IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Lewis Njau
February 10, 2024

Outline

Executive Summary

- Summary of methodologies
  - ✓ Data Collection using SpaceX REST API.
  - ✓ Cleaning the data for analysis
  - ✓ Perform Exploratory data analysis (EDA) using SQL and Python libraries.
  - ✓ Create interactive visual analytics and dashboards.
  - ✓ Build and evaluate ML models (Classification) to predict landing outcomes.
- Summary of all results
  - ✓ Increasing success rate of launches over time.
  - ✓ ES-L1, GEO, HEO, and SSO orbits have the highest success rate.
  - ✓ Most launch sites are close to the coastline
  - ✓ All predictive models performed similarly on the test dataset with a score of 83% accuracy.

Introduction

Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problems you want to find answers

- Rate of successful landings over time
- Explore which factors contribute to a successful landing by utilizing feature engineering.
- Determine the best performing ML model in predicting landing success.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected using the SpaceX REST API and through web scraping Falcon 9 and Falcon Heavy records from Wikipedia.
- Perform data wrangling
  - The data was cleaned, split between train and test, and lastly applied hot encoding to prepare data for analysis and modeling.
- 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, and evaluate classification models

Data Collection

- Data was collected using the request (get) Python library to extract data from SpaceX REST API. The extracted data was then decoded into a JSON response and normalized.
- The data was cleaned, checking for duplicates or missing values. Web scraped Wikipedia for Falcon 9 and Falcon Heavy records using the BeautifulSoup Python library.
- The extracted records were parsed and converted into a Pandas data frame for analysis.

Data Collection – SpaceX API

- To extract SpaceX data using their API, I utilized the request (get) Python library. Converted the data into a JSON response and normalized the data for better analysis

- Notebook link: https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

[7]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

[8]:
```
response = requests.get(spacex_url)
```

[11]:
```
response.status_code
```

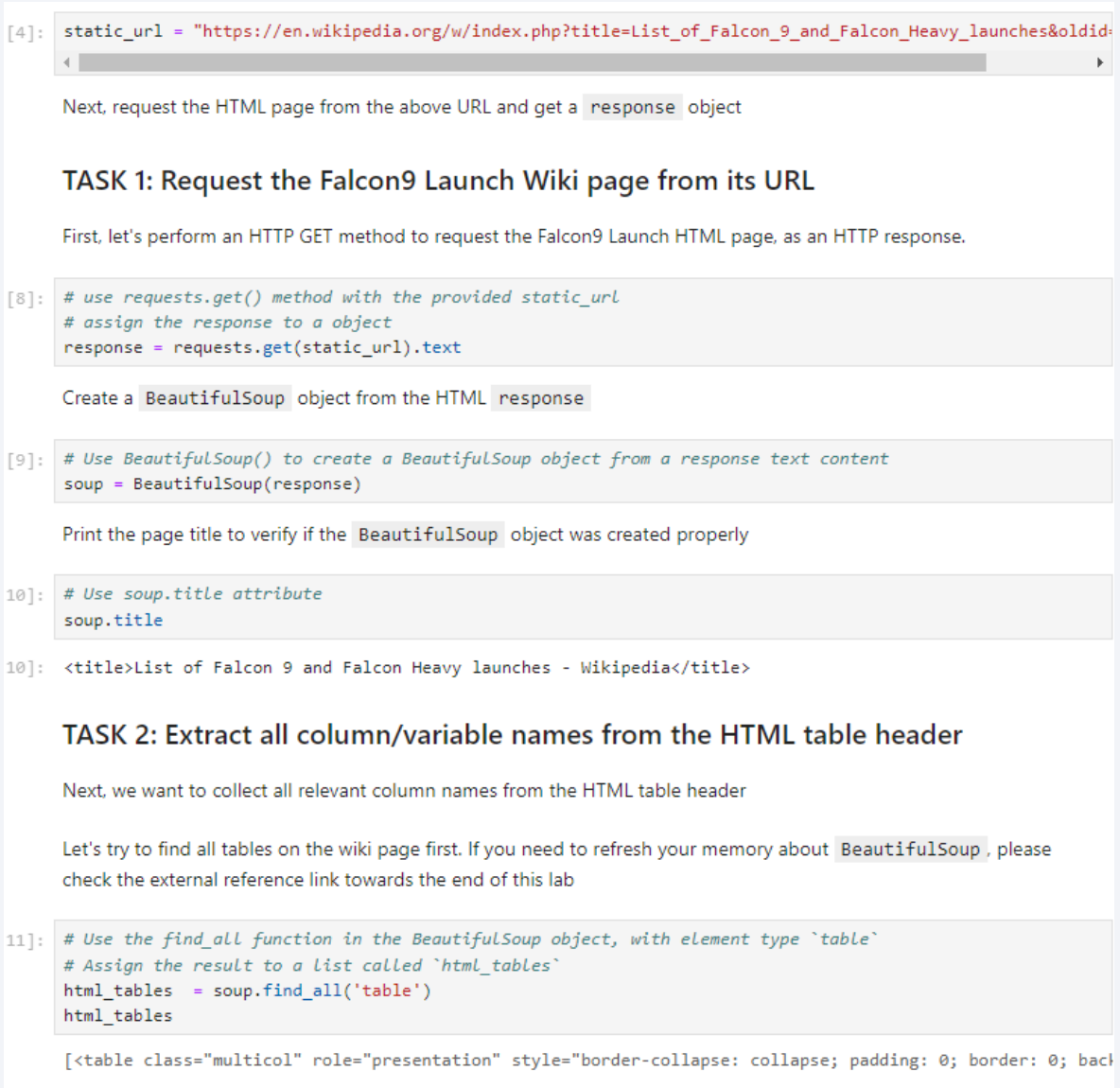
[11]:
```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

[18]:
```
# Use json_normalize meethod to convert the json result into a dataframe
response_json = response.json()
data = pd.json_normalize(response_json)
```

Data Collection - Scraping

- Web scraped Falcon 9 records from Wikipedia using the Beautiful Soup Python library. Collected the data from parsing the HTML tables the converted the data into a data frame.
- Notebook link: https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb



[4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

[8]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

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

[9]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

10]:
```
# Use soup.title attribute
soup.title
```

10]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

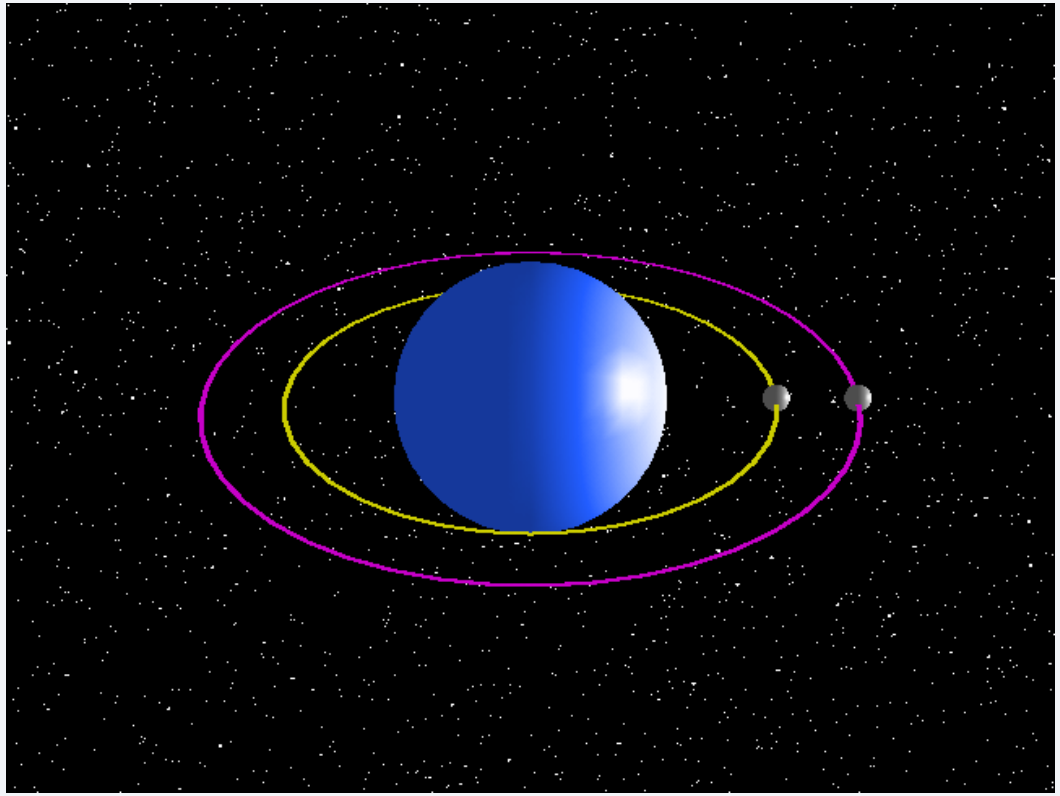
11]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables  = soup.find_all('table')
html_tables
```

```
[<table class="multicol" role="presentation" style="border-collapse: collapse; padding: 0; border: 0; bac
```

