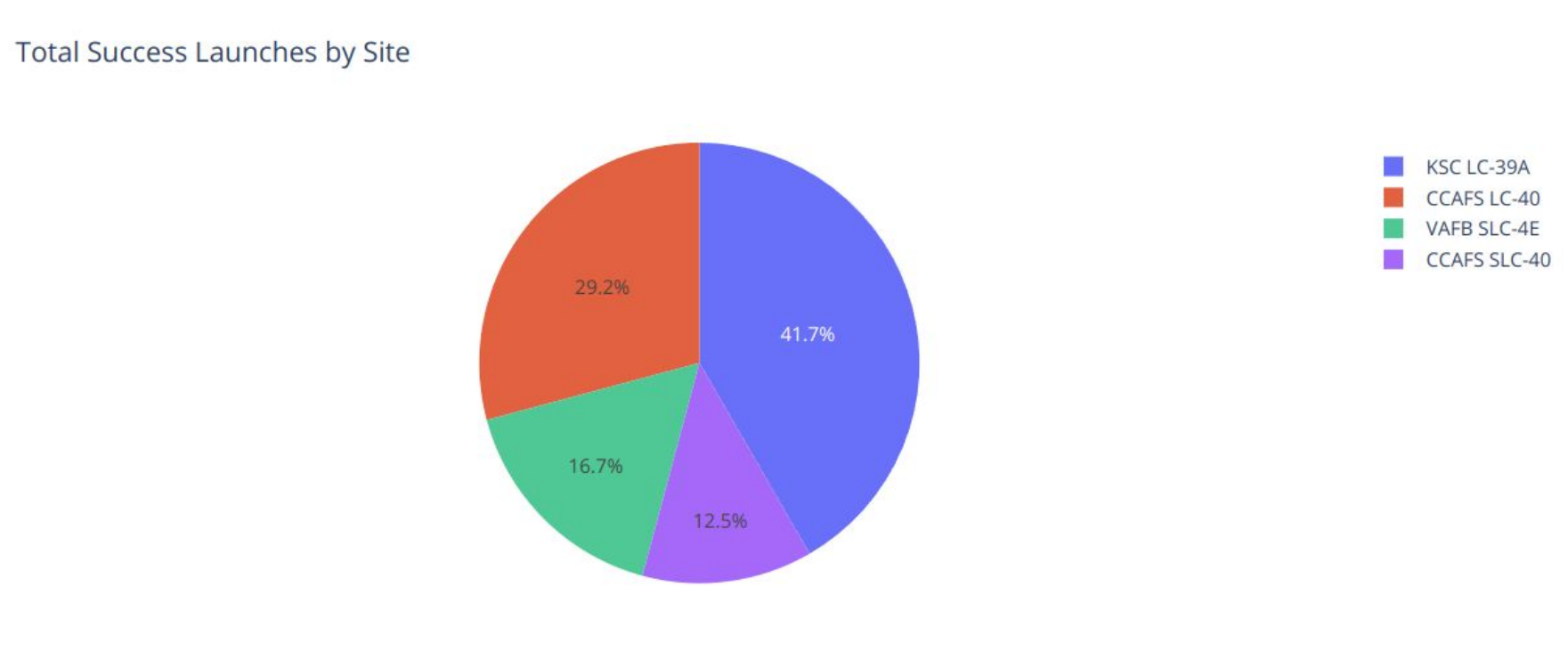
- Launch site are near railroads and sea cost, but far from major cities




