IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Dafina D.>
<May 2024>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using web scrapping and SpaceX API
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with SQL
  - Exploratory Data Analysis (EDA) with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - EDA facilitated the identification of the most predictive features for the success of launchings.
  - Machine Learning Prediction revealed the optimal model for predicting the key characteristics that drive launch success, leveraging all available data.

Introduction

SpaceX is a revolutionary company that has disrupted the space industry by offering rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollars each. Most of these savings are thanks to SpaceX's astounding idea to reuse the first stage of the launch by re-landing the rocket to be used on the next mission. Repeating this process will bring the price down even further. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future.

This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variable and how it is affecting the outcome.
- The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - SpaceX API - https://api.spacexdata.com/v4/rockets/
  - WebScrapping - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- **Perform data wrangling**
  - Classifying landings as successful or unsuccessful.
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**

Methodology

Executive Summary

- **Perform predictive analysis using classification models**
    - The collected data has been normalized, split into training and test sets, and assessed using four distinct machine-learning models. The accuracy of each model was then evaluated.

Data Collection

- Data sets were gathered from the SpaceX API (https://api.spacexdata.com/v4/rockets/) and from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches), using web scraping techniques.

- SpaceX API columns
  - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

- Wikipedia Scraping columns
  - Flight No, Launch site, Payload, Payload mass, Orbit, Customer, Launch outcome, Version Booster, , Booster Landing, Date, Time

Data Collection – SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

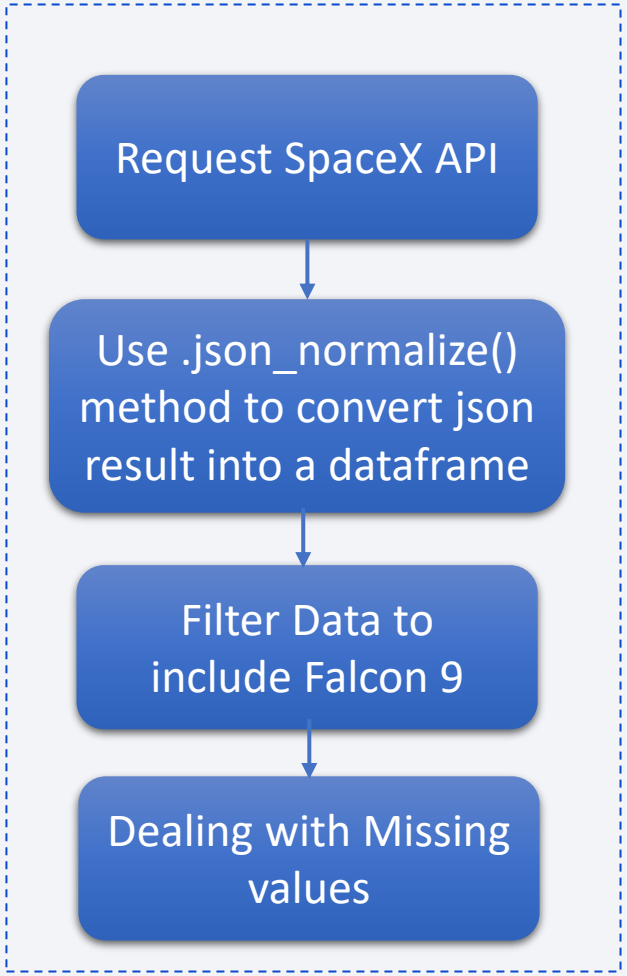
```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```



- Github URL:

https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/Lab1-DataCollectionAPI.ipynb

Data Collection - Scraping

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
    html_data = requests.get(static_url)
    html_data.status_code
```