Data Wrangling

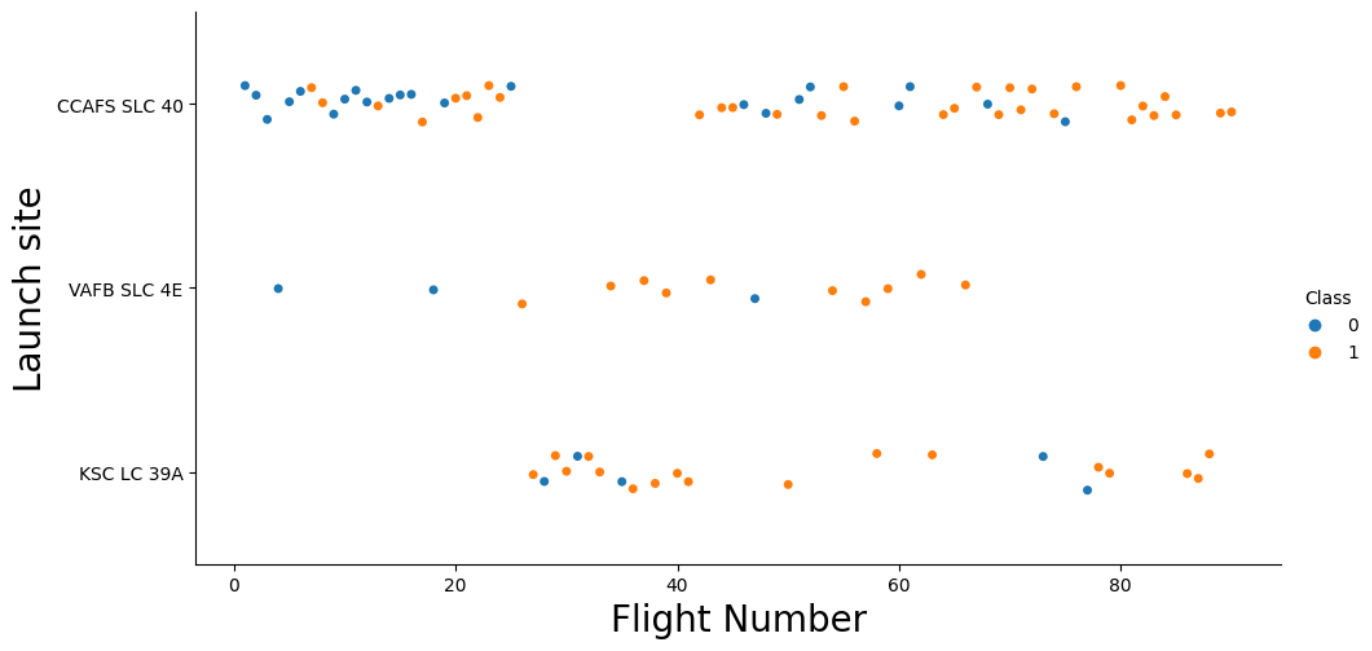
- Performed Exploratory Data Analysis to determine the number of launches for each site and the occurrence of each orbit to flight number.
- The dataset was analyzed to determine training labels for modeling and lastly the results were exported to a CSV file.

- Notebook link: https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Analyzed the data by examining connections between flight numbers and launch sites, payloads and launch sites, the success rates of various orbits, flight numbers, and orbit types, and the annual trend in launch successes.
- Charts:
  - ✓ Flight Number vs Payload Mass
  - ✓ Flight Number vs Launch Site
  - ✓ Payload Mass vs Orbit Type
  - ✓ Payload Mass vs Launch Site



Notebook Link:

https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-dataviz-ipynb-jupyterlite.ipynb

EDA with SQL

- Through the implementation of Exploratory Data Analysis (EDA) and SQL, I gained valuable insights into the dataset. Specifically, I formulated SQL queries to ascertain various details such as:
  - Distinct launch sites used in the space mission
  - Aggregate payload mass of boosters deployed
  - Mean payload mass for the Falcon 9
  - Overall counts of mission outcomes, categorized into successes and failures
  - Details of unsuccessful landings on the drone ship (including booster versions and names of launch sites involved)
- Notebook link:
  - https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- All launch locations were plotted on the folium map, with various map elements like markers, circles, and lines indicating whether launches at each site were successful or not.
- The outcome of each launch, being either a failure or a success, was categorized into two classes: class 0 for a failure and class 1 for a success.
- Through the use of clusters of color-coded markers, I pinpointed launch sites with a notably high rate of successful launches. I measured the distance from each launch site to various nearby features and explored several questions, such as whether the launch sites are situated close to railways, highways, or coastlines, and if they maintain a specific distance from urban areas.

Build a Dashboard with Plotly Dash

- The dropdown list with launch sites allows users to select filter results for each launch site. The pie chart displays all successful and unsuccessful launches as a percent of the total.

- Notebook link:
  - https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/Capstone_Project_Dash.py

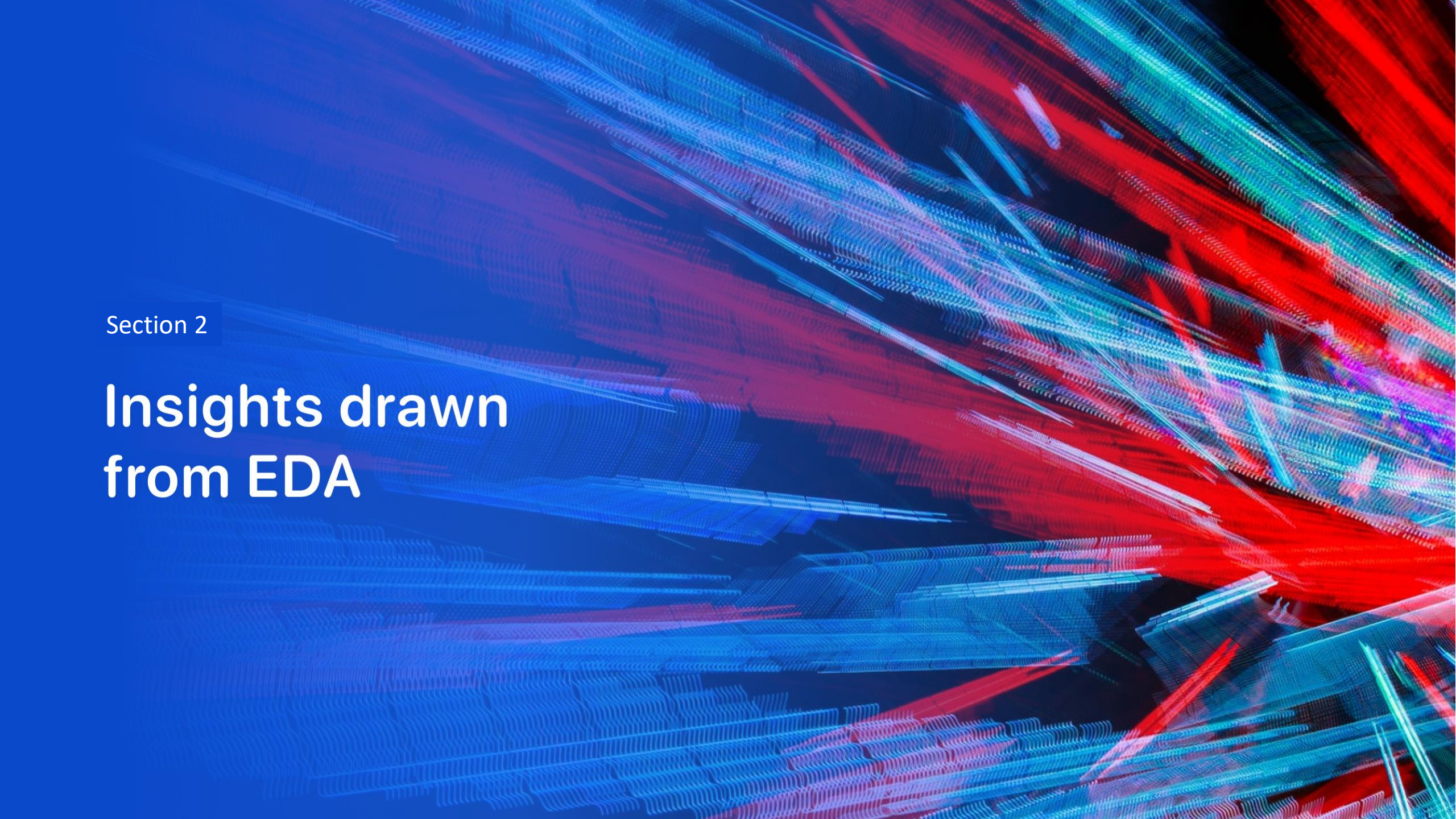
Predictive Analysis (Classification)

- Created a Numpy array from the Class column. I standardized the dataset with the StandardScaler function from the Sklearn Python library and transformed the dataset. Using the 'train_test_split' function, I split the data into two sets (train set and test set).
- I created and applied the GridSearchCV object on different classification models such as Logistic regression, Support Vector Machine (SVC), Decision tree, and K-Nearest Neighbor.
- I calculated the accuracy on the test dataset using the method 'score' for all models, and then lastly, I identified the best model using F1_score, Jaccard_Score, and Accuracy.