Section 4
Build a Dashboard with Plotly Dash

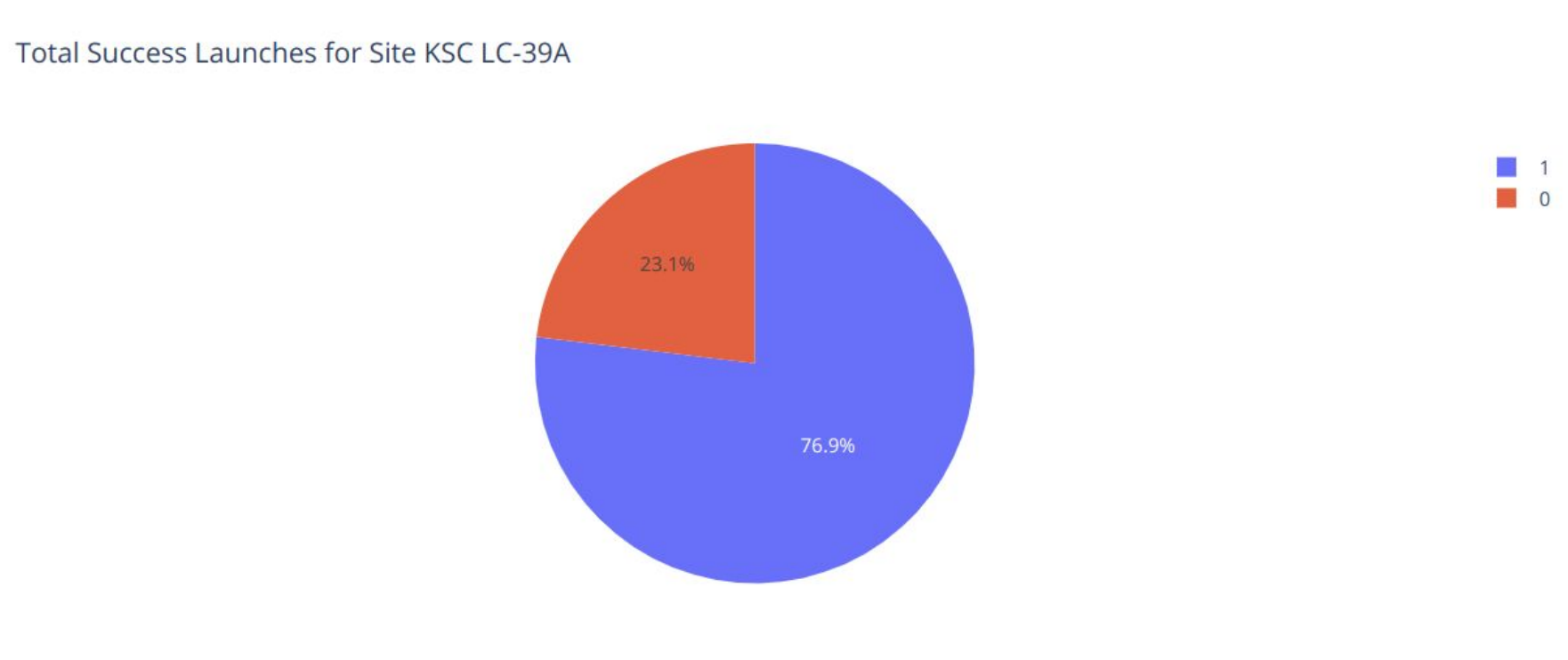
Total Success Launches by Site

- The KSC LC-39A is by far the site with the higher launch success rate



Success Rate for Site KSC LC-39A

- Site KSC LC-39A has the highest launch success rates with 76.9%



Correlation between Payload and Success for all Sites

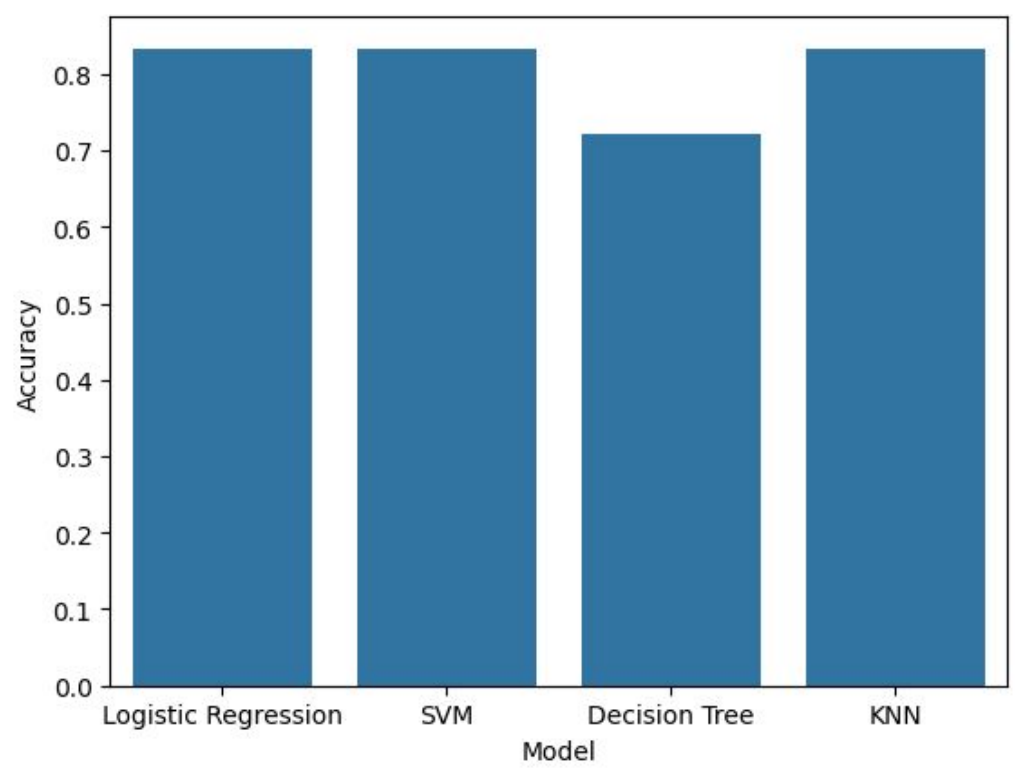
- No sure what this dashboard show us. In other findings the success increases with the increase of payload. Further analysis is required.