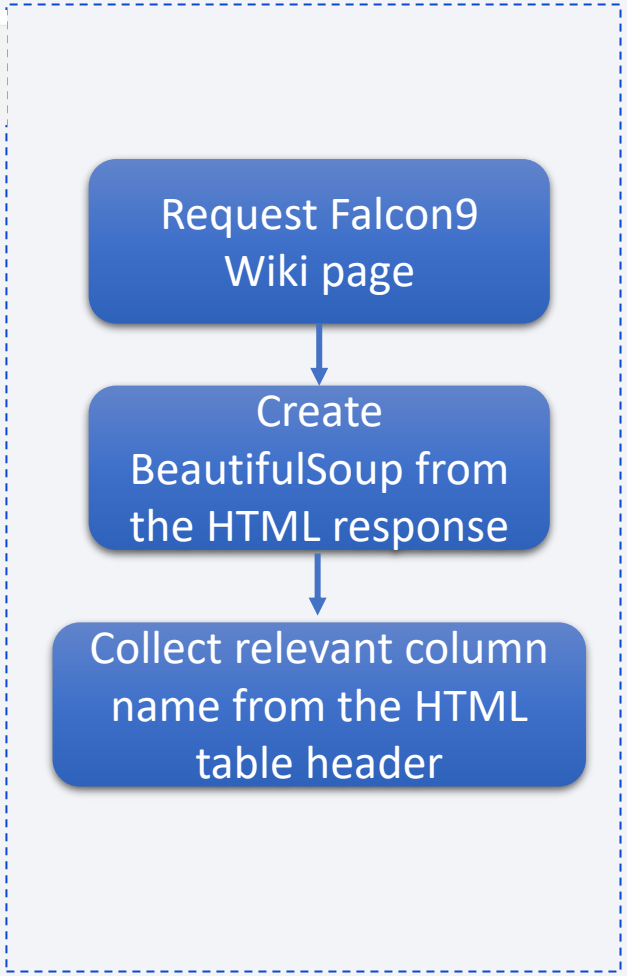
```
200
```

Create a `BeautifulSoup` object from the HTML `response`

```
    # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
    soup = BeautifulSoup(html_data.text,'html5lib')
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for element in first_launch_table.find_all('th'):
    name = extract_column_from_header(element)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

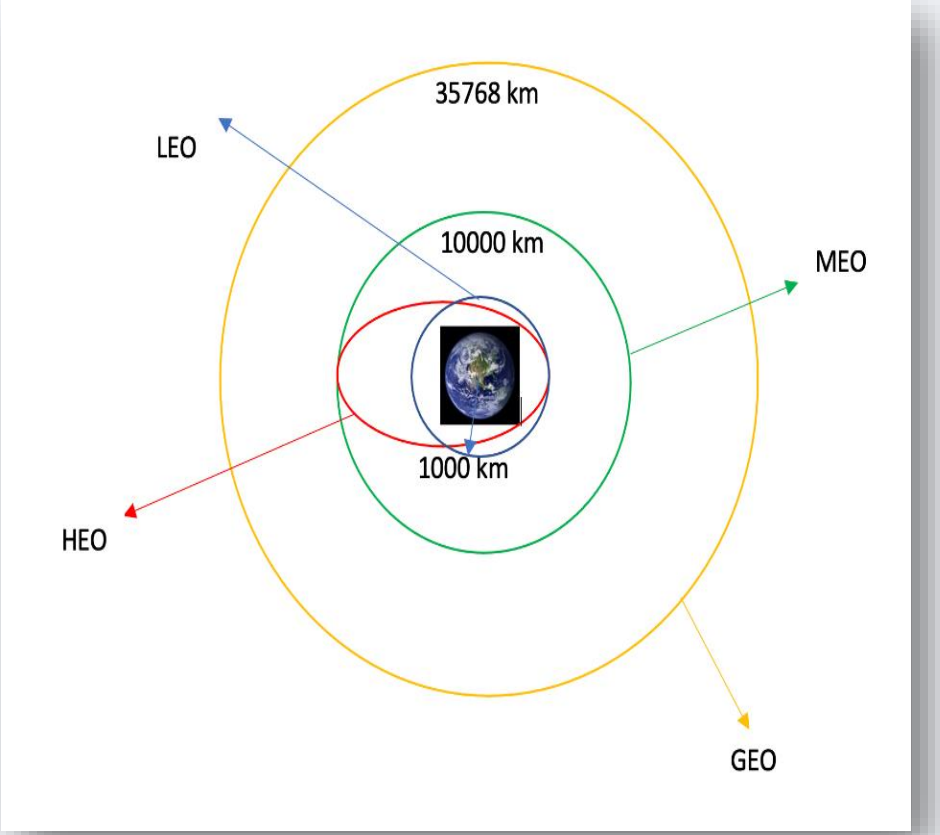


- Github URL: https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/Lab1-DataCollectionWithWebScraping.ipynb

Data Wrangling

- We calculated the number of launches at each site, as well as the number and occurrence of each orbit.
- Next, create a landing outcome label where ‘0’ signifies a bad outcome, and ‘1’ denotes a successful one. Then, export this data to a CSV format.
- Github URL:

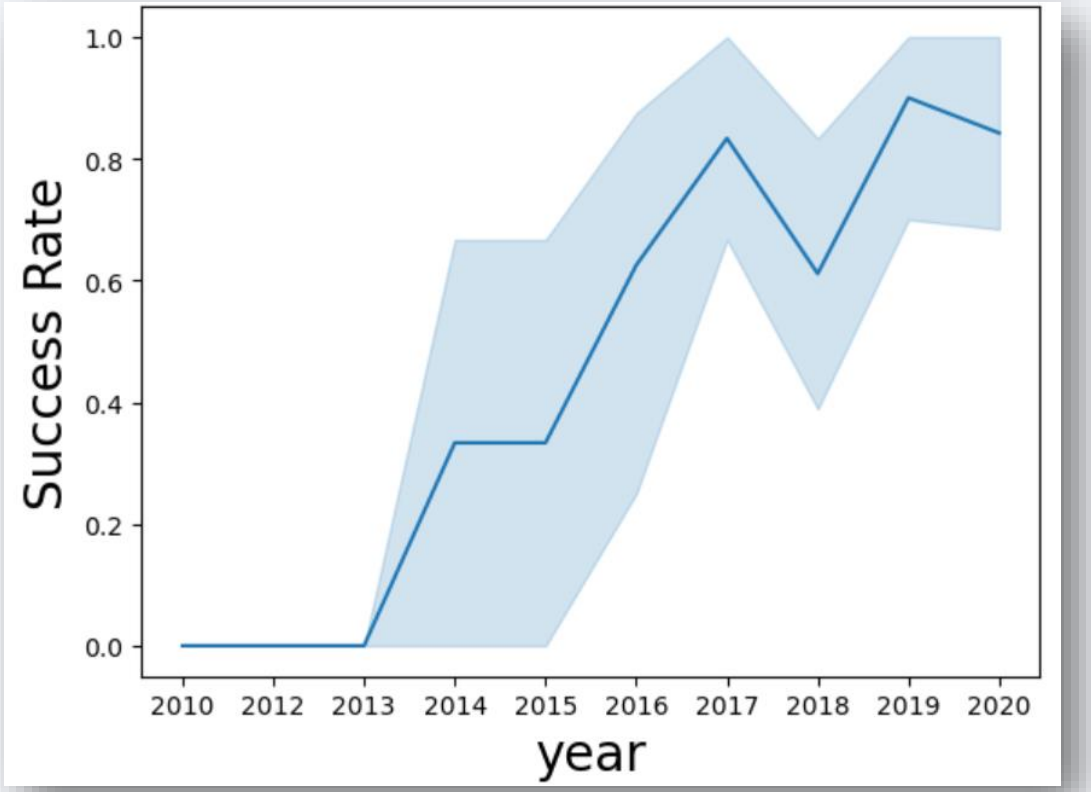
https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/Lab2-DataWrangling.ipynb



EDA with Data Visualization

- Scatter plots, line charts, and bar plots were utilized to evaluate the relationships between variables, with the aim of discerning their potential utility in training the machine learning model.