- Notebook link:
  - https://github.com/LouN99/Applied-Data-Science-Capstone-Project/blob/main/spacex-machine-learning-prediction-part-5-jupyterl.ipynb

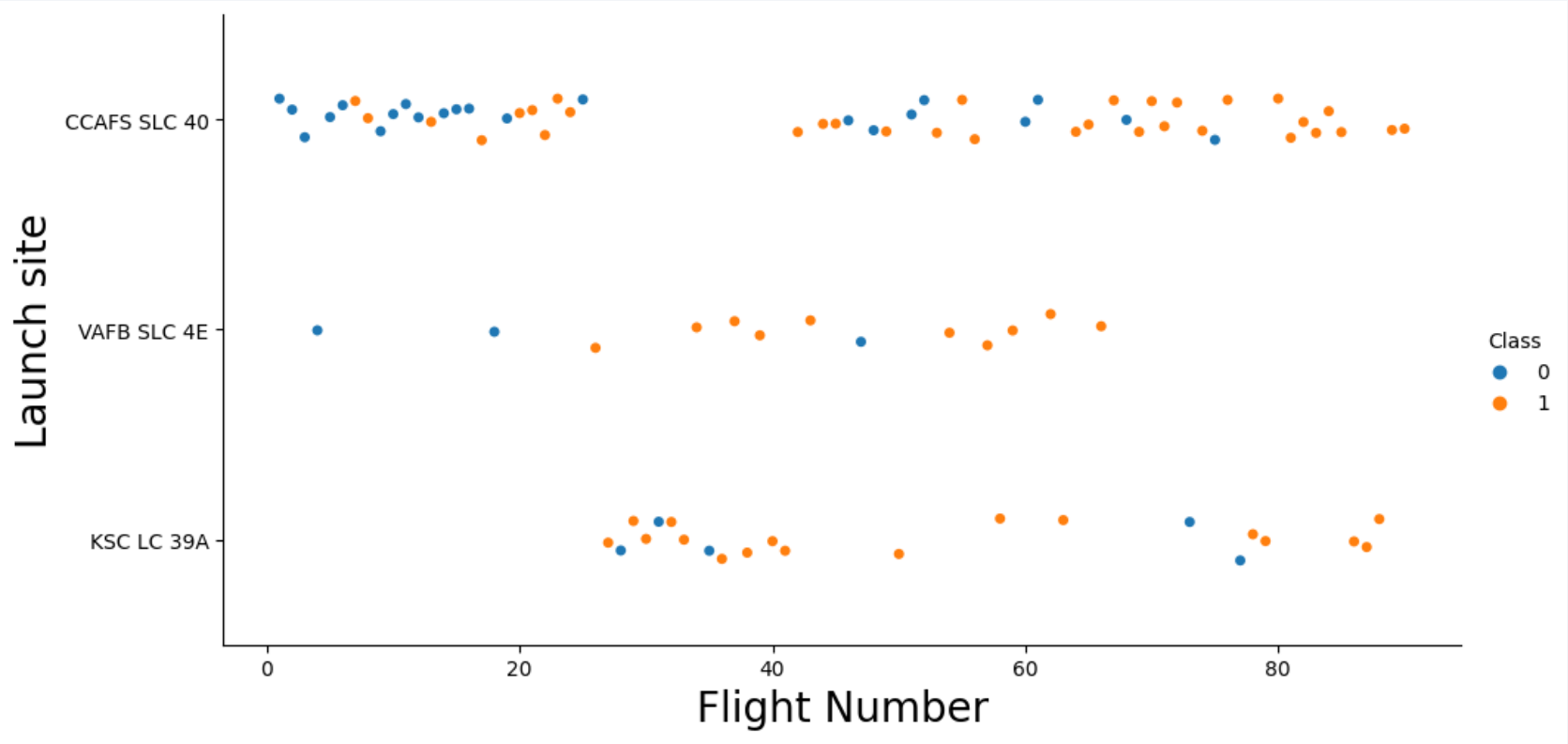
Results

- Exploratory data analysis results
  - KSC LC 39A and VAFB SLC 4E have a success rate of 77%
  - Launch success has improved over time
  - Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate
- Interactive analytics demo in screenshots
  - All launch sites are close to the coastline
  - Using the Folium map, launch sites are located far enough from cities, highways, and railways to protect urban infrastructure in the event of a failed launch.
- Predictive analysis results
  - The Decision tree model performed best with a score of 86%
  - All models performed similarly on the test set with an 83% accuracy score.


Section 2
Insights drawn from EDA

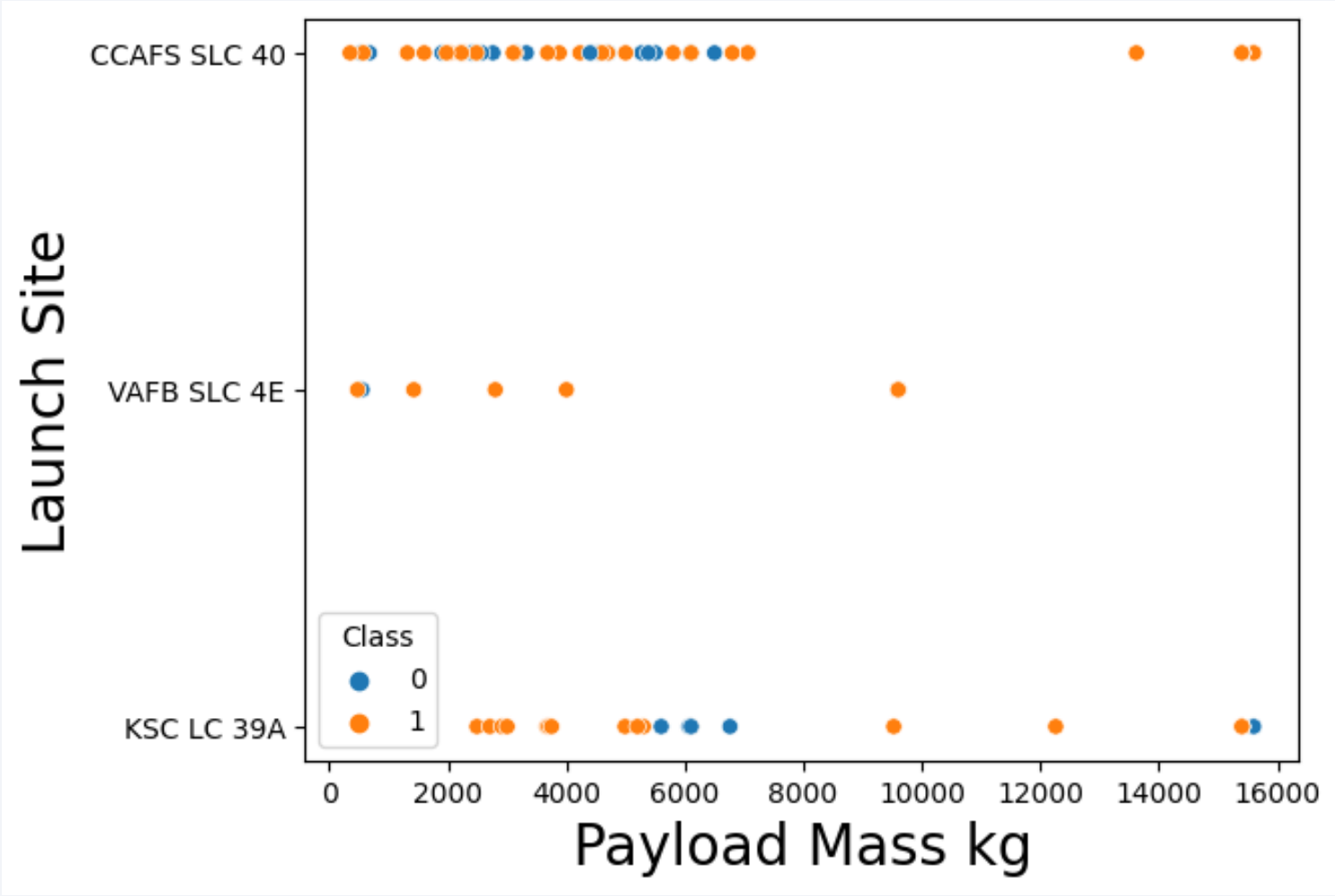
Flight Number vs. Launch Site

- Earlier flights had a lower success rate and were mostly launched at the CCAFS SLC 40 launch site.
- VAFB SLC 4E and KSC LC 39A launch sites have a higher success rate