Section 5

Predictive Analysis (Classification)

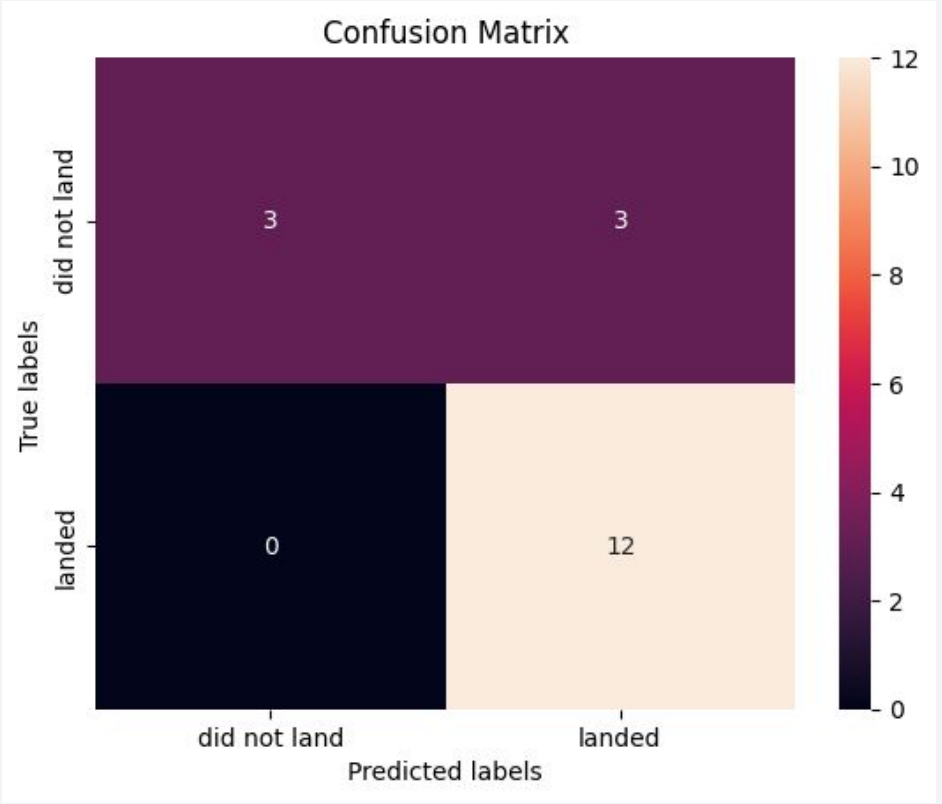
Classification Accuracy

- Almost all models have the same accuracy;
- Other metrics were used and the models still have the same performance, so any one of Logistic Regression, SVM or KNN can be picked as the best predictor.



Confusion Matrix

- The model is really good identifying mission that landed successfully
- However, when it comes to identifying failed launches, the model can only identify 50% of them



Conclusions

- From EDA it is clear to see that Falcon 9 missions became more successful over the years.
- Besides the number of launches, payload mass seem to have a great impact in mission outcome.
- During predictive analysis four models showed promising results with accuracy over 83%, however further improvement on those models is required or, perhaps the training of another one that helps reduce the number of False Positives.

Thank you!