- The data exploration process involved visualizing the relationship between flight number and launch site, payload and launch site, success rates of each orbit type, flight number and orbit type, as well as the yearly trend in launch success.
- Github URL:

https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/EDA%20with%20Visualization.ipynb



EDA with SQL

- Loaded the spaceX dataset into a PostgreSQL.
- The SQL is used to answer the several questions about the data such as:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- GitHub URL:

https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Latitude and longitude coordinates were utilized to display launch data on an interactive map, with circle markers representing each launch site labeled accordingly.
- Launch outcomes, categorized into failure and success, were depicted using red and green markers, assigned to classes 0 and 1 respectively, via MarkerCluster() on the map.
- Then, we calculated the distances between a launch site to its proximities.
- Github URL:

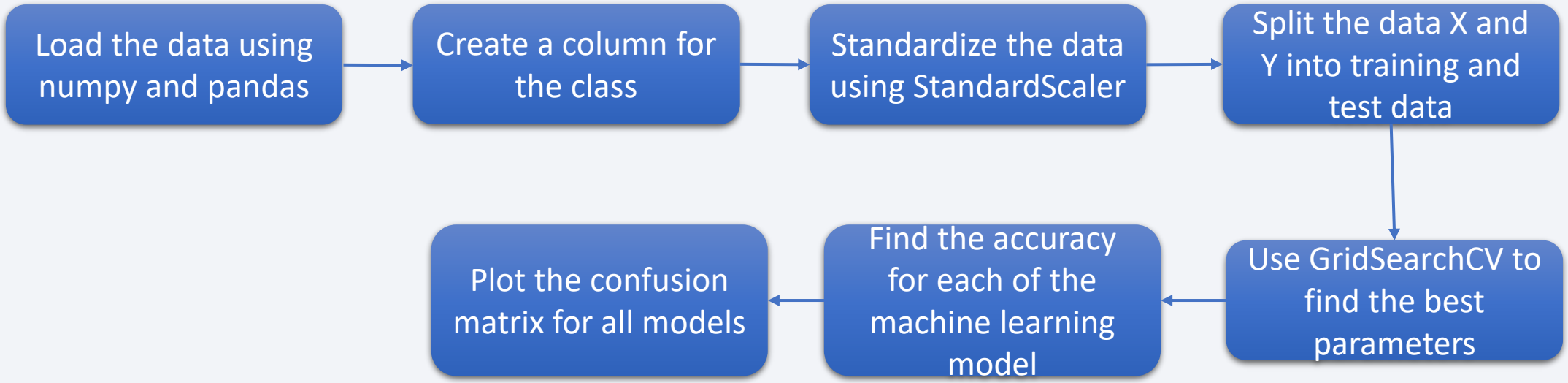