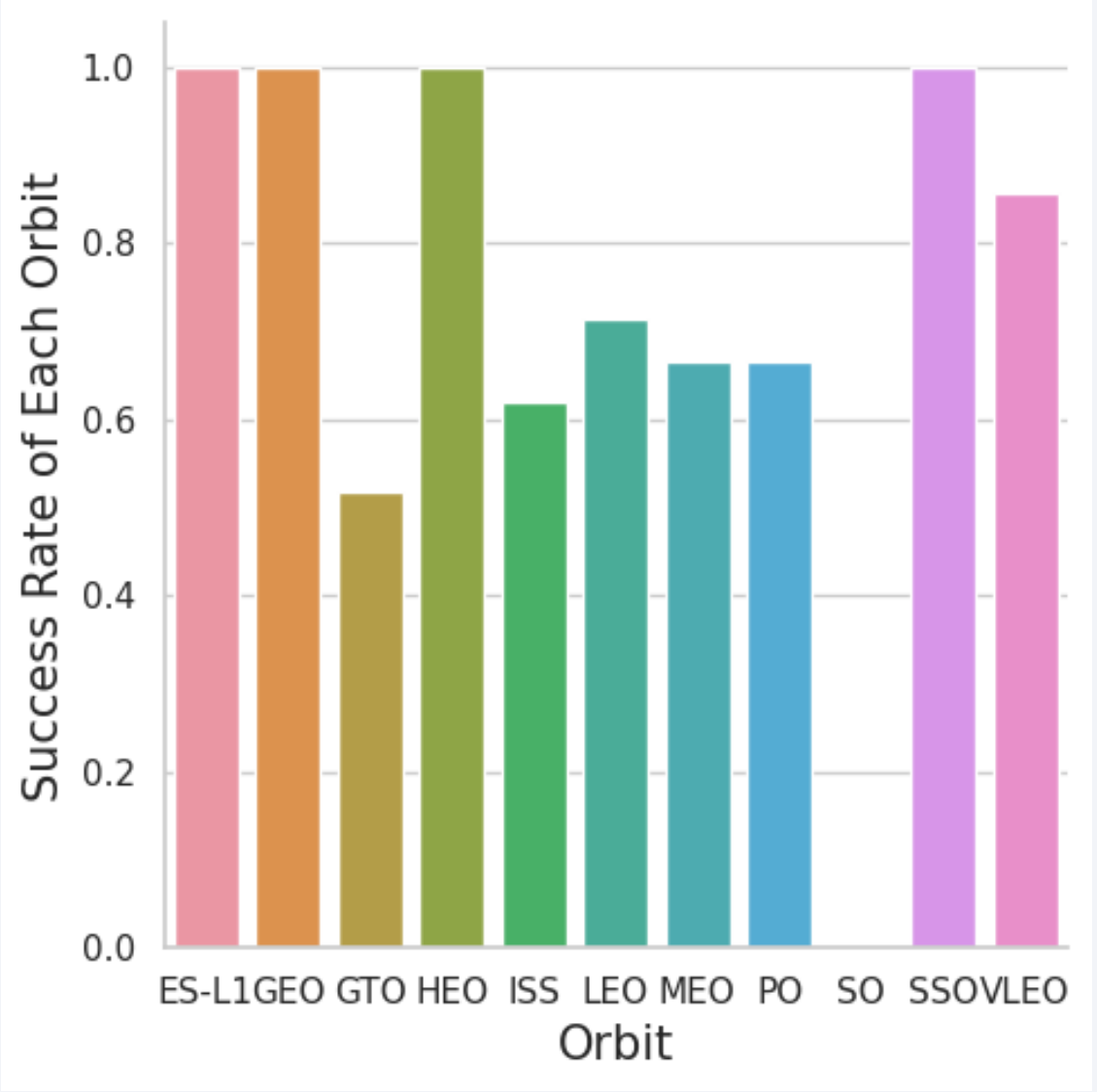
Payload vs. Launch Site

- On average, the higher the payload mass, the higher the success rate
- There are no rockets launched for payload mass greater than 100,000 kg at the VAFB SLC 4E launch site.



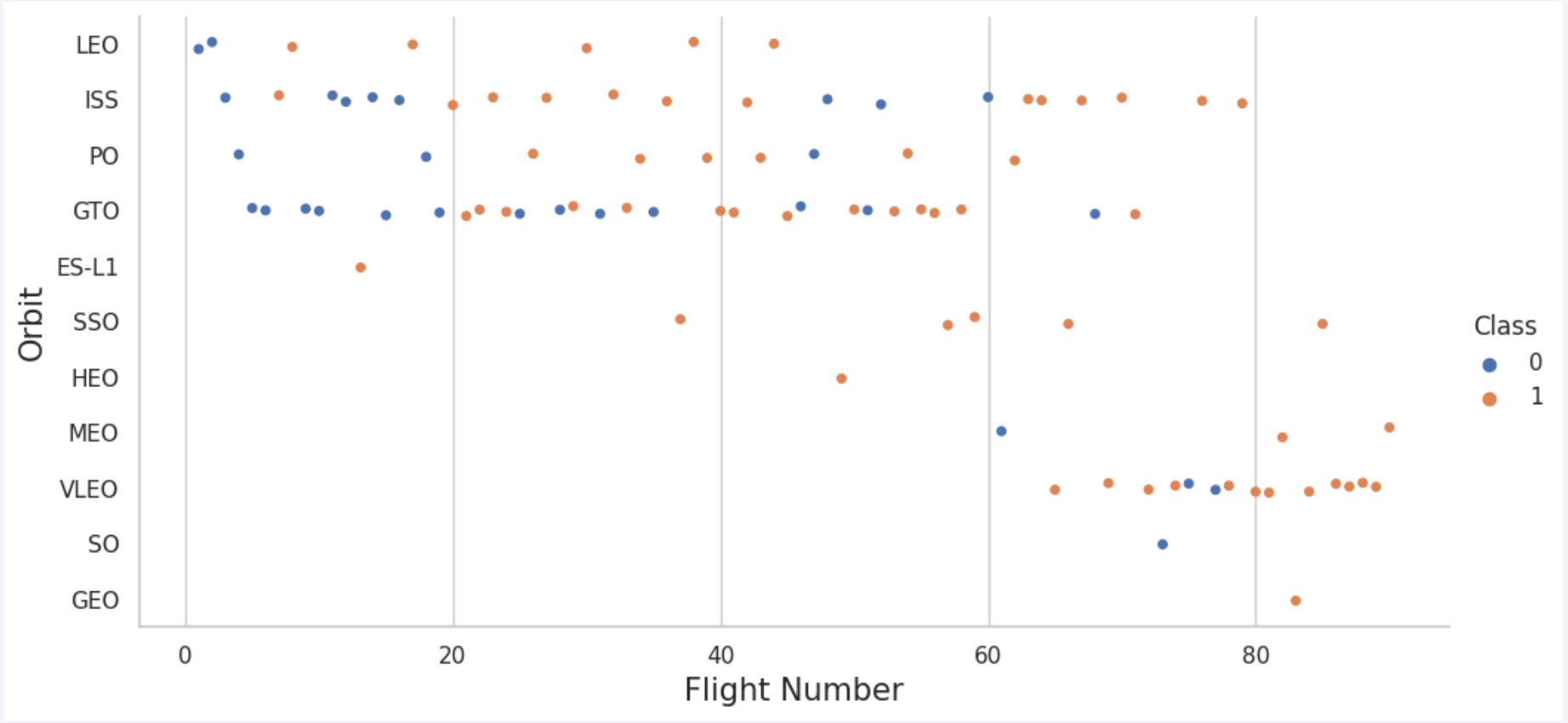
Success Rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO, and SSO had a success rate of 100%



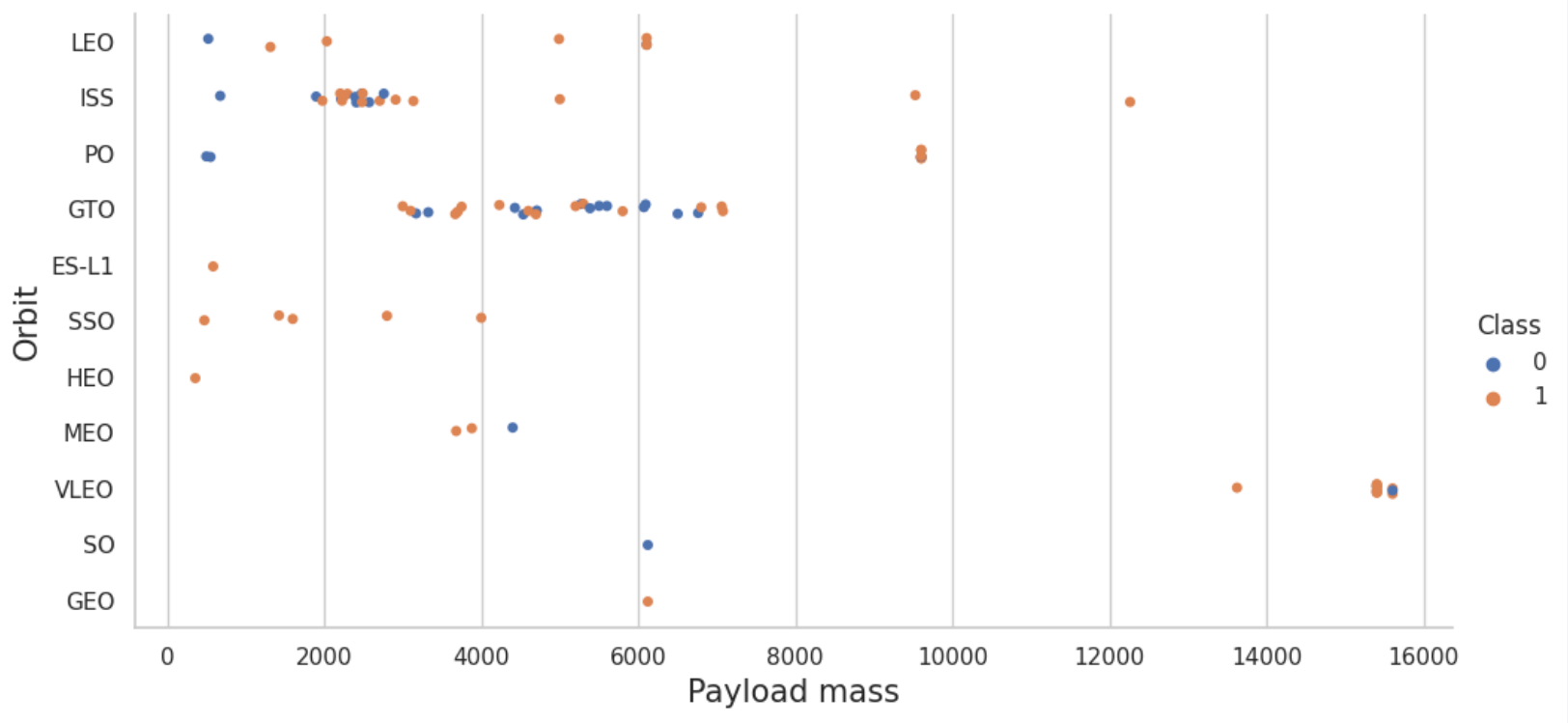
Flight Number vs. Orbit Type

- Overtime, the success rate has been increasing with the number of flights for each orbit, which is highly evident for the LEO orbit.



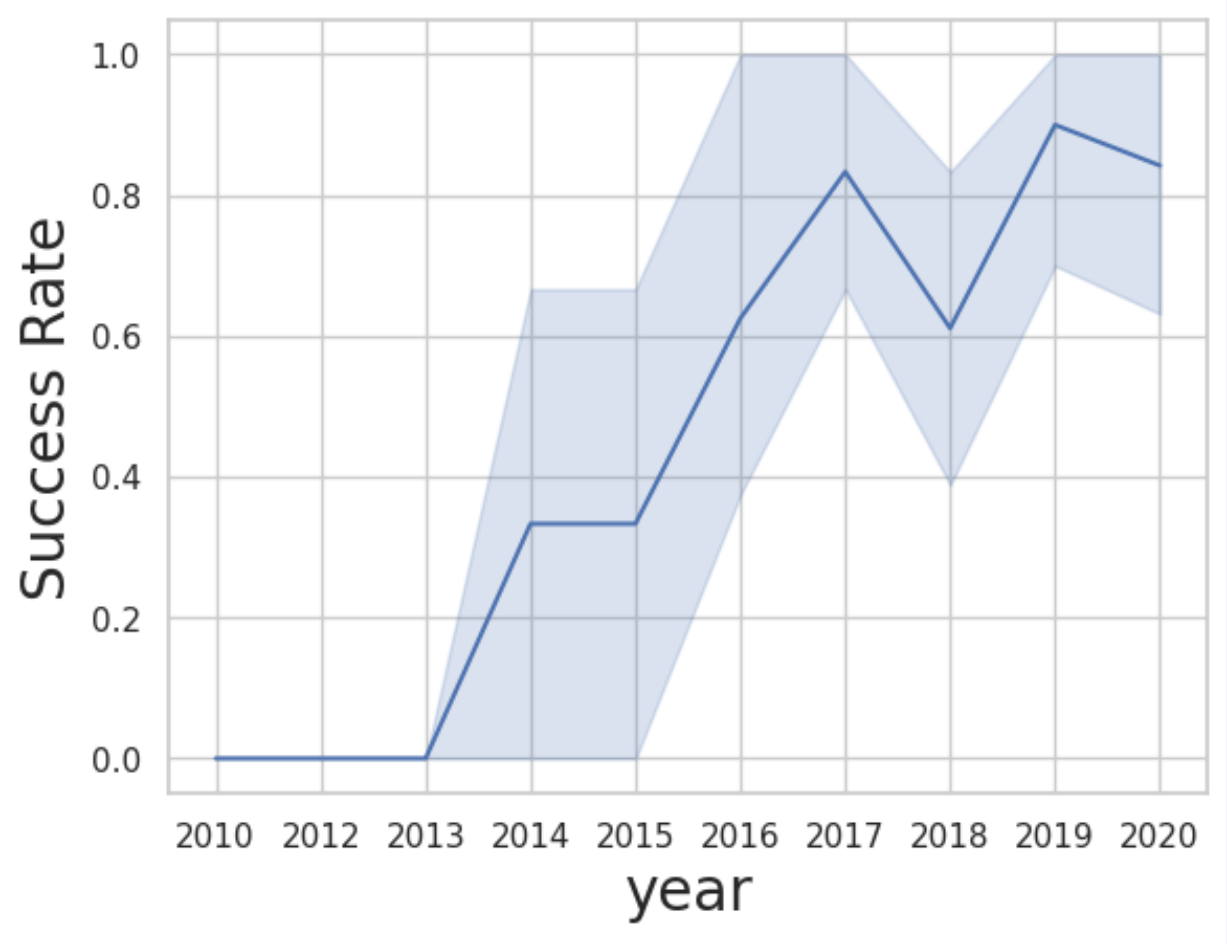
Payload vs. Orbit Type

- Heavy payloads are more suitable with LEO, ISS, and PO orbits, while the GTO orbit presents a mixed relationship between payload and orbit type.