https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/Site%20Location%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Build interactive dashboards including pie charts and scattered plots.
- A pie chart can be selected to show the total success of launches by certain sites.
- A scatter plot illustrating the relationship between Outcome and Payload Mass (kg) across various booster versions.
- GitHub URL:

https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/spacex_dash_app.py

Predictive Analysis (Classification)

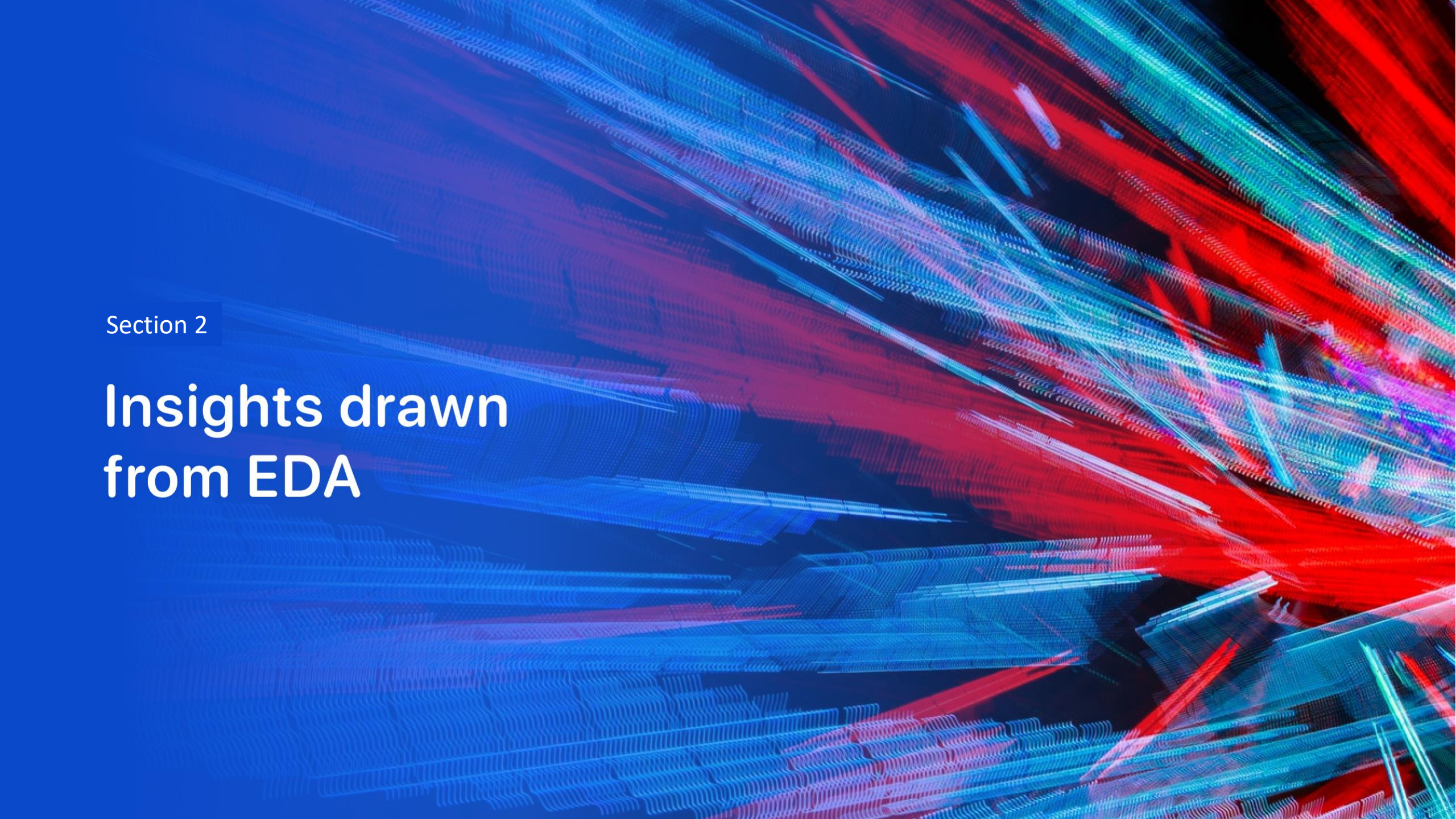


GitHub URL:

https://github.com/dafinad/Coursera/blob/main/IBM_AppliedDataScienceCapstone/SpaceX%20Machine%20Learning%20Prediction.ipynb

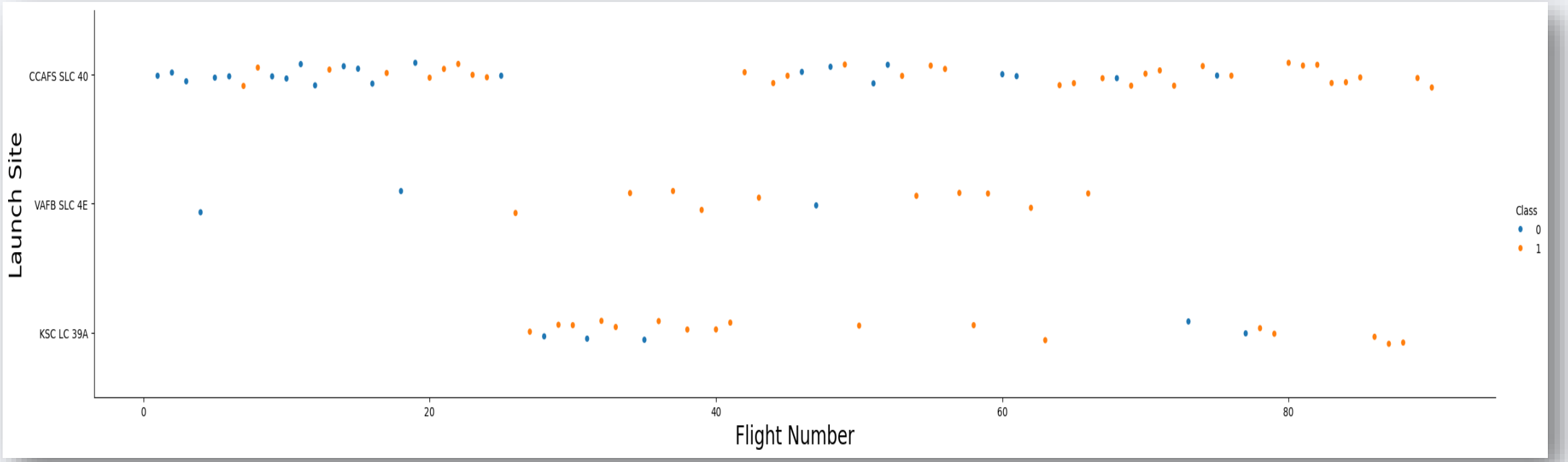
Results

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


Section 2
Insights drawn from EDA

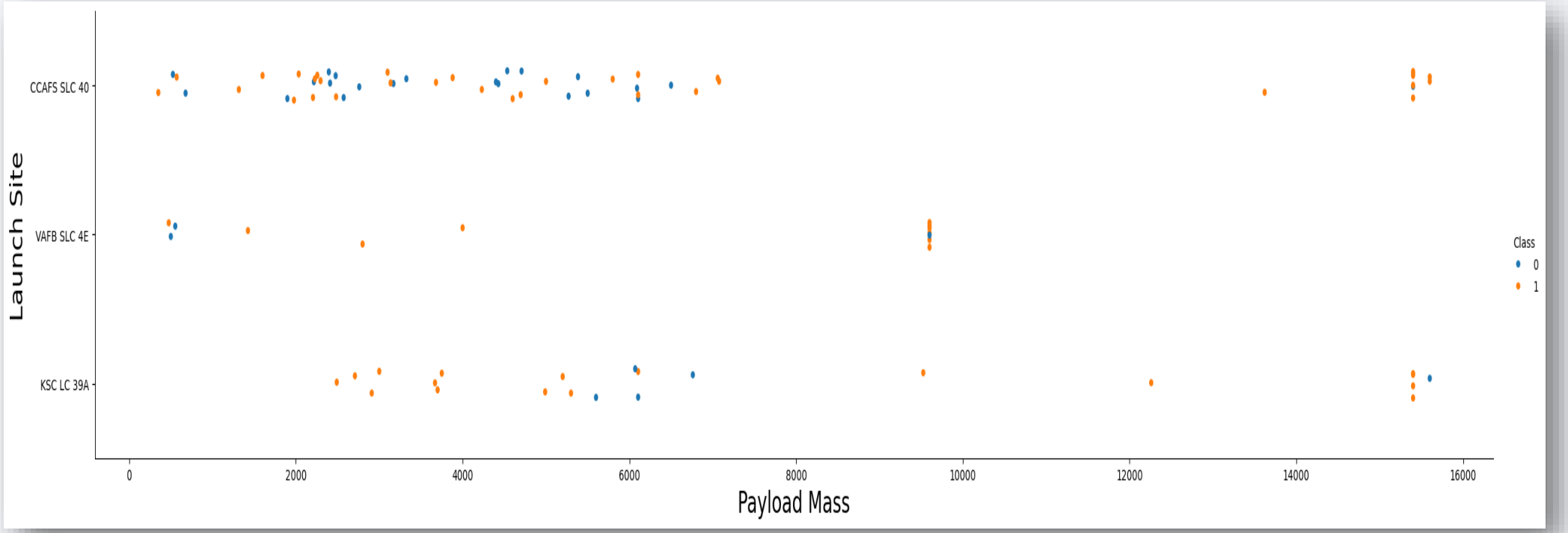
Flight Number vs. Launch Site

- From the plot, it's apparent that the success rate at a launch site tends to increase with a higher volume of flights.



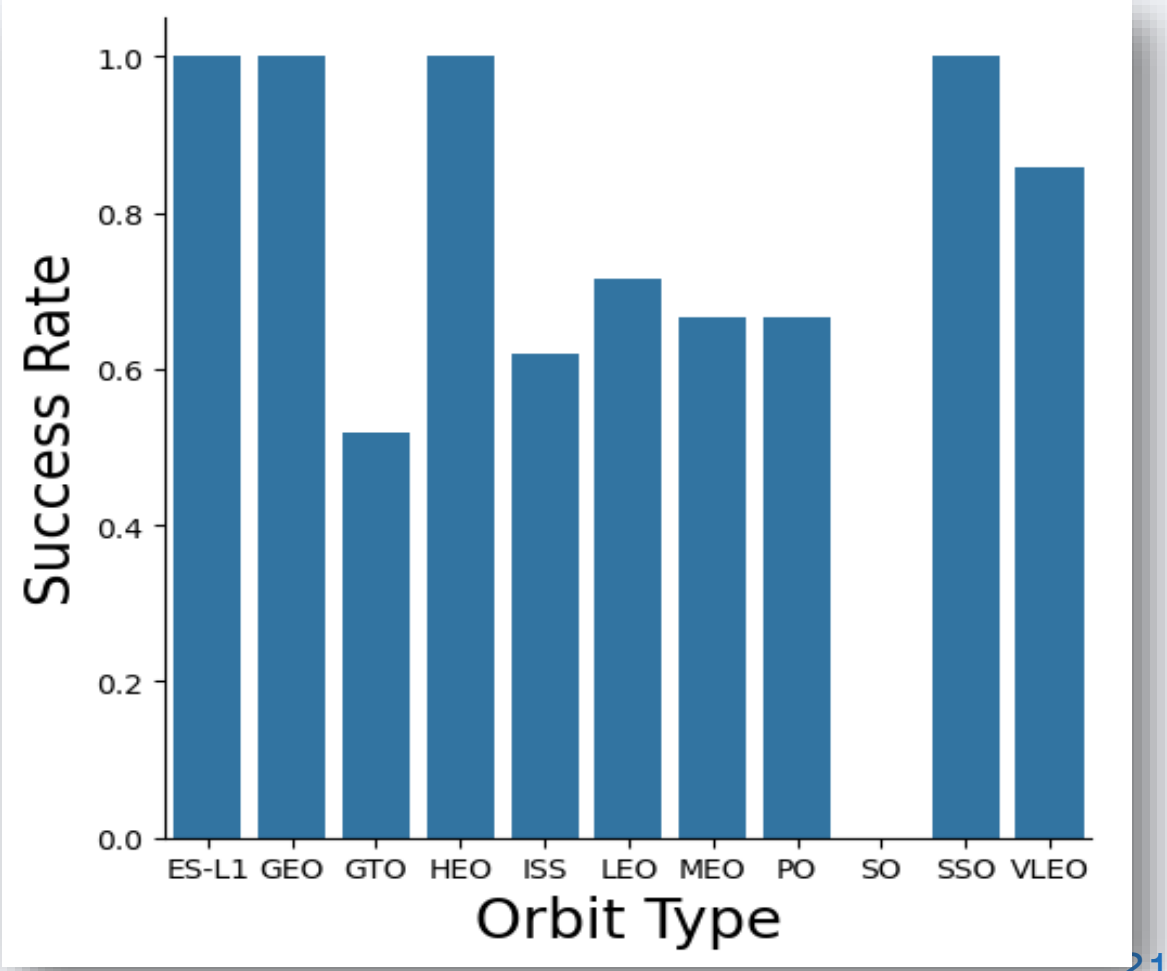
Payload vs. Launch Site

- This scatter plot indicates that once the payload mass exceeds 7000 kg, the probability of success significantly increases.



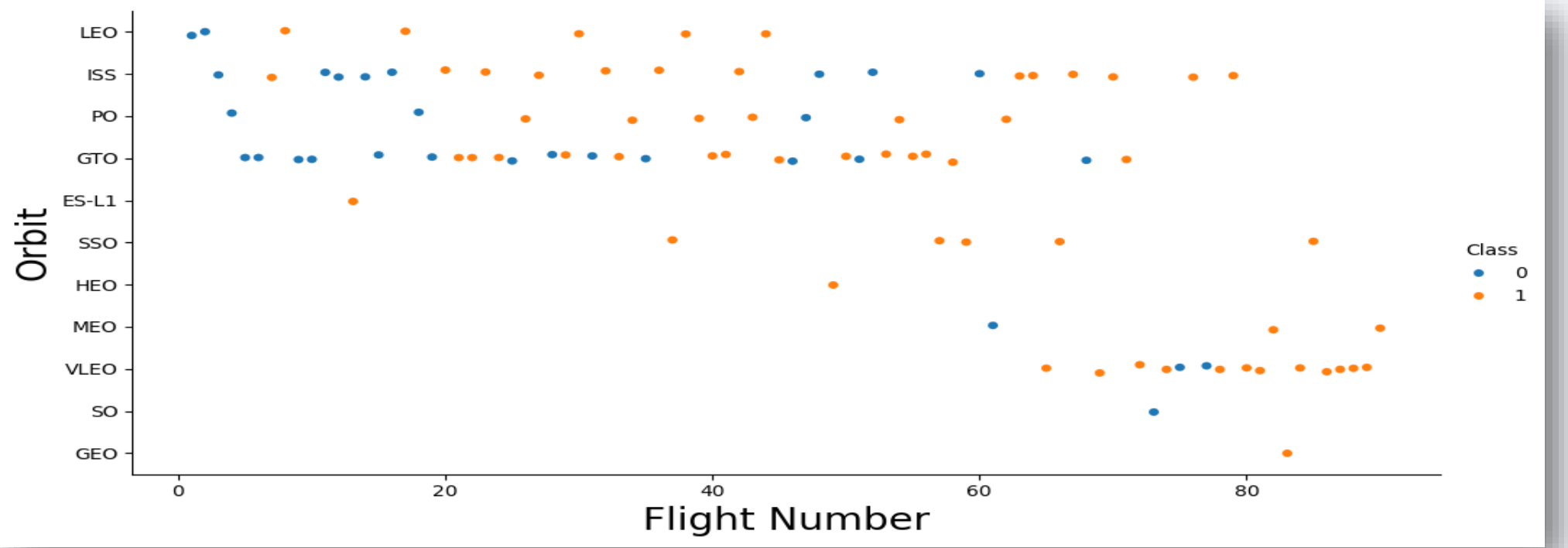
Success Rate vs. Orbit Type

- This figure illustrates how different orbits influence landing outcomes, with some orbits boasting a 100% success rate, such as SSO, HEO, GEO, and ES-L1, while SO orbit, shows a 0% success rate.



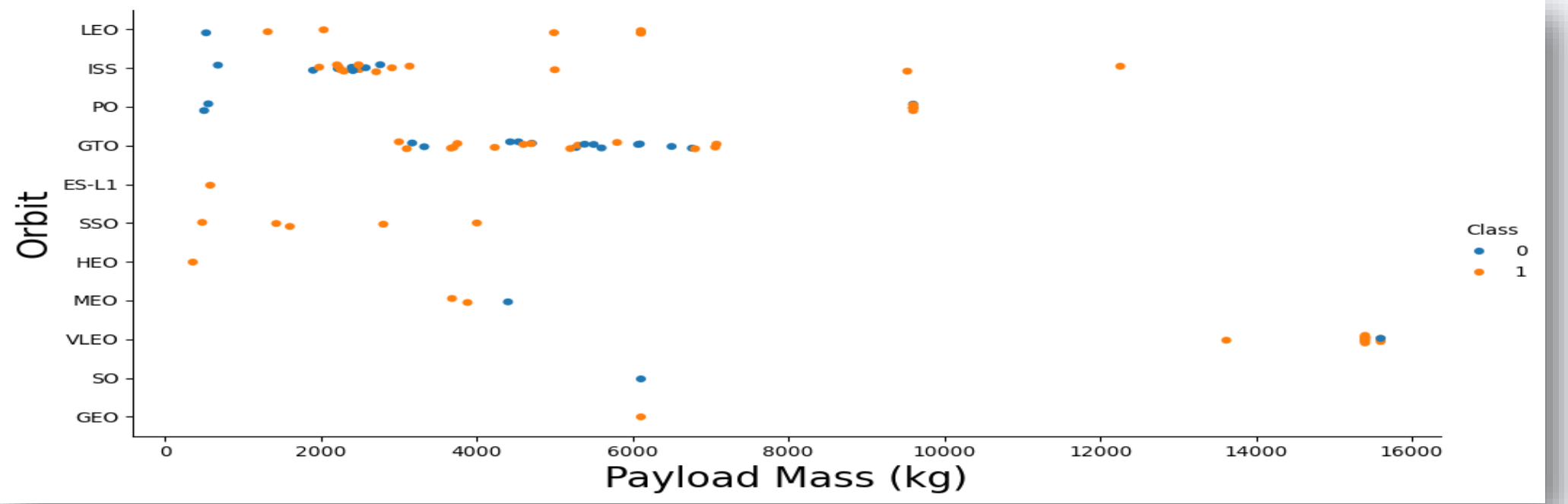
Flight Number vs. Orbit Type

- This scatter plot indicates that, generally, the success rate increases with a higher flight number on each orbit, particularly in the LEO orbit. However, for the GTO orbit, there appears to be no clear relationship between the two attributes.



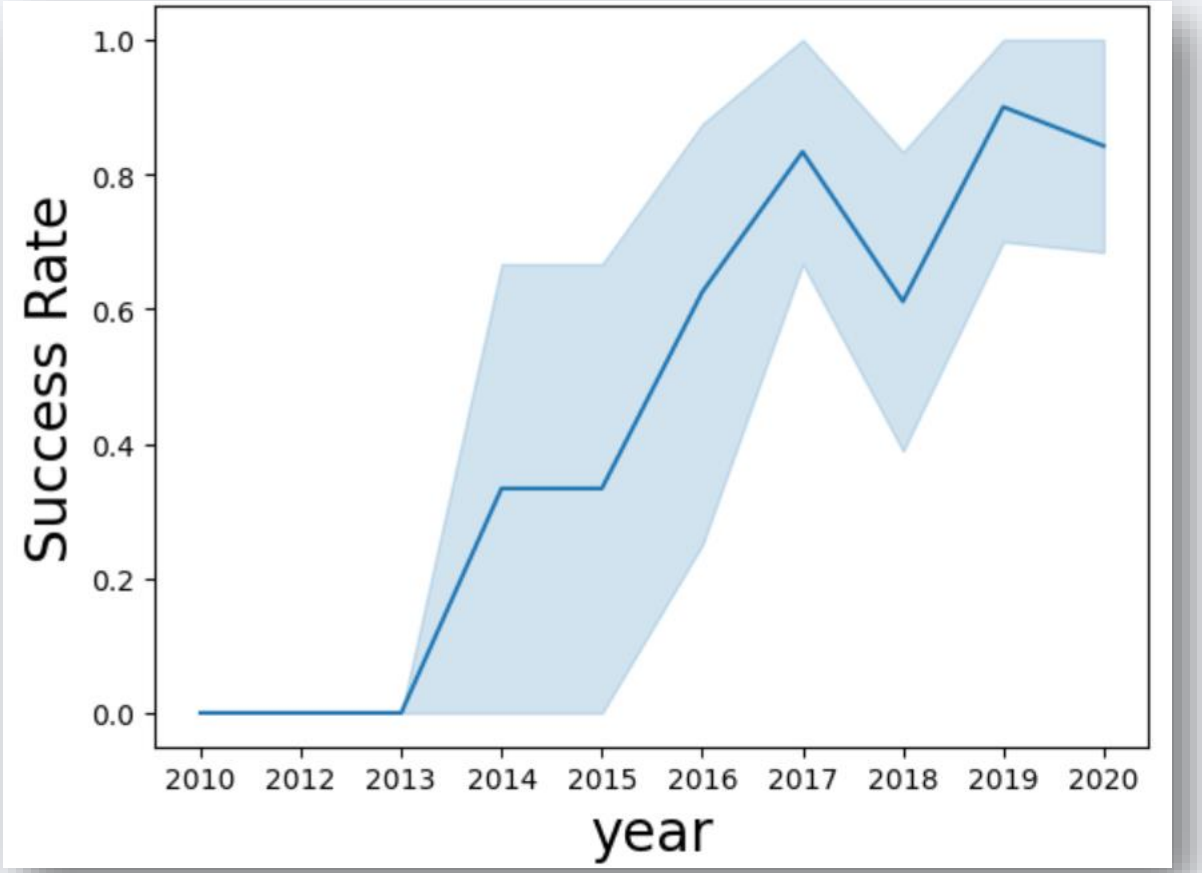
Payload vs. Orbit Type

- It's evident that with heavy payloads, successful landings are more frequent for PO, LEO, and ISS orbits.



Launch Success Yearly Trend

- The figure clearly depict an increasing trend from the year 2013 until 2020, with a slight dip in 2018.



All Launch Site Names

- The DISTINCT keyword was utilized to display only unique launch sites from the SpaceX data.

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1
```