Launch Success Yearly Trend

- Overall, the success rate has improved since 2013
- The highest decline in success rate is observed to be between 2017-2018



All Launch Site Names

- Launch site names:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
%%sql
select
distinct "Launch_Site"
from SPACEXTABLE;
```

```
 * 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'

- The query displays 5 records where the launch sites begin with ‘CCA’

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

```
%%sql
select *
from SPACEXTABLE
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 total payload mass carried by boosters launched by NASA is 45,596 kg

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

```
%%sql
SELECT CUSTOMER, SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass(Kg)"
FROM SPACEXTABLE
WHERE CUSTOMER == "NASA (CRS)";
```

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

Customer	Total Payload Mass(Kg)
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2,928.4 kg

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

```
%%sql
SELECT BOOSTER_VERSION, AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass(kg)"
FROM SPACEXTABLE
WHERE BOOSTER_VERSION == "F9 v1.1";
```

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

Booster_Version	Average Payload Mass(kg)
F9 v1.1	2928.4

First Successful Ground Landing Date

- The first successful ground landing date was on December 22, 2015

```
%%sql
SELECT MIN(DATE) AS "First Success Date", LANDING_OUTCOME
FROM SPACEXTABLE
WHERE LANDING_OUTCOME == "Success (ground pad)";
```

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

First Success Date	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of the boosters which have success in drone ships and have a payload mass between 4000 kg and 6000 kg are:
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2

```
%%sql
SELECT BOOSTER_VERSION, 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	Landing_Outcome
F9 FT B1022	Success (drone ship)
F9 FT B1026	Success (drone ship)
F9 FT B1021.2	Success (drone ship)
F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes was 100 with one among the 100 with an unclear payload status.
- The total number of failure mission outcomes was 1

```
%%sql
SELECT DISTINCT(MISSION_OUTCOME), COUNT(*)
FROM SPACEXTABLE
GROUP BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

- A subquery was used to acquire the list of names of the booster versions that have carried the maximum payload mass

```
%%sql
SELECT DISTINCT(BOOSTER_VERSION)
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
                           FROM SPACEXTABLE);
```

```
 * 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

- The query consists of the ‘WHERE’ clause, ‘SUBSTRING’, and the ‘AND’ conditions to filter for failed landing outcomes in drop ship, their booster versions, and launch site names for the year 2015.

```
%%sql
SELECT SUBSTRING(DATE, 6, 2) AS "Month", MISSION_OUTCOME, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTABLE
where SUBSTRING(DATE, 0, 5) ='2015'
AND LANDING_OUTCOME == "Failure (drone ship)";
```

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

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

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

- We extracted the details of landing outcomes, applying a filter to include only those occurring between June 4, 2010, and March 20, 2017.
- The data was then organized by landing outcomes using the GROUP BY clause, and sorted in descending order with the ORDER BY clause to prioritize the most frequent outcomes.

```
%%sql
SELECT LANDING_OUTCOME, COUNT(*) AS "QTY"
FROM SPACEXTABLE
WHERE DATE BETWEEN "2010-06-04" AND "2017-03-20"
GROUP BY LANDING_OUTCOME
ORDER BY "QTY" DESC;
```

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

Landing_Outcome	QTY
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

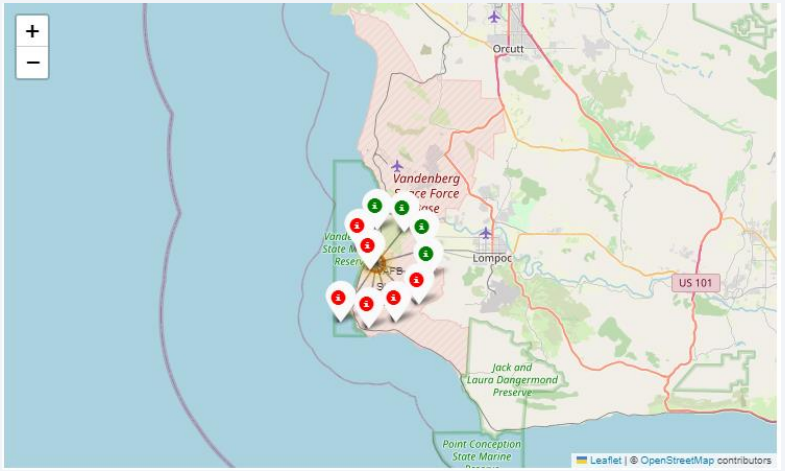

Section 3
Launch Sites
Proximities Analysis

All Launch Sites

- The Folium map displays all the launch sites in the United States, they are all along the coastlines of Florida and California.

Launch Sites With Color Markers

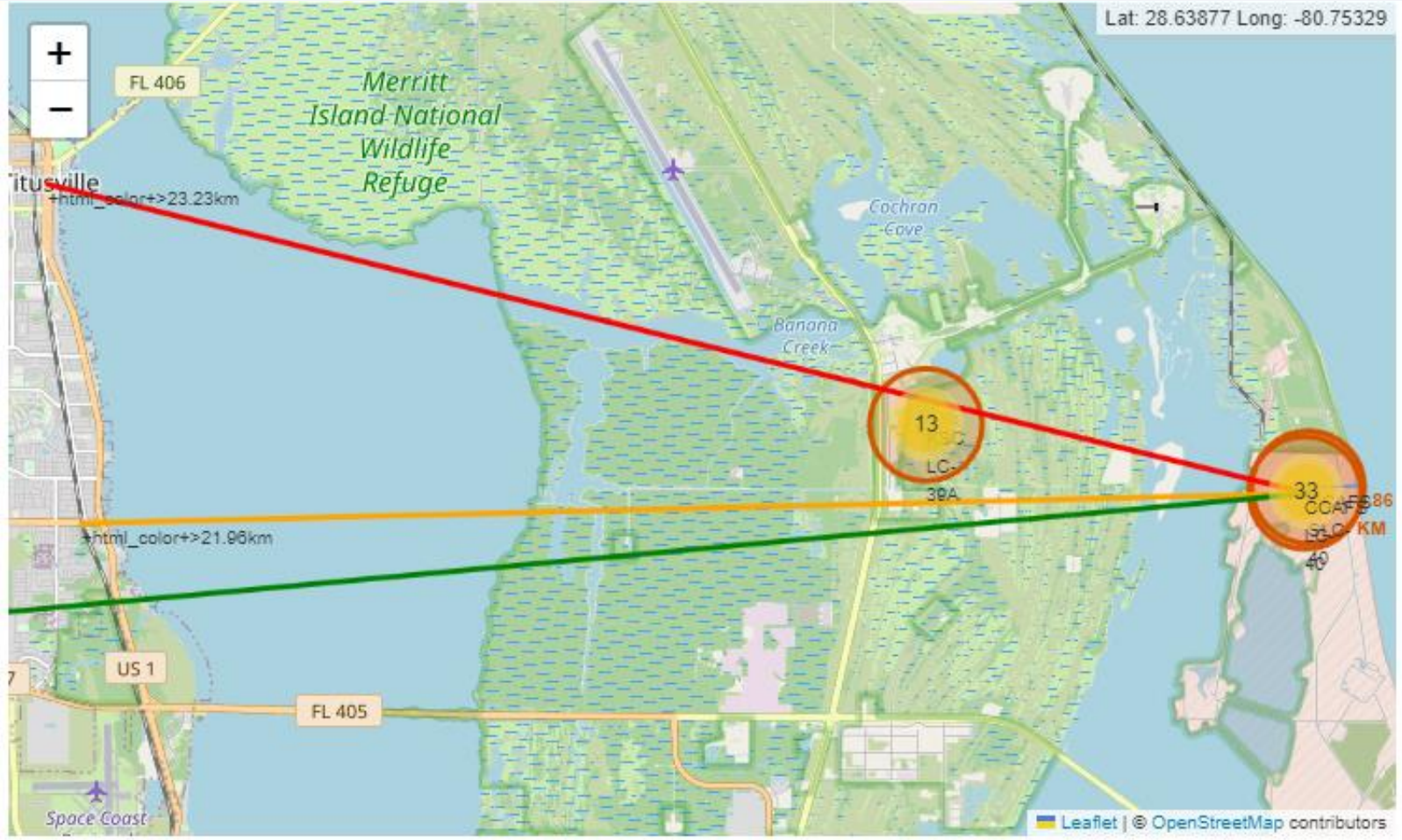
- Florida launch sites:
  - Green makers for successful launches
  - Red markers for unsuccessful launches