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

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

Launch Site Names Begin with 'CCA'

- The query present 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string '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	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

- The query below was used to calculate the total payload mass where NASA was the customer.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster vrsion F9 v1.1.

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

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

AVG (PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- The date of the first successful landing outcome.

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Mission_outcome LIKE 'Success%'
```

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

MIN(Date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- This query returned 4 booster versions that had successful landing outcomes and payload mass between 4000 and 6000.

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND "Landing_Outcome" = 'Success (
```

```
 * 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

- We utilized the wildcard '%' to filter for WHERE MissionOutcome, whether it was a success or a failure.

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME like 'Success%'
```

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

Mission_Outcome	COUNT(MISSION_OUTCOME)
Success	100

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME = 'Failure (in flight)
```

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

Mission_Outcome	COUNT(MISSION_OUTCOME)
Failure (in flight)	1

Boosters Carried Maximum Payload

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL
```

```
 * 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

- We identified the booster that carried the maximum payload by utilizing a subquery within the WHERE clause along with the MAX() function.

2015 Launch Records

- We employed a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes on drone ships, including their booster versions and launch site names for the year 2015.

```
%%sql SELECT "Booster_Version", "Launch_Site" FROM SPACEXTABLE

WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr(Date,1,4) = '2015';
```

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

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql SELECT "LANDING_OUTCOME", COUNT(*) as 'COUNT' FROM SPACEXTBL
WHERE substr(Date,1,4) || substr(Date,6,2) || substr(Date,9,2)
between '20100604' and '20170320' GROUP BY "Landing_Outcome" ORDER BY "COUNT" DESC;
```

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

Landing_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

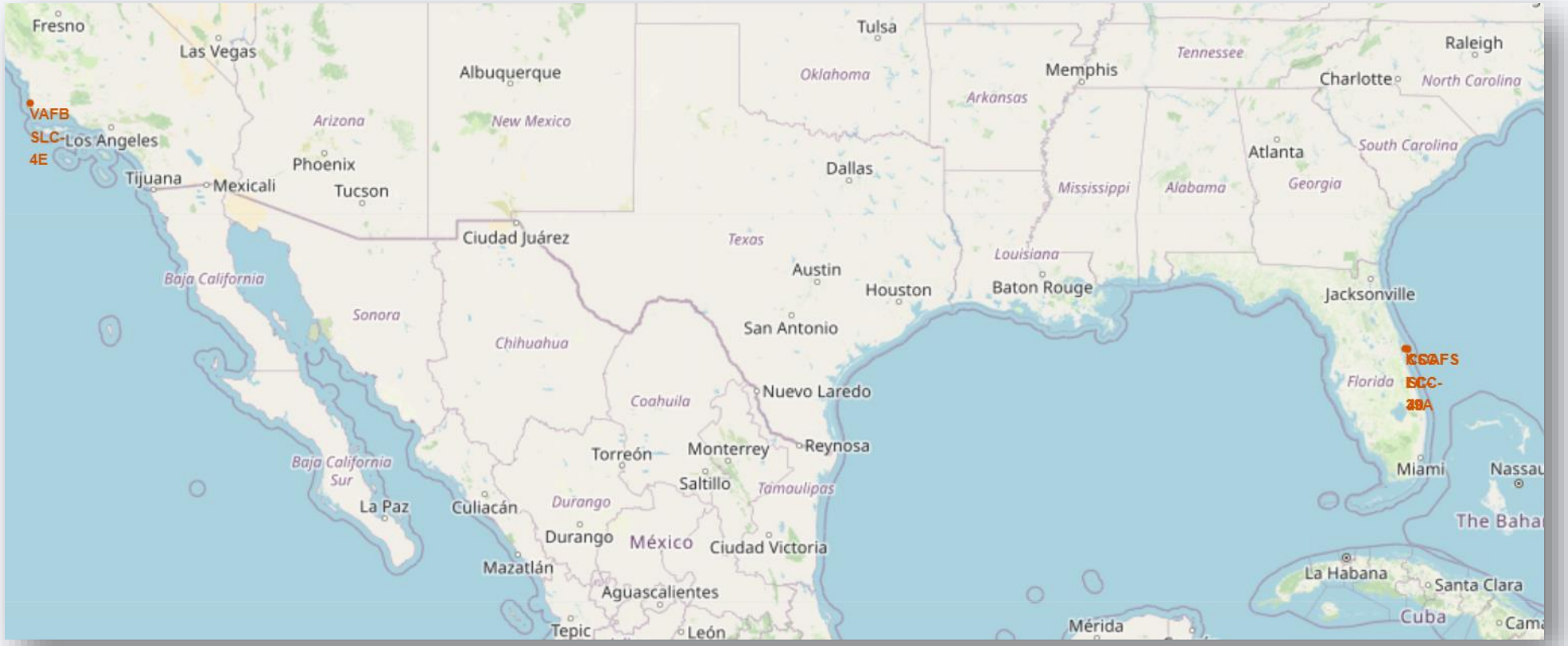
- **SELECT** the Landing Outcome and **Count** the landing outcome and use **WHERE** to filter the landing outcome between **2010-06-04 to 2017-03-20**.
- Next, apply the **GROUP BY** to group the landing outcomes and **ORDER BY** to arrange the landing outcomes in **descending order**.

Section 3

Launch Sites Proximities Analysis

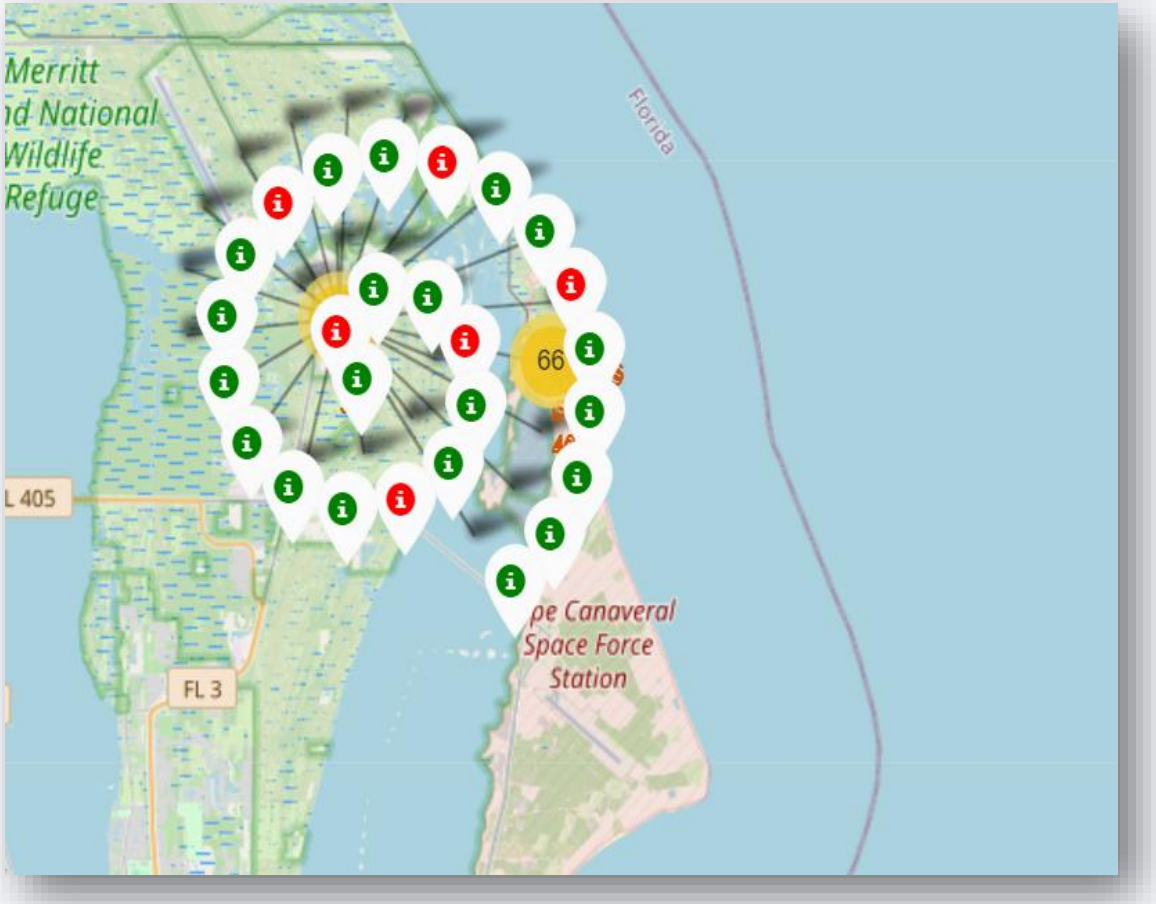
Location of all SpaceX launch sites

- It is clear that all SpaceX launch sites are situated within the United States.

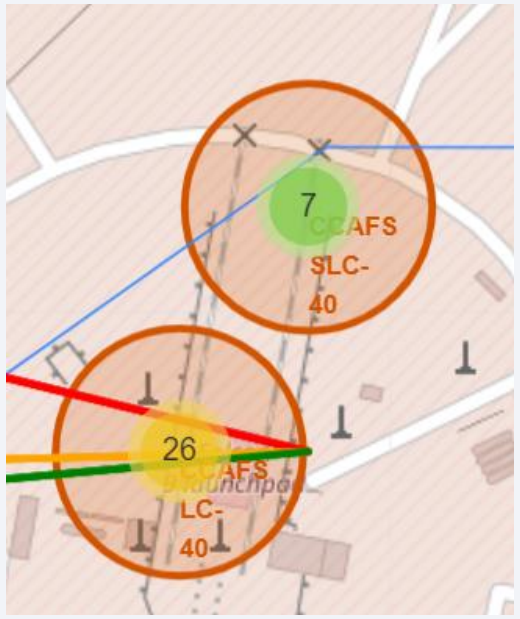


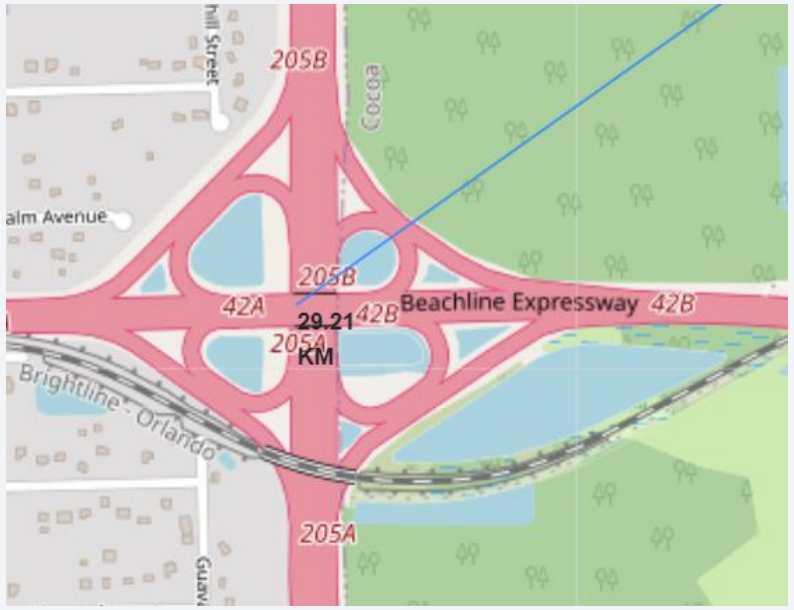
Color Coded Launch Markers

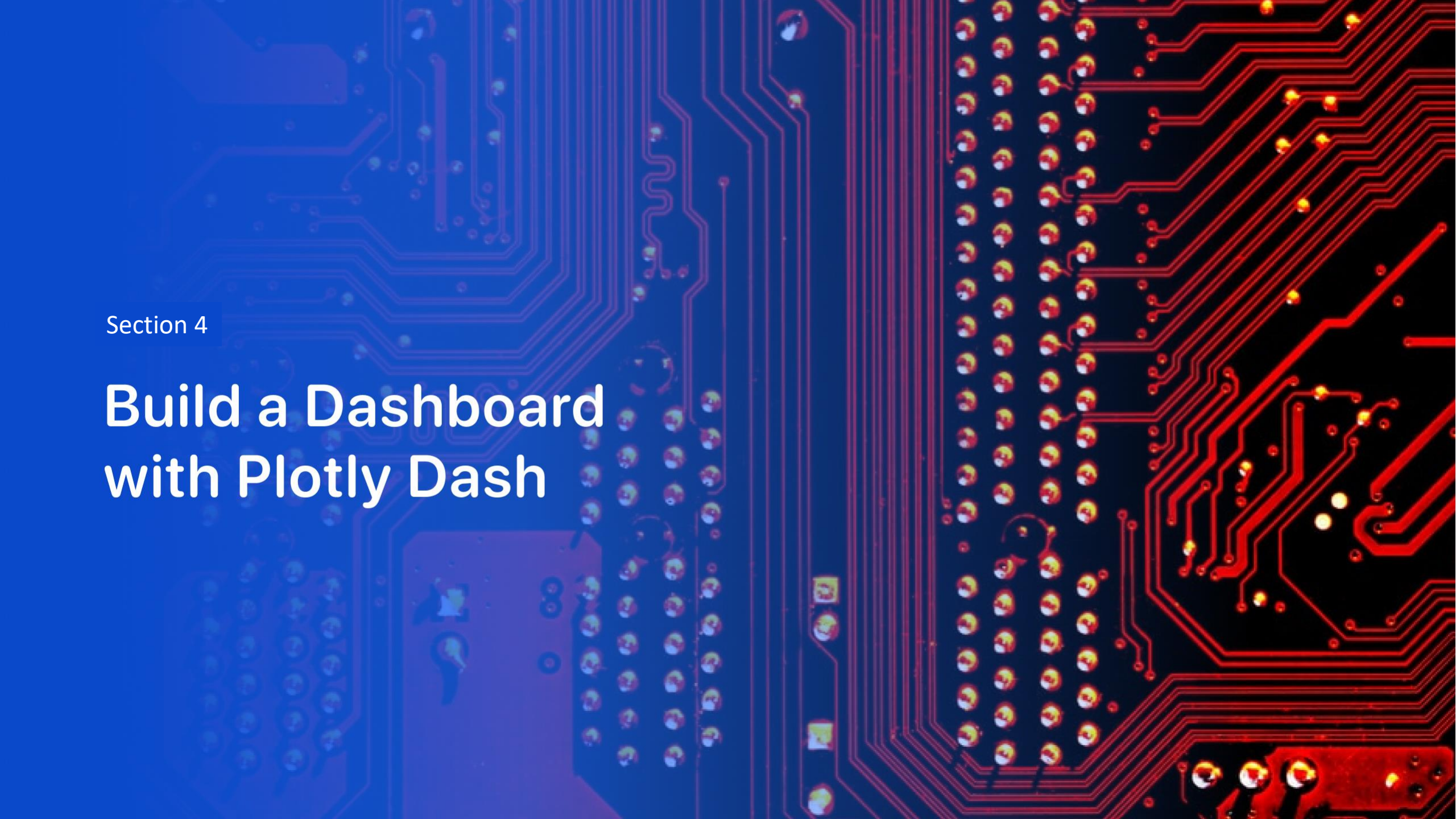
- Green Markers show the successful launches and Red Markers show the failure