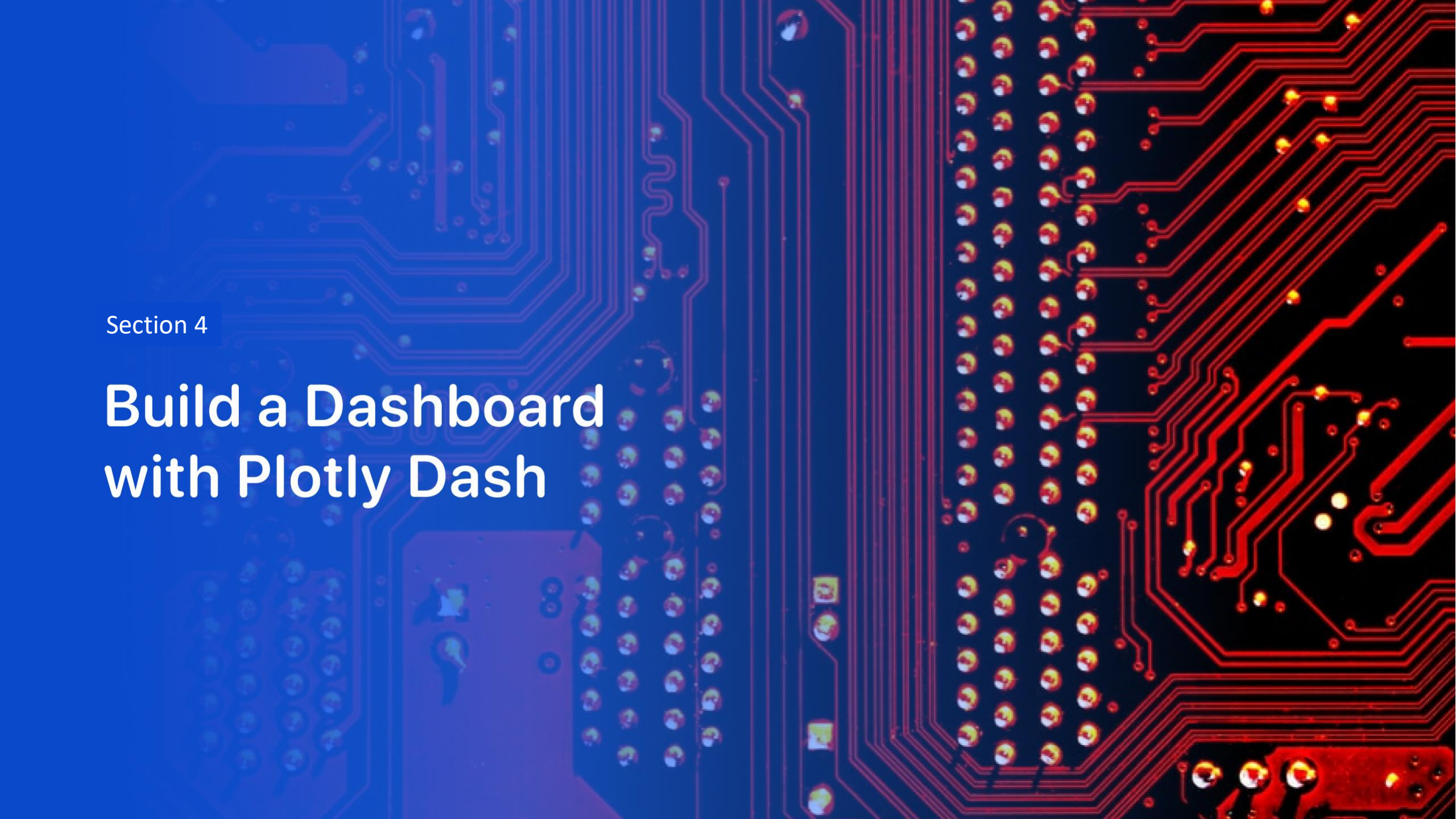


- California launch sites:
  - Green marker for successful launches.
  - Red markers for unsuccessful launches.

Distance to Proximities

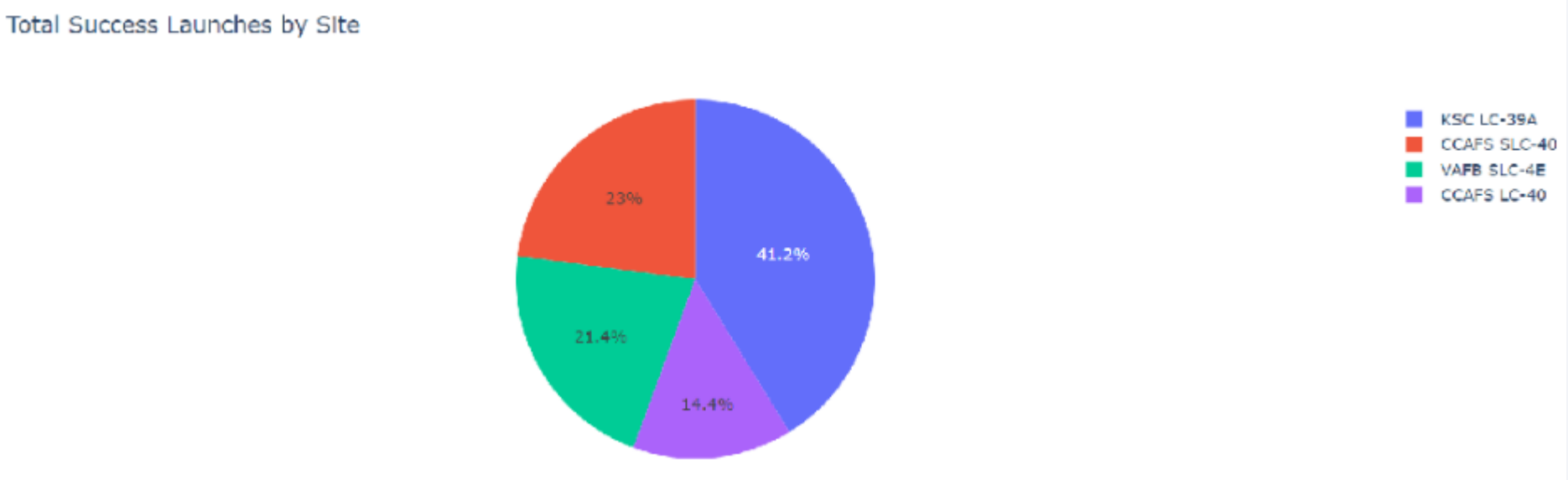
- Proximity to railways: 21.96 km
- Proximity to highways: 26.88 km
- Proximity to coastlines: 0.86 km
- Proximity to cities: 23.23 km




Section 4
Build a Dashboard with Plotly Dash

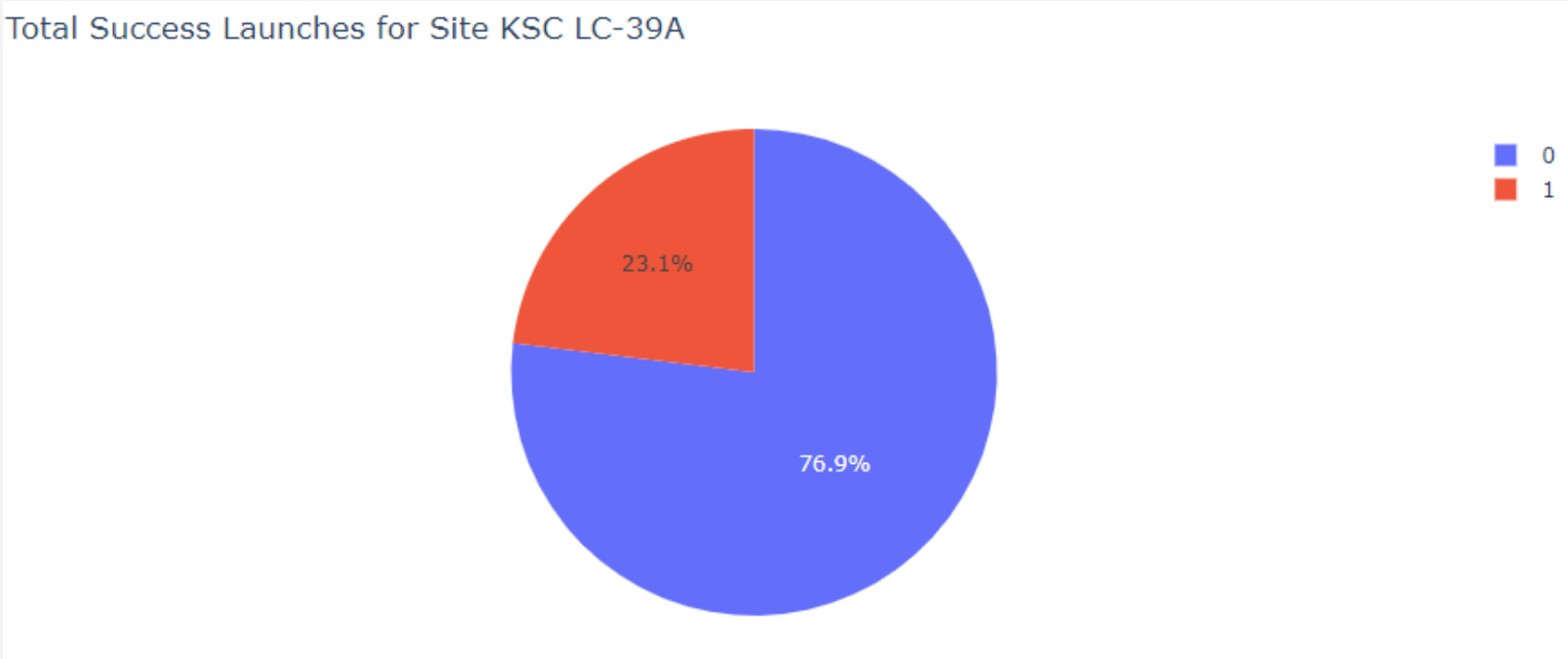
Total Success Launches by Site

- The KSC LC -39A launch site had the most successful launches from all sites



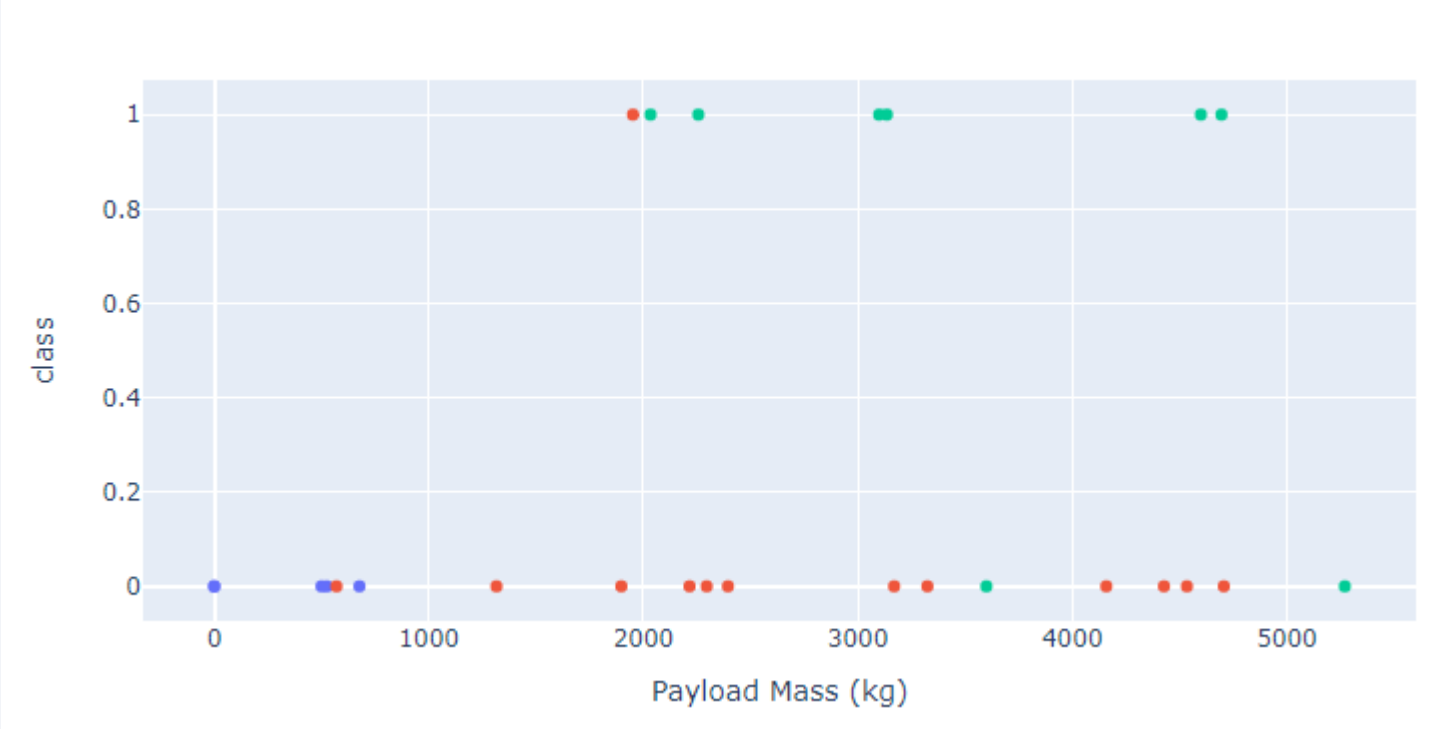
Launch Site With The Highest Launch Success Ratio

- KSC LC-39A launch site has the highest success ratio with a success rate of 76.9% and a failure rate of 23.1%



Payload vs Outcome For All Sites

- Payloads between 2,000 kg and 5,000 kg have the highest success rate.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree model performed best with a score of 86%
- All models performed similarly on the test set with an 83% accuracy score across the board

[61]:

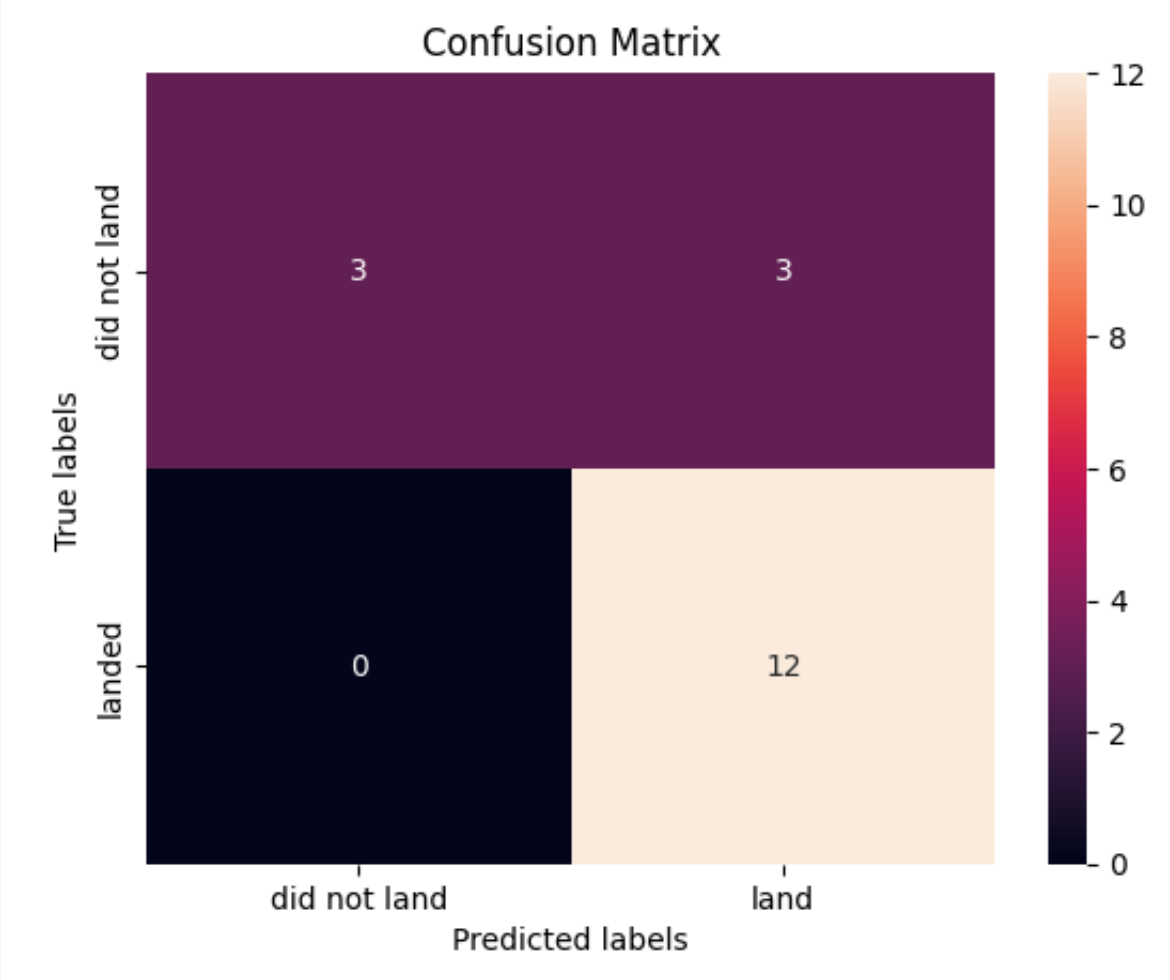
	Model	Best Score
0	Logistic Regression	0.846429
1	Support Vector Machine	0.848214
2	K-Nearest Neighbor	0.848214
3	Decision Tree	0.862500

[62]:

	Model	Accuracy Score
0	Logistic Regression	0.833333
1	Support Vector Machine	0.833333
2	K-Nearest Neighbor	0.833333
3	Decision Tree	0.833333

Confusion Matrix

- The Confusion Matrix had an output of 3 false positives, which means some unsuccessful landings were classified as successful landings by the classification model (Decision Tree).



Conclusions

- The classification models performed similarly on the test set. However, the Decision Tree model performed slightly better with an accuracy score of 86%.
- All the launch sites were located near coastlines, far away from urban infrastructure.
- On average, the success rate of launches increased over time
- The launch site, KSC LC-39A has the highest success rate with a rate of 100% for launches less than 5,000 kg
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate

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!