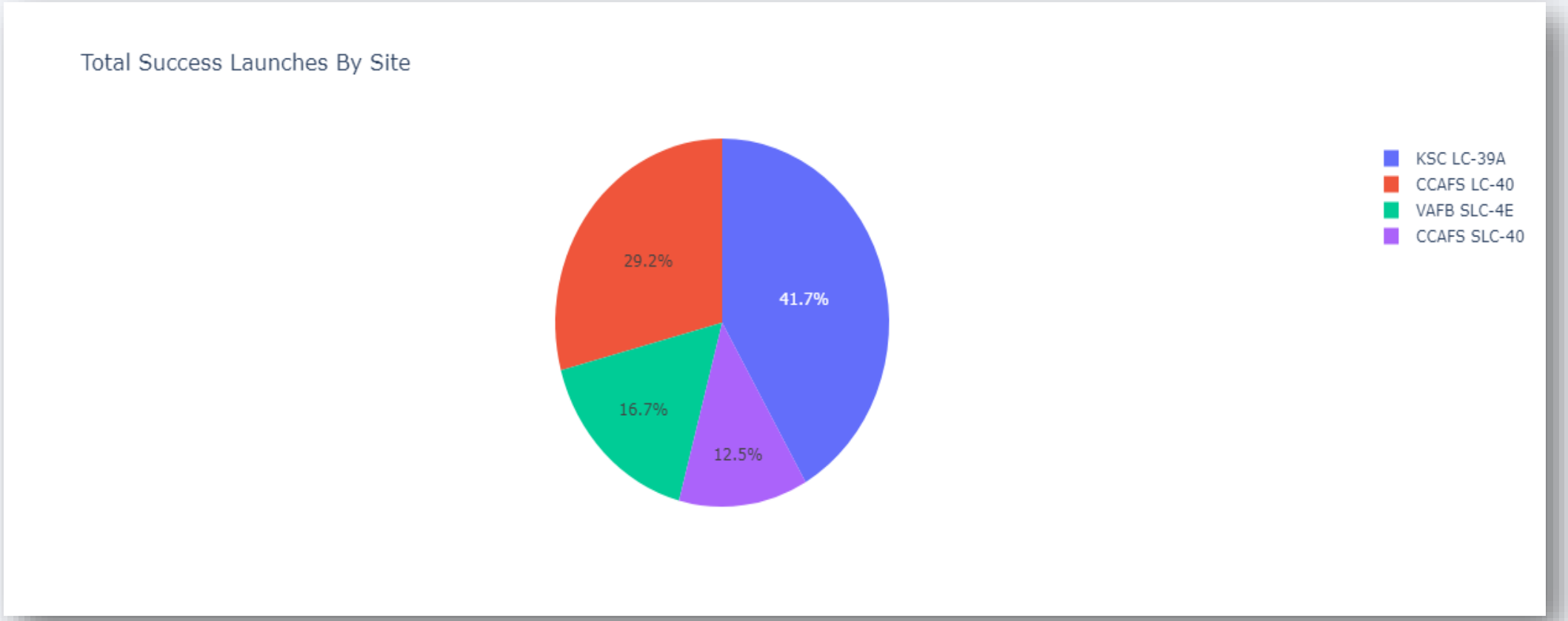
Launch Sites Distances to Landmarks





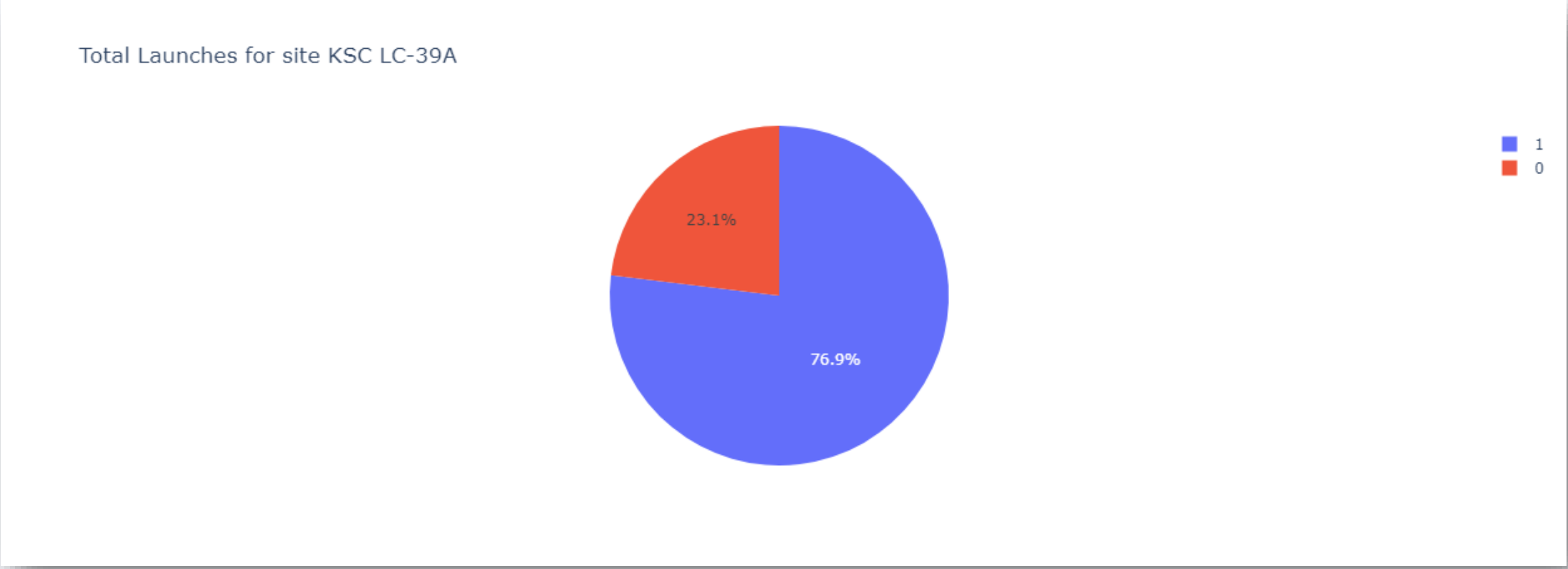

Section 4
Build a Dashboard with Plotly Dash

The success percentage by each sites



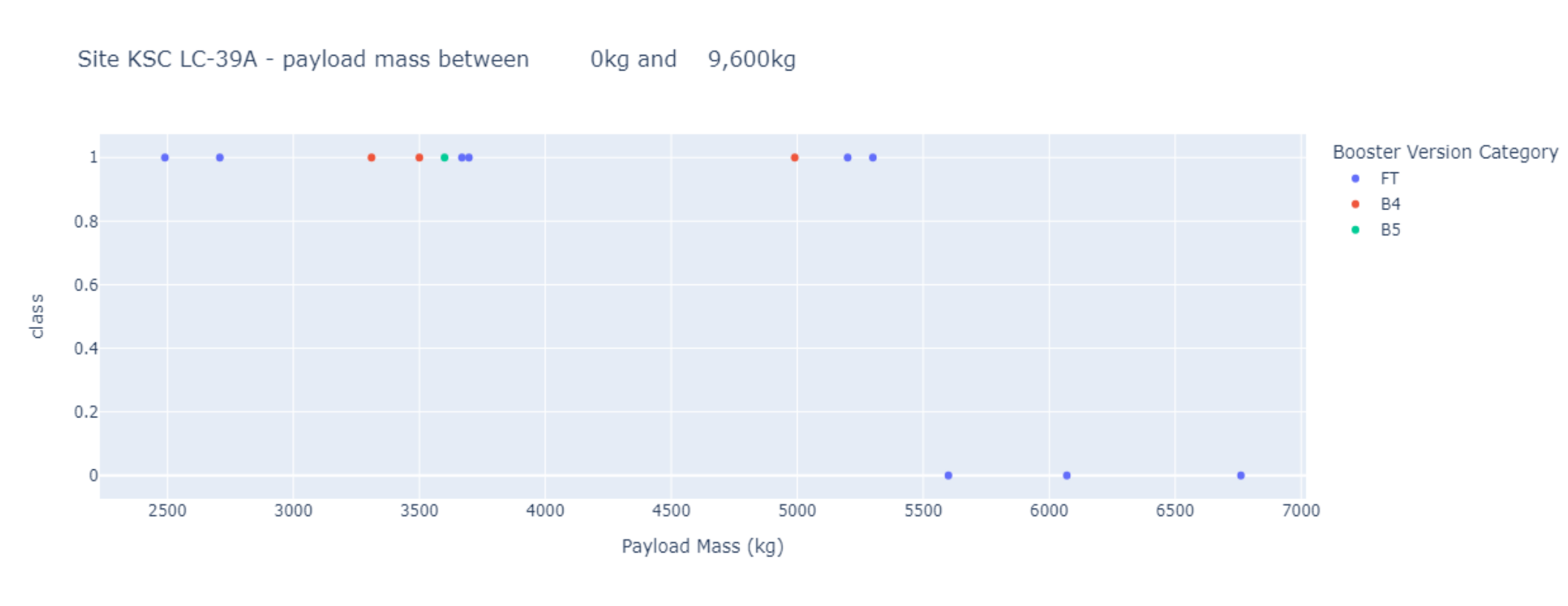
KSC LC-39A achieved the highest success percentage among all other sites.

The Highest Launch Success Ration



KSC LC-39A achieved a success rate of 76.9%, with a corresponding failure rate of 23.1%.

Payload Mass VS Success VS Booster Version Category



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All machine learning models have the same accuracy, F1-Score & Jaccard Score.
- The Decision Tree algorithm emerged as the top performer for the entire dataset.

Test Dataset

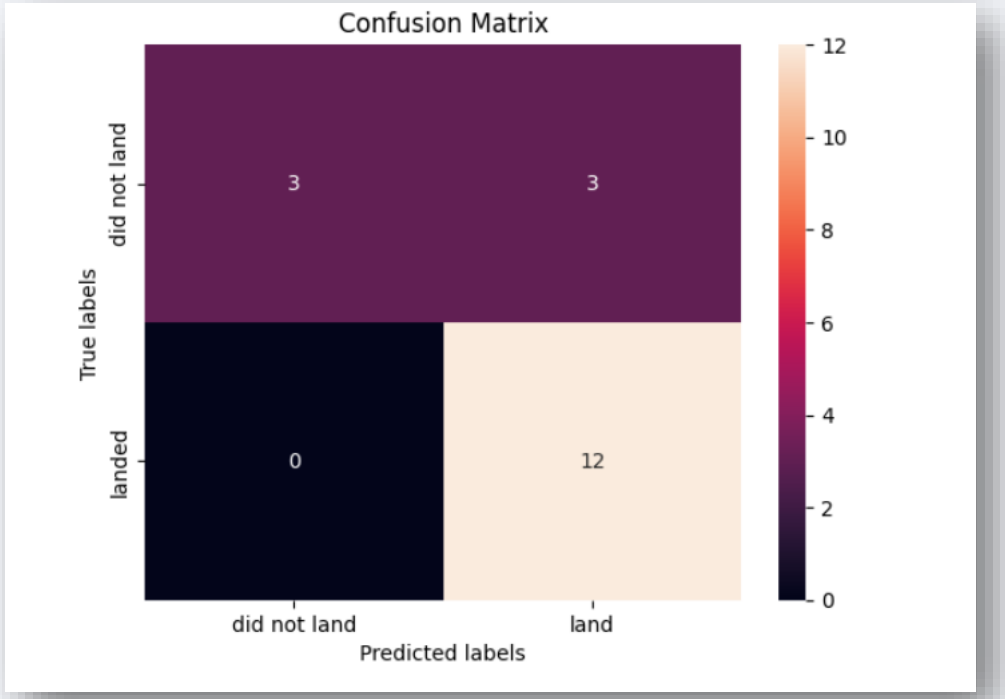
	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

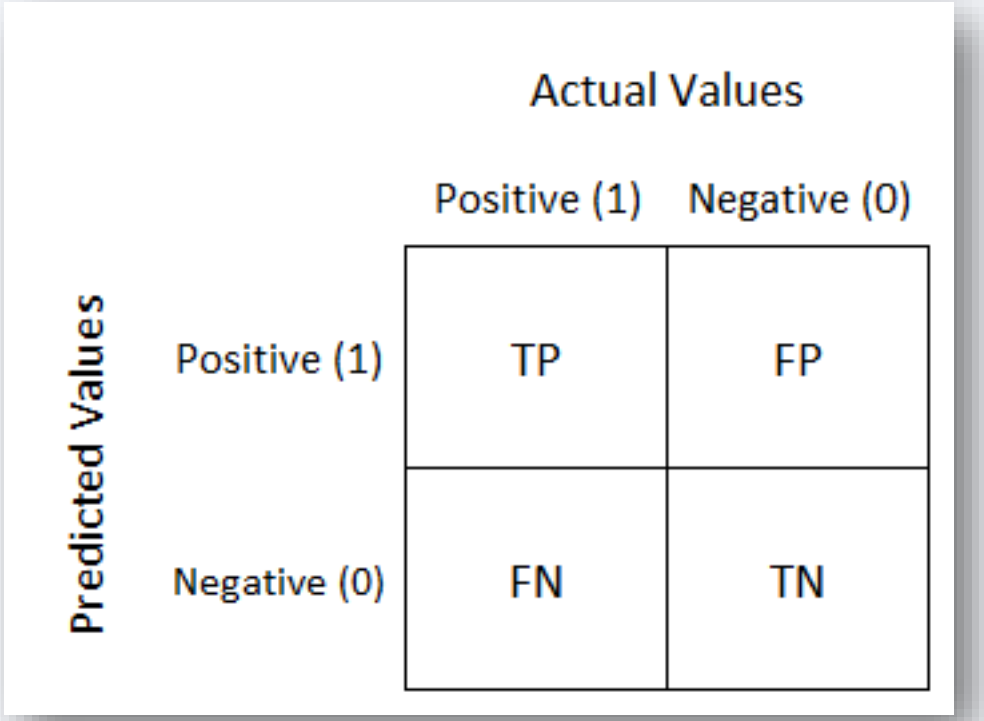
Whole Dataset

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.865672	0.819444
F1_Score	0.909091	0.916031	0.928000	0.900763
Accuracy	0.866667	0.877778	0.900000	0.855556

Confusion Matrix

- The models correctly predicted 12 successful landings and 3 unsuccessful landings. However, they also incorrectly predicted 3 successful landings when the true label was an unsuccessful landing.





Conclusions

- Here are the conclusions drawn:
  - Launch sites with higher flight volumes tend to have greater success rates.
  - The launch success rates from 2013 to 2020 started to increase.
  - Several machine learning algorithms were employed and the Decision Tree algorithm was identified as the best accuracy for this task.
  - SSO has a remarkable success rate of 100%, with more than one occurrence.

Appendix

- **Github:** https://github.com/dafinad/Coursera/tree/main/IBM_AppliedDataScienceCapstone
- **Instructor:**

Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

Thank you!